

DQ11

BASIC LOGIC TEST PART 1
CZDQAD0

AH-8603D-MC
COPYRIGHT 74-78
FICHE 1 OF 1

JAN 1979
digital
MADE IN USA

The microfiche card displays a grid of 100 frames, arranged in 10 rows and 10 columns. Each frame contains a small-scale image of a logic test component, likely a truth table or a circuit diagram. The text and diagrams within the frames are too small to be legible. The right half of the card is a dark, uniform area, possibly representing the back of the microfiche or a blank section.

IDENTIFICATION

PRODUCT CODE: AC-8601D-MC
PRODUCT NAME: CZDQAD0 BLT PRT 1
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 ON 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 8K 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 8K OF MEMORY.
LOCATION 1400 THRU 1600 ARE ESPECIALLY TO BE NOTED AND TO BE UNTOUCHED BY OPERATOR AFTER DQ11 TRIAL PROGRAM HAS BEEN EXECUTED. OR AFTER THE 'AUTO SIZING' HAS BEEN DONE.

3. LOADING PROCEEDURE

3.1 METHOD

ALL PROGRAMS ARE IN ABSOLUTE FORMAT AND

ARE LOADED USING THE ABSOLUTE LOADER.

ABSOLUTE LOADER STARTING ADDRESS *500

MEMORY *
SIZE

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

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

3.1.2 THEN START

4. STARTING PROCEEDURE

A. LOAD LOC. 200

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

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

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

C.THEN START

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

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

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

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

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

4.1 CONTROL SWITCH SETTINGS

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

CONTROL :

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

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

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

4.1.2 SWITCH REGISTER RESTRICTIONS

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

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

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

SW 09 LOOP ON CURRENT DATA:
THIS SWITCH WILL ONLY WORK IF
CALL '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= (REFER TO SECTION 4.1 FOR OPERATOR OPTION)

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

SEE SECTION 4. (PLEASE)

7.2 OPERATING RESTRICTIONS

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

8. MISCELLANEOUS

8.1 EXECUTION TIME

8.2 PASS COMPLETE

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

END PASS AC-8601D-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) 1500/0000000000011111
MEANS THAT DQ11 NO. 00,01,02,03,04
WILL BE TESTED.

EXAMPLE:
(DQACTV) 1500/0000000000010001
MEANS THAT DQ11 NO. 00,04
WILL BE TESTED.

DQCSR (1506) CONTAINS THE RECEIVER CSR OF THE
CURRENT DQ11 UNDER TEST.

DQSTAT (1510) CONTAINS THE STATUS OF THE CURRENT
DQ11 UNDER TEST.

BIT 15 SET: TWO SYNC CHARS/ONE SYNC CHAR
BIT 14 SET: TEST JUMPER INSTALLED/NOT INSTALLED
BIT 13 SET: BB OPTION INSTALLED/NOT INSTALLED
BIT 12 SET: BA OPTION INSTALLED/NOT INSTALLED
BIT 11 SET: ACTIVE ON FIRST NON-SYNC/ACTIVE AFTER NO. OF SYNC
BIT 10 SET: AB OPTION INSTALLED/NOT INSTALLED
BIT 09 SET: ODD VRC/EVEN VRC

BIT 00-08 VECTOR 'A' OF DEVICE

8.5 *** METHOD OF AUTO SIZING ***

8.5.1 FINDING THE CONTROL STATUS REGISTER.

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

8.5.2 ONE SYNC BIT OR TWO?

SINCE TOO MUCH HARDWARE MUST BE TURNED ON TO SENSE THE PRESENTS OF ONE SYNC OR TWO. THE PROGRAM ASSUMES TWO SYNC CHARS. NOTE: THIS ASSUMPTION MAY BE ALTERED AFTER AUTO SIZING BY ALTERING BIT 15 IN APPRIOATE DQSTXX: LOCATION.

8.5.3 'BB' OPTION INSTALLED?

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

8.5.4 'AB' OPTION INSTALLED?

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

8.5.5 'BA' OPTION INSTALLED?

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

8.5.6 JUMPER ON END OF CABLE? ***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 APPRIOATE DQSTXX: AFTER AUTO SIZING

8.5.8 SET FOR ODD OR EVEN PARITY?

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

8.5.9 FINDING THE VECTOR.

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

9. PROGRAM DESCRIPTION
CONTAINED WITHIN LISTING
10. LISTING
FOLLOWING

```
522 .ENABLE AMA
523
524 :CZDQAD0/<377>/DQ11 STATIC LOGIC TEST-PART 1
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 'CZDQAD0/<377>/DQ11 STATIC LOGIC TEST-PART 1'
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 DEFINATIONS 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
616 100000
617
618
619
620
621 000000
622 000001
623 000002
624 000003
625 000004
626 000005
627 000006
628 000007
629
630 000010
631 000011
632 000012
633 000013
634 000014
635 000015
636 000016
637 000017
638
639

ODDBIT=1000
SYNBIT=100000

:DQ11 SECONDARY REGISTER DEFINATIONS

RXBA.P=0 ;RECEIVER BUS ADDRESS PRIMARY.
RXWC.P=1 ;RECEIVER WORD COUNT PRIMARY.
TXBA.P=2 ;TRANSMITTER BUS ADDRESS PRIMARY.
TXWC.P=3 ;TRANSMITTER BUS ADDRESS PRIMARY.
RXBA.S=4 ;RECEIVER BUS ADDRESS SECONDARY.
RXWC.S=5 ;RECEIVER WORD COUNT SECONDARY.
TXBA.S=6 ;TRANSMITTER BUS ADDRESS SECONDARY.
TXWC.S=7 ;TRANSMITTER WORD COUNT SECONDARY.

CHARDT=10 ;CHARACTER DETECT REGISTER.
SYNC.=11 ;SYNC REGISTER.
MISC.=12 ;MISCELLANEOUS REGISTER.
TX.MUX=13 ;TRANSMITTER MUX REGISTER.
SEQ.=14 ;SEQUENCE REGISTER.
RX.BCC=15 ;RECEIVER BCC REGISTER.
TX.BCC=16 ;TRANSMITTER BCC REGISTER.
POLY.=17 ;POLYNOMIAL REGISTER.

TRAPCATCHER FOR UNEXPECTED INTERUPTS

```

640          ;TRAPCATCAER FOR ILLEGAL INTERRUPTS
641          000000          .=0
642          ;STANDARD INTERRUPT VECTORS
643
644          000024          .=24
645 000024 016164          .PFAIL          ;POWER FAIL HANDLER
646 000026 000340          340          ;SERVICE AT LEVEL 7
647 000030 015634          .HLT          ;ERROR HANDLER
648 000032 000340          340          ;SERVICE AT LEVEL 7
649 000034 015602          .TRPSRV         ;GENERAL HANDLER DISPATCH SERVICE
650 000036 000340          340          ;SERVICE AT LEVEL 7
651          000046          .=46
652 000046 014362          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 US 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          000174          .=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$,@#4       ;SET FOR TIME OUT TRAP--NO DEVICE--
707 000256 112761 000012 000005 1$:  MOVVB   #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   MOVVB   #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            ;APPERENTLY NO BB OPTION.
719 000336 052712 020000          BIS      #BBBIT,(R2)   ;SET FOR BB OPTION
720 000342 112761 000017 000005   MOVVB   #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$: TYPE          ;TYPE MESSAGE
762 000604 016525          MERR2          ;I DIDN'T FIND ANY DQ11S. DON'T USE AUTO SIZE.
763 000606 005000          CLR R0         ;
764 000610 000000          HALT         ;HOW CAN I TEST NO DQ11S
765 000612 000776          BR -.2        ;DON'T LET OPR HIT CONT. SW
766 000614 012716 000466          5$: MOV #2$(SP) ;ENTERED BY TIME OUT TRAP
767 000620 000002          RTI         ;GO HOME.
768
769
770          001000          .=1000
771 001000 005377 055103 050504  MTITLE: .ASCIZ <377><12>/CZDQAD0/<377>/DQ11 STATIC LOGIC TEST-PART 1/<377>
772 001006 042101 177460 050504
773 001014 030461 051440 040524
774 001022 044524 020103 047514
775 001030 044507 020103 042524
776 001036 052123 050055 051101
777 001044 020124 177461 000
778
779          001200          .=1200
780          ;INDIRECT POINTERS
781
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 PCINTER 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                ;PROGRAM CONTROL FLAGS
826
827 001310          000          INIFLG: .BYTE 0          ;PROGRAM INITIALIZATION FLAG
828 001311          000          STFLG:  .BYTE 0          ;TEST START FLAG
829 001312          000          ERRFLG: .BYTE 0          ;ERROR OCCURED FLAG
830 001313          000          LOKFLG: .BYTE 0          ;LOCK ON CURRENT TEST FLAG
831                000000          $Y=0
832
833                ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
834                ;POINTERS TO SUBROUTINES CAN BE FOUND
835                ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS
836
837                ;*****
838                ;*****
839 001314          .TRPTAB:
840                SCOPE=TRAP+0          ;CALL TO SCOPE LOOP AND ITERATION HANDLER
841 001314          .SCOPE
842                SCOPE1=TRAP+1         ;CALL TO LOOP ON CURRENT DATA HANDLER
843 001316          .SCOPE1
844                TYPE=TRAP+2          ;CALL TO TELETYPE OUTPUT ROUTINE
845 001320          .TYPE
846                INSTR=TRAP+3         ;CALL TO ASCII STRING INPUT ROUTINE
847 001322          .INSTR
848                INSTER=TRAP+4        ;CALL TO INPUT ERROR HANDLER
849 001324          .INSTER
850                PARAM=TRAP+5         ;CALL TO NUMERICAL DATA INPUT ROUTINE
851 001326          .PARAM
852                SAV05=TRAP+6         ;CALL TO REGISTER SAVE ROUTINE
853 001330          .SAV05
854                RES05=TRAP+7        ;CALL TO REGISTER RESTORE ROUTINE
855 001332          .RES05
856                CONVRT=TRAP+10       ;CALL TO DATA OUTPUT ROUTINE
857 001334          .CONVRT
858                CNVRT=TRAP+11        ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.
859 001336          .CNVRT
860                MSTCLR=TRAP+12       ;CALL TO ISSUE MASTER CLEAR
861 001340          .MSTCLR
862                MEMCLR=TRAP+13      ;CALL TO CLEAR ALL SCRATCH PAD MEMORIES
863 001342          .MEMCLR
864                CKSWR=TRAP+14       ;CALL TO ALLOW SWREG TO BE LOADED FROM TTY
865 001344          .CKSWR
866                CNTLU=TRAP+15      ;CALL TO ALLOW LOADING OF SWREG FROM TTY
867 001346          .CNTLU
868
869                ;*****
870                ;*****
871
872                ;DQ11 VECTOR AND REGISTER INDIRECT POINTERS
873
874 001350          000000          DQRVEC: 0          ;POINTER TO DQ11 RECEIVER INTERRUPT VECTOR
875 001352          000000          DQRLVL: 0         ;POINTER TO DQ11 RECEIVER INTERRUPT SERVICE PS
876 001354          000000          DQTVEC: 0         ;POINTER TO DQ11 TRANSMITTER INTERRUPT VECTOR
877 001356          000000          DQTLVL: 0         ;POINTER TO DQ11 TRANSMITTER INTERRUPT SERVICE PS
878 001360          000000          DGRCSR: 0         ;POINTER TO DQ11 RECEIVER CONTROL REGISTER
879 001362          000000          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
  
```

;DQ11 STATUS TABLE AND ADDRESS ASSIGNMENTS

```

885
886
887
888
889
890      001400      .=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 OF 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 016164 000024 MOV #.PFAIL,@#24 ;SET UP POWER FAIL VECTOR
939 001532 013737 001504 001276 MOV DQNUM,SAVNUM
940 001540 105037 001311 CLRB STFLG ;CLEAR START FLAG
941 001544 005037 001230 CLR PASCNT ;CLEAR PASS COUNT
942 001550 105037 001312 CLRB 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$: TSTB @SWR
974 001746 100402 BMI .+6
975 001750 004737 000220 JSR PC,CSRMAP
976 001754 104402 017012 TYPE ,XHEAD
977 001760 012737 001400 001244 MOV #1400,TEMP1
978 001766 017737 177252 001246 MOV @TEMP1,TEMP2
979 001774 001406 BEQ .+16
980 001776 104410 CONVRT
981 002000 017040 XSTATQ
982 002002 062737 000002 001244 ADD #2,TEMP1
983 002010 000766 BR .-22
984 002012 032777 000001 177160 12$: BIT #SW00,@SWR
985 002020 001424 BEQ 1$
986 002022 104402 TYPE
987 002024 016733 MNEW
988 002026 005000 CLR R0
989 002030 000000 HALT
990 002032 104414 CKSWR
991 002034 027737 177140 001502 CMP @SWR,SAVACT
  
```

```

992 002042 101404      BLOS      11$
993 002044 104402      TYPE
994 002046 016574      MERR3
995 002050 000000      HALT
996 002052 000776      BR
997 002054 017737 177120 001500 11$:  MOV      @SWR,DQACTV
998 002062 013700 001500      MOV      DQACTV,RO
999 002066 000000      HALT
1000 002070 104414      CKSWR
1001 002072 012700 000300      1$:  MOV      #300,RO
1002 002076 012701 000302      MOV      #302,R1
1003 002102 010120      2$:  MOV      R1,(R0)+
1004 002104 005021      CLR      (R1)+
1005 002106 022021      CMP      (R0)+,(R1)+
1006 002110 022700 001000      CMP      #1000,RO
1007 002114 001372      BNE      2$
1008
1009
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
1016 002140 032777 000004 177032      BIT      #BIT2,@SWR   ;CHECK FOR <^G>
1017 002146 001411      BEQ      1$          ;CHECK FOR LOCK ON TEST
1018 002150 104402 016632      TYPE      ,MLOCK
1019 002154 012737 000240 014446      MOV      #NOP,TTST
1020 002162 012737 000240 014450      MOV      #NOP,TTST+2
1021 002170 000406      BR
1022 002172 013737 014544 014446 1$:  MOV      BRW,TTST
1023 002200 013737 014546 014450      MOV      BRX,TTST+2
1024 002206 032777 000002 176764 2$:  BIT      #SW01,@SWR   ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1025 002214 001410      BEQ      3$          ;IF SW01=1, GET STARTING PC
1026 002216 104403      INSTR
1027 002220 016620      MTSTPC
1028 002222 104405      PARAM
1029 002224 002254      TST1
1030 002226 014160      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$:  MOV      #TST1,RETURN ;START AT TEST 1
1036 002244 104402 016522 4$:  TYPE      ,MR        ;TYPE R
1037 002250 000177 176740      JMP      @RETURN     ;START TESTING
1038
1039
1040 002254 012737 000001 001226 ; *****
1041 002262 012737 002644 001214 TST1: MOV      #1,TSTNO
1042 002270 012737 002644 001216      MOV      #TST2,RETURN
1043 002276 105737 001302      MOV      #TST2,NEXT
1044 002302 001010      TSTB     RUNFLG      ;IS THIS MY FIRST TIME HERE?
1045 002304 012737 000001 001304      BNE      1$          ;BR IF FLAG IS SET
1046 002312 012737 000020 001306      MOV      #BIT0,RUN   ;SET RUN POINTER.
1047 002320 105137 001302      MOV      #15,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$: TYPE ;ALLERT OPERATOR OF FATAL ERROR
1062 002412 016525 MERR2 ;NO DQ11 ACTIVE. WHY AM I HERE???
1063 002414 000000 HALT ;YOU MUST RELOAD DQ11 DIAGNOSTIC!!
1064 002416 000776 BR -2 ;STICK HERE ON CONT.
1065 002420 000257 3$: 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,DQRCR
1072 002464 013737 001510 001350 MOV DQSTAT,DQREVC
1073 002472 042737 177007 001350 BIC #177007,DQREVC
1074 002500 013737 001350 001352 MOV DQREVC,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 DQRCR,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
1093 ;ADDRESS SELECTOR TEST
1094 ;ADDRESS RECEIVER CONTROL REGISTER
1095 ;VERIFY THAT RECEIVER CONTROL REGISTER RESPONDS TO ADDRESSING
1096
1097 ; TEST 2
1098 ;*****
1099 002644 012737 000002 001226 TST2: MOV #2,TSTNO
1100 002652 012737 002730 001216 MOV #TST3,NEXT
1101 002660 012737 002706 000004 MOV #1,@#4 ;SET UP TO RETURN FROM
1102 002666 012737 000340 000006 MOV #30,@#6 ;BUS ERROR TRAP
1103 002674 013705 001360 MOV DQRCSR,R5 ;GET ADDRESS OF RECEIVER CONTROL REGISTER
  
```

BASIC DQ11 ADDRESSING TESTS.

```
1104 002700 005777 176454          TST   @DQRCR          ;ADDRESS RECEIVER CONTROL REGISTER
1105 002704 000401                    BR    2$              ;NO TRAP, REGISTER RESPONDED
1106 002706 104000                    1$:  HLT   0          ;RECEIVER CONTROL REGISTER DID NOT
1107                                     ;RESPOND TO ADDRESSING
1108 002710 012706 001200                    2$:  MOV   #STACK,SP  ;RESTORE STACK
1109 002714 012737 000006 000004          MOV   #6,@#4         ;RESTORE TRAPCATCHER
1110 002722 005037 000006                    CLR   @#6
1111 002726 104400                    3$:  SCOPE          ;CHECK FOR ITERATIONS, LOOP
1112
1113                                     ;ADDRESS SELECTOR TEST
1114                                     ;ADDRESS TRANSMITTER CONTROL REGISTER
1115                                     ;VERIFY THAT TRANSMITTER CONTROL REGISTER RESPONDS TO ADDRESSING
1116
1117                                     : TEST 3
1118                                     :*****
1119 002730 012737 000003 001226          TST3: MOV   #3,TSTNO
1120 002736 012737 003014 001216          MOV   #TST4,NEXT
1121 002744 012737 002772 000004          MOV   #1$,@#4       ;SET UP TO RETURN FROM
1122 002752 012737 000340 000006          MOV   #340,@#6      ;BUS ERROR TRAP
1123 002760 013705 001364                    MOV   DQTCR,R5      ;GET ADDRESS OF TRANSMITTER CONTROL REGISTER
1124 002764 005777 176374                    TST   @DQTCR        ;ADDRESS TRANSMITTER CONTROL REGISTER
1125 002770 000401                    BR    2$              ;NO TRAP, REGISTER RESPONDED
1126 002772 104000                    1$:  HLT   0          ;TRANSMITTER CONTROL REGISTER DID NOT
1127                                     ;RESPOND TO ADDRESSING
1128 002774 012706 001200                    2$:  MOV   #STACK,SP  ;RESTORE STACK
1129 003000 012737 000006 000004          MOV   #6,@#4         ;RESTORE TRAPCATCHER
1130 003006 005037 000006                    CLR   @#6
1131 003012 104400                    3$:  SCOPE          ;CHECK FOR ITERATIONS, LOOP
1132
1133                                     ;ADDRESS SELECTOR TEST
1134                                     ;ADDRESS ERROR REGISTER
1135                                     ;VERIFY THAT ERROR REGISTER RESPONDS TO ADDRESSING
1136
1137                                     : TEST 4
1138                                     :*****
1139 003014 012737 000004 001226          TST4: MOV   #4,TSTNO
1140 003022 012737 003100 001216          MOV   #TST5,NEXT
1141 003030 012737 003056 000004          MOV   #1$,@#4       ;SET UP TO RETURN FROM
1142 003036 012737 000340 000006          MOV   #340,@#6      ;BUS ERROR TRAP
1143 003044 013705 001366                    MOV   DQERR,R5      ;GET ADDRESS OF ERROR REGISTER
1144 003050 005777 176312                    TST   @DQERR        ;ADDRESS ERROR REGISTER
1145 003054 000401                    BR    2$              ;NO TRAP, REGISTER RESPONDED
1146 003056 104000                    1$:  HLT   0          ;ERROR REGISTER DID NOT
1147                                     ;RESPOND TO ADDRESSING
1148 003060 012706 001200                    2$:  MOV   #STACK,SP  ;RESTORE STACK
1149 003064 012737 000006 000004          MOV   #6,@#4         ;RESTORE TRAPCATCHER
1150 003072 005037 000006                    CLR   @#6
1151 003076 104400                    3$:  SCOPE          ;CHECK FOR ITERATIONS, LOOP
1152
1153                                     ;ADDRESS SELECTOR TEST
1154                                     ;ADDRESS SECONDARY REGISTER
1155                                     ;VERIFY THAT SECONDARY REGISTER RESPONDS TO ADDRESSING
1156
1157                                     : TEST 5
1158                                     :*****
1159 003100 012737 000005 001226          TST5: MOV   #5,TSTNO
```


BASIC DQ11 ADDRESSING TESTS.

```

1160 003106 012737 003164 001216      MOV    #TST6,NEXT
1161 003114 012737 003142 000004      MOV    #1$,@#4          ;SET UP TO RETURN FROM
1162 003122 012737 000340 000006      MOV    #340,@#6        ;BUS ERROR TRAP
1163 003130 013705 001372              MOV    DQSEC,R5         ;GET ADDRESS OF SECONDARY REGISTER
1164 003134 005777 176232              TST    @DQSEC           ;ADDRESS SECONDARY REGISTER
1165 003140 000401              BR     2$               ;NO TRAP, REGISTER RESPONDED
1166 003142 104000              1$:   HLT    0           ;SECONDARY REGISTER DID NOT
1167                                     ;RESPOND TO ADDRESSING
1168 003144 012706 001200              2$:   MOV    #STACK,SP   ;RESTORE STACK
1169 003150 012737 000006 000004      MOV    #6,@#4          ;RESTORE TRAPCATCHER
1170 003156 005037 000006              CLR    @#6
1171 003162 104400              3$:   SCOPE              ;CHECK FOR ITERATIONS, LOOP
1172
1173                                     ;PRIMARY REGISTER ADDRESSING TEST
1174                                     ;LOAD EACH PRIMARY REGISTER WITH A DIFFERENT
1175                                     ;NUMBER AND VERIFY THAT THE CORRECT REGISTER
1176                                     ;WAS ADDRESSED
1177
1178                                     ; TEST 6
1179                                     ;*****
1180 003164 012737 000006 001226      TST6: MOV    #6,TSTNO
1181 003172 012737 003364 001216      MOV    #TST7,NEXT
1182 003200 012777 000040 176152      MOV    #40,@DQRCSR     ;LOAD RECEIVER CONTROL REGISTER
1183                                     ;WITH BIT5
1184 003206 012777 000100 176150      MOV    #100,@DQTCSR    ;LOAD TRANSMITTER CONTROL
1185                                     ;REGISTER WITH BIT6
1186 003214 012777 000200 176144      MOV    #200,@DQERR     ;LOAD ERROR REGISTER
1187                                     ;WITH BIT7
1188 003222 012777 000400 176142      MOV    #400,@DQSEC     ;LOAD SECONDARY REGISTER
1189                                     ;WITH BIT8
1190 003230 012705 000040              MOV    #40,R5          ;FIRST EXPECTED DATA
1191 003234 017704 176120              MOV    @DQRCSR,R4      ;READ RECEIVER CONTROL REGISTER
1192 003240 013703 001360              MOV    DQRCSR,R3       ;SET UP ADDRESS OF RECEIVER CONTROL REGISTER
1193 003244 020504              CMP    R5,R4           ;WAS RECEIVER CONTROL REGISTER ADDRESSED
1194 003246 001401              BEQ    1$              ;BR IF GOOD
1195 003250 104001              HLT    1               ;REGISTER ADDRESSING ERROR
1196 003252 006305              1$:   ASL    R5          ;NEXT EXPECTED DATA
1197 003254 017704 176104              MOV    @DQTCSR,R4      ;READ TRANSMITTER CONTROL REGISTER
1198 003260 042704 077400              BIC    #77400,R4       ;CLEAR UNWANTED BITS
1199 003264 062703 000002              ADD    #2,R3           ;UPDATE ADDRESS OF EXPECTED REGISTER
1200 003270 020504              CMP    R5,R4           ;WAS TRANSMITTER CONTROL REGISTER ADDRESSED
1201 003272 001401              BEQ    2$              ;BR IF GOOD
1202 003274 104001              HLT    1               ;REGISTER ADDRESSING ERROR
1203 003276 006305              2$:   ASL    R5          ;NEXT EXPECTED DATA
1204 003300 017704 176062              MOV    @DQERR,R4       ;READ ERROR REGISTER
1205 003304 042704 170000              BIC    #170000,R4      ;CLEAR UNWANTED BITS
1206 003310 062703 000002              ADD    #2,R3           ;UPDATE EXPECTED REGISTER ADDRESS
1207 003314 020504              CMP    R5,R4           ;WAS ERROR REGISTER ADDRESSED
1208 003316 001401              BEQ    3$              ;BR IF GOOD
1209 003320 104001              HLT    1               ;REGISTER ADDRESSING ERROR
1210 003322 006305              3$:   ASL    R5          ;NEXT EXPECTED DATA
1211 003324 017704 176042              MOV    @DQSEC,R4       ;READ SECONDARY REGISTER
1212 003330 062703 000002              ADD    #2,R3           ;UPDATE EXPECTED REGISTER ADDRESS
1213 003334 020504              CMP    R5,R4           ;WAS SECONDARY REGISTER ADDRESSED
1214 003336 001401              BEQ    4$              ;BR IF GOOD
1215 003340 104001              HLT    1               ;REGISTER ADDRESSING ERROR

```

1216	003342	005077	176012	4\$:	CLR	@DQRCR	:CLEAR SEL 0
1217	003346	005077	176012		CLR	@DQTCSR	:CLEAR SEL 2
1218	003352	005077	176010		CLR	@DQERR	:CLEAR SEL 4
1219	003356	005077	176010		CLR	@DQSEC	:CLEAR SEL 6
1220	003362	104400		5\$:	SCOPE		:CHECK FOR ITERATIONS, LOOP

BASIC RECEIVER READ/WRITE TESTS.

```
1277
1278
1279
1280
1281
1282
1283
1284 003514 012737 000011 001226
1285 003522 012737 003570 001216
1286 003530 013703 001360
1287
1288 003534 012705 000020
1289
1290 003540 010513
1291 003542 011304
1292
1293 003544 020504
1294 003546 001401
1295 003550 104002
1296 003552 040513
1297
1298 003554 011304
1299 003556 005005
1300
1301 003560 020504
1302 003562 001401
1303 003564 104002
1304 003566 104400
1305
1306
1307
1308
1309
1310
1311
1312 003570 012737 000012 001226
1313 003576 012737 003644 001216
1314 003604 013703 001360
1315
1316 003610 012705 000040
1317
1318 003614 010513
1319 003616 011304
1320
1321 003620 020504
1322 003622 001401
1323 003624 104002
1324 003626 040513
1325
1326 003630 011304
1327 003632 005005
1328
1329 003634 020504
1330 003636 001401
1331 003640 104002
1332 003642 104400

;RECEIVER CONTROL REGISTER READ/WRITE TEST
;SET BIT4, VERIFY BIT4 WAS SET
;CLEAR BIT4, VERIFY BIT4 WAS CLEARED

: TEST 11
:*****
TST11: MOV #11,TSTNO
MOV #TST12,NEXT
MOV DQRCSR,R3 ;LOAD R3 WITH ADDRESS
;OF RECEIVER CONTROL REGISTER
;RECEIVER CONTROL REGISTER WILL
;BE SET TO BIT4
MOV #BIT4,R5 ;LOAD RECEIVER CONTROL
;(R4)=ACTUAL DATA
MOV R5,(R3) ;IN RECEIVER CONTROL REGISTER
;(R4)=ACTUAL DATA
MOV (R3),R4 ;ARE EXPECTED AND RECEIVED VALUES THE SAME ?
CMP R5,R4
BEQ 1$
HLT 2 ;RECEIVER CONTROL REGISTER DATA ERROR
1$: BIC R5,(R3) ;CLEAR BITS SET
;IN RECEIVER CONTROL REGISTER
;READ RECEIVER CONTROL REGISTER
;R5=EXPECTED CONTENTS
;OF RECEIVER CONTROL REGISTER, 0
;WAS RECEIVER CONTROL CLEARED
MOV (R3),R4
CLR R5
CMP R5,R4
BEQ 2$
HLT 2 ;RECEIVER CONTROL REGISTER DATA ERROR
2$: SCOPE ;CHECK FOR ITERATIONS, LOOP

;RECEIVER CONTROL REGISTER READ/WRITE TEST
;SET BIT5, VERIFY BIT5 WAS SET
;CLEAR BIT5, VERIFY BIT5 WAS CLEARED

: TEST 12
:*****
TST12: MOV #12,TSTNO
MOV #TST13,NEXT
MOV DQRCSR,R3 ;LOAD R3 WITH ADDRESS
;OF RECEIVER CONTROL REGISTER
;RECEIVER CONTROL REGISTER WILL
;BE SET TO BIT5
MOV #BIT5,R5 ;LOAD RECEIVER CONTROL
;(R4)=ACTUAL DATA
MOV R5,(R3) ;IN RECEIVER CONTROL REGISTER
;(R4)=ACTUAL DATA
MOV (R3),R4 ;ARE EXPECTED AND RECEIVED VALUES THE SAME ?
CMP R5,R4
BEQ 1$
HLT 2 ;RECEIVER CONTROL REGISTER DATA ERROR
1$: BIC R5,(R3) ;CLEAR BITS SET
;IN RECEIVER CONTROL REGISTER
;READ RECEIVER CONTROL REGISTER
;R5=EXPECTED CONTENTS
;OF RECEIVER CONTROL REGISTER, 0
;WAS RECEIVER CONTROL CLEARED
MOV (R3),R4
CLR R5
CMP R5,R4
BEQ 2$
HLT 2 ;RECEIVER CONTROL REGISTER DATA ERROR
2$: SCOPE ;CHECK FOR ITERATIONS, LOOP
```

```
1333
1334
1335
1336
1337
1338
1339
1340 003644 012737 000013 001226
1341 003652 012737 003720 001216
1342 003660 013703 001360
1343
1344 003664 012705 000100
1345
1346 003670 010513
1347 003672 011304
1348
1349 003674 020504
1350 003676 001401
1351 003700 104002
1352 003702 040513
1353
1354 003704 011304
1355 003706 005005
1356
1357 003710 020504
1358 003712 001401
1359 003714 104002
1360 003716 104400
1361
1362
1363
1364
1365
1366
1367
1368 003720 012737 000014 001226
1369 003726 012737 003774 001216
1370 003734 013703 001360
1371
1372 003740 012705 000200
1373
1374 003744 010513
1375 003746 011304
1376
1377 003750 020504
1378 003752 001401
1379 003754 104002
1380 003756 040513
1381
1382 003760 011304
1383 003762 005005
1384
1385 003764 020504
1386 003766 001401
1387 003770 104002
1388 003772 104400

:RECEIVER CONTROL REGISTER READ/WRITE TEST
:SET BIT6, VERIFY BIT6 WAS SET
:CLEAR BIT6, VERIFY BIT6 WAS CLEARED

: TEST 13
:*****
TST13: MOV #13,TSTNO
MOV #TST14,NEXT
MOV DQRCSR,R3
:LOAD R3 WITH ADDRESS
:OF RECEIVER CONTROL REGISTER
:RECEIVER CONTROL REGISTER WILL
:BE SET TO BIT6
MOV #BIT6,R5
:LOAD RECEIVER CONTROL
MOV R5,(R3)
MOV (R3),R4
:(R4)=ACTUAL DATA
:IN RECEIVER CONTROL REGISTER
:ARE EXPECTED AND RECEIVED VALUES THE SAME ?
CMP R5,R4
BEQ 1$
HLT 2
1$: BIC R5,(R3)
:RECEIVER CONTROL REGISTER DATA ERROR
:CLEAR BITS SET
:IN RECEIVER CONTROL REGISTER
:READ RECEIVER CONTROL REGISTER
MOV (R3),R4
CLR R5
:(R5)=EXPECTED CONTENTS
:OF RECEIVER CONTROL REGISTER, 0
:WAS RECEIVER CONTROL CLEARED
CMP R5,R4
BEQ 2$
HLT 2
2$: SCOPE
:RECEIVER CONTROL REGISTER DATA ERROR
:CHECK FOR ITERATIONS, LOOP

:RECEIVER CONTROL REGISTER READ/WRITE TEST
:SET BIT7, VERIFY BIT7 WAS SET
:CLEAR BIT7, VERIFY BIT7 WAS CLEARED

: TEST 14
:*****
TST14: MOV #14,TSTNO
MOV #CKBBO,NEXT
MOV DQRCSR,R3
:LOAD R3 WITH ADDRESS
:OF RECEIVER CONTROL REGISTER
:RECEIVER CONTROL REGISTER WILL
:BE SET TO BIT7
MOV #BIT7,R5
:LOAD RECEIVER CONTROL
MOV R5,(R3)
MOV (R3),R4
:(R4)=ACTUAL DATA
:IN RECEIVER CONTROL REGISTER
:ARE EXPECTED AND RECEIVED VALUES THE SAME ?
CMP R5,R4
BEQ 1$
HLT 2
1$: BIC R5,(R3)
:RECEIVER CONTROL REGISTER DATA ERROR
:CLEAR BITS SET
:IN RECEIVER CONTROL REGISTER
:READ RECEIVER CONTROL REGISTER
MOV (R3),R4
CLR R5
:(R5)=EXPECTED CONTENTS
:OF RECEIVER CONTROL REGISTER, 0
:WAS RECEIVER CONTROL CLEARED
CMP R5,R4
BEQ 2$
HLT 2
2$: SCOPE
:RECEIVER CONTROL REGISTER DATA ERROR
:CHECK FOR ITERATIONS, LOOP
```

```

1389 003774 032737 020000 001510 CKBB0: BIT #BBBIT,DQSTAT
1390 004002 001530 BEQ CONT.0
1391
1392 ;RECEIVER CONTROL REGISTER READ/WRITE TEST
1393 ;SET BIT8, VERIFY BIT8 WAS SET
1394 ;CLEAR BIT8, VERIFY BIT8 WAS CLEARED
1395
1396 ; TEST 15
1397 :*****
1398 004004 012737 000015 001226 TST15: MOV #15,TSTNO
1399 004012 012737 004060 001216 MOV #TST16,NEXT
1400 004020 013703 001360 MOV DQRCSR,R3 ;LOAD R3 WITH ADDRESS
1401 ;OF RECEIVER CONTROL REGISTER
1402 004024 012705 000400 MOV #BIT8,R5 ;RECEIVER CONTROL REGISTER WILL
1403 ;BE SET TO BIT8
1404 004030 010513 MOV R5,(R3) ;LOAD RECEIVER CONTROL
1405 004032 011304 MOV (R3),R4 ;(R4)=ACTUAL DATA
1406 ;IN RECEIVER CONTROL REGISTER
1407 004034 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED VALUES THE SAME ?
1408 004036 001401 BEQ 1$
1409 004040 104002 HLT 2 ;RECEIVER CONTROL REGISTER DATA ERROR
1410 004042 040513 1$: BIC R5,(R3) ;CLEAR BITS SET
1411 ;IN RECEIVER CONTROL REGISTER
1412 004044 011304 MOV (R3),R4 ;READ RECEIVER CONTROL REGISTER
1413 004046 005005 CLR R5 ;(R5)=EXPECTED CONTENTS
1414 ;OF RECEIVER CONTROL REGISTER, 0
1415 004050 020504 CMP R5,R4 ;WAS RECEIVER CONTROL CLEARED
1416 004052 001401 BEQ 2$
1417 004054 104002 HLT 2 ;RECEIVER CONTROL REGISTER DATA ERROR
1418 004056 104400 2$: SCOPE ;CHECK FOR ITERATIONS, LOOP
1419
1420 ;RECEIVER CONTROL REGISTER READ/WRITE TEST
1421 ;SET BIT9, VERIFY BIT9 WAS SET
1422 ;CLEAR BIT9, VERIFY BIT9 WAS CLEARED
1423
1424 ; TEST 16
1425 :*****
1426 004060 012737 000016 001226 TST16: MOV #16,TSTNO
1427 004066 012737 004134 001216 MOV #TST17,NEXT
1428 004074 013703 001360 MOV DQRCSR,R3 ;LOAD R3 WITH ADDRESS
1429 ;OF RECEIVER CONTROL REGISTER
1430 004100 012705 001000 MOV #BIT9,R5 ;RECEIVER CONTROL REGISTER WILL
1431 ;BE SET TO BIT9
1432 004104 010513 MOV R5,(R3) ;LOAD RECEIVER CONTROL
1433 004106 011304 MOV (R3),R4 ;(R4)=ACTUAL DATA
1434 ;IN RECEIVER CONTROL REGISTER
1435 004110 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED VALUES THE SAME ?
1436 004112 001401 BEQ 1$
1437 004114 104002 HLT 2 ;RECEIVER CONTROL REGISTER DATA ERROR
1438 004116 040513 1$: BIC R5,(R3) ;CLEAR BITS SET
1439 ;IN RECEIVER CONTROL REGISTER
1440 004120 011304 MOV (R3),R4 ;READ RECEIVER CONTROL REGISTER
1441 004122 005005 CLR R5 ;(R5)=EXPECTED CONTENTS
1442 ;OF RECEIVER CONTROL REGISTER, 0
1443 004124 020504 CMP R5,R4 ;WAS RECEIVER CONTROL CLEARED
1444 004126 001401 BEQ 2$

```

BASIC RECEIVER READ/WRITE TESTS.

```
1445 004130 104002          HLT      2          ;RECEIVER CONTROL REGISTER DATA ERROR
1446 004132 104400          2$:     SCOPE          ;CHECK FOR ITERATIONS, LOOP
1447
1448          ;RECEIVER CONTROL REGISTER READ/WRITE TEST
1449          ;SET BIT10, VERIFY BIT10 WAS SET
1450          ;CLEAR BIT10, VERIFY BIT10 WAS CLEARED
1451
1452          ; TEST 17
1453          ;*****
1454 004134 012737 000017 001226 TST17:  MOV     #17,TSTNO
1455 004142 012737 004210 001216      MOV     #TST20,NEXT
1456 004150 013703 001360          MOV     DQRCSR,R3          ;LOAD R3 WITH ADDRESS
1457          ;OF RECEIVER CONTROL REGISTER
1458 004154 012705 002000          MOV     #BIT10,R5        ;RECEIVER CONTROL REGISTER WILL
1459          ;BE SET TO BIT10
1460 004160 010513          MOV     R5,(R3)          ;LOAD RECEIVER CONTROL
1461 004162 011304          MOV     (R3),R4          ;(R4)=ACTUAL DATA
1462          ;IN RECEIVER CONTROL REGISTER
1463 004164 020504          CMP     R5,R4            ;ARE EXPECTED AND RECEIVED VALUES THE SAME ?
1464 004166 001401          BEQ     1$
1465 004170 104002          HLT     2
1466 004172 040513          1$:     BIC     R5,(R3)    ;RECEIVER CONTROL REGISTER DATA ERROR
1467          ;CLEAR BITS SET
1468 004174 011304          MOV     (R3),R4          ;IN RECEIVER CONTROL REGISTER
1469 004176 005005          CLR     R5              ;READ RECEIVER CONTROL REGISTER
1470          ;(R5)=EXPECTED CONTENTS
1471 004200 020504          CMP     R5,R4            ;OF RECEIVER CONTROL REGISTER, 0
1472 004202 001401          BEQ     2$              ;WAS RECEIVER CONTROL CLEARED
1473 004204 104002          HLT     2
1474 004206 104400          2$:     SCOPE          ;RECEIVER CONTROL REGISTER DATA ERROR
1475          ;CHECK FOR ITERATIONS, LOOP
1476          ;RECEIVER CONTROL REGISTER READ/WRITE TEST
1477          ;SET BIT11, VERIFY BIT11 WAS SET
1478          ;CLEAR BIT11, VERIFY BIT11 WAS CLEARED
1479
1480          ; TEST 20
1481          ;*****
1482 004210 012737 000020 001226 TST20:  MOV     #20,TSTNO
1483 004216 012737 004264 001216      MOV     #TST21,NEXT
1484 004224 013703 001360          MOV     DQRCSR,R3          ;LOAD R3 WITH ADDRESS
1485          ;OF RECEIVER CONTROL REGISTER
1486 004230 012705 004000          MOV     #BIT11,R5        ;RECEIVER CONTROL REGISTER WILL
1487          ;BE SET TO BIT11
1488 004234 010513          MOV     R5,(R3)          ;LOAD RECEIVER CONTROL
1489 004236 011304          MOV     (R3),R4          ;(R4)=ACTUAL DATA
1490          ;IN RECEIVER CONTROL REGISTER
1491 004240 020504          CMP     R5,R4            ;ARE EXPECTED AND RECEIVED VALUES THE SAME ?
1492 004242 001401          BEQ     1$
1493 004244 104002          HLT     2
1494 004246 040513          1$:     BIC     R5,(R3)    ;RECEIVER CONTROL REGISTER DATA ERROR
1495          ;CLEAR BITS SET
1496 004250 011304          MOV     (R3),R4          ;IN RECEIVER CONTROL REGISTER
1497 004252 005005          CLR     R5              ;READ RECEIVER CONTROL REGISTER
1498          ;(R5)=EXPECTED CONTENTS
1499 004254 020504          CMP     R5,R4            ;OF RECEIVER CONTROL REGISTER, 0
1500 004256 001401          BEQ     2$              ;WAS RECEIVER CONTROL CLEARED
```

BASIC RECEIVER READ/WRITE TESTS.

```
1501 004260 104002          HLT      2          ;RECEIVER CONTROL REGISTER DATA ERROR
1502 004262 104400          2$: SCOPE          ;CHECK FOR ITERATIONS, LOOP
1503 004264          CONT.0:
1504
1505          ;RECEIVER CONTROL REGISTER READ/WRITE TEST
1506          ;SET BIT12, VERIFY BIT12 WAS SET
1507          ;CLEAR BIT12, VERIFY BIT12 WAS CLEARED
1508
1509          ; TEST 21
1510          ;*****
1511 004264 012737 000021 001226 TST21: MOV     #21,TSTNO
1512 004272 012737 004340 001216      MOV     #CHKBA1,NEXT
1513 004300 013703 001360          MOV     DQRCSR,R3          ;LOAD R3 WITH ADDRESS
1514          ;OF RECEIVER CONTROL REGISTER
1515 004304 012705 010000          MOV     #BIT12,R5        ;RECEIVER CONTROL REGISTER WILL
1516          ;BE SET TO BIT12
1517 004310 010513          MOV     R5,(R3)         ;LOAD RECEIVER CONTROL
1518 004312 011304          MOV     (R3),R4        ;(R4)=ACTUAL DATA
1519          ;IN RECEIVER CONTROL REGISTER
1520 004314 020504          CMP     R5,R4          ;ARE EXPECTED AND RECEIVED VALUES THE SAME ?
1521 004316 001401          BEQ     1$
1522 004320 104002          HLT     2          ;RECEIVER CONTROL REGISTER DATA ERROR
1523 004322 040513          1$: BIC     R5,(R3)    ;CLEAR BITS SET
1524          ;IN RECEIVER CONTROL REGISTER
1525 004324 011304          MOV     (R3),R4        ;READ RECEIVER CONTROL REGISTER
1526 004326 005005          CLR     R5            ;(R5)=EXPECTED CONTENTS
1527          ;OF RECEIVER CONTROL REGISTER, 0
1528 004330 020504          CMP     R5,R4          ;WAS RECEIVER CONTROL CLEARED
1529 004332 001401          BEQ     2$
1530 004334 104002          HLT     2          ;RECEIVER CONTROL REGISTER DATA ERROR
1531 004336 104400          2$: SCOPE          ;CHECK FOR ITERATIONS, LOOP
1532
1533          ;IF DATASET CONTROL OPTION IS INSTALLED,
1534          ;TEST 22 AND TEST 23 WILL BE EXECUTED
1535
1536 004340 032737 010000 001510 CHKBA1: BIT     #BABIT,DQSTAT
1537 004346 001454          BEQ     TST24
1538
1539          ;RECEIVER CONTROL REGISTER READ/WRITE TEST
1540          ;SET BIT13, VERIFY BIT13 WAS SET
1541          ;CLEAR BIT13, VERIFY BIT13 WAS CLEARED
1542
1543          ; TEST 22
1544          ;*****
1545 004350 012737 000022 001226 TST22: MOV     #22,TSTNO
1546 004356 012737 004424 001216      MOV     #TST23,NEXT
1547 004364 013703 001360          MOV     DQRCSR,R3          ;LOAD R3 WITH ADDRESS
1548          ;OF RECEIVER CONTROL REGISTER
1549 004370 012705 020000          MOV     #BIT13,R5        ;RECEIVER CONTROL REGISTER WILL
1550          ;BE SET TO BIT13
1551 004374 010513          MOV     R5,(R3)         ;LOAD RECEIVER CONTROL
1552 004376 011304          MOV     (R3),R4        ;(R4)=ACTUAL DATA
1553          ;IN RECEIVER CONTROL REGISTER
1554 004400 020504          CMP     R5,R4          ;ARE EXPECTED AND RECEIVED VALUES THE SAME ?
1555 004402 001401          BEQ     1$
1556 004404 104002          HLT     2          ;RECEIVER CONTROL REGISTER DATA ERROR
```



```

1557 004406 040513      1$:  BIC      R5,(R3)      ;CLEAR BITS SET
1558                                ;IN RECEIVER CONTROL REGISTER
1559 004410 011304      MOV      (R3),R4      ;READ RECEIVER CONTROL REGISTER
1560 004412 005005      CLR      R5           ;(R5)=EXPECTED CONTENTS
1561                                ;OF RECEIVER CONTROL REGISTER, 0
1562 004414 020504      CMP      R5,R4       ;WAS RECEIVER CONTROL CLEARED
1563 004416 001401      BEQ     2$           ;RECEIVER CONTROL REGISTER DATA ERROR
1564 004420 104002      HLT     2           ;CHECK FOR ITERATIONS, LOOP
1565 004422 104400      2$:  SCOPE
1566                                ;RECEIVER CONTROL REGISTER READ/WRITE TEST
1567                                ;SET BIT14, VERIFY BIT14 WAS SET
1568                                ;CLEAR BIT14, VERIFY BIT14 WAS CLEARED
1569
1570
1571                                ; TEST 23
1572                                ;*****
1573 004424 012737 000023 001226  TST23: MOV      #23,TSTNO
1574 004432 012737 004500 001216  MOV      #TST24,NEXT
1575 004440 013703 001360          MOV      DQRCSR,R3      ;LOAD R3 WITH ADDRESS
1576                                ;OF RECEIVER CONTROL REGISTER
1577 004444 012705 040000          MOV      #BIT14,R5     ;RECEIVER CONTROL REGISTER WILL
1578                                ;BE SET TO BIT14
1579 004450 010513          MOV      R5,(R3)       ;LOAD RECEIVER CONTROL
1580 004452 011304          MOV      (R3),R4       ;(R4)=ACTUAL DATA
1581                                ;IN RECEIVER CONTROL REGISTER
1582 004454 020504          CMP      R5,R4       ;ARE EXPECTED AND RECEIVED VALUES THE SAME ?
1583 004456 001401          BEQ     1$           ;RECEIVER CONTROL REGISTER DATA ERROR
1584 004460 104002          HLT     2           ;CLEAR BITS SET
1585 004462 040513      1$:  BIC      R5,(R3)      ;IN RECEIVER CONTROL REGISTER
1586                                ;READ RECEIVER CONTROL REGISTER
1587 004464 011304          MOV      (R3),R4       ;(R5)=EXPECTED CONTENTS
1588 004466 005005          CLR      R5           ;OF RECEIVER CONTROL REGISTER, 0
1589                                ;WAS RECEIVER CONTROL CLEARED
1590 004470 020504          CMP      R5,R4       ;RECEIVER CONTROL REGISTER DATA ERROR
1591 004472 001401          BEQ     2$           ;CHECK FOR ITERATIONS, LOOP
1592 004474 104002          HLT     2           ;RECEIVER CONTROL REGISTER READ/WRITE TEST
1593 004476 104400      2$:  SCOPE          ;SET BIT15, VERIFY BIT15 WAS SET
1594                                ;CLEAR BIT15, VERIFY BIT15 WAS CLEARED
1595
1596
1597
1598
1599                                ; TEST 24
1600                                ;*****
1601 004500 012737 000024 001226  TST24: MOV      #24,TSTNO
1602 004506 012737 004554 001216  MOV      #TST25,NEXT
1603 004514 013703 001360          MOV      DQRCSR,R3      ;LOAD R3 WITH ADDRESS
1604                                ;OF RECEIVER CONTROL REGISTER
1605 004520 012705 100000          MOV      #BIT15,R5     ;RECEIVER CONTROL REGISTER WILL
1606                                ;BE SET TO BIT15
1607 004524 010513          MOV      R5,(R3)       ;LOAD RECEIVER CONTROL
1608 004526 011304          MOV      (R3),R4       ;(R4)=ACTUAL DATA
1609                                ;IN RECEIVER CONTROL REGISTER
1610 004530 020504          CMP      R5,R4       ;ARE EXPECTED AND RECEIVED VALUES THE SAME ?
1611 004532 001401          BEQ     1$           ;RECEIVER CONTROL REGISTER DATA ERROR
1612 004534 104002          HLT     2
  
```

1613	004536	040513	1\$:	BIC	R5,(R3)	:CLEAR BITS SET
1614						:IN RECEIVER CONTROL REGISTER
1615	004540	011304		MOV	(R3),R4	:READ RECEIVER CONTROL REGISTER
1616	004542	005005		CLR	R5	:(R5)=EXPECTED CONTENTS
1617						:OF RECEIVER CONTROL REGISTER, 0
1618	004544	020504		CMP	R5,R4	:WAS RECEIVER CONTROL CLEARED
1619	004546	001401		BEQ	2\$	
1620	004550	104002		HLT	2	:RECEIVER CONTROL REGISTER DATA ERROR
1621	004552	104400	2\$:	SCOPE		:CHECK FOR ITERATIONS, LOOP

BASIC TRANSMITTER READ/WRITE TESTS.

```
1622
1623                                     :TRANSMITTER CONTROL REGISTER READ/WRITE TEST
1624                                     :SET BIT3, VERIFY BIT3 WAS SET
1625                                     :CLEAR BIT3, VERIFY BIT3 WAS CLEARED
1626
1627                                     : TEST 25
1628                                     :*****
1629 004554 012737 000025 001226 TST25: MOV #25,TSTNO
1630 004562 012737 004640 001216 MOV #CKBA1,NEXT
1631 004570 013703 001360 MOV DQRCR,R3                                     :LOAD R3 WITH ADDRESS
1632                                     :OF TRANSMITTER CONTROL REGISTER
1633 004574 012702 001400 MOV #1400,R2                                     :LOAD R2 WITH 1400
1634                                     :TO CLEAR UNWANTED BITS
1635 004600 012705 000010 MOV #BIT3,R5                                     :TRANSMITTER CONTROL REGISTER WILL
1636                                     :BE SET TO BIT3
1637 004604 010513 MOV R5,(R3)                                     :LOAD TRANSMITTER CONTROL
1638 004606 011304 MOV (R3),R4                                     : (R4)=ACTUAL DATA
1639                                     :IN TRANSMITTER CONTROL REGISTER
1640 004610 040204 BIC R2,R4                                     :CLEAR UNWANTED BITS
1641 004612 020504 CMP R5,R4                                     :ARE EXPECTED AND RECEIVED VALUES THE SAME ?
1642 004614 001401 BEQ 1$
1643 004616 104003 HLT 3                                     :TRANSMITTER CONTROL REGISTER DATA ERROR
1644 004620 040513 1$: BIC R5,(R3)                                     :CLEAR BITS SET
1645                                     :IN TRANSMITTER CONTROL REGISTER
1646 004622 011304 MOV (R3),R4                                     :READ TRANSMITTER CONTROL REGISTER
1647 004624 040204 BIC R2,R4                                     :CLEAR UNWANTED BITS
1648 004626 005005 CLR R5                                     : (R5)=EXPECTED CONTENTS
1649                                     :OF TRANSMITTER CONTROL REGISTER, 0
1650 004630 020504 CMP R5,R4                                     :WAS TRANSMITTER CONTROL CLEARED
1651 004632 001401 BEQ 2$
1652 004634 104003 HLT 3                                     :TRANSMITTER CONTROL REGISTER DATA ERROR
1653 004636 104400 2$: SCOPE
1654 004640 032737 010000 001510 CKBA1: BIT #BABIT,DQSTAT
1655 004646 001432 BEQ CONT.1                                     :CHECK FOR ITERATIONS, LOOP
1656
1657                                     :TRANSMITTER CONTROL REGISTER READ/WRITE TEST
1658                                     :SET BIT4, VERIFY BIT4 WAS SET
1659                                     :CLEAR BIT4, VERIFY BIT4 WAS CLEARED
1660
1661                                     : TEST 26
1662                                     :*****
1663 004650 012737 000026 001226 TST26: MOV #26,TSTNO
1664 004656 012737 004734 001216 MOV #TST27,NEXT
1665 004664 013703 001364 MOV DQTCSR,R3                                     :LOAD R3 WITH ADDRESS
1666                                     :OF TRANSMITTER CONTROL REGISTER
1667 004670 012702 001400 MOV #1400,R2                                     :LOAD R2 WITH 1400
1668                                     :TO CLEAR UNWANTED BITS
1669 004674 012705 000020 MOV #BIT4,R5                                     :TRANSMITTER CONTROL REGISTER WILL
1670                                     :BE SET TO BIT4
1671 004700 010513 MOV R5,(R3)                                     :LOAD TRANSMITTER CONTROL
1672 004702 011304 MOV (R3),R4                                     : (R4)=ACTUAL DATA
1673                                     :IN TRANSMITTER CONTROL REGISTER
1674 004704 040204 BIC R2,R4                                     :CLEAR UNWANTED BITS
1675 004706 020504 CMP R5,R4                                     :ARE EXPECTED AND RECEIVED VALUES THE SAME ?
1676 004710 001401 BEQ 1$
1677 004712 104003 HLT 3                                     :TRANSMITTER CONTROL REGISTER DATA ERROR
```

```

1678 004714 040513 1$: BIC R5,(R3) ;CLEAR BITS SET
1679 ;IN TRANSMITTER CONTROL REGISTER
1680 004716 011304 MOV (R3),R4 ;READ TRANSMITTER CONTROL REGISTER
1681 004720 040204 BIC R2,R4 ;CLEAR UNWANTED BITS
1682 004722 005005 CLR R5 ;(R5)=EXPECTED CONTENTS
1683 ;OF TRANSMITTER CONTROL REGISTER, 0
1684 004724 020504 CMP R5,R4 ;WAS TRANSMITTER CONTROL CLEARED
1685 004726 001401 BEQ 2$
1686 004730 104003 HLT 3 ;TRANSMITTER CONTROL REGISTER DATA ERROR
1687 004732 104400 2$: SCOPE ;CHECK FOR ITERATIONS, LOOP
1688 004734
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
  
```

```

:TRANSMITTER CONTROL REGISTER READ/WRITE TEST
:SET BIT5, VERIFY BIT5 WAS SET
:CLEAR BIT5, VERIFY BIT5 WAS CLEARED
  
```

: TEST 27

:*****

```

1696 004734 012737 000027 001226 TST27: MOV #27,TSTNO
1697 004742 012737 005020 001216 MOV #TST30,NEXT
1698 004750 013703 001364 MOV DQTCSR,R3 ;LOAD R3 WITH ADDRESS
1699 ;OF TRANSMITTER CONTROL REGISTER
1700 004754 012702 001400 MOV #1400,R2 ;LOAD R2 WITH 1400
1701 ;TO CLEAR UNWANTED BITS
1702 004760 012705 000040 MOV #BIT5,R5 ;TRANSMITTER CONTROL REGISTER WILL
1703 ;BE SET TO BIT5
1704 004764 010513 MOV R5,(R3) ;LOAD TRANSMITTER CONTROL
1705 004766 011304 MOV (R3),R4 ;(R4)=ACTUAL DATA
1706 ;IN TRANSMITTER CONTROL REGISTER
1707 004770 040204 BIC R2,R4 ;CLEAR UNWANTED BITS
1708 004772 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED VALUES THE SAME ?
1709 004774 001401 BEQ 1$
1710 004776 104003 HLT 3 ;TRANSMITTER CONTROL REGISTER DATA ERROR
1711 005000 040513 1$: BIC R5,(R3) ;CLEAR BITS SET
1712 ;IN TRANSMITTER CONTROL REGISTER
1713 005002 011304 MOV (R3),R4 ;READ TRANSMITTER CONTROL REGISTER
1714 005004 040204 BIC R2,R4 ;CLEAR UNWANTED BITS
1715 005006 005005 CLR R5 ;(R5)=EXPECTED CONTENTS
1716 ;OF TRANSMITTER CONTROL REGISTER, 0
1717 005010 020504 CMP R5,R4 ;WAS TRANSMITTER CONTROL CLEARED
1718 005012 001401 BEQ 2$
1719 005014 104003 HLT 3 ;TRANSMITTER CONTROL REGISTER DATA ERROR
1720 005016 104400 2$: SCOPE ;CHECK FOR ITERATIONS, LOOP
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
  
```

```

:TRANSMITTER CONTROL REGISTER READ/WRITE TEST
:SET BIT6, VERIFY BIT6 WAS SET
:CLEAR BIT6, VERIFY BIT6 WAS CLEARED
  
```

: TEST 30

:*****

```

1728 005020 012737 000030 001226 TST30: MOV #30,TSTNO
1729 005026 012737 005104 001216 MOV #TST31,NEXT
1730 005034 013703 001364 MOV DQTCSR,R3 ;LOAD R3 WITH ADDRESS
1731 ;OF TRANSMITTER CONTROL REGISTER
1732 005040 012702 001400 MOV #1400,R2 ;LOAD R2 WITH 1400
1733 ;TO CLEAR UNWANTED BITS
  
```

```

1734 005044 012705 000100      MOV      #BIT6,R5      ;TRANSMITTER CONTROL REGISTER WILL
1735                                ;BE SET TO BIT6
1736 005050 010513      MOV      R5,(R3)      ;LOAD TRANSMITTER CONTROL
1737 005052 011304      MOV      (R3),R4      ;(R4)=ACTUAL DATA
1738                                ;IN TRANSMITTER CONTROL REGISTER
1739 005054 040204      BIC      R2,R4        ;CLEAR UNWANTED BITS
1740 005056 020504      CMP      R5,R4        ;ARE EXPECTED AND RECEIVED VALUES THE SAME ?
1741 005060 001401      BEQ      1$
1742 005062 104003      HLT      3
1743 005064 040513      1$: BIC      R5,(R3)    ;TRANSMITTER CONTROL REGISTER DATA ERROR
1744                                ;CLEAR BITS SET
1745                                ;IN TRANSMITTER CONTROL REGISTER
1746 005070 040204      MOV      (R3),R4      ;READ TRANSMITTER CONTROL REGISTER
1747 005072 005005      BIC      R2,R4        ;CLEAR UNWANTED BITS
1748                                ;(R5)=EXPECTED CONTENTS
1749 005074 020504      CMP      R5,R4        ;OF TRANSMITTER CONTROL REGISTER, 0
1750 005076 001401      BEQ      2$
1751 005100 104003      HLT      3
1752 005102 104400      2$: SCOPE            ;TRANSMITTER CONTROL REGISTER DATA ERROR
1753                                ;CHECK FOR ITERATIONS, LOOP
1754                                ;TRANSMITTER CONTROL REGISTER READ/WRITE TEST
1755                                ;SET BIT7, VERIFY BIT7 WAS SET
1756                                ;CLEAR BIT7, VERIFY BIT7 WAS CLEARED
1757
1758                                ; TEST 31
1759                                ;*****
1760 005104 012737 000031 001226  TST31: MOV      #31,TSTNO
1761 005112 012737 005170 001216      MOV      #CHKBA2,NEXT
1762 005120 013703 001364      MOV      DQTCRS,R3    ;LOAD R3 WITH ADDRESS
1763                                ;OF TRANSMITTER CONTROL REGISTER
1764 005124 012702 001400      MOV      #1400,R2     ;LOAD R2 WITH 1400
1765                                ;TO CLEAR UNWANTED BITS
1766 005130 012705 000200      MOV      #BIT7,R5     ;TRANSMITTER CONTROL REGISTER WILL
1767                                ;BE SET TO BIT7
1768 005134 010513      MOV      R5,(R3)     ;LOAD TRANSMITTER CONTROL
1769 005136 011304      MOV      (R3),R4     ;(R4)=ACTUAL DATA
1770                                ;IN TRANSMITTER CONTROL REGISTER
1771 005140 040204      BIC      R2,R4        ;CLEAR UNWANTED BITS
1772 005142 020504      CMP      R5,R4        ;ARE EXPECTED AND RECEIVED VALUES THE SAME ?
1773 005144 001401      BEQ      1$
1774 005146 104003      HLT      3
1775 005150 040513      1$: BIC      R5,(R3)    ;TRANSMITTER CONTROL REGISTER DATA ERROR
1776                                ;CLEAR BITS SET
1777                                ;IN TRANSMITTER CONTROL REGISTER
1778 005152 011304      MOV      (R3),R4      ;READ TRANSMITTER CONTROL REGISTER
1779 005154 040204      BIC      R2,R4        ;CLEAR UNWANTED BITS
1780 005156 005005      CLR      R5          ;(R5)=EXPECTED CONTENTS
1781 005160 020504      CMP      R5,R4        ;OF TRANSMITTER CONTROL REGISTER, 0
1782 005162 001401      BEQ      2$
1783 005164 104003      HLT      3
1784 005166 104400      2$: SCOPE            ;TRANSMITTER CONTROL REGISTER DATA ERROR
1785                                ;CHECK FOR ITERATIONS, LOOP
1786                                ;IF DATASET CONTROL OPTION IS INSTALLED,
1787                                ;TEST 32 AND TEST 33 WILL BE EXECUTED
1788
1789 005170 032737 010000 001510  CHKBA2: BIT      #BABIT,DQSTAT
  
```

BASIC TRANSMITTER READ/WRITE TESTS.

1790 005176 001002
1791 005200 000137 005626

BNE +6
JMP CHKBA3

:TRANSMITTER CONTROL REGISTER READ/WRITE TEST
:SET BIT8 IN TRANSMITER CONTROL REGISTER
:VERIFY THAT BIT8,BIT 10 AND BIT11 ARE SET
:CLEAR BIT8
:VERIFY THAT BIT8,BIT 10 AND BIT11 WERE CLEARED

: TEST 32

1800
1801 005204 012737 000032 001226
1802 005212 012737 005366 001216
1803 005220 013703 001364
1804 005224 012705 000400
1805 005230 010513
1806 005232 112777 000012 174130
1807 005240 012777 000002 174124
1808 005246 005277 174120
1809 005252 005377 174114
1810 005256 004737 005524
1811 005262 011304
1812 005264 032737 040000 001510
1813 005272 001404
1814 005274 052705 006000
1815 005300 052705 100000
1816
1817 005304 020504
1818 005306 001401
1819 005310 104003
1820 005312 042713 000400
1821 005316 112777 000012 174044
1822 005324 012777 000002 174040
1823 005332 005277 174034
1824 005336 005377 174030
1825 005342 004737 005524
1826 005346 011304
1827 005350 042704 100000
1828 005354 005005
1829 005356 020504
1830 005360 001401
1831 005362 104003
1832 005364 104400
1833

TST32: MOV #32,TSTNO
MOV #TST33,NEXT
MOV DQTCSR,R3 ;ADDRESS OF TRANSMITTER CONTROL REGISTER
MOV #BIT8,R5 ;(R5)=BIT8
MOV R5,(R3) ;LOAD TRANSMITTER CONTROL REGISTER
MOVB #12,@DQREG ;TRY TO SEL MISC REGISTER
MOV #2,@DQSEC ;TRY TO SET AUTO/STEP
INC @DQSEC ;CLOCK UP!!
DEC @DQSEC ;CLOCK DN!!
JSR PC,DELAY ;DELAY FOR REAL CABLE.
MOV (R3),R4 ;READ TRANSMITTER CONTROL REGISTER
BIT #JUMBIT,DQSTAT ;IS TEST JUMPER INSTALLED
BEQ +12 ;BR IF NO JUMPER
BIS #BIT10+BIT11,R5 ;EXPECT BIT8,BIT 10 AND BIT11
BIS #BIT15,R5 ;ADJUST EXPECTED RESULTS.
CMP R5,R4 ;FOR DATA SET INTR
BEQ 1\$;ARE EXPECTED AND RECEIVED DATA THE SAME ?
HLT 3 ;TRANSMITTER CONTROL REGISTER DATA ERROR
1\$: BIC #BIT8,(R3) ;CLEAR BIT8
MOVB #12,@DQREG ;TRY AND SELECT THE MISC REG
MOV #2,@DQSEC ;TRY AND SET AUTO/STEP TO STEP
INC @DQSEC ;SET CLOCK UP!
DEC @DQSEC ;CLOCK DOWN!
JSR PC,DELAY ;DELAY.
MOV (R3),R4 ;READ TRANSMITTER CONTROL REGISTER
BIC #BIT15,R4 ;IGNORE BIT 15 FOR NOW.
CLR R5 ;EXPECT 0
CMP R5,R4 ;WAS TRANSMITTER CONTROL REGISTER CLEARED
BEQ 2\$;BR IF GOOD
HLT 3 ;TRANSMITTER CONTROL REGISTER DATA ERROR
2\$: SCOPE ;CHECK FOR ITERATIONS, LOOP

:TRANSMITTER CONTROL REGISTER READ/WRITE TEST
:SET BIT9 IN TRANSMITER CONTROL REGISTER
:VERIFY THAT BIT9,BIT12 AND BIT13 ARE SET
:CLEAR BIT9
:VERIFY THAT BIT9,BIT12 AND BIT13 WERE CLEARED

: TEST 33

1840
1841
1842 005366 012737 000033 001226
1843 005374 012737 005540 001216
1844 005402 013703 001364
1845 005406 012705 001000

TST33: MOV #33,TSTNO
MOV #TST34,NEXT
MOV DQTCSR,R3 ;ADDRESS OF TRANSMITTER CONTROL REGISTER
MOV #BIT9,R5 ;(R5)=BIT9

BASIC TRANSMITTER READ/WRITE TESTS.

```

1846 005412 010513          MOV    R5,(R3)          ;LOAD TRANSMITTER CONTROL REGISTER
1847 005414 004737 005524    JSR    PC,DELAY        ;DELAY FOR REAL CABLE.
1848 005420 011304          MOV    (R3),R4        ;READ TRANSMITTER CONTROL REGISTER
1849 005422 032737 040000 001510  BIT    #JUMBIT,DQSTAT ;IS TEST JUMPER INSTALLED
1850 005430 001404          BEQ    .+12           ;BR IF NO JUMPER
1851 005432 052705 030000    BIS    #BIT12+BIT13,R5 ;EXPECT BIT9,BIT12 AND BIT13
1852 005436 052705 100000    BIS    #BIT15,R5      ;ADJUST EXPECTED RESULTS.
1853                                     ;FOR DATA SET INTR
1854 005442 020504          CMP    R5,R4          ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
1855 005444 001401          BEQ    1$            ;
1856 005446 104003          HLT    3              ;
1857 005450 042713 001000 1$:    BIC    #BIT9,(R3)      ;TRANSMITTER CONTROL REGISTER DATA ERROR
1858 005454 112777 000012 173706  MOVB   #12,@DQREG     ;CLEAR BIT9
1859 005462 012777 000002 173702  MOV    #2,@DQSEC      ;TRY AND SELECT THE MISC REG
1860 005470 005277 173676    INC    @DQSEC         ;TRY AND SET AUTO/STEP TO STEP
1861 005474 005377 173672    DEC    @DQSEC         ;SET CLOCK UP!
1862 005500 004737 005524    JSR    PC,DELAY        ;CLOCK DOWN!
1863 005504 011304          MOV    (R3),R4        ;DELAY.
1864 005506 042704 100000    BIC    #BIT15,R4      ;READ TRANSMITTER CONTROL REGISTER
1865 005512 005005          CLR    R5             ;IGNORE BIT 15 FOR NOW.
1866 005514 020504          CMP    R5,R4          ;EXPECT 0
1867 005516 001401          BEQ    2$            ;WAS TRANSMITTER CONTROL REGISTER CLEARED
1868 005520 104003          HLT    3              ;BR IF GOOD
1869 005522 104400 2$:    SCOPE ;TRANSMITTER CONTROL REGISTER DATA ERROR
1870 005524 010046 DELAY: MOV    R0,-(SP)   ;CHECK FOR ITERATIONS, LOOP
1871 005526 005000          CLR    R0             ;SAVE R0 ON THE STACK
1872 005530 105200          INCB   R0             ;ZERO R0
1873 005532 100376          BPL    .-2            ;DELAY...
1874 005534 012600          MOV    (SP)+,R0      ;DONE YET?
1875 005536 000207          RTS    PC             ;RESTORE R0
1876                                     ;RETURN.
1877
1878                                     ;READ WRITE TEST OF BIT 15 OF TRANSMITTER CSR.
1879                                     ;SET BIT 15 VERIFY SET; CLEAR BIT 15 VERIFY CLEARED.
1880
1881                                     ; TEST 34
1882 005540 012737 000034 001226  TST34:
1883 005546 012737 005626 001216  MOV    #34,TSTNO
1884 005554 013703 001364    MOV    #CHKBA3,NEXT
1885 005560 005005          MOV    DQTCSR,R3     ;GET TX CSR
1886 005562 005013          CLR    R5             ;CLR TX CSR
1887 005564 005013          CLR    (R3)          ;DO IT AGAIN.
1888 005566 011304          MOV    (R3),R4       ;READ TX CSR
1889 005570 001401          BEQ    1$            ;
1890 005572 104003          HLT    3              ;TX CSR NO ZERO.
1891 005574 052705 100000 1$:    BIS    #BIT15,R5     ;SET EXPECTED.
1892 005600 010513          MOV    R5,(R3)       ;SET BIT 15
1893 005602 011304          MOV    (R3),R4       ;READ CSR.
1894 005604 020504          CMP    R5,R4         ;EXPECTED=RECEIVED?
1895 005606 001401          BEQ    2$            ;
1896 005610 104003          HLT    3              ;TRANSMITTER DATA ERROR.
1897 005612 005005 2$:    CLR    R5             ;SET EXPECTED
1898 005614 010513          MOV    R5,(R3)       ;CLEAR BIT 15
1899 005616 011304          MOV    (R3),R4       ;READ CSR
1900 005620 001401          BEQ    .+1           ;
1901 005622 104003          HLT    3              ;TX CSR NOT ZERO
  
```

1902 005624 104400

SCOPE

1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915

;IF DATASET CONTROL OPTION IS NOT INSTALLED,
;TEST 35 WILL BE EXECUTED

1907 005626 032737 010000 001510
1908 005634 001017

CHKBA3: BIT #BABIT,DQSTAT
BNE TST36

;IF DATASET CONTROL OPTION IS NOT INSTALLED,
;THE WHOLE UPPER BYTE OF THE TX CSR SHOULD BE
;EQUAL TO ZERO.

: TEST 35

1916 005636 012737 000J35 001226
1917 005644 012737 005674 001216
1918 005652 013703 001364
1919 005656 005005
1920 005660 012713 177400
1921 005664 011304
1922 005666 001401
1923 005670 104003
1924 005672 104400

TST35: MOV #35,TSTNO
MOV #TST36,NEXT
MOV DQTCSR,R3
CLR R5
MOV #177400,(R3)
MOV (R3),R4
BEQ +4
HLT 3

;LOAD REG
;SET EXPECTED.
;SET UPPER BYTE TO ALL 1'S
;READ IT BACK.

1\$: SCOPE

;TRANSMITTER CSR NOT ZERO.

;ERROR REGISTER READ/WRITE TEST
;SET BIT0, VERIFY BIT0 AND BIT15 WERE SET
;CLEAR BIT0, VERIFY BIT0 AND BIT15 WERE CLEARED

: TEST 36

1933 005674 012737 000036 001226
1934 005702 012737 005764 001216
1935 005710 013703 001366
1936
1937 005714 012702 060000
1938
1939 005720 012705 000001
1940
1941 005724 010513
1942 005726 011304
1943
1944 005730 052705 100000
1945 005734 040204
1946 005736 020504
1947 005740 001401
1948 005742 104004
1949 005744 040513

TST36: MOV #36,TSTNO
MOV #TST37,NEXT
MOV DQERR,R3
MOV #60000,R2
MOV #BIT0,R5
MOV R5,(R3)
MOV (R3),R4
BIS #BIT15,R5
BIC R2,R4
CMP R5,R4
BEQ 1\$
HLT 4
1\$: BIC R5,(R3)

;LOAD R3 WITH ADDRESS
;OF ERROR REGISTER
;LOAD R2 WITH 60000
;TO CLEAR UNWANTED BITS
;ERROR REGISTER WILL
;BE SET TO BIT0
;LOAD ERROR
;(R4)=ACTUAL DATA
;IN ERROR REGISTER
;EXPECT BIT15 TO BE SET ALSO
;CLEAR UNWANTED BITS
;ARE EXPECTED AND RECEIVED VALUES THE SAME ?

1\$: MOV (R3),R4
BIC R2,R4
CLR R5

;ERROR REGISTER DATA ERROR
;CLEAR BITS SET
;IN ERROR REGISTER
;READ ERROR REGISTER
;CLEAR UNWANTED BITS
;(R5)=EXPECTED CONTENTS
;OF ERROR REGISTER, 0
;WAS ERROR CLEARED

1950
1951 005746 011304
1952 005750 040204
1953 005752 005005
1954
1955 005754 020504
1956 005756 001401
1957 005760 104004

CMP R5,R4
BEQ 2\$
HLT 4

;ERROR REGISTER DATA ERROR

ERROR REGISTER READ/WRITE TESTS.

```
1958 005762 104400 2$: SCOPE ;CHECK FOR ITERATIONS, LOOP
1959
1960 ;ERROR REGISTER READ/WRITE TEST
1961 ;SET BIT1, VERIFY BIT1 AND BIT15 WERE SET
1962 ;CLEAR BIT1, VERIFY BIT1 AND BIT15 WERE CLEARED
1963
1964 ; TEST 37
1965 :*****
1966 005764 012737 000037 001226 TST37: MOV #37,TSTNO
1967 005772 012737 006054 001216 MOV #TST40,NEXT
1968 006000 013703 001366 MOV DQERR,R3 ;LOAD R3 WITH ADDRESS
1969 ;OF ERROR REGISTER
1970 006004 012702 060000 MOV #60000,R2 ;LOAD R2 WITH 60000
1971 ;TO CLEAR UNWANTED BITS
1972 006010 012705 000002 MOV #BIT1,R5 ;ERROR REGISTER WILL
1973 ;BE SET TO BIT1
1974 006014 010513 MOV R5,(R3) ;LOAD ERROR
1975 006016 011304 MOV (R3),R4 ;(R4)=ACTUAL DATA
1976 ;IN ERROR REGISTER
1977 006020 052705 100000 BIS #BIT15,R5 ;EXPECT BIT15 TO BE SET ALSO
1978 006024 040204 BIC R2,R4 ;CLEAR UNWANTED BITS
1979 006026 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED VALUES THE SAME ?
1980 006030 001401 BEQ 1$
1981 006032 104004 HLT 4 ;ERROR REGISTER DATA ERROR
1982 006034 040513 1$: BIC R5,(R3) ;CLEAR BITS SET
1983 ;IN ERROR REGISTER
1984 006036 011304 MOV (R3),R4 ;READ ERROR REGISTER
1985 006040 040204 BIC R2,R4 ;CLEAR UNWANTED BITS
1986 006042 005005 CLR R5 ;(R5)=EXPECTED CONTENTS
1987 ;OF ERROR REGISTER, 0
1988 006044 020504 CMP R5,R4 ;WAS ERROR CLEARED
1989 006046 001401 BEQ 2$
1990 006050 104004 HLT 4 ;ERROR REGISTER DATA ERROR
1991 006052 104400 2$: SCOPE ;CHECK FOR ITERATIONS, LOOP
1992
1993 ;ERROR REGISTER READ/WRITE TEST
1994 ;SET BIT2, VERIFY BIT2 AND BIT15 WERE SET
1995 ;CLEAR BIT2, VERIFY BIT2 AND BIT15 WERE CLEARED
1996
1997 ; TEST 40
1998 :*****
1999 006054 012737 000040 001226 TST40: MOV #40,TSTNO
2000 006062 012737 006144 001216 MOV #TST41,NEXT
2001 006070 013703 001366 MOV DQERR,R3 ;LOAD R3 WITH ADDRESS
2002 ;OF ERROR REGISTER
2003 006074 012702 060000 MOV #60000,R2 ;LOAD R2 WITH 60000
2004 ;TO CLEAR UNWANTED BITS
2005 006100 012705 000004 MOV #BIT2,R5 ;ERROR REGISTER WILL
2006 ;BE SET TO BIT2
2007 006104 010513 MOV R5,(R3) ;LOAD ERROR
2008 006106 011304 MOV (R3),R4 ;(R4)=ACTUAL DATA
2009 ;IN ERROR REGISTER
2010 006110 052705 100000 BJS #BIT15,R5 ;EXPECT BIT15 TO BE SET ALSO
2011 006114 040204 BIC R2,R4 ;CLEAR UNWANTED BITS
2012 006116 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED VALUES THE SAME ?
2013 006120 001401 BEQ 1$
```

```

2014 006122 104004
2015 006124 040513      1$:  HLT      4      :ERROR REGISTER DATA ERROR
                               BIC      R5,(R3)  :CLEAR BITS SET
2016                               :IN ERROR REGISTER
2017 006126 011304      MOV      (R3),R4  :READ ERROR REGISTER
2018 006130 040204      BIC      R2,R4    :CLEAR UNWANTED BITS
2019 006132 005005      CLR      R5      : (R5)=EXPECTED CONTENTS
2020                               :OF ERROR REGISTER, 0
2021 006134 020504      CMP      R5,R4    :WAS ERROR CLEARED
2022 006136 001401      BEQ     2$
2023 006140 104004      HLT      4
2024 006142 104400      2$:  SCOPE      :ERROR REGISTER DATA ERROR
                               :CHECK FOR ITERATIONS, LOOP
2025
2026                               :ERROR REGISTER READ/WRITE TEST
2027                               :SET BIT3, VERIFY BIT3 AND BIT15 WERE SET
2028                               :CLEAR BIT3, VERIFY BIT3 AND BIT15 WERE CLEARED
2029
2030                               : TEST 41
2031                               :*****
2032 006144 012737 000041 001226 TST41: MOV      #41,TSTNO
2033 006152 012737 006234 001216      MOV      #TST42,NEXT
2034 006160 013703 001366      MOV      DQERR,R3      :LOAD R3 WITH ADDRESS
2035                               :OF ERROR REGISTER
2036 006164 012702 060000      MOV      #60000,R2    :LOAD R2 WITH 60000
2037                               :TO CLEAR UNWANTED BITS
2038 006170 012705 000010      MOV      #BIT3,R5     :ERROR REGISTER WILL
2039                               :BE SET TO BIT3
2040 006174 010513      MOV      R5,(R3)     :LOAD ERROR
2041 006176 011304      MOV      (R3),R4     :(R4)=ACTUAL DATA
2042                               :IN ERROR REGISTER
2043 006200 052705 100000      BIS      #BIT15,R5   :EXPECT BIT15 TO BE SET ALSO
2044 006204 040204      BIC      R2,R4       :CLEAR UNWANTED BITS
2045 006206 020504      CMP      R5,R4       :ARE EXPECTED AND RECEIVED VALUES THE SAME ?
2046 006210 001401      BEQ     1$
2047 006212 104004      HLT      4
2048 006214 040513      1$:  BIC      R5,(R3)  :ERROR REGISTER DATA ERROR
2049                               :CLEAR BITS SET
2050                               :IN ERROR REGISTER
2051 006216 011304      MOV      (R3),R4    :READ ERROR REGISTER
2052 006220 040204      BIC      R2,R4     :CLEAR UNWANTED BITS
2053 006222 005005      CLR      R5        : (R5)=EXPECTED CONTENTS
2054                               :OF ERROR REGISTER, 0
2055 006224 020504      CMP      R5,R4     :WAS ERROR CLEARED
2056 006226 001401      BEQ     2$
2057 006230 104004      HLT      4
2058 006232 104400      2$:  SCOPE      :ERROR REGISTER DATA ERROR
                               :CHECK FOR ITERATIONS, LOOP
2059
2060                               :ERROR REGISTER READ/WRITE TEST
2061                               :SET BIT4, VERIFY BIT4 AND BIT15 WERE SET
2062                               :CLEAR BIT4, VERIFY BIT4 AND BIT15 WERE CLEARED
2063
2064                               : TEST 42
2065                               :*****
2066 006234 012737 000042 001226 TST42: MOV      #42,TSTNO
2067 006242 012737 006324 001216      MOV      #TST43,NEXT
2068 006250 013703 001366      MOV      DQERR,R3      :LOAD R3 WITH ADDRESS
2069 006254 012702 060000      MOV      #60000,R2    :LOAD R2 WITH 60000

```



```

2182 006556 011304          MOV      (R3),R4          ;READ ERROR REGISTER
2183 006560 040204          BIC      R2,R4           ;CLEAR UNWANTED BITS
2184 006562 005005          CLR      R5              ;(R5)=EXPECTED CONTENTS
2185                                ;OF ERROR REGISTER, 0
2186 006564 020504          CMP      R5,R4           ;WAS ERROR CLEARED
2187 006566 001401          BEQ     2$
2188 006570 104004          HLT     4
2189 006572 104400          2$:    SCOPE              ;ERROR REGISTER DATA ERROR
2190                                ;CHECK FOR ITERATIONS, LOOP
2191                                ;ERROR REGISTER READ/WRITE TEST
2192                                ;SET BIT8, VERIFY BIT8 WAS SET
2193                                ;CLEAR BIT8, VERIFY BIT8 WAS CLEARED
2194
2195                                ; TEST 46
2196                                ;*****
2197 006574 012737 000046 001226  TST46: MOV      #46,TSTNO
2198 006602 012737 006660 001216  MOV      #TST47,NEXT
2199 006610 013703 001366          MOV      DQERR,R3        ;LOAD R3 WITH ADDRESS
2200                                ;OF ERROR REGISTER
2201 006614 012702 060000          MOV      #60000,R2       ;LOAD R2 WITH 60000
2202                                ;TO CLEAR UNWANTED BITS
2203 006620 012705 000400          MOV      #BIT8,R5        ;ERROR REGISTER WILL
2204                                ;BE SET TO BIT8
2205 006624 010513          MOV      R5,(R3)         ;LOAD ERROR
2206 006626 011304          MOV      (R3),R4         ;(R4)=ACTUAL DATA
2207                                ;IN ERROR REGISTER
2208 006630 040204          BIC      R2,R4           ;CLEAR UNWANTED BITS
2209 006632 020504          CMP      R5,R4           ;ARE EXPECTED AND RECEIVED VALUES THE SAME ?
2210 006634 001401          BEQ     1$
2211 006636 104004          HLT     4
2212 006640 040513          1$:    BIC      R5,(R3)    ;ERROR REGISTER DATA ERROR
2213                                ;CLEAR BITS SET
2214                                ;IN ERROR REGISTER
2214 006642 011304          MOV      (R3),R4         ;READ ERROR REGISTER
2215 006644 040204          BIC      R2,R4           ;CLEAR UNWANTED BITS
2216 006646 005005          CLR      R5              ;(R5)=EXPECTED CONTENTS
2217                                ;OF ERROR REGISTER, 0
2218 006650 020504          CMP      R5,R4           ;WAS ERROR CLEARED
2219 006652 001401          BEQ     2$
2220 006654 104004          HLT     4
2221 006656 104400          2$:    SCOPE              ;ERROR REGISTER DATA ERROR
2222                                ;CHECK FOR ITERATIONS, LOOP
2223                                ;ERROR REGISTER READ/WRITE TEST
2224                                ;SET BIT9, VERIFY BIT9 WAS SET
2225                                ;CLEAR BIT9, VERIFY BIT9 WAS CLEARED
2226
2227                                ; TEST 47
2228                                ;*****
2229 006660 012737 000047 001226  TST47: MOV      #47,TSTNO
2230 006666 012737 006744 001216  MOV      #TST50,NEXT
2231 006674 013703 001366          MOV      DQERR,R3        ;LOAD R3 WITH ADDRESS
2232                                ;OF ERROR REGISTER
2233 006700 012702 060000          MOV      #60000,R2       ;LOAD R2 WITH 60000
2234                                ;TO CLEAR UNWANTED BITS
2235 006704 012705 001000          MOV      #BIT9,R5        ;ERROR REGISTER WILL
2236                                ;BE SET TO BIT9
2237 006710 010513          MOV      R5,(R3)         ;LOAD ERROR
  
```

```
2238 006712 011304          MOV      (R3),R4          ;(R4)=ACTUAL DATA
2239                                ;IN ERROR REGISTER
2240 006714 040204          BIC      R2,R4          ;CLEAR UNWANTED BITS
2241 006716 020504          CMP      R5,R4          ;ARE EXPECTED AND RECEIVED VALUES THE SAME ?
2242 006720 001401          BEQ     1$
2243 006722 104004          HLT     4
2244 006724 040513          1$:    BIC      R5,(R3)    ;ERROR REGISTER DATA ERROR
2245                                ;CLEAR BITS SET
2246                                ;IN ERROR REGISTER
2246 006726 011304          MOV      (R3),R4          ;READ ERROR REGISTER
2247 006730 040204          BIC      R2,R4          ;CLEAR UNWANTED BITS
2248 006732 005005          CLR     R5              ;(R5)=EXPECTED CONTENTS
2249                                ;OF ERROR REGISTER, 0
2250 006734 020504          CMP      R5,R4          ;WAS ERROR CLEARED
2251 006736 001401          BEQ     2$
2252 006740 104004          HLT     4
2253 006742 104400          2$:    SCOPE           ;ERROR REGISTER DATA ERROR
2254                                ;CHECK FOR ITERATIONS, LOOP
```

```
2255                                ;ERROR REGISTER READ/WRITE TEST
2256                                ;SET BIT10, VERIFY BIT10 WAS SET
2257                                ;CLEAR BIT10, VERIFY BIT10 WAS CLEARED
```

```
2258
2259 ; TEST 50
```

```
2260 ;*****
```

```
2261 006744 012737 000050 001226 TST50: MOV      #50,TSTNO
2262 006752 012737 007030 001216      MOV      #TST51,NEXT
2263 006760 013703 001366      MOV      DQERR,R3          ;LOAD R3 WITH ADDRESS
2264                                ;OF ERROR REGISTER
2265 006764 012702 060000      MOV      #60000,R2        ;LOAD R2 WITH 60000
2266                                ;TO CLEAR UNWANTED BITS
2267 006770 012705 002000      MOV      #BIT10,R5        ;ERROR REGISTER WILL
2268                                ;BE SET TO BIT10
2269 006774 010513      MOV      R5,(R3)          ;LOAD ERROR
2270 006776 011304      MOV      (R3),R4          ;(R4)=ACTUAL DATA
2271                                ;IN ERROR REGISTER
2272 007000 040204          BIC      R2,R4          ;CLEAR UNWANTED BITS
2273 007002 020504          CMP      R5,R4          ;ARE EXPECTED AND RECEIVED VALUES THE SAME ?
2274 007004 001401          BEQ     1$
2275 007006 104004          HLT     4
2276 007010 040513          1$:    BIC      R5,(R3)    ;ERROR REGISTER DATA ERROR
2277                                ;CLEAR BITS SET
2278                                ;IN ERROR REGISTER
2278 007012 011304          MOV      (R3),R4          ;READ ERROR REGISTER
2279 007014 040204          BIC      R2,R4          ;CLEAR UNWANTED BITS
2280 007016 005005          CLR     R5              ;(R5)=EXPECTED CONTENTS
2281                                ;OF ERROR REGISTER, 0
2282 007020 020504          CMP      R5,R4          ;WAS ERROR CLEARED
2283 007022 001401          BEQ     2$
2284 007024 104004          HLT     4
2285 007026 104400          2$:    SCOPE           ;ERROR REGISTER DATA ERROR
2286                                ;CHECK FOR ITERATIONS, LOOP
```

```
2287                                ;ERROR REGISTER READ/WRITE TEST
2288                                ;SET BIT11, VERIFY BIT11 WAS SET
2289                                ;CLEAR BIT11, VERIFY BIT11 WAS CLEARED
```

```
2290
2291 ; TEST 51
```

```
2292 ;*****
```

```
2293 007030 012737 000051 001226 TST51: MOV      #51,TSTNO
```

ERROR REGISTER READ/WRITE TESTS.

2294	007036	012737	007114	001216	MOV	#TST52,NEXT	
2295	007044	013703	001366		MOV	DQERR,R3	:LOAD R3 WITH ADDRESS
2296							:OF ERROR REGISTER
2297	007050	012702	060000		MOV	#60000,R2	:LOAD R2 WITH 60000
2298							:TO CLEAR UNWANTED BITS
2299	007054	012705	004000		MOV	#BIT11,R5	:ERROR REGISTER WILL
2300							:BE SET TO BIT11
2301	007060	010513			MOV	R5,(R3)	:LOAD ERROR
2302	007062	011304			MOV	(R3),R4	:(R4)=ACTUAL DATA
2303							:IN ERROR REGISTER
2304	007064	040204			BIC	R2,R4	:CLEAR UNWANTED BITS
2305	007066	020504			CMP	R5,R4	:ARE EXPECTED AND RECEIVED VALUES THE SAME ?
2306	007070	001401			BEQ	1\$	
2307	007072	104004			HLT	4	
2308	007074	040513		1\$:	BIC	R5,(R3)	:ERROR REGISTER DATA ERROR
2309							:CLEAR BITS SET
2310	007076	011304			MOV	(R3),R4	:IN ERROR REGISTER
2311	007100	040204			BIC	R2,R4	:READ ERROR REGISTER
2312	007102	005005			CLR	R5	:CLEAR UNWANTED BITS
2313							:(R5)=EXPECTED CONTENTS
2314	007104	020504			CMP	R5,R4	:OF ERROR REGISTER, 0
2315	007106	001401			BEQ	2\$:WAS ERROR CLEARED
2316	007110	104004			HLT	4	
2317	007112	104400		2\$:	SCOPE		:ERROR REGISTER DATA ERROR
							:CHECK FOR ITERATIONS, LOOP

SECONDARY REGISTER ADDRESSING TESTS

```
2318
2319
2320 ;IF CHARACTER DETECT AND BCC OPTIONS ARE OR ARE NOT
2321 ;INSTALLED, TEST 52 WILL BE EXECUTED
2322
2323 ;SECONDARY REGISTER ADDRESSING TEST
2324
2325 ; TEST 52
2326 :*****
2327 007114 012737 000052 001226 TST52: MOV #52,TSTNO
2328 007122 012737 007240 001216 MOV #OPT1,NEXT
2329 007130 012737 007202 001220 MOV #2$,LOCK
2330 007136 012700 000010 MOV #10,R0
2331 007142 012701 020016 MOV #DATAB,R1 ;GET POINTER TO ADDRESS TEST DATA
2332 007146 005003 CLR R3 ;FIRST SECONDARY REGISTER=0
2333 007150 012105 1$: MOV (R1)+,R5 ;GET DATA TO BE LOADE
2334 007152 110377 172212 MOV R3,@DQREG ;SELECTED SECONDARY REGISTER
2335 007156 010577 172210 MOV R5,@DQSEC ;LOAD SECONDARY REGISTER
2336 007162 005203 INC R3 ;ADDRESS OF NEXT SECONDARY REGISTER
2337 007164 005300 DEC R0 ;CONTINUE IF NOT DONE
2338 007166 001370 BNE 1$
2339 007170 012700 000010 MOV #10,R0
2340 007174 012701 020016 MOV #DATAB,R1 ;GET POINTER TO ADDRESS TEST DATA
2341 007200 005003 CLR R3 ;FIRST SECONDARY REGISTER TO BE CHECKED
2342 007202 110377 172162 2$: MOV R3,@DQREG ;SELECT SECONDARY REGISTER
2343 007206 017704 172160 MOV @DQSEC,R4 ;READ SECONDARY REGISTER
2344 007212 011105 MOV (R1),R5 ;GET TEST DATA
2345 007214 020504 CMP R5,R4 ;CHECK DATA
2346 007216 001401 BEQ 3$
2347 007220 104005 HLT 5 ;SECONDARY REGISTER ADDRESSING ERROR
2348 007222 104401 3$: SCOP1 ;CHECK FOR LOOP ON CURENT ADDRESS
2349 007224 005203 INC R3 ;UPDATE ADDRESS
2350 007226 062701 000002 ADD #2,R1 ;UPDATE REGISTER DATA POINTER
2351 007232 005300 DEC R0 ;CONTINUE IF NOT DONE
2352 007234 001362 BNE 2$
2353 007236 104400 4$: SCOPE ;CHECK FOR ITERATIONS, LOOP
2354
2355 ;IF CHARACTER DETECT OPTION IS INSTALLED,
2356 ;TEST 53 WILL BE EXECUTED
2357
2358 007240 032737 020000 001510 OPT1: BIT #BBBIT,DQSTAT
2359 007246 001005 BNE .+14
2360 007250 012737 007554 001214 MOV #OPT2X,RETURN
2361 007256 000177 171732 JMP @RETURN
2362
2363 ;SECONDARY REGISTER ADDRESSING TEST
2364
2365 ; TEST 53
2366 :*****
2367 007262 012737 000053 001226 TST53: MOV #53,TSTNO
2368 007270 012737 007406 001216 MOV #EOPT1,NEXT
2369 007276 012737 007350 001220 MOV #2$,LOCK
2370 007304 012700 000015 MOV #13,R0
2371 007310 012701 020016 MOV #DATAB,R1 ;GET POINTER TO ADDRESS TEST DATA
2372 007314 005003 CLR R3 ;FIRST SECONDARY REGISTER=0
2373 007316 012105 1$: MOV (R1)+,R5 ;GET DATA TO BE LOADE
```



```
2374 007320 110377 172044      MOVB   R3,@DQREG      ;SELECTED SECONDARY REGISTER
2375 007324 010577 172042      MOV    R5,@DQSEC     ;LOAD SECONDARY REGISTER
2376 007330 005203              INC    R3             ;ADDRESS OF NEXT SECONDARY REGISTER
2377 007332 005300              DEC    R0             ;CONTINUE IF NOT DONE
2378 007334 001370              BNE   1$
2379 007336 012700 000015      MOV    #13.,R0
2380 007342 012701 020016      MOV    #DATAB,R1     ;GET POINTER TO ADDRESS TEST DATA
2381 007346 005003              CLR    R3             ;FIRST SECONDARY REGISTER TO BE CHECKED
2382 007350 110377 172014      2$:  MOVB   R3,@DQREG     ;SELECT SECONDARY REGISTER
2383 007354 017704 172012      MOV    @DQSEC,R4     ;READ SECONDARY REGISTER
2384 007360 011105              MOV    (R1),R5       ;GET TEST DATA
2385 007362 020504              CMP    R5,R4         ;CHECK DATA
2386 007364 001401              BEQ   3$
2387 007366 104005              HLT   5
2388 007370 104401              3$:  SCOP1           ;SECONDARY REGISTER ADDRESSING ERROR
2389 007372 005203              INC    R3             ;CHECK FOR LOOP ON CURENT ADDRESS
2390 007374 062701 000002      ADD    #2,R1         ;UPDATE ADDRESS
2391 007400 005300              DEC    R0             ;UPDATE REGISTER DATA POINTER
2392 007402 001362              BNE   2$             ;CONTINUE IF NOT DONE
2393 007404 104400              4$:  SCOPE           ;CHECK FOR ITERATIONS, LOOP
2394 007406 032737 002000 001510 EUP1: BIT    #ABBIT,DQSTAT
2395 007414 001005              BNE   .+14
2396 007416 012737 007554 001214 MOV    #OPT2X,RETURN
2397 007424 000177 171564      JMP    @RETURN
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
```

;IF CHARACTER DETECT AND BCC OPTIONS ARE INSTALLED,
;EXECUTE TEST 54

;SECONDARY REGISTER ADDRESSING TEST

```
; TEST 54
;*****
2408 007430 012737 000054 001226 TST54: MOV    #54,TSTNO
2409 007436 012737 007554 001216      MOV    #TST55,NEXT
2410 007444 012737 007516 001220      MOV    #2$,LOCK
2411 007452 012700 000020              MOV    #16.,R0
2412 007456 012701 020016      MOV    #DATAB,R1     ;GET POINTER TO ADDRESS TEST DATA
2413 007462 005003              CLR    R3             ;FIRST SECONDARY REGISTER=0
2414 007464 012105              1$:  MOV    (R1)+,R5     ;GET DATA TO BE LOADE
2415 007466 110377 171676      MOVB   R3,@DQREG     ;SELECTED SECONDARY REGISTER
2416 007472 010577 171674      MOV    R5,@DQSEC     ;LOAD SECONDARY REGISTER
2417 007476 005203              INC    R3             ;ADDRESS OF NEXT SECONDARY REGISTER
2418 007500 005300              DEC    R0             ;CONTINUE IF NOT DONE
2419 007502 001370              BNE   1$
2420 007504 012700 000020      MOV    #16.,R0
2421 007510 012701 020016      MOV    #DATAB,R1     ;GET POINTER TO ADDRESS TEST DATA
2422 007514 005003              CLR    R3             ;FIRST SECONDARY REGISTER TO BE CHECKED
2423 007516 110377 171646      2$:  MOVB   R3,@DQREG     ;SELECT SECONDARY REGISTER
2424 007522 017704 171644      MOV    @DQSEC,R4     ;READ SECONDARY REGISTER
2425 007526 011105              MOV    (R1),R5       ;GET TEST DATA
2426 007530 020504              CMP    R5,R4         ;CHECK DATA
2427 007532 001401              BEQ   3$
2428 007534 104005              HLT   5
2429 007536 104401              3$:  SCOP1           ;SECONDARY REGISTER ADDRESSING ERROR
;CHECK FOR LOOP ON CURENT ADDRESS
```

2430 007540 005203
2431 007542 062701 000002
2432 007546 005300
2433 007550 001362
2434 007552 104400
2435

INC R3
ADD #2,R1
DEC R0
BNE 2\$
4\$: SCOPE

:UPDATE ADDRESS
:UPDATE REGISTER DATA POINTER
:CONTINUE IF NOT DONE

:CHECK FOR ITERATIONS, LOOP

```
2436 007554 OPT2X:
2437
2438 :SECONDARY REGISTER READ/WRITE TEST
2439 :SET BIT0 IN SYNC REGISTER
2440 :VERIFY THAT BIT0 WAS SET
2441 :CLEAR BIT0
2442 :VERIFY THAT BIT0 WAS CLEARED
2443
2444 : TEST 55
2445 :*****
2446 007554 012737 000055 001226 TST55: MOV #55,TSTNO
2447 007562 012737 007644 001216 MOV #TST56,NEXT
2448 007570 012703 000011 MOV #11,R3 ;ADDRESS OF SECONDARY REGISTER
2449 ;SYNC
2450 007574 110377 171570 MOV R3,@DQREG ;SELECT SYNC REGISTER
2451 007600 012705 000001 MOV #BIT0,R5 ;(R5)=BIT0
2452 007604 010577 171562 MOV R5,@DQSEC ;SET BIT0 IN
2453 ;SYNC REGISTER
2454 007610 017704 171556 MOV @DQSEC,R4 ;(R4)=ACTUAL DATA IN
2455 ;SYNC REGISTER
2456 007614 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
2457 007616 001401 BEQ 1$ ;BR IF GOOD
2458 007620 104006 HLT 6 ;SECONDARY REGISTER DATA ERROR
2459 007622 040577 171544 1$: BIC R5,@DQSEC ;CLEAR BIT0
2460 007626 017704 171540 MOV @DQSEC,R4 ;READ SYNC REGISTER
2461 007632 005005 CLR R5 ;EXPECT SYNC REGISTER
2462 ;TO CONTAIN 0
2463 007634 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
2464 007636 001401 BEQ 2$ ;BR IF GOOD
2465 007640 104006 HLT 6 ;SECONDARY REGISTER DATA ERROR
2466 007642 104400 2$: SCOPE ;CHECK FOR ITERATIONS, LOOP
2467
2468 :SECONDARY REGISTER READ/WRITE TEST
2469 :SET BIT1 IN SYNC REGISTER
2470 :VERIFY THAT BIT1 WAS SET
2471 :CLEAR BIT1
2472 :VERIFY THAT BIT1 WAS CLEARED
2473
2474 : TEST 56
2475 :*****
2476 007644 012737 000056 001226 TST56: MOV #56,TSTNO
2477 007652 012737 007734 001216 MOV #TST57,NEXT
2478 007660 012703 000011 MOV #11,R3 ;ADDRESS OF SECONDARY REGISTER
2479 ;SYNC
2480 007664 110377 171500 MOV R3,@DQREG ;SELECT SYNC REGISTER
2481 007670 012705 000002 MOV #BIT1,R5 ;(R5)=BIT1
2482 007674 010577 171472 MOV R5,@DQSEC ;SET BIT1 IN
2483 ;SYNC REGISTER
2484 007700 017704 171466 MOV @DQSEC,R4 ;(R4)=ACTUAL DATA IN
2485 ;SYNC REGISTER
2486 007704 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
2487 007706 001401 BEQ 1$ ;BR IF GOOD
2488 007710 104006 HLT 6 ;SECONDARY REGISTER DATA ERROR
2489 007712 040577 171454 1$: BIC R5,@DQSEC ;CLEAR BIT1
2490 007716 017704 171450 MOV @DQSEC,R4 ;READ SYNC REGISTER
2491 007722 005005 CLR R5 ;EXPECT SYNC REGISTER
```

```
2492                                     :TO CONTAIN 0
2493 007724 020504                       CMP      R5,R4      :ARE EXPECTED AND RECEIVED DATA THE SAME ?
2494 007726 001401                       BEQ      2$        :BR IF GOOD
2495 007730 104006                       HLT      6         :SECONDARY REGISTER DATA ERROR
2496 007732 104400                       2$: SCOPE        :CHECK FOR ITERATIONS, LOOP
2497
2498                                     :SECONDARY REGISTER READ/WRITE TEST
2499                                     :SET BIT2 IN SYNC REGISTER
2500                                     :VERIFY THAT BIT2 WAS SET
2501                                     :CLEAR BIT2
2502                                     :VERIFY THAT BIT2 WAS CLEARED
2503
2504                                     : TEST 57
2505                                     :*****
2506 007734 012737 000057 001226 TST57: MOV      #57,TSTNO
2507 007742 012737 010024 001216      MOV      #TST60,NEXT
2508 007750 012703 000011              MOV      #11,R3      :ADDRESS OF SECONDARY REGISTER
2509                                     :SYNC
2510 007754 110377 171410              MOV      R3,@DQREG   :SELECT SYNC REGISTER
2511 007760 012705 000004              MOV      #BIT2,R5    : (R5)=BIT2
2512 007764 010577 171402              MOV      R5,@DQSEC   :SET BIT2 IN
2513                                     :SYNC REGISTER
2514 007770 017704 171376              MOV      @DQSEC,R4   : (R4)=ACTUAL DATA IN
2515                                     :SYNC REGISTER
2516 007774 020504                       CMP      R5,R4      :ARE EXPECTED AND RECEIVED DATA THE SAME ?
2517 007776 001401                       BEQ      1$        :BR IF GOOD
2518 010000 104006                       HLT      6         :SECONDARY REGISTER DATA ERROR
2519 010002 040577 171364              1$: BIC      R5,@DQSEC :CLEAR BIT2
2520 010006 017704 171360              MOV      @DQSEC,R4  :READ SYNC REGISTER
2521 010012 005005                       CLR      R5         :EXPECT SYNC REGISTER
2522                                     :TO CONTAIN 0
2523 010014 020504                       CMP      R5,R4      :ARE EXPECTED AND RECEIVED DATA THE SAME ?
2524 010016 001401                       BEQ      2$        :BR IF GOOD
2525 010020 104006                       HLT      6         :SECONDARY REGISTER DATA ERROR
2526 010022 104400                       2$: SCOPE        :CHECK FOR ITERATIONS, LOOP
2527
2528                                     :SECONDARY REGISTER READ/WRITE TEST
2529                                     :SET BIT3 IN SYNC REGISTER
2530                                     :VERIFY THAT BIT3 WAS SET
2531                                     :CLEAR BIT3
2532                                     :VERIFY THAT BIT3 WAS CLEARED
2533
2534                                     : TEST 60
2535                                     :*****
2536 010024 012737 000060 001226 TST60: MOV      #60,TSTNO
2537 010032 012737 010114 001216      MOV      #TST61,NEXT
2538 010040 012703 000011              MOV      #11,R3      :ADDRESS OF SECONDARY REGISTER
2539                                     :SYNC
2540 010044 110377 171320              MOV      R3,@DQREG   :SELECT SYNC REGISTER
2541 010050 012705 000010              MOV      #BIT3,R5    : (R5)=BIT3
2542 010054 010577 171312              MOV      R5,@DQSEC   :SET BIT3 IN
2543                                     :SYNC REGISTER
2544 010060 017704 171306              MOV      @DQSEC,R4   : (R4)=ACTUAL DATA IN
2545                                     :SYNC REGISTER
2546 010064 020504                       CMP      R5,R4      :ARE EXPECTED AND RECEIVED DATA THE SAME ?
2547 010066 001401                       BEQ      1$        :BR IF GOOD
```

```
2548 010070 104006          HLT      6          :SECONDARY REGISTER DATA ERROR
2549 010072 040577 171274    1$:     BIC      R5,@DQSEC :CLEAR BIT3
2550 010076 017704 171270    MOV      @DQSEC,R4  :READ SYNC REGISTER
2551 010102 005005          CLR      R5         :EXPECT SYNC REGISTER
2552                                :TO CONTAIN 0
2553 010104 020504          CMP      R5,R4      :ARE EXPECTED AND RECEIVED DATA THE SAME ?
2554 010106 001401          BEQ      2$         :BR IF GOOD
2555 010110 104006          HLT      6          :SECONDARY REGISTER DATA ERROR
2556 010112 104400    2$:     SCOPE       :CHECK FOR ITERATIONS, LOOP
2557
2558                                :SECONDARY REGISTER READ/WRITE TEST
2559                                :SET BIT4 IN SYNC REGISTER
2560                                :VERIFY THAT BIT4 WAS SET
2561                                :CLEAR BIT4
2562                                :VERIFY THAT BIT4 WAS CLEARED
2563
2564                                : TEST 61
2565                                :*****
2566 010114 012737 000061 001226  TST61:  MOV      #61,TSTNO
2567 010122 012737 010204 001216  MOV      #TST62,NEXT
2568 010130 012703 000011          MOV      #11,R3     :ADDRESS OF SECONDARY REGISTER
2569                                :SYNC
2570 010134 110377 171230    MOV      R3,@DQREG  :SELECT SYNC REGISTER
2571 010140 012705 000020    MOV      #BIT4,R5   : (R5)=BIT4
2572 010144 010577 171222    MOV      R5,@DQSEC  :SET BIT4 IN
2573                                :SYNC REGISTER
2574 010150 017704 171216    MOV      @DQSEC,R4  : (R4)=ACTUAL DATA IN
2575                                :SYNC REGISTER
2576 010154 020504          CMP      R5,R4      :ARE EXPECTED AND RECEIVED DATA THE SAME ?
2577 010156 001401          BEQ      1$         :BR IF GOOD
2578 010160 104006          HLT      6          :SECONDARY REGISTER DATA ERROR
2579 010162 040577 171204    1$:     BIC      R5,@DQSEC :CLEAR BIT4
2580 010166 017704 171200    MOV      @DQSEC,R4  :READ SYNC REGISTER
2581 010172 005005          CLR      R5         :EXPECT SYNC REGISTER
2582                                :TO CONTAIN 0
2583 010174 020504          CMP      R5,R4      :ARE EXPECTED AND RECEIVED DATA THE SAME ?
2584 010176 001401          BEQ      2$         :BR IF GOOD
2585 010200 104006          HLT      6          :SECONDARY REGISTER DATA ERROR
2586 010202 104400    2$:     SCOPE       :CHECK FOR ITERATIONS, LOOP
2587
2588                                :SECONDARY REGISTER READ/WRITE TEST
2589                                :SET BITS IN SYNC REGISTER
2590                                :VERIFY THAT BITS WAS SET
2591                                :CLEAR BITS
2592                                :VERIFY THAT BITS WAS CLEARED
2593
2594                                : TEST 62
2595                                :*****
2596 010204 012737 000062 001226  TST62:  MOV      #62,TSTNO
2597 010212 012737 010274 001216  MOV      #TST63,NEXT
2598 010220 012703 000011          MOV      #11,R3     :ADDRESS OF SECONDARY REGISTER
2599                                :SYNC
2600 010224 110377 171140    MOV      R3,@DQREG  :SELECT SYNC REGISTER
2601 010230 012705 000040    MOV      #BIT5,R5   : (R5)=BIT5
2602 010234 010577 171132    MOV      R5,@DQSEC  :SET BITS IN
2603                                :SYNC REGISTER
```

SYNC REGISTER READ/WRITE TESTS.

```
2604 010240 017704 171126          MOV    @DQSEC,R4          ;(R4)=ACTUAL DATA IN
2605                                     ;SYNC REGISTER
2606 010244 020504          CMP    R5,R4            ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
2607 010246 001401          BEQ    1$              ;BR IF GOOD
2608 010250 104006          HLT    6              ;SECONDARY REGISTER DATA ERROR
2609 010252 040577 171114 1$:    BIC    R5,@DQSEC        ;CLEAR BIT5
2610 010256 017704 171110          MOV    @DQSEC,R4        ;READ SYNC REGISTER
2611 010262 005005          CLR    R5              ;EXPECT SYNC REGISTER
2612                                     ;TO CONTAIN 0
2613 010264 020504          CMP    R5,R4            ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
2614 010266 001401          BEQ    2$              ;BR IF GOOD
2615 010270 104006          HLT    6              ;SECONDARY REGISTER DATA ERROR
2616 010272 104400 2$:    SCOPE          ;CHECK FOR ITERATIONS, LOOP
2617
2618                                     ;SECONDARY REGISTER READ/WRITE TEST
2619                                     ;SET BIT6 IN SYNC REGISTER
2620                                     ;VERIFY THAT BIT6 WAS SET
2621                                     ;CLEAR BIT6
2622                                     ;VERIFY THAT BIT6 WAS CLEARED
2623
2624                                     ; TEST 63
2625                                     ;*****
2626 010274 012737 000063 001226 TST63: MOV    #63,TSTNO
2627 010302 012737 010364 001216      MOV    #TST64,NEXT
2628 010310 012703 000011          MOV    #11,R3          ;ADDRESS OF SECONDARY REGISTER
2629                                     ;SYNC
2630 010314 110377 171050          MOVB   R3,@DQREG        ;SELECT SYNC REGISTER
2631 010320 012705 000100          MOV    #BIT6,R5        ;(R5)=BIT6
2632 010324 010577 171042          MOV    R5,@DQSEC        ;SET BIT6 IN
2633                                     ;SYNC REGISTER
2634 010330 017704 171036          MOV    @DQSEC,R4        ;(R4)=ACTUAL DATA IN
2635                                     ;SYNC REGISTER
2636 010334 020504          CMP    R5,R4            ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
2637 010336 001401          BEQ    1$              ;BR IF GOOD
2638 010340 104006          HLT    6              ;SECONDARY REGISTER DATA ERROR
2639 010342 040577 171024 1$:    BIC    R5,@DQSEC        ;CLEAR BIT6
2640 010346 017704 171020          MOV    @DQSEC,R4        ;READ SYNC REGISTER
2641 010352 005005          CLR    R5              ;EXPECT SYNC REGISTER
2642                                     ;TO CONTAIN 0
2643 010354 020504          CMP    R5,R4            ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
2644 010356 001401          BEQ    2$              ;BR IF GOOD
2645 010360 104006          HLT    6              ;SECONDARY REGISTER DATA ERROR
2646 010362 104400 2$:    SCOPE          ;CHECK FOR ITERATIONS, LOOP
2647
2648                                     ;SECONDARY REGISTER READ/WRITE TEST
2649                                     ;SET BIT7 IN SYNC REGISTER
2650                                     ;VERIFY THAT BIT7 WAS SET
2651                                     ;CLEAR BIT7
2652                                     ;VERIFY THAT BIT7 WAS CLEARED
2653
2654                                     ; TEST 64
2655                                     ;*****
2656 010364 012737 000064 001226 TST64: MOV    #64,TSTNO
2657 010372 012737 010454 001216      MOV    #TST65,NEXT
2658 010400 012703 000011          MOV    #11,R3          ;ADDRESS OF SECONDARY REGISTER
2659                                     ;SYNC
```

```

2660 010404 110377 170760      MOV      R3,@DQREG      ;SELECT SYNC REGISTER
2661 010410 012705 000200      MOV      #BIT7,R5      ;(R5)=BIT7
2662 010414 010577 170752      MOV      R5,@DQSEC     ;SET BIT7 IN
2663                               ;SYNC REGISTER
2664 010420 017704 170746      MOV      @DQSEC,R4     ;(R4)=ACTUAL DATA IN
2665                               ;SYNC REGISTER
2666 010424 020504             CMP      R5,R4         ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
2667 010426 001401             BEQ      1$           ;BR IF GOOD
2668 010430 104006             HLT      6            ;SECONDARY REGISTER DATA ERROR
2669 010432 040577 170734      BIC      R5,@DQSEC     ;CLEAR BIT7
2670 010436 017704 170730      MOV      @DQSEC,R4     ;READ SYNC REGISTER
2671 010442 005005             CLR      R5           ;EXPECT SYNC REGISTER
2672                               ;TO CONTAIN 0
2673 010444 020504             CMP      R5,R4         ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
2674 010446 001401             BEQ      2$           ;BR IF GOOD
2675 010450 104006             HLT      6            ;SECONDARY REGISTER DATA ERROR
2676 010452 104400             SCOPE                ;CHECK FOR ITERATIONS, LOOP
2677
2678                               ;SECONDARY REGISTER READ/WRITE TEST
2679                               ;SET BIT8 IN SYNC REGISTER
2680                               ;VERIFY THAT BIT8 WAS SET
2681                               ;CLEAR BIT8
2682                               ;VERIFY THAT BIT8 WAS CLEARED
2683
2684                               ; TEST 65
2685                               ;*****
2686 010454 012737 000065 001226  TST65: MOV      #65,TSTNO
2687 010462 012737 010544 001216  MOV      #TST66,NEXT
2688 010470 012703 000011          MOV      #11,R3       ;ADDRESS OF SECONDARY REGISTER
2689                               ;SYNC
2690 010474 110377 170670      MOV      R3,@DQREG     ;SELECT SYNC REGISTER
2691 010500 012705 000400      MOV      #BIT8,R5     ;(R5)=BIT8
2692 010504 010577 170662      MOV      R5,@DQSEC     ;SET BIT8 IN
2693                               ;SYNC REGISTER
2694 010510 017704 170656      MOV      @DQSEC,R4     ;(R4)=ACTUAL DATA IN
2695                               ;SYNC REGISTER
2696 010514 020504             CMP      R5,R4         ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
2697 010516 001401             BEQ      1$           ;BR IF GOOD
2698 010520 104006             HLT      6            ;SECONDARY REGISTER DATA ERROR
2699 010522 040577 170644      BIC      R5,@DQSEC     ;CLEAR BIT8
2700 010526 017704 170640      MOV      @DQSEC,R4     ;READ SYNC REGISTER
2701 010532 005005             CLR      R5           ;EXPECT SYNC REGISTER
2702                               ;TO CONTAIN 0
2703 010534 020504             CMP      R5,R4         ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
2704 010536 001401             BEQ      2$           ;BR IF GOOD
2705 010540 104006             HLT      6            ;SECONDARY REGISTER DATA ERROR
2706 010542 104400             SCOPE                ;CHECK FOR ITERATIONS, LOOP
2707
2708                               ;SECONDARY REGISTER READ/WRITE TEST
2709                               ;SET BIT9 IN SYNC REGISTER
2710                               ;VERIFY THAT BIT9 WAS SET
2711                               ;CLEAR BIT9
2712                               ;VERIFY THAT BIT9 WAS CLEARED
2713
2714                               ; TEST 66
2715                               ;*****

```

```

2716 010544 012737 000066 001226 TST66: MOV #66,TSTNO
2717 010552 012737 010634 001216 MOV #TST67,NEXT
2718 010560 012703 000011 MOV #11,R3 ;ADDRESS OF SECONDARY REGISTER
2719 ;SYNC
2720 010564 110377 170600 MOV R3,@DQREG ;SELECT SYNC REGISTER
2721 010570 012705 001000 MOV #BIT9,R5 ;(R5)=BIT9
2722 010574 010577 170572 MOV R5,@DQSEC ;SET BIT9 IN
2723 ;SYNC REGISTER
2724 010600 017704 170566 MOV @DQSEC,R4 ;(R4)=ACTUAL DATA IN
2725 ;SYNC REGISTER
2726 010604 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
2727 010606 001401 BEQ 1$ ;BR IF GOOD
2728 010610 104006 HLT 6 ;SECONDARY REGISTER DATA ERROR
2729 010612 040577 170554 1$: BIC R5,@DQSEC ;CLEAR BIT9
2730 010616 017704 170550 MOV @DQSEC,R4 ;READ SYNC REGISTER
2731 010622 005005 CLR R5 ;EXPECT SYNC REGISTER
2732 ;TO CONTAIN 0
2733 010624 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
2734 010626 001401 BEQ 2$ ;BR IF GOOD
2735 010630 104006 HLT 6 ;SECONDARY REGISTER DATA ERROR
2736 010632 104400 2$: SCOPE ;CHECK FOR ITERATIONS, LOOP
2737
2738 ;SECONDARY REGISTER READ/WRITE TEST
2739 ;SET BIT10 IN SYNC REGISTER
2740 ;VERIFY THAT BIT10 WAS SET
2741 ;CLEAR BIT10
2742 ;VERIFY THAT BIT10 WAS CLEARED
2743
2744 ; TEST 67
2745 ;*****
2746 010634 012737 000067 001226 TST67: MOV #67,TSTNO
2747 010642 012737 010724 001216 MOV #TST70,NEXT
2748 010650 012703 000011 MOV #11,R3 ;ADDRESS OF SECONDARY REGISTER
2749 ;SYNC
2750 010654 110377 170510 MOV R3,@DQREG ;SELECT SYNC REGISTER
2751 010660 012705 002000 MOV #BIT10,R5 ;(R5)=BIT10
2752 010664 010577 170502 MOV R5,@DQSEC ;SET BIT10 IN
2753 ;SYNC REGISTER
2754 010670 017704 170476 MOV @DQSEC,R4 ;(R4)=ACTUAL DATA IN
2755 ;SYNC REGISTER
2756 010674 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
2757 010676 001401 BEQ 1$ ;BR IF GOOD
2758 010700 104006 HLT 6 ;SECONDARY REGISTER DATA ERROR
2759 010702 040577 170464 1$: BIC R5,@DQSEC ;CLEAR BIT10
2760 010706 017704 170460 MOV @DQSEC,R4 ;READ SYNC REGISTER
2761 010712 005005 CLR R5 ;EXPECT SYNC REGISTER
2762 ;TO CONTAIN 0
2763 010714 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
2764 010716 001401 BEQ 2$ ;BR IF GOOD
2765 010720 104006 HLT 6 ;SECONDARY REGISTER DATA ERROR
2766 010722 104400 2$: SCOPE ;CHECK FOR ITERATIONS, LOOP
2767
2768 ;SECONDARY REGISTER READ/WRITE TEST
2769 ;SET BIT11 IN SYNC REGISTER
2770 ;VERIFY THAT BIT11 WAS SET
2771 ;CLEAR BIT11

```


SYNC REGISTER READ/WRITE TESTS.

```
2828                                     :SECONDARY REGISTER READ/WRITE TEST
2829                                     :SET BIT13 IN SYNC REGISTER
2830                                     :VERIFY THAT BIT13 WAS SET
2831                                     :CLEAR BIT13
2832                                     :VERIFY THAT BIT13 WAS CLEARED
2833
2834                                     : TEST 72
2835                                     :*****
2836 011104 012737 000072 001226 TST72: MOV #72,TSTNO
2837 011112 012737 011174 001216 MOV #TST73,NEXT
2838 011120 012703 000011 MOV #11,R3                                     :ADDRESS OF SECONDARY REGISTER
2839                                     :SYNC
2840 011124 110377 170240 MOV#B R3,@DQREG                                     :SELECT SYNC REGISTER
2841 011130 012705 020000 MOV #BIT13,R5                                     :(R5)=BIT13
2842 011134 010577 170232 MOV R5,@DQSEC                                     :SET BIT13 IN
2843                                     :SYNC REGISTER
2844 011140 017704 170226 MOV @DQSEC,R4                                     :(R4)=ACTUAL DATA IN
2845                                     :SYNC REGISTER
2846 011144 020504 CMP R5,R4                                     :ARE EXPECTED AND RECEIVED DATA THE SAME ?
2847 011146 001401 BEQ 1$                                     :BR IF GOOD
2848 011150 104006 HLT 6                                     :SECONDARY REGISTER DATA ERROR
2849 011152 040577 170214 1$: BIC R5,@DQSEC                                     :CLEAR BIT13
2850 011156 017704 170210 MOV @DQSEC,R4                                     :READ SYNC REGISTER
2851 011162 005005 CLR R5                                     :EXPECT SYNC REGISTER
2852                                     :TO CONTAIN 0
2853 011164 020504 CMP R5,R4                                     :ARE EXPECTED AND RECEIVED DATA THE SAME ?
2854 011166 001401 BEQ 2$                                     :BR IF GOOD
2855 011170 104006 HLT 6                                     :SECONDARY REGISTER DATA ERROR
2856 011172 104400 2$: SCOPE                                     :CHECK FOR ITERATIONS, LOOP
2857
2858                                     :SECONDARY REGISTER READ/WRITE TEST
2859                                     :SET BIT14 IN SYNC REGISTER
2860                                     :VERIFY THAT BIT14 WAS SET
2861                                     :CLEAR BIT14
2862                                     :VERIFY THAT BIT14 WAS CLEARED
2863
2864                                     : TEST 73
2865                                     :*****
2866 011174 012737 000073 001226 TST73: MOV #73,TSTNO
2867 011202 012737 011264 001216 MOV #TST74,NEXT
2868 011210 012703 000011 MOV #11,R3                                     :ADDRESS OF SECONDARY REGISTER
2869                                     :SYNC
2870 011214 110377 170150 MOV#B R3,@DQREG                                     :SELECT SYNC REGISTER
2871 011220 012705 040000 MOV #BIT14,R5                                     :(R5)=BIT14
2872 011224 010577 170142 MOV R5,@DQSEC                                     :SET BIT14 IN
2873                                     :SYNC REGISTER
2874 011230 017704 170136 MOV @DQSEC,R4                                     :(R4)=ACTUAL DATA IN
2875                                     :SYNC REGISTER
2876 011234 020504 CMP R5,R4                                     :ARE EXPECTED AND RECEIVED DATA THE SAME ?
2877 011236 001401 BEQ 1$                                     :BR IF GOOD
2878 011240 104006 HLT 6                                     :SECONDARY REGISTER DATA ERROR
2879 011242 040577 170124 1$: BIC R5,@DQSEC                                     :CLEAR BIT14
2880 011246 017704 170120 MOV @DQSEC,R4                                     :READ SYNC REGISTER
2881 011252 005005 CLR R5                                     :EXPECT SYNC REGISTER
2882                                     :TO CONTAIN 0
2883 011254 020504 CMP R5,R4                                     :ARE EXPECTED AND RECEIVED DATA THE SAME ?
```

SYNC REGISTER READ/WRITE TESTS.

```
2884 011256 001401          BEQ      2$          ;BR IF GOOD
2885 011260 104006          HLT      6          ;SECONDARY REGISTER DATA ERROR
2886 011262 104400          2$: SCOPE          ;CHECK FOR ITERATIONS, LOOP
2887
2888                          ;SECONDARY REGISTER READ/WRITE TEST
2889                          ;SET BIT15 IN SYNC REGISTER
2890                          ;VERIFY THAT BIT15 WAS SET
2891                          ;CLEAR BIT15
2892                          ;VERIFY THAT BIT15 WAS CLEARED
2893
2894                          ; TEST 74
2895                          ;*****
2896 011264 012737 000074 001226 TST74: MOV      #74,TSTNO
2897 011272 012737 011354 001216      MOV      #TST75,NEXT
2898 011300 012703 000011          MOV      #11,R3          ;ADDRESS OF SECONDARY REGISTER
2899                          ;SYNC
2900 011304 110377 170060          MOV      R3,@DQREG      ;SELECT SYNC REGISTER
2901 011310 012705 100000          MOV      #BIT15,R5     ;(R5)=BIT15
2902 011314 010577 170052          MOV      R5,@DQSEC     ;SET BIT15 IN
2903                          ;SYNC REGISTER
2904 011320 017704 170046          MOV      @DQSEC,R4     ;(R4)=ACTUAL DATA IN
2905                          ;SYNC REGISTER
2906 011324 020504          CMP      R5,R4         ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
2907 011326 001401          BEQ      1$          ;BR IF GOOD
2908 011330 104006          HLT      6          ;SECONDARY REGISTER DATA ERROR
2909 011332 040577 170034          1$: BIC      R5,@DQSEC ;CLEAR BIT15
2910 011336 017704 170030          MOV      @DQSEC,R4     ;READ SYNC REGISTER
2911 011342 005005          CLR      R5          ;EXPECT SYNC REGISTER
2912                          ;TO CONTAIN 0
2913 011344 020504          CMP      R5,R4         ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
2914 011346 001401          BEQ      2$          ;BR IF GOOD
2915 011350 104006          HLT      6          ;SECONDARY REGISTER DATA ERROR
2916 011352 104400          2$: SCOPE          ;CHECK FOR ITERATIONS, LOOP
```

MISCELLANEOUS REGISTER READ/WRITE TESTS.

```
2917
2918
2919 :SECONDARY REGISTER READ/WRITE TEST
2920 :SET BIT0 IN MISCELLANEOUS REGISTER
2921 :VERIFY THAT BIT0 WAS SET
2922 :CLEAR BIT0
2923 :VERIFY THAT BIT0 WAS CLEARED
2924
2925 : TEST 75
2926 :*****
2927 TST75: MOV #75,TSTNO
2928 MOV #TST76,NEXT
2929 MOV #12,R3 :ADDRESS OF SECONDARY REGISTER
2930 :MISCELLANEOUS
2931 MOV R3,@DQREG :SELECT MISCELLANEOUS REGISTER
2932 MOV #BIT0,R5 : (R5)=BIT0
2933 MOV R5,@DQSEC :SET BIT0 IN
2934 :MISCELLANEOUS REGISTER
2935 MOV @DQSEC,R4 : (R4)=ACTUAL DATA IN
2936 :MISCELLANEOUS REGISTER
2937 :ARE EXPECTED AND RECEIVED DATA THE SAME ?
2938 :BR IF GOOD
2939 :SECONDARY REGISTER DATA ERROR
2940 1$: BIC R5,@DQSEC :CLEAR BIT0
2941 MOV @DQSEC,R4 :READ MISCELLANEOUS REGISTER
2942 CLR R5 :EXPECT MISCELLANEOUS REGISTER
2943 :TO CONTAIN 0
2944 :ARE EXPECTED AND RECEIVED DATA THE SAME ?
2945 :BR IF GOOD
2946 2$: HLT 6 :SECONDARY REGISTER DATA ERROR
2947 :CHECK FOR ITERATIONS, LOOP
2948 :SECONDARY REGISTER READ/WRITE TEST
2949 :SET BIT1 IN MISCELLANEOUS REGISTER
2950 :VERIFY THAT BIT1 WAS SET
2951 :CLEAR BIT1
2952 :VERIFY THAT BIT1 WAS CLEARED
2953
2954 : TEST 76
2955 :*****
2956 TST76: MOV #76,TSTNO
2957 MOV #TST77,NEXT
2958 MOV #12,R3 :ADDRESS OF SECONDARY REGISTER
2959 :MISCELLANEOUS
2960 MOV R3,@DQREG :SELECT MISCELLANEOUS REGISTER
2961 MOV #BIT1,R5 : (R5)=BIT1
2962 MOV R5,@DQSEC :SET BIT1 IN
2963 :MISCELLANEOUS REGISTER
2964 MOV @DQSEC,R4 : (R4)=ACTUAL DATA IN
2965 :MISCELLANEOUS REGISTER
2966 :ARE EXPECTED AND RECEIVED DATA THE SAME ?
2967 :BR IF GOOD
2968 :SECONDARY REGISTER DATA ERROR
2969 1$: BIC R5,@DQSEC :CLEAR BIT1
2970 MOV @DQSEC,R4 :READ MISCELLANEOUS REGISTER
2971 CLR R5 :EXPECT MISCELLANEOUS REGISTER
2972 :TO CONTAIN 0
```

MISCELLANEOUS REGISTER READ/WRITE TESTS.

```
2973 011524 020504          CMP      R5,R4          ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
2974 011526 001401          BEQ      2$             ;BR IF GOOD
2975 011530 104006          HLT      6              ;SECONDARY REGISTER DATA ERROR
2976 011532 104400          2$:      SCOPE          ;CHECK FOR ITERATIONS, LOOP
2977
2978                          ;SECONDARY REGISTER READ/WRITE TEST
2979                          ;SET BIT3 IN MISCELLANEOUS REGISTER
2980                          ;VERIFY THAT BIT3 WAS SET
2981                          ;CLEAR BIT3
2982                          ;VERIFY THAT BIT3 WAS CLEARED
2983
2984                          : TEST 77
2985                          :*****
2986 011534 012737 000077 001226 TST77:  MOV      #77,TSTNO
2987 011542 012737 011624 001216          MOV      #TST100,NEXT
2988 011550 012703 000012          MOV      #12,R3          ;ADDRESS OF SECONDARY REGISTER
2989                          ;MISCELLANEOUS
2990 011554 110377 167610          MOV      R3,@DQREG      ;SELECT MISCELLANEOUS REGISTER
2991 011560 012705 000010          MOV      #BIT3,R5       ;(R5)=BIT3
2992 011564 010577 167602          MOV      R5,@DQSEC      ;SET BIT3 IN
2993                          ;MISCELLANEOUS REGISTER
2994 011570 017704 167576          MOV      @DQSEC,R4      ;(R4)=ACTUAL DATA IN
2995                          ;MISCELLANEOUS REGISTER
2996 011574 020504          CMP      R5,R4          ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
2997 011576 001401          BEQ      1$             ;BR IF GOOD
2998 011600 104006          HLT      6              ;SECONDARY REGISTER DATA ERROR
2999 011602 040577 167564          1$:      BIC      R5,@DQSEC ;CLEAR BIT3
3000 011606 017704 167560          MOV      @DQSEC,R4      ;READ MISCELLANEOUS REGISTER
3001 011612 005005          CLR      R5             ;EXPECT MISCELLANEOUS REGISTER
3002                          ;TO CONTAIN 0
3003 011614 020504          CMP      R5,R4          ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
3004 011616 001401          BEQ      2$             ;BR IF GOOD
3005 011620 104006          HLT      6              ;SECONDARY REGISTER DATA ERROR
3006 011622 104400          2$:      SCOPE          ;CHECK FOR ITERATIONS, LOOP
3007
3008                          ;SECONDARY REGISTER READ/WRITE TEST
3009                          ;SET BIT6 IN MISCELLANEOUS REGISTER
3010                          ;VERIFY THAT BIT6 WAS SET
3011                          ;CLEAR BIT6
3012                          ;VERIFY THAT BIT6 WAS CLEARED
3013
3014                          : TEST 100
3015                          :*****
3016 011624 012737 000100 001226 TST100: MOV      #100,TSTNO
3017 011632 012737 011714 001216          MOV      #TST101,NEXT
3018 011640 012703 000012          MOV      #12,R3          ;ADDRESS OF SECONDARY REGISTER
3019                          ;MISCELLANEOUS
3020 011644 110377 167520          MOV      R3,@DQREG      ;SELECT MISCELLANEOUS REGISTER
3021 011650 012705 000100          MOV      #BIT6,R5       ;(R5)=BIT6
3022 011654 010577 167512          MOV      R5,@DQSEC      ;SET BIT6 IN
3023                          ;MISCELLANEOUS REGISTER
3024 011660 017704 167506          MOV      @DQSEC,R4      ;(R4)=ACTUAL DATA IN
3025                          ;MISCELLANEOUS REGISTER
3026 011664 020504          CMP      R5,R4          ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
3027 011666 001401          BEQ      1$             ;BR IF GOOD
3028 011670 104006          HLT      6              ;SECONDARY REGISTER DATA ERROR
```

MISCELLANEOUS REGISTER READ/WRITE TESTS.

```
3029 011672 040577 167474      1$:  BIC      R5,@DQSEC      :CLEAR BIT6
3030 011676 017704 167470      MOV      @DQSEC,R4        :READ MISCELLANEOUS REGISTER
3031 011702 005005                CLR      R5                :EXPECT MISCELLANEOUS REGISTER
3032                                :TO CONTAIN 0
3033 011704 020504                CMP      R5,R4            :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3034 011706 001401                BEQ      2$                :BR IF GOOD
3035 011710 104006                HLT      6                :SECONDARY REGISTER DATA ERROR
3036 011712 104400      2$:  SCOPE                :CHECK FOR ITERATIONS, LOOP
3037
3038                                :SECONDARY REGISTER READ/WRITE TEST
3039                                :SET BIT7 IN MISCELLANEOUS REGISTER
3040                                :VERIFY THAT BIT7 WAS SET
3041                                :CLEAR BIT7
3042                                :VERIFY THAT BIT7 WAS CLEARED
3043
3044                                : TEST 101
3045                                :*****
3046 011714 012737 000101 001226  TST101: MOV      #101,TSTNO
3047 011722 012737 012004 001216  MOV      #TST102,NEXT
3048 011730 012703 000012                MOV      #12,R3                :ADDRESS OF SECONDARY REGISTER
3049                                :MISCELLANEOUS
3050 011734 110377 167430                MOV      R3,@DQREG            :SELECT MISCELLANEOUS REGISTER
3051 011740 012705 000200                MOV      #BIT7,R5            : (R5)=BIT7
3052 011744 010577 167422                MOV      R5,@DQSEC            :SET BIT7 IN
3053                                :MISCELLANEOUS REGISTER
3054 011750 017704 167416                MOV      @DQSEC,R4            : (R4)=ACTUAL DATA IN
3055                                :MISCELLANEOUS REGISTER
3056 011754 020504                CMP      R5,R4            :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3057 011756 001401                BEQ      1$                :BR IF GOOD
3058 011760 104006                HLT      6                :SECONDARY REGISTER DATA ERROR
3059 011762 040577 167404      1$:  BIC      R5,@DQSEC      :CLEAR BIT7
3060 011766 017704 167400      MOV      @DQSEC,R4        :READ MISCELLANEOUS REGISTER
3061 011772 005005                CLR      R5                :EXPECT MISCELLANEOUS REGISTER
3062                                :TO CONTAIN 0
3063 011774 020504                CMP      R5,R4            :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3064 011776 001401                BEQ      2$                :BR IF GOOD
3065 012000 104006                HLT      6                :SECONDARY REGISTER DATA ERROR
3066 012002 104400      2$:  SCOPE                :CHECK FOR ITERATIONS, LOOP
3067
3068                                :SECONDARY REGISTER READ/WRITE TEST
3069                                :SET BIT8 IN MISCELLANEOUS REGISTER
3070                                :VERIFY THAT BIT8 WAS SET
3071                                :CLEAR BIT8
3072                                :VERIFY THAT BIT8 WAS CLEARED
3073
3074                                : TEST 102
3075                                :*****
3076 012004 012737 000102 001226  TST102: MOV      #102,TSTNO
3077 012012 012737 012074 001216  MOV      #TST103,NEXT
3078 012020 012703 000012                MOV      #12,R3                :ADDRESS OF SECONDARY REGISTER
3079                                :MISCELLANEOUS
3080 012024 110377 167340                MOV      R3,@DQREG            :SELECT MISCELLANEOUS REGISTER
3081 012030 012705 000400                MOV      #BIT8,R5            : (R5)=BIT8
3082 012034 010577 167332                MOV      R5,@DQSEC            :SET BIT8 IN
3083                                :MISCELLANEOUS REGISTER
3084 012040 017704 167326                MOV      @DQSEC,R4            : (R4)=ACTUAL DATA IN
```

MISCELLANEOUS REGISTER READ/WRITE TESTS.

```
3085 ;MISCELLANEOUS REGISTER
3086 012044 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
3087 012046 001401 BEQ 1$ ;BR IF GOOD
3088 012050 104006 HLT 6 ;SECONDARY REGISTER DATA ERROR
3089 012052 040577 167314 1$: BIC R5,@DQSEC ;CLEAR BIT8
3090 012056 017704 167310 MOV @DQSEC,R4 ;READ MISCELLANEOUS REGISTER
3091 012062 005005 CLR R5 ;EXPECT MISCELLANEOUS REGISTER
3092 ;TO CONTAIN 0
3093 012064 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
3094 012066 001401 BEQ 2$ ;BR IF GOOD
3095 012070 104006 HLT 6 ;SECONDARY REGISTER DATA ERROR
3096 012072 104400 2$: SCOPE ;CHECK FOR ITERATIONS, LOOP
3097
3098 ;SECONDARY REGISTER READ/WRITE TEST
3099 ;SET BIT9 IN MISCELLANEOUS REGISTER
3100 ;VERIFY THAT BIT9 WAS SET
3101 ;CLEAR BIT9
3102 ;VERIFY THAT BIT9 WAS CLEARED
3103
3104 ; TEST 103
3105 ;*****
3106 012074 012737 000103 001226 TST103: MOV #103,TSTNO
3107 012102 012737 012164 001216 MOV #TST104,NEXT
3108 012110 012703 000012 MOV #12,R3 ;ADDRESS OF SECONDARY REGISTER
3109 ;MISCELLANEOUS
3110 012114 110377 167250 MOVB R3,@DQREG ;SELECT MISCELLANEOUS REGISTER
3111 012120 012705 001000 MOV #BIT9,R5 ;(R5)=BIT9
3112 012124 010577 167242 MOV R5,@DQSEC ;SET BIT9 IN
3113 ;MISCELLANEOUS REGISTER
3114 012130 017704 167236 MOV @DQSEC,R4 ;(R4)=ACTUAL DATA IN
3115 ;MISCELLANEOUS REGISTER
3116 012134 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
3117 012136 001401 BEQ 1$ ;BR IF GOOD
3118 012140 104006 HLT 6 ;SECONDARY REGISTER DATA ERROR
3119 012142 040577 167224 1$: BIC R5,@DQSEC ;CLEAR BIT9
3120 012146 017704 167220 MOV @DQSEC,R4 ;READ MISCELLANEOUS REGISTER
3121 012152 005005 CLR R5 ;EXPECT MISCELLANEOUS REGISTER
3122 ;TO CONTAIN 0
3123 012154 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
3124 012156 001401 BEQ 2$ ;BR IF GOOD
3125 012160 104006 HLT 6 ;SECONDARY REGISTER DATA ERROR
3126 012162 104400 2$: SCOPE ;CHECK FOR ITERATIONS, LOOP
3127
3128 ;SECONDARY REGISTER READ/WRITE TEST
3129 ;SET BIT10 IN MISCELLANEOUS REGISTER
3130 ;VERIFY THAT BIT10 WAS SET
3131 ;CLEAR BIT10
3132 ;VERIFY THAT BIT10 WAS CLEARED
3133
3134 ; TEST 104
3135 ;*****
3136 012164 012737 000104 001226 TST104: MOV #104,TSTNO
3137 012172 012737 012254 001216 MOV #TST105,NEXT
3138 012200 012703 000012 MOV #12,R3 ;ADDRESS OF SECONDARY REGISTER
3139 ;MISCELLANEOUS
3140 012204 110377 167160 MOVB R3,@DQREG ;SELECT MISCELLANEOUS REGISTER
```

MISCELLANEOUS REGISTER READ/WRITE TESTS.

```
3141 012210 012705 002000      MOV    #BIT10,R5      ;(R5)=BIT10
3142 012214 010577 167152      MOV    R5,@DQSEC     ;SET BIT10 IN
3143                               ;MISCELLANEOUS REGISTER
3144 012220 017704 167146      MOV    @DQSEC,R4     ;(R4)=ACTUAL DATA IN
3145                               ;MISCELLANEOUS REGISTER
3146 012224 020504             CMP    R5,R4         ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
3147 012226 001401             BEQ    1$            ;BR IF GOOD
3148 012230 104006             HLT    6             ;SECONDARY REGISTER DATA ERROR
3149 012232 040577 167134      1$:  BIC    R5,@DQSEC  ;CLEAR BIT10
3150 012236 017704 167130      MOV    @DQSEC,R4     ;READ MISCELLANEOUS REGISTER
3151 012242 005005             CLR    R5            ;EXPECT MISCELLANEOUS REGISTER
3152                               ;TO CONTAIN 0
3153 012244 020504             CMP    R5,R4         ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
3154 012246 001401             BEQ    2$            ;BR IF GOOD
3155 012250 104006             HLT    6             ;SECONDARY REGISTER DATA ERROR
3156 012252 104400      2$:  SCOPE          ;CHECK FOR ITERATIONS, LOOP
3157
3158                               ;SECONDARY REGISTER READ/WRITE TEST
3159                               ;SET BIT11 IN MISCELLANEOUS REGISTER
3160                               ;VERIFY THAT BIT11 WAS SET
3161                               ;CLEAR BIT11
3162                               ;VERIFY THAT BIT11 WAS CLEARED
3163
3164                               ; TEST 105
3165                               ;*****
3166 012254 012737 000105 001226  TST105: MOV    #105,TSTNO
3167 012262 012737 012344 001216  MOV    #TST106,NEXT
3168 012270 012703 000012      MOV    #12,R3        ;ADDRESS OF SECONDARY REGISTER
3169                               ;MISCELLANEOUS
3170 012274 110377 167070      MOV    R3,@DQREG     ;SELECT MISCELLANEOUS REGISTER
3171 012300 012705 004000      MOV    #BIT11,R5     ;(R5)=BIT11
3172 012304 010577 167062      MOV    R5,@DQSEC     ;SET BIT11 IN
3173                               ;MISCELLANEOUS REGISTER
3174 012310 017704 167056      MOV    @DQSEC,R4     ;(R4)=ACTUAL DATA IN
3175                               ;MISCELLANEOUS REGISTER
3176 012314 020504             CMP    R5,R4         ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
3177 012316 001401             BEQ    1$            ;BR IF GOOD
3178 012320 104006             HLT    6             ;SECONDARY REGISTER DATA ERROR
3179 012322 040577 167044      1$:  BIC    R5,@DQSEC  ;CLEAR BIT11
3180 012326 017704 167040      MOV    @DQSEC,R4     ;READ MISCELLANEOUS REGISTER
3181 012332 005005             CLR    R5            ;EXPECT MISCELLANEOUS REGISTER
3182                               ;TO CONTAIN 0
3183 012334 020504             CMP    R5,R4         ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
3184 012336 001401             BEQ    2$            ;BR IF GOOD
3185 012340 104006             HLT    6             ;SECONDARY REGISTER DATA ERROR
3186 012342 104400      2$:  SCOPE          ;CHECK FOR ITERATIONS, LOOP
3187
3188                               ;SECONDARY REGISTER READ/WRITE TEST
3189                               ;SET BIT15 IN MISCELLANEOUS REGISTER
3190                               ;VERIFY THAT BIT15 WAS SET
3191                               ;CLEAR BIT15
3192                               ;VERIFY THAT BIT15 WAS CLEARED
3193
3194                               ; TEST 106
3195                               ;*****
3196 012344 012737 000106 001226  TST106: MOV    #106,TSTNO
```


3197	012352	012737	012434	001216	MOV	#CHKAB1,NEXT	
3198	012360	012703	000012		MOV	#12,R3	:ADDRESS OF SECONDARY REGISTER
3199							:MISCELLANEOUS
3200	012364	110377	167000		MOVB	R3,@DQREG	:SELECT MISCELLANEOUS REGISTER
3201	012370	012705	100000		MOV	#BIT15,R5	: (R5)=BIT15
3202	012374	010577	166772		MOV	R5,@DQSEC	:SET BIT15 IN
3203							:MISCELLANEOUS REGISTER
3204	012400	017704	166766		MOV	@DQSEC,R4	: (R4)=ACTUAL DATA IN
3205							:MISCELLANEOUS REGISTER
3206	012404	020504			CMP	R5,R4	:ARE EXPECTED AND RECEIVED DATA THE SAME ?
3207	012406	001401			BEQ	1\$:BR IF GOOD
3208	012410	104006			HLT	6	:SECONDARY REGISTER DATA ERROR
3209	012412	040577	166754	1\$:	BIC	R5,@DQSEC	:CLEAR BIT15
3210	012416	017704	166750		MOV	@DQSEC,R4	:READ MISCELLANEOUS REGISTER
3211	012422	005005			CLR	R5	:EXPECT MISCELLANEOUS REGISTER
3212							:TO CONTAIN 0
3213	012424	020504			CMP	R5,R4	:ARE EXPECTED AND RECEIVED DATA THE SAME ?
3214	012426	001401			BEQ	2\$:BR IF GOOD
3215	012430	104006			HLT	6	:SECONDARY REGISTER DATA ERROR
3216	012432	104400		2\$:	SCOPE		:CHECK FOR ITERATIONS, LOOP

POLYNOMIAL REGISTER READ/WRITE TESTS.

```
3217 ;IF BCC OPTION IS INSTALLED
3218 ;TESTS 107 THRU 126 WILL BE EXECUTED
3219
3220 012434 032737 002000 001510 CHKAB1: BIT #ABBIT,DQSTAT
3221 012442 001002 BNE TST107
3222 012444 000137 014250 JMP .EOP
3223
3224 ;SECONDARY REGISTER READ/WRITE TEST
3225 ;SET BIT0 IN BCC POLYNOMIAL REGISTER
3226 ;VERIFY THAT BIT0 WAS SET
3227 ;CLEAR BIT0
3228 ;VERIFY THAT BIT0 WAS CLEARED
3229
3230 ; TEST 107
3231 *****
3232 012450 012737 000107 001226 TST107: MOV #107,TSTNO
3233 012456 012737 012540 001216 MOV #TST110,NEXT
3234 012464 012703 000017 MOV #17,R3 ;ADDRESS OF SECONDARY REGISTER
3235 ;BCC POLYNOMIAL
3236 012470 110377 166674 MOVB R3,@DQREG ;SELECT BCC POLYNOMIAL REGISTER
3237 012474 012705 000001 MOV #BIT0,R5 ;(R5)=BIT0
3238 012500 010577 166666 MOV R5,@DQSEC ;SET BIT0 IN
3239 ;BCC POLYNOMIAL REGISTER
3240 012504 017704 166662 MOV @DQSEC,R4 ;(R4)=ACTUAL DATA IN
3241 ;BCC POLYNOMIAL REGISTER
3242 012510 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
3243 012512 001401 BEQ 1$ ;BR IF GOOD
3244 012514 104006 HLT 6 ;SECONDARY REGISTER DATA ERROR
3245 012516 040577 166650 1$: BIC R5,@DQSEC ;CLEAR BIT0
3246 012522 017704 166644 MOV @DQSEC,R4 ;READ BCC POLYNOMIAL REGISTER
3247 012526 005005 CLR R5 ;EXPECT BCC POLYNOMIAL REGISTER
3248 ;TO CONTAIN 0
3249 012530 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
3250 012532 001401 BEQ 2$ ;BR IF GOOD
3251 012534 104006 HLT 6 ;SECONDARY REGISTER DATA ERROR
3252 012536 104400 2$: SCOPE ;CHECK FOR ITERATIONS, LOOP
3253
3254 ;SECONDARY REGISTER READ/WRITE TEST
3255 ;SET BIT1 IN BCC POLYNOMIAL REGISTER
3256 ;VERIFY THAT BIT1 WAS SET
3257 ;CLEAR BIT1
3258 ;VERIFY THAT BIT1 WAS CLEARED
3259
3260 ; TEST 110
3261 *****
3262 012540 012737 000110 001226 TST110: MOV #110,TSTNO
3263 012546 012737 012630 001216 MOV #TST111,NEXT
3264 012554 012703 000017 MOV #17,R3 ;ADDRESS OF SECONDARY REGISTER
3265 ;BCC POLYNOMIAL
3266 012560 110377 166604 MOVB R3,@DQREG ;SELECT BCC POLYNOMIAL REGISTER
3267 012564 012705 000002 MOV #BIT1,R5 ;(R5)=BIT1
3268 012570 010577 166576 MOV R5,@DQSEC ;SET BIT1 IN
3269 ;BCC POLYNOMIAL REGISTER
3270 012574 017704 166572 MOV @DQSEC,R4 ;(R4)=ACTUAL DATA IN
3271 ;BCC POLYNOMIAL REGISTER
3272 012600 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
```

POLYNOMIAL REGISTER READ/WRITE TESTS.

```
3273 012602 001401          BEQ      1$          :BR IF GOOD
3274 012604 104006          HLT      6          :SECONDARY REGISTER DATA ERROR
3275 012606 040577 166560  1$: BIC      R5,@DQSEC :CLEAR BIT1
3276 012612 017704 166554  MOV      @DQSEC,R4 :READ BCC POLYNOMIAL REGISTER
3277 012616 005005          CLR      R5        :EXPECT BCC POLYNOMIAL REGISTER
3278                                     :TO CONTAIN 0
3279 012620 020504          CMP      R5,R4     :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3280 012622 001401          BEQ      2$          :BR IF GOOD
3281 012624 104006          HLT      6          :SECONDARY REGISTER DATA ERROR
3282 012626 104400          2$: SCOPE         :CHECK FOR ITERATIONS, LOOP
3283
3284                                     :SECONDARY REGISTER READ/WRITE TEST
3285                                     :SET BIT2 IN BCC POLYNOMIAL REGISTER
3286                                     :VERIFY THAT BIT2 WAS SET
3287                                     :CLEAR BIT2
3288                                     :VERIFY THAT BIT2 WAS CLEARED
3289
3290                                     : TEST 111
3291                                     :*****
3292 012630 012737 000111 001226 TST111: MOV      #111,TSTNO
3293 012636 012737 012720 001216     MOV      #TST112,NEXT
3294 012644 012703 000017          MOV      #17,R3        :ADDRESS OF SECONDARY REGISTER
3295                                     :BCC POLYNOMIAL
3296 012650 110377 166514          MOV#B    R3,@DQREG    :SELECT BCC POLYNOMIAL REGISTER
3297 012654 012705 000004          MOV      #BIT2,R5    : (R5)=BIT2
3298 012660 010577 166506          MOV      R5,@DQSEC   :SET BIT2 IN
3299                                     :BCC POLYNOMIAL REGISTER
3300 012664 017704 166502          MOV      @DQSEC,R4   : (R4)=ACTUAL DATA IN
3301                                     :BCC POLYNOMIAL REGISTER
3302 012670 020504          CMP      R5,R4     :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3303 012672 001401          BEQ      1$          :BR IF GOOD
3304 012674 104006          HLT      6          :SECONDARY REGISTER DATA ERROR
3305 012676 040577 166470  1$: BIC      R5,@DQSEC :CLEAR BIT2
3306 012702 017704 166464  MOV      @DQSEC,R4 :READ BCC POLYNOMIAL REGISTER
3307 012706 005005          CLR      R5        :EXPECT BCC POLYNOMIAL REGISTER
3308                                     :TO CONTAIN 0
3309 012710 020504          CMP      R5,R4     :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3310 012712 001401          BEQ      2$          :BR IF GOOD
3311 012714 104006          HLT      6          :SECONDARY REGISTER DATA ERROR
3312 012716 104400          2$: SCOPE         :CHECK FOR ITERATIONS, LOOP
3313
3314                                     :SECONDARY REGISTER READ/WRITE TEST
3315                                     :SET BIT3 IN BCC POLYNOMIAL REGISTER
3316                                     :VERIFY THAT BIT3 WAS SET
3317                                     :CLEAR BIT3
3318                                     :VERIFY THAT BIT3 WAS CLEARED
3319
3320                                     : TEST 112
3321                                     :*****
3322 012720 012737 000112 001226 TST112: MOV      #112,TSTNO
3323 012726 012737 013010 001216     MOV      #TST113,NEXT
3324 012734 012703 000017          MOV      #17,R3        :ADDRESS OF SECONDARY REGISTER
3325                                     :BCC POLYNOMIAL
3326 012740 110377 166424          MOV#B    R3,@DQREG    :SELECT BCC POLYNOMIAL REGISTER
3327 012744 012705 000010          MOV      #BIT3,R5    : (R5)=BIT3
3328 012750 010577 166416          MOV      R5,@DQSEC   :SET BIT3 IN
```

POLYNOMIAL REGISTER READ/WRITE TESTS.

```
3329                                     :BCC POLYNOMIAL REGISTER
3330 012754 017704 166412             MOV    @DQSEC,R4             :(R4)=ACTUAL DATA IN
3331                                     :BCC POLYNOMIAL REGISTER
3332 012760 020504                     CMP    R5,R4             :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3333 012762 001401                     BEQ    1$                :BR IF GOOD
3334 012764 104006                     HLT    6                 :SECONDARY REGISTER DATA ERROR
3335 012766 040577 166400             1$:  BIC    R5,@DQSEC      :CLEAR BIT3
3336 012772 017704 166374             MOV    @DQSEC,R4        :READ BCC POLYNOMIAL REGISTER
3337 012776 005005                     CLR    R5                :EXPECT BCC POLYNOMIAL REGISTER
3338                                     :TO CONTAIN 0
3339 013000 020504                     CMP    R5,R4             :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3340 013002 001401                     BEQ    2$                :BR IF GOOD
3341 013004 104006                     HLT    6                 :SECONDARY REGISTER DATA ERROR
3342 013006 104400             2$:  SCOPE                :CHECK FOR ITERATIONS, LOOP
3343
3344                                     :SECONDARY REGISTER READ/WRITE TEST
3345                                     :SET BIT4 IN BCC POLYNOMIAL REGISTER
3346                                     :VERIFY THAT BIT4 WAS SET
3347                                     :CLEAR BIT4
3348                                     :VERIFY THAT BIT4 WAS CLEARED
3349
3350                                     : TEST 113
3351                                     :*****
3352 013010 012737 000113 001226     TST113: MOV    #113,TSTNO
3353 013016 012737 013100 001216     MOV    #TST114,NEXT
3354 013024 012703 000017             MOV    #17,R3            :ADDRESS OF SECONDARY REGISTER
3355                                     :BCC POLYNOMIAL
3356 013030 1*0377 166334             MOV    R3,@DQREG        :SELECT BCC POLYNOMIAL REGISTER
3357 013034 012705 000020             MOV    #BIT4,R5         :(R5)=BIT4
3358 013040 010577 166326             MOV    R5,@DQSEC        :SET BIT4 IN
3359                                     :BCC POLYNOMIAL REGISTER
3360 013044 017704 166322             MOV    @DQSEC,R4        :(R4)=ACTUAL DATA IN
3361                                     :BCC POLYNOMIAL REGISTER
3362 013050 020504                     CMP    R5,R4             :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3363 013052 001401                     BEQ    1$                :BR IF GOOD
3364 013054 104006                     HLT    6                 :SECONDARY REGISTER DATA ERROR
3365 013056 040577 166310             1$:  BIC    R5,@DQSEC      :CLEAR BIT4
3366 013062 017704 166304             MOV    @DQSEC,R4        :READ BCC POLYNOMIAL REGISTER
3367 013066 005005                     CLR    R5                :EXPECT BCC POLYNOMIAL REGISTER
3368                                     :TO CONTAIN 0
3369 013070 020504                     CMP    R5,R4             :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3370 013072 001401                     BEQ    2$                :BR IF GOOD
3371 013074 104006                     HLT    6                 :SECONDARY REGISTER DATA ERROR
3372 013076 104400             2$:  SCOPE                :CHECK FOR ITERATIONS, LOOP
3373
3374                                     :SECONDARY REGISTER READ/WRITE TEST
3375                                     :SET BITS IN BCC POLYNOMIAL REGISTER
3376                                     :VERIFY THAT BITS WAS SET
3377                                     :CLEAR BITS
3378                                     :VERIFY THAT BITS WAS CLEARED
3379
3380                                     : TEST 114
3381                                     :*****
3382 013100 012737 000114 001226     TST114: MOV    #114,TSTNO
3383 013106 012737 013170 001216     MOV    #TST115,NEXT
3384 013114 012703 000017             MOV    #17,R3            :ADDRESS OF SECONDARY REGISTER
```

POLYNOMIAL REGISTER READ/WRITE TESTS.

```
3385                                     :BCC POLYNOMIAL
3386 013120 110377 166244               MOV  R3,@DQREG      :SELECT BCC POLYNOMIAL REGISTER
3387 013124 012705 000040               MOV  #BIT5,R5      :(R5)=BIT5
3388 013130 010577 166236               MOV  R5,@DQSEC     :SET BIT5 IN
3389                                     :BCC POLYNOMIAL REGISTER
3390 013134 017704 166232               MOV  @DQSEC,R4     :(R4)=ACTUAL DATA IN
3391                                     :BCC POLYNOMIAL REGISTER
3392 013140 020504                       CMP  R5,R4         :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3393 013142 001401                       BEQ  1$           :BR IF GOOD
3394 013144 104006                       HLT  6           :SECONDARY REGISTER DATA ERROR
3395 013146 040577 166220 1$:          BIC  R5,@DQSEC     :CLEAR BIT5
3396 013152 017704 166214               MOV  @DQSEC,R4     :READ BCC POLYNOMIAL REGISTER
3397 013156 005005                       CLR  R5           :EXPECT BCC POLYNOMIAL REGISTER
3398                                     :TO CONTAIN 0
3399 013160 020504                       CMP  R5,R4         :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3400 013162 001401                       BEQ  2$           :BR IF GOOD
3401 013164 104006                       HLT  6           :SECONDARY REGISTER DATA ERROR
3402 013166 104400 2$:          SCOPE    :CHECK FOR ITERATIONS, LOOP
3403
3404                                     :SECONDARY REGISTER READ/WRITE TEST
3405                                     :SET BIT6 IN BCC POLYNOMIAL REGISTER
3406                                     :VERIFY THAT BIT6 WAS SET
3407                                     :CLEAR BIT6
3408                                     :VERIFY THAT BIT6 WAS CLEARED
3409
3410                                     : TEST 115
3411                                     :*****
3412 013170 012737 000115 001226 TST115: MOV  #115,TSTND
3413 013176 012737 013260 001216       MOV  #TST116,NEXT
3414 013204 012703 000017               MOV  #17,R3       :ADDRESS OF SECONDARY REGISTER
3415                                     :BCC POLYNOMIAL
3416 013210 110377 166154               MOV  R3,@DQREG     :SELECT BCC POLYNOMIAL REGISTER
3417 013214 012705 000100               MOV  #BIT6,R5     :(R5)=BIT6
3418 013220 010577 166146               MOV  R5,@DQSEC     :SET BIT6 IN
3419                                     :BCC POLYNOMIAL REGISTER
3420 013224 017704 166142               MOV  @DQSEC,R4     :(R4)=ACTUAL DATA IN
3421                                     :BCC POLYNOMIAL REGISTER
3422 013230 020504                       CMP  R5,R4         :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3423 013232 001401                       BEQ  1$           :BR IF GOOD
3424 013234 104006                       HLT  6           :SECONDARY REGISTER DATA ERROR
3425 013236 040577 166130 1$:          BIC  R5,@DQSEC     :CLEAR BIT6
3426 013242 017704 166124               MOV  @DQSEC,R4     :READ BCC POLYNOMIAL REGISTER
3427 013246 005005                       CLR  R5           :EXPECT BCC POLYNOMIAL REGISTER
3428                                     :TO CONTAIN 0
3429 013250 020504                       CMP  R5,R4         :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3430 013252 001401                       BEQ  2$           :BR IF GOOD
3431 013254 104006                       HLT  6           :SECONDARY REGISTER DATA ERROR
3432 013256 104400 2$:          SCOPE    :CHECK FOR ITERATIONS, LOOP
3433
3434                                     :SECONDARY REGISTER READ/WRITE TEST
3435                                     :SET BIT7 IN BCC POLYNOMIAL REGISTER
3436                                     :VERIFY THAT BIT7 WAS SET
3437                                     :CLEAR BIT7
3438                                     :VERIFY THAT BIT7 WAS CLEARED
3439
3440                                     : TEST 116
```

POLYNOMIAL REGISTER READ/WRITE TESTS.

```
3441
3442 013260 012737 000116 001226 TST116: MOV #116,TSTNO
3443 013266 012737 013350 001216 MOV #TST117,NEXT
3444 013274 012703 000017 MOV #17,R3
3445 :ADDRESS OF SECONDARY REGISTER
3446 013300 110377 166064 MOVB R3,@DQREG :BCC POLYNOMIAL
3447 013304 012705 000200 MOV #BIT7,R5 :SELECT BCC POLYNOMIAL REGISTER
3448 013310 010577 166056 MOV R5,@DQSEC :(R5)=BIT7
3449 :SET BIT7 IN
3450 013314 017704 166052 MOV @DQSEC,R4 :BCC POLYNOMIAL REGISTER
3451 :BCC POLYNOMIAL REGISTER
3452 013320 020504 CMP R5,R4 :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3453 013322 001401 BEQ 1$ :BR IF GOOD
3454 013324 104006 HLT 6 :SECONDARY REGISTER DATA ERROR
3455 013326 040577 166040 1$: BIC R5,@DQSEC :CLEAR BIT7
3456 013332 017704 166034 MOV @DQSEC,R4 :READ BCC POLYNOMIAL REGISTER
3457 013336 005005 CLR R5 :EXPECT BCC POLYNOMIAL REGISTER
3458 :TO CONTAIN 0
3459 013340 020504 CMP R5,R4 :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3460 013342 001401 BEQ 2$ :BR IF GOOD
3461 013344 104006 HLT 6 :SECONDARY REGISTER DATA ERROR
3462 013346 104400 2$: SCOPE :CHECK FOR ITERATIONS, LOOP
3463
3464 :SECONDARY REGISTER READ/WRITE TEST
3465 :SET BIT8 IN BCC POLYNOMIAL REGISTER
3466 :VERIFY THAT BIT8 WAS SET
3467 :CLEAR BIT8
3468 :VERIFY THAT BIT8 WAS CLEARED
3469
3470 : TEST 117
3471 :*****
3472 013350 012737 000117 001226 TST117: MOV #117,TSTNO
3473 013356 012737 013440 001216 MOV #TST120,NEXT
3474 013364 012703 000017 MOV #17,R3
3475 :ADDRESS OF SECONDARY REGISTER
3476 013370 110377 165774 MOVB R3,@DQREG :BCC POLYNOMIAL
3477 013374 012705 000400 MOV #BIT8,R5 :SELECT BCC POLYNOMIAL REGISTER
3478 013400 010577 165766 MOV R5,@DQSEC :(R5)=BIT8
3479 :SET BIT8 IN
3480 013404 017704 165762 MOV @DQSEC,R4 :BCC POLYNOMIAL REGISTER
3481 :BCC POLYNOMIAL REGISTER
3482 013410 020504 CMP R5,R4 :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3483 013412 001401 BEQ 1$ :BR IF GOOD
3484 013414 104006 HLT 6 :SECONDARY REGISTER DATA ERROR
3485 013416 040577 165750 1$: BIC R5,@DQSEC :CLEAR BIT8
3486 013422 017704 165744 MOV @DQSEC,R4 :READ BCC POLYNOMIAL REGISTER
3487 013426 005005 CLR R5 :EXPECT BCC POLYNOMIAL REGISTER
3488 :TO CONTAIN 0
3489 013430 020504 CMP R5,R4 :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3490 013432 001401 BEQ 2$ :BR IF GOOD
3491 013434 104006 HLT 6 :SECONDARY REGISTER DATA ERROR
3492 013436 104400 2$: SCOPE :CHECK FOR ITERATIONS, LOOP
3493
3494 :SECONDARY REGISTER READ/WRITE TEST
3495 :SET BIT9 IN BCC POLYNOMIAL REGISTER
3496 :VERIFY THAT BIT9 WAS SET
```


POLYNOMIAL REGISTER READ/WRITE TESTS.

```
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562 013620 012737 000122 001226
3563 013626 012737 013710 001216
3564 013634 012703 000017
3565
3566 013640 110377 165524
3567 013644 012705 004000
3568 013650 010577 165516
3569
3570 013654 017704 165512
3571
3572 013660 020504
3573 013662 001401
3574 013664 104006
3575 013666 040577 165500
3576 013672 017704 165474
3577 013676 005005
3578
3579 013700 020504
3580 013702 001401
3581 013704 104006
3582 013706 104400
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592 013710 012737 000123 001226
3593 013716 012737 014000 001216
3594 013724 012703 000017
3595
3596 013730 110377 165434
3597 013734 012705 010000
3598 013740 010577 165426
3599
3600 013744 017704 165422
3601
3602 013750 020504
3603 013752 001401
3604 013754 104006
3605 013756 040577 165410
3606 013762 017704 165404
3607 013766 005005
3608
```

: SECONDARY REGISTER READ/WRITE TEST
: SET BIT11 IN BCC POLYNOMIAL REGISTER
: VERIFY THAT BIT11 WAS SET
: CLEAR BIT11
: VERIFY THAT BIT11 WAS CLEARED
: TEST 122
: *****
TST122: MOV #122,TSTNO
MOV #TST123,NEXT
MOV #17,R3 ;ADDRESS OF SECONDARY REGISTER
;BCC POLYNOMIAL
MOVB R3,@DQREG ;SELECT BCC POLYNOMIAL REGISTER
MOV #BIT11,R5 ;(R5)=BIT11
MOV R5,@DQSEC ;SET BIT11 IN
;BCC POLYNOMIAL REGISTER
MOV @DQSEC,R4 ;(R4)=ACTUAL DATA IN
;BCC POLYNOMIAL REGISTER
CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
BEQ 1\$;BR IF GOOD
HLT 6 ;SECONDARY REGISTER DATA ERROR
1\$: BIC R5,@DQSEC ;CLEAR BIT11
MOV @DQSEC,R4 ;READ BCC POLYNOMIAL REGISTER
CLR R5 ;EXPECT BCC POLYNOMIAL REGISTER
;TO CONTAIN 0
CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
BEQ 2\$;BR IF GOOD
HLT 6 ;SECONDARY REGISTER DATA ERROR
2\$: SCOPE ;CHECK FOR ITERATIONS, LOOP
: SECONDARY REGISTER READ/WRITE TEST
: SET BIT12 IN BCC POLYNOMIAL REGISTER
: VERIFY THAT BIT12 WAS SET
: CLEAR BIT12
: VERIFY THAT BIT12 WAS CLEARED
: TEST 123
: *****
TST123: MOV #123,TSTNO
MOV #TST124,NEXT
MOV #17,R3 ;ADDRESS OF SECONDARY REGISTER
;BCC POLYNOMIAL
MOVB R3,@DQREG ;SELECT BCC POLYNOMIAL REGISTER
MOV #BIT12,R5 ;(R5)=BIT12
MOV R5,@DQSEC ;SET BIT12 IN
;BCC POLYNOMIAL REGISTER
MOV @DQSEC,R4 ;(R4)=ACTUAL DATA IN
;BCC POLYNOMIAL REGISTER
CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
BEQ 1\$;BR IF GOOD
HLT 6 ;SECONDARY REGISTER DATA ERROR
1\$: BIC R5,@DQSEC ;CLEAR BIT12
MOV @DQSEC,R4 ;READ BCC POLYNOMIAL REGISTER
CLR R5 ;EXPECT BCC POLYNOMIAL REGISTER
;TO CONTAIN 0

POLYNOMIAL REGISTER READ/WRITE TESTS.

```
3609 013770 020504          CMP      R5,R4          ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
3610 013772 001401          BEQ      2$             ;BR IF GOOD
3611 013774 104006          HLT      6              ;SECONDARY REGISTER DATA ERROR
3612 013776 104400          2$:      SCOPE          ;CHECK FOR ITERATIONS, LOOP
3613
3614          ;SECONDARY REGISTER READ/WRITE TEST
3615          ;SET BIT13 IN BCC POLYNOMIAL REGISTER
3616          ;VERIFY THAT BIT13 WAS SET
3617          ;CLEAR BIT13
3618          ;VERIFY THAT BIT13 WAS CLEARED
3619
3620          ; TEST 124
3621          ;*****
3622 014000 012737 000124 001226 TST124: MOV      #124,TSTNO
3623 014006 012737 014070 001216      MOV      #TST125,NEXT
3624 014014 012703 000017          MOV      #17,R3          ;ADDRESS OF SECONDARY REGISTER
3625          ;BCC POLYNOMIAL
3626 014020 110377 165344          MOVVB   R3,@DQREG       ;SELECT BCC POLYNOMIAL REGISTER
3627 014024 012705 020000          MOV      #BIT13,R5      ;(R5)=BIT13
3628 014030 010577 165336          MOV      R5,@DQSEC      ;SET BIT13 IN
3629          ;BCC POLYNOMIAL REGISTER
3630 014034 017704 165332          MOV      @DQSEC,R4      ;(R4)=ACTUAL DATA IN
3631          ;BCC POLYNOMIAL REGISTER
3632 014040 020504          CMP      R5,R4          ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
3633 014042 001401          BEQ      1$             ;BR IF GOOD
3634 014044 104006          HLT      6              ;SECONDARY REGISTER DATA ERROR
3635 014046 040577 165320          1$:      BIC      R5,@DQSEC ;CLEAR BIT13
3636 014052 017704 165314          MOV      @DQSEC,R4      ;READ BCC POLYNOMIAL REGISTER
3637 014056 005005          CLR      R5             ;EXPECT BCC POLYNOMIAL REGISTER
3638          ;TO CONTAIN 0
3639 014060 020504          CMP      R5,R4          ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
3640 014062 001401          BEQ      2$             ;BR IF GOOD
3641 014064 104006          HLT      6              ;SECONDARY REGISTER DATA ERROR
3642 014066 104400          2$:      SCOPE          ;CHECK FOR ITERATIONS, LOOP
3643
3644          ;SECONDARY REGISTER READ/WRITE TEST
3645          ;SET BIT14 IN BCC POLYNOMIAL REGISTER
3646          ;VERIFY THAT BIT14 WAS SET
3647          ;CLEAR BIT14
3648          ;VERIFY THAT BIT14 WAS CLEARED
3649
3650          ; TEST 125
3651          ;*****
3652 014070 012737 000125 001226 TST125: MOV      #125,TSTNO
3653 014076 012737 014160 001216      MOV      #TST126,NEXT
3654 014104 012703 000017          MOV      #17,R3          ;ADDRESS OF SECONDARY REGISTER
3655          ;BCC POLYNOMIAL
3656 014110 110377 165254          MOVVB   R3,@DQREG       ;SELECT BCC POLYNOMIAL REGISTER
3657 014114 012705 040000          MOV      #BIT14,R5      ;(R5)=BIT14
3658 014120 010577 165246          MOV      R5,@DQSEC      ;SET BIT14 IN
3659          ;BCC POLYNOMIAL REGISTER
3660 014124 017704 165242          MOV      @DQSEC,R4      ;(R4)=ACTUAL DATA IN
3661          ;BCC POLYNOMIAL REGISTER
3662 014130 020504          CMP      R5,R4          ;ARE EXPECTED AND RECEIVED DATA THE SAME ?
3663 014132 001401          BEQ      1$             ;BR IF GOOD
3664 014134 104006          HLT      6              ;SECONDARY REGISTER DATA ERROR
```

POLYNOMIAL REGISTER READ/WRITE TESTS.

```
3665 014136 040577 165230      1$:  BIC      R5,@DQSEC      :CLEAR BIT14
3666 014142 017704 165224      MOV      @DQSEC,R4      :READ BCC POLYNOMIAL REGISTER
3667 014146 005005                CLR      R5              :EXPECT BCC POLYNOMIAL REGISTER
3668                                :TO CONTAIN 0
3669 014150 020504                CMP      R5,R4          :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3670 014152 001401                BEQ      2$              :BR IF GOOD
3671 014154 104006                HLT      6                :SECONDARY REGISTER DATA ERROR
3672 014156 104400      2$:  SCOPE                :CHECK FOR ITERATIONS, LOOP
3673
3674                                :SECONDARY REGISTER READ/WRITE TEST
3675                                :SET BIT15 IN BCC POLYNOMIAL REGISTER
3676                                :VERIFY THAT BIT15 WAS SET
3677                                :CLEAR BIT15
3678                                :VERIFY THAT BIT15 WAS CLEARED
3679
3680                                : TEST 126
3681                                :*****
3682 014160 012737 000126 001226  TST126: MOV      #126,TSTNO
3683 014166 012737 014250 001216  MOV      #.EOP,NEXT
3684 014174 012703 000017                MOV      #17,R3          :ADDRESS OF SECONDARY REGISTER
3685                                :BCC POLYNOMIAL
3686 014200 110377 165164                MOV      R3,@DQREG      :SELECT BCC POLYNOMIAL REGISTER
3687 014204 012705 100000                MOV      #BIT15,R5      : (R5)=BIT15
3688 014210 010577 165156                MOV      R5,@DQSEC      :SET BIT15 IN
3689                                :BCC POLYNOMIAL REGISTER
3690 014214 017704 165152                MOV      @DQSEC,R4      : (R4)=ACTUAL DATA IN
3691                                :BCC POLYNOMIAL REGISTER
3692 014220 020504                CMP      R5,R4          :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3693 014222 001401                BEQ      1$              :BR IF GOOD
3694 014224 104006                HLT      6                :SECONDARY REGISTER DATA ERROR
3695 014226 040577 165140      1$:  BIC      R5,@DQSEC      :CLEAR BIT15
3696 014232 017704 165134      MOV      @DQSEC,R4      :READ BCC POLYNOMIAL REGISTER
3697 014236 005005                CLR      R5              :EXPECT BCC POLYNOMIAL REGISTER
3698                                :TO CONTAIN 0
3699 014240 020504                CMP      R5,R4          :ARE EXPECTED AND RECEIVED DATA THE SAME ?
3700 014242 001401                BEQ      2$              :BR IF GOOD
3701 014244 104006                HLT      6                :SECONDARY REGISTER DATA ERROR
3702 014246 104400      2$:  SCOPE                :CHECK FOR ITERATIONS, LOOP
```

```

3703
3704
3705
3706
3707
3708
3709
3710 014250 005037 001234 .EOP: CLR LSTERR ;CLEAR LAST ERROR PC
3711 014254 005037 001312 CLR ERRFLG ;CLEAR ERROR FLAG
3712 014260 005237 001230 INC PASCNT ;UPDATE PASS COUNT
3713 014264 104402 TYPE
3714 014266 016500 MEPASS
3715 014270 104402 TYPE
3716 014272 016661 MCSRX
3717 014274 104411 CNVRT
3718 014276 014406 XCSR
3719 014300 104402 TYPE
3720 014302 016667 MVECX
3721 014304 104411 CNVRT
3722 014306 014414 XVEC
3723 014310 104402 TYPE
3724 014312 016675 MPASSX
3725 014314 104411 CNVRT
3726 014316 014422 XPASS
3727 014320 104402 TYPE
3728 014322 016706 MERRX
3729 014324 104411 CNVRT
3730 014326 014430 XERR
3731 014330 013777 001230 164644 MOV PASCNT,@LIGHTS ;DISPLAY PASS COUNT
3732 014336 005337 001276 DEC SAVNUM
3733 014342 001013 BNE RESTRT
3734 014344 013737 001504 001276 MOV DQNUM,SAVNUM
3735 014352 013701 000042 MOV @#42,R1 ;CHECK FOR ACT-11 OR DDP
3736 014356 001405 BEQ RESTRT ;IF NOT, CONTINUE TESTING
3737 014360 000005 RESET
3738 014362
3739 014362 004711 LOGICAL: JSR PC,(R1)
3740 014364 000240 NOP
3741 014366 000240 NOP
3742 014370 000240 NOP
3743 014372 104414 RESTRT: CKSWR
3744 014374 012737 002254 001214 MOV #TST1,RETURN
3745 014402 000137 002254 JMP TST1
3746 014406 000001 XCSR: 1
3747 014410 006 002 .BYTE 6,2
3748 014412 001360 DQRCSR
3749 014414 000001 XVEC: 1
3750 014416 003 002 .BYTE 3,2
3751 014420 001350 DQRVEC
3752 014422 000001 XPASS: 1
3753 014424 006 002 .BYTE 6,2
3754 014426 001230 PASCNT
3755 014430 000001 XERR: 1
3756 014432 006 002 .BYTE 6,2
3757 014434 001232 ERRCNT
3758
  
```

```

3759                                     ;SCOPE LOOP AND INTERATION HANDLER
3760
3761 014436 104414                       .SCOPE: CKSWR
3762 014440 032777 040000 164532      TTST: BIT #BIT14,@SWR
3763 014446 001407                       BEQ 1$
3764 014450 000432                       BR 3$
3765 014452 105777 164526               TSTB @TKCSR
3766 014456 100027                       BPL 3$
3767 014460 017700 164522               MOV @TKDDBR,R0
3768 014464 000412                       BR 2$
3769 014466 032777 004000 164504 1$: BIT #SW11,@SWR
3770 014474 001006                       BNE 2$
3771 014476 005237 001224               INC LPCNT
3772 014502 023737 001224 001222      CMP LPCNT,ICOUNT
3773 014510 001012                       BNE 3$
3774 014512 105037 001312               2$: CLRB ERRFLG
3775 014516 005037 001224               CLR LPCNT
3776 014522 012737 002000 001222      MOV #2000,ICOUNT
3777 014530 013737 001216 001214      MOV NEXT,RETURN
3778 014536 013716 001214               3$: MOV RETURN,(SP)
3779 014542 000002                       RTI
3780 014544 001407                       BRW: 1407
3781 014546 000432                       BRX: 432
3782
3783                                     ;CHECK FOR FREEZE ON CURRENT DATA
3784
3785 014550 104414                       .SCOPE1: CKSWR
3786 014552 032777 001000 164420      BIT #SW09,@SWR
3787 014560 001402                       BEQ 1$
3788 014562 013716 001220               MOV LOCK,(SP)
3789 014566 000002                       1$: RTI
3790
3791                                     ;TELETYPE OUTPUT ROUTINE
3792
3793 014570 010546                       .TYPE: MOV R5,-(SP)
3794 014572 017605 000002               MOV @2(SP),R5
3795 014576 062766 000002 000002      ADD #2,2(SP)
3796 014604 005737 016260               1$: TST @#RDSW
3797 014610 001004                       BNE 300$
3798 014612 032777 010000 164360      BIT #SW12,@SWR
3799 014620 001024                       BNE 3$
3800 014622 105715                       300$: TSTB (R5)
3801 014624 100014                       BPL 2$
3802 014626 105777 164356               TSTB @TPCSR
3803 014632 100375                       BPL -4
3804 014634 012777 000015 164350      MOV #15,@TPDDBR
3805 014642 105777 164342               TSTB @TPCSR
3806 014646 100375                       BPL -4
3807 014650 012777 000012 164334      MOV #12,@TPDDBR
3808 014656 105777 164326               2$: TSTB @TPCSR
3809 014662 100375                       BPL 2$
3810 014664 112577 164322               MOVB (R5)+,@TPDDBR
3811 014670 001345                       BNE 1$
3812 014672 012605                       3$: MOV (SP)+,R5
3813 014674 000002                       RTI
3814

```

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

```

3815                                     ;ASCII STRING INPUT ROUTINE
3816
3817 014676 010346      .INSTR: MOV     R3,-(SP)
3818 014700 010446      MOV     R4,-(SP)
3819 014702 017637 000004 014720      MOV     @4(SP),.MSG
3820 014710 062766 000002 000004      ADD     #2,4(SP)
3821 014716 104402      .INST1: TYPE
3822 014720 000000      .MSG:  0
3823 014722 012704 017052      MOV     #INBUF,R4
3824 014726 012703 000007      MOV     #7,R3
3825 014732 105777 164246      1$:   TSTB   @TKCSR
3826 014736 100375      BPL     1$
3827 014740 117714 164242      MOVB   @TKDBR,(R4)
3828 014744 142714 000200      BICB   #200,(R4)
3829 014750 121427 000025      CMPB   (R4),#25
3830 014754 001003      BNE     200$
3831 014756 104402 016440      TYPE ,MCRLF
3832 014762 000755      BR     .INST1
3833 014764 122427 000015      200$: CMPB   (R4)+,#15
3834 014770 001423      BEQ    INSTR2
3835 014772 117777 164210 164212      MOVB   @TKDBR,@TPDBR
3836 015000 105777 164204      2$:   TSTB   @TPCSR
3837 015004 100375      BPL     2$
3838 015006 005303      DEC     R3
3839 015010 001350      BNE     1$
3840 015012 000402      BR     .INSTG
3841 015014 010346      .INSTE: MOV    R3,-(SP)
3842 015016 010446      MOV    R4,-(SP)
3843 015020 104402      .INSTG: TYPE
3844 015022 016434      MQM
3845 015024 005737 016260      TST    @#RDSW
3846 015030 001402      BEQ    400$
3847 015032 104402 016440      TYPE ,MCRLF
3848 015036 000727      400$: BR     .INST1
3849 015040 012604      INSTR2: MOV   (SP)+,R4
3850 015042 012603      MOV   (SP)+,R3
3851 015044 000002      RTI
3852
3853                                     ;CONVERT ASCII STRING TO OCTAL
3854
3855 015046 010546      .PARAM: MOV    R5,-(SP)
3856 015050 010446      MOV    R4,-(SP)
3857 015052 016605 000004      MOV    4(SP),R5
3858 015056 012537 015252      MOV    (R5)+,LOLIM
3859 015062 012537 015254      MOV    (R5)+,HILIM
3860 015066 012537 015256      MOV    (R5)+,DEVADR
3861 015072 112537 015260      MOVB   (R5)+,LOBITS
3862 015076 112537 015261      MOVB   (R5)+,ADRCNT
3863 015102 010566 000004      MOV    R5,4(SP)
3864 015106 005005      PARAM1: CLR   R5
3865 015110 012704 017052      MOV    #INBUF,R4
3866 015114 122714 000015      CMPB   #15,(R4)
3867 015120 001420      BEQ    PARERR
3868 015122 121427 000060      1$:   CMPB   (R4),#60
3869 015126 002415      BLT    PARERR
3870 015130 121427 000067      CMPB   (R4),#67
  
```

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

```

3871 015134 003012          BGT    PARERR
3872 015136 142714 000060  BICB   #60,(R4)
3873 015142 152405          BISB   (R4)+,R5
3874 015144 122714 000015  CMPB   #15,(R4)
3875 015150 001414          BEQ    LIMITS
3876 015152 006305          ASL    R5
3877 015154 006305          ASL    R5
3878 015156 006305          ASL    R5
3879 015160 000760          BR     1$
3880 015162 122714 000015  PARERR: CMPB   #15,(R4)      ;IS FIRST CHARACTER A <CR>
3881 015166 001003          BNE    120$
3882 015170 005737 016260  TST    @#RDSW      ;IS CKSWR ROUTINE BEING USED
3883 015174 001023          BNE    PARTI
3884 015176 104404          120$:  INSTER
3885 015200 000742          BR     PARAM1
3886
3887          ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
3888
3889 015202 020537 015254  LIMITS: CMP    R5,HILIM
3890 015206 101365          BHI    PARERR
3891 015210 020537 015252  CMP    R5,LOLIM
3892 015214 103762          BLO    PARERR
3893 015216 133705 015260  BITB   LOBITS,R5
3894 015222 001357          BNE    PARERR
3895
3896          ;STORE NUMBER AT SPECIFIED ADDRESS
3897
3898 015224 013704 015256  1$:    MOV    DEVADR,R4
3899 015230 010524          MOV    R5,(R4)+
3900 015232 062705 000002  ADD    #2,R5
3901 015236 105337 015261  DECB   ADCRNT
3902 015242 001372          BNE    1$
3903 015244 012604          PARTI: MOV    (SP)+,R4
3904 015246 012605          MOV    (SP)+,R5
3905 015250 000002          RTI
3906 015252 000000          LOLIM: 0
3907 015254 000000          HILIM: 0
3908 015256 000000          DEVADR: 0
3909 015260 000000          LOBITS: 0
3910          ADCRNT=LOBITS+1
3911
3912          ;SAVE PC OF TEST THAT FAILED AND R0-R5
3913
3914 015262 016637 000004 001274 .SAV05: MOV    4(SP),SAVPC
3915
3916          ;SAVE R0-R5
3917
3918 015270 010537 001270  SV05:  MOV    R5,SAVR5
3919 015274 010437 001266  MOV    R4,SAVR4
3920 015300 010337 001264  MOV    R3,SAVR3
3921 015304 010237 001262  MOV    R2,SAVR2
3922 015310 010137 001260  MOV    R1,SAVR1
3923 015314 010037 001256  MOV    R0,SAVR0
3924 015320 000002          RTI
3925
3926          ;RESTORE R0-R5
  
```

```

3927
3928 015322 013700 001256      .RES05: MOV      SAVR0,R0
3929 015326 013701 001260      MOV      SAVR1,R1
3930 015332 013702 001262      MOV      SAVR2,R2
3931 015336 013703 001264      MOV      SAVR3,R3
3932 015342 013704 001266      MOV      SAVR4,R4
3933 015346 013705 001270      MOV      SAVR5,R5
3934 015352 000002      RTI
3935
3936      ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
3937
3938 015354 104402      .CONVR: TYPE
3939 015356 016440      MCRLF
3940 015360 010046      .CNVRT: MOV      R0,-(SP)
3941 015362 010146      MOV      R1,-(SP)
3942 015364 010346      MOV      R3,-(SP)
3943 015366 010446      MOV      R4,-(SP)
3944 015370 010546      MOV      R5,-(SP)
3945 015372 017601 000012      MOV      @12(SP),R1
3946 015376 013737 017114 001250      MOV      TEMP,TEMP3
3947 015404 062766 000002 000012      ADD      #2,12(SP)
3948 015412 012137 015574      MOV      (R1)+,WRDCNT
3949 015416 112137 015576      1$: MOVB   (R1)+,CHRCNT
3950 015422 112137 015577      MOVB   (R1)+,SPACNT
3951 015426 013137 015600      MOV      @ (R1)+,BINWRD
3952 015432 013704 015600      2$: MOV      BINWRD,R4
3953 015436 113705 015576      MOVB   CHRCNT,R5
3954 015442 012700 017114      MOV      #TEMP,R0
3955 015446 010403      3$: MOV      R4,R3
3956 015450 042703 177770      BIC     #177770,R3
3957 015454 062703 000060      ADD     #060,R3
3958 015460 110320      MOVB   R3,(R0)+
3959 015462 000241      CLC
3960 015464 006004      ROR    R4
3961 015466 000241      CLC
3962 015470 006004      ROR    R4
3963 015472 000241      CLC
3964 015474 006004      ROR    R4
3965 015476 005305      DEC    R5
3966 015500 001362      BNE    3$
3967 015502 012703 017156      MOV     #MDATA,R3
3968 015506 114023      4$: MOVB   -(R0),(R3)+
3969 015510 105337 015576      DECB   CHRCNT
3970 015514 001374      BNE    4$
3971 015516 105737 015577      TSTB   SPACNT
3972 015522 001405      BEQ    6$
3973 015524 112723 000040      5$: MOVB   #040,(R3)+
3974 015530 105337 015577      DECB   SPACNT
3975 015534 001373      BNE    5$
3976 015536 105013      6$: CLRB   (R3)
3977 015540 104402      TYPE
3978 015542 017156      MDATA
3979 015544 005337 015574      DEC    WRDCNT
3980 015550 001322      BNE    1$
3981 015552 013737 001250 017114      MOV     TEMP3,TEMP
3982 015560 012605      MOV     (SP)+,R5
  
```

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

3983	015562	012604				MOV	(SP)+,R4	
3984	015564	012603				MOV	(SP)+,R3	
3985	015566	012601				MOV	(SP)+,R1	
3986	015570	012600				MOV	(SP)+,R0	
3987	015572	000002				RTI		
3988	015574	000000			WRDCNT:	0		
3989	015576	000000			CHRCNT:	0		
3990		015577			SPACNT=	CHRCNT+1		
3991	015600	000000			BINWRD:	0		
3992							:TRAP DISPATCH SERVICE	
3993							:ARGUMENT OF TRAP IS EXTRACTED	
3994							:AND USED AS OFFSET TO OBTAIN POINTER	
3995							:TO SELECTED SUBROUTINE	
3996								
3997	015602	011646			.TRPSR:	MOV	(SP),-(SP)	:GET PC OF RETURN
3998	015604	162716	000002			SUB	#2,(SP)	:PC OF TRAP
3999	015610	017616	000000			MOV	@(SP),(SP)	:GET TRP
4000	015614	006316			TRPOK:	ASL	(SP)	:MULTIPLY TRAP ARG BY 2
4001	015616	042716	177001			BIC	#177001,(SP)	:CLEAR UNWANTED BITS
4002	015622	062716	001314			ADD	#.TRPTAB,(SP)	:POINTER TO SUBROUTINE ADDRESS
4003	015626	017616	000000			MOV	@(SP),(SP)	:SUBROUTINE ADDRESS
4004	015632	000136				JMP	@(SP)+	:GC TO SUBROUTINE
4005								
4006							:ERROR HANDLER	
4007								
4008	015634	104414			.HLT:	CKSWR		
4009	015636	032777	010000	163334		BIT	#SW12,@SWR	
4010	015644	001406				BEQ	XBX	
4011	015646	105777	163336			TSTB	@TPCSR	
4012	015652	100003				BPL	XBX	
4013	015654	112777	000207	163330		MOVB	#207,@TPDBR	
4014	015662	032777	020000	163310	XBX:	BIT	#SW13,@SWR	
4015	015670	001074				BNE	HALTS	
4016	015672	021637	001234			CMP	(SP),LSTERR	
4017	015676	001404				BEQ	1\$	
4018	015700	011637	001234			MOV	(SP),LSTERR	
4019	015704	105037	001312			CLRB	ERRFLG	
4020	015710	104406			1\$:	SAV05		
4021	015712	011605				MOV	(SP),R5	
4022	015714	162705	000002			SUB	#2,R5	
4023	015720	011504				MOV	(R5),R4	
4024	015722	006304				ASL	R4	
4025	015724	061504				ADD	(R5),R4	
4026	015726	006304				ASL	R4	
4027	015730	042704	177001			BIC	#177001,R4	
4028	015734	062704	017670			ADD	#.ERRTAB,R4	
4029	015740	012437	016032			MOV	(R4)+,ERRMSG	
4030	015744	012437	016044			MOV	(R4)+,DATAHD	
4031	015750	011437	016056			MOV	(R4),DATABP	
4032	015754	105737	001312			TSTB	ERRFLG	
4033	015760	001403				BEQ	TYPMSG	
4034	015762	005737	016056			TST	DATABP	
4035	015766	001027				BNE	TYPDAT	
4036	015770	104402			TYPMSG:	TYPE		
4037	015772	016717				MTSTN		
4038	015774	104411				CNVRT		

4039	015776	016156				XTSTN		
4040	016000	104402				TYPE		
4041	016002	017005				MERRPC		
4042	016004	104411				CNVRT		
4043	016006	016150				ERTABO		
4044	016010	104402				TYPE		
4045	016012	016440				MCRLF		
4046	016014	112737	177777	001312		MOVB	#-1,ERRFLG	
4047	016022	005737	016032			TST	ERRMSG	
4048	016026	001402				BEQ	WRKO.FM	
4049	016030	104402				TYPE		
4050	016032	000000				ERRMSG: 0		
4051	016034					WRKO.FM:		
4052	016034	005737	016044			TST	DATAHD	
4053	016040	001402				BEQ	TYPDAT	
4054	016042	104402				TYPE		
4055	016044	000000				DATAHD: 0		
4056	016046	005737	016056			TYPDAT: TST	DATABP	
4057	016052	001402				BEQ	RESREG	
4058	016054	104410				CONVRT		
4059	016056	000000				DATABP: 0		
4060	016060	104407				RESREG: RES05		
4061	016062	005777	163112			HALTS: TST	@SWR	
4062	016066	100005				BPL	EXITER	
4063	016070	010046				PUSHRO		
4064	016072	016600	000002			MOV	2(SP),R0	
4065	016076	000000				HALT		
4066	016100	012600				POPPO		
4067	016102	104414				EXITER: CKSWR		
4068	016104	005237	001232			INC	ERRCNT	
4069	016110	032777	000400	163062		BIT	#SW08,@SWR	
4070	016116	001007				BNE	1\$	
4071	016120	032777	002000	163052		BIT	#SW10,@SWR	
4072	016126	001407				BEQ	2\$	
4073	016130	013737	001216	001214		MOV	NEXT,RETURN	
4074	016136	012706	001200			1\$: MOV	#STACK,SP	
4075	016142	000177	163046			JMP	@RETURN	
4076	016146	000002				2\$: RTI		
4077	016150	000001				ERTABO: 1		
4078	016152	006	002			.BYTE	6,2	
4079	016154	001274				SAVPC		
4080	016156	000001				XTSTN: 1		
4081	016160	003	002			.BYTE	3,2	
4082	016162	001226				TSTNO		
4083						:ENTER HERE ON POWER FAILURE		
4084								
4085								
4086	016164					.PFAIL:		
4087	016164	012737	016176	000024		MOV	#RESTART,24	:SET UP FOR POWER UP TRAP
4088	016172	000000				HALT		:HALT ON POWER DOWN NORMAL
4089	016174	000777				BR	.	
4090								
4091						:PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED		
4092								
4093	016176					RESTAR:		
4094	016176	012737	016164	000024		MOV	#.PFAIL,24	:SET UP FOR POWER FAILURE

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

4095	016204	012706	001200		MOV	#STACK,SP	
4096	016210	005037	017114		CLR	TEMP	
4097	016214	005237	017114		INC	TEMP	
4098	016220	001375			BNE	.-4	
4099	016222	104402			TYPE		
4100	016224	016442			MPFAIL		
4101	016226	104411			CNVRT		
4102	016230	016252			PFTAB		
4103	016232	005037	001312		CLR	ERRFLG	
4104	016236	005037	001234		CLR	LSTERR	
4105	016242	104412			MSTCLR		
4106	016244	104413			MEMCLR		
4107	016246	000177	162742		JMP	@RETURN	
4108	016252	000001		PFTAB:	1		
4109	016254	003	002	.BYTE	3,2		
4110	016256	001226			TSTNO		
4111							
4112							
4113							
4114							
4115							
4116	016260	000000					
4117							
4118							
4119	016262	005737	000042				
4120	016266	001042		.CKSWR:	TST	@#42	
4121	016270	022737	000176	001200	BNE	OUT	
4122	016276	001036			CMP	#SWREG,SWR	:SOFTWARE SWITCH REGISTER PRESENT
4123	016300	105777	162700		BNE	OUT	:NO, GET OUT
4124	016304	100033			TSTB	@TKCSR	:YES, WAIT FOR
4125	016306	017737	162674	014720	BPL	OUT	:READY, GET CHARACTER
4126	016314	042737	177600	014720	MOV	@TKDBR,.MSG	:AND STRIP OFF
4127	016322	122737	000007	014720	BIC	#177600,.MSG	:THE GARBAGE
4128	016330	001021			CMPB	#7,.MSG	:IS IT A <^G>
4129	016332	104402	016410		BNE	OUT	
4130	016336	005137	016260		TYPE,\$CNTG		
4131	016342	104402	016414		.CNTLU: COM	@#RDSW	
4132	016346	104411	016402		TYPE,\$MSWR		
4133	016352	104403	016423		CNVRT,\$WREGC		
4134	016356	104405			INSTR,\$MNEW		
4135	016360	000000			PARAM		
4136	016362	177777			0		
4137	016364	000176			177777		
4138	016366	000	001		SWREG		
4139	016370	104402	016440		.BYTE	0,1	
4140	016374	005037	016260		TYPE,MCRLF		
4141	016400	000002		OUT:	CLR	@#RDSW	
4142	016402	000001			RTI		
4143	016404	006	002	SWREGC:	1		
4144	016406	000176		.BYTE	6,2		
4145	016410	057377	000107		SWREG		
4146	016414	051777	051127	020075	\$.CNTG:	.ASCIZ <377>/^G/	
4147	016422	000			\$.MSWR:	.ASCIZ <377>/SWR= /	
4148	016423	040	047040	053505	\$.MNEW:	.ASCIZ / NEW= /	
4149	016430	020075	000				
4150	016434				.EVEN		

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

4151	016434	020040	000077		
4152	016440	000377			
4153	016442	050377	051127	043040	
4154	016450	044501	042514	027104	
4155	016456	051040	051505	040524	
4156	016464	052122	040440	020124	
4157	016472	042524	052123	000040	
4158	016500	042777	042116	050040	MEPASS: .ASCIZ <377>/END PASS CZDQA /
4159	016506	051501	020123	055103	
4160	016514	050504	020101	000040	
4161	016522	051377	000		MR: .ASCIZ <377>/R/
4162	016525	377	051120	043517	MERR2: .ASCIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT./
4163	016532	040522	020115	047111	
4164	016540	044504	040503	042524	
4165	016546	020123	047516	042040	
4166	016554	053105	041511	051505	
4167	016562	050040	042522	042523	
4168	016570	052116	000056		
4169	016574	044777	051516	043125	MERR3: .ASCIZ <377>/INSUFFICIENT DATA! /
4170	016602	044506	044503	047105	
4171	016610	020124	040504	040524	
4172	016616	000041			
4173	016620	052377	051505	020124	MTSTPC: .ASCIZ <377>/TEST PC-/
4174	016626	041520	000055		
4175	016632	046377	041517	020113	MLOCK: .ASCIZ <377>/LOCK ON SELECTED TEST/
4176	016640	047117	051440	046105	
4177	016646	041505	042524	020104	
4178	016654	042524	052123	000	
4179	016661	103	051123	020072	MCSRX: .ASCIZ /CSR: /
4180	016666	000			
4181	016667	126	041505	020072	MVECX: .ASCIZ /VEC: /
4182	016674	000			
4183	016675	120	051501	042523	MPASSX: .ASCIZ /PASSES: /
4184	016702	035123	000040		
4185	016706	051105	047522	051522	MERRX: .ASCIZ /ERRORS: /
4186	016714	020072	000		
4187	016717	377	052377	051505	MTSTN: .ASCIZ <377><377> /TEST NO: /
4188	016724	020124	047516	020072	
4189	016732	000			
4190	016733	377	042523	020124	MNEW: .ASCIZ <377>/SET SWITCH REG TO DQ11'S DESIRED ACTIVE./
4191	016740	053523	052111	044103	
4192	016746	051040	043505	052040	
4193	016754	020117	050504	030461	
4194	016762	051447	042040	051505	
4195	016770	051111	042105	040440	
4196	016776	052103	053111	027105	
4197	017004	000			
4198	017005	120	035103	000040	MERRPC: .ASCIZ /PC: /
4199	017012	046777	050101	047440	XHEAD: .ASCIZ <377>/MAP OF DQ11 STATUS/<377>
4200	017020	020106	050504	030461	
4201	017026	051440	040524	052524	
4202	017034	177523	000		
4203		017040			.EVEN
4204	017040	000002			XSTATQ: 2
4205	017042	006	003		.BYTE 6.3
4206	017044	001244			TEMP1

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

4207	017046	006	002		.BYTE 6.2
4208	017050	001246			TEMP2
4209				.EVEN	
4210					
4211					;BUFFERS FOR INPUT-OUTPUT
4212					
4213	017052	000000		INBUF:	0
4214		017114		.=.+40	
4215	017114	000000		TEMP:	0
4216		017156		.=.+40	
4217	017156	000000		MDATA:	0
4218		017220		.=.+40	

TABLES FOR ERROR MESSAGES.

```

4219 017220 000002 .MEMCLR: RTI
4220 017222 000002 .MSTCLR: RTI
4221
017224 042101 051104 051505 EMO: .ASCIZ ;TABLE OF ERROR MESSAGES AND ERROR DATA POINTERS
017261 120 044522 040515 EM1: .ASCIZ /ADDRESS SELECT ERROR-TIMEOUT/
017316 042522 042503 053111 EM2: .ASCIZ /PRIMARY REG ADDRESSING ERROR/
017356 051124 047101 046523 EM3: .ASCIZ /RECEIVER CONTROL REG DATA ERROR/
017421 105 051122 051117 EM4: .ASCIZ /TRANSMITTER CONTROL REG DATA ERROR/
017446 051777 041505 047117 EM5: .ASCIZ /ERROR REG DATA ERROR/
017503 123 041505 047117 EM6: .ASCIZ <377>/SECONDARY REG ADDRESS ERROR/
017534 051377 043505 051511 DH0: .ASCIZ /SECONDARY REG DATA ERROR/
017560 042777 050130 041505 DH1: .ASCIZ <377>/REGISTER ADDRESSED/
017621 377 054105 042520 DH2: .ASCIZ <377>/EXPECTED RECEIVED REG ADDRESS/
.EVEN <377>/EXPECTED RECEIVED SEC ADR SEC REG/
  
```

```

4222 017670 017224 .ERRTAB:EMO
4223 017672 017534 DH0
4224 017674 017750 DT0
4225 017676 017261 EM1
4226 017700 017560 DH1
4227 017702 017756 DT1
4228 017704 017316 EM2
4229 017706 017560 DH1
4230 017710 017756 DT1
4231 017712 017356 EM3
4232 017714 017560 DH1
4233 017716 017756 DT1
4234 017720 017421 EM4
4235 017722 017560 DH1
4236 017724 017756 DT1
4237 017726 000000 0
4238 017730 000000 0
4239 017732 000000 0
4240 017734 017503 EM6
4241 017736 017621 DH2
4242 017740 017774 DT2
4243 017742 017446 EM5
4244 017744 017621 DH2
4245 017746 017774 DT2
4246
4247 017750 000001 DT0: 1
4248 017752 006 000 .BYTE 6,0
4249 017754 001270 SAVR5
4250 017756 000003 DT1: 3
4251 017760 006 004 .BYTE 6,4
4252 017762 001270 SAVR5
4253 017764 006 004 .BYTE 6,4
4254 017766 001266 SAVR4
4255 017770 006 000 .BYTE 6,0
4256 017772 001264 SAVR3
4257 017774 000004 DT2: 4
4258 017776 006 004 .BYTE 6,4
4259 020000 001270 SAVR5
4260 020002 006 004 .BYTE 6,4
4261 020004 001266 SAVR4
4262 020006 006 004 .BYTE 6,4
  
```

TABLES FOR ERROR MESSAGES.

4263	020010	001372			DQSEC
4264	020012	002	000	.BYTE	2,0
4265	020014	001264			SAVR3
4266					:DATA TABLE FOR SECONDARY REGISTER ADDRESSING TEST
4267					
4268	020016	000000		DATAB:	0
4269	020020	010421			10421
4270	020022	021042			21042
4271	020024	031463			31463
4272	020026	042104			42104
4273	020030	052525			52525
4274	020032	063146			63146
4275	020034	073567			73567
4276	020036	104210			104210
4277	020040	114631			114631
4278	020042	005212			5212
4279	020044	000000			0
4280	020046	146314			146314
4281	020050	000000			0
4282	020052	000000			0
4283	020054	177777			177777
4284		000001		.END	

TST40	006054	1967	1999#																		
TST41	006144	2000	2032#																		
TST42	006234	2033	2065#																		
TST43	006324	2066	2098#																		
TST44	006414	2099	2131#																		
TST45	006504	2132	2164#																		
TST46	006574	2165	2197#																		
TST47	006660	2198	2229#																		
TST5	003100	1140	1159#																		
TST50	006744	2230	2261#																		
TST51	007030	2262	2293#																		
TST52	007114	2294	2327#																		
TST53	007262	2367#																			
TST54	007430	2408#																			
TST55	007554	2409	2446#																		
TST56	007644	2447	2476#																		
TST57	007734	2477	2506#																		
TST6	003164	1160	1180#																		
TST60	010024	2507	2536#																		
TST61	010114	2537	2566#																		
TST62	010204	2567	2596#																		
TST63	010274	2597	2626#																		
TST64	010364	2627	2656#																		
TST65	010454	2657	2686#																		
TST66	010544	2687	2716#																		
TST67	010634	2717	2746#																		
TST7	003364	1181	1228#																		
TST70	010724	2747	2776#																		
TST71	011014	2777	2806#																		
TST72	011104	2807	2836#																		
TST73	011174	2837	2866#																		
TST74	011264	2867	2896#																		
TST75	011354	2897	2926#																		
TST76	011444	2927	2956#																		
TST77	011534	2957	2986#																		
TTST	014446	1019*	1020*	1022*	1023*	3763#															
TXBA.P=	000002	623#																			
TXBA.S=	000006	627#																			
TXWC.P=	000003	624#																			
TXWC.S=	000007	628#																			
TX.BCC=	000016	636#																			
TX.MUX=	000013	633#																			
TYPDAT	016046	4035	4053	4056#																	
TYPE =	104402	761	844#	968	976	986	993	1018	1036	1061	3713	3715	3719	3723							
		3727	3821	3831	3843	3847	3938	3977	4036	4040	4044	4049	4054	4099							
		4129	4131	4139																	
TYPMSG	015770	4033	4036#																		
VECMAP	000056	663#	760																		
WRDCNT	015574	3948*	3979*	3988#																	
WRKO.F	016034	4048	4051#																		
XBX	015662	4010	4012	4014#																	
XB1 =	000020	1926#	1927	1959#	1960	1992#	1993	2025#	2026	2058#	2059	2091#	2092	2124#							
		2125	2157#	2158	2190#	2191	2222#	2223	2254#	2255	2286#	2436#	2439	2467#							
		2469	2497#	2499	2527#	2529	2557#	2559	2587#	2589	2617#	2619	2647#	2649							
		2677#	2679	2707#	2709	2737#	2739	2767#	2769	2797#	2799	2827#	2829	2857#							
		2859	2887#	2889	2917#	2919	2947#	2949	2977#	2979	3007#	3009	3037#	3039							

CROSS REFERENCE TABLE -- USER SYMBOLS

.CNVRT	015360	859	3940#			
.CONVR	015354	857	3938#			
.EOP	014250	3222	3683	3710#		
.ERRTA	017670	4028	4222#			
.HLT	015634	647	4008#			
.INSTE	015014	849	3841#			
.INSTG	015020	3840	3843#			
.INSTR	014676	847	3817#			
.INST1	014716	3821#	3832	3848		
.MEMCL	017220	863	4219#			
.MSG	014720	3819*	3822#	4125*	4126*	4127
.MSTCL	017222	861	4220#			
.PARAM	015046	851	3855#			
.PFAIL	016164	645	938	4086#	4094	
.RES05	015322	855	3928#			
.SAV05	015262	853	3914#			
.SCOPE	014436	841	3761#			
.SCOP1	014550	843	3785#			
.START	001512	696	936#	948		
.TRPSR	015602	649	3997#			
.TRPTA	001314	839#	4002			
.TYPE	014570	845	3793#			

. ABS. 020056 000

ERRORS DETECTED: 0

DSKZ:CZDQAD,DSKZ:CZDQAD,SEQ=DSKZ:CZDQXX,MAC,DSKZ:C7DQAD.P11
RUN-TIME: 17 21 1 SECONDS
RUN-TIME RATIO: 111/40=2.7
CORE USED: 20K (39 PAGES)

DOCUMENT PAGES: 94