

# DQ11

INTERRUPT TESTS  
CZDQCE0

AH-8612E-MC  
COPYRIGHT ©74-78  
FICHE 1 OF 1

JAN 1979  
**digital**  
MADE IN USA

Microfiche frame 1: Header information and data	Microfiche frame 2: Data	Microfiche frame 3: Data	Microfiche frame 4: Data	Microfiche frame 5: Data	Microfiche frame 6: Data
Microfiche frame 7: Data	Microfiche frame 8: Data	Microfiche frame 9: Data	Microfiche frame 10: Data	Microfiche frame 11: Data	Microfiche frame 12: Data
Microfiche frame 13: Data	Microfiche frame 14: Data	Microfiche frame 15: Data	Microfiche frame 16: Data	Microfiche frame 17: Data	Microfiche frame 18: Data
Microfiche frame 19: Data	Microfiche frame 20: Data	Microfiche frame 21: Data	Microfiche frame 22: Data	Microfiche frame 23: Data	Microfiche frame 24: Data
Microfiche frame 25: Data	Microfiche frame 26: Data	Microfiche frame 27: Data	Microfiche frame 28: Data	Microfiche frame 29: Data	Microfiche frame 30: Data
Microfiche frame 31: Data	Microfiche frame 32: Data	Microfiche frame 33: Data	Microfiche frame 34: Data	Microfiche frame 35: Data	Microfiche frame 36: Data
Microfiche frame 37: Data	Microfiche frame 38: Data	Microfiche frame 39: Data	Microfiche frame 40: Data	Microfiche frame 41: Data	Microfiche frame 42: Data
Microfiche frame 43: Data	Microfiche frame 44: Data	Microfiche frame 45: Data	Microfiche frame 46: Data	Microfiche frame 47: Data	Microfiche frame 48: Data
Microfiche frame 49: Data	Microfiche frame 50: Data	Microfiche frame 51: Data	Microfiche frame 52: Data	Microfiche frame 53: Data	Microfiche frame 54: Data
Microfiche frame 55: Data	Microfiche frame 56: Data	Microfiche frame 57: Data	Microfiche frame 58: Data	Microfiche frame 59: Data	Microfiche frame 60: Data
Microfiche frame 61: Data	Microfiche frame 62: Data	Microfiche frame 63: Data	Microfiche frame 64: Data	Microfiche frame 65: Data	Microfiche frame 66: Data
Microfiche frame 67: Data	Microfiche frame 68: Data	Microfiche frame 69: Data	Microfiche frame 70: Data	Microfiche frame 71: Data	Microfiche frame 72: Data
Microfiche frame 73: Data	Microfiche frame 74: Data	Microfiche frame 75: Data	Microfiche frame 76: Data	Microfiche frame 77: Data	Microfiche frame 78: Data
Microfiche frame 79: Data	Microfiche frame 80: Data	Microfiche frame 81: Data	Microfiche frame 82: Data	Microfiche frame 83: Data	Microfiche frame 84: Data

IDENTIFICATION

PRODUCT CODE: AC-8610E-MC  
PRODUCT NAME: CZDQCE0 DG11 INTR  
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 4K MEMORY)-WITH  
OR WITHOUT A HARDWARE SWITCH REGISTER (LOC. 177570)  
ASR 33 (OR EQUIVALENT)

DQ11

SYNC MODEM (ONLY REQUIRED FOR ONLINE TEST)

2.2

STORAGE

PROGRAM WILL LOAD AND RUN  
IN 4K OF MEMORY.

LOCATION 1400 THRU 1600 ARE ESPECIALLY TO  
BE NOTED AND TO BE UNTOUCHED BY OPERATOR  
AFTER DQ11 TRIAL PROGRAM HAS BEEN EXECUTED.  
OR AFTER THE "AUTO SIZING" HAS BEEN DONE.

3.

LOADING PROCEEDURE

3.1

METHOD

ALL PROGRAMS ARE IN ABSOLUTE FORMAT AND

ARE LOADED USING THE ABSOLUTE LOADER.

ABSOLUTE LOADER STARTING ADDRESS \*500

MEMORY \*  
 SIZE

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

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

3.1.2 THEN START

4. STARTING PROCEDURE

A. LOAD LOC. 200

B. SET SWR TO ZERO FOR 'AUTO SIZING' OR LEAVE  
 LEAVE SWR BIT 7=1 TO USE EXISTING PARAMETERS SET UP  
 BY DQ11 TRIAL PROGRAM OR A PREVIOUSLY RUN DQ11 DIAGNOSTIC  
 THAT USED THE 'AUTO SIZING'.  
 \*\*\*\*REFER TO SECTION 4.1 FOR SOFTWARE SWITCH REGISTER OPERATION  
 AND OPTIONS.\*\*\*\*

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

C. THEN START

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

'MAP OF DQ11 STATUS'

1400	160010
1402	152300
1404	160020
1406	150310

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

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

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

4. 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 DQ11'S 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 CAN'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=C 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-8610E-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 TST\* (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.  
EXAMPLF:  
(RUN) 1304/0000000001000000  
MEANS THAT DQ11 NO.05 IS THE DQ11 NOW  
RUNNING.

DQCR00-DQCR17  
DQST00-DQST17  
(1400)-(1476)

THESE LOCATIONS CONTAIN THE INFORMATION  
NEEDED TO TEST UP TO 16 (DECIMAL) DQ11S  
SEQUENTIALLY. THEY CONTAIN THE CSR, VECTOR  
AND STATUS CONCERNING THE CONFIGURATION  
OF EACH DQ11.

DQACTV (1500) EACH BIT SET IN THIS LOCATION INDICATES  
THAT THE ASSOCIATED DQ11 WILL BE TESTED  
IN TURN.  
EXAMPLE:  
(DQACTV) 1500/0000000000011111  
MEANS THAT DQ11 NO. 00,01,02,03,04  
WILL BE TESTED.

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

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

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

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

BIT 00-08 VECTOR 'A' OF DEVICE

8.5 \*\*\* METHOD OF AUTO SIZING \*\*\*

8.5.1 FINDING THE CONTROL STATUS REGISTER.

WHEN LOOKING FOR THE CSR IT IS NECESSARY TO TAKE CARE THAT WHEN A CSR IS FOUND THAT IT IS INDEED A DQ11. THAT IS THE METHOD OF MY MADNESS FOR THIS ROUTINE. AN ATTEMPT TO CLEAR THE MISC. REGISTER IS TRIED IF A TIME-OUT TRAP OCCURS POINTERS ARE UPDATED AND ATTEMPTED AGAIN. IF NO TIME-OUT; THE RECEIVER 'ACTIVE BI'' (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 EXISTS

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

8.5.9 FINDING THE VECTOR.

THE PROGRAM SETS 'PRIMARY DONE', 'SECONDARY DONE', AND 'INTERUPT ENABLE' AND LOOKS FOR AN INTERUPT. IF IT INTERUPTS IT IS PICKED UP AND STORED AWAY. IF NO INTERUPT 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          177320          NON.EX=177320
523                               .ENABLE AMA
524
525                               ;CZDQCEO/<377>/DQ11 INTERRUPT AND NPR LOGIC TEST
526                               ;COPYRIGHT 1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
527
528                               ;REVISED 16-DEC-76 BY R. BLACK
529                               ;
530                               ; A)SUPPORTS SOFTWARE SWITCH REGISTER
531                               ; B)SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER
532                               ;   BY <^G>.
533                               ; STARTING PROCEDURE
534                               ; LOAD PROGRAM
535                               ; LOAD ADDRESS 000200
536                               ; PRESS START
537                               ; PROGRAM WILL TYPE 'CZDQCEO/<377>/DQ11 INTERRUPT AND NPR LOGIC TEST'
538                               ; PROGRAM WILL TYPE 'R' TO INDICATE THAT TESTING HAS STARTED
539                               ; AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
540                               ; AND THEN RESUME TESTING
541
542                               ;SWITCH REGISTER OPTIONS
543
544          100000          SW15=100000          ; 1,HALT ON ERROR
545          040000          SW14=40000          ; -1,LOOP ON CURRENT TEST
546          020000          SW13=20000          ; 1,INHIBIT ERROR TIMEOUT
547          010000          SW12=10000          ; -1,DELETE TIMEOUT/BELL ON ERROR.
548          004000          SW11=4000          ; =1,INHIBIT ITERATIONS
549          002000          SW10=2000          ; =1,ESCAPE TO NEXT TEST ON ERROR
550          001000          SW09=1000          ; 1,LOOP WITH CURRENT DATA
551          000400          SW08=400          ; -1,LOOP ON ERROR
552          000100          SW06=100
553          000040          SW05=40
554          000020          SW04=20
555          000010          SW03=10
556          000004          SW02=4          ;LOCK ON TEST SELECT
557          000002          SW01=2          ;RESTART PROGRAM AT SELECTED TEST
558          000001          SW00=1          ;RESELECT DQ11 DESIRED ACTIVE
559                               ;NOTE: THIS MUST NOT EXCEED ORIGINAL COUNT
```

560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615

000000  
000001  
000002  
000003  
000004  
000005  
000006  
000007  
  
  
  
177570  
177570  
177776  
001200  
  
005746  
005726  
010046  
012600  
024646  
022626  
  
  
  
100000  
040000  
020000  
010000  
004000  
002000  
001000  
000400  
000200  
000100  
000040  
000020  
000010  
000004  
000002  
000001  
  
  
002000  
004000  
010000  
020000  
040000

;REGISTER DEFINITIONS

R0=%0 ;GENERAL REGISTER  
R1=%1 ;GENERAL REGISTER  
R2=%2 ;GENERAL REGISTER  
R3=%3 ;GENERAL REGISTER  
R4=%4 ;GENERAL REGISTER  
R5=%5 ;GENERAL REGISTER  
SP=%6 ;PROCESSOR STACK POINTER  
PC=%7 ;PROGRAM COUNTER

;LOCATION EQUIVALENCIES

DSWR= 177570 ;HARDWARE SWITCH REGISTER LOC.  
DLIGHTS=177570 ;HARDWARE DISPLAY REGISTER LOC.  
PS=177776 ;PROCESSOR STATUS WORD  
STACK=1200 ;START OF PROCESSOR STACK

;INSTRUCTION DEFINITIONS

PUSH1SP=5746 ;DECREMENT PROCESSOR STACK 1 WORD  
POP1SP=5726 ;INCREMENT PROCESSOR STACK 1 WORD  
PUSHR0=10046 ;SAVE R0 ON STACK  
POPR0=12600 ;RESTORE R0 FROM STACK  
PUSH2SP=24646 ;DECREMENT STACK TWICE  
POP2SP=22626 ;INCREMENT STACK TWICE  
.EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL

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

;DQ11 OPTIONAL DEFINITIONS

ABBIT=2000  
ACTBIT=4000  
BABIT=10000  
BBBIT 20000  
JUMBIT=40000

GENERAL DEFINATIONS AND EQUIVALENCIES

616 001000 ODDBIT=1000  
617 100000 SYNBIT=100000  
618  
619

;DQ11 SECONDARY REGISTER DEFINATIONS

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

TRAPCATCHER FOR UNEXPECTED INTERRUPTS

```

641                ;TRAPCATCAEP FOR ILLEGAL INTERRUPTS
642                000000          .-0
643                ;STANDARD INTERRUPT VECTORS
644
645                .=-24
646 000024 014564       .PFAIL          ;POWER FAIL HANDLER
647 000026 000340       340           ;SERVICE AT LEVEL 7
648 000030 014234       .HLT           ;ERROR HANDLER
649 000032 000340       340           ;SERVICE AT LEVEL 7
650 000034 014202       .TRPSRV        ;GENERAL HANDLER DISPATCH SERVICE
651 000036 000340       340           ;SERVICE AT LEVEL 7
652                000046          . 46
653 000046 012762       LOGICAL        ;ACT HOOKS
654                000052          .-52
655 000052 000000       .WORD 0
656                ;THIS ROUTINE TRIES TO FORCE THE RECEIVER TO INTERRUPT
657                ;TO ITS VECTOR WHERE IT WILL PICK UP THE STATUS LOCATION
658                ;FOR ITS NEW PC; AND PICK UP AN IOT INSTRUCTION FOR ITS
659                ;NEW PS. WHEN THE NEW PC IS FETCHED AN IOT INSTRUCTION IS
660                ;EXECUTED, TRAPPING TO LOCATION 20 WHERE A ROUTINE IS EXECUTED
661                ;TO TAKE THE PC FROM THE STACK AND US IT AS THE VECTOR ADDRESS
662                000056          . 56
663
664                VECMAP:
665 000056 010120       1$:      MOV     R1,(R0)+          ;START FILLING THE VECTOR AREA
666 000060 012721 000004      MOV     #4,(R1)+          ;WITH .+2; IOT (4)
667 000064 022021          CMP     (R0)+,(R1)+          ;UPDATE THE POINTERS
668 000066 020127 001000      CMP     R1,#1000          ;IS ALL FLOATING VECTOR AREA DONE
669 000072 101771          BLOS    1$              ;BR IF NOT ALL DONE
670 000074 012737 000146 000020      MOV     #4$,@#20          ;SET FOR IOT TRAP BY DQ11
671 000102 013737 001500 001244      MOV     DQACTV,TEMP1     ;GET THE ACTIVE DQ11 S
672 000110 006037 001244          2$:      ROR     TEMP1             ;ARE YOU ACTIVE.. DQ11
673 000114 103023          BCC     5$              ;IF CARRY CLEAR.. NO MORE DQ11S
674 000116 005037 177776          CLR     PS               ;CLEAR PS
675 000122 005722          TST     (R2)+            ;PUT POINTER TO STATUS TABLE
676 000124 012772 000340 177776      MOV     #340,@-2(R2)     ;TRY AND SET PRI/SEC DONE AND IE
677 000132 105200          INCB   R0               ;DELAY.....
678 000134 001376          BNE    .-2              ;.....DELAY
679 000136 112712 000300          MOVB   #300,(R2)         ;NO INTERRUPT ASSUME 300 FIX IN TEST C
680 000142 005722          3$:      TST     (R2)+          ;UPDATE POINTERS
681 000144 000761          BR     2$              ;GO DO IT AGAIN
682 000146 051612          4$:      BIS     (SP),(R2)     ;ENTERD BY IOT TRAP BY DQ11
683 000150 042712 000007          BIC    #7,(R2)          ;CLEAR UNWANTED BITS
684 000154 022626          CMP     (SP)+,(SP)+      ;POP IOT JUNK OFF STACK
685 000156 012716 000142          MOV     #3$,(SP)        ;SET RETURN PC ON STACK
686 000162 000002          RTI                    ;GC HOME
687 000164 000207          5$:      RTS     PC         ;ALL SIZING IS DONE
688
689                ;****SOFTWARE SWITCH REGISTER****
690                .-174
691 000174 000000       DISPREG:        0          ;SOFTWARE DISPLAY REGISTER
692 000176 000000       SWREG:           0          ;SOFTWARE SWITCH REGISTER
693
694                ;PROGRAM START
695
696                .-200
    
```





809	001246	000000	TEMP2: 0	; TEMPORARY STORAGE
810	001250	000000	TEMP3: 0	; TEMPORARY STORAGE
811	001252	000000	TEMP4: 0	; TEMPORARY STORAGE
812	001254	000000	TEMP5: 0	; TEMPORARY STORAGE
813	001256	000000	SAVR0: 0	; R0 STORAGE
814	001260	000000	SAVR1: 0	; R1 STORAGE
815	001262	000000	SAVR2: 0	; R2 STORAGE
816	001264	000000	SAVR3: 0	; R3 STORAGE
817	001266	000000	SAVR4: 0	; R4 STORAGE
818	001270	000000	SAVR5: 0	; R5 STORAGE
819	001272	000000	SAVSP: 0	; STACK POINTER STORAGE
820	001274	000000	SAVPC: 0	; PROGRAM COUNTER STORAGE
821	001276	000000	SAVNUM: 0	
822	001300	000001	CREAM: .BLKW 1	
823	001302	000000	RUNFLG: 0	
824	001304	000000	RUN: 0	
825	001306	000000	RUNCNT: 0	

CZDQCE.P11 22-JUN-78 08:42 PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

826
827
828
829 001310 000
830 001311 000
831 001312 000
832 001313 000
833 000000

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

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

839 ;\*\*\*\*\*
840 ;\*\*\*\*\*
841 001314 .TRPTAB:
842 104400 SCOPE=TRAP+0 ;CALL TO SCOPE LOOP AND ITERATION HANDLER
843 001314 013036 .SCOPE
844 104401 SCOPE1=TRAP+1 ;CALL TO LOOP ON CURRENT DATA HANDLER
845 001316 013150 .SCOPE1
846 104402 TYPE=TRAP+2 ;CALL TO TELETYPE OUTPUT ROUTINE
847 001320 013170 .TYPE
848 104403 INSTR=TRAP+3 ;CALL TO ASCII STRING INPUT ROUTINE
849 001322 013276 .INSTR
850 104404 INSTER=TRAP+4 ;CALL TO INPUT ERROR HANDLER
851 001324 013414 .INSTER
852 104405 PARAM=TRAP+5 ;CALL TO NUMERICAL DATA INPUT ROUTINE
853 001326 013446 .PARAM
854 104406 SAV05=TRAP+6 ;CALL TO REGISTER SAVE ROUTINE
855 001330 013662 .SAV05
856 104407 RES05=TRAP+7 ;CALL TO REGISTER RESTORE ROUTINE
857 001332 013722 .RES05
858 104410 CONVRT=TRAP+10 ;CALL TO DATA OUTPUT ROUTINE
859 001334 013754 .CONVRT
860 104411 CNVRT=TRAP+11 ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.
861 001336 013760 .CNVRT
862 104412 MSTCLR=TRAP+12 ;CALL TO ISSUF MASTER CLEAR
863 001340 015620 .MSTCLR
864 104413 MEMCLR=TRAP+13 ;CALL TO CLEAR ALL SCRATCH PAD MEMORIES
865 001342 015636 .MEMCLR
866 104414 CKSWR=TRAP+14 ;CALL TO ALLOW SWREG TO BE LOADED FROM TTY
867 001344 014662 .CKSWR
868 104415 CNTLU=TRAP+15 ;CALL TO ALLOW LOADING OF SWREG FROM TTY
869 001346 014736 .CNTLU

871 ;\*\*\*\*\*
872 ;\*\*\*\*\*

873 ;DQ11 VECTOR AND REGISTER INDIRECT POINTERS
874
875
876 001350 000000 DQRVEC: 0 ;POINTER TO DQ11 RECEIVER INTERRUPT VECTOR
877 001352 000000 DQRLVL: 0 ;POINTER TO DQ11 RECEIVER INTERRUPT SERVICE PS
878 001354 000000 DQIVEC: 0 ;POINTER TO DQ11 TRANSMITTER INTERRUPT VECTOR
879 001356 000000 DQILVL: 0 ;POINTER TO DQ11 TRANSMITTER INTERRUPT SERVICE PS
880 001360 000000 DGRCSR: 0 ;POINTER TO DQ11 RECEIVER CONTROL REGISTER
881 001362 000000 DGRCSH: 0 ;POINTER TO HIGH BYTE OF DQ11 RECEIVER CONTROL REGISTER



PROGRAM INITIALIZATION AND START UP.

```

938 001512 012737 000340 177776 .START: MOV #340,PS ;LOCK OUT INTERRUPTS
939 001520 012706 001200 MOV #STACK,SP ;SET UP STACK
940 001524 012737 014564 000024 MOV #.PFAIL,@#24 ;SET UP POWER FAIL VECTOR
941 001532 013737 001504 001276 MOV DQNUM,SAVNUM
942 001540 105037 001311 CLRB STFLG ;CLEAR START FLAG
943 001544 005037 001230 CLR PASCNT ;CLEAR PASS COUNT
944 001550 105037 001312 CLRB ERRFLG ;CLEAR ERROR FLAG
945 001554 005037 001302 CLR RUNFLG
946 001560 012737 001400 001300 MOV #1400,CREAM
947 001566 005037 001232 CLR ERRCNT ;CLEAR ERROR COUNT
948 001572 005037 001234 CLR LSTERR ;CLEAR LAST ERROR POINTER
949 001576 012737 000001 001226 MOV #1,TSTNO ;SET UP FOR TEST 1
950 001604 012737 001512 001214 MOV #.START,RETURN ;SET UP FOR POWER FAIL BEFORE
951 ;TESTING STARTS
952 001612 012737 177570 001200 MOV #DSWR,SWR ;MOV HARDWARE SWR TO SWR
953 001620 012737 177570 001202 MOV #DLIGHTS,LIGHTS ;MOV DISPLAY LIGHTS TO LIGHTS
954 001626 013746 000006 MOV @#6,-(SP) ;SAVE VECTORS
955 001632 013746 000004 MOV @#4,-(SP)
956 001636 012737 001656 000004 MOV #64$,@#4 ;SET UP FOR TIMEOUT
957 001644 022777 177777 177326 CMP #-1,@SWR ;REFERENCE HARDWARE SWITCH REGISTER
958 001652 001402 BEQ 65$
959 001654 000407 BR 66$
960 001656 022626 64$: CMP (SP)+,(SP)+ ;ADJUST STACK
961 001660 012737 000176 001200 65$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
962 001666 012737 000174 001202 MOV #DISPREG,LIGHTS ;POINT TO SOFT DISPLAY REG
963 001674 012637 000004 66$: MOV (SP)+,@#4 ;RESTORE VECTORS
964 001700 012637 000006 MOV (SP)+,@#6
965 001704 005737 000042 TST @#42 ;UNDER MONITOR
966 001710 001014 BNE 67$
967 ;*****THE NEXT 4 LINES OF CODE MOVED TO SOLVE PR#2757 (JUNE 78)*****
968 001712 105737 001310 TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED?
969 001716 001035 BNE 12$ ;IF YES, BR
970 001720 104402 001000 TYPE ,MTITLE ;TYPE TITLE MESSAGE
971 001724 105137 001310 COMB INIFLG ;IF NOT SET FLAG AND INIT
972 001730 022737 000176 001200 CMP #SWREG,SWR ;IS SWREG USED
973 001736 001001 BNE 67$
974 001740 104415 CNTLU
975 001742 105777 177232 67$: TSTB @SWR
976 001746 100402 BMI .+6
977 001750 004737 000220 JSR PC,CSRMAP
978 001754 104402 015412 TYPE ,XHEAD
979 001760 012737 001400 001244 MOV #1400,TEMP1
980 001766 017737 177252 001246 MOV @TEMP1,TEMP2
981 001774 001406 BEQ .+16
982 001776 104410 CONVRT
983 002000 015440 XSTATQ
984 002002 062737 000002 001244 ADD #2,TEMP1
985 002010 000766 BR .-22
986 002012 032777 000001 177160 12$: BIT #SW00,@SWR
987 002020 001424 BEQ 1$
988 002022 104402 TYPE
989 002024 015333 MNEW
990 002026 005000 CLR R0
991 002030 000000 HALT
992 002032 104414 CKSWR
993 002034 027737 177140 001502 CMP @SWR,SAVACT

```

```

994 002042 101404          BLOS 11$
995 002044 104402          TYPE
996 002046 015174          MERR3
997 002050 000000          HALT
998 002052 000776          BR
999 002054 017737 177120 001500 11$: MOV @SWR,DQACTV
1000 002062 013700 001500    MOV DQACTV,R0
1001 002066 000000          HALT
1002 002070 104414          CKSWR
1003 002072 012700 000300    1$: MOV #300,R0
1004 002076 012701 000302    MOV #302,R1
1005 002102 010120          2$: MOV R1,(R0)+
1006 002104 005021          CLR (R1)+
1007 002106 022021          CMP (R0)+,(R1)+
1008 002110 022700 001000    CMP #1000,R0
1009 002114 001372          BNE 2$
1010
1011                          ;TEST START AND RESTART
1012
1013 002116 012737 000340 177776 .BEGIN: MOV #340,PS             ;LOCK OUT INTERRUPTS
1014 002124 012706 001200          MOV #STACK,SP         ;SET UP STACK
1015 002130 005737 000042          TST @#42             ;IS PROGRAM UNDER MONITOR CONTROL
1016 002134 001040          BNE 3$
1017 002136 104414          CKSWR                ;CHECK FOR <^G>
1018 002140 032777 000004 177032  BIT #BIT2,@SWR       ;CHECK FOR LOCK ON TEST
1019 002146 001411          BEQ 1$
1020 002150 104402 015232          TYPE ,MLOCK
1021 002154 012737 000240 013046    MOV #NOP,TTST
1022 002162 012737 000240 013050    MOV #NOP,TTST+2
1023 002170 000406          BR 2$                ;SET UP TO LOCK
1024 002172 013737 013144 013046 1$: MOV BRW,TTST
1025 002200 013737 013146 013050    MOV BRX,TTST+2       ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1026 002206 032777 000002 176764 2$: BIT #SW01,@SWR     ;IF SW01 1, GET STARTING PC
1027 002214 001410          BEQ 3$
1028 002216 104403          INSTR
1029 002220 015220          MTSTPC
1030 002222 104405          PARAM
1031 002224 002254          TST1
1032 002226 012442          TLAST
1033 002230 001214          #RETURN
1034 002232          001 .BYTE 1
1035 002233          001 .BYTE 1
1036 002234 000403          BR 4$
1037 002236 012737 002254 001214 3$: MOV #TST1,RETURN     ;START AT TEST 1
1038 002244 104402 015122          4$: TYPE #R           ;TYPE R
1039 002250 000177 176740          JMP @RETURN         ;START TESTING
1040
1041                          ; TEST 1
1042                          ;*****
1042 002254 012737 000001 001226 TST1: MOV #1,TSTNO
1043 002262 012737 002644 001214    MOV #TST2,RETURN
1044 002270 012737 002644 001216    MOV #TST2,NEXT
1045 002276 105737 001302          TSTB RUNFLG         ;IS THIS MY FIRST TIME HERE?
1046 002302 001010          BNE 1$              ;BR IF FLAG IS SET
1047 002304 012737 000001 001304    MOV #BIT0,RUN        ;SET RUN POINTER.
1048 002312 012737 000020 001306    MOV #16,RUNCNT      ;SET FOR MAX OF 16 DQ11'S PER SYSTEM
1049 002320 105137 001302          COMB RUNF_G        ;SET RUN FLAG

```

```

1050 002324 033737 001304 001500 1$: BIT RUN,DQACTV ;FIND AN ACTIVE DQ11 TO TEST.
1051 002332 001032 BNE 3$ ;BR IF I FOUND ONE TO TEST.
1052 002334 005737 001500 TST DQACTV ;FIND OUT IF THERE ARE NO DQ11 ACTIVE.
1053 002340 001423 BEQ 2$ ;BR TO FATAL ERROR. WHY AM I HERE IF NO ACTIVE DQ11'S???
1054 002342 000257 CCC ;CLEAR ALL THE CONDITION CODES OF CPU
1055 002344 006137 001304 ROL RUN ;UPDATE RUN POINTER
1056 002350 062737 000004 001300 ADD #4,CREAM ;UPDATE ADDRESS POINTER.
1057 002356 005337 001306 DEC RUNCNT ;DEC NUMBER OF TIMES I LOOKED AT ACTIVE.
1058 002362 001360 BNE 1$ ;BR AND KEEP LOOKING.
1059 002364 012737 000020 001306 MOV #16,RUNCNT ;START RESTORING MY POINTERS.
1060 002372 012737 001400 001300 MOV #1400,CREAM ;RESTORE ADDRESS POINTER
1061 002400 012737 000001 001304 MOV #1,RUN ;RESTORE RUN POINTER.
1062 002406 000746 BR 1$ ;KEEP ON TESTING.
1063 002410 104402 2$: TYPE ;ALLERT OPERATOR OF FATAL ERROR
1064 002412 015125 MERR2 ;NO DQ11 ACTIVE. WHY AM I HERE???
1065 002414 000000 HALT ;YOU MUST RELOAD DQ11 DIAGNOSTIC..
1066 002416 000776 BR -2 ;STICK HERE ON CONT.
1067 002420 000257 3$: CCC ;CLEAR CPU COND. CODES
1068 002422 006137 001304 ROL RUN ;UPDATE RUN. ACTIVE DQ11 FOUND.
1069 002426 017737 176646 001506 MOV @CREAM,DQCSR ;PLACE ADDRESS OF DQ11 AT DQCSR
1070 002434 062737 000002 001300 ADD #2,CREAM ;UPDATE ADDRESS POINTER
1071 002442 017737 176632 001510 MOV @CREAM,DQSTAT ;PLACE STATUS OF DQ11 AT DQSTAT
1072 002450 062737 000002 001300 ADD #2,CREAM ;UPDATE ADDRESS POINTER
1073 002456 013737 001506 001360 MOV DQCSR,DQRCR ;
1074 002464 013737 001510 001350 MOV DQSTAT,DQVECC ;
1075 002472 042737 177007 001350 BIC #177007,DQVECC ;
1076 002500 013737 001350 001352 MOV DQVECC,DQRLVL ;GENERATE ADDRESS OF RECEIVER INTERRUPT SERVICE PS
1077 002506 062737 000002 001352 ADD #2,DQRLVL ;
1078 002514 013737 001352 001354 MOV DQRLVL,DQTVCC ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT VECTOR
1079 002522 062737 000002 001354 ADD #2,DQTVCC ;
1080 002530 013737 001354 001356 MOV DQTVCC,DQTLVL ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT SERVICE PS
1081 002536 062737 000002 001356 ADD #2,DQTLVL ;
1082 002544 013737 001360 001362 MOV DQRCR,DQRCSH ;
1083 002552 005237 001362 INC DQRCSH ;GENERATE ADDRESS OF HIGH BYTE
1084 002556 013737 001360 001364 MOV DQRCSR,DQTCSR ;GENERATE ADDRESS OF TRANSMITTER CONTROL REGISTER
1085 002564 062737 000002 001364 ADD #2,DQTCSR ;
1086 002572 013737 001364 001366 MOV DQTCSR,DQERR ;GENERATE ADDRESS OF ERROR REGISTER
1087 002600 062737 000002 001366 ADD #2,DQERR ;
1088 002606 013737 001366 001370 MOV DQERR,DQREG ;GENERATE ADDRESS OF HIGH BYTE OF ERROR REGISTER
1089 002614 005237 001370 INC DQREG ;
1090 002620 013737 001370 001372 MOV DQREG,DQSEC ;GENERATE ADDRESS OF SECONDARY REGISTER
1091 002626 005237 001372 INC DQSEC ;
1092 002632 013737 001372 001374 MOV DQSEC,DQSECH ;GENERATE ADDRESS OF HIGH BYTE
1093 002640 005237 001374 INC DQSECH ;
    
```

1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
: STEP MODE VERIFICATION AND CLOCK LOSS TEST  
: SET STEP MODE  
: SET RECEIVER GO  
: SET TRANSMITTER GO  
: EXPECTED RESULTS (AFTER DELAY)  
: TRANSMITTER CLOCK LOSS = 1  
: RECEIVER CLOCK LOSS = 1  
: \*\*\*\*\*NOTE: AS THE "CLOCK UP" OCCURS  
: AN "NPR" SHOULD BE EXECUTED.  
: THEREFORE IF THE DQ11 IS GOING TO "HANG"  
: THE BUS DUE TO NPR'S THIS IS THE  
: FIRST TEST IT WILL HAPPEN IN!!\*\*\*\*\*

: TEST 2

```
*****
1110 002644 012737 000002 001226 ST2: MOV #2,TSTNO
1111 002652 012737 000010 001222 MOV #10,ICOUNT
1112 002660 012737 003014 001216 MOV #TST3,NEXT
1113 002666 012737 000340 177776 MOV #340,PS ;LOCK OUT INTERRUPTS
1114 002674 104413 MEMCLR ;CLEAR MEMORY
1115 002676 104412 MSTCLR ;INIT DQ11
1116 002700 005037 001244 CLR TEMP1 ;ZERO DELAY COUNTER
1117 002704 012737 000010 001246 MOV #10,TEMP2 ;DELAY 8 X 65535 TIMES
1118 002712 112777 000012 176450 MOVB #12,@DQREG ;SELECT MISC REG
1119 002720 012777 000002 176444 MOV #BIT1,@DQSEC ;SET AUTO STEP
1120 002726 012777 010001 176424 MOV #BIT12+BIT0,@DQRCSR ;SET RX GO!!
1121 002734 012777 000001 176422 MOV #BIT0,@DQTCSR ;SET TX GO!!
1122 002742 005277 176424 INC @DQSEC ;CLOCK UP!!
1123 002746 005377 176420 DEC @DQSEC ;CLOCK DN!!
1124 002752 005237 001244 1$: INC TEMP1 ;DO THE DELAY....
1125 002756 001375 BNE 1$ ; DELAY.....
1126 002760 042777 010000 176372 BIC #BIT12,@DQRCSR ;CLEAR RX ACTIVE
1127 002766 005337 001246 DEC TEMP2 ;DELAY.....
1128 002772 001367 BNE 1$ ;DELAY.....
1129 002774 012705 000003 MOV #3,R5 ;SET FOR EXPECTED
1130 003000 117704 176362 MOVB @DQERR,R4 ;READ THE DQERR REGISTER (SEL4)
1131 003004 020504 CMP R5,R4 ;CLOCK LOSS WORKING??
1132 003006 001401 BEQ 2$ ;BR IF YES.
1133 003010 104013 HLT 13 ;TX AND RX CLOCK LOSS ERROR
1134 003012 104400 2$: SCOPE ;SCOPE THIS TEST.
1135 ;TEST LOOP VERIFICATION
1136 ;SET STEP MODE AND TEST LOOP
1137 ;SET RECEIVER GO
1138 ;SET TRANSMITTER GO
1139 ;EXPECTED RESULTS (AFTER DELAY)
1140 ; TRANSMITTER CLOCK LOSS=0
1141 ; RECEIVER CLOCK LOSS=0
```

: TEST 3

```
*****
1145 003014 012737 000003 001226 TST3: MOV #3,TSTNO
1146 003022 012737 000010 001222 MOV #10,ICOUNT
1147 003030 012737 003142 001216 MOV #TST4,NEXT
1148 003036 104413 MEMCLR
1149 003040 012737 000340 177776 MOV #340,PS ;SET PS 7 LOCK OUT INTERRUPTS
```

```

1150 003046 104412                               MSTCLR          :INIT DQ11
1151 003050 005037 001244                         CLR          TEMP1  :SET DELAY COUNTER TO 0
1152 003054 012737 000010 001246                 MOV          #10,TEMP2 :SET FOR 8 X 65535 TIME DELAY
1153 003062 112777 000012 176300                 MOVB         #12,@DQREG :SELECT MISC REGISTER
1154 003070 012777 000012 176274                 MOV          #BIT1+BIT3,@DQSEC :SET TESTLOOP AND AUTO/STEP
1155 003076 012777 000001 176254                 MOV          #BIT0,@DQRCSR :SET RX GO.
1156 003104 012777 000001 176252                 MOV          #BIT0,@DQTCR :SET TX GO.
1157 003112 005237 001244           1$:          INC          TEMP1  :D
1158 003116 001375                                     BNE          1$      :E
1159 003120 005337 001246                                     DEC          TEMP2  :L
1160 003124 001372                                     BNE          1$      :A
1161                                          :Y
1162 003126 005005                               CLR          R5      :SET EXPECTED
1163 003130 117704 176232                 MOVB         @DQERR,R4 :GET ACTUAL
1164 003134 001401                               BEQ          2$      :BR IF LOW BYTE OF DQERR IS 0
1165 003136 104013                               HLT          13      :DQ11 ERROR REG NOT 0
1166 003140 104400           :$:          SCOPE   :SCOPE THIS TEST.
    
```





```
1279 003524 000403                BR      3$
1280 003526 104003                1$:    HLT      3          ;UNEXPECTED RECEIVER INTERRUPT
1281 003530 000401                BR      3$
1282 003532 104002                2$:    HLT      2          ;UNEXPECTED TRANSMITTER INTERRUPT
1283 003534 012706 001200            3$:    MOV      #STACK,SP   ;RESTORE STACK
1284 003540 004737 016074            JSR     PC,RECAT        ;RESTORE TRAPCATCHER
1285 003544 104400                4$:    SCOPE            ;CHECK FOR ITERATIONS, LOOP
1286
1287                ;INDIVIDUAL INTERRUPT FLAG TESTS
1288                ;SET SELECTED INTERRUPT FLAG
1289                ;VERIFY THAT NO INTERRUPT OCCURS
1290
1291
1292                ;INTERRUPT LOGIC TEST
1293                ;SET RECEIVE DONE S INTERRUPT FLAG
1294                ;VERIFY THAT NO INTERRUPT OCCURS
1295
1296                ; TEST 11
1297                ;*****
1298 003546 012737 000011 001226    TST11: MOV      #11,TSTNO
1299 003554 012737 003632 001216          MOV      #TST12,NEXT
1300 003562 104412                MSTCLR           ;CLEAR INTERFACE
1301 003564 004737 016042            JSR     PC,SETV     ;SET UP INTERRUPT VECTORS
1302 003570 003612                1$          ;RECEIVER WILL INTERRUPT TO 1$
1303 003572 003616                2$          ;TRANSMITTER WILL INTERRUPT TO 2$
1304 003574 052777 000100 17555E        BIS     #BIT6,@DQRCSR ;SET RECEIVE DONE S INTERRUPT FLAG
1305 003602 005037 177776            CLR     PS          ;SET PROCESSOR PRIORITY TO 0
1306 003606 000240                NOP             ;WINDOW FOR INTERRUPTS
1307 003610 000403                BR      3$
1308 003612 104003                1$:    HLT      3          ;UNEXPECTED RECEIVER INTERRUPT
1309 003614 000401                BR      3$
1310 003616 104002                2$:    HLT      2          ;UNEXPECTED TRANSMITTER INTERRUPT
1311 003620 012706 001200            3$:    MOV      #STACK,SP   ;RESTORE STACK
1312 003624 004737 016074            JSR     PC,RECAT        ;RESTORE TRAPCATCHER
1313 003630 104400                4$:    SCOPE            ;CHECK FOR ITERATIONS, LOOP
1314
1315                ;INTERRUPT LOGIC TEST
1316                ;SET RECEIVE DONE P INTERRUPT FLAG
1317                ;VERIFY THAT NO INTERRUPT OCCURS
1318
1319                ; TEST 12
1320                ;*****
1321 003632 012737 000012 001226    TST12: MOV      #12,TSTNO
1322 003640 012737 003716 001216          MOV      #TST13,NEXT
1323 003646 104412                MSTCLR           ;CLEAR INTERFACE
1324 003650 004737 016042            JSR     PC,SETV     ;SET UP INTERRUPT VECTORS
1325 003654 003676                1$          ;RECEIVER WILL INTERRUPT TO 1$
1326 003656 003702                2$          ;TRANSMITTER WILL INTERRUPT TO 2$
1327 003660 052777 000200 175472        BIS     #BIT7,@DQRCSR ;SET RECEIVE DONE P INTERRUPT FLAG
1328 003666 005037 177776            CLR     PS          ;SET PROCESSOR PRIORITY TO 0
1329 003672 000240                NOP             ;WINDOW FOR INTERRUPTS
1330 003674 000403                BR      3$
1331 003676 104003                1$:    HLT      3          ;UNEXPECTED RECEIVER INTERRUPT
1332 003700 000401                BR      3$
1333 003702 104002                2$:    HLT      2          ;UNEXPECTED TRANSMITTER INTERRUPT
1334 003704 012706 001200            3$:    MOV      #STACK,SP   ;RESTORE STACK
```

























2007 006670 001401 BEQ .+4 ;BR IF YES  
2008 006672 104000 HLT 0 ;EITHER RX DID NOT INTERRUPT; OR MORE THAN ONCE  
2009 006674 104400 SCOPE ;SCOPE THE TEST  
2010 006676 005100 1\$: COM R0 ;CHECK INTERRUPT  
2011 006700 001401 BEQ .+4 ;BR IF FIRST TIME HERE  
2012 006702 104003 HLT 3 ;RX INTERRUPTED MORE THAN ONCE  
2013 006704 000002 RTI ;GO BACK AND DELAY  
2014 006706 104002 2\$: HLT 2 ;UNEXPECTED TX INTERRUPT  
2015 006710 000002 RTI ;RETURN

;TEST TO SEE IF THE  
;DQ11 TRANSMITTER WILL  
;INTERUPT AT PS LEVEL  
;OF 7 PRIORITY.  
;

; TEST 46

2024  
2025 \*\*\*\*\*  
2026 006712 012737 000046 001226 TST46: MOV #46,TSTNO  
2027 006720 012737 007014 001216 MOV #TST47,NEXT  
2028 006726 104412 MSTCLR ;INIT DQ11  
2029 006730 004737 016042 JSR PC,SETV ;SET VECTORS  
2030 006734 006766 1\$: ;RX INTERRUPTS TO 1\$  
2031 006736 006772 2\$: ;TX INTERRUPTS TO 2\$  
2032 006740 012700 177777 MOV #-1,R0 ;SET CHECKER  
2033 006744 012737 000340 177776 MOV #340,PS ;SET PRIORITY  
2034 006752 012777 000240 172404 MOV #240,@DQ1CSR ;SET PRI DONE AND IE  
2035 006760 000240 NOP ;  
2036 006762 000240 NOP ;  
2037 006764 000403 3\$: BR 3\$ ;CONTINUE TEST  
2038 006766 104003 1\$: HLT 3 ;UNEXPECTED RX INTERRUPT  
2039 006770 000002 RTI ;CONTINUE TEST  
2040 006772 005100 2\$: COM R0 ;CHECK INTERRUPT  
2041 006774 012706 001200 3\$: MOV #STACK,SP ;SET STACK POINTER  
2042 007000 005700 TST R0 ;CHECK INTERRUPT POINTER  
2043 007002 001001 BNE .+4 ;  
2044 007004 104002 HLT 2 ;  
2045 007006 005077 172352 CLR @DQ1CSR ;  
2046 007012 104400 SCOPE ;

;TEST TO SEE IF THE  
;DQ11 TRANSMITTER WILL  
;INTERUPT AT PS LEVEL  
;OF 6 PRIORITY.  
;

; TEST 47

2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054 \*\*\*\*\*  
2055 TST47: MOV #47,TSTNO  
2056 007014 012737 000047 001226 MOV #TST50,NEXT  
2057 007022 012737 007116 001216 MSTCLR ;INIT DQ11  
2058 007030 104412 JSR PC,SETV ;SET VECTORS  
2059 007032 004737 016042 1\$: ;RX INTERRUPTS TO 1\$  
2060 007036 007070 2\$: ;TX INTERRUPTS TO 2\$  
2061 007040 007074 MOV #-1,R0 ;SET CHECKER  
2062 007042 012700 177777













```
2396  
2397 :DONE (SECONDARY)=1, INTERRUPT ENEABLE=1,  
2398 010544 017704 170610 MOV @DQRCR,R4 :P/S=1  
2399 010550 042704 177400 BIC #177400,R4 :(R4)=ACTUAL DATA IN RECEIVER CONTROL REGISTER  
2400 010554 020504 CMP R5,R4 :CLEAR UNWANTED BITS  
2401 010556 001401 BEQ 6$ :ARE EXPECTED AND RECEIVED DATA THE SAME  
2402 010560 104004 HLT 4  
2403 010562 005005 6$: CLR R5 :RECEIVER STATUS ERROR  
2404 010564 013703 001366 MOV DQERR,R3 :(R5)=EXPECTED DATA IN ERROR REGISTER, 0  
2405 010570 117704 170572 MOV B @DQERR,R4 :ADDRESS OF ERROR REGISTER  
2406 010574 001401 BEQ 7$ :(R4) TUAL DATA IN ERROR REGISTER  
2407 010576 104006 HLT 6  
2408 010600 112777 000004 170562 7$: MOV B #4,@DQREG :ERROR FLAG(S) SET  
2409 010606 012702 000004 MOV #4,R2 :SELECT RECEIVER BUS ADDRESS (SECONDARY)  
2410 :ADDRESS OF RECEIVER BUS ADDRESS  
2411 010612 013703 001372 MOV DQSEC,R3 :SECONDARY REGISTER  
2412 010616 012705 017167 MOV #RBUF+1,R5 :ADDRESS OF SECONDARY REGISTER  
2413 : (R5)=EXPECTED DATA IN  
2414 :RECEIVER BUS ADDRESS (SECONDARY) REGISTER,  
2415 010622 017704 170544 MOV @DQSEC,R4 :RBUF+1  
2416 : (R4)=ACTUAL DATA IN RECEIVER  
2417 010626 020504 CMP R5,R4 :BUS ADDRESS REGISTER (SECONDARY)  
2418 010630 001401 BEQ 10$ :ARE EXPECTED AND RECEIVED DATA THE SAME  
2419 010632 104007 HLT 7 :BUS ADDRESS ERROR  
2420 010634 105277 170530 10$: INCB @DQREG  
2421 010640 005202 INC R2  
2422 010642 012705 000000 MOV #0,R5  
2423 010646 017704 170520 MOV @DQSEC,R4  
2424 010652 020504 CMP R5,R4  
2425 010654 001401 BEQ 11$  
2426 010656 104010 HLT 10 :CHARACTER COUNT ERROR  
2427 010660 012705 000000 11$: MOV #0,R5  
2428 010664 012703 017166 MOV #RBUF,R3  
2429 010670 013704 017166 MOV RBUF,R4  
2430 010674 020504 CMP R5,R4  
2431 010676 001401 BEQ QZX  
2432 010700 104011 HLT 11  
2433 010702 104400 QZX: SCOPE  
2434  
2435 :TRANSMITTER BASIC NPR LOGIC TEST (USING SECONDARY BUS ADDRESS AND CHARACTER COU  
2436 :EXPECTED RESULTS  
2437 : TRANSMITTER DONE INTERRUPT OCCURS  
2438 : TRANSMITTER DONE (SECONDARY) 1  
2439 : TRANSMITTER GO = 0  
2440 : TRANSMITTER P/S = 1  
2441 : NO ERROR FLAGS ARE SET  
2442 :  
2443 : TRANSMITTER BUS ADDRESS (SECONDARY) - TBUF+1  
2444 : TRANSMITTER CHARACTER COUNT (SECONDARY) = 0  
2445 : CONTENTS OF TRANSMITTER BUFFER - 177777  
2446 :  
2447 : TEST 56  
2448 :*****  
2449 010704 012737 000056 001226 TST56: MOV #56,TSTNO  
2450 010712 012737 010730 001214 MOV #0,RETURN  
2451 010720 012737 011330 001216 MOV #TST57,NEXT
```





















CZDQC MACY11 30A(1052) 07-JUL-78 08:20 PAGE 59  
 CZDQCE.P11 22-JUN-78 08:42

SEQ 0058

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

```

3012                                     ;SAVE PC OF TEST THAT FAILED AND R0-R5
3013
3014 013662 016637 000004 001274 .SAV05: MOV 4(SP),SAVPC
3015
3016                                     ;SAVE R0-R5
3017
3018 013670 010537 001270          SV05: MOV R5,SAVR5
3019 013674 010437 001266          MOV R4,SAVR4
3020 013700 010337 001264          MOV R3,SAVR3
3021 013704 010237 001262          MOV R2,SAVR2
3022 013710 010137 001260          MOV R1,SAVR1
3023 013714 010037 001256          MOV R0,SAVR0
3024 013720 000002                RTI
3025
3026                                     ;RESTORE R0-R5
3027
3028 013722 013700 001256          .RES05: MOV SAVR0,R0
3029 013726 013701 001260          MOV SAVR1,R1
3030 013732 013702 001262          MOV SAVR2,R2
3031 013736 013703 001264          MOV SAVR3,R3
3032 013742 013704 001266          MOV SAVR4,R4
3033 013746 013705 001270          MOV SAVR5,R5
3034 013752 000002                RTI
3035
3036                                     ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
3037
3038 013754 104402          .CONVR: TYPE
3039 013756 015040          MCRLF
3040 013760 010046          .CNVRT: MOV R0,-(SP)
3041 013752 010146          MOV R1,-(SP)
3042 013764 010346          MOV R3,-(SP)
3043 013766 010446          MOV R4,-(SP)
3044 013770 010546          MOV R5,-(SP)
3045 013772 017601 000012          MOV @12(SP),R1
3046 013776 013737 015514 001250          MOV TEMP,TEMP3
3047 014004 062766 000002 000012          ADD #2,12(SP)
3048 014012 012137 014174          MOV (R1)+,WRDCNT
3049 014016 112137 014176          1$: MOV (R1)+,CHRCNT
3050 014022 112137 014177          MOV (R1)+,SPACNT
3051 014026 013137 014200          MOV @ (R1)+,BINWRD
3052 014032 013704 014200          2$: MOV BINWRD,R4
3053 014036 113705 014176          MOVB CHRCNT,R5
3054 014042 012700 015514          MOV #TEMP,R0
3055 014046 010403          3$: MOV R4,R3
3056 014050 042703 177770          BIC #177770,R3
3057 014054 062703 000060          ADD #060,R3
3058 014060 110320          MOV R3,(R0)+
3059 014062 000241          CLC
3060 014064 006004          ROR R4
3061 014066 000241          CLC
3062 014070 006004          ROR R4
3063 014072 000241          CLC
3064 014074 006004          ROR R4
3065 014076 005305          DEC R5
3066 014100 001362          BNE 3$
3067 014102 012703 015556          MOV #MMDATA,R3

```

































