

DQ11

SEQUENCE REGISTER TESTS
MD-11-DZDQF-D

EP-DZDQF-D-DL-B
COPYRIGHT © 1977
FICHE 1 OF 1

APR 1977

MADE IN USA



TEST NAME	TEST NUMBER	TEST RESULT	TEST DATE	TEST TIME	TEST LOCATION
TEST 1	1	PASS	1977	10:00	MD-11
TEST 2	2	PASS	1977	10:05	MD-11
TEST 3	3	PASS	1977	10:10	MD-11
TEST 4	4	PASS	1977	10:15	MD-11
TEST 5	5	PASS	1977	10:20	MD-11
TEST 6	6	PASS	1977	10:25	MD-11
TEST 7	7	PASS	1977	10:30	MD-11
TEST 8	8	PASS	1977	10:35	MD-11
TEST 9	9	PASS	1977	10:40	MD-11
TEST 10	10	PASS	1977	10:45	MD-11
TEST 11	11	PASS	1977	10:50	MD-11
TEST 12	12	PASS	1977	10:55	MD-11
TEST 13	13	PASS	1977	11:00	MD-11
TEST 14	14	PASS	1977	11:05	MD-11
TEST 15	15	PASS	1977	11:10	MD-11

15

B01

EOF10ZDQEDSEQ

00010000

770325

PDP10 411

08HDR10ZDQFDSEQ

00010000

770325

CO1

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 2
DZDQFD.P11 21-DEC-76 16:34

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDQF-D-D
PRODUCT NAME: SEQUENCE REGISTER TESTS
DATE: MARCH 1977
MAINTAINER: DIAGNOSTIC GROUP

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

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

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

COPYRIGHT (C) 1974, 1977 BY DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

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

THIS TEST CHECKS OUT THE PROTOCOL AND SEQUENCE REGISTER FOR THE DQ11 BB OPTION. THE FIRST PART CHECKS THAT ALL CHARACTERS CAN BE DETECTED AS A SINGLE CHARACTER MATCH AND THEN DOUBLE CHARACTER MATCH IS TESTED. THE SEQUENCE REGISTER FUNCTIONS ARE THEN TESTED ON BOTH THE TRANSMITTER AND RECEIVER TOGETHER IN "FREE RUNNING" MODE. IF THE DATA SET OPTION IS INSTALLED AND THE CABLE TURN AROUND IS INSTALLED; THE DATA IS RUN THROUGH THE CABLE.

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

THE SEVEN DIAGNOSTICS ARE:

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

THERE IS ALSO AN ONLINE TEST TO BE DISCUSSED LATER.

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

AND A PARAMETER INPUT PROGRAM IS AVAILABLE

1. DZDQG [REV] DQ11 TRIAL PROGRAM (PARAMETER INPUT)

2. REQUIREMENTS

2.1 EQUIPMENT

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

2.2 STORAGE

PROGRAM WILL LOAD AND RUN
IN 4K OF MEMORY.

DZDGF MACY11 27(1006) 22-DEC-76 11:36 PAGE 4
 DZDGF.D.P11 21-DEC-76 16:34

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!

F01

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 5
DZDQFD.P11 21-DEC-76 16:34

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 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 4 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 DZDGF-D 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.

DQCROO-DQCR17
 DQSTOO-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

DQCSR (1506) WILL BE TESTED.
 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?

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 APPROPRIATE 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

MO1

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 12
DZDQFD.P11 21-DEC-76 16:34

INTRODUCTION TO DQ11 DIAGNOSTIC

;MAINDEC-11-DZDQF-D/(377)/SEQUENCE REG TEST
;COPYRIGHT 1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

;REVISED 16-DEC-76 BY R. BLACK
;A)SUPPORTS SOFTWARE SWITCH REGISTER
;B)SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER
;BY (↑G).

;STARTING PROCEDURE
;LOAD PROGRAM
;LOAD ADDRESS 000200
;PRESS START
;PROGRAM WILL TYPE "MAINDEC-11-DZDQF-D/(377)/SEQUENCE REG TEST"
;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
;AND THEN RESUME TESTING

;SWITCH REGISTER OPTIONS

552 100000
553 040000
554 020000
555 010000
556 004000
557 002000
558 001000
559 000400
560 000100
561 000040
562 000020
563 000010
564 000004
565 000002
566 000001
567

SW15=100000 ;=1,HALT ON ERROR
SW14=40000 ;=1,LOOP ON CURRENT TEST
SW13=20000 ;=1,INHIBIT ERROR TYPEOUT
SW12=10000 ;=1,DELETE TYPEOUT/BELL ON ERROR.
SW11=4000 ;=1,INHIBIT ITERATIONS
SW10=2000 ;=1,ESCAPE TO NEXT TEST ON ERROR
SW09=1000 ;=1,LOOP WITH CURRENT DATA
SW08=400 ;=1,LOOP ON ERROR
SW06=100
SW05=40
SW04=20
SW03=10
SW02=4
SW01=2
SW00=1

;LOCK ON TEST SELECT
;RESTART PROGRAM AT SELECTED TEST
;RESELECT DQ11 DESIRED ACTIVE
;NOTE: THIS MUST NOT EXCEED ORIGINAL COUNT

NO1

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 13
DZDQFD.P11 21-DEC-76 16:34

GENERAL DEFINITIONS AND EQUIVALENCIES

```
568  
569  
570  
571  
572      000000      RO=%0      ; GENERAL REGISTER  
573      000001      R1=%1      ; GENERAL REGISTER  
574      000002      R2=%2      ; GENERAL REGISTER  
575      000003      R3=%3      ; GENERAL REGISTER  
576      000004      R4=%4      ; GENERAL REGISTER  
577      000005      R5=%5      ; GENERAL REGISTER  
578      000006      SP=%6      ; PROCESSOR STACK POINTER  
579      000007      PC=%7      ; PROGRAM COUNTER  
580  
581  
582  
583  
584      177570      DSWR= 177570 ; HARDWARE SWITCH REGISTER LOC.  
585      177570      DLIGHTS=177570 ; HARDWARE DISPLAY REGISTER LOC.  
586      177776      PS=177776 ; PROCESSOR STATUS WORD  
587      001200      STACK=1200 ; START OF PROCESSOR STACK  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599      100000      BIT15=100000  
600      040000      BIT14=40000  
601      020000      BIT13=20000  
602      010000      BIT12=10000  
603      004000      BIT11=4000  
604      002000      BIT10=2000  
605      001000      BIT9=1000  
606      000400      BIT8=400  
607      000200      BIT7=200  
608      000100      BIT6=100  
609      000040      BIT5=40  
610      000020      BIT4=20  
611      000010      BIT3=10  
612      000004      BIT2=4  
613      000002      BIT1=2  
614      000001      BIT0=1  
615  
616  
617  
618  
619  
620  
621  
622  
623
```

; REGISTER DEFINITIONS

; LOCATION EQUIVALENCIES

; INSTRUCTION DEFINITIONS

; DQ11 OPTIONAL DEFINITIONS

624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648

001000
100000

000000
000001
000002
000003
000004
000005
000006
000007

000010
000011
000012
000013
000014
000015
000016
000017

000BIT=1000
SYNBIT=100000

;DQ11 SECONDARY REGISTER DEFINATIONS

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

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

```

649                                     ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
650                                     .=6
651 000000 000000                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
652 000002 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
653 000004 000006                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
654 000006 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
655 000010 000012                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
656 000012 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
657 000014 000016                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
658 000016 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
659 000020 000022                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
660 000022 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
661 000024 000026                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
662 000026 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
663 000030 000032                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
664 000032 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
665 000034 000036                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
666 000036 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
667 000040 000042                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
668 000042 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
669 000044 000046                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
670 000046 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
671 000050 000052                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
672 000052 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
673 000054 000056                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
674 000056 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
675 000060 000062                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
676 000062 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
677 000064 000066                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
678 000066 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
679 000070 000072                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
680 000072 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
681 000074 000076                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
682 000076 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
683 000100 000102                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
684 000102 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
685 000104 000106                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
686 000106 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
687 000110 000112                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
688 000112 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
689 000114 000116                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
690 000116 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
691 000120 000122                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
692 000122 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
693 000124 000126                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
694 000126 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
695 000130 000132                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
696 000132 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
697 000134 000136                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
698 000136 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
699 000140 000142                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
700 000142 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
701 000144 000146                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
702 000146 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
703 000150 000152                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
704 000152 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
    
```

705	000154	000156	.+2	:UNEXPECTED TRAP TO THIS LOCATION
706	000156	000000	HALT	:EXAMINE STACK TO FIND CAUSE
707	000160	000162	.+2	:UNEXPECTED TRAP TO THIS LOCATION
708	000162	000000	HALT	:EXAMINE STACK TO FIND CAUSE
709	000164	000166	.+2	:UNEXPECTED TRAP TO THIS LOCATION
710	000166	000000	HALT	:EXAMINE STACK TO FIND CAUSE
711	000170	000172	.+2	:UNEXPECTED TRAP TO THIS LOCATION
712	000172	000000	HALT	:EXAMINE STACK TO FIND CAUSE
713	000174	000176	.+2	:UNEXPECTED TRAP TO THIS LOCATION
714	000176	000000	HALT	:EXAMINE STACK TO FIND CAUSE
715	000200	000202	.+2	:UNEXPECTED TRAP TO THIS LOCATION
716	000202	000000	HALT	:EXAMINE STACK TO FIND CAUSE
717	000204	000206	.+2	:UNEXPECTED TRAP TO THIS LOCATION
718	000206	000000	HALT	:EXAMINE STACK TO FIND CAUSE
719	000210	000212	.+2	:UNEXPECTED TRAP TO THIS LOCATION
720	000212	000000	HALT	:EXAMINE STACK TO FIND CAUSE
721	000214	000216	.+2	:UNEXPECTED TRAP TO THIS LOCATION
722	000216	000000	HALT	:EXAMINE STACK TO FIND CAUSE
723	000220	000222	.+2	:UNEXPECTED TRAP TO THIS LOCATION
724	000222	000000	HALT	:EXAMINE STACK TO FIND CAUSE
725	000224	000226	.+2	:UNEXPECTED TRAP TO THIS LOCATION
726	000226	000000	HALT	:EXAMINE STACK TO FIND CAUSE
727	000230	000232	.+2	:UNEXPECTED TRAP TO THIS LOCATION
728	000232	000000	HALT	:EXAMINE STACK TO FIND CAUSE
729	000234	000236	.+2	:UNEXPECTED TRAP TO THIS LOCATION
730	000236	000000	HALT	:EXAMINE STACK TO FIND CAUSE
731	000240	000242	.+2	:UNEXPECTED TRAP TO THIS LOCATION
732	000242	000000	HALT	:EXAMINE STACK TO FIND CAUSE
733	000244	000246	.+2	:UNEXPECTED TRAP TO THIS LOCATION
734	000246	000000	HALT	:EXAMINE STACK TO FIND CAUSE
735	000250	000252	.+2	:UNEXPECTED TRAP TO THIS LOCATION
736	000252	000000	HALT	:EXAMINE STACK TO FIND CAUSE
737	000254	000256	.+2	:UNEXPECTED TRAP TO THIS LOCATION
738	000256	000000	HALT	:EXAMINE STACK TO FIND CAUSE
739	000260	000262	.+2	:UNEXPECTED TRAP TO THIS LOCATION
740	000262	000000	HALT	:EXAMINE STACK TO FIND CAUSE
741	000264	000266	.+2	:UNEXPECTED TRAP TO THIS LOCATION
742	000266	000000	HALT	:EXAMINE STACK TO FIND CAUSE
743	000270	000272	.+2	:UNEXPECTED TRAP TO THIS LOCATION
744	000272	000000	HALT	:EXAMINE STACK TO FIND CAUSE
745	000274	000276	.+2	:UNEXPECTED TRAP TO THIS LOCATION
746	000276	000000	HALT	:EXAMINE STACK TO FIND CAUSE
747	000300	000302	.+2	:UNEXPECTED TRAP TO THIS LOCATION
748	000302	000000	HALT	:EXAMINE STACK TO FIND CAUSE
749	000304	000306	.+2	:UNEXPECTED TRAP TO THIS LOCATION
750	000306	000000	HALT	:EXAMINE STACK TO FIND CAUSE
751	000310	000312	.+2	:UNEXPECTED TRAP TO THIS LOCATION
752	000312	000000	HALT	:EXAMINE STACK TO FIND CAUSE
753	000314	000316	.+2	:UNEXPECTED TRAP TO THIS LOCATION
754	000316	000000	HALT	:EXAMINE STACK TO FIND CAUSE
755	000320	000322	.+2	:UNEXPECTED TRAP TO THIS LOCATION
756	000322	000000	HALT	:EXAMINE STACK TO FIND CAUSE
757	000324	000326	.+2	:UNEXPECTED TRAP TO THIS LOCATION
758	000326	000000	HALT	:EXAMINE STACK TO FIND CAUSE
759	000330	000332	.+2	:UNEXPECTED TRAP TO THIS LOCATION
760	000332	000000	HALT	:EXAMINE STACK TO FIND CAUSE

E02

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 17
DZDQFD.P11 21-DEC-76 16:34

TRAPCATCHER FOR UNEXPECTED INTERRUPTS

761	000334	000336	.+2	: UNEXPECTED TRAP TO THIS LOCATION
762	000336	000000	HALT	: EXAMINE STACK TO FIND CAUSE
763	000340	000342	.+2	: UNEXPECTED TRAP TO THIS LOCATION
764	000342	000000	HALT	: EXAMINE STACK TO FIND CAUSE
765	000344	000346	.+2	: UNEXPECTED TRAP TO THIS LOCATION
766	000346	000000	HALT	: EXAMINE STACK TO FIND CAUSE
767	000350	000352	.+2	: UNEXPECTED TRAP TO THIS LOCATION
768	000352	000000	HALT	: EXAMINE STACK TO FIND CAUSE
769	000354	000356	.+2	: UNEXPECTED TRAP TO THIS LOCATION
770	000356	000000	HALT	: EXAMINE STACK TO FIND CAUSE
771	000360	000362	.+2	: UNEXPECTED TRAP TO THIS LOCATION
772	000362	000000	HALT	: EXAMINE STACK TO FIND CAUSE
773	000364	000366	.+2	: UNEXPECTED TRAP TO THIS LOCATION
774	000366	000000	HALT	: EXAMINE STACK TO FIND CAUSE
775	000370	000372	.+2	: UNEXPECTED TRAP TO THIS LOCATION
776	000372	000000	HALT	: EXAMINE STACK TO FIND CAUSE
777	000374	000376	.+2	: UNEXPECTED TRAP TO THIS LOCATION
778	000376	000000	HALT	: EXAMINE STACK TO FIND CAUSE
779	000400	000402	.+2	: UNEXPECTED TRAP TO THIS LOCATION
780	000402	000000	HALT	: EXAMINE STACK TO FIND CAUSE
781	000404	000406	.+2	: UNEXPECTED TRAP TO THIS LOCATION
782	000406	000000	HALT	: EXAMINE STACK TO FIND CAUSE
783	000410	000412	.+2	: UNEXPECTED TRAP TO THIS LOCATION
784	000412	000000	HALT	: EXAMINE STACK TO FIND CAUSE
785	000414	000416	.+2	: UNEXPECTED TRAP TO THIS LOCATION
786	000416	000000	HALT	: EXAMINE STACK TO FIND CAUSE
787	000420	000422	.+2	: UNEXPECTED TRAP TO THIS LOCATION
788	000422	000000	HALT	: EXAMINE STACK TO FIND CAUSE
789	000424	000426	.+2	: UNEXPECTED TRAP TO THIS LOCATION
790	000426	000000	HALT	: EXAMINE STACK TO FIND CAUSE
791	000430	000432	.+2	: UNEXPECTED TRAP TO THIS LOCATION
792	000432	000000	HALT	: EXAMINE STACK TO FIND CAUSE
793	000434	000436	.+2	: UNEXPECTED TRAP TO THIS LOCATION
794	000436	000000	HALT	: EXAMINE STACK TO FIND CAUSE
795	000440	000442	.+2	: UNEXPECTED TRAP TO THIS LOCATION
796	000442	000000	HALT	: EXAMINE STACK TO FIND CAUSE
797	000444	000446	.+2	: UNEXPECTED TRAP TO THIS LOCATION
798	000446	000000	HALT	: EXAMINE STACK TO FIND CAUSE
799	000450	000452	.+2	: UNEXPECTED TRAP TO THIS LOCATION
800	000452	000000	HALT	: EXAMINE STACK TO FIND CAUSE
801	000454	000456	.+2	: UNEXPECTED TRAP TO THIS LOCATION
802	000456	000000	HALT	: EXAMINE STACK TO FIND CAUSE
803	000460	000462	.+2	: UNEXPECTED TRAP TO THIS LOCATION
804	000462	000000	HALT	: EXAMINE STACK TO FIND CAUSE
805	000464	000466	.+2	: UNEXPECTED TRAP TO THIS LOCATION
806	000466	000000	HALT	: EXAMINE STACK TO FIND CAUSE
807	000470	000472	.+2	: UNEXPECTED TRAP TO THIS LOCATION
808	000472	000000	HALT	: EXAMINE STACK TO FIND CAUSE
809	000474	000476	.+2	: UNEXPECTED TRAP TO THIS LOCATION
810	000476	000000	HALT	: EXAMINE STACK TO FIND CAUSE
811	000500	000502	.+2	: UNEXPECTED TRAP TO THIS LOCATION
812	000502	000000	HALT	: EXAMINE STACK TO FIND CAUSE
813	000504	000506	.+2	: UNEXPECTED TRAP TO THIS LOCATION
814	000506	000000	HALT	: EXAMINE STACK TO FIND CAUSE
815	000510	000512	.+2	: UNEXPECTED TRAP TO THIS LOCATION
816	000512	000000	HALT	: EXAMINE STACK TO FIND CAUSE

817	000514	000516	.+2	:UNEXPECTED TRAP TO THIS LOCATION
818	000516	000000	HALT	:EXAMINE STACK TO FIND CAUSE
819	000520	000522	.+2	:UNEXPECTED TRAP TO THIS LOCATION
820	000522	000000	HALT	:EXAMINE STACK TO FIND CAUSE
821	000524	000526	.+2	:UNEXPECTED TRAP TO THIS LOCATION
822	000526	000000	HALT	:EXAMINE STACK TO FIND CAUSE
823	000530	000532	.+2	:UNEXPECTED TRAP TO THIS LOCATION
824	000532	000000	HALT	:EXAMINE STACK TO FIND CAUSE
825	000534	000536	.+2	:UNEXPECTED TRAP TO THIS LOCATION
826	000536	000000	HALT	:EXAMINE STACK TO FIND CAUSE
827	000540	000542	.+2	:UNEXPECTED TRAP TO THIS LOCATION
828	000542	000000	HALT	:EXAMINE STACK TO FIND CAUSE
829	000544	000546	.+2	:UNEXPECTED TRAP TO THIS LOCATION
830	000546	000000	HALT	:EXAMINE STACK TO FIND CAUSE
831	000550	000552	.+2	:UNEXPECTED TRAP TO THIS LOCATION
832	000552	000000	HALT	:EXAMINE STACK TO FIND CAUSE
833	000554	000556	.+2	:UNEXPECTED TRAP TO THIS LOCATION
834	000556	000000	HALT	:EXAMINE STACK TO FIND CAUSE
835	000560	000562	.+2	:UNEXPECTED TRAP TO THIS LOCATION
836	000562	000000	HALT	:EXAMINE STACK TO FIND CAUSE
837	000564	000566	.+2	:UNEXPECTED TRAP TO THIS LOCATION
838	000566	000000	HALT	:EXAMINE STACK TO FIND CAUSE
839	000570	000572	.+2	:UNEXPECTED TRAP TO THIS LOCATION
840	000572	000000	HALT	:EXAMINE STACK TO FIND CAUSE
841	000574	000576	.+2	:UNEXPECTED TRAP TO THIS LOCATION
842	000576	000000	HALT	:EXAMINE STACK TO FIND CAUSE
843	000600	000602	.+2	:UNEXPECTED TRAP TO THIS LOCATION
844	000602	000000	HALT	:EXAMINE STACK TO FIND CAUSE
845	000604	000606	.+2	:UNEXPECTED TRAP TO THIS LOCATION
846	000606	000000	HALT	:EXAMINE STACK TO FIND CAUSE
847	000610	000612	.+2	:UNEXPECTED TRAP TO THIS LOCATION
848	000612	000000	HALT	:EXAMINE STACK TO FIND CAUSE
849	000614	000616	.+2	:UNEXPECTED TRAP TO THIS LOCATION
850	000616	000000	HALT	:EXAMINE STACK TO FIND CAUSE
851	000620	000622	.+2	:UNEXPECTED TRAP TO THIS LOCATION
852	000622	000000	HALT	:EXAMINE STACK TO FIND CAUSE
853	000624	000626	.+2	:UNEXPECTED TRAP TO THIS LOCATION
854	000626	000000	HALT	:EXAMINE STACK TO FIND CAUSE
855	000630	000632	.+2	:UNEXPECTED TRAP TO THIS LOCATION
856	000632	000000	HALT	:EXAMINE STACK TO FIND CAUSE
857	000634	000636	.+2	:UNEXPECTED TRAP TO THIS LOCATION
858	000636	000000	HALT	:EXAMINE STACK TO FIND CAUSE
859	000640	000642	.+2	:UNEXPECTED TRAP TO THIS LOCATION
860	000642	000000	HALT	:EXAMINE STACK TO FIND CAUSE
861	000644	000646	.+2	:UNEXPECTED TRAP TO THIS LOCATION
862	000646	000000	HALT	:EXAMINE STACK TO FIND CAUSE
863	000650	000652	.+2	:UNEXPECTED TRAP TO THIS LOCATION
864	000652	000000	HALT	:EXAMINE STACK TO FIND CAUSE
865	000654	000656	.+2	:UNEXPECTED TRAP TO THIS LOCATION
866	000656	000000	HALT	:EXAMINE STACK TO FIND CAUSE
867	000660	000662	.+2	:UNEXPECTED TRAP TO THIS LOCATION
868	000662	000000	HALT	:EXAMINE STACK TO FIND CAUSE
869	000664	000666	.+2	:UNEXPECTED TRAP TO THIS LOCATION
870	000666	000000	HALT	:EXAMINE STACK TO FIND CAUSE
871	000670	000672	.+2	:UNEXPECTED TRAP TO THIS LOCATION
872	000672	000000	HALT	:EXAMINE STACK TO FIND CAUSE

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 19
 DZDQFD.P11 21-DEC-76 16:34

TRAPCATCHER FOR UNEXPECTED INTERRUPTS

873	000674	000676	.+2	:UNEXPECTED TRAP TO THIS LOCATION
874	000676	000000	HALT	:EXAMINE STACK TO FIND CAUSE
875	000700	000702	.+2	:UNEXPECTED TRAP TO THIS LOCATION
876	000702	000000	HALT	:EXAMINE STACK TO FIND CAUSE
877	000704	000706	.+2	:UNEXPECTED TRAP TO THIS LOCATION
878	000706	000000	HALT	:EXAMINE STACK TO FIND CAUSE
879	000710	000712	.+2	:UNEXPECTED TRAP TO THIS LOCATION
880	000712	000000	HALT	:EXAMINE STACK TO FIND CAUSE
881	000714	000716	.+2	:UNEXPECTED TRAP TO THIS LOCATION
882	000716	000000	HALT	:EXAMINE STACK TO FIND CAUSE
883	000720	000722	.+2	:UNEXPECTED TRAP TO THIS LOCATION
884	000722	000000	HALT	:EXAMINE STACK TO FIND CAUSE
885	000724	000726	.+2	:UNEXPECTED TRAP TO THIS LOCATION
886	000726	000000	HALT	:EXAMINE STACK TO FIND CAUSE
887	000730	000732	.+2	:UNEXPECTED TRAP TO THIS LOCATION
888	000732	000000	HALT	:EXAMINE STACK TO FIND CAUSE
889	000734	000736	.+2	:UNEXPECTED TRAP TO THIS LOCATION
890	000736	000000	HALT	:EXAMINE STACK TO FIND CAUSE
891	000740	000742	.+2	:UNEXPECTED TRAP TO THIS LOCATION
892	000742	000000	HALT	:EXAMINE STACK TO FIND CAUSE
893	000744	000746	.+2	:UNEXPECTED TRAP TO THIS LOCATION
894	000746	000000	HALT	:EXAMINE STACK TO FIND CAUSE
895	000750	000752	.+2	:UNEXPECTED TRAP TO THIS LOCATION
896	000752	000000	HALT	:EXAMINE STACK TO FIND CAUSE
897	000754	000756	.+2	:UNEXPECTED TRAP TO THIS LOCATION
898	000756	000000	HALT	:EXAMINE STACK TO FIND CAUSE
899	000760	000762	.+2	:UNEXPECTED TRAP TO THIS LOCATION
900	000762	000000	HALT	:EXAMINE STACK TO FIND CAUSE
901	000764	000766	.+2	:UNEXPECTED TRAP TO THIS LOCATION
902	000766	000000	HALT	:EXAMINE STACK TO FIND CAUSE
903	000770	000772	.+2	:UNEXPECTED TRAP TO THIS LOCATION
904	000772	000000	HALT	:EXAMINE STACK TO FIND CAUSE
905	000774	000776	.+2	:UNEXPECTED TRAP TO THIS LOCATION
906	000776	000000	HALT	:EXAMINE STACK TO FIND CAUSE

H02

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 20
 DZDQFD.P11 21-DEC-76 16:34 ROUTINES USED FOR AUTO SIZING.

```

907                                     ;STANDARD INTERRUPT VECTORS
908
909                                     . =24
910 000024 014722                       .PFAIL                       ;POWER FAIL HANDLER
911 000026 000340                       340                          ;SERVICE AT LEVEL 7
912 000030 014372                       .HLT                          ;ERROR HANDLER
913 000032 000340                       340                          ;SERVICE AT LEVEL 7
914 000034 014340                       .TRPSRV                       ;GENERAL HANDLER DISPATCH SERVICE
915 000036 000340                       340                          ;SERVICE AT LEVEL 7
916                                     . =46
917 000046 013120                       LOGICAL                        ;ACT HOOKS
918                                     . =52
919 000052 000000                       .WORD 0
920                                     ;THIS ROUTINE TRIES TO FORCE THE RECEIVER TO INTERRUPT
921                                     ;TO ITS VECTOR WHERE IT WILL PICK UP THE STATUS LOCATION
922                                     ;FOR ITS NEW PC; AND PICK UP AN IOT INSTRUCTION FOR ITS
923                                     ;NEW PS. WHEN THE NEW PC IS FETCHED AN IOT INSTRUCTION IS
924                                     ;EXECUTED, TRAPPING TO LOCATION 20 WHERE A ROUTINE IS EXECUTED
925                                     ;TO TAKE THE PC FROM THE STACK AND USE IT AS THE VECTOR ADDRESS
926                                     . =56
927
928                                     VECMAP:
929 000056 010120 000004                 1$:  MOV      R1,(R0)+          ;START FILLING THE VECTOR AREA
930 000060 012721 000004                 MOV      #4,(R1)+            WITH .+2; IOT (4)
931 000064 022021 001000                 CMP      (R0)+,(R1)+        ;UPDATE THE POINTERS
932 000066 020127 001000                 CMP      R1,#1000           ;IS ALL FLOATING VECTOR AREA DONE
933 000072 101771 000020                 BLOS    1$                  ;BR IF NOT ALL DONE
934 000074 012737 000146 000020         MOV      #45,2#20           ;SET FOR IOT TRAP BY DQ11
935 000102 013737 001500 001244         MOV      DQACTV,TEMP1      ;GET THE ACTIVE DQ11 S
936 000110 006037 001244                 2$:  ROR      TEMP1          ;ARE YOU ACTIVE.. DQ11
937 000114 103023 000000                 BCC     5$                  ;IF CARRY CLEAR.. NO MORE DQ11S
938 000116 005037 177776                 CLR     PS                  ;CLEAR PS
939 000122 005722 000000                 TST     (R2)+              ;PUT POINTER TO STATUS TABLE
940 000124 012772 000340 177776         MOV      #340,2-2(R2)      ;TRY AND SET PRI/SEC DONE AND IE
941 000132 105200 000000                 INCB   RO                   ;DELAY.....
942 000134 001376 000000                 BNE    .-2                 ;.....DELAY
943 000136 112712 000300                 MOVB   #300,(R2)          ;NO INTERRUPT ASSUME 300 FIX IN TEST C
944 000142 005722 000000                 3$:  TST     (R2)+          ;UPDATE POINTERS
945 000144 000761 000000                 BR     2$                  ;GO DO IT AGAIN
946 000146 051612 000007                 4$:  BIS     (SP),(R2)      ;ENTERD BY IOT TRAP BY DQ11
947 000150 042712 000007                 BIC    #7,(R2)            ;CLEAR UNWANTED BITS
948 000154 022626 000142                 CMP     (SP)+,(SP)+        ;POP IOT JUNK OFF STACK
949 000156 012716 000142                 MOV     #3$, (SP)         ;SET RETURN PC ON STACK
950 000162 000002 000002                 RTI    PC                  ;GO HOME
951 000164 000207 000002                 5$:  RTS     PC             ;ALL SIZING IS DONE
952
953                                     ;****SOFTWARE SWITCH REGISTER****
954                                     . =174
955 000174 000000 000000                 DISPRG: 0                   ;SOFTWARE DISPLAY REGISTER
956 000176 000000 000000                 SWREG:  0                   ;SOFTWARE SWITCH REGISTER
957
958                                     ;PROGRAM START
959
960                                     . =200
961 000200 000137 001512                 JMP     .START             ;GO TO START OF PROGRAM
962

```



```

1019 000540 012737 000006 000004      MOV      #6, R0          ; RESET TIME OUT VECTOR
1020 000546 013737 001500 001502      MOV      DACTV, SAVACT ; SAVE ACTIVE
1021 000554 012737 000340 000022      MOV      #340, R22     ; SET IOT TRAP PRIO: TO 7
1022 000562 012702 001400          MOV      #1400, R2     ; SET TABLE POINTER
1023 000566 012700 000300          MOV      #300, R0      ; SET VECTOR START
1024 000572 012701 000302          MOV      #302, R1      ; SET VECTOR+2 START
1025 000576 000137 000056          JMP      VECMAP        ; GO FIND THE VECTORS
1026 000602 104402          4$:      TYPE          ; TYPE MESSAGE
1027 000604 015263          MERR2          ; I DIDN'T FIND ANY DQ11S. DON'T USE AUTO SIZE.
1028 000606 005000          CLR      R0          ;
1029 000610 000000          HALT          ; HOW CAN I TEST NO DQ11S
1030 000612 000776          BR          ; DON'T LET OPR HIT CNT. SW
1031 000614 012716 000466          5$:      MOV      #-2, (SP) ; ENTERED BY TIME OUT TRAP
1032 000620 000002          RTI          ; GO HOME.
1033
1034
1035          001000          . =1000
1036 001000 005377 040515 047111  MTITLE: .ASCIZ <377><12>/MAINDEC-11-DZDGF-D/<377>/SEQUENCE REG TEST/<377>
1037 001006 042504 026503 030461
1038 001014 042055 042132 043121
1039 001022 042055 051777 050505
1040 001030 042525 041516 020105
1041 001036 042522 020107 042524
1042 001044 052123 000377
1043
1044          001200          . =1200
1045          ;INDIRECT POINTERS
1046
1047 001200 177570          SWR:      177570        ; SWITCH REGISTER POINTER
1048 001202 177570          LIGHTS:   177570       ; DISPLAY REGISTER POINTER
1049 001204 177560          TKCSR:    177560       ; TELETYPE KEYBOARD CONTROL REGISTER
1050 001206 177562          TKDBR:    177562       ; TELETYPE KEYBOARD DATA BUFFER
1051 001210 177564          TPCSR:    177564       ; TELEPRINTER CONTROL REGISTER
1052 001212 177566          TPDBR:    177566       ; TELEPRINTER DATA BUFFER
1053
1054          ;PROGRAM CONTROL PARAMETERS
1055
1056 001214 000000          RETURN:   0           ; SCOPE ADDRESS FOR LOOP ON TEST
1057 001216 000000          NEXT:     0           ; ADDRESS OF NEXT TEST TO BE EXECUTED
1058 001220 000000          LOCK:     0           ; ADDRESS FOR LOCK ON CURRENT DATA
1059 001222 000003          ICOUNT:   3           ; NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
1060 001224 000000          LPCNT:    0           ; NUMBER OF ITERATIONS COMPLETED
1061 001226 000000          TSTNO:    0           ; NUMBER OF TEST IN PROGRESS
1062 001230 000000          PASCNT:   0           ; NUMBER OF PASSES COMPLETED
1063 001232 000000          ERRCNT:   0           ; TOTAL NUMBER OF ERRORS
1064 001234 000000          LSTERR:   0           ; PC OF LAST ERROR CALL
1065
1066          ;PROGRAM VARIABLES
1067
1068 001236 000000          CHAR1:    0
1069 001240 000000          CHAR2:    0
1070 001242 000000          CHAR3:    0
1071 001244 000000          TEMP1:    0           ; TEMPORARY STORAGE
1072 001246 000000          TEMP2:    0           ; TEMPORARY STORAGE
1073 001250 000000          TEMP3:    0           ; TEMPORARY STORAGE
1074 001252 000000          TEMP4:    0           ; TEMPORARY STORAGE
    
```

K02

DZQDF MACY11 27(1006) 22-DEC-76 11:36 PAGE 23
DZQDFD.P11 21-DEC-76 16:34

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

1075 001254 000000
1076 001256 000000
1077 001260 000000
1078 001262 000000
1079 001264 000000
1080 001266 000000
1081 001270 000000
1082 001272 000000
1083 001274 000000
1084 001276 000000
1085 001300 000001
1086 001302 000000
1087 001304 000000
1088 001306 000000

TEMPS: 0
SAVR0: 0
SAVR1: 0
SAVR2: 0
SAVR3: 0
SAVR4: 0
SAVR5: 0
SAVSP: 0
SAVPC: 0
SAVNUM: 0
CREAM: .BLKW 1
RUNFLG: 0
RUN: 0
RUNCNT: 0

: TEMPORARY STORAGE
: R0 STORAGE
: R1 STORAGE
: R2 STORAGE
: R3 STORAGE
: R4 STORAGE
: R5 STORAGE
: STACK POINTER STORAGE
: PROGRAM COUNTER STORAGE

```

1089
1090 ;PROGRAM CONTROL FLAGS
1091
1092 001310 000 INIFLG: .BYTE 0 ;PROGRAM INITIALIZATION FLAG
1093 001311 000 STFLG: .BYTE 0 ;TEST START FLAG
1094 001312 000 ERRFLG: .BYTE 0 ;ERROR OCCURED FLAG
1095 001313 000 LOKFLG: .BYTE 0 ;LOCK ON CURRENT TEST FLAG
1096 000000 SY=0
1097
1098 ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
1099 ;POINTERS TO SUBROUTINES CAN BE FOUND
1100 ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS
1101
1102 ;*****
1103 ;*****
1104 001314 .TRPTAB:
1105 104400 SCOPE=TRAP+0 ;CALL TO SCOPE LOOP AND ITERATION HANDLER
1106 001314 013174 .SCOPE
1107 104401 SCOP1=TRAP+1 ;CALL TO LOOP ON CURRENT DATA HANDLER
1108 001316 013306 .SCOP1
1109 104402 TYPE=TRAP+2 ;CALL TO TELETYPE OUTPUT ROUTINE
1110 001320 013326 .TYPE
1111 104403 INSTR=TRAP+3 ;CALL TO ASCII STRING INPUT ROUTINE
1112 001322 013434 .INSTR
1113 104404 INSTER=TRAP+4 ;CALL TO INPUT ERROR HANDLER
1114 001324 013552 .INSTER
1115 104405 PARAM=TRAP+5 ;CALL TO NUMERICAL DATA INPUT ROUTINE
1116 001326 013604 .PARAM
1117 104406 SAVOS=TRAP+6 ;CALL TO REGISTER SAVE ROUTINE
1118 001330 014020 .SAVOS
1119 104407 RESOS=TRAP+7 ;CALL TO REGISTER RESTORE ROUTINE
1120 001332 014060 .RESOS
1121 104410 CONVRT=TRAP+10 ;CALL TO DATA OUTPUT ROUTINE
1122 001334 014112 .CONVRT
1123 104411 CNVRT=TRAP+11 ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.
1124 001336 014116 .CNVRT
1125 104412 MSTCLR=TRAP+12 ;CALL TO ISSUE MASTER CLEAR
1126 001340 010766 .MSTCLR
1127 104413 MEMCLR=TRAP+13 ;CALL TO CLEAR ALL SCRATCH PAD MEMORIES
1128 001342 010652 .MEMCLR
1129 104414 CKSWR=TRAP+14 ;CALL TO ALLOW SWREG TO BE LOADED FROM TTY
1130 001344 015020 .CKSWR
1131 104415 CNTLU=TRAP+15 ;CALL TO ALLOW LOADING OF SWREG FROM TTY
1132 001346 015074 .CNTLU
1133
1134 ;*****
1135 ;*****
1136
1137 ;DQ11 VECTOR AND REGISTER INDIRECT POINTERS
1138
1139 001350 000000 DQAVEC: 0 ;POINTER TO DQ11 RECEIVER INTERRUPT VECTOR
1140 001352 000000 DQRLVL: 0 ;POINTER TO DQ11 RECEIVER INTERRUPT SERVICE PS
1141 001354 000000 DQAVEC: 0 ;POINTER TO DQ11 TRANSMITTER INTERRUPT VECTOR
1142 001356 000000 DQTLVL: 0 ;POINTER TO DQ11 TRANSMITTER INTERRUPT SERVICE PS
1143 001360 000000 DQRCR: 0 ;POINTER TO DQ11 RECEIVER CONTROL REGISTER
1144 001362 000000 DQRCSH: 0 ;POINTER TO HIGH BYTE OF DQ11 RECEIVER CONTROL REGISTER
    
```

M02

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 25
 DZDQFD.P11 21-DEC-76 16:34

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```

1145 001364 000000      DQTCR: 0          ; POINTER TO DQ11 TRANSMITTER CONTROL REGISTER
1146 001366 000000      DQERR: 0          ; POINTER TO DQ11 ERROR REGISTER
1147 001370 000000      DQREG: 0          ; POINTER TO HIGH BYTE OF ERROR REGISTER
1148 001372 000000      DQSEC: 0          ; POINTER TO DQ11 SECONDARY REGISTER
1149 001374 000000      DQSECH: 0         ; POINTER TO HIGH BYTE OF DQ11 SECONDARY REGISTER
  
```

1150
1151
1152
1153
1154

;DQ11 STATUS TABLE AND ADDRESS ASSIGNMENTS

```

1155                001400      .=1400
1156 001400 000001      DQCR00: .BLKW 1      ; CONTROL STATUS REGISTER FOR DEVICE NO: 00
1157 001402 000001      DQST00: .BLKW 1      ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 00
1158 001404 000001      DQCR01: .BLKW 1      ; CONTROL STATUS REGISTER FOR DEVICE NO: 01
1159 001406 000001      DQST01: .BLKW 1      ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 01
1160 001410 000001      DQCR02: .BLKW 1      ; CONTROL STATUS REGISTER FOR DEVICE NO: 02
1161 001412 000001      DQST02: .BLKW 1      ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 02
1162 001414 000001      DQCR03: .BLKW 1      ; CONTROL STATUS REGISTER FOR DEVICE NO: 03
1163 001416 000001      DQST03: .BLKW 1      ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 03
1164 001420 000001      DQCR04: .BLKW 1      ; CONTROL STATUS REGISTER FOR DEVICE NO: 04
1165 001422 000001      DQST04: .BLKW 1      ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 04
1166 001424 000001      DQCR05: .BLKW 1      ; CONTROL STATUS REGISTER FOR DEVICE NO: 05
1167 001426 000001      DQST05: .BLKW 1      ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 05
1168 001430 000001      DQCR06: .BLKW 1      ; CONTROL STATUS REGISTER FOR DEVICE NO: 06
1169 001432 000001      DQST06: .BLKW 1      ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 06
1170 001434 000001      DQCR07: .BLKW 1      ; CONTROL STATUS REGISTER FOR DEVICE NO: 07
1171 001436 000001      DQST07: .BLKW 1      ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 07
1172 001440 000001      DQCR10: .BLKW 1      ; CONTROL STATUS REGISTER FOR DEVICE NO: 10
1173 001442 000001      DQST10: .BLKW 1      ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 10
1174 001444 000001      DQCR11: .BLKW 1      ; CONTROL STATUS REGISTER FOR DEVICE NO: 11
1175 001446 000001      DQST11: .BLKW 1      ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 11
1176 001450 000001      DQCR12: .BLKW 1      ; CONTROL STATUS REGISTER FOR DEVICE NO: 12
1177 001452 000001      DQST12: .BLKW 1      ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 12
1178 001454 000001      DQCR13: .BLKW 1      ; CONTROL STATUS REGISTER FOR DEVICE NO: 13
1179 001456 000001      DQST13: .BLKW 1      ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 13
1180 001460 000001      DQCR14: .BLKW 1      ; CONTROL STATUS REGISTER FOR DEVICE NO: 14
1181 001462 000001      DQST14: .BLKW 1      ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 14
1182 001464 000001      DQCR15: .BLKW 1      ; CONTROL STATUS REGISTER FOR DEVICE NO: 15
1183 001466 000001      DQST15: .BLKW 1      ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 15
1184 001470 000001      DQCR16: .BLKW 1      ; CONTROL STATUS REGISTER FOR DEVICE NO: 16
1185 001472 000001      DQST16: .BLKW 1      ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 16
1186 001474 000001      DQCR17: .BLKW 1      ; CONTROL STATUS REGISTER FOR DEVICE NO: 17
1187 001476 000001      DQST17: .BLKW 1      ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 17
1188 001500 000001      DQACTV: .BLKW 1      ; HOLD ACTIVE BITS FOR TESTING
1189 001502 000001      SAVACT: .BLKW 1      ; SAVE NUMBER OF ACTIVE DQ11S
1190 001504 000001      DONUM: .BLKW 1      ; OCTAL NUMBER OF TOTAL NUMBER OF DQ11S
1191 001506 000001      DQCSR: .BLKW 1      ; CSR OF DQ11 UNDER TEST
1192 001510 000001      DQSTAT: .BLKW 1      ; VECTOR AND CONFIGURATION STATUS OF DQ11 UNDER TEST
  
```

```

1193
1194                ;PROGRAM INITIALIZATION
1195                ;LOCK OUT INTERRUPTS
1196                ;SET UP PROCESSOR STACK
1197                ;SET UP POWER FAIL VECTOR
1198                ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1199                ;TYPE TITLE MESSAGE
1200
  
```

N02

DZDGF MACY11 27(1006) 22-DEC-76 11:36 PAGE 26
 DZDGF.P11 21-DEC-76 16:34 PROGRAM INITIALIZATION AND START UP.

1201	001512	012737	000340	177776	.START:	MOV	#340,PS	;LOCK OUT INTERRUPTS
1202	001520	012706	001200			MOV	#STACK,SP	;SET UP STACK
1203	001524	012737	014722	000024		MOV	#,PFAIL,@#24	;SET UP POWER FAIL VECTOR
1204	001532	013737	001504	001276		MOV	DNUM,SAVNUM	
1205	001540	105037	001311			CLRB	STFLG	;CLEAR START FLAG
1206	001544	005037	001230			CLR	PASCNT	;CLEAR PASS COUNT
1207	001550	105037	001312			CLRB	ERRFLG	;CLEAR ERROR FLAG
1208	001554	005037	001302			CLR	RUNFLG	
1209	001560	012737	001400	001300		MOV	#1400,CREAM	
1210	001566	005037	001232			CLR	ERRCNT	;CLEAR ERROR COUNT
1211	001572	005037	001234			CLR	LSTERR	;CLEAR LAST ERROR POINTER
1212	001576	012737	000001	001226		MOV	#1,ISTNO	;SET UP FOR TEST 1
1213	001604	012737	001512	001214		MOV	#.START,RETURN	;SET UP FOR POWER FAIL BEFORE
1214								;TESTING STARTS
1215	001612	105737	001310			TSTB	INIFLG	;HAS INITIALIZATION BEEN PERFORMED
1216	001616	001075				BNE	12\$	
1217	001620	104402	001000			TYPE	MTITLE	;TYPE TITLE MESSAGE
1218	001624	105137	001310			COMB	INIFLG	;IF NOT SET FLAG AND DO
1219								
1220	001630	012737	177570	001200		MOV	#OSWR,SWR	;MOV HARDWARE SWR TO SWR
1221	001636	012737	177570	001202		MOV	#DLIGHTS,LIGHTS	;MOV DISPLAY LIGHTS TO LIGHTS
1222	001644	013746	000006			MOV	@#6,-(SP)	;SAVE VECTORS
1223	001650	013746	000004			MOV	@#4,-(SP)	
1224	001654	012737	001674	000004		MOV	#64\$,@#4	;SET UP FOR TIMEOUT
1225	001662	022777	177777	177310		CMP	#-1,@SWR	;REFERENCE HARDWARE SWITCH REGISTER
1226	001670	001402				BEQ	65\$	
1227	001672	000407				BR	66\$	
1228	001674	022626			64\$:	CMP	(SP)+,(SP)+	;ADJUST STACK
1229	001676	012737	000176	001200	65\$:	MOV	#SWREG,SWR	;POINT TO SOFTWARE SWITCH REG
1230	001704	012737	000174	001202		MOV	#DISPREG,LIGHTS	;POINT TO SOFT DISPLAY REG
1231	001712	012637	000004		66\$:	MOV	(SP)+,@#4	;RESTORE VECTORS
1232	001716	012637	000006			MOV	(SP)+,@#6	
1233	001722	005737	000042			TST	@#42	;UNDER MONITOR
1234	001726	001005				BNE	67\$	
1235	001730	022737	000176	001200		CMP	#SWREG,SWR	;IS SWREG USED
1236	001736	001001				BNE	67\$	
1237	001740	104415				CNTLU		
1238	001742	105777	177232		67\$:	TSTB	@SWR	
1239	001746	100402				BMI	+.b	
1240	001750	004737	000220			JSR	PC,CSRMAP	
1241	001754	104402	015550			TYPE	XHEAD	
1242	001760	012737	001400	001244		MOV	#1400,TEMP1	
1243	001766	017737	177252	001246		MOV	@TEMP1,TEMP2	
1244	001774	001406				BEQ	+.16	
1245	001776	104410				CONVRT		
1246	002000	015576				XSTATQ		
1247	002002	062737	000002	001244		ADD	#2,TEMP1	
1248	002010	000766				BR	.-22	
1249	002012	032777	000001	177160	12\$:	BIT	#SW00,@SWR	
1250	002020	001424				BEQ	1\$	
1251	002022	104402				TYPE		
1252	002024	015471				MNEW		
1253	002026	000000				CLR	RO	
1254	002028	000000				HALT		
1255	002030	104414				CKSWR		
1256	002034	027737	177140	001502		CMP	@SWR,SAVACT	

B03

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 27
 DZDQFD.P11 21-DEC-76 16:34 PROGRAM INITIALIZATION AND START UP.

```

1257 002042 101404      BLOS      11$
1258 002044 104402      TYPE
1259 01 046 015332     MERR3
1260 01 50 000000     HALT
1261 01 52 000776     BR
1262 01 054 017737   177120 001500 11$:  MOV      @SWR, DQACTV
1263 01 062 013700   001500      MOV      DQACTV, R0
1264 002066 000000     HALT
1265 002070 104414      CKSWR
1266 002072 012700   000300 1$:  MOV      #300, R0
1267 002076 012701   000302      MOV      #302, R1
1268 002102 0.0120     2$:  MOV      R1, (R0)+
1269 002104 005021     CLR      (R1)+
1270 002106 022021     CMP      (R0)+, (R1)+
1271 002110 022700   001000      CMP      #1000, R0
1272 002114 001372     BNE      2$
1273
1274                      ;TEST START AND RESTART
1275
1276 002116 012737   000340 177776 .BEGIN:  MOV      #340, PS          ;LOCK OUT INTERRUPTS
1277 002124 012706   001200      MOV      #STACK, SP      ;SET UP STACK
1278 002130 005737   000042      TST      @#42           ;IS PROGRAM UNDER MONITOR CONTROL
1279 002134 001040     BNE      3$
1280 002136 104414     CKSWR          ;CHECK FOR <IG>
1281 002140 032777   000004 177032  BIT      #BIT2, @SWR     ;CHECK FOR LOCK ON TEST
1282 002146 001411     BEQ      1$
1283 002150 104402   015370      TYPE      #MLOCK
1284 002154 012737   000240 013204  MOV      #NOP, TTST
1285 002162 012737   000240 013206  MOV      #NOP, TTST+2    ;SET UP TO LOCK
1286 002170 000406     BR       2$
1287 002172 013737   013302 013204 1$:  MOV      BRW, TTST
1288 002200 013737   013304 013206  MOV      BRX, TTST+2     ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1289 002206 032777   000002 176764 2$:  BIT      #SW01, @SWR    ;IF SW01=1, GET STARTING PC
1290 002214 001410     BEQ      3$
1291 002216 104403     INSTR
1292 002220 015356     MTSTPC
1293 002222 104405     PARAM
1294 002224 002254     TST1
1295 002226 010214     TLAST
1296 002230 001214     @RETURN
1297 002232 001      .BYTE 1
1298 002233 001      .BYTE 1
1299 002234 000403     BR       4$
1300 002236 012737   002254 001214 3$:  MOV      #TST1, RETURN   ;START AT TEST 1
1301 002244 104402   015260      TYPE      #R          ;TYPE R
1302 002250 000177   176740      JMP      @RETURN        ;START TESTING
1303
1304                      ; TEST 1
1305                      ;*****
1305 002254 012737   000001 001226  TST1:  MOV      #1, TSTNO
1306 002252 012737   002644 001214      MOV      #TST2, RETURN
1307 002270 012737   002644 001216      MOV      #TST2, NEXT
1308 002276 105737   001302      TSTB     RUNFLG        ;IS THIS MY FIRST TIME HERE?
1309 002302 001010     BNE      1$            ;BR IF FLAG IS SET
1310 002304 012737   000001 001304      MOV      #BIT0, RUN     ;SET RUN POINTER.
1311 002312 012737   000020 001306      MOV      #16, RUNCNT   ;SET FOR MAX OF 16 DQ11'S PER SYSTEM
1312 002320 105137   001302      COMB     RUNFLG        ;SET RUN FLAG

```

```

1313 002324 033737 001304 001500 1$: BIT RUN,DQACTV ;FIND AN ACTIVE DQ11 TO TEST.
1314 002332 001032 BNE 3$ ;BR IF I FOUND ONE TO TEST.
1315 002334 005737 001500 TST DQACTV ;FIND OUT IF THERE ARE NO DQ11 ACTIVE.
1316 002340 001423 BEQ 2$ ;BR TO FATAL ERROR. WHY AM I HERE IF NO ACTIVE DQ11'S??
1317 002342 000257 CCC ;CLEAR ALL THE CONDITION CODES OF CPU
1318 002344 006137 001304 ROL RUN ;UPDATE RUN POINTER
1319 002350 002737 000004 001300 ADD #4,CREAM ;UPDATE ADDRESS POINTER.
1320 002356 005337 001306 DEC RUNCNT ;DEC NUMBER OF TIMES I LOOKED AT ACTIVE.
1321 002362 001360 BNE 1$ ;BR AND KEEP LOOKING.
1322 002364 012737 000020 001306 MOV #16,RUNCNT ;START RESTORING MY POINTERS.
1323 002372 012737 001400 001300 MOV #1400,CREAM ;RESTORE ADDRESS POINTER
1324 002400 012737 000001 001304 MOV #1,RUN ;RESTORE RUN POINTER.
1325 002406 000746 BR 1$ ;KEEP ON TESTING.
1326 002410 104402 2$: TYPE ;ALLERT OPERATOR OF FATAL ERROR
1327 002412 015263 MERR2 ;NO DQ11 ACTIVE. WHY AM I HERE??
1328 002414 000000 HALT ;YOU MUST RELOAD DQ11 DIAGNOSTIC!!
1329 002416 000776 BR ;STICK HERE ON CONT.
1330 002420 000257 3$: CCC ;CLEAR CPU COND. CODES
1331 002422 006137 001304 ROL RUN ;UPDATE RUN. ACTIVE DQ11 FOUND.
1332 002426 017737 176646 001506 MOV #2,CREAM,DQCSR ;PLACE ADDRESS OF DQ11 AT DQCSR
1333 002434 062737 000002 001300 ADD #2,CREAM ;UPDATE ADDRESS POINTER
1334 002442 017737 176632 001510 MOV #2,CREAM,DQSTAT ;PLACE STATUS OF DQ11 AT DQSTAT
1335 002450 062737 000002 001300 ADD #2,CREAM ;UPDATE ADDRESS POINTER
1336 002456 013737 001506 001360 MOV DQCSR,DQCSR
1337 002464 013737 001510 001350 MOV DQSTAT,DQVEC
1338 002472 042737 177007 001350 BIC #177007,DQVEC
1339 002480 013737 001350 001352 MOV DQVEC,DQRLVL ;GENERATE ADDRESS OF RECEIVER INTERRUPT SERVICE PS
1340 002486 062737 000002 001352 ADD #2,DQRLVL
1341 002494 013737 001352 001354 MOV DQRLVL,DQVECC ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT VECTOR
1342 002502 062737 000002 001354 ADD #2,DQVECC
1343 002510 013737 001354 001356 MOV DQVECC,DQTLVL ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT SERVICE PS
1344 002516 062737 000002 001356 ADD #2,DQTLVL
1345 002524 013737 001360 001362 MOV DQCSR,DQCSH
1346 002532 005237 001362 INC DQCSH ;GENERATE ADDRESS OF HIGH BYTE
1347 002538 013737 001360 001364 MOV DQCSR,DQCSR ;GENERATE ADDRESS OF TRANSMITTER CONTROL REGISTER
1348 002544 062737 000002 001364 ADD #2,DQCSR
1349 002552 013737 001364 001366 MOV DQCSR,DQERR ;GENERATE ADDRESS OF ERROR REGISTER
1350 002560 062737 000002 001366 ADD #2,DQERR
1351 002566 013737 001366 001370 MOV DQERR,DQREG ;GENERATE ADDRESS OF HIGH BYTE OF ERROR REGISTER
1352 002574 005237 001370 INC DQREG
1353 002582 013737 001370 001372 MOV DQREG,DQSEC ;GENERATE ADDRESS OF SECONDARY REGISTER
1354 002590 005237 001372 INC DQSEC
1355 002598 013737 001372 001374 MOV DQSEC,DQSECH ;GENERATE ADDRESS OF HIGH BYTE
1356 002606 005237 001374 INC DQSECH
1357 ;
1358 ;THIS IS NOT A TEST!!!
1359 ; TEST 2
1360 ;*****
1361 002644 012737 000002 001226 tst2: MOV #2,TSTNO
1362 002652 012737 002662 001216 MOV #TST3,NEXT
1363 ;THE ORIGINAL TEST 2 IS
1364 ;NOW TEST 2 OF TAPE DZDQH
1365 002660 104400 SCOPE
1366
1367
1368

```

```

1369
1370
1371
1372
1373
1374
1375
1376 002662 012737 000003 001226
1377 002670 012737 003214 001216
1378 002676 012737 002744 001220
1379 002704 032737 020000 001510
1380 002712 001005
1381 002714 012737 013006 001214
1382 002722 000177 176266
1383 002726 104413
1384 002730 105037 013005
1385 002734 005037 013002
1386 002740 005037 013000
1387 002744 012737 000010 012776
1388 002752 013702 012776
1389 002756 105077 176400
1390 002762 105077 176402
1391 002766 012777 012154 176376
1392 002774 105277 176370
1393 003000 012777 000200 176364
1394 003006 105077 176350
1395 003012 113737 013005 013002
1396 003020 112777 000010 176342
1397 003026 013777 013004 176336
1398 003034 112777 000014 176326
1399 003042 012777 120000 176322
1400 003050 112777 000012 176312
1401 003056 012777 004012 176306
1402 003064 052777 010001 176266
1403 003072 013737 013004 015652
1404 003100 105137 015653
1405 003104 042777 000200 176260
1406 003112 006037 015652
1407 003116 013703 015652
1408 003122 042703 177577
1409 003126 050377 176240
1410 003132 005277 176234
1411 003136 005377 176230
1412 003142 005302
1413 003144 001357
1414 003146 005777 176206
1415 003152 100401
1416 003154 104002
1417 003156 017737 176176 015652
1418 003164 042737 170377 015652
1419 003172 005737 015652
1420 003176 001401
1421 003200 104002
1422
1423 003202 104401
1424

```

```

:TEST TO SEE IF EVERY CHARACTER FROM
:0 TO 377 CAN BE DETECTED IN CHARACTER
:DETECT ADDRESS ZERO.
:NOTE: SW09=1 WILL FREEZE ON CURRENT DATA.
:
: TEST 3
:*****
TST3: MOV #3,TSTNO
MOV #TST4,NEXT
MOV #15,LOCK
BIT #888BIT,DGSTAT ;DOES THIS DQ11 HAVE THE "BB" OPTION INSTALLED?
BNE 13$ ;BR IF YES
MOV #.EOP,RETURN ;GOTO END PASS
JMP @RETURN ;""
13$: MEMCLR ;CLEAR ALL THE DQ11
65: CLRB DETCAR+1 ;CLEAR THE CHARACTER STORAGE AREA
CLR GOCHAR ;SET FOR ERROR PRINTOUT
CLR ADDR ;SAME
15: MOV #8,COUNT ;EIGHT BITS FOR EIGHT SHIFTS.
25: MOV COUNT,R2 ;GET NUMBER OF SHIFTS PER CHAR.
CLRB @DQRC5H ;GET CHAR ADDR. ZERO
CLRB @DQREG ;GET RX BA PRI.
MOV #RXBUFF,@DQSEC ;LOAD IT
INCB @DQREG ;GET RX HC PRI.
MOV #200,@DQSEC ;LOAD IT
35: CLRB @DQRC5H ;SELECT CHARACTER DET REG 0
MOVB DETCAR+1,GOCHAR ;SELECT THE CHARACTER DET REGISTER.
MOVB #10,@DQREG ;LOAD THE CHARACTER TO BE DETECTED
MOV DETCAR,@DQSEC ;SELECT THE SEQUENCE REGISTER
MOVB #14,@DQREG ;SET SINGLE CHARACTER REC AND SET FLAG.
MOV #BIT15+BIT13,@DQSEC ;COMM
MOV #12,@DQREG ;SELECT EIGHT BITS TEST LOOP AND AUTO STEP
MOV #4012,@DQSEC ;SET RX ACTIVE AND RX GO
BIS #BIT12+BIT0,@DQRC5R ;MOV CHARACTER TO WORKING AREA
MOV DETCAR,TEMP ;COMPLIMENT DATA FOR USE ON THE BIT WINDOW
COMB TEMP+1 ;IF PREVIOUSLY SET; CLEAR THE BIT WINDOW.
45: BIC #BIT7,@DQSEC ;SHIFT OUT THE BIT OF DATA.
ROR TEMP ;STORE CHAR
MOV TEMP,R3 ;CLEAR ALL UNWANTED BITS
BIC #1C(BIT7),R3 ;PLACE DATA ON THE BIT WINDOW
BIS R3,@DQSEC ;CLOCK UP
INC @DQSEC ;CLOCK DOWN
DEC @DQSEC ;IS CHARACTER DONE YET
DEC R2 ;BR IF NOT DONE
BNE 45 ;WAS THE CHARACTER REALLY DETECTED?
TST @DQRC5R ;BR IF GOOD
BMI .+4 ;ERROR CHARACTER NOT DETECTED.
2 ;GET THE RECEIVER CSR.
MOV @DQRC5R,TEMP ;CLEAR ALL BUT THE CHARACTER DET. ADDR.
BIC #1C(7400),TEMP ;WAS THE CHAR DET. IN ADDR ZERO?
TST TEMP ;
BEQ .+4 ;
HLT 2 ;CHAR NOT DETECTED IN ADDR. ZERO..
:-----*LOCK*-----
: SCOPI ;IF SW09=1; THEN GOTO ADDRESS IN "LOCK".
:-----

```

E03

1425 003204 105237 013005
1426 003210 001260
1427 003212 104400
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440 003214 012737 000004 001226
1441 003222 012737 003514 001216
1442 003230 012737 003252 001220
1443 003236 005037 013004
1444 003242 005037 013002
1445 003246 005037 013000
1446 003252
1447 003252 012702 000020
1448 003256 105077 176106
1449 003262 012777 012154 176102
1450 003270 105277 176074
1451 003274 012777 000200 176070
1452 003302 105077 176054
1453 003306 112777 000010 176054
1454 003314 013777 013004 176050
1455 003322 112777 000014 176040
1456 003330 012777 050000 176034
1457 003336 112777 000012 176024
1458 003344 012777 000012 176020
1459 003352 052777 010001 176000
1460 003360 013737 013004 015652
1461 003366 013737 013004 013002
1462 003374 005137 015652
1463 003400 042777 000200 175764
1464 003406 000241
1465 003410 005037 001244
1466 003414 006037 015652
1467 003420 106037 001244
1468 003424 053777 001244 175740
1469 003432 005277 175734
1470 003436 005377 175730
1471 003442 005302
1472 003444 001355
1473 003446 005777 175706
1474 003452 100401
1475 003454 104002
1476 003456 017737 175676 015652
1477 003464 042737 170377 015652
1478 003472 005737 015652
1479 003476 001401
1480 003500 104002

INCB DETCAR+1 ; HAVE I HIT MY LIMIT YET?
BNE 25 ; NO RETURN WITH UPDATED CHAR.
SCOPE ; SCOPE TEST

TEST THAT CHARACTERS FROM
400 TO 177400 CAN BE DETECTED.
IN CHACTER DETECT ADDRESS ZERO.
NOTE: SMO9=1 WILL FREEZE ON CURRENT DATA.

TEST 4

↑ST4: MOV #4,TSTNO ;
MOV #TST5,NEXT ;
MOV #15,LOCK ;
CLR DETCAR ; SET CHAR TO ZERO
CLR GOCHAR ; SET FOR ERROR
CLR ADDR ; SAME
15: ;
25: MOV #16,R2 ; SET COUNT FOR 16 BIT CHARS
35: CLR #200REG ; SEL THE RX BA PRI.
MOV #RXBUFF,200SEC ; LOAD IT
INCB #200REG ; SEL THE RX WC PRI
MOV #200,200SEC ; LOAD IT
CLR #200RCSH ; GET CHAR ADD ZERO
MOV #10,200REG ; GET CHAR ADDRESS
MOV DETCAR,200SEC ; LOAD THE CHARACTER TO BE DETCETED
MOV #14,200REG ; GET THE SEQ REG.
MOV #BIT12+BIT14,200SEC ; LOAD DBL CHAR AND SET FLAG
MOV #12,200REG ; SEL MISC REG.
MOV #BIT1+BIT3,200SEC ; SET TEST LOOP AND AUTO/STEP
BIS #BIT12+BIT0,200RCSR ; SET RX ACTIVE AND GO
MOV DETCAR,TEMP ; GET DATA CHAR.
MOV DETCAR,GOCHAR ; FOR ERROR
COM TEMP ; PREPARE FOR BIT WINDOW
45: BIC #BIT7,200SEC ; ZERO BIT WINDOW
CLC ; CLEAR CARRY
CLR TEMP1 ;
ROR TEMP ;
ROR TEMP1 ;
BIS TEMP1,200SEC ; PLACE DATA ON BIT WINDOW
INC #200SEC ; CLOCK THE
DEC #200SEC ;
DEC R2 ; D011
BNE 45 ; IS ALL THE CHAR DONE?
TST #200RCSR ; BR IF NO
BMI .+4 ; DID THE FLAG SET?
HLT 2 ; BR IF YES
MOV #200RCSR,TEMP ; CHARACTER DET. FLAG NOT SET FOR DBL CHAR.
BIC #C<7400>,TEMP ; GET THE RECEIVER CSR.
TST TEMP ; CLEAR ALL BUT THE CHARACTER DET. ADDR.
BEQ .+4 ; WAS THE CHAR DET. IN ADDR ZERO?
HLT 2 ; CHAR NOT DETECTED IN ADDR. ZERO..

F03

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 31
 DZDQFD.P11 21-DEC-76 16:34 PROGRAM INITIALIZATION AND START UP.

```

1481 ;----- *LOCK* -----
1482 003502 104401 ; SCOPI ; IF SW09=1; THEN GOTO ADDRESS IN "LOCK".
1483 ;-----
1484 003504 105237 013005 INCB DETCAR+1 ; UPDATE THE DATA
1485 003510 001260 BNE IS ; ALL DONE?
1486 003512 104400 SCOPE ; SCOPE WHEN ALL DATA DONE.
1487
1488
1489
1490
1491 ; TEST THAT A CHARACTER CAN
1492 ; BE DETECTED IN ALL 16 CHARACTER
1493 ; DETECT ADDRESSES
1494
1495
1496 ; TEST THAT THE CHARACTER 255
1497 ; CAN BE DETECTED IN CHARACTER
1498 ; DETECT ADDRESS 00
1499
1500 ; TEST 5
1501 ; *****
1502 003514 012737 000005 001226 †ST5: MOV #5,TSTNC
1503 003522 012737 003544 001216 MOV #TST6,NEXT
1504 003530 012737 000000 013000 MOV #00,ADDR ; LOAD THE ADDRESS
1505 003536 004737 004314 JSR PC,CHK.ADD ; GO AND LOAD THE CHARACTER.
1506 003542 104400 SCOPE ; SCOPE THIS TEST
1507
1508
1509 ; TEST THAT THE CHARACTER 255
1510 ; CAN BE DETECTED IN CHARACTER
1511 ; DETECT ADDRESS 01
1512
1513 ; TEST 6
1514 ; *****
1515 003544 012737 000006 001226 †ST6: MOV #6,TSTNO
1516 003552 012737 003574 001216 MOV #TST7,NEXT
1517 003560 012737 000001 013000 MOV #01,ADDR ; LOAD THE ADDRESS
1518 003566 004737 004314 JSR PC,CHK.ADD ; GO AND LOAD THE CHARACTER.
1519 003572 104400 SCOPE ; SCOPE THIS TEST
1520
1521
1522 ; TEST THAT THE CHARACTER 255
1523 ; CAN BE DETECTED IN CHARACTER
1524 ; DETECT ADDRESS 02
1525
1526 ; TEST 7
1527 ; *****
1528 003574 012737 000007 001226 †ST7: MOV #7,TSTNO
1529 003602 012737 003624 001216 MOV #TST10,NEXT
1530 003610 012737 000002 013000 MOV #02,ADDR ; LOAD THE ADDRESS
1531 003616 004737 004314 JSR PC,CHK.ADD ; GO AND LOAD THE CHARACTER.
1532 003622 104400 SCOPE ; SCOPE THIS TEST
1533
1534
1535 ; TEST THAT THE CHARACTER 255
1536 ; CAN BE DETECTED IN CHARACTER
  
```

G03

```

1537                                     ;DETECT ADDRESS 03
1538
1539                                     ; TEST 10
1540                                     ;*****
1541 003624 012737 000010 001226 †ST10: MOV #10,TSTNO
1542 003632 012737 003654 001216      MOV #TST11,NEXT
1543 003640 012737 000003 013000      MOV #03,ADDR ;LOAD THE ADDRESS
1544 003646 004737 004314      JSR PC,CHK.ADD ;GO AND LOAD THE CHARACTER.
1545 003652 104400      SCOPE ;SCOPE THIS TEST
1546
1547
1548                                     ;TEST THAT THE CHARACTER 255
1549                                     ;CAN BE DETECTED IN CHARACTER
1550                                     ;DETECT ADDRESS 04
1551
1552                                     ; TEST 11
1553                                     ;*****
1554 003654 012737 000011 001226 †ST11: MOV #11,TSTNO
1555 003662 012737 003704 001216      MOV #TST12,NEXT
1556 003670 012737 000004 013000      MOV #04,ADDR ;LOAD THE ADDRESS
1557 003676 004737 004314      JSR PC,CHK.ADD ;GO AND LOAD THE CHARACTER.
1558 003702 104400      SCOPE ;SCOPE THIS TEST
1559
1560
1561                                     ;TEST THAT THE CHARACTER 255
1562                                     ;CAN BE DETECTED IN CHARACTER
1563                                     ;DETECT ADDRESS 05
1564
1565                                     ; TEST 12
1566                                     ;*****
1567 003704 012737 000012 001226 †ST12: MOV #12,TSTNO
1568 003712 012737 003734 001216      MOV #TST13,NEXT
1569 003720 012737 000005 013000      MOV #05,ADDR ;LOAD THE ADDRESS
1570 003726 004737 004314      JSR PC,CHK.ADD ;GO AND LOAD THE CHARACTER.
1571 003732 104400      SCOPE ;SCOPE THIS TEST
1572
1573
1574                                     ;TEST THAT THE CHARACTER 255
1575                                     ;CAN BE DETECTED IN CHARACTER
1576                                     ;DETECT ADDRESS 06
1577
1578                                     ; TEST 13
1579                                     ;*****
1580 003734 012737 000013 001226 †ST13: MOV #13,TSTNO
1581 003742 012737 003764 001216      MOV #TST14,NEXT
1582 003750 012737 000006 013000      MOV #06,ADDR ;LOAD THE ADDRESS
1583 003756 004737 004314      JSR PC,CHK.ADD ;GO AND LOAD THE CHARACTER.
1584 003762 104400      SCOPE ;SCOPE THIS TEST
1585
1586
1587                                     ;TEST THAT THE CHARACTER 255
1588                                     ;CAN BE DETECTED IN CHARACTER
1589                                     ;DETECT ADDRESS 07
1590
1591                                     ; TEST 14
1592                                     ;*****

```

H03

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 33
DZDQFD.P11 21-DEC-76 16:34 PROGRAM INITIALIZATION AND START UP.

1593 003764 012737 000014 001226 TST14: MOV #14,TSTNO
1594 003772 012737 004014 001216 MOV #TST15,NEXT
1595 004000 012737 000007 013000 MOV #07,ADDR ;LOAD THE ADDRESS
1596 004006 004737 004314 JSR PC,CHK.ADD ;GO AND LOAD THE CHARACTER.
1597 004012 104400 SCOPE ;SCOPE THIS TEST

1598
1599
1600 ;TEST THAT THE CHARACTER 255
1601 ;CAN BE DETECTED IN CHARACTER
1602 ;DETECT ADDRESS 10
1603
1604 ;

; TEST 15

1605
1606 004014 012737 000015 001226 TST15: MOV #15,TSTNO
1607 004022 012737 004044 001216 MOV #TST16,NEXT
1608 004030 012737 000010 013000 MOV #10,ADDR ;LOAD THE ADDRESS
1609 004036 004737 004314 JSR PC,CHK.ADD ;GO AND LOAD THE CHARACTER.
1610 004042 104400 SCOPE ;SCOPE THIS TEST

1611
1612 ;
1613 ;TEST THAT THE CHARACTER 255
1614 ;CAN BE DETECTED IN CHARACTER
1615 ;DETECT ADDRESS 11
1616 ;

; TEST 16

1617
1618
1619 004044 012737 000016 001226 TST16: MOV #16,TSTNO
1620 004052 012737 004074 001216 MOV #TST17,NEXT
1621 004060 012737 000011 013000 MOV #11,ADDR ;LOAD THE ADDRESS
1622 004066 004737 004314 JSR PC,CHK.ADD ;GO AND LOAD THE CHARACTER.
1623 004072 104400 SCOPE ;SCOPE THIS TEST

1624
1625 ;
1626 ;TEST THAT THE CHARACTER 255
1627 ;CAN BE DETECTED IN CHARACTER
1628 ;DETECT ADDRESS 12
1629 ;

; TEST 17

1630
1631
1632 004074 012737 000017 001226 TST17: MOV #17,TSTNO
1633 004102 012737 004124 001216 MOV #TST20,NEXT
1634 004110 012737 000012 013000 MOV #12,ADDR ;LOAD THE ADDRESS
1635 004116 004737 004314 JSR PC,CHK.ADD ;GO AND LOAD THE CHARACTER.
1636 004122 104400 SCOPE ;SCOPE THIS TEST

1637
1638 ;
1639 ;TEST THAT THE CHARACTER 255
1640 ;CAN BE DETECTED IN CHARACTER
1641 ;DETECT ADDRESS 13
1642 ;

; TEST 20

1643
1644
1645 004124 012737 000020 001226 TST20: MOV #20,TSTNO
1646 004132 012737 004154 001216 MOV #TST21,NEXT
1647 004140 012737 000013 013000 MOV #13,ADDR ;LOAD THE ADDRESS
1648 004146 004737 004314 JSR PC,CHK.ADD ;GO AND LOAD THE CHARACTER.

```

1649 004152 104400          SCOPE                ;SCOPE THIS TEST
1650
1651
1652          ;TEST THAT THE CHARACTER 255
1653          ;CAN BE DETECTED IN CHARACTER
1654          ;DETECT ADDRESS 14
1655
1656          ; TEST 21
1657          ;*****
1658 004154 012737 000021 001226  †ST21: MOV      #21,TSTNO
1659 004162 012737 004204 001216      MOV      #†ST22,NEXT
1660 004170 012737 000014 013000      MOV      #14,ADDR          ;LOAD THE ADDRESS
1661 004176 004737 004314          JSR      PC,CHK.ADD        ;GO AND LOAD THE CHARACTER.
1662 004202 104400          SCOPE                ;SCOPE THIS TEST
1663
1664          ;TEST THAT THE CHARACTER 255
1665          ;CAN BE DETECTED IN CHARACTER
1666          ;DETECT ADDRESS 15
1667
1668          ; TEST 22
1669          ;*****
1670          ;*****
1671 004204 012737 000022 001226  †ST22: MOV      #22,TSTNO
1672 004212 012737 004234 001216      MOV      #†ST23,NEXT
1673 004220 012737 000015 013000      MOV      #15,ADDR          ;LOAD THE ADDRESS
1674 004226 004737 004314          JSR      PC,CHK.ADD        ;GO AND LOAD THE CHARACTER.
1675 004232 104400          SCOPE                ;SCOPE THIS TEST
1676
1677          ;TEST THAT THE CHARACTER 255
1678          ;CAN BE DETECTED IN CHARACTER
1679          ;DETECT ADDRESS 16
1680
1681          ; TEST 23
1682          ;*****
1683          ;*****
1684 004234 012737 000023 001226  †ST23: MOV      #23,TSTNO
1685 004242 012737 004264 001216      MOV      #†ST24,NEXT
1686 004250 012737 000016 013000      MOV      #16,ADDR          ;LOAD THE ADDRESS
1687 004256 004737 004314          JSR      PC,CHK.ADD        ;GO AND LOAD THE CHARACTER.
1688 004262 104400          SCOPE                ;SCOPE THIS TEST
1689
1690          ;TEST THAT THE CHARACTER 255
1691          ;CAN BE DETECTED IN CHARACTER
1692          ;DETECT ADDRESS 17
1693
1694          ; TEST 24
1695          ;*****
1696          ;*****
1697 004264 012737 000024 001226  †ST24: MOV      #24,TSTNO
1698 004272 012737 004562 001216      MOV      #†ST25,NEXT
1699 004300 012737 000017 013000      MOV      #17,ADDR          ;LOAD THE ADDRESS
1700 004306 004737 004314          JSR      PC,CHK.ADD        ;GO AND LOAD THE CHARACTER.
1701 004312 104400          SCOPE                ;SCOPE THIS TEST
1702
1703          CHK.ADD:
1704 004314 104413          MEMCLR                ;CLEAR ALL THE DQ11 REGISTERS.
    
```

J03

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 35
 DZDQFD.P11 21-DEC-76 16:34

PROGRAM INITIALIZATION AND START UP.

1705	004316	113777	013000	175036		MOVB	ADDR, 200RCSH	LOAD THE CHAR DET ADDRESS TO BE USED.
1706	004324	005037	013004			CLR	DETCAR	CLEAR WORKING LOC.
1707	004330	112737	000255	013005		MOVB	#255, DETCAR+1	LOAD THE CHARACTER TO BE DETECTED IN WORKING AREA.
1708	004336	012737	000255	013002		MOV	#255, GDCHAR	LOAD FOR ERROR PRINTOUT
1709	004344	013702	012776		15:	MOV	COUNT, R2	EIGHT SHIFTS FOR EIGHT BITS (NO VRC)
1710	004350	105077	175014		25:	CLRB	200REG	SEL THE RX BA PRI.
1711	004354	012777	012154	175010		MOV	#RX OFF, 200SEC	LOAD IT
1712	004362	105277	175002			INCB	200REG	SEL THE RX MC PRI.
1713	004376	012777	177777	174776		MOV	#-1, 200SEC	SET FOR ONE CHAR.
1714	004374	112777	000010	174766		MOVB	#10, 200REG	SELECT THE CHARACTER DET REGISTER.
1715	004402	013777	013004	174762		MOV	DETCAR, 200SEC	SET THE CHARACTER TO BE DETECTED INTO 0011 CHAR DET REG
1716	004410	112777	000014	174752		MOVB	#14, 200REG	SELECT THE SEQUENCE REGISTER.
1717	004416	012777	120000	174746		MOV	#BIT15+BIT13, 200SEC	SET SINGLE CHAR DET AND SINGLE CHAR DET FLAG SE
1718	004424	112777	000012	174736		MOVB	#12, 200REG	SELECT THE MISC REGISTER.
1719	004432	012777	004012	174732		MOV	#4012, 200SEC	SET EIGHT BITS TEST LOOP AND AUTO STEP
1720	004440	052777	010001	174712		BIS	#BIT12+BIT10, 200RCSR	SET RX ACTIVE AND RX GO.
1721	004446	013737	013004	015652		MOV	DETCAR, TEMP	MOVE THE CHAR TO BE DET TO WORKING AREA
1722	004454	105137	015653			COMB	TEMP+1	COMPLEMENT CHAR FOR USE ON BIT WINDOW.
1723	004460	042777	000200	174704	35:	BIC	#BIT7, 200SEC	IF BIT WINDOW SET ON LAST PASS CLEAR IT!
1724	004466	006037	015652			ROR	TEMP	SHIFT OUT BIT OF DATA.
1725	004472	013703	015652			MOV	TEMP, R3	SAVE IT
1726	004476	042703	177577			BIC	#1C<BIT7>, R3	CLEAR ALL UNWANTED BITS
1727	004502	050377	174664			BIS	R3, 200SEC	PLACE DATA ON BIT WINDOW.
1728	004506	005277	174660			INC	200SEC	CLOCK UP
1729	004512	005377	174654			DEC	200SEC	CLOCK DOWN
1730	004516	005302				DEC	R2	IS CHARACTER DONE YET?
1731	004520	001357				BNE	35	BR IF NOT DONE
1732	004522	005777	174632			TST	200RCSR	WAS CHAR REALLY DETECTED?
1733	004526	100401				BMI	.+4	BR IF GOOD
1734	004530	104002				HLT	2	CHARACTER DETECT FAILED.
1735	004532	017737	174622	015652		MOV	200RCSR, TEMP	GET THE RECEIVER CSR.
1736	004540	042737	170377	015652		BIC	#1C<7400>, TEMP	CLEAR ALL BUT CHARACTER DET. ADDR.
1737	004546	123737	013000	015653		CMPB	ADDR, TEMP+1	WAS THE CHARACTER REALLY DETECTED IN ADDRESS SA ??
1738								
1739	004554	001401				BEG	.+4	
1740	004556	104002				HLT	2	WRONG ADDRESS.
1741	004560	000207				RTS	PC	
1742								
1743								TEST OF RECEIVER AND TRANSMITTER "SET T"
1744								TEST OF BIT ONE OF SEQUENCE REGISTER.
1745								THIS TEST WILL "SET T" AND THEN WILL
1746								SEND A CHAR WHICH WILL "SET DONE CLEAR GO". IF
1747								REALLY IN TRANSPARENT MODE THE CHAR WILL NOT BE DETECTED
1748								AND THE WORD COUNTS WILL GOTO ZERO.
1749								
1750								TEST 25
1751								*****
1752	004562	012737	000025	001226		TST25:	MOV #25, TSTNO	
1753	004570	012737	004722	001216			MOV #TST26, NEXT	
1754								
1755	004576	004737	011024		25:	JSR	PC, SET UP	SET UP ALL NECESSARY FOR TEST.
1756	004602	012777	040002	174562		MOV	#BIT14+BIT1, 200SEC	
1757								SET DEL. CHAR AND SET T
1758	004610	105377	174546			DECB	200RCSH	SELECT ADD 16 (8)
1759	004614	112777	000010	174546		MOVB	#10, 200REG	SELECT CHAR DET. ADDRESS
1760	004622	012777	164400	174542		MOV	#351#400, 200SEC	LOAD THE CHARACTER. SET DONE CLEAR GO.

K03

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 36
 DZDQFD.P11 21-DEC-76 16:34 PROGRAM INITIALIZATION AND START UP.

1761	004630	112777	000014	174532	MOV B #14, 200REG	:SELECT THE SEQ REGISTER.
1762	004636	012777	100200	174526	MOV #BIT15+BIT7, 200SEC	:SET FUNCTION. CLEAR GO SET DONE.
1763						:PREPARE TX BUFFER
1764	004644	012700	011752		MOV #TXBUFF, R0	:LOAD THE BUFFER WITH DATA
1765	004650	012720			MOV (PC)+, (R0)+	:DATA
1766	004652	000	350		.BYTE 000, 350	:LOAD THE BUFFER WITH DATA
1767	004654	012720			MOV (PC)+, (R0)+	:DATA
1768	004656	351	200		.BYTE 351, 200	:SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
1769	004660	004737	011556		JSR PC, ENABLE	:GET THE RX WC PRI.
1770	004664	112777	000001	174476	MOV B #1, 200REG	:RX WC PRI. SHOULD =0
1771	004672	005777	174474		TST 200SEC	:BR IF RX WC PRI =0
1772	004676	001401			BEQ +4	:RX PRI WC NOT =0
1773	004700	104003			HLT 3	:GET TX WC PRI.
1774	004702	112777	000003	174460	MOV B #3, 200REG	:TX WC PRI SHOULD =0
1775	004710	005777	174456		TST 200SEC	:BR IF TX WC =0
1776	004714	001401			BEQ +4	:TX WC PRI NOT =0
1777	004716	104003			HLT 3	
1778	004720	104400			SCOPE	
1779						
1780						
1781						
1782						:TEST OF RECEIVER "CLEAR T"
1783						:TEST OF BIT TWO OF SEQUENCE REGISTER.
1784						:THIS TEST WILL ENTER BOTH THE RECEIVER AND
1785						:TRANSMITTER INTO TRANSPARENCY; THEN A CHARACTER
1786						:WHICH SAYS "CLEAR RX T" WILL BE SENT FOLLOWED
1787						:BY A CHARACTER WHICH SAYS "SET RX CHAR FLAG".
1788						:THE TEST THEN CHECKS THAT THE CHARACTER FLAG IS SET
1789						:WHICH MEANS THAT CHARACTER WAS DETECTED.
1790						
1791						
1792						:TEST 26
1793	004722	012737	000026	001226	TST26: MOV #26, TSTNO	:*****
1794	004730	012737	005072	001216	MOV #TST27, NEXT	
1795	004736	004737	011024		JSR PC, SET.UP	:SET UP ALL NECESSARY FOR TEST.
1796	004742	012777	040002	174422	MOV #BIT14+BIT1, 200SEC	:SET FUNCTION: DBL CHAR AND SET T
1797						:SELECT CHAR ADDRESS 16 (8)
1798	004750	105377	174406		DECB 200RCSH	:SELECT CHAR DET ADDRESS.
1799	004754	112777	000010	174406	MOV B #10, 200REG	:LOAD CHARACTER
1800	004762	012777	165000	174402	MOV #352*400, 200SEC	:SELECT THE SEQ REGISTER.
1801	004770	112777	000014	174372	MOV B #14, 200REG	
1802	004776	012777	040004	174366	MOV #BIT14+BIT2, 200SEC	:SET FOR DBL CHAR AND CLEAR T
1803						:GET NEXT ADDR
1804	005004	105377	174352		DECB 200RCSH	:GET NEXT ADDRESS
1805						:SELECT CHAR DET ADDRESS
1806	005010	112777	000010	174352	MOV B #10, 200REG	:LOAD CHARACTER.
1807	005016	012777	166400	174346	MOV #355*400, 200SEC	:SELECT THE SEQ REGISTER.
1808	005024	112777	000014	174336	MOV B #14, 200REG	
1809	005032	012777	120000	174332	MOV #BIT15+BIT13, 200SEC	:SET FOR SINGLE CHAR AND SET FLAG.
1810						:GET POINTER
1811	005040	012700	011752		MOV #TXBUFF, R0	:LOAD THE BUFFER WITH DATA
1812	005044	012720			MOV (PC)+, (R0)+	:DATA
1813	005046	350	351		.BYTE 350, 351	:LOAD THE BUFFER WITH DATA
1814	005050	012720			MOV (PC)+, (R0)+	:DATA
1815	005052	352	355		.BYTE 352, 355	:SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
1816	005054	004737	011556		JSR PC, ENABLE	

L03

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 37
 DZDQFD.P11 21-DEC-76 16:34 PROGRAM INITIALIZATION AND START UP.

1817	005060	005777	174274		TST	200RCSR	;CHECK CHAR DET FLAG
1818	005064	100401			BMI	.+4	;BR IS SET
1819	005066	104004			HLT	4	;CHARACTER DET FLAG NOT SET
1820	005070	104400			SCOPE		;SCOPE THIS TEST
1821							
1822							
1823							
1824							
1825							
1826							
1827							
1828							
1829							
1830							
1831							
1832							
1833							
1834							
1835	005072	012737	000027	001226	TEST 27		
1836	005100	012737	005350	001216	*****		
1837	005106	004737	011024		TST27: MOV	#27,ISTNO	
1838	005112	012777	100010	174252	MOV	#TST30,NEXT	
1839					JSR	PC,SET,UP	;SET UP ALL NECESSARY FOR TEST.
1840	005120	112777	000017	174242	MOV	#BIT15+BIT3,200SEC	
1841	005126	012777	000200	174236	MOV	#17,200REG	;SET SNGL CHAR AND BCC/START CLEAR
1842	005134	112777	000003	174226	MOV	#200,200SEC	;SEL THE POLY REG.
1843	005142	012777	177576	174222	MOV	#3,200REG	;SET FOR LRC 8
1844	005150	012700	011752		MOV	#-202,200SEC	;SEL THE TX WC PRI.
1845	005154	105020			MOV	#TXBUFF,RO	;SET FOR 202 (8)
1846	015156	022700	012153		MOV	(RO)+	;GET TX BUFFER
1847	015162	001374			15: CLR	(RO)+	;CLEAR ALL THE TX BUFFER
1848	015164	012700	011752		CMP	#TXBUFF+201,RO	;ALL CLEAR?
1849	005170	012720			BNE	15	;BR IF NO
1850	015172	350	225		MOV	#TXBUFF,RO	;GET TX BUFFER
1851	015174	012737	000225	012774	MOV	(PC)+,(RO)+	;LOAD THE BUFFER WITH DATA
1852	015202	012737	000010	012776	.BYTE	350,225	;DATA
1853	005210	105077	174154		MOV	#225,CHAR	;SET EXPECTED BCC CHAR.
1854	015214	012777	012154	174150	MOV	#8,COUNT	;SET FOR 8 BITS
1855	015222	112777	000001	174140	CLR	200REG	;SEL REC PRIMARY
1856	015230	012777	177775	174134	MOV	#RXE FF,200SEC	;SET WITH START ADRS
1857	015236	112777	000004	174124	MOV	#1,200REG	;SEL REC CHAR COUNT
1858	015244	012777	012174	174120	MOV	#-3,200SEC	;SET CHAR COUNT
1859	015252	112777	000005	174110	MOV	#RXE F+20,200SEC	;SEL REC SECONDARY
1860	005260	012777	177577	174104	MOV	#5,200REG	;SET WITH SEC ADRS
1861	005266	004737	011632		MOV	#-201,200SEC	;SEL REC CHAR COUNT
1862	005272	013703	012776		JSR	PC,NEWENA	;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
1863	005276	000241			MOV	COUNT,R3	;GET CHAR SIZE
1864	015270	105100			65: CLC		
1865	015272	005500			ROLB	RO	;SHIFT RX BCC IMAGE
1866	005304	023700	012774		75: ADC	RO	;PICK UP CARRY
1867	015310	001403			CMP	CHAR,RO	;CHECK BCC
1868	015312	005303			BEQ	35	;BCC OK!
1869	005314	001370			DEC	R3	;ALL SHIFTS DONE?
1870	005316	104005			BNE	65	;BR IF NO.
1871	005320	013703	012776		HLT	5	;RX BCC HAS WRONG DATA.
1872	005324	000241			35: MOV	COUNT,R3	;SAVE COUNTER
					85: CLC		

M03

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 38
DZDQFD.P11 21-DEC-76 16:34

PROGRAM INITIALIZATION AND START UP.

1873	005326	106101		ROLB	R1	;SHIFT TX BCC IMAGE
1874	005330	005501		ADC	R1	;PICK UP CARRY
1875	005332	023701	012774	CMP	CHAR,R1	;IS BCC OK?
1876	005336	001403		BEQ	SS	;BR IF OK
1877	005340	005303		DEC	R3	;ALL SHIFTS DONE?
1878	005342	001370		BNE	BS	;BR IF NO
1879	005344	104005		HLT	S	;TX BCC HAS WRONG DATA.
1880	005346	104400		SS:	SCOPE	;SCOPE THE TEST

N03

```

1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892 005350 012737 000030 001226
1893 005356 012737 005532 001216
1894 005364 004737 011024
1895 005370 012777 100100 173774
1896
1897 005376 112737 000350 011754
1898 005404 004737 005446
1899 005410 012737 005416 001220
1900 005416 004737 011212
1901 005422 012777 040100 173742
1902
1903 005430 012737 003721 012364
1904 005436 004737 005446
1905
1906 005442 104401
1907
1908 005444 104400
1909 005446 005277 173706
1910 005452 005277 173706
1911 005456 105777 173702
1912 005462 100375
1913 005464 021616
1914 005466 032777 010000 173664
1915 005474 001401
1916 005476 104006
1917 005500 112777 000001 173662
1918 005506 005777 173660
1919 005512 001001
1920 005514 104006
1921 005516 122777 000001 173634
1922 005524 001401
1923 005526 104006
1924 005530 000207
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
  
```

```

: TEST OF RECEIVER "CLEAR ACTIVE"
: TEST OF BIT SIX OF SEQUENCE REGISTER.
: THIS TEST WILL SEND A CHARACTER WHICH SAYS "CLEAR RX ACTIVE"
: THE PROGRAM WAITS FOR TX DONE THEN LOOKS AT RX ACTIVE
: WHICH SHOULD BE EQUAL TO ZERO; THEN THE PROGRAM LOOKS
: FOR RX GO TO BE SET AND RX PRI DONE TO BE CLEAR.
: ALSO THE RX WC PRI SHOULD BE NOT EQUAL TO ZERO.
  
```

```

: TEST 30
: *****
TST30: MOV #30, TSTNO
MOV #TST31, NEXT
JSR PC, SET.UP ; SET UP ALL NECESSARY FOR TEST.
MOV #BIT15+BIT6, 200SEC ; SET SNGL CHAR AND CLEAR ACTV
MOV #350, TXBUFF+2 ; SET DATA IN TX BUFFER
JSR PC, X.ABG ; GO AND WORK THE DQ11
MOV #1$, LOCK ; SET FOR RETURN IF SW09=1
JSR PC, EXT.UP ; SET THING UP FOR DOUBLE CHAR. (16 BITS)
MOV #BIT14+BIT6, 200SEC
MOV #3721, XTXBUF+2 ; SET DBL CHAR AND CLEAR ACTV
JSR PC, X.ABG ; LOAD THE DATA
; WORK DQ11
: ----- *LOCK* -----
SCOPI ; IF SW09=1; THEN GOTO ADDRESS IN "LOCK".
: -----
X.ABG: SCOPE ; SCOPE THIS TEST
INC 200RCSR ; SET RX GO.
INC 200TCSR ; SET TX GO
TSTB 200TCSR ; HANG HERE FOR TX PRI DONE!!
BPL -4 ; BR IF NOT DONE.
CMP (SP), (SP) ; WAST TIME!
BIT #BIT12, 200RCSR ; IS RX ACTIVE CLEARED?
BEQ +4 ; BR IF YES
HLT 6 ; RX ACTIVE NOT CLEARED
MOVB #1, 200REG ; GET THE RX WC PRI.
TST 200SEC ; IT SHOULD BE NON-ZERO!!
BNE +4 ; BR IF OK
HLT 6 ; RX PRI WC =0
CMPB #001, 200RCSR ; GO SHOULD BE SET AND DONE NPT SET.
BEQ +4 ; BR IF OK!
HLT 6 ; LOW BYTE RXCSR NOT =001
RTS PC
:
  
```

```

: TEST OF RECEIVER AND TRANSMITTER "CLEAR GO/SET DONE"
: TEST OF BIT SEVEN OF SEQUENCE REGISTER.
: CHARACTER "SET DONE/CLEAR GO" IS SENT AND IS DETCETED
: BY BOTH THE TX AND RX. WHEN RX DONE SETS; THE PROGRAM
: VERIFIES THAT BOTH THE TX AND RX WC (PRI) ARE NOT
: EQUAL TO ZERO AND THAT PRI DONE SET, GO IS CLEARED,
: AND PRI/SEC BIT IS CLEARED.
  
```

```

: TEST 31
: *****
  
```

```

1937 005532 012737 000031 001226 TST31: MOV #31,TSTNO
1938 005540 012737 005726 001216 MOV #TST32,NEXT
1939 005546 004737 011024 JSR PC,SET.UP ;SET UP ALL NECESSARY FOR TEST.
1940 005552 012777 100200 173612 MOV #BIT15+BIT7,200SEC
1941 005560 012700 011754 MOV #TXBUFF+2,R0 ;SET SNGL CHAR AND SET DONE CLEAR GO
1942 005564 012710 011754 MOV (PC)+,(R0) ;SET TX BUFFER
1943 005566 350 352 .BYTE 350,352 ;LOAD WITH DATA
1944 005570 004737 005640 JSR PC,X.ABF ;DATA
1945 005574 012737 005602 001220 MOV #18,LOCK ;WORK DQ11
1946 005602 004737 011212 173556 15: JSR PC,EXT.UP ;SET FOR RETURN IF SW09=1
1947 005606 012777 040200 173556 MOV #BIT14+BIT7,200SEC ;SET THING UP FOR DOUBLE CHAR.(16 BITS)
1948 005614 012737 003721 012364 MOV #3721,XTXBUF+2 ;SET DBL CHAR AND SET DONE CLEAR GO
1949 005622 012737 012525 012366 MOV #12525,XTXBUF+4 ;LOAD DATA
1950 005630 004737 005640 JSR PC,X.ABF ;SAME
1951 ;-----*LOCK*----- ;TURN ON DQ11
1952 SCOPE1 ;IF SW09=1; THEN GOTO ADDRESS IN "LOCK".
1953 ;-----*SCOPE*----- ;SCOPE THIS TEST.
1954 SCOPE
1955 X.ABF: JSR PC,ENABLE ;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
1956 MOV #1,200REG ;SEL RX MC PRI
1957 TST 200SEC ;IT SB NON-ZERO.
1958 BNE +4 ;BR IF OK
1959 HLT ? ;RX PRI MC =0
1960 MOV #3,200REG ;SEL THE TX MC PRI
1961 TST 200SEC ;IT SB NON-ZERO
1962 BNE +4 ;BR IF OK
1963 HLT ? ;TX MC PRI IS =0
1964 CMPB #200,200RCR ;DONE=1; P/S=0; GO=0?
1965 BEQ +4 ;BR IF OK
1966 HLT ? ;RX CSR NOT =200 (PRI DONE)
1967 CMPB #200,200TCSR ;DONE=1; P/S=0; GO=0
1968 BEQ +4 ;BR IF OK
1969 HLT ? ;TX PRI DONE SET? (TX CSR=200)
1970 RTS PC
1971 ;
1972 ;TEST OF RECEIVER "CHARACTER STRIP"
1973 ;TEST OF BIT EIGHT OF SEQUENCE REGISTER.
1974 ;THE CHARACTER THAT IS SENT AS "CHARACTER STRIP" IS
1975 ;LOOKED FOR IN THE RX BUFFER; IF IT IS NOT FOUND IT
1976 ;IS ASSUMED THAT THE CHARACTER WAS INDEED "STRIPPED".
1977 ;
1978 ;TEST 32
1979 ;*****
1980 TST32: MOV #32,TSTNO
1981 MOV #TST33,NEXT
1982 JSR PC,SET.UP ;SET UP ALL NECESSARY FOR TEST.
1983 MOV #BIT15+BIT8,200SEC
1984 MOV #TXBUFF+2,R0 ;SET SNGL CHAR AND CHAR STRIP.
1985 MOV (PC)+,(R0) ;SET POINTER
1986 .BYTE 350,321 ;LOAD THE DATA
1987 JSR PC,ENABLE ;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
    
```



```

2049 006212 012737 003721 012364      MOV      #3721,XTXBUF+2 ;LOAD DATA
2050 006220 012737 054321 012366      MOV      #54321,XTXBUF+4 ;SAME
2051 006226 004737 011556                JSR      PC,ENABLE ;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
2052 006232 012700 012154                MOV      #RXBUF,R0 ;GET POINTER
2053 006236 012701 00C010                MOV      #10,R1 ;GET CHAR COUNTER
2054 012242 022720 177777      3S:    CMP      #177777,(R0)+ ;TX PAD IN BUFFER?
2055 012246 001001                BNE     .+4 ;BR IF NO
2056 006250 104011                HLT     11 ;PAD CHARACTER IS IN BUFFER.
2057 006252 005301                DEC     R1 ;ALL CHARS DONE?
2058 006254 001372                BNE     3S ;BR IF NO.
2059 006256 104400      4S:    SCOPE ;
2060
2061
2062
2063
2064 ;TEST OF "BCC EXCLUDE"
2065 ;TEST OF BIT 11 OF SEQUENCE REGISTER
2066 ;"BCC EXCLUDE" IS EXERCIZED ON BOTH THE RX AND TX TOGETHER
2067 ;THE BCC IS TURNED ON AND THEN A CHARACTER IS EXCLUDED
2068 ;FROM THE BCC; WHEN DONE COMES UP THE BCC'S OF BOTH
2069 ;THE RX AND TX ARE SHIFTED AROUND TO SEE IF THE
2070 ;CHARACTER WAS REALLY EXCLUDED.
2071
2072 ; TEST 34
2073 ;*****
2074 006260 012737 000034 001226 1S:    MOV      #34,TSTNO
2075 012266 012737 006600 001216      MOV      #TST35,NEXT
2076 012274 004737 011024                JSR      PC,SETUP ;SET UP ALL NECESSARY FOR TEST.
2077 006300 012777 100010 173064      MOV      #BIT15+BIT3,200SEC
2078 ;SET SNGL CHAR AND BCC START CLEAR
2079 006306 105377 173050      DECB    200RCSH ;SEL CHAR ADD 16(8)
2080 006312 112777 00C010 173050      MOV     #10,200REG ;GET CHAR DET ADDRESS
2081 012270 012777 170400 173044      MOV     #361+400,200SEC ;LOAD CHAR
2082 012276 112777 000014 173034      MOV     #14,200REG ;SEL SEQ REG
2083 006334 012777 104000 173030      MOV     #BIT15+BIT11,200SEC
2084 ;SET SNGL CHAR AND BCC EXCLUDE
2085 006342 112777 000017 173020      MOV     #17,200REG ;SEL POLY REG.
2086 006350 012777 000200 173014      MOV     #200,200SEC ;SET LRC 8
2087 006356 112777 000003 173004      MOV     #3,200REG ;SEL TX WC PRI.
2088 012284 012777 177576 173000      MOV     #-202,200SEC ;SET BIG NUMBER
2089 006372 012700 011752                MOV     #TXBUF,R0 ;SET POINTER
2090 006376 105020                CLR     (R0)+ ;LOAD DATA 000
2091 012400 022700 012153      1S:    CMP     #TXBUF+201,R0 ;CLEAR BUFFER!!
2092 006404 001374                BNE     1S ;BR IF NOT ALL CLEAR.
2093 006406 012700 011752                MOV     #TXBUF,R0 ;SET POINTER
2094 006412 012710                MOV     (PC)+,(R0) ;LOAD THE
2095 006414                .BYTE 350,107 ;DATA
2096 006416 112737 000361 011754      MOV     #361,XTXBUF+2 ;INTO
2097 006424 012737 000010 012776      MOV     #8,COUNT ;THE TX BUFFER!
2098 006432 012737 000107 012774      MOV     #107,CHAR ;SET FOR 8 BITS AND 107 AS THE CHAR IN BCC.
2099 006440 105077 172724                CLR     200REG ;SEL REC PRIMARY
2100 006444 012777 012154 172720      MOV     #RXBUF,200SEC ;SET WITH START ADRS
2101 006452 112777 000001 172710      MOV     #1,200REG ;SEL REC CHAR COUNT
2102 012460 012777 177775 172704      MOV     #-3,200SEC ;SET CHAR COUNT
2103 006466 112777 000004 172674      MOV     #4,200REG ;SET REC SECONDARY
2104 006474 012777 012174 172670      MOV     #RXBUF+20,200SEC ;SET WITH SEC ADRS

```

E04

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 43
 DZDQFD.P11 21-DEC-76 16:34 PROGRAM INITIALIZATION AND START UP.

2105	006502	112777	000005	172660	MOV B #5, 200REG	:SEL CHAR COUNT
2106	01510	012777	177577	172654	MOV #201, 200SEC	:SET CHAR COUNT
2107	016	004737	011632		JSR PC, NEWENA	:SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
2108	012	013703	012776		MOV COUNT, R3	:SAVE COUNTER
2109	016	000241			4\$: CLC	
2110	01J	106100			ROL B R0	:SHIFT RX BCC IMAGE
2111	012	005500			5\$: ADC R0	:SAVE CARRY
2112	006534	023700	012774		CMP CHAR, R0	:BCC OK?
2113	01540	001403			BEQ 3\$:BR IF OK
2114	01542	005303			DEC R3	:ALL SHIFTS DONE?
2115	006544	001370			BNE 4\$:BR IF NO
2116	01546	104012			HLT 12	:RX BCC HAS WRONG DATA.
2117	0170	013703	012776		3\$: MOV COUNT, R3	
2118	006554	000241			6\$: CLC	:CLEAR CARRY
2119	0176	106101			ROL B R1	:SHIFT TX BCC IMAGE
2120	0160	005501			ADC R1	:PICK UP CARRY
2121	006562	023701	012774		CMP CHAR, R1	:BCC OK?
2122	006566	001403			BEQ 7\$:BR IF OK
2123	006570	005303			DEC R3	:ALL SHIFTS DONE?
2124	006572	001370			BNE 6\$:BR IF NO
2125	006574	104012			HLT 12	:TX BCC HAS WRONG DATA.
2126	006576	104400			7\$: SCOPE	:SCOPE THIS TEST
2127						
2128						
2129						
2130						
2131						
2132						
2133						
2134						
2135						
2136						
2137						
2138	006600	012737	000035	001226	TEST 35	
2139	006606	012737	007054	001216	*****	
2140	006614	004737	011024		TST35: MOV #35, TSTNO	
2141	006620	012777	040002	172544	MOV #TST36, NEXT	
2142					JSR PC, SET UP	:SET UP ALL NECESSARY FOR TEST.
2143	006626	105377	172530		MOV #BIT14+BIT1, 200SEC	
2144	01632	112777	000010	172530	DEC B 200RCSH	:SET DEL CHAR AND SET T
2145	01540	013700	001372		MOV #10, 200REG	:SEL CHAR ADD 16
2146	01544	012710			MOV 200SEC, R0	:GET CHAR DET REG
2147	01646	352	352		MOV (PC)+, (R0)	:LOAD THE REGISTER
2148	01550	112777	000014	172512	.BYTE 352, 352	:WITH THIS DATA
2149	006656	012777	101000	172506	MOV #14, 200REG	:SEL THE SEQ REG.
2150					MOV #BIT15+BIT9, 200SEC	
2151	006664	105377	172472		DEC B 200RCSH	:SET SNGL CHAR AND DLE
2152	006670	112777	000010	172472	MOV #10, 200REG	:SEL CHAR ADD 15(8)
2153	006676	012777	166000	172466	MOV #354*400, 200SEC	:GET CHAR ADDRESS
2154	006704	112777	000014	172456	MOV #14, 200REG	:LOAD WITH DATA
2155	006712	012777	100200	172452	MOV #BIT15+BIT7, 200SEC	:SEL THE SEQ REGISTER
2156						
2157	006720	112777	000003	172442	MOV B #3, 200REG	:SET SNGL CHAR AND SET DONE CLEAR GO
2158	006726	012777	177772	172436	MOV #6, 200SEC	:SEL THE TX WC PRI.
2159	006734	112737	000350	011753	MOV #350, TXBUFF+1	:SET FOR 6 CHARS
2160	006742	112737	000357	011756	MOV #357, TXBUFF+4	:LOAD DATA

F04

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 44
 DZDQFD.P11 21-DEC-76 16:34 PROGRAM INITIALIZATION AND START UP.

2161	006750	112737	000354	012012	MOV	#354, TXBUFF+40	;EXIT T SHOULD SEND DLE TTIS SHOULD SET DONE THEN!!
2162	006756	112777	000006	172404	MOV	#6, 200REG	;SEL TX BA SEC.
2163	006764	012777	012012	172400	MOV	#TXBUFF+40, 200SEC	
2164	006772	105277	172372		INCB	200REG	;LOAD BA AND SEL WC SEC.
2165	006776	052777	030000	172362	BIS	#BIT13+BIT12, 200ERR	
2166							;SET WRITE ENABLE AND EXIT T
2167	007004	012777	177770	172360	MOV	#-10, 200SEC	;SET SEC. FOR 10(8) CHARS
2168	007012	004737	011556		JSR	PC, ENABLE	;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
2169	007016	112777	000007	172344	MOV	#7, 200REG	;SEL THE TX WC SEC.
2170	007024	005777	172342		TST	200SEC	;SHOULD BE NON-ZERO.
2171	007030	001001			BNE	.+4	;BR IF OK.
2172	007032	104003			HLT	3	;TX DID NOT DET "SET DONE CLEAR GO"
2173	007034	112777	000001	172326	MOV	#1, 200REG	;SEL THE RX WC PRI.
2174	007042	005777	172324		TST	200SEC	;RX WC PRI S/B NOT=0
2175	007046	001001			BNE	.+4	;BR IF OK
2176	007050	104003			HLT	3	;RX NOT DET CHAR SET DONE CLEAR GO.
2177	007052	104400			SCOPE		;SCOPE THE TEST.
2178							
2179							
2180							
2181							
2182							
2183							;TEST THAT THE TRANSMITTER WILL EXIT T
2184							;WHEN ENTERED BY "SET T"
2185							
2186							
2187							
2188	007054	012737	000036	001226			
2189	007062	012737	007260	001216			
2190	007070	004737	011024				
2191	007074	012777	040002	172270			
2192							
2193	007102	105377	172254				
2194	007106	112777	000010	172254			
2195	007114	012777	166000	172250			
2196	007122	112777	000014	172240			
2197	007130	012777	100200	172234			
2198							
2199	007136	112777	000006	172224			
2200	007144	012777	012012	172220			
2201	007152	105277	172212				
2202	007156	052777	030000	172202			
2203							
2204	007164	012777	177770	172200			
2205	007172	112737	000350	011755			
2206	007200	112737	000354	012013			
2207	007206	005277	172152				
2208	007212	032777	000100	172144	15:		
2209	007220	001774					
2210	007222	112777	000003	172140			
2211	007230	005777	172136				
2212	007234	001401					
2213	007236	104003					
2214	007240	112777	000007	172122			
2215	007246	005777	172120				
2216	007252	001001					

```

;TEST THAT THE TRANSMITTER WILL EXIT T
;WHEN ENTERED BY "SET T"
;
; TEST 36
;*****
;TST36: MOV #36, TSTNO
;        MOV #TST37, NEXT
;        JSR PC, SET.UP
;        MOV #BIT14+BIT1, 200SEC
;        SET UP ALL NECESSARY FOR TEST.
;        DECB 200RCSH
;        GET CHAR ADDR 16(8)
;        MOV #10, 200REG
;        GET CHAR REG.
;        MOV #354+400, 200SEC
;        LOAD CHARACTER
;        MOV #14, 200REG
;        SEL THE SEQ REG.
;        MOV #BIT15+BIT7, 200SEC
;        SET SNGL CHAR AND SET DONE CLEAR GO.
;        MOV #6, 200REG
;        SEL TX BA SEC.
;        MOV #TXBUFF+40, 200SEC
;        INCB 200REG
;        SEL TX WC SEC.
;        BIS #BIT13+BIT12, 200ERR
;        SET WRITE ENABLE AND EXIT T
;        MOV #-10, 200SEC
;        SET FOR 10(8) CHARS
;        MOV #350, TXBUFF+3
;        LOAD DATA
;        MOV #354, TXBUFF+41
;        SAME
;        INC 200TCSR
;        SET TX GO.
;        BIT #BIT6, 200TCSR
;        HANG HERE FOR TX SEC. DONE.
;        BEQ 15
;        BR IF NOT DONE
;        MOV #3, 200REG
;        GET TX WC PRI.
;        TST 200SEC
;        IS IT =0
;        BEQ .+4
;        BR IF OK
;        HLT 3
;        TX WC PRI NOT=0
;        MOV #7, 200REG
;        SEL TX WC SEC
;        TST 200SEC
;        IS NOT =0?
;        BNE .+4
;        BR IF OK

```

G04

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 45
 DZDQFD.P11 21-DEC-76 16:34 PROGRAM INITIALIZATION AND START UP.

```

2217 007254 104003          HLT      3          ;TX SEC WC S/B NON-ZERO.
2218 007256 104400          SCOPE                                ;SCOPE THIS TEST.
2219
2220
2221
2222          ;TEST THAT THE RECEIVER STRIP SYNC IS
2223          ;INHIBITED WHEN IN TRANSPARENT MODE.
2224
2225          ; TEST 37
2226          ;*****
2227 007260 012737 000037 001226 1ST37: MOV      #37,TSTNO
2228 007266 012737 007470 001216      MOV      #TST40,NEXT
2229 007274 004737 011024          JSR      PC,SET.UP          ;SET UP ALL NECESSARY FOR TEST.
2230 007300 012777 040002 172064      MOV      #BIT14+BIT1,200SEC
2231
2232 007306 105377 172050          DECB    200RCSH          ;SET DBL CHAR AND SET T
2233 007312 112777 000010 172050      MOV     #10,200REG      ;SEL CHAR ADD 16(B)
2234 007320 013700 001372          MOV     200SEC,RO      ;SEL CHAR REG
2235 007324 012710          MOV     (PC)+(RO)      ;LOAD THE CHAR
2236 007326          101          101          .BYTE   101,101        ;DATA
2237 007330 112777 000014 172032      MOV     #14,200REG      ;SEL THE SEQ REG
2238 007336 012777 101000 172026      MOV     #BIT15+BIT9,200SEC
2239
2240 007344 112737 000350 011753      MOV     #350,TXBUF+1    ;SET SNGL CHAR AND DLE
2241 007352 012700 011754          MOV     #TXBUF+2,RO     ;LOAD BUFFER
2242 007356 112720 000101          IS:    MOV     #101,(RO)+ ;DLE
2243 007362 112720 000026          MOV     #26,(RO)+      ;SYNC
2244 007366 022700 011764          CMP     #TXBUF+12,RO    ;KEEP STUFFING
2245 007372 001371          BNE     IS
2246 007374 012777 000003 171756      MOV     #3,200RCSR      ;SET STRIP SYNC AND GO(RX)
2247 007402 005277 171756          INC     200TCSR          ;SET TX GO
2248 007406 105777 171746          TSTB   200RCSR          ;HANG HERE FOR RX DONE (P)
2249 007412 100375          BPL     -4
2250 007414 012700 012154          MOV     #RXBUF,RO      ;GET RX POINTER
2251 007420 122720 000000          CMPB   #0,(RO)+        ;FIRST CHAR S/B=0
2252 007424 001401          BEQ     +4
2253 007426 104003          HLT     3                ;FIRST DATA CHAR WRONG
2254 007430 122720 000350          CMPB   #350,(RO)+      ;NEXT CHAR S/B=350
2255 007434 001401          BEQ     +4
2256 007436 104003          HLT     3                ;RX BUFFER WRONG
2257 007440 122720 000101          2S:    CMPB   #101,(RO)+   ;DLE PRESENT?
2258 007444 001401          BEQ     +4
2259 007446 104003          HLT     3                ;DLE NOT THERE
2260 007450 122720 000026          CMPB   #26,(RO)+       ;SYNC PRESENT?
2261 007454 001401          BEQ     +4
2262 007456 104003          HLT     3                ;LOOKS LIKE SYNC STRIPPED?
2263 007460 022700 012164          CMP     #RXBUF+10,RO   ;BUFFER DONE?
2264 007464 001365          BNE     2S              ;BR IF NO
2265 007466 104400          SCOPE                                ;SCOPE THIS TEST
2266
2267
2268
2269          ;VERIFY THAT BIT 8 OF THE SEQUENCE
2270          ;REGISTER STRIPS CHARS FROM CORE BUT NOT
2271          ;FROM THE BCC.
2272

```

H04

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 46
 DZDQFD.P11 21-DEC-76 16:34 PROGRAM INITIALIZATION AND START UP.

```

2273 ; TEST 40
2274 ; *****
2275 007470 012737 000040 001226 †ST40: MOV #40,TSTNO ;
2276 007476 012737 010002 001216 MOV #TST41,NEXT ;
2277 007504 004737 011024 JSR PC,SETUP ; SET UP ALL NECESSARY FOR TEST.
2278 007510 012777 100010 171654 MOV #BIT15+BIT3,200SEC ;
2279 ; SET BCC START CLEAR AND SNGL CHAR
2280 007516 105377 171640 DECB 200RCSH ; SEL CHAR ADD 16(B)
2281 007522 112777 000010 171640 MOV #10,200REG ; SEL CHAR REG
2282 007530 012777 112400 171634 MOV #225*400,200SEC ; LOAD CHAR
2283 007536 112777 000014 171624 MOV #14,200REG ; SEL THE SEQ REG
2284 007544 012777 100400 171620 MOV #BIT15+BIT8,200SEC ;
2285 ; SET SNGL CHAR AND CHAR STRIP.
2286 007552 012700 011752 1S: MOV #TXBUFF,RO ; GET TX POINTER
2287 007556 105020 CLR B (RO)+ ; CLEAR IT OUT
2288 007560 022700 012153 CMP #TXBUFF+201,RO ; ALL DONE?
2289 007564 001374 BNE 1S ; BR IF NO
2290 007566 112737 000350 011753 MOV #350,TXPUFF+1 ; LOAD CHAR
2291 007574 112737 000225 011755 MOV #225,TXPUFF+3 ; SAME
2292 007602 112777 000017 171560 MOV #17,200REG ; SEL POLY REG
2293 007610 012777 000200 171554 MOV #200,200SEC ; SET FOR LRC B
2294 007616 012737 000010 012776 MOV #8,COUNT ; SET FOR 8 BIT CHAR
2295 007624 012737 000225 012774 MOV #225,CHAR ; SET EXPECTED BCC CHAR
2296 007632 112777 000003 171530 MOV #3,200REG ; SEL THE TX WC PRI.
2297 007640 012777 177576 171524 MOV #8-202,200SEC ; SET BIG
2298 007646 105077 171516 CLR B 200REG ; SEL REC PRIMARY
2299 007652 012777 012154 171512 MOV #RXBUFF,200SEC ; SET WITH START ADRS
2300 007660 112777 000001 171502 MOV #1,200REG ; SEL REC CHAR CNT
2301 007666 012777 177775 171476 MOV #8-3,C ; SET CHAR COUNT
2302 007674 112777 000004 171466 MOV #4,C ; SEL REC SECONDARY
2303 007702 012777 012174 171462 MOV #RXE F+20,200SEC ; SET WITH SEC ADRS
2304 007710 112777 000005 171452 MOV #5,200REG ; SEL CHAR COUNT
2305 007716 012777 177577 171446 MOV #8-201,200SEC ; SET CHAR COUNT
2306 007724 004737 011632 JSR PC,NEWENA ; SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
2307 007730 013703 012776 MOV COUNT,R3 ; SAVE COUNT
2308 007734 000241 5S: CLC ;
2309 007736 106100 ROL B RO ; SHIFT RX BCC IMAGE
2310 007740 005500 6S: ADC RO ; SAVE CARRY
2311 007742 023700 012774 CMP CHAR,RO ; GOOD BCC?
2312 007746 001403 BEQ 4S ; BR IF YES
2313 007750 005303 DEC R3 ; ALL SHIFTS DONE?
2314 007752 001370 BNE 5S ; BR IF NO
2315 007754 104013 HLT 13 ; RX BCC WRONG!!
2316 007756 012701 012154 4S: MOV #RXBUFF,R1 ; GET RX BUFFER
2317 007762 123721 012774 7S: CMP B CHAR,(R1)+ ; CHAR STRIPPED
2318 007766 001001 BNE .+4 ;
2319 007770 104013 HLT 13 ; 8 BIT CHAR NOT STRIPPED.
2320 007772 022701 012164 8S: CMP #RXBUFF+10,R1 ; ALL DONE?
2321 007776 001371 BNE 7S ; NOT YET
2322 010000 104400 SCOPE
2323
2324
2325
2326 ; TEST OF BCC TEST/ APPEND
2327 ; TEST OF 1 BCC'S TESTED/APPENDED
2328

```

I04

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 47
 DZDQFD.P11 21-DEC-76 16:34 PROGRAM INITIALIZATION AND START UP.

```

2329
2330
2331 010002 012737 000041 001226
2332 010010 012737 010060 001216
2333 010016 012737 100060 011022
2334 010024 012737 000000 012772
2335 010032 012737 000255 012770
2336 010040 004737 011412
2337 010044 017737 171316 012766
2338 010052 100001
2339 010054 104015
2340 010056 104400
2341
2342
2343
2344
2345
2346
2347 010060 012737 000042 001226
2348 010066 012737 010136 001216
2349 010074 012737 100020 011022
2350 010102 012737 000000 012772
2351 010110 012737 112001 012770
2352 010116 004737 011412
2353 010122 017737 171240 012766
2354 010130 100001
2355 010132 104015
2356 010134 104400
2357
2358
2359
2360
2361
2362
2363 010136 012737 000043 001226
2364 010144 012737 010214 001216
2365 010152 012737 100040 011022
2366 010160 012737 000225 012772
2367 010166 012737 112001 012770
2368 010174 004737 011412
2369 010200 017737 171162 012766
2370 010206 100001
2371 010210 104015
2372 010212 104400

; TEST 41
;*****
↑ST41: MOV #41,TSTNO
MOV #↑ST42,NEXT
MOV #BIT15+BITS+BIT4,FUNCT.
MOV #000,XPOLY ;SET EXTENDED POLY.
MOV #255,POLY ;SET 00-15 POLY
JSR PC,BCC.TA ;GOTO SUBROUTINE
MOV #DQERR,ERR ;IS THE AN ERROR CONDITION?
BPL .+4 ;BR IF NO ERRORS
HLT 15 ;THE DQ11 ERROR FLAG IS SET!!
SCOPE ;SCOPE THIS TEST

;TEST OF BCC TEST/ APPEND
;TEST OF 2 BCC'S TESTED/APPENDED

; TEST 42
;*****
↑ST42: MOV #42,TSTNO
MOV #↑ST43,NEXT
MOV #BIT15+BIT4,FUNCT.
MOV #000,XPOLY ;SET EXTENDED POLY.
MOV #112001,POLY ;SET 00-15 POLY
JSR PC,BCC.TA ;GOTO SUBROUTINE
MOV #DQERR,ERR ;IS THE AN ERROR CONDITION?
BPL .+4 ;BR IF NO ERRORS
HLT 15 ;THE DQ11 ERROR FLAG IS SET!!
SCOPE ;SCOPE THIS TEST

;TEST OF BCC TEST/ APPEND
;TEST OF 3 BCC'S TESTED/APPENDED

; TEST 43
;*****
↑ST43: MOV #43,TSTNO
MOV #↑ST44,NEXT
MOV #BIT15+BIT5,FUNCT.
MOV #225,XPOLY ;SET EXTENDED POLY.
MOV #112001,POLY ;SET 00-15 POLY
JSR PC,BCC.TA ;GOTO SUBROUTINE
MOV #DQERR,ERR ;IS THE AN ERROR CONDITION?
BPL .+4 ;BR IF NO ERRORS
HLT 15 ;THE DQ11 ERROR FLAG IS SET!!
SCOPE ;SCOPE THIS TEST

```

2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428

```

: MULTIPLE FUNCTION!!!!
:-----
: FUNCTIONS EXERCISED
: START CHAR (350)
: 15 SNGL CHAR MATCH
: 14 DEL CHAR MATCH
: 13 SNGL CHAR FLAG
: 12 DEL CHAR FLG
: 08 RX STRIP
: 03 BCC START CLEAR
: 01 RX/TX TRANS

: END CHAR (225)
: 15 SNGL CHAR MATCH
: 13 SNGL CHAR FLAG
: 08 RX STRIP
: 07 CLEAR GO/SET DONE
: 05 BCC TEST /APPEND (3 BCC'S)
: 02 CLR RX TRANS

: DLE STRIP/ADD (20)
: 15 SNGL CHAR MATCH
: 09 DLE STRIP/ADD

```

```

: TEST 44
: *****
TST44: MOV #44,TSTNO
MOV #EOP,NEXT
JSR PC,SETUP ;SET UP ALL NECESSARY FOR TEST.
MOV #170412,200SEC ;LOAD THE SEQ FUNCTIONS
DECB 200RCSH ;GET CHAR ADD 16(8)
MOVB #10,200REG ;GET CHAR REG.
MOV #225*400,200SEC ;LOAD CHAR.
MOVB #14,200REG ;SEL SEQ REG
MOV #120644,200SEC ;LOAD SEQ FUNCTIONS
DECB 200RCSH
MOVB #10,200REG ;SEL CHAR DET
MOV #10020,200SEC ;LOAD FUNCTIONS
MOVB #14,200REG ;SEL SEQ REG
MOV #101000,200SEC ;LOAD DLE (20)
MOVB #6,200REG ;SEL TX SEC ADRS
MOV #170412,200SEC ;SET SEC ADRS
MOVB #1,200REG ;SEL REC PRI SHAR CNT
MOV #-200,200SEC ;SET CHAR CNT
MOVB #67,200REG ;SEL TXSEC CHAR CNT
MOV #-2,200SEC ;SET CHAR CNT
MOVB #17,200REG ;SEL POLY REG
MOV #172516,200SEC ;SET #WILD* POLYNOMIAL
MOVB #12,200REG ;SEL THE MISC REG
BIS #816,200SEC ;SET THE "EXT POLY" BIT
MOVB #17,200REG ;WRITE POLY 16-23

```

```

010214 012737 000044 001226
010222 012737 013006 001216
010230 004737 011024
010234 012777 170412 171130
010242 105377 171114
010246 112777 000010 171114
010254 012777 112400 171110
010262 112777 000014 171100
010270 012777 120644 171074
010276 105377 171060
010302 112777 000010 171060
010310 012777 010020 171054
010316 112777 000014 171044
010324 012777 101000 171040
010332 112777 000006 171030
010340 012777 012022 171024
010346 112777 000001 171014
010354 012777 177600 171010
010362 112777 000067 171000
010370 012777 177776 170774
010376 112777 000017 170764
010404 012777 172516 170760
010412 112777 000012 170750
010420 052777 000100 170744
010426 112777 000017 170734

```

K04

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 49
 DZDQFD.P11 21-DEC-76 16:34 PROGRAM INITIALIZATION AND START UP.

2429	010434	012777	000275	170730	MOV	#275, 200SEC	; GET POINTER
2430	010442	012700	011752		MOV	#RXBUFF, R0	; LOAD THE BUFFER WITH DATA
2431	010446	012720			MOV	(PC)+, (R0)+	; DATA
2432	010450	000	350		.BYTE	000, 350	; LOAD THE BUFFER WITH DATA
2433	010452	012720			MOV	(PC)+, (R0)+	; DATA
2434	010454	311	224		.BYTE	311, 224	; LOAD THE BUFFER WITH DATA
2435	010456	012720			MOV	(PC)+, (R0)+	; DATA
2436	010460	107	201		.BYTE	107, 201	; LOAD THE BUFFER WITH DATA
2437	010462	012720			MOV	(PC)+, (R0)+	; DATA
2438	010464	371	251		.BYTE	371, 251	; LOAD RO
2439	010466	012700	012022		MOV	#TXBUFF+50, R0	; LOAD BUFFER WITH DATA
2440	010472	012720			MOV	(PC)+, (R0)+	; DATA
2441	010474	225	377		.BYTE	225, 377	; SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
2442	010476	004737	011556		JSR	PC, ENABLE	; DID RX CHAR FLAG SET?
2443	010502	005777	170652		TST	200RCSR	; BR IF YES
2444	010506	100401			BMI	.+4	; CHAR DET FLAG NOT SET.
2445	010510	104014			HLT	14	; ANY ERRORS?
2446	010512	017737	170650	012766	MOV	200ERR, ERR	; BR IF NO ERRORS
2447	010520	100001			BPL	.+4	; DQ11 ERROR FLAG SET!
2448	010522	104017			HLT	17	; DONE(P)=1; P/S=0; GO=0?
2449	010524	122777	000200	170626	CMPB	#200, 200RCSR	; YES
2450	010532	001401			BEQ	.+4	; RX CSR WRONG DATA
2451	010534	104014			HLT	14	; DONE(P)=1; P/S=0; GO=0?
2452	010536	122777	000300	170620	CMPB	#300, 200TCSR	; YES
2453	010544	001401			BEQ	.+4	; TX CSR WRONG DATA
2454	010546	104014			HLT	14	; GET RX WC PRI.
2455	010550	112777	000001	170612	MOVB	#1, 200REG	; S/B NOT=0
2456	010556	005777	170610		TST	200SEC	; BR IF OK
2457	010562	001001			BNE	.+4	; RX WC PRI S/B NON-ZERO!!
2458	010564	104014			HLT	14	; SEL THE TX WC PRI.
2459	010566	112777	000003	170574	MOVB	#3, 200REG	; S/B =0
2460	010574	005777	170572		TST	200SEC	; TX WC PRI S/B NON-ZERO
2461	010600	001401			BEQ	.+4	; GET RX BUFFER POINTER
2462	010602	104014			HLT	14	; SET CHAR COUNT
2463	010604	012700	012154		MOV	#RXBUFF, R0	; ARE THE TWO CHARS STRIPPED?
2464	010610	012702	000010		MOV	#10, R2	; CHAR "350" NOT STRIPPED FROM CORE
2465	010614	122710	000350		CMPB	#350, (R0)	; CHAR "225" NOT STRIPPED
2466	010620	001001			BNE	.+4	; IS DLE STRIPPED?
2467	010622	104014			HLT	14	; DLE STUCK -- REFER TO M7817 ECO
2468	010624	122720	000225		CMPB	#225, (R0)+	; CHECK EB4 ONE-SHOT
2469	010630	001001			BNE	.+4	; ALL DONE?
2470	010632	104014			HLT	14	; NO
2471	010634	122720	000020		CMPB	#20, (R0)+	; SCOPE THE TEST
2472	010640	001001			BNE	.+4	
2473	010642	104021			HLT	21	
2474							
2475	010644	005302			DEC	R2	
2476	010646	001362			BNE	15	
2477	010650	104400			SCOPE		

2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533

```

010652
010652 105077 170512
010656 012705 000020
010662 152777 000020 170500
010670 142777 000140 170472
010676 005077 170470
010702 105277 170462
010706 005305
010710 001364
010712 105077 170452
010716 105077 170440
010722 012705 000020
010726 112777 000010 170434
010734 005077 170432
010740 112777 000014 170422
010746 005077 170420
010752 105277 170404
010756 005305
010760 001362
010762 105077 170374
010766
010766 112777 000012 170374
010774 012777 000040 170370
011002 000240
011004 112777 000012 170356
011012 012777 000040 170352
011020 000002
011022 000000
011024
011024 104413
011026 012702 000010
011032 012700 011752
011036 105020
011040 005302
011042 001375
011044 105077 170320
011050 012777 012154 170314
011056 105277 170306
011062 012777 177770 170302
011070 105277 170274
011074 012777 011750 170270
011102 105277 170262
011106 012777 177766 170256
011114 112777 000011 170246
011122 013777 011746 170242
011130 105277 170234
011134 012777 004000 170230
011142 032737 040000 001510
011150 001003
011152 052777 000010 170212
011160 112777 000017 170174
011166 112777 000010 170174
011174 012777 164000 170170
    
```

```

.MEMCLR:
CLR R2, @DQREG
MOV #16, R5
BISB #BIT4, @DQREG
BICB #140, @DQREG
CLR @DQSEC
INCB @DQREG
DEC R5
BNE 15
CLR @DQREG
CLR @DQRC SH
MOV #16, R5
MOVB #10, @DQREG
CLR @DQSEC
MOVB #14, @DQREG
CLR @DQSEC
INCB @DQRC SH
DEC R5
BNE 25
CLR @DQRC SH

.MSTCLR:
MOVB #MISC, @DQREG
MOV #BITS, @DQSEC
NOP
MOVB #MISC, @DQREG
MOV #BITS, @DQSEC
RTI

FUNCT.: 0
SET.UP:
MEMCLR
MOV #10, R2
MOV #TXBUFF, R0
CLR (R0)+
DEC R2
BNE .-4
CLR @DQREG
MOV #RXBUFF, @DQSEC
INCB @DQREG
MOV #-10, @DQSEC
INCB @DQREG
MOV #SYNC, @DQSEC
INCB @DQREG
MOV #-12, @DQSEC
MOVB #11, @DQREG
MOV #SYNC, @DQSEC
INCB @DQREG
MOV #4000, @DQSEC
BIT #JUMBIT, DQSTAT
BNE .+10
BIS #BIT3, @DQSEC
MOVB #17, @DQRC SH
MOVB #10, @DQREG
MOV #350*400, @DQSEC
    
```

```

; CLEAR ALL THE DQ11 REGISTERS
; PREPARE TO CLEAR THE TX BUFFER
; GET THE BUFFERS ADDRESS
; START CLEARING
; ALL CLEAR?
; BR IF NOT DONE.
; SELECT THE RX BA PRI.
; LOAD RX BA PRI.
; GET RX WC PRI.
; SET FOR 8. CHARS.
; GET THE TX BA PRI.
; SET ADDRESS
; SEL THE TX WC PRI.
; SET FOR 8. CHARS AND 2 SYNC'S
; SEL THE SYNC REGISTER
; LOAD WITH SYNC
; SELECT THE MISC REGISTER
; SET FOR EIGHT BITS.
; IF TEST JUMPER AT END OF CABLE;
; RUN DATA THROUGH IT.
; NO TEST JUMPER; SET TEST LOOP
; GET LAST CHAR DET ADDRESS.
; SEL CHAR DET REGISTER
; LOAD THE CHARACTER
    
```

M04

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 51
 DZDQFD.P11 21-DEC-76 16:34 PROGRAM INITIALIZATION AND START UP.

2534	011202	112777	000014	170160	MOV	#14, 200REG	; GET THE SEQ REGISTER
2535	011210	000207			RTS	PC	; LEAVE ROUTINE POINTING TO SEQ REGISTER.
2536	011212				EXT.UP:		
2537	011212	104413			MEMCLR		; CLEAR ALL THE DQ11
2538	011214	012702	000020		MOV	#20, R2	; PREPARE TO CLEAR THE TX BUFFER
2539	011220	012700	012362		MOV	#TXBUF, R0	; GET THE TX BUFFER ADDRESS.
2540	011224	005020			CLR	(R0)+	; START CLEARING
2541	011226	005302			DEC	R2	; DONE?
2542	011230	001375			BNE	.-4	; BR IF NO
2543	011232	012737	013026	012356	MOV	#13026, XSYNC	; LOAD SYNC
2544	011240	012737	013026	012360	MOV	#13026, XSYNC2	; DITTO
2545	011246	105077	170116		CLRB	200REG	; SEL THE RX BA PRI.
2546	011252	012777	012154	170112	MOV	#RXBUF, 200SEC	; LOAD THE ADDRESS
2547	011260	105277	170104		INCB	200REG	; SEL THE RX WC PRI.
2548	011264	012777	177770	170100	MOV	#-10, 200SEC	; SET FOR TEN CHARS
2549	011272	105277	170072		INCB	200REG	; SEL THE TX BA PRI.
2550	011276	012777	012356	170066	MOV	#XSYNC, 200SEC	; LOAD THE ADDRESS.
2551	011304	105277	170060		INCB	200REG	; SEL THE TX WC PRI.
2552	011310	012777	177766	170054	MOV	#-12, 200SEC	; SET FOR TWO SYNC AND 8. CHARS
2553	011316	112777	000011	170044	MOV	#11, 200REG	; SEL THE SYNC REGISTER
2554	011324	013777	011746	170040	MOV	.SYNC, 200SEC	; LOAD SYNC
2555	011332	105277	170032		INCB	200REG	; SEL THE MISC REGISTER.
2556	011336	005077	170030		CLR	200SEC	; SEL 16 BITS PER CHAR.
2557	011342	032737	040000	001510	BIT	#JUMBIT, DQSTAT	; IF TEST JUMPER INSTALLED;
2558	011350	001003			BNE	.+10	; RUN DATA THROUGH CABLE.
2559	011352	052777	000010	170012	BIS	#BIT3, 200SEC	; NO JUMPER; SET TEST LOOP!
2560	011360	112777	000017	167774	MOV	#17, 200REG	; GET LAST CHAR DET ADDRESS
2561	011366	112777	000010	167774	MOV	#10, 200REG	; GET CHAR DET REGISTER
2562	011374	012777	003721	167770	MOV	#3721, 200SEC	; LOAD CHARACTER
2563	011402	112777	000014	167760	MOV	#14, 200REG	; SEL THE SEQ REGISTER
2564	011410	000207			RTS	PC	; LEAVE THE ROUTINE.
2565					BCC.TA:		
2566	011412				JSR	PC, SET.UP	; SET UP ALL NECESSARY FOR TEST.
2567	011412	004737	011024		MOV	#BIT15+BIT3, 200SEC	
2568	011416	012777	100010	167746			; SET SNGL CHAR AND BCC START CLEAR
2569					DECB	200RCSH	; GET NEXT ADDR
2570	011424	105377	167732		MOV	#10, 200REG	; SEL CHAR DET ADDR
2571	011430	112777	000010	167732	MOV	#375+400, 200SEC	; LOAD CHAR
2572	011436	012777	176400	167726	MOV	#14, 200REG	; SEL THE SEQ REG.
2573	011444	112777	000014	167716	MOV	FUNC, 200SEC	; SET THE TEST APPEND FUNCTIONS
2574	011452	013777	011022	167712	MOV	#17, 200REG	; SEL THE POLY REG.
2575	011460	112777	000017	167702	MOV	POLY, 200SEC	; LOAD THE POLYNOMIAL.
2576	011466	013777	012770	167676	MOV	#12, 200REG	; SEL THE MISC REG
2577	011474	112777	000012	167666	BIS	#BIT6, 200SEC	; SEL EXT POLY REG
2578	011502	052777	000100	167662	MOV	#17, 200REG	; RESEL. THE POLY REG
2579	011510	112777	000017	167652	MOV	XPOLY, 200SEC	; SET 16-23 POLY
2580	011516	013777	012772	167646	MOV	#TXBUF, R0	; SET TX POINTER
2581	011524	012700	011752		MOV	(PC)+, (R0)+	; LOAD THE BUFFER WITH DATA
2582	011530	012720			.BYTE	350, 355	; DATA
2583	011532	350	355		MOV	(PC)+, (R0)+	; LOAD THE BUFFER WITH DATA
2584	011534	012720			.BYTE	360, 365	; DATA
2585	011536	360	365		MOV	(PC)+, (R0)+	; LOAD THE BUFFER WITH DATA
2586	011540	012720			.BYTE	370, 375	; DATA
2587	011542	370	375		MOV	(PC)+, (R0)+	; LOAD THE BUFFER WITH DATA
2588	011544	012720			.BYTE	377, 377	; DATA
2589	011546	377	377				

NO4

DZDGF MACY11 27(1006) 22-DEC-76 11:36 PAGE 52
 DZDGF.D.P11 21-DEC-76 16:34 PROGRAM INITIALIZATION AND START UP.

```

2590 011550 004737 011556      ISR      PC,ENABLE      ;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
2591 011554 000207                RTS      PC          ;LEAVE
2592
2593 011556 005037 011626      ENABLE: CLR      3$          ;
2594 011562 012737 000005 011630  MOV      8$ 4$          ;SET DELAY
2595 011570 005277 167564      INC      200RCSR        ;SET RX GO.
2596 011574 005277 167564      INC      200TCSR        ;SET TX GO.
2597 011600 105777 167554      1$:     TSTB      200RCSR  ;RX PRI. DONE?
2598 011604 100407                BMI      2$          ;BR IF YES
2599 011606 005237 011626      INC      3$          ;DELAY.....
2600 011612 001372                BNE      1$
2601 011614 005337 011630      DEC      4$
2602 011620 001367                BNE      1$
2603 011622 104001                HLT
2604 011624 000207      2$:     RTS      PC          ;RX PRI DONE NOT SET.
2605 011626 000100      3$:     000
2606 011630 000000      4$:     000
2607
2608 011632 005037 011742      NEWENA: CLR      3$          ;
2609 011636 012737 000005 011744  MOV      8$ 4$          ;SET DELAY
2610 011644 005277 167510      INC      200RCSR        ;SET RX GO
2611 011650 005277 167510      INC      200TCSR        ;SET TX GO
2612 011654 105777 167500      1$:     TSTB      200RCSR  ;RX PRIMARY DONE?
2613 011660 100410                BMI      2$          ;BR IF YES
2614 011662 005237 011742      INC      3$          ;DELAY
2615 011666 001372                BNE      1$
2616 011670 005337 011744      DEC      4$
2617 011674 001367                BNE      1$
2618 011676 104001                HLT
2619 011700 000417                BR
2620 011702 112777 000012 167460  2$:     MOVB      #12,200REG  ;RX PRI DONE FAILED TO SET
2621 011710 042777 000010 167454  BIC      #813,200SEC    ;LEAVE
2622 011716 112777 000015 167444  MOVB      #15,200REG    ;SEL THE MISC REG
2623 011724 017700 167442      MOV      200SEC,R0     ;STOP DATA
2624 011730 105277 167434      INCB      200REG,SEL TX ;SEL RX BCC
2625 011734 017701 167432      MOV      200SEC,R1    ;READ INTO R0
2626 011740 000207      5$:     RTS      PC          ;READ INTO R1
2627 011742 000000      3$:     .WORD 0
2628 011744 000000      4$:     .WORD 0
2629
2630
2631 011746      026      026      .SYNC: .BYTE 26,26
2632 011750      026      026      SYNC:  .BYTE 26,26
2633 011752 000000      TXBUFF: 0
2634      012154      RXBUFF: 0
2635 012154 000000      .+.200
2636      012356      .+.200
2637 012356      026      026      XSYNC: .BYTE 26,26
2638 012360      026      026      XSYNC2: .BYTE 26,26
2639 012362 000000      XTXBUF: 000000
2640      012564      .+.200
2641 012564 000000      XRXBUF: 000000
2642      012766      .+.200
2643 012766 000000      ERR: 0
2644 012770 000000      POLY: 0
2645 012772 000000      XPOLY: 0

```

2646	012774	000000			CHAR:	0		
2647	012776	000000			COUNT:	0		
2648	013770	000000			ADDR:	0		
2649	013772	000000			GOCHAR:	0		
2650	013004	000000			DETCAR:	0		
2651								
2652								
2653								
2654								
2655								
2656								
2657								
2658	013006	005037	001234		.EOP:	CLR	LSTERR	;CLEAR LAC ERROR PC
2659	013012	005037	001312			CLR	ERRFLG	;CLEAR ERROR FLAG
2660	013016	005237	001230			INC	PASCNT	;UPDATE PASS COUNT
2661	013022	104402				TYPE		
2662	013024	015236				MEPASS		
2663	013026	104402				TYPE		
2664	013030	015417				MCSRX		
2665	013032	104411				CNVRT		
2666	013034	013144				XCSR		
2667	013036	104402				TYPE		
2668	013040	015425				MVECX		
2669	013042	104411				CNVRT		
2670	013044	013152				XVEC		
2671	013046	104402				TYPE		
2672	013050	015433				MPASSX		
2673	013052	104411				CNVRT		
2674	013054	013160				XPASS		
2675	013056	104402				TYPE		
2676	013060	015444				MERRX		
2677	013062	104411				CNVRT		
2678	013064	013166				XERR		
2679	013066	013777	001230	166106		MOV	PASCNT, @LIGHTS	;DISPLAY PASS COUNT
2680	013074	005337	001276			DEC	SAVNUM	
2681	013100	001013				BNE	RESTRT	
2682	013102	013737	001504	001276		MOV	DQNUM, SAVNUM	
2683	013110	013701	000042			MOV	@#42, R1	
2684	013114	001405				BEQ	RESTRT	;CHECK FOR ACT-11 OR DDP
2685	013116	000005				RESET		;IF NOT, CONTINUE TESTING
2686	013120				LOGICAL:			
2687	013120	004711				JSR	PC, (R1)	
2688	013122	000240				NOP		
2689	013124	000240				NOP		
2690	013126	000240				NOP		
2691	013130	104414				NOP		
2692	013132	012737	002254	001214	RESTRT:	CKSWR		
2693	013140	000137	002254			MOV	@TST1, RETURN	
2694	013144	000001				JMP	TST1	
2695	013146	006	002		XCSR:	1		
2696	013150	001360				.BYTE	6,2	
2697	013152	000001				DORCSR		
2698	013154	003	002		XVEC:	1		
2699	013156	001350				.BYTE	3,2	
2700	013160	000001			XPASS:	1		
2701	013162	006	002			.BYTE	6,2	


```

2758 013422 112577 165564          MOVB   (R5)+,@TPDBR
2759 013426 001345          BNE    1$
2760 013430 012605          3$:   MOV   (SP)+,R5
2761 013432 000002          RTI
2762
2763          ;ASCII STRING INPUT ROUTINE
2764
2765 013434 010346          .INSTR: MOV   R3,-(SP)
2766 013436 010446          MOV   R4,-(SP)
2767 013440 017637 000004 013456  MOV   #4(SP),MSG
2768 013446 062766 000002 000004  ADD   #2,4(SP)
2769 013454 104402          .INST1: TYPE
2770 013456 000000          .MSG:   0
2771 013460 012704 015610          MOV   #INBUF,R4
2772 013464 012703 000007          MOV   #7,R3
2773 013470 105777 165510          1$:   TSTB  @TKCSR
2774 013474 100375          BPL   1$
2775 013476 117714 165504          MOVB  @TKDBR,(R4)
2776 013502 142714 000200          BICB  #200,(R4)
2777 013506 121427 000025          CMPB  (R4),#25          ;IS IT (<G)
2778 013512 001003          BNE   200$
2779 013514 104402 015176          TYPE,MCRLF
2780 013 20 000755          BR    .INST1
2781 013 22 122427 000015          200$: CMPB  (R4)+,#15
2782 013 36 001423          BEQ   INSTR2
2783 013 30 117777 165452 165454  MOVB  @TKDBR,@TPCDBR
2784 013536 105777 165446          2$:   TSTB  @TPCSR
2785 013542 100375          BPL   2$
2786 013544 005303          DEC   R3
2787 013546 001350          BNE   1$
2788 013550 000402          BR    .INSTG
2789 013 52 010346          .INSTE: MOV   R3,-(SP)
2790 013554 010446          MOV   R4,-(SP)
2791 013556 104402          .INSTG: TYPE
2792 013560 015172          MCM
2793 013562 005737 015016          TST   @#RDSW
2794 013566 001402          BEQ   400$
2795 013570 104402 015176          TYPE,MCRLF
2796 013574 000727          400$: BR    .INST1
2797 013576 012604          INSTR2: MOV  (SP)+,R4
2798 013600 012603          MOV  (SP)+,R3
2799 013602 000002          RTI
2800
2801          ;CONVERT ASCII STRING TO OCTAL
2802
2803 013604 010546          .PARAM: MOV   R5,-(SP)
2804 013606 010446          MOV   R4,-(SP)
2805 013610 016605 000004          MOV   4(SP),R5
2806 013614 012537 014010          MOV   (R5)+,LOLIM
2807 013620 012537 014012          MOV   (R5)+,HILIM
2808 013624 012537 014014          MOV   (R5)+,DEVADR
2809 013630 112537 014016          MOVB  (R5)+,LOBITS
2810 013634 112537 014017          MOVB  (R5)+,ADRCNT
2811 013640 010566 000004          MOV   R5,4(SP)
2812 013644 005005          PARAM1: CLR  R5
2813 013646 012704 015610          MOV   #INBUF,R4

```

E05

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 56
 DZDQFD.P11 21-DEC-76 16:34 GENERAL UTILITIES (TYPE OUT, ERROR, SCOPE, ETC.)

2814	013652	122714	000015		CMPB	#15, (R4)	
2815	013656	001420			BEQ	PARERR	
2816	013660	121427	000060	15:	CMPB	(R4) #60	
2817	013664	002415			BLT	PARERR	
2818	013666	121427	000067		CMPB	(R4) #67	
2819	013672	003012			BGT	PARERR	
2820	013674	142714	000060		BICB	#60, (R4)	
2821	013700	152405			BISB	(R4)+, R5	
2822	013702	122714	000015		CMPB	#15, (R4)	
2823	013706	001414			BEQ	LIMITS	
2824	013710	006305			ASL	R5	
2825	013712	006305			ASL	R5	
2826	013714	006305			ASL	R5	
2827	013716	000760			BR	15	
2828	013720	122714	000015	PARERR:	CMPB	#15, (R4)	; IS FIRST CHARACTER A <CR>
2829	013724	001003			BNE	1205	
2830	013726	005737	015016		TST	#RDSW	; IS CKSW ROUTINE BEING USED
2831	013732	001023			BNE	PARTI	
2832	013734	104404		1205:	INSTR		
2833	013736	000742			BR	PARAM1	
2834							
2835							; TEST TO SEE IF NUMBER IS WITHIN LIMITS
2836							
2837	013740	020537	014012	LIMITS:	CMP	R5, HILIM	
2838	013744	101365			BHI	PARERR	
2839	013746	020537	014010		CMP	R5, LOLIM	
2840	013752	103762			BLO	PARERR	
2841	013754	133705	014016		BITB	LOBITS, R5	
2842	013760	001357			BNE	PARERR	
2843							
2844							; STORE NUMBER AT SPECIFIED ADDRESS
2845							
2846	013762	013704	014014		MOV	DEVADR, R4	
2847	013766	010524		15:	MOV	R5, (R4)+	
2848	013770	062705	000002		ADD	#2, R5	
2849	013774	105337	014017		DECB	ADRCNT	
2850	014000	001372			BNE	15	
2851	014002	012604		PARTI:	MOV	(SP)+, R4	
2852	014004	012605			MOV	(SP)+, R5	
2853	014006	000002			RTI		
2854	014010	000000		LOLIM:	0		
2855	014012	000000		HILIM:	0		
2856	014014	000000		DEVADR:	0		
2857	014016	000000		LOBITS:	0		
2858		014017		ADRCNT=	LOBITS+1		
2859							
2860							; SAVE PC OF TEST THAT FAILED AND R0-R5
2861							
2862	014020	016637	000004 001274	.SAV05:	MOV	4(SP), SAVPC	
2863							
2864							; SAVE R0-R5
2865							
2866	014026	010537	001270	SV05:	MOV	R5, SAVR5	
2867	014032	010437	001266		MOV	R4, SAVR4	
2868	014036	010337	001264		MOV	R3, SAVR3	
2869	014042	010237	001262		MOV	R2, SAVR2	

F05

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 57
 DZDQFD.P11 21-DEC-76 16:34 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

2870	014046	010137	001260		MOV	R1,SAVR1
2871	014052	010037	001256		MOV	RO,SAVRO
2872	014056	000002			RTI	
2873						
2874					;RESTORE RO-R5	
2875						
2876	014060	013700	001256	.RESOS:	MOV	SAVRO,RO
2877	014064	013701	001260		MOV	SAVR1,R1
2878	014070	013702	001262		MOV	SAVR2,R2
2879	014074	013703	001264		MOV	SAVR3,R3
2880	014100	013704	001266		MOV	SAVR4,R4
2881	014104	013705	001270		MOV	SAVR5,R5
2882	014110	000002			RTI	
2883						
2884					;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER	
2885						
2886	014112	104402		.CONVR:	TYPE	
2887	014114	015176			MCRLF	
2888	014116	010046		.CNVRT:	MOV	RO,-(SP)
2889	014120	010146			MOV	R1,-(SP)
2890	014122	010346			MOV	R3,-(SP)
2891	014124	010446			MOV	R4,-(SP)
2892	014126	010546			MOV	R5,-(SP)
2893	014130	017601	000012		MOV	#12(SP),R1
2894	014134	013737	015652		MOV	TEMP,TEMP3
2895	014142	062766	000002	001250	ADD	#2,12(SP)
2896	014150	012137	014332	000012	MOV	(R1)+,WRCNT
2897	014154	112137	014334		15:	MOVSB (R1)+,CHRCNT
2898	014160	112137	014335			MOVSB (R1)+,SPACNT
2899	014164	013137	014336			MOV #2(R1)+,BINWRD
2900	014170	013704	014336	25:	MOV	BINWRD,R4
2901	014174	113705	014334		MOVSB	CHRCNT,R5
2902	014200	012700	015652		MOV	#TEMP,R0
2903	014204	010403		35:	MOV	R4,R3
2904	014206	042703	177770		BIC	#177770,R3
2905	014212	062703	000060		ADD	#060,R3
2906	014216	110320			MOVSB	R3,(R0)+
2907	014220	000241			CLC	
2908	014222	006004			ROR	R4
2909	014224	000241			CLC	
2910	014226	006004			ROR	R4
2911	014230	000241			CLC	
2912	014232	006004			ROR	R4
2913	014234	005305			DEC	R5
2914	014236	001362			BNE	35
2915	014240	012703	015714		MOV	#MDATA,R3
2916	014244	114023		45:	MOVSB	-(R0),(R3)+
2917	014246	105337	014334		DECB	CHRCNT
2918	014252	001374			BNE	45
2919	014254	105737	014335		TSTB	SPACNT
2920	014260	001405			BEQ	65
2921	014262	112723	000040	55:	MOVSB	#040,(R3)+
2922	014266	105337	014335		DECB	SPACNT
2923	014272	001373			BNE	55
2924	014274	105013		65:	CLRB	(R3)
2925	014276	104402			TYPE	

G05

DZDGF MACY11 27(1006) 22-DEC-76 11:36 PAGE 58
 DZDGF.D.P11 21-DEC-76 16:34 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

2926	014300	015714				MDATA	
2927	014302	005337	014332			DEC	WRDCNT
2928	014306	001322				BNE	IS
2929	014310	013737	001250	015652		MOV	TEMP3,TEMP
2930	014316	012605				MOV	(SP)+,R5
2931	014320	012604				MOV	(SP)+,R4
2932	014322	012603				MOV	(SP)+,R3
2933	014324	012601				MOV	(SP)+,R1
2934	014326	012600				MOV	(SP)+,R0
2935	014330	000002				RTI	
2936	014332	000000				WRDCNT: 0	
2937	014334	000000				CHRCNT: 0	
2938		014335				SPACNT=CHRCNT+1	
2939	014336	000000				BINWRD: 0	
2940						; TRAP DISPATCH SERVICE	
2941						; ARGUMENT OF TRAP IS EXTRACTED	
2942						; AND USED AS OFFSET TO OBTAIN POINTER	
2943						; TO SELECTED SUBROUTINE	
2944							
2945	014340	011646				.TRPSR: MOV	(SP) -(SP) ; GET PC OF RETURN
2946	014342	162716	000002			SUB	#2,(SP) ; =PC OF TRAP
2947	014346	017616	000000			MOV	2(SP),(SP) ; GET TRP
2948	014352	006316			TRPOK: ASL	(SP)	; MULTIPLY TRAP ARG BY 2
2949	014354	042716	177001			BIC	#177001,(SP) ; CLEAR UNWANTED BITS
2950	014360	062716	001314			ADD	#.TRPTAB,(SP) ; POINTER TO SUBROUTINE ADDRESS
2951	014364	017616	000000			MOV	2(SP),(SP) ; SUBROUTINE ADDRESS
2952	014370	000136				JMP	2(SP)+ ; GO TO SUBROUTINE
2953							
2954						; ERROR HANDLER	
2955							
2956	014372	104414				.HLT: CKSWR	
2957	014374	032777	010000	164576		BIT	#SW12,2SWR
2958	014402	001406				BEQ	XBX
2959	014404	105777	164600			TSTB	2TPCSR
2960	014410	100003				BPL	XBX
2961	014412	112777	000207	164572		MOVB	#207,2TPDBR
2962	014420	032777	020000	164552	XBX:	BIT	#SW13,2SWR
2963	014426	001074				BNE	HALTS
2964	014430	021637	001234			CMP	(SP),LSTERR
2965	014434	001404				BEQ	IS
2966	014436	011637	001234			MOV	(SP),LSTERR
2967	014442	105037	001312			CLRB	ERRFLG
2968	014446	104406			IS:	SAVOS	
2969	014450	011605				MOV	(SP),R5
2970	014452	162705	000002			SUB	#2,R5
2971	014456	011504				MOV	(R5),R4
2972	014460	006304				ASL	R4
2973	014462	061504				ADD	(R5),R4
2974	014464	006304				ASL	R4
2975	014466	042704	177001			BIC	#177001,R4
2976	014472	062704	015756			ADD	#.ERRTAB,R4
2977	014476	012437	014570			MOV	(R4)+,ERRMSG
2978	014502	012437	014602			MOV	(R4)+,DATAHD
2979	014506	011437	014614			MOV	(R4),DATAHP
2980	014512	105737	001312			TSTB	ERRFLG
2981	014516	001403				BEQ	TYPMSG

H05

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 59
 DZDQFD.P11 21-DEC-76 16:34 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

2982	014520	005737	014614			TST	DATABP	
2983	014524	001027				BNE	TYPDAT	
2984	014526	104402			TYPMSG:	TYPE		
2985	014530	015455				MTSTN		
2986	014532	104411				CMVRT		
2987	014534	014714				XTSTN		
2988	014536	104402				TYPE		
2989	014540	015543				MERRPC		
2990	014542	104411				CMVRT		
2991	014544	014706				ERTAB0		
2992	014546	104402				TYPE		
2993	014550	015176				MCRLF		
2994	014552	112737	177777	001312		MOVB	#-1,ERRFLG	
2995	014560	005737	014570			TST	ERRMSG	
2996	014564	001402				BEQ	WRKO.FM	
2997	014566	104402				TYPE		
2998	014570	000000			ERRMSG:	0		
2999	014572				WRKO.FM:			
3000	014572	005737	014602			TST	DATAHD	
3001	014576	001402				BEQ	TYPDAT	
3002	014600	104402				TYPE		
3003	014602	000000			DATAHD:	0		
3004	014604	005737	014614		TYPDAT:	TST	DATABP	
3005	014610	001402				BEQ	RESREG	
3006	014612	104410				CONVRT		
3007	014614	000000			DATABP:	0		
3008	014616	104407			RESREG:	RESOS		
3009	014620	005777	164354		HALTS:	TST	QSWR	
3010	014624	100005				BPL	EXITER	
3011	014626	010046				PUSHRO		
3012	014630	016600	000002			MOV	2(SP),RO	
3013	014634	000000				HALT		
3014	014636	012600				POPPO		
3015	014640	104414			EXITER:	CKSWR		
3016	014642	005237	001232			INC	ERRCNT	
3017	014646	032777	000400	164324		BIT	#SW08,QSWR	
3018	014654	001007				BNE	1\$	
3019	014656	032777	002000	164314		BIT	#SW10,QSWR	
3020	014664	001407				BEQ	2\$	
3021	014666	013737	001216	001214		MOV	NEXT,RETURN	
3022	014674	012706	001200		1\$:	MOV	#STACK,SP	
3023	014700	000177	164310			JMP	QRETURN	
3024	014704	000002			2\$:	RTI		
3025	014706	000001			ERTAB0:	1		
3026	014710	006	002			.BYTE	6,2	
3027	014712	001274				SAVPC		
3028	014714	000001			XTSTN:	1		
3029	014716	003	002			.BYTE	3,2	
3030	014720	001226				TSTNO		
3031						;ENTER HERE ON POWER FAILURE		
3032								
3033								
3034	014722				.PFAIL:			
3035	014722	012737	014734	000024		MOV	#RESTART,24	
3036	014730	000000				HALT		;SET UP FOR POWER UP TRAP
3037	014732	000777				BR	.	;HALT ON POWER DOWN NORMAL

```

3038
3039
3040 ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
3041 014734 RESTAR:
3042 014734 012737 014722 000024 MOV #PFAIL, 24 ;SET UP FOR POWER FAILURE
3043 014742 012706 001200 MOV #STACK, SP
3044 014746 005037 015652 CLR TEMP
3045 014752 005237 015652 INC TEMP
3046 014756 001375 BNE .-4
3047 014760 104402 TYPE
3048 014762 015200 MPFAIL
3049 014764 104411 CNVRT
3050 014766 015010 PFTAB
3051 014770 005037 001312 CLR ERRFLG
3052 014774 005037 001234 CLR LSTERR
3053 015000 104412 MSTCLR
3054 015002 104413 MEMCLR
3055 015004 000177 164204 JMP @RETURN
3056 015010 000001 PFTAB: 1
3057 015012 003 002 .BYTE 3,2
3058 015014 001226 TSTNO
3059
3060
3061 ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ↑G TO ALLOW CHANGING
3062 ;OF LOC.176.
3063 ;LOCATIONS USED:
3064 015016 000000 RDSW: .WORD 0
3065
3066
3067 015020 005737 000042 .CKSWR: TST 2042
3068 015024 001042 BNE OUT
3069 015026 022737 000176 001200 CMP #SWREG, SWR ;SOFTWARE SWITCH REGISTER PRESENT
3070 015034 001036 BNE OUT ;NO, GET OUT
3071 015036 105777 164142 TSTB @TKCSR ;YES, WAIT FOR
3072 015042 100033 BPL OUT ;READY, GET CHARACTER
3073 015044 017737 164136 013456 MOV @TKDBR, .MSG ;AND STRIP OFF
3074 015052 042737 177600 013456 BIC #177600, .MSG ;THE GARBAGE
3075 015050 122737 000007 013456 CMPB #7, .MSG ;IS IT A ↑G?
3076 015056 001021 BNE OUT
3077 015070 104402 015146 TYPE, SCNTG
3078 015074 005137 015016 .CNTLU: COM @RDSW
3079 015100 104402 015152 TYPE, SMSWR
3080 015104 104411 015140 CNVRT, SWREGC
3081 015110 104403 015161 INSTR, SMNEW
3082 015114 104405 PARAM
3083 015116 000000 0
3084 015120 177777 177777
3085 015122 000176 SWREG
3086 015124 000 001 .BYTE 0,1
3087 015126 104402 015176 TYPE, MCRLF
3088 015132 005037 015016 OUT: CLR @RDSW
3089 015136 000002 RTI
3090 015140 000001 SWREGC: 1
3091 015142 006 002 .BYTE 6,2
3092 015144 000176 SWREG
3093 015146 057377 000107 SCNTG: .ASCIZ <377>/↑G/
    
```

J05

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 61
 DZDQFD.P11 21-DEC-76 16:34 GENERAL UTILITIES (TYPE OUT, ERROR, SCOPE, ETC.)

3094	015152	051777	051127	020075	SMSWR: .ASCIZ <377>/SWR= /
3095	015160	000			
3096	015161	040	047040	053505	SMNEW: .ASCIZ / NEW= /
3097	015166	020075	000		
3098		015172			.EVEN
3099	015172	020040	000077		MOM: .ASCIZ / ?/
3100	015176	000377			MCRLF: .ASCIZ <377>
3101	015200	050377	051127	043040	MPFAIL: .ASCIZ <377>/PWR FAILED. RESTART AT TEST /
3102	015206	044501	042514	027104	
3103	015214	051040	051505	040524	
3104	015222	052122	040440	020124	
3105	015230	042524	052123	000040	
3106	015236	042777	042116	050040	MEPASS: .ASCIZ <377>/END PASS DZDQF /
3107	015244	051501	020123	055104	
3108	015252	050504	020106	000040	
3109	015260	051377	000		MR: .ASCIZ <377>/R/
3110	015263	377	051120	043517	MERR2: .ASCIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT./
3111	015270	040522	020115	041111	
3112	015276	044504	040503	042524	
3113	015304	020123	047516	042040	
3114	015312	053105	041511	051505	
3115	015320	050040	042522	042523	
3116	015326	052116	000056		
3117	015332	044777	051516	043125	MERR3: .ASCIZ <377>/INSUFFICIENT DATA! /
3118	015340	044506	044503	047105	
3119	015346	020124	040504	040524	
3120	015354	000041			
3121	015356	052377	051505	020124	MTSTPC: .ASCIZ <377>/TEST PC- /
3122	015364	041520	000055		
3123	015370	046377	041517	020113	MLOCK: .ASCIZ <377>/LOCK ON SELECTED TEST /
3124	015376	047117	051440	046105	
3125	015404	041505	042524	020104	
3126	015412	042524	052123	000	
3127	015417	103	051123	020072	MCSRX: .ASCIZ /CSR: /
3128	015424	000			
3129	015426	126	041505	020072	MVECX: .ASCIZ /VEC: /
3130	015432	000			
3131	015433	120	051501	042523	MPASSX: .ASCIZ /PASSES: /
3132	015440	035123	000040		
3133	015444	051105	047522	051522	MERRX: .ASCIZ /ERRORS: /
3134	015452	020072	000		
3135	015455	377	052377	051505	MTSTN: .ASCIZ <377><377> /TEST NO: /
3136	015462	020124	047516	020072	
3137	015470	000			
3138	015471	377	042523	020124	MNEW: .ASCIZ <377>/SET SWITCH REG TO DQ11'S DESIRED ACTIVE. /
3139	015476	053523	052111	044103	
3140	015504	051040	043505	052040	
3141	015512	020117	050504	030461	
3142	015520	051447	042040	051505	
3143	015526	051111	042105	040440	
3144	015534	052103	053111	027105	
3145	015542	000			
3146	015543	120	035103	000040	MERRPC: .ASCIZ /PC: /
3147	015550	046777	050101	047440	XHEAD: .ASCIZ <377>/MAP OF DQ11 STATUS/<377>
3148	015556	020106	050504	030461	
3149	015564	051440	040524	052524	

K05

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 62
 DZDQFD.P11 21-DEC-76 16:34 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

3150	015572	177523	000	
3151		015576		.EVEN
3152	015576	000002		XSTATQ: 2
3153	015600	006	003	.BYTE 6,3
3154	015602	001244		TEMP1
3155	015604	006	002	.BYTE 6,2
3156	015606	001246		TEMP2
3157				.EVEN
3158				
3159				;BUFFERS FOR INPUT-OUTPUT
3160				
3161	015610	000000		INBUF: 0
3162		015652		.=. +40
3163	015652	000000		TEMP: 0
3164		015714		.=. +40
3165	015714	000000		MDATA: 0
3166		015756		.=. +40
3167	015756	000000		.ERRTA: 0
3168	015760	000000		0 ;HALT 0
3169	015762	000000		0
3170	015764	016132		EM0
3171	015766	000000		0 ;HALT 1
3172	015770	000000		0
3173	015772	016252		EM1
3174	015774	016521		DH0 ;HALT 2
3175	015776	017166		DT1
3176	016000	016317		EM2
3177	016002	016562		DH1 ;HALT 3
3178	016004	000000		0
3179	016006	016317		EM2
3180	016010	016601		DH2 ;HALT 4
3181	016012	000000		0
3182	016014	016317		EM2
3183	016016	016622		DH3 ;HALT 5
3184	016020	000000		0
3185	016022	016317		EM2
3186	016024	016653		DH4 ;HALT 6
3187	016026	000000		0
3188	016030	016317		EM2
3189	016032	016701		DH5 ;HALT 7
3190	016034	000000		0
3191	016036	016317		EM2
3192	016040	016735		DH6 ;HALT 10
3193	016042	000000		0
3194	016044	016317		EM2
3195	016046	016766		DH7 ;HALT 11
3196	016050	000000		0
3197	016052	016317		EM2
3198	016054	017006		DH10 ;HALT 12
3199	016056	000000		0
3200	016060	016317		EM2
3201	016062	017033		DH11 ;HALT 13
3202	016064	000000		0
3203	016066	017062		DH12
3204	016070	000000		0 ;HALT 14
3205	016072	000000		0

LOS

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 63
 DZDQFD.P11 21-DEC-76 16:34 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

3206	016074	016341					EM3		
3207	016076	017141					DH14	;HALT 15	
3208	016100	017212					DT3		
3209	016102	016420					EM4		
3210	016104	017116					DH13	;HALT 16	
3211	016106	017200					DT2		
3212	016110	016444					EM5		
3213	016112	017141					DH14	;HALT 17	
3214	016114	017212					DT3		
3215	016116	016471					EM6		
3216	016120	000000					0	;HALT 20	
3217	016122	000000					0		
3218	016124	017062					DH12		
3219	016126	017152					DH15	;HALT 21	
3220	016130	000000					0		
3221	016132	051377	041505	044505	EM0:	.ASCIZ	<377>/RECEIVER DONE NOT SET!/ <377>/T IE CHARACTER DETECT OPTION <BB> IS NOT INSTALLED!!!!/		
	016162	052377	042510	041440	MSG13:	.ASCIZ	<377>/CHARACTER DETECTION TEST <SET FLAG>/		
	016252	041777	040510	040522	EM1:	.ASCIZ	<377>/TEST OF SEQ REG /		
	016317	377	042524	052123	EM2:	.ASCIZ	<377>/BCC TEST-APPEND FAILURE. DQ11 ERROR FLAG SET./		
	016341	377	041502	020103	EM3:	.ASCIZ	<377>/DATA COMPARE ERROR/ <377>/DQ11 ERROR FLAG SET/ <377>/NO RECIEVER INTERUPTS./		
	016420	042377	052101	020101	EM4:	.ASCIZ	<377>/CHAR RECEIVED EXPECTED ADDRESS/		
	016444	042377	030521	020061	EM5:	.ASCIZ	<377>/BIT01 -SET T-/ <377>/BIT02 -CLEAR T-/ <377>/BIT03 -BCC START CLEAR-/ <377>/BIT06 -CLEAR ACTIVE-/ <377>/BIT07 -SET DONE; CLEAR GO-/ <377>/BIT08 -CHARACTER STRIP-/ <377>/BIT10 -TX PAD-/ <377>/BIT11 -BCC EXCLUDE-/ <377>/BIT08 -RX CHAR STRIP-/ <377>/MULTIPLE FUNCTIONS FAILURE/ <377>/EXPECTED RECEIVED/ <377>/DQERR / <377>/DLE STUCK /		
	016471	377	047516	051040	EM6:	.ASCIZ			
	016521	377	044103	051101	DH0:	.ASCIZ			
	016562	041377	052111	030460	DH1:	.ASCIZ			
	016601	377	044502	030124	DH2:	.ASCIZ			
	016622	041377	052111	031460	DH3:	.ASCIZ			
	016653	377	044502	030124	DH4:	.ASCIZ			
	016701	377	044502	030124	DH5:	.ASCIZ			
	016735	377	044502	030124	DH6:	.ASCIZ			
	016766	041377	052111	030061	DH7:	.ASCIZ			
	017006	041377	052111	030461	DH10:	.ASCIZ			
	017033	377	044502	030124	DH11:	.ASCIZ			
	017062	046777	046125	044524	DH12:	.ASCIZ			
	017116	042777	050130	041505	DH13:	.ASCIZ			
	017141	377	050504	051105	DH14:	.ASCIZ			
	017152	042377	042514	051440	DH15:	.ASCIZ			
					.EVEN				
	017166	000002			DT1:	2			
3222	017170	006	011		.BYTE	6,9.			
3223	017172	013002			GDCHAR				
3224	017174	002	002		.BYTE	2,2			
3225	017176	013000			ADDR				
3226	017200	000002			DT2:	2			
3227	017202	003	006		.BYTE	3,6			
3228	017204	013002			GDCHAR				
3229	017206	003	002		.BYTE	3,2			
3230	017210	012774			CHAR				
3231	017212	000001			DT3:	1			
3232	017214	006	002		.BYTE	6,2			
3233	017216	012766			ERR				
3234		000001			.END				

H06

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 74
 DZDQFD.P11 21-DEC-76 16:34 CROSS REFERENCE TABLE -- MACRO NAMES

DOEND	18	2651													
DQFRNT	18	532													
HLT	5968	1416	1421	1475	1480	1734	1740	1773	1777	1819	1870	1879	1916	1920	1923
	1962	1966	1969	1972	1997	2011	2042	2056	2116	2125	2172	2176	2213	2217	2253
	2256	2259	2262	2315	2319	2339	2355	2371	2445	2448	2451	2454	2458	2462	2467
	2470	2473	2603	2618											
IDENT	18														
ORANGE	18	1303													
TESTA1	18														
TESTB1	18														
TESTC1	18														
TESTD1	18														
TESTE1	18														
TESTF1	18	50													
TESTH1	18														
TESTH2	18														
\$PEGGE	13578	2509													
\$EGIN	18	1273													
\$BUFFE	18	3158													
\$CATCH	18	649													
\$CLRVE	18	1238													
\$CONVR	18	2883													
\$EOP	18	2651													
\$EXTSU	13578	2536													
\$GETFL	18														
\$GETPA	18	1291													
\$HEADE	18	532													
\$HLT	18	2953													
\$INSTR	18	2762													
\$INTNP	18														
\$MAINT	18														
\$MSG	18	3099													
\$PARAM	18	2800													
\$PAIL	18	3031													
\$REG	18	2859													
\$SCOPE	18	2706													
\$SCOPI	18	2730													
\$SETFL	18														
\$SETVE	18	907													
\$START	18	1193													
\$SYMB0	18	549													
\$TRAPS	18	1096													
\$TRPDE	18	1105	1107	1109	1111	1113	1115	1117	1119	1121	1123	1125	1127	1129	1131
\$TRPSR	18	2940													
\$STSN	18	1303	1359	1374	1438	1500	1513	1526	1539	1552	1565	1578	1591	1604	1617
	1630	1643	1656	1669	1682	1695	1750	1791	1833	1890	1935	1982	2027	2072	2136
	2186	2225	2273	2329	2345	2361	2402								
\$TYPE	18	2738													
\$VARIA	18	1034													
\$XSCOP	13578	1422	1481	1905	1953	2014									
\$YIPPY	13578	2325	2341	2357											

. ABS. 017220 000

DZDQF MACY11 27(1006) 22-DEC-76 11:36 PAGE 75
DZDQFD.P11 21-DEC-76 16:34 CROSS REFERENCE TABLE -- MACRO NAMES

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

MULE:DZDQFD.BIN,MULE:DZDQFD.SEQ/SOL/CRF=DSKZ:UNIV.P11,DSKZ:DZDQFD.P11
RUN-TIME: 21 32 3 SECONDS
RUN-TIME RATIO: 167/57=2.9
CORE USED: 20K (39 PAGES)

EOF1DZDQFDSEQ 00010000 770325 PDP10 411