

# DQ11

SEQUENCE REGISTER TESTS  
MD-11-DZDQF-C

EP-DZDQF-C-DL-A  
COPYRIGHT © 1976  
FICHE 1 OF 1

NOV 1976  
**digital**  
MADE IN U.S.A.

The microfiche card contains a grid of 24 frames of test data, arranged in 6 rows and 4 columns. Each frame displays a sequence of data points, likely representing the output of a sequence register test. The data is organized into columns, with some frames showing a header or title. The frames are separated by vertical and horizontal lines, creating a grid structure. The data appears to be binary or digital in nature, consistent with the 'SEQUENCE REGISTER TESTS' title.

11-11-76

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDQF-C-D  
PRODUCT NAME: SEQUENCE REGISTER TESTS  
DATE: 21 JUNE 76  
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, 1976 BY DIGITAL EQUIPMENTS 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. DZDQA [REV] BASIS R/W TEST #1
2. DZDQB [REV] BASIC R/W TEST #2
3. DZDQC [REV] BASIC NPR AND INTERRUPT TEST
4. DZDQD [REV] RECEIVER TRANSMITTER EXERCISER TEST
5. DZDQE [REV] MISC. RX AND TX TESTS. PLUS BCC TESTS.
6. DZDQF [REV] CHARACTER DETECT TESTS.
7. DZDQH [REV] CHARACTER LENGTH AND INTERRUPT TESTS.

THERE IS ALSO AN ONLINE TEST TO BE DISCUSSED LATER.

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

AND A PARAMETER INPUT PROGRAM IS AVAILABLE

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

## 2. REQUIREMENTS

## 2.1 EQUIPMENT

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

## 2.2 STORAGE

PROGRAM WILL LOAD AND RUN  
IN 4K OF MEMORY.

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

### 3. LOADING PROCEEDURE

#### 3.1 METHOD

ALL PROGRAMS ARE IN ABSOLUTE FORMAT AND ARE LOADED USING THE ABSOLUTE LOADER.

ABSOLUTE LOADER STARTING ADDRESS \*500

MEMORY \*  
SIZE

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

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

3.1.2 THEN START

### 4. STARTING PROCEEDURE

A. LOAD LOC. 200

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

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

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

C. THEN START

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

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

THE ABOVE IS ONLY AN EXAMPLE!

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

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

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

#### 4.1 CONTROL SWITCH SETTINGS

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

#### CONTROL:

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

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

SW 15	SET: HALT ON ERROR
SW 14	SET: LOOP ON CURRENT TEST
SW 13	SET: INHIBIT ERROR PRINT OUT
SW 12	SET: INHIBIT TYPE OUT/BELL ON ERROR.
SW 11	SET: INHIBIT ITERATIONS

SW 10 SET: ESCAPE TO NEXT TEST  
 SW 09 SET: LOOP WITH CURRENT DATA  
 SW 08 SET: CATCH ERROR AND LOOP ON IT  
 SW 07 SET: USE PREVIOUS STATUS TABLE. CLR-DO AUTO SIZE.  
 SW 06 SET:  
 SW 05 SET:  
 SW 04 SET:  
 SW 03 SET:  
 SW 02 SET: LOCK ON SELECTED TEST  
 SW 01 SET: RESTART PROGRAM AT SELECTED TEST  
 SW 00 SET: RESELECT DQ11'S DESIRED ACTIVE.

## 4.1.2 SWITCH REGISTER RESTRICTIONS

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

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

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

SW 09 LOOP ON CURRENT DATA:  
 THIS SWITCH WILL ONLY WORK IF  
 CALL "SCOPI" IS IN THAT TEST.  
 THE REASON BEING THAT MOST TESTS  
 DEAL WITH BLOCKS OF DIFFERENT DATA  
 TO BE SENT OR RECEIVED ALL AT ONCE  
 THUS IN BLOCK DATA; ONE PATTERN CANN'T BE SINGLED OUT.

## 4.1.3 SWITCH REGISTER PRIORITYS

## ERROR SWITCHES

1. SW 12 DELETE PRINT OUT/BELL ON ERROR.
2. SW 13 DELETE ERROR PRINTOUT.
3. SW 15 HALT ON THE ERROR.
4. SW 08 GOTO BEGINNING OF THE TEST.
5. SW 10 GOTO NEXT TEST ON ERROR.

\*\*\*\*HLT (ERROR) ROUTINE SUPPORTS <↑G> OPERATION\*\*\*\*

## SCOPE SWITCHES

1. SW 09 (IF ENABLED BY "SCOPI")
2. SW 14
3. SW 11

\*\*\*\*SCOPE ROUTINE WILL SUPPORT <↑G> OPERATION\*\*\*\*

## 4.2 STARTING ADDRESS

STARTING ADDRESS IS AT 000200  
THERE ARE NO OTHER STARTING ADDRESSES  
FOR THE DQ11 DIAGNOSTICS PREVIOUSLY MENTIONED

NOTE: IF ADDRESS 000042 IS NON-ZERO  
THE PROGRAM ASSUMES IT IS UNDER  
ACT11 OR DDP CONTROL AND WILL ACT ACCORDINGLY  
AFTER \*ALL\* AVAILABLE DQ11'S ARE TESTED  
THE PROGRAM WILL RETURN TO "DDP2" OR "ACT-11".

## 5. OPERATING PROCEDURE

WHEN PROGRAM IS INITIALLY STARTED MESSAGES AS DESCRIBED IN SECTION  
FOUR WILL BE PRINTED.

AND PROGRAM WILL BEGIN RUNNING THE  
DIAGNOSTIC

## 5.2 PROGRAM AND/OR OPERATOR ACTION

THE TYPICAL APPROACH SHOULD BE

1. HALT ON ERROR (VIA SW 15=1)  
WHEN EVER AN ERROR OCCURS
2. CLEAR SW 15
3. SET SW 14: (LOOP ON THIS TEST)
4. SET SW 13: (INHIBIT ERROR PRINT OUT)

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

CAN BE INTERPEDITED

## 6. ERRORS

AS DESCRIBED PREVIOUSLY THERE WILL ALWAYS BE A TEST NUMBER AND PC TYPED OUT AT THE TIME OF AN ERROR (PROVIDING SW 13=0 AND SW 12=0). IN MOST CASES ADDITIONAL INFORMATION WILL BE SUPPLIED THE THE ERROR MESSAGE WHICH IS TO GIVE THE OPERATOR AN INDICATION OF THE ERROR.

### 6.2 ERROR RECOVERY

IF FOR SOME REASON THE DQ11 SHOULD "HANG THE BUS" (GAIN CONTROL OF BUS SO THAT CONSOLE MANUAL FUNCTIONS ARE INHIBITED) AN INIT OR POWER DOWN/UP IS NECESSARY FOR OPERATOR TO REGAIN CONTROL OF CPU.  
IF THIS SHOULD HAPPEN; LOOK IN LOCATION "TSTNO" (ADDRESS 1222) FOR THE NUMBER OF THE TEST THAT WAS RUNNING AT THE TIME OF THE CATASTROPHIC ERROR.  
IN THIS WAY THE OPERATOR WILL HAVE AN IDEA AS TO WHAT THE DQ11 WAS DOING AT THE TIME OF THE ERROR.

### 6.3 \*\*\*\*HALT RECOVERY WHEN USING SOFTWARE SWITCH REGISTER\*\*\*\*

IF THE SOFTWARE SWITCH REGISTER IS TO BE CHANGED AFTER A HALT THE THE OPERATOR IS REQUIRED TO TYPE A <↑G> BEFORE DEPRESSING CONTINUE.  
THE FOLLOWING WILL BE TYPED:  
SWR=XXXXXX NEW= (REFER TO SECTION 4.1 FOR OPERATOR OPTION)

## 7. RESTRICTIONS

### 7.1 STARTING RESTRICTIONS

SEE SECTION 4. (PLEASE)

### 7.2 OPERATING RESTRICTIONS

DQ11 TRIAL PROGRAM MUST BE RUN PRIOR TO THE FIRST AND ONLY THE FIRST RUNNING OF ANY DQ11 DIAGNOSTIC  
NOTE: IF NO PROGRAM OTHER THAN A DQ11 DIAGNOSTIC WAS LOADED AFTER DQ11 TRIAL OR IF CORE MEMORY HAS NOT BEEN CHANGED; OR IF THERE IS NO DQ11 CONFIGURATION CHANGES; THE DQ11 TRIAL PROGRAM NEED NEVER BE RUN AGAIN.  
HOWEVER IF ANY OF THE ABOVE HAVE BEEN VIOLATED THE DQ11 TRIAL PROGRAM MUST LE 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



DZDQF MACY11 27(732) 24-SEP-76 10:17 PAGE 9  
 DZDQFC.P11

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 DZDQF-C CSR: 160000 VEC: 300 PASSES: 000001 ERRORS: 000000

NOTE: THE NUMBERS FOR CSR AND VEC ARE  
 NOT NECESSARILY THE VALUES FOR THE DEVICE  
 THEY ARE ONLY FOR THIS EXAMPLE.

8.3 TST1 (MINI MONITOR)

THE VERY FIRST "TEST" (TST1)  
 IS \*NOT\* A TEST OF THE DQ11 HARDWARE  
 IT IS A MINI-MONITOR USED TO CYCLE DQ11 IN THE  
 SYSTEM THROUGH THE DIAGNOSTIC.

REMEMBER: TST1 IS NOT A TEST OF DQ11 HARDWARE!!!!!!!

8.4 KEY LOCATIONS

RETURN (1210) CONTAINS THE ADDRESS WHERE PROGRAM WILL  
 RETURN WHEN ITERATION COUNT IS REACHED  
 OR IF LOOP ON TEST IS ASSERTED.

NEXT (1212) CONTAINS THE ADDRESS OF THE NEXT TEST  
 TO BE PERFORMED.

TSTNO (1222) CONTAINS THE NUMBER OF THE TEST NOW  
 BEING PERFORMED.

RUN (1272) THE BIT IN "RUN" ALWAYS POINTS ONE  
 PAST THE DQ11 CURRENTLY BEING TESTED.

EXAMPLE:

(RUN) 1272/0000000001000000

MEANS THAT DQ11 NO.05 IS THE DQ11 NOW  
 RUNNING.

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

THESE LOCATIONS CONTAIN THE INFORMATION  
 NEEDED TO TEST UP TO 16 (DECIMAL) DQ11S  
 SEQUENTIALY. 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 "3B" 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 APPRIOATE DQSTXX: AFTER AUTO SIZING

8.5.8 SET FOR ODD OR EVEN PARITY?

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

8.5.9 FINDING THE VECTOR.

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

9. PROGRAM DESCRIPTION

CONTAINED WITHIN LISTING

10. LISTING

FOLLOWING

532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567

;MAINDEC-11-DZDQF-C/<377>/SEQUENCE REG TEST  
;COPYRIGHT 1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

;REVISED 21-JUNE-76 BY S. CARPENTER  
;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-C/<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

100000  
040000  
020000  
010000  
004000  
002000  
001000  
000400  
000100  
000040  
000020  
000010  
000004  
000002  
000001

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

```

568
569
570           ;REGISTER DEFINITIONS
571
572           000000      R0=%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           ;LOCATION EQUIVALENCIES
582
583           177570      DSWR= 177570    ;HARDWARE SWITCH REGISTER LOC.
584           177570      DLIGHTS=177570 ;HARDWARE DISPLAY REGISTER LOC.
585           177776      PS=177776      ;PROCESSOR STATUS WORD
586           001200      STACK=1200     ;START OF PROCESSOR STACK
587
588           ;INSTRUCTION DEFINITIONS
589
590           005746      PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD
591           005726      POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD
592           010046      PUSHRO=10046     ;SAVE R0 ON STACK
593           012600      POPRO=12600     ;RESTORE R0 FROM STACK
594           024646      PUSH2SP=24646   ;DECREMENT STACK TWICE
595           022626      POP2SP=22626    ;INCREMENT STACK TWICE
596           .EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL
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           ;DQ11 OPTIONAL DEFINITIONS
617
618           002000      ABBIT=2000
619           004000      ACTBIT=4000
620           010000      BABIT=10000
621           020000      BBBIT=20000
622           040000      JUMBIT=40000
623

```

624 001000 ODDBIT=1000  
 625 100000 SYNBIT=100000  
 626  
 627  
 628

;DQ11 SECONDARY REGISTER DEFINATIONS

629			
630	000000	RXBA.P=0	;RECEIVER BUS ADDRESS PRIMARY.
631	000001	RXWC.P=1	;RECEIVER WORD COUNT PRIMARY.
632	000002	TXBA.P=2	;TRANSMITTER BUS ADDRESS PRIMARY.
633	000003	TXWC.P=3	;TRANSMITTER BUS ADDRESS PRIMARY.
634	000004	RXBA.S=4	;RECEIVER BUS ADDRESS SECONDARY.
635	000005	RXWC.S=5	;RECEIVER WORD COUNT SECONDARY.
636	000006	TXBA.S=6	;TRANSMITTER BUS ADDRESS SECONDARY.
637	000007	TXWC.S=7	;TRANSMITTER WORD COUNT SECONDARY.
638			
639	000010	CHARDT=10	;CHARACTER DETECT REGISTER.
640	000011	SYNC.=11	;SYNC REGISTER.
641	000012	MISC.=12	;MISCELLANEOUS REGISTER.
642	000013	TX.MUX=13	;TRANSMITTER MUX REGISTER.
643	000014	SEQ.=14	;SEQUENCE REGISTER.
644	000015	RX.BCC=15	;RECEIVER BCC REGISTER.
645	000016	TX.BCC=16	;TRANSMITTER BCC REGISTER.
646	000017	POLY.=17	;POLYNOMIAL REGISTER.
647			
648			

```

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

```

DZDQF MACY11 27(732) 24-SEP-76 10:17 PAGE 16  
 DZDQFC.P11 TRAPCATCHER FOR UNEXPECTED INTERRUPTS

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



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

DZDQF MACY11 27(732) 24-SEP-76 10:17 PAGE 18  
 DZDQFC.P11 TRAPCATCHER FOR UNEXPECTED INTERRUPTS

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

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

```

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                                     000056
928 000056
929 000056 010120                       VECMAP:
930 000060 012721 000004                1$:  MOV R1,(R0)+                ;START FILLING THE VECTOR AREA
931 000064 022021                       MOV #4,(R1)+                  ;WITH .+2; IOT (4)
932 000066 020127 001000                CMP (R0)+,(R1)+              ;UPDATE THE POINTERS
933 000072 101771                       CMP R1,#1000                 ;IS ALL FLOATING VECTOR AREA DONE
934 000074 012737 000146 000020        BLOS 1$                      ;BR IF NOT ALL DONE
935 000102 013737 001500 001244        MOV #4$,@#20                 ;SET FOR IOT TRAP BY DQ11
936 000110 006037 001244                MOV DQACTV,TEMP1            ;GET THE ACTIVE DQ11 S
937 000114 103023                       ROR TEMP1                    ;ARE YOU ACTIVE... DQ11
938 000116 005037 177776                BCC 5$                      ;IF CARRY CLEAR.. NO MORE DQ11S
939 000122 005722                       CLR PS                       ;CLEAR PS
940 000124 012772 000340 177776        TST (R2)+                   ;PUT POINTER TO STATUS TABLE
941 000132 105200                       MOV #340,@-2(R2)            ;TRY AND SET PRI/SEC DONE AND IE
942 000134 001376                       INCB RO                      ;DELAY.....DELAY
943 000136 112712 000300                BNE .-2                     ;.....DELAY
944 000142 005722                       MOV #300,(R2)               ;NO INTERRUPT ASSUME 300 FIX IN TEST C
945 000144 000761                       TST (R2)+                   ;UPDATE POINTERS
946 000146 051612                       BR 2$                       ;GO DO IT AGAIN
947 000150 042712 000007                BIS (SP),(R2)                ;ENTERD BY IOT TRAP BY DQ11
948 000154 022626                       BIC #7,(R2)                 ;CLEAR UNWANTED BITS
949 000156 012716 000142                CMP (SP)+,(SP)+             ;POP IOT JUNK OFF STACK
950 000162 000002                       MOV #3$, (SP)               ;SET RETURN PC ON STACK
951 000164 000207                       RTI                          ;GO HOME
952                                     5$:  RTS PC                    ;ALL SIZING IS DONE
953
954                                     ;****SOFTWARE SWITCH REGISTER****
955                                     . =174
956 000174 000000                       DISPREG: 0                   ;SOFTWARE DISPLAY REGISTER
957 000176 000000                       SWREG: 0                     ;SOFTWARE SWITCH REGISTER
958
959                                     ;PROGRAM START
960                                     . =200
961 000200 000137 001512                JMP .START                   ;GO TO START OF PROGRAM
962

```



DZDQF MACY11 27(732) 24-SEP-76 10:17 PAGE 22  
 DZDQFC.P11 ROUTINES USED FOR AUTO SIZING.

```

1019 000540 012737 000006 000004      MOV      #6, D#4      ;RESET TIME OUT VECTOR
1020 000546 013737 001500 001502      MOV      DQACTV, SAVACT ;SAVE ACTIVE
1021 000554 012737 000340 000022      MOV      #340, D#22   ;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$:     TYPE          ;TYPE MESSAGE
1027 000604 015263      MERR2    ;I DIDN'T FIND ANY DQ115. DON'T USE AUTO SIZE.
1028 000606 005000      CLR      R0          ;
1029 000610 000000      HALT     ;HOW CAN I TEST NO DQ115
1030 000612 000776      BR       -2          ;DON'T LET OPR HIT CONT. SW
1031 000614 012716 000466      5$:     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-DZDQF-C/<377>/SEQUENCE REG TEST/<377>
1037 001006 042504 026503 030461
1038 001014 042055 042132 043121
1039 001022 041455 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

```

DZDGF MACY11 27(732) 24-SEP-76 10:17 PAGE 23  
 DZDGF.C.P11 PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

1075	001254	000000	TEMPS:	0		; TEMPORARY STORAGE
1076	001256	000000	SAVR0:	00		; R0 STORAGE
1077	001260	000000	SAVR1:	00		; R1 STORAGE
1078	001262	000000	SAVR2:	00		; R2 STORAGE
1079	001264	000000	SAVR3:	00		; R3 STORAGE
1080	001266	000000	SAVR4:	00		; R4 STORAGE
1081	001270	000000	SAVR5:	00		; R5 STORAGE
1082	001272	000000	SAVSP:	00		; STACK POINTER STORAGE
1083	001274	000000	SAVPC:	00		; PROGRAM COUNTER STORAGE
1084	001276	000000	SAVNUM:	0		
1085	001300	000001	CREAM:	.BLKW	1	
1086	001302	000000	RUNFLG:	0		
1087	001304	000000	RUN:	0		
1088	001306	000000	RUNCNT:	0		

```

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 $Y=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 DQRVEC: 0 ;POINTER TO DQ11 RECEIVER INTERRUPT VECTOR
1140 001352 000000 DQRLVL: 0 ;POINTER TO DQ11 RECEIVER INTERRUPT SERVICE PS
1141 001354 000000 DQTVEC: 0 ;POINTER TO DQ11 TRANSMITTER INTERRUPT VECTOR
1142 001356 000000 DQTLVL: 0 ;POINTER TO DQ11 TRANSMITTER INTERRUPT SERVICE PS
1143 001360 000000 DQRCSR: 0 ;POINTER TO DQ11 RECEIVER CONTROL REGISTER
1144 001362 000000 DQRCSH: 0 ;POINTER TO HIGH BYTE OF DQ11 RECEIVER CONTROL REGISTER

```



DZDQF MACY11 27(732) 24-SEP-76 10:17 PAGE 25  
 DZDQFC.P11 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          ; DQ11 STATUS TABLE AND ADDRESS ASSIGNMENTS
1154
1155          . = 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          DQNUM:  .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

```

## M02

DZDQF MACY11 27(732) 24-SEP-76 10:17 PAGE 26  
 DZDQFC.P11 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 DQNUM, SAVNUM
1205 001540 105037 001311 CLR STFLG ;CLEAR START FLAG
1206 001544 005037 001230 CLR PASCNT ;CLEAR PASS COUNT
1207 001550 105037 001312 CLR 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, TSTNO ;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 #DSWR, 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 65$
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$: TSTB @SWR
1239 001746 100402 BMI .+6
1240 001750 004737 000220 JSR PC, CSRMAP
1241 001754 104402 015550 TYPE ,XHEAD
1242 001760 012737 001400 001244 MOV #1400, TEMP1
1243 001766 017737 177252 001246 MOV @TEMP1, TEMP2
1244 001774 001406 BEQ .+16
1245 001776 104410 CONVRT
1246 002000 015576 XSTATQ
1247 002002 062737 000002 001244 ADD #2, TEMP1
1248 002010 000766 BR .-22
1249 002012 032777 000001 177160 12$: BIT #SW00, @SWR
1250 002020 001424 BEQ 1$
1251 002022 104402 TYPE
1252 002024 015471 MNEW
1253 002026 005000 CLR R0
1254 002030 000000 HALT
1255 002032 104414 CKSWR
1256 002034 027737 177140 001502 CMP @SWR, SAVACT

```

```

1257 002042 101404      BLOS      11$
1258 002044 104402      TYPE
1259 002046 015332      MERR3
1260 002050 000000      HALT
1261 002052 000776      BR      -2
1262 002054 017737 177120 001500 11$:  MOV      @SWR,DQACTV
1263 002062 013700 001500      MOV      DQACTV,RO
1264 002066 000000      HALT
1265 002070 104414      CKSWR
1266 002072 012700 000300 1$:  MOV      #300,RO
1267 002076 012701 000302      MOV      #302,R1
1268 002102 010120 2$:  MOV      R1,(R0)+
1269 002104 005021      CLR      (R1)+
1270 002106 022021      CMP      (RO)+,(R1)+
1271 002110 022700 001000      CMP      #1000,RO
1272 002114 001372      BNE      2$
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 <↑G>
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 000207      RETURN
1297 002232 001      .BYTE 1
1298 002233 001      .BYTE 1
1299 002234 000403      BR      4$
1300 002236 012737 002254 001214 3$:  MOV      #TST1,RETURN ;START AT TEST 1
1301 002244 104402 015260 4$:  TYPE      ,MR      ;TYPE R
1302 002250 000177 176740      JMP      @RETURN     ;START TESTING
1303      ; TEST 1
1304      ;*****
1305 002254 012737 000001 001226 TST1: MOV      #1,TSTNO
1306 002262 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 062737 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 001300 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$: TYPE ;ALLERT OPERATOR OF FATAL ERROR
1327 002412 015263 MERR2 ;NO DQ11 ACTIVE. WHY AM I HERE???
1328 002414 000000 HALT ;YOU MUST RELOAD DQ11 DIAGNOSTIC!!
1329 002416 000776 BR -2 ;STICK HERE ON CONT.
1330 002420 000257 3$: CCC ;CLEAR CPU COND. CODES
1331 002422 006137 001304 ROL RUN ;UPDATE RUN. ACTIVE DQ11 FOUND.
1332 002426 017737 176646 001506 MOV @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 @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,DQRCR
1337 002464 013737 001510 001350 MOV DQSTAT,DQVVEC
1338 002472 042737 177007 001350 BIC #177007,DQVVEC
1339 002500 013737 001350 001352 MOV DQVVEC,DQRLVL ;GENERATE ADDRESS OF RECEIVER INTERRUPT SERVICE PS
1340 002506 062737 000002 001352 ADD #2,DQRLVL
1341 002514 013737 001352 001354 MOV DQRLVL,DQVVEC ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT VECTOR
1342 002522 062737 000002 001354 ADD #2,DQVVEC
1343 002530 013737 001354 001356 MOV DQVVEC,DQTLVL ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT SERVICE PS
1344 002536 062737 000002 001356 ADD #2,DQTLVL
1345 002544 013737 001360 001362 MOV DQRCR,DQRCRSH
1346 002552 005237 001362 INC DQRCRSH ;GENERATE ADDRESS OF HIGH BYTE
1347 002556 013737 001360 001364 MOV DQRCR,DQTCR ;GENERATE ADDRESS OF TRANSMITTER CONTROL REGISTER
1348 002564 062737 000002 001364 ADD #2,DQTCR
1349 002572 013737 001364 001366 MOV DQTCR,DQERR ;GENERATE ADDRESS OF ERROR REGISTER
1350 002600 062737 000002 001366 ADD #2,DQERR
1351 002606 013737 001366 001370 MOV DQERR,DQREG ;GENERATE ADDRESS OF HIGH BYTE OF ERROR REGISTER
1352 002614 005237 001370 INC DQREG
1353 002620 013737 001370 001372 MOV DQREG,DQSEC ;GENERATE ADDRESS OF SECONDARY REGISTER
1354 002626 005237 001372 INC DQSEC
1355 002632 013737 001372 001374 MOV DQSEC,DQSECH ;GENERATE ADDRESS OF HIGH BYTE
1356 002640 005237 001374 INC DQSECH
1357 ;THIS IS NOT A TEST!!!
1358
1359 ; TEST 2
1360 *****
1361 002644 012737 000002 001226 TST2: MOV #2,TSTN
1362 002652 012737 002662 001216 MOV #TST3,NEXT
1363 ;THE ORIGINAL TEST 2 IS
1364 ;NOW TEST 2 OF TAPE DZDQH
1365 002660 104400 S PE
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 175266
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 #18,LOCK
BIT #BBBIT,DQSTAT ;DOES THIS DQ11 HAVE THE "BB" OPTION INSTALLED?
BNE 13$ ;BR IF YES
MOV #.EOP,RETURN ;GOTO END PASS
JMP @RETURN
13$: MEMCLR ;CLEAR ALL THE DQ11
6$: CLRB DETCAR+1 ;CLEAR THE CHARACTER STORAGE AREA
CLR GDCHAR ;SET FOR ERROR PRINTOUT
CLR ADDR ;SAME.
MOV #8,COUNT ;EIGHT BITS FOR EIGHT SHIFTS.
2$: MOV COUNT,R2 ;GET NUMBER OF SHIFTS PER CHAR.
CLRB @DQRC5H ;GET CHAR ADDR. ZERO
CLRB @DQREG ;GET RX BA PRI.
MOV #RXBUFF,@DQSEC ;LOAD IT
INCB @DQREG ;GET RX WC PRI.
MOV #200,@DQSEC ;LOAD IT
3$: CLRB @DQRC5H ;SELECT CHARACTER DET REG 0
MOVB DETCAR+1,GDCHAR ;
MOVB #10,@DQREG ;SELECT THE CHARACTER DET REGISTER.
MOV DETCAR,@DQSEC ;LOAD THE CHARACTER TO BE DETECTED
MOVB #14,@DQREG ;SELECT THE SEQUENCE REGISTER
MOV #BIT15+BIT13,@DQSEC ;SET SINGLE CHARACTER REC AND SET FLAG.
MOVB #12,@DQREG ;COMM
MOV #4012,@DQSEC ;SELECT EIGHT BITS TEST LOOP AND AUTO STEP
BIS #BIT12+BIT0,@DQRC5R ;SET RX ACTIVE AND RX GO
MOV DETCAR,TEMP ;MOV CHARACTER TO WORKING AREA
COMB TEMP+1 ;COMPLIMENT DATA FOR USE ON THE BIT WINDOW
4$: BIC #BIT7,@DQSEC ;IF PREVIOUSLY SET; CLEAR THE BIT WINDOW.
ROR TEMP ;SHIFT OUT THE BIT OF DATA.
MOV TEMP,R3 ;STORE CHAR
BIC #1<BIT7>,R3 ;CLEAR ALL UNWANTED BITS
BIS R3,@DQSEC ;PLACE DATA ON THE BIT WINDOW
INC @DQSEC ;CLOCK UP
DEC @DQSEC ;CLOCK DOWN
DEC R2 ;IS CHARACTER DONE YET
BNE 4$ ;BR IF NOT DONE
TST @DQRC5R ;WAS THE CHARACTER REALLY DETECTED?
BMI +4 ;BR IF GOOD
HLT 2 ;ERROR CHARACTER NOT DETECTED.
MOV @DQRC5R,TEMP ;GET THE RECEIVER CSR.
BIC #1<7400>,TEMP ;CLEAR ALL BUT THE CHARACTER DET. ADDR.
TST TEMP ;WAS THE CHAR DET. IN ADDR ZERO?
BEQ +4 ;
HLT 2 ;CHAR NOT DETECTED IN ADDR. ZERO..

:----- *LOCK* -----
SCOPI ;IF SW09=1; THEN GOTO ADDRESS IN "LOCK".
:-----

```

1425	003204	105237	013005	INCB	DETCAR+1	:HAVE I HIT MY LIMIT YET?
1426	003210	001260		BNE	2\$	:NO RETURN WITH UPDATED CHAR.
1427	003212	104400		SCOPE		:SCOPE TEST

```

:
: TEST THAT CHARACTERS FROM
: 400 TO 177400 CAN BE DETECTED.
: IN CHACTER DETECT ADDRESS ZERO.
: NOTE: SW09=1 WILL FREEZE ON CURRENT DATA.

```

: TEST 4

:\*\*\*\*\*

1441	003214	012737	000004	001226	TST4:	MOV	#4, TSTNO	
1442	003222	012737	003514	001216		MOV	#TST5, NEXT	
1443	003230	012737	003252	001220		MOV	#1\$, LOCK	
1444	003236	005037	013004			CLR	DETCAR	:SET CHAR TO ZERO
1445	003242	005037	013002			CLR	GDCHAR	:SET FOR ERROR
1446	003246	005037	013000			CLR	ADDR	:SAME
1447	003252				16:			
1448	003252	012702	000020		26:	MOV	#16, R2	:SET COUNT FOR 16 BIT CHARS
1449	003256	105077	176106		35:	CLR	JDQREG	:SEL THE RX BA PRI.
1450	003262	012777	012154	176102		MOV	#RXBUFF, JDQSEC	:LOAD IT
1451	003270	105277	176074			INCB	JDQREG	:SEL THE RX WC PRI.
1452	003274	012777	000200	176070		MOV	#200, JDQSEC	:LOAD IT
1453	003302	105077	176054			CLR	JDQRC5H	:GET CHAR ADD ZERO
1454	003306	112777	000010	176054		MOVB	#10, JDQREG	:GET CHAR ADDRESS
1455	003314	013777	013004	176050		MOV	DETCAR, JDQSEC	:LOAD THE CHARACTER TO BE DETCETED
1456	003322	112777	000014	176040		MOVB	#14, JDQREG	:GET THE SEQ REG.
1457	003330	012777	050000	176034		MOV	#BIT12+BIT14, JDQSEC	:LOAD DBL CHAR AND SET FLAG
1458	003336	112777	000012	176024		MOVB	#12, JDQREG	:SEL MISC REG.
1459	003344	012777	000012	176020		MOV	#BIT1+BIT3, JDQSEC	:SET TEST LOOP AND AUTO/STEP
1460	003352	052777	010001	176000		BIS	#BIT12+BIT0, JDQRC5R	:SET RX ACTIVE AND GO
1461	003360	013737	013004	015652		MOV	DETCAR, TEMP	:GET DATA CHAR.
1462	003366	013737	013004	013002		MOV	DETCAR, GDCHAR	:FOR ERROR
1463	003374	005137	015652			COM	TEMP	:PREPARE FOR BIT WINDOW
1464	003400	042777	000200	175764	45:	BIC	#BIT7, JDQSEC	:ZERO BIT WINDOW
1465	003406	000241				CLC		:CLEAR CARRY
1466	003410	005037	001244			CLR	TEMP1	
1467	003414	006037	015652			ROR	TEMP	
1468	003420	106037	001244			RORB	TEMP1	
1469	003424	053777	001244	175740		BIS	TEMP1, JDQSEC	:PLACE DATA ON BIT WINDOW
1470	003432	005277	175734			INC	JDQSEC	:CLOCK THE
1471	003436	005377	175730			DEC	JDQSEC	
1472	003442	005302				DEC	R2	:IS ALL THE CHAR DONE?
1473	003444	001355				BNE	4\$	:BR IF NO
1474	003446	005777	175706			TST	JDQRC5R	:DID THE FLAG SET?
1475	003452	100401				BMI	.+4	:BR IF YES
1476	003454	104002				HLT	2	:CHARACTER DET. FLAG NOT SET FOR DBL CHAR.
1477	003456	017737	175676	015652		MOV	JDQRC5R, TEMP	:GET THE RECEIVER CSR.
1478	003464	042737	170377	015652		BIC	#C<7400>, TEMP	:CLEAR ALL BUT THE CHARACTER DET. ADDR.
1479	003472	005737	015652			TST	TEMP	:WAS THE CHAR DET. IN ADDR ZERO?
1480	003476	001401				BEQ	.+4	
1481	003500	104002				HLT	2	:CHAR NOT DETECTED IN ADDR. ZERO..

```

1491          :----- *LOCK* -----
1492 003502 104401          SCOPI          ;IF SW09=1; THEN GOTO ADDRESS IN "LOCK".
1493          :-----
1494 003504 105237 013005  INCB  DETCAR+1  ;UPDATE THE DATA
1495 003510 001260          BNE  1$      ;ALL DONE?
1496 003512 104400          SCOPE          ;SCOPE WHEN ALL DATA DONE.
1497
1498
1499
1500          ;TEST THAT A CHARACTER CAN
1501          ;BE DETECTED IN ALL 16 CHARACTER
1502          ;DETECT ADDRESSES
1503          ;
1504          ;TEST THAT THE CHARACTER 255
1505          ;CAN BE DETECTED IN CHARACTER
1506          ;DETECT ADDRESS 00
1507
1508          ; TEST 5
1509          ;*****
1510 003514 012737 000005 001226  TST5:  MOV  #5, TSTNO
1511 003522 012737 003544 001216  MOV  #TST6, NEXT
1512 003530 012737 000000 013000  MOV  #00, ADDR          ;LOAD THE ADDRESS
1513 003536 004737 004314          JSR  PC, CHK.ADD        ;GO AND LOAD THE CHARACTER.
1514 003542 104400          SCOPE          ;SCOPE THIS TEST
1515
1516          ;TEST THAT THE CHARACTER 255
1517          ;CAN BE DETECTED IN CHARACTER
1518          ;DETECT ADDRESS 01
1519
1520          ; TEST 6
1521          ;*****
1522 003544 012737 000006 001226  TST6:  MOV  #6, TSTNO
1523 003552 012737 003574 001216  MOV  #TST7, NEXT
1524 003560 012737 000001 013000  MOV  #01, ADDR          ;LOAD THE ADDRESS
1525 003566 004737 004314          JSR  PC, CHK.ADD        ;GO AND LOAD THE CHARACTER.
1526 003572 104400          SCOPE          ;SCOPE THIS TEST
1527
1528          ;TEST THAT THE CHARACTER 255
1529          ;CAN BE DETECTED IN CHARACTER
1530          ;DETECT ADDRESS 02
1531
1532          ; TEST 7
1533          ;*****
1534 003574 012737 000007 001226  TST7:  MOV  #7, TSTNO
1535 003602 012737 003624 001216  MOV  #TST10, NEXT
1536 003610 012737 000002 013000  MOV  #02, ADDR          ;LOAD THE ADDRESS
1537 003616 004737 004314          JSR  PC, CHK.ADD        ;GO AND LOAD THE CHARACTER.
1538 003622 104400          SCOPE          ;SCOPE THIS TEST
1539
1540          ;TEST THAT THE CHARACTER 255
1541          ;CAN BE DETECTED IN CHARACTER

```

F03

```

1537                                     ;DETECT ADDRESS 03
1538                                     ;
1539                                     ; TEST 10
1540                                     ;*****
1541 003624 012737 000010 001226 †TST10: 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                                     ;
1549                                     ;TEST THAT THE CHARACTER 255
1550                                     ;CAN BE DETECTED IN CHARACTER
1551                                     ;DETECT ADDRESS 04
1552                                     ;
1553                                     ; TEST 11
1554                                     ;*****
1555 003654 012737 000011 001226 †TST11: MOV #11,TSTNO
1556 003662 012737 003704 001216      MOV #TST12,NEXT
1557 003670 012737 000004 013000      MOV #04,ADDR ;LOAD THE ADDRESS
1558 003676 004737 004314      JSR PC,CHK.ADD ;GO AND LOAD THE CHARACTER.
1559 003702 104400      SCOPE ;SCOPE THIS TEST
1560
1561                                     ;
1562                                     ;TEST THAT THE CHARACTER 255
1563                                     ;CAN BE DETECTED IN CHARACTER
1564                                     ;DETECT ADDRESS 05
1565                                     ;
1566                                     ; TEST 12
1567                                     ;*****
1568 003704 012737 000012 001226 †TST12: MOV #12,TSTNO
1569 003712 012737 003734 001216      MOV #TST13,NEXT
1570 003720 012737 000005 013000      MOV #05,ADDR ;LOAD THE ADDRESS
1571 003726 004737 004314      JSR PC,CHK.ADD ;GO AND LOAD THE CHARACTER.
1572 003732 104400      SCOPE ;SCOPE THIS TEST
1573
1574                                     ;
1575                                     ;TEST THAT THE CHARACTER 255
1576                                     ;CAN BE DETECTED IN CHARACTER
1577                                     ;DETECT ADDRESS 06
1578                                     ;
1579                                     ; TEST 13
1580                                     ;*****
1581 003734 012737 000013 001226 †TST13: MOV #13,TSTNO
1582 003742 012737 003764 001216      MOV #TST14,NEXT
1583 003750 012737 000006 013000      MOV #06,ADDR ;LOAD THE ADDRESS
1584 003756 004737 004314      JSR PC,CHK.ADD ;GO AND LOAD THE CHARACTER.
1585 003762 104400      SCOPE ;SCOPE THIS TEST
1586
1587                                     ;
1588                                     ;TEST THAT THE CHARACTER 255
1589                                     ;CAN BE DETECTED IN CHARACTER
1590                                     ;DETECT ADDRESS 07
1591                                     ;
1592                                     ; TEST 14
1593                                     ;*****

```



```

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 004C12 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
1617 ; TEST 16
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
1630 ; TEST 17
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
1643 ; TEST 20
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.

```

# H03

DZDQF MACY11 27(732) 24-SEP-76 10:17 PAGE 34  
DZDQFC.P11 PROGRAM INITIALIZATION AND START UP.

```
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 TST21: MOV #21,TSTNO
1659 004162 012737 004204 001216 MOV #TST22,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 ;
1665 ;TEST THAT THE CHARACTER 255
1666 ;CAN BE DETECTED IN CHARACTER
1667 ;DETECT ADDRESS 15
1668 ;
1669 ; TEST 22
1670 ;*****
1671 004204 012737 000022 001226 TST22: MOV #22,TSTNO
1672 004212 012737 004234 001216 MOV #TST23,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 ;
1678 ;TEST THAT THE CHARACTER 255
1679 ;CAN BE DETECTED IN CHARACTER
1680 ;DETECT ADDRESS 16
1681 ;
1682 ; TEST 23
1683 ;*****
1684 004234 012737 000023 001226 TST23: MOV #23,TSTNO
1685 004242 012737 004264 001216 MOV #TST24,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 ;
1691 ;TEST THAT THE CHARACTER 255
1692 ;CAN BE DETECTED IN CHARACTER
1693 ;DETECT ADDRESS 17
1694 ;
1695 ; TEST 24
1696 ;*****
1697 004264 012737 000024 001226 TST24: MOV #24,TSTNO
1698 004272 012737 004562 001216 MOV #TST25,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 ;
1704 004314 104413 CHK.ADD: MEMCLR ;CLEAR ALL THE DQ11 REGISTERS.
```

DZDQF MACY11 27(732) 24-SEP-76 10:17 PAGE 35  
 DZDQFC.P11 PROGRAM INITIALIZATION AND START UP.

```

1705 004316 113777 013000 175036      MOVB  ADDR,ADQRCSH      ;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              MOV   COUNT,R2       ;EIGHT SHIFTS FOR EIGHT BITS (NO VRC)
1710 004350 105077 175014              CLRB  ADQREG         ;SEL THE RX BA PRI.
1711 004354 012777 012154 175010      MOV   #RXBUFF,ADQSEC ;LOAD IT
1712 004362 105277 175002              INCB  ADQREG         ;SEL THE RX WC PRI.
1713 004366 012777 177777 174776      MOV   #-1,ADQSEC     ;SET FOR ONE CHAR.
1714 004374 112777 000010 174766      MOVB  #10,ADQREG     ;SELECT THE CHARACTER DET REGISTER.
1715 004402 013777 013004 174762      MOV   DETCAR,ADQSEC  ;SET THE CHARACTER TO BE DETECTED INTO DQ11 CHAR DET REG
1716 004410 112777 000014 174752      MOVB  #14,ADQREG     ;SELECT THE SEQUENCE REGISTER.
1717 004416 012777 120000 174746      MOV   #BIT15+BIT13,ADQSEC ;SET SINGLE CHAR DET AND SINGLE CHAR DET FLAG SE
1718 004424 112777 000012 174736      MOVB  #12,ADQREG     ;SELECT THE MISC REGISTER.
1719 004432 012777 004012 174732      MOV   #4012,ADQSEC  ;SET EIGHT BITS TEST LOOP AND AUTO STEP
1720 004440 052777 010001 174712      BIS   #BIT12+BIT0,ADQRCSR ;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      BIC   #BIT7,ADQSEC   ;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,ADQSEC     ;PLACE DATA ON BIT WINDOW.
1728 004506 005277 174660              INC   ADQSEC        ;CLOCK UP
1729 004512 005377 174654              DEC   ADQSEC        ;CLOCK DOWN
1730 004516 005302              DEC   R2            ;IS CHARACTER DONE YET?
1731 004520 001357              BNE   3$           ;BR IF NOT DONE
1732 004522 005777 174632              TST   ADQRCSR      ;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   ADQRCSR,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
1738                                ;IN ADDRESS $A ??
1739 004554 001401              BEQ   +4          ;
1740 004556 104002              HLT   2           ;WRONG ADDRESS.
1741 004560 000207              RTS   PC          ;
1742                                ;
1743                                ;TEST OF RECEIVER AND TRANSMITTER "SET "T" "
1744                                ;TEST OF BIT ONE OF SEQUENCE REGISTER.
1745                                ;THIS TEST WILL "SET T" AND THEN WILL
1746                                ;SEND A CHAR WHICH WILL "SET DONE CLEAR GO"; IF
1747                                ;REALLY IN TRANSPARENT MODE THE CHAR WILL NOT BE DETECTED
1748                                ;AND THE WORD COUNTS WILL GOTO ZERO.
1749                                ;
1750                                ; TEST 25
1751                                ;*****
1752 004562 012737 000025 001226      †ST25: MOV   #25,TSTNO
1753 004570 012737 004722 001216      MOV   #TST26,NEXT
1754                                ;
1755 004576 004737 011024 174562      2$: JSR   PC,SET.UP  ;SET UP ALL NECESSARY FOR TEST.
1756 004602 012777 040002              MOV   #BIT14+BIT1,ADQSEC ;SET DBL. CHAR AND SET T
1757                                ;SELECT ADD 16 (8)
1758 004610 105377 174546      DECB  ADQRCSH
1759 004614 112777 000010 174546      MOVB  #10,ADQREG    ;SELECT CHAR DET. ADDRESS
1760 004622 012777 164400 174542      MOV   #351*400,ADQSEC ;LOAD THE CHARACTER. SET DONE CLEAR GO.

```

1761	004630	112777	000014	174532	MOVB	#14,ADQREG	:SELECT THE SEQ REGISTER.
1762	004636	012777	100200	174526	MOV	#BIT15+BIT7,ADQSEC	:SET FUNCTION. CLEAR GO SET DONE.
1763							:PREPARE TX BUFFER
1764	004644	012700	011752		MOV	#TXBUFF,RO	:LOAD THE BUFFER WITH DATA
1765	004650	012720			MOV	(PC)+(RO)+	:DATA
1766	004652	000	350		.BYTE	000,350	:LOAD THE BUFFER WITH DATA
1767	004654	012720			MOV	(PC)+(RO)+	: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	MOVB	#1,ADQREG	:RX WC PRI. SHOULD =0
1771	004672	005777	174474		TST	ADQSEC	: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	MOVB	#3,ADQREG	:TX WC PRI SHOULD =0
1775	004710	005777	174456		TST	ADQSEC	: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							
1783							
1784							
1785							
1786							
1787							
1788							
1789							
1790							
1791							
1792							
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
1796	004742	012777	040002	174422		MOV	#BIT14+BIT1,ADQSEC
1797							:SET UP ALL NECESSARY FOR TEST.
1798	004750	105377	174406		DECB	ADQRCSH	:SET FUNCTION: DBL CHAR AND SET T
1799	004754	112777	000010	174406	MOVB	#10,ADQREG	:SELECT CHAR ADDRESS 16 (8)
1800	004762	012777	165000	174402	MOV	#352*400,ADQSEC	:SELECT CHAR DET ADDRESS.
1801	004770	112777	000014	174372	MOV	#14,ADQREG	:LOAD CHARACTER
1802	004776	012777	040004	174366	MOVB	#14,ADQREG	:SELECT THE SEQ REGISTER.
1803					MOV	#BIT14+BIT2,ADQSEC	:SET FOR DBL CHAR AND CLEAR T
1804	005004	105377	174352		DECB	ADQRCSH	:GET NEXT ADDR
1805							:GET NEXT ADDRESS
1806	005010	112777	000010	174352	MOVB	#10,ADQREG	:SELECT CHAR DET ADDRESS
1807	005016	012777	166400	174346	MOV	#355*400,ADQSEC	:LOAD CHARACTER.
1808	005024	112777	000014	174336	MOVB	#14,ADQREG	:SELECT THE SEQ REGISTER.
1809	005032	012777	120000	174332	MOV	#BIT15+BIT13,ADQSEC	:SET FOR SINGLE CHAR AND SET FLAG.
1810							:GET POINTER
1811	005040	012700	011752		MOV	#TXBUFF,RO	:LOAD THE BUFFER WITH DATA
1812	005044	012720			MOV	(PC)+(RO)+	:DATA
1813	005046	350	351		.BYTE	350,351	:LOAD THE BUFFER WITH DATA
1814	005050	012720			MOV	(PC)+(RO)+	: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	

```

:TEST OF RECEIVER "CLEAR "T" "
:TEST OF BIT TWO OF SEQUENCE REGISTER.
:THIS TEST WILL ENTER BOTH THE RECEIVER AND
:TRANSMITTER INTO TRANSPARENCY; THEN A CHARACTER
:WHICH SAYS "CLEAR RX T" WILL BE SENT FOLLOWED
:BY A CHARACTER WHICH SAYS "SET RX CHAR FLAG".
:THE TEST THEN CHECKS THAT THE CHARACTER FLAG IS SET
:WHICH MEANS THAT CHARACTER WAS DETECTED.

```

```

: TEST 26
:*****

```

```

1817 005060 005777 174274      TST      @DQRCR      ;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      ;TEST OF RECEIVER AND TRANSMITTER "BCC/CLEAR START"
1826      ;TEST OF BIT THREE OF SEQUENCE REGISTER.
1827      ;THE TEST STARTS UP THE TRANSMITTER AND RECEIVER BCC
1828      ;AND DEPOSITS ONE CHARACTER INTO IT. THE RECEIVER
1829      ;DONE FLAG COMES UP AND THE DQ11 CLOCK IS STOPPED.
1830      ;THE BCC'S OF BOTH THE TX AND RX ARE THEN "GRABBED"
1831      ;AND SHIFTED LOOKING FOR THAT ONE CHARACTER TO BE PRESENT
1832      ;IN THE BCC OF EACH RX AND TX BCC REGISTER.
1833
1834      ; TEST 27
1835      ;*****
1835 005072 012737 000027 001226  TST27:  MOV      #27,TSTNO
1836 005100 012737 005350 001216      MOV      #TST30,NEXT
1837 005106 004737 011024      JSR      PC,SETUP      ;SET UP ALL NECESSARY FOR TEST.
1838 005112 012777 100010 174252      MOV      #BIT15+BIT3,@DQSEC
1839
1840 005120 112777 000017 174242      MOV      #17,@DQREG      ;SET SNGL CHAR AND BCC/START CLEAR
1841 005126 012777 000200 174236      MOV      #200,@DQSEC      ;SEL THE POLY REG.
1842 005134 112777 000003 174226      MOV      #3,@DQREG      ;SET FOR LRC 8
1843 005142 012777 177576 174222      MOV      #-202,@DQSEC      ;SEL THE TX WC PRI.
1844 005150 012700 011752      MOV      #TXBUFF,RO      ;SET FOR 202 (8)
1845 005154 105020      1$:      CLR      (RO)+          ;GET TX BUFFER
1846 005156 022700 012153      CMP      #TXBUFF+201,RO  ;CLEAR ALL THE TX BUFFER
1847 005162 001374      BNE      1$          ;ALL CLEAR?
1848 005164 012700 011752      MOV      #TXBUFF,RO      ;BR IF NO
1849 005170 012720      MOV      (PC)+,(RO)+      ;GET TX BUFFER
1850 005172      350      225      .BYTE 350,225          ;LOAD THE BUFFER WITH DATA
1851 005174 012737 000225 012774      MOV      #225,CHAR      ;DATA
1852 005202 012737 000010 012776      MOV      #8,COUNT      ;SET EXPECTED BCC CHAR.
1853 005210 105077 174154      CLR      @DQREG      ;SET FOR 8 BITS
1854 005214 012777 012154 174150      MOV      #RXBUFF,@DQSEC  ;SEL REC PRIMARY
1855 005222 112777 000001 174140      MOV      #1,@DQREG      ;SET WITH START ADRS
1856 005230 012777 177775 174134      MOV      #-3,@DQSEC      ;SEL REC CHAR COUNT
1857 005236 112777 000004 174124      MOV      #4,@DQREG      ;SET CHAR COUNT
1858 005244 012777 012174 174120      MOV      #RXBUFF+20,@DQSEC ;SEL REC SECONDARY
1859 005252 112777 000005 174110      MOV      #5,@DQREG      ;SET WITH SEC ADRS
1860 005260 012777 177577 174104      MOV      #-201,@DQSEC    ;SEL REC CHAR COUNT
1861 005266 004737 011632      JSR      PC,NEWENA      ;SET CHAR COUNT
1862 005272 013703 012776      MOV      COUNT,R3      ;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
1863 005276 000241      6$:      CLC                    ;GET CHAR SIZE
1864 005300 106100      ROL      RO            ;SHIFT RX BCC IMAGE
1865 005302 005500      7$:      ADC      RO            ;PICK UP CARRY
1866 005304 023700 012774      CMP      CHAR,RO        ;CHECK BCC
1867 005310 001403      BEQ      3$            ;BCC OK!
1868 005312 005303      DEC      R3            ;ALL SHIFTS DONE?
1869 005314 001370      BNE      6$            ;BR IF NO.
1870 005316 104005      HLT      5             ;RX BCC HAS WRONG DATA.
1871 005320 013703 012776      3$:      MOV      COUNT,R3      ;SAVE COUNTER
1872 005324 000241      8$:      CLC

```

DZDQF MACY11 27(732) 24-SEP-76 10:17 PAGE 38  
DZDQFC.P11 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	5\$	:BR IF OK
1877	005340	005303		DEC	R3	:ALL SHIFTS DONE?
1878	005342	001370		BNE	8\$	:BR IF NO
1879	005344	104005		HLT	5	:TX BCC HAS WRONG DATA.
1880	005346	104400		5\$: SCOPE		:SCOPE THE TEST

```

1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
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,ADQSEC ;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$,LOCK ;SET FOR RETURN IF SW09=1
1$: JSR PC,EXT.UP ;SET THING UP FOR DOUBLE CHAR.(16 BITS)
MOV #BIT14+BIT6,ADQSEC ;SET DBL CHAR AND CLEAR ACTV
MOV #3721,XTXBUF+2 ;LOAD THE DATA
JSR PC,X.ABG ;WORK DQ11
;-----*LOCK*-----
; SCOP1 ; IF SW09=1; THEN GOTO ADDRESS IN "LOCK".
;-----
X.ABG: SCOPE ;SCOPE THIS TEST
INC ADQRCSR ;SET RX GO.
INC ADQTCSR ;SET TX GO
TSTB ADQTCSR ;HANG HERE FOR TX PRI DONE!!
BPL .-4 ;BR IF NOT DONE.
CMP (SP),(SP) ;WAIST TIME!
BIT #BIT12,ADQRCSR ;IS RX ACTIVE CLEARED?
BEQ .+4 ;BR IF YES
HLT 6 ;RX ACTIVE NOT CLEARED
MOVB #1,ADQREG ;GET THE RX WC PRI.
TST ADQSEC ;IT SHOULD BE NON-ZERO!!
BNE .+4 ;BR IF OK
HLT 6 ;RX PRI WC =0
CMPB #001,ADQRCSR ;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,ADQSEC
1941 ;SET SNGL CHAR AND SET DONE CLEAR GO
1942 005560 012700 011754 MOV #TXBUFF+2,RO ;SET TX BUFFER
1943 005564 012710 MOV (PC)+,(RO) ;LOAD WITH DATA
1944 005566 350 352 .BYTE 350,352 ;DATA
1945 005570 004737 005640 JSR PC,X.ABF ;WORK DQ11
1946 005574 012737 005602 001220 MOV #1$,LOCK ;SET FOR RETURN IF SW09=1
1947 005602 004737 011212 1$: JSR PC,EXT.UP ;SET THING UP FOR DOUBLE CHAR.(16 BITS)
1948 005606 012777 040200 173556 MOV #BIT14+BIT7,ADQSEC
1949 ;SET DBL CHAR AND SET DONE CLEAR GO
1950 005614 012737 003721 012364 MOV #3721,XTXBUF+2 ;LOAD DATA
1951 005622 012737 012525 012366 MOV #12525,XTXBUF+4 ;SAME
1952 005630 004737 005640 JSR PC,X.ABF ;TURN ON DQ11
1953 ;-----*LOCK*-----
1954 005634 104401 ;SCOPE1 ;IF SW09=1; THEN GOTO ADDRESS IN "LOCK".
1955 ;-----*
1956 005636 104400 ;SCOPE ;SCOPE THIS TEST.
1957
1958 005640 004737 011556 X.ABF: JSR PC,ENABLE ;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
1959 005644 112777 000001 173516 MOVB #1,ADQREG ;SEL RX WC PRI
1960 005652 005777 173514 TST ADQSEC ;IT SB NON-ZERO.
1961 005656 001001 BNE .+4 ;BR IF OK
1962 005660 104007 HLT ? ;RX PRI WC =0
1963 005662 112777 000003 173500 MOVB #3,ADQREG ;SEL THE TX WC PRI
1964 005670 005777 173476 TST ADQSEC ;IT SB NON-ZERO
1965 005674 001001 BNE .+4 ;BR IF OK.
1966 005676 104007 HLT ? ;TX WC PRI IS =0
1967 005700 122777 000200 173452 CMPB #200,ADQRCSR ;DONE=1; P/S=0; GO=0?
1968 005706 001401 BEQ .+4 ;BR IF OK.
1969 005710 104007 HLT ? ;RX CSR NOT =200 (PRI DONE)
1970 005712 122777 000200 173444 CMPB #200,ADQTCSR ;DONE=1; P/S=0; GO=0
1971 005720 001401 BEQ .+4 ;BR IF OK.
1972 005722 104007 HLT ? ;TX PRI DONE SET? (TX CSR=200)
1973 005724 000207 RTS PC
1974
1975 ;
1976 ;TEST OF RECEIVER "CHARACTER STRIP"
1977 ;TEST OF BIT EIGHT OF SEQUENCE REGISTER.
1978 ;THE CHARACTER THAT IS SENT AS "CHARACTER STRIP" IS
1979 ;LOOKED FOR IN THE RX BUFFER; IF IT IS NOT FOUND IT
1980 ;IS ASSUMED THAT THE CHARACTER WAS INDEED "STRIPPED".
1981 ;
1982 ; TEST 32
1983 ;*****
1984 005726 012737 000032 001226 TST32: MOV #32,TSTNO
1985 005734 012737 006104 001216 MOV #TST33,NEXT
1986 005742 004737 011024 JSR PC,SET.UP ;SET UP ALL NECESSARY FOR TEST.
1987 005746 012777 100400 173416 MOV #BIT15+BIT8,ADQSEC
1988 ;SET SNGL CHAR AND CHAR STRIP.
1989 005754 012700 011754 MOV #TXBUFF+2,RO ;SET POINTER
1990 005760 012710 MOV (PC)+,(RO) ;LOAD THE
1991 005762 350 321 .BYTE 350,321 ;DATA
1992 005764 004737 011556 JSR PC,ENABLE ;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.

```



```

1993 005770 012700 012154      MOV      #RXBUFF,RO      :GET THE RX BUFFER
1994 005774 012701 000010      MOV      #10,R1         :SET FOR 10(8) CHARS
1995 005000 122720 000350      1$: CMPB   #350,(RO)+    :WAS THE CHAR STRIPPED?
1996 006004 001001          SNE      .+4           :BR IF NOT FOUND YET.
1997 006006 104010      HLT     10             :CHARACTER NOT STRIPPED FROM CORE.
1998 006010 005301      DEC     R1            :ADJUST CHAR COUNTER
1999 006012 001372      SNE     1$           :ALL DONE?
2000 006014 012737 006022 001220      MOV     #2$,LOCK       :SET FOR RETURN IF SW09=1
2001 006022 004737 011212      2$: JSR    PC,EXT.UP     :SET THING UP FOR DOUBLE CHAR.(16 BITS)
2002 006026 012777 040400 173336      MOV     #BIT14+BIT8,3DQSEC :SET DBL CHAR AND CHAR STRIP.
2003 006034 012737 003721 012366      MOV     #3721,XTXBUF+4  :LOAD DATA
2004 006042 012737 005672 012370      MOV     #5672,XTXBUF+6  :SAME
2005 006050 004737 011556      JSR    PC,ENABLE       :SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
2006 006054 012700 012154      MOV     #RXBUFF,RO     :GET POINTER
2007 006060 012701 000010      MOV     #10,R1         :SET CHAR COUNT.
2008 006064 022720 003721      3$: CMP   #3721,(RO)+    :CHAR STRIPPED?
2009 006070 001001          SNE     .+4           :SO FAR SO GOOD
2010 006072 104010      HLT     10             :CHARACTER NOT STRIPPED FROM CORE
2011 006074 005301      DEC     R1            :ALL DONE
2012 006076 001372      BNE     3$           :BR IF NO.
-----*LOCK*-----
2014 006100 104401      SCOPE1                :IF SW09=1; THEN GOTO ADDRESS IN "LOCK".
-----
2016 006102 104400      SCOPE                  :

```

```

:TEST OF "TRANS PAD"
:TEST OF BIT TEN OF SEQUENCE REGISTER
:THE PAD CHARACTER IS SENT IN THE MIDDLE OF THE MESSAGE
:AND THE PROGRAM VERIFIES THAT THE PAD WASN'T INSERTED
:IN THE MIDDLE OF THE RX BUFFER.

```

: TEST 33

\*\*\*\*\*

```

2020 006104 012737 000033 001226      TST33: MOV     #33,TSTNO
2021 006112 012737 006260 001216      MOV     #TST34,NEXT
2022 006120 004737 011024      JSR    PC,SET.UP       :SET UP ALL NECESSARY FOR TEST.
2023 006124 012777 102000 173240      MOV     #BIT15+BIT10,3DQSEC :SET SNGL CHAR AND TX PAD
2024 006132 012700 011754      MOV     #TXBUFF+2,RO   :LOAD THE
2025 006136 012710          MOV     (PC)+(RO)      :DATA
2026 006140          350 101              .BYTE 350,101         :INTO THE TX BUFFER
2027 006142 004737 011556      JSR    PC,ENABLE       :SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
2028 006146 012700 012154      MOV     #RXBUFF,RO     :GET POINTER
2029 006152 012701 000010      MOV     #10,R1         :SET CHAR COUNT
2030 006156 122720 000377      1$: CMPB   #377,(RO)+    :PAD PRESENT?
2031 006162 001001          BNE     .+4           :BR IF NO
2032 006164 104011      HLT     11             :PAD CHARACTER IS IN BUFFER.
2033 006166 005301      DEC     R1            :ALL CHAR. DONE?
2034 006170 001372      BNE     1$           :BR IF NO.
2035 006172 012737 006172 001220      2$: MOV     #2$,LOCK       :SET FOR RETURN IF SW09=1
2036 006200 004737 011212      JSR    PC,EXT.UP     :SET THING UP FOR DOUB CHAR.(16 BITS)
2037 006204 012777 042000 173160      MOV     #BIT14+BIT10,3DQSEC :SET DBL CHAR AND TX P
2038

```

```

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      #RXBUFF,RO    ;GET POINTER
2053 006236 012701 000010                MOV      #10,R1        ;GET CHAR COUNTER
2054 006242 022720 177777                3$:     CMP      #177777,(RO)+ ;TX PAD IN BUFFER?
2055 006246 001001                SNE      .+4           ;BR IF NO
2056 006250 104011                HLT      11            ;PAD CHARACTER IS IN BUFFER.
2057 006252 005301                DEC      R1            ;ALL CHARS DONE?
2058 006254 001372                BNE      3$           ;BR IF NO.
2059 006256 104400                4$:     SCOPE

```

```

;TEST OF "BCC EXCLUDE"
;TEST OF PIT 11 OF SEQUENCE REGISTER
;"BCC EXCLL" IS EXERCIZED ON BOTH THE RX AND TX TOGETHER
;THE BCC IS TURNED ON AND THEN A CHARACTER IS EXCLUDED
;FROM THE BCC; WHEN DONE COMES UP THE BCC'S OF BOTH
;THE RX AND TX ARE SHIFTED AROUND TO SEE IF THE
;CHARACTER WAS REALLY EXCLUDED.

```

TEST 34

\*\*\*\*\*

```

2074 006260 012737 000034 001226  TST34:  MOV      #34,TSTNO
2075 006266 012737 006600 001216      MOV      #TST35,NEXT
2076 006274 004737 011024                JSR      PC,SETUP      ;SET UP ALL NECESSARY FOR TEST.
2077 006300 012777 100010 173064      MOV      #BIT15+BIT3,ADQSEC
2078                                ;SET SNGL CHAR AND BCC START CLEAR
2079 006306 105377 173050                DECB    ADQRC5H        ;SEL CHAR ADD 16(8)
2080 006312 112777 000010 173050      MOV      #10,ADQREG    ;GET CHAR DET ADDRESS
2081 006320 012777 170400 173044      MOV      #361*400,ADQSEC ;LOAD CHAR.
2082 006326 112777 000014 173034      MOV      #14,ADQREG    ;SEL SEQ REG
2083 006334 012777 104000 173030      MOV      #BIT15+BIT11,ADQSEC
2084                                ;SET SNGL CHAR AND BCC EXCLUDE
2085 006342 112777 000017 173020      MOV      #17,ADQREG    ;SEL POLY REG.
2086 006350 012777 000200 173014      MOV      #200,ADQSL    ;SET LRC 8
2087 006356 112777 000003 173004      MOV      #3,ADQREG     ;SEL TX WC PRI.
2088 006364 012777 177576 173000      MOV      #-202,ADQSEC  ;SET BIG NUMBER
2089 006372 012700 011752                MOV      #TXBUFF,RO   ;SET POINTER
2090 006376 105020                1$:     CLRB    (RO)+      ;LOAD DATA 000
2091 006400 022700 012153                CMP      #TXBUFF+201,RO ;CLEAR BUFFER!!
2092 006404 001374                BNE      1$           ;BR IF NOT ALL CLEAR.
2093 006406 012700 011752                MOV      #TXBUFF,RO   ;SET POINTER
2094 006412 012710                MOV      (PC)+,(RO)   ;LOAD THE
2095 006414                350 107                .BYTE 350,107        ;DATA
2096 006416 112737 000361 011754      MOV      #361-TXBUFF+2 ;
2097 006424 012737 000010 012776      MOV      #8,COUNT     ;
2098 006432 012737 000107 012774      MOV      #107,CHAR    ;SET FOR 8 BITS AND 107 AS THE CHAR IN BCC.
2099 006440 105077 172724                CLRB    ADQREG        ;SEL REC PRIMARY
2100 006444 012777 012154 172720      MOV      #RXBUFF,ADQSEC ;SET WITH START ADRS
2101 006452 112777 000001 172710      MOV      #1,ADQREG    ;SEL REC CHAR COUNT
2102 006460 012777 177775 172704      MOV      #-3,ADQSEC   ;SET CHAR COUNT
2103 006466 112777 000004 172674      MOV      #4,ADQREG    ;SET REC SECONDARY
2104 006474 012777 012174 172670      MOV      #RXBUFF+20,ADQSEC ;SET WITH SEC ADRS

```

```

2105 006502 112777 000005 172660      MOVB    #5,JDQREG      ;SEL CHAR COUNT
2106 006510 012777 177577 172654      MOV     #-201,JDQSEC  ;SET CHAR COUNT
2107 006516 004737 011632          JSR     PC,NEWENA     ;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
2108 006522 013703 012776          MOV     COUNT,R3     ;SAVE COUNTER
2109 006526 000241          4$:    CLC           ;
2110 006530 106100          ROLB   R0            ;SHIFT RX BCC IMAGE
2111 006532 005500          5$:    ADC     R0     ;SAVE CARRY
2112 006534 023700 012774          CMP     CHAR,R0     ;BCC OK?
2113 006540 001403          BEQ    3$           ;BR IF OK
2114 006542 005303          DEC    R3          ;ALL SHIFTS DONE?
2115 006544 001370          BNE   4$           ;BR IF NO
2116 006546 104012          HLT   12          ;RX BCC HAS WRONG DATA.
2117 006550 013703 012776          3$:    MOV     COUNT,R3
2118 006554 000241          6$:    CLC           ;CLEAR CARRY
2119 006556 106101          ROLB   R1            ;SHIFT TX BCC IMAGE
2120 006560 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

```

```

;
;TEST OF SET TRANSPARENCY FOR TRANSMITTER.
;TEST THAT THE SEQ FUNCTIONS ARE ALLOWED IF
;THEY ARE PRECEDED BY *DLE*.
;THIS TEST THAT WHEN THE TRANSMITTER FLIPS FROM PRI TO SEC
;AND "EXIT T" IS ASSERTED THAT THE TX SENDS A "DLE"
;CHARACTER.

```

```

; TEST 35
;*****

```

```

2138 006600 012737 000035 001226  TST35: MOV     #35,TSTNO
2139 006606 012737 007054 001216      MOV     #TST35,NEXT
2140 006614 004737 011024          JSR     PC,SET.UP    ;SET UP ALL NECESSARY FOR TEST.
2141 006620 012777 040002 172544      MOV     #BIT14+BIT1,JDQSEC
2142          ;SET DBL CHAR AND SET T
2143 006626 105377 172530          DECB   JDQRCSH     ;SEL CHAR ADD 16
2144 006632 112777 000010 172530      MOVB   #10,JDQREG  ;GET CHAR DET REG
2145 006640 013700 001372          MOV    DQSEC,R0
2146 006644 012710          MOV    (PC)+(R0)   ;LOAD THE REGISTER
2147 006646 352 352          .BYTE 352,352     ;WITH THIS DATA
2148 006650 112777 000014 172512      MOVB   #14,JDQREG  ;SEL THE SEQ REG.
2149 006656 012777 101000 172506      MOV    #BIT15+BIT9,JDQSEC
2150          ;SET SNGL CHAR AND DLE
2151 006664 105377 172472          DECB   JDQRCSH     ;SEL CHAR ADD 15(8)
2152 006670 112777 000010 172472      MOVB   #10,JDQREG  ;GET CHAR ADDRESS
2153 006676 012777 166000 172466      MOV    #354*400,JDQSEC ;LOAD WITH DATA
2154 006704 112777 000014 172456      MOVB   #14,JDQREG  ;SEL THE SEQ REGISTER
2155 006712 012777 100200 172452      MOV    #BIT15+BIT7,JDQSEC
2156          ;SET SNGL CHAR AND SET DONE CLEAR GO
2157 006720 112777 000003 172442      MOVB   #3,JDQREG   ;SEL THE TX WC PRI.
2158 006726 012777 177772 172436      MOV    #-6,JDQSEC  ;SET FOR 6 CHARS
2159 006734 112737 000350 011753      MOVB   #350,TXBUFF+1 ;LOAD
2160 006742 112737 000357 011756      MOVB   #357,TXBUFF+4 ; DATA

```

# E04

DZDQF MACY11 27(732) 24-SEP-76 10:17 PAGE 44  
 DZDQFC.P11 PROGRAM INITIALIZATION AND START UP.

```

2161 006750 112737 000354 012012      MOVB   #354, TXBUFF+40  ;EXIT T SHOULD SEND DLE TTIS SHOULD SET DONE THEN!!
2162 006756 112777 000006 172404      MOVB   #6,  ADQREG      ;SEL TX BA SEC.
2163 006764 012777 012012 172400      MOV    #TXBUFF+40, ADQSEC
2164 006772 105277 172372      INCB   ADQREG          ;LOAD BA AND SEL WC SEC.
2165 006776 052777 030000 172362      BIS    #BIT13+BIT12, ADQERR
2166                                     ;SET WRITE ENABLE AND EXIT T
2167 007004 012777 177770 172360      MOV    #-10, ADQSEC    ;SET SEC. FOR 10(8) CHARS
2168 007012 004737 011556                                     ;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
2169 007016 112777 000007 172344      MOVB   #7,  ADQREG      ;SEL THE TX WC SEC.
2170 007024 005777 172342      TST    ADQSEC          ;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      MOVB   #1,  ADQREG      ;SEL THE RX WC PRI.
2174 007042 005777 172324      TST    ADQSEC          ;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
2178
2179
2180
2181
2182
2183                                     ;TEST THAT THE TRANSMITTER WILL EXIT T
2184                                     ;WHEN ENTERED BY "SET T"
2185
2186
2187
2188                                     : TEST 36
2189                                     :*****
2190 ST36: MOV    #36, TSTNO
2191      MOV    #TST37, NEXT
2192      JSR    PC, SET.UP      ;SET UP ALL NECESSARY FOR TEST.
2193      MOV    #BIT14+BIT1, ADQSEC
2194                                     ;SET DBL CHAR AND SET T
2195      DECB   ADQRC5H        ;GET CHAR ADDR 16(8)
2196      MOVB   #10, ADQREG     ;GET CHAR REG.
2197      MOV    #354*400, ADQSEC ;LOAD CHARACTER
2198      MOVB   #14, ADQREG     ;SEL THE SEQ REG.
2199      MOV    #BIT15+BIT7, ADQSEC
2200                                     ;SET SNGL CHAR AND SET DONE CLEAR GO.
2201      MOVB   #6,  ADQREG     ;SEL TX BA SEC.
2202      MOV    #TXBUFF+40, ADQSEC
2203      INCB   ADQREG          ;SEL TX WC SEC.
2204      BIS    #BIT13+BIT12, ADQERR
2205                                     ;SET WRITE ENABLE AND EXIT T
2206      MOV    #-10, ADQSEC    ;SET FOR 10(8) CHARS
2207      MOVB   #350, TXBUFF+3  ;LOAD DATA
2208      MOVB   #354, TXBUFF+41 ;SAME
2209      INC    ADQTCSR        ;SET TX GO.
2210      BIT    #BIT6, ADQTCSR  ;HANG HERE FOR TX SEC. DONE.
2211      BEQ    1$             ;BR IF NOT DONE
2212      MOVB   #3,  ADQREG     ;GET TX WC PRI.
2213      TST    ADQSEC          ;IS IT =0
2214      BEQ    .+4             ;BR IF OK
2215      HLT    3               ;TX WC PRI NOT=0
2216      MOVB   #7,  ADQREG     ;SEL TX WC SEC
2217      TST    ADQSEC          ;IS NOT =0?
2218      BNE    .+4             ;BR IF OK
  
```

```

2217 007254 104003 HLT 3 ;TX SEC WC S/B NON-ZERO.
2218 007256 104400 SCOPE ;SCOPE THIS TEST.
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272

;TEST THAT THE RECEIVER STRIP SYNC IS
;INHIBITED WHEN IN TRANSPARENT MODE.
; TEST 37
*****
TST37: MOV #37,TSTNO
MOV #TST40,NEXT
JSR PC.SET.UP ;SET UP ALL NECESSARY FOR TEST.
MOV #BIT14+BIT1,ADQSEC
DECB ADQRC5H ;SET DBL CHAR AND SET T
MOVB #10,ADQREG ;SEL CHAR ADD 16(8)
MOV DQSEC,RO ;SEL CHAR REG
MOV (PC)+,(RO) ;LOAD THE CHAR
.BYTE 101,101 ;DATA
MOVB #14,ADQREG ;SEL THE SEQ REG
MOV #BIT15+BIT9,ADQSEC
MOVB #350, TXBUFF+1 ;SET SNGL CHAR AND DLE
MOV #TXBUFF+2,RO ;LOAD BUFFER
1$: MOVB #101,(RO)+ ;DLE
MOVB #26,(RO)+ ;SYNC
CMP #TXBUFF+1,RO ;KEEP STUFFING
BNE 1$
MOV #3,ADQREG ;SET STRIP SYNC AND GO(RX)
INC ADQREG ;SET TX GO
TSTB ADQRC5R ;HANG HERE FOR RX DONE (P)
BPL .-4
MOV #RXBUFF,RO ;GET RX POINTER
CMPB #0,(RO)+ ;FIRST CHAR S/B=0
BEQ .+4
HLT 3 ;FIRST DATA CHAR WRONG
CMPB #350,(RO)+ ;NEXT CHAR S/B=350
BEQ .+4
HLT 3 ;RX BUFFER WRONG
2$: CMPB #101,(RO)+ ;DLE PRESENT?
BEQ .+4
HLT 3 ;DLE NOT THERE
CMPB #26,(RO)+ ;SYNC PRESENT?
BEQ .+4
HLT 3 ;LOOKS LIKE SYNC STRIPPED?
CMP #RXBUFF+10,RO ;BUFFER DONE?
BNE 2$ ;BR IF NO
SCOPE ;SCOPE THIS TEST

;VERIFY THAT BIT 8 OF THE SEQUENCE
;REGISTER STRIPS CHARS FROM CORE BUT NOT
;FROM THE BCC.

```

```

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

```

```

2329      : TEST 41
2330      :*****
2331      010002 012737 000041 001226 TST41: MOV #41,TSTNO
2332      010010 012737 010060 001216      MOV #TST42,NEXT
2333      010016 012737 100060 011022      MOV #BIT15+BIT5+BIT4,FUNCT.
2334      010024 012737 000000 012772      MOV #000,XPOLY ;SET EXTENDED POLY.
2335      010032 012737 000255 012770      MOV #255,POLY ;SET 00-15 POLY
2336      010040 004737 011412      JSR PC,BCC.TA ;GOTO SUBROUTINE
2337      010044 017737 171316 012766      MOV @DQERR,ERR ;IS THE AN ERROR CONDITION?
2338      010052 100001      BPL +4 ;BR IF NO ERRORS
2339      010054 104015      HLT 15 ;THE DQ11 ERROR FLAG IS SET!!
2340      010056 104400      SCOPE ;SCOPE THIS TEST
2341
2342      ;TEST OF BCC TEST/ APPEND
2343      ;TEST OF 2 BCC'S TESTED/APPENDED
2344
2345      : TEST 42
2346      :*****
2347      010060 012737 000042 001226 TST42: MOV #42,TSTNO
2348      010066 012737 010136 001216      MOV #TST43,NEXT
2349      010074 012737 100020 011022      MOV #BIT15+BIT4,FUNCT.
2350      010102 012737 000000 012772      MOV #000,XPOLY ;SET EXTENDED POLY.
2351      010110 012737 112001 012770      MOV #112001,POLY ;SET 00-15 POLY
2352      010116 004737 011412      JSR PC,BCC.TA ;GOTO SUBROUTINE
2353      010122 017737 171240 012766      MOV @DQERR,ERR ;IS THE AN ERROR CONDITION?
2354      010130 100001      BPL +4 ;BR IF NO ERRORS
2355      010132 104015      HLT 15 ;THE DQ11 ERROR FLAG IS SET!!
2356      010134 104400      SCOPE ;SCOPE THIS TEST
2357
2358      ;TEST OF BCC TEST/ APPEND
2359      ;TEST OF 3 BCC'S TESTED/APPENDED
2360
2361
2362      : TEST 43
2363      :*****
2364      010136 012737 000043 001226 TST43: MOV #43,TSTNO
2365      010144 012737 010214 001216      MOV #TST44,NEXT
2366      010152 012737 100040 011022      MOV #BIT15+BIT5,FUNCT.
2367      010160 012737 000225 012772      MOV #225,XPOLY ;SET EXTENDED POLY.
2368      010166 012737 112001 012770      MOV #112001,POLY ;SET 00-15 POLY
2369      010174 004737 011412      JSR PC,BCC.TA ;GOTO SUBROUTINE
2370      010200 017737 171162 012766      MOV @DQERR,ERR ;IS THE AN ERROR CONDITION?
2371      010206 100001      BPL +4 ;BR IF NO ERRORS
2372      010210 104015      HLT 15 ;THE DQ11 ERROR FLAG IS SET!!
2372      010212 104400      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 DBL CHAR MATCH
: 13 SNGL CHAR FLAG
: 12 DBL 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, SET_UP ; SET UP ALL NECESSARY FOR TEST.
MOV #170412, @DQSEC ; LOAD THE SEQ FUNCTIONS
DECB @DQRC5H ; GET CHAR ADD 16(8)
MOVB #10, @DQREG ; GET CHAR REG.
MOV #225*400, @DQSEC ; LOAD CHAR.
MOVB #14, @DQREG ; SEL SEQ REG
MOV #120644, @DQSEC ; LOAD SEQ FUNCTIONS
DECB @DQRC5H
MOVB #10, @DQREG ; SEL CHAR DET
MOV #10020, @DQSEC ; LOAD FUNCTIONS
MOVB #14, @DQREG ; SEL SEQ REG
MOV #101000, @DQSEC ; LOAD DLE (20)
MOVB #6, @DQREG ; SEL TX SEC ADRS
MOV #TXBUFF+50, @DQSEC ; SET SEC ADRS
MOVB #1, @DQREG ; SEL REC PRI SHAR CNT
MOV #-200, @DQSEC ; SET CHAR CNT
MOVB #67, @DQREG ; SEL TXSEC CHAR CNT
MOV #-2, @DQSEC ; SET CHAR CNT
MOVB #17, @DQREG ; SEL POLY REG
MOV #172516, @DQSEC ; SET *WILD* POLYNOMIAL
MOVB #12, @DQREG ; SEL THE MISC REG
BIS #BIT6, @DQSEC ; SET THE "EXT POLY" BIT
MOVB #17, @DQREG ; WRITE POLY 16-23

```



2429	010434	012777	000275	170730	MOV	#275,ADQSEC	:
2430	010442	012700	011752		MOV	#TXBUFF,RO	:GET POINTER
2431	010446	012720			MOV	(PC)+,(RO)+	:LOAD THE BUFFER WITH DATA
2432	010450	000	350		.BYTE	000,350	:DATA
2433	010452	012720			MOV	(PC)+,(RO)+	:LOAD THE BUFFER WITH DATA
2434	010454	311	224		.BYTE	311,224	:DATA
2435	010456	012720			MOV	(PC)+,(RO)+	:LOAD THE BUFFER WITH DATA
2436	010460	107	201		.BYTE	107,201	:DATA
2437	010462	012720			MOV	(PC)+,(RO)+	:LOAD THE BUFFER WITH DATA
2438	010464	371	251		.BYTE	371,251	:DATA
2439	010466	012700	012022		MOV	#TXBUFF+50,RO	:LOAD RO
2440	010472	012720			MOV	(PC)+,(RO)+	:LOAD BUFFER WITH DATA
2441	010474	225	377		.BYTE	225,377	:DATA
2442	010476	004737	011556		JSR	PC,ENABLE	:SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
2443	010502	005777	170652		TST	ADQRCR	:DID RX CHAR FLAG SET?
2444	010506	100401			BMI	.+4	:BR IF YES
2445	010510	104014			HLT	14	:CHAR DET FLAG NOT SET.
2446	010512	017737	170650	012766	MOV	ADQERR,ERR	:ANY ERRORS?
2447	010520	100001			BPL	.+4	:BR IF NO ERRORS
2448	010522	104017			HLT	17	:DQ11 ERROR FLAG SET!
2449	010524	122777	000200	170626	CMPB	#200,ADQRCR	:DONE(P)=1; P/S=0; GO=0?
2450	010532	001401			BEQ	.+4	:YES
2451	010534	104014			HLT	14	:RX CSR WRONG DATA
2452	010536	122777	000300	170620	CMPB	#300,ADQTCR	:DONE(P)=1; P/S=0; GO=0?
2453	010544	001401			BEQ	.+4	:YES
2454	010546	104014			HLT	14	:TX CSR WRONG DATA
2455	010550	112777	000001	170612	MOVSB	#1,ADQREG	:GET RX WC PRI.
2456	010556	005777	170610		TST	ADQSEC	:S/B NOT=0
2457	010562	001001			BNE	.+4	:BR IF OK
2458	010564	104014			HLT	14	:RX WC PRI S/B NON-ZERO!!
2459	010566	112777	000003	170574	MOVSB	#3,ADQREG	:SEL THE TX WC PRI.
2460	010574	005777	170572		TST	ADQSEC	:S/B =0
2461	010600	001401			BEQ	.+4	
2462	010602	104014			HLT	14	:TX WC PRI S/B NON-ZERO
2463	010604	012700	012154		MOV	#RXBUFF,RO	:GET RX BUFFER POINTER
2464	010610	012702	000010		MOV	#10,R2	:SET CHAR COUNT
2465	010614	122710	000350		CMPB	#350,(RO)	:ARE THE TWO CHARS STRIPPED?
2466	010620	001001			BNE	.+4	
2467	010622	104014			HLT	14	:CHAR "350" NOT STRIPPED FROM CORE
2468	010624	122720	000225		CMPB	#225,(RO)+	
2469	010630	001001			BNE	.+4	
2470	010632	104014			HLT	14	:CHAR "225" NOT STRIPPED
2471	010634	122720	000020		CMPB	#20,(RO)+	:IS DLE STRIPPED?
2472	010640	001001			BNE	.+4	
2473	010642	104021			HLT	21	:DLE STUCK -- REFER TO M7817 ECO
2474							:CHECK E84 ONE-SHOT
2475	010644	005302			DEC	R2	:ALL DONE?
2476	010646	001362			BNE	1\$	:NO
2477	010650	104400			SCOPE		:SCOPE THE TEST

DZDQF MACY11 27(732) 24-SEP-76 10:17 PAGE 50  
 DZDQFC.P11 PROGRAM INITIALIZATION AND START UP.

2478											
2479											
2480	010652					.MEMCLR:					
2481	010652	105077	170512			CLRB	ADQREG				
2482	010656	012705	000020			MOV	#16, R5				
2483	010662	152777	000020	170500	1\$:	BISB	#BIT4, ADQREG				
2484	010670	142777	000140	170472		SICB	#140, ADQREG				
2485	010676	005077	170470			CLR	ADQSEC				
2486	010702	105277	170462			INCB	ADQREG				
2487	010706	005305				DEC	R5				
2488	010710	001364				BNE	1\$				
2489	010712	105077	170452			CLRB	ADQREG				
2490	010716	105077	170440			CLRB	ADQRCSH				
2491	010722	012705	000020			MOV	#16, R5				
2492	010726	112777	000010	170434	2\$:	MOVB	#10, ADQREG				
2493	010734	005077	170432			CLR	ADQSEC				
2494	010740	112777	000014	170422		MOVB	#14, ADQREG				
2495	010746	005077	170420			CLR	ADQSEC				
2496	010752	105277	170404			INCB	ADQRCSH				
2497	010756	005305				DEC	R5				
2498	010760	001362				BNE	2\$				
2499	010762	105077	170374			CLRB	ADQRCSH				
2500	010766					.MSTCLR:					
2501	010766	112777	000012	170374		MOVB	#MISC, ADQREG				
2502	010774	012777	000040	170370		MOV	#BIT5, ADQSEC				
2503	011002	000240				NOP					
2504	011004	112777	000012	170356		MOVB	#MISC, ADQREG				
2505	011012	012777	000040	170352		MOV	#BIT5, ADQSEC				
2506	011020	000002				RTI					
2507											
2508	011022	000000				FUNCT.: 0					
2509	011024					SET.UP:					
2510	011024	104413				MEMCLR					
2511	011026	012702	000010			MOV	#10, R2				; CLEAR ALL THE DQ11 REGISTERS
2512	011032	012700	011752			MOV	#TXBUFF, R0				; PREPARE TO CLEAR THE TX BUFFER
2513	011036	105020				CLRB	(R0)+				; GET THE BUFFERS ADDRESS
2514	011040	005302				DEC	R2				; START CLEARING
2515	011042	001375				BNE	-4				; ALL CLEAR?
2516	011044	105077	170320			CLRB	ADQREG				; BR IF NOT DONE.
2517	011050	012777	012154	170314		MOV	#RXBUFF, ADQSEC				; SELECT THE RX BA PRI.
2518	011056	105277	170306			INCB	ADQREG				; LOAD RX BA PRI.
2519	011062	012777	177770	170302		MOV	#-10, ADQSEC				; GET RX WC PRI.
2520	011070	105277	170274			INCB	ADQREG				; SET FOR 8. CHARS.
2521	011074	012777	011750	170270		MOV	#SYNC, ADQSEC				; GET THE TX BA PRI.
2522	011102	105277	170262			INCB	ADQREG				; SET ADDRESS
2523	011106	012777	177766	170256		MOV	#-12, ADQSEC				; SEL THE TX WC PRI.
2524	011114	112777	000011	170246		MOVB	#11, ADQREG				; SET FOR 8. CHARS AND 2 SYNC'S
2525	011122	013777	011746	170242		MOV	.SYNC, ADQSEC				; SEL THE SYNC REGISTER
2526	011130	105277	170234			INCB	ADQREG				; LOAD WITH SYNC
2527	011134	012777	004000	170230		MOV	#4000, ADQSEC				; SELECT THE MISC REGISTER
2528	011142	032737	040000	001510		BIT	#JUMBIT, DQSTAT				; SET FOR EIGHT BITS.
2529	011150	001003				BNE	+10				; IF TEST JUMPER AT END OF CABLE;
2530	011152	052777	000010	170212		BIS	#BIT3, ADQSEC				; RUN DATA THROUGH IT.
2531	011160	112777	000017	170174		MOVB	#17, ADQRCSH				; NO TEST JUMPER; SET TEST LOOP
2532	011166	112777	000010	170174		MOVB	#10, ADQREG				; GET LAST CHAR DET ADDRESS.
2533	011174	012777	164000	170170		MOV	#350*400, ADQSEC				; SEL CHAR DET REGISTER
											; LOAD THE CHARACTER

2534	011202	112777	000014	170160	MOV	#14,ADQREG	;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	#XTXBUF,RO	;GET THE TX BUFFER ADDRESS.
2540	011224	005020			CLR	(RO)+	;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	ADQREG	;SEL THE RX BA PRI.
2546	011252	012777	012154	170112	MOV	#RXBUFF,ADQSEC	;LOAD THE ADDRESS
2547	011260	105277	170104		INCB	ADQREG	;SEL THE RX WC PRI.
2548	011264	012777	177770	170100	MOV	#-10,ADQSEC	;SET FOR TEN CHARS
2549	011272	105277	170072		INCB	ADQREG	;SEL THE TX BA PRI.
2550	011276	012777	012356	170066	MOV	#XSYNC,ADQSEC	;LOAD THE ADDRESS.
2551	011304	105277	170060		INCB	ADQREG	;SEL THE TX WC PRI.
2552	011310	012777	177766	170054	MOV	#-12,ADQSEC	;SET FOR TWO SYNC AND 8. CHARS
2553	011316	112777	000011	170044	MOV	#11,ADQREG	;SEL THE SYNC REGISTER
2554	011324	013777	011746	170040	MOV	.SYNC,ADQSEC	;LOAD SYNC
2555	011332	105277	170032		INCB	ADQREG	;SEL THE MISC REGISTER.
2556	011336	005077	170030		CLR	ADQSEC	;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,ADQSEC	;NO JUMPER; SET TEST LOOP!
2560	011360	112777	000017	167774	MOV	#17,ADQRC SH	;GET LAST CHAR DET ADDRESS
2561	011366	112777	000010	167774	MOV	#10,ADQREG	;GET CHAR DET REGISTER
2562	011374	012777	003721	167770	MOV	#3721,ADQSEC	;LOAD CHARACTER
2563	011402	112777	000014	167760	MOV	#14,ADQREG	;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,ADQSEC	
2568	011416	012777	100010	167746			
2569							
2570	011424	105377	167732		DECB	ADQRC SH	;SET SNGL CHAR AND BCC START CLEAR
2571	011430	112777	000010	167732	MOV	#10,ADQREG	;GET NEXT ADDR
2572	011436	012777	176400	167726	MOV	#375*400,ADQSEC	;SEL CHAR DET ADDR
2573	011444	112777	000014	167716	MOV	#14,ADQREG	;LOAD CHAR
2574	011452	013777	011022	167712	MOV	FUNCT.,ADQSEC	;SEL THE SEQ REG.
2575	011460	112777	000017	167702	MOV	#17,ADQREG	;SET THE TEST APPEND FUNCTIONS
2576	011466	013777	012770	167676	MOV	POLY,ADQSEC	;SEL THE POLY REG.
2577	011474	112777	000012	167666	MOV	#12,ADQREG	;LOAD THE POLYNOMIAL.
2578	011502	052777	000100	167662	MOV	#16,ADQSEC	;SEL THE MISC REG
2579	011510	112777	000017	167652	BIS	#BIT6,ADQSEC	;SEL EXT POLY REG
2580	011516	013777	012772	167646	MOV	#17,ADQREG	;RESEL. THE POLY REG
2581	011524	012700	011752		MOV	XPOLY,ADQSEC	;SET 16-23 POLY
2582	011530	012720			MOV	#TXBUFF,RO	;SET TX POINTER
2583	011532	350	355		MOV	(PC)+,(RO)+	;LOAD THE BUFFER WITH DATA
2584	011534	012720			.BYTE	350,355	;DATA
2585	011536	360	365		MOV	(PC)+,(RO)+	;LOAD THE BUFFER WITH DATA
2586	011540	012720			.BYTE	360,365	;DATA
2587	011542	370	375		MOV	(PC)+,(RO)+	;LOAD THE BUFFER WITH DATA
2588	011544	012720			.BYTE	370,375	;DATA
2589	011546	377	377		MOV	(PC)+,(RO)+	;LOAD THE BUFFER WITH DATA
					.BYTE	377,377	;DATA

```

2590 011550 004737 011556 JSR 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 #5,4$ ;SET DELAY
2595 011570 005277 167564 INC @DQRCR ;SET RX GO.
2596 011574 005277 167564 INC @DQTCR ;SET TX GO.
2597 011600 105777 167554 1$: TSTB @DQRCR ;RX PRI. DONE?
2598 011604 100407 BMI 2$ ;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 1 ;RX PRI DONE NOT SET.
2604 011624 000207 2$: RTS PC ;LEAVE
2605 011626 000000 3$: 000
2606 011630 000000 4$: 000
2607
2608 011632 005037 011742 NEWENA: CLR 3$
2609 011636 012737 000005 011744 MOV #5,4$ ;SET DELAY
2610 011644 005277 167510 INC @DQRCR ;SET RX GO
2611 011650 005277 167510 INC @DQTCR ;SET TX GO
2612 011654 105777 167500 1$: TSTB @DQRCR ;RX PRIMARY DONE?
2613 011660 100410 BMI 2$ ;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 1 ;RX PRI DONE FAILED TO SET
2619 011700 000417 BR 5$ ;LEAVE
2620 011702 112777 000012 167460 2$: MOVB #12,@DQREG ;SEL THE MISC.REG
2621 011710 042777 000010 167454 BIC #BIT3,@DQSEC ;STOP DATA
2622 011716 112777 000015 167444 MOVB #15,@DQREG ;SEL RX BCC
2623 011724 017700 167442 MOV @DQSEC,R0 ;READ INTO R0
2624 011730 105277 167434 INCB @DQREG ;SEL TX BCC
2625 011734 017701 167432 MOV @DQSEC,R1 ;READ INTO R1
2626 011740 000207 5$: RTS PC ;RETURN
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 012154 .+.200
2635 012154 000000 RXBUFF: 0
2636 012356 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 012564 .+.200
2641 012564 000000 XRXBUF: 000000
2642 012766 012766 .+.200
2643 012766 000000 ERR: 0
2644 012770 000000 POLY: 0
2645 012772 000000 XPOLY: 0

```





```

2758 013422 112577 165564          MOVB   (R5)+, @TPDBR
2759 013426 001345          BNE    1$
2760 013430 012605          3$:   MOV   (SP)+, R5
2761 013432 000002          RTI

                                     ;ASCII STRING INPUT ROUTINE
2762
2763
2764
2765 013434 010346          .INSTR: MOV   R3, -(SP)
2766 013436 010446          MOV   R4, -(SP)
2767 013440 017637 000004 013456  MOV   34(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
2778 013512 001003          BNE   200$
2779 013514 104402 015176  TYPE, MCRLF
2780 013520 000755          BR    .INST1
2781 013522 122427 000015  200$: CMPB  (R4)+, #15
2782 013526 001423          BEQ   INSTR2
2783 013530 117777 165452 165454  MOVB  @TKDBR, @TPDBR
2784 013536 105777 165446  2$:   TSTB  @TPCSR
2785 013542 100375          BPL   2$
2786 013544 005303          DEC   R3
2787 013546 001350          BNE   1$
2788 013550 000402          BR    .INSTG
2789 013552 010346          .INSTE: MOV   R3, -(SP)
2790 013554 010446          MOV   R4, -(SP)
2791 013556 104402          .INSTG: TYPE
2792 013560 015172          MQM
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

                                     ;CONVERT ASCII STRING TO OCTAL
2800
2801
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

```

```

00014 013652 122714 000015          CMPB  #15,(R4)
00015 013656 001420          BEQ   PARERR
00016 013660 121427 000060          1$:  CMPB  (R4),#60
00017 013664 002415          BLT  PARERR
00018 013666 121427 000067          CMPB  (R4),#67
00019 013672 003012          BGT  PARERR
00020 013674 142714 000060          BICB  #60,(R4)
00021 013700 152405          BISB  (R4)+,R5
00022 013702 122714 000015          CMPB  #15,(R4)
00023 013706 001414          BEQ   LIMITS
00024 013710 006305          ASL  R5
00025 013712 006305          ASL  R5
00026 013714 006305          ASL  R5
00027 013716 000760          BR   1$
00028 013720 122714 000015          PARERR: CMPB  #15,(R4)          ;IS FIRST CHARACTER A <CR>
00029 013724 001003          BNE  120$
00030 013726 005737 015016          TST  3#RDSW          ;IS CKSWR ROUTINE BEING USED
00031 013732 001023          BNE  PARTI
00032 013734 104404          120$: INSTER
00033 013736 000742          BR   PARAM1
00034          ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
00035
00036
00037 013740 020537 014012          LIMITS: CMP  R5,HILIM
00038 013744 101365          BHI  PARERR
00039 013746 020537 014010          CMP  R5,LOLIM
00040 013752 103762          BLO  PARERR
00041 013754 133705 014016          BITB LOBITS,R5
00042 013760 001357          BNE  PARERR
00043          ;STORE NUMBER AT SPECIFIED ADDRESS
00044
00045
00046 013762 013704 014014          1$:  MOV  DEVADR,R4
00047 013766 010524          MOV  R5,(R4)+
00048 013770 062705 000002          ADD  #2,R5
00049 013774 105337 014017          DECB ADRCNT
00050 014000 001372          BNE  1$
00051 014002 012604          PARTI: MOV  (SP)+,R4
00052 014004 012605          MOV  (SP)+,R5
00053 014006 000002          RTI
00054 014010 000000          LOLIM: 0
00055 014012 000000          HILIM: 0
00056 014014 000000          DEVADR: 0
00057 014016 000000          LOBITS: 0
00058          ADRCNT=LOBITS+1
00059          ;SAVE PC OF TEST THAT FAILED AND R0-R5
00060
00061 014018 016637 000004 001274 .SAV05: MOV  4(SP),SAVPC
00062
00063          ;SAVE R0-R5
00064
00065
00066 014026 010537 001274          SV05: MOV  R5,SAVR5
00067 014032 010437 001266          MOV  R4,SAVR4
00068 014036 010337 001264          MOV  R3,SAVR3
00069 014042 010237 001262          MOV  R2,SAVR2

```



2870	014046	010137	001260		MOV	R1,SAVR1
2871	014052	010037	001256		MOV	RO,SAVR0
2872	014056	000002			RTI	
2873						
2874						:RESTORE RO-R5
2875						
2876	014077	013700	001256	.RESOS:	MOV	SAVR0,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	001264		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	017501	000012		MOV	#12(SP),R1
2894	014134	013737	015652	001250	MOV	TEMP,TEMP3
2895	014142	062766	000002	000012	ADD	#2,12(SP)
2896	014150	012137	014332		MOV	(R1)+,WRDCNT
2897	014154	112137	014334	1\$:	MOV	(R1)+,CHRCNT
2898	014160	112137	014335		MOV	(R1)+,SPACNT
2899	014164	013137	014336		MOV	2(R1)+,BINWRD
2900	014170	013704	014336	2\$:	MOV	BINWRD,R4
2901	014174	113705	014334		MOV	CHRCNT,R5
2902	014200	012700	015652		MOV	#TEMP,RO
2903	014204	010403		3\$:	MOV	R4,R3
2904	014206	042703	177770		BIC	#177770,R3
2905	014212	062703	000060		ADD	#060,R3
2906	014216	110320			MOV	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	3\$
2915	014240	012703	015714		MOV	#MDATA,R3
2916	014244	114023		4\$:	MOV	-(R0),(R3)+
2917	014246	105337	014334		DECB	CHRCNT
2918	014252	001374			BNE	4\$
2919	014254	105737	014335		TSTB	SPACNT
2920	014260	001405			BEQ	6\$
2921	014262	112723	000040	5\$:	MOV	#040,(R3)+
2922	014266	105337	014335		DECB	SPACNT
2923	014272	001373			BNE	5\$
2924	014274	105013		6\$:	CLRB	(R3)
2925	014276	104402			TYPE	

2926	014300	015714				MDATA	
2927	014302	005337	014332			DEC	WRDCNT
2928	014306	001322				BNE	1\$
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		01				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	1\$
2966	014436	011637	001234			MOV	(SP), LSTERR
2967	014442	105037	001312			CLRB	ERRFLG
2968	014446	104406				1\$: 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)+, ERAMSG
2978	014502	012437	014602			MOV	(R4)+, DATAHD
2979	014506	011437	014614			MOV	(R4), DATABP
2980	014512	105737	001312			TSTB	ERRFLG
2981	014516	001403				BEQ	TYPMSG

```

2982 014520 005737 014614 TST DATABP
2983 014524 001027 BNE TYPDAT
2984 014526 104402 TYPMSG: TYPE
2985 014530 015455 MTSTN
2986 014532 104411 CNVRT
2987 014534 014714 XTSTN
2988 014536 104402 TYPE
2989 014540 015543 MERRPC
2990 014542 104411 CNVRT
2991 014544 014706 ERTABO
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 000000 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: RESO5
3009 014620 005777 164354 HALTS: TST @SWR
3010 014624 100005 BPL EXITER
3011 014626 010046 PUSHRO
3012 014630 016600 000002 MOV 2(SP),RO
3013 014634 000000 HALT
3014 014636 012600 POPRO
3015 014640 104414 EXITER: CKSWR
3016 014642 005237 001232 INC ERRCNT
3017 014646 032777 000400 164324 BIT #SWO8,@SWR
3018 014654 001007 BNE 1$
3019 014656 032777 002000 164314 BIT #SW10,@SWR
3020 014664 001407 BEQ 2$
3021 014666 013737 001216 001214 MOV NEXT,RETURN
3022 014674 012706 001200 1$: MOV #STACK,SP
3023 014700 000177 164310 JMP @RETURN
3024 014704 000002 2$: RTI
3025 014706 000001 ERTABO: 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 ;SET UP FOR POWER UP TRAP
3036 014730 000000 HALT ;HALT ON POWER DOWN NORMAL
3037 014732 000777 BR

```

# H05

DZDQF MACY11 27(732) 24-SEP-76 10:17 PAGE 60  
 DZDQFC.P11 GENERAL UTILITIES (TYPE OUT, ERROR, SCOPE, ETC.)

```

3038
3039                                     ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
3040
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     @#42
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 015060 122737 000007 013456          CMPB   #7,.MSG            ;IS IT A ↑G
3076 015066 001021                        BNE     OUT
3077 015070 104402 015146                TYPE, $CNTG
3078 015074 005137 015016                .CNTLU: COM     @#RDSW
3079 015100 104402 015152                TYPE, $MSWR
3080 015104 104411 015140                CNVRT, SWREGC
3081 015110 104403 015161                INSTR, $MNEW
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                $CNTG: .ASCIZ <377>/↑G/

```

DZDQF MACY11 27(732) 24-SEP-76 10:17 PAGE 61  
 DZDQFC.P11 GENERAL UTILITIES (TYPE OUT, ERROR, SCOPE, ETC.)

3094	015152	051777	051127	020075	\$MSWR: .ASCIZ <377>/SWR= /
3095	015160	000			
3096	015161	040	047040	053505	\$MNEW: .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	047111	
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	015425	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	

3150	01557.	7523	000		
3151		15576		.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					
3160					
3161	015610	000000			
3162		015652		INBUF: 0	
3163	015652	000000		.=.+40	
3164		015714		TEMP: 0	
3165	015714	000000		.=.+40	
3166		015756		MDATA: 0	
3167	015756	000000		.ERRTA: 0	
3168	015760	000000		0	;HALT 0
3169	015762	000000		0	
3170	015764	016132		EMO	
3171	015766	000000		0	;HALT 1
3172	015770	000000		0	
3173	015772	016252		EM1	
3174	015774	016521		DMO	;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	

```

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



















DZDQF MPCY 27(732) 24-SEP-76 10:17 PAGE 73  
 DZDQFC.P1. CROSS REFERENCE TABLE -- USER SYMBOLS

.MSG	013456	2761*	2770#	3073*	3074*	3075
.MSTCL	010766	1126	2500#			
.PARAM	013604	1116	2803#			
.PTAIL	014722	910	1203	3034#	3042	
.RESOVL	014060	1120	2876#			
.SIVOSUS	014020	1118	2862#			
.SCOPEP	013174	1106	2709#			
.SCOPI	013306	1108	2733#			
.START	001512	961	1201#	113		
.SYNC	011746	2525	2554	2631#		
.TRPSR	014340	914	2945#			
.TRPTA	001314	1104#	2950			
.TYPE	013326	1110	2741#			





ADC	1865	1874	2111	2120	2310										
ADD	1006	1247	1319	1333	1335	1340	1342	1344	1348	1350	2743	2768	2848	2895	2905
ASL	2950	2973	2976												
BCC	2824	2825	2826	2948	2972	2974									
BEG	937														
	993	988	992	995	1011	1226	1244	1250	1282	1290	1316	1420	1479	1739	1772
	1776	1867	1876	1915	1922	1968	1971	2113	2122	2209	2212	2252	2255	2258	2261
	2312	2450	2453	2461	2684	2711	2735	2782	2794	2815	2823	2920	2958	2965	2981
	2996	3001	3005	3020											
BGT	2819														
BHI	2838														
BIC	947	1338	1405	1408	1418	1463	1477	1723	1726	1736	2621	2904	2949	2975	3074
BICB	2484	2776	2820												
BIS	946	978	984	989	993	996	997	998	1402	1409	1459	1468	1720	1727	2165
	2202	2427	2530	2559	2578										
BISB	2483	2621													
BIT	991	994	1249	1281	1289	1313	1379	1914	2208	2528	2557	2710	2717	2734	2746
	2957	2962	3017	3019											
BITB	2841														
BLO	2840														
BLOS	933	1257													
BLT	2817														
BMI	1239	1415	1474	1733	1818	2444	2598	2613							
BNE	942	957	970	1008	1018	1216	1234	1236	1272	1279	1309	1314	1321	1380	1413
	1426	1472	1485	1731	1847	1869	1878	1919	1961	1965	1996	1999	2010	2013	2041
	2044	2055	2058	2092	2115	2124	2171	2175	2216	2245	2264	2289	2314	2318	2321
	2457	2466	2469	2472	2476	2488	2498	2515	2529	2542	2558	2600	2602	2615	2617
	2681	2718	2721	2745	2747	2759	2778	2787	2829	2831	2842	2850	2914	2918	2923
	2928	2963	2983	3018	3046	3068	3070	3076							
BPL	1912	2249	2338	2354	2370	2447	2714	2749	2751	2754	2757	2774	2785	2960	3010
	3072														
BR	945	1030	1227	1248	1261	1286	1299	1325	1329	2619	2712	2716	2780	2788	2796
	2827	2833	3037												
CCC	1317	1330													
CLC	1014	1464	1863	1872	2109	2118	2308	2907	2909	2911					
CLR	938	965	968	973	979	1000	1001	1002	1003	1004	1009	1028	1206	1208	1210
	1211	1253	1269	1385	1386	1443	1444	1445	1465	1706	2485	2493	2495	2540	2556
	2593	2608	2658	2659	2723	2812	3044	3051	3052	3088					
CLRB	1205	1207	1384	1389	1390	1394	1448	1452	1710	1845	1853	2090	2099	2287	2298
	2481	2489	2490	2499	2513	2516	2545	2722	2924	2967					
CMP	931	932	948	966	975	1007	1225	1228	1235	1256	1270	1271	1846	1866	1875
	1913	2009	2054	2091	2112	2121	2244	2263	2288	2311	2320	2720	2837	2839	2964
	3069														
CMPB	1737	1921	1967	1970	1995	2040	2251	2254	2257	2260	2317	2449	2452	2465	2468
	2471	2777	2781	2814	2816	2818	2822	2828	3075						
COM	1462	3078													
COMB	1218	1312	1404	1722											
DEC	1017	1320	1411	1412	1470	1471	1729	1730	1868	1877	1998	2012	2043	2057	2114
	2123	2313	2475	2487	2497	2514	2541	2601	2616	2680	2786	2913	2927	2927	2927
DECB	1758	1798	1804	2079	2143	2151	2193	2232	2280	2408	2413	2570	2849	2917	2922
EMT	596														
HALT	652	654	656	658	660	662	664	666	668	670	672	674	676	678	680
	682	684	686	688	690	692	694	696	698	700	702	704	706	708	710
	712	714	716	718	720	722	724	726	728	730	732	734	736	738	740
	742	744	746	748	750	752	754	756	758	760	762	764	766	768	770
	772	774	776	778	780	782	784	786	788	790	792	794	796	798	800



ROLB	1864	1873	2110	2119	2309										
ROR	936	1406	1466	1724	2908	2910	2912								
RORB	1467														
RTI	950	1032	2506	2727	2737	2761	2799	2853	2872	2882	2935	3024	3089		
RTS	951	1741	1924	1973	2535	2564	2591	2604	2626						
SUB	2946	2970													
TRAP	1105	1107	1109	1111	1113	1115	1117	1119	1121	1123	1125	1127	1129	1131	
TST	939	944	982	987	999	1010	1233	1278	1315	1414	1419	1473	1478	1732	1771
	1775	1817	1918	1960	1964	2170	2174	2211	2215	2443	2456	2460	2744	2793	2830
	2982	2995	3000	3004	3009	3067									
TSTB	1215	1238	1308	1911	2248	2597	2612	2713	2748	2750	2753	2756	2773	2784	2919
	2959	2980	3071												
.ASCIZ	1036	3093	3094	3096	3099	3100	3101	3106	3109	3110	3117	3121	3123	3127	3129
	3131	3133	3135	3138	3146	3147	3221								
.BLKW	1085	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169
	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184
	1185	1186	1187	1188	1189	1190	1191	1192							
.BYTE	1092	1093	1094	1095	1297	1298	1766	1768	1813	1815	1850	1944	1991	2036	2095
	2147	2236	2432	2434	2436	2438	2441	2583	2585	2587	2589	2631	2632	2637	2638
	2695	2698	2701	2704	3026	3029	3057	3086	3091	3153	3155	3222	3224	3227	3229
	3232														
.ENABL	532	549													
.END	3234														
.ENDC	1308	1363	1378	1442	1504	1517	1530	1543	1556	1569	1582	1595	1608	1621	1634
	1647	1660	1673	1686	1699	1754	1795	1837	1894	1939	1986	2031	2076	2140	2190
	2229	2277	2333	2349	2365	2406									
.EQUIV	596														
.EVEN	3098	3151	3157	3221											
.IF	1307	1362	1377	1441	1503	1516	1529	1542	1555	1568	1581	1594	1607	1620	1633
	1646	1659	1672	1685	1698	1753	1794	1836	1893	1938	1985	2030	2075	2139	2189
	2228	2276	2332	2348	2364	2405									
.IFF	1307	1308	1362	1363	1377	1378	1441	1442	1503	1504	1516	1517	1529	1530	1542
	1543	1555	1556	1568	1569	1581	1582	1594	1595	1607	1608	1620	1621	1633	1634
	1646	1647	1659	1660	1672	1673	1685	1686	1698	1699	1753	1754	1794	1795	1836
	1837	1893	1894	1938	1939	1985	1986	2030	2031	2075	2076	2139	2140	2189	2190
	2228	2229	2276	2277	2332	2333	2348	2349	2364	2365	2405				
.IIF	50	62	190	1306	1307	1308	1362	1363	1377	1378	1441	1442	1503	1504	1516
	1517	1529	1530	1542	1543	1555	1556	1568	1569	1581	1582	1594	1595	1607	1608
	1620	1621	1633	1634	1646	1647	1659	1660	1672	1673	1685	1686	1698	1699	1753
	1754	1794	1795	1836	1837	1893	1894	1938	1939	1985	1986	2030	2031	2075	2076
	2139	2140	2189	2190	2228	2229	2276	2277	2332	2333	2348	2349	2364	2365	2405
	2406														
.IRP	1105	1107	1109	1111	1113	1115	1117	1119	1121	1123	1125	1127	1129	1131	1156
	1303	1307	1359	1362	1374	1377	1438	1441	1495	1500	1503	1513	1516	1526	1529
	1539	1542	1552	1555	1565	1568	1578	1581	1591	1594	1604	1607	1617	1620	1630
	1633	1643	1646	1656	1659	1669	1672	1682	1685	1695	1698	1750	1753	1791	1794
	1833	1836	1890	1893	1935	1938	1982	1985	2027	2030	2072	2075	2136	2139	2186
	2189	2225	2228	2273	2276	2329	2332	2345	2348	2361	2364	2402	2405	2652	
.LIST	1	532	549	568	649	907	1034	1107	1109	1111	1113	1115	1117	1119	1121
	1123	1125	1127	1129	1131	1133	1193	1308	1363	1379	1443	1504	1517	1530	1543
	1556	1569	1582	1595	1608	1621	1634	1647	1660	1673	1686	1699	1754	1795	1837
	1894	1939	1986	2031	2076	2140	2190	2229	2277	2333	2349	2365	2373	2406	2652
	2707	3221													
.MACRO	1	1357													
.NLIST	1	532	549	568	649	907	1034	1107	1109	1111	1113	1115	1117	1119	1121
	1123	1125	1127	1129	1131	1133	1193	1308	1363	1379	1443	1504	1517	1530	1543

DZDQF MACY11 27(732) 24-SEP-76 10:17 PAGE 80  
 DZDQFC.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

	1556	1569	1582	1595	1608	1621	1634	1647	1660	1673	1686	1699	1754	1795	1837
	1894	1939	1986	2031	2076	2140	2190	2229	2277	2333	2349	2365	2373	2406	2652
	2707	3221													
.PAGE	1	43	532	568	649	907	1089								
.REM	1	43	50	62	105	190	2373								
.REPT	651														
.SBTTL	532	568	649	907	1034	1193	2652	2707							
.TITLE	549														
.WORD	919	2627	2628	3064											

ERRORS DETECTED: 0  
 DEFAULT GLOBALS GENERATED: 0

\*.DZDQFC.SEG/SOL/CRF/PAGNUM/NL:TOC+UNIV.P11,DZDQFC.P11  
 RUN-TIME: 21 32 5 SECONDS  
 RUN-TIME RATIO: 180/60=2.9  
 CORE USED: 20K (39 PAGES)

