

DQ11

SEQ REG TST  
CZDQFFO

AH-8628F-MC  
FICHE 1 OF 1

AUG 1981  
COPYRIGHT © 75-81  
MADE IN USA



IDENTIFICATION  
-----

PRODUCT CODE: AC-8626F-MC  
PRODUCT NAME: CZDQFF0 SEQ REG TSTS  
DATE: NOV 1980  
MAINTAINER: DIAGNOSTIC GROUP

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

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

COPYRIGHT (C) 1975, 1980 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

1. ABSTRACT

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

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

THE SEVEN DIAGNOSTICS ARE:

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

THERE IS ALSO AN ONLINE TEST TO BE DISCUSSED LATER.  
1. CZDQO [REV] ONLINE TEST. (ITEP OVERLAY)

AND A PARAMETER INPUT PROGRAM IS AVAILABLE  
1. CZDQG [REV] DQ11 TRIAL PROGRAM (PARAMETER INPUT)  
REQUIREMENTS

2.

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

CZDQF MACY11 30(1046) 05-MAY-81 14:18 PAGE 4  
 CZDQFF.P11 05-MAY-81 09:27

SEQ 0003

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  
EXAMPL: 1-1 DQ11; 3-2 DQ11; 7-3 DQ11; 17=4 DQ11 37=5 DQ11 ETC.  
E: NUMBER (IF VALID) WILL BE IN DATA LIGHTS (EXCLUDING 11/05, 11/04, 11/34)  
F: CONTINUE WITH ANY OTHER SWITCH SETTINGS DESIRED.

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

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

## 4.1.3 SWITCH REGISTER PRIORITYS

## ERROR SWITCHES

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

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

## SCOPE SWITCHES

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

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

#### 4.2 STARTING ADDRESS

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

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

#### 5. OPERATING PROCEDURE

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

AND PROGRAM WILL BEGIN RUNNING THE  
DIAGNOSTIC

#### 5.2 PROGRAM AND/OR OPERATOR ACTION

THE TYPICAL APPROACH SHOULD BE

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

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

#### 6. ERRORS

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

#### 6.2 ERROR RECOVERY

IF FOR SOME REASON THE DQ11 SHOULD  
"HANG THE BUS" (GAIN CONTROL OF BUS SO THAT  
CONSOLE MANUAL FUNCTIONS ARE INHIBITED) AN INIT  
OR POWER DOWN/UP IS NECESSARY FOR OPERATOR  
TO REGAIN CONTROL OF CPU.  
IF THIS SHOULD HAPPEN; LOOK IN LOCATION  
'TSTNO' (ADDRESS 1226) FOR THE NUMBER OF THE TEST THAT  
WAS RUNNING AT THE TIME OF THE CATASTROPHIC  
ERROR.  
IN THIS WAY THE OPERATOR WILL HAVE AN IDEA AS TO  
WHAT THE DQ11 WAS DOING AT THE TIME OF THE ERROR.

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

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

## 7. RESTRICTIONS

### 7.1 STARTING RESTRICTIONS

SEE SECTION 4. (PLEASE)

### 7.2 OPERATING RESTRICTIONS

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

## 8. MISCELLANEOUS

### 8.1 EXECUTION TIME

### 8.2 PASS COMPLETE

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

END PASS AC-8626F-MC CSR: 160000 VEC: 300 PASSES: 000001 ERRORS: 000000

NOTE: THE NUMBERS FOR CSR AND VEC ARE  
NOT NECESSARILY THE VALUES FOR THE DEVICE



THEY ARE ONLY FOR THIS EXAMPLE.

8.3 TST1 (MINI MONITOR)

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

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

8.4 KEY LOCATIONS

RETURN (1214) CONTAINS THE ADDRESS WHERE PROGRAM WILL  
RETURN WHEN ITERATION COUNT IS REACHED  
OR IF LOOP ON TEST IS ASSERTED.  
NEXT (1216) CONTAINS THE ADDRESS OF THE NEXT TEST  
TO BE PERFORMED.  
TSTNO (1226) CONTAINS THE NUMBER OF THE TEST NOW  
BEING PERFORMED.  
RUN (1304) THE BIT IN 'RUN' ALWAYS POINTS ONE  
PAST THE DQ11 CURRENTLY BEING TESTED.  
EXAMPLE:  
(RUN) 1304/0000000001000000  
MEANS THAT DQ11 NO.05 IS THE DQ11 NOW  
RUNNING.

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

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

DQACTV (1500) EACH BIT SET IN THIS LOCATION INDICATES  
THAT THE ASSOCIATED DQ11 WILL BE TESTED  
IN TURN.

EXAMPLE:  
(DQACTV) 1500/0000000000011111  
MEANS THAT DQ11 NO. 00,01,02,03,04  
WILL BE TESTED.

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

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

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

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

## BIT 00-08 VECTOR 'A' OF DEVICE

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

## 8.5.1 FINDING THE CONTROL STATUS REGISTER.

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

## 8.5.2 ONE SYNC BIT OR TWO?

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

## 8.5.3 'BB' OPTION INSTALLED?

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

## 8.5.4 'AB' OPTION INSTALLED?

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

## 8.5.5 'BA' OPTION INSTALLED?

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

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

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

## 8.5.7 ACTIVE ON FIRST NON-SYNC?

SINCE TOO MUCH HARDWARE MUST BE TURNED ON TO SENSE FOR WHEN THE DQ11 GUES 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 APPROPRIATE DQSTXX: LOCATION. AFTER AUTO SIZING

## 8.5.9- FINDING THE VECTOR.

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

## 9. PROGRAM DESCRIPTION

CONTAINED WITHIN LISTING

## 10. LISTING

FOLLOWING

522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
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

:CZDQFF0/<377>/SEQ REG TSTS  
:COPYRIGHT 1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS 01754  
:REVISED 16-DEC-76 BY R. BLACK  
: A)SUPPORTS SOFTWARE SWITCH REGISTER  
: B)SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER  
: BY <^G>.  
: STARTING PROCEDURE  
: LOAD PROGRAM  
: LOAD ADDRESS 000200  
: PRESS START  
: PROGRAM WILL TYPE 'CZDQFF0/<377>/SEQ REG TSTS'  
: 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	SW15=100000	: =1,HALT ON ERROR
040000	SW14=40000	: -1,LOOP ON CURRENT TEST
020000	SW13=20000	: =1,INHIBIT ERROR TYPEOUT
010000	SW12=10000	: =1,DELETE TYPEOUT/BELL ON ERROR.
004000	SW11=4000	: =1,INHIBIT ITERATIONS
002000	SW10=2000	: -1,ESCAPE TO NEXT TEST ON ERROR
001000	SW09=1000	: 1,LOOP WITH CURRENT DATA
000400	SW08=400	: =1,LOOP ON ERROR
000100	SW06=100	
000040	SW05=40	
000020	SW04=20	
000010	SW03=10	
000004	SW02=4	: LOCK ON TEST SELECT
000002	SW01=2	: RESTART PROGRAM AT SELECTED TEST
000001	SW00=1	: RESELECT DQ11 DESIRED ACTIVE
		: NOTE: THIS MUST NOT EXCEED ORIGINAL COUNT

```

558
559
560           ;REGISTER DEFINITIONS
561
562           000000      R0=%0           ;GENERAL REGISTER
563           000001      R1=%1           ;GENERAL REGISTER
564           000002      R2=%2           ;GENERAL REGISTER
565           000003      R3=%3           ;GENERAL REGISTER
566           000004      R4=%4           ;GENERAL REGISTER
567           000005      R5=%5           ;GENERAL REGISTER
568           000006      SP=%6          ;PROCESSOR STACK POINTER
569           000007      PC=%7          ;PROGRAM COUNTER
570
571           ;LOCATION EQUIVALENCIES
572
573           177570      DSWR- 177570    ;HARDWARE SWITCH REGISTER LOC.
574           177570      DLIGHTS=177570 ;HARDWARE DISPLAY REGISTER LOC.
575           177776      PS=177776     ;PROCESSOR STATUS WORD
576           001200      STACK=1200    ;START OF PROCESSOR STACK
577
578           ;INSTRUCTION DEFINITIONS
579
580           005746      PUSH1SP=5746   ;DECREMENT PROCESSOR STACK 1 WORD
581           005726      POP1SP=5726    ;INCREMENT PROCESSOR STACK 1 WORD
582           010046      PUSHRO=10046   ;SAVE R0 ON STACK
583           012600      POPRO=12600    ;RESTORE R0 FROM STACK
584           024646      PUSH2SP=24646  ;DECREMENT STACK TWICE
585           022626      POP2SP=22626   ;INCREMENT STACK TWICE
586           .EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL
587
588
589           100000      BIT15=100000
590           040000      BIT14=40000
591           020000      BIT13=20000
592           010000      BIT12=10000
593           004000      BIT11=4000
594           002000      BIT10=2000
595           001000      BIT9=1000
596           000400      BIT8=400
597           000200      BIT7=200
598           000100      BIT6=100
599           000040      BIT5=40
600           000020      BIT4=20
601           000010      BIT3=10
602           000004      BIT2=4
603           000002      BIT1=2
604           000001      BIT0=1
605
606
607           ;DQ11 OPTIONAL DEFINITIONS
608
609           002000      ABBIT=2000
610           004000      ACTBIT=4000
611           010000      BABIT=10000
612           020000      BBBIT=20000
613           040000      JUMBIT=40000

```

614 001000 ODDBIT=1000  
615 100000 SYMBIT=100000

:DQ11 SECONDARY REGISTER DEFINATIONS

619			
620	000000	RXBA.P=0	:RECEIVER BUS ADDRESS PRIMARY.
621	000001	RXWC.P=1	:RECEIVER WORD COUNT PRIMARY.
622	000002	TXBA.P=2	:TRANSMITTER BUS ADDRESS PRIMARY.
623	000003	TXWC.P=3	:TRANSMITTER BUS ADDRESS PRIMARY.
624	000004	RXBA.S=4	:RECEIVER BUS ADDRESS SECONDARY.
625	000005	RXWC.S=5	:RECEIVER WORD COUNT SECONDARY.
626	000006	TXBA.S=6	:TRANSMITTER BUS ADDRESS SECONDARY.
627	000007	TXWC.S=7	:TRANSMITTER WORD COUNT SECONDARY.
628			
629	000010	CHARDT=10	:CHARACTER DETECT REGISTER.
630	000011	SYNC.=11	:SYNC REGISTER.
631	000012	MISC.=12	:MISCELLANEOUS REGISTER.
632	000013	TX.MUX=13	:TRANSMITTER MUX REGISTER.
633	000014	SEQ.=14	:SEQUENCE REGISTER.
634	000015	RX.BCC=15	:RECEIVER BCC REGISTER.
635	000016	TX.BCC=16	:TRANSMITTER BCC REGISTER.
636	000017	POLY.=17	:POLYNOMIAL REGISTER.
637			
638			

```

639          ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
640          000000      .-0
641          ;STANDARD INTERRUPT VECTORS
642
643          .-24
644 000024 014764      .PFAIL          ;POWER FAIL HANDLER
645 000026 000340      340          ;SERVICE AT LEVEL 7
646 000030 014434      .HLT          ;ERROR HANDLER
647 000032 000340      340          ;SERVICE AT LEVEL 7
648 000034 014402      .TRPSRV       ;GENERAL HANDLER DISPATCH SERVICE
649 000036 000340      340          ;SERVICE AT LEVEL 7
650
651 000046 013162      .-46      LOGICAL          ;ACT HOOKS
652          .-52
653 000052 000000      .WORD 0
654          ;THIS ROUTINE TRIES TO FORCE THE RECEIVER TO INTERRUPT
655          ;TO ITS VECTOR WHERE IT WILL PICK UP THE STATUS LOCATION
656          ;FOR ITS NEW PC; AND PICK UP AN IOT INSTRUCTION FOR ITS
657          ;NEW PS. WHEN THE NEW PC IS FETCHED AN IOT INSTRUCTION IS
658          ;EXECUTED, TRAPPING TO LOCATION 20 WHERE A ROUTINE IS EXECUTED
659          ;TO TAKE THE PC FROM THE STACK AND USE IT AS THE VECTOR ADDRESS
660          000056      .-56
661
662 000056      VECMAP:
663 000056 010120      1$:  MOV      R1,(R0)+          ;START FILLING THE VECTOR AREA
664 000060 012721 000004      MOV      #4,(R1)+          ;WITH +2; IOT (4)
665 000064 022021      CMP      (R0)+,(R1)+          ;UPDATE THE POINTERS
666 000066 020127 001000      CMP      R1,#1000          ;IS ALL FLOATING VECTOR AREA DONE
667 000072 101771      BLOS     1$          ;BR IF NOT ALL DONE
668 000074 012737 000146 000020      MOV      #4$,@#20          ;SET FOR IOT TRAP BY DQ11
669 000102 013737 001500 001244      MOV      DQACTV,TEMP1      ;GET THE ACTIVE DQ11 S
670 000110 006037 001244      2$:  ROR      TEMP1          ;ARE YOU ACTIVE.. DQ11
671 000114 103023      BCC     5$          ;IF CARRY CLEAR.. NO MORE DQ11S
672 000116 005037 177776      CLR     PS          ;CLEAR PS
673 000122 005722      TST     (R2)+          ;PUT POINTER TO STATUS TABLE
674 000124 012772 000340 177776      MOV      #340,@-2(R2)      ;TRY AND SET PRI/SEC DONE AND IE
675 000132 105200      INCB   R0          ;DELAY.....
676 000134 001376      BNE     -2          ;.....DELAY
677 000136 112712 000300      MOVB   #300,(R2)          ;NO INTERRUPT ASSUME 300 FIX IN TEST C
678 000142 005722      3$:  TST     (R2)+          ;UPDATE POINTERS
679 000144 000761      BR     2$          ;GO DO IT AGAIN
680 000146 051612      4$:  BIS     (SP),(R2)          ;ENTERD BY IOT TRAP BY DQ11
681 000150 042712 000007      BIC     #7,(R2)          ;CLEAR UNWANTED BITS
682 000154 022626      CMP     (SP)+,(SP)+          ;POP IOT JUNK OFF STACK
683 000156 012716 000142      MOV     #3$,(SP)          ;SET RETURN PC ON STACK
684 000162 000002      RTI          ;GO HOME.
685 000164 000207      5$:  RTS     PC          ;ALL SIZING IS DONE
686
687          ;****SOFTWARE SWITCH REGISTER****
688          .-174
689 000174 000000      DISPREG: 0          ;SOFTWARE DISPLAY REGISTER
690 000176 000000      SWREG:   0          ;SOFTWARE SWITCH REGISTER
691
692          ;PROGRAM START
693
694          .-200

```

```
695 000200 000137 001512          JMP      .START          ;GO TO START OF PROGRAM
696
697          000220          . 220
698 000220 012702 001400      (SRMAP: MOV      #1400,R2          ;CLEAR ALL STATUS TABLE
699 000224 005022          CLR      (R2)+          ;DO CLEAR
700 000226 022702 001512          CMP      #1512,R2          ;ALL TABLE DONE
701 000232 001374          BNE     .-6              ;BR IF MORE TO GO
702 000234 005037 001504          CLR      DQNUM          ;SET NUMBER OF DQ11S TO 0
703 000240 012702 001400          MOV      #1400,R2          ;SET TABLE POINTER
704 000244 012701 160000          MOV      #160000,R1         ;GET FIRST FLOATING ADDRESS
705 000250 012737 000614 000004          MOV      #5$,@#4          ;SET FOR TIME OUT TRAP--NO DEVICE--
706 000256 112761 000012 000005 1$: MOVVB   #12,5(R1)          ;TRY AND SEL MISC REGISTER
707 000264 005061 000006          CLR      6(R1)           ;TRY AND CLEAR MISC REG
708 000270 012711 010000          MOV      #10000,(R1)       ;TRY AND SET RX ACTIVE
709 000274 022761 030000 000006          CMP      #30000,6(R1)      ;LOOK FOR SYNC 1 AND SYNC 2
710 000302 001071          BNE     2$              ;THIS IS NOT A DQ11 IF I BRANCH
711 000304 010122          MOV      R1,(R2)+         ;NOW THIS IS A DQ11 --STORE CSR
712 000306 052712 100000          BIS      #SYNBIT,(R2)      ;SET FOR TWO SYNC CHARS
713 000312 005011          CLR      (R1)            ;CLEAR DQ ACTIVE BIT
714 000314 112761 000010 000005          MOVVB   #10,5(R1)          ;SEL CHAR DET REGISTER
715 000322 012761 177777 000006          MOV      #-1,6(R1)         ;WRITE INTO CHAR DET REG
716 000330 005761 000006          TST     6(R1)            ;WAS THE REGISTER WRITTEN?
717 000334 001402          BEQ     .+6              ;APPARENTLY NO BB OPTION.
718 000336 052712 020000          BIS      #BBBIT,(R2)       ;SET FOR BB OPTION
719 000342 112761 000017 000005          MOVVB   #17,5(R1)          ;SEL POLYNO. REGISTER
720 000350 012761 177777 000006          MOV      #-1,6(R1)         ;WRITE POLYNO.REGISTER
721 000356 005761 000006          TST     6(R1)            ;WAS REG WRITTEN??
722 000362 001402          BEQ     .+6              ;BR IF NO AB OPTION
723 000364 052712 002000          BIS      #ABBIT,(R2)       ;SET FOR AB OPTION
724 000370 012761 001400 000002          MOV      #1400,2(R1)       ;TRY TO SET .DTR. .RS.
725 000376 032761 001400 000002          BIT      #1400,2(R1)       ;DID ANY OF THEM SET
726 000404 001402          BEQ     .+6              ;BR IF NO BA OPTION
727 000406 052712 010000          BIS      #BABIT,(R2)       ;SET FOR BA OPTION
728 000412 032761 030000 000002          BIT      #30000,2(R1)      ;DID .CS. .CO. SET
729 000420 001402          BEQ     .+6              ;BR IF NO JUMPER
730 000422 052712 040000          BIS      #JUMBIT,(R2)      ;SET FOR JUMPER
731 000426 052712 040000          BIS      #ACTBIT,(R2)      ;SET FOR ACTIVE ON FIRST NON-SYNC
732 000432 052712 001000          BIS      #ODDBIT,(R2)     ;SET FOR ODD VRC.....
733 000436 005722          TST     (R2)+            ;POP POINTER
734 000440 005011          CLR      (R1)            ;CLEAR RCSR
735 000442 005061 000002          CLR      2(R1)           ;CLEAR TCSR
736 000446 005061 000002          CLR      2(R1)           ;CLEAR AGAIN
737 000452 005061 000004          CLR      4(R1)           ;CLEAR ERROR REG
738 000456 005061 000006          CLR      6(R1)           ;CLEAR SEC REG
739 000462 005237 001504          INC      DQNUM          ;UPDATE NUMBER OF DQ11S
740 000466 062701 000010 2$: ADD     #10,R1          ;UPDATE CSR POINTER BY 10 (8)
741 000472 022701 164000          CMP      #164000,R1        ;HAVE ALL FLOATING ADDRESSES BEEN CHECKED??
742 000476 001267          BNE     1$              ;BR IF NOT ALL DONE
743 000500 005037 001500          CLR      DQACTV          ;ZERO ACTIVE DQ11S
744 000504 005737 001504          TST     DQNUM          ;WERE ANY DQ11S FOUND
745 000510 001434          BEQ     4$              ;HEY BUDDY. NO DQ11S FOUND IN SYSTEM
746 000512 013701 001504          MOV      DQNUM,R1        ;SAVE NUMBER OF DQ11S
747 000516 010137 001276          MOV      R1,SAVNUM        ;SAVE NUMBER FOR ACT11
748 000522 000241 3$: CLC          ;CLEAR CARRY
749 000524 006137 001500          ROL     DQACTV          ;ACTIVE ADDRESS
750 000530 005237 001500          INC     DQACTV          ;SET BIT 0
```



```

751 000534 005301          DEC      R1          ;DEC NUMBER OF DQ11S
752 000536 001371          BNE      3$          ;BR IF MORE TO GO
753 000540 012737 000006 000004  MOV      #6,@#4      ;RESET TIME OUT VECTOR
754 000546 013737 001500 001502  MOV      DQACTV,SAVACT ;SAVE ACTIVE
755 000554 012737 000340 000022  MOV      #340,@#22   ;SET IOT TRAP PRIO: TO 7
756 000562 012702 001400          MOV      #1400,R2    ;SET TABLE POINTER
757 000566 012700 000300          MOV      #300,R0     ;SET VECTOR START
758 000572 012701 000302          MOV      #302,R1     ;SET VECTOR+2 START
759 000576 000137 000056          JMP      VECMAP      ;GO FIND THE VECTORS
760 000602 104402          4$:      TYPE          ;TYPE MESSAGE
761 000604 015325          MERR2          ;I DIDN'T FIND ANY DQ11S. DON'T USE AUTO SIZE.
762 000606 005000          CLR      R0          ;
763 000610 000000          HALT          ;HOW CAN I TEST NO DQ11S
764 000612 000776          BR          ;DON'T LET OPR HIT CONT. SW
765 000614 012716 000466 5$:      MOV      #-2,(SP)    ;ENTERED BY TIME OUT TRAP
766 000620 000002          RTI          ;GO HOME.
767
768
769          001000          .=1000
770 C01000 005377 055103 050504 MTITLE: .ASCIZ <377><12>/CZDQFF0/<377>/SEQ REG TSTS/<377>
771 001006 043106 177460 042523
772 001014 020121 042522 020107
773 001022 051524 051524 000377
774
775          001200          .=1200
776          ;INDIRECT POINTERS
777
778 001200 177570          SWR:      177570      ;SWITCH REGISTER POINTER
779 001202 177570          LIGHTS: 177570      ;DISPLAY REGISTER POINTER
780 001204 177560          TKCSR:   177560      ;TELETYPE KEYBOARD CONTROL REGISTER
781 001206 177562          TKDBR:   177562      ;TELETYPE KEYBOARD DATA BUFFER
782 001210 177564          TPCSR:   177564      ;TELEPRINTER CONTROL REGISTER
783 001212 177566          TPDBR:   177566      ;TELEPRINTER DATA BUFFER
784
785          ;PROGRAM CONTROL PARAMETERS
786
787 001214 000000          RETURN: 0          ;SCOPE ADDRESS FOR LOOP ON TEST
788 001216 000000          NEXT:   0          ;ADDRESS OF NEXT TEST TO BE EXECUTED
789 001220 000000          LOCK:   0          ;ADDRESS FOR LOCK ON CURRENT DATA
790 001222 000003          ICOUNT: 3          ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
791 001224 000000          LPCNT: 0          ;NUMBER OF ITERATIONS COMPLETED
792 001226 000000          TSTNO:  0          ;NUMBER OF TEST IN PROGRESS
793 001230 000000          PASCNT: 0          ;NUMBER OF PASSES COMPLETED
794 001232 000000          ERRCNT: 0          ;TOTAL NUMBER OF ERRORS
795 001234 000000          LSTERR: 0          ;PC OF LAST ERROR CALL
796
797          ;PROGRAM VARIABLES
798
799 001236 000000          CHAR1:  0
800 001240 000000          CHAR2:  0
801 001242 000000          CHAR3:  0
802 001244 000000          TEMP1:  0          ;TEMPORARY STORAGE
803 001246 000000          TEMP2:  0          ;TEMPORARY STORAGE
804 001250 000000          TEMP3:  0          ;TEMPORARY STORAGE
805 001252 000000          TEMP4:  0          ;TEMPORARY STORAGE
806 001254 000000          TEMP5:  0          ;TEMPORARY STORAGE

```

CZDQF MACY11 30(1046) 05-MAY-81 14:18 PAGE 18  
CZDQFF.P11 05-MAY-81 09:27

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

SEQ 0017

807	001256	000000	SAVR0:	0	;R0 STORAGE
808	001260	000000	SAVR1:	0	;R1 STORAGE
809	001262	000000	SAVR2:	0	;R2 STORAGE
810	001264	000000	SAVR3:	0	;R3 STORAGE
811	001266	000000	SAVR4:	0	;R4 STORAGE
812	001270	000000	SAVR5:	0	;R5 STORAGE
813	001272	000000	SAVSP:	0	;STACK POINTER STORAGE
814	001274	000000	SAVPC:	0	;PROGRAM COUNTER STORAGE
815	001276	000000	SAVNUM:	0	
816	001300	000001	CREAM:	.BLKW 1	
817	001302	000000	RUNFLG:	0	
818	001304	000000	RUN:	0	
819	001306	000000	RUNCNT:	0	

820  
821  
822  
823 001310 000  
824 001311 000  
825 001312 000  
826 001313 000  
827 000000

:PROGRAM CONTROL FLAGS

INIFLG: .BYTE 0 ;PROGRAM INITIALIZATION FLAG  
STFLG: .BYTE 0 ;TEST START FLAG  
ERRFLG: .BYTE 0 ;ERROR OCCURED FLAG  
LOKFLG: .BYTE 0 ;LOCK ON CURRENT TEST FLAG  
SY-0

828  
829  
830  
831

:DEFINITIONS FOR TRAP SUBROUTINE CALLS  
:POINTERS TO SUBROUTINES CAN BE FOUND  
:IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS

832  
833  
834  
835 001314  
836 104400  
837 001314 013236  
838 104401  
839 001316 013350  
840 104402  
841 001320 013370  
842 104403  
843 001322 013476  
844 104404  
845 001324 013614  
846 104405  
847 001326 013646  
848 104406  
849 001330 014062  
850 104407  
851 001332 014122  
852 104410  
853 001334 014154  
854 104411  
855 001336 014160  
856 104412  
857 001340 010766  
858 104413  
859 001342 010652  
860 104414  
861 001344 015062  
862 104415  
863 001346 015136

:\*\*\*\*\*  
:\*\*\*\*\*  
:TRPTAB:  
SCOPE-TRAP+0 ;CALL TO SCOPE LOOP AND ITERATION HANDLER  
.SCOPE  
SCOPI=TRAP+1 ;CALL TO LOOP ON CURRENT DATA HANDLER  
.SCOPI  
TYPE-TRAP+2 ;CALL TO TELETYPE OUTPUT ROUTINE  
.TYPE  
INSTR=TRAP+3 ;CALL TO ASCII STRING INPUT ROUTINE  
.INSTR  
INSTER=TRAP+4 ;CALL TO INPUT ERROR HANDLER  
.INSTER  
PARAM=TRAP+5 ;CALL TO NUMERICAL DATA INPUT ROUTINE  
.PARAM  
SAV05=TRAP+6 ;CALL TO REGISTER SAVE ROUTINE  
.SAV05  
RES05=TRAP+7 ;CALL TO REGISTER RESTORE ROUTINE  
.RES05  
CONVRT=TRAP+10 ;CALL TO DATA OUTPUT ROUTINE  
.CONVRT  
CNVRT=TRAP+11 ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.  
.CNVRT  
MSTCLR=TRAP+12 ;CALL TO ISSUE MASTER CLEAR  
.MSTCLR  
MEMCLR=TRAP+13 ;CALL TO CLEAR ALL SCRATCH PAD MEMORIES  
.MEMCLR  
CKSWR=TRAP+14 ;CALL TO ALLOW SWREG TO BE LOADED FROM TTY  
.CKSWR  
CNTLU=TRAP+15 ;CALL TO ALLOW LOADING OF SWREG FROM TTY  
.CNTLU

864  
865  
866  
867

:\*\*\*\*\*  
:\*\*\*\*\*

:DQ1\* VECTOR AND REGISTER INDIRECT POINTERS

868  
869  
870 001350 000000  
871 001352 000000  
872 001354 000000  
873 001356 000000  
874 001360 000000  
875 001362 000000

DQ1VEC: 0 ;POINTER TO DQ11 RECEIVER INTERRUPT VECTOR  
DQ1LVL: 0 ;POINTER TO DQ11 RECEIVER INTERRUPT SERVICE PS  
DQ1VEC: 0 ;POINTER TO DQ11 TRANSMITTER INTERRUPT VECTOR  
DQ1LVL: 0 ;POINTER TO DQ11 TRANSMITTER INTERRUPT SERVICE PS  
DQ1CSR: 0 ;POINTER TO DQ11 RECEIVER CONTROL REGISTER  
DQ1CSH: 0 ;POINTER TO HIGH BYTE OF DQ11 RECEIVER CONTROL REGISTER

CZDQF MACY11 30(1046) 05-MAY-81 14:18 PAGE 20  
 CZDQFF.P11 05-MAY-81 09:27

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

SEQ 0019

876	001364	000000	DQCSR: 0	: POINTER TO DQ11 TRANSMITTER CONTROL REGISTER
877	001366	000000	DQERR: 0	: POINTER TO DQ11 ERROR REGISTER
878	001370	000000	DQREG: 0	: POINTER TO HIGH BYTE OF ERROR REGISTER
879	001372	000000	DQSEC: 0	: POINTER TO DQ11 SECONDARY REGISTER
880	001374	000000	DQSECH: 0	: POINTER TO HIGH BYTE OF DQ11 SECONDARY REGISTER

881  
882  
883  
884  
885

:DQ11 STATUS TABLE AND ADDRESS ASSIGNMENTS

886		001400	.-1400		
887	001400	000001	DQCR00: .BLKW	1	: CONTROL STATUS REGISTER FOR DEVICE NO: 00
888	001402	000001	DQST00: .BLKW	1	: VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 00
889	001404	000001	DQCR01: .BLKW	1	: CONTROL STATUS REGISTER FOR DEVICE NO: 01
890	001406	000001	DQST01: .BLKW	1	: VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 01
891	001410	000001	DQCR02: .BLKW	1	: CONTROL STATUS REGISTER FOR DEVICE NO: 02
892	001412	000001	DQST02: .BLKW	1	: VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 02
893	001414	000001	DQCR03: .BLKW	1	: CONTROL STATUS REGISTER FOR DEVICE NO: 03
894	001416	000001	DQST03: .BLKW	1	: VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 03
895	001420	000001	DQCR04: .BLKW	1	: CONTROL STATUS REGISTER FOR DEVICE NO: 04
896	001422	000001	DQST04: .BLKW	1	: VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 04
897	001424	000001	DQCR05: .BLKW	1	: CONTROL STATUS REGISTER FOR DEVICE NO: 05
898	001426	000001	DQST05: .BLKW	1	: VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 05
899	001430	000001	DQCR06: .BLKW	1	: CONTROL STATUS REGISTER FOR DEVICE NO: 06
900	001432	000001	DQST06: .BLKW	1	: VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 06
901	001434	000001	DQCR07: .BLKW	1	: CONTROL STATUS REGISTER FOR DEVICE NO: 07
902	001436	000001	DQST07: .BLKW	1	: VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 07
903	001440	000001	DQCR10: .BLKW	1	: CONTROL STATUS REGISTER FOR DEVICE NO: 10
904	001442	000001	DQST10: .BLKW	1	: VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 10
905	001444	000001	DQCR11: .BLKW	1	: CONTROL STATUS REGISTER FOR DEVICE NO: 11
906	001446	000001	DQST11: .BLKW	1	: VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 11
907	001450	000001	DQCR12: .BLKW	1	: CONTROL STATUS REGISTER FOR DEVICE NO: 12
908	001452	000001	DQST12: .BLKW	1	: VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 12
909	001454	000001	DQCR13: .BLKW	1	: CONTROL STATUS REGISTER FOR DEVICE NO: 13
910	001456	000001	DQST13: .BLKW	1	: VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 13
911	001460	000001	DQCR14: .BLKW	1	: CONTROL STATUS REGISTER FOR DEVICE NO: 14
912	001462	000001	DQST14: .BLKW	1	: VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 14
913	001464	000001	DQCR15: .BLKW	1	: CONTROL STATUS REGISTER FOR DEVICE NO: 15
914	001466	000001	DQST15: .BLKW	1	: VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 15
915	001470	000001	DQCR16: .BLKW	1	: CONTROL STATUS REGISTER FOR DEVICE NO: 16
916	001472	000001	DQST16: .BLKW	1	: VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 16
917	001474	000001	DQCR17: .BLKW	1	: CONTROL STATUS REGISTER FOR DEVICE NO: 17
918	001476	000001	DQST17: .BLKW	1	: VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 17
919	001500	000001	DQACTV: .BLKW	1	: HOLD ACTIVE BITS FOR TESTING
920	001502	000001	SAVACT: .BLKW	1	: SAVE NUMBER OF ACTIVE DQ11S
921	001504	000001	DQNUM: .BLKW	1	: OCTAL NUMBER OF TOTAL NUMBER OF DQ11S
922	001506	000001	DQCSR: .BLKW	1	: CSR OF DQ11 UNDER TEST
923	001510	000001	DQSTAT: .BLKW	1	: VECTOR AND CONFIGURATION STATUS OF DQ11 UNDER TEST

924  
925  
926  
927  
928  
929  
930  
931

:PROGRAM INITIALIZATION  
 :LOCK OUT INTERRUPTS  
 :SET UP PROCESSOR STACK  
 :SET UP POWER FAIL VECTOR  
 :CLEAR PROGRAM CONTROL FLAGS AND COUNTS  
 :TYPE TITLE MESSAGE

```

932 001512 012737 000340 177776 .START: MOV #340,PS ;LOCK OUT INTERRUPTS
933 001520 012706 001200 MOV #STACK,SP ;SET UP STACK
934 001524 012737 014764 000024 MOV #.PFAIL,@#24 ;SET UP POWER FAIL VECTOR
935 001532 013737 001504 001276 MOV DQNUM,SAVNUM
936 001540 105037 001311 CLR STFLG ;CLEAR START FLAG
937 001544 005037 001230 CLR PASCNT ;CLEAR PASS COUNT
938 001550 105037 001312 CLR ERRFLG ;CLEAR ERROR FLAG
939 001554 005037 001302 CLR RUNFLG
940 001560 012737 001400 001300 MOV #1400,CREAM
941 001566 005037 001232 CLR ERRCNT ;CLEAR ERROR COUNT
942 001572 005037 001234 CLR LSTERR ;CLEAR LAST ERROR POINTER
943 001576 012737 000001 001226 MOV #1,TSTNO ;SET UP FOR TEST 1
944 001604 012737 001512 001214 MOV #.START,RETURN ;SET UP FOR POWER FAIL BEFORE
945 ;TESTING STARTS
946 001612 012737 177570 001200 MOV #DSWR,SWR ;MOV HARDWARE SWR TO SWR
947 001620 012737 177570 001202 MOV #DLIGHTS,LIGHTS ;MOV DISPLAY LIGHTS TO LIGHTS
948 001626 013746 000006 MOV @#6,-(SP) ;SAVE VECTORS
949 001632 013746 000004 MOV @#4,-(SP)
950 001636 012737 001656 000004 MOV #64$,@#4 ;SET UP FOR TIMEOUT
951 001644 022777 177777 177326 CMP #-1,@SWR ;REFERENCE HARDWARE SWITCH REGISTER
952 001652 001402 BEQ 65$
953 001654 000407 BR 66$
954 001656 022626 64$: CMP (SP)+,(SP)+ ;ADJUST STACK
955 001660 012737 000176 001200 65$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
956 001666 012737 000174 001202 MOV #DISPREG,LIGHTS ;POINT TO SOFT DISPLAY REG
957 001674 012637 000004 66$: MOV (SP)+,@#4 ;RESTORE VECTORS
958 001700 012637 000006 MOV (SP)+,@#6
959 001704 005737 000042 TST @#42 ;UNDER MONITOR
960 001710 001014 BNE 67$
961 ;*****THE NEXT 4 LINES OF CODE MOVED TO SOLVE PR#2757 (JUNE 78)*****
962 001712 105737 001310 TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED?
963 001716 001035 BNE 12$ ;IF YES, BR
964 001720 104402 001000 TYPE ,MTITLE ;TYPE TITLE MESSAGE
965 001724 105137 001310 COMB INIFLG ;IF NOT SET FLAG AND INIT
966 001730 022737 000176 001200 CMP #SWREG,SWR ;IS SWREG USED
967 001736 001001 BNE 67$
968 001740 104415 CNTLU
969 001742 105777 177232 67$: TSTB @SWR
970 001746 100402 BMI .+6
971 001750 004737 000220 JSR PC,CSRMAP
972 001754 104402 015612 TYPE ,XHEAD
973 001760 012737 001400 001244 MOV #1400,TEMP1
974 001766 017737 177252 001246 MOV @TEMP1,TEMP2
975 001774 001406 BEQ .+16
976 001776 104410 CONVRT
977 002000 015640 XSTATQ
978 002002 062737 000002 001244 ADD #2,TEMP1
979 002010 000766 BR .-22
980 002012 032777 000001 177160 12$: BIT #SW00,@SWR
981 002020 001424 BEQ 1$
982 002022 104402 TYPE
983 002024 015533 MNEW
984 002026 005000 CLR R0
985 002030 000000 HALT
986 002032 104414 CKSWR
987 002034 027737 177140 001502 CMP @SWR,SAVACT

```

```

988 002042 101404      BLOS      11$
989 002044 104402      TYPE
990 002046 015374      MERR3
991 002050 000000      HALT
992 002052 000776      BR        .-2
993 002054 017737 177120 001500 11$:  MOV      @SWR,DQACTV
994 002062 013700 001500      MOV      DQACTV,R0
995 002066 000000      HALT
996 002070 104414      CKSWR
997 002072 012700 000300 1$:  MOV      #300,R0
998 002076 012701 000302      MOV      #302,R1
999 002102 010120 2$:  MOV      R1,(R0)+
1000 002104 005021      CLR      (R1)+
1001 002106 022021      CMP      (R0)+,(R1)+
1002 002110 022700 001000      CMP      #1000,R0
1003 002114 001372      BNE      2$
1004
1005                      ;TEST START AND RESTART
1006
1007 002116 012737 000340 177776 .BEGIN: MOV      #340,PS          ;LOCK OUT INTERRUPTS
1008 002124 012706 001200      MOV      #STACK,SP      ;SET UP STACK
1009 002130 005737 000042      TST      @#42          ;IS PROGRAM UNDER MONITOR CONTROL
1010 002134 001040      BNE      3$
1011 002136 104414      CKSWR          ;CHECK FOR <^G>
1012 002140 032777 000004 177032      BIT      #BIT2,@SWR    ;CHECK FOR LOCK ON TEST
1013 002146 001411      BEQ      1$
1014 002150 104402 015432      TYPE      ,M:LOCK
1015 002154 012737 000240 013246      MOV      #NOP,TTST
1016 002162 012737 000240 013250      MOV      #NOP,TTST+2   ;SET UP TO LOCK
1017 002170 000406      BR        2$
1018 002172 013737 013344 013246 1$:  MOV      BRW,TTST
1019 002200 013737 013346 013250      MOV      BRX,TTST+2
1020 002206 032777 000002 176764 2$:  BIT      #SW01,@SWR    ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1021 002214 001410      BEQ      3$          ;IF SW01-1, GET STARTING PC
1022 002216 104403      INSTR
1023 002220 015420      MTSTPC
1024 002222 104405      PARAM
1025 002224 002254      TST1
1026 002226 010214      TLAST
1027 002230 001214      #RETURN
1028 002232 001          .BYTE 1
1029 002233 001          .BYTE 1
1030 002234 000403      BR        4$
1031 002236 012737 002254 001214 3$:  MOV      #TST1,RETURN   ;START AT TEST 1
1032 002244 104402 015322 4$:  TYPE      ,MR          ;TYPE R
1033 002250 000177 176740      JMP      @RETURN      ;START TESTING
1034
1035                      ; TEST 1
1036 002254 012737 000001 001226 TST1: MOV      #1,TSTNO
1037 002262 012737 002644 001214      MOV      #TST2,RETURN
1038 002270 012737 002644 001216      MOV      #TST2,NEXT
1039 002276 105737 001302      TSTB     RUNFLG      ;IS THIS MY FIRST TIME HERE?
1040 002302 001010      BNE      1$          ;BR IF FLAG IS SET
1041 002304 012737 000001 001304      MOV      #BIT0,RUN     ;SET RUN POINTER.
1042 002312 012737 000020 001306      MOV      #16.,RUNCNT  ;SET FOR MAX OF 16 DQ11'S PER SYSTEM
1043 002320 105137 001302      COMB     RUNFLG      ;SET RUN FLAG

```

```

1044 002324 033737 001304 001500 1$: BIT RUN,DQACTV ;FIND AN ACTIVE DQ11 TO TEST.
1045 002332 001032 BNE 3$ ;BR IF I FOUND ONE TO TEST.
1046 002334 005737 001500 TST DQACTV ;FIND OUT IF THERE ARE NO DQ11 ACTIVE.
1047 002340 001423 BEQ 2$ ;BR TO FATAL ERROR. WHY AM I HERE IF NO ACTIVE DQ11'S???
1048 002342 000257 CCC ;CLEAR ALL THE CONDITION CODES OF CPU
1049 002344 006137 001304 ROL RUN ;UPDATE RUN POINTER
1050 002350 062737 000004 001300 ADD #4,CREAM ;UPDATE ADDRESS POINTER.
1051 002356 005337 001306 DEC RUNCNT ;DEC NUMBER OF TIMES I LOOKED AT ACTIVE.
1052 002362 001360 BNE 1$ ;BR AND KEEP LOOKING.
1053 002364 012737 000020 001306 MOV #16,RUNCNT ;START RESTORING MY POINTERS.
1054 002372 012737 001400 001300 MOV #1400,CREAM ;RESTORE ADDRESS POINTER
1055 002400 012737 000001 001304 MOV #1,RUN ;RESTORE RUN POINTER.
1056 002406 000746 BR 1$ ;KEEP ON TESTING.
1057 002410 104402 2$: TYPE ;ALLERT OPERATOR OF FATAL ERROR
1058 002412 015325 MERR2 ;NO DQ11 ACTIVE. WHY AM I HERE???
1059 002414 000000 HALT ;YOU MUST RELOAD DQ11 DIAGNOSTIC.!
1060 002416 000776 BR .-2 ;STICK HERE ON CONT.
1061 002420 000257 3$: CCC ;CLEAR CPU COND. CODES
1062 002422 006137 001304 ROL RUN ;UPDATE RUN. ACTIVE DQ11 FOUND.
1063 002426 017737 176646 001506 MOV @CREAM,DQCSR ;PLACE ADDRESS OF DQ11 AT DQCSR
1064 002434 062737 000002 001300 ADD #2,CREAM ;UPDATE ADDRESS POINTER
1065 002442 017737 176632 001510 MOV @CREAM,DQSTAT ;PLACE STATUS OF DQ11 AT DQSTAT
1066 002450 062737 000002 001300 ADD #2,CREAM ;UPDATE ADDRESS POINTER
1067 002456 013737 00106 001360 MOV DQCSR,DQRCSR
1068 002464 013737 001510 001350 MOV DQSTAT,DQRVEC
1069 002472 042737 177007 001350 BIC #177007,DQRVEC
1070 002500 013737 001350 001352 MOV DQRVEC,DQRLVL ;GENERATE ADDRESS OF RECEIVER INTERRUPT SERVICE PS
1071 002506 062737 000002 001352 ADD #2,DQRLVL
1072 002514 013737 001352 001354 MOV DQRLVL,DQTVEC ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT VECTOR
1073 002522 062737 000002 001354 ADD #2,DQTVEC
1074 002530 013737 001354 001356 MOV DQTVEC,DQTLVL ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT SERVICE PS
1075 002536 062737 000002 001356 ADD #2,DQTLVL
1076 002544 013737 001360 001362 MOV DQRCSR,DQRCSH
1077 002552 005237 001362 INC DQRCSH ;GENERATE ADDRESS OF HIGH BYTE
1078 002556 013737 001360 001364 MOV DQRCSR,DQTCSR ;GENERATE ADDRESS OF TRANSMITTER CONTROL REGISTER
1079 002564 062737 000002 001364 ADD #2,DQTCSR
1080 002572 013737 001364 001366 MOV DQTCSR,DQERR ;GENERATE ADDRESS OF ERROR REGISTER
1081 002600 062737 000002 001366 ADD #2,DQERR
1082 002606 013737 001366 001370 MOV DQERR,DQREG ;GENERATE ADDRESS OF HIGH BYTE OF ERROR REGISTER
1083 002614 005237 001370 INC DQREG
1084 002620 013737 001370 001372 MOV DQREG,DQSEC ;GENERATE ADDRESS OF SECONDARY REGISTER
1085 002626 005237 001372 INC DQSEC
1086 002632 013737 001372 001374 MOV DQSEC,DQSECH ;GENERATE ADDRESS OF HIGH BYTE
1087 002640 005237 001374 INC DQSECH
1088 ;THIS IS NOT A TEST...!
1089 ;
1090 ; TEST 2
1091 ;*****
1092 002644 012737 000002 001226 TST2: MOV #2,TSTNO
1093 002652 012737 002662 001216 MOV #TST3,NEXT
1094 ;THE ORIGINAL TEST 2 IS
1095 ;NOW TEST 2 OF TAPE DZDQH
1096 002660 104400 SCOPE
1097
1098
1099

```

```

1100                                     :TEST TO SEE IF EVERY CHARACTER FROM
1101                                     :0 TO 377 CAN BE DETECTED IN CHARACTER
1102                                     :DETECT ADDRESS ZERO.
1103                                     :NOTE: SW09=1 WILL FREEZE ON CURRENT DATA.
1104
1105                                     : TEST 3
1106                                     :*****
1107 002662 012737 000003 001226 TST3: MOV #3,TSTNO
1108 002670 012737 003214 001216 MOV #TST4,NEXT
1109 002676 012737 002744 001220 MOV #1$,LOCK
1110 002704 032737 020000 001510 BIT #BBBIT,DQSTAT ;DOES THIS DQ11 HAVE THE 'BB' OPTION INSTALLED?
1111 002712 001005 BNE 13$ ;BR IF YES
1112 002714 012737 013050 001214 MOV #.EOP,RETURN ;GOTO END PASS
1113 002722 000177 176266 IMP @RETURN ;
1114 002726 104413 13$: MEMCLR ;CLEAR ALL THE DQ11
1115 002730 105037 013047 6$: CLRB DETCAR+1 ;CLEAR THE CHARACTER STORAGE AREA
1116 002734 005037 013044 CLR GDCHAR ;SET FOR ERROR PRINTOUT
1117 002740 005037 013042 CLR ADDR ;SAME.
1118 002744 012737 000010 013040 1$: MOV #8,COUNT ;EIGHT BITS FOR EIGHT SHIFTS.
1119 002752 013702 013040 2$: MOV COUNT,R2 ;GET NUMBER OF SHIFTS PER CHAR.
1120 002756 105077 176400 CLRB @DQRCSH ;GET CHAR ADDR. ZERO
1121 002762 105077 176402 CLRB @DQREG ;GET RX BA PRI.
1122 002766 012777 012216 176376 MOV #RXBUFF,@DQSEC ;LOAD IT
1123 002774 105277 176370 INCB @DQREG ;GET RX WC PRI.
1124 003000 012777 000200 176364 MOV #200,@DQSEC ;LOAD IT
1125 003006 105077 176350 3$: CLRB @DQRCSH ;SELECT CHARACTER DET REG 0
1126 003012 113737 013047 013044 MOVB DETCAR+1,GDCHAR ;
1127 003020 112777 000010 176342 MOVB #10,@DQREG ;SELECT THE CHARACTER DET REGISTER.
1128 003026 013777 013046 176336 MOV DETCAR,@DQSEC ;LOAD THE CHARACTER TO BE DETECTED
1129 003034 112777 000014 176326 MOVB #14,@DQREG ;SELECT THE SEQUENCE REGISTER
1130 003042 012777 120000 176322 MOV #BIT15+BIT13,@DQSEC ;SET SINGLE CHARACTER REC AND SET FLAG.
1131 003050 112777 000012 176312 MOVB #12,@DQREG ;COMM
1132 003056 012777 004012 176306 MOV #4012,@DQSEC ;SELECT EIGHT BITS TEST LOOP AND AUTO STEP
1133 003064 052777 010001 176266 BIS #BIT12+BIT0,@DQRCSR ;SET RX ACTIVE AND RX GO
1134 003072 013737 013046 015714 MOV DETCAR,TEMP ;MOV CHARACTER TO WORKING AREA
1135 003100 105137 015715 COMB TEMP+1 ;COMPLIMENT DATA FOR USE ON THE BIT WINDOW
1136 003104 042777 000200 176260 4$: BIC #BIT7,@DQSEC ;IF PREVIOUSLY SET; CLEAR THE BIT WINDOW.
1137 003112 006037 015714 ROR TEMP ;SHIFT OUT THE BIT OF DATA.
1138 003116 013703 015714 MOV TEMP,R3 ;STORE CHAR
1139 003122 042703 177577 BIC #^C<BIT7>,R3 ;CLEAR ALL UNWANTED BITS
1140 003126 050377 176240 BIS R3,@DQSEC ;PLACE DATA ON THE BIT WINDOW
1141 003132 005277 176234 INC @DQSEC ;CLOCK UP
1142 003136 005377 176230 DEC @DQSEC ;CLOCK DOWN
1143 003142 005302 DEC R2 ;IS CHARACTER DONE YET
1144 003144 001357 BNE 4$ ;BR IF NOT DONE
1145 003146 005777 176206 TST @DQRCSR ;WAS THE CHARACTER REALLY DETECTED?
1146 003152 100401 BMI .+4 ;BR IF GOOD
1147 003154 104002 HLT 2 ;ERROR CHARACTER NOT DETECTED.
1148 003156 017737 176176 015714 MOV @DQRCSR,TEMP ;GET THE RECEIVER CSR.
1149 003164 042737 170377 015714 BIC #^C<7400>,TEMP ;CLEAR ALL BUT THE CHARACTER DET. ADDR.
1150 003172 005737 015714 TST TEMP ;WAS THE CHAR DET. IN ADDR ZERO?
1151 003176 001401 BEQ .+4 ;
1152 003200 104002 HLT 2 ;CHAR NOT DETECTED IN ADDR. ZERO..
1153 -----*LOCK*-----
1154 003202 104401 SCOP1 ;IF SW09-1; THEN GOTO ADDRESS IN 'LOCK'.
1155 -----

```



1156	003204	105237	013047	INCB	DETCAR+1	;HAVE I HIT MY LIMIT YET?
1157	003210	001260		BNE	2\$	;NO RETURN WITH UPDATED CHAR.
1158	003212	104400		SCOPE		;SCOPE TEST

1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169

```

:
: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

```

*****
TST4:  MOV    #4,TSTNO
      MOV    #TST5,NEXT
      MOV    #1$,LOCK
      CLR    DETCAR           ;SET CHAR TO ZERO
      CLR    GDCHAR          ;SET FOR ERROR
      CLR    ADDR            ;SAME
1$:
2$:    MOV    #16.,R2         ;SET COUNT FOR 16 BIT CHARS
3$:    CLRB   @DQREG          ;SEL THE RX BA PRI.
      MOV    #RXBUFF,@DQSEC ;LOAD IT
      INCB   @DQREG          ;SEL THE RX WC PRI.
      MOV    #200,@DQSEC     ;LOAD IT
      CLRB   @DQRCSH        ;GET CHAR ADD ZERO
      MOVB   #10,@DQREG      ;GET CHAR ADDRESS
      MOV    DETCAR,@DQSEC   ;LOAD THE CHARACTER TO BE DETCETED
      MOVB   #14,@DQREG      ;GET THE SEQ REG.
      MOV    #BIT12+BIT14,@DQSEC ;LOAD DBL CHAR AND SET FLAG
      MOVB   #12,@DQREG      ;SEL MISC REG.
      MOV    #BIT11+BIT3,@DQSEC ;SET TEST LOOP AND AUTO/STEP
      BIS    #BIT12+BIT0,@DQRCSR ;SET RX ACTIVE AND GO
      MOV    DETCAR,TEMP     ;GET DATA CHAR.
      MOV    DETCAR,GDCHAR   ;FOR ERROR
      COM    TEMP            ;PREPARE FOR BIT WINDOW
1194:  BIC    #BIT7,@DQSEC    ;ZERO BIT WINDOW
      CLC
      CLR    TEMP1
      ROR    TEMP
      RORB   TEMP1
      BIS    TEMP1,@DQSEC    ;PLACE DATA ON BIT WINDOW
      INC    @DQSEC          ;CLOCK THE
      DEC    @DQSEC          ;
      DEC    R2              ;IS ALL THE CHAR DONE?
      BNE    4$              ;BR IF NO
      TST    @DQRCSR        ;DID THE FLAG SET?
      BMI    .+4             ;BR IF YES
      HLT    2               ;CHARACTER DET. FLAG NOT SET FOR DBL CHAR.
      MOV    @DQRCSR,TEMP    ;GET THE RECEIVER CSR.
      BIC    #^C<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..

```

1210  
1211

```

1212 ;----- *LOCK* -----
1213 003502 104401 SCOPE1 ;IF SW09=1; THEN GOTO ADDRESS IN 'LOCK'.
1214 ;-----
1215 003504 105237 013047 INCB DETCAR+1 ;UPDATE THE DATA
1216 003510 001260 BNE 1$ ;ALL DONE?
1217 003512 104400 SCOPE ;SCOPE WHEN ALL DATA DONE.
1218
1219
1220
1221
1222 ;TEST THAT A CHARACTER CAN
1223 ;BE DETECTED IN ALL 16 CHARACTER
1224 ;DETECT ADDRESSES
1225
1226
1227 ;TEST THAT THE CHARACTER 255
1228 ;CAN BE DETECTED IN CHARACTER
1229 ;DETECT ADDRESS 00
1230
1231 ; TEST 5
1232 ;*****
1233 003514 012737 000005 001226 TST5: MOV #5,TSTNO
1234 003522 012737 003544 001216 MOV #TST6,NEXT
1235 003530 012737 000000 013042 MOV #00,ADDR ;LOAD THE ADDRESS
1236 003536 004737 004314 JSR PC,CHK.ADD ;GO AND LOAD THE CHARACTER.
1237 003542 104400 SCOPE ;SCOPE THIS TEST
1238
1239
1240 ;TEST THAT THE CHARACTER 255
1241 ;CAN BE DETECTED IN CHARACTER
1242 ;DETECT ADDRESS 01
1243
1244 ; TEST 6
1245 ;*****
1246 003544 012737 000006 001226 TST6: MOV #6,TSTNO
1247 003552 012737 003574 001216 MOV #TST7,NEXT
1248 003560 012737 000001 013042 MOV #01,ADDR ;LOAD THE ADDRESS
1249 003566 004737 004314 JSR PC,CHK.ADD ;GO AND LOAD THE CHARACTER.
1250 003572 104400 SCOPE ;SCOPE THIS TEST
1251
1252
1253 ;TEST THAT THE CHARACTER 255
1254 ;CAN BE DETECTED IN CHARACTER
1255 ;DETECT ADDRESS 02
1256
1257 ; TEST 7
1258 ;*****
1259 003574 012737 000007 001226 TST7: MOV #7,TSTNO
1260 003602 012737 003624 001216 MOV #TST10,NEXT
1261 003610 012737 000002 013042 MOV #02,ADDR ;LOAD THE ADDRESS
1262 003616 004737 004314 JSR PC,CHK.ADD ;GO AND LOAD THE CHARACTER.
1263 003622 104400 SCOPE ;SCOPE THIS TEST
1264
1265
1266 ;TEST THAT THE CHARACTER 255
1267 ;CAN BE DETECTED IN CHARACTER

```

```

1268           ;DETECT ADDRESS 03
1269           ;
1270           ; TEST 10
1271           ;*****
1272 003624 012737 000010 001226 TST10: MOV #10,TSTNO
1273 003632 012737 003654 001216      MOV #TST11,NEXT
1274 003640 012737 000003 013042      MOV #03,ADDR ;LOAD THE ADDRESS
1275 003646 004737 004314           JSR PC,CHK.ADD ;GO AND LOAD THE CHARACTER.
1276 003652 104400           SCOPE ;SCOPE THIS TEST
1277
1278           ;
1279           ;TEST THAT THE CHARACTER 255
1280           ;CAN BE DETECTED IN CHARACTER
1281           ;DETECT ADDRESS 04
1282           ;
1283           ; TEST 11
1284           ;*****
1285 003654 012737 000011 001226 TST11: MOV #11,TSTNO
1286 003662 012737 003704 001216      MOV #TST12,NEXT
1287 003670 012737 000004 013042      MOV #04,ADDR ;LOAD THE ADDRESS
1288 003676 004737 004314           JSR PC,CHK.ADD ;GO AND LOAD THE CHARACTER.
1289 003702 104400           SCOPE ;SCOPE THIS TEST
1290
1291           ;
1292           ;TEST THAT THE CHARACTER 255
1293           ;CAN BE DETECTED IN CHARACTER
1294           ;DETECT ADDRESS 05
1295           ;
1296           ; TEST 12
1297           ;*****
1298 003704 012737 000012 001226 TST12: MOV #12,TSTNO
1299 003712 012737 003734 001216      MOV #TST13,NEXT
1300 003720 012737 000005 013042      MOV #05,ADDR ;LOAD THE ADDRESS
1301 003726 004737 004314           JSR PC,CHK.ADD ;GO AND LOAD THE CHARACTER.
1302 003732 104400           SCOPE ;SCOPE THIS TEST
1303
1304           ;
1305           ;TEST THAT THE CHARACTER 255
1306           ;CAN BE DETECTED IN CHARACTER
1307           ;DETECT ADDRESS 06
1308           ;
1309           ; TEST 13
1310           ;*****
1311 003734 012737 000013 001226 TST13: MOV #13,TSTNO
1312 003742 012737 003764 001216      MOV #TST14,NEXT
1313 003750 012737 000006 013042      MOV #06,ADDR ;LOAD THE ADDRESS
1314 003756 004737 004314           JSR PC,CHK.ADD ;GO AND LOAD THE CHARACTER.
1315 003762 104400           SCOPE ;SCOPE THIS TEST
1316
1317           ;
1318           ;TEST THAT THE CHARACTER 255
1319           ;CAN BE DETECTED IN CHARACTER
1320           ;DETECT ADDRESS 07
1321           ;
1322           ; TEST 14
1323           ;*****

```

```

1324 003764 012737 000014 001226 TST14: MOV #14,TSTNO
1325 003772 012737 004014 001216      MOV #TST15,NEXT
1326 004000 012737 000007 013042      MOV #07,ADDR      ;LOAD THE ADDRESS
1327 004006 004737 004314      JSR PC,CHK.ADD    ;GO AND LOAD THE CHARACTER.
1328 004012 104400      SCOPE             ;SCOPE THIS TEST
1329
1330      ;
1331      ;TEST THAT THE CHARACTER 255
1332      ;CAN BE DETECTED IN CHARACTER
1333      ;DETECT ADDRESS 10
1334      ;
1335      ; TEST 15
1336      ;*****
1337 004014 012737 000015 001226 TST15: MOV #15,TSTNO
1338 004022 012737 004044 001216      MOV #TST16,NEXT
1339 004030 012737 000010 013042      MOV #10,ADDR      ;LOAD THE ADDRESS
1340 004036 004737 004314      JSR PC,CHK.ADD    ;GO AND LOAD THE CHARACTER.
1341 004042 104400      SCOPE             ;SCOPE THIS TEST
1342
1343      ;
1344      ;TEST THAT THE CHARACTER 255
1345      ;CAN BE DETECTED IN CHARACTER
1346      ;DETECT ADDRESS 11
1347      ;
1348      ; TEST 16
1349      ;*****
1350 004044 012737 000016 001226 TST16: MOV #16,TSTNO
1351 004052 012737 004074 001216      MOV #TST17,NEXT
1352 004060 012737 000011 013042      MOV #11,ADDR      ;LOAD THE ADDRESS
1353 004066 004737 004314      JSR PC,CHK.ADD    ;GO AND LOAD THE CHARACTER.
1354 004072 104400      SCOPE             ;SCOPE THIS TEST
1355
1356      ;
1357      ;TEST THAT THE CHARACTER 255
1358      ;CAN BE DETECTED IN CHARACTER
1359      ;DETECT ADDRESS 12
1360      ;
1361      ; TEST 17
1362      ;*****
1363 004074 012737 000017 001226 TST17: MOV #17,TSTNO
1364 004102 012737 004124 001216      MOV #TST20,NEXT
1365 004110 012737 000012 013042      MOV #12,ADDR      ;LOAD THE ADDRESS
1366 004116 004737 004314      JSR PC,CHK.ADD    ;GO AND LOAD THE CHARACTER.
1367 004122 104400      SCOPE             ;SCOPE THIS TEST
1368
1369      ;
1370      ;TEST THAT THE CHARACTER 255
1371      ;CAN BE DETECTED IN CHARACTER
1372      ;DETECT ADDRESS 13
1373      ;
1374      ; TEST 20
1375      ;*****
1376 004124 012737 000020 001226 TST20: MOV #20,TSTNO
1377 004132 012737 004154 001216      MOV #TST21,NEXT
1378 004140 012737 000013 013042      MOV #13,ADDR      ;LOAD THE ADDRESS
1379 004146 004737 004314      JSR PC,CHK.ADD    ;GO AND LOAD THE CHARACTER.

```

```

1380 004152 104400          SCOPE          ;SCOPE THIS TEST
1381
1382
1383          ;TEST THAT THE CHARACTER 255
1384          ;CAN BE DETECTED IN CHARACTER
1385          ;DETECT ADDRESS 14
1386
1387          ; TEST 21
1388          ;*****
1389 004154 012737 000021 001226 TST21: MOV      #21,TSTNO
1390 004162 012737 004204 001216      MOV      #TST22,NEXT
1391 004170 012737 000014 013042      MOV      #14,ADDR          ;LOAD THE ADDRESS
1392 004176 004737 004314          JSR      PC,CHK.ADD        ;GO AND LOAD THE CHARACTER.
1393 004202 104400          SCOPE          ;SCOPE THIS TEST
1394
1395          ;
1396          ;TEST THAT THE CHARACTER 255
1397          ;CAN BE DETECTED IN CHARACTER
1398          ;DETECT ADDRESS 15
1399
1400          ; TEST 22
1401          ;*****
1402 004204 012737 000022 001226 TST22: MOV      #22,TSTNO
1403 004212 012737 004234 001216      MOV      #TST23,NEXT
1404 004220 012737 000015 013042      MOV      #15,ADDR          ;LOAD THE ADDRESS
1405 004226 004737 004314          JSR      PC,CHK.ADD        ;GO AND LOAD THE CHARACTER.
1406 004232 104400          SCOPE          ;SCOPE THIS TEST
1407
1408          ;
1409          ;TEST THAT THE CHARACTER 255
1410          ;CAN BE DETECTED IN CHARACTER
1411          ;DETECT ADDRESS 16
1412
1413          ; TEST 23
1414          ;*****
1415 004234 012737 000023 001226 TST23: MOV      #23,TSTNO
1416 004242 012737 004264 001216      MOV      #TST24,NEXT
1417 004250 012737 000016 013042      MOV      #16,ADDR          ;LOAD THE ADDRESS
1418 004256 004737 004314          JSR      PC,CHK.ADD        ;GO AND LOAD THE CHARACTER.
1419 004262 104400          SCOPE          ;SCOPE THIS TEST
1420
1421          ;
1422          ;TEST THAT THE CHARACTER 255
1423          ;CAN BE DETECTED IN CHARACTER
1424          ;DETECT ADDRESS 17
1425
1426          ; TEST 24
1427          ;*****
1428 004264 012737 000024 001226 TST24: MOV      #24,TSTNO
1429 004272 012737 004562 001216      MOV      #TST25,NEXT
1430 004300 012737 000017 013042      MOV      #17,ADDR          ;LOAD THE ADDRESS
1431 004306 004737 004314          JSR      PC,CHK.ADD        ;GO AND LOAD THE CHARACTER.
1432 004312 104400          SCOPE          ;SCOPE THIS TEST
1433
1434 004314          CHK.ADD:
1435 004314 104413          MEMCLR          ;CLEAR ALL THE DQ11 REGISTERS.

```

```

1436 004316 113777 013042 175036      MOVB  ADDR,@DQRCSH      ;LOAD THE CHAR DET ADDRESS TO BE USED.
1437 004324 005037 013046              CLR  DETCAR             ;CLEAR WORKING LOC.
1438 004330 112737 000255 013047      MOVB  #255,DETCAR+1    ;LOAD THE CHARACTER TO BE DETECTED IN WORKING AREA.
1439 004336 012737 000255 013044      MOV   #255,GDCHAR      ;LOAD FOR ERROR PRINTOUT
1440 004344 013702 013040              1$:  MOV   COUNT,R2      ;EIGHT SHIFTS FOR EIGHT BITS (NO VRC)
1441 004350 105077 175014              2$:  CLRB  @DQREG         ;SEL THE RX BA PRI.
1442 004354 012777 012216 175010      MOV   #RXBUFF,@DQSEC  ;LOAD IT
1443 004362 105277 175002              INCB  @DQREG           ;SEL THE RX WC PRI.
1444 004366 012777 177777 174776      MOV   #-1,@DQSEC      ;SET FOR ONE CHAR.
1445 004374 112777 000010 174766      MOVB  #10,@DQREG       ;SELECT THE CHARACTER DET REGISTER.
1446 004402 013777 013046 174762      MOV   DETCAR,@DQSEC   ;SET THE CHARACTER TO BE DETECTED INTO DQ11 CHAR DET REG
1447 004410 112777 000014 174752      MOVB  #14,@DQREG       ;SELECT THE SEQUENCE REGISTER.
1448 004416 012777 120000 174746      MOV   #BIT15+BIT13,@DQSEC ;SET SINGLE CHAR DET AND SINGLE CHAR DET FLAG SE
1449 004424 112777 000012 174736      MOVB  #12,@DQREG       ;SELECT THE MISC REGISTER.
1450 004432 012777 004012 174732      MOV   #4012,@DQSEC    ;SET EIGHT BITS TEST LOOP AND AUTO STEP
1451 004440 052777 010001 174712      BIS   #BIT12+BIT0,@DQRCSR ;SET RX ACTIVE AND RX GO.
1452 004446 013737 013046 015714      MOV   DETCAR,TEMP     ;MOVE THE CHAR TO BE DET TO WORKING AREA
1453 004454 105137 015715              COMB  TEMP+1           ;COMPLEMENT CHAR FOR USE ON BIT WINDOW.
1454 004460 042777 000200 174704      3$:  BIC   #BIT7,@DQSEC    ;IF BIT WINDOW SET ON LAST PASS CLEAR IT.
1455 004466 006037 015714              ROR   TEMP             ;SHIFT OUT BIT OF DATA.
1456 004472 013703 015714              MOV   TEMP,R3         ;SAVE IT
1457 004476 042703 177577              BIC   #^C<BIT7>,R3    ;CLEAR ALL UNWANTED BITS
1458 004502 050377 174664              BIS   R3,@DQSEC       ;PLACE DATA ON BIT WINDOW.
1459 004506 005277 174660              INC   @DQSEC          ;CLOCK UP
1460 004512 005377 174654              DEC   @DQSEC          ;CLOCK DOWN
1461 004516 005302              DEC   R2              ;IS CHARACTER DONE YET?
1462 004520 001357              BNE   3$              ;BR IF NOT DONE
1463 004522 005777 174632              TST   @DQRCSR         ;WAS CHAR REALLY DETECTED?
1464 004526 100401              BMI   .+4             ;BR IF GOOD
1465 004530 104002              HLT   2               ;CHARACTER DETECT FAILED.
1466 004532 017737 174622 015714      MOV   @DQRCSR,TEMP    ;GET THE RECEIVER CSR.
1467 004540 042737 170377 015714      BIC   #^C<7400>,TEMP ;CLEAR ALL BUT CHARACTER DET. ADDR.
1468 004546 123737 013042 015715      CMPB  ADDR,TEMP+1     ;WAS THE CHARACTER REALLY DETECTED
1469                                ;IN ADDRESS $A ??
1470                                BEQ   .+4             ;
1471                                HLT   2               ;WRONG ADDRESS.
1472                                RTS   PC              ;
1473                                ;
1474                                ;TEST OF RECEIVER AND TRANSMITTER 'SET 'T' '
1475                                ;TEST OF BIT ONE OF SEQUENCE REGISTER.
1476                                ;THIS TEST WILL 'SET T' AND THEN WILL
1477                                ;SEND A CHAR WHICH WILL 'SET DONE CLEAR GO'; IF
1478                                ;REALLY IN TRANSPARENT MODE THE CHAR WILL NOT BE DETECTED
1479                                ;AND THE WORD COUNTS WILL GOTO ZERO.
1480                                ;
1481                                ; TEST 25
1482                                ;*****
1483 004562 012737 000025 001226      TST25: MOV  #25,TSTNO
1484 004570 012737 004722 001216      MOV   #TST26,NEXT
1485
1486 004576 004737 011024              2$:  JSR   PC,SET.UP     ;SET UP ALL NECESSARY FOR TEST.
1487 004602 012777 040002 174562      MOV   #BIT14+BIT11,@DQSEC ;SET DBL. CHAR AND SET T
1488                                ;
1489 004610 105377 174546      DECB  @DQRCSH         ;SELECT ADD 16 (8)
1490 004614 112777 000010 174546      MOVB  #10,@DQREG       ;SELECT CHAR DET. ADDRESS
1491 004622 012777 164400 174542      MOV   #351*400,@DQSEC ;LOAD THE CHARACTER. SET DONE CLEAR GC.

```

```

1492 004630 112777 000014 174532      MOVB   #14,@DQREG      ;SELECT THE SEQ REGISTER.
1493 004636 012777 100200 174526      MOV    #BIT15+BIT7,@DQSEC
1494                                     ;SET FUNCTION. CLEAR GO SET DONE.
1495 004644 012700 012014      MOV    #TXBUFF,R0     ;PREPARE TX BUFFER
1496 004650 012720      MOV    (PC)+,(R0)+    ;LOAD THE BUFFER WITH DATA
1497 004652      000      350      .BYTE  000,350        ;DATA
1498 004654 012720      MOV    (PC)+,(R0)+    ;LOAD THE BUFFER WITH DATA
1499 004656      351      200      .BYTE  351,200        ;DATA
1500 004660 004737 011620      JSR    PC,ENABLE      ;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
1501 004664 112777 000001 174476      MOVB   #1,@DQREG      ;GET THE RX WC PRI.
1502 004672 005777 174474      TST   @DQSEC          ;RX WC PRI. SHOULD =0
1503 004676 001401      BEQ   +4              ;BR IF RX WC PRI =0
1504 004700 104003      HLT   3              ;RX PRI WC NOT =0
1505 004702 112777 000003 174460      MOVB   #3,@DQREG      ;GET TX WC PRI.
1506 004710 005777 174456      TST   @DQSEC          ;TX WC PRI SHOULD =0
1507 004714 001401      BEQ   +4              ;BR IF TX WC =0
1508 004716 104003      HLT   3              ;TX WC PRI NOT =0
1509 004720 104400      SCOPE

```

```

1510
1511
1512
1513 ; TEST OF RECEIVER 'CLEAR 'T' '
1514 ; TEST OF BIT TWO OF SEQUENCE REGISTER.
1515 ; THIS TEST WILL ENTER BOTH THE RECEIVER AND
1516 ; TRANSMITTER INTO TRANSPARENCY; THEN A CHARACTER
1517 ; WHICH SAYS 'CLEAR RX T' WILL BE SENT FOLLOWED
1518 ; BY A CHARACTER WHICH SAYS 'SET RX CHAR FLAG'.
1519 ; THE TEST THEN CHECKS THAT THE CHARACTER FLAG IS SET
1520 ; WHICH MEANS THAT CHARACTER WAS DETECTED.
1521

```

```

; TEST 26
;*****

```

```

1522
1523
1524 004722 012737 000026 001226  TST26: MOV    #26,TSTNO
1525 004730 012737 005072 001216      MOV    #TST27,NEXT
1526 004736 004737 011024      JSR    PC,SET.UP      ;SET UP ALL NECESSARY FOR TEST.
1527 004742 012777 040002 174422      MOV    #BIT14+BIT1,@DQSEC
1528                                     ;SET FUNCTION: DBL CHAR AND SET T
1529 004750 105377 174406      DECB  @DQRCSH         ;SELECT CHAR ADDRESS 16 (8)
1530 004754 112777 000010 174406      MOVB   #10,@DQREG     ;SELECT CHAR DET ADDRESS.
1531 004762 012777 165000 174402      MOV    #352*400,@DQSEC ;LOAD CHARACTER
1532 004770 112777 000014 174372      MOVB   #14,@DQREG     ;SELECT THE SEQ REGISTER.
1533 004776 012777 040004 174366      MOV    #BIT14+BIT2,@DQSEC
1534                                     ;SET FOR DBL CHAR AND CLEAR T
1535 005004 105377 174352      DECB  @DQRCSH         ;GET NEXT ADDR
1536                                     ;GET NEXT ADDRESS
1537 005010 112777 000010 174352      MOVB   #10,@DQREG     ;SELECT CHAR DET ADDRESS
1538 005016 012777 166400 174346      MOV    #355*400,@DQSEC ;LOAD CHARACTER.
1539 005024 112777 000014 174336      MOVB   #14,@DQREG     ;SELECT THE SEQ REGISTER.
1540 005032 012777 120000 174332      MOV    #BIT15+BIT13,@DQSEC
1541                                     ;SET FOR SINGLE CHAR AND SET FLAG.
1542 005040 012700 012014      MOV    #TXBUFF,R0     ;GET POINTER
1543 005044 012720      MOV    (PC)+,(R0)+    ;LOAD THE BUFFER WITH DATA
1544 005046      350      351      .BYTE  350,351        ;DATA
1545 005050 012720      MOV    (PC)+,(R0)+    ;LOAD THE BUFFER WITH DATA
1546 005052      352      355      .BYTE  352,355        ;DATA
1547 005054 004737 011620      JSR    PC,ENABLE      ;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.

```

1548 005060 005777 174274  
1549 005064 100401  
1550 005066 104004  
1551 005070 104400

TST @DQRCR ;CHECK CHAR DET FLAG  
BMI ;+4 ;BR IS SET  
HLT 4 ;CHARACTER DET FLAG NOT SET  
SCOPE ;SCOPE THIS TEST

1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563

TEST OF RECEIVER AND TRANSMITTER 'BCC/CLEAR START'  
TEST OF BIT THREE OF SEQUENCE REGISTER.  
THE TEST STARTS UP THE TRANSMITTER AND RECEIVER BCC  
AND DEPOSITS ONE CHARACTER INTO IT. THE RECEIVER  
DONE FLAG COMES UP AND THE DQ11 CLOCK IS STOPPED.  
THE BCC'S OF BOTH THE TX AND RX ARE THEN 'GRABBED'  
AND SHIFTED LOOKING FOR THAT ONE CHARACTER TO BE PRESENT  
IN THE BCC OF EACH RX AND TX BCC REGISTER.

1564  
1565  
1566 005072 012737 000027 001226  
1567 005100 012737 005350 001216  
1568 005106 004737 011024  
1569 005112 012777 100010 174252  
1570  
1571 005120 112777 000017 174242  
1572 005126 012777 000200 174236  
1573 005134 112777 000003 174226  
1574 005142 012777 177576 174222  
1575 005150 012700 012014  
1576 005154 105020  
1577 005156 022700 012215  
1578 005162 001374  
1579 005164 012700 012014  
1580 005170 012720  
1581 005172 350 225  
1582 005174 012737 000225 013036  
1583 005202 012737 000010 013040  
1584 005210 105077 174154  
1585 005214 012777 012216 174150  
1586 005222 112777 000001 174140  
1587 005230 012777 177775 174134  
1588 005236 112777 000004 174124  
1589 005244 012777 012236 174120  
1590 005252 112777 000005 174110  
1591 005260 012777 177577 174104  
1592 005266 004737 011674  
1593 005272 013703 013040  
1594 005276 000241  
1595 005300 106100  
1596 005302 005500  
1597 005304 023700 013036  
1598 005310 001403  
1599 005312 005303  
1600 005314 001370  
1601 005316 104005  
1602 005320 013703 013040  
1603 005324 000241

TEST 27

\*\*\*\*\*

TST27: MOV #27,TSTNO  
MOV #TST30,NEXT  
JSR PC,SETUP ;SET UP ALL NECESSARY FOR TEST.  
MOV #BIT15+BIT3,@DQSEC ;SET SNGL CHAR AND BCC/START CLEAR  
;SEL THE POLY REG.  
MOVB #17,@DQREG ;SET FOR LRC 8  
MOV #200,@DQSEC ;SEL THE TX WC PRI.  
MOVB #3,@DQREG ;SET FOR 202 (8)  
MOV #-202,@DQSEC ;GET TX BUFFER  
MOV #TXBUFF,R0 ;CLEAR ALL THE TX BUFFER  
1\$: CLRB (R0)+ ;ALL CLEAR?  
CMP #TXBUFF+201,R0 ;BR IF NO  
BNE 1\$ ;GET TX BUFFER  
MOV #TXBUFF,R0 ;LOAD THE BUFFER WITH DATA  
MOV (PC)+,(R0)+ ;DATA  
;SET EXPECTED BCC CHAR.  
MOV #225,CHAR ;SET FOR 8 BITS  
MOV #8,COUNT ;SEL REC PRIMARY  
CLRB @DQREG ;SET WITH START ADRS  
MOV #RXBUFF,@DQSEC ;SEL REC CHAR COUNT  
MOVB #1,@DQREG ;SET CHAR COUNT  
MOV #-3,@DQSEC ;SEL REC SECONDARY  
MOVB #4,@DQREG ;SET WITH SEC ADRS  
MOV #RXBUFF+20,@DQSEC ;SEL REC CHAR COUNT  
MOVB #5,@DQREG ;SET CHAR COUNT  
1592: JSR PC,NEWENA ;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.  
MOV COUNT,R3 ;GET CHAR SIZE  
6\$: CLC ;  
ROLB R0 ;SHIFT RX BCC IMAGE  
7\$: ADC R0 ;PICK UP CARRY  
CMP CHAR,R0 ;CHECK BCC  
BEQ 3\$ ;BCC OK!  
DEC R3 ;ALL SHIFTS DONE?  
BNE 6\$ ;BR IF NO.  
HLT 5 ;RX BCC HAS WRONG DATA.  
3\$: MOV COUNT,R3 ;SAVE COUNTER  
8\$: CLC ;



CZDQF MACY11 30(1046) 05-MAY-81 14:18 PAGE 33  
CZDQFF.P11 05-MAY-81 09:27

PROGRAM INITIALIZATION AND START UP.

SEQ 0032

1604	005326	106101		ROLB	R1	:SHIFT TX BCC IMAGE
1605	005330	005501		ADC	R1	:PICK UP CARRY
1606	005332	023701	013036	CMP	CHAR,R1	:IS BCC OK?
1607	005336	001403		BEQ	5\$	:BR IF OK
1608	005340	005303		DEC	R3	:ALL SHIFTS DONE?
1609	005342	001370		BNE	8\$	:BR IF NO
1610	005344	104005		HLT	5	:TX BCC HAS WRONG DATA.
1611	005346	104400		5\$: SCOPE		:SCOPE THE TEST

1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623 005350 012737 000030 001226  
1624 005356 012737 005532 001216  
1625 005364 004737 011024  
1626 005370 012777 100100 173774  
1627  
1628 005376 112737 000350 012016  
1629 005404 004737 005446  
1630 005410 012737 005416 001220  
1631 005416 004737 011236  
1632 005422 012777 040100 173742  
1633  
1634 005430 012737 003721 012426  
1635 005436 004737 005446  
1636  
1637 005442 104401  
1638  
1639 005444 104400  
1640 005446 005277 173706  
1641 005452 005277 173706  
1642 005456 105777 173702  
1643 005462 100375  
1644 005464 021616  
1645 005466 032777 010000 173664  
1646 005474 001401  
1647 005476 104006  
1648 005500 112777 000001 173662  
1649 005506 005777 173660  
1650 005512 001001  
1651 005514 104006  
1652 005516 122777 000001 173634  
1653 005524 001401  
1654 005526 104006  
1655 005530 000207  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667

```

: 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,@DQSEC
;SET SNGL CHAR AND CLEAR ACTV
MOVB #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,@DQSEC
;SET DBI 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'.
:-----
: SCOPE ;SCOPE THIS TEST
X.ABG: INC @DQRCSR ;SET RX GO.
INC @DQTCR ;SET TX GO
TSTB @DQTCR ;HANG HERE FOR TX PRI DONE!.
BPL .-4 ;BR IF NOT DONE.
CMP (SP),(SP) ;WAST TIME.
BIT #BIT12,@DQRCSR ;IS RX ACTIVE CLEARED?
BEQ .+4 ;BR IF YES
HLT 6 ;RX ACTIVE NOT CLEARED
MOVB #1,@DQREG ;GET THE RX WC PRI.
TST @DQSEC ;IT SHOULD BE NON-ZERO!!
BNE .+4 ;BR IF OK
HLT 6 ;RX PRI WC =0
CMPB #001,@DQRCSR ;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 DETCTED
: 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
:*****

```

```

1668 005532 012737 000031 001226 TST31: MOV #31,TSTNO
1669 005540 012737 005726 001216 MOV #TST32,NEXT
1670 005546 004737 011024 JSR PC,SET.UP ;SET UP ALL NECESSARY FOR TEST.
1671 005552 012777 100200 173612 MOV #BIT15+BIT7,@DQSEC
1672 ;SET SNGL CHAR AND SET DONE CLEAR GO
1673 005560 012700 012016 MOV #TXBUFF+2,R0 ;SET TX BUFFER
1674 005564 012710 MOV (PC)+,(R0) ;LOAD WITH DATA
1675 005566 350 352 .BYTE 350,352 ;DATA
1676 005570 004737 005640 JSR PC,X.ABF ;WORK DQ11
1677 005574 012737 005602 001220 MOV #1$,LOCK ;SET FOR RETURN IF SW09=1
1678 005602 004737 011236 1$: JSR PC,EXT.UP ;SET THING UP FOR DOUBLE CHAR.(16 BITS)
1679 005606 012777 040200 173556 MOV #BIT14+BIT7,@DQSEC
1680 ;SET DBL CHAR AND SET DONE CLEAR GO
1681 005614 012737 003721 012426 MOV #3721,XTXBUF+2 ;LOAD DATA
1682 005622 012737 012525 012430 MOV #12525,XTXBUF+4 ;SAME
1683 005630 004737 005640 JSR PC,X.ABF ;TURN ON DQ11
1684 ;-----*LOCK*-----
1685 005634 104401 SCOPE1 ;IF SW09=1; THEN GOTO ADDRESS IN 'LOCK'.
1686 ;-----
1687 005636 104400 SCOPE ;SCOPE THIS TEST.
1688
1689 005640 004737 011620 X.ABF: JSR PC,ENABLE ;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
1690 005644 112777 000001 173516 MOVB #1,@DQREG ;SET RX WC PRI
1691 005652 005777 173514 TST @DQSEC ;IT SB NON-ZERO.
1692 005656 001001 BNE .+4 ;BR IF OK
1693 005660 104007 HLT 7 ;RX PRI WC -0
1694 005662 112777 000003 173500 MOVB #3,@DQREG ;SEL THE TX WC PRI
1695 005670 005777 173476 TST @DQSEC ;IT SB NON-ZERO
1696 005674 001001 BNE .+4 ;BR IF OK.
1697 005676 104007 HLT 7 ;TX WC PRI IS -0
1698 005700 122777 000200 173452 CMPB #200,@DQRCR ;DONE=1; P/S=0; GO=0?
1699 005706 001401 BEQ .+4 ;BR IF OK.
1700 005710 104007 HLT 7 ;RX CSR NOT -200 (PRI DONE)
1701 005712 122777 000200 173444 CMPB #200,@DQTCR ;DONE=1; P/S=0; GO=0
1702 005720 001401 BEQ .+4 ;BR IF OK.
1703 005722 104007 HLT 7 ;TX PRI DONE SET? (TX CSR=200)
1704 005724 000207 RTS PC
1705
1706 ;
1707 ;TEST OF RECEIVER 'CHARACTER STRIP'
1708 ;TEST OF BIT EIGHT OF SEQUENCE REGISTER.
1709 ;THE CHARACTER THAT IS SENT AS 'CHARACTER STRIP' IS
1710 ;LOOKED FOR IN THE RX BUFFER; IF IT IS NOT FOUND IT
1711 ;IS ASSUMED THAT THE CHARACTER WAS INDEED 'STRIPPED'.
1712 ;
1713 ; TEST 32
1714 ;*****
1715 005726 012737 000032 001226 TST32: MOV #32,TSTNO
1716 005734 012737 006104 001216 MOV #TST33,NEXT
1717 005742 004737 011024 JSR PC,SET.UP ;SET UP ALL NECESSAR FOR TEST.
1718 005746 012777 100400 173416 MOV #BIT15+BIT8,@DQSEC
1719 ;SET SNGL CHAR AND CHAR STRIP.
1720 005754 012700 012016 MOV #TXBUFF+2,R0 ;SET POINTER
1721 005760 012710 MOV (PC)+,(R0) ;LOAD THE
1722 005762 350 321 .BYTE 350,321 ; DATA
1723 005764 004737 011620 JSR PC,ENABLE ;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.

```



```

1780 006212 012737 003721 012426      MOV    #3721,XTXBUF+2 ;LOAD DATA
1781 006220 012737 054321 012430      MOV    #54321,XTXBUF+4 ;SAME
1782 006226 004737 011620                JSR    PC,ENABLE      ;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
1783 006232 012700 012216                MOV    #RXBUFF,R0    ;GET POINTER
1784 006236 012701 000010                MOV    #10,R1        ;GET CHAR COUNTER
1785 006242 022720 177777                3$:   CMP    #177777,(R0)+ ;TX PAD IN BUFFER?
1786 006246 001001                        BNE    .+4           ;BR IF NO
1787 006250 104011                        HLT    11            ;PAD CHARACTER IS IN BUFFER.
1788 006252 005301                        DEC    R1            ;ALL CHARS DONE?
1789 006254 001372                        BNE    3$           ;BR IF NO.
1790 006256 104400                4$:   SCOPE

```

```

1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835

```

```

: TEST OF 'BCC EXCLUDE'
: TEST OF BIT 11 OF SEQUENCE REGISTER
: 'BCC EXCLUDE' 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
:*****

```

```

TST34: MOV    #34,TSTNO
MOV    #TST35,NEXT
JSR    PC,SET.UP ;SET UP ALL NECESSARY FOR TEST.
MOV    #BIT15+BIT3,@DQSEC
;SET SNGL CHAR AND BCC START CLEAR
DECB  @DQRC5H ;SEL CHAR ADD 16(8)
MOVB  #10,@DQREG ;GET CHAR DET ADDRESS
MOV    #361*400,@DQSEC ;LOAD CHAR.
MOVB  #14,@DQREG ;SEL SEQ REG
MOV    #BIT15+BIT11,@DQSEC
;SET SNGL CHAR AND BCC EXCLUDE
MOVB  #17,@DQREG ;SEL POLY REG.
MOV    #200,@DQSEC ;SET LRC 8
MOVB  #3,@DQREG ;SEL TX WC PRI.
MOV    #-202,@DQSEC ;SET BIG NUMBER
MOV    #TXBUFF,R0 ;SET POINTER
1$:   CLRB  (R0)+ ;LOAD DATA 000
CMP    #TXBUFF+201,R0 ;CLEAR BUFFER.!
BNE    1$ ;BR IF NOT ALL CLEAR.
MOV    #TXBUFF,R0 ;SET POINTER
MOV    (PC)+,(R0) ;LOAD THE
; .BYTE 350,107 ; DATA
MOVB  #361,TXBUF+2 ; INTO
MOV    #8,COUNT ; THE TX BUFFER.
MOV    #107,CHAR ;SET FOR 8 BITS AND 107 AS THE CHAR IN BCC.
CLRB  @DQREG ;SEL REC PRIMARY
MOV    #RXBUFF,@DQSEC ;SET WITH START ADRS
MOVB  #1,@DQREG ;SEL REC CHAR COUNT
MOV    #-3,@DQSEC ;SET CHAR COUNT
MOVB  #4,@DQREG ;SET REC SECONDARY
MOV    #RXBUFF+20,@DQSEC ;SET WITH SEC ADRS

```

```

1836 006502 112777 000005 172660      MOVB  #5,@DQREG      ;SEL CHAR COUNT
1837 006510 012777 177577 172654      MOV  #-201,@DQSEC    ;SET CHAR COUNT
1838 006516 004737 011674      JSR  PC,NEWENA      ;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
1839 006522 013703 013040      MOV  COUNT,R3      ;SAVE COUNTER
1840 006526 000241      4$: CLC              ;
1841 006530 106100      ROLB  R0            ;SHIFT RX BCC IMAGE
1842 006532 005500      5$: ADC  R0          ;SAVE CARRY
1843 006534 023700 013036      CMP  (CHAR,R0)     ;BCC OK?
1844 006540 001403      BEQ  3$            ;BR IF OK
1845 006542 005303      DEC  R3            ;ALL SHIFTS DONE?
1846 006544 001370      BNE  4$            ;BR IF NO
1847 006546 104012      HLT  12            ;RX BCC HAS WRONG DATA.
1848 006550 013703 013040      3$: MOV  COUNT,R3
1849 006554 000241      6$: CLC              ;CLEAR CARRY
1850 006556 106101      ROLB  R1            ;SHIFT TX BCC IMAGE
1851 006560 005501      ADC  R1            ;PICK UP CARRY
1852 006562 023701 013036      CMP  (CHAR,R1)     ;BCC OK?
1853 006566 001403      BEQ  7$            ;BR IF OK
1854 006570 005303      DEC  R3            ;ALL SHIFTS DONE?
1855 006572 001370      BNE  6$            ;BR IF NO
1856 006574 104012      HLT  12            ;TX BCC HAS WRONG DATA.
1857 006576 104400      7$: SCOPE          ;SCOPE THIS TEST

```

```

1858
1859
1860      ; TEST OF SET TRANSPARENCY FOR TRANSMITTER.
1861      ; TEST THAT THE SEQ FUNCTIONS ARE ALLOWED IF
1862      ; THEY ARE PRECEDED BY *DLE*.
1863      ; THIS TEST THAT WHEN THE TRANSMITTER FLIPS FROM PRI TO SEC
1864      ; AND 'EXIT T' IS ASSERTED THAT THE TX SENDS A 'DLE'
1865      ; CHARACTER.

```

```

1866
1867      ; TEST 35
1868      ;*****

```

```

1869 006600 012737 000035 001226  TST35: MOV  #35,TSTNO
1870 006606 012737 007054 001216      MOV  #TST36,NEXT
1871 006614 004737 011024      JSR  PC,SET.UP     ;SET UP ALL NECESSARY FOR TEST.
1872 006620 012777 040002 172544      MOV  #BIT14+BIT1,@DQSEC
1873      ;SET DBL CHAR AND SET T
1874 006626 105377 172530      DECB @DQRCSH      ;SEL CHAR ADD 16
1875 006632 112777 000010 172530      MOVB #10,@DQREG   ;GET CHAR DET REG
1876 006640 013700 001372      MOV  DQSEC,R0
1877 006644 012710      MOV  (PC)+,(R0)   ;LOAD THE REGISTER
1878 006646      352      .BYTE 352,352    ;WITH THIS DATA
1879 006650 112777 000014 172512      MOVB #14,@DQREG   ;SEL THE SEQ REG.
1880 006656 012777 101000 172506      MOV  #BIT15+BIT9,@DQSEC
1881      ;SET SNGL CHAR AND DLE
1882 006664 105377 172472      DECB @DQRCSH      ;SEL CHAR ADD 15(8)
1883 006670 112777 000010 172472      MOVB #10,@DQREG   ;GET CHAR ADDRESS
1884 006676 012777 166000 172466      MOV  #354*400,@DQSEC ;LOAD WITH DATA
1885 006704 112777 000014 172456      MOVB #14,@DQREG   ;SEL THE SEQ REGISTER
1886 006712 012777 100200 172452      MOV  #BIT15+BIT7,@DQSEC
1887      ;SET SNGL CHAR AND SET DONE CLEAR GO
1888 006720 112777 000003 172442      MOVB #3,@DQREG    ;SEL THE TX WC PRI.
1889 006726 012777 177772 172436      MOV  #-6,@DQSEC   ;SET FOR 6 CHARS
1890 006734 112737 000350 012015      MOVB #350,TXBUFF+1 ;LOAD
1891 006742 112737 000357 012020      MOVB #357,TXBUFF+4 ; DATA

```

```

1892 006750 112737 000354 012054      MOVB    #354, TXBUFF+40 ;EXIT T SHOULD SEND DLE TTIS SHOULD SET DONE THEN. !
1893 006756 112777 000006 172404      MOVB    #6, @DQREG      ;SEL TX BA SEC.
1894 006764 012777 012054 172400      MOV     #TXBUFF+40, @DQSEC
1895 006772 105277 172372      INCB   @DQREG          ;LOAD BA AND SEL WC SEC.
1896 006776 052777 030000 172362      BIS    #BIT13+BIT12, @DQERR
1897                                     ;SET WRITE ENABLE AND EXIT T
1898 007004 012777 177770 172360      MOV    #-10, @DQSEC    ;SET SEC. FOR 10(8) CHARS
1899 007012 004737 011620      JSR    PC, ENABLE     ;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
1900 007016 112777 000007 172344      MCVB   #7, @DQREG     ;SEL THE TX WC SEC.
1901 007024 005777 172342      TST    @DQSEC        ;SHOULD BE NON-ZERO.
1902 007030 001001      BNE    .+4           ;BR IF OK.
1903 007032 104003      HLT    3             ;TX DID NOT DET 'SET DONE CLEAR GO''
1904 007034 112777 000001 172326      MOVB   #1, @DQREG     ;SEL THE RX WC PRI.
1905 007042 005777 172324      TST    @DQSEC        ;RX WC PRI S/B NOT-0
1906 007046 001001      BNE    .+4           ;BR IF OK
1907 007050 104003      HLT    3             ;RX NOT DET CHAR SET DONE CLEAR GO.
1908 007052 104400      SCOPE                ;SCOPE THE TEST.

```

:  
: TEST THAT THE TRANSMITTER WILL EXIT T  
: WHEN ENTERED BY 'SET T'

: TEST 36

:\*\*\*\*\*

```

1919 007054 012737 000036 001226  TST36:  MOV    #36, TSTNO
1920 007062 012737 007260 001216      MOV    #TST37, NEXT
1921 007070 004737 011024      JSR    PC, SET.UP    ;SET UP ALL NECESSARY FOR TEST.
1922 007074 012777 040002 172270      MOV    #BIT14+BIT1, @DQSEC
1923                                     ;SET DBL CHAR AND SET T
1924 007102 105377 172254      DECB   @DQRC5H      ;GET CHAR ADDR 16(8)
1925 007106 112777 000010 172254      MOVB   #10, @DQREG   ;GET CHAR REG.
1926 007114 012777 166000 172250      MOV    #354*400, @DQSEC ;LOAD CHARACTER
1927 007122 112777 000014 172240      MOVB   #14, @DQREG   ;SEL THE SEQ REG.
1928 007130 012777 100200 172234      MOV    #BIT15+BIT7, @DQSEC
1929                                     ;SET SNGL CHAR AND SET DONE CLEAR GO.
1930 007136 112777 000006 172224      MOVB   #6, @DQREG    ;SEL TX BA SEC.
1931 007144 012777 012054 172220      MOV    #TXBUFF+40, @DQSEC
1932 007152 105277 172212      INCB   @DQREG        ;SEL TX WC SEC.
1933 007156 052777 030000 172202      BIS    #BIT13+BIT12, @DQERR
1934                                     ;SET WRITE ENABLE AND EXIT T
1935 007164 012777 177770 172200      MOV    #-10, @DQSEC  ;SET FOR 10((8) CHARS
1936 007172 112737 000350 012017      MOVB   #350, TXBUFF+3 ;LOAD DATA
1937 007200 112737 000354 012055      MOVB   #354, TXBUFF+41 ;SAME
1938 007206 005277 172152      INC    @DQTC5R      ;SET TX GO.
1939 007212 032777 000100 172144  1$:  BIT    #BIT6, @DQTC5R ;HANG HERE FOR TX SEC. DONE.
1940 007220 001774      BEQ    1$           ;BR IF NOT DONE
1941 007222 112777 000003 172140      MOVB   #3, @DQREG    ;GET TX WC PRI.
1942 007230 005777 172136      TST    @DQSEC        ;IS IT =0
1943 007234 001401      BEQ    .+4           ;BR IF OK
1944 007236 104003      HL     3             ;TX WC PRI NOT-0
1945 007240 112777 000007 172122      MCVB   #7, @DQREG     ;SEL TX WC SEC
1946 007246 005777 172120      TST    @DQSEC        ;IS NOT -0?
1947 007252 001001      BNE    .+4           ;BR IF OK

```

1948 007254 104003 HLT 3 ;TX SEC WC S/B NON-ZERO.  
1949 007256 104400 SCOPE ;SCOPE THIS TEST.

1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
: TEST 37  
:\*\*\*\*\*  
:TEST THAT THE RECEIVER STRIP SYNC IS  
:INHIBITED WHEN IN TRANSPARENT MODE.

1958 007260 012737 000037 001226 TST37: MOV #37,TSTNO  
1959 007266 012737 007470 001216 MOV #TST40,NEXT  
1960 007274 004737 011024 JSR PC,SET.UP ;SET UP ALL NECESSARY FOR TEST.  
1961 007300 012777 040002 172064 MOV #BIT14+BIT1,@DQSEC

1962 007306 105377 172050 DECB @DQRC5H ;SET DBL CHAR AND SET T  
1963 007312 112777 000010 172050 MOV #10,@DQREG ;SEL CHAR ADD 16(8)  
1964 007320 013700 001372 MOV DQSEC,R0 ;SEL CHAR REG  
1965 007324 012710 MOV (PC)+,(R0) ;LOAD THE CHAR  
1966 007326 101 101 .BYTE 101,101 ;DATA  
1967 007330 112777 000014 172032 MOV #14,@DQREG ;SEL THE SEQ REG  
1968 007336 012777 101000 172026 MOV #BIT15+BIT9,@DQSEC

1969 007344 112737 000350 012015 MOV #350,TXBUFF+1 ;SET SNGL CHAR AND DLE  
1970 007352 012700 012016 MOV #TXBUFF+2,R0 ;LOAD BUFFER  
1971 007356 112720 000101 1\$: MOV #101,(R0) ;DLE  
1972 007362 112720 000026 MOV #26,(R0)+ ;SYNC  
1973 007366 022700 012026 CMP #TXBUFF+12,R0 ;KEEP STUFFING  
1974 007372 001371 BNE 1\$ ;

1975 007374 012777 000003 171756 MOV #3,@DQRC5R ;SET STRIP SYNC AND GO(RX)  
1976 007402 005277 171756 INC @DQTC5R ;SET TX GO  
1977 007406 105777 171746 TSTB @DQRC5R ;HANG HERE FOR RX DONE (P)  
1978 007412 100375 BPL -4 ;

1979 007414 012700 012216 MOV #RXBUFF,R0 ;GET RX POINTER  
1980 007420 122720 000000 CMPB #0,(R0)+ ;FIRST CHAR S/B=0  
1981 007424 001401 BEQ +4 ;

1982 007426 104003 HLT 3 ;FIRST DATA CHAR WRONG  
1983 007430 122720 000350 CMPB #350,(R0)+ ;NEXT CHAR S/B=350  
1984 007434 001401 BEQ +4 ;

1985 007436 104003 HLT 3 ;RX BUFFER WRONG  
1986 007440 122720 000101 2\$: CMPB #101,(R0)+ ;DLE PRESENT?  
1987 007444 001401 BEQ +4 ;  
1988 007446 104003 HLT 3 ;DLE NOT THERE  
1989 007450 122720 000026 CMPB #26,(R0)+ ;SYNC PRESENT?  
1990 007454 001401 BEQ +4 ;

1991 007456 104003 HLT 3 ;LOOKS LIKE SYNC STRIPPED?  
1992 007460 022700 012226 CMP #RXBUFF+10,R0 ;BUFFER DONE?  
1993 007464 001365 BNE 2\$ ;BR IF NO  
1994 007466 104400 SCOPE ;SCOPE THIS TEST  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
:VERIFY THAT BIT 8 OF THE SEQUENCE  
:REGISTER STRIPS CHARS FROM CORE BUT NOT  
:FROM THE BCC.  
:



```

2004      : TEST 40
2005      :*****
2006 007470 012737 000040 001226 1ST40: MOV #40,TSTNO
2007 007476 012737 010002 001216 MOV #TST41,NEXT
2008 007504 004737 011024 JSR PC,SET.UP ;SET UP ALL NECESSARY FOR TEST.
2009 007510 012777 100010 171654 MOV #BIT15+BIT3,@DQSEC
2010      ;SET BCC START CLEAR AND SNGL CHAR
2011 007516 105377 171640 DECB @DQRC5H ;SEL CHAR ADD 16(8)
2012 007522 112777 000010 171640 MOVB #10,@DQREG ;SEL CHAR REG
2013 007530 012777 112400 171634 MOV #225*400,@DQSEC ;LOAD CHAR
2014 007536 112777 000014 171624 MOVB #14,@DQREG ;SEL THE SEQ REG
2015 007544 012777 100400 171620 MOV #BIT15+BIT8,@DQSEC
2016      ;SET SNGL CHAR AND CHAR STRIP.
2017 007552 012700 012014 MOV #TXBUFF,R0 ;GET TX POINTER
2018 007556 105020 1$: CLRB (R0)+ ;CLEAR IT OUT
2019 007560 022700 012215 CMP #TXBUFF+201,R0 ;ALL DONE?
2020 007564 001374 BNE 1$ ;BR IF NO
2021 007566 112737 000350 012015 MOVB #350,TXBUFF+1 ;LOAD CHAR
2022 007574 112737 000225 012017 MOVB #225,TXBUFF+3 ;SAME
2023 007602 112777 000017 171560 MOVB #17,@DQREG ;SEL POLY REG
2024 007610 012777 000200 171554 MOV #200,@DQSEC ;SET FOR LRC 8
2025 007616 012737 000010 013040 MOV #8,COUNT ;SET FOR 8 BIT CHAR
2026 007624 012737 000225 013036 MOV #225,CHAR ;SET EXPECTED BCC CHAR
2027 007632 112777 000003 171530 MOVB #3,@DQREG ;SEL THE TX WC PRI.
2028 007640 012777 177576 171524 MOV #-202,@DQSEC ;SET BIG
2029 007646 105077 171516 CLRB @DQREG ;SEL REC PRIMARY
2030 007652 012777 012216 171512 MOV #RXBUFF,@DQSEC ;SET WITH START ADRS
2031 007660 112777 000001 171502 MOVB #1,@DQREG ;SEL REC CHAR CNT
2032 007666 012777 177775 171476 MOV #-3,@DQSEC ;SET CHAR COUNT
2033 007674 112777 000004 171466 MOVB #4,@DQREG ;SEL REC SECONDARY
2034 007702 012777 012236 171462 MOV #RXBUFF+20,@DQSEC ;SET WITH REC ADRS
2035 007710 112777 000005 171452 MOVB #5,@DQREG ;SEL CHAR COUNT
2036 007716 012777 177577 171446 MOV #-201,@DQSEC ;SET CHAR COUNT
2037 007724 004737 011674 JSR PC,NEWENA ;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
2038 007730 013703 013040 MOV COUNT,R3 ;SAVE COUNT
2039 007734 000241 5$: CLC ;
2040 007736 106100 ROLB R0 ;SHIFT RX BCC IMAGE
2041 007740 005500 6$: ADC R0 ;SAVE CARRY
2042 007742 023700 013036 CMP CHAR,R0 ;GOOD BCC?
2043 007746 001403 BEQ 4$ ;BR IF YES
2044 007750 005303 DEC R3 ;ALL SHIFTS DONE?
2045 007752 001370 BNE 5$ ;BR IF NO
2046 007754 104013 HLT 13 ;RX BCC WRONG!!
2047 007756 012701 012216 4$: MOV #RXBUFF,R1 ;GET RX BUFFER
2048 007762 123721 013036 7$: CMPB CHAR,(R1)+ ;CHAR STRIPPED
2049 007766 001001 BNE +4 ;
2050 007770 104013 HLT 13 ;8 BIT CHAR NOT STRIPPED.
2051 007772 022701 012226 8$: CMP #RXBUFF+10,R1 ;ALL DONE?
2052 007776 001371 BNE 7$ ;NOT YET
2053 010000 104400 SCOPE
2054
2055
2056
2057 ;TEST OF BCC TEST/ APPEND
2058 ;TEST OF 1 BCC'S TESTED/APPENDED
2059

```

```

2060      ; TEST 41
2061      ;*****
2062 010002 012737 000041 001226 TST41: MOV #41,TSTNO
2063 010010 012737 010060 001216      MOV #TST42,NEXT
2064 010016 012737 100060 011022      MOV #BIT15+BIT5+BIT4,FUNCT.
2065 010024 012737 000000 013034      MOV #000,XPOLY ;SET EXTENDED POLY.
2066 010032 012737 000255 013032      MOV #255,POLY ;SET 00-15 POLY
2067 010040 004737 011454 ;GOTO SUBROUTINE
2068 010044 017737 171316 013030      MOV @DQERR,ERR ;IS THE AN ERROR CONDITION?
2069 010052 100001 ;BR IF NO ERRORS
2070 010054 104015 ;THE DQ11 ERROR FLAG IS SET..
2071 010056 104400 ;SCOPE THIS TEST
2072
2073      ;TEST OF BCC TEST/ APPEND
2074      ;TEST OF 2 BCC'S TESTED/APPENDED
2075
2076      ; TEST 42
2077      ;*****
2078 010060 012737 000042 001226 TST42: MOV #42,TSTNO
2079 010066 012737 010136 001216      MOV #TST43,NEXT
2080 010074 012737 100020 011022      MOV #BIT15+BIT4,FUNCT.
2081 010102 012737 000000 013034      MOV #000,XPOLY ;SET EXTENDED POLY.
2082 010110 012737 112001 013032      MOV #112001,POLY ;SET 00-15 POLY
2083 010116 004737 011454 ;GOTO SUBROUTINE
2084 010122 017737 171240 013030      MOV @DQERR,ERR ;IS THE AN ERROR CONDITION?
2085 010130 100001 ;BR IF NO ERRORS
2086 010132 104015 ;THE DQ11 ERROR FLAG IS SET..
2087 010134 104400 ;SCOPE THIS TEST
2088
2089      ;TEST OF BCC TEST/ APPEND
2090      ;TEST OF 3 BCC'S TESTED/APPENDED
2091
2092      ; TEST 43
2093      ;*****
2094 010136 012737 000043 001226 TST43: MOV #43,TSTNO
2095 010144 012737 010214 001216      MOV #TST44,NEXT
2096 010152 012737 100040 011022      MOV #BIT15+BIT5,FUNCT.
2097 010160 012737 000225 013034      MOV #225,XPOLY ;SET EXTENDED POLY.
2098 010166 012737 112001 013032      MOV #112001,POLY ;SET 00-15 POLY
2099 010174 004737 011454 ;GOTO SUBROUTINE
2100 010200 017737 171162 013030      MOV @DQERR,ERR ;IS THE AN ERROR CONDITION?
2101 010206 100001 ;BR IF NO ERRORS
2102 010210 104015 ;THE DQ11 ERROR FLAG IS SET..
2103 010212 104400 ;SCOPE THIS TEST

```

2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135 010214 012737 000044 001226  
2136 010222 012737 013050 001216  
2137 010230 004737 011024  
2138 010234 012777 170412 171130  
2139 010242 105377 171114  
2140 010246 112777 000010 171114  
2141 010254 012777 112400 171110  
2142 010262 112777 000014 171100  
2143 010270 012777 120644 171074  
2144 010276 105377 171060  
2145 010302 112777 000010 171060  
2146 010310 012777 010020 171054  
2147 010316 112777 000014 171044  
2148 010324 012777 101000 171040  
2149 010332 112777 000006 171030  
2150 010340 012777 012064 171024  
2151 010346 112777 000001 171014  
2152 010354 012777 177600 171010  
2153 010362 112777 000067 171000  
2154 010370 012777 177776 170774  
2155 010376 112777 000017 170764  
2156 010404 012777 172516 170760  
2157 010412 112777 000012 170750  
2158 010420 052777 000100 170744  
2159 010426 112777 000017 170734

```

: 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 ; SFL 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

```

2160	010434	012777	000275	170730	MOV	#275,@DQSEC	:
2161	010442	012700	012014		MOV	#TXBUFF,R0	:GET POINTER
2162	010446	012720			MOV	(PC)+,(R0)+	:LOAD THE BUFFER WITH DATA
2163	010450	000	350		.BYTE	000,350	:DATA
2164	010452	012720			MOV	(PC)+,(R0)+	:LOAD THE BUFFER WITH DATA
2165	010454	311	224		.BYTE	311,224	:DATA
2166	010456	012720			MOV	(PC)+,(R0)+	:LOAD THE BUFFER WITH DATA
2167	010460	107	201		.BYTE	107,201	:DATA
2168	010462	012720			MOV	(PC)+,(R0)+	:LOAD THE BUFFER WITH DATA
2169	010464	371	251		.BYTE	371,251	:DATA
2170	010466	012700	012064		MOV	#TXBUFF+50,R0	:LOAD R0
2171	010472	012720			MOV	(PC)+,(R0)+	:LOAD BUFFER WITH DATA
2172	010474	225	377		.BYTE	225,377	:DATA
2173	010476	004737	011620		JSR	PC,ENABLE	:SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
2174	010502	005777	170652		TST	@DQRCR	:DID RX CHAR FLAG SET?
2175	010506	100401			BMI	.+4	:BR IF YES
2176	010510	104014			HLT	14	:CHAR DET FLAG NOT SET.
2177	010512	017737	170650	013030	MOV	@DQERR,ERR	:ANY ERRORS?
2178	010520	100001			BPL	.+4	:BR IF NO ERRORS
2179	010522	104017			HLT	17	:DQ11 ERROR FLAG SET.
2180	010524	122777	000200	170626	CMPB	#200,@DQRCR	:DONE(P)-1; P/S=0; GO=0?
2181	010532	001401			BEQ	.+4	:YES
2182	010534	104014			HLT	14	:RX CSR WRONG DATA
2183	010536	122777	000300	170620	CMPB	#300,@DQRCR	:DONE(P)=1; P/S=0; GO 0?
2184	010544	001401			BEQ	.+4	:YES
2185	010546	104014			HLT	14	:TX CSR WRONG DATA
2186	010550	112777	000001	170612	MOVB	#1,@DQREG	:GET RX WC PRI.
2187	010556	005777	170610		TST	@DQSEC	:S/B NOT=0
2188	010562	001001			BNE	.+4	:BR IF OK
2189	010564	104014			HLT	14	:RX WC PRI S/B NON-ZERO!
2190	010566	112777	000003	170574	MOVB	#3,@DQREG	:SEL THE TX WC PRI.
2191	010574	005777	170572		TST	@DQSEC	:S/B =0
2192	010600	001401			BEQ	.+4	:
2193	010602	104014			HLT	14	:TX WC PRI S/B NON-ZERO
2194	010604	012700	012216		MOV	#RXBUFF,R0	:GET RX BUFFER POINTER
2195	010610	012702	000010		MOV	#0,R2	:SET CHAR COUNT
2196	010614	122710	000350	1\$:	CMPB	#350,(R0)	:ARE THE TWO CHARS STRIPPED?
2197	010620	001001			BNE	.+4	:
2198	010622	104014			HLT	14	:CHAR '350' NOT STRIPPED FROM CORE
2199	010624	122720	000225		CMPB	#225,(R0)+	:
2200	010630	001001			BNE	.+4	:
2201	010632	104014			HLT	14	:CHAR '225' NOT STRIPPED
2202	010634	122720	000020		CMPB	#20,(R0)+	:IS DLE STRIPPED?
2203	010640	001001			BNE	.+4	:
2204	010642	104021			HLT	21	:DLE STUCK -- REFER TO M7817 ECO
2205							:CHECK E84 ONE-SHOT
2206	010644	005302			DEC	R2	:ALL DONE?
2207	010646	001362			BNE	1\$	:NO
2208	010650	104400			SCOPE		:SCOPE THE TEST

```

2209
2210
2211 010652          .MEMCLR:
2212 010652 105077 170512      CLRB   @DQREG
2213 010656 012705 000020      MOV    #16.,R5
2214 010662 152777 000020 170500 1$:  BISB   #BIT4,@DQREG
2215 010670 142777 000140 170472  BICB   #140,@DQREG
2216 010676 005077 170470      CLR    @DQSEC
2217 010702 105277 170462      INCB   @DQREG
2218 010706 005305              DEC    R5
2219 010710 001364              BNE    1$
2220 010712 105077 170452      CLRB   @DQREG
2221 010716 105077 170440      CLRB   @DQRC5H
2222 010722 012705 000020      MOV    #16.,R5
2223 010726 112777 000010 170434 2$:  MOVB   #10,@DQREG
2224 010734 005077 170432      CLR    @DQSEC
2225 010740 112777 000014 170422  MOVB   #14,@DQREG
2226 010746 005077 170420      CLR    @DQSEC
2227 010752 105277 170404      INCB   @DQRC5H
2228 010756 005305              DEC    R5
2229 010760 001362              BNE    2$
2230 010762 105077 170374      CLRB   @DQRC5H
2231 010766          .MSTCLR:
2232 010766 112777 000012 170374  MOVB   #MISC.,@DQREG
2233 010774 012777 000040 170370  MOV    #BIT5,@DQSEC
2234 011002 000240              NOP
2235 011004 112777 000012 170356  MOVB   #MISC.,@DQREG
2236 011012 012777 000040 170352  MOV    #BIT5,@DQSEC
2237 011020 000002              RTI
2238
2239 011022 000000          FUNCT.: 0
2240 011024          SET.UP:
2241 011024 104413          MEMCLR
2242 011026 012702 000010      MOV    #10,R2
2243 011032 012700 012014      MOV    #TXBUFF,R0
2244 011036 105020          CLRB   (R0)+
2245 011040 005302          DEC    R2
2246 011042 001375          BNE    .-4
2247 011044 105077 170320      CLRB   @DQREG
2248 011050 012777 012216 170314  MOV    #RXBUFF,@DQSEC
2249 011056 105277 170306      INCB   @DQREG
2250 011062 012777 177770 170302  MOV    #-10,@DQSEC
2251 011070 105277 170274      INCB   @DQREG
2252 011074 005737 001510      TST    DQSTAT
2253 011100 100404          BMI    10$
2254 011102 112737 000377 012012  MOVB   #377,SYNC
2255 011110 000403          BR     20$
2256 011112 112737 000026 012012 10$:  MOVB   #26,SYNC
2257 011120 012777 012012 170244 20$:  MOV    #SYNC,@DQSEC
2258 011126 105277 170236      INCB   @DQREG
2259 011132 012777 177766 170232  MOV    #-12,@DQSEC
2260 011140 112777 000011 170222  MOVB   #11,@DQREG
2261 011146 013777 012010 170216  MOV    .SYNC,@DQSEC
2262 011154 105277 170210      INCB   @DQREG
2263 011160 012777 004000 170204  MOV    #4000,@DQSEC
2264 011166 032737 040000 001510  BIT    #JUMBIT,DQSTAT
  
```

```

;CLEAR ALL THE DQ11 REGISTERS
;PREPARE TO CLEAR THE TX BUFFER
;GET THE BUFFERS ADDRESS
;START CLEARING
;ALL CLEAR?
;BR IF NOT DONE.
;SELECT THE RX BA PRI.
;LOAD RX BA PRI.
;GET RX WC PRI.
;SET FOR 8. CHARS.
;GET THE TX BA PRI.
;ONE SYNC CHARACTER?
;IF NO, BR.
;SET MARK.
;CONTINUE.
;RESTORE SYNC CHARACTER.
;SET ADDRESS
;SEL THE TX WC PRI.
;SET FOR 8. CHARS AND 2 SYNC'S
;SEL THE SYNC REGISTER
;LOAD WITH SYNC
;SELECT THE MISC REGISTER
;SET FOR EIGHT BITS.
;IF TEST JUMPER AT END OF CABLE;
  
```

```

:♦♦E
:♦♦E
:♦♦E
:♦♦E
:♦♦E
:♦♦E
  
```

2265	011174	001003				BNE	+.10		:RUN DATA THROUGH IT.
2266	011176	052777	000010	170166		BIS	#BIT3,@DQSEC		:NO TEST JUMPER; SET TEST LOOP
2267	011204	112777	000017	170150		MOVB	#17,@DQRCSH		:GET LAST CHAR DET ADDRESS.
2268	011212	112777	000010	170150		MOVB	#10,@DQREG		:SEL CHAR DET REGISTER
2269	011220	012777	164000	170144		MOV	#350*400,@DQSEC		:LOAD THE CHARACTER
2270	011226	112777	000014	170134		MOVB	#14,@DQREG		:GET THE SEQ REGISTER
2271	011234	000207				RTS	PC		:LEAVE ROUTINE POINTING TO SEQ REGISTER.
2272	011236							EXT.UP:	
2273	011236	104413				MEMCLR			:CLEAR ALL THE DQ11
2274	011240	012702	000020			MOV	#20,R2		:PREPARE TO CLEAR THE TX BUFFER
2275	011244	012700	012424			MOV	#XTXBUF,RO		:GET THE TX BUFFER ADDRESS.
2276	011250	005020				CLR	(R0)+		:START CLEARING
2277	011252	005302				DEC	R2		:DONE?
2278	011254	001375				BNE	.-4		:BR IF NO
2279	011256	005737	001510			TSI	DQSTAT		:ONE SYNC CHARACTER? ;:++E
2280	011262	100404				BMI	10\$		:IF NO, BR. ;:++E
2281	011264	012737	177777	012420		MOV	#-1, XSYNC		:SET MARK. ;:++E
2282	011272	000403				BR	20\$		:CONTINUE. ;:++E
2283	011274	012737	013026	012420	10\$:	MOV	#13026, XSYNC		:LOAD SYNC'S ;:++E
2284	011302	012737	013026	012422	20\$:	MOV	#13026, XSYNC2		:DITTO ;:++E
2285	011310	105077	170054			CLRB	@DQREG		:SEL THE RX BA PRI.
2286	011314	012777	012216	170050		MOV	#RXBUFF,@DQSEC		:LOAD THE ADDRESS
2287	011322	105277	170042			INCB	@DQREG		:SEL THE RX WC PRI.
2288	011326	012777	177770	170036		MOV	#-10,@DQSEC		:SET FOR TEN CHARS
2289	011334	105277	170030			INCB	@DQREG		:SEL THE TX BA PRI.
2290	011340	012777	012420	170024		MOV	#XSYNC,@DQSEC		:LOAD THE ADDRESS.
2291	011346	105277	170016			INCB	@DQREG		:SEL THE TX WC PRI.
2292	011352	012777	177766	170012		MOV	#-12,@DQSEC		:SET FOR TWO SYNC AND 8. CHARS
2293	011360	112777	000011	170002		MOVB	#11,@DQREG		:SEL THE SYNC REGISTER
2294	011366	013777	012010	167776		MOV	.SYNC,@DQSEC		:LOAD SYNC
2295	011374	105277	167770			INCB	@DQREG		:SEL THE MISC REGISTER.
2296	011400	005077	167766			CLR	@DQSEC		:SEL 16 BITS PER CHAR.
2297	011404	032737	040000	001510		BIT	#JUMBIT,DQSTAT		:IF TEST JUMPER INSTALLED;
2298	011412	001003				BNE	+.10		:RUN DATA THROUGH CABLE.
2299	011414	052777	000010	167750		BIS	#BIT3,@DQSEC		:NO JUMPER; SET TEST LOOP.
2300	011422	112777	000017	167732		MOVB	#17,@DQRCSH		:GET LAST CHAR DET ADDRESS
2301	011430	112777	000010	167732		MOVB	#10,@DQREG		:GET CHAR DET REGISTER
2302	011436	012777	003721	167726		MOV	#3721,@DQSEC		:LOAD CHARACTER
2303	011444	112777	000014	167716		MOVB	#14,@DQREG		:SEL THE SEQ REGISTER
2304	011452	000207				RTS	PC		:LEAVE THE ROUTINE.
2305									
2306	011454							BCC.TA:	
2307	011454	004737	011024			JSR	PC,SET.UP		:SET UP ALL NECESSARY FOR TEST.
2308	011460	012777	100010	167704		MOV	#BIT15+BIT3,@DQSEC		
2309									:SET SNGL CHAR AND BCC START CLEAR
2310	011466	105377	167670			DECB	@DQRCSH		:GET NEXT ADDR
2311	011472	112777	000010	167670		MOVB	#10,@DQREG		:SEL CHAR DET ADDR
2312	011500	012777	176400	167664		MOV	#375*400,@DQSEC		:LOAD CHAR
2313	011506	112777	000014	167654		MOVB	#14,@DQREG		:SEL THE SEQ REG.
2314	011514	013777	011022	167650		MOV	FUNCT.,@DQSEC		:SET THE TEST APPEND FUNCTIONS
2315	011522	112777	000017	167640		MOVB	#17,@DQREG		:SEL THE POLY REG.
2316	011530	013777	013032	167634		MOV	POLY,@DQSEC		:LOAD THE POLYNOMIAL.
2317	011536	112777	000012	167624		MOVB	#12,@DQREG		:SEL THE MISC REG
2318	011544	052777	000100	167620		BIS	#BIT6,@DQSEC		:SEL EXT POLY REG
2319	011552	112777	000017	167610		MOVB	#17,@DQREG		:RESEL. THE POLY REG
2320	011560	013777	013034	167604		MOV	XPOLY,@DQSEC		:SET 16-23 POLY

```
2321 011566 012700 012014      MOV      #TXBJFF,R0      ;SET TX POINTER
2322 011572 012720      MOV      (PC)+,(R0)+    ;LOAD THE BUFFER WITH DATA
2323 011574      350      355      .BYTE   350,355        ;DATA
2324 011576 012720      MOV      (PC)+,(R0)+    ;LOAD THE BUFFER WITH DATA
2325 011600      360      365      .BYTE   360,365        ;DATA
2326 011602 012720      MOV      (PC)+,(R0)+    ;LOAD THE BUFFER WITH DATA
2327 011604      370      375      .BYTE   370,375        ;DATA
2328 011606 012720      MOV      (PC)+,(R0)+    ;LOAD THE BUFFER WITH DATA
2329 011610      377      377      .BYTE   377,377        ;DATA
2330 011612 004737 011620      JSR      PC,ENABLE      ;SET GO BITS FOR RX AND TX; WAIT FOR RX DONE.
2331 011616 000207      RTS      PC            ;LEAVE
2332
2333 011620 005037 011670      ENABLE: CLR      3$      ;
2334 011624 012737 000005 011672      MOV      #5,4$        ;SET DELAY
2335 011632 005277 167522      INC      @DQRCR        ;SET RX GO.
2336 011636 005277 167522      INC      @DQTCR        ;SET TX GO.
2337 011642 105777 167512      1$:     TSTB      @DQRCR    ;RX PRI. DONE?
2338 011646 100407      BMI      2$          ;BR IF YES
2339 011650 005237 011670      INC      5$          ;DELAY.....
2340 011654 001372      BNE      1$          ;
2341 011656 005337 011672      DEC      4$          ;.....
2342 011662 001367      BNE      1$          ;
2343 011664 104001      HLT      1           ;RX PRI DONE NOT SET.
2344 011666 000207      2$:     RTS      PC            ;LEAVE
2345 011670 000000      3$:     000
2346 011672 000000      4$:     000
2347
2348 011674 005037 012004      NEWENA: CLR      3$      ;
2349 011700 012737 000005 012006      MOV      #5,4$        ;SET DELAY
2350 011706 005277 167446      INC      @DQRCR        ;SET RX GO
2351 011712 005277 167446      INC      @DQTCR        ;SET TX GO
2352 011716 105777 167436      1$:     TSTB      @DQRCR    ;RX PRIMARY DONE?
2353 011722 100410      BMI      2$          ;BR IF YES
2354 011724 005237 012004      INC      3$          ;DELAY
2355 011730 001372      BNE      1$          ;
2356 011732 005337 012006      DEC      4$          ;
2357 011736 001367      BNE      1$          ;
2358 011740 104001      HLT      1           ;RX PRI DONE FAILED TO SET
2359 011742 000417      BR       5$          ;LEAVE
2360 011744 112777 000012 167416 2$:     MOVB     #12,@DQREG    ;SEL THE MISC REG
2361 011752 042777 000010 167412      BIC      #BIT3,@DQSEC  ;STOP DATA
2362 011760 112777 000015 167402      MOVB     #15,@DQREG    ;SEL RX BCC
2363 011766 017700 167400      MOV      @DQSEC,R0     ;READ INTO R0
2364 011772 105277 167372      INCB     @DQREG        ;SEL TX BCC
2365 011776 017701 167370      MOV      @DQSEC,R1     ;READ INTO R1
2366 012002 000207      5$:     RTS      PC            ;RETURN
2367 012004 000000      3$:     .WORD 0
2368 012006 000000      4$:     .WORD 0
2369
2370
2371 012010      026      026      .SYNC:  .BYTE   26,26
2372 012012      026      026      SYNC:   .BYTE   26,26
2373 012014 000000      TXBUFF: 0
2374      012216      .=.+200
2375 012216 000000      RXBUFF: 0
2376      012420      .=.+200
```

2377	012420	026	026	XSYNC:	.BYTE 26,26	
2378	012422	026	026	XSYNC2:	.BYTE 26,26	
2379	012424	000000		XIXBUF:	000000	
2380		012626			.=.+200	
2381	012626	000000		XRIBUF:	000000	
2382		013030			.=.+200	
2383	013030	000000		ERR:	0	
2384	013032	000000		POLY:	0	
2385	013034	000000		XPOLY:	0	
2386	013036	000000		CHAR:	0	
2387	013040	000000		COUNT:	0	
2388	013042	000000		ADDR:	0	
2389	013044	000000		GDCHAR:	0	
2390	013046	000000		DETCAR:	0	
2391						
2392						
2393						
2394						
2395						
2396						
2397						
2398	013050	005037	001234	.EOP:	CLR LSTERR	;CLEAR LAST ERROR PC
2399	013054	005037	001312		CLR ERRFLG	;CLEAR ERROR FLAG
2400	013060	005237	001230		INC PASCNT	;UPDATE PASS COUNT
2401	013064	104402			TYPE	
2402	013066	015300			MEPASS	
2403	013070	104402			TYPE	
2404	013072	015461			MCSRX	
2405	013074	104411			CNVRT	
2406	013076	013206			XCSR	
2407	013100	104402			TYPE	
2408	013102	015467			MVECX	
2409	013104	104411			CNVRT	
2410	013106	013214			XVEC	
2411	013110	104402			TYPE	
2412	013112	015475			MPASSX	
2413	013114	104411			CNVRT	
2414	013116	013222			XPASS	
2415	013120	104402			TYPE	
2416	013122	015506			MERRX	
2417	013124	104411			CNVRT	
2418	013126	013230			XERR	
2419	013130	013777	001230 166044		MOV PASCNT,@LIGHTS	;DISPLAY PASS COUNT
2420	013136	005337	001276		DEC SAVNUM	
2421	013142	001013			BNE RESTR	
2422	013144	013737	001504 001276		MOV DQNUM,SAVNUM	
2423	013152	013701	000042		MOV @42,R1	;CHECK FOR ACT-11 OR DDP
2424	013156	001405			BEG RESTR	;IF NOT, CONTINUE TESTING
2425	013160	000005			RESET	
2426	013162			LOGICAL:		
2427	013162	004711			JSR PC,(R1)	
2428	013164	000240			NOP	
2429	013166	000240			NOP	
2430	013170	000240			NOP	
2431	013172	104414		RESTR:	CKSWR	
2432	013174	012737	002254 001214		MOV #TST1,RETURN	



2433	013202	000137	002254						JMP	TS11
2434	013206	000001				XCSR:	1			
2435	013210	006	002					.BYTE	6.2	
2436	013212	001360						DQRCSR		
2437	013214	000001				XVEC:	1			
2438	013216	003	002					.BYTE	3.2	
2439	013220	001350						DQRVEC		
2440	013222	000001				XPASS:	1			
2441	013224	006	002					.BYTE	6.2	
2442	013226	001230						PASCNT		
2443	013230	000001				XERR:	1			
2444	013232	006	002					.BYTE	6.2	
2445	013234	001232						ERRCNT		
2446										
2447										:SCOPE LOOP AND INTERATION HANDLER
2448										
2449	013236	104414				.SCOPE:	CKSWR			
2450	013240	032777	040000	165732		TTST:	BIT	#BIT14,@SWR		
2451	013246	001407					BEQ	1\$		
2452	013250	000432					BR	3\$		
2453	013252	105777	165726				TSTB	@TKCSR		
2454	013256	100027					BPL	3\$		
2455	013260	017700	165722				MOV	@TKDBR,R0		
2456	013264	000412					BR	2\$		
2457	013266	032777	004000	165704		1\$:	BIT	#SW11,@SWR		
2458	013274	001006					BNE	2\$		
2459	013276	005237	001224				INC	LPCNT		
2460	013302	023737	001224	001222			CMP	LPCNT,ICOUNT		
2461	013310	001012					BNE	3\$		
2462	013312	105037	001312			2\$:	CLRB	ERRFLG		
2463	013316	005037	001224				CLR	LPCNT		
2464	013322	012737	000113	001222			MOV	#75.,ICOUNT		
2465	013330	013737	001216	001214			MOV	NEXT,RETURN		
2466	013336	013716	001214			3\$:	MOV	RETURN,(SP)		
2467	013342	000002					RTI			
2468	013344	001407				BRW:	1407			
2469	013346	000432				BRX:	432			
2470										
2471										:CHECK FOR FREEZE ON CURRENT DATA
2472										
2473	013350	104414				.SCOPE1:	CKSWR			
2474	013352	032777	001000	165620			BIT	#SW09,@SWR		
2475	013360	001402					BEQ	1\$		
2476	013362	013716	001220				MOV	LOCK,(SP)		
2477	013366	000002				1\$:	RTI			
2478										
2479										:TELETYPE OUTPUT ROUTINE
2480										
2481	013370	010546				.TYPE:	MOV	R5,-(SP)		
2482	013372	017605	000002				MOV	@2(SP),R5		
2483	013376	062766	000002	000002			ADD	#2,2(SP)		
2484	013404	005737	015060			1\$:	TST	@#RDSW		
2485	013410	001004					BNE	300\$		
2486	013412	032777	010000	165560			BIT	#SW12,@SWR		
2487	013420	001024					BNE	3\$		
2488	013422	105715				300\$:	TSTB	(R5)		

2489	013424	100014			BPL	2\$
2490	013426	105777	165556		TSTB	@TPCSR
2491	013432	100375			BPL	.-4
2492	013434	012777	000015	165550	MOV	#15,@TPDBR
2493	013442	105777	165542		TSTB	@TPCSR
2494	013446	100375			BPL	.-4
2495	013450	012777	000012	165534	MOV	#12,@TPDBR
2496	013456	105777	165526	2\$:	TSTB	@TPCSR
2497	013462	100375			BPL	2\$
2498	013464	112577	165522		MOVB	(R5)+,@TPDBR
2499	013470	001345			BNE	1\$
2500	013472	012605		3\$:	MOV	(SP)+,R5
2501	013474	000002			RTI	

:ASCII STRING INPUT ROUTINE

2505	013476	010346			.INSTR:	MOV	R3,-(SP)
2506	013500	010446				MOV	R4,-(SP)
2507	013502	017637	000004	013520		MOV	@4(SP),.MSG
2508	013510	062766	000002	000004		ADD	#2,4(SP)
2509	013516	104402			.INST1:	TYPE	
2510	013520	000000			.MSG:		0
2511	013522	012704	015652			MOV	#INBUF,R4
2512	013526	012703	000007			MOV	#7,R3
2513	013532	105777	165446	1\$:	TSTB	@TKCSR	
2514	013536	100375			BPL	1\$	
2515	013540	117714	165442		MOVB	@TKDBR,(R4)	
2516	013544	142714	000200		BICB	#200,(R4)	
2517	013550	121427	000025		CMPB	(R4),#25	
2518	013554	001003			BNE	200\$	
2519	013556	104402	015240		TYPE	,MCRLF	
2520	013562	000755			BR	.INST1	
2521	013564	122427	000015	200\$:	CMPB	(R4)+,#15	
2522	013570	001423			BEQ	INSTR2	
2523	013572	117777	165410	165412	MOVB	@TKDBR,@TPDBR	
2524	013600	105777	165404	2\$:	TSTB	@TPCSR	
2525	013604	100375			BPL	2\$	
2526	013606	005303			DEC	R3	
2527	013610	001350			BNE	1\$	
2528	013612	000402			BR	.INSTG	
2529	013614	010346			.INSTE:	MOV	R3,-(SP)
2530	013616	010446				MOV	R4,-(SP)
2531	013620	104402			.INSTG:	TYPE	
2532	013622	015234			MGM		
2533	013624	005737	015060		TST	@WRDSW	
2534	013630	001402			BEQ	400\$	
2535	013632	104402	015240		TYPE	,MCRLF	
2536	013636	000727		400\$:	BR	.INST1	
2537	013640	012604		INSTR2:	MOV	(SP)+,R4	
2538	013642	012603			MOV	(SP)+,R3	
2539	013644	000002			RTI		

:IS IT <^G>

:CONVERT ASCII STRING TO OCTAL

2543	013646	010546			.PARAM:	MOV	R5,-(SP)
2544	013650	010446				MOV	R4,-(SP)

```

2545 013652 016605 000004      MOV      4(SP),R5
2546 013656 012537 014052      MOV      (R5)+,LOLIM
2547 013662 012537 014054      MOV      (R5)+,HILIM
2548 013666 012537 014056      MOV      (R5)+,DEVADR
2549 013672 112537 014060      MOV      (R5)+,LOBITS
2550 013676 112537 014061      MOV      (R5)+,ADRCNT
2551 013702 010566 000004      MOV      R5,4(SP)
2552 013706 005005      PARAM1: CLR      R5
2553 013710 012704 015652      MOV      #INBUF,R4
2554 013714 122714 000015      CMPB     #15,(R4)
2555 013720 001420      BEQ      PARERR
2556 013722 121427 000060      1$:     CMPB     (R4),#60
2557 013726 002415      BLT      PARERR
2558 013730 121427 000067      CMPB     (R4),#67
2559 013734 003012      BGT      PARERR
2560 013736 142714 000060      BICB     #60,(R4)
2561 013742 152405      BISB     (R4)+,R5
2562 013744 122714 000015      CMPB     #15,(R4)
2563 013750 001414      BEQ      LIMITS
2564 013752 006305      ASL      R5
2565 013754 006305      ASL      R5
2566 013756 006305      ASL      R5
2567 013760 000760      BR       1$
2568 013762 122714 000015      PARERR: CMPB     #15,(R4)      ;IS FIRST CHARACTER A <CR>
2569 013766 001003      BNE     120$
2570 013770 005737 015060      TST     @RDSW      ;IS CKSWR ROUTINE BEING USED
2571 013774 001023      BNE     PARTI
2572 013776 104404      120$:   INSTER
2573 014000 000742      BR       PARAM1
2574
2575      ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2576
2577 014002 020537 014054      LIMITS: CMP      R5,HILIM
2578 014006 101365      BHI     PARERR
2579 014010 020537 014052      CMP      R5,LOLIM
2580 014014 103762      BLO     PARERR
2581 014016 133705 014060      BITB     LOBITS,R5
2582 014022 001357      BNE     PARERR
2583
2584      ;STORE NUMBER AT SPECIFIED ADDRESS
2585
2586 014024 013704 014056      1$:     MOV      DEVADR,R4
2587 014030 010524      MOV      R5,(R4)+
2588 014032 062705 000002      ADD      #2,R5
2589 014036 105337 014061      DECB     ADRCNT
2590 014042 001372      BNE     1$
2591 014044 012604      PARTI:  MOV      (SP)+,R4
2592 014046 012605      MOV      (SP)+,R5
2593 014050 000002      RTI
2594 014052 000000      LOLIM:  0
2595 014054 000000      HILIM:  0
2596 014056 000000      DEVADR: 0
2597 014060 000000      LOBITS: 0
2598      ADRCNT=LOBITS+1
2599
2600      ;SAVE PC OF TEST THAT FAILED AND R0-R5

```

```

2601
2602 014062 016637 000004 001274 .SAV05: MOV 4(SP),SAVPC
2603
2604 ;SAVE RC-R5
2605
2606 014070 010537 001270 SV05: MOV R5,SAVR5
2607 014074 010437 001266 MOV R4,SAVR4
2608 014100 010337 001264 MOV R3,SAVR3
2609 014104 010237 001262 MOV R2,SAVR2
2610 014110 010137 001260 MOV R1,SAVR1
2611 014114 010037 001256 MOV R0,SAVR0
2612 014120 000002 RTI
2613
2614 ;RESTORE R0-R5
2615
2616 014122 013700 001256 .RES05: MOV SAVR0,R0
2617 014126 013701 001260 MOV SAVR1,R1
2618 014132 013702 001262 MOV SAVR2,R2
2619 014136 013703 001264 MOV SAVR3,R3
2620 014142 013704 001266 MOV SAVR4,R4
2621 014146 013705 001270 MOV SAVR5,R5
2622 014152 000002 RTI
2623
2624 ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
2625
2626 014154 104402 .CONVR: TYPE
2627 014156 015240 MCRLF
2628 014160 010046 .CNVRT: MOV R0,-(SP)
2629 014162 010146 MOV R1,-(SP)
2630 014164 010346 MOV R3,-(SP)
2631 014166 010446 MOV R4,-(SP)
2632 014170 010546 MOV R5,-(SP)
2633 014172 017601 000012 MOV @12(SP),R1
2634 014176 013737 015714 001250 MOV TEMP,TEMP3
2635 014204 062766 000002 000012 ADD #2,12(SP)
2636 014212 012137 014374 MOV (R1)+,WRDCNT
2637 014216 112137 014376 1$: MOV (R1)+,CHRCNT
2638 014222 112137 014377 MOV (R1)+,SPACNT
2639 014226 013137 014400 MOV @ (R1)+,BINWRD
2640 014232 013704 014400 2$: MOV BINWRD,R4
2641 014236 113705 014376 MOVB CHRCNT,R5
2642 014242 012700 015714 MOV #TEMP,R0
2643 014246 010403 3$: MOV R4,R3
2644 014250 042703 177770 BIC #177770,R3
2645 014254 062703 000060 ADD #060,R3
2646 014260 110320 MOV R3,(R0)+
2647 014262 000241 CLC
2648 014264 006004 ROR R4
2649 014266 000241 CLC
2650 014270 006004 ROR R4
2651 014272 000241 CLC
2652 014274 006004 ROR R4
2653 014276 005305 DEC R5
2654 014300 001362 BNE 3$
2655 014302 012703 015756 MOV #MMDATA,R3
2656 014306 114023 4$: MOVB -(R0),(R3)+

```

2657	014310	105337	014376		DECB	CHRCNT	
2658	014314	001374			BNE	4\$	
2659	014316	105737	014377		TSTB	SPACNT	
2660	014322	001405			BEQ	6\$	
2661	014324	112723	000040	5\$:	MOVB	#040,(R3)+	
2662	014330	105337	014377		DECB	SPACNT	
2663	014334	001373			BNE	5\$	
2664	014336	105013		6\$:	CLRB	(R3)	
2665	014340	104402			TYPE		
2666	014342	015756			MDATA		
2667	014344	005337	014374		DEC	WRDCNT	
2668	014350	001322			BNE	1\$	
2669	014352	013737	001250	015714	MOV	TEMP3,TEMP	
2670	014360	012605			MOV	(SP)+,R5	
2671	014362	012604			MOV	(SP)+,R4	
2672	014364	012603			MOV	(SP)+,R3	
2673	014366	012601			MOV	(SP)+,R1	
2674	014370	012600			MOV	(SP)+,R0	
2675	014372	000002			RTI		
2676	014374	000000			WRDCNT:	0	
2677	014376	000000			CHRCNT:	0	
2678		014377			SPACNT=	CHRCNT+1	
2679	014400	000000			BINWRD:	0	
2680						:TRAP DISPATCH SERVICE	
2681						:ARGUMENT OF TRAP IS EXTRACTED	
2682						:AND USED AS OFFSET TO OBTAIN POINTER	
2683						:TO SELECTED SUBROUTINE	
2684							
2685	014402	011646			.TRPSR:	MOV (SP),-(SP)	:GET PC OF RETURN
2686	014404	162716	000002		SUB	#2,(SP)	:PC OF TRAP
2687	014410	017616	000000		MOV	@(SP),(SP)	:GET TRP
2688	014414	006316			TRPOK:	ASL (SP)	:MULTIPLY TRAP ARG BY 2
2689	014416	042716	177001		BIC	#177001,(SP)	:CLEAR UNWANTED BITS
2690	014422	062716	001314		ADD	#.TRPTAB,(SP)	:POINTER TO SUBROUTINE ADDRESS
2691	014426	017616	000000		MOV	@(SP),(SP)	:SUBROUTINE ADDRESS
2692	014432	000136			JMP	@(SP)+	:GO TO SUBROUTINE
2693							
2694						:ERROR HANDLER	
2695							
2696	014434	104414			.HLT:	CKSWR	
2697	014436	032777	010000	164534	BIT	#SW12,@SWR	
2698	014444	001406			BEQ	XBX	
2699	014446	105777	164536		TSTB	@TPCSR	
2700	014452	100003			BPL	XBX	
2701	014454	112777	000207	164530	MOVB	#207,@TPDBR	
2702	014462	032777	020000	164510	XBX:	BIT #SW13,@SWR	
2703	014470	001074			BNE	HALTS	
2704	014472	021637	001234		CMR	(SP),LSTERR	
2705	014476	001404			BEQ	1\$	
2706	014500	011637	001234		MOV	(SP),LSTERR	
2707	014504	105037	001312		CLRB	ERRFLG	
2708	014510	104406			1\$:	SAV05	
2709	014512	011605			MOV	(SP),R5	
2710	014514	162705	000002		SUB	#2,R5	
2711	014520	011504			MOV	(R5),R4	
2712	014522	006304			ASL	R4	

2713	014524	061504			ADD	(R5),R4
2714	014526	006304			ASL	R4
2715	014530	042704	177001		BIC	#177001,R4
2716	014534	062704	016020		ADD	#.ERRTAB,R4
2717	014540	012437	014632		MOV	(R4)+,ERRMSG
2718	014544	012437	014644		MOV	(R4)+,DATAHD
2719	014550	011437	014656		MOV	(R4),DATABP
2720	014554	105737	001312		TSTB	ERRFLG
2721	014560	001403			BEQ	TYPMSG
2722	014562	005737	014656		TST	DATABP
2723	014566	001027			BNE	TYPDAT
2724	014570	104402			TYPMSG:	TYPE
2725	014572	015517			MTSTN	
2726	014574	104411			CNVRT	
2727	014576	014756			XTSTN	
2728	014600	104402			TYPE	
2729	014602	015605			MERRPC	
2730	014604	104411			CNVRT	
2731	014606	014750			ERTABO	
2732	014610	104402			TYPE	
2733	014612	015240			MCRLF	
2734	014614	112737	177777	001312	MOVB	#-1,ERRFLG
2735	014622	005737	014632		TST	ERRMSG
2736	014626	001402			BEQ	WRKO.FM
2737	014630	104402			TYPE	
2738	014632	000000			ERRMSG:	0
2739	014634				WRKO.FM:	
2740	014634	005737	014644		TST	DATABP
2741	014640	001402			BEQ	TYPDAT
2742	014642	104402			TYPE	
2743	014644	000000			DATABP:	0
2744	014646	005737	014656		TYPDAT:	TST
2745	014652	001402			BEQ	DATABP
2746	014654	104410			CONVRT	RESREG
2747	014656	000000			DATABP:	0
2748	014660	104407			RESREG:	RES05
2749	014662	005777	164312		HALTS:	TST
2750	014666	100005			BPL	@SWR
2751	014670	010046			PUSHRO	EXITER
2752	014672	016600	000002		MOV	2(SP),R0
2753	014676	000000			HALT	
2754	014700	012600			POPPO	
2755	014702	104414			EXITER:	CKSWR
2756	014704	005237	001232		INC	ERRCNT
2757	014710	032777	000400	164262	BIT	#SW08,@SWR
2758	014716	001007			BNE	1\$
2759	014720	032777	002000	164252	BIT	#SW10,@SWR
2760	014726	001407			BEQ	2\$
2761	014730	013737	001216	001214	MOV	NEXT,RETURN
2762	014736	012706	001200		1\$:	MOV
2763	014742	000177	164246		JMP	#STACK,SP
2764	014746	000002			2\$:	RTI
2765	014750	000001			ERTABO:	1
2766	014752	006	002		.BYTE	6,2
2767	014754	001274			SAVPC	
2768	014756	000001			XTSTN:	1

```

2769 014760 003 002 .BYTE 3,2
2770 014762 001226 TSTNO
2771 ;ENTER HERE ON POWER FAILURE
2772
2773
2774 014764 .PFAIL:
2775 014764 012737 014776 000024 MOV #RESTART,24 ;SET UP FOR POWER UP TRAP
2776 014772 000000 HALT ;HALT ON POWER DOWN NORMAL
2777 014774 000777 BR .
2778
2779 ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
2780
2781 014776 RESTAR:
2782 014776 012737 014764 000024 MOV #.PFAIL,24 ;SET UP FOR POWER FAILURE
2783 015004 012706 001200 MOV #STACK,SP
2784 015010 005037 015714 CLR TEMP
2785 015014 005237 015714 INC TEMP
2786 015020 001375 BNE .-4
2787 015022 104402 TYPE
2788 015024 015242 MPFAIL
2789 015026 104411 CNVRT
2790 015030 015052 PFTAB
2791 015032 005037 001312 CLR ERRFLG
2792 015036 005037 001234 CLR LSTERR
2793 015042 104412 MSTCLR
2794 015044 104413 MEMCLR
2795 015046 000177 164142 JMP @RETURN
2796 015052 000001 PFTAB: 1
2797 015054 003 002 .BYTE 3,2
2798 015056 001226 TSTNO
2799
2800
2801 ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ^G TO ALLOW CHANGING
2802 ;OF LOC.176.
2803 ;LOCATIONS USED:
2804 015060 000000 RDSW: .WORD 0
2805
2806
2807 015062 005737 000042 .CKSWR: TST @#42
2808 015066 001042 BNE OUT
2809 015070 022737 000176 001200 CMP #SWREG,SWR ;SOFTWARE SWITCH REGISTER PRESENT
2810 015076 001036 BNE OUT ;NO, GET OUT
2811 015100 105777 164100 TSTB @TKCSR ;YES, WAIT FOR
2812 015104 100033 BPL OUT ;READY, GET CHARACTER
2813 015106 017737 164074 013520 MOV @TKDBR,.MSG ;AND STRIP OFF
2814 015114 042737 177600 013520 BIC #177600,.MSG ;THE GARBAGE
2815 015122 122737 000007 013520 CMPB #7,.MSG ;IS IT A <^G>
2816 015130 001021 BNE OUT
2817 015132 104402 015210 TYPE,$CNTG
2818 015136 005137 015060 .CNTLU: COM @WRDSW
2819 015142 104402 015214 TYPE,$MSWR
2820 015146 104411 015202 CNVRT,$WREGC
2821 015152 104403 015223 INSTR,$MNEW
2822 015156 104405 PARAM
2823 015160 000000 0
2824 015162 177777 177777

```

2825	015164	000176				SWREG	
2826	015166	000	001			.BYTE	0,1
2827	015170	104402	015240			TYPE,MCRLF	
2828	015174	005037	015060			OUT:	CLR @RDSW
2829	015200	000002				RTI	
2830	015202	000001				SWREGC:	1
2831	015204	006	002			.BYTE	6,2
2832	015206	000176				SWREG	
2833	015210	057377	000107			\$CNTG:	.ASCIZ <377>/^G/
2834	015214	051777	051127	020075		\$MSWR:	.ASCIZ <377>/SWR= /
2835	015222	000					
2836	015223	040	047040	053505		\$MNEW:	.ASCIZ / NEW= /
2837	015230	020075	000				
2838		015234				.EVEN	
2839	015234	020040	000077			MQM:	.ASCIZ / ?/
2840	015240	000377				MCRLF:	.ASCIZ <377>
2841	015242	050377	051127	043040		MPFAIL:	.ASCIZ <377>/PWR FAILED. RESTART AT TEST /
2842	015250	044501	042514	027104			
2843	015256	051040	051505	040524			
2844	015264	052122	040440	020124			
2845	015272	042524	052123	000040			
2846	015300	042777	042116	050040		MEPASS:	.ASCIZ <377>/END PASS DZDQF /
2847	015306	051501	020123	055104			
2848	015314	050504	020106	000040			
2849	015322	051377	000			MR:	.ASCIZ <377>/R/
2850	015325	377	051120	043517		MERR2:	.ASCIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT./
2851	015332	040522	020115	047111			
2852	015340	044504	040503	042524			
2853	015346	020123	047516	042040			
2854	015354	053105	041511	051505			
2855	015362	050040	042522	042523			
2856	015370	052116	000056				
2857	015374	044777	051516	043125		MERR3:	.ASCIZ <377>/INSUFFICIENT DATA! /
2858	015402	044506	044503	047105			
2859	015410	020124	040504	040524			
2860	015416	000041					
2861	015420	052377	051505	020124		MTSTPC:	.ASCIZ <377>/TEST PC-/
2862	015426	041520	000055				
2863	015432	046377	041517	020113		MLOCK:	.ASCIZ <377>/LOCK ON SELECTED TEST/
2864	015440	047117	051440	046105			
2865	015446	041505	042524	020104			
2866	015454	042524	052123	000			
2867	015461	103	051123	020072		MCSRX:	.ASCIZ /CSR: /
2868	015466	000					
2869	015467	126	041505	020072		MVECX:	.ASCIZ /VEC: /
2870	015474	000					
2871	015475	120	051501	042523		MPASSX:	.ASCIZ /PASSES: /
2872	015502	035123	000040				
2873	015506	051105	047522	051522		MERRX:	.ASCIZ /ERRORS: /
2874	015514	020072	000				
2875	015517	377	052377	051505		MTSTN:	.ASCIZ <377><377> /TEST NO: /
2876	015524	020124	047516	020072			
2877	015532	000					
2878	015533	377	042523	020124		MNEW:	.ASCIZ <377>/SFT SWITCH REG TO D011'S DESIRED ACTIVE./
2879	015540	053523	052111	044103			
2880	015546	051040	043505	052040			



2881	015554	020117	050504	030461	
2882	015562	051447	042040	051505	
2883	015570	051111	042105	040440	
2884	015576	052103	053111	027105	
2885	015604	000			
2886	015605	120	035103	000040	MERRPC: .ASCIZ /PC: /
2887	015612	046777	050101	047440	XHEAD: .ASCIZ <377>/MAP OF DQ11 STATUS/<377>
2888	015620	020106	050504	030461	
2889	015626	051440	040524	052524	
2890	015634	177523	000		
2891		015640			.EVEN
2892	015640	000002			XSTATQ: 2
2893	015642	006	003		.BYTE 6.3
2894	015644	001244			TEMP1
2895	015646	006	002		.BYTE 6.2
2896	015650	001246			TEMP2
2897					.EVEN
2898					
2899					.BUFFERS FOR INPUT-OUTPUT
2900					
2901	015652	000000			INBUF: 0
2902		015714			.+40
2903	015714	000000			TEMP: 0
2904		015756			-.+40
2905	015756	000000			MDATA: 0
2906		016020			.+40
2907	016020	000000			.ERRTA: 0
2908	016022	000000			0 ;HALT 0
2909	016024	000000			0
2910	016026	016174			EM0
2911	016030	000000			0 ;HALT 1
2912	016032	000000			0
2913	016034	016314			EM1
2914	016036	015563			DH0 ;HALT 2
2915	016040	017230			DT1
2916	016042	016361			EM2
2917	016044	016624			DH1 ;HALT 3
2918	016046	000000			0
2919	016050	016361			EM2
2920	016052	016643			DH2 ;HALT 4
2921	016054	000000			0
2922	016056	016361			EM2
2923	016060	016664			DH3 ;HALT 5
2924	016062	000000			0
2925	016064	016361			EM2
2926	016066	016715			DH4 ;HALT 6
2927	016070	000000			0
2928	016072	016361			EM2
2929	016074	016743			DH5 ;HALT 7
2930	016076	000000			0
2931	016100	016361			EM2
2932	016102	016777			DH6 ;HALT 10
2933	016104	000000			0
2934	016106	016361			EM2
2935	016110	017030			DH7 ;HALT 11
2936	016112	000000			0

2937	016114	016361				EM2	
2938	016116	017050				DH10	;HALT 12
2939	016120	000000				0	
2940	016122	016361				EM2	
2941	016124	017075				DH11	;HALT 13
2942	016126	000000				0	
2943	016130	017124				DH12	
2944	016132	000000				0	;HALT 14
2945	016134	000000				0	
2946	016136	016403				EM3	
2947	016140	017203				DH14	;HALT 15
2948	016142	017254				DT3	
2949	016144	016462				EM4	
2950	016146	017160				DH13	;HALT 16
2951	016150	017242				DT2	
2952	016152	016506				EM5	
2953	016154	017203				DH14	;HALT 17
2954	016156	017254				DT3	
2955	016160	016533				EM6	
2956	016162	000000				0	;HALT 20
2957	016164	000000				0	
2958	016166	017124				DH12	
2959	016170	017214				DH15	;HALT 21
2960	016172	000000				0	
2961	016174	051377	041505	044505	EM0:	.ASCIZ	<377>/RECEIVER DONE NOT SET./

	016224	052377	042510	041440	MSG13:	.ASCIZ	<377>/THE CHARACTER DETECT OPTION <BB> IS NOT INSTALLED!!!!/
	016314	041777	040510	040522	EM1:	.ASCIZ	<377>/CHARACTER DETECTION TEST <SET FLAG>/
	016361	377	042524	052123	EM2:	.ASCIZ	<377>/TEST OF SEQ REG /
	016403	377	041502	020103	EM3:	.ASCIZ	<377>/BCC TEST-APPEND FAILURE. DQ11 ERROR FLAG SET./
	016462	042377	052101	020101	EM4:	.ASCIZ	<377>/DATA COMPARE ERROR/
	016506	042377	030521	020061	EM5:	.ASCIZ	<377>/DQ11 ERROR FLAG SET/
	016533	377	047516	051040	EM6:	.ASCIZ	<377>/NO RECIEVER INTERUPTS./
	016563	377	044103	051101	DH0:	.ASCIZ	<377>/CHAR RECEIVED EXPECTED ADDRESS/
	016624	041377	052111	030460	DH1:	.ASCIZ	<377>/BIT01 -SET T-/
	016643	377	044502	030124	DH2:	.ASCIZ	<377>/BIT02 -CLEAR T-/
	016664	041377	052111	031460	DH3:	.ASCIZ	<377>/BIT03 -BCC START CLEAR-/
	016715	377	044502	030124	DH4:	.ASCIZ	<377>/BIT06 -CLEAR ACTIVE-/
	016743	377	044502	030124	DH5:	.ASCIZ	<377>/BIT07 -SET DONE; CLEAR GO-/
	016777	377	044502	030124	DH6:	.ASCIZ	<377>/BIT08 -CHARACTER STRIP-/
	017030	041377	052111	030061	DH7:	.ASCIZ	<377>/BIT10 -TX PAD-/
	017050	041377	052111	030461	DH10:	.ASCIZ	<377>/BIT11 -BCC EXCLUDE-/
	017075	377	044502	030124	DH11:	.ASCIZ	<377>/BIT08 -RX CHAR STRIP-/
	017124	046777	046125	044524	DH12:	.ASCIZ	<377>/MULTIPLE FUNCTIONS FAILURE/
	017160	042777	050130	041505	DH13:	.ASCIZ	<377>/EXPECTED RECEIVED/
	017203	377	050504	051105	DH14:	.ASCIZ	<377>/DQERR /
	017214	042377	042514	051440	DH15:	.ASCIZ	<377>/DLE STUCK /

	017230	000002			.EVEN		
					DT1:	2	
2962	017232	006	011		.BYTE	6.9.	
2963	017234	013044			GDCHAR		
2964	017236	002	002		.BYTE	2.2	
2965	017240	013042			ADDR		
2966	017242	000002			DT2:	2	
2967	017244	003	006		.BYTE	3.6	
2968	017246	013044			GDCHAR		
2969	017250	003	002		.BYTE	3.2	

CZDQF MACY11 30(1046) 05-MAY-81 14:18 PAGE 59  
CZDQFF.P11 05-MAY-81 09:27 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

SEQ 0058

2970	017252	013036				CHAR
2971	017254	000001			DT3:	1
2972	017256	006	002			.BYTE 6.2
2973	017260	013030				ERR
2974		000001			.END	

















.CNVRT	014160	855	2628#			
.CONVR	014154	853	2626#			
.EOP	013050	1112	2136	2398#		
.ERRTA	016020	2716	2907#			
.HLT	014434	646	2696#			
.INSTE	013614	845	2529#			
.INSTG	013620	2528	2531#			
.INSTR	013476	843	2505#			
.INST1	013516	2509#	2520	2536		
.MEMCL	010652	859	2211#			
.MSG	013520	2507*	2510#	2813*	2814*	2815
.MSTCL	010766	857	2231#			
.PARAM	013646	847	2543#			
.PFAIL	014764	644	934	2774#	2782	
.RESOS	014122	851	2616#			
.SAVOS	014062	849	2602#			
.SCOPE	013236	837	2449#			
.SCOPI	013350	839	2473#			
.START	001512	695	932#	944		
.SYNC	C12010	2261	2294	2371#		
.TRPSR	C14402	648	2685#			
.TRPTA	001314	835#	2690			
.TYPE	013370	841	2481#			

DOEND	1#	2391															
DQFRNT	1#	522															
HLT	586#	1147	1152	1206	1211	1465	1471	1504	1508	1550	1601	1610	1647	1651	1654		
	1693	1697	1700	1703	1728	1742	1773	1787	1847	1856	1903	1907	1944	1948	1984		
	1987	1990	1993	2046	2050	2070	2086	2102	2176	2179	2182	2185	2189	2193	2198		
	2201	2204	2343	2358													
IDENT	1#																
ORANGE	1#	1034															
TESTA1	1#																
TESTB1	1#																
TESTC1	1#																
TESTD1	1#																
TESTE1	1#																
TESTF1	1#																
TESTH1	1#																
TESTH2	1#																
\$BEGGE	1088#	2240															
\$BEGIN	1#	1004															
\$BUFFE	1#	2898															
\$CATCH	1#	639															
\$CLRVE	1#	969															
\$CONVR	1#	2623															
\$EOP	1#	2391															
\$EXTSU	1088#	2272															
\$GETFL	1#																
\$GETPA	1#	1022															
\$HEADE	1#	522															
\$HLT	1#	2693															
\$INSTR	1#	2502															
\$INTNP	1#																
\$MAINT	1#																
\$MSG	1#	2839															
\$PARAM	1#	2540															
\$PFAIL	1#	2771															
\$REG	1#	2599															
\$SCOPE	1#	2446															
\$SCOP1	1#	2470															
\$SETFL	1#																
\$SETVE	1#	641															
\$START	1#	924															
\$SYMBO	1#	539															
\$TRAPS	1#	827															
\$TRPDE	1#	836	838	840	842	844	846	848	850	852	854	856	858	860	862		
\$TRPSR	1#	2680															
\$TSTN	1#	1034	1090	1105	1169	1231	1244	1257	1270	1283	1296	1309	1322	1335	1348		
	1361	1374	1387	1400	1413	1426	1481	1522	1564	1621	1666	1713	1758	1803	1867		
	1917	1956	2004	2060	2076	2092	2133										
\$TYPE	1#	2478															
\$VARIA	1#	768															
\$XSCOP	1088#	1153	1212	1636	1684	1745											
\$YIPPY	1088#	2056	2072	2088													

. ABS. 017262 000

CZDQF MACY11 30(1046) 05-MAY-81 14:18 PAGE 71  
CZDQFF.P11 05-MAY-81 09:27 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0068

ERRORS DETECTED: 0

CZDQFF.BIN,CZDQFF.LST/CRF/SOL/NL:TOC=CZDQXX.MAC,CZDQFF.P11  
RUN-TIME: 7 10 1 SECONDS  
RUN-TIME RATIO: 112/18=5.9  
CORE USED: 21K (42 PAGES)