

IDENTIFICATION

PRODUCT CODE: AC-8615E-MC
PRODUCT NAME: CZDQDE0 DQ11 RCVR & XMTR
DATE: JUNE 1978
MAINTAINER: DIAGNOSTIC GROUP

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

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

COPYRIGHT (C) 1974, 1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

1. ABSTRACT

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

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

THE SEVEN DIAGNOSTICS ARE:

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

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

AND A PARAMETER INPUT PROGRAM IS AVAILABLE

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

2. REQUIREMENTS

2.1 EQUIPMENT

ANY PDP11 FAMILY CPU (WITH MINIMUM 8K MEMORY)-WITH
OR WITHOUT A HARDWARE SWITCH REGISTER (LOC. 177570)
ASR 33 (OR EQUIVALENT)
DQ11
SYNC MODEM (ONLY REQUIRED FOR ONLINE TEST)

2.2 STORAGE

PROGRAM WILL LOAD AND RUN
IN 8K OF MEMORY.
LOCATION 1400 THRU 1600 ARE ESPECIALLY TO
BE NOTED AND TO BE UNTOUCHED BY OPERATOR
AFTER DQ11 TRIAL PROGRAM HAS BEEN EXECUTED.
OR AFTER THE 'AUTO SIZING' HAS BEEN DONE.

3. LOADING PROCEEDURE

3.1 METHOD

ALL PROGRAMS ARE IN ABSOLUTE FORMAT AND

ARE LOADED USING THE ABSOLUTE LOADER.

ABSOLUTE LOADER STARTING ADDRESS *500

MEMORY *
SIZE

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

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

3.1.2 THEN START

4. STARTING PROCEEDURE

A. LOAD LOC. 200

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

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

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

C.THEN START

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

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

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

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

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

4.1 CONTROL SWITCH SETTINGS

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

CONTROL:

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

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

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

4.1.2 SWITCH REGISTER RESTRICTIONS

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

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

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

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

4.1.3 SWITCH REGISTER PRIORITYS

ERROR SWITCHES

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

HLT (ERROR) ROUTINE SUPPORTS <^G> OPERATION

SCOPE SWITCHES

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

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

4.2 STARTING ADDRESS

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

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

5. OPERATING PROCEDURE

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

AND PROGRAM WILL BEGIN RUNNING THE
DIAGNOSTIC

5.2 PROGRAM AND/OR OPERATOR ACTION

THE TYPICAL APPROACH SHOULD BE

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

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

6. ERRORS

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

6.2 ERROR RECOVERY

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

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

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

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

SEE SECTION 4. (PLEASE)

7.2 OPERATING RESTRICTIONS

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

8. MISCELLANEOUS

8.1 EXECUTION TIME

8.2 PASS COMPLETE

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

END PASS AC-8615E-MC CSR: 160000 VEC: 300 PASSES: 000001 ERRORS: 000000

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

THEY ARE ONLY FOR THIS EXAMPLE.

8.3 TST1 (MINI MONITOR)

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

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

8.4 KEY LOCATIONS

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

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

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

DQACTV (1500) EACH BIT SET IN THIS LOCATION INDICATES
THAT THE ASSOCIATED DQ11 WILL BE TESTED
IN TURN.
EXAMPLE:
(DQACTV) 1500/0000000000011111
MEANS THAT DQ11 NO. 00,01,02,03,04
WILL BE TESTED.
EXAMPLE:
(DQACTV) 1500/0000000000010001
MEANS THAT DQ11 NO. 00,04
WILL BE TESTED.

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

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

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

BIT 00-08 VECTOR 'A' OF DEVICE

8.5 *** METHOD OF AUTO SIZING ***

8.5.1 FINDING THE CONTROL STATUS REGISTER.

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

8.5.2 ONE SYNC BIT OR TWO?

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

8.5.3 'BB' OPTION INSTALLED?

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

8.5.4 'AB' OPTION INSTALLED?

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

8.5.5 'BA' OPTION INSTALLED?

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

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

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

8.5.7 ACTIVE ON FIRST NON-SYNC?

SINCE TOO MUCH HARDWARE MUST BE TURNED ON TO SENSE FOR WHEN THE DQ11 GOES ACTIVE THE PROGRAM ASSUMES 'ACTIVE ON FIRST NON-SYNC'. NOTE: THIS CAN BE CHANGED BY ALTERING BIT 11 IN THE APPRIQATE DQSTXX: AFTER AUTO SIZING

8.5.8 SET FOR ODD OR EVEN PARITY?

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

8.5.9 FINDING THE VECTOR.

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

9. PROGRAM DESCRIPTION

CONTAINED WITHIN LISTING

10. LISTING

FOLLOWING

```
522 .ENABLE AMA
523
524 ;CZDQDE0/<377>/TRANSMITTER AND RECEIVER EXERCISER
525 ;COPYRIGHT 1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
526
527 ;REVISED 16-DEC-76 BY R. BLACK
528 : A)SUPPORTS SOFTWARE SWITCH REGISTER
529 : B)SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER
530 : BY <^G>.
531 : STARTING PROCEDURE
532 : LOAD PROGRAM
533 : LOAD ADDRESS 000200
534 : PRESS START
535 : PROGRAM WILL TYPE 'CZDQDE0/<377>/TRANSMITTER AND RECEIVER EXERCISER''
536 : PROGRAM WILL TYPE 'R' TO INDICATE THAT TESTING HAS STARTED
537 : AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
538 : AND THEN RESUME TESTING
539
540
541 ;SWITCH REGISTER OPTIONS
542
543 100000 SW15=100000 :=1,HALT ON ERROR
544 040000 SW14=40000 :=1,LOOP ON CURRENT TEST
545 020000 SW13=20000 :=1,INHIBIT ERROR TYPEOUT
546 010000 SW12=10000 :=1,DELETE TYPEOUT/BELL ON ERROR.
547 004000 SW11=4000 :=1,INHIBIT ITERATIONS
548 002000 SW10=2000 :=1,ESCAPE TO NEXT TEST ON ERROR
549 001000 SW09=1000 :=1,LOOP WITH CURRENT DATA
550 000400 SW08=400 :=1,LOOP ON ERROR
551 000100 SW06=100
552 000040 SW05=40
553 000020 SW04=20
554 000010 SW03=10
555 000004 SW02=4 ;LOCK ON TEST SELECT
556 000002 SW01=2 ;RESTART PROGRAM AT SELECTED TEST
557 000001 SW00=1 ;RESELECT DQ'1 DESIRED ACTIVE
558 ;NOTE: THIS MUST NOT EXCEED ORIGINAL COUNT
```

;REGISTER DEFINITIONS

550			
560			
561			
562			
563	000000	R0=%0	;GENERAL REGISTER
564	000001	R1=%1	;GENERAL REGISTER
565	000002	R2=%2	;GENERAL REGISTER
566	000003	R3=%3	;GENERAL REGISTER
567	000004	R4=%4	;GENERAL REGISTER
568	000005	R5=%5	;GENERAL REGISTER
569	000006	SP=%6	;PROCESSOR STACK POINTER
570	000007	PC=%7	;PROGRAM COUNTER

;LOCATION EQUIVALENCIES

571			
572			
573			
574	177570	DSWR= 177570	;HARDWARE SWITCH REGISTER LOC.
575	177570	DLIGHTS=177570	;HARDWARE DISPLAY REGISTER LOC.
576	177776	PS=177776	;PROCESSOR STATUS WORD
577	001200	STACK=1200	;START OF PROCESSOR STACK

;INSTRUCTION DEFINITIONS

578			
579			
580			
581	005746	PUSH1SP=5746	;DECREMENT PROCESSOR STACK 1 WORD
582	005726	POP1SP=5726	;INCREMENT PROCESSOR STACK 1 WORD
583	010046	PUSHR0=10046	;SAVE R0 ON STACK
584	012600	POPR0=12600	;RESTORE R0 FROM STACK
585	024646	PUSH2SP=24646	;DECREMENT STACK TWICE
586	022626	POP2SP=22626	;INCREMENT STACK TWICE
587		.EQUIV EMT,HLT	;BASIC DEFINITION OF ERROR CALL

588			
589			
590	100000	BIT15=100000	
591	040000	BIT14=40000	
592	020000	BIT13=20000	
593	010000	BIT12=10000	
594	004000	BIT11=4000	
595	002000	BIT10=2000	
596	001000	BIT9=1000	
597	000400	BIT8=400	
598	000200	BIT7=200	
599	000100	BIT6=100	
600	000040	BIT5=40	
601	000020	BIT4=20	
602	000010	BIT3=10	
603	000004	BIT2=4	
604	000002	BIT1=2	
605	000001	BIT0=1	

;DQ11 OPTIONAL DEFINITIONS

606			
607			
608			
609			
610	002000	ABBIT=2000	
611	004000	ACTBIT=4000	
612	010000	BABIT=10000	
613	020000	BBBIT=20000	
614	040000	JUMBIT=40000	

615 001000 ODDBIT=1000
616 100000 SYNBIT=100000

:DQ11 SECONDARY REGISTER DEFINATIONS

621 000000 RXBA.P=0 ;RECEIVER BUS ADDRESS PRIMARY.
622 0000C1 RXWC.P=1 ;RECEIVER WORD COUNT PRIMARY.
623 000002 TXBA.P=2 ;TRANSMITTER BUS ADDRESS PRIMARY.
624 000003 TXWC.P=3 ;TRANSMITTER BUS ADDRESS PRIMARY.
625 000004 RXBA.S=4 ;RECEIVER BUS ADDRESS SECONDARY.
626 000005 RXWC.S=5 ;RECEIVER WORD COUNT SECONDARY.
627 000006 TXBA.S=6 ;TRANSMITTER BUS ADDRESS SECONDARY.
628 000007 TXWC.S=7 ;TRANSMITTER WORD COUNT SECONDARY.
629
630 000010 CHARDT=10 ;CHARACTER DETECT REGISTER.
631 000011 SYNC.=11 ;SYNC REGISTER.
632 000012 MISC.=12 ;MISCELLANEOUS REGISTER.
633 000013 TX.MUX=13 ;TRANSMITTER MUX REGISTER.
634 000014 SEQ.=14 ;SEQUENCE REGISTER.
635 000015 RX.BCC=15 ;RECEIVER BCC REGISTER.
636 000016 TX.BCC=16 ;TRANSMITTER BCC REGISTER.
637 000017 POLY.=17 ;POLYNOMIAL REGISTER.
638
639

TRAPCATCHER FOR UNEXPECTED INTERRUPTS

```

640 ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
641 000000 . = 0
642 ;STANDARD INTERRUPT VECTORS
643
644 . = 24
645 000024 017044 .PFAIL ;POWER FAIL HANDLER
646 000026 000340 340 ;SERVICE AT LEVEL 7
647 000030 016514 .HLT ;ERROR HANDLER
648 000032 000340 340 ;SERVICE AT LEVEL 7
649 000034 016462 .TRPSRV ;GENERAL HANDLER DISPATCH SERVICE
650 000036 000340 340 ;SERVICE AT LEVEL 7
651 . = 46
652 000046 015242 LOGICAL ;ACT HOOKS
653 . = 52
654 000052 000000 .WORD 0
655 ;THIS ROUTINE TRIES TO FORCE THE RECEIVER TO INTERRUPT
656 ;TO ITS VECTOR WHERE IT WILL PICK UP THE STATUS LOCATION
657 ;FOR ITS NEW PC; AND PICK UP AN IOT INSTRUCTION FOR ITS
658 ;NEW PS. WHEN THE NEW PC IS FETCHED AN IOT INSTRUCTION IS
659 ;EXECUTED, TRAPPING TO LOCATION 20 WHERE A ROUTINE IS EXECUTED
660 ;TO TAKE THE PC FROM THE STACK AND USE IT AS THE VECTOR ADDRESS
661 000056 . = 56
662
663 000056 VECMAP:
664 000056 010120 1$: MOV R1,(R0)+ ;START FILLING THE VECTOR AREA
665 000060 012721 000004 MOV #4,(R1)+ ;WITH +2; IOT (4)
666 000064 022021 CMP (R0)+,(R1)+ ;UPDATE THE POINTERS
667 000066 020127 001000 CMP R1,#1000 ;IS ALL FLOATING VECTOR AREA DONE
668 000072 101771 BLOS 1$ ;BR IF NOT ALL DONE
669 000074 012737 000146 000020 MOV #4$,@#20 ;SET FOR IOT TRAP BY DQ11
670 000102 013737 001500 001244 MOV DQACTV,TEMP1 ;GET THE ACTIVE DQ11 S
671 000110 006037 001244 2$: ROR TEMP1 ;ARE YOU ACTIVE.. DQ11
672 000114 103023 BCC 5$ ;IF CARRY CLEAR.. NO MORE DQ11S
673 000116 005037 177776 CLR PS ;CLEAR PS
674 000122 005722 TST (R2)+ ;PUT POINTER TO STATUS TABLE
675 000124 012772 000340 177776 MOV #340,@-2(R2) ;TRY AND SET PRI/SEC DONE AND IE
676 000132 105200 INCB R0 ;DELAY.....
677 000134 001376 BNE -2 ;.....DELAY
678 000136 112712 000300 MOVB #300,(R2) ;NO INTERRUPT ASSUME 300 FIX IN TEST C
679 000142 005722 3$: TST (R2)+ ;UPDATE POINTERS
680 000144 000761 BR 2$ ;GO DO IT AGAIN
681 000146 051612 4$: BIS (SP),(R2) ;ENTERD BY IOT TRAP BY DQ11
682 000150 042712 000007 BIC #7,(R2) ;CLEAR UNWANTED BITS
683 000154 022626 CMP (SP)+,(SP)+ ;POP IOT JUNK OFF STACK
684 000156 012716 000142 MOV #3$,(SP) ;SET RETURN PC ON STACK
685 000162 000002 RTI ;GC HOME
686 000164 000207 5$: RTS PC ;ALL SIZING IS DONE
687
688 ;****SOFTWARE SWITCH REGISTER****
689 . 174
690 000174 000000 DISPREG: 0 ;SOFTWARE DISPLAY REGISTER
691 000176 000000 SWREG: 0 ;SOFTWARE SWITCH REGISTER
692
693 ;PROGRAM START
694
695 000200 . -200
  
```

```

696 000200 000137 001512      JMP      .START      ;GO TO START OF PROGRAM
697
698      000220
699 000220 012702 001400      . =220
700 000224 005022      CSRMAP: MOV      #1400,R2      ;CLEAR ALL STATUS TABLE
701 000226 022702 001512      CLR      (R2)+      ;DO CLEAR
702 000232 001374      CMP      #1512,R2    ;ALL TABLE DONE
703 000234 005037 001504      BNE     .-6         ;BR IF MORE TO GO
704 000240 012702 001400      CLR      DQNUM      ;SET NUMBER OF DQ11S TO 0
705 000244 012701 160000      MOV      #1400,R2    ;SET TABLE POINTER
706 000250 012737 000614 000004      MOV      #160000,R1   ;GET FIRST FLOATING ADDRESS
707 000256 112761 000012 000005 1$:  MOV      #5$,R4      ;SET FOR TIME OUT TRAP--NO DEVICE--
708 000264 005061 000006      MOVVB   #12,5(R1)    ;TRY AND SEL MISC REGISTER
709 000270 012711 010000      CLR      6(R1)      ;TRY AND CLEAR MISC REG
710 000274 022761 030000 000006      MOV      #10000,(R1) ;TRY AND SET RX ACTIVE
711 000302 001071      CMP      #30000,6(R1);LOOK FOR SYNC 1 AND SYNC 2
712 000304 010122      BNE     2$         ;THIS IS NOT A DQ11 IF I BRANCH
713 000306 052712 100000      MOV      R1,(R2)+    ;NOW THIS IS A DQ11 --STORE CSR
714 000312 005011      BIS     #SYNBIT,(R2) ;SET FOR TWO SYNC CHARS
715 000314 112761 000000 000005      CLR      (R1)      ;CLEAR DQ ACTIVE BIT
716 000322 012761 177777 000006      MOVVB   #10,5(R1)   ;SEL CHAR DET REGISTER
717 000330 005761 000006      MOV      #-1,6(R1)  ;WRITE INTO CHAR DET REG
718 000334 001402      TST     6(R1)      ;WAS THE REGISTER WRITTEN?
719 000336 052712 020000      BEQ     .+6        ;APPERENTLY NO BB OPTION.
720 000342 112761 000017 000005      BIS     #BBBIT,(R2) ;SET FOR BB OPTION
721 000350 012761 177777 000006      MOVVB   #17,5(R1)   ;SEL POLYNO. REGISTER
722 000356 005761 000006      MOV      #-1,6(R1)  ;WRITE POLYNO.REGISTER
723 000362 001402      TST     6(R1)      ;WAS REG WRITTEN??
724 000364 052712 002000      BEQ     .+6        ;BR IF NO AB OPTION
725 000370 012761 001400 000002      BIS     #ABBIT,(R2) ;SET FOR AB OPTION
726 000376 032761 001400 000002      MOV      #1400,2(R1);TRY TO SET .DTR. .RS.
727 000404 001402      BIT     #1400,2(R1) ;DID ANY OF THEM SET
728 000406 052712 010000      BEQ     .+6        ;BR IF NO BA OPTION
729 000412 032761 030000 000002      BIS     #BABIT,(R2) ;SET FOR BA OPTION
730 000420 001402      BIT     #30000,2(R1);DID .CS. .CO. SET
731 000422 052712 040000      BEQ     .+6        ;BR IF NO JUMPER
732 000426 052712 004000      BIS     #JUMBIT,(R2);SET FOR JUMPER
733 000432 052712 001000      BIS     #ACTBIT,(R2);SET FOR ACTIVE ON FIRST NON-SYNC
734 000436 005722      BIS     #ODDBIT,(R2);SET FOR ODD VRC.....
735 000440 005011      TST     (R2)+      ;POP POINTER
736 000442 005061 000002      CLR      (R1)      ;CLEAR RCSR
737 000446 005061 000002      CLR      2(R1)     ;CLEAR TCSR
738 000452 005061 000004      CLR      4(R1)     ;CLEAR AGAIN
739 000456 005061 000006      CLR      6(R1)     ;CLEAR ERROR REG
740 000462 005237 001504      CLR      6(R1)     ;CLEAR SEC REG
741 000466 062701 000010 2$:  INC     DQNUM      ;UPDATE NUMBER OF DQ11S
742 000472 022701 164000      ADD     #10,R1     ;UPDATE CSR POINTER BY 10 (8)
743 000476 001267      CMP     #164000,R1 ;HAVE ALL FLOATING ADDRESSES BEEN CHECKED??
744 000500 005037 001500      BNE     1$         ;BR IF NOT ALL DONE
745 000504 005737 001504      CLR     DQACTV     ;ZERO ACTIVE DQ11S
746 000510 001434      TST     DQNUM      ;WERE ANY DQ11S FOUND
747 000512 013701 001504      BEQ     4$         ;HEY BUDDY. NO DQ11S FOUND IN SYSTEM
748 000516 010137 001276      MOV     DQNUM,R1   ;SAVE NUMBER OF DQ11S
749 000522 000241 3$:  MOV     R1,SAVNUM  ;SAVE NUMBER FOR ACT11
750 000524 006137 001500      CLC     ;CLEAR CARRY
751 000530 005237 001500      ROL     DQACTV     ;ACTIVE ADDRESS
      INC     DQACTV  ;SET BIT 0
  
```



```

752 000534 005301          DEC      R1          ;DEC NUMBER OF DQ11S
753 000536 001371          BNE      3$          ;BR IF MORE TO GO
754 000540 012737 000006 000004  MOV      #6,@#4      ;RESET TIME OUT VECTOR
755 000546 013737 001500 001502  MOV      DQACTV,SAVACT ;SAVE ACTIVE
756 000554 012737 000340 000022  MOV      #340,@#22   ;SET IOT TRAP PRIO: TO 7
757 000562 012702 001400          MOV      #1400,R2    ;SET TABLE POINTER
758 000566 012700 000300          MOV      #300,R0     ;SET VECTOR START
759 000572 012701 000302          MOV      #302,R1     ;SET VECTOR+2 START
760 000576 000137 000056          JMP      VECMAP      ;GO FIND THE VECTORS
761 000602 104402          4$:      TYPE        ;TYPE MESSAGE
762 000604 017404          MERR2      ;I DIDN'T FIND ANY DQ11S. DON'T USE AUTO SIZE.
763 000606 005000          CLR      R0          ;
764 000610 000000          HALT      ;HOW CAN I TEST NO DQ11S
765 000612 000776          BR        -2         ;DON'T LET OPR HIT CONT. SW
766 000614 012716 000466          5$:      MOV      #2$, (SP) ;ENTERED BY TIME OUT TRAP
767 000620 000002          RTI          ;GO HOME.
768
769
770          001000          . 1000
771 001000 005377 055103 050504  MTITLE: .ASCIZ <377><12>/CZDQDE0/<377>/TRANSMITTER AND RECEIVER EXERCISER/<377>
772 001006 042504 177460 051124
773 001014 047101 046523 052111
774 001022 042524 020122 047101
775 001030 020104 042522 042503
776 001036 053111 051105 042440
777 001044 042530 041522 051511
778 001052 051105 000377
779
780          001200          .-1200
781          ;INDIRECT POINTERS
782
783 001200 177570          SWR:      177570      ;SWITCH REGISTER POINTER
784 001202 177570          LIGHTS:  177570     ;DISPLAY REGISTER POINTER
785 001204 177560          TKCSR:   177560     ;TELETYPE KEYBOARD CONTROL REGISTER
786 001206 177562          TKDBR:   177562     ;TELETYPE KEYBOARD DATA BUFFER
787 001210 177564          TPCSR:   177564     ;TELEPRINTER CONTROL REGISTER
788 001212 177566          TPDBR:   177566     ;TELEPRINTER DATA BUFFER
789
790          ;PROGRAM CONTROL PARAMETERS
791
792 001214 000000          RETURN:  0          ;SCOPE ADDRESS FOR LOOP ON TEST
793 001216 000000          NEXT:    0          ;ADDRESS OF NEXT TEST TO BE EXECUTED
794 001220 000000          LOCK:    0          ;ADDRESS FOR LOCK ON CURRENT DATA
795 001222 000003          ICOUNT:  3          ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
796 001224 000000          LPCNT:   0          ;NUMBER OF ITERATIONS COMPLETED
797 001226 000000          TSTNO:   0          ;NUMBER OF TEST IN PROGRESS
798 001230 000000          PASCNT:  0          ;NUMBER OF PASSES COMPLETED
799 001232 000000          ERRCNT:  0          ;TOTAL NUMBER OF ERRORS
800 001234 000000          LSTERR:  0          ;PC OF LAST ERROR CALL
801
802          ;PROGRAM VARIABLES
803
804 001236 000000          CHAR1:   0
805 001240 000000          CHAR2:   0
806 001242 000000          CHAR3:   0
807 001244 000000          TEMP1:  0          ;TEMPORARY STORAGE

```

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

808	001246	000000	TEMP2:	0		:TEMPORARY STORAGE
809	001250	000000	TEMP3:	0		:TEMPORARY STORAGE
810	001252	000000	TEMP4:	0		:TEMPORARY STORAGE
811	001254	000000	TEMP5:	0		:TEMPORARY STORAGE
812	001256	000000	SAVR0:	0		:R0 STORAGE
813	001260	000000	SAVR1:	0		:R1 STORAGE
814	001262	000000	SAVR2:	0		:R2 STORAGE
815	001264	000000	SAVR3:	0		:R3 STORAGE
816	001266	000000	SAVR4:	0		:R4 STORAGE
817	001270	000000	SAVR5:	0		:R5 STORAGE
818	001272	000000	SAVSP:	0		:STACK POINTER STORAGE
819	001274	000000	SAVPC:	0		:PROGRAM COUNTER STORAGE
820	001276	000000	SAVNUM:	0		
821	001300	000001	CREAM:	.BLKW 1		
822	001302	000000	RUNFLG:	0		
823	001304	000000	RUN:	0		
824	001306	000000	RUNCNT:	0		

```
825
826                ;PROGRAM CONTROL FLAGS
827
828 001310 000      INIFLG: .BYTE 0          ;PROGRAM INITIALIZATION FLAG
829 001311 000      STFLG:  .BYTE 0          ;TEST START FLAG
830 001312 000      ERRFLG: .BYTE 0          ;ERROR OCCURED FLAG
831 001313 000      LOKFLG: .BYTE 0          ;LOCK ON CURRENT TEST FLAG
832          000000  $Y=0
833
834                ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
835                ;POINTERS TO SUBROUTINES CAN BE FOUND
836                ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS
837
838                ;*****
839                ;*****
840 001314          .TRPTAB:
841          104400  SCOPE=TRAP+0          ;CALL TO SCOPE LOOP AND ITERATION HANDLER
842 001314 015316          .SCOPE
843          104401  SCOPE1=TRAP+1         ;CALL TO LOOP ON CURRENT DATA HANDLER
844 001316 015430          .SCOPE1
845          104402  TYPE=TRAP+2          ;CALL TO TELETYPE OUTPUT ROUTINE
846 001320 015450          .TYPE
847          104403  INSTR=TRAP+3         ;CALL TO ASCII STRING INPUT ROUTINE
848 001322 015556          .INSTR
849          104404  INSTER=TRAP+4        ;CALL TO INPUT ERROR HANDLER
850 001324 015674          .INSTER
851          104405  PARAM=TRAP+5         ;CALI TO NUMERICAL DATA INPUT ROUTINE
852 001326 015726          .PARAM
853          104406  SAV05=TRAP+6         ;CALL TO REGISTER SAVE ROUTINE
854 001330 016142          .SAV05
855          104407  RES05=TRAP+7         ;CALL TO REGISTER RESTORE ROUTINE
856 001332 016202          .RES05
857          104410  CONVRT=TRAP+10       ;CALL TO DATA OUTPUT ROUTINE
858 001334 016234          .CONVRT
859          104411  CNVRT=TRAP+11        ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.
860 001336 016240          .CNVRT
861          104412  MSTCLR=TRAP+12       ;CALL TO ISSUF MASTER CLEAR
862 001340 012116          .MSTCLR
863          104413  MEMCLR=TRAP+13       ;CALL TO CLEAR ALL SCRATCH PAD MEMORIES
864 001342 011772          .MEMCLR
865          104414  CKSWR=TRAP+14        ;CALL TO ALLOW SWREG TO BE LOADED FROM TTY
866 001344 017142          .CKSWR
867          104415  CNTLU=TRAP+15       ;CALL TO ALLOW LOADING OF SWREG FROM TTY
868 001346 017216          .CNTLU
869
870                ;*****
871                ;*****
872
873                ;DQ11 VECTOR AND REGISTER INDIRECT POINTERS
874
875 001350 000000  DQRVEC: 0          ;POINTER TO DQ11 RECEIVER INTERRUPT VECTOR
876 001352 009000  DQRLVL: 0         ;POINTER TO DQ11 RECEIVER INTERRUPT SERVICE PS
877 001354 000000  DQTVEC: 0         ;POINTER TO DQ11 TRANSMITTER INTERRUPT VECTOR
878 001356 000000  DQTLVL: 0         ;POINTER TO DQ11 TRANSMITTER INTERRUPT SERVICE PS
879 001360 000000  DGRCSR: 0         ;POINTER TO DQ11 RECEIVER CONTROL REGISTER
880 001362 000000  DQRCSH: 0         ;POINTER TO HIGH BYTE OF DQ11 RECEIVER CONTROL REGISTER
```

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```

881 001364 000000      DQTCR: 0          ;POINTER TO DQ11 TRANSMITTER CONTROL REGISTER
882 001366 000000      DQERR: 0         ;POINTER TO DQ11 ERROR REGISTER
883 001370 000000      DQREG: 0         ;POINTER TO HIGH BYTE OF ERROR REGISTER
884 001372 000000      DQSEC: 0         ;POINTER TO DQ11 SECONDARY REGISTER
885 001374 000000      DQSECH: 0        ;POINTER TO HIGH BYTE OF DQ11 SECONDARY REGISTER
886
887
888

```

;DQ11 STATUS TABLE AND ADDRESS ASSIGNMENTS

```

890
891          001400      .=1400
892 001400 000001      DQCR00: .BLKW 1   ;CONTROL STATUS REGISTER FOR DEVICE NO: 00
893 001402 000001      DQST00: .BLKW 1   ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 00
894 001404 000001      DQCR01: .BLKW 1   ;CONTROL STATUS REGISTER FOR DEVICE NO: 01
895 001406 000001      DQST01: .BLKW 1   ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 01
896 001410 000001      DQCR02: .BLKW 1   ;CONTROL STATUS REGISTER FOR DEVICE NO: 02
897 001412 000001      DQST02: .BLKW 1   ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 02
898 001414 000001      DQCR03: .BLKW 1   ;CONTROL STATUS REGISTER FOR DEVICE NO: 03
899 001416 000001      DQST03: .BLKW 1   ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 03
900 001420 000001      DQCR04: .BLKW 1   ;CONTROL STATUS REGISTER FOR DEVICE NO: 04
901 001422 000001      DQST04: .BLKW 1   ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 04
902 001424 000001      DQCR05: .BLKW 1   ;CONTROL STATUS REGISTER FOR DEVICE NO: 05
903 001426 000001      DQST05: .BLKW 1   ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 05
904 001430 000001      DQCR06: .BLKW 1   ;CONTROL STATUS REGISTER FOR DEVICE NO: 06
905 001432 000001      DQST06: .BLKW 1   ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 06
906 001434 000001      DQCR07: .BLKW 1   ;CONTROL STATUS REGISTER FOR DEVICE NO: 07
907 001436 000001      DQST07: .BLKW 1   ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 07
908 001440 000001      DQCR10: .BLKW 1   ;CONTROL STATUS REGISTER FOR DEVICE NO: 10
909 001442 000001      DQST10: .BLKW 1   ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 10
910 001444 000001      DQCR11: .BLKW 1   ;CONTROL STATUS REGISTER FOR DEVICE NO: 11
911 001446 000001      DQST11: .BLKW 1   ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 11
912 001450 000001      DQCR12: .BLKW 1   ;CONTROL STATUS REGISTER FOR DEVICE NO: 12
913 001452 000001      DQST12: .BLKW 1   ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 12
914 001454 000001      DQCR13: .BLKW 1   ;CONTROL STATUS REGISTER FOR DEVICE NO: 13
915 001456 000001      DQST13: .BLKW 1   ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 13
916 001460 000001      DQCR14: .BLKW 1   ;CONTROL STATUS REGISTER FOR DEVICE NO: 14
917 001462 000001      DQST14: .BLKW 1   ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 14
918 001464 000001      DQCR15: .BLKW 1   ;CONTROL STATUS REGISTER FOR DEVICE NO: 15
919 001466 000001      DQST15: .BLKW 1   ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 15
920 001470 000001      DQCR16: .BLKW 1   ;CONTROL STATUS REGISTER FOR DEVICE NO: 16
921 001472 000001      DQST16: .BLKW 1   ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 16
922 001474 000001      DQCR17: .BLKW 1   ;CONTROL STATUS REGISTER FOR DEVICE NO: 17
923 001476 000001      DQST17: .BLKW 1   ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 17
924 001500 000001      DQACTV: .BLKW 1  ;HOLD ACTIVE BITS FOR TESTING
925 001502 000001      SAVACT: .BLKW 1  ;SAVE NUMBER OF ACTIVE DQ11S
926 001504 000001      DQNUM:  .BLKW 1  ;OCTAL NUMBER OF TOTAL NUMBER OD DQ11S
927 001506 000001      DQCSR:  .BLKW 1  ;CSR OF DQ11 UNDER TEST
928 001510 000001      DQSTAT: .BLKW 1  ;VECTOR AND CONFIGURATION STATUS OF DQ11 UNDER TEST
929

```

```

930          ;PROGRAM INITIALIZATION
931          ;LOCK OUT INTERRUPTS
932          ;SET UP PROCESSOR STACK
933          ;SET UP POWER FAIL VECTOR
934          ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
935          ;TYPE TITLE MESSAGE
936

```

```
937 001512 012737 000340 177776 .START: MOV #340,PS ;LOCK OUT INTERRUPTS
938 001520 012706 001200 MOV #STACK,SP ;SET UP STACK
939 001524 012737 017044 000024 MOV #.PFAIL,@#24 ;SET UP POWER FAIL VECTOR
940 001532 013737 001504 001276 MOV DQNUM,SAVNUM
941 001540 105037 001311 CLR STFLG ;CLEAR START FLAG
942 001544 005037 001230 CLR PASCNT ;CLEAR PASS COUNT
943 001550 105037 001312 CLR ERRFLG ;CLEAR ERROR FLAG
944 001554 005037 001302 CLR RUNFLG
945 001560 012737 001400 001300 MOV #1400,CREAM
946 001566 005037 001232 CLR ERRCNT ;CLEAR ERROR COUNT
947 001572 005037 001234 CLR LSTERR ;CLEAR LAST ERROR POINTER
948 001576 012737 000001 001226 MOV #1,TSTNO ;SET UP FOR TEST 1
949 001604 012737 001512 001214 MOV #.START,RETURN ;SET UP FOR POWER FAIL BEFORE
950 ;TESTING STARTS
951 001612 012737 177570 001200 MOV #DSWR,SWR ;MOV HARDWARE SWR TO SWR
952 001620 012737 177570 001202 MOV #DLIGHTS,LIGHTS ;MOV DISPLAY LIGHTS TO LIGHTS
953 001626 013746 000006 MOV @#6,-(SP) ;SAVE VECTORS
954 001632 013746 000004 MOV @#4,-(SP)
955 001636 012737 001656 000004 MOV #64$,@#4 ;SET UP FOR TIMEOUT
956 001644 022777 177777 177326 CMP #-1,@SWR ;REFERENCE HARDWARE SWITCH REGISTER
957 001652 001402 BEQ 65$
958 001654 000407 BR 66$
959 001656 022626 64$: CMP (SP)+,(SP)+ ;ADJUST STACK
960 001660 012737 000176 001200 65$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
961 001666 012737 000174 001202 MOV #DISPREG,LIGHTS ;POINT TO SOFT DISPLAY REG
962 001674 012637 000004 66$: MOV (SP)+,@#4 ;RESTORE VECTORS
963 001700 012637 000006 MOV (SP)+,@#6
964 001704 005737 000042 TST @#42 ;UNDER MONITOR
965 001710 001014 BNE 67$
966 ;:*****THE NEXT 4 LINES OF CODE MOVED TO SOLVE PR#2757 (JUNE 78)*****
967 001712 105737 001310 TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED?
968 001716 001035 BNE 12$ ;IF YES, BR
969 001720 104402 001000 TYPE ,MTITLE ;TYPE TITLE MESSAGE
970 001724 105137 001310 COMB INIFLG ;IF NOT SET FLAG AND INIT
971 001730 022737 000176 001200 CMP #SWREG,SWR ;IS SWREG USED
972 001736 001001 BNE 67$
973 001740 104415 CNTLU
974 001742 105777 177232 67$: TSTB @SWR
975 001746 100402 BMI .+6
976 001750 004737 000220 JSR PC,CSRMAP
977 001754 104402 017671 TYPE ,XHEAD
978 001760 012737 001400 001244 MOV #1400,TEMP1
979 001766 017737 177252 001246 MOV @TEMP1,TEMP2
980 001774 001406 BEQ .+16
981 001776 104410 CONVRT
982 002000 017716 XSTATQ
983 002002 062737 000002 001244 ADD #2,TEMP1
984 002010 000766 BR .-22
985 002012 032777 000001 177160 12$: BIT #SW00,@SWR
986 002020 001424 BEQ 1$
987 002022 104402 TYPE
988 002024 017612 MNEW
989 002026 005000 CLR R0
990 002030 000000 HALT
991 002032 104414 CKSWR
992 002034 027737 177140 001502 CMP @SWR,SAVACT
```

```

993 002042 101404      BLOS      11$
994 002044 104402      TYPE
995 002046 017453      MERR3
996 002050 000000      HALT
997 002052 000776      BR
998 002054 017737 177120 001500 11$:  MOV      @SWR,DQACTV
999 002062 013700 001500      MOV      DQACTV,R0
1000 002066 000000      HALT
1001 002070 104414      CKSWR
1002 002072 012700 000300 1$:  MOV      #300,R0
1003 002076 012701 000302      MOV      #302,R1
1004 002102 010120 2$:  MOV      R1,(R0)+
1005 002104 005021      CLR      (R1)+
1006 002106 022021      CMP      (R0)+,(R1)+
1007 002110 022700 001000      CMP      #1000,R0
1008 002114 001372      BNE      2$
1009
1010      ;TEST START AND RESTART
1011
1012 002116 012737 000340 177776 .BEGIN: MOV      #340,PS      ;LOCK OUT INTERRUPTS
1013 002124 012706 001200      MOV      #STACK,SP  ;SET UP STACK
1014 002130 005737 000042      TST      @#42      ;IS PROGRAM UNDER MONITOR CONTROL
1015 002134 001040      BNE      3$
1016 002136 104414      CKSWR      ;CHECK FOR <^G>
1017 002140 032777 000004 177032  BIT      #BIT2,@SWR  ;CHECK FOR LOCK ON TEST
1018 002146 001411      BEQ      1$
1019 002150 104402 017511      TYPE      ,MLOCK
1020 002154 012737 000240 015326  MOV      #NOP,TTST
1021 002162 012737 000240 015330  MOV      #NOP,TTST+2  ;SET UP TO LOCK
1022 002170 000406      BR      2$
1023 002172 013737 015424 015326 1$:  MOV      BRW,TTST
1024 002200 013737 015426 015330  MOV      BRX,TTST+2  ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1025 002206 032777 000002 176764 2$:  BIT      #SW01,@SWR  ;IF SW01=1, GET STARTING PC
1026 002214 001410      BEQ      3$
1027 002216 104403      INSTR
1028 002220 017477      MTSTPC
1029 002222 104405      PARAM
1030 002224 002254      TST1
1031 002226 007756      TLAST
1032 002230 001214      #RETURN
1033 002232 001      .BYTE 1
1034 002233 001      .BYTE 1
1035 002234 000403      BR      4$
1036 002236 012737 002254 001214 3$:  MOV      #TST1,RETURN  ;START AT TEST 1
1037 002244 104402 017401 4$:  TYPE      ,MR      ;TYPE R
1038 002250 000177 176740      JMP      @RETURN  ;START TESTING
1039      : TEST 1
1040      :*****
1041 002254 012737 000001 001226 TST1: MOV      #1,TSTNO
1042 002262 012737 002644 001214      MOV      #TST2,RETURN
1043 002270 012737 002644 001216      MOV      #TST2,NEXT
1044 002276 105737 001302      TSTB     RUNFLG      ;IS THIS MY FIRST TIME HERE?
1045 002302 001010      BNE      1$      ;BR IF FLAG IS SET
1046 002304 012737 000001 001304      MOV      #BIT0,RUN  ;SET RUN POINTER.
1047 002312 012737 000020 001306      MOV      #15,RUNCNT ;SET FOR MAX OF 16 DQ11'S PER SYSTEM
1048 002320 105137 001302      COMB     RUNFLG      ;SET RUN FLAG
  
```

PROGRAM INITIALIZATION AND START UP.

```
1049 002324 033737 001304 001500 1$: BIT RUN,DQACTV ;FIND AN ACTIVE DQ11 TO TEST.
1050 002332 001032 BNE 3$ ;BR IF I FOUND ONE TO TEST.
1051 002334 005737 001500 TST DQACTV ;FIND OUT IF THERE ARE NO DQ11 ACTIVE.
1052 002340 001423 BEQ 2$ ;BR TO FATAL ERROR. WHY AM I HERE IF NO ACTIVE DQ11'S???
1053 002342 000257 CCC ;CLEAR ALL THE CONDITION CODES OF CPU
1054 002344 006137 001304 ROL RUN ;UPDATE RUN POINTER
1055 002350 062737 000004 001300 ADD #4,CREAM ;UPDATE ADDRESS POINTER.
1056 002356 005337 001306 DEC RUNCNT ;DEC NUMBER OF TIMES I LOOKED AT ACTIVE.
1057 002362 001360 BNE 1$ ;BR AND KEEP LOOKING.
1058 002364 012737 000020 001306 MOV #16,RUNCNT ;START RESTORING MY POINTERS.
1059 002372 012737 001400 001300 MOV #1400,CREAM ;RESTORE ADDRESS POINTER
1060 002400 012737 000001 001304 MOV #1,RUN ;RESTORE RUN POINTER.
1061 002406 000746 BR 1$ ;KEEP ON TESTING.
1062 002410 104402 2$: TYPE ;ALERT OPERATOR OF FATAL ERROR
1063 002412 017404 MERR2 ;NO DQ11 ACTIVE. WHY AM I HERE???
1064 002414 000000 HALT ;YOU MUST RELOAD DQ11 DIAGNOSTIC!!
1065 002416 000776 BR .-2 ;STICK HERE ON CONT.
1066 002420 000257 3$: CCC ;CLEAR CPU COND. CODES
1067 002422 006137 001304 ROL RUN ;UPDATE RUN. ACTIVE DQ11 FOUND.
1068 002426 017737 176646 001506 MOV @CREAM,DQCSR ;PLACE ADDRESS OF DQ11 AT DQCSR
1069 002434 062737 000002 001300 ADD #2,CREAM ;UPDATE ADDRESS POINTER
1070 002442 017737 176632 001510 MOV @CREAM,DQSTAT ;PLACE STATUS OF DQ11 AT DQSTAT
1071 002450 062737 000002 001300 ADD #2,CREAM ;UPDATE ADDRESS POINTER
1072 002456 013737 001506 001360 MOV DQCSR,DQRCSR
1073 002464 013737 001510 001350 MOV DQSTAT,DQRVEC
1074 002472 042737 177007 001350 BIC #177007,DQRVEC
1075 002500 013737 001350 001352 MOV DQRVEC,DQRLVL ;GENERATE ADDRESS OF RECEIVER INTERRUPT SERVICE PS
1076 002506 062737 000002 001352 ADD #2,DQRLVL
1077 002514 013737 001352 001354 MOV DQRLVL,DQTVEC ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT VECTOR
1078 002522 062737 000002 001354 ADD #2,DQTVEC
1079 002530 013737 001354 001356 MOV DQTVEC,DQTLVL ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT SERVICE PS
1080 002536 062737 000002 001356 ADD #2,DQTLVL
1081 002544 013737 001360 001362 MOV DQRCSR,DQRCSH
1082 002552 005237 001362 INC DQRCSH ;GENERATE ADDRESS OF HIGH BYTE
1083 002556 013737 001360 001364 MOV DQRCSR,DQTCSR ;GENERATE ADDRESS OF TRANSMITTER CONTROL REGISTER
1084 002564 062737 000002 001364 ADD #2,DQTCSR
1085 002572 013737 001364 001366 MOV DQTCSR,DQERR ;GENERATE ADDRESS OF ERROR REGISTER
1086 002600 062737 000002 001366 ADD #2,DQERR
1087 002606 013737 001366 001370 MOV DQERR,DQREG ;GENERATE ADDRESS OF HIGH BYTE OF ERROR REGISTER
1088 002614 005237 001370 INC DQREG
1089 002620 013737 001370 001372 MOV DQREG,DQSEC ;GENERATE ADDRESS OF SECONDARY REGISTER
1090 002626 005237 001372 INC DQSEC
1091 002632 013737 001372 001374 MOV DQSEC,DQSECH ;GENERATE ADDRESS OF HIGH BYTE
1092 002640 005237 001374 INC DQSECH
1093 :
1094 :
1095 :TEST TO SEE IF TRANSMITTER ACTIVE
1096 :CAN SET.
1097 :AND IF IT DOES SET CHECK TO
1098 :SEE IF IT CAN BE CLEARED BY
1099 :MASTER CLEAR.
1100 :
1101 : TEST 2
1102 002644 012737 000002 001226 *****
1103 002652 012737 003002 001216 TST2: MOV #2,TSTNO
1104 002660 112777 000002 176502 MOVB #2,@DQREG ;SEL TX BA PRI
```

```
1105 002666 012777 014070 176476      MOV    #TMPBUF,@DQSEC ;LOAD TX BA
1106 002674 105277 176470              INCB   @DQREG          ;SEL TTX CC PRI
1107 002700 012777 000200 176464      MOV    #200,@DQSEC    ;LOAD WITH 200
1108 002706 112777 000012 176454      MOV    #MISC.,@DQREG  ;SEL MISC REGISTER
1109 002714 012777 004012 176450      MOV    #4012,@DQSEC   ;SELECT 8 BITS TEST LOOP AUTO STEP
1110 002722 005277 176436              INC    @DQTCR         ;SET TX GO
1111 002726 005277 176440              INC    @DQSEC         ;PRIM THE
1112 002732 005377 176434              DEC    @DQSEC         ;
1113 002736 005277 176430              INC    @DQSEC         ;      TRANSMITTER
1114 002742 032777 040000 176422      BIT    #BIT14,@DQSEC  ;CLOCK THE TRANSMITTER
1115 002750 001001                      BNE    .+4            ;CHECK TX ACTIVE.
1116 002752 104024                      HLT 24                ;BRANCH IF ACTIVE SET
1117 002754 104412                      MSTCLR                ;ERROR TX ACTIVE NOT SET!!
1118 002756 104412                      MSTCLR                ;ISSUE
1119 002760 112777 000012 176402      MOV    #MISC.,@DQREG  ;TWO MASTER CLEARS
1120 002766 032777 040000 176376      BIT    #BIT14,@DQSEC  ;RESELECT THE MISC REGISTER
1121 002774 001401                      BEQ    .+4            ;DID TX ACTIVE CLEAR BY MST CLR
1122 002776 104001                      HLT 1                 ;BRANCH IF ACTIVE CLEAR
1123 003000 104400                      SCOPE                 ;ERROR TX ACTIVE NOT CLEARED BY MST CLR
1124                                     ;SCOPE TEST
1125                                     ;ROUTINE TO SET
1126                                     ;TRANSMITTER POINTER
1127                                     ;CORRECTLY DEPENDING
1128                                     ;UPON THE NUMBER OF SYNC
1129                                     ;CHARACTERS.
1130
1131 003002 032737 100000 001510  CKSYN1: BIT    #SYNBIT,DQSTAT ;CHECK TO FIND OUT IF ONE SYNC OR TWO.
1132 003010 001004                      BNE    1$             ;BRANCH IF TWO SYNC CHARS REQUIRED
1133 003012 112737 000377 014524      MOV    #377,SYNC     ;CLEAR THE FIRST SYNC CHAR
1134 003020 000403                      BR     2$             ;BR TO LEAVE ROUTINE
1135 003022 112737 000026 014524  1$:  MOV    #26,SYNC    ;RESET SYNC CHAR TO 26
1136 003030 000240 2$:  NOP                    ;FALL IN TO NEXT TEST
1137
1138                                     ;
1139                                     ;TEST TO TRANSMITT ONE CHARACTER.
1140                                     ;
1141                                     ;TESTING TO MAKE SURE THAT THE
1142                                     ;CHARACTER COUNT INCREMENTS BY ONE.
1143                                     ;TESTING THAT THE CURRENT ADDRESS
1144                                     ;INCREMENTS BY ONE
1145                                     ;ALSO MAKING SURE THE PRI/SEC BIT SETS.
1146                                     ;
1147                                     ; TEST 3
1148 003032 012737 000003 001226  TST3: *****
1149 003040 012737 003056 001214      MOV    #3,TSTNO
1150 003046 012737 003372 001216      MOV    #A1$,RETURN
1151 003054 104413                      MOV    #TST4,NEXT
1152 003056 104412  A1$:  MEMCLR                ;CLEAR ALL THE DQ11
1153 003060 112777 000002 176302      MSTCLR
1154 003066 012777 014526 176276      MOV    #2,@DQREG     ;SELECT TX CURRENT ADD.
1155 003074 105277 176270              MOV    #TXBUFF,@DQSEC ;SET THE TX CURRENT ADD.
1156 003100 012777 177777 176264      INCB   @DQREG        ;SELECT THE TX CHAR CNT.
1157 003106 112777 000012 176254      MOV    #-1,@DQSEC    ;SET TX CHAR CNT FOR 1 CHARACTER.
1158 003114 012777 004010 176250      MOV    #MISC.,@DQREG ;SELECT THE MISC REGISTER.
1159 003122 005037 014062              MOV    #4010,@DQSEC  ;SET FOR EIGHT BITS. AND TEST LOOP
1160 003126 005277 176232              CLR    D:AY          ;CLEAR THE DELAY
1160                                     INC    @DQTCR        ;SET THE GO BIT AND GO..
```

::++E

DQ11 TRANSMITTER AND RECEIVER EXERCISER.

```

1161 003132 105777 176226      1$:  TSTB   @DQTCR      ;PRIMARY DONE??
1162 003136 100405              BMI    2$          ;BRANCH IF DONE
1163 003140 062737 000001 014062  ADD   #1,DELAY    ;STALL FOR DONE
1164 003146 001371              BNE   1$          ;TO SET.
1165 003150 104002              HLT  2          ;TX PRI DONE FAILED TO SET.
1166 003152 112777 000003 176210 2$:  MOVB  #3,@DQREG   ;SELECT TX CHAR CNT
1167 003160 005777 176206      TST   @DQSEC     ;MAKE SURE IT INCREMENTED
1168 003164 001401              BEQ   .+4        ;BY ONE TO ZERO.
1169 003166 104003              HLT  3          ;TX PRI CHAR CNT NOT ZERO.
1170 003170 112777 000002 176172  MOVB  #2,@DQREG   ;SELECT TX CURRENT ADD.(PRI)
1171 003176 022777 014527 176166 3$:  CMP   #TXBUFF+1,@DQSEC ;
1172 003204 001401              BEQ   .+4
1173 003206 104005              HLT  5          ; CHAR CNT NOT INC BY +1
1174 003210 032777 000004 176146 4$:  BIT   #BIT2,@DQTCR ;DID PRI/SEC SET?
1175 003216 001001              BNE   .+4
1176 003220 104006              HLT  6          ;TX PRI/SEC NOT SET.
1177
1178 ;
1179 ;TEST THAT WITH A CHARACTER
1180 ;COUNT THAT IS EVEN THAT THE
1181 ;CURRENT ADDRESS INCREMENTS BY +2
1182 ;AND THAT THE CHAR CNT GOES TO ZERO.
1183 ;
1184
1185 003222 112777 000006 176140 SECND: MOVB  #6,@DQREG   ;SELECT TX CURRENT ADD.
1186 003230 012777 014526 176134  MOV   #TXBUFF,@DQSEC ;SET THE TX CURRENT ADD.
1187 003236 105277 176126      INCB  @DQREG     ;SELECT THE TX CHAR CNT.
1188 003242 012777 177776 176122  MOV   #-2,@DQSEC   ;SET TX CHAR CNT FOR TWO CHARS.
1189 003250 112777 000012 176112  MOVB  #MISC,@DQREG ;SELECT THE MISC REGISTER.
1190 003256 012777 004010 176106  MOV   #4010,@DQSEC ;SET FOR EIGHT BITS AND TEST LOOP
1191 003264 005037 014062      CLR  DELAY      ;CLEAR THE DELAY
1192 003270 005277 176070      INC  @DQTCR     ;SET THE GO BIT AND GO!!
1193 003274 032777 000100 176062 1$:  BIT   #BIT6,@DQTCR ;SECONDARY DONE??
1194 003302 001005              BNE   2$          ;BRANCH IF DONE
1195 003304 062737 000001 014062  ADD   #1,DELAY    ;STALL FOR DONE
1196 003312 001370              BNE   1$          ;TO SET.
1197 003314 104002              HLT  2          ;TX SEC DONE FAILED TO SET.
1198 003316 112777 000007 176044 2$:  MOVB  #7,@DQREG   ;SELECT TX CHAR CNT
1199 003324 005777 176042      TST   @DQSEC     ;MAKE SURE IT INCREMENTED
1200 003330 001401              BEQ   .+4        ;BY ONE TO ZERO.
1201 003332 104003              HLT  3          ;TX SEC CHAR CNT NOT ZERO.
1202 003334 112777 000006 176026  MOVB  #6,@DQREG   ;SELECT TX CURRENT ADD.(PRI)
1203 003342 022777 014530 176022 3$:  CMP   #TXBUFF+2,@DQSEC ;
1204 003350 001401              BEQ   .+4
1205 003352 104004              HLT  4          ; CHAR CNT NOT INC BY +2
1206 003354 032777 000004 176002 4$:  BIT   #BIT2,@DQTCR ;DID PRI/SEC SET?
1207 003362 001401              BEQ   .+4
1208 003364 104006              HLT  6          ;TX PRI/SEC NOT SET.
1209 003366 104413      MEMCLR
1210 003370 104400      SCOPE
1211 ;
1212 ;TRANSMITTER CHARACTER LENGTH TESTS.
1213 ;
1214 ;TEST TO TRANSMITT A CHARACTER
1215 ; 2 BITS LONG MAKING SURE THAT
1216 ;THE CHARACTER IS ALL ZERO'S

```

```
1217                                     ;AND THAT THE TX LINE GOES BACK TO
1218                                     ;A MARK STATE WHEN DONE.
1219                                     :
1220
1221                                     : TEST 4
1222                                     :*****
1223 003372 012737 000004 001226 TST4:  MOV    #4,TSTNO
1224 003400 012737 003420 001216      MOV    #TST5,NEXT
1225 003406 004537 010722              JSR    R5,TXSTRB      ;JSR TO ROUTINE
1226 003412 000002                          2      ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1227 003414 007000                          7000   ;BIT SELECTION TO BE PLACED INTO MISC REG
1228 003416 104400                          SCOPE  ;SCOPE TEST
1229
1230                                     ;TEST TO TRANSMITT A CHARACTER
1231                                     ; 3 BITS LONG MAKING SURE THAT
1232                                     ;THE CHARACTER IS ALL ZERO'S
1233                                     ;AND THAT THE TX LINE GOES BACK TO
1234                                     ;A MARK STATE WHEN DONE.
1235                                     :
1236
1237                                     : TEST 5
1238                                     :*****
1239 003420 012737 000005 001226 TST5:  MOV    #5,TSTNO
1240 003426 012737 003446 001216      MOV    #TST6,NEXT
1241 003434 004537 010722              JSR    R5,TXSTRB      ;JSR TO ROUTINE
1242 003440 000003                          3      ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1243 003442 006400                          6400   ;BIT SELECTION TO BE PLACED INTO MISC REG
1244 003444 104400                          SCOPE  ;SCOPE TEST
1245
1246                                     ;TEST TO TRANSMITT A CHARACTER
1247                                     ; 4 BITS LONG MAKING SURE THAT
1248                                     ;THE CHARACTER IS ALL ZERO'S
1249                                     ;AND THAT THE TX LINE GOES BACK TO
1250                                     ;A MARK STATE WHEN DONE.
1251                                     :
1252
1253                                     : TEST 6
1254                                     :*****
1255 003446 012737 000006 001226 TST6:  MOV    #6,TSTNO
1256 003454 012737 003474 001216      MOV    #TST7,NEXT
1257 003462 004537 010722              JSR    R5,TXSTRB      ;JSR TO ROUTINE
1258 003466 000004                          4      ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1259 003470 006000                          6000   ;BIT SELECTION TO BE PLACED INTO MISC REG
1260 003472 104400                          SCOPE  ;SCOPE TEST
1261
1262                                     ;TEST TO TRANSMITT A CHARACTER
1263                                     ; 5 BITS LONG MAKING SURE THAT
1264                                     ;THE CHARACTER IS ALL ZERO'S
1265                                     ;AND THAT THE TX LINE GOES BACK TO
1266                                     ;A MARK STATE WHEN DONE.
1267                                     :
1268
1269                                     : TEST 7
1270                                     :*****
1271 003474 012737 000007 001226 TST7:  MOV    #7,TSTNO
1272 003502 012737 003522 001216      MOV    #TST10,NEXT
```

```
1273 003510 004537 010722 JSR R5,TXSTRB ;JSR TO ROUTINE
1274 003514 000005 5 ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1275 003516 005400 5400 ;BIT SELECTION TO BE PLACED INTO MISC REG
1276 003520 104400 SCOPE ;SCOPE TEST
1277
1278 ;TEST TO TRANSMITT A CHARACTER
1279 ; 6 BITS LONG MAKING SURE THAT
1280 ;THE CHARACTER IS ALL ZERO'S
1281 ;AND THAT THE TX LINE GOES BACK TO
1282 ;A MARK STATE WHEN DONE.
1283
1284
1285 ; TEST 10
1286 ;*****
1287 003522 012737 000010 001226 TST10: MOV #10,TSTNO
1288 003530 012737 003550 001216 MOV #TST11,NEXT
1289 003536 004537 010722 JSR R5,TXSTRB ;JSR TO ROUTINE
1290 003542 000006 6 ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1291 003544 005000 5000 ;BIT SELECTION TO BE PLACED INTO MISC REG
1292 003546 104400 SCOPE ;SCOPE TEST
1293
1294 ;TEST TO TRANSMITT A CHARACTER
1295 ; 7 BITS LONG MAKING SURE THAT
1296 ;THE CHARACTER IS ALL ZERO'S
1297 ;AND THAT THE TX LINE GOES BACK TO
1298 ;A MARK STATE WHEN DONE.
1299
1300
1301 ; TEST 11
1302 ;*****
1303 003550 012737 000011 001226 TST11: MOV #11,TSTNO
1304 003556 012737 003576 001216 MOV #TST12,NEXT
1305 003564 004537 010722 JSR R5,TXSTRB ;JSR TO ROUTINE
1306 003570 000007 7 ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1307 003572 004400 4400 ;BIT SELECTION TO BE PLACED INTO MISC REG
1308 003574 104400 SCOPE ;SCOPE TEST
1309
1310 ;TEST TO TRANSMITT A CHARACTER
1311 ; 8 BITS LONG MAKING SURE THAT
1312 ;THE CHARACTER IS ALL ZERO'S
1313 ;AND THAT THE TX LINE GOES BACK TO
1314 ;A MARK STATE WHEN DONE.
1315
1316
1317 ; TEST 12
1318 ;*****
1319 003576 012737 000012 001226 TST12: MOV #12,TSTNO
1320 003604 012737 003624 001216 MOV #TST13,NEXT
1321 003612 004537 010722 JSR R5,TXSTRB ;JSR TO ROUTINE
1322 003616 000010 8 ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1323 003620 004000 4000 ;BIT SELECTION TO BE PLACED INTO MISC REG
1324 003622 104400 SCOPE ;SCOPE TEST
1325
1326
1327 ;TEST OF CHARACTER LENGTH
1328 ;FOR CHARACTERS OVER 8 BITS LONG.
```

```
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340 003624 012737 000013 001226  
1341 003632 012737 003652 001216  
1342 003640 004537 010722  
1343 003644 000011  
1344 003646 003400  
1345 003650 104400  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357 003652 012737 000014 001226  
1358 003660 012737 003700 001216  
1359 003666 004537 010722  
1360 003672 000012  
1361 003674 003000  
1362 003676 104400  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374 003700 012737 000015 001226  
1375 003706 012737 003726 001216  
1376 003714 004537 010722  
1377 003720 000013  
1378 003722 002400  
1379 003724 104400  
1380  
1381  
1382  
1383  
1384
```

TEST 13

TST13: MOV #13,TSTNO
MOV #TST14,NEXT
JSR R5, TXSTRB ;DO JSR TO THE SUBROUTINE
9. ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
3400 ;BIT SELECTION TO BE PLACED INTO MISC REG
SCOPE ;SCOPE THE TEST

TEST 14

TST14: MOV #14,TSTNO
MOV #TST15,NEXT
JSR R5, TXSTRB ;DO JSR TO THE SUBROUTINE
10. ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
3000 ;BIT SELECTION TO BE PLACED INTO MISC REG
SCOPE ;SCOPE THE TEST

TEST 15

TST15: MOV #15,TSTNO
MOV #TST16,NEXT
JSR R5, TXSTRB ;DO JSR TO THE SUBROUTINE
11. ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
2400 ;BIT SELECTION TO BE PLACED INTO MISC REG
SCOPE ;SCOPE THE TEST

TEST TO TRANSMITT A CHARACTER
: 9 BITS LONG MAKING SURE THAT
: THE CHARACTER IS ALL ZERO'S
: AND THAT THE TX LINE GOES BACK TO
: A MARK STATE WHEN DONE.

TEST TO TRANSMITT A CHARACTER
: 10 BITS LONG MAKING SURE THAT
: THE CHARACTER IS ALL ZERO'S
: AND THAT THE TX LINE GOES BACK TO
: A MARK STATE WHEN DONE.

TEST TO TRANSMITT A CHARACTER
: 11 BITS LONG MAKING SURE THAT
: THE CHARACTER IS ALL ZERO'S
: AND THAT THE TX LINE GOES BACK TO
: A MARK STATE WHEN DONE.

TEST TO TRANSMITT A CHARACTER
: 12 BITS LONG MAKING SURE THAT
: THE CHARACTER IS ALL ZERO'S

```
1385                                     ;AND THAT THE TX LINE GOES BACK TO
1386                                     ;A MARK STATE WHEN DONE.
1387                                     ;
1388
1389                                     ; TEST 16
1390                                     ;*****
1391 003726 012737 000016 001226 TST16: MOV #16,TSTNO
1392 003734 012737 003754 001216      MOV #TST17,NEXT
1393 003742 004537 010722      JSR R5,TXSTRB      ;DO JSR TO THE SUBROUTINE
1394 003746 000014      12.                ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1395 003750 002000      2000              ;BIT SELECTION TO BE PLACED INTO MISC REG
1396 003752 104400      SCOPE                ;SCOPE THE TEST
1397
1398
1399                                     ;
1400                                     ;TEST TO TRANSMITT A CHARACTER
1401                                     ; 13 BITS LONG MAKING SURE THAT
1402                                     ;THE CHARACTER IS ALL ZERO'S
1403                                     ;AND THAT THE TX LINE GOES BACK TO
1404                                     ;A MARK STATE WHEN DONE.
1405                                     ;
1406
1407                                     ; TEST 17
1408                                     ;*****
1409 003754 012737 000017 001226 TST17: MOV #17,TSTNO
1410 003762 012737 004002 001216      MOV #TST20,NEXT
1411 003770 004537 010722      JSR R5,TXSTRB      ;DO JSR TO THE SUBROUTINE
1412 003774 000015      13.                ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1413 003776 001400      1400              ;BIT SELECTION TO BE PLACED INTO MISC REG
1414 004000 104400      SCOPE                ;SCOPE THE TEST
1415
1416                                     ;
1417                                     ;TEST TO TRANSMITT A CHARACTER
1418                                     ; 14 BITS LONG MAKING SURE THAT
1419                                     ;THE CHARACTER IS ALL ZERO'S
1420                                     ;AND THAT THE TX LINE GOES BACK TO
1421                                     ;A MARK STATE WHEN DONE.
1422                                     ;
1423
1424                                     ; TEST 20
1425                                     ;*****
1426 004002 012737 000020 001226 TST20: MOV #20,TSTNO
1427 004010 012737 004030 001216      MOV #TST21,NEXT
1428 004016 004537 010722      JSR R5,TXSTRB      ;DO JSR TO THE SUBROUTINE
1429 004022 000016      14.                ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1430 004024 001000      1000              ;BIT SELECTION TO BE PLACED INTO MISC REG
1431 004026 104400      SCOPE                ;SCOPE THE TEST
1432
1433                                     ;
1434                                     ;TEST TO TRANSMITT A CHARACTER
1435                                     ; 15 BITS LONG MAKING SURE THAT
1436                                     ;THE CHARACTER IS ALL ZERO'S
1437                                     ;AND THAT THE TX LINE GOES BACK TO
1438                                     ;A MARK STATE WHEN DONE.
1439                                     ;
1440                                     ; TEST 21
```

DQ11 TRANSMITTER AND RECEIVER EXERCISER.

```
1441
1442 004030 012737 000021 001226 *****
1443 004036 012737 004056 001216 TST21: MOV #21,TSTNO
1444 004044 004537 010722 MOV #TST22,NEXT
1445 004050 000017 JSR R5, TXSTRB ;DO JSR TO THE SUBROUTINE
1446 004052 000400 15. ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1447 004054 104400 400 ;BIT SELECTION TO BE PLACED INTO MISC REG
1448 SCOPE ;SCOPE THE TEST
1449
1450 ;
1451 ;TEST TO TRANSMITT A CHARACTER
1452 ; 16 BITS LONG MAKING SURE THAT
1453 ;THE CHARACTER IS ALL ZERO'S
1454 ;AND THAT THE TX LINE GOES BACK TO
1455 ;A MARK STATE WHEN DONE.
1456 ;
1457 ;
1458 ; TEST 22
1459 004056 012737 000022 001226 *****
1460 004064 012737 004104 001216 TST22: MOV #22,TSTNO
1461 004072 004537 010722 MOV #TST23,NEXT
1462 004076 000020 JSR R5, TXSTRB ;DO JSR TO THE SUBROUTINE
1463 004100 000000 0 ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1464 004102 104400 SCOPE ;BIT SELECTION TO BE PLACED INTO MISC REG
1465 ;SCOPE THE TEST
1466 ;
1467 ;
1468 ;
1469 ;
1470 ;TEST OF TRANSMITTER IDLE SYNC
1471 ;TEST THAT THE TRANSMITTER CAN
1472 ;REALLY IDLE SYNC CHARACTERS
1473 ;
1474 ;
1475 ; TEST 23
1476 004104 012737 000023 001226 *****
1477 004112 012737 004440 001216 TST23: MOV #23,TSTNO
1478 004120 005077 175240 001216 MOV #TST24,NEXT
1479 004124 032777 000002 175232 CLR @DQTCR ;CLR TX STATUS
1480 004132 001401 BIT #BIT1,@DQTCR ;IDLE SET?
1481 004134 104000 BEQ .+4
1482 004136 052777 000002 175220 HLT ;IDLE SHOULD NOT BE SET!
1483 004144 032777 000002 175212 BIS #BIT1,@DQTCR ;SET IDLE BIT
1484 004152 001001 BIT #BIT1,@DQTCR ;IS IDLE SET?
1485 004154 104000 BNE .+4 ;BR IF SET.
1486 004156 042777 000002 175200 HLT ;IDLE BIT SHOULD BE SET.
1487 004164 032777 000002 175172 BIC #BIT1,@DQTCR ;CLEAR IDLE BIT.
1488 004172 001401 BIT #BIT1,@DQTCR ;IS IDLE BIT SET?
1489 004174 104000 BEQ .+4 ;BR IF CLEAR.
1490 004176 052777 000002 175160 HLT ;IDLE BIT NOT CLEARED.
1491 004204 104412 BIS #BIT1,@DQTCR ;SET IDLE
1492 004206 032777 000002 175150 MSTCLR
1493 004214 001401 BIT #BIT1,@DQTCR ;IS IDLE SET?
1494 004216 104000 BEQ .+4
1495 004220 012737 000005 001250 HLT ;IDLE BIT NOT CLEARED BY INIT.
1496 MOV #5,TEMP3
```

```
1497 004226 012737 000377 014060 1$: MOV #377,WORD
1498 004234 112777 000011 175126 MOVB #11,@DQREG
1499 004242 013777 014522 175122 MOV .SYNC,@DQSEC
1500 004250 012737 000010 014064 MOV #10,COUNT ;PICK UP THE NUMBER OF SHIFTS
1501 004256 012737 004000 014066 MOV #4000,BITSEL ;PICK UP NUMBER OF BIT PER CHAR.
1502 004264 112777 000002 175076 MOVB #2,@DQREG ;SELECT THE TRANSMITTER BA PRI.
1503 004272 012777 014060 175072 MOV #WORD,@DQSEC ;LOAD THE BA
1504 004300 105277 175064 INCB @DQREG ;SELECT THE TRANSMITTER CC PRI.
1505 004304 012777 177777 175060 MOV #-1,@DQSEC ;LOAD THE CC WITH -1
1506 004312 112777 000012 175050 MOVB #MISC.,@DQREG ;SELECT THE MISC REGISTER.
1507 004320 053777 014066 175044 BIS BITSEL,@DQSEC ;LOAD MISC REG WITH NUMBER OF BITS PER CHAR.
1508 004326 052777 000012 175036 BIS #12,@DQSEC ;ADD TO THAT TEST LOOP AND AUTO STEP.
1509 004334 052777 000002 175022 BIS #BIT1,@DQTCSR ;SET TRANSMITTER IDLE MODE.
1510 004342 005037 001252 CLR TEMP4
1511 004346 006037 001252 2$: ROR TEMP4 ;SHIFT THE STORAGE OF DATA FROM THE TRANSMITTER.
1512 004352 005277 175014 INC @DQSEC ;CLOCK THE TRANSMITTER -UP-
1513 004356 005377 175010 DEC @DQSEC ;CLOCK THE TRANSMITTER -DOWN-
1514 004362 017702 175004 MOV @DQSEC,R2 ;MOVE THE MISC REG TO R2
1515 004366 042702 177577 BIC #177577,R2 ;CLEAR ALL BUT THE BIT WINDOW.
1516 004372 050237 001252 3$: BIS R2,TEMP4 ;PLACE DATA INTO TEMPORY LOCATION
1517 004376 005337 014064 DEC COUNT ;IS CHARACTER COMPLETELY SHIFTED OUT?
1518 004402 001361 BNE 2$ ;BRANCH IF MORE BITS TO GO.
1519 004404 005137 001252 COM TEMP4 ;COMPLIMENT DATA STORAGE
1520 004410 012737 000026 001254 MOV #26,TEMP5
1521 004416 123737 001254 001252 CMPB TEMP5,TEMP4
1522 004424 001401 BEQ .+4
1523 004426 104012 HLT 12
1524 004430 005337 001250 DEC TEMP3
1525 004434 001274 BNE 1$
1526 004436 104400 SCOPE
1527
1528
1529
1530
1531
1532
1533
1534 ;
1535 ;TRANSMITTER DATA REALIBILITY TEST.
1536 ;TEST TO TRANSMITT AN EIGHT
1537 ;BIT BINARY COUNT PATTERN (000-377)
1538 ;
1539 ;NOTE THIS TEST IS FOR UP TO EIGHT BITS PER CHARACTER.
1540 ;PARITY WILL BE ENABLED WHEN 'PARFLG' IS NON-ZERO
1541 ;
1542 ;
1543 ; TEST 24
1544 004440 012737 000024 001226 *****
1545 004446 012737 004572 001216 TST24: MOV #24,TSTNO
1546 004454 012737 004474 001220 MOV #TST25,NEXT
1547 004462 105037 J12604 MOV #2$,LOCK
1548 004466 005000 1$: CLRB PARFLG ;SET DATA TO ZERO
1549 004470 005037 014054 CLR R0 ;TELL SUBROUTINE THIS IS FOR EIGHT BITS
1550 004474 010037 014060 2$: MOV R0,WORD ;PLACE DATA FOR WORK.
1551 004500 005037 001252 CLR TEMP4 ;CCLEAR WHERE CHAR IS TO BE STORED
1552 004504 104412 MSTCLR ;MASTER CLEAR
```

```
1553 004506 004537 011270 JSR R5,TXSTRD ;GO TO ROUTINE
1554 004512 000010 8. ;NUMBER OF SHIFTS REQUIRED
1555 004514 004000 4000 ;EIGHT BITS
1556 004516 105737 012604 TSTB PARFLG
1557 004522 001402 BEQ .+6
1558 004524 004737 012446 JSR PC,GENPAR
1559 004530 013737 014060 001254 MOV WORD,TEMP5 ;STORE GOOD CHARACTER
1560 004536 123737 001254 001252 CMPB TEMP5,TEMP4 ;COMPARE GOOD CHAR TO TX CHAR
1561 004544 001401 BEQ .+4 ;BR IF SAME
1562 004546 104012 HLT 12 ;DATA COMPARISON ERROR
1563 004550 104401 SCOP1 ;DOES USER WANT TO LOCK ON THIS CHAR?
1564 004552 105200 INCB R0 ;UPDATE GOOD CHARACTER
1565 004554 001347 BNE 2$ ;IF NOT ALL CHARACTERS GO DO AGAIN
1566 004556 012700 000200 MOV #200,R0
1567 004562 105137 012604 COMB PARFLG
1568 004566 001342 BNE 2$
1569 004570 104400 SCOPE ;SCOPE THIS TEST
1570
1571 ;
1572 ;TRANSMITTER DATA REALIBILITY TEST
1573 ;TEST TO TRANSMITT AN EIGHT BIT
1574 ;BINARY COUNT PATTERN (000400-177400)
1575 ;
1576 ;PARITY WILL BE ENABLED WHEN 'PARFLG' IS NON-ZERO
1577 ;NOTE THIS IS FOR 16 BITS PER CHAR. (LOW BYTE IS=0; THE HIGH BYTE =BINARY COUNT.
1578
1579 : TEST 25
1580 004572 012737 000025 001226 *****
1581 004600 012737 004732 001216 TST25: MOV #25,TSTNO
1582 004606 012737 004630 001220 MOV #TST26,NEXT
1583 004614 112737 000377 014054 MOV #2$,LOCK
1584 004622 105037 012604 MOVB #377,EXTFLG ;TELL SUBROUTINE THIS IS FOR 16 BITS PER CHAR
1585 004626 005000 CLRB PARFLG ;NO PARITY CHECKING NOW
1586 004630 010037 014060 1$: CLR R0 ;ZERO DATA POINTER
1587 004634 000337 014060 2$: MOV R0,WORD ;PREPARE DATA FOR SUBROUTINE
1588 004640 005037 001252 SWAB WORD ;PUT DATA IN HIGH BYTE
1589 004644 104412 CLR TEMP4 ;ZERO STORE AREA
1590 004646 004537 011270 MSTCLR ;INIT DQ11
1591 004652 000020 JSR R5,TXSTRD ;GOTO SUBROUTINE
1592 004654 000000 16. ;THIS IS NUMBER OF SHIFTS.
1593 004656 105737 012604 0 ;THIS IS BITS/PER/CHARACTER SELECT
1594 004662 001402 TSTB PARFLG ;IS PARITY ENABLED?
1595 004664 004737 012446 BEQ .+6 ;BR IF NOT ENABLED
1596 004670 013737 014060 001254 JSR PC,GENPAR ;GO CALCULATE THE PARITY
1597 004676 023737 001254 001252 MOV WORD,TEMP5 ;STORE THE CHARACTER
1598 004704 001401 CMP TEMP5,TEMP4 ;IS THE CHARACTER CORRECT
1599 004706 104012 BEQ .+4 ;BR IF GOOD
1600 004710 104401 HLT 12 ;DATA COMPARISON ERROR.
1601 004712 105200 SCOP1 ;LOCK ON DATA? (SW09=1)
1602 004714 001345 INCB R0 ;UPDATE DATA POINTER
1603 004716 012700 000200 BNE 2$ ;BR IF MORE TO GO
1604 004722 105137 012604 MOV #200,R0
1605 004726 001340 COMB PARFLG ;NOW ENABLE THE PARITY TEST.
1606 004730 104400 BNE 2$ ;BR IF FIRST TIME FOR PARITY
SCOPE ;SCOPE THE TEST.
```



```
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662
```

```
      :  
      :RECEIVER CHARACTER LENGTH TEST  
      :TEST THAT ALL CHARACTER  
      :LENGTHS WORK CORRECTLY.  
      :  
      :TEST OF RX CHARACTER LENGTH 2 BITS LONG.  
      :  
      : TEST 26  
      :*****  
TST26: MOV      #26,TSTNO  
      MOV      #TST27,NEXT  
      JSR      R5,RXLNG      :GOTO JSR SUBROUTINE  
      7000      :CHARACTER EXPECTED TO FIND  
      2        :BITS/PER/CHAR TO BE PLACED INTO MISC REG  
      SCOPE    :SCOPE THIS TEST  
      :  
      :TEST OF RX CHARACTER LENGTH 3 BITS LONG.  
      :  
      : TEST 27  
      :*****  
TST27: MOV      #27,TSTNO  
      MOV      #TST30,NEXT  
      JSR      R5,RXLNG      :GOTO JSR SUBROUTINE  
      6400      :CHARACTER EXPECTED TO FIND  
      4        :BITS/PER/CHAR TO BE PLACED INTO MISC REG  
      SCOPE    :SCOPE THIS TEST  
      :  
      :TEST OF RX CHARACTER LENGTH 4 BITS LONG.  
      :  
      : TEST 30  
      :*****  
TST30: MOV      #30,TSTNO  
      MOV      #TST31,NEXT  
      JSR      R5,RXLNG      :GOTO JSR SUBROUTINE  
      6000      :CHARACTER EXPECTED TO FIND  
      10       :BITS/PER/CHAR TO BE PLACED INTO MISC REG  
      SCOPE    :SCOPE THIS TEST  
      :  
      :TEST OF RX CHARACTER LENGTH 5 BITS LONG.  
      :  
      : TEST 31  
      :*****  
TST31: MOV      #31,TSTNO  
      MOV      #TST32,NEXT  
      JSR      R5,RXLNG      :GOTO JSR SUBROUTINE  
      5400      :CHARACTER EXPECTED TO FIND  
      20       :BITS/PER/CHAR TO BE PLACED INTO MISC REG  
      SCOPE    :SCOPE THIS TEST  
      :  
      :TEST OF RX CHARACTER LENGTH 6 BITS LONG.  
      :  
      : TEST 32  
      :*****  
TST32: MOV      #32,TSTNO  
      MOV      #TST33,NEXT
```

```

1663 005076 004537 012134          JSR    R5,RXLNG          ;GOTO JSR SUBROUTINE
1664 005102 005000                    5000          ;CHARACTER EXPECTED TO FIND
1665 005104 000040                    40           ;BITS/PER/CHAR TO BE PLACED INTO MISC REG
1666 005106 104400                    SCOPE        ;SCOPE THIS TEST
1667
1668                                ;TEST OF RX CHARACTER LENGTH 7 BITS LONG.
1669
1670                                ;
1671                                ; TEST 33
1672 005110 012737 000033 001226  *****
1673 005116 012737 005136 001216  TST33: MOV    #33,TSTNO
1674 005124 004537 012134          JSR    R5,RXLNG          ;GOTO JSR SUBROUTINE
1675 005130 004400                    4400          ;CHARACTER EXPECTED TO FIND
1676 005132 000100                    100          ;BITS/PER/CHAR TO BE PLACED INTO MISC REG
1677 005134 104400                    SCOPE        ;SCOPE THIS TEST
1678
1679                                ;TEST OF RX CHARACTER LENGTH 8 BITS LONG.
1680
1681                                ;
1682                                ; TEST 34
1683 005136 012737 000034 001226  *****
1684 005144 012737 005164 001216  TST34: MOV    #34,TSTNO
1685 005152 004537 012134          JSR    R5,RXLNG          ;GOTO JSR SUBROUTINE
1686 005156 004000                    4000          ;CHARACTER EXPECTED TO FIND
1687 005160 000200                    200          ;BITS/PER/CHAR TO BE PLACED INTO MISC REG
1688 005162 104400                    SCOPE        ;SCOPE THIS TEST
1689
1690                                ;
1691                                ;RECEIVER CHARACTER LENGTH TEST
1692                                ;FOR CHARACTERS OVER EIGHT BITS LONG.
1693
1694                                ;
1695                                ;TEST OF CHARACTER LENGTH 9 BITS LONG.
1696
1697                                ;
1698                                ; TEST 35
1699 005164 012737 000035 001226  *****
1700 005172 012737 005212 001216  TST35: MOV    #35,TSTNO
1701 005200 004537 012304          JSR    R5,RXLNG          ;GOTO SUBROUTINE
1702 005204 003400                    3400          ;CHARACTER EXPECTED TO BE FOUND
1703 005206 000400                    400          ;BITS/PER/CHAR TO BE PLACED INTO MISC REG
1704 005210 104400                    SCOPE        ;SCOPE THIS TEST
1705
1706                                ;TEST OF CHARACTER LENGTH 10 BITS LONG.
1707
1708                                ;
1709                                ; TEST 36
1710 005212 012737 000036 001226  *****
1711 005220 012737 005240 001216  TST36: MOV    #36,TSTNO
1712 005226 004537 012304          JSR    R5,RXLNG          ;GOTO SUBROUTINE
1713 005232 003000                    3000          ;CHARACTER EXPECTED TO BE FOUND
1714 005234 001000                    1000          ;BITS/PER/CHAR TO BE PLACED INTO MISC REG
1715 005236 104400                    SCOPE        ;SCOPE THIS TEST
1716
1717                                ;TEST OF CHARACTER LENGTH 11 BITS LONG.
1718

```

```

1719      ; TEST 37
1720      ;*****
1721 005240 012737 000037 001226 TST37: MOV #37,TSTNO
1722 005246 012737 005266 001216      MOV #TST40,NEXT
1723 005254 004537 012304      JSR R5,RXELNG      ;GOTO SUBROUTINE
1724 005260 002400      2400      ;CHARACTER EXPECTED TO BE FOUND
1725 005262 002000      2000      ;BITS/PER/CHAR TO BE PLACED INTO MISC REG
1726 005264 104400      SCOPE      ;SCOPE THIS TEST
1727
1728      ;TEST OF CHARACTER LENGTH 12 BITS LONG.
1729      ;
1730      ; TEST 40
1731      ;*****
1732 005266 012737 000040 001226 TST40: MOV #40,TSTNO
1733 005274 012737 005314 001216      MOV #TST41,NEXT
1734 005302 004537 012304      JSR R5,RXELNG      ;GOTO SUBROUTINE
1735 005306 002000      2000      ;CHARACTER EXPECTED TO BE FOUND
1736 005310 004000      4000      ;BITS/PER/CHAR TO BE PLACED INTO MISC REG
1737 005312 104400      SCOPE      ;SCOPE THIS TEST
1738
1739      ;TEST OF CHARACTER LENGTH 13 BITS LONG.
1740      ;
1741      ; TEST 41
1742      ;*****
1743 005314 012737 000041 001226 TST41: MOV #41,TSTNO
1744 005322 012737 005342 001216      MOV #TST42,NEXT
1745 005330 004537 012304      JSR R5,RXELNG      ;GOTO SUBROUTINE
1746 005334 001400      1400      ;CHARACTER EXPECTED TO BE FOUND
1747 005336 010000      10000     ;BITS/PER/CHAR TO BE PLACED INTO MISC REG
1748 005340 104400      SCOPE      ;SCOPE THIS TEST
1749
1750      ;TEST OF CHARACTER LENGTH 14 BITS LONG.
1751      ;
1752      ; TEST 42
1753      ;*****
1754 005342 012737 000042 001226 TST42: MOV #42,TSTNO
1755 005350 012737 005370 001216      MOV #TST43,NEXT
1756 005356 004537 012304      JSR R5,RXELNG      ;GOTO SUBROUTINE
1757 005362 001000      1000      ;CHARACTER EXPECTED TO BE FOUND
1758 005364 020000      20000     ;BITS/PER/CHAR TO BE PLACED INTO MISC REG
1759 005366 104400      SCOPE      ;SCOPE THIS TEST
1760
1761      ;TEST OF CHARACTER LENGTH 15 BITS LONG.
1762      ;
1763      ; TEST 43
1764      ;*****
1765 005370 012737 000043 001226 TST43: MOV #43,TSTNO
1766 005376 012737 005416 001216      MOV #TST44,NEXT
1767 005404 004537 012304      JSR R5,RXELNG      ;GOTO SUBROUTINE
1768 005410 000400      400       ;CHARACTER EXPECTED TO BE FOUND
1769 005412 040000      40000     ;BITS/PER/CHAR TO BE PLACED INTO MISC REG
1770 005414 104400      SCOPE      ;SCOPE THIS TEST
1771
1772      ;TEST OF CHARACTER LENGTH 16 BITS LONG.
1773      ;
1774      ; TEST 44

```

```
1775  
1776 005416 012737 000044 001226 :*****  
1777 005424 012737 005444 001216 TST44: MOV #44,TSTNO  
1778 005432 004537 012304 JSR R5,RXELNG ;GOTO SUBROUTINE  
1779 005436 000000 0 ;CHARACTER EXPECTED TO BE FOUND  
1780 005440 100000 100000 ;BITS/PER/CHAR TO BE PLACED INTO MISC REG  
1781 005442 104400 SCOPE ;SCOPE THIS TEST  
1782  
1783  
1784 ;  
1785 ;TEST THAT SYNC1 AND SYNC2  
1786 ;SET WHEN RECEIVER ACTIVE SET  
1787 ;AND IF THEY DO THE TEST THAT THEY  
1788 ;CLEAR BY MASTER CLEAR.  
1789  
1790 : TEST 45  
1791 005444 012737 000045 001226 :*****  
1792 005452 012737 005550 001216 TST45: MOV #45,TSTNO  
1793 005460 112777 000012 173702 MOV #TST46,NEXT  
1794 005466 012777 000012 173676 MOV #MISC,@DQREG ;SELECT THE MISC REGISTER  
1795 005474 052777 010000 173656 MOV #12,@DQSEC ;SET TEST LOOP AND AUTO/STEP  
1796 005502 017700 173664 BIS #BIT12,@DQRCR ;SET RX ACTIVE  
1797 005506 042700 147777 MOV @DQSEC,R0 ;READ THE DQSEC  
1798 005512 022700 030000 BIC #147777,R0 ;CLEAR ALL BUT SYNC 1 AND SYNC 2  
1799 005516 001401 BEQ .+4 ;DID BOTH OF THEM SET?  
1800 005520 104016 HLT 16 ;BR IF GOOD  
1801 005522 052777 000040 173642 BIS #BIT5,@DQSEC ;SYNC 1 AND SYNC 2 NOT SET.  
1802 005530 112777 000012 173632 MOV #MISC,@DQREG ;SET MASTER CLEAR  
1803 005536 005777 173630 TST @DQSEC ;RESELECT THE MISC REGISTER  
1804 005542 001401 BEQ .+4 ;IS THE DQSEC =0  
1805 005544 104017 HLT 17 ;BR IF YES  
1806 005546 104400 SCOPE ;DQSEC NOT=0  
1807 ;SCOPE THIS TEST.  
1808  
1809  
1810 ;  
1811 ;SYNC TESTS.  
1812 ;TEST THAT RECEIVER ACTIVE AND SYNC 1 AND SYNC 2  
1813 ;ASSERT AT THE PROPER TIME.  
1814 ;TEST INVOLVES BOTH SYNCING AN AN EIGHT BIT CHAR  
1815 ;AND A SIXTEEN BIT CHAR.  
1816  
1817 ;LOOK AT LOCATION 'WORD'  
1818 ;IF 'WORD IS EQUAL TO 377 THE THE EIGHT  
1819 ;BIT PER CHAR IS BEING EXECUTED.  
1820 ;IF 'WORD' IS EQUAL TO 177777 THEN THE SYXTEEN  
1821 ;BIT PER CHAR IS BEING EXECUTED.  
1822  
1823 : TEST 46  
1824 005550 012737 000046 001226 :*****  
1825 005556 012737 005600 001216 TST46: MOV #46,TSTNO  
1826 005564 004537 005630 JSR R5,SYNST ;GOTO THE ACTUAL TEST.  
1827 005570 000377 377 ;DATA CHAR FOR EIGHT BITS PER CHAR.  
1828 005572 000010 8 ;SHIFTS PER CHAR. NEEDED FOR TEST  
1829 005574 004000 4000 ;BITS PER CHAR SELECTION FOR DQSEC.  
1830 005576 104400 SCOPE
```

```

1831
1832 ;ABOVE TEST FOR EIGHT BITS PER CHAR.
1833 ;BELOW TEST FOR SIXTEEN BITS PER CHAR.
1834
1835 ; TEST 47
1836 :*****
1837 005600 012737 000047 001226 TST47: MOV #47,TSTNO
1838 005606 012737 006240 001216 MOV #TST50,NEXT
1839 005614 004537 005630 JSR R5,SYNTST ;GOTO THE ACTUAL TEST
1840 005620 177777 177777 ;DATA FOR 16 BTS PER CHAR.
1841 005622 000020 16. ;SHIFTS PER CHAR.
1842 005624 000000 0000 ;SELECTION FOR DQSEC BITS/PER CHAR.
1843 005626 104400 SCOPE ;SCOPE THIS TEST
1844
1845
1846
1847 ;
1848 ;TEST THAT SYNC 1 AND SYNC 2
1849 ;SET WHEN DATA IS RECEIVED
1850 ;THIS TEST WILL CHECK FOR EITHER
1851 ;1 OR 2 SYNC CHARACTERS.
1852
1852 005630 SYNTST:
1853 005630 012537 014060 MOV (R5)+,WORD ;GET DATA CHARACTER
1854 005634 011537 005724 MOV (R5),4$ ;GET NUMBER OF SHIFTS.
1855 005640 011537 006046 MOV (R5),6$ ;
1856 005644 162737 000002 006046 SUB #2,6$ ;ADJUST SHIFTS.
1857 005652 012537 006170 MOV (R5)+,8$ ;GET THE SHIFTS
1858 005656 005337 006170 DEC 8$ ;ADJUST THE SHIFTS.
1859 005662 011537 005726 MOV (R5),5$ ;GET THE BITS/PER CHAR.
1860 005666 011537 006050 MOV (R5),7$ ;
1861 005672 012537 006172 MOV (R5)+,9$ ;
1862 005676 010537 006236 MOV R5,10$ ;SAVE THE PC TO RETURN
1863 005702 104412 MSTCLR ;INIT THE DQ11
1864 005704 112777 000011 173456 MOVB #11,@DQREG ;SEL THE SYNC REG
1865 005712 012777 177777 173452 MOV #-1,@DQSEC ;SET SYNC CHAR TO ALL 1'S
1866 005720 004537 011524 JSR R5,RXSTRA ;GOTO THE SUBROUTINE
1867 005724 000001 4$: .BLKW 1 ;NUMBER OF SHIFTS
1868 005726 000001 5$: .BLKW 1 ;MISC FUNCTION
1869 005730 112777 000012 173432 MOVB #MISC,@DQREG ;SELECT THE MISC REGISTER
1870 005736 032777 020000 173426 BIT #BIT13,@DQSEC ;IS SYNC 1 UP YET
1871 005744 001401 BEQ .+4 ;BR IF NO
1872 005746 104000 HLT ;SYNC 1 UP TOO SOON
1873 005750 032777 010000 173402 BIT #BIT12,@DQRCSR ;ACTIVE UP??
1874 005756 001401 BEQ .+4 ;BR IF ACTIVE NOT UP
1875 005760 104000 HLT ;ACTIVE UP TOO SOON.
1876 005762 005277 173404 INC @DQSEC ;CLOCK UP
1877 005766 005377 173400 DEC @DQSEC ;CLOCK DN
1878 005772 032737 100000 001510 BIT #SYNBIT,DQSTAT ;NUMBER OF SYNC CHARS=?
1879 006000 001003 BNE .+10 ;BR IF TWO SYNC CHAR.
1880 006002 005337 006170 DEC 8$ ;ADJUST COUNT WHEN ONE SYNC SELECTED.
1881 006006 000442 BR 1$ ;BR TO TEST ONE SYNC CHAR.
1882 006010 017700 173356 MOV @DQSEC,RO ;READ DQSEC
1883 006014 042700 147777 BIC #147777,RO ;CLEAR GARBAGE
1884 006020 022700 020000 CMP #20000,RO ;IS SYNC 1 UP?
1885 006024 001401 BEQ ., ' ;BR IF YES
1886 006026 104000 HLT ;SYNC ONE NOT SET OR SYNC 2 IS SET

```

```

1887 006030 032777 010000 173322 BIT #BIT12,@DQRCR ;ACTIVE UP?
1888 006036 001401 BEQ .+4 ;BR IF ACTIVE =0
1889 006040 104000 HLT ;ACTIVE UP TOO SOON
1890 006042 004537 011524 JSR R5,RXSTRA ;GOTO THE SUBROUTINE
1891 006046 000001 6$: .BLKW 1 ;NUMBER OF SHIFTS MINUS 2
1892 006050 000001 7$: .BLKW 1 ;MISC FUNCTION (PERS PER CHAR).
1893 006052 017700 173314 MOV @DQSEC,R0 ;READ THE DQSEC
1894 006056 042700 147777 BIC #147777,R0 ;CLEAR ALL BUT SYNC 1 AND SYNC 2
1895 006062 022700 020000 CMP #20000,R0 ;ARE BOTH SYNC 1 *AND* SYNC 2 SET?
1896 006066 001401 BEQ .+4 ;BR IF YES
1897 006070 104000 HLT ;EITHER OR BOTH SYNC 1 OR SYNC 2 NOT SET.
1898 006072 032777 010000 173260 BIT #BIT12,@DQRCR ;ACTIVE UP??
1899 006100 001401 BEQ .+4 ;BR IF ACTIVE NOT SET.
1900 006102 104000 HLT ;ACTIVE UP TOO SOON
1901 006104 005277 173262 INC @DQSEC ;CLOCK UP.
1902 006110 005377 173256 DEC @DQSEC ;CLOCK DN
1903 006114 017700 173252 $: MOV @DQSEC,R0 ;READ AND SAVE DQSEC
1904 006120 042700 147777 BIC #147777,R0 ;CLEAR ALL BUT SYNC 1 AND SYNC 2
1905 006124 022700 030000 CMP #30000,R0 ;ARE BOTH SYNC 1 AND SYNC 2 SET?
1906 006130 001401 BEQ .+4 ;BR IF YES
1907 006132 104000 HLT ;EITHER OR BOTH SYNC 1 OR SYNC 2 NOT SET.
1908 006134 032737 004000 001510 BIT #ACTBIT,DQSTAT ;WHEN DO YOU GO ACTIVE??
1909 006142 001006 BNE 2$ ;BR IF ACTIVE ON FIRST NON-SYNC.
1910 006144 032777 010000 173206 BIT #BIT12,@DQRCR ;IS ACTIVE UP?
1911 006152 001001 BNE .+4 ;*** NOW ACTIVE SHOULD BE SET***
1912 006154 104000 HLT ;NOW ACTIVE SHOULD BE UP..
1913 006156 000424 BR 3$ ;ALL DONE GO HOME
1914 006160 005037 014060 2$: CLR WORD ;SET DATA TO NON-SYNC
1915 006164 004537 011524 JSR R5,RXSTRA ;PUSH IT INTO THE RECEIVER
1916 006170 000001 8$: .BLKW 1 ;NUMBER OF SHIFTS MINUS 1
1917 006172 000001 9$: .BLKW 1 ;MISC FUNCTION.
1918 006174 032777 010000 173156 BIT #BIT12,@DQRCR ;ACTIVE UP
1919 006202 001401 BEQ .+4 ;ONE MORE SHIFT BEFORE ACTIVE=1
1920 006204 104000 HLT ;ACTIVE IS UP TOO SOON
1921 006206 005277 173160 INC @DQSEC ;FINAL CLOCK UP
1922 006212 005377 173154 DEC @DQSEC ;CLOCK DN
1923 006216 032777 010000 173134 BIT #BIT12,@DQRCR ;**** NOW ACTIVE SHOULD BE SET **
1924 006224 001001 BNE .+4 ;BR IF ACTIVE =1
1925 006226 104000 HLT ;ACTIVE ON FIRST NON-SYNC NOT WORKING.
1926 006230 013705 006236 3$: MOV 10$,R5 ;RESTORE PC POINTER
1927 006234 000205 RTS R5 ;GOTO MAIN TEST
1928 006236 000000 10$: 0 ;STORE R5 (PC) HERE.

```

```

1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942

```

```

;
; TEST OF RECEIVER CHARACTER COUNT AND BUSS
; ADDRESS. TEST TO MAKE SURE
; THAT THEY INCREMENT PROPPERELY.
;
; TEST WITH CHARACTER COUNT OF -1 (ODD)
;
; TEST 50
;*****

```

```

1943 006240 012737 000050 001226 TST50: MOV #50,TSTNO
1944 006246 012737 006366 001216 MOV #TST51,NEXT
1945 006254 104412 MSTCLR ;INIT DQ11
1946 006256 105077 173106 CLRB @DQREG ;SEL RX BA PRI.
1947 006262 012777 014120 173102 MOV #RXBUFF,@DQSEC ;SET RX BA PRI.
1948 006270 105277 173074 INCB @DQREG ;SEL RX WC PRI.
1949 006274 012777 177777 173070 MOV #-1,@DQSEC ;ONE CHAR RECEIVE
1950 006302 112777 000012 173060 MOVB #MISC,@DQREG ;SELECT THE MISC REG.
1951 006310 012777 004010 173054 MOV #4010,@DQSEC ;SET EIGHT BITS AND TEST LOOP
1952 006316 012777 010001 173034 MOV #10001,@DQRCR ;SET RX ACTIVE AND RX GO!!
1953 006324 105777 173030 TSTB @DQRCR ;RX PRI DONE?
1954 006330 100375 BPL -4 ;HANG HERE TILL DONE.
1955 006332 105077 173032 CLRB @DQREG ;GET RA BA PRI.
1956 006336 022777 014121 173026 CMP #RXBUFF+1,@DQSEC ;DID BA INC RIGHT?
1957 006344 001401 BEQ +4 ;BR IF BA GOOD
1958 006346 104000 HLT ;RX BA ERROR.
1959 006350 105277 173014 INCB @DQREG ;GET RX WC PRI.
1960 006354 005777 173012 TST @DQSEC ;DID IT GOTO ZERO?
1961 006360 001401 BEQ +4 ;BR IF YES
1962 006362 104000 HLT ;RX WC PRI NOT =0
1963 006364 104400 SCOPE ;SCOPE THE TEST

```

```

1964
1965
1966
1967
1968
1969 ;TEST OF RECEIVER CHARACTER COUNT
1970 ;AND BUSS ADDRESS
1971 ;WITH A CHARACTER COUNT OF -2 (EVEN)
1972 ;MAKING SURE THAT THE CC AND BA
1973 ;INCREMENT CORRECTLY.
1974
1975
1976
1977

```

```

: TEST 51
:*****
1978 006366 012737 000051 001226 TST51: MOV #51,TSTNO
1979 006374 012737 006514 001216 MOV #TST52,NEXT
1980 006402 104412 MSTCLR ;ISSUE CLEAR
1981 006404 105077 172760 CLRB @DQREG ;SELECT THE RX BA PRI
1982 006410 012777 014120 172754 MOV #RXBUFF,@DQSEC ;SET RX BA PRI.
1983 006416 105277 172746 INCB @DQREG ;SELECT RX WC PRI.
1984 006422 012777 177776 172742 MOV #-2,@DQSEC ;SET FOR TWO CHARS
1985 006430 112777 000012 172732 MOVB #MISC,@DQREG ;SELECT THE MISC REGISTER
1986 006436 012777 004010 172726 MOV #4010,@DQSEC ;SET EIGHT BITS AND TEST LOOP
1987 006444 012777 010001 172706 MOV #10001,@DQRCR ;SET RX ACTIVE AND GO!!
1988 006452 105777 172702 TSTB @DQRCR ;WAIT FOR RX PRI DONE.
1989 006456 100375 BPL -4 ;HANG HERE TILL DONE
1990 006460 105077 172704 CLRB @DQREG ;SELECT THE RX BA PRI
1991 006464 022777 014122 72700 CMP #RXBUFF+2,@DQSEC ;DID RX BA INCREMENT RIGHT?
1992 006472 001401 BEQ +4 ;BR IF GOOD
1993 006474 104000 HLT ;RX BA ERROR
1994 006476 105277 172666 INCB @DQREG ;SELECT THE RX WC PRI.
1995 006502 005777 172664 TST @DQSEC ;DID IF GOTO ZERO
1996 006506 001401 BEQ +4 ;BR IF YES
1997 006510 104000 HLT ;RX WC NOT =ZERO
1998 006512 104400 SCOPE ;SCOPE THE TEST

```

1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054

:RECEIVER DATA REALIBILITY TEST.
:TEST TO RECEIVE A SIXTEEN
:BIT BINARY COUNT PATTERN (000000-177777)

:NOTE: IF PARFLG IS NON-ZERO THE PARITY TEST IS
:IN PROGRESS. THERE ARE NO ERRORS EXPECTED
:PARITY TEST DATA (177400-177777)

: TEST 52

:*****

006514 012737 000052 001226
006522 012737 006662 001216
006530 012737 006552 001220
006536 105037 012604
006542 112737 000377 014054
006550 005000
006552 104412
006554 010037 014060
006560 105737 012604
006564 001402
006566 004737 012446
006572 004537 011524
006576 000020
006600 000000
006602 013737 014120 001252
006610 013737 014060 001254
006616 005777 172544
006622 100001
006624 104000
006626 023737 001254 001252
006634 001401
006636 104020
006640 104401
006642 005200
006644 001342
006646 012700 177400
006652 105137 012604
006656 001335
006660 104400

TST52: MOV #52,TSTNO
MOV #TST53,NEXT
MOV #1\$,LOCK
CLRB PARFLG ;SET FOR NO PARITY NOW
MOVB #377,EXTFLG ;TELL SUBROUTINE 16 BIT CHAR.
CLR R0 ;ZERO DATA POINTER
1\$: MSTCLR ;ISSUE CLEAR DQ11
MOV R0,WORD ;LOAD DATA FOR SUB ROUTINE
TSTB PARFLG ;IS PARITY ENABLED?
BEQ .+6 ;BR IF NO
JSR PC,GENPAR ;GO AND FIGURE PARITY.
JSR R5,RXSTRA ;GO PUSH CHARACTER INTO RECEIVER.
16. ;NUMBER OF SHIFTS NEEDED
0000 ;BITS PER/CHAR FOR MISC REG
MOV RXBUFF,TEMP4 ;GET EXPECTED
MOV WORD,TEMP5 ;GET EXPECTED
2\$: TST @DQERR ;ANY ERRORS?
BPL .+4 ;BR IF NO ERRORS
HLT ;DQ11 ERROR FLAG SET CHECK SEL 4
CMP TEMP5,TEMP4 ;DATA OK??
BEQ .+4 ;BR IF GGOD DATA
HLT 20 ;RECEIVER DATA COMPARISON ERROR.
SCOPE ;LOCK ON SLECTED DATA (SW09=1)
INC R0 ;UPDATE DATA POINTER.
BNE 1\$;BR IF MORE CHARS TO GO.
MOV #177400,R0 ;SET FOR PARITY TEST.
COMB PARFLG ;TURN PARITY ON NOW
BNE 1\$;DO TEST WITH PARITY ENABLED NOW.
SCOPE ;SCOPE THE TEST.

:RECEIVER PARITY ERROR TEST.
:THE PARITY WILL PURPOSELY BE MADE INCORRECT AND
:AN ERROR WILL BE EXPECTED EVERY TIME.

:TEST TO RECEIVE A SIXTEEN
:BIT BINARY COUNT PATTERN (000000-000177)

: TEST 53

:*****


```

2055 006662 012737 000053 001226 TST53: MOV #53,TSTNO
2056 006670 012737 007034 001216 MOV #TST54,NEXT
2057 006676 012737 006722 001220 MOV #1$,LOCK
2058 006704 112737 000377 012604 MOV #377,PARFLG ; TELL SUBROUTINE PARITY IS ENABLED.
2059 006712 112737 000377 014054 MOV #377,EXTFLG ; TELL SUBROUTINE THIS IS A 16 BIT CHAR.
2060 006720 005000 CLR R0 ; CLEAR DATA POINTER
2061 006722 104412 1$: MSTCLR ; INIT DQ11
2062 006724 012737 000377 011770 MOV #377,NPRFLG ; SET FOR SUBROUTINE.
2063 006732 010037 014060 MOV R0,WORD ; LOAD DATA
2064 006736 004737 012446 JSR PC,GENPAR ; CALCULATE PARITY.
2065 006742 032737 100000 014060 BIT #BIT15,WORD ; CHECK PARITY BIT
2066 006750 001404 BEQ .+12 ; BR IF PARITY BIT CLEARED
2067 006752 042737 100000 014060 BIC #BIT15,WORD ; PARITY BIT SET ;; SO CLEAR IT.
2068 006760 000403 BR .+10 ; CONTINUE TEST
2069 006762 052737 100000 014060 BIS #BIT15,WORD ; PARITY BIT CLR ;; SO SET IT.
2070 006770 004537 011524 JSR R5,RXSTRA ; PUSH CHARACTER INTO RECEIVER
2071 006774 000020 16. ; SHIFTS NEEDED.
2072 006776 000000 0000 ; BITS PER CHAR SELECT.
2073 007000 013737 014120 001252 MOV RXBUFF,TEMP4 ; GET ACTUAL..
2074 007006 013737 014060 001254 MOV WORD,TEMP5 ; GET EXPECTED..
2075 007014 005777 172346 2$: TST @DQERR ; DID THE ERROR FLAG SET...*...
2076 007020 100401 BMI .+4 ; BR IF AN ERROR OCCURED.
2077 007022 104000 HLT ; ERROR NO ERROR (PARITY ERROR)
2078 007024 104401 SCOP1 ; LOCK ON CHARACTER? (SW09=1)
2079 007026 105200 INCB R0 ; UPDATE DATA POINTER.
2080 007030 100334 BPL 1$ ; BR IF NOT 200(8) CHARS DONE.
2081 007032 104400 SCOPE ; SCOPE THIS TEST
2082
2083
2084
2085 ;
2086 ; TEST OF RECEIVER HALF DUPLEX
2087 ; TEST TO TRANSMITT
2088 ; A TWO HUNDRED CHARACTER BURST OF DATA CHARACTERS
2089 ; WITH THE RECEIVER IN HALF DUPLEX
2090 ; MAKING SURE THAT THE RECEIVER
2091 ; DOESNT RECEIVE ANY CHARACTERS.
2092
2093 ; TEST 54
2094 ; *****
2095 007034 012737 000054 001226 TST54: MOV #54,TSTNO
2096 007042 012737 007436 001216 MOV #TST55,NEXT
2097 007050 005000 CLR R0 ; INIT DATA REG
2098 007052 012704 014526 1$: MOV #TXBUFF,R4 ; PREPARE TO FILL TX BUFFER WITH BINARY COUNT.
2099 007056 110024 MOV R0,(R4)+ ; START FILLING TX BUFF
2100 007060 105200 INCB R0 ; UPDATE DATA REG
2101 007062 100375 BPL 1$ ; BRANCH IF BUFFER HASN'T BEEN FILLED
2102 007064 104413 2$: MEMCLR ; INIT THE DEVICE
2103 007066 005000 CLR R0 ; CLEAR COUNT REG
2104 007070 012704 014120 3$: MOV #RXBUFF,R4 ; PREPARE TO CLEAR THE RECEIVER BUFFER.
2105 007074 105024 CLRB (R4)+ ; START CLEARING RX BUFF
2106 007076 105200 INCB R0 ; UPDATE THE COUNTER
2107 007100 001375 BNE 3$ ; IS RX BUFF ALL CLEARED?
2108 007102 105077 172262 CLRB @DQREG ; SELECT THE RECEIVER BA PRI
2109 007106 012777 014120 172256 MOV #RXBUFF,@DQSEC ; LOAD THE BA
2110 007114 105277 172250 INCB @DQREG ; SELECT THE RECEIVER (C PRI

```



```

2167 007414 012705 014120      MOV      #RXBUFF,R5      ;GET RX BUFFER.
2168 007420 122725 000377      10$:    CMPB     #377,(R5)+   ;MARK STATE IN BUFFER?
2169 007424 001401              BEQ      .+4             ;BR IF YES
2170 007426 104000              HLT                     ;ERROR
2171 007430 105200              INCB    R0              ;ALL DONE?
2172 007432 100372              BPL     10$            ;BR IF NO.
2173 007434 104400              SCOPE                   ;SCOPE THIS TEST.
2174
2175
2176
2177
2178      ;
2179      ;TEST OF DQ11 TRANSMITTER AND RECEIVER
2180      ;DATA REALIBILITY.
2181      ;DATA IS TRANSFERED FULL RATE
2182      ;AT A FOUR HUNDRED CHARACTER BURST
2183      ;
2184      ;
2185      ; TEST 55
2186      ;*****
2186 007436 012737 000055 001226  TST55:  MOV      #55,TSTNO
2187 007444 012737 007756 001216      MOV      #TST56,NEXT
2188 007452 005000              CLR      R0             ;INIT DATA REG
2189 007454 012704 014526      MOV      #TXBUFF,R4     ;PREPARE TO FILL TX BUFFER WITH BINARY COUNT.
2190 007460 110024              1$:    MOV     R0,(R4)+       ;START FILLING TX BUFF
2191 007462 105200              INCB    R0              ;UPDATE DATA REG
2192 007464 001375              BNE     1$             ;BRANCH IF BUFFER HASN'T BEEN FILLED
2193 007466 104413              2$:    MEMCLR            ;INIT THE DEVICE
2194 007470 005000              CLR      R0             ;CLEAR COUNT REG
2195 007472 012704 014120      MOV      #RXBUFF,R4     ;PREPARE TO CLEAR THE RECEIVER BUFFER.
2196 007476 105024              3$:    CLRB    (R4)+       ;START CLEARING RX BUFF
2197 007500 105200              INCB    R0              ;UPDATE THE COUNTER
2198 007502 001375              BNE     3$             ;IS RX BUFF ALL CLEARED?
2199 007504 105077 171660      CLRB    @DQREG          ;SELECT THE RECEIVER BA PRI
2200 007510 012777 014120 171654      MOV      #RXBUFF,@DQSEC ;LOAD THE BA
2201 007516 105277 171646      INCB    @DQREG          ;SELECT THE RECEIVER CC PRI
2202 007522 012777 177400 171642      MOV      #-400,@DQSEC   ;LOAD THE CC WITH -400 (I WANT TO RECEIVE 400 CHARACTERS
2203 007530 105277 171634      INCB    @DQREG          ;SELECT THE TX BA PRI
2204 007534 012777 014524 171630      MOV      #SYNC,@DQSEC   ;LOAD THE TX BA WITH STARTING ADD OF TX DATA PLUS THE SY
2205 007542 105277 171622      INCB    @DQREG          ;SELECT THE TX CC PRI
2206 007546 012777 177376 171616      MOV      #-402,@DQSEC   ;LOAD THE TX CC WITH -402 (FOUR HUNDRED CHARACTERS AND T
2207 007554 112777 000011 171606      MOV     #11,@DQREG      ;SELECT THE SYNC REGISTER
2208 007562 013777 014522 171602      MOV      .SYNC,@DQSEC   ;LOAD IT WITH THE SYNC CHAR
2209 007570 105277 171574      INCB    @DQREG          ;SELECT THE MISC REGISTER
2210 007574 012777 004010 171570      MOV      #4010,@DQSEC   ;LOAD IT WITH EIGHT BITS PER/CHAR AND TEST LOOP
2211 007602 005037 001244      CLR     TEMP1           ;ZERO DELAY LOC1
2212 007606 012737 000020 001246      MOV      #20,TEMP2      ;SET DELAY FOR 20X177777 (8)
2213 007614 005277 171540      INC     @DQRCR          ;SET RECEIVER GO!!
2214 007620 005277 171540      INC     @DQTCR          ;SET TRANSMITTER GO!!
2215 007624 105777 171530      4$:    TSTB    @DQRCR          ;RECEIVER DONE??
2216 007630 100407              BMI     5$             ;BRANCH IF RECEIVER IS DONE.
2217 007632 005237 001244      INC     TEMP1           ;START THE DELAY
2218 007636 001372              BNE     4$             ;DELAY-----
2219 007640 005337 001246      DEC     TEMP2           ;DELAY----- REC DONE?
2220 007644 001367              BNE     4$             ;DELAY-----
2221 007646 104000              HLT                     ;RECEIVER DONE NEVER SET (PRI)
2222 007650 005777 171512      5$:    TST     @DQERR

```

DQ11 TRANSMITTER AND RECEIVER EXERCISER.

```
2223 007654 100001          BPL      .+4
2224 007656 104000          HLT
2225 007660 122777 000204 171472  CMPB    #204,@DQRCR
2226 007666 001401          BEQ      .+4
2227 007670 104000          HLT
2228 007672 122777 000204 171464  CMPB    #204,@DQTCSR
2229 007700 001401          BEQ      .+4
2230 007702 104000          HLT
2231 007704 005000          CLR      R0              ;INIT COUNT REG
2232 007706 012704 014526  MOV     #TXBUFF,R4      ;SET GOOD DATA POINTER
2233 007712 012705 014120  MOV     #RXBUFF,R5      ;SET REC DATA POINTER
2234 007716 005037 001254          CLR     TEMP5
2235 007722 005037 001252          CLR     TEMP4
2236 007726 112437 001254  MOV     (R4)+,TEMP5
2237 007732 112537 001252  MOV     (R5)+,TEMP4
2238 007736 023737 001254 001252  CMP     TEMP5,TEMP4
2239 007744 001401          BEQ      .+4              ;DATA GOOD SO FAR
2240 007746 104025          HLT     25              ;DATA COMPARISON ERROR
2241 007750 105200          INCB    RC              ;UPDATE COUNTER
2242 007752 001361          BNE     6$              ;BRANCH IF MORE DATA TO CHECK
2243 007754 104400          SCOPE
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
```

```
        ;TEST OF THE THREE STRAP SELECTABLE
        ;CHARACTERS
        ;ON THE FIRST PASS THE CHARACTERS
        ;WILL BE TYPED OUT FOR VERIFICATION
        ;ON PASSES AFTER THAT THE CHARACTERS WILL BE VERIFIED
        ;BY THE PROGRAM.
        ;NOTE: IF THE BB OPTION IS INSTALLED
        ;PROCEED TO NEXT TEST.
```

```
        ; TEST 56
        ;*****
2258 007756 012737 000056 001226  TST56:  MOV     #56,TSTNO
2259 007764 012737 015130 001216  MOV     #.EOP,NEXT
2260 007772 012737 010136 001220  MOV     #1$,LOCK
2261 010000 104413          MEMCLR          ;CLEAR ALL
2262 010002 005037 011770          CLR     NPRFLG
2263 010006 032737 020000 001510  BIT     #BBBIT,DQSTAT ;DOES BB OPTION EXIST?
2264 010014 001405          BEQ     .+14        ;BR IF BB NOT THERE.
2265 010016 013737 001216 001214  MOV     NEXT,RETURN  ;DO NEXT TEST.
2266 010024 000177 171164          JMP     @RETURN
2267 010030 012737 000010 010156  MOV     #8,5$       ;EIGHT SHIFTS.
2268 010036 012737 004000 010160  MOV     #400,6$     ;EIGHT BITS PER CHAR.
2269 010044 012737 000400 010274  MOV     #400,15$    ;LAST CHARACTER.
2270 010052 005000          CLR     R0          ;ZERO DATA POINTER
```

```
        ;*****
        ; MAINTAINCE AID.
        ; THE FOLLOWING IS TO HELP TROBLE SHOOT
        ; PROBLEMS IN THE CHARACTER DET. LOGIC
        ; FASTER.
        ;*****
        ;-----
```

```
2271
2272
2273
2274
2275
2276
2277
2278
```

```

2279
2280 010054 000416 BR 36$ ;CHANGE THIS LOCATION TO '240' (NOP)
2281 ;TO LOCK ON SELECTED 8 BIT CHAR.
2282 010056 000000 HALT ;PUT SELECTED CHARACTER IN SWR.
2283 ;HIT CONT.
2284 010060 104414 CKSWR ;CHECK FOR <^G>
2285 010062 017700 171112 MOV @SWR,R0 ;LOAD CHARACTER.
2286 010066 000000 HALT ;PUT DYNAMIC SWR SETTINGS IN SWR AND
2287 ;HIT CONT.
2288 010070 104414 CKSWR ;CHECK FOR <^G>
2289 010072 000407 BR 36$ ;CHANGE THIS LOCATION TO ' ' (NOP)
2290 ;ALONG WITH THE ABOVE FOR CHAR
2291 ;NOTE: BOTH LOCATIONS WILL BE CHANGED
2292 ;FOR A 16 BIT CHAR.
2293 010074 012737 000020 010156 MOV #16.,5$ ;SET FOR 16 SHIFTS.
2294 010102 005037 010160 CLR 6$ ;SET 'BITS/PER/CHAR'
2295 010106 005037 010274 CLR 15$ ;SET LAST LIMIT.
2296
2297 ;
2298 ; NOTE SWR BIT 9 MUST BE SET TO LOCK ON THAT CHAR. SELECTED.
2299 ;
2300 ;-----
2301 010112 012704 014070 36$: MOV #TMPBUF,R4 ;STORAGE POINTER.
2302 010116 005024 CLR (R4)+ ;ZERO STORAGE
2303 010120 022704 014106 CMP #TMPBUF+16,R4 ;ALL CLEAR?
2304 010124 001374 BNE .-6 ;BR IF NO.
2305 010126 005037 014116 CLR NUMBER ;HOW MANY FOUND.
2306 010132 012704 014070 MOV #TMPBUF,R4 ;PREPARE POINTER
2307 010136 005137 011770 1$: COM NPRFLG ;TELL SUBROUTINE NOT TO FORCE RX NPR.
2308 010142 005077 171212 CLR @DQRCSR ;CLEAR RX CSR
2309 010146 010037 014060 MOV R0,WORD ;LOAD CHARACTER
2310 010152 004537 011524 JSR R5,RXSTRA ;PUSH CHARACTER INTO RECEIVER.
2311 010156 000010 5$: 8. ;BEWARE THIS LOCATION WILL CHANGE.
2312 010160 004000 6$: 4000 ;BEWARE THIS LOCATION WILL CHANGE.
2313 010162 005777 171172 TST @DQRCSR ;WAS A CHARACTER DETECTED?
2314 010166 100037 BPL 2$ ;BR IF NO CHAR FOUND.
2315 010170 042777 100000 171162 BIC #BIT15,@DQRCSR ;CLEAR DETECED CHAR FLAG
2316 010176 005700 TST R0 ;WAS THE CHAR=0
2317 010200 001003 BNE 18$ ;BR IF NO.
2318 010202 005737 014116 TST NUMBER ;HOW MANY WERE FOUND?
2319 010206 001410 BEQ 19$ ;BR IF NONE YET.
2320 010210 012702 014070 18$: MOV #TMPBUF,R2 ;POINTER STORE.
2321 010214 020022 13$: CMP R0,(R2)+ ;WAS THIS CHARACTER FOUND BEFORE?
2322 010216 001423 BEQ 2$ ;BR IF YES
2323 010220 005722 TST (R2)+ ;POP POINTER
2324 010222 022702 014110 CMP #TMPBUF+20,R2 ;ALL CHARS CHECKED?
2325 010226 001372 BNE 13$ ;BR IF NO.
2326 010230 010024 19$: MOV R0,(R4)+ ;STOE CHARACTER
2327 010232 017714 171122 MOV @DQRCSR,(R4) ;GET ADDRESS FOUND IN.
2328 010236 042714 170377 BIC #170377,(R4) ;CLEAR ALL GARBAGE.
2329 010242 000324 SWAB (R4)+ ;SWAP AROUND.
2330 010244 005237 014116 INC NUMBER ;UPDATE COUNTER.
2331 010250 022737 000005 014116 CMP #5,NUMBER ;TOO MANY CHARS FOUND??
2332 010256 001003 BNE 2$ ;BR IF OK.
2333 010260 104000 HLT ;ERROR MORE THAN 4 CHARS. WERE DETECTED.
2334 010262 000177 170726 JMP @RETURN ;RESTART TEST. DO NOT CONTINUE IN THIS TEST

```

2335	010266	104401			2\$:	SCOP1		:LOCK ON CHAR (SW09=1)
2336	010270	005200				INC	RO	:UPDATE CHARACTER
2337	010272	020027				CMP	RO,(PC)+	:ALL DONE?
2338	010274	000000			15\$:	0		:LAST CHAR STORED HERE.
2339	010276	001317				BNE	1\$:BR IF NOT DONE
2340	010300	005737	014116			TST	NUMBER	:ANY CHARS FOUND?
2341	010304	001024				BNE	30\$:BR IF NONE FOUND
2342	010306	022737	000020	010156	31\$:	CMP	#16.,5\$:IS TEST ALL DONE?
2343	010314	001434				BEQ	7\$:BR IF YES
2344	010316	012737	000020	010156		MOV	#16.,5\$:DO A 16 BIT CHAR NOW
2345	010324	005037	010160			CLR	6\$:SET FOR 16 BITS PER CHAR.
2346	010330	112777	000012	171032		MOV#B	#MISC.,@DQREG	:SEL MISC REG
2347	010336	042777	177400	171026		BIC	#177400,@DQSEC	:CLEAR THE HIGH BYTE
2348	010344	005037	010274			CLR	15\$:SET LAST CHAR TO 0
2349	010350	005000				CLR	RO	:ZERO DATA POINTER
2350	010352	000137	010136			JMP	1\$:GO AND DO IT AGAIN
2351	010356	022737	000001	014116	50\$:	CMP	#1,NUMBER	:WAS 1 CHAR FOUND?
2352	010364	001010				BNE	7\$:BR IF NO.
2353	010366	022737	000010	014072		CMP	#10,TMPBUF+2	:WAS 'SYNC DET' ENABLED?
2354	010374	001004				BNE	7\$:BR IF NO.
2355	010376	005337	014116			DEC	NUMBER	:ZERO NUMBER.
2356	010402	024444				CMP	-(R4),-(R4)	:ADJUST POINTERS
2357	010404	000740				BR	31\$:KEEP GOING.
2358	010406	005737	014116		7\$:	TST	NUMBER	:ANY FOUND?
2359	010412	001004				BNE	.+12	:BR IF YES
2360	010414	104402	013120			TYPE	,EM4	:ALLERT OPERATOR NONE FOUND.
2361	010420	000137	010630			JMP	10\$:LEAVE
2362	010424	105737	014114			TST#B	XYZFLG	:WAS THIS DONE BEFORE?
2363	010430	001050				BNE	3\$:BR IF TEST WAS DONE BEFORE
2364	010432	012704	014070			MOV	#TMPBUF,R4	:POINTER
2365	010436	012437	010702			MOV	(R4)+,..CHAR1	:STORE CHARACTER 1
2366	010442	012437	010704			MOV	(R4)+,..ADDR1	:STORE ADDRESS 1
2367	010446	012437	010706			MOV	(R4)+,..CHAR2	:STORE CHARACTER 2
2368	010452	012437	010710			MOV	(R4)+,..ADDR2	:STORE ADDRESS 2
2369	010456	012437	010712			MOV	(R4)+,..CHAR3	:STORE CHARACTER 3
2370	010462	012437	010714			MOV	(R4)+,..ADDR3	:STORE ADDRESS 3
2371	010466	012437	010716			MOV	(R4)+,..CHAR4	:STORE CHARACTER 4
2372	010472	012437	010720			MOV	(R4)+,..ADDR4	:STORE ADDRESS 4
2373	010476	013737	014116	001252		MOV	NUMBER,TEMP4	:STORE NUMBER OF CHARACTER FOUND.
2374	010504	104402				TYPE		
2375	010506	013726				MDETC#H		
2376	010510	104410				CONVRT		
2377	010512	010632				XCHAR1		
2378	010514	005337	001252			DEC	TEMP4	
2379	010520	001414				BEQ	3\$	
2380	010522	104410				CONVRT		
2381	010524	010644				XCHAR2		
2382	010526	005337	001252			DEC	TEMP4	
2383	010532	001407				BEQ	3\$	
2384	010534	104410				CONVRT		
2385	010536	010656				XCHAR3		
2386	010540	005337	001252			DEC	TEMP4	
2387	010544	001402				BEQ	3\$	
2388	010546	104410				CONVRT		
2389	010550	010670				XCHAR4		
2390	010552	022737	000001	001504	3\$:	CMP	#1,DQNUM	

2391	010560	001003			BNE	.+10
2392	010562	012737	177777	014114	MOV	#-1,XYZFLG
2393	010570	013737	014116	001252	MOV	NUMBER,TEMP4
2394	010576	012704	014070		MOV	#TMPBUF,R4
2395	010602	012705	010702		MOV	#.CHAR1,R5
2396	010606	022425			4\$: CMP	(R4)+,(R5)+
2397	010610	001401			BEQ	.+4
2398	010612	104022			HLT	22
2399	010614	022425			CMP	(R4)+,(R5)+
2400	010616	001401			BEQ	.+4
2401	010620	104022			HLT	22
2402	010622	005337	001252		DEC	TEMP4
2403	010626	001367			BNE	4\$
2404	010630	104400			10\$: SCOPE	
2405	010632	000002			XCHAR1: 2	
2406	010634	006	002		.BYTE	6,2
2407	010636	010702			.CHAR1	
2408	010640	004	002		.BYTE	4,2
2409	010642	010704			.ADDR1	
2410	010644	000002			XCHAR2: 2	
2411	010646	006	002		.BYTE	6,2
2412	010650	010706			.CHAR2	
2413	010652	004	002		.BYTE	4,2
2414	010654	010710			.ADDR2	
2415	010656	000002			XCHAR3: 2	
2416	010660	006	002		.BYTE	6,2
2417	010662	010712			.CHAR3	
2418	010664	004	002		.BYTE	4,2
2419	010666	010714			.ADDR3	
2420	010670	000002			XCHAR4: 2	
2421	010672	006	002		.BYTE	6,2
2422	010674	010716			.CHAR4	
2423	010676	004	002		.BYTE	4,2
2424	010700	010720			.ADDR4	
2425	010702	000000			.CHAR1: 0	
2426	010704	000000			.ADDR1: 0	
2427	010706	000000			.CHAR2: 0	
2428	010710	000000			.ADDR2: 0	
2429	010712	000000			.CHAR3: 0	
2430	010714	000000			.ADDR3: 0	
2431	010716	000000			.CHAR4: 0	
2432	010720	000000			.ADDR4: 0	

```

2433
2434
2435
2436
2437
2438
2439
2440
2441
2442 010722 104412
2443 010724 005037 014060
2444 010730 010537 014056
2445 010734 012537 014064
2446 010740 012537 014066
2447 010744 112777 000002 170416
2448 010752 012777 014060 170412
2449 010760 105277 170404
2450 010764 012777 177777 170400
2451 010772 112777 000012 170370
2452 011000 013777 014066 170364
2453 011006 052777 000012 170356
2454 011014 005277 170344
2455 011020 027777 170340 170336
2456 011026 027777 170332 170330
2457 011034 027777 170324 170322
2458 011042 005277 170324
2459 011046 005377 170320
2460 011052 005277 170314
2461 011056 005377 170310
2462 011062 032777 000200 170302
2463 011070 001001
2464 011072 104023
2465 011074 005337 014064
2466 011100 001364
2467 011102 005277 170264
2468 011106 005377 170260
2469 011112 032777 000200 170252
2470 011120 001401
2471 011122 104007
2472 011124 000205

```

```

:
:*****
:SUBROUTINE TO STROBE INTO THE TRANSMITTER
:THE CHARACTER 'WORD'.
:ENTERED BY A JSR R5, TXSTRB
:THE NUMBER OF SHIFTS AND THE BIT
:LENGTH SELECT ARE UNDER THE JSR CALL
:*****
:
TXSTRB: MSTCLR
CLR WORD
MOV R5,SAVEPC
MOV (R5)+,COUNT
MOV (R5)+,BITSEL
MOVB #2,@DQREG
MOV #WORD,@DQSEC
INCB @DQREG
MOV #-1,@DQSEC
MOVB #MISC.,@DQREG
MOV BITSEL,@DQSEC
BIS #12,@DQSEC
INC @DQTCR
CMP @DQTCR,@DQTCR ;WAIST TIME
CMP @DQTCR,@DQTCR ;WAIST TIME
CMP @DQTCR,@DQTCR ;WAIST TIME
INC @DQSEC
DEC @DQSEC
1$: INC @DQSEC
DEC @DQSEC
BIT #BIT7,@DQSEC
BNE .+4
HLT 23
DEC COUNT
BNE 1$
INC @DQSEC
DEC @DQSEC
BIT #BIT7,@DQSEC
BEQ .+4
HLT 7
RTS R5

```

```

2473
2474
2475
2476
2477
2478 011126 010537 014056
2479 011132 012537 014064
2480 011136 012537 014066
2481 011142 112777 000002 170220
2482 011150 012777 014060 170214
2483 011156 105277 170206
2484 011162 012777 177777 170202
2485 011170 112777 000012 170172
2486 011176 013777 014066 170166
2487 011204 052777 000012 170160
2488 011212 005277 170146

```

```

TXSTRC: MOV R5,SAVEPC
MOV (R5)+,COUNT
MOV (R5)+,BITSEL
MOVB #2,@DQREG
MOV #WORD,@DQSEC
INCB @DQREG
MOV #-1,@DQSEC
MOVB #MISC.,@DQREG
MOV BITSEL,@DQSEC
BIS #12,@DQSEC
INC @DQTCR

```


2489	011216	027777	170142	170140		CMP	@DQTCR,@DQTCR	:WAIST TIME
2490	011224	027777	170134	170132		CMP	@DQTCR,@DQTCR	:WAIST TIME
2491	011232	027777	170126	170124		CMP	@DQTCR,@DQTCR	:WAIST TIME
2492	011240	005277	170126			INC	@DQSEC	
2493	011244	005377	170122			DEC	@DQSEC	
2494	011250	005277	170116		1\$:	INC	@DQSEC	
2495	011254	005377	170112			DEC	@DQSEC	
2496	011260	005337	014064			DEC	COUNT	
2497	011264	001371				BNE	1\$	
2498	011266	000205				RTS	R5	
2499								
2500								
2501								
2502								
2503								
2504	011270	010537	014056			TXSTRD: MOV	R5,SAVEPC	:SAVE PC OF ROUTINE CALL
2505	011274	012537	014064			MOV	(R5)+,COUNT	:PICK UP THE NUMBER OF SHIFTS
2506	011300	012537	014066			MOV	(R5)+,BITSEL	:PICK UP NUMBER OF BITS PER CHARACTER
2507	011304	112777	000002	170056		MOVB	#2,@DQREG	:SELECT THE TRANSMITTER BA PRI.
2508	011312	012777	014060	170052		MOV	#WORD,@DQSEC	:LOAD THE BA
2509	011320	105277	170044			INCB	@DQREG	:SELECT THE TRANSMITTER CC PRI.
2510	011324	012777	177777	170040		MOV	#-1,@DQSEC	:LOAD THE CC WITH -1
2511	011332	112777	000012	170030		MOVB	#MISC.,@DQREG	:SELECT THE MISC REGISTER.
2512	011340	013777	014066	170024		MOV	BITSEL,@DQSEC	:LOAD MISC REG WITH NUMBER OF BITS PER CHAR.
2513	011346	052777	000012	170016		BIS	#12,@DQSEC	:ADD TO THAT TEST LOOP AND AUTO STEP.
2514	011354	105737	012604			TSTB	PARFLG	:IS PARITY TO BE TURNED ON?
2515	011360	001403				BEQ	#+10	:BR IF NO
2516	011362	052777	100000	170002		BIS	#BIT15,@DQSEC	:TURN PARITY ON.....
2517	011370	005277	167770			INC	@DQTCR	:SET TRANSMITTER GO!!!!
2518	011374	027777	167764	167762		CMP	@DQTCR,@DQTCR	:WAIST TIME
2519	011402	027777	167756	167754		CMP	@DQTCR,@DQTCR	:WAIST TIME
2520	011410	027777	167750	167746		CMP	@DQTCR,@DQTCR	:WAIST TIME
2521	011416	005277	167750			INC	@DQSEC	:PRIME THE
2522	011422	005377	167744			DEC	@DQSEC	:TRANSMITTER.
2523	011426	006037	001252		1\$:	ROR	TEMP4	:SHIFT THE STORAGE OF DATA FROM THE TRANSMITTER.
2524	011432	005277	167734			INC	@DQSEC	:CLOCK THE TRANSMITTER -UP-
2525	011436	005377	167730			DEC	@DQSEC	:CLOCK THE TRANSMITTER -DOWN-
2526	011442	017702	167724			MOV	@DQSEC,R2	:MOVE THE MISC REG TO R2
2527	011446	042702	177577			BIC	#177577,R2	:CLEAR ALL BUT THE BIT WINDOW.
2528	011452	105737	014054			TSTB	EXTFLG	:FIND OUT IF BIT PER CHAR >8
2529	011456	001404				BEQ	2\$:BRANCH IF 8OR<8
2530	011460	106102				ROLB	R2	:SHIFT BIT WINDOW INTO CARRY BIT.
2531	011462	006002				ROR	R2	:SHIFT CARRY INTO R2 (BIT 15 OF R2)
2532	011464	042702	077777			BIC	#77777,R2	:CLEAR ALL BUT THAT BIT OF DATA
2533	011470	050237	001252		2\$:	BIS	R2,TEMP4	:PLACE DATA INTO TEMPORY LOCATION
2534	011474	005337	014064			DEC	COUNT	:IS CHARACTER COMPLETELY SHIFTED OUT?
2535	011500	001352				BNE	1\$:BRANCH IF MORE BITS TO GO.
2536	011502	105737	014054			TSTB	EXTFLG	
2537	011506	001003				BNE	3\$	
2538	011510	105137	001252			COMB	TEMP4	
2539	011514	000402				BR	4\$	
2540	011516	005137	001252		3\$:	COM	TEMP4	:COMPLIMENT DATA STORAGE
2541	011522	000205			4\$:	RTS	R5	:LEAVE THE ROUTINE.
2542								
2543								
2544								

2545						
2546	011524	010537	014056		RXSTRA:	MOV R5,SAVEPC
2547	011530	012537	014064			MOV (R5)+,COUNT
2548	011534	012537	014066			MOV (R5)+,BITSEL
2549	011540	013737	014060	017772		MOV WORD,TEMP
2550	011546	005137	017772			COM TEMP
2551	011552	105077	167612			CLRB @DQREG
2552	011556	012777	014120	167606		MOV #RXBUFF,@DQSEC
2553	011564	105277	167600			INCB @DQREG
2554	011570	012777	000200	167574		MOV #200,@DQSEC
2555	011576	112777	000011	167564		MOVB #11,@DQREG
2556	011604	012777	177777	167560		MOV #-1,@DQSEC
2557	011612	105277	167552			INCB @DQREG
2558	011616	053777	014066	167546		BIS BITSEL,@DQSEC
2559	011624	052777	000012	167540		BIS #12,@DQSEC
2560	011632	105737	012604			TSTB PARFLG
2561	011636	001403				BEQ .+10
2562	011640	052777	100000	167524		BIS #BIT15,@DQSEC
2563	011646	052777	000001	167504		BIS #0001,@DQCSR
2564	011654	005737	011770			TST NPRFLG
2565	011660	001403				BEQ .+10
2566	011662	052777	010000	167470		BIS #BIT12,@DQCSR
2567	011670	112777	000012	167472		MOVB #MISC.,@DQREG
2568	011676	042777	000200	167466	2\$:	BIC #BIT7,@DQSEC
2569	011704	006037	017772			ROR TEMP
2570	011710	106037	001244			RORB TEMP1
2571	011714	042737	177577	001244		BIC #177577,TEMP1
2572	011722	053777	001244	167442		BIS TEMP1,@DQSEC
2573	011730	005277	167436			INC @DQSEC
2574	011734	005377	167432			DEC @DQSEC
2575	011740	005337	014064			DEC COUNT
2576	011744	001354				BNE 2\$
2577	011746	005737	011770			TST NPRFLG
2578	011752	001003				BNE .+10
2579	011754	052777	000020	167410		BIS #BIT4,@DQSEC
2580	011762	005037	011770			CLR NPRFLG
2581	011766	000205				RTS R5
2582	011770	000000				
2583	011772				NPRFLG: 0	
2584	011772	005077	167362		.MEMCLR:	CLR @DQCSR
2585	011776	005077	167362			CLR @DQCSR
2586	012002	005077	167360			CLR @DQERR
2587	012006	012705	000020			MOV #16,R5
2588	012012	152777	000020	167350	1\$:	BISB #BIT4,@DQREG
2589	012020	142777	000140	167342		BICB #140,@DQREG
2590	012026	005077	167340			CLR @DQSEC
2591	012032	105277	167332			INCB @DQREG
2592	012036	005305				DEC R5
2593	012040	001364				BNE 1\$
2594	012042	105077	167322			CLRB @DQREG
2595	012046	105077	167310			CLRB @DQRC5H
2596	012052	012705	000020			MOV #16,R5
2597	012056	112777	000010	167304	2\$:	MOVB #10,@DQREG
2598	012064	005077	167302			CLR @DQSEC
2599	012070	112777	000014	167272		MOVB #'',@DQREG
2600	012076	005077	167270			CLR @DQSEC

:IS PARITY TO BE TURNED ON?
 :BR IF NO
 :TURN PARITY ON.....

2601	012102	105277	167254		INCB	@DQPCSH
2602	012106	005305			DEC	R5
2603	012110	001362			BNE	2\$
2604	012112	105077	167244		CLRB	@DQRCSH
2605	012116				.MSTCLR:	
2606	012116	112777	000012	167244	MOVB	#MISC.,@DQREG
2607	012124	012777	000040	167240	MOV	#BIT5,@DQSEC
2608	012132	000002			RTI	
2609	012134	010537	014056		RXLNG: MOV	R5,SAVEPC
2610	012140	104412			MSTCLR	
2611	012142	105077	167222		CLRB	@DQREG
2612	012146	012777	014120	167216	MOV	#RXBUFF,@DQSEC
2613	012154	005037	014120		CLR	RXBUFF
2614	012160	105277	167204		INCB	@DQREG
2615	012164	012777	000200	167200	MOV	#200,@DQSEC
2616	012172	112777	000011	167170	MOVB	#11,@DQREG
2617	012200	013777	014522	167164	MOV	.SYNC,@DQSEC
2618	012206	105277	167156		INCB	@DQREG
2619	012212	012577	167154		MOV	(R5)+,@DQSEC
2620	012216	052777	000012	167146	BIS	#12,@DQSEC
2621	012224	052777	000001	167126	BIS	#0001,@DQRCR
2622	012232	042777	000200	167132	BIC	#BIT7,@DQSEC
2623	012240	005277	167126		INC	@DQSEC
2624	012244	005377	167122		DEC	@DQSEC
2625	012250	052777	000020	167114	BIS	#BIT4,@DQSEC
2626	012256	000240			NOP	
2627	012260	000240			NOP	
2628	012262	000240			NOP	
2629	012264	000337	014120		SWAB	RXBUFF
2630	012270	122537	014120		1\$: CMPB	(R5)+,RXBUFF
2631	012274	001401			BEQ	+.4
2632	012276	104015			HLT	15
2633	012300	005205			INC	R5
2634	012302	000205			RTS	R5
2635	012304	010537	014056		RXELNG: MOV	R5,SAVEPC
2636	012310	104412			MSTCLR	
2637	012312	105077	167052		CLRB	@DQREG
2638	012316	012777	014120	167046	MOV	#RXBUFF,@DQSEC
2639	012324	005037	014120		CLR	RXBUFF
2640	012330	105277	167034		INCB	@DQREG
2641	012334	012777	000200	167030	MOV	#200,@DQSEC
2642	012342	112777	000011	167020	MOVB	#11,@DQREG
2643	012350	013777	014522	167014	MOV	.SYNC,@DQSEC
2644	012356	105277	167006		INCB	@DQREG
2645	012362	012577	167004		MOV	(R5)+,@DQSEC
2646	012366	052777	000012	166776	BIS	#12,@DQSEC
2647	012374	052777	000001	166756	BIS	#0001,@DQRCR
2648	012402	042777	000200	166762	BIC	#BIT7,@DQSEC
2649	012410	005277	166756		INC	@DQSEC
2650	012414	005377	166752		DEC	@DQSEC
2651	012420	052777	000020	166744	BIS	#BIT4,@DQSEC
2652	012426	000240			NOP	
2653	012430	000240			NOP	
2654	012432	000240			NOP	
2655	012434	022537	014120		CMP	(R5)+,RXBUFF
2656	012440	001401			BEQ	+.4

DQ11 TRANSMITTER AND RECEIVER EXERCISER.

2657	012442	104015				HLT 15
2658	012444	000205				RTS R5
2659	012446				GENPAR:	
2660	012446	010146				MOV R1,-(SP)
2661	012450	010246				MOV R2,-(SP)
2662	012452	010346				MOV R3,-(SP)
2663	012454	105737	014054			TSTB EXTFLG
2664	012460	001003				BNE .+10
2665	012462	042737	000200	014060		BIC #BIT7,WORD
2666	012470	042737	100000	014060		BIC #BIT15,WORD
2667	012476	005002				CLR R2
2668	012500	012703	000020			MOV #16.,R3
2669	012504	013701	014060			MOV WORD,R1
2670	012510	000241				CLC
2671	012512	006001			1\$:	ROR R1
2672	012514	005502				ADC R2
2673	012516	005303				DEC R3
2674	012520	001374				BNE 1\$
2675						
2676	012522	032737	001000	001510		BIT #ODDBIT,DQSTAT
2677	012530	001404				BEQ 2\$
2678	012532	032702	000001			BIT #BIT0,R2
2679	012536	001016				BNE 4\$
2680	012540	000403				BR 3\$
2681	012542	032702	000001		2\$:	BIT #BIT0,R2
2682	012546	001412				BEQ 4\$
2683	012550	105737	014054		3\$:	TSTB EXTFLG
2684	012554	001004				BNE .+12
2685	012556	052737	000200	014060		BIS #BIT7,WORD
2686	012564	000403				BR 4\$
2687	012566	052737	100000	014060		BIS #BIT15,WORD
2688	012574	012603			4\$:	MOV (SP)+,R3
2689	012576	012602				MOV (SP)+,R2
2690	012600	012601				MOV (SP)+,R1
2691	012602	000207				RTS PC
2692	012604	000000			PARFLG: 0	

```

2693 012606          .ERRTAB:
2694 012606 000000      0
2695 012610 000000      0      ;HALT 0
2696 012612 000000      0
2697 012614 013012      EMO
2698 012616 013330      DH1      ;HALT 1
2699 012620 000000      0
2700 012622 013012      EMO
2701 012624 013351      DH2      ;HALT 2
2702 012626 000000      0
2703 012630 013030      EM1
2704 012632 013366      DH3      ;HALT 3
2705 012634 000000      0
2706 012636 013030      EM1
2707 012640 013377      DH4      ;HALT 4
2708 012642 000000      0
2709 012644 013030      EM1
2710 012646 013433      DH5      ;HALT 5
2711 012650 000000      0
2712 012652 013012      EMO
2713 012654 013467      DH6      ;HALT 6
2714 012656 000000      0
2715 012660 013012      EMO
2716 012662 013507      DH7      ;HALT 7
2717 012664 000000      0
2718 012666 013175      EM6
2719 012670 013542      DH9      ;HALT 10
2720 012672 000000      0
2721 012674 000000      0
2722 012676 013536      DH8      ;HALT 11
2723 012700 000000      0
2724 012702 013012      EMO
2725 012704 013550      DH10     ;HALT 12
2726 012706 014042      DT0
2727 012710 013066      EM3
2728 012712 013542      DH9      ;HALT 13
2729 012714 000000      0
2730 012716 013066      EM3
2731 012720 013536      DH8      ;HALT 14
2732 012722 000000      0
2733 012724 013175      EM6
2734 012726 013677      DH13     ;HALT 15
2735 012730 000000      0
2736 012732 013152      EM5
2737 012734 013536      DH8      ;HALT 16
2738 012736 000000      C
2739 012740 013152      FM5
2740 012742 013542      DH9      ;HALT 17
2741 012744 000000      0
2742 012746 013175      EM6
2743 012750 013550      DH10     ;HALT 20
2744 012752 014042      DT0
2745 012754 013120      EM4
2746 012756 000000      0      ;HALT 21
2747 012760 000000      0
2748 012762 013210      EM7

```

```

2749 012764 000000 0 :HALT 22
2750 012766 000000 0
2751 012770 013256 EMB
2752 012772 000000 0 :HALT 23
2753 012774 000000 0
2754 012776 013012 EMO
2755 013000 013310 DH0 :HALT 24
2756 013002 000000 0
2757 013004 000000 0
2758 013006 013550 DH10 :HALT 25
2759 013010 014042 DTO
2760 013012 052377 040522 051516 EMO: .ASCIZ <377>/TRANSMITTER /
013030 052377 040522 051516 EM1: .ASCIZ <377>/TRANSMITTER CHARACTER COUNT /
013066 053377 041522 042440 EM3: .ASCIZ <377>/VRC ERROR BIT SHOULD BE /
013120 047377 020117 044103 EM4: .ASCIZ <377>/NO CHARACTERS DETECTED./<0>
013152 051777 047131 020103 EM5: .ASCIZ <377>/SYNC 1 AND 2 NOT /
013175 377 042522 042503 EM6: .ASCIZ <377>/RECEIVER /
013210 041777 040510 040522 EM7: .ASCIZ <377>/CHARACTER DETECTION COMPARISON ERROR/
013256 041777 040510 040522 EM8: .ASCIZ <377>/CHARACTER NOT ALL ZERO'S/
013310 041501 044524 042526 DH0: .ASCIZ /ACTIVE NOT SET./
013330 041501 044524 042526 DH1: .ASCIZ /ACTIVE NOT CLEAR/
013351 104 047117 020105 DH2: .ASCIZ /DONE NOT SET/
013366 047516 020124 042532 DH3: .ASCIZ /NOT ZERO/
013377 116 052117 044440 DH4: .ASCIZ /NOT INCREMENTED BY PLUS TWO/
013433 116 052117 044440 DH5: .ASCIZ /NOT INCREMENTED BY PLUS ONE/
013467 120 044522 051452 DH6: .ASCIZ /PRI*SEC NOT SET/
013507 114 047111 020105 DH7: .ASCIZ /LINE NOT AT MARK STATE/
013536 042523 000124 DH8: .ASCIZ /SET/
013542 046103 040505 000122 DH9: .ASCIZ /CLEAR/
013550 040504 040524 041440 DH10: .ASCII /DATA COMPARISON ERROR/
013575 377 054105 042520 .ASCIZ <377>/EXPECTED RECEIVED /
013623 123 052105 053440 DH11: .ASCIZ /SET WHEN ACTIVE SET/
013647 103 042514 051101 DH12: .ASCIZ /CLEARED BY MASTER CLEAR/
013677 103 040510 040522 DH13: .ASCIZ /CHARACTER LENGTH ERROR/
013726 051777 042505 040440 MDETC: .ASCII <377>/SEE ABSTRACT OR TEST #56 FOR DETAILS/
013773 377 044103 051101 .ASCII <377>/CHARACTERS DETECTED: /
014022 041777 040510 027122 .ASCIZ <377>/CHAR. ADDR. /

.EVEN
2761 014042 000002 DTO: 2
014044 006 004 .BYTE 6,4
2762 014046 001254 TEMP5
2763 014050 006 002 .BYTE 6,2
2764 014052 001252 TEMP4
2765 014054 000000 EXTFLG: 0
2766 014056 000000 SAVEPC: 0
2767 014060 000000 WORD: 0
2768 014062 000000 DELAY: 0
2769 014064 000000 COUNT: 0
2770 014066 000000 BITSEL: 0
2771 014070 000012 TMPBUF: .BLKW 12
2772 014114 000000 XYZFLG: 0
2773 014116 000000 NUMBER: 0
2774 014120 000000 RXBUFF: 0
2775 014522 .=. +400
2776 014522 026 026 .SYNC: .BYTE 26,26
2777 014524 026 026 SYNC: .BYTE 26,26

```

```

2778 014526 000000 TXBUFF: 0
2779 015130 015130      .=.+400
2780
2781      :END OF PASS
2782      :TYPE NAME OF TEST
2783      :UPDATE PASS COUNT
2784      :CHECK FOR EXIT TO ACT-11
2785      :RESTART TEST
2786
2787 015130 005037 001234 .EOP: CLR LSTERR      :CLEAR LAST ERROR PC
2788 015134 005037 001312 CLR ERRFLG      :CLEAR ERROR FLAG
2789 015140 005237 001230 INC PASCNT      :UPDATE PASS COUNT
2790 015144 104402      TYPE
2791 015146 017360      MEPASS
2792 015150 104402      TYPE
2793 015152 017540      MCSRX
2794 015154 104411      CNVRT
2795 015156 015266      XCSR
2796 015160 104402      TYPE
2797 015162 017546      MVECX
2798 015164 104411      CNVRT
2799 015166 015274      XVEC
2800 015170 104402      TYPE
2801 015172 017554      MPASSX
2802 015174 104411      CNVRT
2803 015176 015301      XPASS
2804 015200 104402      TYPE
2805 015202 017565      MERRX
2806 015204 104411      CNVRT
2807 015206 015310      XERR
2808 015210 013777 001230 163764 MOV PASCNT,@LIGHTS :DISPLAY PASS COUNT
2809 015216 005337 001276 DEC SAVNUM
2810 015222 001013      BNE RESTR
2811 015224 013737 001504 001276 MOV DQNUM,SAVNUM
2812 015232 013701 000042 MOV @#42,R1      :CHECK FOR ACT-11 OR DDP
2813 015236 001405      BEQ RESTR      :IF NOT, CONTINUE TESTING
2814 015240 000005      RESET
2815 015242      LOGICAL:
2816 015242 004711      JSR PC,(R1)
2817 015244 000240      NOP
2818 015246 000240      NOP
2819 015250 000240      NOP
2820 015252 104414      RESTR: CKSWR
2821 015254 012737 002254 001214 MOV #TST1,RETURN
2822 015262 000137 002254 JMP TST1
2823 015266 000001      XCSR: 1
2824 015270 006 002 .BYTE 6,2
2825 015272 001360      DQRCSR
2826 015274 000001      XVEC: 1
2827 015276 003 002 .BYTE 3,2
2828 015300 001350      DQRVEC
2829 015302 000001      XPASS: 1
2830 015304 006 002 .BYTE 6,2
2831 015306 001230      'ERR: PASCNT
2832 015310 000001      1
2833 015312 006 002 .BYTE 6,2

```

```

2834 015314 001232          ERRCNT
2835
2836                          ;SCOPE LOOP AND INTERATION HANDLER
2837
2838 015316 104414          .SCOPE: CKSWR
2839 015320 032777 040000 163652 BIT      #BIT14,@SWR
2840 015326 001407          TTST:  BEQ      1$
2841 015330 000432          BR      3$
2842 015332 105777 163646 TSTB   @TKCSR
2843 015336 100027          BPL   3$
2844 015340 017700 163642 MOV    @TKDBR,R0
2845 015344 000412          BR    2$
2846 015346 032777 004000 163624 1$:  BIT    #SW11,@SWR
2847 015354 001006          BNE   2$
2848 015356 005237 001224 INC    LPCNT
2849 015362 023737 001224 001222 CMP    LPCNT,I COUNT
2850 015370 001012          BNE   3$
2851 015372 105037 001312          2$:  CLRB  ERRFLG
2852 015376 005037 001224 CLR    LPCNT
2853 015402 012737 000010 001222 MOV    #10,I COUNT
2854 015410 013737 001216 001214 MOV    NEXT,RETURN
2855 015416 013716 001214          3$:  MOV    RETURN,(SP)
2856 015422 000002          RTI
2857 015424 001407          BRW:  1407
2858 015426 000432          BRX:  432
2859
2860                          ;CHECK FOR FREEZE ON CURRENT DATA
2861
2862 015430 104414          .SCOPE1: CKSWR
2863 015432 032777 001000 163540 BIT    #SW09,@SWR
2864 015440 001402          BEQ    1$
2865 015442 013716 001220 MOV    LOCK,(SP)
2866 015446 000002          1$:  RTI
2867
2868                          ;TELETYPE OUTPUT ROUTINE
2869
2870 015450 010546          .TYPE:  MOV    R5,-(SP)
2871 015452 017605 000002 MOV    @2(SP),R5
2872 015456 062766 000002 000002 ADD    #2,2(SP)
2873 015464 005737 017140          1$:  TST   @#RDSW
2874 015470 001004          BNE   300$
2875 015472 032777 010000 163500 BIT    #SW12,@SWR
2876 015500 001024          BNE   3$
2877 015502 105715          300$: TSTB  (R5)
2878 015504 100014          BPL   2$
2879 015506 105777 163476 TSTB  @TPCSR
2880 015512 100375          BPL  -4
2881 015514 012777 000015 163470 MOV    #15,@TPDBR
2882 015522 105777 163462 TSTB  @TFCSR
2883 015526 100375          BPL  -4
2884 015530 012777 000012 163454 MOV    #12,@TPDBR
2885 015536 105777 163446          2$:  TSTB  @TPCSR
2886 015542 100375          BPL   2$
2887 015544 112577 163442 MOVB  (R5)+,@TPDBR
2888 015550 001345          BNE   1$
2889 015552 012605          3$:  MOV   (SP)+,R5

```



```

2890 015554 000002          RTI
2891
2892                          ;ASCII STRING INPUT ROUTINE
2893
2894 015556 010346          .INSTR: MOV     R3,-(SP)
2895 015560 010446          MOV     R4,-(SP)
2896 015562 017637 000004 015600 MOV     @4(SP),.MSG
2897 015570 062766 000002 000004 ADD     #2,4(SP)
2898 015576 104402          .INST1: TYPE
2899 015600 000000          .MSG: 0
2900 015602 012704 017730 MOV     #INBUF,R4
2901 015606 012703 000007 MOV     #7,R3
2902 015612 105777 163366 1$:      TSTB   @TKCSR
2903 015616 100375          BPL    1$
2904 015620 117714 163362 MOVB   @TKDBR,(R4)
2905 015624 142714 000200 BICB   #200,(R4)
2906 015630 121427 000025 CMPB   (R4),#25          ;IS IT <^G>
2907 015634 001003          BNE   200$
2908 015636 104402 017320 TYPE,MCRLF
2909 015642 000755          BR    .INST1
2910 015644 122427 000015 200$:  CMPB   (R4)+,#15
2911 015650 001423          BEQ   INSTR2
2912 015652 117777 163330 163332 MOVB   @TKDBR,@TPDBR
2913 015660 105777 163324 2$:      TSTB   @TPCSR
2914 015664 100375          BPL   2$
2915 015666 005303          DEC   R3
2916 015670 001350          BNE   1$
2917 015672 000402          BR    .INSTG
2918 015674 010346          .INSTE: MOV    R3,-(SP)
2919 015676 010446          MOV    R4,-(SP)
2920 015700 104402          .INSTG: TYPE
2921 015702 017314          MQM
2922 015704 005737 017140 TST    @#RDSW
2923 015710 001402          BEQ   400$
2924 015712 104402 017320 TYPE,MCRLF
2925 015716 000727          400$: BR    .INST1
2926 015720 012604 INSTR2: MOV    (SP)+,R4
2927 015722 012603 MOV    (SP)+,R3
2928 015724 000002          RTI
2929
2930                          ;CONVERT ASCII STRING TO OCTAL
2931
2932 015726 010546          .PARAM: MOV    R5,-(SP)
2933 015730 010446          MOV    R4,-(SP)
2934 015732 016605 000004 MOV    4(SP),R5
2935 015736 012537 016132 MOV    (R5)+,LOLIM
2936 015742 012537 016134 MOV    (R5)+,HILIM
2937 015746 012537 016136 MOV    (R5)+,DEVADR
2938 015752 112537 016140 MOVB   (R5)+,LOBITS
2939 015756 112537 016141 MOVB   (R5)+,ADRCNT
2940 015762 010566 000004 MOV    R5,4(SP)
2941 015766 005005          PARAM1: CLR   R5
2942 015770 012704 017730 MOV    #INBUF,R4
2943 015774 122714 000015 CMPB   #15,(R4)
2944 016000 001420          BEQ   PERR
2945 016002 121427 000060 1$:      CMPB   (R4),#60
  
```

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

2946	016006	002415			BLT	PARERR	
2947	016010	121427	000067		CMPB	(R4),#67	
2948	016014	003012			BGT	PARERR	
2949	016016	142714	000060		BICB	#60,(R4)	
2950	016022	152405			BISB	(R4)+,R5	
2951	016024	122714	000015		CMPB	#15,(R4)	
2952	016030	001414			BEQ	LIMITS	
2953	016032	006305			ASL	R5	
2954	016034	006305			ASL	R5	
2955	016036	006305			ASL	R5	
2956	016040	000760			BR	1\$	
2957	016042	122714	000015		PARERR: CMPB	#15,(R4)	;IS FIRST CHARACTER A <CR>
2958	016046	001003			BNE	120\$	
2959	016050	005737	017140		TST	WARD\$W	;IS CKSWR ROUTINE BEING USED
2960	016054	001023			BNE	PARTI	
2961	016056	104404			120\$: INSTER		
2962	016060	000742			BR	PARAM1	
2963							
2964							;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2965							
2966	016062	020537	016134		LIMITS: CMP	R5,HILIM	
2967	016066	101365			BHI	PARERR	
2968	016070	020537	016132		CMP	R5,LOLIM	
2969	016074	103762			BLO	PARERR	
2970	016076	133705	016140		BITB	LOBITS,R5	
2971	016102	001357			BNE	PARERR	
2972							
2973							;STORE NUMBER AT SPECIFIED ADDRESS
2974							
2975	016104	013704	016136		1\$: MOV	DEVADR,R4	
2976	016110	010524			MOV	R5,(R4)+	
2977	016112	062705	000002		ADD	#2,R5	
2978	016116	105337	016141		DECB	ADRCNT	
2979	016122	001372			BNE	1\$	
2980	016124	012604			PARTI: MOV	(SP)+,R4	
2981	016126	012605			MOV	(SP)+,R5	
2982	016130	000002			RTI		
2983	016132	000000			LOLIM: 0		
2984	016134	000000			HILIM: 0		
2985	016136	000000			DEVADR: 0		
2986	016140	000000			LOBITS: 0		
2987		016141			ADRCNT-LOBITS+1		
2988							
2989							;SAVE PC OF TEST THAT FAILED AND R0-R5
2990							
2991	016142	016637	000004	001274	.SAV05: MOV	4(SP),SAVPC	
2992							
2993							;SAVE R0-R5
2994							
2995	016150	010537	001270		SV05: MOV	R5,SAVR5	
2996	016154	010437	001266		MOV	R4,SAVR4	
2997	016160	010337	001264		MOV	R3,SAVR3	
2998	016164	010237	001262		MOV	R2,SAVR2	
2999	016170	010137	001260		MOV	R1,SAVR1	
3000	016174	010037	001256		MOV	R0,SAVR0	
3001	016200	000002			RTI		

CZDQD MACY11 30A(1052) 07-JUL-78 08:33 PAGE 59
 CZDQDE.P11 22-JUN-78 08:42 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

SEQ 0058
 SEQ 0058

```

3002
3003                ;RESTORE R0-R5
3004
3005 016202 013700 001256      .RES05: MOV     SAVR0,R0
3006 016206 013701 001260      MOV     SAVR1,R1
3007 016212 013702 001262      MOV     SAVR2,R2
3008 016216 013703 001264      MOV     SAVR3,R3
3009 016222 013704 001266      MOV     SAVR4,R4
3010 016226 013705 001270      MOV     SAVR5,R5
3011 016232 000002              RTI
3012
3013                ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
3014
3015 016234 104402      .CONVR: TYPE
3016 016236 017320      MCRLF
3017 016240 010046      .CNVRT: MOV     R0,-(SP)
3018 016242 010146      MOV     R1,-(SP)
3019 016244 010346      MOV     R3,-(SP)
3020 016246 010446      MOV     R4,-(SP)
3021 016250 010546      MOV     R5,-(SP)
3022 016252 017601 000012      MOV     @12(SP),R1
3023 016256 013737 017772 001250      MOV     TEMP,TEMP3
3024 016264 062766 000002 000012      ADD     #2,12(SP)
3025 016272 012137 016454      MOV     (R1)+,WRDCNT
3026 016276 112137 016456      1$:   MOV     (R1)+,CHRCNT
3027 016302 112137 016457      MOV     (R1)+,SPACNT
3028 016306 013137 016460      MOV     @(R1)+,BINWRD
3029 016312 013704 016460      2$:   MOV     BINWRD,R4
3030 016316 113705 016456      MOV     CHRCNT,R5
3031 016322 012700 017772      MOV     #TEMP,R0
3032 016326 010403      3$:   MOV     R4,R3
3033 016330 042703 177770      BIC     #177770,R3
3034 016334 062703 000060      ADD     #060,R3
3035 016340 110320      MOV     R3,(R0)+
3036 016342 000241      CLC
3037 016344 006004      ROR     R4
3038 016346 000241      CLC
3039 016350 006004      ROR     R4
3040 016352 000241      CLC
3041 016354 006004      ROR     R4
3042 016356 005305      DEC     R5
3043 016360 001362      BNE     3$
3044 016362 012703 020034      MOV     #MDATA,R3
3045 016366 114023      4$:   MOV     -(R0),(R3)+
3046 016370 105337 016456      DECB   CHRCNT
3047 016374 001374      BNE     4$
3048 016376 105737 016457      TSTB   SPACNT
3049 016402 001405      BEQ     6$
3050 016404 112723 000040      5$:   MOV     #0,(R3)+
3051 016410 105337 016457      DECB   SPACNT
3052 016414 001373      BNE     5$
3053 016416 105013      6$:   CLRB   (R3)
3054 016420 104402      TYPE
3055 016422 020034      MDATA
3056 016424 005337 016454      DEC     WRDCNT
3057 016430 001322      BNF     1$

```

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

```
3058 016432 013737 001250 017772      MOV      TEMP3,TEMP
3059 016440 012605                      MOV      (SP)+,R5
3060 016442 012604                      MOV      (SP)+,R4
3061 016444 012603                      MOV      (SP)+,R3
3062 016446 012601                      MOV      (SP)+,R1
3063 016450 012600                      MOV      (SP)+,R0
3064 016452 000002                      RTI
3065 016454 000000                      WRDCNT: 0
3066 016456 000000                      CHR CNT: 0
3067                                SPACNT=CHR CNT+1
3068 016460 000000                      BINWRD: 0
3069                                ;TRAP DISPATCH SERVICE
3070                                ;ARGUMENT OF TRAP IS EXTRACTED
3071                                ;AND USED AS OFFSET TO OBTAIN POINTER
3072                                ;TO SELECTED SUBROUTINE
3073
3074 016462 011646                      .TRPSR: MOV      (SP),-(SP)          ;GET PC OF RETURN
3075 016464 162716 000002                      SUB      #2,(SP)                ;=PC OF TRAP
3076 016470 017616 000000                      MOV      @ (SP), (SP)          ;GET TRP
3077 016474 006316                      TRPOK:  ASL      (SP)           ;MULTIPLY TRAP ARG BY 2
3078 016476 042716 177001                      BIC      #177001,(SP)         ;CLEAR UNWANTED BITS
3079 016502 062716 001314                      ADD      #.TRPTAB,(SP)       ;PCINTER TO SUBROUTINE ADDRESS
3080 016506 017616 000000                      MOV      @ (SP), (SP)       ;SUBROUTINE ADDRESS
3081 016512 000136                      JMP      @ (SP)+             ;GO TO SUBROUTINE
3082
3083                                ;ERROR HANDLER
3084
3085 016514 104414                      .H.L.T: CKSWR
3086 016516 032777 010000 162454                      BIT      #SW12,@SWR
3087 016524 001406                      BEQ      XBX
3088 016526 105777 162456                      TSTB    @TPCSR
3089 016532 100003                      BPL      XBX
3090 016534 112777 000207 162450                      MOV      #207,@TPDBR
3091 016542 032777 020000 162430                      XBX:    BIT      #SW13,@SWR
3092 016550 001074                      BNE      HALTS
3093 016552 021637 001234                      CMP      (SP),LSTERR
3094 016556 001404                      BEQ      1$
3095 016560 011637 001234                      MOV      (SP),LSTERR
3096 016564 105037 001312                      CLRB    ERRFLG
3097 016570 104406                      1$:    SAVO5
3098 016572 011605                      MOV      (SP),R5
3099 016574 162705 000002                      SUB      #2,R5
3100 016600 011504                      MOV      (R5),R4
3101 016602 006304                      ASL      R4
3102 016604 061504                      ADD      (R5),R4
3103 016606 006304                      ASL      R4
3104 016610 042704 177001                      BIC      #177001,R4
3105 016614 062704 012606                      ADD      #.ERRTAB,R4
3106 016620 012437 016712                      MOV      (R4)+,ERRMSG
3107 016624 012437 016724                      MOV      (R4)+,DATAHD
3108 016630 011437 016736                      MOV      (R4),DATABP
3109 016634 105737 001312                      TSTB    ERRFLG
3110 016640 001403                      BEQ      TYPMSG
3111 016642 005737 016736                      TST     DATABP
3112 016646 001027                      BNF     T'DAT
3113 016650 104402                      TYPMSG: TYPE
```

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

3114	016652	017576			MTSTN	
3115	016654	104411			CNVRT	
3116	016656	017036			XTSTN	
3117	016660	104402			TYPE	
3118	016662	017664			MERRPC	
3119	016664	104411			CNVRT	
3120	016666	017030			ERTABO	
3121	016670	104402			TYPE	
3122	016672	017320			MCRLF	
3123	016674	112737	177777	001312	MOVB	#-1,ERRFLG
3124	016702	005737	016712		TST	ERRMSG
3125	016706	001402			BEQ	WRKO.FM
3126	016710	104402			TYPE	
3127	016712	000000			ERRMSG: 0	
3128	016714				WRKO.FM:	
3129	016714	005737	016724		TST	DATAHD
3130	016720	001402			BEQ	TYPDAT
3131	016722	104402			TYPE	
3132	016724	000000			DATAHD: 0	
3133	016726	005737	016736		TYPDAT: TST	DATABP
3134	016732	001402			BEQ	RESREG
3135	016734	104410			CONVRT	
3136	016736	000000			DATABP: 0	
3137	016740	104407			RESREG: RES05	
3138	016742	005777	162232		HALTS: TST	@SWR
3139	016746	100005			BPL	EXITER
3140	016750	010046			PUSHRO	
3141	016752	016600	000002		MOV	2(SP),R0
3142	016756	000000			HALT	
3143	016760	012600			POPPO	
3144	016762	104414			EXITER: CKSWR	
3145	016764	005237	001232		INC	ERRCNT
3146	016770	032777	000400	162202	BIT	#SW08,@SWR
3147	016776	001007			BNE	1\$
3148	017000	032777	002000	162172	BIT	#SW10,@SWR
3149	017006	001407			BEQ	2\$
3150	017010	013737	001216	001214	MOV	NEXT,RETURN
3151	017016	012706	001200		1\$: MOV	#STACK,SP
3152	017022	000177	162166		JMP	@RETURN
3153	017026	000002			2\$: RTI	
3154	017030	000001			ERTABO: 1	
3155	017032	006	002		.BYTE	6,2
3156	017034	001274			SAVPC	
3157	017036	000001			XTSTN: 1	
3158	017040	003	002		.BYTE	3,2
3159	017042	001226			TSTNO	
3160					;ENTER HERE ON POWER FAILURE	
3161						
3162						
3163	017044				.PFAIL:	
3164	017044	012737	017056	000024	MOV	#RESTART,24 ;SET UP FOR POWER UP TRAP
3165	017052	000000			HALT	;HALT ON POWER DOWN NORMAL
3166	017054	000777			BR	.
3167						
3168					;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED	
3169						

```

3170 017056
3171 017056 012737 017044 000024 RESTAR:
3172 017064 012706 001200 MOV #.PFAIL,24 ;SET UP FOR POWER FAILURE
3173 017070 005037 017772 MOV #STACK,SP
3174 017074 005237 017772 CLR TEMP
3175 017100 001375 INC TEMP
3176 017102 104402 BNE .-4
3177 017104 017322 TYPE
3178 017106 104411 MPFAIL
3179 017110 017132 CNVRT
3180 017112 005037 001312 PFTAB
3181 017116 005037 001234 CLR ERRFLG
3182 017122 104412 CLR LSTERR
3183 017124 104413 MSTCLR
3184 017126 000177 162062 MEMCLR
3185 017132 000001 JMP @RETURN
3186 017134 003 002 PFTAB: 1
3187 017136 001226 .BYTE 3,2
3188
3189
3190
3191 ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ^G TO ALLOW CHANGING
3192 ;OF LOC.176.
3193 017140 000000 ;LOCATIONS USED:
3194 RDSW: .WORD 0
3195
3196 017142 005737 000042 .CKSWR: TST @#42
3197 017146 001042 BNE OUT
3198 017150 022737 000176 001200 CMP #SWREG,SWR ;SOFTWARE SWITCH REGISTER PRESENT
3199 017156 001036 BNE OUT ;NO, GET OUT
3200 017160 105777 162020 TSTB @TKCSR ;YES, WAIT FOR
3201 017164 100033 BPL OUT ;READY, GET CHARACTER
3202 017166 017737 162014 015600 MOV @TKDBR,.MSG ;AND STRIP OFF
3203 017174 042737 177600 015600 BIC #177600,.MSG ;THE GARBAGE
3204 017202 122737 000007 015600 CMPB #7,.MSG ;IS IT A <^G>
3205 017210 001021 BNE OUT
3206 017212 104402 017270 TYPE,SCNTG
3207 017216 005137 017140 .CNTLU: COM @#RDSW
3208 017222 104402 017274 TYPE,SMSWR
3209 017226 104411 017262 CNVRT,SWREGC
3210 017232 104403 017303 INSTR,SMNEW
3211 017236 104405 PARAM
3212 017240 000000 0
3213 017242 177777 177777
3214 017244 000176 SWREG
3215 017246 000 001 .BYTE 0,1
3216 017250 104402 017320 TYPE,MCRLF
3217 017254 005037 017140 OUT: CLR @#RDSW
3218 017260 000002 RTI
3219 017262 000001 SWREGC: 1
3220 017264 006 002 .BYTE 6,2
3221 017266 000176 SWREG
3222 017270 057377 000107 SCNTG: .ASCIZ <377>/^G/
3223 017274 051777 051127 020075 SMSWR: .ASCIZ <377>/SWR- /
3224 017302 000
3225 017303 040 047040 053505 SMNEW: .ASCIZ / NEW- /

```

3226	017310	020075	000		
3227		017314			.EVEN
3228	017314	020040	000077		MQM: .ASCIZ / ?/
3229	017320	000377			MCRLF: .ASCIZ <377>
3230	017322	050377	051127	043040	MPFAIL: .ASCIZ <377>/PWR FAILED. RESTART AT TEST /
3231	017330	044501	042514	027104	
3232	017336	051040	051505	040524	
3233	017344	052122	040440	020124	
3234	017352	042524	052123	000040	
3235	017360	042777	042116	050040	MEPASS: .ASCIZ <377>/END PASS CZDQD /
3236	017366	051501	020123	055103	
3237	017374	050504	020104	000	
3238	017401	377	000122		MR: .ASCIZ <377>/R/
3239	017404	050377	047522	051107	MERR2: .ASCIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT./
3240	017412	046501	044440	042116	
3241	017420	041511	052101	051505	
3242	017426	047040	020117	042504	
3243	017434	044526	042503	020123	
3244	017442	051120	051505	047105	
3245	017450	027124	000		
3246	017453	377	047111	052523	MERR3: .ASCIZ <377>/INSUFFICIENT DATA! /
3247	017460	043106	041511	042511	
3248	017466	052116	042040	052101	
3249	017474	020501	000		
3250	017477	377	042524	052123	MTSTPC: .ASCIZ <377>/TEST PC-/
3251	017504	050040	026503	000	
3252	017511	377	047514	045503	MLOCK: .ASCIZ <377>/LOCK ON SELECTED TEST/
3253	017516	047440	020116	042523	
3254	017524	042514	052103	042105	
3255	017532	052040	051505	000124	
3256	017540	051503	035122	000040	MCSRX: .ASCIZ /CSR: /
3257	017546	042526	035103	000040	MVECX: .ASCIZ /VEC: /
3258	017554	040520	051523	051505	MPASSX: .ASCIZ /PASSES: /
3259	017562	020072	000		
3260	017565	105	051122	051117	MERRX: .ASCIZ /ERRORS: /
3261	017572	035123	000040		
3262	017576	177777	042524	052123	MTSTN: .ASCIZ <377><377> /TEST NO: /
3263	017604	047040	035117	000040	
3264	017612	051777	052105	051440	MNEW: .ASCIZ <377>/SET SWITCH REG TO DQ11'S DESIRED ACTIVE./
3265	017620	044527	041524	020110	
3266	017626	042522	020107	047524	
3267	017634	042040	030521	023461	
3268	017642	020123	042504	044523	
3269	017650	042522	020104	041501	
3270	017656	044524	042526	000056	
3271	017664	041520	020072	000	MERRPC: .ASCIZ /PC: /
3272	017671	377	040515	020120	XHEAD: .ASCIZ <377>/MAP OF DQ11 STATUS/<377>
3273	017676	043117	042040	030521	
3274	017704	020061	052123	052101	
3275	017712	051525	000377		
3276					.EVEN
3277	017716	000002			XSTATQ: 2
3278	017720	006	003		.RYTE 6.3
3279	017722	001244			TEMP1
3280	017724	006	002		.BYTE 6.2
3281	017726	001246			TEMP2

```
3282          .EVEN
3283
3284          ;BUFFERS FOR INPUT-OUTPUT
3285
3286 017730 000000  INBUF: 0
3287          017772  .=. +40
3288 017772 000000  TEMP: 0
3289          020034  .=. +40
3290 020034 000000  MDATA. 0
3291          020076  . . +40
3292          000001  .END
```


RXSTRA	011524	1866	1890	1915	2023	2070	2310	2546#										
RXWC.P=	000001	622#																
RXWC.S=	000005	626#																
RX.BCC=	000015	635#																
SAVACT	001502	755*	925#	992														
SAVEPC	014056	2444*	2478*	2504*	2546*	2609*	2635*	2766#										
SAVNUM	001276	748*	820#	940*	2809*	2811*												
SAVPC	001274	819#	2991*	3156														
SAVR0	001256	812#	3000*	3005														
SAVR1	001260	813#	2999*	3006														
SAVR2	001262	814#	2998*	3007														
SAVR3	001264	815#	2997*	3008														
SAVR4	001266	816#	2996*	3009														
SAVR5	001270	817#	2995*	3010														
SAVSP	001272	818#																
SAV05 =	104406	853#	3097															
SCOPE =	104400	841#	1123	1210	1228	1244	1260	1276	1292	1308	1324	1345	1362	1379				
		1396	1413	1430	1447	1464	1526	1569	1606	1622	1633	1644	1655	1666				
		1677	1688	1704	1715	1726	1737	1748	1759	1770	1781	1806	1830	1843				
		1963	1998	2040	2081	2173	2243	2404										
SCOPI =	104401	843#	1563	1600	2034	2078	2335											
SECND	003222	1185#																
SEQ. =	000014	634#																
SPACNT=	016457	3027*	3048	3051*	3067#													
STACK =	001200	577#	938	1013	3151	3172												
STFLG	001311	829#	941*															
SV05	016150	2995#																
SWR	001200	783#	951*	956	960*	971	974	985	992	998	1017	1025	2285	2839				
		2846	2863	2875	3086	3091	3138	3146	3148	3198								
SWREG	000176	691#	960	971	3198	3214	3221											
SWREGC	017262	3209	3219#															
SW00 =	000001	557#	985															
SW01 =	000002	556#	1025															
SW02 =	000004	555#																
SW03 =	000010	554#																
SW04 =	000020	553#																
SW05 =	000040	552#																
SW06 =	000100	551#																
SW08 =	000400	550#	3146															
SW09 =	001000	549#	2863															
SW10 =	002000	548#	3148															
SW11 =	004000	547#	2846															
SW12 =	010000	546#	2875	3086														
SW13 =	020000	545#	3091															
SW14 =	040000	544#																
SW15 =	100000	543#																
SYNBIT=	100000	616#	713	1131	1878													
SYNC	014524	1133*	1135*	2113	2204	2777#												
SYNC. =	000011	631#																
SYNTST	005630	1826	1839	1852#														
TEMP	017772	2549*	2550*	2569*	3023	3031	3058*	3173*	3174*	3288#								
TEMP1	001244	670*	671*	807#	978*	979	983*	2120*	2126*	2156*	2161*	2211*	2217*	2570*				
		2571*	2572	3279														
TEMP2	001246	808#	979*	2121*	2128*	2157*	2163*	2212*	2219*	3281								
TEMP3	001250	809#	1496*	1524*	3023*	3058												
TEMP4	001252	810#	1510*	1511*	1516*	1519*	1521	1551*	1560	1588*	1597	2026*	2031	2073*				

CROSS REFERENCE TABLE -- USER SYMBOLS

TST54	007034	2056	2095#															
TST55	007436	2096	2186#															
TST56	007756	2187	2258#	2781														
TST57 =	***** U	2259																
TST6	003446	1240	1255#															
TST7	003474	1256	1271#															
TTST	015326	1020*	1021*	1023*	1024*	2840#												
TXBA.P=	000002	623#																
TXBA.S=	000006	627#																
TXBUFF	014526	1154	1171	1186	1203	2098	2189	2232	2778#									
TXSTRB	010722	1225	1241	1257	1273	1289	1305	1321	1342	1359	1376	1393	1410	1427				
		1444	1461	2442#														
TXSTRC	011126	2478#																
TXSTRD	011270	1553	1590	2504#														
TXWC.P=	000003	624#																
TXWC.S=	000007	628#																
TX.BCC=	000016	636#																
TX.MUX=	000013	633#																
TYPDAT	016726	3112	3130	3133#														
TYPE =	104402	761	845#	969	977	987	994	1019	1037	1062	2360	2374	2790	2792				
		2796	2800	2804	2898	2908	2920	2924	3015	3054	3113	3117	3121	3126				
		3131	3176	3206	3208	3216												
		3110	3113#															
TYPMSG	016650	663#	760															
VECMAP	000056	1497*	1503	1550*	1559	1586*	1587*	1596	1853*	1914*	2019*	2027	2063*	2065				
WORD	014060	2067*	2069*	2074	2309*	2443*	2448	2482	2508	2549	2665*	2666*	2669	2685*				
		2687*	2767#															
		3025*	3056*	3065#														
WRDCNT	016454	3125	3128#															
WRKO.F	016714	3087	3089	3091#														
XBX	016542	2377	2405#															
XCHAR1	010632	2381	2410#															
XCHAR2	010644	2385	2415#															
XCHAR3	010656	2389	2420#															
XCHAR4	010670	2795	2823#															
XCSR	015266	2807	2832#															
XERR	015310	977	3272#															
XHEAD	017671	2803	2829#															
XPASS	015302	982	3277#															
XSTATQ	017716	3116	3157#															
XTSTN	017036	2799	2826#															
XVEC	015274	2362	2392*	2772#														
XYZFLG	014114	3206	3222#															
\$CNTG	017270	1#	1043	1044#	1104#	1150	1151#	1224	1225#	1240	1241#	1256	1257#	1272				
\$E -	000060	1273#	1288	1289#	1304	1305#	1320	1321#	1341	1342#	1358	1359#	1375	1376#				
		1392	1393#	1409	1410#	1426	1427#	1443	1444#	1460	1461#	1478	1479#	1545				
		1547#	1581	1583#	1618	1619#	1629	1630#	1640	1641#	1651	1652#	1662	1663#				
		1673	1674#	1684	1685#	1700	1701#	1711	1712#	1722	1723#	1733	1734#	1744				
		1745#	1755	1756#	1766	1767#	1777	1778#	1792	1793#	1825	1826#	1838	1839#				
		1944	1945#	1979	1980#	2013	2015#	2056	2058#	2096	2097#	2187	2188#	2259				
		2261#																
\$MNEW	017303	3210	3225#															
\$MSWR	017274	3208	3223#															
\$N	000056	1#	1039	1044#	1100	1104#	1146	1151#	1221	1225#	1237	1241#	1253	1257#				
		1269	1273#	1285	1289#	1301	1305#	1317	1321#	1338	1342#	1355	1359#	1372				
		1376#	1389	1393#	1406	1410#	1423	1427#	1440	1444#	1457	1461#	1475	1479#				

CROSS REFERENCE TABLE -- USER SYMBOLS

		1542	1547#	1578	1583#	1615	1619#	1626	1630#	1637	1641#	1648	1652#	1659
		1663#	1670	1674#	1681	1685#	1697	1701#	1708	1712#	1719	1723#	1730	1734#
		1741	1745#	1752	1756#	1763	1767#	1774	1778#	1789	1793#	1822	1826#	1835
		1839#	1941	1945#	1976	1980#	2010	2015#	2053	2058#	2093	2097#	2184	2188#
		2256	2261#	2781#										
SY	- 000016	1#	832#	841	843#	845#	847#	849#	851#	853#	855#	857#	859#	861#
	020076	863#	865#	867#	869#									
		641#	642	644#	651#	653#	661#	677	689#	695#	698#	702	718	723
		727	730	765	770#	780#	821#	891#	892#	893#	894#	895#	896#	897#
		898#	899#	900#	901#	902#	903#	904#	905#	906#	907#	908#	909#	910#
		911#	912#	913#	914#	915#	916#	917#	918#	919#	920#	921#	922#	923#
		924#	925#	926#	927#	928#	975	980	984	997	1065	1115	1121	1168
		1172	1175	1200	1204	1207	1481	1485	1489	1494	1522	1557	1561	1594
		1598	1799	1804	1867#	1868#	1871	1874	1879	1885	1888	1891#	1892#	1896
		1899	1906	1911	1916#	1917#	1919	1924	1954	1957	1961	1989	1992	1996
		2021	2029	2032	2066	2068	2076	2134	2169	2223	2226	2229	2239	2264
		2304	2359	2391	2397	2400	2463	2470	2515	2561	2565	2578	2631	2656
		2664	2684	2771#	2775#	2779#	2880	2883	3166	3175	3227#	3287#	3289#	3291#
.ADDR1	010704	2366*	2409	2426#										
.ADDR2	010710	2368*	2414	2428#										
.ADDR3	010714	2370*	2419	2430#										
.ADDR4	010720	2372*	2424	2432#										
.BEGIN	002116	1012#												
.CHAR1	010702	2365*	2395	2407	2425#									
.CHAR2	010706	2367*	2412	2427#										
.CHAR3	010712	2369*	2417	2429#										
.CHAR4	010716	2371*	2422	2431#										
.CKSWR	017142	866	3196#											
.CNTLU	017216	868	3207#											
.CNVRT	016240	860	3017#											
.CONVR	016234	858	3015#											
.EOP	015130	2259	2787#											
.ERRTA	012606	2693#	3105											
.HLT	016514	647	3085#											
.INSTE	015674	850	2918#											
.INSTG	015700	2917	2920#											
.INSTR	015556	848	2894#											
.INST1	015576	2898#	2909	2925										
.MEMCL	011772	864	2583#											
.MSG	015600	2896*	2899#	3202*	3203*	3204								
.MSTCL	012116	862	2605#											
.PARAM	015726	852	2932#											
.PFAIL	017044	645	939	3163#	3171									
.RES05	016202	856	3005#											
.SAV05	016142	854	2991#											
.SCOPE	015316	842	2838#											
.SCOPI	015430	844	2862#											
.START	001512	696	937#	949										
.SYNC	014522	1499	2117	2208	2617	2643	2776#							
.TRPSR	016462	649	3074#											
.TRPTA	001314	840#	3079											
.TYPE	015450	846	2870#											

. ABS. 020076 000

CZDQD MACY11 30A(1052) 07-JUL-78 08:33 PAGE 74
CZDQDE.P11 22-JUN-78 08:42 CROSS REFERENCE TABLE -- USER SYMBOLS

H 6

SEQ 0072
SEQ 0072

ERRORS DETECTED: 0

DSKZ:CZDQDE,DSKZ:CZDQDE,SEQ-DSKZ:CZDQXX.MAC,DSKZ:CZDQDE.P11
RUN-TIME: 7 11 1 SECONDS
RUN-TIME RATIO: 43/20 2.1
CORE USED: 18k (35 PAGES)

DOCUMENT PAGES: 72