

# DQ11

REC AND TRANSTEST  
MD-11-DZDQD-C

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

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

This microfiche card contains a grid of frames. The frames are arranged in approximately 12 rows and 6 columns. Each frame contains a small, high-contrast image or data set, likely representing a specific test result or a segment of a recording. The content is too small to read clearly but appears to be organized in a structured manner. A small white mark is visible near the bottom center of the card.

BU1

000000.F11

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDGD-C-0  
PRODUCT NAME: RECEIVER AND TRANSMITTER TESTS  
DATE: 21 JUNE 1976  
MAINTAINER: DIAGNOSTIC GROUP

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

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

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

CONFIDENTIAL © 1974, 1976 BY DIGITAL EQUIPMENT CORPORATION

## ABSTRACT

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

THIS TEST TEST TRANSMITTER AND RECEIVER CHARACTER LENGTHS FROM 00 TO 16 BITS PER CHARACTER.  
ALSO DATA RELIABILITY FOR TRANSMITTER, RECEIVER AND TRANSMITTER AND RECEIVER TOGETHER.  
CABLE TEST TRANSFERS 400 CHARACTERS THROUGH THE CABLE TO VERIFY CABLE.

WHEN THE PROGRAM ENTERS TEST #56 ON EACH FIRST TIME AFTER STARTING OR IF THERE ARE MULTIPLE DQ11'S UNDER TEST, A MESSAGE WILL BE PRINTED:  
"CHARACTERS DETECTED"  
"CHAR ADDRESS"

THIS TEST IS DONE ONLY IF THE DQ11-BB OPTION IS NOT INSTALLED. THIS TEST IS DETERMINING THE STRAP-SELECTABLE CHARS ON THE M7818 MODULE. DEFAULT CHAR AND ADDRESS IS "CHAR 17777" AND "ADDRESS 17". THIS MAY BE CHANGED AS PER CUSTOMER PREFERENCES AND SHOULD BE PRINTED OUT ACCORDINGLY. IF THERE IS ONLY ONE DQ11 UNDER TEST THIS MESSAGE WILL BE PRINTED ONLY ONCE AFTER EACH START OF PROGRAM. IF THERE ARE MULTIPLE DQ11'S THIS WILL BE PRINTED EACH TIME THROUGH THE TEST. THE ABOVE DESCRIBED MESSAGE IS \*NOT\* AN ERROR BUT MUST BE VERIFIED TO "WHAT WAS PRINTED OUT MATCHES THE M7818 MODULE". SEE TEST #56 FOR MORE DETAIL.

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. DZ00A (REV) BASIC R/W TEST #1
2. DZ00B (REV) BASIC R/W TEST #2
3. DZ00C (REV) BASIC N/F AND INTERRUPT TEST
4. DZ00D (REV) RECEIVER TRANSMITTER EXERCISER TEST
5. DZ00E (REV) MISC. RX AND TX TESTS. PLUS BCC TESTS.
6. DZ00F (REV) CHARACTER DETECT TESTS.
7. DZ00H (REV) CHARACTER LENGTH AND INTERRUPT TESTS.

THERE IS ALSO AN ONLINE TEST TO BE DISCUSSED LATER.  
1. DZ000 (REV) ONLINE TEST. (ITEP OVERLAY)

AND A PARAMETER INPUT PROGRAM IS AVAILABLE  
1. D0100 (REV) D011 TRIAL PROGRAM (PARAMETER INPUT,  
REQUIREMENTS

2.

## 2.1 EQUIPMENT

ANY PDP11 FAMILY CPU (WITH MINIMUM 8K MEMORY)-WITH  
OR WITHOUT A HARDWARE SWITCH REGISTER (LOC. 177570)  
ASR 33 (OR EQUIVALENT)

D011  
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 D011 TRIAL PROGRAM HAS BEEN EXECUTED.  
OR AFTER THE "AUTO SIZING" HAS BEEN DONE.

## 3. LOADING PROCEDURE

## 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	27
16K	37
24K	47
32K	57
40K	67
48K	77
56K	87
64K	97
72K	107
80K	117
88K	127
96K	137
104K	147
112K	157

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

3.1.2 THEN START

## 4. STARTING PROCEDURE

A. LOAD LOC. 200  
B. SET SWR TO ZERO FOR "AUTO SIZING" OR LEAVE  
LEAVE SWR BIT 7=1 TO USE EXISTING PARAMETERS SET UP  
BY D011 TRIAL PROGRAM OR A PREVIOUSLY RUN D011 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

3. 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	167020
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 (1G): 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-DQ AUTO SIZE.  
 SW 06 SET:  
 SW 05 SET:  
 SW 04 SET:  
 SW 03 SET:  
 SW 02 SET: LOCK ON SELECTED TEST  
 SW 01 SET: RESTART PROGRAM AT SELECTED TEST  
 SW 00 SET: RESELECT DQ11'S DESIRED ACTIVE.

#### 4.1.2 SWITCH REGISTER RESTRICTIONS

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

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

SW 01 IT IS STRONGLY SUGGESTED THAT AT LEAST ONE PASS HAS BEEN MADE BEFORE TRYING TO SELECT A TEST

THAT IS NOT IN THE ORDER OF SEQUENCE  
THE REASON BEING IS THAT THE  
PROGRAM HAS TO CLEAR AREAS AND SET  
UP PARAMETERS. ALSO WHEN A TEST IS  
SELECTED ALWAYS START AT THE VERY  
BEGINNING OF THAT TEST.

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

#### 4.1.3 SWITCH REGISTER PRIORITYS

##### ERROR SWITCHES

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

\*\*\*HLT (ERROR) ROUTINE SUPPORTS (IG) OPERATION\*\*\*

##### SCOPE SWITCHES

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

\*\*\*SCOPE ROUTINE WILL SUPPORT (IG) 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

## 5. 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.

### 5.2 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 "TSTNC" (ADDRESS 1222) FOR THE NUMBER OF THE TEST THAT WAS RUNNING AT THE TIME OF THE CATASTROPHIC ERROR.

IN THIS WAY THE OPERATOR WILL HAVE AN IDEA AS TO WHAT THE DQ11 WAS DOING AT THE TIME OF THE ERROR.

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

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

## 7. RESTRICTIONS

### 7.1 STARTING RESTRICTIONS

SEE SECTION 4. (PLEASE)

### 7.2 OPERATING RESTRICTIONS



DQ11 TRIAL PROGRAM MUST BE RUN PRIOR TO THE FIRST AND ONLY THE FIRST RUNNING OF ANY DQ11 DIAGNOSTIC

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

9. MISCELLANEOUS

9.1 EXECUTION TIME

9.2 PASS COMPLETE

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

END PASS DZDQD-C CSR: 160000 VEC: 300 PASSES: 000001 ERRORS: 000000

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

9.3 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!!!!!!!

9.4 KEY LOCATIONS

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

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

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

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

EXAMPLE:  
(RUN) 1272/0000000001000000  
MEANS THAT DQ11 NO.05 IS THE DQ11 NOW RUNNING.

DQCP00-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 APPRIORATE 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?

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 APPRIORATE 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 APPRIORATE 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

L01

DZDQC MACY11 27(732) 24-MAY-75 13:10 PAGE 12  
DZDQDC.P11

MO1

549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585

.ENABLE AMA

:MAINDEC-11-DZDQD-C/<377>/TRANSMITTER AND RECEIVER EXERCISER  
:COPYRIGHT 1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

:REVISED 21-JUNE-76 BY S. CARPENTER

: A)SUPPORTS SOFTWARE SWITCH REGISTER  
: B)SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER  
: BY <↑G>.

:STARTING PROCEDURE

:LOAD PROGRAM

:LOAD ADDRESS 000200

:PRESS START

:PROGRAM WILL TYPE "MAINDEC-11-DZDQD-C/<377>/TRANSMITTER AND RECEIVER EXERCISER"

:PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED

:AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE

:AND THEN RESUME TESTING

:SWITCH REGISTER OPTIONS

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

SW15=100000  
SW14=40000  
SW13=20000  
SW12=10000  
SW11=4000  
SW10=2000  
SW09=1000  
SW08=400  
SW06=100  
SW05=40  
SW04=20  
SW03=10  
SW02=4  
SW01=2  
SW00=1

:=1,HALT ON ERROR  
:=1,LOOP ON CURRENT TEST  
:=1,INHIBIT ERROR TYPEOUT  
:=1,DELETE TYPEOUT/BELL ON ERROR.  
:=1,INHIBIT ITERATIONS  
:=1,ESCAPE TO NEXT TEST ON ERROR  
:=1,LOOP WITH CURRENT DATA  
:=1,LOOP ON ERROR

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

586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640

```

:REGISTER DEFINITIONS
000000 R0=%0 :GENERAL REGISTER
000001 R1=%1 :GENERAL REGISTER
000002 R2=%2 :GENERAL REGISTER
000003 R3=%3 :GENERAL REGISTER
000004 R4=%4 :GENERAL REGISTER
000005 R5=%5 :GENERAL REGISTER
000006 SP=%6 :PROCESSOR STACK POINTER
000007 PC=%7 :PROGRAM COUNTER

;LOCATION EQUIVALENCIES
177570 DSWR= 177570 ;HARDWARE SWITCH REGISTER LOC.
177570 DLIGHTS=177570 ;HARDWARE DISPLAY REGISTER LOC.
177776 PS=177776 ;PROCESSOR STATUS WORD
001200 STACK=1200 ;START OF PROCESSOR STACK

;INSTRUCTION DEFINITIONS
005746 PUSH1SP=5746 ;DECREMENT PROCESSOR STACK 1 WORD
005726 POP1SP=5726 ;INCREMENT PROCESSOR STACK 1 WORD
010046 PUSHRO=10046 ;SAVE R0 ON STACK
012600 POPRO=12600 ;RESTORE R0 FROM STACK
024646 PUSH2SP=24646 ;DECREMENT STACK TWICE
022626 POP2SP=22626 ;INCREMENT STACK TWICE
.EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL

100000 BIT15=100000
040000 BIT14=40000
020000 BIT13=20000
010000 BIT12=10000
004000 BIT11=4000
002000 BIT10=2000
001000 BIT9=1000
000400 BIT8=400
000200 BIT7=200
000100 BIT6=100
000040 BIT5=40
000020 BIT4=20
000010 BIT3=10
000004 BIT2=4
000002 BIT1=2
000001 BIT0=1

;DQ11 OPTIONAL DEFINITIONS
002000 ABBIT=2000
004000 ACTBIT=4000
010000 BABIT=10000
020000 BEBIT=20000
040000 JMBIT=40000
  
```









000334	000336	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000336	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000340	000342	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000342	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000344	000346	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000346	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000350	000352	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000352	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000354	000356	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000356	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000360	000362	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000362	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000364	000366	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000366	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000370	000372	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000372	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000374	000376	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000376	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000400	000402	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000402	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000404	000406	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000406	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000410	000412	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000412	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000414	000416	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000416	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000420	000422	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000422	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000424	000426	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000426	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000430	000432	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000432	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000434	000436	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000436	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000440	000442	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000442	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000444	000446	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000446	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000450	000452	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000452	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000454	000456	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000456	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000460	000462	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000462	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000464	000466	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000466	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000470	000472	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000472	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000474	000476	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000476	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000500	000502	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000502	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000504	000506	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000506	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000510	000512	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000512	000000	HALT	: EXAMINE STACK TO FIND CAUSE

835	000514	000516	.+2	:UNEXPECTED TRAP TO THIS LOCATION
836	000516	000500	HALT	:EXAMINE STACK TO FIND CAUSE
837	000520	000522	.+2	:UNEXPECTED TRAP TO THIS LOCATION
838	000522	000000	HALT	:EXAMINE STACK TO FIND CAUSE
839	000524	000526	.+2	:UNEXPECTED TRAP TO THIS LOCATION
840	000526	000500	HALT	:EXAMINE STACK TO FIND CAUSE
841	000530	000532	.+2	:UNEXPECTED TRAP TO THIS LOCATION
842	000532	000000	HALT	:EXAMINE STACK TO FIND CAUSE
843	000534	000536	.+2	:UNEXPECTED TRAP TO THIS LOCATION
844	000536	000500	HALT	:EXAMINE STACK TO FIND CAUSE
845	000540	000542	.+2	:UNEXPECTED TRAP TO THIS LOCATION
846	000542	000500	HALT	:EXAMINE STACK TO FIND CAUSE
847	000544	000546	.+2	:UNEXPECTED TRAP TO THIS LOCATION
848	000546	000000	HALT	:EXAMINE STACK TO FIND CAUSE
849	000550	000552	.+2	:UNEXPECTED TRAP TO THIS LOCATION
850	000552	000000	HALT	:EXAMINE STACK TO FIND CAUSE
851	000554	000556	.+2	:UNEXPECTED TRAP TO THIS LOCATION
852	000556	000000	HALT	:EXAMINE STACK TO FIND CAUSE
853	000560	000562	.+2	:UNEXPECTED TRAP TO THIS LOCATION
854	000562	000000	HALT	:EXAMINE STACK TO FIND CAUSE
855	000564	000566	.+2	:UNEXPECTED TRAP TO THIS LOCATION
856	000566	000000	HALT	:EXAMINE STACK TO FIND CAUSE
857	000570	000572	.+2	:UNEXPECTED TRAP TO THIS LOCATION
858	000572	000000	HALT	:EXAMINE STACK TO FIND CAUSE
859	000574	000576	.+2	:UNEXPECTED TRAP TO THIS LOCATION
860	000576	000000	HALT	:EXAMINE STACK TO FIND CAUSE
861	000600	000602	.+2	:UNEXPECTED TRAP TO THIS LOCATION
862	000602	000000	HALT	:EXAMINE STACK TO FIND CAUSE
863	000604	000606	.+2	:UNEXPECTED TRAP TO THIS LOCATION
864	000606	000000	HALT	:EXAMINE STACK TO FIND CAUSE
865	000610	000612	.+2	:UNEXPECTED TRAP TO THIS LOCATION
866	000612	000000	HALT	:EXAMINE STACK TO FIND CAUSE
867	000614	000616	.+2	:UNEXPECTED TRAP TO THIS LOCATION
868	000616	000000	HALT	:EXAMINE STACK TO FIND CAUSE
869	000620	000622	.+2	:UNEXPECTED TRAP TO THIS LOCATION
870	000622	000000	HALT	:EXAMINE STACK TO FIND CAUSE
871	000624	000626	.+2	:UNEXPECTED TRAP TO THIS LOCATION
872	000626	000000	HALT	:EXAMINE STACK TO FIND CAUSE
873	000630	000632	.+2	:UNEXPECTED TRAP TO THIS LOCATION
874	000632	000000	HALT	:EXAMINE STACK TO FIND CAUSE
875	000634	000636	.+2	:UNEXPECTED TRAP TO THIS LOCATION
876	000636	000000	HALT	:EXAMINE STACK TO FIND CAUSE
877	000640	000642	.+2	:UNEXPECTED TRAP TO THIS LOCATION
878	000642	000000	HALT	:EXAMINE STACK TO FIND CAUSE
879	000644	000646	.+2	:UNEXPECTED TRAP TO THIS LOCATION
880	000646	000000	HALT	:EXAMINE STACK TO FIND CAUSE
881	000650	000652	.+2	:UNEXPECTED TRAP TO THIS LOCATION
882	000652	000000	HALT	:EXAMINE STACK TO FIND CAUSE
883	000654	000656	.+2	:UNEXPECTED TRAP TO THIS LOCATION
884	000656	000000	HALT	:EXAMINE STACK TO FIND CAUSE
885	000660	000662	.+2	:UNEXPECTED TRAP TO THIS LOCATION
886	000662	000000	HALT	:EXAMINE STACK TO FIND CAUSE
887	000664	000666	.+2	:UNEXPECTED TRAP TO THIS LOCATION
888	000666	000000	HALT	:EXAMINE STACK TO FIND CAUSE
889	000670	000672	.+2	:UNEXPECTED TRAP TO THIS LOCATION
890	000672	000000	HALT	:EXAMINE STACK TO FIND CAUSE

891	000674	000676	.+2	:UNEXPECTED TRAP TO THIS LOCATION
892	000676	000000	HALT	:EXAMINE STACK TO FIND CAUSE
893	000700	000702	.+2	:UNEXPECTED TRAP TO THIS LOCATION
894	000702	000000	HALT	:EXAMINE STACK TO FIND CAUSE
895	000704	000706	.+2	:UNEXPECTED TRAP TO THIS LOCATION
896	000706	000000	HALT	:EXAMINE STACK TO FIND CAUSE
897	000710	000712	.+2	:UNEXPECTED TRAP TO THIS LOCATION
898	000712	000000	HALT	:EXAMINE STACK TO FIND CAUSE
899	000714	000716	.+2	:UNEXPECTED TRAP TO THIS LOCATION
900	000716	000000	HALT	:EXAMINE STACK TO FIND CAUSE
901	000720	000722	.+2	:UNEXPECTED TRAP TO THIS LOCATION
902	000722	000000	HALT	:EXAMINE STACK TO FIND CAUSE
903	000724	000726	.+2	:UNEXPECTED TRAP TO THIS LOCATION
904	000726	000000	HALT	:EXAMINE STACK TO FIND CAUSE
905	000730	000732	.+2	:UNEXPECTED TRAP TO THIS LOCATION
906	000732	000000	HALT	:EXAMINE STACK TO FIND CAUSE
907	000734	000736	.+2	:UNEXPECTED TRAP TO THIS LOCATION
908	000736	000000	HALT	:EXAMINE STACK TO FIND CAUSE
909	000740	000742	.+2	:UNEXPECTED TRAP TO THIS LOCATION
910	000742	000000	HALT	:EXAMINE STACK TO FIND CAUSE
911	000744	000746	.+2	:UNEXPECTED TRAP TO THIS LOCATION
912	000746	000000	HALT	:EXAMINE STACK TO FIND CAUSE
913	000750	000752	.+2	:UNEXPECTED TRAP TO THIS LOCATION
914	000752	000000	HALT	:EXAMINE STACK TO FIND CAUSE
915	000754	000756	.+2	:UNEXPECTED TRAP TO THIS LOCATION
916	000756	000000	HALT	:EXAMINE STACK TO FIND CAUSE
917	000760	000762	.+2	:UNEXPECTED TRAP TO THIS LOCATION
918	000762	000000	HALT	:EXAMINE STACK TO FIND CAUSE
919	000764	000766	.+2	:UNEXPECTED TRAP TO THIS LOCATION
920	000766	000000	HALT	:EXAMINE STACK TO FIND CAUSE
921	000770	000772	.+2	:UNEXPECTED TRAP TO THIS LOCATION
922	000772	000000	HALT	:EXAMINE STACK TO FIND CAUSE
923	000774	000776	.+2	:UNEXPECTED TRAP TO THIS LOCATION
924	000776	000000	HALT	:EXAMINE STACK TO FIND CAUSE

# H02

D2D00 MACY11 27(732) 24-MAY-75 13:10 PAGE 21  
 D2D00C.P11 ROUTINES USED FOR ALTO SIZING.

```

925                                     ;STANDARD INTERRUPT VECTORS
926
927                                     . =24
928 000024 017042 .PFAIL :POWER FAIL HANDLER
929 000026 000340 340 :SERVICE AT LEVEL 7
930 000030 016512 .HLT :ERROR HANDLER
931 000032 000340 340 :SERVICE AT LEVEL 7
932 000034 016460 .TRPSRV :GENERAL HANDLER DISPATCH SERVICE
933 000036 000340 340 :SERVICE AT LEVEL 7
934                                     . =46
935 000046 015240 LOGICAL ;ACT HOOKS
936                                     . =52
937 000052 000000 .WORD 0
938 :THIS ROUTINE TRIES TO FORCE THE RECEIVER TO INTERRUPT
939 :TO ITS VECTOR WHERE IT WILL PICK UP THE STATUS LOCATION
940 :FOR ITS NEW PC; AND PICK UP AN IOT INSTRUCTION FOR ITS
941 :NEW PS. WHEN THE NEW PC IS FETCHED AN IOT INSTRUCTION IS
942 :EXECUTED, TRAPPING TO LOCATION 20 WHERE A ROUTINE IS EXECUTED
943 :TO TAKE THE PC FROM THE STACK AND USE IT AS THE VECTOR ADDRESS
944 000056 . =56
945
946 000056 VECMAP:
947 000056 010120 1$: MOV R1,(R0)+ ;START FILLING THE VECTOR AREA
948 000060 012721 00C004 MOV #4,(R1)+ ;WITH .+2; IOT (4)
949 000064 022021 00C004 CMP (R0)+,(R1)+ ;UPDATE THE POINTERS
950 000066 020127 001000 CMP R1,#1000 ;IS ALL FLOATING VECTOR AREA DONE
951 000072 101771 1$ BLOS 1$ ;BR IF NOT ALL DONE
952 000074 012737 000146 000020 MOV #4$ ,0#20 ;SET FOR IOT TRAP BY DQ11
953 000102 013737 001500 001244 MOV DQACTV,TEMP1 ;GET THE ACTIVE DQ11 S
954 000110 006037 001244 2$: RCR TEMP1 ;ARE YOU ACTIVE.. DQ11
955 000114 103023 5$ BCC 5$ ;IF CARRY CLEAR.. NO MORE DQ11S
956 000116 005037 177776 CLR PS ;CLEAR PS
957 000122 005722 TST (R2)+ ;PUT POINTER TO STATUS TABLE
958 000124 012772 000340 177776 MOV #340,0-2 R2) ;TRY AND SET PRI/SEC DONE AND IE
959 000132 105200 INCB RC ;DELAY.....
960 000134 001376 BNE .-2 ;.....DELAY
961 000136 112712 000300 MOVB #300,(R2) ;NO INTERRUPT ASSUME 300 FIX IN TEST C
962 000142 005722 3$: TST (R2)+ ;UPDATE POINTERS
963 000144 000761 2$ BR 2$ ;GO DO IT AGAIN
964 000146 051612 4$: BIS (SP),(R2) ;ENTERD BY IOT TRAP BY DQ11
965 000150 042712 000007 BIC #7,(R2) ;CLEAR UNWANTED BITS
966 000154 022626 CMP (SP)+,(SP)+ ;POP IOT JUNK OFF STACK
967 000156 012716 000142 MOV #3$, (SP) ;SET RETURN PC ON STACK
968 000162 000002 RTI ;GO HOME
969 000164 000207 5$: RTS PC ;ALL SIZING IS DONE
970
971 :****SOFTWARE SWITCH REGISTER****
972 . =174
973 000174 000000 DISPREG: 0 ;SOFTWARE DISPLAY REGISTER
974 000176 000000 SWREG: 0 ;SOFTWARE SWITCH REGISTER
975
976 ;PROGRAM START
977
978 . =200
979 000200 000137 001512 JMP .START ;GO TO START OF PROGRAM
980

```



```

1037 000540 012737 000006 000004      MOV      #6,2#4      ;RESET TIME OUT VECTOR
1038 000546 013737 001500 001502      MOV      DQACTV, SAVACT ;SAVE ACTIVE
1039 000554 012737 000340 000022      MOV      #340,2#22   ;SET IOT TRAP PRIO: TO 7
1040 000562 012702 001400          MOV      #1400,R2    ;SET TABLE POINTER
1041 000566 012700 000300          MOV      #300,R0     ;SET VECTOR START
1042 000572 012701 000302          MOV      #302,R1     ;SET VECTOR+2 START
1043 000576 000137 000056          JMP      VECMAP      ;GO FIND THE VECTORS
1044 000602 104402          4$:      TYPE      ;TYPE MESSAGE
1045 000604 017402          MERR2    ;I DIDN'T FIND ANY DQ11S. DON'T USE AUTO SIZE.
1046 000606 005000          CLR      R0         ;
1047 000610 000000          HALT     ;HOW CAN I TEST NO DQ11S
1048 000612 000776          BR      -2         ;DON'T LET OPR HIT CONT. SW
1049 000614 012716 000466          5$:      MOV      #2$, (SP) ;ENTERED BY TIME OUT TRAP
1050 000620 000002          RTI     ;GO HOME.
1051
1052
1053          001000          .=1000
1054 001000 005377 040515 047111  MTITLE: .ASCIZ <377><12>/MAINDEC-11-DZDQD-C/<377>/TRANSMITTER AND RECEIVER EXERCISER/<3
1055 001006 042504 026503 030461
1056 001014 042055 042132 042121
1057 001022 041455 052377 040522
1058 001030 051516 044515 052124
1059 001036 051105 040440 042116
1060 001044 051040 041505 044505
1061 001052 042526 020122 054105
1062 001060 051105 044503 042523
1063 001066 177522          000
1064
1065          001200          .=1200
1066          ;INDIRECT POINTERS
1067
1068 001200 177570          SWR:      177570      ;SWITCH REGISTER POINTER
1069 001202 177570          LIGHTS:   177570     ;DISPLAY REGISTER POINTER
1070 001204 177560          TKCSR:    177560     ;TELETYPE KEYBOARD CONTROL REGISTER
1071 001206 177562          TKDBR:    177562     ;TELETYPE KEYBOARD DATA BUFFER
1072 001210 177564          TPCSR:    177564     ;TELEPRINTER CONTROL REGISTER
1073 001212 177566          TPDBR:    177566     ;TELEPRINTER DATA BUFFER
1074
1075          ;PROGRAM CONTROL PARAMETERS
1076
1077 001214 000000          RETURN:   0          ;SCOPE ADDRESS FOR LOOP ON TEST
1078 001216 000000          NEXT:     0          ;ADDRESS OF NEXT TEST TO BE EXECUTED
1079 001220 000000          LOCK:     0          ;ADDRESS FOR LOCK ON CURRENT DATA
1080 001222 000003          ICOUNT:   3          ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
1081 001224 000000          LPCNT:    0          ;NUMBER OF ITERATIONS COMPLETED
1082 001226 000000          TSTNO:    0          ;NUMBER OF TEST IN PROGRESS
1083 001230 000000          PASCNT:   0          ;NUMBER OF PASSES COMPLETED
1084 001232 000000          ERRCNT:   0          ;TOTAL NUMBER OF ERRORS
1085 001234 000000          LSTERR:   0          ;PC OF LAST ERROR CALL
1086
1087          ;PROGRAM VARIABLES
1088
1089 001236 000000          CHAR1:    0
1090 001240 000000          CHAR2:    0
1091 001242 000000          CHAR3:    0
1092 001244 000000          TEMP1:    0          ;TEMPORARY STORAGE

```

DZDQD MACY11 27(732) 24-MAY-76 13:10 PAGE 24  
DZDQDC.P11 PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

1093	001246	000000	TEMP2:	0	; TEMPORARY STORAGE
1094	001250	000000	TEMP3:	00	; TEMPORARY STORAGE
1095	001252	000000	TEMP4:	00	; TEMPORARY STORAGE
1096	001254	000000	TEMP5:	00	; TEMPORARY STORAGE
1097	001256	000000	SAVR0:	00	; R0 STORAGE
1098	001260	000000	SAVR1:	00	; R1 STORAGE
1099	001262	000000	SAVR2:	00	; R2 STORAGE
1100	001264	000000	SAVR3:	00	; R3 STORAGE
1101	001266	000000	SAVR4:	00	; R4 STORAGE
1102	001270	000000	SAVR5:	00	; R5 STORAGE
1103	001272	000000	SAVSP:	00	; STACK POINTER STORAGE
1104	001274	000000	SAVPC:	00	; PROGRAM COUNTER STORAGE
1105	001276	000000	SAVNUM:	0	
1106	001300	000001	CREAM:	.BLKW 1	
1107	001302	000000	RUNFLG:	0	
1108	001304	000000	RUN:	0	
1109	001306	000000	RUNCNT:	0	



```

1110
1111                ;PROGRAM CONTROL FLAGS
1112
1113 001310      000      INIFLG: .BYTE 0                ;PROGRAM INITIALIZATION FLAG
1114 001311      000      STFLG:  .BYTE 0                ;TEST START FLAG
1115 001312      000      ERRFLG: .BYTE 0                ;ERROR OCCURED FLAG
1116 001313      000      LOKFLG: .BYTE 0                ;LOCK ON CURRENT TEST FLAG
1117                SY=0
1118
1119                ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
1120                ;POINTERS TO SUBROUTINES CAN BE FOUND
1121                ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS
1122
1123                ;*****
1124                ;*****
1125 001314      TRPTAB:
1126                SCOPE=TRAP+0                ;CALL TO SCOPE LOOP AND ITERATION HANDLER
1127 001314      015314      .SCOPE
1128                SCOPI=TRAP+1                ;CALL TO LOOP ON CURRENT DATA HANDLER
1129 001316      015426      .SCOPI
1130                TYPE=TRAP+2                ;CALL TO TELETYPE OUTPUT ROUTINE
1131 001320      015446      .TYPE
1132                INSTR=TRAP+3                ;CALL TO ASCII STRING INPUT ROUTINE
1133 001322      015554      .INSTR
1134                INSTER=TRAP+4                ;CALL TO INPUT ERROR HANDLER
1135 001324      015672      .INSTER
1136                PARAM=TRAP+5                ;CALL TO NUMERICAL DATA INPUT ROUTINE
1137 001326      015724      .PARAM
1138                SAVOS=TRAP+6                ;CALL TO REGISTER SAVE ROUTINE
1139 001330      016140      .SAVOS
1140                RESOS=TRAP+7                ;CALL TO REGISTER RESTORE ROUTINE
1141 001332      016200      .RESOS
1142                CONVRT=TRAP+10                ;CALL TO DATA OUTPUT ROUTINE
1143 001334      016232      .CONVRT
1144                CNVRT=TRAP+11                ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.
1145 001336      016236      .CNVRT
1146                MSTCLR=TRAP+12                ;CALL TO ISSUE MASTER CLEAR
1147 001340      012114      .MSTCLR
1148                MEMCLR=TRAP+13                ;CALL TO CLEAR ALL SCRATCH PAD MEMORIES
1149 001342      011770      .MEMCLR
1150                CKSWR=TRAP+14                ;CALL TO ALLOW SWREG TO BE LOADED FROM TTY
1151 001344      017140      .CKSWR
1152                CNTLU=TRAP+15                ;CALL TO ALLOW LOADING OF SWREG FROM TTY
1153 001346      017214      .CNTLU
1154
1155                ;*****
1156                ;*****
1157
1158                ;DQ11 VECTOR AND REGISTER INDIRECT POINTERS
1159
1160 001350      000C00      DQRVEC: 0                ;POINTER TO DQ11 RECEIVER INTERRUPT VECTOR
1161 001352      000000      DQRLVL: 0                ;POINTER TO DQ11 RECEIVER INTERRUPT SERVICE PS
1162 001354      000000      DQTVEC: 0                ;POINTER TO DQ11 TRANSMITTER INTERRUPT VECTOR
1163 001356      000000      DQTLVL: 0                ;POINTER TO DQ11 TRANSMITTER INTERRUPT SERVICE PS
1164 001360      000000      DQRCRS: 0                ;POINTER TO DQ11 RECEIVER CONTROL REGISTER
1165 001362      000000      DQRCSH: 0                ;POINTER TO HIGH BYTE OF DQ11 RECEIVER CONTROL REGISTER

```

# M02

DZDQD MACY11 27(732) 24-MAY-75 13:10 PAGE 26  
 DZDQDC.P11 PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```

1166 001364 000000      DQTCR: 0      ; POINTER TO DQ11 TRANSMITTER CONTROL REGISTER
1167 001366 000000      DQERR: 0     ; POINTER TO DQ11 ERROR REGISTER
1168 001370 000000      DQREG: 0     ; POINTER TO HIGH BYTE OF ERROR REGISTER
1169 001372 000000      DQSEC: 0     ; POINTER TO DQ11 SECONDARY REGISTER
1170 001374 000000      DQSECH: 0    ; POINTER TO HIGH BYTE OF DQ11 SECONDARY REGISTER
1171
1172
1173
1174
1175
1176      001400
1177 001400 000001      . = 1400
1178 001402 000001      DQCR00: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 00
1179 001404 000001      DQST00: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 00
1180 001406 000001      DQCR01: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 01
1181 001410 000001      DQST01: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 01
1182 001412 000001      DQCR02: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 02
1183 001414 000001      DQST02: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 02
1184 001416 000001      DQCR03: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 03
1185 001420 000001      DQST03: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 03
1186 001422 000001      DQCR04: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 04
1187 001424 000001      DQST04: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 04
1188 001426 000001      DQCR05: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 05
1189 001430 000001      DQST05: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 05
1190 001432 000001      DQCR06: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 06
1191 001434 000001      DQST06: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 06
1192 001436 000001      DQCR07: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 07
1193 001440 000001      DQST07: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 07
1194 001442 000001      DQCR10: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 10
1195 001444 000001      DQST10: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 10
1196 001446 000001      DQCR11: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 11
1197 001448 000001      DQST11: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 11
1198 001450 000001      DQCR12: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 12
1199 001452 000001      DQST12: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 12
1200 001454 000001      DQCR13: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 13
1201 001456 000001      DQST13: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 13
1202 001460 000001      DQCR14: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 14
1203 001462 000001      DQST14: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 14
1204 001464 000001      DQCR15: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 15
1205 001466 000001      DQST15: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 15
1206 001470 000001      DQCR16: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 16
1207 001472 000001      DQST16: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 16
1208 001474 000001      DQCR17: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 17
1209 001476 000001      DQST17: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 17
1210 001500 000001      DQACTV: .BLKW 1 ; HOLD ACTIVE BITS FOR TESTING
1211 001502 000001      SAVACT: .BLKW 1 ; SAVE NUMBER OF ACTIVE DQ11S
1212 001504 000001      DQNUM: .BLKW 1 ; OCTAL NUMBER OF TOTAL NUMBER OF DQ11S
1213 001506 000001      DQCSR: .BLKW 1 ; CSR OF DQ11 UNDER TEST
1214 001510 000001      DQSTAT: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS OF DQ11 UNDER TEST
1215
1216      ; PROGRAM INITIALIZATION
1217      ; LOCK OUT INTERRUPTS
1218      ; SET UP PROCESSOR STACK
1219      ; SET UP POWER FAIL VECTOR
1220      ; CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1221      ; TYPE TITLE MESSAGE

```

# N02

DZDQD MACY11 27(732) 24-MAY-76 13:10 PAGE 27  
 DZDQDC.P11 PROGRAM INITIALIZATION AND START UP.

```

1222 001512 012737 000340 177776 .START: MOV #340,PS ;LOCK OUT INTERRUPTS
1223 001520 012706 001200 MOV #STACK,SP ;SET UP STACK
1224 001524 012737 017042 000024 MOV #.PFAIL,@#24 ;SET UP POWER FAIL VECTOR
1225 001532 013737 001504 001276 MOV DQNUM,SAVNUM
1226 001540 105037 001311 CLRB STFLG ;CLEAR START FLAG
1227 001544 005037 001230 CLR PASCNT ;CLEAR PASS COUNT
1228 001550 105037 001312 CLRB ERRFLG ;CLEAR ERROR FLAG
1229 001554 005037 001302 CLR RUNFLG
1230 001560 012737 001400 001300 MOV #1400,CREAM
1231 001566 005037 001232 CLR ERRCNT ;CLEAR ERROR COUNT
1232 001572 005037 001234 CLR LSTERR ;CLEAR LAST ERROR POINTER
1233 001576 012737 000001 001226 MOV #1,TSTNO ;SET UP FOR TEST 1
1234 001604 012737 001512 001214 MOV #.START,RETURN ;SET UP FOR POWER FAIL BEFORE
1235 ;TESTING STARTS
1236 001612 105737 001310 TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED
1237 001616 001075 BNE 12$
1238 001620 104402 001000 TYPE MTITLE ;TYPE TITLE MESSAGE
1239 001624 105137 001310 COMB INIFLG ;IF NOT SET FLAG AND DO
1240
1241 001630 012737 177570 001200 MOV #DSWR,SWR ;MOV HARDWARE SWR TO SWR
1242 001636 012737 177570 001202 MOV #DLIGHTS,LIGHTS ;MOV DISPLAY LIGHTS TO LIGHTS
1243 001644 013746 000006 MOV @#6,-(SP) ;SAVE VECTORS
1244 001650 013746 000004 MOV @#4,-(SP)
1245 001654 012737 001674 000004 MOV #64$,@#4 ;SET UP FOR TIMEOUT
1246 001662 022777 177777 177310 CMP #-1,@SWR ;REFERENCE HARDWARE SWITCH REGISTER
1247 001670 001402 BEQ 65$
1248 001672 000407 BR 66$
1249 001674 022626 64$: CMP (SP)+,(SP)+ ;ADJUST STACK
1250 001676 012737 000176 001200 65$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
1251 001704 012737 000174 001202 MOV #DISPREG,LIGHTS ;POINT TO SOFT DISPLAY REG
1252 001712 012637 000004 66$: MOV (SP)+,@#4 ;RESTORE VECTORS
1253 001716 012637 000006 MOV (SP)+,@#6
1254 001722 005737 000042 TST @#42 ;UNDER MONITOR
1255 001726 001005 BNE 67$
1256 001730 022737 000176 001200 CMP #SWREG,SWR ;IS SWREG USED
1257 001736 001001 BNE 67$
1258 001740 104415 CNTLU
1259 001742 105777 177232 67$: TSTB @SWR
1260 001746 100402 BMI .+6
1261 001750 004737 000220 JSR PC,CSRMAP
1262 001754 104402 017667 TYPE .XHEAD
1263 001760 012737 001400 001244 MOV #1400,TEMP1
1264 001766 017737 177252 001246 MOV @TEMP1,TEMP2
1265 001774 001406 BEQ .+16
1266 001776 104410 CONVRT
1267 002000 017714 XSTATQ
1268 002002 062737 000002 001244 ADD #2,TEMP1
1269 002010 000766 BR .-22
1270 002012 032777 000001 177160 12$: BIT #SW00,@SWR
1271 002020 001424 BEQ 1$
1272 002022 104402 TYPE
1273 002024 017610 MNEW
1274 002026 005000 CLR RG
1275 002030 000000 HALT
1276 002032 104414 CF SWR
1277 002034 027737 177140 001502 CMP @SWR,SAVACT

```



```

1334 002324 033737 001304 001500 15: BIT RUN,DGACTV ;FIND AN ACTIVE DQ11 TO TEST.
1335 002332 001032 001500 001500 35 BNE ;BR IF I FOUND ONE TO TEST.
1336 002334 005737 001500 001500 TST DGACTV ;FIND OUT IF THERE ARE NO DQ11 ACTIVE.
1337 002340 001423 001500 001500 BEQ 25 ;BR TO FATAL ERROR. WHY AM I HERE IF NO ACTIVE DQ11'S??
1338 002342 000257 001500 001500 CCC ;CLEAR ALL THE CONDITION CODES OF CPU
1339 002344 006137 001304 001304 ROL RUN ;UPDATE RUN POINTER
1340 002350 062737 000004 001300 ADD #4,CREAM ;UPDATE ADDRESS POINTER.
1341 002356 005337 001306 001306 DEC RUNCNT ;DEC NUMBER OF TIMES I LOOKED AT ACTIVE.
1342 002362 001360 000020 001306 BNE 15 ;BR AND KEEP LOOKING.
1343 002364 012737 000020 001306 MOV #16,RUNCNT ;START RESTORING MY POINTERS.
1344 002370 012737 001400 001300 MOV #1400,CREAM ;RESTORE ADDRESS POINTER
1345 002400 012737 000001 001304 MOV #1,RUN ;RESTORE RUN POINTER.
1346 002406 000746 000001 001304 BR 15 ;KEEP ON TESTING.
1347 002410 004402 000001 001304 BR 25 ;ALLERT OPERATOR OF FATAL ERROR
1348 002412 017402 000001 001304 MERR2 ;NO DQ11 ACTIVE. WHY AM I HERE??
1349 002414 000000 000001 001304 HALT ;YOU MUST RELOAD DQ11 DIAGNOSTIC!!
1350 002416 000776 000001 001304 BR -2 ;STICK HERE ON CONT.
1351 002420 000257 000001 001304 CCC ;CLEAR CPU COND. CODES
1352 002422 006137 001304 001506 ROL RUN ;UPDATE RUN. ACTIVE DQ11 FOUND.
1353 002424 017737 176646 001506 MOV #2,CREAM,DQCSR ;PLACE ADDRESS OF DQ11 AT DQCSR
1354 002434 062737 000002 001300 ADD #2,CREAM ;UPDATE ADDRESS POINTER
1355 002444 017737 176632 001510 MOV #2,CREAM,DQSTAT ;PLACE STATUS OF DQ11 AT DQSTAT
1356 002450 062737 000002 001300 ADD #2,CREAM ;UPDATE ADDRESS POINTER
1357 002456 013737 001506 001360 MOV DQCSR,DQCSR ;GENERATE ADDRESS OF RECEIVER INTERRUPT SERVICE PS
1358 002464 013737 001510 001350 MOV DQSTAT,DQVEC ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT VECTOR
1359 002472 042737 177007 001350 BIC #177007,DQVEC ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT SERVICE PS
1360 002500 013737 001350 001352 MOV DQVEC,DQRLVL ;GENERATE ADDRESS OF RECEIVER INTERRUPT SERVICE PS
1361 002506 062737 000002 001352 ADD #2,DQRLVL ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT VECTOR
1362 002514 013737 001352 001354 MOV DQRLVL,DQTVL ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT SERVICE PS
1363 002522 062737 000002 001354 ADD #2,DQTVL ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT SERVICE PS
1364 002530 013737 001354 001356 MOV DQTVL,DQTLVL ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT SERVICE PS
1365 002536 062737 000002 001356 ADD #2,DQTLVL ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT SERVICE PS
1366 002544 013737 001360 001362 MOV DQCSR,DQCRSH ;GENERATE ADDRESS OF HIGH BYTE
1367 002552 005237 001362 001364 INC DQCRSH ;GENERATE ADDRESS OF TRANSMITTER CONTROL REGISTER
1368 002558 013737 001360 001364 MOV DQCSR,DQTCR ;GENERATE ADDRESS OF TRANSMITTER CONTROL REGISTER
1369 002564 062737 000002 001364 ADD #2,DQTCR ;GENERATE ADDRESS OF TRANSMITTER CONTROL REGISTER
1370 002572 013737 001364 001366 MOV DQTCR,DQERR ;GENERATE ADDRESS OF ERROR REGISTER
1371 002600 062737 000002 001356 ADD #2,DQERR ;GENERATE ADDRESS OF ERROR REGISTER
1372 002606 013737 001356 001370 MOV DQERR,DQREG ;GENERATE ADDRESS OF HIGH BYTE OF ERROR REGISTER
1373 002614 005237 001370 001372 INC DQREG ;GENERATE ADDRESS OF HIGH BYTE OF ERROR REGISTER
1374 002620 013737 001370 001372 MOV DQREG,DQSEC ;GENERATE ADDRESS OF SECONDARY REGISTER
1375 002626 005737 001372 001372 INC DQSEC ;GENERATE ADDRESS OF SECONDARY REGISTER
1376 002632 013737 001372 001374 MOV DQSEC,DQSECH ;GENERATE ADDRESS OF HIGH BYTE
1377 002640 005237 001374 001374 INC DQSECH ;GENERATE ADDRESS OF HIGH BYTE
1378 : ;
1379 : ;TEST TO SEE IF TRANSMITTER ACTIVE
1380 : ;CAN SET.
1381 : ;AND IF IT DOES SET CHECK TO
1382 : ;SEE IF IT CAN BE CLEARED BY
1383 : ;MASTER CLEAR.
1384 : ;
1385 : ;TEST 2
1386 : *****
1387 002644 012737 000002 001226 TST2: MOV #2,TSTNO
1388 002652 012737 003002 001216 MOV #30,SYN1,NEXT
1389 002660 112777 000002 176502 MOVB #2,DQREG ;SEL TX BA PRI

```

```

1390 002666 012777 014066 176475 MOV #TMPBUF, DQSEC :LOAD TX BA
1391 002674 105277 176470 INCB DQREG :SEL TTX CC PRI
1392 002700 012777 000200 176464 MCV #200, DQSEC :LOAD WITH 200
1393 002706 112777 003012 176454 MOVB #MISC, DQREG :SEL MISC REGISTER
1394 002714 012777 004012 176450 MOV #4012, DQSEC :SELECT 8 BITS TEST LOOP AUTO STEP
1395 002722 005277 176436 INC DQOTCSR :SET TX GO
1396 002726 005277 176440 INC DQSEC :PRIM THE
1397 002732 005377 176434 DEC DQSEC : TRANSMITTER
1398 002736 005277 176430 INC DQSEC :CLOCK THE TRANSMITTER
1399 002742 032777 040000 176422 BIT #BIT14, DQSEC :CHECK TX ACTIVE.
1400 002750 001001 BNE .+4 :BRANCH IF ACTIVE SET
1401 002752 104024 HLT 24 :ERROR TX ACTIVE NOT SET!!
1402 002754 104412 MSTCLR :ISSUE
1403 002756 104412 MSTCLR :TWO MASTER CLEARS
1404 002760 112777 000012 176402 MOVB #MISC, DQREG :RESELECT THE MISC REGISTER
1405 002766 032777 040000 176376 BIT #BIT14, DQSEC :DID TX ACTIVE CLEAR BY MST CLR
1406 002774 001401 BEQ .+4 :BRANCH IF ACTIVE CLEAR
1407 002776 104001 HLT 1 :ERROR TX ACTIVE NOT CLEARED BY MST CLR
1408 003000 104400 SCOPE :SCOPE TEST

:ROUTINE TO SET
:TRANSMITTER POINTER
:CORRECTLY DEPENDING
:UPON THE NUMBER OF SYNC
:CHARACTERS.

1410 003002 032737 100000 001510 CKSYN1: BIT #SYNBIT, DQSTAT :CHECK TO FIND OUT IF ONE SYNC OR TWO
1411 003010 001003 BNE 15 :BRANCH IF TWO SYNC CHARS REQUIRED
1412 003012 105037 014522 CLRB SYNC :CLEAR THE FIRST SYNC CHAR
1413 003016 000403 BR 25 :BR TO LEAVE ROUTINE
1414 003020 112737 000026 014522 15: MOVB #26, SYNC :RESET SYNC CHAR TO 26
1415 003026 030240 25: NOP :FALL IN TO NEXT TEST

:TEST TO TRANSMITT ONE CHARACTER.
:TESTING TO MAKE SURE THAT THE
:CHARACTER COUNT INCREMENTS BY ONE.
:TESTING THAT THE CURRENT ADDRESS
:INCREMENTS BY ONE
:ALSO MAKING SURE THE PRI-SEC BIT SETS.

: TEST 3
*****
1433 003030 012737 000003 001226 TST3: MOV #3, TSTNO
1434 003036 012737 003054 001214 MOV #A15, RETURN
1435 003044 012737 003370 001216 MOV #TST4, NEXT
1436 003052 104413 MEMCLR :CLEAR ALL THE DQ11
1437 003054 104412 MSTCLR
1438 003056 112777 000002 176304 A13: MOVB #2, DQREG :SELECT TX CURRENT ACC.
1439 003064 012777 014524 176300 MCV #TXBUFF, DQSEC :SET THE TX CURRENT ACC.
1440 003072 105277 176272 INCB DQREG :SELECT THE TX CHAR CNT.
1441 003076 012777 177777 176266 MOV #-1, DQSEC :SET TX CHAR CNT FOR 1 CHARACTER.
1442 003104 112777 000012 176256 MOVB #MISC, DQREG :SELECT THE MISC REGISTER.
1443 003112 012777 004010 176252 MOV #4010, DQSEC :SET FOR EIGHT BITS. AND TEST LOOP
1444 003120 005037 014060 CLR DELAY :CLEAR THE DELAY
1445 003124 005277 176234 INC DQOTCSR :SET THE GO BIT AND GO!!

```

E03

```

1446 003130 105777 176230 15: TSTB 3DQTCR ;PRIMARY DONE??
1447 003134 100405 BMI 25 ;BRANCH IF DONE
1448 003136 062737 000001 014060 ADD #1,DELAY ;STALL FOR DONE
1449 003144 001371 BNE 15 ;TO SET.
1450 003146 104002 HLT 2 ;TX PRI DONE FAILED TO SET.
1451 003150 112777 000003 176212 25: MOV8 #3,3DQREG ;SELECT TX CHAR CNT
1452 003156 005777 176210 TST 3DQSEC ;MAKE SURE IT INCREMENTED
1453 003162 001401 BEQ .+4 ;BY ONE TO ZERO.
1454 003164 104003 HLT 3 ;TX PRI CHAR CNT NOT ZERO.
1455 003166 112777 000002 176174 MOV8 #2,3DQREG ;SELECT TX CURRENT ADD..PRI
1456 003174 022777 014525 176170 35: CMP #TXBUFF+1,3DQSEC ;
1457 003202 001401 BEQ .+4 ;
1458 003204 104005 HLT 5 ;CHAR CNT NOT INC BY +1
1459 003206 032777 000004 176150 45: BIT #BIT2,3DQTCR ;DID PRI/SEC SET?
1460 003214 001001 BNE .+4 ;
1461 003216 104006 HLT 6 ;TX PRI/SEC NOT SET.
1462
1463 ;TEST THAT WITH A CHARACTER
1464 ;COUNT THAT IS EVEN THAT THE
1465 ;CURRENT ADDRESS INCREMENTS BY +2
1466 ;AND THAT THE CHAR CNT GOES TO ZERO.
1467
1468
1469
1470 003220 112777 000006 176142 SECD: MOV8 #6,3DQREG ;SELECT TX CURRENT ADD.
1471 003226 012777 014524 176136 MOV #TXBUFF,3DQSEC ;SET THE TX CURRENT ADD.
1472 003234 105277 176130 INCB 3DQREG ;SELECT THE TX CHAR CNT.
1473 003240 012777 177776 176124 MOV #2,3DQSEC ;SET TX CHAR CNT FOR TWO CHARS.
1474 003246 112777 000012 176114 MOV8 #MISC,3DQREG ;SELECT THE MISC REGISTER.
1475 003254 012777 004010 176110 MOV #4010,3DQSEC ;SET FOR EIGHT BITS AND TEST LOOP
1476 003262 005037 014560 CLR DELAY ;CLEAR THE DELAY
1477 003266 005277 176072 INC 3DQTCR ;SET THE GO BIT AND GO!!
1478 003272 032777 000100 176064 15: BIT #BIT6,3DQTCR ;SECONDARY DONE??
1479 003300 001005 BNE 25 ;BRANCH IF DONE
1480 003302 062737 000001 014060 ADD #1,DELAY ;STALL FOR DONE
1481 003310 001370 BNE 15 ;TO SET.
1482 003312 104002 HLT 2 ;TX SEC DONE FAILED TO SET.
1483 003314 112777 000007 176046 25: MOV8 #7,3DQREG ;SELECT TX CHAR CNT
1484 003322 005777 176044 TST 3DQSEC ;MAKE SURE IT INCREMENTED
1485 003326 001401 BEQ .+4 ;BY ONE TO ZERO.
1486 003330 104003 HLT 3 ;TX SEC CHAR CNT NOT ZERO.
1487 003332 112777 000006 176030 MOV8 #6,3DQREG ;SELECT TX CURRENT ADD..PRI
1488 003340 022777 014526 176024 35: CMP #TXBUFF+2,3DQSEC ;
1489 003346 001401 BEQ .+4 ;
1490 003350 104004 HLT 4 ;CHAR CNT NOT INC BY +2
1491 003352 032777 000004 176004 45: BIT #BIT2,3DQTCR ;DID PRI/SEC SET?
1492 003360 001401 BEQ .+4 ;
1493 003362 104006 HLT 6 ;TX PRI/SEC NOT SET.
1494 003364 104413 MEMCLR ;
1495 003366 104400 SCOPE ;
1496
1497 ;TRANSMITTER CHARACTER LENGTH TESTS.
1498
1499 ;TEST TO TRANSMIT A CHARACTER
1500 ; 2 BITS LONG MAKING SURE THAT
1501 ;THE CHARACTER IS ALL ZERO'S

```

F03

1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557

:AND THAT THE TX LINE GOES BACK TO  
:A MARK STATE WHEN DONE.  
:

: TEST 4

:\*\*\*\*\*

003370 012737 000004 001226  
003376 012737 003416 001216  
003404 004537 010720  
003410 000002  
003412 007000  
003414 104400

TST4: MOV #5,TSTNO  
MOV #TST5,NEXT  
JSR R5, TXSTRB ;JSR TO ROUTINE  
2 ;NUMBER OF TIMES CHAR IS TO BE SHIFTED  
7000 ;BIT SELECTION TO BE PLACED INTO MISC REG  
SCOPE ;SCOPE TEST

:TEST TO TRANSMIT A CHARACTER  
: 3 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 5

:\*\*\*\*\*

003416 012737 000005 001226  
003424 012737 003444 001216  
003432 004537 010720  
003436 000003  
003440 006400  
003442 104400

TST5: MOV #5,TSTNO  
MC #TST6,NEXT  
JSR R5, TXSTRB ;JSR TO ROUTINE  
3 ;NUMBER OF TIMES CHAR IS TO BE SHIFTED  
6400 ;BIT SELECTION TO BE PLACED INTO MISC REG  
SCOPE ;SCOPE TEST

:TEST TO TRANSMIT A CHARACTER  
: 4 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 6

:\*\*\*\*\*

003444 012737 000006 001226  
003452 012737 003472 001216  
003460 004537 010720  
003464 000004  
003466 006000  
003470 104400

TST6: MOV #6,TSTNO  
MOV #TST7,NEXT  
JSR R5, TXSTRB ;JSR TO ROUTINE  
4 ;NUMBER OF TIMES CHAR IS TO BE SHIFTED  
6000 ;BIT SELECTION TO BE PLACED INTO MISC REG  
SCOPE ;SCOPE TEST

:TEST TO TRANSMIT A CHARACTER  
: 5 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 7

:\*\*\*\*\*

003472 012737 000007 001226  
003500 012737 003520 001216

TST7: MOV #7,TSTNO  
MOV #TST10,NEXT



G03

```

1558 003506 004537 010720 JSR R5, TXSTRB ;JSR TO ROUTINE
1559 003512 000005 5 ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1560 003514 005400 5400 ;BIT SELECTION TO BE PLACED INTO MISC REG
1561 003516 104400 SCOPE ;SCOPE TEST
1562
1563 ;TEST TO TRANSMITT A CHARACTER
1564 ; 6 BITS LONG MAKING SURE THAT
1565 ; THE CHARACTER IS ALL ZERO'S
1566 ; AND THAT THE TX LINE GOES BACK TO
1567 ; A MARK STATE WHEN DONE.
1568
1569
1570
1571 ; TEST 10
1572 003520 012737 000010 001226 *****
1573 003526 012737 003546 001216 †TST10: MOV #10, TSTNO
1574 003534 004537 010720 MOV #TST11, NEXT
1575 003540 000006 JSR R5, TXSTRB ;JSR TO ROUTINE
1576 003542 005000 5 ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1577 003544 104400 SCOPE ;BIT SELECTION TO BE PLACED INTO MISC REG
1578 ;SCOPE TEST
1579
1580 ;TEST TO TRANSMITT A CHARACTER
1581 ; 7 BITS LONG MAKING SURE THAT
1582 ; THE CHARACTER IS ALL ZERO'S
1583 ; AND THAT THE TX LINE GOES BACK TO
1584 ; A MARK STATE WHEN DONE.
1585
1586
1587
1588 ; TEST 11
1589 003546 012737 000011 001226 *****
1590 003554 012737 003574 001216 †TST11: MOV #11, TSTNO
1591 003562 004537 010720 MOV #TST12, NEXT
1592 003566 000007 JSR R5, TXSTRB ;JSR TO ROUTINE
1593 003572 104400 7 ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1594 ;BIT SELECTION TO BE PLACED INTO MISC REG
1595 ;SCOPE TEST
1596
1597 ;TEST TO TRANSMITT A CHARACTER
1598 ; 8 BITS LONG MAKING SURE THAT
1599 ; THE CHARACTER IS ALL ZERO'S
1600 ; AND THAT THE TX LINE GOES BACK TO
1601 ; A MARK STATE WHEN DONE.
1602
1603
1604 ; TEST 12
1605 003574 012737 000012 001226 *****
1606 003602 012737 003622 001216 †TST12: MOV #12, TSTNO
1607 003610 004537 010720 MOV #TST13, NEXT
1608 003614 000010 JSR R5, TXSTRB ;JSR TO ROUTINE
1609 003616 004000 8 ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1610 003620 104400 4000 ;BIT SELECTION TO BE PLACED INTO MISC REG
1611 ;SCOPE TEST
1612
1613 ;TEST OF CHARACTER LENGTH
;FOR CHARACTERS OVER 8 BITS LONG.

```

# H03

DZDQD MACY11 27(732) 24-MAY-76 13:10 PAGE 34  
DZDQDC.P11 DQ11 TRANSMITTER AND RECEIVER EXERCISER.

```
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625 003622 012737 000013 001226  
1626 003630 012737 003650 001216  
1627 003636 004537 010720  
1628 003642 000011  
1629 003644 003400  
1630 003646 104400  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642 003650 012737 000014 001226  
1643 003656 012737 003676 001216  
1644 003664 004537 010720  
1645 003670 000012  
1646 003672 003000  
1647 003674 104400  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659 003676 012737 000015 001226  
1660 003704 012737 003724 001216  
1661 003712 004537 010720  
1662 003716 000013  
1663 003720 002400  
1664 003722 104400  
1665  
1666  
1667  
1668  
1669
```

```
;  
: 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 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 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 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 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 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  
: 12 BITS LONG MAKING SURE THAT  
: THE CHARACTER IS ALL ZERO'S
```

```

1670                                     ;AND THAT THE TX LINE GOES BACK TO
1671                                     ;A MARK STATE WHEN DONE.
1672                                     ;
1673
1674                                     ; TEST 16
1675                                     ;*****
1676 003724 012737 000016 001226 TST16: MOV #16,TSTNO
1677 003732 012737 003752 001216      MOV #TST17,NEXT
1678 003740 004537 010720      JSR R5, TXSTRB ;DO JSR TO THE SUBROUTINE
1679 003744 000014      12. ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1680 003746 002000      2000 ;BIT SELECTION TO BE PLACED INTO MISC REG
1681 003750 104400      SCOPE ;SCOPE THE TEST
1682
1683
1684                                     ;
1685                                     ;TEST TO TRANSMITT A CHARACTER
1686                                     ; 13 BITS LONG MAKING SURE THAT
1687                                     ; THE CHARACTER IS ALL ZERO'S
1688                                     ;AND THAT THE TX LINE GOES BACK TO
1689                                     ;A MARK STATE WHEN DONE.
1690                                     ;
1691
1692                                     ; TEST 17
1693                                     ;*****
1694 003752 012737 000017 001226 TST17: MOV #17,TSTNO
1695 003760 012737 004000 001216      MOV #TST20,NEXT
1696 003766 004537 010720      JSR R5, TXSTRB ;DO JSR TO THE SUBROUTINE
1697 003772 000015      13. ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1698 003774 001400      1400 ;BIT SELECTION TO BE PLACED INTO MISC REG
1699 003776 104400      SCOPE ;SCOPE THE TEST
1700
1701                                     ;
1702                                     ;TEST TO TRANSMITT A CHARACTER
1703                                     ; 14 BITS LONG MAKING SURE THAT
1704                                     ; THE CHARACTER IS ALL ZERO'S
1705                                     ;AND THAT THE TX LINE GOES BACK TO
1706                                     ;A MARK STATE WHEN DONE.
1707                                     ;
1708
1709                                     ; TEST 20
1710                                     ;*****
1711 004000 012737 000020 001226 TST20: MOV #20,TSTNO
1712 004006 012737 004026 001216      MOV #TST21,NEXT
1713 004014 004537 010720      JSR R5, TXSTRB ;DO JSR TO THE SUBROUTINE
1714 004020 000016      14. ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1715 004022 001000      1000 ;BIT SELECTION TO BE PLACED INTO MISC REG
1716 004024 104400      SCOPE ;SCOPE THE TEST
1717
1718                                     ;
1719                                     ;TEST TO TRANSMITT A CHARACTER
1720                                     ; 15 BITS LONG MAKING SURE THAT
1721                                     ; THE CHARACTER IS ALL ZERO'S
1722                                     ;AND THAT THE TX LINE GOES BACK TO
1723                                     ;A MARK STATE WHEN DONE.
1724                                     ;
1725                                     ; TEST 21

```

# J03

DZDQD MACY11 27(732) 24-MAY-76 13:10 PAGE 36  
 DZDQDC.P11 DQ11 TRANSMITTER AND RECEIVER EXERCISER.

```

1726          :*****
1727 004026 012737 000021 001226 †ST21: MOV #21,TSTNO
1728 004034 012737 004054 001216      MOV #TST22,NEXT
1729 004042 004537 010720      JSR RS,TXSTRB ;DO JSR TO THE SUBROUTINE
1730 004046 000017      15. ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1731 004050 000400      400 ;BIT SELECTION TO BE PLACED INTO MISC REG
1732 004052 104400      SCOPE ;SCOPE THE TEST
1733
1734          ;
1735          ;TEST TO TRANSMITT A CHARACTER
1736          ; 16 BITS LONG MAKING SURE THAT
1737          ; THE CHARACTER IS ALL ZERO'S
1738          ; AND THAT THE TX LINE GOES BACK TO
1739          ; A MARK STATE WHEN DONE.
1740          ;
1741          ;
1742          ; TEST 22
1743          :*****
1744 004054 012737 000022 001226 †ST22: MOV #22,TSTNO
1745 004062 012737 004102 001216      MOV #TST23,NEXT
1746 004070 004537 010720      JSR RS,TXSTRB ;DO JSR TO THE SUBROUTINE
1747 004074 000020      16. ;NUMBER OF TIMES CHAR IS TO BE SHIFTED
1748 004076 000000      0 ;BIT SELECTION TO BE PLACED INTO MISC REG
1749 004100 104400      SCOPE ;SCOPE THE TEST
1750
1751          ;
1752          ;
1753          ;
1754          ;
1755          ;TEST OF TRANSMITTER IDLE SYNC
1756          ;TEST THAT THE TRANSMITTER CAN
1757          ;REALLY IDLE SYNC CHARACTERS
1758
1759          ;
1760          ; TEST 23
1761          :*****
1762 004102 012737 000023 001226 †ST23: MOV #23,TSTNO
1763 004110 012737 004436 001216      MOV #TST24,NEXT
1764 004116 005077 175242      CLR @DQTCR ;CLR TX STATUS
1765 004122 032777 000002 175234      BIT #BIT1,@DQTCR ;IDLE SET?
1766 004130 001401      BEQ .+4
1767 004132 104000      HLT ;IDLE SHOULD NOT BE SET!
1768 004134 052777 000002 175222      BIS #BIT1,@DQTCR ;SET IDLE BIT
1769 004142 032777 000002 175214      BIT #BIT1,@DQTCR ;IS IDLE SET?
1770 004150 001001      BNE .+4 ;BR IF SET.
1771 004152 104000      HLT ;IDLE BIT SHOULD BE SET!
1772 004154 042777 000002 175202      BIC #BIT1,@DQTCR ;CLEAR IDLE BIT.
1773 004162 032777 000002 175174      BIT #BIT1,@DQTCR ;IS IDLE BIT SET?
1774 004170 001401      BEQ .+4 ;BR IF CLEAR.
1775 004172 104000      HLT ;IDLE BIT NOT CLEARED.
1776 004174 052777 000002 175162      BIS #BIT1,@DQTCR ;SET IDLE
1777 004202 104412      MSTCLR
1778 004204 032777 000002 175152      BIT #BIT1,@DQTCR ;IS IDLE SET?
1779 004212 001401      BEQ .+4
1780 004214 104000      HLT ;IDLE BIT NOT CLEARED BY INIT!
1781 004216 012737 000005 001250      MOV #5,TEMP3
  
```

```

1782 004224 012737 000377 014056 1$: MOV #377,WORD
1783 004232 112777 000011 175130 MOVB #11,ADQREG
1784 004240 013777 014520 175124 MOV .SYNC,ADQSEC
1785 004246 012737 009010 014062 MOV #10,COUNT ;PICK UP THE NUMBER OF SHIFTS
1786 004254 012737 004000 014064 MOV #4000,BITSEL ;PICK UP NUMBER OF BIT PER CHAR.
1787 004262 112777 000002 175100 MOVB #2,ADQREG ;SELECT THE TRANSMITTER BA PRI.
1788 004270 012777 014056 175074 MOV #WORD,ADQSEC ;LOAD THE BA
1789 004276 105277 175066 INCB ADQREG ;SELECT THE TRANSMITTER CC PRI.
1790 004302 012777 177777 175062 MOV #-1,ADQSEC ;LOAD THE CC WITH -1
1791 004310 112777 000012 175052 MOVB #MISC.,ADQREG ;SELECT THE MISC REGISTER.
1792 004316 053777 014064 175046 BIS BITSEL,ADQSEC ;LOAD MISC REG WITH NUMBER OF BITS PER CHAR.
1793 004324 052777 000012 175040 BIS #12,ADQSEC ;ADD TO THAT TEST LOOP AND AUTO STEP.
1794 004332 052777 000002 175024 BIS #BIT1,ADQTCR ;SET TRANSMITTER IDLE MODE.
1795 004340 005037 001252 CLR TEMP4
1796 004344 006037 001252 2$: ROR TEMP4 ;SHIFT THE STORAGE OF DATA FROM THE TRANSMITTER.
1797 004350 005277 175016 INC ADQSEC ;CLOCK THE TRANSMITTER -UP-
1798 004354 005377 175012 DEC ADQSEC ;CLOCK THE TRANSMITTER -DOWN-
1799 004360 007702 175006 MOV ADQSEC,R2 ;MOVE THE MISC REG TO R2
1800 004364 042702 177577 BIC #177577,R2 ;CLEAR ALL BUT THE BIT WINDOW.
1801 004370 050237 001252 3$: BIS R2,TEMP4 ;PLACE DATA INTO TEMPORY LOCATION
1802 004374 005337 014062 DEC COUNT ;IS CHARACTER COMPLETELY SHIFTED OUT?
1803 004400 001361 BNE 2$ ;BRANCH IF MORE BITS TO GO.
1804 004402 005137 001252 COM TEMP4 ;COMPLIMENT DATA STORAGE
1805 004406 012737 000026 001254 MOV #26,TEMP5
1806 004414 123737 001254 001252 CMPB TEMP5,TEMP4
1807 004422 001401 BEQ .+4
1808 004424 104012 HLT 12
1809 004426 005337 001250 DEC TEMP3
1810 004432 001274 BNE 1$
1811 004434 104400 SCOPE

;
; TRANSMITTER DATA REALIBILITY TEST.
; TEST TO TRANSMITT AN EIGHT
; BIT BINARY COUNT PATTERN (000-377)
;
; NOTE THIS TEST IS FOR UP TO EIGHT BITS PER CHARACTER.
; PARITY WILL BE ENABLED WHEN "PARFLG" IS NON-ZERO
;
; TEST 24
; *****
1829 004436 012737 000024 001226 †ST24: MOV #24,TSTNO
1830 004444 012737 004570 001216 MOV #TST25,NEXT
1831 004452 012737 004472 001220 MOV #2$,LOCK
1832 004460 105037 012602 CLRB PARFLG
1833 004464 005000 1$: CLR RD ;SET DATA TO ZERO
1834 004466 005037 014052 CLR EXTFLG ;TELL SUBROUTINE THIS IS FOR EIGHT BITS
1835 004472 010037 014056 2$: MOV RD,WORD ;PLACE DATA FOR WORK.
1836 004476 005037 001252 CLR TEMP4 ;CCLEAR WHERE CHAR IS TO BE STORED
1837 004502 104412 MSTCLR ;MASTER CLEAR

```

1838	004504	004537	011266		JSR	RS, TXSTRD		; GO TO ROUTINE
1839	004510	000010			8.			; NUMBER OF SHIFTS REQUIRED
1840	004512	004000			4000			; EIGHT BITS
1841	004514	105737	012602		TSTB	PARFLG		
1842	004520	001402			BEQ	.+6		
1843	004522	004737	012444		JSR	PC, GENPAR		
1844	004526	013737	014056	001254	MOV	WORD, TEMPS		; STORE GOOD CHARACTER
1845	004534	123737	001254	001252	CMPB	TEMPS, TEMP4		; COMPARE GOOD CHAR TO TX CHAR
1846	004542	001401			BEQ	.+4		; BR IF SAME
1847	004544	104012			HLT	12		; DATA COMPARISON ERROR
1848	004546	104401			SCOPI			; DOES USER WANT TO LOCK ON THIS CHAR?
1849	004550	105200			INCB	RO		; UPDATE GOOD CHARACTER
1850	004552	001347			SNE	2\$		; IF NOT ALL CHARACTERS GO DO AGAIN
1851	004554	012700	000200		MOV	#200, RO		
1852	004560	105137	012602		COMB	PARFLG		
1853	004564	001342			BNE	2\$		
1854	004566	104400			SCOPE			; SCOPE THIS TEST
1855								
1856								; TRANSMITTER DATA REALIBILITY TEST
1857								; TEST TO TRANSMITT AN EIGHT BIT
1858								; BINARY COUNT PATTERN (000400-177400)
1859								
1860								; PARITY WILL BE ENABLED WHEN "PARFLG" IS NON-ZERO
1861								; NOTE THIS IS FOR 16 BITS PER CHAR. (LOW BYTE IS=0; THE HIGH BYTE =BINARY COUNT.
1862								
1863								
1864								; TEST 25
1865	004570	012737	000025	001226	TST25:	MOV	#25, TSTNO	
1866	004576	012737	004730	001216		MOV	#TST26, NEXT	
1867	004604	012737	004626	001220		MOV	#2\$, LOCK	
1868	004612	112737	000377	014052		MOV	#377, EXTFLG	
1869	004620	105037	012602			CLRB	PARFLG	; TELL SUBROUTINE THIS IS FOR 16 BITS PER CHAR
1870	004624	005000			1\$:	CLR	RO	; NO PARITY CHECKING NOW
1871	004626	010037	014056		2\$:	MOV	RO, WORD	; ZERO DATA POINTER
1872	004632	000337	014056			SWAB	WORD	; PREPARE DATA FOR SUBROUTINE
1873	004636	005037	001252			CLR	TEMP4	; PUT DATA IN HIGH BYTE
1874	004642	104412				MSTCLR		; ZERO STORE AREA
1875	004644	004537	011266			JSR	RS, TXSTRD	; INIT DQ11
1876	004650	000020				16.		; GOTO SUBROUTINE
1877	004652	000000				0		; THIS IS NUMBER OF SHIFTS.
1878	004654	105737	012602			TSTB	PARFLG	; THIS IS BITS/PER/CHARACTER SELECT
1879	004660	001402				BEQ	.+6	; IS PARITY ENABLED?
1880	004662	004737	012444			JSR	PC, GENPAR	; BR IF NOT ENABLED
1881	004666	013737	014056	001254		MOV	WORD, TEMPS	; GO CALCULATE THE PARITY
1882	004674	023737	001254	001252		CMP	TEMPS, TEMP4	; STORE THE CHARACTER
1883	004702	001401				BEQ	.+4	; IS THE CHARACTER CORRECT
1884	004704	104012				HLT	12	; BR IF GOOD
1885	004706	104401				SCOPI		; DATA COMPARISON ERROR.
1886	004710	105200				INCB	RO	; LOCK ON DATA? (SW09=1)
1887	004712	001345				BNE	2\$	; UPDATE DATA POINTER
1888	004714	012700	000200			MOV	#200, RO	; BR IF MORE TO GO
1889	004720	105137	012602			COMB	PARFLG	; NOW ENABLE THE PARITY TEST.
1890	004724	001340				BNE	2\$	; BR IF FIRST TIME FOR PARITY
1891	004726	104400				SCOPE		; SCOPE THE TEST.

M03

```

1892
1893
1894 ; RECEIVER CHARACTER LENGTH TEST
1895 ; TEST THAT ALL CHARACTER
1896 ; LENGTHS WORK CORRECTLY.
1897
1898 ; TEST OF RX CHARACTER LENGTH 2 BITS LONG.
1899
1900 ;
1901 ; TEST 26
1902 ; *****
1903 ;
1904 ;
1905 ;
1906 ;
1907 ;
1908 ;
1909 ; TEST OF RX CHARACTER LENGTH 3 BITS LONG.
1910 ;
1911 ; TEST 27
1912 ; *****
1913 ;
1914 ;
1915 ;
1916 ;
1917 ;
1918 ;
1919 ;
1920 ; TEST OF RX CHARACTER LENGTH 4 BITS LONG.
1921 ;
1922 ; TEST 30
1923 ; *****
1924 ;
1925 ;
1926 ;
1927 ;
1928 ;
1929 ;
1930 ;
1931 ; TEST OF RX CHARACTER LENGTH 5 BITS LONG.
1932 ;
1933 ; TEST 31
1934 ; *****
1935 ;
1936 ;
1937 ;
1938 ;
1939 ;
1940 ;
1941 ;
1942 ; TEST OF RX CHARACTER LENGTH 6 BITS LONG.
1943 ;
1944 ; TEST 32
1945 ; *****
1946 ;
1947 ;

```

Line	Address	OpCode	OpCode	OpCode	OpCode	Comment
1902	004730	012737	000026	001226	TST26:	MOV #26,TSTNO
1903	004736	012737	004756	001216		MOV #TST27,NEXT
1904	004744	004537	012132			JSR R5,RXLNG ;GOTO JSR SUBROUTINE
1905	004750	007000				7000 ;CHARACTER EXPECTED TO FIND
1906	004752	000002				2 ;BITS/PER/CHAR TO BE PLACED INTO MISC REG
1907	004754	104400				SCOPE ;SCOPE THIS TEST
1913	004756	012737	000027	001226	TST27:	MOV #27,TSTNO
1914	004764	012737	005004	001216		MOV #TST30,NEXT
1915	004772	004537	012132			JSR R5,RXLNG ;GOTO JSR SUBROUTINE
1916	004776	006400				6400 ;CHARACTER EXPECTED TO FIND
1917	005000	000004				4 ;BITS/PER/CHAR TO BE PLACED INTO MISC REG
1918	005002	104400				SCOPE ;SCOPE THIS TEST
1924	005004	012737	000030	001226	TST30:	MOV #30,TSTNO
1925	005012	012737	005032	001216		MOV #TST31,NEXT
1926	005020	004537	012132			JSR R5,RXLNG ;GOTO JSR SUBROUTINE
1927	005024	006000				6000 ;CHARACTER EXPECTED TO FIND
1928	005026	000010				10 ;BITS/PER/CHAR TO BE PLACED INTO MISC REG
1929	005030	104400				SCOPE ;SCOPE THIS TEST
1935	005032	012737	000031	001226	TST31:	MOV #31,TSTNO
1936	005040	012737	005060	001216		MOV #TST32,NEXT
1937	005046	004537	012132			JSR R5,RXLNG ;GOTO JSR SUBROUTINE
1938	005052	005400				5400 ;CHARACTER EXPECTED TO FIND
1939	005054	000020				20 ;BITS/PER/CHAR TO BE PLACED INTO MISC REG
1940	005056	104400				SCOPE ;SCOPE THIS TEST
1946	005060	012737	000032	001226	TST32:	MOV #32,TSTNO
1947	005066	012737	005106	001216		MOV #TST33,NEXT

```

1948 005074 004537 012132 JSR R5,RXLNG ;GOTO JSR SUBROUTINE
1949 005100 005000 5000 ;CHARACTER EXPECTED TO FIND
1950 005102 000040 40 ;BITS/PER/CHAR TO BE PLACED INTO MISC REG
1951 005104 104400 SCOPE ;SCOPE THIS TEST
1952
1953 ;TEST OF RX CHARACTER LENGTH 7 BITS LONG.
1954
1955 ; TEST 33
1956 ;*****
1957 005106 012737 000033 001226 TST33: MOV #33,TSTNO
1958 005114 012737 005134 001216 MOV #TST34,NEXT
1959 005122 004537 012132 JSR R5,RXLNG ;GOTO JSR SUBROUTINE
1960 005126 004400 4400 ;CHARACTER EXPECTED TO FIND
1961 005130 000100 100 ;BITS/PER/CHAR TO BE PLACED INTO MISC REG
1962 005132 104400 SCOPE ;SCOPE THIS TEST
1963
1964 ;TEST OF RX CHARACTER LENGTH 8 BITS LONG.
1965
1966 ; TEST 34
1967 ;*****
1968 005134 012737 000034 001226 TST34: MOV #34,TSTNO
1969 005142 012737 005162 001216 MOV #TST35,NEXT
1970 005150 004537 012132 JSR R5,RXLNG ;GOTO JSR SUBROUTINE
1971 005154 004000 4000 ;CHARACTER EXPECTED TO FIND
1972 005156 000200 200 ;BITS/PER/CHAR TO BE PLACED INTO MISC REG
1973 005160 104400 SCOPE ;SCOPE THIS TEST
1974
1975 ;RECEIVER CHARACTER LENGTH TEST
1976 ;FOR CHARACTERS OVER EIGHT BITS LONG.
1977
1978
1979
1980 ;TEST OF CHARACTER LENGTH 9 BITS LONG.
1981
1982 ; TEST 35
1983 ;*****
1984 005162 012737 000035 001226 TST35: MOV #35,TSTNO
1985 005170 012737 005210 001216 MOV #TST36,NEXT
1986 005176 004537 012302 JSR R5,RXLNG ;GOTO SUBROUTINE
1987 005202 003400 3400 ;CHARACTER EXPECTED TO BE FOUND
1988 005204 000400 400 ;BITS/PER/CHAR TO BE PLACED INTO MISC REG
1989 005206 104400 SCOPE ;SCOPE THIS TEST
1990
1991 ;TEST OF CHARACTER LENGTH 10 BITS LONG.
1992
1993 ; TEST 36
1994 ;*****
1995 005210 012737 000036 001226 TST36: MOV #36,TSTNO
1996 005216 012737 005236 001216 MOV #TST37,NEXT
1997 005224 004537 012302 JSR R5,RXLNG ;GOTO SUBROUTINE
1998 005230 003000 3000 ;CHARACTER EXPECTED TO BE FOUND
1999 005232 001000 1000 ;BITS/PER/CHAR TO BE PLACED INTO MISC REG
2000 005234 104400 SCOPE ;SCOPE THIS TEST
2001
2002 ;TEST OF CHARACTER LENGTH 11 BITS LONG.
2003
  
```





0005414	012737	000044	001226
0005414	012737	005442	001216
0005430	004537	012302	
0005434	000000		
0005436	100000		
0005440	104400		

```

*****
TST44:  MOV      #44,TSTNO
        MOV      #TST45,NEXT
        JSR      RS,RXELNG      :GOTO SUBROUTINE
        O        :CHARACTER EXPECTED TO BE FOUND
        100000 :BITS/PER/CHAR TO BE PLACED INTO MISC REG
        SCOPE   :SCOPE THIS TEST

```

```

:TEST THAT SYNC1 AND SYNC2
:SET WHEN RECEIVER ACTIVE SET
:AND IF THEY DO THE TEST THAT THEY
:CLEAR BY MASTER CLEAR.

```

: TEST 45

0005442	012737	000045	001226
0005442	012737	005546	001216
0005446	112777	000012	173704
0005446	012777	000012	173700
0005446	052777	010000	173660
0005446	017700	173666	
0005446	042700	147777	
0005446	022700	030000	
0005446	001401		
0005446	104016		
0005446	052777	000040	173644
0005446	112777	000012	173634
0005446	005777	173632	
0005446	001401		
0005446	104016		
0005446	104400		

```

*****
TST45:  MOVL     #45,TSTNO
        MOV     #TST46,NEXT
        MOV8    #MISC.,DQREG   :SELECT THE MISC REGISTER
        MOV     #12,DQSEC      :SET TEST LOOP AND AUTO/STEP
        BIS     #BIT12,DQRCR   :SET RX ACTIVE
        MOV     DQSEC,RO       :READ THE DQSEC
        BIC     #147777,RO     :CLEAR ALL BUT SYNC 1 AND SYNC 2
        CMP     #30000,RO      :DID BOTH OF THEM SET?
        BEQ     .+4            :BR IF GOOD
        HLT     16             :SYNC 1 AND SYNC 2 NOT SET.
        BIS     #BITS,DQSEC    :SET MASTER CLEAR
        MOV8    #MISC.,DQREG   :RESELECT THE MISC REGISTER
        TST     DQSEC          :IS THE DQSEC =0
        BEQ     .+4            :BR IF YES
        HLT     17             :DQSEC NOT=0
        SCOPE   :SCOPE THIS TEST.

```

```

:SYNC TESTS.
:TEST THAT RECEIVER ACTIVE AND SYNC 1 AND SYNC 2
:ASSERT AT THE PROPER TIME.
:TEST INVOLVES BOTH SYNCING AN AN EIGHT BIT CHAR
:AND A SIXTEEN BIT CHAR.

```

```

:LOOK AT LOCATION "WORD"
:IF "WORD IS EQUAL TO 377 THE THE EIGHT
:BIT PER CHAR IS BEING EXECUTED.
:IF "WORD" IS EQUAL TO 177777 THEN THE SIXTEEN
:BIT PER CHAR IS BEING EXECUTED.

```

: TEST 46

0005446	012737	000046	001226
0005446	012737	005576	001216
0005446	004537	005626	
0005446	000377		
0005446	000010		
0005446	004000		
0005446	104400		

```

*****
TST46:  MOV      #46,TSTNO
        MOV      #TST47,NEXT
        JSR      RS,SYNST      :GOTO THE ACTUAL TEST.
        377      :DATA CHAR FOR EIGHT BITS PER CHAR.
        8        :SHIFTS PER CHAR. NEEDED FOR TEST
        4000    :BITS PER CHAR SELECTION FOR DQSEC.
        SCOPE

```

: ABOVE TEST FOR EIGHT BITS PER CHAR.  
: BELOW TEST FOR SIXTEEN BITS PER CHAR.

: TEST 47  
: \*\*\*\*\*

005576 012737 000047 001226  
005604 012737 006236 001216  
005612 004537 005626  
005616 177777  
005620 000020  
005622 000000  
005624 104400

ST47: MOV #47, TSTNO ; GOTO THE ACTUAL TEST  
MOV #TST50, NEXT ; DATA FOR 16 BITS PER CHAR.  
JSR RS, SYNTST ; SHIFTS PER CHAR.  
177777 ; SELECTION FOR DQSEC BITS PER CHAR.  
16 ; SCOPE THIS TEST  
0000  
SCOPE

: TEST THAT SYNC 1 AND SYNC 2  
: SET WHEN DATA IS RECEIVED  
: THIS TEST WILL CHECK FOR EITHER  
: 1 OR 2 SYNC CHARACTERS.

SYNTST:

005626 012537 014056  
005626 011537 005722  
005632 011537 006044  
005636 011537 000002 006044  
005642 162737  
005650 012537 006166  
005654 005337 006166  
005660 011537 005724  
005664 011537 006046  
005670 012537 006170  
005674 010537 006234  
005700 104412  
005702 112777 000011 173450  
005710 012777 177777 173454  
005716 004537 011522  
005722 000001  
005724 000001  
005726 112777 000012 173454  
005734 032777 020000 173430  
005742 001401  
005744 104000  
005746 032777 010000 173404  
005754 001401  
005756 104000  
005760 005277 173406  
005764 005377 173402  
005770 032737 100000 001510  
005776 001000  
006000 005337 006166  
006004 000442  
006006 017700 173350  
006012 042700 147777  
006016 022700 020000  
006022 001401  
006024 104000

MOV (RS)+, WORD ; GET DATA CHARACTER  
MOV (RS), 4\$ ; GET NUMBER OF SHIFTS.  
MOV (RS), 6\$  
SUB #2, 6\$ ; ADJUST SHIFTS.  
MOV (RS)+, 8\$ ; GET THE SHIFTS  
DEC 9\$ ; ADJUST THE SHIFTS.  
MOV (RS), 5\$ ; GET THE BITS PER CHAR.  
MOV (RS), 7\$  
MOV (RS)+, 9\$  
MOV RS, 10\$ ; SAVE THE PC TO RETURN  
MSTCLR ; INIT THE DQ11  
MOV #11, DQREG ; SEL THE SYNC REG  
MOV #-1, DQSEC ; SET SYNC CHAR TO ALL 1'S  
JSR RS, RXSTRA ; GOTO THE SUBROUTINE  
4\$: .BLKW 1 ; NUMBER OF SHIFTS  
5\$: .BLKW 1 ; MISC FUNCTION  
MOV #MISC, DQREG ; SELECT THE MISC REGISTER  
BIT #BIT13, DQSEC ; IS SYNC 1 UP YET  
BEQ .+4 ; BR IF NO  
HLT ; SYNC 1 UP TOO SOON  
BIT #BIT12, DQRCR ; ACTIVE UP??  
BEQ .+4 ; BR IF ACTIVE NOT UP  
HLT ; ACTIVE UP TOO SOON.  
INC DQSEC ; CLOCK UP  
DEC DQSEC ; CLOCK DN  
BIT #SYNBIT, DQSTAT ; NUMBER OF SYNC CHARS=  
BNE .+10 ; BR IF TWO SYNC CHAR.  
DEC 6\$ ; ADJUST COUNT WHEN ONE SYNC SELECTED.  
BR 1\$ ; BR TO TEST ONE SYNC CHAR.  
MOV DQSEC, R0 ; READ DQSEC  
BIC #147777, R0 ; CLEAR GARBAGE  
CMP #20000, R0 ; IS SYNC 1 UP?  
BEQ .+4 ; BR IF YES  
HLT ; SYNC ONE NOT SET OR SYNC 2 IS SET

```

2172 006026 032777 010000 173324 BIT #BIT12,0DQRCR ;ACTIVE UP?
2173 006034 001401 BEQ .+4 ;BR IF ACTIVE =0
2174 006036 104000 HLT ;ACTIVE UP TOO SOON
2175 006040 004537 011522 JSR R5,RXSTRA ;GOTO THE SUBROUTINE
2176 006044 000001 55: .BLKW 1 ;NUMBER OF SHIFTS MINUS 2
2177 006046 000001 75: .BLKW 1 ;MISC FUNCTION (PERS PER CHAR.).
2178 006050 017700 173316 MOV 0DQSEC,R0 ;READ THE DQSEC
2179 006054 042700 147777 BIC #147777,R0 ;CLEAR ALL BUT SYNC 1 AND SYNC 2
2180 006060 022700 020000 CMP #20000,R0 ;ARE BOTH SYNC 1 *AND* SYNC 2 SET?
2181 006064 001401 BEQ .+4 ;BR IF YES
2182 006066 104000 HLT ;EITHER OR BOTH SYNC 1 OR SYNC 2 NOT SET.
2183 006070 032777 010000 173252 BIT #BIT12,0DQRCR ;ACTIVE UP?
2184 006076 001401 BEQ .+4 ;BR IF ACTIVE NOT SET.
2185 006100 104000 HLT ;ACTIVE UP TOO SOON
2186 006102 005277 173254 INC 0DQSEC ;CLOCK UP.
2187 006106 005377 173250 DEC 0DQSEC ;CLOCK DN
2188 006112 017700 173254 15: MOV 0DQSEC,R0 ;READ AND SAVE DQSEC
2189 006116 042700 147777 BIC #147777,R0 ;CLEAR ALL BUT SYNC 1 AND SYNC 2
2190 006122 022700 030000 CMP #30000,R0 ;ARE BOTH SYNC 1 AND SYNC 2 SET?
2191 006126 001401 BEQ .+4 ;BR IF YES
2192 006130 104000 HLT ;EITHER OR BOTH SYNC 1 OR SYNC 2 NOT SET.
2193 006132 032737 004000 001510 BIT #ACTBIT,DQSTAT ;WHEN DO YOU GO ACTIVE??
2194 006140 001006 BNE 25 ;BR IF ACTIVE ON FIRST NON-SYNC.
2195 006142 032777 010000 173210 BIT #BIT12,0DQRCR ;IS ACTIVE UP?
2196 006150 001001 BNE .+4 ;*** NOW ACTIVE SHOULD BE SET***
2197 006152 104000 HLT ;NOW ACTIVE SHOULD BE UP..
2198 006154 000424 BR 35 ;ALL DONE GO HOME
2199 006156 005037 014056 25: CLR WORD ;SET DATA TO NON-SYNC
2200 006162 004537 011522 JSR R5,RXSTRA ;PUSH IT INTO THE RECEIVER
2201 006166 000001 55: .BLKW 1 ;NUMBER OF SHIFTS MINUS 1
2202 006170 000001 55: .BLKW 1 ;MISC FUNCTION.
2203 006172 032777 010000 173160 BIT #BIT12,0DQRCR ;ACTIVE UP
2204 006200 001401 BEQ .+4 ;ONE MORE SHIFT BEFORE ACTIVE=1
2205 006202 104000 HLT ;ACTIVE IS UP TOO SOON
2206 006204 005277 173162 INC 0DQSEC ;FINAL CLOCK UP
2207 006210 005377 173156 DEC 0DQSEC ;CLOCK DN
2208 006214 032777 010000 173136 BIT #BIT12,0DQRCR ;**** NOW ACTIVE SHOULD BE SET **
2209 006222 001001 BNE .+4 ;BR IF ACTIVE =1
2210 006224 104000 HLT ;ACTIVE ON FIRST NON-SYNC NOT WORKING.
2211 006226 013705 006234 35: MOV 10$,R5 ;RESTORE PC POINTER
2212 006232 000205 RTS R5 ;GOTO MAIN TEST
2213 006234 000000 105: 0 ;STORE R5 (PC) HERE.

```

```

: TEST OF RECEIVER CHARACTER COUNT AND BUSS
: ADDRESS. TEST TO MAKE SURE
: THAT THEY INCREMENT PROPERELY.

```

```

: TEST WITH CHARACTER COUNT OF -1 (ODD)

```

```

: TEST 50
:*****

```

```

2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227

```

```

22228 006236 012737 000050 001226 TST50: MOV #50,TSTNO
22229 006244 012737 006364 001216 MOV #TST51,NEXT
22230 006252 104412 MSTCLR ;INIT DQ11
22231 006254 105077 173110 CLRB @DQREG ;SEL RX BA PRI.
22232 006260 012777 014116 173104 MOV #RXBUFF,@DQSEC ;SET RX BA PRI.
22233 006266 105277 173076 INCB @DQREG ;SEL RX WC PRI.
22234 006272 012777 177777 173072 MOV #-1,@DQSEC ;ONE CHAR RECEIVE
22235 006300 112777 000012 173062 MOVB #MISC,@DQREG ;SELECT THE MISC REG.
22236 006306 012777 004010 173056 MOV #4010,@DQSEC ;SET EIGHT BITS AND TEST LOOP
22237 006314 012777 010001 173036 MOV #10001,@DQRCSR ;SET RX ACTIVE AND RX GO!!
22238 006322 105777 173032 TSTB @DQRCSR ;RX PRI DONE?
22239 006326 100375 BPL -4 ;HANG HERE TILL DONE.
22240 006330 105077 173034 CLRB @DQREG ;GET RA BA PRI.
22241 006334 022777 014117 173030 CMP #RXBUFF+1,@DQSEC ;DID BA INC RIGHT?
22242 006342 001401 BEQ +4 ;BR IF BA GOOD
22243 006344 104000 HLT ;RX BA ERROR.
22244 006346 105277 173016 INCB @DQREG ;GET RX WC PRI.
22245 006352 005777 173014 TST @DQSEC ;DID IT GOTO ZERO?
22246 006356 001401 BEQ +4 ;BR IF YES
22247 006360 104000 HLT ;RX WC PRI NOT =0
22248 006362 104400 SCOPE ;SCOPE THE TEST

```

```

;TEST OF RECEIVER CHARACTER COUNT
;AND BLSS ADDRESS
;WITH A CHARACTER COUNT OF -2 (EVEN)
;MAKING SURE THAT THE CC AND BA
;INCREMENT CORRECTLY.

```

```

: TEST 51
:*****
TST51: MOV #51,TSTNO
MOV #TST52,NEXT
MSTCLR ;ISSUE CLEAR
CLRB @DQREG ;SELECT THE RX BA PRI
MOV #RXBUFF,@DQSEC ;SET RX BA PRI.
INCB @DQREG ;SELECT RX WC PRI.
MOV #-2,@DQSEC ;SET FOR TWO CHARS
MOVB #MISC,@DQREG ;SELECT THE MISC REGISTER
MOV #4010,@DQSEC ;SET EIGHT BITS AND TEST LOOP
MOV #10001,@DQRCSR ;SET RX ACTIVE AND GO!!
TSTB @DQRCSR ;WAIT FOR RX PRI DONE.
BPL -4 ;HANG HERE TILL DONE
CLRB @DQREG ;SELECT THE RX BA PRI
CMP #RXBUFF+2,@DQSEC ;DID RX BA INCREMENT RIGHT?
BEQ +4 ;BR IF GOOD
HLT ;RX BA ERROR
INCB @DQREG ;SELECT THE RX WC PRI.
TST @DQSEC ;DID IF GOTO ZERO
BEQ +4 ;BR IF YES
HLT ;RX WC NOT =ZERO
SCOPE ;SCOPE THE TEST

```

```

22263 006364 012737 000051 001226
22264 006372 012737 006512 001216
22265 006400 104412
22266 006402 105077 172762
22267 006406 012777 014116 172756
22268 006414 105277 172750
22269 006420 012777 177776 172744
22270 006426 112777 000012 172734
22271 006434 012777 004010 172730
22272 006442 012777 010001 172710
22273 006450 105777 172704
22274 006454 100375
22275 006456 105077 172706
22276 006462 022777 014120 172702
22277 006470 001401
22278 006472 104000
22279 006474 105277 172670
22280 006500 005777 172666
22281 006504 001401
22282 006506 104000
22283 006510 104400

```

2292  
2293  
2294  
2295  
2296  
2297  
2298  
2299  
2300  
2301  
2302  
2303  
2304  
2305  
2306  
2307  
2308  
2309  
2310  
2311  
2312  
2313  
2314  
2315  
2316  
2317  
2318  
2319  
2320  
2321  
2322  
2323  
2324  
2325  
2326  
2327  
2328  
2329  
2330  
2331  
2332  
2333  
2334  
2335  
2336  
2337  
2338  
2339

006512 012737 000052 001226  
006520 012737 006660 001216  
006526 012737 006550 001220  
006534 105037 012602  
006540 112737 000377 014052  
006546 005000  
006550 104412  
006552 010037 014056  
006556 105737 012602  
006562 001402  
006564 004737 012444  
006570 004537 011522  
006574 000020  
006576 000000  
006600 013737 014116 001252  
006606 013737 014056 001254  
006614 005777 172546  
006620 100001  
006622 104000  
006624 023737 001254 001252  
006632 001401  
006634 104020  
006636 104401  
006640 005200  
006642 001342  
006644 012700 177400  
006650 105137 012602  
006654 001335  
006656 104400

: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
:*****
TST52: MOV #52,TSTNO
MOV #TST53,NEXT
MOV #1$,LOCK
CLRB PARFLG ;SET FOR NO PARITY NOW
MOVB #377,EXTFLG ;TELL SUBROUTINE 16 BIT CHAR.
CLR RO ;ZERO DATA POINTER
MSTCLR ;ISSUE CLEAR DQ11
MOV RO,WORD ;LOAD DATA FOR SUB ROUTINE
TSTB PARFLG ;IS PARITY ENABLED?
BEQ .+5 ;BR IF NO
JSR PC,GENPAR ;GO AND FIGURE PARITY.
JSR RS,RXSTRA ;GO PUSH CHARACTER INTO RECEIVER.
16. ;NUMBER OF SHIFTS NEEDED
0000 ;BITS PER/CHAR FOR MISC REG
MOV RXBUFF,TEMP4 ;GET EXPECTED
MOV WORD,TEMP5 ;GET EXPECTED
2$: 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 GOOD DATA
HLT 20 ;RECEIVER DATA COMPARISON ERROR.
SCOPI ;LOCK ON SLECTED DATA (SW-3=1)
INC RO ;UPDATE DATA POINTER.
BNE 1$ ;BR IF MORE CHARS TO GO.
MOV #177400,RO ;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
:*****
```

```

2340 006660 012737 000053 001226 TST53: MOV #53,TSTNO
2341 006666 012737 007032 001216 MOV #TST54,NEXT
2342 006674 012737 006720 001220 MOV #1$,LOCK
2343 006702 112737 000377 012602 MOV #377,PARFLG ; TELL SUBROUTINE PARITY IS ENABLED.
2344 006710 112737 000377 014052 MOV #377,EXTFLG ; TELL SUBROUTINE THIS IS A 16 BIT CHAR.
2345 006716 005000 CLR RO ; CLEAR DATA POINTER
2346 006720 104412 1$: MSTCLR ; INIT DQ11
2347 006722 012737 000377 011766 MOV #377,NPRFLG ; SET FOR SUBROUTINE.
2348 006730 010037 014056 RO,WORD ; LOAD DATA
2349 006734 004737 012444 JSR PC,GENPAR ; CALCULATE PARITY.
2350 006740 032737 100000 014056 BIT #BIT15,WORD ; CHECK PARITY BIT
2351 006746 001404 BEQ .+12 ; BR IF PARITY BIT CLEARED
2352 006750 042737 100000 014056 BIC #BIT15,WORD ; PARITY BIT SET :: SO CLEAR IT.
2353 006756 000403 BR .+10 ; CONTINUE TEST
2354 006760 052737 100000 014056 BIS #BIT15,WORD ; PARITY BIT CLR :: SO SET IT.
2355 006766 004537 011522 JSR RS,RXSTRA ; PUSH CHARACTER INTO RECEIVER
2356 006772 000020 16. ; SHIFTS NEEDED.
2357 006774 000000 0000 ; BITS PER CHAR SELECT.
2358 006776 013737 014116 001252 MOV RXBUFF,TEMP4 ; GET ACTUAL.
2359 007004 013737 014056 001254 MOV WORD,TEMP5 ; GET EXPECTED.
2360 007012 005777 172350 2$: TST @DQERR ; DID THE ERROR FLAG SET...
2361 007016 100401 BMI .+4 ; BR IF AN ERROR OCCURED.
2362 007020 104000 HLT ; ERROR NO ERROR (PARITY ERROR)
2363 007022 104401 SCOP1 ; LOCK ON CHARACTER? (SW09=1)
2364 007024 105200 INCB RO ; UPDATE DATA POINTER.
2365 007026 100334 BPL 1$ ; BR IF NOT 200(8) CHARS DONE.
2366 007030 104400 SCOPE ; SCOPE THIS TEST
2367
2368
2369
2370 ; TEST OF RECEIVER HALF DUPLEX
2371 ; TEST TO TRAIL IIT
2372 ; A TWO HUNDRED CHARACTER BURST OF DATA CHARACTERS
2373 ; WITH THE RECEIVER IN HALF DUPLEX
2374 ; MAKING SURE THAT THE RECEIVER
2375 ; DOESNT RECEIVE ANY CHARACTERS.
2376
2377
2378 ; TEST 54
2379 ; *****
2380 007032 012737 000054 001226 TST54: MOV #54,TSTNO
2381 007040 012737 007434 001216 MOV #TST55,NEXT
2382 007046 005000 CLR RO ; INIT DATA REG
2383 007050 012704 014524 MOV #TXBUFF,R4 ; PREPARE TO FILL TX BUFFER WITH BINARY COUNT.
2384 007054 110024 1$: MOVB RO,(R4)+ ; START FILLING TX BUFF
2385 007056 105200 INCB RO ; UPDATE DATA REG
2386 007060 100375 BPL 1$ ; BRANCH IF BUFFER HASN'T BEEN FILLED
2387 007062 104413 2$: MEMCLR ; INIT THE DEVICE
2388 007064 005000 CLR RO ; CLEAR COUNT REG
2389 007066 012704 014116 MOV #RXBUFF,R4 ; PREPARE TO CLEAR THE RECEIVER BUFFER.
2390 007072 105024 3$: CLRB (R4)+ ; START CLEARING RX BUFF
2391 007074 105200 INCB RO ; UPDATE THE COUNTER
2392 007076 001375 BNE 3$ ; IS RX BUFF ALL CLEARED?
2393 007100 105077 172264 CLRB @DQREG ; SELECT THE RECEIVER BA PRI
2394 007104 012777 014116 172260 MOV #RXBUFF,@DQSEC ; LOAD THE BA
2395 007112 105277 172252 INCB @DQREG ; SELECT THE RECEIVER CC PRI

```

2396	007116	012777	177600	172246		MOV	#-200, @DQSEC	;LOAD THE CC WITH -200 (I WANT TO RECEIVE 200 CHARACTERS
2397	007124	105277	172240			INCB	@DQREG	;SELECT THE TX BA PRI
2398	007130	012777	014522	172234		MOV	#SYNC, @DQSEC	;LOAD THE TX BA WITH STARTING ADD OF TX DATA PLUS THE SY
2399	007136	105277	172226			INCB	@DQREG	;SELECT THE TX CC PRI
2400	007142	012777	177576	172222		MOV	#-202, @DQSEC	;LOAD THE TX CC WITH -202 (FOUR HUNDRED CHARACTERS AND T
2401	007150	112777	000011	172212		MOV	#11, @DQREG	;SELECT THE SYNC REGISTER
2402	007156	013777	014520	172206		MOV	.SYNC, @DQSEC	;LOAD IT WITH THE SYNC CHAR
2403	007164	105277	172200			INCB	@DQREG	;SELECT THE MISC REGISTER
2404	007170	012777	004010	172174		MOV	#4010, @DQSEC	;LOAD IT WITH EIGHT BITS PER/CHAR AND TEST LOOP
2405	007176	005037	001244			CLR	TEMP1	;ZERO DELAY LOC1
2406	007202	012737	000020	001246		MOV	#20, TEMP2	;SET DELAY FOR 20X177777 (8)
2407	007210	012777	000011	172142		MOV	#11, @DQRCR	;SET RECEIVER HALF DUPLEX AND GO!!
2408	007216	005277	172142			INC	@DQCSR	;SET TRANSMITTER GO!!!
2409	007222	105777	172136		4\$:	TSTB	@DQCSR	;TRANSMITTER DONE??
2410	007226	100407				BMI	5\$	;BRANCH IF TRANSMITTER IS DONE.
2411	007230	005237	001244			INC	TEMP1	;START THE DELAY
2412	007234	001372				BNE	4\$	;DELAY-----
2413	007236	005337	001246			DEC	TEMP2	;DELAY-----TRANSMITTER DONE?
2414	007242	001367				BNE	4\$	;DELAY-----
2415	007244	104000				HLT		;TRANSMITTER DONE NEVER SET (PRI)
2416	007246	005000			5\$:	CLR	RO	;INIT COUNT REG
2417	007250	012705	014116			MOV	#RXBUFF, R5	;SET REC DATA POINTER
2418	007254	105725			6\$:	TSTB	(R5)+	;START THE DATA CHECK
2419	007256	001401				BEQ	.+4	;DATA GOOD SO FAR
2420	007260	104000				HLT		;DATA COMPARISON ERROR
2421	007262	105200				INCB	RO	;UPDATE COUNTER
2422	007264	100373				BPL	6\$	;BRANCH IF MORE DATA TO CHECK
2423								
2424								
2425								
2426								
2427								
2428								
2429	007266	104412				MSTCLR		;INIT DQ11
2430	007270	005000				CLR	RO	;ZERO DATA POINTER
2431	007272	012704	014116			MOV	#RXBUFF, R4	;PREPARE TO ZERO RX BUFFER.
2432	007276	105024			7\$:	CLRB	(R4)+	;START CLEARING.
2433	007300	105200				INCB	RO	;DONE?
2434	007302	100375				BPL	7\$	;BR IF MORE TO DO.
2435	007304	105077	172060			CLRB	@DQREG	;SEL RX BA PRI.
2436	007310	012777	014116	172054		MOV	#RXBUFF, @DQSEC	;LOAD IT
2437	007316	105277	172046			INCB	@DQREG	;SEL RX WC PRI.
2438	007322	012777	177600	172042		MOV	#-200, @DQSEC	;LOAD FOR 200 CHARS.
2439	007330	112777	000012	172032		MOV	#MISC, @DQREG	;SLE MISC REGISTER.
2440	007336	012777	004010	172026		MOV	#4010, @DQSEC	;SET EIGHT BITS AND TEST LOOP
2441	007344	005037	001244			CLR	TEMP1	;SET DELAY
2442	007350	012737	000002	001246		MOV	#2, TEMP2	;SET DELAY
2443	007356	012777	010011	171774		MOV	#10011, @DQRCR	;SET ACTIVE HALF DUPLEX. GO
2444	007364	105777	171770		8\$:	TSTB	@DQRCR	;RX DONE PRI?
2445	007370	100407				BMI	9\$	;BR IF YES
2446	007372	005237	001244			INC	TEMP1	;DELAY
2447	007376	001372				BNE	8\$	; " " "
2448	007400	005337	001246			DEC	TEMP2	; " " "
2449	007404	001367				BNE	8\$	; " " "
2450	007406	104000				HLT		;RX PRI. DONE NOT SET..
2451	007410	005000			9\$:	CLR	RO	;INIT COUNTER

;RECEIVER HALF DUPLEX TEST. PART 2  
 ;TEST THAT WHEN TX IS NOT ACTIVE THAT THE RECEIVER  
 ;CAN RECEIVE CHARS.



```

2452 007412 012705 014116      MOV      #RXBUFF,R5      ;GET RX BUFFER.
2453 007416 122725 000377      10$:    CMPB     #377,(R5)+   ;MARK STATE IN BUFFER?
2454 007422 001401                BEQ      .+4             ;BR IF YES
2455 007424 104000                HLT                     ;ERROR
2456 007426 105200                INCB     RO             ;ALL DONE?
2457 007430 100372                BPL      10$           ;BR IF NO.
2458 007432 104400                SCOPE                    ;SCOPE THIS TEST.

```

```

;
; TEST OF DQ11 TRANSMITTER AND RECEIVER
; DATA REALIBILITY.
; DATA IS TRANSFERED FULL RATE
; AT A FOUR HUNDRED CHARACTER BURST
;

```

```

2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471 007434 012737 000055 001226 ; TEST 55
2472 007442 012737 007754 001216 ;*****
2473 007450 005000                TST55:  MOV      #55,TSTNO
2474 007452 012704 014524                MOV      #TST56,NEXT
2475 007456 110024                CLR      RO             ;INIT DATA REG
2476 007460 105200                1$:    MOV      #TXBUFF,R4   ;PREPARE TO FILL TX BUFFER WITH BINARY COUNT.
2477 007462 001375                MOVVB   RO,(R4)+       ;START FILLING TX BUFF
2478 007464 104413                INCB     RO             ;UPDATE DATA REG
2479 007466 005000                BNE     1$             ;BRANCH IF BUFFER HASN'T BEEN FILLED
2480 007470 012704 014116                2$:    MEMCLR                    ;INIT THE DEVICE
2481 007474 105024                CLR      RO             ;CLEAR COUNT REG
2482 007476 105200                MOV      #RXBUFF,R4   ;PREPARE TO CLEAR THE RECEIVER BUFFER.
2483 007500 001375                3$:    CLRB     (R4)+       ;START CLEARING RX BUFF
2484 007502 105077 171662                INCB     RO             ;UPDATE THE COUNTER
2485 007506 012777 014116 171656                BNE     3$             ;IS RX BUFF ALL CLEARED?
2486 007514 105277 171650                CLRB   @DQREG          ;SELECT THE RECEIVER BA PRI
2487 007520 012777 177400 171644                MOV      #RXBUFF,@DQSEC ;LOAD THE BA
2488 007526 105277 171636                INCB   @DQREG          ;SELECT THE RECEIVER CC PRI
2489 007532 012777 014522 171632                MOV      #-400,@DQSEC  ;LOAD THE CC WITH -400 (I WANT TO RECEIVE 400 CHARACTERS)
2490 007540 105277 171624                INCB   @DQREG          ;SELECT THE TX BA PRI
2491 007544 012777 177376 171620                MOV      #SYNC,@DQSEC  ;LOAD THE TX BA WITH STARTING ADD OF TX DATA PLUS THE SY
2492 007552 112777 000011 171610                INCB   @DQREG          ;SELECT THE TX CC PRI
2493 007560 013777 014520 171604                MOV      #-402,@DQSEC  ;LOAD THE TX CC WITH -402 (FOUR HUNDRED CHARACTERS AND T
2494 007566 105277 171576                MOVVB  #11,@DQREG      ;SELECT THE SYNC REGISTER
2495 007572 012777 004010 171572                MOV      .SYNC,@DQSEC  ;LOAD IT WITH THE SYNC CHAR
2496 007600 005037 001244                INCB   @DQREG          ;SELECT THE MISC REGISTER
2497 007604 012737 000020 001246                MOV      #4010,@DQSEC  ;LOAD IT WITH EIGHT BITS PER/CHAR AND TEST LOOP
2498 007612 005277 171542                CLR     TEMP1           ;ZERO DELAY LOC1
2499 007616 005277 171542                MOV     #20,TEMP2      ;SET DELAY FOR 20X177777 (8)
2500 007622 105777 171532                4$:    INC     @DQRCR        ;SET RECEIVER GO!!
2501 007626 100407                INC     @DQRCR        ;SET TRANSMITTER GO!!!
2502 007630 005237 001244                TSTB   @DQRCR         ;RECEIVER DONE??
2503 007634 001372                BMI     5$             ;BRANCH IF RECEIVER IS DONE.
2504 007636 005337 001246                INC     TEMP1           ;START THE DELAY
2505 007642 001367                BNE     4$             ;DELAY-----
2506 007644 104000                DEC     TEMP2           ;DELAY----- REC DONE?
2507 007646 005777 171514                5$:    BNE     4$             ;DELAY-----
                HLT                     ;RECEIVER DONE NEVER SET (PRI)
                TST     @DQERR

```

```

2508 007652 100001          BPL      .+4
2509 007654 104000          HLT
2510 007656 122777 000204 171474  CMPB    #204, @DQRCR
2511 007664 001401          BEQ     .+4
2512 007666 104000          HLT
2513 007670 122777 000204 171466  CMPB    #204, @DQTCR
2514 007676 001401          BEQ     .+4
2515 007700 104000          HLT
2516 007702 005000          CLR     RO              ; INIT COUNT REG
2517 007704 012704 014524  MOV     #TXBUFF, R4     ; SET GOOD DATA POINTER
2518 007710 012705 014116  MOV     #RXBUFF, R5     ; SET REC DATA POINTER
2519 007714 005037 001254 6$: CLR     TEMPS
2520 007720 005037 001252  CLR     TEMP4
2521 007724 112437 001254  MOVB   (R4)+, TEMPS
2522 007730 112537 001252  MOVB   (R5)+, TEMP4
2523 007734 023737 001254  CMP     TEMPS, TEMP4
2524 007742 001401          BEQ     .+4              ; DATA GOOD SO FAR
2525 007744 104025          HLT     25              ; DATA COMPARISON ERROR
2526 007746 105200          INCB   RO              ; UPDATE COUNTER
2527 007750 001361          BNE    6$              ; BRANCH IF MORE DATA TO CHECK
2528 007752 104400          SCOPE

```

```

2529
2530
2531
2532 ; TEST OF THE THREE STRAP SELECTABLE
2533 ; CHARACTERS
2534 ; ON THE FIRST PASS THE CHARACTERS
2535 ; WILL BE TYPED OUT FOR VERIFICATION
2536 ; ON PASSES AFTER THAT THE CHARACTERS WILL BE VERIFIED
2537 ; BY THE PROGRAM.
2538
2539 ; NOTE: IF THE BB OPTION IS INSTALLED
2540 ; PROCEED TO NEXT TEST.

```

```

2541 ; TEST $6
2542 ; *****
2543 007754 012737 000056 001226 TST$6: MOV     #56, TSTNO
2544 007762 012737 015126 001216  MOV     #.EOP, NEXT
2545 007770 012737 010134 001220  MOV     #1$, LOCK
2546 007776 104413          MEMCLR          ; CLEAR ALL
2547 010000 005037 011766          CLR     NPRFLG
2548 010004 032737 020000 001510  BIT     #BBBIT, DQSTAT ; DOES BB OPTION EXIST?
2549 010012 001405          BEQ     .+14       ; BR IF BB NOT THERE.
2550 010014 013737 001216 001214  MOV     NEXT, RETURN ; DO NEXT TEST.
2551 010022 000177 171166          JMP     @RETURN
2552 010026 012737 000010 010154  MOV     #8, 5$      ; EIGHT SHIFTS.
2553 010034 012737 004000 010156  MOV     #4000, 6$   ; EIGHT BITS PER CHAR.
2554 010042 012737 000400 010272  MOV     #400, 15$   ; LAST CHARACTER.
2555 010050 005000          CLR     RO          ; ZERO DATA POINTER

```

```

2556 ; *****
2557 ; MAINTAINANCE AID.
2558 ; THE FOLLOWING IS TO HELP TROUBLE SHOOT
2559 ; PROBLEMS IN THE CHARACTER DET. LOGIC
2560 ; FASTER.
2561 ; *****
2562 ; =====
2563

```

2564										
2565	010052	000416			BR	36\$				; CHANGE THIS LOCATION TO "243" (NOP)
2566										; TO LOCK ON SELECTED 8 BIT CHAR.
2567	010054	000000			HALT					; PUT SELECTED CHARACTER IN SWR.
2568										; HIT CONT.
2569	010056	104414			CKSWR					; CHECK FOR <↑G>
2570	010060	017700	171114		MOV	2SWR,R0				; LOAD CHARACTER.
2571	010064	000000			HALT					; PUT DYNAMIC SWR SETTINGS IN SWR AND
2572										; HIT CONT.
2573	010066	104414			CKSWR					; CHECK FOR <↑G>
2574	010070	000407			BR	36\$				; CHANGE THIS LOCATION TO "240" (NOP)
2575										; ALONG WITH THE ABOVE FOR 16 BIT CHAR
2576										; NOTE: BOTH LOCATIONS ARE TO BE CHANGED
2577										; FOR A 16 BIT CHAR.
2578	010072	012737	000020	010154	MOV	#16..5\$				; SET FOR 16 SHIFTS.
2579	010100	005037	010156		CLR	6\$				; SET "BITS/PER/CHAR"
2580	010104	005037	010272		CLR	15\$				; SET LAST LIMIT.
2581										
2582										
2583										
2584										
2585										
2586	010110	012704	014066		36\$: MOV	#TMPBUF,R4				; STORAGE POINTER.
2587	010114	005024			CLR	(R4)+				; ZERO STORAGE
2588	010116	022704	014104		CMP	#TMPBUF+16,R4				; ALL CLEAR?
2589	010122	001374			BNE	.-6				; BR IF NO.
2590	010124	005037	014114		CLR	NUMBER				; HOW MANY FOUND.
2591	010130	012704	014066		MOV	#TMPBUF,R4				; PREPARE POINTER
2592	010134	005137	011766		1\$: COM	NPRFLG				; TELL SUBROUTINE NOT TO FORCE RX NPR.
2593	010140	005077	171214		CLR	2DQRCR				; CLEAR RX CSR
2594	010144	010037	014056		MOV	R0,WORD				; LOAD CHARACTER
2595	010150	004537	011522		JSR	R5,RXSTRA				; PUSH CHARACTER INTO RECEIVER.
2596	010154	000010			5\$: 8.					; BEWARE THIS LOCATION WILL CHANGE.
2597	010156	004000			6\$: 4000					; BEWARE THIS LOCATION WILL CHANGE.
2598	010160	005777	171174		TST	2DQRCR				; WAS A CHARACTER DETECTED?
2599	010164	100037			2\$: BPL	2\$				; BR IF NO CHAR FOUND.
2600	010166	042777	100000	171164	BIC	#BIT15,2DQRCR				; CLEAR DETECED CHAR FLAG
2601	010174	005700			TST	R0				; WAS THE CHAR=0
2602	010176	001003			BNE	18\$				; BR IF NO.
2603	010200	005737	014114		TST	NUMBER				; HOW MANY WERE FOUND?
2604	010204	001410			BEQ	19\$				; BR IF NONE YET.
2605	010206	012702	014066		18\$: MOV	#TMPBUF,R2				; POINTER STORE.
2606	010212	020022			13\$: CMP	R0,(R2)+				; WAS THIS CHARACTER FOUND BEFORE?
2607	010214	001423			BEQ	2\$				; BR IF YES
2608	010216	005722			TST	(R2)+				; POP POINTER
2609	010220	022702	014106		CMP	#TMPBUF+20,R2				; ALL CHARS CHECKED?
2610	010224	001372			BNE	13\$				; BR IF NO.
2611	010226	010024			19\$: MOV	R0,(R4)+				; STORE CHARACTER
2612	010230	017714	171124		MOV	2DQRCR,(R4)				; GET ADDRESS FOUND IN.
2613	010234	042714	170377		BIC	#170377,(R4)				; CLEAR ALL GARBAGE.
2614	010240	000324			SWAB	(R4)+				; SWAP AROUND.
2615	010242	005237	014114		INC	NUMBER				; UPDATE COUNTER.
2616	010246	022737	000005	014114	CMP	#5,NUMBER				; TOO MANY CHARS FOUND??
2617	010254	001003			BNE	2\$				; BR IF OK.
2618	010256	104050			HLT					; ERROR MORE THAN 4 CHARS. WERE DETECTED.
2619	010260	005177	170730		JMP	2RETURN				; RESTART TEST. DO NOT CONTINUE IN THIS TEST

NOTE SWR BIT 9 MUST BE SET TO LOCK ON THAT CHAR. SELECTED.

=====

DZDQD MACY11 27(732) 24-MAY-76 13:10 PAGE 52  
 DZDQDC.P11 DQ11 TRANSMITTER AND RECEIVER EXERCISER.

2620	010264	104401			2\$:	SCOP1			; LOCK ON CHAR (SW09=1)
2621	010266	005200				INC	RO		; UPDATE CHARACTER
2622	010270	020027				CMP	RO, (PC)+		; ALL DONE?
2623	010272	000000			15\$:	0			; LAST CHAR STORED HERE.
2624	010274	001317				BNE	1\$		; BR IF NOT DONE
2625	010276	005737	014114			TST	NUMBER		; ANY CHARS FOUND?
2626	010302	001024				BNE	30\$		; BR IF NONE FOUND
2627	010304	022737	000020	010154	31\$:	CMP	#16., 5\$		; IS TEST ALL DONE?
2628	010312	001434				BEQ	7\$		; BR IF YES
2629	010314	012737	000020	010154		MOV	#16., 5\$		; DO A 16 BIT CHAR NOW
2630	010322	005037	010156			CLR	6\$		; SET FOR 16 BITS PER CHAR.
2631	010326	112777	000012	171034		MOVB	#MISC., @DQREG		; SEL MISC REG
2632	010334	042777	177400	171030		BIC	#177400, @DQSEC		; CLEAR THE HIGH BYTE
2633	010342	005037	010272			CLR	15\$		; SET LAST CHAR TO 0
2634	010346	005000				CLR	RO		; ZERO DATA POINTER
2635	010350	000137	010134			JMP	1\$		; GO AND DO IT AGAIN
2636	010354	022737	000001	014114	30\$:	CMP	#1, NUMBER		; WAS 1 CHAR FOUND?
2637	010362	001010				BNE	7\$		; BR IF NO.
2638	010364	022737	000010	014070		CMP	#10, TMPBUF+2		; WAS "SYNC DET" ENABLED?
2639	010372	001004				BNE	7\$		; BR IF NO.
2640	010374	005337	014114			DEC	NUMBER		; ZERO NUMBER.
2641	010400	024444				CMP	-(R4), -(R4)		; ADJUST POINTERS
2642	010402	000740				BR	31\$		; KEEP GOING.
2643	010404	005737	014114		7\$:	TST	NUMBER		; ANY FOUND?
2644	010410	001004				BNE	.+12		; BR IF YES
2645	010412	104402	013116			TYPE	.EM4		; ALERT OPERATOR NONE FOUND.
2646	010416	000137	010626			JMP	10\$		; LEAVE
2647	010422	105737	014112			TSTB	XYZFLG		; WAS THIS DONE BEFORE?
2648	010426	001050				BNE	3\$		; BR IF TEST WAS DONE BEFORE
2649	010430	012704	014066			MOV	#TMPBUF, R4		; POINTER
2650	010434	012437	010700			MOV	(R4)+, .CHAR1		; STORE CHARACTER 1
2651	010440	012437	010702			MOV	(R4)+, .ADDR1		; STORE ADDRESS 1
2652	010444	012437	010704			MOV	(R4)+, .CHAR2		; STORE CHARACTER 2
2653	010450	012437	010706			MOV	(R4)+, .ADDR2		; STORE ADDRESS 2
2654	010454	012437	010710			MOV	(R4)+, .CHAR3		; STORE CHARACTER 3
2655	010460	012437	010712			MOV	(R4)+, .ADDR3		; STORE ADDRESS 3
2656	010464	012437	010714			MOV	(R4)+, .CHAR4		; STORE CHARACTER 4
2657	010470	012437	010716			MOV	(R4)+, .ADDR4		; STORE ADDRESS 4
2658	010474	013737	014114	001252		MOV	NUMBER, TEMP4		; STORE NUMBER OF CHARACTER FOUND.
2659	010502	104402				TYPE			
2660	010504	013724				MDETC			
2661	010506	104410				CONVRT			
2662	010510	010630				XCHAR1			
2663	010512	005337	001252			DEC	TEMP4		
2664	010516	001414				BEQ	3\$		
2665	010520	104410				CONVRT			
2666	010522	010642				XCHAR2			
2667	010524	005337	001252			DEC	TEMP4		
2668	010530	001407				BEQ	3\$		
2669	010532	104410				CONVRT			
2670	010534	010654				XCHAR3			
2671	010536	005337	001252			DEC	TEMP4		
2672	010542	001402				BEQ	3\$		
2673	010544	104410				CONVRT			
2674	010546	010666				XCHAR4			
2675	010550	022737	000001	001504	3\$:	CMP	#1, DQNUM		

2676	010556	001003			BNE	+.10
2677	010560	012737	177777	014112	MOV	#-1,XYZFLG
2678	010566	013737	014114	001252	MOV	NUMBER,TEMP4
2679	010574	012704	014066		MOV	#TMPBUF,R4
2680	010600	012705	010700		MOV	#.CHAR1,R5
2681	010604	022425			4\$: CMP	(R4)+,(R5)+
2682	010606	001401			SEQ	+.4
2683	010610	104022			HLT	22
2684	010612	022425			CMP	(R4)+,(R5)+
2685	010614	001401			BEQ	+.4
2686	010616	104022			HLT	22
2687	010620	005337	001252		DEC	TEMP4
2688	010624	001367			SNE	4\$
2689	010626	104400			10\$: SCOPE	
2690	010630	000002			XCHAR1: 2	
2691	010632	006	002		.BYTE	6,2
2692	010634	010700			.CHAR1	
2693	010636	004	002		.BYTE	4,2
2694	010640	010702			.ADDR1	
2695	010642	000002			XCHAR2: 2	
2696	010644	006	002		.BYTE	6,2
2697	010646	010704			.CHAR2	
2698	010650	004	002		.BYTE	4,2
2699	010652	010706			.ADDR2	
2700	010654	000002			XCHAR3: 2	
2701	010656	006	002		.BYTE	6,2
2702	010660	010710			.CHAR3	
2703	010662	004	002		.BYTE	4,2
2704	010664	010712			.ADDR3	
2705	010666	000002			XCHAR4: 2	
2706	010670	006	002		.BYTE	6,2
2707	010672	010714			.CHAR4	
2708	010674	004	002		.BYTE	4,2
2709	010676	010716			.ADDR4	
2710	010700	000000			.CHAR1: 0	
2711	010702	000000			.ADDR1: 0	
2712	010704	000000			.CHAR2: 0	
2713	010706	000000			.ADDR2: 0	
2714	010710	000000			.CHAR3: 0	
2715	010712	000000			.ADDR3: 0	
2716	010714	000000			.CHAR4: 0	
2717	010716	000000			.ADDR4: 0	

TRANSMITTER AND RECEIVER EXERCISER.

\*\*\*\*\*
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: MSTRCLR
CLR WORD
MOV R5, SAVEPC
MOV (R5)+, COUNT
MOV (R5)+, BITSEL
MOVB #2, @DQREG
MOV #WORD, @DQSEC
INCB @DQREG
MOV #-1, @DQSEC
MOV #MISC, @DQREG
MOV BITSEL, @DQSEC
BIS #12, @DQSEC
INC @DQTCR
CHB @DQTCR, @DQTCR : WAIT TIME
CHB @DQTCR, @DQTCR : WAIT TIME
CHB @DQTCR, @DQTCR : WAIT TIME
INCB @DQSEC
DEC @DQSEC
INCB @DQSEC
DEC @DQSEC
BIT #17, @DQSEC
BNE .+4
HLT 23
DEC COUNT
INCB @DQSEC
DEC @DQSEC
BIT #17, @DQSEC
BNE .+4
HLT
RTS

18:

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

Table with 4 columns: Address, Hex, Dec, and Comment. Contains assembly code listings for TXSTRB and TXSTRC.

011214	027777	170144	170142		CMP	2DQTCR,2DQTCR	:WAIST TIME
011222	027777	170136	170134		CMP	2DQTCR,2DQTCR	:WAIST TIME
011230	027777	170130	170126		CMP	2DQTCR,2DQTCR	:WAIST TIME
011236	005277	170130			INC	2DQSEC	
011242	005377	170124			DEC	2DQSEC	
011246	005277	170120		15:	INC	2DQSEC	
011252	005377	170114			DEC	2DQSEC	
011256	005337	014062			DEC	COUNT	
011262	001371				BNE	15	
011264	000205				RTS	R5	
011266	010537	014054		TXSTRD:	MOV	R5,SAVEPC	:SAVE PC OF ROUTINE CALL
011272	012537	014062			MOV	(R5)+,COUNT	:PICK UP THE NUMBER OF SHIFTS
011276	012537	014064			MOV	(R5)+,BITSEL	:PICK UP NUMBER OF BITS PER CHARACTER
011302	112777	000002	170060		MOVB	#2,2DQREG	:SELECT THE TRANSMITTER BA PRI.
011310	012777	014056	170054		MOV	#WORD,2DQSEC	:LOAD THE BA
011316	105277	170046			INCB	2DQREG	:SELECT THE TRANSMITTER CC PRI.
011322	012777	177777	170042		MOV	#-1,2DQSEC	:LOAD THE CC WITH -1
011330	112777	000012	170032		MOVB	#MISC,2DQREG	:SELECT THE MISC REGISTER.
011336	013777	014064	170026		MOV	BITSEL,2DQSEC	:LOAD MISC REG WITH NUMBER OF BITS PER CHAR.
011344	052777	000012	170020		BIS	#12,2DQSEC	:ADD TO THAT TEST LOOP AND AUTO STEP.
011352	105737	012602			TSTB	PARFLG	:IS PARITY TO BE TURNED ON?
011356	001403				BEQ	+10	:BR IF NO
011360	052777	100000	170004		BIS	#BIT15,2DQSEC	:TURN PARITY ON
011366	005277	167772			INC	2DQTCR	:SET TRANSMITTER GO!!!!
011372	027777	167766	167764		CMP	2DQTCR,2DQTCR	:WAIST TIME
011400	027777	167760	167756		CMP	2DQTCR,2DQTCR	:WAIST TIME
011406	027777	167752	167750		CMP	2DQTCR,2DQTCR	:WAIST TIME
011414	005277	167752			INC	2DQSEC	:PRIME THE
011420	005377	167746			DEC	2DQSEC	TRANSMITTER.
011424	006037	001252		15:	ROR	TEMP4	:SHIFT THE STORAGE OF DATA FROM THE TRANSMITTER.
011430	005277	167736			INC	2DQSEC	:CLOCK THE TRANSMITTER -UP-
011434	005377	167732			DEC	2DQSEC	:CLOCK THE TRANSMITTER -DOWN-
011440	017702	167726			MOV	2DQSEC,R2	:MOVE THE MISC REG TO R2
011444	042702	177577			BIC	#177577,R2	:CLEAR ALL BUT THE BIT WINDOW.
011450	105737	014052			TSTB	EXTFLG	:FIND OUT IF BIT PER CHAR > 8
011454	001404				BEQ	25	:BRANCH IF 8OR<8
011456	106102				ROLB	R2	:SHIFT BIT WINDOW INTO CARRY BIT.
011460	006002				ROR	R2	:SHIFT CARRY INTO R2 (BIT 15 OF R2)
011462	042702	077777			BIC	#77777,R2	:CLEAR ALL BUT THAT BIT OF DATA
011466	050237	001252		25:	BIS	R2,TEMP4	:PLACE DATA INTO TEMPORARY LOCATION
011472	005337	014062			DEC	COUNT	:IS CHARACTER COMPLETELY SHIFTED OUT?
011476	001352				BNE	15	:BRANCH IF MORE BITS TO GO.
011500	105737	014052			TSTB	EXTFLG	
011504	001003				BNE	35	
011506	105137	001252			COMB	TEMP4	
011512	000402				BR	45	
011514	005137	001252		35:	COM	TEMP4	:COMPLIMENT DATA STORAGE
011520	000205			45:	RTS	R5	:LEAVE THE ROUTINE.

000000									
000001	011522	010537	014054			RAS*RA:	MOV	R5,SAVEPC	
000002	011526	012537	014062				MOV	(R5)+,COUNT	
000003	011532	012537	014064				MOV	(R5)+,BITSEL	
000004	011536	013737	014056	017770			MOV	WORD,TEMP	
000005	011544	005137	017770				COM	TEMP	
000006	011550	105077	167614				CLRB	JDQREG	
000007	011554	012777	014116	167610			MOV	#RXBUFF,JDQSEC	
000008	011562	105277	167602				INCB	JDQREG	
000009	011566	012777	000200	167576			MOV	#200,JDQSEC	
000010	011574	112777	000011	167566			MOVB	#1,JDQREG	
000011	011602	012777	177777	167562			MOV	#-1,JDQSEC	
000012	011610	105277	167554				INCB	JDQREG	
000013	011614	053777	014064	167550			BIS	BITSEL,JDQSEC	
000014	011622	052777	000012	167542			BIS	#12,JDQSEC	
000015	011630	105737	012602				TSTB	PARFLG	
000016	011634	001403					BEQ	+.10	
000017	011636	052777	100000	167526			BIS	#BIT15,JDQSEC	
000018	011644	052777	000001	167506			BIS	#0001,JDQRCSR	
000019	011652	005737	011766				TST	NPRFLG	
000020	011656	001403					BEQ	+.10	
000021	011660	052777	010000	167472			BIS	#BIT12,JDQRCSR	
000022	011666	112777	000012	167474			MOVB	#MISC,JDQREG	
000023	011674	042777	000200	167470	25:		BIC	#BIT7,JDQSEC	
000024	011702	006037	017770				ROR	TEMP	
000025	011706	106037	001244				RORB	TEMP1	
000026	011712	042737	177577	001244			BIC	#177577,TEMP1	
000027	011720	053777	001244	167444			BIS	TEMP1,JDQSEC	
000028	011726	005277	167440				INC	JDQSEC	
000029	011732	005377	167434				DEC	JDQSEC	
000030	011736	005337	014062				DEC	COUNT	
000031	011742	001354					BNE	25	
000032	011744	005737	011766				TST	NPRFLG	
000033	011750	001003					BNE	+.10	
000034	011752	052777	000020	167412			BIS	#BIT4,JDQSEC	
000035	011760	005037	011766				CLR	NPRFLG	
000036	011764	000205					RTS	R5	
000037	011766	000000							
000038	011770						NPRFLG:	0	
000039	011770	005077	167364				.MEMCLR:		
000040	011774	005077	167364				CLR	JDQRCSR	
000041	012000	005077	167362				CLR	JDQTCR	
000042	012004	012705	000020				CLR	JDQERR	
000043	012010	152777	000020	167352	15:		MOV	#16,R5	
000044	012016	142777	000140	167344			BISB	#BIT4,JDQREG	
000045	012024	005077	167342				BICB	#140,JDQREG	
000046	012030	105277	167334				CLR	JDQSEC	
000047	012034	005305					INCB	JDQREG	
000048	012036	001364					DEC	R5	
000049	012040	105077	167324				BNE	15	
000050	012044	105077	167312				CLRB	JDQREG	
000051	012050	012705	000020				CLRB	JDQRCSR	
000052	012054	112777	000010	167306	25:		MOV	#16,R5	
000053	012062	005077	167304				MOVB	#10,JDQREG	
000054	012066	112777	000014	167274			CLR	JDQSEC	
000055	012074	005077	167272				MOVB	#14,JDQREG	
							CLF	JDQSEC	

: IS PARITY TO BE TURNED ON?  
: BP IF NO  
: TJRN PARITY ON.....



00000006	012100	105277	167256		INCB	ADQRC5H
00000007	012104	005305			DEC	R5
00000008	012106	001362			BNE	Z5
00000009	012110	105077	167246		CLRB	ADQRC5H
00000010	012114			.MSTCLR:		
00000011	012114	112777	000012	167246	MOVB	#15C, ADQREG
00000012	012122	012777	000040	167242	MOV	#BITS, ADQSEC
00000013	012130	000002			RTI	
00000014	012132	010537	014054		RXLNG: MOV	R5, SAVEPC
00000015	012136	104412			MSTCLR	
00000016	012140	105077	167224		CLRB	ADQREG
00000017	012144	012777	014116	167220	MOV	#RXBUFF, ADQSEC
00000018	012152	005037	014116		CLR	RXBUFF
00000019	012156	105277	167206		INCB	ADQREG
00000020	012162	012777	000200	167202	MOV	#200, ADQSEC
00000021	012170	112777	000011	167172	MOVB	#11, ADQREG
00000022	012176	013777	014520	167166	MOV	.SYNC, ADQSEC
00000023	012204	105277	167160		INCB	ADQREG
00000024	012210	012577	167156		MOV	(R5)+, ADQSEC
00000025	012214	052777	000012	167150	BIS	#12, ADQSEC
00000026	012222	052777	000001	167130	BIS	#0001, ADQRC5R
00000027	012230	042777	000200	167134	BIC	#BIT7, ADQSEC
00000028	012236	005277	167130		INC	ADQSEC
00000029	012242	005377	167124		DEC	ADQSEC
00000030	012246	052777	000020	167116	BIS	#BIT4, ADQSEC
00000031	012254	000240			NOP	
00000032	012256	000240			NOP	
00000033	012260	000240			NOP	
00000034	012262	000337	014116		SWAB	RXBUFF
00000035	012266	122537	014116		18: CMPB	(R5)+, RXBUFF
00000036	012272	001401			BEQ	.+4
00000037	012274	104015			HLT	15
00000038	012276	005205			INC	R5
00000039	012300	000205			RTS	R5
00000040	012302	010537	014054		RXLNG: MOV	R5, SAVEPC
00000041	012306	104412			MSTCLR	
00000042	012310	105077	167054		CLRB	ADQREG
00000043	012314	012777	014116	167050	MOV	#RXBUFF, ADQSEC
00000044	012322	005037	014116		CLR	RXBUFF
00000045	012326	105277	167036		INCB	ADQREG
00000046	012332	012777	000200	167032	MOV	#200, ADQSEC
00000047	012340	112777	000011	167022	MOVB	#11, ADQREG
00000048	012346	013777	014520	167016	MOV	.SYNC, ADQSEC
00000049	012354	105277	167010		INCB	ADQREG
00000050	012360	012577	167006		MOV	(R5)+, ADQSEC
00000051	012364	052777	000012	167000	BIS	#12, ADQSEC
00000052	012372	052777	000001	166760	BIS	#0001, ADQRC5R
00000053	012400	042777	000200	166764	BIC	#BIT7, ADQSEC
00000054	012406	005277	166760		INC	ADQSEC
00000055	012412	005277	166754		DEC	ADQSEC
00000056	012416	052777	000020	166746	BIS	#BIT4, ADQSEC
00000057	012424	000240			NOP	
00000058	012426	000240			NOP	
00000059	012430	000240			NOP	
00000060	012432	022537	014116		CMP	(R5)+, RXBUFF
00000061	012436	001401			BEQ	.+4

2942	012440	104015			HLT	15
2943	012442	000205			RTS	R5
2944	012444			GENPAR:		
2945	012444	010146			MOV	R1, -(SP)
2946	012446	010246			MOV	R2, -(SP)
2947	012450	010346			MOV	R3, -(SP)
2948	012452	105737	014052		TSTB	EXTFLG
2949	012456	001003			BNE	.+10
2950	012460	042737	000200	014056	BIC	#BIT7, WORD
2951	012466	042737	100000	014056	BIC	#BIT15, WORD
2952	012474	005002			CLR	R2
2953	012476	012703	000020		MOV	#16., R3
2954	012502	013701	014056		MOV	WORD, R1
2955	012506	000241			CLC	
2956	012510	006001		1\$:	ROR	R1
2957	012512	005502			ADC	R2
2958	012514	005303			DEC	R3
2959	012516	001374			BNE	1\$
2960	012520	032737	001000	001510	BIT	#ODDBIT, DQSTAT
2961	012526	001404			BEQ	2\$
2962	012530	032702	000001		BIT	#BIT0, R2
2963	012534	001016			BNE	4\$
2964	012536	000403			BR	3\$
2965	012540	032702	000001	2\$:	BIT	#BIT0, R2
2966	012544	001412			BEQ	4\$
2967	012546	105737	014052	3\$:	TSTB	EXTFLG
2968	012552	001004			BNE	.+12
2969	012554	052737	000200	014056	BIS	#BIT7, WORD
2970	012562	000403			BR	4\$
2971	012564	052737	100000	014056	BIS	#BIT15, WORD
2972	012572	012603		4\$:	MOV	(SP)+, R3
2973	012574	012602			MOV	(SP)+, R2
2974	012576	012601			MOV	(SP)+, R1
2975	012600	000207			RTS	PC
2976	012602	000000			PARFLG:	0
2977						

Address	Hex	Hex	Label	Comment
2978	012604		.ERRTAB:	
2979	012604	000000	0	
2980	012606	000000	0	:HALT 0
2981	012610	000000	0	
2982	012612	013010	EM0	
2983	012614	013326	DH1	:HALT 1
2984	012616	000000	0	
2985	012620	013010	EM0	
2986	012622	013347	CH2	:HALT 2
2987	012624	000000	0	
2988	012626	013026	EM1	
2989	012630	013364	DH3	:HALT 3
2990	012632	000000	0	
2991	012634	013026	EM1	
2992	012635	013375	DH4	:HALT 4
2993	012640	070000	0	
2994	012642	013026	EM1	
2995	012644	013431	DH5	:HALT 5
2996	012646	000000	0	
2997	012650	013010	EM0	
2998	012652	013465	DH6	:HALT 6
2999	012654	000000	0	
3000	012656	013010	EM0	
3001	012660	013505	DH7	:HALT 7
3002	012662	000000	0	
3003	012664	013173	EM6	
3004	012666	013540	DH9	:HALT 10
3005	012670	000000	0	
3006	012672	000000	0	
3007	012674	013534	DH8	:HALT 11
3008	012676	000000	0	
3009	012700	013010	EM0	
3010	012702	013546	DH10	:HALT 12
3011	012704	014040	DTC	
3012	012706	013064	EM3	
3013	012710	013540	DH9	:HALT 13
3014	012712	000000	0	
3015	012714	013064	EM3	
3016	012716	013534	DH8	:HALT 14
3017	012720	000000	0	
3018	012722	013173	EM6	
3019	012724	013675	DH13	:HALT 15
3020	012726	000000	0	
3021	012730	013150	EM5	
3022	012732	013534	DH8	:HALT 16
3023	012734	000000	0	
3024	012736	013150	EM5	
3025	012740	013540	DH9	:HALT 17
3026	012742	000000	0	
3027	012744	013173	EM6	
3028	012746	013546	DH10	:HALT 20
3029	012750	014040	DTC	
3030	012752	013116	EM4	
3031	012754	000000	0	:HALT 21
3032	012756	000000	0	
3033	012760	013206	EM7	

# H05

DZDQD MACY11 27(732) 24-MAY-76 13:10 PAGE 60  
DZDQDC.P11 DQ11 TRANSMITTER AND RECEIVER EXERCISER.

3034	012762	000000			0	:HALT 22
3035	012764	000000			0	
3036	012766	013254			EM9	
3037	012770	000000			0	:HALT 23
3038	012772	000000			0	
3039	012774	013010			EM0	
3040	012776	013306			DHO	:HALT 24
3041	013000	000000			0	
3042	013002	000000			0	
3043	013004	013546			DH10	:HALT 25
3044	013006	014040			OTO	
3045	013010	052377	040522	051516	EM0:	.ASCIZ <377>/TRANSMITTER /
	013026	052377	040522	051516	EM1:	.ASCIZ <377>/TRANSMITTER CHARACTER COUNT /
	013064	053377	041522	042440	EM3:	.ASCIZ <377>/VRC ERROR BIT SHOULD BE /
	013116	047377	020117	044103	EM4:	.ASCIZ <377>/NO CHARACTERS DETECTED./<0>
	013150	051777	047131	020103	EM5:	.ASCIZ <377>/SYNC 1 AND 2 NOT /
	013173	377	042522	042503	EM6:	.ASCIZ <377>/RECEIVER /
	013206	041777	040510	040522	EM7:	.ASCIZ <377>/CHARACTER DETECTION COMPARISON ERROR/
	013254	041777	040510	040522	EM8:	.ASCIZ <377>/CHARACTER NOT ALL ZERO'S/
	013306	041501	044524	042526	DH0:	.ASCIZ /ACTIVE NOT SET./
	013326	041501	044524	042526	DH1:	.ASCIZ /ACTIVE NOT CLEAR/
	013347	104	047117	020105	DH2:	.ASCIZ /DONE NOT SET/
	013364	047516	020124	042532	DH3:	.ASCIZ /NOT ZERO/
	013375	116	052117	044440	DH4:	.ASCIZ /NOT INCREMENTED BY PLUS TWO/
	013431	116	052117	044440	DH5:	.ASCIZ /NOT INCREMENTED BY PLUS ONE/
	013465	120	044522	051452	CH6:	.ASCIZ /PRI*SEC NOT SET/
	013505	114	047111	020105	DH7:	.ASCIZ /LINE NOT AT MARK STATE/
	013534	042523	000124		DH8:	.ASCIZ /SET/
	013540	046103	040505	000122	DH9:	.ASCIZ /CLEAR/
	013546	040504	040524	041440	DH10:	.ASCIZ /DATA COMPARISON ERROR/
	013573	377	054105	042520		.ASCIZ <377>/EXPECTED RECEIVED /
	013621	123	052105	053440	DH11:	.ASCIZ /SET WHEN ACTIVE SET/
	013645	103	042514	051101	DH12:	.ASCIZ /CLEARED BY MASTER CLEAR/
	013675	103	040510	040522	DH13:	.ASCIZ /CHARACTER LENGTH ERROR/
	013724	051777	042505	040440	MDETC:	.ASCII <377>/SEE ABSTRACT OR TEST #56 FOR DETAILS/
	013771	377	044103	051101		.ASCII <377>/CHARACTERS DETECTED: /
	014020	041777	040510	027122		.ASCIZ <377>/CHAR. ADDR. /
					.EVEN	
	014040	000002			OTO:	2
3046	014042	006	004		.BYTE	6 4
3047	014044	001254			TEMP5	
3048	014046	006	002		.BYTE	6 2
3049	014050	001252			TEMP4	
3050	014052	000000			EXTFLG:	0
3051	014054	000000			SAVEPC:	0
3052	014056	000000			WORD:	0
3053	014060	000000			DELAY:	0
3054	014062	000000			COUNT:	0
3055	014064	000000			BITSEL:	0
3056	014066	000012			TMPBUF:	.BLKW 12
3057	014112	000000			XYZFLG:	0
3058	014114	000000			NUMBER:	0
3059	014116	000000			RXBUF:	0
3060		014520			.+400	
3061	014520	026	026		.SYNC:	.BYTE 26,26
3062	014522	026	026		SYNC:	.BYTE 26,26

3063	014524	000000				TXBUFF: 0			
3064		015126				. = .+400			
3065									
3066									:END OF PASS
3067									:TYPE NAME OF TEST
3068									:UPDPTC PASS COUNT
3069									:CHECK FOR EXIT TO ACT-11
3070									:RESTART TEST
3071									
3072	015126	005037	001234			.EOP:	CLR	LSTERR	:CLEAR LAST ERROR PC
3073	015132	005037	001312				CLR	ERRFLG	:CLEAR ERROR FLAG
3074	015136	005237	001230				INC	PASCNT	:UPDATE PASS COUNT
3075	015142	104402					TYPE		
3076	015144	017356					MEPASS		
3077	015146	104402					TYPE		
3078	015150	017536					MCSRX		
3079	015152	104411					CNVRT		
3080	015154	015264					XCSR		
3081	015156	104402					TYPE		
3082	015160	017544					MVECX		
3083	015162	104411					CNVRT		
3084	015164	015272					XVEC		
3085	015166	104402					TYPE		
3086	015170	017552					MPASSX		
3087	015172	104411					CNVRT		
3088	015174	015300					XPASS		
3089	015176	104402					TYPE		
3090	015200	017563					MERRX		
3091	015202	104411					CNVRT		
3092	015204	015306					XERR		
3093	015206	013777	001230	163766			MOV	PASCNT, @LIGHTS	:DISPLAY PASS COUNT
3094	015214	005337	001276				DEC	SAVNUM	
3095	015220	001013					BNE	RESTR	
3096	015222	013737	001504	001276			MOV	DQNUM, SAVNUM	
3097	015230	013701	000042				MOV	@#42, R1	:CHECK FOR ACT-11 OR DDP
3098	015234	001405					BEQ	RESTR	:IF NOT, CONTINUE TESTING
3099	015236	000005					RESET		
3100	015240					LOGICAL:			
3101	015240	004711					JSR	PC, (R1)	
3102	015242	000240					NOP		
3103	015244	000240					NOP		
3104	015246	000240					NOP		
3105	015250	104414				RESTR:	CKSWR		
3106	015252	012737	002254	001214			MOV	@TST1, RETURN	
3107	015260	000137	002254				JMP	TST1	
3108	015264	000001				XCSR:	1		
3109	015266	006	002				.BYTE	6, 2	
3110	015270	001360					DQRCSR		
3111	015272	000001				XVEC:	1		
3112	015274	003	002				.BYTE	3, 2	
3113	015276	001350					DQRVEC		
3114	015300	000001				XPASS:	1		
3115	015302	006	002				.BYTE	6, 2	
3116	015304	001230					PASCNT		
3117	015306	000001				XERR:	1		
3118	015310	006	002				.BYTE	6, 2	

```

3119 015312 001232          ERRCNT
3120
3121          ;SCOPE LOOP AND INTERATION HANDLER
3122
3123 015214 104414          .SCOPE: CKSWR
3124 015316 032777 040000 163654 TTST: BIT      #BIT14, @SWR
3125 015324 001407          BEQ      1$
3126 015326 000432          BR       3$
3127 015330 105777 163650 TSTB   @TKCSR
3128 015334 100027          BPL     3$
3129 015336 017700 163644 MOV    @TKDBR, R0
3130 015342 000412          BR      2$
3131 015344 032777 004000 163626 1$: BIT    #SW11, @SWR
3132 015352 001006          BNE    2$
3133 015354 005237 001224 INC    LPCNT
3134 015360 023737 001224 001222 CMP    LPCNT, ICOUNT
3135 015366 001012          BNE    3$
3136 015370 105037 001312 2$: CLRB  ERRFLG
3137 015374 005037 001224 CLR    LPCNT
3138 015400 012737 000010 001222 MOV    #10, ICOUNT
3139 015406 013737 001216 001214 MOV    NEXT, RETURN
3140 015414 013716 001214 3$: MOV    RETURN, (SP)
3141 015420 000002          RTI
3142 015422 001407          BRW:   1407
3143 015424 000432          BRX:   432
3144
3145          ;CHECK FOR FREEZE ON CURRENT DATA
3146
3147 015426 104414          .SCOPE1: CKSWR
3148 015430 032777 001000 163542 BIT    #SW09, @SWR
3149 015436 001402          BEQ    1$
3150 015440 013716 001220 MOV    LOCK, (SP)
3151 015444 000002          1$: RTI
3152
3153          ;TELETYPE OUTPUT ROUTINE
3154
3155 015446 010546          .TYPE: MOV    R5, -(SP)
3156 015450 017605 000002 MOV    @2(SP), R5
3157 015454 062766 000002 000002 ADD    #2, 2(SP)
3158 015462 005737 017136 1$: TST   @#RDSW
3159 015466 001004          BNE    300$
3160 015470 032777 010000 163502 BIT    #SW12, @SWR
3161 015476 001024          BNE    3$
3162 015500 105715 300$: TSTB  (R5)
3163 015502 100014          BPL    2$
3164 015504 105777 163500 TSTB  @TPCSR
3165 015510 100375          BPL    .-4
3166 015512 012777 000015 163472 MOV    #15, @TPDBR
3167 015520 105777 163464 TSTB  @TPCSR
3168 015524 100375          BPL    .-4
3169 015526 012777 000012 163456 MOV    #12, @TPDBR
3170 015534 105777 163450 2$: TSTB  @TPCSR
3171 015540 100375          BPL    2$
3172 015542 112577 163444 MOVB  (R5)+, @TPDBR
3173 015546 001345          BNE    1$
3174 015550 012605 3$: MOV    (SP)+, R5
  
```

DZDQD MACY11 27(732) 24-MAY-76 13:10 PAGE 63  
 DZDQDC.P11 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

```

3175 015552 000002 RTI
3176
3177 ;ASCII STRING INPUT ROUTINE
3178
3179 015554 010346 .INSTR: MOV R3,-(SP)
3180 015556 010446 MOV R4,-(SP)
3181 015560 017637 000004 015576 MOV @4(SP),.MSG
3182 015566 062766 000002 000004 ADD #2,4(SP)
3183 015574 104402 .INST1: TYPE
3184 015576 000000 .MSG: 0
3185 015600 012704 017726 MOV #INBUF,R4
3186 015604 012703 000007 MOV #7,R3
3187 015610 105777 163370 1$: TSTB @TKCSR
3188 015614 100375 BPL 1$
3189 015616 117714 163364 MOVB @TKDBR,(R4)
3190 015622 142714 000200 BICB #200,(R4)
3191 015626 121427 000025 CMPB (R4),#25 ;IS IT <↑G>
3192 015632 001003 BNE 200$
3193 015634 104402 017316 TYPE,MCRLF
3194 015640 000755 BR .INST1
3195 015642 122427 000015 200$: CMPB (R4)+,#15
3196 015646 001423 BEQ INSTR2
3197 015650 117777 163332 163334 MOVB @TKDBR,@TPDBR
3198 015656 105777 163326 2$: TSTB @TPCSR
3199 015662 100375 BPL 2$
3200 015664 005303 DEC R3
3201 015666 001350 BNE 1$
3202 015670 000402 BR .INSTG
3203 015672 010346 .INSTE: MOV R3,-(SP)
3204 015674 010446 .INSTG: MOV R4,-(SP)
3205 015676 104402 .INSTG: TYPE
3206 015700 017312 MQM
3207 015702 005737 017136 TST @#RDSW
3208 015706 001402 BEQ 400$
3209 015710 104402 017316 TYPE,MCRLF
3210 015714 000727 400$: BR .INST1
3211 015716 012604 INSTR2: MOV (SP)+,R4
3212 015720 012603 MOV (SP)+,R3
3213 015722 000002 RTI
3214
3215 ;CONVERT ASCII STRING TO OCTAL
3216
3217 015724 010546 .PARAM: MOV R5,-(SP)
3218 015726 010446 MOV R4,-(SP)
3219 015730 016605 000004 MOV 4(SP),R5
3220 015734 012537 016130 MOV (R5)+,LOLIM
3221 015740 012537 016132 MOV (R5)+,HILIM
3222 015744 012537 016134 MOV (R5)+,DEVADR
3223 015750 112537 016136 MOVB (R5)+,LOBITS
3224 015754 112537 016137 MOVB (R5)+,ADRCNT
3225 015760 010566 000004 MOV R5,4(SP)
3226 015764 005005 PARAM1: CLR R5
3227 015766 012704 017726 MOV #INBUF,R4
3228 015772 122714 000015 CMPB #15,(R4)
3229 015776 001420 BEQ PARERR
3230 016000 121427 000060 1$: CMPB (R4),#60

```

3231	016004	002415			BLT	PARERR	
3232	016006	121427	000067		CMPB	(R4),#67	
3233	016012	003012			BGT	PARERR	
3234	016014	142714	000060		BICB	#60,(R4)	
3235	016020	152405			BISB	(R4)+,R5	
3236	016022	122714	000015		CMPB	#15,(R4)	
3237	016026	001414			BEQ	LIMITS	
3238	016030	006305			ASL	R5	
3239	016032	006305			ASL	R5	
3240	016034	006305			ASL	R5	
3241	016036	000760			BR	1\$	
3242	016040	122714	000015	PARERR:	CMPB	#15,(R4)	; IS FIRST CHARACTER A <CR>
3243	016044	001003			BNE	120\$	
3244	016046	005737	017136		TST	@#RDSW	; IS CKSWR ROUTINE BEING USED
3245	016052	001023			BNE	PARTI	
3246	016054	104404		120\$:	INSTER		
3247	016056	000742			BR	PARAM1	
3248							
3249							; TEST TO SEE IF NUMBER IS WITHIN LIMITS
3250							
3251	016060	020537	016132	LIMITS:	CMP	R5,HILIM	
3252	016064	101365			BHI	PARERR	
3253	016066	020537	016130		CMP	R5,LOLIM	
3254	016072	103762			BLO	PARERR	
3255	016074	133705	016136		BITB	LOBITS,R5	
3256	016100	001357			BNE	PARERR	
3257							
3258							; STORE NUMBER AT SPECIFIED ADDRESS
3259							
3260	016102	013704	016134		MOV	DEVADR,R4	
3261	016106	010524		1\$:	MOV	R5,(R4)+	
3262	016110	062705	000002		ADD	#2,R5	
3263	016114	105337	016137		DECB	ADRCNT	
3264	016120	001372			BNE	1\$	
3265	016122	012604		PARTI:	MOV	(SP)+,R4	
3266	016124	012605			MOV	(SP)+,R5	
3267	016126	000002			RTI		
3268	016130	000000		LOLIM:	0		
3269	016132	000000		HILIM:	0		
3270	016134	000000		DEVADR:	0		
3271	016136	000000		LOBITS:	0		
3272		016137			ADRCNT=LOBITS+1		
3273							
3274							; SAVE PC OF TEST THAT FAILED AND R0-R5
3275							
3276	016140	016637	000004	001274	.SAV05:	MOV	4(SP),SAVPC
3277							
3278							; SAVE R0-R5
3279							
3280	016146	010537	001270	SV05:	MOV	R5,SAVR5	
3281	016152	010437	001266		MOV	R4,SAVR4	
3282	016156	010337	001264		MOV	R3,SAVR3	
3283	016162	010237	001262		MOV	R2,SAVR2	
3284	016166	010137	001260		MOV	R1,SAVR1	
3285	016172	010037	001256		MOV	R0,SAVR0	
3286	016176	000002			RTI		



```

3287
3288
3289
3290 016200 013700 001256 .RES05: MOV SAVR0,R0
3291 015204 013701 001260 MOV SAVR1,R1
3292 016210 013702 001262 MOV SAVR2,R2
3293 016214 013703 001264 MOV SAVR3,R3
3294 016220 013704 001266 MOV SAVR4,R4
3295 016224 013705 001270 MOV SAVR5,R5
3296 016230 000002 RTI
3297
3298 ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
3299
3300 016232 104402 .CONVR: TYPE
3301 016234 017316 MCRLF
3302 016236 010046 .CNVRT: MOV RO,-(SP)
3303 016240 010146 MOV R1,-(SP)
3304 016242 010346 MOV R3,-(SP)
3305 016244 010446 MOV R4,-(SP)
3306 016246 010546 MOV R5,-(SP)
3307 016250 017601 000012 MOV @12(SP),R1
3308 016254 013737 017770 001250 MOV TEMP,TEMP3
3309 016262 062766 000002 000012 ADD #2,12(SP)
3310 016270 012137 016452 MOV (R1)+,WRDCNT
3311 016274 112137 016454 1$: MOV (R1)+,CHRCNT
3312 016300 112137 016455 MOV (R1)+,SPACNT
3313 016304 013137 016456 MOV @2(R1)+,BINWRD
3314 016310 013704 016456 2$: MOV BINWRD,R4
3315 016314 113705 016454 MOV CHRCNT,R5
3316 016320 012700 017770 3$: MOV #TEMP,R0
3317 016324 010403 MOV R4,R3
3318 016326 042703 177770 BIC #177770,R3
3319 016332 062703 000060 ADD #060,R3
3320 016336 110320 MOV R3,(R0)+
3321 016340 000241 CLC
3322 016342 006004 ROR R4
3323 016344 000241 CLC
3324 016346 006004 ROR R4
3325 016350 000241 CLC
3326 016352 006004 ROR R4
3327 016354 005305 DEC R5
3328 016356 001362 BNE 3$
3329 016360 012703 020032 MOV #MDATA,R3
3330 016364 114023 4$: MOV -(R0),(R3)+
3331 016366 105337 016454 DECB CHRCNT
3332 016372 001374 BNE 4$
3333 016374 105737 016455 TSTB SPACNT
3334 016400 001405 BEQ 6$
3335 016402 112723 000040 5$: MOV #040,(R3)+
3336 016406 105337 016455 DECB SPACNT
3337 016412 001373 BNE 5$
3338 016414 105013 6$: CLRB (R3)
3339 016416 104402 TYPE
3340 016420 020032 MDATA
3341 016422 005337 016452 DEC WRDCNT
3342 016426 001322 BNE 1$

```

DZDQD MACY11 27(732) 24-MAY-76 13:10 PAGE 66  
 DZDQDC.P11 GENERAL UTILITIES (TYPE OUT, ERROR, SCOPE, ETC.)

```

3343 016430 013737 001250 017770      MOV      TEMP3,TEMP
3344 016436 012605      MOV      (SP)+,R5
3345 016440 012604      MOV      (SP)+,R4
3346 016442 012603      MOV      (SP)+,R3
3347 016444 012601      MOV      (SP)+,R1
3348 016446 012600      MOV      (SP)+,R0
3349 016450 000002      RTI
3350 016452 000000      WRDCNT: 0
3351 016454 000000      CHRCNT: 0
3352          016455      SPACNT=CHRCNT+1
3353 016456 000000      BINWRD: 0
3354          ;TRAP DISPATCH SERVICE
3355          ;ARGUMENT OF TRAP IS EXTRACTED
3356          ;AND USED AS OFFSET TO OBTAIN POINTER
3357          ;TO SELECTED SUBROUTINE
3358
3359 016460 011646      .TRPSR: MOV      (SP)-(SP)      ;GET PC OF RETURN
3360 016462 162716 000002      SUB      #2,(SP)      ;=PC OF TRAP
3361 016466 017616 000000      MOV      @ (SP), (SP) ;GET TRP
3362 016472 006316      TRPOK:  ASL      (SP)      ;MULTIPLY TRAP ARG BY 2
3363 016474 042716 177001      BIC      #177001,(SP) ;CLEAR UNWANTED BITS
3364 016500 062716 001314      ADD      #.TRPTAB,(SP) ;POINTER TO SUBROUTINE ADDRESS
3365 016504 017616 000000      MOV      @ (SP), (SP) ;SUBROUTINE ADDRESS
3366 016510 000136      JMP      @ (SP)+      ;GO TO SUBROUTINE
3367
3368          ;ERROR HANDLER
3369
3370 016512 104414      .HLT:   CKSWR
3371 016514 032777 010000 162456      BIT      #SW12,@SWR
3372 016522 001406      BEQ      XBX
3373 016524 105777 162460      TSTB    @TPCSR
3374 016530 100003      BPL      XBX
3375 016532 112777 000207 162452      MOVB    #207,@TPDBR
3376 016540 032777 020000 162432 XBX:    BIT      #SW13,@SWR
3377 016546 001074      BNE     HALTS
3378 016550 021637 001234      CMP     (SP),LSTERR
3379 016554 001404      BEQ     1$
3380 016556 011637 001234      MOV     (SP),LSTERR
3381 016562 105037 001312      CLRB   ERRFLG
3382 016566 104406      1$:    SAVOS
3383 016570 011605      MOV     (SP),R5
3384 016572 162705 000002      SUB     #2,R5
3385 016576 011504      MOV     (R5),R4
3386 016600 006304      ASL    R4
3387 016602 061504      ADD    (R5),R4
3388 016604 006304      ASL    R4
3389 016606 042704 177001      BIC    #177001,R4
3390 016612 062704 012604      ADD    #.ERRTAB,R4
3391 016616 012437 016710      MOV    (R4)+,ERRMSG
3392 016622 012437 016722      MOV    (R4)+,DATAHD
3393 016626 011437 016734      MOV    (R4),DATABP
3394 016632 105737 001312      TSTB   ERRFLG
3395 016636 001403      BEQ    TYPMSG
3396 016640 005737 016734      TST    DATABP
3397 016644 001027      BNE    TYPDAT
3398 016646 104402      TYPMSG: TYPE

```

001  
002  
003  
004  
005  
006  
007  
008  
009  
010  
011  
012  
013  
014  
015  
016  
017  
018  
019  
020  
021  
022  
023  
024  
025  
026  
027  
028  
029  
030  
031  
032  
033  
034  
035  
036  
037  
038  
039  
040  
041  
042  
043  
044  
045  
046  
047  
048  
049  
050  
051  
052  
053  
054  
055  
056  
057  
058  
059  
060  
061  
062  
063  
064  
065  
066  
067  
068  
069  
070  
071  
072  
073  
074  
075  
076  
077  
078  
079  
080  
081  
082  
083  
084  
085  
086  
087  
088  
089  
090  
091  
092  
093  
094  
095  
096  
097  
098  
099  
100

017000  
017001  
017002  
017003  
017004  
017005  
017006  
017007  
017008  
017009  
017010  
017011  
017012  
017013  
017014  
017015  
017016  
017017  
017018  
017019  
017020  
017021  
017022  
017023  
017024  
017025  
017026  
017027  
017028  
017029  
017030  
017031  
017032  
017033  
017034  
017035  
017036  
017037  
017038  
017039  
017040  
017041  
017042  
017043  
017044  
017045  
017046  
017047  
017048  
017049  
017050  
017051  
017052  
017053  
017054  
017055  
017056  
017057  
017058  
017059  
017060  
017061  
017062  
017063  
017064  
017065  
017066  
017067  
017068  
017069  
017070  
017071  
017072  
017073  
017074  
017075  
017076  
017077  
017078  
017079  
017080  
017081  
017082  
017083  
017084  
017085  
017086  
017087  
017088  
017089  
017090  
017091  
017092  
017093  
017094  
017095  
017096  
017097  
017098  
017099  
017100

001312  
016710  
016722  
016734  
162234  
000002  
001232  
000400 162204  
002000 162174  
001216 001214  
001200  
162170  
002  
002  
002  
002  
012737 017054 000024  
000000  
000777

GENERAL UTILITIES  
TYPE  
MERRPC  
CONVRT  
ERTABO  
TYPE  
PCALF  
MOVB  
TST  
BEQ  
TYPE  
ERRMSG:  
WRCO.FM:  
TST  
BEQ  
TYPE  
DAYHAD:  
TYPDAT:  
TST  
BEQ  
CONVRT  
RESO5  
TST  
BEQ  
PUSHO  
MOV  
HALT  
PCPRJ  
EXITER:  
CKSWR  
INC  
BIT  
BNE  
BIT  
BEQ  
MOV  
MOV  
JMP  
RTI  
ERTABO:  
1  
1  
XTSTN:  
1  
1  
TSTNO  
1  
PFAIL:  
MOV  
HALT  
BR

TYPE  
MERRPC  
CONVRT  
ERTABO  
TYPE  
PCALF  
MOVB  
TST  
BEQ  
TYPE  
ERRMSG:  
WRCO.FM:  
TST  
BEQ  
TYPE  
DAYHAD:  
TYPDAT:  
TST  
BEQ  
CONVRT  
RESO5  
TST  
BEQ  
PUSHO  
MOV  
HALT  
PCPRJ  
EXITER:  
CKSWR  
INC  
BIT  
BNE  
BIT  
BEQ  
MOV  
MOV  
JMP  
RTI  
ERTABO:  
1  
1  
XTSTN:  
1  
1  
TSTNO  
1  
PFAIL:  
MOV  
HALT  
BR

ERRMSG:  
WRCO.FM:  
ERRMSG  
WRCO.FM  
DAYHAD:  
TYPDAT:  
DAYHAD:  
TYPDAT:  
RESO5  
TST  
BEQ  
PUSHO  
MOV  
HALT  
PCPRJ  
EXITER:  
CKSWR  
INC  
BIT  
BNE  
BIT  
BEQ  
MOV  
MOV  
JMP  
RTI  
ERTABO:  
1  
1  
XTSTN:  
1  
1  
TSTNO  
1  
PFAIL:  
MOV  
HALT  
BR  
:SET UP FOR POWER UP TRAP  
:HALT ON POWER DOWN NORMAL  
:PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED

0-1,ERRFLG  
ERRMSG  
WRCO.FM  
DAYHAD  
TYPDAT  
DAYHAD  
TYPDAT  
RESO5  
RESO5  
TST  
BEQ  
PUSHO  
MOV  
HALT  
PCPRJ  
EXITER:  
CKSWR  
INC  
BIT  
BNE  
BIT  
BEQ  
MOV  
MOV  
JMP  
RTI  
ERTABO:  
1  
1  
XTSTN:  
1  
1  
TSTNO  
1  
PFAIL:  
MOV  
HALT  
BR  
:SET UP FOR POWER UP TRAP  
:HALT ON POWER DOWN NORMAL  
:PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED



3511	017306	020075	000
3512		017312	
3513	017312	020040	000077
3514	017316	000377	
3515	017320	050377	051127 043040
3516	017326	044501	042514 027104
3517	017334	051040	051505 040524
3518	017342	052122	040440 020124
3519	017350	042524	052123 000040
3520	017356	042524	042116 050040
3521	017364	051501	020123 055104
3522	017372	050504	020104 000
3523	017377	377	000122
3524	017402	050377	047522 051107
3525	017410	046501	044440 042116
3526	017416	041511	052101 051505
3527	017422	047040	020117 042504
3528	017432	044526	042502 020123
3529	017440	051120	051505 047105
3530	017446	027124	000
3531	017451	377	047111 052523
3532	017456	043106	041511 042511
3533	017464	052116	042040 052101
3534	017472	020501	000
3535	017475	377	042524 052123
3536	017502	050040	026503 000
3537	017507	377	047514 045503
3538	017514	047440	020116 042523
3539	017522	042514	052103 042105
3540	017530	052040	051505 000124
3541	017536	051503	035122 000040
3542	017544	042526	035103 000040
3543	017552	040520	051523 051505
3544	017560	020072	000
3545	017563	105	051122 051117
3546	017570	035123	000040
3547	017574	177777	042524 052123
3548	017602	047040	035117 000040
3549	017610	051777	052105 051440
3550	017616	044527	041524 020110
3551	017624	042522	020107 047524
3552	017632	042040	030521 023461
3553	017640	020123	042504 044523
3554	017646	042522	020104 041501
3555	017654	044524	042526 000056
3556	017662	041520	020072 000
3557	017667	377	040515 020120
3558	017674	043117	042040 030521
3559	017702	020061	052123 052101
3560	017710	051525	000377
3561			
3562	017714	000002	
3563	017716	006	003
3564	017720	001244	
3565	017722	006	002
3566	017724	001246	

```

.EVEN
MQM: .ASCIZ  ?/
MCRLF: .ASCIZ  <377>
MPFAIL: .ASCIZ  <377>/PWR FAILED. RESTART AT TEST /

MEPASS: .ASCIZ  <377> END PASS DZDQC

MR: .ASCIZ  <377>/R/
MERR2: .ASCIZ  <377>/PROGRAM INDICATES NO DEVICES PRESENT./

MERR3: .ASCIZ  <377>/INSUFFICIENT DATA!/

MTSTPC: .ASCIZ  <377>/TEST PC-/

MLOCK: .ASCIZ  <377>/LOCK ON SELECTED TEST/

MCSRX: .ASCIZ  /CSR: /
MVECX: .ASCIZ  /VEC: /
MPASSX: .ASCIZ  /PASSES: /

MERRX: .ASCIZ  /ERRORS: /

MTSTN: .ASCIZ  <377><377> /TEST NO: /

MNEW: .ASCIZ  <377>/SET SWITCH REG TO DQ11'S DESIRED ACTIVE./

MERRPC: .ASCIZ  /PC: /
XHEAD: .ASCIZ  <377>/MAP OF DQ11 STATUS/<377>

.EVEN
XSTATQ: 2
        .BYTE 6,3
        TEMP1
        .BYTE 6,2
        TEMP2

```

020000.P11 MACY11 27(732) 24-MAY-76 13:10 PAGE 70  
GENERAL UTILITIES (TYPE OUT, ERROR, SCOPE, ETC.)

017726  
017770  
020032  
020074  
020001

000000  
017770  
000000  
020032  
000000  
020074  
020001

.EVEN  
:BUFFERS FOR INPUT-OUTPUT  
INBUF: 0  
. = +40  
TEMP: 0  
. = +40  
MDATA: 0  
. = +40  
.END

ABBIT =	002000	DH6	013465	DSWR =	177570	MTITLE	001000	SAVRO	001256
ACTBIT =	004000	DH7	013505	DT0	014040	MTSTN	017574	SAVR1	001260
ADRCNT =	016137	DH8	013534	EM0	013010	MTSTPC	017475	SAVR2	001262
ALB	003054	DH9	013540	EM1	013026	MVECX	017544	SAVR3	001264
BBBIT =	010000	DISPRE	000174	EM3	013054	NEXT	001216	SAVR4	001266
BBBIT =	020000	DLIGHT =	177570	EM4	013116	NPRFLG	011766	SAVR5	001270
BINWRD	016456	DQACTV	001500	EM5	013150	NUMBER	014114	SAVSP	001272
BITSEL	014064	DQCR00	001400	EM6	013173	ODDBIT =	001000	SAVCS =	104406
BIT0 =	000001	DQCR01	001404	EM7	013206	OUT	017252	SCOPE =	:04400
BIT1 =	000002	DQCR02	001410	EM8	013254	PARAM =	104405	SCOPI =	104401
BIT10 =	002000	DQCR03	001414	ERRCNT	001232	PARAM1	015764	SECND	003220
BIT11 =	004000	DQCR04	001420	ERRFLG	001312	PARERR	016040	SEQ.	= 000014
BIT12 =	010000	DQCR05	001424	ERRMSG	016710	PARFLG	012602	SP	=%000006
BIT13 =	020000	DQCR06	001430	ERTAB0	017026	PARTI	016122	SPACNT =	016455
BIT14 =	040000	DQCR07	001434	EXITER	016760	PASCNT	001230	STACK =	001200
BIT15 =	100000	DQCR10	001440	EXTFLG	014052	PC	=%000007	STFLG	001311
BIT2 =	000004	DQCR11	001444	GENPAR	012444	PFTAB	017130	SVCS	016146
BIT3 =	000010	DQCR12	001450	HALTS	016740	POLY.	= 000017	SWR	001200
BIT4 =	000020	DQCR13	001454	HILIM	016132	POPPO =	012600	SWREG	000176
BITS	000040	DQCR14	001460	ICOUNT	001222	POP1SP =	005736	SWREGC	017260
BIT6 =	000100	DQCR15	001464	INBUF	017726	POP2SP =	022626	SW00 =	000001
BIT7 =	000200	DQCR16	001470	INIFLG	001310	PS	= 177776	SW01 =	000002
BIT8 =	000400	DQCR17	001474	INSTER =	104404	PJSHRO =	010046	SW02 =	000004
BIT9 =	001000	DQCSR	001506	INSTR =	104403	PUSHIS =	005746	SW03 =	000010
BRW	015422	DQERR	001366	INSTR2 =	015716	PUSH2S =	024646	SW04 =	000020
BRX	015424	DQNUM	001504	JUMBIT =	040000	RDSW	017136	SW05 =	000040
CHARDT =	000010	DQRCSH	001362	LIGHTS	001202	RESREG	016736	SW06 =	000100
CHAR1	001236	DQRCSCR	001360	LIMITS	016060	RESTAR	017054	SW08 =	000100
CHAR2	001240	DQREG	001370	LOBITS	016136	RESTAT	015250	SW09 =	001000
CHAR3	001242	DQRLVL	001352	LOCK	001220	RESOS =	104407	SW10 =	002000
CHRCNT	016454	DQRVEC	001350	LOGICA	015240	RETURN	001217	SW11 =	004000
CKSWR =	104414	DQSEC	001372	LOKFLG	001313	RUN	001304	SW12 =	010000
CKSYN1	003002	DQSECH	001374	LOLIM	016130	RUNCNT	001306	SW13 =	020000
CNTLU =	104415	DQSTAT	001510	LPCNT	001224	RUNFLG	001302	SW14 =	040000
CONVRT =	104411	DQST00	001402	LSTERR	001234	RXBA.P =	000000	SW15 =	100000
CONVRT =	104410	DQST01	001406	MCRLF	017316	RXBA.S =	000004	SYNBIT =	100000
COUNT	014062	DQST02	001412	MCSRX	017536	RXBUFF	014116	SYNC	014522
CREAM	001300	DQST03	001416	MDATA	020032	RXELNG	012302	SYNC.	= 000011
CSRMAP	000220	DQST04	001422	MDETCH	013724	RXLING	012132	SYNTST	005626
DATABP	016734	DQST05	001426	MEMCLR =	104413	RXSTRA	011522	TEMP	017770
DATAND	015722	DQST06	001432	MEPASS	017356	RXWC.P =	000001	TEMP1	001244
DELAY	014060	DQST07	001436	MERRPC	017662	RXWC.S =	000005	TEMP2	001246
DEVADR	016134	DQST10	001442	MERRX	017563	RX.BCC =	000015	TEMP3	001250
DH0	013306	DQST11	001446	MERR2	017402	R0	=%000000	TEMP4	001252
DH1	013326	DQST12	001452	MERR3	017451	R1	=%000001	TEMP5	001254
DH10	013546	DQST13	001456	MISC. =	000012	R2	=%000002	TKCSR	001204
DH11	013621	DQST14	001462	MLOCK	017507	R3	=%000003	TKDBR	001206
DH12	013645	DQST15	001466	MNEW	017610	R4	=%000004	TLAST =	007754
DH13	013675	DQST16	001472	MPASSX	017552	R5	=%000005	TMPBUF	014066
DH2	013347	DQST17	001476	MPFAIL	017320	SAVACT	001502	TPCSR	001210
DH3	013364	DQTCR	001364	MOM	017312	SAVEPC	014054	TPDBR	001212
DH4	013375	DQTLVL	001356	MR	017377	SAVNUM	001276	TRPOK	016472
DH5	013431	DQTVEC	001354	MSTCLR =	104412	SAVPC	001274	TSTNO	001226

DZDQD MACY11 27(732) 24-MAY-76 13:10 PAGE 73  
 DZDQDC.P11 SYMBOL TABLE

TST1	002254	TST35	005162	TXBA.S=	000006	XPASS	015300	.EOP	015126
TST10	003520	TST36	005210	TXBUFF	014524	XSTATQ	017714	.EPRTA	012604
TST11	003546	TST37	005236	TXSTRB	010720	XTSTN	017034	.HLT	016512
TST12	003574	TST4	003370	TXSTRC	011124	XVEC	015272	.INSTE	015672
TST13	003622	TST40	005264	TXSTRD	011266	XYZFLG	014112	.INSTG	015676
TST14	003650	TST41	005312	TXWC.P=	000003	\$CNTG	017266	.INSTF	015554
TST15	003676	TST42	005340	TXWC.S=	000007	\$E	= 000060	.INST1	015574
TST15	003724	TST43	005366	TX.BCC=	000016	\$MNEW	017301	.MEMCL	011770
TST16	003752	TST44	005414	TX.MUX=	000013	\$MSWR	017272	.MSG	015576
TST2	002644	TST45	005442	TYPDAT	016724	\$N	= 000056	.MSTCL	012114
TST20	004000	TST46	005546	TYPE =	104402	\$Y	= 000016	.PARAM	015724
TST21	004026	TST47	005576	TYPMSG	016646	.ADDR1	010702	.PFAIL	017042
TST22	004054	TST5	003416	VECMAP	000056	.ADDR2	010706	.RESOS	016200
TST23	004102	TST50	006236	WORD	014056	.ADDR3	010712	.SAVOS	016140
TST24	004436	TST51	005364	WRDCNT	016452	.ADDR4	010716	.SCOPE	015314
TST25	004570	TST52	006512	WRKO.F	016712	.BEGIN	002116	.SCOPI	015426
TST26	004730	TST53	006660	XBX	016540	.CHAR1	010700	.START	001512
TST27	004756	TST54	007032	XCHAR1	010630	.CHAR2	010704	.SYNC	014520
TST3	003030	TST55	007434	XCHAR2	010642	.CHAR3	010710	.TRPSR	016460
TST30	005004	TST56	007754	XCHAR3	010654	.CHAR4	010714	.TRPTA	001314
TST31	005032	TST6	003444	XCHAR4	010666	.CKSWR	017140	.TYPE	015446
TST32	005060	TST7	003472	XCSR	015264	.CNTLU	017214	.	= 020074

ERRORS DETECTED: 0  
 DEFAULT GLOBALS GENERATED: 0

\*TUKE: DZDQDC, DZDQDC/SOL+UNIV.LIB, DZDQDC.P11  
 RUN-TIME: 22 33 1 SECONDS  
 RUN-TIME RATIO: 147/57=2.5  
 CORE USED: 19K (37 PAGES)







- :



7.

1

