

DQ11

INTERRUPT TESTS
MD-11-DZDQH-B

EP-DZDQH-B-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN U.S.A.

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDQH-B-D
PRODUCT NAME: CHARACTER LENGTH AND INTERRUPT 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.

COPYRIGHT (C) 1975, 1976 BY DIGITAL EQUIPMEN CORPORATION

1. ABSTRACT

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

THIS TEST WILL CHECK TRANSMITTER AND RECEIVER INTERRUPTS, ENTER T AND EXIT T IF THE AB OPTION IS INSTALLED. IT ALSO CHECKS VRC AND DATA TRANSFERS. THE SECOND PART CHECKS TRANSMITTER AND RECEIVER CHARACTER LENGTHS.

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

THE SEVEN DIAGNOSTICS ARE:

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

THERE IS ALSO AN ONLINE TEST TO BE DISCUSSED LATER.

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

AND A PARAMETER INPUT PROGRAM IS AVAILABLE

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

2. REQUIREMENTS

2.1 EQUIPMENT

ANY PDP11 FAMILY CPU (WITH MINIMUM 4K MEMORY)-WITH
OR WITHOUT A HARDWARE SWITCH REGISTER (LOC. 177570)
ASR 33 (OR EQUIVALENT)

DQ11
SYNC MODEM (ONLY REQUIRED FOR ONLINE TEST)

2.2 STORAGE

PROGRAM WILL LOAD AND RUN
IN 4K OF MEMORY.
LOCATION 1400 THRU 1600 ARE ESPECIALLY TO
BE NOTED AND TO BE UNTOUCHED BY OPERATOR
AFTER DQ11 TRIAL PROGRAM HAS BEEN EXECUTED.
OR AFTER THE "AUTO SIZING" HAS BEEN DONE.

3. LOADING PROCEEDURE

3.1 METHOD

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

ABSOLUTE LOADER STARTING ADDRESS *500

MEMORY *
SIZE

4K	17
8K	27
12K	37
16K	47
20K	57
24K	67
28K	77
32K	87
36K	97
40K	107
44K	117
48K	127
52K	137
56K	147
60K	157

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

3.1.2 THEN START

4. STARTING PROCEEDURE

A. LOAD LOC. 200

B. SET SWR TO ZERO FOR "AUTO SIZING" OR LEAVE
LEAVE SWR BIT 7=1 TO USE EXISTING PARAMETERS SET UP
BY 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

C. THEN START

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

"MAP OF D011 STATUS"

1400	160010
1402	152300
1404	160020
1406	150310

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

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

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

4.1 CONTROL SWITCH SETTINGS

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

CONTROL:

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

- 1) TYPE CONTROL G (<↑G>); THIS WILL ALLOW THE TTY TO ENTER DATA INTO
LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.

NOTE: WHEN RUNNING THIS PROGRAM WITH A
SWITCH REGISTERLESS PROCESSOR IT MAY BE NECESSARY
TO DEPRESS <↑G> MORE THAN ONCE BEFORE IT IS ACCEPTED
DUE TO THE NUMBER OF RESET INSTRUCTIONS THAT ARE EXECUTED.

- 2) THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS
OF THE SOFTWARE SWITCH REGISTER.)
- 3) AFTER THE ''NEW=''
OF THE FOLLOWING AT THE TTY:
 - A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>.
(ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS
WILL BE ALLOWED)
IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH
REGISTER CONTENTS WILL NOT BE CHANGED.
 - B) IF A CONTROL U (<↑U>) IS DEPRESSED THEN THE PROGRAM WILL SEND YOU
BACK TO STEP 2.

SW 15 SET: HALT ON ERROR
SW 14 SET: LOOP ON CURRENT TEST
SW 13 SET: INHIBIT ERROR PRINT OUT

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

4.1.2 SWITCH REGISTER RESTRICTIONS

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

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

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

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

4.1.3 SWITCH REGISTER PRIORITYS

ERROR SWITCHES

1. SW 12 DELETE PRINT OUT/BELL ON ERROR.
2. SW 13 DELETE ERROR PRINTOUT.
3. SW 15 HALT ON THE ERROR.
4. SW 09 GOTO BEGINNING OF THE TEST.
5. SW 10 GOTO NEXT TEST ON ERROR.

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

SCOPE SWITCHES

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

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

4.2 STARTING ADDRESS

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

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

5. OPERATING PROCEDURE

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

AND PROGRAM WILL BEGIN RUNNING THE
DIAGNOSTIC

5.2 PROGRAM AND/OR OPERATOR ACTION

THE TYPICAL APPROACH SHOULD BE

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

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

AND THEN NOTE THE PC OF THE ERROR REPORT
THIS WAY THE EXACT FUNCTIONING OF THE TEST
CAN BE INTERPEDITED

6. ERRORS

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

6.2 ERROR RECOVERY

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

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

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

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

SEE SECTION 4. (PLEASE)

7.2 OPERATING RESTRICTIONS

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

- S. MISCELLANEOUS
- S.1 EXECUTION TIME
- S.2 PASS COMPLETE

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

END PASS DZDQH-B 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.

- S.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!!!!!!!

- S.4 KEY LOCATIONS

RETURN (1210) CONTAINS THE ADDRESS WHERE PROGRAM WILL
RETURN WHEN ITERATION COUNT IS REACHED
OR IF LOOP ON TEST IS ASSERTED.
NEXT (1212) CONTAINS THE ADDRESS OF THE NEXT TEST
TO BE PERFORMED.
TSTNO (1222) CONTAINS THE NUMBER OF THE TEST NOW
BEING PERFORMED.
RUN (1272) THE BIT IN "RUN" ALWAYS POINTS ONE
PAST THE DQ11 CURRENTLY BEING TESTED.
EXAMPLE:
(RUN) 1272/0000000001000000
MEANS THAT DQ11 NO.05 IS THE DQ11 NOW
RUNNING.

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

THESE LOCATIONS CONTAIN THE INFORMATION
NEEDED TO TEST UP TO 16 (DECIMAL) DQ11S
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 APPRIOATE DQSTXX: LOCATION.

8.5.3 "BB" OPTION INSTALLED?

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

8.5.4 "AB" OPTION INSTALLED?

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

8.5.5 "BA" OPTION INSTALLED?

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

8.5.6 JUMPER ON END OF CABLE?

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

8.5.7 ACTIVE ON FIRST NON-SYNC?

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

MO1

DZDQH MACY11 27(732) 24-SEP-76 10:11 PAGE 13
 DZDQH.B.P11 GENERAL DEFINATIONS AND EQUIVALENCIES

570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
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

;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	PUSHRD=10046	: SAVE RD ON STACK
012600	POPPO=12600	: RESTORE RD 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	BBBIT=20000
040000	JUMBIT=40000

626 001000 ODDBIT=1000
 627 100000 SYNBIT=100000
 628
 629
 630

:DQ11 SECONDARY REGISTER DEFINATIONS

631	000000	RXBA.P=0	:RECEIVER BUS ADDRESS PRIMARY.
632	000001	RXWC.P=1	:RECEIVER WORD COUNT PRIMARY.
633	000002	TXBA.P=2	:TRANSMITTER BUS ADDRESS PRIMARY.
634	000003	TXWC.P=3	:TRANSMITTER BUS ADDRESS PRIMARY.
635	000004	RXBA.S=4	:RECEIVER BUS ADDRESS SECONDARY.
636	000005	RXWC.S=5	:RECEIVER WORD COUNT SECONDARY.
637	000006	TXBA.S=6	:TRANSMITTER BUS ADDRESS SECONDARY.
638	000007	TXWC.S=7	:TRANSMITTER WORD COUNT SECONDARY.
639			
640			
641	000010	CHARDT=10	: CHARACTER DETECT REGISTER.
642	000011	SYNC.=11	: SYNC REGISTER.
643	000012	MISC.=12	: MISCELLANEOUS REGISTER.
644	000013	TX.MUX=13	: TRANSMITTER MUX REGISTER.
645	000014	SEQ.=14	: SEQUENCE REGISTER.
646	000015	RX.BCC=15	: RECEIVER BCC REGISTER.
647	000016	TX.BCC=16	: TRANSMITTER BCC REGISTER.
648	000017	POLY.=17	: POLYNOMIAL REGISTER.
649			
650			

707	000155	000156	.+2	:UNEXPECTED TRAP TO THIS LOCATION
708	000156	000000	HALT	:EXAMINE STACK TO FIND CAUSE
709	000160	000162	.+2	:UNEXPECTED TRAP TO THIS LOCATION
710	000162	000000	HALT	:EXAMINE STACK TO FIND CAUSE
711	000164	000166	.+2	:UNEXPECTED TRAP TO THIS LOCATION
712	000166	000000	HALT	:EXAMINE STACK TO FIND CAUSE
713	000170	000172	.+2	:UNEXPECTED TRAP TO THIS LOCATION
714	000172	000000	HALT	:EXAMINE STACK TO FIND CAUSE
715	000174	000176	.+2	:UNEXPECTED TRAP TO THIS LOCATION
716	000176	000000	HALT	:EXAMINE STACK TO FIND CAUSE
717	000200	000202	.+2	:UNEXPECTED TRAP TO THIS LOCATION
718	000202	000000	HALT	:EXAMINE STACK TO FIND CAUSE
719	000204	000206	.+2	:UNEXPECTED TRAP TO THIS LOCATION
720	000206	000000	HALT	:EXAMINE STACK TO FIND CAUSE
721	000210	000212	.+2	:UNEXPECTED TRAP TO THIS LOCATION
722	000212	000000	HALT	:EXAMINE STACK TO FIND CAUSE
723	000214	000216	.+2	:UNEXPECTED TRAP TO THIS LOCATION
724	000216	000000	HALT	:EXAMINE STACK TO FIND CAUSE
725	000220	000222	.+2	:UNEXPECTED TRAP TO THIS LOCATION
726	000222	000000	HALT	:EXAMINE STACK TO FIND CAUSE
727	000224	000226	.+2	:UNEXPECTED TRAP TO THIS LOCATION
728	000226	000000	HALT	:EXAMINE STACK TO FIND CAUSE
729	000230	000232	.+2	:UNEXPECTED TRAP TO THIS LOCATION
730	000232	000000	HALT	:EXAMINE STACK TO FIND CAUSE
731	000234	000236	.+2	:UNEXPECTED TRAP TO THIS LOCATION
732	000236	000000	HALT	:EXAMINE STACK TO FIND CAUSE
733	000240	000242	.+2	:UNEXPECTED TRAP TO THIS LOCATION
734	000242	000000	HALT	:EXAMINE STACK TO FIND CAUSE
735	000244	000246	.+2	:UNEXPECTED TRAP TO THIS LOCATION
736	000246	000000	HALT	:EXAMINE STACK TO FIND CAUSE
737	000250	000252	.+2	:UNEXPECTED TRAP TO THIS LOCATION
738	000252	000000	HALT	:EXAMINE STACK TO FIND CAUSE
739	000254	000256	.+2	:UNEXPECTED TRAP TO THIS LOCATION
740	000256	000000	HALT	:EXAMINE STACK TO FIND CAUSE
741	000260	000262	.+2	:UNEXPECTED TRAP TO THIS LOCATION
742	000262	000000	HALT	:EXAMINE STACK TO FIND CAUSE
743	000264	000266	.+2	:UNEXPECTED TRAP TO THIS LOCATION
744	000266	000000	HALT	:EXAMINE STACK TO FIND CAUSE
745	000270	000272	.+2	:UNEXPECTED TRAP TO THIS LOCATION
746	000272	000000	HALT	:EXAMINE STACK TO FIND CAUSE
747	000274	000276	.+2	:UNEXPECTED TRAP TO THIS LOCATION
748	000276	000000	HALT	:EXAMINE STACK TO FIND CAUSE
749	000300	000302	.+2	:UNEXPECTED TRAP TO THIS LOCATION
750	000302	000000	HALT	:EXAMINE STACK TO FIND CAUSE
751	000304	000306	.+2	:UNEXPECTED TRAP TO THIS LOCATION
752	000306	000000	HALT	:EXAMINE STACK TO FIND CAUSE
753	000310	000312	.+2	:UNEXPECTED TRAP TO THIS LOCATION
754	000312	000000	HALT	:EXAMINE STACK TO FIND CAUSE
755	000314	000316	.+2	:UNEXPECTED TRAP TO THIS LOCATION
756	000316	000000	HALT	:EXAMINE STACK TO FIND CAUSE
757	000320	000322	.+2	:UNEXPECTED TRAP TO THIS LOCATION
758	000322	000000	HALT	:EXAMINE STACK TO FIND CAUSE
759	000324	000326	.+2	:UNEXPECTED TRAP TO THIS LOCATION
760	000326	000000	HALT	:EXAMINE STACK TO FIND CAUSE
761	000330	000332	.+2	:UNEXPECTED TRAP TO THIS LOCATION
762	000332	000000	HALT	:EXAMINE STACK TO FIND CAUSE

763	000334	000336	.+2	:UNEXPECTED TRAP TO THIS LOCATION
764	000336	000000	HALT	:EXAMINE STACK TO FIND CAUSE
765	000340	000342	.+2	:UNEXPECTED TRAP TO THIS LOCATION
766	000342	000000	HALT	:EXAMINE STACK TO FIND CAUSE
767	000344	000346	.+2	:UNEXPECTED TRAP TO THIS LOCATION
768	000346	000000	HALT	:EXAMINE STACK TO FIND CAUSE
769	000350	000352	.+2	:UNEXPECTED TRAP TO THIS LOCATION
770	000352	000000	HALT	:EXAMINE STACK TO FIND CAUSE
771	000354	000356	.+2	:UNEXPECTED TRAP TO THIS LOCATION
772	000356	000000	HALT	:EXAMINE STACK TO FIND CAUSE
773	000360	000362	.+2	:UNEXPECTED TRAP TO THIS LOCATION
774	000362	000000	HALT	:EXAMINE STACK TO FIND CAUSE
775	000364	000366	.+2	:UNEXPECTED TRAP TO THIS LOCATION
776	000366	000000	HALT	:EXAMINE STACK TO FIND CAUSE
777	000370	000372	.+2	:UNEXPECTED TRAP TO THIS LOCATION
778	000372	000000	HALT	:EXAMINE STACK TO FIND CAUSE
779	000374	000376	.+2	:UNEXPECTED TRAP TO THIS LOCATION
780	000376	000000	HALT	:EXAMINE STACK TO FIND CAUSE
781	000400	000402	.+2	:UNEXPECTED TRAP TO THIS LOCATION
782	000402	000000	HALT	:EXAMINE STACK TO FIND CAUSE
783	000404	000406	.+2	:UNEXPECTED TRAP TO THIS LOCATION
784	000406	000000	HALT	:EXAMINE STACK TO FIND CAUSE
785	000410	000412	.+2	:UNEXPECTED TRAP TO THIS LOCATION
786	000412	000000	HALT	:EXAMINE STACK TO FIND CAUSE
787	000414	000416	.+2	:UNEXPECTED TRAP TO THIS LOCATION
788	000416	000000	HALT	:EXAMINE STACK TO FIND CAUSE
789	000420	000422	.+2	:UNEXPECTED TRAP TO THIS LOCATION
790	000422	000000	HALT	:EXAMINE STACK TO FIND CAUSE
791	000424	000426	.+2	:UNEXPECTED TRAP TO THIS LOCATION
792	000426	000000	HALT	:EXAMINE STACK TO FIND CAUSE
793	000430	000432	.+2	:UNEXPECTED TRAP TO THIS LOCATION
794	000432	000000	HALT	:EXAMINE STACK TO FIND CAUSE
795	000434	000436	.+2	:UNEXPECTED TRAP TO THIS LOCATION
796	000436	000000	HALT	:EXAMINE STACK TO FIND CAUSE
797	000440	000442	.+2	:UNEXPECTED TRAP TO THIS LOCATION
798	000442	000000	HALT	:EXAMINE STACK TO FIND CAUSE
799	000444	000446	.+2	:UNEXPECTED TRAP TO THIS LOCATION
800	000446	000000	HALT	:EXAMINE STACK TO FIND CAUSE
801	000450	000452	.+2	:UNEXPECTED TRAP TO THIS LOCATION
802	000452	000000	HALT	:EXAMINE STACK TO FIND CAUSE
803	000454	000456	.+2	:UNEXPECTED TRAP TO THIS LOCATION
804	000456	000000	HALT	:EXAMINE STACK TO FIND CAUSE
805	000460	000462	.+2	:UNEXPECTED TRAP TO THIS LOCATION
806	000462	000000	HALT	:EXAMINE STACK TO FIND CAUSE
807	000464	000466	.+2	:UNEXPECTED TRAP TO THIS LOCATION
808	000466	000000	HALT	:EXAMINE STACK TO FIND CAUSE
809	000470	000472	.+2	:UNEXPECTED TRAP TO THIS LOCATION
810	000472	000000	HALT	:EXAMINE STACK TO FIND CAUSE
811	000474	000476	.+2	:UNEXPECTED TRAP TO THIS LOCATION
812	000476	000000	HALT	:EXAMINE STACK TO FIND CAUSE
813	000500	000502	.+2	:UNEXPECTED TRAP TO THIS LOCATION
814	000502	000000	HALT	:EXAMINE STACK TO FIND CAUSE
815	000504	000506	.+2	:UNEXPECTED TRAP TO THIS LOCATION
816	000506	000000	HALT	:EXAMINE STACK TO FIND CAUSE
817	000510	000512	.+2	:UNEXPECTED TRAP TO THIS LOCATION
818	000512	000000	HALT	:EXAMINE STACK TO FIND CAUSE

000514	000516	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000516	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000520	000522	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000522	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000524	000526	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000526	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000530	000532	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000532	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000534	000536	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000536	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000540	000542	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000542	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000544	000546	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000546	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000548	000549	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000549	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000550	000552	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000552	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000554	000556	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000556	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000560	000562	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000562	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000564	000566	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000566	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000570	000572	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000572	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000574	000576	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000576	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000600	000602	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000602	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000604	000606	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000606	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000610	000612	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000612	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000614	000616	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000616	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000620	000622	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000622	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000624	000626	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000626	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000630	000632	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000632	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000634	000636	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000636	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000640	000642	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000642	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000644	000646	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000646	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000650	000652	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000652	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000654	000656	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000656	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000660	000662	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000662	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000664	000666	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000666	000000	HALT	:EXAMINE STACK TO FIND CAUSE
000670	000672	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000672	000000	HALT	:EXAMINE STACK TO FIND CAUSE

975	000674	000676	.+2	:UNEXPECTED TRAP TO THIS LOCATION
976	000676	000000	HALT	:EXAMINE STACK TO FIND CAUSE
977	000700	000702	.+2	:UNEXPECTED TRAP TO THIS LOCATION
978	000702	000000	HALT	:EXAMINE STACK TO FIND CAUSE
979	000704	000706	.+2	:UNEXPECTED TRAP TO THIS LOCATION
980	000706	000000	HALT	:EXAMINE STACK TO FIND CAUSE
981	000710	000712	.+2	:UNEXPECTED TRAP TO THIS LOCATION
982	000712	000000	HALT	:EXAMINE STACK TO FIND CAUSE
983	000714	000716	.+2	:UNEXPECTED TRAP TO THIS LOCATION
984	000716	000000	HALT	:EXAMINE STACK TO FIND CAUSE
985	000720	000722	.+2	:UNEXPECTED TRAP TO THIS LOCATION
986	000722	000000	HALT	:EXAMINE STACK TO FIND CAUSE
987	000724	000726	.+2	:UNEXPECTED TRAP TO THIS LOCATION
988	000726	000000	HALT	:EXAMINE STACK TO FIND CAUSE
989	000730	000732	.+2	:UNEXPECTED TRAP TO THIS LOCATION
990	000732	000000	HALT	:EXAMINE STACK TO FIND CAUSE
991	000734	000736	.+2	:UNEXPECTED TRAP TO THIS LOCATION
992	000736	000000	HALT	:EXAMINE STACK TO FIND CAUSE
993	000740	000742	.+2	:UNEXPECTED TRAP TO THIS LOCATION
994	000742	000000	HALT	:EXAMINE STACK TO FIND CAUSE
995	000744	000746	.+2	:UNEXPECTED TRAP TO THIS LOCATION
996	000746	000000	HALT	:EXAMINE STACK TO FIND CAUSE
997	000750	000752	.+2	:UNEXPECTED TRAP TO THIS LOCATION
998	000752	000000	HALT	:EXAMINE STACK TO FIND CAUSE
999	000754	000756	.+2	:UNEXPECTED TRAP TO THIS LOCATION
000	000756	000000	HALT	:EXAMINE STACK TO FIND CAUSE
001	000760	000762	.+2	:UNEXPECTED TRAP TO THIS LOCATION
002	000762	000000	HALT	:EXAMINE STACK TO FIND CAUSE
003	000764	000766	.+2	:UNEXPECTED TRAP TO THIS LOCATION
004	000766	000000	HALT	:EXAMINE STACK TO FIND CAUSE
005	000770	000772	.+2	:UNEXPECTED TRAP TO THIS LOCATION
006	000772	000000	HALT	:EXAMINE STACK TO FIND CAUSE
007	000774	000776	.+2	:UNEXPECTED TRAP TO THIS LOCATION
008	000776	000000	HALT	:EXAMINE STACK TO FIND CAUSE

909
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

```
000024 015756
000026 000340
000030 015436
000032 000340
000034 015404
000036 000340
000046 014164
000052 000052
000052 000000
000056 010120
000056 012721 000004
000060 022021
000064 020127 001000
000072 101771
000074 012737 000146 000020
000102 013737 001500 001244
000110 006037 001244
000114 103023
000116 005037 177776
000122 005722
000124 012772 000340 177776
000132 105200
000134 001376
000136 112712 000300
000142 005722
000144 000761
000146 051612
000150 042712 000007
000154 022626
000156 012716 000142
000162 000002
000164 000207
000174 000174
000174 000000
000176 000000
000200 000200
000200 000137 001512
```

```
; STANDARD INTERRUPT VECTORS
.=24
.PFAIL ; POWER FAIL HANDLER
340 ; SERVICE AT LEVEL 7
.HLT ; ERROR HANDLER
340 ; SERVICE AT LEVEL 7
.TRPSRV ; GENERAL HANDLER DISPATCH SERVICE
340 ; SERVICE AT LEVEL 7

.=46
LOGICAL ; ACT HOOKS

.=52
.WORD 0
; THIS ROUTINE TRIES TO FORCE THE RECEIVER TO INTERRUPT
; TO ITS VECTOR WHERE IT WILL PICK UP THE STATUS LOCATION
; FOR ITS NEW PC; AND PICK UP AN IOT INSTRUCTION FOR ITS
; NEW PS. WHEN THE NEW PC IS FETCHED AN IOT INSTRUCTION IS
; EXECUTED, TRAPPING TO LOCATION 20 WHERE A ROUTINE IS EXECUTED
; TO TAKE THE PC FROM THE STACK AND USE IT AS THE VECTOR ADDRESS
.=56

VECMAP:
1$: MOV R1,(R0)+ ; START FILLING THE VECTOR AREA
MOV #4,(R1)+ ; WITH .+2; IOT (4)
CMP (R0)+,(R1)+ ; UPDATE THE POINTERS
CMP R1,#1000 ; IS ALL FLOATING VECTOR AREA DONE
BLOS 1$ ; BR IF NOT ALL DONE
MOV #1$ ,Q#20 ; SET FOR IOT TRAP BY DQ11
MOV DQACTV,TEMP1 ; GET THE ACTIVE DQ11 S
2$: ROR TEMP1 ; ARE YOU ACTIVE.. DQ11
BCC 5$ ; IF CARRY CLEAR.. NO MORE DQ11S
CLR PS ; CLEAR PS
TST (R2)+ ; PUT POINTER TO STATUS TABLE
MOV #340,Q-2(R2) ; TRY AND SET PRI/SEC DONE AND IE
INCB RC ; DELAY.....
BNE .-2 ; .....DELAY
MOVB #300,(R2) ; NO INTERRUPT ASSUME 300 FIX IN TEST C
3$: TST (R2)+ ; UPDATE POINTERS
BR 2$ ; GO DO IT AGAIN
4$: BIS (SP),(R2) ; ENTERD BY IOT TRAP BY DQ11
BIC #7,(R2) ; CLEAR UNWANTED BITS
CMP (SP)+,(SP)+ ; POP IOT JUNK OFF STACK
MOV #3$,(SP) ; SET RETURN PC ON STACK
RTI ; GO HOME
5$: RTS PC ; ALL SIZING IS DONE

;****SOFTWARE SWITCH REGISTER****
.=174
DISPREG: 0 ; SOFTWARE DISPLAY REGISTER
SWREG: 0 ; SOFTWARE SWITCH REGISTER

;PROGRAM START
.=200
JMP .START ; GO TO START OF PROGRAM
```



```

1021 000540 012737 000006 000004      MOV      #6, D#4      ;RESET TIME OUT VECTOR
1022 000546 013737 001500 001502      MOV      DQACTV, SAVACT ;SAVE ACTIVE
1023 000554 012737 000340 000022      MOV      #340, D#22   ;SET IOT TRAP PRIO: TO 7
1024 000562 012702 001400          MOV      #1400, R2    ;SET TABLE POINTER
1025 000566 012700 000300          MOV      #300, R0     ;SET VECTOR START
1026 000572 012701 000302          MOV      #302, R1     ;SET VECTOR+2 START
1027 000576 000137 000056          JMP      VECMAP       ;GO FIND THE VECTORS
1028 000502 104402          4$:      TYPE        ;TYPE MESSAGE
1029 000604 016327          MERR2     ;I DIDN'T FIND ANY DQ115. DON'T USE AUTO SIZE.
1030 000606 005000          CLR      R0          ;
1031 000610 000000          HALT      ;HOW CAN I TEST NO DQ115
1032 000612 000776          BR       .-2        ;DON'T LET OPR HIT CONT. SW
1033 000614 012716 000466          5$:      MOV      #2$, (SP) ;ENTERED BY TIME OUT TRAP
1034 000620 000002          RTI          ;GO HOME.
1035
1036
1037
1038 001000 005377 040515 047111  .=1000      MTITLE: .ASCIZ <377><12>/MAINDEC-11-DZDQH-B/<377>/CHARACTER LENGTH AND INTERRUPT TESTS/
1039 001006 042504 026503 030461
1040 001014 042055 042132 044121
1041 001022 041055 041777 040510
1042 001030 040522 052103 051105
1043 001036 046040 047105 052107
1044 001044 020110 047101 020104
1045 001052 047111 042524 051122
1046 001060 050125 020124 042524
1047 001066 052123 177523      000
1048
1049
1050          001200      .=1200
1051          ;INDIRECT POINTERS
1052 001200 177570      SWR:      177570      ;SWITCH REGISTER POINTER
1053 001202 177570      LIGHTS:   177570     ;DISPLAY REGISTER POINTER
1054 001204 177560      TKCSR:   177560     ;TELETYPE KEYBOARD CONTROL REGISTER
1055 001206 177562      TKDBR:   177562     ;TELETYPE KEYBOARD DATA BUFFER
1056 001210 177564      TPCSR:   177564     ;TELEPRINTER CONTROL REGISTER
1057 001212 177566      TPDBR:   177566     ;TELEPRINTER DATA BUFFER
1058
1059          ;PROGRAM CONTROL PARAMETERS
1060
1061 001214 000000      RETURN:  0          ;SCOPE ADDRESS FOR LOOP ON TEST
1062 001216 000000      NEXT:    0          ;ADDRESS OF NEXT TEST TO BE EXECUTED
1063 001220 000000      LOCK:    0          ;ADDRESS FOR LOCK ON CURRENT DATA
1064 001222 000003      ICOUNT:  3          ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
1065 001224 000000      LPCNT:   0          ;NUMBER OF ITERATIONS COMPLETED
1066 001226 000000      TSTNO:   0          ;NUMBER OF TEST IN PROGRESS
1067 001230 000000      PASCNT:  0          ;NUMBER OF PASSES COMPLETED
1068 001232 000000      ERRCNT:  0          ;TOTAL NUMBER OF ERRORS
1069 001234 000000      LSTERR:  0          ;PC OF LAST ERROR CALL
1070
1071          ;PROGRAM VARIABLES
1072
1073 001236 000000      CHAR1:   0
1074 001240 000000      CHAR2:   0
1075 001242 000000      CHAR3:   0
1076 001244 000000      TEMP1:   0          ;TEMPORARY STORAGE

```


DZDQH MACY11 27(732) 24-SEP-76 10:11 PAGE 23
DZDQHB.P11 PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

1077	001246	000000	TEMP2:	0	: TEMPORARY STORAGE
1078	001250	000000	TEMP3:	00	: TEMPORARY STORAGE
1079	001252	000000	TEMP4:	00	: TEMPORARY STORAGE
1080	001254	000000	TEMP5:	00	: TEMPORARY STORAGE
1081	001256	000000	SAVR0:	00	: R0 STORAGE
1082	001260	000000	SAVR1:	00	: R1 STORAGE
1083	001262	000000	SAVR2:	00	: R2 STORAGE
1084	001264	000000	SAVR3:	00	: R3 STORAGE
1085	001266	000000	SAVR4:	00	: R4 STORAGE
1086	001270	000000	SAVR5:	00	: R5 STORAGE
1087	001272	000000	SAVSP:	00	: STACK POINTER STORAGE
1088	001274	000000	SAVPC:	00	: PROGRAM COUNTER STORAGE
1089	001276	000000	SAVNUM:	0	
1090	001300	000001	CREAM:	.BLKW 1	
1091	001302	000000	RUNFLG:	00	
1092	001304	000000	RUN:	00	
1093	001306	000000	RUNCNT:	0	

DZDQH MACY11 27(732) 24-SEP-76 10:11 PAGE 25
 DZDQHB.P11 PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```

1150 001364 000000 DQTCRS: 0 ; POINTER TO DQ11 TRANSMITTER CONTROL REGISTER
1151 001366 000000 DQERR: 0 ; POINTER TO DQ11 ERROR REGISTER
1152 001370 000000 DQREG: 0 ; POINTER TO HIGH BYTE OF ERROR REGISTER
1153 001372 000000 DQSEC: 0 ; POINTER TO DQ11 SECONDARY REGISTER
1154 001374 000000 DQSECH: 0 ; POINTER TO HIGH BYTE OF DQ11 SECONDARY REGISTER
1155
1156
1157
1158 ;DQ11 STATUS TABLE AND ADDRESS ASSIGNMENTS
1159
1160 001400 001400 . =1400
1161 001400 000001 DQCR00: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 00
1162 001402 000001 DQST00: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 00
1163 001404 000001 DQCR01: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 01
1164 001406 000001 DQST01: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 01
1165 001410 000001 DQCR02: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 02
1166 001412 000001 DQST02: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 02
1167 001414 000001 DQCR03: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 03
1168 001416 000001 DQST03: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 03
1169 001420 000001 DQCR04: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 04
1170 001422 000001 DQST04: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 04
1171 001424 000001 DQCR05: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 05
1172 001426 000001 DQST05: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 05
1173 001430 000001 DQCR06: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 06
1174 001432 000001 DQST06: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 06
1175 001434 000001 DQCR07: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 07
1176 001436 000001 DQST07: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 07
1177 001440 000001 DQCR10: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 10
1178 001442 000001 DQST10: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 10
1179 001444 000001 DQCR11: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 11
1180 001446 000001 DQST11: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 11
1181 001450 000001 DQCR12: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 12
1182 001452 000001 DQST12: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 12
1183 001454 000001 DQCR13: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 13
1184 001456 000001 DQST13: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 13
1185 001460 000001 DQCR14: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 14
1186 001462 000001 DQST14: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 14
1187 001464 000001 DQCR15: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 15
1188 001466 000001 DQST15: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 15
1189 001470 000001 DQCR16: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 16
1190 001472 000001 DQST16: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 16
1191 001474 000001 DQCR17: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 17
1192 001476 000001 DQST17: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 17
1193 001500 000001 DQACTV: .BLKW 1 ; HOLD ACTIVE BITS FOR TESTING
1194 001502 000001 SAVACT: .BLKW 1 ; SAVE NUMBER OF ACTIVE DQ11S
1195 001504 000001 DQNUM: .BLKW 1 ; OCTAL NUMBER OF TOTAL NUMBER OF DQ11S
1196 001506 000001 DQCSR: .BLKW 1 ; CSR OF DQ11 UNDER TEST
1197 001510 000001 DQSTAT: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS OF DQ11 UNDER TEST
1198
1199 ;PROGRAM INITIALIZATION
1200 ;LOCK OUT INTERRUPTS
1201 ;SET UP PROCESSOR STACK
1202 ;SET UP POWER FAIL VECTOR
1203 ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1204 ;TYPE TITLE MESSAGE
1205

```

M02

DZDQH MACY11 27(732) 24-SEP-76 10:11 PAGE 26
 DZDQHB.P11 PROGRAM INITIALIZATION AND START UP.

1206	001512	012737	000340	177776	.START:	MOV	#340,PS	;LOCK OUT INTERRUPTS
1207	001520	012706	001200			MOV	#STACK,SP	;SET UP STACK
1208	001524	012737	015766	000024		MOV	#.PFAIL,@#24	;SET UP POWER FAIL VECTOR
1209	001532	013737	001504	001276		MOV	DQNUM,SAVNUM	
1210	001540	105037	001311			CLRB	STFLG	;CLEAR START FLAG
1211	001544	005037	001230			CLR	PASCNT	;CLEAR PASS COUNT
1212	001550	105037	001312			CLRB	ERRFLG	;CLEAR ERROR FLAG
1213	001554	005037	001302			CLR	RUNFLG	
1214	001560	012737	001400	001300		MOV	#1400,CREAM	
1215	001566	005037	001232			CLR	ERRCNT	;CLEAR ERROR COUNT
1216	001572	005037	001234			CLR	LSTERR	;CLEAR LAST ERROR POINTER
1217	001576	012737	000001	001226		MOV	#1,TSTNO	;SET UP FOR TEST 1
1218	001604	012737	001512	001214		MOV	#.START,RETURN	;SET UP FOR POWER FAIL BEFORE
1219								;TESTING STARTS
1220	001612	105737	001310			TSTB	INIFLG	;HAS INITIALIZATION BEEN PERFORMED
1221	001616	001075				BNE	12\$	
1222	001620	104402	001000			TYPE	.MTITLE	;TYPE TITLE MESSAGE
1223	001624	105137	001310			COMB	INIFLG	;IF NOT SET FLAG AND DO
1224								
1225	001630	012737	177570	001200		MOV	#DSWR,SWR	;MOV HARDWARE SWR TO SWR
1226	001636	012737	177570	001202		MOV	#DLIGHTS,LIGHTS	;MOV DISPLAY LIGHTS TO LIGHTS
1227	001644	013746	000006			MOV	@#6,-(SP)	;SAVE VECTORS
1228	001650	013746	000004			MOV	@#4,-(SP)	
1229	001654	012737	001674	000004		MOV	#64\$,@#4	;SET UP FOR TIMEOUT
1230	001662	022777	177777	177310		CMP	#-1,@SWR	;REFERENCE HARDWARE SWITCH REGISTER
1231	001670	001402				BEQ	65\$	
1232	001672	000407				BR	65\$	
1233	001674	022626			64\$:	CMP	(SP)+,(SP)+	;ADJUST STACK
1234	001676	012737	000176	001200	65\$:	MOV	#SWREG,SWR	;POINT TO SOFTWARE SWITCH REG
1235	001704	012737	000174	001202		MOV	#DISPREG,LIGHTS	;POINT TO SOFT DISPLAY REG
1236	001712	012637	000004		66\$:	MOV	(SP)+,@#4	;RESTORE VECTORS
1237	001716	012637	000006			MOV	(SP)+,@#6	
1238	001722	005737	000042			TST	@#42	;UNDER MONITOR
1239	001726	001005				BNE	67\$	
1240	001730	022737	000176	001200		CMP	#SWREG,SWR	;IS SWREG USED
1241	001736	001001				BNE	67\$	
1242	001740	104415				CNTLU		
1243	001742	105777	177232		67\$:	TSTB	@SWR	
1244	001746	100402				BMI	.+6	
1245	001750	004737	000220			JSR	PC,CSRMAP	
1246	001754	104402	016614			TYPE	.XHEAD	
1247	001760	012737	001400	001244		MOV	#1400,TEMP1	
1248	001766	017737	177252	001246		MOV	@TEMP1,TEMP2	
1249	001774	001406				BEQ	.+16	
1250	001776	104410				CONVRT		
1251	002000	016642				XSTATQ		
1252	002002	062737	000002	001244		ADD	#2,TEMP1	
1253	002010	000766				BR	.-22	
1254	002012	032777	000001	177160	12\$:	BIT	#SW00,@SWR	
1255	002020	001424				BEQ	1\$	
1256	002022	104402				TYPE		
1257	002024	016535				MNEW		
1258	002026	005000				CLR	RO	
1259	002030	000000				HALT		
1260	002032	104414				CKSWR		
1261	002034	027737	177140	001502		CMP	@SWR,SAVACT	


```

1262 002042 101404          BLOS      11$
1263 002044 104402          TYPE
1264 002046 016376          MERR3
1265 002050 000000          HALT
1266 002052 000776          BR      -2
1267 002054 017737 177120 001500 11$:  MOV      @SWR,DQACTV
1268 002062 013700 001500          MOV      DQACTV,RO
1269 002066 000000          HALT
1270 002070 104414          CKSWR
1271 002072 012700 000300 1$:  MOV      #300,RO
1272 002076 012701 000302          MOV      #302,R1
1273 002102 010120          2$:  MOV      R1,(R0)+
1274 002104 005021          CLR      (R1)+
1275 002106 022021          CMP      (R0)+,(R1)+
1276 002110 022700 001000          CMP      #1000,RO
1277 002114 001372          BNE      2$
1278
1279
1280
1281 002116 012737 000340 177776 .BEGIN: MOV      #340,PS          ;LOCK OUT INTERRUPTS
1282 002124 012706 001200          MOV      #STACK,SP      ;SET UP STACK
1283 002130 005737 000042          TST      @#42          ;IS PROGRAM UNDER MONITOR CONTROL
1284 002134 001040          BNE      3$
1285 002136 104414          CKSWR          ;CHECK FOR <↑G>
1286 002140 032777 000004 177032          BIT      #BIT2,@SWR      ;CHECK FOR LOCK ON TEST
1287 002146 001411          BEQ      1$
1288 002150 104402 016434          TYPE      .MLOCK
1289 002154 012737 000240 014250          MOV      #NOP,TTST
1290 002162 012737 000240 014252          MOV      #NOP,TTST+2      ;SET UP TO LOCK
1291 002170 000406          BR      2$
1292 002172 013737 014346 014250 1$:  MOV      BRW,TTST
1293 002200 013737 014350 014252          MOV      BRX,TTST+2      ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1294 002206 032777 000002 176764 2$:  BIT      #SW01,@SWR      ;IF SW01=1, GET STARTING PC
1295 002214 001410          BEQ      3$
1296 002216 104403          INSTR
1297 002220 016422          MTSTPC
1298 002222 104405          PARAM
1299 002224 002254          TST1
1300 002226 011622          TLAST
1301 002230 000207          RETURN
1302 002232 001          .BYTE 1
1303 002233 001          .BYTE 1
1304 002234 000403          BR      4$
1305 002236 012737 002254 001214 3$:  MOV      #TST1,RETURN      ;START AT TEST 1
1306 002244 104402 016324          4$:  TYPE      MR          ;TYPE R
1307 002250 000177 176740          JMP      @RETURN          ;START TESTING
1308
1309
1310 002254 012737 000001 001226 ; TEST 1
1311 002262 012737 002644 001214 *****
1312 002270 012737 002644 001216 TST1: MOV      #1,TSTNO
1313 002276 105737 001302          MOV      #TST2,RETURN
1314 002302 001010          MOV      #TST2,NEXT
1315 002304 012737 000001 001304          TSTB      RUNFLG          ;IS THIS MY FIRST TIME HERE?
1316 002312 012737 000020 001306          BNE      1$          ;BR IF FLAG IS SET
1317 002320 105137 001302          MOV      #BIT0,RUN          ;SET RUN POINTER.
1317 002320 105137 001302          MOV      #16,RUNCNT          ;SET FOR MAX OF 16 DQ11'S PER SYSTEM
1317 002320 105137 001302          COMB      RUNFLG          ;SET RUN FLAG

```



```

: TEST 2
:*****
14101 0030674 012737 000002 001226 *TST2: MOV #2,TSTNO
14102 0030674 012737 002732 001214 MOV #2$,RETURN
14103 0030674 012737 000036 001222 MOV #30,ICOUNT
14104 0030674 012737 003754 001216 MOV #TST3,NEXT
:ADJUST SYNC CHARACTERS.

14105 0030674 032737 100000 001510 BIT #SYNBIT,DOSTAT :ONE SYNC CHAR OR TWO?
14106 0030674 001005 BNE 1$ :BR IF TWO
14107 0030674 105037 012616 CLRB SYNC :SET ONE SYNC
14108 0030674 005037 013422 CLR XSYNC :DBL SYNC SET TO ONE.
14109 0030674 000406 BR 2$ :CONT.
14110 0030674 112737 000026 012616 1$: MOVB #2$,SYNC :LOAD FOR TWO SYNC
14111 0030674 012737 013026 013422 MOV #13026,XSYNC :SAME FOR DBL SYNC
14112 0030674 104413 2$: MEMCLR :CLEAR ALL REGISTERS GIVE MSTCLR
14113 0030674 005037 014046 CLR GDCHAR :ZERO POINTER
14114 0030674 005037 014040 CLR CHAR
14115 0030674 005037 177776 CLR PS :ZERO PROC. PRIO.
14116 0030674 105077 176414 SETON: CLRB @DQREG :SEL THE RX BA PRI.
14117 0030674 012777 013022 176410 MOV #RXBUFF,@DQSEC :LOAD RX BA PRI.
14118 0030674 105277 176402 INCB @DQREG :SEL RX WC PRI.
14119 0030674 012777 177600 176376 MOV #-200,@DQSEC :SET FOR 200 (8) CHARS
14120 0030674 105277 176370 INCB @DQREG :SEL THE TX BA PRI.
14121 0030674 012777 012616 176364 MOV #SYNC,@DQSEC :LOAD WITH SYNC POINTER
14122 0030674 105277 176356 INCB @DQREG :SEL THE TX WC PRI.
14123 0030674 012777 177576 176352 MOV #-202,@DQSEC :SET FOR 2 SYNC AND 200 (8) CHARS.
14124 0030674 105277 176344 INCB @DQREG :SEL THE RX BA SEC
14125 0030674 012777 013630 176340 MOV #XRXBUF,@DQSEC :LOAD RX BA SEC
14126 0030674 105277 176332 INCB @DQREG :SEL RX WC SEC
14127 0030674 012777 177600 176326 MOV #-200,@DQSEC :SET FOR 200(8) CHARS
14128 0030674 105277 176320 INCB @DQREG :SEL THE TX BA SEC
14129 0030674 012777 013426 176314 MOV #XTXBUF,@DQSEC :LOAD IT
14130 0030674 105277 176306 INCB @DQREG :SEL THE TX WC SEC
14131 0030674 012777 177600 176302 MOV #-200,@DQSEC :SET FOR 200 CHARS
14132 0030674 112777 000011 176272 MOVB #11,@DQREG :SEL THE SYNC REGISTER
14133 0030674 013777 012614 176266 MOV #SYNC,@DQSEC :LOAD SYNC
14134 0030674 105277 176260 INCB @DQREG :SEL THE MISC REGISTER
14135 0030674 012777 104000 176254 MOV #104000,@DQSEC :SET 8 BITS PER CHAR AND VRC ENABLE.
14136 0030674 032737 040000 001510 BIT #JUMBIT,DOSTAT :IS JUMPER AT END OF CABLE?
14137 0030674 001003 BNE +10 :BR IF YES
14138 0030674 052777 000010 176236 BIS #BIT3,@DQSEC :NO CABLE SET TEST LOOP FOR DATA TURN AROUND
14139 0030674 112777 000017 176226 MOVB #17,@DQREG :SEL THE POLY REGISTER
14140 0030674 012777 123456 176222 MOV #123456,@DQSEC :SET PLOYNOMIAL.

14141 0030674 012700 012620 MOV #TXBFA,R0 :START TO FILL TX BUFFERS
14142 0030674 012703 000177 MOV #177,R3 :COUNTER
14143 0030674 110320 1$: MOVB R3,(R0)+ :PRIMARY IS BINARY COUNT BACKWARDS.
14144 0030674 105203 DECB R3 :DONE?
14145 0030674 001376 BNE 1$ :NO
14146 0030674 012700 013426 MOV #XTXBUF,R0 :SET SEC BUFFER
14147 0030674 005003 CLR R3 :
14148 0030674 110320 2$: MOVB R3,(R0)+ :SECONDARY IS BINARY COUNT
14149 0030674 105203 INCB R3 :DONE?
14150 0030674 100375 BPL 2$ :NO
14151 0030674 012777 003500 176140 MOV #RXISR,@DQREVC :SET RECEIVER INTERRUPT POINTER

```



```

1473 003210 012777 000240 176134 MOV #240,DDQRLVL :SET PRIO: TO 5
1474 003216 012777 003332 176130 MOV #TXISR,DDQTEC :SET TX VECTOR
1475 003224 012777 000240 176124 MOV #240,DDQTLVL :SET PRIO TO 5
1476 003232 012777 000041 176120 MOV #BITS+BIT0,DDQRCSR :SET RX GO AND IE
1477 003240 012777 000051 176116 MOV #BITS+BIT3+BIT0,DDQTCSR :SET TX GO AND IE AND ERROR IE
1478 003246 005037 001246 CLR TEMP2 :SET TIMER
1479 003252 012737 000113 001250 MOV #75,TEMP3 :SET NUMBER OF INTERRUPTS WANTED
1480 003260 012737 000020 001252 48: MOV #16,TEMP4 :SET FOR 16 REGISTERS
1481 003266 142777 000017 176074 BICB #17,DDQREG :SEL RX BA PRI.
1482 003274 105777 176070 38: TSTB DDQREG :SIT HERE AND MAKE WAVES
1483 003300 005777 176066 TST DDQSEC :WHILE INTERRUPTS OCCUR
1484 003304 105277 176060 INCB DDQREG *****
1485 003310 005037 001252 DEC TEMP4 *****
1486 003314 001067 38: BNE 38: :SAME
1487 003316 005237 001246 INC TEMP2 :UPDATE COUNTER
1488 003320 001066 48: BNE 48: :KEEP GOING
1489 003324 104005 HLT 5 :RX FAILED TO CONTINUSLY INTERUPT
1490 003326 000754 BR 48: :****STRONGLY SUGGEST SW08=1 (GOTO TOP OF TEST OF ERROR)
1491 003330 104400 ENDS2: SCOPE :SCOPE THIS TEST.....

1492 003332 017737 176030 014032 TXISR: MOV DDQERR,ERR :ANY ERRORS
1493 003340 100001 BPL +4 :BR IF NO
1494 003342 104004 HLT 4 :DQ11 ERROR FLAG IS SET.
1495 003344 032777 000004 176012 BIT #BIT2,DDQTCSR :****STRONGLY SUGGEST SW08=1 (GOTO TOP OF TEST OF ERROR)
1496 003352 001425 BEQ 18: :WHO SHOULD I SERVICE PRI OR SEC?
1497 003354 112777 000002 176006 MOVB #2,DDQREG :BR IF SEC NEEDS SERVICE
1498 003362 042777 000200 175774 BIC #BIT7,DDQTCSR :SEL TX BA PRI
1499 003370 012777 012620 175774 MOV #TXBFA,DDQSEC :CLEAR TX PRI DONE.
1500 003376 105277 175766 INCB DDQREG :LOAD THE TX BA PRI
1501 003402 152777 000120 175760 BICB #BITS+BIT4,DDQREG :SEL THE TX WC PRI.
1502 003410 012777 177600 175754 MOV #-200,DDQSEC :SET WRITE EN. AND ENTER T
1503 003416 142777 000017 175744 BICB #17,DDQREG :LOAD TX WC PRI.
1504 003424 000002 RTI :CLEAR REG POINTER.
1505 003426 042777 000100 175730 18: BIC #BIT6,DDQTCSR :EXIT STAGE RIGHT
1506 003434 112777 000006 175726 MOVB #6,DDQREG :CLEAR TX SEC DONE
1507 003442 012777 013426 175722 MOV #TXBUF,DDQSEC :SEL THE TX BA PRI.
1508 003450 105277 175714 INCB DDQREG :LOAD THE TX BA SEC
1509 003454 152777 000060 175706 BICB #BITS+BIT4,DDQREG :SEL THE TX WC SEC
1510 003462 012777 177600 175702 MOV #-200,DDQSEC :SET WRITE EN. AND EXIT T
1511 003470 142777 000017 175672 28: BICB #17,DDQREG :LOAD THE TX WC SEC
1512 003476 000002 RTI :CLEAR REG POINTER.
1513 003500 RXISR: :EXIT STAGE LEFT.

1514 003500 005037 001246 CLR TEMP2 :LET TIMER KNOW THAT RX INTERRUPTED
1515 003504 017737 175656 014032 MOV DDQERR,ERR :ANY ERRORS
1516 003512 100001 BPL +4 :BR IF NO
1517 003514 104004 HLT 4 :DQ11 ERROR FLAG SET!!!!
1518 003516 032777 000004 175634 BIT #BIT2,DDQRCSR :****STRONGLY SUGGEST SW08=1 (GOTO TOP OF TEST OF ERROR)
1519 003524 001426 BEQ 28: :WHO SERVICE PRI OR SEC
1520 003526 042777 000200 175624 BIC #BIT7,DDQRCSR :BR IF SEC NEEDS SERVICE
1521 003534 105077 175630 CLRB DDQREG :CLEAR RX PRI. DONE
1522 003540 012777 013022 175624 MOV #RXBUFF,DDQSEC :SEL RX BA PRI.
:LOAD IT

```


1496	003546	105277	175616		INCB	3DQREG	:SEL THE RX WC PRI.
1497	003556	152777	000120	175610	BISB	#BIT6+BIT4,3DQREG	:SET WRITE EN. AND ENTER T
1498	003560	012777	177600	175604	MOV	#-200,3DQSEC	:LOAD RX WC SEC
1499	003566	012701	012620		MOV	#TXBFA,R1	:PREPARE TO CHECK DATA. SET TX POINTER
1500	003572	012702	013022		MOV	#RXBUF,R2	:SET RX POINTER
1501	003576	000137	003654		JMP	3\$:GO AND CHECK DATA
1502	003602	042777	000100	175550	SIC	#BIT6,3DQRCR	:CLEAR RX SEC DONE
1503	003610	112777	000004	175552	MOVB	#4,3DQREG	:SEL RX BA SEC
1504	003616	012777	013630	175546	MOV	#RXBUF,3DQSEC	:LOAD IT
1505	003624	105277	175540		INCB	3DQREG	:SEL THE RX WC SEC
1506	003630	152777	000060	175532	BISB	#BIT5+BIT4,3DQREG	:SET WRITE EN. AND EXIT T
1507	003636	012777	177600	175526	MOV	#-200,3DQSEC	:WRITE RX WC SEC
1508	003644	012701	013426		MOV	#TXBUF,R1	:GET TX BUFFER POINTER
1509	003650	012702	013630		MOV	#RXBUF,R2	:GET RX POINTER
1510	003654	012700	000200		MOV	#200,R0	:GET NUMBER OF CHARS
1511	003660	142711	000200		BICB	#BIT7,(R1)	:CLEAR VRC
1512	003664	142712	000200		BICB	#BIT7,(R2)	:CLEAR VRC
1513	003670	122122			CMPB	(R1)+,(R2)+	:DATA OK?
1514	003672	001414			BEQ	7\$:BR IF YES
1515	003674	112777	000012	175466	MOVB	#12,3DQREG	:SEL MISC REG
1516	003702	052777	000002	175462	BIS	#BIT1,3DQSEC	:STOP THE DQ11 CLOCK.
1517	003710	114137	014046		MOVB	-(R1),GDCHAR	:STORE GOOD CHAR
1518	003714	114237	014040		MOVB	-(R2),CHAR	:STORE BAD CHAR.
1519	003720	104003			HLT	3	:DATA COMPARE ERROR
1520							:*****STRONGLY SUGGEST SW08=1 (GOTO TOP OF TEST OF ERROR
1521	003722	122122			CMPB	(R1)+,(R2)+	:POP POINTERS
1522	003724	005300			DEC	R0	:ALL DATA CHECKED?
1523	003726	001354			BNE	4\$:BR IF NO
1524	003730	005337	001250		DEC	TEMP3	:ALL INTERRUPTS DONE?
1525	003734	001003			BNE	6\$:NO KEEP INTERRUPTING
1526	003736	000005			RESET		:STOP THE SHOW CLEAR THE WORLD
1527	003740	012716	003330		MOV	#ENDTS2,(SP)	:SET FOR END TEST RETURN
1528	003744	142777	000017	175416	BICB	#17,3DQREG	:CLEAR REG POINTER
1529	003752	000002			RTI		:EXIT STAGE MIDDLE

: TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
 : THIS TEST WILL XMIT AND RECV CHARACTERS
 : AT 2 BITS/PER/CHAR.
 : DATA CHECKING WILL BE PERFORMED!

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
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576

```

003754 012737 000003 001226
003762 012737 004314 001216
003770 104413
003772 012700 013022
003776 005001
004000 005020
004002 105201
004004 100375
004006 112777 000011 175354
004014 013737 012616 001246
004022 012737 177774 012134
004030 143737 012134 001246
004036 000241
004040 106037 001246
004044 143737 012134 001247
004052 000241
004054 106037 001247
004060 013737 001246 012136
004066 013737 001246 012140
004074 013777 001246 175270
004102 105277 175262
004106 012777 000010 175256
004114 012700 000016
004120 000300
004122 050077 175244
004126 052777 000002 175236
004134 042777 000002 175230
004142 105077 175222
004146 012777 013022 175216
004154 105277 175210
004160 012777 177734 175204
004166 105277 175176
004172 012777 012140 175172
004200 105277 175164
004204 012777 177732 175160
004212 005277 175142
004216 005277 175142
004222 005005
004224 105777 175130
004230 100404
004232 062705 000001
004236 001372
004240 104001
004242 012700 012142
004246 012701 013022
004252 012702 000044
004256
  
```

```

: TEST 3
:*****
TST3: MOV #3,TSTNO
      MOV #TST4,NEXT
      MEMCLR
      MOV #RXBUFF,RO
      CLR R1
      CLR (RO)+
      INCB R1
      BPL SS
      MOVB #11,DDQREG
      MOV SYNC,TEMP2
      MOV #177774,MASK
      BICB MASK,TEMP2
      CLC
      RORB TEMP2
      BICB MASK,TEMP2+1
      CLC
      RORB TEMP2+1
      MOV TEMP2,SYNC1
      MOV TEMP2,SYNC2
      MOV TEMP2,DDQSEC
      INCB DDQREG
      MOV #BIT3,DDQSEC
      MOV #16,RO
      SWAB RO
      BIS RO,DDQSEC
      BIS #BIT1,DDQSEC
      BIC #BIT1,DDQSEC
      CLRB DDQREG
      MOV #RXBUFF,DDQSEC
      INCB DDQREG
      MOV #-36,DDQSEC
      INCB DDQREG
      MOV #SYNC2,DDQSEC
      INCB DDQREG
      MOV #-38,DDQSEC
      INC DDQRCR
      INC DDQTCR
      CLR R5
      TSTB DDQRCR
      BMI Z$
      ADD #1,R5
      BNE 1$
      HLT 1
      MOV #TXBUFF,RO
      MOV #RXBUFF,R1
      MOV #36.,R2
      SS:
      1$:
      2$:
      3$:
      : CLEAR ALL THE DD11
      : LOAD THE BUFFER POINTER
      : SET UP TO CLEAR THE BUFFER
      : CLEAR IT
      : DONE?
      : BRANCH IF NO
      : SELECT THE SYNC REG
      : LOAD SYNC
      : LOAD THE MASK
      : SET UP A MASK TO GET THE
      : CORRECT SYNC CHARACTER
      : FOR THIS CHARACTER LENGTH
      : MANIPULATE DATA TO
      : COME UP WITH THE
      : PROPER SYNC CHARACTER
      : LOAD THE CHARACTER
      : DITTO
      : LOAD THE SYNC REGISTER
      : SEL THE MISC REGISTER
      : SET TEST LOOP
      : FLIP THE BYTES
      : SET CHARACTER LENGTH
      : TURN CLOCK OFF...
      : AND ON
      : SEL RX PRIMARY ADDRESS
      : SET ADDRESS
      : SEL RX PRIMARY CHAR COUNT
      : SET CHAR COUNT
      : SEL TX PRIMARY ADDRESS
      : LOAD THE SYNC CHAR
      : SEL TX PRI CHAR COUNT
      : SET CHAR COUNT
      : SET RX GO
      : SET TX GO
      : START TIMING
      : IS DONE UP?
      : BRANCH IF YES
      : WAIT
      : BR IF MORE TO GO
      : ERROR--NO RX DONE
      : LOAD BUFFER POINTER
      : LOAD RX BUFFER POINTER
      : SET UP TO COUNT CHARACTERS
  
```



```

1577 004256 112005          MOVB (R0)+,R5          ;GET A CHARACTER TO COPMARE
1578 004260 005037 001246   CLR  TEMP2           ;
1579 004264 112137 001246   MOVB (R1)+,TEMP2     ;GET REC CHARACTER
1590 004270 013704 001246   MOV  TEMP2,R4        ;MOVE TO R4
1591 004274 043705 012134   BIC  MASK,R5         ;MASK OUT UNWANTED BITS
1592 004300 020504          CMP  R5,R4           ;DO THE CHARACTERS MATCH?
1593 004302 001401          BEQ  4$              ;BR IF OK
1594 004304 104002          HLT  2               ;ERROR--DATA DOESN'T MATCH
1595 004306 005302          4$: DEC  R2           ;ALL DONE?
1596 004310 001362          BNE  3$              ;NO--GO BACK FOR MORE
1597 004312 104400          SCOPE                ;SCOPE THIS TEST
  
```

```

;TEST OF TRANSMITTER AND RECEIVER CHARATER LENGHTHS
;THIS TEST WILL XMIT AND RECV CHARACTERS
;AT 3 BITS/PER/CHAR.
;DATA CHECKING WILL BE PERFORMED!
  
```

TEST 4

```

1597 004314 012737 000004 001226  *TST4: MOV  #4,TSTNO
1598 004322 012737 004654 001216  *TST5,NEXT
1599 004330 104413          MEMCLR                ;CLEAR ALL THE DQ11
1600 004332 012700 013022          MOV  #RXBUFF,R0      ;LOAD THE BUFFER POINTER
1601 004336 005001          CLR  R1               ;SET UP TO CLEAR THE BUFFER
1602 004340 005020          5$: CLR  (R0)+           ;CLEAR IT
1603 004342 105201          INCB R1               ;DONE?
1604 004344 100375          BPL  5$               ;BRANCH IF NO
1605 004346 112777 000011 175014  MOVB #11,DQREG        ;SELECT THE SYNC REG
1606 004354 013737 012616 001246  MOV  SYNC,TEMP2       ;LOAD SYNC
1607 004362 012737 177770 012134  MOV  #177770,MASK     ;LOAD THE MASK
1608 004370 143737 012134 001246  BICB MASK,TEMP2       ;SET UP A MASK TO GET THE
1609 004376 000241          CLC                    ;CORRECT SYNC CHARACTER
1610 004400 106037 001246          RORB TEMP2           ;FOR THIS CHARACTER LENGTH
1611 004404 143737 012134 001247  BICB MASK,TEMP2+1     ;MANIPULATE DATA TO
1612 004412 000241          CLC                    ;COME UP WITH THE
1613 004414 106037 001247          RORB TEMP2+1         ;PROPER SYNC CHARACTER
1614 004420 013737 001246 012136  MOV  TEMP2,SYNC1      ;LOAD THE CHARACTER
1615 004426 013737 001246 012140  MOV  TEMP2,SYNC2      ;DITTO
1616 004434 013777 001246 174730  MOV  TEMP2,DQSEC      ;LOAD THE SYNC REGISTER
1617 004442 105277 174722          INCB DQREG            ;SEL THE MISC REGISTER
1618 004446 012777 000010 174716  MOV  #BIT3,DQSEC      ;SET TEST LOOP
1619 004454 012700 000015          MOV  #15,R0           ;
1620 004460 000300          SWAB R0              ;FLIP THE BYTES
1621 004462 050077 174704          BIS  R0,DQSEC         ;SET CHARACTER LENGTH
1622 004466 052777 000002 174676  BIS  #BIT1,DQSEC      ;TURN CLOCK OFF...
1623 004474 042777 000002 174670  BIC  #BIT1,DQSEC      ;AND ON
1624 004502 105077 174662          CLRB DQREG           ;SEL RX PRIMARY ADRESS
1625 004506 012777 013022 174656  MOV  #RXBUFF,DQSEC    ;SET ADDRESS
1626 004514 105277 174650          INCB DQREG           ;SEL RX PRIMARY CHAR COUNT
1627 004520 012777 177734 174644  MOV  #-36,DQSEC       ;SET CHAR COUNT
1628 004526 105277 174636          INCB DQREG           ;SEL TX PRIMARY ADDRESS
1629 004532 012777 012140 174632  MOV  #SYNC2,DQSEC     ;LOAD THE SYNC CHAR
1630 004540 105277 174624          INCB DQREG           ;SEL TX PRI CHAR COUNT
1631 004544 012777 177732 174620  MOV  #-38,DQSEC       ;SET CHAR COUNT
1632 004552 005277 174602          INC  DQRC5R          ;SET RX GO
  
```

```

1633 004556 005277 174602          INC      JDQTCR          ;SET TX GO
1634 004562 005005                CLR      R5            ;START TIMING
1635 004564 105777 174570          1$:     TSTB     JDQRCR          ;IS DONE UP?
1636 004570 100404                BMI     2$            ;BRANCH IF YES
1637 004572 062705 000001          ADD     #1,R5        ;WAIT
1638 004576 001372                BNE     1$            ;BR IF MORE TO GO
1639 004600 104001                HLT     1              ;ERROR--NO RX DONE
1640 004602 012700 012142          2$:     MOV     #TXBUFF,R0 ;LOAD BUFFER POINTER
1641 004606 012701 013022          MOV     #RXBUFF,R1   ;LOAD RX BUFFER POINTER
1642 004612 012702 000044          MOV     #36.,R2      ;SET UP TO COUNT CHARACTERS
1643 004616                3$:
1644 004616 112005                MOVB    (R0)+,R5      ;GET A CHARACTER TO COPMARE
1645 004620 005037 001246          CLR     TEMP2         ;
1646 004624 112137 001246          MOVB    (R1)+,TEMP2   ;GET REC CHARACTER
1647 004630 013704 001246          MOV     TEMP2,R4     ;MOVE TO R4
1648 004634 043705 012134          BIC     MASK,R5      ;MASK OUT UNWANTED BITS
1649 004640 020504                CMP     R5,R4        ;DO THE CHARACTERS MATCH?
1650 004642 001401                BEQ     4$            ;BR IF OK
1651 004644 104002                HLT     2              ;ERROR--DATA DOESN'T MATCH
1652 004646 005302          4$:     DEC     R2          ;ALL DONE?
1653 004650 001362                BNE     3$            ;NO--GO BACK FOR MORE
1654 004652 104400                SCOPE                ;SCOPE THIS TEST
  
```

```

;TEST OF TRANSMITTER AND RECEIVER CHARATER LENGTHS
;THIS TEST WILL XMIT AND RECV CHARACTERS
;AT 4 BITS/PER/CHAR.
;DATA CHECKING WILL BE PERFORMED!
  
```

```

; TEST 5
;*****
  
```

```

1664 004654 012737 000005 001226  TST5:  MOV     #5,TSTNO
1665 004662 012737 005214 001216  MOV     #TST6,NEXT
1666 004670 104413                MEMCLR                ;CLEAR ALL THE DQ11
1667 004672 012700 013022          MOV     #RXBUFF,R0   ;LOAD THE BUFFER POINTER
1668 004676 005001                CLR     R1            ;SET UP TO CLEAR THE BUFFER
1669 004700 005020          5$:     CLR     (R0)+        ;CLEAR IT
1670 004702 105201                INCB   R1             ;DONE?
1671 004704 100375                BPL     5$            ;BRANCH IF NO
1672 004706 112777 000011 174454  MOVB    #11,JDQREG    ;SELECT THE SYNC REG
1673 004714 013737 012616 001246  MOV     SYNC,TEMP2    ;LOAD SYNC
1674 004722 012737 177760 012134  MOV     #177760,MASK  ;LOAD THE MASK
1675 004730 143737 012134 001246  BICB   MASK,TEMP2    ;SET UP A MASK TO GET THE
1676 004736 000241                CLC                    ;CORRECT SYNC CHARACTER
1677 004740 106037 001246                RORB   TEMP2          ;FOR THIS CHARACTER LENGTH
1678 004744 143737 012134 001247  BICB   MASK,TEMP2+1  ;MANIPULATE DATA TO
1679 004752 000241                CLC                    ;COME UP WITH THE
1680 004754 106037 001247                RORB   TEMP2+1        ;PROPER SYNC CHARACTER
1681 004760 013737 001246 012136  MOV     TEMP2,SYNC1   ;LOAD THE CHARACTER
1682 004766 013737 001246 012140  MOV     TEMP2,SYNC2   ;DITTO
1683 004774 013777 001246 174370  MOV     TEMP2,JDQSEC  ;LOAD THE SYNC REGISTER
1684 005002 105277 174362                INCB   JDQREG         ;SEL THE MISC REGISTER
1685 005006 012777 000010 174356  MOV     #BIT3,JDQSEC  ;SET TEST LOOP
1686 005014 012700 000014                MOV     #14,R0
1687 005020 000300                SWAB   R0             ;FLIP THE BYTES
1688 005022 050077 174344                BIS    R0,JDQSEC     ;SET CHARACTER LENGTH
  
```



```

1689 005026 052777 000002 174336 BIS #BIT1,ADQSEC ;TURN CLOCK OFF...
1690 005034 042777 000002 174330 BIC #BIT1,ADQSEC ;AND ON
1691 005042 105077 174322 CLRB ADQREG ;SEL RX PRIMARY ADDRESS
1692 005046 012777 013022 174316 MOV #RXBUFF,ADQSEC ;SET ADDRESS
1693 005054 105277 174310 INCB ADQREG ;SEL RX PRIMARY CHAR COUNT
1694 005060 012777 177734 174304 MOV #-36,ADQSEC ;SET CHAR COUNT
1695 005066 105277 174276 INCB ADQREG ;SEL TX PRIMARY ADDRESS
1696 005072 012777 012140 174272 MOV #SYNC2,ADQSEC ;LOAD THE SYNC CHAR
1697 005100 105277 174264 INCB ADQREG ;SEL TX PRI CHAR COUNT
1698 005104 012777 177732 174260 MOV #-38,ADQSEC ;SET CHAR COUNT
1699 005112 005277 174242 INC ADQRC5R ;SET RX GO
1700 005116 005277 174242 INC ADQTC5R ;SET TX GO
1701 005122 005005 CLR R5 ;START TIMING
1702 005124 105777 174230 1$: TSTB ADQRC5R ;IS DONE UP?
1703 005130 100404 BMI 2$ ;BRANCH IF YES
1704 005132 062705 000001 ADD #1,R5 ;WAIT
1705 005136 001372 BNE 1$ ;BR IF MORE TO GO
1706 005140 104001 HLT 1 ;ERROR--NO RX DONE
1707 005142 012700 012142 2$: MOV #TXBUFF,R0 ;LOAD BUFFER POINTER
1708 005146 012701 013022 MOV #RXBUFF,R1 ;LOAD RX BUFFER POINTER
1709 005152 012702 000044 MOV #36.,R2 ;SET UP TO COUNT CHARACTERS
1710 005156 3$: MOVB (R0)+,R5 ;GET A CHARACTER TO COPMARE
1711 005156 112005 CLR TEMP2 ;
1712 005160 005037 001246 MOVB (R1)+,TEMP2 ;GET REC CHARACTER
1713 005164 112137 001246 MOV TEMP2,R4 ;MOVE TO R4
1714 005170 013704 001246 BIC MASK,R5 ;MASK OUT UNWANTED BITS
1715 005174 043705 012134 CMP R5,R4 ;DO THE CHARACTERS MATCH?
1716 005200 020504 BEQ 4$ ;BR IF OK
1717 005202 001401 HLT 2 ;ERROR--DATA DOESN'T MATCH
1718 005204 104002 4$: DEC R2 ;ALL DONE?
1719 005206 005302 BNE 3$ ;NO--GO BACK FOR MORE
1720 005210 001362 SCOPE ;SCOPE THIS TEST
1721 005212 104400

```

```

;TEST OF TRANSMITTER AND RECEIVER CHARATER LENGHTHS
;THIS TEST WILL XMIT AND RECV CHARACTERS
;AT 5 BITS/PER/CHAR.
;DATA CHECKING WILL BE PERFORMED!

```

```

1722
1723
1724
1725
1726
1727
1728
1729
1730 ; TEST 6
*****
1731 005214 012737 000006 001226 TST6: MOV #6,TSTNO
1732 005222 012737 005554 001216 MOV #TST7,NEXT
1733 005230 104413 MEMCLR ;CLEAR ALL THE DQ11
1734 005232 012700 013022 MOV #RXBUFF,R0 ;LOAD THE BUFFER POINTER
1735 005236 005001 CLR R1 ;SET UP TO CLEAR THE BUFFER
1736 005240 005020 5$: CLR (R0)+ ;CLEAR IT
1737 005242 105201 INCB R1 ;DONE?
1738 005244 100375 BPL 5$ ;BRANCH IF NO
1739 005246 112777 000011 174114 MOVB #11,ADQREG ;SELECT THE SYNC REG
1740 005254 013737 012616 001246 MOV SYNC,TEMP2 ;LOAD SYNC5
1741 005262 012737 177740 012134 MOV #177740,MASK ;LOAD THE MASK
1742 005270 143737 012134 001246 BICB MASK,TEMP2 ;SET UP A MASK TO GET THE
1743 005276 000241 CLC ;CORRECT SYNC CHARACTER
1744 005300 106037 001246 RORB TEMP2 ;FOR THIS CHARACTER LENGTH

```

```

1745 005304 143737 012134 001247 BICB MASK,TEMP2+1 ;MANIPULATE DATA TO
1746 005312 000241 CLC ;COME UP WITH THE
1747 005314 106037 001247 RORB TEMP2+1 ;PROPER SYNC CHARACTER
1748 005320 013737 001246 012136 MOV TEMP2,SYNC1 ;LOAD THE CHARACTER
1749 005326 013737 001246 012140 MOV TEMP2,SYNC2 ;DITTO
1750 005334 013777 001246 174030 MOV TEMP2,ADQSEC ;LOAD THE SYNC REGISTER
1751 005342 105277 174022 INCB ADQREG ;SEL THE MISC REGISTER
1752 005346 012777 000010 174016 MOV #BIT3,ADQSEC ;SET TEST LOOP
1753 005354 012700 000013 MOV #13,R0 ;
1754 005360 000300 SWAB R0 ;FLIP THE BYTES
1755 005362 050077 174004 BIS RO,ADQSEC ;SET CHARACTER LENGTH
1756 005366 052777 000002 173776 BIS #BIT1,ADQSEC ;TURN CLOCK OFF...
1757 005374 042777 000002 173770 BIC #BIT1,ADQSEC ;AND ON
1758 005402 105077 173762 CLR B ADQREG ;SEL RX PRIMARY ADRESS
1759 005406 012777 013022 173756 MOV #RXBUFF,ADQSEC ;SET ADRESS
1760 005414 105277 173750 INCB ADQREG ;SEL RX PRIMARY CHAR COUNT
1761 005420 012777 177734 173744 MOV #-36,ADQSEC ;SET CHAR COUNT
1762 005426 105277 173736 INCB ADQREG ;SEL TX PRIMARY ADDRESS
1763 005432 012777 012140 173732 MOV #SYNC2,ADQSEC ;LOAD THE SYNC CHAR
1764 005440 105277 173724 INCB ADQREG ;SEL TX PRI CHAR COUNT
1765 005444 012777 177732 173720 MOV #-38,ADQSEC ;SET CHAR COUNT
1766 005452 005277 173702 INC ADQRCSR ;SET RX GO
1767 005456 005277 173702 INC ADQTCSR ;SET TX GO
1768 005462 005005 CLR R5 ;START TIMING
1769 005464 105777 173670 1$: TSTB ADQRCSR ;IS DONE UP?
1770 005470 100404 BMI 2$ ;BRANCH IF YES
1771 005472 062705 000001 ADD #1,R5 ;WAIT
1772 005476 001372 BNE 1$ ;BR IF MORE TO GO
1773 005500 104001 HLT 1 ;ERROR--NO RX DONE
1774 005502 012700 012142 2$: MOV #TXBUFF,R0 ;LOAD BUFFER POINTER
1775 005506 012701 013022 MOV #RXBUFF,R1 ;LOAD RX BUFFER POINTER
1776 005512 012702 000044 MOV #36,R2 ;SET UP TO COUNT CHARACTERS
1777 005516 3$: ;
1778 005516 112005 MOV B (R0)+,R5 ;GET A CHARACTER TO COPMARE
1779 005520 005037 001246 CLR TEMP2 ;
1780 005524 112137 001246 MOV B (R1)+,TEMP2 ;GET REC CHARACTER
1781 005530 013704 001246 MOV TEMP2,R4 ;MOVE TO R4
1782 005534 043705 012134 BIC MASK,R5 ;MASK OUT UNWANTED BITS
1783 005540 020504 CMP R5,R4 ;DO THE CHARACTERS MATCH?
1784 005542 001401 BEQ 4$ ;BR IF OK
1785 005544 104002 HLT 2 ;ERROR--DATA DOESN'T MATCH
1786 005546 005302 4$: DEC R2 ;ALL DONE?
1787 005550 001362 BNE 3$ ;NO--GO BACK FOR MORE
1788 005552 104400 SCOPE ;SCOPE THIS TEST
1789
1790
1791 ;TEST OF TRANSMITTER AND RECEIVER CHARATER LENGTHS
1792 ;THIS TEST WILL XMIT AND RECV CHARACTERS
1793 ;AT 6 BITS/PER/CHAR.
1794 ;DATA CHECKING WILL BE PERFORMED!
1795
1796 ; TEST 7
1797 ;*****
1798 005554 012737 000007 001226 TST7: MOV #7,TSTNO
1799 005562 012737 006114 001216 MOV #TST10,NEXT
1800 005570 104413 MEMCLR ;CLEAR ALL THE DQ11

```


1801	005572	012700	013022		MOV	#RXBUFF, R0	:LOAD THE BUFFER POINTER
1802	005576	005001			CLR	R1	:SET UP TO CLEAR THE BUFFER
1803	005600	005020		5\$:	CLR	(R0)+	:CLEAR IT
1804	005602	105201			INCB	R1	:DONE?
1805	005604	100375			BPL	5\$:BRANCH IF NO
1806	005606	112777	000011	173554	MOVB	#11, @DQREG	:SELECT THE SYNC REG
1807	005614	013737	012616	001246	MOV	SYNC, TEMP2	:LOAD SYNC
1808	005622	012737	177700	012134	MOV	#177700, MASK	:LOAD THE MASK
1809	005630	143737	012134	001246	BICB	MASK, TEMP2	:SET UP A MASK TO GET THE
1810	005636	000241			CLC		:CORRECT SYNC CHARACTER
1811	005640	106037	001246		RORB	TEMP2	:FOR THIS CHARACTER LENGTH
1812	005644	143737	012134	001247	BICB	MASK, TEMP2+1	:MANIPULATE DATA TO
1813	005652	000241			CLC		:COME UP WITH THE
1814	005654	106037	001247		RORB	TEMP2+1	:PROPER SYNC CHARACTER
1815	005660	013737	001246	012136	MOV	TEMP2, SYNC1	:LOAD THE CHARACTER
1816	005666	013737	001246	012140	MOV	TEMP2, SYNC2	:DITTO
1817	005674	013777	001246	173470	MOV	TEMP2, @DQSEC	:LOAD THE SYNC REGISTER
1818	005702	105277	173462		INCB	@DQREG	:SEL THE MISC REGISTER
1819	005706	012777	000010	173456	MOV	#BIT3, @DQSEC	:SET TEST LOOP
1820	005714	012700	000012		MOV	#12, R0	
1821	005720	000300			SWAB	R0	:FLIP THE BYTES
1822	005722	050077	173444		BIS	R0, @DQSEC	:SET CHARACTER LENGTH
1823	005726	052777	000002	173436	BIS	#BIT1, @DQSEC	:TURN CLOCK OFF...
1824	005734	042777	000002	173430	BIC	#BIT1, @DQSEC	:AND ON
1825	005742	105077	173422		CLRB	@DQREG	:SEL RX PRIMARY ADDRESS
1826	005746	012777	013022	173416	MOV	#RXBUFF, @DQSEC	:SET ADDRESS
1827	005754	105277	173410		INCB	@DQREG	:SEL RX PRIMARY CHAR COUNT
1828	005760	012777	177734	173404	MOV	#-36, @DQSEC	:SET CHAR COUNT
1829	005766	105277	173376		INCB	@DQREG	:SEL TX PRIMARY ADDRESS
1830	005772	012777	012140	173372	MOV	#SYNC2, @DQSEC	:LOAD THE SYNC CHAR
1831	006000	105277	173364		INCB	@DQREG	:SEL TX PRI CHAR COUNT
1832	006004	012777	177732	173360	MOV	#-38, @DQSEC	:SET CHAR COUNT
1833	006012	005277	173342		INC	@DQRC5R	:SET RX GO
1834	006016	005277	173342		INC	@DQTC5R	:SET TX GO
1835	006022	005005			CLR	R5	:START TIMING
1836	006024	105777	173330		TSTB	@DQRC5R	:IS DONE UP?
1837	006030	100404			BMI	2\$:BRANCH IF YES
1838	006032	062705	000001		ADD	#1, R5	:WAIT
1839	006036	001372			BNE	1\$:BR IF MORE TO GO
1840	006040	104001			HLT	1	:ERROR--NO RX DONE
1841	006042	012700	012142		MOV	#TXBUFF, R0	:LOAD BUFFER POINTER
1842	006046	012701	013022		MOV	#RXBUFF, R1	:LOAD RX BUFFER POINTER
1843	006052	012702	000044		MOV	#36, R2	:SET UP TO COUNT CHARACTERS
1844	006056			3\$:			
1845	006056	112005			MOVB	(R0)+, R5	:GET A CHARACTER TO COMPARE
1846	006060	005037	001246		CLR	TEMP2	
1847	006064	112137	001246		MOVB	(R1)+, TEMP2	:GET REC CHARACTER
1848	006070	013704	001246		MOV	TEMP2, R4	:MOVE TO R4
1849	006074	043705	012134		BIC	MASK, R5	:MASK OUT UNWANTED BITS
1850	006100	020504			CMP	R5, R4	:DO THE CHARACTERS MATCH?
1851	006102	001401			BEQ	4\$:BR IF OK
1852	006104	104002			HLT	2	:ERROR--DATA DOESN'T MATCH
1853	006106	005302		4\$:	DEC	R2	:ALL DONE?
1854	006110	001362			BNE	3\$:NO--GO BACK FOR MORE
1855	006112	104400			SCOPE		:SCOPE THIS TEST
1856							

M03

DZDQH MACY11 27(732) 24-SEP-76 10:11 PAGE 39
 DZDQHB.P11 PROGRAM INITIALIZATION AND START UP.

```

1913 006420 005037 001246 CLR TEMP2
1914 006424 112137 001246 MOV (R1)+,TEMP2
1915 006430 013704 001246 MOV TEMP2,R4
1916 006434 043705 012134 BIC MASK,R5
1917 006440 020504 CMP R5,R4
1918 006442 001401 BEQ 4$
1919 006444 104002 HLT 2
1920 006446 005302 4$: DEC R2
1921 006450 001362 BNE 3$
1922 006452 104400 SCOPE

```

```

;:
;: GET REC CHARACTER
;: MOVE TO R4
;: MASK OUT UNWANTED BITS
;: DO THE CHARACTERS MATCH?
;: BR IF OK
;: ERROR--DATA DOESN'T MATCH
;: ALL DONE?
;: NO--GO BACK FOR MORE
;: SCOPE THIS TEST

```

```

; TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
; THIS TEST WILL XMIT AND RECV CHARACTERS
; AT 8 BITS/PER/CHAR.
; DATA CHECKING WILL BE PERFORMED!

```

```

; TEST 11
; *****

```

```

1932 006454 012737 000011 001226 TST11: MOV #11,TSTNO
1933 006462 012737 007014 001216 MOV #TST12,NEXT
1934 006470 104413 MEMCLR
1935 006472 012700 013022 MOV #RXBUFF,R0
1936 006476 005001 CLR R1
1937 006500 005020 5$: CLR (R0)+
1938 006502 105201 INCB R1
1939 006504 100375 BPL 5$
1940 006506 112777 000011 172654 MOV #11,ADQREG
1941 006514 013737 012616 001246 MOV SYNC,TEMP2
1942 006522 012737 177400 012134 MOV #177400,MASK
1943 006530 143737 012134 001246 BICB MASK,TEMP2
1944 006536 000241 CLC
1945 006540 106037 001246 001247 RORB TEMP2
1946 006544 143737 012134 001247 BICB MASK,TEMP2+1
1947 006552 000241 CLC
1948 006554 106037 001247 RORB TEMP2+1
1949 006560 013737 001246 012136 MOV TEMP2,SYNC1
1950 006566 013737 001246 012140 MOV TEMP2,SYNC2
1951 006574 013777 001246 172570 MOV TEMP2,ADQSEC
1952 006602 105277 172562 INCB ADQREG
1953 006606 012777 000010 172556 MOV #BIT3,ADQSEC
1954 006614 012700 000010 MOV #10,R0
1955 006620 000300 SWAB R0
1956 006622 050077 172544 BIS R0,ADQSEC
1957 006626 052777 000002 172536 BIS #BIT1,ADQSEC
1958 006634 042777 000002 172530 BIC #BIT1,ADQSEC
1959 006642 105077 172522 CLRB ADQREG
1960 006646 012777 013022 172516 MOV #RXBUFF,ADQSEC
1961 006654 105277 172510 INCB ADQREG
1962 006660 012777 177734 172504 MOV #-36,ADQSEC
1963 006666 105277 172476 INCB ADQREG
1964 006672 012777 012140 172472 MOV #SYNC2,ADQSEC
1965 006700 105277 172464 INCB ADQREG
1966 006704 012777 177732 172460 MOV #-38,ADQSEC
1967 006712 005277 172442 INC ADQRCR
1968 006716 005277 172442 INC ADQTCR

```

```

; CLEAR ALL THE DQ11
; LOAD THE BUFFER POINTER
; SET UP TO CLEAR THE BUFFER
; CLEAR IT
; DONE?
; BRANCH IF NO
; SELECT THE SYNC REG
; LOAD SYNC
; LOAD THE MASK
; SET UP A MASK TO GET THE
; CORRECT SYNC CHARACTER
; FOR THIS CHARACTER LENGTH
; MANIPULATE DATA TO
; COME UP WITH THE
; PROPER SYNC CHARACTER
; LOAD THE CHARACTER
; DITTO
; LOAD THE SYNC REGISTER
; SEL THE MISC REGISTER
; SET TEST LOOP
; FLIP THE BYTES
; SET CHARACTER LENGTH
; TURN CLOCK OFF...
; AND ON
; SEL RX PRIMARY ADDRESS
; SET ADDRESS
; SEL RX PRIMARY CHAR COUNT
; SET CHAR COUNT
; SEL TX PRIMARY ADDRESS
; LOAD THE SYNC CHAR
; SEL TX PRI CHAR COUNT
; SET CHAR COUNT
; SET RX GO
; SET TX GO

```

```

1969 006722 005005          CLR      R5          ;START TIMING
1970 006724 105777 172430 1$:  TSTB   @DQRCR     ;IS DONE UP?
1971 006730 100404          BMI     2$          ;BRANCH IF YES
1972 006732 062705 000001  ADD    #1,R5        ;WAIT
1973 006736 001372          BNE    1$          ;BR IF MORE TO GO
1974 006740 104001          HLT    1            ;ERROR--NO RX DONE
1975 006742 012700 012142 2$:  MOV    #TXBUFF,R0   ;LOAD BUFFER POINTER
1976 006746 012701 013022  MOV    #RXBUFF,R1   ;LOAD RX BUFFER POINTER
1977 006752 012702 000044  MOV    #36.,R2      ;SET UP TO COUNT CHARACTERS
1978 006756
1979 006756 112005          MOVB   (R0)+,R5     ;GET A CHARACTER TO COPMARE
1980 006760 005037 001246  CLR    TEMP2        ;
1981 006764 112137 001246  MOVB   (R1)+,TEMP2  ;GET REC CHARACTER
1982 006770 013704 001246  MOV    TEMP2,R4     ;MOVE TO R4
1983 006774 043705 012134  BIC    MASK,R5      ;MASK OUT UNWANTED BITS
1984 007000 020504          CMP    R5,R4       ;DO THE CHARACTERS MATCH?
1985 007002 001401          BEQ   4$          ;BR IF OK
1986 007004 104002          HLT   2            ;ERROR--DATA DOESN'T MATCH
1987 007006 005302 4$:  DEC   R2           ;ALL DONE?
1988 007010 001362          BNE   3$          ;NO--GO BACK FOR MORE
1989 007012 104400          SCOPE 3$         ;SCOPE THIS TEST

```

```

;TEST OF TRANSMITTER AND RECEIVER CHARATER LENGTHS
;THIS TEST WILL XMIT AND RECV CHARACTERS
;AT 9 BITS/PER/CHAR.
;DATA CHECKING WILL BE PERFORMED!

```

```

; TEST 12
;*****

```

```

1999 007014 012737 000012 001226 †TST12: MOV    #12,TSTNO
2000 007022 012737 007326 001216  MOV    #TST13,NEXT
2001 007030 104413          MEMCLR
2002 007032 012700 013022  MOV    #RXBUFF,R0   ;CLEAR ALL THE DQ11
2003 007036 005001          CLR    R1           ;LOAD THE BUFFER POINTER
2004 007040 005020 5$:  CLR   (R0)+        ;SET UP TO CLEAR THE BUFFER
2005 007042 105201          INCB  R1           ;CLEAR IT
2006 007044 100375          BPL   5$          ;DONE?
2007 007046 112777 000011 172314  MOVB   #11,@DQREG   ;BRANCH IF NO
2008 007054 013737 012616 001246  MOV    SYNC,TEMP2   ;SELECT THE SYNC REG
2009 007062 012737 177000 012134  MOV    #177000,MASK ;LOAD SYNC
2010 007070 043737 012134 001246  BIC    MASK,TEMP2   ;LOAD THE MASK
2011 007076 000241          CLC
2012 007100 006037 001246          ROR   TEMP2        ;SET UP THE MASK FOR THE
2013 007104 013737 001246 012136  MOV    TEMP2,SYNC1  ;CORRECT SYNC CHARACTER
2014 007112 013737 001246 012140  MOV    TEMP2,SYNC2  ;SHIFT IT
2015 007120 013777 001246 172244  MOV    TEMP2,@DQSEC ;LOAD THE CHARACTER
2016 007126 105277 172236          INCB  @DQREG        ;DITTO
2017 007132 012777 000010 172232  MOV    #BIT3,@DQSEC ;LOAD THE SYNC REGISTER
2018 007140 012700 000007          MOV   #7,R0        ;SEL THE MISC REGISTER
2019 007144 000300          SWAB  R0           ;SET TEST LOOP
2020 007146 050077 172220          BIS   R0,@DQSEC    ;FLIP THE BYTES
2021 007152 052777 000002 172212  BIS   #BIT1,@DQSEC  ;SET CHARACTER LENGTH
2022 007160 042777 000002 172204  BIC   #BIT1,@DQSEC  ;TURN CLOCK OFF...
2023 007166 105077 172176          CLRB @DQREG        ;AND ON
2024 007172 012777 013022 172172  MOV   #RXBUFF,@DQSEC ;SEL RX PRIMARY ADRESS

```



```

007300 000000 012700 000000 172160 INCB @DQREG ;SEL RX PRIMARY CHAR COUNT
007301 000000 012700 000000 172160 MOV #13,@DQSEC ;SET CHAR COUNT
007302 000000 012700 000000 172146 INCB @DQREG ;SEL TX PRIMARY ADDRESS
007303 000000 012700 000000 172146 MOV #SYNCL,@DQSEC ;LOAD THE SYNC CHAR
007304 000000 012700 000000 172134 MOV #138,@DQSEC ;SET CHAR COUNT
007305 000000 012700 000000 172134 INC @DQRCR ;SET RX GO
007306 000000 012700 000000 172134 INC @DQTCR ;SET TX GO
007307 000000 012700 000000 172104 CLR @R5 ;START TIMING
007308 000000 012700 000000 172104 TSTB @DQRCR ;IS DONE UP?
007309 000000 012700 000000 000001 BMT @R5 ;BRANCH IF YES
007310 000000 012700 000000 000001 ADD #1,R5 ;WAIT
007311 000000 012700 000000 000001 SNE @R5 ;BR IF MORE TO GO
007312 000000 012700 000000 012146 HLT ;ERROR--NO RX DONE
007313 000000 012700 000000 013022 MOV @TXBUFF,R0 ;LOAD BUFFER POINTER
007314 000000 012700 000000 000044 MOV @RXBUFF,R1 ;LOAD RX BUFFER POINTER
007315 000000 012700 000000 000044 MOV #36,R2 ;SET UP TO COUNT CHARACTERS
007316 000000 012700 000000 012134 MOV (R0)+,R5 ;GET ANOTHER CHAR
007317 000000 012700 000000 012134 MOV (R1)+,R4 ;GET A REC CHAR
007318 000000 012700 000000 012134 BIC MASK,R5 ;MASK OUT UNWANTED BITS
007319 000000 012700 000000 012134 CMPB @R4,R5 ;DO THE CHARACTERS MATCH?
007320 000000 012700 000000 012134 BNE ;BR IF OK
007321 000000 012700 000000 012134 HLT ;ERROR--DATA DOESN'T MATCH
007322 000000 012700 000000 012134 DBC ;ALL DONE?
007323 000000 012700 000000 012134 JMP ;NO--GO BACK FOR MORE
007324 000000 012700 000000 012134 SCOPE ;SCOPE THIS TEST

```

```

:TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
:THIS TEST WILL XMIT AND RECV CHARACTERS
:AT 10 BITS/PER/CHAR.
:DATA CHECKING WILL BE PERFORMED!

```

```

: TEST 13
*****
TST13: MOV #13,TSTNO
MOV #TST14,NEXT
MEMCLR ;CLEAR ALL THE DQ11
MOV @RXBUFF,R0 ;LOAD THE BUFFER POINTER
CLR R1 ;SET UP TO CLEAR THE BUFFER
55: CLR (R0)+ ;CLEAR IT
INCB R1 ;DONE?
BPL 55 ;BRANCH IF NO
MOVB #11,@DQREG ;SELECT THE SYNC REG
MOV SYNC,TEMP2 ;LOAD SYNC
MOV #176000,MASK ;LOAD THE MASK
BIC MASK,TEMP2 ;SET UP THE MASK FOR THE
CLC ;CORRECT SYNC CHARACTER
ROR TEMP2 ;SHIFT IT
MOV TEMP2,SYNCL ;LOAD THE CHARACTER
MOV TEMP2,SYNCR ;DITTO
MOV TEMP2,@DQSEC ;LOAD THE SYNC REGISTER
INCB @DQREG ;SEL THE MISC REGISTER
MOV #BIT3,@DQSEC ;SET TEST LOOP
MOV #6,R0 ;

```

```

007326 012737 000013 001226
007327 012737 007640 001216
007328 104413
007329 012700 013022
007330 005001
007331 005020
007332 105201
007333 100375
007334 112777 000011 172002
007335 013737 012616 001246
007336 012737 176000 012134
007337 043737 012134 001246
007338 000241
007339 006037 001246
007340 013737 001246 012136
007341 013737 001246 012140
007342 013777 001246 171732
007343 105277 171724
007344 012777 000010 171720
007345 012700 000005

```


0007456	000300			SWAB	RO	: FLIP THE BYTES
0007457	0050077	171706		BIS	RO, @DQSEC	: SET CHARACTER LENGTH
0007458	0050077	0000000	171700	BIS	#BIT1, @DQSEC	: TURN CLOCK OFF...
0007459	0050077	0000000	171672	BIC	#BIT1, @DQSEC	: AND ON
0007460	1050077	171664		CLRB	@DQREG	: SEL RX PRIMARY ADDRESS
0007504	0130077	0130020	171660	MOV	#RXBUFF, @DQSEC	: SET ADDRESS
0007505	1050077	171652		INCB	@DQREG	: SEL RX PRIMARY CHAR COUNT
0007506	0130077	177734	171646	MOV	#-36, @DQSEC	: SET CHAR COUNT
0007507	1050077	171640		INCB	@DQREG	: SEL TX PRIMARY ADDRESS
0007508	0130077	012136	171634	MOV	#SYNC1, @DQSEC	: LOAD THE SYNC CHAR
0007509	1050077	171628		INCB	@DQREG	: SEL TX PRI CHAR COUNT
0007510	0130077	177730	171622	MOV	#-38, @DQSEC	: SET CHAR COUNT
0007511	0050077	171604		INC	@DQRCR	: SET RX GO
0007512	0050077	171604		INC	@DQTCR	: SET TX GO
0007513	0050005			CLR	R5	: START TIMING
0007514	1050077	171672	18:	TSTB	@DQRCR	: IS DONE UP?
0007515	100404			BMI	R5	: BRANCH IF YES
0007516	062705	000001		ADD	#1, R5	: WAIT
0007517	001372			BNE	R5	: BR IF MORE TO GO
0007518	104001			HLT		: ERROR--NO RX DONE
0007519	012700	012142	28:	MOV	#TXBUFF, RO	: LOAD BUFFER POINTER
0007520	012701	013020		MOV	#RXBUFF, R1	: LOAD RX BUFFER POINTER
0007521	012702	000044		MOV	#36, R2	: SET UP TO COUNT CHARACTERS
0007522			38:			
0007523	012005			MOV	(R0)+, R5	: GET ANOTHER CHAR
0007524	012005			MOV	(R1)+, R4	: GET A REC CHAR
0007525	043705	012134		BIC	MASK, R5	: MASK OUT UNWANTED BITS
0007526	020504			CMP	R5, R4	: DO THE CHARACTERS MATCH?
0007527	001401			BEG	R5	: BR IF OK
0007528	104002			HLT		: ERROR--DATA DOESN'T MATCH
0007529	005302		48:	DEC	R2	: ALL DONE?
0007530	001367			BNE	R2	: NO--GO BACK FOR MORE
0007531	104400			SCOPE		: SCOPE THIS TEST

: TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
 : THIS TEST WILL XMIT AND RECV CHARACTERS
 : AT 11 BITS/PER/CHAR.
 : DATA CHECKING WILL BE PERFORMED!

: TEST 14

 TST14: MOV #14, TSTNO
 MOV #TST15, NEXT
 MEMCLR : CLEAR ALL THE D011
 MOV #RXBUFF, RO : LOAD THE BUFFER POINTER
 CLR R1 : SET UP TO CLEAR THE BUFFER
 58: CLR (R0)+ : CLEAR IT
 INCB R1 : DONE?
 BPL SS : BRANCH IF NO
 MOVB #11, @DQREG : SELECT THE SYNC REG
 MOV SYNC, TEMP2 : LOAD SYNC
 MOV #174000, MASK : LOAD THE MASK
 BIC MASK, TEMP2 : SET UP THE MASK FOR THE
 CLC : CORRECT SYNC CHARACTER
 ROR TEMP2 : SHIFT IT

0007640	012737	000014	001226
0007646	012737	010152	001216
0007654	104413		
0007656	012700	013022	
0007662	005001		
0007664	005020		
0007666	105201		
0007670	100375		
0007672	112777	000011	171470
0007700	013737	012610	001246
0007706	012737	174000	012134
0007714	043737	012134	001246
0007722	000241		
0007724	006037	001246	


```

010137 007730 013737 001246 012136      MOV      TEMP2,SYNC1      ;LOAD THE CHARACTER
010138 007730 013737 001246 012140      MOV      TEMP2,SYNC2      ;DITTO
010139 007746 013777 001246 171420      MOV      TEMP2,JDQSEC      ;LOAD THE SYNC REGISTER
010140 007752 105277 171413      INCB     JDQREG           ;SEL THE MISC REGISTER
010141 007754 012777 000010 171406      MOV      #BIT3,JDQSEC      ;SET TEST LOOP
010142 007756 012700 000005      MOV      #5,R0            ;
010143 007770 000300      SWAB     R0              ;FLIP THE BYTES
010144 007772 050077 171374      BIS      R0,JDQSEC        ;SET CHARACTER LENGTH
010145 007774 052777 000002 171366      BIS      #BIT1,JDQSEC      ;TURN CLOCK OFF...
010146 007776 042777 000002 171360      BIC      #BIT1,JDQSEC      ;AND ON
010147 010001 105277 171353      CLRB     JDQREG           ;SEL RX PRIMARY ADDRESS
010148 010012 012777 013022 171346      MOV      #RXBUFF,JDQSEC    ;SET ADDRESS
010149 010014 105277 171340      INCB     JDQREG           ;SEL RX PRIMARY CHAR COUNT
010150 010016 012777 177734 171334      MOV      #-35,JDQSEC       ;SET CHAR COUNT
010151 010018 105277 171326      INCB     JDQREG           ;SEL TX PRIMARY ADDRESS
010152 010019 012777 012136 171322      MOV      #SYNC1,JDQSEC     ;LOAD THE SYNC CHAR
010153 010020 105277 171314      INCB     JDQREG           ;SEL TX PRI CHAR COUNT
010154 010021 012777 177732 171310      MOV      #-38,JDQSEC       ;SET CHAR COUNT
010155 010022 005277 171272      INC      JDQRCR           ;SET RX GO
010156 010023 005277 171272      INC      JDQTCR           ;SET TX GO
010157 010024 005005      CLR      R5              ;START TIMING
010158 010025 105777 171260      TSTB     JDQRCR           ;IS DONE UP?
010159 010100 100404      BMI      2$              ;BRANCH IF YES
010160 010102 062705 000001      ADD      #1,R5           ;WAIT
010161 010106 001372      BNE      1$              ;BR IF MORE TO GO
010162 010110 104001      HLT      1$              ;ERROR--NO RX DONE
010163 010112 012700 012142 2$:      MOV      #TXBUFF,R0       ;LOAD BUFFER POINTER
010164 010114 012701 013022      MOV      #RXBUFF,R1       ;LOAD RX BUFFER POINTER
010165 010116 012702 000044      MOV      #36.,R2          ;SET UP TO COUNT CHARACTERS
010166 010118 012005 3$:      MOV      (R0)+,R5         ;GET ANOTHER CHAR
010167 010120 012104      MOV      (R1)+,R4         ;GET A REC CHAR
010168 010122 043705 012134      BIC      MASK,R5          ;MASK OUT UNWANTED BITS
010169 010124 020504      CMP      R5,R4           ;DO THE CHARACTERS MATCH?
010170 010126 001401      BEQ      4$              ;BR IF OK
010171 010128 104002      HLT      2$              ;ERROR--DATA DOESN'T MATCH
010172 010130 005202 4$:      DEC      R2              ;ALL DONE?
010173 010132 001367      BNE      3$              ;NO--GO BACK FOR MORE
010174 010134 104400      SCOPE                    ;SCOPE THIS TEST

```

```

;TEST OF TRANSMITTER AND RECEIVER CHARATER LENGTHS
;THIS TEST WILL XMIT AND RECV CHARACTERS
;AT 12 BITS/PER/CHAR.
;DATA CHECKING WILL BE PERFORMED!

```

```

: TEST 15
:*****
TST15: MOV      #15,TSTNO
      MOV      #TST16,NEXT
      MEMCLR
      MOV      #RXBUFF,R0
      CLR      R1
5$:   CLR      (R0)+
      INCB     R1
      BPL     5$
      ;CLEAR ALL THE DQ11
      ;LOAD THE BUFFER POINTER
      ;SET UP TO CLEAR THE BUFFER
      ;CLEAR IT
      ;DONE?
      ;BRANCH IF NO

```

```

010152 012737 000015 001226
010160 012737 010464 001216
010166 104413
010170 012700 013022
010174 005001
010176 005020
010200 105201
010202 100375

```

```

0193 010204 112777 000011 171156      MOVB      #11,JDQREG      ;SELECT THE SYNC REG
0194 010212 013737 012616 001246      MOV      SYNC,TEMP2    ;LOAD SYNC
0195 010220 012737 170000 012134      MOV      #170000,MASK  ;LOAD THE MASK
0196 010226 043737 012134 001246      BIC      MASK,TEMP2    ;SET UP THE MASK FOR THE
0197 010234 000241                CLC                    ;CORRECT SYNC CHARACTER
0198 010236 006037 001246      ROR      TEMP2         ;SHIFT IT
0199 010242 013737 001246 012136      MOV      TEMP2,SYNC1   ;LOAD THE CHARACTER
0200 010250 013737 001246 012140      MOV      TEMP2,SYNC2   ;DITTO
0201 010256 013777 001246 171106      MOV      TEMP2,JDQSEC  ;LOAD THE SYNC REGISTER
0202 010264 105277 171100      INCB     JDQREG        ;SEL THE MISC REGISTER
0203 010270 012777 000010 171074      MOV      #BIT3,JDQSEC  ;SET TEST LOOP
0204 010276 012700 000004      MOV      #4,R0
0205 010302 000300      SWAB     R0            ;FLIP THE BYTES
0206 010304 050077 171062      BIS      R0,JDQSEC     ;SET CHARACTER LENGTH
0207 010310 052777 000002 171054      BIS      #BIT1,JDQSEC  ;TURN CLOCK OFF...
0208 010316 042777 000002 171046      BIC      #BIT1,JDQSEC  ;AND ON
0209 010324 105077 171040      CLRB     JDQREG        ;SEL RX PRIMARY ADDRESS
0210 010330 012777 013022 171034      MOV      #RXBUFF,JDQSEC ;SET ADDRESS
0211 010336 105277 171026      INCB     JDQREG        ;SEL RX PRIMARY CHAR COUNT
0212 010342 012777 177734 171022      MOV      #-36,JDQSEC   ;SET CHAR COUNT
0213 010350 105277 171014      INCB     JDQREG        ;SEL TX PRIMARY ADDRESS
0214 010354 012777 012136 171010      MOV      #SYNC1,JDQSEC ;LOAD THE SYNC CHAR
0215 010362 105277 171002      INCB     JDQREG        ;SEL TX PRI CHAR COUNT
0216 010366 012777 177732 170776      MOV      #-38,JDQSEC   ;SET CHAR COUNT
0217 010374 005277 170760      INC      JDQRCR        ;SET RX GO
0218 010400 005277 170760      INC      JDQTCR        ;SET TX GO
0219 010404 005005      CLR      R5            ;START TIMING
0220 010406 105777 170746      TSTB     JDQRCR        ;IS DONE UP?
0221 010412 100404      BMI     2$             ;BRANCH IF YES
0222 010414 062705 000001      ADD      #1,R5         ;WAIT
0223 010420 001372      BNE     1$             ;BR IF MORE TO GO
0224 010422 104001      HLT     1              ;ERROR--NO RX DONE
0225 010424 012700 012142      MOV      #TXBUFF,R0    ;LOAD BUFFER POINTER
0226 010430 012701 013022      MOV      #RXBUFF,R1    ;LOAD RX BUFFER POINTER
0227 010434 012702 000044      MOV      #36.,R2       ;SET UP TO COUNT CHARACTERS
0228 010440      3$:      MOV      (R0)+,R5      ;GET ANOTHER CHAR
0229 010442      MOV      (R1)+,R4      ;GET A REC CHAR
0230 010444 043705 012134      BIC      MASK,R5       ;MASK OUT UNWANTED BITS
0231 010450 020504      CMP      R5,R4         ;DO THE CHARACTERS MATCH?
0232 010452 001401      BEQ     4$             ;BR IF OK
0233 010454 104002      HLT     2              ;ERROR--DATA DOESN'T MATCH
0234 010456 005302      4$:      DEC      R2            ;ALL DONE?
0235 010460 001367      BNE     3$             ;NO--GO BACK FOR MORE
0236 010462 104400      SCOPE

```

:TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
:THIS TEST WILL XMIT AND RECV CHARACTERS
:AT 13 BITS/PER/CHAR.
:DATA CHECKING WILL BE PERFORMED!

```

: TEST 16
:*****
TST16: MOV      #16,TSTNO
        MOV      #TST17,NEXT

```

```

010464 012737 000016 001226
010472 012737 010776 001216

```


2249	010500	104413			MEMCLR		:CLEAR ALL THE DQ11
2250	010502	012700	013022		MOV #RXBUFF,RO		:LOAD THE BUFFER POINTER
2251	010506	005001			CLR R1		:SET UP TO CLEAR THE BUFFER
2252	010510	005020			5\$: CLR (R0)+		:CLEAR IT
2253	010512	105201			INCB R1		:DONE?
2254	010514	100375			BPL 5\$:BRANCH IF NO
2255	010516	112777	000011	170644	MOV #11,ADQREG		:SELECT THE SYNC REG
2256	010524	013737	012616	001246	MOV SYNC,TEMP2		:LOAD SYNC
2257	010532	012737	160000	012134	MOV #160000,MASK		:LOAD THE MASK
2258	010540	043737	012134	001246	BIC MASK,TEMP2		:SET UP THE MASK FOR THE
2259	010546	000241			CLC		:CORRECT SYNC CHARACTER
2260	010550	006037	001246		ROR TEMP2		:SHIFT IT
2261	010554	013737	001246	012136	MOV TEMP2,SYNC1		:LOAD THE CHARACTER
2262	010562	013737	001246	012140	MOV TEMP2,SYNC2		:DITTO
2263	010570	013777	001246	170574	MOV TEMP2,ADQSEC		:LOAD THE SYNC REGISTER
2264	010576	105277	170566		INCB ADQREG		:SEL THE MISC REGISTER
2265	010602	012777	000010	170562	MOV #BIT3,ADQSEC		:SET TEST LOOP
2266	010610	012700	000003		MOV #3,RO		
2267	010614	000300			SWAB RO		:FLIP THE BYTES
2268	010616	050077	170550		BIS RO,ADQSEC		:SET CHARACTER LENGTH
2269	010622	052777	000002	170542	BIS #BIT1,ADQSEC		:TURN CLOCK OFF...
2270	010630	042777	000002	170534	BIC #BIT1,ADQSEC		:AND ON
2271	010636	105077	170526		CLRB ADQREG		:SEL RX PRIMARY ADDRESS
2272	010642	012777	013022	170522	MOV #RXBUFF,ADQSEC		:SET ADDRESS
2273	010650	105277	170514		INCB ADQREG		:SEL RX PRIMARY CHAR COUNT
2274	010654	012777	177734	170510	MOV #-36,ADQSEC		:SET CHAR COUNT
2275	010662	105277	170502		INCB ADQREG		:SEL TX PRIMARY ADDRESS
2276	010666	012777	012136	170476	MOV #SYNC1,ADQSEC		:LOAD THE SYNC CHAR
2277	010674	105277	170470		INCB ADQREG		:SEL TX PRI CHAR COUNT
2278	010700	012777	177732	170464	MOV #-38,ADQSEC		:SET CHAR COUNT
2279	010706	005277	170446		INC ADQRCR		:SET RX GO
2280	010712	005277	170446		INC ADQTCR		:SET TX GO
2281	010716	005005			CLR R5		:START TIMING
2282	010720	105777	170434		1\$: TSTB ADQRCR		:IS DONE UP?
2283	010724	100404			BMI 2\$:BRANCH IF YES
2284	010726	062705	000001		ADD #1,R5		:WAIT
2285	010732	001372			BNE 1\$:BR IF MORE TO GO
2286	010734	104001			HLT 1		:ERROR--NO RX DONE
2287	010736	012700	012142		2\$: MOV #TXBUFF,RO		:LOAD BUFFER POINTER
2288	010742	012701	013022		MOV #RXBUFF,R1		:LOAD RX BUFFER POINTER
2289	010746	012702	000044		MOV #36,R2		:SET UP TO COUNT CHARACTERS
2290	010752				3\$:		
2291	010752	012005			MOV (R0)+,R5		:GET ANOTHER CHAR
2292	010754	012104			MOV (R1)+,R4		:GET A REC CHAR
2293	010756	043705	012134		BIC MASK,R5		:MASK OUT UNWANTED BITS
2294	010762	020504			CMP R5,R4		:DO THE CHARACTERS MATCH?
2295	010764	001401			BEQ 4\$:BR IF OK
2296	010766	104002			HLT 2		:ERROR--DATA DOESN'T MATCH
2297	010770	005302			4\$: DEC R2		:ALL DONE?
2298	010772	001367			BNE 3\$:NO--GO BACK FOR MORE
2299	010774	104400			SCOPE		:SCOPE THIS TEST
2300							
2301							
2302							
2303							
2304							

:TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
 :THIS TEST WILL XMIT AND RECV CHARACTERS
 :AT 14 BITS/PER/CHAR.

```

;DATA CHECKING WILL BE PERFORMED!
; TEST 17
*****
TST17: MOV #17,TSTNO
MOV #TST20,NEXT
MEMCLR ;CLEAR ALL THE DQ11
MOV #RXBUFF,RO ;LOAD THE BUFFER POINTER
CLR R1 ;SET UP TO CLEAR THE BUFFER
SS: CLR (R0)+ ;CLEAR IT
INCB R1 ;DONE?
BPL SS ;BRANCH IF NO
MOVB #11,ADQREG ;SELECT THE SYNC REG
MOV SYNC,TEMP2 ;LOAD SYNC
MOV #140000,MASK ;LOAD THE MASK
BIC MASK,TEMP2 ;SET UP THE MASK FOR THE
CLC ;CORRECT SYNC CHARACTER
ROR TEMP2 ;SHIFT IT
MOV TEMP2,SYNC1 ;LOAD THE CHARACTER
MOV TEMP2,SYNC2 ;DITTO
MOV TEMP2,ADQSEC ;LOAD THE SYNC REGISTER
INCB ADQREG ;SEL THE MISC REGISTER
MOV #BIT3,ADQSEC ;SET TEST LOOP
MOV #2,RO
SWAB RO ;FLIP THE BYTES
BIS RO,ADQSEC ;SET CHARACTER LENGTH
BIS #BIT1,ADQSEC ;TURN CLOCK OFF...
BIC #BIT1,ADQSEC ;AND ON
CLRB ADQREG ;SEL RX PRIMARY ADDRESS
MOV #RXBUFF,ADQSEC ;SET ADDRESS
INCB ADQREG ;SEL RX PRIMARY CHAR COUNT
MOV #-36,ADQSEC ;SET CHAR COUNT
INCB ADQREG ;SEL TX PRIMARY ADDRESS
MOV #SYNC1,ADQSEC ;LOAD THE SYNC CHAR
INCB ADQREG ;SEL TX PRI CHAR COUNT
MOV #-38,ADQSEC ;SET CHAR COUNT
INC ADQRCSR ;SET RX GO
INC ADQTCSR ;SET TX GO
CLR R5 ;START TIMING
IS: TSTB ADQRCSR ;IS DONE UP?
BMI 2S ;BRANCH IF YES
ADD #1,R5 ;WAIT
BNE 1S ;BR IF MORE TO GO
HLT 1 ;ERROR--NO RX DONE
2S: MOV #TXBUFF,RO ;LOAD BUFFER POINTER
MOV #RXBUFF,R1 ;LOAD RX BUFFER POINTER
MOV #36.,R2 ;SET UP TO COUNT CHARACTERS
3S: MOV (R0)+,R5 ;GET ANOTHER CHAR
MOV (R1)+,R4 ;GET A REC CHAR
BIC MASK,R5 ;MASK OUT UNWANTED BITS
CMP R5,R4 ;DO THE CHARACTERS MATCH?
BEQ 4S ;BR IF OK
HLT 2 ;ERROR--DATA DOESN'T MATCH
4S: DEC R2 ;ALL DONE?
BNE 3S ;NO--GO BACK FOR MORE

```



```

2361 011306 104400 SCOPE ;SCOPE THIS TEST
2362
2363
2364 :TEST OF TRANSMITTER AND RECEIVER CHARATER LENGHTHS
2365 :THIS TEST WILL XMIT AND RECV CHARACTERS
2366 :AT 15 BITS/PER/CHAR.
2367 :DATA CHECKING WILL BE PERFORMED!
2368
2369
2370 : TEST 20
2371 *****
2372 †TST20: MOV #20,TSTNO
2373 MOV #TST21,NEXT
2374 MEMCLR ;CLEAR ALL THE DQ11
2375 MOV #RXBUFF,RO ;LOAD THE BUFFER POINTER
2376 CLR R1 ;SET UP TO CLEAR THE BUFFER
2377 CLR (RO)+ ;CLEAR IT
2378 INCB R1 ;DONE?
2379 BPL 5$ ;BRANCH IF NO
2380 MOV #11,ADQREG ;SELECT THE SYNC REG
2381 MOV SYNC,TEMP2 ;LOAD SYNC
2382 MOV #100000,MASK ;LOAD THE MASK
2383 BIC MASK,TEMP2 ;SET UP THE MASK FOR THE
2384 CLC ;CORRECT SYNC CHARACTER
2385 ROR TEMP2 ;SHIFT IT
2386 MOV TEMP2,SYNC1 ;LOAD THE CHARACTER
2387 MOV TEMP2,SYNC2 ;DITTO
2388 MOV TEMP2,ADQSEC ;LOAD THE SYNC REGISTER
2389 INCB ADQREG ;SEL THE MISC REGISTER
2390 MOV #BIT3,ADQSEC ;SET TEST LOOP
2391 MOV #1,RO
2392 SWAB RO ;FLIP THE BYTES
2393 BIS RO,ADQSEC ;SET CHARACTER LENGTH
2394 BIS #BIT1,ADQSEC ;TURN CLOCK OFF...
2395 BIC #BIT1,ADQSEC ;AND ON
2396 CLRB ADQREG ;SEL RX PRIMARY ADRESS
2397 MOV #RXBUFF,ADQSEC ;SET ADRESS
2398 INCB ADQREG ;SEL RX PRIMARY CHAR :COUNT
2399 MOV #-36,ADQSEC ;SET CHAR COUNT
2400 INCB ADQREG ;SEL TX PRIMARY ADDRESS
2401 MOV #SYNC1,ADQSEC ;LOAD THE SYNC CHAR
2402 INCB ADQREG ;SEL TX PRI CHAR COUNT
2403 MOV #-38,ADQSEC ;SET CHAR COUNT
2404 INC ADQRC5R ;SET RX GO
2405 INC ADQTC5R ;SET TX GO
2406 CLR R5 ;START TIMING
2407 TSTB ADQRC5R ;IS DONE UP?
2408 BMI 2$ ;BRANCH IF YES
2409 ADD #1,R5 ;WAIT
2410 BNE 1$ ;BR IF MORE TO GO
2411 HLT 1 ;ERROR--NO RX DONE
2412 MOV #TXBUFF,RO ;LOAD BUFFER POINTER
2413 MOV #RXBUFF,R1 ;LOAD RX BUFFER POINTER
2414 MOV #36.,R2 ;SET UP TO COUNT CHARACTERS
2415
2416 1$: MOV (RO)+,R5 ;GET ANOTHER CHAR
MOV (R1)+,R4 ;GET A REC CHAR

```

```

2417 011602 043705 012134 BIC MASK,R5 ;MASK OUT UNWANTED BITS
2418 011606 020504 CMP R5,R4 ;DO THE CHARACTERS MATCH?
2419 011610 001401 BEQ 4$ ;BR IF OK
2420 011612 104002 HLT 2 ;ERROR--DATA DOESN'T MATCH
2421 011614 005302 4$: DEC R2 ;ALL DONE?
2422 011616 001367 BNE 3$ ;NO--GO BACK FOR MORE
2423 011620 104400 SCOPE ;SCOPE THIS TEST

```

```

;TEST OF TRANSMITTER AND RECEIVER CHARATER LENGHTHS
;THIS TEST WILL XMIT AND RECV CHARACTERS
;AT 16 BITS/PER/CHAR.
;DATA CHECKING WILL BE PERFORMED!

```

```

; TEST 21
;*****
2424 011622 012737 000021 001226 TST21: MOV #21,TSTNO
2425 011630 012737 014052 001216 MOV #.EOP,NEXT
2426 011636 104413 MEMCLR ;CLEAR ALL THE DQ11
2427 011640 012700 013022 MOV #RXBUFF,RO ;LOAD THE BUFFER POINTER
2428 011644 005001 CLR R1 ;SET UP TO CLEAR THE BUFFER
2429 011646 005020 5$: CLR (RO)+ ;CLEAR IT
2430 011650 105201 INCB R1 ;DONE?
2431 011652 100375 BPL 5$ ;BRANCH IF NO
2432 011654 112777 000011 167506 MOVB #11,ADQREG ;SELECT THE SYNC REG
2433 011662 013737 012616 001246 MOV SYNC,TEMP2 ;LOAD SYNC
2434 011670 012737 000000 012134 MOV #000000,MASK ;LOAD THE MASK
2435 011676 043737 012134 001246 BIC MASK,TEMP2 ;SET UP THE MASK FOR THE
2436 011704 000241 CLC ;CORRECT SYNC CHARACTER
2437 011706 006037 001246 ROR TEMP2 ;SHIFT IT
2438 011712 013737 001246 012136 MOV TEMP2,SYNC1 ;LOAD THE CHARACTER
2439 011720 013737 001246 012140 MOV TEMP2,SYNC2 ;DITTO
2440 011726 013777 001246 167436 MOV TEMP2,ADQSEC ;LOAD THE SYNC REGISTER
2441 011734 105277 167430 INCB ADQREG ;SEL THE MISC REGISTER
2442 011740 012777 000010 167424 MOV #BIT3,ADQSEC ;SET TEST LOOP
2443 011746 012700 000000 MOV #0,RO
2444 011752 000300 SWAB RO ;FLIP THE BYTES
2445 011754 050077 167412 BIS RO,ADQSEC ;SET CHARACTER LENGTH
2446 011760 052777 000002 167404 BIS #BIT1,ADQSEC ;TURN CLOCK OFF...
2447 011766 042777 000002 167376 BIC #BIT1,ADQSEC ;AND ON
2448 011774 105077 167370 CLRB ADQREG ;SEL RX PRIMARY ADRESS
2449 012000 012777 013022 167364 MOV #RXBUFF,ADQSEC ;SET ADDRESS
2450 012006 105277 167356 INCB ADQREG ;SEL RX PRIMARY CHAR COUNT
2451 012012 012777 177734 167352 MOV #-36.,ADQSEC ;SET CHAR COUNT
2452 012020 105277 167344 INCB ADQREG ;SEL TX PRIMARY ADDRESS
2453 012024 012777 012136 167340 MOV #SYNC1,ADQSEC ;LOAD THE SYNC CHAR
2454 012032 105277 167332 INCB ADQREG ;SEL TX PRI CHAR COUNT
2455 012036 012777 177732 167326 MOV #-38.,ADQSEC ;SET CHAR COUNT
2456 012044 005277 167310 INC ADQRCR ;SET RX GO
2457 012050 005277 167310 INC ADQTCSR ;SET TX GO
2458 012054 005005 CLR R5 ;START TIMING
2459 012056 105777 167276 1$: TSTB ADQRCR ;IS DONE UP?
2460 012062 100404 BMI 2$ ;BRANCH IF YES
2461 012064 062705 000001 ADD #1,R5 ;WAIT
2462 012070 001372 BNE 1$ ;BR IF MORE TO GO
2463 012072 104001 HLT 1 ;ERROR--NO RX DONE

```



```

2529 012252 000100          .BLKW 100
2530 012452          .MEMCLR:
2531 012452 005077 166702    CLR 3DQRCR
2532 012456 005077 166702    CLR 3DQTCR
2533 012462 005077 166700    CLR 3DQERR
2534 012466 012705 000020    MOV #16.,R5
2535 012472 152777 000020 166670 1$: BISB #BIT4,3DQREG
2536 012500 142777 000140 166662 BICB #140,3DQREG
2537 012506 005077 166660    CLR 3DQSEC
2538 012512 105277 166652    INCB 3DQREG
2539 012516 005305          DEC R5
2540 012520 001364          BNE 1$
2541 012522 105077 166642    CLRB 3DQREG
2542 012526 105077 166630    CLRB 3DQRCRSH
2543 012532 012705 000020    MOV #16.,R5
2544 012536 112777 000010 166624 2$: MOVB #10,3DQREG
2545 012544 005077 166622    CLR 3DQSEC
2546 012550 112777 000014 166612 MOVB #14,3DQREG
2547 012556 005077 166610    CLR 3DQSEC
2548 012562 105277 166574    INCB 3DQRCRSH
2549 012566 005305          DEC R5
2550 012570 001362          BNE 2$
2551 012572 105077 166564    CLRB 3DQRCRSH
2552 012576          .MSTCLR:
2553 012576 112777 000012 166564 MOVB #MISC.,3DQREG
2554 012604 012777 000040 166560 MOV #BITS,3DQSEC
2555 012612 000002          RTI
2556 012614 026 026      .SYNC: .BYTE 26,26
2557 012616 026 026      SYNC: .BYTE 26,26
2558 012620 000000      TXBFA: 0
2559 013022 013022          .=. +200
2560 013022 000200      RXBUFF:
2561 013422 026 026      .BLKW 200
2562 013424 026 026      XSYNC: .BYTE 26,26
2563 013426 000000      XSYNC2: .BYTE 26,26
2564 013630 000000      XTXBUF: 0
2565 013630 000000          .=. +200
2566 014032 000000      XRXBUF: 0
2567 014032 000000          .=. +200
2568 014034 000000      ERR: 0
2569 014036 000000      POLY: 0
2570 014040 000000      XPOLY: 0
2571 014042 000000      CHAR: 0
2572 014044 000000      COUNT: 0
2573 014046 000000      ADDR: 0
2574 014050 000000      GDCHAR: 0
2575 014052 000000      DETCAR: 0
2576
2577          ;END OF PASS
2578          ;TYPE NAME OF TEST
2579          ;UPDATE PASS COUNT
2580          ;CHECK FOR EXIT TO ACT-11
2581          ;RESTART TEST
2582
2583 014052 005037 001234      .EOP: CLR LSTERR          ;CLEAR LAST ERROR PC
2584 014056 005037 001312      CLR ERRFLG          ;CLEAR ERROR FLAG

```


2585	014062	005237	001230		INC	PASCNT		;UPDATE PASS COUNT
2586	014066	104402			TYPE			
2587	014070	016302			MEPASS			
2588	014072	104402			TYPE			
2589	014074	016463			MCSRX			
2590	014076	104411			CNVRT			
2591	014100	014210			XCSR			
2592	014102	104402			TYPE			
2593	014104	016471			MVECX			
2594	014106	104411			CNVRT			
2595	014110	014216			XVEC			
2596	014112	104402			TYPE			
2597	014114	016477			MPASSX			
2598	014116	104411			CNVRT			
2599	014120	014224			XPASS			
2600	014122	104402			TYPE			
2601	014124	016510			MERRX			
2602	014126	104411			CNVRT			
2603	014130	014232			XERR			
2604	014132	013777	001230	165042	MOV	PASCNT, @LIGHTS		;DISPLAY PASS COUNT
2605	014140	005337	001276		DEC	SAVNUM		
2606	014144	001013			BNE	RESTR		
2607	014146	013737	001504	001276	MOV	DQNUM, SAVNUM		
2608	014154	013701	000042		MOV	@#42, R1		;CHECK FOR ACT-11 OR DDP
2609	014160	001405			BEQ	RESTR		;IF NOT, CONTINUE TESTING
2610	014162	000005			RESET			
2611	014164				LOGICAL:			
2612	014164	004711			JSR	PC, (R1)		
2613	014166	000240			NOP			
2614	014170	000240			NOP			
2615	014172	000240			NOP			
2616	014174	104414			RESTR:	CKSWR		
2617	014176	012737	002254	001214	MOV	#TST1, RETURN		
2618	014204	000137	002254		JMP	TST1		
2619	014210	000001			XCSR:	1		
2620	014212	006	002		.BYTE	6,2		
2621	014214	001360			DQRCSR			
2622	014216	000001			XVEC:	1		
2623	014220	003	002		.BYTE	3,2		
2624	014222	001350			DQRVEC			
2625	014224	000001			XPASS:	1		
2626	014226	006	002		.BYTE	6,2		
2627	014230	001230			PASCNT			
2628	014232	000001			XERR:	1		
2629	014234	006	002		.BYTE	6,2		
2630	014236	001232			ERRCNT			
2631								
2632								;SCOPE LOOP AND ITERATION HANDLER
2633								
2634	014240	104414			.SCOPE:	CKSWR		
2635	014242	032777	040000	164730	BIT	#BIT14, @SWR		
2636	014250	001407			TTST:	1\$		
2637	014252	000432			BR	3\$		
2638	014254	105777	164724		TSTB	@TKCSR		
2639	014260	100027			BPL	3\$		
2640	014262	017700	164720		MOV	@TKDBR, R0		

DZDQH MACY11 27(732) 24-SEP-76 10:11 PAGE 52
 DZDQH.P11 GENERAL UTILITIES (TYPE OUT, ERROR, SCOPE, ETC.)

```

2641 014266 000412          BR      2$
2642 014270 032777 004000 164702 1$:  BIT      #SW11, @SWR
2643 014276 001006          BNE      2$
2644 014300 005237 001224          INC      LPCNT
2645 014304 023737 001224 001222  CMP      LPCNT, ICOUNT
2646 014312 001012          BNE      3$
2647 014314 105037 001312 2$:  CLR      CLR      ERRFLG
2648 014320 005037 001224          CLR      LPCNT
2649 014324 012737 000017 001222  MOV      #15., ICOUNT
2650 014332 013737 001216 001214  MOV      NEXT, RETURN
2651 014340 013716 001214 3$:  MOV      RETURN, (SP)
2652 014344 000002          RTI
2653 014346 001407  BRW:    1407
2654 014350 000432  BRX:    432
2655
2656          ;CHECK FOR FREEZE ON CURRENT DATA
2657
2658 014352 104414          .SCOPE: CKSWR
2659 014354 032777 001000 164616  BIT      #SW09, @SWR
2660 014362 001402          BEQ      1$
2661 014364 013716 001220  MOV      LOCK, (SP)
2662 014370 000002 1$:  RTI
2663
2664          ;TELETYPE OUTPUT ROUTINE
2665
2666 014372 010546          .TYPE:  MOV      R5, -(SP)
2667 014374 017605 000002  MOV      @2(SP), R5
2668 014400 062766 000002 000002  ADD      #2, 2(SP)
2669 014406 005737 016062 1$:  TST      @#RDSW
2670 014412 001004          BNE      300$
2671 014414 032777 010000 164556  BIT      #SW12, @SWR
2672 014422 001024          BNE      3$
2673 014424 105715 300$:  TSTB    (R5)
2674 014426 100014          BPL      2$
2675 014430 105777 164554  TSTB    @TPCSR
2676 014434 100375          BPL      .-4
2677 014436 012777 000015 164546  MOV      #15, @TPDBR
2678 014444 105777 164540  TSTB    @TPCSR
2679 014450 100375          BPL      .-4
2680 014452 012777 000012 164532  MOV      #12, @TPDBR
2681 014460 105777 164524 2$:  TSTB    @TPCSR
2682 014464 100375          BPL      2$
2683 014466 112577 164520  MOVB    (R5)+, @TPDBR
2684 014472 001345          BNE      1$
2685 014474 012605 3$:  MOV      (SP)+, R5
2686 014476 000002          RTI
2687
2688          ;ASCII STRING INPUT ROUTINE
2689
2690 014500 010346          .INSTR: MOV      R3, -(SP)
2691 014502 010446          MOV      R4, -(SP)
2692 014504 017637 000004 014522  MOV      @4(SP), .MSG
2693 014512 062766 000002 000004  ADD      #2, 4(SP)
2694 014520 104402          .INST1: TYPE
2695 014522 000000          .MSG:   0
2696 014524 012704 016654  MOV      #INBUF, R4

```


2697	014530	012703	000007			MOV	#7,R3	
2698	014534	105777	164444		1\$:	TSTB	ATKCSR	
2699	014540	100375				BPL	1\$	
2700	014542	117714	164440			MOVB	ATKDBR,(R4)	
2701	014546	142714	000200			BICB	#200,(R4)	
2702	014552	121427	000025			CMPB	(R4),#25	;IS IT <1G>
2703	014556	001003				BNE	200\$	
2704	014560	104402	016242			TYPE,MCRLF		
2705	014564	000755				BR	.INST1	
2706	014566	122427	000015		200\$:	CMPB	(R4)+,#15	
2707	014572	001423				BEQ	INSTR2	
2708	014574	117777	164406	164410		MOVB	ATKDBR,ATPDBR	
2709	014602	105777	164402		2\$:	TSTB	ATPCSR	
2710	014606	100375				BPL	2\$	
2711	014610	005303				DEC	R3	
2712	014612	001350				BNE	1\$	
2713	014614	000402				BR	.INSTG	
2714	014616	010346			.INSTE:	MOV	R3,-(SP)	
2715	014620	010446				MOV	R4,-(SP)	
2716	014622	104402			.INSTG:	TYPE		
2717	014624	016236				MQM		
2718	014626	005737	016062			TST	AT#RDSW	
2719	014632	001402				BEQ	400\$	
2720	014634	104402	016242			TYPE,MCRLF		
2721	014640	000727			400\$:	BR	.INST1	
2722	014642	012604			INSTR2:	MOV	(SP)+,R4	
2723	014644	012603				MOV	(SP)+,R3	
2724	014646	000002				RTI		
2725								
2726								;CONVERT ASCII STRING TO OCTAL
2727								
2728	014650	010546			.PARAM:	MOV	R5,-(SP)	
2729	014652	010446				MOV	R4,-(SP)	
2730	014654	016605	000004			MOV	4(SP),R5	
2731	014660	012537	015054			MOV	(R5)+,LOLIM	
2732	014664	012537	015056			MOV	(R5)+,HILIM	
2733	014670	012537	015060			MOV	(R5)+,DEVADR	
2734	014674	112537	015062			MOVB	(R5)+,LOBITS	
2735	014700	112537	015063			MOVB	(R5)+,ADRCNT	
2736	014704	010566	000004			MOV	R5,4(SP)	
2737	014710	005005			PARAM1:	CLR	R5	
2738	014712	012704	016654			MOV	#INBUF,R4	
2739	014716	122714	000015			CMPB	#15,(R4)	
2740	014722	001420				BEQ	PARERR	
2741	014724	121427	000060		1\$:	CMPB	(R4),#60	
2742	014730	002415				BLT	PARERR	
2743	014732	121427	000067			CMPB	(R4),#67	
2744	014736	003012				BGT	PARERR	
2745	014740	142714	000060			BICB	#60,(R4)	
2746	014744	152405				BISB	(R4)+,R5	
2747	014746	122714	000015			CMPB	#15,(R4)	
2748	014752	001414				BEQ	LIMITS	
2749	014754	006305				ASL	R5	
2750	014756	006305				ASL	R5	
2751	014760	006305				ASL	R5	
2752	014762	000760				BR	1\$	

```

014774 122714 000015 PARERR: CMPB #15, (R4) ;IS FIRST CHARACTER A (CR)
014775 0011003 BNE 120$
014776 005737 016062 TST @R0SW ;IS CKSWR ROUTINE BEING USED
014777 0010833 BNE PARTI
015000 104404 120$: INSTER
015001 000753 BR PARAM1

```

:TEST TO SEE IF NUMBER IS WITHIN LIMITS

```

015004 020537 015056 LIMITS: CMP R5, HILIM
015005 101356 BHI PARERR
015006 020537 015054 CMP R5, LOLIM
015007 103766 BLO PARERR
015008 133705 015062 BITB LOBITS, R5
015009 001357 BNE PARERR

```

:STORE NUMBER AT SPECIFIED ADDRESS

```

015010 013704 015060 IS: MOV DEVAR, R4
015011 010564 MOV R5, (R4)+
015012 065706 ADD #2, R5
015013 105227 000002 DECB ADRCNT
015014 001377 015063 BNE IS
015015 013704 PARTI: MOV (SP)+, R4
015016 010564 MOV (SP)+, R5
015017 000000 RTI
015018 000000 LOLIM:
015019 000000 HILIM:
015020 000000 DEVAR:
015021 000000 LOBITS:
015022 000000 ADRCNT=LOBITS+1

```

:SAVE PC OF TEST THAT FAILED AND RO-R5

```

015064 016637 000004 001274 .SAV05: MOV 4(SP), SAVPC
;SAVE RO-R5

```

```

SAV05: MOV R5, SAVR5
MOV R4, SAVR4
MOV R3, SAVR3
MOV R2, SAVR2
MOV R1, SAVR1
MOV R0, SAVR0
RTI

```

:RESTORE RO-R5

```

015124 013700 001256 .RES05: MOV SAVR0, R0
015125 013701 001260 MOV SAVR1, R1
015126 013702 001262 MOV SAVR2, R2
015127 013703 001264 MOV SAVR3, R3
015128 013704 001266 MOV SAVR4, R4
015129 013705 001270 MOV SAVR5, R5
015130 000002 RTI

```



```

015630 001402 BEQ WRKO.FM
015632 104402 TYPE
015634 000000 ERRMSG: 0
015636 005737 015646 WRKO.FM: TST DATAHD
015638 001402 BEQ TYPDAT
015640 104402 TYPE
015642 000000 DATAHD: 0
015644 005737 015660 TYPDAT: TST DATABP
015646 001402 BEQ RESREG
015648 104410 CONVRT
015650 000000 DATABP: 0
015652 104407 RESREG: RESOS
015654 005777 163310 HALTS: TST QSWR
015656 100005 BPL EXITER
015658 010046 PUSHRO
015660 016600 000002 MOV 2(SP),RO
015700 000000 HALT
015702 012600 POPRO
015704 104414 EXITER: CKSWR
015706 005237 001232 INC ERRCNT
015712 032777 000400 163260 BIT #SW08,QSWR
015720 001007 BNE 1$
015722 032777 002000 163250 BIT #SW10,QSWR
015730 001407 BEQ 2$
015732 013737 001216 001214 MOV NEXT,RETURN
015740 012706 001200 1$: MOV #STACK,SP
015744 000177 163244 JMP QRETURN
015750 000002 2$: RTI
015752 000001 ERTABO: 1
015754 006 002 .BYTE 6,2
015756 001274 SAVPC
015760 000001 XTSTN: 1
015762 003 002 .BYTE 3,2
015764 001226 TSTNO
;ENTER HERE ON POWER FAILURE

015766 .PFAIL:
015768 012737 016000 000024 MOV #RESTART,24 ;SET UP FOR POWER UP TRAP
015774 000000 HALT ;HALT ON POWER DOWN NORMAL
015776 000777 BR .
;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED

016000 RESTAR:
016002 012737 015766 000024 MOV #.PFAIL,24 ;SET UP FOR POWER FAILURE
016004 012706 001200 MOV #STACK,SP
016006 005037 016716 CLR TEMP
016008 005237 016716 INC TEMP
016010 001375 BNE .-4
016012 104402 TYPE
016014 016244 MPFAIL
016016 104411 CNVRT
016018 016054 PFTAB
016020 005037 001312 CLR ERRFLG

```

```

2977 016040 005037 001234
2978 016044 104412
2979 016046 104413
2980 016050 000177 163140
2981 016054 000001
2982 016056 003 002
2983 016060 001226
3000 016062 000000
3001 016064 005737 000042
3002 016070 001042
3003 016072 022737 000176 001200
3004 016100 001036
3005 016102 105777 163076
3006 016106 100033
3007 016110 017737 163072 014522
3008 016116 042737 177600 014522
3009 016124 122737 000007 014522
3010 016132 001021
3011 016134 104402 016212
3012 016140 005137 016062
3013 016144 104402 016216
3014 016150 104411 016204
3015 016154 104403 016225
3016 016160 104405
3017 016162 000000
3018 016164 177777
3019 016166 000176
3020 016170 000 001
3021 016172 104402 016242
3022 016176 005037 016062
3023 016202 000002
3024 016204 000001
3025 016206 006 002
3026 016210 000176
3027 016212 057377 000107
3028 016216 051777 051127 020075
3029 016224 000
3030 016225 040 047040 053505
3031 016232 020075 000
3032 016236 020040 000077
3033 016242 000377
3034 016244 050377 051127 043040
3035 016252 044501 042514 027104
3036 016260 051040 051505 040524
3037 016266 052122 040440 020124
3038 016274 042524 052123 000040
3039 016302 042777 042116 050040
3040 016310 051501 020123 055104

```

```

CLR LSTERR
MSTCLR
MEMCLR
JMP @RETURN
PFTAB: 1
.BYTE 3,2
TSTNO

```

```

:CHECK SWITCH REGISTER ROUTINE. CHECKS FOR +G TO ALLOW CHANGING
:OF LOC.176.
:LOCATIONS USED:
RDSW: .WORD 0

```

```

.CKSWR: TST @#42
BNE OUT
CMP #SWREG, SWR :SOFTWARE SWITCH REGISTER PRESENT
BNE OUT :NO GET OUT
TSTB @TKCSR :YES WAIT FOR
BPL OUT :READY GET CHARACTER
MOV @TKDBR, .MSG :AND STRIP OFF
BIC #177600, .MSG :THE GARBAGE
CMPB #7, .MSG :IS IT A +G?
BNE OUT

```

```

.CNTLU: COM @RDSW
TYPE, $MSWR
CNVRT, $WREGC
INSTR, $MNEW
PARAM
0
177777
SWREG

```

```

.BYTE 0,1
TYPE, MCRLF
OUT: CLR @RDSW
RTI

```

```

SWREGC: 1
.BYTE 6,2
SWREG
$CNTG: .ASCIZ <377>/+G/
$MSWR: .ASCIZ <377>/SWR= /
$MNEW: .ASCIZ / NEW= /

```

```

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

```

```

MEPASS: .ASCIZ <377>/END PASS DZDQH /

```


3033	016316	050504	020110	000040	
3034	016324	051377	000		MR: .ASCIZ <377>/R/
3035	016327	377	051120	043517	MERR2: .ASCIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT./
3036	016334	040522	020115	047111	
3037	016342	044504	040503	042524	
3038	016350	020123	047516	042040	
3039	016356	053105	041511	051505	
3040	016364	050040	042522	042523	
3041	016372	052116	000056		
3042	016376	044777	051516	043125	MERR3: .ASCIZ <377>/INSUFFICIENT DATA! /
3043	016404	044506	044503	047105	
3044	016412	020124	040504	040524	
3045	016420	000041			
3046	016422	052377	051505	020124	MTSTPC: .ASCIZ <377>/TEST PC-/
3047	016430	041520	000055		
3048	016434	046377	041517	020113	MLOCK: .ASCIZ <377>/LOCK ON SELECTED TEST/
3049	016442	047117	051440	046105	
3050	016450	041505	042524	020104	
3051	016456	042524	052123	000	
3052	016463	103	051123	020072	MCSRX: .ASCIZ /CSR: /
3053	016470	000			
3054	016471	126	041505	020072	MVECX: .ASCIZ /VEC: /
3055	016476	000			
3056	016477	120	051501	042523	MPASSX: .ASCIZ /PASSES: /
3057	016504	035123	000040		
3058	016510	051105	047522	051522	MERRX: .ASCIZ /ERRORS: /
3059	016516	020072	000		
3060	016521	377	052377	051505	MTSTN: .ASCIZ <377><377> /TEST NO: /
3061	016526	020124	047516	020072	
3062	016534	000			
3063	016535	377	042523	020124	MNEW: .ASCIZ <377>/SET SWITCH REG TO DQ11'S DESIRED ACTIVE./
3064	016542	053523	052111	044103	
3065	016550	051040	043505	052040	
3066	016556	020117	050504	030461	
3067	016564	051447	042040	051505	
3068	016572	051111	042105	040440	
3069	016600	052103	053111	027105	
3070	016606	000			
3071	016607	120	035103	000040	MERRPC: .ASCIZ /PC: /
3072	016614	046777	050101	047440	XHEAD: .ASCIZ <377>/MAP OF DQ11 STATUS/<377>
3073	016622	020106	050504	030461	
3074	016630	051440	040524	052524	
3075	016636	177523	000		
3076		016642			
3077	016642	000002			.EVEN
3078	016644	006	003		XSTATQ: 2
3079	016646	001244			.BYTE 6,3
3080	016650	006	002		TEMP1
3081	016652	001246			.BYTE 6,2
3082					TEMP2
3083					.EVEN
3084					;BUFFERS FOR INPUT-OUTPUT
3085					
3086	016654	000000			INBUF: 0
3087		016716			.+.40
3088	016716	000000			TEMP: 0

```

3089          016760          016760          . = +40
3090 016760 000000          MDATA: 0
3091          017022          . = +40
3092 017022 005015 042522 042503 EM0: .ASCIZ <15><12>/RECEIVER DONE PRIMARY NOT SET!/
017063 015 042012 052101 EM1: .ASCIZ <15><12>/DATA COMPARISON ERROR.../
017116 005015 050504 042440 EM2: .ASCIZ <15><12>/DQ ERROR FLAG SET. /
017144 005015 047516 051040 EM3: .ASCIZ <15><12>/NO RECEIVER INTERRUPTS!!!! /
017202 005015 054105 042520 DHO: .ASCIZ <15><12>/EXPECTED FOUND RX ADDR. TX ADDR. MASK /
017257 015 042412 050130 DH1: .ASCIZ <15><12>/EXPECTED RECEIVED /
017305 015 042012 042521 DH2: .ASCIZ <15><12>/DQERR /
          017322          .EVEN
          000005          DTO: 5
3093 017322 006 004          .BYTE 6,4
3094 017326 001270          SAVR5
3095 017330 006 001          .BYTE 6,1
3096 017332 001266          SAVR4
3097 017334 006 004          .BYTE 6,4
3098 017336 001260          SAVR1
3099 017340 006 004          .BYTE 6,4
3100 017342 001256          SAVR0
3101 017344 006 002          .BYTE 6,2
3102 017346 012134          MASK
3103 017350 000002          DT1: 2
3104 017352 003 006          .BYTE 3,6
3105 017354 014046          GDCHAR
3106 017356 003 002          .BYTE 3,2
3107 017360 014040          CHAR
3108 017362 000001          DT2: 1
3109 017364 006 002          .BYTE 6,2
3110 017366 014032          ERR
3111 017370          .ERRTAB:
3112 017370 000000          0
3113 017372 000000          0
3114 017374 000000          0
3115 017376 017022          EMO
3116 017400 000000          ;HALT 1
3117 017402 000000          0
3118 017404 017063          EM1
3119 017406 017202          DHO ;HALT 2
3120 017410 017322          DTO
3121 017412 017063          EM1
3122 017414 017257          DH1 ;HALT 3
3123 017416 017350          DT1
3124 017420 017116          EM2
3125 017422 017305          DH2 ;HALT 4
3126 017424 017362          DT2
3127 017426 017144          EM3
3128 017430 000000          ;HALT 5
3129 017432 000000          0
3130          000001          .END

```


DZDQH MACY11 27(732) 24-SEP-76 10:11 PAGE 76
 DZDQHB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

	1934	2001	2063	2125	2187	2249	2311	2373	2435	2577	2632	3092			
.MACRO	1	1363													
.NLIST	1	534	551	570	651	909	1036	1112	1114	1116	1118	1120	1122	1124	1126
	1138	1130	1132	1134	1136	1138	1198	1313	1380	1532	1599	1666	1733	1800	1867
	1934	2001	2063	2125	2187	2249	2311	2373	2435	2577	2632	3092			
.PAGE	1	43	534	570	651	909	1094								
.PAGNUM	1	43	50	58	101	186	192								
.SUBTITLE	6														
.TITLE	554	570	651	909	1036	1198	2577	2632							
.WORD	921	2989													

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

* DZDQHB.SEQ/SOL/CRF/PAGNUM/NL:TOC-UNIV.P11,DZDQHB.P11
 RUN-TIME: 20 31 4 SECONDS
 RUN-TIME RATIO: 125/58=2.1
 CORE USED: 20K (39 PAGES)

