

# DU11

OFF-LINE LOGIC TEST  
MD-11-DZDUA-B

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

NOV 1976  
digital  
MADE IN USA

TEST POINT	TEST POINT	TEST POINT	TEST POINT	TEST POINT	TEST POINT
1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36
37	38	39	40	41	42
43	44	45	46	47	48
49	50	51	52	53	54
55	56	57	58	59	60
61	62	63	64	65	66
67	68	69	70	71	72
73	74	75	76	77	78
79	80	81	82	83	84
85	86	87	88	89	90
91	92	93	94	95	96
97	98	99	100	101	102
103	104	105	106	107	108
109	110	111	112	113	114
115	116	117	118	119	120
121	122	123	124	125	126
127	128	129	130	131	132
133	134	135	136	137	138
139	140	141	142	143	144
145	146	147	148	149	150
151	152	153	154	155	156
157	158	159	160	161	162
163	164	165	166	167	168
169	170	171	172	173	174
175	176	177	178	179	180
181	182	183	184	185	186
187	188	189	190	191	192
193	194	195	196	197	198
199	200	201	202	203	204
205	206	207	208	209	210
211	212	213	214	215	216
217	218	219	220	221	222
223	224	225	226	227	228
229	230	231	232	233	234
235	236	237	238	239	240
241	242	243	244	245	246
247	248	249	250	251	252
253	254	255	256	257	258
259	260	261	262	263	264
265	266	267	268	269	270
271	272	273	274	275	276
277	278	279	280	281	282
283	284	285	286	287	288
289	290	291	292	293	294
295	296	297	298	299	300



## I D E N T I F I C A T I O N

PRODUCT NAME: DUII OFFLINE LOGIC TESTS

PRODUCT CODE: MAINDEC-11-DZDUA-D

RELEASE DATE: 21 AUG 1976

MAINTAINER : DIAGNOSTICS

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 OF RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1973, 1976 BY DIGITAL EQUIPMENT CORPORATION

## GENERAL DESCRIPTION

THIS DIAGNOSTIC CAN CHAIN 16 DUII'S. THIS MEANS THAT 16 DEVICES CAN BE SEQUENTIALLY EXERCISED. THE DIAGNOSTIC MAKES ONE PASS BEFORE PROCEEDING TO THE NEXT DEVICE, AND CONTINUES EXERCISING ALL DEVICES IN THIS FASHION UNTIL HALTED.

1. THE DUII OFFLINE LOGIC TESTS VERIFY THAT ALL REGISTERS EXIST AND ALL RESPECTIVE BITS CAN BE MASTER CLEARED, READ, WRITTEN AND/OR READ/WRITTEN

2. REQUIREMENTS

PDP-11 FAMILY STANDARD COMPUTER WITH OR WITHOUT HARDWARE SWITCH REGISTER (LOC. 177570)

DUII SYNCHRONOUS/ISOCRONOUS OPTION

ONE CONSOLE TELETYPE OR EQUIVALENT

- 2.2 STORAGE

THE PROGRAM LOADS AND RUNS IN BK OF MEMORY.

3. LOADING PROCEDURE

THE STANDARD PROCEDURE FOR LOADING ABSOLUTE BINARY TAPES IS TO BE USED.

	STARTING ADDRESS FOR ABSOLUTE LOADER
4K	017500
8K	037500
12K	057500
16K	077500
20K	117500
24K	137500
28K	157500

4. STARTING PROCEDURE

- 4.1 CONTROL SWITCH SETTINGS

NOTE: SOFTWARE SWITCH REGISTER IS DEFINED AS LOC. 176, WHILE THE SOFTWARE DISPLAY REGISTER IS DEFINED AS LOC. 174.

- 4.1.1 AFTER PROGRAM LOAD (INITIAL PROGRAM START)

ALL CONSOLE SWITCHES DOWN

- 4.1.2 TO MODIFY DEVICE VECTOR AND CONTROL REGISTER ADDRESSES  
AFTER PROGRAM RESTART OR TO RUN MULTIPLE DEVICES

SW00=1

- 4.1.3 TO START PROGRAM AT SELECTED TEST AFTER A PROGRAM RESTART  
(ONLY IN SINGLE DEVICE TESTS)

SW01=1

- 4.1.4 TO LOCK ON SELECTED TEST AFTER A PROGRAM RESTART  
(ONLY IN SINGLE DEVICE TESTS)

SW02=1

NOTE1: IN GENERAL SW01 WILL BE USED WHEN SW02=1 IS USED

NOTE2: WITHOUT SW01=1 "LOCK ON TEST" WILL DEFAULT TO TEST 1  
STARTING ADDRESS

- 4.2

THE STARTING ADDRESS FOR ALL TESTS IS 000200

THE RETARTING ADDRESS FOR ALL TESTS IS 000200

THE STARTING ADDRESS TO ENTER A SELECTED TEST IS 000200

THE STARTING ADDRESS TO LOCK ON TEST IS 000200

- 4.3 PROGRAM AND/OR OPERATOR ACTION

- 4.3.1 INITIAL PROGRAM START

4.3.1.1 LOAD PROGRAM INTO MEMORY WITH ABSOLUTE LOADER

4.3.1.2 LOAD ADDRESS 000200

4.3.1.3 CLEAR CONSOLE SWITCHES

4.3.1.4 PRESS START

4.3.1.5 THE PROGRAM WILL TYPE "DUI1 DZDUA-D TAPE A" (ONCE ONLY)

NOTE: IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING  
WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:  
SWR=XXXXXX NEW= (REFER TO SECTION 5. FOR OPERATOR'S OPTION)

4.3.1.7 THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT IS ABOUT  
TO START TESTING , AND THEN TESTING WILL BEGIN

- 4.3.2 PROGRAM RESTART WITH ALL SWITCHES DOWN

4.3.2.1 THE PROGRAM WILL TYPE "R" AND WILL COMMENCE TESTING

- 4.3.3 PROGRAM RESTART WITH SW00=1

4.3.3.1 LOAD ADDRESS 000200

4.3.3.2 SET SW00=1

4.3.3.3 PRESS START

NOTE: IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING  
WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:  
SWR=XXXXXX NEW= (REFER TO SECTION 5. FOR OPERATOR'S OPTION)

4.3.3.4 THE PROGRAM WILL TYPE " 1ST DEVICE RECEIVER CONTROL REGISTER  
ADDRESS" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.5 TYPE IN THE ADDRESS OF THE FIRST RECEIVER CONTROL  
REGISTER ADDRESS OF THE DU11 TO BE TESTED  
FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ADDRESS IS TYPED, THE PROGRAM WILL TYPE "?"  
AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.4

4.3.3.6 THE PROGRAM WILL TYPE "VECTOR ADDRESS-" AND WAIT FOR AN  
INPUT FROM THE TELETYPE KEYBOARD

4.3.3.7 TYPE IN THE BASE RECEIVER INTERRUPT VECTOR ADDRESS  
FOR THE DU11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ADDRESS IS TYPED, THE PROGRAM WILL TYPE "?"  
AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.6

4.3.3.8 THE PROGRAM WILL TYPE "ARE YOU RUNNING MULTIPLE DEVICES ?"  
(Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.9 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A  
<CARRIAGE RETURN>

IF AN INCORRECT ANSWER IS GIVEN, THE PROGRAM WILL TYPE "?"  
AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.8

IF A "NO" ANSWER IS GIVEN: JUMP TO SECTION 4.3.3.12  
IF A "YES" ANSWER IS GIVEN: THE NEXT QUESTION IS ASKED

4.3.3.10 THE PROGRAM WILL TYPE "LAST DEVICE: RECEIVER CONTROL  
REGISTER ADDRESS-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.11 TYPE IN THE ADDRESS OF THE LAST RECEIVER CONTROL REGISTER  
ADDRESS OF THE DU11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ANSWER IS TYPED, THE PROGRAM WILL TYPE "?"  
AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.10  
NOTE: ALL ADDRESSES SHALL BE CONTIGUOUS

4.3.3.11.1 IF AN "OUT OF RANGE" ADDRESS IS TYPED  
IE. MORE THAN 16 (10) DEVICES AWAY (UPWARDS).....THE  
PROGRAM WILL TYPE "OUT OF RANGE:RETYPE LAST DEVICE RXCSR ADDRESS-"  
AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.11.2 TYPE IN THE ADDRESS OF THE LAST RECEIVER CONTROL  
REGISTER ADDRESS OF THE DU11 TO BE TESTED FOLLOWED  
BY A <CARRIAGE RETURN>

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"  
AND WILL REPEAT THE MESSAGE OF 4.3.3.11.1

IF A DEVICE ADDRESS LOWER THAN 1ST DEVICE ADDRESS IS TYPED.....  
.....SCHOOLS OUT.....THERE IS NO PROTECTION FOR THIS.  
THE PROGRAM WILL DEFAULT TO TWO DEVICES ACTIVE (UPWARDS FROM  
1ST DEVICE ADDRESS).THE SAME APPLIES TO IDENTICAL ADDRESSES  
TYPED FOR FIRST AND LAST DEVICE.  
OBSERVE LOCATION @ ACTREG: SEE SECTION 7.2

4.3.3.12 THE PROGRAM WILL TYPE "DU PRIORITY LEVEL-" AND  
WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.13 TYPE IN THE APPROPRIATE DEVICE PRIORITY LEVEL OF THE  
DU11 OR DU11'S TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>  
(NOTE THAT ALL MULTIPLE DEVICES MUST BE AT THE SAME PRIORITY  
LEVEL). IE "5"

IF AN INCORRECT LEVEL IS TYPED ,THE PROGRAM WILL TYPE "?"  
AND REPEAT THE MESSAGE OF 4.3.3.12

4.3.3.14 THE PROGRAM WILL TYPE "# OF SYNC CHARS  
SELECTED (1 OR 2)-" AND WAIT FOR AN INPUT FROM THE TELETYPE  
KEYBOARD

4.3.3.15 TYPE IN THE APPROPRIATE ANSWER "1" OR "2" FOLLOWED  
BY A <CARRIAGE RETURN>.(NOTE:ALL MULTIPLE DEVICES MUST  
BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"  
AND WILL REPEAT THE MESSAGE OF 4.3.3.14

4.3.3.16 THE PROGRAM WILL TYPE " IS SEC XMIT JUMPER #6 IN ? (Y OR N)-"  
AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.17 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED  
BY A <CARRIAGE RETURN>.(NOTE THAT ALL MULTIPLE DEVICES  
MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"  
AND WILL REPEAT THE MESSAGE OF 4.3.3.16

4.3.3.18 THE PROGRAM WILL TYPE "IS SEC REC JUMPER # 5 IN ?  
(Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.19 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED  
BY A <CARRIAGE RETURN>. (NOTE: ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"  
AND WILL REPEAT THE MESSAGE OF 4.3.3.18

4.3.3.20 THE PROGRAM WILL TYPE "IS OPT CLR ENABLE JUMPER  
# 4 IN ? (Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.21 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED  
BY A <CARRIAGE RETURN>. (NOTE: ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"  
AND WILL REPEAT THE MESSAGE OF 4.3.3.20

4.3.3.22 THE PROGRAM WILL TYPE "ARE YOU RUNNING IN MAINT.  
MODE EXTERNAL ? AND .....DO YOU HAVE THE EXTERNAL MODEM  
BYPASS JUMPER (CONNECTOR ON ? (Y OR N)-" AND WAIT FOR AN  
INPUT FROM THE TELETYPE KEYBOARD

4.3.3.23 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY  
A <CARRIAGE RETURN>. (NOTE: ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"  
AND WILL REPEAT THE MESSAGE OF 4.3.3.22

4.3.3.24 THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT  
HAS STARTED AND WILL COMMENCE TESTING AT TEST 1

4.3.4 PROGRAM RESTART WITH SW01=1  
NOTE: THIS WILL ONLY WORK WHEN A SINGLE DEVICE IS SELECTED  
,,IT WILL NOT WORK IF MULTIPLE DEVICES ARE SELECTED

IF MULTIPLE DEVICES WERE PREVIOUSLY SELECTED,LOAD 000200,  
AND SELECT SW0C=1 AND ANSWER "NO" TO THE MULTIPLE DEVICE QUESTION  
SEE 4.3.3

4.3.4.1 LOAD 000200

4.3.4.2 SET SW01=1

4.3.4.3 PRESS START

NOTE: IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING  
WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:  
SWR=XXXXXX NEW= (REFER TO SECTION 5. FOR OPERATOR'S OPTION)

4.3.4.4 THE PROGRAM WILL TYPE "TEST PC-" AND WAIT FOR AN INPUT FROM  
THE TELETYPE KEYBOARD

4.3.4.5 TYPE IN THE ADDRESS OF THE TEST AT WHICH THE PROGRAM IS TO

BE STARTED FOLLOWED BY A <CARRIAGE RETURN>

4.3.4.6 THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT HAS STARTED TESTING AT THE SELECTED TEST

NOTE: CARE MUST BE TAKEN WHEN THIS FEATURE IS USED SINCE THERE IS NO PROTECTION AGAINST SELECTING AN ADDRESS THAT IS IN THE MIDDLE OF A TEST

4.3.5 PROGRAM RESTART WITH SW02 =1  
NOTE: THIS WILL ONLY WORK WHEN A SINGLE DEVICE IS SELECTED  
SEE NOTE IN 4.3.4 FOR MORE DETAILS

4.3.5.1 LOAD ADDRESS 000200

4.3.5.2 SET SW02 =1  
NOTE: IT MAY BE ADVANTAGEOUS TO SET SW01=1 (OPTIONAL)

4.3.5.3 PRESS START

NOTE: IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:  
SWR=XXXXXX NEW= (REFER TO SECTION 5. FOR OPERATOR'S OPTION)

4.3.5.4 THE PROGRAM WILL TYPE "LOCK ON SELECTED TEST ? (Y OR N)-"  
AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.5.5 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A <CARRIAGE RETURN>

IF A NO ANSWER IS GIVEN: THIS LOCK ON TEST WILL BE IGNORED AND THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT HAS STARTED TESTING AT TEST 1

4.3.5.6 IF A YES ANSWER WAS GIVEN: THE PROGRAM WILL ACT AS FOLLOWS...  
THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT HAS STARTED TESTING AT TEST 1 AND WILL REMAIN IN TEST 1 UNTIL HALTED OR IF ANY KEY IS STRUCK ON THE TELETYPE THE PROGRAM WILL FREEZE ON THE NEXT TEST UNTIL A KEY IS STRUCK ON THE TELETYPE AND SO FORTH THRU THE PROGRAM. IF SW01 =1 IT WILL PERFORM AS IN SECTION 4.3.4 ALLOWING ONE TO FREEZE ON A SELECTED TEST RATHER THAN DEFAULTING TO TEST 1

## 5. OPERATING PROCEDURE

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

CONTROL:



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

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

5.1 OPERATIONAL SWITCH SETTINGS

SW15 =1	HALT ON ERROR
SW14 =1	LOOP ON CURRENT TEST
SW13 =1	INHIBIT ERROR TYPEOUT
SW11 =1	INHIBIT ITERATIONS
SW10 =1	ESCAPE TO NEXT TEST ON ERROR
SW08 =1	LOOP ON ERROR
SW02 =1	LOCK ON TEST
SW01 =1	RESTART PROGRAM AT SELECTED TEST
SW00 =1	RESELECT VECTOR AND CONTROL REGISTER ADDRESSES & PARAMETERS AFTER A PROGRAM RESTART

TO INHIBIT "END OF PASS" TYPEOUT - TURN TELETYPE OFF

## 6. ERRORS

6.1 ERROR HALTS  
THERE ARE FOUR DISTINCT ERROR TYPEOUTS

NOTE: 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 5. FOR OPERATOR OPTION)

6.1.1 PC+2 = ERROR PC  
WHERE PC +2 IS THE ADDRESS OF THE CALL TO THE ERROR HANDLER +2

REFER TO THE ABOVE "HLT" IN DIAGNOSTIC FOR ERROR DESCRIPTION

CHECK ADDRESS @ RXCSR: TO LOCATE THE DEVICE PRESENTLY UNDER TEST WHEN RUNNING MULTIPLE DEVICES

6.1.2 PC +2 = REGISTER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING DEVICE REGISTER

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.3 PC +2 = RECEIVER ERROR PC  
REGISTER            EXPECTED            ACTUAL  
16XXXX            YYYYYY            ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING RECEIVER (RXDBUF) REGISTER

WHERE YYYYYY IS THE EXPECTED DATA CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL DATA CONTENTS OF THAT REGISTER

6.1.4 PC +2 = TRANSMITTER ERROR PC  
REGISTER            EXPECTED            ACTUAL  
16XXXX            YYYYYY            ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING TRANSMITTER (TXCSR) REGISTER

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.5 ERROR DESCRIPTIONS  
SEE LISTINGS FOR DETAILS OF ERRORS

6.2 ERROR RECOVERY

6.2.1 SW15 =0  
IF THE PROGRAM IS RUN WITH SW15 =0 ,NO OPERATOR ACTION IS  
REQUIRED TO CONTINUE TESTING

6.2.2 SW15 =1  
IF THE PROGRAM IS RUN WITH SW15 =1 ,TO CONTINUE TESTING  
AFTER THE PROGRAM HAS HALTED ,PRESS THE PROCESSOR  
CONSOLE "CONTINUE SWITCH"

NOTE: THE PC + 2 OF THE "HLT" WILL BE DISPLAYED IN THE DATA LIGHTS

6.2.3 ILLEGAL INTERRUPTS  
IF AN INTERRUPT OCCURS TO A VECTOR ADDRESS NOT SELECTED  
DURING PROGRAM INITIALIZATION, THE PROGRAM WILL HALT IN  
THE TRAPCATCHER. THE ADDRESS AT WHICH THE PROGRAM  
HALTS IS 2 GREATER THAN THE ADDRESS TO WHICH THE INTERRUPT  
OCCURED. THE PROGRAM MUST BE RESTARTED AT 000200 TO  
RECOVER FROM THIS ERROR.

6.2.4 ADDITIONAL TROUBLESHOOTING AIDS    ERRCNT: & PASCNT:  
CHECK THESE TWO TAG LOCATIONS FOR TOTAL # OF ERRORS AND PASSES RESPECTIVELY.  
LOADING 000200 AND RESTARTING WILL CLEAR THESE LOCATIONS.

6.3 END OF PASS ROUTINE

THIS TYPEOUT IS MENTIONED HERE FOR CONVENIENCE  
IT IS IN THE FORM:

END OF PASS TAPE Y  
16XXXX = DEVICE

WHERE Y IS THE TAPE LOADED

WHERE 16XXXX IS THE DEVICE'S BASE REGISTER ADDRESS

TO INHIBIT THIS TYPEOUT - TURN TELETYPE OFF

## 7. RESTRICTIONS

### 7.1 MULTIPLE DEVICES

UP TO 16(10) DEVICES MAY BE TESTED. HOWEVER, THEY  
MUST HAVE CONTIGUOUS ADDRESSES AND VECTORS

NOTE: IF ALL DEVICES UNDER TEST HAVE THE SAME INTERRUPT VECTOR  
YOU CAN CHANGE "ZERO: ADD #10, BASEIV ;NEXT BLOCK  
(VECTORS)" TO "ZERO: ADD #0, BASEIV";  
THEREBY THE VECTOR ADDRESSES WILL NOT BE  
UPDATED AFTER EACH PASS.

### 7.2 DISQUALIFYING DEVICES WHEN RUNNING MULTIPLE DEVICES

WHEN RUNNING MULTIPLE DEVICES AN ACTIVE BIT IS SET  
FOR EACH DEVICE RUNNING UNDER TEST IE. BIT 0 FOR  
DEVICE 0 BIT 15 FOR DEVICE 15  
TO DISQUALIFY DEVICES:

7.2.1 IF DEVICE 0 IS TO BE DISQUALIFIED SIMPLY RESTART  
PROGRAM WITH SW00 =1 AND OMIT THE FIRST DEVICE.

7.2.2 IF HOWEVER, DEVICES 1 THRU 15 OR ANY COMBINATION THEREOF  
ARE TO BE DISQUALIFIED...LOAD THE LOCATION OF ACTREG:  
OBSERVE THE ACTIVE BITS (ACTIVE =1, NONACTIVE = 0)  
AND DEPOSIT 0 WHERE THOSE DEVICES ARE TO BE DISQUALIFIED

7.2.2.1 TO RESTART...LOAD 000200 IN SWR AND DEPRESS START....  
THE PROGRAM WILL CONTINUE WITH THE DEVICE IT WAS IN BEFORE HALTING.

7.2.2.2 .....OR .....LOAD 000200 WITH SW00 =1 AND DEPRESS START....  
ANSWER THE QUESTION :1ST DEVICE : ETC.....  
.....THE PROGRAM WILL CONTINUE WITH DEVICE 0

7.2.2.3 IF ALL DEVICES ARE DISQUALIFIED BY MISTAKE THE PROGRAM  
WILL TYPEOUT AN ERROR MESSAGE.....LOAD & START AT 000200

### 7.3 CABLE DELAYS

NOTE: EXTERNAL LOOP BACK TESTS ONLY (MODEM CABLE WITH H315 CONNECTOR ON)

7.3.1 TO PROVIDE SUFFICIENT DELAY FOR CLOCK SIGNAL OVER THE CABLE,  
LOCATION "HOLD:" MUST BE MODIFIED TO ACCOMODATE FOR FASTER MACHINES.  
PRESENTLY "HOLD:" =20 IS SUFFICIENT TIME ON AN 11/20 MACHINE.  
IF RUNNING ON AN 11/40 OR AN 11/45 "HOLD:" MUST BE PATCHED TO 40 .

BASICALLY DON'T TRY TO EXCEED 10K TO 12K RATE USING THE EIA DRIVERS

7.4 TO USE THE "XOR" TESTER, THE BRANCH AROUND THE "XOR"  
CODE MUST BE PATCHED TO A "NOP". (SEE LISTINGS FOR DETAILS)

8. DEFAULT PARAMETERS:  
1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- RXCSR: 160040

VECTOR ADDRESS- DURIV: 770

ARE YOU RUNNING MULTIPLE DEVICES ?- NO MULTD: 0

LAST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- LASTADD: 0

DU PRIORITY LEVEL- LEVEL 5 DUPRT: LEVEL 5

# OF SYNC CHARS SELECTED - 2 SYNCNO: 377

IS SEC XMIT JUMPER # 6 IN ?- YES SEXMIT: 377

IS SEC REC JUMPER # 5 IN ?- YES SEREC: 377

IS OPT CLR ENABLE JUMPER # 4 IN ?- YES OPTCLR: 377

DO YOU HAVE THE EXTERNAL MODEM BYPASS JUMPER  
CONNECTOR ON (H315)- YES JMRBY: 377

9. PROGRAM DESCRIPTION

9.1 THIS PROGRAM PERFORMS THE OFFLINE LOGIC BIT BANGING  
OF THE DEVICE  
SEE LISTING FOR DETAILS

10. FLOW CHARTS: RECEIVER FLOW, TRANSMITTER FLOW, TRANSMITTER & RECEIVER FLOW

11. LISTINGS

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

.ENABLE ABS

;DU11 DZDUA-D TAPE A  
;COPYRIGHT 1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754;STARTING PROCEDURE  
;LOAD PROGRAM  
;PRESS START  
;PROGRAM WILL TYPE "DU11 DZDUA-D TAPE A "  
;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED  
;AT THE END OF A PASS, PROGRAM WILL TYPE "END OF PASS TAPE A"  
;AND THEN RESUME TESTING

;SWITCH REGISTER OPTIONS

100000  
040000  
020000  
010000  
004000  
002000  
001000  
000400  
000100  
000040  
000020  
000010  
000004  
000002  
000001SW15=100000            ;=1,HALT ON ERROR  
SW14=40000             ;=1,LOOP ON CURRENT TEST  
SW13=20000             ;=1,INHIBIT ERROR TYPEOUT  
SW12=10000  
SW11=4000              ;=1,INHIBIT ITERATIONS  
SW10=2000              ;=1,ESCAPE TO NEXT TEST ON ERROR  
SW09=1000              ;=1,LOOP WITH CURRENT DATA  
SW08=400               ;=1,LOOP ON ERROR  
SW06=100  
SW05=40  
SW04=20  
SW03=10  
SW02=4  
SW01=2  
SW00=1  
  
;LOCK ON TEST SELECT  
;RESTART PROGRAM AT SELECTED TEST  
;RESELECT VECTOR AND CONTROL REGISTER  
;ADDRESS AFTER PROGRAM RESTART



```

620
621
622
623      000000      R0=%0      ;GENERAL REGISTER
624      000001      R1=%1      ;GENERAL REGISTER
625      000002      R2=%2      ;GENERAL REGISTER
626      000003      R3=%3      ;GENERAL REGISTER
627      000004      R4=%4      ;GENERAL REGISTER
628      000005      R5=%5      ;GENERAL REGISTER
629      000006      SP=%6      ;PROCESSOR STACK POINTER
630      000007      PC=%7      ;PROGRAM COUNTER
631
632      ;LOCATION EQUIVALENCIES
633
634      177570      DSWR=177570 ;HARDWARE SWITCH REGISTER LOC.
635      177570      DLIGHTS=177570 ;HARDWARE DISPLAY REGISTER LOC.
636      177776      PS=177776 ;PROCESSOR STATUS WORD
637      001100      STACK=1100 ;START OF PROCESSOR STACK
638
639      ;INSTRUCTION DEFINITIONS
640
641      005746      PUSH1SP=5746 ;DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
642      005726      POP1SP=5726 ;INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
643      010046      PUSHRO=10046 ;SAVE RO ON STACK =MOV RO, -(SP)
644      012600      POPRO=12600 ;RESTORE RO FROM STACK =MOV (SP)+, RO
645      024646      PUSH2SP=24646 ;DECREMENT STACK TWICE =CMP -(SP), -(SP)
646      022626      POP2SP=22626 ;INCREMENT STACK TWICE =CMP (SP)+, (SP)+
647      .EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL
648
649
650      100000      BIT15=100000
651      040000      BIT14=40000
652      020000      BIT13=20000
653      010000      BIT12=10000
654      004000      BIT11=4000
655      002000      BIT10=2000
656      001000      BIT9=1000
657      000400      BIT8=400
658      000200      BIT7=200
659      000100      BIT6=100
660      000040      BIT5=40
661      000020      BIT4=20
662      000010      BIT3=10
663      000004      BIT2=4
664      000002      BIT1=2
665      000001      BIT0=1
666
667      ;PROCESSOR LEVELS
668      000340      LEVEL7=340
669      000300      LEVEL6=300
670      000240      LEVEL5=240
671      000200      LEVEL4=200
672      000140      LEVEL3=140
673      000100      LEVEL2=100
674      000040      LEVEL1=040
675      000000      LEVEL0=000
  
```

```

676
677
678      100000
679      040000
680      020000
681      010000
682      004000
683      002000
684      001000
685      000400
686      000200
687      000100
688      000040
689      000020
690      000010
691      000004
692      000002
693      000001
694
695      100000
696      040000
697      020000
698      010000
699
700      001000
701      000400
702
703      030000
704      020000
705      000000
706      000000
707      002000
708      004000
709      006000
710      000000
711      001000
712      001400
713
714      100000
715      040000
716      020000
717      002000
718      000400
719      000200
720      000100
721      000040
722      000020
723      000010
724      000001
725
726      000000
727      004000
728      010000
729      014000
730

```

```

;REGISTER DEFINITIONS
;RXCSR BIT DEFINITIONS
DSC=BIT15      ;DATA SET CHANGE
RING=BIT14     ;RING
CTS=BIT13      ;CLR TO SEND
CARDET=BIT12   ;CARRIER DETECT
RECACT=BIT11   ;REC ACTIVE
SRD=BIT10      ;SEC REC DATA
CSR=BIT9       ;DATA SET RDY
STPSYN=BIT8    ;STRIP SYNC
RXDONE=BIT7    ;REC DONE
RINTEN=BIT6    ;REC INTR ENABLE
DSINTE=BIT5    ;DSC INTR ENABLE
SYNSCH=BIT4    ;SYNC SEARCH
STD=BIT3       ;SEC XMIT DATA
RTS=BIT2       ;REQ TO SEND
DTR=BIT1       ;DATA TERM RDY
VOID=BIT0

;RXDBUF BIT DEFINITIONS
RXERR=BIT15    ;REC ERROR
OVRUN=BIT14    ;OVERRUN
FRMERR=BIT13   ;FRAME ERROR
PARERR=BIT12   ;PARITY ERROR

;PARCSR BIT DEFINITIONS
PAREN=BIT9     ;PARITY ENABLE
EVPAR=BIT8     ;EVEN PARITY SENSE

;PARCSR WRD DEFINITIONS
SYNINT=30000   ;SYNC EXTERNAL MODE
SYNEXT=20000   ;SYNC INTERNAL MODE
ISYMOD=0       ;ISOC MODE
FIVE=0         ;WORD LENGTH 5 BITS
SIX=2000       ;WORD LENGTH 6 BITS
SEVEN=4000     ;WORD LENGTH 7 BITS
EIGHT=6000    ;WORD LENGTH 8 BITS
NOPAR=0        ;NO PARITY
ODDPAR=1000    ;ODD PARITY
EVEPAR=1400    ;EVEN PARITY

;TXCSR BIT DEFINITIONS
DNA=BIT15      ;DATA NOT AVAILABLE
MTDATA=BIT14   ;MAINT DATA
CLK=BIT13      ;CLK
BITW=BIT10     ;BIT WINDOW
MRESET=BIT8    ;MASTER RESET
TXDONE=BIT7    ;XMIT DONE
TXINTE=BIT6    ;XMIT INTR ENABLE
DNAINTE=BIT5   ;DNA INTR ENAB
SEND=BIT4      ;SEND
HDXEN=BIT3     ;HDX/FDX
BREAK=BIT0     ;BREAK

;TXCSR WRD DEFINITIONS
USER=0         ;USER MODE
MINT=4000      ;MAINT INT MODE
MEXT=10000     ;MAINT EXT MODE
SYSTST=14000  ;SYSTEM TEST MODE

;TRAPCATCHER FOR ILLEGAL INTERRUPTS

```

```

731                                     ;STANDARD INTERRUPT VECTORS
732
733
734                                     . = 24
735 000024 015152                       . PFAIL                       : POWER FAIL HANDLER
736 000026 000340                       340                          : SERVICE AT LEVEL 7
737 000030 014702                       . HLT                          : ERROR HANDLER
738 000032 000340                       340                          : SERVICE AT LEVEL 7
739 000034 014650                       . TRPSRV                       : GENERAL HANDLER DISPATCH SERVICE
740 000036 000340                       340                          : SERVICE AT LEVEL 7
741
742                                     ; SOFTWARE SWITCH REGISTER
743
744                                     . = 174
745 000174 000000                       DISPREG: .WORD 0              : SOFTWARE DISPLAY REG.
746 000176 000000                       SWREG:   .WORD 0              : SOFTWARE SWITCH REGISTER
747 000200 000167 001054                 JMP      .START                : GO TO START OF PROGRAM
748
749
750
751                                     . = 1100
752
753                                     ; INDIRECT POINTERS
754
755 001100 177570                       SWR:    177570                 : SWITCH REGISTER POINTER
756 001102 177570                       LIGHTS: 177570                : DISPLAY REGISTER POINTER
757 001104 177560                       TKCSR:  177560                : TELETYPE KEYBOARD CONTROL REGISTER
758 001106 177562                       TKDBR:  177562                : TELETYPE KEYBOARD DATA BUFFER
759 001110 177564                       TPCSR:  177564                : TELEPRINTER CONTROL REGISTER
760 001112 177566                       TPDBR:  177566                : TELEPRINTER DATA BUFFER
761
762                                     ; PROGRAM CONTROL PARAMETERS
763
764 001114 000000                       RTRN:   0                      : SCOPE ADDRESS FOR LOOP ON TEST
765 001116 000000                       NEXT:   0                      : ADDRESS OF NEXT TEST TO BE EXECUTED
766 001120 000000                       LOCK:   0                      : ADDRESS FOR LOCK ON CURRENT DATA
767 001122 000000                       ICOUNT: 0                      : NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
768 001124 000000                       LPCNT:  0                      : NUMBER OF ITERATIONS COMPLETED
769 001126 000000                       TSTNO:  0                      : NUMBER OF TEST IN PROGRESS
770 001130 000000                       PASCNT: 0                      : NUMBER OF PASSES COMPLETED
771 001132 000000                       ERRCNT: 0                      : TOTAL NUMBER OF ERRORS
772 001134 000000                       LSTERR: 0                      : PC OF LAST ERROR CALL
773
774                                     ; PROGRAM VARIABLES
775
776 001136 000020                       HOLD:   20                     : TEMPORARY STORAGE=DELAY TIME FOR CABLES
777 001140 000000                       SHIFT:  0                      : TEMPORARY STORAGE= # OF SHIFTS PER CHAR
778 001142 000000                       COUNT:  0                      : TEMPORARY STORAGE= # OF TIMES A CHAR WILL BE SENT
779 001144 000000                       TEMP1:  0                      : TEMPORARY STORAGE
780 001146 000000                       TEMP2:  0                      : TEMPORARY STORAGE
781 001150 000000                       TEMP3:  0                      : TEMPORARY STORAGE
782 001152 000000                       TEMP4:  0                      : TEMPORARY STORAGE
783 001154 000000                       TEMP5:  0                      : TEMPORARY STORAGE
784 001156 000000                       SAVR0:  0                      : R0 STORAGE
785 001160 000000                       SAVR1:  0                      : R1 STORAGE
786 001162 000000                       SAVR2:  0                      : R2 STORAGE

```

77	001164	000000
78	001166	000000
79	001170	000000
80	001172	000000
91	001174	000000

SAVR3:	0
SAVR4:	0
SAVR5:	0
SAVSP:	0
SAVPC:	0

:R3	STORAGE
:R4	STORAGE
:R5	STORAGE
:	STACK POINTER STORAGE
:	PROGRAM COUNTER STORAGE

```

792
793 001176 377 ;PROGRAM CONVERSATIONAL PARAMETERS
794 001177 377 ;# OF SYNC CHARS REQ'D FOR SYNC'ZATION
795 001200 377 ;SEC XMIT JUMPER "IN"
796 001201 377 ;SEC REC JUMPER "IN"
797 001202 000 ;OPTIONAL JUMPER CLR "IN"
798 001203 377 ;NO MULTIPLE DEVICE FLAG
799 ;EXTERNAL MODEM BYPASS JUMPER "IN"
800
801 ;PROGRAM MULTIPLE DEVICE PARAMETERS
802 001204 000000 BASEADD: 0 ;PROG CONTROLLED 1ST DEVICE ADDR
803 001206 000000 KEEPADD: 0 ;SAVED 1ST DEVICE ADDR
804 001210 000000 LASTADD: 0 ;LAST DEVICE RXCSR ADDR
805 001212 000000 BASEIV: 0 ;PROG CONTROLLED IV
806 001214 000000 KEEPIV: 0 ;SAVED INTR VECTOR
807 001216 000000 ACTREG: 0 ;ACTIVE REGISTER , , MODIFY THIS
808 ;LOCATION TO DISQUALIFY OR QUALIFY
809 ;DEVICES (1= RUN , 0= DON'T RUN)
810 001220 000000 ROTADD: 0 ;ROTATING POINTER FOR ACTREG..POINTS
811 ;TO DEVICE PRESENTLY UNDER TEST WHEN RUNNING MULTIPLE DE
812
813 ;PROGRAM CONTROL FLAGS
814
815 001222 000 INIFLG: .BYTE 0 ;PROGRAM INITIALIZATION FLAG
816 001223 000 STFLG: .BYTE 0 ;TEST START FLAG
817 001224 000 ERRFLG: .BYTE 0 ;ERROR OCCURED FLAG
818 001225 000 LOKFLG: .BYTE 0 ;LOCK ON CURRENT TEST FLAG
819
820 ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
821 ;POINTERS TO SUBROUTINES CAN BE FOUND
822 ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS
823
824 001226 .TRPTAB:
825 ;*****
826 ;*****
827 :04400
828 001226 013434 .SCOPE SCOPE=TRAP+0 ;CALL TO SCOPE LOOP AND ITERATION HANDLER
829 104401 .SCOPI SCOPI=TRAP+1 ;CALL TO LOOP ON CURRENT DATA HANDLER
830 001230 013620 .SCOPI TYPE=TRAP+2 ;CALL TO TELETYPE OUTPUT ROUTINE
831 104402 .TYPE INSTR=TRAP+3 ;CALL TO ASCII STRING INPUT ROUTINE
832 001232 013640 .INSTR INSTR=TRAP+4 ;CALL TO INPUT ERROR HANDLER
833 104403 .INSTER PARAM=TRAP+5 ;CALL TO NUMERICAL DATA INPUT ROUTINE
834 001234 013700 .PARAM SAVOS=TRAP+6 ;CALL TO REGISTER SAVE ROUTINE
835 104404 .INSTER RESOS=TRAP+7 ;CALL TO REGISTER RESTORE ROUTINE
836 001236 014016 .PARAM CONVRT=TRAP+10 ;CALL TO DATA OUTPUT ROUTINE
837 104405 .PARAM CNVRT=TRAP+11 ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF
838 001240 014050 .PARAM SETFLG=TRAP+12 ;CALL TO FLAG SET ROUTINE
839 104406
840 001242 014264 .SAVOS
841 104407
842 001244 014324 .RESOS
843 104410
844 001246 014356 .CONVRT
845 104411
846 001250 014362 .CNVRT
847 104412
    
```



```

848 001252 014602 .SETFLG
849 104413
850 001254 015316 .CKSWR CKSWR=TRAP+13 ;CALL TO ALLOW SWREG TO BE LOADED FROM TTY
851 104414
852 001256 015372 .CNTLU CNTLU=TRAP+14 ;CALL TO ALLOW LOADING OF SWREG FROM TTY
853 *****
854 *****
855 *****
856 ;PROGRAM INITIALIZATION
857 ;LOCK OUT INTERRUPTS
858 ;SET UP PROCESSOR STACK
859 ;SET UP POWER FAIL VECTOR
860 ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
861 ;TYPE TITLE MESSAGE
862
863 001260 012767 000340 176510 .START: MOV #340,PS ;LOCK OUT INTERRUPTS
864 001266 012706 001100 MOV #STACK,SP ;SET UP STACK
865 001272 012737 015152 000024 MOV #.PFAIL,2#24 ;SET UP POWER FAIL VECTOR
866 001300 005067 177620 CLR LPCNT ;CLEAR # OF ITERATION COMPLETED LOCATION
867 001304 105067 177713 CLR STFLG ;CLEAR START FLAG
868 001310 005067 177614 CLR PASCNT ;CLEAR PASS COUNT
869 001314 105067 177704 CLR ERRFLG ;CLEAR ERROR FLAG
870 001320 005067 177606 CLR ERRCNT ;CLEAR ERROR COUNT
871 001324 005067 177604 CLR LSTERR ;CLEAR LAST ERROR POINTER
872 001330 012767 000001 177570 MOV #1,TSTNO ;SET UP FOR TEST 1
873 001336 012767 001260 177550 MOV #.START,RTRN ;SET UP FOR POWER FAIL BEFORE
874 ;TESTING STARTS
875 001344 105767 177652 TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED
876 001350 001004 BNE ONCE
877 001352 104402 015472 TYPE ,MTITLE ;TYPE TITLE MESSAGE
878 001356 105167 177640 COMB INIFLG ;IF NOT SET FLAG AND DO
879 001362 012767 177570 177510 ONCE: MOV #DSWR,SWR ;RELOAD HARDWARE SWITCH REGISTER INTO POINTER
880 001370 012767 177570 177504 MOV #DLIGHTS,LIGHTS ;RELOAD HARDWARE DISPLAY REGISTER INTO POINTER
881 001376 013746 000006 MOV 2#6,-(SP) ;SAVE VECTORS
882 001402 013746 000004 MOV 2#4,-(SP)
883 001406 012737 001426 000004 MOV #64$,2#4 ;SET UP FOR TIMEOUT
884 001414 022777 177777 177456 CMP #-1,2#SWR ;REFERENCE HARDWARE SWITCH REGISTER
885 001422 001402 BEQ 65$
886 001424 000407 BR 66$
887 001426 022626 64$: CMP (SP)+,(SP)+ ;ADJUST STACK
888 001430 012767 000176 177442 65$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
889 001436 012767 000174 177436 MOV #DISPREG,LIGHTS ;POINT TO SOFT DISPLAY REG
890 001444 012637 000004 66$: MOV (SP)+,2#4 ;RESTORE VECTORS
891 001450 012637 000006 MOV (SP)+,2#6
892 001454 005737 000042 TST 2#42 ;UNDER MONITOR
893 001460 001005 BNE 67$
894 001462 022767 000176 177410 CMP #SWREG,SWR ;IS SWREG USED
895 001470 001001 BNE 67$
896 001472 104414 CNTLU
897 001474 032777 000001 177376 67$: BIT #SW00,2#SWR ;PESELECT VECTOR & CONTROL REG?
898 001502 001002 BNE 1$
899 001504 000167 000446 JMP .BEGIN
900 001510 012700 000300 1$: MOV #300,R0 ;RESTORE VECTOR AREA TO TRAPCATCHER
901 001514 012701 000302 MOV #302,R1 ;START AT LOCATION 300
902 001520 012702 000004 MOV #4,R2
903 001524 010110 2$: MOV R1,(R0)
    
```

904	001526	005011			CLR	(R1)	
905	001530	060200			ADD	R2,R0	
906	001532	060201			ADD	R2,R1	
907	001534	022701	001000		CMP	#1000,R1	;END AT LOCATION 776
908	001540	002771			BLT	2\$	
909	001542	104403			INSTR		;OUTPUT MESSAGE & GET INPUT STRING
910	001544	015546			MREGAD		;MESSAGE
911	001546	104405			PARAM		;CONVERT STRING
912	001550	160000			160000		;LOW LIMIT
913	001552	167776			167776		;HIGH LIMIT
914	001554	017402			DUBASE		;STORE AT THIS LOCATION
915	001556	001			.BYTE	1	;MASK
916	001557	001			.BYTE	1	;HOW MANY TIMES + 2
917	001560	016767	015616	177420	MOV	DUBASE,KEEPADD	;SAVE
918	001566	004767	015456		JSR	PC,DUADDR	
919	001572	016767	177410	177404	MOV	KEEPADD,BASEADD	;RESTORE FOR ROTATION
920	001600	104403			INSTR		;OUTPUT MESSAGE & GET INPUT STRING
921	001602	015524			MVECTO		;MESSAGE
922	001604	104405			PARAM		;CONVERT STRING
923	001606	000300			300		;LOW LIMIT
924	001610	000776			776		;HIGH LIMIT
925	001612	017724			DURIV		;STORE AT THIS LOCATION
926	001614	001			.BYTE	1	;MASK
927	001615	004			.BYTE	4	;HOW MANY TIMES + 2
928	001616	016767	016102	177370	MOV	DURIV,KEEPIV	;SAVE
929	001624	016767	016074	177360	MOV	DURIV,BASEIV	;SET UP FOR ROTATION
930	001632	104403			INSTR		;OUTPUT MESSAGE & GET INPUT STRING
931	001634	015627			MMULT		;MESSAGE
932	001636	104412			SETFLG		;SET FLAG BASED UPON INPUT STRING
933	001640	001202			MULTD		;THIS FLAG
934	001642	105767	177334		TSTB	MULTD	;ARE THERE MULTIPLE DEVICES
935							;ON THE SYSTEM ?
936	001646	100406			BMI	BBB	;YES,ASK NEXT QUESTION
937	001650	005067	177342		CLR	ACTREG	
938	001654	005067	177340		CLR	ROTADD	
939	001660	000167	000140		JMP	OUTMUL	;JUMP AROUND NEXT QUESTION
940	001664				BBB:		
941	001664	104403			INSTR		;OUTPUT MESSAGE & GET INPUT STRING
942	001666	015706			MLASTD		;MESSAGE
943	001670	104405			PARAM		;CONVERT STRING
944	001672	160000			160000		;LOW LIMIT
945	001674	167776			167776		;HIGH LIMIT
946	001676	001210			LASTADD		;STORE AT THIS LOCATION
947	001700	001			.BYTE	1	;MASK
948	001701	001			.BYTE	1	;HOW MANY TIMES + 2
949							;THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
950	001702	012767	000001	177310	1\$:	MOV	#1,ROTADD ;SET UP POINTER
951	001710	005067	177302		CLR	ACTREG	;CLR ACTIVE REGISTER
952	001714	056767	177300	177274	2\$:	BIS	ROTADD,ACTREG ;MAKE THIS DEVICE ACTIVE
953	001722	000241			CLC		
954	001724	006167	177270		ROL	ROTADD	;SET UP POINTER
955	001730	103421			BCS	3\$	;ARE YOU OUT OF RANGE ?
956	001732	062767	000010	177244	ADD	#10,BASEADD	;SET UP BASE ADDRESS
957	001740	026767	177244	177236	CMP	LASTADD,BASEADD	;IS THIS THE LAST DEVICE ?
958	001746	101362			BHI	2\$	;NO DO IT AGAIN
959	001750	056767	177244	177240	BIS	ROTADD,ACTREG	;THIS ASSUMES THAT THERE ARE AT

```

960                                     ;LEAST TWO DEVICES WHEN YOU ANSWER YES TO
961                                     ;MULTIPLE DEVICE QUESTION
962 001756 012767 000001 177234 4$: MOV #1,ROTADD ;SET UP FOR LATER USE IN END OF PASS ROUTINE
963 001764 016767 177216 177212 MOV KEEPADD,BASEADD ;DITTO
964 001772 000414 BR OUTMUL ;CONTINUE QUESTIONS
965 001774 016767 177206 177202 3$: MOV KEEPADD,BASEADD ;RESTORE
966 002002 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
967 002004 016071 MRANGE ;MESSAGE
968 002006 104405 PARAM ;CONVERT STRING
969 002010 160000 160000 ;LOW LIMIT
970 002012 167776 167776 ;HIGH LIMIT
971 002014 001210 LASTADD ;STORE AT THIS LOCATION
972 002016 001 .BYTE 1 ;MASK
973 002017 001 .BYTE 1 ;HOW MANY TIMES + 2
974 002020 000167 177656 JMP 1$ ;DO IT AGAIN
975 002024 OUTMUL:
976 002024 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
977 002026 016355 MLEVEL ;MESSAGE
978 002030 104405 PARAM ;CONVERT STRING
979 002032 000004 4 ;LOW LIMIT
980 002034 000007 7 ;HIGH LIMIT
981 002036 017244 DUPRT ;STORE AT THIS LOCATION
982 002040 000 .BYTE 0 ;MASK
983 002041 001 .BYTE 1 ;HOW MANY TIMES + 2
984 002042 004767 015126 JSR PC,DULEV
985                                     ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
986                                     ;BUFFER TO THE CHARACTERS "1" AND "2"
987                                     ;IF THE CHARACTER IS "1" CLEAR THE FLAG
988                                     ;IF THE CHARACTER IS "2" SET THE FLAG
989 AAA:
990 002046 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
991 002050 016402 MSYNC ;MESSAGE
992 002052 122767 000061 014754 3$: CMPB #'1,INBUF ;IS IT "1" ?
993 002060 001003 BNE 1$
994 002062 105067 177110 CLRB SYNCNO ;000
995 002066 000412 BR 4$
996 002070 122767 000062 014736 1$: CMPB #'2,INBUF ;IS IT "2" ?
997 002076 001004 BNE 2$
998 002100 112767 177777 177070 MOVB #-1,SYNCNO ;377
999 002106 000402 BR 4$
1000 002110 104404 2$: INSTR ;RETRY
1001 002112 000757 BR 3$
1002 002114 000240 4$: NOP
1003 002116 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1004 002120 016450 MWIRE6 ;MESSAGE
1005 002122 104412 SETFLG ;SET FLAG BASED UPON INPUT STRING
1006 002124 001177 SEXMIT ;THIS FLAG
1007 002126 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1008 002130 016516 MWIRE5 ;MESSAGE
1009 002132 104412 SETFLG ;SET FLAG BASED UPON INPUT STRING
1010 002134 001200 SEREC ;THIS FLAG
1011 002136 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1012 002140 016563 MWIRE4 ;MESSAGE
1013 002142 104412 SETFLG ;SET FLAG BASED UPON INPUT STRING
1014 002144 001201 OPTCLR ;THIS FLAG
1015 002146 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING

```

```

1016 002150 016637          MEXTJ          ;MESSAGE
1017 002152 104412          SETFLG        ;SET FLAG BASED UPON INPUT STRING
1018 002154 001203          JMRBY         ;THIS FLAG
1019
1020                          ;TEST START AND RESTART
1021
1022 002156 012767 000340 175612 .BEGIN: MOV      #340,PS          ;LOCK OUT INTERRUPTS
1023 002164 012706 001100          MOV      #STACK,SP      ;SET UP STACK
1024 002170 005737 000042          TST      #42           ;IS PROGRAM UNDER MONITOR CONTROL
1025 002174 001056          BNE      3$
1026 002176 105767 177000          TSTB     MULTD          ;DON'T ALLOW LOCK ON TEST IF RUNNING
1027                          ;MULTIPLE DEVICES
1028 002202 001407          BEQ      5$           ;IF NO TEST FOR LOCK ON TEST
1029 002204 016767 011404 011304          MOV      BRW,TTST       ;RESTORE NORMAL SCOPE LOOP
1030 002212 016767 011400 011300          MOV      BRX,TTST+2     ;DITTO
1031 002220 000444          BR       3$           ;JUMP AROUND IF YES
1032 002222 032777 000004 176650 5$: BIT      #BIT2,DSWR     ;CHECK FOR LOCK ON TEST
1033 002230 001416          BEQ      1$
1034 002232 104403          INSTR                     ;OUTPUT MESSAGE & GET INPUT STRING
1035 002234 016312          MLOCK                    ;MESSAGE
1036 002236 104412          SETFLG                    ;SET FLAG BASED UPON INPUT STRING
1037 002240 001225          LOKFLG                    ;THIS FLAG
1038 002242 105767 176757          TSTB     LOKFLG         ;IS LOCK ON TEST OPTION SELECTED
1039 002246 001407          BEQ      1$
1040 002250 012767 000240 011240          MOV      #NOP,TTST
1041 002256 012767 000240 011234          MOV      #NOP,TTST+2   ;SET UP TO LOCK
1042 002264 000406          BR       2$
1043 002266 016767 011322 011222 1$: MOV      BRW,TTST
1044 002274 016767 011316 011216          MOV      BRX,TTST+2   ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1045 002302 032777 000002 176570 2$: BIT      #SW01,DSWR   ;IF SW01=1, GET STARTING PC
1046 002310 001410          BEQ      3$
1047 002312 104403          INSTR                     ;OUTPUT MESSAGE & GET INPUT STRING
1048 002314 016277          MTSTPC                    ;MESSAGE
1049 002316 104405          PARAM                    ;CONVERT STRING
1050 002320 002350          TST1                      ;LOW LIMIT
1051 002322 012642          TLAST                    ;HIGH LIMIT
1052 002324 001114          RTRN                      ;STORE AT THIS LOCATION
1053 002326 001          .BYTE 1                   ;MASK
1054 002327 001          .BYTE 1                   ;HOW MANY TIMES + 2
1055 002330 000403          BR       4$
1056 002332 012767 002350 176554 3$: MOV      #TST1,RTRN    ;START AT TEST 1
1057 002340 104402 016273 4$: TYPE     MR             ;TYPE R
1058 002344 000177 176544          JMP      #RTRN          ;START TESTING
1059
1060                          ;;THIS TEST PROVES EXISTANCE OF DEVICE REGISTERS
1061
1062 002350 012767 000001 175550 TST1: MOV      #1,TSTNO        ;SAVE THIS
1063 002356 012767 002436 176532          MOV      #TST2,NEXT     ;GO TO THIS TEST WHEN THRU
1064 002364 012737 017642 000004          MOV      #TRPRG,#4      ;SETUP TRAPCATCHER
1065 002372 012737 000340 000006          MOV      #LEVEL7,#6
1066 002400 105277 015274          INCB     #RXCSCR         ;TEST THIS REG
1067 002404 000401          BR       64$           ;IF OK JMP AROUND HLT
1068 002406 104000          HLT                    ;CHECK DEVICE REG ADDRESSES
1069 002410 105277 015266 64$: INCB     #RXCSCR ;TEST UPPER BYTE THIS REGISTER
1070 002414 000401          BR       65$           ;IF OK JMP AROUND HLT
1071 002416 104000          HLT                    ;CHECK DEVICE REG ADDRESSES

```

```

1072 002420 012737 000006 000004 65$: MOV #6, @#4 ;RESTORE TRAPCATCHER
1073 002426 012737 000000 000006 MOV #0, @#6 ;
1074 002434 104400 SCOPE
1075 ;: THIS TEST PROVES EXISTANCE OF DEVICE REGISTERS
1076 ;:
1077 002436 012767 000002 176462 TST2: MOV #2, TSTNO ;SAVE THIS
1078 002444 012767 002524 176444 MOV #TST3, NEXT ;GO TO THIS TEST WHEN THRU
1079 002452 012737 017642 000004 MOV #TRPRÉG, @#4 ;SETUP TRAPCATCHER
1080 002460 012737 000340 000006 MOV #LEVEL7, @#6 ;
1081 002466 105277 015212 INCB @RXDBUF ;TEST THIS REG
1082 002472 000401 BR 64$ ;IF OK JMP AROUND HLT
1083 002474 104000 HLT ;CHECK DEVICE REG ADDRESSES
1084 002476 105277 015204 64$: INCB @HRXDBUF ;TEST UPPER BYTE THIS REGISTER
1085 002502 000401 BR 65$ ;IF OK JMP AROUND HLT
1086 002504 104000 HLT ;CHECK DEVICE REG ADDRESSES
1087 002506 012737 000006 000004 65$: MOV #6, @#4 ;RESTORE TRAPCATCHER
1088 002514 012737 000000 000006 MOV #0, @#6 ;
1089 002522 104400 SCOPE
1090 ;: THIS TEST PROVES EXISTANCE OF DEVICE REGISTERS
1091 ;:
1092 002524 012767 000003 176374 TST3: MOV #3, TSTNO ;SAVE THIS
1093 002532 012767 002612 176356 MOV #TST4, NEXT ;GO TO THIS TEST WHEN THRU
1094 002540 012737 017642 000004 MOV #TRPRÉG, @#4 ;SETUP TRAPCATCHER
1095 002546 012737 000340 000006 MOV #LEVEL7, @#6 ;
1096 002554 105277 015130 INCB @PARCSR ;TEST THIS REG
1097 002560 000401 BR 64$ ;IF OK JMP AROUND HLT
1098 002562 104000 HLT ;CHECK DEVICE REG ADDRESSES
1099 002564 105277 015122 64$: INCB @HPARCSR ;TEST UPPER BYTE THIS REGISTER
1100 002570 000401 BR 65$ ;IF OK JMP AROUND HLT
1101 002572 104000 HLT ;CHECK DEVICE REG ADDRESSES
1102 002574 012737 000006 000004 65$: MOV #6, @#4 ;RESTORE TRAPCATCHER
1103 002602 012737 000000 000006 MOV #0, @#6 ;
1104 002610 104400 SCOPE
1105 ;: THIS TEST PROVES EXISTANCE OF DEVICE REGISTERS
1106 ;:
1107 002612 012767 000004 176306 TST4: MOV #4, TSTNO ;SAVE THIS
1108 002620 012767 002700 176270 MOV #TST5, NEXT ;GO TO THIS TEST WHEN THRU
1109 002626 012737 017642 000004 MOV #TRPRÉG, @#4 ;SETUP TRAPCATCHER
1110 002634 012737 000340 000006 MOV #LEVEL7, @#6 ;
1111 002642 105277 015046 INCB @TXCSR ;TEST THIS REG
1112 002646 000401 BR 64$ ;IF OK JMP AROUND HLT
1113 002650 104000 HLT ;CHECK DEVICE REG ADDRESSES
1114 002652 105277 015040 64$: INCB @HTXCSR ;TEST UPPER BYTE THIS REGISTER
1115 002656 000401 BR 65$ ;IF OK JMP AROUND HLT
1116 002660 104000 HLT ;CHECK DEVICE REG ADDRESSES
1117 002662 012737 000006 000004 65$: MOV #6, @#4 ;RESTORE TRAPCATCHER
1118 002670 012737 000000 000006 MOV #0, @#6 ;
1119 002676 104400 SCOPE
1120 ;: THIS TEST PROVES EXISTANCE OF DEVICE REGISTERS
1121 ;:
1122 002700 012767 000005 176220 TST5: MOV #5, TSTNO ;SAVE THIS
1123 002706 012767 002766 176202 MOV #TST6, NEXT ;GO TO THIS TEST WHEN THRU
1124 002714 012737 017642 000004 MOV #TRPRÉG, @#4 ;SETUP TRAPCATCHER
1125 002722 012737 000340 000006 MOV #LEVEL7, @#6 ;
1126 002730 105277 014764 INCB @TXDBUF ;TEST THIS REG
1127 002734 000401 BR 64$ ;IF OK JMP AROUND HLT
  
```



```

1128 002736 104000          HLT                ;CHECK DEVICE REG. ADDRESSES
1129 002740 105277 014756 64$: INCB 3HTXDBUF      ;TEST UPPER BYTE THIS REGISTER
1130 002744 000401          BR 65$           ;IF OK JMP AROUND HLT
1131 002746 104000          HLT                ;CHECK DEVICE REG ADDRESSES
1132 002750 012737 000006 000004 65$: MOV #6,2#4      ;RESTORE TRAPCATCHER
1133 002756 012737 000000 000006 MOV #0,2#6      ;
1134 002764 104400          SCOPE
1135                      ;;BUS DRIVER TEST
1136                      ;;
1137 002766 012767 000006 176132 TST6: MOV #6,TSTNO      ;SAVE THIS
1138 002774 012767 003016 176114 MOV #TST7,NEXT   ;GO TO THIS TEST WHEN THRU
1139 003002 022777 177777 014710 CMP #177777,2TXDBUF
1140 003010 001401          BEQ +4
1141 003012 104000          HLT                ;READING TXDBUF SHOULD BE ALL 1'S
1142 003014 104400          SCOPE
1143                      ;;THIS TEST PERFORMS MASTER RESET TESTING &
1144                      ;;TESTING OF READ/WRITE BIT DTR
1145                      ;;
1146 003016 012767 000007 176102 TST7: MOV #7,TSTNO      ;SAVE THIS
1147 003024 012767 003146 176064 MOV #TST8,NEXT   ;GO TO THIS TEST WHEN THRU
1148 003032 052777 000002 014640 BIS #DTR,2RXCSR  ;SET THIS BIT
1149 003040 032777 000002 014632 BIT #DTR,2RXCSR  ;TEST THIS BIT
1150 003046 001001          BNE 64$          ;BR IF "1"
1151 003050 104000          HLT                ;THIS BIT SHOULD BE SET
1152 003052          64$:
1153 003052 042777 000002 014620 BIC #DTR,2RXCSR  ;CLR THIS BIT
1154 003060 032777 000002 014612 BIT #DTR,2RXCSR  ;TEST THIS BIT
1155 003066 001401          BEQ 65$          ;BR IF "0"
1156 003070 104000          HLT                ;THIS BIT SHOULD BE CLR
1157 003072          65$:
1158                      ;NOW SET THIS BIT
1159 003072 052777 000002 014600 BIS #DTR,2RXCSR  ;
1160 003100 052777 000400 014606 BIS #MRESET,2TXCSR ;MASTER RESET
1161                      ;;CHECK EXISTANCE OF OPTIONAL CLEAR JUMPER
1162                      ;;
1163 003106 105767 176067 TST8: TSTB OPTCLR      ;TEST FLAG
1164 003112 100006          BPL 1$           ;OPTIONAL CLR JUMPER IS NOT IN
1165 003114 032777 000002 014556 BIT #DTR,2RXCSR  ;TEST THIS BIT
1166 003122 001401          BEQ 66$          ;BR IF "0"
1167 003124 104000          HLT                ;CHECK OUT MASTER RESET LOGIC
1168 003126          66$:
1169 003126 000405          BR 2$           ;JMP AROUND
1170 003130 032777 000002 014542 1$: BIT #DTR,2RXCSR  ;TEST THIS BIT
1171 003136 001001          BNE 67$          ;BR IF "1"
1172 003140 104000          HLT                ;CHECK OUT OPTIONAL CLR JUMPER
1173 003142          67$:
1174 003142 000240          2$: NOP
1175 003144 104400          SCOPE
1176                      ;;THIS TEST PERFORMS MASTER RESET TESTING &
1177                      ;;TESTING OF READ/WRITE BIT RTS
1178                      ;;
1179 003146 012767 000010 175752 TST8: MOV #8,TSTNO      ;SAVE THIS
1180 003154 012767 003276 175734 MOV #TST9,NEXT   ;GO TO THIS TEST WHEN THRU
1181 003162 052777 000004 014510 BIS #RTS,2RXCSR  ;SET THIS BIT
1182 003170 032777 000004 014502 BIT #RTS,2RXCSR  ;TEST THIS BIT
1183 003176 001001          BNE 64$          ;BR IF "1"

```

```

1184 003200 104000          HLT          ;THIS BIT SHOULD BE SET
1185 003202                64$:          BIC      #RTS,DRXCSR ;CLR THIS BIT
1186 003202 042777 000004 014470 BIC      #RTS,DRXCSR ;TEST THIS BIT
1187 003210 032777 000004 014462 BIT      #RTS,DRXCSR ;TEST THIS BIT
1188 003216 001401          BEQ      65$      ;BR IF "0"
1189 003220 104000          HLT          ;THIS BIT SHOULD BE CLR
1190 003222                65$:          ;NOW SET THIS BIT
1191 003222                BIS      #RTS,DRXCSR
1192 003222 052777 000004 014450 BIS      #MRESET,DTXCSR ;MASTER RESET
1193 003230 052777 000400 014456 ;;CHECK EXISTANCE OF OPTIONAL CLEAR JUMPER
1194
1195
1196 003236 105767 175737     TSTB     OPTCLR      ;TEST FLAG
1197 003242 100006          BPL      1$          ;OPTIONAL CLR JUMPER IS NOT IN
1198 003244 032777 000004 014426 BIT      #RTS,DRXCSR ;TEST THIS BIT
1199 003252 001401          BEQ      66$      ;BR IF "0"
1200 003254 104000          HLT          ;CHECK OUT MASTER RESET LOGIC
1201 003256                66$:          BR      2$          ;JMP AROUND
1202 003256 000405          BR      2$          ;TEST THIS BIT
1203 003260 032777 000004 014412 1$:      BIT      #RTS,DRXCSR ;TEST THIS BIT
1204 003266 001001          BNE     67$      ;BR IF "1"
1205 003270 104000          HLT          ;CHECK OUT OPTIONAL CLR JUMPER
1206 003272                67$:          NOP
1207 003272 000240                2$:      SCOPE
1208 003274 104400          ;;THIS TEST PERFORMS MASTER RESET TESTING &
1209          ;;TESTING OF READ/WRITE BIT STD
1210
1211
1212 003276 012767 000011 175622 TST9:  MOV     #9,TSTNO    ;SAVE THIS
1213 003304 012767 003426 175604 MOV     #TST10,NEXT ;GO TO THIS TEST WHEN THRU
1214 003312 052777 000010 014360 BIS     #STD,DRXCSR ;SET THIS BIT
1215 003320 032777 000010 014352 BIT     #STD,DRXCSR ;TEST THIS BIT
1216 003326 001001          BNE     64$      ;BR IF "1"
1217 003330 104000          HLT          ;THIS BIT SHOULD BE SET
1218 003332                64$:          BIC     #STD,DRXCSR ;CLR THIS BIT
1219 003332 042777 000010 014340 BIC     #STD,DRXCSR ;TEST THIS BIT
1220 003340 032777 000010 014332 BIT     #STD,DRXCSR ;TEST THIS BIT
1221 003346 001401          BEQ     65$      ;BR IF "0"
1222 003350 104000          HLT          ;THIS BIT SHOULD BE CLR
1223 003352                65$:          ;NOW SET THIS BIT
1224 003352                BIS     #STD,DRXCSR
1225 003352 052777 000010 014320 BIS     #MRESET,DTXCSR ;MASTER RESET
1226 003360 052777 000400 014326 ;;CHECK EXISTANCE OF OPTIONAL CLEAR JUMPER
1227
1228
1229 003366 105767 175607     TSTB     OPTCLR      ;TEST FLAG
1230 003372 100006          BPL     1$          ;OPTIONAL CLR JUMPER IS NOT IN
1231 003374 032777 000010 014276 BIT     #STD,DRXCSR ;TEST THIS BIT
1232 003402 001401          BEQ     66$      ;BR IF "0"
1233 003404 104000          HLT          ;CHECK OUT MASTER RESET LOGIC
1234 003406                66$:          BR     2$          ;JMP AROUND
1235 003406 000405          BR     2$          ;TEST THIS BIT
1236 003410 032777 000010 014262 1$:      BIT     #STD,DRXCSR ;TEST THIS BIT
1237 003416 001001          BNE     67$      ;BR IF "1"
1238 003420 104000          HLT          ;CHECK OUT OPTIONAL CLR JUMPER
1239 003422                67$:
    
```

```

1240 003422 000240          2$:  NOP
1241 003424 104400          SCOPE
1242                               ;; THIS TEST PERFORMS MASTER RESET TESTING &
1243                               ;; TESTING OF READ/WRITE BIT SYNSCH
1244                               ;;
1245 003426 012767 000012 175472 TST10: MOV    #10,TSTNO      ;SAVE THIS
1246 003434 012767 003532 175454      MOV    #TST11,NEXT      ;GO TO THIS TEST WHEN THRU
1247 003442 052777 000020 014230      BIS    #SYNSCH,ARXCSR   ;SET THIS BIT
1248 003450 032777 000020 014222      BIT    #SYNSCH,ARXCSR   ;TEST THIS BIT
1249 003456 001001                BNE    64$             ;BR IF "1"
1250 003460 104000                HLT                    ;THIS BIT SHOULD BE SET
1251 003462
1252 003462 042777 000020 014210      64$:  BIC    #SYNSCH,ARXCSR ;CLR THIS BIT
1253 003470 032777 000020 014202      BIT    #SYNSCH,ARXCSR   ;TEST THIS BIT
1254 003476 001401                BEQ    65$             ;BR IF "0"
1255 003500 104000                HLT                    ;THIS BIT SHOULD BE CLR
1256 003502
1257                               :NOW SET THIS BIT
1258 003502 052777 000020 014170      BIS    #SYNSCH,ARXCSR   ;
1259 003510 052777 000400 014176      BIS    #MRESET,ARXCSR  ;MASTER RESET
1260 003516 032777 000020 014154      BIT    #SYNSCH,ARXCSR   ;TEST THIS BIT
1261 003524 001401                BEQ    66$             ;BR IF "0"
1262 003526 104000                HLT                    ;CHECK OUT MASTER RESET LOGIC
1263 003530
1264 003530 104400          66$:  SCOPE
1265                               ;; THIS TEST PERFORMS MASTER RESET TESTING &
1266                               ;; TESTING OF READ/WRITE BIT DSINTE
1267                               ;;
1268 003532 012767 000013 175366 TST11: MOV    #11,TSTNO      ;SAVE THIS
1269 003540 012767 003636 175350      MOV    #TST12,NEXT      ;GO TO THIS TEST WHEN THRU
1270 003546 052777 000040 014124      BIS    #DSINTE,ARXCSR   ;SET THIS BIT
1271 003554 032777 000040 014116      BIT    #DSINTE,ARXCSR   ;TEST THIS BIT
1272 003562 001001                BNE    64$             ;BR IF "1"
1273 003564 104000                HLT                    ;THIS BIT SHOULD BE SET
1274 003566
1275 003566 042777 000040 014104      64$:  BIC    #DSINTE,ARXCSR ;CLR THIS BIT
1276 003574 032777 000040 014076      BIT    #DSINTE,ARXCSR   ;TEST THIS BIT
1277 003602 001401                BEQ    65$             ;BR IF "0"
1278 003604 104000                HLT                    ;THIS BIT SHOULD BE CLR
1279 003606
1280                               :NOW SET THIS BIT
1281 003606 052777 000040 014064      BIS    #DSINTE,ARXCSR   ;
1282 003614 052777 000400 014072      BIS    #MRESET,ARXCSR  ;MASTER RESET
1283 003622 032777 000040 014050      BIT    #DSINTE,ARXCSR   ;TEST THIS BIT
1284 003630 001401                BEQ    66$             ;BR IF "0"
1285 003632 104000                HLT                    ;CHECK OUT MASTER RESET LOGIC
1286 003634
1287 003634 104400          66$:  SCOPE
1288                               ;; THIS TEST PERFORMS MASTER RESET TESTING &
1289                               ;; TESTING OF READ/WRITE BIT RINTEN
1290                               ;;
1291 003636 012767 000014 175262 TST12: MOV    #12,TSTNO      ;SAVE THIS
1292 003644 012767 003742 175244      MOV    #TST13,NEXT      ;GO TO THIS TEST WHEN THRU
1293 003652 052777 000100 014020      BIS    #RINTEN,ARXCSR   ;SET THIS BIT
1294 003660 032777 000100 014012      BIT    #RINTEN,ARXCSR   ;TEST THIS BIT
1295 003666 001001                BNE    64$             ;BR IF "1"

```

```

1296 003670 104000          HLT          ;THIS BIT SHOULD BE SET
1297 003672          64$:          BIC          #RINTEN,@RXCSR ;CLR THIS BIT
1298 003672 042777 000100 014000 BIC          #RINTEN,@RXCSR ;TEST THIS BIT
1299 003700 032777 000100 013772 BIT          #RINTEN,@RXCSR ;BR IF "0"
1300 003706 001401          BEQ          65$          ;THIS BIT SHOULD BE CLR
1301 003710 104000          HLT
1302 003712          65$:          :NOW SET THIS BIT
1303 1303          :NOW SET THIS BIT
1304 003712 052777 000100 013760 BIS          #RINTEN,@RXCSR ;MASTER RESET
1305 003720 052777 000400 013766 BIS          #MRESET,@TXCSR ;TEST THIS BIT
1306 003726 032777 000100 013744 BIT          #RINTEN,@RXCSR ;BR IF "0"
1307 003734 001401          BEQ          66$          ;CHECK OUT MASTER RESET LOGIC
1308 003736 104000          HLT
1309 003740          66$:          SCOPE
1310 003740 104400          ;; THIS TEST PERFORMS MASTER RESET TESTING &
1311          ;; TESTING OF READ/WRITE BIT STPSYN
1312          ;;
1313          ;;
1314 003742 012767 000015 175156 TST13: MOV          #13,TSTNO      ;SAVE THIS
1315 003750 012767 004046 175140 MOV          #TST14,NEXT    ;GO TO THIS TEST WHEN THRU
1316 003756 052777 000400 013714 BIS          #STPSYN,@RXCSR ;SET THIS BIT
1317 003764 032777 000400 013706 BIT          #STPSYN,@RXCSR ;TEST THIS BIT
1318 003772 001001          BNE          64$          ;BR IF "1"
1319 003774 104000          HLT          ;THIS BIT SHOULD BE SET
1320 003776          64$:          BIC          #STPSYN,@RYCSR ;CLR THIS BIT
1321 003776 042777 000400 013674 BIC          #STPSYN,@RXCSR ;TEST THIS BIT
1322 004004 032777 000400 013666 BIT          #STPSYN,@RXCSR ;BR IF "0"
1323 004012 001401          BEQ          65$          ;THIS BIT SHOULD BE CLR
1324 004014 104000          HLT
1325 004016          65$:          :NOW SET THIS BIT
1326 1326          :NOW SET THIS BIT
1327 004016 052777 000400 013654 BIS          #STPSYN,@RXCSR ;MASTER RESET
1328 004024 052777 000400 013662 BIS          #MRESET,@TXCSR ;TEST THIS BIT
1329 004032 032777 000400 013640 BIT          #STPSYN,@RXCSR ;BR IF "0"
1330 004040 001401          BEQ          66$          ;CHECK OUT MASTER RESET LOGIC
1331 004042 104000          HLT
1332 004044          66$:          SCOPE
1333 004044 104400          ;; THIS TEST PERFORMS MASTER RESET TESTING &
1334          ;; TESTING OF READ/WRITE BIT BREAK
1335          ;;
1336          ;;
1337 004046 012767 000016 175052 TST14: MOV          #14,TSTNO      ;SAVE THIS
1338 004054 012767 004152 175034 MOV          #TST15,NEXT    ;GO TO THIS TEST WHEN THRU
1339 004062 052777 000001 013624 BIS          #BREAK,@TXCSR  ;SET THIS BIT
1340 004070 032777 000001 013616 BIT          #BREAK,@TXCSR  ;TEST THIS BIT
1341 004076 001001          BNE          64$          ;BR IF "1"
1342 004100 104000          HLT          ;THIS BIT SHOULD BE SET
1343 004102          64$:          BIC          #BREAK,@TXCSR ;CLR THIS BIT
1344 004102 042777 000001 013604 BIC          #BREAK,@TXCSR ;TEST THIS BIT
1345 004110 032777 000001 013576 BIT          #BREAK,@TXCSR ;BR IF "0"
1346 004116 001401          BEQ          65$          ;THIS BIT SHOULD BE CLR
1347 004120 104000          HLT
1348 004122          65$:          :NOW SET THIS BIT
1349 1349          :NOW SET THIS BIT
1350 004122 052777 000001 013564 BIS          #BREAK,@TXCSR ;MASTER RESET
1351 004130 052777 000400 013556 BIS          #MRESET,@TXCSR
    
```



```

1408 004376 052777 000040 013310      BIS      #DNAINTE,@TXCSR ;SET THIS BIT
1409 004404 032777 000040 013302      BIT      #DNAINTE,@TXCSR ;TEST THIS BIT
1410 004412 001001                BNE      64$ ;BR IF "1"
1411 004414 104000                HLT
1412 004416                64$:
1413 004416 042777 000040 013270      BIC      #DNAINTE,@TXCSR ;CLR THIS BIT
1414 004424 032777 000040 013262      BIT      #DNAINTE,@TXCSR ;TEST THIS BIT
1415 004432 001401                BEQ      65$ ;BR IF "0"
1416 004434 104000                HLT
1417 004436                65$:
1418                ;NOW SET THIS BIT
1419 004436 052777 000040 013250      BIS      #DNAINTE,@TXCSR
1420 004444 052777 000040 013242      BIS      #MRESET,@TXCSR ;MASTER RESET
1421 004452 032777 000040 013234      BIT      #DNAINTE,@TXCSR ;TEST THIS BIT
1422 004460 001401                BEQ      66$ ;BR IF "0"
1423 004462 104000                HLT
1424 004464                66$:
1425 004464 104400                SCOPE
1426                ;;THIS TEST PERFORMS MASTER RESET TESTING &
1427                ;;TESTING OF READ/WRITE BIT TXINTE
1428                ;;
1429 004466 012767 000022 174432      TST18:  MOV      #18,TSTNO ;SAVE THIS
1430 004474 012767 004572 174414      MOV      #TST19,NEXT ;GO TO THIS TEST WHEN THRU
1431 004502 052777 000100 013204      BIS      #TXINTE,@TXCSR ;SET THIS BIT
1432 004510 032777 000100 013176      BIT      #TXINTE,@TXCSR ;TEST THIS BIT
1433 004516 001001                BNE      64$ ;BR IF "1"
1434 004520 104000                HLT
1435 004522                64$:
1436 004522 042777 000100 013164      BIC      #TXINTE,@TXCSR ;CLR THIS BIT
1437 004530 032777 000100 013156      BIT      #TXINTE,@TXCSR ;TEST THIS BIT
1438 004536 001401                BEQ      65$ ;BR IF "0"
1439 004540 104000                HLT
1440 004542                65$:
1441                ;NOW SET THIS BIT
1442 004542 052777 000100 013144      BIS      #TXINTE,@TXCSR
1443 004550 052777 000400 013136      BIS      #MRESET,@TXCSR ;MASTER RESET
1444 004556 032777 000100 013130      BIT      #TXINTE,@TXCSR ;TEST THIS BIT
1445 004564 001401                BEQ      66$ ;BR IF "0"
1446 004566 104000                HLT
1447 004570                66$:
1448 004570 104400                SCOPE
1449                ;;TEST MAINT MODE BIT 0
1450                ;;
1451                ;;THIS TEST PERFORMS MASTER RESET TESTING &
1452                ;;TESTING OF READ/WRITE BIT BIT11
1453                ;;
1454 004572 012767 000023 174326      TST19:  MOV      #19,TSTNO ;SAVE THIS
1455 004600 012767 004676 174310      MOV      #TST20,NEXT ;GO TO THIS TEST WHEN THRU
1456 004606 052777 004000 013100      BIS      #BIT11,@TXCSR ;SET THIS BIT
1457 004614 032777 004000 013072      BIT      #BIT11,@TXCSR ;TEST THIS BIT
1458 004622 001001                BNE      64$ ;BR IF "1"
1459 004624 104000                HLT
1460 004626                64$:
1461 004626 042777 004000 013060      BIC      #BIT11,@TXCSR ;CLR THIS BIT
1462 004634 032777 004000 013052      BIT      #BIT11,@TXCSR ;TEST THIS BIT
1463 004642 001401                BEQ      65$ ;BR IF "0"

```

```

1464 004644 104000
1465 004646
1466
1467 004646 052777 004000 013040
1468 004654 052777 000400 013032
1469 004662 032777 004000 013024
1470 004670 001401
1471 004672 104000
1472 004674
1473 004674 104400
1474
1475
1476
1477
1478
1479 004676 012767 000024 174222 TST20: MOV #20,TSTNO ;SAVE THIS
1480 004704 012767 005002 174204 MOV #TST21,NEXT ;GO TO THIS TEST WHEN THRU
1481 004712 052777 010000 012774 BIS #BIT12,@TXCSR ;SET THIS BIT
1482 004720 032777 010000 012766 BIT #BIT12,@TXCSR ;TEST THIS BIT
1483 004726 001001 BNE 64$ ;BR IF "1"
1484 004730 104000 HLT ;THIS BIT SHOULD BE SET
1485 004732
1486 004732 042777 010000 012754 64$: BIC #BIT12,@TXCSR ;CLR THIS BIT
1487 004740 032777 010000 012746 BIT #BIT12,@TXCSR ;TEST THIS BIT
1488 004746 001401 BEQ 65$ ;BR IF "0"
1489 004750 104000 HLT ;THIS BIT SHOULD BE CLR
1490 004752
1491
1492 004752 052777 010000 012734 65$: ;NOW SET THIS BIT
1493 004760 052777 000400 012726 BIS #BIT12,@TXCSR
1494 004766 032777 010000 012720 BIS #MRESET,@TXCSR ;MASTER RESET
1495 004774 001401 BIT #BIT12,@TXCSR ;TEST THIS BIT
1496 004776 104000 BEQ 66$ ;BR IF "0"
1497 005000 HLT ;CHECK OUT MASTER RESET LOGIC
1498 005000 104400
1499
1500
1501
1502 005002 012767 000025 174116 TST21: MOV #21,TSTNO ;SAVE THIS
1503 005010 012767 005106 174100 MOV #TST22,NEXT ;GO TO THIS TEST WHEN THRU
1504 005016 052777 020000 012670 BIS #CLK,@TXCSR ;SET THIS BIT
1505 005024 032777 020000 012662 BIT #CLK,@TXCSR ;TEST THIS BIT
1506 005032 001001 BNE 64$ ;BR IF "1"
1507 005034 104000 HLT ;THIS BIT SHOULD BE SET
1508 005036
1509 005036 042777 020000 012650 64$: BIC #CLK,@TXCSR ;CLR THIS BIT
1510 005044 032777 020000 012642 BIT #CLK,@TXCSR ;TEST THIS BIT
1511 005052 001401 BEQ 65$ ;BR IF "0"
1512 005054 104000 HLT ;THIS BIT SHOULD BE CLR
1513 005056
1514
1515 005056 052777 020000 012630 65$: ;NOW SET THIS BIT
1516 005064 052777 000400 012622 BIS #CLK,@TXCSR
1517 005072 032777 020000 012614 BIS #MRESET,@TXCSR ;MASTER RESET
1518 005100 001401 BIT #CLK,@TXCSR ;TEST THIS BIT
1519 005102 104000 BEQ 66$ ;BR IF "0"
HLT ;CHECK OUT MASTER RESET LOGIC
    
```



```

1520 005104
1521 005104 104400
1522
1523
1524
1525 005106 012767 000026 174012 TST22: MOV #22,TSTNO ;SAVE THIS
1526 005114 012767 005212 173774 MOV #TST23,NEXT ;GO TO THIS TEST WHEN THRU
1527 005122 052777 040000 012564 BIS #MTDATA,@TXCSR ;SET THIS BIT
1528 005130 032777 040000 012556 BIT #MTDATA,@TXCSR ;TEST THIS BIT
1529 005136 001001 BNE 64$ ;BR IF "1"
1530 005140 104000 HLT ;THIS BIT SHOULD BE SET
1531 005142
1532 005142 042777 040000 012544 64$: BIC #MTDATA,@TXCSR ;CLR THIS BIT
1533 005150 032777 040000 012536 BIT #MTDATA,@TXCSR ;TEST THIS BIT
1534 005156 001401 BEQ 65$ ;BR IF "0"
1535 005160 104000 HLT ;THIS BIT SHOULD BE CLR
1536 005162
1537 :NOW SET THIS BIT
1538 005162 052777 040000 012524 BIS #MTDATA,@TXCSR
1539 005170 052777 000400 012516 BIS #MRESET,@TXCSR ;MASTER RESET
1540 005176 032777 040000 012510 BIT #MTDATA,@TXCSR ;TEST THIS BIT
1541 005204 001401 BEQ 66$ ;BR IF "0"
1542 005206 104000 HLT ;CHECK OUT MASTER RESET LOGIC
1543 005210
1544 005210 104400 66$: SCOPE
1545 ::THIS TEST VERIFYS THAT INIT (RESET) CLEARS BITS IN THE
1546 ::RXCSR & TXCSR
1547
1548 005212 012767 000027 173706 TST23: MOV #23,TSTNO ;SAVE THIS
1549 005220 012767 005322 173670 MOV #TST24,NEXT ;GO TO THIS TEST WHEN THRU
1550 005226 012777 177777 012444 MOV #177777,@RXCSR ;SET ALL POSSIBLE BITS
1551 005234 012777 177777 012452 MOV #177777,@TXCSR ;DITTO
1552 005242 000005 RESET
1553 005244 012767 000340 172524 MOV #LEVEL7,PS ;RESTORE NON INTERRUPT STATUS
1554 005252 017701 012422 MOV @RXCSR,R1 ;SAVE
1555 005256 017702 012432 MOV @TXCSR,R2 ;SAVE
1556 005262 005767 173713 TSTB OPTCLR ;IS THE OPTIONAL CLR JUMPER ON ?
1557 005266 100402 BMI 1$ ;YES
1558 005270 042701 000016 BIC #16,R1 ;CLR THE NON RESETABLE BITS
1559 005274 042701 073000 1$: BIC #073000,R1 ;CLR ALL NON-CLEARABLE BITS
1560 005300 005701 TST R1 ;ARE THEY ALL 0 ?
1561 005302 001401 BEQ .+4
1562 005304 104000 HLT ;ALL SPECIFIED BITS SHOULD BE CLEAR
1563 005306 042702 002200 BIC #002200,R2 ;CLEAR ALL NON-CLEARABLE BITS
1564 005312 005702 TST R2 ;ARE THEY ALL 0 ?
1565 005314 001401 BEQ .+4
1566 005316 104000 HLT ;ALL SPECIFIED BITS SHOULD BE CLEAR
1567 005320 104400 SCOPE
1568 ::THIS TEST PERFORMS MASTER RESET TESTING &
1569 ::TESTING OF WRITE ONLY BIT MRESET
1570
1571 005322 012767 000030 173576 TST24: MOV #24,TSTNO ;SAVE THIS
1572 005330 012767 005400 173560 MOV #TST25,NEXT ;GO TO THIS TEST WHEN THRU
1573 005336 052777 000400 012350 BIS #MRESET,@TXCSR ;TRY TO SET THIS BIT
1574 005344 032777 000400 012342 BIT #MRESET,@TXCSR ;TEST THIS BIT
1575 005352 001401 BEQ 64$ ;BR IF "0"

```

```

1576 005354 104000          HLT          ;THIS BIT SHOULD NOT BE SET
1577 005356                64$:          ;
1578 005356 052777 000400 012330  BIS      #MRESET,@TXCSR ;MASTER RESET
1579 005364 032777 000400 012322  BIT      #MRESET,@TXCSR ;TEST THIS BIT
1580 005372 001401          BEQ      65$:          ;BR IF "0"
1581 005374 104000          HLT          ;THIS BIT SHOULD NOT BE SET
1582 005376                65$:          ;
1583                                ;CHECK MASTER RESET LOGIC
1584 005376 104400          SCOPE
1585                                ;;THIS TEST VERIFYS THAT THE RXCSR & TXCSR CAN BE BYTE ADDRESSED (DATOB)
1586                                ;
1587 005400 012767 000031 173520  TST25:  MOV      #25,TSTNO          ;SAVE THIS
1588 005406 012767 005566 173502  MOV      #TST26,NEXT          ;GO TO THIS TEST WHEN THRU
1589 005414 052777 000400 012272  BIS      #MRESET,@TXCSR      ;MASTER RESET
1590 005422 105767 173553  TSTB     OPTCLR          ;IS THE OPTIONAL CLR JUMPER ON ?
1591 005426 100405          BMI      1$:          ;YES
1592 005430 012777 000000 012242  MOV      #0,@RXCSR          ;CLR OUT NON RESETABLE BITS
1593 005436 005777 012236  TST      @RXCSR          ;CLR OUT DSC BY READING RXCSR
1594 005442 152777 000001 012232  1$:      BISB     #BIT0,@RXCSR          ;SET STRIP SYNC UPPER BYTE
1595 005450 017701 012224  MOV      @RXCSR,R1          ;SAVE RXCSR
1596 005454 022701 000400  CMP      #400,R1          ;TEST RXCSR
1597 005460 001401          BEQ      .+4
1598 005462 104000          HLT          ;ONLY STRIP SYNC SHOULD BE SET
1599 005464 105077 012210  CLRB     @RXCSR          ;CLR LOWER JYTE
1600 005470 017701 012204  MOV      @RXCSR,R1          ;SAVE RXCSR
1601 005474 022701 000400  CMP      #400,R1          ;TEST RXCSR
1602 005500 001401          BEQ      .+4
1603 005502 104000          HLT          ;ONLY STRIP SYNC SHOULD BE SET
1604 005504 052777 000400 012202  BIS      #MRESET,@TXCSR      ;MASTER RESET
1605 005512 152777 000040 012176  BISB     #BITS,@TXCSR          ;SET MAINT CLK UPPER BYTE
1606 005520 017701 012170  MOV      @TXCSR,R1          ;SAVE TXCSR
1607 005524 042701 002000  BIC      #BITW,R1          ;CLR BIT WINDOW (DEPENDENT
1608                                ;ON H315 CONNECTOR EXISTANCE)
1609 005530 022701 020200  CMP      #20200,R1          ;TEST TXCSR
1610 005534 001401          BEQ      .+4
1611 005536 104000          HLT          ;ONLY MAINT CLK BIT & TXDONE SHOULD BE SET
1612 005540 105077 012150  CLRB     @TXCSR          ;CLR LOWER BYTE
1613 005544 017701 012144  MOV      @TXCSR,R1          ;SAVE TXCSR
1614 005550 042701 002000  BIC      #BITW,R1          ;CLR BIT WINDOW (DITTO)
1615 005554 022701 020200  CMP      #20200,R1          ;TEST TXCSR
1616 005560 001401          BEQ      .+4
1617 005562 104000          HLT          ;ONLY MAINT CLK BIT & TXDONE SHOULD BE SET
1618 005564 104400          SCOPE
1619                                ;;THIS TEST PERFORMS MASTER RESET TESTING &
1620                                ;;TESTING OF READ ONLY BIT BITW
1621                                ;;MAINT INTERNAL
1622                                ;
1623 005566 012767 000032 173332  TST26:  MOV      #26,TSTNO          ;SAVE THIS
1624 005574 012767 005720 173314  MOV      #TST27,NEXT          ;GO TO THIS TEST WHEN THRU
1625 005602 012777 044001 012104  MOV      #MINT!MTOA!BREAK,@TXCSR ;SET MAINT INT.,BREAK,
1626                                ;&MTOA
1627 005610 032777 002000 012076  BIT      #BITW,@TXCSR          ;TEST BITW
1628 005616 001001          BNE     .+4
1629 005620 104000          HLT          ;BIT WINDOW SHOULD BE SET
1630 005622 042777 040000 012064  BIC      #MTOA,@TXCSR
1631 005630 013702 001136  MOV      @HOLD,R2
    
```

```

1632 005634 005302      1$: DEC      R2
1633 005636 001376      BNE      1$
1634 005640 032777 002000 012046  BIT      @BITW,@TXCSR
1635 005646 001401      BEQ      .+4
1636 005650 104000      HLT      ;BIT SHOULD BE CLR
1637      ;NOW SET THE MTDATA
1638 005652 052777 040000 012034  BIS      @MTDATA,@TXCSR
1639 005660 052777 000400 012026  BIS      @MRESET,@TXCSR ;MASTER RESET
1640 005666 052777 004001 012020  BIS      @MINT!BREAK,@TXCSR
1641 005674 013702 001136      MOV      @#HOLD,R2
1642 005700 005302      2$: DEC      R2
1643 005702 001376      BNE      2$
1644 005704 032777 002000 012002  BIT      @BITW,@TXCSR
1645 005712 001401      BEQ      .+4
1646 005714 104000      HLT      ;BITW SHOULD BE CLR BY MASTER RESET
1647 005716 104400      SCOPE
1648      ;;THIS TEST PERFORMS MASTER RESET TESTING &
1649      ;;TESTING OF READ ONLY BIT BITW
1650      ;;MAINT EXTERNAL
1651      ;;
1652 005720 012767 000033 173200 TST27: MOV      @27,TSTNO ;SAVE THIS
1653 005726 012767 006060 173162  MOV      @TST28,NEXT ;GO TO THIS TEST WHEN THRU
1654      ;TEST TO SEE IF EXTERNAL MODEM BYPASS CONNECTOR
1655      ;IS ON (H315)....IF "NO" JUMP AROUND TEST
1656 005734 105767 173243  TSTB     JMRBY
1657 005740 100046      BPL      1$ ;IT IS NOT ON
1658 005742 012777 050001 011744  MOV      @NEXT!MTDATA!BREAK,@TXCSR ;SET MAINT EXT.,BREAK,
1659      ;@MTDATA
1660 005750 032777 002000 011736  BIT      @BITW,@TXCSR ;TEST BITW
1661 005756 001001      BNE      .+4
1662 005760 104000      HLT      ;BIT WINDOW SHOULD BE SET
1663 005762 042777 040000 011724  BIC      @MTDATA,@TXCSR
1664 005770 013702 001136      MOV      @#HOLD,R2
1665 005774 005302      2$: DEC      R2
1666 005776 001376      BNE      2$
1667 006000 032777 002000 011706  BIT      @BITW,@TXCSR
1668 006006 001401      BEQ      .+4
1669 006010 104000      HLT      ;BIT SHOULD BE CLR
1670      ;NOW SET THE MTDATA
1671 006012 052777 040000 011674  BIS      @MTDATA,@TXCSR
1672 006020 052777 000400 011666  BIS      @MRESET,@TXCSR ;MASTER RESET
1673 006026 052777 010001 011660  BIS      @NEXT!BREAK,@TXCSR
1674 006034 013702 001136      MOV      @#HOLD,R2
1675 006040 005302      3$: DEC      R2
1676 006042 001376      BNE      3$
1677 006044 032777 002000 011642  BIT      @BITW,@TXCSR
1678 006052 001401      BEQ      .+4
1679 006054 104000      HLT      ;BITW SHOULD BE CLR BY MASTER RESET
1680 006056 104400      1$: SCOPE
1681      ;;
1682      ;;THIS TEST PERFORMS MASTER RESET TESTING &
1683      ;;TESTING OF READ ONLY BIT RXDONE
1684      ;;
1685      ;;
1686 006060 012767 000034 173040 TST28: MOV      @28,TSTNO ;SAVE THIS
1687 006066 012767 006116 173022  MOV      @TST29,NEXT ;GO TO THIS TEST WHEN THRU
    
```

```

1688 006074 052777 000400 011612      BIS      #MRESET, @TXCSR ; MASTER RESET
1689 006102 032777 000200 011570      BIT      #RXDONE, @RXCSR ; TEST THIS BIT
1690 006110 001401                BEQ      64$           ; BR IF "0"
1691 006112 104000                HLT                    ; CHECK MASTER RESET LOGIC
1692 006114                64$:
1693                                ; OR SHORT ON THIS BIT
1694 006114 104400      SCOPE
1695                                ;; THIS TEST PERFORMS MASTER RESET TESTING &
1696                                ;; TESTING OF READ ONLY BIT RECACT
1697                                ;;
1698 006116 012767 000035 173002 TST29: MOV      #29, TSTNO      ; SAVE THIS
1699 006124 012767 006154 172764      MOV      #TST30, NEXT    ; GO TO THIS TEST WHEN THRU
1700 006132 052777 000400 011554      BIS      #MRESET, @TXCSR ; MASTER RESET
1701 006140 032777 004000 011532      BIT      #RECACT, @RXCSR ; TEST THIS BIT
1702 006146 001401                BEQ      64$           ; BR IF "0"
1703 006150 104000                HLT                    ; CHECK MASTER RESET LOGIC
1704 006152                64$:
1705                                ; OR SHORT ON THIS BIT
1706 006152 104400      SCOPE
1707                                ;; THIS TEST PERFORMS MASTER RESET TESTING &
1708                                ;; TESTING OF READ ONLY BIT DSC
1709                                ;;
1710 006154 012767 000036 172744 TST30: MOV      #30, TSTNO      ; SAVE THIS
1711 006162 012767 006212 172726      MOV      #TST31, NEXT    ; GO TO THIS TEST WHEN THRU
1712 006170 052777 000400 011516      BIS      #MRESET, @TXCSR ; MASTER RESET
1713 006176 032777 100000 011474      BIT      #DSC, @RXCSR    ; TEST THIS BIT
1714 006204 001401                BEQ      64$           ; BR IF "0"
1715 006206 104000                HLT                    ; CHECK MASTER RESET LOGIC
1716 006210                64$:
1717                                ; OR SHORT ON THIS BIT
1718 006210 104400      SCOPE
1719                                ;; THIS TEST PERFORMS MASTER RESET TESTING &
1720                                ;; TESTING OF READ ONLY BIT TXDONE
1721                                ;;
1722 006212 012767 000037 172706 TST31: MOV      #31, TSTNO      ; SAVE THIS
1723 006220 012767 006250 172670      MOV      #TST32, NEXT    ; GO TO THIS TEST WHEN THRU
1724 006226 052777 000400 011460      BIS      #MRESET, @TXCSR ; MASTER RESET
1725 006234 032777 000200 011452      BIT      #TXDONE, @TXCSR ; TEST THIS BIT
1726 006242 001001                BNE      .+4            ; BR IF "1"
1727 006244 104000                HLT                    ; CHECK MASTER RESET LOGIC
1728                                ; OR SHORT ON THIS BIT
1729 006246 104400      SCOPE
1730                                ;; THIS TEST PERFORMS MASTER RESET TESTING &
1731                                ;; TESTING OF READ ONLY BIT DNA
1732                                ;;
1733 006250 012767 000040 172650 TST32: MOV      #32, TSTNO      ; SAVE THIS
1734 006256 012767 006306 172632      MOV      #TST33, NEXT    ; GO TO THIS TEST WHEN THRU
1735 006264 052777 000400 011422      BIS      #MRESET, @TXCSR ; MASTER RESET
1736 006272 032777 100000 011414      BIT      #DNA, @TXCSR    ; TEST THIS BIT
1737 006300 001401                BEQ      64$           ; BR IF "0"
1738 006302 104000                HLT                    ; CHECK MASTER RESET LOGIC
1739 006304                64$:
1740                                ; OR SHORT ON THIS BIT
1741 006304 104400      SCOPE
1742                                ;; THIS TEST PERFORMS MASTER RESET TESTING &
1743                                ;; TESTING OF READ ONLY WORD RECEIVE DATA

```

```

1744
1745 006306 012767 000041 172612 TST33: MOV #33,TSTNO ;SAVE THIS
1746 006314 012767 006354 172574 MOV #TST34,NEXT ;GO TO THIS TEST WHEN THRU
1747 006322 052777 000400 011364 BIS #MRESET,DTXCSR ;MASTER RESET
1748 006330 016703 011350 MOV RXDBUF,R3 ;FOR ERROR MESSAGE
1749 006334 012700 000377 MOV #377,R0 ;EXPECTED
1750 006340 017701 011340 MOV @RXDBUF,R1 ;ACTUAL
1751 006344 120001 CMPB R0,R1
1752 006346 001401 BEQ .+4 ;BR IF "0"
1753 006350 104002 HLT 2 ;REC DATA SHOULD BE ALL 1'S
1754 006352 104400 SCOPE
1755 ;;THIS TEST PERFORMS MASTER RESET TESTING &
1756 ;;TESTING OF READ ONLY BIT PAPER
1757
1758 006354 012767 000042 172544 TST34: MOV #34,TSTNO ;SAVE THIS
1759 006362 012767 006412 172526 MOV #TST35,NEXT ;GO TO THIS TEST WHEN THRU
1760 006370 052777 000400 011316 BIS #MRESET,DTXCSR ;MASTER RESET
1761 006376 032777 010000 011300 BIT #PAPER,@RXDBUF ;TEST THIS BIT
1762 006404 001401 BEQ 64$ ;BR IF "0"
1763 006406 104000 HLT ;CHECK MASTER RESET LOGIC
1764 006410 64$:
1765 ;OR SHORT ON THIS BIT
1766 006410 104400 SCOPE
1767 ;;THIS TEST PERFORMS MASTER RESET TESTING &
1768 ;;TESTING OF READ ONLY BIT FMERR
1769
1770 006412 012767 000043 172506 TST35: MOV #35,TSTNO ;SAVE THIS
1771 006420 012767 006450 172470 MOV #TST36,NEXT ;GO TO THIS TEST WHEN THRU
1772 006426 052777 000400 011260 BIS #MRESET,DTXCSR ;MASTER RESET
1773 006434 032777 020000 011242 BIT #FMERR,@RXDBUF ;TEST THIS BIT
1774 006442 001401 BEQ 64$ ;BR IF "0"
1775 006444 104000 HLT ;CHECK MASTER RESET LOGIC
1776 006446 64$:
1777 ;OR SHORT ON THIS BIT
1778 006446 104400 SCOPE
1779 ;;THIS TEST PERFORMS MASTER RESET TESTING &
1780 ;;TESTING OF READ ONLY BIT OVRUN
1781
1782 006450 012767 000044 172450 TST36: MOV #36,TSTNO ;SAVE THIS
1783 006456 012767 006506 172432 MOV #TST37,NEXT ;GO TO THIS TEST WHEN THRU
1784 006464 052777 000400 011222 BIS #MRESET,DTXCSR ;MASTER RESET
1785 006472 032777 040000 011204 BIT #OVRUN,@RXDBUF ;TEST THIS BIT
1786 006500 001401 BEQ 64$ ;BR IF "0"
1787 006502 104000 HLT ;CHECK MASTER RESET LOGIC
1788 006504 64$:
1789 ;OR SHORT ON THIS BIT
1790 006504 104400 SCOPE
1791 ;;THIS TEST PERFORMS MASTER RESET TESTING &
1792 ;;TESTING OF READ ONLY BIT RXERR
1793
1794 006506 012767 000045 172412 TST37: MOV #37,TSTNO ;SAVE THIS
1795 006514 012767 006544 172374 MOV #TST38,NEXT ;GO TO THIS TEST WHEN THRU
1796 006522 052777 000400 011164 BIS #MRESET,DTXCSR ;MASTER RESET
1797 006530 032777 100000 011146 BIT #RXERR,@RXDBUF ;TEST THIS BIT
1798 006536 001401 BEQ 64$ ;BR IF "0"
1799 006540 104000 HLT ;CHECK MASTER RESET LOGIC
    
```

```

1800 006542          64$:
1801
1802 006542 104400          SCOPE
1803                                     ;; THIS TEST VERIFYS THAT THE DEVICE REGISTER RXCSR
1804                                     ;; IS CLEARED BY MASTER RESET
1805 006544 012767 000046 172354 TST38: MOV #38,TSTNO ;SAVE THIS
1806 006552 012767 006652 172336 MOV #TST39,NEXT ;GO TO THIS TEST WHEN THRU
1807 006560 012777 177777 011112 MOV #177777,@RXCSR ;SET ALL POSSIBLE BITS
1808 006566 052777 000400 011120 BIS #MRESET,@TXCSR ;MASTER RESET
1809 006574 016703 011100 MOV RXCSR,R3 ;FOR ERROR MESSAGE
1810 006600 017701 011074 MOV @RXCSR,R1 ;SAVE ACTUAL
1811 006604 105767 172371 TSTB OPTCLR ;TEST THE OPT CLR JUMPER FLAG
1812 006610 100010 BPL 15 ;NO,ITS NOT IN
1813 006612 042701 173000 BIC #173000,R1 ;CLR NON-MASTER RESETTABLE
1814                                     ;BITS(SINCE THESE ARE DEPENDENT ON H315 CONNECTORS EXISTANCE)
1815 006616 012700 000000 MOV #0,R0 ;EXPECTED
1816 006622 020001 CMP R0,R1 ;EXPECTED VS ACTUAL
1817 006624 001401 BEQ .+4
1818 006626 104001 HLT 1 ;ALL MASTER RESETABLE BITS SHOULD BE CLR
1819 006630 000407 BR 25 ;JUMP AROUND
1820 006632 042701 073000 15: BIC #73000,R1 ;CLR NON-MASTER RESETTABLE
1821                                     ;BITS(SINCE THESE ARE DEPENDENT ON H315 CONNECTORS EXISTANCE)
1822 006636 012700 000016 MOV #16,R0 ;EXPECTED
1823 006642 020001 CMP R0,R1 ;EXPECTED VS ACTUAL
1824 006644 001401 BEQ .+4
1825 006646 104001 HLT 1 ;ONLY STD,RTS,DTR BITS SHOULD BE SET
1826                                     ;NOTE THAT STD IS READ =1 INDEPENDENT OF
1827                                     ;SEC XMIT #6 STRAP
1828 006650 104400          25: SCOPE
1829
1830                                     ;; THIS TEST VERIFYS THAT THE DEVICE REGISTER TXCSR
1831                                     ;; IS CLEARED BY MASTER RESET
1832
1833 006652 012767 000047 172246 TST39: MOV #39,TSTNO ;SAVE THIS
1834 006660 012767 006726 172230 MOV #TST40,NEXT ;GO TO THIS TEST WHEN THRU
1835 006666 012777 177777 011020 MOV #177777,@TXCSR ;SET ALL POSSIBLE BITS
1836 006674 052777 000400 011012 BIS #MRESET,@TXCSR ;MASTER RESET
1837 006702 016703 011006 MOV TXCSR,R3 ;FOR ERROR MESSAGE
1838 006706 017701 011002 MOV @TXCSR,R1 ;SAVE ACTUAL
1839 006712 012700 000200 MOV #200,R0 ;EXPECTED
1840 006716 020001 CMP R0,R1 ;EXPECTED VS ACTUAL
1841 006720 001401 BEQ .+4
1842 006722 104001 HLT 1 ;ONLY TXDONE SHOULD BE SET
1843 006724 104400          SCOPE
1844
1845                                     ;; THIS TEST VERIFYS THAT THE DEVICE REGISTER RXDBUF
1846                                     ;; IS CLEARED BY MASTER RESET
1847
1848 006726 012767 000050 172172 TST40: MOV #40,TSTNO ;SAVE THIS
1849 006734 012767 006774 172154 MOV #TST41,NEXT ;GO TO THIS TEST WHEN THRU
1850 006742 052777 000400 010744 BIS #MRESET,@TXCSR ;MASTER RESET
1851 006750 016703 010730 MOV RXDBUF,R3 ;FOR ERROR MESSAGE
1852 006754 017701 010724 MOV @RXDBUF,R1 ;SAVE
1853 006760 012700 000377 MOV #377,R0 ;EXPECTED
1854 006764 020001 CMP R0,R1 ;EXPECTED VS ACTUAL
1855 006766 001401 BEQ .+4
    
```

```

1856 006770 104002
1857 006772 104400
1858
1859
1860
1861
1862
1863
1864 006774 012767 000051 172124 TST41:
1865 007002 012767 007710 172106
1866 007010 005077 010664
1867 007014 052777 000400 010672
1868
1869
1870
1871
1872 007022 105767 172155
1873 007026 100402
1874
1875 007030 000167 000652
1876
1877 007034 016703 010640
1878 007040 017701 010634
1879 007044 005000
1880 007046 005701
1881 007050 001401
1882 007052 104001
1883 007054 052777 000002 010616
1884
1885
1886
1887
1888 007062 016702 172050
1889 007066
1890 007066 005302
1891 007070 001376
1892
1893
1894 007072 017701 010602
1895 007076 012700 130002
1896 007102 020001
1897 007104 001401
1898 007106 104001
1899 007110 017701 010564
1900 007114 012700 030002
1901 007120 020001
1902 007122 001401
1903 007124 104001
1904
1905 007126 052777 000004 010544
1906
1907
1908
1909
1910 007134 016702 171776
1911 007140

```

```

HLT 2 ; ONLY REC DATA BITS SHOULD BE SET
SCOPE
; THIS TEST VERIFYS BITS RING, CTS, CARDET, SRD, DSR
; ALSO DSC IS GENERATED WHEN ANY OF THESE BITS ARE SET
; OR CLEARED..... IT ALSO CHECKS THE MODEM BYPASS
; JUMPER AND THAT THESE BITS CAN BE READ
; NOTE: THE MODEM BYPASS JUMPER MUST BE ON (H315)
MOV #41, TSTNO ; SAVE THIS
MOV #TST42, NEXT ; GO TO THIS TEST WHEN THRU
CLR #RXCSR ; TO GET RID OF STD RTS, DTR IF OPTCLR JUMPER #4 IS NOT ON
BIS #MRESET, #TXCSR ; MASTER RESET
; TEST THAT A "YES" ANSWER WAS GIVEN TO QUESTION IN
; THE MONITOR OR BY DEFAULT
; THIS TEST WILL BE BYPASSED IF THE EXTERNAL BYPASS
; JUMPER IS NOT INSTALLED
TSTB JMRBY
BMI .+6 ; THE ANSWER WAS YES.....
; PERFORM THIS TEST
JMP OUT1 ; JUMP AROUND THIS TEST IF THE ANSWER
; WAS NO
MOV RXCSR, R3 ; SET UP FOR ERROR MESSAGE
MOV #RXCSR, R1 ; ACTUAL
CLR R0 ; EXPECTED
TST R1 ; IS IT = 0 ?
BEQ .+4
HLT 1 ; RXCSR SHOULD BE CLR
BIS #DTR, #RXCSR ; SET DTR
; WAIT FOR CABLE DELAYS
; *****
; MODIFY "HOLD:" ACCORDINGLY FOR FASTER OR SLOWER MACHINE
; *****
MOV HOLD, R2 ; SET DELAY TIME
645:
DEC R2
BNE 645 ; WAIT THIS TIME
; OK NOW FALL THRU AND CONTINUE TESTING.....
; EXIT STAGE LEFT....CHINNING!
MOV #RXCSR, R1 ; ACTUAL
MOV #130002, R0 ; DSC, CTS, CARDET, DTR
CMP R0, R1 ; EXPECTED VS ACTUAL
BEQ .+4
HLT 1 ; CHECK BYPASS CONNECTOR
MOV #RXCSR, R1 ; ACTUAL
MOV #30002, R0 ; CTS, CARDET, DTR
CMP R0, R1 ; EXPECTED VS ACTUAL
BEQ .+4
HLT 1 ; PREVIOUS READING OF RXCSR SHOULD
; HAVE CLEARED DSC
BIS #RTS, #RXCSR
; WAIT FOR CABLE DELAYS
; *****
; MODIFY "HOLD:" ACCORDINGLY FOR FASTER OR SLOWER MACHINE
; *****
MOV HOLD, R2 ; SET DELAY TIME
655:

```



1912	007140	005302			DEC	R2	
1913	007142	001376			BNE	65\$	;WAIT THIS TIME
1914							;OK NOW FALL THRU AND CONTINUE TESTING.....
1915							;EXIT STAGE LEFT....CHINNG!
1916	007144	017701	010530		MOV	DRXCSR,R1	
1917	007150	012700	170006		MOV	#170006,R0	;DSC,RING,CTS,CARDET,RTS,DTR
1918	007154	020001			CMP	R0,R1	;EXPECTED VS ACTUAL
1919	007156	001401			BEQ	+4	
1920	007160	104001			HLT	1	;CHECK BYPASS CONNECTOR
1921	007162	017701	010512		MOV	DRXCSR,R1	
1922	007166	012700	070006		MOV	#70006,R0	;RING,CTS,CARDET,RTS,DTR
1923	007172	020001			CMP	R0,R1	;EXPECTED VS ACTUAL
1924	007174	001401			BEQ	+4	
1925	007176	104001			HLT	1	;PREVIOUS READING OF RXCSR SHOULD
1926							;HAVE CLEARED DSC
1927	007200	105767	171773		TSTB	SEXMIT	;IS SEC XMIT JUMPER IN ?
1928	007204	100112			BPL	OUT2	;NO
1929	007206	105767	171766		TSTB	SEREC	;IS SEC REC JUMPER IN ?
1930	007212	100163			BPL	OUT3	;NO
1931	007214	052777	000010	010456	BIS	#STD,DRXCSR	
1932							;WAIT FOR CABLE DELAYS
1933							*****
1934							;MODIFY "HOLD:" ACCORDINGLY FOR FASTER OR SLOWER MACHINE
1935							*****
1936	007222	016702	171710		MOV	HOLD,R2	;SET DELAY TIME
1937	007226			65\$:			
1938	007226	005302			DEC	R2	
1939	007230	001376			BNE	65\$	;WAIT THIS TIME
1940							;OK NOW FALL THRU AND CONTINUE TESTING.....
1941							;EXIT STAGE LEFT....CHINNG!
1942	007232	017701	010442		MOV	DRXCSR,R1	
1943	007236	012700	173016	MASK1:	MOV	#173016,R0	;DSC,RING,CTS,CARDET,SRD,DSR
1944							;STD,RTS,DTR
1945	007242	020001			CMP	R0,R1	;EXPECTED VS ACTUAL
1946	007244	001401			BEQ	+4	
1947	007246	104001			HLT	1	;CHECK BYPASS CONNECTOR
1948	007250	017701	010424		MOV	DRXCSR,R1	
1949	007254	012700	073016	MASK2:	MOV	#73016,R0	;RING,CTS,CARDET,SRD,DSR,STD
1950							;RTS,DTR
1951	007260	020001			CMP	R0,R1	;EXPECTED VS ACTUAL
1952	007262	001401			BEQ	+4	
1953	007264	104001			HLT	1	;PREVIOUS READING OF RXCSR SHOULD
1954							;HAVE CLEARED DSC
1955	007266	042777	000002	010404	BIC	#DTR,DRXCSR	
1956							;WAIT FOR CABLE DELAYS
1957							*****
1958							;MODIFY "HOLD:" ACCORDINGLY FOR FASTER OR SLOWER MACHINE
1959							*****
1960	007274	016702	171636		MOV	HOLD,R2	;SET DELAY TIME
1961	007300			64\$:			
1962	007300	005302			DEC	R2	
1963	007302	001376			BNE	64\$	;WAIT THIS TIME
1964							;OK NOW FALL THRU AND CONTINUE TESTING.....
1965							;EXIT STAGE LEFT....CHINNG!
1966	007304	017701	010370		MOV	DRXCSR,R1	
1967	007310	012700	143014		MOV	#143014,R0	;DSC,RING,SRD,DSR,STD,RTS

1968	007314	020001			CMP	R0,R1	;EXPECTED VS ACTUAL
1969	007316	001401			BEQ	.+4	
1970	007320	104001			HLT	i	;DSC SHOULD BE SET
1971	007322	042777	000004	010350	BIC	#RTS,DRXCSR	
1972							;WAIT FOR CABLE DELAYS
1973							*****
1974							;MODIFY "HOLD:" ACCORDINGLY FOR FASTER OR SLOWER MACHINE
1975							*****
1976	007330	016702	171602		MOV	HOLD,R2	;SET DELAY TIME
1977	007334			65\$:			
1978	007334	005302			DEC	R2	
1979	007336	001376			BNE	65\$	;WAIT THIS TIME
1980							;OK NOW FALL THRU AND CONTINUE TESTING.....
1981							;EXIT STAGE LEFT....CHINNG!
1982	007340	017701	010334		MOV	DRXCSR,R1	
1983	007344	012700	103010	MASK3:	MOV	#103010,R0	;DSC,SRD,DSR,STD
1984	007350	020001			CMP	R0,R1	;EXPECTED VS ACTUAL
1985	007352	001401			BEQ	.+4	
1986	007354	104001			HLT	i	;DSC SHOULD BE SET
1987	007356	042777	000010	010314	BIC	#STD,DRXCSR	
1988							;WAIT FOR CABLE DELAYS
1989							*****
1990							;MODIFY "HOLD:" ACCORDINGLY FOR FASTER OR SLOWER MACHINE
1991							*****
1992	007364	016702	171546		MOV	HOLD,R2	;SET DELAY TIME
1993	007370			64\$:			
1994	007370	005302			DEC	R2	
1995	007372	001376			BNE	64\$	;WAIT THIS TIME
1996							;OK NOW FALL THRU AND CONTINUE TESTING.....
1997							;EXIT STAGE LEFT....CHINNG!
1998	007374	017701	010300		MOV	DRXCSR,R1	
1999	007400	012700	100000		MOV	#100000,R0	;DSC
2000	007404	020001			CMP	R0,R1	;EXPECTED VS ACTUAL
2001	007406	001401			BEQ	.+4	
2002	007410	104001			HLT	i	;DSC SHOULD E. SET
2003	007412	017701	010262		MOV	DRXCSR,R1	
2004	007416	005000			CLR	R0	;NONE
2005	007420	005701			TST	R1	
2006	007422	001401			BEQ	.+4	
2007	007424	104001			HLT	i	;DSC SHOULD BE CLEARED FROM PREVIOUS
2008							;READING OF RXCSR
2009	007426	000167	000254		JMP	OUT1	;JUMP AROUND
2010							;THE FOLLOING ROUTINE HANDLES THE SITUATION WHERE SEC XMIT
2011							;AND SEC REC JUMPERS ARE NOT ON
2012	007432	052777	000010	010240	OUT2:	BIS	#STD,DRXCSR
2013							;WAIT FOR CABLE DELAYS
2014							*****
2015							;MODIFY "HOLD:" ACCORDINGLY FOR FASTER OR SLOWER MACHINE
2016							*****
2017	007440	016702	171472		MOV	HOLD,R2	;SET DELAY TIME
2018	007444			64\$:			
2019	007444	005302			DEC	R2	
2020	007446	001376			BNE	64\$	;WAIT THIS TIME
2021							;OK NOW FALL THRU AND CONTINUE TESTING.....
2022							;EXIT STAGE LEFT....CHINNG!
2023	007450	017701	010224		MOV	DRXCSR,R1	;ACTUAL



```

2080
2081
2082
2083 007624 016702 171306
2084 007630
2085 007630 005302
2086 007632 001376
2087
2088
2089 007634 017701 010040
2090 007640 012700 131012
2091 007644 020001
2092 007646 001401
2093 007650 004001
2094 007652 042777 000002 01002C
2095
2096
2097
2098
2099 007660 016702 171252
2100 007664
2101 007664 005302
2102 007666 001376
2103
2104
2105 007670 017701 010004
2106 007674 012700 10101C
2107 007700 020001
2108 007702 001401
2109 007704 104001
2110 007706 104400
2111
2112
2113
2114
2115
2116 007710 012767 000052 171210
2117 007716 012767 010036 171172
2118 007724 052777 000400 007762
2119 007732 012777 020000 007750
2120 007740 052777 000400 007746
2121
2122
2123 007746 012777 064001 007740
2124
2125
2126 007754 012777 026026 007726
2127 007762 032777 004000 007710
2128 007770 001401
2129 007772 104000
2130 007774
2131 007774 052777 000020 007676
2132 010002 032777 004000 007670
2133 010010 001001
2134 010012 104000
2135 010014

```

```

:*****
:MODIFY "HOLD:" ACCORDINGLY FOR FASTER OR SLOWER MACHINE
:*****
MOV HOLD,R2 ;SET DELAY TIME
655:
DEC R2
BNE 655 ;WAIT THIS TIME
;OK NOW FALL THRU AND CONTINUE TESTING.....
;EXIT STAGE LEFT....CHINING!
MOV @RXCSR,R1 ;ACTUAL
MOV @131012,R0 ;EXPECTED: DSC,CTS,CARDET,DSR,STD,DTR
CMP R0,R1 ;EXPECTED VS ACTUAL
BEQ +4
HLT 1 ;CHECK H315 CONNECTOR
BIC @DTR,@RXCSR
;WAIT FOR CABLE DELAYS
:*****
:MODIFY "HOLD:" ACCORDINGLY FOR FASTER OR SLOWER MACHINE
:*****
MOV HOLD,R2 ;SET DELAY TIME
665:
DEC R2
BNE 665 ;WAIT THIS TIME
;OK NOW FALL THRU AND CONTINUE TESTING.....
;EXIT STAGE LEFT....CHINING!
MOV @RXCSR,R1 ;ACTUAL
MOV @101010,R0 ;EXPECTED: DSC,DSR,STD
CMP R0,R1 ;EXPECTED VS ACTUAL
BEQ +4
HLT 1 ;CHECK H315 CONNECTOR
OUT1: SCOPE

```

```

;; THIS TEST VERIFYS THAT REACT (REC ACTIVE) ASSERTS
;; IMMED. WHEN SYNC EXTERNAL MODE IS SELECTED
;; AND SYNC SEARCH IS SET
TST42: MOV @42,TSTNO ;SAVE THIS
MOV @TST43,NEXT ;GO TO THIS TEST WHEN THRU
BIS @MRESET,@TXCSR ;MASTER RESET
MOV @SYNEXT,@PARCSR ;SET THE MODE
BIS @MRESET,@TXCSR ;MASTER RESET
;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
MOV @MNTDATA!CLK!MINT!BREAK,@TXCSR
;SET MODE # OF BITS,PARITY SENSE,&LOAD SYNC REG
MOV @SYNEXT!EIGHT!NOPAR!26,@PARCSR
BIT @REACT,@RXCSR
BEQ 648
HLT ;REACT SHOULD NOT BE SET
648:
BIS @SYNSCH,@RXCSR ;SET SYNC SEARCH
BIT @REACT,@RXCSR
BNE 655
HLT ;REACT DID NOT ASSERT
655:

```

```

2136 010014 042777 000020 007656 BIC #SYNSCH,@RXCSR ;DROP SEARCH SYNC
2137 010022 032777 004000 007650 BIT #REACT,@RXCSR ;IS IT =0?
2138 010030 001401 BEQ 665
2139 010032 104000 HLT ;REACT SHOULD BE 0
2140 010034 665:
2141 010034 104400 SCOPE
2142 ;:THIS TEST VERIFYS THAT REACT (REC ACTIVE) ASSERTS
2143 ;:IMMED. WHEN ISOCRONOUS MODE IS SELECTED
2144 ;:AND SYNC SEARCH IS SET
2145 ;:
2146 010036 012767 000053 171062 TST43: MOV #43,TSTNO ;SAVE THIS
2147 010044 012767 010164 171044 MOV #TST44,NEXT ;GO TO THIS TEST WHEN THRU
2148 010052 052777 000400 007634 BIS #MRESET,@TXCSR ;MASTER RESET
2149 010060 012777 000000 007622 MOV #ISYMOD,@PARCSR ;SET THE MODE
2150 010056 052777 000400 007620 BIS #MRESET,@TXCSR ;MASTER RESET
2151 ;:
2152 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2153 010074 012777 064001 007612 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2154 ;:
2155 ;SET MODE # OF BITS,PARITY SENSE &LOAD SYNC REG
2156 010102 012777 006026 007600 MOV #ISYMOD!EIGHT!NOPAR!26,@PARCSR
2157 010110 032777 004000 007562 BIT #FECACT,@RXCSR
2158 010116 001401 BEQ 645
2159 010120 104000 HLT ;REACT SHOULD NOT BE SET
2160 010122 645:
2161 010122 052777 000020 007550 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2162 010130 032777 004000 007542 BIT #REACT,@RXCSR
2163 010136 001001 BNE 655
2164 010140 104000 HLT ;REACT DID NOT ASSERT
2165 010142 655:
2166 010142 042777 000020 007530 BIC #SYNSCH,@RXCSR ;DROP SEARCH SYNC
2167 010150 032777 004000 007522 BIT #REACT,@RXCSR ;IS IT =0?
2168 010156 001401 BEQ 665
2169 010160 104000 HLT ;REACT SHOULD BE 0
2170 010162 665:
2171 010162 104400 SCOPE
2172 ;:VERIFY THE MATCH DETECT & DATA RDY FLAGS BY PUMPING
2173 ;:IN TWO # SYNC CHARS THRU MAINT DATA BIT
2174 ;:MATCH THE RECACT BIT
2175 ;:ON THE THIRD # CHARACTER IT SHOULD SET RXDONE
2176 ;:#: DEPENDENT ON MONITOR.
2177 ;:IF ONE SYNC STRAP IS SELECTED THEN IT WILL ONLY
2178 ;:TAKE ONE SYNC CHARACTER FOR RXDONE TO ASSERT
2179 ;:ON THE SECOND CHARACTER
2180 ;:ALSO CHECK THIS CHARACTER IN RXDBUF
2181 ;:AND CHECK OPERATION OF SYNSCH
2182 ;:MODE: SYNC INTERNAL
2183 ;:LENGTH:FIVE
2184 ;:
2185 010164 012767 000054 170734 TST44: MOV #44,TSTNO ;SAVE THIS
2186 010172 012767 010506 170716 MOV #TST45,NEXT ;GO TO THIS TEST WHEN THRU
2187 010200 052777 000400 007506 BIS #MRESET,@TXCSR ;MASTER RESET
2188 010206 012777 030000 007474 MOV #SYNINT,@PARCSR ;SET THE MODE
2189 010214 052777 000400 007472 BIS #MRESET,@TXCSR ;MASTER RESET
2190 ;:
2191 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE

```

```

2192 010222 012777 064001 007464      MOV      #MTDATA:CLK!MINT!BREAK, @TXCSR
2193
2194                                     ;SET MODE # OF BITS, PARITY SENSE, & LOAD SYNC REG
2195 010230 012777 030026 007452      MOV      #SYNINT!FIVE!NOPAR!26, @PARCSR
2196 010236 016703 0C7442                MOV      RXDBUF, R3      ;SET UP FOR ERROR MESSAGE
2197 010242 052777 000020 007430      BIS      #SYNSCH, @RXCSR ;SET SYNC SEARCH
2198                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2199 010250 042777 020000 007436      BIC      #CLK, @TXCSR   ;POKE CLK DOWN
2200 010256 052777 020000 007430      BIS      #CLK, @TXCSR   ;POKE CLK UP
2201                                     ;POKE CLK TO GET LOGIC INTO SYN. RONIZATION
2202 010264 042777 020000 007422      BIC      #CLK, @TXCSR   ;POKE CLK DOWN
2203 010272 052777 020000 007414      BIS      #CLK, @TXCSR   ;POKE CLK UP
2204 010300 012767 000002 170634      MOV      #2, COUNT
2205 010306 012767 000005 170624 15:    MOV      #5, SHIFT      ;# OF SHIFTS
2206 010314 012767 000026 170622      MOV      #26, TEMP1    ;SYNC CHARACTER
2207 010322 004767 007056                JSR      PC, @POKE
2208 010326 005367 170610                DEC      COUNT
2209 010332 001403 25                    BEQ      25
2210                                     ;TEST SYNCNO TO SEE HOW MANY SYNC CHARS WERE SELECTED
2211 010334 105767 170636                TSTB    SYNCNO
2212 010340 100762 25                    BMI      15 ;TWO SYNC CHARS
2213 010342 105777 007332 25:    TSTB    @RXCSR ;CHECK REC DONE BIT
2214 010346 100001 645                    BPL      645
2215 010350 104000 645                    HLT      ;RXDONE SHOULD NOT BE ASSERTED
2216 010352
2217 010352 032777 004000 007320 645:  BIT      #REACT, @RXCSR
2218 010360 001001 655                    BNE
2219 010362 104000 655                    HLT      ;REACT SHOULD BE ASSERTED
2220 010364
2221 010364 012767 000005 170546 655:  MOV      #5, SHIFT
2222 010372 012767 000021 170544      MOV      #21, TEMP1    ;ANY CHARACTER
2223 010400 004767 007000                JSR      PC, @POKE
2224 010404 105777 007270                TSTB    @RXCSR ;CHECK RXDONE
2225 010410 100401 665                    BMI      665
2226 010412 104000 665                    HLT      ;RXDONE SHOULD BE ASSERTED
2227 010414
2228 010414 032777 004000 007256 665:  BIT      #REACT, @RXCSR
2229 010422 001001 675                    BNE
2230 010424 104000 675                    HLT      ;REACT SHOULD STILL BE ASSERTED
2231 010426
2232 010426 042777 000020 007244 675:  BIC      #SYNSCH, @RXCSR ;CLR SYNC SEARCH
2233 010434 032777 004000 007236      BIT      #REACT, @RXCSR ;IT SHOULD DROP IMMEDIATELY
2234 010442 001401 685                    BEQ
2235 010444 104000 685                    HLT      ;REACT SHOULD BE CLR
2236 010446
2237 010446 105777 007226 685:    TSTB    @RXCSR ;RXDONE
2238 010452 100401 695                    BMI
2239 010454 104000 695                    HLT      ;RXDONE SHOULD STILL BE ASSERTED
2240 010456
2241 010456 012700 000021 695:    MOV      #21, R0 ;EXPECTED DATA
2242 010462 017701 007216      MOV      @RXDBUF, R1 ;ACTUAL DATA
2243 010466 020001                CMP      R0, R1 ;COMPARE EXP VS ACT
2244 010470 001401 705                    BEQ
2245 010472 104002                HLT      2 ;DATA CHARS SHOULD COMPARE
2246 010474
2247 010474 105777 007200 705:    TSTB    @RXCSR ;CHECK RXDONE

```

```

2248 010500 100001          BPL      715
2249 010502 104000          HLT      ;RXDONE SHOULD BE CLR FROM
2250 010504                715:
2251                                ;PREVIOUS READING OF RXDBUF
2252 010504 104400          SCOPE
2253                                ;;VERIFY THE MATCH DETECT & DATA RDY FLAGS BY PUMPING
2254                                ;;IN TWO * SYNC CHARS THRU MAINT DATA BIT
2255                                ;;WATCH THE RECACT BIT
2256                                ;;ON THE THIRD * CHARACTER IT SHOULD SET RXDONE
2257                                ;;*: DEPENDENT ON MONITOR.
2258                                ;;IF ONE SYNC STRAP IS SELECTED THEN IT WILL ONLY
2259                                ;;TAKE ONE SYNC CHARACTER FOR RXDONE TO ASSERT
2260                                ;;ON THE SECOND CHARACTER
2261                                ;;ALSO CHECK THIS CHARACTER IN RXDBUF
2262                                ;;AND CHECK OPERATION OF SYN SCH
2263                                ;;MODE: SYNC INTERNAL
2264                                ;;LENGTH: SIX
2265
2266 010506 012767 000055 170412 TST45: MOV      #45, TSTNO      ;SAVE THIS
2267 010514 012767 011030 170374      MOV      #TST46, NEXT      ;GO TO THIS TEST WHEN THRU
2268 010522 052777 000400 007164      BIS      #MRESET, @TXCSR   ;MASTER RESET
2269 010530 012777 030000 007152      MOV      #SYNINT, @PARCSR  ;SET THE MODE
2270 010536 052777 000400 007150      BIS      #MRESET, @TXCSR   ;MASTER RESET
2271
2272                                ;SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
2273 010544 012777 064001 007142      MOV      #MTDATA!CLK!MINT!BREAK, @TXCSR
2274
2275                                ;SET MODE # OF BITS, PARITY SENSE & LOAD SYNC REG
2276 010552 012777 032026 007130      MOV      #SYNINT!SIX!NOPAR!26, @PARCSR
2277 010560 016703 007120                MOV      RXDBUF, R3        ;SET UP FOR ERROR MESSAGE
2278 010564 052777 000020 007106      BIS      #SYNSCH, @RXCSR   ;SET SYNC SEARCH
2279                                ;POKE CLK TO GET RECEIVER INTO SYNCHRONIZATION....
2280 010572 042777 020000 007114      BIC      #CLK, @TXCSR      ;POKE CLK DOWN
2281 010600 052777 020000 007106      BIS      #CLK, @TXCSR      ;POKE CLK UP
2282                                ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
2283 010606 042777 020000 007100      BIC      #CLK, @TXCSR      ;POKE CLK DOWN
2284 010614 052777 020000 007072      BIS      #CLK, @TXCSR      ;POKE CLK UP
2285 010622 012767 000002 170312      MOV      #2, COUNT
2286 010630 012767 000006 170302 15:  MOV      #6, SHIFT      ;# OF SHIFTS
2287 010636 012767 000026 170300      MOV      #26, TEMP1      ;SYNC CHARACTER
2288 010644 004767 006534                JSR      PC, @POKE
2289 010650 005367 170266                DEC      COUNT
2290 010654 001403                BEQ      25
2291                                ;TEST SYNCNO TO SEE HOW MANY SYNC CHARS WERE SELECTED
2292 010656 105767 170314      TSTB     SYNCNO
2293 010662 100762                BMI      15              ;TWO SYNC CHARS
2294 010664 105777 007010 25:  TSTB     @RXCSR          ;CHECK REC DONE BIT
2295 010670 100001                BPL      64$
2296 010672 104000                HLT
2297 010674                64$:
2298 010674 032777 004000 006776      BIT      #RECACT, @RXCSR
2299 010702 001001                BNE      65$
2300 010704 104000                HLT
2301 010706                65$:
2302 010706 012767 000006 170224      MOV      #6, SHIFT
2303 010714 012767 000021 170222      MOV      #21, TEMP1      ;ANY CHARACTER
    
```



```

2304 010722 004767 006456 JSR PC,RPOKE
2305 010726 105777 006746 TSTB @RXCSR ;CHECK RXDONE
2306 010732 100401 BMI 66$
2307 010734 104000 HLT ;RXDONE SHOULD BE ASSERTED
2308 010736 66$:
2309 010736 032777 004000 006734 BIT #REACT,@RXCSR
2310 010744 001001 SNE 67$
2311 010746 104000 HLT ;REACT SHOULD STILL BE ASSERTED
2312 010750 67$:
2313 010750 042777 000020 006722 BIC #SYNSCH,@RXCSR ;CLR SYNC SEARCH
2314 010756 032777 004000 006714 BIT #REACT,@RXCSR ;IT SHOULD DROP IMMEDIATELY
2315 010764 001401 BEQ 68$
2316 010766 104000 HLT ;REACT SHOULD BE CLR
2317 010770 68$:
2318 010770 105777 006704 TSTB @RXCSR ;RXDONE
2319 010774 100401 BMI 69$
2320 010776 104000 HLT ;RXDONE SHOULD STILL BE ASSERTED
2321 011000 69$:
2322 011000 012700 000021 MOV #21,R0 ;EXPECTED DATA
2323 011004 017701 006674 MOV @RXDBUF,R1 ;ACTUAL DATA
2324 011010 020001 CMP R0,R1 ;COMPARE EXP VS ACT
2325 011012 001401 BEQ 70$
2326 011014 104002 HLT 2 ;DATA CHARS SHOULD COMPARE
2327 011016 70$:
2328 011016 105777 006656 TSTB @RXCSR ;CHECK RXDONE
2329 011022 100001 BPL 71$
2330 011024 104000 HLT ;RXDONE SHOULD BE CLR FROM
2331 011026 71$:
2332 ;PREVIOUS READING OF RXDBUF
2333 011026 104400 SCOPE
2334 ;:VERIFY THE MATCH DETECT & DATA RDY FLAGS BY PUMPING
2335 ;:IN TWO * SYNC CHARS THRU MAINT DATA BIT
2336 ;:WATCH THE REACT BIT
2337 ;:ON THE THIR * CHARACTER IT SHOULD SET RXDONE
2338 ;:*: DEPENDENT ON MONITOR.....
2339 ;:IF ONE SYNC STRAP IS SELECTED THEN IT WILL ONLY
2340 ;:TAKE ONE SYNC CHARACTER FOR RXDONE TO ASSERT
2341 ;:ON THE SECOND CHARACTER
2342 ;:ALSO CHECK THIS CHARACTER IN RXDBUF
2343 ;:AND CHECK OPERATION OF SYNSCH
2344 ;:MODE: SYNC INTERNAL
2345 ;:LENGTH: SEVEN
2346 ;:
2347 011030 012767 000056 170070 TST46: MOV #46,TSTNO ;SAVE THIS
2348 011036 012767 011352 170052 MOV #TST47,NEXT ;GO TO THIS TEST WHEN THRU
2349 011044 052777 000400 006642 BIS #MRESET,@TXCSR ;MASTER RESET
2350 011052 012777 030000 006630 MOV #SYNINT,@PARCSR ;SET THE MODE
2351 011060 052777 000400 006626 BIS #MRESET,@TXCSR ;MASTER RESET
2352
2353 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2354 011066 012777 064001 006620 MOV #MTDATA:CLK:MINT:BREAK,@TXCSR
2355
2356 ;SET MODE , # OF BITS,PARITY SENSE &LOAD SYNC REG
2357 011074 012777 034026 006606 MOV #SYNINT:SEVEN:NOPAR:26,@PARCSR
2358 011102 016703 006576 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
2359 011106 052777 000020 006564 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
    
```

```

2360 ;POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
2361 011114 042777 020000 006572 BIC #CLK, @TXCSR ;POKE CLK DOWN
2362 011122 052777 020000 006564 BIS #CLK, @TXCSR ;POKE CLK UP
2363 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2364 011130 042777 020000 006556 BIC #CLK, @TXCSR ;POKE CLK DOWN
2365 011136 052777 020000 006550 BIS #CLK, @TXCSR ;POKE CLK UP
2366 011144 012767 000002 167770 MOV #2, COUNT
2367 011152 012767 000007 167760 15: MOV #7, SHIFT ;# OF SHIFTS
2368 011160 012767 000026 167756 MOV #26, TEMP1 ;SYNC CHARACTER
2369 011166 004767 006212 JSR PC, RPOKE
2370 011172 005367 167744 DEC COUNT
2371 011176 001403 BEQ 25
2372 ;TEST SYNCRNO TO SEE HOW MANY SYNC CHARS WERE SELECTED
2373 011200 105767 167772 TSTB SYNCRNO
2374 011204 100762 BMI 15 ;TWO SYNC CHARS
2375 011206 105777 006466 25: TSTB @RXCSR ;CHECK REC DONE BIT
2376 011212 100001 BPL 64$
2377 011214 104000 HLT ;RXDONE SHOULD NOT BE ASSERTED
2378 011216 64$:
2379 011216 032777 004000 006454 BIT #REACT, @RXCSR
2380 011224 001001 BNE 65$
2381 011226 104000 HLT ;REACT SHOULD BE ASSERTED
2382 011230 65$:
2383 011230 012767 000007 167702 MOV #7, SHIFT
2384 011236 012767 000021 167700 MOV #21, TEMP1 ;ANY CHARACTER
2385 011244 004767 006134 JSR PC, RPOKE
2386 011250 105777 006424 TSTB @RXCSR ;CHECK RXDONE
2387 011254 100401 BMI 66$
2388 011256 104000 HLT ;RXDONE SHOULD BE ASSERTED
2389 011260 66$:
2390 011260 032777 004000 006412 BIT #REACT, @RXCSR
2391 011266 001001 BNE 67$
2392 011270 104000 HLT ;REACT SHOULD STILL BE ASSERTED
2393 011272 67$:
2394 011272 042777 000020 006400 BIC #SYNSCH, @RXCSR ;CLR SYNC SEARCH
2395 011300 032777 004000 006372 BIT #REACT, @RXCSR ;IT SHOULD DROP IMMEDIATELY
2396 011306 001401 BEQ 68$
2397 011310 104000 HLT ;REACT SHOULD BE CLR
2398 011312 68$:
2399 011312 105777 006362 TSTB @RXCSR ;RXDONE
2400 011316 100401 BMI 69$
2401 011320 104000 HLT ;RXDONE SHOULD STILL BE ASSERTED
2402 011322 69$:
2403 011322 012700 000021 MOV #21, R0 ;EXPECTED DATA
2404 011326 017701 006352 MOV @RXDBUF, R1 ;ACTUAL DATA
2405 011332 020001 CMP R0, R1 ;COMPARE EXP VS ACT
2406 011334 001401 BEQ 70$
2407 011336 104002 HLT 2 ;DATA CHARS SHOULD COMPARE
2408 011340 70$:
2409 011340 105777 006334 TSTB @RXCSR ;CHECK RXDONE
2410 011344 100001 BPL 71$
2411 011346 104000 HLT ;RXDONE SHOULD BE CLR FROM
2412 011350 71$:
2413 ;PREVIOUS READING OF RXDBUF
2414 SCOPE
2415 ;;VERIFY THE MATCH DETECT & DATA RDY FLAGS BY PUMPING
    
```

```

16      ; IN TWO * SYNC CHARS THRU MAINT DATA BIT
17      ; WATCH THE RECACT BIT
18      ; ON THE THIRD * CHARACTER IT SHOULD SET RXDONE
19      ; * DEPENDENT ON MONITOR.
20      ; IF ONE SYNC STRAP IS SELECTED THEN IT WILL ONLY
21      ; TAKE ONE SYNC CHARACTER FOR RXDONE TO ASSERT
22      ; ON THE SECOND CHARACTER
23      ; ALSO CHECK THIS CHARACTER IN RXDBUF
24      ; AND CHECK OPERATION OF SYNCSCH
25      ; MODE: SYNC INTERNAL
26      ; LENGTH:EIGHT
27
28      011352 012767 000057 167546 TST47: MOV     #47,TSTNO      ;SAVE THIS
29      011360 012767 011674 167530      MOV     #TST48,NEXT    ;GO TO THIS TEST WHEN THRU
30      011366 052777 000400 006320      BIS     #MRESET,@TXCSR ;MASTER RESET
31      011374 012777 030000 006306      MOV     #SYNINT,@PARCSR ;SET THE MODE
32      011402 052777 000400 006304      BIS     #MRESET,@TXCSR ;MASTER RESET
33
34      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
35      011410 012777 064001 006276      MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
36
37      ;SET MODE , # OF BITS,PARITY SENSE &LOAD SYNC REG
38      011416 012777 036026 006264      MOV     #SYNINT!EIGHT!NOPAR!26,@PARCSR
39      011424 016703 006254 006264      MOV     RXDBUF,R3      ;SET UP FOR ERROR MESSAGE
40      011430 052777 000020 006242      BIS     #SYNCSCH,@RXCSR ;SET SYNC SEARCH
41      ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
42      011436 042777 020000 006250      BIC     #CLK,@TXCSR    ;POKE CLK DOWN
43      011444 052777 020000 006242      BIS     #CLK,@TXCSR    ;POKE CLK UP
44      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
45      011452 042777 020000 006234      BIC     #CLK,@TXCSR    ;POKE CLK DOWN
46      011460 052777 020000 006226      BIS     #CLK,@TXCSR    ;POKE CLK UP
47      011466 012767 000002 167446      MOV     #2,COUNT
48      011474 012767 000010 167436 1$:     MOV     #8,SHIFT      ;# OF SHIFTS
49      011502 012767 000026 167434      MOV     #26,TEMP1     ;SYNC CHARACTER
50      011510 004767 005670      JSR     PC,RPOKE
51      011514 005367 167422      DEC     COUNT
52      011520 001403      BEQ     2$
53      ;TEST SYNCNO TO SEE HOW MANY SYNC CHARS WERE SELECTED
54      011522 105767 167450      TSTB   SYNCNO
55      011526 100762      BMI     1$
56      011530 105777 006144 2$:     TSTB   @RXCSR        ;CHECK REC DONE BIT
57      011534 100001      BPL     64$
58      011536 104000      HLT
59      ;RXDONE SHOULD NOT BE ASSERTED
60      011540      64$:   BIT     #RECACT,@RXCSR
61      011546 001001      BNE     65$
62      011550 104000      HLT
63      ;RECACT SHOULD BE ASSERTED
64      011552      65$:   MOV     #8,SHIFT
65      011560 012767 000010 167360      MOV     #21,TEMP1     ;ANY CHARACTER
66      011566 012767 000021 167356      MOV     #21,TEMP1
67      011566 004767 005612      JSR     PC,RPOKE
68      011572 105777 006102      TSTB   @RXCSR        ;CHECK RXDONE
69      011576 100401      BMI     66$
70      011600 104000      HLT
71      ;RXDONE SHOULD BE ASSERTED
72      011602      66$:   BIT     #RECACT,@RXCSR
73      011602 032777 004000 006070

```

```

2472 011610 001001      BNE      67$
2473 011612 104000      HLT
2474 011614      67$:
2475 011614 042777 000020 006056      BIC      #SYNSCH,@RXCSR ;CLR SYNC SEARCH
2476 011622 032777 004000 006050      BIT      #RECACT,@RXCSR ;IT SHOULD DROP IMMEDIATELY
2477 011630 001401      BEQ      68$
2478 011632 104000      HLT
2479 011634      68$:
2480 011634 105777 006040      TSTB    @RXCSR ;RXDONE
2481 011640 100401      BMI      69$
2482 011642 104000      HLT
2483 011644      69$:
2484 011644 012700 000021      MOV      #21,R0 ;EXPECTED DATA
2485 011650 017701 006030      MOV      @RXDBUF,R1 ;ACTUAL DATA
2486 011654 020001      CMP      R0,R1 ;COMPARE EXP VS ACT
2487 011656 001401      BEQ      70$
2488 011660 104002      HLT
2489 011662      70$:
2490 011662 105777 006012      TSTB    @RXCSR ;CHECK RXDONE
2491 011666 100001      BPL      71$
2492 011670 104000      HLT
2493 011672      71$:
2494      ;PREVIOUS READING OF RXDBUF
2495 011672 104400      SCOPE
2496      ;:THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2497      ;:RECEIVER SECTION,IT USES THE ERROR FLAGS
2498      ;:TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2499      ;:(OVRRUN,RXERR)
2500      ;:MODE:ISYMOD
2501      ;:LENGTH:FIVE
2502      ;:CHAR:25
2503
2504 011674 012767 000060 167224      TST48: MOV      #48,TSTNO ;SAVE THIS
2505 011702 012767 012136 167206      MOV      #TST49,NEXT ;GO TO THIS TEST WHEN THRU
2506 011710 052777 000400 005776      BIS      #MRESET,@TXCSR ;MASTER RESET
2507 011716 012777 000000 005764      MOV      #ISYMOD,@PARCSR ;SET THE MODE
2508 011724 052777 000400 005762      BIS      #MRESET,@TXCSR ;MASTER RESET
2509
2510      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2511 011732 012777 064001 005754      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
2512
2513      ;SET MODE, # OF BITS,PARITY SENSE,&LOAD SYNC REG
2514 011740 012777 000000 005742      MOV      #ISYMOD!FIVE!NOPAR!0,@PARCSR
2515 011746 052777 000020 005724      BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
2516      ;POKE CLK TO GET RECEIVER INTO SYNCRDIZATION....
2517 011754 042777 020000 005732      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2518 011762 052777 020000 005724      BIS      #CLK,@TXCSR ;POKE CLK UP
2519      ;POKE CLK TO GET LOGIC INTO SYNCRDIZATION
2520 011770 042777 020000 005716      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2521 011776 052777 020000 005710      BIS      #CLK,@TXCSR ;POKE CLK UP
2522 012004 016703 005674      MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
2523 012010 012700 000025      MOV      #25,R0 ;EXPECTED
2524 012014 012767 000007 167116      MOV      #7,SHIFT ;# OF SHIFTS
2525 012022 012767 000152 167114      MOV      #152,TEMP1 ;DATA CHAR
2526 012030 004767 005350      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2527 012034 105777 005640      TSTB    @RXCSR ;RXDONE
  
```

2528	012040	100401			BMI	64\$	
2529	012042	104000			HLT		;RXDONE SHOULD BE SET
2530	012044				64\$:		
2531	012044	017701	005634		MOV	@RXDBUF,R1	;ACTUAL
2532	012050	020001			CMP	R0,R1	;COMPARE EXPECTED VS. ACTUAL
2533	012052	001401			BEQ	65\$	
2534	012054	104002			HLT	2	;RECEIVED DATA DID NOT MATCH ;EXPECTED DATA - CHECK MAINT DATA ;OR RECEIVER LOGIC
2535					65\$:		
2536							
2537	012056				MOV	#7,SHIFT	;# OF SHIFTS
2538	012056	012767	000007	167054	MOV	#152,TEMP1	;DATA CHAR
2539	012064	012767	000152	167052	JSR	PC,RPOKE	;SHIFT IN THIS CHAR
2540	012072	004767	005306				;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2541					MOV	#7,SHIFT	;# OF SHIFTS
2542	012076	012767	000007	167034	MOV	#152,TEMP1	;DATA CHAR
2543	012104	012767	000152	167032	JSR	PC,RPOKE	;SHIFT IN THIS CHAR
2544	012112	004767	005266		MOV	#140000!25,R0	;EXPECTED DATA PLUS ;RXERR & OVRUN
2545	012116	012700	140025				
2546					MOV	@RXDBUF,R1	;ACTUAL
2547	012122	017701	005556		CMP	R0,R1	;COMPARE EXP VS. ACT
2548	012126	020001			BEQ	66\$	
2549	012130	001401			HLT	2	;SPECIFICALLY LOOK AT RXERR & ;OVRUN BITS...THEY BOTH SHOULD BE SET
2550	012132	104002			66\$:		
2551							
2552	012134						
2553	012134	104400					
2554							
2555							
2556							
2557							
2558							
2559							
2560							
2561							
2562	012136	012767	000061	166762	TST49: MOV	#49,TSTNO	;SAVE THIS
2563	012144	012767	012400	166744	MOV	#TST50,NEXT	;GO TO THIS TEST WHEN THRU
2564	012152	052777	000400	005534	BIS	#MRESET,@TXCSR	;MASTER RESET
2565	012160	012777	000000	005522	MOV	#ISYMOD,@PARCSR	;SET THE MODE
2566	012166	052777	000400	005520	BIS	#MRESET,@TXCSR	;MASTER RESET
2567							
2568							
2569	012174	012777	064001	005512	;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE		
2570					MOV	#MTDATA!CLK!MINT!BREAK,@TXCSR	
2571							
2572	012202	012777	000000	005500	;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG		
2573	012210	052777	000020	005462	MOV	#ISYMOD!FIVE!NOPAR!0,@PARCSR	
2574					BIS	#SYNSCH,@RXCSR	;SET SYNC SEARCH
2575	012216	042777	020000	005470			;POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
2576	012224	052777	020000	005462	BIC	#CLK,@TXCSR	;POKE CLK DOWN
2577					BIS	#CLK,@TXCSR	;POKE CLK UP
2578	012232	042777	020000	005454	;POKE CLK TO GET LOGIC INTO SYNCRONIZATION		
2579	012240	052777	020000	005446	BIC	#CLK,@TXCSR	;POKE CLK DOWN
2580	012246	016703	005432		BIS	#CLK,@TXCSR	;POKE CLK UP
2581	012252	012700	000012		MOV	RXDBUF,R3	;SET UP FOR ERROR MESSAGE
2582	012256	012767	000007	166654	MOV	#12,R0	;EXPECTED
2583	012264	012767	000124	166652	MOV	#7,SHIFT	;# OF SHIFTS
					MOV	#124,TEMP1	;DATA CHAR

```

2584 012272 004767 005106 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2585 012276 105777 005376 TSTB @RXCSR ;RXDONE
2586 012302 100401 BMI #64$
2587 012304 104000 HLT ;RXDONE SHOULD BE SET
2588 012306 64$:
2589 012306 017701 005372 MOV @RXDBUF,R1 ;ACTUAL
2590 012312 020001 CMP R0,R1 ;COMPARE EXPECTED VS. ACTUAL
2591 012314 001401 BEQ #65$
2592 012316 104002 HLT 2 ;RECEIVED DATA DID NOT MATCH
2593 ;EXPECTED DATA - CHECK MAINT DATA
2594 ;OR RECEIVER LOGIC
2595 012320 65$:
2596 012320 012767 000007 166612 MOV #7,SHIFT ;# OF SHIFTS
2597 012326 012767 000124 166610 MOV #124,TEMP1 ;DATA CHAR
2598 012334 004767 005044 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2599 ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2600 012340 012767 000007 166572 MOV #7,SHIFT ;# OF SHIFTS
2601 012346 012767 000124 166570 MOV #124,TEMP1 ;DATA CHAR
2602 012354 004767 005024 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2603 012360 012700 140012 MOV #140000!12,R0 ;EXPECTED DATA PLUS
2604 ;RXERR & OVRRUN
2605 012364 017701 005314 MOV @RXDBUF,R1 ;ACTUAL
2606 012370 020001 CMP R0,R1 ;COMPARE EXP VS. ACT
2607 012372 001401 BEQ #66$
2608 012374 104002 HLT 2 ;SPECIFICALLY LOOK AT RXERR &
2609 ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2610 012376 66$:
2611 012376 104400 SCOPE
2612 ;:THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2613 ;:RECEIVER SECTION,IT USES THE ERROR FLAGS
2614 ;:TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2615 ;:(OVRRUN,RXERR)
2616 ;:MODE:ISYMOD
2617 ;:LENGTH:FIVE
2618 ;:CHAR:37
2619 ;:
2620 012400 012767 000062 166520 TST50: MOV #50,TSTNO ;SAVE THIS
2621 012406 012767 012642 166502 MOV #TST51,NEXT ;GO TO THIS TEST WHEN THRU
2622 012414 052777 000400 005272 BIS #MRESET,@TXCSR ;MASTER RESET
2623 012422 012777 000000 005260 MOV #ISYMOD,@PARCSR ;SET THE MODE
2624 012430 052777 000400 005256 BIS #MRESET,@TXCSR ;MASTER RESET
2625 ;
2626 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2627 012436 012777 064001 005250 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2628 ;
2629 ;SET MODE # OF BITS,PARITY SENSE,&LOAD SYNC REG
2630 012444 012777 000000 005236 MOV #ISYMOD!FIVE!NOPAR!0,@PARCSR
2631 012452 052777 000020 005220 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2632 ;POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
2633 012460 042777 020000 005226 BIC #CLK,@TXCSR ;POKE CLK DOWN
2634 012466 052777 020000 005220 BIS #CLK,@TXCSR ;POKE CLK UP
2635 ;POKE CLK TO GET LOGIC INTO SYNCRIZATION
2636 012474 042777 020000 005212 BIC #CLK,@TXCSR ;POKE CLK DOWN
2637 012502 052777 020000 005204 BIS #CLK,@TXCSR ;POKE CLK UP
2638 012510 016703 005170 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
2639 012514 012700 000037 MOV #37,R0 ;EXPECTED

```

```

2640 012520 012767 000007 166412      MOV      #7,SHIFT          ;# OF SHIFTS
2641 012526 012767 000176 166410      MOV      #176,TEMP1       ;DATA CHAR
2642 012534 004767 004644              JSR      PC,RPOKE          ;SHIFT IN THIS CHAR
2643 012540 105777 005134              TSTB    @RXCSR ;RXDONE   )
2644 012544 100401              BMI     64$
2645 012546 104000              HLT
2646 012550              64$:
2647 012550 017701 005130      MOV      @RXDBUF,R1        ;ACTUAL
2648 012554 020001              CMP     R0,R1             ;COMPARE EXPECTED VS. ACTUAL
2649 012556 001401              BEQ     65$
2650 012560 104002              HLT     2                 ;RECEIVED DATA DID NOT MATCH
2651                                ;EXPECTED DATA - CHECK MAINT DATA
2652                                ;OR RECEIVER LOGIC
2653 012562              65$:
2654 012562 012767 000007 166350      MOV      #7,SHIFT          ;# OF SHIFTS
2655 012570 012767 000176 166346      MOV      #176,TEMP1       ;DATA CHAR
2656 012576 004767 004602              JSR      PC,RPOKE          ;SHIFT IN THIS CHAR
2657                                ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2658 012602 012767 000007 166330      MOV      #7,SHIFT          ;# OF SHIFTS
2659 012610 012767 000176 166326      MOV      #176,TEMP1       ;DATA CHAR
2660 012616 004767 004562              JSR      PC,RPOKE          ;SHIFT IN THIS CHAR
2661 012622 012700 140037              MOV      #140000!37,R0    ;EXPECTED DATA PLUS
2662                                ;RXERR & OVRUN
2663 012626 017701 005052      MOV      @RXDBUF,R1        ;ACTUAL
2664 012632 020001              CMP     R0,R1             ;COMPARE EXP VS. ACT
2665 012634 001401              BEQ     66$
2666 012636 104002              HLT     2                 ;SPECIFICALLY LOOK AT RXERR &
2667                                ;OVRUN BITS...THEY BOTH SHOULD BE SET
2668 012640              66$:
2669 012640 104400
2670                                SCOPE
2671                                ;; THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2672                                ;; RECEIVER SECTION, IT USES THE ERROR FLAGS
2673                                ;; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2674                                ;; (OVRUN,RXERR)
2675                                ;; MODE: ISYMOD
2676                                ;; LENGTH: FIVE
2677                                ;; CHAR: 0
2678 012642 012767 000063 166256      TST51: MOV      #51,TSTNO     ;SAVE THIS
2679 012650 012767 013104 166240      MOV      #.EOP,NEXT       ;GO TO THIS TEST WHEN THRU
2680 012656 052777 000400 005030      BIS      #MRESET,@TXCSR   ;MASTER RESET
2681 012664 012777 000000 005016      MOV      #ISYMOD,@PARCSR  ;SET THE MODE
2682 012672 052777 000400 005014      BIS      #MRESET,@TXCSR   ;MASTER RESET
2683
2684                                ;SET MAINT DATA,CLK BREAK,&MAINTENANCE MODE
2685 012700 012777 064001 005006      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
2686
2687                                ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2688 012706 012777 000000 004774      MOV      #ISYMOD!FIVE!NOPAR!0,@PARCSR
2689 012714 052777 000020 004756      BIS      #SYNSCH,@RXCSR   ;SET SYNC SEARCH
2690                                ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2691 012722 042777 020000 004764      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
2692 012730 052777 020000 004756      BIS      #CLK,@TXCSR      ;POKE CLK UP
2693                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2694 012736 042777 020000 004750      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
2695 012744 052777 020000 004742      BIS      #CLK,@TXCSR      ;POKE CLK UP
    
```



2696	012752	016703	004726		MOV	RXDBUF,R3	;SET UP FOR ERROR MESSAGE
2697	012756	012700	000000		MOV	#0,R0	;EXPECTED
2698	012762	012767	000007	166150	MOV	#7,SHIFT	;# OF SHIFTS
2699	012770	012767	000100	166146	MOV	#100,TEMP1	;DATA CHAR
2700	012776	004767	004402		JSR	PC,RPOKE	;SHIFT IN THIS CHAR
2701	013002	105777	004672		TSTB	BRXCSR	;RXDONE
2702	013006	100401			SMT	64\$	
2703	013010	104000			HLT		;RXDONE SHOULD BE SET
2704	013012						
2705	013012	017701	004666		MOV	BRXDBUF,R1	;ACTUAL
2706	013016	020001			CMP	R0,R1	;COMPARE EXPECTED VS. ACTUAL
2707	013020	001401			BEQ	65\$	
2708	013022	104002			HLT	2	;RECEIVED DATA DID NOT MATCH ;EXPECTED DATA - CHECK MAINT DATA ;OR RECEIVER LOGIC
2709							
2710							
2711	013024						
2712	013024	012767	000007	166106	MOV	#7,SHIFT	;# OF SHIFTS
2713	013032	012767	000100	166104	MOV	#100,TEMP1	;DATA CHAR
2714	013040	004767	004340		JSR	PC,RPOKE	;SHIFT IN THIS CHAR
2715							;NOW SHIF? IN A SECOND CHARACTER WITHOUT READING RXDBUF
2716	013044	012767	000007	166066	MOV	#7,SHIFT	;# OF SHIFTS
2717	013052	012767	000100	166064	MOV	#100,TEMP1	;DATA CHAR
2718	013060	004767	004320		JSR	PC,RPOKE	;SHIFT IN THIS CHAR
2719	013064	012700	140000		MOV	#140000!0,R0	;EXPECTED DATA PLUS ;RXERR & OVRRUN
2720							
2721	013070	017701	004610		MOV	BRXDBUF,R1	;ACTUAL
2722	013074	020001			CMP	R0,R1	;COMPARE EXP VS. ACT
2723	013076	001401			BEQ	66\$	
2724	013100	104002			HLT	2	;SPECIFICALLY LOOK AT RXERR & ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2725							
2726	013102						
2727	013102	104400					
2728							

```

2729
2730 ;END OF PASS
2731 ;TYPE NAME OF TEST
2732 ;UPDATE PASS COUNT
2733 ;CHECK FOR EXIT TO ACT-11
2734 ;RESTART TEST
2735
2736 013104 104402 .EOP: TYPE ;TYPE NAME OF TEST
2737 013106 016246 MEPASS
2738 013110 104410 013342 CONVRT ,OUTCRY
2739 013114 104402 015767 TYPE ,DEVICE
2740 013120 105767 166056 TSTS MULTD ;ARE YOU RUNNING MULTIPLE DEVICES ?
2741 013124 001511 BEQ CCC ;NO, JUMP AROUND
2742 011126 005767 166064 TST ACTREG ;ARE ANY DEVICES ACTIVE ?
2743 013132 001007 BNE RUNIT ;YES
2744 013134 104402 016001 TYPE MCOV ;NO
2745 013140 016700 166052 MOV ACTREG, RN ;DISPLAY ACTREG
2746 013144 000000 HALT ;SELECT SOMETHING TO RUN @ ACTREG:
2747 ;SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
2748 013146 000167 166106 JMP .START ;START OVER AGAIN.....YOU Deselected EVERYTHING
2749 013152 062767 000010 166024 RUNIT: ADD #10, BASEADD ;NEXT BLOCK (ADDRESSSES)
2750 013160 062767 000010 166024 ZERC: ADD #10, BASEIV ;NEXT BLOCK (VECTORS)
2751 013166 000241 CLC
2752 013170 006167 165024 POL ROTADD ;UP DATE ROTATING POINTER
2753 013174 103410 BCS 2$ ;IS IT THE LAST DEVICE
2754 ;TO BE TESTED IN THIS PASS ?
2755 013176 036767 166016 166012 BIT ROTADD, ACTREG ;TEST THIS DEVICE FOR ACTIVE STATUS
2756 013204 001762 BEQ RUNIT ;IF NOT ACTIVE, TRY NEXT ADDRESS
2757 013206 004767 000034 JSR PC, REPLAY ;CALCULATE NEW PARAMETERS
2758 013212 000167 000174 JMP RESTRT ;YES IT WAS ACTIVE, TEST THIS DEVICE
2759 013215 012767 000001 165774 2$: MOV #1, ROTADD ;OK! NOW SET UP ROTATING
2760 ;POINTER FOR NEXT MULTIPLE PASS
2761 013224 016767 165756 165752 MOV KEEPADD, BASEADD ;RESTORE BASE ADDRESS
2762 013232 016767 165756 165752 MOV KEEPIV, BASEIV ;RESTORE BASE INTERRUPT VECTORS
2763 013240 004767 000002 JSR PC, REPLAY ;CALC NEW PARAMETERS
2764 013244 000441 BR CCC ;JUMP AROUND REPLAY
2765 013246 016767 165732 004126 REPLAY: MOV BASEADD, DUBASE ;SET UP FOR NEW ADDRESSES
2766 013254 004767 003770 JSR PC, DUADDR ;CREATE NEW ADDRESSES
2767 013260 016767 165726 004436 MOV BASEIV, DURIV ;CREATE DURIV
2768 013266 062767 000002 165716 ADD #2, BASEIV
2769 013274 016767 165712 004424 MOV BASEIV, DURIS ;CREATE DURIS
2770 013302 062767 000002 165702 ADD #2, BASEIV
2771 013310 016767 165676 004412 MOV BASEIV, DUTIV ;CREATE DUTIV
2772 013316 062767 000002 165666 ADD #2, BASEIV
2773 013324 016767 165662 004400 MOV BASEIV, DUTIS ;CREATE DUTIS
2774 013332 016767 004366 165652 MOV DURIV, BASEIV ;RESTORE
2775 013340 000207 RTS PC
2776
2777 013342 000001 OUTCRY: 1
2778 013344 006 002 .BYTE 6,2
2779 013346 017700 RXCSR
2780
2781 013350 CCC:
2782 013350 005067 165560 CLR LSTERR ;CLEAR LAST ERROR PC
2783 013354 005067 165644 CLR ERRFLG ;CLEAR ERROR FLAG
2784 013360 005267 165544 INC PASCNT ;UPDATE PASS COUNT

```

```

2785 013364 016777 165540 165510      MOV      PASCNT,ALIGHTS      ;DISFLAY PASS COUNT
2786 013372 013701 000042      MOV      @#42,R1            ;CHECK FOR ACT-11 OR DDP
2787 013376 001405      BEQ      RESTART           ;IF NOT, CONTINUE TESTING
2788 013400 000005      RESET
2789 013402 004711      LOGICAL: JSR      PC,(7)
2790 013404 000240      NOP
2791 013406 000240      NOP
2792 013410 000240      NOP
2793 013412 012767 000340 164356  RESTART: MOV      @340,PS          ;PREVENT INTERRUPTS (PRIC: 7)
2794 013420 104413      CKSWR          ;CHECK FOR IG
2795 013422 012767 002350 165464      MOV      @TST1,RTRN
2796 013430 000167 166714      JMP      TST1
2797
2798      ;SCOPE LOOP AND INTERATION HANDLER
2799
2800 013434      .SCOPE:
2801      ;**** START OF CODE FOR THE X OR TESTER ****
2802 013434 000424      BR      4$
2803
2804 013436 013746 000004      MOV      @#4,-(SP)          ;IF RUNNING ON THE X OR TESTER CHANGE
2805 013442 012737 013462 000004      MOV      @1$,@#4           ;THIS INSTRUCTION TO A "NOP"(NOP=240)
2806 013450 005737 177060      TST      @#177060          ;SAVE CONTENTS OF ERROR VECTOR
2807 013454 012637 000004      MOV      (SP)+,@#4         ;SET FOR TIME OUT
2808 013460 000404      BR      2$                ;TIME OUT ON X OR ?
2809 013462 022626      1$: CMP      (SP)+,(SP)+     ;RESTORE ERROR VECTOR
2810 013464 012637 000004      MOV      (SP)+,@#4         ;GO TO NEXT TEST
2811 013470 000403      BR      3$                ;CLEAR THE STACK AFTER A TIMEOUT
2812 013472 016767 165420 165414  2$: MOV      NEXT,RTRN      ;RESTORE ERROR VECTOR
2813 013500 016716 165410  3$: MOV      RTRN,(SP)      ;LOOP ON PRESENT TEST
2814 013504 000002      RTI                        ;SET UP NEXT TEST IN RTRN
2815 013506      4$:      ;**** END OF CODE FOR THE X OR TESTER **** ;SET UP STACK FOR RTI
2816 013506 104413      CKSWR          ;CHECK FOR IG
2817 013510 032777 040000 165362  TTST: BIT      @SW14,@SWR      ;LOOP ON CURRENT TEST ?
2818 013516 001407      BEQ      1$
2819 013520 000432      BR      3$
2820 013522 105777 165356      TSTB    @TKCSR            ;TEST TTY FLAG
2821 013526 100027      BPL      3$
2822 013530 017700 165352      MOV      @TKDBR,R0        ;CLR DONE BIT
2823 013534 000412      BR      2$                ;IF A TTY KEY IS STRUCK GO TO NEXT TST
2824 013536 032777 004000 165334  1$: BIT      @SW11,@SWR      ;INHIBIT ITERATIONS ?
2825 013544 001006      BNE      2$
2826 013546 005267 165352      INC      LPCNT
2827 013552 026767 165346 165342      CMP      LPCNT,ICOUNT     ;CHECK FOR ITERATION CNT FINISH
2828 013560 101412      BLOS    3$
2829 013562 105067 165436  2$: CLRB    ERRFLG
2830 013566 005067 165332      CLR      LPCNT
2831 013572 012767 000005 165322      MOV      @5,ICOUNT        ;SET UP ITERATION COUNT
2832 013600 016767 165312 165206      MOV      NEXT,RTRN      ;SET UP NEXT TEST IN RTRN
2833 013606 016716 165302  3$: MOV      RTRN,(SP)      ;SET UP STACK FOR RTI
2834 013612 000002      RTI
2835 013614 001407      BRW: 1407                ;RESTORE "BEQ 1$" INSTRUCTION
2836 013616 000432      BRX: 432                 ;RESTORE "BR 3$" INSTRUCTION
2837
2838      ;CHECK FOR FREEZE ON CURRENT DATA
2839
2840 013620 104413      .SCOPE1: CKSWR          ;CHECK FOR IG
    
```

```

2841 013622 032777 001000 165250      BIT      #SW09, @SWR
2842 013630 001402                      BEQ      1$
2843 013632 016716 165262                      MOV      LOCK, (SP)
2844 013636 000002                      1$: RTI
2845
2846                      ;TELETYPE OUTPUT ROUTINE
2847
2848 013640 010546                      .TYPE:  MOV      R5, -(SP)
2849 013642 017605 000002                      MOV      @2(SP), R5
2850 013646 062766 000002 000002          ADD      #2, 2(SP)
2851 013654 105715                      1$: TSTB     (R5)                ;LOOK FOR "0"
2852 013656 001406                      EEQ      3$
2853 013660 105777 165224          2$: TSTB     @TPCSR                ;TEST DONE BIT
2854 013664 100375                      BPL      2$
2855 013666 112577 165220          MOVB     (R5)+, @TPDBR          ;TYPE CHAR
2856 013672 000770                      BR       1$                    ;DO IT AGAIN UNTIL "0" IS SEEN
2857 013674 012605                      3$: MOV      (SP)+, R5
2858 013676 000002                      RTI
2859
2860                      ;ASCII STRING INPUT ROUTINE
2861
2862 013700 010346                      .INSTR: MOV      R3, -(SP)
2863 013702 010446                      MOV      R4, -(SP)
2864 013704 017667 000004 000010          MOV      @4(SP), .MSG
2865 013712 062766 000002 000004          ADD      #2, 4(SP)
2866 013720 104402                      .INST1: TYPE
2867 013722 000000                      .MSG:   0
2868 013724 012704 017034          MOV      #INBUF, R4
2869 013730 012703 000007          MOV      #7, R3
2870 013734 105777 165144          1$: TSTB     @TKCSR
2871 013740 100375                      BPL      1$
2872 013742 117714 165140          MOVB     @TKDBR, (R4)
2873 013746 142714 000200          BICB     #200, (R4)
2874 013752 121427 000025          CMPB     (R4), #25                ;IS IT <1U>
2875 013756 001003                      BNE      200$
2876 013760 104402 016156          TYPE, MCRLF
2877 013764 000755                      BR       .INST1
2878 013766 122427 000015          200$:  CMPB     (R4)+, #15
2879 013772 001423                      BEQ      INSTR2
2880 013774 117777 165106 165110          MOVB     @TKDBR, @TPDBR
2881 014002 105777 165102          2$: TSTB     @TPCSR
2882 014006 100375                      BPL      2$
2883 014010 005303                      DEC      R3
2884 014012 001350                      BNE      1$
2885 014014 000402                      BR       .INSTG
2886 014016 010346                      .INSTE: MOV      R3, -(SP)
2887 014020 010446                      MOV      R4, -(SP)
2888 014022 104402                      .INSTG: TYPE
2889 014024 016152                      MCM
2890 014026 005737 015314          TST      @RDSW
2891 014032 001402                      BEQ      400$
2892 014034 104402 016156          TYPE, MCRLF
2893 014040 000727                      400$:  BR       .INST1
2894 014042 012604          INSTR2: MOV      (SP)+, R4
2895 014044 012603                      MOV      (SP)+, R3
2896 014046 000002                      RTI
    
```

```

2897
2898
2899
2900 014050 010546
2901 014052 010446
2902 014054 016605 000004
2903 014060 012567 000170
2904 014064 012567 000166
2905 014070 012567 000164
2906 014074 112567 000162
2907 014100 112567 000157
2908 014104 010566 000004
2909 014110 005005
2910 014112 012704 017034
2911 014116 122714 000015
2912 014122 001420
2913 014124 121427 000060
2914 014130 002415
2915 014132 121427 000067
2916 014136 003012
2917 014140 142714 000060
2918 014144 152405
2919 014146 122714 000015
2920 014152 001414
2921 014154 006305
2922 014156 006305
2923 014160 006305
2924 014162 000760
2925 014164 122714 000015
2926 014170 001003
2927 014172 005737 015314
2928 014176 001023
2929 014200 104404
2930 014202 000742
2931
2932
2933
2934 014204 020567 000046
2935 014210 101365
2936 014212 020567 000036
2937 014216 103762
2938 014220 136705 000036
2939 014224 001357
2940
2941
2942
2943 014226 016704 000026
2944 014232 010524
2945 014234 062705 000002
2946 014240 105367 000017
2947 014244 001372
2948 014246 012604
2949 014250 012605
2950 014252 000002
2951 014254 000000
2952 014256 000000

```

;CONVERT ASCII STRING TO OCTAL

```

.PARAM: MOV R5, -(SP)
        MOV R4, -(SP)
        MOV 4(SP), R5
        MOV (R5)+, LOLIM
        MOV (R5)+, HILIM
        MOV (R5)+, DEVADR
        MOV (R5)+, LOBITS
        MOV (R5)+, ADCNT
        MOV R5, 4(SP)
PARAM1: CLR R5
        MOV #INBUF, R4
        CMPB #15, (R4)
        BEQ PARERR
IS:     CMPB (R4), #60
        BLT PARERR
        CMPB (R4), #67
        BGT PARERR
        BICB #60, (R4)
        BISB (R4)+, R5
        CMPB #15, (R4)
        BEQ LIMITS
        ASL R5
        ASL R5
        ASL R5
        BR 1
PARERR: CMPB #15, (R4) ;IS FIRST CHARACTER A (CR)
        BNE 120$
        TST #RDSW ;IS CKSWR ROUTINE BEING USED
        BNE PARTI
120$:   INSTER
        BR PARAM1
LIMITS: CMP R5, HILIM
        BHI PARERR
        CMP R5, LOLIM
        BLO PARERR
        BITB LOBITS, R5
        BNE PARERR
;STORE NUMBER AT SPECIFIED ADDRESS
IS:     MOV DEVADR, R4
        MOV R5, (R4)+
        ADD #2, R5
        DEC9 ADCNT
        BNE IS
PARTI:  MOV (SP)+, R4
        MOV (SP)+, R5
LOLIM: 0
HILIM: 0

```

```

2953 014260 000000          DEVADR: 0
2954 014262 000000          LOBITS: 0
2955                014263          ADRCNT=LOBITS+1
2956
2957                ;SAVE PC OF TEST THAT FAILED AND RO-R5
2958
2959 014264 016667 000004 164702 .SAV05: MOV     4(SP),SAVPC
2960
2961                ;SAVE RO-R5
2962
2963 014272 010567 164672          SV05:  MOV     R5,SAVR5
2964 014276 010467 164664          MOV     R4,SAVR4
2965 014302 010367 164656          MOV     R3,SAVR3
2966 014306 010267 164650          MOV     R2,SAVR2
2967 014312 010167 164642          MOV     R1,SAVR1
2968 014316 010067 164634          MOV     R0,SAVR0
2969 014322 000002          RTI
2970
2971                ;RESTORE RO-R5
2972
2973 014324 016700 164626          .RESG5: MOV     SAVR0,R0
2974 014330 016701 164624          MOV     SAVR1,R1
2975 014334 016702 164622          MOV     SAVR2,R2
2976 014340 016703 164620          MOV     SAVR3,R3
2977 014344 016704 164616          MOV     SAVR4,R4
2978 014350 016705 164614          MOV     SAVR5,R5
2979 014354 000002          RTI
2980
2981                ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
2982
2983 014356 104402          .CONVR: TYPE
2984 014360 016156          MCRLF
2985 014362 010046          .CNVRT: MOV     RO,-(SP)
2986 014364 010146          MOV     R1,-(SP)
2987 014366 010346          MOV     R3,-(SP)
2988 014370 010446          MOV     R4,-(SP)
2989 014372 010546          MOV     R5,-(SP)
2990 014374 017601 000012          MOV     2(12(SP),R1
2991 014400 016767 002470 164542          MOV     TEMP,TEMP3
2992 014406 062766 000002 000012          ADD     #2,12(SP)
2993 014414 012167 000154          MOV     (R1)+,WDRCNT
2994 014420 112167 000152          15:  MOVB  (R1)+,CHRCNT
2995 014424 112167 000147          MOVB  (R1)+,SPACNT
2996 014430 013167 000144          MOV     2(R1)+,BINWRD
2997 014434 016704 000140          25:  MOV     BINWRD,R4
2998 014440 116705 000132          MOVB  CHRCNT,R5
2999 014444 012700 017074          MOV     #TEMP,R0
3000 014450 010403          35:  MOV     R4,R3
3001 014452 042703 177770          BIC     #177770,R3
3002 014456 062703 000060          ADD     #060,R3
3003 014462 110320          MOVB  R3,(R0)+
3004 014464 006204          ASR   R4
3005 014466 042704 100000          BIC     #100000,R4
3006 014472 006204          ASR   R4
3007 014474 006204          ASR   R4
3008 014476 005305          DEC   R5

```

```

;SHIFT FOR NEXT #
;CLUGE TO STOP BIT 15 PROPAGATING.
;DITTO
;DITTO

```

3009	014500	001363				BNE	3\$	
3010	014502	012703	017134			MOV	#MDATA,R3	
3011	014506	114023			4\$:	MOVB	-(R0),(R3)+	
3012	014510	105367	000062			DECB	CHRCNT	
3013	014514	001374				BNE	4\$	
3014	014516	105767	000055			TSTB	SPACNT	
3015	014522	001405				SEQ	6\$	
3016	014524	112723	000040		5\$:	MOVB	#040,(R3)+	
3017	014530	105367	000043			DECB	SPACNT	
3018	014534	001373				BNE	5\$	
3019	014536	105013			6\$:	CLRB	(R3)	
3020	014540	104402				TYPE		
3021	014542	017134				MDATA		
3022	014544	005367	000024			DEC	WRDCNT	
3023	014550	001323				BNE	1\$	
3024	014552	016767	164372	002314		MOV	TEMP3,TEMP	
3025	014560	012605				MOV	(SP)+,R5	
3026	014562	012604				MOV	(SP)+,R4	
3027	014564	012603				MOV	(SP)+,R3	
3028	014566	012601				MOV	(SP)+,R1	
3029	014570	012600				MOV	(SP)+,R0	
3030	014572	000002				RTI		
3031	014574	000000				WRDCNT:	0	
3032	014576	000000				CHRCNT:	0	
3033		014577				SPACNT=	CHRCNT+1	
3034	014600	000000				BINWRD:	0	
3035								
3036								
3037								
3038								
3039								
3040								
3041	014602	017605	000000			.SETFLG:	MOV	2(SP),R5
3042	014606	122767	000116	002220		CMPB	#'N,INBUF	: IS IT "N" ?
3043	014614	001002				BNE	1\$	
3044	014616	105015				CLRB	(R5) ;000	
3045	014620	000406				BR	2\$	
3046	014622	122767	000131	002204	1\$:	CMPB	#'Y,INBUF	; IS IT "Y" ?
3047	014630	001005				BNE	3\$	
3048	014632	112715	177777			MOVB	#-1,(R5)	;377
3049	014636	062716	000002		2\$:	ADD	#2,(SP)	
3050	014642	000002				RTI		
3051	014644	104404			3\$:	INSTR		:RETRY
3052	014646	000755				BR	.SETFLG	
3053								
3054								
3055								
3056								
3057								
3058	014650	011646				.TRPSR:	MOV	(SP),-(SP)
3059	014652	162716	000002			SUB	#2,(SP)	;GET PC OF RETURN
3060	014656	017616	000000			MOV	2(SP),(SP)	;=PC OF TRAP
3061	014662	006316				ASL	(SP)	;GET TRP
3062	014664	042716	177001		TRPOK:	BIC	#177001,(SP)	;MULTIPLY TRAP ARG BY 2
3063	014670	062716	001226			ADD	#.TRPTAB,(SP)	;CLEAR UNWANTED BITS
3064	014674	017616	000000			MOV	2(SP),(SP)	;POINTER TO SUBROUTINE ADDRESS

;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT  
 ;BUFFER TO THE CHARACTERS "N" AND "Y"  
 ;IF THE CHARACTER IS "N" CLEAR THE FLAG  
 ;IF THE CHARACTER IS "Y" SET THE FLAG

;TRAP DISPATCH SERVICE  
 ;ARGUMENT OF TRAP IS EXTRACTED  
 ;AND USED AS OFFSET TO OBTAIN POINTER  
 ;TO SELECTED SUBROUTINE

;GET PC OF RETURN  
 ;=PC OF TRAP  
 ;GET TRP  
 ;MULTIPLY TRAP ARG BY 2  
 ;CLEAR UNWANTED BITS  
 ;POINTER TO SUBROUTINE ADDRESS  
 ;SUBROUTINE ADDRESS

```

3065 014700 000136          JMP      2(SP)+          ;GO TO SUBROUTINE
3066
3067                          ;ERROR HANDLER
3068
3069 014702 104413          .HLT:   CKSWR          ;CHECK FOR 1G
3070 014704 032777 020000 164166 BIT      #SW13,2SWR    ;INHIBIT ERROR TYPE OUT ?
3071 014712 001061          BNE     HALTS
3072 014714 021667 164214 CMP      (SP),LSTERR
3073 014720 001404          BEQ     1$
3074 014722 011667 164206 MOV      (SP),LSTERR
3075 014726 105067 164272 CLR     ERRFLG
3076 014732 104406          1$:     SAVOS
3077 014734 011605          MOV      (SP),R5
3078 014736 162705 000002 SUB      #2,R5
3079 014742 011504          MOV      (R5),R4
3080 014744 006304          ASL     R4
3081 014746 061504          ADD     (R5),R4
3082 014750 006304          ASL     R4
3083 014752 042704 177001 BIC     #177001,R4
3084 014756 062704 017650 ADD     #.ERRTAB,R4
3085 014762 012467 000040 MOV      (R4)+,ERRMSG
3086 014766 012467 000046 MOV      (R4)+,DATAHD
3087 014772 011467 000054 MOV      (R4),DATABP
3088 014776 105767 164222 TST     ERRFLG
3089 015002 001403          BEQ     TYPMSG
3090 015004 005767 000042 TST     DATABP
3091 015010 001014          BNE     TYPDAT
3092 015012 104410          TYPMSG: CONVRT
3093 015014 015144          ERTAB0
3094 015016 112767 177777 164200 MOV     #-1,ERRFLG
3095 015024 104402          TYPE
3096 015026 000000          ERRMSG: 0
3097 015030 005767 000004 TST     DATAHD
3098 015034 001402          BEQ     TYPDAT
3099 015036 104402          TYPE
3100 015040 000000          DATAHD: 0
3101 015042 005767 000004 TYPDAT: TST     DATABP
3102 015046 001402          BEQ     RESREG
3103 015050 104410          CONVRT
3104 015052 000000          DATABP: 0
3105 015054 104407          RESREG: RESOS
3106 015056 005777 164016 HALTS:  TST     2SWR
3107 015062 100005          BPL     EXITER
3108 015064 010046          PUSHRO
3109 015066 016600 000002 MOV      2(SP),R0
3110 015072 000000          HALT
3111 015074 012600          POPRO
3112 015076 104413          EXITER: CKSWR          ;CHECK FOR 1G
3113 015100 005267 164026 INC     ERRCNT
3114 015104 032777 000400 163766 BIT     #SW08,2SWR    ;LOOP ON ERROR ?
3115 015112 001007          BNE     1$
3116 015114 032777 002000 163756 BIT     #SW10,2SWR    ;ESCAPE TO NEXT ON ERROR ?
3117 015122 001407          BEQ     2$
3118 015124 016767 163766 163762 MOV     NEXT,RTRN    ;SET UP FOR NEXT TEST
3119 015132 012706 001100 1$:     MOV     #STACK,SP  ;REINITIALIZE SP
3120 015136 000177 163752 JMP

```



```

3121 015142 000002          2$: R/I
3122 015144 000001          ERTABO: 1
3123 015146 006          002      .BYTE 6,2
3124 015150 001174          SAVPC
3125                                     ;ENTER HERE ON POWER FAILURE
3126
3127
3128 015152 010046          .PFAIL: MOV R0,-(SP) ;SAVE R0-R5 ON PROCESSOR STACK
3129 015154 010146          MOV R1,-(SP)
3130 015156 010246          MOV R2,-(SP)
3131 015160 010346          MOV R3,-(SP)
3132 015162 010446          MOV R4,-(SP)
3133 015164 010546          MOV R5,-(SP)
3134 015166 016746 162632    MOV 24,-(SP)
3135 015172 010667 163774    MOV SP,SAVSP ;SAVE STACK POINTER
3136 015176 012767 015210 162620 MOV #RESTART,24 ;SET UP FOR POWER UP TRAP
3137 015204 000000          HALT ;HALT ON POWER DOWN NORMAL
3138 015206 000777          1$: BR 1$
3139
3140                                     ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
3141
3142 015210 016706 163756    RESTAR: MOV SAVSP,SP ;RESTORE STACK POINTER
3143 015214 012605          MOV (SP)+,R5 ;RESTORE R0-R5
3144 015216 012604          MOV (SP)+,R4
3145 015220 012603          MOV (SP)+,R3
3146 015222 012602          MOV (SP)+,R2
3147 015224 012601          MOV (SP)+,R1
3148 015226 012600          MOV (SP)+,R0
3149 015230 012767 015152 162566 MOV #.PFAIL,24 ;SET UP FOR POWER FAILURE
3150 015236 012767 000340 162532 MOV #340,PS
3151 015244 012706 001100    MOV #STACK,SP
3152 015250 005067 001620    CLR TEMP
3153 015254 005267 001614    1$: INC TEMP
3154 015260 001375          BNE 1$
3155 015262 104410          CONVRT
3156 015264 015306          PFTAB
3157 015266 104402          TYPE
3158 015270 016161          MPFAIL
3159 015272 005067 163726    CLR ERRFLG
3160 015276 005067 163632    CLR LSTERR
3161 015302 000177 163606    JMP #RTRN
3162 015306 000001          PFTAB: 1
3163 015310 006          002      .BYTE 6,2
3164 015312 001114          RTRN
3165
3166
3167                                     ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR 1G TO ALLOW CHANGING
3168                                     ;OF LOC.176.
3169                                     ;LOCATIONS USED:
3170 015314 000000          RDSW: .WORD 0
3171
3172
3173 015316 005737 000042          .CKSWR: TST #42
3174 015322 001042          BNE OUT
3175 015324 022767 000176 163546    CMP #SWREG,SWR ;SOFTWARE SWITCH REGISTER PRESENT
3176 015332 001036          BNE OUT ;NO, GET OUT

```

3177	015334	105777	163544		
3178	015340	100033			
3179	015342	017767	163540	176352	
3180	015350	042767	177600	176344	
3181	015356	122767	000007	176336	
3182	015364	001021			
3183	015366	104402	015444		
3184	015372	005137	015314		
3185	015376	104402	015451		
3186	015402	104411	015436		
3187	015406	104403	015461		
3188	015412	10405			
3189	015414	000000			
3190	015416	177777			
3191	015420	000176			
3192	015422	000	001		
3193	015424	104402	016156		
3194	015430	005037	015314		
3195	015434	000002			
3196	015436	000001			
3197	015440	006	002		
3198	015442	000176			
3199	015444	005015	043536	000	
3200	015451	015	051412	051127	
3201	015456	020075	000		
3202	015461	040	047040	053505	
3203	015466	020075	000		
3204		015472			
3205	015472	005015	042012	030525	
3206	015500	020061	055104	052504	
3207	015506	026501	020104	040524	
3208	015514	042520	040440	006440	
3209	015522	000012			
3210	015524	005015	042526	052103	
3211	015532	051117	040440	042104	
3212	015540	042522	051523	000055	
3213	015546	005015	051461	020124	
3214	015554	042504	044526	042503	
3215	015562	020072	042522	042503	
3216	015570	053111	051105	041440	
3217	015576	047117	051124	046117	
3218	015604	051040	043505	051511	
3219	015612	042524	020122	042101	
3220	015620	051104	051505	026523	
3221	015626	000			
3222	015627	015	040412	042522	
3223	015634	054440	052517	051040	
3224	015642	047125	044516	043516	
3225	015650	046440	046125	044524	
3226	015656	046120	020105	042504	
3227	015664	044526	042503	020123	
3228	015672	020077	054450	047440	
3229	015700	020122	024516	000055	
3230	015706	005015	040514	052123	
3231	015714	042040	053105	041511	
3232	015722	035105	042522	042503	

```

T5TB      @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
TYPE, $CNTG
.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  <15><12>/↑G/
$MSWR:    .ASCIZ  <15><12>/SWR= /
$MNEW:    .ASCIZ  / NEW= /
.EVEN
MTITLE:   .ASCIZ  <15><12><12>/DU11 DZDUA-D TAPE A /<15><12>
MVECTO:   .ASCIZ  <15><12>/VECTOR ADDRESS-/
MREGAD:   .ASCIZ  <15><12>/1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS-/
MMULT:    .ASCIZ  <15><12>/ARE YOU RUNNING MULTIPLE DEVICES ? (Y OR N)-/
MLASTD:   .ASCIZ  <15><12>/LAST DEVICE:RECEIVER CONTROL REGISTER ADDRESS-/

```

3233	015730	053111	051105	041440
3234	015736	047117	051124	045117
3235	015744	051040	043505	051511
3236	015752	042524	020122	042101
3237	015760	051104	051505	026523
3238	015766	000		
3239	015767	075	042504	044526
3240	015774	042503	020040	000
3241	016001	015	044012	053517
3242	016006	047040	053517	041040
3243	016014	047522	047127	041440
3244	016022	053517	020077	027056
3245	016030	051456	046105	041505
3246	016036	020124	047523	042515
3247	016044	044124	047111	020107
3248	016052	047524	051040	047125
3249	016060	040040	041501	051124
3250	016066	043505	000	
3251	016071	015	047412	052125
3252	016076	047440	020106	040522
3253	016104	043516	035105	042522
3254	016112	054524	042520	046040
3255	016120	051501	020124	042504
3256	016126	044526	042503	051040
3257	016134	041530	051123	040440
3258	016142	042104	042522	051523
3259	016150	000055		
3260	016152	020040	000077	
3261	016156	005015	000	
3262	016161	040	050040	053517
3263	016166	051105	043040	044501
3264	016174	052514	042522	020054
3265	016202	051120	043517	040522
3266	016210	020115	042522	052123
3267	016216	051101	020124	052101
3268	016224	052040	051505	020124
3269	016232	047111	050040	047522
3270	016240	051107	051505	000123
3271	016246	005015	047105	020104
3272	016254	043117	050040	051501
3273	016262	020123	040524	042520
3274	016270	040440	000	
3275	016273	015	051012	000
3276	016277	015	052012	051505
3277	016304	020124	041520	000055
3278	016312	005015	047514	045503
3279	016320	047440	020116	042523
3280	016326	042514	052103	042105
3281	016334	052040	051505	037524
3282	016342	024040	020131	051117
3283	016350	047040	026451	000
3284	016355	015	042012	020125
3285	016362	051120	047511	044522
3286	016370	054524	046040	053105
3287	016376	046105	000055	
3288	016402	005015	020043	043117

DEVICE: .ASCIZ /=DEVICE /

MCOW: .ASCIZ <15><12>/HOW NOW BROWN COW? ...SELECT SOMETHING TO RUN @ACTREG/

MRANGE: .ASCIZ <15><12>/OUT OF RANGE:RETYPE LAST DEVICE RXCSR ADDRESS-/

MQM: .ASCIZ / ?/

MCRLF: .ASCIZ <15><12>

MPFAIL: .ASCIZ / POWER FAILURE, PROGRAM RESTART AT TEST IN PROGRESS/

MEPASS: .ASCIZ <15><12>/END OF PASS TAPE A/

MR: .ASCIZ <15><12>/R/

MTSTPC: .ASCIZ <15><12>/TEST PC-/

MLOCK: .ASCIZ <15><12>/LOCK ON SELECTED TEST? (Y OR N)-/

MLEVEL: .ASCIZ <15><12>/DU PRIORITY LEVEL-/

MSYNC: .ASCIZ <15><12>/# OF SYNC CHARS SELECTED ( 1 OR 2)-/

```

3289 016410 051440 047131 020103
3290 016416 044103 051101 020123
3291 016424 042523 042514 052103
3292 016432 042105 024040 030440
3293 016440 047440 020122 024462
3294 016446 000055
3295 016450 005015 051511 051440
3296 016456 041505 054040 044515
3297 016464 020124 052512 050115
3298 016472 051105 021440 020066
3299 016500 047111 020077 054450
3300 016506 047440 020122 024516
3301 016514 000055
3302 016516 005015 051511 051440
3303 016524 041505 051040 041505
3304 016532 045040 046525 042520
3305 016540 020122 032443 044440
3306 016546 037516 024040 020131
3307 016554 051117 047040 026451
3308 016562 000
3309 016563 015 044412 020123
3310 016570 050117 020124 046103
3311 016576 020122 047105 041101
3312 016604 042514 045040 046525
3313 016612 042520 020122 032043
3314 016620 044440 037516 024040
3315 016626 020131 051117 047040
3316 016634 026451 000
3317 016637 015 040412 042522
3318 016644 054440 052517 051040
3319 016652 047125 044516 043516
3320 016660 044440 020116 040515
3321 016666 047111 020124 047515
3322 016674 042504 042440 052130
3323 016702 051105 040516 037514
3324 016710 005015 040401 042116
3325 016716 027040 027056 027056
3326 016724 042040 020117 047531
3327 016732 020125 040510 042526
3328 016740 052040 042510 042440
3329 016746 052130 051105 040516
3330 016754 020114 047515 042504
3331 016762 020115 054502 040520
3332 016770 051523
3333 016772 005015 045001 046525
3334 017000 042520 020122 047503
3335 017006 047116 041505 047524
3336 017014 020122 047117 037440
3337 017022 054450 047440 020122
3338 017030 024516 000055
3339
3340
3341
3342
3343 017034 000040
3344 017074 000040

```

MWIRE6: .ASCIZ <15><12>/IS SEC XMIT JUMPER #6 IN? (Y OR N)-/

MWIRE5: .ASCIZ <15><12>/IS SEC REC JUMPER #5 IN? (Y OR N)-/

MWIRE4: .ASCIZ <15><12>/IS OPT CLR ENABLE JUMPER #4 IN? (Y OR N)-/

MEXTJ: .ASCII <15><12>/ARE YOU RUNNING IN MAINT MODE EXTERNAL?/

.ASCII <15><12><1>/AND ..... DO YOU HAVE THE EXTERNAL MODEM BYPASS/

.ASCIZ <15><12><1>/JUMPER CONNECTOR ON ?(Y OR N)-/

.EVEN

;BUFFERS FOR INPUT-OUTPUT

INBUF: .BLKB 40  
TEMP: .BLKB 40

```

3345 017134 000040
3346
3347
3348
3349
3350
3351 017174 006367 000044
3352 017200 006367 000040
3353 017204 006367 000034
3354 017210 006367 000030
3355 017214 006367 000024
3356 017220 016767 000020 000020
3357 017226 162767 000001 000012
3358 017234 042767 000037 000004
3359 017242 000207
3360 017244 000240
3361 017246 000200
3362
3363
3364 017250 016767 000126 000422
3365 017256 005267 000120
3366 017262 016767 000114 000412
3367 017270 005267 000106
3368 017274 016767 000102 000402
3369 017302 016767 000074 000400
3370 017310 005267 000066
3371 017314 016767 000062 000364
3372 017322 016767 000054 000362
3373 017330 005267 000046
3374 017334 016767 000042 000352
3375 017342 005267 000034
3376 017346 016767 000030 000342
3377 017354 005267 000022
3378 017360 016767 000016 000332
3379 017366 005267 000010
3380 017372 016767 000004 000322
3381 017400 000207
3382 017402 000000
3383
3384
3385
3386
3387 017404 042777 040000 000302
3388 017412 005067 161530
3389 017416 006067 161522
3390 017422 006067 161520
3391 017426 006267 161514
3392 017432 042767 100000 161506
3393 017440 056777 161502 000246
3394 017446 042777 020000 000240
3395 017454 052777 020000 000232
3396 017462 005367 161452
3397 017466 001346
3398 017470 000207
3399
3400

```

```

MATA: .BLKB 40
;*****
;UTILITIES
;*****

;THIS UTILITY CALCULATES PRIORITY LEVEL
DULEV: ASL DUPRT ;SHIFT LEFT
        ASL DUPRT
        ASL DUPRT
        ASL DUPRT
        ASL DUPRT
        MOV DUPRT,LESS1 ;MOVE THIS TO LESS1
        SUB #1,LESS1 ;CREATE LESS1
        BIC #37,LESS1 ;CLEAR TNZVC
        RTS PC

DUPRT: LEVEL5
LESS1: LEVEL4 ;LEVEL TO ALLOW INTERRUPTS

;NEW DU ADDRESSES
DUADDR: MOV DUBASE,RXCSR ;XXX0
        INC DUBASE
        MOV DUBASE,HRXCSR ;XXX1
        INC DUBASE
        MOV DUBASE,RXDBUF ;XXX2
        MOV DUBASE,PARCSR ;XXX2
        INC DUBASE
        MOV DUBASE,HRXDBUF ;XXX3
        MOV DUBASE,HPARCSR ;XXX3
        INC DUBASE
        MOV DUBASE,TXCSR ;XXX4
        INC DUBASE
        MOV DUBASE,HTXCSR ;XXX5
        INC DUBASE
        MOV DUBASE,TXDBUF ;XXX6
        INC DUBASE
        MOV DUBASE,HTXDBUF ;XXX7
        RTS PC
DUBASE: 0

;THIS UTILITY POKES THE MAINT DATA BASED UPON THE
;INFORMATION CONTAINED IN TEMP1 AND IT IS
;SHIFTED IN BY THE CONTENTS OF SHIFT
RPOKE: BIC #MATA,@TXCSR
        CLR TEMP2
        ROR TEMP1 ;FORCE CARRY
        ROR TEMP2 ;PICK UP CARRY IN BIT 15
        ASR TEMP2 ;SHIFT INTO BIT 14
        BIC #BIT15,TEMP2 ;CLR BIT 15
        BIS TEMP2,@TXCSR ;POKE MAINT DATA
        BIC #CLK,@TXCSR ;POKE CLK
        BIS #CLK,@TXCSR
        DEC SHIFT
        BNE RPOKE
        RTS PC

```

;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR

```

3401 017472 016767 161446 161446 0008: MOV TEMP1,TEMP2 ;SAVE TEMP1
3402 017500 005067 161444 CLR TEMP3
3403 017504 012727 000010 MOV #8.,(PC)+
3404 017510 000000 1S: 0
3405 017512 006067 161430 2S: ROR TEMP2
3406 017516 005567 161426 ADC TEMP3
3407 017522 005367 177762 DEC 1S
3408 017526 001371 BNE 2S
3409 017530 006067 161414 ROR TEMP3
3410 017534 103404 BCS 3S
3411 017536 052767 000400 161400 BIS #BIT8,TEMP1 ;SET ODD PARITY
3412 017544 000403 BR 4S
3413 017546 042767 000400 161370 3S: BIC #BIT8,TEMP1 ;CLR EVEN PARITY
3414 ;TEMP1 NOW HAS ODD PARITY CHARACTER
3415 017554 000207 4S: RTS PC
3416
3417 ;THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3418 017556 016767 161362 161362 EVEN8: MOV TEMP1,TEMP2 ;SAVE TEMP1
3419 017564 005067 161360 CLR TEMP3
3420 017570 012727 000010 MOV #8.,(PC)+
3421 017574 000000 1S: 0
3422 017576 006067 161344 2S: ROR TEMP2
3423 017602 005567 161342 ADC TEMP3
3424 017606 005367 177762 DEC 1S
3425 017612 001371 BNE 2S
3426 017614 006067 161330 ROR TEMP3
3427 017620 103004 BCC 3S
3428 017622 052767 000400 161314 BIS #BIT8,TEMP1 ;SET EVEN PARITY
3429 017630 000403 BR 4S
3430 017632 042767 000400 161304 3S: BIC #BIT8,TEMP1 ;CLR ODD PARITY
3431 ;TEMP1 NOW HAS EVEN PARITY CHARACTER
3432 017640 000207 4S: RTS PC
3433 017642 062716 000002 TRPREG: ADD #2,(SP) ;ALLOW IT TO "CRUNCH" INTO HLT BACK
3434 ;IN MAIN PART OF THE PROGRAM
3435 017646 000002 RTI
3436 ;ERROR HLT TABLE
3437 017650 017734 .ERRTAB: EMO ;HLT 0 BIT ERPOR (GENERAL)
3438 017652 000000 0
3439 017654 000000 0
3440 017656 017750 EM1 ;HLT 1 REGISTER ERROR
3441 017660 020121 DH1
3442 017662 020142 DT1
3443 017664 020012 EM2 ;HLT 2 RECEIVER ERROR
3444 017666 020121 DH1
3445 017670 020142 DT1
3446 017672 020054 EM3 ;HLT 3 TRANSMITTER ERROR
3447 017674 020121 DH1
3448 017676 020142 DT1
3449
3450 017700 160040 RXCSR: 160040 ;DEFAULT DU ADDRESSES
3451 017702 160041 HRXCSR: 160041
3452 017704 160042 RXDBUF: 160042
3453 017706 160043 HRXDBUF: 160043
3454 017710 160042 PARCSR: 160042
3455 017712 160043 HPARCSR: 160043
3456 017714 160044 TXCSR: 160044

```

3457	017716	160045			HTXCSR: 160045
3458	017720	160046			TXDBUF: 160046
3459	017722	160047			HTXDBUF: 160047
3460					:DEFAULT DU VECTORS
3461	017724	000770			DURIV: 770 ;REC INTR VECTOR
3462	017726	000772			DURIS: 772 ;REC INTR STATUS
3463	017730	000774			DUTIV: 774 ;XMIT INTR VECTOR
3464	017732	000776			DUTIS: 776 ;XMIT INTR STATUS
3465					:ERROR MESSAGES
3466	017734	036440	042440	051122	EMO: .ASCIZ / = ERROR PC/
3467	017742	051117	050040	000103	
3468	017750	036440	051040	043505	EM1: .ASCIZ / = REGISTER ERROR PC/<15><12><1>/REGISTER /
3469	017756	051511	042524	020122	
3470	017764	051105	047522	020122	
3471	017772	041520	005015	051001	
3472	020000	043505	051511	042524	
3473	020006	020122	000040		
3474	020012	036440	051040	041505	EM2: .ASCIZ / = RECEIVER ERROR PC/<15><12><1>/REGISTER /
3475	020020	044505	042526	020122	
3476	020026	051105	047522	020122	
3477	020034	041520	005015	051001	
3478	020042	043505	051511	042524	
3479	020050	020122	000040		
3480	020054	036440	052040	040522	EM3: .ASCIZ / = TRANSMITTER ERROR PC/<15><12><1>/REGISTER /
3481	020062	051516	044515	052124	
3482	020070	051105	042440	051122	
3483	020076	051117	050040	006503	
3484	020104	000412	042522	044507	
3485	020112	052123	051105	020040	
3486	020120	000			
3487					:DATA HEADERS FOR ERROR MESSAGES
3488	020121	105	050130	041505	DH1: .ASCIZ /EXPECTED ACTUAL/
3489	020126	042524	020104	040440	
3490	020134	052103	040525	000114	
3491					.EVEN
3492					:DATA TABLES FOR ERROR MESSAGES
3493	020142	000003			DT1: 3
3494	020144	006	004		.BYTE 6,4
3495	020146	001164			SAVR3 ;REGISTER
3496	020150	006	004		.BYTE 6,4
3497	020152	001156			SAVR0 ;EXPECTED DATA
3498	020154	006	002		.BYTE 6,2
3499	020156	001160			SAVR1 ;ACTUAL DATA
3500		000001			.END









# E06

DZDUA-B MACY11 27.1006) 17-AUG-76 08:17 PAGE 73

DZDUAD.P11 04-AUG-76 00:00

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0069

OUT2	007432	1928	2012*																
OUT3	007562	1930	2062*																
OVRUN=	040000	696*	1785																
PARAM =	104405	837*	911	922	943	968	978	1049	3188										
PARAM1	014110	2909*	2930																
PARCSR	017710	1096*	2119*	2126*	2149*	2156*	2188*	2195*	2269*	2276*	2550*	2357*	2431*	2438*					
		2507*	2514*	2565*	2572*	2623*	2630*	2681*	2688*	3369*	3454*								
PAREN =	001000	700*																	
PARER =	010000	698*	1761																
PARERR	014164	2912	2914	2916	2925*	2935	2937	2939											
PARTI	014246	2928	2948*																
PASCNT	001130	770*	868*	2784*	2785														
PFTAB	015306	3156	3162*																
POPPO =	012600	644*	3111																
POP1SP=	005726	642*																	
POP2SP=	022626	646*																	
PS =	177776	636*	863*	1022*	1553*	2793*	3150*												
PUSHRO=	010046	643*	3108																
PUSH1S=	005746	641*																	
PUSH2S=	024646	645*																	
ROSM	015314	2890	2927	3170*	3184*	3194*													
REACT=	004000	682*	1701	2127	2132	2137	2157	2162	2167	2217	2228	2233	2298	2309					
		2314	2379	2390	2395	2460	2471	2476											
REPLAY	013246	2757	2763	2765*															
RESREG	015054	3102	3105*																
RESTAR	015210	3136	3142*																
RESTR	013412	2758	2787	2793*															
RESOS =	104407	841*	3105																
RING =	040000	679*																	
RINTEN=	000100	687*	1293	1294	1298	1299	1304	1306											
ROTA00	001220	810*	938*	950*	952	954*	959	962*	2752*	2755	2759*								
RPOKE	017404	2207	2223	2288	2304	2369	2385	2450	2466	2526	2540	2544	2584	2598					
		2602	2642	2656	2660	2700	2714	2718	3387*	3397									
RTRN	001114	764*	873*	1052	1056*	1058	2795*	2812*	2813	2832*	2833	3118*	3120	3161					
		3164																	
RTS =	000004	691*	1181	1182	1186	1187	1192	1198	1203	1905	1971	2028	2078						
RUNA =	000000	1	52	69	135	576													
RUNB =	*****	20	57	69	138	581													
RUNC =	*****	20	57	69	138	581													
RUND =	*****	20	57	69	138	581													
RUNE =	*****	20	57	72	138	581													
RUNF =	*****	20	57	72	138	581													
RUNIT	013152	2743	2749*	2756															
RXCSR	017700	1066*	1148*	1149	1153*	1154	1159*	1165	1170	1181*	1182	1186*	1187	1192*					
		1198	1203	1214*	1215	1219*	1220	1225*	1271	1236	1247*	1248	1252*	1253					
		1258*	1260	1270*	1271	1275*	1276	1281*	1273	1293*	1294	1298*	1299	1304*					
		1306	1316*	1317	1321*	1322	1327*	1329	1550*	1554	1592*	1593	1595	1599*					
		1600	1689	1701	1713	1807*	1809	1810	1866*	1877	1878	1883*	1894	1899					
		1905*	1916	1921	1931*	1942	1948	1955*	1966	1971*	1982	1987*	1998	2003					
		2012*	2023	2028*	2039	2045*	2056	2062*	2073	2078*	2089	2094*	2105	2127					
		2131*	2132	2136*	2137	2157	2161*	2162	2166*	2167	2197*	2213	2217	2224					
		2228	2232*	2233	2237	2247	2278*	2294	2298	2305	2309	2313*	2314	2318					
		2328	2359*	2375	2379	2386	2390	2394*	2395	2349	2409	2440*	2456	2460					
		2467	2471	2475*	2476	2480	2490	2515*	2527	2573*	2585	2631*	2643	2689*					
		2701	2779	3364*	3450*														
PXCBUF	017704	1081*	1748	1750	1761	1773	1785	1797	1851	1852	2196	2242	2277	2323					

# F06

DZDUJ-A-B MACY11 27(1006) 17-AUG-76 08:17 PAGE 74  
 DZDUJAD.P11 04-AUG-76 00:00

## CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0070

	2358	2404	2439	2485	2522	2531	2547	2580	2589	2605	2638	2647	2663
	2696	2705	2721	3368*	3452*								
RXDONE= 000200	686#	1689											
RXERR = 100000	655#	1797											
SAVPC = 001174	791#	2959*	3124										
SAVRO = 001156	784#	2968*	2973	3497									
SAVR1 = 001160	785#	2967*	2974	3499									
SAVR2 = 001162	786#	2966*	2975										
SAVR3 = 001164	787#	2965*	2976	3495									
SAVR4 = 001166	788#	2964*	2977										
SAVRS = 001170	789#	2963*	2978										
SAVSP = 001172	790#	3135*	3142										
SAVOS = 104406	839#	3076											
SCOPE = 104400	827#	1074	1089	1104	1119	1134	1142	1175	1208	1241	1264	1287	1310
	1333	1356	1379	1402	1425	1448	1473	1498	1521	1544	1567	1584	1618
	1647	1680	1694	1706	1718	1729	1741	1754	1766	1778	1790	1802	1828
	1843	1857	2110	2141	2171	2252	2333	2414	2495	2553	2611	2669	2727
SCOP1 = 104401	829#												
SEND = 000020	722#	1385	1386	1390	1391	1396	1398						
SEREC = 001200	795#	1010	1929										
SETFLG= 104412	847#	932	1005	1009	1013	1017	1036						
SEVEN = 004000	708#	2357											
SEXMIT = 001177	794#	1006	1927										
SHIFT = 001140	777#	2205*	2221*	2286*	2302*	2367*	2383*	2448*	2464*	2524*	2538*	2542*	2582*
	2596*	2600*	2640*	2654*	2658*	2698*	2712*	2716*	3396*				
SIX = 002000	707#	2276											
SPACNT= 014577	2995#	3014	3017*	3033#									
SRO = 002000	683#												
STACK = 001100	637#	864	1023	3119	3151								
STD = 000010	690#	1214	1215	1219	1220	1225	1231	1236	1931	1987	2012	2062	
STFLG = 001223	816#	867*											
STPSYN= 000400	685#	1316	1317	1321	1322	1327	1329						
SVOS = 014272	2963#												
SWR = 001100	755#	879*	884	888*	894	897	1032	1045	2817	2824	2841	3070	3106
	3114	3116	3175										
SWREG = 000176	746#	888	894	3175	3191	3198							
SWREGC = 015436	3186	3196#											
SW00 = 000001	617#	897											
SW01 = 000002	616#	1045											
SW02 = 000004	615#												
SW03 = 000010	614#												
SW04 = 000020	613#												
SW05 = 000040	612#												
SW06 = 000100	611#												
SW08 = 000400	610#	3114											
SW09 = 001000	609#	2841											
SW10 = 002000	608#	3116											
SW11 = 004000	607#	2824											
SW12 = 010000	606#												
SW13 = 020000	605#	3070											
SW14 = 040000	604#	2817											
SW15 = 100000	603#												
SYNCNO = 001176	793#	994*	998*	2211	2292	2373	2454						
SYNEXT= 020000	704#	2119	2126										
SYNINT= 030000	703#	2188	2195	2269	2276	2350	2357	2431	2438				
SYNSCH= 000020	689#	1247	1248	1252	1253	1258	1260	2131	2136	2161	2166	2197	2232





\$N	= 000063	587#	1062	1064#	1077	1079#	1092	1094#	1107	1109#	1122	1124#	1127	1139#
		1146	1148#	1179	1181#	1212	1214#	1245	1247#	1268	1270#	1291	1293#	1314
		1316#	1337	1339#	1360	1362#	1383	1385#	1406	1408#	1429	1431#	1454	1456#
		1479	1481#	1502	1504#	1525	1527#	1548	1550#	1571	1573#	1587	1589#	1623
		1625#	1652	1654#	1686	1688#	1698	1700#	1710	1712#	1722	1724#	1733	1735#
		1745	1747#	1758	1760#	1770	1772#	1782	1784#	1794	1796#	1805	1807#	1833
		1835#	1848	1850#	1864	1866#	2116	2118#	2146	2148#	2185	2187#	2266	2268#
		2347	2349#	2428	2430#	2504	2506#	2562	2564#	2620	2622#	2678	2680#	2730#
\$Y	= 000015	819#	827	829#	831#	833#	835#	837#	839#	841#	843#	845#	847#	849#
	= 020160	851#	853#											
		731#	734#	744#	751#	1140	1561	1565	1597	1602	1610	1616	1628	1635
		1645	1661	1668	1678	1726	1752	1817	1824	1841	1855	1873	1881	1897
		1902	1919	1924	1946	1952	1969	1985	2001	2006	2026	2043	2059	2076
		2092	2108	3204#	3343#	3344#	3345#							
.BEGIN	002156	899	1022#											
.CKSWR	015316	850	3173#											
.CNTLU	015372	852	3184#											
.CNVRT	014362	846	2985#											
.CONVR	014356	844	2983#											
.EOP	013104	2679	2736#											
.ERRTA	017650	3084	3437#											
.HLT	014702	737	3069#											
.INSTE	014016	836	2886#											
.INSTG	014022	2885	2888#											
.INSTR	013700	834	2862#											
.INST1	013720	2866#	2877	2893										
.MSG	013722	2864#	2867#	3179*	3180*	3181								
.PARAM	014050	838	2900#											
.PFAIL	015152	735	865	3128#	3149									
.RESOS	014324	842	2973#											
.SAVOS	014264	840	2959#											
.SCOPE	013434	828	2800#											
.SCOPI	013620	830	2840#											
.SETFL	014502	848	3041#	3052										
.START	001260	747	863#	873	2748									
.TRPSR	014550	739	3058#											
.TRPTA	001226	824#	3063											
.TYPE	013640	832	2848#											

HLT	647#	1068	1071	1083	1086	1098	1101	1113	1116	1128	1131	1141	1151	1156	1167
	1172	1184	1189	1200	1205	1217	1222	1233	1238	1250	1255	1262	1273	1278	1285
	1296	1301	1308	1319	1324	1331	1342	1347	1354	1365	1370	1377	1388	1393	1400
	1411	1416	1423	1434	1439	1446	1459	1464	1471	1484	1489	1496	1507	1512	1519
	1520	1535	1542	1562	1566	1576	1581	1598	1603	1611	1617	1629	1636	1646	1662
	1669	1679	1691	1703	1715	1727	1738	1753	1763	1775	1787	1799	1818	1825	1842
	1856	1882	1898	1903	1920	1925	1947	1953	1970	1986	2002	2007	2027	2044	2060
	2077	2093	2109	2129	2134	2139	2159	2164	2169	2215	2219	2226	2230	2235	2239
	2245	2249	2296	2300	2307	2311	2316	2320	2326	2330	2377	2381	2388	2392	2397
	2401	2407	2411	2458	2462	2469	2473	2478	2482	2488	2492	2529	2534	2550	2587
	2592	2608	2645	2650	2666	2703	2708	2724							
PRGEN0	587#	2729													
PRGFRT	587#	588													
PUSSYF	587#														
RSETUP	587#	2118	2148	2187	2268	2349	2430	2506	2564	2622	2680				
TSETUP	587#														
SBEGIN	587#	1019													
SBINAR	587#														
SBUFFE	587#	3340													
SCABLE	587#	1884	1906	1932	1956	1972	1988	2013	2029	2046	2063	2079	2095		
SCATCH	587#	730													
SCLRVE	587#	879													
SCONVR	587#	2980													
SOMA	587#														
SEOP	587#	2729													
SGETFL	587#	930	1003	1007	1011	1015	1034								
SGETPA	587#	909	920	940	966	975	1047								
SGETSY	587#	985													
SHEADE	587#	588													
SHLT	587#	3066													
SINSTR	587#	2859													
SISOB	587#														
SMATCH	587#														
SMRR	587#	1683	1695	1707	1730	1755	1767	1779	1791						
SMRW	587#	1143	1176	1209	1242	1265	1288	1311	1334	1357	1380	1403	1426	1451	1476
	1499	1522													
SMRW	587#	1568													
SMSG	587#	3205													
SPARAM	587#	2897													
SPFAIL	587#	3125													
SPOKE	587#														
SPOKER	587#	2201	2282	2363	2444	2519	2577	2635	2693						
SRCNET	587#														
SRECAC	587#	2172	2253	2334	2415										
SREG	587#	2956													
SRESET	587#	1160	1193	1226	1259	1282	1305	1328	1351	1374	1397	1420	1443	1468	1493
	1516	1539	1578	1589	1604	1639	1672	1688	1700	1712	1724	1735	1747	1760	1772
	1784	1796	1808	1836	1850	1867	2118	2120	2148	2150	2187	2189	2268	2270	2349
	2351	2430	2432	2506	2508	2564	2566	2622	2624	2680	2682				
SRXACT	587#	2112	2142												
SSCOPE	587#	2797													
SSCOP1	587#	2837													
SSETFL	587#	3035													
SSETVE	587#	731													
SSTART	587#	855													
SSTRIP	587#														



SSYMBO	587#	600													
SSYNCR	587#	2197	2278	2359	2440	2515	2573	2631	2689						
STRAPS	587#	819													
STRPAR	587#														
STRPDE	587#	827	829	831	833	835	837	839	841	843	845	947	849	851	
STRPSR	587#	3053													
STSTNO	587#	1062	1077	1092	1107	1122	1137	1146	1179	1212	1245	1268	1291	1314	1337
	1360	1383	1406	1429	1454	1479	1502	1525	1548	1571	1587	1623	1652	1686	1698
	1710	1722	1733	1745	1758	1770	1782	1794	1805	1833	1849	1864	2116	2146	2185
	2266	2347	2428	2504	2562	2620	2678								
STYPE	587#	2845													
SUNIBU	587#	1060	1075	1090	1105	1120									
SVARIA	587#	750													
SWORDF	587#														
SWORDO	587#	2496	2554	2612	2670										
SWORDP	587#														

. ABS. 020160 000

ERRORS DETECTED: 0  
DEFAULT GLOBALS GENERATED: 0

DZDUAD, DZDUAD/CRF/CPU:20/SOL=HELLO.P11/EQ:RUNA,PAPA.P11,KEET.P11,DZDUAD.P11  
RUN-TIME: 24 36 3 SECONDS  
RUN-TIME RATIO: 81/64=1.2  
CORE USED: 18K (36 PAGES)