

DUP11

OFLNE SDLC RCVR
CZDPCD0

AH-8581D-MC

COPYRIGHT © 75-79
FICHE 1 OF 1

MAY 1979

digital
MADE IN USA

IDENTIFICATION

PRODUCT CODE: AC-8580D-MC
PRODUCT NAME: CZDPCDO DUP11 OFLINE SDLC RCVR
DATE: JANUARY 1979
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 OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL

COPYRIGHT (C) 1975, 1979 BY DIGITAL EQUIPMENT CORPORATION

1.0 ABSTRACT

THE FUNCTION OF THE DUP11 DIAGNOSTICS IS TO VERIFY THAT THE OPTION OPERATES ACCORDING TO SPECIFICATIONS. THE DIAGNOSTICS VERIFY THAT THERE ARE NO MALFUNCTIONS AND THAT ALL OPERATIONS OF THE DUP11 ARE CORRECT IN ITS ENVIRONMENT. PARAMETERS MAY BE SET TO ALERT DIAGNOSTICS AS TO THE DUP11 CONFIGURATION BY ANSWERING THE PARAMETER DIALOG (LOAD ADDRESS=200, START ADDRESS=1). ALL QUESTIONS SHOULD BE ANSWERED AND THEN EACH DIAGNOSTIC WILL "OVERLAY" THESE PARAMETERS WHICH ARE STORED IN THE "STATUS TABLE" (SEE SECTION 8.4). THE ALTERNATIVE TO THE PARAMETER DIALOG IS DEFAULT PARAMETERS (SEE SECTION 8.5).

THE DIAGNOSTICS WILL RUN UP TO EIGHT CONSECUTIVELY ADDRESSED AND CONSECUTIVELY VECTORED DUP11'S IN A CHAIN MODE, I.E., RUNNING THE DIAGNOSTIC COMPLETELY FOR ONE DEVICE BEFORE STARTING THE NEXT.

CZDPC TESTS ALL RECEIVER SDLC FUNCTIONS IN MAINTENANCE INTERNAL MODE, THAT IS, CLOCKING OF THE DEVICE IS DONE BY THE PROGRAM. THE DEVICE IS SET UP, A SPECIFIC NUMBER OF HALF-CLOCKS ARE DONE, AND A TEST IS MADE FOR A SIGNIFICANT EVENT.

IN CHECKING DATA, THE SOFTWARE EMULATES THE HARDWARE AND USES THE PROCESSOR CARRY BIT AFTER A ROTATE TO PROVIDE AN INPUT TO THE RECEIVER VIA THE MAINTENANCE INPUT DATA BIT. THE PROGRAM CAN THEN LOAD THE RECEIVER DATA BUFFER SERIALLY, WITHOUT USING THE TRANSMITTER.

THE RECEIVER BCC IS CHECKED USING THE CRC.CCITT POLYNOMIAL IN THE SAME WAY AS DATA, WITH ONE EXCEPTION--THE BCC IS CALCULATED FIRST BY THE PROGRAM AND THEN COMPARED TO THE RECEIVER OUTPUT.

CZDPC CHECKS ALL MODEM CONTROL AND INTERRUPT LOGIC DEPENDING ON THE PARAMETER INFORMATION SUPPLIED THROUGH THE OVERLAY MAP. IN ADDITION, ALL EIA GATES, WITH THE EXCEPTION OF THE DATA GATES, ARE CHECKED.

CURRENTLY THERE ARE THREE OFF-LINE DIAGNOSTICS THAT ARE TO BE RUN IN SEQUENCE TO ENSURE THAT IF AN ERROR SHOULD OCCUR IT WILL BE DETECTED AT AN EARLY STAGE AND ESTABLISH THAT DIAGNOSIS OF THE ERROR WILL BE IMMEDIATE TO DISCOVERING THE PROBLEM.

NOTE: ADDITIONAL DIAGNOSTICS MAY BE ADDED IN THE FUTURE.

THE THREE DIAGNOSTICS ARE:

1. CZDPB [REV] BASIC AND OFFLINE TRANSMITTER TESTS
2. CZDPC [REV] OFFLINE RECEIVER AND MODEM CONTROL AND INTERRUPT TESTS
3. CZDPD [REV] OFFLINE SDLC AND DECMODE DATA AND FUNCTION TESTS

NOTE: THERE IS A FOURTH PROGRAM, TAPE CZDPE [REV] WHICH IS A QUICK-VERIFY TAPE THAT REQUIRES ANSWERING A DIALOG. ITS FUNCTION IS TO ENABLE THE OPERATOR TO QUICKLY DETERMINE IF THERE IS A PROBLEM WITH THE DEVICE. SEE THE DOCUMENTATION IN THAT LISTING FOR MORE INFORMATION.

2.0 REQUIREMENTS

2.1 EQUIPMENT

ANY PDP11 FAMILY CPU (WITH MINIMUM 8K MEMORY)
ASR 33 (OR EQUIVALENT)
DUP11

2.2 STORAGE

PROGRAM WILL USE ALL 8K OF MEMORY EXCEPT WHERE ABS AND
BOOTSTRAP LOADER RESIDE. LOCATION 1500 THRU 1560 ARE
ESPECIALLY TO BE NOTED AND LEFT UNTOUCHED BY THE OPERATOR
AFTER THE DUP11 PARAMETER DIALOG HAS BEEN EXECUTED OR AFTER
THE DEFAULT SETUP HAS BEEN DONE.

3.0 LOADING PROCEDURE

3.1 METHOD

ALL PROGRAMS ARE IN ABSOLUTE FORMAT AND ARE LOADED USING THE
ABSOLUTE LOADER. NOTE: IF THE DIAGNOSTICS ARE ON A MEDIA
SUCH AS DISK, MAGTAPE, DECTAPE, OR CASSETTE FOLLOW
INSTRUCTIONS FOR THE MONITOR WHICH HAS BEEN PROVIDED ON THAT
SPECIFIC MEDIA.

ABSOLUTE LOADER STARTING ADDRESS - **500

MEMORY	SIZE
	(*)=
8k	37
12k	57
16k	77
20k	117
24k	137
28k	157

3.1.1 PLACE ADDRESS OF ABS LOADER INTO SWITCH REGISTER. (ALSO PLACE
'HALT' SW UP)

3.1.2 DEPRESS 'LOAD ADDRESS' KEY ON CONSOLE AND RELEASE.

3.1.3 DEPRESS 'START KEY' ON CONSOLE AND RELEASE (PROGRAM SHOULD NOW
BE LOADING INTO CPU)

4.0 STARTING PROCEEDURE

- A. SET SWITCH REGISTER TO 000200
- B. DEPRESS 'LOAD ADDRESS' KEY AND RELEASE
- C. SET SWR TO ZERO FOR DEFAULT PARAMETERS ESTABLISHED IN THE TAPE (SEE SECTION 8.5.3 FOR FULL EXPLANATION OF DEFAULT PARAMETERS), OR LEAVE SWR BIT 7=1 TO USE EXISTING PARAMETERS PREVIOUSLY SET UP BY THE DUP11 PARAMETER DIALOG OR A PREVIOUSLY RUN DUP11 DIAGNOSTIC. SET SWR=1 TO GO THROUGH THE PARAMETER DIALOG. (IT IS NOT NECESSARY TO INPUT NEW PARAMETERS FOR EACH TAPE.) (SECTION 7.2, 8.4 AND 8.5 MAY BE HELPFUL)
- D. DEPRESS 'START KEY' AND RELEASE. THE PROGRAM WILL TYPE MAINDEC NAME AND PROGRAM NAME (IF THIS WAS THE FIRST START UP OF THE PROGRAM) AND ALSO THE FOLLOWING:

'EXAMPLE'

'MAP OF DUP11 STATUS'

1500	160050	CSR OF FIRST DUP11
1502	000300	VECTOR OF FIRST DUP11
1504	140026	STATUS AND SYNC FOR FIRST DUP11
1506	160060	CSR OF SECOND DUP11
1510	000310	VECTOR OF SECOND DUP11
1512	140026	STATUS AND SYNC FOR SECOND DUP11

THE ABOVE IS ONLY AN EXAMPLE! THIS WOULD INDICATE THE STATUS TABLE STARTING AT ADDRESS 1500 IN THE PROGRAM. THE STATUS TABLE MUST BE VERIFIED BY THE USER. FOR INFORMATION ON THE STATUS TABLE SEE SECTION 8.4 FOR HELP.

IT IS POSSIBLE FOR THE OPERATOR TO MANUALLY CHANGE (TOGGLE IN) THE INFORMATION IN THE MAP TO SUIT A SPECIFIC CONFIGURATION OF REVICES, BUT THE RESPONSIBILITY FOR VERIFYING THAT INFORMATION RESTS WITH THE OPERATOR.

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

4.1 CONTROL SWITCH SETTINGS

SW 15	SET:	HALT ON ERROR
SW 14	SET:	LOOP ON CURRENT TEST
SW 13	SET:	INHIBIT ERROR PRINT OUT
SW 12	SET:	INHIBIT TYPE OUT/BELL ON ERROR.
SW 11	SET:	INHIBIT ITERATIONS. (QUICK PASS)
SW 10	SET:	ESCAPE TO NEXT TEST ON ERROR
SW 09	SET:	LOOP WITH CURRENT DATA
SW 08	SET:	CATCH ERROR AND LOOP ON IT
SW 07	SET:	USE PREVIOUS STATUS TABLE.
SW 06	SET:	RESERVED
SW 05	SET:	RESERVED

SW 04 SET: RESERVED
SW 03 SET: SELECT DUP11'S DESIRED ACTIVE
SW 02 SET: LOCK ON SELECTED TEST
SW 01 SET: RESTART PROGRAM AT SELECTED TEST
SW 00 SET: ENTER PARAMETERS USING MANUAL DIALOG

SWITCHES 8 THROUGH 15 ARE DYNAMIC AND SHOULD BE USED AS NEEDED IN THE DIAGNOSTIC. SWITCHES 0 THROUGH 3 ARE STATIC (ONLY ARE OPERABLE WHEN THE MONITOR PORTION OF THE TAPE IS RUNNING) AND SHOULD BE SET UP PRIOR TO STARTING OR RESTARTING THE DIAGNOSTIC.

4.1.2 SWITCH REGISTER RESTRICTIONS

SW 03 RESELECT DUP11'S DESIRED ACTIVE. PLEASE NOTE THAT A MESSAGE IS TYPED OUT FOR SETTING THE SWITCH REGISTER EQUAL TO DUP11'S ACTIVE. THIS MEANS IF THE SYSTEM HAS THREE DUP11S BITS 00, 01, 02 WILL BE SET IN LOC 'DUPACTV' FROM THE SWITCH REGISTER. USING THIS SWITCH(SW03) ALTERS THAT LOCATION. THEREFORE, IF THREE DUP11S ARE IN THE SYSTEM ***DO NOT*** SET SWITCHES GREATER THAN SW 02 IN THE UP POSITION. THIS WOULD BE A FATAL ERROR. DO NOT SELECT MORE ACTIVE DUP11S THAN HAS BEEN GIVEN INFORMATION ABOUT IN THE PARAMETER PROGRAM.

AS EXPLAINED IN SECTION 1.0, DEVICES SHOULD BE CONSECUTIVELY ADDRESSED, AND CAN BE SELECTED OR DESELECTED USING THIS SWITCH.

METHOD: A. LOAD ADDRESS 200
B. START WITH SW 03=1
C. PROGRAM WILL TYPE MESSAGE
D. SET THE BINARY NUMBER OF DUP11S DESIRED ACTIVE. EXAMPLE: 1=1 DUP11; 3=2 DUP11; 7=3 DUP11; 17=4 DUP11 37=5 DUP11 ETC. PRESS CONTINUE.
E. NUMBER (IF VALID) WILL BE IN DATA LIGHTS (EXCLUDING 11/05)
F. SET WITH ANY OTHER SWITCH SETTINGS DESIRED. PRESS CONTINUE.

SW 01 RESTART PROGRAM AT SELECTED TEST. IT IS STRONGLY SUGGESTED THAT AT LEAST ONE PASS HAS BEEN MADE BEFORE TRYING TO SELECT A TEST THAT IS NOT IN THE ORDER OF SEQUENCE. THE REASON FOR THIS IS THAT THE PROGRAM HAS TO CLEAR AREAS AND SET UP PARAMETERS IN THE MONITOR PORTION OF THE PROGRAM. IT IS POSSIBLE TO LD200, AND RAISE SW01, THEN START, PROVIDED PARAMETERS HAVE BEEN PREVIOUSLY SET UP AS DESCRIBED IN SECTION 4.0. ALSO, WHEN A TEST IS SELECTED, ALWAYS START AT THE VERY BEGINNING OF THAT TEST.

SW 09 LOOP ON CURRENT DATA. THIS SWITCH WILL ONLY WORK IF CALL 'SCOPI' IS IN THAT TEST. THE REASON IS THAT MOST

TESTS DEAL WITH BLOCKS OF DIFFERENT DATA TO BE SENT OR RECEIVED ALL AT ONCE, THUS KNOWN AS BLOCK DATA--ONE PATTERN CAN'T BE SINGLED OUT. (SEE SECTION 4.1.3.B.1)

4.1.3 SWITCH REGISTER PRIORITIES

A) ERROR SWITCHES

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

B) SCOPE SWITCHES

1. SW 09 - (IF ENABLED BY 'SCOPI') ON AN ERROR. IF AN ASTERISK '*' IS PRINTED IN FRONT OF THE TEST NUMBER (EX. *TEST NO. 10), SW09 IS INCORPORATED IN THAT TEST AND THEREFORE SW09 IS USUALLY THE BEST SWITCH FOR THE SCOPE LOOP (SW14=0, SW10=0, SW09=1, SW08=0).

IF SW09 IS NOT ENABELED AND THERE IS A *HAPD* ERROR (CONSTANT ERROR) SW08 IS BEST. (SW14=0, SW10=0, SW09=0, SW08=1).

FOR INTERMITTENT ERRORS, SW14-1 WILL LOOP ON TEST REGARDLESS OF ERROR OR NO ERROR. (SW14=1, SW10=0, SW09=0, SW08=1,0)

2. SW 14 - LOOP ON TEST. WILL LOOP ON TEST UNTIL SWITCH IS LOWERED.
3. SW 11 - INHIBIT ITERATIONS (QUICK PASS). ALLOWS ONLY ONE PASS THROUGH A TEST.

4.2 STARTING ADDRESS

STARTING ADDRESS IS AT 000200. THERE ARE NO OTHER STARTING ADDRESSES FOR THE DUP11 DIAGNOSTICS.

NOTE: IF ADDRESS 000042 IS NON-ZERO THE PROGRAM ASSUMES IT IS UNDER ACT11 OR XXDP CONTROL AND WILL ACT ACCORDINGLY. AFTER *ALL* AVAILABLE DUP11'S ARE TESTED THE PROGRAM WILL RETURN TO 'XXDP' OR 'ACT-11'.

5.0 OPERATING PROCEDURE

WHEN THE PROGRAM IS INITIALLY STARTED MESSAGES AS DESCRIBED IN SECTION FOUR WILL BE PRINTED AND PROGRAM WILL BEGIN RUNNING THE DIAGNOSTIC.

5.1 PROGRAM AND/OR OPERATOR ACTION

THE TYPICAL APPROACH SHOULD BE

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

THE TEST NUMBER AND PC WILL BE TYPED OUT AND POSSIBLY AN ERROR MESSAGE (THIS DEPENDS ON THE TEST), TO GIVE THE OPERATOR AN IDEA AS TO THE SOURCE OF THE PROBLEM. IF IT IS NECESSARY TO KNOW MORE INFORMATION CONCERNING THE ERROR REPORT, LOOK IN THE LISTING FOR THAT TEST NUMBER WHICH WAS TYPED OUT AND THEN NOTE THE PC OF THE ERROR REPORT. IN THIS WAY THE EXACT FUNCTIONING OF THE TEST CAN BE INTERPRETED SINCE THE ERROR PC IS THE HLT+2 LOCATION.

IN SOME TESTS, THERE IS A SUBROUTINE CALL THROUGH A REGISTER (E.G., JSR R1,FLAG). THE SUBROUTINE DOES THE DATA CHECKING FOR THE TEST AND WILL REPORT AN ERROR IF ONE OCCURS. THIS MEANS THAT THE FAILING TEST COULD BE IN ONE PART OF THE LISTING WHILE THE SUBROUTINE THAT FOUND THE ERROR IS IN ANOTHER PART. TO DETERMINE THE PC OF THE FAILING TEST, CHECK THE REGISTER USED BY THE SUBROUTINE. IT WILL CONTAIN THE RETURN ADDRESS OF THE FAILING TEST.

6.0 ERRORS

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

6.1 ERROR RECOVERY

IF FOR SOME REASON THE DUP11 SHOULD 'HANG THE BUS' (GAIN CONTROL OF BUS SO THAT CONSOLE MANUAL FUNCTIONS ARE INHIBITED) AN INIT OR POWER DOWN/UP IS NECESSARY FOR OPERATOR TO REGAIN CONTROL OF CPU. IF THIS SHOULD HAPPEN LOOK IN LOCATION 'TSTNO' FOR THE NUMBER OF THE TEST THAT WAS RUNNING AT THE TIME OF THE CATASTROPHIC ERROR. THIS GIVES THE OPERATOR SOME IDEA AS TO WHAT THE DUP11 WAS DOING AT THE TIME OF THE ERROR.

7.0 RESTRICTIONS

7.1 STARTING RESTRICTIONS

SEE SECTION 4 (PLEASE). STATUS TABLE SHOULD BE VERIFIED REGARDLESS OF HOW THE PROGRAM WAS STARTED. ALSO, IT IS IMPORTANT TO USE THE LISTING ALONG WITH THE INFORMATION PRINTED ON THE TTY TO COMPLETELY ISOLATE PROBLEMS.

7.2 OPERATING RESTRICTIONS

DUP11 "PARAMETER DIALOG" MUST BE RUN ONLY ONCE PRIOR TO THE FIRST RUNNING OF ANY DUP11 DIAGNOSTIC IF "DEFAULT PARAMETERS" ARE NOT USED. IF ONLY DUP11 DIAGNOSTICS WERE LOADED AFTER DUP11 PARAMETER SETUP, AND IF CORE MEMORY HAS NOT BEEN CHANGED, I.E., USE OF DIAGNOSTICS OTHER THAN DUP11 DIAGNOSTICS, AND IF THERE WERE NO DUP11 CONFIGURATION CHANGES, THE DUP11 PARAMETER SETUP NEED NEVER BE RUN AGAIN. HOWEVER, IF ANY OF THE ABOVE HAVE BEEN VIOLATED THE DUP11 PARAMETER SETUP MUST BE RUN AGAIN BEFORE RUNNING THE DIAGNOSTICS. UNDER NORMAL OPERATING CONDITIONS IT SHOULD NOT BE NECESSARY TO INPUT NEW PARAMETERS TO SUBSEQUENT DIAGNOSTICS, UNLESS A CHANGE IS REQUIRED.

NOTE: AN ALTERNATIVE TO THE ABOVE IS ATTEMPTING THE DEFAULT PARAMETERS WHEN THE PROGRAM IS INITIALLY STARTED WITH SWR=0.

7.3 HARDWARE CONFIGURATION RESTRICTIONS FOR THE PURPOSE OF RUNNING MULTIPLE DUP11'S IN CHAIN MODE.

1. CSR ADDRESSES MUST BE CONSECUTIVE.
2. VECTORS ARE CONSECUTIVE IF PARAMETER PROGRAM IS USED.
3. ALL JUMPERS ARE ASSUMED TO BE AS SETUP IN PARAMETER DIALOG.
4. PRIORITY LEVEL MUST BE THE SAME FOR ALL DEVICES.

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

ALL DUP11 DEVICE DIAGNOSTICS WILL GIVE AN 'END PASS' MESSAGE (PROVIDING NO ERRORS AND SW12-0) WITHIN 4 MINS. THIS IS ASSUMING SW11=1 (DELETE ITERATIONS) IS SET TO GIVE THE FASTEST POSSIBLE EXECUTION. THE ACTUAL EXECUTION TIME DEPENDS GREATLY ON THE PDP11 CPU CONFIGURATION.

8.2 PASS COMPLETE

NOTE: *EVERY* TIME THE PROGRAM IS STARTED, THE TESTS WILL RUN AS IF SW11 (DELETE ITERATIONS) WAS UP (=1). THIS IS TO VERIFY NO *HARD* ERRORS AS SOON AS POSSIBLE. THEREFORE THE FIRST PASS--EACH TIME PROGRAM IS STARTED--WILL BE A 'QUICK PASS' UNTIL ALL DUP11'S IN SYSTEM ARE TESTED. WHEN THE DIAGNOSTIC HAS COMPLETED A PASS WITH THE NORMAL ITERATION COUNT (ICOUNT 50), THE FOLLOWING IS AN EXAMPLE OF THE PRINT OUT TO BE EXPECTED.

END PASS CZDPCC CSR:160050 VEC:300 PASSES:000001 ERRORS:000000

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

8.3 KEY LOCATIONS

RETURN CONTAINS THE ADDRESS WHERE PROGRAM WILL RETURN WHEN ITERATION COUNT IS REACHED OR !F LOOP ON TEST IS ASSERTED.

NEXT CONTAINS THE ADDRESS OF THE NEXT TEST TO BE PERFORMED.

TSTN() CONTAINS THE NUMBER OF THE TEST NOW BEING PERFORMED.

RUN THE BIT IN 'RUN' ALWAYS POINTS ONE PAST THE DUP11 CURRENTLY BEING TESTED. EXAMPLE: (RUN) /0000000001000000 MEANS THAT DUP11 NO.05 IS THE DUP11 NOW RUNNING.

DUPCR00-DUPCR07 (1500)-(1560) THESE LOCATIONS CONTAIN THE INFORMATION NEEDED TO TEST UP TO 8 (DECIMAL) DUP11S SEQUENTIALY. THEY CONTAIN THE CSR, VECTOR AND STATUS CONCERNING THE CONFIGURATION OF EACH DUP11.

DUPACTV EACH BIT SET IN THIS LOCATION INDICATES THAT THE ASSOCIATED DUP11 WILL BE TESTED IN TURN. EXAMPLE: (DUPACTV) /0000000000011111 MEANS THAT DUP11 NO. 00,01,02,03,04 WILL BE TESTED. EXAMPLE: (DUPACTV) /0000000000010001 MEANS

THAT DUP11 NO. 00,04 WILL BE TESTED.

RXC(SR) CONTAINS THE RECEIVER CSR OF THE CURRENT DUP11 UNDER TEST.

8.4 MORE ON THAT 'STATUS TABLE' (1500-1560)

'MAP OF DUP11 STATUS'

1500	160050
1502	000300
1504	140000

THE ABOVE INFORMATION WILL BE REPEATED FOR EACH OF UP TO 8 DUP11'S IN THE SYSTEM (THESE WILL FOLLOW UNDER THIS TABLE).
EXPLANATION:

1500	160050	THIS IS THE SYSTEM CONTROL REGISTER FOR THE 1ST DUP11 IN THE SYSTEM.
1502	000300	THIS IS VECTOR 'A' FOR THE FIRST DUP11 IN THE SYSTEM.
1504	140026	THIS REPRESENTS SYNC AND SOFTWARE STATUS FOR THE FIRST DUP.

THE BITS ARE AS FOLLOWS:

BIT 15	SET:	OPTIONAL CLEAR JUMPER IN
BIT 14	SET:	TURNAROUND CONNECTOR ON
BIT 13	SET:	
BIT 12	SET:	
BIT 11	SET:	
BIT 10	SET:	
BIT 09	SET:	
BIT 08	SET:	
BIT 07-00		SYNC CHARACTER FOR DECMODE TESTS.

THE ABOVE IS REPEATED FOR EACH DUP11 IN THE SYSTEM. THE TABLE IS FILLED BY DEFAULT PARAMETERS OR BY THE MANUAL PARAMETER INPUT AS DESCRIBED PREVIOUSLY. ALSO, IF DESIRED BY THE USER - THE LOCATIONS MAY BE ALTERED BY HAND (TOGGLED IN) TO SUIT THE SPECIFIC CONFIGURATION, THUS MAKING EACH DEVICE MAP DIFFERENT. IT IS THE RESPONSIBILITY OF THE OPERATOR TO VERIFY THE DATA IN THE MAP.

8.5 METHOD OF DEVELOPING DEFAULT PARAMETERS

8.5.1 DEFAULT PARAMETER ASSUMPTIONS

TOO MUCH HARDWARE WOULD HAVE TO BE ANALYZED TO SIZE THE THE PARAMETERS. THE PROGRAM MUST ASSUME THE VARIATIONS. THE RESULT, IF NOT TO YOUR SPECIFIC CONFIGURATION, MAY BE ALTERED BY HAND (TOGGLE IN) AS DESIRED. IN THIS WAY 95% OF THE

PARAMETER SETUP WAS DONE BY THE PROGRAM AND 5% BY YOU.
THEREFORE:

1) ALL JUMPERS ARE ASSUMED TO BE IN THE FOLLOWING
CONFIGURATION.

	IN	OUT
W1 SECONDARY REC ENABLE	X	
W2 SEC REC DISABLE		X
W3 CLEAR OPTION	X	
W4 SEC TX ENABLE	X	
W5 DSC A CONTROL		X
W6 A+B DS CONTROL	X	
W7 BUS GRANT CONTROL	X	

2) THE H325 TURN AROUND CONNECTOR IS ASSUMED TO BE ON.

3) THE MANUFACTURING OPTION CSR OF 160050 AND VECTOR OF 770
ARE USED.

4) THE BR LEVEL IS ASSUMED TO BE 5.

IN ALL ADJUSTMENTS PLEASE REFER TO SECTION 8.4 FOR GREATER
DETAIL.

INTRODUCTION TO DUP11 DIAGNOSTIC

: *CZDPCCO /<377>/DUP11 OFLNE SDLC RCVR
: *COPYRIGHT(C) 1975,1978, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
: -----

: STARTING PROCEDURE
: LOAD PROGRAM
: LGAD ADDRESS 000200
: PRESS START
: PROGRAM WILL TYPE "CZDPCCO /<377>/DUP11 OFLNE SDLC RCVR "
: PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
: AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
: AND THEN RESUME TESTING

: SWITCH REGISTER OPTIONS
: -----

550	100000	SW15=100000	: -1, HALT ON ERROR
551	040000	SW14=40000	: -1, LOOP ON CURRENT TEST
552	020000	SW13=20000	: -1, INHIBIT ERROR TYPEOUT
553	010000	SW12=10000	: =1, DELETE TYPEOUT/BELL ON ERROR.
554	004000	SW11=4000	: =1, INHIBIT ITERATIONS
555	002000	SW10=2000	: =1, ESCAPE TO NEXT TEST ON ERROR
556	001000	SW09=1000	: =1, LOOP WITH CURRENT DATA
557	000400	SW08=400	: =1, LOOP ON ERROR
558	000200	SW07=200	
559	000100	SW06=100	
560	000040	SW05=40	
561	000020	SW04=20	
562	000010	SW03=10	: SELECT DUP'S DESIRED ACTIVE
563			: NOTE: THIS MUST NOT EXCEED ORIGINAL COUNT
564	000004	SW02=4	: LOCK ON TEST SELECT
565	000002	SW01=2	: RESTART PROGRAM AT SELECTED TEST
566	000001	SW00=1	: ENTER PARAMETERS

GENERAL DEFINITIONS AND EQUIVALENCIES

567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616

000000
000001
000002
000003
000004
000005
000006
000007

:REGISTER DEFINITIONS
:-----

R0=20 :GENERAL REGISTER
R1=21 :GENERAL REGISTER
R2=22 :GENERAL REGISTER
R3=23 :GENERAL REGISTER
R4=24 :GENERAL REGISTER
R5=25 :GENERAL REGISTER
SP=26 :PROCESSOR STACK POINTER
PC=27 :PROGRAM COUNTER

:LOCATION EQUIVALENCIES
:-----

177776
001150

PS=177776 :PROCESSOR STATUS WORD
STACK=1150 :START OF PROCESSOR STACK

:INSTRUCTION DEFINITIONS
:-----

005746
005726
010046
012600
024646
022626

PUSH1SP=5746 :DECREMENT PROCESSOR STACK 1 WORD
POP1SP=5726 :INCREMENT PROCESSOR STACK 1 WORD
PUSHR0=10046 :SAVE R0 ON STACK
POPRO=12600 :RESTORE R0 FROM STACK
PUSH2SP=24646 :DECREMENT STACK TWICE
POP2SP=22626 :INCREMENT STACK TWICE
.EQUIV EMT,HLT :BASIC DEFINITION OF ERROR CALL

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

BIT15=100000
BIT14=40000
BIT13=20000
BIT12=10000
BIT11=4000
BIT10=2000
BIT9=1000
BIT8=400
BIT7=200
BIT6=100
BIT5=40
BIT4=20
BIT3=10
BIT2=4
BIT1=2
BIT0=1

TRAPCATCHER FOR UNEXPECTED INTERRUPTS

```
617 :*****
618 :-----
619 :TRAPCATCHER FOR ILLEGAL INTERRUPTS
620 :THE STANDARD "TRAP CATCHER" IS PLACED
621 :BETWEEN ADDRESS 0 TO ADDRESS 776.
622 :IT LOOKS LIKE "PC+2 HALT".
623 :-----
624 :*****
625
626 000000 . = 0
627 :STANDARD INTERRUPT VECTORS
628 :-----
629
630 . = 24
631 000024 005050 .PFAIL ;POWER FAIL HANDLER
632 000026 000340 340 ;SERVICE AT LEVEL 7
633 000030 004350 .HLT ;ERROR HANDLER
634 000032 000340 340 ;SERVICE AT LEVEL 7
635 000034 004316 .TRPSRV ;GENERAL HANDLER DISPATCH SERVICE
636 000036 000340 340 ;SERVICE AT LEVEL 7
637
638 000040 000000 . - 40
639 000042 000000 0 ;SAVE FOR ACT-11 OR DDP2
640 000044 000000 0 ;RETURN ADDRESS IF UNDER ACT-11 OR DDP2
641 000046 003104 0 ;SAVE FOR ACT-11 OR DDP2
642 000052 000052 $ENDAD ;FOR USE WITH ACT-11 OR DDP2
643 000052 000000 0 ;ACT-11 PROGRAM CHARACTERISTICS
644
645 . = 174
646 000174 000000 DISPREG:0 ;SOFTWARE DISPLAY REGISTER
647 000176 000000 SWREG: 0 ;SOFTWARE SWITCH REGISTER
648 000200 000200 . = 200
649 000200 000137 001562 JMP .START ;GO TO START OF PROGRAM
650
651
652 . = 1000
653 001000 005377 055103 050104 MTITLE: .ASCIZ <377><12>/CZDPCC /<377>/DUP11 OFLNE SDLC RCVR /<377>
654 001200 001200 . = 1200
655 :SWR AND LIGHTS
656 :-----
657
658 001200 177570 DISPLAY: 177570 ;11/45 CONSOLE LIGHTS
659 001202 177570 SWR: 177570 ;INDIRECT POINTER TO SWITCH REGISTER
660
661 :INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
662 :-----
663
664 001204 177560 TKCSR: 177560 ;TELETYPE KEYBOARD CONTROL REGISTER
665 001206 177562 TKDBR: 177562 ;TELETYPE KEYBOARD DATA BUFFER
666 001210 177564 TPCSR: 177564 ;TELEPRINTER CONTROL REGISTER
667 001212 177566 TPDBR: 177566 ;TELEPRINTER DATA BUFFER
668
669 :PROGRAM CONTROL PARAMETERS
670 :-----
671
```

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

672	001214	000000	RETURN: 0	:SCOPE ADDRESS FOR LOOP ON TEST
673	001216	000000	NEXT: 0	:ADDRESS OF NEXT TEST TO BE EXECUTED
674	001220	000000	LOCK: 0	:ADDRESS FOR LOCK ON CURRENT DATA
675	001222	000001	ICOUNT: 1	:NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
676	001224	000000	LPCNT: 0	:NUMBER OF ITERATIONS COMPLETED
677	001226	000000	TSTNO: 0	:NUMBER OF TEST IN PROGRESS
678	001230	000000	PASCNT: 0	:NUMBER OF PASSES COMPLETED
679	001232	000000	ERRCNT: 0	:TOTAL NUMBER OF ERRORS
680	001234	000000	LSTERR: 0	:PC OF LAST ERROR CALL
681				
682			:PROGRAM VARIABLES	
683			:-----	
684				
685	001236	000000	TEMP1: 0	:TEMPORARY STORAGE
686	001240	000000	TEMP2: 0	:TEMPORARY STORAGE
687	001242	000000	TEMP3: 0	:TEMPORARY STORAGE
688	001244	000000	TEMP4: 0	:TEMPORARY STORAGE
689	001246	000000	TEMP5: 0	:TEMPORARY STORAGE
690	001250	000000	SAVR0: 0	:R0 STORAGE
691	001252	000000	SAVR1: 0	:R1 STORAGE
692	001254	000000	SAVR2: 0	:R2 STORAGE
693	001256	000000	SAVR3: 0	:R3 STORAGE
694	001260	000000	SAVR4: 0	:R4 STORAGE
695	001262	000000	SAVR5: 0	:R5 STORAGE
696	001264	000000	SAVSP: 0	:STACK POINTER STORAGE
697	001266	000000	SAVPC: 0	:PROGRAM COUNTER STORAGE
698				
699	001270	000000	SAVR0A: 0	:R0 STORAGE
700	001272	000000	SAVR1A: 0	:R1 STORAGE
701	001274	000000	SAVR2A: 0	:R2 STORAGE
702	001276	000000	SAVR3A: 0	:R3 STORAGE
703	001300	000000	SAVR4A: 0	:R4 STORAGE
704	001302	000000	SAVR5A: 0	:R5 STORAGE
705	001304	000000	SAVSPA: 0	:STACK POINTER STORAGE
706	001306	000000	SAVPCA: 0	:PROGRAM COUNTER STORAGE
707				
708	001310	000001	DUPACTV: .BLKB 1	:DUP11'S SELECTED ACTIVE.
709	001311	000001	DUPNUM: .BLKB 1	:OCTAL NUMBER OF DUP11'S.
710	001312	000001	SAVACT: .BLKB 1	:ORIGINAL ACTV. DEVICES.
711	001313	000001	SAVNUM: .BLKB 1	:WORKABLE NUMBER.
712	001314	000001	RUN: .BLKB 1	:POINTER ONE PAST RUNNING DEVICE.
713		001316	.EVEN	
714	001316	001500	CREAM: DUP.MAP	:TABLE POINTER.

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```
715
716                                     :CONTROL REGISTER DEFINITIONS
717                                     :-----
718                                     :RXCSR BIT DEFINITIONS
719      100000      DSCA=BIT15      ;DATA SET CHANGE A
720      040000      RING=BIT14      ;RING
721      020000      CTS=BIT13      ;CLR TO SEND
722      010000      CARDET=BIT12    ;CARRIER DETECT
723      004000      REACT=BIT11    ;REC ACTIVE
724      002000      SRD=BIT10      ;SEC REC DATA
725      001000      DSR=BIT9       ;DATA SET RDY
726      000400      STPSYN=BIT8    ;STRIP SYNC
727      000200      RXDONE=BIT7    ;REC DONE
728      000100      RINTEN=BIT6    ;REC INTR ENABLE
729      000040      DSINTE=BIT5    ;DSC INTR ENABLE
730      000020      RCVEN=BIT4     ;REC ENABLE
731      000010      STD=BIT3       ;SEC XMIT DATA
732      000004      RTS=BIT2       ;REQ TO SEND
733      000002      DTR=BIT1       ;DATA TERM RDY
734      000001      DSCB=BIT0     ;DATA SET CHANGE B
735                                     :RXDBUF BIT DEFINITIONS
736      100000      RXDERR=BIT15    ;REC DATA ERROR
737      040000      OVERRUN=BIT14  ;OVERRUN ERROR
738      010000      CRCERR=BIT12   ;CRC ERROR
739      002000      RABORT=BIT10   ;REC ABORT
740      001000      REOM=BIT9      ;REC END OF MESSAGE
741      000400      RSOM=BIT8      ;REC START OF MESSAGE
742                                     :PARCSR BIT DEFINITIONS
743      100000      DECMOD=BIT15    ;DEC MODE (DDCMP)
744      001000      CRCEN=BIT9     ;CRC ENABLE
745      010000      PRISEC=BIT12   ;PRI/SEC SELECT
746                                     :TXCSR BIT DEFINITIONS
747      100000      TXDLAT=BIT15    ;TX DATA LATE
748      040000      MTDATA=BIT14    ;MAINT DATA OUT
749      020000      CLK=BIT13      ;CLK
750      010000      MMCDEB=BIT12   ;MAINT MODE B
751      004000      MMODEA=BIT11   ;MAINT MCDE A
752      002000      BITW=BIT10     ;BIT WINDOW INPUT
753      001000      TXACT=BIT9     ;TX ACTIVE
754      000400      MRESET=BIT8    ;MASTER RESET
755      000200      TXDONE=BIT7    ;XMIT DONE
756      000100      TXINTE=BIT6    ;XMIT DONE INTR ENABLE
757      000020      SEND=BIT4     ;SEND
758      000010      HDXEN=BIT3     ;HDX/FDX
759                                     :TXCSR WRD DEFINITIONS
760      000000      USER=0         ;USER MODE
761      014000      MMODE=14000    ;MAINT INT MODE
762      010000      MEXT=10000     ;MAINT EXT MODE
763      004000      SYSTST=4000    ;SYSTEM TEST MODE
764
765                                     :TXDBUF BIT DEFINITIONS
766                                     :-----
767      100000      RCRC7T=BIT15    ;
768      040000      RCRCIN=BIT14    ;
769      020000      TCRC7T=BIT13    ;
770      010000      TCRCIN=BIT12    ;
```

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

771	004000	TIMER=BIT11	;MAINTENANCE TIMER
772	002000	TABORT=BIT10	;TRANSMIT ABORT
773	001000	TEOM BIT9	;TRANSMIT END OF MESSAGE
774	C'0400	TSOM=BIT8	;TRANSMIT START OF MESSAGE

775
776 ;MISC. PROGRAM DEFINITIONS

777 -----
778 001320 000000 PRIRTY: .WORD 0
779 001322 000001 TCNFLG: .BLKB 1
780 001323 000001 OPCLRJ: .BLKB 1
781 001324 000000 DATA: .WORD 0
782 001326 000000 SHIFTS: .WORD 0
783 001330 000000 MIND: .WORD 0
784 001332 000000 FLAG: .WORD 0
785 001334 000001 STJMFL: .BLKW 1
786 001336 000001 SRJMFL: .BLKW 1
787
788

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```
789
790                ;PROGRAM CONTROL FLAGS
791                ;-----
792
793 001340          000          INIFLG: .BYTE 0          ;PROGRAM INITIALIZATION FLAG
794 001341          000          ERRFLG: .BYTF 0          ;ERROR OCCURED FLAG
795 001342          000          LOKFLG: .BYTE 0          ;LOCK ON CURRENT TEST FLAG
796 001343          000          QV.FLG: .BYTE 0          ;QUICK VERIFY FLAG.
797                                     ;ON FIRST PASS OF EACH DUP11 ITERATIONS
798                                     ;WILL BE SUPPRESSED
799
800                .EVEN
801                $Y=0
802
803                ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
804                ;POINTERS TO SUBROUTINES CAN BE FOUND
805                ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS
806                ;*****
807                ;-----
808 001344          .TRPTAB:
809                SCOPE=TRAP+0          ;CALL TO SCOPE LOOP AND ITERATION HANDLER
810                .SCOPE
811                SCOPI=TRAP+1          ;CALL TO LOOP ON CURRENT DATA HANDLER
812                .SCOPI
813                TYPE=TRAP+2          ;CALL TO TELETYPE OUTPUT ROUTINE
814                .TYPE
815                INSTR=TRAP+3          ;CALL TO ASCII STRING INPUT ROUTINE
816                .INSTR
817                INSTER=TRAP+4          ;CALL TO INPUT ERROR HANDLER
818                .INSTER
819                PARAM=TRAP+5          ;CALL TO NUMERICAL DATA INPUT ROUTINE
820                .PARAM
821                SAVO5=TRAP+6          ;CALL TO REGISTER SAVE ROUTINE
822                .SAVO5
823                RESO5=TRAP+7          ;CALL TO REGISTER RESTORE ROUTINE
824                .RESO5
825                CONVRT=TRAP+10          ;CALL TO DATA OUTPUT ROUTINE
826                .CONVRT
827                CNVRT=TRAP+11          ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.
828                .CNVRT
829                PKCLK=TRAP+12          ;CALL TO CLOCK ROUTINE
830                .PKCLK
831                SETFLG=TRAP+13          ;CALL TO TELETYPE INPUT ROUTINE
832                .SETFLG
833
834                ;-----
835                ;*****
```

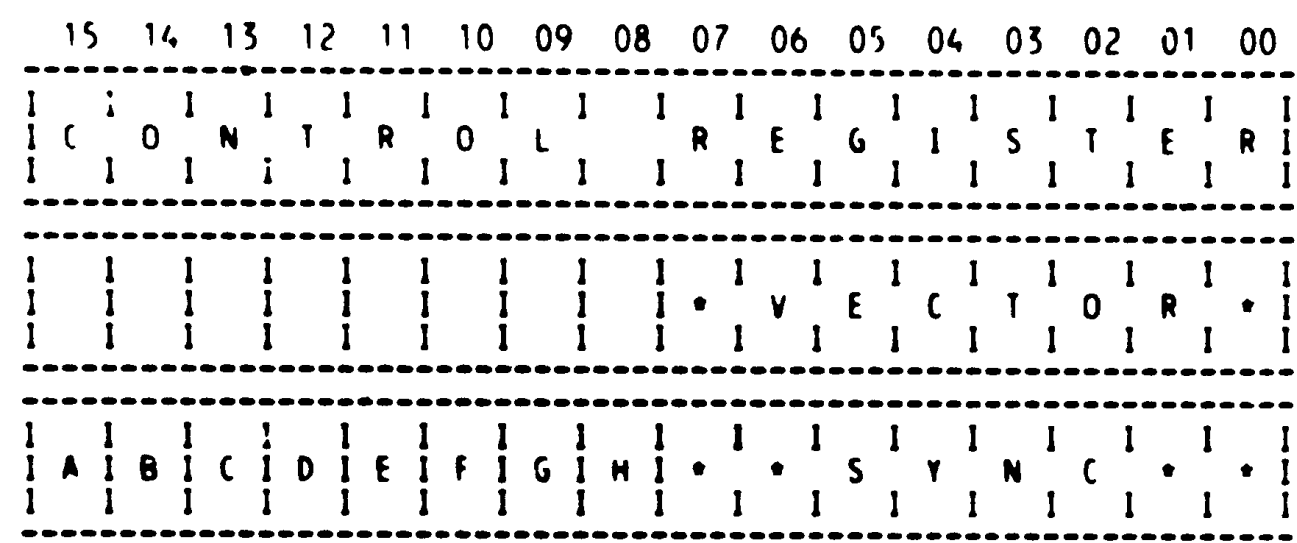
PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```
836                                     ;DUP11 VECTOR AND REGISTER INDIRECT POINTERS
837
838 001374 000000   DUPRVC: 0           ; POINTER TO DUP11 RECEIVER INTERRUPT VECTOR
839 001376 000000   DUPRPS: 0           ; POINTER TO DUP11 RECEIVER INTERRUPT SERVICE PS
840 001400 000000   DUPTVC: 0           ; POINTER TO DUP11 TRANSMITTER INTERRUPT VECTOR
841 001402 000000   DUPTPS: 0          ; POINTER TO DUP11 TRANSMITTER INTERRUPT SERVICE PS
842 001404 000000   RXCSR: 0           ; POINTER TO DUP11 RECEIVER STATUS REGISTER
843 001406 000000   RXDBUF: 0          ; POINTER TO DUP11 RECEIVER DATA BUFFER
844 001410 000000   PARCSR: 0          ; POINTER TO DUP11 PARAMETER STATUS REGISTER
845 001412 000000   TXCSR: 0           ; POINTER TO DUP11 TRANSMITTER STATUS REGISTER
846 001414 000000   TXDBUF: 0          ; POINTER TO DUP11 TRANSMITTER DATA BUFFER
847 001416 000000   DUPSEC: 0          ; POINTER TO DUP11 SECONDARY REGISTER SELECT REGISTER
848 001420 000000   HUPFSR: 0          ; POINTER TO PARAMETER STATUS HIGH BYTE
849 001422 000000   HUPRBF: 0          ; POINTER TO RECEIVER BUFFER HIGH BYTE
850 001424 000000   HUPRCR: 0          ; POINTER TO RECEIVER CONTROL REG HIGH BYTE
851 001426 000000   HUPTBF: 0          ; POINTER TO TRANSMITTER BUFFER HIGH BYTE
852 001430 000000   HUPICR: 0          ; POINTER TO TRANSMITTER CONTROL REG HIGH BYTE
853
854
855                                     ;DUP11 CONTROL INDICATORS FOR CURRENT DUP11 UNDER TEST
856                                     ;-----
857
858 001432 000      MASK.A: .BYTE 000      ; LAST CHAR TO TEST AND PARITY MASK
859
860 001433 010      CLK.A: .BYTE 8.         ; NUMBER OF CLOCKS NEEDED FOR ONE CHAR
861
862 001434 000000   LOO.OO: 000000          ; PARAMETERS
863
```

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```
864                                     ;DUPI1 STATUS TABLE AND ADDRESS ASSIGNMENTS
865                                     ;-----
866
867      001500      001500      . =1500
868      001500      000001      DUP.MAP:
869      001500      000001      DUPCR0: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 0
870      001502      0000C1      DUPTR0: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 0
871      001504      000001      DUPO.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 0
872
873      001506      000001      DUPCR1: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 1
874      001510      000001      DUPTR1: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 1
875      001512      000001      DUP1.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 1
876
877      001514      00J001      DUPCR2: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 2
878      001516      000001      DUPTR2: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 2
879      001520      000001      DUP2.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 2
880
881      001522      000001      DUPCR3: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 3
882      001524      000001      DUPTR3: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 3
883      001526      000001      DUP3.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 3
884
885      001530      000001      DUPCR4: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 4
886      001532      000001      DUPTR4: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 4
887      001534      000001      DUP4.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 4
888
889      001536      000001      DUPCR5: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 5
890      001540      000001      DUPTR5: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 5
891      001542      000001      DUP5.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 5
892
893      001544      000001      DUPCR6: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 6
894      001546      000001      DUPTR6: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 6
895      001550      000001      DUP6.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 6
896
897      001552      000001      DUPCR7: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 7
898      001554      000001      DUPTR7: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 7
899      001556      000001      DUP7.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 7
900
901      001560      000000      DUP.END:      000000
902
903
904
905
906
```

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.



DEFINITIONS

- A- OPTIONAL CLEAR JUMPER IN-1
- B- TURNAROUND CONNECTOR ON 1
- C-
- D-

PROGRAM INITIALIZATION AND START UP.

```

926
927      ;PROGRAM INITIALIZATION
928      ;LOCK OUT INTERRUPTS
929      ;SET UP PROCESSOR STACK
930      ;SET UP POWER FAIL VECTOR
931      ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
932      ;TYPE TITLE MESSAGE
933
934 001562 012737 000340 177776 .START: MOV #340,PS ;LOCK OUT INTERRUPTS
935 001570 012706 001150 MOV #STACK,SP ;SET UP STACK
936 001574 012737 005050 000024 MOV #.PFAIL,@#24 ;SET UP POWER FAIL VECTOR
937 001602 113737 001311 001313 MOV#B DUPNUM,SAVNUM ;SAVE NUMBER OF DEVICES IN SYSTEM
938 001610 005037 001230 CLR PASCNT ;CLEAR PASS COUNT
939 001614 105037 001341 CLRB ERRFLG ;CLEAR ERROR FLAG
940 001620 105037 001343 CLRB QV.FLG ;ZERO QUICK VERIFY FLAG
941 001624 012737 001500 001316 MOV #DUP.MAP,CREAM ;GET MAP POINTER.
942 001632 112737 000001 001314 MOV#B #1,RUN ;POINT POINTER TO FIRST DEVICE.
943 001640 005037 001232 CLR ERRCNT ;CLEAR ERROR COUNT
944 001644 005037 001234 CLR LSTERR ;CLEAR LAST ERROR POINTER
945 001650 012737 000001 001226 MOV #1,TSTNO ;SET UP FOR TEST 1
946 001656 012737 001562 001214 MCV #.START,RETURN ;SET UP FOR POWER FAIL BEFORE
947 ;TESTING STARTS
948 001664 013746 000006 MOV @#6,-(SP) ;SAVE CURRENT VECTORS
949 001670 013746 000004 MOV @#4,-(SP) ;
950 001674 012737 001710 000004 MOV #12$,@#4 ;SETUP FOR TIMEOUT
951 001702 005777 177274 TST @SWR ;REFERENCE HARDWARE SWITCH REG
952 001706 000407 BR 13$ ;BR IF IT EXISTS
953 001710 012737 000176 001202 12$: MOV #SWREG,SWR ;POINT TO SOFT SWR
954 001716 012737 000174 001200 MOV #DISPREG,DISPLAY ;POINT TO SOFT DISPLAY REG
955 001724 022626 CMP (SP)+,(SP)+ ;ADJUST STACK
956 001726 012637 000004 13$: MOV (SP)+,@#4 ;RESTORE VECTORS
957 001732 012637 000006 MOV (SP)+,@#6 ;
958 001736 105737 001340 TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED
959 001742 001401 BEQ 11$
960 001744 000410 BR 6$
961 001746 022737 003104 000042 11$: CMP #SENDAD,@#42 ;IF ACT-11 AUTO MODE,
962 001754 001404 BEQ 6$ ;DON'T TYPE ID
963 001756 104402 001000 TYPE ,MTITLE ;TYPE TITLE MESSAGE
964 001762 105137 001340 COMB INIFLG ;IF NOT SET FLAG AND DO
965 001766 105777 177210 6$: TSTB @SWR ;BIT7-1??
966 001772 100002 BPL 10$
967 001774 000137 002520 JMP 1$
968 002000
969 002000 032777 000001 177174 10$: BIT #SW00,@SWR ;ENTER PARAMETERS
970 002006 001002 BNE .+6 ;YES
971 002010 000137 002360 JMP 21$ ;NO
972 002014 105137 001332 COMB FLAG
973 002020 112737 000001 001340 MOV#B #1,INIFLG ;SET TO MANUAL ENTRY
974 002026 012700 001500 MCV #DUP.MAP,RO ;CLR MAP
975 002032 005020 68$: CLR (RO)+
976 002034 020027 001560 CMP RO,#DUP.END ;DONE WITH MAP?
977 002040 001374 BNE 68$ ;BR IF NO
978 002042 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
979 002044 005473 MCSR ;MESSAGE
980 002046 104405 PARAM ;CONVERT STRING
981 002050 160000 160000 ;LOW LIMIT
  
```

PROGRAM INITIALIZATION AND START UP.

982	002052	175500			175500	:HIGH LIMIT
983	002054	001500			DUPCR0	:STORE AT THIS LOCATION
984	002056	001			.BYTE 1	:MASK
985	002057	001			.BYTE 1	:HOW MANY TIMES + 2
986	002060	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
987	002062	005512			MVEC	:MESSAGE
988	002064	104405			PARAM	:CONVERT STRING
989	002066	000300			300	:LOW LIMIT
990	002070	000770			770	:HIGH LIMIT
991	002072	001502			DUPTRO	:STORE AT THIS LOCATION
992	002074	001			.BYTE 1	:MASK
993	002075	001			.BYTE 1	:HOW MANY TIMES + 2
994	002076	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
995	002100	005702			MVAR	:MESSAGE
996	002102	104405			PARAM	:CONVERT STRING
997	002104	000004			4	:LOW LIMIT
998	002106	000007			7	:HIGH LIMIT
999	002110	001240			TEMP2	:STORE AT THIS LOCATION
1000	002112	000			.BYTE 0	:MASK
1001	002113	001			.BYTE 1	:HOW MANY TIMES + 2
1002	002114	013737	001240	001320	MCV	TEMP2,PRTY ;SAVE PRICRITY
1003	002122	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1004	002124	005647			MTOTAL	:MESSAGE
1005	002126	104405			PARAM	:CONVERT STRING
1006	002130	000001			1	:LOW LIMIT
1007	002132	000010			8.	:HIGH LIMIT
1008	002134	001236			TEMP1	:STORE AT THIS LOCATION
1009	002136	000			.BYTE 0	:MASK
1010	002137	001			.BYTE 1	:HOW MANY TIMES + 2
1011	002140	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1012	002142	005525			MJMPR	:MESSAGE
1013	002144	104413			SETFLG	:SET FLAG BASED UPON INPUT STRING
1014	002146	001323			OPCLRJ	:THIS FLAG
1015	002150	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1016	002152	005600			MTCN	:MESSAGE
1017	002154	104413			SETFLG	:SET FLAG BASED UPON INPUT STRING
1018	002156	001322			TCNFLG	:THIS FLAG
1019	002160	105737	001322		TSTB	TCNFLG
1020	002164	001410			BEQ	71\$
1021	002166	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1022	002170	005726			MSTJM	:MESSAGE
1023	002172	104413			SETFLG	:SET FLAG BASED UPON INPUT STRING
1024	002174	001334			STJMFL	:THIS FLAG
1025	002176	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1026	002200	005761			MSRJM	:MESSAGE
1027	002202	104413			SETFLG	:SET FLAG BASED UPON INPUT STRING
1028	002204	001336			SRJMFL	:THIS FLAG
1029	002206	105737	001323		TSTB	OPCLRJ
1030	002212	001403			BEQ	69\$
1031	002214	052737	100000	001504	BIS	#BIT15,DUPO.A
1032	002222	105737	001322		TSTB	TCNFLG
1033	002226	001403			BEQ	70\$
1034	002230	052737	040000	001504	BIS	#BIT14,DUPO.A
1035	002236	112737	000001	001312	MOVB	#1,SAVACT
1036	002244	113737	001236	001311	MOVB	TEMP1,DUPNUM
1037	002252	113737	001236	001313	MOVB	TEMP1,SAVNUM

PROGRAM INITIALIZATION AND START UP.

1038	002260	005337	001236	65\$:	DEC	TEMP1		
1039	002264	001404			BEQ	64\$		
1040	002266	000261			SEC			
1041	002270	106137	001312		ROLB	SAVACT		
1042	002274	000771			BR	65\$		
1043	002276	113737	001312	001240	64\$:	MOVB	SAVACT,TEMP2	:# OF TIMES
1044	002304	113737	001312	001310		MOVB	SAVACT,DUPACTV	
1045	002312	000241				CLC		
1046	002314	106037	001240			RORB	TEMP2	
1047	002320	012700	001500			MOV	#DUPCRO,R0	
1048	002324	012701	001506			MOV	#DUPCRI,R1	
1049	002330	000241			67\$:	CLC		
1050	002332	106037	001240			RORB	TEMP2	
1051	002336	103051				BCC	66\$	
1052	002340	012011				MOV	(R0)+,(R1)	
1053	002342	062721	000010			ADD	#10,(R1)+	:CSR
1054	002346	012011				MOV	(R0)+,(R1)	
1055	002350	062721	000010			ADD	#10,(R1)+	:VECTOR
1056	002354	012021				MOV	(R0)+,(R1)+	:PARAMETERS
1057	002356	000764				BR	67\$	
1058	002360	012700	001500		21\$:	MOV	#DUP.MAP,R0	:SETUP TO CLEAR MAP
1059	002364	005020			20\$:	CLR	(R0)+	:CLEAR
1060	002366	020027	001560			CMP	R0,#DUP.END	:CHECK FOR FINISH
1061	002372	001374				BNE	20\$:BR IF MORE TO GO
1062	002374	012700	001500			MOV	#DUP.MAP,R0	:SETUP TO DEFAULT
1063	002400	012710	160050			MOV	#160050,(R0)	:LOAD CSR
1064	002404	012760	000770	000002		MOV	#770,2(R0)	:LOAD VECTOR
1065	002412	012760	140026	000004		MOV	#140026,4(R0)	:LOAD PARAMETERS AND SYNC
1066	002420	112737	000005	001320		MOVB	#5,PRTY	:LOAD PRIORITY
1067	002426	012700	000001			MOV	#1,R0	:SAVE CORE THIS WAY
1068	002432	110037	001310			MOVB	R0,DUPACTV	:PRESET PROGRAM CONTROLS
1069	002436	110037	001311			MOVB	R0,DUPNUM	:DITTO
1070	002442	110037	001312			MOVB	R0,SAVACT	:DITTO
1071	002446	110037	001313			MOVB	R0,SAVNUM	:DITTO
1072	002452	110037	001322			MOVB	R0,TLNFLG	:DITTO
1073	002456	110037	001323			MOVB	R0,OPCLRJ	:DITTO
1074	002462				66\$:			
1075	002462	104402	006014		16\$:	TYPE	,XHEAD	:TYPE HEADER
1076	002466	012737	001500	001236		MOV	#DUP.MAP,TEMP1	:SET POINTER
1077	002474	017737	176536	001240	5\$:	MOV	@TEMP1,TEMP2	:SET DATA
1078	002502	001406				BEQ	1\$:ALL DONE WITH DATA
1079	002504	104410				CONVRT		:
1080	002506	006042				XSTATQ		:
1081	002510	062737	000002	001236		ADD	#2,TEMP1	:UPDATE POINTER
1082	002516	000766				BR	5\$	
1083	002520	032777	000001	176454	1\$:	BIT	#SW00,@SWR	
1084	002526	001405				BEQ	7\$	
1085	002530	005737	001332			TST	FLAG	
1086	002534	001002				BNE	7\$	
1087	002536	000137	002000			JMP	10\$	
1088	002542	005037	001332		7\$:	CLR	FLAG	
1089	002546	005737	000042			TST	@#42	:IS PROGRAM RUNNING UNDER MONITOR
1090	002552	001030				BNE	3\$:BR IF YES
1091	002554	032777	000010	176420		BIT	#SW03,@SWR	:SELECT SPECIFIC DEVICES??
1092	002562	001424				BEQ	3\$:BR IF NO.
1093	002564	104402	005413			TYPE	,MNEW	:TYPE THE MESSAGE.

CZDPCC MACY11 30A(1052) 26-FEB-79 09:33 PAGE 27
CZDPCD.P11 08-FEB-79 13:15 PROGRAM INITIALIZATION AND START UP.

```

1094 002570 005000          CLR      RO          ;ZERO DATA LIGHTS
1095 002572 000000          HALT          ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
1096 002574 127737 176402 001312  CMPB     @SWR,SAVACT ;IS THE NUMBER VALID?
1097 002602 101404          BIOS     2$          ;BR IF NUMBER IS OK.
1098 002604 104402 005254   TYPE     ,MERR3     ;TELL USER OF INVALID NUMBER.
1099 002610 000000          HALT          ;STOP EVERY THING.
1100 002612 000776          BR        -2         ;RESTART THE PROGRAM AGAIN.
1101 002614 117737 176362 001310 2$:  MOVB    @SWR,DUPACTV ;GET NEW DEVICE PATTERN
1102 002622 113700 001310   MOVB    DUFACTV,RO  ;SHOW THE USER WHAT HE SELECTED.
1103 002626 042700 177400   BIC     #'C<377>,RO ;USE ONLY LOW BYTE.
1104 002632 000000          HALT          ;CONTINUE DYNAMIC SWITCHES.
1105 002634 012700 000300 3$:  MOV     #300,RO     ;PREPARE TO CLEAR THE FLOATING
1106 002640 012701 000302   MOV     #302,R1     ;VECTOR AREA. 300-776
1107 002644 010120 4$:  MOV     R1,(R0)+    ;START PUTTING 'PC+2 - HALT'
1108 002646 005021          CLR     (R1)+      ;IN VECTOR AREA.
1109 002650 022021          CMP     (R0)+,(R1)+ ;POP POINTERS
1110 002652 022700 001000   CMP     #1000,RO   ;ALL DONE??
1111 002656 001372          BNE     4$         ;BR IF NO.
1112
1113          ;TEST START AND RESTART
1114          ;-----
1115
1116 002660 012737 000340 177776 .BEGIN: MOV     #340,PS     ;LOCK OUT INTERRUPTS
1117 002666 012706 001150   MOV     #STACK,SP  ;SET UP STACK
1118 002672 005737 000042   TST     @#42       ;S PROGRAM UNDER MONITOR CONTROL
1119 002676 001023          BNE     2$         ;BR IF YES
1120 002700 032777 000004 176274  BIT     #BIT2,@SWR ;CHECK FOR LOCK ON TEST
1121 002706 001411          BEQ     1$         ;BR IF NO LOCK DESIRED.
1122 002710 104402 005312   TYPE     ,MLOCK    ;TYPE LOCK SELECTED.
1123 002714 012737 000240 003174  MOV     #NOP,TTST  ;ADJUST SCOPE ROUTINE.
1124 002722 012737 000240 003176  MOV     #NOP,TTST+2 ;SET UP TO LOCK
1125 002730 000406          BR      2$         ;CONTINUE ALONG.
1126 002732 013737 003306 003174 1$:  MOV     BRW,TTST+  ;PREPARE NORMAL SCOPE ROUTINE
1127 002740 013737 003310 003176   MOV     BRX,TTST+2 ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1128 002746 012737 006224 001214 2$:  MOV     #CYCLE,RETURN ;START AT 'CYCLE' FIND WHICH DEVICE TO TEST
1129 002754 104402 005202   TYPE     ,MR       ;TYPE R
1130 002760 000177 176230   JMP     @RETURN    ;START TESTING

```

```

1131                                     :END OF PASS
1132                                     :TYPE NAME OF TEST
1133                                     :UPDATE PASS COUNT
1134                                     :CHECK FOR EXIT TO ACT-11
1135                                     :RESTART TEST
1136
1137 002764 005037 001234      .EOP: CLR      LSTERR      :CLEAR LAST ERROR PC
1138 002770 105037 001341      CLR      ERRFLG      :CLEAR ERROR FLAG
1139 002774 005237 001230      INC      PASCNT      :UPDATE PASS COUNT
1140 003000 013777 001230      MOV      PASCNT,@DISPLAY :DISPLAY PASS COUNT
1141 003006 104402 005157      TYPE    ,MEPASS      :TYPE END PASS
1142 003012 104402 005341      TYPE    ,MCSR        :TYPE CSR
1143 003016 104411 003130      CNVRT   ,XCSR        :SHOW IT
1144 003022 104402 005347      TYPE    ,MVECX       :TYPE VECTOR
1145 003026 104411 003136      CNVRT   ,XVEC        :SHOW IT
1146 003032 104402 005355      TYPE    ,MPASSX      :TYPE PASSES
1147 003036 104411 003144      CNVRT   ,XPASS       :SHOW IT
1148 003042 104402 005366      TYPE    ,MERRX       :TYPE ERRORS
1149 003046 104411 003152      CNVRT   ,XERR        :SHOW IT
1150 003052 105337 001313      DECB    SAVNUM       :ARE ALL DEVICES TESTED?
1151 003056 001017              BNE     RESTR        :BR IF NO.
1152 003060 112737 000377      MOV     #377,QV.FLG  :SET THE QUICK VERIFY FLAG.
1153 003066 113737 001311      MOV     DUPNUM,SAVNUM :RESTORE THE COUNT
1154 003074 013701 000042      MOV     @#42,R1     :CHECK FOR ACT-11 OR DDP
1155 003100 001406              BEQ     RESTR        :IF NOT, CONTINUE TESTING
1156 003102 000005              RESET              :STOP THE SHOW--CLEAR THE WORLD
1157 003104
1158 003104 004711      $ENDAD: JSR     PC,(R1)
1159 003106 000240      NOP
1160 003110 000240      NOP
1161 003112 000240      NOP
1162 003114 000240      NOP
1163 003116 012737 006224 001214 RESTR: MOV     #CYCLE,RETURN
1164 003124 000137 006224      JMP     CYCLE
1165 003130 000001      XCSR:  1
1166 003132 006      .BYTE  6,2
1167 003134 001404      RXCSR
1168 003136 000001      XVEC:  1
1169 003140 003      .BYTE  3,2
1170 003142 001374      DUPRVC
1171 003144 000001      XPASS: 1
1172 003146 006      .BYTE  6,2
1173 003150 001230      PASCNT
1174 003152 000001      XERR:  1
1175 003154 006      .BYTE  6,2
1176 003156 001232      ERRCNT
1177
1178                                     ;SCOPE LOOP AND INTERATION HANDLER
1179
1180 003160 005037 001234      .SCOPE: CLR     LSTERR      :CLEAR LAST ERROR PC
1181 003164 010016      MOV     RO,(SP)     :SAVE RO ON STACK
1182 003166 032777 040000 176006 TTST: BIT     #BIT14,@SWR :LOOP ON TEST?
1183 003174 001407      BEQ    1$           :BR IF NO (IF LOCK SW01 - 1;THIS LOCATION = 240)
1184 003176 000437      BR     3$           :GO TO 3$ (DITTO)
1185 003200 105777 176000      TSTB   @TKCSR      :KYBD DONE?
1186 003204 100034      BPL    3$           :BR IF NO (LOCK: HIT A KEY ON TTY TO GO TO NEXT TEST)

```

```

1187 003206 017700 175774      MOV      @TKDBR,R0      ;CLR DONE BIT
1188 003212 000415              BR      2$             ;CONTINUE
1189 003214 032777 004600 175760 1$:  BIT      #SW11,@SWR    ;DELETE ITERATION (QUICK PASS)?
1190 003222 001011              BNE     2$             ;BR IF YES
1191 003224 105737 001343      TSTB   QV.FLG         ;HAS FIRST PASS BEEN COMPLETED?
1192 003230 001406              BEQ     2$             ;BR IF QUICK VERIFY
1193 003232 005237 001224      INC     LPCNT          ;UPDATE ITERATION COUNTER
1194 003236 023737 001224 001222  CMP     LPCNT,ICOUNT   ;ALL ITERATIONS DONE?
1195 003244 001014              BNE     3$             ;BR IF NOT YET
1196 003246 105037 001341      CLR    ERRFLG         ;PREPARE FOR NEW TEST
1197 003252 005037 001224      CLR    LPCNT          ;START ICOUNT AT ZERO
1198 003256 005037 001220      CLR    LOCK           ;
1199 003262 012737 000050 001222  MOV     #50,ICOUNT     ;RESET ITERATIONS
1200 003270 013737 001216 001214  MOV     NEXT,RETURN    ;GET NEXT TEST
1201 003276 011600              MOV     (SP),R0        ;POP R0 OFF STACK
1202 003300 022626              POP2SP                ;FAKE AN RTI
1203 003302 000177 175706      JMF    @RETURN        ;GO DO THE TEST
1204 003306 001407              BRW:   1407
1205 003310 000437              BRX:   437
1206
1207
1208
1209
1210 003312 032777 001000 175662  .SCOPI: BIT      #SW09,@SWR    ;IS SW09-1(SET)?
1211 003320 001405              BEQ     1$             ;BR IF NOT SET.
1212 003322 005737 001220      TST    LOCK           ;
1213 003326 001402              BEQ     1$             ;
1214 003330 013716 001220      MOV     LOCK,(SP)     ;GOTO THE ADDRESS IN LOCK.
1215 003334 000002              1$:   RTI              ;GO BACK.
1216
1217
1218
1219
1220 003336 010546              .TYPE: MOV     R5,-(SP)    ;SAVE R5 ON THE STACK.
1221 003340 017605 000002      MOV     @2(SP),R5     ;GET ADDRESS OF MESSAGE.
1222 003344 062766 000002 000002  ADD     #2,2(SP)      ;POP OVER ADDRESS.
1223 003352 032777 010000 175622  1$:   BIT      #SW12,@SWR    ;INHIBIT ALL PRINT OUT??
1224 003360 001012              BNE     3$             ;BR IF NO PRINT OUT WANTED (SW12=1)
1225 003362 105715              TSTB   (R5)           ;IS NUMBER MINUS? (MSB-1(BIT7))
1226 003364 100002              BPL     2$             ;BR IF NUMBER IS PLUS
1227 003366 104402 005136      TYPE   ,MCRLF         ;TYPE A CR/LF!
1228 003372 105777 175612      2$:   TSTB   @TPCSR     ;TTY READY?
1229 003376 100375              BPL     2$             ;BR IF NO.
1230 003400 112577 175606      MOV    (R5)+,@TPDBR   ;PRINT CURRENT CHAR.
1231 003404 001362              BNE     1$             ;IF NOT ZERO KEEP PRINTING.
1232 003406 012605      3$:   MOV     (SP)+,R5     ;END OF OUTPUT. RESTORE R5
1233 003410 000002      RTI                    ;GO HOME
1234
1235
1236 003412 010346              .INSTR: MOV     R3,-(SP)    ;SAVE R3 ON STACK
1237 003414 010446              MOV     R4,-(SP)      ;SAVE R4 ON STACK
1238 003416 017637 000004 003434  MOV     @4(SP),.MSG
1239 003424 062766 000002 000004  ADD     #2,4(SP)
1240 003432 104402      .INST1: TYPE
1241 003434 000000      .MSG:  0
1242 003436 012704 006160      MOV     #INBUF,R4
  
```

```

1243 003442 012703 000007      MOV      #7,R3
1244 003446 105777 175532      1$: TSTB   @TKCSR
1245 003452 100375          BPL      1$
1246 003454 117714 175526      MOVB    @TKDBR,(R4)
1247 003460 142714 000200      BICB    #200,(R4)
1248 003464 122427 000015      CMPB    (R4)+,#15
1249 003470 001417          BEQ     INSTR2
1250 003472 105777 175512      2$: TSTB   @TPCSR
1251 003476 100375          BPI     2$
1252 003500 017777 175502 175504      MOV     @TKDBR,@TPDBR
1253 003506 005303          DEC     R3
1254 003510 001356          BNE     1$
1255 003512 012604          MOV     (SP)+,R4
1256 003514 012603          MOV     (SP)+,R3
1257 003516 010346      .INSTE: MOV     R3,-(SP)
1258 003520 010446          MOV     R4,-(SP)
1259 003522 104402 005132          TYPE   ,MQM
1260 003526 000741          BR      .INST1
1261 003530 012604      INSTR2: MOV     (SP)+,R4      ;RESTORE R4
1262 003532 012603          MOV     (SP)+,R3      ;RESTORE R3
1263 003534 000002          RTI
1264
1265          ;CONVERT ASCII STRING TO OCTAL
1266          -----
1267
1268 003536 010546      .PARAM: MOV     R5,-(SP)
1269 003540 010446          MOV     R4,-(SP)
1270 003542 016605 000004          MOV     4(SP),R5
1271 003546 012537 003726          MOV     (R5)+,LOLIM
1272 003552 012537 003730          MOV     (R5)+,HILIM
1273 003556 012537 003732          MOV     (R5)+,DEVADR
1274 003562 112537 003734          MOVB    (R5)+,LOBITS
1275 003566 112537 003735          MOVB    (R5)+,ADRCNT
1276 003572 010566 000004          MOV     R5,4(SP)
1277 003576 005005      PARAM1: CLR     R5
1278 003600 012704 006160          MOV     #INBUF,R4
1279 003604 122714 000015          CMPB    #15,(R4)
1280 003610 001420          BEQ     PARERR
1281 003612 121427 000060      1$: CMPB    (R4),#60
1282 003616 002415          BLT     PARERR
1283 003620 121427 000067          CMPB    (R4),#67
1284 003624 003012          BGT     PARERR
1285 003626 142714 000060          BICB    #60,(R4)
1286 003632 152405          BISB    (R4)+,R5
1287 003634 122714 000015          CMPB    #15,(R4)
1288 003640 001406          BEQ     LIMITS
1289 003642 006305          ASL     R5
1290 003644 006305          ASL     R5
1291 003646 006305          ASL     R5
1292 003650 000760          BR      1$
1293 003652 104404      PARERR: INSTER
1294 003654 000750          BR      PARAM1
1295
1296          ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
1297          -----
1298

```

1299	003656	020537	003730	LIMITS:	CMP	R5,HILIM	
1300	003662	101373			BHI	PARERR	
1301	003664	020537	003726		CMP	R5,LOLIM	
1302	003670	103770			BLO	PARERR	
1303	003672	133705	003734		BITB	LOBITS,R5	
1304	003676	001365			BNE	PARERR	
1305							
1306							
1307							
1308	003700	013704	003732				
1309	003704	010524		1\$:	MOV	DEVADR,R4	
1310	003706	062705	000002		MOV	R5,(R4)+	
1311	003712	105337	003735		ADD	#2,R5	
1312	003716	001372			DECB	ADRCNT	
1313	003720	012604			BNE	1\$	
1314	003722	012605			MOV	(SP)+,R4	
1315	003724	000002			MOV	(SP)+,R5	
1316	003726	000000			RTI		
1317	003730	000000		LOLIM:	0		
1318	003732	000000		HILIM:	0		
1319	003734	000000		DEVADR:	0		
1320		003735		LOBITS:	0		
1321				ADRCNT-LOBITS+1			
1322							
1323							
1324							
1325	003736	016637	000004 001266	.SAV05:	MOV	4(SP),SAVPC	;SAVE R7 (PC)
1326							
1327							
1328							
1329	003744	010537	001262	SV05:	MOV	R5,SAVR5	;SAVE R5
1330	003750	010437	001260		MOV	R4,SAVR4	;SAVE R4
1331	003754	010337	001256		MOV	R3,SAVR3	;SAVE R3
1332	003760	010237	001254		MOV	R2,SAVR2	;SAVE R2
1333	003764	010137	001252		MOV	R1,SAVR1	;SAVE R1
1334	003770	010037	001250		MOV	R0,SAVR0	;SAVE R0
1335	003774	000002			RTI		;LEAVE.
1336							
1337							
1338							
1339	003776	013700	001250	.RES05:	MOV	SAVR0,R0	;RESTORE R0
1340	004002	013701	001252		MOV	SAVR1,R1	;RESTORE R1
1341	004006	013702	001254		MOV	SAVR2,R2	;RESTORE R2
1342	004012	013703	001256		MOV	SAVR3,R3	;RESTORE R3
1343	004016	013704	001260		MOV	SAVR4,R4	;RESTORE R4
1344	004022	013705	001262		MOV	SAVR5,R5	;RESTORE R5
1345	004026	000002			RTI		;LEAVE
1346							
1347							
1348							
1349							
1350							
1351	004030	104402	005136	.CONVR:	TYPE	,MCRLF	
1352	004034	010046		.CNVRT:	MOV	R0,-(SP)	
1353	004036	010146			MOV	R1,-(SP)	
1354	004040	010346			MOV	R3,-(SP)	

1355	004042	010446			MOV	R4,-(SP)
1356	004044	010546			MOV	R5,-(SP)
1357	004046	017601	000012		MOV	@12(SP),R1
1358	004052	062766	000002	000012	ADD	#2,12(SP)
1359	004060	012137	004234		MOV	(R1)+,WRDCNT
1360	004064	112137	004236	18:	MOVB	(R1)+,CHRCNT
1361	004070	112137	004237		MOVB	(R1)+,SPACNT
1362	004074	013137	004240		MOV	@(R1)+,BINWRD
1363	004100	013704	004240	28:	MOV	BINWRD,R4
1364	004104	113705	004236		MOVB	CHRCNT,R5
1365	004110	012700	006054		MOV	#TEMP,R0
1366	004114	010403		38:	MOV	R4,R3
1367	004116	042703	177770		BIC	#177770,R3
1368	004122	062703	000060		ADD	#060,R3
1369	004126	110320			MOVB	R3,(R0)+
1370	004130	000241			CLC	
1371	004132	006004			ROR	R4
1372	004134	000241			CLC	
1373	004136	006004			ROR	R4
1374	004140	000241			CLC	
1375	004142	006004			ROR	R4
1376	004144	005305			DEC	R5
1377	004146	001362			BNE	38
1378	004150	012703	006116		MOV	#MDATA,R3
1379	004154	114023		48:	MOVB	-(R0),(R3)+
1380	004156	105337	004236		DECB	CHRCNT
1381	004162	001374			BNE	48
1382	004164	105737	004237		TSTB	SPACNT
1383	004170	001405			BEQ	68
1384	004172	112723	000040	58:	MOVB	#040,(R3)+
1385	004176	105337	004237		DECB	SPACNT
1386	004202	001373			BNE	58
1387	004204	105013		68:	CLRB	(R3)
1388	004206	104402	006116		TYPE	,MDATA
1389	004212	005337	004234		DEC	WRDCNT
1390	004216	001322			BNE	18
1391	004220	012605			MOV	(SP)+,R5
1392	004222	012604			MOV	(SP)+,R4
1393	004224	012603			MOV	(SP)+,R3
1394	004226	012601			MOV	(SP)+,R1
1395	004230	012600			MOV	(SP)+,R0
1396	004232	000002			RTI	
1397	004234	000000			WRDCNT:	0
1398	004236	000000			CHRCNT:	0
1399		004237			SPACNT-CHRCNT+1	
1400	004240	000000			BINWRD:	0
1401						
1402						
1403						
1404						
1405						
1406						
1407						
1408	004242	017605	000000		.SETFLG:MOV	@(SP),R5
1409	004246	042737	000040	006160	BIC	#40,INBUF
1410	004254	122737	000116	006160	CMPB	#'N',INBUF ;IS IT 'N'?

```

1411 004262 001002          BNE      1$
1412 004264 105015          CLRB    (R5)      ;000
1413 004266 000406          BR      2$
1414 004270 122737 00013i 006160 1$:  CMPB   #'Y,INBUF   ;IS IT 'Y'
1415 004276 001005          BNE     3$
1416 004300 112715 177777          MOVB   #-1,(R5)   ;377
1417 004304 062716 000002 2$:  ADD    #2,(SP)
1418 004310 000002          RTI
1419 004312 104404          3$:  INSTER ;RETRY
1420 004314 000752          BR     .SETFLG
1421
1422
1423          ;TRAP DISPATCH SERVICE
1424          ;ARGUMENT OF TRAP IS EXTRACTED
1425          ;AND USED AS OFFSET TO OBTAIN POINTER
1426          ;TO SELECTED SUBROUTINE
1427
1428 004316 011646          .TRPSR: MOV    (SP),-(SP)      ;GET PC OF RETURN
1429 004320 162716 000002          SUB    #2,(SP)          ;=PC OF TRAP
1430 004324 017616 000000          MOV    @ (SP),(SP)      ;GET TRP
1431 004330 006316          TRPOK: ASI    (SP)      ;MULTIPLY TRAP ARG BY 2
1432 004332 042716 177001          BIC    #177001,(SP)     ;CLEAR UNWANTED BITS
1433 004336 062716 001344          ADD    #.TRPTAB,(SP)   ;POINTER TO SUBROUTINE ADDRESS
1434 004342 017616 000000          MOV    @ (SP),(SP)     ;SUBROUTINE ADDRESS
1435 004346 000136          JMP    @ (SP)+         ;GO TO SUBROUTINE
1436
1437          ;ERROR HANDLER
1438          ;-----
1439
1440 004350 032777 010000 174624 .HLT:  BIT    #SW12,@SWR      ;BELL ON ERROR?
1441 004356 001406          BEQ    XBX             ;BR IF NO BELL
1442 004360 105777 174624          TSTB   @TPCSR         ;TTY READY.
1443 004364 100003          BPL    XBX             ;DON'T WAIT IF TTY NOT READY.
1444 004366 112777 000207 174616          MOVB   #207,@TPDBR    ;PUSH A BELL AT THE TTY.
1445 004374 032777 020000 174600 XBX:  BIT    #SW13,@SWR      ;DELETE ERROR PRINT OUT?
1446 004402 001105          BNE    HALTS          ;BR IF NO PRINT OUT WANTED.
1447 004404 021637 001234          CMP    (SP),LSTERR    ;WAS THIS ERROR FOUND LAST TIME?
1448 004410 001404          BEQ    1$             ;BR IF YES
1449 004412 011637 001234          MOV    (SP),LSTERR    ;RECORD BEING HERE
1450 004416 105037 001341          CLRB   ERRFLG        ;PREPARE HEADER
1451 004422 104406          1$:  SAVO5          ;SAVE ALL PROC REGISTERS
1452 004424 011605          MOV    (SP),R5        ;GET THE PC OF ERROR
1453 004426 162705 000002          SUB    #2,R5          ;GET ADDRESS OF TRAP CALL
1454 004432 011504          MOV    (R5),R4        ;GET HLT INSTRUCTION
1455 004434 006304          ASL    R4              ;MULT BY TWO
1456 004436 061504          ADD    (R5),R4        ;DOUBLE IT
1457 004440 006304          ASL    R4              ;MULT AGAIN
1458 004442 042704 177001          BIC    #177001,R4     ;CLEAR JUNK
1459 004446 062704 024234          ADD    #.ERRTAB,R4   ;GET POINTER
1460 004452 012437 004566          MOV    (R4)+,ERRMSG   ;GET ERROR MESSAGE
1461 004456 012437 004600          MOV    (R4)+,DATAHD   ;GET DATA HEADRER
1462 004462 011437 004612          MOV    (R4),DATABP    ;GET DATA TABLE
1463 004466 105737 001341          TSTB   ERRFLG        ;TYPE HEADREER
1464 004472 001403          BEQ    TYPMSG         ;BR IF YES
1465 004474 005737 004612          TST    DATABP        ;DOES DATA TABLE EXIST?
1466 004500 001040          BNE    TYPDAT         ;BR IF YES.

```

1467	004502	104402	005136		TYPMSG: TYPE	,MCRLF	
1468	004506	104402	005136		TYPE	,MCRLF	
1469	004512	005737	001220		TST	LOCK	
1470	004516	001402			BEQ	1\$	
1471	004520	104402	005411		TYPE	,MASTEK	
1472	004524	104402	005377		1\$: TYPE	,MTSTN	
1473	004530	104411	005000		CNVRT	,XTSTN	:SHOW IT
1474	004534	104402	005466		TYPE	,MERRPC	:TYPE PC.
1475	004540	104411	004772		CNVRT	,ERTABO	:SHOW IT
1476	004544	104402	005136		TYPE	,MCRLF	:GIVE A CR/LF
1477	004550	112737	177777	001341	MOVB	#-1,ERRFLG	:NO MORE HEADER UNLESS NO DATA TABLE.
1478	004556	005737	004566		TST	ERRMSG	:IS THERE AN ERROR MESSAGE?
1479	004562	001402			BEQ	WRKO.FM	:BR IF NO.
1480	004564	104402			TYPE		:TYPE
1481	004566	000000			ERRMSG: 0		: ERROR MESSAGE
1482	004570				WRKO.FM:		
1483	004570	005737	004600		TST	DATAHD	:DATA HEADER?
1484	004574	001402			BEQ	TYPDAT	:BR IF NO
1485	004576	104402			TYPE		:TYPE
1486	004600	000000			DATAHD: 0		: DATA HEADER
1487	004602	005737	004612		TYPDAT: TST	DATABP	:DATA TABLE?
1488	004606	001402			BEQ	RESREG	:BR IF NO.
1489	004610	104410			CONVRT		:SHOW
1490	004612	000000			DATABP: 0		: DATA TABLE
1491	004614	104407			RESREG: RES05		:RESTORE PROC REGISTERS
1492	004616	022737	003104	000042	HALTS: CMP	#SENDAD,@#42	:IF ACT-11 AUTO MODE--HALT.
1493	004624	001403			BEQ	1\$	
1494	004626	005777	174350		TST	@SWR	:HALT ON ERROR?
1495	004632	100035			BPL	EXITER	:BR IF NO HALT ON ERROR
1496	004634	010046			1\$: PUSHRO		:SAVE RO
1497	004636	016600	000002		MOV	2(SP),RO	:SHOW ERROR PC IN DATA LIGHTS
1498	004642	013746	000004		MOV	4,-(SP)	:SAVE OLD TRAP
1499	004646	013746	000006		MOV	6,-(SP)	
1500	004652	012737	004710	000004	MOV	#22\$,4	:FORCE HALT IF TIME-OUT
1501	004660	012737	000340	000006	MOV	#340,6	:WHEN REFERENCING TXCSR
1502	004666	042777	014000	174516	BIC	#SYSTST:MEXT,@TXCSR	
1503	004674	000000			HALT		:HALT
1504	004676	012637	000006		MOV	(SP)+,6	:RESTORE TRAP
1505	004702	012637	000004		MOV	(SP)+,4	
1506	004706	000406			BR	33\$	
1507	004710	000000			22\$: HALT		:HALT
1508	004712	022626			CMP	(SP)+,(SP)+	:POP STACK
1509	004714	012637	000006		MOV	(SP)+,6	:RESTORE TRAP
1510	004720	012637	000004		MOV	(SP)+,4	
1511	004724	012600			33\$: POPRO		:GET RO
1512	004726	005237	001232		EXITER: INC	ERRCNT	:UPDATE ERROR COUNT.
1513	004732	032777	000400	174242	BIT	#SW08,@SWR	:GOTO TOP OF TEST?
1514	004740	001007			BNE	1\$:BR IF YES
1515	004742	032777	002000	174232	BIT	#SW10,@SWR	:GOTO NEXT TEST?
1516	004750	001407			BEQ	2\$:BR IF NO
1517	004752	013737	001216	001214	MOV	NEXT,RETURN	:SET FOR NEXT TEST
1518	004760	012706	001150		1\$: MOV	#STACK,SP	:RESET SP
1519	004764	000177	174224		JMP	@RETURN	:GOTO SPECIFIED TEST
1520	004770	000002			2\$: RTI		:RETURN
1521	004772	000001			ERTABO: 1		
1522	004774	006	002		.BYTE	6,2	

```

1523 004776 001266          SAVPC
1524 005000 000001          XTSTN: 1
1525 005002 003          G02      .BYTE 3,2
1526 005004 001226          TSTNO
1527 005006 017600 000000    .PKCLK: MOV @ (SP),R0      ;GET THE # OF TICKS TO POKE
1528 005012 062716 000002    ADD #2,(SP)      ;POP OVER THE #
1529 005016
1530 005016 052777 020000 174366 1$: BIS #CLK,@TXCSR      ;POKE CLOCK UP
1531 005024 005300          DEC R0          ;ARE WE DONE?
1532 005026 001405          BEQ 2$         ;YES-GO TO 2$
1533 005030 042777 020000 174354 BIC #CLK,@TXCSR      ;POKE CLOCK DOWN
1534 005036 005300          DEC R0          ;ARE WE DONE?
1535 005040 001366          BNE 1$         ;NO-REPEAT
1536 005042 000002          2$: RTS        ;RETURN
1537
1538
1539          ;WAIT ROUTINE
1540 005044 000240          SMALL: NOP      ;STALL
1541 005046 000207          RTS PC        ;RETURN
1542
1543          ;POWER FAIL ROUTINE
1544
1545 005050 012737 005060 000024 .PFAIL: MOV #PWRUP,24      ;LOAD PFAIL VECTOR FOR POWER UP
1546 005056 000000          HALT          ;
1547 005060 000005          PWRUP: RESET      ;WAIT TTY TO COME UP
1548 005062 012706 001150          MOV #STACK,SP      ;REINIT STACK POINTER
1549 005066 012737 005050 000024 MOV #.PFAIL,24      ;LOAD PFAIL VECTOR FOR POWER DOWN
1550 005074 104402          TYPE
1551 005076 005141          MPOWER
1552 005100 000177 174110          JMP @RETURN
1553          ;CLRVEC ROUTINE TO FILL COMMUNICATION VECTOR AREA WITH .+2,HALT
1554
1555 005104 012702 000300          CLRVEC: MOV #300,R2      ;R2 COMM VECTOR AREA ADRS
1556 005110 012701 000302          MOV #302,R1      ;INIT R1 WITH ADRS OF HALT
1557 005114 010122          1$: MOV R1,(R2)+      ;MOV .+2 TO PC
1558 005116 005022          CLR (R2)+      ;MOV HALT TO PC
1559 005120 022121          CMP (R1)+,(R1)+  ;INC TO NEXT VECTOR AREA
1560 005122 022701 000776          CMP #776,R1      ;END OF VECTOR AREA
1561 005126 001372          BNE 1$         ;NO
1562 005130 000207          RTS PC        ;RETURN
1563
1564
1565
1566 005132 020040 000077          MQM: .ASCIZ / ?/
(2) 005136 005015 000          MCRLF: .ASCIZ <15><12>
(2) 005141 377 053520 020122 MPOWER: .ASCIZ <377>/PWR FAILED. /
(2) 005157 015 042777 042116 MEPASS: .ASCIZ <15><377>/END PASS CZDPCC /
(2) 005202 051377 000          MR: .ASCIZ <377>/R/
(2) 005205 377 051120 043517 MERR2: .ASCIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT./
(2) 005254 044777 051516 043125 MERR3: .ASCIZ <377>/INSUFFICIENT DATA!/
(2) 005300 052377 051505 020124 MTSTPC: .ASCIZ <377>/TEST PC-/
(2) 005312 046377 041517 020113 MLOCK: .ASCIZ <377>/LOCK ON SELECTED TEST/
(2) 005341 103 051123 020072 MCSRX: .ASCIZ /CSR: /
(2) 005347 126 041505 020072 MVECX: .ASCIZ /VEC: /
(2) 005355 120 051501 042523 MPASSX: .ASCIZ /PASSES: /
(2) 005366 051105 047522 051522 MERRX: .ASCIZ /ERRORS: /

```

```
(2) 005377 124 051505 020124 MTSTN: .ASCIZ /TEST NO: /
(2) 005411 052 000 MASTEK: .ASCIZ /*/
(2) 005413 377 042523 020124 MNEW: .ASCIZ <377>/SET SWITCH REG TO DUPT'S DESIRED ACTIVE./
(2) 005466 041520 020072 000 MERRPC: .ASCIZ /PC: /
(2) 005473 377 042522 020103 MCSR: .ASCIZ <377>/REC CSR ADRS /
(2) 005512 053377 041505 040440 MVEC: .ASCIZ <377>/VEC ADRS /
(2) 005525 377 051511 052040 MJMPR: .ASCIZ <377>/IS THE OPTIONAL CLR JMPR IN? (Y OR N) /
(2) 005600 044777 020123 044124 MTCN: .ASCIZ <377>/IS THE H325 CONNECTOR ON? (Y OR N) /
(2) 005647 377 020043 043117 MTOTAL: .ASCIZ <377>/# OF DUP'S (IN OCTAL) /
(2) 005702 050377 044522 051117 MPAR: .ASCIZ <377>/PRIORITY (4 TO 7) /
(2) 005726 051777 041505 052040 MSTJM: .ASCIZ <377>/SEC TX JMPR IN? (Y OR N) /
(2) 005761 377 042523 020103 MSRJM: .ASCIZ <377>/SEC RX JMPR IN? (Y OR N) /
(2) 006014 046777 050101 047440 XHEAD: .ASCIZ <377>/MAP OF DUP11 STATUS/<377>
(2) .EVEN
(2) 006042 000002 XSTATQ: 2
1567 006044 005 003 .BYTE 6,3
1568 006046 001236 TEMP1
1569 006050 006 002 .BYTE 6,2
1570 006052 001240 TEMP2
1571 .EVEN
1572
1573 006054 000000 TEMP: 0
1574 006116 .=. +40
1575 006116 000000 MDATA: 0
1576 006160 006160 .=. +40
1577 006160 000000 INBUF: 0
1578 006222 006222 .=. +40
1579 006222 000001 TRP.PC: .BLKW 1
1580
```

```

1581
1582
1583 ;ROUTINE USED TO "CYCLE" THROUGH UP TO EIGHT DUP11'S
1584 ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
1585 ;AND RUNS THE SPECIFIED DUP11'S. THIS ROUTINE *MUST*
1586 ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
1587 ;SETUP NECESSARY.
1588
1589
1590 006224 105737 001310 CYCLE: TSTB DUPACTV ;ARE ANY DUP11'S TO BE TESTED?
1591 006230 001004 BNE 1$ ;BR IF OK.
1592 006232 104402 005205 TYPE ,MERR2 ;NO DUP11'S SELECTED!.
1593 006236 000000 HALT ;STOP THE SHGW.
1594 006240 000776 BR -2 ;DISQUALIFY CONT. SW.
1595 006242 133737 001314 001310 1$: BITB RUN,DUPACTV ;IS THIS ONE "ACTIVE"
1596 006250 001020 BNE 2$ ;BR IF GOOD ONE FOUND.
1597 006252 000241 CLC ;CLEAR PROC. CARRY BIT.
1598 006254 106137 001314 ROLB RUN ;UPDATE POINTER
1599 006260 105537 001314 ADCB RUN ;CATCH CARRY FROM RUN
1600 006264 062737 000006 001316 ADD #6,CREAM ;UPDATE ADDRESS POINTER.
1601 006272 022737 001560 001316 CMP #DUP.END,CREAM
1602 006300 001360 BNE 1$ ;KEEP GOING; NOT ALL TESTED FOR.
1603 006302 012737 001500 001316 MOV #DUP.MAP,CREAM ;RESET ADDRESS POINTER.
1604 006310 000754 BR 1$ ;KEEP LOOKING FOR ACTIVE DUP11
1605 006312 000241 2$: CLC ;CLEAR PROC. CARRY.
1606 006314 106137 001314 ROLB RUN ;UPDATE POINTER.
1607 006320 105537 001314 ADCB RUN ;CATCH CARRY.
1608 006324 013700 001316 MOV CREAM,RO ;GET ADDRESS POINTER.
1609 006330 062737 000006 001316 ADD #6,CREAM ;UPDATE.
1610 006336 022737 001560 001316 CMP #DUP.END,CREAM
1611 ;ALL DONE?
1612 006344 001003 BNE 3$ ;BR IF NO.
1613 006346 012737 001500 001316 MOV #DUP.MAP,CREAM ;RESTORE POINTER.
1614 006354 012037 001404 3$: MOV (RO)+,RXCSR ;LOAD SYSTEM CTRL. REG
1615 006360 012037 001374 MOV (RO)+,DUPRVC ;LOAD VECTOR
1616 006364 012037 001434 MOV (RO)+,LOO.OO ;GET PARAMETERS
1617 006370 012700 000002 MOV #2,RO ;SAVE CORE THIS WAY!
1618 006374 013737 001404 001424 MOV RXCSR,HUPRCR ;GET CONTROL REG HIGH BYTE
1619 006402 005237 001424 INC HUPRCR ;GOT IT
1620 006406 013737 001424 001406 MOV HUPRCR,RXDBUF ;GET RX CONTROL REG BUFFER
1621 006414 005237 001406 INC RXDBUF ;GOT IT
1622 006420 013737 001406 001416 MOV RXDBUF,DUPSEC ;GOT SECONDARY REG SELECT REG
1623 006426 013737 001406 001410 MOV RXDBUF,PARCSR ;GOT PARAMETER STATUS REGISTER
1624 006434 013737 001406 001422 MOV RXDBUF,HUPRBF ;GET RX BUFFER HIGH BYTE
1625 006442 005237 001422 INC HUPRBF ;GOT IT
1626 006446 013737 001422 001420 MOV HUPRBF,HUPPSR ;GOT PAR STATUS REG HIGH BYTE
1627 006454 013737 001420 001412 MOV HUPPSR,TXCSR ;GET TX CONTROL REGISTER
1628 006462 005237 001412 INC TXCSR ;GOT IT
1629 006466 013737 001412 001430 MOV TXCSR,HUPTCR ;GET TX CONTROL REG HIGH BYTE
1630 006474 005237 001430 INC HUPTCR ;GOT IT
1631 006500 013737 001430 001414 MOV HUPTCR,TXDBUF ;BET TX BUFFER
1632 006506 005237 001414 INC TXDBUF ;GOT IT
1633 006512 013737 001414 001426 MOV TXDBUF,HUPTBF ;GET TX BUFFER HIGH BYTE
1634 006520 005237 001426 INC HUPTBF ;GOT IT
1635
1636 006524 013737 001374 001376 MOV DUPRVC,DUPRPS ;RX VECTOR
  
```

```

1637 006532 060037 001376          ADD    RO,DUPRPS      ;RX PRIORITY LEVEL
1638 006536 013737 001376 001400    MOV    DUPRPS,DUPTVC
1639 006544 060037 001400          ADD    RO,DUPTVC     ;TX VECTOR
1640 006550 013737 001400 001402    MOV    DUPTVC,DUPTPS
1641 006556 060037 001402          ADD    RO,DUPTPS     ;TX PRIORITY LEVEL
1642
1643
1644 006562 012700 001434          MOV    #L00.00,RO    ;LOAD STAUS 00-00
1645 006566 012701 001432          MOV    #MASK.A,R1   ;PREPARE MASK.
1646 006572 012702 001433          MOV    #CLK.A,R2    ;PREPARE CLOCKS
1647 006576 004737 006742          JSR    PC,FIX.00    ;GO AND CALCULATE CONFIGURATION.
1648 006602 005737 000042          TST    @#42
1649 006606 001050          BNE    4$
1650 006610 032777 000002 172364    BIT    #SW01,@SWR   ;IF SW01=1,GET STARTING TEST #
1651 006616 001444          BEQ    4$
1652 006620 104402 005136          7$:   TYPE    ,MCRLF
1653 006624 104403          INSTR  ;OUTPUT MESSAGE & GET INPUT STRING
1654 006626 005377          MTSTN  ;MESSAGE
1655 006630 104405          PARAM ;CONVERT STRING
1656 006632 000001          1     ;LOW LIMIT
1657 006634 001000          1000  ;HIGH LIMIT
1658 006636 001226          TSTNO  ;STORE AT THIS LOCATION
1659 006640          000   .BYTE  0     ;MASK
1660 006641          001   .BYTE  1     ;HOW MANY TIMES * 2
1661 006642 012700 007356          MOV    #TST1,RO
1662 006646 022710 012737          5$:   CMP    #12737,(RO)
1663 006652 001017          BNE    6$
1664 006654 023760 001226 000002    CMP    TSTNO,2(RO)
1665 006662 001013          BNE    6$
1666 006664 022760 001226 000004    CMP    #TSTNO,4(RO)
1667 006672 001007          BNE    6$
1668 006674 010037 001214          MOV    RO,RETURN    ;SAVE PC
1669 006700 104402 005136          TYPE  ,MCRLF
1670 006704 104402 005202          TYPE  ,MR
1671 006710 000412          BR     8$
1672 006712 005720          6$:   TST    (RO)+
1673 006714 020027 022546          CMP    RO,#TLAST+10
1674 006720 001352          BNE    5$
1675 006722 104402 005132          TYPE  ,MQM
1676 006726 000734          BR     7$
1677
1678 006730 012737 007356 001214 4$:   MOV    #TST1,RETURN ;PREPARE RETURN ADDRESS
1679 006736 000177 172252          8$:   JMP    @RETURN      ;GO START TESTING.
1680
1681 006742 011003          FIX.00: MOV   (RO),R3    ;GET PARAMETERS.
1682 006744 000207          5$:   RTS    PC        ;

```

1683
 1684
 1685
 1686
 1687
 1688
 1689
 1690
 1691
 1692
 1693
 1694
 1695
 1696
 1697
 1698
 1699
 1700
 1701
 1702
 1703
 1704
 1705
 1706
 1707
 1708
 1709
 1710
 1711
 1712
 1713
 1714
 1715
 1716
 1717
 1718
 1719
 1720
 1721
 1722
 1723
 1724
 1725
 1726
 1727
 1728
 1729
 1730
 1731
 1732
 1733
 1734
 1735
 1736
 1737
 1738

006746 012737 000010 001326
 006754 013700 001324
 006760 012701 002200
 006764 006000
 006766 005537 001330
 006772 022737 000005 001330
 007000 003004
 007002 005037 001330
 007006 005237 001326
 007012 006001
 007014 103363
 007016 000207

 007020 042777 002000 172364
 007026 006037 001324
 007032 103003
 007034 052777 002000 172350
 007042 000240
 007044 104412 000002
 007050 005337 001326
 007054 001361
 007056 000201

 007060 012137 001244
 007064 005037 001246
 007070 042777 002000 172314
 007076 104412 000002
 007102 052777 002000 172302
 007110 104412 000002
 007114 005237 001246
 007120 022737 000006 001246
 007126 001365
 007130 042777 002000 172254
 007136 104412 000002
 007142 005337 001244
 007146 001346
 007150 000201

 007152 012137 001244
 007156 005037 001246
 007162 052777 002000 172222
 007170 104412 000002

:ROUTINE TO DETERMINE IF BIT STUFFING IS REQUIRED

STUFCK: MOV #8,SHIFTS
 MOV DATA,R0
 MOV #200,R1
 1\$: ROR R0
 ADC MIND
 CMP #5,MIND
 BGT 2\$
 CLR MIND
 INC SHIFTS
 2\$: ROR R1
 BCC 1\$
 RTS PC

:THIS ROUTINE POKES THE RECEIVER BIT WINDOW
 :BASED ON THE INFORMATION IN DATA AND SHIFTS

RPOKE: BIC #BITW,@TXCSR
 ROR DATA
 BCC 1\$
 BIS #BITW,@TXCSR
 1\$: NOP
 PKCLK ,2
 DEC SHIFTS
 BNE RPOKE
 RTS R1

RFLG: MOV (R1)+,TEMP4 ;GET THE # OF FLAGS
 1\$: CLR TEMP5 ;CLEAR ONES COUNT
 BIC #BITW,@TXCSR ;SET FIRST BIT
 PKCLK ,2 ;PUSH OUT THE BIT
 2\$: BIS #BITW,@TXCSR ;LOAD THE BIT
 PKCLK ,2 ;PUSH OUT THE BIT
 INC TEMP5 ;INCREMENT 1'S COUNTER
 CMP #6,TEMP5 ;DID WE PUSH OUT 6 ONES
 BNE 2\$;NO-GO BACK
 BIC #BITW,@TXCSR ;SET THE LAST BIT
 PKCLK ,2 ;PUSH OUT THE BIT
 DEC TEMP4 ;ARE WE DONE WITH FLAGS?
 BNE 1\$;BR IF NO
 RTS R1

SFLG: MOV (R1)+,TEMP4 ;GET THE # OF FLAGS
 1\$: CLR TEMP5 ;CLEAR ONES COUNT
 2\$: BIS #BITW,@TXCSR ;LOAD THE BIT
 PKCLK ,2 ;PUSH OUT THE BIT

```

1739 007174 005237 001246      INL      TEMPS      ; INCREMENT 1'S COUNTER
1740 007200 022737 000006 001246      CMP      #6,TEMPS   ; DID WE PUSH OUT 6 ONES
1741 007206 001365                BNE      2$         ; NO-GO BACK
1742 007210 042777 002000 172174      BIC      #BITW,@TXCSR ; SET THE LAST BIT
1743 007216 104412 000002                PKCLK    ,2         ; PUSH OUT THE BIT
1744 007222 005337 001244                DEC      TEMP4      ; ARE WE DONE WITH FLAGS?
1745 007226 001353                BNE      1$         ; BR IF NO
1746 007230 000201                RTS      R1
1747
1748 007232 012577 172136      SETVEC: MOV      (R5)+,@DUPRVC
1749 007236 012577 172136                MOV      (R5)+,@DUPTVC
1750 007242 112577 172130                MOV      (R5)+,@DUPRPS
1751 007246 112577 172130                MOV      (R5)+,@DUPTPS
1752 007252 000205                RTS      R5
1753 007254                NO.ATRIP:
1754 007254 104025                HLT      25
1755 007256 000002                RTI
1756
1757 007260                NO.BTRIP:
1758 007260 104026                HLT      26
1759 007262 000002                RTI
1760
1761                ; THIS ROUTINE PICKS UP THE ADDRESS OF
1762                ; THE JUMPER TABLE AND LOADS R5 WITH
1763                ; THE CORRECT DATA BASED ON THE STATE
1764                ; OF THE JUMPER AND CONNECTOR FLAGS.
1765                ;-----
1766
1767 007264 012100      JUMPER: MOV      (R1)+,R0      ; GET THE TABLE ADDRESS
1768 007266 105737 001322      TSTB    TCNFLG      ; TEST THE TURN AROUND CONNECTOR FLAG
1769 007272 001406                BEQ      2$         ; BRANCH IF CONNECTOR IS MISSING
1770 007274 105737 001323      TSTB    OPCLRJ      ; TEST CLEAR JUMPER FLAG
1771 007300 001403                BEQ      2$         ; BRANCH IF JUMPER IS MISSING
1772 007302 011005                MOV      (R0),R5    ; MOVE THE DATA TO R5, BOTH JUMPER
1773                ; AND CONNECTOR ARE THERE
1774 007304 000137 007330                JMP      5$
1775 007310 022020      2$:      CMP      (R0)+,(R0)+      ; POP POINTER
1776 007312 105737 001323      TSTB    OPCLRJ      ; TEST CLEAR JUMPER FLAG
1777 007316 001403                BEQ      3$         ; BRANCH IF MISSING
1778 007320 011005                MOV      (R0),R5    ; MOVE DATA- JUMPER IN, CONNECTOR OFF
1779 007322 000137 007330                JMP      5$
1780 007326 012005      3$:      MOV      (R0)+,R5      ; NO CONNECTOR OR JUMPER
1781 007330 000201      5$:      RTS      R1          ; RETURN
1782
1783 007332 012100      OJUMPER:MOV     (R1)+,R0      ; GET THE POINTER ADDRESS
1784 007334 105737 001322      TSTB    TCNFLG      ; CHECK FOR TURNAROUND CONNECTOR
1785 007340 001403                BEQ      4$         ; BR IF MISSING
1786 007342 011005                MOV      (R0),R5    ; MOVE THE INFO TO R5
1787 007344 000137 007354                JMP      6$         ; GO BACK
1788 007350 022020      4$:      CMP      (R0)+,(R0)+      ; POP POINTER
1789 007352 011005                MOV      (R0),R5    ; LOAD DATA TO R5
1790 007354 000201      6$:      RTS      R1          ; RETURN
1791
1792                ; ***** TEST 1 *****
1793                ; *TEST TO PROVE THE INTERACTION OF RECEIVER ENABLE*
1794                ; *WITH RECEIVER ACTIVE AND RECEIVE START

```

TEST OF REC ENABLE, RXACT, AND RSOM IN PRIMARY MODE

: OF MESSAGE IN PRIMARY MODE.

:

:

TEST 1

:

1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805 007356 012737 000001 001226
1806 007364 012737 007640 001216
1807 007372 052777 000400 172012
1808 007400 004737 005044
1809 007404 013703 001404
1810 007410 052777 001000 171772
1811 007416 052713 000020
1812 007422 052777 01400C 171762
1813 007430 004137 007060
1814 007434 000001
1815 007436 032713 004000
1816
1817 007442 001401
1818 007444 104006
1819 007446 005037 001324
1820 007452 012737 000011 001326
1821 007460 004137 007020
1822 007464 032713 000200
1823 007470 001004
1824 007472 012705 000200
1825 007476 011304
1826 007500 104003
1827 007502 032777 000400 171676
1828 007510 001006
1829 007512 012705 000400
1830 007516 013703 001406
1831 007522 011304
1832 007524 104003
1833
1834 007526 013703 001404
1835 007532 032713 004000
1836 007536 001001
1837 007540 104014
1838 007542 005037 001324
1839 007546 012737 000011 001326
1840 007554 004137 007020
1841 007560 032713 000200
1842 007564 001004
1843 007566 012705 000200
1844 007572 011304
1845 007574 104003
1846 007576 032777 000400 171602
1847 007604 001406
1848 007606 012705 000400
1849 007612 013703 001406
1850 007616 011304

```
TST1:  MOV #1,@TSTNO  
      MOV #TST2,NEXT  
      BIS #MRESET,@TXCSR ;RESET THE DEVICE  
      JSR PC,SMALL ;WAIT FOR RESET TO FINISH  
      MOV RXCSR,R3 ;GET THE RECEIVER CONTROL REGISTER  
      BIS #CRCEN,@PARCSR ;TURN OFF CRC  
      BIS #RCVEN,(R3) ;TURN ON THE RECEIVER  
      BIS #MMODE,@TXCSR ;ENTER M/MODE  
      JSR R1,RFLG ;PUSH OUT A FLAG  
      BIT #REACT,(R3) ;RXACT SHOULD BE CLEAR WHEN READING FLAG CHAR.  
      BEQ 1$ ;BRANCH IF BIT IS OFF  
      HLT 6 ;BIT WAS SET - SHOULD BE CLEAR  
1$:   CLR DATA ;SETUP FOR CHAR  
      MOV #9,,SHIFTS ;POKE CLOCK SETUP  
      JSR R1,RPOKE ;PUSH DATA INTO RECEIVER  
      BIT #RXDONE,(R3) ;TEST DONE  
      BNE 2$ ;BR IF SET  
      MOV #RXDONE,R5 ;ERROR MSG SETUP  
      MOV (R3),R4 ;SET UP FOUND  
      HLT 3 ;RXDONE FAILED TO SET  
2$:   BIT #RSOM,@RXDBUF ;RSOM SHOULD BE SET  
      BNE 3$ ;BR IF SET  
      MOV #RSOM,R5 ;SET EXPECTED  
      MOV RXDBUF,R3 ;SET UP ERROR MSG  
      MOV (R3),R4 ;SET FOUND  
      HLT 3 ;RSOM FAILED TO SET  
      ;//////////////////////////////////////: *C  
3$:   MOV RXCSR,R3  
      BIT #REACT,(R3) ;SHOULD BE SET FOR START OF FRAME  
      BNE 4$ ;BR IF SET  
      HLT 14 ;FAILED TO SET  
4$:   CLR DATA ;SETUP FOR CHAR  
      MOV #9,,SHIFTS ;POKE CLOCK SETUP  
      JSR R1,RPOKE ;PUSH DATA INTO RECEIVER  
      BIT #RXDONE,(R3) ;RXDONE SHOULD BE SET  
      BNE 5$ ;BR IF SET  
      MOV #RXDONE,R5 ;ERROR MSG SETUP  
      MOV (R3),R4 ;SET UP FOUND  
      HLT 3 ;RXDONE FAILED TO SET  
5$:   BIT #RSOM,@RXDBUF ;RSOM SHOULD BE CLEAR  
      BEQ 10$ ;THIS IS SECOND DATA CHAR OF FRAME  
      MOV #RSOM,R5 ;SET EXPECTED  
      MOV RXDBUF,R3 ;SET UP ERROR MSG  
      MOV (R3),R4 ;SET FOUND
```

TEST OF REC ENABLE, RXACT, AND RSOM IN PRIMARY MODE

1851 007620 104032
1852
1853
1854 007622 013703 001404
1855 007626 032713 004000
1856 007632 001001
1857 007634 104014
1858 007636 104400
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871

HLT 32 ;RSOM FAILED TO CLEAR
:////////////////////
10\$: MOV RXCSR,R3 ;SHOULD BE SET FOR START OF FRAME
BIT #REACT,(R3) ;BR IF SET
BNE 11\$;FAILED TO SET
HLT 14
11\$: SCOPE

:***** TEST 2 *****
: *TEST TO PROVE THE INTERACTION OF RECEIVER ENABLE
: *WITH RECEIVER ACTIVE AND RECEIVE
: *START OF MESSAGE IN SECONDARY MODE.
:*****

:*****
: *
: TEST 2
: *
:*****

1872 007640 012737 000002 001226
1873 007646 012737 010204 001216
1874 007654 052777 000400 171530
1875 007662 004737 005044
1876 007666 013703 001404
1877 007672 052777 001000 171510
1878 007700 052713 000020
1879 007704 052777 010000 171476
1880 007712 052777 014000 171472
1881 007720 004137 007060
1882 007724 000001
1883 007726 032713 004000
1884
1885 007732 001403
1886 007734 005005
1887 007736 011304
1888 007740 104003
1889
1890 007742 032713 000200
1891 007746 001401
1892 007750 104003
1893
1894 007752 065037 001324
1895 007756 012737 000011 001326
1896 007764 004137 007020
1897 007770 032713 004000
1898 007774 001004
1899 007776 012705 004000
1900 010002 011304
1901 010004 104003
1902 010006 032713 000200
1903 010012 001404
1904
1905 010014 012705 000200
1906 010020 011304

:*****
:*****
TST2: MOV #2,@TSTNO
MOV #TST3,NEXT
BIS #MRESET,@TXCSR ;RESET THE DEVICE
JSR PC,SMALL ;WAIT FOR RESET TO FINISH
MOV RXCSR,R3 ;GET THE RECEIVER CONTROL REGISTER
BIS #CRCEN,@PARCSR ;TURN OFF CRC
BIS #RCVEN,(R3) ;TURN ON THE RECEIVER
BIS #PRISEC,@PARCSR ;ENTER SECONDARY MODE
BIS #MMODE,@TXCSR ;ENTER M/MODE
JSR R1,RFLG ;PUSH OUT A FLAG
BIT #REACT,(R3) ;RXACT SHOULD BE CLEAR WHEN READING FLAG CHAR.
BEQ 3\$;BR IF OFF
CLR R5 ;SET EXPECTED
MOV (R3),R4 ;SET FOUND
HLT 3 ;BIT WAS SET AND SHOULD BE CLEARED
;WE ARE IN SECONDARY MODE
3\$: BIT #RXDONE,(R3) ;SHOULD BE CLEAR
BEQ 4\$;BR IF CLEAR
HLT 3 ;DONE IS SET - SHOULD NOT BE BECAUSE
;WE ARE IN SECONDARY MODE AND HAVE NOT PUSHED DATA
4\$: CLR DATA ;CLEAR DATA CHAR
MOV #9,SHIFTS ;LOAD THE # OF BITS TO PUSH
JSR R1,RPOKE ;PUSH DATA TO RECEIVER
BIT #REACT,(R3) ;RXACT SHOULD BE SET
BNE 5\$;BR IF SET
MOV #REACT,R5 ;SET EXPECTED
MOV (R3),R4 ;SET FOUND
HLT 3 ;ACTIVE FAILED TO SET
5\$: BIT #RXDONE,(R3) ;RXDONE SHOULD BE CLEAR
BEQ 6\$;BR IF CLEAR - THE CHAR WAS
;THE SECONDARY ADRS
MOV #RXDONE,R5 ;SET EXPECTED
MOV (R3),R4 ;SET FOUND

TEST OF REC ENABLE, RXACT, AND RSOM IN SECONDARY MODE

```
1907 010022 104003          HLT      3          ;DONE WAS SET AND SHOULDN'T BE - THE FIRST
1908                                ;DATA WAS SECONDARY ADRS, NOT DATA TO THE BUFFER
1909 010024 032777 000400 171354 68:  BIT      #RSOM,@RXDBIF ;CHECK START OF MSG
1910 010032 001401          BEQ      78          ;BR IF CLEAR
1911 010034 104020          HLT      20          ;START OF MSG SHOULD BE CLEAR
1912 010036 112737 000252 001324 78:  MOVB     #252,DATA ;LOAD DATA CHAR
1913 010044 012737 000010 001326  MOV      #8.,SHIFTS ;LOAD CLOCK COUNT
1914 010052 004137 007020          JSR      R1,RPOKE   ;PUSH OUT DATA
1915 010056 104412 000002          PKCLK    ,2         ;POKE ANOTHER FULL CLOCK
1916 010062 032713 000200          BIT      #RXDONE,(R3) ;TEST DONE
1917 010066 001004          BNE     128         ;BR IF SET
1918 010070 012705 000200          MOV      #RXDONE,R5 ;SET EXPECTED
1919 010074 011304          MOV      (R3),R4    ;SET FOUND
1920 010076 104003          HLT      3          ;DONE FAILED TO SET AFTER PUSHING
1921                                ;OUT A CHAR IN SEC. MODE
1922                                ;//////////////////////////////////////:;*C
1923 010100 032777 000400 171300 128: BIT      #RSOM,@RXDBUF ;RSOM SHOULD BE SET
1924 010106 001001          BNE     148         ;BR IF SET
1925 010110 104021          HLT      21          ;RSOM CLEAR
1926 010112 112737 000252 001324 148: MOVB     #252,DATA ;LOAD DATA CHAR
1927 010120 01 37 000010 001326  MOV      #8.,SHIFTS ;LOAD CLOCK COUNT
1928 010126 00 37 007020          JSR      R1,RPOKE   ;PUSH OUT DATA
1929 010132 104412 000002          PKCLK    ,2         ;POKE ANOTHER FULL CLOCK
1930 010136 032713 000200          BIT      #RYDONE,(R3) ;TEST DONE
1931 010142 001004          BNE     158         ;BR IF SET
1932 010144 012705 000200          MOV      #RXDONE,R5 ;SET EXPECTED
1933 010150 011304          MOV      (R3),R4    ;SET FOUND
1934 010152 104003          HLT      3          ;DONE FAILED TO SET
1935 010154 032777 000400 171224 158: BIT      #RSOM,@RXDBUF ;RSOM SHOULD BE CLEAR
1936 010162 001401          BEQ     108         ;BR IF CLEAR
1937 010164 104032          HLT     32         ;FAILED TO CLEAR
1938                                ;//////////////////////////////////////
1939 010166 013703 001404          MOV      RXCSR,R3   ;SHOULD BE SET FOR START OF FRAME
1940 010170 032713 004000          BIT      #REACT,(R3) ;BR IF SET
1941 010176 001001          BNE     118         ;FAILED TO SET
1942 010200 104014          HLT     14
1943 010202 104400          118:  SCOPE
1944
1945                                ;***** TEST 3 *****
1946                                ;*TEST TO PROVE THE INTERACTION OF REOM
1947                                ;*WITH DONE IN PRIMARY MODE
1948
1949                                ;*****
1950                                ;*****
1951                                ;*****
1952                                ;*****
1953                                ;*****
1954                                ;*****
1955                                ;*****
1956 010204 012737 000003 001226  TST3:  MOV      #3,@TSTNO
1957 010212 012737 010472 001216  MOV      #TST4,NEXT
1958 010220 052777 000400 171164  BIS      #MRESET,@TXCSR ;RESET THE DEVICE
1959 010226 004737 005044          JSR      PC,SMALL   ;WAIT FOR RESET TO FINISH
1960 010232 052777 000020 171144  BIS      #PCVEN,@RXCSR ;TURN ON RECEIVER
1961 010240 052777 001000 171142  BIS      #CRCEN,@PARCSR ;TURN OFF CRC
1962 010246 005037 001236          CLR      TEMP1     ;BIT COUNTER
```



```
2019 010526 052777 001000 170654     BIS    #CRCEN,@PARCSR  ;TURN OFF CRC
2020 010534 005037 001236                CLR    TEMP1          ;BIT COUNTER
2021 010540 052777 014000 170644     BIS    #MMODE,@TXCSR  ;ENTER MAINT MODE
2022 010546 004137 007060                JSR    R1,RFLG        ;PUSH OUT A FLAG
2023 010552 000001                1          ;ONE FLAG
2024 010554 004137 007152                JSR    R1,SFLG        ;PUSH OUT SPECIAL FLAGS
2025 010560 000004                4          ;THE # TO DO
2026 010562 012737 000125 001324  MOV    #125,DATA      ;CLEAR OUT CHAR
2027 010570 012737 000010 001326  MOV    #8,,SHIFTS    ;LOAD THE # OF CLOCKS
2028 010576 004137 007020                JSR    R1,RPOKE       ;PUSH OUT THE DATA
2029 010602 042777 002000 170602     BIC    #BITW,@TXCSR  ;CLEAR THE DATA WINDOW
2030 010610 104412 000002                PKCLK  ,2            ;PUSH OUT THE BIT
2031 010614 032777 000200 170562     BIT    #RXDONE,@RXCSR ;CHECK FOR DONE
2032 010622 001001                BNE    5$            ;BR IF SET
2033 010624 104000                HLT                    ;DONE BIT ERROR
2034 010626 117737 170554 006054  5$:   MOVB  @RXDBUF,TEMP    ;GET DATA
2035 010634  2737 000125 006054  CMPB  #125,TEMP      ;CHECK IT
2036 010642 001401                BEQ    64$          ;BR IF A MATCH
2037 010644 104010                HLT    10          ;DATA COMPARE ERROR
2038 010646 052777 002000 170536  64$:  BIS    #BITW,@TXCSR  ;PUT A 1 IN THE WINDOW
2039 010654 104412 000002                PKCLK  ,2            ;PUSH IT OUT
2040 010660 005237 001236                INC    TEMP1         ;INC THE # TO DO
2041 010664 022737 000006 001236     CMP    #6,TEMP1     ;CHECK FOR FINISH
2042 010672 001365                BNE    64$          ;BR IF MORE TO GO
2043 010674 042777 002000 170510     BIC    #BITW,@TXCSR  ;CLEAR THE WINDOW
2044 010702 104412 000004                PKCLK  ,4            ;PUSH OUT 2 BINTS
2045 010706 017737 170472 001240     MOV    @RXCSR,TEMP2  ;GET THE CSR
2046 010714 032777 001000 170464     BIT    #REOM,@RXDBUF ;TEST FOR END OF MSG
2047 010722 001001                BNE    1$            ;BR IF SET
2048 010724 104001                HLT    1            ;BIT FAILED TO SET
2049 010726 032737 000200 001240  1$:   BIT    #RXDONE,TEMP2 ;TEST DONE
2050 010734 001001                BNE    10$          ;BR IF SET
2051 010736 104002                HLT    2            ;DONE FAILED TO SET
2052  ;//////////////////////////////////////:;
2053 010740 004137 007060 10$:   JSR    R1,RFLG        ;OUTPUT A FLAG CHAR
2054 010744 000001                1
2055 010746 017737 170432 001240     MOV    @RXCSR,TEMP2  ;GET CSR
2056 010754 032777 001000 170424     BIT    #RFOM,@RXDBUF ;REOM SHOULD BE CLEAR
2057 010762 001401                BEQ    11$          ;BR IF CLEAR
2058 010764 104030                HLT    30          ;REOM FAILED TO CLEAR
2059 010766 032737 000200 001240  11$:  BIT    #RXDONE,TEMP2 ;RXDONE SHOULD BE SET
2060 010774 001001                BNE    4$            ;BR IF SET
2061 010776 104002                HLT    2            ;FAILED TO SET
2062  ;//////////////////////////////////////:;
2063 011000 104400 4$:   SCOPE                ;SCOPE THIS TEST
2064
2065 ;***** TEST 5 *****
2066 ;*TEST TO PROVE THE INTERACTION OF REOM
2067 ;*WITH DONE IN SECONDARY MODE. TEST FOR REOM
2068 ;*AT THE WRONG ADDRESS, THEN AT THE CORRECT
2069 ;*SECONDARY ADDRESS.
2070 ;*****
2071 ;*****
2072 ;*
2073 ;* TEST 5
2074 ;*
```

TEST OF REOM AND DONE IN SECONDARY MODE

```

2075
2076
2077 011002 012737 000005 001226
2078 011010 012737 011410 001216
2079 011016 052777 000400 170366
2080 011024 004737 005044
2081 011030 052777 000020 170346
2082 011036 052777 001000 170344
2083 011044 005037 001236
2084 011050 052777 010000 170332
2085 011056 052777 014000 170326
2086 011064 004137 007060
2087 011070 000001
2088 011072 012737 000001 001324
2089 011100 012737 000010 001326
2090 011106 004137 007020
2091 011112 042777 002000 170272
2092 011120 104412 000002
2093 011124 032777 000200 170252
2094 011132 001401
2095 011134 104000
2096 011136 105777 170244
2097 011142 052777 002000 170242
2098 011150 104412 000002
2099 011154 005237 001236
2100 011160 022737 000006 001236
2101 011166 001365
2102 011170 042777 002000 170214
2103 011176 104412 000004
2104 011202 032777 001000 170176
2105 011210 001401
2106 011212 104004
2107
2108 011214 004137 007060
2109 011220 000001
2110 011222 005037 001324
2111 011226 012737 000010 001326
2112 011234 004137 007020
2113 011240 012737 000252 001324
2114 011246 012737 000010 001326
2115 011254 004137 007020
2116 011260 042777 002000 170124
2117 011266 104412 000002
2118 011272 032777 000200 170104
2119 011300 001001
2120 011302 104002
2121 011304 005037 001236
2122 011310 105777 170072
2123 011314 052777 002000 170070
2124 011322 104412 000002
2125 011326 005237 001236
2126 011332 022737 000006 001236
2127 011340 001365
2128 011342 042777 002000 170042
2129 011350 104412 000004
2130 011354 017737 170024 001240

```

```

:*****
:*****
TST5:  MOV    #5,@TSTNO
      MOV    #TST6,NEXT
      BIS    #MRESET,@TXCSR ;RESET THE DEVICE
      JSR    PC,SMALL      ;WAIT FOR RESET TO FINISH
      BIS    #RCVEN,@RXCSR ;TURN ON RECEIVER
      BIS    #CRCEN,@PARCSR ;TURN OFF CRC
      CLR    TEMP1         ;BIT COUNTER
      BIS    #PRISEC,@PARCSR ;ENTER SECONDARY MODE
      BIS    #MMODE,@TXCSR ;ENTER MAINT MODE
      JSR    R1,RFLG      ;PUSH OUT A FLAG
      1                ;ONE FLAG
      MOV    #1,DATA      ;LOAD A CHAR
      MOV    #8.,SHIFTS   ;LOAD THE # OF CLOCKS
      JSR    R1,RPOKE     ;PUSH OUT THE DATA
      BIC    #BITW,@TXCSR ;CLEAR THE DATA WINDOW
      PKCLK ,2           ;PUSH OUT THE BIT
      BIT    #RXDONE,@RXCSR ;CHECK FOR DONE
      BEQ    5$          ;BR 1 CLEAR
      HLT
      5$:  TSTB   @RXDBUF   ;READ THE BUFFER
      64$: BIS    #BITW,@TXCSR ;PUT A 1 IN THE WINDOW
      PKCLK ,2           ;PUSH IT OUT
      INC    TEMP1       ;INC THE # TO DO
      CMP    #6,TEMP1    ;CHECK FOR FINISH
      BNE    64$        ;BR IF MORE TO GO
      BIC    #BITW,@TXCSR ;CLEAR THE WINDOW
      PKCLK ,4           ;PUSH OUT 2 BINTS
      BIT    #REOM,@RXDBUF ;TEST REC END OF MSG
      BEQ    2$          ;BR IF NOT SET
      HLT    4           ;BIT IS SET AND SHOULDN'T
                        ;BE - THIS IS SECONDARY MODE
      2$:  JSR    R1,RFLG   ;OUTPUT A FLAG
      1
      CLR    DATA      ;CLEAR DATA
      MOV    #8.,SHIFTS ;LOAD THE # OF CLOCKS
      JSR    R1,RPOKE   ;PUSHOUT SEC. ADRS
      MOV    #252,DATA  ;LOAD DATA
      MOV    #8.,SHIFTS ;LOAD # OF CLOCKS
      JSR    R1,RPOKE   ;PUSHOUT DATA
      BIC    #BITW,@TXCSR ;CLEAR OUT DATA WINDOW
      PKCLK ,2         ;PUSH OUT A BIT
      BIT    #RXDONE,@RXCSR ;CHECK DONE
      BNE    6$        ;BR IF SET
      HLT    2         ;DONE FAILED TO SET
      6$:  CLR    TEMP1   ;CLEAR TO KEEP TRACK OF THE BITS
      TSTB   @RXDBUF   ;READ THE BUFFER
      65$: BIS    #BITW,@TXCSR ;SET THE WINDOW BIT
      PKCLK ,2         ;PUSH IT OUT
      INC    TEMP1     ;CHECK TO SEE IF
      CMP    #6,TEMP1  ;6 ARE DONE YET
      BNE    65$      ;BR IF NO
      BIC    #BITW,@TXCSR ;CLEAR THE WINDOW
      PKCLK ,4         ;PUSH 2 BITS
      MOV    @RXCSR,TEMP2 ;READ THE CSR

```

TEST OF REOM AND DONE IN SECONDARY MODE

```
2131 011362 032777 001000 170016 BIT #REOM,@RXDBUF ;TEST END OF MSG
2132 011370 001001 BNE 3$ ;BR IF SET
2133 011372 104001 HLT 1 ;EOM FAILED TO SET IN SEC. MODE
2134 011374 032737 000200 001240 3$: BIT #RXDONE,TEMP2 ;TEST DONE
2135 011402 001001 BNE 4$ ;BR IF SET
2136 011404 104002 HLT 2 ;BIT FAILED TO SET
2137
2138 011406 104400 4$: SCOPE ;SCOPE THIS TEST
2139
```

```
***** TEST 6 *****
; *TEST TO PROVE THE INTERACTION OF ABORT
; *WITH A DONE AND RX ERROR IN PRIMARY MODE
```

```
*****
;
; TEST 6
;
*****
```

```
2150
2151 011410 012737 000006 001226 TST6: MCV #6,@TSTNO
2152 011416 012737 011622 001216 MOV #TST7,NEXT
2153 011424 052777 000400 167760 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2154 011432 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2155 011436 052777 001000 167744 BIS #CRCEN,@PARCSR ;TURN OFF CRC
2156 011444 052777 000020 167732 BIS #RCVEN,@RXCSR ;TURN ON RECEIVER
2157 011452 052777 014000 167732 BIS #MMODE,@TXCSR ;ENTER MAINT MODE
2158 011460 004137 007060 JSR R1,RFLG ;PUSH OUT A FLAG
2159 011464 000001 1 HLT 1 ;ONE FLAG
2160 011466 005037 001324 CLR DATA ;CLEAR DATA
2161 011472 012737 000010 001326 MOV #8.,SHIFTS ;LOAD CLOCKS
2162 011500 004137 007020 JSR R1,RPOKE ;PUSH OUT DATA CHAR
2163 011504 052777 002000 167700 BIS #BITW,@TXCSR ;SET THE WINDOW
2164 011512 104412 000002 PKCLK, 2 ;PUSH OUT A BIT
2165 011516 105777 167664 TSTB @RXDBUF ;READ A CHAR
2166 011522 112737 000177 001324 MOVB #177,DATA ;LOAD A SECOND CHAR
2167 011530 012737 000010 001326 MOV #8.,SHIFTS ;LOAD CLOCKS
2168 011536 004137 007020 JSR R1,RPOKE ;PUSH OUT THE DATA
2169 011542 017737 167636 001236 MOV @RXCSR,TEMP1 ;READ THE CSR
2170 011550 017700 167632 MOV @RXDBUF,R0 ;SAVE THE BUFFER
2171 011554 032700 002000 BIT #RABORT,R0 ;TEST ABORT BIT
2172 011560 001001 BNE 1$ ;BR IF SET
2173 011562 104005 HLT 5 ;ABORT BIT FAILED TO SET
2174 011564 032737 004000 001236 1$: BIT #REACT,TEMP1 ;TEST REC ACT
2175 011572 001401 BEQ 2$ ;BR IF CLEAR
2176 011574 104006 HLT 6 ;REC ACT. FAILED TO CLEAR
2177 011576 032737 000200 001236 2$: BIT #RXDONE,TEMP1 ;TEST DONE
2178 011604 001001 BNE 3$ ;BR IF SET
2179 011606 104002 HLT 2 ;DONE FAILED TO SET
2180 011610 032700 100000 3$: BIT #RXDERR,R0 ;TEST ERROR BIT
2181 011614 001001 BNE 12$ ;BR IF SET
2182 011616 104007 HLT 7 ;ERROR BIT FAILED TO SET AFTER ABORT
2183 011620 104400 12$: SCOPE ;SCOPE THIS TEST
```

```
***** TEST 7 *****
; *TEST TO PROVE THE INTERACTION OF ABORT IN SECONDARY MODE
; *TEST FOR ABORT AT THE WRONG SECONDARY ADDRESS, THEN TEST
```

```
2184
2185
2186
```

TEST OF ABORT,DONE,AND RXERR IN SEC MODE

```

2187                                     : *AT THE CORRECT SECONDARY ADDRESS
2188                                     : .....
2189                                     : .....
2190                                     : .....
2191                                     : TEST 7
2192                                     : .....
2193                                     : .....
2194                                     : .....
2195 011622 012737 000007 001226 TST7: MOV #7,@TSTNO
2196 011630 012737 012160 001216 MOV #TST10,NEXT
2197 011636 052777 000400 167546 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2198 011644 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2199 011650 052777 001000 167532 BIS #CRCEN,@PARCSR ;TURN OFF CRC
2200 011656 052777 000020 167520 BIS #RCVEN,@RXCSR ;TURN ON RECEIVER
2201 011664 052777 010000 167516 BIS #PRISEC,@PARCSR ;ENTER SECONDARY MODE
2202 011672 052777 014000 167512 BIS #MMODE,@TXCSR ;ENTER MAINT MODE
2203 011700 004137 007060 JSR R1,RFLG ;PUSH OUT A FLAG
2204 011704 000001 1 MOV 1 ;ONE FLAG
2205 011706 012737 000001 001324 MOV #1,DATA ;LOAD WRONG ADRS
2206 011714 012737 000010 001326 MOV #8.,SHIFTS ;LOAD CLOCKS
2207 011722 004137 007020 JSR R1,RPOKE ;PUSH OUT DATA CHAR
2208 011726 052777 002000 167456 BIS #BITW,@TXCSR ;SET THE WINDOW
2209 011734 104412 000002 PKCLK, 2 ;PUSH OUT A BIT
2210 011740 105777 167442 TSTB @RXDBUF ;READ A CHAR
2211 011744 112737 000177 001324 MOVB #177,DATA ;LOAD A SECOND CHAR
2212 011752 012737 000010 001326 MOV #8.,SHIFTS ;LOAD CLOCKS
2213 011760 004137 007020 JSR R1,RPOKE ;PUSH OUT THE DATA
2214 011764 032777 002000 167414 BIT #RABORT,@RXDBUF ;TEST FOR ABORT
2215 011772 001401 BEQ 5$ ;BR IF NOT SET
2216 011774 104011 HLT 11 ;ABORT IS SET AND SHOULDN'T
2217 ;BE - THIS IS A SECONDARY STATION
2218 011776 004137 007060 5$: JSR R1,RFLG ;PUSH OUT A FLAG
2219 012002 000001 1 ;ONE FLAG
2220 012004 005037 001324 CLR DATA ;LOAD ADRS
2221 012010 012737 000010 001326 MOV #8.,SHIFTS ;LOAD CLOCKS
2222 012016 004137 007020 JSR R1,RPOKE ;PUSH OUT THE ADRS
2223 012022 012737 000252 001324 MOV #252,DATA ;LOAD A CHAR
2224 012030 012737 000010 001326 MOV #8.,SHIFTS ;LOAD CLOCKS
2225 012036 004137 007020 JSR R1,RPOKE ;PUSH OUT THE CHAR
2226 012042 052777 002000 167342 BIS #BITW,@TXCSR ;SET THE WINDOW
2227 012050 104412 000002 PKCLK, 2 ;PUSH OUT A BIT
2228 012054 105777 167326 TSTB @RXDBUF ;READ THE CHAR
2229 012060 112737 000177 001324 MOVB #177,DATA ;LOAD THE ABORT
2230 012066 012737 000010 001326 MOV #8.,SHIFTS ;LOAD CLOCKS
2231 012074 004137 007020 JSR R1,RPOKE ;PUSH OUT THE ABORT
2232 012100 017737 167300 001240 MOV @RXCSR,TEMP2 ;READ THE CSR
2233 012106 017700 167274 MOV @RXDBUF,RO ;SAVE THE BUFFER
2234 012112 032700 002000 BIT #RABORT,RO ;TEST THE ABORT BIT
2235 012116 001001 BNE 6$ ;BR IF SET
2236 012120 104005 HLT 5 ;BIT IS CLEAR - SHOULD BE SET
2237 012122 032737 004000 0012'0 6$: BIT #REACT,TEMP2 ;TEST ACTIVE
2238 012130 001401 BEQ 7$ ;BR IF CLEAR
2239 012132 104006 HLT 6 ;BIT FAILED TO CLEAR AFTER ABORT
2240 012134 032737 000200 001,40 7$: BIT #RXDONE,TEMP2 ;TEST DONE
2241 012142 001001 BNE 10$ ;BR IF SET
2242 012144 104002 HLT 2 ;DONE FAILED TO SET

```

TEST OF ABORT,DONE,AND RXERR IN SEC MODE

2243	012146	032700	100000	10\$:	BIT	#RXDERR,RO	:	TEST FOR ERROR
2244	012152	001001			BNE	12\$:	BR IF SET
2245	012154	104007			HLT	7	:	ERROR WASN'T SET AFTER ABORT
2246								
2247	012156	104400		12\$:	SCOPE		:	SCOPE THIS TEST
2248							:	***** TEST 10 *****
2249							:	*TEST EFFECT OF REC ENABLE ON ABORT ;:♦♦C
2250							:	*RXACTIVE, RXDONE, RXERROR.
2251							:	*****
2252							:	*****
2253							:	*****
2254							:	*****
2255							:	TEST 10
2256							:	*****
2257							:	*****
2258							:	*****
2259	012160	012737	000010	001226	TST10:	MOV	#10,@TSTNO	
2260	012166	012737	012422	001216		MOV	#TST11,NEXT	
2261	012174	052777	000400	167210		BIS	#MRESET,@TXCSR	: RESET THE DEVICE
2262	012202	004737	005044			JSR	PC,SMALL	: WAIT FOR RESET TO FINISH
2263	012206	052777	001000	167174		BIS	#CRCEN,@PARCSR	: TURN OFF CRC
2264	012214	052777	000020	167162		BIS	#RCVEN,@RXCSR	: TURN ON RECEIVER
2265	012222	052777	014000	167162		BIS	#MMODE,@TXCSR	: ENTER MAINT MODE
2266	012230	004137	007060			JSR	R1,RFLG	: PUSH OUT A FLAG
2267	012234	000001				1		: ONE FLAG
2268	012236	005037	001324			CLR	DATA	: CLEAR DATA
2269	012242	012737	000010	001326		MOV	#8.,SHIFTS	: LOAD CLOCKS
2270	012250	004137	007020			JSR	R1,RPOKE	: PUSH OUT DATA CHAR
2271	012254	052777	002000	167130		BIS	#BITW,@TXCSR	: SET THE WINDOW
2272	012262	104412	000002			PKCLK,	2	: PUSH OUT A BIT
2273	012266	105777	167114			TSTB	@RXDBUF	: READ A CHAR
2274	012272	112737	000177	001324		MOVB	#177,DATA	: LOAD A SECOND CHAR
2275	012300	012737	000010	001326		MOV	#8.,SHIFTS	: LOAD CLOCKS
2276	012306	004137	007020			JSR	R1,RPOKE	: PUSH OUT THE DATA
2277	012312	017737	167066	001236		MOV	@RXCSR,TEMP1	: READ THE CSR
2278	012320	017700	167062			MOV	@RXDBUF,RO	: SAVE THE BUFFER
2279	012324	032700	002000			BIT	#RABORT,RO	: TEST ABORT BIT
2280	012330	001001				BNE	20\$: BR IF SET
2281	012332	104005				HLT	5	: ABORT BIT FAILED TO SET
2282	012334	042777	000020	167042	20\$:	BIC	#RCVEN,@RXCSR	: CLEAR REC ENABLE
2283	012342	017737	167036	001236		MOV	@RXCSR,TEMP1	: SAVE STATUS
2284	012350	017700	167032			MOV	@RXDBUF,RO	: SAVE REC BUFFER
2285	012354	032700	002000			BIT	#RABORT,RO	: ABORT CLEAR?
2286	012360	001401				BEQ	1\$: IF YES,BR
2287	012362	104030				HLT	30	: ABORT FAILED TO CLEAR
2288	012364	032737	004000	001236	1\$:	BIT	#REACT,TEMP1	: TEST REC ACT
2289	012372	001401				BEQ	2\$: BR IF CLEAR
2290	012374	104006				HLT	6	: REC ACT. FAILED TO CLEAR
2291	012376	032737	000200	001236	2\$:	BIT	#RXDONE,TEMP1	: TEST DONE
2292	012404	001401				BEQ	3\$: BR IF CLEAR
2293	012406	104031				HLT	31	: DONE FAILED TO CLEAR
2294	012410	032700	100000		3\$:	BIT	#RXDERR,RO	: TEST ERROR BIT
2295	012414	001401				BEQ	12\$: BR IF CLEAR
2296	012416	104032				HLT	32	: ERROR BIT FAILED TO CLEAR AFTER RCVEN
2297								: CLEARED.
2298								

TEST OF RCVEN, ABORT, RXACTIVE, RXDONE, RXERROR

2299 012420 104400

12\$: SCOPE
:***** TEST 11 *****
:*TEST THE EFFECTS OF READING THE RXDBUF ;:++C
:*ON THE RXDONE, RXERROR, AND ABORT

2300

2301

2302

2303

2304

2305

2306

2307

2308

2309

2310

2311 012422 012737 000011 001226

2312 012430 012737 012650 001216

2313 012436 052777 000400 166746

2314 012444 004737 005044

2315 012450 052777 001000 166732

2316 012456 052777 000020 166720

2317 012464 052777 014000 166720

2318 012472 004137 007060

2319 012476 000001

2320 012500 005037 001324

2321 012504 012737 000010 001326

2322 012512 004137 007020

2323 012516 052777 002000 166666

2324 012524 104412 000002

2325 012530 105777 166652

2326 012534 112737 000177 001324

2327 012542 012737 000010 001326

2328 012550 004137 007020

2329 012554 017737 166624 001236

2330 012562 017700 166620

2331 012566 032700 002000

2332 012572 001001

2333 012574 104005

2334 012576 005777 166604

2335 012602 017737 166576 001236

2336 012610 017700 166572

2337 012614 032737 000200 001236

2338 012622 001401

2339 012624 104031

2340 012626 032700 002000

2341 012632 001401

2342 012634 104030

2343 012636 032700 100000

2344 012642 001401

2345 012644 104032

2346

2347 012646 104400

2348

2349

2350

2351

2352

2353

2354

```
TST11: MOV #11,@TSTNO
MOV #TST12,NEXT
BIS #MRESET,@TXCSR ;RESET THE DEVICE
JSR PC,SMALL ;WAIT FOR RESET TO FINISH
BIS #RCEN,@PARCSR ;TURN OFF CRC
BIS #RCVEN,@RXCSR ;TURN ON RECEIVER
BIS #MMODE,@TXCSR ;ENTER MAINT MODE
JSR R1,RFLG ;PUSH OUT A FLAG
1 ;ONE FLAG
CLR DATA ;CLEAR DATA
MOV #8.,SHIFTS ;LOAD CLOCKS
JSR R1,RPOKE ;PUSH OUT DATA CHAR
BIS #BITW,@TXCSR ;SET THE WINDOW
PKCLK, 2 ;PUSH OUT A BIT
TSTB @RXDBUF ;READ A CHAR
MOVB #177,DATA ;LOAD A SECOND CHAR
MOV #8.,SHIFTS ;LOAD CLOCKS
JSR R1,RPOKE ;PUSH OUT THE DATA
MOV @RXCSR,TEMP1 ;READ THE CSR
MOV @RXDBUF,RO ;SAVE THE BUFFER
BIT #RABORT,RO ;TEST ABORT BIT
BNE 20$ ;BR IF SET
HLT 5 ;ABORT BIT FAILED TO SET
20$: TST @RXDBUF ;READ RXDBUF
MOV @RXCSR,TEMP1 ;SAVE STATUS
MOV @RXDBUF,RO ;SAVE REC BUFFER
1$: BIT #RXDONE,TEMP1 ;TEST DONE
BEQ 2$ ;BR IF CLEAR
HLT 31 ;DONE FAILED TO CLEAR
2$: BIT #RABORT,RO ;ABORT CLEAR?
BEQ 3$ ;IF YES,BR
HLT 30 ;ABORT FAILED TO CLEAR
3$: BIT #RXDERR,RO ;TEST ERROR BIT
BEQ 12$ ;BR IF CLEAR
HLT 32 ;ERROR BIT FAILED TO CLEAR AFTER RXDBUF
;READ.
```

12\$: SCOPE
:***** TEST 12 *****
:*DATA OVERRUN TEST IN PRIMARY MODE. TEST TO
:*PROVE OVERRUN ERROR AND RX ERROR WILL OCCUR
:*****

:*****
:

```
2355          : TEST 12
2356          :
2357          :*****
2358          :*****
2359 012650 012737 000012 001226 TST12: MOV #12,@TSTNO
2360 012656 012737 013022 001216      MOV #TST13,NEXT
2361 012664 052777 000400 166520      BIS #MRESET,@TXCSR ;RESET THE DEVICE
2362 012672 004737 005044          JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2363 012676 013703 001404          MOV RXCSR,R3 ;LOAD THE CONTROL REGISTER
2364 012702 052777 001000 166500      BIS #CRCEN,@PARCSR ;TURN OFF CRC
2365 012710 052713 000020          BIS #RCVEN,(R3) ;TURN ON THE RECEIVER
2366 012714 052777 014000 166470      BIS #MMODE,@TXCSR ;ENTER MAINTENANCE MODE
2367 012722 004137 007060          JSR R1,RFLG ;PUSH OUT A FLAG
2368 012726 000001          1 ;ONE FLAG
2369 012730 012737 000252 001324      MOV #252,DATA ;LOAD CHAR TO BE OUTPUT
2370 012736 012737 000010 001326      MOV #8.,SHIFTS ;CLOCK SETUP
2371 012744 004137 007020          JSR R1,RPOKE ;PUSH DATA INTO RECEIVER
2372 012750 012737 000070 001324      MOV #70,DATA ;LOAD A SECOND CHARACTER
2373 012756 012737 000011 001326      MOV #9.,SHIFTS ;SETUP #OF CLOCKS
2374 012764 004137 007020          JSR R1,RPOKE ;PUSH THE SECOND CHARACTER INTO RX
2375 012770 017700 166412          MOV @RXDBUF,R0 ;SAVE BUFFER
2376 012774 042700 037777          LIC #^C<RXDERR!OVRUN>,R0 ;CLEAR JUNK
2377 013000 022700 140000          CMP #RXDERR!OVRUN,R0 ;CHECK TO SEE IF BOTH ARE SET
2378 013004 001401          BEQ 2$ ;BR IF OK
2379 013006 104013          HLT 13 ;THEY DIDN'T MATCH
2380
2381 013010 032713 004000          2$: BIT #REACT,(R3) ;TEST REC. ACT
2382 013014 001001          BNE 3$ ;BR IF STILL SET
2383 013016 104014          HLT 14 ;REC. ACTIVE CLEARED AND SHOULD BE SET
2384
2385 013020 104400          3$: SCOPE ;SCOPE THIS TEST
2386
2387          :***** TEST 13 *****
2388          :*DATA OVERRUN TEST IN SECONDARY MODE. TEST TO PROVE
2389          :*THAT OVERRUN DOES NOT OCCUR IF THIS STATION IS NOT
2390          :*SELECTED. THEN SELECT THIS SECONDARY STATION AND
2391          :*PROVE OVERRUN ERROR AND RX ERROR WILL OCCUR
2392          :*****
2393
2394          :*****
2395          :
2396          : TEST 13
2397          :
2398          :*****
2399          :*****
2400 013022 012737 000013 001226 TST13: MOV #13,@TSTNO
2401 013030 012737 013260 001216      MOV #TST14,NEXT
2402 013036 052777 000400 166346      BIS #MRESET,@TXCSR ;RESET THE DEVICE
2403 013044 004737 005044          JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2404 013050 013703 001404          MOV RXCSR,R3 ;LOAD THE CONTROL REGISTER
2405 013054 052777 001000 166326      BIS #CRCEN,@PARCSR ;TURN OFF CRC
2406 013062 052713 000020          BIS #RCVEN,(R3) ;TURN ON THE RECEIVER
2407 013066 052777 010000 166314      BIS #PRISEC,@PARCSR ;ENTER SECONDARY MODE
2408 013074 052777 014000 166310      BIS #MMODE,@TXCSR ;ENTER MAINTENANCE MODE
2409 013102 004137 007060          JSR R1,RFLG ;PUSH OUT A FLAG
2410 013106 000001          1 ;ONE FLAG
```

DATA OVERRUN TEST IN SECONDARY MODE

```

2411 013110 012737 000252 001324      MOV      #252,DATA      ;LOAD AN INCORRECT SEC. STA ADRS
2412 013116 012737 000031 001326      MOV      #25.,SHIFTS   ;PUSH OUT 3 CHARS
2413 013124 004137 007020              JSR      R1,RPOKE      ;THRU THE RECEIVER
2414 013130 032777 040000 166250      BIT      #OVRRUN,@RXDBUF ;TEST FOR OVERRUN
2415 013136 001401              BEQ      4$           ;BR IF NOT SET
2416
2417 013140 104015              HLT      15           ;OVERRUN IS SET AND SHOULDN'T BE
2418
2419
2420 013142 004137 007060      4$:     JSR      R1,RFLG      ;PUSH OUT A FLAG
2421 013146 000001              1           ;ONE FLAG
2422 013150 005037 001324      CLR      DATA        ;GET CORRECT SEC. STATION ADRS
2423 013154 012737 000010 001326      MOV      #8.,SHIFTS   ;MOVE THE #OF CLOCKS TO PUSH
2424 013162 004137 007020              JSR      R1,RPOKE      ;PUSH OUT THE SEC. ADRS
2425 013166 012737 000252 001324      MOV      #252,DATA    ;PUSH OUT
2426 013174 012737 000010 001326      MOV      #8.,SHIFTS   ;TWO
2427 013202 004137 007020              JSR      R1,RPOKE      ;DATA
2428 013206 012737 000070 001324      MOV      #70,DATA     ;CHARACTERS
2429 013214 012737 000011 001326      MOV      #9.,SHIFTS   ;TO FORCE
2430 013222 004137 007020              JSR      R1,RPOKE      ;AN OVERRUN ERROR
2431 013226 017700 166154      MOV      @RXDBUF,R0   ;SAVE
2432 013232 042700 037777      BIC      #^C<RXDERR!OVRRUN>,R0 ;CLEAR JUNK
2433 013236 022700 140000      CMP      #RXDERR!OVRRUN,R0 ;ARE THE ERROR BITS THERE
2434 013242 001401              BEQ      6$           ;BR IF YES
2435 013244 104013              HLT      13           ;ERROR BITS MISSING
2436
2437 013246 032713 004000      6$:     BIT      #REACT,(R3) ;TEST ACTIVE
2438 013252 001001              BNE      3$           ;BR IF SET
2439 013254 104005              HLT      5           ;ACTIVE DROPPED OUT
2440 013256 104400      3$:     SCOPE          ;SCOPE THIS TEST
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452 013260 012737 000014 001226      TST14: MOV      #14,@TSTNO
2453 013266 012737 013130 001216      MOV      #TST15,NEXT
2454 013274 052777 000400 166110      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
2455 013302 004737 005044              JSR      PC,SMALL     ;WAIT FOR RESET TO FINISH
2456 013306 012703 013516      MOV      #TBLA,R3     ;LOAD THE TABLE POINTER
2457 013312 012704 013516      MOV      #TBLA,R4     ;DITTO
2458 013316 012702 000004              MOV      #4,R2        ;LOAD THE # OF CHARS TO DO
2459 013322 052777 001000 166060      BIS      #CRCEN,@PARCSR
2460 013330 005077 166060      CLR      @TXDBUF      ;RESET TXDONE
2461 013334 052777 014000 166050      BIS      #MMODE,@TXCSR ;ENTER M/MODE
2462 013342 052777 000020 166034      BIS      #RCVEN,@RXCSR ;TURN ON THE RECEIVER
2463 013350 052777 000020 166034      BIS      #SEND,@TXCSR ;TURN ON THE TRANSMITTER
2464 013356 052777 000400 166030      BIS      #TSOM,@TXDBUF ;START A FLAG
2465 013364 104412 000010              PKCLK      ,8         ;START A FLAG
2466 013370 012377 166020      MOV      (R3)+,@TXDBUF ;LOAD FIRST DATA CHAR

```

***** TEST 14 *****
 *TEST OF SPECIFIC DATA PATTERNS THRU
 *THE RECEIVER IN PRIMARY MODE

```

:*****
:
: TEST 14
:
:*****
:*****

```

RECEIVER DATA TEST WITHOUT CRC

```

2467 013374 005302          DEC R2          ;LOWER THE # TO DO
2468 013376 104412 000020  PKCLK ,16.     ;PUSH OUT
2469 013402 012377 166006  1$: MOV (R3)+,@TXDBUF ;LOAD DATA
2470 013406 005302          DEC R2          ;LOWER THE # TO DO
2471 013410 104412 000020  PKCLK ,16.     ;PUSH OUT DATA
2472 013414 105777 165764  TSTB @RXCSR    ;CHECK FOR RX DONE
2473 013420 100401          BMI 2$         ;BR IF DONE SET
2474 013422 104000          HLT             ;RX DONE FAILED TO SET
2475 013424 017737 165756 001324 2$: MOV @RXDBUF,DATA ;GET THE BUFFER
2476 013432 121437 001324  CMPB (R4),DATA ;CHECK IT
2477 013436 001401          BEQ 3$         ;BR IF OK
2478 013440 104000          HLT             ;DATA COMPARE ERROR
2479 013442 005724          3$: TST (R4)+
2480 013444 005702          TST R2         ;CHECK FOR FINISH
2481 013446 001355          BNE 1$         ;BR IF MORE TO GO
2482 013450 104412 000020  PKCLK ,16.     ;PUSH OUT LAST CHAR
2483 013454 105777 165724  TSTB @RXCSR    ;CHECK FOR DONE
2484 013460 100401          BMI 4$         ;BR IF DONE SET
2485 013462 104000          HLT             ;DONE FAILED TO SET
2486 013464 017737 165716 001324 4$: MOV @RXDBUF,DATA ;GET THE BUFFER
2487 013472 121437 001324  CMPB (R4),DATA ;CHECK IT
2488 013476 001401          BEQ 5$         ;BR IF A MATCH
2489 013500 104000          HLT             ;DATA COMPARE ERROR
2490 013502
2491 013502 052777 000400 165702 5$: BIS #MRESET,@TXCSR ;RESET THE DEVICE
2492 013510 004737 005044  JSR PC,SMALL  ;WAIT FOR RESET TO FINISH
2493 013514 104400          SCOPE

```

```

;DATA TABLE
TBLA: .WORD 125
      .WORD 252
      .WORD 000
      .WORD 377
      .WORD 000

```

```

***** TEST 15 *****
; *TEST TO PROVE THAT THE DEVICE WILL
; *WORK WITH ALL POSSIBLE SECONDARY
; *STATION ADDRESSES.
*****

```

```

;*****
; TEST 15
;*****

```

```

2514 013530 012737 000015 001226 TST15: MOV #15,@TSTNO
2515 013536 012737 014024 001216  MOV #TST16,NEXT
2516 013544 012737 013572 001220  MOV #20$,LOCK ;SW09 SETUP
2517 013552 005005          CLR R5         ;CLEAR SEC ADRS HOLD
2518 013554 005037 001236          CLR TEMP1      ;CLEAR TEMP STORAGE
2519 013560 005037 001240          CLR TEMP2      ;DITTO
2520 013564 012737 011000 001236  MOV #CRCEN,PRISEC,TEMP1 ;LOAD MODE AND NO CRC
2521 013572
2522 013572 052777 000400 165612 20$: BIS #MRESET,@TXCSR ;RESET THE DEVICE

```

```

2523 013600 004737 005044      JSR      PC,SMALL      ;WAIT FOR RESET TO FINISH
2524 013604 013777 001236 165576  MOV      TEMP1,@PARCSR ;LOAD MODE AND NO CRC AND SEC STATION
2525 013612 052777 000620 165564 1$:  BIS      #RCVEN,@RXCSR ;TURN ON RECEIVER
2526 013620 052777 004020 165564  BIS      #SYSTST!SEND,@TXCSR ;TURN ON TRANSMITTER AND CLOCK
2527 013626 105777 165560      2$:  TSTB    @TXCSR      ;WAIT FOR
2528 013632 100375      BPL      2$          ;DONE
2529 013634 012777 000400 165552  MOV      #TSOM,@TXDBUF ;LOAD START OF MSG
2530 013642 105777 165544      3$:  TSTB    @TXCSR      ;WAIT FOR
2531 013646 100375      BPL      3$          ;DONE AGAIN
2532 013650 013777 001240 165536  MOV      TEMP2,@TXDBUF ;LOAD SEC STATION ADRS AND
2533 013656 105777 165530      4$:  TSTB    @TXCSR      ;WAIT FOR
2534 013662 100375      BPL      4$          ;DONE AGAIN
2535 013664 012777 000252 165522  MOV      #252,@TXDBUF ;NOW LOAD DATA
2536 013672 032777 004000 165514 64$:  BIT      #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2537 013700 001374      BNE      64$        ;BR IF SET
2538 013702 032777 004000 165504 65$:  BIT      #TIMER,@TXDBUF ;CHECK THE TIMEP BIT
2539 013710 001774      BEQ      65$        ;BR IF CLEAR
2540 013712 105777 165466      5$:  TSTB    @RXCSR      ;TEST FOR
2541 013716 100375      BPL      5$          ;RX DONE
2542 013720 005777 165460      TST     @RXCSR      ;TEST FOR ERROR
2543 013724 100001      BPL      6$          ;BR IF NO ERROR
2544 013726 104014      HLT     14          ;ERROR FOUND!
2545 013730 017704 165452      6$:  MOV     @RXDBUF,R4  ;GET THE BUFFER
2546 013734 032704 000400      BIT     #RSOM,R4    ;CHECK FOR START OF MSG
2547 013740 001001      BNE     7$          ;BR IF SET
2548 013742 104000      HLT     ;START OF MSG FAILED TO SET
2549 013744 122704 000252      7$:  CMPB   #252,R4    ;CHECK FOR DATA
2550 013750 001401      BEQ     10$         ;BR IF A MATCH
2551 013752 104012      HLT     ;FAILED TO RECEIVE DATA AS A SEC STATION
2552 013754 104401      10$:  SCOPE1 ;SW09=1?
2553 013756 052777 000400 165426  BIS     #MRESET,@TXCSR ;RESET THE DEVICE
2554 013764 004737 005044      JSR     PC,SMALL    ;WAIT FOR RESET TO FINISH
2555 013770 105205      INCB   R5          ;GET NEXT SEC ADRS
2556 013772 001413      BEQ     11$        ;BR IF ALL DONE
2557 013774 110537 001236  MOVB   R5,TEMP1    ;LOAD THE ADRS FOR PAR CSR
2558 014000 110537 001240  MOVB   R5,TEMP2    ;DITTO FOR TXD BUF
2559 014004 052737 011000 001236  BIS     #CRCEN!PRISEC,TEMP1 ;LOAD FOR PAR CSR
2560 014012 013777 001236 165370  MOV     TEMP1,@PARCSR ;DO IT
2561 014020 000674      BR     1$          ;BR TO DO OVER
2562 014022 104400      11$:  SCOPE ;SCOPE THIS TEST
  
```

```

:***** TEST 16 *****
:*TEST OF SPECIFIC CHARACTER DATA PATTERNS
:*USING BCC CHECK IN PRIMARY MODE.
:*****
  
```

```

:*****
: TEST 16
:*****
  
```

```

2575 014024 012737 000016 001226 TST16: MOV     #16,@TSTNO
2576 014032 012737 014542 001216  MOV     #TST17,NEXT
2577 014040 012704 014350      MOV     #TBLB,R4    ;GET THE TABLE POINTER
2578 014044 012703 000004      MOV     #4,R3       ;GET THE # TO DO
  
```


RECEIVER DATA TEST WITH CRC

2635
2636 014350 000252
2637 014352 000000
2638 014354 000125
2639 014356 000377
2640 014360 000000
2641
2642
2643 014362 010046
2644 014364 010146
2645 014366 010246
2646 014370 012537 001236
2647 014374 012537 001240
2648 014400 012537 001242
2649 014404 005037 014536
2650 014410 013700 001242
2651 014414 006037 001240
2652 014420 005500
2653 014422 032700 000001
2654 014426 001402
2655 014430 005137 014536
2656 014434 013700 014534
2657 014440 005100
2658 014442 040037 014536
2659 014446 000241
2660 014450 006037 001242
2661 014454 013700 014536
2662 014460 013701 001242
2663 014464 010102
2664 014466 040100
2665 014470 043702 014536
2666 014474 050200
2667 014476 043737 014534 001242
2668 014504 050037 001242
2669 014510 005337 001236
2670 014514 001333
2671 014516 013737 001242 014540
2672 014524 012602
2673 014526 012601
2674 014530 012600
2675 014532 000205
2676 014534 000000
2677 014536 000000
2678 014540 000000
2679 120001
2680 102010
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690

:DATA TABLE
TBLB: .WORD 252
.WORD 0
.WORD 125
.WORD 377
.WORD 0

SIMBCC: MOV R0,-(SP)
MOV R1,-(SP)
MOV R2,-(SP)
MOV (R5)+,TEMP1
MOV (R5)+,TEMP2
MOV (R5)+,TEMP3
1\$: CLR BCCFBK
MOV TEMP3,R0
ROR TEMP2
ADC R0
BIT #BIT0,R0
BEQ 2\$
COM BCCFBK
2\$: MOV XPOLY,R0
COM R0
BIC R0,BCCFBK
CLC
ROR TEMP3
MOV BCCFBK,R0
MOV TEMP3,R1
MOV R1,R2
BIC R1,R0
BIC BCCFBK,R2
BIS R2,R0
BIC XPOLY,TEMP3
BIS R0,TEMP3
DEC TEMP1
BNE 1\$
MOV TEMP3,CALBCC
MOV (SP)+,R2
MOV (SP)+,R1
MOV (SP)+,R0
RTS R5

XPOLY: 0
BCCFBK: 0
CALBCC: 0
CRC16=120001
CRC.CCITT=102010

:***** TEST 17 *****
:THIS TEST PROVES THAT THE DEVICE WILL
:OUTPUT SIXTEEN SPACES IN FRONT OF A
:FLAG IF STARTING FROM THE IDLE STATE.
:*****

:*****
:

SPECIAL ZEROES SEQUENCE TEST

```
2691      : TEST 17
2692      : .....
2693      : .....
2694      : .....
2695 014542 012737 000017 001226 TST17: MOV #17,@TSTNO
2696 014550 012737 015050 001216      MOV #TST20,NEXT
2697 014556 052777 000400 164626      BIS #MRESET,@TXCSR ;RESET THE DEVICE
2698 014564 004737 005044      JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2699 014570 005037 001236      CLR TEMP1 ;CLEAR FOR SOFTWARE STORAGE
2700 014574 005037 001240      CLR TEMP2
2701 014600 005037 001242      CLR TEMP3
2702 014604 005077 164604      CLR @TXDBUF ;RESET TXDONE
2703 014610 052777 014000 164574      BIS #MMODE,@TXCSR ;ENTER MAINT MODE
2704 014616 052777 001000 164564      BIS #CRCEN,@PARCSR ;TURN OFF CRC
2705 014624 052777 000020 164560      BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
2706 014632 052777 000020 164544      BIS #RCVEN,@RXCSR ;TURN ON RECEIVER
2707 014640 052777 001400 164546      BIS #TEOM!TSOM,@TXDBUF ;START SPECIAL SEQUENCE
2708 014646 104412 000002      PKCLK ,2 ;SYNC UP DUP
2709 014652 005237 001236      1$: INC TEMP1 ;UPDATE COUNT
2710 014656 022737 000021 001236      CMP #17.,TEMP1 ;CHECK FOR FINISH
2711 014664 001407      BEQ 2$ ;BR IF YES
2712 014666 104412 000002      PKCLK ,2 ;POKE CLOCK
2713 014672 032777 040000 164512      BIT #MTDATA,@TXCSR ;CHECK THE DATA
2714 014700 001764      BEQ 1$ ;BR IF OK
2715 014702 104000      HLT ;DATA SHOULD BE A 0, WAS A 1
2716 014704 104412 000002 2$: PKCLK ,2 ;POKE CLOCK
2717 014710 032777 040000 164474      BIT #MTDATA,@TXCSR ;CHECK THE DATA
2718 014716 001401      BEQ 3$ ;BR IF OK
2719 014720 104000      HLT ;DATA SHOULD BE 0, WAS A ONE
2720 014722 012777 000125 164464 3$: MOV #125,@TXDBUF ;LOAD DATA CHAR
2721 014730 104412 000002 4$: PKCLK ,2 ;POKE CLOCK
2722 014734 032777 040000 164450      BIT #MTDATA,@TXCSR ;CHECK THE BIT
2723 014742 001001      BNE 5$ ;BR IF OK
2724 014744 104000      HLT ;DATA SHOULD BE 1, WAS 0
2725 014746 005237 001240 5$: INC TEMP2 ;UPDATE FOR FINISH
2726 014752 022737 000006 001240      CMP #6,TEMP2 ;CHECK FOR DONE
2727 014760 001363      BNE 4$ ;BR IF MORE TO GO
2728 014762 104412 000002      PKCLK ,2 ;POKE CLOCK
2729 014766 032777 040000 164416      BIT #MTDATA,@TXCSR ;CHECK THE DATA
2730 014774 001401      BEQ 6$ ;BR IF OK
2731 014776 104000      HLT ;DATA SHOULD BE 0, WAS A 1
2732 015000 104412 000006 6$: PKCLK ,6 ;START OUT DATA CHAR
2733 015004 052777 001000 164400      BIS #TEOM,@TXCSR ;TURN OFF TRANSMITTER
2734 015012 104412 000016      PKCLK ,14 ;FINISH
2735 015016 105777 164362      TSTB @RXCSR ;CHECK RECEIVER
2736 015022 100401      BMI 7$ ;BR IF OK
2737 015024 104000      HLT ;RECEIVER FAILED TO ACCEPT SPECIAL CHAR
2738 015026 117737 164354 001242 7$: MOVB @RXDBUF,TEMP3 ;GET THE CHAR
2739 015034 122737 000125 001242      CMPB #125,TEMP3 ;CHECK IT
2740 015042 001401      BEQ 10$
2741 015044 104000      HLT ;DATA FAILED TO MATCH AFTER
2742      ;SPECIAL SPACES CHARACTER
2743 015046 104400 10$: SCOPE ;SCOPE THIS TEST
2744
2745
2746      : ..... TEST 20 .....
```

MODEM CONTROL BITS DTR,RING,AND DSP INTERACTION TEST

```
2747 ;*THIS TEST PROVES THE INTERACTION OF DTR
2748 ;*WITH RING,DSR
2749 ;*AND DATA SET CHANGE ONE AND DATA SET CHANGE TWO.
2750 ;*SET THE BIT AND VERIFY THE OTHER BITS ARE SET. CLEAR
2751 ;*THE BIT AND VERIFY CLEAR. REPEAT FOR MRESET.
2752 ;*****
2753
2754 ;*****
2755 ; TEST 20
2756 ;*****
2757 ;*****
2758 ;*****
2759 ;*****
2760 015050 012737 000020 001226 TST20: MOV #20,@TSTNO
2761 015056 012737 015516 001216 MOV #TST21,NEXT
2762 015064 012737 015144 001220 MOV #1$,LOCK
2763 015072 105737 001322 TSTB TCNFLG
2764 015076 001002 BNE .+6
2765 015100 000137 015504 10$: JMP 6$
2766 015104 005077 164274 CLR @RXCSR ;CLEAR THE REGISTER
2767 015110 004137 007332 JSR R1,OJUMPER ;THIS CALL DETERMINES IF TURNAROUND CONNECTOR
2768 015114 015506 7$ ;AND OPTIONAL JUMPER ARE USED
2769 ;AND LOADS R5 (EXPECTED) ACCORDINGLY.
2770 015116 052777 000400 164266 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2771 015124 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2772 015130 013703 001404 MOV RXCSR,R3 ;LOAD THE RECEIVER CONTROL REGISTER TO R3.
2773 015134 005013 CLR (R3) ;CLEAR OUT EXTRA BITS
2774 015136 052777 010000 164246 BIS #MEXT,@TXCSR ;ENTER EXTERNAL MAINT. MODE
2775 015144 052713 000002 1$: BIS #DTR,(R3) ;TURN ON DTR
2776 015150 012737 000110 015200 MOV #110,68$ ;LOAD THE NUMBER
2777 015156 032777 004000 164230 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2778 015164 001374 BNE 66$ ;BR IF SET
2779 015166 032777 004000 164220 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2780 015174 001774 BEQ 67$ ;BR IF CLEAR
2781 015176 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2782 015200 000110 68$: 110 ;OF TIMES TO REPEAT
2783 015202 001365 BNE 66$ ;BR IF MORE TO GO
2784 015204 011304 MOV (R3),R4 ;GET THE BITS FROM THE RXCSR
2785 015206 020504 CMP R5,R4 ;R5=GOOD R4=?
2786 015210 001423 BEQ 2$ ;BRANCH IF THEY MATCH
2787 015212 104003 HLT 3 ;NO MATCH - SHOW OPR.
2788 015214 104401 SCOP1
2789 015216 012737 000005 015246 MOV #5,73$ ;LOAD THE NUMBER
2790 015224 032777 004000 164162 71$: BIT #1IMER,@TXDBUF ;CHECK THE TIMER BIT
2791 015232 001374 BNE 71$ ;BR IF SET
2792 015234 032777 004000 164152 72$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2793 015242 001774 BEQ 72$ ;BR IF CLEAR
2794 015244 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2795 015246 000005 73$: 5 ;OF TIMES TO REPEAT
2796 015250 001365 BNE 71$ ;BR IF MORE TO GO
2797 015252 032713 040000 BIT #RING,(R3)
2798 015256 001000 BNE 2$
2799 015260 012737 015272 001220 2$: MOV #3$,LOCK ;SW09 SETUP
2800 015266 042705 041002 BIC #RING.DSR!DTR,R5 ;CLEAR OUT UNWANTED BITS
2801 015272 005013 3$: CLR (R3) ;CLEAR OUT THE REGISTER
2802 015274 012737 000005 015324 MOV #5,78$ ;LOAD THE NUMBER
```

MODEM CONTROL BITS DTR,RING,AND DSR INTERACTION TEST

```
2803 015302 032777 004000 164104 76$ BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2804 015310 001374 BNE 76$ ;BR IF SET
2805 015312 032777 004000 164074 77$ BIT #TIMER,@TXDBUF ;CHECK THE BIT
2806 015320 001774 BEQ 77$ ;BR IF CLEAR
2807 015322 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2808 015324 000005 78$: 5 ;OF TIMES TO REPEAT
2809 015326 001365 BNE 76$ ;BR IF MORE TO GO
2810 015330 011304 MOV (R3),R4 ;READ BACK THE REGISTER
2811 015332 020504 CMP R5,R4 ;R5=GOOD R4=?
2812 015334 001402 BEQ 4$ ;BRANCH IF ONLY THE DSC BITS ARE SET
2813 015336 104003 HLT 3 ;NO-GO TELL OPR
2814 015340 104401 SCOPI
2815 015342 012737 015350 001220 4$: MOV #5$,LOCK ;SW09 SETUP
2816 015350 052713 000002 5$: BIS #DTR,(R3) ;TURN ON DTR
2817 015354 012737 000005 015404 MOV #5,83$ ;LOAD THE NUMBER
2818 015362 032777 004000 164024 81$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2819 015370 001374 BNE 81$ ;BR IF SET
2820 015372 032777 004000 164014 82$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2821 015400 001774 BEQ 82$ ;BR IF CLEAR
2822 015402 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2823 015404 000005 83$: 5 ;OF TIMES TO REPEAT
2824 015406 001365 BNE 81$ ;BR IF MORE TO GO
2825 015410 005005 CLR R5 ;CLEAR OUT EXPECTED
2826 015412 005013 CLR (R3) ;CLEAR OUT THE REGISTER
2827 015414 052777 000400 163770 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2828 015422 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2829 015426 052777 010000 163756 BIS #MEXT,@TXCSR ;TURN ON EXTERNAL MODE
2830 015434 012737 000005 015464 MOV #5,88$ ;LOAD THE NUMBER
2831 015442 032777 004000 163744 86$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2832 015450 001374 BNE 86$ ;BR IF SET
2833 015452 032777 004000 163734 87$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2834 015460 001774 BEQ 87$ ;BR IF CLEAR
2835 015462 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2836 015464 000005 88$: 5 ;OF TIMES TO REPEAT
2837 015466 001365 BNE 86$ ;BR IF MORE TO GO
2838 015470 005713 TST (R3) ;STRIP DSCA & DSCB FROM CSR
2839 015472 011304 MOV (R3),R4 ;GET THE REGISTER
2840 015474 020504 CMP R5,R4 ;R5=GOOD,R4=?
2841 015476 001402 BEQ 6$ ;BR IF OK
2842 015500 104003 HLT 3 ;REPORT THE ERROR
2843 015502 104401 SCOPI ;SW09=1?
2844
2845 015504 104400 6$: SCOPE ;SCOPE THE WHOLE TEST
2846 015506 141003 7$: .WORD 141003
2847 015510 141001 .WORD 141001
2848 015512 001002 .WORD 1002
2849 015514 000000 .WORD 0
```

2850
2851
2852
2853
2854
2855
2856
2857
2858

```
***** TEST 21 *****  
; *THIS TEST PROVES THE INTERACTION OF RTS  
; *WITH CTS,CARDET  
; *AND DATA SET CHANGE ONE AND DATA SET CHANGE TWO.  
; *SET THE BIT AND VERIFY THE OTHER BITS ARE SET. CLEAR  
; *THE BIT AND VERIFY CLEAR. REPEAT FOR MRESET.  
*****
```

MODEM CONTROL BITS RTS,CTS,AND CARDET INTERACTION TEST

```

2859
2860
2861
2862
2863
2864
2865
2866 015516 012737 000021 001226 TST21: MOV #21,@TSTNO
2867 015524 012737 016164 001216 MOV #TST22,NEXT
2868 015532 012737 015612 001220 MOV #1$,LOCK
2869 015540 105737 001322 TSTB TCNFLG
2870 015544 001002 BNE .+6
2871 015546 000137 016152 10$: JMP 6$
2872 015552 005077 163626 CLR @RXCSR ;CLEAR THE REGISTER
2873 015556 004137 007332 JSR R1,OJUMPER ;THIS CALL DETERMINES IF TURNAROUND CONNECTOR
2874 015562 016154 7$ ;AND OPTIONAL JUMPER ARE USED
2875 ;AND LOADS R5 (EXPECTED) ACCORDINGLY.
2876 015564 052777 000400 163620 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2877 015572 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2878 015576 013703 001404 MOV RXCSR,R3 ;LOAD THE RECEIVER CONTROL REGISTER TO R3.
2879 015602 005013 CLR (R3) ;CLEAR OUT EXTRA BITS
2880 015604 052777 010000 163600 BIS #MEXT,@TXCSR ;ENTER EXTERNAL MAINT. MODE
2881 015612 052713 000004 1$: BIS #RTS,(R3) ;TURN ON RTS
2882 015616 012737 000110 015646 MOV #110,68$ ;LOAD THE NUMBER
2883 015624 032777 004000 163562 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2884 015632 01374 BNE 66$ ;BR IF SET
2885 015634 032777 004000 163552 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2886 015642 001774 BEQ 67$ ;BR IF CLEAR
2887 015644 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2888 015646 000110 68$: 110 ;OF TIMES TO REPEAT
2889 015650 001365 BNE 66$ ;BR IF MORE TO GO
2890 015652 011304 MOV (R3),R4 ;GET THE BITS FROM THE RXCSR
2891 015654 020504 CMP R5,R4 ;R5=GOOD R4=?
2892 015656 001423 BEQ 2$ ;BRANCH IF THEY MATCH
2893 015660 104003 HLT 3 ;NO MATCH - SHOW OPR.
2894 015662 104401 SCOP1
2895 015664 012737 000005 015714 MOV #5,73$ ;LOAD THE NUMBER
2896 015672 032777 004000 163514 71$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2897 015700 001374 BNE 71$ ;BR IF SET
2898 015702 032777 004000 163504 72$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2899 015710 001774 BEQ 72$ ;BR IF CLEAR
2900 015712 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2901 015714 000005 73$: 5 ;OF TIMES TO REPEAT
2902 015716 001365 BNE 71$ ;BR IF MORE TO GO
2903 015720 032713 040000 BIT #RING,(R3)
2904 015724 001000 BNE 2$
2905 015726 012737 015740 001220 2$: MOV #3$,LOCK ;SW09 SETUP
2906 015734 042705 030004 BIC #CTS!CARDET!RTS,R5 ;CLEAR OUT UNWANTED BITS
2907 015740 005013 3$: CLR (R3) ;CLEAR OUT THE REGISTER
2908 015742 012737 000005 015772 MOV #5,78$ ;LOAD THE NUMBER
2909 015750 032777 004000 163436 76$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2910 015756 001374 BNE 76$ ;BR IF SET
2911 015760 032777 004000 163426 77$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2912 015766 001774 BEQ 77$ ;BR IF CLEAR
2913 015770 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2914 015772 000005 78$: 5 ;OF TIMES TO REPEAT

```

MODEM CONTROL BITS RTS,CTS,AND CARDET INTERACTION TEST

2915	015774	001365				BNE	76\$;BR IF MORE TO GO
2916	015776	011304				MOV	(R3),R4		;READ BACK THE REGISTER
2917	016000	020504				CMF	R5,R4		;R5=GOOD R4=?
2918	016002	001402				BEQ	4\$;BRANCH IF ONLY THE DSC BITS ARE SET
2919	016004	104003				HLT	3		;NO-GO TELL OPR
2920	016006	104401				SCOPI			
2921	016010	012737	016016	001220	4\$:	MOV	#5\$,LOCK		;SW09 SETUP
2922	016016	052713	000004		5\$:	BIS	#RTS,(R3)		;TURN ON RTS
2923	016022	012737	000005	016052		MOV	#5,83\$;LOAD THE NUMBER
2924	016030	032777	004000	163356	81\$:	BIT	#TIMER,@TXDBUF		;CHECK THE TIMER BIT
2925	016036	001374				BNE	81\$;BR IF SET
2926	016040	032777	004000	163346	82\$:	BIT	#TIMER,@TXDBUF		;CHECK THE BIT
2927	016046	001774				BEQ	82\$;BR IF CLEAR
2928	016050	005327				DEC	(PC)+		;DECREMENT THE NUMBER
2929	016052	000005			83\$:	S			;OF TIMES TO REPEAT
2930	016054	001365				BNE	81\$;BR IF MORE TO GO
2931	016056	005005				CLR	R5		;CLEAR OUT EXPECTED
2932	016060	005013				CLR	(R3)		;CLEAR OUT THE REGISTER
2933	016062	052777	000400	163322		BIS	#MRESET,@TXCSR		;RESET THE DEVICE
2934	016070	004737	005044			JSR	PC,SMALL		;WAIT FOR RESET TO FINISH
2935	016074	052777	010000	163310		BIS	#MEXT,@TXCSR		;TURN ON EXTERNAL MODE
2936	016102	012737	000005	016132		MOV	#5,88\$;LOAD THE NUMBER
2937	016110	032777	004000	163276	86\$:	BIT	#TIMER,@TXDBUF		;CHECK THE TIMER BIT
2938	016116	001374				BNE	86\$;BR IF SET
2939	016120	032777	004000	163266	87\$:	BIT	#TIMER,@TXDBUF		;CHECK THE BIT
2940	016126	001774				BEQ	87\$;BR IF CLEAR
2941	016130	005327				DEC	(PC)+		;DECREMENT THE NUMBER
2942	016132	000005			88\$:	S			;OF TIMES TO REPEAT
2943	016134	001365				BNE	86\$;BR IF MORE TO GO
2944	016136	005713				TST	(R3)		;STRIP DSCA & DSCB FROM CSR
2945	016140	011304				MOV	(R3),R4		;GET THE REGISTER
2946	016142	020504				CMF	R5,R4		;R5=GOOD,R4=?
2947	016144	001402				BEQ	6\$;BR IF OK
2948	016146	104003				HLT	3		;REPORT THE ERROR
2949	016150	104401				SCOPI			;SW09-1?
2950									
2951	016152	104400			6\$:	SCOPE			;SCOPE THE WHOLE TEST
2952	016154	130005			7\$:	.WORD	130005		
2953	016156	130001				.WORD	130001		
2954	016160	000004				.WORD	4		
2955	016162	000000				.WORD	0		
2956									
2957									
2958									
2959									
2960									
2961									
2962									
2963									
2964									
2965									
2966									
2967									
2968									
2969									
2970									

***** TEST 22 *****
; *THIS TEST PROVES THE INTERACTION OF STD
; *WITH STD,SRD
; *AND DATA SET CHANGE ONE AND DATA SET CHANGE TWO.
; *SET THE BIT AND VERIFY THE OTHER BITS ARE SET. CLEAR
; *THE BIT AND VERIFY CLEAR. REPEAT FOR MRESET.

; TEST 22

MODEM CONTROL BITS STD AND SRD INTERACTION TEST

```

2971
2972 016164 012737 000022 001226 ;:*****
2973 016172 012737 016646 001216 1ST22: MOV #22,@#TSTND
2974 016200 012737 016274 001220 MOV #TST23,NEXT
2975 016206 105737 001322 MOV #18,LOCK
2976 016212 001002 TSTB TCNFLG
2977 016214 000137 016634 108: BNE .+6
2978 016220 105737 001334 JMP 68
2979 016224 001773 TSTB STJMFL
2980 016226 105737 001336 BEQ 108
2981 016232 001770 TSTB SRJMFL
2982 016234 005077 163144 BEQ 108
2983 016240 004137 007332 CLR @RXCSR ;CLEAR THE REGISTER
2984 016244 016636 JSR R1,UJUMPER ;THIS CALL DETERMINES IF TURNAROUND CONNECTOR
2985 ;AND OPTIONAL JUMPER ARE USED
2986 016246 052777 000400 163136 BIS #MRESET,@TXCSR ;AND LOADS R5 (EXPECTED) ACCORDINGLY.
2987 016254 004737 005044 JSR PC,SMALL ;RESET THE DEVICE
2988 016260 013703 001404 MOV RXCSR,R3 ;WAIT FOR RESET TO FINISH
2989 016264 005013 CLR (R3) ;LOAD THE RECEIVER CONTROL REGISTER TO R3.
2990 016266 052777 010000 163116 BIS #MEXT,@TXCSR ;CLEAR OUT EXTRA BITS
2991 016274 032713 000010 18: BIS #STD,(R3) ;ENTER EXTERNAL MAINT. MODE
2992 016300 012737 000110 016330 MOV #110,688 ;TURN ON STD
2993 016306 032777 004000 163100 668: BIT #TIMER,@TXDBUF ;LOAD THE NUMBER
2994 016314 001374 BNE 668 ;CHECK THE TIMER BIT
2995 016316 032777 004000 163070 678: BIT #TIMER,@TXDBUF ;BR IF SET
2996 016324 001774 BEQ 678 ;CHECK THE BIT
2997 016326 005327 DEC (PC)+ ;BR IF CLEAR
2998 016330 000110 688: 110 ;CHECK THE BIT
2999 016332 001365 BNE 668 ;BR IF CLEAR
3000 016334 011304 MOV (R3),R4 ;DECREMENT THE NUMBER
3001 016336 020504 CMP R5,R4 ;OF TIMES TO REPEAT
3002 016340 001423 BEQ 28 ;BR IF MORE TO GO
3003 016342 104003 HLT 3 ;GET THE BITS FROM THE RXCSR
3004 016344 104401 SCOP1 ;R5=GOOD R4=?
3005 016346 012737 000005 016376 MOV #5,738 ;BRANCH IF THEY MATCH
3006 016354 032777 004000 163032 718: BIT #TIMER,@TXDBUF ;NO MATCH - SHOW OPR.
3007 016362 001374 BNE 718 ;LOAD THE NUMBER
3008 016364 032777 004000 163022 728: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3009 016372 001774 BEQ 728 ;BR IF SET
3010 016374 005327 DEC (PC)+ ;CHECK THE BIT
3011 016376 000005 738: 5 ;BR IF CLEAR
3012 016400 001365 BNE 718 ;CHECK THE BIT
3013 016402 032713 040000 BIT #RING,(R3) ;BR IF CLEAR
3014 016406 001000 BNE 28 ;DECREMENT THE NUMBER
3015 016410 012737 016422 001220 28: MOV #38,LOCK ;OF TIMES TO REPEAT
3016 016416 042705 002010 BIC #STD:SRD,R5 ;BR IF MORE TO GO
3017 016422 0050 3 38: CLR (R3) ;SW09 SETUP
3018 016424 012737 000005 016454 MOV #5,788 ;CLEAR OUT UNWANTED BITS
3019 016432 032777 004000 162754 768: BIT #TIMER,@TXDBUF ;CLEAR OUT THE REGISTER
3020 016440 001374 BNE 768 ;LOAD THE NUMBER
3021 016442 032777 004000 162744 778: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3022 016450 001774 BEQ 778 ;BR IF SET
3023 016452 005327 DEC (PC)+ ;CHECK THE BIT
3024 016454 000005 788: 5 ;BR IF CLEAR
3025 016456 001365 BNE 768 ;DECREMENT THE NUMBER
3026 016460 011304 MOV (R3),R4 ;OF TIMES TO REPEAT
;BR IF MORE TO GO
;READ BACK THE REGISTER
  
```

MODEM CONTROL BITS STD AND SRD INTERACTION TEST

```
3027 016462 020504      CMP      R5,R4      ;R5=GOOD R4=?
3028 016464 001402      BEQ      4$         ;BRANCH IF ONLY THE DSC BITS ARE SET
3029 016466 104003      HLT      3         ;NO-GO TELL OPR
3030 016470 104401      SCOPI
3031 016472 012737 016500 001220 4$:      MOV      #5$,LOCK   ;SW09 SETUP
3032 016500 052713 000010 5$:      BIS      #STD,(R3)  ;TURN ON STD
3033 016504 012737 000005 016534      MOV      #5,83$     ;LOAD THE NUMBER
3034 016512 032777 004000 162674 81$:     BIT      #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3035 016520 001374      BNE      81$       ;BR IF SET
3036 016522 032777 004000 162664 82$:     BIT      #TIMER,@TXDBUF ;CHECK THE BIT
3037 016530 001774      BEQ      82$       ;BR IF CLEAR
3038 016532 005327      DEC      (PC)+     ;DECREMENT THE NUMBER
3039 016534 000005 83$:      5             ;OF TIMES TO REPEAT
3040 016536 001365      BNE      81$       ;BR IF MORE TO GO
3041 016540 005005      CLR      R5        ;CLEAR OUT EXPECTED
3042 016542 005013      CLR      (R3)      ;CLEAR OUT THE REGISTER
3043 016544 052777 000400 162640      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3044 016552 004737 005044      JSR      PC,SMALL  ;WAIT FOR RESET TO FINISH
3045 016556 052777 010000 162626      BIS      #MEXT,@TXCSR ;TURN ON EXTERNAL MODE
3046 016564 012737 000005 016614      MOV      #5,88$     ;LOAD THE NUMBER
3047 016572 032777 004000 162614 86$:     BIT      #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3048 016600 001374      BNE      86$       ;BR IF SET
3049 016602 032777 004000 162604 87$:     BIT      #TIMER,@TXDBUF ;CHECK THE BIT
3050 016610 001774      BEQ      87$       ;BR IF CLEAR
3051 016612 005327      DEC      (PC)+     ;DECREMENT THE NUMBER
3052 016614 000005 88$:      5             ;OF TIMES TO REPEAT
3053 016616 001365      BNE      86$       ;BR IF MORE TO GO
3054 016620 005713      TST      (R3)      ;STRIP DSCA & DSCB FROM CSP
3055 016622 011304      MOV      (R3),R4   ;GET THE REGISTER
3056 016624 020504      CMP      R5,R4     ;R5=GOOD,R4=?
3057 016626 001402      BEQ      6$         ;BR IF OK
3058 016630 104003      HLT      3         ;REPORT THE ERROR
3059 016632 104401      SCOPI             ;SW09=1?
3060
3061 016634 104400      6$:      SCOPE             ;SCOPE THE WHOLE TEST
3062 016636 002011      7$:      .WORD 2011
3063 016640 002001      .WORD 2001
3064 016642 000010      .WORD 10
3065 016644 000000      .WORD 0
```

```
3066
3067
3068 :***** TEST 23 *****
3069 :*THIS TEST PROVES THE INTERACTION OF DTR!RTS!STD
3070 :*WITH RING,DSR,CTS,CARDET,STD,SRD
3071 :*AND DATA SET CHANGE ONE AND DATA SET CHANGE TWO.
3072 :*SET THE BIT AND VERIFY THE OTHER BITS ARE SET. CLEAR
3073 :*THE BIT AND VERIFY CLEAR. REPEAT FOR MRESET.
3074 :*****
```

```
3075
3076 :*****
3077 :
3078 : TEST 23
3079 :
3080 :*****
3081 :*****
3082 016646 012737 000023 001226 TST23: MOV #23,@TSTNO
```

ALL MODEM CONTROL BITS INTERACTION TEST

3083	016654	012737	017330	001216		MOV	#TST24,NEXT	
3084	016662	012737	016756	001220		MOV	#18,LOCK	
3085	016670	105737	001322			TSTB	TCNFLG	
3086	016674	001002				BNE	.+6	
3087	016676	000137	017316		10\$:	JMP	6\$	
3088	016702	105737	001334			TSTB	STJMFL	
3089	016706	001773				BEQ	10\$	
3090	016710	105737	001336			TSTB	SRJMFL	
3091	016714	001770				BEQ	10\$	
3092	016716	005077	162462			CLR	@RXCSR	;CLEAR THE REGISTER
3093	016722	004137	007332			JSR	R1,0JUMPER	;THIS CALL DETERMINES IF TURNAROUND CONNECTOR
3094	016726	017320				7\$;AND OPTIONAL JUMPER ARE USED
3095								;AND LOADS R5 (EXPECTED) ACCORDINGLY.
3096	016730	052777	000400	162454		BIS	#MRESET,@TXCSR	;RESET THE DEVICE
3097	016736	004737	005044			JSR	PC,SMALL	;WAIT FOR RESET TO FINISH
3098	016742	013703	001404			MOV	RXCSR,R3	;LOAD THE RECEIVER CONTROL REGISTER TO R3.
3099	016746	005013				CLR	(R3)	;CLEAR OUT EXTRA BITS
3100	016750	052777	010000	162434		BIS	#MEXT,@TXCSR	;ENTER EXTERNAL MAINT. MODE
3101	016756	052713	000016		1\$:	BIS	#DTR!RTS!STD,(R3)	;TURN ON DTR!RTS!STD
3102	016762	012737	000110	017012		MOV	#110,68\$;LOAD THE NUMBER
3103	016770	032777	004000	162416	66\$:	BIT	#TIMER,@TXDBUF	;CHECK THE TIMER BIT
3104	016776	001374				BNE	66\$;BR IF SET
3105	017000	032777	004000	162406	67\$:	BIT	#TIMER,@TXDBUF	;CHECK THE BIT
3106	017006	001774				BEQ	67\$;BR IF CLEAR
3107	017010	005327				DEC	(PC)+	;DECREMENT THE NUMBER
3108	017012	000110			68\$:	110		;OF TIMES TO REPEAT
3109	017014	001365				BNE	66\$;BR IF MORE TO GO
3110	017016	011304				MOV	(R3),R4	;GET THE BITS FROM THE RXCSR
3111	017020	020504				CMP	R5,R4	;R5=GOOD R4=?
3112	017022	001423				BEQ	2\$;BRANCH IF THEY MATCH
3113	017024	104003				HLT	3	;NO MATCH - SHOW OPR.
3114	017026	104401				SCOPI		
3115	017030	012737	000005	017060		MOV	#5,73\$;LOAD THE NUMBER
3116	017036	032777	004000	162350	71\$:	BIT	#TIMER,@TXDBUF	;CHECK THE TIMER BIT
3117	017044	001374				BNE	71\$;BR IF SET
3118	017046	032777	004000	162340	72\$:	BIT	#TIMER,@TXDBUF	;CHECK THE BIT
3119	017054	001774				BEQ	72\$;BR IF CLEAR
3120	017056	005327				DEC	(PC)+	;DECREMENT THE NUMBER
3121	017060	000005			73\$:	5		;OF TIMES TO REPEAT
3122	017062	001365				BNE	71\$;BR IF MORE TO GO
3123	017064	032713	040000			BIT	#RING,(R3)	
3124	017070	001000				BNE	2\$	
3125	017072	012737	017104	001220	2\$:	MOV	#3\$,LOCK	;SW09 SETUP
3126	017100	042705	073016			BIC	#RING!CTS!CARDET	;SRD!DSR!STD!RTS!DTR,R5
3127	017104	005013			3\$:	CLR	(R3)	;CLEAR OUT UNWANTED BITS
3128	017106	012737	000005	017136		MOV	#5,78\$;LOAD THE NUMBER
3129	017114	032777	004000	162272	76\$:	BIT	#TIMER,@TXDBUF	;CHECK THE TIMER BIT
3130	017122	001374				BNE	76\$;BR IF SET
3131	017124	032777	004000	162262	77\$:	BIT	#TIMER,@TXDBUF	;CHECK THE BIT
3132	017132	001774				BEQ	77\$;BR IF CLEAR
3133	017134	005327				DEC	(PC)+	;DECREMENT THE NUMBER
3134	017136	000005			78\$:	5		;OF TIMES TO REPEAT
3135	017140	001365				BNE	76\$;BR IF MORE TO GO
3136	017142	011304				MOV	(R3),R4	;READ BACK THE REGISTER
3137	017144	020504				CMP	R5,R4	;R5=GOOD R4=?
3138	017146	001402				BEQ	4\$;BRANCH IF ONLY THE DSC BITS ARE SET

ALL MODEM CONTROL BITS INTERACTION TEST

```
3139 017150 104003 HLT 3 ;NO-GO TELL OPR
3140 017152 104401 SCOP1
3141 017154 012737 017162 001220 4$: MOV #5$,LOCK ;SW09 SETUP
3142 017162 052713 000016 5$: BIS #DTR,RTS!STD,(R3) ;TURN ON DTR!RTS!STD
3143 017166 012737 000005 017216 MOV #5,83$ ;LOAD THE NUMBER
3144 017174 032777 004000 162212 81$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3145 017202 001374 BNE 81$ ;BR IF SET
3146 017204 032777 004000 162202 82$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3147 017212 001774 BEQ 82$ ;BR IF CLEAR
3148 017214 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3149 017216 000005 83$: S ;OF TIMES TO REPEAT
3150 017220 001365 BNE 81$ ;BR IF MORE TO GO
3151 017222 005005 CLR R5 ;CLEAR OUT EXPECTED
3152 017224 005013 CLR (R3) ;CLEAR OUT THE REGISTER
3153 017226 052777 000400 162156 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3154 017234 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3155 017240 052777 010000 162144 BIS #MEXT,@TXCSR ;TURN ON EXTERNAL MODE
3156 017246 012737 000005 017276 MOV #5,88$ ;LOAD THE NUMBER
3157 017254 032777 004000 162132 86$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3158 017262 001374 BNE 86$ ;BR IF SET
3159 017264 032777 004000 162122 87$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3160 017272 001774 BEQ 87$ ;BR IF CLEAR
3161 017274 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3162 017276 000005 88$: S ;OF TIMES TO REPEAT
3163 017300 001365 BNE 86$ ;BR IF MORE TO GO
3164 017302 005713 TST (R3) ;STRIP DSCA & DSCB FROM CSR
3165 017304 011304 MOV (R3),R4 ;GET THE REGISTER
3166 017306 020504 CMP R5,R4 ;R5=GOOD,R4=?
3167 017310 001402 BEQ 6$ ;BR IF OK
3168 017312 104003 HLT 3 ;REPORT THE ERROR
3169 017314 104401 SCOP1 ;SW09=1?
3170
3171 017316 104400 6$: SCOPE ;SCOPE THE WHOLE TEST
3172 017320 173017 7$: .WORD 173017
3173 017322 173001 .WORD 173001
3174 017324 001016 .WORD 1016
3175 017326 000000 .WORD 0
3176
3177
```

```
***** TEST 24 *****
*TEST THAT SETTING TRANSMIT INTERRUPT
*ENABLE AND TRANSMITTER DONE PRODUCE
*AN INTERRUPT ON THE TRANSMITTER VECTOR.
*****
```

```
*****
: TEST 24
*****
```

```
3189 017330 012737 000024 001226 TST24: MOV #24,@TSTNO
3190 017336 012737 017446 001216 MOV #TST25,NEXT
3191 017344 012706 001150 MOV #STACK,SP
3192 017350 000005 RESET
3193 017352 012737 000340 177776 MOV #340,PS ;LOCK OUT INTERRUPTS
3194 017360 052777 000400 162024 BIS #MRESET,@TXCSR ;RESET THE DEVICE
```

CZDPCC MACY11 30A(1052) 26-FEB-79 09:33 PAGE 66
CZDPCD.P11 08-FEB-79 13:15 TRANSMITTER DONE BIT INTERRUPT TEST

```

3195 017366 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3196 017372 004537 007232 JSR R5,SETVEC ;SET UP THE VECTORS
3197 017376 007254 NO.ATRAP ;VECTOR "A"
3198 017400 017434 1$ ;VECTOR B
3199 017402 340 340 .BYTE 340,340 ;PRIORITY
3200 017404 052777 004000 162000 BIS #SYSTST,@TXCSR ;ZERO CPU PRIORITY
3201 017412 005037 177776 CLR PS ;TURN ON TXINT ENABLE
3202 017416 052777 000100 161766 BIS #TXINTE,@TXCSR ;STALL
3203 017424 000240 NOP ;DITTO
3204 017426 000240 NOP ;DITTO
3205 017430 000240 NOP ;DITTO
3206 017432 104027 HLT 27 ;DUP FAILED TO INTERRUPT
3207 017434 012706 001150 1$: MOV #STACK,SP ;RESET THE STACK
3208 017440 005077 161746 CLR @TXCSR ;DISABLE DUP11
3209 017444 104400 SCOPE ;SCOPE THIS TEST
3210
3211 ;***** TEST 25 *****
3212 ;*TEST TO VERIFY THAT A TRANSMITTER DONE
3213 ;*INTERRUPT WILL ONLY OCCUR ONCE IF THE
3214 ;*TXCSR AND TXDBUF ARE *NOT* READ OR WRITTEN.
3215 ;*****
3216
3217 ;*****
3218 ; TEST 25
3219 ;*****
3220
3221 ;*****
3222 ;*****
3223 017446 012737 000025 001226 TST25: MOV #25,@TSTNO
3224 017454 012737 017640 001216 MOV #TST26,NEXT
3225 017462 012737 000340 177776 MOV #340,PS ;LOCK OUT INTERRUPTS
3226 017470 052777 000400 161714 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3227 017476 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3228 017502 004537 007232 JSR R5,SETVEC ;SETUP FOR INTERRUPTS
3229 017506 007254 NO.ATRAP ;RECEIVER
3230 017510 017610 1$ ;TRANSMITTER
3231 017512 340 340 .BYTE 340,340 ;LEVEL
3232 017514 052777 004000 161670 BIS #SYSTST,@TXCSR ;TURN ON CLOCK
3233 017522 052777 000100 161662 BIS #TXINTE,@TXCSR ;TURN ON INT. ENABLE
3234 017530 005037 177776 CLR PS ;LOWER PROCESSOR STATUS
3235 017534 000240 NOP ;STALL
3236 017536 000240 NOP ;DITTO
3237 017540 000240 NOP ;DITTO
3238 017542 104027 HLT 27 ;DUP FAILED TO INTERRUPT
3239 017544 000411 BR 5$ ;LEAVE TEST
3240 017546 005037 177776 2$: CLR PS ;LOWER PROCESSOR STATUS
3241 017552 000240 NOP ;STALL
3242 017554 000240 NOP ;DITTO
3243 017556 000240 NOP ;DITTO
3244 017560 105777 161626 4$: TSTB @TXCSR ;CHECK THE DONE BIT
3245 017564 100401 BMI 5$ ;BR IF SET
3246 017566 104024 HLT 24 ;DONE IS CLEARED AND SHOULDN'T BE
3247 017570 012737 000340 177776 5$: MOV #340,PS ;RAISE PROCESSOR STATUS
3248 017576 005077 161610 CLR @TXCSR ;CLEAR OUT DUP
3249 017602 012706 001150 MOV #STACK,SP ;RESET STACK
3250 017606 104400 SCOPE ;SCOPE THIS TEST

```

3251							
3252	017610	012716	017546	1\$:	MOV #2\$, (SP)		:SET UP SECOND PART OF TEST
3253	017614	004537	007232		JSR R5, SETVEC		:SETUP FOR SECOND INTERRUPT TRY
3254	017620	007254			NO.ATRAP		:RECEIVER
3255	017622	017630			3\$:TRANSMITTER
3256	017624	340	340		.BYTE 340,340		:LEVEL
3257	017626	0000C2			RTI		:RETURN
3258	017630			3\$:			
3259	017630	104026			HLT 26		:REPORT THE FACT YOU GOT HERE
3260	017632	012716	017560		MOV #4\$, (SP)		:SETUP FOR END OF TEST
3261	017636	000002			RTI		:RETURN
3262							
3263							:***** TEST 26 *****
3264							:*TEST THAT SETTING DATA SET INTERRUPT
3265							:*ENABLE AND RECEIVING A DATA SET
3266							:*CHANGE 1 OR DATA SET CHANGE 2
3267							:*PRODUCES AN INTERRUPT TO THE
3268							:*RECEIVER VECTOR
3269							:*****
3270							
3271							:*****
3272							:*TEST 26
3273							:*****
3274							:*****
3275							:*****
3276							:*****
3277	017640	012737	000026	001226	TST26: MOV #26, @TSTNO		
3278	017646	012737	020156	001216	MOV #TST27, NEXT		
3279	017654	105737	001322		TSTR TCNFLG		
3280	017660	001002			BNE .+6		
3281	017662	000137	020144		JMP 5\$		
3282	017666	012737	017734	001220	MOV #1\$, LOCK		:SW09 SETUP
3283	017674	012737	000340	177776	MOV #340, PS		:LOCK OUT INTERRUPTS
3284	017702	052777	000400	161502	BIS #MRESET, @TXCSR		:RESET THE DEVICE
3285	017710	004737	0050.4		JSR PC, SMALL		:WAIT FOR RESET TO FINISH
3286	017714	052777	010000	161470	BIS #MEXT, @TXCSR		:ENTER MAINT EXTERNAL MODE
3287	017722	004537	007232		JSR R5, SETVEC		:SET UP VECTORS
3288	017726	020014			2\$:RECEIVER
3289	017730	007260			NO.BTRAP		:TRANSMITTER
3290	017732	340	340		.BYTE 340,340		:PRORITY AT 7
3291	017734	005037	177776		1\$:	CLR PS	:LOWER PS
3292	017740	052777	000040	161436	BIS #DSINTE, @RXCSR		:TURN ON INT. ENABLE
3293	017746	052777	000004	161430	BIS #RTS, @RXCSR		:TURN ON INT. BIT
3294	017754	012737	000005	020004	MOV #5, 68\$:LOAD THE NUMBER
3295	017762	032777	004000	161424	66\$:	BIT #TIMER, @TXDBUF	:CHECK THE TIMER BIT
3296	017770	001374			BNE 66\$:BR IF SET
3297	017772	032777	004000	161414	67\$:	BIT #TIMER, @TXDBUF	:CHECK THE BIT
3298	020000	001774			BEQ 67\$:BR IF CLEAR
3299	020002	005327			DEC (PC)+		:DECREMENT THE NUMBER
3300	020004	000005			68\$:	5	:OF TIMES TO REPEAT
3301	020006	001365			BNE 66\$:BR IF MORE TO GO
3302	020010	104022			HLT 22		:FAILED TO INTERRUPT
3303	020012	104401			SCOP1		:SW09=1
3304	020014	012706	001150		2\$:	MOV #STACK, SP	:RESET THE STACK
3305	020020	005077	161360		CLR @RXCSR		:CLEAR OUT RECEIVER CONTRL REGISTER.
3306	020024	012737	020044	001220	MOV #3\$, LOCK		:SW09 SETUP

RECEIVER DATA SET CHANGE BITS INTERRUPT TEST

```
3307 020032 004537 007232 JSR R5,SETVEC ;SET THE VECTORS
3308 020036 020132 4$ ;RECEIVER
3309 020040 007260 NO.BTRAP ;TRANSMITTERS
3310 020042 340 340 .BYTE 340,340 ;PRIORITY @7
3311 020044 012737 000005 001236 3$: MOV #5,TEMP1 ;LOAD TEMP1
3312 020052 052777 000040 161324 BIS #DSINTE,@RXCSR ;TURN ON INT. ENABLE
3313 020060 005037 177776 CLR PS ;LOWER CPU STATUS
3314 020064 052777 000002 161312 BIS #DTR,@RXCSR ;PUSH OUT INT. BITS
3315 020072 012737 000156 020122 MOV #110,73$ ;LOAD THE NUMBER
3316 020100 032777 004000 161306 71$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3317 020106 001374 BNE 71$ ;BR IF SET
3318 020110 032777 004000 161276 72$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3319 020116 001774 BEQ 72$ ;BR IF CLEAR
3320 020120 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3321 020122 000156 73$: 110. ;OF TIMES TO REPEAT
3322 020124 001365 RNE 71$ ;BR IF MORE TO GO
3323 020126 104022 HLT 22 ;FAILED TO INTERRUPT
3324 020130 104401 SCOPE1 ;BIT09=1?
3325 020132 4$:
3326 020132 032777 100000 161244 BIT #BIT15,@RXCSR ;CHECK FOR INTERRUPT FROM DSC1
3327 020140 001001 BNE 5$ ;IT CAME FROM DSC1
3328 020142 104023 HLT 23 ;BIT15 IS CLEARED - INTERRUPTED
3329 ;FROM THE WRONG DSC BIT.
3330 020144 005077 161234 5$: CLR @RXCSR ;CLEAR OUT RECEIVER CONTROL REG
3331 020150 012706 001150 MOV #STACK,SP ;RESET THE STACK
3332 020154 104400 SCOPE ;SCOPE THIS TEST
3333 ;***** TEST 27 *****
3334 ;*TEST THAT SETTING RECEIVER INTERRUPT
3335 ;*ENABLE AND RECEIVER DONE CAUSES AN
3336 ;*INTERRUPT TO THE RECEIVER VECTOR
3337 ;*****
3338
3339 ;*****
3340 ; TEST 27
3341 ;*****
3342
3343 ;*****
3344 ;*****
3345 020156 012737 000027 001226 TST27: MOV #27,@TSTNO
3346 020164 012737 020436 001216 MOV #TST30,NEXT
3347 020172 012737 000340 177776 MOV #340,PS ;LOCK OUT INTERRUPTS
3348 020200 052777 000400 161204 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3349 020206 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3350 020212 052777 004000 161172 BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
3351 020220 004537 007232 JSR R5,SETVEC ;SET UP VECTORS
3352 020224 020420 3$ ;RECEIVER
3353 020226 007260 NO.BTRAP ;TRANSMITTER
3354 020230 340 340 .BYTE 340,340 ;PRIORITY AT 7
3355 020232 005037 177776 CLR PS ;LOWER PS
3356 020236 052777 000020 161140 BIS #RCVEN,@RXCSR ;TURN ON RECEIVER
3357 020244 052777 000100 161132 BIS #RINTEN,@RXCSR ;TURN ON INT. ENABLE
3358 020252 052777 000020 161132 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
3359 020260 1$:
3360 020260 012737 000005 020310 MOV #5,68$ ;LOAD THE NUMBER
3361 020266 032777 004000 161120 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3362 020274 001374 BNE 66$ ;BR IF SET
```

```

3363 020276 032777 004000 161110 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3364 020304 001774 BEQ 67$ ;BR IF CLEAR
3365 020306 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3366 020310 000005 68$: 5 ;OF TIMES TO REPEAT
3367 020312 001365 BNE 66$ ;BR IF MORE TO GO
3368 020314 032777 000200 161070 BIT #TXDONE,@TXCSR ;TEST TXDONE
3369 020322 001001 BNE 2$ ;BR IF SET
3370 020324 104024 HLT 24 ;TXDONE FAILED TO SET
3371 020326
3372 020326 012777 000400 161060 2$: MOV #400,@TXDBUF ;LOAD TX BUFFER
3373 020334 105777 161052 TSTB @TXCSR ;CHECK FOR
3374 020340 100375 BPL -4 ;DONE
3375 020342 005077 161046 CLR @TXDBUF ;CLEAR TX BUFFER
3376 020346 105777 161040 TSTB @TXCSR ;AND CHECK
3377 020352 100375 BPL -4 ;FOR DONE
3378 020354 012777 001000 161032 MOV #1000,@TXDBUF ;LOAD END OF MSG
3379 020362 012737 000050 020412 MOV #40,,73$ ;LOAD THE NUMBER
3380 020370 032777 004000 161016 71$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3381 020376 001374 BNE 71$ ;BR IF SET
3382 020400 032777 004000 161006 72$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3383 020406 001774 BEQ 72$ ;BR IF CLEAR
3384 020410 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3385 020412 000050 73$: 40. ;OF TIMES TO REPEAT
3386 020414 001365 BNE 71$ ;BR IF MORE TO GO
3387 020416 104022 HLT 22 ;RECEIVER FAILED TO INTERRUPT
3388 020420 012706 001150 3$: MOV #STACK,SP ;RESET STACK
3389 020424 005077 160754 CLR @RXCSR ;CLEAR OUT REGISTER
3390 020430 005077 160756 CLR @TXCSR ;DITTO
3391 020434 104400 SCOPE ;SCOPE THIS TEST
3392
3393
3394 ;***** TEST 30 *****
3395 ;*TEST TO VERIFY THAT A RECEIVER DONE
3396 ;*INTERRUPT WILL ONLY OCCUR ONCE IF THE
3397 ;*RXCSR AND RXDBUF ARE NOT READ OR WRITTEN
3398 ;*****
3399
3400
3401
3402
3403
3404
3405 020436 012737 000030 001226
3406 020444 012737 021016 001216
3407 020452 012737 000340 177776
3408 020460 052777 000400 160724
3409 020466 004737 005044
3410 020472 052777 004000 160712
3411 020500 004537 007232
3412 020504 020766
3413 020506 007260
3414 020510 340 340
3415 020512 052777 000020 160664
3416 020520 052777 000100 160656
3417 020526 052777 000020 160656
3418 020534
  
```

```

:*****
:
: TEST 30
:
:*****
  
```

```

1ST30: MOV #30,@TSTNO
MOV #TST31,NEXT
MOV #340,PS ;LOCK OUT INTERRUPTS
BIS #MRESET,@TXCSR ;RESET THE DEVICE
JSR PC,SMALL ;WAIT FOR RESET TO FINISH
BIS #SYSTST,@TXCSR ;ENTER SYSTST MODE
JSR R5,SETVEC ;SETUP VECTORS
6$ ;RECEIVER VECTOR
NO.BTRAP ;TRANSMITTER VECTOR
.BYTE 340,340 ;LEVEL
BIS #RCVEN,@RXCSR ;TURN ON RECEIVER
BIS #RINTEN,@RXCSR ;TURN ON INT. ENABLE
BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
  
```

ONLY ONE INTERRUPT PER RXDONE TEST

3419	020534	012737	000005	020564		MOV	#5,68\$:LOAD THE NUMBER
3420	020542	032777	004000	160644	66\$:	BIT	#TIMER,@TXDBUF		:CHECK THE TIMER BIT
3421	020550	001374				BNE	66\$:BR IF SET
3422	020552	032777	004000	160634	67\$:	BIT	#TIMER,@TXDBUF		:CHECK THE BIT
3423	020560	001774				BEQ	67\$:BR IF CLEAR
3424	020562	005327				DEC	(PC)+		:DECREMENT THE NUMBER
3425	020564	000005			68\$:	5			:OF TIMES TO REPEAT
3426	020566	001365				BNE	66\$:BR IF MORE TO GO
3427	020570	032777	000200	160614		BIT	#TXDONE,@TXCSR		:TEST TXDONE
3428	020576	001001				BNE	2\$:WAIT
3429	020600	104024				HLT	24		:TXDONE FAILED TO SET
3430	020602	012777	000400	160604	2\$:	MOV	#TSOM,@TXDBUF		:LOAD TX BUFFER
3431	020610	105777	160576			TSTB	@TXCSR		:CHECK DONE
3432	020614	100375				BPL	.-4		:AND THEN
3433	020616	005077	160572			CLR	@TXDBUF		:LOAD BUFFER
3434	020622	105777	160564			TSTB	@TXCSR		:AND CHECK
3435	020626	100375				BPL	.-4		:DONE AGAIN, THEN
3436	020630	012777	001000	160556		MOV	#TEOM,@TXDBUF		:SET END OF MSG
3437	020636	005037	177776			CLR	PS		:LOWER PS
3438	020642				10\$:				
3439	020642	012737	000050	020672		MOV	#40.,73\$:LOAD THE NUMBER
3440	020650	032777	004000	160536	1\$:	BIT	#TIMER,@TXDBUF		:CHECK THE TIMER BIT
3441	020656	001374				BNE	71\$:BR IF SET
3442	020660	032777	004000	160526	72\$:	BIT	#TIMER,@TXDBUF		:CHECK THE BIT
3443	020666	001774				BEQ	72\$:BR IF CLEAR
3444	020670	005327				DEC	(PC)+		:DECREMENT THE NUMBER
3445	020672	000050			73\$:	40.			:OF TIMES TO REPEAT
3446	020674	001365				BNE	71\$:BR IF MORE TO GO
3447	020676	104022				HLT	22		:RECEIVER FAILED TO INTERRUPT
3448	020700	000420				BR	5\$:LEAVE
3449	020702				3\$:				
3450	020702	032777	004000	160504	74\$:	BIT	#TIMER,@TXDBUF		:CHECK THE TIMER BIT
3451	020710	001374				BNE	74\$:BR IF SET
3452	020712	032777	004000	160474	75\$:	BIT	#TIMER,@TXDBUF		:CHECK THE TIMER BIT
3453	020720	001774				BEQ	75\$:BR IF CLEAR
3454	020722	012737	000340	177776		MOV	#340,PS		:RAISE PS
3455	020730	033777	000200	160446		BIT	RXDONE,@RXCSR		:TEST RXDONE
3456	020736	001001				BNE	5\$:BR IF SET
3457	020740	104024				HLT	24		:RXDONE IS NOT SET AND SHOULD BE
3458	020742	012737	000340	177776	5\$:	MOV	#340,PS		:LOCKOUT INTERRUPTS
3459	020750	005077	160430			CLR	@RXCSR		:CLEAR OUT DEVICE
3460	020754	005077	160426			CLR	@RXDBUF		:DITTO
3461	020760	012706	001150			MOV	#STACK,SP		:RESET THE STACK
3462	020764	104400				SCOPE			:SCOPE THIS TEST
3463	020766				6\$:				
3464	020766	012716	020702			MOV	#3\$, (SP)		:2ND PART SETUP
3465	020772	004537	007232			JSR	R5,SETVEC		:SETUP FOR SECOND INTERRUPT TRY
3466	020776	021006				7\$:RECEIVER VECTOR
3467	021000	007260				NO.BTRAP			:TRANSMITTER VECTOR
3468	021002	340	340			.BYTE	340,340		:LEVEL
3469	021004	000002				RTI			:RETURN
3470	021006				7\$:				
3471	021006	104023				HLT	23		:REPORT THE FACT YOU GOT HERE
3472									:YOU TOOK A SECOND INTERRUPT AND SHOULDN'T HAVE
3473	021010	012716	020742			MOV	#5\$, (SP)		:SETUP TO LEAVE TEST
3474	021014	000002				RTI			:LEAVE

3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530

021016	012737	000031	001226
021024	012737	021356	001216
021032	012737	000340	177776
021040	052777	000400	160344
021046	004737	005044	
021052	004537	007232	
021056	021336		
021060	007260		
021062	340	340	
021064	052777	004000	160320
021072	052777	000020	160304
021100	052777	000100	160276
021106	052777	001000	160274
021114	052777	000020	160270
021122			
021122	012737	000005	021152
021130	032777	004000	160256
021136	001374		
021140	032777	004000	160246
021146	001774		
021150	005327		
021152	000005		
021154	001365		
021156	032777	000200	160226
021164	001001		
021166	104024		
021170			
021170	012777	000400	160216
021176	105777	160210	
021202	100375		
021204	005077	160204	
021210	105777	160176	
021214	100375		
021216	012777	001000	160170
021224	012737	000144	021254
021232	032777	004000	160154
021240	001374		
021242	032777	004000	160144
021250	001774		
021252	005327		
021254	000144		
021256	001365		

```

:***** TEST 31 *****
:*TEST TO VERIFY THAT INTERRUPT VECTOR "A"
:*OCCURS BEFORE INTERRUPT VECTOR "B" EVEN
:*WHEN VECTOR "B" IS ENABLED BEFORE
:*VECTOR "A"
:*****

:*****
: TEST 31
:*****

TST31:  MOV    #31,@#TSTNO
        MOV    #TST32,NEXT
        MOV    #340,PS          ;SET PRIORITY TO 7
        BIS    #MRESET,@TXCSR  ;RESET THE DEVICE
        JSR    PC,SMALL        ;WAIT FOR RESET TO FINISH
        JSR    R5,SETVEC      ;SET UP THE VECTORS
        4$
        NO.BTRAP              ;RECEIVER VECTOR
        .BYTE 340,340         ;TRANSMITTER VECTOR
        BIS    #SYSTST,@TXCSR  ;LEVEL
        BIS    #RCVEN,@RXCSR   ;ENTER SYSTEM TEST MODE
        BIS    #RINTEN,@RXCSR  ;TURN ON RECEIVER
        BIS    #CRCEN,@PARCSR  ;TURN ON REC. DONE INT. ENAB. E
        BIS    #SEND,@TXCSR    ;TURN OFF CRC
        BIS    #SEND,@TXCSR    ;TURN ON TRANSMITTER

1$:     MOV    #5,68$          ;LOAD THE NUMBER
66$:   BIT    #TIMER,@TXDBUF  ;CHECK THE TIMER BIT
        BNE   66$             ;BR IF SET
67$:   BIT    #TIMER,@TXDBUF  ;CHECK THE BIT
        BEQ   67$             ;BR IF CLEAR
        DEC   (PC)+           ;DECREMENT THE NUMBER
68$:   5
        BNE   66$             ;OF TIMES TO REPEAT
        BIT   #TXDONE,@TXCSR  ;BR IF MORE TO GO
        BNE   2$             ;TEST TXDONE
        HLT   24              ;BR IF SET
        HLT   24              ;TXDONE FAILED TO SET

2$:     MOV    #400,@TXDBUF    ;LOAD TX BUFFER
        TSTB  @TXCSR          ;CHECK FOR
        BPL   -4              ;DONE
        CL'   @TXDBUF         ;LOAD THE BUFFER
        TSTB  @TXCSR          ;AND CHECK
        BPL   -4              ;DONE AGAIN
        MOV   #1000,@TXDBUF   ;LOAD TEOM
        MOV   #100.,73$      ;LOAD THE NUMBER

71$:   BIT    #TIMER,@TXDBUF  ;CHECK THE TIMER BIT
        BNE   71$             ;BR IF SET
72$:   BIT    #TIMER,@TXDBUF  ;CHECK THE BIT
        BEQ   72$             ;BR IF CLEAR
        DEC   (PC)+           ;DECREMENT THE NUMBER
73$:   100.
        BNE   71$             ;OF TIMES TO REPEAT
        BNE   71$             ;BR IF MORE TO GO

```

DUAL VECTORING---RECEIVER BEFORE TRANSMITTER---TEST

```
3531 021260 105777 160120 TSTB @RXCSR ;CHECK DONE
3532 021264 100401 BMI 58 ;BR IF SET
3533 021266 104024 HLT 24 ;DONE FAILED TO SET
3534 021270 032777 000200 160114 58: BIT #TXDONE,@TXCSR ;TEST TXDONE
3535 021276 001001 BNE 38 ;BR IF SET
3536 021300 104024 HLT 24 ;TXDONE NOT SET SHOULD BE
3537 021302 052777 000100 160102 38: BIS #TXINTE,@TXCSR ;TURN ON TRANSMITTER INT. ENABLE
3538 021310 005037 177776 CLR PS ;LOWER PROCESSOR STATUS
3539 021314 032777 004000 160072 748: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3540 021322 001374 BNE 748 ;BR IF SET
3541 021324 032777 004000 160062 758: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3542 021332 001774 BEQ 758 ;BR IF CLEAR
3543 021334 104027 HLT 27 ;DUP FAILED TO INTERRUPT
3544 021336 012706 001150 48: MOV #STACK,SP ;RESET THE STACK
3545 021342 052777 000400 160042 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3546 021350 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3547 021354 104400 SCOPE ;SCOPE THIS TEST
3548
3549
3550 ;***** TEST 32 *****
3551 ;*TEST TO VERIFY THAT SERVICING THE
3552 ;*TXDONE BIT RE-ARMS THE INTERRUPT
3553 ;*LOGIC IF INTERRUPT ENABLE IS SET.
3554 ;*****
3555
3556 ;*****
3557 ; TEST 32
3558 ;*****
3559
3560 ;*****
```

```
3561 021356 012737 000032 001226 TST32: MOV #32,@TSTNO
3562 021364 012737 021664 001216 MOV #TST33,NEXT
3563 021372 012737 000340 177776 MOV #340,PS ;LOCK OUT INTERRUPTS
3564 021400 052777 000400 160004 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3565 021406 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3566 021412 004537 007232 JSR R5,SETVEC ;INTERRUPT VECTOR SETUP
3567 021416 007254 NO.ATRAP ;RECEIVER VECTOR
3568 021420 021576 48 ;TRANSMITTER VECTOR
3569 021422 340 340 .BYTE 340,340 ;LEVEL
3570 021424 052777 004120 157760 BIS #SYSTST!SEND!TXINTE,@TXCSR ;TURN ON TRANSMITTER, CLOCK
3571 ;AND INTERRUPTS
3572 021432 052777 000400 157754 BIS #TSOM,@TXDBUF ;LOAD START OF MSG
3573 021440 005037 177776 CLR PS ;LOWER PROCESSOR STATUS
3574 021444 012737 000005 021474 MOV #5,688 ;LOAD THE NUMBER
3575 021452 032777 004000 157734 668: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3576 021460 001374 BNE 668 ;BR IF SET
3577 021462 032777 004000 157724 678: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3578 021470 001774 BEQ 678 ;BR IF CLEAR
3579 021472 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3580 021474 000005 688: 5 ;OF TIMES TO REPEAT
3581 021476 001365 BNE 668 ;BR IF MORE TO GO
3582 021500 104027 HLT 27 ;DUP FAILED TO INTERRUPT THE FIRST TIME
3583 021502 000427 BR 38 ;LEAVE THE TEST
3584 021504 005037 177776 18: CLR PS ;LOWER PROCESSOR STATUS
3585 021510 012737 000005 021540 MOV #5,738 ;LOAD THE NUMBER
3586 021516 032777 004000 157670 718: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
```

```

3587 021524 001374          BNE 71$          ;BR IF SET
3588 021526 032777 004000 157660 72$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3589 021534 001774          BEQ 72$          ;BR IF CLEAR
3590 021536 005327          DEC (PC)+        ;DECREMENT THE NUMBER
3591 021540 000005          5              ;OF TIMES TO REPEAT
3592 021542 001365          BNE 71$          ;BR IF MORE TO GO
3593 021544 104027          HLT 27          ;DUP FAILED TO INTERRUPT AFTER SERVICING DONE
3594 021546 000405          BR 3$           ;LEAVE
3595 021550                2$:
3596 021550 005077 157636          CLR @TXCSR      ;SHUT DOWN THE DUP
3597 021554 012716 021562          MOV #3$, (SP)  ;SETUP TO END TEST
3598 021560 000002          RTI            ;RETURN
3599 021562 012737 000340 177776 3$: MOV #340,PS    ;RAISE PROCESSOR STATUS
3600 021570 012706 001150          MOV #STACK,SP  ;RESET STACK
3601 021574 104400          SCOPE          ;SCOPE THIS TEST
3602 021576 032777 000200 157606 4$: BIT #TXDONE,@TXCSR ;CLEAR DONE AND RE-ARM INTERRUPT
3603 021604 005077 157604          CLR @TXDBUF    ;LOAD BUFFER
3604 021610 012737 000005 021640  MOV #5,78$     ;LOAD THE NUMBER
3605 021616 032777 004000 157570 76$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3606 021624 001374          BNE 76$        ;BR IF SET
3607 021626 032777 004000 157560 77$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3608 021634 001774          BEQ 77$        ;BR IF CLEAR
3609 021636 005327          DEC (PC)+        ;DECREMENT THE NUMBER
3610 021640 000005          5              ;OF TIMES TO REPEAT
3611 021642 001365          BNE 76$        ;BR IF MORE TO GO
3612 021644 012716 021504          MOV #1$, (SP)  ;SETUP TO FINISH TEST
3613 021650 004537 007232          JSR R5,SETVEC  ;SETUP VECTORS FOR NEXT PART OF TEST
3614 021654 007254          NO.ATRAPH     ;RECEIVER VECTOR
3615 021656 021550          2$           ;TRANSMITTER VECTOR
3616 021660 340 340          .BYTE 340,340 ;LEVEL
3617 021662 000002          RTI            ;RETURN

```

```

3618
3619
3620 :***** TEST 33 *****
3621 :*TEST TO VERIFY THAT SERVICING THE
3622 :*RXDONE BIT RE-ARMS THE INTERRUPT
3623 :*LOGIC IF INTERRUPT ENABLE IS SET.
3624 :*****
3625

```

```

3626 :*****
3627 :
3628 : TEST 33
3629 :
3630 :*****
3631 :*****

```

```

3632 021664 012737 000033 001226 TST33: MOV #33,@TSTNO
3633 021672 012737 022256 001216      MOV #TST34,NEXT
3634 021700 012737 000340 177776      MOV #340,PS ;LOCK OUT INTERRUPTS
3635 021706 052777 000400 157476      BIS #MRESET,@TXCSR ;RESET THE DEVICE
3636 021714 004737 005044          JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3637 021720 052777 004000 157464      BIS #SYSTST,@TXCSR ;ENTER SYSTST MODE
3638 021726 004537 007232          JSR R5,SETVEC ;SETUP VECTORS
3639 021732 022174          4$          ;RECEIVER VECTOR
3640 021734 007260          NO.BTRAPH    ;TRANSMITTER VECTOR
3641 021736 340 340          .BYTE 340,340 ;LEVEL
3642 021740 052777 000020 157436      BIS #RCVEN,@RXCSR ;TURN ON RECEIVER

```

RXDONE BIT RE-ARM INTERRUPT TEST

3643	021746	052777	000100	157430		BIS	#RINTEN,@RXCSR	:TURN ON INT. ENABLE
3644	021754	052777	000020	157430		BIS	#SEND,@TXCSR	:TURN ON TRANSMITTER
3645	021762				1\$:			
3646	021762	012737	000005	022012		MOV	#5,68\$:LOAD THE NUMBER
3647	021770	032777	004000	157416	66\$:	BIT	#TIMER,@TXDBUF	:CHECK THE TIMER BIT
3648	021776	001374				BNE	66\$:BR IF SET
3649	022000	032777	004000	157406	67\$:	BIT	#TIMER,@TXDBUF	:CHECK THE BIT
3650	022006	001774				BEQ	67\$:BR IF CLEAR
3651	022010	005327				DEC	(PC)+	:DECREMENT THE NUMBER
3652	022012	000005			68\$:	5		:OF TIMES TO REPEAT
3653	022014	001365				BNE	66\$:BR IF MORE TO GO
3654	022016	032777	000200	157366		BIT	#TXDONE,@TXCSR	:TEST TXDONE
3655	022024	001001				BNE	2\$:BR IF SET
3656	022026	104024				HLT	24	:TXDONE FAILED TO SET
3657	022030	005037	177776		2\$:	CLR	PS	:LOWER PROCESSOR STATUS
3658	022034	012777	000400	157352		MOV	#400,@TXDBUF	:LOAD TX BUFFER
3659	022042	105777	157344			TSTB	@TXCSR	
3660	022046	100375				BPL	-.4	
3661	022050	005077	157340			CLR	@TXDBUF	
3662	022054	105777	157332			TSTB	@TXCSR	
3663	022060	100375				BPL	-.4	
3664	022062	012777	001000	157324		MOV	#1000,@TXDBUF	
3665	022070				7\$:			
3666	022070	012737	000050	022120		MOV	#40.,73\$:LOAD THE NUMBER
3667	022076	032777	004000	157310	71\$:	BIT	#TIMER,@TXDBUF	:CHECK THE TIMER BIT
3668	022104	001374				BNE	71\$:BR IF SET
3669	022106	032777	004000	157300	72\$:	BIT	#TIMER,@TXDBUF	:CHECK THE BIT
3670	022114	001774				BEQ	72\$:BR IF CLEAR
3671	022116	005327				DEC	(PC)+	:DECREMENT THE NUMBER
3672	022120	000050			73\$:	40.		:OF TIMES TO REPEAT
3673	022122	001365				BNE	71\$:BR IF MORE TO GO
3674	022124	104022				HLT	22	:RECEIVER FAILED TO INTERRUPT
3675	022126	000445				BR	6\$:GET OUT OF TEST
3676	022130	005037	177776		3\$:	CLR	PS	:LOWER STATUS
3677	022134	012737	000005	022164		MOV	#5,78\$:LOAD THE NUMBER
3678	022142	032777	004000	157244	76\$:	BIT	#TIMER,@TXDBUF	:CHECK THE TIMER BIT
3679	022150	001374				BNE	76\$:BR IF SET
3680	022152	032777	004000	157234	77\$:	BIT	#TIMER,@TXDBUF	:CHECK THE BIT
3681	022160	001774				BEQ	77\$:BR IF CLEAR
3682	022162	005327				DEC	(PC)+	:DECREMENT THE NUMBER
3683	022164	000005			78\$:	5		:OF TIMES TO REPEAT
3684	022166	001365				BNE	76\$:BR IF MORE TO GO
3685	022170	104022				HLT	22	:DUP FAILED TO INTERRUPT AFTER SERVICING DONE
3686	022172	000423				BR	6\$:LEAVE IT
3687								
3688	022174	032777	000200	157202	4\$:	BIT	#RXDONE,@RXCSR	:SERVICE THE DONE BIT
3689	022202	012716	022130			MOV	#3\$, (SP)	:SETUP FOR 2ND PART OF TEST
3690	022206	004537	007232			JSR	R5,SETVEC	:SETUP NEW VECTORS
3691	022212	022222				5\$:RECEIVER VECTOR
3692	022214	007260				NO.BTRAP		:TRANSMITTER VECTOR
3693	022216	340	340			.BYTE	340,340	:LEVEL
3694	022220	000002				RTI		:GO FINISH TEST
3695	022222				5\$:			
3696	022222	052777	000400	157162		BIS	#MRESET,@TXCSR	:RESET THE DEVICE
3697	022230	004737	005044			JSR	PC,SMALL	:WAIT FOR RESET TO FINISH
3698	022234	012716	022242			MOV	#6\$, (SP)	:SETUP TO FINISH TEST

```

3699 022240 000002
3700 022242 012737 000340 177776 68: RTI ;LEAVE
3701 022250 012706 001150 MOV #340,PS ;RAISE PROCESSOR STATUS
3702 022254 104400 MOV #STACK,SP ;RESET THE STACK
3703 SCOPE ;SCOPE THIS TEST
3704 ;***** TEST 34 *****
3705 ;*TEST TO PROVE AN INTERRUPT REQUEST
3706 ;*IS GENERATED WHEN AN ABORT IS RECEIVED.
3707 ;*****
3708
3709 ;*****
3710 ; TEST 34
3711 ;*****
3712 ;*****
3713 ;*****
3714 022256 012737 000034 001226 TST34: MOV #34,@TSTNO
3715 022264 012737 022536 001216 MOV #TST35,NEXT
3716 022272 012737 000340 177776 MOV #340,PS ;LOCK OUT INTERRUPTS
3717 022300 052777 000400 157104 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3718 022306 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3719 022312 052777 004000 157072 BIS #SYSTST,@TXCSR ;ENTER SYS TST MODE
3720 022320 004537 007232 JSR R5,SETVEC ;SET UP VECTORS
3721 022324 022510 2$ ;RECEIVER
3722 022326 007260 NO.BTRAP ;TRANSMITTER
3723 022330 340 340 .BYTE 340,340 ;LEVEL
3724 022332 005037 177776 CLR PS ;LOWER PROCESSOR STATUS
3725 022336 052777 000120 157040 BIS #RCVEN!RINTEN,@RXCSR ;TURN ON RECEIVER AND INTERRUPT ENABLE
3726 022344 052777 000020 157040 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
3727 022352 012737 000005 022402 MOV #5,68$ ;LOAD THE NUMBER
3728 022360 032777 004000 157026 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3729 022366 001374 BNE 66$ ;BR IF SET
3730 022370 032777 004000 157016 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3731 022376 001774 BFC 67$ ;BR IF CLEAR
3732 022400 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3733 022402 000005 68$: 5 ;OF TIMES TO REPEAT
3734 022404 001365 BNE 66$ ;BR IF MORE TO GO
3735 022406 105777 157000 TSTB @TXCSR ;TEST DONE
3736 022412 100401 BMI 1$ ;BR IF SET
3737 022414 104024 HLT 24 ;DONE FAILED TO SET
3738 022416 012777 000400 156770 1$: MOV #T30M,@TXDBUF ;LOAD TX BUFFER
3739 022424 105777 156762 TSTB @TXCSR ;CHECK FOR DONE
3740 022430 100375 BPL .-4 ;BR IF NOT SET
3741 022432 005077 156756 CLR @TXDBUF ;PUSH OUT A CHARACTER
3742 022436 105777 156750 TSTB @TXCSR ;WAIT FOR DONE
3743 022442 100375 BPL .-4 ;BR IF NOT SET
3744 022444 052777 002000 156742 BIS #TABORT,@TXDBUF ;SET ABORT
3745 022452 012737 000113 022502 MOV #75.,73$ ;LOAD THE NUMBER
3746 022460 032777 004000 156726 71$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3747 022466 001374 BNE 71$ ;BR IF SET
3748 022470 032777 004000 156716 72$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3749 022476 001774 BEQ 72$ ;BR IF CLEAR
3750 022500 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3751 022502 000113 73$: 75. ;OF TIMES TO REPEAT
3752 022504 001365 BNE 71$ ;BR IF MORE TO GO
3753 022506 104022 HLT 22 ;RECEIVER FAILED TO INTERRUPT
3754 ;ON AN ABORTED MESSAGE
  
```

```

3755 022510 012737 000340 177776 2$: MOV #340,PS ;SET STATUS TO 7
3756 022516 012706 001150 MOV #STACK,SP ;RESET THE STACK
3757 022522 052777 000400 156662 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3758 022530 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3759 022534 104400 SCOPE
  
```

```

3760
3761 :***** TEST 35 *****
3762 :*THIS TEST IS AN AID FOR DEBUGGING CRC
3763 :*ERRORS. A CHARACTER IS LOADED INTO THE
3764 :*DUP AND PUSHED OUT BIT BY BIT WHILE
3765 :*ALLOWING THE OPERATOR TO MONITOR THE CRC
3766 :*CHARACTER AS IT IS GENERATED. THE DATA CHARACTER
3767 :*CAN ALSO BE CHANGED BY THE OPERATOR.
3768 :*PUT SW09=1 TO LOCK ON BITS. TO CONTINUE HIT
3769 :*ANY KEY ON THE TTY. AFTER 16 TIMES PUT DOWN SW09 TO LEAVE
3770 :NOTE: REMEMBER--IN SDLC A ONF IS A LOGIC LOW IN THE
3771 :CRC GENERATOR.
3772 :*****
  
```

```

:*****
: TEST 35
:*****
  
```

```

3779 022536 012737 000035 001226 TST35: MOV #35,@TSTNO
3780 022544 012737 002764 001216 MOV #.EOP,NEXT
3781 022552 052777 000400 156632 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3782 022560 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3783 022564 012737 102010 014534 MOV #CRC.CCITT,XPOLY ;LOAD THE POLYNOMIAL
3784 022572 012737 000125 022740 MOV #125,3$ ;LOAD DATA TO SOFTWARE BCC-CHANGE CHARACTER HERE
3785 022600 013737 022740 001252 MOV 3$,SAVR1
3786 022606 012737 177777 014540 MOV #-1,CALBCC ;CLEAR FOR SOFTWARE BCC
3787 022614 013737 014540 022742 MOV CALBCC,4$
3788 022622 005037 001242 CLR TEMP3
3789 022626 005037 001244 CLR TEMP4 ;CLEAR BIT COUNTER
3790 022632 005037 001246 CLR TEMP5
3791 022636 005077 156552 CLR @TXDBUF ;RESET TXDONE
3792 022642 052777 014000 156542 BIS #MMODE,@TXCSR ;ENTER MAINT MODE-PROGRAM CLOCKING
3793 022650 052777 000020 156526 BIS #RCVEN,@RXCSR ;TURN ON RECEIVER
3794 022656 052777 000020 156526 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
3795 022664 012777 000400 156522 MOV #TSOM,@TXDBUF
3796 022672 104412 000044 PKCLK ,36. ;PUSH OUT 2
3797 022676 013777 022740 156510 MOV 3$,@TXDBUF ;LOAD DATA
3798 022704 104412 000020 PKCLK ,16. ;PUSH OUT ANOTHER
3799 022710 104412 000002 PKCLK ,2 ;PUSH OUT A BIT
3800 022714 013737 001244 001254 1$: MOV TEMP4,SAVR2 ;SET UP TO TYPE
3801 022722 005237 001242 INC TEMP3
3802 022726 005237 001244 INC TEMP4 ;UPDATE BIT COUNTER
3803 022732 004537 014362 2$: JSR R5,SIMBCC ;CALCULATE SOFTWARE BCC BASED ON THESE PARAMETERS
3804 022736 000001 1 SHIFTS
3805 022740 000000 3$: .WORD 0 ;DATA
3806 022742 000000 4$: .WORD 0 ;PREVIOUS BCC
3807 022744 004737 023042 JSR PC,5$ ;CHECK TO SEE IF WE SHOULD WAIT FOR SCOPING
3808 022750 000241 CLC ;CLEAR FOR NEXT ROTATE
3809 022752 106037 022740 RORB 3$ ;SET UP THE NEXT BIT
3810 022756 013737 014540 022742 MOV CALBCC,4$ ;FOR THE SOFTWARE BCC
  
```

3811	022764	022737	000006	001244		CMP	#6,TEMP4	
3812	022772	001002				BNE	.+6	
3813	022774	005077	156414			CLR	@TXDBUF	
3814	023000	022737	000014	001242		CMP	#12.,TEMP3	
3815	023006	001003				BNE	12\$	
3816	023010	012777	001000	156376		MOV	#TEOM,@TXDBUF	
3817	023016	022737	000020	001244	12\$:	CMP	#16.,TEMP4	: ALL DONE WITH THE CHARACTER?
3818								: INCREASE THE COMPARE NUMBER TO OUTPUT
3819								: CRC TO THE RECEIVER
3820	023024	001331				BNE	1\$: BR IF MORE TO GO
3821	023026	052777	000400	156356		BIS	#MRFSET,@TXCSR	: RESET THE DEVICE
3822	023034	004737	005044			JSR	PC,SMALL	: WAIT FOR RESET TO FINISH
3823	023040	104400				SCOPE		: SCOPE THIS TEST
3824								
3825	023042	032777	001000	156132	5\$:	BIT	#SW09,@SWR	: SW09=1?
3826	023050	001432				BEQ	6\$: BR IF NO
3827	023052	013704	014540			MOV	CALBCC,R4	: THE DATA CHARACTER IS ALWAYS
3828	023056	012737	000001	001256		MOV	#1,SAVR3	: FOLLOWED BY A ZERO CHARACTER.THE
3829	023064	000241				CLC		: DATA BIT IN CRC SHOWS WHICH BIT OF
3830	023066	006004			11\$:	ROR	R4	: THE TWO CHARACTERS IS BEING GENERATED.
3831	023070	006137	001256			RCL	SAVR3	
3832	023074	103374				BCC	11\$	
3833	023076	105737	001246			TSTB	TEMP5	
3834	023102	001006				BNE	10\$	
3835	023104	104402	023601			TYPE	.EM17	: TYPE MSG
3836	023110	104402	023630			TYPE	.MH1	: TYPE HEADER
3837	023114	105137	001246			COMB	TEMP5	
3838	023120	104410			10\$:	CONVRT		
3839	023122	024476				DT1		
3840	023124	105777	156054		7\$:	TSTB	@TKCSR	: CHECK TTY DONE--GO SCOPE THE CRC GENERATOR
3841								: NOTE: THE LSB IS RIGHT JUSTIFIED ON THE TTY OUTPUT.!
3842	023130	100375				BPL	7\$: BR IF NOT YET
3843	023132	017701	156050			MOV	@TKDBR,R1	: READ THE BUFFER
3844	023136	000207			6\$:	RTS	PC	: RETURN
3845								
3846								

```

3847 023140 051377 047505 020115 EM1: .ASCIZ <377>/REOM BIT /
(1) 023153 377 054122 047504 EM2: .ASCIZ <377>/RXDONE BIT /
(1) 023170 051377 040530 047502 EM3: .ASCIZ <377>/RXABORT BIT /
(1) 023206 051377 040530 052103 EM4: .ASCIZ <377>/RXACTIVE BIT /
(1) 023225 377 054122 042504 EM5: .ASCIZ <377>/RXDERR BIT /
(1) 023242 051377 043505 051511 EM6: .ASCIZ <377>/REGISTER ERROR /
(1) 023263 377 054122 042504 EM7: .ASCIZ <377>/RXDERR OR OVERRUN /
(1) 023307 377 053117 051105 EM10: .ASCIZ <377>/OVERRUN BIT /
(1) 023325 377 040504 040524 EM11: .ASCIZ <377>/DATA COMPARE ERROR /
(1) 023352 042377 052101 020101 EM12: .ASCIZ <377>/DATA COMPARE ERROR IN SECONDARY MODE /
(1) 023421 377 051103 020103 EM13: .ASCIZ <377>/CRC CALCULATION ERROR /
(1) 023451 377 051522 046517 EM14: .ASCIZ <377>/RSOM BIT /
(1) 023464 040506 046111 042105 DH1: .ASCIZ /FAILED TO CLEAR /
(1) 023505 106 044501 042514 DH2: .ASCIZ /FAILED TO SET /
(1) 023524 052777 042516 050130 DH3: .ASCIZ <377>/UNEXPECTED /
(1) 023541 377 054105 042520 DH6: .ASCIZ <377>/EXPECTED FOUND REGISTER /
(1) 023601 377 051103 020103 EM17: .ASCIZ <377>/CRC GENERATOR STATUS /
(1) 023630 042377 052101 020101 MH1: .ASCIZ <377>/DATA CHAR DATA BIT IN CRC GEN. CRC FOR THIS BIT /
(1) 023723 015 052012 040522 EM15: .ASCIZ <15><12>/TRANSMITTER /
(1) 023743 015 051012 041505 EM16: .ASCIZ <15><12>/RECEIVER /
(1) 023760 047504 042516 041040 EM23: .ASCIZ /DONE BIT /
(1) 023773 106 044501 042514 EM22: .ASCIZ /FAILED TO SET /
(1) 024012 052777 042516 050130 EM20: .ASCIZ <377>/UNEXPECTED RECEIVER INTERRUPT /
(1) 024053 377 047125 054105 EM21: .ASCIZ <377>/UNEXPECTED TRANSMITTER INTERRUPT /
(1) 024117 040 051522 046517 EM24: .ASCIZ / RSOM BIT /
(1) 024133 106 044501 042514 DH4: .ASCIZ /FAILED TO INTERRUPT. /
(1) 024160 047111 042524 052522 DH5: .ASCIZ /INTERRUPTED UNEXPECTEDLY./
(1) 024211 040 042523 020124 DH7: .ASCIZ / SET PREMATURELY /
(1) 024234 .EVEN
  
```

```

(1)
(1)
(1)
(1) 024234 .ERRTAB:
(1) 024234 000000 0
(1) 024236 000000 0
(1) 024240 000000 0
(1) 024242 023140 EM1
(1) 024244 023505 DH2 ;HALT 1
(1) 024246 000000 0
(1)
(1) 024250 023153 EM2
(1) 024252 023505 DH2 ;HALT 2
(1) 024254 000000 0
(1)
(1) 024256 023242 EM6
(1) 024260 023541 DH6 ;HALT 3
(1) 024262 024514 DT6
(1)
(1) 024264 023140 EM1
(1) 024266 023464 DH1 ;HALT 4
(1) 024270 000000 0
(1)
(1) 024272 023170 EM3
(1) 024274 023505 DH2 ;HALT 5
(1) 024276 000000 0
(1)
  
```

(1)	024300	023206	EM4		
(1)	024302	023464	DH1	:HALT	6
(1)	024304	000000	0		
(1)					
(1)	024306	023225	EM5		
(1)	024310	023505	DH2	:HALT	7
(1)	024312	000000	0		
(1)					
(1)	024314	023325	EM11		
(1)	024316	000000	0	:HALT	10
(1)	024320	000000	0		
(1)					
(1)	024322	023170	EM3		
(1)	024324	023464	DH1	:HALT	11
(1)	024326	000000	0		
(1)					
(1)	024330	023352	EM12		
(1)	024332	000000	0	:HALT	12
(1)	024334	000000	0		
(1)					
(1)	024336	023263	EM7		
(1)	024340	023505	DH2	:HALT	13
(1)	024342	000000	0		
(1)					
(1)	024344	023206	EM4		
(1)	024346	023505	DH2	:HALT	14
(1)	024350	000000	0		
(1)					
(1)	024352	023307	EM10		
(1)	024354	023464	DH1	:HALT	15
(1)	024356	000000	0		
(1)					
(1)	024360	023524	DH3		
(1)	024362	023242	EM6	:HALT	16
(1)	024364	000000	0		
(1)					
(1)	024366	023421	EM13		
(1)	024370	000000	0	:HALT	17
(1)	024372	000000	0		
(1)					
(1)	024374	023451	EM14		
(1)	024376	023464	DH1	:HALT	20
(1)	024400	000000	0		
(1)					
(1)	024402	023451	EM14		
(1)	024404	023505	DH2	:HALT	21
(1)	024406	000000	0		
(1)					
(1)	024410	023743	EM16		
(1)	024412	024133	DH4	:HALT	22
(1)	024414	000000	0		
(1)					
(1)	024416	023743	EM16		
(1)	024420	024160	DH5	:HALT	23
(1)	024422	000000	0		
(1)					

(1)	024424	023760		EM23	
(1)	024426	023773		EM22	:HALT 24
(1)	024430	000000		0	
(1)					
(1)	024432	024012		EM20	
(1)	024434	000000		0	:HALT 25
(1)	024436	000000		0	
(1)					
(1)	024440	024053		EM21	
(1)	024442	000000		0	:HALT 26
(1)	024444	000000		0	
(1)					
(1)	024446	023723		EM15	
(1)	024450	024133		DH4	:HALT 27
(1)	024452	000000		0	
(1)					
(1)	024454	023140		EM1	
(1)	024456	023464		DH1	
(1)	024460	000000		0	:HALT 30
(1)					
(1)	024462	023153		EM2	
(1)	024464	023464		DH1	
(1)	024466	000000		0	:HALT 31
(1)					
(1)	024470	024117		EM24	
(1)	024472	023464		DH1	
(1)	024474	000000		0	:HALT 32
(1)					
(1)	024476	000003		DT1:	3
(1)	024500	006	021	.BYTE	6,17.
(1)	024502	001252		SAVR1	
(1)	024504	006	017	.BYTE	6,15.
(1)	024506	001254		SAVR2	
(1)	024510	006	002	.BYTE	6,2
(1)	024512	001256		SAVR3	
(1)					
(1)	024514	000003		DT6:	3
(1)	024516	006	004	.BYTE	6,4
(1)	024520	001262		SAVR5	
(1)	024522	006	002	.BYTE	6,2
(1)	024524	001260		SAVR4	
(1)	024526	006	002	.BYTE	6,2
(1)	024530	001256		SAVR3	
(1)	024532	000001		CORMAX:	
				.END	

CROSS REFERENCE TABLE -- USER SYMBOLS

RCRC71= 100000	767#													
RCVEN = 000020	730#	1811	1878	1960	2018	2081	2156	2200	2264	2282	2316	2365	2406	
	2462	2525	2593	2706	3356	3415	3499	3642	3725	3793				
REACT= 004000	723#	1815	1835	1855	1883	1897	1899	1940	2174	2237	2288	2381	2437	
REOM = 001000	740#	1983	1993	2046	2056	2104	2131							
RFSREG 004614	1488	1491#												
RESTR 003116	1151	1155	1163#											
RESOS = 104407	823#	1491												
RETURN 001214	672#	946*	1128*	1130	1163*	1200*	1203	1517*	1519	1552	1668*	1678*	1679	
RFLG 007060	1719#	1813	1881	1964	1990	2022	2053	2086	2108	2158	2203	2218	2266	
	2318	2367	2409	2420										
RING = 040000	720#	2797	2800	2903	3013	3123	3126							
RINTEN= 000100	728#	3357	3416	3500	3643	3725								
RPOKE 007020	1708#	1715	1821	1840	1896	1914	1928	1968	2028	2090	2112	2115	2162	
	2168	2207	2213	2222	2225	2231	2270	2276	2322	2328	2371	2374	2413	
	2424	2427	2430											
RSOM = 000400	741#	1827	1829	1846	1848	1909	1923	1935	2546					
RTS = 000004	732#	2881	2906	2922	3101	3126	3142	3293						
PUN 001314	712#	942*	1595	1598*	1599*	1606*	1607*							
RXC SR 001404	842#	1167	1614*	1618	1809	1834	1854	1876	1939	1960*	1971	1982	1992	
	2018*	2031	2045	2055	2081*	2093	2118	2130	2156*	2169	2200*	2232	2264*	
	2277	2282*	2283	2316*	2329	2335	2363	2404	2462*	2472	2483	2525*	2540	
	2542	2593*	2605	2613	2618	2706*	2735	2766*	2772	2872*	2878	2982*	2988	
	3092*	3098	3292*	3293*	3305*	3312*	3314*	3326	3330*	3356*	3357*	3389*	3415*	
	3416*	3455	3459*	3499*	3500*	3531	3642*	3643*	3688	3725*	3793*			
RXDBUF 001406	843#	1620*	1621*	1622	1623	1624	1827	1830	1846	1849	1909	1923	1935	
	1974	1983	1993	2034	2046	2056	2096	2104	2122	2131	2165	2170	2210	
	2214	2228	2233	2273	2278	2284	2325	2330	2334	2336	2375	2414	2431	
	2475	2486	2545	2608	2616	2621	2738	3460*						
RXDERR= 100000	736#	2180	2243	2294	2343	2376	2377	2432	2433					
RXDONE= 000200	727#	1822	1824	1841	1843	1890	1902	1905	1916	1918	1930	1932	1971	
	1986	1996	2031	2049	2059	2093	2118	2134	2177	2240	2291	2337	3455	
	3688													
SAVACT 001312	710#	1035*	1041*	1043	1044	1070*	1096							
SAVNUM 001313	711#	937*	1037*	1071*	1150*	1153*								
SAVPC 001266	697#	1325*	1523											
SAVPCA 001306	706#													
SAVRO 001250	690#	1334*	1339											
SAVROA 001270	699#													
SAVR1 001252	691#	1333*	1340	3785*	3847									
SAVR1A 001272	700#													
SAVR2 001254	692#	1332*	1341	3800*	3847									
SAVR2A 001274	701#													
SAVR3 001256	693#	1331*	1342	3828*	3831*	3847								
SAVR3A 001276	702#													
SAVR4 001260	694#	1330*	1343	3847										
SAVR4A 001300	703#													
SAVR5 001262	695#	1329*	1344	3847										
SAVR5A 001302	704#													
SAVSP 001264	696#													
SAVSPA 001304	705#													
SAVOS = 104406	821#	1451												
SCOPE = 104400	809#	1858	1943	2000	2063	2138	2183	2247	2299	2347	2385	2440	2493	
	2562	2633	2743	2845	2951	3061	3171	3209	3250	3332	3391	3462	3547	
	3601	3702	3759	3823										
SCOPE1 = 104401	811#	2552	2788	2814	2843	2894	2920	2949	3004	3030	3059	3114	3140	

CROSS REFERENCE TABLE -- USER SYMBOLS

SEND = 000020	3169	3303	3324											
SETFLG= 104413	757#	2463	2526	2594	2705	3358	3417	3502	3570	3644	3726	3794		
SETVEC 007232	831#	1013	1017	1023	1027									
SFLG 007152	1748#	3196	3228	3253	3287	3307	3351	3411	3465	3494	3566	3613	3638	
SHIFTS 001326	3690	3720												
	1735#	2024												
	782#	1690*	1698*	1714*	1820*	1839*	1895*	1913*	1927*	1967*	2027*	2089*	2111*	
	2114*	2161*	2167*	2206*	2212*	2221*	2224*	2230*	2269*	2275*	2321*	2327*	2370*	
	2373*	2412*	2423*	2426*	2429*									
SIMBCC 014362	2586	2643#	3803											
SMALL 005044	1540#	1808	1875	1959	2017	2080	2154	2198	2262	2314	2362	2403	2455	
	2492	2523	2554	2591	2698	2771	2828	2877	2934	2987	3044	3097	3154	
	3195	3227	3285	3349	3409	3493	3546	3565	3636	3697	3718	3758	3782	
	3822													
SPACNT= 004237	1361*	1382	1395*	1399#										
SRD = 002000	724#	3016	3126											
SRJMFL 001336	786#	1028	2980	3090										
STACK = 001150	585#	935	1117	1518	1548	3191	3207	3249	3304	3331	3388	3461	3544	
	3600	3701	3756											
STD = 000010	731#	2991	3016	3032	3101	3126	3142							
STJMFL 001334	785#	1024	2978	3088										
STPSYN= 000400	726#													
STUFCK 006746	1690#													
SV05 003744	1329#													
SWR 001202	659#	951	953*	965	969	1083	1091	1096	1101	1120	1182	1189	1210	
	1223	1440	1445	1494	1513	1515	1650	3825						
SWREG 000176	647#	953												
SW00 = 000001	566#	969	1083											
SW01 = 000002	565#	1650												
SW02 = 000004	564#													
SW03 = 000010	562#	1091												
SW04 = 000020	561#													
SW05 = 000040	560#													
SW06 = 000100	559#													
SW07 = 000200	558#													
SW08 = 000400	557#	1513												
SW09 = 001000	556#	1210	3825											
SW10 = 002000	555#	1515												
SW11 = 004000	554#	1189												
SW12 = 010000	553#	1223	1440											
SW13 = 020000	552#	1445												
SW14 = 040000	551#													
SW15 = 100000	550#													
SYSTST= 004000	763#	1502	2526	3200	3232	3350	3410	3498	3570	3637	3719			
TABORT= 002000	772#	3744												
TBLA 013516	2456	2457	2496#											
TBLB 014350	2577	2579	2580	2636#										
TCNFLG 001322	779#	1018	1019	1032	1072*	1768	1784	2763	2869	2975	3085	3279		
TCRCIN= 010000	770#													
TCRC7T= 020000	769#													
TEMP 006054	1365	1573#	2034*	2035										
TEMP1 001236	685#	1008	1036	1037	1038*	1076*	1077	1081*	1568	1962*	1977*	1978	2020*	
	2040*	2041	2083*	2099*	2100	2121*	2125*	2126	2169*	2174	2177	2277*	2283*	
	2288	2291	2329*	2335*	2337	2518*	2520*	2524	2557*	2559*	2560	2646*	2669*	
	2699*	2709*	2710	3311*										
TEMP2 001240	686#	999	1002	1043*	1046*	1050*	1077*	1570	1982*	1986	1992*	1996	2045*	

CROSS REFERENCE TABLE -- USER SYMBOLS

BRAYO = 177777	1#	1792#	1793	1796#	1860#	1861	1864#	1945#	1946	1949#	2002#	2003	2007#
	2065#	2066	2070#	2140#	2141	2144#	2184#	2185	2188#	2248#	2249	2252#	2300#
	2301	2304#	2348#	2349	2351#	2387#	2388	2392#	2442#	2443	2445#	2502#	2503
	2506#	2564#	2565	2567#	2683#	2684	2687#	2746#	2747	2752#	2852#	2853	2858#
	2958#	2959	2964#	3068#	3069	3074#	3177#	3178	3181#	3211#	3212	3215#	3263#
	3264	3269#	3333#	3334	3337#	3393#	3394	3397#	3476#	3477	3481#	3549#	3550
	3553#	3620#	3621	3624#	3703#	3704	3706#	3761#	3762	3772#			
SV - 000014	800#	809	811#	813#	815#	817#	819#	821#	823#	825#	827#	829#	831#
	833#												
= 024532	626#	627	630#	637#	642#	645#	648#	652#	654#	708#	709#	710#	711#
	712#	713#	779#	780#	785#	786#	867#	869#	870#	871#	873#	874#	875#
	877#	878#	879#	881#	882#	883#	885#	886#	887#	889#	890#	891#	893#
	894#	895#	897#	898#	899#	970	1100	1574#	1576#	1578#	1579#	1594	2588#
	2589#	2764	2870	2976	3086	3280	3374	3377	3432	3435	3518	3521	3660
	3663	3740	3743	3812	3847#								
.BEGIN 002660	1116#												
.CNVRT 004034	828	1352#											
.CONVR 004030	826	1351#											
.EOP 002764	1137#	3780											
.ERRTA 024234	1459	3847#											
.HLT 004350	633	1440#											
.INSTE 003516	818	1257#											
.INSTR 003412	816	1236#											
.INST1 003432	1240#	1260											
.MSG 003434	1238#	1241#											
.PARAM 003536	820	1268#											
.PFAIL 005050	631	936	1545#	1549									
.PKCLK 005006	830	1527#											
.RESOS 003776	824	1339#											
.SAVOS 003736	822	1325#											
.SCOPE 003160	810	1180#											
.SCOPI 003312	812	1210#											
.SETFL 004242	832	1408#	1420										
.START 001562	649	934#	946										
.TRPSR 004316	635	1428#											
.TRPTA 001344	808#	1433											
.TYPF 003336	814	1220#											

. ABS. 024532 000

ERRORS DETECTED: 0

CZDPCD,CZDPCD=CZDPXX.MAC,CZDPCD.P11
 RUN-TIME: 12 17 1 SECONDS
 RUN-TIME RATIO: 115/31=3.6
 CORE USED: 27K (53 PAGES)

DOCUMENT PAGES: 88