

DUP11

DUP11 SDLC DECMD TST
CZDPDD0

AH-8585D-MC

COPYRIGHT 75-80

FICHE 1 OF 1

JAN 1980

digital

MADE IN USA

IDENTIFICATION

PRODUCT CODE: AC-8584D-MC
PRODUCT NAME: CZDPDDJ DUP11 SDLC DECMD TST
PRODUCT DATE: JULY 1979
MAINTAINER: DIAGNOSTIC ENGINEERING

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

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

COPYRIGHT (C) 1975, 1979 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUSS
DEC	DECUS	DECTAPE	

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.

CZDPD TESTS THE ABILITY OF THE DEVICE TO RUN A LIMITED SDLC PROTOCOL AND LONG DATA PATTERNS. SPECIFIC DATA PATTERNS ARE RUN TO PROVE BIT-STUFF CAPABILITY. THE EIA DATA GATES ARE PROVEN AND THE PRIORITY LOGIC FUNCTIONS ARE CHECKED.

CZDPD TESTS ALL THE FUNCTIONS OF DECMODE AND DOES LONG AND SHORT DATA TESTS INTERNALLY AND OVER THE CABLE, IF ATTACHED. BCC, USING THE CRC16 POLYNOMIAL IS CHECKED, AND THE DEVICE IS TESTED RUNNING A LIMITED DDCMP PROTOCOL.

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 THE 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 DEVICES, 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/BELL 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 *HARD* 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 13: (INHIBIT ERROR PRINT OUT)

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

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

TSTNO 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 SEQUENTIALLY. 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.

RXCSR 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.

9.0 CHANGE HISTORY

NOTE: HISTORY STARTS WITH REV. D0

CZDPDD0 - IN RECEIVER INT. SERVICE ROUTINE OF TEST 12, LOC. 013174
CONTAINED 032715, SHOULD BE 32705.


```
541      ;*CZDPDD0 /<377>/DUP11 SDLC DECMD TST
542      ;*COPYRIGHT(C) 1975,1979, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
543      ;*-----
544
545      ;STARTING PROCEDURE
546      ;LOAD PROGRAM
547      ;LOAD ADDRESS 000200
548      ;PRESS START
549      ;PROGRAM WILL TYPE 'CZDPDD0 /<377>/DUP11 SDLC DECMD TST '
550      ;PROGRAM WILL TYPE 'R' TO INDICATE THAT TESTING HAS STARTED
551      ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
552      ;AND THEN RESUME TESTING
553
554
555      ;SWITCH REGISTER OPTIONS
556      ;-----
557
558      100000      SW15=100000      ;=1,HALT ON ERROR
559      040000      SW14=40000      ;=1,LOOP ON CURRENT TEST
560      020000      SW13=20000      ;=1,INHIBIT ERROR TYPEOUT
561      010000      SW12=10000      ;=1,DELETE TYPEOUT/BELL ON ERROR.
562      004000      SW11=4000      ;=1,INHIBIT ITERATIONS
563      002000      SW10=2000      ;=1,ESCAPE TO NEXT TEST ON ERROR
564      001000      SW09=1000      ;=1,LOOP WITH CURRENT DATA
565      000400      SW08=400      ;=1,LOOP ON ERROR
566      000200      SW07=200
567      000100      SW06=100
568      000040      SW05=40
569      000020      SW04=20
570      000010      SW03=10
571
572      000004      SW02=4
573      000002      SW01=2
574      000001      SW00=1
                    ;SELECT DUP'S DESIRED ACTIVE
                    ;NOTE:THIS MUST NOT EXCEED ORIGINAL COUNT
                    ;LOCK ON TEST SELECT
                    ;RESTART PROGRAM AT SELECTED TEST
                    ;ENTER PARAMETERS
```


575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624000000
000001
000002
000003
000004
000005
000006
000007177776
001150005746
005726
010046
012600
024646
022626100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

;REGISTER DEFINITIONS

;-----

R0=%0 ;GENERAL REGISTER
R1=%1 ;GENERAL REGISTER
R2=%2 ;GENERAL REGISTER
R3=%3 ;GENERAL REGISTER
R4=%4 ;GENERAL REGISTER
R5=%5 ;GENERAL REGISTER
SP=%6 ;PROCESSOR STACK POINTER
PC=%7 ;PROGRAM COUNTER

;LOCATION EQUIVALENCIES

;-----

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

;INSTRUCTION DEFINITIONS

;-----

PUSH1SP=5746 ;DECREMENT PROCESSOR STACK 1 WORD
POP1SP=5726 ;INCREMENT PROCESSOR STACK 1 WORD
PUSHR0=10046 ;SAVE R0 ON STACK
POPPO=12600 ;RESTORE R0 FROM STACK
PUSH2SP=24646 ;DECREMENT STACK TWICE
POP2SP=22626 ;INCREMENT STACK TWICE
.EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALLBIT15=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


```
625      ;*****
626      ;-----
627      ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
628      ;THE STANDARD 'TRAP CATCHER' IS PLACED
629      ;BETWEEN ADDRESS 0 TO ADDRESS 776.
630      ;IT LOOKS LIKE 'PC+2 HALT'.
631      ;-----
632      ;*****
633
634      000000      .=0
635      ;STANDARD INTERRUPT VECTORS
636      ;-----
637
638      000024      .=24
639      000024 005050      .PFAIL      ;POWER FAIL HANDLER
640      000026 000340      340      ;SERVICE AT LEVEL 7
641      000030 004350      .HLT      ;ERROR HANDLER
642      000032 000340      340      ;SERVICE AT LEVEL 7
643      000034 004316      .TRPSRV      ;GENERAL HANDLER DISPATCH SERVICE
644      000036 000340      340      ;SERVICE AT LEVEL 7
645
646      000040      .=40
647      000042 000000      0      ;SAVE FOR ACT-11 OR DDP2
648      000044 000000      0      ;RETURN ADDRESS IF UNDER ACT-11 OR DDP2
649      000046 003104      0      ;SAVE FOR ACT-11 OR DDP2
650      000052 000052      $ENDAD      ;FOR USE WITH ACT-11 OR DDP2
651      000052 000000      0      ;ACT-11 PROGRAM CHARACTERISTICS
652
653      000174      .=174
654      000174 000000      DISPREG:0      ;SOFTWARE DISPLAY REGISTER
655      000176 000000      SWREG: 0      ;SOFTWARE SWITCH REGISTER
656      000200      .=200
657      000200 000137 001562      JMP .START      ;GO TO START OF PROGRAM
658
659
660      001000      .=1000
661      001000 005377 055103 050104      MTITLE: .ASCIIZ <377><12>/CZDPDD0 /<377>/DUP11 SDLC DECMD TST /<377>
662      001200      .=1200
663      ;SWR AND LIGHTS
664      ;-----
665
666      001200 177570      DISPLAY: 177570      ;11/45 CONSOLE LIGHTS
667      001202 177570      SWR: 177570      ;INDIRECT POINTER TO SWITCH REGISTER
668
669      ;INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
670      ;-----
671
672      001204 177560      TKCSR: 177560      ;TELETYPE KEYBOARD CONTROL REGISTER
673      001206 177562      TKDBR: 177562      ;TELETYPE KEYBOARD DATA BUFFER
674      001210 177564      TPCSR: 177564      ;TELEPRINTER CONTROL REGISTER
675      001212 177566      TPDBR: 177566      ;TELEPRINTER DATA BUFFER
676
677      ;PROGRAM CONTROL PARAMETERS
678      ;-----
679
```


CZDPD-D0 MACY11 30A(1052) 25-JUL-79 15:19 PAGE 16
 CZDPDD.P11 25-JUL-79 15:01

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

SEQ 0015

680	001214	000000	RETURN: 0	;SCOPE ADDRESS FOR LOOP ON TEST
681	001216	000000	NEXT: 0	;ADDRESS OF NEXT TEST TO BE EXECUTED
682	001220	000000	LOCK: 0	;ADDRESS FOR LOCK ON CURRENT DATA
683	001222	000001	ICOUNT: 1	;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
684	001224	000000	LPCNT: 0	;NUMBER OF ITERATIONS COMPLETED
685	001226	000000	TSTNO: 0	;NUMBER OF TEST IN PROGRESS
686	001230	000000	PASCNT: 0	;NUMBER OF PASSES COMPLETED
687	001232	000000	ERRCNT: 0	;TOTAL NUMBER OF ERRORS
688	001234	000000	LSTERR: 0	;PC OF LAST ERROR CALL

;PROGRAM VARIABLES

690				
691				
692				
693	001236	000000	TEMP1: 0	;TEMPORARY STORAGE
694	001240	000000	TEMP2: 0	;TEMPORARY STORAGE
695	001242	000000	TEMP3: 0	;TEMPORARY STORAGE
696	001244	000000	TEMP4: 0	;TEMPORARY STORAGE
697	001246	000000	TEMP5: 0	;TEMPORARY STORAGE
698	001250	000000	SAVR0: 0	;R0 STORAGE
699	001252	000000	SAVR1: 0	;R1 STORAGE
700	001254	000000	SAVR2: 0	;R2 STORAGE
701	001256	000000	SAVR3: 0	;R3 STORAGE
702	001260	000000	SAVR4: 0	;R4 STORAGE
703	001262	000000	SAVR5: 0	;R5 STORAGE
704	001264	000000	SAVSP: 0	;STACK POINTER STORAGE
705	001266	000000	SAVPC: 0	;PROGRAM COUNTER STORAGE
706				
707	001270	000000	SAVR0A: 0	;R0 STORAGE
708	001272	000000	SAVR1A: 0	;R1 STORAGE
709	001274	000000	SAVR2A: 0	;R2 STORAGE
710	001276	000000	SAVR3A: 0	;R3 STORAGE
711	001300	000000	SAVR4A: 0	;R4 STORAGE
712	001302	000000	SAVR5A: 0	;R5 STORAGE
713	001304	000000	SAVSPA: 0	;STACK POINTER STORAGE
714	001306	000000	SAVPCA: 0	;PROGRAM COUNTER STORAGE
715				
716	001310	000001	DUPACTV: .BLKB 1	;DUP11'S SELECTED ACTIVE.
717	001311	000001	DUPNUM: .BLKB 1	;OCTAL NUMBER OF DUP11'S.
718	001312	000001	SAVACT: .BLKB 1	;ORIGINAL ACTV. DEVICES.
719	001313	000001	SAVNUM: .BLKB 1	;WORKABLE NUMBER.
720	001314	000001	RUN: .BLKB 1	;POINTER ONE PAST RUNNING DEVICE.
721		001316	.EVEN	
722	001316	001500	CREAM: DUP.MAP	;TABLE POINTER.


```
723
724                                     ;CONTROL REGISTER DEFINITIONS
725                                     ;-----
726                                     ;RXCSR BIT DEFINITIONS
727      100000      DSCA=BIT15      ;DATA SET CHANGE A
728      040000      RING=BIT14      ;RING
729      020000      CTS=BIT13      ;CLR TO SEND
730      010000      CARDET=BIT12    ;CARRIER DETECT
731      004000      RECACT=BIT11    ;REC ACTIVE
732      002000      SRD=BIT10      ;SEC REC DATA
733      001000      DSR=BIT9       ;DATA SET RDY
734      000400      STPSYN=BIT8    ;STRIP SYNC
735      000200      RXDONE=BIT7    ;REC DONE
736      000100      RINTEN=BIT6    ;REC INTR ENABLE
737      000040      DSINTE=BIT5    ;DSC INTR ENABLE
738      000020      RCVEN=BIT4     ;REC ENABLE
739      000010      STD=BIT3       ;SEC XMIT DATA
740      000004      RTS=BIT2       ;REQ TO SEND
741      000002      DTR=BIT1      ;DATA TERM RDY
742      000001      DSCB=BIT0      ;DATA SET CHANGE R
743                                     ;RXDBUF BIT DEFINITIONS
744      100000      RXDERR=BIT15    ;REC DATA ERROR
745      040000      OVRRUN=BIT14    ;OVERRUN ERROR
746      010000      CRCERR=BIT12   ;CRC ERROR
747      002000      RABORT=BIT10   ;REC ABORT
748      001000      REOM=BIT9      ;REC END OF MESSAGE
749      000400      RSOM=BIT8      ;REC START OF MESSAGE
750                                     ;PARCSR BIT DEFINITIONS
751      100000      DECMOD=BIT15    ;DEC MODE (DDCMP)
752      001000      CRCEN=BIT9      ;CRC ENABLE
753      010000      PRISEC=BIT12    ;PRI/SEC SELECT
754                                     ;TXCSR BIT DEFINITIONS
755      100000      TXDLAT=BIT15    ;TX DATA LATE
756      040000      MTDATA=BIT14    ;MAINT DATA OUT
757      020000      CLK=BIT13      ;CLK
758      010000      MMODEB=BIT12    ;MAINT MODE B
759      004000      MMODEA=BIT11    ;MAINT MODE A
760      002000      BITW=BIT10     ;BIT WINDOW INPUT
761      001000      TXACT=BIT9     ;TX ACTIVE
762      000400      MRESET=BIT8     ;MASTER RESET
763      000200      TXDONE=BIT7    ;XMIT DONE
764      000100      TXINTE=BIT6    ;XMIT DONE INTR ENABLE
765      000020      SEND=BIT4      ;SEND
766      000010      HDXEN=BIT3     ;HDX/FDX
767                                     ;TXCSR WRD DEFINITIONS
768      000000      USER=0          ;USER MODE
769      014000      MMODE=14000     ;MAINT INT MODE
770      010000      MEXT=10000     ;MAINT EXT MODE
771      004000      SYSTST=4000    ;SYSTEM TEST MODE
772
773                                     ;TXDBUF BIT DEFINITIONS
774                                     ;-----
775      100000      RCRC7T=BIT15
776      040000      RCRCIN=BIT14
777      020000      TCRC7T=BIT13
778      010000      TCRCIN=BIT12
```


CZDPD-DO
CZDPDD.P11MACY11 30A(1052) 25-JUL-79 15:19 PAGE 18
25-JUL-79 15:01

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

SEQ 0017

779		004000	TIMER=BIT11	:MAINTENANCE TIMER
780		002000	TABORT=BIT10	:TRANSMIT ABORT
781		001000	TEOM=BIT9	:TRANSMIT END OF MESSAGE
782		000400	TSOM=BIT8	:TRANSMIT START OF MESSAGE
783				
784			:MISC. PROGRAM DEFINITIONS	
785			-----	
786	001320	000000	PRIPTY: .WORD 0	
787	001322	000001	TCNFLG: .BLKB 1	
788	001323	000001	OPCLRJ: .BLKB 1	
789	001324	000000	DATA: .WORD 0	
790	001326	000000	SHIFTS: .WORD 0	
791	001330	000000	MIND: .WORD 0	
792	001332	000000	FLAG: .WORD 0	
793	001334	000001	STJMFL: .BLKW 1	
794	001336	000001	SRJMFL: .BLKW 1	
795				
796				

```
797
798                                     ;PROGRAM CONTROL FLAGS
799                                     ;-----
800
801 001340      000      INIFLG: .BYTE 0      ;PROGRAM INITIALIZATION FLAG
802 001341      000      ERRFLG: .BYTE 0      ;ERROR OCCURED FLAG
803 001342      000      LOKFLG: .BYTE 0      ;LOCK ON CURRENT TEST FLAG
804 001343      000      QV.FLG: .BYTE 0      ;QUICK VERIFY FLAG.
805                                           ;ON FIRST PASS OF EACH DUP11 ITERATIONS
806                                           ;WILL BE SUPPRESSED
807
808      000000      .EVEN
809                  $Y=0
810
811                                     ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
812                                     ;POINTERS TO SUBROUTINES CAN BE FOUND
813                                     ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS
814
815                                     ;*****
816                                     ;-----
817 001344      104400      .TRPTAB:
818      001344      003160      SCOPE=TRAP+0      ;CALL TO SCOPE LOOP AND ITERATION HANDLER
819      104401      .SCOPE
820 001346      003312      SCOP1=TRAP+1      ;CALL TO LOOP ON CURRENT DATA HANDLER
821      104402      .SCOP1
822 001350      003336      TYPE=TRAP+2      ;CALL TO TELETYPE OUTPUT ROUTINE
823      104403      .TYPE
824 001352      003412      INSTR=TRAP+3      ;CALL TO ASCII STRING INPUT ROUTINE
825      104404      .INSTR
826 001354      003516      INSTER=TRAP+4      ;CALL TO INPUT ERROR HANDLER
827      104405      .INSTER
828 001356      003536      PARAM=TRAP+5      ;CALL TO NUMERICAL DATA INPUT ROUTINE
829      104406      .PARAM
830 001360      003736      SAV05=TRAP+6      ;CALL TO REGISTER SAVE ROUTINE
831      104407      .SAV05
832 001362      003776      RES05=TRAP+7      ;CALL TO REGISTER RESTORE ROUTINE
833      104410      .RES05
834 001364      004030      CONVRT=TRAP+10      ;CALL TO DATA OUTPUT ROUTINE
835      104411      .CONVRT
836 001366      004034      CNVRT=TRAP+11      ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.
837      104412      .CNVRT
838 001370      005006      PKCLK=TRAP+12      ;CALL TO CLOCK ROUTINE
839      104413      .PKCLK
840 001372      004242      SETFLG=TRAP+13      ;CALL TO TELETYPE INPUT ROUTINE
841      104413      .SETFLG
842
843                                     ;-----
844                                     ;*****
```


CZDPD-DO
CZDPDD.P11MACY11 30A(1052)
25-JUL-79 15:01

25-JUL-79 15:19 PAGE 20

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

SEQ 0019

```

844                                     ;DUP11 VECTOR AND REGISTER INDIRECT POINTERS
845
846 001374 000000    DUPRVC: 0          ;POINTER TO DUP11 RECEIVER INTERRUPT VECTOR
847 001376 000000    DUPRPS: 0          ;POINTER TO DUP11 RECEIVER INTERRUPT SERVICE PS
848 001400 000000    DUPTVC: 0          ;POINTER TO DUP11 TRANSMITTER INTERRUPT VECTOR
849 001402 000000    DUPTPS: 0          ;POINTER TO DUP11 TRANSMITTER INTERRUPT SERVICE PS
850 001404 000000    RXCSR: 0           ;POINTER TO DUP11 RECEIVER STATUS REGISTER
851 001406 000000    RXDBUF: 0          ;POINTER TO DUP11 RECEIVER DATA BUFFER
852 001410 000000    PARCSR: 0          ;POINTER TO DUP11 PARAMETER STATUS REGISTER
853 001412 000000    TXCSR: 0           ;POINTER TO DUP11 TRANSMITTER STATUS REGISTER
854 001414 000000    TXDBUF: 0          ;POINTER TO DUP11 TRANSMITTER DATA BUFFER
855 001416 000000    DUPSEC: 0          ;POINTER TO DUP11 SECONDARY REGISTER SELECT REGISTER
856 001420 000000    HUPPSR: 0          ;POINTER TO PARAMETER STATUS HIGH BYTE
857 001422 000000    HUPRBF: 0          ;POINTER TO RECEIVER BUFFER HIGH BYTE
858 001424 000000    HUPRCR: 0          ;POINTER TO RECEIVER CONTROL REG HIGH BYTE
859 001426 000000    HUPTBF: 0          ;POINTER TO TRANSMITTER BUFFER HIGH BYTE
860 001430 000000    HUPTCR: 0          ;POINTER TO TRANSMITTER CONTROL REG HIGH BYTE
861
862
863                                     ;DUP11 CONTROL INDICATORS FOR CURRENT DUP11 UNDER TEST
864                                     ;-----
865
866 001432 000      MASK.A: .BYTE 000    ;LAST CHAR TO TEST AND PARITY MASK
867
868 001433 010      CLK.A: .BYTE 8.      ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR
869
870 001434 000000    L00.00: 000000      ;PARAMETERS
871

```

CZDPD-D0
CZDPDD.P11MACY11 30A(1052)
25-JUL-79 15:01

25-JUL-79 15:19 PAGE 21

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

SEQ 0020

;DUP11 STATUS TABLE AND ADDRESS ASSIGNMENTS

;-----;

```

872
873
874
875      001500      001500      . =1500
876      001500      000001      DUP.MAP:
877      001500      000001      DUPCR0: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 0
878      001502      000001      DUPTR0: .BLKW 1      ;VECTOR 'A' FOR DUP11 NUMBER 0
879      001504      000001      DUP0.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 0
880
881      001506      000001      DUPCR1: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 1
882      001510      000001      DUPTR1: .BLKW 1      ;VECTOR 'A' FOR DUP11 NUMBER 1
883      001512      000001      DUP1.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 1
884
885      001514      000001      DUPCR2: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 2
886      001516      000001      DUPTR2: .BLKW 1      ;VECTOR 'A' FOR DUP11 NUMBER 2
887      001520      000001      DUP2.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 2
888
889      001522      000001      DUPCR3: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 3
890      001524      000001      DUPTR3: .BLKW 1      ;VECTOR 'A' FOR DUP11 NUMBER 3
891      001526      000001      DUP3.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 3
892
893      001530      000001      DUPCR4: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 4
894      001532      000001      DUPTR4: .BLKW 1      ;VECTOR 'A' FOR DUP11 NUMBER 4
895      001534      000001      DUP4.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 4
896
897      001536      000001      DUPCR5: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 5
898      001540      000001      DUPTR5: .BLKW 1      ;VECTOR 'A' FOR DUP11 NUMBER 5
899      001542      000001      DUP5.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 5
900
901      001544      000001      DUPCR6: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 6
902      001546      000001      DUPTR6: .BLKW 1      ;VECTOR 'A' FOR DUP11 NUMBER 6
903      001550      000001      DUP6.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 6
904
905      001552      000001      DUPCR7: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 7
906      001554      000001      DUPTR7: .BLKW 1      ;VECTOR 'A' FOR DUP11 NUMBER 7
907      001556      000001      DUP7.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 7
908
909      001560      000000      DUP.END:      000000
910
911
912
913
914

```


[illegible]

DEFINITIONS

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

```
934  
935  
936  
937  
938  
939  
940  
941  
942 001562 012737 000340 177776 .START: MOV #340,PS ;LOCK OUT INTERRUPTS  
943 001570 012706 001150 MOV #STACK,SP ;SET UP STACK  
944 001574 012737 005050 000024 MOV #.PFail,@#24 ;SET UP POWER FAIL VECTOR  
945 001602 113737 001311 001313 MOV# DUPNUM,SAVNUM ;SAVE NUMBER OF DEVICES IN SYSTEM  
946 001610 005037 001230 CLR PASCNT ;CLEAR PASS COUNT  
947 001614 105037 001341 CLRB ERRFLG ;CLEAR ERROR FLAG  
948 001620 105037 001343 CLRB QV.FLG ;ZERO QUICK VERIFY FLAG  
949 001624 012737 001500 001316 MOV #DUP.MAP,CREAM ;GET MAP POINTER.  
950 001632 112737 000001 001314 MOV# #1,RUN ;POINT POINTER TO FIRST DEVICE.  
951 001640 005037 001232 CLR ERRCNT ;CLEAR ERROR COUNT  
952 001644 005037 001234 CLR LSTERR ;CLEAR LAST ERROR POINTER  
953 001650 012737 000001 001226 MOV #1,TSTNO ;SET UP FOR TEST 1  
954 001656 012737 001562 001214 MOV #.START,RETURN ;SET UP FOR POWER FAIL BEFORE  
955 ;TESTING STARTS  
956 001664 013746 000006 MOV @#6,-(SP) ;SAVE CURRENT VECTORS  
957 001670 013746 000004 MOV @#4,-(SP) ;  
958 001674 012737 001710 000004 MOV #12$,@#4 ;SETUP FOR TIMEOUT  
959 001702 005777 177274 TST @SWR ;REFERENCE HARDWARE SWITCH REG  
960 001706 000407 BR 13$ ;BR IF IT EXISTS  
961 001710 012737 000176 001202 12$: MOV #SWREG,SWR ;POINT TO SOFT SWR  
962 001716 012737 000174 001200 MOV #DISPREG,DISPLAY ;POINT TO SOFT DISPLAY REG  
963 001724 022626 CMP (SP)+,(SP)+ ;ADJUST STACK  
964 001726 012637 000004 13$: MOV (SP)+,@#4 ;RESTORE VECTORS  
965 001732 012637 000006 MOV (SP)+,@#6 ;  
966 001736 105737 001340 TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED  
967 001742 001401 BEQ 11$  
968 001744 000410 BR 6$  
969 001746 022737 003104 000042 11$: CMP #SENDAD,@#42 ;IF ACT-11 AUTO MODE,  
970 001754 001404 BEQ 6$ ;DON'T TYPE ID  
971 001756 104402 001000 TYPE ,MTITLE ;TYPE TITLE MESSAGE  
972 001762 105137 001340 COMB INIFLG ;IF NOT SET FLAG AND DO  
973 001766 105777 177210 6$: TSTB @SWR ;BIT7=1??  
974 001772 100002 BPL 10$  
975 001774 000137 002520 JMP 1$  
976 002000 10$:  
977 002000 032777 000001 177174 BIT #SW00,@SWR ;ENTER PARAMETERS  
978 002006 001002 BNE +6 ;YES  
979 002010 000137 002360 JMP 21$ ;NO  
980 002014 105137 001332 COMB FLAG  
981 002020 112737 000001 001340 MOV# #1,INIFLG ;SET TO MANUAL ENTRY  
982 002026 012700 001500 MOV #DUP.MAP,R0 ;CLR MAP  
983 002032 005020 68$: CLR (R0)+  
984 002034 020027 001560 CMP R0,#DUP.END ;DONE WITH MAP?  
985 002040 001374 BNE 68$ ;BR IF NO  
986 002042 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING  
987 002044 005473 MCSR ;MESSAGE  
988 002046 104405 PARAM ;CONVERT STRING  
989 002050 160000 160000 ;LOW LIMIT
```


990	002052	175500		175500	:HIGH LIMIT
991	002054	001500		DUPCR0	:STORE AT THIS LOCATION
992	002056	001	.BYTE	1	:MASK
993	002057	001	.BYTE	1	:HOW MANY TIMES + 2
994	002060	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING
995	002062	005512		MVEC	:MESSAGE
996	002064	104405		PARAM	:CONVERT STRING
997	002066	000300		300	:LOW LIMIT
998	002070	000770		770	:HIGH LIMIT
999	002072	001502		DUPTR0	:STORE AT THIS LOCATION
1000	002074	001	.BYTE	1	:MASK
1001	002075	001	.BYTE	1	:HOW MANY TIMES + 2
1002	002076	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1003	002100	005702		MPAR	:MESSAGE
1004	002102	104405		PARAM	:CONVERT STRING
1005	002104	000004		4	:LOW LIMIT
1006	002106	000007		7	:HIGH LIMIT
1007	002110	001240		TEMP2	:STORE AT THIS LOCATION
1008	002112	000	.BYTE	0	:MASK
1009	002113	001	.BYTE	1	:HOW MANY TIMES + 2
1010	002114	013737	001240 001320	MOV	TEMP2,PRTY :SAVE PRIORITY
1011	002122	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1012	002124	005647		MTOTAL	:MESSAGE
1013	002126	104405		PARAM	:CONVERT STRING
1014	002130	000001		1	:LOW LIMIT
1015	002132	000010		8.	:HIGH LIMIT
1016	002134	001236		TEMP1	:STORE AT THIS LOCATION
1017	002136	000	.BYTE	0	:MASK
1018	002137	001	.BYTE	1	:HOW MANY TIMES + 2
1019	002140	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1020	002142	005525		MJMPR	:MESSAGE
1021	002144	104413		SETFLG	:SET FLAG BASED UPON INPUT STRING
1022	002146	001323		OPCLRJ	:THIS FLAG
1023	002150	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1024	002152	005600		MTCN	:MESSAGE
1025	002154	104413		SETFLG	:SET FLAG BASED UPON INPUT STRING
1026	002156	001322		TCNFLG	:THIS FLAG
1027	002160	105737	001322	TSTB	TCNFLG
1028	002164	001410		BEQ	71\$
1029	002166	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1030	002170	005726		MSIJM	:MESSAGE
1031	002172	104413		SETFLG	:SET FLAG BASED UPON INPUT STRING
1032	002174	001334		STJMFL	:THIS FLAG
1033	002176	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1034	002200	005761		MSRJM	:MESSAGE
1035	002202	104413		SETFLG	:SET FLAG BASED UPON INPUT STRING
1036	002204	001336		SRJMFL	:THIS FLAG
1037	002206	105737	001323	TSTB	OPCLRJ
1038	002212	001403		BEQ	69\$
1039	002214	052737	100000 001504	BIS	#BIT15,DUP0.A
1040	002222	105737	001322	TSTB	TCNFLG
1041	002226	001403		BEQ	70\$
1042	002230	052737	040000 001504	BIS	#BIT14,DUP0.A
1043	002236	112737	000001 001312	MOVB	#1,SAVACT
1044	002244	113737	001236 001311	MOVB	TEMP1,DUPNUM
1045	002252	113737	001236 001313	MOVB	TEMP1,SAVNUM

1046	002260	005337	001236	65\$:	DEC	TEMP1	
1047	002264	001404			BEQ	64\$	
1048	002266	000261			SEC		
1049	002270	106137	001312		ROLB	SAVACT	
1050	002274	000771			BR	65\$	
1051	002276	113737	001312	001240	64\$:	MOVB	SAVACT,TEMP2 ;# OF TIMES
1052	002304	113737	001312	001310		MOVB	SAVACT,DUPACTV
1053	002312	000241			CLC		
1054	002314	106037	001240		RORB	TEMP2	
1055	002320	012700	001500		MOV	#DUPCR0,R0	
1056	002324	012701	001506		MOV	#DUPCR1,R1	
1057	002330	000241			67\$:	CLC	
1058	002332	106037	001240		RORB	TEMP2	
1059	002336	103051			BCC	66\$	
1060	002340	012011			MOV	(R0)+,(R1)	
1061	002342	062721	000010		ADD	#10,(R1)+	;CSR
1062	002346	012011			MOV	(R0)+,(R1)	
1063	002350	062721	000010		ADD	#10,(R1)+	;VECTOR
1064	002354	012021			MOV	(R0)+,(R1)+	;PARAMETERS
1065	002356	000764			BR	67\$	
1066	002360	012700	001500		21\$:	MOV	#DUP.MAP,R0 ;SETUP TO CLEAR MAP
1067	002364	005020			20\$:	CLR	(R0)+ ;CLEAR
1068	002366	020027	001560			CMP	R0,#DUP.END ;CHECK FOR FINISH
1069	002372	001374				BNE	20\$;BR IF MORE TO GO
1070	002374	012700	001500			MOV	#DUP.MAP,R0 ;SETUP TO DEFAULT
1071	002400	012710	160050			MOV	#160050,(R0) ;LOAD CSR
1072	002404	012760	000770	000002		MOV	#770,2(R0) ;LOAD VECTOR
1073	002412	012760	140026	000004		MOV	#140026,4(R0) ;LOAD PARAMETERS AND SYNC
1074	002420	112737	000005	001320		MOVB	#5,PRIORITY ;LOAD PRIORITY
1075	002426	012700	000001			MOV	#1,R0 ;SAVE CORE THIS WAY
1076	002432	110037	001310			MOVB	R0,DUPACTV ;PRESET PROGRAM CONTROLS
1077	002436	110037	001311			MOVB	R0,DUPNUM ;DITTO
1078	002442	110037	001312			MOVB	R0,SAVACT ;DITTO
1079	002446	110037	001313			MOVB	R0,SAVNUM ;DITTO
1080	002452	110037	001322			MOVB	R0,TCNFLAG ;DITTO
1081	002456	110037	001323			MOVB	R0,OPCLRJ ;DITTO
1082	002462				66\$:		
1083	002462	104402	006014		16\$:	TYPE	,XHEAD ;TYPE HEADER
1084	002466	012737	001500	001236		MOV	#DUP.MAP,TEMP1 ;SET POINTER
1085	002474	017737	176536	001240	5\$:	MOV	@TEMP1,TEMP2 ;SET DATA
1086	002502	001406				BEQ	1\$;ALL DONE WITH DATA
1087	002504	104410				CONVRT	
1088	002506	006042				XSTATQ	
1089	002510	062737	000002	001236		ADD	#2,TEMP1 ;UPDATE POINTER
1090	002516	000766				BR	5\$
1091	002520	032777	000001	176454	1\$:	BIT	#SW00,@SWR
1092	002526	001405				BEQ	7\$
1093	002530	005737	001332			TST	FLAG
1094	002534	001002				BNE	7\$
1095	002536	000137	002000			JMP	10\$
1096	002542	005037	001332		7\$:	CLR	FLAG
1097	002546	005737	000042			TST	@#42
1098	002552	001030				BNE	3\$
1099	002554	032777	000010	176420		BIT	#SW03,@SWR
1100	002562	001406				BEQ	3\$
1101	002564	104402	005413			TYPE	,MNEW ;TYPE THE MESSAGE.

1102	002570	005000			CLR	R0		:ZERO DATA LIGHTS
1103	002572	000000			HALT			:WAIT FOR USER TO TELL WHAT DEVICES TO RUN
1104	002574	127737	176402	001312	CMPB	@SWR,SAVACT		:IS THE NUMBER VALID?
1105	002602	101404			BLOS	2\$		:BR IF NUMBER IS OK.
1106	002604	104402	005254		TYPE	,MERR3		:TELL USER OF INVALID NUMBER.
1107	002610	000000			HALT			:STOP EVERY THING.
1108	002612	000776			BR	.-2		:RESTART THE PROGRAM AGAIN.
1109	002614	117737	176362	001310	2\$:	MOV	@SWR,DUPACTV	:GET NEW DEVICE PATTERN
1110	002622	113700	001310		MOV	DUPACTV,R0		:SHOW THE USER WHAT HE SELECTED.
1111	002626	042700	177400		BIC	#^C<377>,R0		:USE ONLY LOW BYTE.
1112	002632	000000			HALT			:CONTINUE DYNAMIC SWITCHES.
1113	002634	012700	000300		3\$:	MOV	#300,R0	:PREPARE TO CLEAR THE FLOATING
1114	002640	012701	000302		MOV	#302,R1		:VECTOR AREA. 300-776
1115	002644	010120			4\$:	MOV	R1,(R0)+	:START PUTTING 'PC+2 - HALT'
1116	002646	005021			CLR	(R1)+		:IN VECTOR AREA.
1117	002650	022021			CMP	(R0)+,(R1)+		:POP POINTERS
1118	002652	022700	001000		CMP	#1000,R0		:ALL DONE??
1119	002656	001372			BNE	4\$		:BR IF NO.
1120								
1121								
1122								
1123								
1124	002660	012737	000340	177776	.BEGIN:	MOV	#340,PS	:LOCK OUT INTERRUPTS
1125	002666	012706	001150		MOV	#STACK,SP		:SET UP STACK
1126	002672	005737	000042		TST	@#42		:IS PROGRAM UNDER MONITOR CONTROL
1127	002676	001023			BNE	2\$		:BR IF YES
1128	002700	032777	000004	176274	BIT	#BIT2,@SWR		:CHECK FOR LOCK ON TEST
1129	002706	001411			BEQ	1\$		:BR IF NO LOCK DESIRED.
1130	002710	104402	005312		TYPE	,MLOCK		:TYPE LOCK SELECTED.
1131	002714	012737	000240	003174	MOV	#NOP,TTST		:ADJUST SCOPE ROUTINE.
1132	002722	012737	000240	003176	MOV	#NOP,TTST+2		:SET UP TO LOCK
1133	002730	000406			BR	2\$		:CONTINUE ALONG.
1134	002732	013737	003306	003174	1\$:	MOV	BRW,TTST	:PREPARE NORMAL SCOPE ROUTINE
1135	002740	013737	003310	003176	MOV	BRX,TTST+2		:LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1136	002746	012737	006224	001214	2\$:	MOV	#CYCLE,RETURN	:START AT 'CYCLE' FIND WHICH DEVICE TO TEST
1137	002754	104402	005202		TYPE	,MR		:TYPE R
1138	002760	000177	176230		JMP	@RETURN		:START TESTING

```
1139                                     :END OF PASS
1140                                     :TYPE NAME OF TEST
1141                                     :UPDATE PASS COUNT
1142                                     :CHECK FOR EXIT TO ACT-11
1143                                     :RESTART TEST
1144
1145 002764 005037 001234      .EOP: CLR      LSTERR      :CLEAR LAST ERROR PC
1146 002770 105037 001341      CLR      ERRFLG      :CLEAR ERROR FLAG
1147 002774 005237 001230      INC      PASCNT      :UPDATE PASS COUNT
1148 003000 013777 001230 176172 MOV      PASCNT,@DISPLAY :DISPLAY PASS COUNT
1149 003006 104402 005157      TYPE      ,MEPASS      :TYPE END PASS
1150 003012 104402 005341      TYPE      ,MCSR      :TYPE CSR
1151 003016 104411 003130      CNVRT      ,XCSR      :SHOW IT
1152 003022 104402 005347      TYPE      ,MVECX      :TYPE VECTOR
1153 003026 104411 003136      CNVRT      ,XVEC      :SHOW IT
1154 003032 104402 005355      TYPE      ,MPASSX      :TYPE PASSES
1155 003036 104411 003144      CNVRT      ,XPASS      :SHOW IT
1156 003042 104402 005366      TYPE      ,MERRX      :TYPE ERRORS
1157 003046 104411 003152      CNVRT      ,XERR      :SHOW IT
1158 003052 105337 001313      DECB      SAVNUM      :ARE ALL DEVICES TESTED?
1159 003056 001017 000377 001343 BNE      RESTRT      :BR IF NO.
1160 003060 112737 000377 001343 MOV      #377,QV.FLG :SET THE QUICK VERIFY FLAG.
1161 003066 113737 001311 001313 MOV      DUPNUM,SAVNUM :RESTORE THE COUNT
1162 003074 013701 000042      MOV      @#42,R1      :CHECK FOR ACT-11 OR DDP
1163 003100 001406 000005      BEQ      RESTRT      :IF NOT, CONTINUE TESTING
1164 003102 000005 000005      RESET      :STOP THE SHOW--CLEAR THE WORLD
1165 003104
1166 003104 004711      $ENDAD: JSR      PC,(R1)
1167 003106 000240      NOP
1168 003110 000240      NOP
1169 003112 000240      NOP
1170 003114 000240      NOP
1171 003116 012737 006224 001214 RESTRT: MOV      #CYCLE,RETURN
1172 003124 000137 006224      JMP      CYCLE
1173 003130 000001      XCSR: 1
1174 003132 006 002      .BYTE 6,2
1175 003134 001404      RXCSR
1176 003136 000001      XVEC: 1
1177 003140 003 002      .BYTE 3,2
1178 003142 001374      DUPRVC
1179 003144 000001      XPASS: 1
1180 003146 006 002      .BYTE 6,2
1181 003150 001230      PASCNT
1182 003152 000001      XERR: 1
1183 003154 006 002      .BYTE 6,2
1184 003156 001232      ERRCNT
1185
1186                                     :SCOPE LOOP AND ITERATION HANDLER
1187
1188 003160 005037 001234      .SCOPE: CLR      LSTERR      :CLEAR LAST ERROR PC
1189 003164 010016 000000 176006 MOV      R0,(SP)      :SAVE R0 ON STACK
1190 003166 032777 040000      BIT      #BIT14,@SWR      :LOOP ON TEST?
1191 003174 001407 000437      TTST: BEQ      1$      :BR IF NO (IF LOCK SW01 = 1;THIS LOCATION = 240)
1192 003176 000437 000437      BR      3$      :GO TO 3$ (DITTO)
1193 003200 105777 176000      TSTB      @TKCSR      :KYBD DONE?
1194 003204 100034 000034      BPL      3$      :BR IF NO (LOCK: HIT A KEY ON TTY TO GO TO NEXT TEST)
```



```

1195 003206 017700 175774      MOV      @TKDBR,R0      ;CLR DONE BIT
1196 003212 000415      BR      2$      ;CONTINUE
1197 003214 032777 004000 175760 1$: BIT      #SW11,@SWR      ;DELETE ITERATION (QUICK PASS)?
1198 003222 001011      BNE      2$      ;BR IF YES
1199 003224 105737 001343      TSTB     QV.FLG      ;HAS FIRST PASS BEEN COMPLETED?
1200 003230 001406      BEQ      2$      ;BR IF QUICK VERIFY
1201 003232 005237 001224      INC      LPCNT      ;UPDATE ITERATION COUNTER
1202 003236 023737 001224 001222 CMP      LPCNT,ICOUNT ;ALL ITERATIONS DONE?
1203 003244 001014      BNE      3$      ;BR IF NOT YET
1204 003246 105037 001341      CLRB     ERRFLG      ;PREPARE FOR NEW TEST
1205 003252 005037 001224      CLR      LPCNT      ;START ICOUNT AT ZERO
1206 003256 005037 001220      CLR      LOCK
1207 003262 012737 000050 001222 MOV      #50,ICOUNT    ;RESET ITERATIONS
1208 003270 013737 001216 001214 MOV      NEXT,RETURN   ;GET NEXT TEST
1209 003276 011600      MOV      (SP),R0      ;POP R0 OFF STACK
1210 003300 022626      POP2SP      ;FAKE AN RTI
1211 003302 000177 175706      JMP      @RETURN      ;GO DO THE TEST
1212 003306 001407      BRW:    1407
1213 003310 000437      BRX:    437

1214
1215      ;CHECK FOR FREEZE ON CURRENT DATA
1216      -----
1217
1218 003312 032777 001000 175662 .SCOP1: BIT      #SW09,@SWR      ;IS SW09=1(SET)?
1219 003320 001405      BEQ      1$      ;BR IF NOT SET.
1220 003322 005737 001220      TST      LOCK
1221 003326 001402      BEQ      1$
1222 003330 013716 001220      MOV      LOCK,(SP)    ;GOTO THE ADDRESS IN LOCK.
1223 003334 000002      1$:      RTI      ;GO BACK.
1224
1225      ;TELETYPE OUTPUT ROUTINE
1226      -----
1227
1228 003336 010546      .TYPE:  MOV      R5,-(SP)      ;SAVE R5 ON THE STACK.
1229 003340 017605      MOV      @2(SP),R5      ;GET ADDRESS OF MESSAGE.
1230 003344 062766 000002 000002 ADD      #2,2(SP)      ;POP OVER ADDRESS.
1231 003352 032777 010000 175622 1$: BIT      #SW12,@SWR      ;INHIBIT ALL PRINT OUT??
1232 003360 001012      BNE      3$      ;BR IF NO PRINT OUT WANTED (SW12=1)
1233 003362 105715      TSTB     (R5)      ;IS NUMBER MINUS? (MSB=1(BIT7))
1234 003364 100002      BPL      2$      ;BR IF NUMBER IS PLUS
1235 003366 104402 005136      TYPE     ,MCRLF      ;TYPE A CR/LF!
1236 003372 105777 175612      2$:      TSTB     @TPCSR      ;TTY READY?
1237 003376 100375      BPL      2$      ;BR IF NO.
1238 003400 112577 175606      MOVB     (R5)+,@TPDBR ;PRINT CURRENT CHAR.
1239 003404 001362      BNE      1$      ;IF NOT ZERO KEEP PRINTING!
1240 003406 012605      3$:      MOV      (SP)+,R5      ;END OF OUTPUT. RESTORE R5
1241 003410 000002      RTI      ;GO HOME
1242
1243      -----
1244 003412 010346      .INSTR: MOV      R3,-(SP)      ;SAVE R3 ON STACK
1245 003414 010446      MOV      R4,-(SP)      ;SAVE R4 ON STACK
1246 003416 017637 000004 003434 MOV      @4(SP),.MSG
1247 003424 062766 000002 000004 ADD      #2,4(SP)
1248 003432 104402      .INST1: TYPE
1249 003434 000000      .MSG:    0
1250 003436 012704 006160      MOV      #INBUF,R4
  
```

1251	003442	012703	000007		MOV	#7,R3	
1252	003446	105777	175532	1\$:	TSTB	@TKCSR	
1253	003452	100375			BPL	1\$	
1254	003454	117714	175526		MOVB	@TKDBR,(R4)	
1255	003460	142714	000200		BICB	#200,(R4)	
1256	003464	122427	000015		CMPS	(R4)+,#15	
1257	003470	001417			BEQ	INSTR2	
1258	003472	105777	175512	2\$:	TSTB	@TPCSR	
1259	003476	100375			BPL	2\$	
1260	003500	017777	175502 175504		MOV	@TKDBR,@TPDBR	
1261	003506	005303			DEC	R3	
1262	003510	001356			BNE	1\$	
1263	003512	012604			MOV	(SP)+,R4	
1264	003514	012603			MOV	(SP)+,R3	
1265	003516	010346		.INSTE:	MOV	R3,-(SP)	
1266	003520	010446			MOV	R4,-(SP)	
1267	003522	104402	005132		TYPE	,MQM	
1268	003526	000741			BR	.INST1	
1269	003530	012604		INSTR2:	MOV	(SP)+,R4	:RESTORE R4
1270	003532	012603			MOV	(SP)+,R3	:RESTORE R3
1271	003534	000002			RTI		
1272							
1273						:CONVERT ASCII STRING TO OCTAL	
1274						:-----	
1275							
1276	003536	010546		.PARAM:	MOV	R5,-(SP)	
1277	003540	010446			MOV	R4,-(SP)	
1278	003542	016605	000004		MOV	4(SP),R5	
1279	003546	012537	003726		MOV	(R5)+,LOLIM	
1280	003552	012537	003730		MOV	(R5)+,HILIM	
1281	003556	012537	003732		MOV	(R5)+,DEVADR	
1282	003562	112537	003734		MOVB	(R5)+,LOBITS	
1283	003566	112537	003735		MOVB	(R5)+,ADRCNT	
1284	003572	010566	000004		MOV	R5,4(SP)	
1285	003576	005005		PARAM1:	CLR	R5	
1286	003600	012704	006160		MOV	#INBUF,R4	
1287	003604	122714	000015		CMPS	#15,(R4)	
1288	003610	001420			BEQ	PARERR	
1289	003612	121427	000060	1\$:	CMPS	(R4),#60	
1290	003616	002415			BLT	PARERR	
1291	003620	121427	000067		CMPS	(R4),#67	
1292	003624	003012			BGT	PARERR	
1293	003626	142714	000060		BICB	#60,(R4)	
1294	003632	152405			BISB	(R4)+,R5	
1295	003634	122714	000015		CMPS	#15,(R4)	
1296	003640	001406			BEQ	LIMITS	
1297	003642	006305			ASL	R5	
1298	003644	006305			ASL	R5	
1299	003646	006305			ASL	R5	
1300	003650	000760			BR	1\$	
1301	003652	104404		PARERR:	INSTR		
1302	003654	000750			BR	PARAM1	
1303							
1304						:TEST TO SEE IF NUMBER IS WITHIN LIMITS	
1305						:-----	
1306							


```
1307 003656 020537 003730      LIMITS: CMP      R5,HILIM
1308 003662 101373              BHI      PARERR
1309 003664 020537 003726      CMP      R5,LOLIM
1310 003670 103770              BLO      PARERR
1311 003672 133705 003734      BITB     LOBITS,R5
1312 003676 001365              BNE      PARERR
1313
1314                          ;STORE NUMBER AT SPECIFIED ADDRESS
1315
1316 003700 013704 003732      1$:      MOV      DEVADR,R4
1317 003704 010524              MOV      R5,(R4)+
1318 003706 062705 000002      ADD      #2,R5
1319 003712 105337 003735      DECB     ADCNT
1320 003716 001372              BNE      1$
1321 003720 012604              MOV      (SP)+,R4
1322 003722 012605              MOV      (SP)+,R5
1323 003724 000002              RTI
1324 003726 000000      LOLIM: 0
1325 003730 000000      HILIM: 0
1326 003732 000000      DEVADR: 0
1327 003734 000000      LOBITS: 0
1328                          ADCNT=LOBITS+1
1329
1330                          ;SAVE PC OF TEST THAT FAILED AND R0-R5
1331                          -----
1332
1333 003736 016637 000004 001266 .SAV05: MOV      4(SP),SAVPC      ;SAVE R7 (PC)
1334
1335                          ;SAVE R0-R5
1336
1337 003744 010537 001262      SV05:  MOV      R5,SAVR5      ;SAVE R5
1338 003750 010437 001260      MOV      R4,SAVR4      ;SAVE R4
1339 003754 010337 001256      MOV      R3,SAVR3      ;SAVE R3
1340 003760 010237 001254      MOV      R2,SAVR2      ;SAVE R2
1341 003764 010137 001252      MOV      R1,SAVR1      ;SAVE R1
1342 003770 010037 001250      MOV      R0,SAVR0      ;SAVE R0
1343 003774 000002              RTI                      ;LEAVE.
1344
1345                          ;RESTORE R0-R5
1346
1347 003776 013700 001250      .RES05: MOV      SAVR0,R0      ;RESTORE R0
1348 004002 013701 001252      MOV      SAVR1,R1      ;RESTORE R1
1349 004006 013702 001254      MOV      SAVR2,R2      ;RESTORE R2
1350 004012 013703 001256      MOV      SAVR3,R3      ;RESTORE R3
1351 004016 013704 001260      MOV      SAVR4,R4      ;RESTORE R4
1352 004022 013705 001262      MOV      SAVR5,R5      ;RESTORE R5
1353 004026 000002              RTI                      ;LEAVE
1354
1355
1356                          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
1357                          -----
1358
1359 004030 104402 005136      .CONVR: TYPE      ,MCRLF
1360 004034 010046      .CNVRT: MOV      R0,-(SP)
1361 004036 010146      MOV      R1,-(SP)
1362 004040 010346      MOV      R3,-(SP)
```

1363	004042	010446			MOV	R4,-(SP)
1364	004044	010546			MOV	R5,-(SP)
1365	004046	017601	000012		MOV	@12(SP),R1
1366	004052	062766	000002	000012	ADD	#2,12(SP)
1367	004060	012137	004234		MOV	(R1)+,WRDCNT
1368	004064	112137	004236	1\$:	MOVB	(R1)+,CHRCNT
1369	004070	112137	004237		MOVB	(R1)+,SPACNT
1370	004074	013137	004240		MOV	@(R1)+,BINWRD
1371	004100	013704	004240	2\$:	MOV	BINWRD,R4
1372	004104	113705	004236		MOVB	CHRCNT,R5
1373	004110	012700	006054		MOV	#TEMP,R0
1374	004114	010403		3\$:	MOV	R4,R3
1375	004116	042703	177770		BIC	#177770,R3
1376	004122	062703	000060		ADD	#060,R3
1377	004126	110320			MOVB	R3,(R0)+
1378	004130	000241			CLC	
1379	004132	006004			ROR	R4
1380	004134	000241			CLC	
1381	004136	006004			ROR	R4
1382	004140	000241			CLC	
1383	004142	006004			ROR	R4
1384	004144	005305			DEC	R5
1385	004146	001362			BNE	3\$
1386	004150	012703	006116		MOV	#MMDATA,R3
1387	004154	114023		4\$:	MOVB	-(R0),(R3)+
1388	004156	105337	004236		DECB	CHRCNT
1389	004162	001374			BNE	4\$
1390	004164	105737	004237		TSTB	SPACNT
1391	004170	001405			BEQ	6\$
1392	004172	112723	000040	5\$:	MOVB	#040,(R3)+
1393	004176	105337	004237		DECB	SPACNT
1394	004202	001373			BNE	5\$
1395	004204	105013		6\$:	CLRB	(R3)
1396	004206	104402	006116		TYPE	,MMDATA
1397	004212	005337	004234		DEC	WRDCNT
1398	004216	001322			BNE	1\$
1399	004220	012605			MOV	(SP)+,R5
1400	004222	012604			MOV	(SP)+,R4
1401	004224	012603			MOV	(SP)+,R3
1402	004226	012601			MOV	(SP)+,R1
1403	004230	012600			MOV	(SP)+,R0
1404	004232	000002			RTI	
1405	004234	000000			WRDCNT:	0
1406	004236	000000			CHRCNT:	0
1407		004237			SPACNT=CHRCNT+1	
1408	004240	000000			BINWRD:	0
1409						
1410						
1411						
1412						
1413						
1414						
1415						
1416	004242	017605	000000		.SETFLG:MOV	@(SP),R5
1417	004246	042737	000040	006160	BIC	#40,INBUF
1418	004254	122737	000116	006160	CMPB	#'N',INBUF

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

;IS IT 'N'?


```
1419 004262 001002      BNE      1$
1420 004264 105015      CLRB     (R5)      ;000
1421 004266 000406      BR       2$
1422 004270 122737 000131 006160 1$:  CMPB     #'Y',INBUF      ;IS IT 'Y' ?
1423 004276 001005      BNE      3$
1424 004300 112715 177777      MOVB     #-1,(R5)      ;377
1425 004304 062716 000002      2$:  ADD      #2,(SP)
1426 004310 000002      RTI
1427 004312 104404      3$:  INSTER   ;RETRY
1428 004314 000752      BR       .SETFLG
1429
1430
1431      ;TRAP DISPATCH SERVICE
1432      ;ARGUMENT OF TRAP IS EXTRACTED
1433      ;AND USED AS OFFSET TO OBTAIN POINTER
1434      ;TO SELECTED SUBROUTINE
1435
1436 004316 011646      .TRPSR: MOV      (SP),-(SP)      ;GET PC OF RETURN
1437 004320 162716 000002      SUB      #2,(SP)      ;=PC OF TRAP
1438 004324 017616 000000      MOV      @ (SP), (SP)      ;GET TRP
1439 004330 006316      TRPOK: ASL      (SP)      ;MULTIPLY TRAP ARG BY 2
1440 004332 042716 177001      BIC      #177001,(SP)      ;CLEAR UNWANTED BITS
1441 004336 062716 001344      ADD      #.TRPTAB,(SP)      ;POINTER TO SUBROUTINE ADDRESS
1442 004342 017616 000000      MOV      @ (SP), (SP)      ;SUBROUTINE ADDRESS
1443 004346 000136      JMP      @ (SP)+      ;GO TO SUBROUTINE
1444
1445      ;ERROR HANDLER
1446      ;-----
1447
1448 004350 032777 010000 174624 .HLT: BIT      #SW12,@SWR      ;BELL ON ERROR?
1449 004356 001406      BEQ      XBX      ;BR IF NO BELL
1450 004360 105777 174624      TSTB     @TPCSR      ;TTY READY.
1451 004364 100003      BPL      XBX      ;DON'T WAIT IF TTY NOT READY.
1452 004366 112777 000207 174616      MOVB     #207,@TPDBR      ;PUSH A BELL AT THE TTY.
1453 004374 032777 020000 174600 XBX: BIT      #SW13,@SWR      ;DELETE ERROR PRINT OUT?
1454 004402 001105      BNE      HALTS      ;BR IF NO PRINT OUT WANTED.
1455 004404 021637 001234      CMP      (SP),LSTERR      ;WAS THIS ERROR FOUND LAST TIME?
1456 004410 001404      BEQ      1$      ;BR IF YES
1457 004412 011637 001234      MOV      (SP),LSTERR      ;RECORD BEING HERE
1458 004416 105037 001341      CLRB     ERRFLG      ;PREPARE HEADER
1459 004422 104406      1$:  SAVO5      ;SAVE ALL PROC REGISTERS
1460 004424 011605      MOV      (SP),R5      ;GET THE PC OF ERROR
1461 004426 162705 000002      SUB      #2,R5      ;GET ADDRESS OF TRAP CALL
1462 004432 011504      MOV      (R5),R4      ;GET HLT INSTRUCTION
1463 004434 006304      ASL      R4      ;MULT BY TWO
1464 004436 061504      ADD      (R5),R4      ;DOUBLE IT
1465 004440 006304      ASL      R4      ;MULT AGAIN
1466 004442 042704 177001      BIC      #177001,R4      ;CLEAR JUNK
1467 004446 062704 023274      ADD      #.ERRTAB,R4      ;GET POINTER
1468 004452 012437 004566      MOV      (R4)+,ERRMSG      ;GET ERROR MESSAGE
1469 004456 012437 004600      MOV      (R4)+,DATAHD      ;GET DATA HEADRER
1470 004462 011437 004612      MOV      (R4),DATABP      ;GET DATA TABLE
1471 004466 105737 001341      TSTB     ERRFLG      ;TYPE HEADREER
1472 004472 001403      BEQ      TYPMSG      ;BR IF YES
1473 004474 005737 004612      TST      DATABP      ;DOES DATA TABLE EXIST?
1474 004500 001040      BNE      TYPDAT      ;BR IF YES.
```

1475	004502	104402	005136		TYPMSG: TYPE	,MCRLF	
1476	004506	104402	005136		TYPE	,MCRLF	
1477	004512	005737	001220		TST	LOCK	
1478	004516	001402			BEQ	1\$	
1479	004520	104402	005411		TYPE	,MASTEK	
1480	004524	104402	005377		1\$: TYPE	,MTSTN	
1481	004530	104411	005000		CNVRT	,XTSTN	:SHOW IT
1482	004534	104402	005466		TYPE	,MERRPC	:TYPE PC.
1483	004540	104411	004772		CNVRT	,ERTABO	:SHOW IT
1484	004544	104402	005136		TYPE	,MCRLF	:GIVE A CR/LF
1485	004550	112737	177777	001341	MOVB	#-1,ERRFLG	:NO MORE HEADER UNLESS NO DATA TABLE.
1486	004556	005737	004566		TST	ERRMSG	:IS THERE AN ERROR MESSAGE?
1487	004562	001402			BEQ	WRKO.FM	:BR IF NO.
1488	004564	104402			TYPE		:TYPE
1489	004566	000000			ERRMSG: 0		: ERROR MESSAGE
1490	004570				WRKO.FM:		:
1491	004570	005737	004600		TST	DATAHD	:DATA HEADER?
1492	004574	001402			BEQ	TYPDAT	:BR IF NO
1493	004576	104402			TYPE		:TYPE
1494	004600	000000			DATAHD: 0		: DATA HEADER
1495	004602	005737	004612		TYPDAT: TST	DATABP	:DATA TABLE?
1496	004606	001402			BEQ	RESREG	:BR IF NO.
1497	004610	104410			CONVRT		:SHOW
1498	004612	000000			DATABP: 0		: DATA TABLE
1499	004614	104407			RESREG: RES05		:RESTORE PROC REGISTERS
1500	004616	022737	003104	000042	HALTS: CMP	#\$ENDAD,@#42	:IF ACT-11 AUTO MODE--HALT!!
1501	004624	001403			BEQ	1\$	
1502	004626	005777	174350		TST	@SWR	:HALT ON ERROR?
1503	004632	100035			BPL	EXITER	:BR IF NO HALT ON ERROR
1504	004634	010046			1\$: PUSHRO		:SAVE RO
1505	004636	016600	000002		MOV	2(SP),RO	:SHOW ERROR PC IN DATA LIGHTS
1506	004642	013746	000004		MOV	4,-(SP)	:SAVE OLD TRAP
1507	004646	013746	000006		MOV	6,-(SP)	
1508	004652	012737	004710	000004	MOV	#22\$,4	:FORCE HALT IF TIME-OUT
1509	004660	012737	000340	000006	MOV	#340,5	:WHEN REFERENCING TXCSR
1510	004666	042777	014000	174516	BIC	#\$YSTST!MEXT,@TXCSR	
1511	004674	000000			HALT		:HALT
1512	004676	012637	000006		MOV	(SP)+,6	:RESTORE TRAP
1513	004702	012637	000004		MOV	(SP)+,4	
1514	004706	000406			BR	33\$	
1515	004710	000000			22\$: HALT		:HALT
1516	004712	022626			CMP	(SP)+,(SP)+	:POP STACK
1517	004714	012637	000006		MOV	(SP)+,6	:RESTORE TRAP
1518	004720	012637	000004		MOV	(SP)+,4	
1519	004724	012600			33\$: POPRO		:GET RO
1520	004726	005237	001232		EXITER: INC	ERRCNT	:UPDATE ERROR COUNT
1521	004732	032777	000400	174242	BIT	#\$W08,@SWR	:GOTO TOP OF TEST?
1522	004740	001007			BNE	1\$	:BR IF YES
1523	004742	032777	002000	174232	BIT	#\$W10,@SWR	:GOTO NEXT TEST?
1524	004750	001407			BEQ	2\$	:BR IF NO
1525	004752	013737	001216	001214	MOV	NEXT,RETURN	:SET FOR NEXT TEST
1526	004760	012706	001150		1\$: MOV	#\$STACK,SP	:RESET SP
1527	004764	000177	174224		JMP	@RETURN	:GOTO SPECIFIED TEST
1528	004770	000002			2\$: RTI		:RETURN
1529	004772	000001			ERTABO: 1		
1530	004774	006	002		.BYTE	6,2	

CZDPD-DO MACY11 30A(1052) 25-JUL-79 15:19 PAGE 34
 CZDPDD.P11 25-JUL-79 15:01 END OF PASS ROUTINE

SEQ 0033

```

1531 004776 001266          SAVPC
1532 005000 000001          XTSTN: 1
1533 005002 003 002          .BYTE 3,2
1534 005004 001226          TSTNO
1535 005006 017600 000000    .PKCLK: MOV @ (SP),R0      ;GET THE # OF TICKS TO POKE
1536 005012 062716 000002    ADD #2,(SP)      ;POP OVER THE #
1537 005016
1538 005016 052777 020000 174366 1$: BIS #CLK,@TXCSR      ;POKE CLOCK UP
1539 005024 005300          DEC R0              ;ARE WE DONE?
1540 005026 001405          BEQ 2$              ;YES-GO TO 2$
1541 005030 042777 020000 174354  BIC #CLK,@TXCSR      ;POKE CLOCK DOWN
1542 005036 005300          DEC R0              ;ARE WE DONE?
1543 005040 001366          BNE 1$              ;NO-REPEAT
1544 005042 000002    2$: RTI                    ;RETURN
1545
1546
1547          ;WAIT ROUTINE
1548 005044 000240          SMALL: NOP              ;STALL
1549 005046 000207          RTS PC                ;RETURN
1550
1551          ;POWER FAIL ROUTINE
1552
1553 005050 012737 005060 000024 .PFAIL: MOV #PWRUP,24      ;LOAD PFAIL VECTOR FOR POWER UP
1554 005056 000000          HALT
1555 005060 000005          PWRUP: RESET          ;WAIT TTY TO COME UP
1556 005062 012706 001150          MOV #STACK,SP      ;REINIT STACK POINTER
1557 005066 012737 005050 000024  MOV #.PFAIL,24      ;LOAD PFAIL VECTOR FOR POWER DOWN
1558 005074 104402          TYPE
1559 005076 005141          MPOWER
1560 005100 000177 174110          JMP @RETURN
1561          ;CLRVEC ROUTINE TO FILL COMMUNICATION VECTOR AREA WITH .+2,HALT
1562
1563 005104 012702 000300          CLRVEC: MOV #300,R2      ;R2 COMM VECTOR AREA ADRS
1564 005110 012701 000302          MOV #302,R1          ;INIT R1 WITH ADRS OF HALT
1565 005114 010122    1$: MOV R1,(R2)+      ;MOV .+2 TO PC
1566 005116 005022          CLR (R2)+          ;MOV HALT TO PC
1567 005120 022121          CMP (R1)+,(R1)+      ;INC TO NEXT VECTOR AREA
1568 005122 022701 000776          CMP #776,R1        ;END OF VECTOR AREA
1569 005126 001372          BNE 1$              ;NO
1570 005130 000207          RTS PC                ;RETURN
1571
1572
1573
1574 005132 020040 000077          MQM: .ASCIIZ / ?/
(2) 005136 005015 000          MCRLF: .ASCIIZ <15><12>
(2) 005141 377 053520 020122    MPOWER: .ASCIIZ <377>/PWR FAILED. /
(2) 005157 015 042777 042116    MEPASS: .ASCIIZ <15><377>/END PASS CZDPDC /
(2) 005202 051377 000          MR: .ASCIIZ <377>/R/
(2) 005205 377 051120 043517    MERR2: .ASCIIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT./
(2) 005254 044777 051516 043125    MERR3: .ASCIIZ <377>/INSUFFICIENT DATA!/
(2) 005300 052377 051505 020124    MTSTPC: .ASCIIZ <377>/TEST PC-/
(2) 005312 046377 041517 020113    MLOCK: .ASCIIZ <377>/LOCK ON SELECTED TEST/
(2) 005341 103 051123 020072    MCSRX: .ASCIIZ /CSR: /
(2) 005347 126 041505 020072    MVECX: .ASCIIZ /VEC: /
(2) 005355 120 051501 042523    MPASSX: .ASCIIZ /PASSES: /
(2) 005366 051105 047522 051522    MERRX: .ASCIIZ /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 DUP11'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
1575 006044 006 003 .BYTE 6,3
1576 006046 001236 TEMP1
1577 006050 006 002 .BYTE 6,2
1578 006052 001240 TEMP2
1579 .EVEN
1580
1581 006054 000000 TEMP: 0
1582 006116 .=. +40
1583 006116 000000 MDATA: 0
1584 006160 006160 .=. +40
1585 006160 000000 INBUF: 0
1586 006222 006222 .=. +40
1587 006222 000001 TRP.PC: .BLKW 1
1588
```



```
1589
1590
1591      ;ROUTINE USED TO 'CYCLE' THROUGH UP TO EIGHT DUP11'S
1592      ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
1593      ;AND RUNS THE SPECIFIED DUP11'S.  THIS ROUTINE *MUST*
1594      ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
1595      ;SETUP NECESSARY.
1596
1597
1598 006224 105737 001310      CYCLE:  TSTB   DUPACTV      ;ARE ANY DUP11'S TO BE TESTED?
1599 006230 001004      BNE     1$                      ;BR IF OK.
1600 006232 104402 005205      TYPE   ,MERR2          ;NO DUP11'S SELECTED!!
1601 006236 000000      HALT                                ;STOP THE SHOW.
1602 006240 000776      BR       .-2                    ;DISQUALIFY CONT. SW.
1603 006242 133737 001314 001310 1$:  BITB   RUN,DUPACTV  ;IS THIS ONE 'ACTIVE'
1604 006250 001020      BNE     2$                      ;BR IF GOOD ONE FOUND.
1605 006252 000241      CLC                                ;CLEAR PROC. CARRY BIT.
1606 006254 106137 001314      ROLB   RUN                ;UPDATE POINTER
1607 006260 105537 001314      ADCB   RUN                ;CATCH CARRY FROM RUN
1608 006264 062737 000006 001316      ADD   #6,CREAM      ;UPDATE ADDRESS POINTER.
1609 006272 022737 001560 001316      CMP   #DUP.END,CREAM
1610 006300 001360      BNE     1$                      ;KEEP GOING; NOT ALL TESTED FOR.
1611 006302 012737 001500 001316      MOV   #DUP.MAP,CREAM ;RESET ADDRESS POINTER.
1612 006310 000754      BR       1$                      ;KEEP LOOKING FOR ACTIVE DUP11
1613 006312 000241      CLC                                ;CLEAR PROC. CARRY.
1614 006314 106137 001314      ROLB   RUN                ;UPDATE POINTER.
1615 006320 105537 001314      ADCB   RUN                ;CATCH CARRY.
1616 006324 013700 001316      MOV   CREAM,R0            ;GET ADDRESS POINTER.
1617 006330 062737 000006 001316      ADD   #6,CREAM      ;UPDATE.
1618 006336 022737 001560 001316      CMP   #DUP.END,CREAM
1619
1620 006344 001003      BNE     3$                      ;ALL DONE?
1621 006346 012737 001500 001316      MOV   #DUP.MAP,CREAM ;BR IF NO.
1622 006354 012037 001404      MOV   (R0)+,RXCSR        ;RESTORE POINTER.
1623 006360 012037 001374      MOV   (R0)+,DUPRVC       ;LOAD SYSTEM CTRL. REG
1624 006364 012037 001434      MOV   (R0)+,L00.00       ;LOAD VECTOR
1625 006370 012700 000002      MOV   #2,R0              ;GET PARAMETERS
1626 006374 013737 001404 001424      MOV   RXCSR,HUPRCR  ;SAVE CORE THIS WAY!
1627 006402 005237 001424      INC     HUPRCR            ;GET CONTROL REG HIGH BYTE
1628 006406 013737 001424 001406      MOV   HUPRCR,RXDBUF ;GOT IT
1629 006414 005237 001406      INC     RXDBUF            ;GET RX CONTROL REG BUFFER
1630 006420 013737 001406 001416      MOV   RXDBUF,DUPSEC  ;GOT IT
1631 006426 013737 001406 001410      MOV   RXDBUF,PARCSR  ;GOT SECONDARY REG SELECT REG
1632 006434 013737 001406 001422      MOV   RXDBUF,HUPRBF ;GOT PARAMETER STATUS REGISTER
1633 006442 005237 001422      INC     HUPRBF            ;GET RX BUFFER HIGH BYTE
1634 006446 013737 001422 001420      MOV   HUPRBF,HUPPSR ;GOT IT
1635 006454 013737 001420 001412      MOV   HUPPSR,TXCSR  ;GOT PAR STATUS REG HIGH BYTE
1636 006462 005237 001412      INC     TXCSR            ;GET TX CONTROL REGISTER
1637 006466 013737 001412 001430      MOV   TXCSR,HUPTCR  ;GOT IT
1638 006474 005237 001430      INC     HUPTCR            ;GET TX CONTROL REG HIGH BYTE
1639 006500 013737 001430 001414      MOV   HUPTCR,TXDBUF ;GOT IT
1640 006506 005237 001414      INC     TXDBUF            ;BET TX BUFFER
1641 006512 013737 001414 001426      MOV   TXDBUF,HUPTBF  ;GOT IT
1642 006520 005237 001426      INC     HUPTBF            ;GET TX BUFFER HIGH BYTE
1643
1644 006524 013737 001374 001376      MOV   DUPRVC,DUPRPS  ;GOT IT
                                     ;RX VECTOR
```

1645	006532	060037	001376		ADD	R0,DUPRPS	;RX PRIORITY LEVEL
1646	006536	013737	001376	001400	MOV	DUPRPS,DUPTVC	
1647	006544	060037	001400		ADD	R0,DUPTVC	;TX VECTOR
1648	006550	013737	001400	001402	MOV	DUPTVC,DUPTPS	
1649	006556	060037	001402		ADD	R0,DUPTPS	;TX PRIORITY LEVEL
1650							
1651							
1652	006562	012700	001434		MOV	#L00.00,R0	;LOAD STAU 00-00
1653	006566	012701	001432		MOV	#MASK.A,R1	;PREPARE MASK.
1654	006572	012702	001433		MOV	#CLK.A,R2	;PREPARE CLOCKS
1655	006576	004737	006742		JSR	PC,FIX.00	;GO AND CALCULATE CONFIGURATION.
1656	006602	005737	000042		TST	@#42	
1657	006606	001050			BNE	4\$	
1658	006610	032777	000002	172364	BIT	#SW01,@SWR	;IF SW01=1,GET STARTING TEST #
1659	006616	001444			BEQ	4\$	
1660	006620	104402	005136		7\$: TYPE	,MCRLF	
1661	006624	104403			INSTR	;OUTPUT MESSAGE & GET INPUT STRING	
1662	006626	005377			MTSTN	;MESSAGE	
1663	006630	104405			PARAM	;CONVERT STRING	
1664	006632	000001			1	;LOW LIMIT	
1665	006634	001000			1000	;HIGH LIMIT	
1666	006636	001226			TSTNO	;STORE AT THIS LOCATION	
1667	006640	000			.BYTE 0	;MASK	
1668	006641	001			.BYTE 1	;HOW MANY TIMES + 2	
1669	006642	012700	007160		MOV	#TST1,R0	
1670	006646	022710	012737		5\$: CMP	#12737,(R0)	
1671	006652	001017			BNE	6\$	
1672	006654	023760	001226	000002	CMP	TSTNO,2(R0)	
1673	006662	001013			BNE	6\$	
1674	006664	022760	001226	000004	CMP	#TSTNO,4(R0)	
1675	006672	001007			BNE	6\$	
1676	006674	010037	001214		MOV	R0,RETURN	;SAVE PC
1677	006700	104402	005136		TYPE	,MCRLF	
1678	006704	104402	005202		TYPE	,MR	
1679	006710	000412			BR	8\$	
1680	006712	005720			6\$: TST	(R0)+	
1681	006714	020027	021464		CMP	R0,#TLAST+10	
1682	006720	001352			BNE	5\$	
1683	006722	104402	005132		TYPE	,MQM	
1684	006726	000734			BR	7\$	
1685							
1686	006730	012737	007160	001214	4\$: MOV	#TST1,RETURN	;PREPARE RETURN ADDRESS
1687	006736	000177	172252		8\$: JMP	@RETURN	;GO START TESTING.
1688							
1689	006742	011003			FIX.00: MOV	(R0),R3	;GET PARAMETERS.
1690	006744	000207			5\$: RTS	PC	;


```
1691
1692
1693
1694
1695 006746 012577 172422      SETVEC: MOV      (R5)+,ADUPRVC
1696 006752 012577 172422      MOV      (R5)+,ADUPTVC
1697 006756 112577 172414      MOV      (R5)+,ADUPRPS
1698 006762 112577 172414      MOV      (R5)+,ADUPTPS
1699 006766 000205              RTS      R5
1700 006770
1701 006770 104012      NO.ATRAP: HLT      12
1702 006772 000002      RTI
1703
1704 006774
1705 006774 104013      NO.BTRAP: HLT      13
1706 006776 000002      RTI
1707
1708 007000 010046      SIMBCC: MOV      R0,-(SP)
1709 007002 010146      MOV      R1,-(SP)
1710 007004 010246      MOV      R2,-(SP)
1711 007006 012537 001236      MOV      (R5)+,TEMP1
1712 007012 012537 001240      MOV      (R5)+,TEMP2
1713 007016 012537 001242      MOV      (R5)+,TEMP3
1714 007022 005037 007154      1$: CLR      BCCFBK
1715 007026 013700 001242      MOV      TEMP3,R0
1716 007032 006037 001240      ROR      TEMP2
1717 007036 005500      ADC      R0
1718 007040 032700 000001      BIT      #BIT0,R0
1719 007044 001402      BEQ      2$
1720 007046 005137 007154      COM      BCCFBK
1721 007052 013700 007152      2$: MOV      XPOLY,R0
1722 007056 005100      COM      R0
1723 007060 040037 007154      BIC      R0,BCCFBK
1724 007064 000241      CLC
1725 007066 006037 001242      ROR      TEMP3
1726 007072 013700 007154      MOV      BCCFBK,R0
1727 007076 013701 001242      MOV      TEMP3,R1
1728 007102 010102      MOV      R1,R2
1729 007104 040100      BIC      R1,R0
1730 007106 043702 007154      BIC      BCCFBK,R2
1731 007112 050200      BIS      R2,R0
1732 007114 043737 007152 001242      BIC      XPOLY,TEMP3
1733 007122 050037 001242      BIS      R0,TEMP3
1734 007126 005337 001236      DEC      TEMP1
1735 007132 001333      BNE      1$
1736 007134 013737 001242 007156      MOV      TEMP3,CALBCC
1737 007142 012602      MOV      (SP)+,R2
1738 007144 012601      MOV      (SP)+,R1
1739 007146 012600      MOV      (SP)+,R0
1740 007150 000205      RTS      R5
1741 007152 000000      XPOLY: 0
1742 007154 000000      BCCFBK: 0
1743 007156 000000      CALBCC: 0
1744      120001      CRC16=120001
1745      102010      CRC.CCITT=102010
1746
```

1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802

007160	012737	000001	001226
007166	012737	007264	001216
007174	012737	000340	177776
007202	052777	000400	172202
007210	004737	005044	
007214	004537	006746	
007220	006770		
007222	006774		
007224	340	340	
007226	012777	004100	172156
007234	012737	000340	177776
007242	000240		
007244	000240		
007246	000240		
007250	005077	172136	
007254	104400		
007256	012716	007250	
007262	000002		

```
***** TEST 1 *****
*PRIORITY INTERRUPT TEST.
*SET PROCESSOR STATUS TO PRIORITY 7
*AND VERIFY THAT THE DUP11 WILL NOT INTERRUPT.
*****

*****
: TEST 1
*****

TST1:  MOV    #1,@TSTNO
      MOV    #TST2,NEXT
      MOV    #340,PS          ;LOCK OUT INTERRUPTS
      BIS    #MRESET,@TXCSR   ;RESET THE DEVICE
      JSR    PC,SMALL         ;WAIT FOR RESET TO FINISH
      JSR    R5,SETVEC        ;SET UP VECTORS
      NO.ATRAP                ;VECTOR 'A'
      NO.BTRAP                ;VECTOR 'B'
      .BYTE  340,340          ;LEVEL
      MOV    #TXINTE!SYSTST,@TXCSR ;TURN ON DUP TX INT. ENABLEAND ENTER SYSTST MODE
      MOV    #340,PS          ;SET CPU PRIORITY--CHANGE HERE IF NOT = 5
      NOP
      NOP                    ;STALL
      NOP                    ;DITTO
      NOP                    ;DITTO
1$:    CLR    @TXCSR          ;DISABLE THE DUP11
      SCOPE
2$:    MOV    #1$,(SP)        ;SCOPE THIS TEST
      RTI                    ;SETUP FOR RETURN
      ;RETURN

***** TEST 2 *****
*PRIORITY INTERRUPT TEST.
*SET PROCESSOR STATUS TO PRIORITY 6
*AND VERIFY THAT THE DUP11 WILL NOT INTERRUPT.
*****

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

TST2:  MOV    #2,@TSTNO
      MOV    #TST3,NEXT
      CMPB   #5,PRTY         ;COMPARE REAL WITH NORMAL
      BNE    1$              ;BR IF NOT A MATCH
      MOV    #340,PS          ;LOCK OUT INTERRUPTS
      BIS    #MRESET,@TXCSR   ;RESET THE DEVICE
      JSR    PC,SMALL         ;WAIT FOR RESET TO FINISH
      JSR    R5,SETVEC        ;SET UP VECTORS
      NO.ATRAP                ;VECTOR 'A'
      NO.BTRAP                ;VECTOR 'B'
      .BYTE  340,340          ;LEVEL
```



```
1803 007342 012777 004100 172042      MOV    #TXINTE!SYSTST,@TXCSR ;TURN ON DUP TX INT. ENABLEAND ENTER SYSTST MODE
1804 007350 012737 000300 177776      MOV    #300,PS ;SET CPU PRIORITY--CHANGE HERE IF NOT = 5
1805 007356 000240                NOP    ;STALL
1806 007360 000240                NOP    ;DITTO
1807 007362 000240                NOP    ;DITTO
1808 007364 005077 172022      1$: CLR    @TXCSR ;DISABLE THE DUP11
1809 007370 104400                SCOPE ;SCOPE THIS TEST
1810 007372 012716 007364      2$: MOV    #1$,(SP) ;SETUP FOR RETURN
1811 007376 000002                RTI    ;RETURN
```

```
***** TEST 3 *****
*PRIORITY INTERRUPT TEST.
*SET PROCESSOR STATUS TO PRIORITY 5
*AND VERIFY THAT THE DUP11 WILL NOT INTERRUPT.
*****
```

```
*****
*
* TEST 3
*
```

```
*****
*****
```

```
1826 007400 012737 000003 001226      TST3: MOV    #3,@TSTNO
1827 007406 012737 007514 001216      MOV    #TST4,NEXT
1828 007414 122737 000005 001320      CMPB   #5,PRTY ;COMPARE REAL WITH NORMAL
1829 007422 001026                BNE     1$ ;BR IF NOT A MATCH
1830 007424 012737 000340 177776      MOV    #340,PS ;LOCK OUT INTERRUPTS
1831 007432 052777 000400 171752      BIS    #MRESET,@TXCSR ;RESET THE DEVICE
1832 007440 004737 005044                JSR    PC,SMALL ;WAIT FOR RESET TO FINISH
1833 007444 004537 006746                JSR    R5,SETVEC ;SET UP VECTORS
1834 007450 006770                NO.ATRAP ;VECTOR 'A'
1835 007452 006774                NO.BTRAP ;VECTOR 'B'
1836 007454 340 340                .BYTE 340,340 ;LEVEL
1837 007456 012777 004100 171726      MOV    #TXINTE!SYSTST,@TXCSR ;TURN ON DUP TX INT. ENABLEAND ENTER SYSTST MODE
1838 007464 012737 000240 177776      MOV    #240,PS ;SET CPU PRIORITY--CHANGE HERE IF NOT = 5
1839 007472 000240                NOP    ;STALL
1840 007474 000240                NOP    ;DITTO
1841 007476 000240                NOP    ;DITTO
1842 007500 005077 171706      1$: CLR    @TXCSR ;DISABLE THE DUP11
1843 007504 104400                SCOPE ;SCOPE THIS TEST
1844 007506 012716 007500      2$: MOV    #1$,(SP) ;SETUP FOR RETURN
1845 007512 000002                RTI    ;RETURN
```

```
***** TEST 4 *****
*PRIORITY INTERRUPT TEST.
*SET PROCESSOR STATUS TO PRIORITY 4
*AND VERIFY THAT THE DUP11 WILL INTERRUPT.
*****
```

```
*****
*
* TEST 4
*
```

```
*****
```

```
1859
1860 007514 012737 000004 001226 TST4: MOV #4,@TSTNO
1861 007522 012737 007632 001216 MOV #TST5,NEXT
1862 007530 122737 000005 001320 CMPB #5,PRIITY ;COMPARE REAL WITH NORMAL
1863 007536 001027 BNE 1$ ;BR IF NOT A MATCH
1864 007540 012737 000340 177776 MOV #340,PS ;LOCK OUT INTERRUPTS
1865 007546 052777 000400 171636 BIS #MRESET,@TXCSR ;RESET THE DEVICE
1866 007554 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
1867 007560 004537 006746 JSR R5,SETVEC ;SET UP VECTORS
1868 007564 006770 NO.ATRAP ;VECTOR 'A'
1869 007566 007624 2$ ;VECTOR 'B'
1870 007570 340 340 .BYTE 340,340 ;LEVEL
1871 007572 012777 004100 171612 MOV #TXINTE!SYSTST,@TXCSR ;TURN ON DUP TX INT. ENABLEAND ENTER SYSTST MODE
1872 007600 012737 000200 177776 MOV #200,PS ;SET CPU PRIORITY--CHANGE HERE IF NOT = 5
1873 007606 000240 NOP ;STALL
1874 007610 000240 NOP ;DITTO
1875 007612 000240 NOP ;DITTO
1876 007614 104013 HLT 13 ;DUP FAILED TO INTERRUPT-POSSBILY WRONG PRIORITY-CHANGE IF NOT 5
1877 007616 005077 171570 1$: CLR @TXCSR ;DISABLE THE DUP11
1878 007622 104400 SCOPE ;SCOPE THIS TEST
1879 007624 012716 007616 2$: MOV #1$, (SP) ;SETUP FOR RETURN
1880 007630 000002 RTI ;RETURN
1881
1882 ;***** TEST 5 *****
1883 ;*TEST TO PROVE THE HALF-DUPLEX FUNCTION
1884 ;*PROVE THAT THE RECEIVER WILL NOT RECOGNIZE
1885 ;*DATA IF SEND IS ASSERTED.
1886 ;*****
1887
1888 ;*****
1889 ;*
1890 ;* TEST 5
1891 ;*
1892 ;*****
1893 ;*****
1894 007632 012737 000005 001226 TST5: MOV #5,@TSTNO
1895 007640 012737 010172 001216 MOV #TST6,NEXT
1896 007646 105737 001322 TSTB TCN:LG
1897 007652 001530 BEQ 1$
1898 007654 012737 000340 177776 MOV #340,PS ;LOCK OUT INTERRUPTS
1899 007662 052777 000400 171522 BIS #MRESET,@TXCSR ;RESET THE DEVICE
1900 007670 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
1901 007674 052777 010010 171510 BIS #MEXT!HDXEN,@TXCSR ;ENTER MAINT EXT AND HALF-DUPLEX MODES
1902 007702 004537 006746 JSR R5,SETVEC ;SET UP VECTORS
1903 007706 010162 2$ ;RECEIVER
1904 007710 006774 NO.BTRAP ;TRANSMITTER
1905 007712 340 340 .BYTE 340,340 ;LEVEL
1906 007714 005037 177776 CLR PS ;LOWER PROC. STATUS
1907 007720 052777 000020 171456 BIS #RCVEN,@RXCSR ;TURN ON RECEIVER
1908 007726 052777 000100 171450 BIS #RINTEN,@RXCSR ;TURN ON INT. ENABLE
1909 007734 052777 000020 171450 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
1910 007742 012737 000005 007772 MOV #5,68$ ;LOAD THE NUMBER
1911 007750 032777 004000 171436 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
1912 007756 001374 BNE 66$ ;BR IF SET
1913 007760 032777 004000 171426 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
1914 007766 001774 BEQ 67$ ;BR IF CLEAR
```



```
1915 007770 005327          DEC      (PC)+      ;DECREMENT THE NUMBER
1916 007772 000005          68$:      5          ;OF TIMES TO REPEAT
1917 007774 001365          BNE      66$          ;BR IF MORE TO GO
1918 007776 105777 171410    TSTB     @TXCSR
1919 010002 100401          BMI      3$
1920 010004 104005          HLT      5          ;TXDONE FAILED TO SET
1921 010006 012777 000400 171400 3$:      MOV     #TSOM,@TXDBUF ;LOAD TX BUFFER
1922 010014 012737 000005 010044      MOV     #5,73$      ;LOAD THE NUMBER
1923 010022 032777 004000 171364 71$:     BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
1924 010030 001374          BNE      71$          ;BR IF SET
1925 010032 032777 004000 171354 72$:     BIT     #TIMER,@TXDBUF ;CHECK THE BIT
1926 010040 001774          BEQ      72$          ;BR IF CLEAR
1927 010042 005327          DEC      (PC)+      ;DECREMENT THE NUMBER
1928 010044 000005          73$:      5          ;OF TIMES TO REPEAT
1929 010046 001365          BNE      71$          ;BR IF MORE TO GO
1930 010050 105777 171336    TSTB     @TXCSR      ;CHECK FOR DONE
1931 010054 100401          BMI      4$
1932 010056 104000          HLT
1933
1934
1935 010060 005077 171330      4$:      CLR     @TXDBUF      ;LOAD A CHARACTER
1936 010064 105777 171322    TSTB     @TXCSR      ;CHECK FOR DONE
1937 010070 100375          BPL      -4          ;BR IF NOT SET
1938 010072 012777 001000 171314      MOV     #TEOM,@TXDBUF ;END THE MESSAGE
1939 010100 012737 000050 010130      MOV     #40,,78$      ;LOAD THE NUMBER
1940 010106 032777 004000 171300 76$:     BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
1941 010114 001374          BNE      76$          ;BR IF SET
1942 010116 032777 004000 171270 77$:     BIT     #TIMER,@TXDBUF ;CHECK THE BIT
1943 010124 001774          BEQ      77$          ;BR IF CLEAR
1944 010126 005327          DEC      (PC)+      ;DECREMENT THE NUMBER
1945 010130 000050          78$:      40.          ;OF TIMES TO REPEAT
1946 010132 001365          BNE      76$          ;BR IF MORE TO GO
1947 010134 012737 000340 177776 1$:      MOV     #340,PS      ;RAISE PROCESSOR STATUS
1948 010142 012706 001150      MOV     #STACK,SP      ;RESET STACK
1949 010146 052777 000400 171236      BIS     #MRESET,@TXCSR ;RESET THE DEVICE
1950 010154 004737 005044      JSR      PC,SMALL      ;WAIT FOR RESET TO FINISH
1951 010160 104400          SCOPE
1952
1953 010162 104007          2$:      HLT      7          ;RECEIVER INTERRUPTED AND SHOULD
1954 010164 012716 010134      MOV     #1$,(SP)      ;NOT HAVE--THIS IS HALF
1955 010170 000002          RTI
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
```

```
***** TEST 6 *****
*TEST OF THE DUP RUNNING A BINARY COUNT
*PATTERN WITHOUT A CRC CALCULATION
*****
```

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

```
1968 010172 012737 000006 001226 TST6:  MOV     #6,@TSTNO
1969 010200 012737 010632 001216      MOV     #TST7,NEXT
1970 010206 052777 000400 171176      BIS     #MRESET,@TXCSR ;RESET THE DEVICE
```

1971	010214	004737	005044		JSR	PC, SMALL	:WAIT FOR RESET TO FINISH
1972	010220	012737	000001	001236	MOV	#1, TEMP1	:LOAD DATA
1973	010226	005037	001240		CLR	TEMP2	:CLEAR EXPECTED
1974	010232	012737	000340	177776	MOV	#340, PS	:PS = 7
1975	010240	052777	004000	171144	BIS	#SYSTST, @TXCSR	:ENTER SYSTEM TEST MODE
1976	010246	004537	006746		JSR	R5, SETVEC	:LOAD INTERRUPT VECTORS
1977	010252	010446			11\$		:RECEIVER
1978	010254	010526			12\$		:TRANSMITER
1979	010256	340	340		.BYTE	340, 340	:LEVEL
1980	010260	052777	001000	171122	BIS	#CRCEN, @PARCSR	:TURN OFF CRC
1981	010266	052777	000020	171110	BIS	#RCVEN, @RXCSR	:TURN ON THE RECEIVER
1982	010274	052777	000100	171102	BIS	#RINTEN, @RXCSR	:TURN ON REC INTERRUPT ENABLE
1983	010302	105777	171104		1\$: TSTB	@TXCSR	:TEST FOR TX DONE
1984	010306	100375			BPL	1\$	:BR IF NOT SET
1985	010310	052777	000020	171074	2\$: BIS	#SEND, @TXCSR	:TURN ON SEND
1986	010316	012777	000400	171070	MOV	#TSOM, @TXDBUF	:TURN ON START OF MESSAGE
1987	010324	012737	000005	010354	MOV	#5, 68\$	:LOAD THE NUMBER
1988	010332	032777	004000	171054	66\$: BIT	#TIMER, @TXDBUF	:CHECK THE TIMER BIT
1989	010340	001374			BNE	66\$	:BR IF SET
1990	010342	032777	004000	171044	67\$: BIT	#TIMER, @TXDBUF	:CHECK THE BIT
1991	010350	001774			BEQ	67\$	:BR IF CLEAR
1992	010352	005327			DEC	(PC)+	:DECREMENT THE NUMBER
1993	010354	000005			68\$: 5		:OF TIMES TO REPEAT
1994	010356	001365			BNE	66\$	:BR IF MORE TO GO
1995	010360	105777	171026		3\$: TSTB	@TXCSR	:WAIT FOR DONE
1996	010364	100401			BMI	4\$	:BR IF SET
1997	010366	104000			HLT		:EXTERNAL CLOCKING STOPPED
1998	010370	005077	171020		4\$: CLR	@TXDBUF	:PUSH OUT DATA
1999	010374	052777	000100	171010	BIS	#TXINTE, @TXCSR	:TURN ON TRANSMITTER INT ENABLE
2000	010402	005037	177776		CLR	PS	:LOWER PROCESOR STATUS
2001	010406				5\$:		
2002	010406	012737	000040	010436	MOV	#32, 73\$	:LOAD THE NUMBER
2003	010414	032777	004000	170772	71\$: BIT	#TIMER, @TXDBUF	:CHECK THE TIMER BIT
2004	010422	001374			BNE	71\$	:BR IF SET
2005	010424	032777	004000	170762	72\$: BIT	#TIMER, @TXDBUF	:CHECK THE BIT
2006	010432	001774			BEQ	72\$	:BR IF CLEAR
2007	010434	005327			DEC	(PC)+	:DECREMENT THE NUMBER
2008	010436	000040			73\$: 32		:OF TIMES TO REPEAT
2009	010440	001365			BNE	71\$	:BR IF MORE TO GO
2010	010442	104001			HLT	1	:FAILED TO INTERRUPT IN TIME
2011	010444	104400			6\$: SCOPE		:SCOPE THIS TEST
2012							
2013							
2014							
2015							
2016							
2017							
2018	010446	017737	170734	001324	11\$: MOV	@RXDBUF, DATA	:GET THE REGISTER AND DATA
2019	010454	123737	001240	001324	CMPB	TEMP2, DATA	:CHECK IT
2020	010462	001401			BEQ	+4	:BR IF OK
2021	010464	104002			HLT	2	:COMPARISON ERROR
2022	010466	105237	001240		INCB	TEMP2	:COUNT UP EXPECTED
2023	010472	105737	001240		TSTB	TEMP2	:CHECK TO SEE IF DONE
2024	010476	001012			BNE	7\$	:BR IF NO
2025	010500	105777	170700		10\$: TSTB	@RXCSR	:CHECK FOR DONE
2026	010504	100375			BPL	10\$	:BR IF NOT YET


```
2027 010506 032777 001000 170672 BIT #REOM,@RXDBUF ;CHECK FOR END OF MSG
2028 010514 001001 BNE .+4 ;BR IF SET
2029 010516 104003 HLT 3 ;END OF MSG FAILED TO SET
2030 010520 012716 010444 MOV #6$,(SP) ;CRUNCH STACK
2031
2032 010524 000002 7$: RTI ;RETURN
2033
2034
2035 ;TRANSMITTER:
2036 010526 113777 001236 170660 12$: MOVB TEMP1,@TXDBUF ;LOAD THE TRANSMITTER BUFFER
2037 010534 105237 001236 INCB TEMP1 ;UP THE COUNT
2038 010540 122737 000377 001236 CMPB #377,TEMP1 ;ARE WE DONE
2039 010546 001026 BNE 13$ ;BR IF NO
2040 010550 012777 010560 170622 MOV #21$,@DUPTVC ;SETUP FOR NEXT PART
2041 010556 000422 BR 13$ ;LEAVE
2042 010560 012777 000377 170626 21$: MOV #377,@TXDBUF ;LOAD BUFFER
2043 010566 012777 010576 170604 MOV #22$,@DUPTVC ;SETUP NEXT PART
2044 010574 000413 BR 13$ ;LEAVE
2045 010576 012777 001000 170610 22$: MOV #TEOM,@TXDBUF ;SET END OF MSG
2046 010604 000240 NOP ;STALL
2047 010606 000240 NOP ;DITTO
2048 010610 042777 000120 170574 BIC #SEND!TXINTE,@TXCSR ;TURN OFF TRANSMITTER
2049 010616 012777 006774 170554 MOV #NO.BTRAP,@DUPTVC ;LOAD VECTOR
2050 010624 012716 010406 13$: MOV #5$,(SP) ;CRUNCH STACK
2051 010630 000002 RTI ;RETURNS
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
```

```
***** TEST 7 *****
*TEST OF THE DUP RUNNING A BINARY COUNT
*PATTERN WITH A CRC CALCULATION
*****
```

```
*****
*
* TEST 7
*
*****
*****
TST7: MOV #7,@TSTNO
MOV #TST10,NEXT
BIS #MRESET,@TXCSR ;RESET THE DEVICE
JSR PC,SMALL ;WAIT FOR RESET TO FINISH
CLR R1 ;CLEAR OUT DATA
MOV #CRC.CCITT,XPOLY ;SET UP THE POLYNOMIAL
MOV #-1,CALBCC ;SETUP FOR THE FIRST TIME
2065 010632 012737 000007 001226 16$: MOV CALBCC,20$ ;ALLOW FOR THE NEXT CHARACTER
2066 010640 012737 011432 001216 MOV R1,17$ ;LOAD DATA
2067 010646 052777 000400 170536 JSR R5,SIMBCC ;GO CALCULATE SOFTWARE BCC
2068 010654 004737 005044 8. ;BASED ON THESE PARAMETERS
2069 010660 005001 17$: .BLKW 1 ;DATA
2070 010662 012737 102010 007152 20$: .BLKW 1 ;PREVIOUS BCC
2071 010670 012737 177777 007156 INCB R1 ;INCREMENT DATA
2072 010676 013737 007156 010720 BNE 16$ ;BR IF MORE TO GO
2073 010704 010137 010716 MOV #1,TEMP1 ;LOAD DATA
2074 010710 004537 007000 CLR TEMP2 ;CLEAR EXPECTED
2075 010714 000010 MOV #340,PS ;PS = 7
2076 010716 000001
2077 010720 000001
2078 010722 105201
2079 010724 001364
2080 010726 012737 000001 001236
2081 010734 005037 001240
2082 010740 012737 000340 177776
```

```
2083 010746 052777 004000 170436 BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
2084 010754 004537 006746 JSR R5,SETVEC ;LOAD INTERRUPT VECTORS
2085 010760 011146 11$ ;RECEIVER
2086 010762 011214 12$ ;TRANSMITER
2087 010764 340 340 .BYTE 340,340 ;LEVEL
2088 010766 052777 000020 170410 BIS #RCVEN,@RXCSR ;TURN ON THE RECEIVER
2089 010774 052777 000100 170402 BIS #RINTEN,@RXCSR ;TURN ON REC INTERRUPT ENABLE
2090 011002 105777 170404 1$: TSTB @TXCSR ;TEST FOR TX DONE
2091 011006 100375 BPL 1$ ;BR IF NOT SET
2092 011010 052777 000020 170374 2$: BIS #SEND,@TXCSR ;TURN ON SEND
2093 011016 012777 000400 170370 MOV #TSOM,@TXDBUF ;TURN ON START OF MESSAGE
2094 011024 012737 000005 011054 MOV #5,68$ ;LOAD THE NUMBER
2095 011032 032777 004000 170354 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2096 011040 001374 BNE 66$ ;BR IF SET
2097 011042 032777 004000 170344 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2098 011050 001774 BEQ 67$ ;BR IF CLEAR
2099 011052 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2100 011054 000005 68$: S ;OF TIMES TO REPEAT
2101 011056 001365 BNE 66$ ;BR IF MORE TO GO
2102 011060 105777 170326 3$: TSTB @TXCSR ;WAIT FOR DONE
2103 011064 100401 BMI 4$ ;BR IF SET
2104 011066 104000 HLT ;EXTERNAL CLOCKING STOPPED
2105 011070 005077 170320 4$: CLR @TXDBUF ;PUSH OUT DATA
2106 011074 052777 000100 170310 BIS #TXINTE,@TXCSR ;TURN ON TRANSMITTER INT ENABLE
2107 011102 005037 177776 CLR PS ;LOWER PROCESOR STATUS
2108 011106 5$:
2109 011106 012737 000040 011136 MOV #32,,73$ ;LOAD THE NUMBER
2110 011114 032777 004000 170272 71$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2111 011122 001374 BNE 71$ ;BR IF SET
2112 011124 032777 004000 170262 72$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2113 011132 001774 BEQ 72$ ;BR IF CLEAR
2114 011134 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2115 011136 000040 73$: S ;OF TIMES TO REPEAT
2116 011140 001365 BNE 71$ ;BR IF MORE TO GO
2117 011142 104001 HLT 1 ;FAILED TO INTERRUPT IN TIME
2118 011144 104400 6$: SCOPE ;SCOPE THIS TEST
2119
2120
2121 ;INTERRUPT SERVICE ROUTINES
2122 -----
2123
2124 ;RECEIVER:
2125 011146 017737 170234 001324 11$: MOV @RXDBUF,DATA ;GET THE REGISTER AND DATA
2126 011154 123737 001240 001324 CMPB TEMP2,DATA ;CHECK IT
2127 011162 001401 BEQ .+4 ;BR IF OK
2128 011164 104002 HLT 2 ;COMPARISON ERROR
2129 011166 105237 001240 INCB TEMP2 ;COUNT UP EXPECTED
2130 011172 105737 001240 TSTB TEMP2 ;CHECK TO SEE IF DONE
2131 011176 001005 BNE 7$ ;BR IF NO
2132 011200 004537 006746 JSR R5,SETVEC ;YES--RESET THE VECTORS
2133 011204 011320 14$ ;RECEIVER
2134 011206 011214 12$ ;TRANSMITTER
2135 011210 340 340 .BYTE 340,340 ;LEVEL
2136
2137 011212 000002 7$: RTI ;RETURN
2138
```



```
2139
2140 ;TRANSMITTER:
2141 011214 113777 001236 170172 12$: MOVB TEMP1,@TXDBUF ;LOAD THE TRANSMITTER BUFFER
2142 011222 105237 001236 INCB TEMP1 ;UP THE COUNT
2143 011226 122737 000377 001236 CMPB #377,TEMP1 ;ARE WE DONE
2144 011234 001026 BNE 13$ ;BR IF NO
2145 011236 012777 011246 170134 MOV #21$,@DUPTVC ;SETUP FOR NEXT PART
2146 011244 000422 BR 13$ ;LEAVE
2147 011246 012777 000377 170140 21$: MOV #377,@TXDBUF ;LOAD BUFFER
2148 011254 012777 011264 170116 MOV #22$,@DUPTVC ;SETUP NEXT PART
2149 011262 000413 BR 13$ ;LEAVE
2150 011264 012777 001000 170122 22$: MOV #TEOM,@TXDBUF ;SET END OF MSG
2151 011272 000240 NOP ;STALL
2152 011274 000240 NOP ;DITTO
2153 011276 042777 000120 170106 BIC #SEND!TXINTE,@TXCSR ;TURN OFF TRANSMITTER
2154 011304 012777 006774 170066 MOV #NO.BTRAP,@DUPTVC ;LOAD VECTOR
2155 011312 012716 011106 13$: MOV #5$, (SP) ;CRUNCH STACK
2156 011316 000002 RTI ;RETURNS
2157
2158 011320 117737 170062 001324 14$: MOVB @RXDBUF,DATA ;GET FIRST PART OF CRC
2159 011326 105777 170052 TSTB @RXCSR ;WAIT FOR SECOND PART
2160 011332 100375 BPL -4 ;DITTO
2161 011334 017737 170046 001242 MOV @RXDBUF,TEMP3 ;GET THE REST OF THE CRC
2162 011342 113737 001242 001325 MOVB TEMP3,DATA+1 ;SET UP CRC CHARACTER
2163 011350 012716 011356 MOV #15$, (SP) ;SETUP FOR RETURN
2164 011354 000002 RTI ;RETURN
2165 011356 012737 000340 177776 15$: MOV #340,PS ;RAISE PS
2166 011364 005137 007156 COM CALBCC ;INVERT BCC
2167 011370 023737 007156 001324 CMP CALBCC,DATA ;COMPARE SOFTWARE AND HARDWARE BCC
2168 011376 001401 BEQ +4 ;BR IF OK
2169 011400 104004 HLT 4 ;BCC COMPARISON ERROR
2170 011402 032737 010000 001242 BIT #CRCERR,TEMP3 ;CHECK THE ERROR BIT
2171 011410 001401 BEQ +4 ;BR IF NO ERROR
2172 011412 104004 HLT 4 ;BCC ERROR--RECEIVER DOESN'T
2173 ;AGREE WITH WHAT TX SENT
2174 011414 052777 000400 167770 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2175 011422 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2176 011426 000137 011144 JMP 6$ ;LEAVE
2177
2178 ;***** TEST 10 *****
2179 ;*TEST OF THE DUP RUNNING A BINARY COUNT
2180 ;*PATTERN WITH A CRC CALCULATION
2181 ;*****
2182
2183 ;*****
2184 ;*
2185 ;* TEST 10
2186 ;*
2187 ;*****
2188 ;*****
```

```
2189 011432 012737 000010 001226 TST10: MOV #10,@TSTNO
2190 011440 012737 012240 001216 MOV #TST11,NEXT
2191 011446 052777 000400 167736 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2192 011454 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2193 011460 105737 001322 TSTB TCNFLAG
2194 011464 001532 BEQ 6$
```

```

2195 011466 005001 CLR R1 ;CLEAR OUT DATA
2196 011470 012737 102010 007152 MOV #CRC.CCITT,XPOLY ;SET UP THE POLYNOMIAL
2197 011476 012737 177777 007156 MOV #-1,CALBCC ;SETUP FOR THE FIRST TIME
2198 011504 013737 007156 011526 16$: MOV CALBCC,20$ ;ALLOW FOR THE NEXT CHARACTER
2199 011512 010137 011524 MOV R1,17$ ;LOAD DATA
2200 011516 004537 007000 JSR R5,SIMBCC ;GO CALCULATE SOFTWARE BCC
2201 011522 000010 8. ;BASED ON THESE PARAMETERS
2202 011524 000001 17$: .BLKW 1 ;DATA
2203 011526 000001 20$: .BLKW 1 ;PREVIOUS BCC
2204 011530 105201 INCB R1 ;INCREMENT DATA
2205 011532 001364 BNE 16$ ;BR IF MORE TO GO
2206 011534 012737 000001 001236 MOV #1,TEMP1 ;LOAD DATA
2207 011542 005037 001240 CLR TEMP2 ;CLEAR EXPECTED
2208 011546 012737 000340 177776 MOV #340,PS ;PS = 7
2209 011554 052777 010000 167630 BIS #MEXT,@TXCSR
2210 011562 004537 006746 JSR R5,SETVEC ;LOAD INTERRUPT VECTORS
2211 011566 011754 11$ ;RECEIVER
2212 011570 012022 12$ ;TRANSMITER
2213 011572 340 340 .BYTE 340,340 ;LEVEL
2214 011574 052777 000020 167602 BIS #RCVEN,@RXCSR ;TURN ON THE RECEIVER
2215 011602 052777 000100 167574 BIS #RINTEN,@RXCSR ;TURN ON REC INTERRUPT ENABLE
2216 011610 105777 167576 1$: TSTB @TXCSR ;TEST FOR TX DONE
2217 011614 100375 BPL 1$ ;BR IF NOT SET
2218 011616 052777 000020 167566 2$: BIS #SEND,@TXCSR ;TURN ON SEND
2219 011624 012777 000400 167562 MOV #TSOM,@TXDBUF ;TURN ON START OF MESSAGE
2220 011632 012737 000005 011662 MOV #5,68$ ;LOAD THE NUMBER
2221 011640 032777 004000 167546 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2222 011646 001374 BNE 66$ ;BR IF SET
2223 011650 032777 004000 167536 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2224 011656 001774 BEQ 67$ ;BR IF CLEAR
2225 011660 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2226 011662 000005 68$: 5 ;OF TIMES TO REPEAT
2227 011664 001365 BNE 66$ ;BR IF MORE TO GO
2228 011666 105777 167520 3$: TSTB @TXCSR ;WAIT FOR DONE
2229 011672 100401 BMI 4$ ;BR IF SET
2230 011674 104000 HLT ;EXTERNAL CLOCKING STOPPED
2231 011676 005077 167512 4$: CLR @TXDBUF ;PUSH OUT DATA
2232 011702 052777 000100 167502 BIS #TXINTE,@TXCSR ;TURN ON TRANSMITTER INT ENABLE
2233 011710 005037 177776 CLR PS ;LOWER PROCESOR STATUS
2234 011714 5$:
2235 011714 012737 000040 011744 MOV #32..73$ ;LOAD THE NUMBER
2236 011722 032777 004000 167464 71$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2237 011730 001374 BNE 71$ ;BR IF SET
2238 011732 032777 004000 167454 72$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2239 011740 001774 BEQ 72$ ;BR IF CLEAR
2240 011742 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2241 011744 000040 73$: 32. ;OF TIMES TO REPEAT
2242 011746 001365 BNE 71$ ;BR IF MORE TO GO
2243 011750 104001 HLT 1 ;FAILED TO INTERRUPT IN TIME
2244 011752 104400 6$: SCOPE ;SCOPE THIS TEST
2245
2246
2247 ;INTERRUPT SERVICE ROUTINES
2248 ;-----
2249
2250 ;RECEIVER:

```



```
2251 011754 017737 167426 001324 11$: MOV @RXDBUF,DATA ;GET THE REGISTER AND DATA
2252 011762 123737 001240 001324 CMPB TEMP2,DATA ;CHECK IT
2253 011770 001401 BEQ .+4 ;BR IF OK
2254 011772 104002 HLT 2 ;COMPARISON ERROR
2255 011774 105237 001240 INCB TEMP2 ;COUNT UP EXPECTED
2256 012000 105737 001240 TSTB TEMP2 ;CHECK TO SEE IF DONE
2257 012004 001005 BNE 7$ ;BR IF NO
2258 012006 004537 006746 JSR R5,SETVEC ;YES--RESET THE VECTORS
2259 012012 012126 14$ RECEIVER
2260 012014 012022 12$ TRANSMITTER
2261 012016 340 340 .BYTE 340,340 ;LEVEL
2262
2263 012020 000002 7$: RTI ;RETURN
2264
2265
2266 ;TRANSMITTER:
2267 012022 113777 001236 167364 12$: MOVB TEMP1,@TXDBUF ;LOAD THE TRANSMITTER BUFFER
2268 012030 105237 001236 INCB TEMP1 ;UP THE COUNT
2269 012034 122737 000377 001236 CMPB #377,TEMP1 ;ARE WE DONE
2270 012042 001026 BNE 13$ ;BR IF NO
2271 012044 012777 012054 167326 MOV #21$,@DUPTVC ;SETUP FOR NEXT PART
2272 012052 000422 BR 13$ ;LEAVE
2273 012054 012777 000377 167332 21$: MOV #377,@TXDBUF ;LOAD BUFFER
2274 012062 012777 012072 167310 MOV #22$,@DUPTVC ;SETUP NEXT PART
2275 012070 000413 BR 13$ ;LEAVE
2276 012072 012777 001000 167314 22$: MOV #TEOM,@TXDBUF ;SET END OF MSG
2277 012100 000240 NOP ;STALL
2278 012102 000240 NOP ;DITTO
2279 012104 042777 000120 167300 BIC #SEND!TXINTE,@TXCSR ;TURN OFF TRANSMITTER
2280 012112 012777 006774 167260 MOV #NO.BTRAP,@DUPTVC ;LOAD VECTOR
2281 012120 012716 011714 13$: MOV #5$, (SP) ;CRUNCH STACK
2282 012124 000002 RTI ;RETURNS
2283
2284 012126 117737 167254 001324 14$: MOVB @RXDBUF,DATA ;GET FIRST PART OF CRC
2285 012134 105777 167244 TSTB @RXCSR ;WAIT FOR SECOND PART
2286 012140 100375 BPL .-4 ;DITTO
2287 012142 017737 167240 001242 MOV @RXDBUF,TEMP3 ;GET THE REST OF THE CRC
2288 012150 113737 001242 001325 MOVB TEMP3,DATA+1 ;SET UP CRC CHARACTER
2289 012156 012716 012164 MOV #15$, (SP) ;SETUP FOR RETURN
2290 012162 000002 RTI ;RETURN
2291 012164 012737 000340 177776 15$: MOV #340,PS ;RAISE PS
2292 012172 005137 007156 COM CALBCC ;INVERT BCC
2293 012176 023737 007156 001324 CMP CALBCC,DATA ;COMPARE SOFTWARE AND HARDWARE BCC
2294 012204 001401 BEQ .+4 ;BR IF OK
2295 012206 104004 HLT 4 ;BCC COMPARISON ERROR
2296 012210 032737 010000 001242 BIT #CRCERR,TEMP3 ;CHECK THE ERROR BIT
2297 012216 001401 BEQ .+4 ;BR IF NO ERROR
2298 012220 104004 HLT 4 ;BCC ERROR--RECEIVER DOESN'T
2299 ;AGREE WITH WHAT TX SENT
2300 012222 052777 000400 167162 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2301 012230 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2302 012234 000137 011752 JMP 6$ ;LEAVE
2303
2304 ;***** TEST 11 *****
2305 ;*THIS TEST WILL CHECK FOR ABORT SEQUENCE
2306 ;*OF THE DUP IN A DATA STREAM
```

2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362012240 012737 000011 001226
012246 012737 012554 001216
012254 052777 000400 167130
012262 004737 005044
012266 004537 006746
012272 012452
012274 006774
012276 340 340
012300 005000
012302 005003
012304 012737 000340 177776
012312 052777 010377 167070
012320 052777 000120 167056
012326 052777 004020 167056
012334 005037 177776
012340 105777 167046
012344 100375
012346 052777 000400 167040
012354 105777 167032
012360 100375
012362 012777 000377 167024
012370 005200
012372 022700 000005
012376 001366
012400 052777 002000 167006
012406 012737 000310 012436
012414 032777 004000 166772
012422 001374
012424 032777 004000 166762
012432 001774
012434 005327
012436 000310
012440 001365
012442 104001
012444 012706 001150
012450 104400
012452 017701 166726
012456 017702 166724
012462 032701 000200
012466 001001
012470 104007
012472 122702 000377
012476 001401
012500 104002
012502 005203
012504 022703 000003
012510 001020
012512 105777 166666:*****
:*****
:*****
: TEST 11
:*****
:*****
TST11: MOV #11,@TSTNO
MOV #TST12,NEXT
BIS #MRESET,@TXCSR ;RESET THE DEVICE
JSR PC,SMALL ;WAIT FOR RESET TO FINISH
JSR R5,SETVEC ;SET UP INTERRUPT VECTORS
4\$;BASED ON THESE
NO.BTRAP ;PARAMETERS
.BYTE 340,340 ;LEVEL
CLR R0 ;CLEAR
CLR R3 ;DITTO
MOV #340,PS ;PS=7
BIS #PRISEC!377,@PARCSR ;LOAD SEC STATION AND ADRS
BIS #RCVEN!RINTEN,@RXCSR ;TURN ON THE RECEIVER
BIS #SEND!SYSTST,@TXCSR ;TURN ON TRANSMITTER
CLR PS
1\$: TSTB @TXCSR ;CHECK FOR TXDONE
BPL 1\$;BR IF NOT SET
BIS #TSOM,@TXDBUF ;TURN ON START OF MSG
2\$: TSTB @TXCSR ;WAIT FOR DONE
BPL 2\$;AND THEN
3\$: MOV #377,@TXDBUF ;LOAD A CHARACTER
INC R0 ;UPDATE CHARACTER COUNTER
CMP #5,R0 ;ARE ALL CHARACTERS LOADED?
BNE 2\$;BR IF NO
BIS #TABORT,@TXDBUF ;TURN ON ABORT
MOV #200.,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\$: 200. ;OF TIMES TO REPEAT
BNE 66\$;BR IF MORE TO GO
HLT 1 ;RECEIVER DID NOT INTERRUPT IN TIME
11\$: MOV #STACK,SP ;RESET STACK
SCOPE ;SCOPE THIS TEST
;RECEIVER INTERRUPT SERVICE ROUTINE
4\$: MOV @RXCSR,R1 ;GET THE CONTROL REGISTER
MOV @RXDBUF,R2 ;GET THE BUFFER
BIT #RXDONE,R1 ;CHECK FOR DONE
BNE 5\$;BR IF DONE SET
HLT 7 ;FALSE INTERRUPT
5\$: CMPB #377,R2 ;CHECK DATA CHARACTER
BEQ 6\$;BR IF A MATCH
HLT 2 ;DATA ERROR
6\$: INC R3 ;INC THE # OF CHARS TO DO
CMP #3,R3 ;CHECK TO SEE IF DONE
BNE 10\$;BR IF MORE TO GO
12\$: TSTB @RXCSR ;CHECK FOR


```
2363 012516 100375          BPL      12$      :DONE
2364 012520 017702 166662    MOV      @RXDBUF,R2  :READ THE BUFFER
2365 012524 032702 002000    BIT      #RABORT,R2  :TEST ABORT
2366 012530 001001          BNE      7$      :BR IF SET
2367 012532 104010          HLT      10      :FAILED TO RECEIVE ABORT
2368 012534 012716 012444    MOV      #11$, (SP)  :SET UP FOR RETURN
2369 012540 052777 000400 166644 7$: BIS      #MRESET,@TXCSR :RESET THE DEVICE
2370 012546 004737 005044    JSR      PC,SMALL  :WAIT FOR RESET TO FINISH
2371 012552 000002          RTI              :RETURN
```

```
***** TEST 12 *****
*THIS TEST PROVES THE RECEIVER WILL STOP
*ACCEPTING DATA IF SHUT OFF IN THE MIDDLE
*OF A MESSAGE, AND THAT IT WILL NOT
*RESTART UNTIL IT RECEIVES A FLAG
*****
```

```
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387 012554 012737 000012 001226 TST12: MOV      #12,@TSTNO
2388 012562 012737 013344 001216    MOV      #TST13,NEXT
2389 012570 052777 000400 166614    BIS      #MRESET,@TXCSR :RESET THE DEVICE
2390 012576 004737 005044          JSR      PC,SMALL  :WAIT FOR RESET TO FINISH
2391 012602 005000          CLR      R0      :CLEAR FOR SOFTWARE
2392 012604 004537 006746          JSR      R5,SETVEC :SET UP THE VECTORS
2393 012610 013034          3$      :RECEIVER
2394 012612 013234          14$     :TRANSMITTER
2395 012614          340          .BYTE 340,340 :LEVEL
2396 012616 012737 000340 177776    MOV      #340,PS :PROC STATUS=7
2397 012624 052777 001000 166556    BIS      #CRCEN,@PARCSR
2398 012632 052777 000120 166544    BIS      #RCVEN!RINTEN,@RXCSR :TURN ON RECEIVER
2399 012640 052777 004020 166544    BIS      #SEND!SYSTST,@TXCSR :START TRANSMITTER
2400 012646 005037 177776          CLR      PS      :LOWER PS
2401 012652 105777 166534          TSTB     @TXCSR  :CHECK FOR DONE
2402 012656 100375          BPL      1$      :BR IF NOT YET
2403 012660 052777 000400 166526    BIS      #TSOM,@TXDBUF :TURN ON START OF MSG
2404 012666 052777 000100 166516    BIS      #TXINTE,@TXCSR :TURN ON INT. ENABLE
2405 012674 012737 000764 012724    MOV      #500.,68$ :LOAD THE NUMBER
2406 012702 032777 004000 166504 66$: BIT      #TIMER,@TXDBUF :CHECK THE TIMER BIT
2407 012710 001374          BNE      66$     :BR IF SET
2408 012712 032777 004000 166474 67$: BIT      #TIMER,@TXDBUF :CHECK THE BIT
2409 012720 001774          BEQ      67$     :BR IF CLEAR
2410 012722 005327          DEC      (PC)+ :DECREMENT THE NUMBER
2411 012724 000764          68$: 500. :OF TIMES TO REPEAT
2412 012726 001365          BNE      66$     :BR IF MORE TO GO
2413 012730 104001          HLT      1      :DEVICE FAILED TO INTERRUPT IN TIME
2414 012732 012706 001150          MOV      #STACK,SP :RESET THE STACK
2415 012736 104400          SCOPE :SCOPE THIS TEST
2416 012740 004537 006746          JSR      R5,SETVEC :SET UP VECTORS
2417 012744 013164          23$     :RECEIVER
2418 012746 006774          NO.BTRAP :TRANSMITTER
```

```
2419 012750      340      340      .BYTE      340,340      ;LEVEL
2420 012752 052777 000020 166424  BIS      #RCVEN,@RXCSR
2421 012760 105777 166426      21$:  TSTB      @TXCSR      ;TEST DONE
2422 012764 100375      BPL      21$      ;BR IF NOT SET
2423 012766 012777 000070 166420  MOV      #70,@TXDBUF      ;PUSH OUT DATA CHARACTER
2424 012774 012737 000062 013024  MOV      #50,73$      ;LOAD THE NUMBER
2425 013002 032777 004000 166404  71$:  BIT      #TIMER,@TXDBUF      ;CHECK THE TIMER BIT
2426 013010 001374      BNE      71$      ;BR IF SET
2427 013012 032777 004000 166374  72$:  BIT      #TIMER,@TXDBUF      ;CHECK THE BIT
2428 013020 001774      BEQ      72$      ;BR IF CLEAR
2429 013022 005327      DEC      (PC)+      ;DECREMENT THE NUMBER
2430 013024 000062      73$:  50.      ;OF TIMES TO REPEAT
2431 013026 001365      BNE      71$      ;BR IF MORE TO GO
2432 013030 104001      HLT      1      ;FAILED TO INTERRUPT IN TIME
2433 013032 000737      BR      2$      ;FINISH
2434      ;INTERRUPT SVC ROUTINES
2435
2436      ;RECEIVER
2437 013034 017704 166344  3$:  MOV      @RXCSR,R4      ;GET THE CONTROL REGISTER
2438 013040 017705 166342      MOV      @RXDBUF,R5      ;GET THE BUFFER
2439 013044 032705 000400      BIT      #RSOM,R5      ;CHECK FOR START OF MSG
2440 013050 001001      BNE      4$      ;BR IF SET
2441 013052 104011      HLT      11      ;FAILED TO RECEIVE SOM
2442 013054 032704 000200      4$:  BIT      #RXDONE,R4      ;CHECK FOR DONE
2443 013060 001001      BNE      5$      ;BR IF SET
2444 013062 104007      HLT      7      ;FALSE INTERRUPT
2445 013064 122705 000377      5$:  CMPB      #377,R5      ;CHECK DATA
2446 013070 001401      BEQ      6$      ;BR IF A MATCH
2447 013072 104002      HLT      2      ;DATA ERROR
2448 013074 012777 013104 166272  6$:  MOV      #10$,@DUPRVC      ;RELOAD THE VECTOR
2449 013102 000002      7$:  RTI      ;RETURN
2450 013104 017705 166276      10$: MOV      @RXDBUF,R5      ;GET THE BUFFER
2451 013110 122705 000377      CMPB      #377,R5      ;CHECK THE CHARACTER
2452 013114 001401      BEQ      11$      ;BR IF A MATCH
2453 013116 104002      HLT      2      ;DATA ERROR
2454 013120 042777 000020 166256  11$: BIC      #RCVEN,@RXCSR      ;TURN OFF THE RECEIVER
2455 013126 012777 013136 166240  MOV      #12$,@DUPRVC      ;RELOAD THE VECTOR
2456 013134 000762      BR      7$      ;RETURN
2457 013136 017704 166242      12$: MOV      @RXCSR,R4      ;GET THE CONTROL REGISTER
2458 013142 017705 166240      MOV      @RXDBUF,R5      ;GET THE BUFFER
2459 013146 122705 000252      CMPB      #252,R5      ;CHECK THE CHARACTER
2460 013152 001402      BEQ      13$      ;BR IF A MATCH
2461 013154 104007      HLT      7      ;FALSE INTERRUPT
2462 013156 000751      BR      7$
2463 013160 104007      13$:  HLT      7      ;DEVICE INTERRUPTED AFTER RX ENABLE
2464 013162 000747      BR      7$      ;WAS CLEARED
2465 013164 017704 166214      23$: MOV      @RXCSR,R4      ;GET THE CONTROL REG
2466 013170 017705 166212      MOV      @RXDBUF,R5      ;GET THE BUFFER
2467 013174 032705 000400      BIT      #RSOM,R5      ;CHECK START OF MSG
2468 013200 001001      BNE      24$      ;BR IF SET
2469 013202 104011      HLT      11      ;SOM FAILED TO SET
2470 013204 122705 000070      24$: CMPB      #70,R5      ;CHECK DATA
2471 013210 001401      BEQ      25$      ;BR IF A MATCH
2472 013212 104002      HLT      2      ;DATA FAILED TO MATCH AFTER
2473      ;RESTARTING RECEIVER
2474 013214      25$:
```



```
2475 013214 052777 000400 166170 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2476 013222 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2477 013226 012716 012732 MOV #2$, (SP) ;CRUNCH STACK
2478 013232 000002 RTI ;RETURN
2479 ;TRANSMITTER
2480 013234 105777 166152 14$: TSTB @TXCSR ;CHECK DONE
2481 013240 100401 BMI 30$ ;BR IF SET
2482 013242 104007 HLT 7 ;FALSE INTERRUPT
2483 013244 012777 000377 166142 30$: MOV #377,@TXDBUF ;LOAD A CHARACTER
2484 013252 005200 INC R0 ;INC THE # TO DO
2485 013254 022700 000002 CMP #2,R0 ;CHECK TO SEE IF ALL ARE SENT
2486 013260 001030 BNE 15$ ;BR IF MORE TO GO
2487 013262 012777 013274 166110 MOV #16$,@DUPTVC ;RELOAD THE VECT
2488 013270 005000 CLR R0 ;CLEAR CHAR COUNT
2489 013272 000423 BR 15$
2490 013274 105777 166112 16$: TSTB @TXCSR ;TEST DONE
2491 013300 100401 BMI 17$ ;BR IF SET
2492 013302 104007 HLT 7 ;FALSE INTERRUPT
2493 013304 012777 000252 166102 17$: MOV #252,@TXDBUF ;LOAD A DATA CHARACTER
2494 013312 005200 INC R0 ;INC THE # TO DO
2495 013314 022700 000003 CMP #3,R0 ;CHECK FOR ALL DONE
2496 013320 001010 BNE 15$ ;BR IF MORE TO GO
2497 013322 012777 001400 166064 MOV #TEOM!TSOM,@TXDBUF ;END MSG
2498 013330 042777 000100 166054 BIC #TXINTE,@TXCSR
2499 013336 012716 012740 MOV #20$, (SP) ;CRUNCH STACK
2500 013342 000002 15$: RTI
```

```
2501
2502 ;***** TEST 13 *****
2503 ;*THIS TEST WILL TRANSMIT CONTIGUOUS ONES CHARACTERS
2504 ;*IN SECONDARY MODE WITH A BCC CHECK.
2505 ;*****
2506
```

```
2507 ;*****
2508 ; TEST 13
2509 ;*****
2510 ;*****
2511 ;*****
2512 013344 012737 000013 001226 TST13: MOV #13,@TSTNO
2513 013352 012737 014004 001216 MOV #TST14,NEXT
2514 013360 052777 000400 166024 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2515 013366 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2516 013372 012737 000340 177776 MOV #340,PS ;SET STATUS=7
2517 013400 005000 CLR R0
2518 013402 005002 CLR R2 ;SETUP FOR SOFTWARE
2519 013404 012701 000377 MOV #377,R1 ;CALCULATION OF BCC
2520 013410 012737 102010 007152 MOV #CRC.CCITT,XPOLY ;LOAD THE POLYNOMIAL
2521 013416 012737 177777 007156 MOV #-1,CALBCC ;SETUP FOR FIRST TIME
2522 013424 013737 007156 013446 1$: MOV CALBCC,3$ ;ALLOW FOR THE NEXT CHARACTER
2523 013432 010137 013444 MOV R1,2$ ;LOAD DATA
2524 013436 004537 007000 JSR R5,SIMBCC ;GO CALCULATE SOFTWARE BCC
2525 013442 000010 8. ;BASED ON THOSE PARAMETERS
2526 013444 000001 2$: .BLKW 1 ;DATA
2527 013446 000001 3$: .BLKW 1 ;PREVIOUS BCC
2528 013450 005200 INC R0 ;INC THE # OF CHARS TO DO
2529 013452 022700 000005 CMP #5,R0 ;ARE WE DONE?
2530 013456 001362 BNE 1$ ;BR IF NO
```

```
2531 013460 005000          CLR      R0          ;CLEAR OUT HOLD
2532 013462 004537 006746   JSR      R5,SETVEC    ;LOAD INTERRUPT VECTORS
2533 013466 013616          6$          ;RECEIVER
2534 013470 013722          11$         ;TRANSMITTER
2535 013472      340      340      .BYTE 340,340    ;LEVEL
2536 013474 052777 010377 165706   BIS      #PR1SEC!377,@PARCSR    ;ENTER SECONDARY MODE
2537 013502 052777 000120 165674   BIS      #RCVEN!RINTEN,@RXCSR ;TURN ON RECEIVER AND INTERRUPTS
2538 013510 052777 004020 165674   BIS      #SEND!SYSTST,@TXCSR ;TURN ON TRANSMITTER
2539 013516 105777 165670          20$: TSTB      @TXCSR
2540 013522 100375          BPL      20$
2541 013524 012777 000400 165662   MOV      #TSOM,@TXDBUF    ;START MESSAGE
2542 013532 052777 000100 165652   BIS      #TXINTE,@TXCSR    ;TURN ON INTERRUPT ENABLE
2543 013540 005037 177776          CLR      PS          ;LOWER PS
2544 013544          4$:
2545 013544 012737 000040 013574   MOV      #32,68$          ;LOAD THE NUMBER
2546 013552 032777 004000 165634   BIT      #TIMER,@TXDBUF    ;CHECK THE TIMER BIT
2547 013560 001374          66$: BNE      66$          ;BR IF SET
2548 013562 032777 004000 165624   BIT      #TIMER,@TXDBUF    ;CHECK THE BIT
2549 013570 001774          67$: BEQ      67$          ;BR IF CLEAR
2550 013572 005327          DEC      (PC)+          ;DECREMENT THE NUMBER
2551 013574 000040          68$: 32$          ;OF TIMES TO REPEAT
2552 013576 001365          BNE      66$          ;BR IF MORE TO GO
2553 013600 104001          HLT      1              ;FAILED TO INTERRUPT IN TIME
2554 013602          5$:
2555 013602 052777 000400 165602   BIS      #MRESET,@TXCSR    ;RESET THE DEVICE
2556 013610 004737 005044          JSR      PC,SMALL    ;WAIT FOR RESET TO FINISH
2557 013614 104400          SCOPE          ;SCOPE THIS TEST
2558
2559          ;INTERRUPT SERVICE ROUTINES.
2560
2561          ;RECEIVER
2562 013616 017737 165564 001324   6$: MOV      @RXDBUF,DATA    ;GET THE DATA
2563 013624 120137 001324          CMPB     R1,DATA          ;CHECK IT
2564 013630 001401          BEQ      .+4              ;BR IF A MATCH
2565 013632 104002          HLT      2              ;DATA ERROR
2566 013634 005200          INC      R0              ;UPDATE THE # OF CHARS TO DO
2567 013636 022700 000004          CMP      #4,R0          ;CHECK FOR ALL CHARS DONE
2568 013642 001003          BNE      7$              ;BR IF MORE TO GO
2569 013644 012777 013654 165522   MOV      #10$,@DUPRVC    ;SETUP TO GET BCC
2570 013652 000002          7$: RTI              ;RETURN
2571
2572 013654 117737 165526 001324   10$: MOVB     @RXDBUF,DATA    ;GET THE FIRST HALF OF BCC
2573 013662 105777 165516          TSTB     @RXCSR          ;WAIT FOR
2574 013666 100375          BPL      .-4              ;THE SECOND HALF
2575 013670 117737 165512 001325   MOVB     @RXDBUF,DATA+1    ;GET THE SECOND HALF
2576 013676 005137 007156          COM      CALBCC          ;INVERT BCC
2577 013702 023737 007156 001324   CMP      CALBCC,DATA      ;CHECK IT
2578 013710 001401          BEQ      .+4              ;BR IF OK
2579 013712 104004          HLT      4              ;BCC COMPARE ERROR
2580 013714 012716 013602          MOV      #5$, (SP)        ;FINISH TEST
2581 013720 000002          RTI              ;RETURN
2582
2583          ;TRANSMITTER
2584 013722 012777 000377 165464   11$: MOV      #377,@TXDBUF    ;LOAD A DATA CHARACTER
2585 013730 005202          INC      R2              ;INC THE # OF CHARS TO DO
2586 013732 022702 000005          CMP      #5,R2          ;CHECK TO SEE OF DONE
```



```
2587 013736 001017      BNE      13$      ;BR IF MORE TO GO
2588 013740 012777 013750 165432      MOV      #12$,@DUPTVC ;SETUP NEXT VECTOR
2589 013746 000413      BR       13$      ;RETURN
2590 013750 012777 001000 165436 12$:  MOV      #TEOM,@TXDBUF ;END MSG
2591 013756 000240      NOP      ;WAIT
2592 013760 000240      NOP      ;DITTO
2593 013762 042777 000120 165422      BIC      #SEND!TXINTE,@TXCSR ;TURN OFF TRANSMITTER
2594 013770 012777 006774 165402      MOV      #NO.BTRAP,@DUPTVC ;RESET THE VECTOR
2595 013776 012716 013544 13$:  MOV      #4$, (SP) ;GO BACK TO WAIT LOOP
2596 014002 000002      RTI       ;RETURN
2597
2598                                     :***** TEST 14 *****
2599                                     :*THIS TEST PROVES THE INTERACTION OF DEC MODE,
2600                                     :*TSOM, SYNC, TXACT, TXDONE
2601                                     :*****
2602
2603                                     :*****
2604                                     :*
2605                                     : TEST 14
2606                                     :*
2607                                     :*****
2608                                     :*****
2609 014004 012737 000014 001226 TST14: MOV      #14,@TSTNO
2610 014012 012737 014156 001216      MOV      #TST15,NEXT
2611 014020 052777 000400 165364      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
2612 014026 004737 005044      JSR      PC,SMALL ;WAIT FOR RESET TO FINISH
2613 014032 012777 101026 165350      MOV      #DECMOD!26!CRCEN,@PARCSR
2614 014040 052777 004000 165344      BIS      #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
2615 014046 052777 000020 165336      BIS      #SEND,@TXCSR ;TURN ON TRANSMITTER
2616 014054 012777 000426 165332      MOV      #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
2617 014062 012737 000005 014112      MOV      #5,68$ ;LOAD THE NUMBER
2618 014070 032777 004000 165316 66$: BIT      #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2619 014076 001374      BNE      66$ ;BR IF SET
2620 014100 032777 004000 165306 67$: BIT      #TIMER,@TXDBUF ;CHECK THE BIT
2621 014106 001774      BEQ      67$ ;BR IF CLEAR
2622 014110 005327      DEC      (PC)+ ;DECREMENT THE NUMBER
2623 014112 000005 68$: 5 ;OF TIMES TO REPEAT
2624 014114 001365      BNE      66$ ;BR IF MORE TO GO
2625 014116 017704 165270      MOV      @TXCSR,R4 ;GET THE CSR
2626 014122 032704 000200      BIT      #TXDONE,R4 ;CHECK TRANSMITTER DONE
2627 014126 001001      BNE      1$ ;BR IF SET
2628 014130 104016      HLT      16 ;TXDONE FAILED TO SET
2629 014132 032704 001000 1$: BIT      #TXACT,R4 ;TEST ACTIVE
2630 014136 001001      BNE      2$ ;BR IF SET
2631 014140 104017      HLT      17 ;ACTIVE FAILED TO SET
2632 014142      2$:
2633 014142 052777 000400 165242      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
2634 014150 004737 005044      JSR      PC,SMALL ;WAIT FOR RESET TO FINISH
2635 014154 104400      SCOPE ;SCOPE THIS TEST
2636
2637
2638                                     :***** TEST 15 *****
2639                                     :*THIS TEST PROVES THE INTERACTION OF TEOM,
2640                                     :*SEND, TXACT AND TXDONE IN DEC MODE.
2641                                     :*****
2642
```

2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698

014156	012737	000015	001226
014164	012737	014352	001216
014172	052777	000400	165212
014200	004737	005044	
014204	012777	101026	165176
014212	052777	004000	165172
014220	052777	000020	165164
014226	012777	000426	165160
014234	105777	165152	
014240	100375		
014242	012777	000426	165144
014250	105777	165136	
014254	100375		
014256	012777	001000	165130
014264	042777	000020	165120
014272	012737	000025	014322
014300	032777	004000	165106
014306	001374		
014310	032777	004000	165076
014316	001774		
014320	005327		
014322	000025		
014324	001365		
014326	105777	165060	
014332	100401		
014334	104016		
014336	032777	001000	165046
014344	001401		
014346	104020		
014350	104400		

```
*****
:
: TEST 15
:
: *****
: *****
TST15: MOV #15,@TSTNO
      MOV #TST16,NEXT
      BIS #MRESET,@TXCSR ;RESET THE DEVICE
      JSR PC,SMALL ;WAIT FOR RESET TO FINISH
      MOV #DECMOD!26!CRCEN,@PARCSR
      BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
      BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
      MOV #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
1$: TSTB @TXCSR ;CHECK FOR DONE
   BPL 1$ ;BR IF NOT YET
   MOV #TSOM!26,@TXDBUF ;LOAD A SECOND SYNC
2$: TSTB @TXCSR ;AND NOW WAIT
   BPL 2$ ;FOR DONE AGAIN
   MOV #TEOM,@TXDBUF ;SET END OF MSG
   BIC #SEND,@TXCSR ;TURN OFF TRANSMITTER
   MOV #25,68$ ;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$: 25 ;OF TIMES TO REPEAT
   BNE 66$ ;BR IF MORE TO GO
   TSTB @TXCSR ;CHECK DONE
   BMI 3$ ;BR IF SET
   HLT 16 ;DONE FAILED TO SET AFTER TURNING OFF TX.
3$: BIT #TXACT,@TXCSR ;CHECK ACTIVE
   BEQ 4$ ;BR IF OFF
   HLT 20 ;ACTIVE IS STILL SET-SHOULD BE RESET
4$: SCOPE ;SCOPE FOR THIS TEST.
```

```
***** TEST 16 *****
: *THIS TEST PROVES THAT THE DUP WILL NOT
: *SYNC UP IN LESS THAN TWO SYNCs
: *****
```

```
*****
:
: TEST 16
:
: *****
: *****
TST16: MOV #16,@TSTNO
      MOV #TST17,NEXT
      BIS #MRESET,@TXCSR ;RESET THE DEVICE
      JSR PC,SMALL ;WAIT FOR RESET TO FINISH
      MOV #DECMOD!26!CRCEN,@PARCSR
      BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
      BIS #RCVEN,@RXCSR ;LOAD RCVEN
```


2699	014422	052777	000020	164762	BIS	#SEND,@TXCSR	:TURN ON TRANSMITTER
2700	014430	012777	000426	164756	MOV	#TSOM!26,@TXDBUF	:OUTPUT A SYNC CHAR
2701	014436	105777	164750		1\$: TSTB	@TXCSR	:CHECK TRANSMITTER DONE
2702	014442	100375			BPL	1\$	:WAIT TILL SET
2703	014444	012777	000125	164742	MOV	#125,@TXDBUF	:LOAD DATA
2704	014452	012737	000005	014502	MOV	#5,68\$	:LOAD THE NUMBER
2705	014460	032777	004000	164726	66\$: BIT	#TIMER,@TXDBUF	:CHECK THE TIMER BIT
2706	014466	001374			BNE	66\$	:BR IF SET
2707	014470	032777	004000	164716	67\$: BIT	#TIMER,@TXDBUF	:CHECK THE BIT
2708	014476	001774			BEQ	67\$	:BR IF CLEAR
2709	014500	005327			DEC	(PC)+	:DECREMENT THE NUMBER
2710	014502	000005			68\$: 5		:OF TIMES TO REPEAT
2711	014504	001365			BNE	66\$	:BR IF MORE TO GO
2712	014506	105777	164672		TSTB	@RXCSR	:CHECK FOR RECEIVER DONE
2713	014512	100002			BPL	2\$	:BR IF NOT SET
2714	014514	104021			HLT	21	:DEVICE SYNC'S UP IN LESS THAN 2 SYNC'S!!
2715	014516	000472			BR	5\$	:LEAVE
2716	014520				2\$:		
2717	014520	052777	000400	164664	BIS	#MRESET,@TXCSR	:RESET THE DEVICE
2718	014526	004737	005044		JSR	PC,SMALL	:WAIT FOR RESET TO FINISH
2719	014532	012777	101026	164650	MOV	#CRCEN!DECMOD!26,@PARCSR	:LOAD THE MODE AND SYNC CHAR
2720	014540	052777	000020	164636	BIS	#RCVEN,@RXCSR	:TURN ON RECEIVER
2721	014546	052777	004000	164636	BIS	#SYSTST,@TXCSR	:ENTER SYSTEM TEST MODE
2722	014554	052777	000020	164630	BIS	#SEND,@TXCSR	:TURN ON TRANSMITTER
2723	014562	012777	000426	164624	MOV	#TSOM!26,@TXDBUF	:OUTPUT A SYNC CHAR
2724	014570	105777	164616		69\$: TSTB	@TXCSR	:CHECK DONE
2725	014574	100375			BPL	69\$	:BR IF NOT SET
2726	014576	012777	000426	164610	MOV	#TSOM!26,@TXDBUF	:SEND SYNC
2727	014604	105777	164602		3\$: TSTB	@TXCSR	:CHECK DONE
2728	014610	100375			BPL	3\$	:WAIT
2729	014612	012777	000125	164574	MOV	#125,@TXDBUF	:LOAD DATA
2730	014620	012737	000020	014650	MOV	#20,74\$	:LOAD THE NUMBER
2731	014626	032777	004000	164560	72\$: BIT	#TIMER,@TXDBUF	:CHECK THE TIMER BIT
2732	014634	001374			BNE	72\$	:BR IF SET
2733	014636	032777	004000	164550	73\$: BIT	#TIMER,@TXDBUF	:CHECK THE BIT
2734	014644	001774			BEQ	73\$	:BR IF CLEAR
2735	014646	005327			DEC	(PC)+	:DECREMENT THE NUMBER
2736	014650	000020			74\$: 20		:OF TIMES TO REPEAT
2737	014652	001365			BNE	72\$	:BR IF MORE TO GO
2738	014654	105777	164524		TSTB	@RXCSR	:CHECK FOR DONE
2739	014660	100401			BMI	4\$	:BR IF SET
2740	014662	104022			HLT	22	:FAILED TO RECEIVE DATA
2741	014664	017737	164516	001236	4\$: MOV	@RXDBUF,TEMP1	:READ DATA
2742	014672	122737	000125	001236	CMPB	#125,TEMP1	:CHECK IT
2743	014700	001401			BEQ	5\$	:BR IF MATCH
2744	014702	104022			HLT	22	:DATA COMPARE ERROR
2745	014704				5\$:		
2746	014704	052777	000400	164500	BIS	#MRESET,@TXCSR	:RESET THE DEVICE
2747	014712	004737	005044		JSR	PC,SMALL	:WAIT FOR RESET TO FINISH
2748	014716	104400			SCOPE		:SCOPE THIS TEST

2749
2750
2751
2752
2753
2754:***** TEST 17 *****
:*THIS TEST PROVES THE RECEIVER WILL STRIP THE FIRST
:*TWO SYNC'S AND WILL PRESENT ALL SUBSEQUENT SYNC'S.
:*****

```
2755
2756
2757
2758
2759
2760
2761
2762 014720 012737 000017 001226 TST17: MOV #17,@TSTNO
2763 014726 012737 015142 001216 MOV #TST20,NEXT
2764 014734 052777 000400 164450 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2765 014742 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2766 014746 012777 101026 164434 MOV #DECMOD!26!CRCEN,@PARCSR
2767 014754 052777 004000 164430 BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
2768 014762 052777 000020 164414 BIS #RCVEN,@RXCSR ;LOAD RCVEN
2769 014770 052777 000020 164414 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
2770 014776 012777 000426 164410 MOV #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
2771 015004 032777 004000 164402 64$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2772 015012 001374 BNE 64$ ;BR IF SET
2773 015014 032777 004000 164372 65$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2774 015022 001774 BEQ 65$ ;BR IF CLEAR
2775 015024 105777 164362 69$: TSTB @TXCSR ;CHECK DONE
2776 015030 100375 BPL 69$ ;BR IF NOT SET
2777 015032 012777 000426 164354 MOV #TSOM!26,@TXDBUF ;SEND SYNC
2778 015040 105777 164346 70$: TSTB @TXCSR ;CHECK DONE
2779 015044 100375 BPL 70$ ;BR IF NOT SET
2780 015046 012777 000426 164340 MOV #TSOM!26,@TXDBUF ;SEND SYNC
2781 015054 012737 000020 015104 MOV #20,75$ ;LOAD THE NUMBER
2782 015062 032777 004000 164324 73$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2783 015070 001374 BNE 73$ ;BR IF SET
2784 015072 032777 004000 164314 74$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2785 015100 001774 BEQ 74$ ;BR IF CLEAR
2786 015102 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2787 015104 000020 75$: 20 ;OF TIMES TO REPEAT
2788 015106 001365 BNE 73$ ;BR IF MORE TO GO
2789 015110 105777 164270 TSTB @RXCSR ;CHECK FOR DONE
2790 015114 100401 BMI 1$ ;BR IF SET
2791 015116 104021 HLT 21 ;DONE NOT SET-DEVICE FAILED TO SYNC UP
2792 015120 117737 164262 001236 1$: MOVB @RXDBUF,TEMP1 ;READ BUFFER
2793 015126 122737 000026 001236 CMPB #26,TEMP1 ;CHECK FOR SYNC
2794 015134 001401 BEQ 2$ ;BR IF OK
2795 015136 104022 HLT 22 ;DATA ERROR
2796 015140 104400 2$: SCOPE ;SCOPE THIS TEST
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810 015142 012737 000020 001226 TST20: MOV #20,@TSTNO
```

```
***** TEST 20 *****
;THIS TEST PROVES THE DUP11 WILL
;IDLE SYNCs. IDLE 64. SYNCs
*****
```

```
*****
;
; TEST 20
;
*****
*****
```



```
2811 015150 012737 015406 001216 MOV #TST21,NEXT
2812 015156 052777 000400 164226 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2813 015164 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2814 015170 012777 101026 164212 MOV #DECMOD!26!CRCEN,@PARCSR
2815 015176 052777 004000 164206 BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
2816 015204 052777 000020 164172 BIS #RCVEN,@RXCSR ;LOAD RCVEN
2817 015212 052777 000020 164172 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
2818 015220 012777 000426 164166 MOV #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
2819 015226 105777 164160 64$: TSTB @TXCSR ;CHECK DONE
2820 015232 100375 BPL 64$ ;BR IF NOT SET
2821 015234 012777 000426 164152 MOV #TSOM!26,@TXDBUF ;SEND SYNC
2822 015242 105777 164144 65$: TSTB @TXCSR ;CHECK DONE
2823 015246 100375 BPL 65$ ;BR IF NOT SET
2824 015250 012777 000426 164136 MOV #TSOM!26,@TXDBUF ;SEND SYNC
2825 015256 005037 001236 CLR TEMP1
2826 015262 005037 001240 CLR TEMP2
2827 015266 012737 000100 001236 MOV #64.,TEMP1 ;LOAD # OF SYNCs
2828 015274 012737 000010 015324 MOV #10,70$ ;LOAD THE NUMBER
2829 015302 032777 004000 164104 68$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2830 015310 001374 BNE 68$ ;BR IF SET
2831 015312 032777 004000 164074 69$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2832 015320 001774 BEQ 69$ ;BR IF CLEAR
2833 015322 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2834 015324 000010 70$: 10 ;OF TIMES TO REPEAT
2835 015326 001365 BNE 68$ ;BR IF MORE TO GO
2836 015330 105777 164056 1$: TSTB @TXCSR ;CHECK DONE
2837 015334 100401 BMI 2$ ;BR IF SET
2838 015336 104016 HLT 16 ;DONE FAILED TO SET
2839 015340 012777 000426 164046 2$: MOV #TSOM!26,@TXDBUF ;LOAD A SYNC
2840 015346 005337 001236 DEC TEMP1 ;LOWER THE # OF SYNCs TO DO
2841 015352 001001 BNE 4$ ;BR IF MORE TO GO
2842 015354 104400 3$: SCOPE ;SCOPE THIS TEST
2843
2844 015356 105777 164022 4$: TSTB @RXCSR ;CHECK RECEIVER DONE
2845 015362 100375 BPL 4$ ;WAIT TILL SET
2846 015364 017737 164016 001240 MOV @RXDBUF,TEMP2 ;GET THE BUFFER
2847 015372 122737 000026 001240 CMPB #26,TEMP2 ;CHECK IT FOR SYNC
2848 015400 001753 BEQ 1$ ;BR IF OK
2849 015402 104021 HLT 21 ;CHARACTER IS TEMP2 NOT A SYNC!
2850 015404 000763 BR 3$ ;LEAVE TEST
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
```

```
***** TEST 21 *****
*THIS TEST PROVES THE STRIP SYNC
*FUNCTION OF THE RECEIVER. SYNC UP
*THE RECEIVER, SEND DATA WITH A SYNC
*CHARACTER IMBEDDED AND CHECK FOR
*THE SYNC TO BE RECEIVED.
```

```
*****
```

```
TEST 21
```

```
*****
```

```
*****
TST21: MOV #21,@TSTNO
```

```
2867 015414 012737 016126 001216      MOV    #TST22,NEXT
2868 015422 012737 000340 177776      MOV    #340,PS          ;RAISE STATUS
2869 015430 004537 006746              JSR    R5,SETVEC        ;SET UP VECTORS
2870 015434 015714              S$              ;BASED ON
2871 015436 006774              NO.BTRAP          ;THESE
2872 015440      340      340      .BYTE    340,340      ;PARAMETERS
2873
2874 015442 052777 000400 163742      BIS    #MRESET,@TXCSR  ;RESET THE DEVICE
2875 015450 004737 005044              JSR    PC,SMALL        ;WAIT FOR RESET TO FINISH
2876 015454 012777 101026 163726      MOV    #DECMOD!26!CRCEN,@PARCSR
2877 015462 052777 004000 163722      BIS    #SYSTST,@TXCSR  ;ENTER SYSTEM TEST MODE
2878 015470 052777 000420 163706      BIS    #RCVEN!STPSYN,@RXCSR ;LOAD RCVEN!STPSYN
2879 015476 052777 000020 163706      BIS    #SEND,@TXCSR    ;TURN ON TRANSMITTER
2880 015504 012777 000426 163702      MOV    #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
2881 015512 105777 163674      64$: TSTB   @TXCSR      ;CHECK DONE
2882 015516 100375              BPL     64$          ;BR IF NOT SET
2883 015520 012777 000426 163666      MOV    #TSOM!26,@TXDBUF ;SEND SYNC
2884 015526 105777 163660      65$: TSTB   @TXCSR      ;CHECK DONE
2885 015532 100375              BPL     65$          ;BR IF NOT SET
2886 015534 012777 000426 163652      MOV    #TSOM!26,@TXDBUF ;SEND SYNC
2887 015542 105777 163644      66$: TSTB   @TXCSR      ;CHECK DONE
2888 015546 100375              BPL     66$          ;BR IF NOT SET
2889 015550 012777 000426 163636      MOV    #TSOM!26,@TXDBUF ;SEND SYNC
2890 015556 005037 177776              CLR    PS          ;LOWER PS
2891 015562 052777 000100 163614      BIS    #RINTEN,@RXCSR   ;TURN ON INTERRUPTS
2892 015570 105777 163616      1$: TSTB   @TXCSR      ;CHECK TX DONE
2893 015574 100375              BPL     1$          ;WAIT FOR SET
2894 015576 012777 000252 163610      MOV    #252,@TXDBUF     ;LOAD A CHARACTER
2895 015604 105777 163602      2$: TSTB   @TXCSR      ;CHECK TX DONE
2896 015610 100375              BPL     2$          ;WAIT TO BE SET
2897 015612 012777 000026 163574      MOV    #26,@TXDBUF      ;LOAD THE SYNC CHAR
2898 015620 105777 163566      3$: TSTB   @TXCSR      ;CHECK DONE AGAIN
2899 015624 100375              BPL     3$          ;WAIT
2900 015626 012777 000125 163560      MOV    #125,@TXDBUF     ;LOAD ANOTHER CHARACTER
2901 015634 105777 163552      4$: TSTB   @TXCSR      ;CHECK DONE
2902 015640 100375              BPL     4$          ;WAIT
2903 015642 012777 001000 163544      MOV    #TEOM,@TXDBUF    ;SET END OF MESSAGE
2904 015650 042777 000020 163534      BIC    #SEND,@TXCSR     ;TURN OFF TRANSMITTER
2905 015656 012737 000050 015706      MOV    #40,,71$        ;LOAD THE NUMBER
2906 015664 032777 004000 163522      69$: BIT    #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2907 015672 001374              BNE     69$          ;BR IF SET
2908 015674 032777 004000 163512      70$: BIT    #TIMER,@TXDBUF ;CHECK THE BIT
2909 015702 001774              BEQ     70$          ;BR IF CLEAR
2910 015704 005327              DEC    (PC)+        ;DECREMENT THE NUMBER
2911 015706 000050      71$: 40.      ;OF TIMES TO REPEAT
2912 015710 001365              BNE     69$          ;BR IF MORE TO GO
2913 015712 104023              HLT     23          ;FAILED TO TAKE A RECEIVER INTERRUPT
2914
2915      ;RECEIVER INTERRUPT SERVICE ROUTINE
2916 015714 017700 163464      5$: MOV    @RXCSR,R0      ;READ CSR
2917 015720 017701 163462      MOV    @RXDBUF,R1     ;READ BUFFER
2918 015724 032700 000200      BIT    #RXDONE,R0      ;CHECK FOR DONE
2919 015730 001001              BNE     6$          ;BR IF SET
2920 015732 104024              HLT     24          ;RX DONE FAILED TO SET-ERRONEOUS INTERRUPT
2921 015734 032700 004000      6$: BIT    #REACT,R0      ;CHECK FOR ACTIVE
2922 015740 001001              BNE     7$          ;BR IF SET
```



```
2923 015742 104025          HLT      25          ;RX ACTIVE FAILED TO SET
2924 015744          7$:          TST      R1          ;CHECK FOR ERROR
2925 015744 005701          BPL      10$          ;BR IF NO ERROR
2926 015746 100001          HLT      26          ;RECEIVER ERROR
2927 015750 104026          CMPB     #252,R1        ;CHECK FOR CORRECT DATA
2928 015752 122701 000252    BEQ      11$          ;BR IF OK
2929 015756 001401          HLT      22          ;DATA FAILED TO MATCH
2930 015760 104022          MOV      #12$,@DUPRVC    ;LOAD VECTOR
2931 015762 012777 015772 163404 11$:    BR      20$          ;CONTINUE
2932 015770 000455          MOV      @RXCSR,R0       ;READ CSR
2933 015772 017700 163406    MOV      @RXDBUF,R1     ;READ BUFFER
2934 015776 017701 163404    BIT      #RXDONE,R0     ;CHECK FOR DONE
2935 016002 032700 000200    BNE      13$          ;BR IF OK
2936 016006 001001          HLT      24          ;RX DONE FAILED TO SET-ERRONEOUS INTERRUPT
2937 016010 104024          13$:    TST      R1          ;TEST FOR ERROR
2938 016012 005701          BPL      14$          ;BR IF NO ERROR
2939 016014 100001          HLT      26          ;ERROR SET
2940 016016 104026          CMPB     #26,R1        ;CHECK CHARACTER
2941 016020 122701 000026    BEQ      16$          ;BR IF OK-IF NOT, THEN
2942 016024 001422          CMPB     #125,R1        ;CHECK FOR CLEARING SYNC
2943 016026 122701 000125    BEQ      15$          ;BR IF A NEXT CHARACTER
2944 016032 001402          HLT      22          ;ERRONEOUS CHARACTER
2945 016034 104022          BR      16$          ;BR TO END OF TEST
2946 016036 000415          HLT      21          ;STRIPPED OUT THE SYNC CHAR!!
2947 016040 104021          MOV      #21$,@DUPRVC    ;SET UP VECTOR
2948 016042 012777 016052 163324 21$:    BR      20$          ;LEAVE
2949 016050 000425          MOV      @RXCSR,R0       ;GET CSR
2950 016052 017700 163326    MOV      @RXDBUF,R1     ;GET BUFFER
2951 016056 017701 163324    CMPB     #125,R1        ;CHECK DATA
2952 016062 122701 000125    BEQ      16$          ;BR IF A MATCH
2953 016066 001401          HLT      22          ;DATA COMPARE ERROR
2954 016070 104022          BIT      #REACT,@RXCSR    ;TEST ACTIVE
2955 016072 032777 004000 163304 16$:    BNE      17$          ;BR IF ON
2956 016100 001001          HLT      25          ;ACTIVE SHOULD BE ON
2957 016102 104025          17$:    BIS      #MRESET,@TXCSR ;RESET THE DEVICE
2958 016104          JSR      PC,SMALL ;WAIT FOR RESET TO FINISH
2959 016104 052777 000400 163300  MOV      #STACK,SP ;RESET STACK
2960 016112 004737 005044          SCOPE          ;SCOPE THIS TEST
2961 016116 012706 001150          RTI           ;RETURN
2962 016122 104400          20$:
2963 016124 000002
2964
2965
2966          ;***** TEST 22 *****
2967          ;*THIS TEST PROVES THAT A BINARY COUNT
2968          ;*PATTERN CAN BE RUN IN DEC MODE
2969          ;*WITHOUT A BCC CALCULATION
2970          ;*****
2971          ;*****
2972          ;*
2973          ;* TEST 22
2974          ;*
2975          ;*****
2976          ;*****
2977 016126 012737 000022 001226  TST22: MOV      #22,@TSTNO
2978 016134 012737 016610 001216  MOV      #TST23,NEXT
```

```
2979 016142 012737 000340 177776      MOV    #340,PS
2980 016150 005000                      CLR    R0                      ;CLR OUT DATA POINTER
2981 016152 005001                      CLR    R1                      ;DITTO
2982 016154 004537 006746      JSR    R5,SETVEC      ;SET UP INTERRUPTS
2983 016160 016370                      4$      ;RECEIVER
2984 016162 016542                      17$      ;TRANSMITTER
2985 016164      340      340      .BYTE 340,340      ;LEVEL
2986
2987 016166 052777 000400 163216      BIS    #MRESET,@TXCSR      ;RESET THE DEVICE
2988 016174 004737 005044      JSR    PC,SMALL      ;WAIT FOR RESET TO FINISH
2989 016200 012777 101026 163202      MOV    #DECMOD!26!CRCEN,@PARCSR
2990 016206 052777 004000 163176      BIS    #SYSTST,@TXCSR      ;ENTER SYSTEM TEST MODE
2991 016214 052777 000020 163162      BIS    #RCVEN,@RXCSR      ;LOAD RCVEN
2992 016222 052777 000020 163162      BIS    #SEND,@TXCSR      ;TURN ON TRANSMITTER
2993 016230 012777 000426 163156      MOV    #TSOM!26,@TXDBUF      ;OUTPUT A SYNC CHAR
2994 016236 032777 004000 163150 64$:  BIT    #TIMER,@TXDBUF      ;CHECK THE TIMER BIT
2995 016244 001374                      BNE    64$      ;BR IF SET
2996 016246 032777 004000 163140 65$:  BIT    #TIMER,@TXDBUF      ;CHECK THE TIMER BIT
2997 016254 001774                      BEQ    65$      ;BR IF CLEAR
2998 016256 105777 163130      69$:  TSTB   @TXCSR      ;CHECK DONE
2999 016262 100375                      BPL    69$      ;BR IF NOT SET
3000 016264 012777 000426 163122      MOV    #TSOM!26,@TXDBUF      ;SEND SYNC
3001 016272 005037 177776      CLR    PS
3002 016276 052777 000100 163100      BIS    #RINTEN,@RXCSR      ;TURN ON INT ENABLES
3003 016304 052777 000100 163100      BIS    #TXINTE,@TXCSR      ;DITTO
3004 016312
3005 016312 012737 000310 016342 30$:  MOV    #200.,74$      ;LOAD THE NUMBER
3006 016320 032777 004000 163066 72$:  BIT    #TIMER,@TXDBUF      ;CHECK THE TIMER BIT
3007 016326 001374                      BNE    72$      ;BR IF SET
3008 016330 032777 004000 163056 73$:  BIT    #TIMER,@TXDBUF      ;CHECK THE BIT
3009 016336 001774                      BEQ    73$      ;BR IF CLEAR
3010 016340 005327                      DEC    (PC)+      ;DECREMENT THE NUMBER
3011 016342 000310      74$:  200.      ;OF TIMES TO REPEAT
3012 016344 001365                      BNE    72$      ;BR IF MORE TO GO
3013 016346 104023                      HLT    23      ;FAILED TO FINISH TEST
3014 016350
3015 016350 052777 000400 163034 3$:  BIS    #MRESET,@TXCSR      ;RESET THE DEVICE
3016 016356 004737 005044      JSR    PC,SMALL      ;WAIT FOR RESET TO FINISH
3017 016362 012706 001150      MOV    #STACK,SP      ;RESET THE STACK
3018 016366 104400      SCOPE      ;SCOPE THIS TEST
3019
3020
3021      ;RECEIVER INT SVC ROUTINE
3022 016370 017702 163010 4$:  MOV    @RXCSR,R2      ;SAVE CSR
3023 016374 017703 163006      MOV    @RXDBUF,R3      ;SAVE BUFFER
3024 016400 032702 004000      BIT    #RECACT,R2      ;TEST RX ACTIVE
3025 016404 001004                      BNE    5$      ;BR IF OK
3026 016406 104025                      HLT    25      ;ACTIVE NOT SET
3027 016410 012716 016350      MOV    #3$, (SP)      ;SETUP FOR RETURN
3028 016414 000432                      BR     12$
3029 016416 032702 000200 5$:  BIT    #RXDONE,R2      ;TEST DONE
3030 016422 001004                      BNE    6$      ;BR IF OK
3031 016424 104024                      HLT    24      ;FALSE INTERRUPT
3032 016426 012716 016350      MOV    #3$, (SP)      ;SETUP FOR RETURN
3033 016432 000423                      BR     12$
3034 016434 005703 6$:  TST    R3      ;CHECK FOR ERROR
```



```

3035 016436 100004      BPL      7$      ;BR IF NO ERROR
3036 016440 104026      HLT      26      ;DATA ERROR
3037 016442 012716 016350 MOV      #3$, (SP) ;SET UP RETURN
3038 016446 000415      BR      12$
3039 016450 120103      7$: CMPB    R1,R3      ;CHECK DATA
3040 016452 001404      BEQ      10$      ;BR IF OK
3041 016454 104022      HLT      22      ;BAD DATA
3042 016456 012716 016350 MOV      #3$, (SP) ;SETUP RETURN
3043 016462 000407      BR      12$
3044 016464 005201      10$: INC     R1      ;UPDATE DATA
3045 016466 001002      BNE      11$      ;BR IF MORE TO GO
3046 016470 012716 016350 MOV      #3$, (SP) ;SETUP RETURN
3047 016474 012777 016504 162672 11$: MOV    #22$, @DUPRVC ;SETUP NEW RETURN FOR INTERRUPT
3048 016502 000002      12$: RTI
3049 016504 017702 162674 22$: MOV    @RXCSR,R2      ;RETURN
3050 016510 017703 162672 MOV    @RXDBUF,R3
3051 016514 005703      TST      R3
3052 016516 100001      BPL      23$
3053 016520 104026      HLT      26      ;ERROR
3054 016522 120103      23$: CMPB    R1,R3
3055 016524 001401      BEQ      24$
3056 016526 104022      HLT      22      ;DATA COMPARE ERROR
3057 016530 105201      24$: INCB    R1
3058 016532 001363      BNE      12$
3059 016534 012716 016350 MOV      #3$, (SP)
3060 016540 000760      BR      12$
3061
3062      ;TRANSMITTER
3063 016542 010077 162646 17$: MOV    R0,@TXDBUF      ;PUSH OUT DATA
3064 016546 105200      INCB    R0      ;UPDATE IT
3065 016550 001014      BNE      21$      ;BR IF MORE
3066 016552 105777 162634 20$: TSTB    @TXCSR      ;CHECK FOR NEXT DONE
3067 016556 100375      BPL      20$      ;WAIT
3068 016560 052777 001000 162626 BIS    #TEOM,@TXDBUF      ;END MSG
3069 016566 042777 000120 162616 BIC    #SEND!TXINTE,@TXCSR      ;SHUT OF TRANSMITTER
3070 016574 012777 006774 162576 MOV    #NO.BTRAP,@DUPTVC      ;RESET VECTOR ADRS
3071 016602 012716 016312 21$: MOV    #30$, (SP)
3072 016606 000002      RTI      ;RETURN
3073
3074
3075
3076      ;***** TEST 23 *****
3077      ;*THIS TEST PROVES THAT A BINARY COUNT
3078      ;*PATTERN CAN BE RUN IN DEC MODE
3079      ;*WITH A BCC CALCULATION USING
3080      ;*THE CRC16 POLYNOMIAL
3081      ;*****
3082      ;*****
3083      ;*****
3084      ; TEST 23
3085      ;*****
3086      ;*****
3087      ;*****
3088 016610 012737 000023 001226 TST23: MOV    #23,@#TSTNO
3089 016616 012737 017432 001216 MOV    #TST24,NEXT
3090 016624 012737 000340 177776 MOV    #340,PS
  
```

3091	016632	005000			CLR	R0		
3092	016634	012737	120001	007152	MOV	#CRC16,XPOLY		;SET THE POLYNOMIAL
3093	016642	005037	007156		CLR	CALBCC		;CLEAR OUT OLD BCC
3094	016646	013737	007156	016670	MOV	CALBCC,35\$		;LOAD BCC
3095	016654	010037	016666		MOV	R0,2\$		;LOAD DATA
3096	016660	004537	007000		JSR	R5,SIMBCC		;CALCULATE A SOFTWARE BCC
3097	016664	000010			8.			;BASED
3098	016666	000000		2\$:	.WORD	0		;ON THESE
3099	016670	000000		35\$:	.WORD	0		;PARAMETERS
3100	016672	105200			INCB	R0		;UPDATE DATA
3101	016674	001364			BNE	1\$		;BR IF MORE TO GO
3102	016676	005000			CLR	R0		;CLR OUT DATA POINTER
3103	016700	005001			CLR	R1		;DITTO
3104	016702	004537	006746		JSR	R5,SETVEC		;SET UP INTERRUPTS
3105	016706	017116			4\$			;RECEIVER
3106	016710	017364			17\$			;TRANSMITTER
3107	016712	340	340		.BYTE	340,340		;LEVEL
3108								
3109	016714	052777	000400	162470	BIS	#MRESET,@TXCSR		;RESET THE DEVICE
3110	016722	004737	005044		JSR	PC,SMALL		;WAIT FOR RESET TO FINISH
3111	016726	012777	100026	162454	MOV	#DECMOD!26,@PARCSR		;LOAD THE MODE AND SYNC CHARACTER
3112	016734	052777	004000	162450	BIS	#SYSTST,@TXCSR		;ENTER SYSTEM TEST MODE
3113	016742	052777	000020	162434	BIS	#RCVEN,@RXCSR		;LOAD RCVEN
3114	016750	052777	000020	162434	BIS	#SEND,@TXCSR		;TURN ON TRANSMITTER
3115	016756	012777	000426	162430	MOV	#TSOM!26,@TXDBUF		;OUTPUT A SYNC CHAR
3116	016764	032777	004000	162422	64\$:	BIT	#TIMER,@TXDBUF	;CHECK THE TIMER BIT
3117	016772	001374			BNE	64\$		;BR IF SET
3118	016774	032777	004000	162412	65\$:	BIT	#TIMER,@TXDBUF	;CHECK THE TIMER BIT
3119	017002	001774			BEQ	65\$		;BR IF CLEAR
3120	017004	105777	162402		69\$:	TSTB	@TXCSR	;CHECK DONE
3121	017010	100375			BPL	69\$		;BR IF NOT SET
3122	017012	012777	000426	162374	MOV	#TSOM!26,@TXDBUF		;SEND SYNC
3123	017020	005037	177776		CLR	PS		
3124	017024	052777	000100	162352	BIS	#RINTEN,@RXCSR		;TURN ON INT ENABLES
3125	017032	052777	000100	162352	BIS	#TXINTE,@TXCSR		;DITTO
3126	017040				30\$:			
3127	017040	012737	000310	017070	MOV	#200.,74\$		;LOAD THE NUMBER
3128	017046	032777	004000	162340	72\$:	BIT	#TIMER,@TXDBUF	;CHECK THE TIMER BIT
3129	017054	001374			BNE	72\$		;BR IF SET
3130	017056	032777	004000	162330	73\$:	BIT	#TIMER,@TXDBUF	;CHECK THE BIT
3131	017064	001774			BEQ	73\$		;BR IF CLEAR
3132	017066	005327			DEC	(PC)+		;DECREMENT THE NUMBER
3133	017070	000310			74\$:	200.		;OF TIMES TO REPEAT
3134	017072	001365			BNE	72\$		;BR IF MORE TO GO
3135	017074	104023			HLT	23		;FAILED TO FINISH TEST
3136	017076				3\$:			
3137	017076	052777	000400	162306	BIS	#MRESET,@TXCSR		;RESET THE DEVICE
3138	017104	004737	005044		JSR	PC,SMALL		;WAIT FOR RESET TO FINISH
3139	017110	012706	001150		MOV	#STACK,SP		;RESET THE STACK
3140	017114	104400			SCOPE			;SCOPE THIS TEST
3141								
3142								
3143								
3144	017116	017702	162262		4\$:	;RECEIVER INT SVC ROUTINE		
3145	017122	017703	162260		MOV	@RXCSR,R2		;SAVE CSR
3146	017126	032702	004000		MOV	@RXDBUF,R3		;SAVE BUFFER
					BIT	#REACT,R2		;TEST RX ACTIVE

3147	017132	001004				BNE	5\$		:BR IF OK
3148	017134	104025				HLT	25		:ACTIVE NOT SET
3149	017136	012716	017076			MOV	#3\$, (SP)		:SETUP FOR RETURN
3150	017142	000433				BR	12\$		
3151	017144	032702	000200		5\$:	BIT	#RXDONE, R2		:TEST DONE
3152	017150	001004				BNE	6\$		:BR IF OK
3153	017152	104024				HLT	24		:FALSE INTERRUPT
3154	017154	012716	017076			MOV	#3\$, (SP)		:SETUP FOR RETURN
3155	017160	000424				BR	12\$		
3156	017162	005703			6\$:	TST	R3		:CHECK FOR ERROR
3157	017164	100004				BPL	7\$		:BR IF NO ERROR
3158	017166	104026				HLT	26		:DATA ERROR
3159	017170	012716	017076			MOV	#3\$, (SP)		:SET UP RETURN
3160	017174	000416				BR	12\$		
3161	017176	120103			7\$:	CMPB	R1, R3		:CHECK DATA
3162	017200	001404				BEQ	10\$		:BR IF OK
3163	017202	104022				HLT	22		:BAD DATA
3164	017204	012716	017076			MOV	#3\$, (SP)		:SETUP RETURN
3165	017210	000410				BR	12\$		
3166	017212	005201			10\$:	INC	R1		:UPDATE DATA
3167	017214	001003				BNE	11\$		:BR IF MORE TO GO
3168	017216	012716	017234			MOV	#13\$, (SP)	:SETUP TO FINISH TEST	
3169	017222	000403				BR	12\$		
3170	017224	012777	017324	162142	11\$:	MOV	#22\$, @DUPRVC		:SETUP NEW RETURN FOR INTERRUPT
3171	017232	000002			12\$:	RTI			:RETURN
3172	017234	105777	162144		13\$:	TSTB	@RXCSR		:TEST DONE
3173	017240	100375				BPL	13\$		:WAIT
3174	017242	017737	162140	001236		MOV	@RXDBUF, TEMP1		:GET DATA
3175	017250	105777	162130		14\$:	TSTB	@RXCSR		:CHECK DONE FOR HALF OF CRC
3176	017254	100375				BPL	14\$		:WAIT
3177	017256	017737	162124	001240		MOV	@RXDBUF, TEMP2		:MOVE IT
3178	017264	113737	001240	001237		MOVB	TEMP2, TEMP1+1		:COMBINE BCC CHARACTER
3179	017272	023737	007156	001236		CMP	CALBCC, TEMP1		:BR IF A MATCH
3180	017300	001401				BEQ	15\$		:AFTER CHECKING IT
3181	017302	104027				HLT	27		:CRC COMPARE ERROR--THE
3182									:SOFTWARE DOESN'T AGREE
3183									:WITH WHAT THE TRANSMITTER
3184									:SENT. SEE THE FRONT OF
3185									:THE LISTING FOR SPECIAL
3186									:CRC DEBUG AID TEST.
3187	017304	032737	010000	001240	15\$:	BIT	#CRCERR, TEMP2		:CHECK FOR ERROR
3188	017312	001001				BNE	16\$		:BR IF OK
3189	017314	104030				HLT	30		:HARDWARE DETECTED CRC ERROR
3190									:RECEIVER DOESN'T AGREE WITH
3191									:WHAT THE TRANSMITTER SENT
3192									:SEE FRONT OF LISTING FOR
3193									:SPECIAL CRC DEBUG AID
3194	017316	012716	017076		16\$:	MOV	#3\$, (SP)		:LOAD END OF TEST
3195	017322	000743				BR	12\$		:RETURN
3196	017324	017702	162054		22\$:	MOV	@RXCSR, R2		
3197	017330	017703	162052			MOV	@RXDBUF, R3		
3198	017334	005703				TST	R3		
3199	017336	100001				BPL	23\$		
3200	017340	104026				HLT	26		:ERROR
3201	017342	120103			23\$:	CMPB	R1, R3		
3202	017344	001401				BEQ	24\$		

```
3203 017346 104022          HLT      22          ;DATA COMPARE ERROR
3204 017350 105201      24$: INCB     R1
3205 017352 001327          BNE     12$
3206 017354 012777 017234 162012  MOV     #13$,@DUPRVC
3207 017362 000723          BR      12$
3208
3209          ;TRANSMITTER
3210 017364 0.0077 162024      17$: MOV     R0,@TXDBUF      ;PUSH OUT DATA
3211 017370 105200          INCB     R0          ;UPDATE IT
3212 017372 001014          BNE     21$          ;BR IF MORE
3213 017374 105777 162012      20$: TSTB     @TXCSR          ;CHECK FOR NEXT DONE
3214 017400 100375          BPL     20$          ;WAIT
3215 017402 052777 001000 162004  BIS     #TEOM,@TXDBUF      ;END MSG
3216 017410 042777 000120 161774  BIC     #SEND!TXINTE,@TXCSR      ;SHUT OF TRANSMITTER
3217 017416 012777 006774 161754  MOV     #NO,BTRAP,@DUPTVC      ;RESET VECTOR ADRS
3218 017424 012716 017040      21$: MOV     #30$, (SP)
3219 017430 000002          RTI          ;RETURN
3220
3221
3222
3223
3224
```

```
***** TEST 24 *****
*TEST TO PROVE THE DEVICE IDLES SYNCs AND
*WILL SHIFT OUT DATA AT THE APPROPRIATE TIME
*****
```

```
3225
3226
3227
3228
3229          :*****
3230          :*
3231          :TEST 24
3232          :*
3233          :*****
3234          :*****
3235 017432 012737 000024 001226  TST24: MOV     #24,@TSTNO
3236 017440 012737 017630 001216  MOV     #TST25,NEXT
3237 017446 052777 000400 161736  BIS     #MRESET,@TXCSR      ;RESET THE DEVICE
3238 017454 004737 005044          JSR     PC,SMALL          ;WAIT FOR RESET TO FINISH
3239 017460 052777 014000 161724  BIS     #MODE,@TXCSR      ;ENTER MAINT MODE
3240 017466 012777 000020 161710  MOV     #RCVEN,@RXCSR          ;TURN ON RECEIVER
3241 017474 012777 100026 161706  MOV     #DECMOD!26,@PARCSR      ;ENTER DECMODE AND SYNC CHAR
3242 017502 052777 000020 161702  BIS     #SEND,@TXCSR          ;TURN ON TRANSMITTER
3243 017510 012777 000426 161676  MOV     #TSOM!26,@TXDBUF      ;PUSH OUT SYNCs
3244 017516 104412 000044          PKCLK     ,36.
3245 017522 012777 000252 161664  MOV     #252,@TXDBUF      ;LOAD DATA
3246 017530 104412 000024          PKCLK     ,20.          ;PUSH OUT ANOTHER SYNC
3247 017534 105777 161644          TSTB     @RXCSR          ;CHECK TO SEE IF SYNC ARRIVED
3248 017540 100401          BMI      1$          ;BR IF YES
3249 017542 104021          HLT      21
3250 017544 017737 161636 001324  1$: MOV     @RXDBUF,DATA      ;GET THE REC CHAR
3251 017552 122737 000026 001324  CMPB     #26,DATA          ;CHECK FOR SYNC
3252 017560 001401          BEQ      2$          ;BR IF MATCH
3253 017562 104021          HLT      21          ;FAILED TO RECEIVE THIRD SYNC
3254 017564 042777 000020 161620  2$: BIC     #SEND,@TXCSR      ;TURN OFF TRANSMITTER
3255 017572 104412 000016          PKCLK     ,14.          ;PUSH OUT DATA
3256 017576 105777 161602          TSTB     @RXCSR          ;CHECK FOR REC DATA
3257 017602 100401          BMI      3$          ;BR IF YES
3258 017604 104026          HLT      26          ;FAILED TO GET A DATA DONE
```



```
3259 017606 017737 161574 001324 3$: MOV @RXDBUF,DATA ;GET THE DATA
3260 017614 122737 000252 001324 CMPB #252,DATA ;COMPARE IT
3261 017622 001401 BEQ 4$ ;BR IF A MATCH
3262 017624 104022 HLT 22 ;DATA ERROR
3263 017626 104400 4$: SCOPE ;SCOPE THIS TEST
3264
3265
3266 ***** TEST 25 *****
3267 ;*TEST TO PROVE THE DEVICE WILL
3268 ;*WORK WITH EVERY POSSIBLE SYNC CHARACTER
3269 ;*****
3270 ;*****
3271 ;*
3272 ; TEST 25
3273 ;*
3274 ;*****
3275 ;*****
3276 017630 012737 000025 001226 TST25: MOV #25,@TSTNO
3277 017636 012737 020110 001216 MOV #TST26,NEXT
3278 017644 012737 000340 177776 MOV #340,PS
3279 017652 012702 000003 MOV #3,R2 ;SET UP # OF SYNC
3280 017656 005037 001236 CLR TEMP1
3281 017662 005037 001240 CLR TEMP2
3282 017666 005000 CLR R0 ;CLEAR OUT DATA
3283 017670 005001 CLR R1 ;CLEAR SYNC
3284 017672 052737 000400 001236 BIS #TSOM,TEMP1 ;LOAD TSOM
3285 017700 052737 101000 001240 BIS #DECMODE!CRCEN,TEMP2 ;LOAD DEC MODE
3286 017706 1$:
3287 017706 052777 000400 161476 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3288 017714 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3289 017720 052777 000020 161456 BIS #RCVEN,@RXCSR
3290 017726 013777 001240 161454 MOV TEMP2,@PARCSR ;LOAD THE MODE AND SYNC CHARACTER
3291 017734 052777 004020 161450 BIS #SYSTST!SEND,@TXCSR ;TURN ON MODE AND TRANSMITTER
3292 017742 105777 161444 2$: TSTB @TXCSR ;TEST FOR DONE
3293 017746 100375 BPL 2$ ;WAIT
3294 017750 013777 001236 161436 MOV TEMP1,@TXDBUF ;LOAD TSOM AND SYNC CHAR
3295 017756 005302 DEC R2 ;DECREMENT # OF SYNCs TO SEND
3296 017760 001370 BNE 2$ ;BR IF MORE TO GO
3297 017762 7$:
3298 017762 032777 004000 161424 64$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3299 017770 001374 BNE 64$ ;BR IF SET
3300 017772 032777 004000 161414 65$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3301 020000 001774 BEQ 65$ ;BR IF CLEAR
3302 020002 105777 161376 TSTB @RXCSR
3303 020006 100405 BMI 3$
3304 020010 005200 INC R0
3305 020012 022700 000040 CMP #40,R0
3306 020016 001361 BNE 7$
3307 020020 104015 HLT 15 ;DONE FAILED TO SET AFTER THIRD SYNC
3308 020022 017737 161360 001244 3$: MOV @RXDBUF,TEMP4 ;GET THE BUFFER
3309 020030 032737 004000 001242 BIT #REACT,TEMP3 ;TEST ACTIVE
3310 020036 001001 BNE 4$ ;BR IF SET
3311 020040 104025 HLT 25 ;ACTIVE FAILED TO SET
3312 020042 005737 001244 4$: TST TEMP4 ;TEST FOR ERROR
3313 020046 100001 BPL 5$ ;BR IF NO ERROR
3314 020050 104026 HLT 26 ;HARDWARE ERROR
```

```
3315 020052 123737 001236 001244 5$: CMPB TEMP1,TEMP4 ;COMPARE SYNCs
3316 020060 001401 BEQ 6$ ;BR IF OK
3317 020062 104022 HLT 22 ;SYNC CHAR DOES NOT MATCH SENT
3318 020064 6$:
3319 020064 012702 000003 MOV #3,R2 ;SET UP FOR NEXT SYNC
3320 020070 005000 CLR R0 ;DITTO
3321 020072 105201 INCB R1 ;DITTO
3322 020074 110137 001236 MOVB R1,TEMP1
3323 020100 110137 001240 MOVB R1,TEMP2
3324 020104 001300 BNE 1$ ;BR IF MORE TO GO
3325 020106 104400 SCOPE ;SCOPE THIS TEST
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339 020110 012737 000026 001226 TST26: MOV #26,@TSTNO
3340 020116 012737 020450 001216 MOV #TST27,NEXT
3341 020124 012737 000340 177776 MOV #340,PS ;RAISE PROCESSOR STATUS
3342 020132 004537 006746 JSR R5,SETVEC ;SETUP VECTORS
3343 020136 020412 6$ ;RECEIVER
3344 020140 020344 3$ ;TRANSMITTER
3345 020142 340 340 .BYTE 340,340 ;LEVEL
3346 020144 005001 CLR R1 ;CLEAR CHAR COUNT
3347
3348 020146 052777 000400 161236 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3349 020154 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3350 020160 012777 100026 161222 MOV #DECMOD!26,@PARCSR ;LOAD THE MODE AND SYNC CHARACTER
3351 020166 052777 004000 161216 BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
3352 020174 052777 000020 161202 BIS #RCVEN,@RXCSR ;LOAD RCVEN
3353 020202 052777 000020 161202 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
3354 020210 012777 000426 161176 MOV #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
3355 020216 105777 161170 64$: TSTB @TXCSR ;CHECK DONE
3356 020222 100375 BPL 64$ ;BR IF NOT SET
3357 020224 012777 000426 161162 MOV #TSOM!26,@TXDBUF ;SEND SYNC
3358 020232 105777 161154 65$: TSTB @TXCSR ;CHECK DONE
3359 020236 100375 BPL 65$ ;BR IF NOT SET
3360 020240 012777 000426 161146 MOV #TSOM!26,@TXDBUF ;SEND SYNC
3361 020246 005037 177776 CLR PS ;LOWER PROCESSOR STATUS
3362 020252 052777 000100 161124 BIS #RINTEN,@RXCSR ;TURN ON INTERRUPT ENABLES
3363 020260 052777 000100 161124 BIS #TXINTE,@TXCSR ;DITTO
3364 020266 1$:
3365 020266 012737 000040 020316 MOV #32..70$ ;LOAD THE NUMBER
3366 020274 032777 004000 161112 68$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3367 020302 001374 BNE 68$ ;BR IF SET
3368 020304 032777 004000 161102 69$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3369 020312 001774 BEQ 69$ ;BR IF CLEAR
3370 020314 005327 DEC (PC)+ ;DECREMENT THE NUMBER
```



```
3371 020316 000040      70$: 32.      ;OF TIMES TO REPEAT
3372 020320 001365      BNE      68$      ;BR IF MORE TO GO
3373 020322 104023      HLT      23      ;FAILED TO FINISH TEST
3374 020324
3375 020324 052777 000400 161060 2$: BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3376 020332 004737 005044      JSR      PC,SMALL ;WAIT FOR RESET TO FINISH
3377 020336 012706 001150      MOV      #STACK,SP ;RESET THE STACK
3378 020342 104400      SCOPE
3379
3380
3381      ; INTERRUPT SERVICE ROUTINES
3382      ; TRANSMITTER
3383 020344 005000      3$: CLR      R0      ;CLEAR DATA
3384 020346 010077 161042      MOV      R0,@TXDBUF ;LOAD DATA TO BUFFER
3385 020352 012777 020362 161020      MOV      #4$,@DUPTVC ;SETUP FOR NEXT INTERRUPT
3386 020360 000411      BR      5$      ;LEAVE
3387 020362 012777 001000 161024 4$: MOV      #TEOM,@TXDBUF ;END OF MSG--OUTPUT CRC
3388 020370 042777 000120 161014      BIC      #SEND!TXINTE,@TXCSR ;TURN OFF THE
3389 020376 012777 006774 160774      MOV      #NO.BTRAP,@DUPTVC ;TRANSMITTER AND TXINTEN
3390 020404 012716 020266      5$: MOV      #1$, (SP) ;SETUP TO RETURN
3391 020410 000002      RTI      ;RETURN
3392
3393      ; RECEIVER
3394 020412 017737 160770 001324 6$: MOV      @RXDBUF,DATA ;GET THE DATA
3395 020420 005201      INC      R1      ;CHECK FOR LAST CHAR
3396 020422 022701 000004      CMP      #4,R1      ;AND BRANCH IF
3397 020426 001007      BNE      10$      ;NOT YET
3398 020430 032737 010000 001324      BIT      #CRCERR,DATA ;CHECK FOR CRC ERROR
3399 020436 001401      BEQ      7$      ;BR IF CRC ERROR SEEN
3400 020440 104014      HLT      14      ;FAILED TO CATCH CRC ERROR!!!!
3401 020442 012716 020324      7$: MOV      #2$, (SP) ;FINISH TEST
3402 020446 000002      10$: RTI      ;RETURN
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
```

```
***** TEST 27 *****
;THIS TEST PROVES THE DEVICE WILL HANDLE THE
;DDCMP PROTOCOL. SEND AND RECEIVE SYNCs,
;FOLLOWED BY DATA,BCC,DATA AND FINAL BCC.
*****
```

```
*****
;
; TEST 27
;
*****
```

```
3417 020450 012737 000027 001226 TST27: MOV      #27,@TSTNO
3418 020456 012737 021454 001216      MOV      #TST30,NEXT
3419 020464 012737 000340 177776      MOV      #340,PS
3420 020472 004537 006746      JSR      R5,SETVEC ;RAISE PROCESSOR STATUS
3421 020476 021104      10$      ;SET UP VECTORS
3422 020500 020726      2$      ;BASED ON
3423 020502 340 340      .BYTE 340,340 ;THESE
3424 020504 005037 001236      CLR      TEMP1 ;PARAMETERS
3425 020510 005037 001240      CLR      TEMP2
3426 020514 005037 001242      CLR      TEMP3
```

```
3427 020520 005037 001244 CLR TEMP4
3428 020524 005037 001246 CLR TEMP5
3429
3430 020530 052777 000400 160654 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3431 020536 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3432 020542 012777 100026 160640 MOV #DECMOD!26,@PARCSR ;LOAD THE MODE AND SYNC CHARACTER
3433 020550 052777 004000 160634 BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
3434 020556 052777 000420 160620 BIS #RCVEN!STPSYN,@RXCSR ;LOAD RCVEN!STPSYN
3435 020564 052777 000020 160620 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
3436 020572 012777 000426 160614 MOV #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
3437 020600 105777 160606 64$: TSTB @TXCSR ;CHECK DONE
3438 020604 100375 BPL 64$ ;BR IF NOT SET
3439 020606 012777 000426 160600 MOV #TSOM!26,@TXDBUF ;SEND SYNC
3440 020614 105777 160572 65$: TSTB @TXCSR ;CHECK DONE
3441 020620 100375 BPL 65$ ;BR IF NOT SET
3442 020622 012777 000426 160564 MOV #TSOM!26,@TXDBUF ;SEND SYNC
3443 020630 052777 000100 160546 BIS #RINTEN,@RXCSR ;TURN ON INTERRUPTS
3444 020636 052777 000100 160546 BIS #TXINTE,@TXCSR ;DITTO
3445 020644 005037 177776 CLR PS ;LOWER PROCESSOR STATUS
3446 020650
3447 020650 012737 000144 020700 MOV #100.,70$ ;LOAD THE NUMBER
3448 020656 032777 004000 160530 68$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3449 020664 001374 BNE 68$ ;BR IF SET
3450 020666 032777 004000 160520 69$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3451 020674 001774 BEQ 69$ ;BR IF CLEAR
3452 020676 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3453 020700 000144 70$: 100. ;OF TIMES TO REPEAT
3454 020702 001365 BNE 68$ ;BR IF MORE TO GO
3455 020704 104023 HLT 23 ;FAILED TO FINISH TEST
3456 020706
3457 020706 052777 000400 160476 1$: BIS #MRESET,@TXCSR ;RESET THE DEVICE
3458 020714 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3459 020720 012706 001150 MOV #STACK,SP ;RESET THE STACK
3460 020724 104400 SCOPE ;SCOPE THIS TEST
3461
3462 ;INTERRUPT SERVICE ROUTINES
3463 ;TRANSMITTER
3464
3465 020726 012777 000252 160460 2$: MOV #252,@TXDBUF ;LOAD FIRST DATA CHAR
3466 020734 012737 000026 001236 MOV #26,TEMP1 ;LOAD DATA
3467 020742 012777 020752 160430 MOV #3$,@DUPTVC ;RELOAD VECTOR
3468 020750 000452 BR 7$ ;LEAVE
3469 020752 013777 001236 160434 3$: MOV TEMP1,@TXDBUF ;MOV DATA TO BUFFER
3470 020760 105237 001236 INCB TEMP1 ;UPDATE DATA
3471 020764 122737 000032 001236 CMPB #32,TEMP1 ;CHECK FOR DONE
3472 020772 001041 BNE 7$ ;BR IF MORE TO SEND
3473 020774 012777 021004 160376 MOV #4$,@DUPTVC ;RELOAD VECTOR
3474 021002 000435 BR 7$ ;RETURN
3475 021004 012777 001000 160402 4$: MOV #TEOM,@TXDBUF ;PUT OUT BCC
3476 021012 012777 021022 160360 MOV #5$,@DUPTVC ;RELOAD VECTOR
3477 021020 000426 BR 7$ ;RETURN
3478 021022 013777 001240 160364 5$: MOV TEMP2,@TXDBUF ;LOAD DATA
3479 021030 105237 001240 INCB TEMP2 ;UPDATE DATA
3480 021034 122737 000100 001240 CMPB #100,TEMP2 ;CHECK FOR FINISH
3481 021042 001015 BNE 7$ ;BR IF MORE TO GO
3482 021044 012777 021054 160326 MOV #6$,@DUPTVC ;RELOAD VECTOR
```


3483	021052	000411				BR	7\$	:RETURN
3484	021054	012777	001000	160332	6\$:	MOV	#TEOM,@TXDBUF	:PUSH OUT DATA BCC
3485	021062	042777	000120	160322		BIC	#SEND!TXINTE,@TXCSR	:SHUT DOWN TRANSMITTER
3486	021070	012777	006774	160302		MOV	#NO.BTRAP,@DUPTVC	:RESET VECTOR
3487	021076	012716	020650		7\$:	MOV	#100\$,(SP)	:SETUP RETURN
3488	021102	000002				RTI		:RETURN
3489								
3490								
3491								
3492	021104	017737	160274	001242	10\$:	MOV	@RXCSR,TEMP3	:SAVE CSR
3493	021112	017737	160270	001244		MOV	@RXDBUF,TEMP4	:SAVE BUFFER
3494	021120	105737	001242			TSTB	TEMP3	:CHECK FOR DONE
3495	021124	100401				BMI	11\$	:BR IF SET
3496	021126	104024				HLT	24	:FALSE INTERRUPT
3497	021130	005737	001244		11\$:	TST	TEMP4	:CHECK FOR ERROR
3498	021134	100001				BPL	12\$	:BR IF NO ERROR
3499	021136	104026				HLT	26	:RECEIVER ERROR
3500	021140	122737	000252	001244	12\$:	CMPB	#252,TEMP4	:CHECK DATA
3501	021146	001401				BEQ	13\$	:BR IF A MATCH
3502	021150	104022				HLT	22	:DATA COMPARE ERROR
3503	021152	012737	000026	001246	13\$:	MOV	#26,TEMP5	:LOAD NEXT EXPECTED
3504	021160	012777	021170	160206		MOV	#14\$,@DUPRVC	:RELOAD VECTOR
3505	021166	000531				BR	26\$	:LEAVE
3506	021170	017737	160212	001244	14\$:	MOV	@RXDBUF,TEMP4	:GET DATA
3507	021176	005737	001244			TST	TEMP4	:CHECK FOR ERROR
3508	021202	100001				BPL	15\$	:BR IF NO ERROR
3509	021204	104026				HLT	26	:DATA ERROR
3510	021206	123737	001246	001244	15\$:	CMPB	TEMP5,TEMP4	:CHECK DATA
3511	021214	001401				BEQ	16\$	:BR IF A MATCH
3512	021216	104022				HLT	22	:DATA COMPARE ERROR
3513	021220	105237	001246		16\$:	INCB	TEMP5	:UPDATE DATA
3514	021224	122737	000032	001246		CMPB	#32,TEMP5	:CHECK FOR FIRST PART FINISH
3515	021232	001107				BNE	26\$	:BR IF MORE TO GO
3516	021234	012777	021244	160132		MOV	#17\$,@DUPRVC	:SET UP NEXT VECTOR
3517	021242	000503				BR	26\$	:LEAVE
3518	021244	017737	160136	001244	17\$:	MOV	@RXDBUF,TEMP4	:GET THE BUFFER
3519	021252	005737	001244			TST	TEMP4	:TEST FOR ERROR
3520	021256	100001				BPL	.+4	:BR IF OK
3521	021260	104026				HLT	26	:RECEIVER ERROR
3522	021262	012777	021272	160104		MOV	#18\$,@DUPRVC	:RELOAD THE VECTOR
3523	021270	000470				BR	26\$	:LEAVE
3524	021272	017737	160110	001324	18\$:	MOV	@RXDBUF,DATA	:GET DATA
3525	021300	032737	010000	001324		BIT	#CRCERR,DATA	:CHECK FOR CRC ERROR
3526	021306	001001				BNE	19\$	:BR IF OK
3527	021310	104014				HLT	14	:CRC ERROR!!!!!!
3528	021312	012777	021326	160054	19\$:	MOV	#20\$,@DUPRVC	:SET UP VECTOR
3529	021320	005037	001330			CLR	MIND	:SETUP FOR NEXT DATA
3530	021324	000452				BR	26\$	:LEAVE
3531	021326	017737	160054	001244	20\$:	MOV	@RXDBUF,TEMP4	:GET DATA
3532	021334	005737	001244			TST	TEMP4	:CHECK FOR ERROR
3533	021340	100001				BPL	21\$	:BR IF NO ERROR
3534	021342	104026				HLT	26	:RECEIVER ERROR
3535	021344	123737	001330	001244	21\$:	CMPB	MIND,TEMP4	:CHECK DATA
3536	021352	001401				BEQ	22\$	:BR IF A MATCH
3537	021354	104022				HLT	22	:DATA ERROR
3538	021356	105237	001330		22\$:	INCB	MIND	:UPDATE SOFTWARE DATA

3539	021362	122737	000100	001330	CMPB	#100,MIND	:CHECK FOR FINISH
3540	021370	001030			BNE	26\$	:BR IF MORE TO GO
3541	021372	012777	021402	157774	MOV	#23\$,@DUPRVC	:RELOAD FINAL VECTOR
3542	021400	000424			BR	26\$	:LEAVE
3543	021402	017737	160000	001244	23\$: MOV	@RXDBUF,TEMP4	:GET DATA
3544	021410	005737	001244		TST	TEMP4	:CHECK FOR ERROR
3545	021414	100001			BPL	24\$	:BR IF OK
3546	021416	104026			HLT	26	:RECEIVER ERROR ON FIRST OCTET
3547							:OF SECOND BCC
3548	021420	105777	157760		24\$: TSTB	@RXCSR	:TEST DONE
3549	021424	100375			BPL	24\$	:BR IF NOT SET
3550	021426	017737	157754	001324	MOV	@RXDBUF,DATA	:GET SECOND BCC OCTET
3551	021434	032737	010000	001324	BIT	#CRCERR,DATA	:CHECK FOR BCC ERROR
3552	021442	001001			BNE	25\$	:BR IF OK
3553	021444	104014			HLT	14	:BCC ERROR ON SECOND PART OF MSG
3554	021446	012716	020706		25\$: MOV	#1\$, (SP)	:SETUP TO FINISH TEST
3555	021452	000002			26\$: RTI		:RETURN

***** TEST 30 *****
:THIS TEST IS AN AID FOR DEBUGGING CRC
:ERRORS. A CHARACTER IS LOADED INTO THE
:DUP AND PUSHED OUT BIT BY BIT WHILE
:ALLOWING THE OPERATOR TO MONITOR THE CRC
:CHARACTER AS IT IS GENERATED. THE DATA CHARACTER
:CAN ALSO BE CHANGED BY THE OPERATOR.
:PUT SW09=1 TO LOCK ON BITS. TO CONTINUE HIT
:ANY KEY ON THE TTY. AFTER 16 TIMES PUT DOWN SW09 TO LEAVE

: TEST 30

3575	021454	012737	000030	001226	TST30: MOV	#30,@TSTNO	
3576	021462	012737	002764	001216	MOV	#.EOP,NEXT	
3577	021470	052777	000400	157714	BIS	#MRESET,@TXCSR	:RESET THE DEVICE
3578	021476	004737	005044		JSR	PC,SMALL	:WAIT FOR RESET TO FINISH
3579	021502	012737	120001	007152	MOV	#CRC16,XPOLY	:LOAD THE POLYNOMIAL
3580	021510	012737	000125	021656	MOV	#125,3\$	:LOAD DATA TO SOFTWARE BCC-CHANGE CHARACTER HERE
3581	021516	013737	021656	001252	MOV	3\$,SAVR1	
3582	021524	005037	007156		CLR	CALBCC	:CLEAR FOR SOFTWARE BCC
3583	021530	013737	007156	021660	MOV	CALBCC,4\$	
3584	021536	005037	001242		CLR	TEMP3	
3585	021542	005037	001244		CLR	TEMP4	:CLEAR BIT COUNTER
3586	021546	005037	001246		CLR	TEMP5	
3587	021552	012777	100026	157630	MOV	#DECMOD!26,@PARCSR	:LOAD MODE AND SYNC CHARACTER
3588	021560	052777	014000	157624	BIS	#MODE,@TXCSR	:ENTER MAINT MODE-PROGRAM CLOCKING
3589	021566	052777	000420	157610	BIS	#RCVEN!STPSYN,@RXCSR	:TURN ON RECEIVER
3590	021574	052777	000020	157610	BIS	#SEND,@TXCSR	:TURN ON TRANSMITTER
3591	021602	012777	000426	157604	MOV	#TSOM!26,@TXDBUF	:LOAD A SYNC
3592	021610	104412	000044		PKCLK	36.	:PUSH OUT 2 SYNC
3593	021614	013777	021656	157572	MOV	3\$,@TXDBUF	:LOAD DATA
3594	021622	104412	000020		PKCLK	.16.	:PUSH OUT ANOTHER SYNC


```
3595 021626 104412 000002 1$: PKCLK ,2 ;PUSH OUT A BIT
3596 021632 013737 001244 001254 MOV TEMP4,SAVR2 ;SET UP TO TYPE
3597 021640 005237 001242 INC TEMP3
3598 021644 005237 001244 INC TEMP4 ;UPDATE BIT COUNTER
3599 021650 004537 007000 2$: JSR R5,SIMBCC ;CALCULATE SOFTWARE BCC BASED ON THESE PARAMETERS
3600 021654 000001 1 ;SHIFTS
3601 021656 000000 3$: .WORD 0 ;DATA
3602 021660 000000 4$: .WORD 0 ;PREVIOUS BCC
3603 021662 004737 021760 JSR PC,5$ ;CHECK TO SEE IF WE SHOULD WAIT FOR SCOPING
3604 021666 000241 CLC ;CLEAR FOR NEXT ROTATE
3605 021670 106037 021656 RORB 3$ ;SET UP THE NEXT BIT
3606 021674 013737 007156 021660 MOV CALBCC,4$ ;FOR THE SOFTWARE BCC
3607 021702 022737 000006 001244 CMP #6,TEMP4
3608 021710 001002 BNE .+6
3609 021712 005077 157476 CLR @TXDBUF
3610 021716 022737 000014 001242 CMP #12,TEMP3
3611 021724 001003 BNE 12$
3612 021726 012777 001000 157460 MOV #TEOM,@TXDBUF
3613 021734 022737 000020 001244 12$: CMP #16,TEMP4 ;ALL DONE WITH THE CHARACTER?
3614 ;INCREASE COMPARE # TO FORCE
3615 ;CRC OUT OF THE GENERATOR
3616 021742 001331 BNE 1$ ;BR IF MORE TO GO
3617 021744 052777 000400 157440 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3618 021752 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3619 021756 104400 SCOPE ;SCOPE THIS TEST
3620
3621 021760 032777 001000 157214 5$: BIT #SW09,@SWR ;SW09=1?
3622 021766 001432 BEQ 6$ ;BR IF NO
3623 021770 013704 007156 MOV CALBCC,R4 ;THE DATA CHARACTER IS ALWAYS
3624 021774 012737 000001 001256 MOV #1,SAVR3 ;FOLLOWED BY A ZERO. THE DATA IN
3625 022002 000241 CLC ;CRC SHOWS WHICH BIT OF THE 2 CHARS
3626 022004 006004 11$: ROR R4 ;IS BEING GENERATED
3627 022006 006137 001256 ROL SAVR3
3628 022012 103374 BCC 11$
3629 022014 105737 001246 TSTB TEMP5
3630 022020 001006 BNE 10$
3631 022022 104402 023152 TYPE ,EM17 ;TYPE MSG
3632 022026 104402 023201 TYPE ,MH1 ;TYPE HEADER
3633 022032 105137 001246 COMB TEMP5
3634 022036 104410 10$: CONVRT
3635 022040 023522 DT1
3636 022042 105777 157136 7$: TSTB @TKCSR ;CHECK TTY DONE--GO SCOPE THE CRC GENERATOR
3637 022046 100375 BPL 7$ ;BR IF NOT YET
3638 022050 017701 157132 MOV @TKDBR,R1 ;READ THE BUFFER
3639 022054 000207 6$: RTS PC ;RETURN
3640
3641
```

3642

(1)	022056	042377	050125	043040	EM1:	.ASCIIZ	<377>/DUP FAILED TO INTERRUPT IN TIME /	
(1)	022120	042377	052101	020101	EM2:	.ASCIIZ	<377>/DATA COMPARE ERROR /	
(1)	022145	377	047105	020104	EM3:	.ASCIIZ	<377>/END OF MESSAGE /	
(1)	022166	041377	041503	042440	EM4:	.ASCIIZ	<377>/BCC ERROR /	
(1)	022202	052377	040522	051516	EM5:	.ASCIIZ	<377>/TRANSMITTER DONE /	
(1)	022225	106	044501	042514	EM6:	.ASCIIZ	/FAILED TO SET /	
(1)	022244	051377	041505	044505	EM7:	.ASCIIZ	<377>/RECEIVER INTERRUPT IN HALF-DUPLEX /	
(1)	022310	043377	046101	042523	EM10:	.ASCIIZ	<377>/FALSE INTERRUPT /	
(1)	022332	040777	047502	052122	EM11:	.ASCIIZ	<377>/ABORT SEQUENCE ERROR /	
(1)	022361	377	052123	051101	EM12:	.ASCIIZ	<377>/START OF MESSAGE /	
(1)	022404	052777	042516	050130	EM13:	.ASCIIZ	<377>/UNEXPECTED RECEIVER INTERRUPT /	
(1)	022444	052777	042516	050130	EM14:	.ASCIIZ	<377>/UNEXPECTED TRANSMITTER INTERRUPT /	
(1)	022507	377	051124	047101	EM20:	.ASCIIZ	<377>/TRANSMITTER DONE /	
(1)	022532	051377	041505	044505	EM21:	.ASCIIZ	<377>/RECEIVER DONE /	
(1)	022552	052377	040522	051516	EM22:	.ASCIIZ	<377>/TRANSMITTER ACTIVE /	
(1)	022577	377	042522	042503	EM23:	.ASCIIZ	<377>/RECEIVER ACTIVE /	
(1)	022621	106	044501	042514	EM24:	.ASCIIZ	/FAILED TO SET. /	
(1)	022641	106	044501	042514	EM25:	.ASCIIZ	/FAILED TO CLEAR. /	
(1)	022663	377	054523	041516	EM26:	.ASCIIZ	<377>/SYNC ERROR /	
(1)	022700	042377	052101	020101	EM27:	.ASCIIZ	<377>/DATA ERROR /	
(1)	022715	377	042504	044526	EM30:	.ASCIIZ	<377>/DEVICE FAILED TO INTERRUPT IN TIME /	
(1)	022762	043377	046101	042523	EM31:	.ASCIIZ	<377>/FALSE INTERRUPT /	
(1)	023004	052377	040522	051516	EM32:	.ASCIIZ	<377>/TRANSMITTER BCC ERROR IN DEC MODE /	
(1)	023051	377	042522	042503	EM33:	.ASCIIZ	<377>/RECEIVER BCC ERROR IN DEC MODE /	
(1)	023113	377	042522	042503	EM15:	.ASCIIZ	<377>/RECEIVER ERROR /	
(1)	023134	041377	041503	042440	EM16:	.ASCIIZ	<377>/BCC ERROR!! /	
(1)	023152	041777	041522	043440	EM17:	.ASCIIZ	<377>/CRC GENERATOR STATUS /	
(1)	023201	377	040504	040524	MH1:	.ASCIIZ	<377>/DATA CHAR DATA BIT IN CRC GEN.	CRC FOR THIS BIT /
(1)								
(1)								
(1)	023274							
(1)	023274	000000			.ERRTAB:	0		
(1)	023276	000000				0		
(1)	023300	000000				0		
(1)	023302	022056			EM1			
(1)	023304	000000			0		:HALT 1	
(1)	023306	000000			0			
(1)								
(1)	023310	022120			EM2			
(1)	023312	000000			0		:HALT 2	
(1)	023314	000000			0			
(1)								
(1)	023316	022145			EM3			
(1)	023320	022225			EM6		:HALT 3	
(1)	023322	000000			0			
(1)								
(1)	023324	022166			EM4			
(1)	023326	000000			0		:HALT 4	
(1)	023330	000000			0			
(1)								
(1)	023332	022202			EM5			
(1)	023334	022225			EM6		:HALT 5	
(1)	023336	000000			0			
(1)								
(1)	023340	022244			EM7			

(1)	023342	000000	0		;HALT 6
(1)	023344	000000	0		
(1)					
(1)	023346	022310	EM10		
(1)	023350	000000	0		;HALT 7
(1)	023352	000000	0		
(1)					
(1)	023354	022332	EM11		
(1)	023356	000000	0		;HALT 10
(1)	023360	000000	0		
(1)					
(1)	023362	022361	EM12		
(1)	023364	022225	EM6		;HALT 11
(1)	023366	000000	0		
(1)					
(1)	023370	022404	EM13		
(1)	023372	000000	0		;HALT 12
(1)	023374	000000	0		
(1)					
(1)	023376	022444	EM14		
(1)	023400	000000	0		;HALT 13
(1)	023402	000000	0		
(1)					
(1)	023404	023134	EM16		
(1)	023406	000000	0	;HALT14	
(1)	023410	000000	0		
(1)					
(1)	023412	022532	EM21		
(1)	023414	022621	EM24	;HALT15	
(1)	023416	000000	0		
(1)					
(1)					
(1)	023420	022507	EM20		
(1)	023422	022621	EM24	;HALT16	
(1)	023424	000000	0		
(1)					
(1)	023426	022552	EM22		
(1)	023430	022621	EM24	;HALT17	
(1)	023432	000000	0		
(1)					
(1)	023434	022552	EM22		
(1)	023436	022641	EM25	;HALT20	
(1)	023440	000000	0		
(1)					
(1)	023442	022663	EM26		
(1)	023444	000000	0	;HALT21	
(1)	023446	000000	0		
(1)					
(1)	023450	022700	EM27		
(1)	023452	000000	0	;HALT22	
(1)	023454	000000	0		
(1)					
(1)	023456	022715	EM30		
(1)	023460	000000	0	;HALT23	
(1)	023462	000000	0		
(1)					

(1)	023464	022762		EM31	
(1)	023466	000000		0	;HALT24
(1)	023470	000000		0	
(1)					
(1)	023472	022577		EM23	
(1)	023474	022621		EM24	;HALT25
(1)	023476	000000		0	
(1)					
(1)	023500	023113		EM15	
(1)	023502	000000		0	;HALT26
(1)	023504	000000		0	
(1)					
(1)	023506	023004		EM32	
(1)	023510	000000		0	;HALT 27
(1)	023512	000000		0	
(1)					
(1)	023514	023051		EM33	
(1)	023516	000000		0	;HALT 30
(1)	023520	000000		0	
(1)	023522	000003		3	
(1)	023524	006	021	DT1:	.BYTE 6,17.
(1)	023526	001252		SAVR1	
(1)	023530	006	017	.BYTE	6,15.
(1)	023532	001254		SAVR2	
(1)	023534	006	002	.BYTE	6,2
(1)	023536	001256		SAVR3	
(1)					
(1)	023540			CORMAX:	
3643		000001		.END	

CZDPD-DO
CZDPDD.P11

MACY11 30A(1052)
25-JUL-79 15:01

25-JUL-79 15:19 PAGE 77

K 6

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0075

ADRCNT= 003735	1283*	1319*	1328#											
BCCFBK 007154	1714*	1720*	1723*	1726	1730	1742#								
BINWRD 004240	1370*	1371	1408#											
BITW = 002000	760#													
BIT0 = 000001	622#	742	1718											
BIT1 = 000002	621#	741												
BIT10 = 002000	612#	732	747	760	780									
BIT11 = 004000	611#	731	759	779										
BIT12 = 010000	610#	730	746	753	758	778								
BIT13 = 020000	609#	729	757	777										
BIT14 = 040000	608#	728	745	756	776	1042	1190							
BIT15 = 100000	607#	727	744	751	755	775	1039							
BIT2 = 000004	620#	740	1128											
BIT3 = 000010	619#	739	766											
BIT4 = 000020	618#	738	765											
BIT5 = 000040	617#	737												
BIT6 = 000100	616#	736	764											
BIT7 = 000200	615#	735	763											
BIT8 = 000400	614#	734	749	762	782									
BIT9 = 001000	613#	733	748	752	761	781								
BRW 003306	1134	1212#												
BRX 003310	1135	1213#												
CALBCC 007156	1736*	1743#	2071*	2072	2166*	2167	2197*	2198	2292*	2293	2521*	2522	2576*	
	2577	3093*	3094	3179	3582*	3583	3606	3623						
CARDET= 010000	730#													
CHRCNT 004236	1368*	1372	1388*	1406#	1407									
CLK = 020000	757#	1538	1541											
CLK.A 001433	868#	1654												
CLRVEC 005104	1563#													
CNVRT = 104411	835#	1151	1153	1155	1157	1481	1483							
CONVRT= 104410	833#	1087	1497	3634										
CORMAX 023540	3642#	3643												
CRcen = 001000	752#	1980	2397	2613	2653	2696	2719	2766	2814	2876	2989	3285		
CRcERR= 010000	746#	2170	2296	3187	3398	3525	3551							
CRC.CC= 102010	1745#	2070	2196	2520										
CRC16 = 120001	1744#	3092	3579											
CREAM 001316	722#	949*	1608*	1609	1611*	1616	1617*	1618	1621*					
CTS = 020000	729#													
CYCLE 006224	1136	1171	1172	1598#										
DATA 001324	789#	2018*	2019	2125*	2126	2158*	2162*	2167	2251*	2252	2284*	2288*	2293	
	2562*	2563	2572*	2575*	2577	3250*	3251	3259*	3260	3394*	3398	3524*	3525	
	3550*	3551												
DATABP 004612	1470*	1473	1495	1498#										
DATAHD 004600	1469*	1491	1494#											
DECMOD= 100000	751#	2613	2653	2696	2719	2766	2814	2876	2989	3111	3241	3285	3350	
	3432	3587												
DEVADR 003732	1281*	1316	1326#											
DISPLA 001200	666#	962*	1148*											
DISPRE 000174	654#	962												
DSCA = 100000	727#													
DSCB = 000001	742#													
DSINTE= 000040	737#													
DSR = 001000	733#													
DTR = 000002	741#													
DT1 023522	3635	3642#												
DUPACT 001310	716#	1052*	1076*	1109*	1110	1598	1603							

[illegible]

MPASSX	005355	1154	1574#												
MPOWER	005141	1559	1574#												
MQM	005132	1267	1574#	1683											
MR	005202	1137	1574#	1678											
MRESET=	000400	623#	762#	1763	1797	1831	1865	1899	1949	1970	2067	2174	2191	2300	
		2316	2369	2389	2475	2514	2555	2611	2633	2651	2694	2717	2746	2764	
		2812	2874	2959	2987	3015	3109	3137	3237	3287	3348	3375	3430	3457	
		3577	3617												
MSRJM	005761	1034	1574#												
MSTJM	005726	1030	1574#												
MTCN	005600	1024	1574#												
MTDATA=	040000	756#													
MTITLE	001000	661#	971												
MTOTAL	005647	1012	1574#												
MTSTN	005377	1480	1574#	1662											
MTSTPC	005300	1574#													
MVEC	005512	995	1574#												
MVECX	005347	1152	1574#												
NEXT	001216	681#	1208	1525	1761*	1793*	1827*	1861*	1895*	1969*	2066*	2190*	2315*	2388*	
		2513*	2610*	2650*	2693*	2763*	2811*	2867*	2978*	3089*	3236*	3277*	3340*	3418*	
		3576*													
NO.ATR	006770	1700#	1766	1800	1834	1868									
NO.BTR	006774	1704#	1767	1801	1835	1904	2049	2154	2280	2320	2418	2594	2871	3070	
		3217	3389	3486											
OPCLRJ	001323	788#	1022	1037	1081*										
OVRUN=	040000	745#													
PARAM =	104405	827#	988	996	1004	1013	1663								
PARAM1	003576	1285#	1302												
PARBIT=	000000	623#													
PARCSR	001410	852#	1631*	1980*	2325*	2397*	2536*	2613*	2653*	2696*	2719*	2766*	2814*	2876*	
		2989*	3111*	3241*	3290*	3350*	3432*	3587*							
PARERR	003652	1288	1290	1292	1301#	1308	1310	1312							
PASCNT	001230	686#	946*	1147*	1148	1181									
PERFOR=	000000	623#													
PKCLK =	104412	837#	3244	3246	3255	3592	3594	3595							
POPRO =	012600	601#	1519												
POP1SP=	005726	599#													
POP2SP=	022626	603#	1210												
PRIPTY	001320	786#	1010*	1074*	1794	1828	1862								
PRISEC=	010000	753#	2325	2536											
PS =	177776	592#	942*	1124*	1762*	1770*	1796*	1804*	1830*	1838*	1864*	1872*	1898*	1906*	
		1947*	1974*	2000*	2082*	2107*	2165*	2208*	2233*	2291*	2324*	2328*	2396*	2400*	
		2516*	2543*	2868*	2890*	2979*	3001*	3090*	3123*	3278*	3341*	3361*	3419*	3445*	
		600#	1504												
PUSHRO=	010046	598#													
PUSH1S=	005746	602#													
PUSH2S=	024646	1553	1555#												
PWRUP	005060	804#	948*	1160*	1199										
QV.FLG	001343	747#	2365												
RABORT=	002000	776#													
RCRCIN=	040000	775#													
RCRC7T=	100000	738#	1907	1981	2088	2214	2326	2398	2420	2454	2537	2698	2720	2768	
RCVEN =	000020	2816	2878	2991	3113	3240	3289	3352	3434	3589					
		731#	2921	2955	3024	3146	3309								
		748#	2027												
REACT=	004000	1496	1499#												
REOM =	001000														
RESREG	004614														

STACK =	001150	593#	943	1125	1526	1556	1948	2348	2414	2961	3017	3139	3377	3459
STD =	000010	739#												
STJMF	001334	793#	1032											
STPSYN=	000400	734#	2878	3434	3589									
SV05	003744	1337#												
SWR	001202	667#	959	961*	973	977	1091	1099	1104	1109	1128	1190	1197	1218
		1231	1448	1453	1502	1521	1523	1658	3621					
SWREG	000176	655#	961											
SW00 =	000001	574#	977	1091										
SW01 =	000002	573#	1658											
SW02 =	000004	572#												
SW03 =	000010	570#	1099											
SW04 =	000020	569#												
SW05 =	000040	568#												
SW06 =	000100	567#												
SW07 =	000200	566#												
SW08 =	000400	565#	1521											
SW09 =	001000	564#	1218	3621										
SW10 =	002000	563#	1523											
SW11 =	004000	562#	1197											
SW12 =	010000	561#	1231	1448										
SW13 =	020000	560#	1453											
SW14 =	040000	559#												
SW15 =	100000	558#												
SYSTST=	004000	771#	1510	1769	1803	1837	1871	1975	2083	2327	2399	2538	2614	2654
		2697	2721	2767	2815	2877	2990	3112	3291	3351	3433			
TABORT=	002000	780#	2338											
TCNFLG	001322	787#	1026	1027	1040	1080*	1896	2193						
TCRCIN=	010000	778#												
TCRC7T=	020000	777#												
TEMP	006054	1373	1581#											
TEMP1	001236	693#	1016	1044	1045	1046*	1084*	1085	1089*	1576	1711*	1734*	1972*	2036
		2037*	2038	2080*	2141	2142*	2143	2206*	2267	2268*	2269	2741*	2742	2792*
		2793	2825*	2827*	2840*	3174*	3178*	3179	3280*	3284*	3294	3315	3322*	3424*
		3466*	3469	3470*	3471									
TEMP2	001240	694#	1007	1010	1051*	1054*	1058*	1085*	1578	1712*	1716*	1973*	2019	2022*
		2023	2081*	2126	2129*	2130	2207*	2252	2255*	2256	2826*	2846*	2847	3177*
		3178	3187	3281*	3285*	3290	3323*	3425*	3478	3479*	3480			
TEMP3	001242	695#	1713*	1715	1725*	1727	1732*	1733*	1736	2161*	2162	2170	2287*	2288
		2296	3309	3426*	3492*	3494	3584*	3597*	3610					
TEMP4	001244	696#	3308*											

TRP.PC	006222	1587#																	
TSDM =	000400	782#	1921	1986	2093	2219	2331	2403	2497	2541	2616	2656	2659	2700					
		2723	2726	2770	2777	2780	2818	2821	2824	2839	2880	2883	2886	2889					
		2993	3000	3115	3122	3243	3284	3354	3357	3360	3436	3439	3442	3591					
TSTNO	001226	685#	953*	1534	1666	1672	1674	1760*	1792*	1826*	1860*	1894*	1968*	2065*					
		2189*	2314*	2387*	2512*	2609*	2649*	2692*	2762*	2810*	2866*	2977*	3088*	3235*					
		3276*	3339*	3417*	3575*														
		1669	1686	1760#															
TST1	007160	2066	2189#																
TST10	011432	2190	2314#																
TST11	012240	2315	2387#																
TST12	012554	2388	2512#																
TST13	013344	2513	2609#																
TST14	014004	2610	2649#																
TST15	014156	2650	2692#																
TST16	014352	2693	2762#																
TST17	014720	1761	1792#																
TST2	007264	2763	2810#																
TST20	015142	2811	2866#																
TST21	015406	2867	2977#																
TST22	016126	2978	3088#																
TST23	016610	3089	3235#																
TST24	017432	3236	3276#																
TST25	017630	3277	3339#																
TST26	020110	3340	3417#																
TST27	020450	1793	1826#																
TST3	007400	3418	3575#	3642															
TST30	021454	3576																	
TST31 =	***** U	1827	1860#																
TST4	007514	1861	1894#																
TST5	007632	1895	1968#																
TST6	010172	1969	2065#																
TST7	010632	1131*	1132*	1134*	1135*	1191#													
TTST	003174	623#																	
TWOSYN=	000000	761#	2629	2675															
TXACT =	001000	853#	1510*	1538*	1541*	1635*	1636*	1637	1763*	1769*	1774*	1797*	1803*	1808*					
TXCSR	001412	1831*	1837*	1842*	1865*	1871*	1877*	1899*	1901*	1909*	1918	1930	1936	1949*					
		1970*	1975*	1983	1985*	1995	1999*	2048*	2067*	2083*	2090	2092*	2102	2106*					
		2153*	2174*	2191*	2209*	2216	2218*	2228	2232*	2279*	2300*	2316*	2327*	2329					
		2332	2369*	2389*	2399*	2401	2404*	2421	2475*	2480	2490	2498*	2514*	2538*					
		2539	2542*	2555*	2593*	2611*	2614*	2615*	2625	2633*	2651*	2654*	2655*	2657					
		2660	2663*	2672	2675	2694*	2697*	2699*	2701	2717*	2721*	2722*	2724	2727					
		2746*	2764*	2767*	2769*	2775	2778	2812*	2815*	2817*	2819	2822	2836	2874*					
		2877*	2879*	2881	2884	2887	2892	2895	2898	2901	2904*	2959*	2987*	2990*					
		2992*	2998	3003*	3015*	3066	3069*	3109*	3112*	3114*	3120	3125*	3137*	3213					
		3216*	3237*	3239*	3242*	3254*	3287*	3291*	3292	3348*	3351*	3353*	3355	3358					
		3363*	3375*	3388*	3430*	3433*	3435*	3437	3440	3444*	3457*	3485*	3577*	3588*					
		3590*	3617*																
		854#	1639*	1640*	1641	1911	1913	1921*	1923	1925	1935*	1938*	1940	1942					
TXDBUF	001414	1986*	1988	1990	1998*	2003	2005	2036*	2042*	2045*	2093*	2095	2097	2105*					
		2110	2112	2141*	2147*	2150*	2219*	2221	2223	2231*	2236	2238	2267*	2273*					
		2276*	2331*	2334*	2338*	2340	2342	2403*	2406	2408	2423*	2425	2427	2483*					
		2493*	2497*	2541*	2546	2548	2584*	2590*	2616*	2618	2620	2656*	2659*	2662*					
		2665	2667	2700*	2703*	2705	2707	2723*	2726*	2729*	2731	2733	2770*	2771					
		2773	2777*	2780*	2782	2784	2818*	2821*	2824*	2829	2831	2839*	2880*	2883*					
		2886*	2889*	2894*	2897*	2900*	2903*	2906	2908	2993*	2994	2996	3000*	3006					

CZDPD-DO
CZDPDD.P11

MACY11 30A(1052)
25-JUL-79 15:01

25-JUL-79 15:19 PAGE 84

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0082

```
TXDLAT= 100000
TXDONE= 000200
TXINTE= 000100
```

TYPDAT 004602
TYPE = 104402

```

TYPMSG      004502
USER        = 000000
WRDCNT      004234
WRKO.F      004570
XBX         004374
XCSR        003130
XERR        003152
XHEAD       006014
XPASS       003144
XPOLY       007152
XSTATQ      006042
XTSTN       005000
XVEC        003136
$E          = 000032

```

SENDER 003104
SN = 000030

SRAYO = 177777

SY = 000014

$$. = 023540$$

```
.BEGIN 002660
.CNVRT 004034
.CONVR 004030
.EOP 002764
.ERRTA 023274
.HLT 004350
.INSTE 003516
```

3008	3063*	3068*	3115*	3116	3118	3122*	3128	3130	3210*	3215*	3243*	3245*
3294*	3298	3300	3354*	3357*	3360*	3366	3368	3384*	3387*	3436*	3439*	3442*
3448	3450	3465*	3469*	3475*	3478*	3484*	3591*	3593*	3609*	3612*		
755#												
763#	2626											
764#	1769	1803	1837	1871	1999	2048	2106	2153	2232	2279	2404	2498
2542	2593	3003	3069	3125	3216	3363	3388	3444	3485			
1474	1492	1495#										
821#	971	1083	1101	1106	1130	1137	1149	1150	1152	1154	1156	1235
1248	1267	1359	1396	1475	1476	1479	1480	1482	1484	1488	1493	1558
1600	1660	1677	1678	1683	3631	3632						
1472	1475#											
768#												
1367*	1397*	1405#										
1487	1490#											
1449	1451	1453#										
1151	1173#											
1157	1182#											
1083	1574#											
1155	1179#											
1721	1732	1741#	2070*	2196*	2520*	3092*	3579*					
1088	1574#											
1481	1532#											
1153	1176#											
1#	1761	1762#	1793	1794#	1827	1828#	1861	1862#	1895	1896#	1969	1970#
2066	2067#	2190	2191#	2315	2316#	2388	2389#	2513	2514#	2610	2611#	2650
2651#	2693	2694#	2763	2764#	2811	2812#	2867	2868#	2978	2979#	3089	3090#
3236	3237#	3277	3278#	3340	3341#	3418	3419#	3576	3577#			
649	969	1165#	1500									
1#	1748	1754	1762#	1780	1786	1794#	1814	1820	1828#	1848	1854	1862#
1882	1888	1896#	1957	1962	1970#	2054	2059	2067#	2178	2183	2191#	2304
2308	2316#	2374	2381	2389#	2502	2506	2514#	2598	2603	2611#	2638	2643
2651#	2681	2686	2694#	2751	2756	2764#	2799	2804	2812#	2853	2860	2868#
2966	2971	2979#	3076	3082	3090#	3224	3229	3237#	3266	3270	3278#	3329
3333	3341#	3405	3411	3419#	3559	3569	3577#	3642#				
1#	1748#	1749	1752#	1780#	1781	1784#	1814#	1815	1818#	1848#	1849	1852#
1882#	1883	1886#	1957#	1958	1960#	2054#	2055	2057#	2178#	2179	2181#	2304#
2305	2307#	2374#	2375	2379#	2502#	2503	2505#	2598#	2599	2601#	2638#	2639
2641#	2681#	2682	2684#	2751#	2752	2754#	2799#	2800	2802#	2853#	2854	2859#
2966#	2967	2970#	3076#	3077	3081#	3224#	3225	3227#	3266#	3267	3269#	3329#
3330	3332#	3405#	3406	3409#	3559#	3560	3568#				</	

CZDPD-DO
CZDPDD.P11MACY11 30A(1052)
25-JUL-79 15:01

25-JUL-79 15:19 PAGE 85

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0083

.INSTR	003412	824	1244#		
.INST1	003432	1248#	1268		
.MSG	003434	1246*	1249#		
.PARAM	003536	828	1276#		
.PFAIL	005050	639	944	1553#	1557
.PKCLK	005006	838	1535#		
.RES05	003776	832	1347#		
.SAV05	003736	830	1333#		
.SCOPE	003160	818	1188#		
.SCOP1	003312	820	1218#		
.SETFL	004242	840	1416#	1428	
.START	001562	657	942#	954	
.TRPSR	004316	643	1436#		
.TRPTA	001344	816#	1441		
.TYPE	003336	822	1228#		

CZDPD-DO MACY11 30A(1052) 25-JUL-79 15:19 PAGE 88
 CZDPDD.P11 25-JUL-79 15:01 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0085

\$SMALL	1#	1546													
\$SYNC	541#	2724	2775	2778	2819	2822	2881	2884	2887	2998	3120	3355	3358	3437	3440
\$TRPDE	1#	817	819	821	823	825	827	829	831	833	835	837	839		
\$TSTN	1#	1754	1786	1820	1854	1888	1962	2059	2183	2308	2381	2506	2603	2643	2686
	2756	2804	2860	2971	3082	3229	3270	3333	3411	3569					
\$VARIA	1#	659													
\$WAIT	1#	1910	1922	1939	1987	2001	2094	2108	2220	2234	2339	2405	2424	2544	2617
	2664	2704	2730	2771	2781	2828	2905	2994	3005	3116	3127	3297	3364	3447	
\$XZ	1#	1748	1752	1780	1784	1814	1818	1848	1852	1882	1886	1957	1960	2054	2057
	2178	2181	2304	2307	2374	2379	2502	2505	2598	2601	2638	2641	2681	2684	2751
	2754	2799	2802	2853	2859	2966	2970	3076	3081	3224	3227	3266	3269	3329	3332
	3405	3409	3559	3568											

. ABS. 023540 000

ERRORS DETECTED: 0

CZDDPDD.BIN,CZDPDD.LST/CRF/SOL/NL:TOC=CZDPXX.MAC,CZDPDD.P11

RUN-TIME: 34 47 3 SECONDS

RUN-TIME RATIO: 1396/86=16.1

CORE USED: 25K (49 PAGES)