

DUP11

SDLC DECMD TST  
CZDPDC0

AH-8585C-MC  
COPYRIGHT © 75-78  
FICHE 1 OF 1

JUL 1978  
digital  
MADE IN USA

Table with 7 columns and 14 rows of data. The content is extremely faint and illegible.





IDENTIFICATION

PRODUCT CODE: AC-8584C-MC  
PRODUCT NAME: CZDPDC0 DUP11 SDLC DECMD TST  
DATE: APRIL 1978  
MAINTAINER: DIAGNOSTICS

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

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, 1978 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 PROCEDURE

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

'EXAMPLE'

'MAP OF DUP11 STATUS'

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

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

IT IS POSSIBLE FOR THE OPERATOR TO MANUALLY CHANGE (TOGGLE IN) THE INFORMATION IN THE MAP TO SUIT A SPECIFIC CONFIGURATION OF 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 CZDPDB 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 SEQUENTIALY. THEY CONTAIN THE CSR, VECTOR AND STATUS CONCERNING THE CONFIGURATION OF EACH DUP11.

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



EXAMPLE: (DUPACTV) /0000000000010001 MEANS  
THAT DUP11 NO. 00,04 WILL BE TESTED.

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.



```
533      ;*CZDPDC0 /<377>/DUP11 SDLC DECMD TST
534      ;*COPYRIGHT(C) 1975,1978, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
535      ;-----
536
537      ;STARTING PROCEDURE
538      ;LOAD PROGRAM
539      ;LOAD ADDRESS 000200
540      ;PRESS START
541      ;PROGRAM WILL TYPE "CZDPDC0 /<377>/DUP11 SDLC DECMD TST "
542      ;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
543      ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
544      ;AND THEN RESUME TESTING
545
546
547      ;SWITCH REGISTER OPTIONS
548      ;-----
549
550      100000      SW15=100000      :=1,HALT ON ERROR
551      040000      SW14=40000      :=1,LOOP ON CURRENT TEST
552      020000      SW13=20000      :=1,INHIBIT ERROR TYPEOUT
553      010000      SW12=10000      :=1,DELETE TYPEOUT/BELL ON ERROR.
554      004000      SW11=4000       :=1,INHIBIT ITERATIONS
555      002000      SW10=2000       :=1,ESCAPE TO NEXT TEST ON ERROR
556      001000      SW09=1000       :=1,LOOP WITH CURRENT DATA
557      000400      SW08=400        :=1,LOOP ON ERROR
558      000200      SW07=200
559      000100      SW06=100
560      000040      SW05=40
561      000020      SW04=20
562      000010      SW03=10
563
564      000004      SW02=4
565      000002      SW01=2
566      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
```



GENERAL DEFINITIONS AND EQUIVALENCIES

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

: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  
POPR0=12600 :RESTORE R0 FROM STACK  
PUSH2SP=24646 :DECREMENT STACK TWICE  
POP2SP=22626 :INCREMENT STACK TWICE  
.EQUIV EMT,HLT :BASIC DEFINITION OF ERROR CALL

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

TRAPCATCHER FOR UNEXPECTED INTERRUPTS

```

617 ;:*****
618 ;-----
619 ; TRAPCATCHER FOR ILLEGAL INTERRUPTS
620 ; THE STANDARD "TRAP CATCHER" IS PLACED
621 ; BETWEEN ADDRESS 0 TO ADDRESS 776.
622 ; IT LOOKS LIKE "PC+2 HALT".
623 ;-----
624 ;:*****
625
626 000000 . =0
627 ; STANDARD INTERRUPT VECTORS
628 ;-----
629
630 000024 . =24
631 000024 005050 .PFAIL ; POWER FAIL HANDLER
632 000026 000340 340 ; SERVICE AT LEVEL 7
633 000030 004350 .HLT ; ERROR HANDLER
634 000032 000340 340 ; SERVICE AT LEVEL 7
635 000034 004316 .TRPSRV ; GENERAL HANDLER DISPATCH SERVICE
636 000036 000340 340 ; SERVICE AT LEVEL 7
637
638 000040 . =40
639 000042 000000 0 ; SAVE FOR ACT-11 OR DDP2
640 000044 000000 0 ; RETURN ADDRESS IF UNDER ACT-11 OR DDP2
641 000046 003104 0 ; SAVE FOR ACT-11 OR DDP2
642 000052 . =52 $ENDAD ; FOR USE WITH ACT-11 OR DDP2
643 000052 000000 0 ; ACT-11 PROGRAM CHARACTERISTICS
644
645 . =174
646 000174 000000 DISPREG:0 ; SOFTWARE DISPLAY REGISTER
647 000176 000000 SWREG: 0 ; SOFTWARE SWITCH REGISTER
648 000200 . =200
649 000200 000137 001562 JMP .START ; GO TO START OF PROGRAM
650
651
652 . =1000
653 001000 005377 055103 050104 MTITLE: .ASCIZ <377><12>/CZDPDC0 /<377>/DUP11 SDLC DECMD TST /<377>
654 (2)
655 001200 . =1200
656 ; SWR AND LIGHTS
657 ;-----
658 001200 177570 DISPLAY: 177570 ; 11/45 CONSOLE LIGHTS
659 001202 177570 SWR: 177570 ; INDIRECT POINTER TO SWITCH REGISTER
660
661 ; INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
662 ;-----
663
664 001204 177560 TKCSR: 177560 ; TELETYPE KEYBOARD CONTROL REGISTER
665 001206 177562 TKDBR: 177562 ; TELETYPE KEYBOARD DATA BUFFER
666 001210 177564 TPCSR: 177564 ; TELEPRINTER CONTROL REGISTER
667 001212 177566 TPDBR: 177566 ; TELEPRINTER DATA BUFFER
668
669 ; PROGRAM CONTROL PARAMETERS
670 ;-----
671

```



PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

672	001214	000000	RETURN: 0	:SCOPE ADDRESS FOR LOOP ON TEST
673	001216	000000	NEXT: 0	:ADDRESS OF NEXT TEST TO BE EXECUTED
674	001220	000000	LOCK: 0	:ADDRESS FOR LOCK ON CURRENT DATA
675	001222	000001	ICOUNT: 1	:NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
676	001224	000000	LPCNT: 0	:NUMBER OF ITERATIONS COMPLETED
677	001226	000000	TSTNO: 0	:NUMBER OF TEST IN PROGRESS
678	001230	000000	PASCNT: 0	:NUMBER OF PASSES COMPLETED
679	001232	000000	ERRCNT: 0	:TOTAL NUMBER OF ERRORS
680	001234	000000	LSTERR: 0	:PC OF LAST ERROR CALL

:PROGRAM VARIABLES

681				
682				
683				
684				
685	001236	000000	TEMP1: 0	:TEMPORARY STORAGE
686	001240	000000	TEMP2: 0	:TEMPORARY STORAGE
687	001242	000000	TEMP3: 0	:TEMPORARY STORAGE
688	001244	000000	TEMP4: 0	:TEMPORARY STORAGE
689	001246	000000	TEMP5: 0	:TEMPORARY STORAGE
690	001250	000000	SAVR0: 0	:R0 STORAGE
691	001252	000000	SAVR1: 0	:R1 STORAGE
692	001254	000000	SAVR2: 0	:R2 STORAGE
693	001256	000000	SAVR3: 0	:R3 STORAGE
694	001260	000000	SAVR4: 0	:R4 STORAGE
695	001262	000000	SAVR5: 0	:R5 STORAGE
696	001264	000000	SAVSP: 0	:STACK POINTER STORAGE
697	001266	000000	SAVPC: 0	:PROGRAM COUNTER STORAGE
698				
699	001270	000000	SAVR0A: 0	:R0 STORAGE
700	001272	000000	SAVR1A: 0	:R1 STORAGE
701	001274	000000	SAVR2A: 0	:R2 STORAGE
702	001276	000000	SAVR3A: 0	:R3 STORAGE
703	001300	000000	SAVR4A: 0	:R4 STORAGE
704	001302	000000	SAVR5A: 0	:R5 STORAGE
705	001304	000000	SAVSPA: 0	:STACK POINTER STORAGE
706	001306	000000	SAVPCA: 0	:PROGRAM COUNTER STORAGE
707				
708	001310	000001	DUPACTV: .BLKB 1	:DUP11'S SELECTED ACTIVE.
709	001311	000001	DUPNUM: .BLKB 1	:OCTAL NUMBER OF DUP11'S.
710	001312	000001	SAVACT: .BLKB 1	:ORIGINAL ACTV. DEVICES.
711	001313	000001	SAVNUM: .BLKB 1	:WORKABLE NUMBER.
712	001314	000001	RUN: .BLKB 1	:POINTER ONE PAST RUNNING DEVICE.
713		001316	.EVEN	
714	001316	001500	CREAM: DUP.MAP	:TABLE POINTER.

```

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

```



PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

771	004000	TIMER=BIT11	:MAINTENANCE TIMER
772	002000	TABORT=BIT10	:TRANSMIT ABORT
773	001000	TEOM=BIT9	:TRANSMIT END OF MESSAGE
774	000400	TSOM=BIT8	:TRANSMIT START OF MESSAGE

775  
776 :MISC. PROGRAM DEFINITIONS

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

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

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



PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```
836                                     ;DUP11 VECTOR AND REGISTER INDIRECT POINTERS
837
838 001374 000000  DUPRVC: 0           ;POINTER TO DUP11 RECEIVER INTERRUPT VECTOR
839 001376 000000  DUPRPS: 0           ;POINTER TO DUP11 RECEIVER INTERRUPT SERVICE PS
840 001400 000000  DUPTVC: 0           ;POINTER TO DUP11 TRANSMITTER INTERRUPT VECTOR
841 001402 000000  DUPTPS: 0          ;POINTER TO DUP11 TRANSMITTER INTERRUPT SERVICE PS
842 001404 000000  RXCSR: 0            ;POINTER TO DUP11 RECEIVER STATUS REGISTER
843 001406 000000  RXDBUF: 0           ;POINTER TO DUP11 RECEIVER DATA BUFFER
844 001410 000000  PARCSR: 0           ;POINTER TO DUP11 PARAMETER STATUS REGISTER
845 001412 000000  TXCSR: 0            ;POINTER TO DUP11 TRANSMITTER STATUS REGISTER
846 001414 000000  TXDBUF: 0           ;POINTER TO DUP11 TRANSMITTER DATA BUFFER
847 001416 000000  DUPSEC: 0           ;POINTER TO DUP11 SECONDARY REGISTER SELECT REGISTER
848 001420 000000  HUPPSR: 0           ;POINTER TO PARAMETER STATUS HIGH BYTE
849 001422 000000  HUPRBF: 0           ;POINTER TO RECEIVER BUFFER HIGH BYTE
850 001424 000000  HUPRCR: 0           ;POINTER TO RECEIVER CONTROL REG HIGH BYTE
851 001426 000000  HUPTBF: 0           ;POINTER TO TRANSMITTER BUFFER HIGH BYTE
852 001430 000000  HUPTCR: 0           ;POINTER TO TRANSMITTER CONTROL REG HIGH BYTE
853
854
855                                     ;DUP11 CONTROL INDICATORS FOR CURRENT DUP11 UNDER TEST
856                                     ;-----
857
858 001432 000      MASK.A: .BYTE 000      ;LAST CHAR TO TEST AND PARITY MASK
859
860 001433 010      CLK.A: .BYTE 8.        ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR
861
862 001434 000000   L00.00: 000000          ;PARAMETERS
863
```

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

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



PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
I	C	O	N	T	R	O	L		R	E	G	I	S	T	E
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
I	A	B	C	D	E	F	G	H	I	*	*	S	Y	N	C
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I

DEFINITIONS

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

PROGRAM INITIALIZATION AND START UP.

```

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



PROGRAM INITIALIZATION AND START UP.

982	002052	175500			175500	:HIGH LIMIT
983	002054	001500			DUPCRO	:STORE AT THIS LOCATION
984	002056	001			.BYTE 1	:MASK
985	002057	001			.BYTE 1	:HOW MANY TIMES + 2
986	002060	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
987	002062	005512			MVEC	:MESSAGE
988	002064	104405			PARAM	:CONVERT STRING
989	002066	000300			300	:LOW LIMIT
990	002070	000770			770	:HIGH LIMIT
991	002072	001502			DUPTR0	:STORE AT THIS LOCATION
992	002074	001			.BYTE 1	:MASK
993	002075	001			.BYTE 1	:HOW MANY TIMES + 2
994	002076	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
995	002100	005702			MPAR	:MESSAGE
996	002102	104405			PARAM	:CONVERT STRING
997	002104	000004			4	:LOW LIMIT
998	002106	000007			7	:HIGH LIMIT
999	002110	001240			TEMP2	:STORE AT THIS LOCATION
1000	002112	000			.BYTE 0	:MASK
1001	002113	001			.BYTE 1	:HOW MANY TIMES + 2
1002	002114	013737	001240	001320	MOV	TEMP2,PRTY ;SAVE PRIORITY
1003	002122	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1004	002124	005647			MTOTAL	:MESSAGE
1005	002126	104405			PARAM	:CONVERT STRING
1006	002130	000001			1	:LOW LIMIT
1007	002132	000010			8.	:HIGH LIMIT
1008	002134	001236			TEMP1	:STORE AT THIS LOCATION
1009	002136	000			.BYTE 0	:MASK
1010	002137	001			.BYTE 1	:HOW MANY TIMES + 2
1011	002140	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1012	002142	005525			MJMPR	:MESSAGE
1013	002144	104413			SETFLG	:SET FLAG BASED UPON INPUT STRING
1014	002146	001323			OPCLRJ	:THIS FLAG
1015	002150	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1016	002152	005600			MTCN	:MESSAGE
1017	002154	104413			SETFLG	:SET FLAG BASED UPON INPUT STRING
1018	002156	001322			TCNFLG	:THIS FLAG
1019	002160	105737	001322		TSTB	TCNFLG
1020	002164	001410			BEQ	71\$
1021	002166	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1022	002170	005726			MSTJM	:MESSAGE
1023	002172	104413			SETFLG	:SET FLAG BASED UPON INPUT STRING
1024	002174	001334			STJMFL	:THIS FLAG
1025	002176	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1026	002200	005761			MSRJM	:MESSAGE
1027	002202	104413			SETFLG	:SET FLAG BASED UPON INPUT STRING
1028	002204	001336			SRJMFL	:THIS FLAG
1029	002206	105737	001323		71\$: TSTB	OPCLRJ
1030	002212	001403			BEQ	69\$
1031	002214	052737	100000	001504	BIS	#BIT15,DUPO.A
1032	002222	105737	001322		69\$: TSTB	TCNFLG
1033	002226	001403			BEQ	70\$
1034	002230	052737	040000	001504	BIS	#BIT14,DUPO.A
1035	002236	112737	000001	001312	70\$: MOVB	#1,SAVACT
1036	002244	113737	001236	001311	MOVB	TEMP1,DUPNUM
1037	002252	113737	001236	001313	MOVB	TEMP1,SAVNUM

PROGRAM INITIALIZATION AND START UP.

```

1038 002260 005337 001236      65$: DEC    TEMP1
1039 002264 001404              BEQ    64$
1040 002266 000261              SEC
1041 002270 106137 001312      ROLB  SAVACT
1042 002274 000771              BR    65$
1043 002276 113737 001312 001240 64$: MOVB  SAVACT,TEMP2 ;# OF TIMES
1044 002304 113737 001312 001310 MOVB  SAVACT,DUPACTV
1045 002312 000241              CLC
1046 002314 106037 001240      RORB  TEMP2
1047 002320 012700 001500      MOV   #DUPCRO,R0
1048 002324 012701 001506      MOV   #DUPCR1,R1
1049 002330 000241              67$: CLC
1050 002332 106037 001240      RORB  TEMP2
1051 002336 103051              BCC   66$
1052 002340 012011              MOV   (R0)+,(R1)
1053 002342 062721 000010      ADD   #10,(R1)+ ;CSR
1054 002346 012011              MOV   (R0)+,(R1)
1055 002350 062721 000010      ADD   #10,(R1)+ ;VECTOR
1056 002354 012021              MOV   (R0)+,(R1)+ ;PARAMETERS
1057 002356 000764              BR    67$
1058 002360 012700 001500      21$: MOV   #DUP.MAP,R0 ;SETUP TO CLEAR MAP
1059 002364 005020      20$: CLR   (R0)+ ;CLEAR
1060 002366 020027 001560      CMP   R0,#DUP.END ;CHECK FOR FINISH
1061 002372 001374              BNE   20$ ;BR IF MORE TO GO
1062 002374 012700 001500      MOV   #DUP.MAP,R0 ;SETUP TO DEFAULT
1063 002400 012710 160050      MOV   #160050,(R0) ;LOAD CSR
1064 002404 012760 000770 000002 MOV   #770,2(R0) ;LOAD VECTOR
1065 002412 012760 140026 000004 MOV   #140026,4(R0) ;LOAD PARAMETERS AND SYNC
1066 002420 112737 000005 001320 MOVB  #5,PRIRTY ;LOAD PRIORITY
1067 002426 012700 000001      MOV   #1,R0 ;SAVE CORE THIS WAY
1068 002432 110037 001310      MOVB  R0,DUPACTV ;PRESET PROGRAM CONTROLS
1069 002436 110037 001311      MOVB  R0,DUPNUM ;DITTO
1070 002442 110037 001312      MOVB  R0,SAVACT ;DITTO
1071 002446 110037 001313      MOVB  R0,SAVNUM ;DITTO
1072 002452 110037 001322      MOVB  R0,TCNFLG ;DITTO
1073 002456 110037 001323      MOVB  R0,OPCLRJ ;DITTO
1074 002462
1075 002462 104402 006014      66$: TYPE  ,XHEAD ;TYPE HEADER
1076 002466 012737 001500 001236 16$: MOV   #DUP.MAP,TEMP1 ;SET POINTER
1077 002474 017737 176536 001240 5$: MOV   @TEMP1,TEMP2 ;SET DATA
1078 002502 001406              BEQ   1$ ;ALL DONE WITH DATA
1079 002504 104410              CONVRT
1080 002506 006042              XSTATQ
1081 002510 062737 000002 001236 ADD   #2,TEMP1 ;UPDATE POINTER
1082 002516 000766              BR    5$
1083 002520 032777 000001 176454 1$: BIT   #SW00,@SWR
1084 002526 001405              BEQ   7$
1085 002530 005737 001332      TST   FLAG
1086 002534 001002              BNE   7$
1087 002536 000137 002000      JMP   10$
1088 002542 005037 001332      7$: CLR   FLAG
1089 002546 005737 000042      TST   @#42 ;IS PROGRAM RUNNING UNDER MONITOR
1090 002552 001030              BNE   3$ ;BR IF YES
1091 002554 032777 000010 176420 BIT   #SW03,@SWR ;SELECT SPECIFIC DEVICES??
1092 002562 001424              BEQ   3$ ;BR IF NO.
1093 002564 104402 005413      TYPE  ,MNEW ;TYPE THE MESSAGE.

```



PROGRAM INITIALIZATION AND START UP.

```

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

```

1131                                     ;END OF PASS
1132                                     ;TYPE NAME OF TEST
1133                                     ;UPDATE PASS COUNT
1134                                     ;CHECK FOR EXIT TO ACT-11
1135                                     ;RESTART TEST
1136
1137 002764 005037 001234      .EOP: CLR      LSTERR      ;CLEAR LAST ERROR PC
1138 002770 105037 001341      CLRB     ERRFLG     ;CLEAR ERROR FLAG
1139 002774 005237 001230      INC      PASCNT     ;UPDATE PASS COUNT
1140 003000 013777 001230 176172  MOV      PASCNT,@DISPLAY ;DISPLAY PASS COUNT
1141 003006 104402 005157      TYPE     ,MEPASS    ;TYPE END PASS
1142 003012 104402 005341      TYPE     ,MCSRX     ;TYPE CSR
1143 003016 104411 003130      CNVRT    ,XCSR      ;SHOW IT
1144 003022 104402 005347      TYPE     ,MVECX     ;TYPE VECTOR
1145 003026 104411 003136      CNVRT    ,XVEC      ;SHOW IT
1146 003032 104402 005355      TYPE     ,MPASSX    ;TYPE PASSES
1147 003036 104411 003144      CNVRT    ,XPASS     ;SHOW IT
1148 003042 104402 005366      TYPE     ,MERRX     ;TYPE ERRORS
1149 003046 104411 003152      CNVRT    ,XERR      ;SHOW IT
1150 003052 105337 001313      DECB    SAVNUM     ;ARE ALL DEVICES TESTED?
1151 003056 001017              BNE      RESTRT     ;BR IF NO.
1152 003060 112737 000377 001343  MOVB    #377,QV.FLG ;SET THE QUICK VERIFY FLAG.
1153 003066 113737 001311 001313  MOVB    DUPNUM,SAVNUM ;RESTORE THE COUNT
1154 003074 013701 000042      MOV      @#42,R1    ;CHECK FOR ACT-11 OR DDP
1155 003100 001406              BEQ      RESTRT     ;IF NOT, CONTINUE TESTING
1156 003102 000005              RESET           ;STOP THE SHOW--CLEAR THE WORLD
1157 003104
1158 003104 004711      $ENDAD: JSR      PC,(R1)
1159 003106 000240      NOP
1160 003110 000240      NOP
1161 003112 000240      NOP
1162 003114 000240      NOP
1163 003116 012737 006224 001214  RESTRT: MOV      #CYCLE,RETURN
1164 003124 000137 006224      JMP      CYCLE
1165 003130 000001      XCSR:   1
1166 003132 006 002      .BYTE   6,2
1167 003134 001404      RXCSR
1168 003136 000001      XVEC:   1
1169 003140 003 002      .BYTE   3,2
1170 003142 001374      DUPRVC
1171 003144 000001      XPASS:  1
1172 003146 006 002      .BYTE   6,2
1173 003150 001230      PASCNT
1174 003152 000001      XERR:   1
1175 003154 006 002      .BYTE   6,2
1176 003156 001232      ERRCNT
1177
1178                                     ;SCOPE LOOP AND INTERATION HANDLER
1179
1180 003160 005037 001234      .SCOPE: CLR      LSTERR      ;CLEAR LAST ERROR PC
1181 003164 010016      MOV      RO,(SP)    ;SAVE RO ON STACK
1182 003166 032777 040000 176006  BIT      #BIT14,@SWR ;LOOP ON TEST?
1183 003174 001407      TTST:  BEQ      1$      ;BR IF NO (IF LOCK SW01 = 1;THIS LOCATION = 240)
1184 003176 000437      BR      3$          ;GO TO 3$ (DITTO)
1185 003200 105777 176000      TSTB    @TKCSR     ;KYBD DONE?
1186 003204 100034      BPL     3$          ;BR IF NO (LOCK: HIT A KEY ON TTY TO GO TO NEXT TEST)

```



```

1187 003206 017700 175774          MOV    @TKDBR,R0      ;CLR DONE BIT
1188 003212 000415                   BR     2$            ;CONTINUE
1189 003214 032777 004000 175760 1$: BIT    #SW11,@SWR    ;DELETE ITERATION (QUICK PASS)?
1190 003222 001011                   BNE   2$            ;BR IF YES
1191 003224 105737 001343          TSTB  QV.FLG        ;HAS FIRST PASS BEEN COMPLETED?
1192 003230 001406                   BEQ   2$            ;BR IF QUICK VERIFY
1193 003232 005237 001224          INC   LPCNT         ;UPDATE ITERATION COUNTER
1194 003236 023737 001224 001222  CMP   LPCNT,ICOUNT  ;ALL ITERATIONS DONE?
1195 003244 001014                   BNE   3$            ;BR IF NOT YET
1196 003246 105037 001341          2$:  CLRB  ERRFLG    ;PREPARE FOR NEW TEST
1197 003252 005037 001224          CLR   LPCNT        ;START ICOUNT AT ZERO
1198 003256 005037 001220          CLR   LOCK
1199 003262 012737 000050 001222  MOV   #50,ICOUNT   ;RESET ITERATIONS
1200 003270 013737 001216 001214  MOV   NEXT,RETURN  ;GET NEXT TEST
1201 003276 011600          3$:  MOV   (SP),R0   ;POP R0 OFF STACK
1202 003300 022626          POP2SP             ;FAKE AN RTI
1203 003302 000177 175706          JMP   @RETURN      ;GO DO THE TEST
1204 003306 001407          BRW: 1407
1205 003310 000437          BRX: 437
1206
1207
1208
1209
1210 003312 032777 001000 175662 .SCOP1: BIT    #SW09,@SWR    ;IS SW09=1(SET)?
1211 003320 001405                   BEQ   1$            ;BR IF NOT SET.
1212 003322 005737 001220          TST   LOCK
1213 003326 001402                   BEQ   1$
1214 003330 013716 001220          MOV   LOCK,(SP)   ;GOTO THE ADDRESS IN LOCK.
1215 003334 000002          1$:  RTI           ;GO BACK.
1216
1217
1218
1219
1220 003336 010546          .TYPE: MOV   R5,-(SP)    ;SAVE R5 ON THE STACK.
1221 003340 017605 000002          MOV   @2(SP),R5   ;GET ADDRESS OF MESSAGE.
1222 003344 062766 000002 000002  ADD   #2,2(SP)    ;POP OVER ADDRESS.
1223 003352 032777 010000 175622 1$:  BIT    #SW12,@SWR    ;INHIBIT ALL PRINT OUT??
1224 003360 001012                   BNE   3$            ;BR IF NO PRINT OUT WANTED (SW12=1)
1225 003362 105715          TSTB  (R5)        ;IS NUMBER MINUS? (MSB=1(BIT7))
1226 003364 100002                   BPL   2$            ;BR IF NUMBER IS PLUS
1227 003366 104402 005136          TYPE  ,MCRLF      ;TYPE A CR/LF!
1228 003372 105777 175612          2$:  TSTB  @TPCSR    ;TTY READY?
1229 003376 100375                   BPL   2$            ;BR IF NO.
1230 003400 112577 175606          MOVB  (R5)+,@TPDBR ;PRINT CURRENT CHAR.
1231 003404 001362                   BNE   1$            ;IF NOT ZERO KEEP PRINTING!
1232 003406 012605          3$:  MOV   (SP)+,R5  ;END OF OUTPUT. RESTORE R5
1233 003410 000002          RTI           ;GO HOME
1234
1235
1236 003412 010346          .INSTR: MOV   R3,-(SP)    ;SAVE R3 ON STACK
1237 003414 010446          MOV   R4,-(SP)    ;SAVE R4 ON STACK
1238 003416 017637 000004 003434  MOV   @4(SP),.MSG
1239 003424 062766 000002 000004  ADD   #2,4(SP)
1240 003432 104402          .INST1: TYPE
1241 003434 000000          .MSG: 0
1242 003436 012704 006160          MOV   #INBUF,R4
  
```

END OF PASS ROUTINE

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

;CONVERT ASCII STRING TO OCTAL  
 ;-----

;TEST TO SEE IF NUMBER IS WITHIN LIMITS  
 ;-----



END OF PASS ROUTINE

```

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

END OF PASS ROUTINE

1355	004042	010446			MOV	R4,-(SP)
1356	004044	010546			MOV	R5,-(SP)
1357	004046	017601	000012		MOV	@12(SP),R1
1358	004052	062766	000002	000012	ADD	#2,12(SP)
1359	004060	012137	004234		MOV	(R1)+,WRDCNT
1360	004064	112137	004236	1\$:	MOVB	(R1)+,CHRCNT
1361	004070	112137	004237		MOVB	(R1)+,SPACNT
1362	004074	013137	004240		MOV	@(R1)+,BINWRD
1363	004100	013704	004240	2\$:	MOV	BINWRD,R4
1364	004104	113705	004236		MOVB	CHRCNT,R5
1365	004110	012700	006054		MOV	#TEMP,R0
1366	004114	010403		3\$:	MOV	R4,R3
1367	004116	042703	177770		BIC	#177770,R3
1368	004122	062703	000060		ADD	#060,R3
1369	004126	110320			MOVB	R3,(R0)+
1370	004130	000241			CLC	
1371	004132	006004			ROR	R4
1372	004134	000241			CLC	
1373	004136	006004			ROR	R4
1374	004140	000241			CLC	
1375	004142	006004			ROR	R4
1376	004144	005305			DEC	R5
1377	004146	001362			BNE	3\$
1378	004150	012703	006116		MOV	#MDATA,R3
1379	004154	114023		4\$:	MOVB	-(R0),(R3)+
1380	004156	105337	004236		DECB	CHRCNT
1381	004162	001374			BNE	4\$
1382	004164	105737	004237		TSTB	SPACNT
1383	004170	001405			BEQ	6\$
1384	004172	112723	000040	5\$:	MOVB	#040,(R3)+
1385	004176	105337	004237		DECB	SPACNT
1386	004202	001373			BNE	5\$
1387	004204	105013		6\$:	CLRB	(R3)
1388	004206	104402	006116		TYPE	,MDATA
1389	004212	005337	004234		DEC	WRDCNT
1390	004216	001322			BNE	1\$
1391	004220	012605			MOV	(SP)+,R5
1392	004222	012604			MOV	(SP)+,R4
1393	004224	012603			MOV	(SP)+,R3
1394	004226	012601			MOV	(SP)+,R1
1395	004230	012600			MOV	(SP)+,R0
1396	004232	000002			RTI	
1397	004234	000000				
1398	004236	000000				
1399		004237				
1400	004240	000000				
1401						
1402						
1403						
1404						
1405						
1406						
1407						
1408	004242	017605	000000		.SETFLG:MOV	@(SP),R5
1409	004246	042737	000040	006160	BIC	#40,INBUF
1410	004254	122737	000116	006160	CMPB	#'N',INBUF ;IS IT 'N' ?

;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



```

1411 004262 001002          BNE      1$
1412 004264 105015          CLRB    (R5)      ;000
1413 004266 000406          BR      2$
1414 004270 122737 000131 006160 1$:  CMPB    #'Y',INBUF      ;IS IT 'Y' ?
1415 004276 001005          BNE     3$
1416 004300 112715 177777          MOVB    #-1,(R5)      ;377
1417 004304 062716 000002          2$:  ADD     #2,(SP)
1418 004310 000002          RTI
1419 004312 104404          3$:  INSTER  ;RETRY
1420 004314 000752          BR      .SETFLG
1421
1422
1423          ;TRAP DISPATCH SERVICE
1424          ;ARGUMENT OF TRAP IS EXTRACTED
1425          ;AND USED AS OFFSET TO OBTAIN POINTER
1426          ;TO SELECTED SUBROUTINE
1427
1428 004316 011646          .TRPSR: MOV    (SP),-(SP)      ;GET PC OF RETURN
1429 004320 162716 000002          SUB     #2,(SP)      ;=PC OF TRAP
1430 004324 017616 000000          MOV     @ (SP), (SP) ;GET TRP
1431 004330 006316          TRPOK: ASL    (SP)      ;MULTIPLY TRAP ARG BY 2
1432 004332 042716 177001          BIC     #177001,(SP) ;CLEAR UNWANTED BITS
1433 004336 062716 001344          ADD     #.TRPTAB,(SP);POINTER TO SUBROUTINE ADDRESS
1434 004342 017616 000000          MOV     @ (SP), (SP) ;SUBROUTINE ADDRESS
1435 004346 000136          JMP     @ (SP)+      ;GO TO SUBROUTINE
1436
1437          ;ERROR HANDLER
1438          ;-----
1439
1440 004350 032777 010000 174624 .HLT:  BIT     #SW12,@SWR      ;BELL ON ERROR?
1441 004356 001406          BEQ     XBX          ;BR IF NO BELL
1442 004360 105777 174624          TSTB   @TPCSR      ;TTY READY.
1443 004364 100003          BPL     XBX          ;DON'T WAIT IF TTY NOT READY.
1444 004366 112777 000207 174616          MOVB   #207,@TPDBR ;PUSH A BELL AT THE TTY.
1445 004374 032777 020000 174600 XBX:  BIT     #SW13,@SWR      ;DELETE ERROR PRINT OUT?
1446 004402 001105          BNE     HALTS        ;BR IF NO PRINT OUT WANTED.
1447 004404 021637 001234          CMP     (SP),LSTERR ;WAS THIS ERROR FOUND LAST TIME?
1448 004410 001404          BEQ     1$          ;BR IF YES
1449 004412 011637 001234          MOV     (SP),LSTERR ;RECORD BEING HERE
1450 004416 105037 001341          CLRB   ERRFLG      ;PREPARE HEADER
1451 004422 104406          1$:  SAVO5          ;SAVE ALL PROC REGISTERS
1452 004424 011605          MOV     (SP),R5      ;GET THE PC OF ERROR
1453 004426 162705 000002          SUB     #2,R5        ;GET ADDRESS OF TRAP CALL
1454 004432 011504          MOV     (R5),R4      ;GET HLT INSTRUCTION
1455 004434 006304          ASL     R4          ;MULT BY TWO
1456 004436 061504          ADD     (R5),R4      ;DOUBLE IT
1457 004440 006304          ASL     R4          ;MULT AGAIN
1458 004442 042704 177001          BIC     #177001,R4   ;CLEAR JUNK
1459 004446 062704 023274          ADD     #.ERRTAB,R4 ;GET POINTER
1460 004452 012437 004566          MOV     (R4)+,ERRMSG ;GET ERROR MESSAGE
1461 004456 012437 004600          MOV     (R4)+,DATAHD ;GET DATA HEADRER
1462 004462 011437 004612          MOV     (R4),DATABP ;GET DATA TABLE
1463 004466 105737 001341          TSTB   ERRFLG      ;TYPE HEADREER
1464 004472 001403          BEQ     TYPMSG      ;BR IF YES
1465 004474 005737 004612          TST     DATABP      ;DOES DATA TABLE EXIST?
1466 004500 001040          BNE     TYPDAT      ;BR IF YES.

```

1467	004502	104402	005136		TYPMSG: TYPE	,MCRLF	
1468	004506	104402	005136		TYPE	,MCRLF	
1469	004512	005737	001220		TST	LOCK	
1470	004516	001402			BEQ	1\$	
1471	004520	104402	005411		TYPE	,MASTEK	
1472	004524	104402	005377		1\$: TYPE	,MTSTN	
1473	004530	104411	005000		CNVRT	,XTSTN	;SHOW IT
1474	004534	104402	005466		TYPE	,MERRPC	;TYPE PC.
1475	004540	104411	004772		CNVRT	,ERTABO	;SHOW IT
1476	004544	104402	005136		TYPE	,MCRLF	;GIVE A CR/LF
1477	004550	112737	177777	001341	MOVB	#-1,ERRFLG	;NO MORE HEADER UNLESS NO DATA TABLE.
1478	004556	005737	004566		TST	ERRMSG	;IS THERE AN ERROR MESSAGE?
1479	004562	001402			BEQ	WRKO.FM	;BR IF NO.
1480	004564	104402			TYPE		;TYPE
1481	004566	000000			ERRMSG: 0		; ERROR MESSAGE
1482	004570				WRKO.FM:		
1483	004570	005737	004600		TST	DATAHD	;DATA HEADER?
1484	004574	001402			BEQ	TYPDAT	;BR IF NO
1485	004576	104402			TYPE		;TYPE
1486	004600	000000			DATAHD: 0		; DATA HEADER
1487	004602	005737	004612		TYPDAT: TST	DATABP	;DATA TABLE?
1488	004606	001402			BEQ	RESREG	;BR IF NO.
1489	004610	104410			CONVRT		;SHOW
1490	004612	000000			DATABP: 0		; DATA TABLE
1491	004614	104407			RESREG: RES05		;RESTORE PROC REGISTERS
1492	004616	022737	003104	000042	HALTS: CMP	#\$ENDAD,@#42	;IF ACT-11 AUTO MODE--HALT!!
1493	004624	001403			BEQ	1\$	
1494	004626	005777	174350		TST	@SWR	;HALT ON ERROR?
1495	004632	100035			BPL	EXITER	;BR IF NO HALT ON ERROR
1496	004634	010046			1\$: PUSHRO		;SAVE RO
1497	004636	016600	000002		MOV	2(SP),RO	;SHOW ERROR PC IN DATA LIGHTS
1498	004642	013746	000004		MOV	4,-(SP)	;SAVE OLD TRAP
1499	004646	013746	000006		MOV	6,-(SP)	
1500	004652	012737	004710	000004	MOV	#22\$,4	;FORCE HALT IF TIME-OUT
1501	004660	012737	000340	000006	MOV	#340,6	;WHEN REFERENCING TXCSR
1502	004666	042777	014000	174516	BIC	#\$YSTST!MEXT,@TXCSR	
1503	004674	000000			HALT		;HALT
1504	004676	012637	000006		MOV	(SP)+,6	;RESTORE TRAP
1505	004702	012637	000004		MOV	(SP)+,4	
1506	004706	000406			BR	33\$	
1507	004710	000000			22\$: HALT		;HALT
1508	004712	022626			CMP	(SP)+,(SP)+	;POP STACK
1509	004714	012637	000006		MOV	(SP)+,6	;RESTORE TRAP
1510	004720	012637	000004		MOV	(SP)+,4	
1511	004724	012600			33\$: POPRO		;GET RO
1512	004726	005237	001232		EXITER: INC	ERRCNT	;UPDATE ERROR COUNT
1513	004732	032777	000400	174242	BIT	#\$W08,@SWR	;GOTO TOP OF TEST?
1514	004740	001007			BNE	1\$	;BR IF YES
1515	004742	032777	002000	174232	BIT	#\$W10,@SWR	;GOTO NEXT TEST?
1516	004750	001407			BEQ	2\$	;BR IF NO
1517	004752	013737	001216	001214	MOV	NEXT,RETURN	;SET FOR NEXT TEST
1518	004760	012706	001150		1\$: MOV	#\$STACK,SP	;RESET SP
1519	004764	000177	174224		JMP	@RETURN	;GOTO SPECIFIED TEST
1520	004770	000002			2\$: RTI		;RETURN
1521	004772	000001			ERTABO: 1		
1522	004774	006	002		.BYTE	6,2	



```

1523 004776 001266          SAVPC
1524 005000 000001          XTSTN: 1
1525 005002 003          002      .BYTE 3,2
1526 005004 001226          TSTNO
1527 005006 017600 000000  .PKCLK: MOV @ (SP),R0      ;GET THE # OF TICKS TO POKE
1528 005012 062716 000002  ADD #2,(SP)      ;POP OVER THE #
1529 005016
1530 005016 052777 020000 174366 1$: BIS #CLK,@TXCSR      ;POKE CLOCK UP
1531 005024 005300          DEC RO      ;ARE WE DONE?
1532 005026 001405          BEQ 2$      ;YES-GO TO 2$
1533 005030 042777 020000 174354  BIC #CLK,@TXCSR      ;POKE CLOCK DOWN
1534 005036 005300          DEC RO      ;ARE WE DONE?
1535 005040 001366          BNE 1$      ;NO-REPEAT
1536 005042 000002 2$: RTI      ;RETURN
1537
1538
1539          ;WAIT ROUTINE
1540 005044 000240          SMALL: NOP      ;STALL
1541 005046 000207          RTS PC      ;RETURN
1542
1543          ;POWER FAIL ROUTINE
1544
1545 005050 012737 005060 000024  .PFAIL: MOV #PWRUP,24      ;LOAD PFAIL VECTOR FOR POWER UP
1546 005056 000000          HALT      ;
1547 005060 000005          PWRUP: RESET      ;WAIT TTY TO COME UP
1548 005062 012706 001150          MOV #STACK,SP      ;REINIT STACK POINTER
1549 005066 012737 005050 000024  MOV #.PFAIL,24      ;LOAD PFAIL VECTOR FOR POWER DOWN
1550 005074 104402          TYPE
1551 005076 005141          MPOWER
1552 005100 000177 174110          JMP @RETURN
1553          ;CLRVEC,ROUTINE TO FILL COMMUNICATION VECTOR AREA WITH .+2,HALT
1554
1555 005104 012702 000300          CLRVEC: MOV #300,R2      ;R2 COMM VECTOR AREA ADRS
1556 005110 012701 000302          MOV #302,R1      ;INIT R1 WITH ADRS OF HALT
1557 005114 010122 1$: MOV R1,(R2)+      ;MOV .+2 TO PC
1558 005116 005022          CLR (R2)+      ;MOV HALT TO PC
1559 005120 022121          CMP (R1)+,(R1)+      ;INC TO NEXT VECTOR AREA
1560 005122 022701 000776          CMP #776,R1      ;END OF VECTOR AREA
1561 005126 001372          BNE 1$      ;NO
1562 005130 000207          RTS PC      ;RETURN
1563
1564
1565
1566 005132 020040 000077          MQM: .ASCIZ / ?/
(2) 005136 005015 000          MCRLF: .ASCIZ <15><12>
(2) 005141 377 053520 020122          MPOWER: .ASCIZ <377>/PWR FAILED. /
(2) 005157 015 042777 042116          MEPASS: .ASCIZ <15><377>/END PASS CZDPDC /
(2) 005202 051377 000          MR: .ASCIZ <377>/R/
(2) 005205 377 051120 043517          MERR2: .ASCIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT./
(2) 005254 044777 051516 043125          MERR3: .ASCIZ <377>/INSUFFICIENT DATA!/
(2) 005300 052377 051505 020124          MTSTPC: .ASCIZ <377>/TEST PC-/
(2) 005312 046377 041517 020113          MLOCK: .ASCIZ <377>/LOCK ON SELECTED TEST/
(2) 005341 103 051123 020072          MCSRX: .ASCIZ /CSR: /
(2) 005347 126 041505 020072          MVECX: .ASCIZ /VEC: /
(2) 005355 120 051501 042523          MPASSX: .ASCIZ /PASSES: /
(2) 005366 051105 047522 051522          MERRX: .ASCIZ /ERRORS: /

```

```
(2) 005377 124 051505 020124 MTSTN: .ASCIZ /TEST NO: /
(2) 005411 052 000 MASTEK: .ASCIZ /*/
(2) 005413 377 042523 020124 MNEW: .ASCIZ <377>/SET SWITCH REG TO 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
1567 006044 006 003 .BYTE 6,3
1568 006046 001236 TEMP1
1569 006050 006 002 .BYTE 6,2
1570 006052 001240 TEMP2
1571 .EVEN
1572
1573 006054 000000 TEMP: 0
1574 006116 .=. +40
1575 006116 000000 MDATA: 0
1576 006160 006160 .=. +40
1577 006160 000000 INBUF: 0
1578 006222 006222 .=. +40
1579 006222 000001 TRP.PC: .BLKW 1
1580
```



```

1581
1582
1583           ;
1584           ;ROUTINE USED TO "CYCLE" THROUGH UP TO EIGHT DUP11'S
1585           ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
1586           ;AND RUNS THE SPECIFIED DUP11'S.  THIS ROUTINE *MUST*
1587           ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
1588           ;SETUP NECESSARY.
1589           ;
1590 006224 105737 001310      CYCLE:  TSTB   DUPACTV      ;ARE ANY DUP11'S TO BE TESTED?
1591 006230 001004           BNE     1$          ;BR IF OK.
1592 006232 104402 005205    TYPE   ,MERR2       ;NO DUP11'S SELECTED!!
1593 006236 000000           HALT                    ;STOP THE SHOW.
1594 006240 000776           BR     .-2         ;DISQUALIFY CONT. SW.
1595 006242 133737 001314 001310 1$:  BITB   RUN,DUPACTV ;IS THIS ONE "ACTIVE"
1596 006250 001020           BNE     2$          ;BR IF GOOD ONE FOUND.
1597 006252 000241           CLC                      ;CLEAR PROC. CARRY BIT.
1598 006254 106137 001314    ROLB   RUN          ;UPDATE POINTER
1599 006260 105537 001314    ADCB   RUN          ;CATCH CARRY FROM RUN
1600 006264 062737 000006 001316    ADD   #6,CREAM      ;UPDATE ADDRESS POINTER.
1601 006272 022737 001560 001316    CMP   #DUP.END,CREAM
1602 006300 001360           BNE     1$          ;KEEP GOING; NOT ALL TESTED FOR.
1603 006302 012737 001500 001316    MOV   #DUP.MAP,CREAM ;RESET ADDRESS POINTER.
1604 006310 000754           BR     1$          ;KEEP LOOKING FOR ACTIVE DUP11
1605 006312 000241           2$:  CLC                      ;CLEAR PROC. CARRY.
1606 006314 106137 001314    ROLB   RUN          ;UPDATE POINTER.
1607 006320 105537 001314    ADCB   RUN          ;CATCH CARRY.
1608 006324 013700 001316    MOV   CREAM,RO      ;GET ADDRESS POINTER.
1609 006330 062737 000006 001316    ADD   #6,CREAM      ;UPDATE.
1610 006336 022737 001560 001316    CMP   #DUP.END,CREAM
1611           ;
1612 006344 001003           BNE     3$          ;ALL DONE?
1613 006346 012737 001500 001316    MOV   #DUP.MAP,CREAM ;BR IF NO.
1614 006354 012037 001404           MOV   (RO)+,RXCSR    ;RESTORE POINTER.
1615 006360 012037 001374           MOV   (RO)+,DUPRVC   ;LOAD SYSTEM CTRL. REG
1616 006364 012037 001434           MOV   (RO)+,LO0.00   ;LOAD VECTOR
1617 006370 012700 000002           MOV   #2,RO          ;GET PARAMETERS
1618 006374 013737 001404 001424    MOV   RXCSR,HUPRCR   ;SAVE CORE THIS WAY!
1619 006402 005237 001424           INC   HUPRCR          ;GET CONTROL REG HIGH BYTE
1620 006406 013737 001424 001406    MOV   HUPRCR,RXDBUF  ;GET IT
1621 006414 005237 001406           INC   RXDBUF          ;GET RX CONTROL REG BUFFER
1622 006420 013737 001406 001416    MOV   RXDBUF,DUPSEC  ;GET IT
1623 006426 013737 001406 001410    MOV   RXDBUF,PARCSR  ;GET SECONDARY REG SELECT REG
1624 006434 013737 001406 001422    MOV   RXDBUF,HUPRBF  ;GET PARAMETER STATUS REGISTER
1625 006442 005237 001422           INC   HUPRBF          ;GET RX BUFFER HIGH BYTE
1626 006446 013737 001422 001420    MOV   HUPRBF,HUPPSR  ;GET IT
1627 006454 013737 001420 001412    MOV   HUPPSR,TXCSR   ;GET PAR STATUS REG HIGH BYTE
1628 006462 005237 001412           INC   TXCSR          ;GET TX CONTROL REGISTER
1629 006466 013737 001412 001430    MOV   TXCSR,HUPTCR   ;GET IT
1630 006474 005237 001430           INC   HUPTCR          ;GET TX CONTROL REG HIGH BYTE
1631 006500 013737 001430 001414    MOV   HUPTCR,TXDBUF  ;GET IT
1632 006506 005237 001414           INC   TXDBUF          ;BET TX BUFFER
1633 006512 013737 001414 001426    MOV   TXDBUF,HUPTBF  ;GET IT
1634 006520 005237 001426           INC   HUPTBF          ;GET TX BUFFER HIGH BYTE
1635
1636 006524 013737 001374 001376    MOV   DUPRVC,DUPRPS  ;RX VECTOR
  
```

```

1637 006532 060037 001376          ADD    RO,DUPRPS      ;RX PRIORITY LEVEL
1638 006536 013737 001376 001400    MOV    DUPRPS,DUPTVC
1639 006544 060037 001400          ADD    RO,DUPTVC     ;TX VECTOR
1640 006550 013737 001400 001402    MOV    DUPTVC,DUPTPS
1641 006556 060037 001402          ADD    RO,DUPTPS     ;TX PRIORITY LEVEL
1642
1643
1644 006562 012700 001434          MOV    #L00.00,RO    ;LOAD STAUS 00-00
1645 006566 012701 001432          MOV    #MASK.A,R1    ;PREPARE MASK.
1646 006572 012702 001433          MOV    #CLK.A,R2     ;PREPARE CLOCKS
1647 006576 004737 006742          JSR    PC,FIX.00     ;GO AND CALCULATE CONFIGURATION.
1648 006602 005737 000042          TST   @#42
1649 006606 001050          BNE   4$
1650 006610 032777 000002 172364    BIT   #SW01,@SWR     ;IF SW01=1,GET STARTING TEST #
1651 006616 001444          BEQ   4$
1652 006620 104402 005136          7$:   TYPE   ,MCRLF
1653 006624 104403          INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1654 006626 005377          MTSTN ;MESSAGE
1655 006630 104405          PARAM ;CONVERT STRING
1656 006632 000001          1     ;LOW LIMIT
1657 006634 001000          1000  ;HIGH LIMIT
1658 006636 001226          TSTNO ;STORE AT THIS LOCATION
1659 006640 000          .BYTE 0             ;MASK
1660 006641 001          .BYTE 1             ;HOW MANY TIMES + 2
1661 006642 012700 007160          MOV    #TST1,RO
1662 006646 022710 012737          5$:   CMP    #12737,(RO)
1663 006652 001017          BNE   6$
1664 006654 023760 001226 000002    CMP    TSTNO,2(RO)
1665 006662 001013          BNE   6$
1666 006664 022760 001226 000004    CMP    #TSTNO,4(RO)
1667 006672 001007          BNE   6$
1668 006674 010037 001214          MOV    RO,RETURN     ;SAVE PC
1669 006700 104402 005136          TYPE  ,MCRLF
1670 006704 104402 005202          TYPE  ,MR
1671 006710 000412          BR    8$
1672 006712 005720          6$:   TST   (RO)+
1673 006714 020027 021464          CMP    RO,#TLAST+10
1674 006720 001352          BNE   5$
1675 006722 104402 005132          TYPE  ,MQM
1676 006726 000734          BR    7$
1677
1678 006730 012737 007160 001214 4$:   MOV    #TST1,RETURN   ;PREPARE RETURN ADDRESS
1679 006736 000177 172252 8$:   JMP    @RETURN        ;GO START TESTING.
1680
1681 006742 011003          FIX.00: MOV   (RO),R3     ;GET PARAMETERS.
1682 006744 000207          5$:   RTS   PC           ;

```



1683					
1684					
1685					
1686	006746	012577	172422	SETVEC:	MOV (R5)+,ADUPRVC
1687	006752	012577	172422		MOV (R5)+,ADUPTVC
1688	006756	112577	172414		MOVB (R5)+,ADUPRPS
1689	006762	112577	172414		MOVB (R5)+,ADUPTPS
1690	006766	000205			RTS R5
1691	006770			NO.ATRAP:	
1692	006770	104012			HLT 12
1693	006772	000002			RTI
1694					
1695	006774			NO.BTRAP:	
1696	006774	104013			HLT 13
1697	006776	000002			RTI
1698					
1699	007000	010046		SIMBCC:	MOV R0,-(SP)
1700	007002	010146			MOV R1,-(SP)
1701	007004	010246			MOV R2,-(SP)
1702	007006	012537	001236		MOV (R5)+,TEMP1
1703	007012	012537	001240		MOV (R5)+,TEMP2
1704	007016	012537	001242		MOV (R5)+,TEMP3
1705	007022	005037	007154	1\$:	CLR BCCFBK
1706	007026	013700	001242		MOV TEMP3,R0
1707	007032	006037	001240		ROR TEMP2
1708	007036	005500			ADC R0
1709	007040	032700	000001		BIT #BIT0,R0
1710	007044	001402			BEQ 2\$
1711	007046	005137	007154		COM BCCFBK
1712	007052	013700	007152	2\$:	MOV XPOLY,R0
1713	007056	005100			COM R0
1714	007060	040037	007154		BIC R0,BCCFBK
1715	007064	000241			CLC
1716	007066	006037	001242		ROR TEMP3
1717	007072	013700	007154		MOV BCCFBK,R0
1718	007076	013701	001242		MOV TEMP3,R1
1719	007102	010102			MOV R1,R2
1720	007104	040100			BIC R1,R0
1721	007106	043702	007154		BIC BCCFBK,R2
1722	007112	050200			BIS R2,R0
1723	007114	043737	007152 001242		BIC XPOLY,TEMP3
1724	007122	050037	001242		BIS R0,TEMP3
1725	007126	005337	001236		DEC TEMP1
1726	007132	001333			BNE 1\$
1727	007134	013737	001242 007156		MOV TEMP3,CALBCC
1728	007142	012602			MOV (SP)+,R2
1729	007144	012601			MOV (SP)+,R1
1730	007146	012600			MOV (SP)+,R0
1731	007150	000205			RTS R5
1732	007152	000000		XPOLY:	0
1733	007154	000000		BCCFBK:	0
1734	007156	000000		CALBCC:	0
1735		120001		CRC16=	120001
1736		102010		CRC.CCITT=	102010
1737					
1738					

INTERRUPT PRIORITY TEST-CPU LEVEL AT 7

1739 ;\*\*\*\*\* TEST 1 \*\*\*\*\*  
1740 ;\*PRIORITY INTERRUPT TEST.  
1741 ;\*SET PROCESSOR STATUS TO PRIORITY 7  
1742 ;\*AND VERIFY THAT THE DUP11 WILL NOT INTERRUPT.  
1743 ;\*\*\*\*\*

1744  
1745 :\*\*\*\*\*  
1746 :\*  
1747 : TEST 1  
1748 :\*  
1749 :\*\*\*\*\*

1750 :\*\*\*\*\*  
1751 007160 012737 000001 001226 TST1: MOV #1,@#TSTNO  
1752 007166 012737 007264 001216 MOV #TST2,NEXT  
1753 007174 012737 000340 177776 MOV #340,PS ;LOCK OUT INTERRUPTS  
1754 007202 052777 000400 172202 BIS #MRESET,@TXCSR ;RESET THE DEVICE  
1755 007210 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH  
1756 007214 004537 006746 JSR R5,SETVEC ;SET UP VECTORS  
1757 007220 006770 NO.ATRAP ;VECTOR 'A'  
1758 007222 006774 NO.BTRAP ;VECTOR 'B'  
1759 007224 340 340 .BYTE 340,340 ;LEVEL  
1760 007226 012777 004100 172156 MOV #TXINTE!SYSTST,@TXCSR ;TURN ON DUP TX INT. ENABLEAND ENTER SYSTST MODE  
1761 007234 012737 000340 177776 MOV #340,PS ;SET CPU PRIORITY--CHANGE HERE IF NOT = 5  
1762 007242 000240 NOP ;STALL  
1763 007244 000240 NOP ;DITTO  
1764 007246 000240 NOP ;DITTO  
1765 007250 005077 172136 1\$: CLR @TXCSR ;DISABLE THE DUP11  
1766 007254 104400 SCOPE ;SCOPE THIS TEST  
1767 007256 012716 007250 2\$: MOV #1\$,(SP) ;SETUP FOR RETURN  
1768 007262 000002 RTI ;RETURN

1769  
1770  
1771 ;\*\*\*\*\* TEST 2 \*\*\*\*\*  
1772 ;\*PRIORITY INTERRUPT TEST.  
1773 ;\*SET PROCESSOR STATUS TO PRIORITY 6  
1774 ;\*AND VERIFY THAT THE DUP11 WILL NOT INTERRUPT.  
1775 ;\*\*\*\*\*

1776 :\*\*\*\*\*  
1777 :\*  
1778 : TEST 2  
1779 :\*  
1780 :\*\*\*\*\*

1781 :\*\*\*\*\*  
1782 :\*\*\*\*\*  
1783 007264 012737 000002 001226 TST2: MOV #2,@#TSTNO  
1784 007272 012737 007400 001216 MOV #TST3,NEXT  
1785 007300 122737 000005 001320 CMPB #5,PRIRTY ;COMPARE REAL WITH NORMAL  
1786 007306 001026 BNE 1\$ ;BR IF NOT A MATCH  
1787 007310 012737 000340 177776 MOV #340,PS ;LOCK OUT INTERRUPTS  
1788 007316 052777 000400 172066 BIS #MRESET,@TXCSR ;RESET THE DEVICE  
1789 007324 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH  
1790 007330 004537 006746 JSR R5,SETVEC ;SET UP VECTORS  
1791 007334 006770 NO.ATRAP ;VECTOR 'A'  
1792 007336 006774 NO.BTRAP ;VECTOR 'B'  
1793 007340 340 340 .BYTE 340,340 ;LEVEL  
1794 007342 012777 004100 172042 MOV #TXINTE!SYSTST,@TXCSR ;TURN ON DUP TX INT. ENABLEAND ENTER SYSTST MODE



INTERRUPT PRIORITY TEST-CPU LEVEL AT 6

```
1795 007350 012737 000300 177776      MOV      #300,PS      ;SET CPU PRIORITY--CHANGE HERE IF NOT = 5
1796 007356 000240                    NOP                    ;STALL
1797 007360 000240                    NOP                    ;DITTO
1798 007362 000240                    NOP                    ;DITTO
1799 007364 005077 172022      1$:     CLR      @TXCSR      ;DISABLE THE DUP11
1800 007370 104400                    SCOPE                 ;SCOPE THIS TEST
1801 007372 012716 007364      2$:     MOV      #1$, (SP)  ;SETUP FOR RETURN
1802 007376 000002                    RTI                    ;RETURN
1803
1804
1805                                     ;***** TEST 3 *****
1806                                     ;*PRIORITY INTERRUPT TEST.
1807                                     ;*SET PROCESSOR STATUS TO PRIORITY 5
1808                                     ;*AND VERIFY THAT THE DUP11 WILL NOT INTERRUPT.
1809                                     ;*****
1810
1811                                     ;*****
1812                                     ;*
1813                                     ;* TEST 3
1814                                     ;*
1815                                     ;*****
1816                                     ;*****
1817 007400 012737 000003 001226      TST3:   MOV      #3,@#TSTNO
1818 007406 012737 007514 001216      MOV      #TST4,NEXT
1819 007414 122737 000005 001320      CMPB    #5,PRIRTY    ;COMPARE REAL WITH NORMAL
1820 007422 001026                    BNE     1$            ;BR IF NOT A MATCH
1821 007424 012737 000340 177776      MOV      #340,PS     ;LOCK OUT INTERRUPTS
1822 007432 052777 000400 171752      BIS     #MRESET,@TXCSR ;RESET THE DEVICE
1823 007440 004737 005044                    JSR     PC,SMALL     ;WAIT FOR RESET TO FINISH
1824 007444 004537 006746                    JSR     R5,SETVEC    ;SET UP VECTORS
1825 007450 006770                    NO.ATRAP              ;VECTOR 'A'
1826 007452 006774                    NO.BTRAP              ;VECTOR 'B'
1827 007454          340          340      .BYTE    340,340    ;LEVEL
1828 007456 012777 004100 171726      MOV      #TXINTE!SYSTST,@TXCSR ;TURN ON DUP TX INT. ENABLEAND ENTER SYSTST MODE
1829 007464 012737 000240 177776      MOV      #240,PS     ;SET CPU PRIORITY--CHANGE HERE IF NOT = 5
1830 007472 000240                    NOP                    ;STALL
1831 007474 000240                    NOP                    ;DITTO
1832 007476 000240                    NOP                    ;DITTO
1833 007500 005077 171706      1$:     CLR      @TXCSR      ;DISABLE THE DUP11
1834 007504 104400                    SCOPE                 ;SCOPE THIS TEST
1835 007506 012716 007500      2$:     MOV      #1$, (SP)  ;SETUP FOR RETURN
1836 007512 000002                    RTI                    ;RETURN
1837
1838
1839                                     ;***** TEST 4 *****
1840                                     ;*PRIORITY INTERRUPT TEST.
1841                                     ;*SET PROCESSOR STATUS TO PRIORITY 4
1842                                     ;*AND VERIFY THAT THE DUP11 WILL INTERRUPT.
1843                                     ;*****
1844
1845                                     ;*****
1846                                     ;*
1847                                     ;* TEST 4
1848                                     ;*
1849                                     ;*****
1850                                     ;*****
```

INTERRUPT PRIORITY TEST-CPU AT LEVEL 4

```
1851 007514 012737 000004 001226 TST4: MOV #4,@#TSTNO
1852 007522 012737 007632 001216 MOV #TST5,NEXT
1853 007530 122737 000005 001320 CMPB #5,PRIRTY ;COMPARE REAL WITH NORMAL
1854 007536 001027 BNE 1$ ;BR IF NOT A MATCH
1855 007540 012737 000340 177776 MOV #340,PS ;LOCK OUT INTERRUPTS
1856 007546 052777 000400 171636 BIS #MRESET,@TXCSR ;RESET THE DEVICE
1857 007554 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
1858 007560 004537 006746 JSR R5,SETVEC ;SET UP VECTORS
1859 007564 006770 NO.ATRAP ;VECTOR 'A'
1860 007566 007624 2$ ;VECTOR 'B'
1861 007570 340 340 .BYTE 340,340 ;LEVEL
1862 007572 012777 004100 171612 MOV #TXINTE!SYSTST,@TXCSR ;TURN ON DUP TX INT. ENABLEAND ENTER SYSTST MODE
1863 007600 012737 000200 177776 MOV #200,PS ;SET CPU PRIORITY--CHANGE HERE IF NOT = 5
1864 007606 000240 NOP ;STALL
1865 007610 000240 NOP ;DITTO
1866 007612 000240 NOP ;DITTO
1867 007614 104013 HLT 13 ;DUP FAILED TO INTERRUPT-POSSBILY WRONG PRIORITY-CHANGE IF NOT 5
1868 007616 005077 171570 1$: CLR @TXCSR ;DISABLE THE DUP11
1869 007622 104400 SCOPE ;SCOPE THIS TEST
1870 007624 012716 007616 2$: MOV #1$,(SP) ;SETUP FOR RETURN
1871 007630 000002 RTI ;RETURN
```

```
***** TEST 5 *****
*TEST TO PROVE THE HALF-DUPLEX FUNCTION
*PROVE THAT THE RECEIVER WILL NOT RECOGNIZE
*DATA IF SEND IS ASSERTED.
*****
```

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

```
1885 007632 012737 000005 001226 TST5: MOV #5,@#TSTNO
1886 007640 012737 010172 001216 MOV #TST6,NEXT
1887 007646 105737 001322 TSTB TCNFLG
1888 007652 001530 BEQ 1$
1889 007654 012737 000340 177776 MOV #340,PS ;LOCK OUT INTERRUPTS
1890 007662 052777 000400 171522 BIS #MRESET,@TXCSR ;RESET THE DEVICE
1891 007670 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
1892 007674 052777 010010 171510 BIS #MEXT!HD Xen,@TXCSR ;ENTER MAINT EXT AND HALF-DUPLEX MODES
1893 007702 004537 006746 JSR R5,SETVEC ;SET UP VECTORS
1894 007706 010162 2$ ;RECEIVER
1895 007710 006774 NO.BTRAP ;TRANSMITTER
1896 007712 340 340 .BYTE 340,340 ;LEVEL
1897 007714 005037 177776 CLR PS ;LOWER PROC. STATUS
1898 007720 052777 000020 171456 BIS #RCVEN,@RXCSR ;TURN ON RECEIVER
1899 007726 052777 000100 171450 BIS #RINTEN,@RXCSR ;TURN ON INT. ENABLE
1900 007734 052777 000020 171450 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
1901 007742 012737 000005 007772 MOV #5,68$ ;LOAD THE NUMBER
1902 007750 032777 004000 171436 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
1903 007756 001374 BNE 66$ ;BR IF SET
1904 007760 032777 004000 171426 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
1905 007766 001774 BEQ 67$ ;BR IF CLEAR
1906 007770 005327 DEC (PC)+ ;DECREMENT THE NUMBER
```



```
1907 007772 000005          68$: 5          ;OF TIMES TO REPEAT
1908 007774 001365          BNE 66$          ;BR IF MORE TO GO
1909 007776 105777 171410   TSTB @TXCSR
1910 010002 100401          BMI 3$
1911 010004 104005          HLT 5          ;TXDONE FAILED TO SET
1912 010006 012777 000400 171400 3$: MOV #TSOM,@TXDBUF ;LOAD TX BUFFER
1913 010014 012737 000005 010044 MOV #5,73$      ;LOAD THE NUMBER
1914 010022 032777 004000 171364 71$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
1915 010030 001374          BNE 71$          ;BR IF SET
1916 010032 032777 004000 171354 72$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
1917 010040 001774          BEQ 72$          ;BR IF CLEAR
1918 010042 005327          DEC (PC)+       ;DECREMENT THE NUMBER
1919 010044 000005          73$: 5          ;OF TIMES TO REPEAT
1920 010046 001365          BNE 71$          ;BR IF MORE TO GO
1921 010050 105777 171336   TSTB @TXCSR    ;CHECK FOR DONE
1922 010054 100401          BMI 4$          ;BR IF SET
1923 010056 104000          HLT            ;EXTERNAL CLOCKING STOPPED
1924                                ;OR DATA WAS NOT RECEIVERD.CHECK
1925                                ;EIA DATA AND CLOCK PATHS
1926 010060 005077 171330   4$: CLR @TXDBUF  ;LOAD A CHARACTER
1927 010064 105777 171322   TSTB @TXCSR    ;CHECK FOR DONE
1928 010070 100375          BPL -4         ;BR IF NOT SET
1929 010072 012777 001000 171314 MOV #TEOM,@TXDBUF ;END THE MESSAGE
1930 010100 012737 000050 010130 MOV #40.,78$   ;LOAD THE NUMBER
1931 010106 032777 004000 171300 76$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
1932 010114 001374          BNE 76$          ;BR IF SET
1933 010116 032777 004000 171270 77$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
1934 010124 001774          BEQ 77$          ;BR IF CLEAR
1935 010126 005327          DEC (PC)+       ;DECREMENT THE NUMBER
1936 010130 000050          78$: 40.        ;OF TIMES TO REPEAT
1937 010132 001365          BNE 76$          ;BR IF MORE TO GO
1938 010134 012737 000340 177776 1$: MOV #340,PS    ;RAISE PROCESSOR STATUS
1939 010142 012706 001150   MOV #STACK,SP  ;RESET STACK
1940 010146 052777 000400 171236 BIS #MRESET,@TXCSR ;RESET THE DEVICE
1941 010154 004737 005044   JSR PC,SMALL   ;WAIT FOR RESET TO FINISH
1942 010160 104400          SCOPE          ;SCOPE THIS TEST
1943
1944 010162 104007          2$: HLT 7          ;RECEIVER INTERRUPTED AND SHOULD
1945 010164 012716 010134   MOV #1$,(SP)  ;NOT HAVE--THIS IS HALF
1946 010170 000002          RTI            ;DUPLEX.
```

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

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

```
1959 010172 012737 000006 001226 TST6: MOV #6,@#TSTNO
1960 010200 012737 010632 001216 MOV #TST7,NEXT
1961 010206 052777 000400 171176 BIS #MRESET,@TXCSR ;RESET THE DEVICE
1962 010214 004737 005044   JSR PC,SMALL   ;WAIT FOR RESET TO FINISH
```

```

1963 010220 012737 000001 001236      MOV      #1,TEMP1      ;LOAD DATA
1964 010226 005037 001240      CLR      TEMP2        ;CLEAR EXPECTED
1965 010232 012737 000340 177776      MOV      #340,PS      ;PS = 7
1966 010240 052777 004000 171144      BIS      #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
1967 010246 004537 006746      JSR      R5,SETVEC    ;LOAD INTERRUPT VECTORS
1968 010252 010446      11$     ;RECEIVER
1969 010254 010526      12$     ;TRANSMITER
1970 010256      340      340      .BYTE    340,340      ;LEVEL
1971 010260 052777 001000 171122      BIS      #CRCEN,@PARCSR ;TURN OFF CRC
1972 010266 052777 000020 171110      BIS      #RCVEN,@RXCSR  ;TURN ON THE RECEIVER
1973 010274 052777 000100 171102      BIS      #RINTEN,@RXCSR ;TURN ON REC INTERRUPT ENABLE
1974 010302 105777 171104      1$:     TSTB     @TXCSR      ;TEST FOR TX DONE
1975 010306 100375      BPL     1$           ;BR IF NOT SET
1976 010310 052777 000020 171074      2$:     BIS      #SEND,@TXCSR ;TURN ON SEND
1977 010316 012777 000400 171070      MOV      #TSOM,@TXDBUF ;TURN ON START OF MESSAGE
1978 010324 012737 000005 010354      MOV      #5,68$      ;LOAD THE NUMBER
1979 010332 032777 004000 171054      66$:    BIT      #TIMER,@TXDBUF ;CHECK THE TIMER BIT
1980 010340 001374      BNE     66$         ;BR IF SET
1981 010342 032777 004000 171044      67$:    BIT      #TIMER,@TXDBUF ;CHECK THE BIT
1982 010350 001774      BEQ     67$         ;BR IF CLEAR
1983 010352 005327      DEC     (PC)+      ;DECREMENT THE NUMBER
1984 010354 000005      68$:    5           ;OF TIMES TO REPEAT
1985 010356 001365      BNE     66$         ;BR IF MORE TO GO
1986 010360 105777 171026      3$:     TSTB     @TXCSR      ;WAIT FOR DONE
1987 010364 100401      BMI     4$           ;BR IF SET
1988 010366 104000      HLT     ;EXTERNAL CLOCKING STOPPED
1989 010370 005077 171020      4$:     CLR     @TXDBUF    ;PUSH OUT DATA
1990 010374 052777 000100 171010      BIS      #TXINTE,@TXCSR ;TURN ON TRANSMITTER INT ENABLE
1991 010402 005037 177776      CLR     PS         ;LOWER PROCESOR STATUS
1992 010406      5$:     ;
1993 010406 012737 000040 010436      MOV      #32.,73$    ;LOAD THE NUMBER
1994 010414 032777 004000 170772      71$:    BIT      #TIMER,@TXDBUF ;CHECK THE TIMER BIT
1995 010422 001374      BNE     71$         ;BR IF SET
1996 010424 032777 004000 170762      72$:    BIT      #TIMER,@TXDBUF ;CHECK THE BIT
1997 010432 001774      BEQ     72$         ;BR IF CLEAR
1998 010434 005327      DEC     (PC)+      ;DECREMENT THE NUMBER
1999 010436 000040      73$:    32.        ;OF TIMES TO REPEAT
2000 010440 001365      BNE     71$         ;BR IF MORE TO GO
2001 010442 104001      HLT     1           ;FAILED TO INTERRUPT IN TIME
2002 010444 104400      6$:     SCOPE      ;SCOPE THIS TEST

```

```

2003
2004
2005      ;INTERRUPT SERVICE ROUTINES
2006      ;-----
2007

```

```

2008      ;RECEIVER:
2009 010446 017737 170734 001324      11$:    MOV      @RXDBUF,DATA ;GET THE REGISTER AND DATA
2010 010454 123737 001240 001324      CMPB    TEMP2,DATA    ;CHECK IT
2011 010462 001401      BEQ     .+4          ;BR IF OK
2012 010464 104002      HLT     2           ;COMPARISON ERROR
2013 010466 105237 001240      INCB    TEMP2        ;COUNT UP EXPECTED
2014 010472 105737 001240      TSTB    TEMP2        ;CHECK TO SEE IF DONE
2015 010476 001012      BNE     7$           ;BR IF NO
2016 010500 105777 170700      10$:    TSTB    @RXCSR      ;CHECK FOR DONE
2017 010504 100375      BPL     10$         ;BR IF NOT YET
2018 010506 032777 001000 170672      BIT     #REOM,@RXDBUF ;CHECK FOR END OF MSG

```



```
2019 010514 001001          BNE      .+4          ;BR IF SET
2020 010516 104003          HLT      3            ;END OF MSG FAILED TO SET
2021 010520 012716 010444    MOV      #6$, (SP)    ;CRUNCH STACK
2022
2023 010524 000002          7$:     RTI            ;RETURN
2024
2025
2026                          ;TRANSMITTER:
2027 010526 113777 001236 170660 12$:     MOV      TEMP1,@TXDBUF ;LOAD THE TRANSMITTER BUFFER
2028 010534 105237 001236          INCB     TEMP1        ;UP THE COUNT
2029 010540 122737 000377 001236    CMPB    #377,TEMP1    ;ARE WE DONE
2030 010546 001026          BNE      13$         ;BR IF NO
2031 010550 012777 010560 170622    MOV      #21$,@DUPTVC ;SETUP FOR NEXT PART
2032 010556 000422          BR       13$         ;LEAVE
2033 010560 012777 000377 170626 21$:     MOV      #377,@TXDBUF ;LOAD BUFFER
2034 010566 012777 010576 170604    MOV      #22$,@DUPTVC ;SETUP NEXT PART
2035 010574 000413          BR       13$         ;LEAVE
2036 010576 012777 001000 170610 22$:     MOV      #TEOM,@TXDBUF ;SET END OF MSG
2037 010604 000240          NOP                     ;STALL
2038 010606 000240          NOP                     ;DITTO
2039 010610 042777 000120 170574    BIC      #SEND!TXINTE,@TXCSR ;TURN OFF TRANSMITTER
2040 010616 012777 006774 170554    MOV      #NO.BTRAP,@DUPTVC ;LOAD VECTOR
2041 010624 012716 010406 13$:     MOV      #5$, (SP)    ;CRUNCH STACK
2042 010630 000002          RTI            ;RETURNS
2043
2044
2045
```

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

```
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056 010632 012737 000007 001226 TST7:  MOV      #7,@TSTNO
2057 010640 012737 011432 001216    MOV      #TST10,NEXT
2058 010646 052777 000400 170536    BIS      #MRESET,@TXCSR ;RESET THE DEVICE
2059 010654 004737 005044          JSR      PC,SMALL     ;WAIT FOR RESET TO FINISH
2060 010660 005001          CLR      R1          ;CLEAR OUT DATA
2061 010662 012737 102010 007152    MOV      #CRC.CCITT,XPOLY ;SET UP THE POLYNOMIAL
2062 010670 012737 177777 007156    MOV      #-1,CALBCC    ;SETUP FOR THE FIRST TIME
2063 010676 013737 007156 010720 16$:     MOV      CALBCC,20$    ;ALLOW FOR THE NEXT CHARACTER
2064 010704 010137 010716          MOV      R1,17$      ;LOAD DATA
2065 010710 004537 007000          JSR      R5,SIMBCC    ;GO CALCULATE SOFTWARE BCC
2066 010714 000010          8.                  ;BASED ON THESE PARAMETERS
2067 010716 000001          17$:     .BLKW    1          ;DATA
2068 010720 000001          20$:     .BLKW    1          ;PREVIOUS BCC
2069 010722 105201          INCB     R1          ;INCREMENT DATA
2070 010724 001364          BNE      16$         ;BR IF MORE TO GO
2071 010726 012737 000001 001236    MOV      #1,TEMP1     ;LOAD DATA
2072 010734 005037 001240          CLR      TEMP2        ;CLEAR EXPECTED
2073 010740 012737 000340 177776    MOV      #340,PS      ;PS = 7
2074 010746 052777 004000 170436    BIS      #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
```

```

2075 010754 004537 006746      JSR      R5,SETVEC      ;LOAD INTERRUPT VECTORS
2076 010760 011146              11$      ;RECEIVER
2077 010762 011214              12$      ;TRANSMITTER
2078 010764      340      340      .BYTE    340,340      ;LEVEL
2079 010766 052777 000020 170410  BIS      #RCVEN,@RXCSR  ;TURN ON THE RECEIVER
2080 010774 052777 000100 170402  BIS      #RINTEN,@RXCSR ;TURN ON REC INTERRUPT ENABLE
2081 011002 105777 170404      1$:      TSTB      @TXCSR      ;TEST FOR TX DONE
2082 011006 100375              BPL      1$            ;BR IF NOT SET
2083 011010 052777 000020 170374  2$:      BIS      #SEND,@TXCSR ;TURN ON SEND
2084 011016 012777 000400 170370  MOV      #TSOM,@TXDBUF ;TURN ON START OF MESSAGE
2085 011024 012737 000005 011054  MOV      #5,68$       ;LOAD THE NUMBER
2086 011032 032777 004000 170354  66$:     BIT      #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2087 011040 001374              BNE      66$          ;BR IF SET
2088 011042 032777 004000 170344  67$:     BIT      #TIMER,@TXDBUF ;CHECK THE BIT
2089 011050 001774              BEQ      67$          ;BR IF CLEAR
2090 011052 005327              DEC      (PC)+        ;DECREMENT THE NUMBER
2091 011054 000005              68$:     5              ;OF TIMES TO REPEAT
2092 011056 001365              BNE      66$          ;BR IF MORE TO GO
2093 011060 105777 170326      3$:      TSTB      @TXCSR      ;WAIT FOR DONE
2094 011064 100401              BMI      4$            ;BR IF SET
2095 011066 104000              HLT                      ;EXTERNAL CLOCKING STOPPED
2096 011070 005077 170320      4$:      CLR      @TXDBUF     ;PUSH OUT DATA
2097 011074 052777 000100 170310  BIS      #TXINTE,@TXCSR ;TURN ON TRANSMITTER INT ENABLE
2098 011102 005037 177776      CLR      PS            ;LOWER PROCESOR STATUS
2099 011106              5$:
2100 011106 012737 000040 011136  MOV      #32.,73$      ;LOAD THE NUMBER
2101 011114 032777 004000 170272  71$:     BIT      #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2102 011122 001374              BNE      71$          ;BR IF SET
2103 011124 032777 004000 170262  72$:     BIT      #TIMER,@TXDBUF ;CHECK THE BIT
2104 011132 001774              BEQ      72$          ;BR IF CLEAR
2105 011134 005327              DEC      (PC)+        ;DECREMENT THE NUMBER
2106 011136 000040              73$:     32.          ;OF TIMES TO REPEAT
2107 011140 001365              BNE      71$          ;BR IF MORE TO GO
2108 011142 104001              HLT      1              ;FAILED TO INTERRUPT IN TIME
2109 011144 104400      6$:      SCOPE          ;SCOPE THIS TEST
2110
2111
2112      ;INTERRUPT SERVICE ROUTINES
2113      ;-----
2114
2115      ;RECEIVER:
2116 011146 017737 170234 001324  11$:     MOV      @RXDBUF,DATA  ;GET THE REGISTER AND DATA
2117 011154 123737 001240 001324  CMPB     TEMP2,DATA    ;CHECK IT
2118 011162 001401              BEQ      .+4           ;BR IF OK
2119 011164 104002              HLT      2              ;COMPARISON ERROR
2120 011166 105237 001240      INCB     TEMP2         ;COUNT UP EXPECTED
2121 011172 105737 001240      TSTB     TEMP2         ;CHECK TO SEE IF DONE
2122 011176 001005              BNE      7$            ;BR IF NO
2123 011200 004537 006746      JSR      R5,SETVEC    ;YES--RESET THE VECTORS
2124 011204 011320              14$      ;RECEIVER
2125 011206 011214              12$      ;TRANSMITTER
2126 011210      340      340      .BYTE    340,340      ;LEVEL
2127
2128 011212 000002      7$:      RTI              ;RETURN
2129
2130

```



```

2131                                     ;TRANSMITTER:
2132 011214 113777 001236 170172 12$:  MOVB  TEMP1,@TXDBUF ;LOAD THE TRANSMITTER BUFFER
2133 011222 105237 001236                INCB  TEMP1          ;UP THE COUNT
2134 011226 122737 000377 001236        CMPB  #377,TEMP1      ;ARE WE DONE
2135 011234 001026                BNE   13$           ;BR IF NO
2136 011236 012777 011246 170134        MOV   #21$,@DUPTVC   ;SETUP FOR NEXT PART
2137 011244 000422                BR    13$           ;LEAVE
2138 011246 012777 000377 170140 21$:  MOV   #377,@TXDBUF   ;LOAD BUFFER
2139 011254 012777 011264 170116        MOV   #22$,@DUPTVC   ;SETUP NEXT PART
2140 011262 000413                BR    13$           ;LEAVE
2141 011264 012777 001000 170122 22$:  MOV   #TEOM,@TXDBUF  ;SET END OF MSG
2142 011272 000240                NOP                    ;STALL
2143 011274 000240                NOP                    ;DITTO
2144 011276 042777 000120 170106        BIC   #SEND!TXINTE,@TXCSR ;TURN OFF TRANSMITTER
2145 011304 012777 006774 170066        MOV   #NO.BTRAP,@DUPTVC ;LOAD VECTOR
2146 011312 012716 011106 13$:  MOV   #5$, (SP)      ;CRUNCH STACK
2147 011316 000002                RTI                    ;RETURNS
2148
2149 011320 117737 170062 001324 14$:  MOVB  @RXDBUF,DATA   ;GET FIRST PART OF CRC
2150 011326 105777 170052                TSTB  @RXCSR         ;WAIT FOR SECOND PART
2151 011332 100375                BPL   .-4            ;DITTO
2152 011334 017737 170046 001242        MOV   @RXDBUF,TEMP3  ;GET THE REST OF THE CRC
2153 011342 113737 001242 001325        MOVB  TEMP3,DATA+1   ;SET UP CRC CHARACTER
2154 011350 012716 011356                MOV   #15$, (SP)     ;SETUP FOR RETURN
2155 011354 000002                RTI                    ;RETURN
2156 011356 012737 000340 177776 15$:  MOV   #340,PS        ;RAISE PS
2157 011364 005137 007156                COM   CALBCC         ;INVERT BCC
2158 011370 023737 007156 001324        CMP   CALBCC,DATA    ;COMPARE SOFTWARE AND HARDWARE BCC
2159 011376 001401                BEQ   .+4            ;BR IF OK
2160 011400 104004                HLT   4              ;BCC COMPARISON ERROR
2161 011402 032737 010000 001242        BIT   #CRCERR,TEMP3  ;CHECK THE ERROR BIT
2162 011410 001401                BEQ   .+4            ;BR IF NO ERROR
2163 011412 104004                HLT   4              ;BCC ERROR--RECEIVER DOESN'T
2164                                     ;AGREE WITH WHAT TX SENT
2165 011414 052777 000400 167770        BIS   #MRESET,@TXCSR ;RESET THE DEVICE
2166 011422 004737 005044                JSR   PC,SMALL       ;WAIT FOR RESET TO FINISH
2167 011426 000137 011144                JMP   6$             ;LEAVE
2168

```

```

***** TEST 10 *****
;*TEST OF THE DUP RUNNING A BINARY COUNT
;*PATTERN WITH A CRC CALCULATION
*****

```

```

*****
*
* TEST 10
*
*****

```

```

2179
2180 011432 012737 000010 001226 TST10: MOV   #10,@#TSTNO
2181 011440 012737 012240 001216        MOV   #TST11,NEXT
2182 011446 052777 000400 167736        BIS   #MRESET,@TXCSR ;RESET THE DEVICE
2183 011454 004737 005044                JSR   PC,SMALL       ;WAIT FOR RESET TO FINISH
2184 011460 105737 001322                TSTB  TCNFLAG
2185 011464 001532                BEQ   6$
2186 011466 005001                CLR   R1             ;CLEAR OUT DATA

```

BINARY PATTERN TEST WITH BCC IN EXTERNAL MODE

```

2187 011470 012737 102010 007152      MOV      #CRC.CCITT,XPOLY      ;SET UP THE POLYNOMIAL
2188 011476 012737 177777 007156      MOV      #-1,CALBCC           ;SETUP FOR THE FIRST TIME
2189 011504 013737 007156 011526 16$:  MOV      CALBCC,20$          ;ALLOW FOR THE NEXT CHARACTER
2190 011512 010137 011524          MOV      R1,17$              ;LOAD DATA
2191 011516 004537 007000          JSR      R5,SIMBCC           ;GO CALCULATE SOFTWARE BCC
2192 011522 000010          8.                            ;BASED ON THESE PARAMETERS
2193 011524 000001          17$: .BLKW 1 ;DATA
2194 011526 000001          20$: .BLKW 1 ;PREVIOUS BCC
2195 011530 105201          INCB    R1 ;INCREMENT DATA
2196 011532 001364          BNE     16$ ;BR IF MORE TO GO
2197 011534 012737 000001 001236  MOV      #1,TEMP1           ;LOAD DATA
2198 011542 005037 001240          CLR     TEMP2              ;CLEAR EXPECTED
2199 011546 012737 000340 177776  MOV      #340,PS            ;PS = 7
2200 011554 052777 010000 167630  BIS      #MEXT,@TXCSR
2201 011562 004537 006746          JSR      R5,SETVEC         ;LOAD INTERRUPT VECTORS
2202 011566 011754          11$
2203 011570 012022          12$
2204 011572 340 340          .BYTE 340,340 ;LEVEL
2205 011574 052777 000020 167602  BIS      #RCVEN,@RXCSR     ;TURN ON THE RECEIVER
2206 011602 052777 000100 167574  BIS      #RINTEN,@RXCSR    ;TURN ON REC INTERRUPT ENABLE
2207 011610 105777 167576          1$: TSTB  @TXCSR ;TEST FOR TX DONE
2208 011614 100375          BPL     1$ ;BR IF NOT SET
2209 011616 052777 000020 167566  2$: BIS  #SEND,@TXCSR ;TURN ON SEND
2210 011624 012777 000400 167562  MOV      #TSOM,@TXDBUF     ;TURN ON START OF MESSAGE
2211 011632 012737 000005 011662  MOV      #5,68$           ;LOAD THE NUMBER
2212 011640 032777 004000 167546  66$: BIT  #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2213 011646 001374          BNE     66$ ;BR IF SET
2214 011650 032777 004000 167536  67$: BIT  #TIMER,@TXDBUF ;CHECK THE BIT
2215 011656 001774          BEQ     67$ ;BR IF CLEAR
2216 011660 005327          DEC     (PC)+ ;DECREMENT THE NUMBER
2217 011662 000005          68$: 5 ;OF TIMES TO REPEAT
2218 011664 001365          BNE     66$ ;BR IF MORE TO GO
2219 011666 105777 167520          3$: TSTB  @TXCSR ;WAIT FOR DONE
2220 011672 100401          BMI     4$ ;BR IF SET
2221 011674 104000          HLT
2222 011676 005077 167512          4$: CLR  @TXDBUF ;EXTERNAL CLOCKING STOPPED
2223 011702 052777 000100 167502  BIS      #TXINTE,@TXCSR   ;PUSH OUT DATA
2224 011710 005037 177776          CLR     PS ;TURN ON TRANSMITTER INT ENABLE
2225 011714          5$: ;LOWER PROCESOR STATUS
2226 011714 012737 000040 011744  MOV      #32.,73$ ;LOAD THE NUMBER
2227 011722 032777 004000 167464  71$: BIT  #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2228 011730 001374          BNE     71$ ;BR IF SET
2229 011732 032777 004000 167454  72$: BIT  #TIMER,@TXDBUF ;CHECK THE BIT
2230 011740 001774          BEQ     72$ ;BR IF CLEAR
2231 011742 005327          DEC     (PC)+ ;DECREMENT THE NUMBER
2232 011744 000040          73$: 32. ;OF TIMES TO REPEAT
2233 011746 001365          BNE     71$ ;BR IF MORE TO GO
2234 011750 104001          HLT     1 ;FAILED TO INTERRUPT IN TIME
2235 011752 104400          6$: SCOPE ;SCOPE THIS TEST
2236
2237
2238 ;INTERRUPT SERVICE ROUTINES
2239 ;-----
2240
2241 ;RECEIVER:
2242 011754 017737 167426 001324 11$: MOV  @RXDBUF,DATA ;GET THE REGISTER AND DATA
  
```



BINARY PATTERN TEST WITH BCC IN EXTERNAL MODE

```

2243 011762 123737 001240 001324      CMPB   TEMP2,DATA      ;CHECK IT
2244 011770 001401                      BEQ     .+4             ;BR IF OK
2245 011772 104002                      HLT     2               ;COMPARISON ERROR
2246 011774 105237 001240      INCB   TEMP2           ;COUNT UP EXPECTED
2247 012000 105737 001240      TSTB  TEMP2           ;CHECK TO SEE IF DONE
2248 012004 001005                      BNE     7$             ;BR IF NO
2249 012006 004537 006746      JSR    R5,SETVEC      ;YES--RESET THE VECTORS
2250 012012 012126                      14$
2251 012014 012022                      12$
2252 012016 340 340      .BYTE  340,340        ;LEVEL
2253
2254 012020 000002      7$:   RTI              ;RETURN
2255
2256
2257      ;TRANSMITTER:
2258 012022 113777 001236 167364 12$:  MOVB   TEMP1,@TXDBUF   ;LOAD THE TRANSMITTER BUFFER
2259 012030 105237 001236                      INCB   TEMP1           ;UP THE COUNT
2260 012034 122737 000377 001236      CMPB   #377,TEMP1     ;ARE WE DONE
2261 012042 001026                      BNE     13$           ;BR IF NO
2262 012044 012777 012054 167326      MOV    #21$,@DUPTVC   ;SETUP FOR NEXT PART
2263 012052 000422                      BR      13$           ;LEAVE
2264 012054 012777 000377 167332 21$:  MOV    #377,@TXDBUF   ;LOAD BUFFER
2265 012062 012777 012072 167310      MOV    #22$,@DUPTVC   ;SETUP NEXT PART
2266 012070 000413                      BR      13$           ;LEAVE
2267 012072 012777 001000 167314 22$:  MOV    #TEOM,@TXDBUF  ;SET END OF MSG
2268 012100 000240                      NOP
2269 012102 000240                      NOP
2270 012104 042777 000120 167300      BIC    #SEND!TXINTE,@TXCSR ;TURN OFF TRANSMITTER
2271 012112 012777 006774 167260      MOV    #NO.BTRAP,@DUPTVC ;LOAD VECTOR
2272 012120 012716 011714      13$:  MOV    #5$, (SP)      ;CRUNCH STACK
2273 012124 000002                      RTI              ;RETURNS
2274
2275 012126 117737 167254 001324 14$:  MOVB   @RXDBUF,DATA    ;GET FIRST PART OF CRC
2276 012134 105777 167244                      TSTB  @RXCSR          ;WAIT FOR SECOND PART
2277 012140 100375                      BPL    .-4            ;DITTO
2278 012142 017737 167240 001242      MOV    @RXDBUF,TEMP3   ;GET THE REST OF THE CRC
2279 012150 113737 001242 001325      MOVB   TEMP3,DATA+1    ;SET UP CRC CHARACTER
2280 012156 012716 012164                      MOV    #15$, (SP)     ;SETUP FOR RETURN
2281 012162 000002                      RTI              ;RETURN
2282 012164 012737 000340 177776 15$:  MOV    #340,PS         ;RAISE PS
2283 012172 005137 007156                      COM    CALBCC          ;INVERT BCC
2284 012176 023737 007156 001324      CMP    CALBCC,DATA     ;COMPARE SOFTWARE AND HARDWARE BCC
2285 012204 001401                      BEQ    .+4            ;BR IF OK
2286 012206 104004                      HLT    4               ;BCC COMPARISON ERROR
2287 012210 032737 010000 001242      BIT    #CRCERR,TEMP3   ;CHECK THE ERROR BIT
2288 012216 001401                      BEQ    .+4            ;BR IF NO ERROR
2289 012220 104004                      HLT    4               ;BCC ERROR--RECEIVER DOESN'T
2290                                ;AGREE WITH WHAT TX SENT
2291 012222 052777 000400 167162      BIS    #MRESET,@TXCSR  ;RESET THE DEVICE
2292 012230 004737 005044                      JSR    PC,SMALL        ;WAIT FOR RESET TO FINISH
2293 012234 000137 011752                      JMP    6$             ;LEAVE
2294
2295
2296
2297
2298

```

```

:***** TEST 11 *****
:*THIS TEST WILL CHECK FOR ABORT SEQUENCE
:*OF THE DUP IN A DATA STREAM
:*****

```

```
2299          ;:*****
2300          ;:
2301          ;: TEST 11
2302          ;:
2303          ;:*****
2304          ;:*****
2305 012240 012737 000011 001226 TST11: MOV #11,@#TSTNO
2306 012246 012737 012554 001216      MOV #TST12,NEXT
2307 012254 052777 000400 167130      BIS #MRESET,@TXCSR ;RESET THE DEVICE
2308 012262 004737 005044      JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2309 012266 004537 006746      JSR R5,SETVEC ;SET UP INTERRUPT VECTORS
2310 012272 012452      4$ ;BASED ON THESE
2311 012274 006774      NO.BTRAP ;PARAMETERS
2312 012276      340 340 .BYTE 340,340 ;LEVEL
2313 012300 005000      CLR R0 ;CLEAR
2314 012302 005003      CLR R3 ;DITTO
2315 012304 012737 000340 177776      MOV #340,PS ;PS=7
2316 012312 052777 010377 167070      BIS #PRISEC!377,@PARCSR ;LOAD SEC STATION AND ADRS
2317 012320 052777 000120 167056      BIS #RCVEN!RINTEN,@RXCSR ;TURN ON THE RECEIVER
2318 012326 052777 004020 167056      BIS #SEND!SYSTST,@TXCSR ;TURN ON TRANSMITTER
2319 012334 005037 177776      CLR PS
2320 012340 105777 167046      1$: TSTB @TXCSR ;CHECK FOR TXDONE
2321 012344 100375      BPL 1$ ;BR IF NOT SET
2322 012346 052777 000400 167040      BIS #TSOM,@TXDBUF ;TURN ON START OF MSG
2323 012354 105777 167032      2$: TSTB @TXCSR ;WAIT FOR DONE
2324 012360 100375      BPL 2$ ;AND THEN
2325 012362 012777 000377 167024      3$: MOV #377,@TXDBUF ;LOAD A CHARACTER
2326 012370 005200      INC R0 ;UPDATE CHARACTER COUNTER
2327 012372 022700 000005      CMP #5,R0 ;ARE ALL CHARACTERS LOADED?
2328 012376 001366      BNE 2$ ;BR IF NO
2329 012400 052777 002000 167006      BIS #TABORT,@TXDBUF ;TURN ON ABORT
2330 012406 012737 000310 012436      MOV #200.,68$ ;LOAD THE NUMBER
2331 012414 032777 004000 166772      66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2332 012422 001374      BNE 66$ ;BR IF SET
2333 012424 032777 004000 166762      67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2334 012432 001774      BEQ 67$ ;BR IF CLEAR
2335 012434 005327      DEC (PC)+ ;DECREMENT THE NUMBER
2336 012436 000310      68$: 200. ;OF TIMES TO REPEAT
2337 012440 001365      BNE 66$ ;BR IF MORE TO GO
2338 012442 104001      HLT 1 ;RECEIVER DID NOT INTERRUPT IN TIME
2339 012444 012706 001150      11$: MOV #STACK,SP ;RESET STACK
2340 012450 104400      SCOPE ;SCOPE THIS TEST
2341          ;RECEIVER INTERRUPT SERVICE ROUTINE
2342 012452 017701 166726      4$: MOV @RXCSR,R1 ;GET THE CONTROL REGISTER
2343 012456 017702 166724      MOV @RXDBUF,R2 ;GET THE BUFFER
2344 012462 032701 000200      BIT #RXDONE,R1 ;CHECK FOR DONE
2345 012466 001001      BNE 5$ ;BR IF DONE SET
2346 012470 104007      HLT 7 ;FALSE INTERRUPT
2347 012472 122702 000377      5$: CMPB #377,R2 ;CHECK DATA CHARACTER
2348 012476 001401      BEQ 6$ ;BR IF A MATCH
2349 012500 104002      HLT 2 ;DATA ERROR
2350 012502 005203      6$: INC R3 ;INC THE # OF CHARS TO DO
2351 012504 022703 000003      CMP #3,R3 ;CHECK TO SEE IF DONE
2352 012510 001020      BNE 10$ ;BR IF MORE TO GO
2353 012512 105777 166666      12$: TSTB @RXCSR ;CHECK FOR
2354 012516 100375      BPL 12$ ;DONE
```



DATA STREAM ABORT SEQUENCE TEST

```
2355 012520 017702 166662      MOV    @RXDBUF,R2      ;READ THE BUFFER
2356 012524 032702 002000      BIT    #RABORT,R2     ;TEST ABORT
2357 012530 001001              BNE    7$             ;BR IF SET
2358 012532 104010              HLT    10             ;FAILED TO RECEIVE ABORT
2359 012534 012716 012444      7$:  MOV    #11$, (SP)   ;SET UP FOR RETURN
2360 012540 052777 000400 166644  BIS    #MRESET,@TXCSR ;RESET THE DEVICE
2361 012546 004737 005044      JSR    PC,SMALL       ;WAIT FOR RESET TO FINISH
2362 012552 000002      10$: RTI              ;RETURN
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378 012554 012737 000012 001226  TST12: MOV    #12,@#TSTNO
2379 012562 012737 013344 001216  MOV    #TST13,NEXT
2380 012570 052777 000400 166614  BIS    #MRESET,@TXCSR ;RESET THE DEVICE
2381 012576 004737 005044      JSR    PC,SMALL       ;WAIT FOR RESET TO FINISH
2382 012602 005000      CLR    R0             ;CLEAR FOR SOFTWARE
2383 012604 004537 006746      JSR    R5,SETVEC     ;SET UP THE VECTORS
2384 012610 013034      3$                   ;RECEIVER
2385 012612 013234      14$                  ;TRANSMITTER
2386 012614      340      340      .BYTE 340,340      ;LEVEL
2387 012616 012737 000340 177776  MOV    #340,PS        ;PROC STATUS=7
2388 012624 052777 001000 166556  BIS    #CRCEN,@PARCSR
2389 012632 052777 000120 166544  BIS    #RCVEN!RINTEN,@RXCSR ;TURN ON RECEIVER
2390 012640 052777 004020 166544  BIS    #SEND!SYSTST,@TXCSR ;START TRANSMITTER
2391 012646 005037 177776      CLR    PS             ;LOWER PS
2392 012652 105777 166534      1$:  TSTB   @TXCSR      ;CHECK FOR DONE
2393 012656 100375      BPL    1$             ;BR IF NOT YET
2394 012660 052777 000400 166526  BIS    #TSOM,@TXDBUF  ;TURN ON START OF MSG
2395 012666 052777 000100 166516  BIS    #TXINTE,@TXCSR ;TURN ON INT. ENABLE
2396 012674 012737 000764 012724  MOV    #500.,68$     ;LOAD THE NUMBER
2397 012702 032777 004000 166504  66$: BIT    #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2398 012710 001374      BNE    66$           ;BR IF SET
2399 012712 032777 004000 166474  67$: BIT    #TIMER,@TXDBUF ;CHECK THE BIT
2400 012720 001774      BEQ    67$           ;BR IF CLEAR
2401 012722 005327      DEC    (PC)+        ;DECREMENT THE NUMBER
2402 012724 000764      68$: 500.          ;OF TIMES TO REPEAT
2403 012726 001365      BNE    66$           ;BR IF MORE TO GO
2404 012730 104001      HLT    1             ;DEVICE FAILED TO INTERRUPT IN TIME
2405 012732 012706 001150      2$:  MOV    #STACK,SP ;RESET THE STACK
2406 012736 104400      SCOPE ;SCOPE THIS TEST
2407 012740 004537 006746      20$: JSR    R5,SETVEC     ;SET UP VECTORS
2408 012744 013164      23$                   ;RECEIVER
2409 012746 006774      NO.BTRAP ;TRANSMITTER
2410 012750      340      340      .BYTE 340,340 ;LEVEL
```

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

```
*****
*
* TEST 12
*
*****
```

```

2411 012752 052777 000020 166424      BIS      #RCVEN,@RXCSR
2412 012760 105777 166426      21$:    TSTB     @TXCSR      ;TEST DONE
2413 012764 100375          BPL      21$          ;BR IF NOT SET
2414 012766 012777 000070 166420      MOV      #70,@TXDBUF ;PUSH OUT DATA CHARACTER
2415 012774 012737 000062 013024      MOV      #50,,73$    ;LOAD THE NUMBER
2416 013002 032777 004000 166404      71$:    BIT      #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2417 013010 001374          BNE      71$          ;BR IF SET
2418 013012 032777 004000 166374      72$:    BIT      #TIMER,@TXDBUF ;CHECK THE BIT
2419 013020 001774          BEQ      72$          ;BR IF CLEAR
2420 013022 005327          DEC      (PC)+      ;DECREMENT THE NUMBER
2421 013024 000062      73$:    50.          ;OF TIMES TO REPEAT
2422 013026 001365          BNE      71$          ;BR IF MORE TO GO
2423 013030 104001          HLT      1           ;FAILED TO INTERRUPT IN TIME
2424 013032 000737          BR       2$          ;FINISH
2425          ;INTERRUPT SVC ROUTINES
2426
2427          ;RECEIVER
2428 013034 017704 166344      3$:    MOV      @RXCSR,R4 ;GET THE CONTROL REGISTER
2429 013040 017705 166342      MOV      @RXDBUF,R5 ;GET THE BUFFER
2430 013044 032705 000400      BIT      #RSOM,R5   ;CHECK FOR START OF MSG
2431 013050 001001          BNE      4$          ;BR IF SET
2432 013052 104011          HLT      11          ;FAILED TO RECEIVE SOM
2433 013054 032704 000200      4$:    BIT      #RXDONE,R4 ;CHECK FOR DONE
2434 013060 001001          BNE      5$          ;BR IF SET
2435 013062 104007          HLT      7           ;FALSE INTERRUPT
2436 013064 122705 000377      5$:    CMPB     #377,R5 ;CHECK DATA
2437 013070 001401          BEQ      6$          ;BR IF A MATCH
2438 013072 104002          HLT      2           ;DATA ERROR
2439 013074 012777 013104 166272      6$:    MOV      #10$,@DUPRVC ;RELOAD THE VECTOR
2440 013102 000002      7$:    RTI          ;RETURN
2441 013104 017705 166276      10$:   MOV      @RXDBUF,R5 ;GET THE BUFFER
2442 013110 122705 000377      CMPB     #377,R5   ;CHECK THE CHARACTER
2443 013114 001401          BEQ      11$         ;BR IF A MATCH
2444 013116 104002          HLT      2           ;DATA ERROR
2445 013120 042777 000020 166256      11$:   BIC      #RCVEN,@RXCSR ;TURN OFF THE RECEIVER
2446 013126 012777 013136 166240      MOV      #12$,@DUPRVC ;RELOAD THE VECTOR
2447 013134 000762          BR       7$          ;RETURN
2448 013136 017704 166242      12$:   MOV      @RXCSR,R4 ;GET THE CONTROL REGISTER
2449 013142 017705 166240      MOV      @RXDBUF,R5 ;GET THE BUFFER
2450 013146 122705 000252      CMPB     #252,R5   ;CHECK THE CHARACTER
2451 013152 001402          BEQ      13$         ;BR IF A MATCH
2452 013154 104007          HLT      7           ;FALSE INTERRUPT
2453 013156 000751          BR       7$          ;
2454 013160 104007      13$:   HLT      7           ;DEVICE INTERRUPTED AFTER RX ENABLE
2455 013162 000747          BR       7$          ;WAS CLEARED
2456 013164 017704 166214      23$:   MOV      @RXCSR,R4 ;GET THE CONTROL REG
2457 013170 017705 166212      MOV      @RXDBUF,R5 ;GET THE BUFFER
2458 013174 032715 000400      BIT      #RSOM,(R5) ;CHECK START OF MSG
2459 013200 001001          BNE      24$         ;BR IF SET
2460 013202 104011          HLT      11          ;SOM FAILED TO SET
2461 013204 122705 000070      24$:   CMPB     #70,R5   ;CHECK DATA
2462 013210 001401          BEQ      25$         ;BR IF A MATCH
2463 013212 104002          HLT      2           ;DATA FAILED TO MATCH AFTER
2464          ;RESTARTING RECEIVER
2465 013214      25$:
2466 013214 052777 000400 166170      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
  
```



```
2467 013222 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2468 013226 012716 012732 MOV #2$, (SP) ;CRUNCH STACK
2469 013232 000002 RTI ;RETURN
2470 ;TRANSMITTER
2471 013234 105777 166152 14$: TSTB @TXCSR ;CHECK DONE
2472 013240 100401 BMI 30$ ;BR IF SET
2473 013242 104007 HLT 7 ;FALSE INTERRUPT
2474 013244 012777 000377 166142 30$: MOV #377,@TXDBUF ;LOAD A CHARACTER
2475 013252 005200 INC R0 ;IN THE # TO DO
2476 013254 022700 000002 CMP #2,R0 ;CHECK TO SEE IF ALL ARE SENT
2477 013260 001030 BNE 15$ ;BR IF MORE TO GO
2478 013262 012777 013274 166110 MOV #16$,@DUPTVC ;RELOAD THE VECT
2479 013270 005000 CLR R0 ;CLEAR CHAR COUNT
2480 013272 000423 BR 15$
2481 013274 105777 166112 16$: TSTB @TXCSR ;TEST DONE
2482 013300 100401 BMI 17$ ;BR IF SET
2483 013302 104007 HLT 7 ;FALSE INTERRUPT
2484 013304 012777 000252 166102 17$: MOV #252,@TXDBUF ;LOAD A DATA CHARACTER
2485 013312 005200 INC R0 ;INC THE # TO DO
2486 013314 022700 000003 CMP #3,R0 ;CHECK FOR ALL DONE
2487 013320 001010 BNE 15$ ;BR IF MORE TO GO
2488 013322 012777 001400 166064 MOV #TEOM!TSOM,@TXDBUF ;END MSG
2489 013330 042777 000100 166054 BIC #TXINTE,@TXCSR
2490 013336 012716 012740 MOV #20$, (SP) ;CRUNCH STACK
2491 013342 000002 15$: RTI
```

```
***** TEST 13 *****
*THIS TEST WILL TRANSMIT CONTIGUOUS ONES CHARACTERS
*IN SECONDARY MODE WITH A BCC CHECK.
*****
```

```
*****
:
: TEST 13
:
: *****
```

```
2503 013344 012737 000013 001226 TST13: MOV #13,@TSTNO
2504 013352 012737 014004 001216 MOV #TST14,NEXT
2505 013360 052777 000400 166024 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2506 013366 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2507 013372 012737 000340 177776 MOV #340,PS ;SET STATUS=7
2508 013400 005000 CLR R0
2509 013402 005002 CLR R2 ;SETUP FOR SOFTWARE
2510 013404 012701 000377 MOV #377,R1 ;CALCULATION OF BCC
2511 013410 012737 102010 007152 MOV #CRC.CCITT,XPOLY ;LOAD THE POLYNOMIAL
2512 013416 012737 177777 007156 MOV #-1,CALBCC ;SETUP FOR FIRST TIME
2513 013424 013737 007156 013446 1$: MOV CALBCC,3$ ;ALLOW FOR THE NEXT CHARACTER
2514 013432 010137 013444 MOV R1,2$ ;LOAD DATA
2515 013436 004537 007000 JSR R5,SIMBCC ;GO CALCULATE SOFTWARE BCC
2516 013442 000010 8. ;BASED ON THOSE PARAMETERS
2517 013444 000001 2$: .BLKW 1 ;DATA
2518 013446 000001 3$: .BLKW 1 ;PREVIOUS BCC
2519 013450 005200 INC R0 ;INC THE # OF CHARS TO DO
2520 013452 022700 000005 CMP #5,R0 ;ARE WE DONE?
2521 013456 001362 BNE 1$ ;BR IF NO
2522 013460 005000 CLR R0 ;CLEAR OUT HOLD
```

```

2523 013462 004537 006746      JSR    R5,SETVEC      ;LOAD INTERRUPT VECTORS
2524 013466 013616              6$      ;RECEIVER
2525 013470 013722              11$     ;TRANSMITTER
2526 013472      340      340      .BYTE   340,340      ;LEVEL
2527 013474 052777 010377 165706  BIS    #PRISEC!377,@PARCSR ;ENTER SECONDARY MODE
2528 013502 052777 000120 165674  BIS    #RCVEN!RINTEN,@RXCSR ;TURN ON RECEIVER AND INTERRUPTS
2529 013510 052777 004020 165674  BIS    #SEND!SYSTST,@TXCSR  ;TURN ON TRANSMITTER
2530 013516 105777 165670      20$:    TSTB   @TXCSR
2531 013522 100375              BPL    20$
2532 013524 012777 000400 165662  MOV    #TSOM,@TXDBUF    ;START MESSAGE
2533 013532 052777 000100 165652  BIS    #TXINTE,@TXCSR   ;TURN ON INTERRUPT ENABLE
2534 013540 005037 177776      CLR    PS              ;LOWER PS
2535 013544              4$:
2536 013544 012737 000040 013574  MOV    #32,,68$        ;LOAD THE NUMBER
2537 013552 032777 004000 165634  66$:    BIT    #TIMER,@TXDBUF  ;CHECK THE TIMER BIT
2538 013560 001374              BNE    66$             ;BR IF SET
2539 013562 032777 004000 165624  67$:    BIT    #TIMER,@TXDBUF  ;CHECK THE BIT
2540 013570 001774              BEQ    67$             ;BR IF CLEAR
2541 013572 005327              DEC    (PC)+           ;DECREMENT THE NUMBER
2542 013574 000040              68$:    32.            ;OF TIMES TO REPEAT
2543 013576 001365              BNE    66$             ;BR IF MORE TO GO
2544 013600 104001              HLT    1              ;FAILED TO INTERRUPT IN TIME
2545 013602              5$:
2546 013602 052777 000400 165602  BIS    #MRESET,@TXCSR  ;RESET THE DEVICE
2547 013610 004737 005044      JSR    PC,SMALL        ;WAIT FOR RESET TO FINISH
2548 013614 104400              SCOPE                 ;SCOPE THIS TEST
2549
2550              ;INTERRUPT SERVICE ROUTINES.
2551
2552              ;RECEIVER
2553 013616 017737 165564 001324  6$:    MOV    @RXDBUF,DATA   ;GET THE DATA
2554 013624 120137 001324      CMPB   R1,DATA         ;CHECK IT
2555 013630 001401              BEQ    .+4             ;BR IF A MATCH
2556 013632 104002              HLT    2              ;DATA ERROR
2557 013634 005200              INC    R0              ;UPDATE THE # OF CHARS TO DO
2558 013636 022700 000004      CMP    #4,R0           ;CHECK FOR ALL CHARS DONE
2559 013642 001003              BNE    7$             ;BR IF MORE TO GO
2560 013644 012777 013654 165522  MOV    #10$,@DUPRVC    ;SETUP TO GET BCC
2561 013652 000002              7$:    RTI              ;RETURN
2562
2563 013654 117737 165526 001324  10$:   MOVB   @RXDBUF,DATA    ;GET THE FIRST HALF OF BCC
2564 013662 105777 165516      TSTB   @RXCSR          ;WAIT FOR
2565 013666 100375              BPL    .-4            ;THE SECOND HALF
2566 013670 117737 165512 001325  MOVB   @RXDBUF,DATA+1 ;GET THE SECOND HALF
2567 013676 005137 007156      COM    CALBCC          ;INVERT BCC
2568 013702 023737 007156 001324  CMP    CALBCC,DATA     ;CHECK IT
2569 013710 001401              BEQ    .+4            ;BR IF OK
2570 013712 104004              HLT    4              ;BCC COMPARE ERROR
2571 013714 012716 013602      MOV    #5$, (SP)      ;FINISH TEST
2572 013720 000002              RTI              ;RETURN
2573
2574              ;TRANSMITTER
2575 013722 012777 000377 165464  11$:   MOV    #377,@TXDBUF    ;LOAD A DATA CHARACTER
2576 013730 005202              INC    R2              ;INC THE # OF CHARS TO DO
2577 013732 022702 000005      CMP    #5,R2          ;CHECK TO SEE OF DONE
2578 013736 001017              BNE    13$           ;BR IF MORE TO GO
  
```



CONTIGUOUS ONES OUTPUT TEST

```

2579 013740 012777 013750 165432      MOV    #12$,@DUPTVC    ;SETUP NEXT VECTOR
2580 013746 000413                    BR     13$             ;RETURN
2581 013750 012777 001000 165436 12$:  MOV    #TEOM,@TXDBUF  ;END MSG
2582 013756 000240                    NOP                      ;WAIT
2583 013760 000240                    NOP                      ;DITTO
2584 013762 042777 000120 165422      BIC    #SEND!TXINTE,@TXCSR ;TURN OFF TRANSMITTER
2585 013770 012777 006774 165402      MOV    #NO.BTRAP,@DUPTVC ;RESET THE VECTOR
2586 013776 012716 013544 13$:      MOV    #4$, (SP)      ;GO BACK TO WAIT LOOP
2587 014002 000002                    RTI                     ;RETURN

```

```

;***** TEST 14 *****
;*THIS TEST PROVES THE INTERACTION OF DEC MODE,
;*TSOM, SYNC, TXACT, TXDONE
;*****

```

```

;*****
; TEST 14
;*****
;*****

```

```

2600 014004 012737 000014 001226 TST14: MOV    #14,@#TSTNO
2601 014012 012737 014156 001216      MOV    #TST15,NEXT
2602 014020 052777 000400 165364      BIS    #MRESET,@TXCSR  ;RESET THE DEVICE
2603 014026 004737 005044                    JSR    PC,SMALL        ;WAIT FOR RESET TO FINISH
2604 014032 012777 101026 165350      MOV    #DECMOD!26!CRCEN,@PARCSR
2605 014040 052777 004000 165344      BIS    #SYSTST,@TXCSR  ;ENTER SYSTEM TEST MODE
2606 014046 052777 000020 165336      BIS    #SEND,@TXCSR    ;TURN ON TRANSMITTER
2607 014054 012777 000426 165332      MOV    #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
2608 014062 012737 000005 014112      MOV    #5,68$         ;LOAD THE NUMBER
2609 014070 032777 004000 165316 66$:  BIT    #TIMER,@TXDBUF  ;CHECK THE TIMER BIT
2610 014076 001374                    BNE    66$             ;BR IF SET
2611 014100 032777 004000 165306 67$:  BIT    #TIMER,@TXDBUF  ;CHECK THE BIT
2612 014106 001774                    BEQ    67$             ;BR IF CLEAR
2613 014110 005327                    DEC    (PC)+           ;DECREMENT THE NUMBER
2614 014112 000005 68$:                    5                       ;OF TIMES TO REPEAT
2615 014114 001365                    BNE    66$             ;BR IF MORE TO GO
2616 014116 017704 165270      MOV    @TXCSR,R4       ;GET THE CSR
2617 014122 032704 000200      BIT    #TXDONE,R4     ;CHECK TRANSMITTER DONE
2618 014126 001001                    BNE    1$             ;BR IF SET
2619 014130 104016                    HLT    16              ;TXDONE FAILED TO SET
2620 014132 032704 001000 1$:      BIT    #TXACT,R4      ;TEST ACTIVE
2621 014136 001001                    BNE    2$             ;BR IF SET
2622 014140 104017                    HLT    17              ;ACTIVE FAILED TO SET
2623 014142 2$:
2624 014142 052777 000400 165242      BIS    #MRESET,@TXCSR  ;RESET THE DEVICE
2625 014150 004737 005044                    JSR    PC,SMALL        ;WAIT FOR RESET TO FINISH
2626 014154 104400                    SCOPE                  ;SCOPE THIS TEST

```

```

;***** TEST 15 *****
;*THIS TEST PROVES THE INTERACTION OF TEOM,
;*SEND, TXACT AND TXDONE IN DEC MODE.
;*****

```

```

;*****

```

2627  
2628  
2629  
2630  
2631  
2632  
2633  
2634

```
2635
2636
2637
2638
2639
2640 014156 012737 000015 001226 TST15: MOV #15,@#TSTNO
2641 014164 012737 014352 001216 MOV #TST16,NEXT
2642 014172 052777 000400 165212 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2643 014200 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2644 014204 012777 101026 165176 MOV #DECMOD!26!CRCEN,@PARCSR
2645 014212 052777 004000 165172 BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
2646 014220 052777 000020 165164 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
2647 014226 012777 000426 165160 MOV #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
2648 014234 105777 165152 1$: TSTB @TXCSR ;CHECK FOR DONE
2649 014240 100375 BPL 1$ ;BR IF NOT YET
2650 014242 012777 000426 165144 MOV #TSOM!26,@TXDBUF ;LOAD A SECOND SYNC
2651 014250 105777 165136 2$: TSTB @TXCSR ;AND NOW WAIT
2652 014254 100375 BPL 2$ ;FOR DONE AGAIN
2653 014256 012777 001000 165130 MOV #TEOM,@TXDBUF ;SET END OF MSG
2654 014264 042777 000020 165120 BIC #SEND,@TXCSR ;TURN OFF TRANSMITTER
2655 014272 012737 000025 014322 MOV #25,68$ ;LOAD THE NUMBER
2656 014300 032777 004000 165106 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2657 014306 001374 BNE 66$ ;BR IF SET
2658 014310 032777 004000 165076 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2659 014316 001774 BEQ 67$ ;BR IF CLEAR
2660 014320 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2661 014322 000025 68$: 25 ;OF TIMES TO REPEAT
2662 014324 001365 BNE 66$ ;BR IF MORE TO GO
2663 014326 105777 165060 TSTB @TXCSR ;CHECK DONE
2664 014332 100401 BMI 3$ ;BR IF SET
2665 014334 104016 HLT 16 ;DONE FAILED TO SET AFTER TURNING OFF TX.
2666 014336 032777 001000 165046 3$: BIT #TXACT,@TXCSR ;CHECK ACTIVE
2667 014344 001401 BEQ 4$ ;BR IF OFF
2668 014346 104020 HLT 20 ;ACTIVE IS STILL SET-SHOULD BE RESET
2669 014350 104400 4$: SCOPE ;SCOPE FOR THIS TEST.
```

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

```
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683 014352 012737 000016 001226 TST16: MOV #16,@#TSTNO
2684 014360 012737 014720 001216 MOV #TST17,NEXT
2685 014366 052777 000400 165016 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2686 014374 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2687 014400 012777 101026 165002 MOV #DECMOD!26!CRCEN,@PARCSR
2688 014406 052777 004000 164776 BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
2689 014414 052777 000020 164762 BIS #RCVEN,@RXCSR ;LOAD RCVEN
2690 014422 052777 000020 164762 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
```



```
2691 014430 012777 000426 164756      MOV    #TSOM!26,@TXDBUF      ;OUTPUT A SYNC CHAR
2692 014436 105777 164750      1$:   TSTB   @TXCSR              ;CHECK TRANSMITTER DONE
2693 014442 100375                BPL    1$                    ;WAIT TILL SET
2694 014444 012777 000125 164742      MOV    #125,@TXDBUF         ;LOAD DATA
2695 014452 012737 000005 014502      MOV    #5,68$              ;LOAD THE NUMBER
2696 014460 032777 004000 164726      66$:  BIT    #TIMER,@TXDBUF      ;CHECK THE TIMER BIT
2697 014466 001374                BNE    66$                  ;BR IF SET
2698 014470 032777 004000 164716      67$:  BIT    #TIMER,@TXDBUF      ;CHECK THE BIT
2699 014476 001774                BEQ    67$                  ;BR IF CLEAR
2700 014500 005327                DEC    (PC)+                ;DECREMENT THE NUMBER
2701 014502 000005                68$:  5                        ;OF TIMES TO REPEAT
2702 014504 001365                BNE    66$                  ;BR IF MORE TO GO
2703 014506 105777 164672      TSTB   @RXCSR              ;CHECK FOR RECEIVER DONE
2704 014512 100002                BPL    2$                    ;BR IF NOT SET
2705 014514 104021                HLT    21                    ;DEVICE SYNC'S UP IN LESS THAN 2 SYNC'S!!
2706 014516 000472                BR     5$                    ;LEAVE
2707 014520                2$:
2708 014520 052777 000400 164664      BIS    #MRESET,@TXCSR      ;RESET THE DEVICE
2709 014526 004737 005044                JSR    PC,SMALL             ;WAIT FOR RESET TO FINISH
2710 014532 012777 101026 164650      MOV    #CRCEN!DECMOD!26,@PARCSR ;LOAD THE MODE AND SYNC CHAR
2711 014540 052777 000020 164636      BIS    #RCVEN,@RXCSR       ;TURN ON RECEIVER
2712 014546 052777 004000 164636      BIS    #SYSTST,@TXCSR      ;ENTER SYSTEM TEST MODE
2713 014554 052777 000020 164630      BIS    #SEND,@TXCSR        ;TURN ON TRANSMITTER
2714 014562 012777 000426 164624      MOV    #TSOM!26,@TXDBUF    ;OUTPUT A SYNC CHAR
2715 014570 105777 164616      69$:  TSTB   @TXCSR              ;CHECK DONE
2716 014574 100375                BPL    69$                  ;BR IF NOT SET
2717 014576 012777 000426 164610      MOV    #TSOM!26,@TXDBUF    ;SEND SYNC
2718 014604 105777 164602      3$:   TSTB   @TXCSR              ;CHECK DONE
2719 014610 100375                BPL    3$                    ;WAIT
2720 014612 012777 000125 164574      MOV    #125,@TXDBUF        ;LOAD DATA
2721 014620 012737 000020 014650      MOV    #20,74$             ;LOAD THE NUMBER
2722 014626 032777 004000 164560      72$:  BIT    #TIMER,@TXDBUF      ;CHECK THE TIMER BIT
2723 014634 001374                BNE    72$                  ;BR IF SET
2724 014636 032777 004000 164550      73$:  BIT    #TIMER,@TXDBUF      ;CHECK THE BIT
2725 014644 001774                BEQ    73$                  ;BR IF CLEAR
2726 014646 005327                DEC    (PC)+                ;DECREMENT THE NUMBER
2727 014650 000020                74$:  20                       ;OF TIMES TO REPEAT
2728 014652 001365                BNE    72$                  ;BR IF MORE TO GO
2729 014654 105777 164524      TSTB   @RXCSR              ;CHECK FOR DONE
2730 014660 100401                BMI    4$                    ;BR IF SET
2731 014662 104022                HLT    22                    ;FAILED TO RECEIVE DATA
2732 014664 017737 164516 001236      4$:   MOV    @RXDBUF,TEMP1     ;READ DATA
2733 014672 122737 000125 001236      CMPB   #125,TEMP1          ;CHECK IT
2734 014700 001401                BEQ    5$                    ;BR IF MATCH
2735 014702 104022                HLT    22                    ;DATA COMPARE ERROR
2736 014704                5$:
2737 014704 052777 000400 164500      BIS    #MRESET,@TXCSR      ;RESET THE DEVICE
2738 014712 004737 005044                JSR    PC,SMALL             ;WAIT FOR RESET TO FINISH
2739 014716 104400                SCOPE                        ;SCOPE THIS TEST
2740
2741
2742
2743 ;***** TEST 17 *****
2744 ;*THIS TEST PROVES THE RECEIVER WILL STRIP THE FIRST
2745 ;*TWO SYNC'S AND WILL PRESENT ALL SUBSEQUENT SYNC'S.
2746 ;*****
```

THIRD AND SUBSEQUENT SYNC TEST

```
2747 ;:*****
2748 ;:
2749 ;: TEST 17
2750 ;:
2751 ;:*****
2752 ;:*****
2753 014720 012737 000017 001226 TST17: MOV #17,@#TSTNO
2754 014726 012737 015142 001216 MOV #TST20,NEXT
2755 014734 052777 000400 164450 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2756 014742 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2757 014746 012777 101026 164434 MOV #DECMOD!26!CRCEN,@PARCSR
2758 014754 052777 004000 164430 BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
2759 014762 052777 000020 164414 BIS #RCVEN,@RXCSR ;LOAD RCVEN
2760 014770 052777 000020 164414 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
2761 014776 012777 000426 164410 MOV #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
2762 015004 032777 004000 164402 64$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2763 015012 001374 BNE 64$ ;BR IF SET
2764 015014 032777 004000 164372 65$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2765 015022 001774 BEQ 65$ ;BR IF CLEAR
2766 015024 105777 164362 69$: TSTB @TXCSR ;CHECK DONE
2767 015030 100375 BPL 69$ ;BR IF NOT SET
2768 015032 012777 000426 164354 MOV #TSOM!26,@TXDBUF ;SEND SYNC
2769 015040 105777 164346 70$: TSTB @TXCSR ;CHECK DONE
2770 015044 100375 BPL 70$ ;BR IF NOT SET
2771 015046 012777 000426 164340 MOV #TSOM!26,@TXDBUF ;SEND SYNC
2772 015054 012737 000020 015104 MOV #20,75$ ;LOAD THE NUMBER
2773 015062 032777 004000 164324 73$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2774 015070 001374 BNE 73$ ;BR IF SET
2775 015072 032777 004000 164314 74$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2776 015100 001774 BEQ 74$ ;BR IF CLEAR
2777 015102 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2778 015104 000020 75$: 20 ;OF TIMES TO REPEAT
2779 015106 001365 BNE 73$ ;BR IF MORE TO GO
2780 015110 105777 164270 TSTB @RXCSR ;CHECK FOR DONE
2781 015114 100401 BMI 1$ ;BR IF SET
2782 015116 104021 HLT 21 ;DONE NOT SET-DEVICE FAILED TO SYNC UP
2783 015120 117737 164262 001236 1$: MOVB @RXDBUF,TEMP1 ;READ BUFFER
2784 015126 122737 000026 001236 CMPB #26,TEMP1 ;CHECK FOR SYNC
2785 015134 001401 BEQ 2$ ;BR IF OK
2786 015136 104022 HLT 22 ;DATA ERROR
2787 015140 104400 2$: SCOPE ;SCOPE THIS TEST
```

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

```
2795 ;:*****
2796 ;:
2797 ;: TEST 20
2798 ;:
2799 ;:*****
2800 ;:*****
2801 015142 012737 000020 001226 TST20: MOV #20,@#TSTNO
2802 015150 012737 015406 001216 MOV #TST21,NEXT
```



CONTINUOUS SYNC TEST

```
2803 015156 052777 000400 164226 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2804 015164 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2805 015170 012777 101026 164212 MOV #DECMOD!26!CRCEN,@PARCSR
2806 015176 052777 004000 164206 BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
2807 015204 052777 000020 164172 BIS #RCVEN,@RXCSR ;LOAD RCVEN
2808 015212 052777 000020 164172 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
2809 015220 012777 000426 164166 MOV #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
2810 015226 105777 164160 64$: TSTB @TXCSR ;CHECK DONE
2811 015232 100375 BPL 64$ ;BR IF NOT SET
2812 015234 012777 000426 164152 MOV #TSOM!26,@TXDBUF ;SEND SYNC
2813 015242 105777 164144 65$: TSTB @TXCSR ;CHECK DONE
2814 015246 100375 BPL 65$ ;BR IF NOT SET
2815 015250 012777 000426 164136 MOV #TSOM!26,@TXDBUF ;SEND SYNC
2816 015256 005037 001236 CLR TEMP1
2817 015262 005037 001240 CLR TEMP2
2818 015266 012737 000100 001236 MOV #64.,TEMP1 ;LOAD # OF SYNCs
2819 015274 012737 000010 015324 MOV #10,70$ ;LOAD THE NUMBER
2820 015302 032777 004000 164104 68$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2821 015310 001374 BNE 68$ ;BR IF SET
2822 015312 032777 004000 164074 69$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2823 015320 001774 BEQ 69$ ;BR IF CLEAR
2824 015322 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2825 015324 000010 70$: 10 ;OF TIMES TO REPEAT
2826 015326 001365 BNE 68$ ;BR IF MORE TO GO
2827 015330 105777 164056 1$: TSTB @TXCSR ;CHECK DONE
2828 015334 100401 BMI 2$ ;BR IF SET
2829 015336 104016 HLT 16 ;DONE FAILED TO SET
2830 015340 012777 000426 164046 2$: MOV #TSOM!26,@TXDBUF ;LOAD A SYNC
2831 015346 005337 001236 DEC TEMP1 ;LOWER THE # OF SYNCs TO DO
2832 015352 001001 BNE 4$ ;BR IF MORE TO GO
2833 015354 104400 3$: SCOPE ;SCOPE THIS TEST
2834
2835 015356 105777 164022 4$: TSTB @RXCSR ;CHECK RECEIVER DONE
2836 015362 100375 BPL 4$ ;WAIT TILL SET
2837 015364 017737 164016 001240 MOV @RXDBUF,TEMP2 ;GET THE BUFFER
2838 015372 122737 000026 001240 CMPB #26,TEMP2 ;CHECK IT FOR SYNC
2839 015400 001753 BEQ 1$ ;BR IF OK
2840 015402 104021 HLT 21 ;CHARACTER IS TEMP2 NOT A SYNC!
2841 015404 000763 BR 3$ ;LEAVE TEST
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
```

```
***** 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
MOV #TST22,NEXT
```

```
2857 015406 012737 000021 001226
2858 015414 012737 016126 001216
```

STRIP SYNC FUNCTION TEST

```

2859 015422 012737 000340 177776      MOV    #340,PS          ;RAISE STATUS
2860 015430 004537 006746              JSR    R5,SETVEC       ;SET UP VECTORS
2861 015434 015714                    5$          ;BASED ON
2862 015436 006774                    NO.BTRAP      ;THESE
2863 015440 340 340                    .BYTE 340,340      ;PARAMETERS
2864
2865 015442 052777 000400 163742      BIS    #MRESET,@TXCSR ;RESET THE DEVICE
2866 015450 004737 005044              JSR    PC,SMALL       ;WAIT FOR RESET TO FINISH
2867 015454 012777 101026 163726      MOV    #DECMOD!26!CRCEN,@PARCSR
2868 015462 052777 004000 163722      BIS    #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
2869 015470 052777 000420 163706      BIS    #RCVEN!STPSYN,@RXCSR ;LOAD RCVEN!STPSYN
2870 015476 052777 000020 163706      BIS    #SEND,@TXCSR   ;TURN ON TRANSMITTER
2871 015504 012777 000426 163702      MOV    #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
2872 015512 105777 163674 64$:      TSTB  @TXCSR          ;CHECK DONE
2873 015516 100375                    BPL    64$           ;BR IF NOT SET
2874 015520 012777 000426 163666      MOV    #TSOM!26,@TXDBUF ;SEND SYNC
2875 015526 105777 163660 65$:      TSTB  @TXCSR          ;CHECK DONE
2876 015532 100375                    BPL    65$           ;BR IF NOT SET
2877 015534 012777 000426 163652      MOV    #TSOM!26,@TXDBUF ;SEND SYNC
2878 015542 105777 163644 66$:      TSTB  @TXCSR          ;CHECK DONE
2879 015546 100375                    BPL    66$           ;BR IF NOT SET
2880 015550 012777 000426 163636      MOV    #TSOM!26,@TXDBUF ;SEND SYNC
2881 015556 005037 177776              CLR    PS            ;LOWER PS
2882 015562 052777 000100 163614      BIS    #RINTEN,@RXCSR ;TURN ON INTERRUPTS
2883 015570 105777 163616 1$:      TSTB  @TXCSR          ;CHECK TX DONE
2884 015574 100375                    BPL    1$            ;WAIT FOR SET
2885 015576 012777 000252 163610      MOV    #252,@TXDBUF   ;LOAD A CHARACTER
2886 015604 105777 163602 2$:      TSTB  @TXCSR          ;CHECK TX DONE
2887 015610 100375                    BPL    2$            ;WAIT TO BE SET
2888 015612 012777 000026 163574      MOV    #26,@TXDBUF   ;LOAD THE SYNC CHAR
2889 015620 105777 163566 3$:      TSTB  @TXCSR          ;CHECK DONE AGAIN
2890 015624 100375                    BPL    3$            ;WAIT
2891 015626 012777 000125 163560      MOV    #125,@TXDBUF  ;LOAD ANOTHER CHARACTER
2892 015634 105777 163552 4$:      TSTB  @TXCSR          ;CHECK DONE
2893 015640 100375                    BPL    4$            ;WAIT
2894 015642 012777 001000 163544      MOV    #TEOM,@TXDBUF ;SET END OF MESSAGE
2895 015650 042777 000020 163534      BIC    #SEND,@TXCSR   ;TURN OFF TRANSMITTER
2896 015656 012737 000050 015706      MOV    #40.,71$      ;LOAD THE NUMBER
2897 015664 032777 004000 163522 69$:  BIT    #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2898 015672 001374                    BNE    69$           ;BR IF SET
2899 015674 032777 004000 163512 70$:  BIT    #TIMER,@TXDBUF ;CHECK THE BIT
2900 015702 001774                    BEQ    70$           ;BR IF CLEAR
2901 015704 005327                    DEC    (PC)+         ;DECREMENT THE NUMBER
2902 015706 000050 71$:      40.                ;OF TIMES TO REPEAT
2903 015710 001365                    BNE    69$           ;BR IF MORE TO GO
2904 015712 104023                    HLT    23            ;FAILED TO TAKE A RECEIVER INTERRUPT
2905
2906 ;RECEIVER INTERRUPT SERVICE ROUTINE
2907 015714 017700 163464 5$:      MOV    @RXCSR,R0      ;READ CSR
2908 015720 017701 163462              MOV    @RXDBUF,R1     ;READ BUFFER
2909 015724 032700 000200              BIT    #RXDONE,R0     ;CHECK FOR DONE
2910 015730 001001                    BNE    6$            ;BR IF SET
2911 015732 104024                    HLT    24            ;RX DONE FAILED TO SET-ERRONEOUS INTERRUPT
2912 015734 032700 004000 6$:      BIT    #REACT,R0     ;CHECK FOR ACTIVE
2913 015740 001001                    BNE    7$            ;BR IF SET
2914 015742 104025                    HLT    25            ;RX ACTIVE FAILED TO SET

```



```
2915 015744 7$:  
2916 015744 005701 TST R1 ;CHECK FOR ERROR  
2917 015746 100001 BPL 10$ ;BR IF NO ERROR  
2918 015750 104026 HLT 26 ;RECEIVER ERROR  
2919 015752 122701 000252 10$: CMPB #252,R1 ;CHECK FOR CORRECT DATA  
2920 015756 001401 BEQ 11$ ;BR IF OK  
2921 015760 104022 HLT 22 ;DATA FAILED TO MATCH  
2922 015762 012777 015772 163404 11$: MOV #12$,@DUPRVC ;LOAD VECTOR  
2923 015770 000455 BR 20$ ;CONTINUE  
2924 015772 017700 163406 12$: MOV @RXCSR,R0 ;READ CSR  
2925 015776 017701 163404 MOV @RXDBUF,R1 ;READ BUFFER  
2926 016002 032700 000200 BIT #RXDONE,R0 ;CHECK FOR DONE  
2927 016006 001001 BNE 13$ ;BR IF OK  
2928 016010 104024 HLT 24 ;RX DONE FAILED TO SET-ERRONEOUS INTERRUPT  
2929 016012 005701 13$: TST R1 ;TEST FOR ERROR  
2930 016014 100001 BPL 14$ ;BR IF NO ERROR  
2931 016016 104026 HLT 26 ;ERROR SET  
2932 016020 122701 000026 14$: CMPB #26,R1 ;CHECK CHARACTER  
2933 016024 001422 BEQ 16$ ;BR IF OK-IF NOT, THEN  
2934 016026 122701 000125 CMPB #125,R1 ;CHECK FOR CLEARING SYNC  
2935 016032 001402 BEQ 15$ ;BR IF A NEXT CHARACTER  
2936 016034 104022 HLT 22 ;ERRONEOUS CHARACTER  
2937 016036 000415 BR 16$ ;BR TO END OF TEST  
2938 016040 104021 15$: HLT 21 ;STRIPPED OUT THE SYNC CHAR!!  
2939 016042 012777 016052 163324 MOV #21$,@DUPRVC ;SET UP VECTOR  
2940 016050 000425 BR 20$ ;LEAVE  
2941 016052 017700 163326 21$: MOV @RXCSR,R0 ;GET CSR  
2942 016056 017701 163324 MOV @RXDBUF,R1 ;GET BUFFER  
2943 016062 122701 000125 CMPB #125,R1 ;CHECK DATA  
2944 016066 001401 BEQ 16$ ;BR IF A MATCH  
2945 016070 104022 HLT 22 ;DATA COMPARE ERROR  
2946 016072 032777 004000 163304 16$: BIT #REACT,@RXCSR ;TEST ACTIVE  
2947 016100 001001 BNE 17$ ;BR IF ON  
2948 016102 104025 HLT 25 ;ACTIVE SHOULD BE ON  
2949 016104 17$:  
2950 016104 052777 000400 163300 BIS #MRESET,@TXCSR ;RESET THE DEVICE  
2951 016112 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH  
2952 016116 012706 001150 MOV #STACK,SP ;RESET STACK  
2953 016122 104400 SCOPE ;SCOPE THIS TEST  
2954 016124 000002 20$: RTI ;RETURN
```

```
2955  
2956  
2957 ;***** TEST 22 *****  
2958 ;*THIS TEST PROVES THAT A BINARY COUNT  
2959 ;*PATTERN CAN BE RUN IN DEC MODE  
2960 ;*WITHOUT A BCC CALCULATION  
2961 ;*****  
2962 ;:*****  
2963 ;: *  
2964 ;: TEST 22  
2965 ;: *  
2966 ;:*****  
2967 ;:*****  
2968 016126 012737 000022 001226 TST22: MOV #22,@TSTNO  
2969 016134 012737 016610 001216 MOV #TST23,NEXT  
2970 016142 012737 000340 177776 MOV #340,PS
```

```

2971 016150 005000          CLR      R0          ;CLR OUT DATA POINTER
2972 016152 005001          CLR      R1          ;DITTO
2973 016154 004537 006746  JSR      R5,SETVEC  ;SET UP INTERRUPTS
2974 016160 016370          4$      ;RECEIVER
2975 016162 016542          17$     ;TRANSMITTER
2976 016164      340      340  .BYTE   340,340     ;LEVEL
2977
2978 016166 052777 000400 163216  BIS     #MRESET,@TXCSR ;RESET THE DEVICE
2979 016174 004737 005044          JSR     PC,SMALL    ;WAIT FOR RESET TO FINISH
2980 016200 012777 101026 163202  MOV     #DECMOD!26!CRCEN,@PARCSR
2981 016206 052777 004000 163176  BIS     #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
2982 016214 052777 000020 163162  BIS     #RCVEN,@RXCSR  ;LOAD RCVEN
2983 016222 052777 000020 163162  BIS     #SEND,@TXCSR   ;TURN ON TRANSMITTER
2984 016230 012777 000426 163156  MOV     #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
2985 016236 032777 004000 163150 64$:   BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2986 016244 001374          BNE     64$         ;BR IF SET
2987 016246 032777 004000 163140 65$:   BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2988 016254 001774          BEQ     65$         ;BR IF CLEAR
2989 016256 105777 163130          69$:   TSTB   @TXCSR      ;CHECK DONE
2990 016262 100375          BPL     69$         ;BR IF NOT SET
2991 016264 012777 000426 163122  MOV     #TSOM!26,@TXDBUF ;SEND SYNC
2992 016272 005037 177776          CLR     PS
2993 016276 052777 000100 163100  BIS     #RINTEN,@RXCSR  ;TURN ON INT ENABLES
2994 016304 052777 000100 163100  BIS     #TXINTE,@TXCSR  ;DITTO
2995 016312          30$:
2996 016312 012737 000310 016342  MOV     #200.,74$      ;LOAD THE NUMBER
2997 016320 032777 004000 163066 72$:   BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2998 016326 001374          BNE     72$         ;BR IF SET
2999 016330 032777 004000 163056 73$:   BIT     #TIMER,@TXDBUF ;CHECK THE BIT
3000 016336 001774          BEQ     73$         ;BR IF CLEAR
3001 016340 005327          DEC     (PC)+        ;DECREMENT THE NUMBER
3002 016342 000310          74$:   200.         ;OF TIMES TO REPEAT
3003 016344 001365          BNE     72$         ;BR IF MORE TO GO
3004 016346 104023          HLT     23          ;FAILED TO FINISH TEST
3005 016350          3$:
3006 016350 052777 000400 163034  BIS     #MRESET,@TXCSR ;RESET THE DEVICE
3007 016356 004737 005044          JSR     PC,SMALL    ;WAIT FOR RESET TO FINISH
3008 016362 012706 001150          MOV     #STACK,SP   ;RESET THE STACK
3009 016366 104400          SCOPE              ;SCOPE THIS TEST
3010
3011
3012          ;RECEIVER INT SVC ROUTINE
3013 016370 017702 163010 4$:   MOV     @RXCSR,R2     ;SAVE CSR
3014 016374 017703 163006  MOV     @RXDBUF,R3   ;SAVE BUFFER
3015 016400 032702 004000  BIT     #REACT,R2    ;TEST RX ACTIVE
3016 016404 001004          BNE     5$          ;BR IF OK
3017 016406 104025          HLT     25          ;ACTIVE NOT SET
3018 016410 012716 016350  MOV     #3$(,SP)     ;SETUP FOR RETURN
3019 016414 000432          BR     12$
3020 016416 032702 000200 5$:   BIT     #RXDONE,R2  ;TEST DONE
3021 016422 001004          BNE     6$          ;BR IF OK
3022 016424 104024          HLT     24          ;FALSE INTERRUPT
3023 016426 012716 016350  MOV     #3$(,SP)     ;SETUP FOR RETURN
3024 016432 000423          BR     12$
3025 016434 005703 6$:   TST     R3          ;CHECK FOR ERROR
3026 016436 100004          BPL     7$          ;BR IF NO ERROR
  
```



```
3027 016440 104026          HLT      26          ;DATA ERROR
3028 016442 012716 016350  MOV      #3$, (SP)  ;SET UP RETURN
3029 016446 000415          BR       12$
3030 016450 120103          7$: CMPB   R1,R3      ;CHECK DATA
3031 016452 001404          BEQ     10$          ;BR IF OK
3032 016454 104022          HLT     22          ;BAD DATA
3033 016456 012716 016350  MOV      #3$, (SP)  ;SETUP RETURN
3034 016462 000407          BR       12$
3035 016464 005201          10$: INC    R1          ;UPDATE DATA
3036 016466 001002          BNE     11$          ;BR IF MORE TO GO
3037 016470 012716 016350  MOV      #3$, (SP)  ;SETUP RETURN
3038 016474 012777 016504 162672 11$: MOV    #22$,@DUPRVC ;SETUP NEW RETURN FOR INTERRUPT
3039 016502 000002          12$: RTI
3040 016504 017702 162674 22$: MOV    @RXCSR,R2
3041 016510 017703 162672  MOV    @RXDBUF,R3
3042 016514 005703          TST     R3
3043 016516 100001          BPL     23$
3044 016520 104026          HLT     26          ;ERROR
3045 016522 120103          23$: CMPB   R1,R3
3046 016524 001401          BEQ     24$
3047 016526 104022          HLT     22          ;DATA COMPARE ERROR
3048 016530 105201          24$: INCB   R1
3049 016532 001363          BNE     12$
3050 016534 012716 016350  MOV      #3$, (SP)
3051 016540 000760          BR       12$
3052
3053          ; TRANSMITTER
3054 016542 010077 162646 17$: MOV    R0,@TXDBUF  ;PUSH OUT DATA
3055 016546 105200          INCB   R0          ;UPDATE IT
3056 016550 001014          BNE     21$          ;BR IF MORE
3057 016552 105777 162634 20$: TSTB   @TXCSR      ;CHECK FOR NEXT DONE
3058 016556 100375          BPL     20$          ;WAIT
3059 016560 052777 001000 162626  BIS    #TEOM,@TXDBUF ;END MSG
3060 016566 042777 000120 162616  BIC    #SEND!TXINTE,@TXCSR ;SHUT OF TRANSMITTER
3061 016574 012777 006774 162576  MOV    #NO.BTRAP,@DUPTVC ;RESET VECTOR ADRS
3062 016602 012716 016312 21$: MOV    #30$, (SP)
3063 016606 000002          RTI          ;RETURN
3064
3065
3066
3067          ;***** TEST 23 *****
3068          ;*THIS TEST PROVES THAT A BINARY COUNT
3069          ;*PATTERN CAN BE RUN IN DEC MODE
3070          ;*WITH A BCC CALCULATION USING
3071          ;*THE CRC16 POLYNOMIAL
3072          ;*****
3073          ;*****
3074          ;*
3075          ;* TEST 23
3076          ;*
3077          ;*****
3078          ;*****
3079 016610 012737 000023 001226 TST23: MOV    #23,@#TSTNO
3080 016616 012737 017432 001216  MOV    #TST24,NEXT
3081 016624 012737 000340 177776  MOV    #340,PS
3082 016632 005000          CLR    R0
```





DATA TEST WITH BCC CHECK

3139	017134	104025				HLT	25		:ACTIVE NOT SET
3140	017136	012716	017076			MOV	#3\$, (SP)		:SETUP FOR RETURN
3141	017142	000433				BR	12\$		
3142	017144	032702	000200		5\$:	BIT	#RXDONE, R2		:TEST DONE
3143	017150	001004				BNE	6\$		:BR IF OK
3144	017152	104024				HLT	24		:FALSE INTERRUPT
3145	017154	012716	017076			MOV	#3\$, (SP)		:SETUP FOR RETURN
3146	017160	000424				BR	12\$		
3147	017162	005703			6\$:	TST	R3		:CHECK FOR ERROR
3148	017164	100004				BPL	7\$		:BR IF NO ERROR
3149	017166	104026				HLT	26		:DATA ERROR
3150	017170	012716	017076			MOV	#3\$, (SP)		:SET UP RETURN
3151	017174	000416				BR	12\$		
3152	017176	120103			7\$:	CMPB	R1, R3		:CHECK DATA
3153	017200	001404				BEQ	10\$		:BR IF OK
3154	017202	104022				HLT	22		:BAD DATA
3155	017204	012716	017076			MOV	#3\$, (SP)		:SETUP RETURN
3156	017210	000410				BR	12\$		
3157	017212	005201			10\$:	INC	R1		:UPDATE DATA
3158	017214	001003				BNE	11\$		:BR IF MORE TO GO
3159	017216	012716	017234			MOV	#13\$, (SP)		:SETUP TO FINISH TEST
3160	017222	000403				BR	12\$		
3161	017224	012777	017324	162142	11\$:	MOV	#22\$, @DUPRVC		:SETUP NEW RETURN FOR INTERRUPT
3162	017232	000002			12\$:	RTI			:RETURN
3163	017234	105777	162144		13\$:	TSTB	@RXCSR		:TEST DONE
3164	017240	100375				BPL	13\$		:WAIT
3165	017242	017737	162140	001236		MOV	@RXDBUF, TEMP1		:GET DATA
3166	017250	105777	162130		14\$:	TSTB	@RXCSR		:CHECK DONE FOR HALF OF CRC
3167	017254	100375				BPL	14\$		:WAIT
3168	017256	017737	162124	001240		MOV	@RXDBUF, TEMP2		:MOVE IT
3169	017264	113737	001240	001237		MOVB	TEMP2, TEMP1+1		:COMBINE BCC CHARACTER
3170	017272	023737	007156	001236		CMP	CALBCC, TEMP1		:BR IF A MATCH
3171	017300	001401				BEQ	15\$		:AFTER CHECKING IT
3172	017302	104027				HLT	27		:CRC COMPARE ERROR--THE
3173									:SOFTWARE DOESN'T AGREE
3174									:WITH WHAT THE TRANSMITTER
3175									:SENT. SEE THE FRONT OF
3176									:THE LISTING FOR SPECIAL
3177									:CRC DEBUG AID TEST.
3178	017304	032737	010000	001240	15\$:	BIT	#CRCERR, TEMP2		:CHECK FOR ERROR
3179	017312	001001				BNE	16\$		:BR IF OK
3180	017314	104030				HLT	30		:HARDWARE DETECTED CRC ERROR
3181									:RECEIVER DOESN'T AGREE WITH
3182									:WHAT THE TRANSMITTER SENT
3183									:SEE FRONT OF LISTING FOR
3184									:SPECIAL CRC DEBUG AID
3185	017316	012716	017076		16\$:	MOV	#3\$, (SP)		:LOAD END OF TEST
3186	017322	000743				BR	12\$		:RETURN
3187	017324	017702	162054		22\$:	MOV	@RXCSR, R2		
3188	017330	017703	162052			MOV	@RXDBUF, R3		
3189	017334	005703				TST	R3		
3190	017336	100001				BPL	23\$		
3191	017340	104026				HLT	26		:ERROR
3192	017342	120103			23\$:	CMPB	R1, R3		
3193	017344	001401				BEQ	24\$		
3194	017346	104022				HLT	22		:DATA COMPARE ERROR

DATA TEST WITH BCC CHECK

```
3195 017350 105201          24$:  INCB  R1
3196 017352 001327          BNE   12$
3197 017354 012777 017234 162012  MOV   #13$,@DUPRVC
3198 017362 000723          BR    12$
3199
3200          ;TRANSMITTER
3201 017364 010077 162024 17$:  MOV   RO,@TXDBUF      ;PUSH OUT DATA
3202 017370 105200          INCB  RO              ;UPDATE IT
3203 017372 001014          BNE   21$            ;BR IF MORE
3204 017374 105777 162012 20$:  TSTB  @TXCSR         ;CHECK FOR NEXT DONE
3205 017400 100375          BPL   20$            ;WAIT
3206 017402 052777 001000 162004  BIS   #TEOM,@TXDBUF   ;END MSG
3207 017410 042777 000120 161774  BIC   #SEND!TXINTE,@TXCSR ;SHUT OF TRANSMITTER
3208 017416 012777 006774 161754  MOV   #NO.BTRAP,@DUPTVC ;RESET VECTOR ADRS
3209 017424 012716 017040 21$:  MOV   #30$, (SP)
3210 017430 000002          RTI                   ;RETURN
3211
3212
3213
3214
3215          ;***** TEST 24 *****
3216          ;*TEST TO PROVE THE DEVICE IDLES SYNCs AND
3217          ;*WILL SHIFT OUT DATA AT THE APPROPRIATE TIME
3218          ;:*****
3219
3220          ;:*****
3221          ;:
3222          ; TEST 24
3223          ;:
3224          ;:*****
3225          ;:*****
3226 017432 012737 000024 001226 TST24: MOV   #24,@TSTNO
3227 017440 012737 017630 001216  MOV   #TST25,NEXT
3228 017446 052777 000400 161736  BIS   #MRESET,@TXCSR ;RESET THE DEVICE
3229 017454 004737 005044          JSR   PC,SMALL       ;WAIT FOR RESET TO FINISH
3230 017460 052777 014000 161724  BIS   #MMODE,@TXCSR  ;ENTER MAINT MODE
3231 017466 012777 000020 161710  MOV   #RCVEN,@RXCSR  ;TURN ON RECEIVER
3232 017474 012777 100026 161706  MOV   #DECMOD!26,@PARCSR ;ENTER DECMODE AND SYNC CHAR
3233 017502 052777 000020 161702  BIS   #SEND,@TXCSR   ;TURN ON TRANSMITTER
3234 017510 012777 000426 161676  MOV   #TSOM!26,@TXDBUF ;PUSH OUT SYNCs
3235 017516 104412 000044          PKCLK ,36.
3236 017522 012777 000252 161664  MOV   #252,@TXDBUF   ;LOAD DATA
3237 017530 104412 000024          PKCLK ,20.           ;PUSH OUT ANOTHER SYNC
3238 017534 105777 161644          TSTB  @RXCSR         ;CHECK TO SEE IF SYNC ARRIVED
3239 017540 100401          BMI   1$             ;BR IF YES
3240 017542 104021          HLT   21
3241 017544 017737 161636 001324 1$:  MOV   @RXDBUF,DATA   ;GET THE REC CHAR
3242 017552 122737 000026 001324  CMPB  #26,DATA       ;CHECK FOR SYNC
3243 017560 001401          BEQ   2$             ;BR IF MATCH
3244 017562 104021          HLT   21            ;FAILED TO RECEIVE THIRD SYNC
3245 017564 042777 000020 161620 2$:  BIC   #SEND,@TXCSR   ;TURN OFF TRANSMITTER
3246 017572 104412 000016          PKCLK ,14.           ;PUSH OUT DATA
3247 017576 105777 161602          TSTB  @RXCSR         ;CHECK FOR REC DATA
3248 017602 100401          BMI   3$             ;BR IF YES
3249 017604 104026          HLT   26            ;FAILED TO GET A DATA DONE
3250 017606 017737 161574 001324 3$:  MOV   @RXDBUF,DATA   ;GET THE DATA
```





TEST OF ALL POSSIBLE SYNC CHARACTERS

```

3307 020060 001401          BEQ      6$          ;BR IF OK
3308 020062 104022          HLT      22          ;SYNC CHAR DOES NOT MATCH SENT
3309 020064                6$:
3310 020064 012702 000003    MOV      #3,R2       ;SET UP FOR NEXT SYNC
3311 020070 005000          CLR      R0          ;DITTO
3312 020072 105201          INCB    R1           ;DITTO
3313 020074 110137 001236    MOVB    R1,TEMP1
3314 020100 110137 001240    MOVB    R1,TEMP2
3315 020104 001300          BNE     1$          ;BR IF MORE TO GO
3316 020106 104400          SCOPE                ;SCOPE THIS TEST

```

```

3317
3318
3319
3320 ;***** TEST 26 *****
3321 ;*THIS TEST PROVES THAT THE CRC ERROR BIT FUNCTIONS
3322 ;*CORRECTLY. FORCE AN ERROR AND VERIFY THE BIT.
3323 ;*****

```

```

3324 ;:*****
3325 ;:
3326 ;: TEST 26
3327 ;:
3328 ;:*****
3329 ;:*****

```

```

3330 020110 012737 000026 001226 TST26: MOV      #26,@#TSTNO
3331 020116 012737 020450 001216    MOV      #TST27,NEXT
3332 020124 012737 000340 177776    MOV      #340,PS      ;RAISE PROCESSOR STATUS
3333 020132 004537 006746          JSR      R5,SETVEC    ;SETUP VECTORS
3334 020136 020412          6$          ;RECEIVER
3335 020140 020344          3$          ;TRANSMITTER
3336 020142      340      340    .BYTE    340,340     ;LEVEL
3337 020144 005001          CLR      R1          ;CLEAR CHAR COUNT
3338
3339 020146 052777 000400 161236    BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3340 020154 004737 005044          JSR      PC,SMALL    ;WAIT FOR RESET TO FINISH
3341 020160 012777 100026 161222    MOV      #DECMOD!26,@PARCSR ;LOAD THE MODE AND SYNC CHARACTER
3342 020166 052777 004000 161216    BIS      #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
3343 020174 052777 000020 161202    BIS      #RCVEN,@RXCSR ;LOAD RCVEN
3344 020202 052777 000020 161202    BIS      #SEND,@TXCSR ;TURN ON TRANSMITTER
3345 020210 012777 000426 161176    MOV      #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
3346 020216 105777 161170          64$: TSTB    @TXCSR      ;CHECK DONE
3347 020222 100375          BPL     64$          ;BR IF NOT SET
3348 020224 012777 000426 161162    MOV      #TSOM!26,@TXDBUF ;SEND SYNC
3349 020232 105777 161154          65$: TSTB    @TXCSR      ;CHECK DONE
3350 020236 100375          BPL     65$          ;BR IF NOT SET
3351 020240 012777 000426 161146    MOV      #TSOM!26,@TXDBUF ;SEND SYNC
3352 020246 005037 177776          CLR     PS           ;LOWER PROCESSOR STSUS
3353 020252 052777 000100 161124    BIS      #RINTEN,@RXCSR ;TURN ON INTERRUPT ENABLES
3354 020260 052777 000100 161124    BIS      #TXINTE,@TXCSR ;DITTO
3355 020266
3356 020266 012737 000040 020316          1$: MOV      #32,,70$    ;LOAD THE NUMBER
3357 020274 032777 004000 161112          68$: BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3358 020302 001374          BNE     68$          ;BR IF SET
3359 020304 032777 004000 161102          69$: BIT     #TIMER,@TXDBUF ;CHECK THE BIT
3360 020312 001774          BEQ     69$          ;BR IF CLEAR
3361 020314 005327          DEC     (PC)+        ;DECREMENT THE NUMBER
3362 020316 000040          70$:          ;OF TIMES TO REPEAT

```



```
3363 020320 001365          BNE      68$          ;BR IF MORE TO GO
3364 020322 104023          HLT      23          ;FAILED TO FINISH TEST
3365 020324                2$:
3366 020324 052777 000400 161060  BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3367 020332 004737 005044          JSR      PC,SMALL    ;WAIT FOR RESET TO FINISH
3368 020336 012706 001150          MOV      #STACK,SP   ;RESET THE STACK
3369 020342 104400          SCOPE
3370
3371
3372                          ;INTERRUPT SERVICE ROUTINES
3373                          ;TRANSMITTER
3374 020344 005000          3$:  CLR      R0          ;CLEAR DATA
3375 020346 010077 161042          MOV      R0,@TXDBUF  ;LOAD DATA TO BUFFER
3376 020352 012777 020362 161020  MOV      #4$,@DUPTVC ;SETUP FOR NEXT INTERRUPT
3377 020360 000411          BR       5$          ;LEAVE
3378 020362 012777 001000 161024  4$:  MOV      #TEOM,@TXDBUF ;END OF MSG--OUTPUT CRC
3379 020370 042777 000120 161014  BIC      #SEND!TXINTE,@TXCSR ;TRUN OFF THE
3380 020376 012777 006774 160774  MOV      #NO.BTRAP,@DUPTVC ;TRANSMITTER AND TXINTEN
3381 020404 012716 020266          5$:  MOV      #1$, (SP)    ;SETUP TO RETURN
3382 020410 000002          RTI          ;RETURN
3383
3384                          ;RECEIVER
3385 020412 017737 160770 001324  6$:  MOV      @RXDBUF,DATA ;GET THE DATA
3386 020420 005201          INC      R1          ;CHECK FOR LAST CHAR
3387 020422 022701 000004          CMP      #4,R1      ;AND BRANCH IF
3388 020426 001007          BNE     10$          ;NOT YET
3389 020430 032737 010000 001324  BIT      #CRCERR,DATA ;CHECK FOR CRC ERROR
3390 020436 001401          BEQ     7$          ;BR IF CRC ERROR SEEN
3391 020440 104014          HLT     14          ;FAILED TO CATCH CRC ERROR!!!!
3392 020442 012716 020324          7$:  MOV      #2$, (SP)    ;FINISH TEST
3393 020446 000002          10$: RTI          ;RETURN
3394
3395
3396                          ;***** TEST 27 *****
3397                          ;*THIS TEST PROVES THE DEVICE WILL HANDLE THE
3398                          ;*DDCMP PROTOCOL. SEND AND RECEIVE SYNCs,
3399                          ;*FOLLOWED BY DATA,BCC,DATA AND FINAL BCC.
3400                          ;*****
3401
3402                          ;*****
3403                          ; *
3404                          ; TEST 27
3405                          ; *
3406                          ;*****
3407                          ;*****
3408 020450 012737 000027 001226  TST27: MOV      #27,@#TSTNO
3409 020456 012737 021454 001216  MOV      #TST30,NEXT
3410 020464 012737 000340 177776  MOV      #340,PS
3411 020472 004537 006746          JSR      R5,SETVEC   ;RAISE PROCESSOR STATUS
3412 020476 021104          10$:          ;SET UP VECTORS
3413 020500 020726          2$:          ;BASED ON
3414 020502 340 340          .BYTE 340,340      ;THESE
3415 020504 005037 001236          CLR     TEMP1        ;PARAMETERS
3416 020510 005037 001240          CLR     TEMP2
3417 020514 005037 001242          CLR     TEMP3
3418 020520 005037 001244          CLR     TEMP4
```

```

3419 020524 005037 001246 CLR TEMP5
3420
3421 020530 052777 000400 160654 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3422 020536 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3423 020542 012777 100026 160640 MOV #DECMOD!26,@PARCSR ;LOAD THE MODE AND SYNC CHARACTER
3424 020550 052777 004000 160634 BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
3425 020556 052777 000420 160620 BIS #RCVEN!STPSYN,@RXCSR ;LOAD RCVEN!STPSYN
3426 020564 052777 000020 160620 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
3427 020572 012777 000426 160614 MOV #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
3428 020600 105777 160606 64$: TSTB @TXCSR ;CHECK DONE
3429 020604 100375 BPL 64$ ;BR IF NOT SET
3430 020606 012777 000426 160600 MOV #TSOM!26,@TXDBUF ;SEND SYNC
3431 020614 105777 160572 65$: TSTB @TXCSR ;CHECK DONE
3432 020620 100375 BPL 65$ ;BR IF NOT SET
3433 020622 012777 000426 160564 MOV #TSOM!26,@TXDBUF ;SEND SYNC
3434 020630 052777 000100 160546 BIS #RINTEN,@RXCSR ;TURN ON INTERRUPTS
3435 020636 052777 000100 160546 BIS #TXINTE,@TXCSR ;DITTO
3436 020644 005037 177776 CLR PS ;LOWER PROCESSOR STATUS
3437 020650 100$:
3438 020650 012737 000144 020700 MOV #100.,70$ ;LOAD THE NUMBER
3439 020656 032777 004000 160530 68$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3440 020664 001374 BNE 68$ ;BR IF SET
3441 020666 032777 004000 160520 69$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3442 020674 001774 BEQ 69$ ;BR IF CLEAR
3443 020676 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3444 020700 000144 70$: 100. ;OF TIMES TO REPEAT
3445 020702 001365 BNE 68$ ;BR IF MORE TO GO
3446 020704 104023 HLT 23 ;FAILED TO FINISH TEST
3447 020706 1$:
3448 020706 052777 000400 160476 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3449 020714 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3450 020720 012706 001150 MOV #STACK,SP ;RESET THE STACK
3451 020724 104400 SCOPE ;SCOPE THIS TEST
3452
3453 ;INTERRUPT SERVICE ROUTINES
3454 ;TRANSMITTER
3455
3456 020726 012777 000252 160460 2$: MOV #252,@TXDBUF ;LOAD FIRST DATA CHAR
3457 020734 012737 000026 001236 MOV #26,TEMP1 ;LOAD DATA
3458 020742 012777 020752 160430 MOV #3$,@DUPTVC ;RELOAD VECTOR
3459 020750 000452 BR 7$ ;LEAVE
3460 020752 013777 001236 160434 3$: MOV TEMP1,@TXDBUF ;MOV DATA TO BUFFER
3461 020760 105237 001236 INCB TEMP1 ;UPDATE DATA
3462 020764 122737 000032 001236 CMPB #32,TEMP1 ;CHECK FOR DONE
3463 020772 001041 BNE 7$ ;BR IF MORE TO SEND
3464 020774 012777 021004 160376 MOV #4$,@DUPTVC ;RELOAD VECTOR
3465 021002 000435 BR 7$ ;RETURN
3466 021004 012777 001000 160402 4$: MOV #TEOM,@TXDBUF ;PUT OUT BCC
3467 021012 012777 021022 160360 MOV #5$,@DUPTVC ;RELOAD VECTOR
3468 021020 000426 BR 7$ ;RETURN
3469 021022 013777 001240 160364 5$: MOV TEMP2,@TXDBUF ;LOAD DATA
3470 021030 105237 001240 INCB TEMP2 ;UPDATE DATA
3471 021034 122737 000100 001240 CMPB #100,TEMP2 ;CHECK FOR FINISH
3472 021042 001015 BNE 7$ ;BR IF MORE TO GO
3473 021044 012777 021054 160326 MOV #6$,@DUPTVC ;RELOAD VECTOR
3474 021052 000411 BR 7$ ;RETURN

```



```

3475 021054 012777 001000 160332 6$:  MOV    #TEOM,@TXDBUF    ;PUSH OUT DATA BCC
3476 021062 042777 000120 160322    BIC    #SEND!TXINTE,@TXCSR    ;SHUT DOWN TRANSMITTER
3477 021070 012777 006774 160302    MOV    #NO.BTRAP,@DUPTVC    ;RESET VECTOR
3478 021076 012716 020650    7$:  MOV    #100$,(SP)    ;SETUP RETURN
3479 021102 000002    RTI    ;RETURN
3480
3481    ;RECEIVER
3482
3483 021104 017737 160274 001242 10$:  MOV    @RXCSR,TEMP3    ;SAVE CSR
3484 021112 017737 160270 001244    MOV    @RXDBUF,TEMP4    ;SAVE BUFFER
3485 021120 105737 001242    TSTB   TEMP3    ;CHECK FOR DONE
3486 021124 100401    BMI    11$    ;BR IF SET
3487 021126 104024    HLT    24    ;FALSE INTERRUPT
3488 021130 005737 001244    11$:  TST    TEMP4    ;CHECK FOR ERROR
3489 021134 100001    BPL    12$    ;BR IF NO ERROR
3490 021136 104026    HLT    26    ;RECEIVER ERROR
3491 021140 122737 000252 001244 12$:  CMPB   #252,TEMP4    ;CHECK DATA
3492 021146 001401    BEQ    13$    ;BR IF A MATCH
3493 021150 104022    HLT    22    ;DATA COMPARE ERROR
3494 021152 012737 000026 001246 13$:  MOV    #26,TEMP5    ;LOAD NEXT EXPECTED
3495 021160 012777 021170 160206    MOV    #14$,@DUPRVC    ;RELOAD VECTOR
3496 021166 000531    BR     26$    ;LEAVE
3497 021170 017737 160212 001244 14$:  MOV    @RXDBUF,TEMP4    ;GET DATA
3498 021176 005737 001244    TST    TEMP4    ;CHECK FOR ERROR
3499 021202 100001    BPL    15$    ;BR IF NO ERROR
3500 021204 104026    HLT    26    ;DATA ERROR
3501 021206 123737 001246 001244 15$:  CMPB   TEMP5,TEMP4    ;CHECK DATA
3502 021214 001401    BEQ    16$    ;BR IF A MATCH
3503 021216 104022    HLT    22    ;DATA COMPARE ERROR
3504 021220 105237 001246 001246 16$:  INCB   TEMP5    ;UPDATE DATA
3505 021224 122737 000032 001246    CMPB   #32,TEMP5    ;CHECK FOR FIRST PART FINISH
3506 021232 001107    BNE    26$    ;BR IF MORE TO GO
3507 021234 012777 021244 160132    MOV    #17$,@DUPRVC    ;SET UP NEXT VECTOR
3508 021242 000503    BR     26$    ;LEAVE
3509 021244 017737 160136 001244 17$:  MOV    @RXDBUF,TEMP4    ;GET THE BUFFER
3510 021252 005737 001244    TST    TEMP4    ;TEST FOR ERROR
3511 021256 100001    BPL    .+4    ;BR IF OK
3512 021260 104026    HLT    26    ;RECEIVER ERROR
3513 021262 012777 021272 160104    MOV    #18$,@DUPRVC    ;RELOAD THE VECTOR
3514 021270 000470    BR     26$    ;LEAVE
3515 021272 017737 160110 001324 18$:  MOV    @RXDBUF,DATA    ;GET DATA
3516 021300 032737 010000 001324    BIT    #CRCERR,DATA    ;CHECK FOR CRC ERROR
3517 021306 001001    BNE    19$    ;BR IF OK
3518 021310 104014    HLT    14    ;CRC ERROR!!!!!!
3519 021312 012777 021326 160054 19$:  MOV    #20$,@DUPRVC    ;SET UP VECTOR
3520 021320 005037 001330    CLR    MIND    ;SETUP FOR NEXT DATA
3521 021324 000452    BR     26$    ;LEAVE
3522 021326 017737 160054 001244 20$:  MOV    @RXDBUF,TEMP4    ;GET DATA
3523 021334 005737 001244    TST    TEMP4    ;CHECK FOR ERROR
3524 021340 100001    BPL    21$    ;BR IF NO ERROR
3525 021342 104026    HLT    26    ;RECEIVER ERROR
3526 021344 123737 001330 001244 21$:  CMPB   MIND,TEMP4    ;CHECK DATA
3527 021352 001401    BEQ    22$    ;BR IF A MATCH
3528 021354 104022    HLT    22    ;DATA ERROR
3529 021356 105237 001330 001330 22$:  INCB   MIND    ;UPDATE SOFTWARE DATA
3530 021362 122737 000100 001330    CMPB   #100,MIND    ;CHECK FOR FINISH

```

```
3531 021370 001030          BNE      26$          ;BR IF MORE TO GO
3532 021372 012777 021402 157774  MOV     #23$,@DUPRVC ;RELOAD FINAL VECTOR
3533 021400 000424          BR       26$          ;LEAVE
3534 021402 017737 160000 001244 23$:  MOV     @RXDBUF,TEMP4 ;GET DATA
3535 021410 005737 001244          TST     TEMP4         ;CHECK FOR ERROR
3536 021414 100001          BPL     24$          ;BR IF OK
3537 021416 104026          HLT     26           ;RECEIVER ERROR ON FIRST OCTET
3538                                ;OF SECOND BCC
3539 021420 105777 157760          24$:  TSTB   @RXCSR         ;TEST DONE
3540 021424 100375          BPL     24$          ;BR IF NOT SET
3541 021426 017737 157754 001324  MOV     @RXDBUF,DATA ;GET SECOND BCC OCTET
3542 021434 032737 010000 001324  BIT     #CRCERR,DATA ;CHECK FOR BCC ERROR
3543 021442 001001          BNE     25$          ;BR IF OK
3544 021444 104014          HLT     14           ;BCC ERROR ON SECOND PART OF MSG
3545 021446 012716 020706          25$:  MOV     #1$, (SP)   ;SETUP TO FINISH TEST
3546 021452 000002          26$:  RTI              ;RETURN
```

3547  
3548  
3549  
3550  
3551  
3552  
3553  
3554  
3555  
3556  
3557  
3558  
3559  
3560  
3561  
3562  
3563  
3564  
3565

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

```
*****  
TST30: MOV     #30,@#TSTNO  
MOV     #.EOP,NEXT  
BIS     #MRESET,@TXCSR ;RESET THE DEVICE  
JSR     PC,SMALL      ;WAIT FOR RESET TO FINISH  
MOV     #CRC16,XPOLY  ;LOAD THE POLYNOMIAL  
MOV     #125,3$       ;LOAD DATA TO SOFTWARE BCC-CHANGE CHARACTER HERE  
MOV     3$,SAVR1  
CLR     CALBCC        ;CLEAR FOR SOFTWARE BCC  
MOV     CALBCC,4$  
CLR     TEMP3  
CLR     TEMP4        ;CLEAR BIT COUNTER  
CLR     TEMP5  
MOV     #DECMOD!26,@PARCSR ;LOAD MODE AND SYNC CHARACTER  
BIS     #MMODE,@TXCSR  ;ENTER MAINT MODE-PROGRAM CLOCKING  
BIS     #RCVEN!STPSYN,@RXCSR ;TURN ON RECEIVER  
BIS     #SEND,@TXCSR   ;TURN ON TRANSMITTER  
MOV     #TSOM!26,@TXDBUF ;LOAD A SYNC  
PKCLK   ,36.         ;PUSH OUT 2 SYNCs  
MOV     3$,@TXDBUF    ;LOAD DATA  
PKCLK   ,16.         ;PUSH OUT ANOTHER SYNC  
1$:    PKCLK   ,2     ;PUSH OUT A BIT
```



```

3587 021632 013737 001244 001254      MOV    TEMP4,SAVR2    ;SET UP TO TYPE
3588 021640 005237 001242              INC    TEMP3
3589 021644 005237 001244              INC    TEMP4          ;UPDATE BIT COUNTER
3590 021650 004537 007000      2$:   JSR    R5,SIMBCC    ;CALCULATE SOFTWARE BCC BASED ON THESE PARAMETERS
3591 021654 000001              1      ;SHIFTS
3592 021656 000000      3$:   .WORD 0          ;DATA
3593 021660 000000      4$:   .WORD 0          ;PREVIOUS BCC
3594 021662 004737 021760      JSR    PC,5$          ;CHECK TO SEE IF WE SHOULD WAIT FOR SCOPING
3595 021666 000241              CLC                    ;CLEAR FOR NEXT ROTATE
3596 021670 106037 021656      RORB   3$             ;SET UP THE NEXT BIT
3597 021674 013737 007156 021660      MOV    CALBCC,4$      ;FOR THE SOFTWARE BCC
3598 021702 022737 000006 001244      CMP    #6,TEMP4
3599 021710 001002              BNE    .+6
3600 021712 005077 157476      CLR    @TXDBUF
3601 021716 022737 000014 001242      CMP    #12.,TEMP3
3602 021724 001003              BNE    12$
3603 021726 012777 001000 157460      MOV    #TEOM,@TXDBUF
3604 021734 022737 000020 001244      12$:  CMP    #16.,TEMP4    ;ALL DONE WITH THE CHARACTER?
3605                                ;INCREASE COMPARE # TO FORCE
3606                                ;CRC OUT OF THE GENERATOR
3607 021742 001331              BNE    1$             ;BR IF MORE TO GO
3608 021744 052777 000400 157440      BIS    #MRESET,@TXCSR ;RESET THE DEVICE
3609 021752 004737 005044              JSR    PC,SMALL      ;WAIT FOR RESET TO FINISH
3610 021756 104400              SCOPE                 ;SCOPE THIS TEST
3611
3612 021760 032777 001000 157214      5$:   BIT    #SW09,@SWR    ;SW09=1?
3613 021766 001432              BEQ    6$             ;BR IF NO
3614 021770 013704 007156      MOV    CALBCC,R4      ;THE DATA CHARACTER IS ALWAYS
3615 021774 012737 000001 001256      MOV    #1,SAVR3      ;FOLLOWED BY A ZERO. THE DATA IN
3616 022002 000241              CLC                    ;CRC SHOWS WHICH BIT OF THE 2 CHARS
3617 022004 006004      11$:  ROR    R4             ;IS BEING GENERATED
3618 022006 006137 001256      ROL    SAVR3
3619 022012 103374              BCC    11$
3620 022014 105737 001246      TSTB   TEMP5
3621 022020 001006              BNE    10$
3622 022022 104402 023152      TYPE   ,EM17          ;TYPE MSG
3623 022026 104402 023201      TYPE   ,MH1           ;TYPE HEADER
3624 022032 105137 001246      COMB   TEMP5
3625 022036 104410      10$:  CONVRT
3626 022040 023522      DT1
3627 022042 105777 157136      7$:   TSTB   @TKCSR      ;CHECK TTY DONE--GO SCOPE THE CRC GENERATOR
3628 022046 100375              BPL    7$             ;BR IF NOT YET
3629 022050 017701 157132      MOV    @TKDBR,R1     ;READ THE BUFFER
3630 022054 000207      6$:   RTS    PC          ;RETURN
3631
3632

```

```

3633
(1) 022056 042377 050125 043040 EM1: .ASCIZ <377>/DUP FAILED TO INTERRUPT IN TIME /
(1) 022120 042377 052101 020101 EM2: .ASCIZ <377>/DATA COMPARE ERROR /
(1) 022145 377 047105 020104 EM3: .ASCIZ <377>/END OF MESSAGE /
(1) 022166 041377 041503 042440 EM4: .ASCIZ <377>/BCC ERROR /
(1) 022202 052377 040522 051516 EM5: .ASCIZ <377>/TRANSMITTER DONE /
(1) 022225 106 044501 042514 EM6: .ASCIZ /FAILED TO SET /
(1) 022244 051377 041505 044505 EM7: .ASCIZ <377>/RECEIVER INTERRUPT IN HALF-DUPLEX /
(1) 022310 043377 046101 042523 EM10: .ASCIZ <377>/FALSE INTERRUPT /
(1) 022332 040777 047502 052122 EM11: .ASCIZ <377>/ABORT SEQUFNCE ERROR /
(1) 022361 377 052123 051101 EM12: .ASCIZ <377>/START OF MESSAGE /
(1) 022404 052777 042516 050130 EM13: .ASCIZ <377>/UNEXPECTED RECEIVER INTERRUPT /
(1) 022444 052777 042516 050130 EM14: .ASCIZ <377>/UNEXPECTED TRANSMITTER INTERRUPT /
(1) 022507 377 051124 047101 EM20: .ASCIZ <377>/TRANSMITTER DONE /
(1) 022532 051377 041505 044505 EM21: .ASCIZ <377>/RECEIVER DONE /
(1) 022552 052377 040522 051516 EM22: .ASCIZ <377>/TRANSMITTER ACTIVE /
(1) 022577 377 042522 042503 EM23: .ASCIZ <377>/RECEIVER ACTIVE /
(1) 022621 106 044501 042514 EM24: .ASCIZ /FAILED TO SET. /
(1) 022641 106 044501 042514 EM25: .ASCIZ /FAILED TO CLEAR. /
(1) 022663 377 054523 041516 EM26: .ASCIZ <377>/SYNC ERROR /
(1) 022700 042377 052101 020101 EM27: .ASCIZ <377>/DATA ERROR /
(1) 022715 377 042504 044526 EM30: .ASCIZ <377>/DEVICE FAILED TO INTERRUPT IN TIME /
(1) 022762 043377 046101 042523 EM31: .ASCIZ <377>/FALSE INTERRUPT /
(1) 023004 052377 040522 051516 EM32: .ASCIZ <377>/TRANSMITTER BCC ERROR IN DEC MODE /
(1) 023051 377 042522 042503 EM33: .ASCIZ <377>/RECEIVER BCC ERROR IN DEC MODE /
(1) 023113 377 042522 042503 EM15: .ASCIZ <377>/RECEIVER ERROR /
(1) 023134 041377 041503 042440 EM16: .ASCIZ <377>/BCC ERROR!! /
(1) 023152 041777 041522 043440 EM17: .ASCIZ <377>/CRC GENERATOR STATUS /
(1) 023201 377 040504 040524 MH1: .ASCIZ <377>/DATA CHAR DATA BIT IN CRC GEN. CRC FOR THIS BIT /
(1) .EVEN
(1)
(1) 023274 .ERRTAB:
(1) 023274 000000 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)	023346	022310	EM10		
(1)	023350	000000	0		;HALT 7
(1)	023352	000000	0		
(1)	023354	022332	EM11		
(1)	023356	000000	0		;HALT 10
(1)	023360	000000	0		
(1)	023362	022361	EM12		
(1)	023364	022225	EM6		;HALT 11
(1)	023366	000000	0		
(1)	023370	022404	EM13		
(1)	023372	000000	0		;HALT 12
(1)	023374	000000	0		
(1)	023376	022444	EM14		
(1)	023400	000000	0		;HALT 13
(1)	023402	000000	0		
(1)	023404	023134	EM16		
(1)	023406	000000	0	;HALT14	
(1)	023410	000000	0		
(1)	023412	022532	EM21		
(1)	023414	022621	EM24	;HALT15	
(1)	023416	000000	0		
(1)					
(1)	023420	022507	EM20		
(1)	023422	022621	EM24	;HALT16	
(1)	023424	000000	0		
(1)	023426	022552	EM22		
(1)	023430	022621	EM24	;HALT17	
(1)	023432	000000	0		
(1)	023434	022552	EM22		
(1)	023436	022641	EM25	;HALT20	
(1)	023440	000000	0		
(1)	023442	022663	EM26		
(1)	023444	000000	0	;HALT21	
(1)	023446	000000	0		
(1)	023450	022700	EM27		
(1)	023452	000000	0	;HALT22	
(1)	023454	000000	0		
(1)	023456	022715	EM30		
(1)	023460	000000	0	;HALT23	
(1)	023462	000000	0		

(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:		
3634	000001			.END		





















TRP.PC	006222	1579#												
TSOM =	000400	774#	1912	1977	2084	2210	2322	2394	2488	2532	2607	2647	2650	2691
		2714	2717	2761	2768	2771	2809	2812	2815	2830	2871	2874	2877	2880
		2984	2991	3106	3113	3234	3275	3345	3348	3351	3427	3430	3433	3582
TSTNO	001226	677#	945*	1526	1658	1664	1666	1751*	1783*	1817*	1851*	1885*	1959*	2056*
		2180*	2305*	2378*	2503*	2600*	2640*	2683*	2753*	2801*	2857*	2968*	3079*	3226*
		3267*	3330*	3408*	3566*									
TST1	007160	1661	1678	1751#										
TST10	011432	2057	2180#											
TST11	012240	2181	2305#											
TST12	012554	2306	2378#											
TST13	013344	2379	2503#											
TST14	014004	2504	2600#											
TST15	014156	2601	2640#											
TST16	014352	2641	2683#											
TST17	014720	2684	2753#											
TST2	007264	1752	1783#											
TST20	015142	2754	2801#											
TST21	015406	2802	2857#											
TST22	016126	2858	2968#											
TST23	016610	2969	3079#											
TST24	017432	3080	3226#											
TST25	017630	3227	3267#											
TST26	020110	3268	3330#											
TST27	020450	3331	3408#											
TST3	007400	1784	1817#											
TST30	021454	3409	3566#	3633										
TST31 =	***** U	3567												
TST4	007514	1818	1851#											
TST5	007632	1852	1885#											
TST6	010172	1886	1959#											
TST7	010632	1960	2056#											
TTST	003174	1123*	1124*	1126*	1127*	1183#								
TWOSYN=	000000	615#												
TXACT =	001000	753#	2620	2666										
TXCSR	001412	845#	1502*	1530*	1533*	1627*	1628*	1629	1754*	1760*	1765*	1788*	1794*	1799*
		1822*	1828*	1833*	1856*	1862*	1868*	1890*	1892*	1900*	1909	1921	1927	1940*
		1961*	1966*	1974	1976*	1986	1990*	2039*	2058*	2074*	2081	2083*	2093	2097*
		2144*	2165*	2182*	2200*	2207	2209*	2219	2223*	2270*	2291*	2307*	2318*	2320
		2323	2360*	2380*	2390*	2392	2395*	2412	2466*	2471	2481	2489*	2505*	2529*
		2530	2533*	2546*	2584*	2602*	2605*	2606*	2616	2624*	2642*	2645*	2646*	2648
		2651	2654*	2663	2666	2685*	2688*	2690*	2692	2708*	2712*	2713*	2715	2718
		2737*	2755*	2758*	2760*	2766	2769	2803*	2806*	2808*	2810	2813	2827	2865*
		2868*	2870*	2872	2875	2878	2883	2886	2889	2892	2895*	2950*	2978*	2981*
		2983*	2989	2994*	3006*	3057	3060*	3100*	3103*	3105*	3111	3116*	3128*	3204
		3207*	3228*	3230*	3233*	3245*	3278*	3282*	3283	3339*	3342*	3344*	3346	3349
		3354*	3366*	3379*	3421*	3424*	3426*	3428	3431	3435*	3448*	3476*	3568*	3579*
		3581*	3608*											
TXDBUF	001414	846#	1631*	1632*	1633	1902	1904	1912*	1914	1916	1926*	1929*	1931	1933
		1977*	1979	1981	1989*	1994	1996	2027*	2033*	2036*	2084*	2086	2088	2096*
		2101	2103	2132*	2138*	2141*	2210*	2212	2214	2222*	2227	2229	2258*	2264*
		2267*	2322*	2325*	2329*	2331	2333	2394*	2397	2399	2414*	2416	2418	2474*
		2484*	2488*	2532*	2537	2539	2575*	2581*	2607*	2609	2611	2647*	2650*	2653*
		2656	2658	2691*	2694*	2696	2698	2714*	2717*	2720*	2722	2724	2761*	2762
		2764	2768*	2771*	2773	2775	2809*	2812*	2815*	2820	2822	2830*	2871*	2874*
		2877*	2880*	2885*	2888*	2891*	2894*	2897	2899	2984*	2985	2987	2991*	2997





. INSTR	003412	816	1236#		
. INST1	003432	1240#	1260		
. MSG	003434	1238*	1241#		
. PARAM	003536	820	1268#		
. PFAIL	005050	631	936	1545#	1549
. PKCLK	005006	830	1527#		
. RES05	003776	824	1339#		
. SAV05	003736	822	1325#		
. SCOPE	003160	810	1180#		
. SCOP1	003312	812	1210#		
. SETFL	004242	832	1408#	1420	
. START	001562	649	934#	946	
. TRPSR	004316	635	1428#		
. TRPTA	001344	808#	1433		
. TYPE	003336	814	1220#		

. ABS. 023540 000

ERRORS DETECTED: 0

DSKZ:CZDPDC,DSKZ:CZDPDC.SEQ=DSKZ:DUP11.MAC,DSKZ:CZDPDC.P11

RUN-TIME: 10 15 1 SECONDS

RUN-TIME RATIO: 54/27=1.9

CORE USED: 25K (49 PAGES)

DOCUMENT PAGES: 83