

DUP11

DUP11 CONFIDENCE TST
CZDPEB0

AH-8589B-MC
COPYRIGHT © 74-79
FICHE 1 OF 1

SEP 1979
digital
MADE IN USA

This microfiche card contains a grid of frames. The frames on the left side of the card contain data, while the right side is mostly blank. The data in the frames is organized into columns and rows, with some frames containing headers and footers. The data appears to be a list of items, possibly related to a confidence test, with various fields and values. The text is small and difficult to read due to the resolution of the scan.

.REM @
.

IDENTIFICATION

PRODUCT CODE: AC-8588B-MC
PRODUCT NAME: CZDPEB0 DUP11 CONFIDENCE TST
DATE: JANUARY 1979
MAINTAINER: DIAGNOSTICS

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

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) 1974, 1979 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

1.0 ABSTRACT

THE FUNCTION OF THE PROGRAM IS TO PROVIDE A LEVEL OF CONFIDENCE IN THE OPERATION OF THE DUP-11 WITHOUT CHANGING JUMPERS OR SWITCHES FROM CUSTOMER CONFIGURATION.

THE OPTION IS TESTED IN SDLC MODE (BIT-STUFFING), THEN IN DEC MODE USING A SIMULATED DDCMP-LINE PROTOCOL WITH AN IMBEDDED CRC CHARACTER. BOTH OF THESE MODES WILL BE TESTED OVER A CABLE IF A TURNAROUND IS POSSIBLE.

THE MODEM CONTROL LEADS WILL ALSO BE TESTED, IF THE H325 TURN-AROUND CONNECTOR IS USED. THE DETERMINATION OF WHAT WILL BE TESTED IS DONE BY ANSWERING A 'PARAMETER DIALOG' (LOAD ADDRESS 200, START ADDRESS 0 OR 1.) ALL QUESTIONS MUST BE ANSWERED. IF AN ERROR SHOULD OCCUR, A TYPEOUT WILL EXPLAIN THE FUNCTIONAL AREA OF THE DEVICE WHICH FAILED. TO REPAIR THE OPTION, THE DIAGNOSTICS WILL HAVE TO BE RUN.

ADDITIONALLY THE MODEM DATA LEADS MAY NOW BE TESTED IF A MODEM HAS THE ANALOG LOOPBACK FEATURE ENABLED.

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:

- 1.1 CZDPB [REV] BASIC AND OFFLINE TRANSMITTER TESTS.
- 1.2 CZDPC [REV] OFFLINE RECEIVER TESTS, MODEM CONTROL AND INTERRUPT TESTS
- 1.3 CZDPD [REV] OFFLINE SDLC DATA AND FUNCTION OFFLINE DECMODE DATA AND FUNCTION TESTS
- 1.4 CZDPO [REV] DP11 OVERLAY FOR INTERPROCESSOR TEST PROGRAM

NOTE: THE FOURTH TAPE IS:

- 1.5 CZDPE [REV] THIS CONFIDENCE TEST.

NOTE

CZDPE WILL ONLY TEST ONE DUP11 AT A TIME. IF TESTING OF ANOTHER DUP11 IS DESIRED, RESTART AT 200 AND ENTER NEW PARAMETERS.

2.0 REQUIREMENTS

2.1 EQUIPMENT

- 2.1.1. ANY PDP11 FAMILY CPU (WITH MINIMUM 4K MEMORY)
- 2.1.2. ASR 33 (OR EQUIVALENT)
- 2.1.3. DUP11
- 2.1.4. H325 TEST CONNECTOR
- 2.1.5. MODEM WITH ANALOG LOOPBACK FEATURE

2.2 STORAGE

PROGRAM WILL USE ALL 4K 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
	(*)=
4K	17
8K	37
12K	57
16K	77
20K	117
24K	137
28K	157

- 3.1.1 PLACE ADDRESS OF ABS LOADER INTO SWITCH REGISTER. (ALSO PLACE 'HALT' SW UP)
- 3.1.2 DEPRESS 'LOAD ADDRESS' KEY ON CONSOLE AND RELEASE.
- 3.1.3 DEPRESS 'START KEY' ON CONSOLE AND RELEASE (PROGRAM SHOULD NOW BE LOADING INTO CPU)

4.0 STARTING PROCEEDURE

- A. SET SWITCH REGISTER TO 000200
- B. DEPRESS 'LOAD ADDRESS' KEY AND RELEASE
- C. LEAVE SWR BIT 7=1 TO USE EXISTING PARAMETERS PREVIOUSLY SET UP BY THE DUP11 PARAMETER DIALOG. WHENEVER PROGRAM IS STARTED AT ADR 200 WITH SWR BIT7=0, NEW PARAMETERS MUST BE LOADED.

'EXAMPLE'

'MAP OF DUP11 STATUS'

1500	160050	CSR OF DUP11
1502	000300	VECTOR OF 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.

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

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

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: RESERVED
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: RESERVED
SW 02	SET: LOCK ON SELECTED TEST
SW 01	SET: RESTART PROGRAM AT SELECTED TEST
SW 00	SET: RESERVED

SWITCHES 8 THROUGH 15 ARE DYNAMIC AND SHOULD BE USED AS NEEDED IN THE DIAGNOSTIC. SWITCHES 0 THROUGH 2 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 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.

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 14 - LOOP ON TEST. WILL LOOP ON TEST UNTIL SWITCH IS LOWERED.
2. 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 OR THIS EXERCISER.

NOTE: IF ADDRESS 000042 IS NON-ZERO THE PROGRAM ASSUMES IT IS UNDER ACT11 OR XXDP CONTROL AND WILL ACT ACCORDINGLY. AFTER DUP11 IS TESTED, THE PROGRAM WILL RETURN TO 'XXDP' OR 'ACT-11'. ALSO, UNDER 'ACT11' OR 'XXDP' CONTROL THE FOLLOWING DEFAULT PARAMETERS ARE ASSUMED:

4.2.1. ALL JUMPERS ARE ASSUMED TO BE IN THE FOLLOWING CONFIGURATION:

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

4.2.2. H325 TEST CONNECTOR IS ASSUMED TO BE ON.

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

4.2.4. THE BR LEVEL IS ASSUMED TO BE 5.

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.

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.

AT THIS POINT IT IS RECOMMENDED THAT THE NORMAL DIAGNOSTICS BE RUN TO ISOLATE THE ERROR CONDITION.

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). 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.

7.2 OPERATING RESTRICTIONS

DUP11 PARAMETER DIALOG MUST BE ANSWERED BEFORE RUNNING THIS PROGRAM.

1. ANSWER THE COMPLETE PARAMETER DIALOG AGAIN.
2. TOGGLE IN THE NEW CSR AT 1500 AND THE VECTOR AT 1502, THEN RESTART THE PROGRAM WITH SW07=1.

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 CZDPB? 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.

***NOTE**

IT SHOULD BE REMEMBERED THAT CZDPE IS NOT A DIAGNOSTIC , BUT
IT IS A CONFIDENCE TEST DESIGNED TO EVALUATE ONLY THAT THE
DEVICE IS OPERATIONAL. IF THERE IS A FAILURE, THE DUP11 DAGNOSTICS
WILL HAVE TO BE RUN FOR SPECIFIC FAULT DETECTION.

379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412

;*CZDPEB0 /<377>/DUP11 CONFIDENCE TST
;*COPYRIGHT 1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
;*

;STARTING PROCEDURE
;LOAD PROGRAM
;LOAD ADDRESS 000200
;PRESS START
;PROGRAM WILL TYPE 'CZDPEB0 /<377>/DUP11 CONFIDENCE TST ''
;PROGRAM WILL TYPE 'R' TO INDICATE THAT TESTING HAS STARTED
;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
;AND THEN RESUME TESTING

;SWITCH REGISTER OPTIONS
;-----

100000	SW15=100000	=1,HALT ON ERROR
040000	SW14=40000	=1,LOOP ON CURRENT TEST
020000	SW13=20000	=1,INHIBIT ERROR TYPEOUT
010000	SW12=10000	=1,DELETE TYPEOUT/BELL ON ERROR.
004000	SW11=4000	=1,INHIBIT ITERATIONS
002000	SW10=2000	=1,ESCAPE TO NEXT TEST ON ERROR
001000	SW09=1000	=1,LOOP WITH CURRENT DATA
000400	SW08=400	=1,LOOP ON ERROR
000200	SW07=200	
000100	SW06=100	
000040	SW05=40	
000020	SW04=20	
000010	SW03=10	
		;SELECT DUP'S DESIRED ACTIVE
		;NOTE:THIS MUST NOT EXCEED ORIGINAL COUNT
000004	SW02=4	;LOCK ON TEST SELECT
000002	SW01=2	;RESTART PROGRAM AT SELECTED TEST
000001	SW0=1	;ENTER PARAMETERS

```
413
414
415           ;REGISTER DEFINITIONS
416           ;-----
417
418           000000      R0=%0           ;GENERAL REGISTER
419           000001      R1=%1           ;GENERAL REGISTER
420           000002      R2=%2           ;GENERAL REGISTER
421           000003      R3=%3           ;GENERAL REGISTER
422           000004      R4=%4           ;GENERAL REGISTER
423           000005      R5=%5           ;GENERAL REGISTER
424           0C0006      SP=%6           ;PROCESSOR STACK POINTER
425           000007      PC=%7           ;PROGRAM COUNTER
426
427           ;LOCATION EQUIVALENCIES
428           ;-----
429
430           177776      PS=177776       ;PROCESSOR STATUS WORD
431           001150      STACK=1150      ;START OF PROCESSOR STACK
432
433           ;INSTRUCTION DEFINITIONS
434           ;-----
435
436           005746      PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD
437           005726      POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD
438           010046      PUSHRO=10046    ;SAVE R0 ON STACK
439           012600      POPRO=12600     ;RESTORE R0 FROM STACK
440           024646      PUSH2SP=24646   ;DECREMENT STACK TWICE
441           022626      POP2SP=22626    ;INCREMENT STACK TWICE
442           .EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL
443
444
445           100000      BIT15=100000
446           040000      BIT14=40000
447           020000      BIT13=20000
448           010000      BIT12=10000
449           004000      BIT11=4000
450           002000      BIT10=2000
451           001000      BIT9=1000
452           000400      BIT8=400
453           000200      BIT7=200
454           000100      BIT6=100
455           000040      BIT5=40
456           000020      BIT4=20
457           000010      BIT3=10
458           000004      BIT2=4
459           000002      BIT1=2
460           000001      BIT0=1
461
462
```

```
463 ;*****
464 ;-----
465 ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
466 ;THE STANDARD 'TRAP CATCHER' IS PLACED
467 ;BETWEEN ADDRESS 0 TO ADDRESS 776.
468 ;IT LOOKS LIKE 'PC+2 HALT'.
469 ;-----
470 ;*****
471
472      0000C0      .=0      ;STANDARD INTERRUPT VECTORS
473 ;-----
474
475
476      000024      .=24      .PFAIL      ;POWER FAIL HANDLER
477 000024 004312      340      ;SERVICE AT LEVEL 7
478 000026 000340      .HLT      ;ERROR HANDLER
479 000030 003730      340      ;SERVICE AT LEVEL 7
480 000032 000340      .TRPSRV   ;GENERAL HANDLER DISPATCH SERVICE
481 000034 003676      340      ;SERVICE AT LEVEL 7
482 000036 000340      .=40
483      000040      0      ;SAVE FOR ACT-11 OR DDP2
484 000040 000000      0      ;RETURN ADDRESS IF UNDER ACT-11 OR DDP2
485 000042 000000      0      ;SAVE FOR ACT-11 OR DDP2
486 000044 000000      $ENDAD   ;FOR USE WITH ACT-11 OR DDP2
487 000046 002464      .=52
488      000052      0      ;ACT-11 PROGRAM CHARACTERISTICS
489 000052 000000
490
491      000174      .=174
492 000174 0000C0      DISPREG: 0      0      ;SOFTWARE DISPLAY REGISTER
493 000176 000000      SWREG: 0
494      000200      .=200      ;SOFTWARE SWITCH REGISTER
495 000200 000137 001510      JMP      .START      ;GO TO START OF PROGRAM
496
497
498      001000      .=1000
499 001000 005377 055103 050104      MTITLE: .ASCIZ <377><12>/CZDPEB0 /<377>/DUP11 CONFIDENCE TST /<377>
500      001200      .=1200
501 ;SWR AND LIGHTS
502 ;-----
503
504 001200 177570      DISPLAY: 177570      ;11/45 CONSOLE LIGHTS
505 001202 177570      SWR: 177570      ;INDIRECT POINTER TO SWITCH REGISTER
506
507 ;INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
508 ;-----
509
510 001204 177560      TKCSR: 177560      ;TELETYPE KEYBOARD CONTROL REGISTER
511 001206 177562      TKDBR: 177562      ;TELETYPE KEYBOARD DATA BUFFER
512 001210 177564      TPCSR: 177564      ;TELEPRINTER CONTROL REGISTER
513 001212 177566      TPDBR: 177566      ;TELEPRINTER DATA BUFFER
514
515 ;PROGRAM CONTROL PARAMETERS
516 ;-----
517
```

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

518	001214	000000	RETURN: 0	:SCOPE ADDRESS FOR LOOP ON TEST
519	001216	000000	NEXT: 0	:ADDRESS OF NEXT TEST TO BE EXECUTED
520	001220	000000	LOCK: 0	:ADDRESS FOR LOCK ON CURRENT DATA
521	001222	000001	ICOUNT: 1	:NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
522	001224	000000	LPCNT: 0	:NUMBER OF ITERATIONS COMPLETED
523	001226	000000	TSTNO: 0	:NUMBER OF TEST IN PROGRESS
524	001230	000000	PASCNT: 0	:NUMBER OF PASSES COMPLETED
525	001232	000000	ERRCNT: 0	:TOTAL NUMBER OF ERRORS
526	001234	000000	LSTERR: 0	:PC OF LAST ERROR CALL
527				
528			:PROGRAM VARIABLES	
529			:-----	
530	001236	000000	TEMP1: 0	:TEMPORARY STORAGE
531	001240	000000	TEMP2: 0	:TEMPORARY STORAGE
532	001242	000000	TEMP3: 0	:TEMPORARY STORAGE
533	001244	000000	TEMP4: 0	:TEMPORARY STORAGE
534	001246	000000	TEMP5: 0	:TEMPORARY STORAGE
535	001250	000000	SAVR0: 0	:R0 STORAGE
536	001252	000000	SAVR1: 0	:R1 STORAGE
537	001254	000000	SAVR2: 0	:R2 STORAGE
538	001256	000000	SAVR3: 0	:R3 STORAGE
539	001260	000000	SAVR4: 0	:R4 STORAGE
540	001262	000000	SAVR5: 0	:R5 STORAGE
541	001264	000000	SAVSP: 0	:STACK POINTER STORAGE
542	001266	000000	SAVPC: 0	:PROGRAM COUNTER STORAGE
543				
544	001270	000000	SAVR0A: 0	:R0 STORAGE
545	001272	000000	SAVR1A: 0	:R1 STORAGE
546	001274	000000	SAVR2A: 0	:R2 STORAGE
547	001276	000000	SAVR3A: 0	:R3 STORAGE
548	001300	000000	SAVR4A: 0	:R4 STORAGE
549	001302	000000	SAVR5A: 0	:R5 STORAGE
550	001304	000000	SAVSPA: 0	:STACK POINTER STORAGE
551	001306	000000	SAVPCA: 0	:PROGRAM COUNTER STORAGE
552				
553	001310	000005	REPEAT: 5	:REPEAT CTS DLY
554	001312	177777	CTSDLY: -1	:CTS DELAY
555	001314	100000	DELAY: 100000	:DELAY FOR DATA LEAD TESTS
556	001316	000001	DUPACTV: .BLKB 1	:DUP11'S SELECTED ACTIVE.
557	001317	000001	DUPNUM: .BLKB 1	:OCTAL NUMBER OF DUP11'S.
558	001320	000001	SAVNUM: .BLKB 1	:WORKABLE NUMBER.
559		001322	.EVEN	

```
560
561                                     ;CONTROL REGISTER DEFINITIONS
562                                     :-----
563                                     ;RXCSR BIT DEFINITIONS
564      100000      DSCA=BIT15      ;DATA SET CHANGE A
565      040000      RING=BIT14      ;RING
566      020000      CTS=BIT13      ;CLR TO SEND
567      010000      CARDET=BIT12    ;CARRIER DETECT
568      004000      RECACT=BIT11    ;REC ACTIVE
569      002000      SRD=BIT10      ;SEC REC DATA
570      001000      DSR=BIT9       ;DATA SET RDY
571      000400      STPSYN=BIT8     ;STRIP SYNC
572      000200      RXDONE=BIT7    ;REC DONE
573      000100      RINTEN=BIT6    ;REC INTR ENABLE
574      000040      DSINTE=BIT5    ;DSC INTR ENABLE
575      000020      RCVEN=BIT4     ;REC ENABLE
576      000010      STD=BIT3       ;SEC XMIT DATA
577      000004      RTS=BIT2       ;REQ TO SEND
578      000002      DTR=BIT1      ;DATA TERM RDY
579      000001      DSCB=BIT0     ;DATA SET CHANGE B
580                                     ;RXDBUF BIT DEFINITIONS
581      100000      RXDERR=BIT15    ;REC DATA ERROR
582      040000      OVRRUN=BIT14    ;OVERRUN ERROR
583      010000      CRCERR=BIT12    ;CRC ERROR
584      002000      RABORT=BIT10    ;REC ABORT
585      001000      REGM=BIT9      ;REC END OF MESSAGE
586      000400      RSOM=BIT8      ;REC START OF MESSAGE
587                                     ;PARCSR BIT DEFINITIONS
588      100000      DECMOD=BIT15    ;DEC MODE (DDCMP)
589      001000      CRCEN=BIT9      ;CRC ENABLE
590      010000      PRISEC=BIT12    ;PRI/SEC SELECT
591                                     ;TXCSR BIT DEFINITIONS
592      100000      TXDLAT=BIT15    ;TX DATA LATE
593      040000      MTDATA=BIT14    ;MAINT DATA OUT
594      020000      CLK=BIT13      ;CLK
595      010000      MMODEB=BIT12    ;MAINT MODE B
596      004000      MMODEA=BIT11    ;MAINT MODE A
597      002000      BITW=BIT10     ;BIT WINDOW INPUT
598      001000      TXACT=BIT9     ;TX ACTIVE
599      000400      MRESET=BIT8    ;MASTER RESET
600      000200      TXDONE=BIT7    ;XMIT DONE
601      000100      TXINTE=BIT6    ;XMIT DONE INTR ENABLE
602      000020      SEND=BIT4      ;SEND
603      000010      HDXEN=BIT3     ;HDX/FDX
604                                     ;TXCSR WRD DEFINITIONS
605      000000      USER=0         ;USER MODE
606      014000      MMODE=14000    ;MAINT INT MODE
607      010000      MEXT=10000     ;MAINT EXT MODE
608      004000      SYSTST=4000    ;SYSTEM TEST MODE
609
610                                     ;TXDBUF BIT DEFINITIONS
611                                     :-----
612      100000      RCRC7T=BIT15
613      040000      RCRCIN=BIT14
614      020000      TCRC7T=BIT13
615      010000      TCRCIN=BIT12
```

616		004000	TIMER=BIT11	:MAINTENANCE TIMER
617		002000	TABORT=BIT10	:TRANSMIT ABORT
618		001000	TEOM=BIT9	:TRANSMIT END OF MESSAGE
619		000400	TSOM=BIT8	:TRANSMIT START OF MESSAGE

620
621 :MISC. PROGRAM DEFINITIONS
622 :-----

623	001322	000000	PRIRTY: .WORD 0
624	001324	000001	TCNFLG: .BLKB 1
625	001325	000001	OPCLRJ: .BLKB 1
626	001326	000000	DATA: .WORD 0
627	001330	000000	SHIFTS: .WORD 0
628	001332	000000	MIND: .WORD 0
629	001334	000000	FLAG: .WORD 0
630	001336	000001	STJMFL: .BLKW 1
631	001340	000001	SRJMFL: .BLKW 1
632	001342	000000	MDMFLG: .WORD 0
633	001344	000000	ALJMFL: .WORD 0
634	001346	000000	DSCFLG: .WORD 0

635
636
637 :PROGRAM CONTROL FLAGS
638 :-----
639

640	001350	000	INIFLG: .BYTE 0	:PROGRAM INITIALIZATION FLAG
641	001351	000	ERRFLG: .BYTE 0	:ERROR OCCURED FLAG
642	001352	000	LOKFLG: .BYTE 0	:LOCK ON CURRENT TEST FLAG
643	001353	000	QV.FLG: .BYTE 0	:QUICK VERIFY FLAG.
644				:ON FIRST PASS OF EACH DUP11 ITERATIONS
645				:WILL BE SUPPRESSED
646			.EVEN	

647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678

:DEFINITIONS FOR TRAP SUBROUTINE CALLS
:POINTERS TO SUBROUTINES CAN BE FOUND
:IN THE TABLE IMMEDIATELY FOLLOWING THE DEFINITIONS

```
:*****  
:-----  
:TRPTAB:  
SCOPE=TRAP+0 ;CALL TO SCOPE LOOP AND ITERATION HANDLER  
  .SCOPE  
SCOPE1=TRAP+1 ;CALL TO LOOP ON CURRENT DATA HANDLER  
  .SCOPE1  
TYPE=TRAP+2 ;CALL TO TELETYPE OUTPUT ROUTINE  
  .TYPE  
INSTR=TRAP+3 ;CALL TO ASCII STRING INPUT ROUTINE  
  .INSTR  
INSTER=TRAP+4 ;CALL TO INPUT ERROR HANDLER  
  .INSTER  
PARAM=TRAP+5 ;CALL TO NUMERICAL DATA INPUT ROUTINE  
  .PARAM  
SAV05=TRAP+6 ;CALL TO REGISTER SAVE ROUTINE  
  .SAV05  
RES05=TRAP+7 ;CALL TO REGISTER RESTORE ROUTINE  
  .RES05  
CONVRT=TRAP+10 ;CALL TO DATA OUTPUT ROUTINE  
  .CONVRT  
CNVRT=TRAP+11 ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.  
  .CNVRT  
SETFLG=TRAP+12 ;CALL TO TELETYPE INPUT ROUTINE  
  .SETFLG  
:-----  
:*****
```



```
679                                     ;DUP11 VECTOR AND REGISTER INDIRECT POINTERS
680
681 001402 000000 DUPRVC: 0           ;POINTER TO DUP11 RECEIVER INTERRUPT VECTOR
682 001404 000000 DUPRPS: 0           ;POINTER TO DUP11 RECEIVER INTERRUPT SERVICE PS
683 001406 000000 DUPTVC: 0           ;POINTER TO DUP11 TRANSMITTER INTERRUPT VECTOR
684 001410 000000 DUPTPS: 0           ;POINTER TO DUP11 TRANSMITTER INTERRUPT SERVICE PS
685 001412 000000 RXCSR: 0            ;POINTER TO DUP11 RECEIVER STATUS REGISTER
686 001414 000000 RXDBUF: 0           ;POINTER TO DUP11 RECEIVER DATA BUFFER
687 001416 000000 PARCSR: 0           ;POINTER TO DUP11 PARAMETER STATUS REGISTER
688 001420 000000 TXCSR: 0            ;POINTER TO DUP11 TRANSMITTER STATUS REGISTER
689 001422 000000 TXDBUF: 0           ;POINTER TO DUP11 TRANSMITTER DATA BUFFER
690 001424 000000 DUPSEC: 0           ;POINTER TO DUP11 SECONDARY REGISTER SELECT REGISTER
691 001426 000000 HUPPSR: 0           ;POINTER TO PARAMETER STATUS HIGH BYTE
692 001430 000000 HUPRBF: 0           ;POINTER TO RECEIVER BUFFER HIGH BYTE
693 001432 000000 HUPRCR: 0           ;POINTER TO RECEIVER CONTROL REG HIGH BYTE
694 001434 000000 HUPTBF: 0           ;POINTER TO TRANSMITTER BUFFER HIGH BYTE
695 001436 000000 HUPTCR: 0           ;POINTER TO TRANSMITTER CONTROL REG HIGH BYTE
696
697
698                                     ;DUP11 CONTROL INDICATORS FOR CURRENT DUP11 UNDER TEST
699                                     ;-----
700
701 001440 000 MASK.A: .BYTE 000      ;LAST CHAR TO TEST AND PARITY MASK
702
703 001441 010 CLK.A: .BYTE 8.        ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR
704
705 001442 000000 L00.00: 000000      ;PARAMETERS
706
707
708                                     ;DUP11 STATUS TABLE AND ADDRESS ASSIGNMENTS
709                                     ;-----
710
711                                     . =1500
712 001500 DUP.MAP:
713 001500 000001 DUPCRO: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 0
714 001502 000001 DUPTRO: .BLKW 1      ;VECTOR 'A' FOR DUP11 NUMBER 0
715 001504 000001 DUPO.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 0
716
717 001506 000000 DUP.END: 000000
718
719
720
721
```

```

722
723           :PROGRAM INITIALIZATION
724           :LOCK OUT INTERRUPTS
725           :SET UP PROCESSOR STACK
726           :SET UP POWER FAIL VECTOR
727           :CLEAR PROGRAM CONTROL FLAGS AND COUNTS
728           :TYPE TITLE MESSAGE
729
730 001510 012737 000340 177776 .START: MOV #340,PS           :LOCK OUT INTERRUPTS
731 001516 012706 001150           MOV #STACK,SP         :SET UP STACK
732 001522 012737 004312 000024   MOV #.PFAIL,@#24     :SET UP POWER FAIL VECTOR
733 001530 005037 001230           CLR PASCNT           :CLEAR PASS COUNT
734 001534 105037 001351           CLRB ERRFLG         :CLEAR ERROR FLAG
735 001540 105037 001353           CLRB QV.FLG        :ZERO QUICK VERIFY FLAG
736 001544 105037 001317           CLRB DUPNUM
737 001550 105037 001320           CLRB SAVNUM
738 001554 105037 001316           CLRB DUPACTV
739 001560 005037 001232           CLR ERRCNT          :CLEAR ERROR COUNT
740 001564 005037 001234           CLR LSTERR          :CLEAR LAST ERROR POINTER
741 001570 012737 000001 001226   MOV #1,TSTNO        :SET UP FOR TEST 1
742 001576 012737 001510 001214   MOV #.START,RETURN  :SET UP FOR POWER FAIL BEFORE
743                                     :TESTING STARTS
744 001604 013746 000006           MOV @#6,-(SP)       :SAVE CURRENT VECTORS
745 001610 013746 000004           MOV @#4,-(SP)
746 001614 012737 001630 000004   MOV #12$,@#4
747 001622 005777 177354           TST @SWR
748 001626 000407           BR 13$              :REFERENCE HARDWARE SWITCH REG
749 001630 012737 000176 001202 12$: MOV #SWREG,SWR       :BR IF IT EXISTS
750 001636 012737 000174 001200   MOV #DISPREG,DISPLAY :POINT TO SOFT SWR
751 001644 022626           CMP (SP)+,(SP)+    :POINT TO SOFT DISPLAY REG
752 001646 012637 000004 13$:   MOV (SP)+,@#4      :ADJUST STACK
753 001652 012637 000006           MOV (SP)+,@#6      :RESTORE VECTORS
754 001656 005737 000042 11$:   TST @#42
755 001662 001123           BNE 66$             :IF ACT11 OUTO MODE
756 001664 104402 001000           TYPE ,MTITLE       :DON'T TYPE ID
757 001670 105777 177306 6$:   TSTB @SWR          :TYPE TITLE MESSAGE
758 001674 100002           BPL 10$             :USE SAME PARAMETERS?
759 001676 000137 002246           JMP .BEGIN         :IF NO, BR AND INPUT NEW PARAMETERS
760 001702 10$:
761 001702 012700 001500           MOV #DUP.MAP,R0    :CLR MAP
762 001706 005020 68$:   CLR (R0)+
763 001710 020027 001506           CMP R0,#DUP.END   :DONE WITH MAP?
764 001714 001374           BNE 68$            :BR IF NO
765 001716 105037 001342           CLRB MDMFLG
766 001722 105037 001344           CLRB ALJMFL
767 001726 104403           INSTR :OUTPUT MESSAGE & GET INPUT STRING
768 001730 004707           MCSR :MESSAGE
769 001732 104405           PARAM :CONVERT STRING
770 001734 160000           160000 :LOW LIMIT
771 001736 175500           175500 :HIGH LIMIT
772 001740 001500           DUPCRO :STORE AT THIS LOCATION
773 001742 001 .BYTE 1 :MASK
774 001743 001 .BYTE 1 :HOW MANY TIMES + 2
775 001744 104403           INSTR :OUTPUT MESSAGE & GET INPUT STRING
776 001746 004726           MVEC :MESSAGE
777 001750 104405           PARAM :CONVERT STRING
  
```

PROGRAM INITIALIZATION AND START UP.

778	001752	000300		300	:LOW LIMIT	
779	001754	000770		770	:HIGH LIMIT	
780	001756	001502		DUPTR0	:STORE AT THIS LOCATION	
781	001760	001		1	:MASK	
782	001761	001	.BYTE	1	:HOW MANY TIMES + 2	
783	001762	104403	.BYTE	INSTR	:OUTPUT MESSAGE & GET INPUT STRING	
784	001764	004741		MMODEM	:MESSAGE	
785	001766	104412		SETFLG	:SET FLAG BASED UPON INPUT STRING	
786	001770	001342		MDMFLG	:THIS FLAG	
787	001772	105737	001342	TSTB	MDMFLG	:MODEM FLAG SET?
788	001776	001405		BEQ	71\$:IF BR.
789	002000	105037	001324	CLRB	TCNFLG	
790	002004	105037	001344	CLRB	ALJMFL	
791	002010	000436		BR	70\$	
792	002012			71\$:		
793	002012	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING	
794	002014	005112		MTCN	:MESSAGE	
795	002016	104412		SETFLG	:SET FLAG BASED UPON INPUT STRING	
796	002020	001324		TCNFLG	:THIS FLAG	
797	002022	105737	001324	TSTB	TCNFLG	
798	002026	001427		BEQ	70\$	
799	002030	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING	
800	002032	005321		MALLJM	:MESSAGE	
801	002034	104412		SETFLG	:SET FLAG BASED UPON INPUT STRING	
802	002036	001344		ALJMFL	:THIS FLAG	
803	002040	105737	001344	TSTB	ALJMFL	
804	002044	001020		BNE	70\$	
805	002046	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING	
806	002050	005037		MJMPR	:MESSAGE	
807	002052	104412		SETFLG	:SET FLAG BASED UPON INPUT STRING	
808	002054	001325		OPCLRJ	:THIS FLAG	
809	002056	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING	
810	002060	005205		MSTJM	:MESSAGE	
811	002062	104412		SETFLG	:SET FLAG BASED UPON INPUT STRING	
812	002064	001336		STJMFL	:THIS FLAG	
813	002066	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING	
814	002070	005240		MSRJM	:MESSAGE	
815	002072	104412		SETFLG	:SET FLAG BASED UPON INPUT STRING	
816	002074	001340		SRJMFL	:THIS FLAG	
817	002076	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING	
818	002100	005375		MDSC	:MESSAGE	
819	002102	104412		SETFLG	:SET FLAG BASED UPON INPUT STRING	
820	002104	001346		DSCFLG	:THIS FLAG	
821	002106	105237	001317	70\$:	INCB	DUPNUM
822	002112	105237	001320	INCB	SAVNUM	
823	002116	105237	001316	INCB	DUPACTV	
824	002122	012700	001500	MOV	#DUPCRO,RO	
825	002126	000137	002246	66\$:	JMP	.BEGIN
826	002132					:LET'S GO.

```

827                                     ;** FOLLOWING PARAMETERS ARE LOADED IF UNDER ACT11 OR XXDP CONTROL.**
828
829 002132 012700 001500                MOV    #DUP,MAP,RO                ;SET UP POINTER
830 002136 105237 001316                INCB   DUPACTV                  ;DEVICE ACTIVE FOR TESTING
831 002142 012710 160050                MCV    #160050,(RO)             ;CSR
832 002146 012760 000770 000002        MOV    #770,2(RO)              ;VECTOR
833 002154 012760 140026 000004        MOV    #140026,4(RO)           ;STATUS AND SYNC
834 002162 112737 000005 001322        MOVB   #5,PRIORITY             ;PRIORITY
835 002170 113737 001236 001324        MOVB   TEMP1,TCNFLG            ;TURN-AROUND-CONNECTER FLAG
836 002176 113737 001236 001325        MOVB   TEMP1,OPCLRJ           ;OPTIONAL CLEAR JUMPER
837 002204 105037 001342                CLRB   MDMFLG                  ;MODEM FLAG
838 002210 104402 005273                TYPE   ,XHEAD                  ;TYPE HEADER
839 002214 012737 001500 001236        MOV    #DUP,MAP,TEMP1          ;SET POINTER
840 002222 017737 177010 001240 5$:    MOV    @TEMP1,TEMP2            ;SET DATA
841 002230 001406                        BEQ    .BEGIN                  ;ALL DONE WITH DATA
842 002232 104410
843 002234 005442
844 002236 062737 000002 001236        ADD    #2,TEMP1                ;UPDATE POINTER
845 002244 000766                        BR     5$
846
847                                     ;TEST START AND RESTART
848                                     ;-----
849
850 002246 012737 000340 177776 .BEGIN: MOV    #340,PS                ;LOCK OUT INTERRUPTS
851 002254 012706 001150                MOV    #STACK,SP              ;SET UP STACK
852 002260 005737 000042                TST    @#42                    ;IS PROGRAM UNDER MONITOR CONTROL
853 002264 001023                        BNE    2$                      ;BR IF YES
854 002266 032777 000004 176706        BIT    #BIT2,@SWR             ;CHECK FOR LOCK ON TEST
855 002274 001411                        BEQ    1$                      ;BR IF NO LOCK DESIRED.
856 002276 104402 004526                TYPE   ,MLOCK                 ;TYPE LOCK SELECTED.
857 002302 012737 000240 002554        MOV    #NOP,TTST              ;ADJUST SCOPE ROUTINE.
858 002310 012737 000240 002556        MOV    #NOP,TTST+2            ;SET UP TO LOCK
859 002316 000406                        BR     2$                      ;CONTINUE ALONG.
860 002320 013737 002666 002554 1$:    MOV    BRW,TTST                ;PREPARE NORMAL SCOPE ROUTINE
861 002326 013737 002670 002556        MOV    BRX,TTST+2             ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
862 002334 012737 005624 001214 2$:    MOV    #CYCLE,RETURN          ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
863 002342 104402 004416                TYPE   ,MR                    ;TYPE R
864 002346 000177 176642                JMP    @RETURN                ;START TESTING

```

```

865                                     :END OF PASS
866                                     :TYPE NAME OF TEST
867                                     :UPDATE PASS COUNT
868                                     :CHECK FOR EXIT TO ACT-11
869                                     :RESTART TEST
870
871 002352 005037 001234                .EOP: CLR      LSTERR      :CLEAR LAST ERROR PC
872 002356 105037 001351                CLR      ERRFLG     :CLEAR ERROR FLAG
873 002362 005237 001230                INC      PASCNT     :UPDATE PASS COUNT
874 002366 013777 001230 176604        MOV      PASCNT,@DISPLAY :DISPLAY PASS COUNT
875 002374 104402 004373                TYPE    ,MEPASS    :TYPE END PASS
876 002400 104402 004555                TYPE    ,MCSR      :TYPE CSR
877 002404 104411 002510                CNVRT   ,XCSR      :SHOW IT
878 002410 104402 004563                TYPE    ,MVECX     :TYPE VECTOR
879 002414 104411 002516                CNVRT   ,XVEC      :SHOW IT
880 002420 104402 004571                TYPE    ,MPASSX    :TYPE PASSES
881 002424 104411 002524                CNVRT   ,XPASS     :SHOW IT
882 002430 104402 004602                TYPE    ,MERRX     :TYPE ERRORS
883 002434 104411 002532                CNVRT   ,XERR      :SHOW IT
884 002440 112737 000377 001353        MOV      #377,QV.FLG :SET THE QUICK VERIFY FLAG.
885 002446 113737 001317 001320        MOV      DUPNUM,SAVNUM :RESTORE THE COUNT
886 002454 013701 000042                MOV      @#42,R1    :CHECK FOR ACT-11 OR DDP
887 002460 001406                        BEQ     RESTRT     :IF NOT, CONTINUE TESTING
888 002462 000005                        RESET                    :STOP THE SHOW--CLEAR THE WORLD
889 002464
890 002464 004711                $ENDAD: JSR     PC,(R1)
891 002466 000240                NOP
892 002470 000240                NOP
893 002472 000240                NOP
894 002474 000240                NOP
895 002476 012737 005624 001214        RESTRT: MOV     #CYCLE,RETURN
896 002504 000137 005624                JMP     CYCLE
897 002510 000001                XCSR:   1
898 002512 006 002                .BYTE  6,2
899 002514 001412                RXCSR
900 002516 000001                XVEC:   1
901 002520 003 002                .BYTE  3,2
902 002522 001402                DUPRVC
903 002524 000001                XPASS:  1
904 002526 006 002                .BYTE  6,2
905 002530 001230                PASCNT
906 002532 000001                XERR:   1
907 002534 006 002                .BYTE  6,2
908 002536 001232                ERRCNT
909
910                                     :SCOPE LOOP AND INTERATION HANDLER
911
912 002540 005037 001234                .SCOPE: CLR     LSTERR      :CLEAR LAST ERROR PC
913 002544 010016                MOV     R0,(SP)    :SAVE R0 ON STACK
914 002546 032777 040000 176426        BIT     #BIT14,@SWR :LOOP ON TEST?
915 002554 001407                TTST:  BEQ     1$     :BR IF NO (IF LOCK SW01 = 1;THIS LOCATION = 240)
916 002556 000437                BR      3$         :GO TO 3$ (DITTO)
917 002560 105777 176420                TSTB   @TKCSR     :KYBD DONE?
918 002564 100034                BPL     3$         :BR IF NO (LOCK: HIT A KEY UN TTY TO GO TO NEXT TEST)
919 002566 017700 176414                MOV     @TKDBR,R0 :CLR DONE BIT
920 002572 000415                BR      2$         :CONTINUE

```

```

921 002574 032777 004000 176400 1$: BIT #SW11,@SWR ;DELETE ITERATION (QUICK PASS)?
922 002602 001011 BNE 2$ ;BR IF YES
923 002604 105737 001353 TSTB QV.FLG ;HAS FIRST PASS BEEN COMPLETED?
924 002610 001406 BEQ 2$ ;BR IF QUICK VERIFY
925 002612 005237 001224 INC LPCNT ;UPDATE ITERATION COUNTER
926 002616 023737 001224 001222 CMP LPCNT,ICOUNT ;ALL ITERATIONS DONE?
927 002624 001014 BNE 3$ ;BR IF NOT YET
928 002626 105037 001351 2$: CLRB ERRFLG ;PREPARE FOR NEW TEST
929 002632 005037 001224 CLR LPCNT ;START ICOUNT AT ZERO
930 002636 005037 001220 CLR LOCK
931 002642 012737 000050 001222 MOV #50,ICOUNT ;RESET ITERATIONS
932 002650 013737 001216 001214 MOV NEXT,RETURN ;GET NEXT TEST
933 002656 011600 3$: MOV (SP),R0 ;POP R0 OFF STACK
934 002660 022626 POP2SP ;FAKE AN RTI
935 002662 000177 176326 JMP @RETURN ;GO DO THE TEST
936 002666 001407 BRW: 1407
937 002670 000437 BRX: 437
938 ;TELETYPE OUTPUT ROUTINE
939 -----
940
941 002716 010546 .TYPE: MOV R5,-(SP) ;SAVE R5 ON THE STACK.
942 002720 017605 MOV @2(SP),R5 ;GET ADDRESS OF MESSAGE.
943 002724 062766 000002 000002 ADD #2,2(SP) ;POP OVER ADDRESS.
944 002732 032777 010000 176242 1$: BIT #SW12,@SWR ;INHIBIT ALL PRINT OUT??
945 002740 001012 BNE 3$ ;BR IF NO PRINT OUT WANTED (SW12=1)
946 002742 105715 TSTB (R5) ;IS NUMBER MINUS? (MSB=1(BIT7))
947 002744 100002 BPL 2$ ;BR IF NUMBER IS PLUS
948 002746 104402 004352 TYPE ,MCRLF ;TYPE A CR/LF!
949 002752 105777 176232 2$: TSTB @TPCSR ;TTY READY?
950 002756 100375 BPL 2$ ;BR IF NO.
951 002760 112577 176226 MOVB (R5)+,@TPDBR ;PRINT CURRENT CHAR.
952 002764 001362 BNE 1$ ;IF NOT ZERO KEEP PRINTING!
953 002766 012605 3$: MOV (SP)+,R5 ;END OF OUTPUT. RESTORE R5
954 002770 000002 RTI ;GO HOME
955 -----
956
957 002772 010346 .INSTR: MOV R3,-(SP) ;SAVE R3 ON STACK
958 002774 010446 MOV R4,-(SP) ;SAVE R4 ON STACK
959 002776 017637 000004 003014 MOV @4(SP),MSG
960 003004 062766 000002 000004 ADD #2,4(SP)
961 003012 104402 .INST1: TYPE
962 003014 000000 .MSG: 0
963 003016 012704 005560 MOV #INBUF,R4
964 003022 012703 000007 MOV #7,R3
965 003026 105777 176152 1$: TSTB @TKCSR
966 003032 100375 BPL 1$
967 003034 117714 176146 MOVB @TKDBR,(R4)
968 003040 142714 000200 BICB #200,(R4)
969 003044 122427 000015 CMPB (R4)+,#15
970 003050 001417 BEQ INSTR2
971 003052 105777 176132 2$: TSTB @TPCSR
972 003056 100375 BPL 2$
973 003060 017777 176122 176124 MOV @TKDBR,@TPDBR
974 003066 005303 DEC R3
975 003070 001356 BNE 1$
976 003072 012604 MOV (SP)+,R4
  
```

977	003074	012603			MOV	(SP)+,R3	
978	003076	010346			.INSTE: MOV	R3,-(SP)	
979	003100	010446			MOV	R4,-(SP)	
980	003102	104402	004346		TYPE	,MQM	
981	003106	000741			BR	.INST1	
982	003110	012604			INSTR2: MOV	(SP)+,R4	:RESTORE R4
983	003112	012603			MOV	(SP)+,R3	:RESTORE R3
984	003114	000002			RTI		
985							
986						:CONVERT ASCII STRING TO OCTAL	
987						-----	
988							
989	003116	010546			.PARAM: MOV	R5,-(SP)	
990	003120	010446			MOV	R4,-(SP)	
991	003122	016605	000004		MOV	4(SP),R5	
992	003126	012537	003306		MOV	(R5)+,LOLIM	
993	003132	012537	003310		MOV	(R5)+,HILIM	
994	003136	012537	003312		MOV	(R5)+,DEVADR	
995	003142	112537	003314		MOVB	(R5)+,LOBITS	
996	003146	112537	003315		MOVB	(R5)+,ADRCNT	
997	003152	010566	000004		MOV	R5,4(SP)	
998	003156	005005			PARAM1: CLR	R5	
999	003160	012704	005560		MOV	#INBUF,R4	
1000	003164	122714	000015		CMPB	#15,(R4)	
1001	003170	001420			BEQ	PARERR	
1002	003172	121427	000060		1\$: CMPB	(R4),#60	
1003	003176	002415			BLT	PARERR	
1004	003200	121427	000067		CMPB	(R4),#67	
1005	003204	003012			BGT	PARERR	
1006	003206	142714	000060		BICB	#60,(R4)	
1007	003212	152405			BISB	(R4)+,R5	
1008	003214	122714	000015		CMPB	#15,(R4)	
1009	003220	001406			BEQ	LIMITS	
1010	003222	006305			ASL	R5	
1011	003224	006305			ASL	R5	
1012	003226	006305			ASL	R5	
1013	003230	000760			BR	1\$	
1014	003232	104404			PARERR: INSTER		
1015	003234	000750			BR	PARAM1	
1016							
1017						:TEST TO SEE IF NUMBER IS WITHIN LIMITS	
1018						-----	
1019							
1020	003236	020537	003310		LIMITS: CMP	R5,HILIM	
1021	003242	101373			BHI	PARERR	
1022	003244	020537	003306		CMP	R5,LOLIM	
1023	003250	103770			BLO	PARERR	
1024	003252	133705	003314		BITB	LOBITS,R5	
1025	003256	001365			BNE	PARERR	
1026							
1027						:STORE NUMBER AT SPECIFIED ADDRESS	
1028							
1029	003260	013704	003312		1\$: MOV	DEVADR,R4	
1030	003264	010524			MOV	R5,(R4)+	
1031	003266	062705	000002		ADD	#2,R5	
1032	003272	105337	003315		DECB	ADRCNT	

1033	003276	001372			BNE	1\$
1034	003300	012604			MOV	(SP)+,R4
1035	003302	012605			MOV	(SP)+,R5
1036	003304	000002			RTI	
1037	003306	000000			LOLIM:	0
1038	003310	000000			HILIM:	0
1039	003312	000000			DEVADR:	0
1040	003314	000000			LOBITS:	0
1041		003315			ADRCNT=L	LOBITS+1
1042					:CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER	
1043					:-----	
1044						
1045	003410	104402	004352		.CONVR: TYPE	.MCRLF
1046	003414	010046			.CNVRT: MOV	R0,-(SP)
1047	003416	010146			MOV	R1,-(SP)
1048	003420	010346			MOV	R3,-(SP)
1049	003422	010446			MOV	R4,-(SP)
1050	003424	010546			MOV	R5,-(SP)
1051	003426	017601	000012		MOV	@12(SP),R1
1052	003432	062766	000002	000012	ADD	#2,12(SP)
1053	003440	012137	003614		MOV	(R1)+,WRDCNT
1054	003444	112137	003616		1\$: MOV	(R1)+,CHRCNT
1055	003450	112137	003617		MOVB	(R1)+,SPACNT
1056	003454	013137	005620		MOV	@(R1)+,BINWRD
1057	003460	013704	003620		2\$: MOV	BINWRD,R4
1058	003464	113705	003616		MOVB	CHRCNT,R5
1059	003470	012700	005454		MOV	#TEMP,R0
1060	003474	010403			3\$: MOV	R4,R3
1061	003476	042703	177770		BIC	#177770,R3
1062	003502	062703	000060		ADD	#060,R3
1063	003506	110320			MOVB	R3,(R0)+
1064	003510	000241			CLC	
1065	003512	006004			ROR	R4
1066	003514	000241			CLC	
1067	003516	006004			ROR	R4
1068	003520	000241			CLC	
1069	003522	006004			ROR	R4
1070	003524	005305			DEC	R5
1071	003526	001362			BNE	3\$
1072	003530	012703	005516		MOV	#MDATA,R3
1073	003534	114023			4\$: MOV	-(R0),(R3)+
1074	003536	105337	003616		DECB	CHRCNT
1075	003542	001374			BNE	4\$
1076	003544	105737	003617		TSTB	SPACNT
1077	003550	001405			BEQ	6\$
1078	003552	112723	000040		5\$: MOV	#040,(R3)+
1079	003556	105337	003617		DECB	SPACNT
1080	003562	001373			BNE	5\$
1081	003564	105013			6\$: CLRB	(R3)
1082	003566	104402	005516		TYPE	.MDATA
1083	003572	005337	003614		DEC	WRDCNT
1084	003576	001322			BNE	1\$
1085	003600	012605			MOV	(SP)+,R5
1086	003602	012604			MOV	(SP)+,R4
1087	003604	012603			MOV	(SP)+,R3
1088	003606	012601			MOV	(SP)+,R1


```

1089 003610 012600          MOV      (SP)+,R0
1090 003612 000002          RTI
1091 003614 000030          WRDCNT: 0
1092 003616 000000          CHRCNT: 0
1093          003617          SPACNT=CHRCNT+1
1094 003620 000000          BINWRD: 0
1095
1096
1097          :COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
1098          :BUFFER TO THE CHARACTERS 'N' AND 'Y'.
1099          :IF THE CHARACTER IS 'N' CLEAR THE FLAG
1100          :IF THE CHARACTER IS 'Y' SET THE FLAG
1101
1102 003622 017605 000000      .SETFLG:MOV    @ (SP),R5
1103 003626 042737 000040 005560  BIC      #40,INBUF
1104 003634 122737 000116 005560  CMPB    #'N',INBUF      ;IS IT 'N' ?
1105 003642 001002          BNE     1$
1106 003644 105015          CLRB   (R5)      ;000
1107 003646 000406          BR     2$
1108 003650 122737 000131 005560  1$:  CMPB    #'Y',INBUF      ;IS IT 'Y' ?
1109 003656 001005          BNE     3$
1110 003660 112715 177777          MOVB   #-1,(R5)      ;377
1111 003664 062716 000002          2$:  ADD     #2,(SP)
1112 003670 000002          RTI
1113 003672 104404          3$:  INSTER  :RETRY
1114 003674 000752          BR     .SETFLG
1115
1116
1117          :TRAP DISPATCH SERVICE
1118          :ARGUMENT OF TRAP IS EXTRACTED
1119          :AND USED AS OFFSET TO OBTAIN POINTER
1120          :TO SELECTED SUBROUTINE
1121
1122 003676 011646          .TRPSR:MOV    (SP),-(SP)      ;GET PC OF RETURN
1123 003700 162716 000002          SUB    #2,(SP)      ;=PC OF TRAP
1124 003704 017616 000000          MOV    @ (SP),(SP)      ;GET TRP
1125 003710 006316          TRPOK:ASL    (SP)      ;MULTIPLY TRAP ARG BY 2
1126 003712 042716 177001          BIC    #177001,(SP)      ;CLEAR UNWANTED BITS
1127 003716 062716 001354          ADD    #.TRPTAB,(SP)      ;POINTER TO SUBROUTINE ADDRESS
1128 003722 017616 000000          MOV    @ (SP),(SP)      ;SUBROUTINE ADDRESS
1129 003726 000136          JMP    @ (SP)+      ;GO TO SUBROUTINE
1130
1131          :ERROR HANDLER
1132          :-----
1133
1134 003730 032777 010000 175244  .HLT:  BIT    #SW12,@SWR      ;BELL ON ERROR?
1135 003736 001406          BEQ    XBX              ;BR IF NO BELL
1136 003740 105777 175244          TSTB   @TPCSR          ;TTY READY.
1137 003744 100003          BPL    XBX              ;DON'T WAIT IF TTY NOT READY.
1138 003746 112777 000207 175236  MOVB   #207,@TPDBR      ;PUSH A BELL AT THE TTY.
1139 003754 032777 020000 175220  XBX:  BIT    #SW13,@SWR      ;DELETE ERROR PRINT OUT?
1140 003762 001105          BNE    HALTS            ;BR IF NO PRINT OUT WANTED.
1141 003764 021637 001234          CMP    (SP),LSTERR      ;WAS THIS ERROR FOUND LAST TIME?
1142 003770 001404          BEQ    1$              ;BR IF YES
1143 003772 011637 001234          MOV    (SP),LSTERR      ;RECORD BEING HERE
1144 003776 105037 001351          CLRB   ERRFLG          ;PREPARE HEADER
  
```

1145	004002	104406		1\$:	SAV05		:SAVE ALL PROC REGISTERS
1146	004004	011605			MOV (SP),R5		:GET THE PC OF ERROR
1147	004006	162705	000002		SUB #2,R5		:GET ADDRESS OF TRAP CALL
1148	004012	011504			MOV (R5),R4		:GET HLT INSTRUCTION
1149	004014	006304			ASL R4		:MULT BY TWO
1150	004016	061504			ADD (R5),R4		:DOUBLE IT
1151	004020	006304			ASL R4		:MULT AGAIN
1152	004022	042704	177001		BIC #177001,R4		:CLEAR JUNK
1153	004026	062704	011550		ADD #.ERRTAB,R4		:GET POINTER
1154	004032	012437	004146		MOV (R4)+,ERRMSG		:GET ERROR MESSAGE
1155	004036	012437	004160		MOV (R4)+,DATAHD		:GET DATA HEADRER
1156	004042	011437	004172		MOV (P4),DATABP		:GET DATA TABLE
1157	004046	105737	001351		TSTB ERRFLG		:TYPE HEADREER
1158	004052	001403			BEQ TYPMSG		:BR IF YES
1159	004054	005737	004172		TST DATABP		:DOES DATA TABLE EXIST?
1160	004060	001040			BNE TYPDAT		:BR IF YES.
1161	004062	104402	004352		TYPMSG: TYPE ,MCRLF		
1162	004066	104402	004352		TYPE ,MCRLF		
1163	004072	005737	001220		TST LOCK		
1164	004076	001402			BEQ 1\$		
1165	004100	104402	004625		TYPE ,MASTEK		
1166	004104	104402	004613		1\$: TYPE ,MTSTN		
1167	004110	104411	004300		CNVRT ,XTSTN		:SHOW IT
1168	004114	104402	004702		TYPE ,MERRPC		:TYPE PC.
1169	004120	104411	004272		CNVRT ,ERTAB0		:SHOW IT
1170	004124	104402	004352		TYPE ,MCRLF		:GIVE A CR/LF
1171	004130	112737	177777	001351	MOVB #-1,ERRFLG		:NO MORE HEADER UNLESS NO DATA TABLE.
1172	004136	005737	004146		TST ERRMSG		:IS THERE AN ERROR MESSAGE?
1173	004142	001402			BEQ WRKO.FM		:BR IF NO.
1174	004144	104402			TYPE		:TYPE
1175	004146	000000			ERRMSG: 0		: ERROR MESSAGE
1176	004150				WRKO.FM:		
1177	004150	005737	004160		TST DATAHD		:DATA HEADER?
1178	004154	001402			BEQ TYPDAT		:BR IF NO
1179	004156	104402			TYPE		:TYPE
1180	004160	000000			DATAHD: 0		: DATA HEADER
1181	004162	005737	004172		TYPDAT: TST DATABP		:DATA TABLE?
1182	004166	001402			BEQ RESREG		:BR IF NO.
1183	004170	104410			CONVRT		:SHOW
1184	004172	000000			DATABP: 0		: DATA TABLE
1185	004174	104407			RESREG: RES05		:RESTORE PROC REGISTERS
1186	004176	022737	002464	000042	HALTS: CMP #SENDAD,@#42		:IF ACT-11 AUTO MODE--HALT!!
1187	004204	001403			BEQ 1\$		
1188	004206	005777	174770		TST @SWR		:HALT ON ERROR?
1189	004212	100005			BPL EXITER		:BR IF NO HALT ON ERROR
1190	004214	010046			1\$: PUSHRO		:SAVE RO
1191	004216	016600	000002		MOV 2(SP),R0		:SHOW ERROR PC IN DATA LIGHTS
1192	004222	000000			HALT		:HALT
1193	004224	012600			POPRO		:GET RO
1194	004226	005237	001232		EXITER: INC ERRCNT		:UPDATE ERROR COUNT
1195	004232	032777	000400	174742	BIT #SW08,@SWR		:GOTO TOP OF TEST?
1196	004240	001007			BNE 1\$:BR IF YES
1197	004242	032777	002000	174732	BIT #SW10,@SWR		:GOTO NEXT TEST?
1198	004250	001407			BEQ 2\$:BR IF NO
1199	004252	013737	001216	001214	MOV NEXT,RETURN		:SET FOR NEXT TEST
1200	004260	012706	001150		1\$: MOV #STACK,SP		:RESET SP

1201	004264	000177	174724		JMP	@RETURN	:GOTO SPECIFIED TEST
1202	004270	000002		2\$:	RTI		:RETURN
1203	004272	000001		ERTAB0:	1		
1204	004274	006	002		.BYTE	6,2	
1205	004276	001266			SAVPC		
1206	004300	000001		XTSTN:	1		
1207	004302	003	002		.BYTE	3,2	
1208	004304	001226			TSTNO		
1209							

```

1210
1211 004306 000240 ;WAIT ROUTINE
1212 004310 000207 SMALL: NOP ;STALL
;RTS PC ;RETURN
1213
1214
1215 ;POWER FAIL ROUTINE
1216
1217 004312 012737 004322 000024 .PFAIL: MOV #PWRUP,24 ;LOAD PFAIL VECTOR FOR POWER UP
1218 004320 000000 HALT ;
1219 004322 000005 PWRUP: RESET ;WAIT TTY TO COME UP
1220 004324 012706 001150 MOV #STACK,SP ;REINIT STACK POINTER
1221 004330 012737 004312 000024 MOV #.PFAIL,24 ;LOAD PFAIL VECTOR FOR POWER DOWN
1222 004336 104402 TYPE
1223 004340 004355 MPOWER
1224 004342 000177 174646 JMP @RETURN
1225
1226 004346 020040 000077 MQM: .ASCIZ / ?/
1227 004352 005015 000 MCRLF: .ASCIZ <15><12>
1228 004355 377 053520 020122 MPOWER: .ASCIZ <377>/PWR FAILED. /
1229 004362 040506 046111 042105
1230 004370 020056 000
1231 004373 015 042777 042116 MEPASS: .ASCIZ <15><377>/END PASS CZDPEB /
1232 004400 050040 051501 020123
1233 004406 055103 050104 041105
1234 004414 000040
1235 004416 051377 000 MR: .ASCIZ <377>/R/
1236 004421 377 051120 043517 MERR2: .ASCIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT./
1237 004426 040522 020115 047111
1238 004434 044504 040503 042524
1239 004442 020123 047516 042040
1240 004450 053105 041511 051505
1241 004456 050040 042522 042523
1242 004464 052116 000056
1243 004470 044777 051516 043125 MERR3: .ASCIZ <377>/INSUFFICIENT DATA!/
1244 004476 044506 044503 047105
1245 004504 020124 040504 040524
1246 004512 000041
1247 004514 052377 051505 020124 MTSTPC: .ASCIZ <377>/TEST PC-/
1248 004522 041520 000055
1249 004526 046377 041517 020113 MLOCK: .ASCIZ <377>/LOCK ON SELECTED TEST/
1250 004534 047117 051440 046105
1251 004542 041505 042524 020104
1252 004550 042524 052123 000
1253 004555 103 051123 020072 MCSRX: .ASCIZ /CSR: /
1254 004562 000
1255 004563 126 041505 020072 MVECX: .ASCIZ /VEC: /
1256 004570 000
1257 004571 120 051501 042523 MPASSX: .ASCIZ /PASSES: /
1258 004576 035123 000040
1259 004602 051105 047522 051522 MERRX: .ASCIZ /ERRORS: /
1260 004610 020072 000
1261 004613 124 051505 020124 MTSTN: .ASCIZ /TEST NO: /
1262 004620 047516 020072 000
1263 004625 052 000
1264 004627 377 042523 020124 MASTEK: .ASCIZ /*/
MNEW: .ASCIZ <377>/SET SWITCH REG TO DUP11'S DESIRED ACTIVE./
1265 004634 053523 052111 044103

```

1266	004642	051040	043505	052040	
1267	004650	020117	052504	030520	
1268	004656	023461	020123	042504	
1269	004664	044523	042522	020104	
1270	004672	041501	044524	042526	
1271	004700	000056			
1272	004702	041520	020072	000	MERRPC: .ASCIZ /PC: /
1273	004707	377	042522	020103	MCSR: .ASCIZ <377>/REC CSR ADRS /
1274	004714	051503	020122	042101	
1275	004722	051522	000040		
1276	004726	053377	041505	040440	MVEC: .ASCIZ <377>/VEC ADRS /
1277	004734	051104	020123	000	
1278	004741	377	051511	040440	MMODEM: .ASCIZ <377>/IS A MODEM WITH ANALOG LOOPBACK ENABLED CONNECTED? (Y OR N)
1279	004746	046440	042117	046505	
1280	004754	053440	052111	020110	
1281	004762	047101	046101	043517	
1282	004770	046040	047517	041120	
1283	004776	041501	020113	047105	
1284	005004	041101	042514	020104	
1285	005012	047503	047116	041505	
1286	005020	042524	037504	024040	
1287	005026	020131	051117	047040	
1288	005034	004451	000		
1289	005037	377	051511	052040	MJMPR: .ASCIZ <377>/IS THE OPTIONAL CLR JMPR IN? (Y OR N) /
1290	005044	042510	047440	052120	
1291	005052	047511	040516	020114	
1292	005060	046103	020122	046512	
1293	005066	051120	044440	037516	
1294	005074	020040	054450	047440	
1295	005102	020122	024516	020040	
1296	005110	000040			
1297	005112	044777	020123	044124	MTCN: .ASCIZ <377>/IS THE H325 CONNECTOR ON? (Y OR N) /
1298	005120	020105	031510	032462	
1299	005126	041440	047117	042516	
1300	005134	052103	051117	047440	
1301	005142	037516	024040	020131	
1302	005150	051117	047040	020051	
1303	005156	020040	000		
1304	005161	377	051120	047511	MPAR: .ASCIZ <377>/PRIORITY (4 TO 7) /
1305	005166	044522	054524	024040	
1306	005174	020064	047524	033440	
1307	005202	020051	000		
1308	005205	377	042523	020103	MSTJM: .ASCIZ <377>/SEC TX JMPR IN? (Y OR N) /
1309	005212	054124	045040	050115	
1310	005220	020122	047111	020077	
1311	005226	054450	047440	020122	
1312	005234	024516	000040		
1313	005240	051777	041505	051040	MSRJM: .ASCIZ <377>/SEC RX JMPR IN? (Y OR N) /
1314	005246	020130	046512	051120	
1315	005254	044440	037516	024040	
1316	005262	020131	051117	047040	
1317	005270	020051	000		
1318	005273	377	040515	020120	XHEAD: .ASCIZ <377>/MAP OF DUP11 STATUS/<377>
1319	005300	043117	042040	050125	
1320	005306	030461	051440	040524	
1321	005314	052524	177523	000	

1322	005321	377	051101	020105	MALLJM: .ASCIZ <377>/ARE THE DEFAULT JUMPERS ALL IN? (Y OR N) /
1323	005326	044124	020105	042504	
1324	005334	040506	046125	020124	
1325	005342	052512	050115	051105	
1326	005350	020123	046101	020114	
1327	005356	047111	020077	054450	
1328	005364	047440	020122	024516	
1329	005372	020040	000		
1330	005375	377	051101	020105	MDSC: .ASCIZ <377>/ARE DSC 1 AND 2 BOTH IN? (Y OR N) /
1331	005402	051504	020103	020061	
1332	005410	047101	020104	020062	
1333	005416	047502	044124	044440	
1334	005424	037516	024040	020131	
1335	005432	051117	047040	020051	
1336	005440	000040			
1337					.EVEN
1338	005442	000002			XSTATQ: 2
1339	005444	006	003		.BYTE 6,3
1340	005446	001236			TEMP1
1341	005450	006	002		.BYTE 6,2
1342	005452	001240			TEMP2
1343					.EVEN
1344					TEMP: 0
1345	005454	000000			.=. +40
1346		005516			MDATA: 0
1347	005516	000000			.=. +40
1348		005560			INBUF: 0
1349	005560	000000			.=. +40
1350		005622			TRP.PC: .BLKW 1
1351	005622	000001			
1352					

```

1353
1354
1355
1356
1357
1358
1359
1360
1361
1362 005624 105737 001316      CYCLE: TSTB      DUPACTV      ;ARE ANY DUP11'S TO BE TESTED?
1363 005630 001004              BNE          1$      ;BR IF OK.
1364 005632 104402 004421      TYPE        ,MERR2  ;NO DUP11'S SELECTED!!
1365 005636 000000              HALT        ;STOP THE SHOW.
1366 005640 000776              BR          -2      ;DISQUALIFY CONT. SW.
1367 005642 012700 001500      1$: MOV      #DUP.MAP,R0 ;RESTORE POINTER.
1368 005646 012037 001412      MOV      (R0)+,RXCSR ;LOAD SYSTEM CTRL. REG
1369 005652 012037 001402      MOV      (R0)+,DUPRVC ;LOAD VECTOR
1370 005656 012037 001442      MOV      (R0)+,L00.00 ;GET PARAMETERS
1371 005662 012700 000002      MOV      #2,R0      ;SAVE CORE THIS WAY!
1372 005666 013737 001412 001432      MOV      RXCSR,HUPRCR ;GET CONTROL REG HIGH BYTE
1373 005674 005237 001432      INC      HUPRCR      ;GOT IT
1374 005700 013737 001432 001414      MOV      HUPRCR,RXDBUF ;GET RX CONTROL REG BUFFER
1375 005706 005237 001414      INC      RXDBUF      ;GOT IT
1376 005712 013737 001414 001424      MOV      RXDBUF,DUPSEC ;GOT SECONDARY REG SELECT REG
1377 005720 013737 001414 001416      MOV      RXDBUF,PARCSR ;GOT PARAMETER STATUS REGISTER
1378 005726 013737 001414 001430      MOV      RXDBUF,HUPRBF ;GET RX BUFFER HIGH BYTE
1379 005734 005237 001430      INC      HUPRBF      ;GOT IT
1380 005740 013737 001430 001426      MOV      HUPRBF,HUPPSR ;GOT PAR STATUS REG HIGH BYTE
1381 005746 013737 001426 001420      MOV      HUPPSR,TXCSR ;GET TX CONTROL REGISTER
1382 005754 005237 001420      INC      TXCSR      ;GOT IT
1383 005760 013737 001420 001436      MOV      TXCSR,HUPTCR ;GET TX CONTROL REG HIGH BYTE
1384 005766 005237 001436      INC      HUPTCR      ;GOT IT
1385 005772 013737 001436 001422      MOV      HUPTCR,TXDBUF ;BET TX BUFFER
1386 006000 005237 001422      INC      TXDBUF      ;GOT IT
1387 006004 013737 001422 001434      MOV      TXDBUF,HUPTBF ;GET TX BUFFER HIGH BYTE
1388 006012 005237 001434      INC      HUPTBF      ;GOT IT
1389
1390 006016 013737 001402 001404      MOV      DUPRVC,DUPRPS ;RX VECTOR
1391 006024 060037 001404      ADD      R0,DUPRPS   ;RX PRIORITY LEVEL
1392 006030 013737 001404 001406      MOV      DUPRPS,DUPTVC
1393 006036 060037 001406      ADD      R0,DUPTVC   ;TX VECTOR
1394 006042 013737 001406 001410      MOV      DUPTVC,DUPTPS
1395 006050 060037 001410      ADD      R0,DUPTPS   ;TX PRIORITY LEVEL
1396
1397
1398 006054 012700 001442      MOV      #L00.00,R0 ;LOAD STAU 00-00
1399 006060 012701 001440      MOV      #MASK.A,R1 ;PREPARE MASK.
1400 006064 012702 001441      MOV      #CLK.A,R2  ;PREPARE CLOCKS
1401 006070 004737 006234      JSR      PC,FIX.00  ;GO AND CALCULATE CONFIGURATION.
1402 006074 005737 000042      TST      @#42
1403 006100 001050              BNE          4$
1404 006102 032777 000002 173072      BIT      #SW01,@SWR ;IF SW01=1,GET STARTING TEST #
1405 006110 001444              BEQ          4$
1406 006112 104402 004352      7$: TYPE    ,MCRLF
1407 006116 104403      INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1408 006120 004613      MTSTN   ;MESSAGE

```

1409	006122	104405			PARAM	: CONVERT STRING	
1410	006124	000001			1	: LOW LIMIT	
1411	006126	001000			1000	: HIGH LIMIT	
1412	006130	001226			TSTNO	: STORE AT THIS LOCATION	
1413	006132	000			0	: MASK	
1414	006133	001			1	: HOW MANY TIMES + 2	
1415	006134	012700	006364		MOV	#TST1,R0	
1416	006140	022710	012737	5\$:	CMP	#12737,(R0)	
1417	006144	001017			BNE	6\$	
1418	006146	023760	001226	000002	CMP	TSTNO,2(R0)	
1419	006154	001013			BNE	6\$	
1420	006156	022760	001226	000004	CMP	#TSTNO,4(R0)	
1421	006164	001007			BNE	6\$	
1422	006166	010037	001214		MOV	R0,RETURN ;SAVE PC	
1423	006172	104402	004352		TYPE	,MCRLF	
1424	006176	104402	004416		TYPE	,MR	
1425	006202	000412			BR	8\$	
1426	006204	005720		6\$:	TST	(R0)+	
1427	006206	020027	010040		CMP	R0,#TLAST+10	
1428	006212	001352			BNE	5\$	
1429	006214	104402	004346		TYPE	,MQM	
1430	006220	000734			BR	7\$	
1431							
1432	006222	012737	006364	001214	4\$:	MOV	#TST1,RETURN ;PREPARE RETURN ADDRESS
1433	006230	000177	172760		8\$:	JMP	@RETURN ;GO START TESTING.
1434							
1435	006234	011003			FIX.00:	MOV	(R0),R3 ;GET PARAMETERS.
1436	006236	000207			5\$:	RTS	PC ;


```

1437                                     ;THIS ROUTINE PICKS UP THE ADDRESS OF
1438                                     ;THE JUMPER TABLE AND LOADS R5 WITH
1439                                     ;THE CORRECT DATA BASED ON THE STATE
1440                                     ;OF THE JUMPER AND CONNECTOR FLAGS.
1441                                     ;-----
1442
1443 006240 012100          JUMPER: MOV      (R1)+,R0          ;GET THE TABLE ADDRESS
1444 006242 105737 001324  TSTB      TCNFLG          ;TEST THE TURN AROUND CONNECTOR FLAG
1445 006246 001406          BEQ       2$              ;BRANCH IF CONNECTOR IS MISSING
1446 006250 105737 001325  TSTB      OPCLRJ          ;TEST CLEAR JUMPER FLAG
1447 006254 001403          BEQ       2$              ;BRANCH IF JUMPER IS MISSING
1448 006256 011005          MOV       (R0),R5          ;MOVE THE DATA TO R5, BOTH JUMPER
1449                                     ;AND CONNECTOR ARE THERE
1450 006260 000137 006304          JMP       5$
1451 006264 022020          2$:      CMP      (R0)+,(R0)+      ;POP POINTER
1452 006266 105737 001325  TSTB      OPCLRJ          ;TEST CLEAR JUMPER FLAG
1453 006272 001403          BEQ       3$              ;BRANCH IF MISSING
1454 006274 011005          MOV       (R0),R5          ;MOVE DATA- JUMPER IN, CONNECTOR OFF
1455 006276 000137 006304          JMP       5$
1456 006302 012005          3$:      MOV      (R0)+,R5          ;NO CONNECTOR OR JUMPER
1457 006304 000201          5$:      RTS       R1              ;RETURN
1458
1459 006306 012100          O.JUMPER:MOV     (R1)+,R0          ;GET THE POINTER ADDRESS
1460 006310 105737 001324  TSTB      TCNFLG          ;CHECK FOR TURNAROUND CONNECTOR
1461 006314 001403          BEQ       4$              ;BR IF MISSING
1462 006316 011005          MOV       (R0),R5          ;MOVE THE INFO TO R5
1463 006320 000137 006330          JMP       6$
1464 006324 022020          4$:      CMP      (R0)+,(R0)+      ;POP POINTER
1465 006326 011005          MOV       (R0),R5          ;LOAD DATA TO R5
1466 006330 000201          6$:      RTS       R1              ;RETURN
1467
1468
1469                                     ;ROUTINE TO SET UP INTERRUPT VECTORS
1470 006332 012577 173044  SETVEC: MOV     (R5)+,@DUPRVC
1471 006336 012577 173044  MOV       (R5)+,@DUPTVC
1472 006342 112577 173036  MOVB     (R5)+,@DUPRPS
1473 006346 112577 173036  MCVB     (R5)+,@DUPTPS
1474 006352 000205          RTS       R5
1475 006354          NO.ATRAP:
1476 006354 104001          HLT      1
1477 006356 000002          RTI
1478
1479          NO.BTRAP:
1480 006360 104002          HLT      2
1481 006362 000002          RTI
1482

```

1483
 1484
 1485
 1486
 1487
 1488
 1489
 1490
 1491
 1492
 1493
 1494
 1495
 1496
 1497
 1498
 1499
 1500
 1501
 1502
 1503
 1504
 1505
 1506
 1507
 1508
 1509
 1510
 1511
 1512
 1513
 1514
 1515
 1516
 1517
 1518
 1519
 1520
 1521
 1522
 1523
 1524
 1525
 1526
 1527
 1528
 1529
 1530
 1531
 1532
 1533
 1534
 1535
 1536
 1537
 1538

006364 012737 000001 001226
 006372 012737 006744 001216
 006400 012746 173017
 006404 012746 173001
 006410 012746 001016
 006414 012746 000000
 006420 105737 001324
 006424 001532
 006426 105737 001344
 006432 001034
 006434 012701 002010
 006440 105737 001336
 006444 001007
 006446 040137 006734
 006452 040137 006736
 006456 040137 006740
 006462 000411
 006464 105737 001340
 006470 001006
 006472 040137 006734
 006476 040137 006736
 006502 040137 006740
 006506 105737 001346
 006512 001004
 006514 005337 006734
 006520 005337 006736
 006524 004137 005306
 006530 006734
 006532 005077 172654
 006536 052777 000400 172654
 006544 004737 004306
 006550 013703 001412
 006554 052777 010000 172636
 006562 052713 000016
 006566 012737 000110 006616
 006574 032777 004000 172620
 006602 001374
 006604 032777 004000 172610
 006612 001774
 006614 005327

```

***** TEST 1 *****
*THIS TEST PROVES THE INTERACTION OF DTR!RTS!STD
*WITH RING,DSR,CTS,CARDET,STD,SRD
*AND DATA SET CHANGE ONE AND DATA SET CHANGE TWO.
*SET THE BIT AND VERIFY THE OTHER BITS ARE SET. CLEAR
*THE BIT AND VERIFY CLEAR. REPEAT FOR MRESET.
*****

:*****
:
: TEST 1
:
:*****
:*****
TST1:  MOV    #1,@TSTNO
      MOV    #TST2,NEXT
      MOV    #173017,-(SP)          ;SAVE
      MOV    #173001,-(SP)
      MOV    #1016,-(SP)
      MOV    #0,-(SP)
      TSTB   TCNFLG                ;H325 TEST CONNECTOR ON?
      BEQ    6$                    ;IF NO, SKIP TEST.
100$:  TSTB   ALJMFL
      BNE   12$
      MOV    #STD!SRD,R1
      TSTB   STJMFL
      BNE   101$
      BIC   R1,7$
      BIC   R1,7$+2
      BIC   R1,7$+4
      BR    102$
101$:  TSTB   SRJMFL
      BNE   102$
      BIC   R1,7$
      BIC   R1,7$+2
      BIC   R1,7$+4
102$:  TSTB   DSCFLG
      BNE   12$
      DEC   7$
      DEC   7$+2
12$:   JSR   R1,OJUMPER            ;THIS CALL DETERMINES IF TURNAROUND CONNECTOR
                                       ;AND OPTIONAL JUMPER ARE USED
                                       ;AND LOADS R5 (EXPECTED) ACCORDINGLY.
      CLR   @RXCSR
      BIS   #MRESET,@TXCSR        ;RESET THE DEVICE
      JSR   PC,SMALL              ;WAIT FOR RESET TO FINISH
      MOV   RXCSR,R3              ;LOAD THE RECEIVER CONTROL REGISTER TO R3.
      BIS   #MEXT,@TXCSR          ;ENTER EXTERNAL MAINT. MODE
      BIS   #DTR!RTS!STD,(R3)     ;TURN ON DTR!RTS!STD
      MOV   #110,68$              ;LOAD THE NUMBER
      BIT   #TIMER,@TXDBUF        ;CHECK THE TIMER BIT
      BNE   66$                    ;BR IF SET
      BIT   #TIMER,@TXDBUF        ;CHECK THE BIT
      BEQ   67$                    ;BR IF CLEAR
      DEC   (PC)+                 ;DECREMENT THE NUMBER
  
```

```
1539 006616 000110      68$: 110      ;OF TIMES TO REPEAT
1540 006620 001365      BNF      66$      ;BR IF MORE TO GO
1541 006622 011304      MOV      (R3),R4  ;GET THE BITS FROM THE RXCSR
1542 006624 020504      CMP      R5,R4   ;R5=GOOD R4=?
1543 006626 001401      BEQ      2$      ;BRANCH IF THEY MATCH
1544 006630 104003      HLT      3       ;NO MATCH - SHOW OPR.
1545 006632 012737 006644 001220 2$:  MOV      #3$,LOCK ;SWO9 SETUP
1546 006640 042705 073016      BIC      #RING!CTS!CARDET!SRD!DSR!STD!RTS!DTR,R5 ;CLEAR OUT UNWANTED BITS
1547 006644 005013      CLR      (R3)   ;CLEAR OUT THE REGISTER
1548 006646 012737 000005 006676 3$:  MOV      #5,73$  ;LOAD THE NUMBER
1549 006654 032777 004000 172540 71$: BIT      #TIMER,@TXDBUF ;CHECK T; TIMER BIT
1550 006662 001374      BNE      71$    ;BR IF SET
1551 006664 032777 004000 172530 72$: BIT      #TIMER,@TXDBUF ;CHECK THE BIT
1552 006672 001774      BEQ      72$    ;BR IF CLEAR
1553 006674 005327      DEC      (PC)+  ;DECREMENT THE NUMBER
1554 006676 000005      73$: 5         ;OF TIMES TO REPEAT
1555 006700 001365      BNE      71$    ;BR IF MORE TO GO
1556 006702 011304      MOV      (R3),R4 ;READ BACK THE REGISTER
1557 006704 020504      CMP      R5,R4  ;R5=GOOD R4=?
1558 006706 001401      BEQ      6$     ;BRANCH IF ONLY THE DSC BITS ARE SET
1559 006710 104003      HLT      3     ;NO-GO TELL OPR
1560 006712      6$:
1561 006712 012637 006742      MOV      (SP)+,7$+6 ;RESTORE
1562 006716 012637 006740      MOV      (SP)+,7$+4
1563 006722 012637 006736      MOV      (SP)+,7$+2
1564 006726 012637 006734      MOV      (SP)+,7$
1565 006732 104400      SCOPE      ;SCOPE THE WHOLE TEST
1566 006734 173017      7$: .WORD 173017
1567 006736 173001      .WORD 173001
1568 006740 001016      .WORD 1016
1569 006742 000000      .WORD 0
```

```
:***** TEST 2 *****
:*TEST OF THE DUP RUNNING A BINARY COUNT
:*PATTERN WITH A CRC CALCULATION AS A SECONDARY STATION
:*****
```

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

```
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583 006744 012737 000002 001226 TST2: MOV      #2,@TSTNO
1584 006752 012737 010030 001216      MOV      #TST3,NEXT
1585 006760 052777 000400 172432      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
1586 006766 004737 004306      JSR      PC,SMALL ;WAIT FOR RESET TO FINISH
1587 006772 105737 001324      TSTB    TCNFLAG ;H325 TEST CONNECTOR ON?
1588 006776 001403      BEQ      102$   ;IF NO, BR.
1589 007000 012737 010000 007104      MOV      #MEXT,103$+2 ;IF YES, SET MAINT
1590 007006 012737 000005 001244 102$: MOV      #5,TEMP4
1591 007014 005001      CLR      R1     ;CLEAR OUT DATA
1592 007016 012737 102010 010022      MOV      #CRC.CCITT,XPOLY ;SET UP THE POLYNOMIAL
1593 007024 012737 177777 010026      MOV      #-1,CALBCC ;SETUP FOR THE FIRST TIME
1594 007032 013737 010026 007054 16$: MOV      CALBCC,20$ ;ALLOW FOR THE NEXT CHARACTER
```

```

1595 007040 010137 007052      MOV      R1,17$      ;LOAD DATA
1596 007044 004537 007650      JSR      R5,SIMBCC   ;GO CALCULATE SOFTWARE BCC
1597 007050 000010                8.          ;BASED ON THESE PARAMETERS
1598 007052 000001      17$: .BLKW 1 ;DATA
1599 007054 000001      20$: .BLKW 1 ;PREVIOUS BCC
1600 007056 105201      INCB     R1          ;INCREMENT DATA
1601 007060 001364      BNE     16$         ;BR IF MORE TO GO
1602 007062 012737 000001 001236      MOV      #1,TEMP1   ;LOAD DATA
1603 007070 005037 001240      CLR     TEMP2       ;CLEAR EXPECTED
1604 007074 012737 000340 177776      MOV      #340,PS    ;PS = 7
1605 007102 052777 004000 172310 103$: BIS      #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
1606 007110 004537 006332      JSR      R5,SETVEC  ;LOAD INTERRUPT VECTORS
1607 007114 007404      11$          ;RECEIVER
1608 007116 007452      12$          ;TRANSMITER
1609 007120      340      340      .BYTE 340,340      ;LEVEL
1610
1611      ;*****
1612      ;**FOLLOWING CODE FOR TESTING DUP11 CONNECTED TO MODEM WITH ANALOG LOOPBACK ENABLED**
1613      ;*****
1613 007122 105737 001342      TSTB    MDMFLG      ;MODEM CONNECTED?
1614 007126 001430      BEQ     104$        ;IF NO, BR.
1615 007130 013703 001310      MOV     REPEAT,R3   ;REPEAT CTS DLY
1616 007134 012777 010000 172256      MOV     #MEXT,@TXCSR ;SET EXTERNAL MAINTENANCE
1617 007142 013737 001314 007374      MOV     DELAY,73$   ;SET TIMER
1618 007150 052777 000006 172234      BIS     #DTR!RTS,@RXCSR ;ENABLE MODEM.
1619 007156 013702 001312      MOV     CTS DLY,R2  ;WAIT FOR CTS
1620 007162 032777 020000 172222 106$: BIT     #CTS,@RXCSR ;CLEAR TO SEND UP?
1621 007170 001007      BNE     104$        ;IF YES, BR.
1622 007172 005302      DEC     R2          ;STEP COUNTER IF NO.
1623 007174 001372      BNE     105$        ;IF NO TRY AGAIN.
1624 007176 104006      HLT     6
1625 007200 005303      DEC     R3          ;ALLOW CTS DLY TO REPEAT
1626 007202 001365      BNE     106$        ;BR IF NOT COMPLETE
1627 007204 104007      HLT     7          ;CTS STILL NOT ACTIVE
1628 007206 000475      BR      6$         ;DO NOT CONTINUE TEST
1629
1630 007210 052777 000020 172174 104$: BIS     #RCVEN,@RXCSR ;TURN ON THE RECEIVER
1631 007216 052777 000100 172166      BIS     #RINTEN,@RXCSR ;TURN ON REC INTERRUPT ENABLE
1632 007224 105777 172170      1$: TSTB    @TXCSR   ;TEST FOR TX DONE
1633 007230 100375      BPL     1$         ;BR IF NOT SET
1634 007232 052777 000020 172160 2$: BIS     #SEND,@TXCSR ;TURN ON SEND
1635 007240 012777 000400 172154      MOV     #TSOM,@TXDBUF ;TURN ON START OF MESSAGE
1636 007246
1637 007246 012737 000005 007276 101$: MOV     #5,68$      ;LOAD THE NUMBER
1638 007254 032777 004000 172140 66$: BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
1639 007262 001374      BNE     66$        ;BR IF SET
1640 007264 032777 004000 172130 67$: BIT     #TIMER,@TXDBUF ;CHECK THE BIT
1641 007272 001774      BEQ     67$        ;BR IF CLEAR
1642 007274 005327      DEC     (PC)+      ;DECREMENT THE NUMBER
1643 007276 000005      68$: 5           ;OF TIMES TO REPEAT
1644 007300 001365      BNE     66$        ;BR IF MORE TO GO
1645 007302 005337 001244      DEC     TEMP4
1646 007306 001001      BNE     3$         ;
1647 007310 104004      HLT     4
1648 007312 105777 172102      3$: TSTB    @TXCSR   ;WAIT FOR DONE
1649 007316 100353      BPL     101$       ;BR IF NOT SET
1650 007320 005077 172076      4$: CLR     @TXDBUF  ;PUSH OUT DATA

```

```

1651 007324 052777 000100 172066      BIS    #TXINTE,@TXCSR ;TURN ON TRANSMITTER INT ENABLE
1652 007332 005037 177776      CLR    PS             ;LOWER PROCESOR STATUS
1653 007336
1654
1655 007336 105737 001342      TSTB   MDMFLG        ;MODEM FLAG SET?
1656 007342 001003      BNE    71$           ;IF YES, BR
1657
1658 007344 012737 000040 007374      MOV    #32.,73$     ;LOAD THE NUMBER
1659 007352 032777 004000 172042      71$:   BIT    #TIMER,@TXDBUF ;CHECK THE TIMER BIT
1660 007360 001374      BNE    71$           ;BR IF SET
1661 007362 032777 004000 172032      72$:   BIT    #TIMER,@TXDBUF ;CHECK THE BIT
1662 007370 001774      BEQ    72$           ;BR IF CLEAR
1663 007372 005327      DEC    (PC)+        ;DECREMENT THE NUMBER
1664 007374 000040      73$:   32.           ;OF TIMES TO REPEAT
1665 007376 001365      BNE    71$           ;BR IF MORE TO GO
1666 007400 104004      HLT    4             ;FAILED TO INTERRUPT IN TIME
1667 007402 104400      6$:   SCOPE          ;SCOPE THIS TEST
1668
1669
1670      :INTERRUPT SERVICE ROUTINES
1671      :-----
1672      :RECEIVER:
1672 007404 017737 172004 001326      11$:   MOV    @RXDBUF,DATA ;GET THE REGISTER AND DATA
1673 007412 123737 001240 001326      CMPB   TEMP2,DATA    ;CHECK IT
1674 007420 001401      BEQ    .+4          ;BR IF OK
1675 007422 104004      HLT    4            ;COMPARISON ERROR
1676 007424 105237 001240      INCB   TEMP2        ;COUNT UP EXPECTED
1677 007430 105737 001240      TSTB   TEMP2        ;CHECK TO SEE IF DONE
1678 007434 001005      BNE    7$           ;BR IF NO
1679 007436 004537 006332      JSR    R5,SETVEC    ;YES--RESET THE VECTORS
1680 007442 007556      14$
1681 007444 007452      12$
1682 007446      340      340      .BYTE 340,340      ;RECEIVER
1683 007450 000002      7$:   RTI            ;TRANSMITTER
1684
1685 007452 113777 001236 171742      12$:   MOVB   TEMP1,@TXDBUF ;LOAD THE TRANSMITTER BUFFER
1686 007460 105237 001236      INCB   TEMP1        ;UP THE COUNT
1687 007464 122737 000377 001236      CMPB   #377,TEMP1   ;ARE WE DONE
1688 007472 001026      BNE    13$         ;BR IF NO
1689 007474 012777 007504 171704      MOV    #21$,@DUPTVC ;SETUP FOR NEXT PART
1690 007502 000422      BR    13$         ;LEAVE
1691 007504 012777 000377 171710      21$:   MOV    #377,@TXDBUF ;LOAD BUFFER
1692 007512 012777 007522 171666      MOV    #22$,@DUPTVC ;SETUP NEXT PART
1693 007520 000413      BR    13$         ;LEAVE
1694 007522 012777 001000 171672      22$:   MOV    #TEOM,@TXDBUF ;SET END OF MSG
1695 007530 000240      NOP
1696 007532 000240      NOP
1697 007534 042777 000120 171656      BIC    #SEND!TXINTE,@TXCSR ;TURN OFF TRANSMITTER
1698 007542 012777 006360 171636      MOV    #NO.BTRAP,@DUPTVC ;LOAD VECTOR
1699 007550 012716 007336      13$:   MOV    #5$, (SP)    ;CRUNCH STACK
1700 007554 000002      RTI            ;RETURNS
1701
1702      :CRC CATCH INT SVC
1702 007556 117737 171632 001326      14$:   MOVB   @RXDBUF,DATA ;GET FIRST PART OF CRC
1703 007564 105777 171622      TSTB   @RXCSR       ;WAIT FOR SECOND PART
1704 007570 100375      BPL    .-4          ;DITTO
1705 007572 117737 171616 001327      MOVB   @RXDBUF,DATA+1 ;GET THE REST OF THE CRC
1706 007600 012716 007606      MOV    #15$, (SP)   ;SETUP FOR RETURN
  
```

```

1707 007604 000002      RTI          :RETURN
1708 007606 012737 000340 177776 15$: MOV      #340,PS  :RAISE PS
1709 007614 005137 010026      COM      CALBCC    :INVERT BCC
1710 007620 023737 010026 001326  CMP      CALBCC,DATA :COMPARE SOFTWARE AND HARDWARE BCC
1711 007626 001401      BEQ      .+4      :BR IF OK
1712 007630 104004      HLT      4        :BCC COMPARISON ERROR
1713 007632 052777 000400 171560  BIS      #MRESET,@TXCSR :RESET THE DEVICE
1714 007640 004737 004306      JSR      PC,SMALL  :WAIT FOR RESET TO FINISH
1715 007644 000137 007402      JMP      6$      :LEAVE
1716
1717

```

```

1718 007650 010046      SIMBCC: MOV      R0,-(SP)
1719 007652 010146      MOV      R1,-(SP)
1720 007654 010246      MOV      R2,-(SP)
1721 007656 012537 001236  MOV      (R5)+,TEMP1
1722 007662 012537 001240  MOV      (R5)+,TEMP2
1723 007666 012537 001242  MOV      (R5)+,TEMP3
1724 007672 005037 010024  1$: CLR      BCCFBK
1725 007676 013700 001242  MOV      TEMP3,R0
1726 007702 006037 001240  ROR      TEMP2
1727 007706 005500      ADC      R0
1728 007710 032700 000001  BIT      #BIT0,R0
1729 007714 001402      BEQ      2$
1730 007716 005137 010024  COM      BCCFBK
1731 007722 013700 010022  2$: MOV      XPOLY,R0
1732 007726 005100      COM      R0
1733 007730 040037 010024  BIC      R0,BCCFBK
1734 007734 000241      CLC
1735 007736 006037 001242  ROR      TEMP3
1736 007742 013700 010024  MOV      BCCFBK,R0
1737 007746 013701 001242  MOV      TEMP3,R1
1738 007752 010102      MOV      R1,R2
1739 007754 040100      BIC      R1,R0
1740 007756 043702 010024  BIC      BCCFBK,R2
1741 007762 050200      BIS      R2,R0
1742 007764 043737 010022 001242  BIC      XPOLY,TEMP3
1743 007772 050037 001242      BIS      R0,TEMP3
1744 007776 005337 001236      DEC      TEMP1
1745 010002 001333      BNE      1$
1746 010004 013737 001242 010026  MOV      TEMP3,CALBCC
1747 010012 012602      MOV      (SP)+,R2
1748 010014 012601      MOV      (SP)+,R1
1749 010016 012600      MOV      (SP)+,R0
1750 010020 000205      RTS      R5

```

```

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

```

```

1751 010022 000000
1752 010024 000000
1753 010026 000000
1754      120001
1755      102010
1756
1757
1758 :***** TEST 3 *****
1759 :*THIS TEST PROVES THE DEVICE WILL HANDLE THE
1760 :*DDCMP PROTOCOL. SEND AND RECEIVE SYNCs.
1761 :*FOLLOWED BY DATA,BCC,DATA AND FINAL BCC.
1762 :*****

```

```
1763
1764
1765
1766
1767
1768
1769
1770 010030 012737 000003 001226 TST3: MOV #3,@TSTNO
1771 010036 012737 002352 001216 MOV #.EOP,NEXT
1772 010044 105737 001324 TSTB TCNFLG ;H325 TEST CONNECTOR ON?
1773 010050 001403 BEQ 101$ ;IF NO, BRF
1774 010052 012737 010000 010146 MOV #MEXT,102$+2 ;IF YES, SET MAINTENANCE
1775 010060 012737 000340 177776 101$: MOV #340,PS ;RAISE PROCESSOR STATUS
1776 010066 004537 006332 JSR R5,SETVEC ;SET UP VECTORS
1777 010072 010632 10$ ;BASED ON
1778 010074 010454 2$ ;THESE
1779 010076 340 340 .BYTE 340,340 ;PARAMETERS
1780 010100 005037 001236 CLR TEMP1
1781 010104 005037 001240 CLR TEMP2
1782 010110 005037 001242 CLR TEMP3
1783 010114 005037 001244 CLR TEMP4
1784 010120 005037 001246 CLR TEMP5
1785 010124 052777 000400 171266 BIS #MRESET,@TXCSR ;RESET THE DEVICE
1786 010132 004737 004306 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
1787 010136 012777 100026 171252 MOV #DECMOD!26,@PARCSR ;LOAD THE MODE AND SYNC CHARACTER
1788 010144
1789 010144 052777 004000 171246 102$: BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
1790
1791
1792
1793 010152 105737 001342
1794 010156 001430
1795 010160 013703 001310
1796 010164 012777 010000 171226
1797 010172 013737 001314 010426
1798 010200 052777 000006 171204
1799 010206 013702 001312 105$: MOV CTSDLY,R2 ;WAIT FOR CTS
1800 010212 032777 020000 171172 104$: BIT #CTS,@RXCSR ;CLEAR TO SEND UP?
1801 010220 001007 BNE 103$ ;IF YES, BR.
1802 010222 005302 DEC R2 ;STEP COUNT IF NO.
1803 010224 001372 BNE 104$ ;TRY AGAIN.
1804 010226 104006 HLT 6
1805 010230 005303 DEC R3 ;ALLOW CTSDLY TO REPEAT
1806 010232 001365 BNE 105$ ;BR IF NOT COMPLETE
1807 010234 104007 HLT 7 ;CTS STILL NOT ACTIVE
1808 010236 000476 BR 1$ ;DO NOT CONTINUE TEST.
1809
1810 010240
1811 010240 052777 000420 171144 103$: BIS #RCVEN!STPSYN,@RXCSR ;LOAD RCVEN!STPSYN
1812 010246 052777 000020 171144 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
1813 010254 012777 000426 171140 MOV #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
1814 010262 012737 000005 010312 MOV #5,68$ ;LOAD THE NUMBER
1815 010270 032777 004000 171124 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
1816 010276 001374 BNE 66$ ;BR IF SET
1817 010300 032777 004000 171114 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
1818 010306 001774 BEQ 67$ ;BR IF CLEAR
```

```

1819 010310 005327          DEC      (PC)+      ;DECREMENT THE NUMBER
1820 010312 000005      68$:      5          ;OF TIMES TO REPEAT
1821 010314 001365          BNE      66$          ;BR IF MORE TO GO
1822 010316 105777 171076  TSTB    @TXCSR
1823 010322 100401          BMI      +4
1824 010324 104005          HLT      5          ;EXTERNAL CLOCKING STOPPED
1825 010326 012777 000426 171066  MOV     #TSOM!26,@TXDBUF
1826 010334 105777 171060  69$:    TSTB    @TXCSR      ;CHECK DONE
1827 010340 100375          BPL     69$          ;BR IF NOT SET
1828 010342 012777 000426 171052  MOV     #TSOM!26,@TXDBUF ;SEND SYNC
1829 010350 052777 000100 171034  BIS    #RINTEN,@RXCSR ;TURN ON INTERRUPTS
1830 010356 052777 000100 171034  BIS    #TXINTE,@TXCSR ;DITTO
1831 010364 005037 177776  CLR     PS          ;LOWER PROCESSOR STATUS
1832 010370
1833 010370 105737 001342 100$:    ;:*****
1834 010370 105737 001342          TSTB    MDMFLG      ;MODEM FLAG SET?
1835 010374 001003          BNE     72$          ;IF YES, BR
1836 010374 001003          ;:*****
1837 010376 012737 000144 010426  MOV     #100,74$      ;LOAD THE NUMBER
1838 010404 032777 004000 171010  72$:    BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
1839 010412 001374          BNE     72$          ;BR IF SET
1840 010414 032777 004000 171000  73$:    BIT     #TIMER,@TXDBUF ;CHECK THE BIT
1841 010422 001774          BEQ     73$          ;BR IF CLEAR
1842 010424 005327          DEC     (PC)+      ;DECREMENT THE NUMBER
1843 010426 000144          74$:    100.        ;OF TIMES TO REPEAT
1844 010430 001365          BNE     72$          ;BR IF MORE TO GO
1845 010432 104005          HLT     5          ;FAILED TO FINISH TEST
1846 010434
1847 010434 052777 000400 170756  1$:    BIS    #MRESET,@TXCSR ;RESET THE DEVICE
1848 010442 004737 004306  JSR    PC,SMALL     ;WAIT FOR RESET TO FINISH
1849 010446 012706 001150  MOV    #STACK,SP   ;RESET THE STACK
1850 010452 104400          SCOPE ;SCOPE THIS TEST
1851
1852          ;INTERRUPT SERVICE ROUTINES
1853          ;TRANSMITTER
1854
1855 010454 012777 000252 170740  2$:    MOV    #252,@TXDBUF ;LOAD FIRST DATA CHAR
1856 010462 012737 000026 001236  MOV    #26,TEMP1   ;LOAD DATA
1857 010470 012777 010500 170710  MOV    #3$,@DUPTVC ;RELOAD VECTOR
1858 010476 000452          BR     7$          ;LEAVE
1859 010500 013777 001236 170714  3$:    MOV    TEMP1,@TXDBUF ;MOV DATA TO BUFFER
1860 010506 105237 001236  INCB   TEMP1       ;UPDATE DATA
1861 010512 122737 000032 001236  CMPB  #32,TEMP1   ;CHECK FOR DONE
1862 010520 001041          BNE    7$          ;BR IF MORE TO SEND
1863 010522 012777 010532 170656  MOV    #4$,@DUPTVC ;RELOAD VECTOR
1864 010530 000435          BR     7$          ;RETURN
1865 010532 012777 001000 170662  4$:    MOV    #TEOM,@TXDBUF ;PUT OUT BCC
1866 010540 012777 010550 170640  MOV    #5$,@DUPTVC ;RELOAD VECTOR
1867 010546 000426          BR     7$          ;RETURN
1868 010550 013777 001240 170644  5$:    MOV    TEMP2,@TXDBUF ;LOAD DATA
1869 010556 105237 001240  INCB   TEMP2       ;UPDATE DATA
1870 010562 122737 000100 001240  CMPB  #100,TEMP2  ;CHECK FOR FINISH
1871 010570 001015          BNE    7$          ;BR IF MORE TO GO
1872 010572 012777 010602 170606  MOV    #6$,@DUPTVC ;RELOAD VECTOR
1873 010600 000411          BR     7$          ;RETURN
1874 010602 012777 001000 170612  6$:    MOV    #TEOM,@TXDBUF ;PUSH OUT DATA BCC
  
```



```

1875 010610 042777 000120 170602      BIC      #SEND!TXINTE,@TXCSR      ;SHUT DOWN TRANSMITTER
1876 010616 012777 006360 170562      MOV      #NO.BTRAP,@DUPTVC      ;RESET VECTOR
1877 010624 012716 010370      7$:     MOV      #100$, (SP)           ;SETUP RETURN
1878 010630 000002      RTI      ;RETURN
1879
1880
1881      ;RECEIVER
1882 010632 017737 170554 001242 10$:     MOV      @RXCSR,TEMP3           ;SAVE CSR
1883 010640 017737 170550 001244      MOV      @RXDBUF,TEMP4         ;SAVE BUFFER
1884 010646 105737 001242      TSTB    TEMP3                 ;CHECK FOR DONE
1885 010652 100401      BMI     11$                   ;BR IF SET
1886 010654 104005      HLT     5                     ;FALSE INTERRUPT
1887 010656 005737 001244      11$:     TST      TEMP4                 ;CHECK FOR ERROR
1888 010662 100001      BPL     12$                   ;BR IF NO ERROR
1889 010664 104005      HLT     5                     ;RECEIVER ERROR
1890 010666 122737 000252 001244 12$:     CMPB    #252,TEMP4            ;CHECK DATA
1891 010674 001401      BEQ     13$                   ;BR IF A MATCH
1892 010676 104005      HLT     5                     ;DATA COMPARE ERROR
1893 010700 012737 000026 001246 13$:     MOV      #26,TEMP5             ;LOAD NEXT EXPECTED
1894 010706 012777 010716 170466      MOV      #14$,@DUPRVC         ;RELOAD VECTOR
1895 010714 000531      BR      26$                   ;LEAVE
1896 010716 017737 170472 001244 14$:     MOV      @RXDBUF,TEMP4         ;GET DATA
1897 010724 005737 001244      TST     TEMP4                 ;CHECK FOR ERROR
1898 010730 100001      BPL     15$                   ;BR IF NO ERROR
1899 010732 104005      HLT     5                     ;DATA ERROR
1900 010734 123737 001246 001244 15$:     CMPB    TEMP5,TEMP4           ;CHECK DATA
1901 010742 001401      BEQ     16$                   ;BR IF A MATCH
1902 010744 104005      HLT     5                     ;DATA COMPARE ERROR
1903 010746 105237 001246      INCB    TEMP5                 ;UPDATE DATA
1904 010752 122737 000032 001246 16$:     CMPB    #32,TEMP5            ;CHECK FOR FIRST PART FINISH
1905 010760 001107      BNE     26$                   ;BR IF MORE TO GO
1906 010762 012777 010772 170412      MOV      #17$,@DUPRVC         ;SET UP NEXT VECTOR
1907 010770 000503      BR      26$                   ;LEAVE
1908 010772 017737 170416 001244 17$:     MOV      @RXDBUF,TEMP4         ;GET THE BUFFER
1909 011000 005737 001244      TST     TEMP4                 ;TEST FOR ERROR
1910 011004 100001      BPL     .+4                   ;BR IF OK
1911 011006 104005      HLT     5                     ;RECEIVER ERROR
1912 011010 012777 011020 170364      MOV      #18$,@DUPRVC         ;RELOAD THE VECTOR
1913 011016 000470      BR      26$                   ;LEAVE
1914 011020 017737 170370 001326 18$:     MOV      @RXDBUF,DATA         ;GET DATA
1915 011026 032737 010000 001326      BIT     #CRCERR,DATA          ;CHECK FOR CRC ERROR
1916 011034 001001      BNE     19$                   ;BR IF OK
1917 011036 104005      HLT     5                     ;CRC ERROR!!!!!!
1918 011040 012777 011054 170334 19$:     MOV      #20$,@DUPRVC         ;SET UP VECTOR
1919 011046 005037 001332      CLR     MIND                  ;SETUP FOR NEXT DATA
1920 011052 000452      BR      26$                   ;LEAVE
1921 011054 017737 170334 001244 20$:     MOV      @RXDBUF,TEMP4         ;GET DATA
1922 011062 005737 001244      TST     TEMP4                 ;CHECK FOR ERROR
1923 011066 100001      BPL     21$                   ;BR IF NO ERROR
1924 011070 104005      HLT     5                     ;RECEIVER ERROR
1925 011072 123737 001332 001244 21$:     CMPB    MIND,TEMP4           ;CHECK DATA
1926 011100 001401      BEQ     22$                   ;BR IF A MATCH
1927 011102 104005      HLT     5                     ;DATA ERROR
1928 011104 105237 001332      INCB    MIND                  ;UPDATE SOFTWARE DATA
1929 011110 122737 000100 001332 22$:     CMPB    #100,MIND            ;CHECK FOR FINISH
1930 011116 001030      BNE     26$                   ;BR IF MORE TO GO

```

1931	011120	012777	011130	170254		MOV	#23\$,@DUPRVC	:RELOAD FINAL VECTOR
1932	011126	000424				BR	26\$:LEAVE
1933	011130	017737	170260	001244	23\$:	MOV	@RXDBUF,TEMP4	:GET DATA
1934	011136	005737	001244			TST	TEMP4	:CHECK FOR ERROR
1935	011142	100001				BPL	24\$:BR IF OK
1936	011144	104005				HLT	5	:RECEIVER ERROR ON FIRST OCTET
1937								:OF SECOND BCC
1938	011146	105777	170240		24\$:	TSTB	@RXCSR	:TEST DONE
1939	011152	100375				BPL	24\$:BR IF NOT SET
1940	011154	017737	170234	001326		MOV	@RXDBUF,DATA	:GET SECOND BCC OCTET
1941	011162	032737	010000	001326		BIT	#CRCERR,DATA	:CHECK FOR BCC ERROR
1942	011170	001001				BNE	25\$:BR IF OK
1943	011172	104005				HLT	5	:BCC ERROR ON SECOND PART OF MSG
1944	011174	012716	010434		25\$:	MOV	#1\$, (SP)	:SETUP TO FINISH TEST
1945	011200	000002			26\$:	RTI		:RETURN
1946								
1947								

```
1948 011202 052777 042516 050130 EM1: .ASCIZ <377>/UNEXPECTED INTERRUPT ON VECTOR 'A' /
(1) 011247 377 047125 054105 EM2: .ASCIZ <377>/UNEXPECTED INTERRUPT ON VECTOR 'B' /
(1) 011314 042777 051122 051117 EM3: .ASCIZ <377>/ERROR WHEN USING MODEM LEADS /
(1) 011353 377 051105 047522 EM4: .ASCIZ <377>/ERROR IN SDLC /
(1) 011373 377 051105 047522 EM5: .ASCIZ <377>/ERROR IN DECMODE /
(1) 011416 051377 047125 042040 EM6: .ASCIZ <377>/RUN DIAGNOSTICS /
(1) 011440 041777 042514 051101 EM7: .ASCIZ <377>/CLEAR TO SEND NOT ACTIVE/
(1) 011472 041777 042514 051101 EM8: .ASCIZ <377>/CLEAR TO SEND STILL NOT ACTIVE-TEST ABORTED/
(1) 011550 .EVEN
(1) 011550 .ERRTAB:
(1) 011550 000000 0
(1) 011552 000000 0
(1) 011554 000000 0
(1) 011556 011202 EM1
(1) 011560 011416 EM6 ;HALT 1
(1) 011562 000000 0
(1) 011564 011247 EM2
(1) 011566 011416 EM6 ;HALT 2
(1) 011570 000000 0
(1) 011572 011314 EM3
(1) 011574 011416 EM6 ;HALT 3
(1) 011576 000000 0
(1) 011600 011353 EM4
(1) 011602 011416 EM6 ;HALT 4
(1) 011604 000000 0
(1) 011606 011373 EM5
(1) 011610 011416 EM6 ;HALT 5
(1) 011612 000000 0
(1) 011614 011440 EM7
(1) 011616 000000 0 ;HALT 6
(1) 011620 000000 0
(1) 011622 011472 EM8
(1) 011624 000000 0 ;HALT 7
(1) 011626 000000 0
(1) 011630 000001 CORMAX:
1949 .END
```


MEPASS	004373	875	1231#						
MERRPC	004702	1168	1272#						
MERRX	004602	882	1259#						
MERR2	004421	1236#	1364						
MERR3	004470	1243#							
MEXT =	010000	607#	1531	1589	1616	1774	1796		
MIND	001332	628#	1919*	1925	1928*	1929			
MJMPR	005037	806	1289#						
MLOCK	004526	856	1249#						
MMODE =	014000	606#							
MMODEA=	004000	596#							
MMODEB=	010000	595#							
MMODEM	004741	784	1278#						
MNEW	004627	1264#							
MPAR	005161	1304#							
MPASSX	004571	880	1257#						
MPOWER	004355	1223	1228#						
MQM	004346	980	1226#	1429					
MR	004416	863	1235#	1424					
MRESET=	000400	461#	599#	1528	1585	1713	1785	1847	
MSRJM	005240	814	1313#						
MSTJM	005205	810	1308#						
MTCN	005112	794	1297#						
MTDATA=	040000	593#							
MTITLE	001000	499#	756						
MTSTN	004613	1166	1261#	1408					
MTSTPC	004514	1247#							
MVEC	004726	776	1276#						
MVECX	004563	878	1255#						
NEXT	001216	519#	932	1199	1499*	1584*	1771*		
NO.ATR	006354	1475#							
NO.BTR	006360	1479#	1698	1876					
OJUMPE	006306	1459#	1524						
OPCLRJ	001325	625#	808	836*	1446	1452			
OVERRUN=	040000	582#							
PARAM =	104405	664#	769	777	1409				
PARAM1	003156	998#	1015						
PARBIT=	000000	461#							
PARCSR	001416	687#	1377*	1787*					
PARERR	003232	1001	1003	1005	1014#	1021	1023	1025	
PASCNT	001230	524#	733*	873*	874	905			
PERFOR=	000000	461#							
FUPRO =	012600	439#	1193						
POP1SP=	005726	437#							
POP2SP=	022626	441#	934						
PRIRTY	001322	623#	834*						
PRISEC=	010000	590#							
PS =	177776	430#	730*	850*	1604*	1652*	1708*	1775*	1831*
PUSHRO=	010046	438#	1190						
PUSH1S=	005746	436#							
PUSH2S=	024646	440#							
PWRUP	004322	1217	1219#						
QV.FLG	001353	643#	735*	884*	923				
RABORT=	002000	584#							
RCRCIN=	040000	613#							
RCRC7T=	100000	612#							

SW01 = 000002	411#	1404												
SW02 = 000004	410#													
SW03 = 000010	408#													
SW04 = 000020	407#													
SW05 = 000040	406#													
SW06 = 000100	405#													
SW07 = 000200	404#													
SW08 = 000400	403#	1195												
SW09 = 001000	402#	938												
SW10 = 002000	401#	1197												
SW11 = 004000	400#	921												
SW12 = 010000	399#	944	1134											
SW13 = 020000	398#	1139												
SW14 = 040000	397#													
SW15 = 100000	396#													
SYSTST= 004000	608#	1605	1789											
TABORT= 002000	617#													
TCNFLG 001324	624#	789*	796	797	835*	1444	1460	1504	1587	1772				
TCRCIN= 010000	615#													
TCRC7T= 020000	614#													
TEMP 005454	1059	1345#												
TEMP1 001236	530#	835	836	839*	840	844*	1340	1602*	1685	1686*	1687	1721*	1744*	
	1780*	1856*	1859	1860*	1861									
TEMP2 001240	531#	840*	1342	1603*	1673	1676*	1677	1722*	1726*	1781*	1868	1869*	1870	
TEMP3 001242	532#	1723*	1725	1735*	1737	1742*	1743*	1746	1782*	1882*	1884			
TEMP4 001244	533#	1590*	1645*	1783*	1883*	1887	1890	1896*	1897	1900	1908*	1909	1921*	
	1922	1925	1933*	1934										
TEMP5 001246	534#	1784*	1893*	1900	1903*	1904								
TEOM = 001000	618#	1694	1865	1874										
TIMER = 004000	616#	1534	1536	1549	1551	1638	1640	1659	1661	1815	1817	1838	1840	
TKCSR 001204	510#	917	965											
TKDBR 001206	511#	919	967	973										
TLAST = 010030	1427	1948#												
TPCSR 001210	512#	949	971	1136										
TPDBR 001212	513#	951*	973*	1138*										
TRPOK 003710	1125#													
TRP_PC 005622	1351#													
TSOM = 000400	619#	1635	1813	1825	1828									
TSTNO 001226	523#	741*	1208	1412	1418	1420	1498*	1583*	1770*					
TST1 006364	1415	1432	1498#											
TST2 006744	1499	1583#												
TST3 010030	1584	1770#	1948											
TST4 = ***** U	1771													
TTST 002554	857*	858*	860*	861*	915#									
TWOSYN= 000000	461#													
TXACT = 001000	598#													
TXCSR 001420	688#	1381*	1382*	1383	1528*	1531*	1585*	1605*	1616*	1632	1634*	1648	1651*	
	1697*	1713*	1785*	1789*	1796*	1812*	1822	1826	1830*	1847*	1875*			
TXDBUF 001422	689#	1385*	1386*	1387	1534	1536	1549	1551	1635*	1638	1640	1650*	1659	
	1661	1685*	1691*	1694*	1813*	1815	1817	1825*	1828*	1838	1840	1855*	1859*	
	1865*	1868*	1874*											
TXDLAT= 100000	592#													
TXDONE= 000200	600#													
TXINTE= 000100	601#	1651	1697	1830	1875									
TYPDAT 004162	1160	1178	1181#											
TYPE = 104402	658#	756	838	856	863	875	876	878	880	882	948	961	980	

CROSS REFERENCE TABLE -- USER SYMBOLS

	1045	1082	1161	1162	1165	1166	1168	1170	1174	1179	1222	1364	1406
TYPMSG 004062	1045	1082	1161	1162	1165	1166	1168	1170	1174	1179	1222	1364	1406
USER = 000000	1423	1424	1429										
WRDCNT 003614	1158	1161#											
WRKO.F 004150	605#												
XBX 003754	1053*	1083*	1091#										
XCSR 002510	1173	1176#											
XERR 002532	1135	1137	1139#										
XHEAD 005273	877	897#											
XPASS 002524	883	906#											
XPOLY 010022	838	1318#											
XSTATQ 005442	881	903#											
XTSTN 004300	1592*	1731	1742	1751#									
XVEC 002516	843	1338#											
\$E = 000005	1167	1206#											
\$ENDAD 002464	879	900#											
\$N = 000003	1#	1499	1500#	1584	1585#	1771	1772#						
\$RAYO = 177777	487	889#	1186										
\$Y = 000013	1#	1484	1492	1500#	1572	1577	1585#	1758	1764	1772#	1948#		
.	1#	1484#	1485	1490#	1572#	1573	1575#	1758#	1759	1762#			
	647#	654	656#	658#	660#	662#	664#	666#	668#	670#	672#	674#	676#
	472#	473	476#	483#	488#	491#	494#	498#	500#	556#	557#	558#	559#
	624#	625#	630#	631#	711#	713#	714#	715#	1346#	1348#	1350#	1351#	1366
	1598#	1599#	1674	1704	1711	1823	1910	1948#					
.BEGIN 002246	759	825	841	850#									
.CNVRT 003414	673	1046#											
.CONVR 003410	671	1045#											
.EOP 002352	871#	1771											
.ERRTA 011550	1153	1948#											
.HLT 003730	479	1134#											
.INSTE 003076	663	978#											
.INSTR 002772	661	957#											
.INST1 003012	961#	981											
.MSG 003014	959*	962#											
.PARAM 003116	665	989#											
.PFAIL 004312	477	732	1217#	1221									
.RES05 003356	669	1042#											
.SA'05 003316	667	1042#											
.SCOPE 002540	655	912#											
.SCOPI 002672	657	938#											
.SETFL 003622	675	1102#	1114										
.START 001510	495	730#	742										
.TRPSR 003676	481	1122#											
.TRPTA 001354	653#	1127											
.TYPE 002716	659	941#											

. ABS. 011630 000

ERRORS DETECTED: 0

DSKZ:CZDPEB,DSKZ:CZDPEB,SEQ=DSKZ:CZDPEB,MAC,DSKZ:CZDPEB.P11
 RUN-TIME: 5 7 .6 SECONDS
 RUN-TIME RATIO: 29/13=2.2
 CORE USED: 17K (33 PAGES)

CZDPE MACY11 30A(1052) 19-JAN-79 16:09 PAGE 52
CZDPEB.P11 19-JAN-79 16:07 CROSS REFERENCE TABLE -- USER SYMBOLS

K 4

SEQ 0050
SEQ 0049

DOCUMENT PAGES: 49