

DL11-E,C,D

DL11 E C/D OFLNE TST
CZDLAHO

AH 8517H MC

COPYRIGHT 72 79

FICHE 1 OF 1

NOV 1979

digital

MADE IN USA

.REM a
.REPT 0

IDENTIFICATION

PRODUCT CODE: AC-8516H-MC
PRODUCT NAME: CZDLAHO DL11-E,C/D OFLNE TST
DATE: JULY 1979
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: ROBERT WHITTON

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1972,1979 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPOPATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

1. ABSTRACT

TWO SEPARATE DIAGNOSTIC PROGRAMS ARE PROVIDED FOR THE DL11-E (ASYNCRONOUS MODEM INTERFACE), CZDLA (DL11-E OFF LINE

TESTS) AND CZDLB (DL11-E ON LINE TESTS). THE OFF LINE TEST TESTS ALL DL11-E LOGIC . THE OFF LINE TESTS DO NOT REQUIRE THE USE OF A MODEM, HOWEVER A SPECIAL JUMPER CONNECTOR H315 IS REQUIRED. THE ON LINE TESTS ARE ESSENTIALLY DATA RELIABILITY TESTS REQUIRING THE USE OF MODEMS AND A SUITABLE TERMINAL DEVICE.

THE DL11-C AND DL11-D CAN ALSO BE TESTED WITH THIS OFF LINE TEST. THESE ARE BOTH TESTED IN MAINTENANCE MODE AND ONLY THOSE TESTS MARKED C,D IN THE TEST NUMBER ARE EXECUTED. IN ORDER TO TEST C AND D VERSIONS IT IS NECESSARY TO MODIFY THE TABLE AT LOCATION 1300 ACCORDING TO THE INSTRUCTIONS CONTAINED THERE.

TESTS WHICH ARE NOT EXECUTED FOR DL11C+D CAN BE PERFORMED BY USING THE SELECT SWITCH OPTION (SR9). TEST 56 IS A DATA TEST WHICH CAN BE USED FOR CABLE TESTING DL11-D'S. WARNING--A FAILURE IN THIS TEST MAY OCCUR DUE TO A SPLIT BAUD RATE OF RCVTR/TXVTR.

THIS DOCUMENT DESCRIBES THE OFF LINE TESTS.

THE AVAILABLE TESTS ARE:

PRG0	INPUT/OUTPUT LOGIC TESTS
PRG1	TRANSMITTER SCOPE LOOP
PRG2	RECEIVER SCOPE LOOP
PRG3	SINGLE CHARACTER MAINT. MODE DATA TEST
PRG4	SPECIAL BINARY COUNT MAINTENANCE MODE DATA TEST

2. REQUIREMENTS

2.1 EQUIPMENT

A. PDP 11 SYSTEM

B. DL11-E OR DL11-C OR DL11-D

C. SPECIAL JUMPER CONNECTOR H315 (SEE DL11 MAINTENANCE MANUAL FOR DETAILED DESCRIPTION) IF DL11-E.

2.2 STORAGE

THIS PROGRAM USES ALL OF CORE (4K) EXCEPT THAT AREA RESERVED FOR THE BOOTSTRAP AND ABSOLUTE LOADERS.

3. LOADING PROCEDURE

THE ABSOLUTE LOADER IS USED TO LOAD THE PROGRAM.

4. USE PROCEDURE

THIS PROGRAM HAS BEEN MODIFIED TO RUN WITH OR WITHOUT A CONSOLE PROCESSOR.
IF A CONSOLE MACHINE IS USED; THEN THE PROGRAM

LOOKS AT THE HARDWARE SWITCH REGISTER.
IF A CONSOLE-LESS MACHINE IS USED; THEN THE PROGRAM
AUTOMATICALLY LOOKS AT THE CONTENTS OF LOCATION
SOFTSR (176) AS A SWITCH REGISTER.

THE PROGRAM WILL PRINT OUT THE PRESENT CONTENTS OF THE SOFT. SWITCH REG.
WHEN THE PROGRAM IS STARTED. IT WILL THEN ASK FOR THE NEW CONTENTS TO
BE INPUT TO THE SOFTWARE SWITCH REGISTER. TYPE CARRIAGE RETURN TO FINISH INPUT.

BEFORE STARTING ANY OF THE SELECTABLE PROGRAMS MAKE SURE
THAT THE TTY IS IN REMOTE MODE; AND THAT THE PROGRAM SELECTED
IS A LEGAL PROGRAM, I.E. NO. 0-4, OTHERWISE AN ERROR MESSAGE
WILL OCCUR. TERMINATE ALL INPUTS WITH A CARRIAGE RETURN.

A MAP OF DEVICES PRESENT WILL BE TYPED AT RUN TIME. THIS MAP WILL NOT
BE TYPED OUT AGAIN UNLESS THE PROGRAM IS
RESTARTED AT LOCATION 200. A RESTART FROM THIS LOCATION WILL CAUSE
THE MAP OF DEVICES TO BE TYPED OUT AGAIN AND THEN A NORMAL START
WILL OCCUR.

4.1 PRGO INPUT/OUTPUT LOGIC TESTS

- A. LOAD ADDRESS = 000200 (RESTART LOAD ADDR. = 000204)
TYPE PROGRAM NUMBER = 0.
THE DIAGNOSTIC WILL IDENTIFY THE PROGRAM YOU SELECTED.
DISCONNECT THE DL11-E FROM THE MODEM AND INSERT THE JUMPER CON-
NECTOR IN THE MODEM END OF THE CABLE, AND TYPE CARRIAGE RETURN.
NOTE, IF THE CABLE IS LEFT CONNECTED TO THE MODEM THE FOL-
LOWING TESTS WILL FAIL:
AT22, AT23, AT25, AT30, AT32, AT56
- B. THE PROGRAM WILL TYPE OUT INSTRUCTIONS TO SET IN THE DESIRED
SR OPTIONS. TYPE CARRIAGE RETURN WHEN THE OPTIONS ARE IN THE SR.
THE AVAILABLE OPTIONS ARE:
SR 0-6 ROUTINE TO BE RUN (IF ENABLED BY SR9)
SR7 DISABLE STALL MODE
SR9 LOOP SELECTED ROUTINE
SR10 HALT AT END OF CURRENT TEST
SR11 INHIBIT ITERATION
SR12 SELECT LINE NUMBER AND LOCK ON IT
SR13 INHIBIT PRINTOUT
SR14 SCOPE
SR15 HALT ON ERROR.
- C. THE PROGRAM WILL NOW REQUEST THE LINE # (IF SR12=1) YOU WISH TO
TEST. TYPE THE LINE # AS REQUESTED, FOLLOWED BY A CARRIAGE RETURN.
LINE NUMBER REFERS TO THE ADDRESSES TO WHICH THE DL11-E RESPONDS.

LINE 00 77561X	LINE 10 77571X	LINE 20 77601X	LINE 30 77611X
LINE 01 77562X	LINE 11 77572X	LINE 21 77602X	LINE 31 77612X
LINE 02 77563X	LINE 12 77573X	LINE 22 77603X	LINE 32 77613X
LINE 03 77564X	LINE 13 77574X	LINE 23 77604X	LINE 33 77614X
LINE 04 77565X	LINE 14 77575X	LINE 24 77605X	LINE 34 77615X
LINE 05 77566X	LINE 15 77576X	LINE 25 77606X	LINE 35 77616X
LINE 06 77567X	LINE 16 77577X	LINE 26 77607X	LINE 36 77617X

LINE 07 77570X LINE 17 77600X LINE 27 77610X

- D. THE PROGRAM WILL NOW BEGIN TESTING THE DL11-E OR C/D YOU SELECTED. ALL DL11'S WILL BE TESTED AUTOMATICALLY AND SEQUENTIALLY UNLESS SR12 IS SELECTED.

NOTE: ALL LOGIC TESTS WILL NOT BE RUN AUTOMATICALLY. THERE ARE TWO TESTS WHICH REQUIRE MANUAL INTERVENTION WHICH ARE USED TO TEST THE SPEED SELECTION SWITCHES. THESE ARE TESTS T34, T40. TO EXECUTE THESE TESTS USE S'9 AND SR 0-6 TO SELECT THEM.

- E. REFER TO SECTION 5.1.2 FOR ERROR DESCRIPTION
- F. AFTER ONE COMPLETE PASS THE BELL WILL RING FOLLOWED BY 'END PASS = ' WITH THE NUMBER OF PASSES COMPLETED SINCE PROGRAM LAST STARTED AND THE DEVICE ADDRESS UNDER TEST AND ITS TRAP VECTOR.

4.2 PRG1 - TRANSMITTER SCOPE LOOP

- A. LOAD ADDRESS = 000200 (RESTART = 000204)
TYPE PROGRAM NUMBER = 1.
THE DIAGNOSTIC WILL IDENTIFY THE PROGRAM YOU SELECTED, AND REQUEST THE LINE # YOU WISH TO TEST. TYPE THE LINE # AS REQUESTED, FOLLOWED BY A CARRIAGE RETURN.
- B. THE PROGRAM WILL REQUEST A CHARACTER CODE, AND A DELAY TIME. THE CHARACTER CODE IS THE DATA THE DL11-E WILL TRANSMIT AND THE DELAY IS THE TIME ELAPSED BETWEEN SUCCESSIVE TRANSMISSIONS OF ONE CHARACTER.
- C. THE PROGRAM WILL RUN WITHOUT ERROR OR END TYPEOUTS.

4.3 PRG2 - RECEIVER SCOPE LOOP

- A. LOAD ADDRESS = 000200 (RESTART = 000204)
TYPE PROGRAM NUMBER = 2.
THE DIAGNOSTIC WILL IDENTIFY THE PROGRAM YOU SELECTED, AND REQUEST THE LINE # YOU WISH TO TEST. TYPE THE LINE # AS REQUESTED, FOLLOWED BY A CARRIAGE RETURN.
- B. THE PROGRAM WILL REQUEST A TEST CHARACTER CODE, AND A DELAY TIME. THE CHARACTER CODE IS THE DATA THAT THE DL11-E WILL BE TRANSMITTING AND THE DELAY IS THE ELAPSED TIME BETWEEN SUCCESSIVE CHARACTERS.
- C. THE PROGRAM WILL NOW RUN WITHOUT ERROR OR END TYPEOUTS.

4.4 PRG3 - SINGLE CHARACTER MAINT MODE DATA TEST

- A. LOAD ADDRESS = 000200 (RESTART = 000204)
TYPE PROGRAM NUMBER = 3.
THE DIAGNOSTIC WILL IDENTIFY THE PROGRAM YOU SELECTED, AND

REQUEST THE LINE # YOU WISH TO TEST. TYPE THE LINE # AS REQUESTED, FOLLOWED BY A CARRIAGE RETURN.

- B. THE PROGRAM WILL REQUEST A TEST CHARACTER. TYPE THE TEST CHARACTER, FOLLOWED BY A CARRIAGE RETURN.
- C. THE PROGRAM WILL NOW RUN CONTINUOUSLY REPORTING ANY DATA FAILURES.

4.5 PRG4 - SPECIAL BINARY COUNT MAINT. MODE DATA TEST

- A. LOAD ADDRESS = 000200
TYPE PROGRAM NUMBER = 4.
THE DIAGNOSTIC WILL IDENTIFY THE PROGRAM YOU SELECTED, AND REQUEST THE LINE # YOU WISH TO TEST. TYPE THE LINE # AS REQUESTED, FOLLOWED BY A CARRIAGE RETURN.
- B. THE PROGRAM WILL BEGIN TESTING THE LINE YOU SELECTED. AND REPORT ANY DATA ERRORS.

5. PROGRAM DESCRIPTIONS

5.1 PRG0 - INPUT/OUTPUT LOGIC TESTS

THE INPUT/OUTPUT LOGIC TESTS CONSIST OF 57(B) ROUTINES WHICH MAY BE RUN IN SEQUENTIAL ORDER OR INDIVIDUALLY LOOPED (SEE SECT 4.1, C FOR SWITCH SETTINGS). THE JUMPER CONNECTOR MUST BE INSERTED BEFORE STARTING IF DL11-E.

5.1.1 ROUTINE DESCRIPTIONS

ROUTINE	TESTS
AT0-AT3 AT4-AT27	ADDRESSABILITY OF CSRS & DBRS DIDDLES ALL BITS IN THE CSRS AND CHECKS THAT THEY CAN BE READ/WRITTEN PROPERLY.
AT31-AT32 AT33 AT34	PROPER OPERATION OF RESET INSTRUCTION PROPER OPERATION OF READY BIT PROPER OPERATION OF TRANSMIT SPEED SELECTION
AT35-AT37	PROPER OPERATION OF DONE BIT
AT40	PROPER OPERATION RECEIVER SPEED SELECT
AT41	PROPER OPERATION OF DATA OVERRUN
AT42-AT52	PROPER OPERATION OF INTERRUPTS
AT53	READING RXCSR DOES NOT CLEAR DONE
AT54	ERROR CAUSES INTERRUPT

AT55 DATA TEST MAINTENANCE MODE
AT56 DATA TEST WITH JUMPER
AT57 PROPER OPERATION OF BREAK BIT

5.1.2 ERROR DESCRIPTION

IF A ROUTINE FAILS AND THE INHIBIT PRINTOUT SWITCH IS NOT ENABLED (SR13) A PRINTOUT RESULTS. THE PRINTOUT FORMAT IS:

T(ROUTINE#) PC=(PC OF ERROR CALL) RXCSR=(ADDRESS OF DEVICE UNDER TEST) AND AN ADDITIONAL/MESSAGE (IF APPLICABLE)

T005 PC-XXXX RXCSR=XXXX

T122 PC-XXXX RXCSR-XXXX DATA S/B:---WAS:---
INDICATING A DATA ERROR

TO RESUME TESTING PRESS CONTINUE.
IF THE VECTOR PROVIDED BY THE INTERRUPTING DL11-E IS INCORRECT A TRAP TO THE WRONG LOCATION WILL OCCUR AND AN ERROR MESSAGE WILL OCCUR.

5.1.3 JUMPER CONNECTOR

THE JUMPER CONNECTOR TESTS THOSE F/F'S, GATES (RING INDICATOR, CARRIER TRANSITION, CLEAR TO SEND, AND SUPERVISORY RECEIVE DATA) WHICH CANNOT BE TESTED UNLESS A DATA SET IS ACTUALLY CONNECTED TO THE DL11-E. IN ADDITION TO TESTING DL11-E LOGIC THE JUMPER ALSO TESTS CABLE WIRING TO/FROM THE DL11-E/DATA SET. THE FOLLOWING TESTS WILL FAIL IF THE CABLE IS NOT INSTALLED IN THE DL11-E:

AT22,AT23,AT25,AT30,AT32,AT56

5.2 PRG1-TRANSMITTER SCOPE LOOP

THE PURPOSE OF PRG1 IS TO ALLOW SCOPING OF TRANSMITTER FUNCTIONS IN A RUN CONDITION USING USER SPECIFIED DL11-E PARAMETERS AND DATA. NO ERROR PRINTOUTS ARE PROVIDED.

5.3 PRG2-RECEIVER SCOPE LOOP

THE PURPOSE OF PRG2 IS TO ALLOW SCOPING OF RECEIVER FUNCTIONS IN A RUN CONDITION USING USER SPECIFIED DL11-E PARAMETERS AND DATA. NO ERROR PRINTOUTS ARE PROVIDED.

5.4 PRG3-SINGLE CHARACTER MAINT MODE DATA TEST

PRG3 TRANSMITS, RECEIVES AND CHECKS RECEIVED DATA USING USER SPECIFIED DL11-E PARAMETERS, AND DATA.

5.4.1 ERROR PRINTOUTS

SELF EXPLANATORY ERROR PRINTOUTS ARE PROVIDED.

5.5 PRG4-SPECIAL BINARY COUNT MAINT MODE DATA TEST

PRG4 IS THE SAME AS PRG0 ROUTINE 54 EXCEPT THAT THE USER SPECIFIES DL11-E RUNNING PARAMETERS.

5.5.1 ERROR PRINTOUTS

SELF EXPLANATORY PRINTOUTS ARE PROVIDED.

6.0 POWER FAIL

A POWER FAIL ROUTINE IS INCLUDED IN THE PROGRAM. WHEN THE POWER FAILS THE PROGRAM WILL AUTOMATICALLY RESTART USING THE PRESENT SR OPTIONS AND THE LINE PREVIOUSLY SELECTED. NOTE: THE POWER MAY FAIL WHEN THE PROGRAM IS EXECUTING A 'RESET' INSTRUCTION, IN THIS CASE OPERATOR INTERVENTION IS NEEDED TO PRESS CONTINUE. AN ERROR TYPEOUT RESULTS AND WILL TYPE THE PROGRAM #, THE ROUTINE THAT WAS RUNNING AT THE TIME THE POWER FAILED (PROGRAM 0 ONLY), AND THE PC OF THE POWER FAIL ERROR CALL.

RECOVERED FROM POWER FAILURE.
P(PRG#) T(ROUTINE #) PC = (ADDRESS OF ERROR CALL)

***** ECO HISTORY *****
CHGH1 - NEW STORAGE LOCATIONS FOR REFERENCE IN ^G MODE
CHGH2 - CHANGE START UP FOR ENTERING SOFTWARE SWITCH REGISTER INFO.
CHGH3 - CHECK FOR ^G BETWEEN TESTS.
CHGH4 - NEW EMT LOCATION
CHGH5 - ROUTINE TO CHECK PRINTOUT, UPDATE SOFTWARE SWITCH REG.
CHGH6 - NEW MESSAGES FOR SOFTWARE SWITCH REG ROUTINES.
CHGH7 - NEW EMT ROUTINE TO PRINTOUT A <CR> AND <LF>.
CHGH8 - DELETED HALT AND INSERTED WAIT FOR TTY INPUT
CHGH9 - DELETED HALT AND INSERTED WAIT FOR TTY INPUT

.ENDR

385
386
387
388
389
390
391
392

.LIST SEQ,BIN
.ENABLE ABS,AMA
;DL11-E,C/D DIAGNOSTIC PROGRAM (OFF LINE TESTS)
;
;PRG0- INPUT-OUTPUT LOGIC TESTS
;PRG1- TRANSMITTER SCOPE LOOP
;PRG2- RECEIVER SCOPE LOOP


```
393 ;PRG3- SINGLE CHARACTER MAINTENANCE MODE DATA TEST
394 ;PRG4- SPECIAL BINARY COUNT MAINTENANCE MODE DATA TEST
395
396 ;STANDARD SR SWITCH OPTIONS (SWITCH SET TO A 1 )
397
398 ;SR15- HALT ON ERROR
399 ;SR14- SCOPE.
400 ;SR13- INHIBIT PRINTOUT
401 ;SR12- SELECT LINE NUMBER AND LOCK ON IT
402 ;SR11- INHIBIT ITERATION.
403 ;SR10- HALT AT END CURRENT TEST, TEST NO. IN DATA LIGHTS
404 ;SR9- SELECT ROUTINE.
405 ;SR7- DISABLE STALL MODE AND RUN FULL SPEED.
406 ;SR6 THROUGH SR0 - NUMBER OF ROUTINE TO BE SELECTED.
407
408 ;STANDARD CONFIGURATION
409 ;CHARACTER LENGTH 8
410 ;STOP CODE 2
411
412 000000 000000 . =0
413 000002 000000 ERTP ;UNASSIGNED TRAP
414 000004 006042 0
415 000006 000040 MACHER: ERTP ;SP OVERFLOW, BUS ERROR TRAP
416 000010 006042 40
417 000012 000100 ERTP ;RESERVED INSTRUCTION TRAP
418 000014 006042 100
419 000016 000140 ERTP ;TRACE TRAP
420 000020 006150 MAPVEC ;TRAP TO MAP VECTOR
421 000022 000340 PRTY7
422 000024 006370 PFAIL ;POWER FAIL TRAP
423 000026 000340 PRTY7
424 000030 003064 EMTINT ;EMT TRAP
425 000032 000340 PRTY7
426 000034 006042 ERTP
427 000036 000340 340
428 000040 000042 . +2
429 000042 000000 HALT
430 000046 000046 . =46
431 000046 005446 LOGIC
432 000165 .REPT 117.
433 . +2 ;TRAP TO TRAP REPORTER
434 4
435 .ENDR
436
437
438 ;EQUATE STATEMENTS
439 177776 PSW=177776
440 001176 SPBOT=1176
441 000240 NOP=240
442 000000 OPEN=0
443 100000 MANUAL=BIT15
444 100000 BIT15=100000
445 040000 BIT14=40000
446 020000 BIT13=20000
447 010000 BIT12=10000
448 004000 BIT11=4000
```

449	002000	BIT10=2000
450	001000	BIT9=1000
451	000400	BIT8=400
452	000200	BIT7=200
453	000100	BIT6=100
454	000040	BIT5=40
455	000020	BIT4=20
456	000010	BIT3=10
457	000004	BIT2=4
458	000002	BIT1=2
459	000001	BIT0=1
460	005726	POPSP=5726
461	022626	POPSP2=022626
462	000340	PRTY7=340
463	000300	PRTY6=300
464	000240	PRTY5=240
465	000200	PRTY4=200
466	000140	PRTY3=140
467	000100	PRTY2=100
468	000040	PRTY1=40
469	000000	PRTY0=0

:POP THE STACK. SAME AS TST (6)+
:POP STACK TWICE. SAME AS CMP (6)+,(6)+
:PRIORITY LEVEL DEFINITIONS

473	104000	TYPE=EMT+0
474	104001	TYPES=EMT+1
475	104002	STALL=EMT+2
476	104003	ERROR=EMT+3
477	104004	DATCHK=EMT+4
478	104005	CHALT=EMT+5
479	104006	STRXV=EMT+6
480	104007	STTXV=EMT+7
481	104010	EHALT=EMT+10
482	104011	SRESET=EMT+11
483	104012	SCOPE=EMT+12
484	104013	SAVREG=EMT+13
485	104014	RSTREG=EMT+14
486	104015	ERROR1=EMT+15
487	104016	DELAY=EMT+16
488	104017	TIMERX=EMT+17
489	104020	TIMETX=EMT+20
490	104021	SCRLF=EMT+21
491	177777	ATLAST=-1
492	100000	CD=100000

:CHG4
:FLAG FOR C/D TESTS

500		.LIST	ME	
501		.=170		
502	000170	177570	SPPTR:	:SWITCH REG POINTER
503	000172	177570	DISPREG:	:DISPLAY REGISTER
504	000174	000000	DISPLAY:	:SOFTWARE DISPLAY REG

```
505 000176 000000 SWREG: OPEN ;SOFTWARE REGISTER
506 000200 000200 .=-200
507 000200 000137 001656 JMP @#STARTZ ;GO TO START OF PROGRAM.
508 000204 000204 .=-204
509 000204 000137 006550 JMP @#RESTART
510 001200 .=-1200
511
512 ;DEVICE ADDRESS LIST
513 ;LSB BIT0 IS SET TO A 1 BY MAPPER IF DEVICE NOT FOUND
514 ;TO TEST THAT LINE NOT FOUND CLEAR BIT0 IN THAT DEVICE ADDRESS
515 ;IN THIS TABLE AFTER MAPPING DONE
516 ;*****
517 001200 175610 RXCR0: 175610 ;LINE 0 DEVICE ADDRESS (RXCSR)
518 001202 175620 RXCR1: 175620 ;LINE 1 DEVICE ADDRESS (RXCSR)
519 001204 175630 RXCR2: 175630 ;LINE 2 DEVICE ADDRESS (RXCSR)
520 001206 175640 RXCR3: 175640 ;LINE 3 DEVICE ADDRESS (RXCSR)
521 001210 175650 RXCR4: 175650 ;LINE 4 DEVICE ADDRESS (RXCSR)
522 001212 175660 RXCR5: 175660 ;LINE 5 DEVICE ADDRESS (RXCSR)
523 001214 175670 RXCR6: 175670 ;LINE 6 DEVICE ADDRESS (RXCSR)
524 001216 175700 RXCR7: 175700 ;LINE 7 DEVICE ADDRESS (RXCSR)
525 001220 175710 RXCR10: 175710 ;LINE 10 DEVICE ADDRESS (RXCSR)
526 001222 175720 RXCR11: 175720 ;LINE 11 DEVICE ADDRESS (RXCSR)
527 001224 175730 RXCR12: 175730 ;LINE 12 DEVICE ADDRESS (RXCSR)
528 001226 175740 RXCR13: 175740 ;LINE 13 DEVICE ADDRESS (RXCSR)
529 001230 175750 RXCR14: 175750 ;LINE 14 DEVICE ADDRESS (RXCSR)
530 001232 175760 RXCR15: 175760 ;LINE 15 DEVICE ADDRESS (RXCSR)
531 001234 175770 RXCR16: 175770 ;LINE 16 DEVICE ADDRESS (RXCSR)
532 001236 176000 RXCR17: 176000 ;LINE 17 DEVICE ADDRESS (RXCSR)
533 001240 176010 RXCR20: 176010 ;LINE 20 DEVICE ADDRESS (RXCSR)
534 001242 176020 RXCR21: 176020 ;LINE 21 DEVICE ADDRESS (RXCSR)
535 001244 176030 RXCR22: 176030 ;LINE 22 DEVICE ADDRESS (RXCSR)
536 001246 176040 RXCR23: 176040 ;LINE 23 DEVICE ADDRESS (RXCSR)
537 001250 176050 RXCR24: 176050 ;LINE 24 DEVICE ADDRESS (RXCSR)
538 001252 176060 RXCR25: 176060 ;LINE 25 DEVICE ADDRESS (RXCSR)
539 001254 176070 RXCR26: 176070 ;LINE 26 DEVICE ADDRESS (RXCSR)
540 001256 176100 RXCR27: 176100 ;LINE 27 DEVICE ADDRESS (RXCSR)
541 001260 176110 RXCR30: 176110 ;LINE 30 DEVICE ADDRESS (RXCSR)
542 001262 176120 RXCR31: 176120 ;LINE 31 DEVICE ADDRESS (RXCSR)
543 001264 176130 RXCR32: 176130 ;LINE 32 DEVICE ADDRESS (RXCSR)
544 001266 176140 RXCR33: 176140 ;LINE 33 DEVICE ADDRESS (RXCSR)
545 001270 176150 RXCR34: 176150 ;LINE 34 DEVICE ADDRESS (RXCSR)
546 001272 176160 RXCR35: 176160 ;LINE 35 DEVICE ADDRESS (RXCSR)
547 001274 176170 RXCR36: 176170 ;LINE 36 DEVICE ADDRESS (RXCSR)
548 001276 177777 XORADD: 177777 ;LINE 37 SPECIAL ADDRESS FOR XOR
549 001300 177777 RXEND: 177777 ;LINE XX DEVICE ADDRESS (RXCSR)
550
551 ;CHARACTER LENGTH, PRIORITY, C/D MASK
552 ;INITIALLY SET FOR DL11-E, PRIORITY=4, CHARACTER LENGTH=8
553 ;BIT 15 SET TO A 1 = THAT LINE HAS DL11-C OR DL11-D
554 ;EX: 140377 = DL11C OR DL11D, PRIORITY = 4, CHARACTER LENGTH 8
555 ;BITS 12-14 = PRIORITY LEVEL THAT LINE
556 ;BITS 0-7 = CHARACTER MASK EX. 377=8, 177=7, 77=6, 37=5
557 ;*****
558 001302 040377 CMAS0: 040377 ;LINE 0 CHARACTER MASK, PRIORITY, C/D FLAG
559 001304 040377 CMAS1: 040377 ;LINE 1 CHARACTER MASK, PRIORITY, C/D FLAG
560 001306 040377 CMAS2: 040377 ;LINE 2 CHARACTER MASK, PRIORITY, C/D FLAG
```

561	001310	040377	CMAS3:	040377	;LINE 3	CHARACTER MASK, PRIORITY, C/D FLAG
562	001312	040377	CMAS4:	040377	;LINE 4	CHARACTER MASK, PRIORITY, C/D FLAG
563	001314	040377	CMAS5:	040377	;LINE 5	CHARACTER MASK, PRIORITY, C/D FLAG
564	001316	040377	CMAS6:	040377	;LINE 6	CHARACTER MASK, PRIORITY, C/D FLAG
565	001320	040377	CMAS7:	040377	;LINE 7	CHARACTER MASK, PRIORITY, C/D FLAG
566	001322	040377	CMAS10:	040377	;LINE 10	CHARACTER MASK, PRIORITY, C/D FLAG
567	001324	040377	CMAS11:	040377	;LINE 11	CHARACTER MASK, PRIORITY, C/D FLAG
568	001326	040377	CMAS12:	040377	;LINE 12	CHARACTER MASK, PRIORITY, C/D FLAG
569	001330	040377	CMAS13:	040377	;LINE 13	CHARACTER MASK, PRIORITY, C/D FLAG
570	001332	040377	CMAS14:	040377	;LINE 14	CHARACTER MASK, PRIORITY, C/D FLAG
571	001334	040377	CMAS15:	040377	;LINE 15	CHARACTER MASK, PRIORITY, C/D FLAG
572	001336	040377	CMAS16:	040377	;LINE 16	CHARACTER MASK, PRIORITY, C/D FLAG
573	001340	040377	CMAS17:	040377	;LINE 17	CHARACTER MASK, PRIORITY, C/D FLAG
574	001342	040377	CMAS20:	040377	;LINE 20	CHARACTER MASK, PRIORITY, C/D FLAG
575	001344	040377	CMAS21:	040377	;LINE 21	CHARACTER MASK, PRIORITY, C/D FLAG
576	001346	040377	CMAS22:	040377	;LINE 22	CHARACTER MASK, PRIORITY, C/D FLAG
577	001350	040377	CMAS23:	040377	;LINE 23	CHARACTER MASK, PRIORITY, C/D FLAG
578	001352	040377	CMAS24:	040377	;LINE 24	CHARACTER MASK, PRIORITY, C/D FLAG
579	001354	040377	CMAS25:	040377	;LINE 25	CHARACTER MASK, PRIORITY, C/D FLAG
580	001356	040377	CMAS26:	040377	;LINE 26	CHARACTER MASK, PRIORITY, C/D FLAG
581	001360	040377	CMAS27:	040377	;LINE 27	CHARACTER MASK, PRIORITY, C/D FLAG
582	001362	040377	CMAS30:	040377	;LINE 30	CHARACTER MASK, PRIORITY, C/D FLAG
583	001364	040377	CMAS31:	040377	;LINE 31	CHARACTER MASK, PRIORITY, C/D FLAG
584	001366	040377	CMAS32:	040377	;LINE 32	CHARACTER MASK, PRIORITY, C/D FLAG
585	001370	040377	CMAS33:	040377	;LINE 33	CHARACTER MASK, PRIORITY, C/D FLAG
586	001372	040377	CMAS34:	040377	;LINE 34	CHARACTER MASK, PRIORITY, C/D FLAG
587	001374	040377	CMAS35:	040377	;LINE 35	CHARACTER MASK, PRIORITY, C/D FLAG
588	001376	040377	CMAS36:	040377	;LINE 36	CHARACTER MASK, PRIORITY, C/D FLAG
589	001400	040377	CMAS37:	040377	;LINE 37	SPECIAL ADDRESS FOR XOR
590			:			
591	001402	000000	UMASK:	0		:MASK FOR DEVICE UT
592	001404	000000	RMASK:	0		:MASK FOR CHAR LENGTH FOR DEVICE UT
593	001406	177740	STLMSK:	177740		:MASK FOR MAX RANDOM STALL
594			:			
595	001410	000000	RXCSR:	0		:RECEIVER UNDER TEST
596	001412	000000	RXBUF:	0		:RECEIVER BUFFER UNDER TEST
597	001414	000000	TXCSR:	0		:TRANSMITTER CSR UNDER TEST
598	001416	000000	TXBUF:	0		:TRANSMITTER BUFFER UNDER TEST
599	001420	000000	RXVTR:	0		:RECEIVER VECTOR UNDER TEST
600	001422	000000	RXLVL:	0		:RECEIVER PRIORITY LEVEL UT
601	001424	000000	TXVTR:	0		:TRANSMITTER VECTOR UNDER TEST
602	001426	000000	TXLVL:	0		:TRANSMITTER PRIORITY LEVEL UT
603			:			
604			:			
605	001430	177560	TKS:	177560		:LSR CSR
606	001432	177562	TKB:	177562		:LSR BUFFER
607	001434	177564	TPS:	177564		:LSP CSR
608	001436	177566	TPB:	177566		:LSP BUFFER
609	001440	000060	TKVTR:	60		:LSR INTERRUPT VECTOR
610	001442	000200	TKLVL:	PRTY4		:LSR PRIORITY LEVEL
611	001444	000064	TPVTR:	64		:LSP INTERRUPT VECTOR
612	001446	000200	TPLVL:	PRTY4		:LSP PRIORITY LEVEL
613	001450	000000	PRGNUM:	OPEN		:CONTAINS CURRENT PROGRAM#
614	001452	000000	KSTART:	OPEN		:CURRENT PROGRAM START ADDRESS.
615	001454	000000	CURTST:	OPEN		:CONTAINS ADDR OF CURRENT TEST.
616	001456	000000	RTNNO:	OPEN		:CONTAINS CURRENT TEST #.

617 001460 000000
618 001462 000000
619 001464 000000
620 001466 000000
621 001470 000000
622 001472 000000
623 001474 006602
624 001476 014662
625 001500 014734
626 001502 015032
627 001504 015072
628 001506 005244
629 001510 005244
630 001512 005244
631 001514 003440
632 001516 003562
633 001520 004072
634 001522 005652
635 001524 005560
636 001526 000000
637 001530 003216
638 001532 003252
639 001534 005516
640 001536 003306
641 001540 002632
642 001542 003116
643 001544 003156
644 001546 005674
645 001550 004024
646 001552 004142
647 001554 004152
648 001556 004212
649
650 001560 000000
651 001562 000000
652 001564 000000
653 001566 000000
654 001570 000000
655 001572 000000
656 001574 000000
657 001576 000000
658 001600 000000
659 001602 000000
660 001604 000000
661 001606 000000
662 001610 000000
663 001612 000000
664 001614 000000
665 001616 000000
666 001620 000000
667 001622 000000
668 001624 000000
669 001626 000000
670 001630 000000
671 001632 000000
672 001634 000000

TNNO: 0
NXTST: OPEN
ICTR: OPEN
SCOPTR: OPEN
OLDPS: 0
FMAP: 0
PRGTAB: PRGO
PRG1
PRG2
PRG3
PRG4
INCRPG
INCRPG
INCRPG
EMTTAB: TYP
TYP5
STAL
ERR
DTCHK
OPEN
STLSRV
STLSPV
EHLT
SRSETT
CHAINN
SAVRG
RSTRG
ERR1
DLY
TMRX
TMTX
\$SCRLF
CRBUF: OPEN
CRBUFA: OPEN
CRBUFB: OPEN
CTR0: OPEN
CTR1: OPEN
CTR2: OPEN
CTR3: OPEN
CTR4: OPEN
CTR5: OPEN
CTR6: OPEN
CTR7: OPEN
TXCSRT: OPEN
RXCSRT: OPEN
RXBUFT: OPEN
FOUNDV: 0
LINENO: 0
TEMP: OPEN
TEMP1: 0
TEMP2: 0
COUNT: 0
FTITLE: 0
FNONE: 0
TOPC: 0

:CONTAINES EDITED TNUM
:CONTAINS ADDR OF NEXT TEST.
:CONTAINS CURRENT ITERATION COUNT
:CONTAINS CURRENT SCOPE POINTER.
:PS SAVED FROM TRAP TO EMT ROUTINE
:MAPPING FLAG, 1= MAPPING IN PROGRESS
:PRGO START ADDRESS
:PRG1 START ADDRESS
:PRG2 START ADDRESS
:PRG3 START ADDRESS
:PRG4 START ADDRESS
:INCORRECT PROGRAM SELECTED

:POINTER TO TYPEOUT ROUTINE
:POINTER TO CHAINED MESSAGES ROUTINE
:POINTER TO RANDOM STALL ROUTINE
:POINTER TO ERROR ROUTINE

:POINTER TO ERROR HALT ROUTINE.

673	001636	000000			FROMPC: 0	
674	001640	000000			PASCNT: 0	
675					:***** CHGH1 *****	
676	001642	000000			TIB: OPEN	:TEMPORARY KYBD BUFFER STORAGE
677	001644	000000			TEMPST: OPEN	:TEMPORARY WORK LOCATION
678	001646	000000			COUNT1: OPEN	:LOOP COUNT FOR TEST
679	001650	000000			FILL: OPEN	:ZERO FILL SWITCH
680	001652	000000			MODE: OPEN	:NUMBER OF DIGITS TO TYPE
681	001654	000000			CNT: OPEN	:ITERATION COUNT
682					:*****	
683						
684						
685						
686						
687						
688	001656	012706	001176		STARTZ: MOV #SPBOT,%6	:SET UP STACK POINTER
689					:*****	
690	001662	013746	000006		CHGH2: MOV 6,-(SP)	:SAVE CURRENT VECTOR
691	001666	013746	000004		MOV 4,-(SP)	
692	001672	012737	001712	000004	MOV #64\$,@#4	:SET UP LOC. 4 FOR NON-EXIS MEM TRAP
693	001700	022777	177777	176262	CMP #-1,@SRPTR	:REFERENCE HARDWARE SWITCH REGISTER
694	001706	001402			BEQ 65\$:IF IT = -1,USE SOFT SW REG
695	001710	000407			BR 66\$:THEN USE HARDWARE SW REG
696	001712	022626			64\$: POPSP2	:CORRECT THE STACK, NO HDWE SW REG
697	001714	012737	000176	000170	65\$: MOV #SWREG,SRPTR	:POINT TO SOFT SW REG
698	001722	012737	000174	000172	MOV #DISPLAY,DISPREG	:POINT TO SOFT DISP REG
699	001730	012637	000004		66\$: MOV (SP)+,@#4	:RESTORE VECTORS
700	001734	012637	000006		MOV (SP)+,@#6	:SAME AS ABOVE
701	001740	022737	000176	000170	67\$: CMP #SWREG,SRPTR	:IS SOFT SWREG SELECTED
702	001746	001015			BNE 60\$:IF NOT, BR OVER SOFTWARE OPER
703	001750	004737	015256		JSR PC,CNTLU	:SOFT SW REG INPUT ROUTINE
704	001754	013701	001440		MOV TKVTR,R1	:SET UP TO INSTALL VECTORS
705	001760	012721	015502		MOV #TTINTS,(R1)+	:VECTOR ADDRESS TO LOC 60
706	001764	013721	001442		MOV TKLVL,(R1)+	:PRIORITY TO LOC 62
707	001770	005777	177436		TST @TKB	:CLEAR DONE FLAG
708	001774	012777	000100	177426	MOV #100,@TKS	:SET TTY INTERRUPT ON
709					:*****	
710	002002	005037	001630		60\$: CLR @#FTITLE	
711	002006	013746	000004		MOV @#4,-(%6)	
712	002012	012737	002112	000004	MOV #XORA,@#4	
713	002020	005737	177060		TST @#177060	
714	002024	012637	000004		MOV (%6)+,@#4	
715	002030	012737	174000	001276	MOV #174000,@#XORADD	
716	002036	012737	177777	002110	MOV #-1,@#XORFLG	
717	002044	104000			TYPE	
718	002046	002054			MESS1	
719	002050	000137	002136		JMP @#START	
720	002054	005015	047531	020125	MESS1: .ASCII <15><12>'YOU ARE ON AN XOR TESTER@'	
721	002062	051101	020105	047117		
722	002070	040440	020116	047530		
723	002076	020122	042524	052123		
724	002104	051105	100			
725		002110			.FVEN	
726	002110	000000			XORFLG: .WORD 0	
727						
728	002112	022626			XORA: CMP (%6)+,(%6)+	

```
729 002114 012637 000004          MOV    (%6)+,%#4
730 002120 012737 177777 001276    MOV    #-1,%#XORADD
731 002126 005037 002110          CLR    %#XORFLG
732 002132 000137 002136          JMP    %#START
733
734
735                                .MACR  TSTAA  AX,B,C,D,E
736                                ;*****
737                                AT'E':  C          ;TEST NUMBER
738                                AT'D'    ;ADDRESS OF NEXT TEST
739                                B          ;ITERATION COUNT
740                                'AX'A    ;SCOPE ENTRY POINT
741                                X=X+1
742                                ;*****
743                                .ENDM
744                                .MACR  TSTA   B,AX,Z
745                                TSTAA  AX,B,\X+1+Z,\X+2,\X+1
746                                .ENDM
747
748 002136 012706 001176          START: MOV    #SPBOT,%6          ;SET BOTTOM OF SP STACK.
749 002142 012737 006370 000024    MOV    #PFAIL,%24
750 002150 005037 001614          CLR    FOUNDV
751 002154 005037 001472          CLR    FMAP
752 002160 004737 003376          JSR    %7,CLRCD          ;CLEAR DEVICE UT PARAMETERS
753 002164 004737 003614          JSR    %7,OVRLAY        ;OVERLAY TRAP AREA
754 002170 005737 001630          TST    FTITLE           ;TITLE PRINTED AND MAP MADE
755 002174 001054                    BNE    START1           ;YES, SKIP OVER THIS
756 002176 104000                    TYPE
757 002200 015710                    MTIT
758 002202 005237 001630          INC    FTITLE
759 002206 005037 001632          CLR    FNONE           ;CLEAR DEVICE PRESENT FLAG
760 002212 012737 002252 000004    MOV    #MAPNE,MACHER   ;SET UP NO DEVICE PRESENT RETURN
761 002220 012704 001200          MOV    #RXCRO,%4       ;SET UP DEVICE POINTER
762 002224 021437 001300          MAPA: CMP    (%4),%#RXEND ;LAST DEVICE
763 002230 001430                    BEQ    MAPEND           ;YES, EXIT
764 002232 042714 000001          BIC    #BITO,(4)       ;CLEAR ODD ADDRESS
765 002236 005037 177776          CLR    PSW
766 002242 005774 000000          TST    @(%4)           ;TEST DEVICE
767 002246 000240                    NOP
768 002250 000404                    BR    MAPOK
769 002252 052724 000001          MAPNE: BIS    #BITO,(4)+ ;NOT LIVING
770 002256 022626                    POPSP2
771 002260 000761                    BR    MAPA
772 002262 012437 001622          MAPOK: MOV    (%4)+,TEMP1 ;SAVE DEVICE ADDRESS FOR TYPING
773 002266 004537 004624          JSR    %5,OACNV
774 002272 001622                    TEMP1
775 002274 016001                    MDEVAD
776 002276 000006                    6
777 002300 104000                    TYPE
778 002302 016001                    MDEVAD
779 002304 005237 001632          INC    FNONE           ;SET HAVE DEVICE
780 002310 000745                    BR    MAPA
781 002312 012737 006042 000004    MAPEND: MOV   #ERTIP,MACHER ;RESET TRAPS
782 002320 005737 001632          TST    FNONE           ;ANY DEVICES PRESENT
783 002324 001424                    BEQ    MAPERR          ;NO, ERROR
784 002326 012701 001200          START1: MOV   #RXCRO,%1
```

785	002332	032711	000001		START2:	BIT	#BIT0,(1)	;IS DEVICE LIVING
786	002336	001013				BNE	START3	;NO, CHECK FOR END
787	002340	010137	001616			MOV	%1,LINENO	;CALCULATE LINE NUMBER UNDER TEST
788	002344	162737	001200	001616		SUB	#RXCRO,LINENO	
789	002352	006237	001616			ASR	LINENO	
790	002356	011101				MOV	(1),%1	;YES, LOAD AND EXIT
791	002360	004737	006226			JSR	%7,FORMAD	
792	002364	000420				BR	START4	
793	002366	005721			START3:	TST	(1)+	
794	002370	020127	001300			CMP	%1,#RXEND	;END OF TABLE
795	002374	001356				BNE	START2	;NO, LOOP
796	002376	104000			MAPERR:	TYPE		
797	002400	016055				MNONE		
798	002402	005737	000042			TST	@#42	;MONITOR LOAD
799	002406	001402				BEQ	.+6	;NO, CONTINUE
800	002410	000137	005436			JMP	PRGXTL	;YES, EXIT
801	002414	005037	001630			CLR	FTITLE	
802	002420	000000				HALT		
803	002422	000137	002136			JMP	START	
804	002426	012737	000001	001640	START4:	MOV	#1,PASCNT	
805	002434	005037	177776			CLR	PSW	
806	002440	005037	001456			CLR	RT:NO	
807	002444	104000				TYPE		;CALL FOR PROGRAM NUMBER.
808	002446	016021				PGMSG		
809	002450	004737	003652			JSR	PC,RDOCT	;READ IN PROGRAM NUMBER.
810	002454	012600				MOV	(SP)+,%0	;INPUT DATA TO R0
811	002456	042700	177770			BIC	#177770,%0	;LIMIT (SR) TO BITS 3-0
812	002462	010037	001450			MOV	%0,PRGNUM	;SAVE PROGRAM #
813	002466	006300				ASL	%0	
814	002470	000170	001474			JMP	@PRGTAB(0)	;GO TO SELECTED PROGRAM.
815								
816	002474	013737	001452	001462	GETRDY:	MOV	KSTART,NXTST	;ADDR OF 1ST ROUTINE TO NXTST
817	002502	012737	006042	000004	GTRDYX:	MOV	#ERTP,MACHER	;RESET MACHER TRAP.
818	002510	012737	000040	000006		MOV	#40,MACHER+2	
819	002516	005037	001472			CLR	FMAP	
820	002522	012706	001176			MOV	#SPBOT,%6	;SET BOTTOM OF STACK.
821	002526	104011				SRESET		;ISSUE RESET.
822	002530	005037	177776			CLR	PSW	
823	002534	004737	003032		GTRDYA:	JSR	%7,FORWD	;ROLL FORWARD TO "NEXT" ROUTINE.
824	002540	032777	001000	175422		BIT	#BIT9,@SRPTR	;CHECK SELECT ROUTINE SWITCH
825	002546	001011				BNE	GTRDYC	;BRANCH IF SELECT ROUTINE SWITCH IS SET.
826	002550	005737	001402			TST	UMASK	;C/D DEVICE
827	002554	100003				BPL	GTRDA1	;NO, CONTINUE
828	002556	005737	001456			TST	RTNNO	;THIS A C/D TEST
829	002562	100364				BPL	GTRDYA	;NO, DO NEXT TEST
830	002564	000177	176664		GTRDA1:	JMP	@CURTST	;GO RUN CURRENT ROUTINE.
831	002570	000466				BR	CHNB	;NO GO. MANUAL RTN BYPASSED.
832	002572	017700	175372		GTRDYC:	MOV	@SRPTR,%0	; (SR) TO R0
833	002576	042700	177600			BIC	#177600,%0	;MASK UNDESIRED BITS
834	002602	123700	001456			CMPB	RTNNO,%0	;COMPARE RTNNO TO (R0)
835	002606	001002				BNE	GTRDYD	;BRANCH IF ROUTINE NOT FOUND.
836	002610	000177	176640			JMP	@CURTST	;GO RUN ROUTINE.
837	002614	022737	177777	001462	GTRDYD:	CMP	#-1,NXTST	;NO, CHECK FOR LAST ROUTINE.
838	002622	001344				BNE	GTRDYA	;BRANCH IF NOT LAST ROUTINE.
839	002624	004737	005234			JSR	%7,INCRTN	;YES, INCORRECT ROUTINE SELECTED.
840	002630	000721				BR	GETRDY	;START OVER.


```
841  
842 002632  
843 002632 004737 015204  
844 002636 032777 040000 175324  
845 002644 001403  
846 002646 013716 001466  
847 002652 000002  
848 002654 005737 002110  
849 002660 100011  
850 002662 013746 000004  
851 002666 012737 002774 000004  
852 002674 005737 177060  
853 002700 012637 000004  
854 002704 032777 004000 175256 1$:  
855 002712 001003  
856 002714 005337 001464  
857 002720 001352  
858 002722 022626  
859 002724 032777 002000 175236  
860 002732 001405  
861 002734 013700 001456  
862 002740 042700 100000  
863 002744 000000  
864 002746 032777 001000 175214  
865 002754 001247  
866 002756 022737 177777 001462  
867 002764 001246  
868 002766 004737 005256  
869 002772 000640  
870  
871 002774 022626  
872 002776 012637 000004  
873 003002 000721  
874  
875  
876  
877 003004 005737 001402  
878 003010 100007  
879 003012 052777 000004 176374  
880 003020 104016  
881 003022 002734  
882 003024 005777 176362  
883 003030 000207  
884  
885  
886 003032 013705 001462  
887 003036 012537 001456  
888 003042 012537 001462  
889 003046 012537 001464  
890 003052 012537 001466  
891 003056 010537 001454  
892 003062 000207  
893  
894 003064 011646  
895 003066 162716 000002  
896 003072 017616 000000
```

CHGH3:
CHAINN: JSR PC,CKSWR ;SEE IF A ^G HAS BEEN GIVEN
BIT #BIT14,@SRPTR ;CHECK FOR SCOPE OPTION.
BEQ CHNA ;BRANCH IF SCOPE SW NOT SET.
CHNAB: MOV SCOPTR,@%6 ;SET UP TO RETURN TO ROUTINE.
RTI ;RETURN TO ROUTINE.
CHNA: TST @#XORFLG
BPL 1\$
MOV @#4,-(%6)
MOV #XOR,@#4
TST @#177060 ;TEST FOR XOR
MOV (%6)+,@#4
BIT #BIT11,@SRPTR ;TEST INHIBIT ITERATION SWITCH
BNE CHNAA ;BRANCH IF INHIBIT ITERATION SW SET.
DEC ICTR ;DECREMENT ITERATION COUNT.
BNE CHNAB ;BRANCH IF COUNT NOT 0.
CHNAA: POPSP2 ;POP STACK TWICE
BIT #BIT10,@SRPTR
BEQ CHNB
MOV RTNNO,%0
BIC #BIT15,%0
HALT
CHNB: BIT #BIT9,@SRPTR ;CHECK SELECT ROUTINE SWITCH
BNE GETRDY ;BRANCH IF SELECT RTN SW SET
CMP #-1,NXTST ;LAST TEST?
BNE GTRDYX ;BRANCH IF NOT LAST TEST.
JSR %7,PRGEND ;PROGRAM END.
BR GETRDY
XOR: CMP (%6)+,(%6)+
MOV (%6)+,@#4
BR CHNAB
;
;INIT FOR C/D - WITHOUT JUMPER RESET STARTS ASSEMBLING CHARACTER SETTING DONE
;SET MAINT, DELAY, CLEAR RX DONE
CDINIT: TST UMASK ;C-D DEVICE
BPL CDINX ;NO, EXIT
BIS #BIT2,@TXCSR ;SET MAINT BIT
DELAY ;WAIT 1.5 SEC
1500.
TST @RXBUF ;CLEAR RX DONE
CDINX: RTS %7
;
FORWD: MOV NXTST,%5 ;ADDR OF NEXT ROUTINE TO R5.
MOV (5)+,RTNNO ;GET NEXT ROUTINE NUMBER.
MOV (5)+,NXTST ;GET ADDR OF NEXT "NEXT" ROUTINE.
MOV (5)+,ICTR ;GET ITERATION COUNT.
MOV (5)+,SCOPTR ;GET SCOPE LOOP ENTRY POINTER.
MOV %5,CURTST ;ADDR OF NOW CURRENT TEST TO CURTST.
RTS %7 ;EXIT FORWD SUBROUTINE.
;
EMTINT: MOV @%6,-(6) ;GET SAVED PC.
SUB #2,@%6 ;DECREMENT PC BY 2.
MOV @6),@%6

```
897 003076 006316          EMTA:  ASL      @%6          ; EMT ARG X 2.
898 003100 042716 177001    BIC      #177001,@%6      ; REMOVE 7 MSB.
899 003104 062716 001514    ADD      #EMTTAB,@%6    ; FORM EMT RTN ADDR.
900 003110 017616 000000    MOV      @(%),@%6      ;
901 003114 000136          JMP      @(%)+          ; GO TO EMT ROUTINE.
902
903          ; SAVE REGS 0 TO 4 SUBROUTINE.
904
905 003116 012637 003152    SAVRG:  MOV      (%)+,SVRPC      ; SAVE PC AND PSW.
906 003122 012637 003154    MOV      (%)+,SVRPSW
907 003126 010446          MOV      %4,-(%),      ; SAVE REGS 0 - 4
908 003130 010346          MOV      %3,-(%),      ; IN STACK.
909 003132 010246          MOV      %2,-(%),
910 003134 010146          MOV      %1,-(%),
911 003136 010046          MOV      %0,-(%),
912 003140 013746 003154    MOV      SVRPSW,-(%),   ; RESTORE PC AND PSW.
913 003144 013746 003152    MOV      SVRPC,-(%),
914 003150 000002          RTI
915 003152 000000          SVRPC:  OPEN
916 003154 000000          SVRPSW: OPEN
917
918          ; RESTORE REGS 0 TO 4 SUBROUTINE.
919
920 003156 012637 003212    RSTRG:  MOV      (%)+,RSTPC      ; SAVE PC AND PSW.
921 003162 012637 003214    MOV      (%)+,RSTPSW
922 003166 012600          MOV      (%)+,%0
923 003170 012601          MOV      (%)+,%1
924 003172 012602          MOV      (%)+,%2
925 003174 012603          MOV      (%)+,%3
926 003176 012604          MOV      (%)+,%4
927 003200 013746 003214    MOV      RSTPSW,-(%),   ; RESTORE PC AND PSW.
928 003204 013746 003212    MOV      RSTPC,-(%),
929 003210 000002          RTI
930 003212 000000          RSTPC:  OPEN
931 003214 000000          RSTPSW: OPEN
932
933          ; ROUTINE TO SET RECEIVER INTERRUPT VECTOR AND PRIORITY
934 003216 004737 006424    STLSRV: JSR      %7,TSTVEC
935 003222 017637 000000 003242    MOV      @(%),STPRA+2    ; MOVE VECTOR ADDR TO STPRA+2
936 003230 062716 000002    ADD      #2,@%6          ; SET UP EXIT
937 003234 013701 001420    MOV      RXVTR,%1
938 003240 012721 000000    STPRA:  MOV      #OPEN,(1)+   ; SET VECTOR ADDRESS
939 003244 013721 001422    MO.     RXLVL,(1)+       ; SET PRIORITY
940 003250 000002          RTI
941
942          ; ROUTINE TO SET TRANSMITTER INTERRUPT VECTOR AND PRIORITY.
943 003252 004737 006424    STLSPV: JSR      %7,TSTVEC
944 003256 017637 000000 003276    MOV      @(%),STPPA+2    ; MOVE VECTOR ADDR TO STPPA+2
945 003264 062716 000002    ADD      #2,@%6          ; SET UP EXIT
946 003270 013701 001424    MOV      TXVTR,%1
947 003274 012721 000000    STPPA:  MOV      #OPEN,(1)+   ; SET VECTOR ADDRESS.
948 003300 013721 001426    MOV      TXLVL,(1)+       ; SET PRIORITY
949 003304 000002          RTI
950
951          ; ROUTINE TO ISSUE RESET.
952 003306 012700 052525    SRSETT: MOV      #52525,%0    ; DATA TO R0.
```

```
953 003312 005100          COM      %0          ;COMPLEMENT (R0).
954 003314 010037 003310  MOV      %0,SRSETT+2 ;(R0) TO SRSETT+2.
955 003320 000005          RESET    ;ISSUE RESET. (R0) IS
956 003322 000007          RTI      ;DISPLAYED. EXIT.
957
958          ;RANDOM NUMBER GENERATOR. ROUTINE EXITS WITH NUMBER IN REGISTER 0.
959 003324 013700 003372  RNGEN:  MOV      RP1,%0
960 003330 006100          ROL      %0
961 003332 006100          ROL      %0
962 003334 063700 003374  ADD      RP2,%0
963 003340 010037 003372  MOV      %0,RP1
964 003344 006100          ROL      %0
965 003346 006100          ROL      %0
966 003350 063700 003374  ADD      RP2,%0
967 003354 006100          ROL      %0
968 003356 006100          ROL      %0
969 003360 010037 003374  MOV      %0,RP2
970 003364 013700 003372  MOV      RP1,%0
971 003370 000207          RTS      %7          ;EXIT. NUMBER IN R0
972 003372 001233          RP1:    1233
973 003374 007622          RP2:    7622
974
975          ;CLRCD - CLEAR CURRENT DEVICE PARAMETERS
976 003376 005037 001416  CLRCD:  CLR      TXBUF
977 003402 005037 001414  CLR      TXCSR
978 003406 005037 001410  CLR      RXCSR
979 003412 005037 001412  CLR      RXBUF
980 003416 005037 001420  CLR      RXVTR
981 003422 005037 001424  CLR      TXVTR
982 003426 005037 001422  CLR      RXLVL
983 003432 005037 001426  CLR      TXLVL
984 003436 000207          RTS      %7
985
986          ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
987 003440 011600          TYP:    MOV      @%6,%0          ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS.
988 003442 062716 000002  ADD      #2,@%6          ;SET UP EXIT.
989 003446 011000          MOV      @%0,%0          ;ADDRESS OF MESSAGE TO R0.
990 003450 112037 003560  TYPA:  MOVVB   (0)+,TYPDAT      ;GET CHARACTER
991 003454 122737 000100 003560  CMPB    #100,TYPDAT      ;CHECK FOR '@' CHARACTER
992 003462 001001          BNE     TYPC            ;BRANCH IF NOT '@'.
993 003464 000002          RTI     ;TERMINATOR CHAR. DONE. EXIT.
994 003466 122737 000045 003560  TYPC:  CMPB    #45,TYPDAT      ;CHECK FOR 'X'.
995 003474 001416          BEQ     TYPF            ;BRANCH IF 'X'.
996 003476 122737 000043 003560  CMPB    #43,TYPDAT      ;NOT 'X'. CHECK FOR '#'.
997 003504 001417          BEQ     TYPG            ;BRANCH IF '#'.
998 003506 004737 003514  JSR     %7,TYPD          ;TYPE CHAR IN TYPDAT
999 003512 000756          BR      TYPA
1000 003514 113777 003560 175714  TYPD:  MOVVB   TYPDAT,@TPB      ;OUTPUT CHARACTER TO PRINTER
1001 003522 105777 175706  TSTB   @TPS            ;WAIT FOR DONE FLAG.
1002 003526 100375          BPL     -4
1003 003530 000207          RTS     %7          ;EXIT
1004 003532 112737 000015 003560  TYPF:  MOVVB   #15,TYPDAT      ;MOVE CARRIAGE RETURN CODE TO TYPDAT
1005 003540 004737 003514  JSR     %7,TYPD          ;GO TYPE CHAR.
1006 003544 112737 000012 003560  TYPG:  MOVVB   #12,TYPDAT      ;MOVE LF CODE TO TYPDAT.
1007 003552 004737 003514  JSR     %7,TYPD          ;GO TYPE CHAR.
1008 003556 000734          BR      TYPA
```

```
1009 003560 000000 TYPDAT: OPEN
1010
1011 :SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER
1012 003562 011600 TYPDAT: OPEN
1013 003564 062716 000002 TYPDAT: OPEN
1014 003570 011037 003610 TYPDAT: OPEN
1015 003574 022737 177777 003610 TYPDAT: OPEN
1016 003602 001001 TYPDAT: OPEN
1017 003604 000002 TYPDAT: OPEN
1018 003606 104000 TYPDAT: OPEN
1019 003610 000000 TYPDAT: OPEN
1020 003612 000763 TYPDAT: OPEN
1021
1022 :OVERLAY VECTOR AREA
1023 003614 012701 000300 OVRLAY: MOV #300,%1 ;GET DL11-E VECTOR BASE ADDRESS
1024 003620 012702 000302 OVRLAY: MOV #302,%2
1025 003624 012703 000004 OVRLAY: MOV #4,%3
1026 003630 010221 OVRLAY: MOV %2,(1)+ ;LOAD VECTOR WITH IOT ERROR TRAP
1027 003632 010321 OVRLAY: MOV %3,(1)+
1028 003634 062702 000004 OVRLAY: ADD #4,%2
1029 003640 020127 001000 OVRLAY: CMP %1,#1000 ;ALL VECTORS BEEN LOADED
1030 003644 001401 OVRLAY: BEQ OVRLYB
1031 003646 000770 OVRLAY: BR OVRLYA
1032 003650 000207 OVRLYB: RTS 7 ;EXIT
1033
1034 :SUBROUTINE TO READ OCTAL DATA FROM THE TELETYPE PRINTER
1035 003652 011646 RDOCT: MOV (SP),-(SP) ;MAKE ROOM FOR DATA WORD
1036 003654 010046 RDOCT: MOV %0,-(SP) ;SAVE R0
1037 003656 010146 RDOCT: MOV %1,-(SP) ;SAVE R1
1038 003660 005001 !NDAT: CLR %1 ;CLEAR DATA WORD
1039 003662 005037 001626 !NDAT: CLR COUNT ;SET NO. OF DIGITS - 0
1040 003666 105777 175536 RDDAT: TSTB @TKS ;TEST TTY READ STATUS
1041 003672 100375 RDDAT: BPL RDDAT ;WAIT
1042 003674 117746 175532 RDDAT: MOVB @TKB,-(SP) ;PUSH DIGIT ON STACK
1043 003700 042716 177600 RDDAT: BIC #177600,(SP) ;:***G
1044 003704 105777 175524 ECDAT: TSTB @TPS ;TEST TTY PRINT STATUS
1045 003710 100375 ECDAT: BPL ECDAT ;WAIT
1046 003712 111677 175520 ECDAT: MOVB (SP),@TPB ;ECHO CHARACTER
1047 003716 122716 000015 ECDAT: CMPB #15,(SP) ;IS IT A TERMINATOR?
1048 003722 001432 ECDAT: BEQ RETRN ;BR IF YES
1049 003724 122716 000177 ECDAT: CMPB #177,(SP) ;IS IT A RUBOUT?
1050 003730 001423 ECDAT: BEQ RREAD ;BR IF YES
1051 003732 122716 000060 ECDAT: CMPB #60,(SP) ;IS IT AN OCTAL DIGIT?
1052 003736 003020 ECDAT: BGT RREAD ;BR IF NO
1053 003740 122716 000067 ECDAT: CMPB #67,(SP) ;TEST AGAIN
1054 003744 002415 ECDAT: BLT RREAD ;BR IF NO
1055 003746 005237 001626 ECDAT: INC COUNT ;INC NO. OF DIGITS
1056 003752 022737 000007 001626 ECDAT: CMP #7,COUNT ;MORE THAN SIX DIGITS? :***G
1057 003760 003407 ECDAT: BLE RREAD ;BR IF YES
1058 003762 006301 ECDAT: ASL %1 ;CLEAR LOWEST THREE BITS
1059 003764 006301 ECDAT: ASL %1 ;OF DATA WORD
1060 003766 006301 ECDAT: ASL %1
1061 003770 162716 000060 ECDAT: SUB #60,(SP) ;CONVERT TO BINARY
1062 003774 062601 ECDAT: ADD (SP),%1 ;ADD DIGIT TO DATA WORD
1063 003776 000733 ECDAT: BR RDDAT ;GET NEXT DIGIT
1064 004000 104000 RREAD: TYPE ;TELL USER ABOUT ILLEGAL CHARACTER
```



```
1065 004002 020142          DIERR
1066 004004 005726          TST      (SP)+          ;GET RID OF ILLEGAL CHARACTER
1067 004006 000724          BR       INDAT          ;START SUBROUTINE AGAIN
1068 004010 010166 000010    RETRN:  MOV      %1,10(SP) ;STORE DATA WORD ON STACK
1069 004014 005726          TST      (SP)+          ;INC STACK POINTER
1070 004016 012601          MOV      (SP)+,%1       ;RESTORE R1
1071 004020 012600          MOV      (SP)+,%0       ;RESTORE R0
1072 004022 000207          RTS       PC           ;RETURN
1073
1074          ;SUBROUTINE TO DELAY A SPECIFIED NUMBER OF MILLISECONDS
1075 004024 011637 004070    DLY:    MOV      @%6,DLCNT ;GET DELAY COUNT ADDRESS.
1076 004030 062716 000002          ADD      #2,@%6         ;SET UP EXIT ADDRESS
1077 004034 017746 000030          MOV      @DLCNT,-(6)    ;DELAY COUNT TO STACK
1078 004040 001411          BEQ      DLYC
1079 004042 005037 177776          CLR      PSW           ;SET PRIORITY 0
1080 004046 012746 000226    DLYA:  MOV      #226,-(6)  ;1 MSEC COUNT TO STACK
1081 004052 005316    DLYB:  DEC      @%6         ;DECREMENT 1 MSEC COUNT
1082 004054 001376          BNE      DLYB          ;BRANCH IF NOT 0.
1083 004056 005726          POPSP
1084 004060 005316          DEC      @%6         ;ZERO. UNCOVER MSECS. COUNT.
1085 004062 001371          BNE      DLYA         ;DECREMENT IT
1086 004064 005726    DLYC:  POPSP
1087 004066 000002          RTI
1088 004070 000000    DLCNT: OPEN          ;EXIT.
1089          ;CONTAINS MILLISECONDS COUNT ADDRESS.
1090          ;SUBROUTINE TO STALL A RANDOM NUMBER OF MILLISECONDS. MAXIMUM STALL
1091          ;DETERMINED BY CONTENTS OF LOC STLMSK.
1092 004072 004737 003324    STAL:  JSR      %7,RNGEN  ;GO GET RANDOM NUMBER.
1093 004076 043700 001406          BIC      STLMSK,%0     ;# IN R0. APPLY STALL MASK.
1094 004102 001404          BEQ      STALB        ;BRANCH IF RESULT IS 0.
1095 004104 010037 004112          MOV      %0,STALA
1096 004110 104016          DELAY
1097 004112 000000    STALA: OPFN          ;DELAY
1098 004114 000002    STALB: RTI           ;DELAY COUNT
1099          ;DONE. EXIT.
1100          ;SUBROUTINE TO GENERATE RANDOM CHARACTER COUNT
1101 004116 004737 003324    GRCNT: JSR      %7,RNGEN  ;GET RANDOM NUMBER
1102 004122 043700 004136          BIC      RCMSK,%0     ;APPLY MASK
1103 004126 001773          BEQ      GRCNT        ;TRY AGAIN IF RESULT 0
1104 004130 010037 004140          MOV      %0,RNCNT    ;COUNT TO RNCNT
1105 004134 000207          RTS       %7         ;EXIT.
1106 004136 000000    RCMSK: OPEN          ;RANDOM CHARACTER MASK.
1107 004140 000000    RNCNT: OPEN          ;RANDOM CHARACTER COUNT.
1108
1109          ;SUBROUTINE TO SKIP CN FLAG AND TIME OUT IF SKIP FAILS
1110 004142 013737 001410 004210 TMRX:  MOV      RXCSR,S1OT  ;SET UP RXCSR ADDRESS
1111 004150 000403          BR       TIME1
1112 004152 013737 001414 004210 TMTX:  MOV      TXCSR,S1OT  ;SET UP TXCSR ADDRESS
1113 004160 005037 004206    TIME1: CLR      TIMER
1114 004164 005237 004206    TIME2: INC      TIMER
1115 004170 001405          BEQ      TIMEX        ;BRANCH IF COUNTER OVERFLOW
1116 004172 105777 000012          TSTB     @S1OT
1117 004176 100372          BPL      TIME2
1118 004200 062716 000002          ADD      #2,@%6         ;SET UP EXIT RETURN
1119 004204 000002    TIMEX: RTI
1120 004206 000000    TIMER: 0
```

```

1121 004210 000000          SIOT: 0
1122                      ;*****
1123
1124                      ;ROUTINE TO PRINT A CARRIAGE RETURN AND A LINE FEED
1125
1126 004212          CHGH7:
1127 004212 105777 175216  $SCRLF: TSTB  @TPS          ; IS PRINTER READY
1128 004216 100375          BPL  $SCRLF          ; TRY AGAIN IS SO
1129 004220 112777 000015 175210  MOVB  #15,@TPB          ; PRINT <CR>
1130 004226 105777 175202 1$:  TSTB  @TPS          ; IS PRINTER BUSY
1131 004232 100375          BPL  1$          ; TRY AGAIN IF SO
1132 004234 112777 000012 175174  MOVB  #12,@TPB          ; PRINT <LF>
1133 004242 000002          RTI          ; RETURN TO INTERRUPT
1134                      ;*****
1135
1136                      ;SUBROUTINE TO SELECT LINE
1137 004244 032777 010000 173716  LINSXL: BIT  #BIT12,@SRPTR
1138 004252 001003          BNE  LINSXL          ; BRANCH IF SET
1139 004254 005037 001614          CLR  FOUNDV
1140 004260 000205          RTS  5
1141 004262 004737 003614  LINSXL: JSR  %7,OVRLAY
1142 004266 004737 003376          JSR  %7,CLRCD
1143 004272 104000          TYPE
1144 004274 017546          LDLINE
1145 004276 004737 003652          JSR  PC,RDOCT
1146 004302 012637 001620          MOV  (SP)+,TEMP
1147 004306 042737 177740 001620  BIC  #177740,TEMP
1148 004314 013737 001620 001616  MOV  TEMP,LINENO          ; SAVE FOR TYPING
1149 004322 006337 001620          ASL  TEMP
1150 004326 013701 001620          MOV  TEMP,%1
1151 004332 016101 001200          MOV  RXCR0(1),%1          ; GET RXCSR DEVICE ADDRESS
1152 004336 032701 000001          BIT  #BIT0,%1          ; IS DEVICE THERE
1153 004342 001403          BEQ  LINB          ; YES
1154 004344 104000          LINA: TYPE          ; NO. REPORT
1155 004346 020077          MNOLIN
1156 004350 000744          BR  LINSXL
1157 004352 004737 006226  LINB: JSR  %7,FORMAD
1158 004356 005037 177776          CLR  PSW
1159 004362 052737 000001 001472  BIS  #BIT0,FMAP          ; SET MAPPING FLAG
1160 004370 042777 000100 175016  BIC  #BIT6,@TXCSR
1161 004376 052777 000100 175010  BIS  #BIT6,@TXCSR
1162 004404 000240          NOP
1163 004406 000240          NOP
1164 004410 005737 001420          TST  RXVTR
1165 004414 001753          BEQ  LINA
1166 004416 042777 000100 174770  BIC  #BIT6,@TXCSR
1167 004424 012737 000340 177776  MOV  #PRTY7,PSW
1168 004432 004537 004624          JSR  5,OACNV          ; TYPE LINE #
1169 004436 001616          LINENO
1170 004440 017605          SELINE
1171 004442 000002          2
1172 004444 104000          TYPE
1173 004446 017574          ALINE
1174 004450 000205          RIS  5
1175
1176                      ;SUBROUTINE TO INITIALIZE BINARY COUNT PATTERNS

```

```

1177 004452 012737 177777 004474 INBIN: MOV #1,RIND ;SET ALL VARIABLES
1178 004460 004537 004712 JSR #5,BMOVE ;TO MINUS 1.
1179 004464 004474 RIND
1180 004466 004475 RIND+1
1181 004470 000013 11.
1182 004472 000207 RTS #7 ;EXIT
1183 004474 000000 RIND: OPEN
1184 004476 000000 PTO: OPEN
1185 004500 000000 PT1: OPEN
1186 004502 000000 PIND: OPEN
1187 004504 000000 PTOP: OPEN
1188 004506 000000 PTIP: OPEN
1189
1190 ;SPECIAL BINARY COUNT PATTERN SUBROUTINE. EXITS WITH BIN CHAR IN RO
1191 004510 013737 004476 004500 GTBIN: MOV PTO,PT1 ;PREVIOUS BIN CHAR TO PT1
1192 004516 005137 004500 COM PT1
1193 004522 005137 004474 COM RIND
1194 004526 001002 BNE .+6
1195 004530 005237 004500 INC PT1
1196 004534 042737 177400 004500 BIC #177400,PT1 ;MASK TO 8 BITS
1197 004542 013737 004500 004476 MOV PT1,PTO ;SAVE BIN CHAR IN PTO
1198 004550 013700 004500 MOV PT1,#0 ;BIN CHAR TO RO.
1199 004554 000207 RTS #7 ;EXIT.
1200 004556 013737 004504 004506 GTBINP: MOV PTOP,PTIP ;PREVIOUS BIN CHAR TO PTIP
1201 004564 005137 004506 COM PTIP
1202 004570 005137 004502 COM PIND
1203 004574 001002 BNE .+6
1204 004576 005237 004506 INC PTIP
1205 004602 042737 177400 004506 BIC #177400,PTIP ;MASK TO 8 BITS.
1206 004610 013737 004506 004504 MOV PTIP,PTOP ;SAVE BIN CHAR IN PTOP.
1207 004616 013701 004506 MOV PTIP,#1 ;BIN CHAR TO RI.
1208 004622 000207 RTS #7 ;EXIT.
1209
1210 ;OCTAL TO ASCII CONVERT ROUTINE
1211 004624 104013 OACNV: SAVREG
1212 004626 013537 004710 MOV @5+,OACNVX ;GET OCTAL VALUE.
1213 004632 012501 MOV 5+,#1 ;GET DESTINATION ADDR.
1214 004634 012502 MOV 5+,#2 ;GET CONVERT COUNT.
1215 004636 060201 ADD #2,#1 ;DEVELOP ADDR TO STORE 1ST CHAR.
1216 004640 013703 004710 OACNVA: MOV OACNVX,#3
1217 004644 042703 177770 BIC #177770,#3 ;ISOLATE LEAST SIGNIFICANT DIGIT.
1218 004650 062703 000060 ADD #60,#3 ;CONVERT DIGIT TO ASCII.
1219 004654 110341 MOVB #3,-(1) ;STORE ASCII CHARACTER.
1220 004656 042737 000007 004710 BIC #7,OACNVX
1221 004664 006037 004710 ROR OACNVX
1222 004670 006037 004710 ROR OACNVX
1223 004674 006037 004710 ROR OACNVX
1224 004700 005302 DEC #2 ;DONE ALL DIGITS?
1225 004702 001356 BNE OACNVA ;BRANCH IF NOT DONE.
1226 004704 104014 RSTREG
1227 004706 000205 RTS #5 ;DONE. EXIT.
1228 004710 000000 OACNVX: OPEN
1229
1230 ;SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES.
1231 004712 104013 BMOVE: SAVREG ;SAVE REGS.
1232 004714 012501 MOV 5+,#1 ;GET 'FROM' ADDRESS

```

1233	004716	012502				MOV	(5)+,%2		;GET 'TO' ADDRESS
1234	004720	012503				MOV	(5)+,%3		;GET COUNT
1235	004722	112122			BMOVA:	MOVB	(1)+,(2)+		;MOVE BYTE
1236	004724	005303				DEC	%3		;DECREMENT COUNT
1237	004726	001375				BNE	BMOVA		;BRANCH IF NOT DONE.
1238	004730	104014				RSTREG			;RESTORE REGS.
1239	004732	000205				RTS	%5		;DONE EXIT
1240									
1241									
1242	004734	104013							
1243	004736	012700	005112			BDCNV:	SAVREG		
1244	004742	013501				MOV	#DECVAL,%0		;SET UP ADDR TO STORE DECIMAL ASCII IN R0
1245	004744	012537	005022			MOV	@(5)+,%1		;BINARY VALUE TO R1.
1246	004750	012537	005024			MOV	(5)+,BDCNV		;GET DEST ADDR
1247	004754	012702	005100			MOV	(5)+,BDCNVD		;GET CHAR COUNT
1248	004760	012737	000005	005072		MOV	#ADTENP,%2		;ADDR OF TEN POWER STRING TO R2.
1249	004766	012237	005076			MOV	#5,CNVCTR		;SET UP FOR 5 POWER CONVERSIONS.
1250	004772	004737	005032			BDCNVA:	MOV	(2)+,TENPWR	;MOVE POWER OF TEN VALUE TO TENPWR.
1251	004776	005337	005072				JSR	%7,SUBTEN	;PERFORM CONVERSION
1252	005002	001371					DEC	CNVCTR	;DONE 5 CONVERSIONS?
1253	005004	163700	005024				BNE	BDCNVA	;BRANCH IF NOT YET 5.
1254	005010	010037	005020				SUB	BDCNVD,%0	
1255	005014	004537	004712				MOV	%0,BDCNVB	
1256	005020	000000					JSR	%5,BMOVE	
1257	005022	000000				BDCNVB:	0		
1258	005024	000000				BDCNVC:	0		
1259	005026	104014				BDCNVD:	0		
1260	005030	000205					RSTREG		
1261	005032	005037	005074				RTS	%5	;YES, EXIT.
1262	005036	163701	005076			SUBTEN:	CLR	DIGIT	;CLEAR DIGIT
1263	005042	103403				SUBTNA:	SUB	TENPWR,%1	;SUBTRACT TEN POWER FROM BINARY VALUE.
1264	005044	005237	005074				BCS	SUBTNB	;BRANCH IF UNSUCCESSFUL SUBTRAC ION.
1265	005050	000772					INC	DIGIT	
1266	005052	063701	005076				BR	SUBTNA	
1267	005056	062737	000060	005074		SUBTNB:	ADD	TENPWR,%1	;RESTORE SUBTRACTED VALUE.
1268	005064	113720	005074				ADD	#60,DIGIT	;CONVERT (DIGIT) TO ASCII
1269	005070	000207					MOVB	DIGIT,(0)+	;MOVE ASCII CHAR TO DECVAL FIELD.
1270	005072	000000					RTS	%7	;EXIT.
1271	005074	000000				CNVCTR:	OPEN		
1272	005076	000000				DIGIT:	OPEN		
1273	005100	023420				TENPWR:	OPEN		
1274	005102	001750				ADTENP:	10000.		
1275	005104	000144					1000.		
1276	005106	000012					100.		
1277	005110	000001					10.		
1278	005112	040	040	040		DECVAL:	.BYTE	040,040,040,040,040,040	
1279	005115	040	040	040					
1280	005120	042777	000002	174262		DATTST:	BIC	#BIT1,@RXCSR	;CLEAR DATA TERM. READY
1281	005126	052777	000004	174260			BIS	#BIT2,@TXCSR	;SET MAINTENANCE BIT
1282	005134	012737	000144	001566			MOV	#100.,CTRO	;GET CHARACTER COUNT
1283	005142	105777	174246			DATAA:	TSTB	@TXCSR	;WAIT FOR
1284	005146	100375					BPL	.-4	;READY FLAG
1285	005150	004737	004556				JSR	7,GTBINP	;GET CHARACTER
1286	005154	110137	001562				MOVB	%1,CRBUFA	;MOVE CHARACTER
1287	005160	004737	005530				JSR	7,MASKIT	;MASK OFF NON TRANSMITTED BITS
1288	005164	110177	174226				MOVB	%1,@TXBUF	;TRANSMIT CHARACTER

1289	005170	105777	174214		TSTB	@RXCSR		;WAIT FOR
1290	005174	100375			BPL	.-4		;DONE FLAG
1291	005176	117737	174210	001560	MOVB	@RXBUF,CRBUF		;GET RECEIVED CHARACTER
1292	005204	104004			DATCHK			;CHK DATA
1293	005206	005337	001566		DEC	CTRO		;DECREMENT CHARACTER COUNT
1294	005212	001353			BNE	DATAA		
1295	005214	005726			TST	(6)+		;POP STACK
1296	005216	104012			SCOPE			
1297								
1298	005220	104000			SETS: TYPE			;TYPE SELECT OPTION MESSAGE.
1299	005222	016763			ASETSR			
1300	005224	105777	174200		CHGH8: TSTB	@TKS		;SEE IF ANY INPUT
1301	005230	100375			BPL	CHGH8		;WAIT FOR INPUT
1302	005232	000207			RTS	%7		;EXIT.
1303	005234	104000			INCRTN: TYPE			;TYPE INCORRECT ROUTINE SELECTED.
1304	005236	017105			AINCRT			
1305	005240	000000			HALT			;COMMON HALT.
1306	005242	000207			RTS	%7		;EXIT.
1307	005244	104000			INCRPG: TYPE			
1308	005246	017226			AINCPCG			
1309	005250	000000			HALT			
1310	005252	000137	002136		JMP	START		
1311	005256	005037	001614		PRGEND: CLR	FOUNOV		
1312	005262	032777	020000	172700	BIT	#BIT13,@SRPTR		;INHIBIT PRINT SET?
1313	005270	001026			BNE	PRGEXT		;BR IF SET
1314	005272	004537	004734		JSR	%5,BDCNV		
1315	005276	001640			PASCNT			
1316	005300	017276			APCNT			
1317	005302	000006			6			
1318	005304	004537	004624		JSR	%5,OACNV		;CONVERT LINE NUMBER
1319	005310	001616			LINENO			
1320	005312	017316			ACLIN			
1321	005314	000002			2			
1322	005316	004537	004624		JSR	%5,OACNV		;CONVERT RXCSR
1323	005322	001410			RXCSR			
1324	005324	017332			APRXC			
1325	005326	000006			6			
1326	005330	004537	004624		JSR	%5,OACNV		;CONVERT VECTOR
1327	005334	001420			RXVTR			
1328	005336	017353			APVEC			
1329	005340	000004			4			
1330	005342	104000			TYPE			;TYPE PROGRAM END.
1331	005344	017261			APGEND			
1332	005346	032777	010000	172614	PRGEXT: BIT	#BIT12,@SRPTR		;LOCK ON LINE
1333	005354	001403			BEQ	PRGXT1		;BR IF NOT SET
1334	005356	005237	001640		INC	PASCNT		
1335	005362	000425			BR	PRGXTL		
1336	005364	013737	001616	001620	PRGXT1: MOV	LINENO,TEMP		;GET LINENO
1337	005372	006337	001620		ASL	TEMP		
1338	005376	062737	000002	001620	PRGEC: ADD	#2,TEMP		;UPDATE LINE NUMBER
1339	005404	013701	001620		PRGEA: MOV	TEMP,%1		
1340	005410	016101	001200		MOV	RXCRO(1),%1		;GET RXCSR DEVICE ADDRESS
1341	005414	022701	177777		CMP	#177777,%1		;LAST ONE
1342	005420	001023			BNE	PRGEB		;NO,CONTINUE
1343	005422	005237	001640		INC	PASCNT		
1344	005426	005037	001616		CLR	LINENO		

```

1345 005432 005037 001620
1346 005436 013705 000042
1347 005442 001405
1348 005444 000005
1349 005446 004715
1350 005450 000240
1351 005452 000240
1352 005454 000240
1353 005456 032777 010000 172504
1354 005464 001747
1355 005466 000207
1356 005470 032701 000001
1357 005474 001340
1358 005476 006237 001620
1359 005502 013737 001620 001616
1360 005510 004737 006226
1361 005514 000207
1362
1363
1364 005516 005777 172446
1365 005522 100001
1366 005524 000000
1367 005526 000002
1368
1369
1370 005530 013737 001402 001404
1371 005536 042737 177000 001404
1372 005544 005137 001404
1373 005550 043737 001404 001562
1374 005556 000207
1375
1376
1377 005560 017737 173626 001564
1378 005566 032737 170000 001564
1379 005574 001004
1380 005576 023737 001560 001562
1381 005604 001421
1382 005606 004537 004624
1383 005612 001560
1384 005614 016740
1385 005616 000003
1386 005620 004537 004624
1387 005624 001562
1388 005626 016727
1389 005630 000003
1390 005632 004537 004624
1391 005636 001564
1392 005640 016753
1393 005642 000006
1394 005644 104015
1395 005646 016715
1396 005650 000002
1397
1398
1399 005652 012737 177777 006022
1400 005660 012737 000240 006024

      CLR      TEMP
      PRGXTL: MOV   @#42,%5
              BEQ   CONT
              RESET
      LOGIC:  JSR   7,(5)
              NOP
              NOP
              NOP
      CONT:   BIT   #BIT12,@SRPTR ;LOCK ON LINE
              BEQ   PRGEA ;BRANCH IF NOT SET
              RTS   7
      PRGEB:  BIT   #BIT0,%1 ;DEVICE THERE
              BNE   PRGEC ;NO
              ASR   TEMP
              MOV   TEMP,LINENO
              JSR   %7,FORMAD
              RTS   %7 ;EXIT.

      ;CONDITIONAL ERROR HALT ROUTINE.
      EHLT:   TST   @SRPTR ;CHECK FOR HALT ON ERROR.
              BPL   EHLTA ;BRANCH IF NO HALT DESIRED.
              HALT ;HALT.
      EHLTA:  RTI ;IN DATA LIGHTS.

      ;MASKIT - MASK DATA ACCORDING TO LINE NUMBER
      MASKIT: MOV   UMASK,RMASK ;GET MASK
              BIC   #177000,RMASK ;REMOVE C/D FLAG+PRIORITY
              COM   RMASK
              BIC   RMASK,CRBUFA ;MASK DESIRED BITS
              RTS   7

      ;DATA CHECK ROUTINE, TEST ERROR BITS
      DTCHK:  MOV   @RXBUF,CRBUFB ;DID ANY ERROR BITS SET
              BIT   #170000,CRBUFB
              BNE   DTCHKX ;YES, TYPE ERROR
              CMP   CRBUF,CRBUFA ;COMPARE EXPECTED AND RECEIVED
              BEQ   DTCHKA ;CHARS. BRANCH IF SAME.
      DTCHKX: JSR   %5,OACNV ;GO TO OCTAL TO ASCII CONVERT.
              CRBUF ;SOURCE ADDR.
              AWAS ;DESTINATION ADDR.
              3 ;#OF DIGITS TO CONVERT.
              JSR   %5,OACNV ;GO TO OCTAL TO ASCII CONVERT.
              CRBUFA ;SOURCE ADDR.
              AASB ;DESTINATION ADDR.
              3 ;#OF DIGITS TO CONVERT.
              JSR   %5,OACNV
              CRBUFB
              ARXBUF
              6
              ERROR1
              ERDAT
      DTCHKA: RTI

      ;ERROR HANDLER
      ERR:    MOV   #-1,ERRB ;SET UP ONE MESSAGE CALL.
              MOV   #240,ERRB+2
  
```



```

1401 005666 005037 006040          CLR      ERRE
1402 005672 000413          BR       ERRA
1403 005674 011637 006022          MOV      @%6,ERRB      ;DEVELOP ADDT'L MESSAGE ADDR.
1404 005700 017737 000116 006022          MOV      @ERRB,ERRB    ;STORE AT ERRB.
1405 005706 012737 177777 006024          MOV      #-1,ERRB+2
1406 005714 012737 000002 006040          MOV      #2,ERRE
1407 005722 032777 020000 172240          BIT      #BIT13,@SRPTR ;INHIBIT ERROR PRINT?
1408 005730 001036          BNE     ERRC           ;BRANCH TO INHIBIT PRINT.
1409 005732 011637 006036          MOV      @%6,ERRD      ;DEVELOP CALLING ADDR.
1410 005736 162737 000002 006036          SUB      #2,ERRD
1411 005744 013737 001456 001460          MOV      RTNNO,TNNO
1412 005752 042737 100000 001460          BIC      #BIT15,TNNO
1413 005760 004537 004624          JSR      %5,OACNV      ;GO TO OCTAL TO ASCII CONVERT.
1414 005764 006036          ERD      ;SOURCE ADDR.
1415 005766 016104          APC      ;DESTINATION ADDR.
1416 005770 000006          6        ;#OF DIGITS TO CONVERT.
1417 005772 004537 004624          JSR      %5,OACNV      ;GO TO OCTAL TO ASCII CONVERT.
1418 005776 001410          RXCSR    ;SOURCE ADDR.
1419 006000 016123          MRXNUM   ;DESTINATION ADDR.
1420 006002 000006          6        ;#OF DIGITS TO CONVERT.
1421 006004 004537 004624          JSR      %5,OACNV      ;GO TO OCTAL TO ASCII CONVERT.
1422 006010 001456          RTNNO    ;SOURCE ADDR.
1423 006012 016074          ATNUMB   ;DESTINATION ADDR.
1424 006014 000003          3        ;#OF DIGITS TO CONVERT.
1425 006016 104001          TYPES    ;TYPE:
1426 006020 016072          EMO      ;ERROR HEADER,
1427 006022 000000          ERRB:    OPEN        ;ADDT'L ERROR MESSAGE IF ANY.
1428 006024 177777          -1
1429 006026 104010          ERRC:    EHALT       ;GO ERR HALT IF DESIRED.
1430 006030 063716 006040          ADD      ERRE,@%6
1431 006034 000002          RTI      ;EXIT.
1432 006036 000000          ERD:    OPEN
1433 006040 000000          ERRE:    OPEN
1434
1435          ;
1436 006042 013737 177776 001470          ;ERROR TRAP HANDLER - TYPE TO AND FROM WHERE ERROR TRAP OCCURRED
1437 006050 012737 000340 177776          ERTP:    MOV      PSW,OLDPS      ;SAVE OLD STATUS
1438 006056 006237 001470          MOV      #PRTY7,PSW
1439 006062 006237 001470          ASR      OLDPS
1440 006066 006237 001470          ASR      OLDPS
1441 006072 042737 177740 001470          ASR      OLDPS
1442 006100 013737 001470 001634          BIC      #177740,OLDPS
1443 006106 011637 001636          MOV      OLDPS,TOPC
1444 006112 004537 004624          ERTPA:   MOV      @%6,FROMPC      ;GET FROM PC
1445 006116 001634          JSR      %5,OACNV
1446 006120 020034          TOPC
1447 006122 000006          MTO
1448 006124 004537 004624          6        JSR      %5,OACNV
1449 006130 001636          FROMPC
1450 006132 020066          MFROM
1451 006134 000006          6
1452 006136 104000          TYPE
1453 006140 017767          MTERR
1454 006142 000000          HALT
1455 006144 000137 002136          JMP      START
1456

```

```
1457
1458 006150 011637 001634 ;MAPVEC - MAP VECTOR OR REPORT ERROR DEPENDING ON FMAP FLAG
1459 006154 022626 MAPVEC: MOV @%6, TOPC
1460 006156 011637 001636 POPSP2
1461 006162 162737 000004 001634 MOV @%6, FROMPC
1462 006170 005737 001472 SUB #4, TOPC
1463 006174 001746 TST FMAP
1464 006176 013737 001634 001424 BEQ ERTPA ;NOT MAPPING, REPORT ERROR
1465 006204 162737 000004 001634 MOV TOPC, TXVTR ;STORE VECTOR
1466 006212 013737 001634 001420 SUB #4, TOPC
1467 006220 005037 001472 MOV TOPC, RXVTR
1468 006224 000002 CLR FMAP
1469
1470 ;
1471 006226 010137 001410 ;FORMAD-FORM DEVICE AT ADDRESSES
1472 006232 062701 000002 FORMAD: MOV %1, RXCSR
1473 006236 010137 001412 ADD #2, %1
1474 006242 062701 000002 MOV %1, RXBUF
1475 006246 010137 001414 ADD #2, %1
1476 006252 062701 000002 MOV %1, TXCSR
1477 006256 010137 001416 ADD #2, %1
1478 006262 013737 001616 001620 MOV %1, TXBUF
1479 006270 006337 001620 MOV LINENO, TEMP ;GET PRIORITY
1480 006274 062737 001302 001620 ASL TEMP
1481 006302 017737 173312 001622 ADD #CMASO, TEMP
1482 006310 013737 001622 001402 MOV @TEMP, TEMP1
1483 006316 000337 001622 MOV TEMP1, UMASK
1484 006322 006337 001622 SWAB TEMP1
1485 006326 042737 177437 001622 ASL TEMP1
1486 006334 013737 001622 001422 BIC #177437, TEMP1
1487 006342 013737 001622 001426 MOV TEMP1, RXLVL
1488 006350 000207 MOV TEMP1, TXLVL
1489
1490 ;
1491 ;DOTHIS - SELECTABLE TEST DECISION MAKER
1492 006352 032777 001000 171610 DOTHIS: BIT #BIT9, @SRPTR ;IS SELECT TEST SWITCH SET
1493 006360 001002 BNE GOBACK ;RETURN TO TEST IF SW SET
1494 006362 000137 002502 JMP GTRDYX ;GO TO NEXT TEST
1495 006366 000207 GOBACK: RTS
1496
1497 006370 012737 006400 000024 PFAIL: MOV #PWRUP, %24
1498 006376 000000 HALT
1499 006400 012737 006370 000024 PWRUP: MOV #PFAIL, %24
1500 006406 000005 RESET
1501 006410 012706 001176 MOV #SPBOT, %26
1502 006414 104000 TYPE
1503 006416 020263 MPWRF
1504 006420 104003 ERROR
1505 006422 000452 BR RESTART
1506
1507 ;
1508 006424 022737 000000 001614 ;DECIDE IF VECTOR TO BE MAPPED AND MAP
1509 006432 001045 TSTVEC: CMP #0, FOUNDV ;NEED VECTOR MAPPING
1510 006434 004737 003614 BNE TSTVEX ;NO, EXIT
1511 006440 005037 001420 JSR %7, OVRLAY
1512 006444 005037 177776 CLR RXVTR
CLR PSW
```

```
1513 006450 052737 000001 001472 BIS #BIT0,FMAP ;SET MAPPING FLAG
1514 006456 042777 000100 172730 BIC #BIT6,@TXCSR ;CAUSE INTERRUPT
1515 006464 052777 000100 172722 BIS #BIT6,@TXCSR
1516 006472 000240 NOP
1517 006474 000240 NOP
1518 006476 005737 001420 TST RXVTR ;DID TRAP OCCUR?
1519 006502 001011 BNE TSTVA ;YES, OK
1520 006504 032777 020000 171456 BIT #BIT13,@SRPTR
1521 006512 001344 BNE TSTVEC
1522 006514 104000 TYPE ;NO, ERROR
1523 006516 020145 INTER
1524 006520 104003 ERROR
1525 006522 000137 006424 JMP TSTVEC
1526 006526 042777 000100 172660 TSTVA: BIC #BIT6,@TXCSR
1527 006534 012737 000340 177776 MOV #PRTY7,PSW ;RAISE PRIORITY, RETURN
1528 006542 005237 001614 INC FOUNDV
1529 006546 000207 TSTVEX: RTS %7
1530
1531 ;RESTART ROUTINE
1532 006550 013700 001450 RESTART:MOV PRGNUM,%0
1533 006554 006300 ASL %0
1534 006556 000170 006562 JMP @RSTART(0) ;GO RESTART SELECTED PROGRAM
1535
1536 006562 006640 RSTART: PRG0A ;PROGRAM 0 RESTART ADDRESS
1537 006564 014706 PRG1A ;PROGRAM 1 RESTART ADDRESS
1538 006566 014760 PRG2A ;PROGRAM 2 RESTART ADDRESS
1539 006570 015056 PRG3A ;PROGRAM 3 RESTART ADDRESS
1540 006572 015106 PRG4A ;PROGRAM 4 RESTART ADDRESS
1541 006574 005244 INCRPG
1542 006576 005244 INCRPG
1543 006600 005244 INCRPG
1544
1545
1546 ;PRG0 - INPUT-OUTPUT LOGIC TESTS
1547
1548 006602 012737 006644 001452 PRG0: MOV #ATO,KSTART
1549 006610 005737 000042 TST @#42 ;MONITOR LOAD
1550 006614 001307 BNE PRG0B ;YES, START TEST
1551 006616 104000 TYPE ;TYPE TITLE AND INSTRUCTIONS
1552 006620 016134 POTIT
1553 006622 105777 172602 CHG9: TSTB @TKS ;SEE IF ANY INPUT
1554 006626 100375 BPL CHG9 ;WAIT FOR INPUT
1555 006630 004737 005220 JSR 7,SETSR
1556 006634 004537 004244 PRG0B: JSR 5,LINSEL ;GO GET LINE # FROM USER
1557 006640 000137 002474 PRG0A: JMP GETRDY ;GET STARTED.
1558 177777 X=-1
1559 006644 TSTA 1000.,AA,CD
1560 006644 TSTAA AA,1000.,\X+1+CD,\X+2,\X+1
1561
1562 006644 100000 ATO: 100000 ;TEST NUMBER
1563 006646 006676 AT1 ;ADDRESS OF NEXT TEST
1564 006650 001750 1000. ;ITERATION COUNT
1565 006652 006654 AAA ;SCOPE ENTRY POINT
1566 000300 X=X+1
1567
1568 ;TEST ABILITY TO REFERENCE RECEIVER CSR WITHOU' TRAPPING
```

```
1569 006654 012737 006670 000004 AAA: MOV #AAE,MACHER ;SET UP MACHINE ERROR TRAP.
1570 006662 005077 172522 CLR @RXCSR ;REFERENCE RXCSR
1571 006666 104012 AAB: SCOPE ;OK IF NO TRAP. SCOPE
1572 006670 022626 AAE: POPSP2
1573 006672 104003 ERROR ;TRAPPED WHEN REFERENCING RXCSR.
1574 006674 000774 BR AAB
1575 006676 TSTA 1000.,AB,CD
1576 006676 TSTAA AB,1000.,\X+1+CD,\X+2,\X+1
1577 ;*****
1578 006676 100001 AT1: 100001 ;TEST NUMBER
1579 006700 006736 AT2 ;ADDRESS OF NEXT TEST
1580 006702 001750 1000. ;ITERATION COUNT
1581 006704 C06706 ABA ;SCOPE ENTRY POINT
1582 000001 X=X+1
1583 ;*****
1584 ;TEST ABILITY TO REFERENCE RECEIVER BUFFER WITHOUT TRAPPING
1585 006706 012737 006730 000004 ABA: MOV #ABE,MACHER ;SET UP MACHINE ERROR TRAP.
1586 006714 005737 002110 TST @#XORFLG
1587 006720 100402 BMI ABB
1588 006722 005777 172464 TST @RXBUF ;REFERENCE RXBUF
1589 006726 104012 ABB: SCOPE ;OK IF NO TRAP SCOPE
1590 006730 022626 ABE: POPSP2
1591 006732 104003 ERROR ;TRAPPED WHEN REFERENCING RXBUF
1592 006734 000774 BR ABB
1593 006736 TSTA 1000.,AC,CD
1594 006736 TSTAA AC,1000.,\X+1+CD,\X+2,\X+1
1595 ;*****
1596 006736 100002 AT2: 100002 ;TEST NUMBER
1597 006740 006770 AT3 ;ADDRESS OF NEXT TEST
1598 006742 001750 1000. ;ITERATION COUNT
1599 006744 006746 ACA ;SCOPE ENTRY POINT
1600 000002 X=X+1
1601 ;*****
1602 ;TEST ABILITY TO REFERENCE TRANSMITTER CSR WITHOUT TRAPPING.
1603 006746 012737 006762 000004 ACA: MOV #ACE,MACHER ;SET UP MACHINE ERROR TRAP.
1604 006754 005777 172434 TST @TXCSR ;REFERENCE TXCSR
1605 006760 104012 ACB: SCOPE ;SCOPE
1606 006762 022626 ACE: POPSP2
1607 006764 104003 ERROR ;TRAPPED WHEN REFERENCING TXCSR
1608 006766 000774 BR ACB
1609 006770 TSTA 1000.,AD,CD
1610 006770 TSTAA AD,1000.,\X+1+CD,\X+2,\X+1
1611 ;*****
1612 006770 100003 AT3: 100003 ;TEST NUMBER
1613 006772 007022 AT4 ;ADDRESS OF NEXT TEST
1614 006774 001750 1000. ;ITERATION COUNT
1615 006776 007000 ADA ;SCOPE ENTRY POINT
1616 000003 X=X+1
1617 ;*****
1618 ;TEST ABILITY TO REFERENCE TRANSMITTER BUFFER WITHOUT TRAPPING
1619 007000 012737 007014 000004 ADA: MOV #ADE,MACHER ;SET UP MACHINE ERROR TRAP.
1620 007006 005777 172404 TST @TXBUF ;REFERENCE TX BUF.
1621 007012 104012 ADB: SCOPE ;SCOPE
1622 007014 022626 ADE: POPSP2
1623 007016 104003 ERROR ;TRAPPED WHEN REFERENCING TXBUF
1624 007020 000774 BR ADB
```

```
1625 007022
1626 007022
1627
1628 007022 100004
1629 007024 007122
1630 007026 000012
1631 007030 007032
1632 000004
1633
1634
1635
1636 007032 032777 000001 172354
1637 007040 001402
1638 007042 104003
1639 007044 000421
1640 007046 052777 000001 172340
1641 007054 032777 000001 172332
1642 007062 001002
1643 007064 104003
1644 007066 000410
1645 007070 042777 000001 172316
1646 007076 032777 000001 172310
1647 007104 001401
1648 007106 104003
1649 007110 052777 000001 172276
1650 007116 104011
1651 007120 104012
1652 007122
1653 007122
1654
1655 007122 100005
1656 007124 007222
1657 007126 000012
1658 007130 007132
1659 000005
1660
1661
1662 007132 032777 000004 172254
1663 007140 001402
1664 007142 104003
1665 007144 000421
1666 007146 052777 000004 172240
1667 007154 032777 000004 172232
1668 007162 001002
1669 007164 104003
1670 007166 000410
1671 007170 042777 000004 172216
1672 007176 032777 000004 172210
1673 007204 001401
1674 007206 104003
1675 007210 052777 000004 172176
1676 007216 104011
1677 007220 104012
1678 007222
1679 007222
1680
```

```

TSTA 10.,AE,CD
TSTAA AE,10.,\X+1+CD,\X+2,\X+1
:*****
AT4: 100004 ;TEST NUMBER
AT5 ;ADDRESS OF NEXT TEST
10. ;ITERATION COUNT
AEA ;SCOPE ENTRY POINT
X=X+1
:*****
:TEST THAT TXCSR BIT 0 (BREAK) CAN BE SET AND CLEARED
:AND THAT RESET CLEARS IT
AEA: BIT #BIT0,@TXCSR ;SEE IF BIT IS CLEAR
BEQ AEB ;BR IF CLEAR
ERROR ;RESET DID NOT CLEAR IT
BR AED
AEB: BIS #BIT0,@TXCSR ;SET TXCSR BIT 0
BIT #BIT0,@TXCSR ;DID IT SET
BNE AEC ;YES, GO ON
ERROR ;TXCSR BIT0 FAILED TO SET
BR AED
AEC: BIC #BIT0,@TXCSR ;CLEAR TXCSR BIT 0
BIT #BIT0,@TXCSR ;DID IT CLEAR
BEQ AED
ERROR ;TXCSR BIT 0 DID NOT CLEAR
AED: BIS #BIT0,@TXCSR ;ISSUE RESET TO CLEAR
SRESET
SCOPE
TSTA 10.,AG,CD
TSTAA AG,10.,\X+1+CD,\X+2,\X+1
:*****
AT5: 100005 ;TEST NUMBER
AT6 ;ADDRESS OF NEXT TEST
10. ;ITERATION COUNT
AGA ;SCOPE ENTRY POINT
X=X+1
:*****
:TEST THAT TXCSR BIT2 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
AGA: BIT #BIT2,@TXCSR ;SEE IF TXCSR BIT2 IS CLEAR.
BEQ AGB ;BRANCH IF BIT IS CLEAR.
ERROR ;RESET DID NOT CLEAR TXCSR BIT2
BR AGD
AGB: BIS #BIT2,@TXCSR ;SET TXCSR BIT2.
BIT #BIT2,@TXCSR ;SEE IF BIT IS SET.
BNE AGC ;BRANCH IF BIT IS SET.
ERROR ;TXCSR BIT2 FAILED TO SET.
BR AGD
AGC: BIC #BIT2,@TXCSR ;CLEAR TXCSR BIT2
BIT #BIT2,@TXCSR ;SEE IF BIT IS CLEAR.
BEQ AGD
ERROR ;TXCSR BIT2 FAILED TO CLEAR.
AGD: BIS #BIT2,@TXCSR ;SET TXCSR BIT2.
SRESET ;ISSUE RESET TO CLEAR BIT.
SCOPE ;SCOPE
TSTA 10.,AJ,CD
TSTAA AJ,10.,\X+1+CD,\X+2,\X+1
:*****
```

```
1681 007222 100006 AT6: 100006 ;TEST NUMBER
1682 007224 007330 AT7 ;ADDRESS OF NEXT TEST
1683 007226 000012 10. ;ITERATION COUNT
1684 007230 007232 AJA ;SCOPE ENTRY POINT
1685 000006 X=X+1
1686
1687 ;*****
1688 007232 012737 000340 177776 ;TEST THAT TXCSR BIT6 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
1689 007240 032777 000100 172146 AJA: MOV #PRTY7,PSW ;SET PRIORITY 7.
1690 007246 001402 BIT #BIT6,@TXCSR ;SEE IF TXCSR BIT6 IS CLEAR.
1691 007250 104003 BEQ A,B ;BRANCH IF BIT IS CLEAR.
1692 007252 000421 ERROR ;RESET DID NOT CLEAR TXCSR BIT6
1693 007254 052777 000100 172132 AJB: BR AJD
1694 007262 032777 000100 172124 AJB: BIS #BIT6,@TXCSR ;SET TXCSR BIT6.
1695 007270 001002 BIT #BIT6,@TXCSR ;SEE IF BIT IS SET.
1696 007272 104003 BNE AJC ;BRANCH IF BIT IS SET.
1697 007274 000410 ERROR ;TXCSR BIT6 FAILED TO SET.
1698 007276 042777 000100 172110 AJC: BR AJD
1699 007304 032777 000100 172102 AJC: BIC #BIT6,@TXCSR ;CLEAR TXCSR BIT6
1700 007312 001401 BIT #BIT6,@TXCSR ;SEE IF BIT IS CLEAR.
1701 007314 104003 BEQ AJD
1702 007316 052777 000100 172070 AJD: ERROR ;TXCSR BIT6 FAILED TO CLEAR.
1703 007324 104011 BIS #BIT6,@TXCSR ;SET TXCSR BIT6.
1704 007326 104012 SRESET ;ISSUE RESET TO CLEAR BIT.
1705 007330 SCOPE ;SCOPE
1706 007330 TSTA 100.,AK,CD
1707 TSTAA AK,100.,\X+1+CD,\X+2,\X+1
1708 007330 100007 AT7: 100007 ;TEST NUMBER
1709 007332 007354 AT10 ;ADDRESS OF NEXT TEST
1710 007334 000144 100. ;ITERATION COUNT
1711 007336 007340 AKA ;SCOPE ENTRY POINT
1712 000007 X=X+1
1713
1714 ;*****
1715 ;TEST THAT TXCSR BIT 7 (READY BIT) IS SET UPON ENTERING ROUTINE AND
1716 007340 105777 172050 ;THAT IT CAN BE READ RELIABLY.
1717 007344 100402 AKA: TSTB @TXCSR ;SEE IF TXCSR BIT 7 IS SET.
1718 007346 104003 BMI AKB ;BRANCH IF SET.
1719 007350 104011 ERROR ;TXCSR BIT 7 NOT SET.
1720 007352 104012 SRESET ;ISSUE RESET TO CLEAR BIT IF ERROR
1721 007354 SCOPE ;SCOPE
1722 007354 TSTA 100.,AL,0
1723 TSTAA AL,100.,\X+1+0,\X+2,\X+1
1724 007354 000010 AT10: 10 ;TEST NUMBER
1725 007356 007436 AT11 ;ADDRESS OF NEXT TEST
1726 007360 000144 100. ;ITERATION COUNT
1727 007362 007364 ALA ;SCOPE ENTRY POINT
1728 000010 X=X+1
1729
1730 ;*****
1731 007364 042777 000002 172016 ;TEST THAT RXCSR BIT 1 CAN BE SET + CLEARED
1732 007372 052777 000002 172010 ALA: BIC #BIT1,@RXCSR
1733 007400 032777 000002 172002 BIS #BIT1,@RXCSR ;SET RXCSR BIT1
1734 007406 001002 BIT #BIT1,@RXCSR ;SEE IF BIT IS SET
1735 007410 104003 BNE ALY ;BRANCH IF SET
1736 007412 000410 ERROR ;RXCSR BIT 1 FAILED TO SET
1736 BR ALZ
```



```
1737 007414 042777 000002 171766 ALY: BIC #BIT1,@RXCSR ;CLEAR RXCSR BIT 1
1738 007422 032777 000002 171760 BIT #BIT1,@RXCSR ;SEE IF BIT IS CLEAR
1739 007430 001401 BEQ ALZ
1740 007432 104003 ERROR ;RXCSR BIT 1 FAILED TO CLEAR
1741 007434 104012 ALZ: SCOPE ;SCOPE
1742 007436 TSTA 10.,AP,0
1743 007436 TSTAA AP,10.,\X+1+0,\X+2,\X+1
1744
;*****
1745 007436 000011 AT11: 11 ;TEST NUMBER
1746 007440 007536 AT12 ;ADDRESS OF NEXT TEST
1747 007442 000012 10. ;ITERATION COUNT
1748 007444 007446 APA ;SCOPE ENTRY POINT
1749 000011 X=X+1
1750
;*****
1751 ;TEST THAT RXCSR BIT2 IS CLEAR AND CAN BE READ RELIABLY.
1752 007446 032777 000004 171734 APA: BIT #BIT2,@RXCSR ;SEE IF RXCSR BIT2 IS CLEAR.
1753 007454 001402 BEQ APB ;BRANCH IF BIT IS CLEAR.
1754 007456 104003 ERROR ;RXCSR BIT2 IS NOT CLEAR.
1755 007460 000421 BR APD
1756 007462 052777 000004 171720 APB: BIS #BIT2,@RXCSR ;SET RXCSR BIT2
1757 007470 032777 000004 171712 BIT #BIT2,@RXCSR ;SEE IF BIT IS SET
1758 007476 001002 BNE APCX ;BRANCH IF SET
1759 007500 104003 ERROR ;RXCSR BIT2 FAILED TO SET
1760 007502 000410 BR APD
1761 007504 042777 000004 171676 APCX: BIC #BIT2,@RXCSR ;CLEAR RXCSR BIT2
1762 007512 032777 000004 171670 BIT #BIT2,@RXCSR ;SEE IF BIT IS CLEAR
1763 007520 001401 BEQ APD
1764 007522 104003 ERROR ;RXCSR BIT2 FAILED TO CLEAR
1765 007524 052777 000004 171656 APD: BIS #BIT2,@RXCSR ;SET BIT
1766 007532 104011 SRESET ;ISSUE RESET TO CLEAR BIT
1767 007534 104012 SCOPE
1768 007536 TSTA 10.,AQ,0
1769 007536 TSTAA AQ,10.,\X+1+0,\X+2,\X+1
1770
;*****
1771 007536 000012 AT12: 12 ;TEST NUMBER
1772 007540 007636 AT13 ;ADDRESS OF NEXT TEST
1773 007542 000012 10. ;ITERATION COUNT
1774 007544 007546 AQA ;SCOPE ENTRY POINT
1775 000012 X=X+1
1776
;*****
1777 ;TEST THAT RXCSR BIT3 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
1778 007546 032777 000010 171634 AQA: BIT #BIT3,@RXCSR ;SEE IF RXCSR BIT3 IS CLEAR.
1779 007554 001402 BEQ AQB ;BRANCH IF BIT IS CLEAR.
1780 007556 104003 ERROR ;RESET DID NOT CLEAR RXCSR BIT3
1781 007560 000421 BR AQD
1782 007562 052777 000010 171620 AQB: BIS #BIT3,@RXCSR ;SET RXCSR BIT3.
1783 007570 032777 000010 171612 BIT #BIT3,@RXCSR ;SEE IF BIT IS SET.
1784 007576 001002 BNE AQC ;BRANCH IF BIT IS SET.
1785 007600 104003 ERROR ;RXCSR BIT3 FAILED TO SET.
1786 007602 000410 BR AQD
1787 007604 042777 000010 171576 AQC: BIC #BIT3,@RXCSR ;CLEAR RXCSR BIT3
1788 007612 032777 000010 171570 BIT #BIT3,@RXCSR ;SEE IF BIT IS CLEAR.
1789 007620 001401 BEQ AQD
1790 007622 104003 ERROR ;RXCSR BIT3 FAILED TO CLEAR.
1791 007624 052777 000010 171556 AQD: BIS #BIT3,@RXCSR ;SET RXCSR BIT3.
1792 007632 104011 SRESET ;ISSUE RESET TO CLEAR BIT.
```

```
1793 007634 104012          SCOPE          :SCOPE
1794 007636                TSTA          10.,AR,0
1795 007636                TSTAA         AR,10.,\X+1+0,\X+2,\X+1
1796
1797 007636 000013          :*****
1798 007640 007744          AT13: 13          :TEST NUMBER
1799 007642 000012          :*****
1800 007644 007646          AT14          :ADDRESS OF NEXT TEST
1801 000013                10.          :ITERATION COUNT
1802                ARA          :SCOPE ENTRY POINT
1803                X=X+1
1804 007646 012737 000340 177776 :*****
1805 007654 032777 000040 171526 :TEST THAT RXCSR BITS CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
1806 007662 001402          ARA:  MOV        #PRTY7,PSW  ;PRTY7 TO INHIBIT ANY INT
1807 007664 104003          BIT        #BIT5,@RXCSR  ;SEE IF RXCSR BITS IS CLEAR.
1808 007666 000421          BEQ        ARB          ;BRANCH IF BIT IS CLEAR.
1809 007670 052777 000040 171512 :*****
1810 007676 032777 000040 171504 ARB:  BR        ARD          ;RESET DID NOT CLEAR RXCSR BITS
1811 007704 001002          BR        ARD
1812 007706 104003          BIS        #BIT5,@RXCSR  ;SET RXCSR BITS.
1813 007710 000410          BIT        #BIT5,@RXCSR  ;SEE IF BIT IS SET.
1814 007712 042777 000040 171470 :*****
1815 007720 032777 000040 171462 ARC:  BNE        ARC          ;BRANCH IF BIT IS SET.
1816 007726 001401          ERROR     ;RXCSR BITS FAILED TO SET.
1817 007730 104003          BR        ARD
1818 007732 052777 000040 171450 :*****
1819 007740 104011          ARC:  BIC        #BIT5,@RXCSR ;CLEAR RXCSR BITS
1820 007742 104012          BIT        #BIT5,@RXCSR  ;SEE IF BIT IS CLEAR.
1821 007744                BEQ        ARD
1822 007744                ERROR     ;RXCSR BIT4 FAILED TO CLEAR.
1823                ARD
1824 007744 100014          :*****
1825 007746 010052          AT14: 100014      :TEST NUMBER
1826 007750 000012          AT15          :ADDRESS OF NEXT TEST
1827 007752 007754          10.          :ITERATION COUNT
1828                ASA          :SCOPE ENTRY POINT
1829                X=X+1
1830                :*****
1831 007754 012737 000340 177776 :TEST THAT RXCSR BIT6 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
1832 007762 032777 000100 171420 :*****
1833 007770 001402          ASA:  MOV        #PRTY7,PSW  ;SET PRIORITY 7.
1834 007772 104003          BIT        #BIT6,@RXCSR  ;SEE IF RXCSR BIT6 IS CLEAR.
1835 007774 000421          BEQ        ASE          ;BRANCH IF BIT IS CLEAR.
1836 007776 052777 000100 171404 :*****
1837 010004 032777 000100 171376 ASB:  BR        ASD          ;RESET DID NOT CLEAR RXCSR BIT6
1838 010012 001002          BR        ASD
1839 010014 104003          BIS        #BIT6,@RXCSR  ;SET RXCSR BIT6.
1840 010016 000410          BIT        #BIT6,@RXCSR  ;SEE IF BIT IS SET.
1841 010020 042777 000100 171362 :*****
1842 010026 032777 000100 171354 ASC:  BNE        ASC          ;BRANCH IF BIT IS SET.
1843 010034 001401          ERROR     ;RXCSR BIT6 FAILED TO SET.
1844 010036 104003          BR        ASD
1845 010040 052777 000100 171342 :*****
1846 010046 104011          ASC:  BIC        #BIT6,@RXCSR ;CLEAR RXCSR BIT6
1847 010050 104012          BIT        #BIT6,@RXCSR  ;SEE IF BIT IS CLEAR.
1848 010052                BEQ        ASD
1848                ERROR     ;RXCSR BIT6 FAILED TO CLEAR.
1848                ASD
1848                SRESET   ;SET RXCSR BIT6.
1848                SCOPE    ;ISSUE RESET TO CLEAR BIT.
1848                TSTA     100.,AT,0 ;SCOPE
```

```
1849 010052
1850
1851 010052 000015
1852 010054 010100
1853 010056 000144
1854 010060 010062
1855 000015
1856
1857
1858 010062 032777 000200 171320
1859 010070 001402
1860 010072 104003
1861 010074 104011
1862 010076 104012
1863 010100
1864 010100
1865
1866 010100 000016
1867 010102 010126
1868 010104 000144
1869 010106 010110
1870 000016
1871
1872
1873 010110 032777 002000 171272
1874 010116 001402
1875 010120 104003
1876 010122 104011
1877 010124 104012
1878 010126
1879 010126
1880
1881 010126 100017
1882 010130 010154
1883 010132 000144
1884 010134 010136
1885 000017
1886
1887
1888 010136 032777 004000 171244
1889 010144 001402
1890 010146 104003
1891 010150 104011
1892 010152 104012
1893 010154
1894 010154
1895
1896 010154 100020
1897 010156 010202
1898 010160 000144
1899 010162 010164
1900 000020
1901
1902
1903 010164 032777 040000 171216
1904 010172 001402
```

TSTAA AT,100.,\X+1+0,\X+2,\X+1
:.....
AT15: 15 ;TEST NUMBER
AT16 ;ADDRESS OF NEXT TEST
100. ;ITERATION COUNT
ATA ;SCOPE ENTRY POINT
X=X+1 ;
:.....
;TEST THAT RXCSR BIT7 IS CLEAR AND CAN BE READ RELIABLY.
ATA: BIT #BIT7,@RXCSR ;SEE IF RXCSR BIT7 IS CLEAR.
BEQ ATB ;BRANCH IF BIT IS CLEAR.
ERROR ;RXCSR BIT7 IS NOT CLEAR.
SRESET ;RESET IF ERROR
ATB: SCOPE ;SCOPE
TSTA 100.,AX,0
TSTAA AX,100.,\X+1+0,\X+2,\X+1
:.....
AT16: 16 ;TEST NUMBER
AT17 ;ADDRESS OF NEXT TEST
100. ;ITERATION COUNT
AXA ;SCOPE ENTRY POINT
X=X+1 ;
:.....
;TEST THAT RXCSR BIT10 IS CLEAR AND CAN BE READ RELIABLY.
AXA: BIT #BIT10,@RXCSR ;SEE IF RXCSR BIT10 IS CLEAR.
BEQ AXB ;BRANCH IF BIT IS CLEAR.
ERROR ;RXCSR BIT10 IS NOT CLEAR.
SRESET ;RESET BIT IF ERROR
AXB: SCOPE ;SCOPE
TSTA 100.,AY,CD
TSTAA AY,100.,\X+1+CD,\X+2,\X+1
:.....
AT17: 100017 ;TEST NUMBER
AT20 ;ADDRESS OF NEXT TEST
100. ;ITERATION COUNT
AYA ;SCOPE ENTRY POINT
X=X+1 ;
:.....
;TEST THAT RXCSR BIT11 IS CLEAR AND CAN BE READ RELIABLY.
AYA: BIT #BIT11,@RXCSR ;SEE IF RXCSR BIT11 IS CLEAR.
BEQ AYB ;BRANCH IF BIT IS CLEAR.
ERROR ;RXCSR BIT11 IS NOT CLEAR.
SRESET ;RESET BIT IF ERROR
AYB: SCOPE ;SCOPE
TSTA 100.,AZ,CD
TSTAA AZ,100.,\X+1+CD,\X+2,\X+1
:.....
AT20: 100020 ;TEST NUMBER
AT21 ;ADDRESS OF NEXT TEST
100. ;ITERATION COUNT
AZA ;SCOPE ENTRY POINT
X=X+1 ;
:.....
;TEST THAT RXCSR BIT14 IS CLEAR AND CAN BE READ RELIABLY.
AZA: BIT #BIT14,@RXCSR ;SEE IF RXCSR BIT14 IS CLEAR.
BEQ AZB ;BRANCH IF BIT IS CLEAR.

```
1905 010174 104003          ERROR          ;RXCSR BIT14 IS NOT CLEAR.
1906 010176 104011          SRESET         ;RESET BIT IF ERROR
1907 010200 104012          AZB: SCOPE     ;SCOPE
1908 010202          TSTA 100.,AAA,CD
1909 010202          TSTAA AAA,100.,\X+1+CD,\X+2,\X+1
1910
1911 010202 100021          AT21: 100021   ;TEST NUMBER
1912 010204 010230          AT22         ;ADDRESS OF NEXT TEST
1913 010206 000144          100.         ;ITERATION COUNT
1914 010210 010212          AAAA        ;SCOPE ENTRY POINT
1915          X=X+1
1916
1917          ;*****
1918 010212 032777 100000 171170 AAAA: BIT #BIT15,@RXCSR ;TEST THAT RXCSR BIT15 IS CLEAR AND CAN BE READ RELIABLY.
1919 010220 001402          BEQ AAAB    ;SEE IF RXCSR BIT15 IS CLEAR.
1920 010222 104003          ERROR      ;BRANCH IF BIT IS CLEAR.
1921 010224 104011          SRESET     ;RXCSR BIT15 IS NOT CLEAR.
1922 010226 104012          AAAB: SCOPE ;RESET BIT IF ERROR
1923
1924          ;SCOPE
1925          ;ALL PREVIOUS TESTS MUST HAVE BEEN RUN SUCCESSFULLY PRIOR
1926          ;TO RUNNING THE FOLLOWING TESTS. ALSO, THE JUMPER CONNECTOR
1927          ;MUST BE INSERTED IN THE DL11-E CABLE IN PLACE OF THE MODEM. COMMENTS
1928          ;REFER TO OPERATION WITH JUMPER INSERTED.
1929
1929 010230          TSTA 100.,AFB,0
1930 010230          TSTAA AFB,100.,\X+1+0,\X+2,\X+1
1931
1932 010230 000022          AT22: 22      ;TEST NUMBER
1933 010232 010314          AT23         ;ADDRESS OF NEXT TEST
1934 010234 000144          100.         ;ITERATION COUNT
1935 010236 010240          AFBA        ;SCOPE ENTRY POINT
1936          X=X+1
1937
1938          ;*****
1939          ;TEST THAT CARRIER DETECT SETS AND CLEARS WHEN DATA TERMINAL
1940          ;READY SETS AND CLEARS.
1941 010240 052777 000002 171142 AFBA: BIS #BIT1,@RXCSR ;SET DATA TERMINAL READY
1942 010246 004737 012122          JSR %7,TIME ;DELAY
1943 010252 032777 010000 171130 BIT #BIT12,@RXCSR ;TEST CARRIER DETECT
1944 010260 001002          BNE AFBB    ;SHOULD BE SET
1945 010262 104003          ERROR      ;WASN'T
1946 010264 000412          BR AFBC
1947 010266 042777 000002 171114 AFBB: BIC #BIT1,@RXCSR ;CLEAR DATA TERMINAL READY
1948 010274 004737 012122          JSR %7,TIME ;DELAY
1949 010300 032777 010000 171102 BIT #BIT12,@RXCSR ;TEST CARRIER DETECT
1950 010306 001401          BEQ AFBC    ;WAS SET, ERROR /
1951 010310 104003          ERRGR
1952 010312 104012          AFBC: SCOPE
1953          TSTA 100.,AGB,0
1954          TSTAA AGB,100.,\X+1+0,\X+2,\X+1
1955
1955 010314 000023          AT23: 23      ;TEST NUMBER
1956 010316 010466          AT24         ;ADDRESS OF NEXT TEST
1957 010320 000144          100.         ;ITERATION COUNT
1958 010322 010324          AGBA        ;SCOPE ENTRY POINT
1959          X=X+1
1960          ;*****
```

```
1961                                     :TEST THAT MODEM INTERRUPT (BIT 15) SETS WHEN CARRIER DETECT
1962                                     :CHANGES STATE, AND IS CLEARED WHEN RXCSR IS READ.
1963 010324 042777 000002 171056 AGBA: BIC #BIT1,@RXCSR ;CLEAR DATA TERMINAL READY
1964 010332 004737 012122          JSR %7,TIME ;DELAY
1965 010336 017737 171046 001610 MOV @RXCSR,RXCSRT ;READ RXCSR
1966 010344 032777 100000 171036 BIT #BIT15,@RXCSR ;TEST MODEM INTERRUPT
1967 010352 001402          BEQ AGBB ;WAS CLEAR GO TO AGBB
1968 010354 104003          ERROR ;WASN'T CLEAR
1969 010356 000442          BR AGBE ;GO TO SCOPE
1970 010360 052777 000002 171022 AGBB: BIS #BIT1,@RXCSR ;SETTING DATA TERMINAL READY
1971                                     ;CAUSES CARRIER DETECT TO SET
1972                                     ;WHICH CAUSES MODEM INTERRUPT TO SET
1973 010366 004737 012122          JSR %7,TIME ;DELAY
1974 010372 017737 171012 001610 MOV @RXCSR,RXCSRT ;MOVE RXCSR TO TEMPORARY LOCATION
1975 010400 032737 100000 001610 BIT #BIT15,RXCSRT ;TEST MODEM INTERRUPT
1976 010406 001002          BNE AGBC ;SHOULD BE SET GO TO AGBC
1977 010410 104003          ERROR ;WAS CLEAR
1978 010412 000424          BR AGBE ;GO TO SCOPE
1979 010414 032777 100000 170766 AGBC: BIT #BIT15,@RXCSR ;MODEM INTERRUPT BIT SHOULD
1980                                     ;HAVE BEEN CLEARED
1981 010422 001402          BEQ AGBD ;IT WAS GO TO AGBD
1982 010424 104003          ERROR ;IT WASN'T
1983 010426 000416          BR AGBE ;GO TO SCOPE
1984 010430 042777 000002 170752 AGBD: BIC #BIT1,@RXCSR ;CLEARING DATA TERMINAL READY
1985                                     ;CAUSES CARRIER DETECT TO CLEAR
1986                                     ;BUT MODEM INTERRUPT WILL SET
1987 010436 004737 012122          JSR %7,TIME ;DELAY
1988 010442 017737 170742 001610 MOV @RXCSR,RXCSRT ;MOVE RXCSR TO TEMPORARY LOCATION
1989 010450 032737 100000 001610 BIT #BIT15,RXCSRT ;TEST MODEM INTERRUPT
1990 010456 001002          BNE AGBE ;SHOULD BE SET
1991 010460 104003          ERROR ;IT WASN'T
1992 010462 000400          BR AGBE
1993 010464 104012          AGBE: SCOPE ;SCOPE
1994 010466          TSTA 100.,AJB,0
1995 010466          TSTAA AJB,100.,\X+1+0,\X+2,\X+1
1996                                     ;*****
1997 010466 000024          AT24: 24 ;TEST NUMBER
1998 010470 010600          AT25 ;ADDRESS OF NEXT TEST
1999 010472 000144          100. ;ITERATION COUNT
2000 010474 010476          AJBA ;SCOPE ENTRY POINT
2001 000024          X=X+1
2002                                     ;*****
2003                                     :TEST THAT CLEAR TO SEND (BIT13) SETS/CLEARs WHEN DATA TERMINAL
2004                                     :READY SETS/CLEARs.
2005 010476 042777 000002 170704 AJBA: BIC #BIT1,@RXCSR ;CLEAR DATA TERMINAL READY
2006 010504 004737 012122          JSR %7,TIME ;DELAY
2007 010510 032777 020000 170672 BIT #BIT13,@RXCSR ;TEST CLEAR TO SEND
2008 010516 001402          BEQ AJBB
2009 010520 104003          ERROR ;CLEAR TO SEND SHOULD BE CLEAR
2010 010522 000425          BR AJBD
2011 010524 052777 000002 170656 AJBB: BIS #BIT1,@RXCSR ;SET DATA TERMINAL READY
2012 010532 004737 012122          JSR %7,TIME ;DELAY
2013 010536 032777 020000 170644 BIT #BIT13,@RXCSR ;TEST CLEAR TO SEND
2014 010544 001002          BNE AJBC ;BRANCH IF SET
2015 010546 104003          ERROR ;CLEAR TO SEND SHOULD BE SET
2016 010550 000412          BR AJBD
```

```
2017 010552 042777 000002 170630 AJBC: BIC #BIT1,@RXCSR ;CLEAR DATA TERMINAL READY
2018 010560 004737 012122 JSR %7,TIME ;DELAY
2019 010564 032777 020000 170616 BIT #BIT13,@RXCSR ;TEST CLEAR TO SEND
2020 010572 001401 BEQ AJBD
2021 010574 104003 ERROR ;CLEAR TO SEND SHOULD BE CLEAR
2022 010576 104012 AJBD: SCOPE ;SCOPE
2023 010600 TSTA 100.,AKB,0
2024 010600 TSTAA AKB,100.,\X+1+0,\X+2,\X+1
2025 :*****
2026 010600 000025 AT25: 25 ;TEST NUMBER
2027 010602 010674 AT26 ;ADDRESS OF NEXT TEST
2028 010604 000144 100. ;ITERATION COUNT
2029 010606 010610 AKBA ;SCOPE ENTRY POINT
2030 000025 X=X+1
2031 :*****
2032 :TEST THAT RING (BIT 14 RXCSR) SETS WHEN REQUEST TO
2033 :SEND SETS AND CLEARS AND RESET CLEARS RING
2034 010610 042777 000004 170572 AKBA: BIC #BIT2,@RXCSR ;CLEAR REQUEST TO SEND
2035 010616 004737 012122 JSR %7,TIME ;DELAY
2036 010622 052777 000004 170560 BIS #BIT2,@RXCSR ;SET REQUEST TO SEND
2037 010630 004737 012122 JSR %7,TIME ;DELAY
2038 010634 032777 040000 170546 BIT #BIT14,@RXCSR ;TEST RING
2039 010642 001001 BNE AKBC
2040 010644 104003 ERROR ;RING SHOULD BE SET
2041 010646 042777 000004 170534 AKBC: BIC #BIT2,@RXCSR ;CLEAR REQUEST TO SEND
2042 010654 004737 012122 JSR %7,TIME ;DELAY
2043 010660 032777 040000 170522 BIT #BIT14,@RXCSR ;TEST RING
2044 010666 001401 BEQ .+4 ;SHOULD BE CLEAR
2045 010670 104003 ERROR
2046 010672 104012 SCOPE ;SCOPE
2047 010674 TSTA 100.,AOB,0
2048 010674 TSTAA AOB,100.,\X+1+0,\X+2,\X+1
2049 :*****
2050 010674 000026 AT26: 26 ;TEST NUMBER
2051 010676 011006 AT27 ;ADDRESS OF NEXT TEST
2052 010700 000144 100. ;ITERATION COUNT
2053 010702 010704 AOBA ;SCOPE ENTRY POINT
2054 000026 X=X+1
2055 :*****
2056 :TEST THAT MODEM INTERRUPT (BIT 15 RXCSR) SETS WHEN RING SETS.
2057 010704 042777 000004 170476 AUBA: BIC #BIT2,@RXCSR ;CLEAR REQUEST TO SEND
2058 010712 004737 012122 JSR %7,TIME ;DELAY
2059 010716 032777 100000 170464 BIT #BIT15,@RXCSR ;TEST MODEM INTERRUPT BIT
2060 010724 001402 BEQ AOBBD
2061 010726 104003 ERROR
2062 010730 000425 BR AOBBD
2063 010732 052777 000004 170450 AOBBD: BIS #BIT2,@RXCSR ;SET REQUEST TO SEND
2064 010740 004737 012122 JSR %7,TIME ;DELAY
2065 010744 032777 100000 170436 BIT #BIT15,@RXCSR ;TEST MODEM INTERRUPT BIT
2066 010752 001002 BNE AOBBC
2067 010754 104003 ERROR
2068 010756 000412 BR AOBBD
2069 010760 042777 000004 170422 AOBBC: BIC #BIT2,@RXCSR ;CLEAR REQUEST TO SEND
2070 010766 004737 012122 JSR %7,TIME ;DELAY
2071 010772 032777 100000 170410 BIT #BIT15,@RXCSR ;TEST MODEM INTERRUPT BIT
2072 011000 001401 BEQ AOBBD
```



```

2073 011002 104003
2074 011004 104012
2075 011006
2076 011006
2077
2078 011006 000027
2079 011010 011120
2080 011012 000144
2081 011014 011016
2082 000027
2083
2084
2085
2086 011016 042777 000010 170364
2087 011024 004737 012122
2088 011030 032777 002000 170352
2089 011036 001402
2090 011040 104003
2091 011042 000425
2092 011044 052777 000010 170336
2093 011052 004737 012122
2094 011056 032777 002000 170324
2095 011064 001002
2096 011066 104003
2097 011070 000412
2098 011072 042777 000010 170310
2099 011100 004737 012122
2100 011104 032777 002000 170276
2101 011112 001401
2102 011114 104003
2103 011116 104012
2104 011120
2105 011120
2106
2107 011120 000030
2108 011122 011260
2109 011124 000144
2110 011126 011130
2111 000030
2112
2113
2114 011130 042777 000010 170252
2115 011136 004737 012122
2116 011142 052777 000010 170240
2117 011150 004737 012122
2118 011154 032777 100000 170226
2119 011162 001002
2120 011164 104003
2121 011166 000433
2122 011170 032777 100000 170212
2123 011176 001402
2124 011200 104003
2125 011202 000425
2126 011204 042777 000010 170176
2127 011212 004737 012122
2128 011216 032777 100000 170164

AOBD: ERROR SCOPE ;SCOPE
SCOPE
TSTA 100.,ALB,0
TSTAA ALB,100.,\X+1+0,\X+2,\X+1
:*****
AT27: 27 ;TEST NUMBER
AT30 ;ADDRESS OF NEXT TEST
100. ;ITERATION COUNT
ALBA ;SCOPE ENTRY POINT
X=X+1
:*****
:TEST THAT SUPERVISORY RECEIVE DATA (BIT 10 RXCSR) SETS/CLEAR
:WHEN SUPERVISORY XMIT DATA SETS/CLEAR.
ALBA: BIC #BIT3,@RXCSR ;CLEAR SUPERVISOR XMIT DATA
JSR %7,TIME ;DELAY
BIT #BIT10,@RXCSR ;TEST SUPERVISORY RECEIVE DATA.
BEQ ALBB
ERROR ;SHOULD HAVE BEEN CLEAR
BR ALBD
ALBB: BIS #BIT3,@RXCSR ;SET SUPERVISORY XMIT DATA
JSR %7,TIME ;DELAY
BIT #BIT10,@RXCSR ;TEST SUPERVISORY RECEIVE DATA
BNE ALBC
ERROR ;SHOULD HAVE BEEN SET
BR ALBD
ALBC: BIC #BIT3,@RXCSR ;CLEAR SUPERVISORY XMIT DATA
JSR %7,TIME ;DELAY
BIT #BIT10,@RXCSR ;TEST SUPERVISORY RECEIVE DATA
BEQ ALBD
ERROR ;SHOULD HAVE BEEN CLEAR
ALBD: SCOPE ;SCOPE
TSTA 100.,AMB,0
TSTAA AMB,100.,\X+1+0,\X+2,\X+1
:*****
AT30: 30 ;TEST NUMBER
AT31 ;ADDRESS OF NEXT TEST
100. ;ITERATION COUNT
AMBA ;SCOPE ENTRY POINT
X=X+1
:*****
:TEST THAT SUP REC DATA TRANSISTIONS SET MODEM INTERRUPT
AMBA: BIC #BIT3,@RXCSR ;CLEAR SUP REC
JSR %7,TIME ;DELAY
BIS #BIT3,@RXCSR ;SET SUP REC
JSR %7,TIME ;DELAY
BIT #BIT15,@RXCSR ;TEST MODEM INTERRUPT
BNE AMBB ;MODEM INTERRUPT SHOULD BE SET
ERROR
BR AMBE
AMBB: BIT #BIT15,@RXCSR ;MODEM INTERRUPT SHOULD BE
BEQ AMBC ;CLEARED BY PREVIOUS READ
ERROR
BR AMBE
AMBC: BIC #BIT3,@RXCSR ;1-0 TRANS OF SUP REC DATA
JSR %7,TIME ;DELAY
BIT #BIT15,@RXCSR ;TEST MODEM INTERRUPT

```

```

2129 011224 001002          BNE      AMBD          ; SHOULD BE SET
2130 011226 104003          ERROR
2131 011230 000412          BR       AMBE
2132 011232 052777 000010 170150 AMBD:  BIS    #BIT3,@RXCSR  ; C-1 TRANS OF SUP REC DATA
2133 011240 004737 012122          JSR     %7,TIME      ; DELAY
2134 011244 032777 100000 170136  BIT    #BIT15,@RXCSR ; TEST MODEM INTERRUPT
2135 011252 001001          BNE     AMBE         ; SHOULD BE SET
2136 011254 104003          ERROR
2137 011256 104012          AMBE:  SCOPE
2138 011260          TSTA   10.,ABA,CD
2139 011260          TSTAA  ABA,10.,\X+1+CD,\X+2,\X+1
2140          ;*****
2141 011260 100031          AT31:  100031          ; TEST NUMBER
2142 011262 011364          AT32          ; ADDRESS OF NEXT TEST
2143 011264 000012          10.          ; ITERATION COUNT
2144 011266 011270          ABAA          ; SCOPE ENTRY POINT
2145          X=X+1
2146          ;*****
2147          ; TEST THAT RESET CLEARS ALL TXCSR BITS, AND SETS BIT 7 (READY)
2148 011270 012737 000340 177776 ABAA:  MOV    #PRTY7,PSW  ; SET PRIORITY 7.
2149 011276 012777 177777 170110  MOV    #-1,@TXCSR   ; SET ALL POSSIBLE BITS IN TXCSR
2150 011304 104011          SRESET          ; ISSUE RESET TO CLEAR BITS
2151 011306 022777 000200 170100  CMP    #BIT7,@TXCSR ; SEE IF ONLY BIT 7 IS SET.
2152 011314 001422          BEQ    ABAB        ; BRANCH IF ONLY BIT 7 IS SET
2153 011316 017737 170072 001606  MOV    @TXCSR,TXCSR ; SAVE CONTENTS OF TXCSR
2154 011324 012737 000200 001620  MOV    #BIT7,TEMP   ; MOVE EXPECTED TXCSR TO TEMP.
2155 011332 004537 004624          JSR    %5,OACNV     ; GO TO OCTAL TO ASCII CONVERT.
2156 011336 001620          TEMP          ; SOURCE ADDR.
2157 011340 016306          ATXSB          ; DESTINATION ADDR.
2158 011342 000006          6             ; #OF DIGITS TO CONVERT.
2159 011344 004537 004624          JSR    %5,OACNV     ; GO TO OCTAL TO ASCII CONVERT.
2160 011350 001606          TXCSR         ; SOURCE ADDR.
2161 011352 016323          ATXWAS        ; DESTINATION ADDR.
2162 011354 000006          6             ; #OF DIGITS TO CONVERT.
2163 011356 104015          ERROR1        ; RESET FAILED TO CLEAR ALL BITS EXCEPT
2164 011360 016273          ATXCSR        ; BIT 7 - SEE PRINTOUT
2165 011362 104012          ABAB:  SCOPE
2166 011364          TSTA   10.,ACA,0
2167 011364          TSTAA  ACA,10.,\X+1+0,\X+2,\X+1
2168          ;*****
2169 011364 000032          AT32:  32          ; TEST NUMBER
2170 011366 011534          AT33          ; ADDRESS OF NEXT TEST
2171 011370 000012          10.          ; ITERATION COUNT
2172 011372 011374          ACAA          ; SCOPE ENTRY POINT
2173          X=X+1
2174          ;*****
2175          ; TEST THAT RESET CLEARS ALL RXCSR BITS EXCEPT DATA TERMINAL READY, RING
2176          ; CLEAR TO SEND, CARRIER DET
2177 011374 012737 000340 177776 ACAA:  MOV    #PRTY7,PSW  ; SET PRIORITY 7
2178 011402 042777 000002 170000  BIC    #BIT1,@RXCSR ; CLEAR DATA TERM.READY
2179 011410 012777 177777 167772  MOV    #-1,@RXCSR   ; SET ALL POSSIBLE BITS IN RXCSR
2180 011416 052777 000004 167770  BIS    #4,@TXCSR    ; SET MAINT BIT
2181 011424 005077 167766          CLR    @TXBUF      ; TRANSMIT A CHAR
2182 011430 104020          TIMETX        ; TIME OUT TX DONE
2183 011432 104003          ERROR        ; ERROR DONE NOT SETTING
2184 011434 012777 000001 167754  MOV    #1,@TXBUF    ; TRANSMIT ANOTHER CHAR.

```

```
2185 011442 104017          TIMERX          :TIME OUT RX DONE
2186 011444 104003          ERROR           :ERROR DONE NOT SETTING
2187 011446 104011          SRESET         :ISSUE RESET TO CLEAR BITS.
2188 011450 017737 167734 001610 MOV @RXCSR,RXCRT :MOVE RXCSR CONTENTS TO RXCSR
2189 011456 022737 030002 001610 CMP #30002,RXCRT :SEE IF ONLY BITS 1,12,13 SET
2190 011464 001417          BEQ ACAB        :BRANCH IF ONLY BITS 1,12,13 SET.
2191 011466 012737 030002 001620 MOV #30002,TEMP
2192 011474 004537 004624 JSR %5,OACNV     :GO TO OCTAL TO ASCII CONVERT.
2193 011500 001620          TEMP         :SOURCE ADDR.
2194 011502 016345          ARXSB         :DESTINATION ADDR.
2195 011504 000006          6            :#OF DIGITS TO CONVERT.
2196 011506 004537 004624 JSR %5,OACNV     :GO TO OCTAL TO ASCII CONVERT.
2197 011512 001610          RXCSR        :SOURCE ADDR.
2198 011514 016362          ARXWAS       :DESTINATION ADDR.
2199 011516 000006          6            :#OF DIGITS TO CONVERT.
2200 011520 104015          ERROR1        :RESET FAILED TO CLEAR ALL BITS EXCEPT
2201 011522 016332          ARXCSR       :BIT 0. SEE ERROR PRINTOUT.
2202 011524 042777 000002 167656 ACAB: BIC #BIT1,@RXCSR :CLEAR DATA TERM. READY
2203 011532 104012          SCOPE       :SCOPE
2204 011534          TSTA 10.,ADA,CD
2205 011534          TSTAA ADA,%10.,\X+1+CD,\X+2,\X+1
2206
2207 011534 100033          AT33: 100033 :TEST NUMBER
2208 011536 011614          AT34        :ADDRESS OF NEXT TEST
2209 011540 000012          10.         :ITERATION COUNT
2210 011542 011544          ADA        :SCOPE ENTRY POINT
2211          000033
2212
2213          :*****
2214          :TEST THAT LOADING TXBUF (TRANSMITTER BUFFER) CLEARS TXCSR BIT 7 (READY)
2215          :AND WITHOUT MAINT SET THAT TXDONE SETS READY
2215 011544 005077 167646          ADAA: CLR @TXBUF :LOAD TXBUF
2216 011550 104020          TIMETX      :TIME OUT TX DONE
2217 011552 104003          ERROR       :ERROR, DCNE NOT SETTING
2218 011554 005077 167636          CLR @TXBUF   :LOAD TX BUF
2219 011560 105777 167630          TSTB @TXCSR  :TEST TXCSR BIT 7 (READY BIT)
2220 011564 100002          BPL ADAB     :BRANCH IF BIT NOT SET.
2221 011566 104003          ERROR      :ERROR. LOADING TXBUF FAILED TO CLEAR READY.
2222 011570 000407          BR ADAC
2223 011572 104020          ADAB: TIMETX :WAIT FOR DONE
2224 011574 104003          ERROR      :DONE NEVER SET
2225 011576 032777 000200 167610          BIT #BIT7,@TXCSR
2226 011604 001001          BNE .+4
2227 011606 104003          ERROR      :READY DID NOT SET
2228 011610 104011          ADAC: SRESET
2229 011612 104012          SCOPE      :SCOPE.
2230 011614          TSTA 1.,AIA,CD
2231 011614          TSTAA AIA,%10.,\X+1+CD,\X+2,\X+1
2232
2233 011614 100034          AT34: 100034 :TEST NUMBER
2234 011616 012136          AT35        :ADDRESS OF NEXT TEST
2235 011620 000001          1.         :ITERATION COUNT
2236 011622 011624          AIA        :SCOPE ENTRY POINT
2237          000034
2238
2239          :*****
2240          :TEST THAT TRANSMIT SPEEDS ARE ARRANGED IN ASCENDING ORDER BY CHECKING THAT TIME
          :TO READY BIT (TXCSR BIT 7) DECREASES AS A HIGHER SPEED IS SELECTED.
```

2241	011624	004737	006352	AIAA:	JSR	%7,DOTHIS	;TEST IF THIS TEST SELECTED
2242	011630	104001			TYPES		
2243	011632	017653			MSETTX		
2244	011634	017701			MSETC		
2245	011636	020163			MS0		
2246	011640	177777			-1		
2247	011642	000000			HALT		
2248	011644	004737	012052		JSR	%7,AIAS	;OUTPUT CHAR AND TIME.
2249	011650	013737	012134	001566	MOV	AIAS,CTR0	;MOVE ELAPSED TIME TO CTR0.
2250	011656	104000			TYPE		
2251	011660	020173			MS1		
2252	011662	000000			HALT		
2253	011664	004737	012052		JSR	%7,AIAS	;OUTPUT CHAR AND TIME.
2254	011670	013737	012134	001570	MOV	AIAS,CTR1	;MOVE ELAPSED TIME TO CTR1.
2255	011676	104000			TYPE		
2256	011700	020203			MS2		
2257	011702	000000			HALT		
2258	011704	004737	012052		JSR	%7,AIAS	;OUTPUT CHAR AND TIME.
2259	011710	013737	012134	001572	MOV	AIAS,CTR2	;MOVE ELAPSED TIME TO CTR2.
2260	011716	104000			TYPE		
2261	011720	020213			MS3		
2262	011722	000000			HALT		
2263	011724	004737	012052		JSR	%7,AIAS	;OUTPUT CHAR AND TIME.
2264	011730	013737	012134	001574	MOV	AIAS,CTR3	;MOVE ELAPSED TIME TO CTR3.
2265	011736	104000			TYPE		
2266	011740	020223			MS4		
2267	011742	000000			HALT		
2268	011744	004737	012052		JSR	%7,AIAS	;OUTPUT CHAR AND TIME
2269	011750	013737	012134	001576	MOV	AIAS,CTR4	;MOVE ELAPSED TIME TO CTR4
2270	011756	104000			TYPE		
2271	011760	020233			MS5		
2272	011762	000000			HALT		
2273	011764	004737	012052		JSR	%7,AIAS	;OUTPUT CHAR AND TIME
2274	011770	013737	012134	001600	MOV	AIAS,CTR5	;MOVE ELAPSED TIME TO CTR5
2275	011776	104000			TYPE		
2276	012000	020243			MS6		
2277	012002	000000			HALT		
2278	012004	004737	012052		JSR	%7,AIAS	;OUTPUT CHAR AND TIME
2279	012010	013737	012134	001602	MOV	AIAS,CTR6	;MOVE ELAPSED TIME TO CTR6
2280	012016	104000			TYPE		
2281	012020	020253			MS7		
2282	012022	000000			HALT		
2283	012024	004737	012052		JSR	%7,AIAS	;OUTPUT CHAR AND TIME
2284	012030	013737	012134	001604	MOV	AIAS,CTR7	;MOVE ELAPSED TIME TO CTR7
2285	012036	004737	014564		JSR	%7,CMPT	;CHECK THAT CTR0 THROUGH CTR7 CONTAIN
2286	012042	000402			BR	AIAF	;DESCENDING VALUES
2287	012044	104015			ERROR1		;TRANSMIT SPEEDS NOT ARRANGED IN
2288	012046	016372			ETXTIM		;ASCENDING ORDER.
2289	012050	104012			SCOPE		;SCOPE
2290				AIAF:			
2291	012052	005037	012134	AIAS:	CLR	AIAS	;CLEAR ELAPSED TIME COUNTER.
2292	012056	105777	167332		TSTB	@TXCSR	;WAIT FOR TX READY.
2293	012062	100375			BPL	.-4	
2294	012064	005077	167326		CLR	@TXBUF	
2295	012070	105777	167320		TSTB	@TXCSR	
2296	012074	100375			BPL	.-4	

2297	012076	005077	167314		CLR	@TXBUF	:LOAD TXBUF.
2298	012102	004737	012122	AIASA:	JSR	%7,TIME	:WAIT 75 US
2299	012106	005237	012134		INC	AIAS	:INCREMENT ELAPSED TIME COUNTER.
2300	012112	105777	167276		TSTB	@TXCSR	:READY SET?
2301	012116	100371			BPL	AIASA	:BRANCH IF READY NOT SET.
2302	012120	000207			RTS	%7	:EXIT.
2303							
2304	012122	012700	000017	TIME:	MOV	#15.,%0	
2305	012126	005300		TIM1:	DEC	%0	
2306	012130	001376			BNE	TIM1	
2307	012132	000207			RTS	%7	
2308	012134	000000		AIAS:	OPEN		
2309	012136				TSTA	10.,ALA,0	
2310	012136				TSTAA	ALA,10.,\X+1+0,\X+2,\X+1	
2311							
2312	012136	000035		AT35:	35		:TEST NUMBER
2313	012140	012212			AT36		:ADDRESS OF NEXT TEST
2314	012142	000012			10.		:ITERATION COUNT
2315	012144	012146			ALAA		:SCOPE ENTRY POINT
2316		000035			X=X+1		
2317							
2318							
2319							
2320							
2321							
2322	012146	052777	000004	167240	ALAA:	BIS #BIT2,@TXCSR	:SET MAINTENANCE BIT
2323	012154	005077	167236		CLR	@TXBUF	:LOAD TXBUF
2324	012160	104016			DELAY		:WAIT 500 MSECS.
2325	012162	000764			500.		
2326	012164	105777	167220		TSTB	@RXCSR	:SEE IF DONE BIT IS SET
2327	012170	100402			BMI	ALAB	:BRANCH IF DONE BIT IS SET
2328	012172	104003			ERROR		:DONE BIT FAILED TO SET
2329	012174	000405			BR	ALAC	
2330	012176	104011			ALAB:	SRESET	:ISSUE RESET TO CLEAR DONE BIT
2331	012200	105777	167204		TSTB	@RXCSR	:SEE IF DONE BIT IS CLEARED
2332	012204	100001			BPL	ALAC	:BRANCH IF DONE BIT IS CLEARED
2333	012206	104003			ERROR		:RESET FAILED TO CLEAR DONE BIT
2334	012210	104012			ALAC:	SCOPE	:SCOPE
2335	012212				TSTA	100.,AMA,CD	
2336	012212				TSTAA	AMA,100.,\X+1+CD,\X+2,\X+1	
2337							
2338	012212	100036		AT36:	100036		:TEST NUMBER
2339	012214	012256			AT37		:ADDRESS OF NEXT TEST
2340	012216	000144			100.		:ITERATION COUNT
2341	012220	012222			AMAA		:SCOPE ENTRY POINT
2342		000036			X=X+1		
2343							
2344							
2345							
2346	012222	052777	000004	167164	AMAA:	BIS #BIT2,@TXCSR	:SET MAINTENANCE BIT (TXCSR BIT 2)
2347	012230	005077	167162		CLR	@TXBUF	:LOAD TXBUF
2348	012234	104017			TIMERX		:WAIT FOR DONE BIT TO SET.
2349	012236	104003			ERROR		
2350	012240	005777	167146		TST	@RXBUF	:READ RXBUF TO CLEAR DONE BIT
2351	012244	105777	167140		TSTB	@RXCSR	:SEE IF DONE BIT IS CLEAR
2352	012250	100001			BPL	AMAC	:BRANCH IF DONE BIT IS CLEAR

```
2353 012252 104003
2354 012254 104012
2355 012256
2356 012256
2357
2358 012256 100037
2359 012260 012366
2360 012262 000144
2361 012264 012272
2362 000037
2363
2364
2365
2366 012266 004737 003004
2367 012272 052777 000004 167114
2368 012300 005077 167112
2369 012304 005037 001620
2370 012310 032777 004000 167072
2371 012316 001402
2372 012320 005237 001620
2373 012324 105777 167060
2374 012330 100367
2375 012332 023727 001620 000000
2376 012340 001002
2377 012342 104003
2378 012344 000405
2379 012346 032777 004000 167034
2380 012354 001401
2381 012356 104003
2382 012360 005777 167026
2383 012364 104012
2384 012366
2385 012366
2386
2387 012366 000040
2388 012370 012670
2389 012372 000001
2390 012374 012576
2391 000040
2392
2393
2394
2395
2396
2397
2398 012376 004737 006352
2399 012402 104001
2400 012404 017625
2401 012406 017701
2402 012410 020163
2403 012412 177777
2404 012414 000000
2405 012416 004737 012624
2406 012422 013737 012666 001566
2407 012430 104000
2408 012432 020173
```

AMAC: ERROR ;READING RXBUF FAILED TO CLEAR DONE BIT
SCOPE ;SCOPE
TSTA 100.,AOA,CD
TSTAA AOA,100.,\X+1+CD,\X+2,\X+1

AT37: 100037 ;TEST NUMBER
AT40 ;ADDRESS OF NEXT TEST
100. ;ITERATION COUNT
AOAA ;SCOPE ENTRY POINT
X=X+1

:TEST THAT RECEIVER ACTIVE SETS WHEN CHAR STARTS AND
:CLEARS WHEN RECEIVER DONE SETS

AOAA: JSR %7,CDINIT ;INIT IF C-D DEVICE
BIS #BIT2,@TXCSR ;SET MAINT
CLR @TXBUF ;TRANSMIT CHAR
CLR TEMP ;CLEAR BUSY INDICATOR
AOAB: BIT #BIT11,@RXCSR ;IS RECEIVER ACTIVE SET
BEQ AOAB1 ;BRANCH IF CLEAR
INC TEMP ;YES, REMEMBER THAT
AOAB1: TSTB @RXCSR ;SEE IF DONE SET
BPL AOAB
CMP TEMP,#0 ;DID RECEIVER ACTIVE SET
BNE AOAC
ERROR ;RECEIVER ACTIVE NEVER SET
BR AOAD
AOAC: BIT #BIT11,@RXCSR ;DID DONE CLEAR ACTIVE
BEQ AOAD
ERROR ;NO, RECEIVER ACTIVE DID NOT CLEAR
AOAD: TST @RXBUF ;CLEAR RX DONE
SCOPE
TSTA 1.,AOA,0
TSTAA AOA,1.,\X+1+0,\X+2,\X+1

AT40: 40 ;TEST NUMBER
AT41 ;ADDRESS OF NEXT TEST
1. ;ITERATION COUNT
AOAA ;SCOPE ENTRY POINT
X=X+1

:TEST THAT RECEIVF SPEEDS ARE ARRANGED IN ASCENDING ORDER BY CHECKING THAT TIME
:ELAPSED TO DONE BIT SETTING (RXCSR BIT 7) DECREASES AS A HIGHER SPEED
:THIS IS NOT DONE IN MAINTENANCE MODE TX AND RX
:POTS MUST BE STEPPED TOGETHER
:IS SELECTED.

AOAA: JSR %7,DOTHIS ;CHECK IF THIS TEST TO BE DONE
TYPES
MSETRX
MSETC
MS0
-1
HALT
JSR %7,AQAS ;OUTPUT CHARACTER AND TIME DONE BIT
MOV AQAST,CTRO ;MOVE ELAPSED TIME TO CTRO
TYPE
MS1


```

2409 012434 000000          HALT
2410 012436 004737 012624  JSR      %7,AQAS      ;OUTPUT CHARACTER AND TIME DONE BIT
2411 012442 013737 012666 001570  MOV      AQAST,CTR1   ;MOVE ELAPSED TIME TO CTR1
2412 012450 104000          TYPE
2413 012452 020203          MS2
2414 012454 000000          HALT
2415 012456 004737 012624  JSR      %7,AQAS      ;OUTPUT CHARACTER AND TIME DONE BIT.
2416 012462 013737 012666 001572  MOV      AQAST,CTR2   ;MOVE ELAPSED TIME TO CTR2.
2417 012470 104000          TYPE
2418 012472 020213          MS3
2419 012474 000000          HALT
2420 012476 004737 012624  JSR      %7,AQAS      ;OUTPUT CHARACTER AND TIME DONE BIT
2421 012502 013737 012666 001574  MOV      AQAST,CTR3   ;MOVE ELAPSED TIME TO CTR3.
2422 012510 104000          TYPE
2423 012512 020223          MS4
2424 012514 000000          HALT
2425 012516 004737 012624  JSR      %7,AQAS      ;OUTPUT CHARACTER AND TIME DONE BIT
2426 012522 013737 012666 001576  MOV      AQAST,CTR4   ;MOVE ELAPSED TIME TO CTR4.
2427 012530 104000          TYPE
2428 012532 020233          MS5
2429 012534 000000          HALT
2430 012536 004737 012624  JSR      %7,AQAS      ;OUTPUT CHARACTER AND TIME DONE BIT
2431 012542 013737 012666 001600  MOV      AQAST,CTR5   ;MOVE ELAPSED TIME TO CTR5.
2432 012550 104000          TYPE
2433 012552 020243          MS6
2434 012554 000000          HALT
2435 012556 004737 012624  JSR      %7,AQAS      ;OUTPUT CHARACTER AND TIME DONE BIT
2436 012562 013737 012666 001602  MOV      AQAST,CTR6   ;MOVE ELAPSED TIME TO CTR6.
2437 012570 104000          TYPE
2438 012572 020253          MS7
2439 012574 000000          HALT
2440 012576 004737 012624  JSR      %7,AQAS      ;OUTPUT CHARACTER AND TIME DONE BIT
2441 012602 013737 012666 001604  MOV      AQAST,CTR7   ;MOVE ELAPSED TIME TO CTR7.
2442 012610 004737 014564  JSR      %7,CMPT      ;CHECK THAT CTR0 THROUGH CTR3 CONTAIN
2443 012614 000402          BR        AQAB         ;DESCENDING VALUES.
2444 012616 104015          ERROR1   ;RECEIVE SPEEDS NOT ARRANGED IN
2445 012620 016434          ERXTIM   ;ASCENDING ORDER.
2446 012622 104012          AQAB:    SCOPE       ;SCOPE
2447
2448 012624 005037 012666  AQAS:    CLR        AQAST      ;CLEAR ELAPSED TIME COUNTER AQAST
2449 012630 105777 166560  TSTB    @TXCSR      ;WAIT FOR TX READY.
2450 012634 100375          BPL     -4
2451 012636 005777 166550  TST     @RXBUF      ;CLEAR DONE BIT IF SET
2452 012642 005077 166550  CLR     @TXBUF      ;LOAD TXBUF
2453 012646 004737 012122  AQASA:  JSR      %7,TIME
2454 012652 005237 012666  INC     AQAST        ;INCREMENT ELAPSED TIME COUNTER
2455 012656 105777 166526  TSTB    @RXCSR      ;DONE SET?
2456 012662 100371          BPL     AQASA        ;BRANCH IF DONE NOT SET
2457 012664 000207          RTS     %7          ;EXIT
2458 012666 000000          AQAST:  OPEN     ;ELAPSED TIME COUNTER
2459 012670          TSTA    10.,ARA,CD
2460 012670          TSTAA   ARA,10.,\X+1+(D,\X+2,\X+1
2461
2462 012670 100041          AT41:   100041          ;TEST NUMBER
2463 012672 013034          AT42   ;ADDRESS OF NEXT TEST
2464 012674 000012          10.    ;ITERATION COUNT

```

```
2465 012676 012700 ARAA ;SCOPE ENTRY POINT
2466 00004i X-X+1
2467 :*****
2468 :TEST CORRECT OPERATION OF DATA OVERRUN BIT (RXBUF BIT 14)
2469 012700 004737 013014 ARAA: JSR %7,ARAS ;OUTPUT CHARACTER AND WAIT 500 MSECS
2470 012704 004737 013014 JSR %7,ARAS ;OUTPUT CHARACTER AND WAIT 500 MSECS
2471 012710 017737 166476 001612 MOV @RXBUF,RXBUFT ;SAVE RXBUF CONTENTS + CLEAR DONE
2472 012716 032737 040000 001612 BIT #BIT14,RXBUFT ;SEE IF DATA OVERRUN BIT WAS SET
2473 012724 001002 BNE .+6 ;BRANCH IF BIT WAS SET
2474 012726 104003 ERROR
2475 012730 104012 SCOPE
2476 012732 005737 001612 TST RXBUFT ;SEE THAT ERROR BIT WAS SET (RXBUF BIT 15)
2477 012736 100402 BMI .+6
2478 012740 104003 ERROR ;ERROR BIT FAILED TO SET WHEN OVERRUN SET
2479 012742 104012 SCOPE
2480 012744 032777 040000 166440 BIT #BIT14,@RXBUF ;SEE THAT DATA OVERRUN WAS NOT
2481 ;CLEARED WHEN RXBUF WAS READ
2482 012752 001002 BNE .+6 ;BRANCH IF SET
2483 012754 104003 ERROR ;READING RXBUF CLEARED DATA OVERRUN
2484 012756 104012 SCOPE
2485 012760 004737 013014 JSR %7,ARAS
2486 012764 032777 100000 166420 BIT #BIT15,@RXBUF ;OUTPUT CHAR +WAIT 500MS
2487 012772 001402 BEQ .+6 ;TEST THAT ERROR CLEARED
2488 012774 104003 ERROR
2489 012776 104012 SCOPE
2490 013000 032777 040000 166404 BIT #BIT14,@RXBUF ;TEST THAT OVERRUN CLEARED
2491 013006 001401 BEQ .+4
2492 013010 104003 ERROR
2493 013012 104012 SCOPE ;SCOPE
2494 013014 052777 000004 166372 ARAA: BIS #BIT2,@TXCSR ;SET MAINTENANCE BIT
2495 013022 005077 166370 CLR @TXBUF ;LOAD TXBUF
2496 013026 104016 DELAY ;DELAY 500 MSECS
2497 013030 000764 500.
2498 013032 000207 RTS %7 ;EXIT
2499 013034 TSTA 10.,ATA,CD
2500 013034 TSTAA ATA,10.,\X+1+CD,\X+2,\X+1
2501 :*****
2502 013034 100042 AT42: 100042 ;TEST NUMBER
2503 013036 013122 AT43 ;ADDRESS OF NEXT TEST
2504 013040 000012 10. ;ITERATION COUNT
2505 013042 013054 ATAA ;SCOPE ENTRY POINT
2506 000042 X=X+1
2507 :*****
2508 ;TEST THAT TRANSMITTER IS ABLE TO INTERRUPT. IF THE INTERRUPT IS SERVICED.
2509 ;IT WILL HAVE OCCURRED AT THE CORRECT VECTOR.
2510 013044 004737 003614 JSR 7,OVRLAY ;GO TO OVER LAY ROUTINE
2511 013050 104007 STTXV ;SET TX INTERRUPT SERVICE
2512 013052 013110 ATAC ;TO ATAC
2513 013054 042777 000100 166332 ARAA: BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPT
2514 013062 005037 177776 CLR PSW ;SET PROCESSOR PRIORITY TO 0
2515 013066 052777 000100 166320 BIS #BIT6,@TXCSR ;ENABLE TX INTERRUPT
2516 013074 000240 NOP
2517 013076 104003 ERROR ;READY DID NOT CAUSE AN INTERRUPT
2518 013100 042777 000100 166306 BIC #BIT6,@TXCSR
2519 013106 104012 ATAB: SCOPE ;SCOPE
2520 013110 042777 000100 166276 ATAC: BIC #BIT6,@TXCSR ;HERE IF INT. DISABLE TX INT
```

```
2521 013116 022626 POPSP2
2522 013120 000772 BR ATAB
2523 013122 TSTA 1000.,AUA,CD
2524 013122 TSTAA AUA,1000.,\X+1+CD,\X+2,\X+1
2525
*****
2526 013122 100043 AT43: 100043 ;TEST NUMBER
2527 013124 013200 AT44 ;ADDRESS OF NEXT TEST
2528 013126 001750 1000. ;ITERATION COUNT
2529 013130 013136 AUAA ;SCOPE ENTRY POINT
2530 000043 X=X+1
*****
2531
;TEST THAT READY DOES NOT CAUSE AN INTERRUPT WHEN THE PROCESSOR IS
2532 ;AT THE SAME PRIORITY AS THE TRANSMITTER INTERRUPT REQUEST LEVEL
2533
2534 013132 104007 STTXV ;SET TX INTERRUPT SERVICE TO
2535 013134 013172 AUAC
2536 013136 013737 001426 177776 AUAA: MOV TXLVL,PSW ;SET PROCESSOR PRIORITY SAME AS TX PRIORITY
2537 013144 042777 000100 166242 BIC #BIT6,@TXCSR
2538 013152 052777 000100 166234 BIS #BIT6,@TXCSR ;ENABLE TX INTERRUPTS
2539 013160 000240 NOP
2540 013162 042777 000100 166224 AUAB: BIC #BIT6,@TXCSR ;OK IF NO INTERRUPT OCCURS. DISABLE INTERRUPTS
2541 013170 104012 SCOPE ;SCOPE
2542 013172 022626 AUAC: POPSP2 ;HERE IF INTERRUPT OCCURS. POP STOCK TWICE
2543 013174 104003 ERROR ;TX INTERRUPTED WITH PROCESSOR AT SAME
2544 013176 000771 BR AUAB ;PRIORITY AS THE TRANSMITTER
2545 013200 TSTA 10.,AVA,CD
2546 013200 TSTAA AVA,10.,\X+1+CD,\X+2,\X+1
*****
2547
2548 013200 100044 AT44: 100044 ;TEST NUMBER
2549 013202 013264 AT45 ;ADDRESS OF NEXT TEST
2550 013204 000012 10. ;ITERATION COUNT
2551 013206 013214 AVAA ;SCOPE ENTRY POINT
2552 000044 X=X+1
*****
2553
;TEST THAT TRANSMITTER INTERRUPTS WHEN PROCESSOR IS AT PRIORITY ONE LEVEL
2554 ;LOWER THAN THE TRANSMITTER INTERRUPT PRIORITY.
2555
2556 013210 104007 STTXV ;SET TX INTERRUPT SERVICE TO AVAB
2557 013212 013252 AVAB
2558 013214 042777 000100 166172 AVAA: BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPTS
2559 013222 013737 001426 177776 MOV TXLVL,PSW ;SET PROCESSOR PRIORITY TO ONE LEVEL
2560 013230 162737 000040 177776 SUB #40,PSW ;LOWER THAN TX PRIORITY
2561 013236 052777 000100 166150 BIS #BIT6,@TXCSR ;ENABLE TX INTERRUPTS
2562 013244 000240 NOP
2563 013246 104003 ERROR ;TX FAILED TO INTERRUPT
2564 013250 000401 BR AVAC
2565 013252 022626 AVAB: POPSP2 ;HERE IF INTERRUPT OCCURS. POP STOCK TWICE
2566 013254 042777 000100 166132 AVAC: BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPTS
2567 013262 104012 SCOPE ;SCOPE
2568 013264 TSTA 100.,AWA,CD
2569 013264 TSTAA AWA,100.,\X+1+CD,\X+2,\X+1
*****
2570
2571 013264 100045 AT45: 100045 ;TEST NUMBER
2572 013266 013362 AT46 ;ADDRESS OF NEXT TEST
2573 013270 000144 100. ;ITERATION COUNT
2574 013272 013274 AWAA ;SCOPE ENTRY POINT
2575 000045 X=X+1
*****
2576
```

```
2577 ;TEST THAT TRANSMITTER DOES NOT REINTERRUPT AFTER THE INITIAL INTERRUPT HAS
2578 ;OCCURRED AND HAS BEEN SERVICED.
2579 013274 104007 AWAA: STTXV ;SET TX INTERRUPT SERVICE TO AWAC
2580 013276 013334 AWAC
2581 013300 042777 000100 166106 BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPTS
2582 013306 005037 177776 CLR PSW ;SET PROCESSOR PRIORITY TO 0
2583 013312 052777 000100 166074 BIS #BIT6,@TXCSR ;ENABLE TX INTERRUPTS
2584 013320 000240 NOP
2585 013322 104003 ERROR ;TRANSMITTER FAILED TO INTERRUPT
2586 013324 042777 000100 166062 AWAB: BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPTS
2587 013332 104012 SCOPE ;SCOPE
2588 013334 012777 013354 166062 AWAC: MOV #AWAE,@TXVTR ;HERE IF INTERRUPT OCCURS. CHANGE EXIT
2589 013342 012716 013350 MOV #AWAD,@%6 ;POINTER TO AWAD AND EXIT INTERRUPT
2590 013346 000002 RTI
2591 013350 000240 AWAD: NOP ;OK IF NO INTERRUPT REOCCURS.
2592 013352 000764 BR AWAB
2593 013354 022626 AWAE: POPSP2 ;HERE IF INTERRUPT REOCCURS
2594 013356 104003 ERROR ;TX REINTERRUPTED AFTER RTI
2595 013360 000761 BR AWAB
2596 013362 TSTA 10.,AXA,CD
2597 013362 TSTAA AXA,10.,\X+1+CD,\X+2,\X+1
2598 ;*****
2599 013362 100046 AT46: 100046 ;TEST NUMBER
2600 013364 013446 AT47 ;ADDRESS OF NEXT TEST
2601 013366 000012 10. ;ITERATION COUNT
2602 013370 013406 AXAA ;SCOPE ENTRY POINT
2603 000046 X=X+1
2604 ;*****
2605 ;TEST THAT RECEIVER DONE BIT IS ABLE TO INTERRUPT. IF THE INTERRUPT IS
2606 ;SERVICED IT WILL HAVE OCCURRED AT THE CORRECT VECTOR.
2607 013372 004737 003614 JSR 7,OVRLAY ;GO TO OVERLAY ROUTINE
2608 013376 104006 STRXV ;SET RX INTERRUPT SERVICE TO AXAB
2609 013400 013434 AXAB
2610 013402 004737 014544 JSR %7,STRXD ;SET RX DONE BIT
2611 013406 042777 000100 165774 AXAA: BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
2612 013414 005037 177776 CLR PSW ;SET PROCESSOR PRIORITY TO 0
2613 013420 052777 000100 165762 BIS #BIT6,@RXCSR ;ENABLE RX INTERRUPTS
2614 013426 000240 NOP
2615 013430 104003 ERROR ;RX FAILED TO INTERRUPT
2616 013432 000401 BR AXAC
2617 013434 022626 AXAB: POPSP2 ;HERE IF INTERRUPT OCCURS
2618 013436 042777 000100 165744 AXAC: BIC #BIT6,@RXCSR ;DISABLE INT EN
2619 013444 104012 SCOPE ;SCOPE
2620 013446 TSTA 10.,AX1,0
2621 013446 TSTAA AX1,10.,\X+1+0,\X+2,\X+1
2622 ;*****
2623 013446 000047 AT47: 47 ;TEST NUMBER
2624 013450 013530 AT50 ;ADDRESS OF NEXT TEST
2625 013452 000012 10. ;ITERATION COUNT
2626 013454 013466 AX1A ;SCOPE ENTRY POINT
2627 000047 X=X+1
2628 ;*****
2629 ;TEST THAT MODEM INTERRUPT BIT IS ABLE TO INTERRUPT. IF THE INTERRUPT IS
2630 ;SERVICED IT WILL HAVE OCCURRED AT THE CORRECT VECTOR.
2631 013456 004737 003614 JSR 7,OVRLAY ;GO TO OVERLAY ROUTINE
2632 013462 104006 STRXV ;SET RX INTERRUPT SERVICE TO AXAB
```

```
2633 013464 013516 AX1B: AX1B
2634 013466 042777 000044 165714 AX1A: BIC #44,@RXCSR ;DISABLE MODEM INTERRUPTS
2635 013474 005037 177776 CLR PSW ;SET PROCESSOR PRIORITY TO 0
2636 013500 052777 000044 165702 BIS #44,@RXCSR ;ENABLE MODEM INTERRUPTS,RQ TO SND
2637 013506 104016 DELAY
2638 013510 000005 5.
2639 013512 104003 ERROR ;MODEM FAILED TO INTERRUPT
2640 013514 000401 BR AX1C
2641 013516 022626 AX1B: POPSP2 ;HERE IF INTERRUPT OCCURS
2642 013520 042777 000040 165662 AX1C: BIC #BIT5,@RXCSR ;DISABLE INT EN
2643 013526 104012 SCOPE
2644 013530 TSTA 1000.,AYA,CD
2645 013530 TSTAA AYA,1000.,\X+1+CD,\X+2,\X+1
2646 .....
2647 013530 100050 AT50: 100050 ;TEST NUMBER
2648 013532 013612 AT51 ;ADDRESS OF NEXT TEST
2649 013534 001750 1000. ;ITERATION COUNT
2650 013536 013550 AYAA ;SCOPE ENTRY POINT
2651 000050 X=X+1
2652 .....
2653 ;TEST THAT RECEIVER DONE BIT DOES NOT CAUSE AN INTERRUPT WHEN THE PROCESSOR
2654 ;IS AT THE SAME PRIORITY LEVEL AS THE RECEIVER INTERRUPT REQUEST LEVEL
2655 013540 104006 STRXV ;SET RX INTERRUPT SERVICE TO AYAC
2656 013542 013604 AYAC
2657 013544 004737 014544 JSR %7,STRXD ;SET RX DONE BIT
2658 013550 042777 000100 165632 AYAA: BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
2659 013556 013737 001422 177776 MOV RXLVL,PSW ;SET PROCESSOR PRIORITY SAME AS RECEIVER'S
2660 013564 052777 000100 165616 BIS #BIT6,@RXCSR ;ENABLE RX INTERRUPTS
2661 013572 000240 NOP
2662 013574 042777 000100 165606 AYAB: BIC #BIT6,@RXCSR ;OK IF NO INTERRUPT. DISABLE RX INTERRUPTS
2663 013602 104012 SCOPE ;SCOPE
2664 013604 022626 AYAC: POPSP2 ;HERE IF INTERRUPT OCCURS. POP STOCK TWICE
2665 013606 104003 ERROR ;RX INTERRUPTED WITH PROCESOR AT SAME
2666 013610 000771 BR AYAB ;PRIORITY AS THE RECEIVER
2667 013612 TSTA 10.,AZA,CD
2668 013612 TSTAA AZA,10.,\X+1+CD,\X+2,\X+1
2669 .....
2670 013612 100051 AT51: 100051 ;TEST NUMBER
2671 013614 013702 AT52 ;ADDRESS OF NEXT TEST
2672 013616 000012 10. ;ITERATION COUNT
2673 013620 013632 AZAA ;SCOPE ENTRY POINT
2674 000051 X=X+1
2675 .....
2676 ;TEST THAT RECEIVER DONE BIT CAUSES INTERRUPT WHEN PROCESSOR IS AT PRIORITY
2677 ;ONE LEVEL LOWER THAN THE RECEIVER'S INTERRUPT REQUEST LEVEL
2678 013622 104006 STRXV ;SET RX INTERRUPT TO AZAB
2679 013624 013670 AZAB
2680 013626 004737 014544 JSR %7,STRXD ;SET RX DONE BIT
2681 013632 042777 000100 165550 AZAA: BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
2682 013640 013737 001422 177776 MOV RXLVL,PSW ;SET PROCESSOR PRIORITY ONE LEVEL
2683 013646 162737 000040 177776 SUB #40,PSW ;LOWER THAN RECEIVER'S PRIORITY
2684 013654 052777 000100 165526 BIS #BIT6,@RXCSR ;ENABLE RX INTERRUPTS
2685 013662 000240 NOP
2686 013664 104003 ERROR ;RX FAILED TO INTERRUPT WITH PROCESSOR AT
2687 013666 000401 BR AZAC ;PRIORITY ONE LEVEL LOWER THAN RECEIVER'S
2688 013670 022626 AZAB: POPSP2 ;HERE IF INTERRUPT OCCURS
```

```

2689 013672 042777 000100 165510 AZAC: BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
2690 013700 104012 SCOPE ;SCOPE
2691 013702 TSTA 100.,AAB,CD
2692 013702 TSTAA AAB,100.,\X+1+CD,\X+2,\X+1
2693
2694 013702 100052 AT52: 100052 ;TEST NUMBER
2695 013704 014000 AT53 ;ADDRESS OF NEXT TEST
2696 013706 000144 100. ;ITERATION COUNT
2697 013710 013716 AABA ;SCOPE ENTRY POINT
2698 000052 X=X+1
2699
2700 ;*****
2701 ;TEST THAT RECEIVER DOES NOT INTERRUPT AFTER THE INITIAL INTERRUPT HAS
2702 ;OCCURED AND DONE BIT HAS NOT BEEN CLEARED
2702 013712 004737 014544 JSR #7,STRXD ;SET RX DONE BIT
2703 013716 104006 AABA: STRXV ;SET RX INTERRUPT SERVICE TO AABC
2704 013720 013752 AABC
2705 013722 042777 000100 165460 BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
2706 013730 052777 000100 165452 BIS #BIT6,@RXCSR ;ENABLE RX INTERRUPTS
2707 013736 000240 NOP
2708 013740 104003 ERROR ;RX FAILED TO INTERRUPT
2709 013742 042777 000100 165440 AABB: BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
2710 013750 104012 SCOPE ;SCOPE
2711 013752 012777 013772 165440 AABC: MOV #AABE,@RXVTR ;HERE IF INTERRUPT OCCURS. CHANGE SERVICE TO
2712 013760 012716 013766 MOV #AABD,@X6 ;AABE, SET EXIT POINTER TO AABD
2713 013764 000002 RTI ;EXIT INTERRUPT SERVICE
2714 013766 000240 AABD: NOP ;OK IF NO INTERRUPT REOCCURS
2715 013770 000764 BR AABB
2716 013772 022626 AABE: POPSP2 ;HERE IF INTERRUPT REOCCURS
2717 013774 104003 ERROR ;RX REINTERRUPTED AFTER RTI
2718 013776 000761 BR AABB
2719 014000 TSTA 100.,ABB,CD
2720 014000 TSTAA ABB,100.,\X+1+CD,\X+2,\X+1
2721
2722 014000 100053 AT53: 100053 ;TEST NUMBER
2723 014002 014040 AT54 ;ADDRESS OF NEXT TEST
2724 014004 000144 100. ;ITERATION COUNT
2725 014006 014010 ABBA ;SCOPE ENTRY POINT
2726 000053 X=X+1
2727
2728 ;*****
2729 ;TEST THAT READING RXCSR DOES NOT CLEAR DONE BIT (RXCSR BIT 7 )
2729 014010 004737 014544 ABBA: JSR #7,STRXD ;SET RX DONE BIT
2730 014014 017737 165370 001610 MOV @RXCSR,RXCST ;SAVE CONTENT OF RXCSR
2731 014022 105777 165362 TSTB @RXCSR ;SEE IF DONE BIT IS CLEAR
2732 014026 100401 BMI ABBB ;BRANCH IF DONE BIT IS NOT CLEAR
2733 014030 104003 ERROR
2734 014032 005777 165354 ABBB: TST @RXBUF ;CLEAR DONE BIT IF SET
2735 014036 104012 SCOPE ;SCOPE
2736 014040 TSTA 100.,ACB,CD
2737 014040 TSTAA ACB,100.,\X+1+CD,\X+2,\X+1
2738
2739 014040 100054 AT54: 100054 ;TEST NUMBER
2740 014042 014124 AT55 ;ADDRESS OF NEXT TEST
2741 014044 000144 100. ;ITERATION COUNT
2742 014046 014054 ACBA ;SCOPE ENTRY POINT
2743 000054 X=X+1
2744
;*****

```

```

2745 ;TEST THAT DONE CAN CAUSE INT WITH ERROR SET
2746 014050 104006 STRXV ;SET RX INTERRUPT SERVICE TO ACBB.
2747 014052 014112 ACBB
2748 014054 004737 014544 ACBA: JSR 7,STRXD ;SET RX DONE BIT
2749 014060 004737 014544 JSR 7,STRXD ;SET RX DATA OFLOW
2750 014064 042777 000100 165316 BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
2751 014072 005037 177776 CLR PSW ;SET PROCESSOR PRIORITY TO 0
2752 014076 052777 000100 165304 BIS #BIT6,@RXCSR ;ENABLE RX INTERRUPTS
2753 014104 000240 NOP
2754 014106 104003 ERROR ;RX DONE FAILED TO CAUSE INTERRUPT
2755 014110 000401 BR ACBC
2756 014112 022626 ACBB: POPSP2 ;HERE IF INTERRUPT OCCURS. POP STOCK TWICE
2757 014114 042777 000100 165266 ACBC: BIC #BIT6,@RXCSR
2758 014122 104012 SCOPE
2759 014124 TSTA 3.,ADD,CD
2760 014124 TSTAA ADD,3.,\X+1+CD,\X+2,\X+1
2761 ;*****
2762 014124 100055 AT55: 100055 ;TEST NUMBER
2763 014126 014146 AT56 ;ADDRESS OF NEXT TEST
2764 014130 000003 3. ;ITERATION COUNT
2765 014132 014140 ADDA ;SCOPE ENTRY POINT
2766 000055 X=X+1
2767 ;*****
2768 ;DATA TEST USING NORMAL CONFIGURATION
2769 014134 004737 003004 JSR 7,CDINIT ;INIT IF C-D DEVICE
2770 014140 004537 005120 ADDA: JSR 5,DATIST
2771 014144 104012 SCOPE
2772 014146 TSTA 3.,APB,0
2773 014146 TSTAA APB,3.,\X+1+0,\X+2,\X+1
2774 ;*****
2775 014146 000056 AT56: 56 ;TEST NUMBER
2776 014150 014240 AT57 ;ADDRESS OF NEXT TEST
2777 014152 000003 3. ;ITERATION COUNT
2778 014154 014162 APBA ;SCOPE ENTRY POINT
2779 000056 X=X+1
2780 ;*****
2781 ;DATA TEST USING JUMPER CONNECTOR.
2782 ;USES SPECIAL BINARY COUNT PATTERN FOR DATA. NO INTERRUPT.
2783 014156 004737 004452 JSR 7,INBIN ;INITIALIZE BINARY COUNT PATTERN
2784 014162 012737 001750 001566 APBA: MOV #1000.,CTRO ;SET CHARACTER COUNT TO 1000
2785 014170 104020 APBB: TIMETX ;TIME OUT TX DONE
2786 014172 104003 ERROR ;ERROR DONE NOT SETTING
2787 014174 004737 004556 JSR 7,GTBINP ;GET BINARY CHARACTER
2788 014200 110137 001562 MOVB %1,CRBUFA ;SAVE CHAR IN CRBUFA AND
2789 014204 004737 005530 JSR 7,MASKIT ;MASK OFF NON TRANSMITTED BITS
2790 014210 110177 165202 MOVB %1,@TXBUF ;LOAD CHAR.
2791 014214 104017 TIMERX ;TIME OUT RX DONE
2792 014216 104003 ERROR ;ERROR DONE NOT SETTING
2793 014220 117737 165166 001560 MOVB @RXBUF,CRBUF ;LOAD RECEIVED DATA INTO CRBUF
2794 014226 104004 DATCHK ;CHECK DATA
2795 014230 005337 001566 DEC CTRO ;TESTED 1000 CHARACTERS
2796 014234 001355 BNE APBB ;BRANCH IF NOT
2797 014236 104012 SCOPE ;YES. SCOPE
2798 014240 TSTA 3.,EXT,0
2799 014240 TSTAA EXT,3.,\X+1+0,\X+2,\X+1
2800 ;*****

```



```

2801 014240 000057          AT57: 57          ;TEST NUMBER
2802 014242 014324          AT60          ;ADDRESS OF NEXT TEST
2803 014244 000003          3.           ;ITERATION COUNT
2804 014246 014250          EXTA         ;SCOPE ENTRY POINT
2805          000057          X=X+1
2806          ;*****
2807
2808          ;TEST THAT RDR BUSY TURNS OFF HDR ENABLE
2809          ;WHEN RUN ON AN XOR TESTER
2810 014250 000005          EXTA: RESET      ;RESET
2811 014252 005277 165132    INC @RXCSR      ;SET RDR ENABLE, SEE IF RDE IS TURNED OFF BY RDR BUSY
2812 014256 012737 177770 014314 MOV #-10,3$+2
2813 014264 005237 014314 2$: INC 3$+2      ;WAIT LOOP FOR XOR TESTER
2814 014270 001375          BNE 2$
2815 014272 005077 165120    CLR @TXBUF     ;SHIP OUT CHAR.
2816 014276 012737 130000 014314 MOV #-50000,3$+2
2817 014304 105777 165100    5$: TSTB @RXCSR  ;TEST COMPLETE
2818 014310 100404          BMI 6$
2819 014312 005227 177770    3$: INC #-10     ;ALLOW TIME FOR PDR DONE TO SET
2820 014316 001372          BNE 5$
2821 014320 104003          ERROR        ;FAILURE OF RDR DONE TO SET
2822 014322 104012          6$: SCOPE
2823 014324          TSTA 10.,EX,0
2824 014324          TSTAA EX,10.,\X+1+0,\X+2,\X+1
2825          ;*****
2826 014324 000060          AT60: 60          ;TEST NUMBER
2827 014326 014374          AT61          ;ADDRESS OF NEXT TEST
2828 014330 000012          10.         ;ITERATION COUNT
2829 014332 014334          EXA         ;SCOPE ENTRY POINT
2830          000060          X=X+1
2831          ;*****
2832          ;TEST THAT WHEN RDR ENABLE IS SET THAT THE RXCSR DONE
2833          ;BIT IS CLEARED
2834 014334 000005          EXA: RESET      ;SET RCVR DONE
2835 014336 004737 014544    JSR PC,STRXD   ;SET ENABLE
2836 014342 005277 165042    INC @RXCSR    ;DONE SHOULD CLEAR
2837 014346 105777 165036    TSTB @RXCSR
2838 014352 100001          BPL 1$
2839 014354 104003          ERROR        ;DONE NOT CLEAR
2840 014356 012737 177770 014366 1$: MOV #-10,3$+2
2841 014364 005227 177770    3$: INC #-10     ;WAIT 100MIC. SEC. FOR XOR
2842 014370 001375          BNE 3$
2843 014372 104012          SCOPE
2844 014374          TSTA 3.,EXA,0
2845 014374          TSTAA EXA,3.,\X+1+0,\X+2,\X+1
2846          ;*****
2847 014374 000061          AT61: 61          ;TEST NUMBER
2848 014376 014430          AT62          ;ADDRESS OF NEXT TEST
2849 014400 000003          3.           ;ITERATION COUNT
2850 014402 014404          EXAA        ;SCOPE ENTRY POINT
2851          000061          X=X+1
2852          ;*****
2853 014404 005737 002110    EXAA: TST XORFLG ;CHECKING JUMPER CONNECTIONS FOR XOR, RCVR
2854 014410 100006          BPL 3$
2855 014412 012777 177777 164770 MOV #-1,@RXCSR
2856 014420 005777 164764    TST @RXCSR

```

N 4

```

2857 014424 000005
2858 014426 104012
2859 014430
2860 014430
2861
2862 014430 000062
2863 014432 014464
2864 014434 000003
2865 014436 014440
2866 000062
2867
2868 014440 005737 002110
2869 014444 100006
2870 014446 012777 177677 164740
2871 014454 005777 164734
2872 014460 000005
2873 014462 104012
2874 014464
2875
2876 014464 100063
2877 014466 177777
2878 014470 000012
2879 014472 014474
2880 000063
2881
2882
2883
2884 014474 004737 003004
2885 014500 052777 000004 164706
2886 014506 052777 000001 164700
2887 014514 012777 000252 164674
2888 014522 104017
2889 014524 104003
2890 014526 127727 164660 000000
2891 014534 001401
2892 014536 104003
2893 014540 104011
2894 014542 104012
2895
2896
2897
2898 014544 052777 000004 164642
2899 014552 005077 164640
2900 014556 104017
2901 014560 104003
2902 014562 000207
2903
2904 014564 023737 001566 001570
2905 014572 101430
2906 014574 023737 001570 001572
2907 014602 101424
2908 014604 023737 001572 001574
2909 014612 101420
2910 014614 023737 001574 001576
2911 014622 101414
2912 014624 023737 001576 001600

```

```

3$: RESET
SCOPE
TSTA 3.,EXB,0
TSTAA EXB,3.,\X+1+0,\X+2,\X+1
:*****
AT62: 62 ;TEST NUMBER
AT63 ;ADDRESS OF NEXT TEST
3. ;ITERATION COUNT
EXBA ;SCOPE ENTRY POINT
X=X+1
:*****
EXBA: TST XORFLG ;SAME AS ABOVE BUT FOR XMTR
BPL 4$
MOV #177677,@TXCSR
TST @TXCSR
4$: RESET
SCOPE
TSTAA AQB,10.,\X+1+CD, LAST,\X+1
:*****
AT63: 100063 ;TEST NUMBER
ATLAST ;ADDRESS OF NEXT TEST
10. ;ITERATION COUNT
AQBA ;SCOPE ENTRY POINT
X=X+1
:*****
:TEST THAT WHEN TXCSR BIT 0 IS SET THAT THE OUTPUT DATA LINE
:IS PULLED TO A SPACE.
AQBA: JSR %7,CDINIT ;INIT IF C-D DEVICE
BIS #BIT2,@TXCSR ;SET MAINTENANCE BIT IN TXCSR
BIS #BIT0,@TXCSR ;SET BREAK BIT
MOV #252,@TXBUF ;LOAD BUFFER
TIMERX ;TIME OUT RX DONE
ERROR ;ERROR DONE NOT SETTING
CMPB @RXBUF,#0 ;CHARACTER RECEIVED SHOULD BE 0
BEQ .+4
ERROR ;CHARACTER OTHER THAN 0
SRESET ;ISSUE RESET
SCOPE
:
:SUBROUTINE TO SET RXCSR DONE BIT.
STRXD: BIS #BIT2,@TXCSR ;SET MAINTENANCE BIT.
CLR @TXBUF ;LOAD TXBUF.
TIMERX ;TIME OUT TX DONE
ERROR ;ERROR DONE NOT SETTING
RTS %7 ;EXIT.
:SUBROUTINE TO CHECK THAT CTR0 THROUGH CTR3 CONTAIN DESCENDING VALUES.
CMPT: CMP CTR0,CTR1
BLOS CMPTNG
CMP CTR1,CTR2
BLOS CMPTNG
CMP CTR2,CTR3
BLOS CMPTNG
CMP CTR3,CTR4
BLOS CMPTNG
CMP CTR4,CTR5

```

```
2913 014632 101410          BLOS  CMTNG
2914 014634 023737 001600 001602    CMP  CTR5,CTR6
2915 014642 101404          BLOS  CMTNG
2916 014644 023737 001602 001604    CMP  CTR6,CTR7
2917 014652 101002          BHI  CMPTOK
2918 014654 062716 000002    CMTNG: ADD  #2,@%6
2919 014660 000207    CMPTOK: RTS  %7
2920
2921
2922
2923      ;*****
2924      ;PRG1 - TRANSMITTER SCOPE LOOP
2925      ;*****
2925 014662 104000    PRG1:  TYPE          ;TYPE PROGRAM TITLE.
2926 014664 016476          PITIT
2927 014666 004537 004262    JSR    5,LINSLX      ;GO GET LINE # FROM USER
2928 014672 104000          TYPE          ;TYPE SELECT CHAR AND DELAY.
2929 014674 016573          SELCAD
2930 014676 004737 003652    JSR    PC,RDOCT      ;READ IN DATA.
2931 014702 012637 001624          MOV    (SP)+,TEMP2    ;STORE DATA.
2932 014706 113737 001624 014730 PRG1A: MOVB  TEMP2,PRG B    ;DELAY COUNT TO PRG1B.
2933 014714 113777 001625 164474    MOVB  TEMP2+1,@TXBUF ;LOAD TXBUF.
2934 014722 105777 164466    TSTB  @TXCSR        ;TEST FOR DONE.          ;:++G
2935 014726 104016          DELAY          ;DELAY # OF MSECS. SET AT SR.
2936 014730 000000    PRG1B: OPEN
2937 014732 000765          BR    PRG1A        ;REPEAT.
2938
2939      ;*****
2940      ;PRG2 - RECEIVER SCOPE LOOP.
2941      ;*****
2941 014734 104000    PRG2:  TYPE          ;TYPE PROGRAM TITLE.
2942 014736 016536          P2TIT
2943 014740 004537 004262    JSR    5,LINSLX      ;GO GET LINE # FROM USER
2944 014744 104000          TYPE          ;TYPE SELECT CHAR AND DELAY.
2945 014746 016573          SELCAD
2946 014750 004737 003652    JSR    PC,RDOCT      ;READ IN DATA.
2947 014754 012637 001624          MOV    (SP)+,TEMP2    ;STORE DATA.
2948 014760 052777 000004 164426 PRG2A: BIS  #BIT2,@TXCSR ;SET MAINTENANCE BIT.
2949 014766 113737 001624 015010    MOVB  TEMP2,PRG2B    ;DELAY COUNT TO PRG2B.
2950 014774 113777 001625 164414    MOVB  TEMP2+1,@TXBUF ;LOAD TXBUF.
2951 015002 105777 164406    TSTB  @TXCSR        ;TEST FOR DONE          ;:++G
2952 015006 104016          DELAY          ;DELAY # OF MSECS. SET IN SR.
2953 015010 000000    PRG2B: OPEN
2954 015012 017700 164374    MOV    @RXBUF,%0     ;RXBUF CONTENTS TO R0.
2955 015016 000005          RESET          ;DISPLAY CONTENTS OF RXBUF (IN R0).
2956 015020 000005          RESET          ;BY ISSUING 5 RESET INSTRUCTIONS
2957 015022 000005          RESET
2958 015024 000005          RESET
2959 015026 000005          RESET
2960 015030 000753          BR    PRG2A
2961
2962
2963      ;*****
2964      ;PRG3 - SINGLE CHARACTER MAINTENANCE MODE DATA TEST.
2965      ;*****
2966 015032 104000    PRG3:  TYPE          ;TYPE PROGRAM TITLE.
2967 015034 017364          P3TIT
2968 015036 004537 004262    JSR    5,LINSLX      ;GO GET LINE # FROM USER
```

```
2969 015042 104000          TYPE                ;TYPE: SELECT CHARACTER.
2970 015044 017507          SELCAR
2971 015046 004737 003652    JSR      PC,RDOCT      ;GET TEST CHAR AND DELAY FROM USER.
2972 015052 012637 001624    MOV      (SP)+,TEMP2   ;STORE TEST CHAR AND DELAY
2973 015056 113737 001625 001562 PRG3A: MOVB  TEMP2+1,CRBUFA ;MOVE DATA CHAR TO CRBUFA.
2974 015064 004737 015124    JSR      %7,MOUTIN     ;GO OUTPUT, RECEIVE, AND CHECK DATA.
2975 015070 000772          BR      PRG3A
2976
2977          ;*****
2978          ;PRG4 - SPECIAL BINARY COUNT MAINTENANCE MODE DATA TEST.
2979          ;*****
2979 015072 104000          PRG4.  TYPE                ;TYPE PROGRAM TITLE.
2980 015074 017434          P4TIT
2981 015076 004537 004262    JSR      5,LINSLX      ;GO GET LINE # FROM JSER
2982 015102 004737 004452    JSR      %7,INBIN      ;INITIALIZE BINARY COUNT.
2983 015106 004737 004556    PRG4A: JSR      %7,GTBINP ;GET BINARY CHARACTER.
2984 015112 110137 001562    MOVB    %1,CRBUFA      ;SAVE AT CRBUFA.
2985 015116 004737 015124    JSR      %7,MOUTIN     ;GO OUTPUT, RECEIVE, AND CHECK DATA.
2986 015122 000771          BR      PRG4A          ;REPEAT.
2987          ;SUBROUTINE TO OUTPUT, RECEIVE, AND CHECK DATA WITH MAINTENANCE BIT SET.
2988 015124 032777 000200 163036 MOUTIN: BIT  #BIT7,@SRPTR ;SEE IF BIT 7 IS SET.
2989 015132 001001          BNE     .+4            ;BRANCH IF SET.
2990 015134 104002          STALL                ;SET. DO A RANDOM STALL.
2991 015136 104020          TIMETX              ;TIME OUT TX DONE
2992 015140 104003          ERROR              ;ERROR DONE NOT SETTING
2993 015142 052777 000004 164244 BIS      #BIT2,@TXCSR   ;SET MAINTENANCE BIT.
2994 015150 005777 164236    TST     @RXBUF         ;CLR RX DONE
2995 015154 013777 001562 164234 MOV      CRBUFA,@TXBUF  ;LOAD TXBUF.
2996 015162 004737 005530    JSR      7,MASKIT     ;MASK OFF NON TRANSMITTED BITS
2997 015166 104017          TIMERX              ;TIME OUT RX DONE
2998 015170 104003          ERROR              ;ERROR DONE NOT SETTING
2999 015172 017737 164214 001560 MOV      @RXBUF,CRBUF   ;MOVE CHAR IN RX BUFFER TO CRBUF.
3000 015200 104004          DATCHK              ;COMPARE EXPECTED AND RECEIVED DATA
3001 015202 000207          RTS      %7          ;EXIT.
3002
3003          ;*****
3004          ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ^G TO ALLOW CHANGING
3005          ;OF LOC 176. ROUTINE IS ENTERED AT CNTLU FOR START UP PURPOSES.
3006          ;WHEN A ^G IS GIVEN, THE PROGRAM ENTERS AT CKSWR. THE PROGRAM
3007          ;GETS CONTENTS OF SOFT. SWITCH REG. TYPES IT OUT, AND THEN SEEKS
3008          ;NEW DATA FROM OPERATOR. ONCE DATA IS SUPPLIED, IT INSERTS THIS
3009          ;DATA INTO THE SOFT.SWITCH REG. AND RESUMES OPERATION IN THE
3010          ;MAIN PROGRAM. INCORRECT ENTRIES (SUCH AS 8,9,LETTERS) ARE DELETED.
3011          ;AND THE PROCESS RESTARTED. TYPING ^U ALLOWS THE PRESENT
3012          ;ENTRY TO BE DELETED AND THE PROCESS RESTARTED.
3013          ;*****
3014
3015
3016
3017
3018 015204          CHG5:
3019 015204 022737 000176 000170 CKSWR: CMP      #SWREG,SRPTR ;SOFTWARE SW REG PRES?
3020 015212 001132          BNE     OUT           ;NO, GET OUT
3021 015214 105777 164210    TSTB   @TKS           ;YES, IS CHARACTER READY?
3022 015220 100127          BPL     OUT           ;IF NOT, GET OUT
3023 015222 017737 164204 001642 MOV      @TKB,11B      ;STORE BUFFER
3024 015230 042737 177600 001642 BIC     #177600,11B    ;STRIP OFF GARBAGE
```



```
3081
3082
3083
3084 015520 112737 000001 001650 TYP0C:  MOVB #1,FILL ;SET THE ZERO FILL SWITCH
3085 015526 112737 000006 001653          MOVB #6,MODE+1 ;SET FOR SIX(6) DIGITS
3086 015534 112737 000005 001654 TYP0N:  MOVB #5,CNT ;SET THE ITERATION COUNT
3087 015542 010346          MOV R3,-(SP) ;SAVE R3
3088 015544 010446          MOV R4,-(SP) ;SAVE R4
3089 015546 010546          MOV R5,-(SP) ;SAVE R5
3090 015550 113704 001653          MOVB MODE+1,R4 ;GET THE NUMBER OF DIGITS TO TYPE
3091 015554 005404          NEG R4
3092 015556 062704 000006          ADD #6,R4 ;SUBTRACT IT FOR MAX. ALLOWED
3093 015562 110437 001652          MOVB R4,MODE ;SAVE IT FOR USE
3094 015566 113704 001650          MOVB FILL,R4 ;GET THE ZERO FILL SWITCH
3095 015572 016605 000010          MOV 10(SP),R5 ;PICKUP THE INPUT NUMBER
3096 015576 005003          CLR R3 ;CLEAR THE OUTPUT WORD
3097 015600 006105          1$: ROL R5 ;ROTATE MSB INTO 'C'
3098 015602 000404          BR 3$ ;GO DO MSB
3099 015604 006105          2$: ROL R5 ;FORM THIS DIGIT
3100 015606 006105          ROL R5
3101 015610 006105          ROL R5
3102 015612 010503          MOV R5,R3
3103 015614 006103          3$: ROL R3 ;GET LSB OF THIS DIGIT
3104 015616 105337 001652          DECB MODE ;TYPE THIS DIGIT?
3105 015622 100020          BPL 7$ ;BR IF NO
3106 015624 042703 177770          BIC #177770,R3 ;GET RID OF JUNK
3107 015630 001002          BNE 4$ ;TEST FOR 0
3108 015632 005704          TST R4 ;SUPPRESS THIS 0
3109 015634 001403          BEQ 5$ ;BR IF YES
3110 015636 005204          4$: INC R4 ;DON'T SUPPRESS ANYMORE 0'S
3111 015640 052703 000060          BIS #60,R3 ;MAKE THIS DIGIT ASCII
3112 015644 105777 163564          5$: TSTB @TPS ;IS PRINTER READY FOR CHARACTER?
3113 015650 100375          BPL 5$ ;IF NOT, TRY AGAIN
3114 015652 110377 163560          MOVB R3,@TPB ;TYPE OUT NUMBER
3115 015656 105777 163552          8$: TSTB @TPS ;MAKE SURE LAST DIGIT TYPES
3116 015662 100375          BPL 8$
3117 015664 105337 001654          7$: DECB CNT ;COUNT BY 1
3118 015670 003345          BGT 2$ ;BR IF MORE TO DO
3119 015672 002402          BLT 6$ ;BR IF DONE
3120 015674 005204          INC R4 ;INSURE LAST DIGIT ISN'T A BLANK
3121 015676 000742          BR 2$ ;GO DO THE LAST DIGIT
3122 015700 012605          6$: MOV (SP)+,R5 ;RESTORE R5
3123 015702 012604          MOV (SP)+,R4 ;RESTORE R4
3124 015704 012603          MOV (SP)+,R3 ;RESTORE R3
3125 015706 000207          RTS PC ;RETURN FROM INTERRUPT PC
3126
3127 ;*****
3128
3129 ;ASCII MESSAGES
3130
3131 015710 042045 030514 026461 MTIT: .ASCII '%DL11-E,C/D OFLNE TST - CZDLA-M%'
3132 015716 026105 027503 020104
3133 015724 043117 047114 020105
3134 015732 051524 020124 020055
3135 015740 055103 046104 026501
3136 015746 022510
```

3137	015750	046445	050101	047440	.ASCII	'%MAP OF DEVICES PRESENT%'
3138	015756	020106	042504	044526		
3139	015764	042503	020123	051120		
3140	015772	051505	047105	022524		
3141	016000	100				
3142	016001	040	020040	020040	MDEVAD: .ASCII	' %'
3143	016006	020040	020040	020040		
3144	016014	020040	022440	100		
3145	016021	045	054524	042520	PGMSG: .ASCII	'%TYPE IN PROGRAM NUMBER @'
3146	016026	044440	020116	051120		
3147	016034	043517	040522	020115		
3148	016042	052516	041115	051105		
3149	016050	020040	020040	100		
3150	016055	045	047516	042516	MNONE: .ASCII	'%NONE FOUND%'
3151	016062	043040	052517	042116		
3152	016070	040045				
3153	016072	052045			EMO: .ASCII	'%T'
3154	016074	020040	020040	041520	ATNUMB: .ASCII	' PC= '
3155	016102	020075				
3156	016104	020040	020040	020040	APC: .ASCII	' RXCSR= '
3157	016112	020040	054122	051503		
3158	016120	036522	040			
3159	016123	040	020040	020040	MRXNUM: .ASCII	' @'
3160	016130	020040	040040			
3161	016134	022445	051120	030107	POTIT: .ASCII	'%PRGO - INPUT-OUTPUT LOGIC TESTS. '
3162	016142	026440	044440	050116		
3163	016150	052125	047455	052125		
3164	016156	052520	020124	047514		
3165	016164	044507	020103	042524		
3166	016172	052123	027123	040		
3167	016177	045	044504	041523	.ASCII	'%DISCONNECT DL11-E FROM MODEM'
3168	016204	047117	042516	052103		
3169	016212	042040	030514	026461		
3170	016220	020105	051106	046517		
3171	016226	046440	042117	046505		
3172	016234	040440	042116	041440	.ASCII	' AND CONNECT JUMPER TO CABLE.%'
3173	016242	047117	042516	052103		
3174	01625^	045040	046525	042520		
3175	016256	020122	047524	041440		
3176	016264	041101	042514	022456		
3177	016272	100				
3178	016273	124	041530	051123	ATXCSR: .ASCII	'TXCSR S/B: '
3179	016300	051440	041057	020072		
3180	016306	020040	020040	020040	ATXSB: .ASCII	' WAS: '
3181	016314	020040	040527	035123		
3182	016322	040				
3183	016323	040	020040	020040	ATXWAS: .ASCII	' @'
3184	016330	040040				
3185	016332	054122	051503	020122	ARXCSR: .ASCII	'RXCSR S/B: '
3186	016340	027523	035102	040		
3187	016345	040	020040	020040	ARXSB: .ASCII	' WAS: '
3188	016352	020040	053440	051501		
3189	016360	020072				
3190	016362	020040	020040	020040	ARXWAS: .ASCII	' @'
3191	016370	040040				
3192	016372	054124	051440	042520	ETXTIM: .ASCII	'TX SPEEDS NOT IN ASCENDING ORDER.%'

3193	016400	042105	020123	047516	
3194	016406	020124	047111	040440	
3195	016414	041523	047105	044504	
3196	016422	043516	047440	042122	
3197	016430	051105	040056		
3198	016434	054122	051440	042520	ERXTIM: .ASCII 'RX SPEEDS NOT IN ASCENDING ORDER.@"
3199	016442	042105	020123	047516	
3200	016450	020124	047111	040440	
3201	016456	041523	047105	044504	
3202	016464	043516	047440	042122	
3203	016472	051105	040056		
3204	016476	022445	051120	030507	P1TIT: .ASCII '%PRG1 - TRANSMITTER SCOPE LOOP@"
3205	016504	026440	052040	040522	
3206	016512	051516	044515	052124	
3207	016520	051105	051440	047503	
3208	016526	042520	046040	047517	
3209	016534	040120			
3210	016536	022445	051120	031107	P2TIT: .ASCII '%PRG2 - RECEIVER SCOPE LOOP@"
3211	016544	026440	051040	041505	
3212	016552	044505	042526	020122	
3213	016560	041523	050117	020105	
3214	016566	047514	050117	100	
3215	016573	045	054524	042520	SELCAD: .ASCII '%TYPE TEST CHAR. CODE IN BITS 15-8,TYPE DELAY TIME IN BITS 7-0'
3216	016600	052040	051505	020124	
3217	016606	044103	051101	020056	
3218	016614	047503	042504	044440	
3219	016622	020116	044502	051524	
3220	016630	030440	026465	026070	
3221	016636	054524	042520	042040	
3222	016644	046105	054501	052040	
3223	016652	046511	020105	047111	
3224	016660	041040	052111	020123	
3225	016666	026467	060		
3226	016671	045	043117	040440	.ASCII '%OF AN OCTAL WORD.@"
3227	016676	020116	041517	040524	
3228	016704	020114	047527	042122	
3229	016712	022456	100		
3230	016715	104	052101	020101	ERDAT: .ASCII 'DATA S/B: '
3231	016722	027523	035102	040	
3232	016727	040	020040	053440	AASR: .ASCII ' WAS: '
3233	016734	051501	020072		
3234	016740	020040	040		AWAS: .ASCII ' ' ' .ASCII ' RXBUF: '
3235	016743	040	054122	052502	
3236	016750	035106	040		
3237	016753	040	020040	020040	ARXBUF: .ASCII ' @'
3238	016760	020040	100		
3239	016763	045	042523	020124	ASETSR: .ASCII '%SET HARDWARE SR OPTIONS. NORMAL OPERATION '
3240	016770	040510	042122	040527	
3241	016776	042522	051440	020122	
3242	017004	050117	044524	047117	
3243	017012	027123	047040	051117	
3244	017020	040515	020114	050117	
3245	017026	051105	052101	047511	
3246	017034	020116			
3247	017036	051511	053440	052111	.ASCII 'IS WITH SR=00000. IF NONE, TYPE <CR>@"
3248	017044	020110	051123	030075	

3249	017052	030060	030060	027060		
3250	017060	044440	020106	047516		
3251	017066	042516	020054	054524		
3252	017074	042520	036040	051103		
3253	017102	022476	100			
3254	017105	045	047111	047503	AINCRT: .ASCII	'%INCORRECT ROUTINE SELECTED, PLACE CORRECT PROGRAM'
3255	017112	051122	041505	020124		
3256	017120	047522	052125	047111		
3257	017126	020105	042523	042514		
3258	017134	052103	042105	020054		
3259	017142	046120	041501	020105		
3260	017150	047503	051122	041505		
3261	017156	020124	051120	043517		
3262	017164	040522	115			
3263	017167	045	047111	051440	.ASCII	'%IN SR 0-2 AND PRESS CONTINUE.@'
3264	017174	020122	026460	020062		
3265	017202	047101	020104	051120		
3266	017210	051505	020123	047503		
3267	017216	052116	047111	042525		
3268	017224	040056				
3269	017226	044445	053116	046101	AINCPG: ASCII	'%INVALID PROGRAM SELECTED.@'
3270	017234	042111	050040	047522		
3271	017242	051107	046501	051440		
3272	017250	046105	041505	042524		
3273	017256	027104	100			
3274	017261	207			APGEND: .BYTE	207
3275	017262	042445	042116	050040	.ASCII	'%END PASS = '
3276	017270	051501	020123	020075		
3277	017276	020040	020040	020040	APCNT: .ASCII	' LINE = '
3278	017304	020040	044514	042516		
3279	017312	036440	020040			
3280	017316	020040	020040	054122	ACLIN: .ASCII	' RXCSR = '
3281	017324	051503	020122	020075		
3282	017332	020040	020040	020040	APRXC: .AS	' VECTOR = '
3283	017340	020040	042526	052103		
3284	017346	051117	036440	040		
3285	017353	040	020040	020040	APVEC: .ASCII	' @'
3286	017360	020040	040040			
3287	017364	022445	051120	031507	P3TIT: .ASCII	'%PRG3-SINGLE CHAR MAINT MODE DATA TEST@'
3288	017372	051455	047111	046107		
3289	017400	020105	044103	051101		
3290	017406	046440	044501	052116		
3291	017414	046440	042117	020105		
3292	017422	040504	040524	052040		
3293	017430	051505	040124			
3294	017434	022445	051120	032107	P4TIT: .ASCII	'%PRG4-SPEC BIN COUNT MAINT MODE DATA TEST@'
3295	017442	051455	042520	020103		
3296	017450	044502	020116	047503		
3297	017456	047125	020124	040515		
3298	017464	047111	020124	047515		
3299	017472	042504	042040	052101		
3300	017500	020101	042524	052123		
3301	017506	100				
3302	017507	045	054524	042520	SELCAR: .ASCII	'%TYPE IN TEST CHAR. CODE. @'
3303	017514	044440	020116	042524		
3304	017522	052123	041440	040510		

3305	017530	027122	041440	042117	
3306	017536	027 05	020040	020040	
3307	017544	040045			
3308	017546	052045	050131	020105	LDLINE: .ASCII '%TYPE IN LINE NO. @'
3309	017554	047111	046040	047111	
3310	017562	020105	047516	020056	
3311	017570	020040	040040		
3312	017574	046045	047111	020105	ALINE: .ASCII '%LINE NO.'
3313	017602	047516	056		
3314	017605	040	020040	040527	SELINE: .ASCII ' WAS SELECTED@'
3315	017612	020123	042523	042514	
3316	017620	052103	042105	100	
3317	017625	045	042522	042503	MSETRX: .ASCII '%RECEIVER SPEED CHECK@'
3318	017632	053111	051105	051440	
3319	017640	042520	042105	041440	
3320	017646	042510	045503	100	
3321	017653	045	051124	047101	MSETTX: .ASCII '%TRANSMIT SPEED CHECK@'
3322	017660	046523	052111	051440	
3323	017666	042520	042105	041440	
3324	017674	042510	045503	100	
3325	017701	045	042523	020124	MSETC: .ASCII '%SET CLOCK SWITCHES TO POSITION, THEN PRESS CONTINUE.@'
3326	017706	046103	041517	020113	
3327	017714	053523	052111	044103	
3328	017722	051505	052040	020117	
3329	017730	047520	044523	044524	
3330	017736	047117	020054	044124	
3331	017744	047105	050040	042522	
3332	017752	051523	041440	047117	
3333	017760	044524	052516	027105	
3334	017766	100			
3335	017767	045	051105	047522	MTERR: .ASCII '%ERROR - UNEXPECTED TRAP'
3336	017774	020122	020055	047125	
3337	020002	054105	042520	052103	
3338	020010	042105	052040	040522	
3339	020016	120			
3340	020017	045	051124	050101	.ASCII '%TRAPPED TO '
3341	020024	042520	020104	047524	
3342	020032	020040			
3343	020034	020040	020040	020040	MTO: .ASCII ' ' '
3344	020042	020040			
3345	020044	052045	040522	050120	.ASCII '%TRAPPED FROM PC '
3346	020052	042105	043040	047522	
3347	020060	020115	041520	020040	
3348	020066	020040	020040	020040	MFROM: .ASCII ' @'
3349	020074	020040	100		
3350	020077	045	047516	042040	MNOLIN: .ASCII '%NO DEVICE PRESENT - THIS LINE NO.@'
3351	020104	053105	041511	020105	
3352	020112	051120	051505	047105	
3353	020120	020124	020055	044124	
3354	020126	051511	046040	047111	
3355	020134	020105	047516	040056	
3356	020142	022477	100		DTERR: .ASCII '?%@'
3357	020145	045	047516	044440	INTER: .ASCII '%NO INTERRUPT@'
3358	020152	052116	051105	052522	
3359	020160	052120	100		
3360	020163	045	051503	036440	MSO: .ASCII '%CS = 0@'

3361	020170	030040	100						
3362	020173	045	051503	036440	MS1:	.ASCII	'%CS = 1@'		
3363	020200	030440	100						
3364	020203	045	051503	036440	MS2:	.ASCII	'%CS = 2@'		
3365	020210	031040	100						
3366	020213	045	051503	036440	MS3:	.ASCII	'%CS = 3@'		
3367	020220	031440	100						
3368	020223	045	051503	036440	MS4:	.ASCII	'%CS = 4@'		
3369	020230	032040	100						
3370	020233	045	051503	036440	MS5:	.ASCII	'%CS = 5@'		
3371	020240	032440	100						
3372	020243	045	051503	036440	MS6:	.ASCII	'%CS = 6@'		
3373	020250	033040	100						
3374	020253	045	051503	036440	MS7:	.ASCII	'%CS = 7@'		
3375	020260	033440	100						
3376	020263	045	042522	047503	MPWRF:	.ASCII	'%RECOVERED FROM POWER FAILURE@'		
3377	020270	042526	042522	020104					
3378	020276	051106	046517	050040					
3379	020304	053517	051105	043040					
3380	020312	044501	052514	042522					
3381	020320	100							
3382									
3383	020321	040	057040	020107	:***** CHGM6 *****				
3384	020326	020100			CNTG: .ASCII / ^G @ /				
3385	020330	020040	053523	036522	SWR: .ASCII / SWR= @/				
3386	020336	020040	100						
3387	020341	040	047040	053505	NEW: .ASCII / NEW= @/				
3388	020346	020075	040040						
3389	020352	020040	020077	040040	QUEST: .ASCII / ? @/				
3390					:*****				
3391					.EVEN				
3392	000001				.END				

AAA	006654	1565	1569#							
AAAA	010212	1914	1918#							
AAAB	010226	1919	1922#							
AAB	006666	1571#	1574							
AABA	013716	2697	2703#							
AABB	013742	2709#	2715	2718						
AABC	013752	2704	2711#							
AABD	013766	2712	2714#							
AABE	013772	2711	2716#							
AAE	006670	1569	1572#							
AASB	016727	1388	3232#							
ABA	006706	1581	1585#							
ABAA	011270	2144	2148#							
ABAB	011362	2152	2165#							
ABB	006726	1587	1589#	1592						
ABBA	014010	2725	2729#							
ABBB	014032	2732	2734#							
ABE	006730	1585	1590#							
ACA	006746	1599	1603#							
ACAA	011374	2172	2177#							
ACAB	011524	2190	2202#							
ACB	006760	1605#	1608							
ACBA	014054	2742	2748#							
ACBB	014112	2747	2756#							
ACBC	014114	2755	2757#							
ACE	006762	1603	1606#							
ACLIN	017316	1320	3280#							
ADA	007000	1615	1619#							
ADAA	011544	2210	2215#							
ADAB	011572	2220	2223#							
ADAC	011610	2222	2228#							
ADB	007012	1621#	1624							
ADDA	014140	2765	2770#							
ADE	007014	1619	1622#							
ADTENP	005100	1247	1273#							
AEA	007032	1631	1636#							
AEB	007046	1637	1640#							
AEC	007070	1642	1645#							
AED	007110	1639	1644	1647	1649#					
AFBA	010240	1935	1940#							
AFBB	010266	1943	1946#							
AFBC	010312	1945	1949	1951#						
AGA	007132	1658	1662#							
AGB	007146	1663	1666#							
AGBA	010324	1958	1963#							
AGBB	010360	1967	1970#							
AGBC	010414	1976	1979#							
AGBD	010430	1981	1984#							
AGBE	010464	1969	1978	1983	1990	1992	1993#			
AGC	007170	1668	1671#							
AGD	007210	1665	1670	1673	1675#					
AJAA	011624	2236	2241#							
AJAF	012050	2286	2289#							
AJAS	012052	2248	2253	2258	2263	2268	2273	2278	2283	2291#
AJASA	012102	2298#	2301							
AJAST	012134	2249	2254	2259	2264	2269	2274	2279	2284	2291• 2299• 2308#

AINCPG	017226	1308	3269#							
AINCRT	017105	1304	3254#							
AJA	007232	1684	1688#							
AJB	007254	1690	1693#							
AJBA	010476	2000	2005#							
AJBB	010524	2008	2011#							
AJBC	010552	2014	2017#							
AJBD	010576	2010	2016	2020	2022#					
AJC	007276	1695	1698#							
AJD	007316	1692	1697	1700	1702#					
AKA	007340	1711	1716#							
AKB	007352	1717	1720#							
AKBA	010610	2029	2034#							
AKBC	010646	2039	2041#							
ALA	007364	1727	1731#							
ALAA	012146	2315	2322#							
ALAB	012176	2327	2330#							
ALAC	012210	2329	2332	2334#						
ALBA	011016	2081	2086#							
ALBB	011044	2089	2092#							
ALBC	011072	2095	2098#							
ALBD	011116	2091	2097	2101	2103#					
ALINE	017574	1173	3312#							
ALY	007414	1734	1737#							
ALZ	007434	1736	1739	1741#						
AMAA	012222	2341	2346#							
AMAC	012254	2352	2354#							
AMBA	011130	2110	2114#							
AMBB	011170	2119	2122#							
AMBC	011204	2123	2126#							
AMBD	011232	2129	2132#							
AMBE	011256	2121	2125	2131	2135	2137#				
AOAA	012272	2361	2367#							
AOAB	012310	2370#	2374							
AOAB1	012324	2371	2373#							
AOAC	012346	2376	2379#							
AOAD	012360	2378	2380	2382#						
AOBA	010704	2053	2057#							
AOBB	010732	2060	2063#							
AOBC	010760	2066	2069#							
AOBD	011004	2062	2068	2072	2074#					
APA	007446	1748	1752#							
APB	007462	1753	1756#							
APBA	014162	2778	2784#							
APBB	014170	2785#	2796							
APC	016104	1415	3156#							
APCNT	017276	1316	3277#							
APCX	007504	1758	1761#							
APD	007524	1755	1760	1763	1765#					
APGEND	017261	1331	3274#							
APRXC	017332	1324	3282#							
APVEC	017353	1328	3285#							
AQA	007546	1774	1778#							
AQAA	012376	2390	2398#							
AQAB	012622	2443	2446#							
AQAS	012624	2405	2410	2415	2420	2425	2430	2435	2440	2448#

AT35	012136	2234	2312#
AT36	012212	2313	2338#
AT37	012256	2339	2358#
AT4	007022	1613	1628#
AT40	012366	2359	2387#
AT41	012670	2388	2462#
AT42	013034	2463	2502#
AT43	013122	2503	2526#
AT44	013200	2527	2548#
AT45	013264	2549	2571#
AT46	013362	2572	2599#
AT47	013446	2600	2623#
AT5	007122	1629	1655#
AT50	013530	2624	2647#
AT51	013612	2648	2670#
AT52	013702	2671	2694#
AT53	014000	2695	2722#
AT54	014040	2723	2739#
AT55	014124	2740	2762#
AT56	014146	2763	2775#
AT57	014240	2776	2801#
AT6	007222	1656	1681#
AT60	014324	2802	2826#
AT61	014374	2827	2847#
AT62	014430	2848	2862#
AT63	014464	2863	2876#
AT7	007330	1682	1708#
AUAA	013136	2529	2536#
AUAB	013162	2540#	2544
AUAC	013172	2535	2542#
AVAA	013214	2551	2558#
AVAB	013252	2557	2565#
AVAC	013254	2564	2566#
AWAA	013274	2574	2579#
AWAB	013324	2586#	2592
AWAC	013334	2580	2588#
AWAD	013350	2589	2591#
AWAE	013354	2588	2593#
AWAS	016740	1384	3234#
AXA	010110	1869	1873#
AXAA	013406	2602	2611#
AXAB	013434	2609	2617#
AXAC	013436	2616	2618#
AXB	010124	1874	1877#
AX1A	013466	2626	2634#
AX1B	013516	2633	2641#
AX1C	013520	2640	2642#
AYA	010136	1884	1888#
AYAA	013550	2650	2658#
AYAB	013574	2662#	2666
AYAC	013604	2656	2664#
AYB	010152	1889	1892#
AZA	010164	1899	1903#
AZAA	013632	2673	2681#
AZAB	013670	2679	2688#
AZAC	013672	2687	2689#

2595

CMAS12	001326	568#								
CMAS13	001330	569#								
CMAS14	001332	570#								
CMAS15	001334	571#								
CMAS16	001336	572#								
CMAS17	001340	573#								
CMAS2	001306	560#								
CMAS20	001342	574#								
CMAS21	001344	575#								
CMAS22	001346	576#								
CMAS23	001350	577#								
CMAS24	001352	578#								
CMAS25	001354	579#								
CMAS26	001356	580#								
CMAS27	001360	581#								
CMAS3	001310	561#								
CMAS30	001362	582#								
CMAS31	001364	583#								
CMAS32	001366	584#								
CMAS33	001370	585#								
CMAS34	001372	586#								
CMAS35	001374	587#								
CMAS36	001376	588#								
CMAS37	001400	589#								
CMAS4	001312	562#								
CMAS5	001314	563#								
CMAS6	001316	564#								
CMAS7	001320	565#								
CMPT	014564	2285	2442	2904#						
CMPTNG	014654	2905	2907	2909	2911	2913	2915	2918#		
CMPTOK	014660	2917	2919#							
CNT	001654	681#	3086*	3117*						
CNTG	020321	3028	3383#							
CNTLU	015256	703	3030#	3048						
CNVCTR	005072	1248*	1251*	1270#						
CONT	005456	1347	1353#							
COUNT	001626	669#	1039*	1055*	1056					
COUNT1	001646	678#	3038*	3052	3066*					
CRBUF	001560	650#	1291*	1380	1383	2793*	2999*			
CRBUFA	001562	651#	1286*	1373*	1380	1387	2788*	2973*	2984*	2995
CRBUFB	001564	652#	1377*	1378	1391					
CTRO	001566	653#	1282*	1293*	2249*	2406*	2784*	2795*	2904	
CTR1	001570	654#	2254*	2411*	2904	2906				
CTR2	001572	655#	2259*	2416*	2906	2908				
CTR3	001574	656#	2264*	2421*	2908	2910				
CTR4	001576	657#	2269*	2426*	2910	2912				
CTR5	001600	658#	2274*	2431*	2912	2914				
CTR6	001602	659#	2279*	2436*	2914	2916				
CTR7	001604	660#	2284*	2441*	2916					
CURTST	001454	615#	830	836	891*					
DATAA	005142	1283#	1294							
DATCHK=	104004	477#	1292	2794	3000					
DATTST	005120	1280#	2770							
DECVAL	005112	1243	1278#							
DELAY -	104016	487#	880	1096	2324	2496	2637	2935	2952	
DIGIT	005074	1261*	1264*	1267*	1268	1271#				

RXCR36	001274	547#																
RXCR4	001210	521#																
RXCR5	001212	522#																
RXCR6	001214	523#																
RXCR7	001216	524#																
RXCSR	001410	595#	978*	1110	1280*	1289	1323	1418	1471*	1570*	1731*	1732*	1733	1737*				
		1738	1752	1756*	1757	1761*	1762	1765*	1778	1782*	1783	1787*	1788	1791*				
		1805	1809*	1810	1814*	1815	1818*	1832	1836*	1837	1841*	1842	1845*	1858				
		1873	1888	1903	1918	1940*	1942	1946*	1948	1963*	1965	1966	1970*	1974				
		1979	1984*	1988	2005*	2007	2011*	2013	2017*	2019	2034*	2036*	2038	2041*				
		2043	2057*	2059	2063*	2065	2069*	207	2086*	2088	2092*	2094	2098*	2100				
		2114*	2116*	2118	2122	2126*	2128	2132*	2134	2178*	2179*	2188	2202*	2326				
		2331	2351	2370	2373	2379	2455	2611*	2613*	2618*	2634*	2636*	2642*	2658*				
		2660*	2662*	2681*	2684*	2689*	2705*	2706*	2709*	2730	2731	2750*	2752*	2757*				
		2811*	2817	2836*	2837	2855*	2856											
RXCSR1	001610	662#	1965*	1974*	1975	1988*	1989	2188*	2189	2197	2730*							
RXEND	001300	549#	762	794														
RXLVL	001422	600#	939	982*	1486*	2659	2682											
RXVTR	001420	599#	937	980*	1164	1327	1466*	1511*	1518	2711*								
SAVREG=	104013	484#	1211	1231	1242													
SAVRG	003116	642	905#															
SCOPE -	104012	483#	1296	1571	1589	1605	1621	1651	1677	1704	1720	1741	1767	1793				
		1820	1847	1862	1877	1892	1907	1922	1951	1993	2022	2046	2074	2103				
		2137	2165	2203	2229	2289	2334	2354	2383	2446	2475	2479	2484	2489				
		2493	2519	2541	2567	2587	2619	2643	2663	2690	2710	2735	2758	2771				
		2797	2822	2843	2858	2873	2894											
SCOPTR	001466	620#	846	890*														
SCRLF	104021	490#	3051															
SELCAD	016573	2929	2945	3215#														
SELCAR	017507	2970	3302#															
SELINE	017605	1170	3314#															
SETSR	005220	1298#	1555															
SIOT	004210	1110*	1112*	1116	1121#													
SPBOT -	001176	440#	688	748	820	1501												
SRESET	104011	482#	821	1650	1676	1703	1719	1766	1792	1819	1846	1861	1876	1891				
		1906	1921	2150	2187	2228	2330	2893										
SRPTR	000170	502#	693	697*	701	824	832	844	854	859	864	1137	1312	1332				
		1353	1364	1407	1492	1520	2988	3019	3032	3069*								
SRSETT	003306	640	952#	954*														
STAL	004072	633	1092#															
STALA	004112	1095*	1097#															
STALB	004114	1094	1098#															
STALL -	104002	475#	2990															
START	002136	719	732	748#	803	1310	1455											
STARTZ	001656	507	688#															
START1	002326	755	784#															
START2	002332	785#	795															
START3	002366	786	793#															
START4	002426	792	804#															
STLMSK	001406	593#	1093															
STLSPV	003252	638	943#															
STLSRV	003216	637	934#															
STPPA	003274	944*	947#															
STPRA	003240	935*	938#															
STRXD	014544	2610	2657	2680	2702	2729	2748	2749	2835	2898#								
STRXV	104006	479#	2608	2632	2655	2678	2703	2746										

CROSS REFERENCE TABLE -- MACRO NAMES

TSTA	744#	1559	1575	1593	1609	1625	1652	1678	1705	1721	1742	1768	1794	1821	1848
	1863	1878	1893	1908	1929	1952	1994	2023	2047	2075	2104	2138	2166	2204	2230
	2309	2335	2355	2384	2459	2499	2523	2545	2568	2596	2620	2644	2667	2691	2719
	2736	2759	2772	2798	2823	2844	2859								
TSTAA	735#	1560	1576	1594	1610	1626	1653	1679	1706	1722	1743	1769	1795	1822	1849
	1864	1879	1894	1909	1930	1953	1995	2024	2048	2076	2105	2139	2167	2205	2231
	2310	2336	2356	2385	2460	2500	2524	2546	2569	2597	2621	2645	2668	2692	2720
	2737	2760	2773	2799	2824	2845	2860	2874							

. ABS. 020360 000

ERRORS DETECTED: 0

CZDLAH.BIN,CZDLAH.LST/CRF/SOL/NL:TOC=CZDLAH.P11

RUN-TIME: 11 23 4 SECONDS

RUN-TIME RATIO: 122/39=3.0

CORE USED: 11K (21 PAGES)