

.REM @
.REPT 0

IDENTIFICATION

PRODUCT CODE: AC-8516G-MC
PRODUCT NAME: CZDLAGO DL11-E,C/D OFLNE TST
DATE: JUNE 1978
MAINTAINER: DIAGNOSTIC GROUP
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.

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

COPYRIGHT (C) 1972,1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

1. ABSTRACT

TWO SEPARATE DIAGNOSTIC PROGRAMS ARE PROVIDED FOR THE DL11-E (ASYNCHRONOUS 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. PAGE 2

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.

IT'S THE RESPONSIBILITY OF THE OPERATOR TO SET UP THIS LOCATION PRIOR TO STARTING THE PROGRAM.

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 CONNECTOR IN THE MODEM END OF THE CABLE, AND PRESS CONTINUE.
NOTE, IF THE CABLE IS LEFT CONNECTED TO THE MODEM THE FOLLOWING TESTS WILL FAIL:
AT22, AT23, AT25, AT30, AT32, AT56
- B. THE PROGRAM WILL TYPE OUT INSTRUCTIONS TO SET IN THE DESIRED SR OPTIONS. PRESS CONTINUE 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 776J2X 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 SR9 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 RE-
QUESTED, 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 TRANS-
MISSIONS 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 REQ-
UESTED, 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 SUCCES-
SIVE CHARACTERS.

C. THE PROGRAM WILL NOW RUN WITHOUT ERROR OR END TYPEOUTS.

4.4 PRG3 - SINGLE CHARACTER MAINT MODE DATA TEST

- A. LOAD ADDRESS = 00200 (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 REQ-
UESTED, FOLLOWED BY A CARRIAGE RETURN.
- B. THE PROGRAM WILL REQUEST A TEST CHARACTER. TYPE THE TEST CHAR-
ACTER, FOLLOWED BY A CARRIAGE RETURN.
- C. THE PROGRAM WILL NOW RUN CONTINUOUSLY REPORTING ANY DATA FAIL-
URES.

4.5 PRG4 - SPECIAL BINARY COUNT MAINT. MODE DATA TEST

- A. LOAD ADDRESS = 00200
TYPE PROGRAM NUMBER = 4.
THE DIAGNOSTIC WILL IDENTIFY THE PROGRAM YOU SELECTED, AND
REQUEST THE LINE # YOU WISH TO TEST. TYPE THE LINE # AS REQ-
UESTED, 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(8) 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 ER-
 ROR CALL.

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

```

375 .LIST SEQ,BIN
376 .ENABLE ABS,AMA
377
378 ;DL11-E,C/D DIAGNOSTIC PROGRAM (OFF LINE TESTS)
379 ;
380 ;PRG0- INPUT-OUTPUT LOGIC TESTS
381 ;PRG1- TRANSMITTER SCOPE LOOP
382 ;PRG2- RECEIVER SCOPE LOOP
383 ;PRG3- SINGLE CHARACTER MAINTENANCE MODE DATA TEST
384 ;PRG4- SPECIAL BINARY COUNT MAINTENANCE MODE DATA TEST
385 ;
386 ;STANDARD SR SWITCH OPTIONS (SWITCH SET TO A 1 )
387 ;
388 ;SR15- HALT ON ERROR
389 ;SR14- SCOPE.
390 ;SR13- INHIBIT PRINTOUT
391 ;SR12- SELECT LINE NUMBER AND LOCK ON IT
  
```

```
392 ;SR11- INHIBIT ITERATION.  
393 ;SR10- HALT AT END CURRENT TEST, TEST NO. IN DATA LIGHTS  
394 ;SR9- SELECT ROUTINE.  
395 ;SR7- DISABLE STALL MODE AND RUN FULL SPEED.  
396 ;SR6 THROUGH SRO - NUMBER OF ROUTINE TO BE SELECTED.  
397 ;  
398 ;STANDARD CONFIGURATION  
399 ;CHARACTER LENGTH 8  
400 ;STOP CODE 2  
401 .=0  
402 000000 005706 ERTF ;UNASSIGNED TRAP  
403 000002 000000 0  
404 000004 005706 MACHER: ERTF ;SP OVERFLOW, BUS ERROR TRAP  
405 000006 000040 40  
406 000010 005706 ERTF ;RESERVED INSTRUCTION TRAP  
407 000012 000100 100  
408 000014 005706 ERTF ;TRACE TRAP  
409 000016 000140 140  
410 000020 006014 MAPVEC ;TRAP TO MAP VECTOR  
411 000022 000340 PRTY7  
412 000024 006234 PFAIL ;POWER FAIL TRAP  
413 000026 000340 PRTY7  
414 000030 002766 EMTINT ;EMT TRAP  
415 000032 000340 PRTY7  
416 000034 005706 ERTF  
417 000036 000340 340  
418 000040 000042 .+2  
419 000042 000000 HALT  
420 000046 000046 .=46  
421 000046 005312 LOGIC 117.  
422 000165 .REPT ;TRAP TO TRAP REPORTER  
423 .+2  
424 4  
425 .ENDR  
426  
427  
428 ;EQUATE STATEMENTS  
429 PSW=177776  
430 001176 SPBOT=1176  
431 000240 NOP=240  
432 000000 OPEN=0  
433 100000 MANUAL=BIT15  
434 100000 BIT15=100000  
435 040000 BIT14=40000  
436 020000 BIT13=20000  
437 010000 BIT12=10000  
438 004000 BIT11=4000  
439 002000 BIT10=2000  
440 001000 BIT9=1000  
441 000400 BIT8=400  
442 000200 BIT7=200  
443 000100 BIT6=100  
444 000040 BIT5=40  
445 000020 BIT4=20  
446 000010 BIT3=10  
447 000004 BIT2=4
```

```
448 000002 BIT1=2  
449 000001 BIT0=1  
450 005726 PDSP=5726 ;POP THE STACK. SAME AS TST (6)+  
451 022626 PDSP2=022626 ;POP STACK TWICE. SAME AS CMP (6)+.(6)+  
452 000340 PRTY7=340 ;PRIORITY LEVEL DEFINITIONS  
453 000300 PRTY6=300  
454 000240 PRTY5=240  
455 000200 PRTY4=200  
456 000140 PRTY3=140  
457 000100 PRTY2=100  
458 000040 PRTY1=40  
459 000000 PRTY0=0  
460 104000 TYPE=EMT+0  
461 104001 TYPES=EMT+1  
462 104002 STALL=EMT+2  
463 104003 ERROR=EMT+3  
464 104004 DATCHK=EMT+4  
465 104005 CHALT=EMT+5  
466 104006 STRXV=EMT+6  
467 104007 STTXV=EMT+7  
468 104010 EHALT=EMT+10  
469 104011 SRESET=EMT+11  
470 104012 SCOPE=EMT+12  
471 104013 SAVREG=EMT+13  
472 104014 RSTREG=EMT+14  
473 104015 ERROR1=EMT+15  
474 104016 DELAY=EMT+16  
475 104017 TIMERX=EMT+17  
476 104020 TIMETX=EMT+20  
477 177777 ATLAST=-1  
478 100000 CD=100000 ;FLAG FOR C/D TESTS  
479  
480 .LIST ME  
481 .=174  
482 000174 177570 SRPTR: 177570  
483 000176 000000 SQFTSR: 000000  
484 000200 .=200  
485 000200 000137 JMP @#STARTZ ;GO TO START OF PROGRAM.  
486 000204 .=204  
487 000204 000137 JMP @#RESTART  
488 001200 .=1200  
489  
490 ;DEVICE ADDRESS LIST  
491 ;LSB BIT0 IS SET TO A 1 BY MAPPER IF DEVICE NOT FOUND  
492 ;TO TEST THAT LINE NOT FOUND CLEAR BIT0 IN THAT DEVICE ADDRESS  
493 ;IN THIS TABLE AFTER MAPPING DONE  
494 ;*****  
495 001200 175610 RXCR0: 175610 ;LINE 0 DEVICE ADDRESS (RXCSR)  
496 001202 175620 RXCR1: 175620 ;LINE 1 DEVICE ADDRESS (RXCSR)  
497 001204 175630 RXCR2: 175630 ;LINE 2 DEVICE ADDRESS (RXCSR)  
498 001206 175640 RXCR3: 175640 ;LINE 3 DEVICE ADDRESS (RXCSR)  
499 001210 175650 RXCR4: 175650 ;LINE 4 DEVICE ADDRESS (RXCSR)  
500 001212 175660 RXCR5: 175660 ;LINE 5 DEVICE ADDRESS (RXCSR)  
501 001214 175670 RXCR6: 175670 ;LINE 6 DEVICE ADDRESS (RXCSR)  
502 001216 175700 RXCR7: 175700 ;LINE 7 DEVICE ADDRESS (RXCSR)  
503 001220 175710 RXCR10: 175710 ;LINE 10 DEVICE ADDRESS (RXCSR)
```



```

504 001222 175720 RXCR11: 175720 ;LINE 11 DEVICE ADDRESS (RXCSR)
505 001224 175730 RXCR12: 175730 ;LINE 12 DEVICE ADDRESS (RXCSR)
506 001226 175740 RXCR13: 175740 ;LINE 13 DEVICE ADDRESS (RXCSR)
507 001230 175750 RXCR14: 175750 ;LINE 14 DEVICE ADDRESS (RXCSR)
508 001232 175760 RXCR15: 175760 ;LINE 15 DEVICE ADDRESS (RXCSR)
509 001234 175770 RXCR16: 175770 ;LINE 16 DEVICE ADDRESS (RXCSR)
510 001236 176000 RXCR17: 176000 ;LINE 17 DEVICE ADDRESS (RXCSR)
511 001240 176010 RXCR20: 176010 ;LINE 20 DEVICE ADDRESS (RXCSR)
512 001242 176020 RXCR21: 176020 ;LINE 21 DEVICE ADDRESS (RXCSR)
513 001244 176030 RXCR22: 176030 ;LINE 22 DEVICE ADDRESS (RXCSR)
514 001246 176040 RXCR23: 176040 ;LINE 23 DEVICE ADDRESS (RXCSR)
515 001250 176050 RXCR24: 176050 ;LINE 24 DEVICE ADDRESS (RXCSR)
516 001252 176060 RXCR25: 176060 ;LINE 25 DEVICE ADDRESS (RXCSR)
517 001254 176070 RXCR26: 176070 ;LINE 26 DEVICE ADDRESS (RXCSR)
518 001256 176100 RXCR27: 176100 ;LINE 27 DEVICE ADDRESS (RXCSR)
519 001260 176110 RXCR30: 176110 ;LINE 30 DEVICE ADDRESS (RXCSR)
520 001262 176120 RXCR31: 176120 ;LINE 31 DEVICE ADDRESS (RXCSR)
521 001264 176130 RXCR32: 176130 ;LINE 32 DEVICE ADDRESS (RXCSR)
522 001266 176140 RXCR33: 176140 ;LINE 33 DEVICE ADDRESS (RXCSR)
523 001270 176150 RXCR34: 176150 ;LINE 34 DEVICE ADDRESS (RXCSR)
524 001272 176160 RXCR35: 176160 ;LINE 35 DEVICE ADDRESS (RXCSR)
525 001274 176170 RXCR36: 176170 ;LINE 36 DEVICE ADDRESS (RXCSR)
526 001276 177777 XORADD: 177777 ;LINE 37 SPECIAL ADDRESS FOR XOR
527 001300 177777 RXEND: 177777 ;LINE XX DEVICE ADDRESS (RXCSR)
528 ;
529 ;CHARACTER LENGTH, PRIORITY, C/D MASK
530 ;INITIALLY SET FOR DL11-E, PRIORITY=4, CHARACTER LENGTH=8
531 ;BIT 15 SET TO A 1 = THAT LINE HAS DL11-C OR DL11-D
532 ;EX: 140377 = DL11C OR DL11D, PRIORITY = 4, CHARACTER LENGTH = 8
533 ;BITS 12-14 = PRIORITY LEVEL THAT LINE
534 ;BITS 0-7 = CHARACTER MASK EX. 377=8, 177=7, 77=6, 37=5
535 ;*****
536 001302 040377 CMAS0: 040377 ;LINE 0 CHARACTER MASK, PRIORITY, C/D FLAG
537 001304 040377 CMAS1: 040377 ;LINE 1 CHARACTER MASK, PRIORITY, C/D FLAG
538 001306 040377 CMAS2: 040377 ;LINE 2 CHARACTER MASK, PRIORITY, C/D FLAG
539 001310 040377 CMAS3: 040377 ;LINE 3 CHARACTER MASK, PRIORITY, C/D FLAG
540 001312 040377 CMAS4: 040377 ;LINE 4 CHARACTER MASK, PRIORITY, C/D FLAG
541 001314 040377 CMAS5: 040377 ;LINE 5 CHARACTER MASK, PRIORITY, C/D FLAG
542 001316 040377 CMAS6: 040377 ;LINE 6 CHARACTER MASK, PRIORITY, C/D FLAG
543 001320 040377 CMAS7: 040377 ;LINE 7 CHARACTER MASK, PRIORITY, C/D FLAG
544 001322 040377 CMAS10: 040377 ;LINE 10 CHARACTER MASK, PRIORITY, C/D FLAG
545 001324 040377 CMAS11: 040377 ;LINE 11 CHARACTER MASK, PRIORITY, C/D FLAG
546 001326 040377 CMAS12: 040377 ;LINE 12 CHARACTER MASK, PRIORITY, C/D FLAG
547 001330 040377 CMAS13: 040377 ;LINE 13 CHARACTER MASK, PRIORITY, C/D FLAG
548 001332 040377 CMAS14: 040377 ;LINE 14 CHARACTER MASK, PRIORITY, C/D FLAG
549 001334 040377 CMAS15: 040377 ;LINE 15 CHARACTER MASK, PRIORITY, C/D FLAG
550 001336 040377 CMAS16: 040377 ;LINE 16 CHARACTER MASK, PRIORITY, C/D FLAG
551 001340 040377 CMAS17: 040377 ;LINE 17 CHARACTER MASK, PRIORITY, C/D FLAG
552 001342 040377 CMAS20: 040377 ;LINE 20 CHARACTER MASK, PRIORITY, C/D FLAG
553 001344 040377 CMAS21: 040377 ;LINE 21 CHARACTER MASK, PRIORITY, C/D FLAG
554 001346 040377 CMAS22: 040377 ;LINE 22 CHARACTER MASK, PRIORITY, C/D FLAG
555 001350 040377 CMAS23: 040377 ;LINE 23 CHARACTER MASK, PRIORITY, C/D FLAG
556 001352 040377 CMAS24: 040377 ;LINE 24 CHARACTER MASK, PRIORITY, C/D FLAG
557 001354 040377 CMAS25: 040377 ;LINE 25 CHARACTER MASK, PRIORITY, C/D FLAG
558 001356 040377 CMAS26: 040377 ;LINE 26 CHARACTER MASK, PRIORITY, C/D FLAG
559 001360 040377 CMAS27: 040377 ;LINE 27 CHARACTER MASK, PRIORITY, C/D FLAG

```

```

560 001362 040377 CMAS30: 040377 ;LINE 30 CHARACTER MASK, PRIORITY, C/D FLAG
561 001364 040377 CMAS31: 040377 ;LINE 31 CHARACTER MASK, PRIORITY, C/D FLAG
562 001366 040377 CMAS32: 040377 ;LINE 32 CHARACTER MASK, PRIORITY, C/D FLAG
563 001370 040377 CMAS33: 040377 ;LINE 33 CHARACTER MASK, PRIORITY, C/D FLAG
564 001372 040377 CMAS34: 040377 ;LINE 34 CHARACTER MASK, PRIORITY, C/D FLAG
565 001374 040377 CMAS35: 040377 ;LINE 35 CHARACTER MASK, PRIORITY, C/D FLAG
566 001376 040377 CMAS36: 040377 ;LINE 36 CHARACTER MASK, PRIORITY, C/D FLAG
567 001400 040377 CMAS37: 040377 ;LINE 37 SPECIAL ADDRESS FOR XOR
568 ;
569 001402 000000 UMASK: 0 ;MASK FOR DEVICE UT
570 001404 000000 RMASK: 0 ;MASK FOR CHAR LENGTH FOR DEVICE UT
571 001406 177740 STLMSK: 177740 ;MASK FOR MAX RANDOM STALL
572 ;
573 001410 000000 RXCSR: 0 ;RECEIVER UNDER TEST
574 001412 000000 RXBUF: 0 ;RECEIVER BUFFER UNDER TEST
575 001414 000000 TXCSR: 0 ;TRANSMITTER CSR UNDER TEST
576 001416 000000 TXBUF: 0 ;TRANSMITTER BUFFER UNDER TEST
577 001420 000000 RXVTR: 0 ;RECEIVER VECTOR UNDER TEST
578 001422 000000 RXLVL: 0 ;RECEIVER PRIORITY LEVEL UT
579 001424 000000 TXVTR: 0 ;TRANSMITTER VECTOR UNDER TEST
580 001426 000000 TXLVL: 0 ;TRANSMITTER PRIORITY LEVEL UT
581 ;
582 ;*****
583 001430 177560 TKS: 177560 ;LSR CSR
584 001432 177562 TKB: 177562 ;LSR BUFFER
585 001434 177564 TPS: 177564 ;LSP CSR
586 001436 177566 TPB: 177566 ;LSP BUFFER
587 001440 000060 TKVTR: 60 ;LSR INTERRUPT VECTOR
588 001442 000200 TKLVL: PRTY4 ;LSR PRIORITY LEVEL
589 001444 000064 TPVTR: 64 ;LSP INTERRUPT VECTOR
590 001446 000200 TPLVL: PRTY4 ;LSP PRIORITY LEVEL
591 001450 000000 PRGNUM: OPEN ;CONTAINS CURRENT PROGRAM#
592 001452 000000 KSTART: OPEN ;CURRENT PROGRAM START ADDRESS.
593 001454 000000 CURTST: OPEN ;CONTAINS ADDR OF CURRENT TEST.
594 001456 000000 RTNND: OPEN ;CONTAINS CURRENT TEST #.
595 001460 000000 TNND: 0 ;CONTAINES EDITED TNUM
596 001462 000000 NXTST: OPEN ;CONTAINS ADDR OF NEXT TEST.
597 001464 000000 ICTR: OPEN ;CONTAINS CURRENT ITERATION COUNT
598 001466 000000 SCOPTR: OPEN ;CONTAINS CURRENT SCOPE POINTER.
599 001470 000000 QLDPS: 0 ;PS SAVED FROM TRAP TO EMT ROUTINE
600 001472 000000 FMAP: 0 ;MAPPING FLAG, 1= MAPPING IN PROGRESS
601 001474 006446 PRGTAB: PRG0 ;PRG0 START ADDRESS
602 001476 014522 PRG1 ;PRG1 START ADDRESS
603 001500 014574 PRG2 ;PRG2 START ADDRESS
604 001502 014672 PRG3 ;PRG3 START ADDRESS
605 001504 014732 PRG4 ;PRG4 START ADDRESS
606 001506 005110 INCRPG ;INCORRECT PROGRAM SELECTED
607 001510 005110 INCRPG
608 001512 005110 INCRPG
609 001514 003342 EMTTAB: TYP ;POINTER TO TYPEOUT ROUTINE
610 001516 003464 TYP ;POINTER TO CHAINED MESSAGES ROUTINE
611 001520 003774 STAL ;POINTER TO RANDOM STALL ROUTINE
612 001522 005516 ERR ;POINTER TO ERROR ROUTINE
613 001524 005424 DTCHK
614 001526 000000 OPEN
615 001530 003120 STLSRV

```

```

616 001532 003154          STLSPV
617 001534 005362          EHLT
618 001536 003210          SRSETT
619 001540 002540          CHAINN
620 001542 003020          SAVRG
621 001544 003060          RSTRG
622 001546 005540          ERR1
623 001550 003726          DLY
624 001552 004044          TMRX
625 001554 004054          TMTX
626
627 001556 000000          CRBUF: OPEN
628 001560 000000          CRBUFA: OPEN
629 001562 000000          CRBUFB: OPEN
630 001564 000000          CTRO: OPEN
631 001566 000000          CTR1: OPEN
632 001570 000000          CTR2: OPEN
633 001572 000000          CTR3: OPEN
634 001574 000000          CTR4: OPEN
635 001576 000000          CTR5: OPEN
636 001600 000000          CTR6: OPEN
637 001602 000000          CTR7: OPEN
638 001604 000000          TXCSRT: OPEN
639 001606 000000          RXCSRT: OPEN
640 001610 000000          RXBUFT: OPEN
641 001612 000000          FOUNDV: 0
642 001614 000000          LINEND: 0
643 001616 000000          TEMP: OPEN
644 001620 000000          TEMP1: 0
645 001622 000000          TEMP2: 0
646 001624 000000          COUNT: 0
647 001626 000000          FTITLE: 0
648 001630 000000          FNONE: 0
649 001632 000000          TOPC: 0
650 001634 000000          FROMPC: 0
651 001636 000000          PASCNT: 0
652 001640 012706 001176  STARTZ: MOV #SPBOT,%6
653 001644 013746 000006  MOV 6,-(SP)
654 001650 013746 000004  MOV 4,-(SP)
655 001654 012737 001670 000004  MOV #1$,4
656 001662 005777 176306  TST @SRPTR
657
658 001666 000404          BR 2$
659 001670 012737 000176 000174 1$  MOV #SOFTSR,SRPTR
660
661 001676 022626          CMP (6)+,(6)+
662 001700 012637 000004 2$  MOV (6)+,4
663 001704 012637 000006  MOV (6)+,6
664 001710 005037 001626  CLR @#FTITLE
665 001714 013746 000004  MOV @#4,-(%6)
666 001720 012737 002020 000004  MOV #XORA,@#4
667 001726 005737 177060  TST @#177060
668 001732 012637 000004  MOV (%6)+,@#4
669 001736 012737 174000 001276  MOV #174000,@#XORADD
670 001744 012737 177777 002016  MOV #-1,@#XORFLG
671 001752 104000          TYPE

```

; POINTER TO ERROR HALT ROUTINE.

; SAVE CURRENT VECTOR

```

;SET UP TIME OUT VECTOR
;TRY TO REFERENCE THE
;HARDWARE SWITCH REGISTER
;BRANCH IF NO TIME OUT TRAP OCCURRS
;CHANGE THE SWITCH REGISTER POINTER
;TO POINT TO A SOFTWARE SWITCH REGISTER
;RESTORE THE STACK
;RESTORE TIME OUT VECTOR

```

```

672 001754 001762          MESS1
673 001756 000137 002044  JMP @#START
674 001762 005015 047531 020125  MESS1: .ASCII <15><12>'YOU ARE ON AN XOR TESTER@'
675 001770 051101 020105 047117
676 001776 040440 020116 047530
677 002004 020122 042524 052123
678 002012 051105 100
679 002016
680 002016 000000          XORFLG: .EVEN .WORD 0
681
682 002020 022626          XORA: CMP (%6)+,(%6)+
683 002022 012637 000004  MOV (%6)+,@#4
684 002026 012737 177777 001276  MOV #-1,@#XORADD
685 002034 005037 002016  CLR @#XORFLG
686 002040 000137 002044  JMP @#START
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702 002044 012706 001176  START: MOV #SPBOT,%6
703 002050 012737 006234 000024  MOV #PFAIL,24
704 002056 005037 001612  CLR FOUNDV
705 002062 005037 001472  CLR FMAP
706 002066 004737 003300  JSR %7,CLRCD
707 002072 004737 003516  JSR %7,OVRLAY
708 002076 005737 001626  TST FTITLE
709 002102 001054          BNE START1
710 002104 104000          TYPE
711 002106 015044          MTTI
712 002110 005237 001626  INC FTITLE
713 002114 005037 001630  CLR FNONE
714 002120 012737 002160 000004  MOV #MAPNE,MAPCHER
715 002126 012704 001200  MOV #RXCR0,%4
716 002132 021437 001300  MAPA: CMP (%4),@#RXEND
717 002136 001430          BEQ MAPEND
718 002140 042714 000001  BIC #BIT0,(4)
719 002144 005037 177776  CLR PSW
720 002150 005774 000000  TST @ (4)
721 002154 000240          NOP
722 002156 000404          BR MAPOK
723 002160 052724 000001  MAPNE: BIS #BIT0,(4)+
724 002164 022626          POPSP2
725 002166 000761          BR MAPA
726 002170 012437 001620  MAPOK: MOV (4)+,TEMP1
727 002174 004537 004474  JSR %5,DACNV

```

```

;*****
;MACR TSTAA AX,B,C,D,E
;*****
AT'E': C ;TEST NUMBER
AT'D' ;ADDRESS OF NEXT TEST
B ;ITERATION COUNT
'AX'A ;SCOPE ENTRY POINT
X=X+1
;*****

```

```

;*****
;MACR TSTA B,AX,Z
;*****
TSTAA AX,B,\X+1+Z,\X+2,\X+1
;*****

```

```

;SET BOTTOM OF SP STACK.
;CLEAR DEVICE UT PARAMETERS
;OVERLAY TRAP AREA
;TITLE PRINTED AND MAP MADE
;YES, SKIP OVER THIS
;CLEAR DEVICE PRESENT FLAG
;SET UP NO DEVICE PRESENT RETURN
;SET UP DEVICE POINTER
;LAST DEVICE
;YES, EXIT
;CLEAR ODD ADDRESS
;TEST DEVICE
;NOT LIVING
;SAVE DEVICE ADDRESS FOR TYPING

```

```

728 002200 001620          TEMP1
729 002202 015135          MDEVAD
730 002204 000006          6
731 002206 104000          TYPE
732 002210 015135          MDEVAD
733 002212 005237 001630  INC      FNONE      ;SET HAVE DEVICE
734 002216 000745          BR        MAPA
735 002220 012737 005706 000004 MAPEND: MOV    #ERTP,MACHER ;RESET TRAPS
736 002226 005737 001630  TST     FNONE      ;ANY DEVICES PRESENT
737 002232 001424          BEQ     MAPERR     ;NO, ERROR
738 002234 012701 001200  START1: MOV   #RXCRO,%1
739 002240 032711 000001  START2: BIT   #BIT0,(1) ;IS DEVICE LIVING
740 002244 001013          BNE     START3     ;NO, CHECK FOR END
741 002246 010137 001614  MOV     %1,LINENO  ;CALCULATE LINE NUMBER UNDER TEST
742 002252 162737 001200 001614  SUB     #RXCRO,LINENO
743 002260 006237 001614  ASR     LINENO
744 002264 011101          MOV     (1),%1     ;YES, LOAD AND EXIT
745 002266 004737 006072  JSR     %7,FORMAD
746 002272 000420          BR     START4
747 002274 005721          TST     (1)+
748 002276 020127 001300  START3: CMP   %1,#RXEND ;END OF TABLE
749 002302 001356          BNE     START2     ;NO, LOOP
750 002304 104000          MAPERR: TYPE
751 002306 015211          MNONE
752 002310 005737 000042  TST     @#42      ;MONITOR LOAD
753 002314 001402          BEQ     .+6        ;NO, CONTINUE
754 002316 000137 005302  JMP     PRGXTL     ;YES, EXIT
755 002322 005037 001626  CLR     FTITLE
756 002326 000000          HALT
757 002330 000137 002044  JMP     START
758 002334 012737 000001 001636  START4: MOV   #1,PASCNT
759 002342 005037 177776  CLR     PSW
760 002346 005037 001456  CLR     RTNNO
761 002352 104000          TYPE
762 002354 015155          PGMMSG ;CALL FOR PROGRAM NUMBER.
763 002356 004737 003554  JSR     PC,RDOCT  ;READ IN PROGRAM NUMBER.
764 002362 012600          MOV     (SP)+,%0  ;INPUT DATA TO R0
765 002364 042700 177770  BIC     #177770,%0 ;LIMIT (SR) TO BITS 3-0
766 002370 010037 001450  MOV     %0,PRGNUM ;SAVE PROGRAM #
767 002374 006300          ASL     %0
768 002376 000170 001474  JMP     @PRGTAB(0) ;GO TO SELECTED PROGRAM.
769
770 002402 013737 001452 001462 ; GETRDY: MOV   KSTART,NXTST ;ADDR OF 1ST ROUTINE TO NXTST
771 002410 012737 005706 000004 GTRDYX: MOV   #ERTP,MACHER ;RESET MACHER TRAP.
772 002416 012737 000040 000006  MOV     #40,MACHER+2
773 002424 005037 001472  CLR     FMAP
774 002430 012706 001176  MOV     #SPBOT,%6 ;SET BOTTOM OF STACK.
775 002434 104011          SRESET ;ISSUE RESET.
776 002436 005037 177776  CLR     PSW
777 002442 004737 002734  GTRDYA: JSR   %7,FORWD ;ROLL FORWARD TO "NEXT" ROUTINE.
778 002446 032777 001000 175520  BIT     #BIT9,@SRPTR ;CHECK SELECT ROUTINE SWITCH
779 002454 001011          BNE     GTRDYC     ;BRANCH IF SELECT ROUTINE SWITCH IS SET.
780 002456 005737 001402  UMASK  ;C/D DEVICE
781 002462 100003          BPL     GTRDA1    ;NO, CONTINUE
782 002464 005737 001456  TST     RTNNO     ;THIS A C/D TEST
783 002470 100364          BPL     GTRDYA    ;NO, DO NEXT TEST

```

```

784 002472 000177 176756  GTRDA1: JMP   @CURTST ;GO RUN CURRENT ROUTINE.
785 002476 000464          BR     CHNB        ;NO GO. MANUAL RTN BYPASSED.
786 002500 017700 175470  GTRDYC: MOV   @SRPTR,%0 ;(SR) TO R0
787 002504 042700 176000  BIC     #177600,%0 ;MASK UNDESIRED BITS
788 002510 123700 001456  CMPPB   RTNNO,%0 ;COMPARE RTNNO TO (R0)
789 002514 001002          BNE     GTRDYD     ;BRANCH IF ROUTINE NOT FOUND YET.
790 002516 000177 176732  JMP     @CURTST ;GO RUN ROUTINE.
791 002522 022737 177777 001462  GTRDYD: CMP   #-1,NXTST ;NO. CHECK FOR LAST ROUTINE.
792 002530 001344          BNE     GTRDYA    ;BRANCH IF NOT LAST ROUTINE.
793 002532 004737 005100  JSR     %7,INCRTN ;YES. INCORRECT ROUTINE SELECTED.
794 002536 000721          BR     GETRDY     ;START OVER.
795
796 002540 032777 040000 175426  CHAINN: BIT   #BIT14,@SRPTR ;CHECK FOR SCOPE OPTION.
797 002546 001403          BEQ     CHNA
798 002550 013716 001466  CHNAB: MOV   SCOPTR,%6 ;BRANCH IF SCOPE SW NOT SET.
799 002554 000002          RTI ;SET UP TO RETURN TO ROUTINE.
800 002556 005737 002016  CHNA: TST    @#XORFLG ;RETURN TO ROUTINE.
801 002562 100011          BPL     1$
802 002564 013746 000004  MOV     @#4,-(%6)
803 002570 012737 002676 000004  MOV     #XOR,@#4
804 002576 005737 177060  TST     @#177060 ;TEST FOR XOR
805 002602 012637 000004  MOV     (%6)+,@#4
806 002606 032777 004000 175360 1$: BIT   #BIT11,@SRPTR ;TEST INHIBIT ITERATION SWITCH
807 002614 001003          BNE     CHNAA     ;BRANCH IF INHIBIT ITERATION SW SET.
808 002616 005337 001464  DEC     ICTR ;DECREMENT ITERATION COUNT.
809 002622 001352          BNE     CHNAB     ;BRANCH IF COUNT NOT 0.
810 002624 022626          POPSP2 ;POP STACK TWICE
811 002626 032777 002000 175340  CHNAA: BIT   #BIT10,@SRPTR
812 002634 001405          BEQ     CHNB
813 002636 013700 001456  MOV     RTNNO,%0
814 002642 042700 100000  BIC     #BIT15,%0
815 002646 000000          HALT
816 002650 032777 001000 175316  CHNB: BIT   #BIT9,@SRPTR ;CHECK SELECT ROUTINE SWITCH
817 002656 001251          BNE     GETRDY    ;BRANCH IF SELECT RTN SW SET
818 002660 022737 177777 001462  CMP     #-1,NXTST ;LAST TEST?
819 002666 001250          BNE     GTRDYX    ;BRANCH IF NOT LAST TEST.
820 002670 004737 005122  JSR     %7,PRGEND ;PROGRAM END.
821 002674 000642          BR     GETRDY
822
823 002676 022626          XOR: CMP     (%6)+,(%6)+
824 002700 012637 000004  MOV     (%6)+,@#4
825 002704 000721          BR     CHNAB
826
827 ;INIT FOR C/D - WITHOUT JUMPER RESET STARTS ASSEMBLING CHARACTER SETTING DONE
828 ;SET MAINT, DELAY, CLEAR RX DONE
829 002706 005737 001402  CDINIT: TST   UMASK ;C-D DEVICE
830 002712 100007          BPL     CDINX    ;NO, EXIT
831 002714 052777 000004 176472  BIS     #BIT2,@TXCSR ;SET MAINT BIT
832 002722 104016          DELAY ;WAIT 1.5 SEC
833 002724 002734          1500.
834 002726 005777 176460  TST     @RXBUF
835 002732 000207          RTS ;CLEAR RX DONE
836
837
838 002734 013705 001462  ;FORWD: MOV   NXTST,%5 ;ADDR OF NEXT ROUTINE TO R5.
839 002740 012537 001456  MOV     (5)+,RTNNO ;GET NEXT ROUTINE NUMBER.

```

```
840 002744 012537 001462      MOV      (5)+,NXTST      ;GET ADDR OF NEXT "NEXT" ROUTINE.
841 002750 012537 001464      MOV      (5)+,ICTR      ;GET ITERATION COUNT.
842 002754 012537 001466      MOV      (5)+,SCOPTR    ;GET SCOPE LOOP ENTRY POINTER.
843 002760 010537 001454      MOV      %5,CURTST     ;ADDR OF NOW CURRENT TEST TO CURTST.
844 002764 000207              RTS                    ;EXIT FORWD SUBROUTINE.
845
;
846 002766 011646              EMTINT: MOV    @%6,-(6)   ;GET SAVED PC.
847 002770 162716 000002      SUB      #2,@%6        ;DECREMENT PC BY 2.
848 002774 017616 000000      MOV      @(%6),%6      ;
849 003000 006316              EMTA:  ASL      %6        ;EMT ARG X 2.
850 003002 042716 177001      BIC      #177001,%6    ;REMOVE 7 MSB.
851 003006 062716 001514      ADD      #EMTTAB,%6    ;FORM EMT RTN ADDR.
852 003012 017616 000000      MOV      @(%6),%6      ;
853 003016 000136              JMP      @(%6)+        ;GO TO EMT ROUTINE.
854
;
855
856
;SAVE REGS 0 TO 4 SUBROUTINE.
;
857 003020 012637 003054      SAVRG:  MOV    (6)+,SVRPC ;SAVE PC AND PSW.
858 003024 012637 003056      MOV      (6)+,SVRPSW   ;
859 003030 010446              MOV      %4,-(6)      ;SAVE REGS 0 - 4
860 003032 010346              MOV      %3,-(6)      ;IN STACK.
861 003034 010246              MOV      %2,-(6)
862 003036 010146              MOV      %1,-(6)
863 003040 010046              MOV      %0,-(6)
864 003042 013746 003056      MOV      SVRPSW,-(6)   ;RESTORE PC AND PSW.
865 003046 013746 003054      MOV      SVRPC,-(6)
866 003052 000002              RTI                    ;EXIT.
867 003054 000000      SVRPC:  OPEN
868 003056 000000      SVRPSW: OPEN
869
;
870
871
;RESTORE REGS 0 TO 4 SUBROUTINE.
;
872 003060 012637 003114      RSTRG:  MOV    (6)+,RSTPC ;SAVE PC AND PSW.
873 003064 012637 003116      MOV      (6)+,RSTPSW   ;
874 003070 012600              MOV      (6)+,%0      ;RESTORE REGS 0 - 4
875 003072 012601              MOV      (6)+,%1      ;FROM STACK.
876 003074 012602              MOV      (6)+,%2
877 003076 012603              MOV      (6)+,%3
878 003100 012604              MOV      (6)+,%4
879 003102 013746 003116      MOV      RSTPSW,-(6)   ;RESTORE PC AND PSW.
880 003106 013746 003114      MOV      RSTPC,-(6)
881 003112 000002              RTI                    ;EXIT
882 003114 000000      RSTPC:  OPEN
883 003116 000000      RSTPSW: OPEN
884
;ROUTINE TO SET RECEIVER INTERRUPT VECTOR AND PRIORITY
885
886 003120 004737 006270      STLSRV: JSR    %7,TSTVEC ;
887 003124 017637 000000      MOV      @(%6),STPRA+2 ;MOVE VECTOR ADDR TO STPRA+2
888 003132 062716 000002      ADD      #2,@%6        ;SET UP EXIT
889 003136 013701 001420      MOV      RXVTR,%1
890 003142 012721 000000      STPRA:  MOV    #OPEN,(1)+ ;SET VECTOR ADDRESS
891 003146 013721 001422      MOV      RXLVL,(1)+    ;SET PRIORITY
892 003152 000002              RTI                    ;EXIT
893
;ROUTINE TO SET TRANSMITTER INTERRUPT VECTOR AND PRIORITY.
894
895 003154 004737 006270      STLSPV: JSR    %7,TSTVEC
```

```
896 003160 017637 000000 003200      MOV      @(%6),STPPA+2 ;MOVE VECTOR ADDR TO STPPA+2
897 003166 062716 000002      ADD      #2,@%6        ;SET UP EXIT
898 003172 013701 001424      STPPA:  MOV    TXVTR,%1
899 003176 012721 000000      MOV      #OPEN,(1)+    ;SET VECTOR ADDRESS.
900 003202 013721 001426      MOV      TXLVL,(1)+    ;SET PRIORITY
901 003206 000002              RTI                    ;EXIT.
902
;ROUTINE TO ISSUE RESET.
903
904 003210 012700 052525      SRSETT: MOV    #52525,%0 ;DATA TO R0.
905 003214 005100              COM      %0            ;COMPLEMENT (R0).
906 003216 010037 003212      MOV      %0,SRSETT+2   ;(R0) TO SRSETT+2.
907 003222 000005              RESET
908 003224 000002              RTI                    ;ISSUE RESET. (R0) IS
909
;DISPLAYED. EXIT.
910
;RANDOM NUMBER GENERATOR. ROUTINE EXITS WITH NUMBER IN REGISTER 0.
911 003226 013700 003274      RNGEN:  MOV    RP1,%0
912 003232 006100              ROL      %0
913 003234 006100              ROL      %0
914 003236 063700 003276      ADD      RP2,%0
915 003242 010037 003274      MOV      %0,RP1
916 003246 006100              ROL      %0
917 003250 006100              ROL      %0
918 003252 063700 003276      ADD      RP2,%0
919 003256 006100              ROL      %0
920 003260 006100              ROL      %0
921 003262 010037 003276      MOV      %0,RP2
922 003266 013700 003274      MOV      RP1,%0
923 003272 000207              RTS                    ;EXIT. NUMBER IN R0
924 003274 001233      RP1: 1233
925 003276 007622      RP2: 7622
926
;
927
;CLRCD - CLEAR CURRENT DEVICE PARAMETERS
928 CLRCD:  CLR    TXBUF
929          CLR    TXCSR
930          CLR    RXCSR
931          CLR    RXBUF
932          CLR    RXVTR
933          CLR    TXVTR
934          CLR    RXLVL
935          CLR    TXLVL
936          RTS      %7
937
;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
938
939 003342 011600      TYP:  MOV      @%6,%0    ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS.
940 003344 062716 000002      ADD      #2,@%6        ;SET UP EXIT.
941 003350 011000      MOV      @%0,%0        ;ADDRESS OF MESSAGE TO R0.
942 003352 112037 003462      TYPA:  MOVB   (0)+,TYPDAT ;GET CHARACTER
943 003356 122737 000100 003462      CMPB   #100,TYPDAT     ;CHECK FOR"@"CHARACTER
944 003364 001001      BNE    TYPC           ;BRANCH IF NOT"@".
945 003366 000002      RTI                    ;TERMINATOR CHAR. DONE. EXIT.
946 003370 122737 000045 003462      TYPC:  CMPB   #45,TYPDAT ;CHECK FOR"%".
947 003376 001416      BEQ    TYPF           ;BRANCH IF"%".
948 003400 122737 000043 003462      CMPB   #43,TYPDAT     ;NOT"%".CHECK FOR"#".
949 003406 001417      BEQ    TYPG           ;BRANCH IF "#".
950 003410 004737 003416      JSR    %7,TYPD        ;TYPE CHAR IN TYPDAT
951 003414 000756      BR     TYPA
```

```
952 003416 113777 003462 176012 TYPD: MOV B TYPDAT,@TPB ;OUTPUT CHARACTER TO PRINTER
953 003424 105777 176004 TST B @TPS ;WAIT FOR DONE FLAG.
954 003430 100375 BPL -4
955 003432 000207 RTS ;EXIT
956 003434 112737 000015 003462 TYPF: MOV B #15,TYPDAT ;MOVE CARRIAGE RETURN CODE TO TYPDAT
957 003442 004737 003416 JSR #7,TYPD ;GO TYPE CHAR.
958 003446 112737 000012 003462 TYPG: MOV B #12,TYPDAT ;MOVE LF CODE TO TYPDAT.
959 003454 004737 003416 JSR #7,TYPD ;GO TYPE CHAR.
960 003460 000734 BR TYPA
961 003462 000000 TYPDAT: OPEN
962 ;
963 ;
964 003464 011600 ;SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER
965 003466 062716 000002 TYP S: MOV @%G,%O ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
966 003472 011037 003512 ADD #2,@%G ;UPDATE TO NEXT MESSAGE ADDRESS
967 003476 022737 177777 003512 MOV @%0,TYP S B ;ADDRESS OF MESSAGE TO TYP S B
968 003504 001001 CMP #-1,TYP S B ;CHECK FOR TERMINATOR
969 003506 000002 BNE TYP S A ;BRANCH IF NOT TERMINATOR.
970 003510 104000 RTI ;TERMINATOR, EXIT
971 003512 000000 TYP S A: TYPE ;CALL ON TYP SUB TO TYPE MESSAGE
972 003514 000763 TYP S B: OPEN ;ADDRESS OF MESSAGE GOES HERE
973 BR TYP S ;GO PROCESS NEXT MESSAGE
974 ;
975 003516 012701 000300 ;OVERLAY VECTOR AREA
976 003522 012702 000302 OVRLAY: MOV #300,%1 ;GET DL11-E VECTOR BASE ADDRESS
977 003526 012703 000004 MOV #302,%2
978 003532 010221 OVRLY A: MOV #4,%3
979 003534 010321 MOV #2,(1)+ ;LOAD VECTOR WITH IOT ERROR TRAP
980 003536 062702 000004 ADD #3,(1)+
981 003542 020127 001000 CMP #4,%2
982 003546 001401 BEQ OVRLY B ;ALL VECTORS BEEN LOADED
983 003550 000770 BR OVRLY B
984 003552 000207 OVRLY B: RTS 7 ;EXIT
985
986 ;SUBROUTINE TO READ OCTAL DATA FROM THE TELETYPE PRINTER
987 003554 011646 RDOCT: MOV (SP),-(SP) ;MAKE ROOM FOR DATA WORD
988 003556 010046 MOV %0,-(SP) ;SAVE R0
989 003560 010146 MOV %1,-(SP) ;SAVE R1
990 003562 005001 INDAT: CLR %1 ;CLEAR DATA WORD
991 003564 005037 CLR COUNT ;SET NO. OF DIGITS = 0
992 003570 105777 175634 RDDAT: TST B @TKS ;TEST TTY READ STATUS
993 003574 100375 BPL RDDAT ;WAIT
994 003576 117746 175630 MOV B @TKB,-(SP) ;PUSH DIGIT ON STACK
995 003602 042716 177600 BIC #177600,(SP) ;:++G
996 003606 105777 175622 ECDAT: TST B @TPS ;TEST TTY PRINT STATUS
997 003612 100375 BPL ECDAT ;WAIT
998 003614 111677 175616 MOV B (SP),@TPB ;ECHO CHARACTER
999 003620 122716 000015 CMP B #15,(SP) ;IS IT A TERMINATOR?
1000 003624 001432 BEQ RETRN ;BR IF YES
1001 003626 122716 000177 CMP B #177,(SP) ;IS IT A RUBOUT?
1002 003632 001423 BEQ RREAD ;BR IF YES
1003 003634 122716 000060 CMP B #60,(SP) ;IS IT AN OCTAL DIGIT?
1004 003640 003020 BGT RREAD ;BR IF NO
1005 003642 122716 000067 CMP B #67,(SP) ;TEST AGAIN
1006 003646 002415 BLT RREAD ;BR IF NO
1007 003650 005237 001624 INC COUNT ;INC NO. OF DIGITS
```

```
1008 003654 022737 000007 001624 CMP #7,COUNT ;MORE THAN SIX DIGITS? ;:++G
1009 003662 003407 BLE RREAD ;BR IF YES
1010 003664 006301 ASL %1 ;CLEAR LOWEST THREE BITS
1011 003666 006301 ASL %1 ;OF DATA WORD
1012 003670 006301 ASL %1
1013 003672 162716 000060 SUB #60,(SP) ;CONVERT TO BINARY
1014 003676 062601 ADD (SP)+,%1 ;ADD DIGIT TO DATA WORD
1015 003700 007233 BR RDDAT ;GET NEXT DIGIT
1016 003702 104000 RREAD: TYPE ;TELL USER ABOUT ILLEGAL CHARACTER
1017 003704 017253 DTERR
1018 003706 005726 TST (SP)+ ;GET RID OF ILLEGAL CHARACTER
1019 003710 000724 BR INDAT ;START SUBROUTINE AGAIN
1020 003712 010166 000010 RETRN: MOV %1,10(SP) ;STORE DATA WORD ON STACK
1021 003716 005726 TST (SP)+ ;INC STACK POINTER
1022 003720 012601 MOV (SP)+,%1 ;RESTORE R1
1023 003722 012600 MOV (SP)+,%0 ;RESTORE R0
1024 003724 000207 RTS PC ;RETURN
1025
1026 ;SUBROUTINE TO DELAY A SPECIFIED NUMBER OF MILLISECONDS
1027 003726 011637 003772 DLY: MOV @%6,DLCNT ;GET DELAY COUNT ADDRESS.
1028 003732 062716 000002 ADD #2,@%6 ;SET UP EXIT ADDRESS
1029 003736 017746 000030 MOV @DLCNT,-(6) ;DELAY COUNT TO STACK
1030 003742 001411 BEQ DLYC
1031 003744 005037 177776 CLR PSW ;SET PRIORITY 0
1032 003750 012746 000226 DLY A: MOV #226,-(6) ;1 MSEC COUNT TO STACK
1033 003754 005316 DLY B: DEC @%6 ;DECREMENT 1 MSEC COUNT
1034 003756 001376 BNE DLY B ;BRANCH IF NOT 0.
1035 003760 005726 POPSP ;ZERO. UNCOVER MSECS. COUNT.
1036 003762 005316 DEC @%6 ;DECREMENT IT
1037 003764 001371 BNE DLY A ;BR IF NOT DONE DELAYING
1038 003766 005726 DLY C: POPSP ;DONE
1039 003770 000002 RTI ;EXIT.
1040 003772 000000 DLCNT: OPEN ;CONTAINS MILLISECONDS COUNT ADDRESS.
1041 ;
1042 ;SUBROUTINE TO STALL A RANDOM NUMBER OF MILLISECONDS. MAXIMUM STALL
1043 ;DETERMINED BY CONTENTS OF LOC STLMSK.
1044 003774 004737 003226 STAL: JSR %7,RNGEN ;GO GET RANDOM NUMBER.
1045 040000 043700 001406 BIC STLMSK,%0 ;# IN RO. APPLY STALL MASK.
1046 004004 001404 BEQ STAL B ;BRANCH IF RESULT IS 0.
1047 004006 010037 004014 MOV %0,STAL A
1048 004012 104016 DELAY ;DELAY
1049 004014 000000 STAL A: OPEN ;DELAY COUNT
1050 004016 000002 STAL B: RTI ;DONE. EXIT.
1051 ;
1052 ;SUBROUTINE TO GENERATE RANDOM CHARACTER COUNT
1053 004020 004737 003226 GRCNT: JSR %7,RNGEN ;GET RANDOM NUMBER
1054 004024 043700 004040 BIC RCMSK,%0 ;APPLY MASK
1055 004030 001773 BEQ GRCNT ;TRY AGAIN IF RESULT 0
1056 004032 010037 004042 MOV %0,RNCNT ;COUNT TO RNCNT
1057 004036 000207 RTS %7 ;EXIT.
1058 004040 000000 RCMSK: OPEN ;RANDOM CHARACTER MASK.
1059 004042 000000 RNCNT: OPEN ;RANDOM CHARACTER COUNT.
1060 ;
1061 ;SUBROUTINE TO SKIP ON FLAG AND TIME OUT IF SKIP FAILS
1062 004044 013737 001410 004112 TMRX: MOV RXCSR,SLOT ;SET UP RXCSR ADDRESS
1063 004052 000403 BR TIME1
```

```
1064 004054 013737 001414 004112  TMX:  MOV  TXCSR,SIOT  ;SET UP TXCSR ADDRESS
1065 004062 005037 004110          TIME1: CLR  TIMER
1066 004066 005237 004110          TIME2: INC  TIMER
1067 004072 001405                    BEQ  TIMEX  ;BRANCH IF COUNTER OVERFLOW
1068 004074 105777 000012          TSTB @SIOT
1069 004100 100372                    BPL  TIME2
1070 004102 062716 000002          ADD  #2,%6  ;SET UP EXIT RETURN
1071 004106 000002          TIMEX: RTI
1072 004110 000000          TIMER: 0
1073 004112 000000          SIOT: 0
1074
1075          ;SUBROUTINE TO SELECT LINE
1076 004114 032777 010000 174052  LINSEL: BIT  #BIT12,@SRPTR
1077 004122 001003                    BNE  LINS LX ;BRANCH IF SET
1078 004124 005037 001612          CLR  FOUNDV
1079 004130 000205                    RTS  5
1080 004132 004737 003516          LINS LX: JSR %7,OVRLAY
1081 004136 004737 003300          JSR  %7,CLRCD
1082 004142 104000                    TYPE
1083 004144 016657                    LDLINE
1084 004146 004737 003554          JSR  PC,RDOCT
1085 004152 012637 001616          MOV  (SP)+,TLMP
1086 004156 042737 177740 001616  BIC  #177740,TEMP
```

```
1087 004164 013737 001616 001614  MOV  TEMP,LINENO ;SAVE FOR TYPING
1088 004172 006337 001616          ASL  TEMP
```

```
1089 004176 013701 001616      MOV     TEMP,%1
1090 004202 016101 001200      MOV     RXCR0(1),%1      ;GET RXCSR DEVICE ADDRESS
1091 004206 032701 000001      BIT     #BIT0,%1        ;IS DEVICE THERE
1092 004212 001403              BEQ     LINB             ;YES
1093 004214 104000              LINA:   TYPE            ;NO, REPORT
1094 004216 017210              MNOLIN
1095 004220 000744              BR     LINS LX
1096 004222 004737 006072      LINB:   JSR     %7,FORMAD
1097 004226 005037 177776      CLR     PSW
1098 004232 052737 000001 001472  BIS     #BIT0,FMAP      ;SET MAPPING FLAG
1099 004240 042777 000100 175146  BIC     #BIT6,@TXCSR
1100 004246 052777 000100 175140  BIS     #BIT6,@TXCSR
1101 004254 000240              NOP
1102 004256 000240              TST
1103 004260 005737 001420      TST     RXVTR
1104 004264 001753              BEQ     LINA
1105 004266 042777 000100 175120  BIC     #BIT6,@TXCSR
1106 004274 012737 000340 177776  MOV     #PRTY7,PSW
1107 004302 004537 004474      JSR     5,OACNV        ;TYPE LINE #
1108 004306 001614              LINENO
1109 004310 016716              SELINE
1110 004312 000002              2
1111 004314 104000              TYPE
1112 004316 016705              ALINE
1113 004320 000205              RTS     5
1114
1115      ;
1116 004322 012737 177777 004344  INBIN:  MOV     #-1,RIND      ;SET ALL VARIABLES
1117 004330 004537 004562      JSR     %5,BMOVE       ;TO MINUS 1.
1118 004334 004344              RIND
1119 004336 004345              RIND+1
1120 004340 000013              11.
1121 004342 000207              RTS     %7             ;EXIT
1122 004344 000000      RIND:   OPEN
1123 004346 000000      PT0:    OPEN
1124 004350 000000      PT1:    OPEN
1125 004352 000000      PIND:   OPEN
1126 004354 000000      PTOP:   OPEN
1127 004356 000000      PT1P:   OPEN
1128
1129      ;
1130 004360 013737 004346 004350  GTBIN:  MOV     PTO,PT1      ;PREVIOUS BIN CHAR TO PT1
1131 004366 005137 004350      CDM     PT1
1132 004372 005137 004344      CDM     RIND
1133 004376 001002              BNE     .+6
1134 004400 005237 004350      INC     PT1
1135 004404 042737 177400 004350  BIC     #177400,PT1     ;MASK TO 8 BITS
1136 004412 013737 004350 004346  MOV     PT1,PT0        ;SAVE BIN CHAR IN PTO
1137 004420 013700 004350      MOV     PT1,%0         ;BIN CHAR TO R0.
1138 004424 000207              RTS     %7             ;EXIT.
1139 004426 013737 004354 004356  GTBINP: MOV     PTOP,PT1P     ;PREVIOUS BIN CHAR TO PT1P
1140 004434 005137 004356      CDM     PT1P
1141 004440 005137 004352      CDM     PIND
1142 004444 001002              BNE     .+6
1143 004446 005237 004356 004356  INC     PT1P
1144 004452 042737 177400      BIC     #177400,PT1P   ;MASK TO 8 BITS.
```

```
1145 004460 013737 004356 004354  MOV     PT1P,PTOP      ;SAVE BIN CHAR IN PTO.
1146 004466 013701 004356      MOV     PT1P,%1       ;BIN CHAR TO R1.
1147 004472 000207              RTS     %7             ;EXIT.
1148
1149      ;
1150 004474 104013              OACNV:  SAVREG
1151 004476 013537 004560      MOV     @(5)+,OACNVX   ;GET OCTAL VALUE.
1152 004502 012501              MOV     (5)+,%1        ;GET DESTINATION ADDR.
1153 004504 012502              MOV     (5)+,%2        ;GET CONVERT COUNT.
1154 004506 060201              ADD     %2,%1          ;DEVELOP ADDR TO STORE 1ST CHAR.
1155 004510 013703 004560      OACNVA: MOV     OACNVX,%3
1156 004514 042703 177770      BIC     #177770,%3     ;ISOLATE LEAST SIGNIFICANT DIGIT.
1157 004520 062703 000060      ADD     #60,%3         ;CONVERT DIGIT TO ASCII.
1158 004524 110341              MOV     %3,-(1)        ;STORE ASCII CHARACTER.
1159 004526 042737 000007 004560  BIC     #7,OACNVX
1160 004534 006037 004560      ROR     OACNVX
1161 004540 006037 004560      ROR     OACNVX
1162 004544 006037 004560      ROR     OACNVX
1163 004550 005302              DEC     %2             ;DONE ALL DIGITS?
1164 004552 001356              BNE     OACNVA         ;BRANCH IF NOT DONE.
1165 004554 104014              RSTREG
1166 004556 000205              RTS     %5             ;DONE. EXIT.
1167 004560 000000      OACNVX: OPEN
1168
1169      ;
1170 004562 104013              BMOVE:  SAVREG
1171 004564 012501              MOV     (5)+,%1        ;SAVE REGS.
1172 004566 012502              MOV     (5)+,%2        ;GET"FROM"ADDRESS
1173 004570 012503              MOV     (5)+,%3        ;GET"TO"ADDRESS
1174 004572 112122              BMOVA:  MOV     (1)+,(2)+   ;GET COUNT
1175 004574 005303              DEC     %3             ;MOVE BYTE
1176 004576 001375              BNE     BMDVA         ;DECREMENT COUNT
1177 004600 104014              RSTREG              ;BRANCH IF NOT DONE.
1178 004602 000205              RTS     %5             ;RESTORE REGS.
1179
1180      ;
1181 004604 104013              BDCNV:  SAVREG
1182 004606 012700 004762      MOV     #DECVAL,%0     ;SET UP ADDR TO STORE DECIMAL ASCII IN R0
1183 004612 013501              MOV     @(5)+,%1        ;BINARY VALUE TO R1.
1184 004614 012537 004672      MOV     (5)+,BDCNVC    ;GET DEST ADDR
1185 004620 012537 004674      MOV     (5)+,BDCNVL    ;GET CHAR COUNT
1186 004624 012702 004750      MOV     #ADTENP,%2     ;ADDR OF TEN POWER STRING TO R2.
1187 004630 012737 000005 004742  MOV     #5,CNVCTR      ;SET UP FOR 5 POWER CONVERSIONS.
1188 004636 012237 004746      BDCNVA: MOV     (2)+,TENPWR   ;MOVE POWER OF TEN VALUE TO TENPWR.
1189 004642 004737 004702      JSR     %7,SUBTEN      ;PERFORM CONVERSION
1190 004646 005337 004742      DEC     CNVCTR         ;MOVE POWER OF TEN VALUE TO TENPWR.
1191 004652 001371              BNE     BDCNVA         ;DONE 5 CONVERSIONS?
1192 004654 163700 004674      SUB     BDCNV,%0       ;BRANCH IF NOT YET 5.
1193 004660 010037 004670      MOV     %0,BDCNV
1194 004664 004537 004562      JSR     %5,BMOVE
1195 004670 000000      BDCNVB: 0
1196 004672 000000      BDCNVC: 0
1197 004674 000000      BDCNVD: 0
1198 004676 104014              RSTREG
1199 004700 000205              RTS     %5             ;YES, EXIT.
1200 004702 005037 004744      SUBTEN: CLR     DIGIT   ;CLEAR DIGIT
```

```
1201 004706 163701 004746 SUBTNA: SUB TENPWR,%1 ;SUBTRACT TEN POWER FROM BINARY VALUE.
1202 004712 103403 BCS SUBTNB ;BRANCH IF UNSUCCESSFUL SUBTRACTION.
1203 004714 005237 004744 INC DIGIT
1204 004720 000772 BR SUBTNA
1205 004722 063701 004746 SUBTNB: ADD TENPWR,%1 ;RESTORE SUBTRACTED VALUE.
1206 004726 062737 000060 004744 ADD ;CONVERT (DIGIT) TO ASCII
1207 004734 113720 004744 MOV# #60,DIGIT ;MOVE ASCII CHAR TO DECVAL FIELD.
1208 004740 000207 RTS ;EXIT.
1209 004742 000000 CNVCTR: OPEN
1210 004744 000000 DIGIT: OPEN
1211 004746 000000 TENPWR: OPEN
1212 004750 023420 ADTENP: 10000.
1213 004752 001750 1000.
1214 004754 000144 100.
1215 004756 000012 10.
1216 004760 000001 1
1217 004762 040 040 DECVAL: .BYTE 040,040,040,040,040,040
1218 004765 040 0.0 040
1219 004770 042777 000002 174412 DATTST: BIC #BIT1,@RXCSR ;CLEAR DATA TERM. READY
1220 004776 052777 000004 174410 BIS #BIT2,@TXCSR ;SET MAINTENANCE BIT
1221 005004 012737 000144 001564 MOV #100.,CTRO ;GET CHARACTER COUNT
1222 005012 105777 174376 DATA: TSTB @TXCSR ;WAIT FOR
1223 005016 100375 BPL .-4 ;READY FLAG
1224 005020 004737 004426 JSR 7,GTBINP ;GET CHARACTER
1225 005024 110137 001560 MOV# %1,CRBUFA ;MOVE CHARACTER
1226 005030 004737 005374 JSR 7,MASKIT ;MASK OFF NON TRANSMITTED BITS
1227 005034 110177 174356 MOV# %1,@TXBUF ;TRANSMIT CHARACTER
1228 005040 105777 174344 TSTB @RXCSR ;WAIT FOR
1229 005044 100375 BPL .-4 ;DONE FLAG
1230 005046 117737 174340 001556 MOV# @RXBUF,CRBUF ;GET RECEIVED CHARACTER
1231 005054 104004 DATCHK ;CHK DATA
1232 005056 005337 001564 DEC CTRO ;DECREMENT CHARACTER COUNT
1233 005062 001353 BNE DATAA
1234 005064 005726 TST (6)+ ;POP STACK
1235 005066 104012 SCOPE
;
1237 005070 104000 ;SETSR: TYPE ;TYPE SELECT OPTION MESSAGE.
1238 005072 016117 ASETSR
1239 005074 000000 HALT ;COMMON HALT.
1240 005076 000207 RTS ;EXIT.
1241 005100 104000 INCRTN: TYPE ;TYPE INCORRECT ROUTINE SELECTED.
1242 005102 016216 AINCR
1243 005104 000000 HALT ;COMMON HALT.
1244 005106 000207 RTS ;EXIT.
1245 005110 104000 INCRPG: TYPE
1246 005112 016337 AINCPG
1247 005114 000000 HALT
1248 005116 000137 002044 JMP START
1249 005122 005037 001612 PRGEND: CLR FOUNDV
1250 005126 032777 020000 173040 BIT #BIT13,@SRPTR ;INHIBIT PRINT SET?
1251 005134 001026 BNE PRGEXT ;BR IF SET
1252 005136 004537 004604 JSR %5,BDCNV
1253 005142 001636 PASCNT
1254 005144 016407 APCNT
1255 005146 000006 6
1256 005150 004537 004474 JSR %5,DACNV ;CONVERT LINE NUMBER
```

```
1257 005154 001614 LINENO
1258 005156 016427 ACLIN
1259 005160 000002 2
1260 005162 004537 004474 JSR %5,DACNV ;CONVERT RXCSR
1261 005166 001410 RXCSR
1262 005170 016443 APRXC
1263 005172 000006 6
1264 005174 004537 004474 JSR %5,DACNV ;CONVERT VECTOR
1265 005200 001420 RXVTR
1266 005202 016464 APVEC
1267 005204 000004 4
1268 005206 104000 TYPE PROGRAM END.
1269 005210 016372 APGEND
1270 005212 032777 010000 172754 PRGEXT: BIT #BIT12,@SRPTR ;LOCK ON LINE
1271 005220 001403 BEQ PRGXT1 ;BR IF NOT SET
1272 005222 005237 001636 INC PASCNT
1273 005226 000425 BR PRGXTL
1274 005230 013737 001614 001616 PRGXT1: MOV LINENO,TEMP ;GET LINENO
1275 005236 006337 001616 ASL TEMP
1276 005242 062737 000002 001616 PRGEC: ADD #2,TEMP ;UPDATE LINE NUMBER
1277 005250 013701 001616 PRGEA: MOV TEMP,%1
1278 005254 016101 001200 MOV RXCR0(1),%1 ;GET RXCSR DEVICE ADDRESS
1279 005260 022701 177777 CMP #177777,%1 ;LAST ONE
1280 005264 001023 BNE PRGEB ;NO,CONTINUE
1281 005266 005237 001636 INC PASCNT
1282 005272 005037 001614 CLR LINENO
1283 005276 005037 001616 CLR TEMP
1284 005302 013705 000042 PRGXTL: MOV @#42,%5
1285 005306 001405 BEQ CONT
1286 005310 000005 RESET
1287 005312 004715 LOGIC: JSR 7,(5)
1288 005314 000240 NOP
1289 005316 000240 NOP
1290 005320 000240 NOP
1291 005322 032777 010000 172644 CONT: BIT #BIT12,@SRPTR ;LOCK ON LINE
1292 005330 001747 BEQ PRGEA ;BRANCH IF NOT SET
1293 005332 000207 RTS 7
1294 005334 032701 000001 PRGEB: BIT #BIT0,%1 ;DEVICE THERE
1295 005340 001340 BNE PRGEC ;NO
1296 005342 006237 001616 ASR TEMP
1297 005346 013737 001616 001614 MOV TEMP,LINENO
1298 005354 004737 006072 JSR %7,FORMAD
1299 005360 000207 RTS %7 ;EXIT.
;
1301 ;CONDITIONAL ERROR HALT ROUTINE.
1302 005362 005777 172606 EHLT: TST @SRPTR ;CHECK FOR HALT ON ERROR.
1303 005366 100001 BPL EHLTA ;BRANCH IF NO HALT DESIRED.
1304 005370 000000 HALT ;HALT.
1305 005372 000002 EHLTA: RTI ;IN DATA LIGHTS.
;
1307 ;MASKIT - MASK DATA ACCORDING TO LINE NUMBER
1308 005374 013737 001402 001404 MASKIT: MOV UMASK,RMASK ;GET MASK
1309 005402 042737 177000 0014C3 BIC #177000,RMASK ;REMOVE C/D FLAG+PRIORITY
1310 005410 005137 001404 COM RMASK
1311 005414 043737 001404 001560 BIC RMASK,CRBUFA ;MASK DESIRED BITS
1312 005422 000207 RTS 7
```



```

1313 ;
1314 ;
1315 005424 017737 173762 001562 DTCHK: MOV @RXBUF,CRBUF  ;DID ANY ERROR BITS SET
1316 005432 032737 170000 001562 BIT #170000,CRBUF
1317 005440 001004 BNE DTCHKX ;YES, TYPE ERROR
1318 005442 023737 001556 001560 CMP CRBUF,CRBUFA ;COMPARE EXPECTED AND RECEIVED
1319 005450 001421 BEQ DTCHKX ;CHARS. BRANCH IF SAME.
1320 005452 004537 004474 DTCHKX: JSR %5,OACNV ;GO TO OCTAL TO ASCII CONVERT.
1321 005456 001566 CRBUF ;SOURCE ADDR.
1322 005460 016074 AWAS ;DESTINATION ADDR.
1323 005462 000003 3 ;#OF DIGITS TO CONVERT.
1324 005464 004537 004474 JSR %5,OACNV ;GO TO OCTAL TO ASCII CONVERT.
1325 005470 001560 CRBUFA ;SOURCE ADDR.
1326 005472 016063 AASB ;DESTINATION ADDR.
1327 005474 000003 3 ;#OF DIGITS TO CONVERT.
1328 005476 004537 004474 JSR %5,OACNV
1329 005502 001562 CRBUF
1330 005504 016107 ARXBUF
1331 005506 000006 6
1332 005510 104015 ERROR1
1333 005512 016051 ERDAT
1334 005514 000002 DTCHKA: RTI
1335 ;
1336 ;
1337 005516 012737 177777 005666 ;ERROR HANDLER
1338 005524 012737 000240 005670 ERR: MOV #-1,ERRB ;SET UP ONE MESSAGE CALL.
1339 005532 005037 005704 MOV #240,ERRB+2
1340 005536 000413 CLR ERRE
1341 005540 011637 005666 BR ERRA
1342 005544 017737 000116 005666 ERR1: MOV @%6,ERRB ;DEVELOP ADDT'L MESSAGE ADDR.
1343 005552 012737 177777 005670 MOV @ERRB,ERRB ;STORE AT ERRB.
1344 005560 012737 000002 005704 MOV #-1,ERRB+2
1345 005566 032777 020000 172400 ERR2: MOV #2,ERRE
1346 005574 001036 BIT #BIT13,@SRPTR ;INHIBIT ERROR PRINT?
1347 005576 011637 005702 BNE ERRC ;BRANCH TO INHIBIT PRINT.
1348 005602 162737 000002 005702 MOV @%6,ERRD ;DEVELOP CALLING ADDR.
1349 005610 013737 001456 001460 SUB #2,ERRD
1350 005616 042737 100000 001460 MOV RTNNO,TNNO
1351 005624 004537 004474 BIC #BIT15,TNNO
1352 005630 005702 JSR %5,OACNV ;GO TO OCTAL TO ASCII CONVERT.
1353 005632 015240 ERRD ;SOURCE ADDR.
1354 005634 000006 APC ;DESTINATION ADDR.
1355 005636 004537 004474 6 ;#OF DIGITS TO CONVERT.
1356 005642 001410 JSR %5,OACNV ;GO TO OCTAL TO ASCII CONVERT.
1357 005644 015257 RXCSR ;SOURCE ADDR.
1358 005646 000006 MRXNUM ;DESTINATION ADDR.
1359 005650 004537 004474 6 ;#OF DIGITS TO CONVERT.
1360 005654 001456 JSR %5,OACNV ;GO TO OCTAL TO ASCII CONVERT.
1361 005656 015230 RTNNO ;SOURCE ADDR.
1362 005660 000003 ATNUMB ;DESTINATION ADDR.
1363 005662 104001 3 ;#OF DIGITS TO CONVERT.
1364 005664 015226 TYPES ;TYPE:
1365 005666 000000 EMO ;ERROR HEADER,
1366 005670 177777 -1 ;ADDT'L ERROR MESSAGE IF ANY.
1367 005672 104010 ERRB: OPEN
1368 005674 063716 005704 ERRC: EHALT ;GO ERR HALT IF DESIRED.
ADD ERRE,%6

```

```

1369 005700 000002 RTI ;EXIT.
1370 005702 000000 ERRD: OPEN
1371 005704 000000 ERRE: OPEN
1372 ;
1373 ;
1374 005706 013737 177776 001470 ;ERROR TRAP HANDLER - TYPE TO AND FROM WHERE ERROR TRAP OCCURRED
1375 005714 012737 000340 177776 ETRP: MOV PSW,OLDPS ;SAVE OLD STATUS
1376 005722 006237 001470 MOV #PRTY7,PSW
1377 005726 006237 001470 ASR OLDPS
1378 005732 006237 001470 ASR OLDPS
1379 005736 042737 177740 001470 ASR OLDPS
1380 005744 013737 001470 001632 BIC #177740,OLDPS
1381 005752 011637 001634 MOV OLDPS,TOPC
1382 005756 004537 004474 ERTPA: JSR %5,OACNV ;GET FROM PC
1383 005762 001632 TOPC
1384 005764 017145 MTO
1385 005766 000006 6
1386 005770 004537 004474 JSR %5,OACNV
1387 005774 001634 FROMPC
1388 005776 017177 MFROM
1389 006000 000006 6
1390 006002 104000 TYPE
1391 006004 017100 MTERR
1392 006006 000000 HALT
1393 006010 000137 002044 JMP START
1394 ;
1395 ;
1396 006014 011637 001632 ;MAPVEC - MAP VECTOR OR REPORT ERROR DEPENDING ON FMAP FLAG
1397 006020 022626 MAPVEC: MOV @%6,TOPC
1398 006022 011637 001634 POPSP2
1399 006026 162737 000004 001632 MOV @%6,FROMPC
1400 006034 005737 001472 SUB #4,TOPC
1401 006040 001746 TST FMAP
1402 006042 013737 001632 001424 BEQ ERTPA ;NOT MAPPING, REPORT ERROR
1403 006050 162737 000004 001632 MOV TOPC,TXVTR ;STORE VECTOR
1404 006056 013737 001632 001420 SUB #4,TOPC
1405 006064 005037 001472 MOV TOPC,RXVTR
1406 006070 000002 CLR FMAP
1407 RTI
1408 ;
1409 006072 010137 001410 ;FORMAD - FORM DEVICE AT ADDRESSES
1410 006076 062701 000002 FORMAD: MOV %1,RXCSR
1411 006102 010137 001412 ADD #2,%1
1412 006106 062701 000002 MOV %1,RXBUF
1413 006112 010137 001414 ADD #2,%1
1414 006116 062701 000002 MOV %1,TXCSR
1415 006122 010137 001416 ADD #2,%1
1416 006126 013737 001614 001616 MOV %1,TXBUF ;GET PRIORITY
1417 006134 006337 001616 ASL TEMP
1418 006140 062737 001302 001616 ADD #CMASO,TEMP
1419 006146 017737 173444 001620 MOV @TEMP,TEMP1
1420 006154 013737 001620 001400 MOV TEMP1,UMASK
1421 006162 000337 001620 SWAB TEMP1
1422 006166 006337 001620 ASL TEMP1
1423 006172 042737 177437 001620 BIC #177437,TEMP1
1424 006200 013737 001620 001422 MOV TEMP1,RXLVL

```

```
1425 005208 013737 001620 001426      MOV     TEMP1, TXLVL
1426 006214 000207                    RTS     %7
1427
1428
1429
1430 006216 032777 001000 171750      DOTHIS: BIT     #BIT9, @SRPTR ;IS SELECT TEST SWITCH SET
1431 006224 001002                    BNE     GOBACK ;RETURN TO TEST IF SW SET
1432 006226 000137 002410                    JMP     GTRDYX ;GO TO NEXT TEST
1433 006232 000207                    GOBACK: RTS     %7
1434
1435 006234 012737 006244 000024      PFAIL: MOV     #PWRUP, %24
1436 006242 000000                    HALT
1437 006244 012737 006234 000024      PWRUP: MOV     #PFAIL, %24
1438 006252 000005                    RESET
1439 006254 012706 001176                    MOV     #SPBOT, %6
1440 006260 104000                    TYPE
1441 006262 017374                    MPWRF
1442 006264 104003                    ERROR
1443 006266 000452                    BR      RESTART
1444
1445
1446 006270 022737 000000 001612      ;DECIDE IF VECTOR TO BE MAPPED AND MAP
1447 006276 001045                    TSTVEC: CMP    #0, FOUNDV ;NEED VECTOR MAPPING
1448 006300 004737 003516                    BNE     TSTVEX ;NO, EXIT
1449 006304 005037 001420                    JSR     %7, OVRLAY
1450 006310 005037 177776                    CLR     RXVTR
1451 006314 052737 000001 001472                    CLR     PSW
1452 006322 042777 000100 173064                    BIS     #BIT0, FMAP ;SET MAPPING FLAG
1453 006330 052777 000100 173056                    BIC     #BIT6, @TXCSR ;CAUSE INTERRUPT
1454 006336 000240                    NOP
1455 006340 000240                    NOP
1456 006342 005737 001420                    TST     RXVTR ;DID TRAP OCCUR?
1457 006346 001011                    BNE     TSTVA ;YES, OK
1458 006350 032777 020000 171616                    BIT     #BIT13, @SRPTR
1459 006355 001344                    BNE     TSTVEC ;NO, ERROR
1460 006360 104000                    TYPE
1461 006362 017256                    INTER
1462 006364 104003                    ERROR
1463 006366 000137 006270                    JMP     TSTVEC
1464 006372 042777 000100 173014      TSTVA: BIC     #BIT6, @TXCSR
1465 006400 012737 000340 177776                    MOV     #PRTY7, PSW ;RAISE PRIORITY, RETURN
1466 006406 005237 001612                    INC     FOUNDV
1467 006412 000207                    TSTVEX: RTS    %7
1468
1469
1470 006414 013700 001450                    ;RESTART ROUTINE
1471 006420 006300                    RESTART: MOV   PRGNUM, %0
1472 006422 000170 006426                    ASL     %0
1473
1474 006426 006500                    JMP     @RSTART(0) ;GO RESTART SELECTED PROGRAM
1475 006430 014546                    RSTART: PRGOA ;PROGRAM 0 RESTART ADDRESS
1476 006432 014620                    PRG1A ;PROGRAM 1 RESTART ADDRESS
1477 006434 014716                    PRG2A ;PROGRAM 2 RESTART ADDRESS
1478 006436 014746                    PRG3A ;PROGRAM 3 RESTART ADDRESS
1479 006440 005110                    PRG4A ;PROGRAM 4 RESTART ADDRESS
1480 006442 005110                    INCRPG
1481
1482
1483
1484
1485
```

```
1481 006444 005110                    INCRPG
1482
1483
1484
1485
1486 006446 012737 006504 001452      ;PRGO - INPUT-OUTPUT LOGIC TESTS
1487 006454 005737 000042      PRGO: MOV     #AT0, KSTART
1488 006460 001005                    TST     @#42 ;MONITOR LOAD
1489 006462 104000                    BNE     PRGOB ;YES, START TEST
1490 006464 015270                    TYPE ;TYPE TITLE AND INSTRUCTIONS
1491 006466 000000                    POTIT
1492 006470 000000                    HALT
1493 006474 004537 004114      PRGOB: JSR     7, SETSR
1494 006500 000137 002402      PRGOA: JSR     5, LINSSEL ;GO GET LINE # FROM USER
1495 006500 177777                    JMP     GETRDY ;GET STARTED.
1496 006504                    X=-1
1497 006504                    TSTA 1000., AA, CD
1498
1499 006504 100000                    TSTAA AA, 1000., \X+1+CD, \X+2, \X+1
1500 006506 006536                    ;*****
1501 006510 001750                    AT0: 100000 ;TEST NUMBER
1502 006512 006514                    AT1 ;ADDRESS OF NEXT TEST
1503 006512 000000                    1000. ;ITERATION COUNT
1504
1505 006514 012737 006530 000004      AAA: MOV     #AAE, MACHER ;SET UP MACHINE ERROR TRAP.
1506 006522 005077 172662                    CLR     @RXCSR ;REFERENCE RXCSR
1507 006526 104012                    AAB: SCOPE ;OK IF NO TRAP. SCOPE
1508 006530 022626                    AAE: PDPSP2
1509 006532 104003                    ERRDR ;TRAPPED WHEN REFERENCING RXCSR.
1510 006534 000774                    BR      AAB
1511 006536                    TSTA 1000., AB, CD
1512 006536                    TSTAA AB, 1000., \X+1+CD, \X+2, \X+1
1513 006536
1514
1515 006536 100001                    ;*****
1516 006540 006576                    AT1: 100001 ;TEST NUMBER
1517 006542 001750                    AT2 ;ADDRESS OF NEXT TEST
1518 006544 006546                    1000. ;ITERATION COUNT
1519 006544 000001                    ABA ;SCOPE ENTRY POINT
1520
1521 006546 012737 006570 000004      ;TEST ABILITY TO REFERENCE RECEIVER BUFFER WITHOUT TRAPPING
1522 006554 005737 002016      ABA: MOV     #ABE, MACHER ;SET UP MACHINE ERROR TRAP.
1523 006560 100402                    TST     @#XORFLG
1524 006562 005777 172624                    BMI     ABB
1525 006566 104012                    TST     @RXBUF ;REFERENCE RXBUF
1526 006570 022626                    ABB: SCOPE ;OK IF NJ TRAP SCOPE
1527 006572 104003                    ABE: PDPSP2
1528 006574 000774                    ERRDR ;TRAPPED WHEN REFERENCING RXBUF
1529 006576                    BR      ABB
1530 006576                    TSTA 1000., AC, CD
1531 006576                    TSTAA AC, 1000., \X+1+CD, \X+2, \X+1
1532
1533 006576 100002                    ;*****
1534 006600 006630                    AT2: 100002 ;TEST NUMBER
1535 006602 001750                    AT3 ;ADDRESS OF NEXT TEST
1536 006604 006606                    1000. ;ITERATION COUNT
1537 006604 006606                    ACA ;SCOPE ENTRY POINT
```

```
1537          000002          X=X+1          ;
1538          ;*****
1539          ;TEST ABILITY TO REFERENCE TRANSMITTER CSR WITHOUT TRAPPING.
1540 006606 012737 006622 000004 ACA:  MOV  #ACE,MACHER  ;SET UP MACHINE ERROR TRAP.
1541 006614 005777 172574          TST  @TXCSR      ;REFERENCE TXCSR
1542 006620 104012          ACB:  SCOPE      ;SCOPE
1543 006622 022626          ACE:  POPSP2    ;
1544 006624 104003          ERROR      ;TRAPPED WHEN REFERENCING TXCSR
1545 006626 000774          BR      ACB
1546 006630          TSTA 1000.,AD,CD
1547 006630          TSTAA AD,1000.,\X+1+CD,\X+2,\X+1
1548          ;*****
1549 006630 100003          AT3: 100003    ;TEST NUMBER
1550 006632 006662          AT4          ;ADDRESS OF NEXT TEST
1551 006634 001750          1000.      ;ITERATION COUNT
1552 006636 006640          ADA          ;SCOPE ENTRY POINT
1553          X=X+1          ;
1554          ;*****
1555          ;TEST ABILITY TO REFERENCE TRANSMITTER BUFFER WITHOUT TRAPPING
1556 006640 012737 006654 000004 ADA:  MOV  #ADE,MACHER  ;SET UP MACHINE ERROR TRAP.
1557 006646 005777 172544          TST  @TXBUF     ;REFERENCE TX BUF.
1558 006652 104012          ADB:  SCOPE     ;SCOPE
1559 006654 022626          ADE:  POPSP2    ;
1560 006656 104003          ERROR      ;TRAPPED WHEN REFERENCING TXBUF
1561 006660 000774          BR      ADB
1562 006662          TSTA 10.,AE,CD
1563 006662          TSTAA AE,10.,\X+1+CD,\X+2,\X+1
1564          ;*****
1565 006662 100004          AT4: 100004    ;TEST NUMBER
1566 006664 006762          AT5          ;ADDRESS OF NEXT TEST
1567 006666 000012          10.        ;ITERATION COUNT
1568 006670 006672          AEA          ;SCOPE ENTRY POINT
1569          000004          X=X+1          ;
1570          ;*****
1571          ;TEST THAT TXCSR BIT 0 (BREAK) CAN BE SET AND CLEARED
1572          ;AND THAT RESET CLEARS IT
1573 006672 032777 000001 172514 AEA:  BIT  #BIT0,@TXCSR ;SEE IF BIT IS CLEAR
1574 006700 001402          BEQ  AEB        ;BR IF CLEAR
1575 006702 104003          ERROR      ;RESET DID NOT CLEAR IT
1576 006704 000421          BR      AED
1577 006706 052777 000001 172500 AEB:  BIS  #BIT0,@TXCSR ;SET TXCSR BIT 0
1578 006714 032777 000001 172472          BIT  #BIT0,@TXCSR ;DID IT SET
1579 006722 001002          BNE  AEC        ;YES, GO ON
1580 006724 104003          ERROR      ;TXCSR BIT0 FAILED TO SET
1581 006726 000410          BR      AED
1582 006730 042777 000001 172456 AEC:  BIC  #BIT0,@TXCSR ;CLEAR TXCSR BIT 0
1583 006736 032777 000001 172450          BIT  #BIT0,@TXCSR ;DID IT CLEAR
1584 006744 001401          BEQ  AED        ;
1585 006746 104003          ERROR      ;TXCSR BIT 0 DID NOT CLEAR
1586 006750 052777 000001 172436 AED:  BIS  #BIT0,@TXCSR ;TXCSR BIT 0 DID NOT CLEAR
1587 006756 104011          SRESET      ;ISSUE RESET TO CLEAR
1588 006760 104012          SCOPE
1589 006762          TSTA 10.,AG,CD
1590 006762          TSTAA AG,10.,\X+1+CD,\X+2,\X+1
1591          ;*****
1592 006762 100005          AT5: 100005    ;TEST NUMBER
```

```
1593 006764 007062          AT6          ;ADDRESS OF NEXT TEST
1594 006766 000012          10.        ;ITERATION COUNT
1595 006770 006772          AGA          ;SCOPE ENTRY POINT
1596          000005          X=X+1          ;
1597          ;*****
1598          ;TEST THAT TXCSR BIT2 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
1599 006772 032777 000004 172414 AGA:  BIT  #BIT2,@TXCSR ;SEE IF TXCSR BIT2 IS CLEAR.
1600 007000 001402          BEQ  AGB        ;BRANCH IF BIT IS CLEAR.
1601 007002 104003          ERROR      ;RESET DID NOT CLEAR TXCSR BIT2
1602 007004 000421          BR      AGD
1603 007006 052777 000004 172400 AGB:  BIS  #BIT2,@TXCSR ;SET TXCSR BIT2.
1604 007014 032777 000004 172372          BIT  #BIT2,@TXCSR ;SEE IF BIT IS SET.
1605 007022 001002          BNE  AGC        ;BRANCH IF BIT IS SET.
1606 007024 104003          ERROR      ;TXCSR BIT2 FAILED TO SET.
1607 007026 000410          BR      AGD
1608 007030 042777 000004 172356 AGC:  BIC  #BIT2,@TXCSR ;CLEAR TXCSR BIT2
1609 007036 032777 000004 172350          BIT  #BIT2,@TXCSR ;SEE IF BIT IS CLEAR.
1610 007044 001401          BEQ  AGD        ;
1611 007046 104003          ERROR      ;TXCSR BIT2 FAILED TO CLEAR.
1612 007050 052777 000004 172336 AGD:  BIS  #BIT2,@TXCSR ;SET TXCSR BIT2.
1613 007056 104011          SRESET      ;ISSUE RESET TO CLEAR BIT.
1614 007060 104012          SCOPE
1615 007062          TSTA 10.,AJ,CD
1616 007062          TSTAA AJ,10.,\X+1+CD,\X+2,\X+1
1617          ;*****
1618 007062 100006          AT6: 100006    ;TEST NUMBER
1619 007064 007170          AT7          ;ADDRESS OF NEXT TEST
1620 007066 000012          10.        ;ITERATION COUNT
1621 007070 007072          AJA          ;SCOPE ENTRY POINT
1622          000006          X=X+1          ;
1623          ;*****
1624          ;TEST THAT TXCSR BIT6 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
1625 007072 012737 000340 177776 AJA:  MOV  #PRTY7,PSW  ;SET PRIORITY 7.
1626 007100 032777 000100 172306          BIT  #BIT6,@TXCSR ;SEE IF TXCSR BIT6 IS CLEAR.
1627 007106 001402          BEQ  AJB        ;BRANCH IF BIT IS CLEAR.
1628 007110 104003          ERROR      ;RESET DID NOT CLEAR TXCSR BIT6
1629 007112 000421          BR      AJD
1630 007114 052777 000100 172272 AJB:  BIS  #BIT6,@TXCSR ;SET TXCSR BIT6.
1631 007122 032777 000100 172264          BIT  #BIT6,@TXCSR ;SEE IF BIT IS SET.
1632 007130 001002          BNE  AJC        ;BRANCH IF BIT IS SET.
1633 007132 104003          ERROR      ;TXCSR BIT6 FAILED TO SET.
1634 007134 000410          BR      AJD
1635 007136 042777 000100 172250 AJC:  BIC  #BIT6,@TXCSR ;CLEAR TXCSR BIT6
1636 007144 032777 000100 172242          BIT  #BIT6,@TXCSR ;SEE IF BIT IS CLEAR.
1637 007152 001401          BEQ  AJD        ;
1638 007154 104003          ERROR      ;TXCSR BIT6 FAILED TO CLEAR.
1639 007156 052777 000100 172230 AJD:  BIS  #BIT6,@TXCSR ;SET TXCSR BIT6.
1640 007164 104011          SRESET      ;ISSUE RESET TO CLEAR BIT.
1641 007166 104012          SCOPE
1642 007170          TSTA 100.,AK,CD
1643 007170          TSTAA AK,100.,\X+1+CD,\X+2,\X+1
1644          ;*****
1645 007170 100007          AT7: 100007    ;TEST NUMBER
1646 007172 007214          AT10         ;ADDRESS OF NEXT TEST
1647 007174 000144          100.       ;ITERATION COUNT
1648 007176 007200          AKA          ;SCOPE ENTRY POINT
```

```
1649          000007          ; X=X+1 ; *
1650          ; ***** ;
1651          ;TEST THAT TXCSR BIT 7 (READY BIT) IS SET UPON ENTERING ROUTINE AND *
1652          ;THAT IT CAN BE READ RELIABLY. *
1653 007200 105777 172210 AKA: TSTB @TXCSR ;SEE IF TXCSR BIT 7 IS SET. *
1654 007204 100402          BMT AKB ;BRANCH IF SET. *
1655 007206 104003          ERROR ;TXCSR BIT 7 NOT SET. *
1656 007210 104011          SRESET ;ISSUE RESET TO CLEAR BIT IF ERROR *
1657 007212 104012          AKB: SCOPE ;SCOPE *
1658 007214          TSTA 100.,AL,0 *
1659 007214          TSTAA AL,100.,\X+1+0,\X+2,\X+1 *
1660          ; ***** ;
1661 007214 000010          AT10: 10 ;TEST NUMBER *
1662 007216 007276          AT11 ;ADDRESS OF NEXT TEST *
1663 007220 000144          100. ;ITERATION COUNT *
1664 007222 007224          ALA ;SCOPE ENTRY POINT *
1665          X=X+1 ; *
1666          ; ***** ;
1667          ;TEST THAT RXCSR BIT 1 CAN BE SET + CLEARED *
1668 007224 042777 000002 172156 ALA: BIC #BIT1,@RXCSR ;SET RXCSR BIT1 *
1669 007232 052777 000002 172150 BIS #BIT1,@RXCSR ;SEE IF BIT IS SET *
1670 007240 032777 000002 172142 BIT #BIT1,@RXCSR ;BRANCH IF SET *
1671 007246 001002          BNE ALY ;RXCSR BIT 1 FAILED TO SET *
1672 007250 104003          ERROR *
1673 007252 000410          BR ALZ *
1674 007254 042777 000002 172126 ALY: BIC #BIT1,@RXCSR ;CLEAR RXCSR BIT 1 *
1675 007262 032777 000002 172120 BIT #BIT1,@RXCSR ;SEE IF BIT IS CLEAR *
1676 007270 001401          BEQ ALZ *
1677 007272 104003          ERROR ;RXCSR BIT 1 FAILED TO CLEAR *
1678 007274 104012          ALZ: SCOPE ;SCOPE *
1679 007276          TSTA 10.,AP,0 *
1680 007276          TSTAA AP,10.,\X+1+0,\X+2,\X+1 *
1681          ; ***** ;
1682 007276 000011          AT11: 11 ;TEST NUMBER *
1683 007300 007376          AT12 ;ADDRESS OF NEXT TEST *
1684 007302 000012          10. ;ITERATION COUNT *
1685 007304 007306          APA ;SCOPE ENTRY POINT *
1686          X=X+1 ; *
1687          ; ***** ;
1688          ;TEST THAT RXCSR BIT2 IS CLEAR AND CAN BE READ RELIABLY. *
1689 007306 032777 000004 172074 APA: BIT #BIT2,@RXCSR ;SEE IF RXCSR BIT2 IS CLEAR. *
1690 007314 001402          BEQ APB ;BRANCH IF BIT IS CLEAR. *
1691 007316 104003          ERROR ;RXCSR BIT2 IS NOT CLEAR. *
1692 007320 000421          BR APD *
1693 007322 052777 000004 172060 APB: BIS #BIT2,@RXCSR ;SET RXCSR BIT2 *
1694 007330 032777 000004 172052 BIT #BIT2,@RXCSR ;SEE IF BIT IS SET *
1695 007336 001002          BNE APCX ;BRANCH IF SET *
1696 007340 104003          ERROR ;RXCSR BIT2 FAILED TO SET *
1697 007342 000410          BR APD *
1698 007344 042777 000004 172036 APCX: BIC #BIT2,@RXCSR ;CLEAR RXCSR BIT2 *
1699 007352 032777 000004 172030 BIT #BIT2,@RXCSR ;SEE IF BIT IS CLEAR *
1700 007360 001401          BEQ APD *
1701 007362 104003          ERROR ;RXCSR BIT2 FAILED TO CLEAR *
1702 007364 052777 000004 172016 APD: BIS #BIT2,@RXCSR ;SET BIT *
1703 007372 104011          SRESET ;ISSUE RESET TO CLEAR BIT *
1704 007374 104012          SCOPE
```

```
1705 007376          TSTA 10.,AQ,0 *
1706 007376          TSTAA AQ,10.,\X+1+0,\X+2,\X+1 *
1707          ; ***** ;
1708 007376 000012          AT12: 12 ;TEST NUMBER *
1709 007400 007476          AT13 ;ADDRESS OF NEXT TEST *
1710 007402 000012          10. ;ITERATION COUNT *
1711 007404 007406          AQA ;SCOPE ENTRY POINT *
1712          X=X+1 ; *
1713          ; ***** ;
1714          ;TEST THAT RXCSR BIT3 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT. *
1715 007406 032777 000010 171774 AQA: BIT #BIT3,@RXCSR ;SEE IF RXCSR BIT3 IS CLEAR. *
1716 007414 001402          BEQ AQB ;BRANCH IF BIT IS CLEAR. *
1717 007416 104003          ERROR ;RESET DID NOT CLEAR RXCSR BIT3 *
1718 007420 000421          BR AQB *
1719 007422 052777 000010 171760 AQB: BIS #BIT3,@RXCSR ;SET RXCSR BIT3. *
1720 007430 032777 000010 171752 BIT #BIT3,@RXCSR ;SEE IF BIT IS SET. *
1721 007436 001002          BNE AQC ;BRANCH IF BIT IS SET. *
1722 007440 104003          ERROR ;RXCSR BIT3 FAILED TO SET. *
1723 007442 000410          BR AQB *
1724 007444 042777 000010 171736 AQC: BIC #BIT3,@RXCSR ;CLEAR RXCSR BIT3 *
1725 007452 032777 000010 171730 BIT #BIT3,@RXCSR ;SEE IF BIT IS CLEAR. *
1726 007460 001401          BEQ AQB *
1727 007462 104003          ERROR ;RXCSR BIT3 FAILED TO CLEAR. *
1728 007464 052777 000010 171716 AQB: SRESET ;SET RXCSR BIT3. *
1729 007472 104011          SRESET ;ISSUE RESET TO CLEAR BIT. *
1730 007474 104012          SCOPE ;SCOPE *
1731 007476          TSTA 10.,AR,0 *
1732 007476          TSTAA AR,10.,\X+1+0,\X+2,\X+1 *
1733          ; ***** ;
1734 007476 000013          AT13: 13 ;TEST NUMBER *
1735 007500 007604          AT14 ;ADDRESS OF NEXT TEST *
1736 007502 000012          10. ;ITERATION COUNT *
1737 007504 007506          ARA ;SCOPE ENTRY POINT *
1738          X=X+1 ; *
1739          ; ***** ;
1740          ;TEST THAT RXCSR BIT5 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT. *
1741 007506 012737 000340 177776 ARA: MOV #PRTY7,PSW ;PRTY7 TO INHIBIT ANY INT *
1742 007514 032777 000040 171666 BIT #BIT5,@RXCSR ;SEE IF RXCSR BIT5 IS CLEAR. *
1743 007522 001402          BEQ ARB ;BRANCH IF BIT IS CLEAR. *
1744 007524 104003          ERROR ;RESET DID NOT CLEAR RXCSR BIT5 *
1745 007526 000421          BR ARD *
1746 007530 052777 000040 171652 ARB: BIS #BIT5,@RXCSR ;SET RXCSR BIT5. *
1747 007536 032777 000040 171644 BIT #BIT5,@RXCSR ;SEE IF BIT IS SET. *
1748 007544 001002          BNE ARC ;BRANCH IF BIT IS SET. *
1749 007546 104003          ERROR ;RXCSR BIT5 FAILED TO SET. *
1750 007550 000410          BR ARD *
1751 007552 042777 000040 171630 ARC: BIC #BIT5,@RXCSR ;CLEAR RXCSR BIT5 *
1752 007560 032777 000040 171622 BIT #BIT5,@RXCSR ;SEE IF BIT IS CLEAR. *
1753 007566 001401          BEQ ARD *
1754 007570 104003          ERROR ;RXCSR BIT4 FAILED TO CLEAR. *
1755 007572 052777 000040 171610 ARD: BIS #BIT5,@RXCSR ;SET RXCSR BIT5. *
1756 007600 104011          SRESET ;ISSUE RESET TO CLEAR BIT. *
1757 007602 104012          SCOPE ;SCOPE *
1758 007604          TSTA 10.,AS,CD *
1759 007604          TSTAA AS,10.,\X+1+CD,\X+2,\X+1 *
1760          ; ***** ;
```

```
1761 007604 100014 AT14: 100014 ;TEST NUMBER *
1762 007606 007712 AT15 ;ADDRESS OF NEXT TEST *
1763 007610 000012 10. ;ITERATION COUNT *
1764 007612 007614 ASA ;SCOPE ENTRY POINT *
1765 000014 X=X+1 ; *
1766 ; *
1767 ;TEST THAT RXCSR BIT6 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT. *
1768 007614 012737 000340 177776 ASA: MOV #PRTY7,PSW ;SET PRIORITY 7. *
1769 007622 032777 000100 171560 BIT #BIT6,@RXCSR ;SEE IF RXCSR BIT6 IS CLEAR. *
1770 007630 001402 BEQ ASB ;BRANCH IF BIT IS CLEAR. *
1771 007632 104003 ERROR ;RESET DID NOT CLEAR RXCSR BIT6 *
1772 007634 000421 BR ASD *
1773 007636 052777 000100 171544 ASB: BIS #BIT6,@RXCSR ;SET RXCSR BIT6. *
1774 007644 032777 000100 171536 BIT #BIT6,@RXCSR ;SEE IF BIT IS SET. *
1775 007652 001002 BNE ASC ;BRANCH IF BIT IS SET. *
1776 007654 104003 ERROR ;RXCSR BIT6 FAILED TO SET. *
1777 007656 000410 BR ASD *
1778 007660 042777 000100 171522 ASC: BIC #BIT6,@RXCSR ;CLEAR RXCSR BIT6 *
1779 007666 032777 000100 171514 BIT #BIT6,@RXCSR ;SEE IF BIT IS CLEAR. *
1780 007674 001401 BEQ ASD *
1781 007676 104003 ERROR ;RXCSR BIT6 FAILED TO CLEAR. *
1782 007700 052777 000100 171502 ASD: BIS #BIT6,@RXCSR ;SET RXCSR BIT6. *
1783 007706 104011 SRESET ;ISSUE RESET TO CLEAR BIT. *
1784 007710 104012 SCOPE ;SCOPE *
1785 007712 TSTA 100.,AT,0 *
1786 007712 TSTAA AT,100.,\X+1+0,\X+2,\X+1 *
1787 ; *
1788 007712 000015 AT15: 15 ;TEST NUMBER *
1789 007714 007740 AT16 ;ADDRESS OF NEXT TEST *
1790 007716 000144 100. ;ITERATION COUNT *
1791 007720 007722 ATA ;SCOPE ENTRY POINT *
1792 000015 X=X+1 ; *
1793 ; *
1794 ;TEST THAT RXCSR BIT7 IS CLEAR AND CAN BE READ RELIABLY. *
1795 007722 032777 000200 171460 ATA: BIT #BIT7,@RXCSR ;SEE IF RXCSR BIT7 IS CLEAR. *
1796 007730 001402 BEQ ATB ;BRANCH IF BIT IS CLEAR. *
1797 007732 104003 ERROR ;RXCSR BIT7 IS NOT CLEAR. *
1798 007734 104011 SRESET ;RESET IF ERROR *
1799 007736 104012 ATB: SCOPE ;SCOPE *
1800 007740 TSTA 100.,AX,0 *
1801 007740 TSTAA AX,100.,\X+1+0,\X+2,\X+1 *
1802 ; *
1803 007740 000016 AT16: 16 ;TEST NUMBER *
1804 007742 007766 AT17 ;ADDRESS OF NEXT TEST *
1805 007744 000144 100. ;ITERATION COUNT *
1806 007746 007750 AXA ;SCOPE ENTRY POINT *
1807 000016 X=X+1 ; *
1808 ; *
1809 ;TEST THAT RXCSR BIT10 IS CLEAR AND CAN BE READ RELIABLY. *
1810 007750 032777 002000 171432 AXA: BIT #BIT10,@RXCSR ;SEE IF RXCSR BIT10 IS CLEAR. *
1811 007756 001402 BEQ AXB ;BRANCH IF BIT IS CLEAR. *
1812 007760 104003 ERROR ;RXCSR BIT10 IS NOT CLEAR. *
1813 007762 104011 SRESET ;RESET BIT IF ERROR *
1814 007764 104012 AXB: SCOPE ;SCOPE *
1815 007766 TSTA 100.,AY,CD *
1816 007766 TSTAA AY,100.,\X+1+CD,\X+2,\X+1 *
```

```
1817 ; *
1818 007766 100017 AT17: 100017 ;TEST NUMBER *
1819 007770 010014 AT20 ;ADDRESS OF NEXT TEST *
1820 007772 000144 100. ;ITERATION COUNT *
1821 007774 007776 AYA ;SCOPE ENTRY POINT *
1822 000017 X=X+1 ; *
1823 ; *
1824 ;TEST THAT RXCSR BIT11 IS CLEAR AND CAN BE READ RELIABLY. *
1825 007776 032777 004000 171404 AYA: BIT #BIT11,@RXCSR ;SEE IF RXCSR BIT11 IS CLEAR. *
1826 010004 001402 BEQ AYB ;BRANCH IF BIT IS CLEAR. *
1827 010006 104003 ERROR ;RXCSR BIT11 IS NOT CLEAR. *
1828 010010 104011 SRESET ;RESET BIT IF ERROR *
1829 010012 104012 AYB: SCOPE ;SCOPE *
1830 010014 TSTA 100.,AZ,CD *
1831 010014 TSTAA AZ,100.,\X+1+CD,\X+2,\X+1 *
1832 ; *
1833 010014 100020 AT20: 100020 ;TEST NUMBER *
1834 010016 010042 AT21 ;ADDRESS OF NEXT TEST *
1835 010020 000144 100. ;ITERATION COUNT *
1836 010022 010024 AZA ;SCOPE ENTRY POINT *
1837 000020 X=X+1 ; *
1838 ; *
1839 ;TEST THAT RXCSR BIT14 IS CLEAR AND CAN BE READ RELIABLY. *
1840 010024 032777 040000 171356 AZA: BIT #BIT14,@RXCSR ;SEE IF RXCSR BIT14 IS CLEAR. *
1841 010032 001402 BEQ AZB ;BRANCH IF BIT IS CLEAR. *
1842 010034 104003 ERROR ;RXCSR BIT14 IS NOT CLEAR. *
1843 010036 104011 SRESET ;RESET BIT IF ERROR *
1844 010040 104012 AZB: SCOPE ;SCOPE *
1845 010042 TSTA 100.,AAA,CD *
1846 010042 TSTAA AAA,100.,\X+1+CD,\X+2,\X+1 *
1847 ; *
1848 010042 100021 AT21: 100021 ;TEST NUMBER *
1849 010044 010070 AT22 ;ADDRESS OF NEXT TEST *
1850 010046 000144 100. ;ITERATION COUNT *
1851 010050 010052 AAAA ;SCOPE ENTRY POINT *
1852 000021 X=X+1 ; *
1853 ; *
1854 ;TEST THAT RXCSR BIT15 IS CLEAR AND CAN BE READ RELIABLY. *
1855 010052 032777 100000 171330 AAAA: BIT #BIT15,@RXCSR ;SEE IF RXCSR BIT15 IS CLEAR. *
1856 010060 001402 BEQ AAAB ;BRANCH IF BIT IS CLEAR. *
1857 010062 104003 ERROR ;RXCSR BIT15 IS NOT CLEAR. *
1858 010064 104011 SRESET ;RESET BIT IF ERROR *
1859 010066 104012 AAAB: SCOPE ;SCOPE *
1860 ; *
1861 ;ALL PREVIOUS TESTS MUST HAVE BEEN RUN SUCCESSFULLY PRIOR *
1862 ;TO RUNNING THE FOLLOWING TESTS. ALSO, THE JUMPER CONNECTOR *
1863 ;MUST BE INSERTED IN THE DL11-E CABLE IN PLACE OF THE MODEM. COMMENTS *
1864 ;REFER TO OPERATION WITH JUMPER INSERTED. *
1865 ; *
1866 010070 TSTA 100.,AFB,0 *
1867 010070 TSTAA AFB,100.,\X+1+0,\X+2,\X+1 *
1868 ; *
1869 010070 000022 AT22: 22 ;TEST NUMBER *
1870 010072 010154 AT23 ;ADDRESS OF NEXT TEST *
1871 010074 000144 100. ;ITERATION COUNT *
1872 010076 010100 AFBA ;SCOPE ENTRY POINT *
```

```
1873          000022          X=X+1          ;
1874          ;*****/*****
1875          ;TEST THAT CARRIER DETECT SETS AND CLEARS WHEN DATA TERMINAL
1876          ;READY SETS AND CLEARS.
1877 010100 052777 000002 171302 AFBA: BIS #BIT1,@RXCSR ;SET DATA TERMINAL READY
1878 010106 004737 011762          JSR %7,TIME ;DELAY
1879 010112 032777 010000 171270 BIT #BIT12,@RXCSR ;TEST CARRIER DETECT
1880 010120 001002          BNE AFBB ;SHOULD BE SET
1881 010122 104003          ERROR ;WASN'T
1882 010124 000412          BR AFBC
1883 010126 042777 000002 171254 AFBB: BIC #BIT1,@RXCSR ;CLEAR DATA TERMINAL READY
1884 010134 004737 011762          JSR %7,TIME ;DELAY
1885 010140 032777 010000 171242 BIT #BIT12,@RXCSR ;TEST CARRIER DETECT
1886 010146 001401          BEQ AFBC ;SHOULD BE SET
1887 010150 104003          ERROR ;WAS SET, ERROR
1888 010152 104012          AFBC: SCOPE
1889 010154          TSTA 100.,AGB,0
1890 010154          TSTAA AGB,100.,\X+1+0,\X+2,\X+1
1891          ;*****
1892 010154 000023          AT23: 23 ;TEST NUMBER
1893 010156 010326          AT24 ;ADDRESS OF NEXT TEST
1894 010160 000144          100. ;ITERATION COUNT
1895 010162 010164          AGBA ;SCOPE ENTRY POINT
1896          X=X+1
1897          ;*****
1898          ;TEST THAT MODEM INTERRUPT (BIT 15) SETS WHEN CARRIER DETECT
1899          ;CHANGES STATE, AND IS CLEARED WHEN RXCSR IS READ.
1900 010164 042777 000002 171216 AGBA: BIC #BIT1,@RXCSR ;CLEAR DATA TERMINAL READY
1901 010172 004737 011762          JSR %7,TIME ;DELAY
1902 010176 017737 171206 001606 MOV @RXCSR,RXCSTR ;READ RXCSR
1903 010204 032777 100000 171176 BIT #BIT15,@RXCSR ;TEST MODEM INTERRUPT
1904 010212 001402          BEQ AGBB ;WAS CLEAR GO TO AGBB
1905 010214 104003          ERROR ;WASN'T CLEAR
1906 010216 000442          BR AGBE ;GO TO SCOPE
1907 010220 052777 000002 171162 AGBB: BIS #BIT1,@RXCSR ;SETTING DATA TERMINAL READY
1908          ;CAUSES CARRIER DETECT TO SET
1909          ;WHICH CAUSES MODEM INTERRUPT TO SET
1910 010226 004737 011762          JSR %7,TIME ;DELAY
1911 010232 017737 171152 001606 MOV @RXCSR,RXCSTR ;MOVE RXCSR TO TEMPORARY LOCATION
1912 010240 032737 100000 001606 BIT #BIT15,RXCSTR ;TEST MODEM INTERRUPT
1913 010246 001002          BNE AGBC ;SHOULD BE SET GO TO AGBC
1914 010250 104003          ERROR ;WAS CLEAR
1915 010252 000424          BR AGBE ;GO TO SCOPE
1916 010254 032777 100000 171126 AGBC: BIT #BIT15,@RXCSR ;MODEM INTERRUPT BIT SHOULD
1917          ;HAVE BEEN CLEARED
1918 010262 001402          BEQ AGBD ;IT WAS GO TO AGBD
1919 010264 104003          ERROR ;IT WASN'T
1920 010266 000416          BR AGBE ;GO TO SCOPE
1921 010270 042777 000002 171112 AGBD: BIC #BIT1,@RXCSR ;CLEARING DATA TERMINAL READY
1922          ;CAUSES CARRIER DETECT TO CLEAR
1923          ;BUT MODEM INTERRUPT WILL SET
1924 010276 004737 011762          JSR %7,TIME ;DELAY
1925 010302 017737 171102 001606 MOV @RXCSR,RXCSTR ;MOV RXCSR TO TEMPORARY LOCATION
1926 010310 032737 100000 001606 BIT #BIT15,RXCSTR ;TEST MODEM INTERRUPT
1927 010316 001002          BNE AGBE ;SHOULD BE SET
1928 010320 104003          ERROR ;IT WASN'T
```

```
1929          010322 000400          BR AGBE
1930 010324 104012          AGBE: SCOPE ;SCOPE
1931 010326          TSTA 100.,AJB,0
1932 010326          TSTAA AJB,100.,\X+1+0,\X+2,\X+1
1933          ;*****
1934 010326 000024          AT24: 24 ;TEST NUMBER
1935 010330 010440          AT25 ;ADDRESS OF NEXT TEST
1936 010332 000144          100. ;ITERATION COUNT
1937 010334 010336          AJBA ;SCOPE ENTRY POINT
1938          X=X+1
1939          ;*****
1940          ;TEST THAT CLEAR TO SEND (BIT13) SETS/CLEARS WHEN DATA TERMINAL
1941          ;READY SETS/CLEARS.
1942 010336 042777 000002 171044 AJBA: BIC #BIT1,@RXCSR ;CLEAR DATA TERMINAL READY
1943 010344 004737 011762          JSR %7,TIME ;DELAY
1944 010350 032777 020000 171032 BIT #BIT13,@RXCSR ;TEST CLEAR TO SEND
1945 010356 001402          BEQ AJBB ;TEST CLEAR TO SEND
1946 010360 104003          ERROR ;CLEAR TO SEND SHOULD BE CLEAR
1947 010362 000425          BR AJBD
1948 010364 052777 000002 171016 AJBB: BIS #BIT1,@RXCSR ;SET DATA TERMINAL READY
1949 010372 004737 011762          JSR %7,TIME ;DELAY
1950 010376 032777 020000 171004 BIT #BIT13,@RXCSR ;TEST CLEAR TO SEND
1951 010404 001002          BNE AJBC ;BRANCH IF SET
1952 010406 104003          ERROR ;CLEAR TO SEND SHOULD BE SET
1953 010410 000412          BR AJBD
1954 010412 042777 000002 170770 AJBC: BIC #BIT1,@RXCSR ;CLEAR DATA TERMINAL READY
1955 010420 004737 011762          JSR %7,TIME ;DELAY
1956 010424 032777 020000 170756 BIT #BIT13,@RXCSR ;TEST CLEAR TO SEND
1957 010432 001401          BEQ AJBD ;TEST CLEAR TO SEND
1958 010434 104003          ERROR ;CLEAR TO SEND SHOULD BE CLEAR
1959 010436 104012          AJBD: SCOPE
1960 010440          TSTA 100.,AKB,0
1961 010440          TSTAA AKB,100.,\X+1+0,\X+2,\X+1
1962          ;*****
1963 010440 000025          AT25: 25 ;TEST NUMBER
1964 010442 010534          AT26 ;ADDRESS OF NEXT TEST
1965 010444 000144          100. ;ITERATION COUNT
1966 010446 010450          AKBA ;SCOPE ENTRY POINT
1967          X=X+1
1968          ;*****
1969          ;TEST THAT RING (BIT 14 RXCSR) SETS WHEN REQUEST TO
1970          ;SEND SETS AND CLEARS AND RESET CLEARS RING
1971 010450 042777 000004 170732 AKBA: BIC #BIT2,@RXCSR ;CLEAR REQUEST TO SEND
1972 010456 004737 011762          JSR %7,TIME ;DELAY
1973 010462 052777 000004 170720 BIS #BIT2,@RXCSR ;SET REQUEST TO SEND
1974 010470 004737 011762          JSR %7,TIME ;DELAY
1975 010474 032777 040000 170706 BIT #BIT14,@RXCSR ;TEST RING
1976 010502 001001          BNE AKBC ;TEST RING
1977 010504 104003          ERROR ;RING SHOULD BE SET
1978 010506 042777 000004 170674 AKBC: BIC #BIT2,@RXCSR ;CLEAR REQUEST TO SEND
1979 010514 004737 011762          JSR %7,TIME ;DELAY
1980 010520 032777 040000 170662 BIT #BIT14,@RXCSR ;TEST RING
1981 010526 001401          BEQ .+4 ;SHOULD BE CLEAR
1982 010530 104003          ERROR
1983 010532 104012          SCOPE ;SCOPE
1984 010534          TSTA 100.,AOB,0
```

```
1985 010534 TSTAA AOB,100.,\X+1+0,\X+2,\X+1
1986 ;*****
1987 010534 000026 AT26: 26 ;TEST NUMBER *
1988 010536 010646 AT27 ;ADDRESS OF NEXT TEST *
1989 010540 000144 100. ;ITERATION COUNT *
1990 010542 010544 AOB4 ;SCOPE ENTRY POINT *
1991 000026 X=X+1 ;
1992 ;*****
1993 ;TEST THAT MODEM INTERRUPT (BIT 15 RXCSR) SETS WHEN RING SETS.
1994 010544 042777 000004 170636 AOB4: BIC #BIT2,@RXCSR ;CLEAR REQUEST TO SEND *
1995 010552 004737 011762 JSR %7,TIME ;DELAY *
1996 010556 032777 100000 170624 BIT #BIT15,@RXCSR ;TEST MODEM INTERRUPT BIT *
1997 010564 001402 BEQ AOB5 ;
1998 010566 104003 ERROR ;
1999 010570 000425 BR AOB6 ;
2000 010572 052777 000004 170610 AOB7: BIS #BIT2,@RXCSR ;SET REQUEST TO SEND *
2001 010600 004737 011762 JSR %7,TIME ;DELAY *
2002 010604 032777 100000 170576 BIT #BIT15,@RXCSR ;TEST MODEM INTERRUPT BIT *
2003 010612 001002 BNE AOB8 ;
2004 010614 104003 ERROR ;
2005 010616 000412 BR AOB9 ;
2006 010620 042777 000004 170562 AOB8: BIC #BIT2,@RXCSR ;CLEAR REQUEST TO SEND *
2007 010626 004737 011762 JSR %7,TIME ;DELAY *
2008 010632 032777 100000 170550 BIT #BIT15,@RXCSR ;TEST MODEM INTERRUPT BIT *
2009 010640 001401 BEQ AOB9 ;
2010 010642 104003 ERROR ;
2011 010644 104012 AOB9: SCOPE ;SCOPE *
2012 010646 TSTA 100.,ALB,0 ;
2013 010646 TSTAA ALB,100.,\X+1+0,\X+2,\X+1 ;
2014 ;*****
2015 010646 000027 AT27: 27 ;TEST NUMBER *
2016 010650 001760 AT30 ;ADDRESS OF NEXT TEST *
2017 010652 000144 100. ;ITERATION COUNT *
2018 010654 010656 ALBA ;SCOPE ENTRY POINT *
2019 000027 X=X+1 ;
2020 ;*****
2021 ;TEST THAT SUPERVISORY RECEIVE DATA (BIT 10 RXCSR) SETS/CLEAR
2022 ;WHEN SUPERVISORY XMIT DATA SETS/CLEAR.
2023 010656 042777 000010 170524 ALBA: BIC #BIT3,@RXCSR ;CLEAR SUPERVISOR XMIT DATA *
2024 010664 004737 011762 JSR %7,TIME ;DELAY *
2025 010670 032777 002000 170512 BIT #BIT10,@RXCSR ;TEST SUPERVISORY RECEIVE DATA. *
2026 010676 001402 BEQ ALBB ;
2027 010700 104003 ERROR ;SHOULD HAVE BEEN CLEAR *
2028 010702 000425 BR ALBD ;
2029 010704 052777 000010 170476 ALBB: BIS #BIT3,@RXCSR ;SET SUPERVISORY XMIT DATA *
2030 010712 004737 011762 JSR %7,TIME ;DELAY *
2031 010716 032777 002000 170464 BIT #BIT10,@RXCSR ;TEST SUPERVISORY RECEIVE DATA *
2032 010724 001002 BNE ALBC ;
2033 010726 104003 ERROR ;SHOULD HAVE BEEN SET *
2034 010730 000412 BR ALBD ;
2035 010732 042777 000010 170450 ALBC: BIC #BIT3,@RXCSR ;CLEAR SUPERVISORY XMIT DATA *
2036 010740 004737 011762 JSR %7,TIME ;DELAY *
2037 010744 032777 002000 170436 BIT #BIT10,@RXCSR ;TEST SUPERVISORY RECEIVE DATA *
2038 010752 001401 BEQ ALBD ;
2039 010754 104003 ERRDR ;SHOULD HAVE BEEN CLEAR *
2040 010756 104012 ALBD: SCOPE ;SCOPE *
```

```
2041 010760 TSTA 100.,AMB,0
2042 010760 TSTAA AMB,100.,\X+1+0,\X+2,\X+1
2043 ;*****
2044 010760 000030 AT30: 30 ;TEST NUMBER *
2045 010762 011120 AT31 ;ADDRESS OF NEXT TEST *
2046 010764 000144 100. ;ITERATION COUNT *
2047 010766 010770 AMBA ;SCOPE ENTRY POINT *
2048 000030 X=X+1 ;
2049 ;*****
2050 ;TEST THAT SUP REC DATA TRANSITIONS SET MODEM INTERRUPT
2051 010770 042777 000010 170412 AMBA: BIC #BIT3,@RXCSR ;CLEAR SUP REC *
2052 010776 004737 011762 JSR %7,TIME ;DELAY *
2053 011002 052777 000010 170400 BIS #BIT3,@RXCSR ;SET SUP REC *
2054 011010 004737 011762 JSR %7,TIME ;DELAY *
2055 011014 032777 100000 170366 BIT #BIT15,@RXCSR ;TEST MODEM INTERRUPT *
2056 011022 001002 BNE AMBB ;MODEM INTERRUPT SHOULD BE SET *
2057 011024 104003 ERROR ;
2058 011026 000433 BR AMBE ;
2059 011030 032777 100000 170352 AMBB: BIT #BIT15,@RXCSR ;MODEM INTERRUPT SHOULD BE *
2060 011036 001402 BEQ AMBC ;CLEARED BY PREVIOUS READ *
2061 011040 104003 ERROR ;
2062 011042 000425 BR AMBE ;
2063 011044 042777 000010 170336 AMBC: BIC #BIT3,@RXCSR ;1-0 TRANS OF SUP REC DATA *
2064 011052 004737 011762 JSR %7,TIME ;DELAY *
2065 011056 032777 100000 170324 BIT #BIT15,@RXCSR ;TEST MODEM INTERRUPT *
2066 011064 001002 BNE AMBD ;SHOULD BE SET *
2067 011066 104003 ERROR ;
2068 011070 000412 BR AMBE ;
2069 011072 052777 000010 170310 AMBD: BIS #BIT3,@RXCSR ;0-1 TRANS OF SUP REC DATA *
2070 011100 004737 011762 JSR %7,TIME ;DELAY *
2071 011104 032777 100000 170276 BIT #BIT15,@RXCSR ;TEST MODEM INTERRUPT *
2072 011112 001001 BNE AMBE ;SHOULD BE SET *
2073 011114 104003 ERROR ;
2074 011116 104012 AMBE: SCOPE *
2075 011120 TSTA 10.,ABA,CD *
2076 011120 TSTAA ABA,10.,\X+1+CD,\X+2,\X+1 *
2077 ;*****
2078 011120 100031 AT31: 100031 ;TEST NUMBER *
2079 011122 011224 AT32 ;ADDRESS OF NEXT TEST *
2080 011124 000012 10. ;ITERATION COUNT *
2081 011126 011130 ABAA ;SCOPE ENTRY POINT *
2082 000031 X=X+1 ;
2083 ;*****
2084 ;TEST THAT RESET CLEARS ALL TXCSR BITS, AND SETS BIT 7 (READY)
2085 011130 012737 000340 177776 ABAA: MOV #PRTY7,PSW ;SET PRIORITY 7. *
2086 011136 012777 177777 170250 MOV #-1,@TXCSR ;SET ALL POSSIBLE BITS IN TXCSR *
2087 011144 104011 SRESET ;ISSUE RESET TO CLEAR BITS *
2088 011146 022777 000200 170240 CMP #BIT7,@TXCSR ;SEE IF ONLY BIT 7 IS SET. *
2089 011154 001422 BEQ ABAB ;BRANCH IF ONLY BIT 7 IS SET *
2090 011156 017737 170232 001604 MOV @TXCSR,TXCSRT ;SAVE CONTENTS OF TXCSR *
2091 011164 012737 000200 001616 MOV #BIT7,TEMP ;MOVE EXPECTED TXCSR TO TEMP. *
2092 011172 004537 004474 JSR %5,DACNV ;GO TO OCTAL TO ASCII CONVERT. *
2093 011176 001616 TEMP ;SOURCE ADDR. *
2094 011200 015442 ATXSB ;DESTINATION ADDR. *
2095 011202 000006 6 ;#OF DIGITS TO CONVERT. *
2096 011204 004537 004474 JSR %5,DACNV ;GO TO OCTAL TO ASCII CONVERT. *
```

```
2097 011210 001604 TXCSR ;SOURCE ADDR.
2098 011212 015457 ATXWAS ;DESTINATION ADDR.
2099 011214 000006 6 ;#OF DIGITS TO CONVERT.
2100 011216 104015 ERROR1 ;RESET FAILED TO CLEAR ALL BITS EXCEPT
2101 011220 015427 ATXCSR ;BIT 7 - SEE PRINTOUT
2102 011222 104012 SCOPE ;SCOPE
ABAB: TSTA 10.,ACA,0
TSTAA ACA,10.,\X+1+0,\X+2,\X+1
;*****
2106 011224 000032 AT32: 32 ;TEST NUMBER
2107 011226 011374 AT33 ;ADDRESS OF NEXT TEST
2108 011230 000012 10 ;ITERATION COUNT
2109 011232 011234 ACAA ;SCOPE ENTRY POINT
2110 000032 X=X+1
;*****
;TEST THAT RESET CLEARS ALL RXCSR BITS EXCEPT DATA TERMINAL READY, RING
;CLEAR TO SEND,CARRIER DET
ACAA: MOV #PRTY7,PSW ;SET PRIORITY 7
BIC #BIT1,@RXCSR ;CLEAR DATA TERM.READY
MOV #-1,@RXCSR ;SET ALL POSSIBLE BITS IN RXCSR
BIS #4,@TXCSR ;SET MAINT BIT
CLR @TXBUF ;TRANSMIT A CHAR
TIMETX ;TIME OUT TX DONE
ERROR ;ERROR DONE NOT SETTING
MOV #1,@TXBUF ;TRANSMIT ANOTHER CHAR.
TIMERX ;TIME OUT RX DONE
ERROR ;ERROR DONE NOT SETTING
SRESET ;ISSUE RESET TO CLEAR BITS.
MOV @RXCSR,RXCSR ;MOVE RXCSR CONTENTS TO RXCSR
CMP #30002,RXCSR ;SEE IF ONLY BITS 1,12,13 SET
BEQ ACAB ;BRANCH IF ONLY BITS 1,12,13 SET.
2128 011326 012737 030002 001616 MOV #30002,TEMP
2129 011334 004537 004474 JSR %5,OACNV ;GO TO OCTAL TO ASCII CONVERT.
2130 011340 001616 TEMP ;SOURCE ADDR.
2131 011342 015501 ARXSB ;DESTINATION ADDR.
2132 011344 000006 6 ;#OF DIGITS TO CONVERT.
2133 011346 004537 004474 JSR %5,OACNV ;GO TO OCTAL TO ASCII CONVERT.
2134 011352 001606 RXCSR ;SOURCE ADDR.
2135 011354 015516 ARXWAS ;DESTINATION ADDR.
2136 011356 000006 6 ;#OF DIGITS TO CONVERT.
2137 011360 104015 ERROR1 ;RESET FAILED TO CLEAR ALL BITS EXCEPT
2138 011362 015466 ARXCSR ;BIT 0. SEE ERROR PRINTOUT.
2139 011364 042777 000002 170016 ACAB: BIC #BIT1,@RXCSR ;CLEAR DATA TERM. READY
2140 011372 104012 SCOPE ;SCOPE
2141 011374 TSTA 10.,ADA,CD
2142 011374 TSTAA ADA,10.,\X+1+CD,\X+2,\X+1
;*****
2144 011374 100033 AT33: 100033 ;TEST NUMBER
2145 011376 011454 AT34 ;ADDRESS OF NEXT TEST
2146 011400 000012 10 ;ITERATION COUNT
2147 011402 011404 ADAA ;SCOPE ENTRY POINT
2148 000033 X=X+1
;*****
;TEST THAT LOADING TXBUF (TRANSMITTER BUFFER) CLEARS TXCSR BIT 7 (READY)
;AND WITHOUT MAINT SET THAT TXDONE SETS READY
2152 011404 005077 170006 ADAA: CLR @TXBUF ;LOAD TXBUF
```

```
2153 011410 104020 TIMETX ;TIME OUT TX DONE
2154 011412 104003 ERROR ;ERROR, DONE NOT SETTING
2155 011414 005077 167776 CLR @TXBUF ;LOAD TX BUF
2156 011420 105777 167770 TSTB @TXCSR ;TEST TXCSR BIT 7 (READY BIT)
2157 011424 100002 BPL ADAB ;BRANCH IF BIT NOT SET.
2158 011426 104003 ERROR ;ERROR. LOADING TXBUF FAILED TO CLEAR READY.
2159 011430 000407 BR ADAC
2160 011432 104020 ADAB: TIMETX ;WAIT FOR DONE
2161 011434 104003 ERROR ;DONE NEVER SET
2162 011436 032777 000200 167750 BIT #BIT7,@TXCSR
2163 011444 001001 BNE ;+4
2164 011446 104003 ERROR ;READY DID NOT SET
2165 011450 104011 ADAC: SRESET ;SCOPE.
2166 011452 104012 SCOPE
2167 011454 TSTA 1.,AIA,CD
2168 011454 TSTAA AIA,1.,\X+1+CD,\X+2,\X+1
;*****
2170 011454 100034 AT34: 100034 ;TEST NUMBER
2171 011456 011776 AT35 ;ADDRESS OF NEXT TEST
2172 011460 000001 1 ;ITERATION COUNT
2173 011462 011464 AIAA ;SCOPE ENTRY POINT
2174 000034 X=X+1
;*****
;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.
2178 011464 004737 006216 AIAA: JSR %7,DOTHIS ;TEST IF THIS TEST SELECTED
2179 011470 104001 TYPES
2180 011472 016764 MSETTX
2181 011474 017012 MSETC
2182 011476 017274 MS0
2183 011500 177777 -1
2184 011502 000000 HALT
2185 011504 004737 011712 JSR %7,AIAS ;OUTPUT CHAR AND TIME.
2186 011510 013737 011774 001564 MOV AIAST,CTRO ;MOVE ELAPSED TIME TO CTRO.
2187 011516 104000 TYPE
2188 011520 017304 MS1
2189 011522 000000 HALT
2190 011524 004737 011712 JSR %7,AIAS ;OUTPUT CHAR AND TIME.
2191 011530 013737 011774 001566 MOV AIAST,CTR1 ;MOVE ELAPSED TIME TO CTR1.
2192 011536 104000 TYPE
2193 011540 017314 MS2
2194 011542 000000 HALT
2195 011544 004737 011712 JSR %7,AIAS ;OUTPUT CHAR AND TIME.
2196 011550 013737 011774 001570 MOV AIAST,CTR2 ;MOVE ELAPSED TIME TO CTR2.
2197 011556 104000 TYPE
2198 011560 017324 MS3
2199 011562 000000 HALT
2200 011564 004737 011712 JSR %7,AIAS ;OUTPUT CHAR AND TIME.
2201 011570 013737 011774 001572 MOV AIAST,CTR3 ;MOVE ELAPSED TIME TO CTR3.
2202 011576 104000 TYPE
2203 011600 017334 MS4
2204 011602 000000 HALT
2205 011604 004737 011712 JSR %7,AIAS ;OUTPUT CHAR AND TIME
2206 011610 013737 011774 001574 MOV AIAST,CTR4 ;MOVE ELAPSED TIME TO CTR4
2207 011616 104000 TYPE
2208 011620 017344 MSS
```



```

2209 011622 000000          HALT
2210 011624 004737 011712    JSR      %7, AIAS          ;OUTPUT CHAR AND TIME
2211 011630 013737 011774 001576  MOV      AIAST,CTR5      ;MOVE ELAPSED TIME TO CTR5
2212 011636 104000          TYPE
2213 011640 017354          MS6
2214 011642 000000          HALT
2215 011644 004737 011712    JSR      %7, AIAS          ;OUTPUT CHAR AND TIME
2216 011650 013737 011774 001600  MOV      AIAST,CTR6      ;MOVE ELAPSED TIME TO CTR6
2217 011656 104000          TYPE
2218 011660 017364          MS7
2219 011662 000000          HALT
2220 011664 004737 011712    JSR      %7, AIAS          ;OUTPUT CHAR AND TIME
2221 011670 013737 011774 001602  MOV      AIAST,CTR7      ;MOVE ELAPSED TIME TO CTR7
2222 011676 004737 014424    JSR      %7, CMPT         ;CHECK THAT CTR0 THROUGH CTR7 CONTAIN
2223 011702 000402          BR        AIAF            ;DESCENDING VALUES
2224 011704 104015          ERROR1   ;TRANSMIT SPEEDS NOT ARRANGED IN
2225 011706 015526          ETXTIM   ;ASCENDING ORDER.
2226 011710 104012          SCOPE
2227
2228 011712 005037 011774    AIAF:   CLR        AIAST          ;CLEAR ELAPSED TIME COUNTER.
2229 011716 105777 167472    ; AIAS:  TSTB      @TXCSR        ;WAIT FOR TX READY.
2230 011722 100375          BPL      .-4
2231 011724 005077 167466    CLR      @TXBUF
2232 011730 105777 167460    TSTB    @TXCSR
2233 011734 100375          BPL      .-4
2234 011736 005077 167454    CLR      @TXBUF          ;LOAD TXBUF.
2235 011742 004737 011762    JSR      %7, TIME        ;WAIT 75 US
2236 011746 005237 011774    INC      AIAST          ;INCREMENT ELAPSED TIME COUNTER.
2237 011752 105777 167436    TSTB    @TXCSR          ;READY SET?
2238 011756 100371          BPL      AIASA          ;BRANCH IF READY NOT SET.
2239 011760 000207          RTS      %7             ;EXIT.
2240
2241 011762 012700 000017    TIME:   MOV      #15.,%0
2242 011766 005300          TIM1:   DEC      %0
2243 011770 001376          BNE     TIM1
2244 011772 000207          RTS      %7
2245 011774 000000          AIAST:  OPEN     TSTA 10.,ALA,0
2246 011776          TSTAA  ALA,10.,\X+1+0,\X+2,\X+1
2247 011778          ;*****
2248          AT35:  35          ;TEST NUMBER
2249 011776 000035          AT36:  AT36          ;ADDRESS OF NEXT TEST
2250 012000 012052          10.          ;ITERATION COUNT
2251 012002 000012          ALAA       ;SCOPE ENTRY POINT
2252 012004 012006          X=X+1
2253 000035          ;*****
2254          ;TEST THAT OUTPUTTING A CHARACTER WITH THE MAINTENANCE BIT SET (TXCSR BIT 2)
2255          ;RESULTS IN DONE BIT SETTING (RXCSR BIT 7) NO LATER THAN 500 MSECS. AND
2256          ;THAT RESET INSTRUCTION CLEARS THE DONE BIT
2257
2258 012006 052777 000004 167400 ALAA:   BIS      #BIT2,@TXCSR          ;SET MAINTENANCE BIT
2259 012014 005077 167376    CLR      @TXBUF          ;LOAD TXBUF
2260 012020 104016          DELAY     500.          ;WAIT 500 MSECS.
2261 012022 000764          TSTB     @RXCSR          ;SEE IF DONE BIT IS SET
2262 012024 105777 167360    BMI      ALAB            ;BRANCH IF DONE BIT IS SET
2263 012030 100402

```

```

2265 012032 104003          ERROR
2266 012034 000405          BR        ALAC            ;DONE BIT FAILED TO SET
2267 012036 104011          SRESET   ALAC
2268 012040 105777 167344    ALAB:   TSTB      @RXCSR          ;ISSUE RESET TO CLEAR DONE BIT
2269 012044 100001          BPL      ALAC            ;SEE IF DONE BIT IS CLEARED
2270 012046 104003          ERROR    ;BRANCH IF DONE BIT IS CLEARED
2271 012050 104012          ALAC:   SCOPE          ;RESET FAILED TO CLEAR DONE BIT
2272 012052          TSTA 100.,AMA,CD
2273 012052          TSTAA  AMA,100.,\X+1+CD,\X+2,\X+1
2274          ;*****
2275 012052 100036          AT36:  100036          ;TEST NUMBER
2276 012054 012116          AT37:  AT37          ;ADDRESS OF NEXT TEST
2277 012056 000144          100.          ;ITERATION COUNT
2278 012060 012062          AMAA       ;SCOPE ENTRY POINT
2279 000036          X=X+1
2280          ;*****
2281          ;TEST THAT DONE BIT (RXCSR BIT 7) IS CLEARED BY READING RXBUF.
2282          ;DONE SET BY OUTPUTTING CHARACTER WITH MAINTENANCE BIT SET (TXCSR BIT 2)
2283 012062 052777 000004 167324 AMAA:   BIS      #BIT2,@TXCSR          ;SET MAINTENANCE BIT (TXCSR BIT 2)
2284 012070 005077 167322    CLR      @TXBUF          ;LOAD TXBUF
2285 012074 104017          TIMERX   ;WAIT FOR DONE BIT TO SET.
2286 012076 104003          ERROR
2287 012100 005777 167306    TST      @RXBUF          ;READ RXBUF TO CLEAR DONE BIT
2288 012104 105777 167300    TSTB    @RXCSR          ;SEE IF DONE BIT IS CLEAR
2289 012110 100001          BPL      AMAC            ;BRANCH IF DONE BIT IS CLEAR
2290 012112 104003          ERROR    ;READING RXBUF FAILED TO CLEAR DONE BIT
2291 012114 104012          AMAC:   SCOPE          ;SCOPE
2292 012116          TSTA 100.,ADA,CD
2293 012116          TSTAA  ADA,100.,\X+1+CD,\X+2,\X+1
2294          ;*****
2295 012116 100037          AT37:  100037          ;TEST NUMBER
2296 012120 012226          AT40       ;ADDRESS OF NEXT TEST
2297 012122 000144          100.          ;ITERATION COUNT
2298 012124 012132          A0AA       ;SCOPE ENTRY POINT
2299 000037          X=X+1
2300          ;*****
2301          ;TEST THAT RECEIVER ACTIVE SETS WHEN CHAR STARTS AND
2302          ;CLEARS WHEN RECEIVER DONE SETS
2303 012126 004737 002706          JSR      %7,CDINIT        ;INIT IF C-D DEVICE
2304 012132 052777 000004 167254 AQAA:   BIS      #BIT2,@TXCSR          ;SET MAINT
2305 012140 005077 167252    CLR      @TXBUF          ;TRANSMIT CHAR
2306 012144 005037 001616    CLR      TEMP            ;CLEAR BUSY INDICATOR
2307 012150 032777 004000 167232 AQAB:   BIT      #BIT11,@RXCSR          ;IS RECEIVER ACTIVE SET
2308 012156 001402          BEQ      AQAB1          ;BRANCH IF CLEAR
2309 012160 005237 001616    INC      TEMP            ;YES, REMEMBER THAT
2310 012164 105777 167220    TSTB    @RXCSR          ;SEE IF DONE SET
2311 012170 100367          BPL      AQAB           ;DID RECEIVER ACTIVE SET
2312 012172 032727 001616 000000    CMP      TEMP,%0
2313 012200 001002          BNE     AQAC            ;RECEIVER ACTIVE NEVER SET
2314 012202 104003          ERROR
2315 012204 000405          BR        AQAC
2316 012206 032777 004000 167174 AQAC:   BIT      #BIT11,@RXCSR          ;DID DONE CLEAR ACTIVE
2317 012214 001401          BEQ      AQAD
2318 012216 104003          ERROR
2319 012220 005777 167166    TST      @RXBUF          ;NO, RECEIVER ACTIVE DID NOT CLEAR
2320 012224 104012          SCOPE

```

```
2321 012226          TSTA 1.,AQA,0
2322 012226          TSTAA AQA,1.,\X+1+0,\X+2,\X+1
2323
2324 012226 000040    ;*****
AT40: 40             ;TEST NUMBER *
2325 012230 012530    AT41             ;ADDRESS OF NEXT TEST *
2326 012232 000001    1.              ;ITERATION COUNT *
2327 012234 012236    AQA             ;SCOPE ENTRY POINT *
2328          000040    X=X+1           ; *
2329
2330 ;*****
2331 ;TEST THAT RECEIVE SPEEDS ARE ARRANGED IN ASCENDING ORDER BY CHECKING THAT TIME
2332 ;ELAPSED TO DONE BIT SETTING (RXCSR BIT 7) DECREASES AS A HIGHER SPEED
2333 ;THIS IS NOT DONE IN MAINTENANCE MODE TX AND RX
2334 ;POTS MUST BE STEPPED TOGETHER
2335 ;IS SELECTED.
2335 012236 004737 006216 AQA: JSR %7,DOTHIS ;CHECK IF THIS TEST TO BE DONE
2336 012242 104001          TYPES
2337 012244 016736          MSETRX
2338 012246 017012          MSETC
2339 012250 017274          MS0
2340 012252 177777          -1
2341 012254 000000          HALT
2342 012256 004737 012464 JSR %7,AQAS ;OUTPUT CHARACTER AND TIME DONE BIT
2343 012262 013737 012526 001564 MOV AQAST,CTRO ;MOVE ELAPSED TIME TO CTRO
2344 012270 104000          TYPE
2345 012272 017304          MS1
2346 012274 000000          HALT
2347 012276 004737 012464 JSR %7,AQAS ;OUTPUT CHARACTER AND TIME DONE BIT
2348 012302 013737 012526 001566 MOV AQAST,CTR1 ;MOVE ELAPSED TIME TO CTR1
2349 012310 104000          TYPE
2350 012312 017314          MS2
2351 012314 000000          HALT
2352 012316 004737 012464 JSR %7,AQAS ;OUTPUT CHARACTER AND TIME DONE BIT.
2353 012322 013737 012526 001570 MOV AQAST,CTR2 ;MOVE ELAPSED TIME TO CTR2.
2354 012330 104000          TYPE
2355 012332 017324          MS3
2356 012334 000000          HALT
2357 012336 004737 012464 JSR %7,AQAS ;OUTPUT CHARACTER AND TIME DONE BIT
2358 012342 013737 012526 001572 MOV AQAST,CTR3 ;MOVE ELAPSED TIME TO CTR3.
2359 012350 104000          TYPE
2360 012352 017334          MS4
2361 012354 000000          HALT
2362 012356 004737 012464 JSR %7,AQAS
2363 012362 013737 012526 001574 MOV AQAST,CTR4
2364 012370 104000          TYPE
2365 012372 017344          MS5
2366 012374 000000          HALT
2367 012376 004737 012464 JSR %7,AQAS
2368 012402 013737 012526 001576 MOV AQAST,CTRS
2369 012410 104000          TYPE
2370 012412 017354          MS6
2371 012414 000000          HALT
2372 012416 004737 012464 JSR %7,AQAS
2373 012422 013737 012526 001600 MOV AQAST,CTR6
```

```
2374 012430 104000          TYPE
2375 012432 017364          MS7
2376 012434 000000          HALT
2377 012436 004737 012464 JSR %7,AQAS
2378 012442 013737 012526 001602 MOV AQAST,CTR7
2379 012450 004737 014424 JSR %7,CMPT ;CHECK THAT CTRO THROUGH CTR3 CONTAIN
2380 012454 000402          BR AQAB ;DESCENDING VALUES.
2381 012456 104015          ERROR1 ;RECEIVE SPEEDS NOT ARRANGED IN
2382 012460 015570          ERXTIM ;ASCENDING ORDER.
2383 012462 104012          SCOPE ;SCOPE
2384
2385 012464 005037 012526 AQA: CLR AQAST ;CLEAR ELAPSED TIME COUNTER AQAST
2386 012470 105777 166720 TSTB @TXCSR ;WAIT FOR TX READY.
2387 012474 100375          BPL -.4
2388 012476 005777 166710 TST @RXBUF ;CLEAR DONE BIT IF SET
2389 012502 005077 166710 CLR @TXBUF ;LOAD TXBUF
2390 012506 004737 011782 AQASA: JSR %7,TIME
2391 012512 005237 012526 INC AQAST ;INCREMENT ELAPSED TIME COUNTER
2392 012516 105777 166666 TSTB @RXCSR ;DONE SET?
2393 012522 100371          BPL AQASA ;BRANCH IF DONE NOT SET
2394 012524 000207          RTS ;EXIT
2395 012526 000000          AQA: OPEN ;ELAPSED TIME COUNTER
2396 012530          TSTA 10.,ARA,CD
2397 012530          TSTAA ARA,10.,\X+1+CD,\X+2,\X+1
2398
2399 ;*****
2400 100041             ;TEST NUMBER *
2401 012532 012674          AT41: AT42             ;ADDRESS OF NEXT TEST *
2402 012534 000012          10.            ;ITERATION COUNT *
2403 012536 012540          ARA             ;SCOPE ENTRY POINT *
2404          X=X+1           ; *
2405 ;*****
2406 ;TEST CORRECT OPERATION OF DATA OVERRUN BIT (RXBUF BIT 14)
2407 012540 004737 012654 ARAA: JSR %7,ARAS ;OUTPUT CHARACTER AND WAIT 500 MSECS
2408 012544 004737 012654 JSR %7,ARAS ;OUTPUT CHARACTER AND WAIT 500 MSECS
2409 012550 017737 166636 001610 MOV @RXBUF,RXBUFT ;SAVE RXBUF CONTENTS + CLEAR DONE
2410 012556 032737 040000 001610 BIT #BIT14,RXBUFT ;SEE IF DATA OVERRUN BIT WAS SET
2411 012564 001002          BNE .+6 ;BRANCH IF BIT WAS SET
2412 012566 104003          ERROR
2413 012570 104012          SCOPE
2414 012572 005737 001610 TST RXBUFT ;SEE THAT ERROR BIT WAS SET (RXBUF BIT 15)
2415 012576 100402          BMI .+6
2416 012600 104003          ERROR ;ERROR BIT FAILED TO SET WHEN OVERRUN SET
2417 012602 104012          SCOPE
2418 012604 032777 040000 166600 BIT #BIT14,@RXBUF ;SEE THAT DATA OVERRUN WAS NOT
2419          BNE .+6 ;CLEARED WHEN RXBUF WAS READ
2420 012612 104003          ERROR ;BRANCH IF SET
2421 012614 104003          SCOPE ;READING RXBUF CLEARED DATA OVERRUN
2422 012616 104012          SCOPE
2423 012620 004737 012654 JSR %7,ARAS
2424 012624 032777 100000 166560 BIT #BIT15,@RXBUF ;OUTPUT CHAR +WAIT 500MS
2425 012632 001402          BEQ .+6 ;TEST THAT ERROR CLEARED
2426 012634 104003          ERROR
2427 012636 104012          SCOPE
2428 012640 032777 040000 166544 BIT #BIT14,@RXBUF ;TEST THAT OVERRUN CLEARED
2429 012646 001401          BEQ .+4
2430 012650 104003          ERROR
```

```
2430 012652 104012 SCOPE ;SCOPE
2431 012654 052777 BIS #BIT2,@TXCSR ;SET MAINTENANCE BIT
2432 012662 005077 166530 ARAS: CLR @TXBUF ;LOAD TXBUF
2433 012666 104016 DELAY ;DELAY 500 MSECS
2434 012670 000764 500.
2435 012672 000207 RTS %7 ;EXIT
2436 012674 TSTA 10.,ATA,CD
2437 012674 TSTAA ATA,10.,\X+1+CD,\X+2,\X+1
;*****
2438 012674 100042 AT42: 100042 ;TEST NUMBER *
2440 012676 012762 AT43 ;ADDRESS OF NEXT TEST *
2441 012700 000012 10. ;ITERATION COUNT *
2442 012702 012714 ATAA ;SCOPE ENTRY POINT *
2443 000042 X=X+1 ;
;*****
;TEST THAT TRANSMITTER IS ABLE TO INTERRUPT. IF THE INTERRUPT IS SERVICED,
;IT WILL HAVE OCCURRED AT THE CORRECT VECTOR.
2447 012704 004737 003516 JSR 7,OVRLAY ;GO TO OVER LAY ROUTINE
2448 012710 104007 STTXV ;SET TX INTERRUPT SERVICE
2449 012712 012750 ATAC ;TO ATAC
2450 012714 042777 000100 166472 ATAA: BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPT
2451 012722 005037 177776 CLR PSW ;SET PROCESSOR PRIORITY TO 0
2452 012726 052777 000100 166460 BIS #BIT6,@TXCSR ;ENABLE TX INTERRUPT
2453 012734 000240 NOP
2454 012736 104003 ERROR ;READY DID NOT CAUSE AN INTERRUPT
2455 012740 042777 000100 166446 ATAB: BIC #BIT6,@TXCSR
2456 012746 104012 ATAB: SCOPE ;SCOPE
2457 012750 042777 000100 166436 ATAC: BIC #BIT6,@TXCSR ;HERE IF INT, DISABLE TX INT
2458 012756 022626 POPSP2
2459 012760 000772 BR ATAB
2460 012762 TSTA 1000.,AUA,CD
2461 012762 TSTAA AUA,1000.,\X+1+CD,\X+2,\X+1
;*****
2463 012762 100043 AT43: 100043 ;TEST NUMBER *
2464 012764 013040 AT44 ;ADDRESS OF NEXT TEST *
2465 012766 001750 1000. ;ITERATION COUNT *
2466 012770 012776 AUA ;SCOPE ENTRY POINT *
2467 000043 X=X+1 ;
;*****
;TEST THAT READY DOES NOT CAUSE AN INTERRUPT WHEN THE PROCESSOR IS
;AT THE SAME PRIORITY AS THE TRANSMITTER INTERRUPT REQUEST LEVEL
2470 STTXV ;SET TX INTERRUPT SERVICE TO
2471 012772 104007 AUAC
2472 012774 013032 AUAC
2473 012776 013737 001426 177776 AUAA: MOV TXLVL,PSW ;SET PROCESSOR PRIORITY SAME AS TX PRIORITY
2474 013004 042777 000100 166402 BIC #BIT6,@TXCSR
2475 013012 052777 000100 166374 BIS #BIT6,@TXCSR ;ENABLE TX INTERRUPTS
2476 013020 002440 NOP
2477 013022 042777 000100 166364 AUAB: BIC #BIT6,@TXCSR ;OK IF NO INTERRUPT OCCURS. DISABLE INTERRUPTS
2478 013030 104012 SCOPE ;SCOPE
2479 013032 022626 AUAC: POPSP2 ;HERE IF INTERRUPT OCCURS. POP STOCK TWICE
2480 013034 104003 ERROR ;TX INTERRUPTED WITH PROCESSOR AT SAME
2481 013036 000771 BR AUAB ;PRIORITY AS THE TRANSMITTER
2482 013040 TSTA 10.,AVA,CD
2483 013040 TSTAA AVA,10.,\X+1+CD,\X+2,\X+1
;*****
2485 013040 100044 AT44: 100044 ;TEST NUMBER *
```

```
2486 013042 013124 AT45 ;ADDRESS OF NEXT TEST *
2487 013044 000012 10. ;ITERATION COUNT *
2488 013046 013054 AVAA ;SCOPE ENTRY POINT *
2489 000044 X=X+1 ;
;*****
;TEST THAT TRANSMITTER INTERRUPTS WHEN PROCESSOR IS AT PRIORITY ONE LEVEL
;LOWER THAN THE TRANSMITTER INTERRUPT PRIORITY.
2493 013050 104007 STTXV ;SET TX INTERRUPT SERVICE TO AVAB
2494 013052 013112 AVAB
2495 013054 042777 000100 166332 AVAA: BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPTS
2496 013062 013737 001426 177776 MOV TXLVL,PSW ;SET PROCESSOR PRIORITY TO ONE LEVEL
2497 013070 162737 000040 177776 SUB #40,PSW ;LOWER THAN TX PRIORITY
2498 013076 052777 000100 166310 BIS #BIT6,@TXCSR ;ENABLE TX INTERRUPTS
2499 013104 000240 NOP
2500 013106 104003 ERROR ;TX FAILED TO INTERRUPT
2501 013110 000401 BR AVAC
2502 013112 022626 AVAB: POPSP2 ;HERE IF INTERRUPT OCCURS. POP STOCK TWICE
2503 013114 042777 000100 166272 AVAC: BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPTS
2504 013122 104012 SCOPE ;SCOPE
2505 013124 TSTA 100.,AWA,CD
2506 013124 TSTAA AWA,100.,\X-1+CD,\X+2,\X+1
;*****
2508 013124 100045 AT45: 100045 ;TEST NUMBER *
2509 013126 013222 AT46 ;ADDRESS OF NEXT TEST *
2510 013130 000144 100. ;ITERATION COUNT *
2511 013132 013134 AWA ;SCOPE ENTRY POINT *
2512 000045 X=X+1 ;
;*****
;TEST THAT TRANSMITTER DOES NOT REINTERRUPT AFTER THE INITIAL INTERRUPT HAS
;OCCURRED AND HAS BEEN SERVICED.
2516 013134 104007 AWA: STTXV ;SET TX INTERRUPT SERVICE TO AWAC
2517 013136 013174 AWAC
2518 013140 042777 000100 166246 BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPTS
2519 013146 005037 177776 CLR PSW ;SET PROCESSOR PRIORITY TO 0
2520 013152 052777 000100 166234 BIS #BIT6,@TXCSR ;ENABLE TX INTERRUPTS
2521 013160 000240 NOP
2522 013162 104003 ERROR ;TRANSMITTER FAILED TO INTERRUPT
2523 013164 042777 000100 166222 AWAB: BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPTS
2524 013172 104012 SCOPE ;SCOPE
2525 013174 104277 013214 166222 AWAC: MOV #AWAE,@TXVTR ;HERE IF INTERRUPT OCCURS. CHANGE EXIT
2526 013202 012716 013210 MOV #AWAD,%X6 ;POINTER TO AWAD AND EXIT INTERRUPT
2527 013206 000002 RTI
2528 013210 000240 AWAD: NOP ;OK IF NO INTERRUPT REOCCURS.
2529 013212 000764 BR AWAB
2530 013214 022626 AWAE: POPSP2 ;HERE IF INTERRUPT REOCCURS
2531 013216 104003 ERROR ;TX REINTERRUPTED AFTER RTI
2532 013220 000761 BR AWAB
2533 013222 TSTA 10.,AXA,CD
2534 013222 TSTAA AXA,10.,\X+1+CD,\X+2,\X+1
;*****
2536 013222 100046 AT46: 100046 ;TEST NUMBER *
2537 013224 013306 AT47 ;ADDRESS OF NEXT TEST *
2538 013226 000012 10. ;ITERATION COUNT *
2539 013230 013246 AXAA ;SCOPE ENTRY POINT *
2540 000046 X=X+1 ;
;*****
2541
```

```
2542 ;TEST THAT RECEIVER DONE BIT IS ABLE TO INTERRUPT. IF THE INTERRUPT IS
2543 ;SERVICED IT WILL HAVE OCCURRED AT THE CORRECT VECTOR.
2544 013232 004737 003516 JSR 7,DVRLAY ;GO TO OVERLAY ROUTINE
2545 013236 104006 STRXV ;SET RX INTERRUPT SERVICE TO AXAB
2546 013240 013274 JSR AXAB
2547 013242 004737 014404 JSR %7,STRXD ;SET RX DONE BIT
2548 013246 042777 000100 166134 AXAA: BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
2549 013254 005037 177776 CLR PSW ;SET PROCESSOR PRIORITY TO 0
2550 013260 052777 000100 166122 BIS #BIT6,@RXCSR ;ENABLE RX INTERRUPTS
2551 013266 000240 NOP
2552 013270 104003 ERROR ;RX FAILED TO INTERRUPT
2553 013272 000401 BR AXAC
2554 013274 022626 AXAB: POPSP2 ;HERE IF INTERRUPT OCCURS
2555 013276 042777 000100 166104 AXAC: BIC #BIT6,@RXCSR ;DISABLE INT EN
2556 013304 104012 SCOPE ;SCOPE
2557 013306 TSTA 10.,AX1,0
2558 013306 TSTAA AX1,10.,\X+1+0,\X+2,\X+1
2559 ;*****
2560 013306 000047 AT47: 47 ;TEST NUMBER *
2561 013310 013370 AT50 ;ADDRESS OF NEXT TEST *
2562 013312 000012 10. ;ITERATION COUNT *
2563 013314 013326 AX1A ;SCOPE ENTRY POINT *
2564 000047 X=X+1 ;
2565 ;*****
2566 ;TEST THAT MODEM INTERRUPT BIT IS ABLE TO INTERRUPT. IF THE INTERRUPT IS
2567 ;SERVICED IT WILL HAVE OCCURRED AT THE CORRECT VECTOR.
2568 013316 004737 003516 JSR 7,DVRLAY ;GO TO OVERLAY ROUTINE
2569 013322 104006 STRXV ;SET RX INTERRUPT SERVICE TO AXAB
2570 013324 013356 AX1B
2571 013326 042777 000044 166054 AX1A: BIC #44,@RXCSR ;DISABLE MODEM INTERRUPTS
2572 013334 005037 177776 CLR PSW ;SET PROCESSOR PRIORITY TO 0
2573 013340 052777 000044 166044 BIS #44,@RXCSR ;ENABLE MODEM INTERRUPTS,RQ TO SND
2574 013346 104016 DELAY
2575 013350 000005 5.
2576 013352 104003 ERROR ;MODEM FAILED TO INTERRUPT
2577 013354 000401 BR AX1C
2578 013356 022626 AX1B: POPSP2 ;HERE IF INTERRUPT OCCURS
2579 013360 042777 000040 166022 AX1C: BIC #BIT5,@RXCSR ;DISABLE INT EN
2580 013366 104012 SCOPE
2581 013370 TSTA 1000.,AYA,CD
2582 013370 TSTAA AYA,1000.,\X+1+CD,\X+2,\X+1
2583 ;*****
2584 013370 100050 AT50: 100050 ;TEST NUMBER *
2585 013372 013452 AT51 ;ADDRESS OF NEXT TEST *
2586 013374 001750 100. ;ITERATION COUNT *
2587 013376 013410 AYAA ;SCOPE ENTRY POINT *
2588 000050 X=X+1 ;
2589 ;*****
2590 ;TEST THAT RECEIVER DONE BIT DOES NOT CAUSE AN INTERRUPT WHEN THE PROCESSOR
2591 ;IS AT THE SAME PRIORITY LEVEL AS THE RECEIVER INTERRUPT REQUEST LEVEL
2592 013400 104006 STRXV ;SET RX INTERRUPT SERVICE TO AYAC
2593 013402 013444 AYAC
2594 013404 004737 014404 JSR %7,STRXD ;SET RX DONE BIT
2595 013410 042777 000100 165772 AYAA: BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
2596 013416 013737 001422 177776 MOV RXLV,PSW ;SET PROCESSOR PRIORITY SAME AS RECEIVER'S
2597 013424 052777 000100 165756 BIS #BIT6,@RXCSR ;ENABLE RX INTERRUPTS
```

```
2598 013432 000240 NOP
2599 013434 042777 000100 165746 AYAB: BIC #BIT6,@RXCSR ;OK IF NO INTERRUPT. DISABLE RX INTERRUPTS
2600 013442 104012 SCOPE ;SCOPE
2601 013444 022626 AYAC: POPSP2 ;HERE IF INTERRUPT OCCURS. POP STOCK TWICE
2602 013446 104003 ERROR ;RX INTERRUPTED WITH PROCESOR AT SAME
2603 013450 000771 BR AYAB ;PRIORITY AS THE RECEIVER
2604 013452 TSTA 10.,AZA,CD
2605 013452 TSTAA AZA,10.,\X+1+CD,\X+2,\X+1
2606 ;*****
2607 013452 100051 AT51: 100051 ;TEST NUMBER *
2608 013454 013542 AT52 ;ADDRESS OF NEXT TEST *
2609 013456 000012 10. ;ITERATION COUNT *
2610 013460 013472 AZAA ;SCOPE ENTRY POINT *
2611 000051 X=X+1 ;
2612 ;*****
2613 ;TEST THAT RECEIVER DONE BIT CAUSES INTERRUPT WHEN PROCESSOR IS AT PRIORITY
2614 ;ONE LEVEL LOWER THAN THE RECEIVER'S INTERRUPT REQUEST LEVEL
2615 013462 104006 STRXV ;SET RX INTERRUPT TO AZAB
2616 013464 013530 AZAB
2617 013466 004737 014404 JSR %7,STRXD ;SET RX DONE BIT
2618 013472 042777 000100 165710 AZAA: BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
2619 013500 013737 001422 177776 MOV RXLV,PSW ;SET PROCESSOR PRIORITY ONE LEVEL
2620 013506 162737 000040 177776 SUB #40,PSW ;LOWER THAN RECEIVER'S PRIORITY
2621 013514 052777 000100 165666 BIS #BIT6,@RXCSR ;ENABLE RX INTERRUPTS
2622 013522 000240 NOP
2623 013524 104003 ERROR ;RX FAILED TO INTERRUPT WITH PROCESSOR AT
2624 013526 000401 BR AZAC ;PRIORITY ONE LEVEL LOWER THAN RECEIVER'S
2625 013530 022626 AZAB: POPSP2 ;HERE IF INTERRUPT OCCURS
2626 013532 042777 000100 165650 AZAC: BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
2627 013540 104012 SCOPE ;SCOPE
2628 013542 TSTA 100.,AAB,CD
2629 013542 TSTAA AAB,100.,\X+1+CD,\X+2,\X+1
2630 ;*****
2631 013542 100052 AT52: 100052 ;TEST NUMBER *
2632 013544 013640 AT53 ;ADDRESS OF NEXT TEST *
2633 013546 000144 100. ;ITERATION COUNT *
2634 013550 013556 AABA ;SCOPE ENTRY POINT *
2635 000052 X=X+1 ;
2636 ;*****
2637 ;TEST THAT RECEIVER DOES NOT INTERRUPT AFTER THE INITIAL INTERRUPT HAS
2638 ;OCCURED AND DONE BIT HAS NOT BEEN CLEARED
2639 013552 004737 014404 JSR %7,STRXD ;SET RX DONE BIT
2640 013556 104006 STRXV ;SET RX INTERRUPT SERVICE TO AABA
2641 013560 013612 AABC
2642 013562 042777 000100 165620 AABA: BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
2643 013570 052777 000100 165612 BIS #BIT6,@RXCSR ;ENABLE RX INTERRUPTS
2644 013576 000240 NOP
2645 013600 104003 ERROR ;RX FAILED TO INTERRUPT
2646 013602 042777 000100 165600 AABB: BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
2647 013610 104012 SCOPE ;SCOPE
2648 013612 012777 013632 165600 AABC: MOV #AAB,@RXVTR ;HERE IF INTERRUPT OCCURS. CHANGE SERVICE TO
2649 013620 012716 013626 MOV #AAB,@%6 ;AAB, SET EXIT POINTER TO AABD
2650 013624 000002 RTI ;EXIT INTERRUPT SERVICE
2651 013626 000240 AABD: NOP ;OK IF NO INTERRUPT REOCCURS
2652 013630 000764 BR AABB
2653 013632 022626 AABE: POPSP2 ;HERE IF INTERRUPT REOCCURS
```

```
2654 013634 104003 ERROR ;RX REINTERRUPTED AFTER RTI
2655 013636 000761 BR AAB B ;
2656 013640 TSTA 100.,ABB,CD ;
2657 013640 TSTAA ABB,100.,\X+1+CD,\X+2,\X+1 ;
;*****
2658 ;
2659 013640 100053 AT53: 100053 ;TEST NUMBER *
2660 013642 013700 AT54 ;ADDRESS OF NEXT TEST *
2661 013644 000144 100. ;ITERATION COUNT *
2662 013646 013650 ABBA ;SCOPE ENTRY POINT *
2663 000053 X=X+1 ; *
2664 ; *
2665 ;*****
;TEST THAT READING RXCSR DOES NOT CLEAR DONE BIT (RXCSR BIT 7)
2666 013650 004737 014404 ABBA: JSR %7,STRXD ;SET RX DONE BIT
2667 013654 017737 165530 001606 MOV @RXCSR,RXCSRT ;SAVE CONTENT OF RXCSR
2668 013656 105777 165522 TSTB @RXCSR ;SEE IF DONE BIT IS CLEAR
2669 013656 104001 BMI ABBB ;BRANCH IF DONE BIT IS NOT CLEAR
2670 013670 104003 ERROR ;
2671 013672 005777 165514 ABBB: TST @RXBUF ;CLEAR DONE BIT IF SET
2672 013676 104012 SCOPE ;SCOPE
2673 013700 TSTA 100.,ACB,CD ;
2674 013700 TSTAA ACB,100.,\X+1+CD,\X+2,\X+1 ;
;*****
2675 ;
2676 013700 100054 AT54: 100054 ;TEST NUMBER *
2677 013702 013764 AT55 ;ADDRESS OF NEXT TEST *
2678 013704 000144 100. ;ITERATION COUNT *
2679 013706 013714 ACBA ;SCOPE ENTRY POINT *
2680 000054 X=X+1 ; *
2681 ; *
2682 ;*****
;TEST THAT DONE CAN CAUSE INT WITH ERROR SET
2683 013710 104006 STRXV ;SET RX INTERRUPT SERVICE TO ACBB.
2684 013712 013752 ACBB ;
2685 013714 004737 014404 ACBA: JSR %7,STRXD ;SET RX DONE BIT
2686 013720 004737 014404 JSR %7,STRXD ;SET RX DATA OFLOW
2687 013724 042777 000100 165456 BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
2688 013732 005037 177776 CLR PSW ;SET PROCESSOR PRIORITY TO 0
2689 013736 052777 000100 165444 BIS #BIT6,@RXCSR ;ENABLE RX INTERRUPTS
2690 013744 000240 NOP ;
2691 013746 104003 ERROR ;RX DONE FAILED TO CAUSE INTERRUPT
2692 013750 000401 BR ACBC ;
2693 013752 022626 ACBB: POPSP2 ;HERE IF INTERRUPT OCCURS. POP STOCK TWICE
2694 013754 042777 000100 165426 ACBC: BIC #BIT6,@RXCSR ;
2695 013762 104012 SCOPE ;
2696 013764 TSTA 3.,ADD,CD ;
2697 013764 TSTAA ADD,3.,\X+1+CD,\X+2,\X+1 ;
;*****
2698 ;
2699 013764 100055 AT55: 100055 ;TEST NUMBER *
2700 013766 014006 AT56 ;ADDRESS OF NEXT TEST *
2701 013770 000003 3. ;ITERATION COUNT *
2702 013772 014000 ADDA ;SCOPE ENTRY POINT *
2703 000055 X=X+1 ; *
2704 ; *
2705 ;*****
;DATA TEST USING NORMAL CONFIGURATION
2706 013774 004737 002706 ADDA: JSR %7,CDINIT ;INIT IF C-D DEVICE
2707 014000 004537 004770 JSR 5,DATTST ;
2708 014004 104012 SCOPE ;
2709 014006 TSTA 3.,APB,0 ;
```

```
2710 014006 TSTAA APB,3.,\X+1+0,\X+2,\X+1 ;
2711 ;*****
2712 014006 000056 AT56: 56 ;TEST NUMBER *
2713 014010 014100 AT57 ;ADDRESS OF NEXT TEST *
2714 014012 000003 3. ;ITERATION COUNT *
2715 014014 014022 APBA ;SCOPE ENTRY POINT *
2716 000056 X=X+1 ; *
2717 ; *
2718 ;*****
;DATA TEST USING JUMPER CONNECTOR.
2719 ;USES SPECIAL BINARY COUNT PATTERN FOR DATA. NO INTERRUPT.
2720 014016 004737 004322 JSR 7,INBIN ;INITIALIZE BINARY COUNT PATTERN
2721 014022 012737 001750 001564 APBA: MOV #1000.,CTRO ;SET CHARACTER COUNT TO 1000
2722 014030 104020 APBB: TIMETX ;TIME OUT TX DONE
2723 014032 104003 ERROR ;ERROR DONE NOT SETTING
2724 014034 004737 004426 JSR 7,GTBINP ;GET BINARY CHARACTER
2725 014040 110137 001560 MOVB %1,CBUBFA ;SAVE CHAR IN CRUBFA AND
2726 014044 004737 005374 JSR 7,MASKIT ;MASK OFF NON TRANSMITTED BITS
2727 014050 110177 165342 MOVB %1,@TXBUF ;LOAD CHAR.
2728 014054 104017 TIMERX ;TIME OUT RX DONE
2729 014056 104003 ERROR ;ERROR DONE NOT SETTING
2730 014060 117737 165326 001556 MOVB @RXBUF,CRBUF ;LOAD RECEIVED DATA INTO CRBUF
2731 014066 104004 DATCHK ;CHECK DATA
2732 014070 005337 001564 DEC CTRO ;TESTED 1000 CHARACTERS
2733 014074 001355 BNE APBB ;BRANCH IF NOT
2734 014076 104012 SCOPE ;YES. SCOPE
2735 014100 TSTA 3.,EXT,0 ;
2736 014100 TSTAA EXT,3.,\X+1+0,\X+2,\X+1 ;
;*****
2737 ;
2738 014100 000057 AT57: 57 ;TEST NUMBER *
2739 014102 014164 AT60 ;ADDRESS OF NEXT TEST *
2740 014104 000003 3. ;ITERATION COUNT *
2741 014106 000057 EXTA ;SCOPE ENTRY POINT *
2742 000057 X=X+1 ; *
2743 ; *
2744 ;*****
;TEST THAT RDR BUSY TURNS OFF RDR ENABLE
;WHEN RUN ON AN XOR TESTER
2745 ;
2746 ;
2747 014110 000005 EXTA: RESET ;RESET
2748 014112 005277 165272 INC @RXCSR ;SET RDR ENABLE, SEE IF RDE IS TURNED OFF BY RDR BUSY
2749 014116 012737 177770 014154 MOV #-10,3$+2 ;
2750 014124 005237 014154 2$: INC 3$+2 ;WAIT LOOP FOR XOR TESTER
2751 014130 001375 BNE 2$ ;
2752 014132 005077 165260 CLR @TXBUF ;SHIP OUT CHAR.
2753 014136 012737 130000 014154 MOV #-50000,3$+2 ;
2754 014144 105777 165240 5$: TSTB @RXCSR ;TEST COMPLETE
2755 014150 104004 BMI 6$ ;
2756 014152 005227 177770 3$: INC #-10 ;ALLOW TIME FOR RDR DONE TO SET
2757 014156 001372 BNE 5$ ;
2758 014160 104003 ERROR ;FAILURE OF RDR DONE TO SET
2759 014162 104012 6$: SCOPE ;
2760 014164 TSTA 10.,EX,0 ;
2761 014164 TSTAA EX,10.,\X+1+0,\X+2,\X+1 ;
;*****
2762 ;
2763 014164 000060 AT60: 60 ;TEST NUMBER *
2764 014166 014234 AT61 ;ADDRESS OF NEXT TEST *
2765 014170 000012 10. ;ITERATION COUNT *
```

```
2766 014172 014174          EXA          ;SCOPE ENTRY POINT          *
2767          000060          X=X+1          ;
2768          ;*****
2769          ;TEST THAT WHEN RDR ENABLE IS SET THAT THE RXCSR DONE
2770          ;BIT IS CLEARED
2771 014174 000005          EXA:  RESET
2772 014176 004737 014404          JSR  PC,STRXD          ;SET RCVR DONE
2773 014202 005277 165202          INC  @RXCSR          ;SET ENABLE
2774 014206 105777 165176          TSTB @RXCSR          ;DONE SHOULD CLEAR
2775 014212 100001          BPL  1$
2776 014214 104003          ERROR          ;DONE NOT CLEAR
2777 014216 012737 177770 014226 1$:  MOV  #-10,3$+2
2778 014224 005227 177770 3$:  INC  #-10          ;WAIT 100MIC. SEC. FOR XDR
2779 014230 001375          BNE  3$
2780 014232 104012          SCOPE
2781 014234          TSTA  3.,EXA,0
2782 014234          TSTAA EXA,3.,\X+1+0,\X+2,\X+1
2783          ;*****
2784 014234 000061          AT61:  61          ;TEST NUMBER          *
2785 014236 014270          AT62          ;ADDRESS OF NEXT TEST          *
2786 014240 000003          3.          ;ITERATION COUNT          *
2787 014242 014244          EXAA          ;SCOPE ENTRY POINT          *
2788          X=X+1          ;
2789          ;*****
2790 014244 005737 002016          EXAA:  TST  XORFLG          ;CHECKING JUMPER CONNECTIONS FOR XOR, RCVR
2791 014250 100006          BPL  3$
2792 014252 012777 177777 165130          MOV  #-1,@RXCSR
2793 014260 005777 165124          TST  @RXCSR
2794 014264 000005          RESET
2795 014266 104012          3$:  SCOPE
2796 014270          TSTA  3.,EXB,0
2797 014270          TSTAA EXB,3.,\X+1+0,\X+2,\X+1
2798          ;*****
2799 014270 000062          AT62:  62          ;TEST NUMBER          *
2800 014272 014324          AT63          ;ADDRESS OF NEXT TEST          *
2801 014274 000003          3.          ;ITERATION COUNT          *
2802 014276 014300          EXBA          ;SCOPE ENTRY POINT          *
2803          X=X+1          ;
2804          ;*****
2805 014300 005737 002016          EXBA:  TST  XORFLG          ;SAME AS ABOVE BUT FOR XMTR
2806 014304 100006          BPL  4$
2807 014306 012777 177677 165100          MOV  #177677,@TXCSR
2808 014314 005777 165074          TST  @TXCSR
2809 014320 000005          RESET
2810 014322 104012          4$:  SCOPE
2811 014324          TSTAA AQB,10.,\X+1+CD, LAST,\X+1
2812          ;*****
2813 014324 100063          AT63:  100063          ;TEST NUMBER          *
2814 014326 177777          ATLAST          ;ADDRESS OF NEXT TEST          *
2815 014330 000012          10.          ;ITERATION COUNT          *
2816 014332 014334          AQBA          ;SCOPE ENTRY POINT          *
2817          X=X+1          ;
2818          ;*****
2819          ;TEST THAT WHEN TXCSR BIT 0 IS SET THAT THE OUTPUT DATA LINE
2820          ;IS PULLED TO A SPACE.
2821 014334 004737 002706          AQBA:  JSR  %7,CDINIT          ;INIT IF C-D DEVICE
```

```
2822 014340 052777 000004 165046          BIS  #BIT2,@TXCSR          ;SET MAINTENANCE BIT IN TXCSR
2823 014346 052777 000001 165040          BIS  #BIT0,@XCSR          ;SET BREAK BIT
2824 014354 012777 000252 165034          MOV  #252,@TXBUF          ;LOAD BUFFER
2825 014362 104017          TIMERX          ;TIME OUT RX DONE
2826 014364 104003          ERROR          ;ERROR DONE NOT SETTING
2827 014366 127727 165020 000000          CMPB @RXBUF,#0          ;CHARACTER RECEIVED SHOULD BE 0
2828 014374 001401          BEQ  .+4
2829 014376 104003          ERROR          ;CHARACTER OTHER THAN 0
2830 014400 104011          SRESET          ;ISSUE RESET
2831 014402 104012          SCOPE
```

```
2832 ;
2833 ;SUBROUTINE TO SET RXCSR DONE BIT.
2834 014404 052777 000004 165002 STRXD: BIS #BIT2,@TXCSR ;SET MAINTENANCE BIT.
2835 014412 005077 165000 CLR @TXBUF ;LOAD TXBUF.
2836 014416 104017 TIMERX ;TIME OUT TX DONE
2837 014420 104003 ERROR ;ERROR DONE NOT SETTING
2838 014422 000207 RTS %7 ;EXIT.
2839 ;SUBROUTINE TO CHECK THAT CTR0 THROUGH CTR3 CONTAIN DESCENDING VALUES.
2840 014424 023737 001564 001566 CMPT: CMP CTR0,CTR1
2841 014432 101430 BLOS CMPTNG
2842 014434 023737 001566 001570 CMP CTR1,CTR2
2843 014442 101424 BLOS CMPTNG
2844 014444 023737 001570 001572 CMP CTR2,CTR3
2845 014452 101420 BLOS CMPTNG
2846 014454 023737 001572 001574 CMP CTR3,CTR4
2847 014462 101414 BLOS CMPTNG
2848 014464 023737 001574 001576 CMP CTR4,CTR5
2849 014472 101410 BLOS CMPTNG
2850 014474 023737 001576 001600 CMP CTR5,CTR6
2851 014502 101404 BLOS CMPTNG
2852 014504 023737 001600 001602 CMP CTR6,CTR7
2853 014512 101002 BHI CMPTOK
2854 014514 062716 000002 CMPTNG: ADD #2,%6
2855 014520 000207 CMPTOK: RTS %7
```

```
2856
2857 ;*****
2858 ;PRG1 - TRANSMITTER SCOPE LOOP
2859 ;*****
2860 014522 104000 PRG1: TYPE ;TYPE PROGRAM TITLE.
2861 014524 015632 P2IIT
2862 014526 004537 004132 JSR 5,LINSLX ;GO GET LINE # FROM USER
2863 014532 104000 TYPE ;TYPE SELECT CHAR AND DELAY.
2864 014534 015727 SELCAD
2865 014536 004737 003554 JSR PC,RDOCT ;READ IN DATA.
2866 014542 012637 001622 MOV (SP)+,TEMP2 ;STORE DATA.
2867 014546 113737 001622 014570 PRG1A: MOVB TEMP2,PRG1B ;DELAY COUNT TO PRG1B.
2868 014554 113777 001623 164634 MOVB TEMP2+1,@TXBUF ;LOAD TXBUF.
2869 014562 105777 164626 TSTB @TXCSR ;TEST FOR DONE. ;++;G
2870 014566 104016 DELAY ;DELAY # OF MSECS. SET AT SR.
2871 014570 000000 PRG1B: OPEN
2872 014572 000765 BR PRG1A ;REPEAT.
2873 ;*****
2874 ;PRG2 - RECEIVER SCOPE LOOP.
2875 ;*****
2876 014574 104000 PRG2: TYPE ;TYPE PROGRAM TITLE.
2877 014576 015672 P2IIT
2878 014600 004537 004132 JSR 5,LINSLX ;GO GET LINE # FROM USER
2879 014604 104000 TYPE ;TYPE SELECT CHAR AND DELAY.
2880 014606 015727 SELCAD
2881 014610 004737 003554 JSR PC,RDOCT ;READ IN DATA.
2882 014614 012637 001622 MOV (SP)+,TEMP2 ;STORE DATA.
2883 014620 052777 000004 164566 PRG2A: BIS #BIT2,@TXCSR ;SET MAINTENANCE BIT.
2884 014626 113737 001622 014650 MOVB TEMP2,PRG2B ;DELAY COUNT TO PRG2B.
2885 014534 113777 001623 164554 MOVB TEMP2+1,@TXBUF ;LOAD TXBUF.
2886 014642 105777 164546 TSTB @TXCSR ;TEST FOR DONE ;++;G
2887 014646 104016 DELAY ;DELAY # OF MSECS. SET IN SR.
2888 014650 000000 PRG2B: OPEN
2889 014652 017700 164534 MOV @RXBUF,%0 ;RXBUF CONTENTS TO R0.
2890 014656 000005 RESET ;DISPLAY CONTENTS OF RXBUF (IN R0).
2891 014660 000005 RESET ;BY ISSUING 5 RESET INSTRUCTIONS
2892 014662 000005 RESET
2893 014664 000005 RESET
2894 014666 000005 RESET
2895 014670 000753 BR PRG2A
```

```
2896  
2897  
2898  
2899  
2900 014672 104000  
2901 014674 016475  
2902 014676 004537 004132  
2903 014702 104000  
2904 014704 016620  
2905 014706 004737 003554  
2906 014712 012637 001622  
2907 014716 113737 001623 001560  
2908 014724 004737 014764  
2909 014730 000772  
2910  
2911  
2912  
2913 014732 104000  
2914 014734 016545  
2915 014736 004537 004132  
2916 014742 004737 004322  
2917 014746 004737 004426  
2918 014752 110137 001560  
2919 014756 004737 014764  
2920 014762 000771  
2921  
2922 014764 032777 000200 163202  
2923 014772 001001  
2924 014774 104002  
2925 014776 104020  
2926 015000 104003  
2927 015002 052777 000004 164404  
2928 015010 005777 164376  
2929 015014 013777 001560 164374  
2930 015022 004737 005374  
2931 015026 104017  
2932 015030 104003  
2933 015032 017737 164354 001556  
2934 015040 104004  
2935 015042 000207  
;*****  
;PRG3 - SINGLE CHARACTER MAINTENANCE MODE DATA TEST.  
;*****  
PRG3: TYPE ;TYPE PROGRAM TITLE.  
P3TIT  
JSR 5,LINSLX ;GO GET LINE # FROM USER  
TYPE ;TYPE: SELECT CHARACTER.  
SELCAR  
JSR PC,RDOCT ;GET TEST CHAR AND DELAY FROM USER.  
MOV (SP)+,TEMP2 ;STORE TEST CHAR AND DELAY.  
PRG3A: MOVB TEMP2+1,CRBUFA ;MOVE DATA CHAR TO CRBUFA.  
JSR %7,MOUTIN ;GO OUTPUT, RECEIVE, AND CHECK DATA.  
BR PRG3A  
;*****  
;PRG4 - SPECIAL BINARY COUNT MAINTENANCE MODE DATA TEST.  
;*****  
PRG4: TYPE ;TYPE PROGRAM TITLE.  
P4TIT  
JSR 5,LINSLX ;GO GET LINE # FROM USER  
JSR %7,INBIN ;INITIALIZE BINARY COUNT.  
PRG4A: JSR %7,GTBINP ;GET BINARY CHARACTER.  
MOVB %1,CRBUFA ;SAVE AT CRBUFA.  
JSR %7,MOUTIN ;GO OUTPUT, RECEIVE, AND CHECK DATA.  
BR PRG4A ;REPEAT.  
;SUBROUTINE TO OUTPUT, RECEIVE, AND CHECK DATA WITH MAINTENANCE BIT SET.  
MOUTIN: BIT #BIT7,@SRPTR ;SEE IF BIT 7 IS SET.  
BNE .+4 ;BRANCH IF SET.  
STALL ;SET. DO A RANDOM STALL.  
TIMETX ;TIME OUT TX DONE  
ERROR ;ERROR DONE NOT SETTING  
BIS #BIT2,@TXCSR ;SET MAINTENANCE BIT.  
TST @RXBUF ;CLR RX DONE  
MOVB CRBUFA,@TXBUF ;LOAD TXBUF.  
JSR 7,MASKIT ;MASK OFF NON TRANSMITTED BITS  
TIMERX ;TIME OUT RX DONE  
ERROR ;ERROR DONE NOT SETTING  
MOVB @RXBUF,CRBUF ;MOVE CHAR IN RX BUFFER TO CRBUF.  
DATCHK ;COMPARE EXPECTED AND RECEIVED DATA  
RTS %7 ;EXIT.
```

```
2936  
2937  
2938  
2939 015044 042045 030514 026461  
2940 015052 026105 027503 020104  
2941 015060 043117 047114 020105  
2942 015066 051524 020124 020055  
2943 015074 055103 046104 026501  
2944 015102 022507  
2945 015104 046445 050101 047440  
2946 015112 020106 042504 044526  
2947 015120 042503 020123 051120  
2948 015126 051505 047105 022524  
2949 015134 100  
2950 015135 040 020040 020040 MDEVAD: .ASCII ' %'  
2951 015142 020040 020040 020040  
2952 015150 020040 022440 100  
2953 015155 045 054524 042520 PGMSG: .ASCII '%TYPE IN PROGRAM NUMBER @'  
2954 015162 044440 020116 051120  
2955 015170 043517 040522 020115  
2956 015176 052516 041115 051105  
2957 015204 020040 020040 100  
2958 015211 045 047516 042516 MNONE: .ASCII '%NONE FOUND%'  
2959 015216 043040 052517 042116  
2960 015224 040045  
2961 015226 052045  
2962 015230 020040 020040 041520 EMO: .ASCII '%T'  
ATNUMB: .ASCII ' PC= '  
2963 015236 020075  
2964 015240 020040 020040 020040 APC: .ASCII ' RXCSR= '  
2965 015246 020040 054122 051503  
2966 015254 036522 040  
2967 015257 040 020040 020040 MRXNUM: .ASCII ' @'  
2968 015264 020040 040040  
2969 015270 022445 051120 030107 P0TIT: .ASCII '%%PRGO - INPUT-OUTPUT LOGIC TESTS. '  
2970 015276 026440 044440 050116  
2971 015304 052125 047455 052125  
2972 015312 052520 020124 047514  
2973 015320 044507 020103 042524  
2974 015326 052123 027123 040  
2975 015333 045 044504 041523 .ASCII '%DISCONNECT DL11-E FROM MODEM'  
2976 015340 047117 042516 052103  
2977 015346 042040 030514 026461  
2978 015354 020105 051106 046517  
2979 015362 046440 042117 046505  
2980 015370 040440 042116 041440 .ASCII ' AND CONNECT JUMPER TO CABLE.%'  
2981 015376 047117 042516 052103  
2982 015404 045040 046525 042520  
2983 015412 020122 047524 041440  
2984 015420 041101 042514 022456  
2985 015426 100  
2986 015427 124 041530 051123 ATXCSR: .ASCII '%TXCSR S/B: '  
2987 015434 051440 041057 020072  
2988 015442 020040 020040 020040 ATXSB: .ASCII ' WAS: '  
2989 015450 020040 040527 035123  
2990 015456 040  
2991 015457 040 020040 020040 ATXWAS: .ASCII ' @'
```



```
2992 015464 040040
2993 015466 054122 051503 020122 ARXCSR: .ASCII 'RXCSR S/B: '
2994 015474 027523 035102 040
2995 015501 040 020040 020040 ARXSB: .ASCII ' WAS: '
2996 015506 020040 053440 051501
2997 015514 020072
2998 015516 020040 020040 020040 ARXWAS: .ASCII ' @'
2999 015524 040040
3000 015526 054124 051440 042520 ETXTIM: .ASCII 'TX SPEEDS NOT IN ASCENDING ORDER.@'
3001 015534 042105 020123 047516
3002 015542 020124 047111 040440
3003 015550 041523 047105 044504
3004 015556 043516 047440 042122
3005 015564 051105 040056
3006 015570 054122 051440 042520 ERXTIM: .ASCII 'RX SPEEDS NOT IN ASCENDING ORDER.@'
3007 015576 042105 020123 047516
3008 015604 020124 047111 040440
3009 015612 041523 047105 044504
3010 015620 043516 047440 042122
3011 015626 051105 040056
3012 015632 022445 051120 030507 P1TIT: .ASCII '%XPRG1 - TRANSMITTER SCOPE LOOP@'
3013 015640 026440 052040 040522
3014 015646 051516 044515 052124
3015 015654 051105 051440 047503
3016 015662 042520 046040 047517
3017 015670 040120
3018 015672 022445 051120 031107 P2TIT: .ASCII '%XPRG2 - RECEIVER SCOPE LOOP@'
3019 015700 026440 051040 041505
3020 015706 044505 042526 020122
3021 015714 041523 050117 020105
3022 015722 047514 050117 100
3023 015727 045 054524 042520 SELCAD: .ASCII '%XTYPE TEST CHAR. CODE IN BITS 15-8,TYPE DELAY TIME IN BITS 7-0'
3024 015734 052040 051505 020121
3025 015742 044103 051101 020056
3026 015750 047503 042504 044440
3027 015756 020116 044502 051524
3028 015764 030440 026465 026070
3029 015772 054524 042520 042040
3030 016000 046105 054501 052040
3031 016006 046511 020105 047111
3032 016014 041040 052111 020123
3033 016022 026467 060
3034 016025 045 043117 040440 .ASCII '%OF AN OCTAL WORD.%@'
3035 016032 020116 041517 040524
3036 016040 020114 047527 042122
3037 016046 022456 100
3038 016051 104 052101 020101 ERDAT: .ASCII 'DATA S/B: '
3039 016056 027523 035102 040
3040 016063 040 020040 053440 AASB: .ASCII ' WAS: '
3041 016070 051501 020072
3042 016074 020040 040
3043 016077 040 054122 052502 AWAS: .ASCII ' '
3044 016104 035106 040 .ASCII ' RXBUF: '
3045 016107 040 020040 020040 ARXBUF: .ASCII ' @'
3046 016114 020040 100
3047 016117 045 042523 020124 ASETSR: .ASCII '%SET DESIRED SR OPTIONS. NORMAL OPERATION '
```

```
3048 016124 042504 044523 042522
3049 016132 020104 051123 047440
3050 016140 052120 047511 051516
3051 016146 020056 047516 046522
3052 016154 046101 047440 042520
3053 016162 040522 044524 047117
3054 016170 040
3055 016171 111 020123 044527 .ASCII '%IS WITH SR = 000000%@'
3056 016176 044124 051440 020122
3057 016204 020075 030060 030060
3058 016212 030060 040045
3059 016216 044445 041516 051117 AINCRT: .ASCII '%INCORRECT ROUTINE SELECTED, PLACE CORRECT PROGRAM'
3060 016224 042522 052103 051040
3061 016232 052517 044524 042516
3062 016240 051440 046105 041505
3063 016246 042524 026104 050040
3064 016254 040514 042503 041440
3065 016262 051117 042522 052103
3066 016270 050040 047522 051107
3067 016276 046501
3068 016300 044445 020116 051123 .ASCII '%IN SR 0-1 AND PRESS CONTINUE.@'
3069 016306 030040 031055 040440
3070 016314 042116 050040 042522
3071 016322 051523 041440 047117
3072 016330 044524 052516 027105
3073 016336 100
3074 016337 045 047111 040526 AINCPG: .ASCII '%INVALID PROGRAM SELECTED.@'
3075 016344 044514 020104 051120
3076 016352 043517 040522 020115
3077 016360 042523 042514 052103
3078 016366 042105 040056
3079 016372 207
3080 016373 045 047105 020104 APGEND: .BYTE 207
3081 016400 040520 051523 036440 .ASCII '%END PASS = '
3082 016406 040
3083 016407 040 020040 020040 APCNT: .ASCII ' LINE = '
3084 016414 020040 046040 047111
3085 016422 020105 020075 040
3086 016427 040 020040 051040 ACLIN: .ASCII ' RXCSR = '
3087 016434 041530 051123 036440
3088 016442 040
3089 016443 040 020040 020040 APRXC: .ASCII ' VECTOR = '
3090 016450 020040 053040 041505
3091 016456 047524 020122 020075
3092 016464 020040 020040 020040 APVEC: .ASCII ' @'
3093 016472 020040 100
3094 016475 045 050045 043522 P3TIT: .ASCII '%XPRG3-SINGLE CHAR MAINT MODE DATA TEST@'
3095 016502 026463 044523 043516
3096 016510 042514 041440 040510
3097 016516 020122 040515 047111
3098 016524 020124 047515 042504
3099 016532 042040 052101 020101
3100 016540 042524 052123 100
3101 016545 045 050045 043522 P4TIT: .ASCII '%XPRG4-SPEC BIN COUNT MAINT MODE DATA TEST@'
3102 016552 026464 050123 041505
3103 016560 041040 047111 041440
```

3104	016566	052517	052116	046440						
3105	016574	044501	052116	046440						
3106	016602	042117	020105	040504						
3107	016610	040524	052040	051505						
3108	016616	040124								
3109	016620	052045	050131	020105	SELCAR: .ASCII	'%TYPE IN TEST CHAR. CODE.		%@'		
3110	016626	047111	052040	051505						
3111	016634	020124	044103	051101						
3112	016642	020056	047503	042504						
3113	016650	020056	020040	022440						
3114	016656	100								
3115	016657	045	054524	042520	LDLINE: .ASCII	'%TYPE IN LINE NO.		@'		
3116	016664	044440	020116	044514						
3117	016672	042516	047040	027117						
3118	016700	020040	020040	100						
3119	016705	045	044514	042516	ALINE: .ASCII	'%LINE NO.'				
3120	016712	047040	027117							
3121	016716	020040	053440	051501	SELINE: .ASCII	' WAS SELECTED@'				
3122	016724	051440	046105	041505						
3123	016732	042524	040104							
3124	016735	051045	041505	044505	MSETRX: .ASCII	'%RECEIVER SPEED CHECK@'				
3125	016744	042526	020122	050123						
3126	016752	042505	020104	044103						
3127	016760	041505	040113							
3128	016764	052045	040522	051516	MSETTX: .ASCII	'%TRANSMIT SPEED CHECK@'				
3129	016772	044515	020124	050123						
3130	017000	042505	020104	044103						
3131	017006	041505	040113							
3132	017012	051445	052105	041440	MSETC: .ASCII	'%SET CLOCK SWITCHES TO POSITION, THEN PRESS CONTINUE.@'				
3133	017020	047514	045503	051440						
3134	017026	044527	041524	042510						
3135	017034	020123	047524	050040						
3136	017042	051517	052111	047511						
3137	017050	026116	052040	042510						
3138	017056	020116	051120	051505						
3139	017064	020123	047503	052116						
3140	017072	047111	042525	040056						
3141	017100	042445	051122	051117	MTERR: .ASCII	'%ERROR - UNEXPECTED TRAP'				
3142	017106	026440	052440	042516						
3143	017114	050130	041505	042524						
3144	017122	020104	051124	050101						
3145	017130	052045	040522	050120	.ASCII	'%TRAPPED TO '				
3146	017136	042105	052040	020117						
3147	017144	040								
3148	017145	040	020040	020040	MTD: .ASCII	' '				
3149	017152	020040	040							
3150	017155	045	051124	050101	.ASCII	'%TRAPPED FROM PC '				
3151	017162	042520	020104	051106						
3152	017170	046517	050040	020103						
3153	017176	040								
3154	017177	040	020040	020040	MFROM: .ASCII	' '@'				
3155	017204	020040	040040							
3156	017210	047045	020117	042504	MNOLIN: .ASCII	'%NO DEVICE PRESENT - THIS LINE NO.@'				
3157	017216	044526	042503	050040						
3158	017224	042522	042523	052116						
3159	017232	026440	052040	044510						

3160	017240	020123	044514	042516						
3161	017246	047040	027117	100						
3162	017253	077	040045		DTERR: .ASCII	'?%@'				
3163	017256	047045	020117	047111	INTER: .ASCII	'%NO INTERRUPT@'				
3164	017264	042524	051122	050125						
3165	017272	040124								
3166	017274	041445	020123	020075	MS0: .ASCII	'%CS = 0@'				
3167	017302	040060								
3168	017304	041445	020123	020075	MS1: .ASCII	'%CS = 1@'				
3169	017312	040061								
3170	017314	041445	020123	020075	MS2: .ASCII	'%CS = 2@'				
3171	017322	040062								
3172	017324	041445	020123	020075	MS3: .ASCII	'%CS = 3@'				
3173	017332	040063								
3174	017334	041445	020123	020075	MS4: .ASCII	'%CS = 4@'				
3175	017342	040064								
3176	017344	041445	020123	020075	MS5: .ASCII	'%CS = 5@'				
3177	017352	040065								
3178	017354	041445	020123	020075	MS6: .ASCII	'%CS = 6@'				
3179	017362	040066								
3180	017364	041445	020123	020075	MS7: .ASCII	'%CS = 7@'				
3181	017372	040067								
3182	017374	051045	041505	053117	MPWRF: .ASCII	'%RECOVERED FROM POWER FAILURE@'				
3183	017402	051105	042105	043040						
3184	017410	047522	020115	047520						
3185	017416	042527	020122	040506						
3186	017424	046111	051125	040105						
3187					.EVEN					
3188		000001			.END					

XORFLG	002016	670*	680#	685*	800	1523	2790	2805										
.	= 017432	401#	418	420#	426	481#	484#	486#	488#	679#	753	954	1133	1142				
		1223	1229	1981	2163	2230	2233	2387	2410	2414	2419	2424	2428	2828				
		2923																

. ABS. 017432 000

ERRORS DETECTED: 0

CZDLAG,CZDLAG=CZDLAG.P11
RUN-TIME: 3 7 1 SECONDS
RUN-TIME RATIO: 49/13=3.6
CORE USED: 11K (21 PAGES)

DOCUMENT PAGES: 74