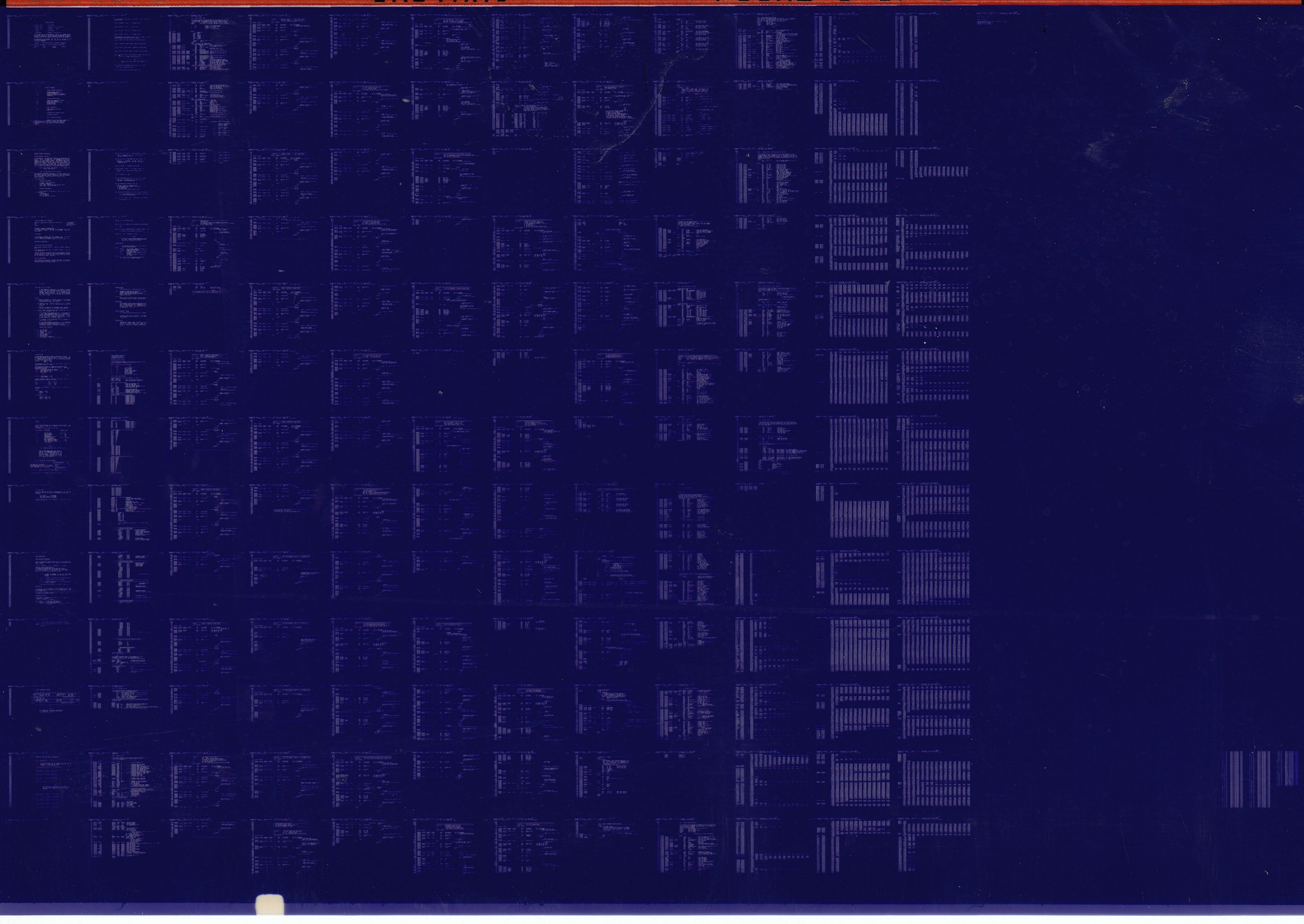
DLV11-E

OFF LINE TEST
CNDVAAO

AH-T436A-MC
FICHE 1 OF 1



```
8213                            .REM    @
8214
8215
8216
8217                                            IDENTIFICATION
8218                                            --------------
8219
8220                                    PRODUCT CODE:           AC-T435A-MC
8221
8222                                    PRODUCT NAME:           CNDVAA0 DLV11-E OFFLINE TEST
8223
8224                                    PRODUCT DATE:           DECEMBER,     1982
8225
8226                                    AUTHOR:                 ODES CHOATE
8227
8228                                    MAINTAINER:             DIAGNOSTIC SERVICES/ISS
8229
8230                                    THE INFORMATION IN THIS DOCUMENT IS  SUBJECT  TO  CHANGE WITHOUT
8231                                    NOTICE  AND  SHOULD  NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
8232                                    EQUIPMENT CORPORATION.  DIGITAL EQUIPMENT CORPORATION ASSUMES NO
8233                                    RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.
8234
8235                                    NO RESPONSIBILITY IS ASSUMED  FOR  THE  USE  OR  RELIABILITY  OF
8236                                    SOFTWARE ON EQUIPMENT THAT IS NOT  SUPPLIED  BY  DIGITAL  OR ITS
8237                                    AFFILIATED COMPANIES.
8238
8239                                    COPYRIGHT (C) 1982,1983 BY DIGITAL EQUIPMENT CORPORATION
8240
8241                            THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:
8242
8243                                    DIGITAL         PDP             UNIBUS          MASSBUS
8244                                    DEC             DECUS           DECTAPE
8245
```

TABLE OF CONTENTS

NOTE: CVDVAC DIAGNOSTIC WAS MODIFIED TO RUN ON 11/21 PROCESSOR
BY LOWERING PRIORITY 7 TO 6 AND ALSO ADDING A MACRO CALL
TO CNMAC2.SML TO INIT BRKVEC AND LKVEC AND WAS RENAMED
TO CNDVAA.

```
8291
8292                    1.0     GENERAL PROGRAM INFORMATION.
8293                            ------- ------- -----------
8294
8295                    1.1     PROGRAM PURPOSE (ABSTRACT).
8296
8297                            THIS DIAGNOSTIC IS A LOGIC TEST TO VERIFY THE OPERATION OF THE
8298                            DLV11-E  SERIAL  LINE INTERFACE.  THE PROGRAM AS SET INITIALLY
8299                            DEFAULTS  TO  ALL  OPTIONS,  EXCEPT  PROGRAMMABLE  BAUD  RATE,
8300                            ENABLED  AND A WRAP CABLE CONNECTED.  THE USER CAN SELECTIVELY
8301                            ENABLE AND DISABLE TESTING OF  THE  OPTIONS  BY  ALTERING  THE
8302                            CONTENTS  OF  '$USER'.  THE DIAGNOSTIC IS DESIGNED TO TEST AND
8303                            DETECT FAULTS TO THE LOGIC LEVEL  (NOT  TO  THE  CHIP  LEVEL).
8304                            THIS TEST OPERATES ON UP TO SIXTEEN(16) IDENTICALLY CONFIGURED
8305                            DLV11-E SERIAL LINE INTERFACES.  THE DEFAULT ADDRESSES ARE:
8306
8307                                    175610 -FIRST SERIAL LINE ADDRESS  OF  16  CONSECUTIVE
8308                                            SERIAL LINE DEVICES.
8309
8310
8311                                    300 - VECTOR FOR FIRST OF 16 DEVICES.
8312
8313                            THIS PROGRAM IS DESIGNED TO RUN  ON  ANY  PDP-11  WITH  4K  OF
8314                            MEMORY AND A DLV11-E (LSI-BUS) MODULE.  IT CAN RUN UNDER XXDP,
8315                            APT, AND ACT MONITORS, AND  ON  PROCESSORS  WITH  NO  HARDWARE
8316                            SWITCH REGISTER.  A POWER FAILURE WILL CAUSE THE DIAG-
8317                            NOSTIC TO RESTART.
8318
8319                    1.2     SYSTEM REQUIREMENTS.
8320
8321                         1.         HARDWARE REQUIREMENTS:
8322
8323                                    ANY PDP-11 FAMILY PROCESSOR
8324                                    4K MEMORY - MINIMUM
8325                                    H315 - CABLE TURN AROUND PLUG (OR EQUIVALENT)
8326                                    MODEM CABLE - BC01V-X OR BC05C-X
8327
8328
8329                                    SOFTWARE REQUIREMENTS:
8330
8331
8332                                    THIS DIAGNOSTIC IS DESIGNED TO  RUN  IN  ANY  OF  THE
8333                            FOLLOWING WAYS:
8334                                    STAND ALONE
8335                                    WITH APT MONITOR
8336                                    WITH ACT MONITOR
8337                                    WITH XXDP MONITOR (CHAINABLE)
8338
```

```
8340
8341                            1.3     RELATED DOCUMENTS AND STANDARDS.
8342
8343                                    DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS        175-003-009-02
8344                                    APT                                                     MD-11-DZZMA
8345                                    ACT                                                     AUTOCAT-11-QZAUB
8346                                    SYSMAC                                                  MD-11-DZQAC
8347
8348
8349                            1.4     DIAGNOSTIC HIERARCHY PREREQUISITES.
8350                                    NO SPECIAL DIAGNOSTICS ARE REQUIRED TO RUN  BEFORE  THIS,  BUT
8351                                    THE  PROCESSOR,  MEMORY,  AND  BUS  ARE  ASSUMMED  TO BE FULLY
8352                                    OPERATIONAL.
8353
8354
8355                            1.5     ASSUMPTIONS.
8356
8357
8358                                    THIS DIAGNOSTIC ASSUMES  THAT  THE  OPERATOR  HAS  INITIALIZED
8359                                    LOCATION '$USWR' AND '$DEVM' TO THE PROPER VALUES.
8360                                    THE (H) JUMPER MUST BE REMOVED FROM ALL DLV11-E'S UNDER TEST.
8361
8362
8363                            2.0     OPERATING INSTRUCTIONS.
8364                                    --------- -------------
8365
8366
8367                            2.1     LOADING AND STARTING PROCEDURES.
8368
8369                                    USE STANDARD PROCEDURE FOR PDP-11  ABSOLUTE  BINARY  FORMATTED
8370                                    MEDIA.
8371
8372                                    THIS DIAGNOSTIC HAS ONLY ONE (1) STARTING  ADDRESS.   200  FOR
8373                                    START AND RESTART.
8374
8375                                    THE USER CAN SELECT A SPECIFIC TEST TO BE EXECUTED BY  SETTING
8376                                    SWITCH 8 IN THE SWITCH REGISTER AND THE TEST NUMBER (IN OCTAL)
8377                                    IN THE LOWER BYTE.  (NOTE:  ALL TESTS PREVIOUS TO THE SELECTED
8378                                    ONE ARE EXECUTED WITHOUT ITERATIONS.)
8379
8380
8381                            2.2     SPECIAL ENVIRONMENTS.
8382
8383                                    THIS DIAGNOSTIC FOLLOWS THE  STANDARD  PROCEDURE  FOR  RUNNING
8384                                    UDER  APT,ACT,XXDP  MONITORS, AS DESCRIBED IN THEIR RESPECTIVE
8385                                    PROCEDURES MANUAL AND SYSMAC PACKAGE.
8386
```

2.3 OPERATIONAL SWITCH SETTINGS

IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER. IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH REGISTER (LOC. 176) IS USED.

CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY DOING THE FOLLOWING:

1) TYPE CONTROL G <^G>; THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.

2) THE MACHINE WILL THEN TYPE:' SWR=XXXXXX NEW=' (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)

3) AFTER THE 'NEW=' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE OF THE FOLLOWING AT THE TTY:

   A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>. (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED). LEADING ZEROS NEED NOT BE TYPED, AND IF MORE THAN 6 DIGITS ARE TYPED THE LAST 6 WILL BE USED. IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.

   B) IF A CONTROL U <^U> IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 3.

   C) IF THE INPUT CHARACTER IS NOT ONE OF THE CHARACTERS MENTIONED ABOVE THEN A QUESTION MARK (?) WILL BE TYPED FOLLOWED BY A CARRAGE RETURN AND A LINE FEED
   THEN PROCEED FROM STEP 3 (ERASING ALL PREVIOUS INPUT.)

   DYNAMIC SWITCH REGISTER
   ------- ------ --------

```
BIT 15 - HALT ON ERROR
    14 - LOOP ON TEST
    13 - INHIBIT ERROR TYPEOUTS
    12 - (UNUSED)
    11 - INHIBIT ITERATIONS
    10 - BELL ON ERROR
     9 - LOOP ON ERROR
     8 - LOOP ON TEST IN SWR<7:0>
   7:0 - TEST NUMBER TO LOOP ON (USED WITH BIT 8)
```

```
8444
8445                                2.4     PROGRAM OPTIONS.
8446
8447                                        THIS PROGRAM WILL SUPPORT TESTING OF MULTIPLE  DLV11-E'S.   IT
8448                                        REQUIRES THE ADDRESS OF THE FIRST RCSR (STORED AT '$BASE') AND
8449                                        ITS INTERRUPT VECTOR (STORED AT '$VECT1');  AND WILL  BE  ABLE
8450                                        TO  ADDRESS ANY DLV11-E STARTING AT THE SPECIFIED BASE ADDRESS
8451                                        UP TO 16 CONSECUTIVE DEVICES.
8452                                        EXAMPLES:       $BASE:  175610
8453                                                        $VECT1:    300
8454
8455                                        THE PROGRAM WILL BE  ABLE  TO  TEST  ANY  DLV11-E  WITHIN  THE
8456                                        ADDRESS RANGE 175610 --> 176000
8457
8458                                        $BASE AND $VECT1 DEFAULT TO 175610 AND 300 RESPECTIVELY.
8459                                        THE PROGRAM ASSOCIATES UNIT NUMBERS AS FOLLOWS:   (NUMBERS  IN
8460                                        PARENTHESIS ARE OCTAL)
8461                                        UNIT#0 -- BASE ADDRESS STORED AT '$BASE'
8462                                                  ASSOCIATED BASE VECTOR STORED AT '$VECT1'
8463                                        UNIT#1 -- BASE ADDRESS + (10)
8464                                                  BASE VECTOR  + (10)
8465                                                  !
8466                                                  !     UP TO
8467                                                  !
8468                                        UNIT#15 -- BASE ADDRESS + (170)
8469                                                   BASE VECTOR  + (170)
8470
8471                                        LOCATION '$DEVM' IS USED AS A BIT MAP TO INDICATE  WHICH  UNIT
8472                                        NUMBERS ARE PRESENT AND WILL BE TESTED.
8473
8474                                        BIT 15                      BIT 1    BIT 0
8475                                        ------------------------------------------
8476                                        !UNIT!                     !UNIT  !   UNIT!
8477                                        ! 15 !                     ! #1   !    #0 !
8478                                        ------------------------------------------
8479
8480                                        A BIT MAP CAN BE ENTERED AT  '$DEVM'  PRIOR  TO  STARTING  THE
8481                                        PROGRAM.
8482
8483                                        EXAMPLE:
8484                                                $BASE:  175610
8485                                                $VECTOR:   300
8486                                                $DEVM:      13
8487
8488                                        THE PROGRAM WILL TEST-
8489
8490                                                UNIT#0  175610  300
8491                                                UNIT#1  175620  310
8492                                                UNIT#3  175640  330
8493
```

OPTIONS

LOCATION $USWR CONTAINS ALL THE USER SELECTABLE OPTIONS. THE VALUES IN THIS WORD MUST CONFORM TO THE ACTUAL BOARD CONFIGURATION.
THE DEFAULT VALUE OF $USWR IS AS FOLLOWS:

BIT POSITION	DEFINITION	DEFAULT VALUE
0-3	#OF DATA BITS	10(8) = 8
4	PARITY ENABLED	0 = NO
5	EVEN ODD PARITY	0 = ODD
6	COMMON SPEED	1 = YES
7	PROGRAMMABLE BAUD RATE	0 = NO
8-11	BAUD RATE OFFSET (SEE FOLLOWING NOTE)	05(8) = 110 BAUD
12	BREAK GENERATION ENABLED	1 = YES
13	CABLE TERMINATED (H315)	1 = YES
14	(-FR) AND (-FD) JUMPERS IN	1 = YES
15	(NOT DEFINED)	

NOTE

THIS DIAGNOSTIC DOES NOT TEST THE PARITY LOGIC.

WHEN THE PROGRAMABLE BAUD RATE OPTION IS ENABLED THE PROGRAMABLE BAUD RATE TEST WILL EXIT WITH THE BAUD RATE SET TO THE SELECTED VALUE. TO CHANGE THE DEFAULT VALUE OF 110 BAUD REPLACE BITS <11:8> WITH THE OFFSET INDICATED IN THE TABLE AT THE END OF THE PBR TEST.

DLV11-E INDIVIDUAL TEST REQUIREMENTS

	TESTS NOT EXECUTED	IF BIT =
APT TEST (BIT 1 OF $ENV)	T25,T37	1
PROGAMMABLE BAUD RATE (BIT 7 OF $USWR)	T32	1
BREAK GENERATION CIRCUIT (BIT 12 OF $USWR)	T2,T40	0
CABLE TERMINATED (BIT 13 OF $USWR)	T15,T16,T17,T20,T21 T22,T23,PARTS OF T34, T36	1
(-FR) & (-FD) JUMPERS IN (BIT 14 OF $USWR)	T5,T6,T16,T17,T20,T22, T23	1

ALL OTHER TESTS WILL RUN REGARDLESS OF ANY BIT SETTINGS.

```
8551
8552
8553                                    2.5     EXECUTION TIMES.
8554
8555                                            EXECUTION TIMES ARE FOR AN LSI-11 PROCESSOR WITH  ALL  OPTIONS
8556                                            ENABLED ON THE DLV11-E (EXECPT FOR PROGRAMMABLE BAUD RATE), AT
8557                                            110 BAUD.
8558
8559                                                    FIRST PASS-         90 SECONDS
8560                                                    ADDITIONAL PASSES   95 SECONDS
8561                                                    ADDITIONAL DEVICES  95 SECONDS
8562
8563                                            THE TEST TIME IS  BAUD  RATE  DEPENDANT;   HIGHER  BAUD  GIVES
8564                                            SHORTER PASS TIMES.
8565
8566
```

3.0 ERROR INFORMATION.

3.1 ERROR REPORTING PROCEDURE.

SINCE THIS DIAGNOSTIC WAS DESIGNED TO FIT IN 4-K OF MEMORY THE ERROR TYPEOUT IS VERY BRIEF. THE FORMAT OF THE ERROR TYPEOUT IS AS FOLLOWS:

TEST#_____,ERROR#_____,PC=_____,ADDRESS=______,VECTOR=_____

WHERE ALL VALUES TYPED ARE OCTAL.
THE ADDRESS AND VECTOR REFER TO THE FAILING DLV11-E.
FOR FURTHER INFORMATION THE LISTING MUST BE CONSULTED.
BITS 15,13,10 AND 9 OF THE SWITCH REGISTER CONTROL THE SEQUENCE OF EVENTS AFTER AN ERROR IS CAUGHT.

BIT 15 - CAUSES THE PROGRAM TO HALT IN THE ERROR ROUTINE. CONTINUEING THE PROGRAM CAUSES IT TO PROCEED.

BIT 13 - DISABLES THE PRINTING OF THE ERROR MESSAGE.

BIT 10 - CAUSES THE BELL TO RING ON ERROR.

BIT 9 - CAUSES THE DIAGNOSTIC TO LOOP FROM BEGINNING OF TEST TO ERROR.

THE ERROR ROUTINE SUPPORTS THE CONTROL G FUNCTION.

3.2 ERROR HALTS.

THE ONLY HALT IN THIS DIAGNOSTIC IS IN THE ERROR ROUTINE, AND IS EXECUTED ONLY IF BIT 15 OF THE SWITCH REGISTER IS A ONE WHEN AN ERROR OCCURS.

4.0 PERFORMANCE AND PROGRESS REPORTS.

4.1 PERFORMANCE REPORTS.

AS EACH DEVICE COMPLETES ONE PASS OF THE DIAGNOSTIC THE FOLLOWING WILL BE TYPED:
CSR:_____,VECTOR:_____,ERRORS:_____

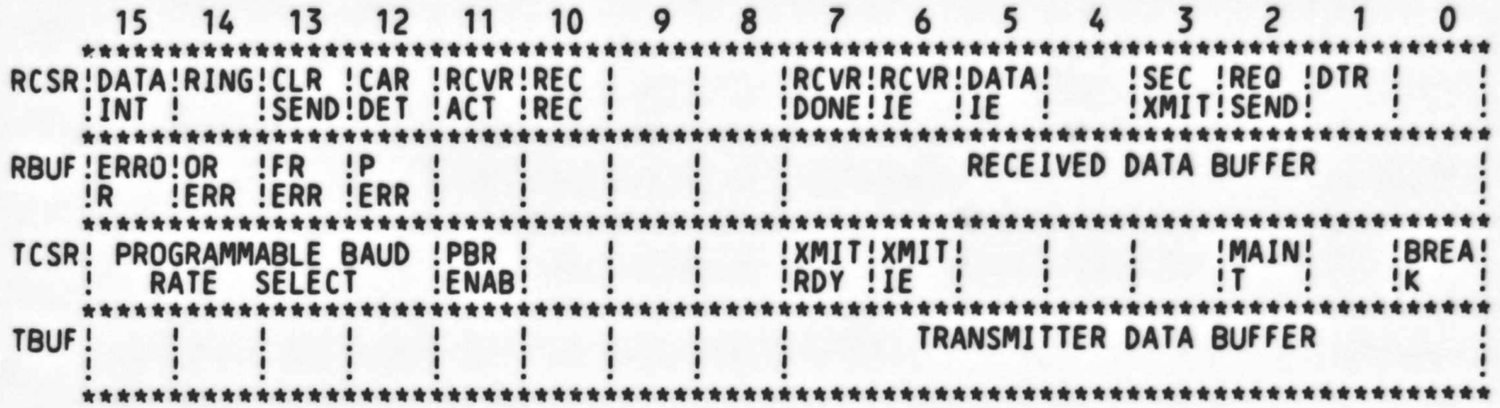
WHERE.'CSR:____' IS THE DEVICE CSR UNDER TEST
'VECTOR:__' IS THE ASSOCIATED VECTOR
AND 'ERRORS:__' IS THE TOTAL NUMBER OF ERRORS ON THIS DEVICE ON THIS PASS.

NOTE

```
                        THIS IS TYPED AFTER THE DEVICE HAS COMPLETED ITS PASS.

                    AFTER ALL DEVICES HAVE BEEN EXERCISED AN END PASS STATEMENT IS
                    TYPED:
                            "ENDPASS#_____."
```

5.0 DEVICE INFORMATION TABLES.



NOTE

BLANK BOXES INDICATE UNUSED AND RESERVED BIT POSITIONS. SEE THE LISTING FOR AN EXPLANATION OF THE BITS.

```
8660
8661                            6.0     SUMMARY OF TESTS AND SPECIAL SUBROUTINES.
8662                                    ------- -- ----- --- ------- ------------
8663
8664
8665                                    TEST 1 ADDRESSABILITY
8666                                    ---- -
8667
8668                                            THIS TEST VERIFIES THAT THE ADDRESS AS PLACED  IN  THE
8669                                            HARDWARE   P-TABLE  TO  BE  CORRECT  AND  THE  DLV11-E
8670                                            RESPONDS TO THAT ADDRESS SPACE.
8671
8672                                    THE FOLLOWING 8 TESTS TEST ALL 'READ WRITE' BITS
8673
8674
8675
8676                                    TEST 2 BREAK - TCSR0 SET, CLEAR, RESET
8677                                    ---- -
8678
8679
8680                                    TEST 3 MAINT - TCSR2 SET, CLEAR, RESET
8681                                    ---- -
8682
8683
8684                                    TEST 4 XMITIE - TCSR6 SET, CLEAR, RESET
8685                                    ---- -
8686
8687                                    TEST 5 DTR - RCSR1 SET, CLEAR
8688                                    ---- -
8689
8690
8691                                                    NOTE
8692
8693                                            RESET DOES NOT CLEAR THIS BIT.  WE CANNOT TEST
8694                                            FOR  AN  INITIAL  CONDITION  AS  THIS  BIT  IS
8695                                            UNDEFINED  UPON  POWER  UP  AND  INIT  DOESN'T
8696                                            AFFECT IT.
8697
8698
8699
8700
8701                                    TEST 6 REQSEND - RCSR2 SET, CLEAR, RESET
8702                                    ---- -
8703
8704                                            THIS TEST ASSUMES THAT JUMPER FR IS IN.
8705
8706
8707                                    TEST 7 SECXMIT - RCSR3 SET, CLEAR, RESET
8708                                    ---- -
8709
8710
8711                                    TEST 10 DATAIE - RCSR5 SET, CLEAR, RESET
8712                                    ---- --
8713
8714
8715                                    TEST 11 RCVRIE - RCSR6 SET, CLEAR, RESET
```

```
8716                                        ---- --
8717
```

```
8719
8720                                    THE FOLLOWING 4 TESTS VERIFY  THAT  RESET  (INIT)  INITIALIZES
8721                                    READ ONLY BITS.
8722
8723
8724                                    TEST 12 RCVRDONE - RCSR 7 - IS CLEARED BY INIT
8725                                    ---- --
8726
8727
8728                                    TEST 13 XMITRDY - TCSR 7 - IS SET BY INIT
8729                                    ---- --
8730
8731
8732                                    TEST 14 DATAINT - RCSR 15 - IS CLEARED BY INIT.
8733                                    ---- --
8734
8735
8736                                    TEST 15 RCVRACT - RCSR 11 - 15 CLEARED BY INIT
8737                                    ---- --
8738
8739                                    THE FOLLOWING 4 TESTS VERIFY  THAT  THE  EIA  SIGNALS  CAN  BE
8740                                    TRANSMITTED AND RECEIVED THROUGH THE CABLE.
8741
8742                                    TEST 16 CARDET SETS AND CLEARS AS DTR SETS AND CLEARS
8743                                    ---- --
8744
8745
8746                                    TEST 17 CLRSEND SETS AND CLEARS AS DTR SETS AND CLEARS
8747                                    ---- --
8748
8749
8750                                    TEST 20 RING SETS AND CLEARS AS REQSEND SETS AND CLEARS
8751                                    ---- --
8752
8753
8754                                    TEST 21 SECREC SETS AND CLEARS AS SECXMIT SETS AND CLEARS
8755                                    ---- --
8756
8757
8758                                    TEST 22 DATAINT (RCSR-15) SETS WHEN DTR CHANGES STATE AND THAT
8759                                    ---- --
8760                                        DATAINT IS CLEARED AFTER READING RCSR
8761
8762
8763                                                            NOTE
8764
8765                                            DTR IS TIED TO BOTH CARDET AND CLRSEND BY  THE
8766                                            H315.
8767
8768
8769                                    TEST 23 DATAINT SETS WHEN RING SETS AND  THAT  DATAINT
8770                                    ---- --
8771                                        DOES NOT SET WHEN RING CLEARS
8772
8773
8774                                    TEST 24 DATAINT SETS WHEN SECREC CHANGES STATE
```

```
                                        ---- --
```

```
8779
8780                    TEST 25 XMIT RDY - TCSR 7 - CLEARS WHEN TBUF IS LOADED
8781                    ---- --
8782                        WITH  A  CHARACTER  AND  THAT IT SETS WITHIN A
8783                        REASONABLE AMOUNT OF TIME.
8784
8785
8786                    TEST 26 OUTPUTTING A CHAR FROM TBUF (WITH  MAINT  SET)
8787                    ---- --
8788                        RESULTS   IN   RCVRDONE   SETTING   WITHIN   A
8789                        REASONABLE  AMOUNT  OF  TIME  AND  THAT  RESET
8790                        CLEARS THE BIT.
8791
8792
8793                    TEST 27 RCVRDONE IS CLEARED BY READING RBUF
8794                    ---- --
8795
8796                    TEST 30 RCVRACT - RCSR 11 - SETS WHEN A START  BIT  IS
8797                    ---- --
8798                        RECEIVED  AND  CLEARS WHEN RCVRDONE - RCSR 7 -
8799                        SETS
8800
8801
8802                    TEST 31 OVERRUN BIT - RBUF 14
8803                    ---- --
8804
8805
8806                    TEST 32 PROGRAMMABLE BAUD RATE TEST TEST AT ALL SPEEDS
8807                    ---- --
8808                        AVAILABLE  A COMPARISON WILL BE MADE TO SEE IF
8809                        NEW TIME IS LESS THAN PREVIOUS.
8810
8811
8812                    TEST 33 TRANSMITTER INTERRUPT LOGIC TEST
8813                    ---- --
8814                        LOGICALLLY THIS IS 4 SEPARATE TESTS
8815                        A) DOES TRANSMITTER INTERRUPT LOGIC WORK
8816                        B) AT PRIORITY OF 0
8817                        C) AND ONLY ONCE
8818                        D) BUT NOT WITH INTERRUPT ENABLE CLEAR
8819
8820
8821                    TEST 34 RECEIVER INTERRUPT LOGIC TEST THIS TEST COVERS ALL
8822                    ---- --
8823                           OF  THE RECEIVER SIDE OF THE INTERRUPT LOGIC, BOTH
8824                           DATASET AND CHARACTER MODES.
8825
8826
8827                    TEST  35  TEST  ACTUAL   DATA   TRANSFERED   NON-INTERRUPT
8828                    ----  --
8829                           MAINTENANCE BIT SET
8830
8831
```

```
8833
8834                    TEST 36 TEST DATA THROUGH CABLE
8835                    ---- --
8836
8837
8838                    TEST 37 FULL DATA TRANSFER WITH INTERRUPTS AND MAINTENANCE
8839                    ---- --
8840                            MODE.
8841
8842
8843                    TEST 40 TEST BREAK GENERATION LOGIC  TRANSMIT  KNOWN  CHAR
8844                    ---- --
8845                            WITH BREAK SET AND COMPARE RECEIVED WITH 0.
8846
8847
8848                    TEST 41 NOT A TEST - SEND BACK TO LOOP
8849                    ---- --
8850
8851                                                NOTE
8852
8853                                FOR ALL OF THE FOLLOWING ROUTINES THE  USE
8854                                OF  (R5)  IS PART OF THE LINKAGE MECHANISM
8855                                BETWEEN THE CALLER AND THE CALLED.
8856
8857
8858                    ROUTINE:TIMER
8859                    -------------
8860
8861                                THIS ROUTINE IS USED TO TEST THE STATUS OF
8862                                ANY BIT IN ANY REGISTER.
8863                    INPUTS:
8864                            HOWLONG     THE  MAXIMUM  AMOUNT  OF  TIME  TO
8865                                        SPEND IN THIS ROUTINE.
8866                            WHICHBIT    A MASK WITH THE  BIT(S)  SET  THAT
8867                                        ARE TO BE CHECKED
8868                            REG         A POINTER TO THE  REGISTER  TO  BE
8869                                        CHECKED
8870                            SETCLR      THE DESIRED RESULTS -- EITHER  SET
8871                                        OR  CLEAR
8872                    OUTPUT:
8873                            THE 'C' BIT IS SET TO INDICATE AN ERROR BUT IT
8874                            IS TESTED BY THE IF.ERROR STATEMENT.
8875
8876
```

```
8878
8879                                    ROUTINE:DATLNG
8880                                    --------------
8881                                            THIS ROUTINE SETS UP A MASK FOR DATA, WITH -
8882
8883                                    INPUT:  NOTHING IS PASSED TO THIS ROUTINE  BUT  GLOBAL
8884                                            INFORMATION IS ASSUMED TO EXIST:
8885                                            $USWR-- THE WORD FOR SOFTWARE PARAMETERS
8886                                            DATA-- A MASK FOR THE LOCATION  OF  THE  OCTAL
8887                                            NUMBER OF DATA BITS
8888
8889                                    OUTPUT----
8890                                            MASK-- A MASK OF BINARY ZEROS  RIGHT-JUSTIFIED
8891                                            THE NUMBER OF WHICH IS DEFINED IN $USWR WORD.
8892
8893
8894                                    ROUTINE:WAIT
8895                                    ------------
8896                                            THIS ROUTINE IS USED TO DELAY EXECUTION OF THE
8897                                            MAIN  PROGRAM  FOR A SPECIFIED AMOUNT OF TIME.
8898                                            THIS  IS  ACCOMPLISHED  BY  INCREMENTING  A
8899                                            REGISTER UP TO A LIMIT.  THE INNER LOOP IS SET
8900                                            TO APPROXIMATE 1 MILLI SEC.
8901
8902
8903
8904                                    SERVICE ROUTINE:  INTSRV
8905                                    ------- --------  ------
8906                                            THIS GLOBAL ROUTINE DOES NOTHING BUT INCREMENT
8907
8908
8909                                            'INTFLAG'  EACH TIME IT IS CALLED.  IT ASSUMES
8910                                            THAT THE MAIN CALLING ROUTINE WILL  KNOW  WHAT
8911                                            TO LOOK FOR.
8912
8913
8914                                    ROUTINE:CYCLE
8915                                    -------------
8916                                            THIS ROUTINE  CAUSES  ADRS  TO  POINT  TO  THE
8917                                            ADDRESS  OF  DLV11-E  UNDER  TEST,  ADRS +2 TO
8918                                            POINT TO THE VECTOR OF THE DLV11-E UNDER TEST.
8919                                            IT  KEEPS  TRACK OF THE CURRENT DEVICE AND BIT
8920                                            MASKS.
```

```
                                @
                                .TITLE  MAINDEC-ZZ-CNDVA-A
                                ;*COPYRIGHT (C) 1982
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY ODES CHOATE
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                                ;*
                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          --------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        13            INHIBIT ERROR TYPEOUTS
                                ;*        11            INHIBIT ITERATIONS
                                ;*        10            BELL ON ERROR
                                ;*         9            LOOP ON ERROR
                                ;*         8            LOOP ON TEST IN SWR<7:0>

                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100
                                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                                ;*MISCELLANEOUS DEFINITIONS
000011                          HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                          LF=     12              ;;CODE FOR LINE FEED
000015                          CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                          CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                          PS=     177776          ;;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                          DDISP=  177570          ;;HARDWARE DISPLAY REGISTER
                                ;*****  THE FOLLOWING ODT START ADDRESS FOR SBC 11/21 IS ADDED
170000                          ODTST=  170000
                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                          R0=     %0              ;;GENERAL REGISTER
000001                          R1=     %1              ;;GENERAL REGISTER
000002                          R2=     %2              ;;GENERAL REGISTER
000003                          R3=     %3              ;;GENERAL REGISTER
000004                          R4=     %4              ;;GENERAL REGISTER
000005                          R5=     %5              ;;GENERAL REGISTER
000006                          R6=     %6              ;;GENERAL REGISTER
000007                          R7=     %7              ;;GENERAL REGISTER
000006                          SP=     %6              ;;STACK POINTER
000007                          PC=     %7              ;;PROGRAM COUNTER
```

```
(1)                         ;*PRIORITY LEVEL DEFINITIONS
(1)             000000      PR0=    0               ;;PRIORITY LEVEL 0
(1)             000040      PR1=    40              ;;PRIORITY LEVEL 1
(1)             000100      PR2=    100             ;;PRIORITY LEVEL 2
(1)             000140      PR3=    140             ;;PRIORITY LEVEL 3
(1)             000200      PR4=    200             ;;PRIORITY LEVEL 4
(1)             000240      PR5=    240             ;;PRIORITY LEVEL 5
(1)             000300      PR6=    300             ;;PRIORITY LEVEL 6
(1)             000340      PR7=    340             ;;PRIORITY LEVEL 7
(1)
(1)                         ;*''SWITCH REGISTER'' SWITCH DEFINITIONS
(1)             100000      SW15=   100000
(1)             040000      SW14=   40000
(1)             020000      SW13=   20000
(1)             010000      SW12=   10000
(1)             004000      SW11=   4000
(1)             002000      SW10=   2000
(1)             001000      SW09=   1000
(1)             000400      SW08=   400
(1)             000200      SW07=   200
(1)             000100      SW06=   100
(1)             000040      SW05=   40
(1)             000020      SW04=   20
(1)             000010      SW03=   10
(1)             000004      SW02=   4
(1)             000002      SW01=   2
(1)             000001      SW00=   1
(1)                         .EQUIV  SW09,SW9
(1)                         .EQUIV  SW08,SW8
(1)                         .EQUIV  SW07,SW7
(1)                         .EQUIV  SW06,SW6
(1)                         .EQUIV  SW05,SW5
(1)                         .EQUIV  SW04,SW4
(1)                         .EQUIV  SW03,SW3
(1)                         .EQUIV  SW02,SW2
(1)                         .EQUIV  SW01,SW1
(1)                         .EQUIV  SW00,SW0
(1)
(1)                         ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
(1)             100000      BIT15=  100000
(1)             040000      BIT14=  40000
(1)             020000      BIT13=  20000
(1)             010000      BIT12=  10000
(1)             004000      BIT11=  4000
(1)             002000      BIT10=  2000
(1)             001000      BIT09=  1000
(1)             000400      BIT08=  400
(1)             000200      BIT07=  200
(1)             000100      BIT06=  100
(1)             000040      BIT05=  40
(1)             000020      BIT04=  20
(1)             000010      BIT03=  10
(1)             000004      BIT02=  4
(1)             000002      BIT01=  2
(1)             000001      BIT00=  1
(1)                         .EQUIV  BIT09,BIT9
```

```
(1)                             .EQUIV  BIT08,BIT8
(1)                             .EQUIV  BIT07,BIT7
(1)                             .EQUIV  BIT06,BIT6
(1)                             .EQUIV  BIT05,BIT5
(1)                             .EQUIV  BIT04,BIT4
(1)                             .EQUIV  BIT03,BIT3
(1)                             .EQUIV  BIT02,BIT2
(1)                             .EQUIV  BIT01,BIT1
(1)                             .EQUIV  BIT00,BIT0
(1)
(1)                             ;*BASIC "CPU" TRAP VECTOR ADDRESSES
(1)     000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
(1)     000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
(1)     000014                  TBITVEC=14              ;;"T" BIT
(1)     000014                  TRTVEC= 14              ;;TRACE TRAP
(1)     000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
(1)     000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
(1)     000024                  PWRVEC= 24              ;;POWER FAIL
(1)     000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
(1)     000034                  TRAPVEC=34              ;;"TRAP" TRAP
(1)     000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
(1)     000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
(1)                             ;*****  THE FOLLOWING BREAK VECTOR AND LINE CLOCK VECTOR ARE INCLUDED
(1)     000100                  LKVEC=  100             ;;LINE CLOCK VECTOR
(1)     000140                  BRKVEC= 140             ;;BREAK VECTOR
(1)     000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
8928    000004                          ILLMEM= 4
8929    000001                          ADRS=   R1
8930    000002                          GOOD=   R2
8931    000003                          BAD=    R3
8932    000001                          REGISTER=R1
8933    000002                          BIT=    R2
8934    000003                          FUNCT=  R3
8935    000002                          LEAD=   R2
8936    000004                          FOLLOW= R4
8937    175610                          DLADDR= 175610
8938
8939                                    ; THE FOLLOWING DEFINITIONS APPLY TO THE GLOBAL SUBS
8940    177777                          SET=    -1
8941    000000                          CLR=    0
8942
8943
8944                            ;;**************************************************************
8945                                    ; RCSR REGISTER BIT NAMES
8946                            ;;**************************************************************
8947    100000                          DATAINT=        BIT15           ; DATASET INTERRUPT
8948    040000                          RING=           BIT14           ; RINGING SIGNAL INDICATOR
8949    020000                          CLRSEND=        BIT13           ; CLEAR TO SEND FROM DATASET
8950    010000                          CARDET=         BIT12           ; CARRIER DETECT
8951    004000                          RCVRACT=        BIT11           ; RECEIVER ACTIVE INDICATOR
8952    002000                          SECREC=         BIT10           ; SECONDARY RECEIVE
8953                                    ; UNUSED        BIT09
8954                                    ; UNUSED        BIT08
8955    000200                          RCVRDONE=       BIT07           ; RECEIVER DONE
8956    000100                          RCVRIE=         BIT06           ; RECEIVER INTERRUPT ENABLE
8957    000040                          DATAIE=         BIT05           ; DATASET INTERRUPT ENABLE
```

```
                                ; UNUSED        BIT04
000010                          SECXMIT=        BIT03           ; SECONDARY TRANSMIT DATA
000004                          REQSEND=        BIT02           ; REQUEST TO SEND
000002                          DTR=            BIT01           ; DATA TERMINAL READY
                                ; UNUSED        BIT00

                        ;;**************************************************************
                                ; RBUF REGISTER BIT NAMES
                        ;;**************************************************************
100000                          ERROR=          BIT15           ; ERROR INDICATOR
040000                          ORERR=          BIT14           ; OVERRUN ERROR
020000                          FRERR=          BIT13           ; FRAMING ERROR
010000                          PERR=           BIT12           ; PARITY ERROR
                                ; UNUSED        BIT11
                                ; UNUSED        BIT10
                                ; UNUSED        BIT09
                                ; UNUSED        BIT08
000200                          RDATA7=         BIT07           ; \
000100                          RDATA6=         BIT06           ;  !
000040                          RDATA5=         BIT05           ;  !
000020                          RDATA4=         BIT04           ;   \   RECEIVED DATA BITS
000010                          RDATA3=         BIT03           ;   /
000004                          RDATA2=         BIT02           ;  !
000002                          RDATA1=         BIT01           ;  !
000001                          RDATA0=         BIT00           ; /

                        ;;**************************************************************
                                ; TCSR REGISTER BIT NAMES
                        ;;**************************************************************
100000                          PBAUD3=         BIT15           ;  \
040000                          PBAUD2=         BIT14           ;   \   PROGRAMMABLE BAUD
020000                          PBAUD1=         BIT13           ;   /   RATE  BITS
010000                          PBAUD0=         BIT12           ;  /
004000                          PBAUDSET=       BIT11           ; ENABLE SETTING OF
                                                                ; PROGRAMMABLE BAUDE RATE
                                ; UNUSED        BIT10
                                ; UNUSED        BIT09
                                ; UNUSED        BIT08
000200                          XMITRDY=        BIT07           ; TRANSMITTER READY
000100                          XMITIE=         BIT06           ; TRANSMITTER INTERRUPT ENABLE
                                ; UNUSED        BIT05
                                ; UNUSED        BIT04
                                ; UNUSED        BIT03
000004                          MAINT=          BIT02           ; MAINTENANCE SET BIT
                                ; UNUSED        BIT01
000001                          BREAK=          BIT00           ; SEND BREAK (CONTINUOUS SPACE)


                        ;;**************************************************************
                                ; TBUF REGISTER BIT NAMES
                        ;;**************************************************************
                                ; UNUSED        BIT15
```

```
                                        ; UNUSED        BIT14
                                        ; UNUSED        BIT13
                                        ; UNUSED        BIT12
                                        ; UNUSED        BIT11
                                        ; UNUSED        BIT10
                                        ; UNUSED        BIT09
                                        ; UNUSED        BIT08
            000200                      TDATA7=         BIT07       ; \
            000100                      TDATA6=         BIT06       ; !
            000040                      TDATA5=         BIT05       ; !
            000020                      TDATA4=         BIT04       ; \ TRANSMITTER DATA BUFFER
            000010                      TDATA3=         BIT03       ; /
            000004                      TDATA2=         BIT02       ; !
            000002                      TDATA1=         BIT01       ; !
            000001                      TDATA0=         BIT00       ; /

                                ;;**************************************************************************
                                        ; FLAG BITS TO BE USE OR CLEARED IN $USWR.

            000017                      DATA =          17
            000020                      PARITY =        20
            000040                      EVENODD =       40
            000100                      COMSPD =        100
            000200                      PBR =           200

                                        ; BAUDE MUST BE ON THE UPPER
                                        ;  BYTE BOUNDRY OF $USWR.--4 BITS
            007400                      BAUD =          7400
            010000                      BRK =           10000
            020000                      CABLE =         20000
            040000                      FRFD =          40000
                                ;;**************************************************************************
                                .SBTTL  TRAP CATCHER
 (1)
 (1)        000000                      .=0
 (1)                            ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
 (1)                            ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
 (1)                            ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
 (1)        000174                      .=174
 (1)  000174  000000            DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
 (1)  000176  000000            SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
 (1)                            .SBTTL  STARTING ADDRESS(ES)
 (1)  000200  000137  001336            JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
                                .SBTTL  ACT11 HOOKS
 (1)
 (2)                            ;;**************************************************************************
 (1)                            ;HOOKS REQUIRED BY ACT11
 (1)        000204                      $SVPC=.                 ;SAVE PC
 (1)        000046                      .=46
 (1)  000046  012446                    $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
 (1)        000052                      .=52
 (1)  000052  000000                    .WORD   0               ;;2)SET LOC.52 TO ZERO
 (1)        000204                      .=$SVPC                 ;; RESTORE PC
                                .=1000
```

```
 9048                                   .SBTTL  APT PARAMETER BLOCK
  (1)
  (2)                                   ;;**********************************************************************
  (1)                                   ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
  (2)                                   ;;**********************************************************************
  (1)           001000                          .$X=.    ;;SAVE CURRENT LOCATION
  (1)           000024                          .=24     ;;SET POWER FAIL TO POINT TO START OF PROGRAM
  (1)  000024   000200                          200      ;;FOR APT START UP
  (1)           000044                          .=44     ;;POINT TO APT INDIRECT ADDRESS PNTR.
  (1)  000044   001000                          $APTHDR  ;;POINT TO APT HEADER BLOCK
  (1)           001000                          .=.$X    ;;RESET LOCATION COUNTER
  (2)                                   ;;**********************************************************************
  (1)                                   ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
  (1)                                   ;INTERFACE SPEC.
  (1)
  (1)  001000                           $APTHD:
  (1)  001000   000000                  $HIBTS: .WORD   0          ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
  (1)  001002   001174                  $MBADR: .WORD   $MAIL      ;;ADDRESS OF APT MAILBOX (BITS 0-15)
  (1)  001004   000005                  $TSTM:  .WORD   5          ;;RUN TIM OF LONGEST TEST
  (1)  001006   000055                  $PASTM: .WORD   45.        ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
  (1)  001010   000036                  $UNITM: .WORD   30.        ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
  (1)  001012   000030                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
 9049                                   .SBTTL  COMMON TAGS
  (1)
  (2)                                   ;;**********************************************************************
  (1)                                   ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
  (1)                                   ;*USED IN THE PROGRAM.
  (1)
  (1)           001100                          .=1100
  (1)  001100                           $CMTAG:                         ;;START OF COMMON TAGS
  (1)  001100  000000                           .WORD   0
  (1)  001102     000                   $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
  (1)  001103     000                   $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
  (1)  001104  000000                   $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
  (1)  001106  000000                   $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
  (1)  001110  000000                   $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
  (1)  001112  000000                   $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
  (1)  001114     000                   $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
  (1)  001115     001                   $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
  (1)  001116  000000                   $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
  (1)  001120  000000                   $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
  (1)  001122  000000                   $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
  (1)  001124  000000                   $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
  (1)  001126  000000                   $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
  (1)  001130  000000                           .WORD   0               ;;RESERVED--NOT TO BE USED
  (1)  001132  000000                           .WORD   0
  (1)  001134     000                   $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
  (1)  001135     000                   $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
  (1)  001136  000000                           .WORD   0
  (1)  001140  177570                   SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
  (1)  001142  177570                   DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
  (1)  001144  177560                   $TKS:   177560                  ;;TTY KBD STATUS
  (1)  001146  177562                   $TKB:   177562                  ;;TTY KBD BUFFER
  (1)  001150  177564                   $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
  (1)  001152  177566                   $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
  (1)  001154     000                   $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
  (1)  001155     002                   $FILLS: .BYTE   2
  (1)                                           ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
  (1)  001156     012                   $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
  (1)  001157     000                   $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
  (1)  001160  000000                   $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
  (1)  001162  000000                   $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
  (1)  001164  177607  000377           $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
  (1)  001170     077                   $QUES:  .ASCII  /?/             ;;QUESTION MARK
  (1)  001171     015                   $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
  (1)  001172  000012                   $LF:    .ASCIZ  <12>            ;;LINE FEED
  (2)                                   ;;**********************************************************************
  (2)                                   .SBTTL  APT MAILBOX-ETABLE
  (2)
  (3)                                   ;;**********************************************************************
  (2)                                   .EVEN
  (2)  001174                           $MAIL:                          ;;APT MAILBOX
  (2)  001174  000000                   $MSGTY: .WORD   AMSGTY           ;;MESSAGE TYPE CODE
  (2)  001176  000000                   $FATAL: .WORD   AFATAL           ;;FATAL ERROR NUMBER
  (2)  001200  000000                   $TESTN: .WORD   ATESTN           ;;TEST NUMBER
  (2)  001202  000000                   $PASS:  .WORD   APASS            ;;PASS COUNT
  (2)  001204  000000                   $DEVCT: .WORD   ADEVCT           ;;DEVICE COUNT
  (2)  001206  000000                   $UNIT:  .WORD   AUNIT            ;;I/O UNIT NUMBER
```

```
(2)  001210  000000                $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
(2)  001212  000000                $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
(2)  001214                        $ETABLE:                ;;APT ENVIRONMENT TABLE
(2)  001214     000                $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
(2)  001215     000                $ENVM:  .BYTE   AENVM
(2)                                ;;ENVIRONMENT MODE BITS
(2)  001216  000000                $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
(2)  001220  071110                $USWR:  .WORD   AUSWR   ;;USER SWITCHES
(2)  001222  000000                $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
(2)                                ;*                      BITS 15-11=CPU TYPE
(2)                                ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
(2)                                ;*                              11/70=06,PDQ=07,Q=10
(2)                                ;*                      BIT 10=REAL TIME CLOCK
(2)                                ;*                      BIT  9=FLOATING POINT PROCESSOR
(2)                                ;*                      BIT  8=MEMORY MANAGEMENT
(2)  001224     000                $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
(2)  001225     000                $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
(2)                                ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
(2)                                ;*                              900 NSEC CORE=001
(2)                                ;*                              300 NSEC BIPOLAR=002
(2)                                ;*                              500 NSEC MOS=003
(2)  001226  000000                $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
(2)                                ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF ''TYPE'' ABOVE
(2)  001230     000                $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
(2)  001231     000                $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
(2)  001232  000000                $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
(2)  001234     000                $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
(2)  001235     000                $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
(2)  001236  000000                $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
(2)  001240     000                $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
(2)  001241     000                $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
(2)  001242  000000                $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
(2)  001244  000300                $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
(2)  001246  000000                $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
(2)  001250  175610                $BASE:  .WORD   ABASE
(2)                                                ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
(2)  001252  000001                $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
(2)  001254                        $ETEND:
(2)                                .MEXIT
```

```
                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT


(1)  001254                     $ERRTB:
9050                            ;;              GLOBAL DATA
9051  001254  175610                  DLADD:  DLADDR
9052  001256  000300                  DLVEC:  300
9053  001260  175610                  RCSR:   DLADDR  +  0
9054  001262  175612                  RBUF:   DLADDR  +  2
9055  001264  175614                  TCSR:   DLADDR  +  4
9056  001266  175615                  TCSRHI: DLADDR  +  5
9057  001270  175616                  TBUF:   DLADDR  +  6
9058  001272  000000                  I:      0
9059  001274  000020                  .BLKW 20        ;FOR R5 STACK
9060  001334  000000                  R5STACK:        .WORD 0
9062  001336                    START:
(1)                             .SBTTL  INITIALIZE THE COMMON TAGS
(1)                             ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
(1)   001336  012706  001100            MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
(1)   001342  005026                    CLR     (R6)+           ;;CLEAR MEMORY LOCATION
(1)   001344  022706  001140            CMP     #SWR,R6 ;;DONE?
(1)   001350  001374                    BNE     .-6             ;;LOOP BACK IF NO
(1)   001352  012706  001100            MOV     #STACK,SP       ;;SETUP THE STACK POINTER
(1)                             ;;INITIALIZE A FEW VECTORS
(1)   001356  012737  014370  000020    MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
(1)   001364  012737  000300  000022    MOV     #PR6,@#IOTVEC+2 ;;LEVEL 6
(1)   001372  012737  014170  000030    MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
(1)   001400  012737  000300  000032    MOV     #PR6,@#EMTVEC+2 ;;LEVEL 6
(1)                             ;;BIT02
(1)   001406  012737  015322  000034    MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
(1)   001414  012737  000300  000036    MOV     #PR6,@#TRAPVEC+2;LEVEL 6
(1)   001422  012737  012502  000024    MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
(1)   001430  012737  000300  000026    MOV     #PR6,@#PWRVEC+2 ;;LEVEL 6
(1)   001436  016767  010752  010742    MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
(1)   001444  005067  177510            CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
(1)   001450  005067  177506            CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
(1)   001454  112767  000001  177433    MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
(1)   001462  012767  001462  177416    MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
(1)   001470  012767  001470  177412    MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
(2)                             ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
(2)                             ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
(2)   001476  013746  000004            MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
(2)   001502  012737  001536  000004    MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
(2)   001510  012767  177570  177422    MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
(2)   001516  012767  177570  177416    MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
```

```
   (2)  001524  022777  177777  177406          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
   (2)  001532  001012                           BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
   (2)                                                                   ;;AND  THE HARDWARE SWR IS NOT = -1
   (2)  001534  000403                           BR      65$             ;;BRANCH IF NO TIMEOUT
   (2)  001536  012716  001544          64$:     MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
   (2)  001542  000002                           RTI
   (2)  001544  012767  000176  177366  65$:     MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
   (2)  001552  012767  000174  177362          MOV     #DISPREG,DISPLAY
   (2)  001560  012637  000004          66$:     MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
   (1)
   (2)  001564  005067  177412                   CLR     $PASS           ;;CLEAR PASS COUNT
   (2)  001570  132767  000200  177417          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
   (2)  001576  001403                           BEQ     67$             ;;YES,USE NON-APT SWITCH
   (2)  001600  012767  001216  177332          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
   (2)  001606                          67$:
9063                                    .SBTTL  TYPE PROGRAM NAME
   (1)                                  ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
   (1)  001606  005227  177777                   INC     #-1             ;;FIRST TIME?
   (1)  001612  001051                           BNE     68$             ;;BRANCH IF NO
   (1)  001614  022737  012446  000042          CMP     #$ENDAD,@#42    ;;ACT-11?
   (1)  001622  001445                           BEQ     68$             ;;BRANCH IF YES
   (1)  001624  104401  001672                   TYPE    ,69$            ;;TYPE ASCIZ STRING
   (2)                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
   (2)  001630  005737  000042                   TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
   (2)  001634  001012                           BNE     70$             ;;BRANCH IF YES
   (2)  001636  126727  177352  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
   (2)  001644  001406                           BEQ     70$             ;;BRANCH IF YES
   (2)  001646  026727  177266  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
   (2)  001654  001005                           BNE     71$             ;;BRANCH IF NO
   (2)  001656  104406                           GTSWR                   ;;GET SOFT-SWR SETTINGS
   (2)  001660  000403                           BR      71$
   (2)  001662  112767  000001  177244  70$:     MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
   (2)  001670                          71$:
   (1)  001670  000422                           BR      68$             ;;GET OVER THE ASCIZ
   (1)                                  ;;69$:    .ASCIZ  <CRLF>*ZZ-CNDVA-A  DLV11-E OFFLINE TEST*<CRLF>
   (1)  001736                          68$:
9068    001736                                  LET INITFLAG := #1
   (4)  001736  012767  000001  010404          MOV     #1,INITFLAG
9069    001744                          LOOP:
9070    001744                                  CALL    CYCLE           ; NO ARGUMENTS--ADDRS -> NEXT ADDRESS
   (3)  001744  004767  010250                   JSR     PC,CYCLE
9071                                            ;                       ADDRS+2 -> NEXT VECTOR
9072                                                                    ;GET UNIT ADDRESS
9073    001750                                                  LET     DLADD := (ADRS)+
   (4)  001750  012167  177300                   MOV     (ADRS)+,DLADD
9074                                                                    ;GET UNIT VECTOR
9075    001754                                                  LET     DLVEC := (ADRS)
   (4)  001754  011167  177276                   MOV     (ADRS),DLVEC
9076    001760                                                  LET     ADRS  := DLADD
   (4)  001760  016701  177270                   MOV     DLADD,ADRS
9077                                                                    ;RCSR = DLADD  +  0
9078    001764                                                  LET     RCSR  := DLADD
   (4)  001764  016767  177264  177266          MOV     DLADD,RCSR
9079    001772                                                  LET     RBUF  := DLADD + #2
   (4)  001772  016767  177256  177262          MOV     DLADD,RBUF
   (7)  002000  062767  000002  177254          ADD     #2,RBUF
```

```
 9080  002006                                                    LET     TCSR  := DLADD + #4
  (4)  002006  016767  177242  177250          MOV     DLADD,TCSR
  (7)  002014  062767  000004  177242          ADD     #4,TCSR
 9081  002022                                                    LET     TCSRHI := DLADD + #5
  (4)  002022  016767  177226  177236          MOV     DLADD,TCSRHI
  (7)  002030  062767  000005  177230          ADD     #5,TCSRHI
 9082  002036                                                    LET     TBUF  := DLADD + #6
  (4)  002036  016767  177212  177224          MOV     DLADD,TBUF
  (7)  002044  062767  000006  177216          ADD     #6,TBUF
 9083  002052                                                    LET     R5    := #R5STACK
  (4)  002052  012705  001334                  MOV     #R5STACK,R5
 9088                                                            ;;BRESET
  (1)  002056  000005                          RESET
```

```
 9099
 9100                                   ;;*********************************************************************
  (3)                                   ;*TEST 1        ADDRESSABILITY
  (4)                                   ;*             THIS TEST VERIFIES THAT THE ADDRESS AS PLACED IN
  (4)                                   ;*             THE HARDWARE P-TABLE TO BE CORRECT AND THE DLV11-E RESPONDS
  (4)                                   ;*             TO THAT ADDRESS SPACE
  (3)                                   ;*********************************************************************
  (2) 002060  000004                    TST1:   SCOPE
  (2)
  (1) 002062  012767  000002  177070            MOV     #2,$TIMES       ;;DO 2 ITERATIONS
  (2) 002070  012767  000001  177102            MOV     #1,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
 9105 002076                                                                    LET     ADRS := DLADD
  (4) 002076  016701  177152                    MOV     DLADD,ADRS
 9106                                                                           ; SET UP INTERRUPT
 9107 002102                                                                    SETVEC  #ILLMEM,#INTSRV,#PR6
  (5) 002102  010146                            MOV     R1,-(SP)
  (5) 002104  012701  000004                    MOV     #ILLMEM,R1
  (5) 002110  012721  012026                    MOV     #INTSRV,(R1)+
  (5) 002114  012711  000300                    MOV     #PR6,(R1)
  (5) 002120  012601                            MOV     (SP)+,R1
 9108 002122                                                                    LET     I := #0
  (4) 002122  005067  177144                    CLR     I
 9109 002126                                                                    REPEAT
  (3) 002126                            $1:
 9110 002126                                                                        BGNSUB
  (5) 002126  012767  002134  176754            MOV     #64$,$LPERR
 9111                                                                               ;CLEAR FLAG
 9112 002134                                                                        LET INTFLAG := #0
  (4) 002134  005067  007674                    CLR     INTFLAG
 9113
 9114                                                                               ;READ FLAG
 9115 002140  005711                            TST @ADRS
 9116 002142                                                                        IF INTFLAG NE #0 THEN
  (6) 002142  005767  007666                    TST     INTFLAG
  (9) 002146  001401                            BEQ     $2
 9117
 9118 002150                                                                        ; FATAL ERROR
  (1) 002150  104001                            ERROR   1                             ERRDF 1,,NODL
 9119 002152                                                                        ENDIF
  (4) 002152                            $2:
 9120 002152                                                                    ENDSUB
 9121 002152                                                                    LET     I := I + #2
  (7) 002152  062767  000002  177112            ADD     #2,I
 9122 002160                                                                    LET     ADRS := DLADD + I
  (4) 002160  016701  177070                    MOV     DLADD,ADRS
  (7) 002164  066701  177102                    ADD     I,ADRS
 9123 002170                                                                UNTIL I EQ #8.
  (3) 002170  026727  177076  000010            CMP     I,#8.
  (6) 002176  001353                            BNE     $1
 9124 002200                                                                CLRVEC ILLMEM
  (3) 002200  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
  (3) 002202  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
  (5) 002204  012701  000004                    MOV     #ILLMEM,R1
  (5) 002210  010102                            MOV     R1,R2
  (8) 002212  062702  000002                    ADD     #2,R2
  (5) 002216  010221                            MOV     R2,(R1)+
```

```
   (5)  002220  005011                    CLR     (R1)
   (3)  002222  012602                    MOV     (SP)+,R2        ;;POP STACK INTO R2
   (3)  002224  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
  9125                                                                    ;END OF TEST
  9126  002226                                              ENDTST
  9131                              ;;*****************************************************************
  9132                              ;* THE FOLLOWING 8 TESTS  TEST ALL 'READ WRITE' BITS
  9133                              ;;*****************************************************************
  9134
  9135
```

```
                                ;;*****************************************************************
                                ;*TEST 2         BREAK - TCSR0 SET, CLEAR, RESET
                                ;*      NOTE:  THE (H) JUMPER MUST BE REMOVED FOR THIS
                                ;*             TEST TO FUNCTION PROPERLY.
                                ;;*****************************************************************
002226  000004                  TST2:   SCOPE
002230  012767  000010  176722          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
002236  012767  000002  176734          MOV     #2,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
002244                                                          IF #BRK NOTSETIN $USWR THEN
002244  032767  010000  176746          BIT     #BRK,$USWR
002252  001004                          BNE     $3
                                                                    EXIT TST
002254  012767  000001  176676          MOV     #1,$TIMES
002262  000452                          BR      TST3            ;;;EXIT THIS TEST
                                                                ENDIF
002264                          $3:
                                                        ; SEE IF IT IS CLEAR
                                                        BGNSUB
002264  012767  002272  176616          MOV     #64$,$LPERR
002272                                                          IF      #BREAK SETIN @TCSR THEN
002272  032777  000001  176764          BIT     #BREAK,@TCSR
002300  001401                          BEQ     $4
                                                                    ; BREAK DID NOT RESET IN TCSR
                                                                    ERRHRD 2,,DIDNOT
002302  104002                          ERROR   2
002304                                                          ENDIF
002304                          $4:
                                                        ENDSUB

                                                ; TRY TO SET BREAK BIT
002304                                          BGNSUB
002304  012767  002312  176576          MOV     #64$,$LPERR
002312                                                          LET     @TCSR := @TCSR SET.BY #BREAK
002312  052777  000001  176744          BIS     #BREAK,@TCSR
                                                                ; STUCK TO 0
002320                                                          IF      #BREAK NOTSETIN @TCSR THEN
002320  032777  000001  176736          BIT     #BREAK,@TCSR
002326  001001                          BNE     $5
                                                                    ; BREAK DID NOT SET IN TCSR
                                                                    ERRHRD 3,,DIDNOT
002330  104003                          ERROR   3
002332                                                          ENDIF
002332                          $5:
                                                        ENDSUB

                                                ; TRY TO CLEAR A SET BIT
002332                                          BGNSUB
002332  012767  002340  176550          MOV     #64$,$LPERR
002340                                                          LET     @TCSR := @TCSR CLR.BY #BREAK
002340  042777  000001  176716          BIC     #BREAK,@TCSR
                                                                ; SHOULD HAVE CLEARED
```

```
 9175  002346                                                          IF      #BREAK SETIN @TCSR THEN
  (6)  002346  032777  000001  176710          BIT     #BREAK,@TCSR
  (9)  002354  001401                          BEQ     $6
 9176                                                                      ; BREAK  DID NOT CLEAR IN TCSR
 9177  002356                                                             ERRHRD 4,,DIDNOT
  (1)  002356  104004                          ERROR   4
 9178  002360                                                          ENDIF
  (4)  002360                          $6:
 9179  002360                                                      ENDSUB
 9180
 9181                                                    ; NOW SEE IF RESET CLEARS IT
 9182  002360                                                      BGNSUB
  (5)  002360  012767  002366  176522          MOV     #64$,$LPERR
 9183
 9184  002366                                                          LET     @TCSR := @TCSR SET.BY #BREAK
  (7)  002366  052777  000001  176670          BIS     #BREAK,@TCSR
 9185                                                                      ; ISSUE BUS RESET
 9186  002374                                                          BRESET
  (1)  002374  000005                          RESET
 9187  002376                                                          IF      #BREAK SETIN @TCSR THEN
  (6)  002376  032777  000001  176660          BIT     #BREAK,@TCSR
  (9)  002404  001401                          BEQ     $7
 9188                                                                      ; BREAK DID NOT RESET IN TCSR
 9189  002406                                                             ERRHRD 5,,DIDNOT
  (1)  002406  104005                          ERROR   5
 9190  002410                                                          ENDIF
  (4)  002410                          $7:
 9191  002410                                                      ENDSUB
 9192  002410                                                  ENDTST
 9193
 9194
 9199                    ;;*****************************************************************
```

```
 9201
 9202                                   ;;*****************************************************************
  (3)                                   ;*TEST 3          MAINT - TCSR2 SET, CLEAR, RESET
  (3)                                   ;;*****************************************************************
  (2) 002410  000004                    TST3:   SCOPE
  (2)
  (1) 002412  012767  000010  176540            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 002420  012767  000003  176552            MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
 9207
 9208                                                   ; SEE IF IT IS CLEAR
 9209 002426                                                               BGNSUB
  (5) 002426  012767  002434  176454            MOV     #64$,$LPERR
 9210
 9211 002434                                                               IF      #MAINT SETIN @TCSR THEN
  (6) 002434  032777  000004  176622            BIT     #MAINT,@TCSR
  (9) 002442  001401                            BEQ     $10
 9212
 9213 002444                                                                   ; MAINT DID NOT RESET IN TCSR
  (1) 002444  104006                            ERROR   6                      ERRHRD 6,,DIDNOT
 9214 002446                                                               ENDIF
  (4) 002446                            $10:
 9215 002446                                                           ENDSUB
 9216
 9217                                                   ; TRY TO SET MAINT BIT
 9218 002446                                                               BGNSUB
  (5) 002446  012767  002454  176434            MOV     #64$,$LPERR
 9219 002454                                                               LET     @TCSR := @TCSR SET.BY #MAINT
  (7) 002454  052777  000004  176602            BIS     #MAINT,@TCSR
 9220                                                                          ; STUCK TO 0
 9221 002462                                                               IF      #MAINT NOTSETIN @TCSR THEN
  (6) 002462  032777  000004  176574            BIT     #MAINT,@TCSR
  (9) 002470  001001                            BNE     $11
 9222                                                                          ; MAINT DID NOT SET IN TCSR
 9223 002472                                                                   ERRHRD 7,,DIDNOT
  (1) 002472  104007                            ERROR   7
 9224 002474                                                               ENDIF
  (4) 002474                            $11:
 9225 002474                                                           ENDSUB
 9226
 9227                                                   ; TRY TO CLEAR A SET BIT
 9228 002474                                                               BGNSUB
  (5) 002474  012767  002502  176406            MOV     #64$,$LPERR
 9229
 9230 002502                                                               LET     @TCSR := @TCSR CLR.BY #MAINT
  (7) 002502  042777  000004  176554            BIC     #MAINT,@TCSR
 9231                                                                          ; SHOULD HAVE CLEARED
 9232 002510                                                               IF      #MAINT SETIN @TCSR THEN
  (6) 002510  032777  000004  176546            BIT     #MAINT,@TCSR
  (9) 002516  001401                            BEQ     $12
 9233                                                                          ; MAINT DID NOT CLEAR INTCSR
 9234 002520                                                                   ERRHRD 10,,DIDNOT
  (1) 002520  104010                            ERROR   10
 9235 002522                                                               ENDIF
  (4) 002522                            $12:
 9236 002522                                                           ENDSUB
 9237
```

```
                                                ; NOW SEE IF RESET CLEARS IT
                                                        BGNSUB
002522
002522  012767  002530  176360          MOV     #64$,$LPERR
002530
                                                        LET     @TCSR := @TCSR SET.BY #MAINT
002530  052777  000004  176526          BIS     #MAINT,@TCSR
                                                        ; ISSUE BUS RESET
002536                                                  BRESET
002536  000005                          RESET
                                                        IF      #MAINT SETIN @TCSR THEN
002540  032777  000004  176516          BIT     #MAINT,@TCSR
002546  001401                          BEQ     $13
                                                        ; MAINT DID NOT RESET IN TCSR
                                                        ERRHRD 11,,DIDNOT
002550  104011                          ERROR   11
                                                        ENDIF
002552                          $13:
                                                    ENDSUB
                                                ENDTST
002552
002552
002552
002552
                                ;;**************************************************************************
```

```
                                ;;*****************************************************************
                                ;*TEST 4          XMITIE - TCSR6 SET, CLEAR, RESET
                                ;;*****************************************************************
002552  000004                  TST4:   SCOPE
                                
002554  012767  000010  176376          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
002562  012767  000004  176410          MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
                                                                ; USE PRIORITY OF 6
002570  012746  000300                  MOV     #PR6,-(SP)      ;;PUT NEW PS ON STACK
002574  012746  002602                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
002600  000002                          RTI                     ;;POP NEW PC AND PS
002602                          64$:
                                
                                                ; SEE IF IT IS CLEAR
                                                                BGNSUB
002602
002602  012767  002610  176300          MOV     #65$,$LPERR
                                
                                                                IF      #XMITIE SETIN @TCSR THEN
002610
002610  032777  000100  176446          BIT     #XMITIE,@TCSR
002616  001401                          BEQ     $14
                                
                                                                ; XMITIE DID NOT RESET IN TCSR
002620                                                          ERRHRD 12,,DIDNOT
002620  104012                          ERROR   12
                                                                ENDIF
002622
002622                          $14:
                                                                ENDSUB
                                
                                                ; TRY TO SET XMITIE BIT
                                                                BGNSUB
002622
002622  012767  002630  176260          MOV     #64$,$LPERR
                                                                LET     @TCSR := @TCSR SET.BY #XMITIE
002630
002630  052777  000100  176426          BIS     #XMITIE,@TCSR
                                                                ; STUCK TO 0
                                                                IF      #XMITIE NOTSETIN @TCSR THEN
002636
002636  032777  000100  176420          BIT     #XMITIE,@TCSR
002644  001001                          BNE     $15
                                
                                                                ; XMIT DID NOT RESET IN TCSR
002646                                                          ERRHRD 13,,DIDNOT
002646  104013                          ERROR   13
                                                                ENDIF
002650
002650                          $15:
                                                                ENDSUB
                                
                                                ; TRY TO CLEAR A SET BIT
                                                                BGNSUB
002650
002650  012767  002656  176232          MOV     #64$,$LPERR
                                
                                                                LET     @TCSR := @TCSR CLR.BY #XMITIE
002656
002656  042777  000100  176400          BIC     #XMITIE,@TCSR
                                                                ; SHOULD HAVE CLEARED
                                                                IF      #XMITIE SETIN @TCSR THEN
002664
002664  032777  000100  176372          BIT     #XMITIE,@TCSR
002672  001401                          BEQ     $16
                                
                                                                ;  XMIT DID NOT CLEAR IN TCSR
002674                                                          ERRHRD 14,,DIDNOT
```

```
  (1)  002674  104014                    ERROR   14
                                                                    ENDIF
 9303  002676
  (4)  002676                    $16:
                                                                  ENDSUB
 9304  002676
 9305
 9306                                                ; NOW SEE IF RESET CLEARS IT
 9307  002676                                                     BGNSUB
  (5)  002676  012767  002704  176204    MOV     #64$,$LPERR
 9308
 9309  002704                                                     LET    @TCSR := @TCSR SET.BY #XMITIE
  (7)  002704  052777  000100  176352    BIS     #XMITIE,@TCSR
 9310                                                            ; ISSUE BUS RESET
 9311  002712                                                     BRESET
  (1)  002712  000005                    RESET
 9312  002714                                                     IF     #XMITIE SETIN @TCSR THEN
  (6)  002714  032777  000100  176342    BIT     #XMITIE,@TCSR
  (9)  002722  001401                    BEQ     $17
 9313
 9314  002724                                                         ; XMIT DID NOT RESET IN TCSR
                                                                     ERRHRD 15,,DIDNOT
  (1)  002724  104015                    ERROR   15
 9315  002726                                                     ENDIF
  (4)  002726                    $17:
                                                                  ENDSUB
 9316  002726
 9317  002726                                                ENDTST
 9318
 9319
 9320
 9325                            ;;*****************************************************************
```

```
 9334
 9335                                   ;;*********************************************************************
  (3)                                   ;*TEST 5          DTR - RCSR1    SET, CLEAR
  (4)                                   ;*               NOTE:  RESET DOES NOT CLEAR THIS BIT
  (4)                                   ;*               WE CANNOT TEST FOR AN INITIAL CONDITION
  (4)                                   ;*               AS THIS BIT IS UNDEFINED UPON POWER UP AND
  (4)                                   ;*               INIT DOESN'T AFFECT IT.
  (4)                                   ;*              THE (-FD) JUMPER MUST BE IN FOR THIS TEST TO WORK.
  (3)                                   ;*********************************************************************
  (2) 002726  000004                    TST5:   SCOPE
  (2)
  (1) 002730  012767  000010  176222            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 002736  012767  000005  176234            MOV     #5,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
 9340 002744                                                                    IF #FRFD NOTSETIN $USWR THEN
  (6) 002744  032767  040000  176246            BIT     #FRFD,$USWR
  (9) 002752  001004                            BNE     $20
 9341 002754                                                                        EXIT TST
  (5) 002754  012767  000001  176176            MOV     #1,$TIMES
  (3) 002762  000441                            BR      TST6            ;;;EXIT THIS TEST
 9342 002764                                                                    ENDIF
  (4) 002764                            $20:
 9343                                                           ; TRY TO CLEAR DTR BIT
 9344 002764                                                                    BGNSUB
  (5) 002764  012767  002772  176116            MOV     #64$,$LPERR
 9345 002772                                                                        LET    @RCSR := @RCSR CLR.BY #DTR
  (7) 002772  042777  000002  176260            BIC     #DTR,@RCSR
 9346                                                                                   ; STUCK TO 0
 9347 003000                                                                        IF     #DTR SETIN @RCSR THEN
  (6) 003000  032777  000002  176252            BIT     #DTR,@RCSR
  (9) 003006  001401                            BEQ     $21
 9348                                                                                   ; DTR DID NOT CLEAR IN RCSR
 9349 003010                                                                        ERRHRD 16,,DIDNOT
  (1) 003010  104016                            ERROR   16
 9350 003012                                                                    ENDIF
  (4) 003012                            $21:
 9351 003012                                                                ENDSUB
 9352
 9353                                                           ; TRY TO SET  DTR
 9354 003012                                                                BGNSUB
  (5) 003012  012767  003020  176070            MOV     #64$,$LPERR
 9355
 9356 003020                                                                    LET    @RCSR := @RCSR SET.BY #DTR
  (7) 003020  052777  000002  176232            BIS     #DTR,@RCSR
 9357 003026                                                                    IF     #DTR NOTSETIN @RCSR THEN
  (6) 003026  032777  000002  176224            BIT     #DTR,@RCSR
  (9) 003034  001001                            BNE     $22
 9358                                                                               ; DTR DID NOT SET IN RCSR
 9359 003036                                                                    ERRHRD 17,,DIDNOT
  (1) 003036  104017                            ERROR   17
 9360 003040                                                                ENDIF
  (4) 003040                            $22:
 9361 003040                                                            ENDSUB
 9362
 9363                                                           ; TRY TO CLEAR IT AGAIN
 9364 003040                                                            BGNSUB
  (5) 003040  012767  003046  176042            MOV     #64$,$LPERR
```

```
 9365  003046
  (7)  003046  042777  000002  176204          BIC     #DTR,@RCSR
 9366
 9367  003054
  (6)  003054  032777  000002  176176          BIT     #DTR,@RCSR
  (9)  003062  001401                          BEQ     $23
 9368
 9369  003064
  (1)  003064  104020                          ERROR   20
 9370  003066
  (4)  003066                          $23:
 9371  003066
 9372  003066
 9373
 9374
 9375
 9380                          ;;*****************************************************************

                                                LET     @RCSR := @RCSR CLR.BY #DTR

                                                    ; SHOULD HAVE CLEARED IT
                                                IF      #DTR SETIN @RCSR THEN


                                                    ; DTR DID NOT CLEAR IN RCSR
                                                    ERRHRD 20,,DIDNOT

                                                ENDIF

                                             ENDSUB
                                          ENDTST
```

```
 9385
 9386                                   ;;*****************************************************************
  (3)                                   ;*TEST 6          REQSEND - RCSR2          SET, CLEAR, RESET
  (4)                                   ;*               THIS TEST ASSUMES THAT JUMPER -(FR) IS IN
  (3)                                   ;;*****************************************************************
  (2)  003066  000004                   TST6:   SCOPE
  (2)
  (1)  003070  012767  000010  176062           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2)  003076  012767  000006  176074           MOV     #6,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
 9391  003104                                                           IF #FRFD NOTSETIN $USWR THEN
  (6)  003104  032767  040000  176106           BIT     #FRFD,$USWR
  (9)  003112  001004                           BNE     $24
 9392  003114                                                               EXIT TST
  (5)  003114  012767  000001  176036           MOV     #1,$TIMES
  (3)  003122  000452                           BR      TST7            ;;;EXIT THIS TEST
 9393  003124                                                           ENDIF
  (4)  003124                           $24:
 9394
 9395                                                   ; SEE IF IT IS CLEAR
 9396  003124                                                           BGNSUB
  (5)  003124  012767  003132  175756           MOV     #64$,$LPERR
 9397
 9398  003132                                                               IF      #REQSEND SETIN @RCSR THEN
  (6)  003132  032777  000004  176120           BIT     #REQSEND,@RCSR
  (9)  003140  001401                           BEQ     $25
 9399                                                                          ; REQSEND DID NOT RESET IN RCSR
 9400  003142                                                                  ERRHRD 21,,DIDNOT
  (1)  003142  104021                           ERROR   21
 9401  003144                                                               ENDIF
  (4)  003144                           $25:
 9402  003144                                                           ENDSUB
 9403
 9404                                                   ; TRY TO SET REQSEND BIT
 9405  003144                                                           BGNSUB
  (5)  003144  C12767  003152  175736           MOV     #64$,$LPERR
 9406  003152                                                               LET     @RCSR := @RCSR SET.BY #REQSEND
  (7)  003152  052777  000004  176100           BIS     #REQSEND,@RCSR
 9407                                                                          ; STUCK TO 0
 9408  003160                                                                IF      #REQSEND NOTSETIN @RCSR THEN
  (6)  003160  032777  000004  176072           BIT     #REQSEND,@RCSR
  (9)  003166  001001                           BNE     $26
 9409                                                                          ; REQSEND DID NOT SET IN RCSR
 9410  003170                                                                  ERRHRD 22,,DIDNOT
  (1)  003170  104022                           ERROR   22
 9411  003172                                                               ENDIF
  (4)  003172                           $26:
 9412  003172                                                           ENDSUB
 9413
 9414                                                   ; TRY TO CLEAR A SET BIT
 9415  003172                                                           BGNSUB
  (5)  003172  012767  003200  175710           MOV     #64$,$LPERR
 9416
 9417  003200                                                               LET     @RCSR := @RCSR CLR.BY #REQSEND
  (7)  003200  042777  000004  176052           BIC     #REQSEND,@RCSR
 9418                                                                          ; SHOULD HAVE CLEARED
 9419  003206                                                                IF      #REQSEND SETIN @RCSR THEN
```

```
  (6)  003206  032777  000004  176044          BIT     #REQSEND,@RCSR
  (9)  003214  001401                          BEQ     $27
9420                                                        ; REQSEND DID NOT CLEAR IN RCSR
9421  003216                                                ERRHRD 23,,DIDNOT
  (1)  003216  104023                          ERROR   23
9422  003220                                                ENDIF
  (4)  003220                              $27:
9423  003220                                                ENDSUB
9424
9425                                                  ; NOW SEE IF RESET CLEARS IT
9426  003220                                                BGNSUB
  (5)  003220  012767  003226  175662          MOV     #64$,$LPERR
9427
9428  003226                                                LET     @RCSR := @RCSR SET.BY #REQSEND
  (7)  003226  052777  000004  176024          BIS     #REQSEND,@RCSR
9429                                                        ; ISSUE BUS RESET
9430  003234                                                BRESET
  (1)  003234  000005                          RESET
9431  003236                                                IF      #REQSEND SETIN @RCSR THEN
  (6)  003236  032777  000004  176014          BIT     #REQSEND,@RCSR
  (9)  003244  001401                          BEQ     $30
9432                                                        ; REQSEND DID NOT RESET IN RCSR
9433  003246                                                ERRHRD 24,,DIDNOT
  (1)  003246  104024                          ERROR   24
9434  003250                                                ENDIF
  (4)  003250                              $30:
9435  003250                                                ENDSUB
9436  003250                                              ENDTST
9437
9438
9439
9444                                    ;;*****************************************************************
```

```
                                ;;*****************************************************************
                                ;*TEST 7          SECXMIT - RCSR3         SET, CLEAR, RESET
                                ;;*****************************************************************
003250  000004          TST7:   SCOPE
003252  012767  000010  175700          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003260  012767  000007  175712          MOV     #7,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
                                                        ; SEE IF IT IS CLEAR
003266                                                          BGNSUB
003266  012767  003274  175614          MOV     #64$,$LPERR
003274                                                              IF      #SECXMIT SETIN @RCSR THEN
003274  032777  000010  175756          BIT     #SECXMIT,@RCSR
003302  001401                          BEQ     $31
                                                                    ; SECXMIT DID NOT RESET IN RCSR
003304                                                              ERRHRD 25,,DIDNOT
003304  104025                          ERROR   25
003306                                                              ENDIF
003306                          $31:
003306                                                          ENDSUB
                                                        ; TRY TO SET SECXMIT BIT
003306                                                          BGNSUB
003306  012767  003314  175574          MOV     #64$,$LPERR
003314                                                              LET     @RCSR := @RCSR SET.BY #SECXMIT
003314  052777  000010  175736          BIS     #SECXMIT,@RCSR
                                                                    ; STUCK TO 0
003322                                                              IF      #SECXMIT NOTSETIN @RCSR THEN
003322  032777  000010  175730          BIT     #SECXMIT,@RCSR
003330  001001                          BNE     $32
                                                                    ; SECXMIT DID NOT SET IN RCSR
003332                                                              ERRHRD 26,,DIDNOT
003332  104026                          ERROR   26
003334                                                              ENDIF
003334                          $32:
003334                                                          ENDSUB
                                                        ; TRY TO CLEAR A SET BIT
003334                                                          BGNSUB
003334  012767  003342  175546          MOV     #64$,$LPERR
003342                                                              LET     @RCSR := @RCSR CLR.BY #SECXMIT
003342  042777  000010  175710          BIC     #SECXMIT,@RCSR
                                                                    ; SHOULD HAVE CLEARED
003350                                                              IF      #SECXMIT SETIN @RCSR THEN
003350  032777  000010  175702          BIT     #SECXMIT,@RCSR
003356  001401                          BEQ     $33
                                                                    ; SECXMIT DID NOT CLEAR IN RCSR
003360                                                              ERRHRD 27,,DIDNOT
003360  104027                          ERROR   27
003362                                                              ENDIF
003362                          $33:
003362                                                          ENDSUB
003362                                                          BGNSUB
```

```
  (5)  003362  012767  003370  175520          MOV     #64$,$LPERR
9483                                                    ; NOW SEE IF RESET CLEARS IT
9484
9485  003370                                                            LET     @RCSR := @RCSR SET.BY #SECXMIT
  (7)  003370  052777  000010  175662          BIS     #SECXMIT,@RCSR
9486                                                                    ; ISSUE BUS RESET
9487  003376                                                            BRESET
  (1)  003376  000005                          RESET
9488  003400                                                            IF      #SECXMIT SETIN @RCSR THEN
  (6)  003400  032777  000010  175652          BIT     #SECXMIT,@RCSR
  (9)  003406  001401                          BEQ     $34
9489                                                                        ; SECXMIT DID NOT RESET IN RCSR
9490  003410                                                                ERRHRD 30,,DIDNOT
  (1)  003410  104030                          ERROR   30
9491  003412                                                            ENDIF
  (4)  003412                          $34:
9492  003412                                                        ENDSUB
9493  003412                                                    ENDTST
9494
9495
9496
9501                                 ;;*****************************************************************
```

```
 9503
 9504                                   ;;*****************************************************************
  (3)                                   ;*TEST 10        DATAIE - RCSR5 SET, CLEAR, RESET
  (3)                                   ;;*****************************************************************
  (2)  003412  000004                   TST10:  SCOPE
  (2)
  (1)  003414  012767  000010  175536           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2)  003422  012767  000010  175550           MOV     #10,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 9509                                                   ; SEE IF IT IS CLEAR
 9510  003430                                                            BGNSUB
  (5)  003430  012767  003436  175452           MOV     #64$,$LPERR
 9511
 9512  003436                                                                  IF      #DATAIE SETIN @RCSR THEN
  (6)  003436  032777  000040  175614           BIT     #DATAIE,@RCSR
  (9)  003444  001401                           BEQ     $35
 9513                                                                        ; DATAIE DID NOT RESET IN RCSR
 9514  003446                                                                 ERRHRD 31,,DIDNOT
  (1)  003446  104031                           ERROR   31
 9515  003450                                                                 ENDIF
  (4)  003450                           $35:
 9516  003450                                                            ENDSUB
 9517
 9518                                                   ; TRY TO SET DATAIE BIT
 9519  003450                                                            BGNSUB
  (5)  003450  012767  003456  175432           MOV     #64$,$LPERR
 9520  003456                                                                  LET     @RCSR := @RCSR SET.BY #DATAIE
  (7)  003456  052777  000040  175574           BIS     #DATAIE,@RCSR
 9521                                                                       ; STUCK TO 0
 9522  003464                                                                 IF      #DATAIE NOTSETIN @RCSR THEN
  (6)  003464  032777  000040  175566           BIT     #DATAIE,@RCSR
  (9)  003472  001001                           BNE     $36
 9523                                                                        ; DATAIE DID NOT SET IN RCSR
 9524  003474                                                                 ERRHRD 32,,DIDNOT
  (1)  003474  104032                           ERROR   32
 9525  003476                                                                 ENDIF
  (4)  003476                           $36:
 9526  003476                                                            ENDSUB
 9527
 9528                                                   ; TRY TO CLEAR A SET BIT
 9529  003476                                                            BGNSUB
  (5)  003476  012767  003504  175404           MOV     #64$,$LPERR
 9530
 9531  003504                                                                  LET     @RCSR := @RCSR CLR.BY #DATAIE
  (7)  003504  042777  000040  175546           BIC     #DATAIE,@RCSR
 9532                                                                       ; SHOULD HAVE CLEARED
 9533  003512                                                                 IF      #DATAIE SETIN @RCSR THEN
  (6)  003512  032777  000040  175540           BIT     #DATAIE,@RCSR
  (9)  003520  001401                           BEQ     $37
 9534                                                                        ; DATAIE DID NOT CLEAR IN RCSR
 9535  003522                                                                 ERRHRD 33,,DIDNOT
  (1)  003522  104033                           ERROR   33
 9536  003524                                                                 ENDIF
  (4)  003524                           $37:
 9537  003524                                                            ENDSUB
 9538
 9539                                                   ; NOW SEE IF RESET CLEARS IT
```

```
 9540  003524                                                          BGNSUB
  (5)  003524  012767  003532  175356          MOV     #64$,$LPERR
 9541
 9542  003532                                                              LET     @RCSR := @RCSR SET.BY #DATAIE
  (7)  003532  052777  000040  175520          BIS     #DATAIE,@RCSR
 9543                                                                      ; ISSUE BUS RESET
 9544  003540                                                              BRESET
  (1)  003540  000005                          RESET
 9545  003542                                                              IF      #DATAIE SETIN @RCSR THEN
  (6)  003542  032777  000040  175510          BIT     #DATAIE,@RCSR
  (9)  003550  001401                          BEQ     $40
 9546                                                                        ; DATAIE DID NOT RESET IN RCSR
 9547  003552                                                                ERRHRD 34,,DIDNOT
  (1)  003552  104034                          ERROR   34
 9548  003554                                                              ENDIF
  (4)  003554                          $40:
 9549  003554                                                          ENDSUB
 9550  003554                                                      ENDTST
 9551
 9552
 9553
 9558                          ;;*****************************************************************
```

```
 9560
 9561                                   ;;*****************************************************************
  (3)                                   ;*TEST 11        RCVRIE - RCSR6 SET, CLEAR, RESET
  (3)                                   ;;*****************************************************************
  (2) 003554  000004                    TST11:  SCOPE
  (2)
  (1) 003556  012767  000010  175374            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 003564  012767  000011  175406            MOV     #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 9566                                                   ; SEE IF IT IS CLEAR
 9567 003572                                                             BGNSUB
  (5) 003572  012767  003600  175310            MOV     #64$,$LPERR
 9568
 9569 003600                                                               IF      #RCVRIE SETIN @RCSR THEN
  (6) 003600  032777  000100  175452            BIT     #RCVRIE,@RCSR
  (9) 003606  001401                            BEQ     $41
 9570                                                                          ; RCVRIE DID NOT RESET IN RCSR
 9571 003610                                                                   ERRHRD 35,,DIDNOT
  (1) 003610  104035                            ERROR   35
 9572 003612                                                               ENDIF
  (4) 003612                            $41:
 9573 003612                                                             ENDSUB
 9574
 9575                                                   ; TRY TO SET RCVRIE BIT
 9576 003612                                                             BGNSUB
  (5) 003612  012767  003620  175270            MOV     #64$,$LPERR
 9577 003620                                                               LET     @RCSR := @RCSR SET.BY #RCVRIE
  (7) 003620  052777  000100  175432            BIS     #RCVRIE,@RCSR
 9578                                                                      ; STUCK TO 0
 9579 003626                                                               IF      #RCVRIE NOTSETIN @RCSR THEN
  (6) 003626  032777  000100  175424            BIT     #RCVRIE,@RCSR
  (9) 003634  001001                            BNE     $42
 9580                                                                          ; RCVRIE DID NOT SET IN RCSR
 9581 003636                                                                   ERRHRD 36,,DIDNOT
  (1) 003636  104036                            ERROR   36
 9582 003640                                                               ENDIF
  (4) 003640                            $42:
 9583 003640                                                             ENDSUB
 9584
 9585                                                   ; TRY TO CLEAR A SET BIT
 9586 003640                                                             BGNSUB
  (5) 003640  012767  003646  175242            MOV     #64$,$LPERR
 9587
 9588 003646                                                               LET     @RCSR := @RCSR CLR.BY #RCVRIE
  (7) 003646  042777  000100  175404            BIC     #RCVRIE,@RCSR
 9589                                                                      ; SHOULD HAVE CLEARED
 9590 003654                                                               IF      #RCVRIE SETIN @RCSR THEN
  (6) 003654  032777  000100  175376            BIT     #RCVRIE,@RCSR
  (9) 003662  001401                            BEQ     $43
 9591                                                                          ; RCVRIE DID NOT CLEAR IN RCSR
 9592 003664                                                                   ERRHRD 37,,DIDNOT
  (1) 003664  104037                            ERROR   37
 9593 003666                                                               ENDIF
  (4) 003666                            $43:
 9594 003666                                                             ENDSUB
 9595
 9596                                                   ; NOW SEE IF RESET CLEARS IT
```

```
 9597  003666                                                          BGNSUB
  (5)  003666  012767  003674  175214          MOV     #64$,$LPERR
 9598
 9599  003674                                                              LET     @RCSR := @RCSR SET.BY #RCVRIE
  (7)  003674  052777  000100  175356          BIS     #RCVRIE,@RCSR
 9600                                                                      ; ISSUE BUS RESET
 9601  003702  .                                                           BRESET
  (1)  003702  000005                          RESET
 9602  003704                                                              IF      #RCVRIE SETIN @RCSR THEN
  (6)  003704  032777  000100  175346          BIT     #RCVRIE,@RCSR
  (9)  003712  001401                          BEQ     $44
 9603                                                                        ; RCVRIE DID NOT RESET IN RCSR
 9604  003714                                                                ERRHRD 40,,DIDNOT
  (1)  003714  104040                          ERROR   40
 9605  003716                                                              ENDIF
  (4)  003716                          $44:
 9606  003716                                                              CKLOOP
 9607  003716                                                          ENDSUB
 9608  003716                                                      ENDTST
 9609
 9610
 9611
 9612
 9617                          ;;*****************************************************************
 9618                          ;* THE FOLLOWING 4 TESTS VERIFY
 9619                          ;* THAT RESET (INIT) INITIALIZES READ ONLY BITS.
 9620                          ;;*****************************************************************
 9621
 9622
```

```
 9624
 9625                                ;;*****************************************************************
  (3)                                ;*TEST 12        TEST THAT RCVRDONE - RCSR 7 - IS CLEARED BY INIT
  (3)                                ;;*****************************************************************
  (2)  003716  000004                TST12:  SCOPE
  (2)
  (1)  003720  012767  000010  175232        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2)  003726  012767  000012  175244        MOV     #12,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 9626
 9631
 9632
 9633
 9634
 9635  003734                                                              BGNSUB
  (5)  003734  012767  003742  175146        MOV     #64$,$LPERR
 9636  003742                                                                  IF      #RCVRDONE SETIN @RCSR THEN
  (6)  003742  032777  000200  175310        BIT     #RCVRDONE,@RCSR
  (9)  003750  001402                        BEQ     $45
 9637
 9638                                                                        ;RCVRDONE SHOULD HAVE CLEARED BY INIT
 9639                                                                        ; RCVRDONE DID NOT CLEAR IN RCSR
 9640  003752                                                                ERRHRD 41,HRESET, DIDNOT
  (1)  003752  104041                        ERROR   41
 9641                                                                        ;REISSUE RESET
 9642  003754                                                                BRESET
  (1)  003754  000005                        RESET
 9643  003756                                                              ENDIF
  (4)  003756                        $45:
 9644                                                                        ;ALLOW LOOPING AFTER ERROR
 9645  003756                                                              CKLOOP
 9646  003756                                                          ENDSUB
 9647  003756                                                      ENDTST
 9648
 9649
 9650
```

```
 9656
 9657                                ;;*****************************************************************
  (3)                                ;*TEST 13        TEST THAT XMITRDY - TCSR 7 - IS SET BY INIT
  (3)                                ;;*****************************************************************
  (2)  003756  000004                TST13:  SCOPE
  (2)
  (1)  003760  012767  000010  175172        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2)  003766  012767  000013  175204        MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 9658
 9663
 9664
 9665
 9666  003774                                                            BGNSUB
  (5)  003774  012767  004002  175106        MOV     #64$,$LPERR
 9667
 9668  004002                                                                IF      #XMITRDY NOTSETIN @TCSR THEN
  (6)  004002  032777  000200  175254        BIT     #XMITRDY,@TCSR
  (9)  004010  001002                        BNE     $46
 9669
 9670                                                                          ;RESET SHOULD HAVE SET BIT.
 9671                                                                          ;XMITRDY DID NOT SET IN TCSR (AFTER RESE
 9672  004012                                                                  ERRHRD 42,HRESET,DIDNOT
  (1)  004012  104042                        ERROR   42
 9673                                                                          ;ISSUE ANOTHER RESET
 9674  004014                                                                  BRESET
  (1)  004014  000005                        RESET
 9675  004016                                                                ENDIF
  (4)  004016                        $46:
 9676                                                                          ;ALLOW LOOPING ON ERROR
 9677  004016                                                                CKLOOP
 9678  004016                                                            ENDSUB
 9679  004016                                                        ENDTST
 9680
 9681
 9682
```

```
 9688
 9689                                ;;*****************************************************************
  (3)                                ;*TEST 14        TEST THAT DATAINT - RCSR 15 - IS CLEARED BY INIT.
  (3)                                ;;*****************************************************************
  (2)  004016  000004                TST14:  SCOPE
  (2)
  (1)  004020  012767  000010  175132        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2)  004026  012767  000014  175144        MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 9690
 9691
 9692
 9697
 9698  004034                                                              BGNSUB
  (5)  004034  012767  004042  175046        MOV     #64$,$LPERR
 9699  004042                                                                  IF      #DATAINT SETIN @RCSR THEN
  (6)  004042  032777  100000  175210        BIT     #DATAINT,@RCSR
  (9)  004050  001402                        BEQ     $47
 9700
 9701  004052                                                                      ERRHRD 43, HRESET, DIDNOT
  (1)  004052  104043                        ERROR   43
 9702
 9703                                                                            ;TESTING EFFECT OF RESET ON BIT
 9704
 9705                                                                            ;DATAINT DID NOT CLEAR IN RCSR
 9706                                                                            ;ALLOW A FRESH START
 9707  004054                                                                    BRESET
  (1)  004054  000005                        RESET
 9708  004056                                                                  ENDIF
  (4)  004056                        $47:
 9709  004056                                                                  CKLOOP
 9710  004056                                                              ENDSUB
 9711  004056                                                          ENDTST
 9712
 9713
 9714
 9719                                ;;*****************************************************************
```

```
 9721
 9722                                ;;*****************************************************************
  (3)                                ;*TEST 15        TEST THAT RCVRACT - RCSR 11 - 15 CLEARED BY INIT
  (3)                                ;;*****************************************************************
  (2)  004056  000004                TST15:  SCOPE
  (2)
  (1)  004060  012767  000010  175072        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2)  004066  012767  000015  175104        MOV     #15,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 9723
 9724
 9729  004074                                                        IF      #CABLE NOTSETIN $USWR THEN
  (6)  004074  032767  020000  175116        BIT     #CABLE,$USWR
  (9)  004102  001004                        BNE     $50
 9730                                                                    ; CAN'T TEST WITHOUT BERG OR H315.
 9731  004104                                                        EXIT TST
  (5)  004104  012767  000001  175046        MOV     #1,$TIMES
  (3)  004112  000411                        BR      TST16           ;;;EXIT THIS TEST
 9732  004114                                                        ENDIF
  (4)  004114                        $50:
 9733
 9734
 9735
 9736  004114                                                        BGNSUB
  (5)  004114  012767  004122  174766        MOV     #64$,$LPERR
 9737
 9738  004122                                                            IF      #RCVRACT SETIN @RCSR THEN
  (6)  004122  032777  004000  175130        BIT     #RCVRACT,@RCSR
  (9)  004130  001402                        BEQ     $51
 9739
 9740                                                                        ;RESET SHOULD HAVE CLEARED RCVRACT
 9741  004132                                                                ERRHRD 44, HRESET, DIDNOT
  (1)  004132  104044                        ERROR   44
 9742
 9743                                                                        ;TESTING EFFECT OF RESET ON BIT
 9744
 9745                                                                        ;RCVRACT DID NOT CLEAR IN RCSR
 9746
 9747                                                                        ;ALLOW ANOTHER TRY
 9748  004134                                                                BRESET
  (1)  004134  000005                        RESET
 9749  004136                                                            ENDIF
  (4)  004136                        $51:
 9750                                                                        ;ALLOW LOOPING ON ERROR
 9751  004136                                                            CKLOOP
 9752  004136                                                        ENDSUB
 9753  004136                                                    ENDTST
 9754
```

```
 9756
 9761                                   ;;*****************************************************************
 9762                                   ;* THE FOLLOWING 4 TESTS VERIFY
 9763                                   ;* THAT THE EIA SIGNALS CAN BE TRANSMITTED
 9764                                   ;* AND RECEIVED THROUGH THE CABLE
 9765                                   ;;*****************************************************************
 9766
 9767
 9768
 9773                                   ;;*****************************************************************
  (3)                                   ;*TEST 16        TEST THAT CARDET SETS AND CLEARS
  (4)                                   ;*              AS  DTR   SETS AND CLEARS
  (4)                                   ;*              THE (-FD) JUMPER MUST BE IN FOR THIS TEST.
  (3)                                   ;;*****************************************************************
  (2) 004136  000004                    TST16:  SCOPE
  (2)
  (1) 004140  012767  000010  175012            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 004146  012767  000016  175024            MOV     #16,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 9778                                                                           ; CAN WE USE THE WRAPAROUND??
 9779 004154                                                            IF      #CABLE+FRFD NOTSETIN $USWR THEN
  (6) 004154  032767  060000  175036            BIT     #CABLE+FRFD,$USWR
  (9) 004162  001004                            BNE     $52
 9780                                                                           ; CAN'T TEST WITHOUT BERG OR H315
 9781                                                                           ; OR WITH (-FD) JUMPER OUT.
 9782                                                                           ; OR WITH (-FR) JUMPER OUT.
 9783 004164                                                                EXIT TST
  (5) 004164  012767  000001  174766            MOV     #1,$TIMES
  (3) 004172  000441                            BR      TST17           ;;;EXIT THIS TEST
 9784 004174                                                                ENDIF
  (4) 004174                            $52:
 9785
 9786                                                                           ; DTR AND
 9787                                                                           ; CARDET ARE CONNECTED
 9788
 9789                                                                   ; BY THE H315 OR EQUIV.
 9790
 9791                                                           ; CLEAR
 9792 004174                                                                BGNSUB
  (5) 004174  012767  004202  174706            MOV     #64$,$LPERR
 9793
 9794                                                                           ; CLEAR  DTR
 9795 004202                                                            LET     @RCSR := @RCSR CLR.BY #DTR
  (7) 004202  042777  000002  175050            BIC     #DTR,@RCSR
 9796                                                                           ; CARDET SHOULD FOLLOW
 9797 004210                                                            IF      #CARDET SETIN @RCSR THEN
  (6) 004210  032777  010000  175042            BIT     #CARDET,@RCSR
  (9) 004216  001401                            BEQ     $53
 9798                                                                           ; CARDET DID NOT
 9799 004220                                                                    ERRHRD 45,,FORCE
  (1) 004220  104045                            ERROR   45
 9800
 9801                                                                           ;  CLEAR WITH DTR
 9802 004222                                                                ENDIF
  (4) 004222                            $53:
 9803 004222                                                            ENDSUB
 9804
```

```
 9805                                                   ; SET
 9806  004222                                                          BGNSUB
  (5)  004222  012767  004230  174660          MOV     #64$,$LPERR
 9807
 9808                                                               ; SET DTR
 9809  004230                                                        LET     @RCSR := @RCSR SET.BY #DTR
  (7)  004230  052777  000002  175022          BIS     #DTR,@RCSR
 9810                                                               ; CARDET SHOULD FOLLOW
 9811  004236                                                        IF      #CARDET NOTSETIN @RCSR THEN
  (6)  004236  032777  010000  175014          BIT     #CARDET,@RCSR
  (9)  004244  001001                          BNE     $54
 9812
 9813  004246                                                            ; CARDET DID NOT SET
                                                                       ERRHRD 46,,FORCE
  (1)  004246  104046                          ERROR   46
 9814
 9815                                                               ;  WITH DTR
 9816  004250                                                        ENDIF
  (4)  004250                          $54:
 9817  004250                                                          ENDSUB
 9818
 9819                                                   ; CLEAR
 9820  004250                                                          BGNSUB
  (5)  004250  012767  004256  174632          MOV     #64$,$LPERR
 9821
 9822                                                               ; CLEAR  DTR
 9823  004256                                                        LET     @RCSR := @RCSR CLR.BY #DTR
  (7)  004256  042777  000002  174774          BIC     #DTR,@RCSR
 9824                                                               ; CARDET SHOULD FOLLOW
 9825  004264                                                        IF      #CARDET SETIN @RCSR THEN
  (6)  004264  032777  010000  174766          BIT     #CARDET,@RCSR
  (9)  004272  001401                          BEQ     $55
 9826                                                                   ; CARDET DID NOT
 9827  004274                                                            ERRHRD 47,,FORCE
  (1)  004274  104047                          ERROR   47
 9828
 9829                                                               ;  CLEAR WITH DTR
 9830  004276                                                        ENDIF
  (4)  004276                          $55:
 9831  004276                                                      ENDSUB
 9832  004276                                                  ENDTST
 9833
 9834
 9835
 9836
```

```
 9838
 9843                    ;;*****************************************************************
 9848                    ;;*****************************************************************
  (3)                    ;*TEST 17        TEST THAT CLRSEND SETS AND CLEARS
  (4)                    ;*              AS  DTR   SETS AND CLEARS
  (4)                    ;*              (-FD) JUMPER MUST BE IN FOR THIS TEST TO WORK
  (3)                    ;;*****************************************************************
  (2) 004276 000004      TST17:  SCOPE
  (2)
  (1) 004300 012767 000010 174652        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 004306 012767 000017 174664        MOV     #17,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 9853                                                                    ; CAN WE USE THE WRAPAROUND??
 9854 004314                                                      IF      #CABLE+FRFD NOTSETIN $USWR THEN
  (6) 004314 032767 060000 174676        BIT     #CABLE+FRFD,$USWR
  (9) 004322 001004                      BNE     $56
 9855                                                                    ; CAN'T TEST WITHOUT BERG OR H315
 9856 004324                                                          EXIT TST
  (5) 004324 012767 000001 174626        MOV     #1,$TIMES
  (3) 004332 000441                      BR      TST20           ;;;EXIT THIS TEST
 9857 004334                                                      ENDIF
  (4) 004334                            $56:
 9858
 9859                                                                    ; DTR AND
 9860                                                                    ; CLRSEND ARE CONNECTED
 9861
 9862                                                             ; BY THE H315 OR EQUIV.
 9863
 9864                                                  ; CLEAR
 9865 004334                                                      BGNSUB
  (5) 004334 012767 004342 174546        MOV     #64$,$LPERR
 9866
 9867                                                                ; CLEAR  DTR
 9868 004342                                                      LET     @RCSR := @RCSR CLR.BY #DTR
  (7) 004342 042777 000002 174710        BIC     #DTR,@RCSR
 9869                                                                ; CLRSEND SHOULD FOLLOW
 9870 004350                                                      IF      #CLRSEND SETIN @RCSR THEN
  (6) 004350 032777 020000 174702        BIT     #CLRSEND,@RCSR
  (9) 004356 001401                      BEQ     $57
 9871                                                                ; CLRSEND DID NOT
 9872 004360                                                      ERRHRD 50,,FORCE
  (1) 004360 104050                      ERROR   50
 9873
 9874                                                                ;  CLEAR WITH DTR
 9875 004362                                                      ENDIF
  (4) 004362                            $57:
 9876 004362                                                   ENDSUB
 9877
 9878                                                  ; SET
 9879 004362                                                   BGNSUB
  (5) 004362 012767 004370 174520        MOV     #64$,$LPERR
 9880
 9881                                                                ; SET DTR
 9882 004370                                                      LET     @RCSR := @RCSR SET.BY #DTR
  (7) 004370 052777 000002 174662        BIS     #DTR,@RCSR
 9883                                                                ; CLRSEND SHOULD FOLLOW
 9884 004376                                                      IF      #CLRSEND NOTSETIN @RCSR THEN
```

```
  (6)  004376  032777  020000  174654          BIT     #CLRSEND,@RCSR
  (9)  004404  001001                          BNE     $60
9885                                                                    ; CLRSEND DID NOT SET
9886  004406                                                            ERRHRD 51,,FORCE
  (1)  004406  104051                          ERROR   51
9887                                                                    ;  WITH DTR
9888                                                                ENDIF
9889  004410
  (4)  004410                          $60:                         ENDSUB
9890  004410
9891
9892                                                                ; CLEAR
9893  004410                                                        BGNSUB
  (5)  004410  012767  004416  174472          MOV     #64$,$LPERR
9894
9895                                                                    ; CLEAR  DTR
9896  004416                                                        LET    @RCSR := @RCSR CLR.BY #DTR
  (7)  004416  042777  000002  174634          BIC     #DTR,@RCSR
9897                                                                    ; CLRSEND SHOULD FOLLOW
9898  004424                                                        IF     #CLRSEND SETIN @RCSR THEN
  (6)  004424  032777  020000  174626          BIT     #CLRSEND,@RCSR
  (9)  004432  001401                          BEQ     $61
9899                                                                    ; CLRSEND DID NOT
9900  004434                                                            ERRHRD 52,,FORCE
  (1)  004434  104052                          ERROR   52
9901                                                                    ;  CLEAR WITH DTR
9902                                                                ENDIF
9903  004436
  (4)  004436                          $61:                         ENDSUB
9904  004436                                                    ENDSUB
9905  004436                                                ENDTST
9906
9907
9908
9909
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 20        TEST THAT RING SETS AND CLEARS
                                ;*              AS  REQSEND    SETS AND CLEARS
                                ;*              THE (-FR) JUMPER MUST BE IN FOR THIS TEST.
                                ;;*****************************************************************
004436  000004                  TST20:  SCOPE

004440  012767  000010  174512          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004446  012767  000020  174524          MOV     #20,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                ; CAN WE USE THE WRAPAROUND??
004454                                                      IF      #CABLE+FRFD NOTSETIN $USWR THEN
004454  032767  060000  174536          BIT     #CABLE+FRFD,$USWR
004462  001004                          BNE     $62
                                                                ; CAN'T TEST WITHOUT BERG OR H315
                                                                ; OR WITH (-FR) JUMPER OUT.
004464                                                        EXIT TST
004464  012767  000001  174466          MOV     #1,$TIMES
004472  000441                          BR      TST21           ;;;EXIT THIS TEST
                                                              ENDIF
004474                          $62:
                                                                ; REQSEND AND
                                                                ; RING ARE CONNECTED

                                                              ; BY THE H315 OR EQUIV.

                                        ; CLEAR
004474                                                        BGNSUB
004474  012767  004502  174406          MOV     #64$,$LPERR

                                                                ; CLEAR  REQSEND
                                                            LET   @RCSR := @RCSR CLR.BY #REQSEND
004502  042777  000004  174550          BIC     #REQSEND,@RCSR
                                                                ; RING SHOULD FOLLOW
                                                            IF    #RING SETIN @RCSR THEN
004510  032777  040000  174542          BIT     #RING,@RCSR
004516  001401                          BEQ     $63

                                                                ; RING DID NOT
                                                                ERRHRD 53,,FORCE
004520  104053                          ERROR   53

                                                                ;  CLEAR WITH REQSEND
                                                            ENDIF
004522                          $63:
004522                                                      ENDSUB

                                                              ; SET
004522                                                      BGNSUB
004522  012767  004530  174360          MOV     #64$,$LPERR

                                                                ; SET REQSEND
                                                            LET   @RCSR := @RCSR SET.BY #REQSEND
004530  052777  000004  174522          BIS     #REQSEND,@RCSR
                                                                ; RING SHOULD FOLLOW
```

```
9958  004536                                                        IF      #RING NOTSETIN @RCSR THEN
 (6)  004536  032777  040000  174514          BIT     #RING,@RCSR
 (9)  004544  001001                          BNE     $64
9959                                                                    ; RING DID NOT SET
9960  004546                                                            ERRHRD 54,,FORCE
 (1)  004546  104054                          ERROR   54
9961                                                                    ;  WITH REQSEND
9962                                                                ENDIF
9963  004550
 (4)  004550                          $64:
9964  004550                                                    ENDSUB
9965
9966                                                                    ; CLEAR
9967  004550                                                    BGNSUB
 (5)  004550  012767  004556  174332          MOV     #64$,$LPERR
9968
9969                                                                    ; CLEAR  REQSEND
9970  004556                                                            LET     @RCSR := @RCSR CLR.BY #REQSEND
 (7)  004556  042777  000004  174474          BIC     #REQSEND,@RCSR
9971                                                                    ; RING SHOULD FOLLOW
9972  004564                                                            IF      #RING SETIN @RCSR THEN
 (6)  004564  032777  040000  174466          BIT     #RING,@RCSR
 (9)  004572  001401                          BEQ     $65
9973                                                                    ; RING DID NOT
9974  004574                                                            ERRHRD 55,,FORCE
 (1)  004574  104055                          ERROR   55
9975
9976                                                                    ;  CLEAR WITH REQSEND
9977  004576                                                        ENDIF
 (4)  004576                          $65:
9978  004576                                                    ENDSUB
9979  004576                                                ENDTST
9980
9981
9982
9983
```

```
 9985
 9990                    ;;*****************************************************************
 9994                    ;;*****************************************************************
  (3)                    ;*TEST 21        TEST THAT SECREC SETS AND CLEARS
  (4)                    ;*              AS  SECXMIT    SETS AND CLEARS
  (3)                    ;;*****************************************************************
  (2) 004576 000004      TST21:  SCOPE
  (2)
  (1) 004600 012767 000010 174352        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 004606 012767 000021 174364        MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 9999                                                                    ; CAN WE USE THE WRAPAROUND??
10000 004614                                                          IF      #CABLE NOTSETIN $USWR THEN
  (6) 004614 032767 020000 174376        BIT     #CABLE,$USWR
  (9) 004622 001004                      BNE     $66
10001                                                                       ; CAN'T TEST WITHOUT BERG OR H315.
10002 004624                                                             EXIT TST
  (5) 004624 012767 000001 174326        MOV     #1,$TIMES
  (3) 004632 000441                      BR      TST22           ;;;EXIT THIS TEST
10003 004634                                                          ENDIF
  (4) 004634                           $66:
10004
10005                                                                       ; SECXMIT AND
10006                                                                       ; SECREC ARE CONNECTED
10007
10008                                                                 ; BY THE H315 OR EQUIV.
10009
10010                                                       ; CLEAR
10011 004634                                                          BGNSUB
  (5) 004634 012767 004642 174246        MOV     #64$,$LPERR
10012
10013                                                                   ; CLEAR  SECXMIT
10014 004642                                                        LET     @RCSR := @RCSR CLR.BY #SECXMIT
  (7) 004642 042777 000010 174410        BIC     #SECXMIT,@RCSR
10015                                                                   ; SECREC SHOULD FOLLOW
10016 004650                                                        IF      #SECREC SETIN @RCSR THEN
  (6) 004650 032777 002000 174402        BIT     #SECREC,@RCSR
  (9) 004656 001401                      BEQ     $67
10017                                                                   ; SECREC DID NOT
10018 004660                                                        ERRHRD 56,,FORCE
  (1) 004660 104056                      ERROR   56
10019
10020                                                                   ;  CLEAR WITH SECXMIT
10021 004662                                                        ENDIF
  (4) 004662                           $67:
10022 004662                                                  ENDSUB
10023
10024                                                       ; SET
10025 004662                                                  BGNSUB
  (5) 004662 012767 004670 174220        MOV     #64$,$LPERR
10026
10027                                                                   ; SET SECXMIT
10028 004670                                                        LET     @RCSR := @RCSR SET.BY #SECXMIT
  (7) 004670 052777 000010 174362        BIS     #SECXMIT,@RCSR
10029                                                                   ; SECREC SHOULD FOLLOW
10030 004676                                                        IF      #SECREC NOTSETIN @RCSR THEN
  (6) 004676 032777 002000 174354        BIT     #SECREC,@RCSR
```

```
   (9)  004704  001001                    BNE     $70
10031                                                          ; SECREC DID NOT SET
10032  004706                                                  ERRHRD 57,,FORCE
   (1)  004706  104057                    ERROR   57
10033
10034                                                          ;  WITH SECXMIT
10035  004710                                                      ENDIF
   (4)  004710                        $70:
10036  004710                                                  ENDSUB
10037
10038                                     ; CLEAR
10039  004710                                                  BGNSUB
   (5)  004710  012767  004716  174172    MOV     #64$,$LPERR
10040
10041                                                          ; CLEAR  SECXMIT
10042  004716                                                      LET     @RCSR := @RCSR CLR.BY #SECXMIT
   (7)  004716  042777  000010  174334    BIC     #SECXMIT,@RCSR
10043                                                          ; SECREC SHOULD FOLLOW
10044  004724                                                      IF      #SECREC SETIN @RCSR THEN
   (6)  004724  032777  002000  174326    BIT     #SECREC,@RCSR
   (9)  004732  001401                    BEQ     $71
10045                                                          ; SECREC DID NOT
10046  004734                                                  ERRHRD 60,,FORCE
   (1)  004734  104060                    ERROR   60
10047
10048                                                          ;  CLEAR WITH SECXMIT
10049  004736                                                      ENDIF
   (4)  004736                        $71:
10050  004736                                                  ENDSUB
10051  004736                                              ENDTST
10052
10053
10054
10055
```

```
10057
10062                                   ;;*****************************************************************
10069                                   ;;*****************************************************************
  (3)                                   ;*TEST 22        TEST THAT DATAINT (RCSR-15) SETS
  (4)                                   ;*              WHEN DTR CHANGES STATE
  (4)                                   ;*              AND THAT DATAINT IS CLEARED AFTER READING RCSR
  (4)                                   ;*              NOTE DTR IS TIED TO BOTH CARDET AND CLRSEND BY THE H315
  (4)                                   ;*              THE (-FD) JUMPER MUST BE IN FOR THIS TEST.
  (3)                                   ;;*****************************************************************
  (2)  004736  000004                   TST22:  SCOPE
  (2)
  (1)  004740  012767  000010  174212           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2)  004746  012767  000022  174224           MOV     #22,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
10070                                                                           ; CAN WE USE THE WRAPAROUND??
10075  004754                                                            IF      #CABLE+FRFD NOTSETIN $USWR THEN
  (6)  004754  032767  060000  174236           BIT     #CABLE+FRFD,$USWR
  (9)  004762  001004                           BNE     $72
10076                                                                           ; CAN'T TEST WITHOUT BERG OR H315
10077                                                                           ; OR WITH (-FD) JUMPER OUT.
10078  004764                                                           EXIT TST
  (5)  004764  012767  000001  174166           MOV     #1,$TIMES
  (3)  004772  000463                           BR      TST23           ;;;EXIT THIS TEST
10079  004774                                                                   ENDIF
  (4)  004774                           $72:
10080
10081                                                                           ;MAKE SURE NOTHING UNEXPECTED HAPPENS
10086  004774  012746  000300                   MOV     #PR6,-(SP)      ;;PUT NEW PS ON STACK
  (1)  005000  012746  005006                   MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
  (1)  005004  000002                           RTI                     ;;POP NEW PC AND PS
  (1)  005006                           64$:
10091
10092                                                           ;READ TWICE - CLEARS
10093  005006                                                                   BGNSUB
  (5)  005006  012767  005014  174074           MOV     #65$,$LPERR
10094                                                                           ; CLEAR DTR
10095  005014                                                           LET     @RCSR := @RCSR CLR.BY #DTR
  (7)  005014  042777  000002  174236           BIC     #DTR,@RCSR
10096                                                                           ;WAIT 1 MILLI-SEC FOR CABLE
10097  005022                                                           WAITMS  1
  (4)  005022  010546                           MOV     R5,-(SP)
  (5)  005024  012745  000001                   MOV     #1,-(R5)
  (4)  005030  004767  004712                   JSR     PC,WAIT
  (4)  005034  012605                           MOV     (SP)+,R5
10098                                                                           ; READ RCSR - TO CLEAR DATAINT
10099  005036                                                           LET     R3 := @RCSR
  (4)  005036  017703  174216                   MOV     @RCSR,R3
10100                                                                           ; READ RCSR AGAIN
10101  005042                                                           IF      #DATAINT SETIN @RCSR THEN
  (6)  005042  032777  100000  174210           BIT     #DATAINT,@RCSR
  (9)  005050  001401                           BEQ     $73
10102                                                                           ; READING RCSR DID NOT CLEAR DATAINT
10103  005052                                                                   ERRHRD 61,EDATAINT
  (1)  005052  104061                           ERROR   61
10104  005054                                                           ENDIF
  (4)  005054                           $73:
10105
```

```
10106  005054                                                                  ENDSUB
10107
10108                                                          ;  DTR SETTING SETS DATAINT
10109  005054                                                                  BGNSUB
  (5)  005054  012767  005062  174026          MOV     #64$,$LPERR
10110
10111                                                                      ;SET DTR
10112  005062                                                            LET     @RCSR  := @RCSR SET.BY #DTR
  (7)  005062  052777  000002  174170          BIS     #DTR,@RCSR
10113  005070                                                            IF      #DATAINT NOTSETIN @RCSR THEN
  (6)  005070  032777  100000  174162          BIT     #DATAINT,@RCSR
  (9)  005076  001001                          BNE     $74
10114                                                                        ;SETTING DTR DID NOT SET DATAINT
10115  005100                                                                ERRHRD 62,, E2DATA
  (1)  005100  104062                          ERROR   62
10116  005102                                                            ENDIF
  (4)  005102                          $74:
10117
10118  005102                                                            IF      #DATAINT SETIN @RCSR THEN
  (6)  005102  032777  100000  174150          BIT     #DATAINT,@RCSR
  (9)  005110  001401                          BEQ     $75
10119                                                                        ;READING RCSR DID NOT CLEAR DATAINT
10120  005112                                                                ERRHRD 63,EDATAINT
  (1)  005112  104063                          ERROR   63
10121  005114                                                            ENDIF
  (4)  005114                          $75:
10122  005114                                                                  ENDSUB
10123
10124                                                          ; DTR CLEARING SETS DATAINT
10125  005114                                                                  BGNSUB
  (5)  005114  012767  005122  173766          MOV     #64$,$LPERR
10126
10127                                                                      ;CLEAR DTR
10128  005122                                                              LET     @RCSR := @RCSR CLR.BY #DTR
  (7)  005122  042777  000002  174130          BIC     #DTR,@RCSR
10129  005130                                                            IF        #DATAINT NOTSETIN @RCSR THEN
  (6)  005130  032777  100000  174122          BIT     #DATAINT,@RCSR
  (9)  005136  001001                          BNE     $76
10130                                                                        ;CLEARING DTR DID NOT SET DATAINT
10131  005140                                                                ERRHRD 64,,E2DATA
  (1)  005140  104064                          ERROR   64
10132  005142                                                              ENDIF
  (4)  005142                          $76:
10133  005142                                                                ENDSUB
10134  005142                                                          ENDTST
10135
10136
10137
```

```
10139
10144                          ;;*****************************************************************
10149                          ;;*****************************************************************
  (3)                          ;*TEST 23        TEST THAT DATAINT SETS WHEN RING SETS
  (4)                          ;*               AND THAT DATAINT DOES NOT SET WHEN RING CLEARS
  (4)                          ;*               THE (-FR) JUMPER MUST BE IN FOR THIS TEST.
  (3)                          ;;*****************************************************************
  (2) 005142 000004            TST23:  SCOPE
  (2)
  (1) 005144 012767 000010 174006      MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 005152 012767 000023 174020      MOV     #23,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
10150                                                          ; CAN WE USE THE WRAPAROUND??
10155 005160                                                     IF      #CABLE+FRFD NOTSETIN $USWR THEN
  (6) 005160 032767 060000 174032      BIT     #CABLE+FRFD,$USWR
  (9) 005166 001004                    BNE     $77
10156                                                          ; CAN'T TEST WITHOUT BERG OR H315
10157                                                          ; OR WITH (-FR) JUMPER OUT.
10158 005170                                                   EXIT TST
  (5) 005170 012767 000001 173762      MOV     #1,$TIMES
  (3) 005176 000473                    BR      TST24           ;;;EXIT THIS TEST
10159 005200                                                   ENDIF
  (4) 005200                   $77:
10160
10161                                                          ;NO INTERRUPTS
10166 005200 012746 000300           MOV     #PR6,-(SP)      ;;PUT NEW PS ON STACK
  (1) 005204 012746 005212           MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
  (1) 005210 000002                  RTI                     ;;POP NEW PC AND PS
  (1) 005212                   64$:
10171
10172                                                  ;START OFF WITH EVERYTHING CLEAR
10173 005212                                                   BGNSUB
  (5) 005212 012767 005220 173670      MOV     #65$,$LPERR
10174
10175                                                          ;CLEAR RING
10176 005220                                                   LET     @RCSR := @RCSR CLR.BY #REQSEND
  (7) 005220 042777 000004 174032      BIC     #REQSEND,@RCSR
10177                                                          ;WAIT 1 MILLI-SEC FOR CABLE
10178 005226                                                   WAITMS  1
  (4) 005226 010546                    MOV     R5,-(SP)
  (5) 005230 012745 000001             MOV     #1,-(R5)
  (4) 005234 004767 004506             JSR     PC,WAIT
  (4) 005240 012605                    MOV     (SP)+,R5
10179                                                          ;READ ONCE
10180 005242                                                   LET     R3 := @RCSR
  (4) 005242 017703 174012             MOV     @RCSR,R3
10181                                                          ;READ TWICE
10182 005246                                                   IF      #DATAINT SETIN @RCSR THEN
  (6) 005246 032777 100000 174004      BIT     #DATAINT,@RCSR
  (9) 005254 001401                    BEQ     $100
10183                                                          ;READING RCSR DID NOT CLEAR DATAINT
10184 005256                                                   ERRHRD 65, EDATAINT
  (1) 005256 104065                    ERROR   65
10185 005260                                                   ENDIF
  (4) 005260                   $100:
10186 005260                                                   ENDSUB
10187
```

```
10188                                                   ;       SET RING --> SET DATAINT
10189  005260                                                           BGNSUB
  (5)  005260  012767  005266  173622          MOV     #64$,$LPERR
10190
10191                                                                   ;WE ARE SETTING RING
10192                                                                   ; SET RING
10193  005266                                                       LET     @RCSR := @RCSR SET.BY #REQSEND
  (7)  005266  052777  000004  173764          BIS     #REQSEND,@RCSR
10194                                                                   ;WAIT 1 MILLI-SEC FOR CABLE
10195  005274                                                       WAITMS  1
  (4)  005274  010546                          MOV     R5,-(SP)
  (5)  005276  012745  000001                  MOV     #1,-(R5)
  (4)  005302  004767  004440                  JSR     PC,WAIT
  (4)  005306  012605                          MOV     (SP)+,R5
10196  005310                                                       IF      #DATAINT NOTSETIN @RCSR THEN
  (6)  005310  032777  100000  173742          BIT     #DATAINT,@RCSR
  (9)  005316  001001                          BNE     $101
10197                                                                   ;SETTING RING DID NOT SET DATAINT
10198  005320                                                           ERRHRD 122,, E2DATA
  (1)  005320  104122                          ERROR   122
10199  005322                                                       ENDIF
  (4)  005322                          $101:
10200  005322                                                   ENDSUB
10201
10202                                                   ;CLEAR RING CAUSES NO CHANGE IN DATAINT (CLEAR)
10203  005322                                                           BGNSUB
  (5)  005322  012767  005330  173560          MOV     #64$,$LPERR
10204
10205                                                                   ;CLEAR RING
10206  005330                                                       LET     @RCSR := @RCSR CLR.BY #REQSEND
  (7)  005330  042777  000004  173722          BIC     #REQSEND,@RCSR
10207                                                                   ;WAIT 1 MILLI-SEC FOR CABLE
10208  005336                                                       WAITMS  1
  (4)  005336  010546                          MOV     R5,-(SP)
  (5)  005340  012745  000001                  MOV     #1,-(R5)
  (4)  005344  004767  004376                  JSR     PC,WAIT
  (4)  005350  012605                          MOV     (SP)+,R5
10209  005352                                                       IF      #DATAINT SETIN @RCSR THEN
  (6)  005352  032777  100000  173700          BIT     #DATAINT,@RCSR
  (9)  005360  001401                          BEQ     $102
10210                                                                   ;CLEARING RING SET DATAINT
10211  005362                                                           ERRHRD 123, CLRRNG
  (1)  005362  104123                          ERROR   123
10212  005364                                                       ENDIF
  (4)  005364                          $102:
10213  005364                                                   ENDSUB
10214  005364                                                   EXIT    ;SKIP AROUND MESSAGE
  (3)  005364  000400                          BR      TST24           ;;;EXIT THIS TEST
10215  005366                                                       ENDTST
10216
10217
10218
```

```
10220
10225                                   ;;*****************************************************************
10226                                   ;;*****************************************************************
  (3)                                   ;*TEST 24        TEST THAT DATAINT SETS WHEN SECREC CHANGES STATE
  (3)                                   ;;*****************************************************************
  (2)  005366  000004                   TST24:  SCOPE
  (2)
  (1)  005370  012767  000010  173562           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2)  005376  012767  000024  173574           MOV     #24,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
10227                                                                           ; CAN WE USE THE WRAPAROUND??
10232  005404                                                                 IF      #CABLE NOTSETIN $USWR THEN
  (6)  005404  032767  020000  173606           BIT     #CABLE,$USWR
  (9)  005412  001004                           BNE     $103
10233                                                                               ; CAN'T TEST WITHOUT BERG OR H315.
10234  005414                                                                   EXIT TST
  (5)  005414  012767  000001  173536           MOV     #1,$TIMES
  (3)  005422  000454                           BR      TST25           ;;;EXIT THIS TEST
10235  005424                                                                   ENDIF
  (4)  005424                           $103:
10236
10237                                                                           ;NO INTERRUPTS
10242  005424  012746  000300                   MOV     #PR6,-(SP)      ;;PUT NEW PS ON STACK
  (1)  005430  012746  005436                   MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
  (1)  005434  000002                           RTI                     ;;POP NEW PC AND PS
  (1)  005436                           64$:
10243
10248
10249                                                                   ;START FRESH
10250                                                                           ;CLEAR SECREC
10251  005436                                                           LET     @RCSR := @RCSR CLR.BY #SECXMIT
  (7)  005436  042777  000010  173614           BIC     #SECXMIT,@RCSR
10252  005444                                                           LET     R3 := @RCSR
  (4)  005444  017703  173610                   MOV     @RCSR,R3
10253
10254                                                           ;SET SECREC --> DATAINT SET
10255  005450                                                   BGNSUB
  (5)  005450  012767  005456  173432           MOV     #65$,$LPERR
10256
10257                                                                   ;SET SECREC
10258  005456                                                       LET     @RCSR := @RCSR SET.BY #SECXMIT
  (7)  005456  052777  000010  173574           BIS     #SECXMIT,@RCSR
10259                                                                   ;WAIT 1 MILLI-SEC FOR CABLE
10260  005464                                                       WAITMS  1
  (4)  005464  010546                           MOV     R5,-(SP)
  (5)  005466  012745  000001                   MOV     #1,-(R5)
  (4)  005472  004767  004250                   JSR     PC,WAIT
  (4)  005476  012605                           MOV     (SP)+,R5
10261  005500                                                       IF      #DATAINT NOTSETIN @RCSR THEN
  (6)  005500  032777  100000  173552           BIT     #DATAINT,@RCSR
  (9)  005506  001001                           BNE     $104
10262                                                                   ;SETTING SECREC DID NOT SET DATAINT
10263  005510                                                           ERRHRD 124,, E2DATA
  (1)  005510  104124                           ERROR   124
10264  005512                                                       ENDIF
  (4)  005512                           $104:
10265  005512                                                   ENDSUB
```

```
10266
10267                                                   ;CLEAR SECREC --> DATAINT SET
10268   005512                                                  BGNSUB
  (5)   005512  012767  005520  173370          MOV     #64$,$LPERR
10269                                                               ;CLEAR SECREC
10270   005520                                                  LET     @RCSR := @RCSR CLR.BY #SECXMIT
  (7)   005520  042777  000010  173532          BIC     #SECXMIT,@RCSR
10271                                                               ;WAIT 1 MILLI-SEC FOR CABLE
10272   005526                                                  WAITMS  1
  (4)   005526  010546                          MOV     R5,-(SP)
  (5)   005530  012745  000001                  MOV     #1,-(R5)
  (4)   005534  004767  004206                  JSR     PC,WAIT
  (4)   005540  012605                          MOV     (SP)+,R5
10273   005542                                                  IF      #DATAINT NOTSETIN @RCSR THEN
  (6)   005542  032777  100000  173510          BIT     #DATAINT,@RCSR
  (9)   005550  001001                          BNE     $105
10274                                                               ;CLEARING SECREC DID NOT SET DATAINT
10275   005552                                                      ERRHRD 125,, E2DATA
  (1)   005552  104125                          ERROR   125
10276   005554                                                  ENDIF
  (4)   005554                          $105:
10277   005554                                              ENDSUB
10278   005554                                          ENDTST
10279
10280
10281
```

```
10283
10288                                   ;;*****************************************************************
10293                                   ;;*****************************************************************
  (3)                                   ;*TEST 25        TEST THAT XMIT RDY - TCSR 7 - CLEARS
  (4)                                   ;*               WHEN TBUF IS LOADED WITH A CHARACTER
  (4)                                   ;*               AND THAT IT SETS WITHIN A REASONABLE AMOUNT OF TIME.
  (3)                                   ;;*****************************************************************
  (2)  005554  000004                   TST25:  SCOPE
  (2)
  (1)  005556  012767  000001  173374           MOV     #1,$TIMES       ;;DO 1 ITERATION
  (2)  005564  012767  000025  173406           MOV     #25,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
10298
10299                                                   ; THIS TEST IS 'BREAK OR HALT' SINSATIVE.
10300  005572                                                           IF #APTENV SETIN $ENV THEN
  (6)  005572  032767  000001  173414           BIT     #APTENV,$ENV
  (9)  005600  001404                           BEQ     $106
10301  005602                                                                   EXIT TEST
  (5)  005602  012767  000001  173350           MOV     #1,$TIMES
  (3)  005610  000454                           BR      TST26           ;;;EXIT THIS TEST
10302  005612                                                           ENDIF
  (4)  005612                           $106:
10303  005612                                                                   BGNSUB
  (5)  005612  012767  005620  173270           MOV     #64$,$LPERR
10304                                                       ; LOAD TBUF WITH ONE CHARACTER
10305                                                       ; WAIT FOR READY TO SET
10306                                                       ; (SHOULD BE VERY SHORT WAIT
10307                                                       ; SINCE UART DOUBLE BUFFERS ITS INPUT)
10308
10309                                                                       ;SEND A CHARACTER
10310  005620                                                           LET @TBUF :B= #0
  (4)  005620  105077  173444                   CLRB    @TBUF
10311                                                                      ;WAIT A MAXIMUM
10312                                                                      ;OF 50 MSEC FOR
10313                                                                      ;XMIT RDY TO SET IN TCSR
10314  005624                                                           CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
  (3)  005624  010546                           MOV     R5,-(SP)
  (7)  005626  012745  177777                   MOV     #SET,-(R5)
  (6)  005632  016745  173426                   MOV     TCSR,-(R5)
  (5)  005636  012745  000200                   MOV     #XMITRDY,-(R5)
  (4)  005642  012745  000500                   MOV     #500,-(R5)
  (3)  005646  004767  003616                   JSR     PC,TIMER
  (3)  005652  012605                           MOV     (SP)+,R5
10315                                                                      ;TIMER RETURNS AN ERROR IF BIT DID
10316                                                                      ;NOT MEET CONDITION WITHIN TIME LIMIT
10317  005654                                                           IF.ERROR THEN
  (6)  005654  103001                           BCC     $107
10318                                                                      ;XMIT RDY DID NOT SET IN TCSR
10319  005656                                                              ERRHRD 66,,DIDNOT
  (1)  005656  104066                           ERROR   66
10320  005660                                                           ENDIF
  (4)  005660                           $107:
10321  005660                                                       ENDSUB
10322
10323  005660                                                       BGNSUB
  (5)  005660  012767  005666  173222           MOV     #64$,$LPERR
10324                                                       ; LOAD TBUF WITH A SECOND CHARACTER
```

```
10325                                                   ; CHECK IMMEDIATELY THAT XMITRDY IS CLEAR
10326                                                   ; AND THEN WAIT FOR IT TO SET
10327
10328                                                                   ;SEND SECOND CHARACTER
10329  005666                                                        LET @TBUF :B= #0
  (4)  005666  105077  173376          CLRB    @TBUF
10330  005672  000240                          NOP                ; GIVE IT TIME TO CLEAR
10331                                                                  ; XMITRDY SHOULD HAVE CLEARED UPON
10332                                                                  ; RECEIPT OF A CHARACTER
10333  005674                                                        IF #XMITRDY SETIN @TCSR THEN
  (6)  005674  032777  000200  173362    BIT     #XMITRDY,@TCSR
  (9)  005702  001401                    BEQ     $110
10334
10335  005704                                                            ; XMITRDY DID NOT CLEAR IN TCSR
                                                                         ERRHRD 67,,DIDNOT
  (1)  005704  104067                    ERROR   67
10336  005706                                                        ENDIF
  (4)  005706                    $110:
10337
10338                                                                  ;WAIT A MAXIMUM
10339                                                                  ;OF 50 MSEC FOR
10340                                                                  ;XMIT RDY TO SET IN TCSR
10341  005706                                                        CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
  (3)  005706  010546                    MOV     R5,-(SP)
  (7)  005710  012745  177777            MOV     #SET,-(R5)
  (6)  005714  016745  173344            MOV     TCSR,-(R5)
  (5)  005720  012745  000200            MOV     #XMITRDY,-(R5)
  (4)  005724  012745  000500            MOV     #500,-(R5)
  (3)  005730  004767  003534            JSR     PC,TIMER
  (3)  005734  012605                    MOV     (SP)+,R5
10342  005736                                                        IF.ERROR THEN
  (6)  005736  103001                    BCC     $111
10343                                                                   ;XMIT RDY DID NOT SET IN TCSR
10344  005740                                                           ERRHRD 70,,DIDNOT
  (1)  005740  104070                    ERROR   70
10345  005742                                                        ENDIF
  (4)  005742                    $111:
10346  005742                                                    ENDSUB
10347  005742                                                ENDTST
```

```
10349
10354                                   ;;*****************************************************************
10359                                   ;;*****************************************************************
  (3)                                   ;*TEST 26        TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH MAINT SET)
  (4)                                   ;*               RESULTS IN RCVRDONE SETTING WITHIN A REASONABLE AMOUNT OF TIME
  (4)                                   ;*               AND THAT RESET CLEARS THE BIT.
  (3)                                   ;*****************************************************************
  (2)  005742  000004                   TST26:  SCOPE
  (2)
  (1)  005744  012767  000010  173206           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2)  005752  012767  000026  173220           MOV     #26,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
10364
10365                                                                           ; SET THE MAINTENANCE BIT
10366  005760                                                                 LET @TCSR := @TCSR SET.BY #MAINT
  (7)  005760  052777  000004  173276           BIS     #MAINT,@TCSR
10367
10368  005766                                                                 BGNSUB
  (5)  005766  012767  005774  173114           MOV     #64$,$LPERR
10369                                               ; SEND A CHARACTER AND LET IT WRAP AROUND
10370
10371  005774                                                                   LET @TBUF :B= #0
  (4)  005774  105077  173270                   CLRB    @TBUF
10372
10373                                                                           ; WAIT A MAXIMUM OF 50 MSEC
10374                                                                           ; FOR RCVR DONE TO SET IN
10375                                                                           ; RCSR
10376  006000                                                                   CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
  (3)  006000  010546                           MOV     R5,-(SP)
  (7)  006002  012745  177777                   MOV     #SET,-(R5)
  (6)  006006  016745  173246                   MOV     RCSR,-(R5)
  (5)  006012  012745  000200                   MOV     #RCVRDONE,-(R5)
  (4)  006016  012745  000500                   MOV     #500,-(R5)
  (3)  006022  004767  003442                   JSR     PC,TIMER
  (3)  006026  012605                           MOV     (SP)+,R5
10377
10378                                                                              ;DIDN'T SET IN TIME
10379  006030                                                                   IF.ERROR THEN
  (6)  006030  103001                           BCC     $112
10380                                                                              ; RCVRDONE DID NOT SET IN RCSR
10381  006032                                                                      ERRHRD 71,,DIDNOT
  (1)  006032  104071                           ERROR   71
10382  006034                                                                   ENDIF
  (4)  006034                           $112:
10383
10384  006034                                                                 ENDSUB
10385
10386  006034                                                                 BGNSUB
  (5)  006034  012767  006042  173046           MOV     #64$,$LPERR
10387                                               ; NOW THAT IT IS SET SEE IF IT CAN BE RESET
10388                                               ; THIS ALSO WILL CLEAR THE MAINT. BIT
10389  006042                                                                   BRESET
  (1)  006042  000005                           RESET
10390
10391  006044                                                                   IF #RCVRDONE SETIN @RCSR THEN
  (6)  006044  032777  000200  173206           BIT     #RCVRDONE,@RCSR
  (9)  006052  001401                           BEQ     $113
```

```
10392                                                                   ; RCVRDONE DID NOT RESET IN RCSR.
10393  006054                                                           ERRHRD 72,,DIDNOT
  (1)  006054  104072                   ERROR   72
10394  006056                                                               ENDIF
  (4)  006056                  $113:
10395  006056                                                           ENDSUB
10396  006056                                                       ENDTST
```

```
10398
10403                                   ;;*****************************************************************
10404                                   ;;*****************************************************************
  (3)                                   ;*TEST 27        TEST THAT RCVRDONE IS CLEARED BY READING RBUF
  (3)                                   ;;*****************************************************************
  (2) 006056  000004                    TST27:  SCOPE
  (2)
  (1) 006060  012767  000010  173072            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 006066  012767  000027  173104            MOV     #27,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
10405
10410                                                                   ; SET MAINT. BIT
10411 006074                                                          LET @TCSR := @TCSR SET.BY #MAINT
  (7) 006074  052777  000004  173162            BIS     #MAINT,@TCSR
10412 006102                                                          BGNSUB
  (5) 006102  012767  006110  173000            MOV     #64$,$LPERR
10413                                               ; OUTPUT A CHARACTER WITH MAINTENANCE
10414                                               ; SET, AND WAIT FOR XMITRDY TO SET.
10415
10416                                                                       ; OUTPUT A CHARACTER
10417 006110                                                            LET @TBUF :B= #0
  (4) 006110  105077  173154                    CLRB    @TBUF
10418                                                                   ; WAIT MAXIMUM OF 500 MSEC
10419                                                                   ; FOR RCVRDONE TO SET IN
10420                                                                   ; RCSR
10421 006114                                                            CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
  (3) 006114  010546                            MOV     R5,-(SP)
  (7) 006116  012745  177777                    MOV     #SET,-(R5)
  (6) 006122  016745  173132                    MOV     RCSR,-(R5)
  (5) 006126  012745  000200                    MOV     #RCVRDONE,-(R5)
  (4) 006132  012745  000500                    MOV     #500,-(R5)
  (3) 006136  004767  003326                    JSR     PC,TIMER
  (3) 006142  012605                            MOV     (SP)+,R5
10422                                                                       ; DID IT BECAME READY?
10423 006144                                                            IF.ERROR  THEN
  (6) 006144  103001                            BCC     $114
10424                                                                       ;RCVRDONE DID NOT SET IN RCSR
10425 006146                                                                ERRHRD 73,, DIDNOT
  (1) 006146  104073                            ERROR   73
10426 006150                                                            ENDIF
  (4) 006150                            $114:
10427 006150                                                          ENDSUB
10428
10429                                               ; NOW THAT IT IS SET LETS SEE IF READING THE
10430                                               ; BUFFER CLEARS RCVRDONE.
10431
10432                                                                       ;READ BUFFER
10433 006150                                                            LET R0 :B= @RBUF
  (4) 006150  117700  173106                    MOVB    @RBUF,R0
10434
10435 006154                                                            IF #RCVRDONE SETIN @RCSR THEN
  (6) 006154  032777  000200  173076            BIT     #RCVRDONE,@RCSR
  (9) 006162  001401                            BEQ     $115
10436                                                                       ;RCVRDONE DID NOT CLEAR IN RCSR
10437 006164                                                                ERRHRD 74,DIDNOT
  (1) 006164  104074                            ERROR   74
10438 006166                                                            ENDIF
```

```
   (4)  006166                    $115:
10439   006166                                        ENDTST
```

```
10441
10446                                   ;;*****************************************************************
10451                                   ;;*****************************************************************
  (3)                                   ;*TEST 30       TEST THAT RCVRACT - RCSR 11 - SETS
  (4)                                   ;*              WHEN A START BIT IS RECEIVED AND
  (4)                                   ;*              CLEARS WHEN RCVRDONE - RCSR 7 - SETS
  (3)                                   ;;*****************************************************************
  (2) 006166  000004                    TST30:  SCOPE
  (2)
  (1) 006170  012767  000010  172762            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 006176  012767  000030  172774            MOV     #30,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
10456
10457                                                   ; THIS TEST IS 'BREAK OR HALT' SINSATIVE.
10458 006204                                                            IF #APTENV SETIN $ENV THEN
  (6) 006204  032767  000001  173002            BIT     #APTENV,$ENV
  (9) 006212  001404                            BEQ     $116
10459 006214                                                                            EXIT TEST
  (5) 006214  012767  000001  172736            MOV     #1,$TIMES
  (3) 006222  000500                            BR      TST31           ;;;EXIT THIS TEST
10460 006224                                                            ENDIF
  (4) 006224                            $116:
10461 006224                                                            LET @TCSR := @TCSR SET.BY #MAINT
  (7) 006224  052777  000004  173032            BIS     #MAINT,@TCSR
10462 006232                                                                    LET R0 := #CLR
  (4) 006232  012700  000000                    MOV     #CLR,R0
10463 006236                                                                    LET R1 := #0
  (4) 006236  005001                            CLR     R1
10464                                                       ;LOAD A CHARACTER INTO TBUF
10465                                                       ;WAIT FOR RCVRACT TO SET
10466
10467                                                               ;SEND A CHARACTER
10468 006240                                                    LET @TBUF :B= #0
  (4) 006240  105077  173024                    CLRB    @TBUF
10469 006244                                                REPEAT
  (3) 006244                            $117:
10470 006244                                                    IF #RCVRACT SETIN @RCSR THEN
  (6) 006244  032777  004000  173006            BIT     #RCVRACT,@RCSR
  (9) 006252  001403                            BEQ     $120
10471 006254                                                        LET R0 := #SET
  (4) 006254  012700  177777                    MOV     #SET,R0
10472 006260                                                    ELSE
  (4) 006260  000401                            BR      $121
  (3) 006262                            $120:
10473 006262                                                        LET R1 := R1 + #1
  (7) 006262  005201                            INC     R1
10474 006264                                                    ENDIF
  (4) 006264                            $121:
10475 006264                                                UNTIL R0 EQ #SET OR R1 HI MAX
  (4) 006264  020027  177777                    CMP     R0,#SET
  (6) 006270  001403                            BEQ     $122
  (4) 006272  020167  000124                    CMP     R1,MAX
  (7) 006276  101762                            BLOS    $117
  (4) 006300                            $122:
10476 006300                                                IF R1 HI MAX THEN
  (6) 006300  020167  000116                    CMP     R1,MAX
  (9) 006304  101407                            BLOS    $123
```

```
10477                                                                   ;IT NEVER SET
10478                                                                   ;RCVRACT DID NOT SET IN RCSR.
10479  006306                                                           ERRHRD 75,, DIDNOT
  (1)  006306  104075                    ERROR   75
10480  006310                                                           LET R0 := @RBUF ; CLEAR BUFFER
  (4)  006310  017700  172746            MOV     @RBUF,R0
10481  006314                                                           EXIT TEST
  (5)  006314  012767  000001  172636    MOV     #1,$TIMES
  (3)  006322  000440                    BR      TST31           ;;;EXIT THIS TEST
10482  006324                                                                   ENDIF
  (4)  006324                        $123:
10483
10484
10485                                                   ;CHECK FOR TIMING OF RCVRACT. CLEARING
10486                                                   ;VS RCVRDONE SETTING
10487
10488
10489  006324                                                           WHILE #RCVRACT SETIN @RCSR DO
  (4)  006324                        $124:
  (6)  006324  032777  004000  172726    BIT     #RCVRACT,@RCSR
  (9)  006332  001416                    BEQ     $125
10490
10491  006334                                                           IF #RCVRDONE SETIN @RCSR THEN
  (6)  006334  032777  000200  172716    BIT     #RCVRDONE,@RCSR
  (9)  006342  001411                    BEQ     $126
10492  006344                                                              IF #RCVRACT SETIN @RCSR THEN
  (6)  006344  032777  004000  172706    BIT     #RCVRACT,@RCSR
  (9)  006352  001405                    BEQ     $127
10493                                                                   ;RCVRDONE AND RCVRACT
10494                                                                       ;BOTH SET
10495  006354                                                               ERRHRD 76, DONEACT
  (1)  006354  104076                    ERROR   76
10496                                                                   ;NO USE CONTINUING
10497  006356                                                               EXIT TST
  (5)  006356  012767  000001  172574    MOV     #1,$TIMES
  (3)  006364  000417                    BR      TST31           ;;;EXIT THIS TEST
10498  006366                                                                ENDIF
  (4)  006366                        $127:
10499  006366                                                              ENDIF
  (4)  006366                        $126:
10500  006366                                                           ENDDO
  (4)  006366  000756                    BR      $124
  (3)  006370                        $125:
10501
10502                                                                   ;RCVRACT = 0 NOW.
10503  006370                                                           IF #RCVRDONE NOTSETIN @RCSR THEN
  (6)  006370  032777  000200  172662    BIT     #RCVRDONE,@RCSR
  (9)  006376  001001                    BNE     $130
10504                                                                       ;RCVRDONE DID NOT SET IN RCSR
10505  006400                                                               ERRHRD 77,,DIDNOT
  (1)  006400  104077                    ERROR   77
10506                                                                       ;SET IT BACK.
10507  006402                                                           ENDIF
  (4)  006402                        $130:
10508                                                   ;TEST THAT READING THE RECEIVER
10509                                                   ;BUFFER CLEARS RCVRDONE
```

```
10510
10511
10512                                                                   ;READ CHAR.
10513  006402                                                         LET R0 := @RBUF
  (4)  006402  017700  172654          MOV     @RBUF,R0
10514
10515  006406                                                         IF #RCVRDONE SETIN @RCSR THEN
  (6)  006406  032777  000200  172644  BIT     #RCVRDONE,@RCSR
  (9)  006414  001401                  BEQ     $131
10516
10517  006416                                                             ;RCVRDONE DID NOT CLEAR IN RCSR
                                                                          ERRHRD 100,,DIDNOT
  (1)  006416  104100                  ERROR   100
10518  006420                                                         ENDIF
  (4)  006420                      $131:
10519
10520  006420                                                                 EXIT
  (3)  006420  000401                  BR      TST31              ;;;EXIT THIS TEST
10521  006422  070000                  MAX:70000
10522
10523  006424                                                             ENDTST
10524
```

```
10526
10531
10532                          ;;*****************************************************************
10533                          ;;*****************************************************************
  (3)                          ;*TEST 31       TEST THE OVERRUN BIT - RBUF 14
  (3)                          ;;*****************************************************************
  (2) 006424  000004           TST31:  SCOPE
  (2)
  (1) 006426  012767  000010  172524        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 006434  012767  000031  172536        MOV     #31,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
10538
10539 006442                                                              BGNSUB
  (5) 006442  012767  006450  172440        MOV     #64$,$LPERR
10540                                           ;OUTPUT 2 CHARACTERS WITH
10541                                           ;AMPLE DELAYS BETWEEN FOR RECEPTION.
10542                                           ;THIS SHOULD AN CAUSE OVERRUN ERROR.
10543
10544                                                                   ;OUTPUT 1 CHARACTER
10545 006450                                                         LET @TBUF :B= #0
  (4) 006450  105077  172614                CLRB    @TBUF
10546                                                                   ;GO AWAY FOR 200. M SEC
10547 006454                                                         WAITMS 200.
  (4) 006454  010546                        MOV     R5,-(SP)
  (5) 006456  012745  000310                MOV     #200.,-(R5)
  (4) 006462  004767  003260                JSR     PC,WAIT
  (4) 006466  012605                        MOV     (SP)+,R5
10548
10549                                                                   ;OUTPUT 2ND CHARACTER
10550 006470                                                         LET @TBUF :B= #0
  (4) 006470  105077  172574                CLRB    @TBUF
10551                                                                   ;LET OVERRUN HAPPEN
10552 006474                                                         WAITMS 200.
  (4) 006474  010546                        MOV     R5,-(SP)
  (5) 006476  012745  000310                MOV     #200.,-(R5)
  (4) 006502  004767  003240                JSR     PC,WAIT
  (4) 006506  012605                        MOV     (SP)+,R5
10553
10554                                                                   ;READ BUFFER AND ERROR BITS
10555 006510                                                         LET R4 := @RBUF
  (4) 006510  017704  172546                MOV     @RBUF,R4
10556
10557                                                                   ;IT DIDN'T SET
10558 006514                                                         IF #ORERR NOTSETIN R4 THEN
  (6) 006514  032704  040000                BIT     #ORERR,R4
  (9) 006520  001005                        BNE     $132
10559                                                                   ;ORERR DID NOT SET IN RBUF
10560 006522                                                         ERRHRD 101,,DIDNOT
  (1) 006522  104101                        ERROR   101
10561
10562                                                                   ;NO USE COMPOUNDING ERRORS
10563 006524                                                         EXIT TST
  (5) 006524  012767  000001  172426        MOV     #1,$TIMES
  (3) 006532  000456                        BR      TST32       ;;;EXIT THIS TEST
10564 006534                                                                 ENDIF
  (4) 006534                          $132:
10565 006534                                                              ENDSUB
```

```
10566
10567                                                   ;NOW SEE IF ERROR BIT SET WITH OVERRUN  ERROR:
10568  006534                                                                   BGNSUB
  (5)  006534  012767  006542  172346          MOV     #64$,$LPERR
10569  006542                                                                       IF #ERROR NOTSETIN R4 THEN
  (6)  006542  032704  100000                  BIT     #ERROR,R4
  (9)  006546  001005                          BNE     $133
10570
10571                                                                                ;ERROR DID NOT SET IN RBUF
10572  006550                                                                        ERRHRD 102,,DIDNOT
  (1)  006550  104102                          ERROR   102
10573
10574                                                                                ;-WHEN ORERR SET.
10575                                                                                ;GET OUT NOW.
10576  006552                                                                        EXIT TST
  (5)  006552  012767  000001  172400          MOV     #1,$TIMES
  (3)  006560  000443                          BR      TST32           ;;;EXIT THIS TEST
10577  006562                                                                    ENDIF
  (4)  006562                          $133:
10578  006562                                                                   ENDSUB
10579
10580  006562                                                                   BGNSUB
  (5)  006562  012767  006570  172320          MOV     #64$,$LPERR
10581                                               ;CHECK REAL RBUF TO SEE IF ORERR IS STILL SET.
10582
10583  006570                                                                       IF #ORERR NOTSETIN @RBUF THEN
  (6)  006570  032777  040000  172464          BIT     #ORERR,@RBUF
  (9)  006576  001002                          BNE     $134
10584
10585                                                                                ;READING RBUF CLEARED ORERR.
10586  006600                                                                        ERRHRD 103,ITCLRED
  (1)  006600  104103                          ERROR   103
10587                                                                                ;SKIP REST OF TEST
10588  006602                                                                        EXIT
  (3)  006602  000432                          BR      TST32           ;;;EXIT THIS TEST
10589  006604                                                                    ENDIF
  (4)  006604                          $134:
10590  006604                                                                   ENDSUB
10591
10592  006604                                                                   BGNSUB
  (5)  006604  012767  006612  172276          MOV     #64$,$LPERR
10593                                               ;NOW SEE IF THEY CLEAR WHEN ANOTHER CHAR. IS RECEIVED
10594
10595                                                                                ;SEND A CHARACTER AROUND.
10596  006612                                                                    LET @TBUF :B= #0
  (4)  006612  105077  172452                  CLRB    @TBUF
10597                                                                                ;LET IT CIRCULATE
10598  006616                                                                    WAITMS 200.
  (4)  006616  010546                          MOV     R5,-(SP)
  (5)  006620  012745  000310                  MOV     #200.,-(R5)
  (4)  006624  004767  003116                  JSR     PC,WAIT
  (4)  006630  012605                          MOV     (SP)+,R5
10599
10600  006632                                                                       IF #ORERR SETIN @RBUF THEN
  (6)  006632  032777  040000  172422          BIT     #ORERR,@RBUF
  (9)  006640  001405                          BEQ     $135
```

```
10601                                                                   ;ORERR DID NOT CLEAR IN RBUF
10602  006642                                                           ERRHRD 104,,DIDNOT
  (1)  006642  104104                    ERROR   104
10603
10604                                                                   ;-AFTER RECEIVING ANOTHER CHAR
10605                                                                   ;SKIP AROUND REST
10606  006644                                                           EXIT TST
  (5)  006644  012767  000001  172306    MOV     #1,$TIMES
  (3)  006652  000406                    BR      TST32          ;;;EXIT THIS TEST
                                                                            ENDIF
10607  006654
  (4)  006654                    $135:
10608
10609  006654                                                                IF #ERROR SETIN @RBUF THEN
  (6)  006654  032777  100000  172400    BIT     #ERROR,@RBUF
  (9)  006662  001401                    BEQ     $136
10610                                                                       ;ERROR DID NOT CLEAR IN RBUF
10611  006664                                                               ERRHRD 105,,DIDNOT
  (1)  006664  104105                    ERROR   105
10612
10613  006666                                                                ENDIF
  (4)  006666                    $136:
10614  006666                                                           ENDSUB
10615  006666                                                           EXIT
  (3)  006666  000400                    BR      TST32          ;;;EXIT THIS TEST
10616                                                                    .EVEN
10617  006670                                                       ENDTST
10618
```

```
10620
10621
10626                                   ;;*****************************************************************
10632                                   ;;*****************************************************************
  (3)                                   ;*TEST 32        PROGRAMMABLE BAUD RATE TEST
  (4)                                   ;*               TEST AT ALL SPEEDS AVAILABLE
  (4)                                   ;*               A COMPARISON WILL BE MADE TO SEE
  (4)                                   ;*               IF NEW TIME IS LESS THAN PREVIOUS.
  (3)                                   ;;*****************************************************************
  (2)  006670  000004                   TST32:  SCOPE
  (2)
  (1)  006672  012767  000010  172260           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2)  006700  012767  000032  172272           MOV     #32,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                                IF #PBR NOTSETIN $USWR THEN
10637  006706
  (6)  006706  032767  000200  172304           BIT     #PBR,$USWR
  (9)  006714  001004                           BNE     $137
                                                                                    EXIT TST
10638  006716
  (5)  006716  012767  000001  172234           MOV     #1,$TIMES
  (3)  006724  000552                           BR      TST33           ;;;EXIT THIS TEST
                                                                                ENDIF
10639  006726
  (4)  006726                           $137:
                                                        ; THIS TEST IS 'BREAK OR HALT' SINSATIVE.
                                                                                IF #APTENV SETIN $ENV THEN
10641  006726
  (6)  006726  032767  000001  172260           BIT     #APTENV,$ENV
  (9)  006734  001404                           BEQ     $140
                                                                                    EXIT TEST
10642  006736
  (5)  006736  012767  000001  172214           MOV     #1,$TIMES
  (3)  006744  000542                           BR      TST33           ;;;EXIT THIS TEST
                                                                                ENDIF
10643  006746
  (4)  006746                           $140:
10644
10645  006746                                                                    LET OLD := #-1
  (4)  006746  012767  177777  000272           MOV     #-1,OLD
10646  006754                                                                    LET OLD+2 := #-1
  (4)  006754  012767  177777  000266           MOV     #-1,OLD+2
10647  006762                                                                    LET @TCSR := @TCSR SET.BY #MAINT
  (7)  006762  052777  000004  172274           BIS     #MAINT,@TCSR
10648                                                           ;EACH BAUD RATE
10649  006770                                                   INCR R3 FROM #0 TO #15. BY #1
  (4)  006770  005003                           CLR     R3
  (5)  006772  000401                           BR      $141
  (4)  006774                           $142:
  (8)  006774  005203                           INC     R3
  (5)  006776                           $141:
  (5)  006776  020327  000017                   CMP     R3,#15.
  (7)  007002  003060                           BGT     $143
10650  007004                                                                    LET R0 := @RBUF
  (4)  007004  017700  172252                   MOV     @RBUF,R0
10651                                                           ;CHANGE BAUDE RATE
10652  007010                                                   LET @TCSRHI :B= RATES(R3)
  (4)  007010  116377  007170  172250           MOVB    RATES(R3),@TCSRHI
10653                                                           ;FLAG
10654  007016                                                   LET BIT := #0
  (4)  007016  005002                           CLR     BIT
10655                                                           ;OUTPUT THE CHARACTER
10656  007020                                                   LET @TBUF := #0
```

```
  (4)  007020  005077  172244          CLR     @TBUF
10657
10658  007024
  (4)  007024  005067  000212          CLR     NEW
10659  007030
  (4)  007030  005067  000210          CLR     NEW+2
10660  007034
  (4)  007034                  $144:
  (6)  007034  005702          TST     BIT
  (9)  007036  001014          BNE     $145
10661  007040
  (6)  007040  032777  000200  172212  BIT     #RCVRDONE,@RCSR
  (9)  007046  001403          BEQ     $146
10662
10663  007050
  (4)  007050  012702  000001          MOV     #1,BIT
10664  007054
  (4)  007054  000404          BR      $147
  (3)  007056                  $146:
10665
10666  007056
  (7)  007056  005267  000160          INC     NEW
10667  007062
  (7)  007062  005567  000156          ADC     NEW+2
10668  007066
  (4)  007066                  $147:
10669
10670  007066
  (4)  007066  000762          BR      $144
  (3)  007070                  $145:
10671
10672  007070
  (6)  007070  026767  000150  000152  CMP     NEW+2,OLD+2
  (9)  007076  103001          BHIS    $150
10673
10674  007100
  (4)  007100  000412          BR      $151
  (3)  007102                  $150:
10675
10676  007102
  (6)  007102  026767  000136  000140  CMP     NEW+2,OLD+2
  (9)  007110  001005          BNE     $152
  (6)  007112  026767  000124  000126  CMP     NEW,OLD
  (9)  007120  103001          BHIS    $152
10677
10678  007122
  (4)  007122  000401          BR      $153
  (3)  007124                  $152:
10679
10680
10681
10682
10683  007124
  (1)  007124  104126          ERROR   126
10684  007126
  (4)  007126                  $153:
```

```
    ;INITIALIZE COUNTER
LET NEW := #0

LET NEW+2 := #0

WHILE BIT EQ #0 DO


    IF #RCVRDONE SETIN @RCSR THEN

    ;DONE - ITS READY
        LET BIT := #1

    ELSE

    ;OTHERWISE-INCREMENT TIME
        LET NEW := NEW + #1

        LET NEW+2 := NEW+2 + CARRY

    ENDIF

    ;SIGNALS DONE
ENDDO


IF NEW+2 LO OLD+2 THEN

    ; OK
ELSE

    ; NEW+2 >= OLD+2
    IF NEW+2 EQ OLD+2 AND NEW LO OLD THEN


        ;OK
    ELSE

        ;NEW+2 > OLD+2 OR
        ;(NEW+2 = OLD+2 AND
        ; NEW >= OLD)
        ;BAUD RATE DIDN'T CHANGE
        ERRHRD 126, BAUDRATE

    ENDIF
```

```
10685  007126                                                          ENDIF
  (4)  007126                          $151:
10686
10687  007126                                                              ;UPDATE OLD TIME
  (4)  007126  016767  000110  000112          MOV     NEW,OLD          LET OLD := NEW
10688  007134                                                          LET OLD+2 := NEW+2
  (4)  007134  016767  000104  000106          MOV     NEW+2,OLD+2
10689
10690  007142                                                      ENDINC  ;BAUD RATE
  (4)  007142  000714                          BR      $142
  (3)  007144                          $143:
10691  007144                                                  LET R3 :B= $USWR+1 AND #17      ; PUT BAUD BACK
  (4)  007144  116703  172051                  MOVB    $USWR+1,R3
  (7)  007150  110346                          MOVB    R3,-(SP)
  (7)  007152  142716  000017                  BICB    #17,(SP)
  (7)  007156  142603                          BICB    (SP)+,R3
10692  007160                                                  LET @TCSRHI :B= RATES(R3)       ; LIKE HE WANTED IT
  (4)  007160  116377  007170  172100          MOVB    RATES(R3),@TCSRHI
10693
10694  007166                                                      EXIT    ;SKIP TABLE
  (3)  007166  000431                          BR      TST33           ;;;EXIT THIS TEST
10695
10696  007170                               RATES:  ;A TABLE OF THE ACTUAL BYTES TO MOVE INTO THE
10697                                       ;UPPER BYTE OF XCSR FOR EACH BAUD RATE
10698                                       ;** NOTE::      THE VALUE INDICATED IN THE COLUMN 'OFFSET
10699                                       ;**             INTO TABLE' CAN BE PLACED INTO BITS<11:8>
10700                                       ;**             OF LOCATION '$USWR' TO CAUSE THE CORROSPONDING
10701                                       ;**             BAUD TO BE SELECTED IN THE DLV11-E UPON
10702                                       ;**             COMPLETION OF THIS TEST.
10703
10704                                                       ;       BAUD    OFFSET INTO TABLE
10705  007170       010                     R0050:  .BYTE   010     ;       50      0
10706  007171       030                     R0070:  .BYTE   030     ;       70      1
10707  007172       050                     R0110:  .BYTE   050     ;       110     2
10708  007173       070                     R0135:  .BYTE   070     ;       135     3
10709  007174       110                     R0150:  .BYTE   110     ;       150     4
10710  007175       130                     R0300:  .BYTE   130     ;       300     5
10711  007176       150                     R0600:  .BYTE   150     ;       600     6
10712  007177       170                     R0200:  .BYTE   170     ;       1200    7
10713  007200       210                     R1800:  .BYTE   210     ;       1800    10
10714  007201       230                     R2000:  .BYTE   230     ;       2000    11
10715  007202       250                     R2400:  .BYTE   250     ;       2400    12
10716  007203       270                     R3600:  .BYTE   270     ;       3600    13
10717  007204       310                     R4800:  .BYTE   310     ;       4800    14
10718  007205       330                     R7200:  .BYTE   330     ;       7200    15
10719  007206       350                     R9600:  .BYTE   350     ;       9600    16
10720  007207       370                     R10000: .BYTE   370     ;       19200   17
10721
10722  007210  040502  042125  051040  BAUDRATE:       .ASCIZ  /BAUD RATE DIDN'T CHANGE./
       007216  052101  020105  044504
       007224  047104  052047  041440
       007232  040510  043516  027105
       007240     000
10723          007242                  .EVEN
10724  007242  000000  000000          NEW:    0,0
10725  007246  000000  000000          OLD:    0,0
```

```
 10726  007252                                                  ENDTST
 10727
 10728
 10729
```

```
10731
10736                                   ;;*****************************************************************
10744                                   ;;*****************************************************************
  (3)                                   ;*TEST 33       TRANSMITTER INTERRUPT LOGIC TEST
  (4)                                   ;*              LOGICALLLY THIS IS 4 SEPARATE TESTS
  (4)                                   ;*              A) DOES TRANSMITTER INTERRUPT LOGIC WORK
  (4)                                   ;*              B) AT PRIORITY OF 0
  (4)                                   ;*              C) AND ONLY ONCE
  (4)                                   ;*              D) BUT NOT WITH INTERRUPT ENABLE CLEAR
  (3)                                   ;;*****************************************************************
  (2)  007252  000004                   TST33:  SCOPE
  (2)
  (1)  007254  012767  000010  171676           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2)  007262  012767  000033  171710           MOV     #33,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
10749                                                                           ;CLEAR 'INTERRUPT OCCURED' FLAG
10750  007270                                                           LET INTFLAG := #0
  (4)  007270  005067  002540                   CLR     INTFLAG
10751
10752                                                                           ;GET VECTOR ADDRESS
10753  007274                                                           LET R3 := DLVEC
  (4)  007274  016703  171756                   MOV     DLVEC,R3
10754                                                                           ;FOR THE TRANSMITTER
10755  007300                                                           LET R3 := R3 + #4
  (7)  007300  062703  000004                   ADD     #4,R3
10756                                                                           ;SET VECTOR TO POINT TO TRANS.SRV AT PRI
10757  007304                                                           SETVEC R3, #INTSRV, #PR6
  (5)  007304  010146                           MOV     R1,-(SP)
  (5)  007306  010301                           MOV     R3,R1
  (5)  007310  012721  012026                   MOV     #INTSRV,(R1)+
  (5)  007314  012711  000300                   MOV     #PR6,(R1)
  (5)  007320  012601                           MOV     (SP)+,R1
10758  007322                                                           BGNSUB
  (5)  007322  012767  007330  171560           MOV     #64$,$LPERR
10759
10760  007330                                                                   ;CLEAR INTERRUPT ENABLE
  (7)  007330  042777  000100  171726           BIC     #XMITIE,@TCSR   LET @TCSR := @TCSR CLR.BY #XMITIE
10761
10766                                                                           ;SET IT TO 0
10767  007336  012746  000000                   MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
  (1)  007342  012746  007350                   MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
  (1)  007346  000002                           RTI                     ;;POP NEW PC AND PS
  (1)  007350                           65$:
10772
10773                                                                           ;NOW SET I.E. BIT
10774  007350                                                           LET @TCSR := @TCSR SET.BY #XMITIE
  (7)  007350  052777  000100  171706           BIS     #XMITIE,@TCSR
10775
10776                                                                           ;LET INTERRUPT HAVE TIME TO OCCUR
10777  007356                                                           WAITMS 200.
  (4)  007356  010546                           MOV     R5,-(SP)
  (5)  007360  012745  000310                   MOV     #200.,-(R5)
  (4)  007364  004767  002356                   JSR     PC,WAIT
  (4)  007370  012605                           MOV     (SP)+,R5
10778
10779                                                                           ;DID EXACTLY 1 INTERRUPT OCCUR
10780  007372                                                           IF INTFLAG NE #1 THEN
```

```
   (6)  007372  026727  002436  000001          CMP     INTFLAG,#1
   (9)  007400  001406                          BEQ     $154
10781                                                                   ;NO - WAS IT 0 OR MORE THAN ONCE
10782  007402                                                           IF INTFLAG EQ #0 THEN
   (6)  007402  005767  002426                  TST     INTFLAG
   (9)  007406  001002                          BNE     $155
10783                                                                   ;TRANSMITTER DID NOT INTERRUPT IN TIME
10784  007410                                                               ERRHRD 106,,DIDNOT
   (1)  007410  104106                          ERROR   106
10785  007412                                                           ELSE
   (4)  007412  000401                          BR      $156
   (3)  007414                          $155:
10786                                                                   ;TWICE
10787                                                                   ;TRANSMITTER INTERRUPTED TWICE
10788  007414                                                               ERRHRD 107,,TWICE
   (1)  007414  104107                          ERROR   107
10789  007416                                                           ENDIF
   (4)  007416                          $156:
10790  007416                                                       ENDIF
   (4)  007416                          $154:
10791  007416                                                   ENDSUB
10792                                                       ;INTERRUPT WITHOUT INTERRUPT ENABLE SET
10793  007416                                                   BGNSUB
   (5)  007416  012767  007424  171464          MOV     #64$,$LPERR
10794                                                                   ;CLEAR 'INTERRUPT OCCURED' FLAG
10795  007424                                                       LET INTFLAG := #0
   (4)  007424  005067  002404                  CLR     INTFLAG
10796                                                                   ;CLEAR INTERRUPT ENABLE
10797  007430                                                       LET @TCSR := @TCSR CLR.BY #XMITIE
   (7)  007430  042777  000100  171626          BIC     #XMITIE,@TCSR
10798                                                                   ;NO INTERRUPTS SHOULD OCCUR.
10803  007436  012746  000000                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
   (1)  007442  012746  007450                  MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
   (1)  007446  000002                          RTI                     ;;POP NEW PC AND PS
   (1)  007450                          65$:
10808                                                                   ;DARE IT TO HAPPEN
10809  007450                                                       WAITMS 2
   (4)  007450  010546                          MOV     R5,-(SP)
   (5)  007452  012745  000002                  MOV     #2,-(R5)
   (4)  007456  004767  002264                  JSR     PC,WAIT
   (4)  007462  012605                          MOV     (SP)+,R5
10810  007464                                                       IF INTFLAG NE #0 THEN
   (6)  007464  005767  002344                  TST     INTFLAG
   (9)  007470  001401                          BEQ     $157
10811                                                                   ;INTERRUPT OCCURED WITH I E CLEARED
10812  007472                                                           ERRHRD 110,NOTENAB
   (1)  007472  104110                          ERROR   110
10813  007474                                                       ENDIF
   (4)  007474                          $157:
10814  007474                                                       BRESET
   (1)  007474  000005                          RESET
10815  007476                                                   ENDSUB
10816                                                                   ;RESTORE VECTOR AREA
10817  007476                                                   CLRVEC R3
   (3)  007476  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
   (3)  007500  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
```

```
   (5)  007502  012701  000003          MOV     #R3,R1
   (5)  007506  010102                  MOV     R1,R2
   (8)  007510  062702  000002          ADD     #2,R2
   (5)  007514  010221                  MOV     R2,(R1)+
   (5)  007516  005011                  CLR     (R1)
   (3)  007520  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
   (3)  007522  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
10818
10819  007524                                                    ENDTST
10820
10821
10822
10823
10824
10825
```

```
10827
10832                                   ;;*****************************************************************
10838                                   ;;*****************************************************************
  (3)                                   ;*TEST 34       RECEIVER INTERRUPT LOGIC TEST
  (4)                                   ;*             THIS TEST COVERS ALL OF THE RECEIVER
  (4)                                   ;*             SIDE OF THE INTERRUPT LOGIC, BOTH DATASET
  (4)                                   ;*             AND CHARACTER MODES.
  (3)                                   ;;*****************************************************************
  (2) 007524  000004                    TST34:  SCOPE
  (2)
  (1) 007526  012767  000010  171424            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 007534  012767  000034  171436            MOV     #34,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
10843                                                                           ;CLEAR INTERRUPT OCCURED FLAG
10844                                                                           ;SET UP RECEIVER INTER.VECTOR
10845 007542                                                                SETVEC DLVEC,#INTSRV,#PR6
  (5) 007542  010146                    MOV     R1,-(SP)
  (5) 007544  016701  171506            MOV     DLVEC,R1
  (5) 007550  012721  012026            MOV     #INTSRV,(R1)+
  (5) 007554  012711  000300            MOV     #PR6,(R1)
  (5) 007560  012601                    MOV     (SP)+,R1
10846                                           ;PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-RCVRIE
10847 007562                                                                BGNSUB
  (5) 007562  012767  007570  171320            MOV     #64$,$LPERR
10848 007570                                                                LET INTFLAG := #0
  (4) 007570  005067  002240            CLR     INTFLAG
10849                                                                           ;SET MAINT. BIT
10850 007574                                                                LET @TCSR := @TCSR SET.BY #MAINT
  (7) 007574  052777  000004  171462            BIS     #MAINT,@TCSR
10851                                                                           ;CLEAR INTERRUPTS
10852 007602                                                                    LET @RCSR := @RCSR CLR.BY #RCVRIE
  (7) 007602  042777  000100  171450            BIC     #RCVRIE,@RCSR
10853                                                                           ;CHANGE PRIORITY
10854                                                                           ;..TO 0
10859 007610  012746  000000                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
  (1) 007614  012746  007622                  MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
  (1) 007620  000002                          RTI                     ;;POP NEW PC AND PS
  (1) 007622                        65$:
10864
10865                                                                           ;SEND A CHARACTER
10866 007622                                                                LET @TBUF :B= #0
  (4) 007622  105077  171442            CLRB    @TBUF
10867                                                                           ;WAIT A MAXIMUM
10868                                                                       ;OF 500 MSEC FOR
10869                                                                       ;RCVR RDY TO SET IN RCSR
10870 007626                                                                CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
  (3) 007626  010546                    MOV     R5,-(SP)
  (7) 007630  012745  177777            MOV     #SET,-(R5)
  (6) 007634  016745  171420            MOV     RCSR,-(R5)
  (5) 007640  012745  000200            MOV     #RCVRDONE,-(R5)
  (4) 007644  012745  000500            MOV     #500,-(R5)
  (3) 007650  004767  001614            JSR     PC,TIMER
  (3) 007654  012605                    MOV     (SP)+,R5
10871                                                                           ;SET INTERRUPT ENABLE
10872 007656                                                                LET @RCSR := @RCSR SET.BY #RCVRIE
  (7) 007656  052777  000100  171374            BIS     #RCVRIE,@RCSR
10873                                                                           ;LET IT COME IN.
```

```
10874  007664
  (4)  007664  010546                        MOV     R5,-(SP)
  (5)  007666  012745  000001                MOV     #1,-(R5)
  (4)  007672  004767  002050                JSR     PC,WAIT
  (4)  007676  012605                        MOV     (SP)+,R5
10875  007700
  (4)  007700  017700  171356                MOV     @RBUF,R0
10876
10877
10878  007704
  (6)  007704  026727  002124  000001        CMP     INTFLAG,#1
  (9)  007712  001406                        BEQ     $160
10879
10880  007714
  (6)  007714  005767  002114                TST     INTFLAG
  (9)  007720  001002                        BNE     $161
10881
10882  007722
  (1)  007722  104111                        ERROR   111
10883
10884  007724
  (4)  007724  000401                        BR      $162
  (3)  007726                        $161:
10885
10886  007726
  (1)  007726  104112                        ERROR   112
10887  007730
  (4)  007730                        $162:
10888  007730
  (4)  007730                        $160:
10889
10890  007730
  (7)  007730  042777  000004  171326        BIC     #MAINT,@TCSR
10891
10892  007736
  (7)  007736  042777  000100  171314        BIC     #RCVRIE,@RCSR
10893  007744
10894
10895
10896
10897
10898
10899
10900
10901                                        ;PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-DATAIE
10902  007744
  (5)  007744  012767  007752  171136        MOV     #64$,$LPERR
10903  007752
  (6)  007752  032767  020000  171240        BIT     #CABLE,$USWR
  (9)  007760  001004                        BNE     $163
10904
10905  007762
  (5)  007762  012767  000001  171170        MOV     #1,$TIMES
  (3)  007770  000466                        BR      TST35
10906  007772
  (4)  007772                        $163:

                                             WAITMS 1



                                             LET R0 := @RBUF ; CLEAR RCVRDONE

                                                 ;DID HE DO IT RIGHT?
                                             IF INTFLAG NE #1 THEN

                                                 ;NONE OCCURED
                                                 IF INTFLAG EQ #0 THEN

                                                 ;RECEIVER DID NOT INTERRUPT IN TIME
                                                     ERRHRD 111,,DIDNOT

                                                 ;TWICE OR MORE
                                                 ELSE

                                                 ;RECEIVER INTERRUPTED TWICE
                                                     ERRHRD 112,,TWICE

                                                 ENDIF

                                             ENDIF

                                                     ;RESET MAINT. BIT.
                                             LET @TCSR := @TCSR CLR.BY #MAINT
                                                     ; CLEAR INTERRUPT ENABLE
                                             LET @RCSR := @RCSR CLR.BY #RCVRIE
                                             ENDSUB



                                             BGNSUB
                                             IF #CABLE NOTSETIN $USWR THEN

                                                     ;CAN'T TEST WITHOUT A CABLE
                                                 EXIT TST
                                     ;;;EXIT THIS TEST
                                             ENDIF
```

```
10907                                                                   ; CLEAR 'INTFLAG'
10908  007772                                                         LET INTFLAG := #0
  (4)  007772  005067  002036          CLR     INTFLAG
10909                                                                   ;CLEAR INTERRUPTS
10910  007776                                                       LET @RCSR := @RCSR CLR.BY #DATAIE
  (7)  007776  042777  000040  171254  BIC     #DATAIE,@RCSR
10911                                                                   ;CHANGE PRIORITY
10912                                                                   ;..TO 0
10917  010004  012746  000000          MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
  (1)  010010  012746  010016          MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
  (1)  010014  000002                  RTI                     ;;POP NEW PC AND PS
  (1)  010016                  64$:
10922  010016                                                       LET @RCSR := @RCSR CLR.BY #REQSEND
  (7)  010016  042777  000004  171234  BIC     #REQSEND,@RCSR
10923                                                                   ;SET INTERRUPT ENABLE
10924  010024                                                       LET @RCSR := @RCSR SET.BY #DATAIE
  (7)  010024  052777  000040  171226  BIS     #DATAIE,@RCSR
10925  010032                                                       LET    @RCSR := @RCSR SET.BY #REQSEND
  (7)  010032  052777  000004  171220  BIS     #REQSEND,@RCSR
10926                                                                   ;LET IT COME IN.
10927  010040                                                       WAITMS 1
  (4)  010040  010546                  MOV     R5,-(SP)
  (5)  010042  012745  000001          MOV     #1,-(R5)
  (4)  010046  004767  001674          JSR     PC,WAIT
  (4)  010052  012605                  MOV     (SP)+,R5
10928
10929                                                                   ; DID IT DO IT RIGHT?
10930  010054                                                       IF INTFLAG NE #1 THEN
  (6)  010054  026727  001754  000001  CMP     INTFLAG,#1
  (9)  010062  001406                  BEQ     $164
10931                                                                   ;NONE OCCURED
10932  010064                                                           IF INTFLAG EQ #0 THEN
  (6)  010064  005767  001744          TST     INTFLAG
  (9)  010070  001002                  BNE     $165
10933                                                                   ;DATAINT  DID NOT INTERRUPT IN TIME
10934  010072                                                               ERRHRD 113,,DIDNOT
  (1)  010072  104113                  ERROR   113
10935                                                                   ;TWICE OR MORE
10936  010074                                                           ELSE
  (4)  010074  000401                  BR      $166
  (3)  010076                  $165:
10937                                                                   ; DATAINT INTERRUPTED TWICE
10938  010076                                                               ERRHRD 114,,TWICE
  (1)  010076  104114                  ERROR   114
10939  010100                                                           ENDIF
  (4)  010100                  $166:
10940  010100                                                       ENDIF
  (4)  010100                  $164:
10941  010100                                                       LET @RCSR := @RCSR CLR.BY #DATAIE
  (7)  010100  042777  000040  171152  BIC     #DATAIE,@RCSR
10942  010106                                                       LET @RCSR := @RCSR CLR.BY #REQSEND
  (7)  010106  042777  000004  171144  BIC     #REQSEND,@RCSR
10943  010114                                                   ENDSUB
10944
10945  010114                                                   LET R4 := @DLVEC
  (4)  010114  017704  171136          MOV     @DLVEC,R4
```

```
10946  010120                                                    CLRVEC R4
  (3)  010120  010146                    MOV     R1,-(SP)       ;;PUSH R1 ON STACK
  (3)  010122  010246                    MOV     R2,-(SP)       ;;PUSH R2 ON STACK
  (5)  010124  012701  000004            MOV     #R4,R1
  (5)  010130  010102                    MOV     R1,R2
  (8)  010132  062702  000002            ADD     #2,R2
  (5)  010136  010221                    MOV     R2,(R1)+
  (5)  010140  005011                    CLR     (R1)
  (3)  010142  012602                    MOV     (SP)+,R2       ;;POP STACK INTO R2
  (3)  010144  012601                    MOV     (SP)+,R1       ;;POP STACK INTO R1
10947  010146                                                    ENDTST
```

```
10949
10954                                   ;;*****************************************************************
10958                                   ;;*****************************************************************
  (3)                                   ;*TEST 35       TEST ACTUAL DATA TRANSFERED
  (4)                                   ;*              NON-INTERRUPT MAINTENANCE BIT SET
  (3)                                   ;;*****************************************************************
  (2)  010146  000004                   TST35:  SCOPE
  (2)
  (1)  010150  012767  000001  171002           MOV     #1,$TIMES       ;;DO 1 ITERATION
  (2)  010156  012767  000035  171014           MOV     #35,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
10963                                                                           ;SET MAINT. BIT
10964  010164                                                           LET @TCSR := @TCSR SET.BY #MAINT
  (7)  010164  052777  000004  171072           BIS     #MAINT,@TCSR
10965
10966                                                                           ;CHANGE PRIORITY
10967                                                                           ;..TO 0
10972  010172  012746  000000                MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
  (1)  010176  012746  010204                MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
  (1)  010202  000002                        RTI                     ;;POP NEW PC AND PS
  (1)  010204                           64$:
10977                                                                           ;GET DATA MASK.
10978  010204                                                           CALL DATLNG OUT <R1>
  (4)  010204  162705  000002                SUB     #1*2,R5
  (3)  010210  004767  001432                JSR     PC,DATLNG
  (4)  010214  012501                        MOV     (R5)+,R1
10979                                                                           ; START CLEAN
10980  010216                                                           LET R0 := @RBUF
  (4)  010216  017700  171040                MOV     @RBUF,R0
10981
10982                                                                           ;ALL BINARY CHAR.
10983  010222                                                           INCR R2 FROM #0 TO #377 BY #1
  (4)  010222  005002                        CLR     R2
  (5)  010224  000401                        BR      $167
  (4)  010226                           $170:
  (8)  010226  005202                        INC     R2
  (5)  010230                           $167:
  (5)  010230  020227  000377                CMP     R2,#377
  (7)  010234  003047                        BGT     $171
10984
10985
10986                                                                   ;TRANSMIT CHAR IN R2
10987
10988  010236                                                           CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
  (3)  010236  010546                        MOV     R5,-(SP)
  (7)  010240  012745  177777                MOV     #SET,-(R5)
  (6)  010244  016745  171014                MOV     TCSR,-(R5)
  (5)  010250  012745  000200                MOV     #XMITRDY,-(R5)
  (4)  010254  012745  000500                MOV     #500,-(R5)
  (3)  010260  004767  001204                JSR     PC,TIMER
  (3)  010264  012605                        MOV     (SP)+,R5
10989
10990                                                                           ;TRANSMIT IT
10991  010266                                                           LET @TBUF :B= R2
  (4)  010266  110277  170776                MOVB    R2,@TBUF
10992
10993  010272                                                           CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
```

```
   (3)  010272  010546                          MOV     R5,-(SP)
   (7)  010274  012745  177777                  MOV     #SET,-(R5)
   (6)  010300  016745  170754                  MOV     RCSR,-(R5)
   (5)  010304  012745  000200                  MOV     #RCVRDONE,-(R5)
   (4)  010310  012745  000500                  MOV     #500,-(R5)
   (3)  010314  004767  001150                  JSR     PC,TIMER
   (3)  010320  012605                          MOV     (SP)+,R5
10994
10995  010322                                                    ;AND SAVE IT
   (4)  010322  017703  170734                  MOV     @RBUF,R3 LET R3 := @RBUF
10996
10997
10998                                                     ;COMPARE TO SEE IF WE RECEIVED IT ALL
10999
11000                                                         ;CLEAN OFF NON-DATA BITS
11001                                                         ;ON BOTH TRANSMITTED AND
11002  010326                                         LET R4 := R2 CLR.BY R1
   (4)  010326  010204                          MOV     R2,R4
   (7)  010330  040104                          BIC     R1,R4
11003  010332                                     LET R3 := R3 CLR.BY R1
   (7)  010332  040103                          BIC     R1,R3
11004
11005                                                     ;RECEIVED DATA
11006  010334                                     IF R4 NE R3 THEN
   (6)  010334  020403                          CMP     R4,R3
   (9)  010336  001405                          BEQ     $172
11007
11008  010340                                                    ;DATA COMPARE ERROR
   (1)  010340  104116                          ERROR   116      ERRHRD 116,COMP,SBWAS
11009  010342                                                    EXIT TST  ; ON ERROR
   (5)  010342  012767  000001  170610          MOV     #1,$TIMES
   (3)  010350  000404                          BR      TST36    ;;;EXIT THIS TEST
11010  010352                                                 ENDIF
   (4)  010352                          $172:
11011  010352                                              ENDINC  ; R2
   (4)  010352  000725                          BR      $170
   (3)  010354                          $171:
11012
11013                                                           ;RESET MAINT. BIT.
11014  010354                                          LET @TCSR := @TCSR CLR.BY #MAINT
   (7)  010354  042777  000004  170702          BIC     #MAINT,@TCSR
11015  010362                                       ENDTST
11016
11017
11018
```

```
11020
11025                                   ;;*****************************************************************
11026                                   ;;*****************************************************************
  (3)                                   ;*TEST 36       TEST DATA THROUGH CABLE
  (3)                                   ;;*****************************************************************
  (2) 010362  000004                    TST36:  SCOPE
  (2)
  (1) 010364  012767  000001  170566            MOV     #1,$TIMES       ;;DO 1 ITERATION
  (2) 010372  012767  000036  170600            MOV     #36,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                                IF #CABLE NOTSETIN $USWR THEN
11031 010400
  (6) 010400  032767  020000  170612            BIT     #CABLE,$USWR
  (9) 010406  001004                            BNE     $173
                                                                                        ;CAN'T TEST WITHOUT A CABLE
11032                                                                           EXIT TST
11033 010410
  (5) 010410  012767  000001  170542            MOV     #1,$TIMES
  (3) 010416  000474                            BR      TST37           ;;;EXIT THIS TEST
                                                                                ENDIF
11034 010420
  (4) 010420                            $173:
                                                                                        ;DON'T USE MAINT.
11035                                                                           LET @TCSR := @TCSR CLR.BY #MAINT
11036 010420
  (7) 010420  042777  000004  170636            BIC     #MAINT,@TCSR
                                                                                        ;CHANGE PRIORITY
11037                                                                                   ;...TO 0
11038
11043 010426  012746  000000                    MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
  (1) 010432  012746  010440                    MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
  (1) 010436  000002                            RTI                     ;;POP NEW PC AND PS
  (1) 010440                            64$:
                                                                                ;GET DATA MASK
11048                                                                   CALL DATLNG OUT <R1>
11049 010440
  (4) 010440  162705  000002                    SUB     #1*2,R5
  (3) 010444  004767  001176                    JSR     PC,DATLNG
  (4) 010450  012501                            MOV     (R5)+,R1
11050 010452                                                            LET R0 := @RBUF ; START CLEAN
  (4) 010452  017700  170604                    MOV     @RBUF,R0
                                                                                ;BINARY COUNT PATTERN
11051                                                                   INCR R2 FROM #0 TO #377 BY #1
11052 010456
  (4) 010456  005002                            CLR     R2
  (5) 010460  000401                            BR      $174
  (4) 010462                            $175:
  (8) 010462  005202                            INC     R2
  (5) 010464                            $174:
  (5) 010464  020227  000377                    CMP     R2,#377
  (7) 010470  003047                            BGT     $176
11053
11054                                                                   ;TRANSMIT THE CHAR. IN R2.
11055
11056                                                                   CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
11057 010472
  (3) 010472  010546                            MOV     R5,-(SP)
  (7) 010474  012745  177777                    MOV     #SET,-(R5)
  (6) 010500  016745  170560                    MOV     TCSR,-(R5)
  (5) 010504  012745  000200                    MOV     #XMITRDY,-(R5)
  (4) 010510  012745  000500                    MOV     #500,-(R5)
  (3) 010514  004767  000750                    JSR     PC,TIMER
  (3) 010520  012605                            MOV     (SP)+,R5
11058
```

```
11059
11060  010522
  (4)  010522  110277  170542                MOVB    R2,@TBUF
11061  010526
  (3)  010526  010546                        MOV     R5,-(SP)
  (7)  010530  012745  177777                MOV     #SET,-(R5)
  (6)  010534  016745  170520                MOV     RCSR,-(R5)
  (5)  010540  012745  000200                MOV     #RCVRDONE,-(R5)
  (4)  010544  012745  000500                MOV     #500,-(R5)
  (3)  010550  004767  000714                JSR     PC,TIMER
  (3)  010554  012605                        MOV     (SP)+,R5
11062
11063
11064  010556
  (4)  010556  017703  170500                MOV     @RBUF,R3
11065
11066
11067  010562
  (4)  010562  010204                        MOV     R2,R4
  (7)  010564  040104                        BIC     R1,R4
11068  010566
  (7)  010566  040103                        BIC     R1,R3
11069
11070
11071  010570
  (6)  010570  020403                        CMP     R4,R3
  (9)  010572  001405                        BEQ     $177
11072
11073  010574
  (1)  010574  104117                        ERROR   117
11074  010576
  (5)  010576  012767  000001  170354        MOV     #1,$TIMES
  (3)  010604  000401                        BR      TST37
11075  010606
  (4)  010606                        $177:
11076
11077  010606
  (4)  010606  000725                        BR      $175
  (3)  010610                        $176:
11078
11079
11080
11081  010610
11082
11083
11084
11085

                    ;START IT ON ITS WAY
LET @TBUF :B= R2

CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>

                    ;RETRIEVE
LET R3 := @RBUF

                    ;STRIP OFF JUNK ON BOTH
LET R4 := R2 CLR.BY R1

LET R3 := R3 CLR.BY R1

                    ;WE HAVE TROUBLE
IF R4 NE R3 THEN

                    ;DATA COMPARE ERROR
                    ERRHRD 117,COMP,SBWAS

                    EXIT TST  ; ON ERROR

;;;EXIT THIS TEST
                    ENDIF

                ENDINC  ; R2

ENDTST
```

```
11087
11092                                   ;;*****************************************************************
11096                                   ;;*****************************************************************
  (3)                                   ;*TEST 37        FULL DATA TRANSFER WITH INTERRUPTS
  (4)                                   ;*               AND MAINTENANCE MODE.
  (3)                                   ;;*****************************************************************
  (2)  010610  000004                   TST37:  SCOPE
  (2)
  (1)  010612  012767  000001  170340           MOV     #1,$TIMES       ;;DO 1 ITERATION
  (2)  010620  012767  000037  170352           MOV     #37,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
11101
11102
11103                                                   ; THIS TEST IS 'BREAK OR HALT' SINSATIVE.
11104  010626                                                           IF #APTENV SETIN $ENV THEN
  (6)  010626  032767  000001  170360           BIT     #APTENV,$ENV
  (9)  010634  001404                           BEQ     $200
11105  010636                                                                   EXIT TEST
  (5)  010636  012767  000001  170314           MOV     #1,$TIMES
  (3)  010644  000550                           BR      TST40           ;;;EXIT THIS TEST
11106  010646                                                           ENDIF
  (4)  010646                           $200:
11107                                                                            ;GET DATA MASK
11108  010646                                                                    CALL DATLNG OUT <R3>
  (4)  010646  162705  000002                   SUB     #1*2,R5
  (3)  010652  004767  000770                   JSR     PC,DATLNG
  (4)  010656  012503                           MOV     (R5)+,R3
11109
11110
11111                                           ; THIS TEST WILL RUN BOTH TRANSMITTER AND
11112                                           ; RECIEVER AT FULL SPEED TESTING
11113                                           ; THE ABILITY OF THE MODULE
11114                                           ; TO HANDLE INTERRUPTS FROM BOTH SIDES
11115                                           ; AT ONCE. ALSO, THE DOUBLE BUFFERING LOGIC
11116                                           ; OF THE UART WILL BE FULLY TESTED.
11117                                           ;  THIS TEST WILL TRANSFER A MAXIMUM OF 400(8)
11118                                           ; CHARACTERS THROUGH THE MODULE, BUT IF AN ERROR
11119                                           ; IS DETECTED BY THE TEST A PREMATURE SHUTDOWN OCCURS.
11120
11121
11122                                                                            ;CHANGE PRIORITY
11123                                                                            ;..TO 0
11128  010660  012746  000000                           MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
  (1)  010664  012746  010672                           MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
  (1)  010670  000002                                   RTI                     ;;POP NEW PC AND PS
  (1)  010672                           64$:
11133                                                                            ;GET VECTOR ADDRESS
11134  010672                                                                  LET R1 := DLVEC
  (4)  010672  016701  170360                   MOV     DLVEC,R1
11135                                                                            ;RCVR VECTOR
11136  010676                                                                  LET (R1)+ := #REC
  (4)  010676  012721  011072                   MOV     #REC,(R1)+
11137  010702                                                                  LET (R1)+ := #PR6
  (4)  010702  012721  000300                   MOV     #PR6,(R1)+
11138                                                                            ;POINT TO TRANSMITTER VECTOR
11139                                                                           ;AND SET IT UP ALSO
11140  010706                                                                  LET (R1)+ := #TRAN
```

```
  (4)  010706  012721  011030              MOV     #TRAN,(R1)+
11141  010712                                                    LET (R1)  := #PR6
  (4)  010712  012711  000300              MOV     #PR6,(R1)
11142
11143                                                                   ; INITIALIZE COUNTERS
11144  010716                                                    LET R1 := #-1
  (4)  010716  012701  177777              MOV     #-1,R1
11145                                                                   ;RECEIVER STORAGE
11146  010722                                                    LET R2 := #0
  (4)  010722  005002                      CLR     R2
11147                                                                   ;# OF RECEIVED CHAR. COUNT.
11148  010724                                                    LET R4 := #-1
  (4)  010724  012704  177777              MOV     #-1,R4
11149
11150                                                                   ; CLEAR ERROR COUNT.
11151  010730                                                    LET ERRCNT := #0
  (4)  010730  005067  000066              CLR     ERRCNT
11152
11153  010734                                                    BRESET  ;SET UP ALL REGISTERS
  (1)  010734  000005                      RESET
11154                                                                   ;SET UP MAINTENANCE
11155  010736                                                    LET @TCSR := @TCSR SET.BY #MAINT
  (7)  010736  052777  000004  170320      BIS     #MAINT,@TCSR
11156
11157                                                                   ;SET I.E. IN TRANSMITTER
11158  010744                                                    LET @TCSR := @TCSR SET.BY #XMITIE
  (7)  010744  052777  000100  170312      BIS     #XMITIE,@TCSR
11159                                                                   ;AND RECEIVER
11160  010752                                                    LET @RCSR := @RCSR SET.BY #RCVRIE
  (7)  010752  052777  000100  170300      BIS     #RCVRIE,@RCSR
11161
11162
11163                                                           ;NOW WE WAIT UNTIL R4 COUNT (RECEIVED) IS EQUAL
11164  010760                                                   REPEAT
  (3)  010760                       $201:
11165  010760                                                   UNTIL R4 EQ NUMBER OR ERRCNT GT #0
  (4)  010760  020467  000040              CMP     R4,NUMBER
  (6)  010764  001403                      BEQ     $202
  (4)  010766  005767  000030              TST     ERRCNT
  (7)  010772  003772                      BLE     $201
  (4)  010774                       $202:
11166
11167                                                                   ;DATA COMPARE ERRORS.
11168  010774                                                   IF ERRCNT NE #0 THEN
  (6)  010774  005767  000022              TST     ERRCNT
  (9)  011000  001401                      BEQ     $203
11169                                                                   ;DATA COMPARE ERROR
11170  011002                                                       ERRHRD 120,COMP,FIRST
  (1)  011002  104120                      ERROR   120
11171  011004                                                   ENDIF
  (4)  011004                       $203:
11172
11173  011004                                                   LET @TCSR := @TCSR CLR.BY #XMITIE
  (7)  011004  042777  000100  170252      BIC     #XMITIE,@TCSR
11174  011012                                                   LET @RCSR := @RCSR CLR.BY #XMITIE
  (7)  011012  042777  000100  170240      BIC     #XMITIE,@RCSR
```

```
11175
11176  011020                                                   EXIT    ;SKIP OVER SUPPORT ROUTINES & STORAGE
  (3)  011020  000462                BR      TST40          ;;;EXIT THIS TEST
11177
11178  011022  000000                                           ERRCNT: 0
11179  011024  000400                                           NUMBER: 400
11180  011026     000                                           SB:     .BYTE 0
11181  011027     000                                           WAS:    .BYTE 0
11182
11183
11184
11185                          ;*************************************************************
11186                                  ;TRANSMIT INTERRUPT HANDLER
11187
11188  011030                                                   BGNSRV  TRAN
  (1)  011030                  TRAN:
11189                          ;*************************************************************
11190
11191                                                               ;INCREMENT CHAR COUNT
11192  011030                                                     LET R1 := R1 + #1
  (7)  011030  005201                INC     R1
11193                                                              ;SET UP FOR TRANSFER
11194  011032                                                     LET HOLD := R1      CLR.BY R3
  (4)  011032  010167  000030        MOV     R1,HOLD
  (7)  011036  040367  000024        BIC     R3,HOLD
11195                                                              ;AND SEND.
11196  011042                                                     LET @TBUF := HOLD
  (4)  011042  016777  000020  170220  MOV   HOLD,@TBUF
11197                                                              ;ALL DONE
11198  011050                                                     IF R1 EQ NUMBER THEN
  (6)  011050  020167  177750        CMP     R1,NUMBER
  (9)  011054  001003                BNE     $204
11199                                                                 ;STOP   INTERRUPT PROCESSING
11200  011056                                                         LET @TCSR := @TCSR CLR.BY #XMITIE
  (7)  011056  042777  000100  170200  BIC   #XMITIE,@TCSR
11201  011064                                                       ENDIF
  (4)  011064                  $204:
11202
11203  011064  000401                BR      ZZZ                    ; EXIT SRV
11204
11205  011066  000000                                               HOLD:0
11206
11207  011070                       ZZZ:                          ENDSRV
  (1)  011070  000002                RTI
11208
11209
11210                          ;*************************************************************
11211
11212                                  ;RECEIVER INTERRUPT HANDLER
11213  011072                                                     BGNSRV  REC
  (1)  011072                  REC:
11214                          ;*************************************************************
11215
11216                                                              ;COUNT THIS CHAR.
11217  011072                                                     LET R4 := R4 + #1
  (7)  011072  005204                INC     R4
```

```
11218
11219  011074
  (4)  011074  017702  170162          MOV     @RBUF,R2          ;GET CHAR IN + MASK IT
  (7)  011100  040302                  BIC     R3,R2             LET R2 := @RBUF CLR.BY R3
11220
11221  011102
  (4)  011102  010467  000054          MOV     R4,RHLD           ;RHLD WILL CONTAIN EXPECTED INPUT
  (7)  011106  040367  000050          BIC     R3,RHLD           LET RHLD := R4 CLR.BY R3
11222
11223
11224  011112
  (6)  011112  020267  000044          CMP     R2,RHLD           ;DO THEY COMPARE
  (9)  011116  001412                  BEQ     $205              IF R2 NE RHLD THEN
11225
11226  011120
  (6)  011120  005767  177676          TST     ERRCNT                ;FIRST ERROR
  (9)  011124  001005                  BNE     $206                  IF ERRCNT EQ #0 THEN
11227
11228  011126
  (4)  011126  116767  000030  177672  MOVB    RHLD,SB                   ;SAVE RECORD OF FIRST MISS
11229  011134                                                                LET SB :B= RHLD
  (4)  011134  110267  177667          MOVB    R2,WAS
11230  011140                                                                LET WAS :B= R2
  (4)  011140                        $206:                              ENDIF
11231
11232  011140
  (7)  011140  005267  177656          INC     ERRCNT                ;COUNT IT.
11233  011144                                                           LET ERRCNT := ERRCNT + #1
  (4)  011144                        $205:                           ENDIF
11234
11235
11236  011144                                                           ;ALL DONE?
  (6)  011144  020467  177654          CMP     R4,NUMBER             IF R4 EQ NUMBER THEN
  (9)  011150  001003                  BNE     $207
11237                                                                      ;STOP RECEIVER INTERRUPTS
11238  011152                                                              LET @RCSR := @RCSR CLR.BY #RCVRIE
  (7)  011152  042777  000100  170100  BIC     #RCVRIE,@RCSR
11239                                                                      ;MAIN REPEAT LOOP IS CHECKING
11240  011160                                                           ENDIF
  (4)  011160                        $207:
11241
11242                                                                    ;FOR 'R4 = NUMBER' ALSO
11243  011160  000401                  BR      ZZZZ                  ;  EXIT SRV
11244
11245  011162  000000                                                   RHLD:0
11246  011164                       ZZZZ:
11247  011164                                                         ENDSRV
  (1)  011164  000002                  RTI
11248
11249  011166                                                   ENDTST
11250
11251
11252
```

```
11254
11259                           ;;*****************************************************************
11264                           ;;*****************************************************************
  (3)                           ;*TEST 40       TEST BREAK GENERATION LOGIC
  (4)                           ;*              TRANSMIT KNOWN CHAR WITH BREAK SET
  (4)                           ;*              AND COMPARE RECEIVED WITH 0.
  (3)                           ;;*****************************************************************
  (2) 011166  000004            TST40:  SCOPE
  (2)
  (1) 011170  012767  000010  167762            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 011176  012767  000040  167774            MOV     #40,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
11269 011204                                                                    IF #BRK NOTSETIN $USWR THEN
  (6) 011204  032767  010000  170006            BIT     #BRK,$USWR
  (9) 011212  001004                            BNE     $210
11270 011214                                                                            EXIT TST
  (5) 011214  012767  000001  167736            MOV     #1,$TIMES
  (3) 011222  000452                            BR      TST41           ;;;EXIT THIS TEST
11271 011224                                                                    ENDIF
  (4) 011224                    $210:
11272                                                                           ;SET MAINTENANCE BIT
11273 011224                                                                    LET @TCSR := @TCSR SET.BY #MAINT
  (7) 011224  052777  000004  170032            BIS     #MAINT,@TCSR
11274
11275                                                                           ; CLEAR RCVRDONE JUST IN CASE
11276 011232                                                                    LET R0 := @RBUF
  (4) 011232  017700  170024                    MOV     @RBUF,R0
11277
11278
11279                                                                           ;SET BREAK BIT
11280 011236                                                                    LET @TCSR := @TCSR SET.BY #BREAK
  (7) 011236  052777  000001  170020            BIS     #BREAK,@TCSR
11281                                                                           ;NON-ZERO CHAR. '*'
11282 011244                                                                    LET @TBUF := #252
  (4) 011244  012777  000252  170016            MOV     #252,@TBUF
11283 011252                                                                    CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
  (3) 011252  010546                            MOV     R5,-(SP)
  (7) 011254  012745  177777                    MOV     #SET,-(R5)
  (6) 011260  016745  167774                    MOV     RCSR,-(R5)
  (5) 011264  012745  000200                    MOV     #RCVRDONE,-(R5)
  (4) 011270  012745  000500                    MOV     #500,-(R5)
  (3) 011274  004767  000170                    JSR     PC,TIMER
  (3) 011300  012605                            MOV     (SP)+,R5
11284 011302                                                                    IF.ERROR THEN
  (6) 011302  103001                            BCC     $211
11285                                                                                   ; RECIEVER DONE DID NOT SET
11286 011304                                                                        ERRHRD 115
  (1) 011304  104115                            ERROR   115
11287 011306                                                                    ENDIF
  (4) 011306                    $211:
11288
11289 011306                                                                    IFB @RBUF NE #0 THEN
  (6) 011306  105777  167750                    TSTB    @RBUF
  (9) 011312  001401                            BEQ     $212
11290                                                                                   ; BREAK DID NOT EQUAL 0
11291 011314                                                                        ERRHRD 121 ,BADBRK
  (1) 011314  104121                            ERROR   121
```

```
11292  011316                                                    ENDIF
  (4)  011316                          $212:
11293  011316                                                    BRESET  ;CLEAN UP
  (1)  011316  000005                          RESET
11294  011320                                                    EXIT
  (3)  011320  000413                          BR      TST41    ;;;EXIT THIS TEST
11295  011322  051102  040505  020113                           BADBRK: .ASCIZ  /BREAK DID NOT EQUAL 0/
       011330  044504  020104  047516
       011336  020124  050505  040525
       011344  020114  000060
11296
11297  011350                                                ENDTST
```

```
 11299
 11304
 11305                                  ;;*********************************************************************
   (3)                                  ;*TEST 41       NOT A TEST - SEND BACK TO LOOP
   (3)                                  ;;*********************************************************************
   (2)  011350  000004                  TST41:  SCOPE
   (2)
   (1)  011352  012767  000001  167600          MOV     #1,$TIMES       ;;DO 1 ITERATION
 11306  011360  104401  011366                  TYPE    ,65$            ;;TYPE ASCIZ STRING
   (1)  011364  000404                  BR      64$             ;;GET OVER THE ASCIZ
   (1)                                  ;;65$:  .ASCIZ  <CRLF>*CSR: *
   (1)  011376                          64$:
 11307  011376  016746  167652                  MOV     DLADD,-(SP)     ;;SAVE DLADD FOR TYPEOUT
   (1)  011402  104402                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 11308  011404  104401  011412                  TYPE    ,67$            ;;TYPE ASCIZ STRING
   (1)  011410  000405                  BR      66$             ;;GET OVER THE ASCIZ
   (1)                                  ;;67$:  .ASCIZ  *,VECTOR: *
   (1)  011424                          66$:
 11309  011424  016746  167626                  MOV     DLVEC,-(SP)     ;;SAVE DLVEC FOR TYPEOUT
   (1)  011430  104402                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 11310  011432  104401  011440                  TYPE    ,69$            ;;TYPE ASCIZ STRING
   (1)  011436  000405                  BR      68$             ;;GET OVER THE ASCIZ
   (1)                                  ;;69$:  .ASCIZ  *,ERRORS: *
   (1)  011452                          68$:
 11311  011452  016746  167434                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
   (1)  011456  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 11312  011460  005067  167426                  CLR     $ERTTL          ; RESET FOR NEXT DEVICE/PASS
 11313  011464  000167  170254                  JMP     LOOP            ; BACK UP TO THE BEGINNING
```

```
11319
11324
11325                                   ;;BGNMOD        SUBS
11326                                   ;;*****************************************************************
11331  011470                                                   ROUTINE TIMER <HOWLONG,WHICHBIT,REG,SETCLR>
  (2)  011470                           TIMER:
11332                                   ;* ROUTINE:TIMER
11333                                   ;*      THIS ROUTINE IS USED TO TEST THE STATUS OF ANY BIT
11334                                   ;*      IN ANY REGISTER.
11335                                   ;*      INPUTS:
11336                                   ;*            HOWLONG    THE MAXIMUM AMOUNT OF TIME TO SPEND IN
11337                                   ;*                       THIS ROUTINE.
11338                                   ;*            WHICHBIT     A MASK WITH THE BIT(S) SET THAT ARE
11339                                   ;*                         TO BE CHECKED.
11340                                   ;*            REG     A POINTER TO THE REGISTER TO BE CHECKED
11341                                   ;*            SETCLR  THE DESIRED RESULTS
11342                                   ;*                    EITHER #SET OR #CLEAR
11343                                   ;*      OUTPUT:
11344                                   ;*            THE 'C' BIT IS SET TO INDICATE AN ERROR
11345                                   ;*            BUT IT IS TESTED BY THE IF.ERROR STATEMENT
11346                                   ;*
11347                                   ;*
11348                                   ;*            NOTE::  THE USE OF (R5) IS PART OF THE LINKAGE
11349                                   ;*                    MECHANISM BETWEEN THE CALLER AND THE CALLED
11354                                   ;;*****************************************************************
11359
11360          000001                                                   TRUE=   1
11361          000000                                                   FALSE=  0
11362
11363  011470                                                   LET     REGSAV := REG(R5) ; GET POINTER TO REGIST
  (4)  011470  016567  000004  000136           MOV     REG(R5),REGSAV
11364  011476                                                   LET     TIMSAV := HOWLONG(R5) ; SAVE HOWLONG FOR
  (4)  011476  016567  000000  000132           MOV     HOWLONG(R5),TIMSAV
11365  011504                                                   LET     FLAG :B= #FALSE ; INITIALIZE THE EXIT FLA
  (4)  011504  112767  000000  000126           MOVB    #FALSE,FLAG
11366
11367
11368                                                    ;       START OF AN INFINITE LOOP
11369
11370  011512                                           LOOP
  (3)  011512                           $215:
11371                                                            ; TEST TO SEE IF WHICHBIT IS SET
11372  011512                                                IF      WHICHBIT(R5)  NOTSETIN @REGSAV THEN
  (6)  011512  036577  000002  000114           BIT     WHICHBIT(R5),@REGSAV
  (9)  011520  001004                           BNE     $217
11373  011522                                                       LET     HOLDSC :B= #CLR
  (4)  011522  112767  000000  000111           MOVB    #CLR,HOLDSC
11374  011530                                                   ELSE
  (4)  011530  000403                           BR      $220
  (3)  011532                           $217:
11375  011532                                                       LET     HOLDSC :B= #SET ; REMEMBER THIS
  (4)  011532  112767  177777  000101           MOVB    #SET,HOLDSC
11376  011540                                                   ENDIF
  (4)  011540                           $220:
11377
11378                                                            ; NOW SEE IF THAT WAS WHAT WE WANTED
```

```
11379  011540                                                  IFB     HOLDSC EQ SETCLR(R5) THEN
  (6)  011540  126765  000075  000006        CMPB    HOLDSC,SETCLR(R5)
  (9)  011546  001003                        BNE     $221
11380                                                       ; JUST THE THING WE NEEDED
11381  011550                                                       LET     FLAG :B= #TRUE
  (4)  011550  112767  000001  000062        MOVB    #TRUE,FLAG
11382  011556                                                  ENDIF
  (4)  011556                        $221:
11383
11384  011556                                                EXIFB   FLAG EQ #TRUE OR TIMSAV LE #0
  (4)  011556  126727  000056  000001        CMPB    FLAG,#TRUE
  (6)  011564  001414                        BEQ     $216
  (4)  011566  005767  000044                TST     TIMSAV
  (6)  011572  003411                        BLE     $216
11385                                                       ; ONE WAY OR THE OTHER, WE ARE DONE
11386                                                     ; IF WE ARE STILL HERE THEN HANG AROUND A WHILE
11387
11388  011574                                                       WAITMS  1          ;WAIT FOR 10 MILLI-SECONDS
  (4)  011574  010546                        MOV     R5,-(SP)
  (5)  011576  012745  000001                MOV     #1,-(R5)
  (4)  011602  004767  000140                JSR     PC,WAIT
  (4)  011606  012605                        MOV     (SP)+,R5
11389  011610                                                       LET     TIMSAV := TIMSAV - #1 ; COUNTING DOWN
  (7)  011610  005367  000022                DEC     TIMSAV
11390  011614                                                  ENDLOOP                    ; CONTINUED AT THE TOP
  (4)  011614  000736                        BR      $215
  (3)  011616                        $216:
11391
11392                                                   ; ONLY 2 WAYS TO GET HERE
11393                                                   ;       1). WE RAN OUT OF TIME---ERROR ! !
11394                                                   ;       2). THE BIT IS IN THE CORRECT CONDITION--GOOD ! !
11395
11396  011616                                                  IFB     FLAG EQ #TRUE THEN
  (6)  011616  126727  000016  000001        CMPB    FLAG,#TRUE
  (9)  011624  001001                        BNE     $222
11397  011626                                                       RETURN  NO.ERROR    ; GOOD
  (4)  011626  000405                        BR      $213
11398  011630                                                  ENDIF
  (4)  011630                        $222:
11399  011630                                                  RETURN  ERROR           ; BAD
  (2)  011630  000261                        SEC
  (4)  011632  000404                        BR      $214
11400
11401  011634  000000                                        REGSAV: .WORD 0
11402  011636  000000                                        TIMSAV: .WORD 0
11403  011640     000                                        FLAG:   .BYTE 0
11404  011641     000                                        HOLDSC: .BYTE 0
11405                                                   ; WE ARE DONE  GO BACK HOME
11406  011642                                                       ENDRTN
  (3)  011642                        $213:
  (2)  011642  000241                        CLC
  (3)  011644                        $214:
  (2)  011644  000207                        RTS     PC
```

```
 11408
 11409
 11414
 11419  011646                          ;;*****************************************************************
   (2)  011646                          ROUTINE DATLNG <MASK>
                                        DATLNG:
 11420                                  ;* ROUTINE:DATLNG
 11421                                  ;*       THIS ROUTINE SETS UP A MASK FOR DATA, WITH
 11422                                  ;*  INPUT - NOTHING IS PASSED TO THIS ROUTINE
 11423                                  ;*      BUT GLOBAL INFORMATION IS ASSUMED TO EXIST:
 11424                                  ;*      $USWR-- THE WORD FOR SOFTWARE PARAMETERS
 11425                                  ;*      DATA-- A MASK FOR THE LOCATION OF THE OCTAL
 11426                                  ;*               NUMBER OF DATA BITS
 11427                                  ;*  OUTPUT----
 11428                                  ;*      MASK-- A MASK OF BINARY ONES RIGHT-JUSTIFIED
 11429                                  ;*              THE NUMBER OF WHICH IS DEFINED IN $USWR WORD.
 11430
 11435                                  ;;*****************************************************************
 11440
 11441  011646                                                                  LET MASK(R5) := #0              ; START
   (4)  011646  005065  000000                  CLR     MASK(R5)
 11442  011652                                                                  LET NUMBR := $USWR AND #DATA
   (4)  011652  016767  167342  000062          MOV     $USWR,NUMBR
   (7)  011660  016746  000056                  MOV     NUMBR,-(SP)
   (7)  011664  042716  000017                  BIC     #DATA,(SP)
   (7)  011670  042667  000046                  BIC     (SP)+,NUMBR
 11443
 11444  011674                                                                  INCR I FROM #1 TO NUMBR BY #1
   (4)  011674  012767  000001  167370          MOV     #1,I
   (5)  011702  000402                          BR      $225
   (4)  011704                          $226:
   (8)  011704  005267  167362                  INC     I
   (5)  011710                          $225:
   (5)  011710  026767  167356  000024          CMP     I,NUMBR
   (7)  011716  003006                          BGT     $227
 11445  011720                                                                      LET MASK(R5) := MASK(R5) SHIFT #1
   (8)  011720  006365  000000                  ASL     MASK(R5)
 11446  011724                                                                      LET MASK(R5) := MASK(R5) SET.BY #1
   (7)  011724  052765  000001  000000          BIS     #1,MASK(R5)
 11447  011732                                                                  ENDINC
   (4)  011732  000764                          BR      $226
   (3)  011734                          $227:
 11448  011734                                                                  LET MASK(R5) := COMP MASK(R5)
   (7)  011734  005165  000000                  COM     MASK(R5)
 11449  011740                                                              RETURN
   (4)  011740  000401                          BR      $223
 11450  011742  000000                          NUMBR:0
 11451  011744                                                          ENDRTN
   (3)  011744                          $223:
   (3)  011744                          $224:
   (2)  011744  000207                          RTS     PC
```

```
 11453
 11454
 11459                                  ;;*****************************************************************
 11464  011746                          ROUTINE WAIT    <TIME>
   (2)  011746                          WAIT:
 11465                                  ;* ROUTINE:WAIT
 11466                                  ;*       THIS ROUTINE IS USED TO DELAY EXECUTION OF THE
 11467                                  ;*       MAIN  PROGRAM  FOR A SPECIFIED AMOUNT OF TIME.
 11468                                  ;*       THIS   IS   ACCOMPLISHED   BY  INCREMENTING  A
 11469                                  ;*       REGISTER UP TO A LIMIT.  THE INNER LOOP IS SET
 11470                                  ;*       TO APPROXIMATE 1 MILLI SEC.
 11475                                  ;;*****************************************************************
 11476  011746  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
   (2)  011750  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
   (2)  011752  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 11481  011754                                  LET R1 := TIME(R5)
   (4)  011754  016501  000000                  MOV     TIME(R5),R1
 11482  011760                                  INCRU R2 FROM #1 TO R1 BY #1
   (4)  011760  012702  000001                  MOV     #1,R2
   (5)  011764  000402                          BR      $232
   (4)  011766                          $233:
   (8)  011766  062702  000001                  ADD     #01,R2
   (5)  011772                          $232:
   (5)  011772  020201                          CMP     R2,R1
   (7)  011774  101010                          BHI     $234
 11483  011776                                  INCR R3 FROM #0 TO #100 BY #1
   (4)  011776  005003                          CLR     R3
   (5)  012000  000401                          BR      $235
   (4)  012002                          $236:
   (8)  012002  005203                          INC     R3
   (5)  012004                          $235:
   (5)  012004  020327  000100                  CMP     R3,#100
   (7)  012010  003001                          BGT     $237
 11484  012012                                  ENDINC
   (4)  012012  000773                          BR      $236
   (3)  012014                          $237:
 11485  012014                                  ENDINC
   (4)  012014  000764                          BR      $233
   (3)  012016                          $234:
 11490  012016  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
   (2)  012020  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
   (2)  012022  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 11495  012024                          ENDRTN
   (3)  012024                          $230:
   (3)  012024                          $231:
   (2)  012024  000207                          RTS     PC
```

```
 11497
 11502
 11503                                  .SBTTL  INTSRV  INTERRUPT SERVICE ROUTINE
 11504                                  ;;*****************************************************************
 11505  012026                          INTSRV:
 11506                                  ;* SERVICE ROUTINE: INTSRV
 11507                                  ;*       THIS GLOBAL ROUTINE DOES NOTHING BUT INCREMENT
 11508                                  ;*       'INTFLAG'  EACH TIME IT IS CALLED.  IT ASSUMES
 11509                                  ;*       THAT THE MAIN CALLING ROUTINE WILL  KNOW  WHAT
 11510                                  ;*       TO LOOK FOR.
 11515                                  ;;*****************************************************************
 11520
 11521                                                  ;ADD 1 TO 'INTERRUPT OCCURED' FLAG
 11522  012026                                                          LET INTFLAG := INTFLAG + #1
   (7)  012026  005267  000002                  INC     INTFLAG
 11523  012032                                  ENDSRV                  ;THAT'S ALL
   (1)  012032  000002                          RTI
 11524  012034  000000                  INTFLAG:  0
```

```
11526
11527
11528  012036                          ROUTINE MYTYPE
  (2)  012036                          MYTYPE:
11533                                  ;;*****************************************************************************
11534  012036  104401  012044                TYPE    ,65$                ;;TYPE ASCIZ STRING
  (1)  012042  000405                        BR      64$                 ;;GET OVER THE ASCIZ
  (1)                                 ;;65$:  .ASCIZ  <CRLF>*TEST # *
  (1)  012056                         64$:
11535  012056  016746  167116                MOV     $TESTN,-(SP)        ;;SAVE $TESTN FOR TYPEOUT
  (1)  012062  104402                        TYPOC                       ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
11536  012064  104401  012072                TYPE    ,67$                ;;TYPE ASCIZ STRING
  (1)  012070  000405                        BR      66$                 ;;GET OVER THE ASCIZ
  (1)                                 ;;67$:  .ASCIZ  *,ERROR # *
  (1)  012104                         66$:
11537  012104  116767  167004  167064        MOVB    $ITEMB,$FATAL       ; APT FATAL ERROR NUMBER
11538  012112  016746  167060                MOV     $FATAL,-(SP)        ;;SAVE $FATAL FOR TYPEOUT
  (1)  012116  104402                        TYPOC                       ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
11539  012120  104401  012126                TYPE    ,69$                ;;TYPE ASCIZ STRING
  (1)  012124  000404                        BR      68$                 ;;GET OVER THE ASCIZ
  (1)                                 ;;69$:  .ASCIZ  *,PC = *
  (1)  012136                         68$:
11540  012136  016746  166754                MOV     $ERRPC,-(SP)        ;;SAVE $ERRPC FOR TYPEOUT
  (1)  012142  104402                        TYPOC                       ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
11541  012144  104401  012152                TYPE    ,71$                ;;TYPE ASCIZ STRING
  (1)  012150  000404                        BR      70$                 ;;GET OVER THE ASCIZ
  (1)                                 ;;71$:  .ASCIZ  *,CSR: *
  (1)  012162                         70$:
11542  012162  016746  167066                MOV     DLADD,-(SP)         ;;SAVE DLADD FOR TYPEOUT
  (1)  012166  104402                        TYPOC                       ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
11543  012170  104401  012176                TYPE    ,73$                ;;TYPE ASCIZ STRING
  (1)  012174  000405                        BR      72$                 ;;GET OVER THE ASCIZ
  (1)                                 ;;73$:  .ASCIZ  *,VECTOR: *
  (1)  012210                         72$:
11544  012210  016746  167042                MOV     DLVEC,-(SP)         ;;SAVE DLVEC FOR TYPEOUT
  (1)  012214  104402                        TYPOC                       ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
11549  012216                          ENDRTN
  (3)  012216                          $240:
  (3)  012216                          $241:
  (2)  012216  000207                        RTS     PC
```

```
11551
11552  012220                           ROUTINE CYCLE
  (2)  012220                           CYCLE:
11557                                   ;;*****************************************************************
11558                                   ;* ROUTINE:     CYCLE
11559                                   ;*       THIS  ROUTINE  CAUSES  ADRS   TO POINT TO  THE
11560                                   ;*       ADDRESS  OF  DLV11-E  UNDER  TEST,  ADRS +2 TO
11561                                   ;*       POINT TO THE VECTOR OF THE DLV11-E UNDER TEST.
11562                                   ;*      IT  KEEPS  TRACK  OF THE CURRENT DEVICE AND BIT
11563                                   ;*      MASKS.
11564                                   ;;*****************************************************************
11569  012220                                                  REPEAT
  (3)  012220                           $244:
11570  012220                                                      IF BITMASK EQ #0 THEN
  (6)  012220  005767  000122                   TST     BITMASK
  (9)  012224  001027                           BNE     $245
11571  012226                                                          IF INITFLAG EQ #1 THEN
  (6)  012226  026727  000116  000001           CMP     INITFLAG,#1
  (9)  012234  001003                           BNE     $246
11572  012236                                                              LET INITFLAG := #0
  (4)  012236  005067  000106                   CLR     INITFLAG
11573  012242                                                          ELSE
  (4)  012242  000403                           BR      $247
  (3)  012244                           $246:
11574  012244                                                              CALL $EOP   ; AS A SUBROUTINE
  (3)  012244  004767  000110                   JSR     PC,$EOP
11575
11576  012250                           SPECIALADDRESS:                ; BECAUSE $EOP RETURNS AS A JUMP
11577  012250                                                              LET R0 := POP
  (4)  012250  012600                           MOV     (SP)+,R0
11578  012252                                                          ENDIF
  (4)  012252                           $247:
11579  012252                                                          LET BITMASK := #1
  (4)  012252  012767  000001  000066           MOV     #1,BITMASK
11580  012260                                                          LET $DEVCT := #1
  (4)  012260  012767  000001  166716           MOV     #1,$DEVCT
11581  012266                                                          LET ADDRESS := $BASE
  (4)  012266  016767  166756  000056           MOV     $BASE,ADDRESS
11582  012274                                                          LET VECTOR := $VECT1
  (4)  012274  016767  166744  000052           MOV     $VECT1,VECTOR
11583  012302                                                      ELSE
  (4)  012302  000410                           BR      $250
  (3)  012304                           $245:
11584  012304                                                          LET R4 := #10
  (4)  012304  012704  000010                   MOV     #10,R4
11585  012310                                                          LET BITMASK := BITMASK ROTATE 1
  (8)  012310  006167  000032                   ROL     BITMASK
11586  012314                                                          LET ADDRESS  := ADDRESS + R4
  (7)  012314  060467  000032                   ADD     R4,ADDRESS
11587  012320                                                          LET VECTOR := VECTOR + R4
  (7)  012320  060467  000030                   ADD     R4,VECTOR
11588  012324                                                      ENDIF
  (4)  012324                           $250:
11589  012324                                                  UNTIL   BITMASK SETIN $DEVM
  (3)  012324  036767  000016  166720           BIT     BITMASK,$DEVM
  (6)  012332  001732                           BEQ     $244
```

```
11590
11591  012334                                                    LET ADRS := #ADDRESS
  (4)  012334  012701  012352                MOV     #ADDRESS,ADRS
11592  012340                                                    LET $DEVCT := $DEVCT + #1
  (7)  012340  005267  166640                INC     $DEVCT
11593  012344                                                    RETURN
  (4)  012344  000404                        BR      $242
11594  012346  000000              BITMASK:  0
11595  012350  000001              INITFLAG: 1
11596  012352  000000              ADDRESS:  0
11597  012354  000000              VECTOR:   0
11598
11599  012356                                                    ENDRTN
  (3)  012356                      $242:
  (3)  012356                      $243:
  (2)  012356  000207                        RTS     PC
11600
11601
```

```
11603
11608                                   .SBTTL  END OF PASS ROUTINE
  (1)
  (2)                                   ;;**********************************************************************
  (1)                                   ;*INCREMENT THE PASS NUMBER ($PASS)
  (1)                                   ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
  (1)                                   ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
  (1)                                   ;*IF THERES A MONITOR GO TO IT
  (1)                                   ;*IF THERE ISN'T JUMP TO SPECIALADDRESS
  (1)
  (1)  012360                           $EOP:
  (1)  012360  000004                           SCOPE
  (1)  012362  005067  166514                   CLR     $TSTNM          ;;ZERO THE TEST NUMBER
  (1)  012366  005067  166566                   CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
  (1)  012372  005267  166604                   INC     $PASS           ;;INCREMENT THE PASS NUMBER
  (1)  012376  042767  100000  166576           BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
  (1)  012404  005327                           DEC     (PC)+           ;;LOOP?
  (1)  012406  000001           $EOPCT: .WORD   1
  (1)  012410  003022                           BGT     $DOAGN          ;;YES
  (1)  012412  012737                           MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
  (1)  012414  000001           $ENDCT: .WORD   1
  (1)  012416  012406                           $EOPCT
  (1)  012420  104401  012465                   TYPE    ,$ENDMG         ;;TYPE "END PASS #"
  (2)  012424  016746  166552                   MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
  (2)  012430  104405                           TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
  (1)  012432  104401  012462                   TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
  (1)  012436  013700  000042   $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
  (1)  012442  001405                           BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
  (1)  012444  000005                           RESET                   ;;CLEAR THE WORLD
  (1)  012446  004710           $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
  (1)  012450  000240                           NOP                     ;;SAVE ROOM
  (1)  012452  000240                           NOP                     ;;FOR
  (1)  012454  000240                           NOP                     ;;ACT11
  (1)  012456                   $DOAGN:
  (1)  012456  000137                           JMP     @(PC)+          ;;RETURN
  (1)  012460  012250           $RTNAD: .WORD   SPECIALADDRESS
  (1)
  (1)
  (1)  012462     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
  (1)  012465     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
  (1)  012472  050040  051501  020123
  (1)  012500  000043
```

```
11610
11611                                   .SBTTL  POWER DOWN AND UP ROUTINES
  (1)
  (2)                                   ;;*****************************************************************
  (1)                                   ;POWER DOWN ROUTINE
  (1)  012502  012737  012646  000024   $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
  (1)  012510  012737  000300  000026          MOV     #PR6,@#PWRVEC+2 ;;PRIO:6
  (3)  012516  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
  (3)  012520  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
  (3)  012522  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
  (3)  012524  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
  (3)  012526  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
  (3)  012530  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
  (3)  012532  017746  166402                  MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
  (1)  012536  010667  000110                  MOV     SP,$SAVR6       ;;SAVE SP
  (1)  012542  012737  012554  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
  (1)  012550  000000                          HALT
  (1)  012552  000776                          BR      .-2             ;;HANG UP
  (1)
  (2)                                   ;;*****************************************************************
  (1)                                   ;POWER UP ROUTINE
  (1)  012554  012737  012646  000024   $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
  (1)  012562  016706  000064                  MOV     $SAVR6,SP       ;;GET SP
  (1)  012566  005067  000060                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
  (1)  012572  005267  000054          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
  (1)  012576  001375                          BNE     1$              ;;OF  WORD
  (3)  012600  012677  166334                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
  (3)  012604  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
  (3)  012606  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
  (3)  012610  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
  (3)  012612  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
  (3)  012614  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
  (3)  012616  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
  (1)  012620  012737  012502  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
  (1)  012626  012737  000300  000026          MOV     #PR6,@#PWRVEC+2 ;;PRIO:6
  (1)  012634  104401                          TYPE                    ;REPORT THE POWER FAILURE
  (1)  012636  012654          $PWRMG: .WORD   $POWER                  ;;POWER FAIL MESSAGE POINTER
  (1)  012640  012716                  MOV     (PC)+,(SP)              ;;RESTART AT START
  (1)  012642  001336          $PWRAD: .WORD   START                   ;;RESTART ADDRESS
  (1)  012644  000002                  RTI
  (1)  012646  000000          $ILLUP: HALT                            ;;THE POWER UP SEQUENCE WAS STARTED
  (1)  012650  000776                  BR      .-2                     ;; BEFORE THE POWER DOWN WAS COMPLETE
  (1)  012652  000000          $SAVR6: 0                               ;,PUT THE SP HERE
  (1)  012654  005015  047520  042527  $POWER: .ASCIZ  <15><12>''POWER''
  (1)  012662  000122
  (1)                                   .EVEN
```

```
11613
11614                                   .SBTTL  TYPE ROUTINE
  (1)
  (2)                                   ;;**********************************************************************
  (1)                                   ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
  (1)                                   ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
  (1)                                   ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
  (1)                                   ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
  (1)                                   ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
  (1)                                   ;*
  (1)                                   ;*CALL:
  (1)                                   ;*1) USING A TRAP INSTRUCTION
  (1)                                   ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
  (1)                                   ;*OR
  (1)                                   ;*      TYPE
  (1)                                   ;*      MESADR
  (1)                                   ;*
  (1)
  (1)  012664  105767  166267           $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
  (1)  012670  100002                           BPL     1$              ;;BR IF YES
  (1)  012672  000000                           HALT                    ;;HALT HERE IF NO TERMINAL
  (1)  012674  000430                           BR      3$              ;;LEAVE
  (1)  012676  010046                   1$:     MOV     R0,-(SP)        ;;SAVE R0
  (1)  012700  017600  000002                   MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
  (1)  012704  122767  000001  166302           CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
  (1)  012712  001011                           BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
  (1)  012714  132767  000100  166273           BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
  (1)  012722  001405                           BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
  (1)  012724  010067  000004                   MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
  (1)  012730  004767  000774                   JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
  (1)  012734  000000                   61$:    .WORD   0               ;;MESSAGE ADDRESS
  (1)  012736  132767  000040  166251   62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
  (1)  012744  001003                           BNE     60$             ;;YES,SKIP TYPE OUT
  (1)  012746  112046                   2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
  (1)  012750  001005                           BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
  (1)  012752  005726                           TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
  (1)  012754  012600                   60$:    MOV     (SP)+,R0        ;;RESTORE R0
  (1)  012756  062716  000002           3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
  (1)  012762  000002                           RTI                     ;;RETURN
  (1)  012764  122716  000011           4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
  (1)  012770  001430                           BEQ     8$
  (1)  012772  122716  000200                   CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
  (1)  012776  001006                           BNE     5$
  (1)  013000  005726                           TST     (SP)+           ;;POP  <CR><LF> EQUIV
  (1)  013002  104401                           TYPE                    ;;TYPE A CR AND LF
  (1)  013004  001171                           $CRLF
  (1)  013006  105067  000130                   CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
  (1)  013012  000755                           BR      2$              ;;GET NEXT CHARACTER
  (1)  013014  004767  000056           5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
  (1)  013020  126726  166132           6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
  (1)  013024  001350                           BNE     2$              ;;IF NO GO GET NEXT CHAR.
  (1)  013026  016746  166122                   MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
  (1)                                           ;;AND THE NULL CHAR.
  (1)  013032  105366  000001           7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
  (1)  013036  002770                           BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
  (1)  013040  004767  000032                   JSR     PC,$TYPEC       ;;GO TYPE A NULL
```

```
(1)  013044  105367  000072            DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
(1)  013050  000770                    BR      7$              ;;LOOP
(1)
(1)                                ;HORIZONTAL TAB PROCESSOR
(1)
(1)  013052  112716  000040       8$:  MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
(1)  013056  004767  000014       9$:  JSR     PC,$TYPEC       ;;TYPE A SPACE
(1)  013062  132767  000007  000052    BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
(1)  013070  001372                    BNE     9$              ;;TAB STOP
(1)  013072  005726                    TST     (SP)+           ;;POP SPACE OFF STACK
(1)  013074  000724                    BR      2$              ;;GET NEXT CHARACTER
(1)  013076  105777  166046  $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
(1)  013102  100375                    BPL     $TYPEC
(1)  013104  116677  000002  166040    MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
(1)
(1)  013112  122766  000015  000002  CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
(1)  013120  001003                    BNE     1$              ;;BRANCH IF NO
(1)  013122  105067  000014            CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
(1)  013126  000406                    BR      $TYPEX          ;;EXIT
(1)  013130  122766  000012  000002  1$:  CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
(1)  013136  001402                    BEQ     $TYPEX          ;;BRANCH IF YES
(1)  013140  105227                    INCB    (PC)+           ;;COUNT THE CHARACTER
(1)  013142  000000            $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
(1)  013144  000207            $TYPEX: RTS     PC
(1)
(1)
(1)
```

```
 11616
 11617                                  .SBTTL  TTY INPUT ROUTINE
   (1)
   (2)                                  ;;**********************************************************************
   (1)
   (1)
   (1)                                  .ENABL  LSB
   (1)
   (2)                                  ;;**********************************************************************
   (1)                                  ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
   (1)                                  ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
   (1)                                  ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
   (1)                                  ;*WHEN OPERATING IN TTY FLAG MODE.
   (1)  013146  022767  000176  165764  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
   (1)  013154  001074                          BNE     15$             ;;BRANCH IF NO
   (1)  013156  105777  165762                  TSTB    @$TKS           ;;CHAR THERE?
   (1)  013162  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
   (1)  013164  117746  165756                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
   (1)  013170  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
   (1)  013174  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
   (1)  013200  001062                          BNE     15$             ;;NO, RETURN TO USER
   (1)  013202  126727  165726  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
   (1)  013210  001456                          BEQ     15$             ;;BRANCH IF YES
   (1)
   (1)  013212  104401  013673                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
   (1)  013216  104401  013700  $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
   (2)  013222  016746  164750                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
   (2)  013226  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
   (1)  013230  104401  013711                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
   (1)  013234  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
   (1)  013236  005046                          CLR     -(SP)
   (1)                                          ;;THE NEW SWR
   (1)  013240  105777  165700          7$:     TSTB    @$TKS           ;;CHAR THERE?
   (1)  013244  100375                          BPL     7$              ;;IF NOT TRY AGAIN
   (1)
   (1)  013246  117746  165674                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
   (1)  013252  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
   (1)
   (1)
   (1)
   (1)  013256  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
   (1)  013262  001005                          BNE     10$             ;;BRANCH IF NOT
   (1)  013264  104401  013666                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
   (1)  013270  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
   (1)  013274  000757                          BR      19$             ;;LET'S TRY IT AGAIN
   (1)
   (1)
   (1)  013276  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
   (1)  013302  001022                          BNE     16$             ;;BRANCH IF NO
   (1)  013304  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
   (1)  013310  001403                          BEQ     11$             ;;BRANCH IF YES
   (1)  013312  016677  000002  165620          MOV     2(SP),@SWR      ;;SAVE NEW SWR
   (1)  013320  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
   (1)  013324  104401  001171          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
   (1)  013330  126727  165601  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
   (1)  013336  001003                          BNE     15$             ;;BRANCH IF NOT
```

```
(1)  013340  012777  000100  165576          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
(1)  013346  000002                  15$:     RTI                     ;;RETURN
(1)  013350  004767  177522          16$:     JSR     PC,$TYPEC       ;;ECHO CHAR
(1)  013354  021627  000060                   CMP     (SP),#60        ;;CHAR < 0?
(1)  013360  002420                           BLT     18$             ;;BRANCH IF YES
(1)  013362  021627  000067                   CMP     (SP),#67        ;;CHAR > 7?
(1)  013366  003015                           BGT     18$             ;;BRANCH IF YES
(1)  013370  042726  000060                   BIC     #60,(SP)+       ;;STRIP-OFF ASCII
(1)  013374  005766  000002                   TST     2(SP)           ;;IS THIS THE FIRST CHAR
(1)  013400  001403                           BEQ     17$             ;;BRANCH IF YES
(1)  013402  006316                           ASL     (SP)            ;;NO, SHIFT PRESENT
(1)  013404  006316                           ASL     (SP)            ;;    CHAR OVER TO MAKE
(1)  013406  006316                           ASL     (SP)            ;;    ROOM FOR NEW ONE.
(1)  013410  005266  000002          17$:     INC     2(SP)           ;;KEEP COUNT OF CHAR
(1)  013414  056616  177776                   BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
(1)  013420  000707                           BR      7$              ;;GET THE NEXT ONE
(1)  013422  104401  001170          18$:     TYPE    ,$QUES          ;;TYPE ?<CR><LF>
(1)  013426  000720                           BR      20$             ;;SIMULATE CONTROL-U
(1)                                 .DSABL   LSB
(1)
(1)
(2)                                 ;;**************************************************************************
(1)                                 ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
(1)                                 ;*CALL:
(1)                                 ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
(1)                                 ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
(1)                                 ;*                              ;;WITH PARITY BIT STRIPPED OFF
(1)                                 ;
(1)
(1)  013430  011646                 $RDCHR:  MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
(1)  013432  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
(1)  013440  105777  165500          1$:      TSTB    @$TKS           ;;WAIT FOR
(1)  013444  100375                           BPL     1$              ;;A CHARACTER
(1)  013446  117766  165474  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
(1)  013454  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
(1)  013462  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
(1)  013470  001013                           BNE     3$              ;;BRANCH IF NO
(1)  013472  105777  165446          2$:      TSTB    @$TKS           ;;WAIT FOR A CHARACTER
(1)  013476  100375                           BPL     2$              ;;LOOP UNTIL ITS THERE
(1)  013500  117746  165442                   MOVB    @$TKB,-(SP)     ;;GET CHARACTER
(1)  013504  042716  177600                   BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
(1)  013510  022627  000021                   CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
(1)  013514  001366                           BNE     2$              ;;IF NOT DISCARD IT
(1)  013516  000750                           BR      1$              ;;YES, RESUME
(1)  013520  026627  000004  000140  3$:      CMP     4(SP),#140      ;;IS IT UPPER CASE?
(1)  013526  002407                           BLT     4$              ;;BRANCH IF YES
(1)  013530  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
(1)  013536  003003                           BGT     4$              ;;BRANCH IF YES
(1)  013540  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
(1)  013546  000002                  4$:      RTI                     ;;GO BACK TO USER
(2)                                 ;;**************************************************************************
(1)                                 ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
(1)                                 ;*CALL:
(1)                                 ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
(1)                                 ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
(1)                                 ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
```

```
(1)
(1)  013550  010346                    $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
(1)  013552  012703  013656            1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
(1)  013556  022703  013666            2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
(1)  013562  101405                            BLOS    4$              ;;BR IF YES
(1)  013564  104410                            RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
(1)  013566  112613                            MOVB    (SP)+,(R3)      ;;GET CHARACTER
(1)  013570  122713  000177            10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
(1)  013574  001003                            BNE     3$              ;;SKIP IF NOT
(1)  013576  104401  001170            4$:     TYPE    ,$QUES          ;;TYPE A '?'
(1)  013602  000763                            BR      1$              ;;CLEAR THE BUFFER AND LOOP
(1)  013604  111367  000044            3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
(1)  013610  104401  013654                    TYPE    ,9$
(1)  013614  122723  000015                    CMPB    #15,(R3)+       ;;CHECK FOR RETURN
(1)  013620  001356                            BNE     2$              ;;LOOP IF NOT RETURN
(1)  013622  105063  177777                    CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
(1)  013626  104401  001172                    TYPE    ,$LF            ;;TYPE A LINE FEED
(1)  013632  012603                            MOV     (SP)+,R3        ;;RESTORE R3
(1)  013634  011646                            MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
(1)  013636  016666  000004  000002            MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
(1)  013644  012766  013656  000004            MOV     #$TTYIN,4(SP)
(1)  013652  000002                            RTI                     ;;RETURN
(1)  013654     000            9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
(1)  013655     000                    .BYTE   0                       ;;TERMINATOR
(1)  013656  000010            $TTYIN: .BLKB   8.                      ;;RESERVE 8 BYTES FOR TTY INPUT
(1)  013666  052536  005015  000  $CNTLU: .ASCIZ  /^U/<15><12>         ;;CONTROL "U"
(1)  013673     136  006507  000012 $CNTLG: .ASCIZ  /^G/<15><12>      ;;CONTROL "G"
(1)  013700  005015  053523  020122 $MSWR:  .ASCIZ  <15><12>/SWR = /
(1)  013706  020075     000
(1)  013711     040  047040  053505 $MNEW:  .ASCIZ  /  NEW = /
(1)  013716  036440  000040
```

```
                                        .SBTTL  APT COMMUNICATIONS ROUTINE
                                        ;;
                                        ;;**********************************************************************
(1)  013722  112767  000001  000236  $ATY1:  MOVB    #1,$FFLG          ;;TO REPORT FATAL ERROR
(1)  013730  112767  000001  000226  $ATY3:  MOVB    #1,$MFLG          ;;TO TYPE A MESSAGE
(1)  013736  000403                          BR      $ATYC
(1)  013740  112767  000001  000220  $ATY4:  MOVB    #1,$FFLG          ;;TO ONLY REPORT FATAL ERROR
(1)  013746                          $ATYC:
(3)  013746  010046                          MOV     R0,-(SP)          ;;PUSH R0 ON STACK
(3)  013750  010146                          MOV     R1,-(SP)          ;;PUSH R1 ON STACK
(1)  013752  105767  000206                  TSTB    $MFLG             ;;SHOULD TYPE A MESSAGE?
(1)  013756  001450                          BEQ     5$                ;;IF NOT:  BR
(1)  013760  122767  000001  165226          CMPB    #APTENV,$ENV      ;;OPERATING UNDER APT?
(1)  013766  001031                          BNE     3$                ;;IF NOT:  BR
(1)  013770  132767  000100  165217          BITB    #APTSPOOL,$ENVM   ;;SHOULD SPOOL MESSAGES?
(1)  013776  001425                          BEQ     3$                ;;IF NOT:  BR
(1)  014000  017600  000004                  MOV     @4(SP),R0         ;;GET MESSAGE ADDR.
(1)  014004  062766  000002  000004          ADD     #2,4(SP)                  ;;BUMP RETURN ADDR.
(1)  014012  005767  165156          1$:     TST     $MSGTYPE          ;;SEE IF DONE W/ LAST XMISSION?
(1)  014016  001375                          BNE     1$                ;;IF NOT:  WAIT
(1)  014020  010067  165164                  MOV     R0,$MSGAD
(1)                                          ;;PUT ADDR IN MAILBOX
(1)  014024  105720                  2$:     TSTB    (R0)+             ;;FIND END OF MESSAGE
(1)  014026  001376                          BNE     2$
(1)  014030  166700  165154                  SUB     $MSGAD,R0         ;;SUB START OF MESSAGE
(1)  014034  006200                          ASR     R0                ;;GET MESSAGE LNGTH IN WORDS
(1)  014036  010067  165150                  MOV     R0,$MSGLGT        ;;PUT LENGTH IN MAILBOX
(1)  014042  012767  000004  165124          MOV     #4,$MSGTYPE       ;;TELL APT TO TAKE MSG.
(1)  014050  000413                          BR      5$
(1)  014052  017667  000004  000016  3$:     MOV     @4(SP),4$         ;;PUT MSG ADDR IN JSR LINKAGE
(1)  014060  062766  000002  000004          ADD     #2,4(SP)                  ;;BUMP RETURN ADDRESS
(3)  014066  016746  163704                  MOV     177776,-(SP)      ;;PUSH 177776 ON STACK
(1)  014072  004767  176566                  JSR     PC,$TYPE          ;;CALL TYPE MACRO
(1)  014076  000000                  4$:     .WORD   0
(1)  014100                          5$:
(1)  014100  105767  000062          10$:    TSTB    $FFLG             ;;SHOULD REPORT FATAL ERROR?
(1)  014104  001416                          BEQ     12$               ;;IF NOT:  BR
(1)  014106  005767  165102                  TST     $ENV              ;;RUNNING UNDER APT?
(1)  014112  001413                          BEQ     12$               ;;IF NOT:  BR
(1)  014114  005767  165054          11$:    TST     $MSGTYPE          ;;FINISHED LAST MESSAGE?
(1)  014120  001375                          BNE     11$               ;;IF NOT:  WAIT
(1)  014122  017667  000004  165046          MOV     @4(SP),$FATAL     ;;GET ERROR #
(1)  014130  062766  000002  000004          ADD     #2,4(SP)                  ;;BUMP RETURN ADDR.
(1)  014136  005267  165032                  INC     $MSGTYPE          ;;TELL APT TO TAKE ERROR
(1)  014142  105067  000020          12$:    CLRB    $FFLG             ;;CLEAR FATAL FLAG
(1)  014146  105067  000013                  CLRB    $LFLG             ;;CLEAR LOG FLAG
(1)  014152  105067  000006                  CLRB    $MFLG             ;;CLEAR MESSAGE FLAG
(3)  014156  012601                          MOV     (SP)+,R1          ;;POP STACK INTO R1
(3)  014160  012600                          MOV     (SP)+,R0          ;;POP STACK INTO R0
(1)  014162  000207                          RTS     PC                ;;RETURN
(1)  014164     000          $MFLG:  .BYTE   0                 ;;MESSG. FLAG
(1)  014165     000          $LFLG:  .BYTE   0
(1)                                          ;;LOG FLAG
(1)  014166     000          $FFLG:  .BYTE   0                 ;;FATAL FLAG
(1)          014170                  .EVEN
```

```
    (1)         000200                  APTSIZE=200
    (1)         000001                  APTENV=001
    (1)         000100                  APTSPOOL=100
    (1)         000040                  APTCSUP=040
```

```
                                .SBTTL  ERROR HANDLER ROUTINE

                                ;;*********************************************************************
                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                ;*AND GO TO MYTYPE ON ERROR
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW15=1         HALT ON ERROR
                                ;*SW13=1         INHIBIT ERROR TYPEOUTS
                                ;*SW10=1         BELL ON ERROR
                                ;*SW09=1         LOOP ON ERROR
                                ;*CALL
                                ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

(1)  014170                     $ERROR:
(1)  014170  104407                     CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
(1)  014172  105267  164705     7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
(1)  014176  001775                     BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
(1)  014200  016777  164676  164734     MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
(1)  014206  032777  002000  164724     BIT     #BIT10,@SWR     ;;BELL ON ERROR?
(1)  014214  001402                     BEQ     1$              ;;NO - SKIP
(1)  014216  104401  001164             TYPE    ,$BELL          ;;RING BELL
(1)  014222  005267  164664     1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
(1)  014226  011667  164664             MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
(1)  014232  162767  000002  164656     SUB     #2,$ERRPC
(1)  014240  117767  164652  164646     MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
(1)  014246  032777  020000  164664     BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
(1)  014254  001004                     BNE     20$             ;;SKIP TYPEOUTS
(1)  014256  004767  175554             JSR     PC,MYTYPE       ;;GO TO USER ERROR ROUTINE
(1)  014262  104401  001171             TYPE    ,$CRLF
(1)  014266                     20$:
(1)  014266  122767  000001  164720     CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
(1)  014274  001007                     BNE     2$              ;;NO,SKIP APT ERROR REPORT
(1)  014276  116767  164612  000004     MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
(1)  014304  004767  177430             JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
(1)  014310     000             21$:    .BYTE   0
(1)  014311     000                     .BYTE   0
(1)  014312  000777             22$:    BR      22$             ;;APT ERROR LOOP
(1)  014314  005777  164620     2$:     TST     @SWR            ;;HALT ON ERROR
(1)  014320  100002                     BPL     3$              ;;SKIP IF CONTINUE
(1)  014322  000000                     HALT                    ;;HALT ON ERROR!
(1)  014324  104407                     CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
(1)  014326  032777  001000  164604     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
(1)  014334  001402                     BEQ     4$              ;;BR IF NO
(1)  014336  016716  164546             MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
(1)  014342  005767  164614     4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
(1)  014346  001402                     BEQ     5$              ;;BR IF NONE
(1)  014350  016716  164606             MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
(1)  014354                     5$:
(1)  014354  022737  012446  000042     CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
(1)  014362  001001                     BNE     6$              ;;BRANCH IF NO
(1)  014364  000000                     HALT                    ;;YES
(1)  014366                     6$:
(1)  014366  000002                     RTI                     ;;RETURN
```

```
11625
11626                                   .SBTTL  SCOPE HANDLER ROUTINESTARS
  (1)                                   ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
  (1)                                   ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
  (1)                                   ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
  (1)                                   ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
  (1)                                   ;*SW14=1          LOOP ON TEST
  (1)                                   ;*SW11=1          INHIBIT ITERATIONS
  (1)                                   ;*SW09=1          LOOP ON ERROR
  (1)                                   ;*SW08=1          LOOP ON TEST IN SWR<7:0>
  (1)                                   ;*CALL
  (1)                                   ;*      SCOPE             ;;SCOPE=IOT
  (1)
  (1)  014370                           $SCOPE:
  (1)  014370  104407                           CKSWR                       ;;TEST  FOR CHANGE IN SOFT-SWR
  (1)  014372  032777  040000  164540   1$:     BIT     #BIT14,@SWR         ;;LOOP ON PRESENT TEST?
  (1)  014400  001114                           BNE     $OVER               ;;YES IF SW14=1
  (1)                                   ;#####START OF CODE FOR THE XOR TESTER#####
  (1)  014402  000416                   $XTSTR: BR      6$                  ;;IF RUNNING ON THE "XOR" TESTER CHANGE
  (1)                                                                       ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
  (1)  014404  013746  000004                   MOV     @#ERRVEC,-(SP)      ;;SAVE THE CONTENTS OF THE ERROR VECTOR
  (1)  014410  012737  014430  000004           MOV     #5$,@#ERRVEC        ;;SET FOR TIMEOUT
  (1)  014416  005737  177060                   TST     @#177060            ;;TIME OUT ON XOR?
  (1)  014422  012637  000004                   MOV     (SP)+,@#ERRVEC      ;;RESTORE THE ERROR VECTOR
  (1)  014426  000463                           BR      $SVLAD              ;;GO TO THE NEXT TEST
  (1)  014430  022626                   5$:     CMP     (SP)+,(SP)+         ;;CLEAR THE STACK AFTER A TIME OUT
  (1)  014432  012637  000004                   MOV     (SP)+,@#ERRVEC      ;;RESTORE THE ERROR VECTOR
  (1)  014436  000423                           BR      7$                  ;;LOOP ON THE PRESENT TEST
  (1)  014440                           6$:;#####END OF CODE FOR THE XOR TESTER#####
  (1)  014440  032777  000400  164472           BIT     #BIT08,@SWR         ;;LOOP ON SPEC. TEST?
  (1)  014446  001404                           BEQ     2$                  ;;BR IF NO
  (1)  014450  127767  164464  164424           CMPB    @SWR,$TSTNM         ;;ON THE RIGHT TEST?    SWR<7:0>
  (1)  014456  001465                           BEQ     $OVER               ;;BR IF YES
  (1)  014460  105767  164417           2$:     TSTB    $ERFLG              ;;HAS AN ERROR OCCURRED?
  (1)  014464  001421                           BEQ     3$                  ;;BR IF NO
  (1)  014466  126767  164423  164407           CMPB    $ERMAX,$ERFLG       ;;MAX. ERRORS FOR THIS TEST OCCURRED?
  (1)  014474  101015                           BHI     3$                  ;;BR IF NO
  (1)  014476  032777  001000  164434           BIT     #BIT09,@SWR         ;;LOOP ON ERROR?
  (1)  014504  001404                           BEQ     4$                  ;;BR IF NO
  (1)  014506  016767  164376  164372   7$:     MOV     $LPERR,$LPADR       ;;SET LOOP ADDRESS TO LAST SCOPE
  (1)  014514  000446                           BR      $OVER
  (1)  014516  105067  164361           4$:     CLRB    $ERFLG              ;;ZERO THE ERROR FLAG
  (1)  014522  005067  164432                   CLR     $TIMES              ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
  (1)  014526  000415                           BR      1$                  ;;ESCAPE TO THE NEXT TEST
  (1)  014530  032777  004000  164402   3$:     BIT     #BIT11,@SWR         ;;INHIBIT ITERATIONS?
  (1)  014536  001011                           BNE     1$                  ;;BR IF YES
  (1)  014540  005767  164436                   TST     $PASS               ;;IF FIRST PASS OF PROGRAM
  (1)  014544  001406                           BEQ     1$                  ;;      INHIBIT ITERATIONS
  (1)  014546  005267  164332                   INC     $ICNT               ;;INCREMENT ITERATION COUNT
  (1)  014552  026767  164402  164324           CMP     $TIMES,$ICNT        ;;CHECK THE NUMBER OF ITERATIONS MADE
  (1)  014560  002024                           BGE     $OVER               ;;BR IF MORE ITERATION REQUIRED
  (1)  014562  012767  000001  164314   1$:     MOV     #1,$ICNT            ;;REINITIALIZE THE ITERATION COUNTER
  (1)  014570  016767  000052  164362           MOV     $MXCNT,$TIMES       ;;SET NUMBER OF ITERATIONS TO DO
  (1)  014576  105267  164300           $SVLAD: INCB    $TSTNM              ;;COUNT TEST NUMBERS
  (1)  014602  116767  164274  164370           MOVB    $TSTNM,$TESTN       ;;SET TEST NUMBER IN APT MAILBOX
  (1)  014610  011667  164272                   MOV     (SP),$LPADR         ;;SAVE SCOPE LOOP ADDRESS
```

```
(1)  014614  011667  164270                MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
(1)  014620  005067  164336                CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
(1)  014624  112767  000001  164263        MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
(1)  014632  016777  164244  164302  $OVER: MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
(1)  014640  016716  164242                MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
(1)  014644  000002                        RTI                     ;;FIXES PS
(1)  014646  003720                $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
```

```
 11628
 11629                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
   (1)
   (2)                                  ;;**********************************************************************
   (1)                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
   (1)                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
   (1)                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
   (1)                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
   (1)                                  ;*REPLACED WITH SPACES.
   (1)                                  ;*CALL:
   (1)                                  ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
   (1)                                  ;*      TYPDS                   ;;GO TO THE ROUTINE
   (1)
   (1)  014650                          $TYPDS:
   (3)  014650  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
   (3)  014652  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
   (3)  014654  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
   (3)  014656  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
   (3)  014660  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
   (1)  014662  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
   (1)  014666  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
   (1)  014672  100004                          BPL     1$              ;;BR IF INPUT IS POS.
   (1)  014674  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
   (1)  014676  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
   (1)  014704  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
   (1)  014706  012703  015064                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
   (1)  014712  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
   (1)  014716  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
   (1)  014720  016001  015054                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
   (1)  014724  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
   (1)  014726  002402                          BLT     4$              ;;BR IF DONE
   (1)  014730  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
   (1)  014732  000774                          BR      3$
   (1)  014734  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
   (1)  014736  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
   (1)  014740  001002                          BNE     5$              ;;FALL THROUGH IF 0
   (1)  014742  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
   (1)  014744  100407                          BMI     7$              ;;BR IF YES
   (1)  014746  106316                  5$:     ASLB    (SP)            ;;MSD?
   (1)  014750  103003                          BCC     6$              ;;BR IF NO
   (1)  014752  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
   (1)  014760  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
   (1)  014764  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
   (1)  014770  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
   (1)  014772  005720                          TST     (R0)+           ;;JUST INCREMENTING
   (1)  014774  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
   (1)  015000  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
   (1)  015002  003002                          BGT     8$              ;;GO TO EXIT
   (1)  015004  010502                          MOV     R5,R2           ;;GET THE LSD
   (1)  015006  000764                          BR      6$              ;;GO CHANGE TO ASCII
   (1)  015010  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
   (1)  015012  100003                          BPL     9$              ;;BR IF NO
   (1)  015014  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
   (1)  015022  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
   (3)  015024  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
   (3)  015026  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
```

```
   (3)  015030  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
   (3)  015032  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
   (3)  015034  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
   (1)  015036  104401  015064                TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
   (1)  015042  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
   (1)  015050  012616                        MOV     (SP)+,(SP)
   (1)  015052  000002                        RTI                     ;;RETURN TO USER
   (1)  015054  023420                $DTBL:  10000.
   (1)  015056  001750                        1000.
   (1)  015060  000144                        100.
   (1)  015062  000012                        10.
   (1)  015064  000004                $DBLK:  .BLKW   4
```

```
11631
11632                                   .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
  (1)
  (2)                                   ;;**********************************************************************
  (1)                                   ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
  (1)                                   ;*OCTAL (ASCII) NUMBER AND TYPE IT.
  (1)                                   ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
  (1)                                   ;*CALL:
  (1)                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
  (1)                                   ;*      TYPOS                   ;;CALL FOR TYPEOUT
  (1)                                   ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
  (1)                                   ;*      .BYTE   M               ;;M=1 OR 0
  (1)                                   ;*                                      ;;1=TYPE LEADING ZEROS
  (1)                                   ;*                                      ;;0=SUPPRESS LEADING ZEROS
  (1)                                   ;*
  (1)                                   ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
  (1)                                   ;*$TYPOS OR $TYPOC
  (1)                                   ;*CALL:
  (1)                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
  (1)                                   ;*      TYPON                   ;;CALL FOR TYPEOUT
  (1)                                   ;*
  (1)                                   ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
  (1)                                   ;*CALL:
  (1)                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
  (1)                                   ;*      TYPOC                   ;;CALL FOR TYPEOUT
  (1)
  (1)  015074  017646  000000           $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
  (1)  015100  116667  000001  000211           MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
  (1)  015106  112667  000207                   MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
  (1)  015112  062716  000002                   ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
  (1)  015116  000406                           BR      $TYPON
  (1)  015120  112767  000001  000171  $TYPOC:  MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
  (1)  015126  112767  000006  000165           MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
  (1)  015134  112767  000005  000154  $TYPON:  MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
  (1)  015142  010346                           MOV     R3,-(SP)        ;;SAVE R3
  (1)  015144  010446                           MOV     R4,-(SP)        ;;SAVE R4
  (1)  015146  010546                           MOV     R5,-(SP)        ;;SAVE R5
  (1)  015150  116704  000145                   MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
  (1)  015154  005404                           NEG     R4
  (1)  015156  062704  000006                   ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
  (1)  015162  110467  000132                   MOVB    R4,$OMODE       ;;SAVE IT FOR USE
  (1)  015166  116704  000125                   MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
  (1)  015172  016605  000012                   MOV     12(SP),R5       ;,PICKUP THE INPUT NUMBER
  (1)  015176  005003                           CLR     R3              ;;CLEAR THE OUTPUT WORD
  (1)  015200  006105                   1$:     ROL     R5              ;;ROTATE MSB INTO "C"
  (1)  015202  000404                           BR      3$              ;;GO DO MSB
  (1)  015204  006105                   2$:     ROL     R5              ;;FORM THIS DIGIT
  (1)  015206  006105                           ROL     R5
  (1)  015210  006105                           ROL     R5
  (1)  015212  010503                           MOV     R5,R3
  (1)  015214  006103                   3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
  (1)  015216  105367  000076                   DECB    $OMODE          ;;TYPE THIS DIGIT?
  (1)  015222  100016                           BPL     7$              ;;BR IF NO
  (1)  015224  042703  177770                   BIC     #177770,R3      ;;GET RID OF JUNK
  (1)  015230  001002                           BNE     4$              ;;TEST FOR 0
  (1)  015232  005704                           TST     R4              ;;SUPPRESS THIS 0?
```

```
(1)  015234  001403                        BEQ     5$              ;;BR IF YES
(1)  015236  005204                 4$:    INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
(1)  015240  052703  000060                BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
(1)  015244  052703  000040         5$:    BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
(1)  015250  110367  000040                MOVB    R3,8$           ;;SAVE FOR TYPING
(1)  015254  104401  015314                TYPE    ,8$             ;;GO TYPE THIS DIGIT
(1)  015260  105367  000032         7$:    DECB    $OCNT           ;;COUNT BY 1
(1)  015264  003347                        BGT     2$              ;;BR IF MORE TO DO
(1)  015266  002402                        BLT     6$              ;;BR IF DONE
(1)  015270  005204                        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
(1)  015272  000744                        BR      2$              ;;GO DO THE LAST DIGIT
(1)  015274  012605                 6$:    MOV     (SP)+,R5        ;;RESTORE R5
(1)  015276  012604                        MOV     (SP)+,R4        ;;RESTORE R4
(1)  015300  012603                        MOV     (SP)+,R3        ;;RESTORE R3
(1)  015302  016666  000002  000004        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
(1)  015310  012616                        MOV     (SP)+,(SP)
(1)  015312  000002                        RTI                     ;;RETURN
(1)  015314     000                 8$:    .BYTE   0               ;;STORAGE FOR ASCII DIGIT
(1)  015315     000                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
(1)  015316     000                 $OCNT: .BYTE   0               ;;OCTAL DIGIT COUNTER
(1)  015317     000                 $OFILL:.BYTE   0               ;;ZERO FILL SWITCH
(1)  015320  000000                 $OMODE:.WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
11634
11635                                   .SBTTL  TRAP DECODER
  (1)
  (2)                                   ;;**********************************************************************
  (1)                                   ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
  (1)                                   ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
  (1)                                   ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
  (1)                                   ;*GO TO THAT ROUTINE.
  (1)
  (1)  015322  010046                   $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
  (1)  015324  016600  000002                   MOV     2(SP),R0        ;;GET TRAP ADDRESS
  (1)  015330  005740                           TST     -(R0)           ;;BACKUP BY 2
  (1)  015332  111000                           MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
  (1)  015334  006300                           ASL     R0              ;;POSITION FOR INDEXING
  (1)  015336  016000  015356                   MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
  (1)  015342  000200                           RTS     R0              ;;GO TO ROUTINE
  (1)
  (1)
  (1)                                   ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
  (1)
  (1)  015344  011646                   $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
  (1)  015346  016666  000004  000002           MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
  (1)  015354  000002                           RTI                     ;;RESTORE THE PSW
  (1)
  (3)                                   .SBTTL  TRAP TABLE
  (3)
  (3)                                   ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
  (3)                                   ;*BY THE "TRAP" INSTRUCTION.
  (3)
  (3)                                   ;       ROUTINE
  (3)                                   ;       -------
  (3)  015356  015344                   $TRPAD: .WORD   $TRAP2
  (3)  015360  012664                           $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
  (3)  015362  015120                           $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
  (3)  015364  015074                           $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
  (3)  015366  015134                           $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
  (3)  015370  014650                           $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
  (1)
  (3)  015372  013216                           $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
  (1)
  (3)  015374  013146                           $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
  (3)  015376  013430                           $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
  (3)  015400  013550                           $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
11636                                   ;;THE FOLLOWING MACRO CALL TO CNMAC2.SML WAS ADDED TO INIT
11637                                   ;;SBC11/21 SPECIFIC VECTORS
11638           015402                  POINT=.                         ;SAVE POINTER
  (1)           000100                  .=100
  (1)  000100  015402                   $CLKVEC                         ;LKVEC  HANDLER
  (1)  000102  000300                   300                             ;INTERRUPT HANDLER PRI
  (1)           000140                  .=140                           ;BRKVEC
  (1)  000140  170000                   170000                          ;ODT START ADDRESS
  (1)  000142  000300                   300                             ;PRIORITY
  (1)           015402                  .=POINT                         ;RESTORE POINTER
  (1)  015402  104401  015410           $CLKVEC:        TYPE,CLKMES
  (1)  015406  000000                                   HALT
  (1)  015410  005015  045514  042526   CLKMES:  .ASCIZ <15><12>/LKVEC INTERRUPT - DISCONNECT LTC /
```

```
   (1)  015416  020103  047111  042524
   (1)  015424  051122  050125  020124
   (1)  015432  020055  044504  041523
   (1)  015440  047117  042516  052103
   (1)  015446  046040  041524  000040
 11639          000001                .END
```

```
ABASE = 175610          7969#   9049
ACDW1 = 000000          9049
ACDW2 = 000000          9049
ACPUOP= 000000          9049
ADDRES  012352         11581*  11586*  11591   11596#
ADDW0 = 000000          9049
ADDW1 = 000000          9049
ADDW10= 000000          9049
ADDW11= 000000          9049
ADDW12= 000000          9049
ADDW13= 000000          9049
ADDW14= 000000          9049
ADDW15= 000000          9049
ADDW2 = 000000          9049
ADDW3 = 000000          9049
ADDW4 = 000000          9049
ADDW5 = 000000          9049
ADDW6 = 000000          9049
ADDW7 = 000000          9049
ADDW8 = 000000          9049
ADDW9 = 000000          9049
ADEVCT= 000000          9049
ADEVM = 000001          7970#   9049
AENV  = 000000          9049
AENVM = 000000          9049
AFATAL= 000000          9049
AMADR1= 000000          9049
AMADR2= 000000          9049
AMADR3= 000000          9049
AMADR4= 000000          9049
AMAMS1= 000000          9049
AMAMS2= 000000          9049
AMAMS3= 000000          9049
AMAMS4= 000000          9049
AMSGAD= 000000          9049
AMSGLG= 000000          9049
AMSGTY= 000000          9049
AMTYP1= 000000          9049
AMTYP2= 000000          9049
AMTYP3= 000000          9049
AMTYP4= 000000          9049
APASS = 000000          9049
APRIOR= 000000          9049
APTCSU= 000040         11614   11620#
APTENV= 000001         10300   10458   10641   11104   11614   11620#  11623
APTSIZ= 000200          9062   11620#
APTSPO= 000100         11614   11620#
ASWREG= 000000          9049
ATESTN= 000000          9049
AUNIT = 000000          9049
AUSWR = 071110          7972#   9049
AVECT1= 000300          7971#   9049
AVECT2= 000000          9049
BADBRK  011322         11295#
BAUD  = 007400          9040#
BAUDRA  007210         10722#
```

```
BITMAS  012346          11570   11579*  11585*  11589   11594#
BIT0   = 000001          8927#
BIT00  = 000001          8927#   8982    9003    9026
BIT01  = 000002          8927#   8961    8981    9025
BIT02  = 000004          8927#   8960    8980    9001    9024
BIT03  = 000010          8927#   8959    8979    9023
BIT04  = 000020          8927#   8978    9022
BIT05  = 000040          8927#   8957    8977    9021
BIT06  = 000100          8927#   8956    8976    8997    9020
BIT07  = 000200          8927#   8955    8975    8996    9019
BIT08  = 000400          8927#   11626
BIT09  = 001000          8927#   11623   11626
BIT1   = 000002          8927#
BIT10  = 002000          8927#   8952    11623
BIT11  = 004000          8927#   8951    8991    11626
BIT12  = 010000          8927#   8950    8970    8990
BIT13  = 020000          8927#   8949    8969    8989    11623
BIT14  = 040000          8927#   8948    8968    8988    11626
BIT15  = 100000          8927#   8947    8967    8987
BIT2   = 000004          8927#
BIT3   = 000010          8927#
BIT4   = 000020          8927#
BIT5   = 000040          8927#
BIT6   = 000100          8927#
BIT7   = 000200          8927#
BIT8   = 000400          8927#
BIT9   = 001000          8927#
BPTVEC= 000014           8927#
BREAK  = 000001          9003#   9154    9162    9164    9173    9175    9184    9187    11280
BRK    = 010000          9041#   9148    11269
BRKVEC= 000140           8927#
CABLE  = 020000          9042#   9729    9779    9854    9927    10000   10075   10155   10232   10903   11031
CARDET= 010000           8950#   9797    9811    9825
CKSWR  = 104407          11623   11626   11635#
CLKMES  015410           11638#
CLR    = 000000          8941#   10462   11373
CLRSEN= 020000           8949#   9870    9884    9898
COMSPD= 000100           9035#
CR     = 000015          8927#   11614
CRLF   = 000200          8927#   9063    11306   11534   11614
CYCLE   012220           9070    11552#
DATA   = 000017          9032#   11442
DATAIE= 000040           8957#   9512    9520    9522    9531    9533    9542    9545    10910   10924   10941
DATAIN= 100000           8947#   9699    10101   10113   10118   10129   10182   10196   10209   10261   10273
DATLNG  011646           10978   11049   11108   11419#
DDISP  = 177570          8927#   9049    9062
DISPLA  001142           9049#   9062*   11623*  11626*
DISPRE  000174           9045#   9062
DLADD   001254           9051#   9073*   9076    9078    9079    9080    9081    9082    9105    9122    11307   11542
DLADDR= 175610           8937#   9051    9053    9054    9055    9056    9057
DLVEC   001256           9052#   9075*   10753   10845   10945   11134   11309   11544
DSWR   = 177570          8927#   9049    9062
DTR    = 000002          8961#   9345    9347    9356    9357    9365    9367    9795    9809    9823    9868    9882    9896
                         10095   10112   10128
EMTVEC= 000030           8927#   9062*
ERRCNT  011022           11151*  11165   11168   11178#  11226   11232*
```

```
ERROR = 100000          8967#  10569  10609
ERRVEC= 000004          8927#   9062* 11626*
EVENOD= 000040          9034#
FALSE = 000000         11361# 11365
FLAG    011640         11365* 11381* 11384  11396  11403#
FRERR = 020000          8969#
FRFD  = 040000          9043#  9340   9391   9779   9854   9927  10075  10155
GNS   = ****** U        9045   9063  11306  11308  11310  11534  11536  11539  11541  11543  11635
GTSWR = 104406          9063  11635#
HOLD    011066         11194* 11196  11205#
HOLDSC  011641         11373* 11375* 11379  11404#
HOWLON= 000000         11331# 11364
HT    = 000011          8927# 11614
I       001272          9058#  9108*  9121*  9122   9123  11444*
ILLMEM= 000004          8928#  9107   9124
INITFL  012350          9068* 11571  11572* 11595#
INTFLA  012034          9112*  9116  10750* 10780  10782  10795* 10810  10848* 10878  10880  10908* 10930  10932
                       11522* 11524#
INTSRV  012026          9107  10757  10845  11505#
IOTVEC= 000020          8927#  9062*
LF    = 000012          8927# 11614
LKVEC = 000100          8927#
LOOP    001744          9069# 11313
MAINT = 000004          9001#  9211   9219   9221   9230   9232   9241   9244  10366  10411  10461  10647  10850
                       10890  10964  11014  11036  11155  11273
MASK  = 000000         11419# 11441* 11445* 11446* 11448*
MAX     006422         10475  10476  10521#
MYTYPE  012036         11528# 11623
NEW     007242         10658* 10659* 10666* 10667* 10672  10676  10687  10688  10724#
NUMBER  011024         11165  11179#  11198  11236
NUMBR   011742         11442* 11444   11450#
ODTST = 170000          8927#
OLD     007246         10645* 10646* 10672  10676  10687* 10688* 10725#
ORERR = 040000          8968#  10558  10583  10600
PARITY= 000020          9033#
PBAUDS= 004000          8991#
PBAUD0= 010000          8990#
PBAUD1= 020000          8989#
PBAUD2= 040000          8988#
PBAUD3= 100000          8987#
PBR   = 000200          9036#  10637
PERR  = 010000          8970#
PIRQ  = 177772          8927#
PIRQVE= 000240          8927#
POINT = 015402         11638#
PR0   = 000000          8927#  10767  10803  10859  10917  10972  11043  11128
PR1   = 000040          8927#
PR2   = 000100          8927#
PR3   = 000140          8927#
PR4   = 000200          8927#
PR5   = 000240          8927#
PR6   = 000300          8927#   9062   9107   9266  10086  10166  10242  10757  10845  11137  11141  11611
PR7   = 000340          8927#
PS    = 177776          8927#
PSW   = 177776          8927#
PWRVEC= 000024          8927#   9062* 11611*
```

```
RATES   007170          10652   10692   10696#
RBUF    001262           9054#   9079*  10433   10480   10513   10555   10583   10600   10609   10650   10875   10980   10995
                        11050   11064   11219   11276   11289
RCSR    001260           9053#   9078*   9345*   9347    9356*   9357    9365*   9367    9398    9406*   9408    9417*   9419
                         9428*   9431    9455    9463*   9465    9474*   9476    9485*   9488    9512    9520*   9522    9531*
                         9533    9542*   9545    9569    9577*   9579    9588*   9590    9599*   9602    9636    9699    9738
                         9795*   9797    9809*   9811    9823*   9825    9868*   9870    9882*   9884    9896*   9898    9942*
                         9944    9956*   9958    9970*   9972   10014*  10016   10028*  10030   10042*  10044   10095*  10099
                        10101   10112*  10113   10118   10128*  10129   10176*  10180   10182   10193*  10196   10206*  10209
                        10251*  10252   10258*  10261   10270*  10273   10376   10391   10421   10435   10470   10489   10491
                        10492   10503   10515   10661   10852*  10870   10872*  10892*  10910*  10922*  10924*  10925*  10941*
                        10942*  10993   11061   11160*  11174*  11238*  11283
RCVRAC= 004000           8951#   9738   10470   10489   10492
RCVRDO= 000200           8955#   9636   10376   10391   10421   10435   10491   10503   10515   10661   10870   10993   11061
                        11283
RCVRIE= 000100           8956#   9569    9577    9579    9588    9590    9599    9602   10852   10872   10892   11160   11238
RDATA0= 000001           8982#
RDATA1= 000002           8981#
RDATA2= 000004           8980#
RDATA3= 000010           8979#
RDATA4= 000020           8978#
RDATA5= 000040           8977#
RDATA6= 000100           8976#
RDATA7= 000200           8975#
RDCHR = 104410          11617   11635#
RDLIN = 104411          11635#
REC     011072          11136   11213#
REG   = 000004          11331#  11363
REGSAV  011634          11363*  11372   11401#
REQSEN= 000004           8960#   9398    9406    9408    9417    9419    9428    9431    9942    9956    9970   10176   10193
                        10206   10922   10925   10942
RESVEC= 000010           8927#
RHLD    011162          11221*  11224   11228   11245#
RING  = 040000           8948#   9944    9958    9972
R0050   007170          10705#
R0070   007171          10706#
R0110   007172          10707#
R0135   007173          10708#
R0150   007174          10709#
R0200   007177          10712#
R0300   007175          10710#
R0600   007176          10711#
R10000  007207          10720#
R1800   007200          10713#
R2000   007201          10714#
R2400   007202          10715#
R3600   007203          10716#
R4800   007204          10717#
R5STAC  001334           9060#   9083
R7200   007205          10718#
R9600   007206          10719#
SB      011026          11180#  11228*
SECREC= 002000           8952#  10016   10030   10044
SECXMI= 000010           8959#   9455    9463    9465    9474    9476    9485    9488   10014   10028   10042   10251   10258
                        10270
SET   = 177777           8940#  10314   10341   10376   10421   10471   10475   10870   10988   10993   11057   11061   11283
```

```
                    11375
SETCLR= 000006      11331#  11379
SPECIA  012250      11576#  11608
STACK = 001100       8927#   9062
START   001336       9045    9062#  11611
STKLMT= 177774       8927#
SWR     001140       9049#   9062*   9063   11611*  11617*  11623   11626
SWREG   000176       9045#   9062    9063   11617
SW0   = 000001       8927#
SW00  = 000001       8927#
SW01  = 000002       8927#
SW02  = 000004       8927#
SW03  = 000010       8927#
SW04  = 000020       8927#
SW05  = 000040       8927#
SW06  = 000100       8927#
SW07  = 000200       8927#
SW08  = 000400       8927#
SW09  = 001000       8927#
SW1   = 000002       8927#
SW10  = 002000       8927#
SW11  = 004000       8927#
SW12  = 010000       8927#
SW13  = 020000       8927#
SW14  = 040000       8927#
SW15  = 100000       8927#
SW2   = 000004       8927#
SW3   = 000010       8927#
SW4   = 000020       8927#
SW5   = 000040       8927#
SW6   = 000100       8927#
SW7   = 000200       8927#
SW8   = 000400       8927#
SW9   = 001000       8927#
TBITVE= 000014       8927#
TBUF    001270       9057#   9082*  10310*  10329*  10371*  10417*  10468*  10545*  10550*  10596*  10656*  10866*  10991*
                    11060*  11196*  11282*
TCSR    001264       9055#   9080*   9154    9162*   9164    9173*   9175    9184*   9187    9211    9219*   9221    9230*
                     9232    9241*   9244    9279    9287*   9289    9298*   9300    9309*   9312    9668   10314   10333
                    10341   10366*  10411*  10461*  10647*  10760*  10774*  10797*  10850*  10890*  10964*  10988   11014*
                    11036*  11057   11155*  11158*  11173*  11200*  11273*  11280*
TCSRHI  001266       9056#   9081*  10652*  10692*
TDATA0= 000001       9026#
TDATA1= 000002       9025#
TDATA2= 000004       9024#
TDATA3= 000010       9023#
TDATA4= 000020       9022#
TDATA5= 000040       9021#
TDATA6= 000100       9020#
TDATA7= 000200       9019#
TIME  = 000000      11464#  11481
TIMER   011470      10314   10341   10376   10421   10870   10988   10993   11057   11061   11283   11331#
TIMSAV  011636      11364*  11384   11389*  11402#
TKVEC = 000060       8927#
TPVEC = 000064       8927#
TRAN    011030      11140   11188#
```

```
TRAPVE= 000034          8927#   9062*
TRTVEC= 000014          8927#
TRUE  = 000001         11360#  11381   11384   11396
TST1    002060          9100#
TST10   003412          9504#
TST11   003554          9561#
TST12   003716          9625#
TST13   003756          9657#
TST14   004016          9689#
TST15   004056          9722#
TST16   004136          9731    9773#
TST17   004276          9783    9848#
TST2    002226          9142#
TST20   004436          9856    9921#
TST21   004576          9930    9994#
TST22   004736         10002   10069#
TST23   005142         10078   10149#
TST24   005366         10158   10214   10226#
TST25   005554         10234   10293#
TST26   005742         10301   10359#
TST27   006056         10404#
TST3    002410          9149    9202#
TST30   006166         10451#
TST31   006424         10459   10481   10497   10520   10533#
TST32   006670         10563   10576   10588   10606   10615   10632#
TST33   007252         10638   10642   10694   10744#
TST34   007524         10838#
TST35   010146         10905   10958#
TST36   010362         11009   11026#
TST37   010610         11033   11074   11096#
TST4    002552          9260#
TST40   011166         11105   11176   11264#
TST41   011350         11270   11294   11305#
TST5    002726          9335#
TST6    003066          9341    9386#
TST7    003250          9392    9447#
TYPDS = 104405         11311   11608   11635#
TYPE  = 104401          9063   11306   11308   11310   11534   11536   11539   11541   11543   11608   11611   11614   11617
                       11623   11629   11632   11635#  11638
TYPOC = 104402         11307   11309   11535   11538   11540   11542   11544   11617   11635#
TYPON = 104404         11635#
TYPOS = 104403         11635#
VECTOR  012354         11582*  11587*  11597#
WAIT    011746         10097   10178   10195   10208   10260   10272   10547   10552   10598   10777   10809   10874   10927
                       11388   11464#
WAS     011027         11181#  11229*
WHICHB= 000002         11331#  11372
XMITIE= 000100          8997#   9279    9287    9289    9298    9300    9309    9312   10760   10774   10797   11158   11173
                       11174   11200
XMITRD= 000200          8996#   9668   10314   10333   10341   10988   11057
ZZZ     011070         11203   11207#
ZZZZ    011164         11243   11246#
$APTHD  001000          9048#
$ASTAT= ****** U       11620
$ATYC   013746         11620#
$ATY1   013722         11620#
```

```
$ATY3   013730          11614   11620#
$ATY4   013740          11620#  11623
$AUTOB  001134           9049#   9063*  11617
$BASE   001250           9049#  11581
$BDADR  001122           9049#
$BDDAT  001126           9049#
$BELL   001164           9049#  11623
$BGNLE= 177777           7962#
$CHARC  013142          11614#*
$CKSWR  013146          11617#  11635
$CLKVE  015402          11638#
$CMTAG  001100           9049#   9062
$CM3  = 000000           9049#
$CNTLG  013673          11617#
$CNTLU  013666          11617#
$CPUOP  001222           9049#
$CRLF   001171           9049#  11614   11617   11623
$DBLK   015064          11629#
$DEVCT  001204           9049#  11580*  11592*
$DEVM   001252           9049#  11589
$DOAGN  012456          11608#
$DTBL   015054          11629#
$ENDAD  012446           9046    9063   11608#  11623
$ENDCT  012414           9062   11608#
$ENDMG  012465          11608#
$ENULL  012462          11608#
$ENV    001214           9049#   9063   10300   10458   10641   11104   11614   11620   11623
$ENVM   001215           9049#   9062   11614   11620
$EOP    012360          11574   11608#
$EOPCT  012406           9062*  11608#
$ERFLG  001103           9049#  11623*  11626*
$ERMAX  001115           9049#   9062*  11626*
$ERRFL= 000000           9068#   9070#   9073#   9075#   9076#   9078#   9079#   9080#   9081#   9082#   9083#   9105#   9107#
                         9108#   9110#   9112#   9121#   9122#   9124#   9149#   9152#   9161#   9162#   9171#   9173#   9182#
                         9184#   9209#   9218#   9219#   9228#   9230#   9239#   9241#   9277#   9286#   9287#   9296#   9298#
                         9307#   9309#   9341#   9344#   9345#   9354#   9356#   9364#   9365#   9392#   9396#   9405#   9406#
                         9415#   9417#   9426#   9428#   9453#   9462#   9463#   9472#   9474#   9482#   9485#   9510#   9519#
                         9520#   9529#   9531#   9540#   9542#   9567#   9576#   9577#   9586#   9588#   9597#   9599#   9635#
                         9666#   9698#   9731#   9736#   9783#   9792#   9795#   9806#   9809#   9820#   9823#   9856#   9865#
                         9868#   9879#   9882#   9893#   9896#   9930#   9939#   9942#   9953#   9956#   9967#   9970#  10002#
                        10011#  10014#  10025#  10028#  10039#  10042#  10078#  10093#  10095#  10097#  10099#  10109#  10112#
                        10125#  10128#  10158#  10173#  10176#  10178#  10180#  10189#  10193#  10195#  10203#  10206#  10208#
                        10234#  10251#  10252#  10255#  10258#  10260#  10268#  10270#  10272#  10301#  10303#  10310#  10314#
                        10323#  10329#  10341#  10366#  10368#  10371#  10376#  10386#  10411#  10412#  10417#  10421#  10433#
                        10459#  10461#  10462#  10463#  10468#  10471#  10473#  10480#  10481#  10497#  10513#  10539#  10545#
                        10547#  10550#  10552#  10555#  10563#  10568#  10576#  10580#  10592#  10596#  10598#  10606#  10638#
                        10642#  10645#  10646#  10647#  10649#  10650#  10652#  10654#  10656#  10658#  10659#  10663#  10666#
                        10667#  10687#  10688#  10691#  10692#  10750#  10753#  10755#  10757#  10758#  10760#  10774#  10777#
                        10793#  10795#  10797#  10809#  10817#  10845#  10847#  10848#  10850#  10852#  10866#  10870#  10872#
                        10874#  10875#  10890#  10892#  10902#  10905#  10908#  10910#  10922#  10924#  10925#  10927#  10941#
                        10942#  10945#  10946#  10964#  10978#  10980#  10983#  10988#  10991#  10993#  10995#  11002#  11003#
                        11009#  11014#  11033#  11036#  11049#  11050#  11052#  11057#  11060#  11061#  11064#  11067#  11068#
                        11074#  11105#  11108#  11134#  11136#  11137#  11140#  11141#  11144#  11146#  11148#  11151#  11155#
                        11158#  11160#  11173#  11174#  11192#  11194#  11196#  11200#  11217#  11219#  11221#  11228#  11229#
                        11232#  11238#  11270#  11273#  11276#  11280#  11282#  11283#  11363#  11364#  11365#  11373#  11375#
                        11381#  11388#  11389#  11397#  11399#  11441#  11442#  11444#  11445#  11446#  11448#  11481#  11482#
```

```
                    11483#  11522#  11572#  11574#  11577#  11579#  11580#  11581#  11582#  11584#  11585#  11586#  11587#
                    11591#  11592#
$ERROR  014170       9062   11623#
$ERRPC  001116       9049#  11540   11623*
$ERRTB  001254       9049#
$ERTTL  001112       9049#  11311   11312*  11623*
$ESCAP  001162       9049#   9062*  11623   11626*
$ETABL  001214       9049#
$ETEND  001254       9048    9049#
$FATAL  001176       9049#  11537*  11538   11620*
$FFLG   014166      11620#*
$FILLC  001156       9049#  11614
$FILLS  001155       9049#  11614
$F$AND= 000310       7962#   9116    9148    9154    9164    9175    9187    9211    9221    9232    9244    9279    9289
                     9300    9312    9340    9347    9357    9367    9391    9398    9408    9419    9431    9455    9465
                     9476    9488    9512    9522    9533    9545    9569    9579    9590    9602    9636    9668    9699
                     9729    9738    9779    9797    9811    9825    9854    9870    9884    9898    9927    9944    9958
                     9972   10000   10016   10030   10044   10075   10101   10113   10118   10129   10155   10182   10196
                    10209   10232   10261   10273   10300   10333   10391   10435   10458   10470   10476   10489   10491
                    10492   10503   10515   10558   10569   10583   10600   10609   10637   10641   10660   10661   10672
                    10676   10780   10782   10810   10878   10880   10903   10930   10932   11006   11031   11071   11104
                    11168   11198   11224   11226   11236   11269   11289   11372   11379   11396   11570   11571
$F$BAD= 000401       7962#   9116    9148    9154    9164    9175    9187    9211    9221    9232    9244    9279    9289
                     9300    9312    9340    9347    9357    9367    9391    9398    9408    9419    9431    9455    9465
                     9476    9488    9512    9522    9533    9545    9569    9579    9590    9602    9636    9668    9699
                     9729    9738    9779    9797    9811    9825    9854    9870    9884    9898    9927    9944    9958
                     9972   10000   10016   10030   10044   10075   10101   10113   10118   10129   10155   10182   10196
                    10209   10232   10261   10273   10300   10333   10391   10435   10458   10470   10476   10489   10491
                    10492   10503   10515   10558   10569   10583   10600   10609   10637   10641   10660   10661   10672
                    10676   10780   10782   10810   10878   10880   10903   10930   10932   11006   11031   11071   11104
                    11168   11198   11224   11226   11236   11269   11289   11372   11379   11396   11570   11571
$F$BLA= 000170       7962#
$F$CAS= 000150       7962#
$F$DEC= 000220       7962#
$F$GOO= 000400       7962#   9116    9148    9154    9164    9175    9187    9211    9221    9232    9244    9279    9289
                     9300    9312    9340    9347    9357    9367    9391    9398    9408    9419    9431    9455    9465
                     9476    9488    9512    9522    9533    9545    9569    9579    9590    9602    9636    9668    9699
                     9729    9738    9779    9797    9811    9825    9854    9870    9884    9898    9927    9944    9958
                     9972   10000   10016   10030   10044   10075   10101   10113   10118   10129   10155   10182   10196
                    10209   10232   10261   10273   10300   10317   10333   10342   10379   10391   10423   10435   10458
                    10470   10476   10489   10491   10492   10503   10515   10558   10569   10583   10600   10609   10637
                    10641   10660   10661   10672   10676   10780   10782   10810   10878   10880   10903   10930   10932
                    11006   11031   11071   11104   11168   11198   11224   11226   11236   11269   11284   11289   11372
                    11379   11396   11570   11571
$F$IF  = 000110      7962#   9116    9119    9148    9150    9154    9157    9164    9167    9175    9178    9187    9190
                     9211    9214    9221    9224    9232    9235    9244    9247    9279    9282    9289    9292    9300
                     9303    9312    9315    9340    9342    9347    9350    9357    9360    9367    9370    9391    9393
                     9398    9401    9408    9411    9419    9422    9431    9434    9455    9458    9465    9468    9476
                     9479    9488    9491    9512    9515    9522    9525    9533    9536    9545    9548    9569    9572
                     9579    9582    9590    9593    9602    9605    9636    9643    9668    9675    9699    9708    9729
                     9732    9738    9749    9779    9784    9797    9802    9811    9816    9825    9830    9854    9857
                     9870    9875    9884    9889    9898    9903    9927    9931    9944    9949    9958    9963    9972
                     9977   10000   10003   10016   10021   10030   10035   10044   10049   10075   10079   10101   10104
                    10113   10116   10118   10121   10129   10132   10155   10159   10182   10185   10196   10199   10209
                    10212   10232   10235   10261   10264   10273   10276   10300   10302   10317   10320   10333   10336
                    10342   10345   10379   10382   10391   10394   10423   10426   10435   10438   10458   10460   10470
```

```
                    10472   10474   10476   10482   10491   10492   10498   10499   10503   10507   10515   10518   10558
                    10564   10569   10577   10583   10589   10600   10607   10609   10613   10637   10639   10641   10643
                    10661   10664   10668   10672   10674   10676   10678   10684   10685   10780   10782   10785   10789
                    10790   10810   10813   10878   10880   10884   10887   10888   10903   10906   10930   10932   10936
                    10939   10940   11006   11010   11031   11034   11071   11075   11104   11106   11168   11171   11198
                    11201   11224   11226   11230   11233   11236   11240   11269   11271   11284   11287   11289   11292
                    11372   11374   11376   11379   11382   11396   11398   11570   11571   11573   11578   11583   11588
$F$INC= 000210       7962#  10649   10690   10983   11011   11052   11077   11444   11447   11482   11483   11484   11485
$F$LOO= 000200       7962#  11370   11384   11390
$F$NAM= 000160       7962#
$F$NO = 000403       7962#   9116    9148    9154    9164    9175    9187    9211    9221    9232    9244    9279    9289
                     9300    9312    9340    9347    9357    9367    9391    9398    9408    9419    9431    9455    9465
                     9476    9488    9512    9522    9533    9545    9569    9579    9590    9602    9636    9668    9699
                     9729    9738    9779    9797    9811    9825    9854    9870    9884    9898    9927    9944    9958
                     9972   10000   10016   10030   10044   10075   10101   10113   10118   10129   10155   10182   10196
                    10209   10232   10261   10273   10300   10333   10391   10435   10458   10470   10476   10489   10491
                    10492   10503   10515   10558   10569   10583   10600   10609   10637   10641   10660   10661   10672
                    10676   10780   10782   10810   10878   10880   10903   10930   10932   11006   11031   11071   11104
                    11168   11198   11224   11226   11236   11269   11372   11570   11571
$F$OR = 000320       7962#   9116    9148    9154    9164    9175    9187    9211    9221    9232    9244    9279    9289
                     9300    9312    9340    9347    9357    9367    9391    9398    9408    9419    9431    9455    9465
                     9476    9488    9512    9522    9533    9545    9569    9579    9590    9602    9636    9668    9699
                     9729    9738    9779    9797    9811    9825    9854    9870    9884    9898    9927    9944    9958
                     9972   10000   10016   10030   10044   10075   10101   10113   10118   10129   10155   10182   10196
                    10209   10232   10261   10273   10300   10333   10391   10435   10458   10470   10476   10489   10491
                    10492   10503   10515   10558   10569   10583   10600   10609   10637   10641   10660   10661   10672
                    10676   10780   10782   10810   10878   10880   10903   10930   10932   11006   11031   11071   11104
                    11168   11198   11224   11226   11236   11269   11289   11372   11379   11396   11570   11571
$F$RTN= 000300       7962#  11331   11406   11419   11451   11464   11495   11528   11549   11552   11599
$F$SEL= 000140       7962#
$F$UNT= 000130       7962#   9109    9123   10469   10475   11164   11165   11569   11589
$F$WHI= 000120       7962#  10489   10500   10660   10670   10676
$F$YES= 000402       7962#   9116    9148    9154    9164    9175    9187    9211    9221    9232    9244    9279    9289
                     9300    9312    9340    9347    9357    9367    9391    9398    9408    9419    9431    9455    9465
                     9476    9488    9512    9522    9533    9545    9569    9579    9590    9602    9636    9668    9699
                     9729    9738    9779    9797    9811    9825    9854    9870    9884    9898    9927    9944    9958
                     9972   10000   10016   10030   10044   10075   10101   10113   10118   10129   10155   10182   10196
                    10209   10232   10261   10273   10300   10333   10391   10435   10458   10470   10476   10489   10491
                    10492   10503   10515   10558   10569   10583   10600   10609   10637   10641   10660   10661   10672
                    10676   10780   10782   10810   10878   10880   10903   10930   10932   11006   11031   11071   11104
                    11168   11198   11224   11226   11236   11269   11289   11372   11379   11396   11570   11571
$GDADR  001120       9049#
$GDDAT  001124       9049#
$GET42  012436      11608#
$GTSWR  013216      11617#  11635
$HD    = 000000      8924
$HIBTS  001000       9048#
$ICNT   001104       9049#  11626*
$IFLEV= 177777       7962#   9116#   9119#   9148#   9150#   9154#   9157#   9164#   9167#   9175#   9178#   9187#   9190#
                     9211#   9214#   9221#   9224#   9232#   9235#   9244#   9247#   9279#   9282#   9289#   9292#   9300#
                     9303#   9312#   9315#   9340#   9342#   9347#   9350#   9357#   9360#   9367#   9370#   9391#   9393#
                     9398#   9401#   9408#   9411#   9419#   9422#   9431#   9434#   9455#   9458#   9465#   9468#   9476#
                     9479#   9488#   9491#   9512#   9515#   9522#   9525#   9533#   9536#   9545#   9548#   9569#   9572#
                     9579#   9582#   9590#   9593#   9602#   9605#   9636#   9643#   9668#   9675#   9699#   9708#   9729#
                     9732#   9738#   9749#   9779#   9784#   9797#   9802#   9811#   9816#   9825#   9830#   9854#   9857#
                     9870#   9875#   9884#   9889#   9898#   9903#   9927#   9931#   9944#   9949#   9958#   9963#   9972#
```

```
                      9977#  10000#  10003#  10016#  10021#  10030#  10035#  10044#  10049#  10075#  10079#  10101#  10104#
                     10113#  10116#  10118#  10121#  10129#  10132#  10155#  10159#  10182#  10185#  10196#  10199#  10209#
                     10212#  10232#  10235#  10261#  10264#  10273#  10276#  10300#  10302#  10317#  10320#  10333#  10336#
                     10342#  10345#  10379#  10382#  10391#  10394#  10423#  10426#  10435#  10438#  10458#  10460#  10470#
                     10474#  10476#  10482#  10491#  10492#  10498#  10499#  10503#  10507#  10515#  10518#  10558#  10564#
                     10569#  10577#  10583#  10589#  10600#  10607#  10609#  10613#  10637#  10639#  10641#  10643#  10661#
                     10668#  10672#  10676#  10684#  10685#  10780#  10782#  10789#  10790#  10810#  10813#  10878#  10880#
                     10887#  10888#  10903#  10906#  10930#  10932#  10939#  10940#  11006#  11010#  11031#  11034#  11071#
                     11075#  11104#  11106#  11168#  11171#  11198#  11201#  11224#  11226#  11230#  11233#  11236#  11240#
                     11269#  11271#  11284#  11287#  11289#  11292#  11372#  11376#  11379#  11382#  11396#  11398#  11570#
                     11571#  11578#  11588#
$ILLUP  012646       11611#
$INTAG  001135        9049#  11617
$ISKO = 000001        9116#   9119   9148#   9150   9154#   9157   9164#   9167   9175#   9178   9187#   9190   9211#
                      9214    9221#  9224   9232#  9235   9244#  9247   9279#  9282   9289#  9292   9300#  9303
                      9312#   9315   9340#  9342   9347#  9350   9357#  9360   9367#  9370   9391#  9393   9398#
                      9401    9408#  9411   9419#  9422   9431#  9434   9455#  9458   9465#  9468   9476#  9479
                      9488#   9491   9512#  9515   9522#  9525   9533#  9536   9545#  9548   9569#  9572   9579#
                      9582    9590#  9593   9602#  9605   9636#  9643   9668#  9675   9699#  9708   9729#  9732
                      9738#   9749   9779#  9784   9797#  9802   9811#  9816   9825#  9830   9854#  9857   9870#
                      9875    9884#  9889   9898#  9903   9927#  9931   9944#  9949   9958#  9963   9972#  9977
                     10000#  10003   10016#  10021  10030#  10035  10044#  10049  10075#  10079  10101#  10104  10113#
                     10116   10118#  10121   10129# 10132   10155# 10159   10182# 10185   10196# 10199   10209# 10212
                     10232#  10235   10261#  10264  10273#  10276  10300#  10302  10317#  10320  10333#  10336  10342#
                     10345   10379#  10382   10391# 10394   10423# 10426   10435# 10438   10458# 10460   10470# 10474
                     10476#  10482   10491#  10499  10503#  10507  10515#  10518  10558#  10564  10569#  10577  10583#
                     10589   10600#  10607   10609# 10613   10637# 10639   10641# 10643   10661# 10668   10672# 10685
                     10780#  10790   10810#  10813  10878#  10888  10903#  10906  10930#  10940  11006#  11010  11031#
                     11034   11071#  11075   11104# 11106   11168# 11171   11198# 11201   11224# 11233   11236# 11240
                     11269#  11271   11284#  11287  11289#  11292  11372#  11376  11379#  11382  11396#  11398  11570#
                     11588
$ISK1 = 000001        10492#  10498   10676#  10684  10782#  10789  10880#  10887  10932#  10939  11226#  11230  11571#
                      11578
$ITEMB  001114        9049#  11537   11623*
$LF     001172        9049#  11614   11617   11623
$LFLG   014165       11620#*
$LOCTA= 177777        7962#   9109   9116   9119   9123   9148   9150   9154   9157   9164   9167   9175   9178
                      9187    9190   9211   9214   9221   9224   9232   9235   9244   9247   9279   9282   9289
                      9292    9300   9303   9312   9315   9340   9342   9347   9350   9357   9360   9367   9370
                      9391    9393   9398   9401   9408   9411   9419   9422   9431   9434   9455   9458   9465
                      9468    9476   9479   9488   9491   9512   9515   9522   9525   9533   9536   9545   9548
                      9569    9572   9579   9582   9590   9593   9602   9605   9636   9643   9668   9675   9699
                      9708    9729   9732   9738   9749   9779   9784   9797   9802   9811   9816   9825   9830
                      9854    9857   9870   9875   9884   9889   9898   9903   9927   9931   9944   9949   9958
                      9963    9972   9977   10000  10003  10016  10021  10030  10035  10044  10049  10075  10079
                     10101   10104  10113  10116  10118  10121  10129  10132  10155  10159  10182  10185  10196
                     10199   10209  10212  10232  10235  10261  10264  10273  10276  10300  10302  10317  10320
                     10333   10336  10342  10345  10379  10382  10391  10394  10423  10426  10435  10438  10458
                     10460   10469  10470  10472  10474  10475  10476  10482  10489  10491  10492  10498  10499
                     10500   10503  10507  10515  10518  10558  10564  10569  10577  10583  10589  10600  10607
                     10609   10613  10637  10639  10641  10643  10649  10660  10661  10664  10668  10670  10672
                     10674   10676  10678  10684  10685  10690  10780  10782  10785  10789  10790  10810  10813
                     10878   10880  10884  10887  10888  10903  10906  10930  10932  10936  10939  10940  10983
                     11006   11010  11011  11031  11034  11052  11071  11075  11077  11104  11106  11164  11165
                     11168   11171  11198  11201  11224  11226  11230  11233  11236  11240  11269  11271  11284
                     11287   11289  11292  11331  11370  11372  11374  11376  11379  11382  11384  11390  11396
```

```
                    11397   11398   11399   11406   11419   11444   11447   11449   11451   11464   11482   11483   11484
                    11485   11495   11528   11549   11552   11569   11570   11571   11573   11578   11583   11588   11589
                    11593   11599
$LPADR  001106       9049#   9062*  11626*
$LPERR  001110       9049#   9062*   9110*   9152*   9161*   9171*   9182*   9209*   9218*   9228*   9239*   9277*   9286*
                     9296*   9307*   9344*   9354*   9364*   9396*   9405*   9415*   9426*   9453*   9462*   9472*   9482*
                     9510*   9519*   9529*   9540*   9567*   9576*   9586*   9597*   9635*   9666*   9698*   9736*   9792*
                     9806*   9820*   9865*   9879*   9893*   9939*   9953*   9967*  10011*  10025*  10039*  10093*  10109*
                    10125*  10173*  10189*  10203*  10255*  10268*  10303*  10323*  10368*  10386*  10412*  10539*  10568*
                    10580*  10592*  10758*  10793*  10847*  10902*  11623   11626*
$LSTCN= 177777       7962#   7965#   9109    9116    9119    9123    9148    9150    9154    9157    9164    9167    9175
                     9178    9187    9190    9211    9214    9221    9224    9232    9235    9244    9247    9279    9282
                     9289    9292    9300    9303    9312    9315    9340    9342    9347    9350    9357    9360    9367
                     9370    9391    9393    9398    9401    9408    9411    9419    9422    9431    9434    9455    9458
                     9465    9468    9476    9479    9488    9491    9512    9515    9522    9525    9533    9536    9545
                     9548    9569    9572    9579    9582    9590    9593    9602    9605    9636    9643    9668    9675
                     9699    9708    9729    9732    9738    9749    9779    9784    9797    9802    9811    9816    9825
                     9830    9854    9857    9870    9875    9884    9889    9898    9903    9927    9931    9944    9949
                     9958    9963    9972    9977   10000   10003   10016   10021   10030   10035   10044   10049   10075
                    10079   10101   10104   10113   10116   10118   10121   10129   10132   10155   10159   10182   10185
                    10196   10199   10209   10212   10232   10235   10261   10264   10273   10276   10300   10302   10317
                    10320   10333   10336   10342   10345   10379   10382   10391   10394   10423   10426   10435   10438
                    10458   10460   10469   10470   10472   10474   10475   10476   10482   10489   10491   10492   10498
                    10499   10500   10503   10507   10515   10518   10558   10564   10569   10577   10583   10589   10600
                    10607   10609   10613   10637   10639   10641   10643   10649   10660   10661   10664   10668   10670
                    10672   10674   10676   10678   10684   10685   10690   10780   10782   10785   10789   10790   10810
                    10813   10878   10880   10884   10887   10888   10903   10906   10930   10932   10936   10939   10940
                    10983   11006   11010   11011   11031   11034   11052   11071   11075   11077   11104   11106   11164
                    11165   11168   11171   11198   11201   11224   11226   11230   11233   11236   11240   11269   11271
                    11284   11287   11289   11292   11331   11370   11372   11374   11376   11379   11382   11390   11396
                    11398   11406   11419   11444   11447   11451   11464   11482   11483   11484   11485   11495   11528
                    11549   11552   11569   11570   11571   11573   11578   11583   11588   11589   11599
$LSTIN= 000000        7962#   7963#   9068    9070    9073    9075    9076    9078    9079    9080    9081    9082    9083
                      9105    9107    9108    9110    9112    9116    9121    9122    9123    9124    9148    9149    9152
                      9154    9161    9162    9164    9171    9173    9175    9182    9184    9187    9209    9211    9218
                      9219    9221    9228    9230    9232    9239    9241    9244    9277    9279    9286    9287    9289
                      9296    9298    9300    9307    9309    9312    9340    9341    9344    9345    9347    9354    9356
                      9357    9364    9365    9367    9391    9392    9396    9398    9405    9406    9408    9415    9417
                      9419    9426    9428    9431    9453    9455    9462    9463    9465    9472    9474    9476    9482
                      9485    9488    9510    9512    9519    9520    9522    9529    9531    9533    9540    9542    9545
                      9567    9569    9576    9577    9579    9586    9588    9590    9597    9599    9602    9635    9636
                      9666    9668    9698    9699    9729    9731    9736    9738    9779    9783    9792    9795    9797
                      9806    9809    9811    9820    9823    9825    9854    9856    9865    9868    9870    9879    9882
                      9884    9893    9896    9898    9927    9930    9939    9942    9944    9953    9956    9958    9967
                      9970    9972   10000   10002   10011   10014   10016   10025   10028   10030   10039   10042   10044
                     10075   10078   10093   10095   10097   10099   10101   10109   10112   10113   10118   10125   10128
                     10129   10155   10158   10173   10176   10178   10180   10182   10189   10193   10195   10196   10203
                     10206   10208   10209   10232   10234   10251   10252   10255   10258   10260   10261   10268   10270
                     10272   10273   10300   10301   10303   10310   10314   10317   10323   10329   10333   10341   10342
                     10366   10368   10371   10376   10379   10386   10391   10411   10412   10417   10421   10423   10433
                     10435   10458   10459   10461   10462   10463   10468   10470   10471   10472   10473   10475   10476
                     10480   10481   10489   10491   10492   10497   10500   10503   10513   10515   10539   10545   10547
                     10550   10552   10555   10558   10563   10568   10569   10576   10580   10583   10592   10596   10598
                     10600   10606   10609   10637   10638   10641   10642   10645   10646   10647   10649   10650   10652
                     10654   10656   10658   10659   10660   10661   10663   10664   10666   10667   10670   10672   10674
                     10676   10678   10687   10688   10690   10691   10692   10750   10753   10755   10757   10758   10760
```

```
                    10774   10777   10780   10782   10785   10793   10795   10797   10809   10810   10817   10845   10847
                    10848   10850   10852   10866   10870   10872   10874   10875   10878   10880   10884   10890   10892
                    10902   10903   10905   10908   10910   10922   10924   10925   10927   10930   10932   10936   10941
                    10942   10945   10946   10964   10978   10980   10983   10988   10991   10993   10995   11002   11003
                    11006   11009   11011   11014   11031   11033   11036   11049   11050   11052   11057   11060   11061
                    11064   11067   11068   11071   11074   11077   11104   11105   11108   11134   11136   11137   11140
                    11141   11144   11146   11148   11151   11155   11158   11160   11165   11168   11173   11174   11192
                    11194   11196   11198   11200   11217   11219   11221   11224   11226   11228   11229   11232   11236
                    11238   11269   11270   11273   11276   11280   11282   11283   11284   11289   11363   11364   11365
                    11372   11373   11374   11375   11379   11381   11384   11388   11389   11390   11396   11397   11399
                    11406   11441   11442   11444   11445   11446   11447   11448   11449   11451   11481   11482   11483
                    11484   11485   11495   11522   11549   11570   11571   11572   11573   11574   11577   11579   11580
                    11581   11582   11583   11584   11585   11586   11587   11589   11591   11592   11593   11599
$LSTST= 177777       7962#   7966#   9109    9116    9119    9123    9148    9150    9154    9157    9164    9167    9175
                     9178    9187    9190    9211    9214    9221    9224    9232    9235    9244    9247    9279    9282
                     9289    9292    9300    9303    9312    9315    9340    9342    9347    9350    9357    9360    9367
                     9370    9391    9393    9398    9401    9408    9411    9419    9422    9431    9434    9455    9458
                     9465    9468    9476    9479    9488    9491    9512    9515    9522    9525    9533    9536    9545
                     9548    9569    9572    9579    9582    9590    9593    9602    9605    9636    9643    9668    9675
                     9699    9708    9729    9732    9738    9749    9779    9784    9797    9802    9811    9816    9825
                     9830    9854    9857    9870    9875    9884    9889    9898    9903    9927    9931    9944    9949
                     9958    9963    9972    9977    10000   10003   10016   10021   10030   10035   10044   10049   10075
                    10079   10101   10104   10113   10116   10118   10121   10129   10132   10155   10159   10182   10185
                    10196   10199   10209   10212   10232   10235   10261   10264   10273   10276   10300   10302   10317
                    10320   10333   10336   10342   10345   10379   10382   10391   10394   10423   10426   10435   10438
                    10458   10460   10469   10470   10472   10474   10475   10476   10482   10489   10491   10492   10498
                    10499   10500   10503   10507   10515   10518   10558   10564   10569   10577   10583   10589   10600
                    10607   10609   10613   10637   10639   10641   10643   10649   10660   10661   10664   10668   10670
                    10672   10674   10676   10678   10684   10685   10690   10780   10782   10785   10789   10790   10810
                    10813   10878   10880   10884   10887   10888   10903   10906   10930   10932   10936   10939   10940
                    10983   11006   11010   11011   11031   11034   11052   11071   11075   11077   11104   11106   11164
                    11165   11168   11171   11198   11201   11224   11226   11230   11233   11236   11240   11269   11271
                    11284   11287   11289   11292   11331   11370   11372   11374   11376   11379   11382   11384   11390
                    11396   11398   11406   11419   11444   11447   11451   11464   11482   11483   11484   11485   11495
                    11528   11549   11552   11569   11570   11571   11573   11578   11583   11588   11589   11599
$LSTTA= 000000        7962#   7964#   9109    9119    9150    9157    9167    9178    9190    9214    9224    9235    9247
                      9282    9292    9303    9315    9342    9350    9360    9370    9393    9401    9411    9422    9434
                      9458    9468    9479    9491    9515    9525    9536    9548    9572    9582    9593    9605    9643
                      9675    9708    9732    9749    9784    9802    9816    9830    9857    9875    9889    9903    9931
                      9949    9963    9977    10003   10021   10035   10049   10079   10104   10116   10121   10132   10159
                     10185   10199   10212   10235   10264   10276   10302   10320   10336   10345   10382   10394   10426
                     10438   10460   10469   10472   10474   10475   10482   10489   10498   10499   10500   10507   10518
                     10564   10577   10589   10607   10613   10639   10643   10649   10660   10664   10668   10670   10674
                     10678   10684   10685   10690   10785   10789   10790   10813   10884   10887   10888   10906   10936
                     10939   10940   10983   11010   11011   11034   11052   11075   11077   11106   11164   11165   11171
                     11201   11230   11233   11240   11271   11287   11292   11331   11370   11374   11376   11382   11390
                     11398   11406   11419   11444   11447   11451   11464   11482   11483   11484   11485   11495   11528
                     11549   11552   11569   11573   11578   11583   11588   11599
$MADR1  001226       9049#
$MADR2  001232       9049#
$MADR3  001236       9049#
$MADR4  001242       9049#
$MAIL   001174        9048    9049#   9062    9063    9100    9142    9202    9260    9335    9386    9447    9504    9561
                      9625    9657    9689    9722    9773    9848    9921    9994   10069   10149   10226   10293   10359
                     10404   10451   10533   10632   10744   10838   10958   11026   11096   11264   11614   11623   11626
$MAMS1  001224        9049#
```

```
$MAMS2  001230          9049#
$MAMS3  001234          9049#
$MAMS4  001240          9049#
$MBADR  001002          9048#
$MCALL= 000000          5717#   7962
$MFLG   014164         11620#*
$MNEW   013711         11617#
$MSGAD  001210          9049#  11620*
$MSGLG  001212          9049#  11620*
$MSGTY  001174          9049#  11620*
$MSWR   013700         11617#
$MTYP1  001225          9049#
$MTYP2  001231          9049#
$MTYP3  001235          9049#
$MTYP4  001241          9049#
$MXCNT  014646         11626#
$NESTL= 177777          7962#   9109#   9116#   9119#   9123#   9148#   9150#   9154#   9157#   9164#   9167#   9175#   9178#
                        9187#   9190#   9211#   9214#   9221#   9224#   9232#   9235#   9244#   9247#   9279#   9282#   9289#
                        9292#   9300#   9303#   9312#   9315#   9340#   9342#   9347#   9350#   9357#   9360#   9367#   9370#
                        9391#   9393#   9398#   9401#   9408#   9411#   9419#   9422#   9431#   9434#   9455#   9458#   9465#
                        9468#   9476#   9479#   9488#   9491#   9512#   9515#   9522#   9525#   9533#   9536#   9545#   9548#
                        9569#   9572#   9579#   9582#   9590#   9593#   9602#   9605#   9636#   9643#   9668#   9675#   9699#
                        9708#   9729#   9732#   9738#   9749#   9779#   9784#   9797#   9802#   9811#   9816#   9825#   9830#
                        9854#   9857#   9870#   9875#   9884#   9889#   9898#   9903#   9927#   9931#   9944#   9949#   9958#
                        9963#   9972#   9977#  10000#  10003#  10016#  10021#  10030#  10035#  10044#  10049#  10075#  10079#
                       10101#  10104#  10113#  10116#  10118#  10121#  10129#  10132#  10155#  10159#  10182#  10185#  10196#
                       10199#  10209#  10212#  10232#  10235#  10261#  10264#  10273#  10276#  10300#  10302#  10317#  10320#
                       10333#  10336#  10342#  10345#  10379#  10382#  10391#  10394#  10423#  10426#  10435#  10438#  10458#
                       10460#  10469#  10470#  10472   10474#  10475#  10476#  10482#  10489#  10491#  10492#  10498#  10499#
                       10500#  10503#  10507#  10515#  10518#  10558#  10564#  10569#  10577#  10583#  10589#  10600#  10607#
                       10609#  10613#  10637#  10639#  10641#  10643#  10649#  10660#  10661#  10664   10668#  10670#  10672#
                       10674   10676#  10678   10684#  10685#  10690#  10780#  10782#  10785   10789#  10790#  10810#  10813#
                       10878#  10880#  10884   10887#  10888#  10903#  10906#  10930#  10932#  10936   10939#  10940#  10983#
                       11006#  11010#  11011#  11031#  11034#  11052#  11071#  11075#  11077#  11104#  11106#  11164#  11165#
                       11168#  11171#  11198#  11201#  11224#  11226#  11230#  11233#  11236#  11240#  11269#  11271#  11284#
                       11287#  11289#  11292#  11331#  11370#  11372#  11374   11376#  11379#  11382#  11384   11390#  11396#
                       11398#  11406#  11419#  11444#  11447#  11451#  11464#  11482#  11483#  11484#  11485#  11495#  11528#
                       11549#  11552#  11569#  11570#  11571#  11573   11578#  11583   11588#  11589#  11599#
$NSKO = 000300          9109#   9123    9148#   9150    9154#   9157    9164#   9167    9175#   9178    9187#   9190    9211#
                        9214    9221#   9224    9232#   9235    9244#   9247    9279#   9282    9289#   9292    9300#   9303
                        9312#   9315    9340#   9342    9347#   9350    9357#   9360    9367#   9370    9391#   9393    9398#
                        9401    9408#   9411    9419#   9422    9431#   9434    9455#   9458    9465#   9468    9476#   9479
                        9488#   9491    9512#   9515    9522#   9525    9533#   9536    9545#   9548    9569#   9572    9579#
                        9582    9590#   9593    9602#   9605    9636#   9643    9668#   9675    9699#   9708    9729#   9732
                        9738#   9749    9779#   9784    9797#   9802    9811#   9816    9825#   9830    9854#   9857    9870#
                        9875    9884#   9889    9898#   9903    9927#   9931    9944#   9949    9958#   9963    9972#   9977
                       10000#  10003   10016#  10021   10030#  10035   10044#  10049   10075#  10079   10101#  10104   10113#
                       10116   10118#  10121   10129#  10132   10155#  10159   10182#  10185   10196#  10199   10209#  10212
                       10232#  10235   10261#  10264   10273#  10276   10300#  10302   10317#  10320   10333#  10336   10342#
                       10345   10379#  10382   10391#  10394   10423#  10426   10435#  10438   10458#  10460   10469#  10475
                       10476#  10482   10489#  10500   10503#  10507   10515#  10518   10558#  10564   10569#  10577   10583#
                       10589   10600#  10607   10609#  10613   10637#  10639   10641#  10643   10649#  10690   10780#  10790
                       10810#  10813   10878#  10888   10903#  10906   10930#  10940   10983#  11011   11031#  11034   11052#
                       11077   11104#  11106   11164#  11165   11168#  11171   11198#  11201   11224#  11233   11236#  11240
                       11269#  11271   11284#  11287   11289#  11292   11331#  11406   11419#  11451   11464#  11495   11528#
                       11549   11552#  11599
```

```
$NSK1 = 000130          9116#   9119   10470#  10472   10474   10491#  10499   10660#  10670   10672#  10674   10685   10782#
                        10785   10789   10880#  10884   10887   10932#  10936   10939   11006#  11010   11071#  11075   11226#
                        11230   11370#  11384   11390   11396#  11398   11444#  11447   11482#  11485   11569#  11589
$NSK2 = 000110          10492#  10498   10661#  10664   10668   10676#  10678   10684   11372#  11374   11376   11379#  11382
                        11483#  11484   11570#  11583   11588
$NSK3 = 000110          11571#  11573   11578
$NULL   001154          9049#   11614
$NWTST= 000001          9100#   9142#   9202#   9260#   9335#   9386#   9447#   9504#   9561#   9625#   9657#   9689#   9722#
                        9773#   9848#   9921#   9994#   10069#  10149#  10226#  10293#  10359#  10404#  10451#  10533#  10632#
                        10744#  10838#  10958#  11026#  11096#  11264#  11305#
$OCNT   015316          11632#*
$OMODE  015320          11632#*
$OVER   014632          11626#
$PASS   001202          9049#   9062*   11608*  11626
$PASTM  001006          9048#
$POWER  012654          11611#
$PWRAD  012642          11611#
$PWRDN  C12502          9062    11611#
$PWRMG  012636          11611#
$PWRUP  012554          11611#
$QUES   001170          9049#   11614   11617   11623
$RDCHR  013430          11617#  11635
$RDDEC= ****** U        11635
$RDLIN  013550          11617#  11635
$RDOCT= ****** U        11635
$RDSZ = 000010          11617#
$RTNAD  012460          11608#
$R2A  = ****** U        11635
$SAVLE= 177777          7962#   10500#  10649#  10670#  10983#  11052#  11390#  11444#  11482#  11483#
$SAVRE= ****** U        11635
$SAVR6  012652          11611#*
$SCOPE  014370          9062    11626#
$SETUP= 000137          9061#   9062    9063    11608   11617   11623   11626
$SSK0 = 000236          10500#  10649#  10670#  10983#  11052#  11390#  11444#  11482#  11483#
$STUP = 177777          9061#
$SVLAD  014576          11626#
$SVPC = 000204          9046#
$SWR  = 167400          7967#   8924    8925    9049    9062    9100    9142    9202    9260    9335    9386    9447    9504
                        9561    9625    9657    9689    9722    9773    9848    9921    9994    10069   10149   10226   10293
                        10359   10404   10451   10533   10632   10744   10838   10958   11026   11096   11264   11305   11608
                        11611   11623   11626
$SWREG  001216          9049#   9062
$SWRMK= 000000          8925    11626
$TAGLE= 177777          7962#   9109#   9116#   9119#   9123#   9148#   9150#   9154#   9157#   9164#   9167#   9175#   9178#
                        9187#   9190#   9211#   9214#   9221#   9224#   9232#   9235#   9244#   9247#   9279#   9282#   9289#
                        9292#   9300#   9303#   9312#   9315#   9340#   9342#   9347#   9350#   9357#   9360#   9367#   9370#
                        9391#   9393#   9398#   9401#   9408#   9411#   9419#   9422#   9431#   9434#   9455#   9458#   9465#
                        9468#   9476#   9479#   9488#   9491#   9512#   9515#   9522#   9525#   9533#   9536#   9545#   9548#
                        9569#   9572#   9579#   9582#   9590#   9593#   9602#   9605#   9636#   9643#   9668#   9675#   9699#
                        9708#   9729#   9732#   9738#   9749#   9779#   9784#   9797#   9802#   9811#   9816#   9825#   9830#
                        9854#   9857#   9870#   9875#   9884#   9889#   9898#   9903#   9927#   9931#   9944#   9949#   9958#
                        9963#   9972#   9977#   10000#  10003#  10016#  10021#  10030#  10035#  10044#  10049#  10075#  10079#
                        10101#  10104#  10113#  10116#  10118#  10121#  10129#  10132#  10155#  10159#  10182#  10185#  10196#
                        10199#  10209#  10212#  10232#  10235#  10261#  10264#  10273#  10276#  10300#  10302#  10317#  10320#
                        10333#  10336#  10342#  10345#  10379#  10382#  10391#  10394#  10423#  10426#  10435#  10438#  10458#
                        10460#  10469#  10470#  10472#  10474#  10475#  10476#  10482#  10489#  10491#  10492#  10498#  10499#
```

```
$TESTN  001200          9049# 9100* 9142* 9202* 9260* 9335* 9386* 9447* 9504* 9561* 9625* 9657* 9689*
                        9722* 9773* 9848* 9921* 9994* 10069* 10149* 10226* 10293* 10359* 10404* 10451* 10533*
                       10632* 10744* 10838* 10958* 11026* 11096* 11264* 11535 11626*
$TIMES  001160          9049# 9062* 9100* 9142* 9149* 9202* 9260* 9335* 9341* 9386* 9392* 9447* 9504*
                        9561* 9625* 9657* 9689* 9722* 9731* 9773* 9783* 9848* 9856* 9921* 9930* 9994*
                       10002* 10069* 10078* 10149* 10158* 10226* 10234* 10293* 10301* 10359* 10404* 10451* 10459*
                       10481* 10497* 10533* 10563* 10576* 10606* 10632* 10638* 10642* 10744* 10838* 10905* 10958*
                       11009* 11026* 11033* 11074* 11096* 11105* 11264* 11270* 11305* 11608* 11626*
$TKB    001146          9049# 11617
$TKS    001144          9049# 11617*
$TN   = 000042          7968# 8924  9100# 9142# 9149  9202# 9260# 9335# 9341  9386# 9392  9447# 9504#
                        9561# 9625# 9657# 9689# 9722# 9731  9773# 9783  9848# 9856  9921# 9930  9994#
                       10002  10069# 10078  10149# 10158  10214  10226# 10234  10293# 10301  10359# 10404# 10451#
                       10459  10481  10497  10520  10533# 10563  10576  10588  10606  10615  10632# 10638  10642
                       10694  10744# 10838# 10905  10958# 11009  11026# 11033  11074  11096# 11105  11176  11264#
                       11270  11294  11305#
$TPB    001152          9049# 11614*
$TPFLG  001157          9049# 11614
$TPS    001150          9049# 11614
$TRAP   015322          9062  11635#
$TRAP2  015344         11635#
$TRP  = 000012         11635#
$TRPAD  015356         11635#
$TSK0 = 000244          9109# 9123  9148# 9150  9154# 9157  9164# 9167  9175# 9178  9187# 9190  9211#
                        9214  9221# 9224  9232# 9235  9244# 9247  9279# 9282  9289# 9292  9300# 9303
                        9312# 9315  9340# 9342  9347# 9350  9357# 9360  9367# 9370  9391# 9393  9398#
                        9401  9408# 9411  9419# 9422  9431# 9434  9455# 9458  9465# 9468  9476# 9479
                        9488# 9491  9512# 9515  9522# 9525  9533# 9536  9545# 9548  9569# 9572  9579#
                        9582  9590# 9593  9602# 9605  9636# 9643  9668# 9675  9699# 9708  9729# 9732
                        9738# 9749  9779# 9784  9797# 9802  9811# 9816  9825# 9830  9854# 9857  9870#
                        9875  9884# 9889  9898# 9903  9927# 9931  9944# 9949  9958# 9963  9972# 9977
                       10000# 10003 10016# 10021 10030# 10035 10044# 10049 10075# 10079 10101# 10104 10113#
                       10116  10118# 10121  10129# 10132  10155# 10159  10182# 10185  10196# 10199  10209# 10212
                       10232# 10235  10261# 10264  10273# 10276  10300# 10302  10317# 10320  10333# 10336  10342#
                       10345  10379# 10382  10391# 10394  10423# 10426  10435# 10438  10458# 10460  10469# 10475
                       10476# 10482  10489# 10500  10503# 10507  10515# 10518  10558# 10564  10569# 10577  10583#
                       10589  10600# 10607  10609# 10613  10637# 10639  10641# 10643  10649# 10690  10780# 10790
                       10810# 10813  10878# 10888  10903# 10906  10930# 10940  10983# 11011  11031# 11034  11052#
                       11077  11104# 11106  11164# 11165  11168# 11171  11198# 11201  11224# 11233  11236# 11240
                       11269# 11271  11284# 11287  11289# 11292  11370# 11390  11396# 11398  11444# 11447  11482#
                       11485  11569# 11589
$TSK1 = 000250          9116#  9119  10470# 10472# 10474  10489# 10500  10649# 10690  10782# 10785# 10789  10880#
                       10884# 10887  10932# 10936# 10939  10983# 11011  11052# 11077  11226# 11230  11370# 11384
                       11390  11444# 11447  11482# 11485  11570# 11583# 11588
$TSK2 = 000247         10491# 10499  10660# 10670  10672# 10674# 10685  11006# 11010  11071# 11075  11372# 11374#
                       11376  11379# 11382  11483# 11484  11571# 11573# 11578
$TSK3 = 000236         10492# 10498  10660# 10670  10676# 10678# 10684  11483# 11484
$TSK4 = 000147         10661# 10664# 10668
$TSTM   001004          9048#
$TSTNM  001102          9049# 11608* 11623  11626*
$TTYIN  013656         11617#
$TYPBN= ****** U       11635
$TYPDS  014650         11629# 11635
$TYPE   012664         11614# 11620  11635
$TYPEC  013076         11614# 11617
$TYPEX  013144         11614#
```

```
$TYPOC  015120          11632#  11635
$TYPON  015134          11632#  11635
$TYPOS  015074          11632#  11635
$UNIT   001206           9049#
$UNITM  001010           9048#
$USWR   001220           9049#   9148   9340   9391   9729   9779   9854   9927  10000  10075  10155  10232  10637
                        10691   10903  11031  11269  11442
$VECT1  001244           9049#  11582
$VECT2  001246           9049#
$XTSTR  014402          11626#
$YESNO= 000001           9079#   9080#  9081#  9082#  9121#  9122#  9124#  9162#  9173#  9184#  9219#  9230#  9241#
                         9287#   9298#  9309#  9345#  9356#  9365#  9406#  9417#  9428#  9463#  9474#  9485#  9520#
                         9531#   9542#  9577#  9588#  9599#  9795#  9809#  9823#  9868#  9882#  9896#  9942#  9956#
                         9970#  10014#  10028#  10042#  10095#  10112#  10128#  10176#  10193#  10206#  10251#  10258#  10270#
                        10366#  10411#  10461#  10473#  10647#  10649#  10666#  10667#  10691#  10755#  10760#  10774#  10797#
                        10817#  10850#  10852#  10872#  10890#  10892#  10910#  10922#  10924#  10925#  10941#  10942#  10946#
                        10964#  10983#  11002#  11003#  11014#  11036#  11052#  11067#  11068#  11155#  11158#  11160#  11173#
                        11174#  11192#  11194#  11200#  11217#  11219#  11221#  11232#  11238#  11273#  11280#  11389#  11442#
                        11444#  11445#  11446#  11448#  11482#  11483#  11522#  11585#  11586#  11587#  11592#
$$ARGC= 000000          11331#  11419#  11464#  11528#  11552#
$$BYTE= 000403           9116#   9148#  9154#  9164#  9175#  9187#  9211#  9221#  9232#  9244#  9279#  9289#  9300#
                         9312#   9340#  9347#  9357#  9367#  9391#  9398#  9408#  9419#  9431#  9455#  9465#  9476#
                         9488#   9512#  9522#  9533#  9545#  9569#  9579#  9590#  9602#  9636#  9668#  9699#  9729#
                         9738#   9779#  9797#  9811#  9825#  9854#  9870#  9884#  9898#  9927#  9944#  9958#  9972#
                        10000#  10016#  10030#  10044#  10075#  10101#  10113#  10118#  10129#  10155#  10182#  10196#  10209#
                        10232#  10261#  10273#  10300#  10333#  10391#  10435#  10458#  10470#  10476#  10489#  10491#  10492#
                        10503#  10515#  10558#  10569#  10583#  10600#  10609#  10637#  10641#  10660#  10661#  10672#  10676#
                        10780#  10782#  10810#  10878#  10880#  10903#  10930#  10932#  11006#  11031#  11071#  11104#  11168#
                        11198#  11224#  11226#  11236#  11269#  11289#  11372#  11379#  11396#  11570#  11571#
$$DST = 000067          10691#  11442#
$$FLAG= 000001           9116#   9119#  9148#  9150#  9154#  9157#  9164#  9167#  9175#  9178#  9187#  9190#  9211#
                         9214#   9221#  9224#  9232#  9235#  9244#  9247#  9279#  9282#  9289#  9292#  9300#  9303#
                         9312#   9315#  9340#  9342#  9347#  9350#  9357#  9360#  9367#  9370#  9391#  9393#  9398#
                         9401#   9408#  9411#  9419#  9422#  9431#  9434#  9455#  9458#  9465#  9468#  9476#  9479#
                         9488#   9491#  9512#  9515#  9522#  9525#  9533#  9536#  9545#  9548#  9569#  9572#  9579#
                         9582#   9590#  9593#  9602#  9605#  9636#  9643#  9668#  9675#  9699#  9708#  9729#  9732#
                         9738#   9749#  9779#  9784#  9797#  9802#  9811#  9816#  9825#  9830#  9854#  9857#  9870#
                         9875#   9884#  9889#  9898#  9903#  9927#  9931#  9944#  9949#  9958#  9963#  9972#  9977#
                        10000#  10003#  10016#  10021#  10030#  10035#  10044#  10049#  10075#  10079#  10101#  10104#  10113#
                        10116#  10118#  10121#  10129#  10132#  10155#  10159#  10182#  10185#  10196#  10199#  10209#  10212#
                        10232#  10235#  10261#  10264#  10273#  10276#  10300#  10302#  10317#  10320#  10333#  10336#  10342#
                        10345#  10379#  10382#  10391#  10394#  10423#  10426#  10435#  10438#  10458#  10460#  10470#  10474#
                        10476#  10482#  10489#  10491#  10492#  10498#  10499#  10503#  10507#  10515#  10518#  10558#  10564#
                        10569#  10577#  10583#  10589#  10600#  10607#  10609#  10613#  10637#  10639#  10641#  10643#  10660#
                        10661#  10668#  10672#  10676#  10684#  10685#  10780#  10782#  10789#  10790#  10810#  10813#  10878#
                        10880#  10887#  10888#  10903#  10906#  10930#  10932#  10939#  10940#  11006#  11010#  11031#  11034#
                        11071#  11075#  11104#  11106#  11168#  11171#  11198#  11201#  11224#  11226#  11230#  11233#  11236#
                        11240#  11269#  11271#  11284#  11287#  11289#  11292#  11372#  11376#  11379#  11382#  11396#  11398#
                        11570#  11571#  11578#  11588#
$$FROM= 000000           9070#  10097#  10178#  10195#  10208#  10260#  10272#  10314#  10341#  10376#  10421#  10547#  10552#
                        10598#  10777#  10809#  10870#  10874#  10927#  10978#  10988#  10993#  11049#  11057#  11061#  11108#
                        11283#  11388#  11574#
$$GET4= 000000          11608#
$$LOC = 012332           9116#   9123#  9148#  9154#  9164#  9175#  9187#  9211#  9221#  9232#  9244#  9279#  9289#
                         9300#   9312#  9340#  9347#  9357#  9367#  9391#  9398#  9408#  9419#  9431#  9455#  9465#
                         9476#   9488#  9512#  9522#  9533#  9545#  9569#  9579#  9590#  9602#  9636#  9668#  9699#
```

```
                      9729#   9738#   9779#   9797#   9811#   9825#   9854#   9870#   9884#   9898#   9927#   9944#   9958#
                      9972#  10000#  10016#  10030#  10044#  10075#  10101#  10113#  10118#  10129#  10155#  10182#  10196#
                     10209#  10232#  10261#  10273#  10300#  10317#  10333#  10342#  10379#  10391#  10423#  10435#  10458#
                     10470#  10475#  10476#  10489#  10491#  10492#  10503#  10515#  10558#  10569#  10583#  10600#  10609#
                     10637#  10641#  10660#  10661#  10672#  10676#  10780#  10782#  10810#  10878#  10880#  10903#  10930#
                     10932#  11006#  11031#  11071#  11104#  11165#  11168#  11198#  11224#  11226#  11236#  11269#  11284#
                     11289#  11372#  11379#  11396#  11570#  11571#  11589#
$$LOCN= 000000       11331#  11406   11419#  11451   11464#  11495   11528#  11549   11552#  11599
$$RETU= 000000       11331#  11397#  11399#  11406   11419#  11451   11464#  11495   11528#  11549   11552#  11599
$$RTN1= 000242       11331#  11397   11406   11419#  11449   11451   11464#  11495   11528#  11549   11552#  11593   11599
$$RTN2= 000243       11331#  11399   11406   11419#  11451   11464#  11495   11528#  11549   11552#  11599
$$SRC = 000027       10691#  11442#
$$TO  = 000000        9070#  10097#  10178#  10195#  10208#  10260#  10272#  10314#  10341#  10376#  10421#  10547#  10552#
                     10598#  10777#  10809#  10870#  10874#  10927#  10978#  10988#  10993#  11049#  11057#  11061#  11108#
                     11283#  11388#  11574#
$OFILL  015317       11632#*
$1      002126        9109#   9123
$10     002446        9211    9214#
$100    005260       10182   10185#
$101    005322       10196   10199#
$102    005364       10209   10212#
$103    005424       10232   10235#
$104    005512       10261   10264#
$105    005554       10273   10276#
$106    005612       10300   10302#
$107    005660       10317   10320#
$11     002474        9221    9224#
$110    005706       10333   10336#
$111    005742       10342   10345#
$112    006034       10379   10382#
$113    006056       10391   10394#
$114    006150       10423   10426#
$115    006166       10435   10438#
$116    006224       10458   10460#
$117    006244       10469#  10475
$12     002522        9232    9235#
$120    006262       10470   10472#
$121    006264       10472   10474#
$122    006300       10475#
$123    006324       10476   10482#
$124    006324       10489#  10500
$125    006370       10489   10500#
$126    006366       10491   10499#
$127    006366       10492   10498#
$13     002552        9244    9247#
$130    006402       10503   10507#
$131    006420       10515   10518#
$132    006534       10558   10564#
$133    006562       10569   10577#
$134    006604       10583   10589#
$135    006654       10600   10607#
$136    006666       10609   10613#
$137    006726       10637   10639#
$14     002622        9279    9282#
$140    006746       10641   10643#
$141    006776       10649#
```

```
$142    006774          10649#  10690
$143    007144          10649   10690#
$144    007034          10660#  10670
$145    007070          10660   10670#
$146    007056          10661   10664#
$147    007066          10664   10668#
$15     002650           9289    9292#
$150    007102          10672   10674#
$151    007126          10674   10685#
$152    007124          10676   10678#
$153    007126          10678   10684#
$154    007416          10780   10790#
$155    007414          10782   10785#
$156    007416          10785   10789#
$157    007474          10810   10813#
$16     002676           9300    9303#
$160    007730          10878   10888#
$161    007726          10880   10884#
$162    007730          10884   10887#
$163    007772          10903   10906#
$164    010100          10930   10940#
$165    010076          10932   10936#
$166    010100          10936   10939#
$167    010230          10983#
$17     002726           9312    9315#
$170    010226          10983#  11011
$171    010354          10983   11011#
$172    010352          11006   11010#
$173    010420          11031   11034#
$174    010464          11052#
$175    010462          11052#  11077
$176    010610          11052   11077#
$177    010606          11071   11075#
$2      002152           9116    9119#
$20     002764           9340    9342#
$200    010646          11104   11106#
$201    010760          11164#  11165
$202    010774          11165#
$203    011004          11168   11171#
$204    011064          11198   11201#
$205    011144          11224   11233#
$206    011140          11226   11230#
$207    011160          11236   11240#
$21     003012           9347    9350#
$210    011224          11269   11271#
$211    011306          11284   11287#
$212    011316          11289   11292#
$213    011642          11397   11406#
$214    011644          11399   11406#
$215    011512          11370#  11390
$216    011616          11384   11390#
$217    011532          11372   11374#
$22     003040           9357    9360#
$220    011540          11374   11376#
$221    011556          11379   11382#
$222    011630          11396   11398#
```

```
$223    011744          11449   11451#
$224    011744          11451#
$225    011710          11444#
$226    011704          11444#  11447
$227    011734          11444   11447#
$23     003066           9367    9370#
$230    012024          11495#
$231    012024          11495#
$232    011772          11482#
$233    011766          11482#  11485
$234    012016          11482   11485#
$235    012004          11483#
$236    012002          11483#  11484
$237    012014          11483   11484#
$24     003124           9391    9393#
$240    012216          11549#
$241    012216          11549#
$242    012356          11593   11599#
$243    012356          11599#
$244    012220          11569#  11589
$245    012304          11570   11583#
$246    012244          11571   11573#
$247    012252          11573   11578#
$25     003144           9398    9401#
$250    012324          11583   11588#
$26     003172           9408    9411#
$27     003220           9419    9422#
$3      002264           9148    9150#
$30     003250           9431    9434#
$31     003306           9455    9458#
$32     003334           9465    9468#
$33     003362           9476    9479#
$34     003412           9488    9491#
$35     003450           9512    9515#
$36     003476           9522    9525#
$37     003524           9533    9536#
$4      002304           9154    9157#
$40     003554           9545    9548#
$40CAT= ****** U        11623   11626
$41     003612           9569    9572#
$42     003640           9579    9582#
$43     003666           9590    9593#
$44     003716           9602    9605#
$45     003756           9636    9643#
$46     004016           9668    9675#
$47     004056           9699    9708#
$5      002332           9164    9167#
$50     004114           9729    9732#
$51     004136           9738    9749#
$52     004174           9779    9784#
$53     004222           9797    9802#
$54     004250           9811    9816#
$55     004276           9825    9830#
$56     004334           9854    9857#
$57     004362           9870    9875#
$6      002360           9175    9178#
```

```
$60     004410          9884    9889#
$61     004436          9898    9903#
$62     004474          9927    9931#
$63     004522          9944    9949#
$64     004550          9958    9963#
$65     004576          9972    9977#
$66     004634         10000   10003#
$67     004662         10016   10021#
$7      002410          9187    9190#
$70     004710         10030   10035#
$71     004736         10044   10049#
$72     004774         10075   10079#
$73     005054         10101   10104#
$74     005102         10113   10116#
$75     005114         10118   10121#
$76     005142         10129   10132#
$77     005200         10155   10159#
.       = 015454        9045#   9046#   9047#   9048#   9049#   9059#   9062    9063#   9116    9123    9148    9154    9164
                        9175    9187    9211    9221    9232    9244    9279    9289    9300    9312    9340    9347    9357
                        9367    9391    9398    9408    9419    9431    9455    9465    9476    9488    9512    9522    9533
                        9545    9569    9579    9590    9602    9636    9668    9699    9729    9738    9779    9797    9811
                        9825    9854    9870    9884    9898    9927    9944    9958    9972   10000   10016   10030   10044
                       10075   10101   10113   10118   10129   10155   10182   10196   10209   10232   10261   10273   10300
                       10317   10333   10342   10379   10391   10423   10435   10458   10470   10475   10476   10489   10491
                       10492   10503   10515   10558   10569   10583   10600   10609   10637   10641   10660   10661   10672
                       10676   10723#  10780   10782   10810   10878   10880   10903   10930   10932   11006   11031   11071
                       11104   11165   11168   11198   11224   11226   11236   11269   11284   11289   11306#  11372   11379
                       11396   11534#  11539#  11541#  11570   11571   11589   11608   11611   11614   11617#  11620#  11623
                       11626   11629#  11638#
.$ASTA= ****** U       11620
.$X     = 001000        9048#
```

```
BEGIN    7390#
BGNHRD   7973#
BGNHW    7976#
BGNINI   7979#
BGNMOD   7982#  11325
BGNMSG   7985#
BGNSFT   7989#
BGNSRV   7992#  11188  11213  11505
BGNSUB   7997#   9110   9152   9161   9171   9182   9209   9218   9228   9239   9277   9286   9296   9307   9344
          9354   9364   9396   9405   9415   9426   9453   9462   9472   9482   9510   9519   9529   9540   9567
          9576   9586   9597   9635   9666   9698   9736   9792   9806   9820   9865   9879   9893   9939   9953
          9967  10011  10025  10039  10093  10109  10125  10173  10189  10203  10255  10268  10303  10323  10368
         10386  10412  10539  10568  10580  10592  10758  10793  10847  10902
BGNSW    8006#
BRESET   8009#   9088   9186   9243   9311   9430   9487   9544   9601   9642   9674   9707   9748  10389  10814
         11153  11293
CALL     7063#   9070  10097  10178  10195  10208  10260  10272  10314  10341  10376  10421  10547  10552  10598
         10777  10809  10870  10874  10927  10978  10988  10993  11049  11057  11061  11108  11283  11388  11574
CASE     7329#
CKLOOP   8021#   9606   9645   9677   9709   9751
CLRVEC   8024#   9124  10817  10946
COMMEN   1566#   8927#
DECR     6996#
DECRU    7002#
DEFAUL   7372#
DEVREG   8049#
DEVTYP   8052#
DISPAT   8055#
ELSE     6427#  10472  10664  10674  10678  10785  10884  10936  11374  11573  11583
END      7407#
ENDCLN   8058#
ENDCOM   1578#   8927#
ENDDEC   7007#
ENDDO    6498#  10500  10670
ENDHRD   8061#
ENDHW    8064#
ENDIF    6445#   9119   9150   9157   9167   9178   9190   9214   9224   9235   9247   9282   9292   9303   9315
          9342   9350   9360   9370   9393   9401   9411   9422   9434   9458   9468   9479   9491   9515   9525
          9536   9548   9572   9582   9593   9605   9643   9675   9708   9732   9749   9784   9802   9816   9830
          9857   9875   9889   9903   9931   9949   9963   9977  10003  10021  10035  10049  10079  10104  10116
         10121  10132  10159  10185  10199  10212  10235  10264  10276  10302  10320  10336  10345  10382  10394
         10426  10438  10460  10474  10482  10498  10499  10507  10518  10564  10577  10589  10607  10613  10639
         10643  10668  10684  10685  10789  10790  10813  10887  10888  10906  10939  10940  11010  11034  11075
         11106  11171  11201  11230  11233  11240  11271  11287  11292  11376  11382  11398  11578  11588
ENDINC   6983#  10690  11011  11077  11447  11484  11485
ENDINI   8067#
ENDLOO   6871#  11390
ENDMOD   8070#
ENDMSG   8073#
ENDRTN   7225#  11406  11451  11495  11549  11599
ENDSEL   7316#
ENDSFT   8077#
ENDSRV   8080#  11207  11247  11523
ENDSUB   8091#   9120   9158   9168   9179   9191   9215   9225   9236   9248   9283   9293   9304   9316   9351
          9361   9371   9402   9412   9423   9435   9459   9469   9480   9492   9516   9526   9537   9549   9573
          9583   9594   9607   9646   9678   9710   9752   9803   9817   9831   9876   9890   9904   9950   9964
```

```
          9978   10022  10036  10050  10106  10122  10133  10186  10200  10213  10265  10277  10321  10346  10384
         10395   10427  10565  10578  10590  10614  10791  10815  10893  10943
ENDSW     8094#
ENDTST    8097#   9126   9132   9249   9317   9372   9436   9493   9550   9608   9647   9679   9711   9753   9832
          9905   9979  10051  10134  10215  10278  10347  10396  10439  10523  10617  10726  10819  10947  11015
         11081  11249  11297
EQUALS    8100#
ERRDF     8104#   9118
ERRHRD    8116#   9156   9166   9177   9189   9213   9223   9234   9246   9281   9291   9302   9314   9349   9359
          9369   9400   9410   9421   9433   9457   9467   9478   9490   9514   9524   9535   9547   9571   9581
          9592   9604   9640   9672   9701   9741   9799   9813   9827   9872   9886   9900   9946   9960   9974
         10018  10032  10046  10103  10115  10120  10131  10184  10198  10211  10263  10275  10319  10335  10344
         10381  10393  10425  10437  10479  10495  10505  10517  10560  10572  10586  10602  10611  10683  10784
         10788  10812  10882  10886  10934  10938  11008  11073  11170  11286  11291
ERROR     8927#   9118   9156   9166   9177   9189   9213   9223   9234   9246   9281   9291   9302   9314   9349
          9359   9369   9400   9410   9421   9433   9457   9467   9478   9490   9514   9524   9535   9547   9571
          9581   9592   9604   9640   9672   9701   9741   9799   9813   9827   9872   9886   9900   9946   9960
          9974  10018  10032  10046  10103  10115  10120  10131  10184  10198  10211  10263  10275  10319  10335
         10344  10381  10393  10425  10437  10479  10495  10505  10517  10560  10572  10586  10602  10611  10683
         10784  10788  10812  10882  10886  10934  10938  11008  11073  11170  11286  11291
ESCAPE    1698#   8927#
EXIF      6829#
EXIFB     6851#  11384
EXIT      8128#   9149   9341   9392   9731   9783   9856   9930  10002  10078  10158  10214  10234  10301  10459
         10481  10497  10520  10563  10576  10588  10606  10615  10638  10642  10694  10905  11009  11033  11074
         11105  11176  11270  11294
GETPRI    1312#   8927#
GETSWR    1771#   8927#   9063#
GPHARD    8152#
GPRMA     8155#
GPRMD     8158#
GPRML     8161#
HEADER    8164#
IF        6332#   9116   9148   9154   9164   9175   9187   9211   9221   9232   9244   9279   9289   9300   9312
          9340   9347   9357   9367   9391   9398   9408   9419   9431   9455   9465   9476   9488   9512   9522
          9533   9545   9569   9579   9590   9602   9636   9668   9699   9729   9738   9779   9797   9811   9825
          9854   9870   9884   9898   9927   9944   9958   9972  10000  10016  10030  10044  10075  10101  10113
         10118  10129  10155  10182  10196  10209  10232  10261  10273  10300  10333  10391  10435  10458  10470
         10476  10491  10492  10503  10515  10558  10569  10583  10600  10609  10637  10641  10661  10672  10676
         10780  10782  10810  10878  10880  10903  10930  10932  11006  11031  11071  11104  11168  11198  11224
         11226  11236  11269  11372  11570  11571
IFB       6338#  11289  11379  11396
IFCOND    6358#
IF.ERR    6368#  10317  10342  10379  10423  11284
IF.NO.    6380#
INCR      6972#  10649  10983  11052  11444  11483
INCRU     6978#  11482
INLINE    7946#
LASTAD    8167#
LEAVE     7428#
LET       7922#   9068   9073   9075   9076   9078   9079   9080   9081   9082   9083   9105   9107   9108   9110
          9112   9121   9122   9124   9149   9152   9161   9162   9171   9173   9182   9184   9209   9218   9219
          9228   9230   9239   9241   9277   9286   9287   9296   9298   9307   9309   9341   9344   9345   9354
          9356   9364   9365   9392   9396   9405   9406   9415   9417   9426   9428   9453   9462   9463   9472
          9474   9482   9485   9510   9519   9520   9529   9531   9540   9542   9567   9576   9577   9586   9588
          9597   9599   9635   9666   9698   9731   9736   9783   9792   9795   9806   9809   9820   9823   9856
```

```
        9865    9868    9879    9882    9893    9896    9930    9939    9942    9953    9956    9967    9970   10002   10011
       10014   10025   10028   10039   10042   10078   10093   10095   10099   10109   10112   10125   10128   10158   10173
       10176   10180   10189   10193   10203   10206   10234   10251   10252   10255   10258   10268   10270   10301   10303
       10310   10323   10329   10366   10368   10371   10386   10411   10412   10417   10433   10459   10461   10462   10463
       10468   10471   10473   10480   10481   10497   10513   10539   10545   10550   10555   10563   10568   10576   10580
       10592   10596   10606   10638   10642   10645   10646   10647   10650   10652   10654   10656   10658   10659   10663
       10666   10667   10687   10688   10691   10692   10750   10753   10755   10757   10758   10760   10774   10793   10795
       10797   10817   10845   10847   10848   10850   10852   10866   10872   10875   10890   10892   10902   10905   10908
       10910   10922   10924   10925   10941   10942   10945   10946   10964   10980   10991   10995   11002   11003   11009
       11014   11033   11036   11050   11060   11064   11067   11068   11074   11105   11134   11136   11137   11140   11141
       11144   11146   11148   11151   11155   11158   11160   11173   11174   11192   11194   11196   11200   11217   11219
       11221   11228   11229   11232   11238   11270   11273   11276   11280   11282   11363   11364   11365   11373   11375
       11381   11389   11441   11442   11445   11446   11448   11481   11522   11572   11577   11579   11580   11581   11582
       11584   11585   11586   11587   11591   11592
LOCAL   7143#
LOOP    6691#  11370
MSG     9093#   9100    9136#   9142    9326#   9335    9381#   9386    9769#   9773    9844#   9848    9917#   9921    9991#
        9994   10063#  10069   10145#  10149   10289#  10293   10355#  10359   10447#  10451   10627#  10632   10737#  10744
       10833#  10838   10955#  10958   11093#  11096   11260#  11264
MULT    4484#   8927#
NEWTST  1626#   8927#   9100    9142    9202    9260    9335    9386    9447    9504    9561    9625    9657    9689    9722
        9773    9848    9921    9994   10069   10149   10226   10293   10359   10404   10451   10533   10632   10744   10838
       10958   11026   11096   11264   11305
NOLOCA  7183#
POINTE  8171#
POP     2153#   8927#   9124   10817   10946   11490   11611   11620   11629
PRINTB  8174#
PUSH    2145#   8927#   9124   10817   10946   11476   11611   11620   11629
REPEAT  6516#   9109   10469   11164   11569
REPORT  5463#   7960#   8927#
RETURN  7196#  11397   11399   11449   11593
ROUTIN  7111#  11331   11419   11464   11528   11552
SAVR14  7138#
SCOPE   8927#   9100    9142    9202    9260    9335    9386    9447    9504    9561    9625    9657    9689    9722    9773
        9848    9921    9994   10069   10149   10226   10293   10359   10404   10451   10533   10632   10744   10838   10958
       11026   11096   11264   11305   11608
SELECT  7250#
SETPRI  1279#   8927#   9266   10086   10166   10242   10767   10803   10859   10917   10972   11043   11128
SETTRA 11635#
SETUP   1337#   8927#   9062
SETVEC  8177#   9107   10757   10845
SKIP    1733#   8927#   9149    9341    9392    9731    9783    9856    9930   10002   10078   10158   10214   10234   10301
       10459   10481   10497   10520   10563   10576   10588   10606   10615   10638   10642   10694   10905   11009   11033
       11074   11105   11176   11270   11294
SLASH   1517#   8927#
SPACE   8927#
STARS   1485#   8927#   8944    8946    8964    8966    8984    8986    9006    9008    9029    9044    9046    9048    9049
        9100    9131    9133    9142    9199    9202    9257    9260    9325    9335    9380    9386    9444    9447    9501
        9504    9558    9561    9617    9620    9625    9657    9689    9719    9722    9761    9765    9773    9843    9848
        9916    9921    9990    9994   10062   10069   10144   10149   10225   10226   10288   10293   10354   10359   10403
       10404   10446   10451   10532   10533   10626   10632   10736   10744   10832   10838   10954   10958   11025   11026
       11092   11096   11259   11264   11305   11326   11354   11414   11435   11459   11475   11504   11515   11533   11557
       11564   11608   11611   11614   11617   11620   11623   11629   11632   11635
STRUCT  5720#   7961#   7962
SWRSU   1453#   8927#   9062#
TRMTRP 11635#
```

```
TYPBIN   2088#  8927#
TYPDEC   2058#  8927#  11311  11608
TYPNAM   1826#  8927#   9063
TYPNUM   2025#  8927#
TYPOCS   1978#  8927#
TYPOCT   1941#  8927#  11307  11309  11535  11538  11540  11542  11544  11617
TYPTXT   1894#  8927#  11306  11308  11310  11534  11536  11539  11541  11543
UNTIL    6650#  9123   10475  11165  11589
UNTILB   6671#
WAITMS   8188#  10097  10178  10195  10208  10260  10272  10547  10552  10598  10777  10809  10874  10927  11388
WHILE    6487#  10489  10660
WHILEB   6492#
$ADDON   5805#  9109   9116   9148   9154   9164   9175   9187   9211   9221   9232   9244   9279   9289   9300
         9312   9340   9347   9357   9367   9391   9398   9408   9419   9431   9455   9465   9476   9488   9512
         9522   9533   9545   9569   9579   9590   9602   9636   9668   9699   9729   9738   9779   9797   9811
         9825   9854   9870   9884   9898   9927   9944   9958   9972   10000  10016  10030  10044  10075  10101
         10113  10118  10129  10155  10182  10196  10209  10232  10261  10273  10300  10317  10333  10342  10379
         10391  10423  10435  10458  10469  10470  10472  10475  10476  10489  10491  10492  10500  10503  10515
         10558  10569  10583  10600  10609  10637  10641  10649  10660  10661  10664  10670  10672  10674  10676
         10678  10780  10782  10785  10810  10878  10880  10884  10903  10930  10932  10936  10983  11006  11031
         11052  11071  11104  11164  11165  11168  11198  11224  11226  11236  11269  11284  11289  11331  11370
         11372  11374  11379  11390  11396  11419  11444  11464  11482  11483  11528  11552  11569  11570  11571
         11573  11583
$AND     6092#  10676
$BRANC   5931#  9116   9123   9148   9154   9164   9175   9187   9211   9221   9232   9244   9279   9289   9300
         9312   9340   9347   9357   9367   9391   9398   9408   9419   9431   9455   9465   9476   9488   9512
         9522   9533   9545   9569   9579   9590   9602   9636   9668   9699   9729   9738   9779   9797   9811
         9825   9854   9870   9884   9898   9927   9944   9958   9972   10000  10016  10030  10044  10075  10101
         10113  10118  10129  10155  10182  10196  10209  10232  10261  10273  10300  10317  10333  10342  10379
         10391  10423  10435  10458  10470  10472  10475  10476  10489  10491  10492  10500  10503  10515  10558
         10569  10583  10600  10609  10637  10641  10649  10660  10661  10664  10670  10672  10674  10676  10678
         10690  10780  10782  10785  10810  10878  10880  10884  10903  10930  10932  10936  10983  11006  11011
         11031  11052  11071  11077  11104  11165  11168  11198  11224  11226  11236  11269  11284  11289  11372
         11374  11379  11384  11390  11396  11397  11399  11444  11447  11449  11482  11483  11484  11485  11570
         11571  11573  11583  11589  11593
$BRCOD   6230#  10475  10649  10983  11052  11165  11384  11444  11482  11483
$CALL    7039#  9070   10097  10178  10195  10208  10260  10272  10314  10341  10376  10421  10547  10552  10598
         10777  10809  10870  10874  10927  10978  10988  10993  11049  11057  11061  11108  11283  11388  11574
$CHECK   6288#  9116   9148   9154   9164   9175   9187   9211   9221   9232   9244   9279   9289   9300   9312
         9340   9347   9357   9367   9391   9398   9408   9419   9431   9455   9465   9476   9488   9512   9522
         9533   9545   9569   9579   9590   9602   9636   9668   9699   9729   9738   9779   9797   9811   9825
         9854   9870   9884   9898   9927   9944   9958   9972   10000  10016  10030  10044  10075  10101  10113
         10118  10129  10155  10182  10196  10209  10232  10261  10273  10300  10333  10391  10435  10458  10470
         10476  10489  10491  10492  10503  10515  10558  10569  10583  10600  10609  10637  10641  10660  10661
         10672  10676  10780  10782  10810  10878  10880  10903  10930  10932  11006  11031  11071  11104  11168
         11198  11224  11226  11236  11269  11289  11372  11379  11396  11570  11571
$CHK1    7725#  9068   9073   9075   9076   9078   9083   9105   9107   9108   9110   9112   9124   9149   9152
         9161   9171   9182   9209   9218   9228   9239   9277   9286   9296   9307   9341   9344   9354   9364
         9392   9396   9405   9415   9426   9453   9462   9472   9482   9510   9519   9529   9540   9567   9576
         9586   9597   9635   9666   9698   9731   9736   9783   9792   9806   9820   9856   9865   9879   9893
         9930   9939   9953   9967   10002  10011  10025  10039  10078  10093  10099  10109  10125  10158  10173
         10180  10189  10203  10234  10252  10255  10268  10301  10303  10310  10323  10329  10368  10371  10386
         10412  10417  10433  10459  10462  10463  10468  10471  10480  10481  10497  10513  10539  10545  10550
         10555  10563  10568  10576  10580  10592  10596  10606  10638  10642  10645  10646  10649  10650  10652
         10654  10656  10658  10659  10663  10687  10688  10692  10750  10753  10757  10758  10793  10795  10817
         10845  10847  10848  10866  10875  10902  10905  10908  10945  10946  10980  10983  10991  10995  11009
```

```
        11033   11050   11052   11060   11064   11074   11105   11134   11136   11137   11140   11141   11144   11146   11148
        11151   11196   11228   11229   11270   11276   11282   11363   11364   11365   11373   11375   11381   11441   11444
        11448   11481   11482   11483   11572   11577   11579   11580   11581   11582   11584   11591
$CKOP2   7826#   9079    9080    9081    9082    9121    9122    9124    9162    9173    9184    9219    9230    9241    9287
         929E    9309    9345    9356    9365    9406    9417    9428    9463    9474    9485    9520    9531    9542    9577
         9586    9599    9795    9809    9823    9868    9882    9896    9942    9956    9970    10014   10028   10042   10095
        10112   10128   10176   10193   10206   10251   10258   10270   10366   10411   10461   10473   10647   10666   10667
        10691   10755   10760   10774   10797   10817   10850   10852   10872   10890   10892   10910   10922   10924   10925
        10941   10942   10946   10964   11002   11003   11014   11036   11067   11068   11155   11158   11160   11173   11174
        11192   11194   11200   11217   11219   11221   11232   11238   11273   11280   11389   11442   11445   11446   11522
        11585   11586   11587   11592
$CKR6    7554#  10691   11442
$CMND    6119#   9116    9148    9154    9164    9175    9187    9211    9221    9232    9244    9279    9289    9300    9312
         9340    9347    9357    9367    9391    9398    9408    9419    9431    9455    9465    9476    9488    9512    9522
         9533    9545    9569    9579    9590    9602    9636    9668    9699    9729    9738    9779    9797    9811    9825
         9854    9870    9884    9898    9927    9944    9958    9972    10000   10016   10030   10044   10075   10101   10113
        10118   10129   10155   10182   10196   10209   10232   10261   10273   10300   10333   10391   10435   10458   10470
        10476   10489   10491   10492   10503   10515   10558   10569   10583   10600   10609   10637   10641   10660   10661
        10672   10676   10780   10782   10810   10878   10880   10903   10930   10932   11006   11031   11071   11104   11168
        11198   11224   11226   11236   11269   11289   11372   11379   11396   11570   11571
$COMPA   6280#   9116    9148    9154    9164    9175    9187    9211    9221    9232    9244    9279    9289    9300    9312
         9340    9347    9357    9367    9391    9398    9408    9419    9431    9455    9465    9476    9488    9512    9522
         9533    9545    9569    9579    9590    9602    9636    9668    9699    9729    9738    9779    9797    9811    9825
         9854    9870    9884    9898    9927    9944    9958    9972    10000   10016   10030   10044   10075   10101   10113
        10118   10129   10155   10182   10196   10209   10232   10261   10273   10300   10317   10333   10342   10379   10391
        10423   10435   10458   10470   10476   10489   10491   10492   10503   10515   10558   10569   10583   10600   10609
        10637   10641   10649   10660   10661   10672   10676   10780   10782   10810   10878   10880   10903   10930   10932
        10983   11006   11031   11052   11071   11104   11168   11198   11224   11226   11236   11269   11284   11289   11372
        11379   11396   11444   11482   11483   11570   11571
$COUNT   7032#   9070   10097   10178   10195   10208   10260   10272   10314   10341   10376   10421   10547   10552   10598
        10777   10809   10870   10874   10927   10978   10988   10993   11049   11057   11061   11108   11283   11388   11574
$DO      6046#  10489   10660
$ELSE    6387#
$ERRMS   6002#
$EXIFA   6705#
$EXIFO   6742#
$EXIF2   6800#  11384
$EXIF3   6779#
$GENBR   5924#   9116    9123    9148    9154    9164    9175    9187    9211    9221    9232    9244    9279    9289    9300
         9312    9340    9347    9357    9367    9391    9398    9408    9419    9431    9455    9465    9476    9488    9512
         9522    9533    9545    9569    9579    9590    9602    9636    9668    9699    9729    9738    9779    9797    9811
         9825    9854    9870    9884    9898    9927    9944    9958    9972    10000   10016   10030   10044   10075   10101
        10113   10118   10129   10155   10182   10196   10209   10232   10261   10273   10300   10317   10333   10342   10379
        10391   10423   10435   10458   10470   10472   10475   10476   10489   10491   10492   10500   10503   10515   10558
        10569   10583   10600   10609   10637   10641   10649   10660   10661   10664   10670   10672   10674   10676   10678
        10690   10780   10782   10785   10810   10878   10880   10884   10903   10930   10932   10936   10983   11006   11011
        11031   11052   11071   11077   11104   11165   11168   11198   11224   11226   11236   11269   11284   11289   11372
        11374   11379   11384   11390   11396   11397   11399   11444   11447   11449   11482   11483   11484   11485   11570
        11571   11573   11583   11589   11593
$GENTA   5900#   9109    9119    9150    9157    9167    9178    9190    9214    9224    9235    9247    9282    9292    9303
         9315    9342    9350    9360    9370    9393    9401    9411    9422    9434    9458    9468    9479    9491    9515
         9525    9536    9548    9572    9582    9593    9605    9643    9675    9708    9732    9749    9784    9802    9816
         9830    9857    9875    9889    9903    9931    9949    9963    9977    10003   10021   10035   10049   10079   10104
        10116   10121   10132   10159   10185   10199   10212   10235   10264   10276   10302   10320   10336   10345   10382
        10394   10426   10438   10460   10469   10472   10474   10475   10482   10489   10498   10499   10500   10507   10518
        10564   10577   10589   10607   10613   10639   10643   10649   10660   10664   10668   10670   10674   10678   10684
```

```
        10685   10690   10785   10789   10790   10813   10884   10887   10888   10906   10936   10939   10940   10983   11010
        11011   11034   11052   11075   11077   11106   11164   11165   11171   11201   11230   11233   11240   11271   11287
        11292   11370   11374   11376   11382   11390   11398   11406   11444   11447   11451   11482   11483   11484   11485
        11495   11549   11569   11573   11578   11583   11588   11599
$IF      6306#   9116    9148    9154    9164    9175    9187    9211    9221    9232    9244    9279    9289    9300    9312
         9340    9347    9357    9367    9391    9398    9408    9419    9431    9455    9465    9476    9488    9512    9522
         9533    9545    9569    9579    9590    9602    9636    9668    9699    9729    9738    9779    9797    9811    9825
         9854    9870    9884    9898    9927    9944    9958    9972   10000   10016   10030   10044   10075   10101   10113
        10118   10129   10155   10182   10196   10209   10232   10261   10273   10300   10333   10391   10435   10458   10470
        10476   10491   10492   10503   10515   10558   10569   10583   10600   10609   10637   10641   10661   10672   10676
        10780   10782   10810   10878   10880   10903   10930   10932   11006   11031   11071   11104   11168   11198   11224
        11226   11236   11269   11289   11372   11379   11396   11570   11571
$IFCOD   6178#   9116    9123    9148    9154    9164    9175    9187    9211    9221    9232    9244    9279    9289    9300
         9312    9340    9347    9357    9367    9391    9398    9408    9419    9431    9455    9465    9476    9488    9512
         9522    9533    9545    9569    9579    9590    9602    9636    9668    9699    9729    9738    9779    9797    9811
         9825    9854    9870    9884    9898    9927    9944    9958    9972   10000   10016   10030   10044   10075   10101
        10113   10118   10129   10155   10182   10196   10209   10232   10261   10273   10300   10333   10391   10435   10458
        10470   10475   10476   10489   10491   10492   10503   10515   10558   10569   10583   10600   10609   10637   10641
        10660   10661   10672   10676   10780   10782   10810   10878   10880   10903   10930   10932   11006   11031   11071
        11104   11165   11168   11198   11224   11226   11236   11269   11289   11372   11379   11396   11570   11571   11589
$IFCON   6345#  10317   10342   10379   10423   11284
$IFOPR   5944#   9116    9123    9148    9154    9164    9175    9187    9211    9221    9232    9244    9279    9289    9300
         9312    9340    9347    9357    9367    9391    9398    9408    9419    9431    9455    9465    9476    9488    9512
         9522    9533    9545    9569    9579    9590    9602    9636    9668    9699    9729    9738    9779    9797    9811
         9825    9854    9870    9884    9898    9927    9944    9958    9972   10000   10016   10030   10044   10075   10101
        10113   10118   10129   10155   10182   10196   10209   10232   10261   10273   10300   10317   10333   10342   10379
        10391   10423   10435   10458   10470   10475   10476   10489   10491   10492   10503   10515   10558   10569   10583
        10600   10609   10637   10641   10660   10661   10672   10676   10780   10782   10810   10878   10880   10903   10930
        10932   11006   11031   11071   11104   11165   11168   11198   11224   11226   11236   11269   11284   11289   11372
        11379   11396   11570   11571   11589
$LET     7909#   9068    9073    9075    9076    9078    9079    9080    9081    9082    9083    9105    9107    9108    9110
         9112    9121    9122    9124    9149    9152    9161    9162    9171    9173    9182    9184    9209    9218    9219
         9228    9230    9239    9241    9277    9286    9287    9296    9298    9307    9309    9341    9344    9345    9354
         9356    9364    9365    9392    9396    9405    9406    9415    9417    9426    9428    9453    9462    9463    9472
         9474    9482    9485    9510    9519    9520    9529    9531    9540    9542    9567    9576    9577    9586    9588
         9597    9599    9635    9666    9698    9731    9736    9783    9792    9795    9806    9809    9820    9823    9856
         9865    9868    9879    9882    9893    9896    9930    9939    9942    9953    9956    9967    9970   10002   10011
        10014   10025   10028   10039   10042   10078   10093   10095   10099   10109   10112   10125   10128   10158   10173
        10176   10180   10189   10193   10203   10206   10234   10251   10252   10255   10258   10268   10270   10301   10303
        10310   10323   10329   10366   10368   10371   10386   10411   10412   10417   10433   10459   10461   10462   10463
        10468   10471   10473   10480   10481   10497   10513   10539   10545   10550   10555   10563   10568   10576   10580
        10592   10596   10606   10638   10642   10645   10646   10647   10650   10652   10654   10656   10658   10659   10663
        10666   10667   10687   10688   10691   10692   10750   10753   10755   10757   10758   10760   10774   10793   10795
        10797   10817   10845   10847   10848   10850   10852   10866   10872   10875   10890   10892   10902   10905   10908
        10910   10922   10924   10925   10941   10942   10945   10946   10964   10980   10991   10995   11002   11003   11009
        11014   11033   11036   11050   11060   11064   11067   11068   11074   11105   11134   11136   11137   11140   11141
        11144   11146   11148   11151   11155   11158   11160   11173   11174   11192   11194   11196   11200   11217   11219
        11221   11228   11229   11232   11238   11270   11273   11276   11280   11282   11363   11364   11365   11373   11375
        11381   11389   11441   11442   11445   11446   11448   11481   11522   11572   11577   11579   11580   11581   11582
        11584   11585   11586   11587   11591   11592
$LPCNT   6927#  10649   10983   11052   11444   11482   11483
$OPADD   7566#   9079    9080    9081    9082    9121    9122    9124   10473   10649   10666   10667   10755   10817   10946
        10983   11052   11192   11217   11232   11444   11482   11483   11522   11586   11587   11592
$OPAND   7608#  10691   11442
$OPCD1   7649#   9079    9080    9081    9082    9121    9122    9124    9162    9173    9184    9219    9230    9241    9287
         9298    9309    9345    9356    9365    9406    9417    9428    9463    9474    9485    9520    9531    9542    9577
```

```
                 9588    9599    9795    9809    9823    9868    9882    9896    9942    9956    9970   10014   10028   10042   10095
                10112   10128   10176   10193   10206   10251   10258   10270   10366   10411   10461   10473   10647   10649   10666
                10667   10691   10755   10760   10774   10797   10817   10850   10852   10872   10890   10892   10910   10922   10924
                10925   10941   10942   10946   10964   10983   11002   11003   11014   11036   11052   11067   11068   11155   11158
                11160   11173   11174   11192   11194   11200   11217   11219   11221   11232   11238   11273   11280   11389   11442
                11444   11445   11446   11448   11482   11483   11522   11585   11586   11587   11592
$OPCD2           7524#  11445   11448   11585
$OPCOD           7710#   9079    9080    9081    9082    9121    9122    9124    9162    9173    9184    9219    9230    9241    9287
                 9298    9309    9345    9356    9365    9406    9417    9428    9463    9474    9485    9520    9531    9542    9577
                 9588    9599    9795    9809    9823    9868    9882    9896    9942    9956    9970   10014   10028   10042   10095
                10112   10128   10176   10193   10206   10251   10258   10270   10366   10411   10461   10473   10647   10649   10666
                10667   10691   10755   10760   10774   10797   10817   10850   10852   10872   10890   10892   10910   10922   10924
                10925   10941   10942   10946   10964   10983   11002   11003   11014   11036   11052   11067   11068   11155   11158
                11160   11173   11174   11192   11194   11200   11217   11219   11221   11232   11238   11273   11280   11389   11442
                11444   11445   11446   11448   11482   11483   11522   11585   11586   11587   11592
$OPCOM           7461#  11448
$OPDEF           5909#   9068    9070    9073    9075    9076    9078    9079    9080    9081    9082    9083    9105    9107    9108
                 9110    9112    9116    9121    9122    9123    9124    9148    9149    9152    9154    9161    9162    9164    9171
                 9173    9175    9182    9184    9187    9209    9211    9218    9219    9221    9228    9230    9232    9239    9241
                 9244    9277    9279    9286    9287    9289    9296    9298    9300    9307    9309    9312    9340    9341    9344
                 9345    9347    9354    9356    9357    9364    9365    9367    9391    9392    9396    9398    9405    9406    9408
                 9415    9417    9419    9426    9428    9431    9453    9455    9462    9463    9465    9472    9474    9476    9482
                 9485    9488    9510    9512    9519    9520    9522    9529    9531    9533    9540    9542    9545    9567    9569
                 9576    9577    9579    9586    9588    9590    9597    9599    9602    9635    9636    9666    9668    9698    9699
                 9729    9731    9736    9738    9779    9783    9792    9795    9797    9806    9809    9811    9820    9823    9825
                 9854    9856    9865    9868    9870    9879    9882    9884    9893    9896    9898    9927    9930    9939    9942
                 9944    9953    9956    9958    9967    9970    9972   10000   10002   10011   10014   10016   10025   10028   10030
                10039   10042   10044   10075   10078   10093   10095   10097   10099   10101   10109   10112   10113   10118   10125
                10128   10129   10155   10158   10173   10176   10178   10180   10182   10189   10193   10195   10196   10203   10206
                10208   10209   10232   10234   10251   10252   10255   10258   10260   10261   10268   10270   10272   10273   10300
                10301   10303   10310   10314   10317   10323   10329   10333   10341   10342   10366   10368   10371   10376   10379
                10386   10391   10411   10412   10417   10421   10423   10433   10435   10458   10459   10461   10462   10463   10468
                10470   10471   10472   10473   10475   10476   10480   10481   10489   10491   10492   10497   10500   10503   10513
                10515   10539   10545   10547   10550   10552   10555   10558   10563   10568   10569   10576   10580   10583   10592
                10596   10598   10600   10606   10609   10637   10638   10641   10642   10645   10646   10647   10649   10650   10652
                10654   10656   10658   10659   10660   10661   10663   10664   10666   10667   10670   10672   10674   10676   10678
                10687   10688   10690   10691   10692   10750   10753   10755   10757   10758   10760   10774   10777   10780   10782
                10785   10793   10795   10797   10809   10810   10817   10845   10847   10848   10850   10852   10866   10870   10872
                10874   10875   10878   10880   10884   10890   10892   10902   10903   10905   10908   10910   10922   10924   10925
                10927   10930   10932   10936   10941   10942   10945   10946   10964   10978   10980   10983   10988   10991   10993
                10995   11002   11003   11006   11009   11011   11014   11031   11033   11036   11049   11050   11052   11057   11060
                11061   11064   11067   11068   11071   11074   11077   11104   11105   11108   11134   11136   11137   11140   11141
                11144   11146   11148   11151   11155   11158   11160   11165   11168   11173   11174   11192   11194   11196   11198
                11200   11217   11219   11221   11224   11226   11228   11229   11232   11236   11238   11269   11270   11273   11276
                11280   11282   11283   11284   11289   11363   11364   11365   11372   11373   11374   11375   11379   11381   11384
                11388   11389   11390   11396   11397   11399   11406   11441   11442   11444   11445   11446   11447   11448   11449
                11451   11481   11482   11483   11484   11485   11495   11522   11549   11570   11571   11572   11573   11574   11577
                11579   11580   11581   11582   11583   11584   11585   11586   11587   11589   11591   11592   11593   11599
$OPEQU           7642#
$OPNAN           7627#
$OPNEG           7473#
$OPNOR           7621#
$OPNOT           7616#   9173    9230    9298    9345    9365    9417    9474    9531    9588    9795    9823    9868    9896    9942
                 9970   10014   10042   10095   10128   10176   10206   10251   10270   10760   10797   10852   10890   10892   10910
                10922   10941   10942   11002   11003   11014   11036   11067   11068   11173   11174   11194   11200   11219   11221
                11238
```

```
$OPOR     7603#  9162   9184   9219   9241   9287   9309   9356   9406   9428   9463   9485   9520   9542   9577
          9599   9809   9882   9956   10028  10112  10193  10258  10366  10411  10461  10647  10774  10850  10872
          10924  10925  10964  11155  11158  11160  11273  11280  11446
$OPROT    7493#  11585
$OPRO     7752#  9068   9073   9075   9076   9078   9083   9105   9107   9108   9110   9112   9124   9149   9152
          9161   9171   9182   9209   9218   9228   9239   9277   9286   9296   9307   9341   9344   9354   9364
          9392   9396   9405   9415   9426   9453   9462   9472   9482   9510   9519   9529   9540   9567   9576
          9586   9597   9635   9666   9698   9731   9736   9783   9792   9806   9820   9856   9865   9879   9893
          9930   9939   9953   9967   10002  10011  10025  10039  10078  10093  10099  10109  10125  10158  10173
          10180  10189  10203  10234  10252  10255  10268  10301  10303  10310  10323  10329  10368  10371  10386
          10412  10417  10433  10459  10462  10463  10468  10471  10480  10481  10497  10513  10539  10545  10550
          10555  10563  10568  10576  10580  10592  10596  10606  10638  10642  10645  10646  10649  10650  10652
          10654  10656  10658  10659  10663  10687  10688  10692  10750  10753  10757  10758  10793  10795  10817
          10845  10847  10848  10866  10875  10902  10905  10908  10945  10946  10980  10983  10991  10995  11009
          11033  11050  11052  11060  11064  11074  11105  11134  11136  11137  11140  11141  11144  11146  11148
          11151  11196  11228  11229  11270  11276  11282  11363  11364  11365  11373  11375  11381  11441  11444
          11481  11482  11483  11572  11577  11579  11580  11581  11582  11584  11591
$OPR1     7799#  10649  10983  11052  11444  11448  11482  11483
$OPR2     7868#  9079   9080   9081   9082   9121   9122   9124   9162   9173   9184   9219   9230   9241   9287
          9298   9309   9345   9356   9365   9406   9417   9428   9463   9474   9485   9520   9531   9542   9577
          9588   9599   9795   9809   9823   9868   9882   9896   9942   9956   9970   10014  10028  10042  10095
          10112  10128  10176  10193  10206  10251  10258  10270  10366  10411  10461  10473  10647  10666  10667
          10691  10755  10760  10774  10797  10817  10850  10852  10872  10890  10892  10910  10922  10924  10925
          10941  10942  10946  10964  11002  11003  11014  11036  11067  11068  11155  11158  11160  11173  11174
          11192  11194  11200  11217  11219  11221  11232  11238  11273  11280  11389  11442  11445  11446  11522
          11585  11586  11587  11592
$OPSHF    7510#  11445
$OPSUB    7586#  11389
$OPSWB    7485#
$OPXOR    7633#
$OR       6065#
$PUT      7022#  10097  10178  10195  10208  10260  10272  10314  10341  10376  10421  10547  10552  10598  10777
          10809  10870  10874  10927  10988  10993  11057  11061  11283  11388
$STRUC    5765#
$SUBON    5811#  9119   9123   9150   9157   9167   9178   9190   9214   9224   9235   9247   9282   9292   9303
          9315   9342   9350   9360   9370   9393   9401   9411   9422   9434   9458   9468   9479   9491   9515
          9525   9536   9548   9572   9582   9593   9605   9643   9675   9708   9732   9749   9784   9802   9816
          9830   9857   9875   9889   9903   9931   9949   9963   9977   10003  10021  10035  10049  10079  10104
          10116  10121  10132  10159  10185  10199  10212  10235  10264  10276  10302  10320  10336  10345  10382
          10394  10426  10438  10460  10472  10474  10475  10482  10498  10499  10500  10507  10518  10564  10577
          10589  10607  10613  10639  10643  10649  10664  10668  10670  10674  10678  10684  10685  10690  10785
          10789  10790  10813  10884  10887  10888  10906  10936  10939  10940  10983  11010  11011  11034  11052
          11075  11077  11106  11165  11171  11201  11230  11233  11240  11271  11287  11292  11374  11376  11382
          11390  11398  11406  11444  11447  11451  11482  11483  11484  11485  11495  11549  11573  11578  11583
          11588  11589  11599
$THEN     6027#  9116   9148   9154   9164   9175   9187   9211   9221   9232   9244   9279   9289   9300   9312
          9340   9347   9357   9367   9391   9398   9408   9419   9431   9455   9465   9476   9488   9512   9522
          9533   9545   9569   9579   9590   9602   9636   9668   9699   9729   9738   9779   9797   9811   9825
          9854   9870   9884   9898   9927   9944   9958   9972   10000  10016  10030  10044  10075  10101  10113
          10118  10129  10155  10182  10196  10209  10232  10261  10273  10300  10333  10391  10435  10458  10470
          10476  10491  10492  10503  10515  10558  10569  10583  10600  10609  10637  10641  10661  10672  10676
          10780  10782  10810  10878  10880  10903  10930  10932  11006  11031  11071  11104  11168  11198  11224
          11226  11236  11269  11289  11372  11379  11396  11570  11571
$TILA     6526#
$TILO     6563#
$UNTL2    6621#  10475  11165
```

```
$UNTL3   6600#
$WHILE   6460#  10489  10660
$$CMRE   9049#
$$CMTM   9049#
$$DEFA   7354#
$$ENDS   7305#
$$ERRO   6012#
$$ESCA   1711#  8927#
$$GEN    5888#  9109   9119   9150   9157   9167   9178   9190   9214   9224   9235   9247   9282   9292   9303
          9315   9342   9350   9360   9370   9393   9401   9411   9422   9434   9458   9468   9479   9491   9515
          9525   9536   9548   9572   9582   9593   9605   9643   9675   9708   9732   9749   9784   9802   9816
          9830   9857   9875   9889   9903   9931   9949   9963   9977  10003  10021  10035  10049  10079  10104
         10116  10121  10132  10159  10185  10199  10212  10235  10264  10276  10302  10320  10336  10345  10382
         10394  10426  10438  10460  10469  10472  10474  10475  10482  10489  10498  10499  10500  10507  10518
         10564  10577  10589  10607  10613  10639  10643  10649  10660  10664  10668  10670  10674  10678  10684
         10685  10690  10785  10789  10790  10813  10884  10887  10888  10906  10936  10939  10940  10983  11010
         11011  11034  11052  11075  11077  11106  11164  11165  11171  11201  11230  11233  11240  11271  11287
         11292  11331  11370  11374  11376  11382  11390  11398  11406  11419  11444  11447  11451  11464  11482
         11483  11484  11485  11495  11528  11549  11552  11569  11573  11578  11583  11588  11599
$$GETS   5851#  9119   9123   9150   9157   9167   9178   9190   9214   9224   9235   9247   9282   9292   9303
          9315   9342   9350   9360   9370   9393   9401   9411   9422   9434   9458   9468   9479   9491   9515
          9525   9536   9548   9572   9582   9593   9605   9643   9675   9708   9732   9749   9784   9802   9816
          9830   9857   9875   9889   9903   9931   9949   9963   9977  10003  10021  10035  10049  10079  10104
         10116  10121  10132  10159  10185  10199  10212  10235  10264  10276  10302  10320  10336  10345  10382
         10394  10426  10438  10460  10472  10474  10475  10482  10498  10499  10500  10507  10518  10564  10577
         10589  10607  10613  10639  10643  10649  10664  10668  10670  10674  10678  10684  10685  10690  10785
         10789  10790  10813  10884  10887  10888  10906  10936  10939  10940  10983  11010  11011  11034  11052
         11075  11077  11106  11165  11171  11201  11230  11233  11240  11271  11287  11292  11374  11376  11382
         11384  11390  11398  11406  11444  11447  11451  11482  11483  11484  11485  11495  11549  11573  11578
         11583  11588  11589  11599
$$GETT   5861#  10472  10664  10674  10678  10785  10884  10936  11374  11384  11573  11583
$$LPCN   6887#  10649  10983  11052  11444  11482  11483
$$NEWT   1662#  8927#  9100   9142   9202   9260   9335   9386   9447   9504   9561   9625   9657   9689   9722
          9773   9848   9921   9994  10069  10149  10226  10293  10359  10404  10451  10533  10632  10744  10838
         10958  11026  11096  11264  11305
$$POP    5875#  9119   9123   9150   9157   9167   9178   9190   9214   9224   9235   9247   9282   9292   9303
          9315   9342   9350   9360   9370   9393   9401   9411   9422   9434   9458   9468   9479   9491   9515
          9525   9536   9548   9572   9582   9593   9605   9643   9675   9708   9732   9749   9784   9802   9816
          9830   9857   9875   9889   9903   9931   9949   9963   9977  10003  10021  10035  10049  10079  10104
         10116  10121  10132  10159  10185  10199  10212  10235  10264  10276  10302  10320  10336  10345  10382
         10394  10426  10438  10460  10472  10474  10475  10482  10498  10499  10500  10507  10518  10564  10577
         10589  10607  10613  10639  10643  10649  10664  10668  10670  10674  10678  10684  10685  10690  10785
         10789  10790  10813  10884  10887  10888  10906  10936  10939  10940  10983  11010  11011  11034  11052
         11075  11077  11106  11165  11171  11201  11230  11233  11240  11271  11287  11292  11374  11376  11382
         11390  11398  11406  11444  11447  11451  11482  11483  11484  11485  11495  11549  11573  11578  11583
         11588  11589  11599
$$PUSH   5839#  9109   9116   9148   9154   9164   9175   9187   9211   9221   9232   9244   9279   9289   9300
          9312   9340   9347   9357   9367   9391   9398   9408   9419   9431   9455   9465   9476   9488   9512
          9522   9533   9545   9569   9579   9590   9602   9636   9668   9699   9729   9738   9779   9797   9811
          9825   9854   9870   9884   9898   9927   9944   9958   9972  10000  10016  10030  10044  10075  10101
         10113  10118  10129  10155  10182  10196  10209  10232  10261  10273  10300  10317  10333  10342  10379
         10391  10423  10435  10458  10469  10470  10472  10476  10489  10491  10492  10500  10503  10515  10558
         10569  10583  10600  10609  10637  10641  10649  10660  10661  10664  10670  10672  10674  10676  10678
         10780  10782  10785  10810  10878  10880  10884  10903  10930  10932  10936  10983  11006  11031  11052
         11071  11104  11164  11168  11198  11224  11226  11236  11269  11284  11289  11331  11370  11372  11374
         11379  11390  11396  11419  11444  11464  11482  11483  11528  11552  11569  11570  11571  11573  11583
```

```
$$SELE   7244#
$$SET   11635#
$$SETM   9062#
$$SETS   5820#   9109   9116   9148   9154   9164   9175   9187   9211   9221   9232   9244   9279   9289   9300
         9312    9340   9347   9357   9367   9391   9398   9408   9419   9431   9455   9465   9476   9488   9512
         9522    9533   9545   9569   9579   9590   9602   9636   9668   9699   9729   9738   9779   9797   9811
         9825    9854   9870   9884   9898   9927   9944   9958   9972  10000  10016  10030  10044  10075  10101
        10113   10118  10129  10155  10182  10196  10209  10232  10261  10273  10300  10317  10333  10342  10379
        10391   10423  10435  10458  10469  10470  10472  10476  10489  10491  10492  10500  10503  10515  10558
        10569   10583  10600  10609  10637  10641  10649  10660  10661  10664  10670  10672  10674  10676  10678
        10780   10782  10785  10810  10878  10880  10884  10903  10930  10932  10936  10983  11006  11031  11052
        11071   11104  11164  11168  11198  11224  11226  11236  11269  11284  11289  11331  11370  11372  11374
        11379   11390  11396  11419  11444  11464  11482  11483  11528  11552  11569  11570  11571  11573  11583
$$SETT   5830#
$$SKIP   1746#   8927#   9149   9341   9392   9731   9783   9856   9930  10002  10078  10158  10214  10234  10301
        10459   10481  10497  10520  10563  10576  10588  10606  10615  10638  10642  10694  10905  11009  11033
        11074   11105  11176  11270  11294
.EQUAT    189#   7959#   8927
.HEADE     61#   7958#   8924
.INIT    5658#   8927#  11638
.SETUP   1213#   7958#   9061
.SWRHI    104#   7959#   8925
.SWRLO   8925#
.$ACT1   5064#   7960#   9046
.$APTB   5109#   7960#   9049#
.$APTH   5370#   7960#   9048
.$APTY   5547#   7960#  11620
.$ASTA   5417#
.$CATC    932#   7958#   9045
.$CMTA   1047#   7958#   9049
.$DB2D   4686#
.$DB2O   4812#
.$DIV    4587#
.$EOP    2214#   7958#  11608
.$ERRO   2700#   7959#  11623
.$ERRT   2896#
.$MULT   4523#
.$POWE   4229#   7959#  11611
.$RAND   4307#
.$RDDE   3891#
.$RDOC   3797#
.$READ   3395#   7958#  11617
.$R2AZ   4958#
.$SAVE   3969#
.$SB2D   4771#
.$SB2O   4874#
.$SCOP   2454#   7959#  11626
.$SIZE   4361#
.$SUPR   4913#
.$TRAP   4073#   7959#  11635
.$TYPB   3287#
.$TYPD   3209#   7959#  11629
.$TYPE   2985#   7958#  11614
.$TYPO   3112#   7960#  11632
.$40CA    972#
```

. ABS. 015454 000

ERRORS DETECTED: 0

CNDVAA,CNDVAA/CRF/NL:TOC=CNMAC2.SML,CNMAC.MAC,CNDVAA.P11
RUN-TIME: 85 90 5 SECONDS
RUN-TIME RATIO: 341/182=1.8
CORE USED: 43K (86 PAGES)