DLV11-F

DLV11-F OFFLINE TEST
CVDVCB0

AH-E007B-MC

FICHE 1 OF 1

JUN 1978
digital
MADE IN USA

```
                        REM     ∂



                                        IDENTIFICATION
                                        --------------

                            PRODUCT CODE        AC-E006B-MC

                            PRODUCT NAME        CVDVCB0 DLV11-F OFFLINE TEST

                            PRODUCT DATE        MARCH, 1978

                            AUTHOR               ODES CHOATE

                            MAINTAINER          DIAGNOSTIC ENGINEERING GROUP

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

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

                           COPYRIGHT (C) 1977, 1978 BY DIGITAL EQUIPMENT CORPORATION

                       THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION

                           DIGITAL        PDP            UNIBUS         MASSBUS
                           DEC            DECUS          DECTAPE
```

TABLE OF CONTENTS

```
                                1 0     GENERAL PROGRAM INFORMATION
                                        ------- ---- -- ---------- -

                                1 1     PROGRAM PURPOSE (ABSTRACT)

                                        THIS DIAGNOSTIC IS A LOGIC TEST TO VERIFY THE OPERATION OF THE
                                        DLV11-F  SERIAL  LINE INTERFACE   THE USER CAN SELECTIVELY
                                        ENABLE AND DISABLE TESTING OF  THE  OPTIONS BY  ALTERING  THE
                                        CONTENTS  OF  '$USER'   THE DIAGNOSTIC IS DESIGNED TO TEST AND
                                        DETECT FAULTS TO THE LOGIC LEVEL  (NOT  TO  THE  CHIP  LEVEL)
                                        THIS TEST OPERATES ON UP TO SIXTEEN(16) IDENTICALLY CONFIGURED
                                        DLV11-F SERIAL LINE INTERFACES   THE DEFAULT ADDRESSES ARE

                                                177560 -CONSOLE INTERFACE DEVICE ADDRESS
                                                175610 -FIRST SERIAL LINE ADDRESS  OF  15  CONSECUTIVE
                                                        SERIAL LINE DEVICES


                                                   60 - VECTOR FOR CONSOLE DEVICE INTERFACE
                                                  300 - VECTOR FOR FIRST OF 15 DEVICES

                                        THIS PROGRAM IS DESIGNED TO RUN  ON  ANY  PDP-11  WITH  4K  OF
                                        MEMORY AND A DLV11-F (LSI-BUS) MODULE   IT CAN RUN UNDER XXDP,
                                        APT, AND ACT MONITORS, AND  ON  PROCESSORS  WITH  NO  HARDWARE
                                        SWITCH REGISTER   A POWER FAILURE WILL CAUSE THE DIAG-
                                        NOSTIC TO RESTART

                                1 2     SYSTEM REQUIREMENTS

                                                HARDWARE REQUIREMENTS

                                                ANY PDP-11 F MILY PROCESSOR
                                                4K MEMORY - MINIMUM
                                               A SPECIAL WRAP CONNECTOR OR EQUIVALENT (OPTIONAL)


                                                SOFTWARE REQU REMENTS


                                                THIS DIAGNOSTIC  S DES GNED TO  RUN  IN  ANY  OF  THE
                                         FOLLOWING WAYS
                                                STAND ALONE
                                                WITH APT MON TOR
                                                WITH ACT MON TOR
                                                WITH XXDP MONITOR (CHAINABLE)

                                1 3     RELATED DOCUMENTS AND STANDARDS

                                        DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS        175-003-009-02
                                        APT                                                     MD-11-DZZMA
                                        ACT                                                     AUTOCAT-11-QZAUB
                                        SYSMAC                                                  MD-11-DZQAC


                                1 4     DIAGNOSTIC HIERARCHY PREREQUIS TES
```

```
                                NO SPECIAL DIAGNOSTICS ARE REQUIRED TO RUN  BEFORE  THIS,  BUT
                                THE  PROCESSOR,  MEMORY,  AND  BUS  ARE  ASSUMED  TO BE FULLY
                                OPERATIONAL


                        1 5     ASSUMPTIONS


                                THIS DIAGNOSTIC ASSUMES  THAT  THE  OPERATOR  HAS  IN TIALIZED
                                LOCATION '$USWR' AND '$DEVM' TO THE PROPER VALUES


                        2 0     OPERATING INSTRUCTIONS
                                --------- -------------


                        2 1     LOADING AND STARTING PROCEDURES

                                USE STANDARD PROCEDURE FOR PDP-11  ABSOLUTE  BINARY  FORMATTED
                                MEDIA

                                THIS DIAGNOSTIC HAS ONLY ONE (1) STARTING  ADDRESS    200  FOP
                                START AND RESTART

                                THE USER CAN SELECT A SPEC FIC TEST TO BE EXECUTED BY  SETTING
                                SWITCH 8 IN THE SWITCH REGISTER AND THE TEST NUMBER (IN OCTAL)
                                IN THE LOWER BYTE   (NOTE   ALL TESTS PREVIOUS TO THE SELECTED
                                ONE ARE EXECUTED W.THOUT  TERATIONS )


                        2 2     SPECIAL ENVIRONMENTS

                                TH S DIAGNOSTIC FOLLOWS THE  STANDARD  PROCEDURE  FOR  RUNN NG
                                UDER  APT,ACT,XXDP  MONITORS,  AS DESCRIBED IN THEIR RESPECTIVE
                                PROCEDURES MANUAL AND SYSMAC PACKAGE

                        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 SW TCH REGISTER
                                        (LOC   176) IS USED


                                CONTROL

                                THIS PROGRAM ALSO SUPPORTS THE DYNAM C 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 P 'INTS W THIN 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 SEQUENCE
                    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 B T 8)


        2 4     PROGRAM OPTIONS

                THIS PROGRAM WILL SUPPORT TESTING OF MULTIPLE  DLV11-F'S    IT
                REQUIRES THE ADDRESS OF THE FIRST RCSR (STORED AT '$BASE') AND
                ITS INTERRUPT VECTOR (STORED AT '$VECT1'),  AND WILL  BE  ABLE
                TO  ADDRESS ANY DLV11-F STARTING AT THE SPECIFIED BASE ADDRESS
                UP TO 16 CONSECUTIVE DEVICES
                EXAMPLES        $BASE   175610
                                $VECT1     300

                THE PROGRAM WILL BE  ABLE  TO  TEST  ANY  DLV11-F  WITHIN  THE
                ADDRESS RANGE 175610 --> 176000

                $BASE AND $VECT1 DEFAULT TO 175610 AND 300 RESPECT VELY
```

```
                        THE PROGRAM ASSOCIATES UNIT NUMBERS AS FOLLOWS    (NUMBERS  IN
                        PARENTHESIS ARE OCTAL)
                        UNIT#0 -- BASE ADDRESS STORED AT '$BASE'
                                  ASSOCIATED BASE VECTOR STORED AT '$VECT1'
                        UNIT#1 -- BASE ADDRESS + (10)
                                  BASE VECTOR  + (10)
                                  !
                                  !    UP TO
                                  !
                        UNIT#14 -- BASE ADDRESS + (160)
                                   BASE VECTOR  + (160)

                        LOCATION '$DEVM' IS USED AS A BIT MAP TO INDICATE  WHICH  UNIT
                        NUMBERS ARE PRESENT AND WILL BE TESTED

                         BIT 15 BIT 14  -    -    -       BIT 1  BIT 0
                        ---------------------------------------------
                        ' CON- ! UNIT !                  ! UNIT ! UNIT !
                        ' SOLE !  14  !                  ! #1   !  #0  !
                        ---------------------------------------------

                        A B T MAP CAN BE ENTERED AT  '$DEVM'  PRIOR  TO  STARTING  THE
                        PROGRAM

                        EXAMPLE
                                $BASE   175610
                                $VECTOR    300
                                $DEVM   100013

                        THE PROGRAM WILL TEST-

                                UNIT#0  175610  300
                                UNIT#1  175620  310
                                UNIT#3  175640  330
                                CONSOLE 177560   60

                        OPTIONS
                        -------


                        LOCAT ON $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

B T POSITION	DEFINITION	DEFAULT VALUE
0-3	# OF DATA BITS	10(8) = 8
4	PARITY ENABLED - (SEE	0 = NO
5	EVEN ODD PARITY-/ NOTE)	0 = ODD
6	COMMON SPEED	1 = YES
7	PROGRAMMABLE BAUD RATE	0 = NO
8-11	BAUD RATE OFFSET (SEE FOLLOWING NOTE)	02(8) = 110 BAUD
12	BREAK GENERATION ENABLED	1 = YES
13	WRAP CONNECTOR INSTALLED	0 = NO
14	MAINT JUMPER (SEE NOTE)	0 = NO
15	ERROR BITS ENABLED	0 = NO

NOTE ON BITS <4 5>

TH S DIAGNOSTIC DOES NOT TEST THE PARITY LOGIC

NOTE ON BITS <7 11>

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 (TEST #16)

NOTE ON BIT 13

THIS SW TCH WHEN ON WILL ALLOW THE TESTING OF THE EIA DRIVERS AND RECEIVERS OR THE 20 MA DRIVERS AND RECEIVERS

IN ORDER TO TEST THE EIA DRIVERS AND RECEIVERS A WRAP CONNECTOR THAT CONNECTS PINS F TO J AND PINS M TO E MUST BE INSTALLED IN THE 40-PIN HEADER

IN ORDER TO TEST THE 20MA DRIVERS AND RECEIVERS A WRAP CONNECTOR THAT CONNECTS PINS E TO H AND PINS K TO KK MUST BE INSTALLED IN THE 40-PIN HEADER

NOTE ON BIT 14

TH S SWITCH WHEN ON WILL ALLOW THE DIAGNOSTIC TO TEST IN MAINTAINCE MODE IT IS ASSUMED THAT THE MAINTAINCE JUMPER IS INSTALLED ON ALL OF THE DLV11-F MODULES WHEN THIS BIT S SET

DLV11-F INDIVIDUAL TEST REQUIREMENTS TABLE

| 1

TEST #	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20	21	22	23	SEQ 0008
CONSOLE DEVICE			++				**		++	++	++	++	++			++	++		++	++
APT ENVIRONMENT		++							++			++		++	++					
(MAINT) BIT SET			--				**			--	--	--	--	--		--	--		--	--
(WRAP CON) BIT SET							**											--		
(ERROR BITS) BIT SET													--							
(COM SPD) BIT SET																		--		
(BREAK) BIT SET		--																		--
(PROG BAUD RATE) BIT SET														--						

++ TEST WILL NOT RUN IF THIS CONDITION IS TRUE
-- TEST WILL NOT RUN IF THIS CONDITION IS FALSE
** TEST WILL NOT RUN IF ALL OF THE CONDITIONS IN THIS COLUMN ARE FALSE

```
                        2 5     EXECUTION TIMES

                                EXECUTION TIMES ARE FOR AN LSI-11 PROCESSOR WITH  ALL  OPTIONS
                                ENABLED ON THE DLV11-F (EXECPT FOR PROGRAMMABLE BAUD RATE), AT
                                110 BAUD, AND NOT AT THE CONSOLE ADDRESS

                                        FIRST PASS-         90 SECONDS
                                        ADDITIONAL PASSES   95 SECONDS
                                        ADDITIONAL DEVICES  95 SECONDS

                                THE TEST TIME IS  BAUD  RATE  DEPENDANT,   HIGHER  BAUD  GIVES
                                SHORTER PASS TIMES

                                IF THE DIAGNOSTIC IS RUN AT THE CONSOLE
                                ADDRESS THE RUNNING TIME IS 5 SECONDS PER PASS

                        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 SLU'S
                                FOR FURTHER INFORMATION THE LISTING MUST  BE  CONSULTED
                                B TS  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

                                        B T  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 TH S DIAGNOST C 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
                                        ------ ----------- -------


                                          15   14   13   12   11   10    9    8    7    6    5    4    3    2    1    0
                                        ***********************************************************************************
                                RCSR    !    !    !    !    !RCVR!    !    !    !RCVR!RCVR!    !    !    !    !    !RDR !
                                        !    !    !    !    !ACT !    !    !    !DONE!IE  !    !    !    !    !    !ENB !
                                        ***********************************************************************************
                                RBUF    !ERRO!OR  !FR  !P   !    !    !    !    !                RECEIVED DATA BUFFER          !
                                        !R   !ERR !ERR !ERR !    !    !    !    !                                              !
                                        ***********************************************************************************
                                TCSR    ! PROGRAMMABLE BAUD !PBR !    !    !    !XMIT!XMIT!    !    !    !MAIN!    !BREA!
                                        !   RATE  SELECT    !ENAB!    !    !    !RDY !IE  !    !    !    !T   !    !K   !
                                        ***********************************************************************************
                                TBUF    !    !    !    !    !    !    !    !                TRANSMITTER DATA BUFFER       !
                                        !    !    !    !    !    !    !    !                                              !
                                        ***********************************************************************************


                                                                  NOTE

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

```
                                6 0     SUMMARY OF TESTS AND SPECIAL SUBROUTINES
                                        ------- -- ----- --- ------- ---------- -


                                        TEST 1 ADDRESSABILITY
                                        ---- -

                                               THIS TEST VERIFIES THAT THE ADDRESS AS PLACED  IN  THE
                                               HARDWARE   P-TABLE  TO  BE  CORRECT  AND  THE  DLV11-F
                                               RESPONDS TO THAT ADDRESS SPACE


                                        TEST 2 BREAK - TCSR0 SET, CLEAR, RESET
                                        ---- -


                                        TEST 3 MAINT - TCSR2 SET, CLEAR, RESET
                                        ---- -


                                        TEST 4 XMITIE - TCSR6 SET, CLEAR, RESET
                                        ---- -


                                        TEST  5 RCVRIE - RCSR6 SET, CLEAR, RESET
                                        ----  -

                                        THE FOLLOWING 4 TESTS VERIFY  THAT  RESET  (INIT)  INITIALIZES
                                        READ ONLY BITS


                                        TEST  6 RCVRDONE - RCSR 7 - IS CLEARED BY INIT
                                        ---- --


                                        TEST 7 RCVRACT - RCSR 11 - 15 CLEARED BY INIT
                                        ---- -



                                        TEST 10 XMITRDY - TCSR 7 - IS SET BY INIT
                                        ---- --

                                        TEST 11 XMIT RDY - TCSR 7 - CLEARS WHEN TBUF  IS LOADED
                                        ---- --
                                               WITH  A  CHARACTER  AND  THAT IT SETS WITHIN A
                                               REASONABLE AMOUNT OF TIME


                                        TEST 12 OUTPUTTING A CHAR FROM TBUF (WITH  MAINT  SET)
                                        ---- --
                                               RESULTS   IN   RCVRDONE   SETTING   WITH N   A
                                               REASONABLE  AMOUNT  OF  TIME  AND  THAT  RESET
                                               CLEARS THE B T
```

```
                        TEST 13 RCVRDONE IS CLEARED BY READING RBUF
                        ---- --

                        TEST 14 RCVRACT - RCSR 11 - SETS WHEN A START  BIT  IS
                        ---- --
                                RECEIVED  AND  CLEARS WHEN RCVRDONE - RCSR 7 -
                                SETS


                        TEST 15 OVERRUN BIT - RBUF 14
                        ---- --


                        TEST 16 PROGRAMMABLE BAUD RATE TEST TEST AT ALL SPEEDS
                        ---- --
                                AVAILABLE  A COMPARISON WILL BE MADE TO SEE IF
                                NEW TIME IS LESS THAN PREVIOUS


                        TEST 17 TRANSMITTER INTERRUPT LOGIC TEST
                        ---- --
                                LOGICALLLY THIS IS 4 SEPARATE TESTS
                                    A) DOES TRANSMITTER INTERRUPT LOGIC WORK
                                    B) AT PRIORITY OF 0
                                    C) AND ONLY ONCE
                                    D) BUT NOT WITH INTERRUPT ENABLE CLEAR


                        TEST 20 RECEIVER INTERRUPT LOGIC TEST THIS TEST COVERS ALL
                        ---- --
                                OF  THE RECEIVER S DE OF THE INTERRUPT LOGIC IN
                                 CHARACTER MODE


                        TEST  21  TEST  ACTUAL   DATA   TRANSFERED   NON-INTERRUPT
                        ----  --
                                MAINTENANCE BIT SET


                        TEST 22 TEST DATA THROUGH WRAP
                        ---- --


                        TEST 23 FULL DATA TRANSFER WITH INTERRUPTS AND MAINTENANCE
                        ---- --
                                MODE


                        TEST 24 TEST BREAK GENERATION LOGIC  TRANSMIT  KNOWN  CHAR
                        ---- --
                                WITH BREAK SET AND COMPARE RECEIVED WITH 0


                        TEST 25 NOT A TEST - SEND BACK TO LOOP
                        ---- --
```

```
                                                NOTE

                                    FOR ALL OF THE FOLLOWING ROUTINES THE  USE
                                    OF  (R5)  IS PART OF THE LINKAGE MECHANISM
                                    BETWEEN THE CALLER AND THE CALLED


                                ROUTINE TIMER
                                ---------- --

                                    THIS ROUTINE IS USED TO TEST THE STATUS OF
                                    ANY BIT IN ANY REGISTER
                                INPUTS
                                        HOWLONG     THE  MAXIMUM  AMOUNT  OF  TIME  TO
                                                    SPEND IN THIS ROUTINE
                                        WHICHBIT    A MASK WITH THE  BIT(S)  SET  THAT
                                                    ARE TO BE CHECKED
                                        REG         A POINTER TO THE  REGISTER  TO  BE
                                                    CHECKED
                                        SETCLR      THE DESIRED RESULTS -- EITHER  SET
                                                    OR  CLEAR
                                OUTPUT.
                                        THE 'C' BIT IS SET TO INDICATE AN ERROR BUT IT
                                        IS TESTED BY THE IF ERROR STATEMENT


                                ROUTINE DATLNG
                                --------------
                                        THIS ROUTINE SETS UP A MASK FOR DATA, WITH -

                                INPUT   NOTHING IS PASSED TO THIS ROUTINE  BUT  GLOBAL
                                        INFORMATION IS ASSUMED TO EXIST
                                        $USWR-- THE WORD FOR SOFTWARE PARAMETERS
                                        DATA-- A MASK FOR THE LOCATION  OF  THE  OCTAL
                                        NUMBER OF DATA BITS

                                OUTPUT----
                                        MASK-- A MASK OF BINARY ZEROS  RIGHT-JUST FIED
                                        THE NUMBER OF WHICH IS DEFINED IN $USWR WORD


                                ROUTINE WAIT
                                ------------
                                        THIS ROUTINE IS USED TO DELAY EXECUTION OF THE
                                        MAIN  PROGRAM  FOR A SPECIFIED AMOUNT OF TIME
                                        THIS  IS  ACCOMPLISHED  BY  INCREMENTING  A
                                        REGISTER UP TO A LIMIT  THE INNER LOOP IS SET
                                        TO APPROXIMATE 1 MICRO SEC


                                SERVICE ROUTINE   INTSRV
                                ------- --------  ------
                                        TH S GLOBAL ROUTINE DOES NOTHING BUT INCREMENT
```

```
                        'INTFLAG'  EACH TIME IT IS CALLED   IT ASSUMES
                        THAT THE MAIN CALLING ROUTINE WILL  KNOW  WHAT
                        TO LOOK FOR


                ROUTINE CYCLE
                -------------
                        THIS ROUTINE  CAUSES  ADRS  TO  POINT  TO  THE
                        ADDRESS  OF  DLV11-F  UNDER  TEST,  ADRS +2 TO
                        POINT TO THE VECTOR OF THE DLV11-F UNDER TEST
                        IT  KEEPS  TRACK OF THE CURRENT DEVICE AND BIT
                        MASKS
```

```
                                ;
                                 TITLE  MAINDEC-ZZ-CVDVCB-
                                ;*COPYRIGHT (C) 1977
                                ;*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

                                ;*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

                                ;*PRIORITY LEVEL DEFINITIONS
000000                          PR0=    0               ;;PRIORITY LEVEL 0
```

```
                                PR1=    40              ;;PRIORITY LEVEL 1
                                PR2=    100             ;;PRIORITY LEVEL 2
                                PR3=    140             ;;PRIORITY LEVEL 3
                                PR4=    200             ;;PRIORITY LEVEL 4
                                PR5=    240             ;;PRIORITY LEVEL 5
                                PR6=    300             ;;PRIORITY LEVEL 6
                                PR7=    340             ;;PRIORITY LEVEL 7

                                ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                          SW15=   100000
040000                          SW14=   40000
020000                          SW13=   20000
010000                          SW12=   10000
004000                          SW11=   4000
002000                          SW10=   2000
001000                          SW09=   1000
000400                          SW08=   400
000200                          SW07=   200
000100                          SW06=   100
000040                          SW05=   40
000020                          SW04=   20
000010                          SW03=   10
000004                          SW02=   4
000002                          SW01=   2
000001                          SW00=   1
                                 EQUIV  SW09,SW9
                                 EQUIV  SW08,SW8
                                 EQUIV  SW07,SW7
                                 EQUIV  SW06,SW6
                                 EQUIV  SW05,SW5
                                 EQUIV  SW04,SW4
                                 EQUIV  SW03,SW3
                                 EQUIV  SW02,SW2
                                 EQUIV  SW01,SW1
                                 EQUIV  SW00,SW0

                                ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                          BIT15=  100000
040000                          BIT14=  40000
020000                          BIT13=  20000
010000                          BIT12=  10000
004000                          BIT11=  4000
002000                          BIT10=  2000
001000                          BIT09=  1000
000400                          BIT08=  400
000200                          BIT07=  200
000100                          BIT06=  100
000040                          BIT05=  40
000020                          BIT04=  20
000010                          BIT03=  10
000004                          BIT02=  4
000002                          BIT01=  2
000001                          BIT00=  1
                                 EQUIV  BIT09,BIT9
                                 EQUIV  BIT08,BIT8
                                 EQUIV  BIT07,BIT7
```

```
                        EQUIV  BIT06,BIT6
                        EQUIV  BIT05,BIT5
                        EQUIV  BIT04,BIT4
                        EQUIV  BIT03,BIT3
                        EQUIV  BIT02,BIT2
                        EQUIV  BIT01,BIT1
                        EQUIV  BIT00,BIT0

                       ,*BASIC "CPU" TRAP VECTOR ADDRESSES
000004                 ERRVEC= 4              ,,TIME OUT AND OTHER ERRORS
000010                 RESVEC= 10             ,,RESERVED AND ILLEGAL INSTRUCTIONS
000014                 TBITVEC=14             ,,"T" BIT
000014                 TRTVEC= 14             ,,TRACE TRAP
000014                 BPTVEC= 14             ,,BREAKPOINT TRAP (BPT)
000020                 IOTVEC= 20             ,,INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                 PWRVEC= 24             ,,POWER FAIL
000030                 EMTVEC= 30             ,,EMULATOR TRAP (EMT) **ERROR**
000034                 TRAPVEC=34             ,,"TRAP" TRAP
000060                 TKVEC=  60             ,,TTY KEYBOARD VECTOR
000064                 TPVEC=  64             ,,TTY PRINTER VECTOR
000240                 PIRQVEC=240            ,,PROGRAM INTERRUPT REQUEST VECTOR
000004                          ILLMEM= 4
000001                          ADRS=   R1
000002                          GOOD=   R2
000003                          BAD=    R3
000001                          REGISTER=R1
000002                          BIT=    R2
000003                          FUNCT=  R3
000002                          LEAD=   R2
000004                          FOLLOW= R4
175610                          DLADDR= 175610

                                , THE FOLLOWING DEFINITIONS APPLY TO THE GLOBAL SUBS
177777                          SET=    -1
000000                          CLR=    0


                       ,,*****************************************************************
                                , RCSR REGISTER BIT NAMES
                       ,,*****************************************************************
                                , UNUSED       BIT15
                                , UNUSED       BIT14
                                , UNUSED       BIT13
                                , UNUSED       BIT12
004000                          RCVRACT=       BIT11        , RECEIVER ACTIVE INDICATOR
                                , UNUSED       BIT10
                                , UNUSED       BIT09
                                , UNUSED       BIT08
000200                          RCVRDONE=      BIT07        , RECEIVER DONE
000100                          RCVRIE=        BIT06        , RECEIVER INTERRUPT ENABLE
                                , UNUSED       BIT05
                                , UNUSED       BIT04
                                , UNUSED       BIT03
                                , UNUSED       BIT02
                                , UNUSED       BIT01
000001                          RDRRUN=        BIT00        , READER RUN
```

```
                        ;*****************************************************************
                        ; 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  B TS
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 INTERPUPT 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        B T11
                        ; UNUSED        B T10
                        ; UNUSED        BIT09
                        ; UNUSED        B T08
000200                  TDATA7=         BIT07           ;
000100                  TDATA6=         B T06           ;
```

```
          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                           WRAP    =      20000
          040000                           MAINTJUMP =    40000
          100000                           ERRBITS =     100000
                                   ;;*************************************************************
                                    SBTTL  TRAP CATCHER

          000000                         =0
                                   ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A " +2,HALT"
                                   ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                   ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
          000174                         =174
  000174  000000                   DISPREG   WORD  0               ;;SOFTWARE DISPLAY REGISTER
  000176  000000                   SWREG     WORD  0               ;;SOFTWARE SWITCH REGISTER
                                    SBTTL  STARTING ADDRESS(ES)
  000200  000137  001336                   JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
```

```
                        SBTTL  ACT11 HOOKS

                ;;*****************************************************************
                ;HOOKS REQUIRED BY ACT11
                        $SVPC=                  ;SAVE PC
000204                   =46
000046                  $ENDAD                  ;;1)SET LOC 46 TO ADDRESS OF $ENDAD IN  $EOP
011404                   =52
000F52                   WORD   0               ;;2)SET LOC 52 TO ZERO
000000                   =$SVPC                 ;; RESTORE PC
000204          =1000
001000          SBTTL  APT PARAMETER BLOCK

                ;;*****************************************************************
                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                ;;*****************************************************************
                        $X=     ;;SAVE CURRENT LOCATION
001000                   =24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024                  200     ;;FOR APT START UP
000200                   =44    ;;POINT TO APT INDIRECT ADDRESS PNTR
000024                  $APTHDR ;;POINT TO APT HEADER BLOCK
000044                   = $X   ;;RESET LOCATION COUNTER
001000
                ;;*****************************************************************
                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                ;INTERFACE SPEC

001000          $APTHD
001000  000000  $HIBTS   WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR
001002  001174  $MBADR   WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001004  000005  $TSTM    WORD   5       ;;RUN TIM OF LONGEST TEST
001006  000055  $PASTM   WORD   45      ;;RUN TIME IN SECS  OF 1ST PASS ON 1 UNIT (QUICK VER FY)
001010  000036  $UN TM   WORD   30      ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADD TIONAL UNIT
001012  000030          WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                                     SBTTL  COMMON TAGS

                                    ,,********************************************************************
                                    ,*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                    ,*USED IN THE PROGRAM

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

                                    ,,********************************************************************
                                     EVEN
        001174                      $MAIL                       ,,APT MAILBOX
        001174  000000              $MSGTY  WORD    AMSGTY      ,,MESSAGE TYPE CODE
        001176  000000              $FATAL  WORD    AFATAL      ,,FATAL ERROR NUMBER
        001200  000000              $TESTN  WORD    ATESTN      ,,TEST NUMBER
        001202  000000              $PASS   WORD    APASS       ,,PASS COUNT
        001204  000000              $DEVCT  WORD    ADEVCT      ,,DEVICE COUNT
        001206  000000              $UNIT   WORD    AUNIT       ,,I/O UNIT NUMBER
        001210  000000              $MSGAD  WORD    AMSGAD      ,,MESSAGE ADDRESS
```

```
 965  001212  000000                $MSGLG   WORD   AMSGLG  ,,MESSAGE LENGTH
 966  001214                      $ETABLE                  ,,APT ENVIRONMENT TABLE
 967  001214      000             $ENV     BYTE   AENV    ,,ENVIRONMENT BYTE
 968  001215      000             $ENVM    BYTE   AENVM   ,,ENVIRONMENT MODE BITS
 969  001216  000000              $SWREG   WORD   ASWREG  ,,APT SWITCH REGISTER
 970  001220  011110              $USWR    WORD   AUSWR   ,,USER SWITCHES
 971  001222  000000              $CPUOP   WORD   ACPUOP  ,,CPU TYPE,OPTIONS
 972                              ,*                      BITS 15-11=CPU TYPE
 973                              ,*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
 974                              ,*                              11/70=06,PDQ=07,Q=10
 975                              ,*                      BIT 10=REAL TIME CLOCK
 976                              ,*                      BIT  9=FLOATING POINT PROCESSOR
 977                              ,*                      BIT  8=MEMORY MANAGEMENT
 978  001224      000             $MAMS1   BYTE   AMAMS1  ,,HIGH ADDRESS,M.S.  BYTE
 979  001225      000             $MTYP1   BYTE   AMTYP1  ,,MEM  TYPE,BLK#1
 980                              ,*                      MEM TYPE BYTE    --  (HIGH BYTE)
 981                              ,*                              900 NSEC CORE=001
 982                              ,*                              300 NSEC BIPOLAR=002
 983                              ,*                              500 NSEC MOS=003
 984  001226  000000              $MADR1   WORD   AMADR1  ,,HIGH ADDRESS,BLK#1
 985                              ,*                        MEM LAST ADDR =3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
 986  001230      000             $MAMS2   BYTE   AMAMS2  ,,HIGH ADDRESS,M.S.  BYTE
 987  001231      000             $MTYP2   BYTE   AMTYP2  ;,MEM TYPE,BLK#2
 988  001232  000000              $MADR2   WORD   AMADP2  ,,MEM LAST ADDRESS,BLK#2
 989  001234      000             $MAMS3   BYTE   AMAMS3  ,,HIGH ADDRESS,M.S BYTE
 990  001235      000             $MTYP3   BYTE   AMTYP3  ,,MEM TYPE,BLK#3
 991  001236  000000              $MADR3   WORD   AMADR3  ,,MEM LAST ADDRESS,BLK#3
 992  001240      000             $MAMS4   BYTE   AMAMS4  ,,HIGH ADDRESS,M S BYTE
 993  001241      000             $MTYP4   BYTE   AMTYP4  ,,MEM TYPE,BLK#4
 994  001242  000000              $MADR4   WORD   AMADR4  ,,MEM LAST ADDRESS,BLK#4
 995  001244  000300              $VECT1   WORD   AVECT1  ,,INTERRUPT VECTOR#1,BUS PRIORITY#1
 996  001246  000000              $VECT2   WORD   AVECT2  ,,INTERRUPT VECTOR#2BUS PRIORITY#2
 997  001250  175610              $BASE    WORD   ABASE   ,,BASE ADDRESS OF EQUIPMENT UNDER TEST
 998  001252  100000              $DEVM    WORD   ADEVM   ,,DEVICE MAP
 999  001254                      $ETEND
1000                               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


001254                  $ERRTB
                        ;;              GLOBAL DATA
001254  175610              DLADD   DLADDR
001256  000300              DLVEC   300
001260  175610              RCSR    DLADDR  +  0
001262  175612              RBUF    DLADDR  +  2
001264  175614              TCSR    DLADDR  +  4
001266  175615              TCSRHI  DLADDR  +  5
001270  175616              TBUF    DLADDR  +  6
001272  000000                      0
001274  000020               BLKW 20    ;FOR R5 STACK
001334  000000              R5STACK      WORD 0
```

```
1027  001336                                START
1028                                         SBTTL  INITIALIZE THE COMMON TAGS
1029                                        ; CLEAR THE COMMON TAGS ($CMTAG) AREA
1030  001336  012706  001100                       MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1031  001342  005026                               CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1032  001344  022706  001140                       CMP     #SWR,R6 ;;DONE?
1033  001350  001374                               BNE     -6              ;;LOOP BACK IF NO
1034  001352  012706  001100                       MOV     #STACK,SP       ;;SETUP THE STACK POINTER
1035                                        ;;INITIALIZE A FEW VECTORS
1036  001356  012737  013326  000020               MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1037  001364  012737  000340  000022               MOV     #340,@#IOTVEC+2 ;;LEVEL 7
1038  001372  012737  013126  000030               MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1039  001400  012737  000340  000032               MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1040  001406  012737  014260  000034               MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1041  001414  012737  000340  000036               MOV     #340,@#TRAPVEC+2;;LEVEL 7
1042  001422  012737  011440  000024               MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
1043  001430  012737  000340  000026               MOV     #340,@#PWRVEC+2 ;;LEVEL 7
1044  001436  016767  007710  007700               MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
1045  001444  005067  177510                       CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
1046  001450  005067  177506                       CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1047  001454  112767  000001  177433               MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
1048  001462  012767  001462  177416               MOV     # ,$LPADR        ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1049  001470  012767  001470  177412               MOV     # ,$LPERR        ;;SETUP THE ERROR LOOP ADDRESS
1050                                        ;;SIZE FOR A HARDWARE SWITCH REGISTER  IF NOT FOUND OR IT IS
1051                                        ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER
1052  001476  013746  000004               MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1053  001502  012737  001536  000004               MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
1054  001510  012767  177570  177422               MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1055  001516  012767  177570  177416               MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTEF
1056  001524  022777  177777  177406               CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
1057  001532  001012                               BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1058                                                                        ;;AND  THE HARDWARE SWR IS NOT = -1
1059  001534  000403                               BR      65$             ;;BRANCH IF NO TIMEOUT
1060  001536  012716  001544          64$          MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1061  001542  000002                               RTI
1062  001544  012767  000176  177366  65$          MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1063  001552  012767  000174  177362               MOV     #DISPREG,DISPLAY
1064  001560  012637  000004          66$          MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1065
1066  001564  005067  177412                       CLR     $F              ;;CLEAR PASS COUNT
1067  001570  132767  000200  177417               BITB    #AP SIZE,$ENVM  ;;TEST USER SIZE UNDER APT
1068  001576  001403                               BEQ     67$             ;;YES,USE NON-APT SWITCH
1069  001600  012767  001216  177332               MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
1070  001606                          67$
1071                                         SBTTL  TYPE PROGRAM NAME
1072                                        ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
1073  001606  005227  177777                       INC     #-1             ;;FIRST TIME?
1074  001612  001037                               BNE     68$             ;;BRANCH IF NO
1075  001614  022737  011404  000042               CMP     #$ENDAD,@#42    ;;ACT-11?
1076  001622  001433                               BEQ     68$             ;;BRANCH IF YES
1077  001624  104401  001672                       TYPE    ,69$            ;;TYPE ASCIZ STRING
1078                                         SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1079  001630  005737  000042                       TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
1080  001634  001012                               BNE     70$             ;;BRANCH IF YES
1081  001636  126727  177352  000001               CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
1082  001644  001406                               BEQ     70$             ;;BRANCH IF YES
```

```
 1083  001646  026727  177266  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
 1084  001654  001005                          BNE     71$             ;;BRANCH IF NO
 1085  001656  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
 1086  001660  000403                          BR      71$
 1087  001662  112767  000001  177244  70$     MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
 1088  001670                          71$
 1089  001670  000410                          BR      68$             ;;GET OVER THE ASCIZ
 1090                                  ;;69$    ASCIZ   <CRLF>*MD-ZZ-CVDVC-B*<CRLF>
 1091  001712                          68$
```

```
1092  001712                                                    WHILE $DEVM EQ #0 DO
1093  001712                          $1
1094  001712  005767  177334                TST     $DEVM
1095  001716  001101                        BNE     $2
1096  001720                                  TYPTXT <<CRLF>'I HAVE NO DEVICE TO TEST '>
1097  001762                                TYPTXT  <<CRLF>'SET UP $DEVM TO INDICATE ACTUAL CONFIGURATION '>
1098  002050                                TYPTXT  <<CRLF>'TYPE PROCEED (P) TO CONTINUE '>
1099  002116  000000                        HALT
1100  002120                                                    ENDDO
1101  002120  000674                        BR      $1
1102  002122                          $2
1103  002122                                                    LET INITFLAG  = #1
1104  002122  012767  000001  006754        MOV     #1,INITFLAG
1105  002130                                                    LET BITMASK   = #BIT15    ; START AT CONSOLE
1106  002130  012767  100000  006744        MOV     #BIT15,BITMASK
1107  002136                          LOOP
1108  002136                                CALL    CYCLE           ; NO ARGUMENTS--ADDRS -> NEXT ADDRESS
1109  002136  004767  006520                JSR     PC,CYCLE
1110                                                      ;               ADDRS+2 -> NEXT VECTOR
1111                                                      ;GET UNIT ADDRESS
1112  002142                                                    LET     DLADD  = (ADRS)+
1113  002142  012167  177106                MOV     (ADRS)+,DLADD
1114                                                      ;GET UNIT VECTOR
1115  002146                                                    LET     DLVEC  = (ADRS)
1116  002146  011167  177104                MOV     (ADRS),DLVEC
1117  002152                                                    LET     ADRS   = DLADD
1118  002152  016701  177076                MOV     DLADD,ADRS
1119                                                      ;RCSR = DLADD  +  0
1120  002156                                                LET     RCSR   = DLADD
1121  002156  016767  177072  177074        MOV     DLADD,RCSR
1122  002164                                                LET     RBUF   = DLADD + #2
1123  002164  016767  177064  177070        MOV     DLADD,RBUF
1124  002172  062767  000002  177062        ADD     #2,RBUF
1125  002200                                                LET     TCSR   = DLADD + #4
1126  002200  016767  177050  177056        MOV     DLADD,TCSR
1127  002206  062767  000004  177050        ADD     #4,TCSR
1128  002214                                                LET     TCSRHI  = DLADD + #5
1129  002214  016767  177034  177044        MOV     DLADD,TCSRHI
1130  002222  062767  000005  177036        ADD     #5,TCSRHI
1131  002230                                                LET     TBUF   = DLADD + #6
1132  002230  016767  177020  177032        MOV     DLADD,TBUF
1133  002236  062767  000006  177024        ADD     #6,TBUF
1134  002244                                                LET     R5     = #R5STACK
1135  002244  012705  001334                MOV     #R5STACK,R5
1136                                                      ;,BRESET
1137  002250  000005                        RESET
```

```
1138                                    ;;**********************************************************************
1139                                    ;*TEST 1         ADDRESSABILITY
1140                                    ;*              THIS TEST VERIFIES THAT THE ADDRESS AS PLACED IN
1141                                    ;*              THE HARDWARE P-TABLE TO BE CORRECT AND THE DLV11-F RESPONDS
1142                                    ;*              TO THAT ADDRESS SPACE
1143                                    ;;**********************************************************************
1144  002252  000004                   TST1    SCOPE
1145  002254  012767  000002  176676           MOV     #2,$TIMES       ;;DO 2 ITERATIONS
1146  002262  012767  000001  176710           MOV     #1,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1147  002270                                                                   LET     ADRS  = DLADD
1148  002270  016701  176760                   MOV     DLADD,ADRS
1149                                                                             ; SET UP INTERRUPT
1150  002274                                                                   SETVEC  #ILLMEM,#INTSRV,#PR7
1151  002274  010146                           MOV     R1,-(SP)
1152  002276  012701  000004                   MOV     #ILLMEM,R1
1153  002302  012721  010652                   MOV     #INTSRV,(R1)+
1154  002306  012711  000340                   MOV     #PR7,(R1)
1155  002312  012601                           MOV     (SP)+,R1
1156  002314                                                                   LET     I  = #0
1157  002314  005067  176752                   CLR     I
1158  002320                                                                   REPEAT
1159  002320                           $3
1160  002320                                                                       BGNSUB
1161  002320  012767  002326  176562           MOV     #64$,$LPERR
1162                                                                                 ;CLEAR FLAG
1163  002326                                                                         LET INTFLAG  = #0
1164  002326  005067  006326                   CLR     INTFLAG
1165
1166                                                                                 ;READ FLAG
1167  002332  005711                           TST @ADRS
1168  002334                                                                         IF INTFLAG NE #0 THEN
1169  002334  005767  006320                   TST     INTFLAG
1170  002340  001401                           BEQ     $4
1171                                                                                 ; FATAL ERROR
1172  002342                                                                           ERRDF 1,,NODL
1173  002342  104001                           ERROR   1
1174  002344                                                                         ENDIF
1175  002344                           $4
1176  002344                                                                     ENDSUB
1177  002344                                                                     LET     I  =   + #2
1178  002344  062767  000002  176720           ADD     #2,I
1179  002352                                                                     LET     ADRS  = DLADD +
1180  002352  016701  176676                   MOV     DLADD,ADRS
1181  002356  066701  176710                   ADD     I,ADRS
1182  002362                                                                UNT L I EQ #8
1183  002362  026727  176704  000010           CMP     I,#8
1184  002370  001353                           BNE     $3
1185  002372                                                                CLRVEC ILLMEM
1186  002372  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
1187  002374  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
1188  002376  012701  000004                   MOV     #ILLMEM,R1
1189  002402  010102                           MOV     R1,R2
1190  002404  062702  000002                   ADD     #2,R2
1191  002410  010221                           MOV     R2,(R1)+
1192  002412  005011                           CLR     (R1)
1193  002414  012602                           MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
1194  002416  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
1195                                                          ;END OF TEST
1196  002420                              ENDTST
```

```
                                ,,*****************************************************************
                                ,* THE FOLLOWING 8 TESTS  TEST ALL 'READ WRITE' BITS
                                ,,*****************************************************************


                                ,,*****************************************************************
                                ,*TEST 2        BREAK - TCSR0 SET, CLEAR, RESET
                                ,*              THE BREAK BIT IS USUALLY USED ON THE CONSOLE
                                ,*              DEVICE  IF ADDITIONAL DLV OPTIONS ARE USED
                                 *              IT IS RECOMMENDED TO REMOVE THE 'BG' JUMPER AND
                                ,*              CLEAR BIT 12 IN $USWR WHICH WILL CAUSE THIS
                                ,*              TEST TO BE SKIPPED
                                ,,*****************************************************************
002420  000004                  TST2    SCOPE
002422  012767  000010  176530          MOV     #10,$TIMES      ,,DO 10 ITERATIONS
002430  012767  000002  176542          MOV     #2,$TESTN       ,,SET TEST NUMBER IN APT MAIL BOX

00243o                                                          IF #BRK NOTSETIN $USWR OR #APTENV SETIN $ENV THE
002436  032767  010000  176554          BIT     #BRK,$USWR
002444  001404                          BEQ     $5
002446  032767  000001  176540          B T     #APTENV,$ENV
002454  001404                          BEQ     $6
002456                          $5
002456                                                              EXIT TEST   , BREAK NOT INSTALLED
002456  012767  000001  176474          MOV     #1,$TIMES
002464  000452                          BR      TST3            ,,,EXIT THIS TEST
002466                                                          ENDIF
002466                          $6

                                                        , SEE IF IT IS CLEAR
002466                                                  BGNSUB
002466  012767  002474  176414          MOV     #64$,$LPERR

002474                                                      IF      #BREAK SETIN @TCSR THEN
002474  032777  000001  176562          B T     #BREAK,@TCSR
002502  001401                          BEQ     $7

                                                            , BREAK DID NOT RESET IN TCSR
002504                                                      ERRHRD 2,,DIDNOT
002504  104002                          ERROR   2
002506                                                  ENDIF
002506                          $7
002506                                                  ENDSUB

                                                        , TRY TO SET BREAK BIT
002506                                                  BGNSUB
002506  012767  002514  176374          MOV     #64$,$LPERR
002514                                                      LET     @TCSR  = @TCSR SET BY #BREAK
002514  052777  000001  176542          BIS     #BREAK,@TCSR

                                                            , STUCK TO 0
002522                                                      IF      #BREAK NOTSETIN @TCSR THEN
002522  032777  000001  176534          BIT     #BREAK,@TCSR
002530  001001                          BNE     $10

                                                            , BREAK DID NOT SET IN TCSR
002532                                                      ERRHRD 3,,DIDNOT
002532  104003                          ERROR   3
002534                                                  ENDIF
```

```
1253  002534                          $10
1254  002534                                                          ENDSUB
1255
1256                                                          , TRY TO CLEAR A SET BIT
1257  002534                                                          BGNSUB
1258  002534  012767  002542  176346          MOV     #64$,$LPERR
1259
1260  002542                                                              LET     @TCSR  = @TCSR CLR BY #BREAK
1261  002542  042777  000001  176514          BIC     #BREAK,@TCSR
1262                                                                  , SHOULD HAVE CLEARED
1263  002550                                                              IF      #BREAK SETIN @TCSR THEN
1264  002550  032777  000001  176506          B T     #BREAK,@TCSR
1265  002556  001401                          BEQ     $11
1266                                                                      , BREAK  DID NOT CLEAR 'N TCSR
1267  002560                                                              ERRHRD 4,,D DNOT
1268  002560  104004                          ERROR   4
1269  002562                                                              ENDIF
1270  002562                          $11
1271  002562                                                          ENDSUB
1272
1273                                                          , NOW SEE IF RESET CLEARS IT
1274  002562                                                          BGNSUB
1275  002562  012767  002570  176320          MOV     #64$,$LPERR
1276
1277  002570                                                              LET     @TCSR  = @TCSR SET BY #BREAK
1278  002570  052777  000001  176466          BIS     #BREAK,@TCSR
1279                                                                  , ISSUE BUS RESET
1280  002576                                                              BRESET
1281  002576  000005                          RESET
1282  002600                                                              IF      #BREAK SETIN @TCSR THEN
1283  002600  032777  000001  176456          BIT     #BREAK,@TCSR
1284  002606  001401                          BEQ     $12
1285                                                                      , BREAK DID NOT RESET IN TCSR
1286  002610                                                              ERRHRD 5,,DIDNOT
1287  002610  104005                          ERROR   5
1288  002612                                                              ENDIF
1289  002612                          $12
1290  002612                                                      ENDSUB
1291  002612                                                  ENDTST
1292
1293
```

```
1294                                    ;;**********************************************************************
1295                                    ;;**********************************************************************
1296                                    ;*TEST 3          MAINT - TCSR2 SET, CLEAR, RESET
1297                                    ;;**********************************************************************
1298  002612  000004                    TST3    SCOPE
1299  002614  012767  000010  176336            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1300  002622  012767  000003  176350            MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1301
1302  002630                                                    IF #MAINTJUMP NOTSETIN $USWR ORB CONSOLE EQ #TRU
1303  002630  032767  040000  176362            B T     #MAINTJUMP,$USWR
1304  002636  001404                            BEQ     $13
1305  002640  126727  006255  000001            CMPB    CONSOLE,#TRUE
1306  002646  001004                            BNE     $14
1307  002650                            $13
1308  002650                                                       EXIT TEST
1309  002650  012767  000001  176302            MOV     #1,$TIMES
1310  002656  000452                            BR      TST4            ;;;EXIT THIS TEST
1311  002660                                                    ENDIF
1312  002660                            $14
1313
1314                                                ; SEE IF IT IS CLEAR
1315  002660                                                    BGNSUB
1316  002660  012767  002666  176222            MOV     #64$,$LPERR
1317
1318  002666                                                       IF      #MAINT SETIN @TCSR THEN
1319  002666  032777  000004  176370            BIT     #MAINT,@TCSR
1320  002674  001401                            BEQ     $15
1321                                                          ; MAINT DID NOT RESET IN TCSR
1322  002676                                                    ERRHRD 6,,DIDNOT
1323  002676  104006                            ERROR   6
1324  002700                                                 ENDIF
1325  002700                            $15
1326  002700                                              ENDSUB
1327
1328                                                ; TRY TO SET MAINT BIT
1329  002700                                                    BGNSUB
1330  002700  012767  002706  176202            MOV     #64$,$LPERR
1331  002706                                                    LET     @TCSR  = @TCSR SET BY #MAINT
1332  002706  052777  000004  176350            BIS     #MAINT,@TCSR
1333                                                          ; STUCK TO 0
1334  002714                                                    IF      #MAINT NOTSETIN @TCSR THEN
1335  002714  032777  000004  176342            BIT     #MAINT,@TCSR
1336  002722  001001                            BNE     $16
1337                                                          ; MAINT DID NOT SET IN TCSR
1338  002724                                                    ERRHRD 7,,DIDNOT
1339  002724  104007                            ERROR   7
1340  002726                                                 ENDIF
1341  002726                            $16
1342  002726                                              ENDSUB
1343
1344                                                ; TRY TO CLEAR A SET BIT
1345  002726                                                    BGNSUB
1346  002726  012767  002734  176154            MOV     #64$,$LPERR
1347
1348  002734                                                    LET     @TCSR  = @TCSR CLR BY #MAINT
1349  002734  042777  000004  176322            B C     #MAINT,@TCSR
```

```
1350                                                                    , SHOULD HAVE CLEARED
1351  002742                                                          IF      #MAINT SETIN @TCSR THEN
1352  002742  032777  000004  176314          BIT     #MAINT,@TCSR
1353  002750  001401                          BEQ     $17
1354                                                                    , MAINT DID NOT CLEAR INTCSR
1355  002752                                                            ERRHRD 10,,DIDNOT
1356  002752  104010                          ERROR   10
1357  002754                                                          ENDIF
1358  002754                          $17
1359  002754                                                        ENDSUB
1360
1361                                                    , NOW SEE IF RESET CLEARS IT
1362  002754                                                        BGNSUB
1363  002754  012767  002762  176126          MOV     #64$,$LPERR
1364
1365  002762                                                          LET     @TCSR  = @TCSR SET BY #MAINT
1366  002762  052777  000004  176274          B S     #MAINT,@TCSR
1367                                                                  , ISSUE BUS RESET
1368  002770                                                        BRESET
1369  002770  000005                          RESET
1370  002772                                                          IF      #MAINT SETIN @TCSR THEN
1371  002772  032777  000004  176264          B T     #MAINT,@TCSR
1372  003000  001401                          BEQ     $20
1373                                                                  , MAINT DID NOT RESET IN TCSR
1374  003002                                                            ERRHRD 11,,DIDNOT
1375  003002  104011                          ERROR   11
1376  003004                                                          ENDIF
1377  003004                          $20
1378  003004                                                        ENDSUB
1379  003004                                                      ENDTST
1380
1381
1382
```

```
1383                                    ;;**********************************************************************
1384                                    ;;**********************************************************************
1385                                    ;*TEST 4          XMITIE - TCSR6 SET, CLEAR, RESET
1386                                    ;;**********************************************************************
1387  003004  000004                   TST4    SCOPE
1388  003006  012767  000010  176144           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1389  003014  012767  000004  176156           MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1390                                                                    ; USE PRIORITY OF 7
1391  003022  012746  000340                   MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
1392  003026  012746  003034                   MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
1393  003032  000002                           RTI                     ;;POP NEW PC AND PS
1394  003034                           64$
1395
1396                                                    ; SEE IF IT IS CLEAR
1397  003034                                                  BGNSUB
1398  003034  012767  003042  176046           MOV     #65$,$LPERR
1399
1400  003042                                                      IF      #XMITIE SETIN @TCSR THEN
1401  003042  032777  000100  176214           BIT     #XMITIE,@TCSR
1402  003050  001401                           BEQ     $21
1403                                                        ; XMITIE DID NOT RESET IN TCSR
1404  003052                                                  ERRHRD 12,,DIDNOT
1405  003052  104012                           ERROR   12
1406  003054                                                  ENDIF
1407  003054                           $21
1408  003054                                                  ENDSUB
1409
1410                                                  ; TRY TO SET XMITIE BIT
1411  003054                                                  BGNSUB
1412  003054  012767  003062  176026           MOV     #64$,$LPERR
1413  003062                                                      LET     @TCSR  = @TCSR SET BY #XMITIE
1414  003062  052777  000100  176174           B S     #XMITIE,@TCSR
1415                                                      ; STUCK TO 0
1416  003070                                                  IF      #XM TIE NOTSETIN @TCSR THEN
1417  003070  032777  000100  176166           B T     #XM TIE,@TCSR
1418  003076  001001                           BNE     $22
1419                                                        ; XMIT DID NOT RESET IN TCSR
1420  003100                                                  ERRHRD 13,,DIDNOT
1421  003100  104013                           ERROR   13
1422  003102                                                  ENDIF
1423  003102                           $22
1424  003102                                                  ENDSUB
1425
1426                                                  ; TRY TO CLEAR A SET BIT
1427  003102                                                  BGNSUB
1428  003102  012767  003110  176000           MOV     #64$,$LPERR
1429
1430  003110                                                      LET     @TCSR  = @TCSR CLR BY #XMITIE
1431  003110  042777  000100  176146           BIC     #XMITIE,@TCSR
1432                                                      ; SHOULD HAVE CLEARED
1433  003116                                                  IF      #XMITIE SETIN @TCSR THEN
1434  003116  032777  000100  176140           BIT     #XMITIE,@TCSR
1435  003124  001401                           BEQ     $23
1436                                                        ; XMIT DID NOT CLEAR IN TCSR
1437  003126                                                  ERRHRD 14,,DIDNOT
1438  003126  104014                           ERROR   14
```

```
003130                                                                  ENDIF
003130                              $23
003130                                                              ENDSUB

                                                     , NOW SEE IF RESET CLEARS IT
003130                                                              BGNSUB
003130  012767  003136  175752          MOV     #64$,$LPERR

003136                                                                  LET     @TCSR  = @TCSR SET BY #XMITIE
003136  052777  000100  176120          BIS     #XMITIE,@TCSR
                                                                            , ISSUE BUS RESET
003144                                                                  BRESET
003144  000005                          RESET
003146                                                                  IF      #XMITIE SETIN @TCSR THEN
003146  032777  000100  176110          BIT     #XMITIE,@TCSR
003154  001401                          BEQ     $24

                                                                            , XMIT DID NOT RESET IN TCSR
003156                                                                      ERRHRD 15,,DIDNOT
003156  104015                          ERROR   15
003160                                                                  ENDIF
003160                              $24
003160                                                              ENDSUB
003160                                                          ENDTST
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 5          RCVRIE - RCSR6 SET, CLEAR, RESET
                                ;;*****************************************************************
003160  000004                  TST5    SCOPE
003162  012767  000010  175770          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003170  012767  000005  176002          MOV     #5,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
                                                        ; SEE IF  T  S CLEAR
003176                                                  BGNSUB
003176  012767  003204  175704          MOV     #64$,$LPERR

003204                                                      IF      #RCVRIE SETIN @RCSR THEN
003204  032777  000100  176046          B T     #RCVRIE,@RCSR
003212  001401                          BEQ     $25
                                                            ; RCVRIE DID NOT RESET IN RCSR
003214                                                      ERRHRD 35,,DIDNOT
003214  104035                          ERROR   35
003216                                                  ENDIF
003216                          $25
003216                                                ENDSUB

                                                ; TRY TO SET RCVRIE BIT
003216                                                  BGNSUB
003216  012767  003224  175664          MOV     #64$,$LPERR
003224                                                      LET     @RCSR  = @RCSR SET BY #RCVR E
003224  052777  000100  176026          B S     #RCVRIE,@RCSR
                                                            ; STUCK TO 0
003232                                                      IF      #RCVRIE NOTSETIN @RCSR THEN
003232  032777  000100  176020          B T     #RCVR E,@RCSR
003240  001001                          BNE     $26
                                                            ; RCVRIE DID NOT SET IN RCSR
003242                                                      ERRHRD 36,,D.DNOT
003242  104036                          ERROR   36
003244                                                  ENDIF
003244                          $26
003244                                                ENDSUB

                                                ; TRY TO CLEAR A SET BIT
003244                                                  BGNSUB
003244  012767  003252  175636          MOV     #64$ $LPERR

003252                                                      LET     @RCSR  = @RCSR CLR BY #RCVRIE
003252  042777  000100  176000          BIC     #RCVRIE,@RCSR
                                                            ; SHOULD HAVE CLEARED
003260                                                      IF      #RCVRIE SETIN @RCSR THEN
003260  032777  000100  175772          BIT     #RCVRIE,@RCSR
003266  001401                          BEQ     $27
                                                            ; RCVRIE DID NOT CLEAR IN RCSR
003270                                                      ERRHRD 37,,DIDNOT
003270  104037                          ERROR   3
003272                                                  ENDIF
003272                          $27
003272                                                ENDSUB

                                                ; NOW SEE  F RESET CLEARS IT
003272                                                  BGNSUB
```

```
 1521  003272  012767  003300  175610          MOV     #64$,$LPERR
 1522
 1523  003300                                                          LET     @RCSR  = @RCSR SET BY #RCVRIE
 1524  003300  052777  000100  175752          B S     #RCVRIE,@RCSR
 1525                                                                        , ISSUE BUS RESET
 1526  003306                                                          BRESET
 1527  003306  000005                          RESET
 1528  003310                                                          IF      #RCVRIE SETIN @RCSR THEN
 1529  003310  032777  000100  175742          B T     #RCVRIE,@RCSR
 1530  003316  001401                          BEQ     $30
 1531                                                                       , RCVRIE DID NOT RESET IN RCSR
 1532  003320                                                              ERRHRD 40,,DIDNOT
 1533  003320  104040                          ERROR   40
 1534  003322                                                          ENDIF
 1535  003322                          $30
 1536  003322                                                          CKLOOP
 1537  003322                                                      ENDSUB
 1538  003322                                                  ENDTST
 1539
 1540
 1541
 1542
```

```
1543                                ,,*****************************************************************
1544                                ,* THE FOLLOWING 4 TESTS VERIFY
1545                                ,* THAT RESET (INIT) INITIALIZES READ ONLY BITS
1546                                ,,*****************************************************************
1547                                ,,*****************************************************************
1548                                ,*TEST 6          TEST THAT RCVRDONE - RCSR 7 - IS CLEARED BY INIT
1549                                ,,*****************************************************************
1550  003322  000004                TST6    SCOPE
1551  003324  012767  000010  175626        MOV     #10,$TIMES      ,,DO 10 ITERATIONS
1552  003332  012767  000006  175640        MOV     #6,$TESTN       ,,SET TEST NUMBER IN APT MAIL BOX
1553
1554
1555
1556
1557  003340                                                                BGNSUB
1558  003340  012767  003346  175542        MOV     #64$,$LPERR
1559  003346                                          IF      #RCVRDONE SETIN @RCSR THEN
1560  003346  032777  000200  175704        B T     #RCVRDONE,@RCSR
1561  003354  001402                        BEQ     $31
1562
1563                                                        ,RCVRDONE SHOULD HAVE CLEARED BY INIT
1564                                                        , RCVRDONE DID NOT CLEAR IN RCSR
1565  003356                                                ERRHRD 41,HRESET, DIDNOT
1566  003356  104041                        ERROR   41
1567                                                        ,REISSUE RESET
1568  003360                                                BRESET
1569  003360  000005                        RESET
1570  003362                                          ENDIF
1571  003362                        $31
1572                                                        ,ALLOW LOOPING AFTER ERROR
1573  003362                                          CKLOOP
1574  003362                                      ENDSUB
1575  003362                                ENDTST
1576
1577
1578
```

```
1579                                    ;;**********************************************************************
1580                                    ;;**********************************************************************
1581                                    ;*TEST 7         TEST THAT RCVRACT - RCSR 11 - IS CLEARED BY INIT
1582                                    ;;**********************************************************************
1583  003362  000004                    TST7    SCOPE
1584  003364  012767  000010  175566            MOV     #10.,$TIMES         ;;DO 10 ITERATIONS
1585  003372  012767  000007  175600            MOV     #7.,$TESTN          ;;SET TEST NUMBER IN APT MAIL BOX
1586
1587
1588
1589
1590  003400                                                                IFB CONSOLE EQ #TRUE THEN
1591  003400  126727  005515  000001            CMPB    CONSOLE,#TRUE
1592  003406  001001                            BNE     $32
1593                                                                            ;; EXECUTE TEST
1594  003410                                                                ELSE
1595  003410  000416                            BR      $33
1596  003412                            $32
1597  003412                                                                IF #WRAP SETIN $USWR THEN
1598  003412  032767  020000  175600            BIT     #WRAP,$USWR
1599  003420  001401                            BEQ     $34
1600                                                                            ;; EXECUTE TEST
1601  003422                                                                ELSE
1602  003422  000411                            BR      $35
1603  003424                            $34
1604  003424                                                                IF #MAINT SETIN $USWR THEN
1605  003424  032767  000004  175566            BIT     #MAINT,$USWR
1606  003432  001401                            BEQ     $36
1607                                                                            ;;EXECUTE TEST
1608  003434                                                                ELSE
1609  003434  000404                            BR      $37
1610  003436                            $36
1611  003436                                                                   EXIT TEST    ; LINE MUST BE TERMINATED
1612  003436  012767  000001  175514            MOV     #1,$TIMES
1613  003444  000414                            BR      TST10               ;;;EXIT THIS TEST
1614  003446                                                                ENDIF
1615  003446                            $37
1616  003446                                                                ENDIF
1617  003446                            $35
1618  003446                                                                ENDIF
1619  003446                            $33
1620
1621  003446                                                                      BGNSUB
1622  003446  012767  003454  175434            MOV     #64$,$LPERR
1623
1624  003454                                                                          IF      #RCVRACT SETIN @RCSR THEN
1625  003454  032777  004000  175576            BIT     #RCVRACT,@RCSR
1626  003462  001405                            BEQ     $40
1627
1628                                                                                    ;RESET SHOULD HAVE CLEARED RCVRACT
1629  003464                                                                            LET @TCSR  = @TCSR CLR BY #MAINT
1630  003464  042777  000004  175572            BIC     #MAINT,@TCSR
1631  003472                                                                            ERRHRD 44, HRESET, DIDNOT
1632  003472  104044                            ERROR   44
1633
1634                                                                                    ;TESTING EFFECT OF RESET ON BIT
```

```
1635
1636                                                        ;RCVRACT DID NOT CLEAR IN RCSR
1637
1638                                                        ;ALLOW ANOTHER TRY
1639  003474                                                BRESET
1640  003474  000005                    RESET
1641  003476                                              ENDIF
1642  003476                    $40
1643                                                        ;ALLOW LOOPING ON ERROR
1644  003476                                              CKLOOP
1645  003476                                          ENDSUB
1646  003476                                      ENDTST
1647
```

```
                                        ;;*****************************************************************
                                        ;;*****************************************************************
                                        ;*TEST 10        TEST THAT XMITRDY - TCSR 7 - IS SET BY INIT
                                        ;;*****************************************************************
003476  000004                          TST10:  SCOPE
003500  012767  000010  175452                  MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003506  012767  000010  175464                  MOV     #10,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX




003514                                                                          BGNSUB
003514  012767  003522  175366                  MOV     #64$,$LPERR

003522                                                                              IF      #XMITRDY NOTSETIN @TCSR THEN
003522  032777  000200  175534                  BIT     #XMITRDY,@TCSR
003530  001002                                  BNE     $41

                                                                                        ;RESET SHOULD HAVE SET BIT
                                                                                        ;XMITRDY DID NOT SET IN TCSR (AFTER RESE
003532                                                                                  ERRHRD 42,HRESET,DIDNOT
003532  104042                                  ERROR   42

                                                                                        ;ISSUE ANOTHER RESET
003534                                                                                  BRESET
003534  000005                                  RESET
003536                                                                              ENDIF
003536                                  $41:

                                                                                        ;ALLOW LOOPING ON ERROR
003536                                                                              CKLOOP
003536                                                                          ENDSUB
003536                                                                      ENDTST



```

```
                                        ;;*****************************************************************
                                        ;;*****************************************************************
                                        ;*TEST 11            TEST THAT XMIT RDY - TCSR 7 - CLEARS
                                        ;*                   WHEN TBUF IS LOADED WITH A CHARACTER
                                        ;*                   AND THAT IT SETS WITHIN A REASONABLE AMOUNT OF TIME
                                        ;;*****************************************************************
003536  000004                          TST11   SCOPE
003540  012767  000001  175412                  MOV     #1,$TIMES       ;;DO 1 ITERATION
003546  012767  000011  175424                  MOV     #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

003554                                                                          IFB CONSOLE EQ #TRUE OR #APTENV SETIN $ENV THEN
003554  126727  005341  000001                  CMPB    CONSOLE,#TRUE
003562  001404                                  BEQ     $42
003564  032767  000001  175422                  BIT     #APTENV,$ENV
003572  001404                                  BEQ     $43
003574                                  $42
003574                                                                              EXIT TEST
003574  012767  000001  175356                  MOV     #1,$TIMES
003602  000513                                  BR      TST12           ;;;EXIT THIS TEST
003604                                                                          ENDIF
003604                                  $43

003604                                                                          LET PASS  = #1  ;INIT COUNT OF TIMES THRU
003604  012767  000001  000212                  MOV     #1,PASS
003612                                                                          LOOP            ; START OF LOOP
003612                                  $44
                                                                                                ;  MAX OF 2 TIMES THRU
003612                                                                              LET ERRORFLAG  = #CLR
003612  012767  000000  000206                  MOV     #CLR,ERRORFLAG
003620                                                                              LET EXITFLAG  = #CLR
003620  012767  000000  000202                  MOV     #CLR,EXITFLAG
                                                                ; LOAD TBUF WITH ONE CHARACTER
                                                                ; WAIT FOR READY TO SET
                                                                ; (SHOULD BE VERY SHORT WAIT
                                                                ; SINCE UART DOUBLE BUFFERS ITS INPUT)

                                                                                        ;SEND A CHARACTER
003626                                                                              LET @TBUF  B= #0
003626  105077  175436                          CLRB    @TBUF
                                                                                        ;WAIT A MAXIMUM
                                                                                        ;OF 500 MSEC FOR
                                                                                        ;XMIT RDY TO SET IN TCSR
003632                                                                              CALL TIMER IN (#500,#XMITRDY,TCSR,#SET)
003632  010546                                  MOV     R5,-(SP)
003634  012745  177777                          MOV     #SET,-(R5)
003640  016745  175420                          MOV     TCSR,-(R5)
003644  012745  000200                          MOV     #XMITRDY,-(R5)
003650  012745  000500                          MOV     #500,-(R5)
003654  004767  004434                          JSR     PC,TIMER
003660  012605                                  MOV     (SP)+,R5
                                                                                        ;TIMER RETURNS AN ERROR IF BIT DID
                                                                                        ;NOT MEET CONDITION WITHIN TIME LIMIT
003662                                                                              IF ERROR THEN
003662  103001                                  BCC     $46
                                                                                        ;XMIT RDY D D NOT SET IN TCSR
003664                                                                                  ERRHRD 66,,DIDNOT
```

```
003664  104066                          ERROR   66
003666                                                                  ENDIF
003666                          $46

                                                        , LOAD TBUF WITH A SECOND CHARACTER
                                                        , CHECK IMMEDIATELY THAT XMITRDY IS CLEAR
                                                        , AND THEN WAIT FOR IT TO SET

                                                                            ,SEND SECOND CHARACTER
003666                                                                  LET @TBUF  B= #0
003666  105077  175376                  CLRB    @TBUF
003672  000240                                          NOP             , GIVE IT TIME TO CLEAR
                                                                            , XMITRDY SHOULD HAVE CLEARED UPON
                                                                            , RECEIPT OF A CHARACTER
003674                                                                  IF #XMITRDY SETIN @TCSR THEN
003674  032777  000200  175362          B T     #XMITRDY,@TCSR
003702  001404                          BEQ     $47

                                                                            , XMITRDY DID NOT CLEAR IN TCSR
003704                                                                      LET ERRORFLAG  = #SET
003704  012767  177777  000114          MOV     #SET,ERRORFLAG
                                                                                  , DEFER ERROR TYPEOUT

003712                                                              ELSE
003712  000416                          BR      $50
003714                          $47
                                                                            ,WAIT A MAXIMUM
                                                                            ,OF 500 MSEC FOR
                                                                            ,XMIT RDY TO SET IN TCSR
003714                                                                  CALL TIMER IN (#500,#XMITRDY,TCSR,#SET)
003714  010546                          MOV     R5,-(SP)
003716  012745  177777                  MOV     #SET,-(R5)
003722  016745  175336                  MOV     TCSR,-(R5)
003726  012745  000200                  MOV     #XMITRDY,-(R5)
003732  012745  000500                  MOV     #500,-(R5)
003736  004767  004352                  JSR     PC,TIMER
003742  012605                          MOV     (SP)+,R5
003744                                                                  IF EPROR THEN
003744  103001                          BCC     $51

                                                                            ,XMIT RDY D D NOT SET IN TCSR
003746                                                                      ERRHRD 70,,DIDNOT
003746  104070                          ERROR   70
003750                                                                  ENDIF
003750                          $51
003750                                                                ENDIF . OF DEFERED ERROR CALL
003750                          $50
003750                                                              IF ERRORFLAG EQ #SET THEN
003750  026727  000052  177777          CMP     ERRORFLAG,#SET
003756  001011                          BNE     $52
003760                                                                  IF PASS GT #1 THEN
003760  026727  000040  000001          CMP     PASS,#1
003766  003404                          BLE     $53

                                                                            , CALL ERROR IF 2ND TRY
003770                                                                      ERRHRD 67,,DIDNOT
003770  104067                          ERROR   67
003772                                                                      LET EXITFLAG  = #SET
003772  012767  177777  000030          MOV     #SET,EXITFLAG
```

```
 1794  004000                                                             ENDIF
 1795  004000                           $53
 1796  004000                                                        ELSE             ; NO ERROR
 1797  004000  000403                           BR      $54
 1798  004002                           $52
 1799  004002                                                             LET EXITFLAG  = #SET
 1800  004002  012767  177777  000020           MOV     #SET,EXITFLAG
 1801  004010                                                        ENDIF
 1802  004010                           $54
 1803  004010                                                   EXIF         EXITFLAG EQ #SET
 1804  004010  026727  000014  177777           CMP     EXITFLAG,#SET
 1805  004016  001401                           BEQ     $45
 1806  004020                                                   ENDLOOP
 1807  004020  000674                           BR      $44
 1808  004022                           $45
 1809  004022                                                        EXIT , SKIP AROUND FLAG WORDS
 1810  004022  000403                           BR      TST12          ;;;EXIT THIS TEST
 1811  004024  000000                   PASS        0
 1812  004026  000000                   ERRORFLAG   0
 1813  004030  000000                   EX TFLAG    0
 1814  004032                                                   ENDTST
```

```
1815                                    ;;**********************************************************************
1816                                    ;;**********************************************************************
1817                                    ;*TEST 12        TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH MAINT SET)
1818                                    ;*              RESULTS IN RCVRDONE SETTING WITHIN A REASONABLE AMOUNT OF TIME
1819                                    ;*              AND THAT RESET CLEARS THE BIT
1820                                    ;;**********************************************************************
1821  004032  000004                   TST12   SCOPE
1822  004034  012767  000001  175116           MOV     #1,$TIMES       ;;DO 1 ITERATION
1823  004042  012767  000012  175130           MOV     #12,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1824
1825  004050                                                           IF #MAINTJUMP NOTSETIN $USWR ORB CONSOLE EQ #TRU
1826  004050  032767  040000  175142           BIT     #MAINTJUMP,$USWR
1827  004056  001404                           BEQ     $55
1828  004060  126727  005035  000001           CMPB    CONSOLE,#TRUE
1829  004066  001004                           BNE     $56
1830  004070                           $55
1831  004070                                                              EXIT TEST
1832  004070  012767  000001  175062           MOV     #1,$TIMES
1833  004076  000442                           BR      TST13           ;;;EXIT THIS TEST
1834  004100                                                           ENDIF
1835  004100                           $56
1836
1837                                                                        ; SET THE MAINTENANCE BIT
1838  004100                                                           LET @TCSR  = @TCSR SET BY #MAINT
1839  004100  052777  000004  175156           BIS     #MAINT,@TCSR
1840
1841  004106                                                           BGNSUB
1842  004106  012767  004114  174774           MOV     #64$,$LPERR
1843                                              ; SEND A CHARACTER AND LET IT WRAP AROUND
1844
1845  004114                                                           LET @TBUF  B= #0
1846  004114  105077  175150                   CLRB    @TBUF
1847
1848                                                                   ; WAIT A MAXIMUM OF 50 MSEC
1849                                                                   ; FOR RCVR DONE TO SET IN
1850                                                                   ; RCSR
1851  004120                                                           CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
1852  004120  010546                           MOV     R5,-(SP)
1853  004122  012745  177777                   MOV     #SET,-(R5)
1854  004126  016745  175126                   MOV     RCSR,-(R5)
1855  004132  012745  000200                   MOV     #RCVRDONE,-(R5)
1856  004136  012745  000500                   MOV     #500,-(R5)
1857  004142  004767  004146                   JSR     PC,TIMER
1858  004146  012605                           MOV     (SP)+,R5
1859
1860                                                                      ;DIDN'T SET  N TIME
1861  004150                                                           IF ERROR THEN
1862  004150  103004                           BCC     $57
1863
1864                                                                      ; RCVRDONE DID NOT SET IN RCSR
1865  004152                                                              ; CAN NOT LEAVE WITH MAINT SET
1866  004152  042777  000004  175104           BIC     #MAINT,@TCSR    LET    @TCSR  = @TCSR CLR BY #MAINT
1867  004160
1868  004160  104071                           ERROR   71                 ERRHRD 71,,DIDNOT
1869  004162                                                           ENDIF
1870  004162                           $57
```

```
1871
1872  004162                                                    ENDSUB
1873
1874  004162                                                    BGNSUB
1875  004162  012767  004170  174720          MOV     #64$,$LPERR
1876                                              ; NOW THAT IT IS SET SEE IF IT CAN BE RESET
1877                                              ; THIS ALSO W LL CLEAR THE MAINT  BIT
1878  004170                                                        BRESET
1879  004170  000005                  RESET
1880
1881  004172                                                        IF #RCVRDONE SETIN @RCSR THEN
1882  004172  032777  000200  175060          BIT     #RCVRDONE,@RCSR
1883  004200  001401                  BEQ     $60
1884
1885  004202                                                            ; RCVRDONE DID NOT RESET IN RCSR
1886  004202  104072                  ERROR   ;2                        ERRHRD 72,,DIDNOT
1887  004204                                                        ENDIF
1888  004204                          $60
1889  004204                                                    ENDSUB
1890  004204                                                  ENDTST
```

```
1891                                    ;;*****************************************************************
1892                                    ;;*****************************************************************
1893                                    ;*TEST 13        TEST THAT RCVRDONE IS CLEARED BY READING RBUF
1894                                    ;;*****************************************************************
1895  004204  000004                   TST13   SCOPE
1896  004206  012767  000010  174744           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1897  004214  012767  000013  174756           MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1898
1899  004222                                                           IF #MAINTJUMP NOTSETIN $USWR ORB CONSOLE EQ #TRU
1900  004222  032767  040000  174770           B T     #MAINTJUMP,$USWR
1901  004230  001404                          BEQ     $61
1902  004232  126727  004663  000001          CMPB    CONSOLE,#TRUE
1903  004240  001004                          BNE     $62
1904  004242                           $61
1905  004242                                                              EXIT TEST
1906  004242  012767  000001  174710           MOV     #1,$TIMES
1907  004250  000440                          BR      TST14           ;;;EXIT THIS TEST
1908  004252                                                           ENDIF
1909  004252                           $62
1910
1911                                                                        ; SET MAINT  BIT
1912  004252                                                           LET @TCSR  = @TCSR SET BY #MAINT
1913  004252  052777  000004  175004           BIS     #MAINT,@TCSR
1914  004260                                                           BGNSUB
1915  004260  012767  004266  174622           MOV     #64$,$LPERR
1916                                              ; OUTPUT A CHARACTER WITH MAINTENANCE
1917                                              ; SET, AND WAIT FOR XMITRDY TO SET
1918
1919                                                                        ; OUTPUT A CHARACTER
1920  004266                                                           LET @TBUF  B= #0
1921  004266  105077  174776                  CLRB    @TBUF
1922                                                                   ; WAIT MAXIMUM OF 500 MSEC
1923                                                                   ; FOR RCVRDONE TO SET IN
1924                                                                   ; RCSR
1925  004272                                                           CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
1926  004272  010546                          MOV     R5,-(SP)
1927  004274  012745  177777                  MOV     #SET,-(R5)
1928  004300  016745  174754                  MOV     RCSR,-(R5)
1929  004304  012745  000200                  MOV     #RCVRDONE,-(R5)
1930  004310  012745  000500                  MOV     #500,-(R5)
1931  004314  004767  003774                  JSR     PC,TIMER
1932  004320  012605                          MOV     (SP)+,R5
1933  004322                                                           LET    @TCSR  = @TCSR CLR BY #MAINT
1934  004322  042777  000004  174734           BIC     #MAINT,@TCSR
1935                                                                      ; DID IT BECAME READY?
1936  004330                                                           IF ERROR  THEN
1937  004330  103001                          BCC     $63
1938                                                                      ;RCVRDONE DID NOT SET IN RCSR
1939  004332                                                              ERRHRD 73,, DIDNOT
1940  004332  104073                          ERROR   73
1941  004334                                                           ENDIF
1942  004334                           $63
1943  004334                                                        ENDSUB
1944
1945                                              ; NOW THAT IT  S SET LETS SEE  F READING THE
1946                                              ; BUFFER CLEARS RCVRDONE
```

```
1947
1948                                                                    ;READ BUFFER
1949  004334                                                          LET R0  B= @RBUF
1950  004334  117700  174722          MOVB    @RBUF,R0
1951
1952  004340                                                          IF #RCVRDONE SETIN @RCSR THEN
1953  004340  032777  000200  174712  BIT     #RCVRDONE,@RCSR
1954  004346  001401                  BEQ     $64
1955                                                                    ;RCVRDONE DID NOT CLEAR  N RCSR
1956  004350                                                            ERRHRD 74,,DIDNOT
1957  004350  104074                  ERROR   74
1958  004352                                                          ENDIF
1959  004352                  $64
1960  004352                                                  ENDTST
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 14        TEST THAT RCVRACT - RCSR 11 - SETS
                                ;*              WHEN A START BIT IS RECEIVED AND
                                ;*              CLEARS WHEN RCVRDONE - RCSR 7 - SETS
                                ;;*****************************************************************
004352  000004                  TST14   SCOPE
004354  012767  000010  174576          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004362  012767  000014  174610          MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
004370                                                          IFB CONSOLE EQ #TRUE OR #MAINTJUMP NOTSET N $USW
004370  126727  004525  000001          CMPB    CONSOLE,#TRUE
004376  001404                          BEQ     $65
004400  032767  040000  174612          BIT     #MAINTJUMP,$USWR
004406  001004                          BNE     $66
004410                          $65
004410                                                              EXIT TEST
004410  012767  000001  174542          MOV     #1,$TIMES
004416  000526                          BR      TST15           ;;;EXIT THIS TEST
004420                                                          ENDIF
004420                          $66
004420                                                          IF #APTENV SETIN $ENV THEN
004420  032767  000001  174566          B T     #APTENV,$ENV
004426  001404                          BEQ     $67
004430                                                              EXIT TEST
004430  012767  000001  174522          MOV     #1,$TIMES
004436  000516                          BR      TST15           ;;;EXIT THIS TEST
004440                                                          ENDIF
004440                          $67
                                ;
                                ;
                                ;
004440                                                      LET @TCSR   = @TCSR SET BY #MAINT
004440  052777  000004  174616          BIS     #MAINT,@TCSR
004446                                                          LET R0  = #CLR
004446  012700  000000                  MOV     #CLR,R0
004452                                                          LET R1  = #0
004452  005001                          CLR     R1
                                            ;LOAD A CHARACTER INTO TBUF
                                            ;WAIT FOR RCVRACT TO SET
                                ;
                                ;
                                ;
                                                            ;SEND A CHARACTER
004454                                                  LET @TBUF  B= #0
004454  105077  174610          CLRB    @TBUF
004460                                                  REPEAT
004460                          $70
004460                                                      IF #RCVRACT SETIN @RCSR THEN
004460  032777  004000  174572          B T     #RCVRACT,@RCSR
004466  001403                          BEQ     $71
004470                                                          LET R0  = #SET
004470  012700  177777                  MOV     #SET,R0
004474                                                      ELSE
004474  000401                          BR      $72
004476                          $71
004476                                                          LET R1  = R1 + #1
004476  005201                          INC     R1
004500                                                      END F
```

```
2017  004500                          $72
2018  004500                                                          UNTIL R0 EQ #SET OR R1 H  MAX
2019  004500  020027  177777          CMP     R0,#SET
2020  004504  001403                  BEQ     $73
2021  004506  020167  000160          CMP     R1,MAX
2022  004512  101762                  BLOS    $70
2023  004514                          $73
2024  004514                                                          IF R1 HI MAX THEN
2025  004514  020167  000152          CMP     R1,MAX
2026  004520  101410                  BLOS    $74
2027                                                                        ,IT NEVER SET
2028                                                                        ,RCVRACT DID NOT SET IN RCSR
2029                                                                        , CAN NOT LEAVE WITH MAINT SET
2030  004522                                                              LET    @TCSR  = @TCSR CLR BY #MAINT
2031  004522  042777  000004  174534  B C     #MAINT,@TCSR
2032  004530                                                              ERRHRD 75,, DIDNOT
2033  004530  104075                  ERROR   75
2034  004532                                                              EXIT TEST
2035  004532  012767  000001  174420  MOV     #1,$TIMES
2036  004540  000455                  BR      TST15           ,,,EXIT THIS TEST
2037  004542                                                          ENDIF
2038  004542                          $74
2039
2040
2041                                                    ,CHECK FOR TIMING OF RCVRACT  CLEARING
2042                                                    ,VS RCVRDONE SETTING
2043
2044
2045  004542                                                          WHILE #RCVRACT SETIN @RCSR DO
2046  004542                          $75
2047  004542  032777  004000  174510  BIT     #RCVRACT,@RCSR
2048  004550  001421                  BEQ     $76
2049
2050  004552                                                              IF #RCVRDONE SETIN @RCSR THEN
2051  004552  032777  000200  174500  B T     #RCVRDONE,@RCSR
2052  004560  001414                  BEQ     $77
2053  004562                                                                  IF #RCVRACT SETIN @RCSR THEN
2054  004562  032777  004000  174470  BIT     #RCVRACT,@RCSR
2055  004570  001410                  BEQ     $100
2056                                                                      ,RCVRDONE AND RCVRACT
2057                                                                          ,BOTH SET
2058                                                                      , CAN NOT LEAVE WITH MAINT SET
2059  004572                                                          LET    @TCSR  = @TCSR CLR BY #MAINT
2060  004572  042777  000004  174464  BIC     #MAINT,@TCSR
2061  004600                                                              ERRHRD 76, DONEACT
2062  004600  104076                  ERROR   76
2063                                                                      ,NO USE CONTINUING
2064  004602                                                              EXIT TST
2065  004602  012767  000001  174350  MOV     #1,$TIMES
2066  004610  000431                  BR      TST15           ,,,EXIT THIS TEST
2067  004612                                                          ENDIF
2068  004612                          $100
2069  004612                                                      ENDIF
2070  004612                          $77
2071  004612                                                  ENDDO
2072  004612  000753                  BR      $75
```

```
2073  004614                          $76
2074
2075                                                        ;RCVRACT = 0 NOW
2076  004614                                                IF #RCVRDONE NOTSETIN @RCSR THEN
2077  004614  032777  000200  174436        BIT     #RCVRDONE,@RCSR
2078  004622  001010                        BNE     $101
2079                                                            ;RCVRDONE DID NOT SET IN RCSR
2080                                                            ; CAN NOT LEAVE WITH MAINT SE
2081  004624                                                LET     @TCSR  = @TCSR CLR BY #MAINT
2082  004624  042777  000004  174432        BIC     #MAINT,@TCSR
2083  004632                                                    ERRHRD 77,,DIDNOT
2084  004632  104077                        ERROR   77
2085  004634                                                    EXIT TEST
2086  004634  012767  000001  174316        MOV     #1,$TIMES
2087  004642  000414                        BR      TST15           ;;;EXIT THIS TEST
2088  004644                                                ENDIF
2089  004644                          $101
2090                                                  ;TEST THAT READING THE RECEIVER
2091                                                  ;BUFFER CLEARS RCVRDONE
2092
2093
2094                                                        ;READ CHAR
2095  004644                                                LET R0  = @RBUF
2096  004644  017700  174412                MOV     @RBUF,R0
2097
2098  004650                                                IF #RCVRDONE SETIN @RCSR THEN
2099  004650  032777  000200  174402        BIT     #RCVRDONE,@RCSR
2100  004656  001404                        BEQ     $102
2101                                                            ;RCVRDONE DID NOT CLEAR IN RCSR
2102                                                            ; CAN NOT LEAVE WITH MAINT SET
2103  004660                                                LET     @TCSR  = @TCSR CLR BY #MAINT
2104  004660  042777  000004  174376        BIC     #MAINT,@TCSR
2105  004666                                                    ERRHRD 100,,DIDNOT
2106  004666  104100                        ERROR   100
2107  004670                                                ENDIF
2108  004670                          $102
2109
2110  004670                                                    EXIT
2111  004670  000401                        BR      TST15           ;;;EXIT THIS TEST
2112  004672  070000                        MAX 70000
2113
2114  004674                                                ENDTST
2115
```

```
                                ;*****************************************************************
                                ;*****************************************************************
                                ;*TEST 15       TEST THE OVERRUN BIT - RBUF 14
                                ;*****************************************************************
004674  000004                  TST15   SCOPE
004676  012767  000010  174254          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004704  012767  000015  174266          MOV     #15,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

004712                                                          IF #ERRBITS NOTSETIN $USWR ORB CONSOLE EQ #TRUE
004712  032767  100000  174300          BIT     #ERRBITS,$USWR
004720  001404                        BEQ     $103
004722  126727  004173  000001        CMPB    CONSOLE,#TRUE
004730  001004                        BNE     $104
004732                          $103
004732                                                              EXIT TEST
004732  012767  000001  174220          MOV     #1,$TIMES
004740  000547                        BR      TST16           ;;;EXIT THIS TEST
004742                                                          ENDIF
004742                          $104
004742                                                          IF #MAINTJUMP NOTSET N $USWR THEN
004742  032767  040000  174250          BIT     #MAINTJUMP,$USWR
004750  001004                        BNE     $105
004752                                                              EXIT TEST
004752  012767  000001  174200          MOV     #1,$TIMES
004760  000537                        BR      TST16           ;;;EXIT THIS TEST
004762                                                          ENDIF
004762                          $105

004762                                                          LET @TCSR = @TCSR SET BY #MAINT
004762  052777  000004  174274          BIS     #MAINT,@TCSR


004770                                                          BGNSUB
004770  012767  004776  174112          MOV     #64$,$LPERR
                                        ;OUTPUT 2 CHARACTERS WITH
                                        ;AMPLE DELAYS BETWEEN FOR RECEPTION
                                        ;THIS SHOULD AN CAUSE OVERRUN ERROR

                                                              ;OUTPUT 1 CHARACTER
004776                                                      LET @TBUF  B= #0
004776  105077  174266                CLRB    @TBUF
                                                              ;GO AWAY FOR 200 M SEC
005002                                                      WAITMS 200
005002  010546                        MOV     R5,-(SP)
005004  012745  000310                MOV     #200,-(R5)
005010  004767  003556                JSR     PC,WAIT
005014  012605                        MOV     (SP)+,R5

                                                              ;OUTPUT 2ND CHARACTER
005016                                                      LET @TBUF  B= #0
005016  105077  174246                CLRB    @TBUF
                                                              ;LET OVERRUN HAPPEN
005022                                                      WAITMS 200
005022  010546                        MOV     R5,-(SP)
```

```
2172  005024  012745  000310          MOV     #200 ,-(R5)
2173  005030  004767  003536          JSR     PC,WAIT
2174  005034  012605                  MOV     (SP)+,R5
2175
2176                                                        ;READ BUFFER AND ERROR BITS
2177  005036                                              LET R4  = @RBUF
2178  005036  017704  174220          MOV     @RBUF,R4
2179
2180                                                        ;IT DIDN'T SET
2181  005042                                              IF #ORERR NOTSETIN R4 THEN
2182  005042  032704  040000          B T     #ORERR,R4
2183  005046  001010                  BNE     $106
2184
2185                                                        ;ORERR DID NOT SET IN RBUF
2186  005050                                              ; CAN NOT LEAVE WITH MAINT SET
2187  005050  042777  000004  174206  BIC     #MAINT,@TCSR  LET   @TCSR  = @TCSR CLR BY #MAINT
2188  005056
2189  005056  104101                  ERROR   101             ERRHRD 101,,DIDNOT
2190
2191                                                        ;NO USE COMPOUNDING ERRORS
2192  005060                                                EXIT TST
2193  005060  012767  000001  174072  MOV     #1,$TIMES
2194  005066  000474                  BR      TST16          ;;;EXIT THIS TEST
2195  005070                                                  ENDIF
2196  005070                  $106
2197  005070                                              ENDSUB
2198
2199                                                  ;NOW SEE IF ERROR BIT SET WITH OVERRUN  ERROR
2200  005070                                              BGNSUB
2201  005070  012767  005076  174012          MOV     #64$,$LPERR
2202  005076                                                  IF #ERROR NOTSETIN R4 THEN
2203  005076  032704  100000          BIT     #ERROR,R4
2204  005102  001010                  BNE     $107
2205
2206                                                        ;ERROR DID NOT SET IN RBUF
2207                                                        ; CAN NOT LEAVE WITH MAINT SET
2208  005104                                              LET   @TCSR  = @TCSR CLR BY #MAINT
2209  005104  042777  000004  174152  BIC     #MAINT,@TCSR
2210  005112                                                  ERRHRD 102,,DIDNOT
2211  005112  104102                  ERROR   102
2212
2213                                                        ;-WHEN ORERR SET
2214                                                        ;GET OUT NOW
2215  005114                                                EXIT TST
2216  005114  012767  000001  174036  MOV     #1,$TIMES
2217  005122  000456                  BR      TST16          ;;;EXIT THIS TEST
2218  005124                                                  ENDIF
2219  005124                  $107
2220  005124                                              ENDSUB
2221
2222  005124                                              BGNSUB
2223  005124  012767  005132  173756          MOV     #64$,$LPERR
2224                                                  ;CHECK REAL RBUF TO SEE IF ORERR IS STILL SET
2225
2226  005132                                                  IF #ORERR NOTSETIN @RBUF THEN
2227  005132  032777  040000  174122          B T     #ORERR,@RBUF
```

```
2228  005140  001010                    BNE     $110
2229
2230                                                        ;READING RBUF CLEARED ORERR
2231                                                        ; CAN NOT LEAVE WITH MAINT SET
2232  005142                                              LET    @TCSR  = @TCSR CLR BY #MAINT
2233  005142  042777  000004  174114    BIC     #MAINT,@TCSR
2234  005150                                              ERRHRD 103,ITCLRED
2235  005150  104103                    ERROR   103
2236                                                        ;SKIP REST OF TEST
2237  005152                                              EXIT TEST
2238  005152  012767  000001  174000    MOV     #1,$TIMES
2239  005160  000437                    BR      TST16           ;;;EXIT THIS TEST
2240  005162                                              ENDIF
2241  005162                    $110
2242  005162                                            ENDSUB
2243
2244  005162                                            BGNSUB
2245  005162  012767  005170  173720    MOV     #64$,$LPERR
2246                                        ;NOW SEE IF THEY CLEAR WHEN ANOTHER CHAR  IS RECEIVED
2247
2248                                                        ;SEND A CHARACTER AROUND
2249  005170                                              LET @TBUF  B= #0
2250  005170  105077  174074            CLRB    @TBUF
2251                                                        ;LET IT CIRCULATE
2252  005174                                              WAITMS 200
2253  005174  010546                    MOV     R5,-(SP)
2254  005176  012745  000310            MOV     #200,-(R5)
2255  005202  004767  003364            JSR     PC,WAIT
2256  005206  012605                    MOV     (SP)+,R5
2257
2258  005210                                              IF #ORERR SETIN @RBUF THEN
2259  005210  032777  040000  174044    BIT     #ORERR,@RBUF
2260  005216  001410                    BEQ     $111
2261                                                        ;ORERR DID NOT CLEAR IN RBUF
2262                                                        ; CAN NOT LEAVE WITH MAINT SET
2263  005220                                              LET    @TCSR  = @TCSR CLR BY #MAINT
2264  005220  042777  000004  174036    BIC     #MAINT,@TCSR
2265  005226                                              ERRHRD 104,,DIDNOT
2266  005226  104104                    ERROR   104
2267
2268                                                        ;-AFTER RECEIVING ANOTHER CHAR
2269                                                        ;SKIP AROUND REST
2270  005230                                              EXIT TST
2271  005230  012767  000001  173722    MOV     #1,$TIMES
2272  005236  000410                    BR      TST16           ;;;EXIT THIS TEST
2273  005240                                              ENDIF
2274  005240                    $111
2275
2276  005240                                              IF #ERROR SET N @RBUF THEN
2277  005240  032777  100000  174014    BIT     #ERROR,@RBUF
2278  005246  001404                    BEQ     $112
2279                                                        ;ERROR DID NOT CLEAR  N RBUF
2280                                                        ; CAN NOT LEAVE WITH MAINT SET
2281  005250                                              LET    @TCSR  = @TCSR CLR BY #MA NT
2282  005250  042777  000004  174006    B C     #MAINT,@TCSR
2283  005256                                              ERRHRD 105,,DIDNOT
```

```
 2284  005256  104105                    ERROR   105
 2285
 2286  005260                                                        ENDIF
 2287  005260                          $112
 2288  005260                                                    ENDSUB
 2289  005260                                                ENDTST
 2290
```

```
2291
2292                                    ,,**********************************************************************
2293                                    ,,**********************************************************************
2294                                    ,*TEST 16       PROGRAMMABLE BAUD RATE TEST
2295                                    ,*              TEST AT ALL SPEEDS AVAILABLE
2296                                    ,*              A COMPARISON WILL BE MADE TO SEE
2297                                    ,*              IF NEW TIME IS LESS THAN PREVIOUS
2298                                    ,**********************************************************************
2299  005260  000004                    TST16   SCOPE
2300  005262  012767  000010  173670            MOV     #10,$TIMES      ,,DO 10 ITERATIONS
2301  005270  012767  000016  173702            MOV     #16,$TESTN      ,,SET TEST NUMBER IN APT MAIL BOX
2302
2303
2304
2305  005276                                                            IF #PBR NOTSETIN $USWR OR #MAINTJUMP NOTSETIN $U
2306  005276  032767  000200  173714            B T     #PBR,$USWR
2307  005304  001404                            BEQ     $113
2308  005306  032767  040000  173704            BIT     #MAINTJUMP,$USWR
2309  005314  001004                            BNE     $114
2310  005316                          $113
2311  005316                                                                EXIT TEST
2312  005316  012767  000001  173634            MOV     #1,$TIMES
2313  005324  000553                            BR      TST17           ,,,EXIT THIS TEST
2314  005326                                                            ENDIF
2315  005326                          $114
2316
2317  005326                                                            IFB #APTENV SETIN $ENV THEN
2318  005326  132767  000001  173660            BITB    #APTENV,$ENV
2319  005334  001404                            BEQ     $115
2320  005336                                                                EXIT TST
2321  005336  012767  000001  173614            MOV     #1,$TIMES
2322  005344  000543                            BR      TST17           ,,,EXIT THIS TEST
2323  005346                                                            ENDIF
2324  005346                          $115
2325
2326  005346                                                            LET ERRCHK  = #0        , CLEAR ERROR WORD
2327  005346  005067  002614                    CLR     ERRCHK
2328  005352                                                            LET OLD  = #-1
2329  005352  012767  177777  000270            MOV     #-1,OLD
2330  005360                                                            LET OLD+2  = #-1
2331  005360  012767  177777  000264            MOV     #-1,OLD+2
2332  005366                                                            LET @TCSR  = @TCSR SET BY #MA NT
2333  005366  052777  000004  173670            B S     #MA NT,@TCSR
2334                                                                          ,EACH BAUD RATE
2335  005374                                                            INCR R3 FROM #0 TO #15  BY #1
2336  005374  005003                            CLR     R3
2337  005376  000401                            BR      $116
2338  005400                          $117
2339  005400  005203                            INC     R3
2340  005402                          $116
2341  005402  020327  000017                    CMP     R3,#15
2342  005406  003062                            BGT     $120
2343  005410                                                            LET R0  = @RBUF
2344  005410  017700  173646                    MOV     @RBUF,R0
2345                                                                          ,CHANGE BAUDE RATE
2346  005414                                                                LET @TCSRHI  B= RATES(R3)
```

```
2347  005414  116377  005624  173644          MOVB    RATES(R3),@TCSRHI
2348
2349  005422
2350  005422  005002                          CLR     BIT
2351
2352  005424
2353  005424  005077  173640                  CLR     @TBUF
2354
2355  005430
2356  005430  005067  000210                  CLR     NEW
2357  005434
2358  005434  005067  000206                  CLR     NEW+2
2359  005440
2360  005440                        $121
2361  005440  005702                          TST     B T
2362  005442  001014                          BNE     $122
2363  005444
2364  005444  032777  000200  173606          B T     #RCVRDONE @RCSR
2365  005452  001403                          BEQ     $123
2366
2367  005454
2368  005454  012702  000001                  MOV     #1 B T
2369  005460
2370  005460  000404                          BR      $124
2371  005462                        $123
2372
2373  005462
2374  005462  005267  000156                  INC     NEW
2375  005466
2376  005466  005567  000154                  ADC     NEW+2
2377  005472
2378  005472                        $124
2379
2380  005472
2381  005472  000762                          BR      $121
2382  005474                        $122
2383
2384  005474
2385  005474  026767  000146  000150          CMP     NEW+2,OLD+2
2386  005502  103001                          BHIS    $125
2387
2388  005504
2389  005504  000414                          BR      $126
2390  005506                        $125
2391
2392  005506
2393  005506  026767  000134  000136          CMP     NEW+2,OLD+2
2394  005514  001005                          BNE     $127
2395  005516  026767  000122  000124          CMP     NEW,OLD
2396  005524  103001                          BHIS    $127
2397
2398  005526
2399  005526  000403                          BR      $130
2400  005530                        $127
2401
2402
```

```
                    ,FLAG
LET BIT  = #0

                    ,OUTPUT THE CHARACTER
LET @TBUF  = #0

                    ,INITIALIZE COUNTER
LET NEW  = #0

LET NEW+2  = #0

WHILE BIT EQ #0 DO


    IF #RCVRDONE SETIN @RCSR THEN


    ,DONE -  TS READY
        LET BIT  = #1

    ELSE


    ,OTHERWISE-INCREMENT TIME
        LET NEW  = NEW + #1

        LET NEW+2  = NEW+2 + CARRY

    ENDIF

    ,SIGNALS DONE
ENDDO


 F 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
```

```
2403                                                                    , NEW >= OLD)
2404                                                                    ,BAUD RATE DIDN'T CHANGE
2405  005530                                                          LET ERRCHK  = #BIT2 , SET ERROR INDICATOR
2406  005530  012767  000004  002430          MOV     #BIT2,ERRCHK
2407  005536                                                              ENDIF
2408  005536                          $130
2409  005536                                                          ENDIF
2410  005536                          $126
2411                                                                    ,UPDATE OLD T ME
2412  005536                                                          LET OLD  = NEW
2413  005536  016767  000102  000104          MOV     NEW,OLD
2414  005544                                                          LET OLD+2  = NEW+2
2415  005544  016767  000076  000100          MOV     NEW+2,OLD+2
2416
2417  005552                                                      ENDINC  ,BAUD RATE
2418  005552  000712                          BR      $117
2419  005554                          $120
2420  005554                                                  LET R3  B= $USWR+1 AND #17      , PUT BAUD BACK
2421  005554  116703  173441                  MOVB    $USWR+1,R3
2422  005560  110346                          MOVB    R3,-(SP)
2423  005562  142716  000017                  BICB    #17,(SP)
2424  005566  142603                          BICB    (SP)+,R3
2425  005570                                                  LET R3  = R3 CLR BY #177400
2426  005570  042703  177400                  BIC     #177400,R3
2427  005574                                                  LET @TCSRHI  B= RATES(R3)       , LIKE HE WANTED IT
2428  005574  116377  005624  173464          MOVB    RATES(R3),@TCSRHI
2429
2430                                                                  , CAN NOT LEAVE WITH MAINT SET
2431  005602                                                        LET    @TCSR  = @TCSR CLR BY #MA NT
2432  005602  042777  000004  173454          BIC     #MAINT,@TCSR
2433  005610                                                      IF #BIT2 SETIN ERRCHK THEN
2434  005610  032767  000004  002350          B T     #BIT2,ERRCHK
2435  005616  001401                          BEQ     $131
2436                                                                  , REPORT DEFERED ERROR
2437  005620                                                        ERRHRD      126
2438  005620  104126                          ERROR   126
2439  005622                                                      ENDIF
2440  005622                          $131
2441  005622                                                      EXIT    ,SKIP TABLE
2442  005622  000414                          BR      TST17           ,,,EXIT THIS TEST
2443
2444  005624                          RATES   ,A TABLE OF THE ACTUAL BYTES TO MOVE INTO THE
2445                                  ,UPPER BYTE OF XCSR FOR EACH BAUD RATE
2446                                  ,** NOTE        THE VALUE INDICATED IN THE COLUMN 'OFFSET
2447                                  ,**             INTO TABLE' CAN BE PLACED INTO BITS<11 8>
2448                                  ,**             OF LOCATION '$USWR' TO CAUSE THE CORROSPONDING
2449                                  ,**             BAUD TO BE SELECTED IN THE DLV11-F UPON
2450                                  ,**             COMPLETION OF THIS TEST
2451
2452                                                          ,       BAUD    OFFSET INTO TABLE
2453  005624     010                  R0050   BYTE    010     ,       50      0
2454  005625     030                  R0070   BYTE    030     ,       70      1
2455  005626     050                  R0110   BYTE    050     ,       110     2
2456  005627     070                  R0135   BYTE    070     ,       135     3
2457  005630     110                  R0150   BYTE    110     ,       150     4
2458  005631     130                  R0300   BYTE    130     ,       300     5
```

```
005632     150                  R0600    BYTE   150       ,      600      6
005633     170                  R0200    BYTE   170       ,      1200     7
005634     210                  R1800    BYTE   210       ,      1800     10
005635     230                  R2000    BYTE   230       ,      2000     11
005636     250                  R2400    BYTE   250       ,      2400     12
005637     270                  R3600    BYTE   270       ,      3600     13
005640     310                  R4800    BYTE   310       ,      4800     14
005641     330                  R7200    BYTE   330       ,      7200     15
005642     350                  R9600    BYTE   350       ,      9600     16
005643     370                  R10000   BYTE   370              19200    17

005644  000000  000000          NEW     0,0
005650  000000  000000          OLD     0,0
005654                                                        ENDTST
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 17        TRANSMITTER INTERRUPT LOGIC TEST
                                ;*              LOGICALLLY THIS IS 4 SEPARATE TESTS
                                ;*              A) DOES TRANSMITTER INTERRUPT LOGIC WORK
                                ;*              B) AT PRIORITY OF 0
                                ;*              C) AND ONLY ONCE
                                ;*              D) BUT NOT WITH INTERRUPT ENABLE CLEAR
                                ;;*****************************************************************
005654  000004                  TST17   SCOPE
005656  012767  000010  173274          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
005664  012767  000017  173306          MOV     #17,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX


005672                                                          IF #APTENV SETIN $ENV THEN
005672  032767  000001  173314          BIT     #APTENV,$ENV
005700  001404                        BEQ     $132
005702                                                              EXIT TEST
005702  012767  000001  173250          MOV     #1,$TIMES
005710  000532                        BR      TST20           ;;;EXIT THIS TEST
005712                                                          ENDIF
005712                          $132


                                                                    ;CLEAR 'INTERRUPT OCCURED' FLAG
005712                                                          LET INTFLAG  = #0
005712  005067  002742                CLR     INTFLAG


                                                                ;GET VECTOR ADDRESS
005716                                                          LET R3  = DLVEC
005716  016703  173334                MOV     DLVEC,R3

                                                                ;FOR THE TRANSM TTER
005722                                                          LET R3  = R3 + #4
005722  062703  000004                ADD     #4,R3

                                                                ;SET VECTOR TO POINT TO TRANS SRV AT PRI
005726                                                          SETVEC R3, #INTSRV, #PR7
005726  010146                      MOV     R1,-(SP)
005730  010301                      MOV     R3,R1
005732  012721  010652                MOV     #INTSRV,(R1)+
005736  012711  000340                MOV     #PR7,(R1)
005742  012601                      MOV     (SP)+,R1


005744                                                          BGNSUB
005744  012767  005752  173136          MOV     #64$,$LPERR

                                                                ;; MAKE SURE THAT TRANSMITTER READY IS SET
005752                                                          CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
005752  010546                      MOV     R5,-(SP)
005754  012745  177777                MOV     #SET,-(R5)
005760  016745  173300                MOV     TCSR,-(R5)
005764  012745  000200                MOV     #XMITRDY,-(R5)
005770  012745  000500                MOV     #500,-(R5)
005774  004767  002314                JSR     PC,TIMER
006000  012605                      MOV     (SP)+,R5


                                                                    ;CLEAR INTERRUPT ENABLE
006002                                                          LET @TCSR  = @TCSR CLR BY #XMITIE
```

```
006002  042777  000100  173254          BIC     #XMITIE,@TCSR


                                                                                  ,SET IT TO 0
006010  012746  000000                  MOV     #PR0,-(SP)       ,,PUT NEW PS ON STACK
006014  012746  006022                  MOV     #65$,-(SP)       ,,PUT NEW PC ON  STACK
006020  000002                          RTI                      ,,POP NEW PC AND PS
006022                          65$


                                                                                  ,NOW SET I E  BIT
006022                                                                        LET @TCSR  = @TCSR SET BY #XMITIE
006022  052777  000100  173234          BIS     #XMITIE,@TCSR

                                                                                  ,LET INTERRUPT HAVE TIME TO OCCUR
006030                                                                        WAITMS 20C
006030  010546                          MOV     R5,-(SP)
006032  012745  000310                  MOV     #200 ,-(R5)
006036  004767  002530                  JSR     PC,WAIT
006042  012605                          MOV     (SP)+,R5

                                                                                  ,DID EXACTLY 1 INTERRUPT OCCUR
006044                                                                        IF INTFLAG NE #1 THEN
006044  026727  002610  000001          CMP     INTFLAG,#1
006052  001406                          BEQ     $133
                                                                                  ,NO - WAS IT 0 OR MORE THAN ONCE
006054                                                                            IF INTFLAG EQ #0 THEN
006054  005767  002600                  TST      NTFLAG
006060  001002                          BNE     $134
                                                                                  ,TRANSMITTER DID NOT INTERRUPT IN TIME
006062                                                                                ERRHRD 106,,DIDNOT
006062  104106                          ERROR   106
006064                                                                            ELSE
006064  000401                          BR      $135
006066                          $134
                                                                                  ,TWICE
                                                                                  ,TRANSMITTER INTERRUPTED TWICE
006066                                                                                ERRHRD 107,,TWICE
006066  104107                          ERROR   107
006070                                                                            ENDIF
006070                          $135
006070                                                                        END F
006070                          $133
006070                                                                    ENDSUB
                                                                  ,INTERRUPT WITHOUT INTERRUPT ENABLE SET
006070                                                                    BGNSUB
006070  012767  006076  173012          MOV     #64$,$LPERR
                                                                                  ,CLEAR 'INTERRUPT OCCURED' FLAG
006076                                                                        LET INTFLAG  = #0
006076  005067  002556                  CLR      NTFLAG
                                                                                  ,CLEAR INTERRUPT ENABLE
006102                                                                        LET @TCSR  = @TCSR CLR BY #XMIT E
006102  042777  000100  173154          BIC     #XMITIE,@TCSR
                                                                                  ,NO  NTERRUPTS SHOULD OCCUR
006110  012746  000000                  MOV     #PR0,-(SP)       ,,PUT NEW PS ON STACK
006114  012746  006122                  MOV     #65$,-(SP)       ,,PUT NEW PC ON  STACK
006120  000002                          RT                       ,,POP NEW PC AND PS
006122                          65$
```

```
2588                                                                    ;DARE IT TO HAPPEN
2589  006122                                                          WAITMS 2
2590  006122  010546                    MOV     R5,-(SP)
2591  006124  012745  000002            MOV     #2,-(R5)
2592  006130  004767  002436            JSR     PC,WAIT
2593  006134  012605                    MOV     (SP)+,R5
2594  006136                                                          IF INTFLAG NE #0 THEN
2595  006136  005767  002516            TST     INTFLAG
2596  006142  001401                    BEQ     $136
2597                                                                    ;INTERRUPT OCCURED WITH I E CLEARED
2598  006144                                                            ERRHRD 110,NOTENAB
2599  006144  104110                    ERROR   110
2600  006146                                                          ENDIF
2601  006146                          $136
2602  006146                                                          BRESET
2603  006146  000005                    RESET
2604  006150                                                        ENDSUB
2605                                                                    ;RESTORE VECTOR AREA
2606  006150                                                        CLRVEC R3
2607  006150  010146                    MOV     R1,-(SP)        ;;PUSH R1 ON STACK
2608  006152  010246                    MOV     R2,-(SP)        ;;PUSH R2 ON STACK
2609  006154  012701  000003            MOV     #R3,R1
2610  006160  010102                    MOV     R1,R2
2611  006162  062702  000002            ADD     #2,R2
2612  006166  010221                    MOV     R2,(R1)+
2613  006170  005011                    CLR     (R1)
2614  006172  012602                    MOV     (SP)+,R2        ;;POP STACK INTO R2
2615  006174  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
2616
2617  006176                                                        ENDTST
2618
2619
2620
2621
2622
2623
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 20        RECEIVER INTERRUPT LOGIC TEST
                                ;*              THIS TEST COVERS ALL OF THE RECEIVER
                                ;*              SIDE OF THE INTERRUPT LOGIC IN
                                ;*              CHARACTER MODE
                                ;;*****************************************************************
006176  000004                  TST20   SCOPE
006200  012767  000010  172752          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
006206  012767  000020  172764          MOV     #20,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
006214                                                          IF #MAINTJUMP NOTSETIN $USWR ORB CONSOLE EQ #TRU
006214  032767  040000  172776          BIT     #MAINTJUMP,$USWR
006222  001404                          BEQ     $137
006224  126727  002671  000001          CMPB    CONSOLE,#TRUE
006232  001002                          BNE     $140
006234                          $137
006234  000167  000242                  JMP     TST21                   ;       EXIT TEST
006240                                                          ENDIF
006240                          $140
                                                                        ;CLEAR INTERRUPT OCCURED FLAG
                                                                        ;SET UP RECEIVER INTER VECTOR
006240                                                          SETVEC DLVEC,#INTSRV,#PR7
006240  010146                          MOV     R1,-(SP)
006242  016701  173010                  MOV     DLVEC,R1
006246  012721  010652                  MOV     #INTSRV,(R1)+
006252  012711  000340                  MOV     #PR7,(R1)
006256  012601                          MOV     (SP)+,R1
                                        ;PRIORITY 0 AND MULTIPLE INTERRUPT TEST -RCVRIE
006260                                                          BGNSUB
006260  012767  006266  172622          MOV     #64$,$LPERR
006266                                                          LET INTFLAG  = #0
006266  005067  002366                  CLR     INTFLAG
                                                                        ;SET MAINT  BIT
006272                                                          LET @TCSR  = @TCSR SET BY #MAINT
006272  052777  000004  172764          B S     #MAINT,@TCSR
                                                                        ;CLEAR INTERRUPTS
006300                                                              LET @RCSR  = @RCSR CLR BY #RCVRIE
006300  042777  000100  172752          B C     #RCVR E,@RCSR
                                                                        ;CHANGE PRIORITY
                                                                        ;  TO 0
006306  012746  000000                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
006312  012746  006320                  MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
006316  000002                          RTI                     ;;POP NEW PC AND PS
006320                          65$
                                                                        ;SEND A CHARACTER
006320                                                              LET @TBUF  B= #0
006320  105077  172744                  CLRB    @TBUF
                                                                        ;WAIT A MAXIMUM
                                                                        ;OF 500 MSEC FOR
                                                                        ;RCVR DONE TO SET IN RCSR
006324                                                              CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
006324  010546                          MOV     R5,-(SP)
006326  012745  177777                  MOV     #SET,-(R5)
006332  016745  172722                  MOV     RCSR,-(R5)
```

```
2680  006336  012745  000200              MOV     #RCVRDONE,-(R5)
2681  006342  012745  000500              MOV     #500.,-(R5)
2682  006346  004767  001742              JSR     PC,TIMER
2683  006352  012605                      MOV     (SP)+,R5
2684
2685  006354                                                      ;SET INTERRUPT ENABLE
2686  006354  052777  000100  172676      BIS     #RCVRIE,@RCSR   LET @RCSR  = @RCSR SET BY #RCVRIE
2687
2688  006362                                                      ;LET IT COME IN
2689  006362  010546                      MOV     R5,-(SP)        WAITMS 1
2690  006364  012745  000001              MOV     #1,-(R5)
2691  006370  004767  002176              JSR     PC,WAIT
2692  006374  012605                      MOV     (SP)+,R5
2693
2694  006376
2695  006376  017700  172660              MOV     @RBUF,R0        LET R0  = @RBUF , CLEAR RCVRDONE
2696
2697  006402                                                      ;DID HE DO IT RIGHT?
2698  006402  026727  002252  000001      CMP     INTFLAG,#1      IF INTFLAG NE #1 THEN
2699  006410  001411                      BEQ     $141
2700
2701                                                              ;NONE OCCURED
2702  006412                                                      ; CAN NOT LEAVE WITH MAINT SET
2703  006412  042777  000004  172644      BIC     #MAINT,@TCSR    LET    @TCSR  = @TCSR CLR BY #MAINT
2704  006420
2705  006420  005767  002234              TST     INTFLAG         IF INTFLAG EQ #0 THEN
2706  006424  001002                      BNE     $142
2707
2708  006426                                                      ;RECEIVER DID NOT INTERRUPT IN TIME
2709  006426  104111                      ERROR   111                 ERRHRD 111,,D DNOT
2710
2711  006430                                                      ;TWICE OR MORE
2712  006430  000401                      BR      $143            ELSE
2713  006432                          $142
2714
2715  006432                                                      ;RECEIVER INTERRUPTED TWICE
2716  006432  104112                      ERROR   112                 ERRHRD 112,,TW CE
2717  006434
2718  006434                          $143                        ENDIF
2719  006434
2720  006434                          $141                    ENDIF
2721  006434                                                  ENDSUB
2722
2723
2724                                                              ; CLEAR THE WORLD
2725  006434                                                  LET @RCSR  = @RCSR CLR BY #RCVRIE
2726  006434  042777  000100  172616      BIC     #RCVRIE,@RCSR
2727
2728
2729                                                              ;RESET MAINT  BIT
2730  006442                                                  LET @TCSR  = @TCSR CLR BY #MAINT
2731  006442  042777  000004  172614      BIC     #MAINT,@TCSR
2732
2733  006450                                                  LET R4  = @DLVEC
2734  006450  017704  172602              MOV     @DLVEC,R4
2735  006454                                                  CLRVEC R4
```

```
 2736  006454  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 2737  006456  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 2738  006460  012701  000004                MOV     #R4,R1
 2739  006464  010102                        MOV     R1,R2
 2740  006466  062702  000002                ADD     #2,R2
 2741  006472  010221                        MOV     R2,(R1)+
 2742  006474  005011                        CLR     (R1)
 2743  006476  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
 2744  006500  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 2745  006502                                                          ENDTST
```

```
2746                                    ;;*****************************************************************
2747                                    ;;*****************************************************************
2748                                    ;*TEST 21        TEST ACTUAL DATA TRANSFERED
2749                                    ;*               NON-INTERRUPT MAINTENANCE BIT SET
2750                                    ;;*****************************************************************
2751  006502  000004                    TST21   SCOPE
2752  006504  012767  000001  172446            MOV     #1,$TIMES       ;;DO 1 ITERATION
2753  006512  012767  000021  172460            MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2754  006520                                                                IF #MAINTJUMP NOTSETIN $USWR ORB CONSOLE EQ #TRU
2755  006520  032767  040000  172472            BIT     #MAINTJUMP,$USWR
2756  006526  001404                            BEQ     $144
2757  006530  126727  002365  000001            CMPB    CONSOLE,#TRUE
2758  006536  001004                            BNE     $145
2759  006540                            $144
2760  006540                                                                  EXIT TEST
2761  006540  012767  000001  172412            MOV     #1,$TIMES
2762  006546  000526                            BR      TST22           ;;;EXIT THIS TEST
2763  006550                                                                ENDIF
2764  006550                            $145
2765
2766  006550
2767  006550  005067  001412                    CLR     ERRCHK                  LET ERRCHK  = #0
2768
2769  006554                                                                ;SET MAINT  BIT
2770  006554  052777  000004  172502            BIS     #MAINT,@TCSR            LET @TCSR  = @TCSR SET BY #MAINT
2771
2772
2773                                                                        ;CHANGE PRIORITY
                                                                            ;  TO 0
2774  006562  012746  000000                    MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
2775  006566  012746  006574                    MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
2776  006572  000002                            RTI                     ;;POP NEW PC AND PS
2777  006574                            64$
2778
2779  006574                                                                ;GET DATA MASK
2780  006574  162705  000002                    SUB     #1*2,R5                 CALL DATLNG OUT <R1>
2781  006600  004767  001666                    JSR     PC,DATLNG
2782  006604  012501                            MOV     (R5)+,R1
2783  006606
2784  006606  017700  172450                    MOV     @RBUF,R0                LET R0  = @RBUF , START CLEAR
2785
2786
2787  006612                                                                ;ALL BINARY CHAR
2788  006612  005002                            CLR     R2                      INCR R2 FROM #0 TO #377 BY #1
2789  006614  000401                            BR      $146
2790  006616                            $147
2791  006616  005202                            INC     R2
2792  006620                            $146
2793  006620  020227  000377                    CMP     R2,#377
2794  006624  003062                            BGT     $150
2795
2796
2797
2798                                                                        ;TRANSMIT CHAR IN R2
2799  006626
2800  006626  010546                            MOV     R5,-(SP)                CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
2801  006630  012745  177777                    MOV     #SET,-(R5)
```

```
2802  006634  016745  172424          MOV     TCSR,-(R5)
2803  006640  012745  000200          MOV     #XMITRDY,-(R5)
2804  006644  012745  000500          MOV     #500,-(R5)
2805  006650  004767  001440          JSR     PC,TIMER
2806  006654  012605                  MOV     (SP)+,R5
2807  006656                                                  IF ERROR THEN
2808  006656  103003                  BCC     $151
2809  006660                                                      LET ERRCHK  = ERRCHK SET BY #BIT3
2810  006660  052767  000010  001300  BIS     #B T3,ERRCHK
2811  006666                                                  ENDIF
2812  006666                  $151
2813
2814                                                          ,TRANSMIT IT
2815  006666                                                  LET @TBUF  B= R2
2816  006666  110277  172376          MOVB    R2,@TBUF
2817
2818  006672                                                  CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
2819  006672  010546                  MOV     R5,-(SP)
2820  006674  012745  177777          MOV     #SET,-(R5)
2821  006700  016745  172354          MOV     RCSR,-(R5)
2822  006704  012745  000200          MOV     #RCVRDONE,-(R5)
2823  006710  012745  000500          MOV     #500,-(R5)
2824  006714  004767  001374          JSR     PC,TIMER
2825  006720  012605                  MOV     (SP)+,R5
2826  006722                                                  IF ERROR THEN
2827  006722  103003                  BCC     $152
2828  006724                                                      LET ERRCHK  = ERRCHK SET BY #BIT4
2829  006724  052767  000020  001234  B S     #BIT4,ERRCHK
2830  006732                                                  ENDIF
2831  006732                  $152
2832                                                          ,AND SAVE IT
2833  006732                                                  LET R3  = @RBUF
2834  006732  017703  172324          MOV     @RBUF,R3
2835
2836
2837                                                          ,COMPARE TO SEE  F WE RECE VED  T ALL
2838
2839                                                              ,CLEAN OFF NON-DATA BITS
2840                                                              ,ON BOTH TRANSMITTED AND
2841  006736                                                  LET R4  = R2 CLR BY R1
2842  006736  010204                  MOV     R2,R4
2843  006740  040104                  BIC     R1,R4
2844  006742                                                  LET R3  = R3 CLR BY R1
2845  006742  040103                  BIC     R1,R3
2846
2847                                                          ,RECEIVED DATA
2848  006744                                                  IF R4 NE P3 THEN
2849  006744  020403                  CMP     R4,R3
2850  006746  001410                  BEQ     $153
2851                                                              ,DATA COMPARE ERROR
2852                                                              , CAN NOT LEAVE WITH MAINT SET
2853  006750                                                  LET    @TCSR  = @TCSR CLR BY #MAINT
2854  006750  042777  000004  172306  BIC     #MAINT,@TCSR
2855  006756                                                      ERRHRD 116,COMP,SBWAS
2856  006756  104116                  ERROR   116
2857  006760                                                      EXIT TEST , ON ERROR
```

```
2858  006760  012767  000001  172172          MOV     #1,$TIMES
2859  006766  000416                          BR      TST22    ;       ;;;EXIT THIS TEST
2860  006770                                                                  ENDIF
2861  006770                          $153
2862  006770                                                          ENDINC  ; R2
2863  006770  000712                          BR      $147
2864  006772                          $150
2865
2866                                                                  ;RESET MAINT  BIT
2867  006772                                                          LET @TCSR  = @TCSR CLR BY #MAINT
2868  006772  042777  000004  172264          BIC     #MAINT,@TCSR
2869  007000                                                          IF #BIT3 SETIN ERRCHK THEN
2870  007000  032767  000010  001160          BIT     #BIT3,ERRCHK
2871  007006  001401                          BEQ     $154
2872  007010                                                                  ERRHRD 130
2873  007010  104130                          ERROR   130
2874  007012                                                          ENDIF
2875  007012                          $154
2876  007012                                                          IF #BIT4 SETIN ERRCHK THEN
2877  007012  032767  000020  001146          BIT     #BIT4,ERRCHK
2878  007020  001401                          BEQ     $155
2879  007022                                                                  ERRHRD 131
2880  007022  104131                          ERROR   131
2881  007024                                                          ENDIF
2882  007024                          $155
2883  007024                                                  ENDTST
2884
2885
2886
```

```
2887                                    ;;**********************************************************************
2888                                    ;;**********************************************************************
2889                                    ;*TEST 22        TEST DATA THROUGH WRAP
2890                                    ;;**********************************************************************
2891  007024  000004                    TST22   SCOPE
2892  007026  012767  000001  172124            MOV     #1,$TIMES       ;;DO 1 ITERATION
2893  007034  012767  000022  172136            MOV     #22,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2894  007042                                                    IF #WRAP NOTSETIN $USWR OR #COMSPD NOTSETIN $USW
2895  007042  032767  020000  172150            BIT     #WRAP,$USWR
2896  007050  001404                            BEQ     $156
2897  007052  032767  000100  172140            BIT     #COMSPD,$USWR
2898  007060  001004                            BNE     $157
2899  007062                            $156
2900                                                              ;CAN'T TEST WITHOUT A WRAP
2901  007062                                                    EXIT TST
2902  007062  012767  000001  172070            MOV     #1,$TIMES
2903  007070  000516                            BR      TST23           ;;;EXIT THIS TEST
2904  007072                                                    ENDIF
2905  007072                            $157
2906                                                              ;DON'T USE MAINT
2907  007072                                                    LET @TCSR  = @TCSR CLR BY #MAINT
2908  007072  042777  000004  172164            BIC     #MAINT,@TCSR
2909                                                              ; IF A SPECIAL TURN AROUND CARD IS
2910                                                              ;   CONNECTED IN PLACE OF THE WRAP
2911                                                              ;  SETTING READER RUN WILL ENABLE IT
2912                                                              ; THIS MODULE IS ONLY USED IN MANUFACTUR
2913                                                              ; AND ONLY ON THE CONSOLE DLV11-F
2914                                                              ; IF NO SPECIAL MODULE IS AVAILABLE,
2915                                                              ; AND THE WRAP BIT IS SET IN $USWR
2916                                                              ; THEN THIS TEST WILL ERROR ON THE CONSO
2917
2918  007100                                                  LET @RCSR  = @RCSR SET BY #11
2919  007100  052777  000011  172152            B S     #11,@RCSR
2920                                                              ;CHANGE PRIORITY
2921                                                              ;  TO 0
2922  007106  012746  000000                    MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
2923  007112  012746  007120                    MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
2924  007116  000002                            RT                      ;;POP NEW PC AND PS
2925  007120                            64$
2926                                                              ;GET DATA MASK
2927  007120                                                  CALL DATLNG OUT <R1>
2928  007120  162705  000002                    SUB     #1*2,R5
2929  007124  004767  001342                    JSR     PC,DATLNG
2930  007130  012501                            MOV     (R5)+,R1
2931  007132                                                  LET R0  = @RBUF , START CLEAN
2932  007132  017700  172124                    MOV     @RBUF,R0
2933                                                              ;BINARY COUNT PATTERN
2934  007136                                                  INCR R2 FROM #0 TO #377 BY #1
2935  007136  005002                            CLR     R2
2936  007140  000401                            BR      $160
2937  007142                            $161
2938  007142  005202                            INC     R2
2939  007144                            $160
2940  007144  020227  000377                    CMP     R2,#377
2941  007150  003063                            BGT     $162
2942
```

```
2943
2944                                                        ;TRANSMIT THE CHAR  IN R2
2945
2946                                                            ; MAKE SURE IT'S READY
2947  007152                                                CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
2948  007152  010546                   MOV     R5,-(SP)
2949  007154  012745  177777           MOV     #SET,-(R5)
2950  007160  016745  172100           MOV     TCSR,-(R5)
2951  007164  012745  000200           MOV     #XMITRDY,-(R5)
2952  007170  012745  000500           MOV     #500,-(R5)
2953  007174  004767  001114           JSR     PC,TIMER
2954  007200  012605                   MOV     (SP)+,R5
2955  007202                                                IF ERROR THEN
2956  007202  103005                   BCC     $163
2957  007204  104123                   ERROR   123                 ; TRANSMITTER NEVER BECAME READY
2958  007206                                                    EXIT TEST
2959  007206  012767  000001  171744   MOV     #1,$TIMES
2960  007214  000444                   BR      TST23          ;;;EXIT THIS TEST
2961  007216                                                ENDIF
2962  007216                      $163
2963
2964                                                            ;START IT ON ITS WAY
2965  007216                                                LET @TBUF  B= R2
2966  007216  110277  172046           MOVB    R2,@TBUF
2967                                                        ; NOW WAIT FOR RECIEVER DONE
2968  007222                                                CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
2969  007222  010546                   MOV     R5,-(SP)
2970  007224  012745  177777           MOV     #SET,-(R5)
2971  007230  016745  172024           MOV     RCSR,-(R5)
2972  007234  012745  000200           MOV     #RCVRDONE,-(R5)
2973  007240  012745  000500           MOV     #500,-(R5)
2974  007244  004767  001044           JSR     PC,TIMER
2975  007250  012605                   MOV     (SP)+,R5
2976  007252                                                    IF ERROR THEN
2977  007252  103005                   BCC     $164
2978  007254                                                        ERRHRD 124
2979  007254  104124                   ERROR   124
2980                                                            ; RECIEVER NEVER BECAME READY
2981  007256                                                    EXIT TEST
2982  007256  012767  000001  171674   MOV     #1,$TIMES
2983  007264  000420                   BR      TST23          ;;;EXIT THIS TEST
2984  007266                                                ENDIF
2985  007266                      $164
2986
2987                                                            ;RETRIEVE
2988  007266                                                LET R3  = @RBUF
2989  007266  017703  171770           MOV     @RBUF,R3
2990
2991                                                            ;STRIP OFF JUNK ON BOTH
2992  007272                                                LET R4  = R2 CLR BY R1
2993  007272  010204                   MOV     R2,R4
2994  007274  040104                   BIC     R1,R4
2995  007276                                                LET R3  = R3 CLR BY R1
2996  007276  040103                   BIC     R1,R3
2997
2998                                                            ;WE HAVE TROUBLE
```

```
2999  007300                                                    IF R4 NE R3 THEN
3000  007300  020403                     CMP     R4,R3
3001  007302  001405                     BEQ     $165
3002                                                              ,DATA COMPARE ERROR
3003  007304                                                      ERRHRD 117,COMP,SBWAS
3004  007304  104117                     ERROR   117
3005  007306                                                      EXIT TEST , ON ERROR
3006  007306  012767  000001  171644     MOV     #1,$TIMES
3007  007314  000404                     BR      TS^23          ,,,EXIT THIS TEST
3008  007316                                                    ENDIF
3009  007316                      $165
3010
3011  007316                                                ENDINC  , R2
3012  007316  000711                     BR      $161
3013  007320                      $162
3014                                                              , NOW THAT THE TEST IS DONE
3015                                                              ,  WE WILL TOGGLE READER RUN
3016                                                              ,  TO TURN OFF THE SPECIAL MODULE
3017  007320                                                LET @RCSR  = @RCSR SET BY #11
3018  007320  052777  000011  171732     B S     #11,@RCSR
3019
3020
3021
3022  007326                                              ENDTST
3023
3024
3025
3026
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 23        FULL DATA TRANSFER WITH INTERRUPTS
                                ;*              AND MAINTENANCE MODE
                                ;;*****************************************************************
007326  000004                  TST23   SCOPE
007330  012767  000001  171622          MOV     #1,$TIMES       ;;DO 1 ITERATION
007336  012767  000023  171634          MOV     #23,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

007344                                                          IF #MAINTJUMP NOTSETIN $USWR ORB CONSOLE EQ #TRU
007344  032767  040000  171646          B T     #MAINTJUMP,$USWR
007352  001404                        BEQ     $166
007354  126727  001541  000001        CMPB    CONSOLE,#TRUE
007362  001004                        BNE     $167
007364                          $166
007364                                                              EXIT TEST
007364  012767  000001  171566          MOV     #1,$TIMES
007372  000553                        BR      TST24           ;;;EXIT THIS TEST
007374                                                          ENDIF
007374                          $167

                                                                        ;GET DATA MASK
007374                                                          CALL DATLNG OUT <R3>
007374  162705  000002                SUB     #1*2,R5
007400  004767  001066                JSR     PC,DATLNG
007404  012503                        MOV     (R5)+,R3


                                        ; THIS TEST WILL RUN BOTH TRANSMITTER AND
                                        ; RECIEVER AT FULL SPEED TESTING
                                        ; THE ABILITY OF THE MODULE
                                        ; TO HANDLE INTERRUPTS FROM BOTH SIDES
                                        ; AT ONCE  ALSO, THE DOUBLE BUFFERING LOG C
                                        ; OF THE UART WILL BE FULLY TESTED
                                        ;  THIS TEST WILL TRANSFER A MAXIMUM OF 400(8)
                                        ; CHARACTERS THROUGH THE MODULE, BUT IF AN ERROR
                                        ; IS DETECTED BY THE TEST A PREMATURE SHUTDOWN OCCURS


                                                                        ;CHANGE PRIORITY
                                                                        ;  TO 0
007406  012746  000000                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
007412  012746  007420                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
007416  000002                          RTI                     ;;POP NEW PC AND PS
007420                          64$
                                                                        ;GET VECTOR ADDRESS
007420                                                          LET R1  = DLVEC
007420  016701  171632                  MOV     DLVEC,R1
                                                                        ;RCVR VECTOR
007424                                                          LET (R1)+  = #REC
007424  012721  007626                  MOV     #REC,(R1)+
007430                                                          LET (R1)+  = #PR7
007430  012721  000340                  MOV     #PR7,(R1)+
                                                                        ;POINT TO TRANSMITTER VECTOR
                                                                        ;AND SET IT UP ALSO
007434                                                          LET (R1)+  = #TRAN
007434  012721  007564                  MOV     #TRAN,(R1)+
```

```
3083  007440                                                    LET (R1)   = #PR7
3084  007440  012711  000340          MOV     #PR7,(R1)
3085
3086                                                            ,CLEAR ERROR COUNTER
3087  007444                                                    LET ERRCNT  = #0
3088  007444  005067  000106          CLR     ERRCNT
3089                                                            ,INITIAL ZE COUNTERS
3090  007450                                                    LET R1  = #-1
3091  007450  012701  177777          MOV     #-1,R1
3092                                                            ,RECEIVER STORAGE
3093  007454                                                    LET R2  = #0
3094  007454  005002                  CLR     R2
3095                                                            ,# OF RECEIVED CHAR  COUNT
3096  007456                                                    LET R4  = #-1
3097  007456  012704  177777          MOV     #-1,R4
3098
3099  007462                                                    BRESET  ,SET UP ALL REG STERS
3100  007462  000005                  RESET
3101                                                            ,SET UP MAINTENANCE
3102  007464                                                    LET @TCSR  = @TCSR SET BY #MAINT
3103  007464  052777  000004  171572  BIS     #MAINT,@TCSR
3104
3105                                                            ,SET I E  IN TRANSMITTER
3106  007472                                                    LET @TCSR  = @TCSR SET BY #XMIT E
3107  007472  052777  000100  171564  BIS     #XMITIE,@TCSR
3108                                                            ,AND RECEIVER
3109  007500                                                    LET @RCSR  = @RCSR SET BY #RCVRIE
3110  007500  052777  000100  171552  BIS     #RCVRIE,@RCSR
3111
3112
3113                                                            ,NOW WE WAIT UNTIL R4 COUNT (RECEIVED) IS EQUAL
3114  007506                                                    REPEAT
3115  007506                  $170
3116  007506                                                    UNTIL R4 EQ NUMBER OR ERRCNT GT #0
3117  007506  020467  000046          CMP     R4,NUMBER
3118  007512  001403                  BEQ     $171
3119  007514  005767  000036          TST     ERRCNT
3120  007520  003772                  BLE     $170
3121  007522                  $171
3122
3123  007522                                                    LET @TCSR  = @TCSR CLR BY #MAINT
3124  007522  042777  000004  171534  BIC     #MAINT,@TCSR
3125                                                            ,CHECK FOR DATA COMPARE ERRORS
3126  007530                                                    IF ERRCNT NE #0 THEN
3127  007530  005767  000022          TST     ERRCNT
3128  007534  001401                  BEQ     $172
3129                                                                ,DATA COMPARE ERROR
3130  007536                                                        ERRHRD 120,COMP,FIRST
3131  007536  104120                  ERROR   120
3132  007540                                                    ENDIF
3133  007540                  $172
3134
3135  007540                                                    LET @TCSR  = @TCSR CLR BY #XMITIE
3136  007540  042777  000100  171516  B C     #XMITIE,@TCSR
3137  007546                                                    LET @RCSR  = @RCSR CLR BY #RCVRIE
3138  007546  042777  000100  171504  B C     #RCVR E,@RCSR
```

```
007554                                                                      EXIT     ,SKIP OVER SUPPORT POUTINES & STORAGE
007554  000462                                  BR      TST24          ,,EXIT THIS TEST

007556  000000                        ERRCNT   0
007560  001000                        NUMBER   1000
007562     000                        SB        BYTE 0
007563     000                        WAS       BYTE 0


                                      ,,*************************************************************
                                                                  ,TRANSMIT INTERRUPT HANDLER
007564                                                                      BGNSRV  TRAN
007564                                TRAN
                                      ,,*************************************************************


                                                                               ,INCREMENT CHAR COUNT
007564                                                                      LET R1  = R1 + #1
007564  005201                                  INC     R1

                                                                               ,SET UP FOR TRANSFER
007566                                                                      LET HOLD  = R1 CLR BY R3
007566  010167  000030                          MOV     R1,HOLD
007572  040367  000024                          BIC     R3,HOLD

                                                                               ,AND SEND
007576                                                                      LET @TBUF  = HOLD
007576  016777  000020  171464                  MOV     HOLD,@TBUF

                                                                               ,ALL DONE
007604                                                                      IF R1 EQ NUMBER THEN
007604  020167  177750                          CMP     R1,NUMBER
007610  001003                                  BNE     $173

                                                                               ,STOP   INTERRUPT PROCESS NG
007612                                                                         LET @TCSR  = @TCSR CLR BY #XM T E
007612  042777  000100  171444                  B C     #XMITIE,@TCSR
007620                                                                      ENDIF
007620                                $173

007620  000401                                  BR      ZZZ                ,  EXIT SRV

007622  000000                            HOLD 0

007624                                ZZZ                                   ENDSRV
007624  000002                                  RTI




                                      ,,*************************************************************
                                                                  ,RECEIVER INTERRUPT HANDLER
007626                                                                      BGNSRV  REC
007626                                REC
                                      ,,*************************************************************


                                                                               ,COUNT THIS CHAR
007626                                                                      LET R4  = R4 + #1
007626  005204                                  INC     R4

                                                                               ,GET CHAR IN + MASK IT
007630                                                                      LET R2  = @RBUF CLR BY R3
007630  017702  171426                          MOV     @RBUF,R2
```

```
3195  007634  040302                      BIC     R3,R2
3196                                                        ;RHLD WILL CONTAIN EXPECTED INPUT
3197  007636                                              LET RHLD  = R4 CLR BY R3
3198  007636  010467  000054              MOV     R4,RHLD
3199  007642  040367  000050              BIC     R3,RHLD
3200
3201                                                        ;DO THEY COMPARE
3202  007646                                              IF R2 NE RHLD THEN
3203  007646  020267  000044              CMP     R2,RHLD
3204  007652  001412                      BEQ     $174
3205
3206  007654                                                  ;FIRST ERROR
3207  007654  005767  177676              TST     ERRCNT     IF ERRCNT EQ #0 THEN
3208  007660  001005                      BNE     $175
3209
3210  007662                                                  ;SAVE RECORD OF FIRST MISS
3211  007662  116767  000030  177672      MOVB    RHLD,SB        LET SB  B= RHLD
3212  007670
3213  007670  110267  177667              MOVB    R2,WAS         LET WAS  B= R2
3214  007674                                              ENDIF
3215  007674                          $175
3216                                                      ;COUNT IT
3217  007674                                              LET ERRCNT  = ERRCNT + #1
3218  007674  005267  177656              INC     ERRCNT
3219  007700                                              ENDIF
3220  007700                          $174
3221
3222                                                      ;ALL DONE?
3223  007700                                              IF R4 EQ NUMBER THEN
3224  007700  020467  177654              CMP     R4,NUMBER
3225  007704  001003                      BNE     $176
3226                                                          ;STOP RECEIVER INTERRUPTS
3227  007706                                                  LET @RCSR  = @RCSR CLR BY #RCVRIE
3228  007706  042777  000100  171344      BIC     #RCVRIE,@RCSR
3229                                                                  ; INDICATE ALL DONE TO T MER
3230                                                          ;MAIN REPEAT LOOP IS CHECKING
3231                                                          ;FOR 'R4 = NUMBER' ALSO
3232  007714                                              ENDIF
3233  007714                          $176
3234
3235  007714  000401                      BR      ZZZZ                ;  EXIT SRV
3236
3237  007716  000000                                              RHLD 0
3238  007720                          ZZZZ
3239  007720                                          ENDSRV
3240  007720  000002                      RTI
3241
3242  007722                                      ENDTST
3243
3244
3245
```

```
3246                                    ;;**********************************************************************
3247                                    ;;**********************************************************************
3248                                    ;*TEST 24        TEST BREAK GENERATION LOGIC
3249                                    ;*              TRANSMIT KNOWN CHAR WITH BREAK SET
3250                                    ;*              AND COMPARE RECEIVED WITH 0
3251                                    ;*              FRAMING ERROR WILL ALSO BE CHECKED
3252                                    ;*              IF ERROR BITS ARE ENABLED
3253                                    ;;**********************************************************************
3254  007722  000004                   TST24   SCOPE
3255  007724  012767  000010  171226            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
3256  007732  012767  000024  171240            MOV     #24,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3257  007740                                                            IF #MAINTJUMP NOTSETIN $USWR OR #BRK NOTSETIN $U
3258  007740  032767  040000  171252            BIT     #MAINTJUMP,$USWR
3259  007746  001404                           BEQ     $177
3260  007750  032767  010000  171242            B T     #BRK,$USWR
3261  007756  001004                           BNE     $200
3262  007760                            $177
3263  007760                                                                EXIT TEST
3264  007760  012767  000001  171172            MOV     #1,$TIMES
3265  007766  000500                           BR      TST25           ;;;EXIT THIS TEST
3266  007770                                                            ENDIF
3267  007770                            $200
3268  007770                                                            IFB CONSOLE EQ #TRUE THEN
3269  007770  126727  001125  000001            CMPB    CONSOLE,#TRUE
3270  007776  001004                           BNE     $201
3271  010000                                                                EXIT TEST
3272  010000  012767  000001  171152            MOV     #1,$TIMES
3273  010006  000470                           BR      TST25           ;;;EXIT THIS TEST
3274  010010                                                            ENDIF
3275  010010                            $201
3276
3277  010010                                                            LET ERRCHK  = #0        ; CLEAR ERROR WORD
3278  010010  005067  000152                   CLR     ERRCHK
3279                                                                    ;SET MAINTENANCE BIT
3280  010014                                                            LET @TCSR  = @TCSR SET BY #MAINT
3281  010014  052777  000004  171242            B S     #MAINT,@TCSR
3282                                                                    ;SET BREAK BIT
3283  010022                                                            LET @TCSR  = @TCSR SET BY #BREAK
3284  010022  052777  000001  171234            B S     #BREAK,@TCSR
3285                                                                    ;NON-ZERO CHAR  '*'
3286  010030                                                            LET @TBUF  = #252
3287  010030  012777  000252  171232            MOV     #252,@TBUF
3288                                                                    ;WAIT FOR DONE
3289  010036                                                            CALL TIMER IN <#500,*RCVRDONE,RCSR,#SET>
3290  010036  010546                           MOV     R5,-(SP)
3291  010040  012745  177777                   MOV     #SET,-(R5)
3292  010044  016745  171210                   MOV     RCSR,-(R5)
3293  010050  012745  000200                   MOV     #RCVRDONE,-(R5)
3294  010054  012745  000500                   MOV     #500,-(R5)
3295  010060  004767  000230                   JSR     PC,TIMER
3296  010064  012605                           MOV     (SP)+,R5
3297  010066                                                            IF ERROR THEN
3298  010066  103001                           BCC     $202
3299                                                                        ;RECIEVER DONE DID NOT SET
3300  010070                                                                ERRHRD 115
3301  010070  104115                           ERROR   115
```

```
3302  010072                                                          ENDIF
3303  010072                        $202
3304
3305  010072                                                          IFB @RBUF NE #0 THEN
3306  010072  105777  171164               TSTB    @RBUF
3307  010076  001404                       BEQ     $203
3308                                                                        , BREAK DID NOT EQUAL 0
3309  010100                                                              LET ERRCHK  = ERRCHK SET BY #B TO
3310  010100  052767  000001  000060       B S     #B TO,ERRCHK
3311  010106                                                          ELSE
3312  010106  000413                       BR      $204
3313  010110                        $203
3314  010110                                                          IF #ERRBITS SETIN $USWR THEN
3315  010110  032767  100000  171102       BIT     #ERRBITS,$USWR
3316  010116  001407                       BEQ     $205
3317  010120                                                                  IF #FRERR NOTSETIN @RBUF THEN
3318  010120  032777  020000  171134       BIT     #FRERR,@RBUF
3319  010126  001003                       BNE     $206
3320  010130                                                                      LET ERRCHK  = ERRCHK SET BY #BIT1
3321  010130  052767  000002  000030       BIS     #BIT1,ERRCHK
3322  010136                                                                  ENDIF
3323  010136                        $206
3324  010136                                                          ENDIF
3325  010136                        $205
3326  010136                                                          ENDIF
3327  010136                        $204
3328
3329  010136                                                          BRESET  ,CLEAN UP
3330  010136  000005                       RESET
3331
3332  010140                                                          IF #BIT0 SETIN ERRCHK THEN
3333  010140  032767  000001  000020       BIT     #BIT0,ERRCHK
3334  010146  001401                       BEQ     $207
3335  010150                                                              ERRHRD 121  ,BREAK ERROR
3336  010150  104121                       ERROR   121
3337  010152                                                          ENDIF
3338  010152                        $207
3339  010152                                                          IF #BIT1 SETIN ERRCHK THEN
3340  010152  032767  000002  000006       BIT     #BIT1,ERRCHK
3341  010160  001401                       BEQ     $210
3342  010162                                                              ERRHRD 122  , FRAMING ERROR
3343  010162  104122                       ERROR   122
3344  010164                                                          ENDIF
3345  010164                        $210
3346  010164                                                          EXIT
3347  010164  000401                       BR      TST25           ,,,EXIT THIS TEST
3348  010166  000000                ERRCHK  WORD 0
3349  010170                                                  ENDTST
3350
```

```
3351
3352                                    ;;**********************************************************************
3353                                    ;*TEST 25        NOT A TEST - SEND BACK TO LOOP
3354                                    ;;**********************************************************************
3355  010170  000004                    TST25   SCOPE
3356  010172  012767  000001  170760            MOV     #1,$TIMES       ;;DO 1 ITERATION
3357  010200  104401  010206                    TYPE    ,65$            ;;TYPE ASCIZ STRING
3358  010204  000404                    64$     BR      64$             ;;GET OVER THE ASCIZ
3359                                    ;;65$    ASCIZ   <CRLF>*CSR  *
3360  010216                            64$
3361  010216  016746  171032                    MOV     DLADD,-(SP)     ;;SAVE DLADD FOR TYPEOUT
3362  010222  104402                            TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3363  010224  104401  010232                    TYPE    ,67$            ;;TYPE ASCIZ STRING
3364  010230  000405                    66$     BR      66$             ;;GET OVER THE ASCIZ
3365                                    ;;67$    ASCIZ   *,VECTOR  *
3366  010244                            66$
3367  010244  016746  171006                    MOV     DLVEC,-(SP)     ;;SAVE DLVEC FOR TYPEOUT
3368  010250  104402                            TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3369  010252  104401  010260                    TYPE    ,69$            ;;TYPE ASCIZ STRING
3370  010256  000405                    68$     BR      68$             ;;GET OVER THE ASCIZ
3371                                    ;;69$    ASCIZ   *,ERRORS  *
3372  010272                            68$
3373  010272  016746  170614                    MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
3374  010276  104405                            TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
3375  010300  005067  170606                    CLR     $ERTTL          ; RESET FOR NEXT DEVICE/PASS
3376  010304  104401  001171                    TYPE    ,$CRLF
3377  010310  000167  171622                    JMP     LOOP            ; BACK UP TO THE BEGINNING
```

```
3378
3379                                        ,,BGNMOD        SUBS
3380                                    ,,**************************************************************
3381  010314                              ROUTINE TIMER (HOWLONG,WHICHBIT,REG,SETCLR)
3382  010314                            TIMER
3383                                    ,* ROUTINE TIMER
3384                                    ,*      THIS ROUTINE IS USED TO TEST THE STATUS OF ANY BIT
3385                                    ,*      IN ANY REGISTER
3386                                    ,*      INPUTS
3387                                    ,*            HOWLONG    THE MAXIMUM AMOUNT OF TIME TO SPEND IN
3388                                    ,*                       THIS ROUTINE
3389                                    ,*            WHICHBIT     A MASK WITH THE BIT(S) SET THAT ARE
3390                                    ,*                         TO BE CHECKED
3391                                    ,*            REG     A POINTER TO THE REGISTER TO BE CHECKED
3392                                    ,*            SETCLR  THE DESIRED RESULTS
3393                                    ,*                    EITHER #SET OR #CLEAR
3394                                    ,*      OUTPUT
3395                                    ,*            THE 'C' BIT IS SET TO INDICATE AN ERROR
3396                                    ,*            BUT IT IS TESTED BY THE IF.ERROR STATEMENT
3397                                    ,*
3398                                    ,*
3399                                    ,*            NOTE    THE USE OF (R5) IS PART OF THE LINKAGE
3400                                    ,*                    MECHANISM BETWEEN THE CALLER AND THE CALLED
3401                                    ,,**************************************************************
3402
3403          000001                                                    TRUE=   1
3404          000000                                                    FALSE=  0
3405
3406  010314
3407  010314  016567  000004  000136            MOV     REG(R5),REGSAV
3408  010322
3409  010322  016567  000000  000132            MOV     HOWLONG(R5),TIMSAV
3410  010330
3411  010330  112767  000000  000126            MOVB    #FALSE,FLAG
3412
3413
3414
3415
3416  010336
3417  010336                          $213
3418
3419  010336
3420  010336  036577  000002  000114            BIT     WHICHB T(R5),@REGSAV
3421  010344  001004                            BNE     $215
3422  010346
3423  010346  112767  000000  000111            MOVB    #CLR,HOLDSC
3424  010354
3425  010354  000403                            BR      $216
3426  010356                          $215
3427  010356
3428  010356  112767  177777  000101            MOVB    #SET,HOLDSC
3429  010364
3430  010364                          $216
3431
3432
3433  010364
```

```
LET     REGSAV  = REG(R5) , GET POINTER TO REGIST
LET     TIMSAV  = HOWLONG(R5) , SAVE HOWLONG FOR
LET     FLAG  B= #FALSE , INITIALIZE THE EXIT FLA


,       START OF AN INFIN TE LOOP

LOOP

        , TEST TO SEE IF WHICHBIT IS SET
    IF      WHICHBIT(R5)  NOTSETIN @REGSAV THEN

        LET     HOLDSC  B= #CLR

    ELSE

        LET     HOLDSC  B= #SET , REMEMBER TH S

    ENDIF

    , NOW SEE IF THAT WAS WHAT WE WANTED
    IFB     HOLDSC EQ SETCLR(R5) THEN
```

```
3434  010364  126765  000075  000006          CMPB    HOLDSC,SETCLR(R5)
3435  010372  001003                          BNE     $217
3436                                                        ; JUST THE THING WE NEEDED
3437  010374                                                    LET     FLAG  B= #TRUE
3438  010374  112767  000001  000062          MOVB    #TRUE,FLAG
3439  010402                                                  ENDIF
3440  010402                          $217
3441
3442  010402                                                EXIFB   FLAG EQ #TRUE OR TIMSAV LE #0
3443  010402  126727  000056  000001          CMPB    FLAG,#TRUE                              ,
3444  010410  001414                          BEQ     $214
3445  010412  005767  000044                  TST     TIMSAV
3446  010416  003411                          BLE     $214
3447                                                        ; ONE WAY OR THE OTHER, WE ARE DONE
3448                                                      ; IF WE ARE STILL HERE THEN HANG AROUND A WHILE
3449
3450  010420                                                    WAITMS  1       ;WAIT FOR 1 MILLI-SECONDS
3451  010420  010546                          MOV     R5,-(SP)
3452  010422  012745  000001                  MOV     #1,-(R5)
3453  010426  004767  000140                  JSR     PC,WAIT
3454  010432  012605                          MOV     (SP)+,R5
3455  010434                                                    LET     TIMSAV  = TIMSAV - #1 ; COUNTING DOWN
3456  010434  005367  000022                  DEC     TIMSAV
3457  010440                                                  ENDLOOP                 ; CONTINUED AT THE TOP
3458  010440  000736                          BR      $213
3459  010442                          $214
3460
3461                                            ; ONLY 2 WAYS TO GET HERE
3462                                            ;       1)  WE RAN OUT OF TIME---ERROR ! !
3463                                            ;       2)  THE BIT IS IN THE CORRECT CONDITION--GOOD ! !
3464
3465  010442                                                    IFB     FLAG EQ #TRUE THEN
3466  010442  126727  000016  000001          CMPB    FLAG,#TRUE
3467  010450  001001                          BNE     $220
3468  010452                                                      RETURN  NO ERROR    ; GOOD
3469  010452  000405                          BR      $211
3470  010454                                                    ENDIF
3471  010454                          $220
3472  010454                                                    RETURN  ERROR           ; BAD
3473  010454  000261                          SEC
3474  010456  000404                          BR      $212
3475
3476  010460  000000                                            REGSAV  WORD 0
3477  010462  000000                                            TIMSAV  WORD 0
3478  010464     000                                            FLAG    BYTE 0
3479  010465     000                                            HOLDSC  BYTE 0
3480                                                  ; WE ARE DONE  GO BACK HOME
3481  010466                                                    ENDRTN
3482  010466                          $211
3483  010466  000241                          CLC
3484  010470                          $212
3485  010470  000207                          RTS     PC
```

```
                                ;;*****************************************************************
010472                          ROUTINE DATLNG (MASK)
010472                          DATLNG
                                ;* ROUTINE DATLNG
                                ;*       THIS ROUTINE SETS UP A MASK FOR DATA, WITH
                                ;*   NPUT - NOTHING IS PASSED TO THIS ROUTINE
                                ;*      BUT GLOBAL INFORMATION IS ASSUMED TO EXIST
                                ;*      $USWR-- THE WORD FOR SOFTWARE PARAMETERS
                                ;*      DATA-- A MASK FOR THE LOCATION OF THE OCTAL
                                ;*               NUMBER OF DATA BITS
                                ;*  OUTPUT----
                                ;*      MASK-- A MASK OF BINARY ONES RIGHT-JUSTIFIED
                                ;*               THE NUMBER OF WHICH IS DEFINED IN $USWR WORD
                                ;*****************************************************************
010472                                                                  LET MASK(R5)  = #0              , START
010472  005065  000000                  CLR     MASK(R5)
010476                                                                  LET NUMBR  = $USWR AND #DATA
010476  016767  170516  000062          MOV     $USWR,NUMBR
010504  016746  000056                  MOV     NUMBR,-(SP)
010510  042716  000017                  BIC     #DATA,(SP)
010514  042667  000046                  BIC     (SP)+,NUMBR
010520                                                                  INCR I FROM #1 TO NUMBR BY #1
010520  012767  000001  170544          MOV     #1,I
010526  000402                          BR      $223
010530                          $224
010530  005267  170536                  INC     I
010534                          $223
010534  026767  170532  000024          CMP     I,NUMBR
010542  003006                          BGT     $225
010544                                                                      LET MASK(R5)  = MASK(R5) SHIFT 1
010544  006365  000000                  ASL     MASK(R5)
010550                                                                      LET MASK(R5)  = MASK(R5) SET BY #1
010550  052765  000001  000000          BIS     #1,MASK(R5)
010556                                                                  ENDINC
010556  000764                          BR      $224
010560                          $225
010560                                                                  LET MASK(R5)  = COMP MASK(R5)
010560  005165  000000                  COM     MASK(R5)
010564                                                              RETURN
010564  000401                          BR      $221
010566  000000                  NUMBR 0
010570                                                          ENDRTN
010570                          $221
010570                          $222
010570  000207                          RTS     PC
```

```

                                        ;;*************************************************************
010572                                  ROUTINE WAIT    <TIME>
010572                                  WAIT
                                        ;* ROUTINE WAIT
                                        ;*       THIS ROUTINE IS USED TO DELAY EXECUTION OF THE
                                        ;*       MAIN  PROGRAM  FOR A SPECIFIED AMOUNT OF TIME
                                        ;*       THIS   IS   ACCOMPLISHED   BY  INCREMENTING  A
                                        ;*       REGISTER UP TO A LIMIT   THE INNER LOOP IS SET
                                        ;*       TO APPROXIMATE 1 MILLI SEC
                                        ;;*************************************************************
010572  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
010574  010246                                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
010576  010346                                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
010600                                          LET R1 := TIME(R5)
010600  016501  000000                          MOV     TIME(R5),R1
010604                                          INCRU R2 FROM #1 TO R1 BY #1
010604  012702  000001                          MOV     #1,R2
010610  000402                                  BR      $230
010612                                  $231
010612  062702  000001                          ADD     #01,R2
010616                                  $230
010616  020201                                  CMP     R2,R1
010620  101010                                  BHI     $232
010622                                                  INCR R3 FROM #0 TO #100 BY #1
010622  005003                                  CLR     R3
010624  000401                                  BR      $233
010626                                  $234
010626  005203                                  INC     R3
010630                                  $233
010630  020327  000100                          CMP     R3,#100
010634  003001                                  BGT     $235
010636                                                  ENDINC
010636  000773                                  BR      $234
010640                                  $235
010640                                          ENDINC
010640  000764                                  BR      $231
010642                                  $232
010642  012603                                  MOV     (SP)+,R3        ;;POP STACK INTO R3
010644  012602                                  MOV     (SP)+,R2        ;;POP STACK INTO R2
010646  012601                                  MOV     (SP)+,R1        ;;POP STACK INTO R1
010650                                  ENDRTN
010650                                  $226
010650                                  $227
010650  000207                                  RTS     PC
```

```
3580
3581                                    SBTTL  INTSRV  INTERRUPT SERVICE ROUTINE
3582                                    ;;*****************************************************************
3583  010652                            NTSRV
3584                                    ;* SERV CE ROUTINE  INTSRV
3585                                    ;*       THIS GLOBAL ROUT NE DOES NOTHING BUT INCREMENT
3586                                    ;*       'INTFLAG'  EACH TIME IT IS CALLED   IT ASSUMES
3587                                    *       THAT THE MAIN CALL NG ROUTINE WILL  KNOW  WHAT
3588                                    *       TO LOOK FOR
3589                                    ;;*****************************************************************
3590
3591                                                ADD 1 TO  INTERRUPT OCCURED' FLAG
3592  010652                                                        LET INTFLAG  = INTFLAG + #1
3593  010652  005267  000002                        NC      INTFLAG
3594  010656                                    ENDSRV                          ;THAT S ALL
3595  010656  000002                            RT
3596  010660  000000                    NTFLAG   0
```

```
3597  010662                          ROUTINE CYCLE
3598  010662                          CYCLE
3599                                  ,,*****************************************************************
3600                                  ,* ROUTINE      CYCLE
3601                                  ,*       THIS  ROUTINE  CAUSES  ADRS   TO POINT TO  THE
3602                                  ,*       ADDRESS  OF  DLV11-F  UNDER  TEST,  ADRS +2 TO
3603                                  ,*       POINT TO THE VECTOR OF THE DLV11-F UNDER TEST
3604                                  ,*      IT  KEEPS  TRACK  OF THE CURRENT DEVICE AND BIT
3605                                  ,*      MASKS  THE CONSOLE IS TREATED SPECIAL BY THIS ROUTINE
3606                                  ,*      IT IS ONLY TESTED ONCE IF UNDER APT  IF NOT UNDER APT
3607                                  ,*      ALL TESTS THAT REQUIRE THE MAINT BIT ARE NOT RUN
3608                                  ,,*****************************************************************
3609  010662                                                  LET APTCON  B= #FALSE   , SET DEFAULT VALUE
3610  010662  112767  000000  000230         MOVB    #FALSE,APTCON
3611  010670                                                  LET CONSOLE  B= #FALSE
3612  010670  112767  000000  000223         MOVB    #FALSE,CONSOLE
3613  010676                                             REPEAT       , UNTIL BITMASK SETIN $DEVM
3614  010676                          $240
3615  010676                                              IF BITMASK EQ #0 THEN
3616  010676  005767  000200                 TST     BITMASK
3617  010702  001027                         BNE     $241
3618  010704                                                  IF INITFLAG EQ #1 THEN
3619  010704  026727  000174  000001         CMP     IN TFLAG,#1
3620  010712  001003                         BNE     $242
3621  010714                                                      LET INITFLAG  = #0
3622  010714  005067  000164                 CLR     INITFLAG
3623  010720                                                  ELSE
3624  010720  000403                         BR      $243
3625  010722                          $242
3626  010722                                                      CALL $EOP    , AS A SUBROUTINE
3627  010722  004767  000370                 JSR     PC,$EOP
3628
3629  010726                          SPECIALADDRESS                  , BECAUSE $EOP RETURNS AS A JUMP
3630  010726                                                          LET R0  = POP
3631  010726  012600                 MOV     (SP)+,R0
3632  010730                                                      ENDIF
3633  010730                          $243
3634  010730                                                      LET BITMASK  = #1
3635  010730  012767  000001  000144         MOV     #1,BITMASK
3636  010736                                                      LET $DEVCT  = #1
3637  010736  012767  000001  170240         MOV     #1,$DEVCT
3638  010744                                                      LET ADDRESS  = $BASE
3639  010744  016767  170300  000134         MOV     $BASE,ADDRESS
3640  010752                                                      LET VECTOR  = $VECT1
3641  010752  016767  170266  000130         MOV     $VECT1,VECTOR
3642  010760                                                  ELSE
3643  010760  000410                         BR      $244
3644  010762                          $241
3645  010762                                                      LET R4  = #10
3646  010762  012704  000010                 MOV     #10,R4
3647  010766                                                      LET BITMASK  = BITMASK ROTATE 1
3648  010766  006167  000110                 ROL     B TMASK
3649  010772                                                      LET ADDRESS  = ADDRESS + R4
3650  010772  060467  000110                 ADD     R4,ADDRESS
3651  010776                                                      LET VECTOR  = VECTOR + R4
3652  010776  060467  000106                 ADD     R4,VECTOR
```

```
3653  011002                                                        ENDIF
3654  011002                        $244
3655  011002                                                    UNT L   BITMASK SET N $DEVM
3656  011002  036767  000074  170242          BIT     B TMASK,$DEVM
3657  011010  001732                          BEQ     $240
3658  011012                                                IF B TMASK EQ #BIT15 THEN
3659  011012  026727  000064  100000          CMP     BITMASK,#B T15
3660  011020  001023                          BNE     $245
3661  011022                                                LET CONSOLE  B= #TRUE
3662  011022  112767  000001  000071          MOVB    #TRUE,CONSOLE
3663  011030                                                LET ADDRESS  = CONADR
3664  011030  016767  000060  000050          MOV     CONADR,ADDRESS
3665  011036                                                LET VECTOR   = CONVECT
3666  011036  016767  000054  000044          MOV     CONVECT,VECTOR
3667                                ;;;;;;                  IF #CONMAINT NOTSETIN $USWR THEN
3668                                ;;;;;;                      LET NOCONMAINT  B= #TRUE
3669                                ;;;;;;                  ENDIF
3670  011044                                                IF #APTENV SETIN $ENV THEN , APT MODE
3671  011044  032767  000001  170142          BIT     #APTENV,$ENV
3672  011052  001406                          BEQ     $246
3673  011054                                                   IF $PASS NE #0 THEN  , NOT FIRST PASS
3674  011054  005767  170122                  TST     $PASS
3675  011060  001403                          BEQ     $247
3676                                                                    , DEFINE DEVICE AS APT CONSOLE
3677  011062                                                        LET APTCON  B= #TRUE
3678  011062  112767  000001  000030          MOVB    #TRUE,APTCON
3679  011070                                                   ENDIF        , FIRST PASS
3680  011070                        $247
3681  011070                                                ENDIF    , APT
3682  011070                        $246
3683  011070                                            ENDIF                , BITMASK
3684  011070                        $245
3685
3686  011070                                                    LET ADRS  = #ADDRESS
3687  011070  012701  011106          MOV     #ADDRESS,ADRS
3688  011074                                                LET $DEVCT  = $DEVCT + #1
3689  011074  005267  170104          INC     $DEVCT
3690  011100                                                RETURN
3691  011100  000411                  BR      $236
3692  011102  100000                B TMASK           100000                  , CONSOLE FIRST
3693  011104  000001                 NITFLAG           1
3694  011106  000000                ADDRESS           0
3695  011110  000000                VECTOR            0
3696  011112  000000                OK                0
3697  011114  177560                CONADR            177560                  , CONSOLE ADDRESS
3698  011116  000060                CONVECT             60                    , CONSOLE VECTOR
3699  011120     000                APTCON             BYTE 0
3700  011121     000                CONSOLE            BYTE 0
3701  011122     000                NOCONMANT          BYTE 0
3702           011124                          EVEN
3703
3704  011124                                                ENDRTN
3705  011124                        $236
3706  011124                        $237
3707  011124  000207                  RTS     PC
3708
```

```
3709
```

```

011126                                ROUT NE MYTYPE
011126                                MYTYPE
                                      ,,**************************************************************
011126  104401  011134                        TYPE    ,65$              ,,TYPE ASCIZ STRING
011132  000405                                BR      64$               ,,GET OVER THE ASCIZ
                                      ,,65$    ASCIZ  <CRLF>*TEST # *
011146                                64$
011146  016746  170026                        MOV     $TESTN,-(SP)      ,,SAVE $TESTN FOR TYPEOUT
011152  104402                                TYPOC                     ,,GO TYPE--OCTAL ASCII(ALL DIGITS)
011154  104401  011162                        TYPE    ,67$              ,,TYPE ASCIZ STRING
011160  000405                                BR      66$               ,,GET OVER THE ASCIZ
                                      ,,67$    ASC Z  *,ERROR # *
011174                                66$
011174  116767  167714  167774                MOVB    $ITEMB,$FATAL     , APT FATAL ERROR NUMBER
011202  016746  167770                        MOV     $FATAL,-(SP)      ,,SAVE $FATAL FOR TYPEOUT
011206  104403                                TYPOS                     ,,GO TYPE--OCTAL ASCII
011210     006                                 BYTE   6                 ,,TYPE 6 DIGITS
011211     000                                 BYTE   0                 ,,SUPPRESS LEADING ZEROS
011212  104401  011220                        TYPE    ,69$              ,,TYPE ASCIZ STRING
011216  000404                                BR      68$               ,,GET OVER THE ASCIZ
                                      ,,69$    ASCIZ  *,PC = *
011230                                68$
011230  016746  167662                        MOV     $ERRPC,-(SP)      ,,SAVE $ERRPC FOR TYPEOUT
011234  104402                                TYPOC                     ,,GO TYPE--OCTAL ASCII(ALL DIGITS)
011236  104401  011244                        TYPE    ,71$              ,,TYPE ASCIZ STRING
011242  000404                                BR      70$               ,,GET OVER THE ASCIZ
                                      ,,71$    ASCIZ  *,CSR  *
011254                                70$
011254  016746  167774                        MOV     DLADD,-(SP)       ,,SAVE DLADD FOR TYPEOUT
011260  104402                                TYPOC                     ,,GO TYPE--OCTAL ASCII(ALL DIGITS)
011262  104401  011270                        TYPE    ,73$              ,,TYPE ASCIZ STRING
011266  000405                                BR      72$               ,,GET OVER THE ASCIZ
                                      ,,73$    ASCIZ  *,VECTOR  *
011302                                72$
011302  016746  167750                        MOV     DLVEC,-(SP)       ,,SAVE DLVEC FOR TYPEOUT
011306  104402                                TYPOC                     ,,GO TYPE--OCTAL ASCII(ALL DIGITS)
011310  104401  001171                        TYPE    ,$CRLF
011314                                ENDRTN
011314                                $250
011314                                $251
011314  000207                                RTS     PC
```

```
                                        SBTTL  END OF PASS ROUTINE

                                        ;;*****************************************************************
                                        ;*INCREMENT THE PASS NUMBER ($PASS)
                                        ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
                                        ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                        ;*IF THERES A MONITOR GO TO IT
                                        ;*IF THERE ISN'T JUMP TO SPECIALADDRESS

011316                                  $EOP:
011316  000004                                  SCOPE
011320  005067  167556                          CLR     $TSTNM          ;;ZERO THE TEST NUMBER
011324  005067  167630                          CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
011330  005267  167646                          INC     $PASS           ;;INCREMENT THE PASS NUMBER
011334  042767  100000  167640                  BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
011342  005327                                  DEC     (PC)+           ;;LOOP?
011344  000001                          $EOPCT: .WORD   1
011346  003022                                  BGT     $DOAGN          ;;YES
011350  012737                                  MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
011352  000001                          $ENDCT: .WORD   1
011354  011344                                  $EOPCT
011356  104401  011423                          TYPE    ,$ENDMG         ;;TYPE "END PASS #"
011362  016746  167614                          MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
011366  104405                                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
011370  104401  011420                          TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
011374  013700  000042                  $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
011400  001405                                  BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
011402  000005                                  RESET                   ;;CLEAR THE WORLD
011404  004710                          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
011406  000240                                  NOP                     ;;SAVE ROOM
011410  000240                                  NOP                     ;;FOR
011412  000240                                  NOP                     ;;ACT11
011414                                  $DOAGN:
011414  000137                                  JMP     @(PC)+          ;;RETURN
011416  010726                          $RTNAD: .WORD   SPECIALADDRESS
011420     377     377     000          $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
011423     015  042412  042116          $ENDMG: .ASCIZ  <15><12>/END PASS #/
011430  050040  051501  020123
011436  000043
```

```
                                        SBTTL  POWER DOWN AND UP ROUTINES

                                ;;*****************************************************************
                                ;POWER DOWN ROUT NE
011440  012737  011604  000024  $PWRDN  MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
011446  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO 7
011454  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
011456  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
011460  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
011462  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
011464  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
011466  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
011470  017746  167444                  MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
011474  010667  000110                  MOV     SP,$SAVR6       ;;SAVE SP
011500  012737  011512  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
011506  000000                          HALT
011510  000776                          BR       -2             ;;HANG UP

                                ;;*****************************************************************
                                ;POWER UP ROUTINE
011512  012737  011604  000024  $PWRUP  MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
011520  016706  000064                  MOV     $SAVR6,SP       ;;GET SP
011524  005067  000060                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
011530  005267  000054          1$      INC     $SAVR6          ;;WAIT FOR THE INC
011534  001375                          BNE     1$              ;;OF  WORD
011536  012677  167376                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
011542  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
011544  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
011546  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
011550  012602                          MC      (SP)+,R2        ;;POP STACK INTO R2
011552  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
011554  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
011556  012737  011440  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
011564  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO 7
011572  104401                          TYPE                    ;REPORT THE POWER FAILURE
011574  011612                  $PWRMG   WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
011576  012716                  MOV     (PC)+,(SP)      ;;RESTART AT START
011600  001336                  $PWRAD   WORD   START           ;;RESTART ADDRESS
011602  000002                  RTI
011604  000000                  $ILLUP  HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
011606  000776                  BR       -2                     ;;  BEFORE THE POWER DOWN WAS COMPLETE
011610  000000                  $SAVR6  0                       ;;PUT THE SP HERE
011612  005015  047520  042527  $POWER   ASCIZ  <15><12>"POWER"
011620  000122
                                 EVEN
```

```
                                          SBTTL  TYPE ROUTINE

                                         ;;*******************************************************************
                                         ;*ROUTINE TO TYPE ASCIZ MESSAGE  MESSAGE MUST TERMINATE WITH A 0 BYTE
                                         ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED
                                         ;*NOTE1         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER
                                         ;*NOTE2         $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED
                                         ;*NOTE3         $FILLC CONTAINS THE CHARACTER TO FILL AFTER
                                         ;*
                                         ;*CALL
                                         ;*1) USING A TRAP INSTRUCTION
                                         ;*       TYPE    ,MESADR           ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                         ;*OR
                                         ;*       TYPE
                                         ;*       MESADR
                                         ;*

 011622  105767  167331          $TYPE   TSTB    $TPFLG           ;;IS THERE A TERMINAL?
 011626  100002                          BPL     1$               ;;BR IF YES
 011630  000000                          HALT                     ;;HALT HERE IF NO TERMINAL
 011632  000430                          BR      3$               ;;LEAVE
 011634  010046                  1$      MOV     R0,-(SP)         ;;SAVE R0
 011636  017600  000002                  MOV     @2(SP),R0        ;;GET ADDRESS OF ASCIZ STRING
 011642  122767  000001  167344          CMPB    #APTENV,$ENV     ;;RUNNING IN APT MODE
 011650  001011                          BNE     62$              ;;NO,GO CHECK FOR APT CONSOLE
 011652  132767  000100  167335          BITB    #APTSPOOL,$ENVM  ;;SPOOL MESSAGE TO APT
 011660  001405                          BEQ     62$              ;;NO,GO CHECK FOR CONSOLE
 011662  010067  000004                  MOV     R0,61$           ;;SETUP MESSAGE ADDRESS FOR APT
 011666  004767  000774                  JSR     PC,$ATY3         ;;SPOOL MESSAGE TO APT
 011672  000000                  61$     .WORD   0                ;;MESSAGE ADDRESS
 011674  132767  000040  167313  62$     BITB    #APTCSUP,$ENVM   ;;APT CONSOLE SUPPRESSED
 011702  001003                          BNE     60$              ;;YES,SKIP TYPE OUT
 011704  112046                  2$      MOVB    (R0)+,-(SP)      ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 011706  001005                          BNE     4$               ;;BR IF IT ISN'T THE TERMINATOR
 011710  005726                          TST     (SP)+            ;;IF TERMINATOR POP IT OFF THE STACK
 011712  012600                  60$     MOV     (SP)+,R0         ;;RESTORE R0
 011714  062716  000002          3$      ADD     #2,(SP)          ;;ADJUST RETURN PC
 011720  000002                          RTI                      ;;RETURN
 011722  122716  000011          4$      CMPB    #HT,(SP)         ;;BRANCH IF <HT>
 011726  001430                          BEQ     8$
 011730  122716  000200                  CMPB    #CRLF,(SP)       ;;BRANCH IF NOT <CRLF>
 011734  001006                          BNE     5$
 011736  005726                          TST     (SP)+            ;;POP  <CR><LF> EQUIV
 011740  104401                          TYPE                     ;;TYPE A CR AND LF
 011742  001171                          $CRLF
 011744  105067  000130                  CLRB    $CHARCNT         ;;CLEAR CHARACTER COUNT
 011750  000755                          BR      2$               ;;GET NEXT CHARACTER
 011752  004767  000056          5$      JSR     PC,$TYPEC        ;;GO TYPE THIS CHARACTER
 011756  126726  167174          6$      CMPB    $FILLC,(SP)+     ;;IS IT TIME FOR FILLER CHARS ?
 011762  001350                          BNE     2$               ;;IF NO GO GET NEXT CHAR
 011764  016746  167164                  MOV     $NULL,-(SP)      ;;GET # OF FILLER CHARS  NEEDED
                                                                  ;;AND THE NULL CHAR
 011770  105366  000001          7$      DECB    1(SP)            ;;DOES A NULL NEED TO BE TYPED?
 011774  002770                          BLT     6$               ;;BR IF NO--GO POP THE NULL OFF OF STACK
 011776  004767  000032                  JSR     PC,$TYPEC        ;;GO TYPE A NULL
 012002  105367  000072                  DECB    $CHARCNT         ;;DO NOT COUNT AS A COUNT
```

```
3892  012006  000770                        BR      7$              ;;LOOP
3893
3894                                  ;HORIZONTAL TAB PROCESSOR
3895
3896  012010  112716  000040          8$      MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
3897  012014  004767  000014          9$      JSR     PC,$TYPEC       ;;TYPE A SPACE
3898  012020  132767  000007  000052          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
3899  012026  001372                          BNE     9$              ;;TAB STOP
3900  012030  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
3901  012032  000724                          BR      2$              ;;GET NEXT CHARACTER
3902  012034  105777  167110          $TYPEC  TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
3903  012040  100375                          BPL     $TYPEC
3904  012042  116677  000002  167102          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG
3905  012050  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
3906  012056  001003                          BNE     1$              ;;BRANCH IF NO
3907  012060  105067  000014                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
3908  012064  000406                          BR      $TYPEX          ;;EXIT
3909  012066  122766  000012  000002  1$      CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
3910  012074  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
3911  012076  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
3912  012100  000000                  $CHARCNT  WORD  0               ;;CHARACTER COUNT STORAGE
3913  012102  000207                  $TYPEX  RTS     PC
3914
```

```
                                                SBTTL  TTY INPUT ROUTINE

                                        ;;*************************************************************
                                                ENABL  LSB

                                        ;;*************************************************************
                                        ;;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE
                                        ;;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
                                        ;;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
                                        ;;*WHEN OPERATING IN TTY FLAG MODE
012104  022767  000176  167026  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
012112  001074                          BNE     15$             ;;BRANCH IF NO
012114  105777  167024                  TSTB    @$TKS           ;;CHAR THERE?
012120  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
012122  117746  167020                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
012126  042716  177600                  BIC     # C177,(SP)     ;;STRIP-OFF THE ASCII
012132  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
012136  001062                          BNE     15$             ;;NO, RETURN TO USER
012140  126727  166770  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
012146  001456                          BEQ     15$             ;;BRANCH IF YES

012150  104401  012631                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G ( G)
012154  104401  012636          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
012160  016746  166012                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
012164  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
012166  104401  012647                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
012172  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
012174  005046                          CLR     -(SP)           ;;THE NEW SWR
012176  105777  166742          7$:     TSTB    @$TKS           ;;CHAR THERE?
012202  100375                          BPL     7$              ;;IF NOT TRY AGAIN

012204  117746  166736                  MOVB    @$TKB,-(SP)     ;;P CK UP CHAR
012210  042716  177600                  BIC     # C177,(SP)     ;;MAKE IT 7-BIT ASCII



012214  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
012220  001005                          BNE     10$             ;;BRANCH IF NOT
012222  104401  012624                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U ( U)
012226  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
012232  000757                          BR      19$             ;;LET'S TRY IT AGAIN


012234  021627  000015          10$:    CMP     (SP),#15        ;;IS IT ^ <CR>?
012240  001022                          BNE     16$             ;;BRANC   NO
012242  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
012246  001403                          BEQ     11$             ;;BRANCH IF YES
012250  016677  000002  166662          MOV     2(SP),@SWR      ;;SAVE NEW SWR
012256  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
012262  104401  001171          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
012266  126727  166643  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
012274  001003                          BNE     15$             ;;BRANCH IF NOT
012276  012777  000100  166640          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
012304  000002                  15$:    RTI                     ;;RETURN
012306  004767  177522          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
012312  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
```

```
3971  012316  002420                          BLT     18$             ;;BRANCH IF YES
3972  012320  021627  000067                  CMP     (SP),#67        ;;CHAR > 77
3973  012324  003015                          BGT     18$             ;;BRANCH IF YES
3974  012326  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
3975  012332  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
3976  012336  001403                          BEQ     17$             ;;BRANCH IF YES
3977  012340  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
3978  012342  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
3979  012344  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE
3980  012346  005266  000002          17$     INC     2(SP)           ;;KEEP COUNT OF CHAR
3981  012352  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
3982  012356  000707                          BR      7$              ;;GET THE NEXT ONE
3983  012360  104401  001170          18$     TYPE    ,$QUES          ;;TYPE ?<CR><LF>
3984  012364  000720                          BR      20$             ;;SIMULATE CONTROL-U
3985                                  DSABL   LSB
3986
3987
3988                                  ;;*****************************************************************
3989                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
3990                                  ;*CALL
3991                                  ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
3992                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
3993                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
3994                                  ;
3995
3996  012366  011646                  $RDCHR  MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
3997  012370  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
3998  012376  105777  166542          1$      TSTB    @$TKS           ;;WAIT FOR
3999  012402  100375                          BPL     1$              ;;A CHARACTER
4000  012404  117766  166536  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
4001  012412  042766  177600  000004          BIC     #^C(177),4(SP)  ;;GET RID OF JUNK IF ANY
4002  012420  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
4003  012426  001013                          BNE     3$              ;;BRANCH IF NO
4004  012430  105777  166510          2$      TSTB    @$TKS           ;;WAIT FOR A CHARACTER
4005  012434  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
4006  012436  117746  166504                  MOVB    @$TKB,-(SP)     ;;GET CHARACTER
4007  012442  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
4008  012446  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
4009  012452  001366                          BNE     2$              ;;IF NOT DISCARD IT
4010  012454  000750                          BR      1$              ;;YES, RESUME
4011  012456  026627  000004  000140  3$      CMP     4(SP),#140      ;;IS IT UPPER CASE?
4012  012464  002407                          BLT     4$              ;;BRANCH IF YES
4013  012466  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
4014  012474  003003                          BGT     4$              ;;BRANCH IF YES
4015  012476  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
4016  012504  000002                  4$      RTI                     ;;GO BACK TO USER
4017                                  ;;*****************************************************************
4018                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
4019                                  ;*CALL
4020                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
4021                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
4022                                  ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
4023
4024  012506  010346                  $RDLIN  MOV     R3,-(SP)        ;;SAVE R3
4025  012510  012703  012614          1$      MOV     #$TTYIN,R3      ;;GET ADDRESS
4026  012514  022703  012624          2$      CMP     #$TTYIN+8,R3    ;;BUFFER FULL?
```

```
 4027  012520  101405                          BLOS    4$              ;;BR IF YES
 4028  012522  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
 4029  012524  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
 4030  012526  122713  000177          10$     CMPB    #177,(R3)       ;;IS IT A RUBOUT
 4031  012532  001003                          BNE     3$              ;;SKIP IF NOT
 4032  012534  104401  001170          4$      TYPE    ,$QUES          ;;TYPE A '?'
 4033  012540  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
 4034  012542  111367  000044          3$      MOVB    (R3),9$         ;;ECHO THE CHARACTER
 4035  012546  104401  012612                  TYPE    ,9$
 4036  012552  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
 4037  012556  001356                          BNE     2$              ;;LOOP IF NOT RETURN
 4038  012560  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
 4039  012564  104401  001172                  TYPE    ,$LF            ;;TYPE A LINE FEED
 4040  012570  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 4041  012572  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 4042  012574  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
 4043  012602  012766  012614  000004          MOV     #$TTYIN,4(SP)
 4044  012610  000002                          RTI                     ;;RETURN
 4045  012612     000          9$       BYTE    0                       ;;STORAGE FOR ASCII CHAR  TO TYPE
 4046  012613     000                   BYTE    0                       ;;TERMINATOR
 4047  012614  000010          $TTYIN   BLKB    8                       ;;RESERVE 8 BYTES FOR TTY INPUT
 4048  012624  052536  005015     000  $CNTLU   ASCIZ  / U/<15><12>     ;;CONTROL "U"
 4049  012631     136  006507  000012  $CNTLG   ASC Z  / G/<15><12>     ;;CONTROL "G"
 4050  012636  005015  053523  020122  $MSWR    ASCIZ  <15><12>/SWR = /
 4051  012644  020075     000
 4052  012647     040  047040  053505  $MNEW    ASCIZ  /  NEW = /
 4053  012654  036440  000040
```

```
                        SBTTL  APT COMMUNICATIONS ROUTINE

                        ;;*****************************************************************
012660  112767  000001  000236  $ATY1    MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
012666  112767  000001  000226  $ATY3    MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
012674  000403                  $ATY4    BR      $ATYC
012676  112767  000001  000220  $ATY4    MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
012704                          $ATYC
012704  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
012706  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
012710  105767  000206                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
012714  001450                          BEQ     5$              ;;IF NOT   BR
012716  122767  000001  166270          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
012724  001031                          BNE     3$              ;;IF NOT   BR
012726  132767  000100  166261          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
012734  001425                          BEQ     3$              ;;IF NOT   BR
012736  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR
012742  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR
012750  005767  166220          1$      TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
012754  001375                          BNE     1$              ;;IF NOT   WAIT
012756  010067  166226                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
012762  105720                  2$      TSTB    (R0)+           ;;FIND END OF MESSAGE
012764  001376                          BNE     2$
012766  166700  166216                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
012772  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
012774  010067  166212                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
013000  012767  000004  166166          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG
013006  000413                          BR      5$
013010  017667  000004  000016  3$      MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR L NKAGE
013016  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
013024  016746  164746                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
013030  004767  176566                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
013034  000000                  4$       WORD   0
013036                          5$
013036  105767  000062          10$     TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
013042  001416                          BEQ     12$             ;;  F NOT   BR
013044  005767  166144                  TST     $ENV            ;;RUNNING UNDER APT?
013050  001413                          BEQ     12$             ;;IF NOT   BR
013052  005767  166116          11$     TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
013056  001375                          BNE     11$             ;;IF NOT   WAIT
013060  017667  000004  166110          MOV     @4(SP),$FATAL   ;;GET ERROR #
013066  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR
013074  005267  166074                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
013100  105067  000020          12$     CLRB    $FFLG           ;;CLEAR FATAL FLAG
013104  105067  000013                  CLRB    $LFLG           ;;CLEAR LOG FLAG
013110  105067  000006                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
013114  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
013116  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
013120  000207                          RTS     PC              ;;RETURN
013122     000          $MFLG    BYTE   0               ;;MESSG  FLAG
013123     000          $LFLG    BYTE   0               ;;LOG FLAG
013124     000          $FFLG    BYTE   0               ;;FATAL FLAG
013126                          EVEN
000200                  APTSIZE=200
000001                  APTENV=001
000100                  APTSPOOL=100
```

```
 4110           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

013126                                    $ERROR
013126  104407                                    CKSWR                     ;;TEST FOR CHANGE IN SOFT-SWR
013130  105267  165747                    7$:     INCB    $ERFLG            ;;SET THE ERROR FLAG
013134  001775                                    BEQ     7$                ;;DON'T LET THE FLAG GO TO ZERO
013136  016777  165740  165776                    MOV     $TSTNM,@DISPLAY   ;;DISPLAY TEST NUMBER AND ERROR FLAG
013144  032777  002000  165766                    BIT     #BIT10,@SWR       ;;BELL ON ERROR?
013152  001402                                    BEQ     1$                ;;NO - SKIP
013154  104401  001164                            TYPE    ,$BELL            ;;RING BELL
013160  005267  165726                    1$:     INC     $ERTTL            ;;COUNT THE NUMBER OF ERRORS
013164  011667  165726                            MOV     (SP),$ERRPC       ;;GET ADDRESS OF ERROR INSTRUCTION
013170  162767  000002  165720                    SUB     #2,$ERRPC
013176  117767  165714  165710                    MOVB    @$ERRPC,$ITEMB    ;;STRIP AND SAVE THE ERROR ITEM CODE
013204  032777  020000  165726                    BIT     #BIT13,@SWR       ;;SKIP TYPEOUT IF SET
013212  001004                                    BNE     20$               ;;SKIP TYPEOUTS
013214  004767  175706                            JSR     PC,MYTYPE         ;;GO TO USER ERROR ROUTINE
013220  104401  001171                            TYPE    ,$CRLF
013224                                    20$:
013224  122767  000001  165762                    CMPB    #APTENV,$ENV      ;;RUNNING IN APT MODE
013232  001007                                    BNE     2$                ;;NO,SKIP APT ERROR REPORT
013234  116767  165654  000004                    MOVB    $ITEMB,21$        ;;SET ITEM NUMBER AS ERROR NUMBER
013242  004767  177430                            JSR     PC,$ATY4          ;;REPORT FATAL ERROR TO APT
013246     000                            21$:    .BYTE   0
013247     000                                    .BYTE   0
013250  000777                            22$:    BR      22$               ;;APT ERROR LOOP
013252  005777  165662                    2$:     TST     @SWR              ;;HALT ON ERROR
013256  100002                                    BPL     3$                ;;SKIP IF CONTINUE
013260  000000                                    HALT                      ;;HALT ON ERROR!
013262  104407                                    CKSWR                     ;;TEST FOR CHANGE IN SOFT-SWR
013264  032777  001000  165646            3$:     BIT     #BIT09,@SWR       ;;LOOP ON ERROR SWITCH SET?
013272  001402                                    BEQ     4$                ;;BR IF NO
013274  016716  165610                            MOV     $LPERR,(SP)       ;;FUDGE RETURN FOR LOOPING
013300  005767  165656                    4$:     TST     $ESCAPE           ;;CHECK FOR AN ESCAPE ADDRESS
013304  001402                                    BEQ     5$                ;;BR IF NONE
013306  016716  165650                            MOV     $ESCAPE,(SP)      ;;FUDGE RETURN ADDRESS FOR ESCAPE
013312                                    5$:
013312  022737  011404  000042                    CMP     #$ENDAD,@#42      ;;ACT-11 AUTO-ACCEPT?
013320  001001                                    BNE     6$                ;;BRANCH IF NO
013322  000000                                    HALT                      ;;YES
013324                                    6$:
013324  000002                                    RTI                       ;;RETURN
```

```
                                SBTTL  SCOPE HANDLER ROUT NE

                        ;;**********************************************************************
                        ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS  IT WILL INCREMENT
                        ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG (DISPLAY<7 0>)
                        ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15 08>
                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE
                        ;*SW14=1         LOOP ON TEST
                        ;*SW11=1         INHIBIT ITERATIONS
                        ;*SW09=1         LOOP ON ERROR
                        ;*SW08=1         LOOP ON TEST IN SWR<7 0>
                        ;*CALL
                        ;*      SCOPE           ;;SCOPE=IOT

013326                  $SCOPE
013326  104407                 CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
013330  032777  040000  165602 1$      BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
013336  001114                 BNE     $OVER           ;;YES IF SW14=1
                        ;#####START OF CODE FOR THE XOR TESTER#####
013340  000416          $XTSTR BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                       ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
013342  013746  000004         MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
013346  012737  013366  000004 MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
013354  005737  177060         TST     @#177060        ;;TIME OUT ON XOR?
013360  012637  000004         MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
013364  000463                 BR      $SVLAD          ;;GO TO THE NEXT TEST
013366  022626          5$     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
013370  012637  000004         MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
013374  000423                 BR      7$              ;;LOOP ON THE PRESENT TEST
013376                  6$ ;#####END OF CODE FOR THE XOR TESTER#####
013376  032777  000400  165534         BIT     #BIT08,@SWR     ;;LOOP ON SPEC  TEST?
013404  001404                 BEQ     2$              ;;BR IF NO
013406  127767  165526  165466         CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7 0>
013414  001465                 BEQ     $OVER           ;;BR IF YES
013416  105767  165461          2$     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
013422  001421                 BEQ     3$              ;;BR IF NO
013424  126767  165465  165451         CMPB    $ERMAX,$ERFLG   ;;MAX  ERRORS FOR THIS TEST OCCURRED?
013432  101015                 BH      3$              ;;BR IF NO
013434  032777  001000  165476         BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
013442  001404                 BEQ     4$              ;;BR IF NO
013444  016767  165440  165434 7$      MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
013452  000446                 BR      $OVER
013454  105067  165423          4$     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
013460  005067  165474         CLR     $T MES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
013464  000415                 BR      1$              ;;ESCAPE TO THE NEXT TEST
013466  032777  004000  165444 3$      BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
013474  001011                 BNE     1$              ;;BR IF YES
013476  005767  165500         TST     $PASS           ;;IF FIRST PASS OF PROGRAM
013502  001406                 BEQ     1$              ;;      INHIBIT ITERATIONS
013504  005267  165374          NC     $ICNT           ;;INCREMENT ITERATION COUNT
013510  026767  165444  165366         CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
013516  002024                 BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
013520  012767  000001  165356 1$      MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
013526  016767  000052  165424         MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
013534  105267  165342          $SVLAD INCB    $TSTNM          ;;COUNT TEST NUMBERS
013540  116767  165336  165432         MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER  N APT MA LBOX
```

```
 4221  013546  011667  165334                MOV     (SP),$LPADR       ;SAVE SCOPE LOOP ADDRESS
 4222  013552  011667  165332                MOV     (SP),$LPERR       ;SAVE ERROR LOOP ADDRESS
 4223  013556  005067  165400                CLR     $ESCAPE           ;CLEAR THE ESCAPE FROM ERROR ADDRESS
 4224  013562  112767  000001  165325        MOVB    #1,$ERMAX         ;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 4225  013570  016777  165306  165344 $OVER  MOV     $TSTNM,@D SPLAY   ;DISPLAY TEST NUMBER
 4226  013576  016716  165304                MOV     $LPADR,(SP)       ;FUDGE RETURN ADDRESS
 4227  013602  000002                        RTI                       ;F XES PS
 4228  013604  003720                $MXCNT  2000                      ;MAX  NUMBER OF ITERATIONS
```

```
                                        SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

                                        ;;****************************************************************
                                        ;;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                        ;;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT  DEPENDING ON WHETHER THE
                                        ;;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                        ;;*BEFORE THE FIRST DIGIT OF THE NUMBER  LEADING ZEROS WILL ALWAYS BE
                                        ;;*REPLACED WITH SPACES
                                        ;;*CALL
                                        ;;*       MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                        ;;*       TYPDS                   ;;GO TO THE ROUTINE

013606                                  $TYPDS
013606  010046                                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
013610  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
013612  010246                                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
013614  010346                                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
013616  010546                                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
013620  012746  020200                          MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
013624  016605  000020                          MOV     20(SP),R5       ;;GET THE INPUT NUMBER
013630  100004                                  BPL     1$              ;;BR IF INPUT IS POS
013632  005405                                  NEG     R5              ;;MAKE THE BINARY NUMBER POS
013634  112766  000055  000001                  MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG
013642  005000                          1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
013644  012703  014022                          MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
013650  112723  000040                          MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
013654  005002                          2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
013656  016001  014012                          MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
013662  160105                          3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
013664  002402                                  BLT     4$              ;;BR  F DONE
013666  005202                                  INC     R2              ;;INCREASE THE BCD DIGIT BY 1
013670  000774                                  BR      3$
013672  060105                          4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
013674  005702                                  TST     R2              ;;CHECK IF BCD DIGIT=0
013676  001002                                  BNE     5$              ;;FALL THROUGH IF 0
013700  105716                                  TSTB    (SP)            ;;STILL DOING LEADING 0'S?
013702  100407                                  BMI     7$              ;;BR IF YES
013704  106316                          5$:     ASLB    (SP)            ;;MSD?
013706  103003                                  BCC     6$              ;;BR IF NO
013710  116663  000001  177777                  MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
013716  052702  000060                  6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
013722  052702  000040                  7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIG T
013726  110223                                  MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
013730  005720                                  TST     (R0)+           ;;JUST INCREMENTING
013732  020027  000010                          CMP     R0,#10          ;;CHECK THE TABLE INDEX
013736  002746                                  BLT     2$              ;;GO DO THE NEXT DIGIT
013740  003002                                  BGT     8$              ;;GO TO EXIT
013742  010502                                  MOV     R5,R2           ;;GET THE LSD
013744  000764                                  BR      6$              ;;GO CHANGE TO ASCII
013746  105726                          8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
013750  100003                                  BPL     9$              ;;BR IF NO
013752  116663  177777  177776                  MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYP NG
013760  105013                          9$:     CLRB    (R3)            ;;SET THE TERMINATOR
013762  012605                                  MOV     (SP)+,R5        ;;POP STACK INTO R5
013764  012603                                  MOV     (SP)+,R3        ;;POP STACK INTO R3
013766  012602                                  MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
 4285  013770  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 4286  013772  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 4287  013774  104401  014022                TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 4288  014000  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 4289  014006  012616                        MOV     (SP)+,(SP)
 4290  014010  000002                        RTI                     ;;RETURN TO USER
 4291  014012  023420                $DTBL   10000
 4292  014014  001750                        1000
 4293  014016  000144                        100
 4294  014020  000012                        10
 4295  014022  000004                $DBLK    BLKW   4
```

```
                                SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                        ;*****************************************************************
                        ;*TH S ROUTINE IS USED TO CHANGE A 16-BIT B NARY NUMBER TO A 6-DIG T
                        ;*OCTAL (ASCII) NUMBER AND TYPE IT
                        ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                        ;*CALL
                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                        ;*      TYPOS                   ;;CALL FOR TYPEOUT
                        ;*       BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                        ;*       BYTE   M               ;;M=1 OR 0
                        ;*                                      ;;1=TYPE LEADING ZEROS
                        ;*                                      ;;0=SUPPRESS LEADING ZEROS
                        ;*
                        ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                        ;*$TYPOS OR $TYPOC
                        ;*CALL
                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                        ;*      TYPON                   ;;CALL FOR TYPEOUT
                        ;*
                        ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                        ;*CALL
                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                        ;*      TYPOC                   ;;CALL FOR TYPEOUT
014032  017646  000000  $TYPOS  MOV     @(SP),-(SP)     ;;PICKUP THE MODE
014036  116667  000001  000211  MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
014044  112667  000207          MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
014050  062716  000002          ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
014054  000406                  BR      $TYPON
014056  112767  000001  000171  $TYPOC  MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
014064  112767  000006  000165          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
014072  112767  000005  000154  $TYPON  MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
014100  010346                  MOV     R3,-(SP)        ;;SAVE R3
014102  010446                  MOV     R4,-(SP)        ;;SAVE R4
014104  010546                  MOV     R5,-(SP)        ;;SAVE R5
014106  116704  000145          MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
014112  005404                  NEG     R4
014114  062704  000006          ADD     #6,R4           ;;SUBTRACT IT FOR MAX  ALLOWED
014120  110467  000132          MOVB    R4,$OMODE       ;;SAVE IT FOR USE
014124  116704  000125          MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
014130  016605  000012          MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
014134  005003                  CLR     R3              ;;CLEAR THE OUTPUT WORD
014136  006105          1$      ROL     R5              ;;ROTATE MSB INTO "C"
014140  000404                  BR      3$              ;;GO DO MSB
014142  006105          2$      ROL     R5              ;;FORM THIS DIGIT
014144  006105                  ROL     R5
014146  006105                  ROL     R5
014150  010503                  MOV     R5,R3
014152  006103          3$      ROL     R3              ;;GET LSB OF THIS DIGIT
014154  105367  000076          DECB    $OMODE          ;;TYPE THIS DIGIT?
014160  100016                  BPL     7$              ;;BR IF NO
014162  042703  177770          BIC     #177770,R3      ;;GET RID OF JUNK
014166  001002                  BNE     4$              ;;TEST FOR 0
014170  005704                  TST     R4              ;;SUPPRESS THIS 0?
014172  001403                  BEQ     5$              ;;BR IF YES
```

```
 4352  014174  005204                  4$      INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 4353  014176  052703  000060          B S     #'0,R3          ;;MAKE THIS DIGIT ASCII
 4354  014202  052703  000040          5$      B S     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 4355  014206  110367  000040          MOVB    R3,8$           ;;SAVE FOR TYPING
 4356  014212  104401  014252          TYPE    ,8$             ;;GO TYPE THIS DIGIT
 4357  014216  105367  000032          7$      DECB    $OCNT           ;;COUNT BY 1
 4358  014222  003347                  BGT     2$              ;;BR IF MORE TO DO
 4359  014224  002402                  BLT     6$              ;;BR IF DONE
 4360  014226  005204                  INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 4361  014230  000744                  BR      2$              ;;GO DO THE LAST DIGIT
 4362  014232  012605                  6$      MOV     (SP)+,R5        ;;RESTORE R5
 4363  014234  012604                  MOV     (SP)+,R4        ;;RESTORE R4
 4364  014236  012603                  MOV     (SP)+,R3        ;;RESTORE R3
 4365  014240  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 4366  014246  012616                  MOV     (SP)+,(SP)
 4367  014250  000002                  RTI                             ;;RETURN
 4368  014252     000                  8$       BYTE   0               ;;STORAGE FOR ASCII DIGIT
 4369  014253     000                           BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 4370  014254     000                  $OCNT    BYTE   0               ;;OCTAL DIGIT COUNTER
 4371  014255     000                  $OFILL   BYTE   0               ;;ZERO FILL SWITCH
 4372  014256  000000                  $OMODE   WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
                                           SBTTL  TRAP DECODER

                                          ;;*********************************   **************************************
                                          ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                                          ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                          ;*OF THE DESIRED ROUTINE  THEN USING THE ADDRESS OBTAINED IT WILL
                                          ;*GO TO THAT ROUTINE

 014260  010046                           $TRAP   MOV     R0,-(SP)          ;;SAVE R0
 014262  016600  000002                           MOV     2(SP),R0          ;;GET TRAP ADDRESS
 014266  005740                                   TST     -(R0)             ;;BACKUP BY 2
 014270  111000                                   MOVB    (R0),R0           ;;GET RIGHT BYTE OF TRAP
 014272  006300                                   ASL     R0                ;;POSITION FOR INDEXING
 014274  016000  014314                           MOV     $TRPAD(R0),R0     ;;INDEX TO TABLE
 014300  000200                                   RTS     R0                ;;GO TO ROUTINE


                                          ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

 014302  011646                           $TRAP2  MOV     (SP),-(SP)        ;;MOVE THE PC DOWN
 014304  016666  000004  000002                   MOV     4(SP),2(SP)       ;;MOVE THE PSW DOWN
 014312  000002                                   RTI                       ;;RESTORE THE PSW

                                           SBTTL  TRAP TABLE

                                          ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                          ;*BY THE "TRAP" INSTRUCTION

                                          ;       ROUTINE
                                          ;       -------
 014314  014302                           $TRPAD   WORD   $TRAP2
 014316  011622                                   $TYPE   ;;CALL=TYPE      TRAP+1(104401)  TTY TYPEOUT ROUTINE
 014320  014056                                   $TYPOC  ;;CALL=TYPOC     TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
 014322  014032                                   $TYPOS  ;;CALL=TYPOS     TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
 014324  014072                                   $TYPON  ;;CALL=TYPON     TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
 014326  013606                                   $TYPDS  ;;CALL=TYPDS     TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

 014330  012154                                   $GTSWR  ;;CALL=GTSWR     TRAP+6(104406)  GET SOFT-SWR SETTING

 014332  012104                                   $CKSWR  ;;CALL=CKSWR     TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
 014334  012366                                   $RDCHR  ;;CALL=RDCHR     TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
 014336  012506                                   $RDLIN  ;;CALL=RDLIN     TRAP+11(104411) TTY TYPEIN STRING ROUTINE
         000001                            END
```

```
ABASE = 175610          1#    956     997
ACDW1 = 000000          956
ACDW2 = 000000          956
ACPUOP= 000000          956     971
ADDRES  011106         3639*   3650*   3664*   3687    3694#
ADDW0 = 000000          956
ADDW1 = 000000          956
ADDW10= 000000          956
ADDW11= 000000          956
ADDW12= 000000          956
ADDW13= 000000          956
ADDW14= 000000          956
ADDW15= 000000          956
ADDW2 = 000000          956
ADDW3 = 000000          956
ADDW4 = 000000          956
ADDW5 = 000000          956
ADDW6 = 000000          956
ADDW7 = 000000          956
ADDW8 = 000000          956
ADDW9 = 000000          956
ADEVCT= 000000          956     962
ADEVM = 100000          1#      956     998
AENV  = 000000          956     967
AENVM = 000000          956     968
AFATAL= 000000          956     959
AMADR1= 000000          956     984
AMADR2= 000000          956     988
AMADR3= 000000          956     991
AMADR4= 000000          956     994
AMAMS1= 000000          956     978
AMAMS2= 000000          956     986
AMAMS3= 000000          956     989
AMAMS4= 000000          956     992
AMSGAD= 000000          956     964
AMSGLG= 000000          956     965
AMSGTY= 000000          956     958
AMTYP1= 000000          956     979
AMTYP2= 000000          956     987
AMTYP3= 000000          956     990
AMTYP4= 000000          956     993
APASS = 000000          956     96?
APRIOR= 000000          956
APTCON  011120         3610*   3678*   3699#
APTCSU= 000040         3866    4110#
APTENV= 000001         1217    1695    1982    2318    2492    3671    3859    4066    4108#   4142
APTSIZ= 000200         1067    4107#
APTSPO= 000100         3861    4068    4109#
ASWREG= 000000          956     969
ATESTN= 000000          956     960
AUNIT = 000000          956     963
AUSWR = 011110          1#      956     970
AVECT1= 000300          1#      956     995
AVECT2= 000000          956     996
BAUD  = 007400          859#
B TMAS  011102         1106*   3616    3635*   3648*   3656    3659    3692#
```

```
BIT0   = 000001   734#  3310  3333
BIT00  = 000001   724#   734   783   803   824   845
BIT01  = 000002   723#   733   802   844
BIT02  = 000004   722#   732   801   822   843
BIT03  = 000010   721#   731   800   842
BIT04  = 000020   720#   730   799   841
BIT05  = 000040   719#   729   798   840
BIT06  = 000100   718#   728   777   797   818   839
BIT07  = 000200   717#   727   776   796   817   838
BIT08  = 000400   716#   726  4195
BIT09  = 001000   715#   725  4153  4203
BIT1   = 000002   733#  3321  3340
BIT10  = 002000   714#  4130
BIT11  = 004000   713#   772   812  4210
BIT12  = 010000   712#   791   811
BIT13  = 020000   711#   790   810  4137
BIT14  = 040000   710#   789   809  4181
BIT15  = 100000   709#   788   808  1106  3659
BIT2   = 000004   732#  2406  2434
BIT3   = 000010   731#  2810  2870
BIT4   = 000020   730#  2829  2877
BIT5   = 000040   729#
BIT6   = 000100   728#
BIT7   = 000200   727#
BIT8   = 000400   726#
BIT9   = 001000   725#
BPTVEC= 000014   741#
BREAK  = 000001   824#  1231  1244  1247  1261  1264  1278  1283  3284
BRK    = 010000   860#  1215  3260
CKSWR  = 104407  4126  4152  4180  4412#
CLR    = 000000   762#  1710  1712  1995  3423
COMSPD= 000100   854#  2897
CONADR  011114  3664  3697#
CONSOL  011121  1305  1591  1693  1828  1902  1971  2128  2637  2757  3039  3269  3612*  3662*
                3700#
CONVEC  011116  3666  3698#
CR     = 000015   649#  3905  3915
CRLF   = 000200   650#  1091  1097  1098  1099  3360  3717  3876  3915
CYCLE   010662  1109  3598#
DATA   = 000017   851#  3507  3508
DATLNG  010472  2781  2929  3051  3489#
DDISP  = 177570   656#   937  1055
DISPLA  001142   937#  1055*  1063*  4129*  4225*
DISPRE  000174   872#  1063
DLADD   001254  1017#  1113*  1118  1121  1123  1126  1129  1132  1148  1180  3361  3739
DLADDR= 175610   758#  1017  1019  1020  1021  1022  1023
DLVEC   001256  1018#  1116*  2507  2648  2734  3073  3367  3745
DSWR   = 177570   655#   936  1054
EMTVEC= 000030   744#  1038*  1039*
ERRBIT= 100000   863#  2126  3315
ERRCHK  010166  2327*  2406*  2434  2767*  2810*  2829*  2870  2877  3278*  3310*  3321*  3333  3340
                3348#
ERRCNT  007556  3088*  3119  3127  3142#  3207  3218*
ERROR  = 100000   788#  2203  2277
ERRORF  004026  1710*  1757*  1784  1812#
ERRVEC= 000004   737#  1052  1053*  1064*  4186  4187*  4189*  4192*
```

```
EVENOD= 000040            853#
EXITFL  004030           1712*   1793*   1800*   1804    1813#
FALSE = 000000           3404#   3411    3610    3612
FLAG    010464           3411*   3438*   3443    3466    3478#
FRERR = 020000            790#   3318
CNS   = ****** U          871    1090    1097    1098    1099    3359    3365    3371    3716    3722    3731    3737    3743
                         4404    4405    4406    4407    4408    4410    4412    4413    4414
GTSWR = 104406           1085    4410#
HOLD    007622           3159*   3160*   3163    3176#
HOLDSC  010465           3423*   3428*   3434    3479#
HOWLON= 000000           3383#   3409
HT    = 000011            647#   3874    3915
I       001272           1024#   1157*   1178*   1181    1183    3512*   3515*   3517
ILLMEM= 000004            749#   1152    1188
INITFL  011104           1104*   3619    3622*   3693#
INTFLG  010660           1164*   1169    2503*   2553    2557    2579*   2595    2656*   2698    2705    3593*   3596#
INTSRV  010652           1153    2515    2649    3583#
IOTVEC= 000020            742#   1036*   1037*
LF    = 000012            648#   3909    3915
LOOP    002136           1107#   3377
MAINT = 000004            822#   1319    1332    1335    1349    1352    1366    1371    1605    1630    1839    1866    1913
                         1934    1993    2031    2060    2082    2104    2146    2187    2209    2233    2264    2282    2333
                         2432    2659    2703    2731    2770    2854    2868    2908    3103    3124    3281
MAINTJ= 040000            862#   1303    1826    1900    1973    2137    2308    2635    2755    3037    3258
MASK  = 000000           3490#   3504*   3520*   3522*   3527*
MAX     004672           2021    2025    2112#
MYTYPE  011126           3712#   4139
NEW     005644           2356*   2358*   2374*   2376*   2385    2393    2395    2413    2415    2470#
NOCONM  011122           3701#
NUMBER  007560           3117    3143#   3166    3224
NUMBR   010566           3506*   3507    3509*   3517    3530#
OK      011112           3696#
OLD     005650           2329*   2331*   2385    2393    2395    2413*   2415*   2471#
ORERR = 040000            789#   2182    2227    2259
PARITY= 000020            852#
PASS    004024           1705*   1787    1811#
PBAUDS= 004000            812#
PBAUDO= 010000            811#
PBAUD1= 020000            810#
PBAUD2= 040000            809#
PBAUD3= 100000            808#
PBR   = 000200            855#   2306
PERR  = 010000            791#
PIRQ  = 177772            654#
PIRQVE= 000240            748#
PR0   = 000000            671#   2535    2584    2665    2774    2922    3067
PR1   = 000040            672#
PR2   = 000100            673#
PR3   = 000140            674#
PR4   = 000200            675#
PR5   = 000240            676#
PR6   = 000300            677#
PR7   = 000340            678#   1154    1391    2516    2650    3078    3084
PS    = 177776            651#    652
PSW   = 177776            652#
PWRVEC= 000024            743#   1042*   1043*   3795*   3796*   3805*   3811*   3823*   3824*
```

```
PATES   005624          2347    2428    2444#
RBUF    001262          1020#   1123*   1124*   1950    2096    2178    2227    2259    2277    2344    2695    2784    2834
                        2932    2989    3194    3306    3318
RCSR    001260          1019#   1121*   1477    1490*   1493    1507*   1510    1524*   1529    1560    1625    1834    1882
                        1928    1953    2007    2047    2051    2054    2077    2099    2364    2662*   2679    2836*   2726*
                        2821    2919*   2971    3018*   3110*   3138*   3228*   3292
RCVRAC= 004000           772#   1625    2007    2047    2054
RCVRDO= 000200           776#   1560    1855    1882    1929    1953    2051    2077    2099    2364    2680    2822    2972
                        3293
RCVRIE= 000100           777#   1477    1490    1493    1507    1510    1524    1529    2662    2686    2726    3110    3138
                        3228
RDATA0= 000001           803#
RDATA1= 000002           802#
RDATA2= 000004           801#
RDATA3= 000010           800#
RDATA4= 000020           799#
RDATA5= 000040           798#
RDATA6= 000100           797#
RDATA7= 000200           796#
RDCHR = 104410          4028    4413#
RDLIN = 104411          4414#
RDRRUN= 000001           783#
REC     007626          3076    3186#
REG   = 000004          3383#   3407
REGSAV  010460          3407*   3420    3476#
RESVEC= 000010           738#
RHLD    007716          3198*   3199*   3203    3211    3237#
R0050   005624          2453#
R0070   005625          2454#
R0110   005626          2455#
R0135   005627          2456#
R0150   005630          2457#
R0200   005633          2460#
R0300   005631          2458#
R0600   005632          2459#
R10000  005643          2468#
R1800   005634          2461#
R2000   005635          2462#
R2400   005636          2463#
R3600   005637          2464#
R4800   005640          2465#
R5STAC  001334          1026#   1135
R7200   005641          2466#
R9600   005642          2467#
SB      007562          3144#   3211*
SET   = 177777           761#   1726    1757    1768    1784    1793    1800    1804    1853    1927    2010    2019    2523
                        2678    2801    2820    2949    2970    3291    3428
SETCLR= 000006          3383#   3434
SPECIA  010726          3629#   3786
STACK = 001100           642#   1034
START   001336           875    1027#   3828
STKLMT= 177774           653#
SWR     001140           936#   1032    1054*   1056    1062*   1069*   1083    3803    3816*   3925    3962*   4130    4137
                        4149    4153    4181    4195    4197    4203    4210
SWREG   000176           873#   1062    1083    3925    3938
SWO   = 000001           706#
```

```
SW00  = 000001            696#    706
SW01  = 000002            695#    705
SW02  = 000004            694#    704
SW03  = 000010            693#    703
SW04  = 000020            692#    702
SW05  = 000040            691#    701
SW06  = 000100            690#    700
SW07  = 000200            689#    699
SW08  = 000400            688#    698
SW09  = 001000            687#    697
SW1   = 000002            705#
SW10  = 002000            686#
SW11  = 004000            685#
SW12  = 010000            684#
SW13  = 020000            683#
SW14  = 040000            682#
SW15  = 100000            681#
SW2   = 000004            704#
SW3   = 000010            703#
SW4   = 000020            702#
SW5   = 000040            701#
SW6   = 000100            700#
SW7   = 000200            699#
SW8   = 000400            698#
SW9   = 001000            697#
TBITVE= 000014            739#
TBUF    001270           1023#   1132#   1133#   1720#   1748#   1846#   1921#   2003#   2158#   2168#   2250#   2353#   2672#
                         2816#   2966#   3163#   3287#
TCSR    001264           1021#   1126#   1127#   1231    1244#   1247    1261#   1264    1278#   1283    1319    1332#   1335
                         1349#   1352    1366#   1371    1401    1414#   1417    1431#   1434    1448#   1453    1630#   1663
                         1727    1753    1769    1839#   1866#   1913#   1934#   1993#   2031#   2060#   2082#   2104#   2146#
                         2187#   2209#   2233#   2264#   2282#   2333#   2432#   2524    2532#   2542#   2582#   2659#   2703#
                         2731#   2770#   2802    2854#   2868#   2908#   2950    3103#   3107#   3124#   3136#   3170#   3281#
                         3284#
TCSRHI  001266           1022#   1129#   1130#   2347#   2428#
TDATA0= 000001            845#
TDATA1= 000002            844#
TDATA2= 000004            843#
TDATA3= 000010            842#
TDATA4= 000020            841#
TDATA5= 000040            840#
TDATA6= 000100            839#
TDATA7= 000200            838#
TIME  = 000000           3539#   3550
TIMER   010314           1730    1772    1857    1931    2527    2682    2805    2824    2953    2974    3295    3382#
TIMSAV  010462           3409#   3445    3456#   3477#
TKVEC = 000060            746#
TPVEC = 000064            747#
TRAN    007564           3082    3151#
TRAPVE= 000034            745#   1040#   1041#
TRTVEC= 000014            740#
TRUE  = 000001           1305    1591    1693    1828    1902    1971    2128    2637    2757    3039    3269    3403#   3438
                         3443    3466    3662    3678
TST1    002252           1144#
TST10   003476           1613    1652#
TST11   003536           1688#
```

```
TST12   004032          1700    1810    1821#
TST13   004204          1833    1895#
TST14   004352          1907    1967#
TST15   004674          1978    1986    2036    2066    2087    2111    2121#
TST16   005260          2133    2141    2194    2217    2239    2272    2299#
TST17   005654          2313    2322    2442    2485#
TST2    002420          1210#
TST20   006176          2496    2631#
TST21   006502          2640    2751#
TST22   007024          2762    2859    2891#
TST23   007326          2903    2960    2983    3007    3032#
TST24   007722          3044    3140    3254#
TST25   010170          3265    3273    3347    3355#
TST3    002612          1222    1298#
TST4    003004          1310    1387#
TST5    003160          1469#
TST6    003322          1550#
TST7    003362          1583#
TYPDS = 104405          3374    3775    4408#
TYPE  = 104401          1077    1097    1098    1099    3357    3363    3369    3376    3714    3720    3729    3735    3741
                        3747    3773    3776    3825    3879    3936    3937    3940    3953    3964    3983    4032    4035
                        4039    4132    4140    4287    4356    4404#
TYPOC = 104402          3362    3368    3719    3734    3740    3746    3939    4405#
TYPON = 104404          4407#
TYPOS = 104403          3726    4406#
VECTOR  011110          3641*   3652*   3666*   3695#
WAIT    010572          2163    2173    2255    2548    2592    2691    3453    3538#
WAS     007563          3145#   3213*
WHICHB= 000002          3383#   3420
WRAP  = 020000           861#   1598    2895
XMIT E= 000100           818#   1401    1414    1417    1431    1434    1448    1453    2532    2542    2582    3107    3136
                        3170
XMITRD= 000200           817#   1663    1728    1753    1770    2525    2803    2951
ZZZ     007624          3174    3178#
ZZZZ    007720          3235    3238#
$APTHD  001000           896     902#
$ASTAT= ****** U        4088    4103
$ATYC   012704          4059    4061#
$ATY1   012660          4057#
$ATY3   012666          3864    4058#
$ATY4   012676          4060#   4145
$AUTOB  001134           933#   1087*   3933    4054
$BASE   001250           997#   3639
$BDADR  001122           928#
$BDDAT  001126           930#
$BELL   001164           948#   4132    4165
$BGNLE= 177777             1#
$CHARC  012100          3881*   3891*   3898    3907*   3912#
$CKSWR  012104          3925#   4412
$CMTAG  001100           916#   1029    1030    1038    1044    1045    1046
$CM3  = 000000           946#
$CNTLG  012631          3936    4049#
$CNTLU  012624          3953    4048#
$CPUOP  001222           971#
$CRLF   001171           950#   3376    3747    3880    3915    3964    4048    4140    4165
$DBLK   014022          4253    4287    4295#
```

```
$DEVCT  001204   962#  3637*  3689*
$DEVM   001252   998#  1094   3656
$DOAGN  011414  3769   3778   3784#
$DTBL   014012  4256   4291#
$ENDAD  011404   882   1075   3780#  4160
$ENDCT  011352  1044   3771#
$ENDMG  011423  3773   3788#
$ENULL  011420  3776   3787#
$ENV    001214   967#  1081   1217   1695   1982   2318   2492   3671   3859   4066   4090   4142
$ENVM   001215   968#  1067   3861   3866   4068
$EOP    011316  3627   3761#
$EOPCT  011344  1044*  3768#  3772
$ERFLG  001103   919#  4127*  4165   4170   4199   4201   4207*  4229
$ERMAX  001115   925#  1047*  4201   4224*  4229
$ERRFL= 000000  1104#  1105   1106#  1107   1109#  1113#  1114   1116#  1117   1118#  1119   1121#  1122
                1123#  1125   1126#  1128   1129#  1131   1132#  1134   1135#  1136   1148#  1149   1151#
                1152#  1153#  1154#  1155#  1156   1157#  1158   1161#  1162   1164#  1165   1178#  1179
                1180#  1182   1188#  1189#  1191#  1192#  1193   1221#  1222   1228#  1229   1242#  1243
                1244#  1245   1258#  1259   1261#  1262   1275#  1276   1278#  1279   1309#  1310   1316#
                1317   1330#  1331   1332#  1333   1346#  1347   1349#  1350   1363#  1364   1366#  1367
                1398#  1399   1412#  1413   1414#  1415   1428#  1429   1431#  1432   1445#  1446   1448#
                1449   1474#  1475   1488#  1489   1490#  1491   1504#  1505   1507#  1508   1521#  1522
                1524#  1525   1558#  1559   1612#  1613   1622#  1623   1630#  1631   1660#  1661   1699#
                1700   1705#  1706   1710#  1711   1712#  1713   1720#  1721   1725#  1748#  1749   1757#
                1758   1767#  1793#  1794   1800#  1801   1832#  1833   1839#  1840   1842#  1843   1846#
                1847   1852#  1866#  1867   1875#  1876   1906#  1907   1913#  1914   1915#  1916   1921#
                1922   1926#  1934#  1935   1950#  1951   1977#  1978   1985#  1986   1993#  1994   1995#
                1996   1997#  1998   2003#  2004   2010#  2011   2015#  2016   2031#  2032   2035#  2036
                2060#  2061   2065#  2066   2082#  2083   2086#  2087   2096#  2097   2104#  2105   2132#
                2133   2140#  2141   2146#  2147   2151#  2152   2158#  2159   2161#  2168#  2169   2171#
                2178#  2179   2187#  2188   2193#  2194   2201#  2202   2209#  2210   2216#  2217   2223#
                2224   2233#  2234   2238#  2239   2245#  2246   2250#  2251   2257*  2264#  2265   2271#
                2272   2282#  2283   2312#  2313   2321#  2322   2327#  2328   2329#  2330   2331#  2332
                2333#  2334   2336#  2337   2339   2343   2344#  2345   2347#  2348   2350#  2351   2353#
                2354   2356#  2357   2358#  2359   2368#  2369   2374#  2375   2376#  2377   2406#  2407
                2413#  2414   2415#  2416   2421#  2425   2426#  2427   2428#  2429   2432#  2433   2495#
                2496   2503#  2504   2507#  2508   2510#  2511   2513#  2514#  2515#  2516#  2517#  2518
                2519#  2520   2522#  2532#  2533   2542#  2543   2546#  2576#  2577   2579#  2580   2582#
                2583   2590#  2609#  2610#  2612#  2613#  2614   2647#  2648#  2649#  2650#  2651#  2652
                2654#  2655   2656#  2657   2659#  2660   2662#  2663   2672#  2673   2677#  2686#  2687
                2689#  2695#  2696   2703#  2704   2726#  2727   2731#  2732   2734#  2735   2738#  2739#
                2741#  2742#  2743   2761#  2762   2767#  2768   2770#  2771   2780#  2784#  2785   2788#
                2789   2791   2795   2800#  2810#  2811   2816#  2817   2819#  2829#  2830   2834#  2835
                2842#  2844   2845#  2846   2854#  2855   2858#  2859   2868#  2869   2902#  2903   2908#
                2909   2919#  2920   2928#  2932#  2933   2935#  2936   2938   2942   2948#  2959#  2960
                2966#  2967   2969#  2982#  2983   2989#  2990   2993#  2995   2996#  2997   3006#  3007
                3018#  3019   3043#  3044   3050#  3073#  3074   3076#  3077   3078#  3079   3082#  3083
                3084#  3085   3088#  3089   3091#  3092   3094#  3095   3097#  3098   3103#  3104   3107#
                3108   3110#  3111   3124#  3125   3136#  3137   3138#  3139   3156#  3157   3159#  3161
                3163#  3164   3170#  3171   3191#  3192   3194#  3196   3198#  3200   3211#  3212   3213#
                3214   3218#  3219   3228#  3229   3264#  3265   3272#  3273   3278#  3279   3281#  3282
                3284#  3285   3287#  3288   3290#  3310#  3311   3321#  3322   3407#  3408   3409#  3410
                3411#  3412   3423#  3424   3428#  3429   3438#  3439   3451#  3456#  3457   3469#  3470
                3473#  3475   3504#  3505   3506#  3510   3512#  3513   3515   3519   3520#  3521   3522#
                3523   3527#  3528   3550#  3551   3552#  3553   3555   3559   3560#  3561   3563   3567
                3593#  3594   3610#  3611   3612#  3613   3622#  3623   3627#  3631#  3632   3635#  3636
```

```
                      3637#   3638    3639#   3640    3641#   3642    3646#   3647    3648#   3649    3650#   3651    3652#
                      3653    3662#   3663    3664#   3665    3666#   3667    3678#   3679    3687#   3688    3689#   3690
$ERROR   013126       1038    4125#
$ERRPC   001116        926#   3733    4134*   4135*   4136    4165
$ERRTB   001254       1015#
$ERTTL   001112        923#   3373    3375*   4133*   4165
$ESCAP   001162        947#   1046*   4156    4158    4165    4223*
$ETABL   001214        966#
$ETEND   001254        908     999#
$FATAL   001176        959#   3724*   3725    4094*
$FFLG    013124       4057*   4060*   4088    4097*   4105#
$FILLC   001156        944#   3884    3915
$FILLS   001155        943#   3915
SF$AND=  000310          1#   1096    1171    1217    1233    1249    1266    1285    1305    1321    1337    1354    1373
                      1403    1419    1436    1455    1479    1495    1512    1531    1562    1593    1600    1607    1627
                      1665    1695    1755    1786    1789    1828    1884    1902    1955    1973    1984    2009    2027
                      2049    2053    2056    2079    2101    2128    2139    2184    2205    2229    2261    2279    2308
                      2320    2363    2366    2387    2395    2436    2494    2555    2559    2597    2637    2700    2707
                      2757    2851    2872    2879    2897    3002    3039    3129    3168    3205    3209    3226    3260
                      3271    3308    3317    3320    3335    3342    3422    3436    3468    3618    3621    3661    3673
                      3676
SF$BAD=  000401          1#   1096    1171    1220    1233    1249    1266    1285    1308    1321    1337    1354    1373
                      1403    1419    1436    1455    1479    1495    1512    1531    1562    1593    1600    1607    1627
                      1665    1698    1755    1786    1789    1831    1884    1905    1955    1976    1984    2009    2027
                      2049    2053    2056    2079    2101    2131    2139    2184    2205    2229    2261    2279    2311
                      2320    2363    2366    2387    2397    2436    2494    2555    2559    2597    2640    2700    2707
                      2760    2851    2872    2879    2900    3002    3042    3129    3168    3205    3209    3226    3263
                      3271    3308    3317    3320    3335    3342    3422    3436    3468    3618    3621    3661    3673
                      3676
SF$BLA=  000170          1#
SF$CAS=  000150          1#
SF$DEC=  000220          1#
SF$GOO=  000400          1#   1093    1094    1169    1215    1231    1247    1264    1283    1303    1319    1335    1352
                      1371    1401    1417    1434    1453    1477    1493    1510    1529    1560    1591    1598    1605
                      1625    1663    1693    1735    1753    1775    1784    1787    1826    1862    1882    1900    1937
                      1953    1971    1982    2007    2025    2046    2047    2051    2054    2077    2099    2126    2137
                      2182    2203    2227    2259    2277    2306    2318    2360    2361    2364    2385    2393    2434
                      2492    2553    2557    2595    2635    2698    2705    2755    2808    2827    2849    2870    2877
                      2895    2956    2977    3000    3037    3127    3166    3203    3207    3224    3258    3269    3298
                      3306    3315    3318    3333    3340    3420    3434    3466    3616    3619    3659    3671    3674
SF$IF    000110          1#   1169    1175    1215    1217    1224    1231    1237    1247    1253    1264    1270    1283
                      1289    1303    1305    1312    1319    1325    1335    1341    1352    1358    1371    1377    1401
                      1407    1417    1423    1434    1440    1453    1459    1477    1483    1493    1499    1510    1516
                      1529    1535    1560    1571    1591    1595    1598    1602    1605    1609    1615    1617    1619
                      1625    1642    1663    1674    1693    1695    1702    1735    1740    1753    1761    1775    1780
                      1782    1784    1787    1795    1797    1802    1826    1828    1835    1862    1870    1882    1888
                      1900    1902    1909    1937    1942    1953    1959    1971    1973    1980    1982    1988    2007
                      2012    2017    2025    2038    2051    2054    2068    2070    2077    2089    2099    2108    2126
                      2128    2135    2137    2143    2182    2196    2203    2219    2227    2241    2259    2274    2277
                      2287    2306    2308    2315    2318    2324    2364    2370    2378    2385    2389    2393    2395
                      2399    2408    2410    2434    2440    2492    2498    2553    2557    2563    2570    2572    2595
                      2601    2635    2637    2642    2698    2705    2712    2718    2720    2755    2757    2764    2808
                      2812    2827    2831    2849    2861    2870    2875    2877    2882    2895    2897    2905    2956
                      2962    2977    2985    3000    3009    3037    3039    3046    3127    3133    3166    3172    3203
                      3207    3215    3220    3224    3233    3258    3260    3267    3269    3275    3298    3303    3306
                      3312    3315    3318    3323    3325    3327    3333    3338    3340    3345    3420    3425    3430
```

```
                    3434    3440    3466    3471    3616    3619    3624    3633    3643    3654    3659    3671    3674
                    3680    3682    3684
$FSINC= 000210       1#     2336    2418    2788    2863    2935    3012    3512    3524    3552    3560    3568    3571
$FSLOO= 000200       1#     1707    1804    1807    3417    3443    3458
$FSNAM= 000160       1#
$FSNO = 000403       1#     1094    1169    1215    1217    1231    1247    1264    1283    1303    1319    1335    1352
                    1371    1401    1417    1434    1453    1477    1493    1510    1529    1560    1598    1605    1625
                    1663    1695    1753    1784    1787    1826    1882    1900    1953    1973    1982    2007    2025
                    2047    2051    2054    2077    2099    2126    2137    2182    2203    2227    2259    2277    2306
                    2308    2361    2364    2385    2393    2395    2434    2492    2553    2557    2595    2635    2698
                    2705    2755    2849    2870    2877    2895    2897    3000    3037    3127    3166    3203    3207
                    3224    3258    3260    3315    3318    3333    3340    3420    3616    3619    3659    3671    3674
$FSOP = 000320       1#     1096    1171    1217    1233    1249    1266    1285    1305    1321    1337    1354    1373
                    1403    1419    1436    1455    1479    1495    1512    1531    1562    1593    1600    1607    1627
                    1665    1695    1755    1786    1789    1828    1884    1902    1955    1973    1984    2009    2027
                    2049    2053    2056    2079    2101    2128    2139    2184    2205    2229    2261    2279    2308
                    2320    2363    2366    2387    2395    2397    2436    2494    2555    2559    2597    2637    2700
                    2707    2757    2851    2872    2879    2897    3002    3039    3129    3168    3205    3209    3226
                    3260    3271    3308    3317    3320    3335    3342    3422    3436    3468    3618    3621    3661
                    3673    3676
$FSRTN= 000300       1#     3383    3482    3490    3532    3539    3577    3599    3705    3713    3749
$FSSEL= 000140       1#
$FSUNT= 000130       1#     1159    1183    2005    2019    3115    3117    3614    3656
$FSWHI= 000120       1#     1093    1094    1101    1217    1305    1695    1828    1902    1973    2046    2047    2072
                    2128    2308    2360    2361    2381    2395    2637    2757    2897    3039    3260
$FSYES= 000402       1#     1094    1169    1215    1217    1231    1247    1264    1283    1303    1305    1319    1335
                    1352    1371    1401    1417    1434    1453    1477    1493    1510    1529    1560    1591    1598
                    1605    1625    1663    1693    1695    1753    1784    1787    1826    1828    1882    1900    1902
                    1953    1971    1973    1982    2007    2025    2047    2051    2054    2077    2099    2126    2128
                    2137    2182    2203    2227    2259    2277    2306    2308    2318    2361    2364    2385    2393
                    2395    2434    2492    2553    2557    2595    2635    2637    2698    2705    2755    2757    2849
                    2870    2877    2895    2897    3000    3037    3039    3127    3166    3203    3207    3224    3258
                    3260    3269    3306    3315    3318    3333    3340    3420    3434    3466    3616    3619    3659
                    3671    3674
SGDADR  001120       927#
SGDDAT  001124       929#
SGET42  011374      3777#
SGTSWR  012154      3937#    4410
$HD   = 000000       627
$HIBTS  001000       903#
$ICNT   001104       920#    4214#   4215    4217#   4228
$ FLEV= 177777         1#    1169#   1175#   1215#   1224#   1231#   1237#   1247#   1253#   1264#   1270#   1283#   1289#
                    1303#   1312#   1319#   1325#   1335#   1341#   1352#   1358#   1371#   1377#   1401#   1407#   1417#
                    1423#   1434#   1440#   1453#   1459#   1477#   1483#   1493#   1499#   1510#   1516#   1529#   1535#
                    1560#   1571#   1591#   1598#   1605#   1615#   1617#   1619#   1625#   1642#   1663#   1674#   1693#
                    1702#   1735#   1740#   1753#   1775#   1780#   1782#   1784#   1787#   1795#   1802#   1826#   1835#
                    1862#   1870#   1882#   1888#   1900#   1909#   1937#   1942#   1953#   1959#   1971#   1980#   1982#
                    1988#   2007#   2017#   2025#   2038#   2051#   2054#   2068#   2070#   2077#   2089#   2099#   2108#
                    2126#   2135#   2137#   2143#   2182#   2196#   2203#   2219#   2227#   2241#   2259#   2274#   2277#
                    2287#   230 #   2315#   2318#   2324#   2364#   2378#   2385#   2393#   2408#   2410#   2434#   2440#
                    2492#   24      2553#   2557#   2570#   2572#   2595#   2601#   2635#   2642#   2698#   2705#   2718#
                    2720#   275     2764#   2808#   2812#   2827#   2831#   2849#   2861#   2870#   2875#   2877#   2882#
                    2895#   2905    2956#   2962#   2977#   2985#   3000#   3009#   3037#   3046#   3127#   3133#   3166#
                    3172#   3203#   3207#   3215#   3220#   3224#   3233#   3258#   3267#   3269#   3275#   3298#   3303#
                    3306#   3315#   3318#   3323#   3325#   3327#   3333#   3338#   3340#   3345#   3420#   3430#   3434#
                    3440#   3466#   3471#   3616#   3619#   3633#   3654#   3659#   3671#   3674#   3680#   3682#   3684#
```

```
$ILLUP  011604          3795    3811    3830#
$INTAG  001135           934#   3965    4054
$ISK0 = 000001          1169#   1175    1215#   1224    1231#   1237    1247#   1253    1264#   1270    1283#   1289    1303#
                        1312    1319#   1325    1335#   1341    1352#   1358    1371#   1377    1401#   1407    1417#   1423
                        1434#   1440    1453#   1459    1477#   1483    1493#   1499    1510#   1516    1529#   1535    1560#
                        1571    1591#   1619    1625#   1642    1663#   1674    1693#   1702    1735#   1740    1753#   1782
                        1784#   1802    1826#   1835    1862#   1870    1882#   1888    1900#   1909    1937#   1942    1953#
                        1959    1971#   1980    1982#   1988    2007#   2017    2025#   2038    2051#   2070    2077#   2089
                        2099#   2108    2126#   2135    2137#   2143    2182#   2196    2203#   2219    2227#   2241    2259#
                        2274    2277#   2287    2306#   2315    2318#   2324    2364#   2378    2385#   2410    2434#   2440
                        2492#   2498    2553#   2572    2595#   2601    2635#   2642    2698#   2720    2755#   2764    2808#
                        2812    2827#   2831    2849#   2861    2870#   2875    2877#   2882    2895#   2905    2956#   2962
                        2977#   2985    3000#   3009    3037#   3046    3127#   3133    3166#   3172    3203#   3220    3224#
                        3233    3258#   3267    3269#   3275    3298#   3303    3306#   3327    3333#   3338    3340#   3345
                        3420#   3430    3434#   3440    3466#   3471    3616#   3654    3659#   3684
$ISK1 = 000001          1598#   1617    1775#   1780    1787#   1795    2054#   2068    2393#   2408    2557#   2570    2705#
                        2718    3207#   3215    3315#   3325    3619#   3633    3671#   3682
$ISK2 = 000001          1605#   1615    3318#   3323    3674#   3680
$ITEMB  001114           924#   3724    4136*   4144    4165
$LF     001172           951#   3915    4039    4048    4165
$LFLG   013123          4098*   4104#
$LOCTA= 177777             1#   1093    1094    1095    1096    1101    1102    1103    1159    1160    1170    1171    1175
                        1176    1184    1185    1216    1217    1218    1219    1220    1224    1225    1232    1233    1237
                        1238    1248    1249    1253    1254    1265    1266    1270    1271    1284    1285    1289    1290
                        1304    1305    1306    1307    1308    1312    1313    1320    1321    1325    1326    1336    1337
                        1341    1342    1353    1354    1358    1359    1372    1373    1377    1378    1402    1403    1407
                        1408    1418    1419    1423    1424    1435    1436    1440    1441    1454    1455    1459    1460
                        1478    1479    1483    1484    1494    1495    1499    1500    1511    1512    1516    1517    1530
                        1531    1535    1536    1561    1562    1571    1572    1592    1593    1595    1596    1597    1599
                        1600    1602    1603    1604    1606    1607    1609    1610    1611    1615    1616    1617    1618
                        1619    1620    1626    1627    1642    1643    1664    1665    1674    1675    1694    1695    1696
                        1697    1698    1702    1703    1707    1708    1735    1736    1740    1741    1754    1755    1761
                        1762    1763    1775    1776    1780    1781    1782    1783    1785    1786    1788    1789    1795
                        1796    1797    1798    1799    1802    1803    1805    1806    1807    1808    1809    1827    1828
                        1829    1830    1831    1835    1836    1862    1863    1870    1871    1883    1884    1888    1889
                        1901    1902    1903    1904    1905    1909    1910    1937    1938    1942    1943    1954    1955
                        1959    1960    1972    1973    1974    1975    1976    1980    1981    1983    1984    1988    1989
                        2005    2006    2008    2009    2012    2013    2014    2017    2018    2020    2021    2022    2023
                        2024    2026    2027    2038    2039    2046    2047    2048    2049    2052    2053    2055    2056
                        2068    2069    2070    2071    2072    2073    2074    2078    2079    2089    2090    2100    2101
                        2108    2109    2127    2128    2129    2130    2131    2135    2136    2138    2139    2143    2144
                        2183    2184    2196    2197    2204    2205    2219    2220    2228    2229    2241    2242    2260
                        2261    2274    2275    2278    2279    2287    2288    2307    2308    2309    2310    2311    2315
                        2316    2319    2320    2324    2325    2337    2338    2339    2340    2341    2342    2343    2360
                        2361    2362    2363    2365    2366    2370    2371    2372    2378    2379    2381    2382    2383
                        2386    2387    2389    2390    2391    2394    2395    2396    2397    2399    2400    2401    2408
                        2409    2410    2411    2418    2419    2420    2435    2436    2440    2441    2493    2494    2498
                        2499    2554    2555    2558    2559    2563    2564    2565    2570    2571    2572    2573    2596
                        2597    2601    2602    2636    2637    2638    2639    2640    2642    2643    2699    2700    2706
                        2707    2712    2713    2714    2718    2719    2720    2721    2756    2757    2758    2759    2760
                        2764    2765    2789    2790    2791    2792    2793    2794    2795    2808    2809    2812    2813
                        2827    2828    2831    2832    2850    2851    2861    2862    2863    2864    2865    2871    2872
                        2875    2876    2878    2879    2882    2883    2896    2897    2898    2899    2900    2905    2906
                        2936    2937    2938    2939    2940    2941    2942    2956    2957    2962    2963    2977    2978
                        2985    2986    3001    3002    3009    3010    3012    3013    3014    3038    3039    3040    3041
                        3042    3046    3047    3115    3116    3118    3119    3120    3121    3122    3128    3129    3133
```

```
                    3134    3167    3168    3172    3173    3204    3205    3208    3209    3215    3216    3220    3221
                    3225    3226    3233    3234    3259    3260    3261    3262    3263    3267    3268    3270    3271
                    3275    3276    3298    3299    3303    3304    3307    3308    3312    3313    3314    3316    3317
                    3319    3320    3323    3324    3325    3326    3327    3328    3334    3335    3338    3339    3341
                    3342    3345    3346    3382    3417    3418    3421    3422    3425    3426    3427    3430    3431
                    3435    3436    3440    3441    3444    3445    3446    3447    3458    3459    3460    3467    3468
                    3469    3470    3471    3472    3474    3475    3482    3483    3484    3485    3489    3513    3514
                    3515    3516    3517    3518    3519    3524    3525    3526    3529    3530    3532    3533    3534
                    3538    3553    3554    3555    3556    3557    3558    3559    3561    3562    3563    3564    3565
                    3566    3567    3568    3569    3570    3571    3572    3573    3577    3578    3579    3598    3614
                    3615    3617    3618    3620    3621    3624    3625    3626    3633    3634    3643    3644    3645
                    3654    3655    3657    3658    3660    3661    3672    3673    3675    3676    3680    3681    3682
                    3683    3684    3685    3691    3692    3705    3706    3707    3712    3749    3750    3751
SLPADR  001106       921#    1048#   4205#   4221#   4226    4228
SLPERR  001110       922#    1049#   1161#   1228#   1242#   1258#   1275#   1316#   1330#   1346#   1363#   1398#   1412#
                    1428#   1445#   1474#   1488#   1504#   1521#   1558#   1622#   1660#   1842#   1875#   1915#   2151#
                    2201#   2223#   2245#   2519#   2576#   2654#   4155    4205    4222#   4228
SLSTCN= 177777         1#   1093    1094    1096    1101    1102    1159    1160    1169    1171    1175    1183    1215
                    1220    1224    1231    1233    1237    1247    1249    1253    1264    1266    1270    1283    1285
                    1289    1303    1308    1312    1319    1321    1325    1335    1337    1341    1352    1354    1358
                    1371    1373    1377    1401    1403    1407    1417    1419    1423    1434    1436    1440    1453
                    1455    1459    1477    1479    1483    1493    1495    1499    1510    1512    1516    1529    1531
                    1535    1560    1562    1571    1591    1593    1596    1597    1598    1600    1603    1604    1605
                    1607    1610    1611    1615    1617    1619    1625    1627    1642    1663    1665    1674    1693
                    1698    1702    1707    1708    1735    1736    1740    1753    1755    1762    1763    1775    1776
                    1780    1782    1784    1786    1787    1789    1795    1798    1799    1802    1807    1808    1826
                    1831    1835    1862    1863    1870    1882    1884    1888    1900    1905    1909    1937    1938
                    1942    1953    1955    1959    1971    1976    1980    1982    1984    1988    2005    2006    2007
                    2009    2013    2014    2017    2019    2024    2025    2027    2038    2046    2047    2049    2051
                    2053    2054    2056    2068    2070    2072    2073    2077    2079    2089    2099    2101    2108
                    2126    2131    2135    2137    2139    2143    2182    2184    2196    2203    2205    2219    2227
                    2229    2241    2259    2261    2274    2277    2279    2287    2306    2311    2315    2318    2320
                    2324    2336    2338    2339    2340    2343    2360    2361    2363    2364    2366    2371    2372
                    2378    2381    2382    2385    2387    2390    2391    2393    2397    2400    2401    2408    2410
                    2418    2419    2434    2436    2440    2492    2494    2498    2553    2555    2557    2559    2564
                    2565    2570    2572    2595    2597    2601    2635    2640    2642    2698    2700    2705    2707
                    2713    2714    2718    2720    2755    2760    2764    2788    2790    2791    2792    2795    2808
                    2809    2812    2827    2828    2831    2849    2851    2861    2863    2864    2870    2872    2875
                    2877    2879    2882    2895    2900    2905    2935    2937    2938    2939    2942    2956    2957
                    2962    2977    2978    2985    3000    3002    3009    3012    3013    3037    3042    3046    3115
                    3116    3117    3122    3127    3129    3133    3166    3168    3172    3203    3205    3207    3209
                    3215    3220    3224    3226    3233    3258    3263    3267    3269    3271    3275    3298    3299
                    3303    3306    3308    3313    3314    3315    3317    3318    3320    3323    3325    3327    3333
                    3335    3338    3340    3342    3345    3383    3417    3418    3420    3422    3426    3427    3430
                    3434    3436    3440    3458    3459    3466    3468    3471    3482    3490    3512    3514    3515
                    3516    3519    3524    3525    3532    3539    3552    3554    3555    3556    3559    3560    3562
                    3563    3564    3567    3568    3569    3571    3572    3577    3599    3614    3615    3616    3618
                    3619    3621    3625    3626    3633    3644    3645    3654    3656    3659    3661    3671    3673
                    3674    3676    3680    3682    3684    3705    3713    3749
SLSTIN= 000000       1#   1094    1095    1096    1101    1102    1104    1:05    1106    1107    1109    1110    1113
                    1114    1116    1117    1118    1119    1121    1122    1123    1124    1125    1126    1127    1128
                    1129    1130    1131    1132    1133    1134    1135    1136    1148    1149    1151    1152    1153
                    1154    1155    1156    1157    1158    1161    1162    1164    1165    1169    1170    1171    1178
                    1179    1180    1181    1182    1183    1184    1185    1188    1189    1190    1191    1192    1193
                    1215    1216    1217    1218    1219    1221    1222    1228    1229    1231    1232    1233    1242
                    1243    1244    1245    1247    1248    1249    1258    1259    1261    1262    1264    1265    1266
```

1275	1276	1278	1279	1283	1284	1285	1303	1304	1305	1306	1307	1309
1310	1316	1317	1319	1320	1321	1330	1331	1332	1333	1335	1336	1337
1346	1347	1349	1350	1352	1353	1354	1363	1364	1366	1367	1371	1372
1373	1398	1399	1401	1402	1403	1412	1413	1414	1415	1417	1418	1419
1428	1429	1431	1432	1434	1435	1436	1445	1446	1448	1449	1453	1454
1455	1474	1475	1477	1478	1479	1488	1489	1490	1491	1493	1494	1495
1504	1505	1507	1508	1510	1511	1512	1521	1522	1524	1525	1529	1530
1531	1558	1559	1560	1561	1562	1591	1592	1593	1595	1596	1598	1599
1600	1602	1603	1605	1606	1607	1609	1610	1612	1613	1622	1623	1625
1626	1627	1630	1631	1660	1661	1663	1664	1665	1693	1694	1695	1696
1697	1699	1700	1705	1706	1710	1711	1712	1713	1720	1721	1725	1726
1727	1728	1729	1730	1731	1732	1735	1736	1748	1749	1753	1754	1755
1757	1758	1761	1762	1767	1768	1769	1770	1771	1772	1773	1774	1775
1776	1784	1785	1786	1787	1788	1789	1793	1794	1797	1798	1800	1801
1804	1805	1806	1807	1808	1826	1827	1828	1829	1830	1832	1833	1839
1840	1842	1843	1846	1847	1852	1853	1854	1855	1856	1857	1858	1859
1862	1863	1866	1867	1875	1876	1882	1883	1884	1900	1901	1902	1903
1904	1906	1907	1913	1914	1915	1916	1921	1922	1926	1927	1928	1929
1930	1931	1932	1933	1934	1935	1937	1938	1950	1951	1953	1954	1955
1971	1972	1973	1974	1975	1977	1978	1982	1983	1984	1985	1986	1993
1994	1995	1996	1997	1998	2003	2004	2007	2008	2009	2010	2011	2012
2013	2015	2016	2019	2020	2021	2022	2023	2025	2026	2027	2031	2032
2035	2036	2047	2048	2049	2051	2052	2053	2054	2055	2056	2060	2061
2065	2066	2072	2073	2077	2078	2079	2082	2083	2086	2087	2096	2097
2099	2100	2101	2104	2105	2126	2127	2128	2129	2130	2132	2133	2137
2138	2139	2140	2141	2146	2147	2151	2152	2158	2159	2161	2162	2163
2164	2165	2168	2169	2171	2172	2173	2174	2175	2178	2179	2182	2183
2184	2187	2188	2193	2194	2201	2202	2203	2204	2205	2209	2210	2216
2217	2223	2224	2227	2228	2229	2233	2234	2238	2239	2245	2246	2250
2251	2253	2254	2255	2256	2257	2259	2260	2261	2264	2265	2271	2272
2277	2278	2279	2282	2283	2306	2307	2308	2309	2310	2312	2313	2318
2319	2320	2321	2322	2327	2328	2329	2330	2331	2332	2333	2334	2336
2337	2338	2339	2340	2341	2342	2343	2344	2345	2347	2348	2350	2351
2353	2354	2356	2357	2358	2359	2361	2362	2363	2364	2365	2366	2368
2369	2370	2371	2374	2375	2376	2377	2381	2382	2385	2386	2387	2389
2390	2393	2394	2395	2396	2397	2399	2400	2406	2407	2413	2414	2415
2416	2418	2419	2421	2422	2423	2424	2425	2426	2427	2428	2429	2432
2433	2434	2435	2436	2492	2493	2494	2495	2496	2503	2504	2507	2508
2510	2511	2513	2514	2515	2516	2517	2518	2519	2520	2522	2523	2524
2525	2526	2527	2528	2529	2532	2533	2542	2543	2546	2547	2548	2549
2550	2553	2554	2555	2557	2558	2559	2563	2564	2576	2577	2579	2580
2582	2583	2590	2591	2592	2593	2594	2595	2596	2597	2609	2610	2611
2612	2613	2614	2635	2636	2637	2638	2639	2647	2648	2649	2650	2651
2652	2654	2655	2656	2657	2659	2660	2662	2663	2672	2673	2677	2678
2679	2680	2681	2682	2683	2684	2686	2687	2689	2690	2691	2692	2693
2695	2696	2698	2699	2700	2703	2704	2705	2706	2707	2712	2713	2726
2727	2731	2732	2734	2735	2738	2739	2740	2741	2742	2743	2755	2756
2757	2758	2759	2761	2762	2767	2768	2770	2771	2780	2781	2782	2783
2784	2785	2788	2789	2790	2791	2792	2793	2794	2795	2800	2801	2802
2803	2804	2805	2806	2807	2808	2809	2810	2811	2816	2817	2819	2820
2821	2822	2823	2824	2825	2826	2827	2828	2829	2830	2834	2835	2842
2843	2844	2845	2846	2849	2850	2851	2854	2855	2858	2859	2863	2864
2868	2869	2870	2871	2872	2877	2878	2879	2895	2896	2897	2898	2899
2902	2903	2908	2909	2919	2920	2928	2929	2930	2931	2932	2933	2935
2936	2937	2938	2939	2940	2941	2942	2948	2949	2950	2951	2952	2953
2954	2955	2956	2957	2959	2960	2966	2967	2969	2970	2971	2972	2973

CROSS REFERENCE TABLE -- USER SYMBOLS

	2974	2975	2976	2977	2978	2982	2983	2989	2990	2993	2994	2995	2996
	2997	3000	3001	3002	3006	3007	3012	3013	3018	3019	3037	3038	3039
	3040	3041	3043	3044	3050	3051	3052	3053	3073	3074	3076	3077	3078
	3079	3082	3083	3084	3085	3088	3089	3091	3092	3094	3095	3097	3098
	3103	3104	3107	3108	3110	3111	3117	3118	3119	3120	3121	3124	3125
	3127	3128	3129	3136	3137	3138	3139	3156	3157	3159	3160	3161	3163
	3164	3166	3167	3168	3170	3171	3191	3192	3194	3195	3196	3198	3199
	3200	3203	3204	3205	3207	3208	3209	3211	3212	3213	3214	3218	3219
	3224	3225	3226	3228	3229	3258	3259	3260	3261	3262	3264	3265	3269
	3270	3271	3272	3273	3278	3279	3281	3282	3284	3285	3287	3288	3290
	3291	3292	3293	3294	3295	3296	3297	3298	3299	3306	3307	3308	3310
	3311	3312	3313	3315	3316	3317	3318	3319	3320	3321	3322	3333	3334
	3335	3340	3341	3342	3407	3408	3409	3410	3411	3412	3420	3421	3422
	3423	3424	3425	3426	3428	3429	3434	3435	3436	3438	3439	3443	3444
	3445	3446	3447	3451	3452	3453	3454	3455	3456	3457	3458	3459	3466
	3467	3468	3469	3470	3473	3474	3475	3483	3484	3485	3486	3504	3505
	3506	3507	3508	3509	3510	3512	3513	3514	3515	3516	3517	3518	3519
	3520	3521	3522	3523	3524	3525	3527	3528	3529	3530	3534	3535	3550
	3551	3552	3553	3554	3555	3556	3557	3558	3559	3560	3561	3562	3563
	3564	3565	3566	3567	3568	3569	3571	3572	3579	3580	3593	3594	3610
	3611	3612	3613	3616	3617	3618	3619	3620	3621	3622	3623	3624	3625
	3627	3628	3631	3632	3635	3636	3637	3638	3639	3640	3641	3642	3643
	3644	3646	3647	3648	3649	3650	3651	3652	3653	3656	3657	3658	3659
	3660	3661	3662	3663	3664	3665	3666	3667	3671	3672	3673	3674	3675
	3676	3678	3679	3687	3688	3689	3690	3691	3692	3707	3708	3751	3752
$LSTST= 177777	1#	1093	1094	1096	1101	1102	1159	1160	1169	1171	1175	1183	1215
	1220	1224	1231	1233	1237	1247	1249	1253	1264	1266	1270	1283	1285
	1289	1303	1308	1312	1319	1321	1325	1335	1337	1341	1352	1354	1358
	1371	1373	1377	1401	1403	1407	1417	1419	1423	1434	1436	1440	1453
	1455	1459	1477	1479	1483	1493	1495	1499	1510	1512	1516	1529	1531
	1535	1560	1562	1571	1591	1593	1595	1596	1597	1598	1600	1602	1603
	1604	1605	1607	1609	1610	1611	1615	1617	1619	1625	1627	1642	1663
	1665	1674	1693	1698	1702	1707	1708	1735	1736	1740	1753	1755	1761
	1762	1763	1775	1776	1780	1782	1784	1786	1787	1789	1795	1797	1798
	1799	1802	1804	1807	1808	1826	1831	1835	1862	1863	1870	1882	1884
	1888	1900	1905	1909	1937	1938	1942	1953	1955	1959	1971	1976	1980
	1982	1984	1988	2005	2006	2007	2009	2012	2013	2014	2017	2019	2025
	2027	2038	2046	2047	2049	2051	2053	2054	2056	2068	2070	2072	2073
	2077	2079	2089	2099	2101	2108	2126	2131	2135	2137	2139	2143	2182
	2184	2196	2203	2205	2219	2227	2229	2241	2259	2261	2274	2277	2279
	2287	2306	2311	2315	2318	2320	2324	2336	2338	2339	2340	2343	2360
	2361	2363	2364	2366	2370	2371	2372	2378	2381	2382	2385	2387	2389
	2390	2391	2393	2397	2399	2400	2401	2408	2410	2418	2419	2434	2436
	2440	2492	2494	2[illegible]98	2553	2555	2557	2559	2563	2564	2565	2570	2572
	2595	2597	2601	2635	2640	2642	2698	2700	2705	2707	2712	2713	2714
	2718	2720	2755	2760	2764	2788	2790	2791	2792	2795	2808	2809	2812
	2827	2828	2831	2849	2851	2861	2863	2864	2870	2872	2875	2877	2879
	2882	2895	2900	2905	2935	2937	2938	2939	2942	2956	2957	2962	2977
	2978	2985	3000	3002	3009	3012	3013	3037	3042	3046	3115	3116	3117
	3127	3129	3133	3166	3168	3172	3203	3205	3207	3209	3215	3220	3224
	3226	3233	3258	3263	3267	3269	3271	3275	3298	3299	3303	3306	3308
	3312	3313	3314	3315	3317	3318	3320	3323	3325	3327	3333	3335	3338
	3340	3342	3345	3383	3417	3418	3420	3422	3425	3426	3427	3430	3434
	3436	3440	3443	3458	3459	3466	3468	3471	3482	3490	3512	3514	3515
	3516	3519	3524	3525	3532	3539	3552	3554	3555	3556	3559	3560	3562
	3563	3564	3567	3568	3569	3571	3572	3577	3599	3614	3615	3616	3618

```
                    3619    3621    3624    3625    3626    3633    3643    3644    3645    3654    3656    3659    3661
                    3671    3673    3674    3676    3680    3682    3684    3705    3713    3749
$LSTTA= 000000        1#    1093    1094    1102    1103    1159    1160    1175    1176    1219    1220    1224    1225
                    1237    1238    1253    1254    1270    1271    1289    1290    1307    1308    1312    1313    1325
                    1326    1341    1342    1358    1359    1377    1378    1407    1408    1423    1424    1440    1441
                    1459    1460    1483    1484    1499    1500    1516    1517    1535    1536    1571    1572    1596
                    1597    1603    1604    1610    1611    1615    1616    1617    1618    1619    1620    1642    1643
                    1674    1675    1697    1698    1702    1703    1707    1708    1740    1741    1762    1763    1780
                    1781    1782    1783    1795    1796    1798    1799    1802    1803    1808    1809    1830    1831
                    1835    1836    1870    1871    1888    1889    1904    1905    1909    1910    1942    1943    1959
                    1960    1975    1976    1980    1981    1988    1989    2005    2006    2013    2014    2017    2018
                    2023    2024    2038    2039    2046    2047    2068    2069    2070    2071    2073    2074    2089
                    2090    2108    2109    2130    2131    2135    2136    2143    2144    2196    2197    2219    2220
                    2241    2242    2274    2275    2287    2288    2310    2311    2315    2316    2324    2325    2338
                    2339    2340    2341    2360    2361    2371    2372    2378    2379    2382    2383    2390    2391
                    2400    2401    2408    2409    2410    2411    2419    2420    2440    2441    2498    2499    2564
                    2565    2570    2571    2572    2573    2601    2602    2639    2640    2642    2643    2713    2714
                    2718    2719    2720    2721    2759    2760    2764    2765    2790    2791    2792    2793    2812
                    2813    2831    2832    2861    2862    2864    2865    2875    2876    2882    2883    2899    2900
                    2905    2906    2937    2938    2939    2940    2962    2963    2985    2986    3009    3010    3013
                    3014    3041    3042    3046    3047    3115    3116    3121    3122    3133    3134    3172    3173
                    3215    3216    3220    3221    3233    3234    3262    3263    3267    3268    3275    3276    3303
                    3304    3313    3314    3323    3324    3325    3326    3327    3328    3338    3339    3345    3346
                    3382    3383    3417    3418    3426    3427    3430    3431    3440    3441    3459    3460    3471
                    3472    3482    3483    3484    3485    3489    3490    3514    3515    3516    3517    3525    3526
                    3532    3533    3534    3538    3539    3554    3555    3556    3557    3562    3563    3564    3565
                    3569    3570    3572    3573    3577    3578    3579    3598    3599    3614    3615    3625    3626
                    3633    3634    3644    3645    3654    3655    3680    3681    3682    3683    3684    3685    3705
                    3706    3707    3712    3713    3749    3750    3751
$MADR1  001226       984#
$MADR2  001232       988#
$MADR3  001236       991#
$MADR4  001242       994#
$MAIL   001174       904     908     957#    1066    1081    1146    1212    1300    1389    1471    1552    1585    1654
                    1690    1823    1897    1969    2123    2301    2487    2633    2753    2893    3034    3256    3859
                    4142    4220
$MAMS1  001224       978#
$MAMS2  001230       986#
$MAMS3  001234       989#
$MAMS4  001240       992#
$MBADR  001002       904#
$MCALL= 000000         1#
$MFLG   013122      4058#   4064    4099#   4103#
$MNEW   012647      3940    4052#
$MSGAD  001210       964#   4074#   4077
$MSGLG  001212       965#   4079#
$MSGTY  001174       958#   4072    4080#   4092    4096#
$MSWR   012636      3937    4050#
$MTYP1  001225       979#
$MTYP2  001231       987#
$MTYP3  001235       990#
$MTYP4  001241       993#
$MXCNT  013604      4218    4228#
$NESTL= 177777         1#   1093#   1101#   1159#   1169#   1175#   1183#   1215#   1224#   1231#   1237#   1247#   1253#
                    1264#   1270#   1283#   1289#   1303#   1312#   1319#   1325#   1335#   1341#   1352#   1358#   1371#
                    1377#   1401#   1407#   1417#   1423#   1434#   1440#   1453#   1459#   1477#   1483#   1493#   1499#
```

```
                    1510#  1516#  1529#  1535#  1560#  1571#  1591#  1595   1598#  1602   1605#  1609   1615#
                    1617#  1619#  1625#  1642#  1663#  1674#  1693#  1702#  1707#  1735#  1740#  1753#  1761
                    1775#  1780#  1782#  1784#  1787#  1795#  1797   1802#  1804   1807#  1826#  1835#  1862#
                    1870#  1882#  1888#  1900#  1909#  1937#  1942#  1953#  1959#  1971#  1980#  1982#  1988#
                    2005#  2007#  2012   2017#  2019#  2025#  2038#  2046#  2051#  2054#  2068#  2070#  2072#
                    2077#  2089#  2099#  2108#  2126#  2135#  2137#  2143#  2182#  2196#  2203#  2219#  2227#
                    2241#  2259#  2274#  2277#  2287#  2306#  2315#  2318#  2324#  2336#  2360#  2364#  2370
                    2378#  2381#  2385#  2389   2393#  2399   2408#  2410#  2418#  2434#  2440#  2492#  2498#
                    2553#  2557#  2563   2570#  2572#  2595#  2601#  2635#  2642#  2698#  2705#  2712   2718#
                    2720#  2755#  2764#  2788#  2808#  2812#  2827#  2831#  2849#  2861#  2863#  2870#  2875#
                    2877#  2882#  2895#  2905#  2935#  2956#  2962#  2977#  2985#  3000#  3009#  3012#  3037#
                    3046#  3115#  3117#  3127#  3133#  3166#  3172#  3203#  3207#  3215#  3220#  3224#  3233#
                    3258#  3267#  3269#  3275#  3298#  3303#  3306#  3312   3315#  3318#  3323#  3325#  3327#
                    3333#  3338#  3340#  3345#  3383#  3417#  3420#  3425   3430#  3434#  3440#  3443   3458#
                    3466#  3471#  3482#  3490#  3512#  3524#  3532#  3539#  3552#  3560#  3568#  3571#  3577#
                    3599#  3614#  3616#  3619#  3624   3633#  3643   3654#  3656#  3659#  3671#  3674#  3680#
                    3682#  3684#  3705#  3713#  3749#
$NSKO = 000300       1093#  1101   1159#  1183   1215#  1224   1231#  1237   1247#  1253   1264#  1270   1283#
                    1289   1303#  1312   1319#  1325   1335#  1341   1352#  1358   1371#  1377   1401#  1407
                    1417#  1423   1434#  1440   1453#  1459   1477#  1483   1493#  1499   1510#  1516   1529#
                    1535   1560#  1571   1591#  1595   1619   1625#  1642   1663#  1674   1693#  1702   1707#
                    1804   1807   1826#  1835   1862#  1870   1882#  1888   1900#  1909   1937#  1942   1953#
                    1959   1971#  1980   1982#  1988   2005#  2019   2025#  2038   2046#  2072   2077#  2089
                    2099#  2108   2126#  2135   2137#  2143   2182#  2196   2203#  2219   2227#  2241   2259#
                    2274   2277#  2287   2306#  2315   2318#  2324   2336#  2418   2434#  2440   2492#  2498
                    2553#  2572   2595#  2601   2635#  2642   2698#  2720   2755#  2764   2788#  2863   2870#
                    2875   2877#  2882   2895#  2905   2935#  3012   3037#  3046   3115#  3117   3127#  3133
                    3166#  3172   3203#  3220   3224#  3233   3258#  3267   3269#  3275   3298#  3303   3306#
                    3312   3327   3333#  3338   3340#  3345   3383#  3482   3490#  3532   3539#  3577   3599#
                    3705   3713#  3749
$NSK1 = 000110       1169#  1175   1598#  1602   1617   1735#  1740   1753#  1761   1782   1784#  1797   1802
                    2007#  2012   2017   2051#  2070   2360#  2381   2385#  2389   2410   2557#  2563   2570
                    2705#  2712   2718   2808#  2812   2827#  2831   2849#  2861   2956#  2962   2977#  2985
                    3000#  3009   3207#  3215   3315#  3325   3417#  3443   3458   3466#  3471   3512#  3524
                    3552#  3571   3614#  3656   3659#  3684
$NSK2 = 000110       1605#  1609   1615   1775#  1780   1787#  1795   2054#  2068   2364#  2370   2378   2393#
                    2399   2408   3318#  3323   3420#  3425   3430   3434#  3440   3560#  3568   3616#  3643
                    3654   3671#  3682
$NSK3 = 000110       3619#  3624   3633   3674#  3680
$NULL  001154         942#  3886   3915
$NWTST= 000001       1138#  1140   1202#  1204   1295#  1384#  1466#  1468   1547#  1580#  1649#  1683#  1685
                    1816#  1818   1892#  1962#  1964   2118#  2293#  2295   2477#  2479   2625#  2627   2747#
                    2749   2888#  3028#  3030   3247#  3249   3352#
$OCNT  014254        4328*  4357*  4370#
$OMODE 014256        4323*  4327*  4332   4335*  4346*  4372#
$OVER  013570        4182   4198   4206   4216   4225#
$PASS  001202         961#  1066*  3674   3765*  3766*  3774   3787   4212   4229
$PASTM 001006         906#
$POWER 011612        3826   3833#
$PWRAD 011600        3828#
$PWRDN 011440        1042   3795#  3823
$PWRMG 011574        3826#
$PWRUP 011512        3805   3811#
$QUES  001170         949#  3915   3983   4032   4048   4165
$RDCHR 012366        3996#  4413
$RDDEC= ****** U     4415
```

```
$RDLIN  012506          4024#   4414
$RDOCT= ****** U        4415
$RDSZ = 000010          4017#
$RTNAD  011416          3786#
$R29  = ****** U        4415
$SAVLE= 177777             1#   1101#   1102#   1807#   1808#   2072#   2073#   2339#   2343#   2381#   2382#   2791#   2795#
                        2938#   2942#   3458#   3459#   3515#   3519#   3555#   3559#   3563#   3567#
$SAVRE= ****** U        4415
$SAVR6  011610          3804*   3812    3813*   3814*   3832#
$SCOPE  013326          1036    4179#
$SETUP= 000137          1027#   1035    1036    1038    1040    1042    1044    1045    1046    1048    1075    1078    3763
                        3920    4054    4126    4152    4160    4180
$SSKO = 000234          1101#   1102    1807#   1808    2072#   2073    2339#   2343    2381#   2382    2791#   2795    2938#
                        2942    3458#   3459    3515#   3519    3555#   3559    3563#   3567
$STUP = 177777          1027#
$SVLAD  013534          4190    4219#
$SVPC = 000204           880#    885
$SWR  = 167400             1#    627     631     632     633     634     635     636     637     946     947     948     1045
                        1046    1048    1049    1145    1211    1299    1388    1470    1551    1584    1653    1689    1822
                        1896    1968    2122    2300    2486    2632    2752    2892    3033    3255    3356    3758    3764
                        3779    3785    3787    3829    4117    4118    4119    4120    4121    4130    4137    4149    4153
                        4165    4171    4172    4173    4174    4175    4181    4193    4195    4196    4199    4200    4201
                        4208    4209    4210    4222    4225    4228
$SWREG  001216           969#   1069
$SWRMK= 000000           637     638    4175    4176    4197
$TAGLE= 177777             1#   1094#   1096#   1101#   1160#   1171#   1175#   1183#   1220#   1224#   1233#   1237#   1249#
                        1253#   1266#   1270#   1285#   1289#   1308#   1312#   1321#   1325#   1337#   1341#   1354#   1358#
                        1373#   1377#   1403#   1407#   1419#   1423#   1436#   1440#   1455#   1459#   1479#   1483#   1495#
                        1499#   1512#   1516#   1531#   1535#   1562#   1571#   1593#   1596#   1597#   1600#   1603#   1604#
                        1607#   1610#   1611#   1615#   1617#   1619#   1627#   1642#   1665#   1674#   1698#   1702#   1708#
                        1736#   1740#   1755#   1762#   1763#   1776#   1780#   1782#   1786#   1789#   1795#   1798#   1799#
                        1802#   1804    1807#   1831#   1835#   1863#   1870#   1884#   1888#   1905#   1909#   1938#   1942#
                        1955#   1959#   1976#   1980#   1984#   1988#   2006#   2009#   2013#   2014#   2017#   2019#   2027#
                        2038#   2047#   2049#   2053#   2056#   2068#   2070#   2072#   2079#   2089#   2101#   2108#   2131#
                        2135#   2139#   2143#   2184#   2196#   2205#   2219#   2229#   2241#   2261#   2274#   2279#   2287#
                        2311#   2315#   2320#   2324#   2338#   2340#   2343#   2361#   2363#   2366#   2371#   2372#   2378#
                        2381#   2387#   2390#   2391#   2397#   2400#   2401#   2408#   2410#   2418#   2419#   2436#   2440#
                        2494#   2498#   2555#   2559#   2564#   2565#   2570#   2572#   2597#   2601#   2640#   2642#   2700#
                        2707#   2713#   2714#   2718#   2720#   2760#   2764#   2790#   2792#   2795#   2809#   2812#   2828#
                        2831#   2851#   2861#   2863#   2864#   2872#   2875#   2879#   2882#   2900#   2905#   2937#   2939#
                        2942#   2957#   2962#   2978#   2985#   3002#   3009#   3012#   3013#   3042#   3046#   3116#   3117#
                        3129#   3133#   3168#   3172#   3205#   3209#   3215#   3220#   3226#   3233#   3263#   3267#   3271#
                        3275#   3299#   3303#   3308#   3313#   3314#   3317#   3320#   3323#   3325#   3327#   3335#   3338#
                        3342#   3345#   3418#   3422#   3426#   3427#   3430#   3436#   3440#   3443    3458#   3468#   3471#
                        3514#   3516#   3519#   3524#   3525#   3554#   3556#   3559#   3562#   3564#   3567#   3568#   3569#
                        3571#   3572#   3615#   3618#   3621#   3625#   3626#   3633#   3644#   3645#   3654#   3656#   3661#
                        3673#   3676#   3680#   3682#   3684#
$TAGNU= 000252              1#   1093    1094#   1095    1096#   1159    1160#   1170    1171#   1216    1218    1219    1220#
                        1232    1233#   1248    1249#   1265    1266#   1284    1285#   1304    1306    1307    1308#   1320
                        1321#   1336    1337#   1353    1354#   1372    1373#   1402    1403#   1418    1419#   1435    1436#
                        1454    1455#   1478    1479#   1494    1495#   1511    1512#   1530    1531#   1561    1562#   1592
                        1593#   1595    1597#   1599    1600#   1602    1604#   1606    1607#   1609    1611#   1626    1627#
                        1664    1665#   1694    1696    1697    1698#   1707    1708#   1735    1736#   1754    1755#   1761
                        1763#   1775    1776#   1785    1786#   1788    1789#   1797    1799#   1827    1829    1830    1831#
                        1862    1863#   1883    1884#   1901    1903    1904    1905#   1937    1938#   1954    1955#   1972
                        1974    1975    1976#   1983    1984#   2005    2006#   2008    2009#   2012    2014#   2020    2023
```

```
                         2024#   2026    2027#   2046    2047#   2048    2049#   2052    2053#   2055    2056#   2078    2079#
                         2100    2101#   2127    2129    2130    2131#   2138    2139#   2183    2184#   2204    2205#   2228
                         2229#   2260    2261#   2278    2279#   2307    2309    2310    2311#   2319    2320#   2337    2338#
                         2339#   2342    2343#   2360    2361#   2362    2363#   2365    2366#   2370    2372#   2386    2387#
                         2389    2391#   2394    2396    2397#   2399    2401#   2435    2436#   2493    2494#   2554    2555#
                         2558    2559#   2563    2565#   2596    2597#   2636    2638    2639    2640#   2699    2700#   2706
                         2707#   2712    2714#   2756    2758    2759    2760#   2789    2790#   2791#   2794    2795#   2808
                         2809#   2827    2828#   2850    2851#   2871    2872#   2878    2879#   2896    2898    2899    2900#
                         2936    2937#   2938#   2941    2942#   2956    2957#   2977    2978#   3001    3002#   3038    3040
                         3041    3042#   3115    3116#   3118    3121    3122#   3128    3129#   3167    3168#   3204    3205#
                         3208    3209#   3225    3226#   3259    3261    3262    3263#   3270    3271#   3298    3299#   3307
                         3308#   3312    3314#   3316    3317#   3319    3320#   3334    3335#   3341    3342#   3383#   3417
                         3418#   3421    3422#   3425    3427#   3435    3436#   3467    3468#   3490#   3513    3514#   3515#
                         3518    3519#   3539#   3553    3554#   3555#   3558    3559#   3561    3562#   3563#   3566    3567#
                         3599#   3614    3615#   3617    3618#   3620    3621#   3624    3626#   3643    3645#   3660    3661#
                         3672    3673#   3675    3676#   3713#
STEMP  - 000300          1101#   1102#   1104#   1105#   1106#   1107#   1113#   1114#   1116#   1117#   1118#   1119#   1121#
                         1122#   1123#   1125#   1126#   1128#   1129#   1131#   1132#   1134#   1135#   1136#   1148#   1149#
                         1151#   1152#   1153#   1154#   1155#   1156#   1157#   1158#   1161#   1162#   1164#   1165#   1175#
                         1178#   1179#   1180#   1182#   1183#   1184    1188#   1189#   1191#   1192#   1193#   1221#   1222#
                         1224#   1228#   1229#   1237#   1242#   1243#   1244#   1245#   1253#   1258#   1259#   1261#   1262#
                         1270#   1275#   1276#   1278#   1279#   1289#   1309#   1310#   1312#   1316#   1317#   1325#   1330#
                         1331#   1332#   1333#   1341#   1346#   1347#   1349#   1350#   1358#   1363#   1364#   1366#   1367#
                         1377#   1398#   1399#   1407#   1412#   1413#   1414#   1415#   1423#   1428#   1429#   1431#   1432#
                         1440#   1445#   1446#   1448#   1449#   1459#   1474#   1475#   1483#   1488#   1489#   1490#   1491#
                         1499#   1504#   1505#   1507#   1508#   1516#   1521#   1522#   1524#   1525#   1535#   1558#   1559#
                         1571#   1595#   1596#   1602#   1603#   1609#   1610#   1612#   1613#   1615#   1617#   1619#   1622#
                         1623#   1630#   1631#   1642#   1660#   1661#   1674#   1699#   1700#   1702#   1705#   1706#   1710#
                         1711#   1712#   1713#   1720#   1721#   1740#   1748#   1749#   1757#   1758#   1761#   1762#   1780#
                         1782#   1793#   1794#   1795#   1797#   1798#   1800#   1801#   1802#   1804#   1805    1807#   1808#
                         1832#   1833#   1835#   1839#   1840#   1842#   1843#   1846#   1847#   1866#   1867#   1870#   1875#
                         1876#   1888#   1906#   1907#   1909#   1913#   1914#   1915#   1916#   1921#   1922#   1934#   1935#
                         1942#   1950#   1951#   1959#   1977#   1978#   1980#   1985#   1986#   1988#   1993#   1994#   1995#
                         1996#   1997#   1998#   2003#   2004#   2010#   2011#   2012#   2013#   2015#   2016#   2017#   2019#
                         2022    2031#   2032#   2035#   2036#   2038#   2060#   2061#   2065#   2066#   2068#   2070#   2072#
                         2073#   2082#   2083#   2086#   2087#   2089#   2096#   2097#   2104#   2105#   2108#   2132#   2133#
                         2135#   2140#   2141#   2143#   2146#   2147#   2151#   2152#   2158#   2159#   2168#   2169#   2178#
                         2179#   2187#   2188#   2193#   2194#   2196#   2201#   2202#   2209#   2210#   2216#   2217#   2219#
                         2223#   2224#   2233#   2234#   2238#   2239#   2241#   2245#   2246#   2250#   2251#   2264#   2265#
                         2271#   2272#   2274#   2282#   2283#   2287#   2312#   2313#   2315#   2321#   2322#   2324#   2327#
                         2328#   2329#   2330#   2331#   2332#   2333#   2334#   2340#   2343#   2344#   2345#   2347#   2348#
                         2350#   2351#   2353#   2354#   2356#   2357#   2358#   2359#   2368#   2369#   2370#   2371#   2374#
                         2375#   2376#   2377#   2378#   2381#   2382#   2389#   2390#   2399#   2400#   2406#   2407#   2408#
                         2410#   2413#   2414#   2415#   2416#   2418#   2419#   2421#   2425#   2426#   2427#   2428#   2429#
                         2432#   2433#   2440#   2495#   2496#   2498#   2503#   2504#   2507#   2508#   2510#   2511#   2513#
                         2514#   2515#   2516#   2517#   2518#   2519#   2520#   2532#   2533#   2542#   2543#   2563#   2564#
                         2570#   2572#   2576#   2577#   2579#   2580#   2582#   2583#   2601#   2609#   2610#   2612#   2613#
                         2614#   2642#   2647#   2648#   2649#   2650#   2651#   2652#   2654#   2655#   2656#   2657#   2659#
                         2660#   2662#   2663#   2672#   2673#   2686#   2687#   2695#   2696#   2703#   2704#   2712#   2713#
                         2718#   2720#   2726#   2727#   2731#   2732#   2734#   2735#   2738#   2739#   2741#   2742#   2743#
                         2761#   2762#   2764#   2767#   2768#   2770#   2771#   2784#   2785#   2792#   2795#   2810#   2811#
                         2812#   2816#   2817#   2829#   2830#   2831#   2834#   2835#   2842#   2844#   2845#   2846#   2854#
                         2855#   2858#   2859#   2861#   2863#   2864#   2868#   2869#   2875#   2882#   2902#   2903#   2905#
                         2908#   2909#   2919#   2920#   2932#   2933#   2939#   2942#   2959#   2960#   2962#   2966#   2967#
                         2982#   2983#   2985#   2989#   2990#   2993#   2995#   2996#   2997#   3006#   3007#   3009#   3012#
                         3013#   3018#   3019#   3043#   3044#   3046#   3073#   3074#   3076#   3077#   3078#   3079#   3082#
```

```
                    3083#  3084#  3085#  3088#  3089#  3091#  3092#  3094#  3095#  3097#  3098#  3103#  3104#
                    3107#  3108#  3110#  3111#  3117#  3120   3124#  3125#  3133#  3136#  3137#  3138#  3139#
                    3156#  3157#  3159#  3161#  3163#  3164#  3170#  3171#  3172#  3191#  3192#  3194#  3196#
                    3198#  3200#  3211#  3212#  3213#  3214#  3215#  3218#  3219#  3220#  3228#  3229#  3233#
                    3264#  3265#  3267#  3272#  3273#  3275#  3278#  3279#  3281#  3282#  3284#  3285#  3287#
                    3288#  3303#  3310#  3311#  3312#  3313#  3321#  3322#  3323#  3325#  3327#  3338#  3345#
                    3407#  3408#  3409#  3410#  3411#  3412#  3423#  3424#  3425#  3426#  3428#  3429#  3430#
                    3438#  3439#  3440#  3443#  3444   3446   3456#  3457#  3458#  3459#  3471#  3482#  3504#
                    3505#  3506#  3510#  3516#  3519#  3520#  3521#  3522#  3523#  3524#  3525#  3527#  3528#
                    3532#  3550#  3551#  3556#  3559#  3564#  3567#  3568#  3569#  3571#  3572#  3577#  3593#
                    3594#  3610#  3611#  3612#  3613#  3622#  3623#  3624#  3625#  3631#  3632#  3633#  3635#
                    3636#  3637#  3638#  3639#  3640#  3641#  3642#  3643#  3644#  3646#  3647#  3648#  3649#
                    3650#  3651#  3652#  3653#  3654#  3656#  3657   3662#  3663#  3664#  3665#  3666#  3667#
                    3678#  3679#  3680#  3682#  3684#  3687#  3688#  3689#  3690#  3705#  3749#
$STESTN  001200      960#  1146*  1212*  1300*  1389*  1471*  1552*  1585*  1654*  1690*  1823*  1897*  1969*
                    2123*  2301*  2487*  2633*  2753*  2893*  3034*  3256*  3718   4220*
$STIMES  001160      946#  1045*  1145*  1211*  1221*  1299*  1309*  1388*  1470*  1551*  1584*  1612*  1653*
                    1689*  1699*  1822*  1832*  1896*  1906*  1968*  1977*  1985*  2035*  2065*  2086*  2122*
                    2132*  2140*  2193*  2216*  2238*  2271*  2300*  2312*  2321*  2486*  2495*  2632*  2752*
                    2761*  2858*  2892*  2902*  2959*  2982*  3006*  3033*  3043*  3255*  3264*  3272*  3356*
                    3764*  4208*  4215   4218*  4228
$STKB    001146      939#  3918   3929   3946   4000   4006
$STKS    001144      938#  3918   3927   3943   3967*  3998   4004
$STN   = 000026        1#   627   1138   1145#  1146   1202   1211#  1212   1222   1295   1299#  1300   1310
                    1384   1388#  1389   1466   1470#  1471   1547   1551#  1552   1580   1584#  1585   1613
                    1649   1653#  1654   1683   1689#  1690   1700   1810   1816   1822#  1823   1833   1892
                    1896#  1897   1907   1962   1968#  1969   1978   1986   2036   2066   2087   2111   2118
                    2122#  2123   2133   2141   2194   2217   2239   2272   2293   2300#  2301   2313   2322
                    2442   2477   2486#  2487   2496   2625   2632#  2633   2747   2752#  2753   2762   2859
                    2888   2892#  2893   2903   2960   2983   3007   3028   3033#  3034   3044   3140   3247
                    3255#  3256   3265   3273   3347   3352   3356#
$STPB    001152      941#  3904*  3915
$STPFLG  001157      945#  3853   3915
$STPS    001150      940#  3902   3915
$STRAP   014260     1040   4381#
$STRAP2  014302     4392#  4403
$STRP  = 000012     4396#  4405#  4406#  4407#  4408#  4409#  4410   4411#  4412   4413#  4414#  4415#
$STRPAD  014314     4386   4403#
$STSKO = 000245     1094#  1101   1160#  1183   1220#  1224   1233#  1237   1249#  1253   1266#  1270   1285#
                    1289   1308#  1312   1321#  1325   1337#  1341   1354#  1358   1373#  1377   1403#  1407
                    1419#  1423   1436#  1440   1455#  1459   1479#  1483   1495#  1499   1512#  1516   1531#
                    1535   1562#  1571   1593#  1596   1597#  1619   1627#  1642   1665#  1674   1698#  1702
                    1708#  1807   1831#  1835   1863#  1870   1884#  1888   1905#  1909   1938#  1942   1955#
                    1959   1976#  1980   1984#  1988   2006#  2019   2027#  2038   2047#  2072   2079#  2089
                    2101#  2108   2131#  2135   2139#  2143   2184#  2196   2205#  2219   2229#  2241   2261#
                    2274   2279#  2287   2311#  2315   2320#  2324   2338#  2340   2343#  2419   2436#  2440
                    2494#  2498   2555#  2572   2597#  2601   2640#  2642   2700#  2720   2760#  2764   2790#
                    2792   2795#  2864   2872#  2875   2879#  2882   2900#  2905   2937#  2939   2942#  3013
                    3042#  3046   3116#  3117   3129#  3133   3168#  3172   3205#  3220   3226#  3233   3263#
                    3267   3271#  3275   3299#  3303   3308#  3313   3314#  3327   3335#  3338   3342#  3345
                    3418#  3458   3468#  3471   3514#  3516   3519#  3525   3554#  3556   3559#  3572   3615#
                    3656   3661#  3684
$STSK1 = 000246     1096#  1101   1171#  1175   1600#  1603   1604#  1617   1708#  1804   1807   2009#  2013
                    2014#  2017   2049#  2072   2343#  2418   2559#  2564   2565#  2570   2707#  2713   2714#
                    2718   2795#  2863   2942#  3012   3209#  3215   3317#  3325   3418#  3443   3458   3519#
                    3524   3559#  3571   3618#  3644   3645#  3654   3673#  3682
```

$TSK2 = 000247	1607#	1610	1611#	1615	1736#	1740	1755#	1762	1763#	1782	1786#	1798	1799#
	1802	2053#	2070	2361#	2381	2387#	2390	2391#	2410	2809#	2812	2828#	2831
	2851#	2861	2957#	2962	2978#	2985	3002#	3009	3320#	3323	3422#	3426	3427#
	3430	3436#	3440	3562#	3564	3567#	3569	3621#	3625	3626#	3633	3676#	3680
$TSK3 = 000234	1776#	1780	1789#	1795	2056#	2068	2363#	2381	2397#	2400	2401#	2408	3567#
	3568												
$TSK4 = 000124	2366#	2371	2372#	2378									
$TSTM 001004	905#												
$TSTNM 001102	918#	3763#	4129	4165	4170	4197	4219#	4220	4225	4229			
$TTYIN 012614	4025	4026	4043	4047#									
$TYPBN= ****** U	4409												
$TYPDS 013606	4241#	4408											
$TYPE 011622	3853#	4085	4396	4404									
$TYPEC 012034	3883	3890	3897	3902#	3903	3969							
$TYPEX 012102	3908	3910	3913#										
$TYPOC 014056	4326#	4405											
$TYPON 014072	4325	4328#	4407										
$TYPOS 014032	4321#	4406											
$UNIT 001206	963#												
$UNITM 001010	907#												
$USWR 001220	970#	1215	1303	1598	1605	1826	1900	1973	2126	2137	2306	2308	2421
	2635	2755	2895	2897	3037	3258	3260	3315	3506				
$VECT1 001244	995#	3641											
$VECT2 001246	996#												
$XTSTR 013340	4184#												
$YESNO= 000001	1124#	1125#	1127#	1128#	1130#	1131#	1133#	1134#	1178#	1179#	1181#	1182#	1190#
	1191#	1244#	1245#	1261#	1262#	1278#	1279#	1332#	1333#	1349#	1350#	1366#	1367#
	1414#	1415#	1431#	1432#	1448#	1449#	1490#	1491#	1507#	1508#	1524#	1525#	1630#
	1631#	1839#	1840#	1866#	1867#	1913#	1914#	1934#	1935#	1993#	1994#	2015#	2016#
	2031#	2032#	2060#	2061#	2082#	2083#	2104#	2105#	2146#	2147#	2187#	2188#	2209#
	2210#	2233#	2234#	2264#	2265#	2282#	2283#	2333#	2334#	2339#	2340#	2374#	2375#
	2376#	2377#	2422#	2425#	2426#	2427#	2432#	2433#	2510#	2511#	2532#	2533#	2542#
	2543#	2582#	2583#	2611#	2612#	2659#	2660#	2662#	2663#	2686#	2687#	2703#	2704#
	2726#	2727#	2731#	2732#	2740#	2741#	2770#	2771#	2791#	2792#	2810#	2811#	2829#
	2830#	2843#	2844#	2845#	2846#	2854#	2855#	2868#	2869#	2908#	2909#	2919#	2920#
	2938#	2939#	2994#	2995#	2996#	2997#	3018#	3019#	3103#	3104#	3107#	3108#	3110#
	3111#	3124#	3125#	3136#	3137#	3138#	3139#	3156#	3157#	3160#	3161#	3170#	3171#
	3191#	3192#	3195#	3196#	3199#	3200#	3218#	3219#	3228#	3229#	3281#	3282#	3284#
	3285#	3310#	3311#	3321#	3322#	3456#	3457#	3507#	3510#	3515#	3516#	3520#	3521#
	3522#	3523#	3527#	3528#	3555#	3556#	3563#	3564#	3593#	3594#	3648#	3649#	3650#
	3651#	3652#	3653#	3689#	3690#								
$$ARGC= 000000	3383#	3490#	3539#	3599#	3713#								
$$BYTE= 000403	1094#	1169#	1215#	1217#	1231#	1247#	1264#	1283#	1303#	1305#	1319#	1335#	1352#
	1371#	1401#	1417#	1434#	1453#	1477#	1493#	1510#	1529#	1560#	1591#	1598#	1605#
	1625#	1663#	1693#	1695#	1753#	1784#	1787#	1826#	1828#	1882#	1900#	1902#	1953#
	1971#	1973#	1982#	2007#	2025#	2047#	2051#	2054#	2077#	2099#	2126#	2128#	2137#
	2182#	2203#	2227#	2259#	2277#	2306#	2308#	2318#	2361#	2364#	2385#	2393#	2395#
	2434#	2492#	2553#	2557#	2595#	2635#	2637#	2698#	2705#	2755#	2757#	2849#	2870#
	2877#	2895#	2897#	3000#	3037#	3039#	3127#	3166#	3203#	3207#	3224#	3258#	3260#
	3269#	3306#	3315#	3318#	3333#	3340#	3420#	3434#	3466#	3616#	3619#	3659#	3671#
	3674#												
$$DST = 000067	2422#	3507#											
$$FLAG= 000001	1093#	1094#	1096	1169#	1171	1175#	1215#	1217#	1220	1224#	1231#	1233	1237#
	1247#	1249	1253#	1264#	1266	1270#	1283#	1285	1289#	1303#	1305#	1308	1312#
	1319#	1321	1325#	1335#	1337	1341#	1352#	1354	1358#	1371#	1373	1377#	1401#
	1403	1407#	1417#	1419	1423#	1434#	1436	1440#	1453#	1455	1459#	1477#	1479

		1483# 1493# 1495 1499# 1510# 1512 1516# 1529# 1531 1535# 1560# 1562 1571#
		1591# 1593 1598# 1600 1605# 1607 1615# 1617# 1619# 1625# 1627 1642# 1663#
		1665 1674# 1693# 1695# 1698 1702# 1735# 1740# 1753# 1755 1775# 1780# 1782#
		1784# 1786 1787# 1789 1795# 1802# 1826# 1828# 1831 1835# 1862# 1870# 1882#
		1884 1888# 1900# 1902# 1905 1909# 1937# 1942# 1953# 1955 1959# 1971# 1973#
		1976 1980# 1982# 1984 1988# 2007# 2009 2017# 2025# 2027 2038# 2046# 2047#
		2049 2051# 2053 2054# 2056 2068# 2070# 2077# 2079 2089# 2099# 2101 2108#
		2126# 2128# 2131 2135# 2137# 2139 2143# 2182# 2184 2196# 2203# 2205 2219#
		2227# 2229 2241# 2259# 2261 2274# 2277# 2279 2287# 2306# 2308# 2311 2315#
		2318# 2320 2324# 2360# 2361# 2363 2364# 2366 2378# 2385# 2387 2393# 2395#
		2397 2408# 2410# 2434# 2436 2440# 2492# 2494 2498# 2553# 2555 2557# 2559
		2570# 2572# 2595# 2597 2601# 2635# 2637# 2640 2642# 2698# 2700 2705# 2707
		2718# 2720# 2755# 2757# 2760 2764# 2808# 2812# 2827# 2831# 2849# 2851 2861#
		2870# 2872 2875# 2877# 2879 2882# 2895# 2897# 2900 2905# 2956# 2962# 2977#
		2985# 3000# 3002 3009# 3037# 3039# 3042 3046# 3127# 3129 3133# 3166# 3168
		3172# 3203# 3205 3207# 3209 3215# 3220# 3224# 3226 3233# 3258# 3260# 3263
		3267# 3269# 3271 3275# 3298# 3303# 3306# 3308 3315# 3317 3318# 3320 3323#
		3325# 3327# 3333# 3335 3338# 3340# 3342 3345# 3420# 3422 3430# 3434# 3436
		3440# 3466# 3468 3471# 3616# 3618 3619# 3621 3633# 3654# 3659# 3661 3671#
		3673 3674# 3676 3680# 3682# 3684#
$$FROM=	000000	1109# 1725# 1767# 1852# 1926# 2161# 2171# 2253# 2522# 2546# 2590# 2677# 2689#
		2780# 2800# 2819# 2928# 2948# 2969# 3050# 3290# 3451# 3627#
$$GET4=	000000	3779#
$$LOC =	011060	1095# 1096 1170# 1171 1184# 1185 1218# 1219 1232# 1233 1248# 1249 1265#
		1266 1284# 1285 1306# 1307 1320# 1321 1336# 1337 1353# 1354 1372# 1373
		1402# 1403 1418# 1419 1435# 1436 1454# 1455 1478# 1479 1494# 1495 1511#
		1512 1530# 1531 1561# 1562 1592# 1593 1599# 1600 1606# 1607 1626# 1627
		1664# 1665 1696# 1697 1735# 1736 1754# 1755 1775# 1776 1785# 1786 1788#
		1789 1829# 1830 1862# 1863 1883# 1884 1903# 1904 1937# 1938 1954# 1955
		1974# 1975 1983# 1984 2008# 2009 2022# 2023 2026# 2027 2048# 2049 2052#
		2053 2055# 2056 2078# 2079 2100# 2101 2129# 2130 2138# 2139 2183# 2184
		2204# 2205 2228# 2229 2260# 2261 2278# 2279 2309# 2310 2319# 2320 2362#
		2363 2365# 2366 2386# 2387 2394# 2395 2396# 2397 2435# 2436 2493# 2494
		2554# 2555 2558# 2559 2596# 2597 2638# 2639 2699# 2700 2706# 2707 2758#
		2759 2808# 2809 2827# 2828 2850# 2851 2871# 2872 2878# 2879 2898# 2899
		2956# 2957 2977# 2978 3001# 3002 3040# 3041 3120# 3121 3128# 3129 3167#
		3168 3204# 3205 3208# 3209 3225# 3226 3261# 3262 3270# 3271 3298# 3299
		3307# 3308 3316# 3317 3319# 3320 3334# 3335 3341# 3342 3421# 3422 3435#
		3436 3467# 3468 3617# 3618 3620# 3621 3657# 3658 3660# 3661 3672# 3673
		3675# 3676
$$LOCN=	000000	3383# 3485 3490# 3534 3539# 3579 3599# 3707 3713# 3751
$$RETU=	000000	3383# 3469# 3473# 3483 3490# 3533 3539# 3578 3599# 3706 3713# 3750
$$RTN1=	000250	3383# 3469 3482 3490# 3529 3532 3539# 3577 3599# 3691 3705 3713# 3749
$$RTN2=	000251	3383# 3474 3484 3490# 3533 3539# 3578 3599# 3706 3713# 3750
$$SRC =	000027	2422# 3507#
$$TO =	000000	1109# 1110 1725# 1731 1767# 1773 1852# 1858 1926# 1932 2161# 2164 2171#
		2174 2253# 2256 2522# 2528 2546# 2549 2590# 2593 2677# 2683 2689# 2692
		2780# 2781 2782 2800# 2806 2819# 2825 2928# 2929 2930 2948# 2954 2969#
		2975 3050# 3051 3052 3290# 3296 3451# 3454 3627# 3628
$OFILL	014255	4322# 4326# 4336 4371#
$1	001712	1093# 1101
$10	002534	1248 1253#
$100	004612	2055 2068#
$101	004644	2078 2089#
$102	004670	2100 2108#
$103	004732	2127 2130#

$104	004742	2129	2135#	
$105	004762	2138	2143#	
$106	005070	2183	2196#	
$107	005124	2204	2219#	
$11	002562	1265	1270#	
$110	005162	2228	2241#	
$111	005240	2260	2274#	
$112	005260	2278	2287#	
$113	005316	2307	2310#	
$114	005326	2309	2315#	
$115	005346	2319	2324#	
$116	005402	2337	2340#	
$117	005400	2338#	2418	
$12	002612	1284	1289#	
$120	005554	2342	2419#	
$121	005440	2360#	2381	
$122	005474	2362	2382#	
$123	005462	2365	2371#	
$124	005472	2370	2378#	
$125	005506	2386	2390#	
$126	005536	2389	2410#	
$127	005530	2394	2396	2400#
$13	002650	1304	1307#	
$130	005536	2399	2408#	
$131	005622	2435	2440#	
$132	005712	2493	2498#	
$133	006070	2554	2572#	
$134	006066	2558	2564#	
$135	006070	2563	2570#	
$136	006146	2596	2601#	
$137	006234	2636	2639#	
$14	002660	1306	1312#	
$140	006240	2638	2642#	
$141	006434	2699	2720#	
$142	006432	2706	2713#	
$143	006434	2712	2718#	
$144	006540	2756	2759#	
$145	006550	2758	2764#	
$146	006620	2789	2792#	
$147	006616	2790#	2863	
$15	002700	1320	1325#	
$150	006772	2794	2864#	
$151	006666	2808	2812#	
$152	006732	2827	2831#	
$153	006770	2850	2861#	
$154	007012	2871	2875#	
$155	007024	2878	2882#	
$156	007062	2896	2899#	
$157	007072	2898	2905#	
$16	002726	1336	1341#	
$160	007144	2936	2939#	
$161	007142	2937#	3012	
$162	007320	2941	3013#	
$163	007216	2956	2962#	
$164	007266	2977	2985#	
$165	007316	3001	3009#	

$166	007364	3038	3041#	
$167	007374	3040	3046#	
$17	002754	1353	1358#	
$170	007506	3115#	3120	
$171	007522	3118	3121#	
$172	007540	3128	3133#	
$173	007620	3167	3172#	
$174	007700	3204	3220#	
$175	007674	3208	3215#	
$176	007714	3225	3233#	
$177	007760	3259	3262#	
$2	002122	1095	1102#	
$20	003004	1372	1377#	
$200	007770	3261	3267#	
$201	010010	3270	3275#	
$202	010072	3298	3303#	
$203	010110	3307	3313#	
$204	010136	3312	3327#	
$205	010136	3316	3325#	
$206	010136	3319	3323#	
$207	010152	3334	3338#	
$21	003054	1402	1407#	
$210	010164	3341	3345#	
$211	010466	3469	3482#	
$212	010470	3474	3484#	
$213	010336	3417#	3458	
$214	010442	3444	3446	3459#
$215	010356	3421	3426#	
$216	010364	3425	3430#	
$217	010402	3435	3440#	
$22	003102	1418	1423#	
$220	010454	3467	3471#	
$221	010570	3529	3532#	
$222	010570	3533#		
$223	010534	3513	3516#	
$224	010530	3514#	3524	
$225	010560	3518	3525#	
$226	010650	3577#		
$227	010650	3578#		
$23	003130	1435	1440#	
$230	010616	3553	3556#	
$231	010612	3554#	3571	
$232	010642	3558	3572#	
$233	010630	3561	3564#	
$234	010626	3562#	3568	
$235	010640	3566	3569#	
$236	011124	3691	3705#	
$237	011124	3706#		
$24	003160	1454	1459#	
$240	010676	3614#	3657	
$241	010762	3617	3644#	
$242	010722	3620	3625#	
$243	010730	3624	3633#	
$244	011002	3643	3654#	
$245	011070	3660	3684#	
$246	011070	3672	3682#	

```
$247    011070            3675    3680#
$25     003216            1478    1483#
$250    011314            3749#
$251    011314            3750#
$26     003244            1494    1499#
$27     003272            1511    1516#
$3      002320            1159#   1184
$30     003322            1530    1535#
$31     003362            1561    1571#
$32     003412            1592    1596#
$33     003446            1595    1619#
$34     003424            1599    1603#
$35     003446            1602    1617#
$36     003436            1606    1610#
$37     003446            1609    1615#
$4      002344            1170    1175#
$40     003476            1626    1642#
$40CAT= ****** U          4139    4181
$41     003536            1664    1674#
$42     003574            1694    1697#
$43     003604            1696    1702#
$44     003612            1707#   1807
$45     004022            1805    1808#
$46     003666            1735    1740#
$47     003714            1754    1762#
$5      002456            1216    1219#
$50     003750            1761    1782#
$51     003750            1775    1780#
$52     004002            1785    1798#
$53     004000            1788    1795#
$54     004010            1797    1802#
$55     004070            1827    1830#
$56     004100            1829    1835#
$57     004162            1862    1870#
$6      002466            1218    1224#
$60     004204            1883    1888#
$61     004242            1901    1904#
$62     004252            1903    1909#
$63     004334            1937    1942#
$64     004352            1954    1959#
$65     004410            1972    1975#
$66     004420            1974    1980#
$67     004440            1983    1988#
$7      002506            1232    1237#
$70     004460            2005#   2022
$71     004476            2008    2013#
$72     004500            2012    2017#
$73     004514            2020    2023#
$74     004542            2026    2038#
$75     004542            2046#   2072
$76     004614            2048    2073#
$77     004612            2052    2070#
.     = 014340             867#    871#    880     881#    883#    885#    886#    892     893#    895#    897#    915#    952
                          1025#   1033    1048    1049    1095    1096    1097#   1099#   1170    1171    1184    1185    1218
                          1219    1232    1233    1248    1249    1265    1266    1284    1285    1306    1307    1320    1321
                          1336    1337    1353    1354    1372    1373    1402    1403    1418    1419    1435    1436    1454
```

```
                    1455    1478    1479    1494    1495    1511    1512    1530    1531    1561    1562    1592    1593
                    1599    1600    1606    1607    1626    1627    1664    1665    1696    1697    1735    1736    1754
                    1755    1775    1776    1785    1786    1788    1789    1829    1830    1862    1863    1883    1884
                    1903    1904    1937    1938    1954    1955    1974    1975    1983    1984    2008    2009    2022
                    2023    2026    2027    2048    2049    2052    2053    2055    2056    2078    2079    2100    2101
                    2129    2130    2138    2139    2183    2184    2204    2205    2228    2229    2260    2261    2278
                    2279    2309    2310    2319    2320    2362    2363    2365    2366    2386    2387    2394    2395
                    2396    2397    2435    2436    2493    2494    2554    2555    2558    2559    2596    2597    2638
                    2639    2699    2700    2706    2707    2758    2759    2808    2809    2827    2828    2850    2851
                    2871    2872    2878    2879    2898    2899    2956    2957    2977    2978    3001    3002    3040
                    3041    3120    3121    3128    3129    3167    3168    3204    3205    3208    3209    3225    3226
                    3261    3262    3270    3271    3298    3299    3307    3308    3316    3317    3319    3320    3334
                    3335    3341    3342    3360#   3421    3422    3435    3436    3467    3468    3617    3618    3620
                    3621    3657    3658    3660    3661    3672    3673    3675    3676    3702#   3717#   3732#   3738#
                    3787    3791    3807    3831    3915    3918    4047#   4048    4054    4106#   4165    4228    4229
                    4295#
$ASTA= ****** U     4058    4061
$X   = 001000        892#    897
```

```
BEGIN       1#
BGNHRD      1#
BGNHW       1#
BGNINI      1#
BGNMOD      1#    3379
BGNMSG      1#
BGNSFT      1#
BGNSRV      1#    3150    3185    3583
BGNSUB      1#    1160    1227    1241    1257    1274    1315    1329    1345    1362    1397    1411    1427    1444    1473
          1487    1503    1520    1557    1621    1659    1841    1874    1914    2150    2200    2222    2244    2518    2575
          2653
BGNSW       1#
BRESET      1#    1136    1280    1368    1450    1526    1568    1639    1671    1878    2602    3099    3329
CALL        1#    1108    1724    1766    1851    1925    2161    2171    2253    2521    2546    2590    2676    2689    2779
          2799    2818    2927    2947    2968    3049    3289    3451    3626
CASE        1#
CKLOOP      1#    1536    1573    1644    1676
CLRVEC      1#    1185    2606    2735
COMMEN    749#
DECR        1#
DECRU       1#
DEFAUL      1#
DEVREG      1#
DEVTYP      1#
DISPAT      1#
ELSE        1#    1594    1601    1608    1760    1796    2011    2369    2388    2398    2562    2711    3311    3424    3623
          3642
END         1#
ENDCLN      1#
ENDCOM    749#
ENDDEC      1#
ENDDO       1#    1100    2071    2380
ENDHRD      1#
ENDHW       1#
ENDIF       1#    1174    1223    1236    1252    1269    1288    1311    1324    1340    1357    1376    1406    1422    1439
          1458    1482    1498    1515    1534    1570    1614    1616    1618    1641    1673    1701    1739    1779    1781
          1794    1801    1834    1869    1887    1908    1941    1958    1979    1987    2016    2037    2067    2069    2088
          2107    2134    2142    2195    2218    2240    2273    2286    2314    2323    2377    2407    2409    2439    2497
          2569    2571    2600    2641    2717    2719    2763    2811    2830    2860    2874    2881    2904    2961    2984
          3008    3045    3132    3171    3214    3219    3232    3266    3274    3302    3322    3324    3326    3337    3344
          3429    3439    3470    3632    3653    3679    3681    3683
ENDINC      1#    2417    2862    3011    3523    3567    3570
ENDINI      1#
ENDLOO      1#    1806    3457
ENDMOD      1#
ENDMSG      1#
ENDRTN      1#    3481    3531    3576    3704    3748
ENDSEL      1#
ENDSFT      1#
ENDSRV      1#    3178    3239    3594
ENDSUB      1#    1176    1238    1254    1271    1290    1326    1342    1359    1378    1408    1424    1441    1460    1484
          1500    1517    1537    1574    1645    1677    1872    1889    1943    2197    2220    2242    2288    2573    2604
          2721
ENDSW       1#
ENDTST      1#    1196    1291    1379    1461    1538    1575    1646    1678    1814    1890    1960    2114    2289    2472
          2617    2745    2883    3022    3242    3349
```

```
EQUALS      1#
ERRDF       1#  1172
ERRHRD      1#  1234  1250  1267  1286  1322  1338  1355  1374  1404  1420  1437  1456  1480  1496
          1513  1532  1565  1631  1668  1737  1777  1790  1867  1885  1939  1956  2032  2061  2083
          2105  2188  2210  2234  2265  2283  2437  2560  2567  2598  2708  2715  2853  2872  2879
          2978  3003  3130  3300  3335  3342
ERROR      643#  1173  1235  1251  1268  1287  1323  1339  1356  1375  1405  1421  1438  1457  1481
          1497  1514  1533  1566  1632  1669  1738  1778  1791  1868  1886  1940  1957  2033  2062
          2084  2106  2189  2211  2235  2266  2284  2438  2561  2568  2599  2709  2716  2856  2873
          2880  2957  2979  3004  3131  3301  3336  3343
ESCAPE     749#
EXIF        1#  1803
EXIFB       1#  3442
EXIT        1#  1220  1308  1611  1698  1809  1831  1905  1976  1984  2034  2064  2085  2110  2131
          2139  2192  2215  2237  2270  2311  2320  2441  2494  2760  2857  2901  2958  2981  3005
          3042  3139  3263  3271  3346
GETPRI     749#
GETSWR     749#  1078#
GPHARD      1#
GPRMA       1#
GPRMD       1#
GPRML       1#
HEADER      1#
IF          1#  1168  1214  1230  1246  1263  1282  1302  1318  1334  1351  1370  1400  1416  1433
          1452  1476  1492  1509  1528  1559  1597  1604  1624  1662  1752  1783  1786  1825  1881
          1899  1952  1981  2006  2024  2050  2053  2076  2098  2125  2136  2181  2202  2226  2258
          2276  2305  2363  2384  2392  2433  2491  2552  2556  2594  2634  2697  2704  2754  2848
          2869  2876  2894  2999  3036  3126  3165  3202  3206  3223  3257  3314  3317  3332  3339
          3419  3615  3618  3658  3670  3673
IFB         1#  1590  1692  1970  2317  3268  3305  3433  3465
IFCOND      1#
IF ERR      1#  1734  1774  1861  1936  2807  2826  2955  2976  3297
IF NO       1#
INCR        1#  2335  2787  2934  3511  3559
INCRU       1#  3551
INLINE      1#
LASTAD      1#
LEAVE       1#
LET         1#  1103  1105  1112  1115  1117  1120  1122  1125  1128  1131  1134  1147  1151  1152
          1153  1154  1155  1156  1161  1163  1177  1179  1188  1189  1191  1192  1221  1228  1242
          1243  1258  1260  1275  1277  1309  1316  1330  1331  1346  1348  1363  1365  1398  1412
          1413  1428  1430  1445  1447  1474  1488  1489  1504  1506  1521  1523  1558  1612  1622
          1629  1660  1699  1704  1709  1711  1719  1747  1756  1792  1799  1832  1838  1842  1845
          1865  1875  1906  1912  1915  1920  1933  1949  1977  1985  1992  1994  1996  2002  2009
          2014  2030  2035  2059  2065  2081  2086  2095  2103  2132  2140  2145  2151  2157  2167
          2177  2186  2193  2201  2208  2216  2223  2232  2238  2245  2249  2263  2271  2281  2312
          2321  2326  2328  2330  2332  2343  2346  2349  2352  2355  2357  2367  2373  2375  2405
          2412  2414  2420  2425  2427  2431  2495  2502  2506  2509  2513  2514  2515  2516  2517
          2519  2531  2541  2576  2578  2581  2609  2610  2612  2613  2647  2648  2649  2650  2651
          2654  2655  2658  2661  2671  2685  2694  2702  2725  2730  2733  2738  2739  2741  2742
          2761  2766  2769  2783  2809  2815  2828  2833  2841  2844  2853  2858  2867  2902  2907
          2918  2931  2959  2965  2982  2988  2992  2995  3006  3017  3043  3072  3075  3077  3081
          3083  3087  3090  3093  3096  3102  3106  3109  3123  3135  3137  3155  3158  3162  3169
          3190  3193  3197  3210  3212  3217  3227  3264  3272  3277  3280  3283  3286  3309  3320
          3406  3408  3410  3422  3427  3437  3455  3503  3505  3519  3521  3526  3549  3592  3609
          3611  3621  3630  3634  3676  3638  3640  3645  3647  3649  3651  3661  3663  3665  3677
```

```
                3686    3688
LOCAL              1#
LOOP               1#   1706    3416
MSG             1138#   1140    1202#   1204    1466#   1468    1683#   1685    1816#   1818    1962#   1964    2293#   2295    2477#
                2479    2625#   2627    2747#   2749    3028#   3030    3247#   3249
MULT             749#
NEWTST           749#   1138    1202    1295    1384    1466    1547    1580    1649    1683    1816    1892    1962    2118    2293
                2477    2625    2747    2888    3028    3247    3352
NOLOCA             1#
POINTE             1#
POP              749#   1193    2614    2743    3573    3816    3817    4100    4101    4282
PRINTB             1#
PUSH             749#   1186    2607    2736    3546    3797    3803    4061    4063    4084    4241
REPEAT             1#   1158    2004    3114    3613
REPORT             1#    749#
RETURN             1#   3468    3472    3528    3690
ROUTIN             1#   3381    3488    3537    3597    3711
SAVR14             1#
SCOPE            644#   1144    1210    1298    1387    1469    1550    1583    1652    1688    1821    1895    1967    2121    2299
                2485    2631    2751    2891    3032    3254    3355    3762
SELECT             1#
SETPRI           749#   1391    2535    2584    2665    2774    2922    3067
SETTRA          4396#   4405    4406    4407    4408    4410    4412    4413    4414
SETUP            749#   1027
SETVEC             1#   1150    2512    2646
SKIP             749#   1222    1310    1613    1700    1810    1833    1907    1978    1986    2036    2066    2087    2111    2133
                2141    2194    2217    2239    2272    2313    2322    2442    2496    2762    2859    2903    2960    2983    3007
                3044    3140    3265    3273    3347
SLASH            749#
SPACE            749#
STARS            749#    765     767     785     787     805     807     827     829     848     864     878     889     891     898
                 911     952     955    1138    1143    1197    1199    1202    1209    1294    1295    1297    1383    1384    1386
                1465    1466    1468    1543    1546    1547    1549    1579    1580    1582    1648    1649    1651    1682    1683
                1687    1815    1816    1820    1891    1892    1894    1961    1962    1966    2117    2118    2120    2292    2293
                2298    2476    2477    2484    2624    2625    2630    2746    2747    2750    2887    2888    2890    3027    3028
                3031    3148    3152    3183    3187    3246    3247    3253    3352    3354    3380    3401    3487    3501    3536
                3545    3582    3589    3599    3608    3713    3754    3793    3809    3838    3917    3920    3988    4017    4056
                4113    4167    4231    4298    4375
STRUCT             1#
SWRSU            749#   1050#
TRMTRP          4396#
TYPBIN           749#
TYPDEC           749#   3373    3774
TYPNAM           749#   1071
TYPNUM           749#
TYPOCS           749#   3725
TYPOCT           749#   3361    3367    3718    3733    3739    3745    3938
TYPTXT           749#   1096    1097    1098    3357    3363    3369    3714    3720    3729    3735    3741
UNTIL              1#   1182    2018    3116    3655
UNTILB             1#
WAITMS             1#   2160    2170    2252    2545      39    2688    3450
WHILE              1#   1092    2045    2359
WHILEB             1#
$ADDON             1#   1093    1094    1096    1101    1159    1160    1169    1171    1215    1220    1231    1233    1247    1249
                1264    1266    1283    1285    1303    1308    1319    1321    1335    1337    1352    1354    1371    1373    1401
                1403    1417    1419    1434    1436    1453    1455    1477    1479    1493    1495    1510    1512    1529    1531
```

```
            1560    1562    1591    1593    1597    1598    1600    1604    1605    1607    1611    1625    1627    1663    1665
            1693    1698    1707    1708    1735    1736    1753    1755    1763    1775    1776    1784    1786    1787    1789
            1799    1807    1826    1831    1862    1863    1882    1884    1900    1905    1937    1938    1953    1955    1971
            1976    1982    1984    2005    2006    2007    2009    2014    2024    2025    2027    2046    2047    2049    2051
            2053    2054    2056    2072    2077    2079    2099    2101    2126    2131    2137    2139    2182    2184    2203
            2205    2227    2229    2259    2261    2277    2279    2306    2311    2318    2320    2336    2338    2339    2343
            2360    2361    2363    2364    2366    2372    2381    2385    2387    2391    2393    2397    2401    2434    2436
            2492    2494    2553    2555    2557    2559    2565    2595    2597    2635    2640    2698    2700    2705    2707
            2714    2755    2760    2788    2790    2791    2795    2808    2809    2827    2828    2849    2851    2870    2872
            2877    2879    2895    2900    2935    2937    2938    2942    2956    2957    2977    2978    3000    3002    3037
            3042    3115    3116    3122    3127    3129    3166    3168    3203    3205    3207    3209    3224    3226    3258
            3263    3269    3271    3298    3299    3306    3308    3314    3315    3317    3318    3320    3333    3335    3340
            3342    3383    3417    3418    3420    3422    3427    3434    3436    3458    3466    3468    3490    3512    3514
            3515    3519    3539    3552    3554    3555    3559    3560    3562    3563    3567    3599    3614    3615    3616
            3618    3619    3621    3626    3645    3659    3661    3671    3673    3674    3676    3713
$AND          1#    2393
$BRANC        1#    1095    1101    1170    1184    1216    1218    1232    1248    1265    1284    1304    1306    1320    1336
            1353    1372    1402    1418    1435    1454    1478    1494    1511    1530    1561    1592    1595    1599    1602
            1606    1609    1626    1664    1694    1696    1735    1754    1761    1775    1785    1788    1797    1805    1807
            1827    1829    1862    1883    1901    1903    1937    1954    1972    1974    1983    2008    2012    2020    2022
            2026    2048    2052    2055    2072    2078    2100    2127    2129    2138    2183    2204    2228    2260    2278
            2307    2309    2319    2337    2342    2362    2365    2370    2381    2386    2389    2394    2396    2399    2418
            2435    2493    2554    2558    2563    2596    2636    2638    2699    2706    2712    2756    2758    2789    2794
            2808    2827    2850    2863    2871    2878    2896    2898    2936    2941    2956    2977    3001    3012    3038
            3040    3118    3120    3128    3167    3204    3208    3225    3259    3261    3270    3298    3307    3312    3316
            3319    3334    3341    3421    3425    3435    3444    3446    3458    3467    3469    3474    3513    3518    3524
            3529    3553    3558    3561    3566    3568    3571    3617    3620    3624    3643    3657    3660    3672    3675
            3691
$BRCOD        1#    1215    1303    1693    1804    1826    1900    1971    2019    2126    2306    2341    2635    2755    2793
            2895    2940    3037    3117    3258    3443    3445    3517    3557    3565
$CALL         1#    1109    1725    1767    1852    1926    2161    2171    2253    2522    2546    2590    2677    2689    2780
            2800    2819    2928    2948    2969    3050    3290    3451    3627
$CHECK        1#    1094    1169    1215    1231    1247    1264    1283    1303    1319    1335    1352    1371    1401    1417
            1434    1453    1477    1493    1510    1529    1560    1591    1598    1605    1625    1663    1693    1753    1784
            1787    1826    1882    1900    1953    1971    1982    2007    2025    2047    2051    2054    2077    2099    2126
            2137    2182    2203    2227    2259    2277    2306    2318    2361    2364    2385    2393    2434    2492    2553
            2557    2595    2635    2698    2705    2755    2849    2870    2877    2895    3000    3037    3127    3166    3203
            3207    3224    3258    3269    3306    3315    3318    3333    3340    3420    3434    3466    3616    3619    3659
            3671    3674
$CHK1         1#    1104    1106    1113    1116    1118    1121    1135    1148    1151    1152    1153    1154    1155    1157
            1161    1164    1188    1191    1192    1221    1228    1242    1258    1275    1309    1316    1330    1346    1363
            1398    1412    1428    1445    1474    1488    1504    1521    1558    1612    1622    1660    1699    1705    1710
            1712    1720    1748    1757    1793    1800    1832    1842    1846    1875    1906    1915    1921    1950    1977
            1985    1995    1997    2003    2010    2035    2065    2086    2096    2132    2140    2151    2158    2168    2178
            2193    2201    2216    2223    2238    2245    2250    2271    2312    2321    2327    2329    2331    2336    2339
            2344    2347    2350    2353    2356    2358    2368    2406    2413    2415    2428    2495    2503    2507    2513
            2514    2515    2516    2517    2519    2576    2579    2609    2612    2613    2647    2648    2649    2650    2651
            2654    2656    2672    2695    2734    2738    2741    2742    2761    2767    2784    2788    2791    2816    2834
            2858    2902    2932    2935    2938    2959    2966    2982    2989    3006    3043    3073    3076    3078    3082
            3084    3088    3091    3094    3097    3163    3211    3213    3264    3272    3278    3287    3407    3409    3411
            3423    3428    3438    3504    3512    3515    3527    3550    3552    3555    3560    3563    3610    3612    3622
            3631    3635    3637    3639    3641    3646    3662    3664    3666    3678    3687
$CYOP2        1#    1123    1126    1129    1132    1173    1180    1189    1244    1261    1278    1332    1349    1366    1414
            1431    1448    1490    1507    1524    1630    1839    1866    1913    1934    1993    2015    2031    2060    2082
            2104    2146    2187    2209    2233    2264    2282    2333    2374    2376    2421    2426    2432    2510    2532
            2542    2582    2610    2659    2662    2686    2703    2726    2731    2739    2770    2810    2829    2842    2845
```

	2854	2868	2908	2919	2993	2996	3018	3103	3107	3110	3124	3136	3138	3156	3159
	3170	3191	3194	3198	3218	3228	3281	3284	3310	3321	3456	3506	3520	3522	3593
	3648	3650	3652	3689											
$CKR6	1#	2422	3507												
$CMND	1#	1094	1169	1215	1217	1231	1247	1264	1283	1303	1305	1319	1335	1352	1371
	1401	1417	1434	1453	1477	1493	1510	1529	1560	1591	1598	1605	1625	1663	1693
	1695	1753	1784	1787	1826	1828	1882	1900	1902	1953	1971	1973	1982	2007	2025
	2047	2051	2054	2077	2099	2126	2128	2137	2182	2203	2227	2259	2277	2306	2308
	2318	2361	2364	2385	2393	2395	2434	2492	2553	2557	2595	2635	2637	2698	2705
	2755	2757	2849	2870	2877	2895	2897	3000	3037	3039	3127	3166	3203	3207	3224
	3258	3260	3269	3306	3315	3318	3333	3340	3420	3434	3466	3616	3619	3659	3671
	3674														
$COMPA	1#	1094	1169	1215	1231	1247	1264	1283	1303	1319	1335	1352	1371	1401	1417
	1434	1453	1477	1493	1510	1529	1560	1591	1598	1605	1625	1663	1693	1735	1753
	1775	1784	1787	1826	1862	1882	1900	1937	1953	1971	1982	2007	2025	2047	2051
	2054	2077	2099	2126	2137	2182	2203	2227	2259	2277	2306	2318	2336	2361	2364
	2385	2393	2434	2492	2553	2557	2595	2635	2693	2705	2755	2788	2808	2827	2849
	2870	2877	2895	2935	2956	2977	3000	3037	3127	3166	3203	3207	3224	3258	3269
	3298	3306	3315	3318	3333	3340	3420	3434	3466	3512	3552	3560	3616	3619	3659
	3671	3674													
$COUNT	1#	1109	1725	1767	1852	1926	2161	2171	2253	2522	2546	2590	2677	2689	2780
	2800	2819	2928	2948	2969	3050	3290	3451	3627						
$DO	1#	1094	2047	2361											
$ELSE	1#														
$ERRMS	1#														
$EXIFA	1#														
$EXIFO	1#														
$EXIF2	1#	3443													
$EXIF3	1#														
$GENBR	1#	1095	1101	1170	1184	1216	1218	1232	1248	1265	1284	1304	1306	1320	1336
	1353	1372	1402	1418	1435	1454	1478	1494	1511	1530	1561	1592	1595	1599	1602
	1606	1609	1626	1664	1694	1696	1735	1754	1761	1775	1785	1788	1797	1805	1807
	1827	1829	1862	1883	1901	1903	1937	1954	1972	1974	1983	2008	2012	2020	2022
	2026	2048	2052	2055	2072	2078	2100	2127	2129	2138	2183	2204	2228	2260	2278
	2307	2309	2319	2337	2342	2362	2365	2370	2381	2386	2389	2394	2396	2399	2418
	2435	2493	2554	2558	2563	2596	2636	2638	2699	2706	2712	2756	2758	2789	2794
	2808	2827	2850	2863	2871	2878	2896	2898	2936	2941	2956	2977	3001	3012	3038
	3040	3118	3120	3128	3167	3204	3208	3225	3259	3261	3270	3298	3307	3312	3316
	3319	3334	3341	3421	3425	3435	3444	3446	3458	3467	3469	3474	3513	3518	3524
	3529	3553	3558	3561	3566	3568	3571	3617	3620	3624	3643	3657	3660	3672	3675
	3691														
$GENTA	1#	1093	1102	1159	1175	1219	1224	1237	1253	1270	1289	1307	1312	1325	1341
	1358	1377	1407	1423	1440	1459	1483	1499	1516	1535	1571	1596	1603	1610	1615
	1617	1619	1642	1674	1697	1702	1707	1740	1762	1780	1782	1795	1798	1802	1808
	1830	1835	1870	1888	1904	1909	1942	1959	1975	1980	1988	2005	2013	2017	2023
	2038	2046	2068	2070	2073	2089	2108	2130	2135	2143	2196	2219	2241	2274	2287
	2310	2315	2324	2338	2340	2360	2371	2378	2382	2390	2400	2408	2410	2419	2440
	2498	2564	2570	2572	2601	2639	2642	2713	2718	2720	2759	2764	2790	2792	2812
	2831	2861	2864	2875	2882	2899	2905	2937	2939	2962	2985	3009	3013	3041	3046
	3115	3121	3133	3172	3215	3220	3233	3262	3267	3275	3303	3313	3323	3325	3327
	3338	3345	3417	3426	3430	3440	3459	3471	3482	3484	3514	3516	3525	3532	3533
	3554	3556	3562	3564	3569	3572	3577	3578	3614	3625	3633	3644	3654	3680	3682
	3684	3705	3706	3749	3750										
$ F	1#	1169	1215	1231	1247	1264	1283	1303	1319	1335	1352	1371	1401	1417	1434
	1453	1477	1493	1510	1529	1560	1591	1598	1605	1625	1663	1693	1753	1784	1787
	1826	1882	1900	1953	1971	1982	2007	2025	2051	2054	2077	2099	2126	2137	2182

```
          2203  2227  2259  2277  2306  2318  2364  2385  2393  2434  2492  2553  2557  2595  2635
          2698  2705  2755  2849  2870  2877  2895  3000  3037  3127  3166  3203  3207  3224  3258
          3269  3306  3315  3318  3333  3340  3420  3434  3466  3616  3619  3659  3671  3674
$IFCOD      1#  1094  1169  1183  1217  1231  1247  1264  1283  1305  1319  1335  1352  1371  1401
          1417  1434  1453  1477  1493  1510  1529  1560  1591  1598  1605  1625  1663  1695  1753
          1784  1787  1828  1882  1902  1953  1973  1982  2007  2021  2025  2047  2051  2054  2077
          2099  2128  2137  2182  2203  2227  2259  2277  2308  2318  2361  2364  2385  2393  2395
          2434  2492  2553  2557  2595  2637  2698  2705  2757  2849  2870  2877  2897  3000  3039
          3119  3127  3166  3203  3207  3224  3260  3269  3306  3315  3318  3333  3340  3420  3434
          3466  3616  3619  3656  3659  3671  3674
$IFCON      1#  1735  1775  1862  1937  2808  2827  2956  2977  3298
$IFOPR      1#  1095  1170  1184  1218  1232  1248  1265  1284  1306  1320  1336  1353  1372  1402
          1418  1435  1454  1478  1494  1511  1530  1561  1592  1599  1606  1626  1664  1696  1735
          1754  1775  1785  1788  1829  1862  1883  1903  1937  1954  1974  1983  2008  2022  2026
          2048  2052  2055  2078  2100  2129  2138  2183  2204  2228  2260  2278  2309  2319  2362
          2365  2386  2394  2396  2435  2493  2554  2558  2596  2638  2699  2706  2758  2808  2827
          2850  2871  2878  2898  2956  2977  3001  3040  3120  3128  3167  3204  3208  3225  3261
          3270  3298  3307  3316  3319  3334  3341  3421  3435  3467  3617  3620  3657  3660  3672
          3675
$LET        1#  1104  1106  1113  1116  1118  1121  1123  1126  1129  1132  1135  1148  1151  1152
          1153  1154  1155  1157  1161  1164  1178  1180  1188  1189  1191  1192  1221  1228  1242
          1244  1258  1261  1275  1278  1309  1316  1330  1332  1346  1349  1363  1366  1398  1412
          1414  1428  1431  1445  1448  1474  1488  1490  1504  1507  1521  1524  1558  1612  1622
          1630  1660  1699  1705  1710  1712  1720  1748  1757  1793  1800  1832  1839  1842  1846
          1866  1875  1906  1913  1915  1921  1934  1950  1977  1985  1993  1995  1997  2003  2010
          2015  2031  2035  2060  2065  2082  2086  2096  2104  2132  2140  2146  2151  2158  2168
          2178  2187  2193  2201  2209  2216  2223  2233  2238  2245  2250  2264  2271  2282  2312
          2321  2327  2329  2331  2333  2344  2347  2350  2353  2356  2358  2368  2374  2376  2406
          2413  2415  2421  2426  2428  2432  2495  2503  2507  2510  2513  2514  2515  2516  2517
          2519  2532  2542  2576  2579  2582  2609  2610  2612  2613  2647  2648  2649  2650  2651
          2654  2656  2659  2662  2672  2686  2695  2703  2726  2731  2734  2738  2739  2741  2742
          2761  2767  2770  2784  2810  2816  2829  2834  2842  2845  2854  2858  2868  2902  2908
          2919  2932  2959  2966  2982  2989  2993  2996  3006  3018  3043  3073  3076  3078  3082
          3084  3088  3091  3094  3097  3103  3107  3110  3124  3136  3138  3156  3159  3163  3170
          3191  3194  3198  3211  3213  3218  3228  3264  3272  3278  3281  3284  3287  3310  3321
          3407  3409  3411  3423  3428  3438  3456  3504  3506  3520  3522  3527  3550  3593  3610
          3612  3622  3631  3635  3637  3639  3641  3646  3648  3650  3652  3662  3664  3666  3678
          3687  3689
$LPCNT      1#  2336  2788  2935  3512  3552  3560
$OPADD      1#  1124  1127  1130  1133  1178  1181  1190  2015  2339  2374  2376  2510  2611  2740
          2791  2938  3156  3191  3218  3515  3555  3563  3593  3650  3652  3689
$OPAND      1#  2422  3507
$OPCD1      1#  1124  1127  1130  1133  1178  1181  1190  1244  1261  1278  1332  1349  1366  1414
          1431  1448  1490  1507  1524  1630  1839  1866  1913  1934  1993  2015  2031  2060  2082
          2104  2146  2187  2209  2233  2264  2282  2333  2339  2374  2376  2422  2426  2432  2510
          2532  2542  2582  2611  2659  2662  2686  2703  2726  2731  2740  2770  2791  2810  2829
          2843  2845  2854  2868  2908  2919  2938  2994  2996  3018  3103  3107  3110  3124  3136
          3138  3156  3160  3170  3191  3195  3199  3218  3228  3281  3284  3310  3321  3456  3507
          3515  3520  3522  3527  3555  3563  3593  3648  3650  3652  3689
$OPCD2      1#  3520  3527  3648
$OPCOD      1#  1124  1127  1130  1133  1178  1181  1190  1244  1261  1278  1332  1349  1366  1414
          1431  1448  1490  1507  1524  1630  1839  1866  1913  1934  1993  2015  2031  2060  2082
          2104  2146  2187  2209  2233  2264  2282  2333  2339  2374  2376  2422  2426  2432  2510
          2532  2542  2582  2611  2659  2662  2686  2703  2726  2731  2740  2770  2791  2810  2829
          2843  2845  2854  2868  2908  2919  2938  2994  2996  3018  3103  3107  3110  3124  3136
          3138  3156  3160  3170  3191  3195  3199  3218  3228  3281  3284  3310  3321  3456  3507
```

	3515	3520	3522	3527	3555	3563	3593	3648	3650	3652	3689				
$OPCOM	1#	3527													
$OPDEF	1#	1094	1095	1101	1104	1106	1109	1113	1116	1118	1121	1123	1124	1126	1127
	1129	1130	1132	1133	1135	1148	1151	1152	1153	1154	1155	1157	1161	1164	1169
	1170	1178	1180	1181	1183	1184	1188	1189	1190	1191	1192	1215	1216	1217	1218
	1221	1228	1231	1232	1242	1244	1247	1248	1258	1261	1264	1265	1275	1278	1283
	1284	1303	1304	1305	1306	1309	1316	1319	1320	1330	1332	1335	1336	1346	1349
	1352	1353	1363	1366	1371	1372	1398	1401	1402	1412	1414	1417	1418	1428	1431
	1434	1435	1445	1448	1453	1454	1474	1477	1478	1488	1490	1493	1494	1504	1507
	1510	1511	1521	1524	1529	1530	1558	1560	1561	1591	1592	1595	1598	1599	1602
	1605	1606	1609	1612	1622	1625	1626	1630	1660	1663	1664	1693	1694	1695	1696
	1699	1705	1710	1712	1720	1725	1726	1727	1728	1729	1730	1731	1735	1748	1753
	1754	1757	1761	1767	1768	1769	1770	1771	1772	1773	1775	1784	1785	1787	1788
	1793	1797	1800	1804	1805	1807	1826	1827	1828	1829	1832	1839	1842	1846	1852
	1853	1854	1855	1856	1857	1858	1862	1866	1875	1882	1883	1900	1901	1902	1903
	1906	1913	1915	1921	1926	1927	1928	1929	1930	1931	1932	1934	1937	1950	1953
	1954	1971	1972	1973	1974	1977	1982	1983	1985	1993	1995	1997	2003	2007	2008
	2010	2012	2015	2019	2020	2021	2022	2025	2026	2031	2035	2047	2048	2051	2052
	2054	2055	2060	2065	2072	2077	2078	2082	2086	2096	2099	2100	2104	2126	2127
	2128	2129	2132	2137	2138	2140	2146	2151	2158	2161	2162	2163	2164	2168	2171
	2172	2173	2174	2178	2182	2183	2187	2193	2201	2203	2204	2209	2216	2223	2227
	2228	2233	2238	2245	2250	2253	2254	2255	2256	2259	2260	2264	2271	2277	2278
	2282	2306	2307	2308	2309	2312	2318	2319	2321	2327	2329	2331	2333	2336	2337
	2339	2341	2342	2344	2347	2350	2353	2356	2358	2361	2362	2364	2365	2368	2370
	2374	2376	2381	2385	2386	2389	2393	2394	2395	2396	2399	2406	2413	2415	2418
	2421	2422	2423	2424	2426	2428	2432	2434	2435	2492	2493	2495	2503	2507	2510
	2513	2514	2515	2516	2517	2519	2522	2523	2524	2525	2526	2527	2528	2532	2542
	2546	2547	2548	2549	2553	2554	2557	2558	2563	2576	2579	2582	2590	2591	2592
	2593	2595	2596	2609	2610	2611	2612	2613	2635	2636	2637	2638	2647	2648	2649
	2650	2651	2654	2656	2659	2662	2672	2677	2678	2679	2680	2681	2682	2683	2686
	2689	2690	2691	2692	2695	2698	2699	2703	2705	2706	2712	2726	2731	2734	2738
	2739	2740	2741	2742	2755	2756	2757	2758	2761	2767	2770	2780	2781	2782	2784
	2788	2789	2791	2793	2794	2800	2801	2802	2803	2804	2805	2806	2808	2810	2816
	2819	2820	2821	2822	2823	2824	2825	2827	2829	2834	2842	2843	2845	2849	2850
	2854	2858	2863	2868	2870	2871	2877	2878	2895	2896	2897	2898	2902	2908	2919
	2928	2929	2930	2932	2935	2936	2938	2940	2941	2948	2949	2950	2951	2952	2953
	2954	2956	2959	2966	2969	2970	2971	2972	2973	2974	2975	2977	2982	2989	2993
	2994	2996	3000	3001	3006	3012	3018	3037	3038	3039	3040	3043	3050	3051	3052
	3073	3076	3078	3082	3084	3088	3091	3094	3097	3103	3107	3110	3117	3118	3119
	3120	3124	3127	3128	3136	3138	3156	3159	3160	3163	3166	3167	3170	3191	3194
	3195	3198	3199	3203	3204	3207	3208	3211	3213	3218	3224	3225	3228	3258	3259
	3260	3261	3264	3269	3270	3272	3278	3281	3284	3287	3290	3291	3292	3293	3294
	3295	3296	3298	3306	3307	3310	3312	3315	3316	3318	3319	3321	3333	3334	3340
	3341	3407	3409	3411	3420	3421	3423	3425	3428	3434	3435	3438	3443	3444	3445
	3446	3451	3452	3453	3454	3456	3458	3466	3467	3469	3473	3474	3483	3485	3504
	3506	3507	3508	3509	3512	3513	3515	3517	3518	3520	3522	3524	3527	3529	3534
	3550	3552	3553	3555	3557	3558	3560	3561	3563	3565	3566	3568	3571	3579	3593
	3610	3612	3616	3617	3619	3620	3622	3624	3627	3631	3635	3637	3639	3641	3643
	3646	3648	3650	3652	3656	3657	3659	3660	3662	3664	3666	3671	3672	3674	3675
	3678	3687	3689	3691	3707	3751									
$OPEQU	1#														
$OPNAN	1#														
$OPNEG	1#														
$OPNOR	1#														
$OPNOT	1#	1261	1349	1431	1507	1630	1866	1934	2031	2060	2082	2104	2187	2209	2233
	2264	2282	2426	2432	2532	2582	2662	2703	2726	2731	2843	2845	2854	2868	2908

```
          2994    2996    3124    3136    3138    3160    3170    3195    3199    3228
$OPOR        1#  1244    1278    1332    1366    1414    1448    1490    1524    1839    1913    1993    2146    2333    2542
          2659    2686    2770    2810    2829    2919    3018    3103    3107    3110    3281    3284    3310    3321    3522
$OPROT       1#  3648
$OPRO        1#  1104    1106    1113    1116    1118    1121    1135    1148    1151    1152    1153    1154    1155    1157
          1161    1164    1188    1191    1192    1221    1228    1242    1258    1275    1309    1316    1330    1346    1363
          1398    1412    1428    1445    1474    1488    1504    1521    1558    1612    1622    1660    1699    1705    1710
          1712    1720    1748    1757    1793    1800    1832    1842    1846    1875    1906    1915    1921    1950    1977
          1985    1995    1997    2003    2010    2035    2065    2086    2096    2132    2140    2151    2158    2168    2178
          2193    2201    2216    2223    2238    2245    2250    2271    2312    2321    2327    2329    2331    2336    2344
          2347    2350    2353    2356    2358    2368    2406    2413    2415    2428    2495    2503    2507    2513    2514
          2515    2516    2517    2519    2576    2579    2609    2612    2613    2647    2648    2649    2650    2651    2654
          2656    2672    2695    2734    2738    2741    2742    2761    2767    2784    2788    2816    2834    2858    2902
          2932    2935    2959    2966    2982    2989    3006    3043    3073    3076    3078    3082    3084    3088    3091
          3094    3097    3163    3211    3213    3264    3272    3278    3287    3407    3409    3411    3423    3428    3438
          3504    3512    3550    3552    3560    3610    3612    3622    3631    3635    3637    3639    3641    3646    3662
          3664    3666    3678    3687
$OPR1        1#  2339    2791    2938    3515    3527    3555    3563
$OPR2        1#  1123    1126    1129    1132    1178    1180    1189    1244    1261    1278    1332    1349    1366    1414
          1431    1448    1490    1507    1524    1630    1839    1866    1913    1934    1993    2015    2031    2060    2082
          2104    2146    2187    2209    2233    2264    2282    2333    2374    2376    2421    2426    2432    2510    2532
          2542    2582    2610    2659    2662    2686    2703    2726    2731    2739    2770    2810    2829    2842    2845
          2854    2868    2908    2919    2993    2996    3018    3103    3107    3110    3124    3136    3138    3156    3159
          3170    3191    3194    3198    3218    3228    3281    3284    3310    3321    3456    3506    3520    3522    3593
          3648    3650    3652    3689
$OPSHF       1#  3520
$OPSUB       1#  3456
$OPSWB       1#
$OPXOR       1#
$OR          1#  1215    1303    1693    1826    1900    1971    2126    2306    2635    2755    2895    3037    3258
$PUT         1#  1726    1768    1853    1927    2162    2172    2254    2523    2547    2591    2678    2690    2801    2820
          2949    2970    3291    3452
$STRUC       1#
$SUBON       1#  1101    1102    1175    1183    1224    1237    1253    1270    1289    1312    1325    1341    1358    1377
          1407    1423    1440    1459    1483    1499    1516    1535    1571    1596    1603    1610    1615    1617    1619
          1642    1674    1702    1740    1762    1780    1782    1795    1798    1802    1807    1808    1835    1870    1888
          1909    1942    1959    1980    1988    2013    2017    2019    2038    2068    2070    2072    2073    2089    2108
          2135    2143    2196    2219    2241    2274    2287    2315    2324    2340    2343    2371    2378    2381    2382
          2390    2400    2408    2410    2418    2419    2440    2498    2564    2570    2572    2601    2642    2713    2718
          2720    2764    2792    2795    2812    2831    2861    2863    2864    2875    2882    2905    2939    2942    2962
          2985    3009    3012    3013    3046    3117    3133    3172    3215    3220    3233    3267    3275    3303    3313
          3323    3325    3327    3338    3345    3426    3430    3440    3458    3459    3471    3482    3516    3519    3524
          3525    3532    3556    3559    3564    3567    3568    3569    3571    3572    3577    3625    3633    3644    3654
          3656    3680    3682    3684    3705    3749
$THEN        1#  1169    1217    1231    1247    1264    1283    1305    1319    1335    1352    1371    1401    1417    1434
          1453    1477    1493    1510    1529    1560    1591    1598    1605    1625    1663    1695    1753    1784    1787
          1828    1882    1902    1953    1973    1982    2007    2025    2051    2054    2077    2099    2128    2137    2182
          2203    2227    2259    2277    2308    2318    2364    2385    2395    2434    2492    2553    2557    2595    2637
          2698    2705    2757    2849    2870    2877    2897    3000    3039    3127    3166    3203    3207    3224    3260
          3269    3306    3315    3318    3333    3340    3420    3434    3466    3616    3619    3659    3671    3674
$TILA        1#
$TILO        1#
$UNTL2       1#  2019    3117
$UNTL3       1#
$WHILE       1#  1093    2046    2360
$$CMRE     909#
```

```
$$CMTM    909#
$$DEFA      1#
$$ENDS      1#
$$EPRO      1#
$$ESCA    749#
$$GEN       1#  1093  1102  1159  1175  1219  1224  1237  1253  1270  1289  1307  1312  1325  1341
         1358  1377  1407  1423  1440  1459  1483  1499  1516  1535  1571  1596  1603  1610  1615
         1617  1619  1642  1674  1697  1702  1707  1740  1762  1780  1782  1795  1798  1802  1808
         1830  1835  1870  1888  1904  1909  1942  1959  1975  1980  1988  2005  2013  2017  2023
         2038  2046  2068  2070  2073  2089  2108  2130  2135  2143  2196  2219  2241  2274  2287
         2310  2315  2324  2338  2340  2360  2371  2378  2382  2390  2400  2408  2410  2419  2440
         2498  2564  2570  2572  2601  2639  2642  2713  2718  2720  2759  2764  2790  2792  2812
         2831  2861  2864  2875  2882  2899  2905  2937  2939  2962  2985  3009  3013  3041  3046
         3115  3121  3133  3172  3215  3220  3233  3262  3267  3275  3303  3313  3323  3325  3327
         3338  3345  3382  3417  3426  3430  3440  3459  3471  3482  3484  3489  3514  3516  3525
         3532  3533  3538  3554  3556  3562  3564  3569  3572  3577  3578  3598  3614  3625  3633
         3644  3654  3680  3682  3684  3705  3706  3712  3749  3750
$$GETS      1#  1101  1102  1175  1183  1224  1237  1253  1270  1289  1312  1325  1341  1358  1377
         1407  1423  1440  1459  1483  1499  1516  1535  1571  1595  1596  1602  1603  1609  1610
         1615  1617  1619  1642  1674  1702  1740  1761  1762  1780  1782  1795  1797  1798  1802
         1804  1807  1808  1835  1870  1888  1909  1942  1959  1980  1988  2012  2013  2017  2019
         2038  2068  2070  2072  2073  2089  2108  2135  2143  2196  2219  2241  2274  2287  2315
         2324  2340  2343  2370  2371  2378  2381  2382  2389  2390  2399  2400  2408  2410  2418
         2419  2440  2498  2563  2564  2570  2572  2601  2642  2712  2713  2718  2720  2764  2792
         2795  2812  2831  2861  2863  2864  2875  2882  2905  2939  2942  2962  2985  3009  3012
         3013  3046  3117  3133  3172  3215  3220  3233  3267  3275  3303  3312  3313  3323  3325
         3327  3338  3345  3425  3426  3430  3440  3443  3458  3459  3471  3482  3516  3519  3524
         3525  3532  3556  3559  3564  3567  3568  3569  3571  3572  3577  3624  3625  3633  3643
         3644  3654  3656  3680  3682  3684  3705  3749
$$GETT      1#  1595  1602  1609  1761  1797  1804  2012  2370  2389  2399  2563  2712  3312  3425
         3443  3624  3643
$$LPCN      1#  2339  2791  2938  3515  3555  3563
$$NEWT    749#  1138  1202  1295  1384  1466  1547  1580  1649  1683  1816  1892  1962  2118  2293
         2477  2625  2747  2888  3028  3247  3352
$$POP       1#  1101  1102  1175  1183  1224  1237  1253  1270  1289  1312  1325  1341  1358  1377
         1407  1423  1440  1459  1483  1499  1516  1535  1571  1596  1603  1610  1615  1617  1619
         1642  1674  1702  1740  1762  1780  1782  1795  1798  1802  1807  1808  1835  1870  1888
         1909  1942  1959  1980  1988  2013  2017  2019  2038  2068  2070  2072  2073  2089  2108
         2135  2143  2196  2219  2241  2274  2287  2315  2324  2340  2343  2371  2378  2381  2382
         2390  2400  2408  2410  2418  2419  2440  2498  2564  2570  2572  2601  2642  2713  2718
         2720  2764  2792  2795  2812  2831  2861  2863  2864  2875  2882  2905  2939  2942  2962
         2985  3009  3012  3013  3046  3117  3133  3172  3215  3220  3233  3267  3275  3303  3313
         3323  3325  3327  3338  3345  3426  3430  3440  3458  3459  3471  3482  3516  3519  3524
         3525  3532  3556  3559  3564  3567  3568  3569  3571  3572  3577  3625  3633  3644  3654
         3656  3680  3682  3684  3705  3749
$$PUSH      1#  1093  1094  1096  1101  1159  1160  1169  1171  1215  1220  1231  1233  1247  1249
         1264  1266  1283  1285  1303  1308  1319  1321  1335  1337  1352  1354  1371  1373  1401
         1403  1417  1419  1434  1436  1453  1455  1477  1479  1493  1495  1510  1512  1529  1531
         1560  1562  1591  1593  1597  1598  1600  1604  1605  1607  1611  1625  1627  1663  1665
         1693  1698  1707  1708  1735  1736  1753  1755  1763  1775  1776  1784  1786  1787  1789
         1799  1807  1826  1831  1862  1863  1882  1884  1900  1905  1937  1938  1953  1955  1971
         1976  1982  1984  2005  2006  2007  2009  2014  2025  2027  2046  2047  2049  2051  2053
         2054  2056  2072  2077  2079  2099  2101  2126  2131  2137  2139  2182  2184  2203  2205
         2227  2229  2259  2261  2277  2279  2306  2311  2318  2320  2336  2338  2339  2343  2360
         2361  2363  2364  2366  2372  2381  2385  2387  2391  2393  2397  2401  2434  2436  2492
         2494  2553  2555  2557  2559  2565  2595  2597  2635  2640  2698  2700  2705  2707  2714
```

H 11

```
            2755    2760    2788    2790    2791    2795    2808    2809    2827    2828    2849    2851    2870    2872    2877
            2879    2895    2900    2935    2937    2938    2942    2956    2957    2977    2978    3000    3002    3037    3042
            3115    3116    3127    3129    3166    3168    3203    3205    3207    3209    3224    3226    3258    3263    3269
            3271    3298    3299    3306    3308    3314    3315    3317    3318    3320    3333    3335    3340    3342    3383
            3417    3418    3420    3422    3427    3434    3436    3458    3466    3468    3490    3512    3514    3515    3519
            3539    3552    3554    3555    3559    3560    3562    3563    3567    3599    3614    3615    3616    3618    3619
            3621    3626    3645    3659    3661    3671    3673    3674    3676    3713
$$SELE       1#
$$SET     4396#  4405    4406    4407    4408    4410    4412    4413    4414
$$SETM    1066#
$$SETS       1#  1093    1094    1096    1101    1159    1160    1169    1171    1215    1220    1231    1233    1247    1249
            1264    1266    1283    1285    1303    1308    1319    1321    1335    1337    1352    1354    1371    1373    1401
            1403    1417    1419    1434    1436    1453    1455    1477    1479    1493    1495    1510    1512    1529    1531
            1560    1562    1591    1593    1597    1598    1600    1604    1605    1607    1611    1625    1627    1663    1665
            1693    1698    1707    1708    1735    1736    1753    1755    1763    1775    1776    1784    1786    1787    1789
            1799    1807    1826    1831    1862    1863    1882    1884    1900    1905    1937    1938    1953    1955    1971
            1976    1982    1984    2005    2006    2007    2009    2014    2025    2027    2046    2047    2049    2051    2053
            2054    2056    2072    2077    2079    2099    2101    2126    2131    2137    2139    2182    2184    2203    2205
            2227    2229    2259    2261    2277    2279    2306    2311    2318    2320    2336    2338    2339    2343    2360
            2361    2363    2364    2366    2372    2381    2385    2387    2391    2393    2397    2401    2434    2436    2492
            2494    2553    2555    2557    2559    2565    2595    2597    2635    2640    2698    2700    2705    2707    2714
            2755    2760    2788    2790    2791    2795    2808    2809    2827    2828    2849    2851    2870    2872    2877
            2879    2895    2900    2935    2937    2938    2942    2956    2957    2977    2978    3000    3002    3037    3042
            3115    3116    3127    3129    3166    3168    3203    3205    3207    3209    3224    3226    3258    3263    3269
            3271    3298    3299    3306    3308    3314    3315    3317    3318    3320    3333    3335    3340    3342    3383
            3417    3418    3420    3422    3427    3434    3436    3458    3466    3468    3490    3512    3514    3515    3519
            3539    3552    3554    3555    3559    3560    3562    3563    3567    3599    3614    3615    3616    3618    3619
            3621    3626    3645    3659    3661    3671    3673    3674    3676    3713
$$SETT       1#
$$SKIP     749#  1222    1310    1613    1700    1810    1833    1907    1978    1986    2036    2066    2087    2111    2133
            2141    2194    2217    2239    2272    2313    2322    2442    2496    2762    2859    2903    2960    2983    3007
            3044    3140    3265    3273    3347
 EQUAT       1#   639
 HEADE       1#   617
 SETUP       1#  1027
 SWRHI       1#   627
 SWRLO     638#
 $ACT1       1#   876
 $APTB       1#   953#
 $APTH       1#   887
 $APTY       1#  4054
 $CATC       1#   865
 $CMTA       1#   909
 $EOP        1#  3752
 $ERRO       1#  4111
 $POWE       1#  3791
 $READ       1#  3915
 $SCOP       1#  4165
 $TRAP       1#  4373
 $TYPD       1#  4229
 $TYPE       1#  3836
 $TYPO       1#  4296

  ABS    014340    000
```

```
ERRORS DETECTED   0

CVDVCB,CVDVCB SEQ/NL TOC=SPMAC MAC,CVDVCB P11
RUN-TIME  105 95 7 SECONDS
RUN TIME RATIO  1737/207=8 3
CORE USED   34K  (67 PAGES)
```