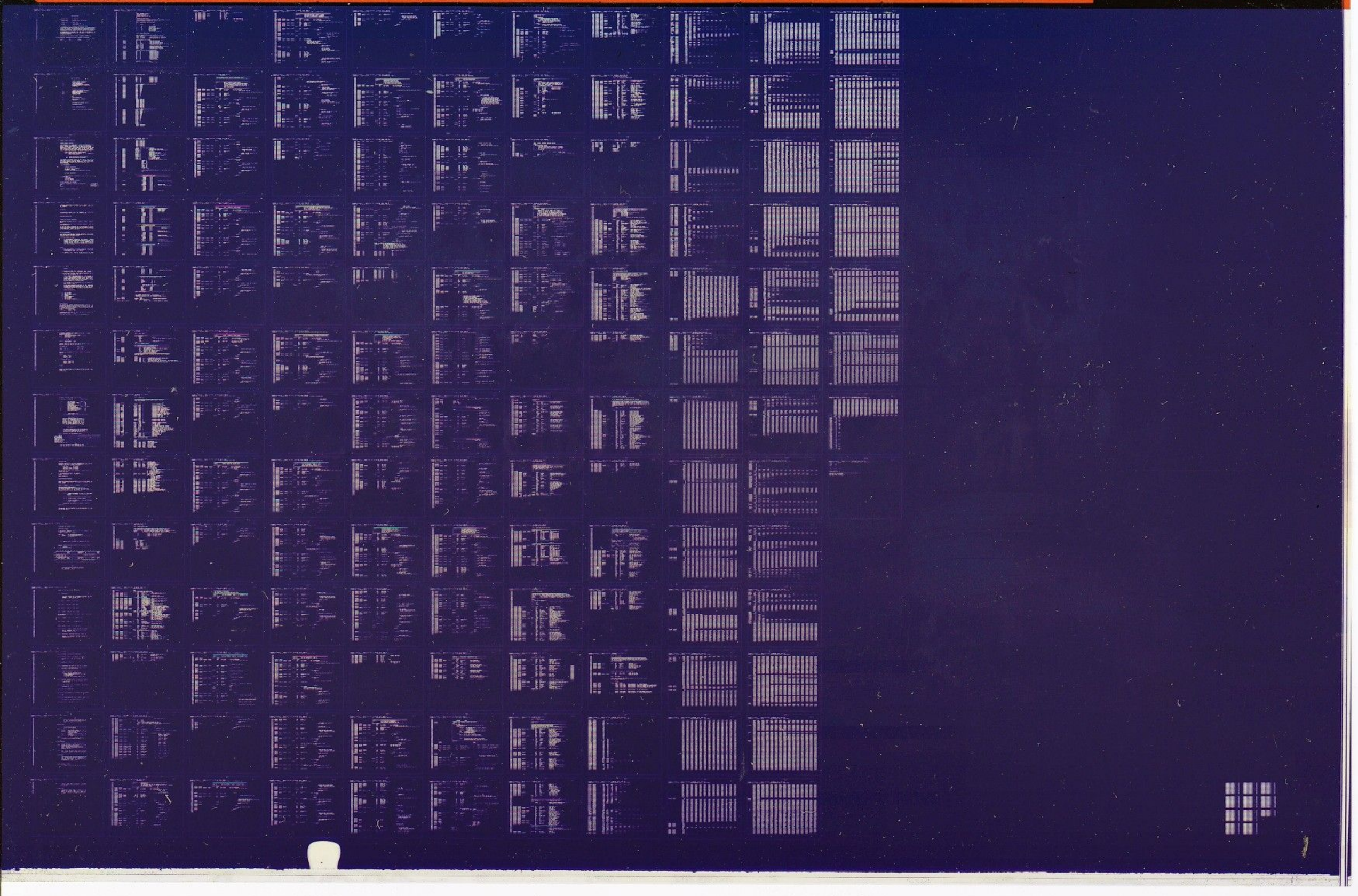
DLV-11F DLV-11F OFFLNE TST
CVDVCCO

AH-E007C-MC
FICHE 01 OF 01

FEB 1985
digital
Made In USA



```
                        .REM    0


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

                                PRODUCT CODE:       AC-E006C-MC

                                PRODUCT NAME:       CVDVCCO DLV11-F OFFLINE TEST

                                PRODUCT DATE:       AUGUST 1984

                                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 DIGITAL
                        EQUIPMENT CORPORATION.  DIGITAL EQUIPMENT CORPORATION ASSUMES NO
                        RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

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

                        COPYRIGHT (C) 1977,1984 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 FAMILY PROCESSOR
                                                4K MEMORY - MINIMUM
                                               A SPECIAL WRAP CONNECTOR OR EQUIVALENT (OPTIONAL)


                                                SOFTWARE REQUIREMENTS:


                                                THIS DIAGNOSTIC IS DESIGNED TO  RUN  IN  ANY  OF  THE
                                         FOLLOWING WAYS:
                                                STAND ALONE
                                                WITH APT MONITOR
                                                WITH ACT MONITOR
                                                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 PREREQUISITES.
```

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

1.5 ASSUMPTIONS.

THIS DIAGNOSTIC ASSUMES THAT THE OPERATOR HAS INITIALIZED 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 FOR START AND RESTART.

THE USER CAN SELECT A SPECIFIC 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 WITHOUT ITERATIONS.)

2.2 SPECIAL ENVIRONMENTS.

THIS DIAGNOSTIC FOLLOWS THE STANDARD PROCEDURE FOR RUNNING 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 SWITCH REGISTER (LOC. 176) is USED.

CONTROL:

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

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

```
                        PROGRAM.

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

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

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

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

                        C)  IF THE INPUT CHARACTER IS NOT ONE  OF  THE  CHARACTERS
                        MENTIONED ABOVE THEN A QUESTION MARK (?) WILL BE TYPED
                        FOLLOWED BY A CARRAGE RETURN AND A LINE FEED 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 BIT 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 RESPECTIVELY.
```

```
                        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 BIT 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
                        -------


                        LOCATION $USWR CONTAINS ALL THE USER SELECTABLE OPTIONS.   THE
                        VALUES   IN  THIS  WORD  MUST  CONFORM  TO  THE  ACTUAL  BOARD
                        CONFIGURATION.
```

THE DEFAULT VALUE OF $USWR IS AS FOLLOWS:

BIT POSITION	DEFINITION	DEFAULT VALUE
0-3	# OF DATA BITS	10(8) = 8
4	PARITY ENABLED -\(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>

THIS 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 14

THIS 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 IS SET.

DLV11-F INDIVIDUAL TEST REQUIREMENTS TABLE

TEST #	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20	21	22	23	24
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.
                                BITS  15,13,10  AND  9  OF  THE  SWITCH  REGISTER  CONTROL THE
                                SEQUENCE OF EVENTS AFTER AN ERROR IS CAUGHT.


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

                                        BIT 13 - DISABLES THE PRINTING OF THE ERROR MESSAGE.

                                        BIT 10 - CAUSES THE BELL TO RING ON ERROR.

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

                                       THE ERROR ROUTINE SUPPORTS THE CONTROL G FUNCTION.

                        3.2     ERROR HALTS.
                                ----- ------

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

4.0 PERFORMANCE AND PROGRESS REPORTS.

4.1 PERFORMANCE REPORTS.

AS EACH DEVICE COMPLETES ONE PASS OF THE DIAGNOSTIC THE FOLLOWING WILL BE TYPED:

```
CSR:_____.VECTOR:_____.ERRORS:_____
```

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

NOTE

THIS IS TYPED AFTER THE DEVICE HAS COMPLETED ITS PASS.

AFTER ALL DEVICES HAVE BEEN EXERCISED AN END PASS STATEMENT IS TYPED:

"ENDPASS#_____."

5.0 DEVICE INFORMATION TABLES.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCSR					RCVR ACT				RCVR DONE	RCVR IE						RDR ENB
RBUF	ERRO R	OR ERR	FR ERR	P ERR					RECEIVED DATA BUFFER							
TCSR	PROGRAMMABLE BAUD RATE SELECT				PBR ENAB				XMIT RDY	XMIT IE			MAINT		BREAK	
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   WITHIN   A
                                               REASONABLE  AMOUNT  OF  TIME  AND  THAT  RESET
                                               CLEARS THE BIT.
```

```
                                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 SIDE 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-JUSTIFIED
                                                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
                                        ------- --------  ------
                                                THIS 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-11-DVDVC-C
                        ;*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
```

```
                        
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  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  BITS
010000                  PBAUD0=         BIT12           ; /
004000                  PBAUDSET=       BIT11           ; ENABLE SETTING OF
                                                        ; PROGRAMMABLE BAUDE RATE
                        ; UNUSED        BIT10
                        ; UNUSED        BIT09
                        ; UNUSED        BIT08
000200                  XMITRDY=        BIT07           ; TRANSMITTER READY
000100                  XMITIE=         BIT06           ; TRANSMITTER INTERRUPT ENABLE
                        ; UNUSED        BIT05
                        ; UNUSED        BIT04
                        ; UNUSED        BIT03
000004                  MAINT=          BIT02           ; MAINTENANCE SET BIT
                        ; UNUSED        BIT01
000001                  BREAK=          BIT00           ; SEND BREAK (CONTINUOUS SPACE)


                        ;;*****************************************************************
                        ; TBUF REGISTER BIT NAMES
                        ;;*****************************************************************
                        ; UNUSED        BIT15
                        ; UNUSED        BIT14
                        ; UNUSED        BIT13
                        ; UNUSED        BIT12
                        ; UNUSED        BIT11
                        ; UNUSED        BIT10
                        ; UNUSED        BIT09
                        ; UNUSED        BIT08
000200                  TDATA7=         BIT07           ; \
000100                  TDATA6=         BIT06           ; !
```

```
                                        TDATA5=         BIT05           ; !
                                        TDATA4=         BIT04           ;  \ TRANSMITTER DATA BUFFER
                                        TDATA3=         BIT03           ;  /
                                        TDATA2=         BIT02           ; !
                                        TDATA1=         BIT01           ; !
                                        TDATA0=         BIT00           ; /


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

                                        DATA    =               17
                                        PARITY  =               20
                                        EVENODD =               40
                                        COMSPD  =              100
                                        PBR     =              200

                                        ; BAUDE MUST BE ON THE UPPER
                                        ;  BYTE BOUNDRY OF $USWR.--4 BITS
                                        BAUD    =             7400
                                        BRK     =            10000
                                        WRAP    =            20000
                                        MAINTJUMP =          40000
                                        ERRBITS =           100000
                                ;;*******************************************************************
                                .SBTTL  TRAP CATCHER

                                        .=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
                                        .=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
011410                                  .=52
000052                                  .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.
000044                                  $APTHDR ;;POINT TO APT HEADER BLOCK
001000                                  .=.$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 VERIFY)
001010  000036                  $UNITM: .WORD   30.     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL 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                     .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  DDISP           ;;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          $TIMES: 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   AMADR2  ;;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  ;;IXTERRUPT 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                      I:      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     #1$CMTAG,R6     ;;FIRST LOCATION TO BE CLEARED
1031  001342  005026                                 CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1032  001344  022706  001140                         CMP     #SWR,R6 ;;DONE?
1033  001350  001374                                 BNE     .-6             ;;LOOP BACK IF NO
1034  001352  012706  001100                         MOV     #STACK,SP       ;;SETUP THE STACK POINTER
1035                                          ;;INITIALIZE A FEW VECTORS
1036  001356  012737  013404  000020                 MOV     #1$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1037  001364  012737  000340  000022                 MOV     #340,@#IOTVEC+2 ;;LEVEL 7
1038  001372  012737  013204  000030                 MOV     #1$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1039  001400  012737  000340  000032                 MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1040  001406  012737  014336  000034                 MOV     #1$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1041  001414  012737  000340  000036                 MOV     #340,@#TRAPVEC+2;LEVEL 7
1042  001422  012737  011444  000024                 MOV     #1$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
1043  001430  012737  000340  000026                 MOV     #340,@#PWRVEC+2 ;;LEVEL 7
1044  001436  016767  007714  007704                 MOV     #ENDCT,1$EOPCT  ;;SETUP END-OF-PROGRAM COUNTER
1045  001444  005067  177510                        CLR     1$TIMES         ;;INITIALIZE NUMBER OF ITERATIONS
1046  001450  005067  177506                        CLR     1$ESCAPE        ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1047  001454  112767  000001  177433                 MOVB    #1,1$ERMAX      ;;ALLOW ONE ERROR PER TEST
1048  001462  012767  001462  177416                 MOV     #.,1$LPADR      ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1049  001470  012767  001470  177412                 MOV     #.,1$LPERR      ;;SETUP THE ERROR LOOP ADDRESS
1050                                          ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1051                                          ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1052  001476  013746  000004                         MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1053  001502  012737  001536  000004                 MOV     #64$,@#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 REGISTER
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     1$PASS          ;;CLEAR PASS COUNT
1067  001570  132767  000200  177417                 BITB    #APTSIZE,1$ENVM ;;TEST USER SIZE UNDER APT
1068  001576  001403                                 BEQ     67$             ;;YES,USE NON-APT SWITCH
1069  001600  012767  001216  177332                 MOV     #1$SWREG,SWR    ;;NO,USE APT SWITCH REGISTER
1070  001606                          67$:
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  011410  000042                 CMP     #1$ENDAD,@#42   ;;ACT-11?
1076  001622  001433                                 BEQ     68$             ;;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    1$ENV,#1        ;;ARE WE RUNNING UNDER APT?
1082  001644  001406                                 BEQ     70$             ;;BRANCH IF YES
```

L2


```
 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$:   .ASCIZ  <CRLF>*MD-11-DVDVC-C*<CRLF>
 1091  001712                          68$:
```

```
001712                                                         WHILE $DEVM EQ #0 DO
001712                          50000$:
001712  005767  177334                   TST     $DEVM
001716  001101                           BNE     50001$
001720                                     TYPTXT <<CRLF>!I HAVE NO DEVICE TO TEST.!>
001762                                   TYPTXT  <<CRLF>!SET UP $DEVM TO INDICATE ACTUAL CONFIGURATION.!>
002050                                   TYPTXT  <<CRLF>!TYPE PROCEED (P) TO CONTINUE.!>
002116  000000                           HALT
002120                                                         ENDDO
002120  000674                           BR      50000$
002122                          50001$:
002122                                                         LET INITFLAG := #1
002122  012767  000001  006760           MOV     #1,INITFLAG
002130                                                         LET BITMASK := #BIT15    ; START AT CONSOLE
002130  012767  100000  006750           MOV     #BIT15,BITMASK
002136                          LOOP:
002136                                   CALL    CYCLE           ; NO ARGUMENTS--ADDRS -> NEXT ADDRESS
002136  004767  006524                   JSR     PC,CYCLE
                                                                 ;             ADDRS+2 -> NEXT VECTOR
                                                                               ;GET UNIT ADDRESS
002142                                                         LET     DLADD := (ADRS)+
002142  012167  177106                   MOV     (ADRS)+,DLADD
                                                                               ;GET UNIT VECTOR
002146                                                         LET     DLVEC := (ADRS)
002146  011167  177104                   MOV     (ADRS),DLVEC
002152                                                         LET     ADRS  := DLADD
002152  016701  177076                   MOV     DLADD,ADRS
                                                                               ;RCSR = DLADD  +  0
002156                                                         LET     RCSR  := DLADD
002156  016767  177072  177074           MOV     DLADD,RCSR
002164                                                         LET     RBUF  := DLADD + #2
002164  016767  177064  177070           MOV     DLADD,RBUF
002172  062767  000002  177062           ADD     #2,RBUF
002200                                                         LET     TCSR  := DLADD + #4
002200  016767  177050  177056           MOV     DLADD,TCSR
002206  062767  000004  177050           ADD     #4,TCSR
002214                                                         LET     TCSRHI := DLADD + #5
002214  016767  177034  177044           MOV     DLADD,TCSRHI
002222  062767  000005  177036           ADD     #5,TCSRHI
002230                                                         LET     TBUF  := DLADD + #6
002230  016767  177020  177032           MOV     DLADD,TBUF
002236  062767  000006  177024           ADD     #6,TBUF
002244                                                         LET     R5    := #R5STACK
002244  012705  001334                   MOV     #R5STACK,R5
                                                                               ;;BRESET
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  010656                   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                           50002$:
1160  002320                                                                       BGNSUB
1161  002320  012767  002326  176562           MOV     #64$,$LPERR
1162                                                                                 ;CLEAR FLAG
1163  002326                                                                         LET INTFLAG := #0
1164  002326  005067  006332                   CLR     INTFLAG
1165
1166                                                                                 ;READ FLAG
1167  002332  005711                           TST @ADRS
1168  002334                                                                         IF INTFLAG NE #0 THEN
1169  002334  005767  006324                   TST     INTFLAG
1170  002340  001401                           BEQ     50003$
1171                                                                                 ; FATAL ERROR
1172  002342                                                                           ERRDF 1,,NODL
1173  002342  104001                           ERROR   1
1174  002344                                                                         ENDIF
1175  002344                           50003$:
1176  002344                                                                     ENDSUB
1177  002344                                                                     LET     I := I + #2
1178  002344  062767  000002  176720           ADD     #2,I
1179  002352                                                                     LET     ADRS := DLADD + I
1180  002352  016701  176676                   MOV     DLADD,ADRS
1181  002356  066701  176710                   ADD     I,ADRS
1182  002362                                                                   UNTIL I EQ #8.
1183  002362  026727  176704  000010           CMP     I,#8.
1184  002370  001353                           BNE     50002$
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
```

```
1197                                    ;;**********************************************************************
1198                                    ;* THE FOLLOWING 8 TESTS  TEST ALL 'READ WRITE' BITS
1199                                    ;;**********************************************************************
1200
1201
1202                                    ;;**********************************************************************
1203                                    ;*TEST 2          BREAK - TCSR0 SET, CLEAR, RESET
1204                                    ;*               THE BREAK BIT IS USUALLY USED ON THE CONSOLE
1205                                    ;*               DEVICE. IF ADDITIONAL DLV OPTIONS ARE USED
1206                                    ;*               IT IS RECOMMENDED TO REMOVE THE 'BG' JUMPER AND
1207                                    ;*               CLEAR BIT 12 IN @USWR WHICH WILL CAUSE THIS
1208                                    ;*               TEST TO BE SKIPPED.
1209                                    ;;**********************************************************************
1210  002420  000004                   TST2:   SCOPE
1211  002422  012767  000010  176530            MOV     #10.,@TIMES         ;;DO 10 ITERATIONS
1212  002430  012767  000002  176542            MOV     #2.,@TESTN          ;;SET TEST NUMBER IN APT MAIL BOX
1213
1214  002436                                                                IF @BRK NOTSETIN @USWR OR @APTENV SETIN @ENV THE
1215  002436  032767  010000  176554            BIT     #BRK,@USWR
1216  002444  001404                           BEQ     50004$
1217  002446  032767  000001  176540            BIT     #APTENV.,@ENV
1218  002454  001404                           BEQ     50005$
1219  002456                                  50004$:
1220  002456                                                                    EXIT TEST    ; BREAK NOT INSTALLED
1221  002456  012767  000001  176474            MOV     #1,@TIMES
1222  002464  000452                           BR      TST3                ;;;EXIT THIS TES
1223  002466                                                                ENDIF
1224  002466                                  50005$:
1225
1226                                                                    ; SEE IF IT IS CLEAR
1227  002466                                                            BGNSUB
1228  002466  012767  002474  176414            MOV     #64$,@LPERR
1229
1230  002474                                                                IF      #BREAK SETIN @TCSR THEN
1231  002474  032777  000001  176562            BIT     #BREAK,@TCSR
1232  002502  001401                           BEQ     50006$
1233                                                                        ; BREAK DID NOT RESET IN TCSR
1234  002504                                                                ERRHRD 2,,DIDNOT
1235  002504  104002                           ERROR   2
1236  002506                                                                ENDIF
1237  002506                                  50006$:
1238  002506                                                            ENDSUB
1239
1240                                                                    ; TRY TO SET BREAK BIT
1241  002506                                                            BGNSUB
1242  002506  012767  002514  176374            MOV     #65$,@LPERR
1243  002514                                                                LET     @TCSR := @TCSR SET.BY #BREAK
1244  002514  052777  000001  176542            BIS     #BREAK,@TCSR
1245                                                                        ; STUCK TO 0
1246  002522                                                                IF      #BREAK NOTSETIN @TCSR THEN
1247  002522  032777  000001  176534            BIT     #BREAK,@TCSR
1248  002530  001001                           BNE     50007$
1249                                                                        ; BREAK DID NOT SET IN TCSR
1250  002532                                                                ERRHRD 3,,DIDNOT
1251  002532  104003                           ERROR   3
1252  002534                                                                ENDIF
```

```
002534                                  50007$:
002534                                                                          ENDSUB
                                                               ; TRY TO CLEAR A SET BIT
002534                                                                          BGNSUB
002534  012767  002542  176346          MOV     #66$,$LPERR
002542                                                                              LET     @TCSR := @TCSR CLR.BY #BREAK
002542  042777  000001  176514          BIC     #BREAK,@TCSR
                                                                                        ; SHOULD HAVE CLEARED
002550                                                                              IF      #BREAK SETIN @TCSR THEN
002550  032777  000001  176506          BIT     #BREAK,@TCSR
002556  001401                          BEQ     50010$
                                                                                        ; BREAK  DID NOT CLEAR IN TCSR
002560                                                                                  ERRHRD 4.,DIDNOT
002560  104004                          ERROR   4
002562                                                                              ENDIF
002562                                  50010$:
002562                                                                          ENDSUB
                                                               ; NOW SEE IF RESET CLEARS IT
002562                                                                          BGNSUB
002562  012767  002570  176320          MOV     #67$,$LPERR
002570                                                                              LET     @TCSR := @TCSR SET.BY #BREAK
002570  052777  000001  176466          BIS     #BREAK,@TCSR
                                                                                        ; ISSUE BUS RESET
002576                                                                              BRESET
002576  000005                          RESET
002600                                                                              IF      #BREAK SETIN @TCSR THEN
002600  032777  000001  176456          BIT     #BREAK,@TCSR
002606  001401                          BEQ     50011$
                                                                                        ; BREAK DID NOT RESET IN TCSR
002610                                                                                  ERRHRD 5.,DIDNOT
002610  104005                          ERROR   5
002612                                                                              ENDIF
002612                                  50011$:
002612                                                                          ENDSUB
002612                                                                      ENDTST
```

```
                                12      BREAK - TCSR0 SET, CLEAR, RESET

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 $USWR ORB CONSOLE EQ @TRU
1303   002630  032767  040000  176362           BIT     @MAINTJUMP,$USWR
1304   002636  001404                           BEQ     50012$
1305   002640  126727  006261  000001           CMPB    CONSOLE,@TRUE
1306   002646  001004                           BNE     50013$
1307   002650                           50012$:
1308   002650                                                                   EXIT TEST
1309   002650  012767  000001  176302           MOV     #1,$TIMES
1310   002656  000452                           BR      TST4                    ;;;EXIT THIS TEST
1311   002660                                                                   ENDIF
1312   002660                           50013$:
1313
1314                                                    ; SEE IF IT IS CLEAR
1315   002660                                                                   BGNSUB
1316   002660  012767  002666  176222           MOV     #64$,$LPERR
1317
1318   002666                                                                   IF      @MAINT SETIN @TCSR THEN
1319   002666  032777  000004  176370           BIT     @MAINT,@TCSR
1320   002674  001401                           BEQ     50014$
1321
1322   002676                                                                       ; MAINT DID NOT RESET IN TCSR
1323   002676  104006                           ERROR   6                           ERRHRD 6,,DIDNOT
1324   002700                                                                   ENDIF
1325   002700                           50014$:
1326   002700                                                                   ENDSUB
1327
1328                                                    ; TRY TO SET MAINT BIT
1329   002700                                                                   BGNSUB
1330   002700  012767  002706  176202           MOV     #65$,$LPERR
1331   002706                                                                   LET     @TCSR := @TCSR SET.BY @MAINT
1332   002706  052777  000004  176350           BIS     @MAINT,@TCSR
1333                                                                               ; STUCK TO 0
1334   002714                                                                   IF      @MAINT NOTSETIN @TCSR THEN
1335   002714  032777  000004  176342           BIT     @MAINT,@TCSR
1336   002722  001001                           BNE     50015$
1337                                                                               ; MAINT DID NOT SET IN TCSR
1338   002724                                                                      ERRHRD 7,,DIDNOT
1339   002724  104007                           ERROR   7
1340   002726                                                                   ENDIF
1341   002726                           50015$:
1342   002726                                                                   ENDSUB
1343
1344                                                    ; TRY TO CLEAR A SET BIT
1345   002726                                                                   BGNSUB
1346   002726  012767  002734  176154           MOV     #66$,$LPERR
1347
1348   002734                                                                   LET     @TCSR :- @TCSR CLR.BY @MAINT
1349   002734  042777  000004  176322           BIC     @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     50016$
1354                                                                        ; MAINT DID NOT CLEAR INTCSR
1355  002752                                                                ERRHRD 10,,DIDNOT
1356  002752  104010                          ERROR   10
1357  002754                                                            ENDIF
1358  002754                          50016$:
1359  002754                                                          ENDSUB
1360
1361                                                          ; NOW SEE IF RESET CLEARS IT
1362  002754                                                          @GNSUB
1363  002754  012767  002762  176126          MOV     #67$,$LPERR
1364
1365  002762                                                          LET     @TCSR := @TCSR SET.BY @MAINT
1366  002762  052777  000004  176274          BIS     @MAINT,@TCSR
1367                                                                      ; ISSUE BUS RESET
1368  002770                                                          @RESET
1369  002770  000005                          RESET
1370  002772                                                          IF      @MAINT SETIN @TCSR THEN
1371  002772  032777  000004  176264          BIT     @MAINT,@TCSR
1372  003000  001401                          BEQ     50017$
1373                                                                        ; MAINT DID NOT RESET IN TCSR
1374  003002                                                                ERRHRD 11,,DIDNOT
1375  003002  104011                          ERROR   11
1376  003004                                                            ENDIF
1377  003004                          50017$:
1378  003004                                                          ENDSUB
1379  003004                                                      ENDTST
1380
1381
1382
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 4          XMITIE - TCSR6 SET, CLEAR, RESET
                                ;;**********************************************************************
003004  000004                  TST4:   SCOPE
003006  012767  000010  176144          MOV     #10.,$TIMES     ;;DO 10 ITERATIONS
003014  012767  000004  176156          MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
                                                                ; USE PRIORITY OF 7
003022  012746  000340                  MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
003026  012746  003034                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
003032  000002                          RTI                     ;;POP NEW PC AND PS
003034                          64$:

                                                ; SEE IF IT IS CLEAR
003034                                                  BGNSUB
003034  012767  003042  176046          MOV     #65$,$LPERR

003042                                                  IF      #XMITIE SETIN @TCSR THEN
003042  032777  000100  176214          BIT     #XMITIE,@TCSR
003050  001401                          BEQ     50020$
                                                        ; XMITIE DID NOT RESET IN TCSR
003052                                                  ERRHRD 12.,DIDNOT
003052  104012                          ERROR   12
003054                                                  ENDIF
003054                          50020$:
003054                                                  ENDSUB

                                                ; TRY TO SET XMITIE BIT
003054                                                  BGNSUB
003054  012767  003062  176026          MOV     #66$,$LPERR
003062                                                  LET     @TCSR := @TCSR SET.BY #XMITIE
003062  052777  000100  176174          BIS     #XMITIE,@TCSR
                                                        ; STUCK TO 0
003070                                                  IF      #XMITIE NOTSETIN @TCSR THEN
003070  032777  000100  176166          BIT     #XMITIE,@TCSR
003076  001001                          BNE     50021$
                                                        ; XMIT DID NOT RESET IN TCSR
003100                                                  ERRHRD 13.,DIDNOT
003100  104013                          ERROR   13
003102                                                  ENDIF
003102                          50021$:
003102                                                  ENDSUB

                                                ; TRY TO CLEAR A SET BIT
003102                                                  BGNSUB
003102  012767  003110  176000          MOV     #67$,$LPERR

003110                                                  LET     @TCSR := @TCSR CLR.BY #XMITIE
003110  042777  000100  176146          BIC     #XMITIE,@TCSR
                                                        ; SHOULD HAVE CLEARED
003116                                                  IF      #XMITIE SETIN @TCSR THEN
003116  032777  000100  176140          BIT     #XMITIE,@TCSR
003124  001401                          BEQ     50022$
                                                        ;  XMIT DID NOT CLEAR IN TCSR
003126                                                  ERRHRD 14.,DIDNOT
003126  104014                          ERROR   14
```

```
003130                                                                      ENDIF
003130                              50022$:
003130                                                                  ENDSUB

                                                      ; NOW SEE IF RESET CLEARS IT
003130                                                                  BGNSUB
003130  012767  003136  175752          MOV     #68$,$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     50023$
                                                                                ; XMIT DID NOT RESET IN TCSR
003156                                                                          ERRHRD 15,,DIDNOT
003156  104015                          ERROR   15
003160                                                                      ENDIF
003160                              50023$:
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 IT IS CLEAR
003176                                                          BGNSUB
003176  012767  003204  175704          MOV     #64$,$LPERR
                                                                IF      @RCVRIE SETIN @RCSR THEN
003204  032777  000100  176046          BIT     @RCVRIE,@RCSR
003212  001401                          BEQ     50024$
                                                                    ; RCVRIE DID NOT RESET IN RCSR
                                                                    ERRHRD 35,,DIDNOT
003214  104035                          ERROR   35
                                                                ENDIF
003216                          50024$:
                                                            ENDSUB
                                                ; TRY TO SET RCVRIE BIT
003216                                                      BGNSUB
003216  012767  003224  175664          MOV     #65$,$LPERR
                                                                LET     @RCSR := @RCSR SET.BY @RCVRIE
003224  052777  000100  176026          BIS     @RCVRIE,@RCSR
                                                                ; STUCK TO 0
003232                                                          IF      @RCVRIE NOTSETIN @RCSR THEN
003232  032777  000100  176020          BIT     @RCVRIE,@RCSR
003240  001001                          BNE     50025$
                                                                    ; RCVRIE DID NOT SET IN RCSR
                                                                    ERRHRD 36,,DIDNOT
003242  104036                          ERROR   36
                                                                ENDIF
003244                          50025$:
                                                            ENDSUB
                                                ; TRY TO CLEAR A SET BIT
003244                                                      BGNSUB
003244  012767  003252  175636          MOV     #66$,$LPERR
                                                                LET     @RCSR := @RCSR CLR.BY @RCVRIE
003252  042777  000100  176000          BIC     @RCVRIE,@RCSR
                                                                ; SHOULD HAVE CLEARED
003260                                                          IF      @RCVRIE SETIN @RCSR THEN
003260  032777  000100  175772          BIT     @RCVRIE,@RCSR
003266  001401                          BEQ     50026$
                                                                    ; RCVRIE DID NOT CLEAR IN RCSR
                                                                    ERRHRD 37,,DIDNOT
003270  104037                          ERROR   37
                                                                ENDIF
003272                          50026$:
                                                            ENDSUB
                                                ; NOW SEE IF RESET CLEARS IT
003272                                                      BGNSUB
```

```
1521  003272  012767  003300  175610          MOV     #67$,$LPERR
1522
1523  003300                                                          LET     @RCSR := @RCSR SET.BY #RCVRIE
1524  003300  052777  000100  175752          BIS     #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          BIT     #RCVRIE,@RCSR
1530  003316  001401                          BEQ     50027$
1531                                                                      ; RCVRIE DID NOT RESET IN RCSR
1532  003320                                                              ERRHRD 40,,DIDNOT
1533  003320  104040                          ERROR   40
1534  003322                                                          ENDIF
1535  003322                          50027$:
1536  003322                                                          CKLOOP
1537  003322                                                      ENDSUB
1538  003322                                                  ENDTST
1539
1540
1541
1542
```

```
                                   ;;*****************************************************************
                                   ;* THE FOLLOWING 4 TESTS VERIFY
                                   ;* THAT RESET (INIT) INITIALIZES READ ONLY BITS.
                                   ;;*****************************************************************
                                   ;;*****************************************************************
                                   ;*TEST 6         TEST THAT RCVRDONE - RCSR 7 - IS CLEARED BY INIT
                                   ;;*****************************************************************
003322  000004                     TST6:   SCOPE
003324  012767  000010  175626             MOV     #10,$TIMES          ;;DO 10 ITERATIONS
003332  012767  000006  175640             MOV     #6,$TESTN           ;;SET TEST NUMBER IN APT MAIL BOX
003340                                                                     BGNSUB
003340  012767  003346  175542             MOV     #64$,$LPERR
003346                                                                         IF      #RCVRDONE SETIN @RCSR THEN
003346  032777  000200  175704             BIT     #RCVRDONE,@RCSR
003354  001402                             BEQ     50030$
                                                                                   ;RCVRDONE SHOULD HAVE CLEARED BY INIT
                                                                                   ; RCVRDONE DID NOT CLEAR IN RCSR
003356                                                                             ERRHRD 41,HRESET, DIDNOT
003356  104041                             ERROR   41
                                                                                   ;REISSUE RESET
003360                                                                             @RESET
003360  000005                             RESET
003362                                                                         ENDIF
003362                             50030$:
                                                                                   ;ALLOW LOOPING AFTER ERROR
003362                                                                         CKLOOP
003362                                                                     ENDSUB
003362                                                                 ENDTST
```

```
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                                                                    IFB CONSOLE EQ #TRUE THEN
1590  003400
1591  003400  126727  005521  000001            CMPB    CONSOLE,#TRUE
1592  003406  001001                            BNE     50031$
1593                                                                        ;; EXECUTE TEST
1594  003410                                                            ELSE
1595  003410  000416                            BR      50032$
1596  003412                            50031$:
1597  003412                                                            IF #WRAP SETIN $USWR THEN
1598  003412  032767  020000  175600            BIT     #WRAP,$USWR
1599  003420  001401                            BEQ     50033$
1600                                                                        ;; EXECUTE TEST
1601  003422                                                            ELSE
1602  003422  000411                            BR      50034$
1603  003424                            50033$:
1604  003424                                                            IF #MAINT SETIN $USWR THEN
1605  003424  032767  000004  175566            BIT     #MAINT,$USWR
1606  003432  001401                            BEQ     50035$
1607                                                                        ;;EXECUTE TEST
1608  003434                                                            ELSE
1609  003434  000404                            BR      50036$
1610  003436                            50035$:
1611  003436                                                               EXIT TEST    ; LINE MUST BE TERMINATED
1612  003436  012767  000001  175514            MOV     #1,$TIMES
1613  003444  000414                            BR      TST10           ;;;EXIT THIS TEST
1614  003446                                                            ENDIF
1615  003446                            50036$:
1616  003446                                                            ENDIF
1617  003446                            50034$:
1618  003446                                                            ENDIF
1619  003446                            50032$:
1620
1621  003446                                                                BGNSUB
1622  003446  012767  003454  175434            MOV     #64$,$LPERR
1623
1624  003454                                                                IF     #RCVRACT SETIN @RCSR THEN
1625  003454  032777  004000  175576            BIT     #RCVRACT,@RCSR
1626  003462  001405                            BEQ     50037$
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                    50037$:
1643                                                        ;ALLOW LOOPING ON ERROR
1644  003476                                                CKLOOP
1645  003476                                            ENDSUB
1646  003476                                        ENDTST
1647
```

```
1648                                    ;;**********************************************************************
1649                                    ;;**********************************************************************
1650                                    ;*TEST 10        TEST THAT XMITRDY - TCSR 7 - IS SET BY INIT
1651                                    ;;**********************************************************************
1652  003476  000004                    TST10:  SCOPE
1653  003500  012767  000010  175452           MOV     #10.,$TIMES         ;;DO 10 ITERATIONS
1654  003506  012767  000010  175464           MOV     #10.,$TESTN         ;;SET TEST NUMBER IN APT MAIL BOX
1655
1656
1657
1658                                                                            BGNSUB
1659  003514
1660  003514  012767  003522  175366           MOV     #64$,$LPERR
1661
1662  003522                                                                        IF      #XMITRDY NOTSETIN @TCSR THEN
1663  003522  032777  000200  175534           BIT     #XMITRDY,@TCSR
1664  003530  001002                           BNE     50040$
1665
1666                                                                                ;RESET SHOULD HAVE SET BIT.
1667                                                                                ;XMITRDY DID NOT SET IN TCSR (AFTER RESE
1668  003532                                                                        ERRHRD 42,HRESET,DIDNOT
1669  003532  104042                           ERROR   42
1670                                                                                ;ISSUE ANOTHER RESET
1671  003534                                                                        BRESET
1672  003534  000005                           RESET
1673  003536                                                                    ENDIF
1674  003536                          50040$:
1675                                                                                ;ALLOW LOOPING ON ERROR
1676  003536                                                                    CKLOOP
1677  003536                                                                ENDSUB
1678  003536                                                            ENDTST
1679
1680
1681
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*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 $ENV THEN
003554  126727  005345  000001        CMPB    CONSOLE,@TRUE
003562  001404                        BEQ     50041$
003564  032767  000001  175422        BIT     @APTENV,$ENV
003572  001404                        BEQ     50042$
003574                          50041$:
003574                                                              EXIT TEST
003574  012767  000001  175356        MOV     #1,$TIMES
003602  000513                        BR      TST12           ;;;EXIT THIS TEST
003604                                                          ENDIF
003604                          50042$:

003604                                                          LET PASS := #1  ;INIT COUNT OF TIMES THRU
003604  012767  000001  000212        MOV     #1,PASS
003612                                                          LOOP            ; START OF LOOP
003612                          50043$:
                                                                                ;  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  004440                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     50045$
                                                                ;XMIT RDY DID NOT SET IN TCSR
003664                                                          ERRHRD 66,,DIDNOT
```

```
1738  003664  104066                          ERROR   66
1739  003666                                                                  ENDIF
1740  003666                          50045$:
1741
1742                                                    ; LOAD TBUF WITH A SECOND CHARACTER
1743                                                    ; CHECK IMMEDIATELY THAT XMITRDY IS CLEAR
1744                                                    ; AND THEN WAIT FOR IT TO SET
1745
1746                                                                    ;SEND SECOND CHARACTER
1747  003666                                                          LET @TBUF :B= #0
1748  003666  105077  175376                 CLRB    @TBUF
1749  003672  000240                                 NOP             ; GIVE IT TIME TO CLEAR
1750                                                                  ; XMITRDY SHOULD HAVE CLEARED UPON
1751                                                                  ; RECEIPT OF A CHARACTER
1752  003674                                                          IF @XMITRDY SETIN @TCSR THEN
1753  003674  032777  000200  175362         BIT     @XMITRDY,@TCSR
1754  003702  001404                         BEQ     50046$
1755
1756  003704                                                                  ; XMITRDY DID NOT CLEAR IN TCSR
1757  003704  012767  177777  000114         MOV     @SET,ERRORFLAG          LET ERRORFLAG := @SET
1758                                                                          ; DEFER ERROR TYPEOUT
1759
1760  003712                                                          ELSE
1761  003712  000416                         BR      50047$
1762  003714                          50046$:
1763                                                                  ;WAIT A MAXIMUM
1764                                                                  ;OF 500 MSEC FOR
1765                                                                  ;XMIT RDY TO SET IN TCSR
1766  003714                                                          CALL TIMER IN <@500,@XMITRDY,TCSR,@SET>
1767  003714  010546                         MOV     R5,-(SP)
1768  003716  012745  177777                 MOV     @SET,-(R5)
1769  003722  016745  175336                 MOV     TCSR,-(R5)
1770  003726  012745  000200                 MOV     @XMITRDY,-(R5)
1771  003732  012745  000500                 MOV     @500,-(R5)
1772  003736  004767  004356                 JSR     PC,TIMER
1773  003742  012605                         MOV     (SP)+,R5
1774  003744                                                          IF.ERROR THEN
1775  003744  103001                         BCC     50050$
1776                                                                      ;XMIT RDY DID NOT SET IN TCSR
1777  003746                                                              ERRHRD 70,,DIDNOT
1778  003746  104070                         ERROR   70
1779  003750                                                          ENDIF
1780  003750                          50050$:
1781  003750                                                      ENDIF ; OF DEFERED ERROR CALL
1782  003750                          50047$:
1783  003750                                                  IF ERRORFLAG EQ @SET THEN
1784  003750  026727  000052  177777         CMP     ERRORFLAG,@SET
1785  003756  001011                         BNE     50051$
1786  003760                                                          IF PASS GT @1 THEN
1787  003760  026727  000040  000001         CMP     PASS,@1
1788  003766  003404                         BLE     50052$
1789                                                                      ; CALL ERROR IF 2ND TRY
1790  003770                                                              ERRHRD 67,,DIDNOT
1791  003770  104067                         ERROR   67
1792  003772                                                              LET EXITFLAG := @SET
1793  003772  012767  177777  000030         MOV     @SET,EXITFLAG
```

```
004000                                                      ENDIF
004000                          50052$:
004000                                                   ELSE              ; NO ERROR
004000  000403                          BR      50053$
004002                          50051$:
004002                                                      LET EXITFLAG := #SET
004002  012767  177777  000020          MOV     #SET,EXITFLAG
004010                                                   ENDIF
004010                          50053$:
004010                                              EXIF        EXITFLAG EQ #SET
004010  026727  000014  177777          CMP     EXITFLAG,#SET
004016  001401                          BEQ     50044$
004020                                              ENDLOOP
004020  000674                          BR      50043$
004022                          50044$:
004022                                                   EXIT ; SKIP AROUND FLAG WORDS
004022  000403                          BR      TST12        ;;;EXIT THIS TEST
004024  000000                  PASS:           0
004026  000000                  ERRORFLAG:      0
004030  000000                  EXITFLAG:       0
004032                                              ENDTST
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 12        TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH MAINT SET)
                                ;*              RESULTS IN RCVRDONE SETTING WITHIN A REASONABLE AMOUNT OF TIME
                                ;*              AND THAT RESET CLEARS THE BIT.
                                ;;**********************************************************************
004032  000004                TST12:  SCOPE
004034  012767  000001  175116        MOV     #1,$TIMES       ;;DO 1 ITERATION
004042  012767  000012  175130        MOV     #12,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                IF #MAINTJUMP NOTSETIN $USWR ORB CONSOLE EQ #TRU
004050  032767  040000  175142        BIT     #MAINTJUMP,$USWR
004056  001404                        BEQ     50054$
004060  126727  005041  000001        CMPB    CONSOLE,#TRUE
004066  001004                        BNE     50055$
004070                          50054$:                             EXIT TEST
004070  012767  000001  175062        MOV     #1,$TIMES
004076  000442                        BR      TST13           ;;;EXIT THIS TEST
                                                                ENDIF
004100                          50055$:
                                                                ; SET THE MAINTENANCE BIT
004100                                                          LET @TCSR := @TCSR SET.BY #MAINT
004100  052777  000004  175156        BIS     #MAINT,@TCSR
                                                                BGNSUB
004106  012767  004114  174774        MOV     #64$,$LPERR
                                        ; SEND A CHARACTER AND LET IT WRAP AROUND
                                                                LET @TBUF :B= #0
004114  105077  175150                CLRB    @TBUF
                                                                ; WAIT A MAXIMUM OF 50 MSEC
                                                                ; FOR RCVR DONE TO SET IN
                                                                ; RCSR
004120                                                          CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
004120  010546                        MOV     R5,-(SP)
004122  012745  177777                MOV     #SET,-(R5)
004126  016745  175126                MOV     RCSR,-(R5)
004132  012745  000200                MOV     #RCVRDONE,-(R5)
004136  012745  000500                MOV     #500,-(R5)
004142  004767  004152                JSR     PC,TIMER
004146  012605                        MOV     (SP)+,R5
                                                                ;DIDN'T SET IN TIME
004150                                                          IF.ERROR THEN
004150  103004                        BCC     50056$
                                                                ; RCVRDONE DID NOT SET IN RCSR
                                                                ; CAN NOT LEAVE WITH MAINT SET
004152                                                          LET   @TCSR := @TCSR CLR.BY #MAINT
004152  042777  000004  175104        BIC     #MAINT,@TCSR
004160                                                          ERRHRD 71,,DIDNOT
004160  104071                        ERROR   71
004162                                                          ENDIF
004162                          50056$:
```

```
1871
1872  004162                                                        ENDSUB
1873
1874  004162                                                        BGNSUB
1875  004162  012767  004170  174720          MOV     #65$,$LPERR
1876                                              ; NOW THAT IT IS SET SEE IF IT CAN BE RESET
1877                                              ; THIS ALSO WILL 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     50057$
1884                                                                        ; RCVRDONE DID NOT RESET IN RCSR.
1885  004202                                                                ERRHRD 72.,DIDNOT
1886  004202  104072                          ERROR   72
1887  004204                                                            ENDIF
1888  004204                          50057$:
1889  004204                                                        ENDSUB
1890  004204                                                    ENDTST
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 13        TEST THAT RCVRDONE IS CLEARED BY READING RBUF
                                ;;**********************************************************************
004204  000004                  TST13:  SCOPE
004206  012767  000010  174744          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004214  012767  000013  174756          MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                IF @MAINTJUMP NOTSETIN $USWR ORB CONSOLE EQ @TRU
004222  032767  040000  174770          BIT     @MAINTJUMP,$USWR
004230  001404                        BEQ     50060$
004232  126727  004667  000001        CMPB    CONSOLE,@TRUE
004240  001004                        BNE     50061$
004242                          50060$:
                                                                    EXIT TEST
004242  012767  000001  174710        MOV     #1,$TIMES
004250  000440                        BR      TST14           ;;;EXIT THIS TEST
                                                                ENDIF
004252                          50061$:
                                                                    ; SET MAINT. BIT
                                                                LET @TCSR := @TCSR SET.BY @MAINT
004252  052777  000004  175004        BIS     @MAINT,@TCSR
                                                                BGNSUB
004260  012767  004266  174622        MOV     @64$,$LPERR
                                        ; OUTPUT A CHARACTER WITH MAINTENANCE
                                        ; SET, AND WAIT FOR XMITRDY TO SET.

                                                                    ; OUTPUT A CHARACTER
                                                                LET @TBUF :B= #0
004266  105077  174776                  CLRB    @TBUF
                                                                ; WAIT MAXIMUM OF 500 MSEC
                                                                ; FOR RCVRDONE TO SET IN
                                                                ; RCSR
004272                                                          CALL TIMER IN <#500,@RCVRDONE,RCSR,#SET>
004272  010546                        MOV     R5,-(SP)
004274  012745  177777                MOV     #SET,-(R5)
004300  016745  174754                MOV     RCSR,-(R5)
004304  012745  000200                MOV     @RCVRDONE,-(R5)
004310  012745  000500                MOV     #500,-(R5)
004314  004767  004000                JSR     PC,TIMER
004320  012605                        MOV     (SP)+,R5
004322                                                          LET     @TCSR := @TCSR CLR.BY @MAINT
004322  042777  000004  174734        BIC     @MAINT,@TCSR
                                                                ; DID IT BECAME READY?
                                                                IF.ERROR  THEN
004330  103001                        BCC     50062$
                                                                ;RCVRDONE DID NOT SET IN RCSR
                                                                ERRMRD 73,, DIDNOT
004332  104073                        ERROR   73
004334                                                          ENDIF
004334                          50062$:
                                                            ENDSUB

                                ; NOW THAT IT IS SET LETS SEE IF READING THE
                                ; 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     50063$
1955                                                                        ;RCVRDONE DID NOT CLEAR IN RCSR
1956  004350                                                                ERRHRD 74,,DIDNOT
1957  004350  104074                     ERROR   74
1958  004352                                                          ENDIF
1959  004352                          50063$:
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 NOTSETIN $USW
004370  126727  004531  000001          CMPB    CONSOLE,#TRUE
004376  001404                          BEQ     50064$
004400  032767  040000  174612          BIT     #MAINTJUMP,$USWR
004406  001004                          BNE     50065$
004410                          50064$:
004410                                                                      EXIT TEST
004410  012767  000001  174542          MOV     #1,$TIMES
004416  000526                          BR      TST15           ;;;EXIT THIS TEST
004420                                                                  ENDIF
004420                          50065$:
004420                                                                  IF #APTENV SETIN $ENV THEN
004420  032767  000001  174566          BIT     #APTENV,$ENV
004426  001404                          BEQ     50066$
004430                                                                      EXIT TEST
004430  012767  000001  174522          MOV     #1,$TIMES
004436  000516                          BR      TST15           ;;;EXIT THIS TEST
004440                                                                  ENDIF
004440                          50066$:


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                          50067$:
004460                                                      IF #RCVRACT SETIN @RCSR THEN
004460  032777  004000  174572          BIT     #RCVRACT,@RCSR
004466  001403                          BEQ     50070$
004470                                                          LET R0 := #SET
004470  012700  177777                  MOV     #SET,R0
004474                                                      ELSE
004474  000401                          BR      50071$
004476                          50070$:
004476                                                          LET R1 := R1 + #1
004476  005201                          INC     R1
004500                                                      ENDIF
```

```
2017  004500                          50071$:
2018  004500                                                          UNTIL R0 EQ #SET OR R1 HI MAX
2019  004500  020027  177777                CMP     R0,#SET
2020  004504  001403                        BEQ     50072$
2021  004506  020167  000160                CMP     R1,MAX
2022  004512  101762                        BLOS    50067$
2023  004514                          50072$:
2024  004514                                                          IF R1 HI MAX THEN
2025  004514  020167  000152                CMP     R1,MAX
2026  004520  101410                        BLOS    50073$
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        BIC     #MAINT,@TCSR
2032  004530                                                              ERRHRD 75,, DIDNOT
2033  004530  104075                        ERROR   75
2034  004532                                                              EXIT TEST
2035  004532  012767  000001  174420        MOV     #1,$TIMES
2036  004540  000455                        BR      TST15           ;;;EXIT THIS TEST
2037  004542                                                          ENDIF
2038  004542                          50073$:
2039
2040                                                          ;CHECK FOR TIMING OF RCVRACT. CLEARING
2041                                                          ;VS RCVRDONE SETTING
2042
2043
2044
2045  004542                                                  WHILE #RCVRACT SETIN @RCSR DO
2046  004542                          50074$:
2047  004542  032777  004000  174510        BIT     #RCVRACT,@RCSR
2048  004550  001421                        BEQ     50075$
2049
2050  004552                                                      IF #RCVRDONE SETIN @RCSR THEN
2051  004552  032777  000200  174500        BIT     #RCVRDONE,@RCSR
2052  004560  001414                        BEQ     50076$
2053  004562                                                        IF #RCVRACT SETIN @RCSR THEN
2054  004562  032777  004000  174470        BIT     #RCVRACT,@RCSR
2055  004570  001410                        BEQ     50077$
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,$TIMES
2066  004610  000431                        BR      TST15           ;;;EXIT THIS TEST
2067  004612                                                        ENDIF
2068  004612                          50077$:
2069  004612                                                      ENDIF
2070  004612                          50076$:
2071  004612                                                  ENDDO
2072  004612  000753                        BR      50074$
```

```
2073  004614                          50075$:
2074
2075                                                                ;RCVRACT = 0 NOW.
2076  004614                                                        IF #RCVRDONE NOTSETIN @RCSR THEN
2077  004614  032777  000200  174436          BIT     #RCVRDONE,@RCSR
2078  004622  001010                          BNE     50100$
2079                                                                    ;RCVRDONE DID NOT SET IN RCSR
2080                                                                    ; CAN NOT LEAVE WITH MAINT SET
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,$TIMES
2087  004642  000414                          BR      TST15           ;;;EXIT THIS TEST
2088  004644                                                        ENDIF
2089  004644                          50100$:
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     50101$
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                          50101$:
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     50102$
004722  126727  004177  000001                    CMPB    CONSOLE,#TRUE
004730  001004                                    BNE     50103$
004732                                    50102$:
004732                                                                                        EXIT TEST
004732  012767  000001  174220                    MOV     #1,$TIMES
004740  000547                                    BR      TST16                  ;;;EXIT THIS TEST
004742                                                                                    ENDIF
004742                                    50103$:
004742                                                                                    IF #MAINTJUMP NOTSETIN $USWR THEN
004742  032767  040000  174250                    BIT     #MAINTJUMP,$USWR
004750  001004                                    BNE     50104$
004752                                                                                        EXIT TEST
004752  012767  000001  174200                    MOV     #1,$TIMES
004760  000537                                    BR      TST16                  ;;;EXIT THIS TEST
004762                                                                                    ENDIF
004762                                    50104$:

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  003562                            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  003542                  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                  BIT     #ORERR,R4
2183  005046  001010                          BNE     50105$
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,$TIMES
2194  005066  000474                          BR      TST16           ;;;EXIT THIS TEST
2195  005070                                                              ENDIF
2196  005070                          50105$:
2197  005070                                                      ENDSUB
2198
2199                                                          ;NOW SEE IF ERROR BIT SET WITH OVERRUN  ERROR:
2200  005070                                                      BGNSUB
2201  005070  012767  005076  174012          MOV     #65$,$LPERR
2202  005076                                                          IF #ERROR NOTSETIN R4 THEN
2203  005076  032704  100000                  BIT     #ERROR,R4
2204  005102  001010                          BNE     50106$
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,$TIMES
2217  005122  000456                          BR      TST16           ;;;EXIT THIS TEST
2218  005124                                                              ENDIF
2219  005124                          50106$:
2220  005124                                                      ENDSUB
2221
2222  005124                                                      BGNSUB
2223  005124  012767  005132  173756          MOV     #66$,$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          BIT     #ORERR,@RBUF
```

```
2228  005140  001010                    BNE     50107$
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,$TIMES
2239  005160  000437                    BR      TST16           ;;;EXIT THIS TEST
2240  005162                                                    ENDIF
2241  005162                        50107$:
2242  005162                                                ENDSUB
2243
2244  005162                                                BGNSUB
2245  005162  012767  005170  173720    MOV     #67$,$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  003370            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     50110$
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,$TIMES
2272  005236  000410                    BR      TST16           ;;;EXIT THIS TEST
2273  005240                                                    ENDIF
2274  005240                        50110$:
2275
```

```
2276  005240                                                          IF @ERROR SETIN @RBUF THEN
2277  005240  032777  100000  174014          BIT     @ERROR,@RBUF
2278  005246  001404                          BEQ     501110
2279                                                                      ;ERROR DID NOT CLEAR IN RBUF
2280                                                                      ; CAN NOT LEAVE WITH MAINT SET
2281  005250                                                          LET     @TCSR := @TCSR CLR.BY @MAINT
2282  005250  042777  000004  174006          BIC     @MAINT,@TCSR
2283  005256                                                              ERRHRD 105,,DIDNOT
2284  005256  104105                          ERROR   105
2285
2286  005260                                                          ENDIF
2287  005260                          501110:
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                                                                        IF @PBR NOTSETIN $USWR OR @MAINTJUMP NOTSETIN $U
2305  005276
2306  005276  032767  000200  173714            BIT     @PBR,$USWR
2307  005304  001404                            BEQ     S0112$
2308  005306  032767  040000  173704            BIT     @MAINTJUMP,$USWR
2309  005314  001004                            BNE     S0113$
2310  005316                            S0112$:
2311  005316                                                                      EXIT TEST
2312  005316  012767  000001  173634            MOV     #1,$TIMES
2313  005324  000553                            BR      TST17               ;;;EXIT THIS TEST
2314  005326                                                                  ENDIF
2315  005326                            S0113$:
2316
2317  005326                                                                  IFB @APTENV SETIN $ENV THEN
2318  005326  132767  000001  173660            BITB    @APTENV,$ENV
2319  005334  001404                            BEQ     S0114$
2320  005336                                                                      EXIT TST
2321  005336  012767  000001  173614            MOV     #1,$TIMES
2322  005344  000543                            BR      TST17               ;;;EXIT THIS TEST
2323  005346                                                                  ENDIF
2324  005346                            S0114$:
2325
2326  005346                                                                  LET ERRCHK := #0        ; CLEAR ERROR WORD
2327  005346  005067  002620                    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 @MAINT
2333  005366  052777  000004  173670            BIS     @MAINT,@TCSR
2334                                                                              ;EACH BAUD RATE
2335  005374                                                                  INCR R3 FROM #0 TO #15. BY #1
2336  005374  005003                            CLR     R3
2337  005376  000401                            BR      S0115$
2338  005400                            S0116$:
2339  005400  005203                            INC     R3
2340  005402                            S0115$:
2341  005402  020327  000017                    CMP     R3,#15.
2342  005406  003062                            BGT     S0117$
2343  005410                                                                  LET R0 := @RBUF
2344  005410  017700  173646                    MOV     @RBUF,R0
2345                                                                              ;CHANGE BAUDE RATE
2346  005414                                                                  LET @TCSRHI :@= RATES(R3)
```

```
2347  005414  116377  005624  173644          MOVB    RATES(R3),@TCSRMI
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                          50120$:
2361  005440  005702                          TST     BIT
2362  005442  001014                          BNE     50121$
2363  005444
2364  005444  032777  000200  173606          BIT     #RCVRDONE,@RCSR
2365  005452  001403                          BEQ     50122$
2366
2367  005454
2368  005454  012702  000001                  MOV     #1,BIT
2369  005460
2370  005460  000404                          BR      50123$
2371  005462                          50122$:
2372
2373  005462
2374  005462  005267  000156                  INC     NEW
2375  005466
2376  005466  005567  000154                  ADC     NEW+2
2377  005472
2378  005472                          50123$:
2379
2380  005472
2381  005472  000762                          BR      50120$
2382  005474                          50121$:
2383
2384  005474
2385  005474  026767  000146  000150          CMP     NEW+2,OLD+2
2386  005502  103001                          BHIS    50124$
2387
2388  005504
2389  005504  000414                          BR      50125$
2390  005506                          50124$:
2391
2392  005506
2393  005506  026767  000134  000136          CMP     NEW+2,OLD+2
2394  005514  001005                          BNE     50126$
2395  005516  026767  000122  000124          CMP     NEW,OLD
2396  005524  103001                          BHIS    50126$
2397
2398  005526
2399  005526  000403                          BR      50127$
2400  005530                          50126$:
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 - ITS READY
        LET BIT := #1

    ELSE

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

        LET NEW+2 := NEW+2 + CARRY

    ENDIF

        ;SIGNALS DONE
ENDDO


IF NEW+2 LO OLD+2 THEN

        ; OK
ELSE

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



        ;OK
    ELSE

        ;NEW+2 > OLD+2 OR
        ;(NEW+2 = OLD+2 AND
```

```
                                                                        ; NEW >= OLD)
                                                                        ;BAUD RATE DIDN'T CHANGE
                                                                    LET ERRCHK := #BIT2 ; SET ERROR INDICATOR
                                                                        ENDIF
                                                                    ENDIF
                                                                        ;UPDATE OLD TIME
                                                                    LET OLD := NEW
                                                                    LET OLD+2 := NEW+2
                                                                ENDINC  ;BAUD RATE
                                                            LET R3 :B= $USWR+1 AND #17      ; PUT BAUD BACK
                                                            LET R3 := R3 CLR.BY #177400
                                                            LET @TCSRHI :B= RATES(R3)       ; LIKE HE WANTED IT
                                                                                ; CAN NOT LEAVE WITH MAINT SET
                                                                            LET    @TCSR := @TCSR CLR.BY #MAINT
                                                                IF #BIT2 SETIN ERRCHK THEN
                                                                                ; REPORT DEFERED ERROR
                                                                            ERRHRD      126
                                                                ENDIF
                                                                    EXIT    ;SKIP TABLE
                                                            ;;;EXIT THIS TEST

RATES:  ;A TABLE OF THE ACTUAL BYTES TO MOVE INTO THE
;UPPER BYTE OF XCSR FOR EACH BAUD RATE
;** NOTE::      THE VALUE INDICATED IN THE COLUMN 'OFFSET
;**             INTO TABLE' CAN BE PLACED INTO BITS<11:8>
;**             OF LOCATION '$USWR' TO CAUSE THE CORROSPONDING
;**             BAUD TO BE SELECTED IN THE DLV11-F UPON
;**             COMPLETION OF THIS TEST.

                                    ;       BAUD    OFFSET INTO TABLE
R0050:  .BYTE   010     ;       50      0
R0070:  .BYTE   030     ;       70      1
R0110:  .BYTE   050     ;       110     2
R0135:  .BYTE   070     ;       135     3
R0150:  .BYTE   110     ;       150     4
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
```

```
2476                                    ;;**********************************************************************
2477                                    ;;**********************************************************************
2478                                    ;*TEST 17        TRANSMITTER INTERRUPT LOGIC TEST
2479                                    ;*              LOGICALLLY THIS IS 4 SEPARATE TESTS
2480                                    ;*              A) DOES TRANSMITTER INTERRUPT LOGIC WORK
2481                                    ;*              B) AT PRIORITY OF 0
2482                                    ;*              C) AND ONLY ONCE
2483                                    ;*              D) BUT NOT WITH INTERRUPT ENABLE CLEAR
2484                                    ;;**********************************************************************
2485  005654  000004                   TST17:  SCOPE
2486  005656  012767  000010  173274           MOV     #10.,$TIMES         ;;DO 10 ITERATIONS
2487  005664  012767  000017  173306           MOV     #17.,$TESTN         ;;SET TEST NUMBER IN APT MAIL BOX
2488
2489
2490
2491  005672                                                               IF @APTENV SETIN $ENV THEN
2492  005672  032767  000001  173314           BIT     @APTENV,$ENV
2493  005700  001404                           BEQ     50131$
2494  005702                                                                   EXIT TEST
2495  005702  012767  000001  173250           MOV     #1,$TIMES
2496  005710  000534                           BR      TST20               ;;;EXIT THIS TEST
2497  005712                                                               ENDIF
2498  005712                                50131$:
2499
2500
2501                                                                             ;CLEAR 'INTERRUPT OCCURED' FLAG
2502  005712                                                               LET INTFLAG := #0
2503  005712  005067  002746                  CLR     INTFLAG
2504
2505                                                                             ;GET VECTOR ADDRESS
2506  005716                                                               LET R3 := DLVEC
2507  005716  016703  173334                  MOV     DLVEC,R3
2508                                                                             ;FOR THE TRANSMITTER
2509  005722                                                               LET R3 := R3 + #4
2510  005722  062703  000004                  ADD     #4,R3
2511                                                                             ;SET VECTOR TO POINT TO TRANS.SRV AT PRI
2512  005726                                                               SETVEC R3, #INTSRV, #PR7
2513  005726  010146                          MOV     R1,-(SP)
2514  005730  010301                          MOV     R3,R1
2515  005732  012721  010656                  MOV     #INTSRV,(R1)+
2516  005736  012711  000340                  MOV     #PR7,(R1)
2517  005742  012601                          MOV     (SP)+,R1
2518  005744                                                               BGNSUB
2519  005744  012767  005752  173136           MOV     #64$,$LPERR
2520                                                                       ;; MAKE SURE THAT TRANSMITTER READY IS SET
2521  005752                                                               CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
2522  005752  010546                          MOV     R5,-(SP)
2523  005754  012745  177777                  MOV     #SET,-(R5)
2524  005760  016745  173300                  MOV     TCSR,-(R5)
2525  005764  012745  000200                  MOV     #XMITRDY,-(R5)
2526  005770  012745  000500                  MOV     #500,-(R5)
2527  005774  004767  002320                  JSR     PC,TIMER
2528  006000  012605                          MOV     (SP)+,R5
2529
2530                                                                             ;CLEAR INTERRUPT ENABLE
2531  006002                                                               LET @TCSR := @TCSR CLR.BY #XMITIE
```

```
2532  006002  042777  000100  173254          BIC     #XMITIE,@TCSR
2533
2534                                                                    ;SET IT TO 0
2535  006010  012746  000000                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
2536  006014  012746  006022                  MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
2537  006020  000002                          RTI                     ;;POP NEW PC AND PS
2538  006022                          65$:
2539
2540                                                                    ;NOW SET I.E. BIT
2541  006022                                                          LET @TCSR := @TCSR SET.BY #XMITIE
2542  006022  052777  000100  173234          BIS     #XMITIE,@TCSR
2543
2544                                                                    ;LET INTERRUPT HAVE TIME TO OCCUR
2545  006030                                                          WAITMS 200.
2546  006030  010546                          MOV     R5,-(SP)
2547  006032  012745  000310                  MOV     #200.,-(R5)
2548  006036  004767  002534                  JSR     PC,WAIT
2549  006042  012605                          MOV     (SP)+,R5
2550
2551                                                                    ;DID EXACTLY 1 INTERRUPT OCCUR
2552  006044                                                          IF INTFLAG NE #1 THEN
2553  006044  026727  002614  000001          CMP     INTFLAG,#1
2554  006052  001406                          BEQ     50132$
2555                                                                        ;NO - WAS IT 0 OR MORE THAN ONCE
2556  006054                                                                IF INTFLAG EQ #0 THEN
2557  006054  005767  002604                  TST     INTFLAG
2558  006060  001002                          BNE     50133$
2559                                                                            ;TRANSMITTER DID NOT INTERRUPT IN TIME
2560  006062                                                                        ERRHRD 106,,DIDNOT
2561  006062  104106                          ERROR   106
2562  006064                                                                    ELSE
2563  006064  000401                          BR      50134$
2564  006066                          50133$:                               ;TWICE
2565                                                                        ;TRANSMITTER INTERRUPTED TWICE
2566                                                                            ERRHRD 107,,TWICE
2567  006066
2568  006066  104107                          ERROR   107
2569  006070                                                                ENDIF
2570  006070                          50134$:
2571  006070                                                            ENDIF
2572  006070                          50132$:
2573  006070                                                        ENDSUB
2574                                                        ;INTERRUPT WITHOUT INTERRUPT ENABLE SET
2575  006070                                                        BGNSUB
2576  006070  012767  006076  173012          MOV     #66$,$LPERR
2577                                                                    ;CLEAR 'INTERRUPT OCCURED' FLAG
2578  006076                                                            LET INTFLAG := #0
2579  006076  005067  002562                  CLR     INTFLAG
2580                                                                    ;CLEAR INTERRUPT ENABLE
2581  006102                                                            LET @TCSR := @TCSR CLR.BY #XMITIE
2582  006102  042777  000100  173154          BIC     #XMITIE,@TCSR
2583                                                                    ;NO INTERRUPTS SHOULD OCCUR.
2584  006110  012746  000000                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
2585  006114  012746  006122                  MOV     #67$,-(SP)      ;;PUT NEW PC ON  STACK
2586  006120  000002                          RTI                     ;;POP NEW PC AND PS
2587  006122                          67$:
```

```
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  002442            JSR     PC,WAIT
2593  006134  012605                    MOV     (SP)+,R5
2594  006136                                                          IF INTFLAG NE #0 THEN
2595  006136  005767  002522            TST     INTFLAG
2596  006142  001401                    BEQ     50135$
2597                                                                      ;INTERRUPT OCCURED WITH I E CLEARED
2598  006144                                                              ERRMRD 110,NOTENAB
2599  006144  104110                    ERROR   110
2600  006146                                                          ENDIF
2601  006146                        50135$:
2602  006146                                                          BRESET
2603  006146  000005                    RESET
2604  006150                                                      ENDSUB
2605                                                                      ;RESTORE VECTOR AREA
2606  006150                                                      LET R4 := @DLVEC
2607  006150  017704  173102            MOV     @DLVEC,R4
2608  006154                                                      CLRVEC R4
2609  006154  010146                    MOV     R1,-(SP)        ;;PUSH R1 ON STACK
2610  006156  010246                    MOV     R2,-(SP)        ;;PUSH R2 ON STACK
2611  006160  012701  000004            MOV     #R4,R1
2612  006164  010102                    MOV     R1,R2
2613  006166  062702  000002            ADD     #2,R2
2614  006172  010221                    MOV     R2,(R1)+
2615  006174  005011                    CLR     (R1)
2616  006176  012602                    MOV     (SP)+,R2        ;;POP STACK INTO R2
2617  006200  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
2618  006202                                                        ENDTST
2619
2620
2621
2622
2623
2624
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 20        RECEIVER INTERRUPT LOGIC TEST
                                ;*              THIS TEST COVERS ALL OF THE RECEIVER
                                ;*              SIDE OF THE INTERRUPT LOGIC IN
                                ;*              CHARACTER MODE.
                                ;;**********************************************************************
006202  000004                  TST20:  SCOPE
006204  012767  000010  172746          MOV     #10.,$TIMES         ;;DO 10 ITERATIONS
006212  012767  000020  172760          MOV     #20.,$TESTN         ;;SET TEST NUMBER IN APT MAIL BOX
006220                                                              IF #MAINTJUMP NOTSETIN $USWR ORB CONSOLE EQ #TRU
006220  032767  040000  172772          BIT     #MAINTJUMP,$USWR
006226  001404                          BEQ     50136$
006230  126727  002671  000001          CMPB    CONSOLE,#TRUE
006236  001002                          BNE     50137$
006240                          50136$:
006240  000167  000242                  JMP     TST21                       ;       EXIT TEST
006244                                                              ENDIF
006244                          50137$:
                                                                            ;CLEAR INTERRUPT OCCURED FLAG
                                                                            ;SET UP RECEIVER INTER.VECTOR
006244                                                              SETVEC DLVEC,#INTSRV,#PR7
006244  010146                          MOV     R1,-(SP)
006246  016701  173004                  MOV     DLVEC,R1
006252  012721  010656                  MOV     #INTSRV,(R1)+
006256  012711  000340                  MOV     #PR7,(R1)
006262  012601                          MOV     (SP)+,R1
                                        ;PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-RCVRIE
006264                                                              BGNSUB
006264  012767  006272  172616          MOV     #64$,$LPERR
006272                                                              LET INTFLAG := #0
006272  005067  002366                  CLR     INTFLAG
                                                                            ;SET MAINT. BIT
006276                                                              LET @TCSR := @TCSR SET.BY #MAINT
006276  052777  000004  172760          BIS     #MAINT,@TCSR
                                                                            ;CLEAR INTERRUPTS
006304                                                                  LET @RCSR := @RCSR CLR.BY #RCVRIE
006304  042777  000100  172746          BIC     #RCVRIE,@RCSR
                                                                            ;CHANGE PRIORITY
                                                                            ;..TO 0
006312  012746  000000                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
006316  012746  006324                  MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
006322  000002                          RTI                     ;;POP NEW PC AND PS
006324                          65$:
                                                                            ;SEND A CHARACTER
006324                                                                  LET @TBUF :B= #0
006324  105077  172740                  CLRB    @TBUF
                                                                            ;WAIT A MAXIMUM
                                                                    ;OF 500 MSEC FOR
                                                                    ;RCVR DONE TO SET IN RCSR
                                                                    CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
006330
006330  010546                          MOV     R5,-(SP)
006332  012745  177777                  MOV     #SET,-(R5)
006336  016745  172716                  MOV     RCSR,-(R5)
```

```
006342 012745 000200                   MOV    #RCVRDONE,-(R5)
006346 012745 000500                   MOV    #500,-(R5)
006352 004767 001742                   JSR    PC,TIMER
006356 012605                          MOV    (SP)+,R5

                                                                   ;SET INTERRUPT ENABLE
006360                                                         LET @RCSR := @RCSR SET.BY #RCVRIE
006360 052777 000100 172672            BIS    #RCVRIE,@RCSR

                                                                   ;LET IT COME IN.
006366                                                         WAITMS 1
006366 010546                          MOV    R5,-(SP)
006370 012745 000001                   MOV    #1,-(R5)
006374 004767 002176                   JSR    PC,WAIT
006400 012605                          MOV    (SP)+,R5


006402                                                         LET R0 := @RBUF ; CLEAR RCVRDONE
006402 017700 172654                   MOV    @RBUF,R0

                                                                   ;DID WE DO IT RIGHT?
006406                                                         IF INTFLAG NE #1 THEN
006406 026727 002252 000001            CMP    INTFLAG,#1
006414 001411                          BEQ    50140$

                                                                   ;NONE OCCURED
                                                                   ; CAN NOT LEAVE WITH MAINT SET
006416                                                         LET    @TCSR := @TCSR CLR.BY #MAINT
006416 042777 000004 172640            BIC    #MAINT,@TCSR
006424                                                             IF INTFLAG EQ #0 THEN
006424 005767 002234                   TST    INTFLAG
006430 001002                          BNE    50141$

                                                                   ;RECEIVER DID NOT INTERRUPT IN TIME
006432                                                                 ERRHRD 111,,DIDNOT
006432 104111                          ERROR  111

                                                                   ;TWICE OR MORE
006434                                                             ELSE
006434 000401                          BR     50142$
006436                          50141$:

                                                                   ;RECEIVER INTERRUPTED TWICE
006436                                                                 ERRHRD 112,,TWICE
006436 104112                          ERROR  112
006440                                                             ENDIF
006440                          50142$:
006440                                                         ENDIF
006440                          50140$:
006440                                                     ENDSUB



                                                                   ; CLEAR THE WORLD
006440                                                     LET @RCSR := @RCSR CLR.BY #RCVRIE
006440 042777 000100 172612            BIC    #RCVRIE,@RCSR



                                                                   ;RESET MAINT. BIT.
006446                                                     LET @TCSR := @TCSR CLR.BY #MAINT
006446 042777 000004 172610            BIC    #MAINT,@TCSR

006454                                                     LET R4 := @DLVEC
006454 017704 172576                   MOV    @DLVEC,R4
006460                                                     CLRVEC R4
```

```
2737  006460  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
2738  006462  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
2739  006464  012701  000004                MOV     #R4,R1
2740  006470  010102                        MOV     R1,R2
2741  006472  062702  000002                ADD     #2,R2
2742  006476  010221                        MOV     R2,(R1)+
2743  006500  005011                        CLR     (R1)
2744  006502  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
2745  006504  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
2746  006506                                                                ENDTST
```

```
2747                                    ;;**********************************************************************
2748                                    ;;**********************************************************************
2749                                    ;*TEST 21        TEST ACTUAL DATA TRANSFERED
2750                                    ;*               NON-INTERRUPT MAINTENANCE BIT SET
2751                                    ;;**********************************************************************
2752  006506  000004                   TST21:  SCOPE
2753  006510  012767  000001  172442           MOV     #1,$TIMES       ;;DO 1 ITERATION
2754  006516  012767  000021  172454           MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2755  006524                                                             IF @#MAINTJUMP NOTSETIN $USWR ORB CONSOLE EQ #TRU
2756  006524  032767  040000  172466           BIT     @#MAINTJUMP,$USWR
2757  006532  001404                           BEQ     50143$
2758  006534  126727  002365  000001           CMPB    CONSOLE,#TRUE
2759  006542  001004                           BNE     50144$
2760  006544                                  50143$:
2761  006544                                                                 EXIT TEST
2762  006544  012767  000001  172406           MOV     #1,$TIMES
2763  006552  000526                           BR      TST22           ;;;EXIT THIS TEST
2764  006554                                                             ENDIF
2765  006554                                  50144$:
2766
2767  006554                                                             LET ERRCHK := #0
2768  006554  005067  001412                   CLR     ERRCHK
2769                                                                     ;SET MAINT. BIT
2770  006560                                                             LET @TCSR := @TCSR SET.BY #MAINT
2771  006560  052777  000004  172476           BIS     #MAINT,@TCSR
2772
2773                                                                     ;CHANGE PRIORITY
2774                                                                     ;..TO 0
2775  006566  012746  000000                   MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
2776  006572  012746  006600                   MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
2777  006576  000002                           RTI                     ;;POP NEW PC AND PS
2778  006600                                  64$:
2779                                                                     ;GET DATA MASK.
2780  006600                                                             CALL DATLNG OUT <R1>
2781  006600  162705  000002                   SUB     #1*2,R5
2782  006604  004767  001666                   JSR     PC,DATLNG
2783  006610  012501                           MOV     (R5)+,R1
2784  006612                                                             LET R0 := @RBUF ; START CLEAR
2785  006612  017700  172444                   MOV     @RBUF,R0
2786
2787                                                                     ;ALL BINARY CHAR.
2788  006616                                                             INCR R2 FROM #0 TO #377 BY #1
2789  006616  005002                           CLR     R2
2790  006620  000401                           BR      50145$
2791  006622                                  50146$:
2792  006622  005202                           INC     R2
2793  006624                                  50145$:
2794  006624  020227  000377                   CMP     R2,#377
2795  006630  003062                           BGT     50147$
2796
2797
2798                                                                 ;TRANSMIT CHAR IN R2
2799
2800  006632                                                         CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
2801  006632  010546                           MOV     R5,-(SP)
2802  006634  012745  177777                   MOV     #SET,-(R5)
```

```
2803  006640  016745  172420                  MOV     TCSR,-(R5)
2804  006644  012745  000200                  MOV     #XMITRDY,-(R5)
2805  006650  012745  000500                  MOV     #500,-(R5)
2806  006654  004767  001440                  JSR     PC,TIMER
2807  006660  012605                          MOV     (SP)+,R5
2808  006662
2809  006662  103003                          BCC     50150$
2810  006664
2811  006664  052767  000010  001300          BIS     #BIT3,ERRCHK
2812  006672
2813  006672                          50150$:
2814
2815
2816  006672
2817  006672  110277  172372                  MOVB    R2,@TBUF
2818
2819  006676
2820  006676  010546                          MOV     R5,-(SP)
2821  006700  012745  177777                  MOV     #SET,-(R5)
2822  006704  016745  172350                  MOV     RCSR,-(R5)
2823  006710  012745  000200                  MOV     #RCVRDONE,-(R5)
2824  006714  012745  000500                  MOV     #500,-(R5)
2825  006720  004767  001374                  JSR     PC,TIMER
2826  006724  012605                          MOV     (SP)+,R5
2827  006726
2828  006726  103003                          BCC     50151$
2829  006730
2830  006730  052767  000020  001234          BIS     #BIT4,ERRCHK
2831  006736
2832  006736                          50151$:
2833
2834  006736
2835  006736  017703  172320                  MOV     @RBUF,R3
2836
2837
2838
2839
2840
2841
2842  006742
2843  006742  010204                          MOV     R2,R4
2844  006744  040104                          BIC     R1,R4
2845  006746
2846  006746  040103                          BIC     R1,R3
2847
2848
2849  006750
2850  006750  020403                          CMP     R4,R3
2851  006752  001410                          BEQ     50152$
2852
2853
2854  006754
2855  006754  042777  000004  172302          BIC     #MAINT,@TCSR
2856  006762
2857  006762  104116                          ERROR   116
2858  006764
```

```
IF.ERROR THEN

    LET ERRCHK := ERRCHK SET.BY #BIT3

ENDIF


    ;TRANSMIT IT
LET @TBUF :B= R2

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




IF.ERROR THEN

    LET ERRCHK := ERRCHK SET.BY #BIT4

ENDIF

    ;AND SAVE IT
LET R3 := @RBUF


;COMPARE TO SEE IF WE RECEIVED IT ALL

    ;CLEAN OFF NON-DATA BITS
    ;ON BOTH TRANSMITTED AND
LET R4 := R2 CLR.BY R1

LET R3 := R3 CLR.BY R1

;RECEIVED DATA
IF R4 NE R3 THEN

    ;DATA COMPARE ERROR
    ; CAN NOT LEAVE WITH MAINT SET
LET    @TCSR := @TCSR CLR.BY #MAINT

    ERRHRD 116,COMP,SBWAS

    EXIT TEST ; ON ERROR
```

```
2859  006764  012767  000001  172166          MOV     #1,$TIMES
2860  006772  000416                          BR      TST22           ;;;EXIT THIS TEST
2861  006774                                                                  ENDIF
2862  006774                          50152$:
2863  006774                                                          ENDINC  ; R2
2864  006774  000712                          BR      50146$
2865  006776                          50147$:
2866
2867                                                                  ;RESET MAINT. BIT.
2868  006776                                                          LET @TCSR := @TCSR CLR.BY @MAINT
2869  006776  042777  000004  172260          BIC     #MAINT,@TCSR
2870  007004                                                          IF @BIT3 SETIN ERRCHK THEN
2871  007004  032767  000010  001160          BIT     #BIT3,ERRCHK
2872  007012  001401                          BEQ     50153$
2873  007014                                                                  ERRHRD 130
2874  007014  104130                          ERROR   130
2875  007016                                                          ENDIF
2876  007016                          50153$:
2877  007016                                                          IF @BIT4 SETIN ERRCHK THEN
2878  007016  032767  000020  001146          BIT     #BIT4,ERRCHK
2879  007024  001401                          BEQ     50154$
2880  007026                                                                  ERRHRD 131
2881  007026  104131                          ERROR   131
2882  007030                                                          ENDIF
2883  007030                          50154$:
2884  007030                                                  ENDTST
2885
2886
2887
```

```
                        ;;*******************************************************************
                        ;;*******************************************************************
                        ;*TEST 22        TEST DATA THROUGH WRAP
                        ;;*******************************************************************
007030  000004          TST22:  SCOPE
007032  012767  000001  172120          MOV     #1,$TIMES       ;;DO 1 ITERATION
007040  012767  000022  172132          MOV     #22.,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
007046                                                                  IF @WRAP NOTSETIN $USWR OR @COMSPD NOTSETIN $USW
007046  032767  020000  172144          BIT     @WRAP,$USWR
007054  001404                          BEQ     50155$
007056  032767  000100  172134          BIT     @COMSPD,$USWR
007064  001004                          BNE     50156$
007066                          50155$:
                                                                        ;CAN'T TEST WITHOUT A WRAP
007066                                                                  EXIT TST
007066  012767  000001  172064          MOV     #1,$TIMES
007074  000516                          BR      TST23           ;;;EXIT THIS TEST
007076                                                                  ENDIF
007076                          50156$:
                                                                        ;DON'T USE MAINT.
007076                                                                  LET @TCSR := @TCSR CLR.BY @MAINT
007076  042777  000004  172160          BIC     @MAINT,@TCSR
                                                                        ; IF A SPECIAL TURN AROUND CARD IS
                                                                        ;   CONNECTED IN PLACE OF THE WRAP
                                                                        ;  SETTING READER RUN WILL ENABLE IT.
                                                                        ; THIS MODULE IS ONLY USED IN MANUFACTUR
                                                                        ; AND ONLY ON THE CONSOLE DLV11-F.
                                                                        ; IF NO SPECIAL MODULE IS AVAILABLE.
                                                                        ; AND THE WRAP BIT IS SET IN $USWR
                                                                        ; THEN THIS TEST WILL ERROR ON THE CONSO
007104                                                                  LET @RCSR := @RCSR SET.BY #11
007104  052777  000011  172146          BIS     #11,@RCSR
                                                                        ;CHANGE PRIORITY
                                                                        ;..TO 0
007112  012746  000000                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
007116  012746  007124                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
007122  000002                          RTI                     ;;POP NEW PC AND PS
007124                          64$:
                                                                        ;GET DATA MASK
007124                                                                  CALL DATLNG OUT <R1>
007124  162705  000002                  SUB     #1+2,R5
007130  004767  001342                  JSR     PC,DATLNG
007134  012501                          MOV     (R5)+,R1
007136                                                                  LET R0 := @RBUF ; START CLEAN
007136  017700  172120                  MOV     @RBUF,R0
                                                                        ;BINARY COUNT PATTERN
007142                                                                  INCR R2 FROM #0 TO #377 BY #1
007142  005002                          CLR     R2
007144  000401                          BR      50157$
007146                          50160$:
007146  005202                          INC     R2
007150                          50157$:
007150  020227  000377                  CMP     R2,#377
007154  003063                          BGT     50161$
```

```
2944
2945                                                    ;TRANSMIT THE CHAR. IN R2.
2946
2947                                                        ; MAKE SURE IT'S READY
2948  007156                                            CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
2949  007156  010546                   MOV     R5,-(SP)
2950  007160  012745  177777           MOV     #SET,-(R5)
2951  007164  016745  172074           MOV     TCSR,-(R5)
2952  007170  012745  000200           MOV     #XMITRDY,-(R5)
2953  007174  012745  000500           MOV     #500,-(R5)
2954  007200  004767  001114           JSR     PC,TIMER
2955  007204  012605                   MOV     (SP)+,R5
2956  007206                                            IF.ERROR THEN
2957  007206  103005                   BCC     50162$
2958  007210  104123                   ERROR   123                  ; TRANSMITTER NEVER BECAME READY
2959  007212                                                       EXIT TEST
2960  007212  012767  000001  171740   MOV     #1,$TIMES
2961  007220  000444                   BR      TST23        ;;;EXIT THIS TEST
2962  007222                                                   ENDIF
2963  007222                       50162$:
2964
2965                                                        ;START IT ON ITS WAY
2966  007222                                            LET @TBUF :B= R2
2967  007222  110277  172042           MOVB    R2,@TBUF
2968                                                    ; NOW WAIT FOR RECIEVER DONE
2969  007226                                            CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
2970  007226  010546                   MOV     R5,-(SP)
2971  007230  012745  177777           MOV     #SET,-(R5)
2972  007234  016745  172020           MOV     RCSR,-(R5)
2973  007240  012745  000200           MOV     #RCVRDONE,-(R5)
2974  007244  012745  000500           MOV     #500,-(R5)
2975  007250  004767  001044           JSR     PC,TIMER
2976  007254  012605                   MOV     (SP)+,R5
2977  007256                                                IF.ERROR THEN
2978  007256  103005                   BCC     50163$
2979  007260                                                    ERRMRD 124
2980  007260  104124                   ERROR   124
2981                                                        ; RECIEVER NEVER BECAME READY
2982  007262                                                EXIT TEST
2983  007262  012767  000001  171670   MOV     #1,$TIMES
2984  007270  000420                   BR      TST23        ;;;EXIT THIS TEST
2985  007272                                                ENDIF
2986  007272                       50163$:
2987
2988                                                        ;RETRIEVE
2989  007272                                            LET R3 := @RBUF
2990  007272  017703  171764           MOV     @RBUF,R3
2991
2992                                                        ;STRIP OFF JUNK ON BOTH
2993  007276                                            LET R4 := R2 CLR.BY R1
2994  007276  010204                   MOV     R2,R4
2995  007300  040104                   BIC     R1,R4
2996  007302                                            LET R3 := R3 CLR.BY R1
2997  007302  040103                   BIC     R1,R3
2998
2999                                                        ;WE HAVE TROUBLE
```

```
007304                                                                  IF R4 NE R3 THEN
007304  020403                          CMP     R4,R3
007306  001405                          BEQ     50164$
                                                                           ;DATA COMPARE ERROR
007310                                                                     ERRHRD 117,COMP,SBWAS
007310  104117                          ERROR   117
007312                                                                     EXIT TEST ; ON ERROR
007312  012767  000001  171640          MOV     #1,$TIMES
007320  000404                          BR      TST23           ;;;EXIT THIS TEST
007322                                                                  ENDIF
007322                          50164$:

007322                                                               ENDINC  ; R2
007322  000711                          BR      50160$
007324                          50161$:
                                                                           ; NOW THAT THE TEST IS DONE
                                                                           ;  WE WILL TOGGLE READER RUN
                                                                           ;  TO TURN OFF THE SPECIAL MODULE.
007324                                                               LET @RCSR := @RCSR SET.BY #11
007324  052777  000011  171726          BIS     #11,@RCSR



007332                                                            ENDTST
```

```
3028                                    ;;*****************************************************************
3029                                    ;;*****************************************************************
3030                                    ;*TEST 23        FULL DATA TRANSFER WITH INTERRUPTS
3031                                    ;*              AND MAINTENANCE MODE.
3032                                    ;;*****************************************************************
3033  007332  000004                    TST23:  SCOPE
3034  007334  012767  000001  171616            MOV     #1,$TIMES       ;;DO 1 ITERATION
3035  007342  012767  000023  171630            MOV     #23.,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
3036                                                                    IF #MAINTJUMP NOTSETIN $USWR ORB CONSOLE EQ #TRU
3037  007350
3038  007350  032767  040000  171642            BIT     #MAINTJUMP,$USWR
3039  007356  001404                            BEQ     50165$
3040  007360  126727  001541  000001            CMPB    CONSOLE,#TRUE
3041  007366  001004                            BNE     50166$
3042  007370                            50165$:
3043  007370                                                            EXIT TEST
3044  007370  012767  000001  171562            MOV     #1,$TIMES
3045  007376  000553                            BR      TST24           ;;;EXIT THIS TEST
3046  007400                                                            ENDIF
3047  007400                            50166$:
3048
3049                                                                    ;GET DATA MASK
3050  007400                                                            CALL DATLNG OUT <R3>
3051  007400  162705  000002                    SUB     #1*2,R5
3052  007404  004767  001066                    JSR     PC,DATLNG
3053  007410  012503                            MOV     (R5)+,R3
3054
3055
3056                                    ; THIS TEST WILL RUN BOTH TRANSMITTER AND
3057                                    ; RECIEVER AT FULL SPEED TESTING
3058                                    ; THE ABILITY OF THE MODULE
3059                                    ; TO HANDLE INTERRUPTS FROM BOTH SIDES
3060                                    ; AT ONCE. ALSO, THE DOUBLE BUFFERING LOGIC
3061                                    ; OF THE UART WILL BE FULLY TESTED.
3062                                    ;  THIS TEST WILL TRANSFER A MAXIMUM OF 400(8)
3063                                    ; CHARACTERS THROUGH THE MODULE, BUT IF AN ERROR
3064                                    ; IS DETECTED BY THE TEST A PREMATURE SHUTDOWN OCCURS.
3065
3066                                                                    ;CHANGE PRIORITY
3067                                                                    ;..TO 0
3068  007412  012746  000000                    MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
3069  007416  012746  007424                    MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
3070  007422  000002                            RTI                     ;;POP NEW PC AND PS
3071  007424                            64$:
3072                                                                    ;GET VECTOR ADDRESS
3073  007424                                                            LET R1 := DLVEC
3074  007424  016701  171626                    MOV     DLVEC,R1
3075                                                                    ;RCVR VECTOR
3076  007430                                                            LET (R1)+ := #REC
3077  007430  012721  007632                    MOV     #REC,(R1)+
3078  007434                                                            LET (R1)+ := #PR7
3079  007434  012721  000340                    MOV     #PR7,(R1)+
3080                                                                    ;POINT TO TRANSMITTER VECTOR
3081                                                                    ;AND SET IT UP ALSO
3082  007440                                                            LET (R1)+ := #TRAN
3083  007440  012721  007570                    MOV     #TRAN,(R1)+
```

```
3084  007444                                                    LET (R1)  := @PR7
3085  007444  012711  000340          MOV     @PR7,(R1)
3086
3087                                                                  ; CLEAR ERROR COUNTER
3088  007450                                                    LET ERRCNT := @0
3089  007450  005067  000106          CLR     ERRCNT
3090                                                                  ; INITIALIZE COUNTERS
3091  007454                                                    LET R1 := @-1
3092  007454  012701  177777          MOV     @-1,R1
3093                                                                  ;RECEIVER STORAGE
3094  007460                                                    LET R2 := @0
3095  007460  005002                  CLR     R2
3096                                                                  ;# OF RECEIVED CHAR. COUNT.
3097  007462                                                    LET R4 := @-1
3098  007462  012704  177777          MOV     @-1,R4
3099
3100  007466                                                    @RESET  ;SET UP ALL REGISTERS
3101  007466  000005                  RESET
3102                                                                  ;SET UP MAINTENANCE
3103  007470                                                    LET @TCSR := @TCSR SET.BY @MAINT
3104  007470  052777  000004  171566  BIS     @MAINT,@TCSR
3105
3106                                                                  ;SET I.E. IN TRANSMITTER
3107  007476                                                    LET @TCSR := @TCSR SET.BY @XMITIE
3108  007476  052777  000100  171560  BIS     @XMITIE,@TCSR
3109                                                                  ;AND RECEIVER
3110  007504                                                    LET @RCSR := @RCSR SET.BY @RCVRIE
3111  007504  052777  000100  171546  BIS     @RCVRIE,@RCSR
3112
3113
3114                                                            ;NOW WE WAIT UNTIL R4 COUNT (RECEIVED) IS EQUAL
3115  007512                                                    REPEAT
3116  007512                          50167$:
3117  007512                                                    UNTIL R4 EQ NUMBER OR ERRCNT GT @0
3118  007512  020467  000046          CMP     R4,NUMBER
3119  007516  001403                  BEQ     50170$
3120  007520  005767  000036          TST     ERRCNT
3121  007524  003772                  BLE     50167$
3122  007526                          50170$:
3123
3124  007526                                                    LET @TCSR := @TCSR CLR.BY @MAINT
3125  007526  042777  000004  171530  BIC     @MAINT,@TCSR
3126                                                                  ; CHECK FOR DATA COMPARE ERRORS.
3127  007534                                                    IF ERRCNT NE @0 THEN
3128  007534  005767  000022          TST     ERRCNT
3129  007540  001401                  BEQ     50171$
3130                                                                  ;DATA COMPARE ERROR
3131  007542                                                        ERRHRD 120,COMP,FIRST
3132  007542  104120                  ERROR   120
3133  007544                                                    ENDIF
3134  007544                          50171$:
3135
3136  007544                                                    LET @TCSR := @TCSR CLR.BY @XMITIE
3137  007544  042777  000100  171512  BIC     @XMITIE,@TCSR
3138  007552                                                    LET @RCSR := @RCSR CLR.BY @RCVRIE
3139  007552  042777  000100  171500  BIC     @RCVRIE,@RCSR
```

```
                                                                EXIT    ;SKIP OVER SUPPORT ROUTINES & STORAGE
007560  000462                          BR      TST24           ;;;EXIT THIS TEST

007562  000000                  ERRCNT: 0
007564  001000                  NUMBER: 1000
007566     000                  SB:     .BYTE 0
007567     000                  WAS:    .BYTE 0


                        ;;*************************************************************************
                                                ;TRANSMIT INTERRUPT HANDLER
                                                        BGNSRV  TRAN
007570                  TRAN:
                        ;;*************************************************************************

                                                        ;INCREMENT CHAR COUNT
                                                LET R1 := R1 + #1
007570  005201                  INC     R1
                                                        ;SET UP FOR TRANSFER
                                                LET HOLD := R1 CLR.BY R3
007572  010167  000030          MOV     R1,HOLD
007576  040367  000024          BIC     R3,HOLD
                                                        ;AND SEND.
                                                LET @TBUF := HOLD
007602  016777  000020  171460  MOV     HOLD,@TBUF
                                                        ;ALL DONE
                                                IF R1 EQ NUMBER THEN
007610  020167  177750          CMP     R1,NUMBER
007614  001003                  BNE     50172$
                                                    ;STOP   INTERRUPT PROCESSING
                                                    LET @TCSR := @TCSR CLR.BY #XMITIE
007616  042777  000100  171440      BIC     #XMITIE,@TCSR
                                                ENDIF
007624                  50172$:
007624  000401                  BR      ZZZ                     ; EXIT SRV

007626  000000              HOLD:0

007630                  ZZZ:                                    ENDSRV
007630  000002                  RTI


                        ;;*************************************************************************
                                                ;RECEIVER INTERRUPT HANDLER
                                                        BGNSRV  REC
007632                  REC:
                        ;;*************************************************************************

                                                        ;COUNT THIS CHAR.
                                                LET R4 := R4 + #1
007632  005204                  INC     R4
                                                        ;GET CHAR IN + MASK IT
                                                LET R2 := @RBUF CLR.BY R3
007634  017702  171422          MOV     @RBUF,R2
```

```
007640  040302                          BIC     R3,R2
                                                                        ;RHLD WILL CONTAIN EXPECTED INPUT
007642                                                          LET RHLD := R4 CLR.BY R3
007642  010467  000054                  MOV     R4,RHLD
007646  040367  000050                  BIC     R3,RHLD

                                                                        ;DO THEY COMPARE
007652                                                          IF R2 NE RHLD THEN
007652  020267  000044                  CMP     R2,RHLD
007656  001412                          BEQ     50173$
                                                                        ;FIRST ERROR
007660                                                                  IF ERRCNT EQ #0 THEN
007660  005767  177676                  TST     ERRCNT
007664  001005                          BNE     50174$
                                                                        ;SAVE RECORD OF FIRST MISS
007666                                                                      LET SB :B= RHLD
007666  116767  000030  177672          MOVB    RHLD,SB
007674                                                                      LET WAS :B= R2
007674  110267  177667                  MOVB    R2,WAS
007700                                                                  ENDIF
007700                          50174$:
                                                                        ;COUNT IT.
007700                                                                  LET ERRCNT := ERRCNT + #1
007700  005267  177656                  INC     ERRCNT
007704                                                          ENDIF
007704                          50173$:

                                                                        ;ALL DONE?
007704                                                          IF R4 EQ NUMBER THEN
007704  020467  177654                  CMP     R4,NUMBER
007710  001003                          BNE     50175$
                                                                        ;STOP RECEIVER INTERRUPTS
007712                                                                  LET @RCSR := @RCSR CLR.BY #RCVRIE
007712  042777  000100  171340          BIC     #RCVRIE,@RCSR
                                                                                ; INDICATE ALL DONE TO TIMER
                                                                        ;MAIN REPEAT LOOP IS CHECKING
                                                                        ;FOR 'R4 = NUMBER' ALSO
007720                                                          ENDIF
007720                          50175$:

007720  000401                          BR      ZZZZ                  ;  EXIT SRV

007722  000000                                                  RHLD:0
007724                          ZZZZ:
007724                                                       ENDSRV
007724  000002                          RTI

007726                                                  ENDTST
```

```
3247                                    ;;**********************************************************************
3248                                    ;;**********************************************************************
3249                                    ;*TEST 24        TEST BREAK GENERATION LOGIC
3250                                    ;*              TRANSMIT KNOWN CHAR WITH BREAK SET
3251                                    ;*              AND COMPARE RECEIVED WITH 0.
3252                                    ;*              FRAMING ERROR WILL ALSO BE CHECKED
3253                                    ;*              IF ERROR BITS ARE ENABLED.
3254                                    ;;**********************************************************************
3255  007726  000004                    TST24:  SCOPE
3256  007730  012767  000010  171222            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
3257  007736  012767  000024  171234            MOV     #24,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3258  007744                                                                    IF #MAINTJUMP NOTSETIN $USWR OR #BRK NOTSETIN $U
3259  007744  032767  040000  171246            BIT     #MAINTJUMP,$USWR
3260  007752  001404                            BEQ     50176$
3261  007754  032767  010000  171236            BIT     #BRK,$USWR
3262  007762  001004                            BNE     50177$
3263  007764                                50176$:
3264  007764                                                                        EXIT TEST
3265  007764  012767  000001  171166            MOV     #1,$TIMES
3266  007772  000500                            BR      TST25           ;;;EXIT THIS TEST
3267  007774                                                                    ENDIF
3268  007774                                50177$:
3269  007774                                                                    IFB CONSOLE EQ #TRUE THEN
3270  007774  126727  001125  000001            CMPB    CONSOLE,#TRUE
3271  010002  001004                            BNE     50200$
3272  010004                                                                        EXIT TEST
3273  010004  012767  000001  171146            MOV     #1,$TIMES
3274  010012  000470                            BR      TS125           ;;;EXIT THIS TEST
3275  010014                                                                    ENDIF
3276  010014                                50200$:
3277
3278  010014
3279  010014  005067  000152                    CLR     ERRCHK                  LET ERRCHK := #0        ; CLEAR ERROR WORD
3280
3281  010020                                                                        ;SET MAINTENANCE BIT
3282  010020  052777  000004  171236            BIS     #MAINT,@TCSR            LET @TCSR := @TCSR SET.BY #MAINT
3283
3284  010026                                                                        ;SET BREAK BIT
3285  010026  052777  000001  171230            BIS     #BREAK,@TCSR            LET @TCSR := @TCSR SET.BY #BREAK
3286
3287  010034                                                                        ;NON-ZERO CHAR. '*'
3288  010034  012777  000252  171226            MOV     #252,@TBUF              LET @TBUF := #252
3289
3290  010042                                                                        ;WAIT FOR DONE
3291  010042  010546                            MOV     R5,-(SP)                CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
3292  010044  012745  177777                    MOV     #SET,-(R5)
3293  010050  016745  171204                    MOV     RCSR,-(R5)
3294  010054  012745  000200                    MOV     #RCVRDONE,-(R5)
3295  010060  012745  000500                    MOV     #500,-(R5)
3296  010064  004767  000230                    JSR     PC,TIMER
3297  010070  012605                            MOV     (SP)+,R5
3298  010072                                                                    IF.ERROR THEN
3299  010072  103001                            BCC     50201$
3300                                                                            ; RECIEVER DONE DID NOT SET
3301  010074                                                                    ERRHRD 115
3302  010074  104115                            ERROR   115
```

```
010076                                                          ENDIF
010076                              50201$:
010076                                                          IFB @RBUF NE #0 THEN
010076  105777  171160                      TSTB    @RBUF
010102  001404                              BEQ     50202$
                                                                        ; BREAK DID NOT EQUAL 0
010104                                                             LET ERRCHK := ERRCHK SET.BY #BIT0
010104  052767  000001  000060              BIS     #BIT0,ERRCHK
010112                                                          ELSE
010112  000413                              BR      50203$
010114                              50202$:
010114                                                          IF #ERRBITS SETIN $USWR THEN
010114  032767  100000  171076              BIT     #ERRBITS,$USWR
010122  001407                              BEQ     50204$
010124                                                                  IF #FRERR NOTSETIN @RBUF THEN
010124  032777  020000  171130              BIT     #FRERR,@RBUF
010132  001003                              BNE     50205$
010134                                                                     LET ERRCHK := ERRCHK SET.BY #BIT1
010134  052767  000002  000030              BIS     #BIT1,ERRCHK
010142                                                                  ENDIF
010142                              50205$:
010142                                                          ENDIF
010142                              50204$:
010142                                                          ENDIF
010142                              50203$:

010142                                                          BRESET  ;CLEAN UP
010142  000005                              RESET

010144                                                          IF #BIT0 SETIN ERRCHK THEN
010144  032767  000001  000020              BIT     #BIT0,ERRCHK
010152  001401                              BEQ     50206$
010154                                                             ERRHRD 121   ;BREAK ERROR
010154  104121                              ERROR   121
010156                                                          ENDIF
010156                              50206$:
010156                                                          IF #BIT1 SETIN ERRCHK THEN
010156  032767  000002  000006              BIT     #BIT1,ERRCHK
010164  001401                              BEQ     50207$
010166                                                             ERRHRD 122   ; FRAMING ERROR
010166  104122                              ERROR   122
010170                                                          ENDIF
010170                              50207$:
010170                                                          EXIT
010170  000401                              BR      TST25              ;;;EXIT THIS TEST
010172  000000                      ERRCHK: .WORD 0
010174                                                     ENDTST
```

```
3352
3353                                    ;;*******************************************************************
3354                                    ;*TEST 25        NOT A TEST - SEND BACK TO LOOP
3355                                    ;;*******************************************************************
3356  010174  000004                   TST25:  SCOPE
3357  010176  012767  000001  170754           MOV     #1,$TIMES       ;;DO 1 ITERATION
3358  010204  104401  010212                   TYPE    ,65$            ;;TYPE ASCIZ STRING
3359  010210  000404                           BR      64$             ;;GET OVER THE ASCIZ
3360                                    ;;65$:  .ASCIZ  <CRLF>*CSR: *
3361  010222                            64$:
3362  010222  016746  171026                   MOV     DLADD,-(SP)     ;;SAVE DLADD FOR TYPEOUT
3363  010226  104402                           TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3364  010230  104401  010236                   TYPE    ,67$            ;;TYPE ASCIZ STRING
3365  010234  000405                           BR      66$             ;;GET OVER THE ASCIZ
3366                                    ;;67$:  .ASCIZ  *,VECTOR: *
3367  010250                            66$:
3368  010250  016746  171002                   MOV     DLVEC,-(SP)     ;;SAVE DLVEC FOR TYPEOUT
3369  010254  104402                           TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3370  010256  104401  010264                   TYPE    ,69$            ;;TYPE ASCIZ STRING
3371  010262  000405                           BR      68$             ;;GET OVER THE ASCIZ
3372                                    ;;69$:  .ASCIZ  *,ERRORS: *
3373  010276                            68$:
3374  010276  016746  170610                   MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
3375  010302  104405                           TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
3376  010304  005067  170602                   CLR     $ERTTL          ; RESET FOR NEXT DEVICE/PASS
3377  010310  104401  001171                   TYPE    ,$CRLF
3378  010314  000167  171616                   JMP     LOOP            ; BACK UP TO THE BEGINNING
```

```
3379
3380                                    ;;BGNMOD        SUBS
3381                                    ;;*************************************************************************
3382  010320                            ROUTINE TIMER <HOWLONG,WHICHBIT,REG,SETCLR>
3383  010320                            TIMER:
3384                                    ;* ROUTINE:TIMER
3385                                    ;*      THIS ROUTINE IS USED TO TEST THE STATUS OF ANY BIT
3386                                    ;*      IN ANY REGISTER.
3387                                    ;*      INPUTS:
3388                                    ;*          HOWLONG    THE MAXIMUM AMOUNT OF TIME TO SPEND IN
3389                                    ;*                     THIS ROUTINE.
3390                                    ;*          WHICHBIT     A MASK WITH THE BIT(S) SET THAT ARE
3391                                    ;*                       TO BE CHECKED.
3392                                    ;*          REG      A POINTER TO THE REGISTER TO BE CHECKED
3393                                    ;*          SETCLR   THE DESIRED RESULTS
3394                                    ;*                   EITHER @SET OR @CLEAR
3395                                    ;*      OUTPUT:
3396                                    ;*          THE 'C' BIT IS SET TO INDICATE AN ERROR
3397                                    ;*          BUT IT IS TESTED BY THE IF.ERROR STATEMENT
3398                                    ;*
3399                                    ;*
3400                                    ;*          NOTE::  THE USE OF (R5) IS PART OF THE LINKAGE
3401                                    ;*                  MECHANISM BETWEEN THE CALLER AND THE CALLED
3402                                    ;;*************************************************************************
3403                                    .ENABL LSB
3404          000001                                                    TRUE=  1
3405          000000                                                    FALSE=  0
3406
3407  010320
3408  010320  016567  000004  000136            MOV     REG(R5),REGSAV          LET     REGSAV := REG(R5) ; GET POINTER TO REGIST
3409  010326
3410  010326  016567  000000  000132            MOV     HOWLONG(R5),TIMSAV      LET     TIMSAV := HOWLONG(R5) ; SAVE HOWLONG FOR
3411  010334
3412  010334  112767  000000  000126            MOVB    @FALSE,FLAG             LET     FLAG :B= @FALSE ; INITIALIZE THE EXIT FLA
3413
3414
3415                                                                            ;       START OF AN INFINITE LOOP
3416
3417  010342                                                                    LOOP
3418  010342                            50002$:
3419                                                                                ; TEST TO SEE IF WHICHBIT IS SET
3420  010342                                                                        IF      WHICHBIT(R5)  NOTSETIN @REGSAV THEN
3421  010342  036577  000002  000114            BIT     WHICHBIT(R5),@REGSAV
3422  010350  001004                            BNE     50004$
3423  010352                                                                        LET     HOLDSC :B= @CLR
3424  010352  112767  000000  000111            MOVB    @CLR,HOLDSC
3425  010360                                                                    ELSE
3426  010360  000403                            BR      50005$
3427  010362                            50004$:
3428  010362                                                                        LET     HOLDSC :B= @SET ; REMEMBER THIS
3429  010362  112767  177777  000101            MOVB    @SET,HOLDSC
3430  010370                                                                    ENDIF
3431  010370                            50005$:
3432                                                                            ; NOW SEE IF THAT WAS WHAT WE WANTED
3433                                                                            IFB     HOLDSC EQ SETCLR(R5) THEN
3434  010370
```

```
3435  010370  126765  000075  000006          CMPB    HOLDSC,SETCLR(R5)
3436  010376  001003                          BNE     50006$
3437                                                                  ; JUST THE THING WE NEEDED
3438  010400                                                              LET     FLAG :B= #TRUE
3439  010400  112767  000001  000062          MOVB    #TRUE,FLAG
3440  010406                                                            ENDIF
3441  010406                          50006$:
3442
3443  010406                                                          EXIFB   FLAG EQ #TRUE OR TIMSAV LE #0
3444  010406  126727  000056  000001          CMPB    FLAG,#TRUE
3445  010414  001414                          BEQ     50003$
3446  010416  005767  000044                  TST     TIMSAV
3447  010422  003411                          BLE     50003$
3448                                                                  ; ONE WAY OR THE OTHER, WE ARE DONE
3449                                                              ; IF WE ARE STILL HERE THEN HANG AROUND A WHILE
3450
3451  010424                                                              WAITMS  1       ;WAIT FOR 1 MILLI-SECONDS
3452  010424  010546                          MOV     R5,-(SP)
3453  010426  012745  000001                  MOV     #1,-(R5)
3454  010432  004767  000140                  JSR     PC,WAIT
3455  010436  012605                          MOV     (SP)+,R5
3456  010440                                                              LET     TIMSAV := TIMSAV - #1 ; COUNTING DOWN
3457  010440  005367  000022                  DEC     TIMSAV
3458  010444                                                          ENDLOOP                 ; CONTINUED AT THE TOP
3459  010444  000736                          BR      50002$
3460  010446                          50003$:
3461
3462                                          ; ONLY 2 WAYS TO GET HERE
3463                                          ;       1). WE RAN OUT OF TIME---ERROR ! !
3464                                          ;       2). THE BIT IS IN THE CORRECT CONDITION--GOOD ! !
3465
3466  010446                                                          IFB     FLAG EQ #TRUE THEN
3467  010446  126727  000016  000001          CMPB    FLAG,#TRUE
3468  010454  001001                          BNE     50007$
3469  010456                                                              RETURN  NO.ERROR    ; GOOD
3470  010456  000405                          BR      50000$
3471  010460                                                          ENDIF
3472  010460                          50007$:
3473  010460                                                          RETURN  ERROR           ; BAD
3474  010460  000261                          SEC
3475  010462  000404                          BR      50001$
3476
3477  010464  000000                                                  REGSAV: .WORD 0
3478  010466  000000                                                  TIMSAV: .WORD 0
3479  010470     000                                                  FLAG:   .BYTE 0
3480  010471     000                                                  HOLDSC: .BYTE 0
3481                                                          ; WE ARE DONE  GO BACK HOME
3482  010472                                                              ENDRTN
3483  010472                          50000$:
3484  010472  000241                          CLC
3485  010474                          50001$:
3486  010474  000207                          RTS     PC
3487                                  .DSABL LSB
```

```
3488
3489                                    ;;*****************************************************************
3490  010476                            ROUTINE DATLNG <MASK>
3491  010476                            DATLNG:
3492                                    ;* ROUTINE:DATLNG
3493                                    ;*       THIS ROUTINE SETS UP A MASK FOR DATA, WITH
3494                                    ;*  INPUT - NOTHING IS PASSED TO THIS ROUTINE
3495                                    ;*      BUT GLOBAL INFORMATION IS ASSUMED TO EXIST:
3496                                    ;*      $USWR-- THE WORD FOR SOFTWARE PARAMETERS
3497                                    ;*      DATA-- A MASK FOR THE LOCATION OF THE OCTAL
3498                                    ;*               NUMBER OF DATA BITS
3499                                    ;*  OUTPUT----
3500                                    ;*      MASK-- A MASK OF BINARY ONES RIGHT-JUSTIFIED
3501                                    ;*             THE NUMBER OF WHICH IS DEFINED IN $USWR WORD.
3502
3503                                    ;;*****************************************************************
3504                                    .ENABL LSB
3505  010476                                                                    LET MASK(R5) := #0              ; START
3506  010476  005065  000000                   CLR     MASK(R5)
3507  010502                                                                    LET NUMBR := $USWR AND #DATA
3508  010502  016767  170512  000062           MOV     $USWR,NUMBR
3509  010510  016746  000056                   MOV     NUMBR,-(SP)
3510  010514  042716  000017                   BIC     #DATA,(SP)
3511  010520  042667  000046                   BIC     (SP)+,NUMBR
3512
3513  010524                                                                    INCR I FROM #1 TO NUMBR BY #1
3514  010524  012767  000001  170540           MOV     #1,I
3515  010532  000402                           BR      50002$
3516  010534                            50003$:
3517  010534  005267  170532                   INC     I
3518  010540                            50002$:
3519  010540  026767  170526  000024           CMP     I,NUMBR
3520  010546  003006                           BGT     50004$
3521  010550                                                                        LET MASK(R5) := MASK(R5) SHIFT 1
3522  010550  006365  000000                   ASL     MASK(R5)
3523  010554                                                                        LET MASK(R5) := MASK(R5) SET.BY #1
3524  010554  052765  000001  000000           BIS     #1,MASK(R5)
3525  010562                                                                    ENDINC
3526  010562  000764                           BR      50003$
3527  010564                            50004$:
3528  010564                                                                    LET MASK(R5) := COMP MASK(R5)
3529  010564  005165  000000                   COM     MASK(R5)
3530  010570                                                                RETURN
3531  010570  000401                           BR      50000$
3532  010572  000000                           NUMBR:0
3533  010574                                                            ENDRTN
3534  010574                            50000$:
3535  010574                            50001$:
3536  010574  000207                           RTS     PC
3537                                    .DSABL LSB
```

```
                                ;;**********************************************************************
                                ROUTINE WAIT    <TIME>
                                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.
                                ;;**********************************************************************
010576  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
010600  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
010602  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
010604                                  LET R1 := TIME(R5)
010604  016501  000000                  MOV     TIME(R5),R1
010610                                  INCRU R2 FROM #1 TO R1 BY #1
010610  012702  000001                  MOV     #1,R2
010614  000402                          BR      50002$
010616                          50003$:
010616  062702  000001                  ADD     #01,R2
010622                          50002$:
010622  020201                          CMP     R2,R1
010624  101010                          BHI     50004$
010626                                  INCR R3 FROM #0 TO #100 BY #1
010626  005003                          CLR     R3
010630  000401                          BR      50005$
010632                          50006$:
010632  005203                          INC     R3
010634                          50005$:
010634  020327  000100                  CMP     R3,#100
010640  003001                          BGT     50007$
010642                                  ENDINC
010642  000773                          BR      50006$
010644                          50007$:
010644                                  ENDINC
010644  000764                          BR      50003$
010646                          50004$:
010646  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
010650  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
010652  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
010654                          ENDRTN
010654                          50000$:
010654                          50001$:
010654  000207                          RTS     PC
```

```
3583
3584                                    .SBTTL  INTSRV  INTERRUPT SERVICE ROUTINE
3585                                    ;;**********************************************************************
3586  010656                            INTSRV:
3587                                    ;* SERVICE ROUTINE: INTSRV
3588                                    ;*       THIS GLOBAL ROUTINE DOES NOTHING BUT INCREMENT
3589                                    ;*       'INTFLAG'  EACH TIME IT IS CALLED.  IT ASSUMES
3590                                    ;*       THAT THE MAIN CALLING ROUTINE WILL  KNOW  WHAT
3591                                    ;*       TO LOOK FOR.
3592                                    ;;**********************************************************************
3593
3594                                                    ;ADD 1 TO 'INTERRUPT OCCURED' FLAG
3595  010656                                                            LET INTFLAG := INTFLAG + #1
3596  010656  005267  000002                   INC     INTFLAG
3597  010662                                    ENDSRV                  ;THAT'S ALL
3598  010662  000002                           RTI
3599  010664  000000                   INTFLAG:  0
```

```
                                ROUTINE CYCLE
                                CYCLE:
                                ;;*****************************************************************
                                ;* 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. THE CONSOLE IS TREATED SPECIAL BY THIS ROUTINE.
                                ;*      IT IS ONLY TESTED ONCE IF UNDER APT. IF NOT UNDER APT
                                ;*      ALL TESTS THAT REQUIRE THE MAINT BIT ARE NOT RUN.
                                ;;*****************************************************************
                                .ENABL LSB
                                                                LET APTCON :B= #FALSE   ; SET DEFAULT VALUE
010666 112767 000000 000230             MOVB    #FALSE,APTCON
                                                                LET CONSOLE :B= #FALSE
010674 112767 000000 000223             MOVB    #FALSE,CONSOLE
                                                              REPEAT        ; UNTIL BITMASK SETIN $DEVM
                                50002$:
                                                               IF BITMASK EQ #0 THEN
010702 005767 000200                    TST     BITMASK
010706 001027                           BNE     50003$
                                                                  IF INITFLAG EQ #1 THEN
010710 026727 000174 000001             CMP     INITFLAG,#1
010716 001003                           BNE     50004$
                                                                     LET INITFLAG := #0
010720 005067 000164                    CLR     INITFLAG
                                                                  ELSE
010724 000403                           BR      50005$
                                50004$:
                                                                     CALL $EOP    ; AS A SUBROUTINE
010726 004767 000370                    JSR     PC,$EOP
                                SPECIALADDRESS:                 ; BECAUSE $EOP RETURNS AS A JUMP
                                                                     LET R0 := POP
010732 012600                           MOV     (SP)+,R0
                                                                  ENDIF
                                50005$:
                                                                  LET BITMASK := #1
010734 012767 000001 000144             MOV     #1,BITMASK
                                                                  LET $DEVCT := #1
010742 012767 000001 170234             MOV     #1,$DEVCT
                                                                  LET ADDRESS := $BASE
010750 016767 170274 000134             MOV     $BASE,ADDRESS
                                                                  LET VECTOR := $VECT1
010756 016767 170262 000130             MOV     $VECT1,VECTOR
                                                               ELSE
010764 000410                           BR      50006$
                                50003$:
                                                                  LET R4 := #10
010766 012704 000010                    MOV     #10,R4
                                                                  LET BITMASK := BITMASK ROTATE 1
010772 006167 000110                    ROL     BITMASK
                                                                  LET ADDRESS  := ADDRESS + R4
010776 060467 000110                    ADD     R4,ADDRESS
                                                                  LET VECTOR := VECTOR + R4
011002
```

```
 3656  011002  060467  000106            ADD     R4,VECTOR
 3657  011006                                                    ENDIF
 3658  011006                        50006$:
 3659  011006                                                  UNTIL   BITMASK SETIN $DEVM
 3660  011006  036767  000074  170236    BIT     BITMASK,$DEVM
 3661  011014  001732                    BEQ     50002$
 3662  011016                                              IF BITMASK EQ #BIT15 THEN
 3663  011016  026727  000064  100000    CMP     BITMASK,#BIT15
 3664  011024  001023                    BNE     50007$
 3665  011026                                                  LET CONSOLE :B= #TRUE
 3666  011026  112767  000001  000071    MOVB    #TRUE,CONSOLE
 3667  011034                                                  LET ADDRESS := CONADR
 3668  011034  016767  000060  000050    MOV     CONADR,ADDRESS
 3669  011042                                                  LET VECTOR  := CONVECT
 3670  011042  016767  000054  000044    MOV     CONVECT,VECTOR
 3671                          ;;;;;;    IF #CONMAINT NOTSETIN $USWR THEN
 3672                          ;;;;;;        LET NOCONMAINT :B= #TRUE
 3673                          ;;;;;;    ENDIF
 3674  011050                            IF #APTENV SETIN $ENV THEN ; APT MODE
 3675  011050  032767  000001  170136    BIT     #APTENV,$ENV
 3676  011056  001406                    BEQ     50010$
 3677  011060                                          IF $PASS NE #0 THEN  ; NOT FIRST PASS
 3678  011060  005767  170116            TST     $PASS
 3679  011064  001403                    BEQ     50011$
 3680                                                          ; DEFINE DEVICE AS APT CONSOLE
 3681  011066                                              LET APTCON :B= #TRUE
 3682  011066  112767  000001  000030    MOVB    #TRUE,APTCON
 3683  011074                                          ENDIF        ; FIRST PASS
 3684  011074                        50011$:
 3685  011074                                      ENDIF   ; APT
 3686  011074                        50010$:
 3687  011074                                  ENDIF               ; BITMASK
 3688  011074                        50007$:
 3689
 3690  011074                                  LET ADRS := #ADDRESS
 3691  011074  012701  011112            MOV     #ADDRESS,ADRS
 3692  011100                                LET $DEVCT := $DEVCT + #1
 3693  011100  005267  170100            INC     $DEVCT
 3694  011104                                RETURN
 3695  011104  000411                    BR      50000$
 3696  011106  100000            BITMASK:        100000                  ; CONSOLE FIRST
 3697  011110  000001            INITFLAG:       1
 3698  011112  000000            ADDRESS:        0
 3699  011114  000000            VECTOR:         0
 3700  011116  000000            OK:             0
 3701  011120  177560            CONADR:         177560                  ; CONSOLE ADDRESS
 3702  011122  000060            CONVECT:        60                      ; CONSOLE VECTOR
```

```
 3703  011124      000                 APTCON:         .BYTE 0
 3704  011125      000                 CONSOLE:        .BYTE 0
 3705  011126      000                 NOCONMANT:      .BYTE 0
 3706            011130                          .EVEN
 3707
 3708  011130                                                  ENDRTN
 3709  011130                          50000$:
 3710  011130                          50001$:
 3711  011130  000207                         RTS     PC
 3712
 3713                                  .DSABL LSB
```

```
                                ROUTINE MYTYPE
                                MYTYPE:
                                ;;*****************************************************************************
011132  104401  011140              TYPE    .65$                    ;;TYPE ASCIZ STRING
011136  000405                      BR      64$                     ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF>*TEST # *
                                64$:
011152  016746  170022              MOV     $TESTN,-(SP)            ;;SAVE $TESTN FOR TYPEOUT
011156  104402                      TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
011160  104401  011166              TYPE    .67$                    ;;TYPE ASCIZ STRING
011164  000405                      BR      66$                     ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  *,ERROR # *
                                66$:
011200  116767  167710  167770      MOVB    $ITEMB,$FATAL           ; APT FATAL ERROR NUMBER
011206  016746  167764              MOV     $FATAL,-(SP)            ;;SAVE $FATAL FOR TYPEOUT
011212  104403                      TYPOS                           ;;GO TYPE--OCTAL ASCII
011214     006                      .BYTE   6                       ;;TYPE 6 DIGITS
011215     000                      .BYTE   0                       ;;SUPPRESS LEADING ZEROS
011216  104401  011224              TYPE    .69$                    ;;TYPE ASCIZ STRING
011222  000404                      BR      68$                     ;;GET OVER THE ASCIZ
                                ;;69$:  .ASCIZ  *,PC = *
                                68$:
011234  016746  167656              MOV     $ERRPC,-(SP)            ;;SAVE $ERRPC FOR TYPEOUT
011240  104402                      TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
011242  104401  011250              TYPE    .71$                    ;;TYPE ASCIZ STRING
011246  000404                      BR      70$                     ;;GET OVER THE ASCIZ
                                ;;71$:  .ASCIZ  *,CSR: *
                                70$:
011260  016746  167770              MOV     DLADD,-(SP)             ;;SAVE DLADD FOR TYPEOUT
011264  104402                      TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
011266  104401  011274              TYPE    .73$                    ;;TYPE ASCIZ STRING
011272  000405                      BR      72$                     ;;GET OVER THE ASCIZ
                                ;;73$:  .ASCIZ  *,VECTOR: *
                                72$:
011306  016746  167744              MOV     DLVEC,-(SP)             ;;SAVE DLVEC FOR TYPEOUT
011312  104402                      TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
011314  104401  001171              TYPE    .$CRLF
                                ENDRTN
                                50000$:
                                50001$:
011320  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

011322                  $EOP:
011322  000004                  SCOPE
011324  005067  167552          CLR     $TSTNM          ;;ZERO THE TEST NUMBER
011330  005067  167624          CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
011334  005267  167642          INC     $PASS           ;;INCREMENT THE PASS NUMBER
011340  042767  100000  167634  BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
011346  005327                  DEC     (PC)+           ;;LOOP?
011350  000001          $EOPCT: .WORD   1
011352  003022                  BGT     $DOAGN          ;;YES
011354  012737                  MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
011356  000001          $ENDCT: .WORD   1
011360  011350                  $EOPCT
011362  104401  011427          TYPE    ,$ENDMG         ;;TYPE "END PASS #"
011366  016746  167610          MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
011372  104405                  TYPOS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
011374  104401  011424          TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
011400  013700  000042  $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
011404  001405                  BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
011406  000005                  RESET                   ;;CLEAR THE WORLD
011410  004710          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
011412  000240                  NOP                     ;;SAVE ROOM
011414  000240                  NOP                     ;;FOR
011416  000240                  NOP                     ;;ACT11
011420                  $DOAGN:
011420  000137                  JMP     @(PC)+          ;;RETURN
011422  010732          $RTNAD: .WORD   SPECIALADDRESS
011424     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
011427     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
011434  050040  051501  020123
011442  000043
```

```
                                .SBTTL  POWER DOWN AND UP ROUTINES

                                ;;**********************************************************************
                                ;POWER DOWN ROUTINE
011444  012737  011610  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
011452  012737  000340  000026         MOV     #340,@#PWRVEC+2 ;;PRIO:7
011460  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
011462  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
011464  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
011466  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
011470  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
011472  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
011474  017746  167440               MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
011500  010667  000110               MOV     SP,$SAVR6       ;;SAVE SP
011504  012737  011516  000024       MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
011512  000000                       HALT
011514  000776                       BR      .-2             ;;HANG UP

                                ;;**********************************************************************
                                ;POWER UP ROUTINE
011516  012737  011610  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
011524  016706  000064               MOV     $SAVR6,SP       ;;GET SP
011530  005067  000060               CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
011534  005267  000054       1$:     INC     $SAVR6          ;;WAIT FOR THE INC
011540  001375                       BNE     1$              ;;OF  WORD
011542  012677  167372               MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
011546  012605                       MOV     (SP)+,R5        ;;POP STACK INTO R5
011550  012604                       MOV     (SP)+,R4        ;;POP STACK INTO R4
011552  012603                       MOV     (SP)+,R3        ;;POP STACK INTO R3
011554  012602                       MOV     (SP)+,R2        ;;POP STACK INTO R2
011556  012601                       MOV     (SP)+,R1        ;;POP STACK INTO R1
011560  012600                       MOV     (SP)+,R0        ;;POP STACK INTO R0
011562  012737  011444  000024       MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
011570  012737  000340  000026       MOV     #340,@#PWRVEC+2 ;;PRIO:7
011576  104401                       TYPE                    ;REPORT THE POWER FAILURE
011600  011616               $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
011602  012716                       MOV     (PC)+,(SP)      ;;RESTART AT START
011604  001336               $PWRAD: .WORD   START           ;;RESTART ADDRESS
011606  000002                       RTI
011610  000000               $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
011612  000776                       BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
011614  000000               $SAVR6: 0                       ;;PUT THE SP HERE
011616  005015  047520  042527 $POWER: .ASCIZ  <15><12>"POWER"
011624  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
                                ;*

011626  105767  167325          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
011632  100002                          BPL     1$              ;;BR IF YES
011634  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
011636  000430                          BR      3$              ;;LEAVE
011640  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
011642  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
011646  122767  000001  167340          CMPB    @APTENV,$ENV    ;;RUNNING IN APT MODE
011654  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
011656  132767  000100  167331          BITB    @APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
011664  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
011666  010067  000004                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
011672  004767  001046                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
011676  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
011700  132767  000040  167307  62$:    BITB    @APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
011706  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
011710  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
011712  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
011714  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
011716  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
011720  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
011724  000002                          RTI                     ;;RETURN
011726  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
011732  001430                          BEQ     8$
011734  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
011740  001006                          BNE     5$
011742  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
011744  104401                          TYPE                    ;;TYPE A CR AND LF
011746  001171                          $CRLF
011750  105067  000202                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
011754  000755                          BR      2$              ;;GET NEXT CHARACTER
011756  004767  000056          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
011762  126726  167170          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
011766  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
011770  016746  167160                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
011774  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
012000  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
012002  004767  000032                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
012006  105367  000144                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
```

```
3896  012012  000770                        BR      7$              ;;LOOP
3897
3898                                  ;HORIZONTAL TAB PROCESSOR
3899
3900  012014  112716  000040          8$:   MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
3901  012020  004767  000014          9$:   JSR     PC,$TYPEC       ;;TYPE A SPACE
3902  012024  132767  000007  000124        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
3903  012032  001372                        BNE     9$              ;;TAB STOP
3904  012034  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
3905  012036  000724                        BR      2$              ;;GET NEXT CHARACTER
3906  012040                        $TYPEC:
3907  012040  105777  167100                TSTB    @#TKS           ;;CHAR IN KYBD BUFFER?          ;MJD001
3908  012044  100022                        BPL     10$             ;;BR IF NOT                     ;MJD001
3909  012046  017746  167074                MOV     @#TKB,-(SP)     ;;GET CHAR                      ;MJD001
3910  012052  042716  177600                BIC     #177600,(SP)    ;;STRIP EXTRANEOUS BITS         ;MJD001
3911  012056  122716  000023                CMPB    #^XOFF,(SP)     ;;WAS CHAR XOFF                 ;MJD001
3912  012062  001012                        BNE     102$            ;;BR IF NOT                     ;MJD001
3913  012064                        101$:                           ;                               ;MJD001
3914  012064  105777  167054                TSTB    @#TKS           ;;WAIT FOR CHAR                 ;MJD001
3915  012070  100375                        BPL     101$            ;                               ;MJD001
3916  012072  117716  167050                MOVB    @#TKB,(SP)      ;;GET CHAR                      ;MJD001
3917  012076  042716  177600                BIC     #177600,(SP)    ;;STRIP IT                      ;MJD001
3918  012102  122716  000021                CMPB    #^XON,(SP)      ;;WAS IT XON?                   ;MJD001
3919  012106  001366                        BNE     101$            ;;BR IF NOT                     ;MJD001
3920  012110                        102$:                           ;                               ;MJD001
3921  012110  005726                        TST     (SP)+           ;;FIX STACK                     ;MJD001
3922  012112                        10$:                            ;                               ;MJD001
3923  012112  105777  167032                TSTB    @#TPS           ;;WAIT UNTIL PRINTER IS READY
3924  012116  100375                        BPL     10$             ;                               ;MJD001
3925  012120  116677  000002  167024        MOVB    2(SP),@#TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
3926  012126  122766  000015  000002        CMPB    #^CR,2(SP)      ;;IS CHARACTER A CARRIAGE RETURN?
3927  012134  001003                        BNE     1$              ;;BRANCH IF NO
3928  012136  105067  000014                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
3929  012142  000406                        BR      $TYPEX          ;;EXIT
3930  012144  122766  000012  000002  1$:   CMPB    #^LF,2(SP)      ;;IS CHARACTER A LINE FEED?
3931  012152  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
3932  012154  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
3933  012156  000000          $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
3934  012160  000207          $TYPEX: RTS     PC
3935
```

```
                                .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.
012162  022767  000176  166750  $CKSWR: CMP     @SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
012170  001074                        BNE     15$             ;;BRANCH IF NO
012172  105777  166746                TSTB    @$TKS           ;;CHAR THERE?
012176  100071                        BPL     15$             ;;IF NO, DON'T WAIT AROUND
012200  117746  166742                MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
012204  042716  177600                BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
012210  022726  000007                CMP     #7,(SP)+        ;;IS IT A CONTROL G?
012214  001062                        BNE     15$             ;;NO, RETURN TO USER
012216  126727  166712  000001        CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
012224  001456                        BEQ     15$             ;;BRANCH IF YES

012226  104401  012707                TYPE    .$CNTLG         ;;ECHO THE CONTROL-G (^G)
012232  104401  012714        $GTSWR: TYPE    .$MSWR          ;;TYPE CURRENT CONTENTS
012236  016746  165734                MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
012242  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
012244  104401  012725                TYPE    .$MNEW          ;;PROMPT FOR NEW SWR
012250  005046                19$:    CLR     -(SP)           ;;CLEAR COUNTER
012252  005046                        CLR     -(SP)           ;;THE NEW SWR
012254  105777  166664        7$:     TSTB    @$TKS           ;;CHAR THERE?
012260  100375                        BPL     7$              ;;IF NOT TRY AGAIN

012262  117746  166660                MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
012266  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII


012272  021627  000025        9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
012276  001005                        BNE     10$             ;;BRANCH IF NOT
012300  104401  012702                TYPE    .$CNTLU         ;;YES, ECHO CONTROL-U (^U)
012304  062706  000006        20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
012310  000757                        BR      19$             ;;LET'S TRY IT AGAIN


012312  021627  000015        10$:    CMP     (SP),#15        ;;IS IT A <CR>?
012316  001022                        BNE     16$             ;;BRANCH IF NO
012320  005766  000004                TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
012324  001403                        BEQ     11$             ;;BRANCH IF YES
012326  016677  000002  166604        MOV     2(SP),@SWR      ;;SAVE NEW SWR
012334  062706  000006        11$:    ADD     #6,SP           ;;CLEAR UP STACK
012340  104401  001171        14$:    TYPE    .$CRLF          ;;ECHO <CR> AND <LF>
012344  126727  166565  000001        CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
012352  001003                        BNE     15$             ;;BRANCH IF NOT
012354  012777  000100  166562        MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
012362  000002                15$:    RTI                     ;;RETURN
012364  004767  177450        16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
012370  021627  000060                CMP     (SP),#60        ;;CHAR < 0?
```

```
                                        BLT     18$             ;;BRANCH IF YES
                                        CMP     (SP),#67        ;;CHAR > 7?
                                        BGT     18$             ;;BRANCH IF YES
                                        BIC     #60,(SP)+       ;;STRIP-OFF ASCII
                                        TST     2(SP)           ;;IS THIS THE FIRST CHAR
                                        BEQ     17$             ;;BRANCH IF YES
                                        ASL     (SP)            ;;NO, SHIFT PRESENT
                                        ASL     (SP)            ;;   CHAR OVER TO MAKE
                                        ASL     (SP)            ;;   ROOM FOR NEW ONE.
                                17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
                                        BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
                                        BR      7$              ;;GET THE NEXT ONE
                                18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
                                        BR      20$             ;;SIMULATE CONTROL-U
                                .DSABL  LSB

                                ;;**************************************************************************
                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*CALL:
                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
                                ;

                                $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
                                        MOV     4(SP),2(SP)     ;;SAVE THE PS
                                1$:     TSTB    @#TKS           ;;WAIT FOR
                                        BPL     1$              ;;A CHARACTER
                                        MOVB    @#TKB,4(SP)     ;;READ THE TTY
                                        BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
                                        CMP     4(SP),#23       ;;IS IT A CONTROL-S?
                                        BNE     3$              ;;BRANCH IF NO
                                2$:     TSTB    @#TKS           ;;WAIT FOR A CHARACTER
                                        BPL     2$              ;;LOOP UNTIL ITS THERE
                                        MOVB    @#TKB,-(SP)     ;;GET CHARACTER
                                        BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
                                        CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
                                        BNE     2$              ;;IF NOT DISCARD IT
                                        BR      1$              ;;YES, RESUME
                                3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
                                        BLT     4$              ;;BRANCH IF YES
                                        CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
                                        BGT     4$              ;;BRANCH IF YES
                                        BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
                                4$:     RTI                     ;;GO BACK TO USER
                                ;;**************************************************************************
                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                ;*CALL:
                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S

                                $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
                                1$:     MOV     @#TTYIN,R3      ;;GET ADDRESS
                                2$:     CMP     @#TTYIN+8.,R3   ;;BUFFER FULL?
```

```
4048  012576  101405                        BLOS    4$              ;;BR IF YES
4049  012600  104410                        RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
4050  012602  112613                        MOVB    (SP)+,(R3)      ;;GET CHARACTER
4051  012604  122713  000177        10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
4052  012610  001003                        BNE     3$              ;;SKIP IF NOT
4053  012612  104401  001170        4$:     TYPE    ,$QUES          ;;TYPE A '?'
4054  012616  000763                        BR      1$              ;;CLEAR THE BUFFER AND LOOP
4055  012620  111367  000044        3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
4056  012624  104401  012670                TYPE    ,9$
4057  012630  122723  000015                CMPB    #15,(R3)+       ;;CHECK FOR RETURN
4058  012634  001356                        BNE     2$              ;;LOOP IF NOT RETURN
4059  012636  105063  177777                CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
4060  012642  104401  001172                TYPE    ,$LF            ;;TYPE A LINE FEED
4061  012646  012603                        MOV     (SP)+,R3        ;;RESTORE R3
4062  012650  011646                        MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
4063  012652  016666  000004  000002        MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
4064  012660  012766  012672  000004        MOV     #$TTYIN,4(SP)
4065  012666  000002                        RTI                     ;;RETURN
4066  012670     000          9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
4067  012671     000                  .BYTE   0                       ;;TERMINATOR
4068  012672  000010          $TTYIN: .BLKB   8.                      ;;RESERVE 8 BYTES FOR TTY INPUT
4069  012702  052536  005015   000  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
4070  012707     136  006507  000012  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
4071  012714  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
4072  012722  020075     000
4073  012725     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
4074  012732  036440  000040
```

```
                                        .SBTTL  APT COMMUNICATIONS ROUTINE

                                        ;;*****************************************************************************
012736  112767  000001  000236  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
012744  112767  000001  000226  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
012752  000403                         BR      $ATYC
012754  112767  000001  000220  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
012762                          $ATYC:
012762  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
012764  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
012766  105767  000206                 TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
012772  001450                         BEQ     5$              ;;IF NOT:  BR
012774  122767  000001  166212         CMPB    @APTENV,$ENV    ;;OPERATING UNDER APT?
013002  001031                         BNE     3$              ;;IF NOT:  BR
013004  132767  000100  166203         BITB    @APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
013012  001425                         BEQ     3$              ;;IF NOT:  BR
013014  017600  000004                 MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
013020  062766  000002  000004         ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
013026  005767  166142          1$:    TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
013032  001375                         BNE     1$              ;;IF NOT:  WAIT
013034  010067  166150                 MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
013040  105720                  2$:    TSTB    (R0)+           ;;FIND END OF MESSAGE
013042  001376                         BNE     2$
013044  166700  166140                 SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
013050  006200                         ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
013052  010067  166134                 MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
013056  012767  000004  166110         MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
013064  000413                         BR      5$
013066  017667  000004  000016  3$:    MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
013074  062766  000002  000004         ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
013102  016746  164670                 MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
013106  004767  176514                 JSR     PC,$TYPE        ;;CALL TYPE MACRO
013112  000000                  4$:    .WORD   0
013114                          5$:
013114  105767  000062          10$:   TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
013120  001416                         BEQ     12$             ;;IF NOT:  BR
013122  005767  166066                 TST     $ENV            ;;RUNNING UNDER APT?
013126  001413                         BEQ     12$             ;;IF NOT:  BR
013130  005767  166040          11$:   TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
013134  001375                         BNE     11$             ;;IF NOT:  WAIT
013136  017667  000004  166032         MOV     @4(SP),$FATAL   ;;GET ERROR #
013144  062766  000002  000004         ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
013152  005267  166016                 INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
013156  105067  000020          12$:   CLRB    $FFLG           ;;CLEAR FATAL FLAG
013162  105067  000013                 CLRB    $LFLG           ;;CLEAR LOG FLAG
013166  105067  000006                 CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
```

```
 4121  013172  012601                      MOV     (SP)+,R1        ;;POP STACK INTO R1
 4122  013174  012600                      MOV     (SP)+,R0        ;;POP STACK INTO R0
 4123  013176  000207                      RTS     PC              ;;RETURN
 4124  013200     000              $MFLG:  .BYTE   0               ;;MESSG. FLAG
 4125  013201     000              $LFLG:  .BYTE   0               ;;LOG FLAG
 4126  013202     000              $FFLG:  .BYTE   0               ;;FATAL FLAG
 4127           013204                      .EVEN
 4128           000200              APTSIZE=200
 4129           000001              APTENV=001
 4130           000100              APTSPOOL=100
 4131           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

013204                          $ERROR:
013204  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
013206  105267  165671          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
013212  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
013214  016777  165662  165720          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
013222  032777  002000  165710          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
013230  001402                          BEQ     1$              ;;NO - SKIP
013232  104401  001164                  TYPE    .$BELL          ;;RING BELL
013236  005267  165650          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
013242  011667  165650                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
013246  162767  000002  165642          SUB     #2,$ERRPC
013254  117767  165636  165632          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
013262  032777  020000  165650          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
013270  001004                          BNE     20$             ;;SKIP TYPEOUTS
013272  004767  175634                  JSR     PC,MYTYPE       ;;GO TO USER ERROR ROUTINE
013276  104401  001171                  TYPE    .$CRLF
013302                          20$:
013302  122767  000001  165704          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
013310  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
013312  116767  165576  000004          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
013320  004767  177430                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
013324     000                  21$:    .BYTE   0
013325     000                          .BYTE   0
013326  000777                  22$:    BR      22$             ;;APT ERROR LOOP
013330  005777  165604          2$:     TST     @SWR            ;;HALT ON ERROR
013334  100002                          BPL     3$              ;;SKIP IF CONTINUE
013336  000000                          HALT                    ;;HALT ON ERROR!
013340  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
013342  032777  001000  165570  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
013350  001402                          BEQ     4$              ;;BR IF NO
013352  016716  165532                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
013356  005767  165600          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
013362  001402                          BEQ     5$              ;;BR IF NONE
013364  016716  165572                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
013370                          5$:
013370  022737  011410  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
013376  001001                          BNE     6$              ;;BRANCH IF NO
013400  000000                          HALT                    ;;YES
013402                          6$:
013402  000002                          RTI                     ;;RETURN
```

```
                                .SBTTL  SCOPE HANDLER ROUTINE

                                ;;**********************************************************************
                                ;*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

013404                          $SCOPE:
                                        CKSWR                           ;;TEST  FOR CHANGE IN SOFT-SWR
013404  104407                  1$:     BIT     #BIT14,@SWR             ;;LOOP ON PRESENT TEST?
013406  032777  040000  165524          BNE     $OVER                   ;;YES IF SW14=1
013414  001114
                                ;@@@@@START OF CODE FOR THE XOR TESTER@@@@@
013416  000416                  $XTSTR: BR      6$                      ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
013420  013746  000004                  MOV     @#ERRVEC,-(SP)          ;;SAVE THE CONTENTS OF THE ERROR VECTOR
013424  012737  013444  000004          MOV     #5$,@#ERRVEC            ;;SET FOR TIMEOUT
013432  005737  177060                  TST     @#177060                ;;TIME OUT ON XOR?
013436  012637  000004                  MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
013442  000463                          BR      $SVLAD                  ;;GO TO THE NEXT TEST
013444  022626                  5$:     CMP     (SP)+,(SP)+             ;;CLEAR THE STACK AFTER A TIME OUT
013446  012637  000004                  MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
013452  000423                          BR      7$                      ;;LOOP ON THE PRESENT TEST
013454                          6$:;@@@@@END OF CODE FOR THE XOR TESTER@@@@@
013454  032777  000400  165456          BIT     #BIT08,@SWR             ;;LOOP ON SPEC. TEST?
013462  001404                          BEQ     2$                      ;;BR IF NO
013464  127767  165450  165410          CMPB    @SWR,$TSTNM             ;;ON THE RIGHT TEST?    SWR<7:0>
013472  001465                          BEQ     $OVER                   ;;BR IF YES
013474  105767  165403          2$:     TSTB    $ERFLG                  ;;HAS AN ERROR OCCURRED?
013500  001421                          BEQ     3$                      ;;BR IF NO
013502  126767  165407  165373          CMPB    $ERMAX,$ERFLG           ;;MAX. ERRORS FOR THIS TEST OCCURRED?
013510  101015                          BHI     3$                      ;;BR IF NO
013512  032777  001000  165420          BIT     #BIT09,@SWR             ;;LOOP ON ERROR?
013520  001404                          BEQ     4$                      ;;BR IF NO
013522  016767  165362  165356  7$:     MOV     $LPERR,$LPADR           ;;SET LOOP ADDRESS TO LAST SCOPE
013530  000446                          BR      $OVER
013532  105067  165345          4$:     CLRB    $ERFLG                  ;;ZERO THE ERROR FLAG
013536  005067  165416                  CLR     $TIMES                  ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
013542  000415                          BR      1$                      ;;ESCAPE TO THE NEXT TEST
013544  032777  004000  165366  3$:     BIT     #BIT11,@SWR             ;;INHIBIT ITERATIONS?
013552  001011                          BNE     1$                      ;;BR IF YES
013554  005767  165422                  TST     $PASS                   ;;IF FIRST PASS OF PROGRAM
013560  001406                          BEQ     1$                      ;;      INHIBIT ITERATIONS
013562  005267  165316                  INC     $ICNT                   ;;INCREMENT ITERATION COUNT
013566  026767  165366  165310          CMP     $TIMES,$ICNT            ;;CHECK THE NUMBER OF ITERATIONS MADE
013574  002024                          BGE     $OVER                   ;;BR IF MORE ITERATION REQUIRED
013576  012767  000001  165300  1$:     MOV     #1,$ICNT                ;;REINITIALIZE THE ITERATION COUNTER
013604  016767  000052  165346          MOV     $MXCNT,$TIMES           ;;SET NUMBER OF ITERATIONS TO DO
013612  105267  165264          $SVLAD: INCB    $TSTNM                  ;;COUNT TEST NUMBERS
013616  116767  165260  165354          MOVB    $TSTNM,$TESTN           ;;SET TEST NUMBER IN APT MAILBOX
```

```
 4242  013624  011667  165256                    MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 4243  013630  011667  165254                    MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 4244  013634  005067  165322                    CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 4245  013640  112767  000001  165247            MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 4246  013646  016777  165230  165266  $OVER:    MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 4247  013654  016716  165226                    MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 4248  013660  000002                            RTI                     ;;FIXES PS
 4249  013662  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

013664                                  $TYPDS:
013664  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
013666  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
013670  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
013672  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
013674  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
013676  012746  020200                 MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
013702  016605  000020                 MOV     20(SP),R5       ;;GET THE INPUT NUMBER
013706  100004                         BPL     1$              ;;BR IF INPUT IS POS.
013710  005405                         NEG     R5              ;;MAKE THE BINARY NUMBER POS.
013712  112766  000055  000001         MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
013720  005000                 1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
013722  012703  014100                 MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
013726  112723  000040                 MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
013732  005002                 2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
013734  016001  014070                 MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
013740  160105                 3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
013742  002402                         BLT     4$              ;;BR IF DONE
013744  005202                         INC     R2              ;;INCREASE THE BCD DIGIT BY 1
013746  000774                         BR      3$
013750  060105                 4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
013752  005702                         TST     R2              ;;CHECK IF BCD DIGIT=0
013754  001002                         BNE     5$              ;;FALL THROUGH IF 0
013756  105716                         TSTB    (SP)            ;;STILL DOING LEADING 0'S?
013760  100407                         BMI     7$              ;;BR IF YES
013762  106316                 5$:     ASLB    (SP)            ;;MSD?
013764  103003                         BCC     6$              ;;BR IF NO
013766  116663  000001  177777         MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
013774  052702  000060         6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
014000  052702  000040         7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
014004  110223                         MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
014006  005720                         TST     (R0)+           ;;JUST INCREMENTING
014010  020027  000010                 CMP     R0,#10          ;;CHECK THE TABLE INDEX
014014  002746                         BLT     2$              ;;GO DO THE NEXT DIGIT
014016  003002                         BGT     8$              ;;GO TO EXIT
014020  010502                         MOV     R5,R2           ;;GET THE LSD
014022  000764                         BR      6$              ;;GO CHANGE TO ASCII
014024  105726                 8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
014026  100003                         BPL     9$              ;;BR IF NO
014030  116663  177777  177776         MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
014036  105013                 9$:     CLRB    (R3)            ;;SET THE TERMINATOR
014040  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
014042  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
014044  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
 4306  014046  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 4307  014050  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 4308  014052  104401  014100                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 4309  014056  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 4310  014064  012616                          MOV     (SP)+,(SP)
 4311  014066  000002                          RTI                     ;;RETURN TO USER
 4312  014070  023420                  $DTBL:  10000.
 4313  014072  001750                          1000.
 4314  014074  000144                          100.
 4315  014076  000012                          10.
 4316  014100  000004                  $DBLK:  .BLKW   4
```

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

                                ;;*****************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                ;*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

014110  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
014114  116667  000001  000211          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
014122  112667  000207                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
014126  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
014132  000406                          BR      $TYPON
014134  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
014142  112767  000006  000165          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
014150  112767  000005  000154  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
014156  010346                          MOV     R3,-(SP)        ;;SAVE R3
014160  010446                          MOV     R4,-(SP)        ;;SAVE R4
014162  010546                          MOV     R5,-(SP)        ;;SAVE R5
014164  116704  000145                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
014170  005404                          NEG     R4
014172  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
014176  110467  000132                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
014202  116704  000125                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
014206  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
014212  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
014214  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
014216  000404                          BR      3$              ;;GO DO MSB
014220  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
014222  006105                          ROL     R5
014224  006105                          ROL     R5
014226  010503                          MOV     R5,R3
014230  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
014232  105367  000076                  DECB    $OMODE          ;;TYPE THIS DIGIT?
014236  100016                          BPL     7$              ;;BR IF NO
014240  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
014244  001002                          BNE     4$              ;;TEST FOR 0
014246  005704                          TST     R4              ;;SUPPRESS THIS 0?
014250  001403                          BEQ     5$              ;;BR IF YES
```

```
 4373  014252  005204                    4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 4374  014254  052703  000060                    BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 4375  014260  052703  000040            5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 4376  014264  110367  000040                    MOVB    R3,8$           ;;SAVE FOR TYPING
 4377  014270  104401  014330                    TYPE    ,8$             ;;GO TYPE THIS DIGIT
 4378  014274  105367  000032            7$:     DECB    $OCNT           ;;COUNT BY 1
 4379  014300  003347                            BGT     2$              ;;BR IF MORE TO DO
 4380  014302  002402                            BLT     6$              ;;BR IF DONE
 4381  014304  005204                            INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 4382  014306  000744                            BR      2$              ;;GO DO THE LAST DIGIT
 4383  014310  012605                    6$:     MOV     (SP)+,R5        ;;RESTORE R5
 4384  014312  012604                            MOV     (SP)+,R4        ;;RESTORE R4
 4385  014314  012603                            MOV     (SP)+,R3        ;;RESTORE R3
 4386  014316  016666  000002  000004            MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 4387  014324  012616                            MOV     (SP)+,(SP)
 4388  014326  000002                            RTI                     ;;RETURN
 4389  014330     000                    8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 4390  014331     000                            .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 4391  014332     000                $OCNT:      .BYTE   0               ;;OCTAL DIGIT COUNTER
 4392  014333     000                $OFILL:     .BYTE   0               ;;ZERO FILL SWITCH
 4393  014334  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.

014336  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
014340  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
014344  005740                          TST     -(R0)           ;;BACKUP BY 2
014346  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
014350  006300                          ASL     R0              ;;POSITION FOR INDEXING
014352  016000  014372                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
014356  000200                          RTS     R0              ;;GO TO ROUTINE

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

014360  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
014362  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
014370  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
014372  014360                  $TRPAD: .WORD   $TRAP2
014374  011626                          $TYPE   ;;CALL-TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
014376  014134                          $TYPOC  ;;CALL-TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
014400  014110                          $TYPOS  ;;CALL-TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
014402  014150                          $TYPON  ;;CALL-TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
014404  013664                          $TYPDS  ;;CALL-TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

014406  012232                          $GTSWR  ;;CALL-GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

014410  012162                          $CKSWR  ;;CALL-CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
014412  012444                          $RDCHR  ;;CALL-RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
014414  012564                          $RDLIN  ;;CALL-RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
        000001                  .END
```

```
ABASE = 175610            1#   956     997
ACDM1 = 000000           956
ACDM2 = 000000           956
ACPUDP= 000000           956   971
ADDRES  011112          3643*  3654*  3668*  3691   3698#
ADDM0 = 000000           956
ADDM1 = 000000           956
ADDM10= 000000           956
ADDM11= 000000           956
ADDM12= 000000           956
ADDM13= 000000           956
ADDM14= 000000           956
ADDM15= 000000           956
ADDM2 = 000000           956
ADDM3 = 000000           956
ADDM4 = 000000           956
ADDM5 = 000000           956
ADDM6 = 000000           956
ADDM7 = 000000           956
ADDM8 = 000000           956
ADDM9 = 000000           956
ADEVCT= 000000           956   962
ADEVM = 100000            1#   956     998
AENV  = 000000           956   967
AENVM = 000000           956   968
AFATAL= 000000           956   959
AHADR1= 000000           956   984
AHADR2= 000000           956   988
AHADR3= 000000           956   991
AHADR4= 000000           956   994
AHWS1= 000000            956   978
AHWS2= 000000            956   986
AHWS3= 000000            956   989
AHWS4= 000000            956   992
AHSGAD= 000000           956   964
AHSGLG= 000000           956   965
AHSGTY= 000000           956   958
AHTYP1= 000000           956   979
AHTYP2= 000000           956   987
AHTYP3= 000000           956   990
AHTYP4= 000000           956   993
APASS = 000000           956   961
APRIOR= 000000           956
APTCON  011124          3614*  3682*  3703#
APTCSU= 000040          3870   4131#
APTENV= 000001          1217   1695   1982   2318   2492   3675   3863   4087   4129#   4163
APTSIZ= 000200          1067   4128#
APTSPO= 000100          3865   4089   4130#
ASSEMB= 000010             1
ASWREG= 000000           956   969
ATESTN= 000000           956   960
AUNIT = 000000           956   963
AUSMR = 011110            1#   956     970
AVECT1= 000300            1#   956     995
AVECT2= 000000           956   996
BAUD  = 007400          859#
```

```
$ITMAS  011106          1106#   3620    3639#   3652#   3660    3663    3696#
$IT0  = 000001           734#   3311    3334
$IT00 = 000001           724#    734     783     803     824     845
$IT01 = 000002           723#    733     802     844
$IT02 = 000004           722#    732     801     822     843
$IT03 = 000010           721#    731     800     842
$IT04 = 000020           720#    730     799     841
$IT05 = 000040           719#    729     798     840
$IT06 = 000100           718#    728     777     797     818     839
$IT07 = 000200           717#    727     776     796     817     838
$IT08 = 000400           716#    726    4216
$IT09 = 001000           715#    725    4174    4224
$IT1  = 000002           733#   3322    3341
$IT10 = 002000           714#   4151
$IT11 = 004000           713#    772     812    4231
$IT12 = 010000           712#    791     811
$IT13 = 020000           711#    790     810    4158
$IT14 = 040000           710#    789     809    4202
$IT15 = 100000           709#    788     808    1106    3663
$IT2  = 000004           732#   2406    2434
$IT3  = 000010           731#   2811    2871
$IT4  = 000020           730#   2830    2878
$IT5  = 000040           729#
$IT6  = 000100           728#
$IT7  = 000200           727#
$IT8  = 000400           726#
$IT9  = 001000           725#
$PTVEC= 000014           741#
BREAK = 000001           824#   1231    1244    1247    1261    1264    1278    1283    3285
BRK   = 010000           860#   1215    3261
CKSWR = 104407          4147    4173    4201   4433#
CLR   = 000000           762#   1710    1712    1995    3424
CONSPD= 000100           854#   2898
CONADR  011120          3668    3701#
CONSOL  011125          1305    1591    1693    1828    1902    1971    2128    2638    2758    3040    3270    3616#   3666#
                        3704#
CONVEC  011122          3670    3702#
CR    = 000015           649#   3926    3936
CRLF  = 000200           650#   1091    1097    1098    1099    3361    3721    3880    3936
CYCLE   010666          1109    3601#
DATA  = 000017           851#   3509    3510
DATLNG  010476          2782    2930    3052   3491#
DDISP = 177570           656#    937    1055
DIAGNC= 000000             1
DISPLA  001142           937#   1055#   1063#   4150#   4246#
DISPRE  000174           872#   1063
DLADD   001254          1017#   1113#   1118    1121    1123    1126    1129    1132    1148    1180    3362    3743
DLADDR= 175610           758#   1017    1019    1020    1021    1022    1023
DLVEC   001256          1018#   1116#   2507    2607    2649    2735    3074    3368    3749
DSWR  = 177570           655#    936    1054
EMTVEC= 000030           744#   1038#   1039#
ERRBIT= 100000           863#   2126    3316
ERRCHK  010172          2327#   2406#   2434    2768#   2811#   2830#   2871    2878    3279#   3311#   3322#   3334    3341
                        3349#
ERRCNT  007562          3089#   3120    3128   3143#   3208    3219#
ERROR = 100000           788#   2203    2277
```

```
ERRORF  004026          1710* 1757* 1784  1812#
ERRVEC- 000004           737# 1052  1053* 1064* 4207  4208* 4210* 4213*
EVENOD- 000040           853#
EXITFL  004030          1712* 1793* 1800* 1804  1813#
FALSE   - 000000        3405# 3412  3614  3616
FLAG    G10470          3412* 3439* 3444  3467  3479#
FRERR   - 020000         790# 3319
GNS     - ****** U       871  1090  1097  1098  1099  3360  3366  3372  3720  3726  3735  3741  3747
                        4425  4426  4427  4428  4429  4431  4433  4434  4435
GTSWR   - 104406        1085  4431#
HOLD    007626          3160* 3161* 3164  3177#
HOLDSC  010471          3424* 3429* 3435  3480#
HOWLON- 000000          3384# 3410
HT      - 000011         647# 3878  3936
I       001272          1024# 1157* 1178* 1181  1183  3514* 3517* 3519
ILLMEM- 000004           749# 1152  1188
INITFL  011110          1104* 3623  3626* 3697#
INTFLA  010664          1164* 1169  2303* 2353  2557  2579* 2595  2657* 2699  2706  3596* 3599#
INTSRV  010656          1153  2515  2650  3586#
IOTVEC- 000020           742# 1036* 1037*
LF      - 000012         648# 3930  3936
LOOP    002136          1107# 3378
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  2660  2704  2732  2771  2855  2869  2909  3104  3125  3282
MAINTJ- 040000          862# 1303  1826  1900  1973  2137  2308  2636  2756  3038  3259
MASK    - 000000        3492# 3506* 3522* 3524* 3529*
MAX     004672          2021  2025  2112#
MYTYPE  011132          3716# 4160
NEW     005644          2356* 2358* 2374* 2376* 2385  2393  2395  2413  2415  2470#
NOCOMM  011126          3705#
NUMBER  007564          3118  3144# 3167  3225
NUMBR   010572          3508* 3509  3511* 3519  3532#
OK      011116          3700#
OLD     005650          2329* 2331* 2385  2393  2395  2413* 2415* 2471#
OVERR   - 040000         789# 2182  2227  2259
PARITY- 000020           852#
PASS    004024          1705* 1787  1811#
PBAUDS- 004000           812#
PBAUD0- 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  2666  2775  2923  3068
PR1     - 000040         672#
PR2     - 000100         673#
PR3     - 000140         674#
PR4     - 000200         675#
PR5     - 000240         676#
PR6     - 000300         677#
PR7     - 000340         678# 1154  1391  2516  2651  3079  3085
PS      - 177776         651#  652
```

```
PSW    = 177776            6520#
PWRVEC= 000024             7430#  1042#  1043#  3799#  3800#  3809#  3815#  3827#  3828#
RATES   005624             2347   2428   2444#
RBUF    001262            10200#  1123#  1124#  1950   2096   2178   2227   2259   2277   2344   2696   2785   2835
                           2933   2990   3195   3307   3319
RCSR    001260            10190#  1121#  1477   1490#  1493   1507#  1510   1524#  1529   1560   1625   1854   1882
                           1928   1953   2007   2047   2051   2054   2077   2099   2364   2663#  2680   2687#  2727#
                           2822   2920#  2972   3019#  3111#  3139#  3229#  3293
RCVRAC= 004000             7720#  1625   2007   2047   2054
RCVRDO= 000200             7760#  1560   1855   1882   1929   1953   2051   2077   2099   2364   2631   2823   2973
                           3294
RCVRIE= 000100             7770#  1477   1490   1493   1507   1510   1524   1529   2663   2687   2727   3111   3139
                           3229
RDATA0= 000001             8030#
RDATA1= 000002             8020#
RDATA2= 000004             8010#
RDATA3= 000010             8000#
RDATA4= 000020             7990#
RDATA5= 000040             7980#
RDATA6= 000100             7970#
RDATA7= 000200             7960#
RDCHR  = 104410            4049   44340#
RDLIN  = 104411            4435#
RDPRUN= 000001             7830#
REC     007632             3077   31870#
REG    = 000004            3384#  3408
REGSAV  010464             3408#  3421   3477#
RESVEC= 000010             7380#
RHLD    007722             3199#  3200#  3204   3212   32380#
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      007566             3145#  3212#
SET    = 177777            7610#  1726   1757   1768   1784   1793   1800   1804   1853   1927   2010   2019   2523
                           2679   2802   2821   2950   2971   3292   3429
SETCLR= 000006             3384#  3435
SPECIA  010732             3633#  3790
STACK  = 001100            6420#  1034
START   001336             875    1027#  3832
STKLMT= 177774             6530#
SWR     001140             9360#  1032   1054#  1056   1062#  1069#  1083   3807   3820#  3946   3983#  4151   4158
                           4170   4174   4202   4216   4218   4224   4231
```

```
SWREG   000176            8734#  1062    1083    3946    3959
SW0   = 000001            7064#
SW00  = 000001            6964#   706
SW01  = 000002            6954#   705
SW02  = 000004            6944#   704
SW03  = 000010            6934#   703
SW04  = 000020            6924#   702
SW05  = 000040            6914#   701
SW06  = 000100            6904#   700
SW07  = 000200            6894#   699
SW08  = 000400            6884#   698
SW09  = 001000            6874#   697
SW1   = 000002            7054#
SW10  = 002000            6864#
SW11  = 004000            6854#
SW12  = 010000            6844#
SW13  = 020000            6834#
SW14  = 040000            6824#
SW15  = 100000            6814#
SW2   = 000004            7044#
SW3   = 000010            7034#
SW4   = 000020            7024#
SW5   = 000040            7014#
SW6   = 000100            7004#
SW7   = 000200            6994#
SW8   = 000400            6984#
SW9   = 001000            6974#
TBITVE= 000014            7394#
TBUF    001270           10234#  1132#  1133#  1720#  1748#  1846#  1921#  2003#  2158#  2168#  2250#  2353#  2673#
                          2817#  2967#  3164#  3288#
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#  2660#  2704#
                         2732#  2771#  2803   2855#  2869#  2909#  2951   3104#  3108#  3125#  3137#  3171#  3282#
                         3285#
TCSRHI  001266           1022#  1129#  1130#  2347#  2428#
TDATA0= 000001            8454#
TDATA1= 000002            8444#
TDATA2= 000004            8434#
TDATA3= 000010            8424#
TDATA4= 000020            8414#
TDATA5= 000040            8404#
TDATA6= 000100            8394#
TDATA7= 000200            8384#
TIME  = 000000           3542#  3553
TIMER   010320           1730   1772   1857   1931   2527   2683   2806   2825   2954   2975   3296   3383#
TIMSAV  010466           3410#  3446   3457#  3478#
TKVEC = 000060            7464#
TPVEC = 000064            7474#
TRAN    007570           3083   3152#
TRAPVE= 000034            7454#  1040#  1041#
TRTVEC= 000014            7404#
TRUE  = 000001           1305   1591   1693   1828   1902   1971   2128   2638   2758   3040   3270   3404#  3439
                         3444   3467   3666   3682
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    1966    2036    2066    2087    2111    2121#
TST16   005260            2133    2141    2194    2217    2239    2272    2299#
TST17   005654            2313    2322    2442    2485#
TST2    002420            1210#
TST20   006202            2496    2632#
TST21   006506            2641    2752#
TST22   007030            2763    2860    2892#
TST23   007332            2904    2961    2984    3008    3033#
TST24   007726            3045    3141    3255#
TST25   010174            3266    3274    3348    3356#
TST3    002612            1222    1298#
TST4    003004            1310    1387#
TST5    003160            1469#
TST6    003322            1550#
TST7    003362            1583#
TYPDS = 104405            3375    3779    4429#
TYPE  = 104401            1077    1097    1098    1099    3358    3364    3370    3377    3718    3724    3733    3739    3745
                          3751    3777    3780    3829    3883    3957    3958    3961    3974    3985    4004    4053    4056
                          4060    4153    4161    4308    4377    4425#
TYPOC = 104402            3363    3369    3723    3738    3744    3750    3960    4426#
TYPON = 104404            4428#
TYPOS = 104403            3730    4427#
VECTOR  011114            3645*   3656*   3670*   3699#
WAIT    010576            2163    2173    2255    2548    2592    2692    3454    3541#
WAS     007367            3146#   3214*
WHICHB= 000002            3304#   3421
WRAP  = 020000             861#   1598    2896
XMITIE= 000100             818#   1401    1414    1417    1431    1434    1448    1453    2532    2542    2582    3108    3137
                          3171
XMITRD= 000200             817#   1663    1728    1753    1770    2525    2804    2952
ZZZ     007630            3175    3179#
ZZZZ    007724            3236    3239#
$APTHD  001000             896     902#
$ASTAT= ****** U          4109    4124
$ATYC   012762            4080    4082#
$ATY1   012736            4078#
$ATY3   012744            3868    4079#
$ATY4   012754            4081#   4166
$AUTOB  001134             933#   1087*   3954    4075
$BASE   001250             997#   3643
$BDADR  001122             928#
$BDDAT  001126             930#
$BELL   001164             948#   4153    4186
$BGMLE= 177777              1#
$CHARC  012156            3885*   3895*   3902    3928*   3933#
$CKSMR  012162            3946#   4433
$CMTAG  001100             916#   1029    1030    1038    1044    1045    1046
$CMS  = 000000             946#
$CNTLG  012707            3957    4070#
$CNTLU  012702            3974    4069#
$CPUOP  001222             971#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
$ENV    001214          967#  1081  1217  1695  1982  2318  2492  3675  3863  4087  4111  4163
$ENVM   001215          968#  1067  3865  3870  4089
$EOP    011322         3631  3765#
$EOPCT  011350         1044#  3772#  3776
$ERFLG  001103          919#  4148#  4186  4191  4220  4222  4228#  4250
$ERMAX  001115          925#  1047#  4222  4245#  4250
$ERROR  013204         1038  4146#
$ERRPC  001116          926#  3737  4155#  4156#  4157  4186
$ERRTB  001254         1015#
$ERTTL  001112          923#  3374  3376#  4154#  4186
$ESCAP  001162          947#  1046#  4177  4179  4186  4244#
$ETABL  001214          966#
$ETEND  001254          908  999#
$FATAL  001176          959#  3728#  3729  4115#
$FFLG   013202         4078#  4081#  4109  4118#  4126#
$FILLC  001156          944#  3888  3936
$FILLS  001155          943#  3936
$F$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  2638  2701  2708
                       2758  2852  2873  2980  2898  3003  3040  3130  3169  3206  3210  3227  3261
                       3272  3309  3318  3321  3336  3343  3423  3437  3469  3622  3625  3665  3677
                       3680
$F$BAD= 000401            1#  1096  1104  1106  1113  1116  1118  1121  1123  1126  1129  1132  1135
                       1148  1151  1152  1153  1154  1155  1157  1161  1164  1171  1178  1180  1188
                       1189  1191  1192  1220  1221  1228  1233  1242  1244  1249  1258  1261  1266
                       1275  1278  1285  1308  1309  1316  1321  1330  1332  1337  1346  1349  1354
                       1363  1366  1373  1398  1403  1412  1414  1419  1428  1431  1436  1445  1448
                       1455  1474  1479  1488  1490  1495  1504  1507  1512  1521  1524  1531  1558
                       1562  1593  1600  1607  1612  1622  1627  1630  1660  1665  1698  1699  1705
                       1710  1712  1720  1748  1755  1757  1786  1789  1793  1800  1831  1832  1839
                       1842  1846  1866  1875  1884  1905  1906  1913  1915  1921  1934  1950  1955
                       1976  1977  1984  1985  1993  1995  1997  2003  2009  2010  2015  2027  2031
                       2035  2049  2053  2056  2060  2065  2079  2082  2086  2096  2101  2104  2131
                       2132  2139  2140  2146  2151  2158  2168  2178  2184  2187  2193  2201  2205
                       2209  2216  2223  2229  2233  2238  2245  2250  2261  2264  2271  2279  2282
                       2311  2312  2320  2321  2327  2329  2331  2333  2336  2339  2344  2347  2350
                       2353  2356  2358  2363  2366  2368  2374  2376  2387  2397  2406  2413  2415
                       2421  2426  2428  2432  2436  2494  2495  2503  2507  2510  2513  2514  2515
                       2516  2517  2519  2532  2542  2555  2559  2576  2579  2582  2597  2607  2611
                       2612  2614  2615  2641  2648  2649  2650  2651  2652  2655  2657  2660  2663
                       2673  2687  2696  2701  2704  2708  2727  2732  2735  2739  2740  2742  2743
                       2761  2762  2768  2771  2785  2789  2792  2811  2817  2830  2835  2843  2846
                       2852  2855  2859  2869  2873  2880  2901  2903  2909  2920  2933  2936  2939
                       2960  2967  2983  2990  2994  2997  3003  3007  3019  3043  3044  3074  3077
                       3079  3083  3085  3089  3092  3095  3098  3104  3108  3111  3125  3130  3137
                       3139  3157  3160  3164  3169  3171  3192  3195  3199  3206  3210  3212  3214
                       3219  3227  3229  3244  3265  3272  3273  3279  3282  3285  3288  3309  3311
                       3318  3321  3322  3336  3343  3408  3410  3412  3423  3424  3429  3437  3439
                       3457  3469  3506  3508  3514  3517  3522  3524  3529  3553  3555  3558  3563
                       3566  3596  3614  3616  3622  3625  3626  3635  3639  3641  3643  3645  3650
                       3652  3654  3656  3665  3666  3668  3670  3677  3680  3682  3691  3693
$F$BLA= 000170             1#
$F$CAS= 000150             1#
```

CROSS REFEHENCE TABLE -- USER SYMBOLS

```
$F$DEC= 000220          10#
$F$DO = 000340          10#  1096  2049  2363
$F$FAL= 000405          10#
$F$GOO= 000400          10#  1093  1094  1104  1105  1106  1107  1109  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  1178  1179  1180  1181  1182  1188  1189
                1190    1191  1192  1193  1215  1221  1222  1228  1229  1231  1242  1243  1244
                1245    1247  1258  1259  1261  1262  1264  1275  1276  1278  1279  1283  1303
                1309    1310  1316  1317  1319  1330  1331  1332  1333  1335  1346  1347  1349
                1350    1352  1363  1364  1366  1367  1371  1398  1399  1401  1412  1413  1414
                1415    1417  1428  1429  1431  1432  1434  1445  1446  1448  1449  1453  1474
                1475    1477  1488  1489  1490  1491  1493  1504  1505  1507  1508  1510  1521
                1522    1524  1525  1529  1558  1559  1560  1591  1598  1605  1612  1613  1622
                1623    1625  1630  1631  1660  1661  1663  1693  1699  1700  1705  1706  1710
                1711    1712  1713  1720  1721  1725  1735  1748  1749  1753  1757  1758  1767
                1775    1784  1787  1793  1794  1800  1801  1826  1832  1833  1839  1840  1842
                1843    1846  1847  1852  1862  1866  1867  1875  1876  1882  1900  1906  1907
                1913    1914  1915  1916  1921  1922  1926  1934  1935  1937  1950  1951  1953
                1971    1977  1978  1982  1985  1986  1993  1994  1995  1996  1997  1998  2003
                2004    2007  2010  2011  2015  2016  2025  2031  2032  2035  2036  2046  2047
                2051    2054  2060  2061  2065  2066  2077  2082  2083  2086  2087  2096  2097
                2099    2104  2105  2126  2132  2133  2137  2140  2141  2146  2147  2151  2152
                2158    2159  2161  2168  2169  2171  2178  2179  2182  2187  2188  2193  2194
                2201    2202  2203  2209  2210  2216  2217  2223  2224  2227  2233  2234  2238
                2239    2245  2246  2250  2251  2253  2259  2264  2265  2271  2272  2277  2282
                2283    2306  2312  2313  2318  2321  2322  2327  2328  2329  2330  2331  2332
                2333    2334  2336  2337  2339  2340  2343  2344  2345  2347  2348  2350  2351
                2353    2354  2356  2357  2358  2359  2360  2361  2364  2368  2369  2374  2375
                2376    2377  2385  2393  2406  2407  2413  2414  2415  2416  2421  2422  2425
                2426    2427  2428  2429  2432  2433  2434  2492  2495  2496  2503  2504  2507
                2508    2510  2511  2513  2514  2515  2516  2517  2518  2519  2520  2522  2532
                2533    2542  2543  2546  2553  2557  2576  2577  2579  2580  2582  2583  2590
                2595    2607  2608  2611  2612  2613  2614  2615  2616  2636  2648  2649  2650
                2651    2652  2653  2655  2656  2657  2658  2660  2661  2663  2664  2673  2674
                2678    2687  2688  2690  2696  2697  2699  2704  2705  2706  2727  2728  2732
                2733    2735  2736  2739  2740  2741  2742  2743  2744  2756  2762  2763  2768
                2769    2771  2772  2781  2785  2786  2789  2790  2792  2793  2796  2801  2809
                2811    2812  2817  2818  2820  2828  2830  2831  2835  2836  2843  2844  2845
                2846    2847  2850  2855  2856  2859  2860  2869  2870  2871  2878  2896  2903
                2904    2909  2910  2920  2921  2929  2933  2934  2936  2937  2939  2940  2943
                2949    2957  2960  2961  2967  2968  2970  2978  2983  2984  2990  2991  2994
                2995    2996  2997  2998  3001  3007  3008  3019  3020  3038  3044  3045  3051
                3074    3075  3077  3078  3079  3080  3083  3084  3085  3086  3089  3090  3092
                3093    3095  3096  3098  3099  3104  3105  3108  3109  3111  3112  3125  3126
                3128    3137  3138  3139  3140  3157  3158  3160  3161  3162  3164  3165  3167
                3171    3172  3192  3193  3195  3196  3197  3199  3200  3201  3204  3208  3212
                3213    3214  3215  3219  3220  3225  3229  3230  3259  3265  3266  3270  3273
                3274    3279  3280  3282  3283  3285  3286  3288  3289  3291  3299  3307  3311
                3312    3316  3319  3322  3323  3334  3341  3408  3409  3410  3411  3412  3413
                3421    3424  3425  3429  3430  3435  3439  3440  3452  3457  3458  3467  3506
                3507    3508  3509  3512  3514  3515  3517  3518  3521  3522  3523  3524  3525
                3529    3530  3553  3554  3555  3556  3558  3559  3562  3563  3564  3566  3567
                3570    3596  3597  3614  3615  3616  3617  3620  3623  3626  3627  3631  3635
                3636    3639  3640  3641  3642  3643  3644  3645  3646  3650  3651  3652  3653
                3654    3655  3656  3657  3663  3666  3667  3668  3669  3670  3671  3675  3678
```

```
                    2921  2933  2934  2960  2961  2967  2968  2983  2984  2990  2991  2994  2996
                    2997  2998  3001  3007  3008  3019  3020  3038  3044  3045  3074  3075  3077
                    3078  3079  3080  3083  3084  3085  3086  3089  3090  3092  3093  3095  3096
                    3098  3099  3104  3105  3108  3109  3111  3112  3125  3126  3128  3137  3138
                    3139  3140  3157  3158  3160  3162  3164  3165  3167  3171  3172  3192  3193
                    3195  3197  3199  3201  3204  3208  3212  3213  3214  3215  3219  3220  3225
                    3229  3230  3259  3261  3265  3266  3273  3274  3279  3280  3282  3285  3285
                    3286  3288  3289  3311  3312  3316  3319  3322  3323  3334  3341  3408  3409
                    3410  3411  3412  3413  3421  3424  3425  3429  3430  3439  3440  3457  3458
                    3471  3476  3506  3507  3508  3512  3522  3523  3524  3525  3529  3530  3532
                    3553  3554  3596  3597  3614  3615  3616  3617  3620  3623  3626  3627  3635
                    3636  3639  3640  3641  3642  3643  3644  3645  3646  3650  3651  3652  3653
                    3654  3655  3656  3657  3663  3666  3667  3668  3669  3670  3671  3675  3678
                    3682  3683  3691  3692  3693  3694  3696
$F$OR = 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  2638  2701
                    2708  2758  2852  2873  2880  2898  3003  3040  3130  3169  3206  3210  3227
                    3261  3272  3309  3318  3321  3336  3343  3423  3437  3469  3622  3625  3665
                    3677  3680
$F$RTI= 000350          1#
$F$RTN= 000300          1#  3384  3483  3492  3534  3542  3580  3602  3709  3717  3753
$F$SEL= 000140          1#
$F$THE= 000330          1#  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  2053
                    2056  2079  2101  2131  2139  2184  2205  2229  2261  2279  2311  2320  2366
                    2387  2397  2436  2494  2555  2559  2597  2641  2701  2708  2761  2852  2873
                    2880  2901  3003  3043  3130  3169  3206  3210  3227  3264  3272  3309  3318
                    3321  3336  3343  3423  3437  3469  3622  3625  3665  3677  3680
$F$TRU= 000404          1#
$F$UNT= 000130          1#  1159  1183  2005  2019  3116  3118  3618  3660
$F$WHI= 000120          1#  1093  1094  1101  1217  1305  1695  1828  1902  1973  2046  2047  2072
                    2128  2308  2360  2361  2381  2395  2638  2758  2898  3040  3261
$F$YES= 000402          1#  1094  1105  1107  1114  1117  1119  1122  1125  1128  1131  1134  1136
                    1149  1152  1153  1154  1155  1156  1158  1162  1165  1169  1175  1179  1182
                    1189  1191  1192  1193  1215  1217  1222  1224  1229  1231  1237  1243  1245
                    1247  1253  1259  1262  1264  1270  1276  1279  1283  1289  1303  1305  1310
                    1312  1317  1319  1325  1331  1333  1335  1341  1347  1350  1352  1358  1364
                    1367  1371  1377  1399  1401  1407  1413  1415  1417  1423  1429  1432  1434
                    1440  1446  1449  1453  1459  1475  1477  1483  1489  1491  1493  1499  1505
                    1508  1510  1516  1522  1525  1529  1535  1559  1560  1571  1591  1595  1598
                    1602  1605  1609  1613  1615  1617  1619  1623  1625  1631  1642  1661  1663
                    1674  1693  1695  1700  1702  1706  1711  1713  1721  1735  1740  1749  1753
                    1758  1761  1775  1780  1782  1784  1787  1794  1795  1797  1801  1802  1826
                    1828  1833  1835  1840  1843  1847  1862  1867  1870  1876  1882  1888  1900
                    1902  1907  1909  1914  1916  1922  1935  1937  1942  1951  1953  1959  1971
                    1973  1978  1980  1982  1986  1988  1994  1996  1998  2004  2007  2011  2012
                    2016  2017  2025  2032  2036  2038  2047  2051  2054  2061  2066  2068  2070
                    2077  2083  2087  2089  2097  2099  2105  2108  2126  2128  2133  2135  2137
                    2141  2143  2147  2152  2159  2169  2179  2182  2188  2194  2196  2202  2203
                    2210  2217  2219  2224  2227  2234  2239  2241  2246  2251  2259  2265  2272
                    2274  2277  2283  2287  2306  2308  2313  2315  2318  2322  2324  2328  2330
                    2332  2334  2345  2348  2351  2354  2357  2359  2361  2364  2369  2370  2375
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
                    2377    2378    2385    2389    2393    2395    2399    2407    2408    2410    2414    2416    2425
                    2427    2429    2433    2434    2440    2492    2496    2498    2504    2508    2511    2514    2515
                    2516    2517    2518    2520    2533    2543    2553    2557    2563    2570    2572    2577    2580
                    2583    2595    2601    2608    2612    2614    2615    2616    2636    2638    2643    2649    2650
                    2651    2652    2653    2656    2658    2661    2664    2674    2688    2697    2699    2705    2706
                    2713    2719    2721    2728    2733    2736    2740    2742    2743    2744    2756    2758    2763
                    2765    2769    2772    2786    2809    2812    2813    2818    2828    2831    2832    2836    2845
                    2847    2850    2856    2860    2862    2870    2871    2876    2878    2883    2896    2898    2904
                    2906    2910    2921    2934    2957    2961    2963    2968    2978    2984    2986    2991    2996
                    2998    3001    3008    3010    3020    3038    3040    3045    3047    3075    3078    3080    3084
                    3086    3090    3093    3096    3099    3105    3109    3112    3126    3128    3134    3138    3140
                    3158    3162    3165    3167    3172    3173    3193    3197    3201    3204    3208    3213    3215
                    3216    3220    3221    3225    3230    3234    3259    3261    3266    3268    3270    3274    3276
                    3280    3283    3286    3289    3299    3304    3307    3312    3313    3316    3319    3323    3324
                    3326    3328    3334    3339    3341    3346    3409    3411    3413    3421    3425    3426    3430
                    3431    3435    3440    3441    3458    3467    3472    3507    3512    3523    3525    3530    3554
                    3597    3615    3617    3620    3623    3627    3628    3636    3637    3640    3642    3644    3646
                    3647    3651    3653    3655    3657    3658    3663    3667    3669    3671    3675    3678    3683
                    3684    3686    3688    3692    3694
$GDADR  001120      9270
$GDDAT  001124      9290
$GET42  011400     3781#
$GTSWR  012232     3958#    4431
$HD   = 000000      627
$HIBTS  001000      903#
$ICNT   001104      920#    4235*   4236    4238*   4249
$IFLEV= 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#   2306#   2315#   2318#   2324#   2364#   2378#   2385#   2393#   2408#   2410#   2434#   2440#
                    2492#   2498#   2553#   2557#   2570#   2572#   2595#   2601#   2636#   2643#   2699#   2706#   2719#
                    2721#   2756#   2765#   2809#   2813#   2828#   2832#   2850#   2862#   2871#   2876#   2878#   2883#
                    2896#   2906#   2957#   2963#   2978#   2986#   3001#   3010#   3038#   3047#   3078#   3128#   3134#
                    3167#   3173#   3204#   3221#   3225#   3234#   3259#   3268#   3270#   3276#   3299#   3304#   3307#
                    3328#   3334#   3339#   3341#   3346#   3421#   3431#   3435#   3441#   3467#   3472#   3620#   3658#
                    3663#   3688#
$ILLUP  011610      3799    3815    3834#
$INTAG  001135       934#   3986    4075
$ISKO = 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    2636#   2643    2699#   2721    2756#   2765    2809#
                    2813    2828#   2832    2850#   2862    2871#   2876    2878#   2883    2896#   2906    2957#   2963
                    2978#   2986    3001#   3010    3038#   3047    3128#   3134    3167#   3173    3204#   3221    3225#
                    3234    3259#   3268    3270#   3276    3299#   3304    3307#   3328    3334#   3339    3341#   3346
                    3421#   3431    3435#   3441    3467#   3472    3620#   3658    3663#   3688
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
$ISK1 = 000001          1598#  1617   1775#  1780   1787#  1795   2054#  2068   2393#  2408   2557#  2570   2706#
                        2719   3208#  3216   3316#  3326   3623#  3637   3675#  3686
$ISK2 = 000001          1605#  1615   3319#  3324   3678#  3684
$ITEMB  001114           924#  3728   4157*  4165   4186
$LF     001172           951#  3936   4060   4069   4186
$LFLG   013201          4119*  4125#
$LOCTA= 177777            1#   1093   1095   1101   1102   1159   1170   1175   1184   1216   1218   1219   1224
                        1232   1237   1248   1253   1265   1270   1284   1289   1304   1306   1307   1312   1320
                        1325   1336   1341   1353   1358   1372   1377   1402   1407   1418   1423   1435   1440
                        1454   1459   1478   1483   1494   1499   1511   1516   1530   1535   1561   1571   1592
                        1595   1596   1599   1602   1603   1606   1609   1610   1615   1617   1619   1626   1642
                        1664   1674   1694   1696   1697   1702   1707   1735   1740   1754   1761   1762   1775
                        1780   1782   1785   1788   1795   1797   1798   1802   1805   1807   1808   1827   1829
                        1830   1835   1862   1870   1883   1888   1901   1903   1904   1909   1937   1942   1954
                        1959   1972   1974   1975   1980   1983   1988   2005   2008   2012   2013   2017   2020
                        2022   2025   2026   2038   2046   2048   2052   2055   2068   2070   2072   2073   2078
                        2089   2100   2108   2127   2129   2130   2135   2138   2143   2183   2196   2204   2219
                        2228   2241   2260   2274   2278   2287   2307   2309   2310   2315   2319   2324   2337
                        2338   2340   2342   2360   2362   2365   2370   2371   2378   2381   2382   2386   2389
                        2390   2394   2396   2399   2400   2408   2410   2418   2419   2435   2440   2493   2498
                        2554   2558   2563   2564   2570   2572   2596   2601   2637   2639   2640   2643   2700
                        2707   2713   2714   2719   2721   2757   2759   2760   2765   2790   2791   2793   2795
                        2809   2813   2828   2832   2851   2862   2864   2865   2872   2876   2879   2883   2897
                        2899   2900   2906   2937   2938   2940   2942   2957   2963   2978   2986   3002   3010
                        3013   3014   3039   3041   3042   3047   3116   3119   3121   3122   3129   3134   3168
                        3173   3205   3209   3216   3221   3226   3234   3260   3262   3263   3268   3271   3276
                        3299   3304   3308   3313   3314   3317   3320   3324   3326   3328   3335   3339   3342
                        3346   3383   3418   3422   3426   3427   3431   3436   3441   3445   3447   3459   3460
                        3468   3470   3472   3475   3483   3485   3491   3515   3516   3518   3520   3526   3527
                        3531   3534   3535   3541   3556   3557   3559   3561   3564   3565   3567   3569   3571
                        3572   3574   3575   3580   3581   3601   3618   3621   3624   3628   3629   3637   3647
                        3648   3658   3661   3664   3676   3679   3684   3686   3688   3695   3709   3710   3716
                        3753   3754
$LPADR  001106           921#  1048*  4226*  4242*  4247   4249
$LPERR  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*  2655*  4176   4226   4243*  4249
$LSTCN= 177777            1#
$LSTIN= 000000            1#   1094   1095   1096   1101   1102   1104   1105   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
```

CROSS REFERENCE TABLE -- USER SYMBOLS

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	2607	2608	2611
2612	2613	2614	2615	2616	2636	2637	2638	2639	2640	2648	2649	2650
2651	2652	2653	2655	2656	2657	2658	2660	2661	2663	2664	2673	2674
2678	2679	2680	2681	2682	2683	2684	2685	2687	2688	2690	2691	2692
2693	2694	2696	2697	2699	2700	2701	2704	2705	2706	2707	2708	2713
2714	2727	2728	2732	2733	2735	2736	2739	2740	2741	2742	2743	2744
2756	2757	2758	2759	2760	2762	2763	2768	2769	2771	2772	2781	2782
2783	2784	2785	2786	2789	2790	2791	2792	2793	2794	2795	2796	2801
2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	2817	2818
2820	2821	2822	2823	2824	2825	2826	2827	2828	2829	2830	2831	2835
2836	2843	2844	2845	2846	2847	2850	2851	2852	2855	2856	2859	2860
2864	2865	2869	2870	2871	2872	2873	2878	2879	2880	2896	2897	2898
2899	2900	2903	2904	2909	2910	2920	2921	2929	2930	2931	2932	2933
2934	2936	2937	2938	2939	2940	2941	2942	2943	2949	2950	2951	2952
2953	2954	2955	2956	2957	2958	2960	2961	2967	2968	2970	2971	2972
2973	2974	2975	2976	2977	2978	2979	2983	2984	2990	2991	2994	2995
2996	2997	2998	3001	3002	3003	3007	3008	3013	3014	3019	3020	3038
3039	3040	3041	3042	3044	3045	3051	3052	3053	3054	3074	3075	3077
3078	3079	3080	3083	3084	3085	3086	3089	3090	3092	3093	3095	3096
3098	3099	3104	3105	3108	3109	3111	3112	3118	3119	3120	3121	3122
3125	3126	3128	3129	3130	3137	3138	3139	3140	3157	3158	3160	3161
3162	3164	3165	3167	3168	3169	3171	3172	3192	3193	3195	3196	3197
3199	3200	3201	3204	3205	3206	3208	3209	3210	3212	3213	3214	3215
3219	3220	3225	3226	3227	3229	3230	3259	3260	3261	3262	3263	3265
3266	3270	3271	3272	3273	3274	3279	3280	3282	3283	3285	3286	3288
3289	3291	3292	3293	3294	3295	3296	3297	3298	3299	3300	3307	3308

```
                    3309    3311    3312    3313    3314    3316    3317    3318    3319    3320    3321    3322    3323
                    3334    3335    3336    3341    3342    3343    3408    3409    3410    3411    3412    3413    3421
                    3422    3423    3424    3425    3426    3427    3429    3430    3435    3436    3437    3439    3440
                    3444    3445    3446    3447    3448    3452    3453    3454    3455    3456    3457    3458    3459
                    3460    3467    3468    3469    3470    3471    3474    3475    3476    3484    3485    3486    3487
                    3506    3507    3508    3509    3510    3511    3512    3514    3515    3516    3517    3518    3519
                    3520    3521    3522    3523    3524    3525    3526    3527    3529    3530    3531    3532    3536
                    3537    3553    3554    3555    3556    3557    3558    3559    3560    3561    3562    3563    3564
                    3565    3566    3567    3568    3569    3570    3571    3572    3574    3575    3582    3583    3596
                    3597    3614    3615    3616    3617    3620    3621    3622    3623    3624    3625    3626    3627
                    3628    3629    3631    3632    3635    3636    3639    3640    3641    3642    3643    3644    3645
                    3646    3647    3648    3650    3651    3652    3653    3654    3655    3656    3657    3660    3661
                    3662    3663    3664    3665    3666    3667    3668    3669    3670    3671    3675    3676    3677
                    3678    3679    3680    3682    3683    3691    3692    3693    3694    3695    3696    3711    3712
                    3755    3756
#LSTST= 177777        1#
#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    2640    2641    2643    2644    2714    2715
                    2719    2720    2721    2722    2760    2761    2765    2766    2791    2792    2793    2794    2813
                    2814    2832    2833    2862    2863    2865    2866    2876    2877    2883    2884    2900    2901
                    2906    2907    2938    2939    2940    2941    2963    2964    2986    2987    3010    3011    3014
                    3015    3042    3043    3047    3048    3116    3117    3122    3123    3134    3135    3173    3174
                    3216    3217    3221    3222    3234    3235    3263    3264    3268    3269    3276    3277    3304
                    3305    3314    3315    3324    3325    3326    3327    3328    3329    3339    3340    3346    3347
                    3383    3384    3418    3419    3427    3428    3431    3432    3441    3442    3460    3461    3472
                    3473    3483    3484    3485    3486    3491    3492    3516    3517    3518    3519    3527    3528
                    3534    3535    3536    3541    3542    3557    3558    3559    3560    3565    3566    3567    3568
                    3572    3573    3575    3576    3580    3581    3582    3601    3602    3618    3619    3629    3630
                    3637    3638    3648    3649    3658    3659    3684    3685    3686    3687    3688    3689    3709
                    3710    3711    3716    3717    3753    3754    3755
#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    2634    2754    2894    3035    3257    3863
                  4163    4241
#MAMS1  001224        978#
#MAMS2  001230        986#
#MAMS3  001234        989#
#MAMS4  001240        992#
#MBADR  001002        904#
#MFLG   013200    4079*   4085    4120*   4124#
```

```
$MNEW   012725    3961  4073#
$MSGAD  001210     964# 4095# 4098
$MSGLG  001212     965# 4100#
$MSGTY  001174     958# 4093  4101# 4113  4117#
$MSWR   012714    3958  4071#
$MTYP1  001225     979#
$MTYP2  001231     987#
$MTYP3  001235     990#
$MTYP4  001241     993#
$MXCNT  013662    4239  4249#
$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# 2636# 2643# 2699# 2706# 2713  2719#
                  2721# 2756# 2765# 2789# 2809# 2813# 2828# 2832# 2850# 2862# 2864# 2871# 2876#
                  2878# 2883# 2896# 2906# 2936# 2957# 2963# 2978# 2986# 3001# 3010# 3013# 3038#
                  3047# 3116# 3118# 3128# 3134# 3167# 3173# 3204# 3208# 3216# 3221# 3225# 3234#
                  3259# 3268# 3270# 3276# 3299# 3304# 3307# 3313  3316# 3319# 3324# 3326# 3328#
                  3334# 3339# 3341# 3346# 3384# 3418# 3421# 3426  3431# 3435# 3441# 3444  3459#
                  3467# 3472# 3483# 3492# 3514# 3526# 3534# 3542# 3555# 3563# 3571# 3574# 3580#
                  3602# 3618# 3620# 3623# 3628  3637# 3647  3658# 3660# 3663# 3675# 3678# 3684#
                  3686# 3688# 3709# 3717# 3753#
$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  2636# 2643  2699# 2721  2756# 2765  2789# 2864  2871#
                  2876  2878# 2883  2896# 2906  2936# 3013  3038# 3047  3116# 3118  3128# 3134
                  3167# 3173  3204# 3221  3225# 3234  3259# 3268  3270# 3276  3299# 3304  3307#
                  3313  3328  3334# 3339  3341# 3346  3384# 3483  3492# 3534  3542# 3580  3602#
                  3709  3717# 3753
$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
                  2706# 2713  2719  2809# 2813  2828# 2832  2850# 2862  2957# 2963  2978# 2986
                  3001# 3010  3208# 3216  3316# 3326  3418# 3444  3459  3467# 3472  3514# 3526
                  3555# 3574  3618# 3660  3663# 3688
$NSK2  = 000110   1605# 1609  1615  1775# 1780  1787# 1795  2054# 2068  2364# 2370  2378  2393#
                  2399  2408  3319# 3324  3421# 3426  3431  3435# 3441  3563# 3571  3620# 3647
                  3658  3675# 3686
$NSK3  = 000110   3623# 3628  3637  3678# 3684
$NULL   001154     942# 3890  3936
$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  2626# 2628  2748#
                  2750  2889# 3029# 3031  3248# 3250  3353#
```

```
$OCNT   014332          4349*  4378*  4391#
$OMODE  014334          4344*  4348*  4353   4356*  4367*  4393#
$OVER   013646          4203   4219   4227   4237   4246#
$PASS   001202           961#  1066*  3678   3769*  3770*  3778   3791   4233   4250
$PASTM  001006           906#
$POWER  011616          3830   3837#
$PWRAD  011604          3832#
$PWRDN  011444          1042   3799#  3827
$PWRMG  011600          3830#
$PWRUP  011516          3809   3815#
$QUES   001170           949#  3936   4004   4053   4069   4186
$RDCHR  012444          4017#  4434
$RDDEC- ****** U        4436
$RDLIN  012564          4045#  4435
$RDOCT- ****** U        4436
$RDSZ = 000010          4038#
$RTNAD  011422          3790#
$R2A  = ****** U        4436
$SAVLE- 177777             1#  1101#  1102#  1807#  1808#  2072#  2073#  2339#  2343#  2381#  2382#  2792#  2796#
                        2939#  2943#  3459#  3460#  3517#  3521#  3558#  3562#  3566#  3570#
$SAVRE- ****** U        4436
$SAVRG  011614          3808*  3816   3817*  3818*  3836#
$SCOPE  013404          1036   4200#
$SETUP- 000137          1027#  1035   1036   1038   1040   1042   1044   1045   1046   1048   1075   1078   3767
                        3941   4075   4147   4173   4181   4201
$SSKO = 050006          1101#  1102   1807#  1808   2072#  2073   2339#  2343   2381#  2382   2792#  2796   2939#
                        2943   3459#  3460   3517#  3521   3558#  3562   3566#  3570
$STUP = 177777          1027#
$SVLAD  013612          4211   4740#
$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   2633   2753   2893   3034   3256   3357   3762   3768
                        3783   3789   3791   3833   4138   4139   4140   4141   4142   4151   4158   4170   4174
                        4186   4192   4193   4194   4195   4196   4202   4214   4216   4217   4220   4221   4222
                        4229   4230   4231   4243   4246   4249
$SWREG  001216          9690   1069#
$SWRMK- 000000           637    638    4196   4197   4218
$TAGLE- 177777             1#  10940  10960  11010  11600  11710  11750  11830  12200  12240  12330  12370  12490
                        12530  12660  12700  12850  12890  13080  13120  13210  13250  13370  13410  13540  13580
                        13730  13770  14030  14070  14190  14230  14360  14400  14550  14590  14790  14830  14950
                        14990  15120  15160  15310  15350  15620  15710  15930  15960  15970  16000  16030  16040
                        16070  16100  16110  16150  16170  16190  16270  16420  16650  16740  16980  17020  17080
                        17360  17400  17550  17620  17630  17760  17800  17820  17860  17890  17950  17980  17990
                        18020  1804   18070  18310  18350  18630  18700  18840  18880  19050  19090  19380  19420
                        19550  19590  19760  19800  19840  19880  20060  20090  20130  20140  20170  20190  20270
                        20380  20470  20490  20530  20560  20680  20700  20720  20790  20890  21010  21080  21310
                        21350  21390  21430  21840  21960  22050  22190  22290  22410  22610  22740  22790  22870
                        23110  23150  23200  23240  23380  23400  23430  23610  23630  23660  23710  23720  23780
                        23810  23870  23900  23910  23970  24000  24010  24080  24100  24180  24190  24360  24400
                        24940  24980  25550  25590  25640  25650  25700  25720  25970  26010  26410  26430  27010
                        27080  27140  27150  27190  27210  27610  27650  27910  27930  27960  28100  26130  28290
                        28320  28520  28620  28640  28650  28730  28760  28800  28830  29010  29060  29380  29400
                        29430  29580  29630  29790  29860  30030  30100  30130  30140  30430  30470  31170  31180
                        31300  31340  31690  31730  32060  32100  32160  32210  32270  32340  32640  32680  32720
                        32760  33000  33040  33090  33140  33150  33180  33210  33240  33260  33280  33360  33390
```

	3343#	3346#	3419#	3423#	3427#	3428#	3431#	3437#	3441#	3444	3459#	3469#	3472#
	3516#	3518#	3521#	3526#	3527#	3557#	3559#	3562#	3565#	3567#	3570#	3571#	3572#
	3574#	3575#	3619#	3622#	3625#	3629#	3630#	3637#	3648#	3649#	3658#	3660#	3665#
	3677#	3680#	3684#	3686#	3688#								
$TAGNU= 050002	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#	2637	2639	2640	2641#	2700	2701#	2707
	2708#	2713	2715#	2757	2759	2760	2761#	2790	2791#	2792#	2795	2796#	2809
	2810#	2828	2829#	2851	2852#	2872	2873#	2879	2880#	2897	2899	2900	2901#
	2937	2938#	2939#	2942	2943#	2957	2958#	2978	2979#	3002	3003#	3039	3041
	3042	3043#	3116	3117#	3119	3122	3123#	3129	3130#	3168	3169#	3205	3206#
	3209	3210#	3226	3227#	3260	3262	3263	3264#	3271	3272#	3299	3300#	3308
	3309#	3313	3315#	3317	3318#	3320	3321#	3335	3336#	3342	3343#	3383#	3384#
	3418	3419#	3422	3423#	3426	3428#	3436	3437#	3468	3469#	3491#	3492#	3515
	3516#	3517#	3520	3521#	3541#	3542#	3556	3557#	3558#	3561	3562#	3564	3565#
	3566#	3569	3570#	3601#	3602#	3618	3619#	3621	3622#	3624	3625#	3628	3630#
	3647	3649#	3664	3665#	3676	3677#	3679	3680#	3716#	3717#			
$TEMP = 000300	11010#	11020#	11040#	11050#	11060#	11070#	11130#	11140#	11160#	11170#	11180#	11190#	11210#
	11220#	11230#	11250#	11260#	11280#	11290#	11310#	11320#	11340#	11350#	11360#	11480#	11490#
	11510#	11520#	11530#	11540#	11550#	11560#	11570#	11580#	11610#	11620#	11640#	11650#	11750#
	11780#	11790#	11800#	11820#	11830#	1184	11880#	11890#	11910#	11920#	11930#	12210#	12220#
	12240#	12280#	12290#	12370#	12420#	12430#	12440#	12450#	12530#	12580#	12590#	12610#	12620#
	12700#	12750#	12760#	12780#	12790#	12890#	13090#	13100#	13120#	13160#	13170#	13250#	13300#
	13310#	13320#	13330#	13410#	13460#	13470#	13490#	13500#	13580#	13630#	13640#	13660#	13670#
	13770#	13980#	13990#	14070#	14120#	14130#	14140#	14150#	14230#	14280#	14290#	14310#	14320#
	14400#	14450#	14460#	14480#	14490#	14590#	14740#	14750#	14830#	14880#	14890#	14900#	14910#
	14990#	15040#	15050#	15070#	15080#	15160#	15210#	15220#	15240#	15250#	15350#	15580#	15590#
	15710#	15950#	15960#	16020#	16030#	16090#	16100#	16120#	16130#	16150#	16170#	16190#	16220#
	16230#	16300#	16310#	16420#	16600#	16610#	16740#	16990#	17000#	17020#	17050#	17060#	17100#
	17110#	17120#	17130#	17200#	17210#	17400#	17480#	17490#	17570#	17580#	17610#	17620#	17800#
	17820#	17930#	17940#	17950#	17970#	17980#	18000#	18010#	18020#	18040#	1805	18070#	18080#
	18320#	18330#	18350#	18390#	18400#	18420#	18430#	18460#	18470#	18660#	18670#	18700#	18750#
	18760#	18880#	19060#	19070#	19090#	19130#	19140#	19150#	19160#	19210#	19220#	19340#	19350#
	19420#	19500#	19510#	19590#	19770#	19780#	19800#	19850#	19860#	19880#	19930#	19940#	19950#
	19960#	19970#	19980#	20030#	20040#	20100#	20110#	20120#	20130#	20150#	20160#	20170#	20190#
	2022	20310#	20320#	20350#	20360#	20340#	20600#	20610#	20650#	20660#	20680#	20700#	20720#
	20730#	20820#	20830#	20860#	20870#	20890#	20960#	20970#	21040#	21050#	21080#	21320#	21330#
	21350#	21400#	21410#	21430#	21460#	21470#	21510#	21520#	21580#	21590#	21680#	21690#	21780#
	21790#	21870#	21880#	21930#	21940#	21960#	22010#	22020#	22090#	22100#	22160#	22170#	22190#
	22230#	22240#	22330#	22340#	22380#	22390#	22410#	22450#	22460#	22500#	22510#	22640#	22650#
	22710#	22720#	22740#	22820#	22830#	22870#	23120#	23130#	23150#	23210#	23220#	23240#	23270#
	23280#	23290#	23300#	23310#	23320#	23330#	23340#	23400#	23430#	23440#	23450#	23470#	23480#
	23500#	23510#	23530#	23540#	23560#	23570#	23580#	23590#	23680#	23690#	23700#	23710#	23740#
	23750#	23760#	23770#	23780#	23810#	23820#	23890#	23900#	23990#	24000#	24060#	24070#	24080#

E10


```
                    24100  24130  24140  24150  24160  24180  24190  24210  24250  24260  24270  24280  24290
                    24320  24330  24400  24950  24960  24980  25030  25040  25070  25080  25100  25110  25130
                    25140  25150  25160  25170  25180  25190  25200  25320  25330  25420  25430  25630  25640
                    25700  25720  25760  25770  25790  25800  25820  25830  26010  26070  26080  26110  26120
                    26140  26150  26160  26430  26480  26490  26500  26510  26520  26530  26550  26560  26570
                    26580  26600  26610  26630  26640  26730  26740  26870  26880  26960  26970  27040  27050
                    27130  27140  27190  27210  27270  27280  27320  27330  27350  27360  27390  27400  27420
                    27430  27440  27620  27630  27650  27680  27690  27710  27720  27850  27860  27930  27960
                    28110  28120  28130  28170  28180  28300  28310  28320  28350  28360  28430  28450  28460
                    28470  28550  28560  28590  28600  28620  28640  28650  28690  28700  28760  28830  29030
                    29040  29060  29090  29100  29200  29210  29330  29340  29400  29430  29600  29610  29630
                    29670  29680  29830  29840  29860  29900  29910  29940  29960  29970  29980  30070  30080
                    30100  30130  30140  30190  30200  30440  30450  30470  30740  30750  30770  30780  30790
                    30800  30830  30840  30850  30860  30890  30900  30920  30930  30950  30960  30980  30990
                    31040  31050  31080  31090  31110  31120  31180  3121   31250  31260  31340  31370  31380
                    31390  31400  31570  31580  31600  31620  31640  31650  31710  31720  31730  31920  31930
                    31950  31970  31990  32010  32120  32130  32140  32150  32160  32190  32200  32210  32290
                    32300  32340  32650  32660  32680  32730  32740  32760  32790  32800  32820  32830  32850
                    32860  32880  32890  33040  33110  33120  33130  33140  33220  33230  33240  33260  33280
                    33390  33460  34080  34090  34100  34110  34120  34130  34240  34250  34260  34270  34290
                    34300  34310  34390  34400  34410  34440  3445   3447   34570  34580  34590  34600  34720
                    34830  35060  35070  35080  35120  35180  35210  35220  35230  35240  35250  35260  35270
                    35290  35300  35340  35330  35540  35590  35620  35670  35700  35710  35720  35740  35750
                    35800  35960  35970  36140  36150  36160  36170  36260  36270  36280  36290  36350  36360
                    36370  36390  36400  36410  36420  36430  36440  36450  36460  36470  36480  36500  36510
                    36520  36530  36540  36550  36560  36570  36580  36600  3661   36660  36670  36680  36690
                    36700  36710  36820  36830  36840  36860  36880  36910  36920  36930  36940  37090  37530
$TESTN  001200       9600   1146*  1212*  1300*  1389*  1471*  1552*  1585*  1654*  1690*  1823*  1897*  1969*
                    2123*  2301*  2487*  2634*  2734*  2894*  3035*  3257*  3722   4241*
$TIMES  001160       9460   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*  2633*  2753*
                    2762*  2859*  2893*  2903*  2960*  2983*  3007*  3034*  3044*  3256*  3265*  3273*  3357*
                    3768*  4229*  4236   4239*  4249
$TKB    001146       9390   3909   3916   3939   3950   3967   4021   4027
$TKS    001144       9380   3907   3914   3939   3948   3964   3988*  4019   4025
$TN   = 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   2626   2633*  2634   2748   2753*  2754   2763   2860
                    2889   2893*  2894   2904   2961   2984   3008   3029   3034*  3035   3045   3141   3248
                    3256*  3257   3266   3274   3348   3353   3357*
$TPB    001152       9410   3925*  3936
$TPFLG  001157       9450   3857   3936
$TPS    001150       9400   3923   3936
$TRAP   014336       1040   4402*
$TRAP2  014360       4413*  4424
$TRP  = 000012       4417*  4426*  4427*  4428*  4429*  4430*  4431   4432*  4433   4434*  4435*  4436*
$TRPAD  014372       4407   4424*
$TSKO = 050007       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    2641#   2643    2701#   2721    2761#   2765    2791#
                    2793    2796#   2865    2873#   2876    2880#   2883    2901#   2906    2938#   2940    2943#   3014
                    3043#   3047    3117#   3118    3130#   3134    3169#   3173    3206#   3221    3227#   3234    3264#
                    3268    3272#   3276    3300#   3304    3309#   3314    3315#   3328    3336#   3339    3343#   3346
                    3419#   3459    3469#   3472    3516#   3518    3521#   3527    3557#   3559    3562#   3575    3619#
                    3660    3665#   3688
$TSK1 = 050010      1096#   1101    1171#   1175    1600#   1603    1604#   1617    1708#   1804    1807    2009#   2013
                    2014#   2017    2049#   2072    2343#   2418    2559#   2564    2565#   2570    2708#   2714    2715#
                    2719    2796#   2864    2943#   3013    3210#   3216    3318#   3326    3419#   3444    3459    3521#
                    3526    3562#   3574    3622#   3648    3649#   3658    3677#   3686
$TSK2 = 050011      1607#   1610    1611#   1615    1736#   1740    1755#   1762    1763#   1782    1786#   1798    1799#
                    1802    2053#   2070    2361#   2381    2387#   2390    2391#   2410    2810#   2813    2829#   2832
                    2852#   2862    2958#   2963    2979#   2986    3003#   3010    3321#   3324    3423#   3427    3428#
                    3431    3437#   3441    3565#   3567    3570#   3572    3625#   3629    3630#   3637    3680#   3684
$TSK3 = 050006      1776#   1780    1789#   1795    2056#   2068    2363#   2381    2397#   2400    2401#   2408    3570#
                    3571
$TSK4 = 050123      2366#   2371    2372#   2378
$TSTM   001004       905#
$TSTNM  001102       918#   3767#   4150    4186    4191    4218    4240#   4241    4246    4250
$TTYIN  012672      4046    4047    4064    4068#
$TYPBN= ****** U    4430
$TYPDS  013664      4262#   4429
$TYPE   011626      3857#   4106    4417    4425
$TYPEC  012040      3887    3894    3901    3906#   3990
$TYPEX  012160      3929    3931    3934#
$TYPOC  014134      4347#   4426
$TYPON  014150      4346    4349#   4428
$TYPOS  014110      4342#   4427
$UNIT   001206       963#
$UNITM  001010       907#
$USWR   001220       970#   1215    1303    1598    1605    1826    1900    1973    2126    2137    2306    2308    2421
                    2636    2756    2895    2898    3038    3259    3261    3316    3508
$VECT1  001244       995#   3645
$VECT2  001246       996#
$XOFF = 000023      3911    3936
$XON  = 000021      3918    3936
$XTSTR  013416      4205#
$$ARGC= 000000        1#    3384#   3492#   3542#   3602#   3717#
$$BYTE= 000403        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#   2023#   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#   2636#   2638#   2699#   2706#   2756#   2758#   2850#
                    2871#   2878#   2896#   2898#   3001#   3038#   3040#   3128#   3167#   3204#   3208#   3225#   3259#
                    3261#   3270#   3307#   3316#   3319#   3334#   3341#   3421#   3435#   3467#   3620#   3623#   3663#
                    3675#   3678#
$$CASE= 000000        1#
$$DST = 000067        1#    2422#   3509#
$$ELOC= 000403        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#   1595    1598#   1602    1605#   1609    1615#   1617#   1619#   1625#   1642#
```

```
                    16630  16740  16930  17020  17350  17400  17530  1761   17750  17800  17820  17840  17870
                    17950  1797   18020  18260  18350  18620  18700  18820  18880  19000  19090  19370  19420
                    19530  19590  19710  19800  19820  19880  20070  2012   20170  20250  20380  20510  20540
                    20680  20700  20770  20890  20990  21080  21260  21350  21370  21430  21820  21960  22030
                    22190  22270  22410  22590  22740  22770  22870  23060  23150  23180  23240  23640  2370
                    23780  23850  2389   23930  2399   24080  24100  24340  24400  24920  24980  25530  25570
                    2563   25700  25720  25950  26010  26360  26430  26990  27060  2713   27190  27210  27560
                    27650  28090  28130  28280  28320  28500  28620  28710  28760  28780  28830  28960  29060
                    29570  29630  29780  29860  30010  30100  30380  30470  31280  31340  31670  31730  32040
                    32080  32160  32210  32250  32340  32590  32680  32700  32760  32990  33040  33070  3313
                    33160  33190  33240  33260  33280  33340  33390  33410  33460  34210  3426   34310  34350
                    34410  34670  34710  34720  34760  35320  36200  36230  3628   36370  3647   36580  36630
                    36750  36780  36840  36860  36880  36960
$$ERFL= 000000        1#
$$FLAG= 000001        1#   10930  10940  10960  11690  11710  11750  12150  12170  12200  12240  12310  12330
                    12370  12470  12490  12530  12640  12660  12700  12830  12850  12890  13030  13050  13080
                    13120  13190  13210  13250  13350  13370  13410  13520  13540  13580  13710  13730  13770
                    14010  14030  14070  14170  14190  14230  14340  14360  14400  14530  14550  14590  14770
                    14790  14830  14930  14950  14990  15100  15120  15160  15290  15310  15350  15600  15620
                    15710  15910  15930  15980  16000  16050  16070  16150  16170  16190  16250  16270  16420
                    16630  16650  16740  16930  16950  16980  17020  17350  17400  17530  17550  17750  17800
                    17820  17840  17860  17870  17890  17950  18020  18260  18280  18310  18350  18620  18700
                    18820  18840  18880  19000  19020  19050  19090  19370  19420  19530  19550  19590  19710
                    19730  19760  19800  19820  19840  19880  20070  20090  20170  20250  20270  20380  20460
                    20470  20490  20510  20530  20540  20560  20680  20700  20770  20790  20890  20990  21010
                    21080  21260  21280  21310  21350  21370  21390  21430  21820  21840  21960  22030  22050
                    22190  22270  22290  22410  22590  22610  22740  22770  22790  22870  23060  23080  23110
                    23150  23180  23200  23240  23600  23610  23630  23640  23660  23780  23850  23870  23930
                    23950  23970  24080  24100  24340  24360  24400  24920  24940  24980  25530  25550  25570
                    25590  25700  25720  25950  25970  26010  26360  26380  26410  26430  26990  27010  27060
                    27080  27190  27210  27560  27580  27610  27650  28090  28130  28280  28320  28500  28520
                    28620  28710  28730  28760  28780  28800  28830  28960  28980  29010  29060  29570  29630
                    29780  29860  30010  30030  30100  30380  30400  30430  30470  31280  31300  31340  31670
                    31690  31730  32040  32060  32080  32100  32160  32210  32250  32270  32340  32590  32610
                    32640  32680  32700  32720  32760  32990  33040  33070  33090  33160  33180  33190  33210
                    33240  33260  33280  33340  33360  33390  33410  33430  33460  34210  34230  34310  34350
                    34370  34410  34670  34690  34720  36200  36220  36230  36250  36370  36580  36630  36650
                    36750  36770  36780  36800  36840  36860  36880
$$FROM= 000000        1#   11090  17250  17670  18520  19260  21610  21710  22530  25220  25460  25900  26780
                    26900  27810  28010  28200  29290  29490  29700  30510  32910  34520  36310
$$GET4= 000000      37830
$$LOC = 011064        1#   10950  1096   11700  1171   11840  1185   12180  1219   12320  1233   12480  1249
                    12650  1266   12840  1285   13060  1307   13200  1321   13360  1337   13530  1354   13720
                    1373   14020  1403   14180  1419   14350  1436   14540  1455   14780  1479   14940  1495
                    15110  1512   15300  1531   15610  1562   15920  1593   15990  1600   16060  1607   16260
                    1627   16640  1665   16960  1697   17350  1736   17540  1755   17750  1776   17850  1786
                    17880  1789   18290  1830   18620  1863   18830  1884   19030  1904   19370  1938   19540
                    1955   19740  1975   19830  1984   20080  2009   20220  2023   20260  2027   20480  2049
                    20520  2053   20550  2056   20780  2079   21000  2101   21290  2130   21380  2139   21830
                    2184   22040  2205   22280  2229   22600  2261   22780  2279   23090  2310   23190  2320
                    23620  2363   23650  2366   23860  2387   23940  2395   23960  2397   24350  2436   24930
                    2494   25540  2555   25580  2559   25960  2597   26390  2640   27000  2701   27070  2708
                    27590  2760   28090  2810   28280  2829   28510  2852   28720  2873   28790  2880   28990
                    2900   29570  2958   29780  2979   30020  3003   30410  3042   31210  3122   31290  3130
                    31680  3169   32050  3206   32090  3210   32260  3227   32620  3263   32710  3272   32990
                    3300   33080  3309   33170  3318   33200  3321   33350  3336   33420  3343   34220  3423
```

```
                    3436#  3437    3468#  3469    3621#  3622    3624#  3625    3661#  3662    3664#  3665    3676#
                    3677   3679#   3680
$$LOCN= 000000         1#  3384#   3486   3492#   3536   3542#   3582   3602#   3711   3717#   3755
$$REG = 177777         1#
$$RETU= 000000         1#  3384#   3470#  3474#   3484   3492#   3535   3542#   3581   3602#   3710   3717#   3754
$$RTN1= 050000         1#  3384#   3470   3483    3492#  3531    3534   3542#   3580   3602#   3695   3709    3717#
                    3753
$$RTN2= 050001         1#  3384#   3475   3485    3492#  3535    3542#  3581    3602#   3710   3717#   3754
$$SRC = 000027         1#  2422#   3509#
$$TGSV= 000000         1#
$$TGS1= 000000         1#
$$TGS2= 000000         1#
$$TO  = 000000         1#  1109#   1110   1725#   1731   1767#   1773   1852#   1858   1926#   1932   2161#   2164
                    2171#  2174    2253#  2256    2522#  2528    2546#  2549    2590#   2593   2678#   2684   2690#
                    2693   2781#   2782   2783    2801#  2807    2820#  2826    2929#   2930   2931    2949#  2955
                    2970#  2976    3051#  3052    3053   3291#   3297   3452#   3455    3631#  3632
$$$TAG= 050000         1#  3383    3491   3541    3601   3716
$OFILL  014333      4343*  4347*   4357   4392#
$40CAT= ****** U    4160   4202
.       = 014416     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   2639
                    2640   2700    2701   2707    2708   2759    2760   2809    2810    2828   2829    2851   2852
                    2872   2873    2879   2880    2899   2900    2957   2958    2978    2979   3002    3003   3041
                    3042   3121    3122   3129    3130   3168    3169   3205    3206    3209   3210    3226   3227
                    3262   3263    3271   3272    3299   3300    3308   3309    3317    3318   3320    3321   3335
                    3336   3342    3343   3361#   3422   3423    3436   3437    3468    3469   3621    3622   3624
                    3625   3661    3662   3664    3665   3676    3677   3679    3680    3706#  3721#   3736#  3742#
                    3791   3795    3811   3835    3936   3939    4068#  4069    4075    4127#  4186    4249   4250
                    4316#
.$ASTA= ****** U    4079   4082
.$X   = 001000       892#   897
```

```
BGNHRD      1#
BGNHW       1#
BGNINI      1#
BGNMOD      1#  3380
BGNMSG      1#
BGNSFT      1#
BGNSRV      1#  3151  3186  3586
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
          2654
BGNSW       1#
BRESET      1#  1136  1280  1368  1450  1526  1568  1639  1671  1878  2602  3100  3330
CALL      1108  1724  1766  1851  1925  2161  2171  2253  2521  2546  2590  2677  2690  2780  2800
          2819  2928  2948  2969  3050  3290  3452  3630
CKLOOP      1#  1536  1573  1644  1676
CLRVEC      1#  1185  2608  2736
COMMEN      1#  7490
DEALLO    3483#  3534#  3580#  3709#  3753#
DEVREG      1#
DEVTYP      1#
DISPAT      1#
ELSE      1594  1601  1608  1760  1796  2011  2369  2388  2398  2562  2712  3312  3425  3627  3646
ENDCLN      1#
ENDCOM      1#  7490
ENDDO     1100  2071  2380
ENDHRD      1#
ENDHW       1#
ENDIF     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  2642  2718  2720  2764  2812  2831  2861  2875  2882  2905  2962  2985  3009
          3046  3133  3172  3215  3220  3233  3267  3275  3303  3323  3325  3327  3338  3345  3430
          3440  3471  3636  3657  3683  3685  3687
ENDINC    2417  2863  3012  3525  3570  3573
ENDINI      1#
ENDLOO    1806  3458
ENDMOD      1#
ENDMSG      1#
ENDRTN    3482  3533  3579  3708  3752
ENDSFT      1#
ENDSRV      1#  3179  3240  3597
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
          2722
ENDSW       1#
ENDTST      1#  1196  1291  1379  1461  1538  1575  1646  1678  1814  1890  1960  2114  2289  2472
          2618  2746  2884  3023  3243  3350
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  2709  2716  2856  2873  2880
          2979  3004  3131  3301  3336  3343
ERROR     6439  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    2710    2717    2857    2874
          2881    2958    2980    3005    3132    3302    3337    3344
ESCAPE      1#    7490
EXIF      1803
EXIFB     3443
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    2761    2858    2902    2959    2982    3006
          3043    3140    3264    3272    3347
GETPRI      1#    7490
GETSWR      1#    7490   10780
GPHARD      1#
GPRMA       1#
GPRMD       1#
GPRML       1#
HEADER      1#
IF        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    2635    2698    2705    2755    2849    2870
          2877    2895    3000    3037    3127    3166    3203    3207    3224    3258    3315    3318    3333    3340    3420
          3619    3622    3662    3674    3677
IFB       1590    1692    1970    2317    3269    3306    3434    3466
IF.ERR    1734    1774    1861    1936    2808    2827    2956    2977    3298
INCR      2335    2788    2935    3513    3562
INCRU     3554
LASTAD      1#
LET       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    2606    2611    2612    2614    2615    2648    2649    2650    2651    2652
          2655    2656    2659    2662    2672    2686    2695    2703    2726    2731    2734    2739    2740    2742    2743
          2762    2767    2770    2784    2810    2816    2829    2834    2842    2845    2854    2859    2868    2903    2908
          2919    2932    2960    2966    2983    2989    2993    2996    3007    3018    3044    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    3265    3273    3278    3281    3284    3287    3310    3321
          3407    3409    3411    3423    3428    3438    3456   3470#   3474#    3505    3507    3521    3523    3528   3531#
          3552    3595    3613    3615    3625    3634    3638    3640    3642    3644    3649    3651    3653    3655    3665
          3667    3669    3681    3690    3692   3695#
LOOP      1706    3417
MSG       1138#   1140   1202#    1204   1466#    1468   1683#    1685   1816#    1818   1962#    1964   2293#    2295   2477#
          2479   2626#    2628   2748#    2750   3029#    3031   3248#    3250
MULT        1#    7490
NEWTST      1#    7490    1138    1202    1295    1384    1466    1547    1580    1649    1683    1816    1892    1962    2118
          2293    2477    2626    2748    2889    3029    3248    3353
POINTE      1#
POP         1#    7490    1193    2616    2744    3576    3820    3821    4121    4122    4303
PRINTB      1#
PUSH        1#    7490    1186    2609    2737    3549    3801    3807    4082    4084    4105    4262
REPEAT    1158    2004    3115    3617
```

```
REPORT      1#    7490
RETURN   3469    3473    3530    3694
ROUTIN   3382    3490    3540    3600    3715
SCOPE     6440#  1144    1210    1298    1387    1469    1550    1583    1652    1688    1821    1895    1967    2121    2299
         2485    2632    2752    2892    3033    3255    3356    3766
SETPRI      1#    7490    1391    2535    2584    2666    2775    2923    3068
SETTRA   4417#   4426    4427    4428    4429    4431    4433    4434    4435
SETUP       1#    7490    1027
SETVEC      1#    1150    2512    2647
SKIP        1#    7490    1222    1310    1613    1700    1810    1833    1907    1978    1986    2036    2066    2087    2111
         2133    2141    2194    2217    2239    2272    2313    2322    2442    2496    2763    2860    2904    2961    2984
         3008    3045    3141    3266    3274    3348
SLASH       1#    7490
SPACE    7490
STARS       1#    7490    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    2623    2626    2631    2747    2748    2751    2888    2889    2891    3028
         3029    3032    3149    3153    3184    3186    3247    3248    3254    3353    3355    3381    3402    3489    3503
         3539    3548    3585    3592    3602    3611    3717    3758    3797    3813    3842    3938    3941    4009    4038
         4077    4134    4188    4252    4319    4396
STRUCT      1#
SWRSU       1#    7490    10500
TRMTRP   4417#
TYPBIN      1#    7490
TYPDEC      1#    7490    3374    3778
TYPNAM      1#    7490    1071
TYPNUM      1#    7490
TYPOCS      1#    7490    3729
TYPOCT      1#    7490    3362    3368    3722    3737    3743    3749    3959
TYPTXT      1#    7490    1096    1097    1098    3358    3364    3370    3718    3724    3733    3739    3745
UNTIL    1182    2018    3117    3659
WAITMS      1#    2160    2170    2252    2545    2589    2689    3451
WHILE    1092    2045    2359
$ADDON   10930   10940   10960   11010   11590   11600   11690   11710   12150   12170   12200   12310   12330   12470   12490
         12640   12660   12830   12850   13030   13050   13080   13190   13210   13350   13370   13520   13540   13710   13730
         14010   14030   14170   14190   14340   14360   14530   14550   14770   14790   14930   14950   15100   15120   15290
         15310   15600   15620   15910   15930   15950   15970   15980   16000   16020   16040   16050   16070   16090   16110
         16250   16270   16630   16650   16930   16950   16980   17070   17080   17350   17360   17530   17550   17610   17630
         17750   17760   17840   17860   17870   17890   17970   17990   18070   18260   18280   18310   18620   18630   18820
         18840   19000   19020   19050   19370   19380   19530   19550   19710   19730   19760   19820   19840   20050   20060
         20070   20090   20120   20140   20190   2024    20250   20270   20460   20470   20490   20510   20530   20540   20560
         20720   20770   20790   20990   21010   21260   21280   21310   21370   21390   21820   21840   22030   22050   22270
         22290   22590   22610   22770   22790   23060   23080   23110   23180   23200   23360   23380   23390   23430   23600
         23610   23630   23640   23660   23700   23720   23810   23650   23870   23890   23910   23930   23950   23970   23990
         24010   24340   24360   24920   24940   25530   25550   25570   25590   25630   25650   25950   25970   26360   26380
         26410   26990   27010   27060   27080   27130   27150   27560   27580   27610   27890   27910   27920   27960   28090
         28100   28280   28290   28500   28520   28710   28730   28780   28800   28960   28980   29010   29360   29380   29390
         29430   29570   29580   29780   29790   30010   30030   30380   30400   30430   31160   31170   31180   3123    31280
         31300   31670   31690   32040   32060   32080   32100   32250   32270   32590   32610   32640   32700   32720   32990
         33000   33070   33090   33130   33150   33160   33180   33190   33210   33340   33360   33410   33430   33830   33840
         34180   34190   34210   34230   34260   34280   34350   34370   34440   34590   34670   34690   34910   34920   35140
         35160   35170   35210   35410   35420   35550   35570   35580   35620   35630   35650   35660   35700   36010   36020
         36180   36190   36200   36220   36230   36250   36280   36300   36470   36490   36630   36650   36750   36770   36780
         36800   37160   37170
```

```
            20460#  2047#   20510#  20540#  20770#  20990#  21260#  2128#   21370#  21820#  22030#  22270#  22590#  22770#  23060#
            2308#   23180#  23600#  2361#   23640#  23850#  23930#  2395#   24340#  24920#  25530#  25570#  25950#  26360#  2638#
            26990#  27060#  27560#  2758#   28500#  28710#  28780#  28960#  2898#   30010#  30380#  3040#   31280#  31670#  32040#
            32080#  32250#  32590#  3261#   32700#  33070#  33160#  33190#  33340#  33410#  34210#  34350#  34670#  36200#  36230#
            36630#  36750#  36780#
$COMPA      10940#  11690#  12150#  12310#  12470#  12640#  12830#  13030#  13190#  13350#  13520#  13710#  14010#  14170#  14340#
            14530#  14770#  14930#  15100#  15290#  15600#  15910#  15980#  16050#  16250#  16630#  16930#  17350#  17530#  17750#
            17840#  17870#  18260#  18620#  18820#  19000#  19370#  19530#  19710#  19820#  20070#  20250#  20470#  20510#  20540#
            20770#  20990#  21260#  21370#  21820#  22030#  22270#  22590#  22770#  23060#  23180#  23360#  23610#  23640#  23850#
            23930#  24340#  24920#  25530#  25570#  25950#  26360#  26990#  27060#  27560#  27890#  28090#  28280#  28500#  28710#
            28780#  28960#  29360#  29570#  29780#  30010#  30380#  31280#  31670#  32040#  32080#  32250#  32590#  32700#  32990#
            33070#  33160#  33190#  33340#  33410#  34210#  34350#  34670#  35140#  35350#  35630#  36200#  36230#  36630#  36750#
            36780#
$COUNT      11090#  17230#  17670#  18320#  19260#  21610#  21710#  22530#  25220#  25460#  25900#  26780#  26900#  27810#  28010#
            28200#  29290#  29490#  29700#  30510#  32910#  34520#  36310#
$DO         10940#  11690#  12150#  12170#  12310#  12470#  12640#  12830#  13030#  13050#  13190#  13350#  13520#  13710#  14010#
            14170#  14340#  14530#  14770#  14930#  15100#  15290#  15600#  15910#  15980#  16050#  16250#  16630#  16930#  16950#
            17530#  17840#  17870#  18260#  18200#  18820#  19000#  19020#  19530#  19710#  19730#  19820#  20070#  20250#  20470#
            20510#  20540#  20770#  20990#  21260#  21200#  21370#  21820#  22030#  22270#  22590#  22770#  23060#  23080#  23180#
            23610#  23640#  23850#  23930#  23950#  24340#  24920#  25530#  25570#  25950#  26360#  26380#  26990#  27060#  27560#
            27580#  28500#  28710#  28780#  28960#  28980#  30010#  30380#  30400#  31280#  31670#  32040#  32080#  32250#  32590#
            32610#  32700#  33070#  33160#  33190#  33340#  33410#  34210#  34350#  34670#  36200#  36230#  36630#  36750#  36780#
$ELSE       15950#  16020#  16090#  17610#  17970#  20120#  23700#  23890#  23990#  25630#  27130#  33130#  34260#  36280#  36470#
$ERRMS      10930#  10940#  10950#  11010#  11020#  11040#  11060#  11090#  11130#  11160#  11180#  11210#  11230#  11240#  11260#
            11270#  11290#  11300#  11320#  11330#  11350#  11480#  11510#  11520#  11530#  11540#  11550#  11570#  11610#  11640#
            11690#  11700#  11750#  11780#  11800#  11810#  11830#  11840#  11880#  11890#  11900#  11910#  11920#  12150#  12170#
            12180#  12210#  12240#  12280#  12310#  12320#  12370#  12420#  12440#  12470#  12480#  12530#  12580#  12610#  12640#
            12650#  12700#  12750#  12780#  12830#  12840#  12890#  13030#  13050#  13060#  13090#  13120#  13160#  13190#  13200#
            13250#  13300#  13320#  13350#  13360#  13410#  13460#  13490#  13520#  13530#  13580#  13630#  13660#  13710#  13720#
            13770#  13980#  14010#  14020#  14070#  14120#  14140#  14170#  14180#  14230#  14280#  14310#  14340#  14350#  14400#
            14450#  14480#  14530#  14540#  14590#  14740#  14770#  14780#  14830#  14880#  14900#  14930#  14940#  14990#  15040#
            15070#  15100#  15110#  15160#  15210#  15240#  15290#  15300#  15350#  15580#  15600#  15610#  15710#  15910#  15920#
            15950#  15960#  15980#  15990#  16020#  16030#  16050#  16060#  16090#  16100#  16120#  16150#  16170#  16190#  16220#
            16250#  16260#  16300#  16420#  16600#  16630#  16640#  16740#  16930#  16950#  16960#  16990#  17020#  17050#  17100#
            17120#  17200#  17250#  17350#  17400#  17480#  17530#  17540#  17570#  17610#  17620#  17670#  17750#  17800#  17820#
            17840#  17850#  17870#  17880#  17930#  17950#  17970#  17980#  18000#  18020#  18040#  18070#  18080#  18260#  18280#
            18290#  18320#  18350#  18390#  18420#  18[illegible]  18520#  18620#  18660#  18700#  18750#  18820#  18830#  18880#  19000#
            19020#  19030#  19060#  19090#  19130#  19150#  19210#  19260#  19340#  19370#  19420#  19500#  19530#  19540#  19590#
            19710#  19730#  19740#  19770#  19800#  19820#  19830#  19850#  19880#  19930#  19950#  19970#  20030#  20070#  20080#
            20100#  20120#  20130#  20150#  20170#  20190#  20210#  20220#  20250#  20260#  20310#  20350#  20380#  20460#  20470#
            20480#  20510#  20520#  20540#  20550#  20600#  20650#  20680#  20700#  20720#  20730#  20770#  20780#  20820#  20860#
            20890#  20960#  20990#  21000#  21040#  21080#  21260#  21280#  21290#  21320#  21350#  21370#  21380#  21400#  21430#
            21460#  21510#  21580#  21610#  21680#  21710#  21780#  21820#  21830#  21870#  21930#  21960#  22010#  22030#  22040#
            22090#  22160#  22190#  22230#  22270#  22280#  22350#  22380#  22410#  22450#  22500#  22530#  22590#  22600#  22640#
            22710#  22740#  22770#  22780#  22820#  22870#  23060#  23080#  23090#  23120#  23150#  23180#  23190#  23210#  23240#
            23270#  23290#  23310#  23330#  23360#  23390#  23400#  23410#  23430#  23440#  23470#  23500#  23530#  23560#  23580#
            23600#  23610#  23620#  23640#  23650#  23680#  23700#  23710#  23740#  23760#  23780#  23810#  23820#  23850#  23860#
            23890#  23900#  23930#  23940#  23950#  23960#  23990#  24000#  24060#  24080#  24100#  24130#  24150#  24180#  24190#
            24210#  24220#  24260#  24280#  24320#  24340#  24350#  24400#  24920#  24930#  24950#  24980#  25030#  25070#  25100#
            25130#  25140#  25150#  25160#  25170#  25190#  25220#  25320#  25420#  25460#  25530#  25540#  25570#  25580#  25630#
            25640#  25700#  25720#  25760#  25790#  25820#  25900#  25950#  25960#  26010#  26070#  26110#  26120#  26130#  26140#
            26150#  26360#  26380#  26390#  26430#  26480#  26490#  26500#  26510#  26520#  26550#  26570#  26600#  26630#  26730#
            26780#  26870#  26900#  26960#  26990#  27000#  27040#  27060#  27070#  27130#  27140#  27190#  27210#  27270#  27320#
            27350#  27390#  27400#  27410#  27420#  27430#  27560#  27580#  27590#  27620#  27650#  27680#  27710#  27810#  27850#
            27890#  27920#  27930#  27940#  27960#  28010#  28090#  28110#  28130#  28170#  28200#  28280#  28300#  28320#  28350#
            28430#  28440#  28460#  28500#  28510#  28550#  28590#  28620#  28640#  28650#  28690#  28710#  28720#  28760#  28780#
```

	28790	28830	28960	28980	28990	29030	29060	29090	29200	29290	29330	29360	29390	29400	29410
	29430	29490	29570	29600	29630	29670	29700	29780	29830	29860	29900	29940	29950	29970	30010
	30020	30070	30100	30130	30140	30190	30380	30400	30410	30440	30470	30510	30740	30770	30790
	30830	30850	30890	30920	30950	30980	31040	31080	31110	31180	31200	31210	31250	31280	31290
	31340	31370	31390	31570	31600	31610	31640	31670	31680	31710	31730	31920	31950	31960	31990
	32000	32040	32050	32080	32090	32120	32140	32160	32190	32210	32250	32260	32290	32340	32590
	32610	32620	32650	32680	32700	32710	32730	32760	32790	32820	32850	32880	32910	32990	33040
	33070	33080	33110	33130	33140	33160	33170	33190	33200	33220	33240	33260	33280	33340	33350
	33390	33410	33420	33460	34080	34100	34120	34210	34220	34240	34260	34270	34290	34310	34350
	34360	34390	34410	34440	34460	34520	34570	34590	34600	34670	34680	34700	34720	34740	34830
	35060	35080	35090	35140	35170	35180	35190	35210	35220	35240	35260	35270	35290	35310	35340
	35530	35550	35580	35590	35600	35620	35630	35660	35670	35680	35700	35710	35720	35740	35750
	35800	35960	36140	36160	36200	36210	36230	36240	36260	36280	36290	36310	36350	36370	36390
	36410	36430	36450	36470	36480	36500	36520	36540	36560	36580	36600	36610	36630	36640	36660
	36680	36700	36750	36760	36780	36790	36820	36840	36860	36880	36910	36930	36950	37090	37530
$EXIF2	18040	34440													
$EXIF3	18040	34440													
$GENBR	10950	11010	11700	11840	12160	12180	12320	12480	12650	12840	13040	13060	13200	13360	13530
	13720	14020	14180	14350	14540	14780	14940	15110	15300	15610	15920	15950	15990	16020	16060
	16090	16260	16640	16940	16960	17350	17540	17610	17750	17850	17880	17970	18050	18070	18270
	18290	18620	18830	19010	19030	19370	19540	19720	19740	19830	20080	20120	20200	20220	20260
	20480	20520	20550	20720	20780	21000	21270	21290	21380	21830	22040	22280	22600	22780	23070
	23090	23190	23370	23420	23620	23650	23700	23810	23860	23890	23940	23960	23990	24180	24350
	24930	25540	25580	25630	25960	26370	26390	27000	27070	27130	27570	27590	27900	27950	28090
	28280	28510	28640	28720	28790	28970	28990	29370	29420	29570	29780	30020	30130	30390	30410
	31190	31210	31290	31680	32050	32090	32260	32600	32620	32710	32990	33080	33130	33170	33200
	33350	33420	34220	34260	34360	34450	34470	34590	34680	34700	34750	35150	35200	35260	35310
	35560	35610	35640	35690	35710	35740	36210	36240	36280	36470	36610	36640	36760	36790	36950
$GENTA	10930	10940	11010	1102	11590	11690	11750	12150	12170	1219	12240	12310	12370	12470	12530
	12640	12700	12830	12890	13030	13050	1307	13120	13190	13250	13350	13410	13520	13580	13710
	13770	14010	14070	14170	14230	14340	14400	14530	14590	14770	14830	14930	14990	15100	15160
	15290	15350	15600	15710	15910	15950	1596	15980	16020	1603	16050	16090	1610	16150	16170
	16190	16250	16420	16630	16740	16930	16950	1697	17020	17070	17400	17530	17610	1762	17800
	17820	17840	17870	17950	17970	1798	18020	18070	1808	18260	18280	1830	18350	18700	18820
	18880	19000	19020	1904	19090	19420	19530	19590	19710	19730	1975	19800	19820	19880	20050
	20070	20120	2013	20170	20190	2023	20250	20380	20460	20470	20510	20540	20680	20700	20720
	2073	20770	20890	20990	21080	21260	21280	2130	21350	21370	21430	21820	21960	22030	22190
	22270	22410	22590	22740	22770	22870	23060	23080	2310	23150	23180	23240	23360	2338	23390
	2340	23600	23610	23640	23700	2371	23780	23810	2382	23850	23890	2390	23930	23950	23990
	2400	24080	24100	24180	2419	24340	24400	24920	24980	25530	25570	25630	2564	25700	25720
	25950	26010	26360	26380	2640	26430	26990	27060	27130	2714	27190	27210	27560	27580	2760
	27650	27890	2791	27920	2793	28130	28320	28500	28620	28640	2865	28710	28760	28780	28830
	28960	28980	2900	29060	29360	2938	29390	2940	29630	29860	30010	30100	30130	3014	30380
	30400	3042	30470	31160	31180	3122	31280	31340	31670	31730	32040	32080	32160	32210	32250
	32340	32590	32610	3263	32680	32700	32760	33040	33070	33130	3314	33160	33190	33240	33260
	33280	33340	33390	33410	33460	34180	34210	34260	3427	34310	34350	34410	34440	34590	3460
	34670	34720	34830	3485	35140	3516	35170	3518	35260	3527	35340	3535	35550	3557	35580
	3559	35630	3565	35660	3567	35710	3572	35740	3575	35800	3581	36180	36200	36230	36280
	3629	36370	36470	3648	36580	36630	36750	36780	36840	36860	36880	37090	3710	37530	3754
$IF	11690	12150	12310	12470	12640	12830	13030	13190	13350	13520	13710	14010	14170	14340	14530
	14770	14930	15100	15290	15600	15910	15980	16050	16250	16630	16930	17530	17840	17870	18260
	18820	19000	19530	19710	19820	20070	20250	20510	20540	20770	20990	21260	21370	21820	22030
	22270	22590	22770	23060	23180	23640	23850	23930	24340	24920	25530	25570	25950	26360	26990
	27060	27560	28500	28710	28780	28960	30010	30380	31280	31670	32040	32080	32250	32590	32700
	33070	33160	33190	33340	33410	34210	34350	34670	36200	36230	36630	36750	36780		
$IFCOD	10940	11690	11830	12170	12310	12470	12640	12830	13050	13190	13350	13520	13710	14010	14170

	14340	14530	14770	14930	15100	15290	15600	15910	15980	16050	16250	16630	16950	17530	17840
	17870	18200	18820	19020	19530	19750	19820	20070	20190	20210	20250	20470	20510	20540	20770
	20990	21200	21370	21820	22030	22270	22590	22770	23080	23180	23610	23640	23850	23930	23950
	24340	24920	25330	25570	25950	26380	26990	27060	27580	28500	28710	28780	28980	30010	30400
	31180	31200	31280	31670	32040	32080	32250	32610	32700	33070	33160	33190	33340	33410	34210
	34350	34440	34670	36200	36230	36600	36630	36750	36780						
$IFCON	17350	17750	18620	19570	28090	28280	29570	29780	32990						
$IFOPR	10940	1095	11690	1170	11830	1184	12170	1218	12310	1232	12470	1248	12640	1265	12830
	1284	13050	1306	13190	1320	13350	1336	13520	1353	13710	1372	14010	1402	14170	1418
	14340	1435	14530	1454	14770	1478	14930	1494	15100	1511	15290	1530	15600	1561	15910
	1592	15980	1595	16050	1606	16250	1626	16630	1664	16950	1696	17350	17530	1754	17750
	17840	1785	17870	1788	18200	1829	18620	18820	1883	19020	1903	19370	19530	1954	19730
	1974	19820	1983	20070	2008	20210	2022	20250	2026	20470	2048	20510	2052	20540	2055
	20770	2078	20990	2100	21200	2129	21370	2138	21820	2183	22030	2204	22270	2228	22590
	2260	22770	2278	23080	2309	23180	2319	23610	2362	23640	2365	23850	2386	23930	2394
	23950	2396	24340	2455	24920	2493	25330	2554	25570	2558	25950	2596	26380	2639	26990
	2700	27060	2707	27580	2759	28090	28280	28500	2851	28710	2872	28780	2879	28980	2899
	29570	29780	30010	3002	30400	3041	31200	3121	31280	3129	31670	3168	32040	3205	32080
	3209	32250	3226	32610	3262	32700	3271	32990	33070	3308	33160	3317	33190	3320	33340
	3335	33410	3342	34210	3422	34350	3436	34670	3468	36200	3621	36230	3624	36600	3661
	36630	3664	36750	3676	36780	3679									
$LET	11040	11060	11130	11160	11180	11210	11230	11260	11290	11320	11350	11480	11510	11520	11530
	11540	11350	11570	11610	11640	11780	11800	11880	11890	11910	11920	12210	12280	12420	12440
	12580	12610	12750	12780	13090	13160	13300	13320	13460	13490	13630	13660	13980	14120	14140
	14280	14310	14450	14480	14740	14880	14900	15040	15070	15210	15240	15580	16120	16220	16300
	16600	16990	17030	17100	17120	17200	17480	17570	17930	18000	18320	18390	18420	18460	18650
	18750	19060	19130	19150	19210	19340	19500	19770	19850	19930	19950	19970	20030	20100	20150
	20310	20350	20600	20650	20820	20860	20960	21040	21320	21400	21460	21510	21580	21680	21780
	21870	21930	22010	22090	22160	22230	22330	22380	22450	22500	22640	22710	22820	23120	23210
	23270	23290	23310	23330	23440	23470	23500	23530	23560	23580	23680	23740	23760	24060	24130
	24150	24210	24260	24280	24320	24950	25030	25070	25100	25130	25140	25150	25160	25170	25190
	25320	25420	25760	25790	25820	26070	26110	26120	26140	26150	26480	26490	26500	26510	26520
	26550	26570	26600	26630	26730	26870	26960	27040	27270	27320	27350	27390	27400	27420	27430
	27620	27680	27710	27850	28110	28170	28300	28350	28430	28460	28550	28590	28690	29030	29090
	29200	29330	29600	29670	29830	29900	29940	29970	30070	30190	30440	30740	30770	30790	30830
	30850	30890	30920	30950	30980	31040	31080	31110	31250	31370	31390	31570	31600	31640	31710
	31920	31950	31990	32120	32140	32190	32290	32650	32730	32790	32820	32850	32880	33110	33220
	34080	34100	34120	34240	34290	34390	34570	35060	35080	35220	35240	35290	35530	35960	36140
	36160	36260	36350	36390	36410	36430	36450	36500	36520	36540	36560	36660	36680	36700	36820
	36910	36930													
$LPCNT	23360	27890	29360	35140	35350	35630									
$MCHIG	10														
$MCLOW	10														
$OPABS	35290														
$OPADD	11240	11270	11300	11330	11780	11810	11900	12440	12610	12780	13320	13490	13660	14140	14310
	14480	14900	15070	15240	16300	18390	18660	19130	19340	19930	20150	20310	20600	20820	21040
	21460	21870	22090	22330	22640	22820	23330	23390	23740	23760	24220	24260	24320	25100	25320
	25420	25820	26130	26600	26630	26870	27040	27270	27320	27410	27710	27920	28110	28300	28440
	28460	28550	28690	29090	29200	29390	29950	29970	30190	31040	31080	31110	31250	31370	31390
	31570	31610	31710	31920	31960	32000	32190	32290	32820	32850	33110	33220	34570	35090	35170
	35220	35240	35580	35660	35980	36520	36540	36560	36930						
$OPAND	11240	11270	11300	11330	11780	11810	11900	12440	12610	12780	13320	13490	13660	14140	14310
	14480	14900	15070	15240	16300	18390	18660	19130	19340	19930	20150	20310	20600	20820	21040
	21460	21870	22090	22330	22640	22820	23330	23390	23740	23760	24220	24260	24320	25100	25320
	25420	25820	26130	26600	26630	26870	27040	27270	27320	27410	27710	27920	28110	28300	28440
	28460	28550	28690	29090	29200	29390	29950	29970	30190	31040	31080	31110	31250	31370	31390

```
                31570#  31610#  31710#  31920#  31960#  32000#  32190#  32290#  32620#  32850#  33110#  33220#  34570#  35090#  35170#
                35220#  35240#  35580#  35660#  35960#  36520#  36540#  36560#  36930#
$OPCD1          35290#
$OPCD2          11230#  1124    11260#  1127    11290#  1130    11320#  1133    11780#  11800#  1181    11890#  1190    12440#  12610#
                12780#  13320#  13490#  13660#  14140#  14310#  14480#  14900#  15070#  15240#  16300#  18390#  18660#  19130#  19340#
                19930#  20150#  20310#  20600#  20820#  21040#  21460#  21870#  22090#  22350#  22640#  22820#  23330#  23390#  23740#
                23760#  24210#  2422    24260#  24320#  25100#  25320#  25420#  25820#  26120#  2613    26600#  26630#  26870#  27040#
                27270#  27320#  27400#  2741    27710#  27920#  28110#  28300#  28430#  2844    28460#  28550#  28690#  29090#  29200#
                29390#  29940#  2995    29970#  30190#  31040#  31080#  31110#  31250#  31370#  31390#  31570#  31600#  3161    31710#
                31920#  31950#  3196    31990#  3200    32190#  32290#  32820#  32850#  33110#  33220#  34570#  35080#  3509    35170#
                35220#  35240#  35580#  35660#  35960#  36520#  36540#  36560#  36930#
$OPCOM          35290#
$OPDEF          10940#  10950#  11010#  11040#  11060#  11090#  11130#  11160#  11180#  11210#  11230#  11240#  11260#  11270#  11290#
                11300#  11320#  11330#  11350#  11480#  11510#  11520#  11530#  11540#  11550#  11570#  11610#  11640#  11690#  11700#
                11780#  11800#  11810#  11830#  11840#  11880#  11890#  11900#  11910#  11920#  12150#  12160#  12170#  12180#  12210#
                12280#  12310#  12320#  12420#  12440#  12470#  12480#  12580#  12610#  12640#  12650#  12750#  12780#  12830#  12840#
                13030#  13040#  13050#  13060#  13090#  13160#  13190#  13200#  13300#  13320#  13350#  13360#  13460#  13490#  13520#
                13530#  13630#  13660#  13710#  13720#  13980#  14010#  14020#  14120#  14140#  14170#  14180#  14280#  14310#  14340#
                14350#  14450#  14480#  14530#  14540#  14740#  14770#  14780#  14880#  14900#  14930#  14940#  15040#  15070#  15100#
                15110#  15210#  15340#  15290#  15300#  15580#  15600#  15610#  15910#  15920#  15950#  15980#  15990#  16020#  16050#
                16060#  16090#  16120#  16220#  16250#  16260#  16300#  16600#  16630#  16640#  16930#  16940#  16950#  16960#  16990#
                17050#  17100#  17120#  17200#  17250#  17260#  1727    1728    1729    1730    1731    17350#  17400#  17530#  17540#
                17570#  17610#  17670#  17680#  1769    1770    1771    1772    1773    17750#  17840#  17850#  17870#  17880#  17930#
                17970#  18000#  18040#  18050#  18070#  18260#  18270#  18280#  18290#  18320#  18390#  18420#  18460#  18520#  18530#
                1854    1855    1856    1857    1858    18620#  18660#  18750#  18820#  18830#  19000#  19010#  19020#  19030#  19060#
                19130#  19150#  19210#  19260#  19270#  1928    1929    1930    1931    1932    19340#  19370#  19500#  19530#  19540#
                19710#  19720#  19730#  19740#  19770#  19820#  19830#  19850#  19930#  19950#  19970#  20030#  20070#  20080#  20100#
                20120#  20150#  20190#  20200#  20210#  20220#  20250#  20260#  20310#  20350#  20470#  20480#  20510#  20520#  20540#
                20550#  20600#  20650#  20720#  20770#  20780#  20820#  20860#  20960#  20990#  21000#  21040#  21260#  21270#  21280#
                21290#  21320#  21370#  21380#  21400#  21460#  21510#  21580#  21610#  21620#  2163    2164    21680#  21710#  21720#
                2173    2174    21780#  21820#  21830#  21870#  21930#  2201#   22030#  22040#  22090#  22160#  22230#  22270#  22280#
                22330#  22380#  22450#  22500#  22530#  22540#  2255    2256    22590#  22600#  22640#  22710#  22770#  22780#  22820#
                23060#  23070#  23080#  23090#  23120#  23180#  23190#  23210#  23270#  23290#  23310#  23330#  23360#  23370#  23390#
                23410#  23420#  23440#  23470#  23500#  23530#  23560#  23580#  23610#  23620#  23640#  23650#  23680#  23700#  23740#
                23760#  23810#  23850#  23860#  23890#  23930#  23940#  23950#  23960#  23990#  24060#  24130#  24150#  24180#  24210#
                24220#  2423    2424    24260#  24280#  24320#  24340#  24350#  24920#  24930#  24950#  25030#  25070#  25100#  25130#
                25140#  25150#  25160#  25170#  25190#  25220#  25230#  2524    2525    2526    2527    2528    25320#  25420#  25460#
                25470#  2548    2549    25530#  25540#  25570#  25580#  25630#  25760#  25790#  25820#  25900#  25910#  2592    2593
                25950#  25960#  26070#  26110#  26120#  26130#  26140#  26150#  26360#  26370#  26380#  26390#  26480#  26490#  26500#
                26510#  26520#  26550#  26570#  26600#  26630#  26730#  26780#  26790#  2680    2681    2682    2683    2684    26870#
                26900#  26910#  2692    2693    26960#  26990#  27000#  27040#  27060#  27070#  27130#  27270#  27320#  27350#  27390#
                27400#  27410#  27420#  27430#  27560#  27570#  27580#  27590#  27620#  27680#  27710#  27810#  2782    2783    27850#
                27890#  27900#  27920#  27940#  27950#  28010#  28020#  2803    2804    2805    2806    2807    28090#  28110#  28170#
                28200#  28210#  2822    2823    2824    2825    2826    28280#  28300#  28350#  28430#  28440#  28460#  28500#  28510#
                28550#  28590#  28640#  28690#  28710#  28720#  28780#  28790#  28960#  28970#  28980#  28990#  29030#  29090#  29200#
                29290#  2930    2931    29330#  29360#  29370#  29390#  29410#  29420#  29490#  29500#  2951    2952    2953    2954
                2955    29570#  29600#  29670#  29700#  29710#  2972    2973    2974    2975    2976    29780#  29830#  29900#  29940#
                29950#  29970#  30010#  30020#  30070#  30130#  30190#  30380#  30390#  30400#  30410#  30440#  30510#  3052    3053
                30740#  30770#  30790#  30830#  30850#  30890#  30920#  30950#  30980#  31040#  31080#  31110#  31180#  31190#  31200#
                31210#  31250#  31280#  31290#  31370#  31390#  31570#  31600#  31610#  31640#  31670#  31680#  31710#  31920#  31950#
                31960#  31990#  32000#  32040#  32050#  32080#  32090#  32120#  32140#  32190#  32250#  32260#  32290#  32590#  32600#
                32610#  32620#  32650#  32700#  32710#  32730#  32790#  32820#  32850#  32880#  32910#  32920#  3293    3294    3295
                3296    3297    32990#  33070#  33080#  33110#  33130#  33160#  33170#  33190#  33200#  33220#  33340#  33350#  33410#
                33420#  34080#  34100#  34120#  34210#  34220#  34240#  34260#  34290#  34350#  34360#  34390#  34440#  34450#  34460#
                34470#  34520#  34530#  3454    3455    34570#  34590#  34670#  34680#  34700#  34740#  34750#  34830#  3484    3486
                35060#  35080#  35090#  3510    3511    35140#  35150#  35170#  35190#  35200#  35220#  35240#  35260#  35290#  35310#
```

	35340	3536	35534	35554	35564	35584	35604	35614	35634	35644	35664	35684	35694	35714	35744
	35804	3582	35964	36144	36164	36204	36214	36234	36244	36264	36284	36314	36354	36394	36414
	36434	36454	36474	36504	36524	36544	36564	36604	36614	36634	36644	36664	36684	36704	36754
	36764	36784	36794	36824	36914	36934	36954	37094	3711	37534	3755				
$OPEQU	11244	11274	11304	11334	11784	11814	11904	12444	12614	12784	13324	13494	13664	14144	14314
	14484	14904	15074	15244	16304	18394	18664	19134	19344	19934	20154	20314	20604	20824	21044
	21464	21874	22094	22334	22644	22824	23334	23394	23744	23764	24224	24264	24324	25104	25324
	25424	25824	26134	26604	26634	26874	27044	27274	27324	27414	27714	27924	28114	28304	28444
	28464	28554	28694	29094	29204	29394	29954	29974	30194	31044	31084	31114	31254	31374	31394
	31574	31614	31714	31924	31964	32004	32194	32294	32824	32854	33114	33224	34574	35094	35174
	35224	35244	35584	35664	35964	36524	36544	36564	36934						
$OPNAN	11244	11274	11304	11334	11784	11814	11904	12444	12614	12784	13324	13494	13664	14144	14314
	14484	14904	15074	15244	16304	18394	18664	19134	19344	19934	20154	20314	20604	20824	21044
	21464	21874	22094	22334	22644	22824	23334	23394	23744	23764	24224	24264	24324	25104	25324
	25424	25824	26134	26604	26634	26874	27044	27274	27324	27414	27714	27924	28114	28304	28444
	28464	28554	28694	29094	29204	29394	29954	29974	30194	31044	31084	31114	31254	31374	31394
	31574	31614	31714	31924	31964	32004	32194	32294	32824	32854	33114	33224	34574	35094	35174
	35224	35244	35584	35664	35964	36524	36544	36564	36934						
$OPNEG	35294														
$OPNOR	11244	11274	11304	11334	11784	11814	11904	12444	12614	12784	13324	13494	13664	14144	14314
	14484	14904	15074	15244	16304	18394	18664	19134	19344	19934	20154	20314	20604	20824	21044
	21464	21874	22094	22334	22644	22824	23334	23394	23744	23764	24224	24264	24324	25104	25324
	25424	25824	26134	26604	26634	26874	27044	27274	27324	27414	27714	27924	28114	28304	28444
	28464	28554	28694	29094	29204	29394	29954	29974	30194	31044	31084	31114	31254	31374	31394
	31574	31614	31714	31924	31964	32004	32194	32294	32824	32854	33114	33224	34574	35094	35174
	35224	35244	35584	35664	35964	36524	36544	36564	36934						
$OPNOT	11244	11274	11304	11334	11784	11814	11904	12444	12614	12784	13324	13494	13664	14144	14314
	14484	14904	15074	15244	16304	18394	18664	19134	19344	19934	20154	20314	20604	20824	21044
	21464	21874	22094	22334	22644	22824	23334	23394	23744	23764	24224	24264	24324	25104	25324
	25424	25824	26134	26604	26634	26874	27044	27274	27324	27414	27714	27924	28114	28304	28444
	28464	28554	28694	29094	29204	29394	29954	29974	30194	31044	31084	31114	31254	31374	31394
	31574	31614	31714	31924	31964	32004	32194	32294	32824	32854	33114	33224	34574	35094	35174
	35224	35244	35584	35664	35964	36524	36544	36564	36934						
$OPOR	11244	11274	11304	11334	11784	11814	11904	12444	12614	12784	13324	13494	13664	14144	14314
	14484	14904	15074	15244	16304	18394	18664	19134	19344	19934	20154	20314	20604	20824	21044
	21464	21874	22094	22334	22644	22824	23334	23394	23744	23764	24224	24264	24324	25104	25324
	25424	25824	26134	26604	26634	26874	27044	27274	27324	27414	27714	27924	28114	28304	28444
	28464	28554	28694	29094	29204	29394	29954	29974	30194	31044	31084	31114	31254	31374	31394
	31574	31614	31714	31924	31964	32004	32194	32294	32824	32854	33114	33224	34574	35094	35174
	35224	35244	35584	35664	35964	36524	36544	36564	36934						
$OPROT	11244	11274	11304	11334	11784	11814	11904	12444	12614	12784	13324	13494	13664	14144	14314
	14484	14904	15074	15244	16304	18394	18664	19134	19344	19934	20154	20314	20604	20824	21044
	21464	21874	22094	22334	22644	22824	23334	23394	23744	23764	24224	24264	24324	25104	25324
	25424	25824	26134	26604	26634	26874	27044	27274	27324	27414	27714	27924	28114	28304	28444
	28464	28554	28694	29094	29204	29394	29954	29974	30194	31044	31084	31114	31254	31374	31394
	31574	31614	31714	31924	31964	32004	32194	32294	32824	32854	33114	33224	34574	35094	35174
	35224	35244	35584	35664	35964	36524	36544	36564	36934						
$OPRO	11044	11064	11134	11164	11184	11214	11234	11264	11294	11324	11354	11484	11514	11524	11534
	11544	11554	11574	11614	11644	11784	11804	11884	11894	11914	11924	12214	12284	12424	12444
	12584	12614	12754	12784	13094	13164	13304	13324	13464	13494	13634	13664	13984	14124	14144
	14284	14314	14454	14484	14744	14884	14904	15044	15074	15214	15244	15584	16124	16224	16304
	16604	16994	17054	17104	17124	17204	17484	17574	17934	18004	18324	18394	18424	18464	18664
	18754	19064	19134	19154	19214	19344	19504	19774	19854	19934	19954	19974	20034	20104	20154
	20314	20354	20604	20654	20824	20864	20964	21044	21324	21404	21464	21514	21584	21684	21784
	21874	21934	22014	22094	22164	22234	22334	22384	22454	22504	22644	22714	22824	23124	23214
	23274	23294	23314	23334	23364	23444	23474	23504	23534	23564	23584	23684	23744	23764	24064

	24130	24150	24210	24260	24280	24320	24950	25030	25070	25100	25130	25140	25150	25160	25170
	25190	25320	25420	25760	25790	25820	26070	26110	26120	26140	26150	26480	26490	26500	26510
	26520	26550	26570	26600	26630	26730	26870	26960	27040	27270	27320	27350	27390	27400	27420
	27430	27620	27680	27710	27850	27890	28110	28170	28300	28350	28430	28460	28550	28590	28690
	29030	29090	29200	29330	29360	29600	29670	29830	29900	29940	29970	30070	30190	30440	30740
	30770	30790	30830	30850	30890	30920	30950	30980	31040	31080	31110	31250	31370	31390	31570
	31600	31640	31710	31920	31950	31990	32120	32140	32190	32290	32650	32730	32790	32820	32850
	32880	33110	33220	34080	34100	34120	34240	34290	34390	34570	35060	35080	35140	35220	35240
	35290	35530	35550	35630	35960	36140	36160	36260	36350	36390	36410	36430	36450	36500	36520
	36540	36560	36660	36680	36700	36820	36910	36930							
$OPR1	11040	11060	11130	11160	11180	11210	11230	11260	11290	11320	11350	11480	11510	11520	11530
	11540	11550	11570	11610	11640	11780	11800	11880	11890	11910	11920	12210	12280	12420	12440
	12580	12610	12750	12780	13090	13160	13300	13320	13460	13490	13630	13660	13980	14120	14140
	14280	14310	14450	14480	14740	14880	14900	15040	15070	15210	15240	15580	16120	16220	16300
	16600	16990	17050	17100	17120	17200	17480	17570	17930	18000	18320	18390	18420	18460	18660
	18750	19060	19130	19150	19210	19340	19500	19770	19850	19930	19950	19970	20030	20100	20150
	20310	20350	20600	20650	20820	20860	20960	21040	21320	21400	21460	21510	21580	21680	21780
	21870	21930	22010	22090	22160	22230	22330	22380	22450	22500	22640	22710	22820	23120	23210
	23270	23290	23310	23330	23440	23470	23500	23530	23560	23580	23680	23740	23760	24060	24130
	24150	24210	24260	24280	24320	24950	25030	25070	25100	25130	25140	25150	25160	25170	25190
	25320	25420	25760	25790	25820	26070	26110	26120	26140	26150	26480	26490	26500	26510	26520
	26550	26570	26600	26630	26730	26870	26960	27040	27270	27320	27350	27390	27400	27420	27430
	27620	27680	27710	27850	28110	28170	28300	28350	28430	28460	28550	28590	28690	29030	29090
	29200	29330	29600	29670	29830	29900	29940	29970	30070	30190	30440	30740	30770	30790	30830
	30850	30890	30920	30950	30980	31040	31080	31110	31250	31370	31390	31570	31600	31640	31710
	31920	31950	31990	32120	32140	32190	32290	32650	32730	32790	32820	32850	32880	33110	33220
	34080	34100	34120	34240	34290	34390	34570	35060	35080	35220	35240	35290	35530	35960	36140
	36160	36260	36350	36390	36410	36430	36450	36500	36520	36540	36560	36660	36680	36700	36820
	36910	36930													
$OPR2	11040	11060	11130	11160	11180	11210	11230	11260	11290	11320	11350	11480	11510	11520	11530
	11540	11550	11570	11610	11640	11780	11800	11880	11890	11910	11920	12210	12280	12420	12440
	12580	12610	12750	12780	13090	13160	13300	13320	13460	13490	13630	13660	13980	14120	14140
	14280	14310	14450	14480	14740	14880	14900	15040	15070	15210	15240	15580	16120	16220	16300
	16600	16990	17050	17100	17120	17200	17480	17570	17930	18000	18320	18390	18420	18460	18660
	18750	19060	19130	19150	19210	19340	19500	19770	19850	19930	19950	19970	20030	20100	20150
	20310	20350	20600	20650	20820	20860	20960	21040	21320	21400	21460	21510	21580	21680	21780
	21870	21930	22010	22090	22160	22230	22330	22380	22450	22500	22640	22710	22820	23120	23210
	23270	23290	23310	23330	23390	23440	23470	23500	23530	23560	23580	23680	23740	23760	24060
	24130	24150	24210	24260	24280	24320	24950	25030	25070	25100	25130	25140	25150	25160	25170
	25190	25320	25420	25760	25790	25820	26070	26110	26120	26140	26150	26480	26490	26500	26510
	26520	26550	26570	26600	26630	26730	26870	26960	27040	27270	27320	27350	27390	27400	27420
	27430	27620	27680	27710	27850	27920	28110	28170	28300	28350	28430	28460	28550	28590	28690
	29030	29090	29200	29330	29390	29600	29670	29830	29900	29940	29970	30070	30190	30440	30740
	30770	30790	30830	30850	30890	30920	30950	30980	31040	31080	31110	31250	31370	31390	31570
	31600	31640	31710	31920	31950	31990	32120	32140	32190	32290	32650	32730	32790	32820	32850
	32880	33110	33220	34080	34100	34120	34240	34290	34390	34570	35060	35080	35170	35220	35240
	35290	35530	35580	35660	35960	36140	36160	36260	36350	36390	36410	36430	36450	36500	36520
	36540	36560	36660	36680	36700	36820	36910	36930							
$OPSHF	11240	11270	11300	11330	11780	11810	11900	12440	12610	12780	13320	13490	13660	14140	14310
	14480	14900	15070	15240	16300	18390	18660	19130	19340	19930	20150	20310	20600	20820	21040
	21460	21870	22090	22330	22640	22820	23330	23390	23740	23760	24220	24260	24320	25100	25320
	25420	25820	26130	26600	26630	26870	27040	27270	27320	27410	27710	27920	28110	28300	28440
	28460	28550	28690	29090	29200	29390	29950	29970	30190	31040	31080	31110	31250	31370	31390
	31570	31610	31710	31920	31960	32000	32190	32290	32820	32850	33110	33220	34570	35090	35170
	35220	35240	35580	35660	35960	36520	36540	36560	36930						
$OPSUB	11240	11270	11300	11330	11780	11810	11900	12440	12610	12780	13320	13490	13660	14140	14310

```
            14480  14900  15070  15240  16300  16390  16660  19130  19340  19930  20150  20310  20600  20820  21040
            21460  21870  22090  22330  22640  22820  23330  23590  23740  23760  24220  24260  24320  25100  25320
            25420  25820  26130  26600  26630  26870  27040  27270  27320  27410  27710  27920  28110  28300  28440
            28460  28550  28690  29090  29200  29390  29950  29970  30190  31040  31080  31110  31250  31370  31390
            31570  31610  31710  31920  31960  32000  32190  32290  32820  32850  33110  33220  34570  35090  35170
            35220  35240  35580  35660  35960  36520  36540  36560  36930
$OPSMB      35290
$OPXOR      11240  11270  11300  11330  11780  11810  11900  12440  12610  12780  13320  13490  13660  14140  14310
            14480  14900  15070  15240  16300  16390  16660  19130  19340  19930  20150  20310  20600  20820  21040
            21460  21870  22090  22330  22640  22820  23330  23590  23740  23760  24220  24260  24320  25100  25320
            25420  25820  26130  26600  26630  26870  27040  27270  27320  27410  27710  27920  28110  28300  28440
            28460  28550  28690  29090  29200  29390  29950  29970  30190  31040  31080  31110  31250  31370  31390
            31570  31610  31710  31920  31960  32000  32190  32290  32820  32850  33110  33220  34570  35090  35170
            35220  35240  35580  35660  35960  36520  36540  36560  36930
$OR         10940  11690  12150  12170  12310  12470  12640  12830  13030  13050  13190  13350  13520  13710  14010
            14170  14340  14530  14770  14930  15100  15290  15600  15910  15980  16050  16250  16630  16930  16950
            17530  17840  17870  18260  18280  18420  19000  19020  19530  19710  19730  19820  20070  20250  20470
            20510  20540  20770  20990  21260  21280  21370  21820  22030  22270  22590  22770  23060  23080  23180
            23610  23640  23850  23930  23950  24340  24920  25530  25570  25950  26360  26380  26990  27060  27560
            27580  28500  28710  28780  28960  28980  30010  30380  30400  31280  31670  32040  32080  32250  32590
            32610  32700  33070  33160  33190  33340  33410  34210  34350  34670  36200  36230  36630  36750  36780
$PUT        11090  17250  1726   17670  1768   18520  1853   19260  1927   21610  2162   21710  2172   22530  2254
            25220  2523   25460  2547   25900  2591   26780  2679   26900  2691   27810  28010  2802   28200  2821
            29290  29490  2950   29700  2971   30510  32910  3292   34520  3453   36310
$SUBON      11010  11020  11750  11830  12240  12370  12530  12700  12890  13120  13250  13410  13580  13770  14070
            14230  14400  14590  14830  14990  15160  15350  15710  15960  16030  16100  16150  16170  16190  16420
            16740  17020  17400  17620  17800  17820  17950  17980  18020  18070  18080  18350  18700  18880  19090
            19420  19590  19800  19880  20130  20170  20190  20380  20680  20700  20720  20730  20890  21080  21350
            21430  21960  22190  22410  22740  22870  23150  23240  23400  23430  23710  23780  23810  23820  23900
            24000  24080  24100  24180  24190  24400  24980  25640  25700  25720  26010  26430  27140  27190  27210
            27650  27930  27960  28130  28320  28620  28640  28650  28760  28830  29060  29400  29430  29630  29860
            30100  30130  30140  30470  31180  31340  31730  32160  32210  32340  32680  32760  33040  33140  33240
            33260  33280  33390  33460  34270  34310  34410  34590  34600  34720  34830  35180  35210  35260  35270
            35340  35590  35620  35670  35700  35710  35720  35740  35750  35800  36290  36370  36480  36580  36600
            36840  36860  36880  37090  37530
$THEN       10940  11690  12150  12170  12310  12470  12640  12830  13030  13050  13190  13350  13520  13710  14010
            14170  14340  14530  14770  14930  15100  15290  15600  15910  15980  16050  16250  16630  16930  16950
            17530  17840  17870  18260  18280  18820  19000  19020  19530  19710  19730  19820  20070  20250  20470
            20510  20540  20770  20990  21260  21280  21370  21820  22030  22270  22590  22770  23060  23080  23180
            23610  23640  23850  23930  23950  24340  24920  25530  25570  25950  26360  26380  26990  27060  27560
            27580  28500  28710  28780  28960  28980  30010  30380  30400  31280  31670  32040  32080  32250  32590
            32610  32700  33070  33160  33190  33340  33410  34210  34350  34670  36200  36230  36630  36750  36780
$UNTL2      11830  20190  31180  36600
$UNTL3      11830  20190  31180  36600
$WHILE      10930  20460  23600
$$CMRE       9090
$$CMTM       9090
$$ESCA         10  7490
$$GEN       10930  11020  11590  11750  12190  12240  12370  12530  12700  12890  13070  13120  13250  13410  13580
            13770  14070  14230  14400  14590  14830  14990  15160  15350  15710  15960  16030  16100  16150  16170
            16190  16420  16740  16970  17020  17070  17400  17620  17800  17820  17950  17980  18020  18080  18300
            18350  18700  18880  19040  19090  19420  19590  19750  19800  19880  20050  20130  20170  20230  20380
            20460  20680  20700  20730  20890  21080  21300  21350  21430  21960  22190  22410  22740  22870  23100
            23150  23240  23380  23400  23600  23710  23780  23820  23900  24000  24080  24100  24190  24400  24980
            25640  25700  25720  26010  26400  26430  27140  27190  27210  27600  27650  27910  27930  28130  28320
            28620  28650  28760  28830  29000  29060  29380  29400  29630  29860  30100  30140  30420  30470  31160
```

```
            31220  31340  31730  32160  32210  32340  32630  32680  32760  33040  33140  33240  33260  33280  33390
            33460  33830  34180  34270  34310  34410  34600  34720  34830  34850  34910  35160  35180  35270  35340
            35350  35410  35570  35590  35650  35670  35720  35750  35800  35810  36010  36180  36290  36370  36480
            36580  36840  36860  36880  37090  37100  37160  37530  37540
$$GETS      11010  11020  11750  11830  12240  12370  12530  12700  12890  13120  13250  13410  13580  13770  14070
            14230  14400  14590  14830  14990  15160  15350  15710  15950  15960  16020  16030  16090  16100  16150
            16170  16190  16420  16740  17020  17400  17610  17620  17800  17820  17950  17970  17980  18020  18040
            18070  18080  18350  18700  18880  19090  19420  19590  19800  19880  20120  20130  20170  20190  20380
            20680  20700  20720  20730  20890  21080  21350  21430  21960  22190  22410  22740  22870  23150  23240
            23400  23430  23700  23710  23780  23810  23820  23890  23900  23990  24000  24080  24100  24180  24190
            24400  24980  25630  25640  25700  25720  26010  26430  27130  27140  27190  27210  27650  27930  27960
            28130  28320  28620  28640  28650  28760  28830  29060  29400  29430  29630  29860  30100  30130  30140
            30470  31180  31340  31730  32160  32210  32340  32680  32760  33040  33130  33140  33240  33260  33280
            33390  33460  34260  34270  34310  34410  34440  34590  34600  34720  34830  35180  35210  35260  35270
            35340  35590  35620  35670  35700  35710  35720  35740  35750  35800  36280  36290  36370  36470  36480
            36580  36600  36840  36860  36880  37090  37530
$$GETT      15950  16020  16090  17610  17970  18040  20120  23700  23890  23990  25630  27130  33130  34260  34440
            36280  36470
$$LPCN      23360  2339   27890  2792   29360  2939   35140  3517   35550  3558   35630  3566
$$NEWT         10   7490   1138   1202   1295   1384   1466   1547   1580   1649   1683   1816   1892   1962   2118
            2293   2477   2626   2748   2889   3029   3248   3353
$$POP       11010  1102   11750  11830  12240  12370  12530  12700  12890  13120  13250  13410  13580  13770  14070
            14230  14400  14590  14830  14990  15160  15350  15710  15950  1596   16020  1603   16090  1610   16150
            16170  16190  16420  16740  17020  17400  17610  1762   17800  17820  17950  17970  1798   18020  18070
            1808   18350  18700  18880  19090  19420  19590  19800  19880  20120  2013   20170  20190  20380  20680
            20700  20720  2073   20890  21080  21350  21430  21960  22190  22410  22740  22870  23150  23240  23360
            23390  2340   2343   23700  2371   23780  23810  2382   23890  2390   23990  2400   24080  24100  24180
            2419   24400  24980  25630  2564   25700  25720  26010  26430  27130  2714   27190  27210  27650  27890
            27920  2793   2796   28130  28320  28620  28640  2865   28760  28830  29060  29360  29390  2940   2943
            29630  29860  30100  30130  3014   30470  31180  31340  31730  32160  32210  32340  32680  32760  33040
            33130  3314   33240  33260  33280  33390  33460  34260  3427   34310  34410  34590  3460   34720  34830
            35140  35170  3518   3521   35260  3527   35340  35550  35580  3559   3562   35630  35660  3567   3570
            35710  3572   35740  3575   35800  36280  3629   36370  36470  3648   36580  36600  36840  36860  36880
            37090  37530
$$PUSH      10930  1094   1096   11010  11590  1160   11690  1171   12150  1220   12310  1233   12470  1249   12640
            1266   12830  1285   13030  1308   13190  1321   13350  1337   13520  1354   13710  1373   14010  1403
            14170  1419   14340  1436   14530  1455   14770  1479   14930  1495   15100  1512   15290  1531   15600
            1562   15910  1593   15950  1597   15980  1600   16020  1604   16050  1607   16090  1611   16250  1627
            16630  1665   16930  1698   17070  1708   17350  1736   17530  1755   17610  1763   17750  1776   17840
            1786   17870  1789   17970  1799   18070  18260  1831   18620  1863   18820  1884   19000  1905   19370
            1938   19530  1955   19710  1976   19820  1984   20050  2006   20070  2009   20120  2014   20250  2027
            20460  2047   2049   20510  2053   20540  2056   20720  20770  2079   20990  2101   21260  2131   21370
            2139   21820  2184   22030  2205   22270  2229   22590  2261   22770  2279   23060  2311   23180  2320
            23360  2338   2339   2343   23600  2361   2363   23640  2366   23700  2372   23810  23850  2387   23890
            2391   23930  2397   23990  2401   24340  2436   24920  2494   25530  2555   25570  2559   25630  2565
            25950  2597   26360  2641   26990  2701   27060  2708   27130  2715   27560  2761   27890  2791   2792
            2796   28090  2810   28280  2829   28500  2852   28710  2873   28780  2880   28960  2901   29360  2938
            2939   2943   29570  2958   29780  2979   30010  3003   30380  3043   31160  3117   31280  3130   31670
            3169   32040  3206   32080  3210   32250  3227   32590  3264   32700  3272   32990  3300   33070  3309
            33130  3315   33160  3318   33190  3321   33340  3336   33410  3343   33830  3384   34180  3419   34210
            3423   34260  3428   34350  3437   34590  34670  3469   34910  3492   35140  3516   3517   3521   35410
            3542   35550  3557   3558   3562   35630  3565   3566   3570   36010  3602   36180  3619   36200  3622
            36230  3625   36280  3630   36470  3649   36630  3665   36750  3677   36780  3680   37160  3717
$$SET       44170  4426   4427   4428   4429   4431   4433   4434   4435
$$SETM      10660
$$SETS      10930  10940  10960  11010  11590  11600  11690  11710  12150  12200  12310  12330  12470  12490  12640
```

```
            12660   12830   12850   13030   13080   13190   13210   13350   13370   13520   13540   13710   13730   14010   14030
            14170   14190   14340   14360   14530   14550   14770   14790   14930   14950   15100   15120   15290   15310   15600
            15620   15910   15930   15970   15980   16000   16040   16050   16070   16110   16250   16270   16630   16650   16930
            16980   17070   17080   17350   17360   17530   17550   17630   17750   17760   17840   17860   17870   17890   17990
            18070   18260   18310   18620   18630   18820   18840   19000   19050   19370   19380   19530   19550   19710   19760
            19820   19840   20050   20060   20070   20090   20140   20250   20270   20460   20470   20490   20510   20530   20540
            20560   20720   20770   20790   20990   21010   21260   21310   21370   21390   21820   21840   22030   22050   22270
            22290   22590   22610   22770   22790   23060   23110   23180   23200   23360   23380   23590   23430   23600   23610
            23630   23640   23660   23720   23810   23850   23870   23910   23930   23970   24010   24340   24360   24920   24940
            25530   25550   25570   25590   25650   25950   25970   26360   26410   26990   27010   27060   27080   27150   27560
            27610   27890   27910   27920   27960   28090   28100   28280   28290   28500   28520   28710   28730   28780   28800
            28960   29010   29360   29380   29390   29430   29570   29580   29780   29790   30010   30030   30380   30430   31160
            31170   31280   31300   31670   31690   32040   32060   32080   32100   32250   32270   32590   32640   32700   32720
            32990   33000   33070   33090   33150   33160   33180   33190   33210   33340   33360   33410   33430   33840   34180
            34190   34210   34230   34280   34350   34370   34590   34670   34690   34920   35140   35160   35170   35210   35420
            35550   35570   35580   35620   35630   35650   35660   35700   36020   36180   36190   36200   36220   36230   36250
            36300   36490   36630   36650   36750   36770   36780   36800   37170
$$SKIP        10    7490    1222    1310    1613    1700    1810    1833    1907    1978    1986    2036    2066    2087    2111
            2133    2141    2194    2217    2239    2272    2313    2322    2442    2496    2763    2860    2904    2961    2984
            3008    3045    3141    3266    3274    3348
.EQUAT        1#     639
.HEADE        1#     617
.KT11         1#
.SETUP        1#    1027
.SWRMI        1#     627
.SWRLO     6380#
.$AC11        1#     876
.$APTB        1#    9530
.$APTH        1#     887
.$APTY        1#    4075
.$ASTA        1#
.$CATC        1#     865
.$CMTA        1#     909
.$DB2D        1#
.$DB20        1#
.$DIV         1#
.$EOP         1#    3756
.$ERRO        1#    4132
.$ERRT        1#
.$MULT        1#
.$POWE        1#    3795
.$RAND        1#
.$RDDE        1#
.$RDOC        1#
.$READ        1#    3936
.$R2AZ        1#
.$SAVE        1#
.$SB2D        1#
.$SB20        1#
.$SCOP        1#    4186
.$SIZE        1#
.$SUPR        1#
.$TRAP        1#    4394
.$TYPB        1#
.$TYPD        1#    4250
.$TYPE        1#    3840
```

```
MAINDEC-11-DVDVC-C      MACY11 30A(1052)  12-SEP-84  15:41  PAGE 141
CVDVCC.P11    12-SEP-84 08:55           CROSS REFERENCE TABLE -- MACRO NAMES

.$TYPO      1#   4317
.$40CA      1#
.1170       1#


. ABS.  014416     000


 ERRORS DETECTED:  0

 CVDVCC,CVDVCC/SOL/CRF=CVDVCC.MLB/ML,SYSMAC.SML,CVDVCC.P11
 RUN-TIME:  77 77 6 SECONDS
 RUN-TIME RATIO: 246/162=1.5
 CORE USED:  43K  (85 PAGES)
```