VT61

EXERCISER

MD-11-DZVTJ-A

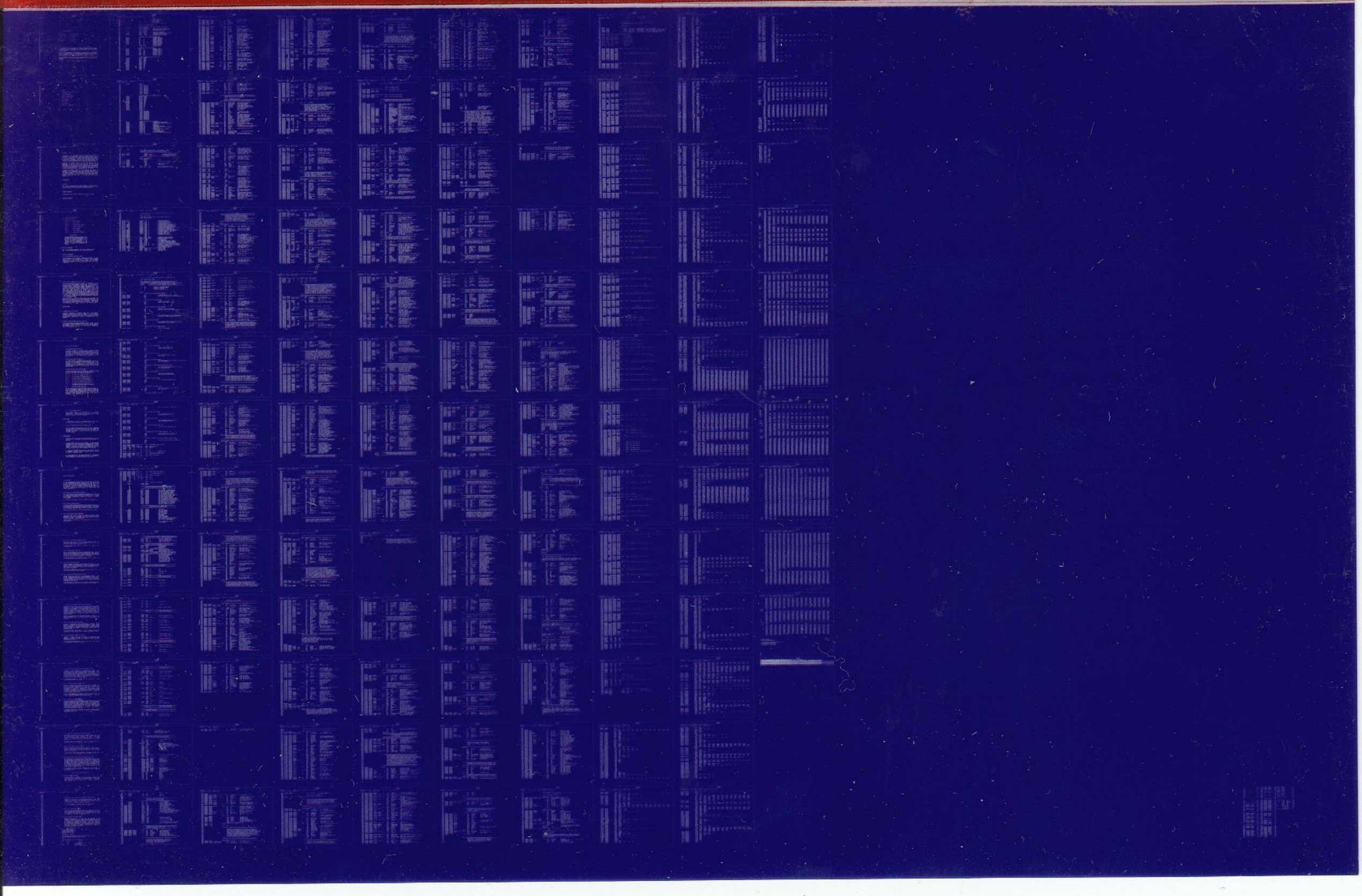
EP-DZVTJ-A-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN USA



.REM %

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZVTJ-A
PRODUCT NAME:	VT61 EXERCISOR
DATE:	30-JAN-76
MAINTENANCE:	DIAGNOSTIC GROUP
AUTHOR:	PAUL NELSON

TABLE OF CONTENTS

1. ABSTRACT

THIS PROGRAM IS AN ACCEPTANCE TEST FOR THE ENTIRE VT61 FAMILY OF TERMINALS. THE FUNCTIONAL TESTING IS BASED UPON A SET OF TERMINAL FUNCTIONS WHICH ARE COMMON THROUGHOUT THE ENTIRE FAMILY OF VT61 TYPE TERMINALS. THE FUNCTIONS AND THEIR DERIVED TESTING IS DESIGNED TO COMPLETELY CHECK(AT THE FUNCTIONAL LEVEL) THE TERMINAL MICRO-PROCESSOR AND ASSOCIATED RAMS. ALL TRANSMISSIONS TO THE VT61 WILL BE PRECEEDED BY A SOM AND TERMINATED BY AEOM//.

THERE ARE TWO DISTINCT MODES IN WHICH THE PROGRAM CAN BE OPERATED. IN "AUTO" MODE UP TO 2 DJ11'S WITH UP TO 32 OPERATIONAL VT61'S WILL BE MAPPED AND ALL WILL BE TESTED SEQUENTIALLY. ALL TESTS WHICH DO NOT REQUIRE MANUAL INTERVENTION OR VISUAL SCREEN OBSERVATION (TESTS 1 THRU 20) WILL BE EXECUTED FOR EACH VT61 REPETITIVELY. ALL ERRORS WILL BE REPORTED ON THE SYSTEM CONSOLE (WHICH IS NOT TESTED EVEN IF IT IS A VT61).

IN MANUAL MODE CONSOLE ENTRY OF THE ADDRESSES AND TESTS IS REQUIRED. THE ADDRESSES AND TESTS CAN BE ENTERED IN A NON-SEQUENTIAL MANNER AND THE SUBSEQUENT EXECUTION WILL FOLLOW THE ENTRY SEQUENCE. THIS MODE MUST BE UTILIZED TO ENTER THE KEYBOARD TESTS, DATA LOOP TEST, AND PRINTER CONTROLLER TEST. SEQUENCE COMPLETION WILL EXIT TO THE RE-START POINT FOR THE MANUAL TEST.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP 11 FAMILY COMPUTER WITH 8K WORDS OF MEMORY, A CONSOLE, AND UP TO 32 VT61'S CONNECTED TO THE HOST COMPUTER VIA DJ11(S). VT61 MUST BE IN REMOTE; FULL DUPLEX AND AT LEAST 300 BAUD.

3. LOADING PROCEDURE

PROCEDURE FOR NORMAL BINARY PAPERTAPES SHOULD BE FOLLOWED.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

STANDARD PDP 11 FORMAT

SW15 = 1	HALT ON ERROR.
SW14 = 1	LOOP ON TEST
SW13 = 1	INHIBIT ERROR TYPEOUTS
SW11 = 1	INHIBIT ITERATIONS
SW10 = 1	BELL ON ERROR
SW9 = 1	LOOP ON ERROR
SW8 = 1	LOOP ON TEST IN SWR<7:0>

SPECIAL NOTE

IF THE COMPUTER UTILIZED IS A LSI 11 OR A COMPUTER WITHOUT A SWITCH REGISTER. THE PROGRAM WILL UTILIZE LOCATIONS 174 AND 176 AS A "DISPLAY" REGISTER AND A "SWITCH" REGISTER RESPECTIVELY. THE OPERATOR WILL BE RESPONSIBLE FOR THE LOADING OF THE "SWITCH" REGISTER LOCATION PRIOR TO STARTING OR RESTARTING THE PROGRAM.

4.2 STARTING ADDRESSES

200 IS THE STARTING ADDRESS OF THE "AUTO" ACCEPTANCE TEST
204 IS THE STARTING ADDRESS ON THE "MANUAL" SELECT TEST.

5. OPERATING PROCEDURE

5.1 AUTO ACCEPTANCE MODE (SA = 200).

IN THIS MODE THE ONLY OPERATOR INTERVENTION REQUIRED IS SWR OPTION SELECTIONS SUCH AS LOOP ON TEST (SWR 11), BELL ON ERROR (SWR 0), ECT.. THE PROGRAM WILL, WITHOUT ANY EXTERNAL INTERVENTION, LOCATE THE DJ11(S)/LINES WITH VT61 TYPE UNITS ATTACHED AND SEQUENTIALLY TEST ALL UNITS REPETITIVELY WITH TESTS 1 THRU 20.

5.2 MANUAL UNIT/TEST SELECTION MODE (SA = 204)

THIS MODE REQUIRES THE OPERATOR TO ENTER THE ADDRESSES OF THE DJ11'S TO BE TESTED (FORMAT IS 17XXXX, ECT, -UP TO 2 ENTRIES). THE ENTRIES MUST BE SEPARATED BY COMMAS AND TERMINATED WITH A CARRIAGE RETURN. ENTERING AN ILLEGAL ADDRESS WILL RESULT IN A "?" BEING TYPED AND THE ADDRESS IGNORED! THE PROGRAM WILL THEN REQUEST THE LINES TO BE TESTED ,IN BINARY FORMAT. TO TEST LINES ON TWO DJ11S, INSERT A -1(177777) AT THE END OF THE LINE LIST FOR EACH DJ11 AND TERMINATE THE ENTIRE LIST WITH A 0 WORD(000000).EXAMPLE- TEST LINE 1 OF 1ST DJ11 AND LINE 4 OF 2ND DJ11; ENTERED LIST WOULD BE 000002,177777 ,000020,177777,000000 C/R. THE OPERATOR MUST THEN, UPON PROGRAM REQUEST, ENTER A LIST OF TESTS TO BE EXECUTED IN THE SAME FORMAT AS THE ADDRESS ENTRY (I.E.YY,ZZ,C/R). PRECEEDING THE TERMINATING CARRIAGE RETURN WITH A 377 OCTAL WILL RESULT IN THE TESTS BEING REPETITIVELY EXECUTED FOR ALL ADDRESSES ENTERED.

SIMPLY DEPRESSING A CARRIAGE RETURN WHEN UNIT ADDRESSES ARE REQUESTED WILL RESULT IN THE MAPPING AND TESTING OF ALL GOOD DJ11(S)/LINES WITH OPERATIONAL VT61'S ATTACHED. HOWEVER, THE TEST LIST MUST STILL BE ENTERED VIA THE CONSOLE!! WHEN RUNNING THE EXERCISOR IN MANUAL MODE A CONTROL C (03 OCTAL) WILL RESULT IN THE TERMINATION OF TESTING AT THE END OF THE CURRENT SUBTEST.

6. ERRORS-GENERAL

6.1 NO OPERATIONAL VT61 ATTACHED

IF THE UNIT SELECTED (IN "MANUAL" MODE) OR IN THE MAPPING OPERATION ("AUTO" MODE) DOES NOT RESULT IN A UNIT WHICH IS CAPABLE OF RESPONDING TO THE TEST THE MESSAGE "NO VT61 RESPONDED TO ESCZ SEQ. AUTO RETRY IN 30 SEC". WILL BE DISPLAYED ON THE CONSOLE EVERY 30 SECONDS UNTIL THE TEST IS STOPPED OR A UNIT RESPONDS.

6.2 EXCESSIVE "FATAL" ERRORS FROM UNIT UNDER TEST

IF TEN FATAL ERRORS (INCOMPLETE TRANSMIT/RECIEVE CYCLES) OCCURS THE MESSAGE "TESTING ABORTED-TOO MANY FATAL XMITS" WILL BE DISPLAYED AND THE TEST WILL EXIT TO THE INITIAL SETUP SEQUENCE OF THE REQUESTED MODE. IF THE TEST THEN LOCATES AN OPERATIONAL UNIT, IT WILL BEGIN TESTING IT.

6.3 COMMON ERROR MESSAGES

A. ESCAPE SEQUENCE ERROR (ERROR 1)

THIS ERROR MESSAGE IS RETURNED WHEN A SPECIFIC ESCAPE SEQUENCE DID NOT ELICIT THE EXPECTED RESPONSE FROM THE UNIT UNDER TEST. MESSAGE RETURNS TEST #, ERROR PROGRAM COUNT AND TWO WORDS WHICH CONTAIN UP TO 4 BYTES OF THE FAILING ESCAPE SEQUENCE (I.E. IF "TRANSMIT ALL" FAILED;THE ESC,O, V WOULD BE DISPLAYED IN THE FORMAT BYTE 1+2=015517,BYTE 3+4= 000126).

B. RECEIVE STATUS ERROR (ERROR 2)

THIS ERROR MESSAGE IS RETURNED IF ANY OF BITS 12, 13, OR 14 ARE SET IN THE INTERFACE RECEIVE BUFFER REGISTER. DATA DISPLAYED IS THE ADDRESS OF THE CSR (CONTROL AND STATUS REGISTER) OF THE FAILING UNIT, THE CONTENTS OF THE FOREMENTIONED CSR , THE ERROR BITS FROM THE RECEIVE BUFFER REGISTER, AND THE CHARACTER WHICH WAS STORED WHEN THE ERRORS WERE DETECTED.

C. SOFTWARE STATUS (VSTAT) ERROR (ERROR 3)

THE LOCATION TAGGED "VSTAT" IS USED BY THE PROGRAM TO STORE DYNAMIC CONDITIONS RELATING TO THE UNIT UNDER TEST. THE BITS WHICH MAY CAUSE A SOFTWARE STATUS ERROR ARE:

BIT 15	SET FOR XOFF, CLEARED FOR XON
BIT 14	SET WHEN START OF MESSAGE RECEIVED
BIT 13	SET WHEN END OF MESSAGE RECEIVED
BIT 12	SET FOR A PERIPHERAL ABORT MESSAGE
BIT 10	SET WHEN AN INTERFACE ERROR DETECTED
BIT 7	SET WHEN AN XOFF WAS DETECTED AND THE TRANSMITTER WAS SHUT DOWN BY THE SOFTWARE.
BIT 1	SET WHEN TRANSMIT COMPLETE

THE ONLY BIT WHICH WILL UNCONDITIONALLY CAUSE THIS ERROR IS BIT 12 (PERIPHERAL ABORT) ALL OTHER BITS WILL BE SET AND RESET AND AN ERROR IS DEPENDENT UPON EXPECTED CONDITIONS (I.E. AFTER A COMPLETE TRANSMISSION BITS 1, 13 AND 14 MUST BE SET AND OTHERS MENTIONED RESET OR AN ERROR WILL BE REPORTED). DATA DISPLAYED IS THE PASS #, THE TEST #, EXPECTED STATUS AND ACTUAL STATUS.

D. VT61 HUNG ERROR (ERROR 11)

THIS ERROR MESSAGE IS DISPLAYED IF A COMPLETE TRANSMISSION(S) DOES NOT RESULT IN A SOM(S), AN EOM(S) AND TRANSMIT DONE. THIS ERROR IS A FATAL ERROR AND TEN OF THESE ERRORS WILL RESULT IN THE TEST ABORTING.

7. RESTRICTIONS

A. IT IS IMPERATIVE THAT BOTH THE INTERFACE AND THE VT61 SHOULD BE PLACED IN FULL DUPLEX AND REMOTE (NOT LOCAL) MODE.

B. UNIT TO BE TESTED CANNOT BE THE CONSOLE DEVICE.

C. FOR THE AUTOMATIC TEST MAPPING OF THE DJ11'S, ALL ADDRESSES FOR THE UNITS TO BE TESTED MUST BE WITHIN THE STANDARD DEC ADDRESSES AND VECTORS. IF THIS IS NOT THE CASE, THE PROCEDURE OUTLINED IN SECTION 8-B MUST BE FOLLOWED BEFORE TESTING IS BEGUN.

8. MISCELLANEOUS

A. EXECUTION TIME FOR THE AUTO SELECTION TESTS (TEST 1-20) WITH UNITS SET TO A BAUD RATE OF 9600 BAUD IS APPROXIMATELY 90 SECONDS.

B. TO TEST A DEVICE (DJ11 WITH VT61 ATTACHED) AT NON-STANDARD ADDRESSES THE LOCATION "STRTAB" CAN BE MODIFIED TO CONTAIN THE LOWEST OF THE NON-STANDARD ADDRESSES AND LOCATON "ENDTAB" MODIFIED TO CONTAIN THE HIGHEST NON-STANDARD ADDRESS. ALL INTERFACES WITHIN THE NEW ADDRESSES WILL BE MAPPED AND TESTED IF THE PROPER RESPONSES ARE OBTAINED.

C. TO CHANGE THE NUMBER OF FATAL ERRORS ALLOWED BEFORE TESTING IS ABORTED, LOCATION "ALWCNT" (LOADED WITH 10) CAN BE MODIFIED TO THE DESIRED COUNT.

D. ALL TESTS EXCEPT TEST 1 AND TEST 23 ARE RUN IN MAINTENANCE MODE, THEREFORE ALL TRANSMISSIONS FROM THE VT61 ARE EXPECTED TO BE PRECEEDED BY A SOM AND TERMINATED WITH A EOM.

9. PROGRAM DESCRIPTION

9.0 INITIALIZATION

IN "AUTO" SEQUENCE MODE THIS SECTION OF THE TEST MAPS ALL DEVICES IN THE PRE-DETERMINED AREAS. DEVICES ARE THEN TESTED FOR INTERRUPT CAPABILITY VIA THE "MAINTENANCE" BIT AND ALL UNITS WHICH DO NOT OR CANNOT RESPOND ARE PURGED FROM THE TABLE. ALL UNITS ARE THEN ISSUED THE "ESCAPE Z" SEQUENCE AND THOSE WHICH DO NOT RESPOND, OR DO NOT RESPOND WITH THE PROPER "IDENT" ARE PURGED. ALL OPERATIONAL UNITS ARE STORED IN A TABLE(DLTBL) AND TESTED SEQUENTIALLY.

9.1 TEST 1 CHECK ALL COMMON ESCAPE SEQUENCES.

THIS TEST ISSUES ALL ESCAPE SEQUENCES AND INSURES THE VT61 HAS NOT FAILED DURING AN ESC SEQUENCE BY ISSUING A ESC Z TO FORCE A VT61 RESPONSE. THE PURPOSE OF THE TEST IS TO ATTEMPT TO INSURE THAT SUBSEQUENT TESTS WILL NOT RESULT IN A "HUNG" UNIT. DATA IS NOT EVALUATED.

ALL ERRORS ARE REPORTEDED AS ESCAPE SEQUENCE FAILURES(ERROR 1).

9.2 TEST 2 CHECK MAINTENANCE MODE.

ROUTINE TO INSURE ENTERING MAINTENANCE MODE CAUSES SOM AND EOM TO BE APPENDED TO ALL TRANSMITS FROM VT61 UNDER TEST. MAINTENANCE MODE IS ENTERED, THEN AN ESCAPE Z SEQUENCE IS ISSUED TO THE UNIT AND THE RESULTING RESPONSE FROM THE VT61 IS CHECKED FOR SOM/EOM.

ERROR 22 WILL BE ISSUED IF EITHER COMPONENT(SOM/EOM) IS MISSING.

9.3 TEST 3 CHECK DIRECT CURSOR ADDRESSING

THIS TEST INSURES THAT THE CURSOR WILL RESPOND TO DIRECT CURSOR ADDRESSING. THE UNIT IS RESET AND THE CURSOR POSITION IS VERIFIED TO BE HOME. THE CURSOR IS THEN MOVED TO ROW 23 COLUMN 80 AND THE POSITION IS AGAIN VERIFIED.

CURSOR POSITIONING ERRORS(ERROR 7) ARE REPORTED IF THE POSITIONS ARE INCORRECT.

9.4 TEST 4 CHECK LINEAR ADDRESSING MODE.

ROUTINE TO INSURE THE UNIT CAN ENTER LINEAR ADDRESSING MODE. 81 CHARACTERS ARE ISSUED TO THE UNIT UNDER TEST THEN THE CURSOR POSITION IS READ AND MUST BE ROW1, COL.0.

AN ESCAPE SEQUENCE ERROR (ERROR 1)IS ISSUED IF THE CURSOR IS NOT AT ROW1,COL.0

9.5 TEST 5 CHECK XON/XOFF FROM VT61

TEST TO INSURE OPERATION OF XON/XOFF COMMANDS FROM VT61. XOFF IS FORCED BY TRANSMITTING THE DATA ON LINE 23 WHILE SIMULTANEOUSLY FILLING THE SILO WITH NEW DATA. AFTER SENSING THE XOFF, THE TEST WAITS FOR THE TRANSMIT TO FINISH AND INSURES XON OCCURS BEFORE THE MAXIMUM TRANSFER TIME HAS ELAPSED. (30 SECONDS)

ERRORS ARE REPORTED IF THE FORMAT OF ERROR 3(VSTAT ERRORS) AND WILL REFLECT EITHER LACK OR EXCESS OF BIT 15.

9.6 TEST 6 CHECK XON/XOFF TO VT61

ROUTINE TO VERIFY OPERATION OF XOFF AND XON TO THE VT61. A FULL SCREEN TRANSMIT IS INITIATED AND A SERIES OF XOFFS AND XONS ARE ISSUED TO THE TERMINAL SEQUENTIALLY. ERRORS ARE REPORTED IF A XOFF DOES NOT STOP, OR A XON RESTART THE TRANSMISSION. TEST IS ENDED WHEN EOM IS SENSED.

ERRORS ARE REPORTED(ERROR 15 FOR XOFF FAILURE AND ERROR 16 FOR A XON FAILURE)AS SPECIFIC ERROR MESSAGES.

9.7 TEST 7 CHECK RAM AND COMMUNICATIONS PATHS

ROUTINE TO TEST VT61 RAM AND THE COMMUNICATION PATHS. THIS ROUTINE ISSUES A SERIES OF FULL SCREEN PATTERNS (77/100, 100/77, 52/125, INCREMENTING, AND REV. VIDEO INCREMENTING) TO THE VT61. THE FULL SCREEN IS THEN TRANSMITTED TO THE HOST AND AFTER EACH ITERATION RECEIVED DATA IS CHECKED AND ALL ERRORS (INCLUDING TRANSMISSION) ARE REPORTED.

ERRORS REPORTED COULD BE ERROR 2 FOR A RECEIVE STATUS ERROR,ERROR 4 FOR DATA ERRORS AND ERROR 5 FOR A RECEIVE BYTE COUNT ERROR.

9.10 TEST 10 CHECK TRANSMIT AND RECEIVE CHECKSUMS.

ROUTINE TO TEST THE ABILITY OF THE VT61 TO CALCULATE AND TRANSMIT CHECKSUMS OF BOTH TRANSMITTED AND RECEIVED DATA. SUBTEST "A" TRANSMITS A FULL BUFFER UPDATING A CALCULATED CHECKSUM ON EACH CHARACTER TRANSMITTED. AN ESCAPE SEQUENCE REQUESTING THE RECEIVER CHECKSUM IS EMBEDDED AT THE END OF XMIT BUFFER AND THE RECEIVED CHECKSUM IS COMPARED TO THE CALCULATED. SUBTEST "B" PERFORMS THE SAME TYPE OF CHECK ON THE VT61 TRANSMIT CHECKSUM, UTILIZING THE DATA SENT TO THE VT61 IN SUBTEST "A", DURING A FULL SCREEN TRANSMIT.

ERROR 13 IS ISSUED(WITH CALCULATED AND RECEIVED CHECKSUM) IF A RECEIVE CHECKSUM ERROR IS DETECTED. ERROR 14 IS ISSUED (WITH SAME DATA AS ERROR 13) IF A VT61 TRANSMIT CHECKSUM ERROR IS DETECTED.

9.11 TEST 11 CHECK BASIC CURSOR COMMANDS

ROUTINE TO INSURE BASIC CURSOR COMMANDS RESULT IN CORRECT CURSOR MOVEMENT. COMMANDS ARE ISSUED IN THE SEQUENCE: RESET, CURSOR RIGHT, CURSOR DOWN, CURSOR LEFT, AND CURSOR UP. THE READ CURSOR POSITION COMMAND IS ISSUED AFTER EVERY MOVE CURSOR COMMAND AND RECEIVED POSITION IS COMPARED TO THE EXPECTED POSITION AND ANY ERRORS REPORTED.

AN ESCAPE SEQUENCE ERROR(ERROR 1) AND A CURSOR POSITIONING ERROR(ERROR 6) ARE ISSUED IF ANY FUNCTIONS ARE DETECTED TO FAIL.

9.12 TEST 12 CHECK READ CHARACTER AT CURSOR

ROUTINE TO INSURE THAT READ CHARACTER AT CURSOR FUNCTIONS CORRECTLY. COMMAND SEQUENCE IS: RESET, A, CURSOR LEFT, READ CHARACTER AT CURSOR. AN ERROR IS REPORTED IF THE CHARACTER RECEIVED IS NOT AN "A".

AN ESCAPE SEQUENCE ERROR(ERROR 1) AND A DATA COMPARE ERROR(ERROR 4) ARE ISSUED IF A FAILURE IS DETECTED.

9.13 TEST 13 CHECK REPLACE AND INSERT CHARACTER MODES

ROUTINE TO VERIFY OPERATION OF REPLACE AND INSERT MODE. INITIALLY ROW 0 IS WRITTEN TO 80 INCREMENTING CHARACTERS; ON THE FIRST PASS (REPLACE MODE) A CHARACTER(172) IS REPLACED AT THE HOME POSITION AND THE CHARACTERS AT ROW0, COL.0 AND ROW1, COL.0 ARE READ AND VERIFIED TO BE A "172" AND A "NULL" RESPECTIVELY. ON THE SECOND PASS, INSERT MODE IS ENTERED AND THE RESULTING INSERTION (AT THE HOME POSITION) IS VERIFIED. ROW0, COL.0 SHOULD BE "172" AND ROW1, COL.0 SHOULD BE "161".

IF AN ERROR IS DETECTED IN EITHER MODE,THE APPROPRIATE ESCAPE SEQUENCE ERROR(ERROR 1) IS ISSUED.

9.14 TEST 14 CHECK VT61 SCROLL CAPABILITIES.

ROUTINE TO INSURE VT61 WILL SCROLL IF A LINE FEED IS ISSUED FROM ROW 23 OR A DATA INSERT FROM ROW 23 COL. 79. IN SUBTEST "A", ROW 0 IS INITIALLY WRITTEN TO A 0 AND ROW 1 A 1. AFTER COMPLETION OF A LINE FEED (AND RESULTING SCROLL) ROW 00, COL.00 IS EXPECTED TO CONTAIN A 1. IN SUBTEST "B", THE CURSOR IS PLACED AT ROW23, COL.79 AND A DATA CHARACTER "A" IS ENTERED. THE CURSOR POSITION IS THEN READ AND SHOULD BE ROW23, COL.00. THE CHAR. AT HOME IS VERIFIED TO BE A NULL.

A SCROLL ERROR(ERROR 23) IS ISSUED IF EITHER FUNCTIONS FAIL TO ELICIT THE PROPER RESPONSE FROM THE UNIT UNDER TEST. THE ERROR PC WILL DISTINGUISH BETWEEN THE FAILING FUNCTIONS.

9.15 TEST 15 CHECK ALL SCREEN ADDRESSES.

THIS TEST INSURES THAT THE VT61 CURSOR CAN BE POSITIONED TO EVERY POSSIBLE ROW/COLUMN POSITION ON THE SCREEN. THIS IS TESTED BY FILLING THE COMPLETE SCREEN (EXCEPT ROW 23,COL.79 WHICH WILL CONTAIN A "NULL") WITH THE CHARACTER "A" AND THEN POSITIONING THE CURSOR (VIA DCA) TO EVERY POSITION AND THE "A" AT THAT POSITION IS REPLACED WITH A SPACE(OCTAL 40). THE SCREEN IS THEN READ TO VERIFY THAT ONLY SPACES EXIST ON THE SCREEN. ALL POSITIONS CONTAINING NON-SPACES ARE REPORTED.

ALL ERRORS DETECTED WILL BE REPORTED AS DIRECT CURSOR ADDRESS ERROS(ERROR 7),AND WILL CONTAIN THE POSITION THE BAD DATA(NON -SPACE) WAS DETECTED AT.

9.16 TEST 16 CHECK LINE FEED AND CARRIAGE RETURN

ROUTINE TO INSURE PROPER OPERATION OF CARRIAGE RETURN AND LINE FEED DURING NORMAL MODE. INITIALLY THE CURSOR IS SET (VIA D.C.A.) TO ROW0, COL 20 AND A LINE FEED IS ISSUED, THE CURSOR POSITION IS THEN READ AND MUST BE ROW1, COL.20. A CARRIAGE RETURN IS THEN ISSUED AND CURSOR POSITION VERIFIED TO BE ROW1, COL0.

AN ESCAPE SEQUENCE ERROR(ERROR 1) AND A CURSOR POSITIONING ERROR(ERROR 6) WILL BE ISSUED IF AN ERROR IS DETECTED.

9.17 TEST 17 CHECK ERASE TO END OF SCREEN

ROUTINE TO VERIFY PROPER OPERATION OF ERASE TO END-OF-SCREEN. SCREEN IS WRITTEN TO 1920 INCREMENTING CHAR. ERASE TO END OF SCREEN IS THEN ISSUED AND THE ENTIRE SCREEN IS READ VERIFYING THAT IT IS ALL NULLS.

IF ANY NON-NULL POSITIONS ARE DETECTED, AND ESCAPE SEQUENCE ERROR (ERROR 1) AND A DATA ERROR(ERROR 4) WILL BE ISSUED.

9.20 TEST 20 CHECK SELF TEST, COPIER, AND ISSUE END OF PASS.

SELF TEST (ESC T) IS ISSUED TO THE UNIT UNDER TEST AND AN SELF TEST ERROR(ERROR 10) IS ISSUED IF THE UNIT CANNOT RESPOND TO AN "ESCAPE Z" SEQUENCE AFTER SELF TEST IS COMPLETE. IF SELF TEST IS SUCCESSFUL THE SCREEN IS WRITTEN TO 23 LINES OF INCREMENTING CHARACTERS AND 23 LINES OF INCREMENTING CHAR. IN REVERSE VIDEO. THE "IDENT" IS THEN CHECKED AND IF A COPIER IS PRESENT A COPY SCREEN COMMAND IS ISSUED (NOTE: THIS COMMAND WILL CAUSE THE UNIT TO BE "BUSY" AND NOT RESPOND TO ANY FURTHER COMMANDS UNTIL THE SCREEN HAS BEEN COMPLETELY COPIED.)

IF THE IDENT INDICATES A COPIER IS PRESENT AND THE COPY SCREEN IS INITIATED,BUT NOT COMPLETED, A "PERIPHERAL ABORT" (ERROR 20) ERROR IS ISSUED.

END OF AUTO-ACCEPTANCE TESTS

9.21 TEST 21 KEYBOARD ECHO TEST

ROUTINE TO ECHO THE KEYBOARD. KEYS FOR TAB, BELL, CARRIAGE AND LINE FEED ECHO A MNEMONIC, NON-DISPLAY CHAR. ECHO OCTAL EQUIVALENTS AND DISPLAY CHAR. ECHO THEMSELVES. (EXAMPLES- CHAR., SPACE, ESC, SPACE OR 037, SPACE.) A CONTROL C (003) WILL CAUSE A TEST EXIT.

9.22 TEST 22 TEST A LINE PRINTER(PRINTER CONTROLLER MODE)

ROUTINE TO UTILIZE THE VT61 AS A PRINTER CONTROLLER. AFTER TEST MESSAGE IS DISPLAYED, THE TEST WAITS FOR A C/R BEFORE ACTUALLY ENTERING TEST. A PATTERN OF INCREMENTING, ROLLING CHAR. WILL BE OUTPUTTED UNTIL A CONTROL C (003) IS RECEIVED.

IF THE LINE PRINTER IS DISABLED AFTER THE INITIALIZATION OF THE TEST, A "PERIPHERAL ABORT" (ERROR 20) IS ISSUED.

9.23 TEST 23 UNIT SIMULATOR TEST

ROUTINE TO LOOP DATA/COMMANDS FROM THE VT61 BACK TO THE VT61. DATA TRANSMISSIONS RESULTING FROM A ESC SEQUENCE WILL ALSO BE LOOPED AND WILL ENTER THE SCREEN AT THE CURSOR POSITION. THIS TEST CAN BE USED TO SIMULATE, OR CREATE, SPECIFIC SCREEN PATTERNS AND OPERATIONS. A CONTROL C (003) EXITS TEST.

9.24 TEST 24 PRODUCTION KEYBOARD TEST

PRODUCTION KEYBOARD TEST. ALL KEYS MUST BE DEPRESSED IN THE SEQUENCE INDICATED ON THE SCREEN. ALL ERRORS OR MISTAKES ARE DISPLAYED IN OCTAL POSITIONAL FORMAT AND THE CORRECT KEY POSITION IN THE ROW IS DISPLAYED IN DECIMAL. THIS TEST IS RUN IN MAINTENANCE MODE, THEREFORE THE KEYS WILL ECHO THEIR POSITION, NOT THEIR INDICATED MNEMONIC. THE EXCEPTIONS ARE THE INDIVIDUAL TESTS FOR THE SHIFT AND CONTROL FUNCTIONS.THESE TESTS ARE EXPLICITELY DEFINED BY MESSAGES TO THE OPERATOR. 10 ERRORS WILL CAUSE AN AUTOMATIC EXIT FROM TEST.

```
        %
        .NLIST  MD,MC,CND
        .LIST   ME
.TITLE  MAINDEC-11-DZVTJ-A
;*COPYRIGHT (C) 1975
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
;*
;*PROGRAM BY P. NELSON
;*
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-B1),AUG 29,1975.
;*

.SBTTL  OPERATIONAL SWITCH SETTINGS
;*
;*      SWITCH                  USE
;*      ------          --------------------
;*        15            HALT ON ERROR
;*        14            LOOP ON TEST
;*        13            INHIBIT ERROR TYPEOUTS
```

```
                                ;*          12          INHIBIT TRACE TRAP
                                ;*          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
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
                                .EQUIV  R6,SP           ;;STACK POINTER
                                .EQUIV  R7,PC           ;;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
```

```
                        SW01=   2
                        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

                        .SBTTL  TRAP CATCHER
```

```
                                ;*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
                                ;*****************************************************************

                                .SBTTL  ACT11 HOOKS
                                ;HOOKS REQUIRED BY ACT11
                                $SVPC=.                         ;SAVE PC
                                .=46
                                $ENDAD                          ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                                .=52
                                .WORD   0                       ;;2)SET LOC.52 TO ZERO
                                .=$SVPC                         ;; RESTORE PC
                                .=200
000200  000137  002266  START:  JMP     AUTO                    ;USE AUTO SELECTION OF UNITS
000204  000137  002320  MSTRT:  JMP     MANS                    ;ALLOW OPERATOR SELECTIONOF UNITS/TESTS
```

```
                                ;*****************************************************************

                                .SBTTL  COMMON TAGS

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

                    001100              .=1100
001100                          $CMTAG:                 ;;START OF COMMON TAGS
001100  000000      $PASS:      .WORD   0               ;;CONTAINS PASS COUNT
001102  000         $TSTNM:     .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103  000         $ERFLG:     .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000      $ICNT:      .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000      $LPADR:     .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000      $LPERR:     .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000      $ERTTL:     .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114  000         $ITEMB:     .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115  001         $ERMAX:     .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000      $ERRPC:     .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000      $GDADR:     .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000      $BDADR:     .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000      $GDDAT:     .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000      $BDDAT:     .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134  000000                  .WORD   0
001136  177570      SWR:        .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001140  177570      DISPLAY:    .WORD   DDISP           ;;ADDRESS OF DISPLAY REGISTER
001142  177560      $TKS:       177560                  ;;TTY KBD STATUS
001144  177562      $TKB:       177562                  ;;TTY KBD BUFFER
001146  177564      $TPS:       177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001150  177566      $TPB:       177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001152  000         $NULL:      .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001153  002         $FILLS:     .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001154  012         $FILLC:     .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001155  000         $TPFLG:     .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001156  000000      $TIMES:     0                       ;;MAX. NUMBER OF ITERATIONS
001160  000000      $ESCAPE:    0                       ;;ESCAPE ON ERROR ADDRESS
001162  177607  000377  $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001166  077         $QUES:      .ASCII  /?/             ;;QUESTION MARK
001167  015         $CRLF:      .ASCII  <15>            ;;CARRIAGE RETURN
001170  000012      $LF:        .ASCIZ  <12>            ;;LINE FEED
```

```
                                ;*****************************************************************

                                .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


001172                          $ERRTB:
                                        ;GENERAL ESCAPE SEQUENCE ERROR MESSAGE

001172  024460                          EM1                     ;AN ESCAPE SEQUENCE TO VT61   FAILED.
001174  024545                          DH1                     ;TEST#,ERROR PC,2 SEQUENCE BYTES,2 SEQUENCE BYTES.
001176  001422                          DT0
001200  001442                          DF0

                                        ;RECEIVE STATUS ERROR MESSAGE

001202  024610                          EM2                     ;RECEIVE STATUS ERROR
001204  024640                          DH2                     ;ADDRESS,STATUS ,ERR. BITS,CHAR.
001206  001452                          DT2
001210  001442                          DF0


                                ;RECIEVE SOFTWARE STATUS ERROR MESSAGE.

001212  024677                          EM3                     ;SOFTWARE (VSTAT) STATUS ERROR
001214  024740                          DH3                     ;PASS#,TEST#,GOOD STATUS,RECEIVED STATUS
001216  001500                          DT4
001220  001543                          DF6

                                ;DATA ERROR

001222  025007                          EM4                     ;DATA EXPECTED DOES NOT MATCH RECEIVE DATA.
001224  025053                          DH4                     ;TEST#,REC.CNT.,EXPECTED DATA, RECEIVE DATA
001226  001512                          DT5
001230  001442                          DF0

                                ;RECEIVE BYTE COUNT ERROR

001232  025122                          EM5                     ;BYTES EXPECTED DOES NOT EQUAL BYTES RECEIVED.
001234  025201                          DH5                     ;BYTES EXPECTED, BYTES RECEIVED
001236  001434                          DT1
001240  001450                          DF2

                                ;GENERAL DIRECT CURSOR ADDRESS FAILURE

001242  025232                          EM6                     ;CURSOR POSITION ERROR
```

```
 893  001244  025265                    DH6                     ;GD LINE, GD COL., BD LINE, BAD COL.
 894  001246  001452                    DT2
 895  001250  001474                    DF3
 896
 897                                     ;DIRECT CURSOR ADDRESS ERROR
 898
 899  001252  025332                    EM7                     ;DIRECT CURSOR ADDRESS ERROR
 900  001254  025432                    DH10                    ;PASS#,TEST#,BD. ROW,BD. COL.
 901  001256  001500                    DT4
 902  001260  001543                    DF6
 903
 904                                     ;LAST TEST-SELF TEST FAILED
 905
 906
 907  001262  025634                    EM10                    ;VT61 FAILED SELF-TEST FUNCTION
 908  001264  026211                    DH11                    ;CSR, VECTOR
 909  001266  001434                    DT1
 910  001270  001446                    DF!
 911
 912                                     ;VT61 FAIL/HUNG ERROR MESSAGE
 913  001272  025471                    EM11                    ;LAST TRANSMISSION TO VT61 CAUSED VT61 TO FAIL/HANG
 914  001274  025375                    DH7                     ;PASS#,TEST#,ERROR PC
 915  001276  001464                    DT3
 916  001300  001534                    DF4
 917
 918                                     ;GENERAL TEST FAILURE-PRECEEDS DATA/POSITION ERROR
 919
 920  001302  025557                    EM12                    ;VT61 UNDERR TEST FAILED-ERROR DATA FOLLOWS
 921  001304  025375                    DH7                     ;PASS#,TEST#,ERROR PC.
 922  001306  001464                    DT3
 923  001310  001534                    DF4
 924
 925                                     ;RECEIVE CHECKSUM ERROR
 926
 927  001312  026010                    EM13                    ;VT61 RECEIVER CHECKSUM ERROR
 928  001314  025675                    DH12                    ;PASS#,TEST#,GD.CKSUM,BD CKSUM
 929  001316  001500                    DT4
 930  001320  001543                    DF6
 931
 932                                     ;TRANSMITTER CHECKSUM ERRROR
 933
 934  001322  026057                    EM14                    ;VT61 TRANSMITTER CHECKSUM ERROR
 935  001324  025675                    DH12
 936  001326  001500                    DT4
 937  001330  001543                    DF6
 938
 939
 940
 941                                     ;XOFF FAILED TO HALT BLOCK XMIT
 942
 943  001332  026304                    EM15                    ;XOFF TO VT61 FAILED TO HALT BLOCK XMIT
 944  001334  027021                    DH13                    ;PASS,TEST,VSTAT
 945  001336  001524                    DT6
 946  001340  001534                    DF4
 947
 948                                     ;XON FAILED TO RESTART BLOCK XMIT
```

```
                                        EM16                    ;XON TO VT61 FAILED TO RESTART BLOCK XMIT
001342  026355                          DH13
001344  027021                          DT6
001346  001524                          DF4
001350  001534

                                        ;NO XON AFTER UNIT WAS RESET

001352  026430                          EM17                    ;NO XON AFTER UNIT WAS RESET.
001354  025375                          DH7                     ;PASS#,TEST#,ERROR PC
001356  001524                          DT6
001360  001534                          DF4

                                        ;PERIPHERAL ABORT ERROR

001362  026506                          EM20                    ;LAST PERIPHERAL OPERATION ABORTED.
001364  027053                          DH14                    ;PASS,TEST,ERROR PC, VSTAT
001366  001500                          DT4
001370  001543                          DF6

                                        ;CANT CLEAR PERIPHERAL ABORT FLAG.

001372  026552                          EM21                    ;COULD NOT CLEAR LAST ABORT FLAG.
001374  027053                          DH14
001376  001500                          DT4
001400  001543                          DF6

                                        ;MAINTENANCE MODE DID NOT FORCE A SOM/EOM.

001402  026615                          EM22                    ;SOM OR EOM NOT REC. IN MAINT. MODE.
001404  024740                          DH3                     ;PASS#,TEST#,EXP.STAT, ACT.STAT
001406  001500                          DT4
001410  001543                          DF6

                                        ;LINE FEED OR CURSOR RIGHT AT ROW 23 DID NOT CAUSE A SCROLL.

001412  026703                          EM23                    ;NO SCROLL FROM LINE FEED OR CURSOR RIGHT.
001414  025375                          DH7
001416  001524                          DT6
001420  001534                          DF4

001422  002264  001116  001124  DT0:    .WORD   TSTNM,$ERRPC,$GDDAT,$BDDAT,0
001430  001126  000000
001434  001124  001126  000000  DT1:    .WORD   $GDDAT,$BDDAT,0
001442     000     000     000  DF0:    .BYTE   0,0,0,0
001445     000
001446     000     000          DF1:    .BYTE   0,0

001450     001     001          DF2:    .BYTE   1,1             ;DECIMAL TYPE

001452  001120  001124  001122  DT2:    .WORD   $GDADR,$GDDAT,$BDADR,$BDDAT,0
001460  001126  000000
001464  001100  002264  001116  DT3:    .WORD   $PASS,TSTNM,$ERRPC,0
001472  000000
001474     001     001     001  DF3:    .BYTE   1,1,1,1
001477     001
```

```
1005  001500  001100  002264  001124  DT4:    .WORD   $PASS,TSTNM,$GDDAT,$BDDAT,0
1006  001506  001126  000000
1007  001512  002264  001120  001124  DT5:    .WORD   TSTNM,$GDADR,$GDDAT,$BDDAT,0
1008  001520  001126  000000
1009  001524  001100  002264  001120  DT6:    .WORD   $PASS,TSTNM,$GDADR,0
1010  001532  000000
1011  001534     001     000     000  DF4:    .BYTE   1,0,0
1012  001537     000     000     001  DF5:    .BYTE   0,0,1,1
1013  001542     001
1014  001543     001     000     000  DF6:    .BYTE   1,0,0,0
1015  001546     000
1016          001550                          .EVEN
1017                                          ;INSTRUCTION DEFINITIONS
1018          022626                  POP2SP  =22626
1019          024646                  PUSH2SP =24646
1020
1021                                  ;*****************************************************************
1022                                          ;DEFINITION SOFTWARE STATUS(VSTAT) REGISTER BITS
1023                                  ;*****************************************************************
1024
1025          100000                  RXOFF   =100000                 ;SET FOR XOFF, CLEARED FOR XON
1026          040000                  RSOM    =040000                 ;SET FOR SOM (START OF MESSAGE).
1027          020000                  REOM    =020000                 ;SET FOR EOM (END OF MESSAGE).
1028          010000                  PABRT   =010000                 ;SET FOR A PERIPHERAL ABORT.
1029          004000                  RSTT    =004000                 ;SET FOR RECEIVE STATUS ERROR.
1030          002000                  CKSUM   =002000                 ;SET TO CALCULATE 61 REC.  CHECKSUM
1031          001000                  EPL     =001000                 ;SET WHEN END OF LINE DETECTED
1032          000400                  ESC     =000400                 ;SET WHEN OCTAL 33 RECEIVED.
1033          000200                  XMKIL   =000200                 ;SET WHEN TRANSMIT KILLLED.
1034          000100                  TXSUM   =000100                 ;SET TO CALCULATE 61 XMIT CHECKSUM
1035          000040                  REVID   =000040                 ;SET WHEN REVERSE VIDEO MODE RECEIVED.
1036          000020                  COMGP   =000020                 ;SET TO CONVERT REC. CHAR. BY -137.
1037          000010                  ILLNE   =000010                 ;SET FOR REC. INT. ON NON-SELECTED LINE.
1038          000004                  CURPOS  =000004                 ;SET WHEN CURSOR POS. RECEIVED
1039          000002                  TRMID   =000002                 ;SET WHEN TERMINAL I.D. RECEVED.
1040          000001                  XMDNE   =000001                 ;SET UPON TRANSMIT COMPLETE
1041
1042                                  ;*****************************************************************
1043                                          ;DEFINITION OF DJ11 CONTROL BITS
1044                                  ;*****************************************************************
1045
1046          100000                  TRDY    =100000                 ;XMIT READY
1047          040000                  XENA    =040000                 ;XMIT INT. ENABLE
1048          000400                  XSCN    =000400                 ;XMIT SCAN ENABLE.
1049          000200                  RECDN   =000200                 ;RECEIVER DONE.
1050          000100                  RENA    =000100                 ;REC. INT. ENABLE.
1051          000020                  BCLR    =000020                 ;MOS CLEAR BUSY.
1052          000010                  MCLR    =000010                 ;MOS CLEAR.
1053          000004                  MAINT   =000004                 ;MAINTENANCE MODE.
1054          000001                  RSCN    =000001                 ;REC. SCAN ENABLE.
1055          040004                  TCOMB   =040004                 ;MAINT. MODE AND XMIT INT. ENABLE.
1056          000401                  SCAN    =000401                 ;REC. AND XMIT SCAN ENABLES.
1057          100000                  RRDY    =100000                 ;REG. 2, REC. BUFFER READY FLAG.
1058
1059          003600                  TOTCH   =1920.                  ;TOTAL CHARACTERS ON SCREEN
1060          003601                  TOTC1   =1921.                  ;TOTAL SCREEN +1
```

```
                                ;*****************************************************************
                                ;FOLLOWING ARE DJ11 ADDRESS, VECTOR AND LINE STORAGE TABLES
                                ;*****************************************************************
001550  000004                VVECT:  .BLKW   4               ;GOOD DJ11 VECTOR TABLE
001560  000004                DJTBL:  .BLKW   4               ;GOOD DJ11 ADDRESS TABLE
001570  000020                INTAB:  .BLKW   20              ;TABLE OF POSSIBLE DJ11 ADDRESSES
001630  000050                DJLNE:  .BLKW   40.             ;TABLE OF DJ11 LINESL

                                ;*****************************************************************
                                ;CURRENT POINTERS FOR ADDRESSES, VECTORS AND LINES
                                ;*****************************************************************
001750  000000                VECPT:  .WORD                   ;VECTOR INDEX
001752  000000                DJAPT:  .WORD                   ;ADDRESS INDEX
001754  000000                LNEPT:  .WORD           ;DJ11 LINE POINTER.
                                ;ADDRESS TABLES FOR DJ11 INTERFACES
001756  160010                STRTAB: .WORD   160010          ;BEGININNING OF FLOATING ADD.
001760  164000                ENDTAB: .WORD   164000          ;END OF FLOATING ADD.
                                ;*****************************************************************
                                ;VT61 ADDRESSES IN TABLE REFLECT UNIT UNDER TEST
                                ;*****************************************************************
001762  000000                VJCSR:  .WORD   0               ;DJ11 CONTROL AND STATUS.
001764  000000                VRBUF:  .WORD   0               ;RECEIVE DATA BUFFER
001766  000000                VXTCR:  .WORD   0               ;XMIT LINE CONTROL.
001770  000000                VXBUF:  .WORD   0               ;XMITTER DATA BUFFER
001772  000000                VECT:   .WORD   0               ;VECTOR FOR UNIT UNDER TEST
001774  000000                TSTLNE: .WORD   0               ;DJ11 LINE UNDER TEST.
001776  000000                OCTLNE: .WORD   0               ;OCT. EQUIV. OF TSTLNE (BIT8-11)
002000  000000                CRCSR:  .WORD   0               ;CONSOLE RECEIVE CSR
002002  000000                CRBUF:  .WORD   0               ;CONSOLE DATA BUFFER


                                ;*****************************************************************
                                ;TABLE OF VT61 COMMAND AND SEQUENCES
                                ;*****************************************************************

        000007                  .BEL    =007
002004  000007                BEL:    .WORD   007             ;BELL
        000015                  .CARRT  =015
002006  000015                CARRT:  .WORD   015             ;CARRIAGE RETURN
        000012                  .LNFED  =012
002010  000012                LNFED:  .WORD   012             ;LINE FEED
        000011                  .TAB    =011
002012  000011                TAB:    .WORD   011             ;TAB
                                ;*****************************************************************
002014  000001                        .WORD   01              ;TABLE DELIMITER (ESCN)
                                ;*****************************************************************

        000110                  .CHOM   =110
002016  000110                CHOM:   .WORD   110             ;HOME CURSOR  H

        000103                  .CRT    =103
002020  000103                CRT:    .WORD   103             ;CURSOR RIGHT  C

        000102                  .CDWN   =102
002022  000102                CDWN:   .WORD   102             ;CURSOR DOWN  B
```

```
1117            000104                  .CLFT   =104
1118  002024    000104                  CLFT:   .WORD   104             ;CURSOR LEFT  D
1119
1120            000101                  .CUP    =101
1121  002026    000101                  CUP:    .WORD   101             ;CURSOR UP  A
1122
1123            000112                  .EOS    =112
1124  002030    000112                  EOS:    .WORD   112             ;ERASE TO END OF SCREEN  J
1125
1126
1127                                    ;*****************************************************************
1128  002032    000002                          .WORD   2               ;TABLE DELIMITER (ESC0)
1129                                    ;*****************************************************************
1130
1131
1132            000101                  .EMAIN  =101
1133  002034    000101                  EMAIN:  .WORD   101             ;ENTER MAINTENANCE MODE  A
1134            000141                  .DMAIN  =141
1135  002036    000141                  DMAIN:  .WORD   141             ;EXIT MAINTENANCE MODE  SA
1136
1137            000105                  .LKKB   =105
1138  002040    000105                  LKKB:   .WORD   105             ;LOCK KEYBOARD  E
1139            000145                  .UNLKKB =145
1140  002042    000145                  UNLKKB: .WORD   145             ;UNLOCK KEYBOARD  SE
1141
1142            000103                  .DRECT  =103
1143  002044    000103                  DRECT:  .WORD   103             ;ENABLE LINEAR MODE  C
1144
1145            000133                  .CLRCK  =133
1146  002046    000133                  CLRCK:  .WORD   133             ;CLEAR RECEIVER CHECKSUM [
1147
1148            000134                  .CLTCK  =134
1149  002050    000134                  CLTCK:  .WORD   134             ;CLEAR TRANSMITTER CHECKSUM
1150
1151
1152            000112                  .EEMP   =112
1153  002052    000112                  EEMP:   .WORD   112             ;ENABLE REVERSE  VIDEO  J
1154            000152                  .DEMP   =152
1155  002054    000152                  DEMP:   .WORD   152             ;DISABLE REVERSE VIDEO  SJ
1156
1157            000137                  .IABT   =137
1158  002056    000137                  IABT:   .WORD   137             ;INITIALIZE ABORT FLAG  -
1159
1160                                    ;*****************************************************************
1161  002060    000003                          .WORD   3               ;TABLE DELIMITER (ESCAPE P)
1162                                    ;*****************************************************************
1163
1164            000131                  .EAPNT  =131
1165  002062    000131                  EAPNT:  .WORD   131             ;ENABLE AUTO PRINT MODE  Y
1166            000171                  .DAPNT  =171
1167  002064    000171                  DAPNT:  .WORD   171             ;DISABLE AUTO PRINT MODE  SY
1168
1169            000111                  .EINST  =111
1170  002066    000111                  EINST:  .WORD   111             ;ENABLE INSERT  I
1171            000151                  .ERPL   =151
1172  002070    000151                  ERPL:   .WORD   151             ;ENABLE REPLACE  SI
```

```
                                ;*****************************************************************
002072  000004                          .WORD   4               ;TABLE DELIMITER (I/O)
                                ;*****************************************************************

        054433                  .DCRAD  =054433
002074  054433                  DCRAD:  .WORD   054433          ;DIRECT CURSOR ADDRESSING
        067467                  .R23C79 =067467
002076  067467                  R23C79: .WORD   067467          ;CURSOR TO LOWER RIGHT
002100  000000                          .WORD   0

002102  047433                  RCUR:   .WORD   047433          ;DIRECT CURSOR ADDRESSING
        000131                  .Y      =131
        000131                  .RDCUR  =00131
002104  000131                  RDCUR:  .WORD   00131           ;READ CURSOR POSITION  Y
002106  000000                          .WORD   0

        000117                  .O      =117
002110  047433                  ESCO:   .WORD   047433          ;ESCAPE O
        000126                  .XMTAL  =000126
002112  000126                  XMTAL:  .WORD   000126          ;TRANSMIT ALL  V
002114  000000                          .WORD   0

002116  047433                          .WORD   047433          ;ESCAPE O
        000127                  .TCUCH  =127
002120  000127                  TCUCH:  .WORD   127             ;XMIT CHARACTER AT CURSOR.  W
002122  000000                          .WORD   0

002124  047433                          .WORD   047433          ;ESCAPE O
        000135                  .TXRCK  =135
002126  000135                  TXRCK:  .WORD   135             ;XMIT RECIEVER CHECKSUM ]
002130  000000                          .WORD   0

002132  047433                          .WORD   047433          ;ESCAPE O
        000136                  .TXTCK  =136
002134  000136                  TXTCK:  .WORD   136             ;XMIT TRANSMITTER CHECKSUM
002136  000000                          .WORD   0

002140  147433                          .WORD   147433          ;ESCAPE O
        000140                  .RABT   =140
002142  000140                  RABT:   .WORD   140             ;READ THE ABORT FLAG.  \
002144  000000                          .WORD   0

                                ;*****************************************************************
002146  177777                          .WORD   -1              ;END OF TABLE TERMINATOR
                                ;*****************************************************************

                                ;*****************************************************************
                                        ;PERIPHERAL COMMANDS
                                ;*****************************************************************

        000127                  .EPNT   =127
002150  000127                  EPNT:   .WORD   127             ;ENABLE PRINT MODE.  W
        000130                  .DPNT   =130
002152  000130                  DPNT:   .WORD   130             ;DISABLE PRINT MODE  X
```

```
1229
1230            000135                  .CPYSC  =135            ;COPY SCREEN  ]
1231            000136                  .ENAC   =136            ;ENABLE AUTO COPY MODE  ESC ↑
1232            000137                  .DISAC  =137            ;DISABLE AUTO COPY MODE  ESC -
1233            000150                  .PSCRN  =000150         ;PRINT THE SCREEN  H/SH
1234
1235
1236                                    ;*****************************************************************
1237                                            ;ESCAPE CODE EQUIVALENCES AND IDENTIFIERS
1238                                    ;*****************************************************************
1239
1240            000033                  .ESC    =033                            ;PRIMARY ESCAPE CODE.
1241            000120                  .P      =120
1242  002154    050033                  ESCP:   .WORD   050033          ;ESCAPE P
1243            000124                  .TSTER  =124
1244  002156    000124                  TSTER:  .WORD   124             ;TEST TERMINAL(ESC 0 T)
1245            002074                  ESCYI   =DCRAD                  ;ESCYI EQUALS DCRAD/DCRADI
1246            000057                  SLSH    =000057                 ;SLASH CODE FOR TERMINAL  IDENT ESC.
1247            000106                  CKGP    =106                    ;ENABLE REC.TO SUB 137 FROM ALL REC DATA
1248            000107                  NCKGP   =107                    ;ENABLE NORMAL RECEIVED DATA.
1249            000171                  CPABRT  =171                    ;COPIER ABORT
1250            000172                  PRABRT  =172                    ;PRINTER ABORT
1251            000170                  NABRT   =170                    ;NO ABORT  SX
1252  002160    000000                  IDENT:  .WORD   0               ;VT61 IDENT CODE
1253            002110                  ESCOI   =ESCO
1254            002154                  ESCPI   =ESCP
1255            002162                  ESCZI   =ESCZ
1256            055033                  .ESCZ   =055033
1257  002162    055033                  ESCZ:   .WORD   055033          ;OCTAL EQUIV. OF ESZ SEQUENCE
1258            000122                  .RESET  =122
1259  002164    000122                  RESET:  .WORD   122             ;VT61 INITIALIZE R
1260
1261  002166    000033                  ESCN:   .WORD   000033          ;ESCAPE N-FLAG
1262  002170    020041                  R01C00: .WORD   020041          ;ROW1,COL. 0
1263  002172    032041                  R01C20: .WORD   032041          ;ROW01,COLUMN 20
1264  002174    020066                  R22C00: .WORD   020066          ;ROW22,COL.00
1265  002176    020054                  R12C00: .WORD   020054          ;ROW 12,COLUMN 00
1266            020067                  .R23C00 =020067
1267  002200    020067                  R23C00: .WORD   020067          ;ROW23,COL.00
1268            025440                  .R00C11 =025440
1269  002202    025440                  R00C11: .WORD   025440          ;ROW,COL.11
1270            032040                  .R00C20 =032040
1271  002204    032040                  R00C20: .WORD   032040          ;ROW 0,COLUMN 20
1272  002206    024040                  R00C08: .WORD   024040          ;ROW 00,COLUMN 8
1273  002210    020040                  CUHME:  .WORD   020040          ;OCTAL EQUIV. OF CURSOR HOME.
1274  002212    067440                  R00C80: .WORD   067440          ;ROW 0, COLUMN 80.
1275  002214    067067                  R23C78: .WORD   067067          ;ROW 23,COL. 78.
1276            000040                  .R00    =40                     ;ROW 0
1277            000041                  .R01    =41                     ;ROW 1
1278            000054                  .R12    =54                     ;ROW 12
1279            000066                  .R22    =66                     ;ROW 22
1280            000067                  .R23    =67                     ;ROW 23
1281            000040                  .C00    =40                     ;COLUMN 0
1282            000043                  .C03    =43                     ;COL. 3
1283            000050                  .C08    =50                     ;COL. 8
1284            000053                  .C11    =53                     ;COL. 11
```

```
                    000064                  .C20    =64                     ;COL. 20
                    000065                  .C21    =65                     ;COL. 21
                    000110                  .C40    =110                    ;COL. 40
                    000157                  .C79    =157                    ;COL. 79

                                            ;*****************************************************************
                                                    ;TEMPORARY STORAGE LOCATIONS AND
                                                    ;SPECIAL RECEIVE CODE EQUIVALENCES.
                                            ;*****************************************************************
                    000002                  $OM     =02                     ;START OF MESSAGE
                    000004                  EOM     =04                     ;END OF MESSAGE
                    000023                  XOFF    =23                     ;TURN OFF TRANSMISSION
                    000021                  XON     =21                     ;TURN ON TRANSMISSION
002216  000000      CHRD:   .WORD   0       ;STORAGE FOR SINGLE CH. READ
002220  000000      SVER1:  .WORD           ;TEMP. STORAGE R1.
002222  000000      SVER2:  .WORD           ;TEMP. STORAGE R2.
002224  000000      ZERO:   .WORD   0       ;MUST BE LEFT AS ZERO.
002226  003000      TYP6:   .WORD   3000    ;TYPE 6 OCTAL CHAR-NO ZEROS
002230  000000      TSTPTR: .WORD   0       ;TEST POINTER IN MANUAL SELECT MODE
002232  000000      MODE:   .WORD   0       ;BYTE0=TESTING MODE,BYTE1=INTERFACE TYPE
002234  000000      FTLCNT: .WORD   0       ;COUNT OF INCOMPLETE XMITS.
002236  000012      ALWCNT: .WORD   10.     ;# OF ALLOWABLE INCOMPLETE XMITS.
002240  000001      ONE:    .WORD   1
002242  000000      TOADD:  .WORD
002244  000000      BUBCT:  .WORD
002246  000000      TPREG:  0
002250  000000      PRESC:  .WORD           ;PRIMARY ESC COMMAND
002252  000000      ESSEQ:  .WORD           ;SEQUENCE ASSEMBLY AREA
002254  000000      DLAY:   .WORD
002256  000000      ROSVE:  .WORD           ;TEMP STORAGE FOR R0 ONLY.
002260  000000      VSTAT:  .WORD   0
002262  000000      BLKM:   .WORD   0       ;FLAG LOCATION FOR BLOCK MODE XMITS.
002264  000000      TSTNM:  .WORD   0       ;DISPLAY STORAGE FOR TEST NUMBER.


                                            ;*****************************************************************
                                                    ;AUTOMATIC SELECTION OF UNITS. TESTS 1 THROUGH 33 WILL BE
                                                    ;REPITIVELY EXECUTED FOR ALL UNITS.
                                            ;*****************************************************************

002266  005037  002232  AUTO:   CLR     MODE            ;ZERO THE MODE SWITCH
002272  000137  012154          JMP     SETA            ;;DO VECTOR SETUP
002276  004037  012670  AUTOA:  JSR     R0,TRPVEC       ;GO FIND GOOD DJ11S
002302  004037  012776          JSR     R0,CDEV         ;CHECK DJ11S FOUND
002306  004037  013536          JSR     R0,INITA        ;INSURE VT61S ON DJ11
002312  000137  002626          JMP     MODCK           ;VT61 PRESENT -BEGIN TESTING
002316  000767                  BR      AUTOA           ;NO VT61 FOUND LOOP IN CHECKING


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

                                                    ;MANUAL UNIT AND TEST SELECTION.  UNITS CAN BE
                                                    ;SELECTED VIA CONSOLE OR AUTO SELECTION CAN
                                                    ;BE UTILIZED.  TESTS ENTERED VIA CONSOLE WILL
                                                    ;BE EXECUTED IN THE ORDER ENTERED.

                                            ;*****************************************************************
```

```
1341
1342  002320  012737  000001  002232  MANS:   MOV     #1,MODE         ;SET MODE TO MANUAL SELECT.
1343  002326  000137  012154          JMP     SETA            ;GO SET UP CONSTANTS
1344  002332  104400  024233  MANSA:  TYPE    ,DMANA
1345  002336  004037  012670          JSR     R0,TRPVEC       ;FIND GOOD DJ11'S
1346  002342  012703  001570          MOV     #INTAB,R3
1347  002346  005002          BLDADD: CLR     R2
1348
1349  002350  004037  020570  BLDADA: JSR     R0,GTNUM        ;GET A KEYBOARD INPUT
1350  002354  120127  000054          CMPB    R1,#54          ;CHAR. = COMMA?
1351  002360  001004                  BNE     1$              ;NO
1352  002362  004037  012624          JSR     R0,TMNAD        ;YES-VERIFY THIS ADDRESS,
1353  002366  010223                  MOV     R2,(R3)+        ;STORE THIS ADDRESS
1354  002370  000766                  BR      BLDADD          ;AND LOOK FOR ANOTHER ADDRESS.
1355  002372  120137  002010  1$:     CMPB    R1,LNFED        ;CHAR. = LINE FEED?
1356  002376  001025                  BNE     3$              ;NO
1357  002400  005702                  TST     R2              ;ANY ENTRIES CREATED?
1358  002402  001414                  BEQ     2$              ;NO USE AUTO SELECTION OF UNITS
1359  002404  004037  012624          JSR     R0,TMNAD        ;YES-VERIFY THIS ADDRESS,
1360  002410  010223                  MOV     R2,(R3)+        ;STORE LAST ADDRESS,
1361  002412  013723  002224          MOV     ZERO,(R3)+      ;AND SET A TERMINATOR IN TABLE.
1362  002416  004037  012776          JSR     R0,CDEV         ;CHECK DJ11 ON VT 61 SELECTED
1363  002422  005737  001560          TST     DJTBL           ;ANY DJ11S GOOD?
1364  002426  001741                  BEQ     MANSA           ;NO-BACK TO SQUARE ONE
1365  002430  000137  002460          JMP     BLDLNE          ;YES- GO GET TESTS
1366  002434  004037  012776  2$:     JSR     R0,CDEV         ;CHECK DJ11'S
1367  002440  004037  013536          JSR     R0,INITA        ;VERIFY DJ11 HAVE VT61 ATTACHED
1368  002444  000137  002460          JMP     BLDLNE          ;BEGIN LINE SELECTION
1369  002450  000730                  BR      MANSA           ;NO UNIT FOUND-LOOP
1370  002452  004037  020466  3$:     JSR     R0,OCTBIN       ;KEEP BUILDING ADDRESS
1371  002456  000734                  BR      BLDADA
1372  002460  104400  024363  BLDLNE: TYPE    ,DMANL          ;TYPE ENTER LINES MESSAGE.
1373  002464  012703  001630          MOV     #DJLNE,R3       ;SET FIRST LINE ADDRESS.
1374  002470  005002          BLDLNA: CLR     R2
1375  002472  004037  020570  10$:    JSR     R0,GTNUM        ;GET A KEYBOARD INPUT
1376  002476  120127  000054          CMPB    R1,#54          ;CHAR. = COMMA?
1377  002502  001002                  BNE     1$              ;NO
1378  002504  010223                  MOV     R2,(R3)+        ;YES - STORE THIS ADDRESS
1379  002506  000770                  BR      BLDLNA          ;AND LOOK FOR ANOTHER LINE ENTRY.
1380  002510  120137  002010  1$:     CMPB    R1,LNFED        ;CHAR. = LINE FEED?
1381  002514  001403                  BEQ     2$              ;YES-SET TERMINATIORS AND EXIT.
1382  002516  004037  020466          JSR     R0,OCTBIN       ;NO-KEEP BUILDING ADDRESS
1383  002522  000763                  BR      10$
1384  002524  010223  2$:     MOV     R2,(R3)+        ;STORE LAST ADDRESS.
1385  002526  012723  177777          MOV     #-1,(R3)+       ;STORE END OF ADD. TERMINATOR.
1386  002532  005013                  CLR     (R3)            ;STORE LAST LINE TERMINATOR
1387
1388  002534  104400  024333  BLDTST: TYPE    ,DMANB          ;TYPE 2ND PART OF MANUAL MESSAGE
1389  002540  012703  001570          MOV     #INTAB,R3       ;USE INTAB AS TEST # STORAGE.
1390  002544  005004                  CLR     R4              ;CLEAR TEST COUNTER
1391  002546  005002  11$:     CLR     R2              ;CLEAR ASSEMBL WORD
1392  002550  004037  020570  10$:    JSR     R0,GTNUM        ;GET A NUMERIC CHAR.
1393  002554  120127  000054          CMPB    R1,#54          ;CHAR.=COMMA?
1394  002560  001006                  BNE     1$              ;NO
1395  002562  110223                  MOVB    R2,(R3)+        ;YES STORE A TEST #
1396  002564  005204                  INC     R4              ;AND INCREMENT TEST COUNT.
```

```
1397  002566  020437  000040          CMP     R4,32.          ;COUNT =32?
1398  002572  001415                  BEQ     MODCK           ;YES ACCEPT NO MORE ENTRIES.
1399  002574  000764                  BR      11$             ;NO KEEP LOOKING
1400  002576  120137  002030  1$:     CMPB    R1,LNFED        ;CHAR. = LINE FEED?
1401  002602  001006                  BNE     2$              ;NO
1402  002604  110223                  MOVB    R2,(R3)+        ;LOAD THE LAST TEST
1403  002606  105013                  CLRB    (R3)            ;AND INSERT TEST TABLE TERMINATOR
1404  002610  112737  000001  002232  MOVB    #1,MODE         ;SET MODE SWITCH TO MANUAL
1405  002616  000403                  BR      MODCK           ;AND BEGIN TESTING.
1406
1407  002620  004037  020466  2$:     JSR     R0,OCTBIN       ;CONVERT CHAR.
1408  002624  000751                  BR      10$
1409
1410                          ;*****************************************************************
1411                          ;THIS ROUTINE LOOKS FOR THE OPERATIONAL MODE REQUESTED AND
1412                          ;SELECTS THE NEXT UNIT TO BE TESTED.
1413
1414                          ;MODE 0 = ACCEPTANCE TYPE TEST
1415                          ;MODE 1 = OPERATOR SELECTION OF UNITS AND SEQUENCE OF TESTS.
1416                          ;*****************************************************************
1417
1418  002626  012737  001560  001752  MODCK:  MOV     #DJTBL,DJAPT    ;INITIAL SETUP OF ADDRESS
1419  002634  012737  001550  001750          MOV     #VVECT,VECPT    ;AND VECTOR POINTERS.
1420  002642  012737  001630  001754          MOV     #DJLNE,LNEPT    ;LOAD LINE POINTER.
1421  002650  012701  001762          MODCO:  MOV     #VJCSR,R1       ;LOAD ADDRESS DESTINATION
1422  002654  013702  001752                  MOV     DJAPT,R2        ;LOAD CURRENT ADDRESS POINTER
1423  002660  017703  177064                  MOV     @VECPT,R3       ;LOAD CURRENT VECTOR POINTER
1424  002664  005712                  TST     (R2)            ;ALL UNITS CHECKED?
1425  002666  001013                  BNE     1$              ;NO - CONTINUE
1426  002670  005737  002232          TST     MODE            ;CHECK MODE
1427  002674  001002                  BNE     10$
1428  002676  000137  002276          JMP     AUTOA           ;GO RESTART AUTO MODE
1429  002702  105777  177322  10$:     TSTB    @TSTPTR         ;MANUAL LOOP REQUESTED?
1430  002706  100001                  BPL     2$              ;NO
1431  002710  000746                  BR      MODCK           ;YES-RESTART COMPLETE TEST.
1432  002712  000137  002332  2$:     JMP     MANSA           ;GO RESTART MANUAL MODE
1433  002716  004037  014076  1$:     JSR     R0,LDADD        ;NO-LOAD NEXT ADDRESSES
1434  002722  010237  001752          MOV     R2,DJAPT        ;SAVE ADDRESS POINTER.
1435  002726  010337  001772          MOV     R3,VECT         ;STORE VECT. OF UNIT UNDER TEST
1436  002732  012723  015040          MOV     #INTRC,(R3)+    ;YES - NOW SET UP RECEIVE VECTOR
1437  002736  012723  000340          MOV     #340,(R3)+      ;AND SET RECEIVER PSW TO 7
1438  002742  012723  016014          MOV     #INTXM,(R3)+    ;SET UP TRANSMIT VECTOR
1439  002746  012723  000340          MOV     #340,(R3)+      ;AND SET PSW TO 7.
1440  002752  017737  176776  001774  MODCA:  MOV     @LNEPT,TSTLNE   ;LOAD LINE TO UTILIZED.
1441  002760  023727  001774  177777          CMP     TSTLNE,#-1      ;THIS LINE REALLY A SEPARATOR?
1442  002766  001007                  BNE     12$             ;NO-TEST IT.
1443  002770  062737  000002  001754          ADD     #2,LNEPT        ;YES-BUMP LINE POINTER ,UPDATE VECTOR
1444  002776  062737  000002  001750          ADD     #2,VECPT        ; POINTER AND GET NEXT ADDRESS.
1445  003004  000721                  BR      MODCO
1446  003006  005046                  12$:    CLR     -(SP)           ;CLEAR THE PSW,LSI11 STYLE.
1447  003010  012746  003016          MOV     #100$,-(SP)
1448  003014  000002                  RTI
1449  003016  012737  031617  015756  100$:   MOV     #RCRLB+477,REBUF        ;SET UP END OF BUFFER
1450  003024  012737  032317  016264          MOV     #TCRLB+477,TEBUF
1451  003032  012737  031120  015754          MOV     #RCRLB,RBBUF    ;INITIIALIZE REC.BUFFER.
1452  003040  012737  031620  016262          MOV     #TCRLB,TBBUF    ;INITIALIZE TRANSMIT BUFFER.
```

```
1453  003046  004037  017262                    JSR     R0,RESPTR       ;RESET INTERRUPT POINTERS.
1454  003052  005037  002262                    CLR     BLKM            ;CLEAR BLOCK MODE FLAG.
1455  003056  005037  002264                    CLR     TSTNM           ;CLEAR CURRENT TEST LOCATION.
1456  003062  005037  021652                    CLR     XMZER           ;CLEAR ZERO TRANSMIT FLAG
1457  003066  005037  002260                    CLR     VSTAT           ;CLEAR ALL INTERRUPT FLAGS
1458  003072  004037  020510                    JSR     R0,CONVLN       ;CONVERT BINARY LINE # TO ,OCTAL.
1459  003076  052777  000010  176656            BIS     #MCLR,@VJCSR    ;CLEAR SILO AND UARTS.
1460  003104  000240                            NOP
1461  003106  013777  001774  176652            MOV     TSTLNE,@VXTCR   ;LOAD THE XMITTER LINE #.
1462  003114  052777  000401  176640            BIS     #SCAN,@VJCSR    ;ALLOW REC. AND XMIT. SCANS.
1463  003122  004037  016452                    JSR     R0,ZFLAG        ;ISSUE ESC Z TO VT61
1464  003126  012637  002160                    MOV     (SP)+,IDENT     ;;POP STACK INTO IDENT
1465  003132  100002                            BPL     11$             ;IF IDENT IS -1,CLEAR IT.
1466  003134  005037  002160                    CLR     IDENT
1467  003140                          11$:
1468  003140  012637  002216                    MOV     (SP)+,CHRD      ;;POP STACK INTO CHRD
1469  003144  001375                            BNE     11$
1470  003146  105037  002161                    CLRB    IDENT+1         ;CLEAR ALL BUT IDENT BITS.
1471  003152  104400  001167                    TYPE    ,$CRLF
1472  003156  104400  026132                    TYPE    ,DVUNIT         ;ISSUE UNIT UNDER TEST MESSAGE
1473  003162  013746  001762                    MOV     VJCSR,-(SP)     ;;SAVE VJCSR FOR TYPEOUT
1474                                                                    ;;TYPE THE ADDRESS
1475  003166  104402                            TYPOS                   ;;GO TYPE--OCTAL ASCII
1476  003170     006                            .BYTE   6               ;;TYPE 6 DIGIT(S)
1477  003171     001                            .BYTE   1               ;;TYPE LEADING ZEROS
1478  003172  017746  176552                    MOV     @VECPT,-(SP)    ;;SAVE @VECPT FOR TYPEOUT
1479                                                                    ;;TYPE THE VECTOR
1480  003176  104402                            TYPOS                   ;;GO TYPE--OCTAL ASCII
1481  003200     006                            .BYTE   6               ;;TYPE 6 DIGIT(S)
1482  003201     000                            .BYTE   0               ;;SUPPRESS LEADING ZEROS
1483  003202  013737  001776  002216            MOV     OCTLNE,CHRD
1484  003210  000337  002216                    SWAB    CHRD
1485  003214  013746  002216                    MOV     CHRD,-(SP)      ;;SAVE CHRD FOR TYPEOUT
1486                                                                    ;;TYPE THE LINE
1487  003220  104402                            TYPOS                   ;;GO TYPE--OCTAL ASCII
1488  003222     006                            .BYTE   6               ;;TYPE 6 DIGIT(S)
1489  003223     000                            .BYTE   0               ;;SUPPRESS LEADING ZEROS
1490  003224  013746  002160                    MOV     IDENT,-(SP)     ;;SAVE IDENT FOR TYPEOUT
1491                                                                    ;;TYPE THE IDENT
1492  003230  104402                            TYPOS                   ;;GO TYPE--OCTAL ASCII
1493  003232     006                            .BYTE   6               ;;TYPE 6 DIGITS
1494  003233     000                            .BYTE   0               ;;SUPPRESS LEADING ZEROS
1495  003234  104400  001167                    TYPE    ,$CRLF          ;CARRIAGE RETURN AND LINE FEED
1496  003240  032737  000001  002160            BIT     #BIT00,IDENT    ;UNIT HAVE A COPIER?
1497  003246  001402                            BEQ     20$             ;NO
1498  003250  104400  026257                    TYPE    ,DCOPYR         ;YES-ISSUE COPIER MESSAGE
1499  003254  032737  000002  002160  20$:     BIT     #BIT01,IDENT    ;UNIT HAVE A PRINTER?
1500  003262  001402                            BEQ     21$             ;NO
1501  003264  104400  026231                    TYPE    ,DPRTR          ;YES-ISSUE PRINTER MESSAGE.
1502  003270  005037  002234            21$:    CLR     FTLCNT          ;CLEAR COUNT OF FATAL XMITS.
1503  003274  062737  000002  001754            ADD     #2,LNEPT        ;UPDATE LINE POINTER.
1504  003302  012737  032322  032320            MOV     #ABBUF,ABUFP    ;RESET THE REC. DATA POINTER
1505  003310  052777  000100  176444            BIS     #RENA,@VJCSR    ;SET THE REC. INT. ENABLE  FOR TESTS
1506  003316  105737  002232                    TSTB    MODE            ;CHECK TESTING MODE
1507  003322  001403                            BEQ     ASTRT           ;AUTO MODE
1508  003324  012737  001570  002230            MOV     #INTAB,TSTPTR   ;LOAD THE INITIAL TEST NUMBER
```

```
1509
1510                                    ;*****************************************************************
1511                                    ;*****************************************************************
1512                                            ;THIS TEST ISSUES ALL ESCAPE SEQUENCES AND
1513                                            ;INSURES THE VT61 HAS NOT FAILED DURING AN
1514                                            ;ESC SEQUENCE BY ISSUING A ESC Z TO FORCE A
1515                                            ;VT61 RESPONSE.  THE PURPOSE OF THE TEST IS TO ATTEMPT TO
1516                                            ;INSURE THAT SUBSEQUENT TESTS WILL NOT RESULT IN
1517                                            ;A "HUNG" UNIT.  DATA IS NOT EVALUATED.
1518                                    ;*****************************************************************
1519
1520  003332                            ASTRT:
1521                                    ;*****************************************************************
1522  003332  000004                    TST1:   SCOPE
1523  003334  012737  000001  001156            MOV     #1,$TIMES       ;;DO 1 ITERATION
1524  003342  012737  003350  001106            MOV     #ESTST,$LPADR   ;;SET SCOPE LOOP ADDRESS
1525
1526  003350  012701  002004            ESTST:  MOV     #BEL,R1         ;POINT TO FIRST COMMAND
1527  003354  042777  000100  176400            BIC     #RENA,@VJCSR    ;CLEAR REC. INT. ENABLE
1528  003362  113737  001102  002264            MOVB    $TSTNM,TSTNM    ;LOAD THE TEST NUMBER.
1529  003370  005037  002250                    CLR     PRESC
1530  003374  005004                            CLR     R4
1531  003376                            ZERST:
1532  003376  013746  002224                    MOV     ZERO,-(SP)      ;;PUSH ZERO ON STACK
1533  003402  012702  002250                    MOV     #PRESC,R2       ;SET UP SEQNENCE ADDRESS
1534  003406  012103                    GCMD:   MOV     (R1)+,R3        ;LOAD THE COMMAND
1535  003410  001405                            BEQ     1$              ;IF CHAR. ZERO,MUST BE XMIT TERMINATOR
1536  003412  100537                            BMI     ESTEX           ;TABLE EXPENDED - EXIT TEST.
1537  003414  120327  000004                    CMPB    R3,#4           ;IS COMMAND ACTUALLY A DELIMITER?
1538  003420  103444                            BLO     DELIM           ;YES, GO UPDATE FUNCTIONS
1539  003422  001473                            BEQ     SPTN            ;NO, ITS A "10" - SPECIAL CASE.
1540  003424  005704                    1$:     TST     R4              ;SEE IF FLAG INDICATING SEQ.
1541  003426  100474                            BMI     SEQ4            ;4 IS SET. - YES EXIT
1542  003430  010337  002252            2$:     MOV     R3,ESSEQ        ;PUSH THE SEQUENCE TO BE TESTED
1543  003434                            INXMT:
1544  003434  013746  002252                    MOV     ESSEQ,-(SP)     ;;PUSH ESSEQ ON STACK
1545  003440  005704                            TST     R4              ;DOES THIS SEQUENCE REQUIRE
1546  003442  001402                            BEQ     3$              ;ADDITIONAL ESC?
1547  003444  013746  002250                    MOV     PRESC,-(SP)     ;;PUSH PRESC ON STACK
1548
1549  003450  004037  014370            3$:     JSR     R0,TESC         ;GO TRANSMIT THIS SEQUENCE.
1550
1551  003454  005704                    4$:     TST     R4              ;IN I/O SEQUENCES?
1552  003456  100011                            BPL     40$             ;NO
1553  003460  012737  000054  020220            MOV     #44.,DCOUNT     ;YES,SET UP TO DELAY 1+ SEC.
1554  003466  004037  020156                    JSR     R0,DELAY
1555  003472  032777  100000  176264            BIT     #RRDY,@VRBUF    ;CLEAR THE SILO
1556  003500  001374                            BNE     .-6
1557
1558  003502  004037  016452            40$:    JSR     R0,ZFLAG        ;ISSUE ESC Z SEQUENCE-GET IDENT
1559  003506                            5$:
1560  003506  012637  002216                    MOV     (SP)+,CHRD      ;;POP STACK INTO CHRD
1561  003512  123737  002216  002160            CMPB    CHRD,IDENT      ;HAVE WE POPPED THE IDENT?
1562  003520  001045                            BNE     T1ERR           ;NO-ERROR CONDITION
1563  003522                            POPIT:
1564  003522  012637  002216                    MOV     (SP)+,CHRD      ;;POP STACK INTO CHRD
```

```
1565  003526  001375                          BNE     .-4
1566  003530  000722                          BR      ZERST           ;GET NEXT COMMAND
1567
1568  003532  120327  000001        DELIM:    CMPB    R3,#1
1569  003536  001407                          BEQ     1$              ;FIRST DELIMITER - SET ESCN
1570
1571  003540  120327  000002                  CMPB    R3,#BIT01       ;SECOND DELIMITER - SET ESCO
1572  003544  001412                          BEQ     2$
1573
1574  003546  120327  000003                  CMPB    R3,#3           ;THIRD DELIMITER - SET ESCP
1575  003552  001413                          BEQ     3$
1576  003554  000714                          BR      GCMD            ;INVALID CHARACTER - GET ANOTHER
1577
1578  003556  012704  000001        1$:       MOV     #1,R4           ;SET FIRST DELIMITER FLAG.
1579  003562  013737  002166  002250          MOV     ESCN,PRESC      ;INSERT ESCN.
1580  003570  000706                          BR      GCMD
1581
1582  003572  013737  002110  002250  2$:     MOV     ESCO,@#PRESC    ;INSERT ESCO
1583  003600  000702                          BR      GCMD
1584
1585  003602  013737  002154  002250  3$:     MOV     ESCP,@#PRESC    ;INSERT ESCP
1586  003610  000676                          BR      GCMD
1587
1588  003612  012704  177777        SPTN:     MOV     #-1,R4          ;SET FLAG INDICATING I/O
1589  003616  000673                          BR      GCMD            ;SEQUENCES.
1590
1591  003620  005703                SEQ4:     TST     R3              ;CHECK IF COMMAND = 0
1592  003622  001704                          BEQ     INXMT           ;YES, COMPLETE SEQUENCE ASSEMBLED
1593  003624  110322                          MOVB    R3,(R2)+        ;NO - KEEP ASSEMBLING
1594  003626  000303                          SWAB    R3              ;POSITION HIGH  ORDER BIT
1595  003630  110322                          MOVB    R3,(R2)+        ;AND ASSEMBLE IT
1596  003632  000665                          BR      GCMD            ;GET ANOTHER BYTE
1597
1598  003634  004037  016654        T1ERR:    JSR     R0,CLREG
1599  003640  013737  002250  001124          MOV     PRESC,$GDDAT    ;AND INSERT IN ERROR
1600  003646  000337  001124                  SWAB    $GDDAT          ;REASSEMBLE FAILING SEQUENCES
1601  003652  013737  002252  001126          MOV     ESSEQ,$BDDAT
1602  003660  105737  002253                  TSTB    ESSEQ+1         ;IF UPPER BYTE IS CLEAR DO NOT SWAP
1603  003664  001402                          BEQ     1$
1604  003666  000337  001126                  SWAB    $BDDAT          ;MESSAGE 1
1605  003672  104001                1$:       ERROR   1               ;ISSUE ERROR MESSAGE
1606  003674  005237  002234                  INC     FTLCNT          ;INCREMENT FATAL XMIT COUNT.
1607  003700  023737  002234  002236          CMP     FTLCNT,ALWCNT   ;FATAL XMITS EXCEEDED ALLOWED?
1608  003706  103003                          BHIS    FTEX1           ;YES-EXIT.
1609  003710  000704                          BR      POPIT           ;CLEAR THE STACK AND TRY ANOTHER COMMAND
1610  003712                        ESTEX:
1611  003712  012637  002216                  MOV     (SP)+,CHRD      ;;POP STACK INTO CHRD
1612  003716  052777  000100  176036  FTEX1:  BIS     #RENA,@VJCSR    ;SET THE REC. INT. ENABLE FOR TESTS
1613
1614                                ;*****************************************************************
1615                                        ;ROUTINE TO INSURE ENTERING MAINTENANCE MODE CAUSES SOM AND
1616                                        ;EOM TO BE APPENDED TO ALL TRANSMITS FROM VT61 UNDER TEST.
1617                                        ;MAINTENANCE MODE IS ENTERED , THEN AN ESCAPE Z SEQUNECE
1618                                        ;IS ISSUED TO THE UNIT AND THE RESULTING TRANSMISSION IS
1619                                        ;CHECKED OF SOM/EOM.
1620                                ;*****************************************************************
```

```
1621
1622                                    ;*****************************************************************
1623  003724  000004                    TST2:   SCOPE
1624  003726  012737  000005  001156            MOV     #5,$TIMES       ;;DO 5 ITERATIONS
1625  003734  012737  003742  001106            MOV     #CKMNT,$LPADR   ;;SET SCOPE LOOP ADDRESS
1626
1627  003742  004037  016272            CKMNT:  JSR     R0,RESETV       ;RESET THE UNIT AND SETMAINT. MODE.
1628  003746  112777  000002  012312            MOVB    #SOM,@TBUFP     ;XMIT THE START OF MESSAGE.
1629  003754  004037  017152                    JSR     R0,XMIT1
1630  003760  113777  002162  012300            MOVB    ESCZ,@TBUFP
1631  003766  004037  017152                    JSR     R0,XMIT1        ;SEND AN IDENT REQUEST.
1632  003772  113777  002163  012266            MOVB    ESCZ+1,@TBUFP
1633  004000  004037  017152                    JSR     R0,XMIT1
1634  004004  112777  000004  012254            MOVB    #EOM,@TBUFP     ;XMIT END OF MESSAGE.
1635  004012  004037  017152                    JSR     R0,XMIT1
1636  004016  005037  002254                    CLR     DLAY            ;SET UP SOM DELAY OF 100M.S.
1637  004022  032737  040000  002260  1$:       BIT     #RSOM,VSTAT     ;RECEIVED THE START OF MESSAGE?
1638  004030  001003                            BNE     CKEOM           ;YES-GO LOOK FOR EOM.
1639  004032  005337  002254                    DEC     DLAY            ;NO-RUN TIMEOUT DELAY
1640  004036  001371                            BNE     1$              ;AND KEEP LOOKING.
1641
1642  004040  012701  000062            CKEOM:  MOV     #50.,R1         ;SET MAX DELAY FOR 500 M.S.
1643  004044  032737  020000  002260  1$:       BIT     #REOM,VSTAT     ;REECIVED END OF MESSAGE?
1644  004052  001007                            BNE     10$             ;YES-CHECK FOR BOTH RECEIVED.
1645  004054  012737  000001  020220            MOV     #1,DCOUNT       ;DELAY FOR 10 M..S.
1646  004062  004037  020156                    JSR     R0,DELAY
1647  004066  005301                            DEC     R1              ;AND KEEP LOOKING.
1648  004070  001365                            BNE     1$
1649  004072  032737  040000  002260  10$:      BIT     #RSOM,VSTAT     ;RECEIVED SOM?
1650  004100  001404                            BEQ     2$              ;NO ISSUE ERROR
1651  004102  032737  020000  002260            BIT     #REOM,VSTAT     ;RECEIVED EOM?
1652  004110  001007                            BNE     EXMNT           ;YES, NO ERRORS-EXIT.
1653  004112  012737  006001  001124  2$:       MOV     #6001,$GDDAT    ;LOAD ERROR WITH EXPECTED
1654  004120  013737  002260  001126            MOV     VSTAT,$BDDAT    ;AND ACTUAL STATUS.
1655  004126  104022                            ERROR   22
1656
1657  004130  000240                    EXMNT:  NOP
1658                                    ;*****************************************************************
1659                                            ;THIS TEST INSURES THAT THE CURSOR WILL RESPOND
1660                                            ;TO DIRECT CUSROR ADDRESSING  THE UNIT IS RESET AND THE CURSOR
1661                                            ;POSITION IS VERIFIED TO BE HOME . THE CURSOR IS THEN MOVED
1662                                            ;TO POSITION ROW 23 COLUMN 80 AND THE POSITION IS AGAIN
1663                                            ;VERIFIED. ERRORS ARE REPORTED IF THE POSITIONS ARE INCORRECT.
1664
1665                                    ;*****************************************************************
1666
1667
1668                                    ;*****************************************************************
1669  004132  000004                    TST3:   SCOPE
1670  004134  012737  000005  001156            MOV     #5,$TIMES       ;;DO 5 ITERATIONS
1671  004142  012737  004150  001106            MOV     #CURS1,$LPADR   ;;SET SCOPE LOOP ADDRESS
1672
1673  004150  013701  016262            CURS1:  MOV     TBBUF,R1        ;USE R1 AS XMIT BUFFER POINTER.
1674  004154  004037  016272                    JSR     R0,RESETV       ;RESET THE UNIT AND WAIT FOR XON.
1675  004160  013721  002110                    MOV     ESCOI,(R1)+     ;CLFT. RESET, READ CURSOR
1676  004164  113721  002104                    MOVB    RDCUR,(R1)+     ;POSITION, CURSOR LEFT.
```

```
1677  004170  012737  000003  016270          MOV     #3,XMCNT        ;XMIT 3 BYTES
1678
1679  004176  004037  016676                  JSR     R0,XMREC        ;XMIT AND RECEIVE.
1680  004202  000402                          BR      10$             ;NORMAL EXIT.
1681  004204  104011                          ERROR   11              ;TRANSMISSION CAUSED VT61 TO FAIL/HANG
1682  004206  000446                          BR      2$              ;EXIT TEST.
1683  004210  013701  031120          10$:    MOV     RCRLB,R1        ;GET THE CURRENT CURSOR POSITION.
1684  004214  020137  002210                  CMP     R1,CUHME        ;CURSOR REALLY HOME?
1685  004220  001405                          BEQ     1$              ;YES EXIT
1686  004222  104012                          ERROR   12              ;VT61 FAILURE MESSAGE
1687  004224  013746  002210                  MOV     CUHME,-(SP)     ;;PUSH CUHME ON STACK
1688  004230  004037  017342                  JSR     R0,CURER        ;GO LOAD AND ISSUE CURSOR ERROR
1689
1690  004234  013701  016262          1$:     MOV     TBBUF,R1        ;LOAD XMIT BUFFER WITH
1691  004240  013721  002074                  MOV     DCRAD,(R1)+
1692  004244  013721  002076                  MOV     R23C79,(R1)+    ;CURSOR TO ROW 23,COL.79
1693  004250  013721  002110                  MOV     ESCOI,(R1)+     ;READ CURSOR POSITION
1694  004254  013721  002104                  MOV     RDCUR,(R1)+     ;IT AND CURSOR RIGHT
1695  004260  012737  000007  016270          MOV     #7,XMCNT        ;XMIT 7 BYTES.
1696  004266  004037  016676                  JSR     R0,XMREC        ;XMIT AND RECEIVE
1697  004272  000402                          BR      20$             ;NORMAL EXIT.
1698  004274  104011                          ERROR   11              ;TRANSMISSION CAUSED VT61 TO FAIL/HANG
1699  004276  000412                          BR      2$              ;EXIT TEST.
1700  004300  012701  031120          20$:    MOV     #RCRLB,R1
1701
1702  004304  023711  002076                  CMP     R23C79,(R1)     ;CHECK CURSOR POSITION TO LOWER RT.
1703  004310  001405                          BEQ     2$              ;OK, EXIT
1704  004312  104012                          ERROR   12              ;VT61 FAILURE MESSAGE
1705  004314  013746  002076                  MOV     R23C79,-(SP)    ;;PUSH R23C79 ON STACK
1706  004320  004037  017342                  JSR     R0,CURER        ;LOAD AND ISSUE CURSOR ERROR .
1707  004324  000240                  2$:     NOP
1708                                  ;*****************************************************************
1709                                          ;ROUTINE TO INSURE THE UNIT CAN ENTER LINEAR ADDRESSING
1710                                          ;MODE.  81 CHARACTERS ARE ISSUED TO THE UNIT UNDER TEST
1711                                          ;THEN THE CURSOR POSITION IS READ AND MUST BE ROW1,COL.0.
1712                                  ;*****************************************************************
1713
1714                                  ;*****************************************************************
1715  004326  000004                  TST4:   SCOPE
1716  004330  012737  000005  001156          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
1717  004336  012737  004344  001106          MOV     #CKLIN,$LPADR   ;;SET SCOPE LOOP ADDRESS
1718
1719  004344  004037  016272          CKLIN:  JSR     R0,RESETV       ;RESET THE UNIT-SET MAINT AND LINEAR MODES
1720  004350  013701  016262                  MOV     TBBUF,R1
1721  004354  012703  000120                  MOV     #80.,R3
1722  004360  004037  020222                  JSR     R0,BLDINC       ;LOAD XMIT BUFFER WITH 80 CHAR AND
1723  004364  013721  002102                  MOV     RCUR,(R1)+
1724  004370  013721  002104                  MOV     RDCUR,(R1)+     ;READ CURSOR POSINION.
1725  004374  012737  000123  016270          MOV     #83.,XMCNT
1726  004402  004037  016676                  JSR     R0,XMREC        ;XMIT THE BUFFER.
1727  004406  000402                          BR      1$
1728  004410  104011                          ERROR   11              ;LAST XMIT CAUSED UNIT TO HANG.
1729  004412  000421                          BR      LINXT           ;EXIT TEST
1730  004414  023777  002170  011332  1$:     CMP     R01C00,@RBBUF   ;CURSOR  AT ROW1,COL. 0?
1731  004422  001415                          BEQ     LINXT           ;YES-EXIT
1732  004424  013737  002110  001124          MOV     ESC0,$GDDAT
```

```
1733  004432  000337  001124                  SWAB    $GDDAT
1734  004436  013737  002044  001126          MOV     DRECT,$BDDAT    ;ISSUE ESC SEQUENCE AND CURSOR
1735  004444  104001                          ERROR   1
1736  004446  013746  002170                  MOV     RD1CDD,-(SP)    ;;PUSH RD1CDD ON STACK
1737  004452  004037  017342                  JSR     R0,CURER
1738  004456  000240                  LINXT:  NOP
1739
1740
1741                                  ;*****************************************************************
1742                                  ;TEST TO INSURE OPERATION OF XON/XOFF COMMANDS
1743                                  ;FROM VT61. XOFF IS FORCED BY TRANSMITTING LINE 23 WHILE SIMUL-
1744                                  ;TANEOUSLY FILLING THE SILO WITH DATA. AFTER SENSING
1745                                  ;THE XOFF, THE TEST WAITS FOR THE TRANSMIT TO FINISH AND
1746                                  ;INSURES XON OCCURS BEFORE THE MAX. TRANSFER TIME HAS ELAPSED.
1747                                  ;(30 SECONDS)
1748                                  ;*****************************************************************
1749
1750                                  ;*****************************************************************
1751  004460  000004                  TST5:   SCOPE
1752  004462  012737  000010  001156          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1753  004470  012737  004476  001106          MOV     #BASC3,$LPADR   ;;SET SCOPE LOOP ADDRESS
1754  004476  013701  016262          BASC3:  MOV     TBBUF,R1        ;R1 = 1ST XMIT BUFFER ADDRESS.
1755  004502  012737  001001  002262          MOV     #1001,BLKM      ;SET XMIT TO SOM- DATA -EOM.
1756  004510  005037  002260                  CLR     VSTAT
1757  004514  004037  016272                  JSR     R0,RESETV       ;RESET THE UNIT AND WAIT FOR XON.
1758  004520  013721  002074                  MOV     DCRAD,(R1)+
1759  004524  013721  002200                  MOV     R23CDD,(R1)+    ;CURSOR TO ROW 23, COL.0
1760  004530  013721  002110                  MOV     ESCO,(R1)+
1761  004534  013721  002112                  MOV     XMTAL,(R1)+     ;TRANSMIT THE LINE.
1762  004540  012703  000050                  MOV     #40.,R3
1763  004544  004037  020222                  JSR     R0,BLDINC       ;40 CHAR. OF INCREMENTING CHAR.
1764  004550  012737  000057  016270          MOV     #47.,XMCNT      ;SET UP TO XMIT 47 BYTES
1765  004556  052777  040000  175176          BIS     #XENA,@VJCSR    ;TRANSMIT ENABLES
1766  004564  012703  000050                  MOV     #40.,R3         ;MAXIMUM DELAY EQUAL 400 M.S.
1767  004570  012737  000001  020220  2$:     MOV     #1,DCOUNT
1768  004576  004037  020156                  JSR     R0,DELAY        ;DELAY FOR 10 MILLISEC.
1769  004602  032737  100000  002260          BIT     #RXOFF,VSTAT    ;CHECK FOR XOFF
1770  004610  001007                          BNE     3$              ;FOUND IT EXIT THIS SECTION.
1771  004612  005303                          DEC     R3              ;DELAYED 400 M.S.?
1772  004614  001365                          BNE     2$              ;NO-KEEP LOOKING FOR XOFF.
1773  004616  104012                          ERROR   12              ;GENERAL VT61 FAILURE MESSAGE
1774  004620  012746  100000                  MOV     #100000,-(SP)   ;;PUSH #100000 ON STACK
1775  004624  004037  016512                  JSR     R0,CKSFT        ;GO REPORT ERROR
1776  004630                          3$:
1777  004630  012746  000001                  MOV     #XMDNE,-(SP)    ;;PUSH #XMDNE ON STACK
1778  004634  012746  000062                  MOV     #50.,-(SP)      ;;PUSH #50. ON STACK
1779  004640  004037  021654                  JSR     R0,WTBGND
1780  004644  000411                          BR      EXIT3           ;TIMEOUT-EXIT TEST.
1781  004646  127727  025446  000021          CMPB    @ABUFP,#XON     ;RECEIVED A XON?
1782  004654  001405                          BEQ     EXIT3           ;YES-NO ERROR-EXIT
1783
1784  004656  104012                          ERROR   12              ;GENERAL VT61 FAILURE MESSAGE
1785  004660  012746  000001                  MOV     #000001,-(SP)   ;;PUSH #000001 ON STACK
1786  004664  004037  016512                  JSR     R0,CKSFT
1787  004670  004037  017262          EXIT3:  JSR     R0,RESPTR       ;RESET INTERRUPT POINTERS.
1788
```

```
1789                                    ;*****************************************************************
1790                                    ;ROUTINE TO VERIFY OPERATION OF XOFF AND XON TO THE VT61.
1791                                    ;A FULL SCREEN TRANSMIT IS INITIATED AND A SERIES OF XOFF AND
1792                                    ;XON ARE ISSUED TO THE TERMINAL SEQUENTIALLY.
1793                                    ;ERRORS ARE REPORTED IF XOFF DOES NOT STOP,OR XON RESTART
1794                                    ;THE TRANSMISSION. TEST IS ENDED WHEN EOM IS SENSED.
1795                                    ;*****************************************************************
1796
1797                                    ;*****************************************************************
1798  004674  000004                    TST6:   SCOPE
1799  004676  012737  000001  001156            MOV     #1,$TIMES       ;;DO 1 ITERATION
1800  004704  012737  004712  001106            MOV     #ONOF61,$LPADR  ;;SET SCOPE LOOP ADDRESS
1801
1802  004712  004037  016272            ONOF61: JSR     R0,RESETV       ;RESET THE UNIT AND WAIT FOR XON.
1803  004716  042737  077577  002260            BIC     #77577,VSTAT    ;CLEAR THE FLAGS
1804  004724  013746  002224                    MOV     ZERO,-(SP)      ;;PUSH ZERO ON STACK
1805  004730  013746  002112                    MOV     XMTAL,-(SP)     ;;PUSH XMTAL ON STACK
1806  004734  013746  002110                    MOV     ESCO,-(SP)      ;;PUSH ESCO ON STACK
1807  004740  004037  014370                    JSR     R0,TESC
1808  004744  012737  000010  020220    ONOFLP: MOV     #10,DCOUNT      ;ALLOW 100 M.S. FOR OPERATION
1809  004752  004037  020156                    JSR     R0,DELAY        ;TO BEGIN.
1810  004756  112777  000023  011302            MOVB    #XOFF,@TBUFP
1811  004764  004037  017152                    JSR     R0,XMIT1        ;SEND A XOFF TO VT61.
1812  004770  012704  000036                    MOV     #30.,R4
1813  004774  013705  032320            OFFLP:  MOV     ABUFP,R5        ;ALLOW 300M.S. FOR XMIT TO CEASE
1814  005000  012737  000001  020220            MOV     #1,DCOUNT
1815  005006  004037  020156                    JSR     R0,DELAY
1816  005012  023705  032320                    CMP     ABUFP,R5
1817  005016  001406                            BEQ     ONOFA           ;XMIT STOPPED-GO RESTART IT.
1818  005020  005304                            DEC     R4
1819  005022  001364                            BNE     OFFLP           ;COUNTER NO EQUAL 300 MS-LOOP
1820  005024  013737  002260  001120            MOV     VSTAT,$GDADR    ;UNIT DID NOT RESPOND TO XOFF
1821  005032  104015                            ERROR   15              ;ISSUE ERROR
1822
1823  005034  112777  000021  011224    ONOFA:  MOVB    #XON,@TBUFP
1824  005042  004037  017152                    JSR     R0,XMIT1        ;SEND A XON TO THE VT61.
1825  005046  012704  000036                    MOV     #30.,R4         ;SET UP FOR 300MS DELAY.
1826  005052  032737  020000  002260    ONLP:   BIT     #REOM,VSTAT     ;EOM RECEIVED?
1827  005060  001020                            BNE     ONOFXT          ;YES-EXIT
1828  005062  013705  032320                    MOV     ABUFP,R5
1829  005066  012737  000001  020220            MOV     #1,DCOUNT
1830  005074  004037  020156                    JSR     R0,DELAY        ;ALLOW 300 MS FOR XMIT TO RESTART
1831  005100  023705  032320                    CMP     ABUFP,R5
1832  005104  001317                            BNE     ONOFLP          ;IT RESTARTED-GO STOP IT.
1833  005106  005304                            DEC     R4
1834  005110  001360                            BNE     ONLP            ;NOT YET 300 MS LOOP.
1835  005112  013737  002260  001120            MOV     VSTAT,$GDADR    ;XMIT DIT NOT RESTART-ISSUE
1836  005120  104016                            ERROR   16              ;ERROR AND EXIT
1837  005122  000240                    ONOFXT: NOP
1838
1839                                    ;*****************************************************************
1840                                    ;ROUTINE TO TEST VT61 RAM AND THE COMMUNICATION PATHS.
1841                                    ;THIS ROUTINE ISSUES A SERIES OF PATTERNS(77/100,100/77,
1842                                    ;52/125,INCREMENTING,AND REV. VIDEO INCREMENTING) TO THE VT61.
1843                                    ;THE SCREEN IS THEN TRANSMITTED TO THE HOST AND AFTER EACH
1844                                    ;ITERATION RECEIVED DATA IS CHECKED AND ALL ERRORS(INCLUDING
```

```
                                        ;TRANSMISSION) ARE REPORTED.
                                        ;*****************************************************************
                                        ;*****************************************************************
005124  000004                  TST7:   SCOPE
005126  012737  000001  001156          MOV     #1,$TIMES       ;;DO 1 ITERATION
005134  012737  005142  001106          MOV     #MEM1,$LPADR    ;;SET SCOPE LOOP ADDRESS
005142  004037  016272          MEM1:   JSR     R0,RESETV       ;RESET THE UNIT AND WAIT FOR XON.
005146  005005                          CLR     R5              ;CLEAR PATTERN OFFSET.
005150  016504  005656          MEMA:   MOV     MPATT(R5),R4    ;LOAD PATTERN TO BE TRANSMMITTED
005154  004037  017262                  JSR     R0,RESPTR       ;RESET POINTERS
005160  042737  077577  002260          BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL
005166  112777  000002  011072          MOVB    #SOM,@TBUFP     ;XMIT THE START OF MESSAGE.
005174  004037  017152                  JSR     R0,XMIT1
005200  012702  003600                  MOV     #TOTCH,R2       ;LOAD A COUNT OF SCREEN
005204  005302          MEMB:   DEC     R2              ;DECREMENT XMIT COUNT
005206  001414                  BEQ     10$             ;COUNT = ZERO?
005210  004037  005624          12$:    JSR     R0,PATGN        ;NO-GENERATE NEXT BYTE TO XMIT.
005214  110477  011046                  MOVB    R4,@TBUFP       ;LOAD THE CHARACTER.
005220  004037  017152                  JSR     R0,XMIT1        ;NO-XMIT ANOTHER BYTE.
005224  023737  002234  002236          CMP     FTLCNT,ALWCNT   ;EXCEEDED FATAL ERROR COUNT?
005232  103764                  BLO     MEMB            ;NO-CHECK IF ANOTHER TRANSMISSION REQUIRED.
005234  000137  005676                  JMP     MEMXT           ;YES-GO ABORT TEST.
005240  112777  000004  011020  10$:    MOVB    #EOM,@TBUFP     ;XMIT END OF MESSAGE.
005246  004037  017152                  JSR     R0,XMIT1
005252  004037  017262                  JSR     R0,RESPTR       ;RESET INTERRUPT POINTERS.
005256  013701  016262                  MOV     TBBUF,R1        ;LOAD XMIT BUFFER WITH
005262  013721  002166                  MOV     ESCN,(R1)+
005266  013721  002016                  MOV     CHOM,(R1)+      ;CURSOR HOME
005272  013721  002162                  MOV     ESCZI,(R1)+     ;ESCAPE Z
005276  013721  002110                  MOV     ESCO,(R1)+
005302  013721  002112                  MOV     XMTAL,(R1)+     ;TRANSMIT ALL
005306  013711  002010                  MOV     LNFED,(R1)      ;LINE FEED.
005312  012737  000010  016270          MOV     #8.,XMCNT       ;SET UP TO XMIT 8 BYTES
005320  004037  016676                  JSR     R0,XMREC        ;XMIT,WAIT FOR REC. EOM
005324  000402                  BR      1$              ;NORMAL EXIT
005326  104011                  ERROR   11              ;LAST TRANSMIT CAUSED VT61 TO HANG
005330  000562                  BR      MEMXT           ;EXIT TEST
005332  042737  077577  002260  1$:     BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL
005340  005002                  CLR     R2              ;CLEAR RECEIVE COUNTER.
005342  016504  005656                  MOV     MPATT(R5),R4    ;LOAD PATTERN
005346  012703  032120                  MOV     #TCRLB+300,R3   ;SET UT ERROR STORAGE
005352  013701  015754                  MOV     RBBUF,R1        ;SET UP RECEIVE POINTER
005356  005037  002254          MEMC:   CLR     DLAY            ;SET UP TIME OUT DELAY
005362  013737  015754  015760          MOV     RBBUF,RBUFP     ;RESET RECEIVE POINTER
005370  023701  015760          1$:     CMP     RBUFP,R1        ;RECEIVED A CHAR?
005374  001013                  BNE     MEMD            ;YES-GO CHECK IT.
005376  032737  020000  002260          BIT     #REOM,VSTAT     ;HAVE WE RECEIVED EOM?
005404  001033                  BNE     CKDAT           ;YES, GO CHECK FOR DATA ERRORS
005406  005337  002254                  DEC     DLAY            ;RUN TIME OUT DELAY.
005412  001366                  BNE     1$              ;NOT EXPIRED-KEEP LOOKING.
005414  005237  002234                  INC     FTLCNT          ;TRANSMISSION FAILED-INCR. FATAL COUNT
```

```
1900  005420  104011                        ERROR   11
1901  005422  000525                        BR      MEMXT
1902  005424  005202                MEMD:   INC     R2              ;DATA IN. INCREMENT COUNTER
1903  005426  004037  005624                JSR     R0,PATGN        ;GET GOOD CHARACTER ,PUT IN R4 AND
1904  005432  122705  000010                CMPB    #10,R5          ;CHECKING REV. VIDEO DATA?
1905  005436  001002                        BNE     1$              ;NO-DO NOT MODIFY
1906  005440  052704  000200                BIS     #BIT07,R4       ;YES-FORCE BIT 7.
1907  005444  121104                1$:     CMPB    (R1),R4         ;COMPARE DATA
1908  005446  001743                        BEQ     MEMC
1909  005450  020227  003600                CMP     R2,#TOTCH       ;COMPARING LAST CHAR?
1910  005454  001740                        BEQ     MEMC            ;YES-NEVER COUNT AS A ERROR.
1911
1912  005456  020327  032170                CMP     R3,#TCRLB+350   ;STORED 20 ERRORS?
1913  005462  103335                        BHIS    MEMC            ;YES-STORE NO MORE.
1914  005464  110423                        MOVB    R4,(R3)+        ;STORE THE GOOD DATA.
1915  005466  111123                        MOVB    (R1),(R3)+      ;STORE THE BAD DATA.
1916  005470  010223                        MOV     R2,(R3)+        ;STORE THE RECEIVE COUNT.
1917  005472  000731                        BR      MEMC
1918  005474  022703  032120        CKDAT:  CMP     #TCRLB+300,R3
1919  005500  001415                        BEQ     CKMEM
1920  005502  012701  032120                MOV     #TCRLB+300,R1   ;LOAD FIRST ERROR ADDRESS.
1921  005506  004037  016654        1$:     JSR     R0,CLREG        ;CLEAR ERROR REGISTERS
1922  005512  112137  001124                MOVB    (R1)+,$GDDAT    ;LOAD THE GOOD DATA.
1923  005516  112137  001126                MOVB    (R1)+,$BDDAT    ;LOAD THE ERROR BUFFER
1924  005522  012137  001120                MOV     (R1)+,$GDADR    ;LOAD RECEIVE COUNT
1925  005526  104004                        ERROR   4               ;ISSUE DATA ERROR MESSAGE.
1926  005530  020103                        CMP     R1,R3           ;ISSUED ALL ERRORS?
1927  005532  103765                        BLO     1$              ;NO-CONTINUE
1928
1929  005534  020227  003600        CKMEM:  CMP     R2,#TOTCH       ;DID WE XFER 1920 TIMES?
1930  005540  001406                        BEQ     1$              ;YES - GO CHECK STATUS
```

```
 1931  005542  012737  003600  001124          MOV     #TOTCH,$GDDAT   ;NO, PUT GOOD COUNT IN GDDAT
 1932  005550  010237  001126                  MOV     R2,$BDDAT       ;AND ACTUAL COUNT IN BDDAT.
 1933  005554  104005                          ERROR   5               ;ISSUE COUNT ERROR.
```

```
1934
1935  005556                                  1$:
1936  005556  012746  060000                          MOV     #60000,-(SP)    ;;PUSH #60000 ON STACK
1937  005562  004037  016512                          JSR     R0,CKSFT
1938  005566  062705  000002                          ADD     #2,R5           ;INCREMENT PATTERN POINTER
1939  005572  005765  005656                          TST     MPATT(R5)       ;TEST NEXT PATTERN
1940  005576  001437                                  BEQ     MEMXT           ;ZERO-END OF TEST EXIT.
1941  005600  100007                                  BPL     2$              ;NOT INCRMENTING PATTERN.
1942  005602  122705  000010                          CMPB    #10,R5          ;SET REVERSE VIDEO?
1943  005606  001004                                  BNE     2$              ;NO.
1944  005610  012703  005672                          MOV     #SETREV,R3      ;YES-ENTER REVERSE VIDEO
1945  005614  004037  017222                          JSR     R0,LDXMIT       ;AND RE-ISSUE INCREMENTING PATTERN.
1946  005620  000137  005150          2$:             JMP     MEMA            ;NOT ZERO, GO EXERCISE IT.
1947
1948  005624  042704  000200          PATGN:          BIC     #200,R4         ;CLEAR REV. VIDEO BIT IF SET.
1949  005630  005704                                  TST     R4              ;CHECK R4 FOR PATTERN
1950  005632  100402                                  BMI     1$              ;IF MINUS, DO INCREMENTING.
1951  005634  000304                                  SWAB    R4              ;OTHERWISE SWAP BYTES AND
1952  005636  000200                                  RTS     R0              ;EXIT.
1953  005640  105204                          1$:     INCB    R4              ;ADD ONE TO INCREMENTING
1954  005642  120427  000177                          CMPB    R4,#177         ;HAVE WE EXCEEDED LIMIT
1955  005646  103402                                  BLO     2$              ;NO, EXIT
1956  005650  016504  005656                          MOV     MPATT(R5),R4    ;YES, RESET PATTERN AND
1957  005654  000200                          2$:     RTS     R0              ;EXIT.
1958                                                  .EVEN
1959          005656                          MPATT   =.
1960  005656  037500                                  .WORD   037500          ;PATTERN 77,100
1961  005660  040077                                  .WORD   040077          ;PATTERN 100,77
1962  005662  025125                                  .WORD   025125          ;PATTERN 52,125
1963  005664  100040                                  .WORD   100040          ;PATTERN INCREMENTING
1964  005666  100040                                  .WORD   100040          ;PATTERN INCREMENTING-REV. VIDEO.
1965  005670  000000                                  .WORD   0               ;PATTERN TABLE TERMINATOR
1966                                                  ;SEQUENCE TO ENTER REVERSE VIDEO.
1967  005672      033     117     112   SETREV:       .BYTE   .ESC,.O,.EEMP,0
1968  005675      000
1969  005676  000240                    MEMXT:        NOP
1970
1971                                    ;*****************************************************************
1972
1973                                          ;ROUTINE TO TEST THE ABILITY OF THE VT61 TO CALCULATE
1974                                          ;AND TRANSMIT CHECKSUMS OF BOTH TRANSMITTED AND RECEIVED
1975                                          ;DATA.  SUBTEST A TRANSMITS A FULL BUFFER UPDATING A CALCULATED
1976                                          ;CHECKSUM ON EACH CHARACTER TRANSMITTED.  AN ESCAPE SEQUENCE
1977                                          ;REQUESTING THE RECEIVER CHECKSUM IS EMBEDDED AT THE END OF
1978                                          ;XMIT BUFFER AND THE RECEIVED CHECKSUM IS COMPARED TO THE
1979                                          ;CALCULATED.  SUBTEST B PERFORMS THE SAME TYPE OF CHECK ON
1980                                          ;THE VT61 TRANSMIT CHECKSUM,UTILIZING THE DATA SENT TO THE VT61
1981                                          ;IN SUBTEST A,DURING A FULL SCREEN TRANSMIT.
1982
1983                                    ;*****************************************************************
1984
1985                                    ;*****************************************************************
1986  005700  000004                    TST10:  SCOPE
1987  005702  012737  000003  001156            MOV     #3,$TIMES       ;;DO 3 ITERATIONS
1988  005710  012737  005716  001106            MOV     #CKSUMA,$LPADR  ;;SET SCOPE LOOP ADDRESS
1989
```

```
1990  005716  004037  016272          CKSUMA: JSR     RO,RESETV       ;RESET THE UNIT AND WAIT FOR XON.
1991  005722  004037  017262                  JSR     RO,RESPTR       ;RESET INTERRUPT POINTERS
1992  005726  012737  001001  002262          MOV     #1001,BLKM      ;SET XMIT TO SOM- DATA -EOM.
1993  005734  012703  006344                  MOV     #ITSUMA,R3      ;DIS. RECT. MODE AND CLEAR CHECKSUM
1994  005740  004037  017222                  JSR     RO,LDXMIT
1995  005744  042737  077577  002260          BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL
1996  005752  013701  016262                  MOV     TBBUF,R1        ;LOAD XMIT BUFFER WITH
1997  005756  012703  000473                  MOV     #315.,R3
1998  005762  004037  020222                  JSR     RO,BLDINC       ;314 INCREMENTING CHAR.
1999  005766  113721  002166                  MOVB    ESCN,(R1)+
2000  005772  113721  002016                  MOVB    CHOM,(R1)+      ;CURSOR HOME
2001  005776  113721  002110                  MOVB    ESCO,(R1)+
2002  006002  113721  002111                  MOVB    ESCO+1,(R1)+
2003  006006  113711  002126                  MOVB    TXRCK,(R1)      ;TRANSMIT RECEIVER CHECKSUM.
2004  006012  005004                          CLR     R4              ;CLEAR CHECKSUM REGISTER
2005  006014  012705  000004                  MOV     #EOM,R5         ;PRELOAD CHECKSUM REG. WITH
2006  006020  004037  020700                  JSR     RO,CALCK        ;EOM FROM PRIOR XMIT.
2007  006024  052737  002000  002260          BIS     #CKSUM,VSTAT    ;REQUEST CHECKSUM CALCULATIONS.
2008  006032  012737  000500  016270          MOV     #320.,XMCNT     ;SETUP TO XMIT 320 BYTES
2009  006040  052777  040000  173714          BIS     #XENA,@VJCSR    ;ENABLE XMIT INTERRUPTS
2010  006046  012746  020000                  MOV     #REOM,-(SP)     ;;PUSH #REOM ON STACK
2011  006052  012746  000012                  MOV     #10.,-(SP)      ;;PUSH #10. ON STACK
2012  006056  004037  021654                  JSR     RO,WTBGND       ;LOOK FOR EOM.
2013  006062  000534                          BR      CKEXT           ;ERROR EXIT IF NOT FOUND
2014  006064  127704  007664                  CMPB    @RBBUF,R4       ;COMPARE CHECKSUMS
2015  006070  001414                          BEQ     CKSUMB          ;GOOD GO TO SUBTEST B
2016  006072  004037  016654                  JSR     RO,CLREG        ;BAD COMPARE
2017  006076  110437  001124                  MOVB    R4,$GDDAT       ;LOAD CALCULATED CHECKSUM
2018  006102  117737  007646  001126          MOVB    @RBBUF,$BDDAT   ;AND VT61 RECEIVER CHECKSUM
2019  006110  104013                          ERROR   13              ;ISSUE ERROR
2020  006112  012746  060001                  MOV     #60001,-(SP)    ;;PUSH #60001 ON STACK
2021  006116  004037  016512                  JSR     RO,CKSFT        ;ERROR.
2022
2023  006122  042737  077577  002260  CKSUMB: BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL
2024  006130  005004                          CLR     R4              ;CLEAR CHECKSUM REGISTER
2025  006132  052737  000100  002260          BIS     #TXSUM,VSTAT    ;SET UP FOR XMIT CHECKSUM GENERATION.
2026  006140  012737  001001  002262          MOV     #1001,BLKM      ;SET XMIT TO SOM- DATA -EOM.
2027  006146  013701  016262                  MOV     TBBUF,R1
2028  006152  004037  020746                  JSR     RO,LDBUF        ;LOAD THE BUFFER WITH:
2029  006156      033     117      134        .BYTE   .ESC,.0,.CLTCK,.ESC,.0,.XMTAL,.ESC,.0,.TXTCK,0
2030  006161      033     117      126
2031  006164      033     117      136
2032  006167      000
2033  006170  012737  000011  016270          MOV     #9.,XMCNT       ;SET UP TO XMIT 9 BYTES
2034  006176  052777  040000  173556          BIS     #XENA,@VJCSR    ;ALLOW XMIT INTERRUPTS
2035  006204  012746  000001                  MOV     #XMDNE,-(SP)    ;;PUSH #XMDNE ON STACK
2036  006210  012746  000002                  MOV     #2,-(SP)        ;;PUSH #2 ON STACK
2037  006214  004037  021654                  JSR     RO,WTBGND       ;LOOK FOR XMIT DONE.
2038  006220  000455                          BR      CKEXT           ;TIME OUT - EXIT TEST.
2039  006222  005037  002254          CKSRC:  CLR     DLAY            ;SET UP TIME OUT DELAY
2040  006226  013702  032320                  MOV     ABUFP,R2        ;RESET THE RECEIVER FLAG
2041  006232  023702  032320          1$:     CMP     ABUFP,R2        ;RECEIVED A CHAR?
2042  006236  001007                          BNE     2$              ;YES-GO CHECK IT.
2043  006240  005337  002254                  DEC     DLAY            ;RUN TIME OUT DELAY.
2044  006244  001372                          BNE     1$
2045  006246  005237  002234                  INC     FTLCNT          ;TIMED OUT-INCREMENT FATAL XMIT COUNT
```

```
2046  006252  104011                          ERROR   11              ;ISSUE HUNG MESSAGE AND EXIT.
2047  006254  000437                          BR      CKEXT
2048  006256  122777  000004  024034  2$:     CMPB    #EOM,@ABUFP     ;RECEIVED EOM CHAR?
2049  006264  001356                          BNE     CKSRC
2050  006266  042737  020000  002260          BIC     #REOM,VSTAT     ;CLEAR THE EOM FLAG
2051  006274  032737  020000  002260          BIT     #REOM,VSTAT     ;NOW WAIT FOR LAST EOM FLAG
2052  006302  001774                          BEQ     .-6             ;FROM XMIT TRANSMITTER CHECKSUM.
2053  006304  120477  007444                  CMPB    R4,@RBBUF       ;COMPARE 61 TO HOST CHECKSUM.
2054  006310  001421                          BEQ     CKEXT           ;EQUAL - EXIT TEST
2055  006312  004037  016654                  JSR     R0,CLREG
2056  006316  110437  001124                  MOVB    R4,$GDDAT       ;LOAD THE HOST CALCULATED CHECKSUM
2057  006322  117737  007426  001126          MOVB    @RBBUF,$BDDAT   ;LOAD THE VT61 TRANSMITTED CHECKSUM
2058  006330  104014                          ERROR   14              ;ISSUE VT61 XMIT CHECKSUM ERROR
2059  006332  012746  060001                  MOV     #60001,-(SP)    ;;PUSH #60001 ON STACK
2060  006336  004037  016512                  JSR     R0,CKSFT        ;CHECK FOR STATUS ERROR
2061  006342  000404                          BR      CKEXT
2062
2063  006344     033     117     103  ITSUMA: .BYTE   .ESC,.0,.DRECT,.ESC,.0,.CLRCK,0,0
2064  006347     033     117     133
2065  006352     000     000
2066
2067  006354  004037  017262          CKEXT:  JSR     R0,RESPTR
2068
2069                                  ;*****************************************************************
2070                                          ;ROUTINE TO INSURE BASIC CURSOR COMMANDS
2071                                          ;RESULT IN CORRECT CURSOR MOVEMENT. COMMANDS
2072                                          ;ARE ISSUED IN THE SEQUENCE:  RESET, CURSOR RIGHT,
2073                                          ;CURSOR DOWN, CURSOR LEFT, AND CURSOR UP. THE READ
2074                                          ;CURSOR POSITION COMMAND IS ISSUED AFTER EVERY
2075                                          ;CURSOR COMMAND AND CURRENT IS COMPARED TO GOOD
2076                                          ;AND ANY ERRORS REPORTED.
2077                                  ;*****************************************************************
2078
2079                                  ;*****************************************************************
2080  006360  000004                  TST11:  SCOPE
2081  006362  012737  000005  001156          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
2082  006370  012737  006376  001106          MOV     #CURS1A,$LPADR  ;;SET SCOPE LOOP ADDRESS
2083
2084  006376  013701  016262          CURS1A: MOV     TBBUF,R1        ;LOAD XMIT BUFFER ADDRESS
2085  006402  004037  016272                  JSR     R0,RESETV       ;RESET THE UNIT AND WAIT FOR XON.
2086  006406  004037  020746                  JSR     R0,LDBUF        ;LOAD THE BUFFER WITH:
2087  006412     033     103     033          .BYTE   .ESC,.CRT,.ESC,.0,.RDCUR,.ESC,.CDWN,.ESC
2088  006415     117     131     033
2089  006420     102     033
2090  006422     117     131     033          .BYTE   .0,.RDCUR,.ESC,.CLFT,.ESC,.0,.RDCUR
2091  006425     104     033     117
2092  006430     131
2093  006431     033     101     033          .BYTE   .ESC,.CUP,.ESC,.0,.RDCUR,.BEL,0
2094  006434     117     131     007
2095  006437     000
2096  006440  012737  000024  016270          MOV     #20.,XMCNT      ;SET TO XMIT 20 CHARACTERS
2097  006446  012737  000004  017144          MOV     #4,RECITT       ;SET RECEIVE ITERATION TO 4
2098  006454  012737  031720  017146          MOV     #TCRLB+100,WDSTOR;SET UP WORD STORAGE POINTER
2099  006462  004037  016676                  JSR     R0,XMREC        ;XMIT ,AND WAIT FOR REC.DONE
2100  006466  000402                          BR      11$             ;NORMAL EXIT
2101  006470  104011                          ERROR   11              ;LAST XMIT CAUSED VT61 TO HANG.
```

```
2102  006472  000436                        BR      CUR1XT          ;EXIT TEST
2103  006474  012701  006560          11$:  MOV     #GDCURP,R1      ;R1=GOOD POSITION TABLE
2104  006500  012702  031720                MOV     #TCRLB+100,R2   ;R2=ACTUAL CURSOR POSITION
2105  006504  012703  002020                MOV     #CRT,R3         ;R3=CURSOR COMMAND TABLE
2106
2107  006510  021112                  12$:  CMP     (R1),(R2)       ;COMPARE GOOD TO ACTUAL
2108  006512  001415                        BEQ     2$              ;OK-GO UPDATE POINTERS.
2109  006514  113737  002166  001125        MOVB    ESCN,$GDDAT+1
2110  006522  111337  001124                MOVB    (R3),$GDDAT     ;LOAD COMMAND IN ESC ERROR
2111  006526  005037  001126                CLR     $BDDAT
2112  006532  104001                        ERROR   1               ;AND ISSUE IT
2113  006534  011237  031120                MOV     (R2),RCRLB      ;LOAD BAD CURSOR POSITION
2114  006540  011146                        MOV     (R1),-(SP)      ;;PUSH (R1) ON STACK
2115  006542  004037  017342                JSR     R0,CURER        ;LOAD AND ISSUE CURSRO ERROR MESSAGE
2116  006546  022122                  2$:   CMP     (R1)+,(R2)+     ;INCREMENT POSITION POINTERS.
2117  006550  022337  002026                CMP     (R3)+,CUP       ;CHECK FOR COMMAND TERM.(CUP).
2118  006554  001355                        BNE     12$             ;NOT AT TERMINATOR-COMPARE AGAIN
2119  006556  000404                        BR      CUR1XT          ;EXIT TEST
2120
2121
2122  006560  020440                GDCURP: .WORD   20440           ;ROW 0, COL. 1
2123  006562  020441                        .WORD   20441           ;ROW 1, COL. 1
2124  006564  020041                        .WORD   20041           ;ROW 1, COL. 0
2125  006566  020040                        .WORD   20040           ;ROW 0, COL. 0
2126  006570  000240                CUR1XT: NOP
2127
2128                                  ;*****************************************************************
2129                                          ;ROUTINE TO INSURE THAT READ CHARACTER AT CURSOR
2130                                          ;FUNCTIONS CORRECTLY. COMMAND SEQUENCE IS:  RESET, A, CURSOR
2131                                          ;LEFT, READ  CHARACTER AT CURSOR.
2132                                          ;AN ERROR IS REPORTED IF THE LAST READ IS NOT AN "A".
2133                                  ;*****************************************************************
2134
2135                                  ;*****************************************************************
2136  006572  000004                TST12:  SCOPE
2137  006574  012737  000005  001156        MOV     #5,$TIMES       ;;DO 5 ITERATIONS
2138  006602  012737  006610  001106        MOV     #CURS1B,$LPADR  ;;SET SCOPE LOOP ADDRESS
2139
2140  006610  013701  016262          CURS1B: MOV     TBBUF,R1
2141  006614  004037  016272                JSR     R0,RESETV       ;RESET THE UNIT AND WAIT FOR XON.
2142  006620  012721  000101                MOV     #101,(R1)+      ;A
2143  006624  113721  002166                MOVB    ESCN,(R1)+
2144  006630  113721  002024                MOVB    CLFT,(R1)+      ;CURSOR LEFT
2145  006634  013721  002110                MOV     ESCOI,(R1)+
2146  006640  013711  002120                MOV     TCUCH,(R1)      ;TRANSMIT CH. AT CURSOR
2147  006644  012737  000006  016270        MOV     #6,XMCNT        ;SET UP TO XMIT 6 CHARACTERS
2148  006652  004037  016676                JSR     R0,XMREC        ;XMIT STRING AND WAIT FOR EOM.
2149  006656  000402                        BR      10$             ;NORMAL EXIT
2150  006660  104011                        ERROR   11              ;LAST XMIT CAUSED VT61 TO HANG/FAIL
2151  006662  000430                        BR      2$              ;EXIT TEST
2152  006664  127727  007064  000101  10$:  CMPB    @RBBUF,#101     ;CHARACTER READ=A
2153  006672  001424                        BEQ     2$              ;YES-NEXT SUBTEST
2154  006674  013737  002110  001124        MOV     ESCOI,$GDDAT
2155  006702  000337  001124                SWAB    $GDDAT          ;REASSEMBLE ESC DATA
2156  006706  005037  001126                CLR     $BDDAT
2157  006712  113737  002120  001127        MOVB    TCUCH,$BDDAT+1  ;LOAD FAILING ESC SEQUENCE
```

```
2158  006720  104001                        ERROR   1                       ;AND ISSUE IT
2159  006722  004037  016654                JSR     R0,CLREG
2160  006726  112737  000101  001124        MOVB    #101,$GDDAT             ;LOAD GOOD CH. AND CH.
2161  006734  117737  007014  001126        MOVB    @RBBUF,$BDDAT
2162  006742  104004                        ERROR   4                       ;READ AND ISSUE THEM.
2163
2164  006744  000240                2$:     NOP                             ;END OF TEST
2165                                ;*****************************************************************
2166                                        ;ROUTINE TO VERIFY OPERATION OF REPLACE AND INSERT MODE.
2167                                        ;INITIALLY ROW 0 IS WRITTEN TO 80 INCREMENTING CHAR.
2168                                        ;ON THE FIRST PASS(REPLACE MODE) A CHARACTER IS REPLACED
2169                                        ;AT HOME AND THE CHAR. AT ROW0,COL.0(172) AND ROW1,COL0(NULL)
2170                                        ;ARE VERIFIED. ON THE SECOND PASS, INSERT MODE IS ENTERED
2171                                        ;AND THE RESULTING INSERTION(AT HOME) IS VERIFIED.ROW0,COL0
2172                                        ;SHOULD BE 172 AND ROW1,COL0 SHOULD BE 161.
2173                                ;*****************************************************************
2174
2175                                ;*****************************************************************
2176  006746  000004                TST13:  SCOPE
2177  006750  012737  000005  001156        MOV     #5,$TIMES               ;;DO 5 ITERATIONS
2178  006756  012737  006764  001106        MOV     #INRPL,$LPADR           ;;SET SCOPE LOOP ADDRESS
2179
2180  006764  004037  016272        INRPL:  JSR     R0,RESETV               ;RESET THE UNIT
2181  006770  013701  016262                MOV     TBBUF,R1
2182  006774  005201                        INC     R1                      ;LEAVE ROOM IN BUFFER FOR SOM.
2183  006776  012703  000120                MOV     #80.,R3                 ;CREATE A LINE OF 80 INCREMENTING
2184  007002  004037  020222                JSR     R0,BLDINC               ;CHAR. ON THE SCREEN.
2185  007006  105011                        CLRB    (R1)
2186  007010  013703  016262                MOV     TBBUF,R3
2187  007014  004037  017222                JSR     R0,LDXMIT
2188  007020  005005                        CLR     R5                      ;USE R5 AS TEST INDEXER.
2189  007022  012737  000002  017144  INAG: MOV     #2,RECITT               ;SET UP TO RECEIVE 2 CHAR.
2190  007030  012737  032020  017150        MOV     #TCRLB+200,BYSTOR       ;SET UP STORAGE AREA.
2191  007036  013701  016262                MOV     TBBUF,R1
2192  007042  004037  020746                JSR     R0,LDBUF                ;LOAD THE BUFFER WITH:
2193  007046     033     110     172        .BYTE   .ESC,.CHOM,172,.ESC,.CHOM,.ESC,.0,.TCUCH
2194  007051     033     110     033
2195  007054     117     127
2196  007056     033     102     033        .BYTE   .ESC,.CDWN,.ESC,.0,.TCUCH,0
2197  007061     117     127     000
2198  007064  012737  000015  016270        MOV     #13.,XMCNT              ;SET UP TO XMIT 13 CAHR.
2199  007072  004037  016676                JSR     R0,XMREC
2200  007076  000402                        BR      1$                      ;NORMAL EXIT
2201  007100  104011                        ERROR   11                      ;LAST XMIT CAUSED UNIT TO HANG.
2202  007102  000433                        BR      INRXT                   ;EXIT TEST.
2203  007104  026537  007162  032020  1$:   CMP     TDATA(R5),TCRLB+200     ;COMPARE GOOD TO REC.DATA.
2204  007112  001407                        BEQ     2$                      ;GOOD-LOOP OR EXIT.
2205  007114  016537  007154  001126        MOV     TFUNCT(R5),$BDDAT
2206  007122  013737  002154  001124        MOV     ESCP,$GDDAT             ;LOAD ESCAPE SEQ. ERROR.
2207  007130  104001                        ERROR   1
2208  007132  005725                2$:     TST     (R5)+                   ;INCREMENT INDEXER.
2209  007134  020527  000004                CMP     R5,#4                   ;THRU WITH TEST?
2210  007140  001414                        BEQ     INRXT                   ;YES-EXIT.
2211  007142  012703  007166                MOV     #ENSRT,R3               ;NO-SECOND PASS- ENTER
2212  007146  004037  017222                JSR     R0,LDXMIT               ;INSERT MODE AND DO AGAIN.
2213  007152  000723                        BR      INAG
```

```
                                                                
007154  000151  000111  177777  TFUNCT: .WORD   .ERPL,.EINST,-1
007162    172     000     172   TDATA:  .BYTE   172,0,172,160
007165    160
007166    033     120     111   ENSRT:  .BYTE   .ESC,.P,.EINST,0
007171    000
007172  000240                  INRXT:  NOP
                                
                                ;****************************************************************
                                        ;ROUTINE TO INSURE VT61 WILL SCROLL IF A LINE FEED
                                        ;IS ISSUED FROM ROW 23 OR A DATA ENTRY FROM ROW23,COL. 79.
                                        ;IN SUBTEST A, ROW 0 IS INTIIALLY WRITTEN TO A 0 AND ROW 1
                                        ;A 1. AFTER COMPLETION OF A LINE FEED(AND RESULTING SCROLL)
                                        ;ROW 00,COL.00 IS EXPECTED TO CONTAIN A 1.
                                        ;IN SUBTEST B, THE CURSOR IS PLACED AT ROW23,COL.79
                                        ;AND A DATA CHARACTER "A" IS ENTERED. THE CURSOR
                                        ;POSITION IS THEN READ AND SHOULD BE ROW23,COL.00. THE
                                        ;CHARACTER AT HOME IS VERIFIED TO BE A NULL.
                                ;****************************************************************
                                
                                ;****************************************************************
007174  000004                  TST14:  SCOPE
007176  012737  000005  001156          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
007204  012737  007212  001106          MOV     #CKSCRA,$LPADR  ;;SET SCOPE LOOP ADDRESS
                                
007212  004037  016272          CKSCRA: JSR     R0,RESETV       ;RESET THE UNIT.
007216  013701  016262                  MOV     TBBUF,R1
007222  004037  020746                  JSR     R0,LDBUF        ;LOAD THE XMIT BUFFER WITH:
007226    060     033     102           .BYTE   60,.ESC,.CDWN,.ESC,.CLFT,61,.ESC,.Y,.R23,.C00
007231    033     104     061
007234    033     131     067
007237    040
007240    012     033     110           .BYTE   .LNFED,.ESC,.CHOM,.ESC,.0,.TCUCH,.BEL,0
007243    033     117     127
007246    007     000
007250  012737  000020  016270          MOV     #16.,XMCNT      ;SET UP TO XMIT 16 BYTES.
007256  004037  016676                  JSR     R0,XMREC
007262  000402                          BR      1$              ;NORMAL EXIT
007264  104011                          ERROR   11              ;LAST XMIT CAUSED UNIT TO HANG.
007266  000452                          BR      GDSCRL          ;EXIT TEST.
007270  127727  006460  000061  1$:     CMPB    @RBBUF,#61      ;CHARACTER AT HOME A 1?
007276  001401                          BEQ     CKSCRB          ;YES-NEXT TEST
007300  104023                          ERROR   23              ;NO-ISSUE NO SCROLL ERROR.
007302  012737  000002  017144  CKSCRB: MOV     #2,RECITT       ;SET UP FOR TWO REC. LOOPS.
007310  012737  032020  017146          MOV     #TCRLB+200,WDSTOR       ;SET UP CURSOR POSITION STROAGE.
007316  013701  016262                  MOV     TBBUF,R1
007322  004037  020746                  JSR     R0,LDBUF        ;LOAD XMIT BUFFER WITH:
007326    033     131     067           .BYTE   .ESC,.Y,.R23,.C79,101,.ESC,.0,.RDCUR
007331    157     101     033
007334    117     131
007336    033     110     033           .BYTE   .ESC,.CHOM,.ESC,.0,.TCUCH,0
007341    117     127     000
007344  012737  000015  016270          MOV     #13.,XMCNT      ;SET UP TO XMIT 13 BYTES.
007352  004037  016676                  JSR     R0,XMREC        ;XMIT AND WAIT FOR RECEIVED DONE.
007356  000402                          BR      1$
007360  104011                          ERROR   11              ;LAST XMIT CAUSED VT61 TO HANG.
```

```
2270  007362  000414                          BR      GDSCRL          ;ERROR EXIT
2271  007364  127737  006364  002224  1$:     CMPB    @RBBUF,ZERO     ;NULL RECEIVED?
2272  007372  001410                          BEQ     GDSCRL          ;YES-EXIT TEST
2273  007374  104023                          ERROR   23              ;NO-ISSUE NO SCROLL ERROR.
2274  007376  013777  032020  006350          MOV     TCRLB+200,@RBBUF        ;LOAD RECEIVED CURSOR POSITION.
2275  007404  013746  002200                  MOV     R23C00,-(SP)    ;;PUSH R23C00 ON STACK
2276  007410  004037  017342                  JSR     R0,CURER        ;GO ISSUE CURSOR ERROR.
2277  007414  000240                  GDSCRL: NOP
2278
2279
2280                                  ;*****************************************************************
2281                                          ;THIS TEST INSURES THAT THE VT61 CURSOR CAN BE
2282                                          ;POSITIONED TO VERY POSSIBLE ROW/COLUMN POSITON
2283                                          ;ON THE SCREEN.  THIS IS TESTED BY FILLING THE
2284                                          ;COMPLETE SCREEN WITH A CHARACTER(A) AND THEN
2285                                          ;POSITONING THE CURSOR (VIA DCA) TO EVERY POSITION
2286                                          ;AND THE "A" AT THAT POSITION IS REPLACED WITH A SPACE.
2287                                          ;THE SCREEN IS THEN READ TO VERIFY THAT ONLY SPACES
2288                                          ;EXIST ON THE SCREEN.  ALL POSITIONS CONTAINING
2289                                          ;NON-SPACES ARE REPORTED.
2290
2291                                  ;*****************************************************************
2292
2293                                  ;*****************************************************************
2294  007416  000004                  TST15:  SCOPE
2295  007420  012737  000001  001156          MOV     #1,$TIMES       ;;DO 1 ITERATION
2296  007426  012737  007434  001106          MOV     #CURS2,$LPADR   ;;SET SCOPE LOOP ADDRESS
2297
2298  007434  042737  077577  002260  CURS2:  BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL
2299  007442  004037  016272                  JSR     R0,RESETV       ;RESET THE UNIT AND WAIT FOR XON.
2300  007446  012702  003600                  MOV     #TOTCH,R2       ;LOAD A COUNT OF SCREEN(1920).
2301  007452  112777  000002  006606          MOVB    #SOM,@TBUFP     ;XMIT THE START OF MESSAGE.
2302  007460  004037  017152                  JSR     R0,XMIT1
2303  007464  005302                  1$:     DEC     R2              ;DECREMENT XMIT COUNT
2304  007466  001413                          BEQ     10$             ;COUNT = ZERO?
2305
2306  007470  112777  000101  006570          MOVB    #101,@TBUFP     ;LOAD THE CHARACTER(A).
2307  007476  004037  017152                  JSR     R0,XMIT1        ;NO-XMIT ANOTHER BYTE.
2308  007502  023737  002234  002236          CMP     FTLCNT,ALWCNT   ;EXCEEDED FATAL ERROR COUNT?
2309  007510  103765                          BLO     1$              ;NO-CHECK IF XMIT COMPLETE.
2310  007512  000137  010102                  JMP     C2XT            ;YES-GO ABORT TEST.
2311  007516  004037  017262          10$:    JSR     R0,RESPTR       ;RESET INTERRUPT POINTERS.
2312  007522  013737  002214  017544          MOV     R23C78,LNRW     ;SET UP 1ST ADDRESS
2313  007530  013701  016262                  MOV     TBBUF,R1        ;LOAD XMIT BUFFER WITH
2314  007534  013721  002074                  MOV     DCRAD,(R1)+
2315  007540  010102                          MOV     R1,R2
2316  007542  013721  002214                  MOV     R23C78,(R1)+    ;CURSOR TO LOWER RIGHT -1.
2317  007546  112721  000040                  MOVB    #40,(R1)+       ;SPACE
2318  007552  012737  000005  016270  2$:     MOV     #5,XMCNT        ;SET UP TO XMIT 5 CHARACTERS
2319  007560  042737  077577  002260          BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL
2320  007566  052777  040000  172166          BIS     #XENA,@VJCSR    ;XMIT INTERRUPTS.
2321  007574  012746  000001                  MOV     #XMDNE,-(SP)    ;;PUSH #XMDNE ON STACK
2322  007600  012746  000002                  MOV     #2,-(SP)        ;;PUSH #2 ON STACK
2323  007604  004037  021654                  JSR     R0,WTBGND       ;LOOK FOR  XMIT DONE
2324  007610  000534                          BR      C2XT            ;NOT FOUND-ERROR EXIT
2325  007612  021237  002210                  CMP     (R2),CUHME      ;DELETED TO HOME?
```

```
2326  007616  001405                          BEQ     3$              ;YES
2327  007620  004037  017440                  JSR     R0,CMPOS        ;NO-GET NEXT POSITION TO BE DELETED
2328  007624  013712  017544                  MOV     LNRW,(R2)       ;LOAD IT IN XMIT BUFFER
2329  007630  000750                          BR      2$              ;AND DELETE IT.
2330  007632  004037  017262          3$:     JSR     R0,RESPTR       ;RESET INTERRUPT POINTERS
2331  007636  013737  002210  017544          MOV     CUHME,LNRW      ;LOAD INITIAL CHECK POSITION(HOME)
2332  007644  013701  016262                  MOV     TBBUF,R1        ;LOAD XMIT BUFFER WITH
2333  007650  010102                          MOV     R1,R2           ;STORE ERRORS IN XMIT BUFFER
2334  007652  042737  077577  002260          BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL
2335  007660  012737  001001  002262          MOV     #1001,BLKM      ;SET XMIT TO SOM- DATA -EOM.
2336  007666  013721  002166                  MOV     ESCN,(R1)+
2337  007672  013721  002016                  MOV     CHOM,(R1)+      ;CURSOR HOME
2338  007676  013721  002110                  MOV     ESCO,(R1)+
2339  007702  013721  002112                  MOV     XMTAL,(R1)+     ;TRANSMIT ALL
2340  007706  012737  000005  016270          MOV     #5,XMCNT
2341  007714  052777  040000  172040          BIS     #XENA,@VJCSR    ;SET XMIT ENABLE
2342  007722  012746  000001                  MOV     #XMDNE,-(SP)    ;;PUSH #XMDNE ON STACK
2343  007726  012746  000003                  MOV     #3,-(SP)        ;;PUSH #3 ON STACK
2344  007732  004037  021654                  JSR     R0,WTBGND       ;LOOK FOR SOM OR XMIT DONE
2345  007736  000461                          BR      C2XT            ;NOT FOUND-ERROR EXIT
2346  007740  013701  015760          4$:     MOV     RBUFP,R1        ;SET UP RECEIVE FLAG
2347  007744  005037  002254                  CLR     DLAY            ;SET UP TIME OUT DELAY
2348  007750  020137  015760          40$:    CMP     R1,RBUFP        ;CHARACTER RECEIVED?
2349  007754  103411                          BLO     41$             ;YES-GO CHECK IT.
2350  007756  032737  020000  002260          BIT     #REOM,VSTAT     ;LOOK FOR END OF MESSAGE
2351  007764  001025                          BNE     C2CK            ;FOUND IT, EXIT TEST
2352  007766  005337  002254                  DEC     DLAY            ;RUN TIME OUT DELAY.
2353  007772  001366                          BNE     40$             ;AND LOOK FOR RECEIVED CH.
2354  007774  104011                          ERROR   11              ;LAST XMIT CAUSED VT61 TO HANG.
2355  007776  000420                          BR      C2CK            ;GO SEE IF ANY ERRORS STORED.
2356  010000  013737  015754  015760  41$:    MOV     RBBUF,RBUFP     ;RESET RECEIVE POINTER
2357  010006  127727  005742  000040          CMPB    @RBBUF,#40      ;CHAR EQUAL A SPACE?
2358  010014  001003                          BNE     6$              ;NOT A SPACE-MUST BE ERROR-STORE IT
2359  010016  004037  017502          5$:     JSR     R0,CPPOS        ;UPDATE CURSOR POSITION
2360  010022  000746                          BR      4$
2361  010024  022702  031644          6$:     CMP     #TCRLB+20.,R2   ;STORED 10 ERRORS?
2362  010030  101772                          BLOS    5$              ;YES-IGNORE ANY FURTHER ERRORS.
2363  010032  013722  017544                  MOV     LNRW,(R2)+      ;STORE FAILING CURSOR POSITION
2364  010036  000767                          BR      5$
2365
2366  010040  020237  016262          C2CK:   CMP     R2,TBBUF        ;ANY ERRORS STORED?
2367  010044  001416                          BEQ     C2XT            ;NO EXIT TEST
2368  010046  013701  016262                  MOV     TBBUF,R1        ;USE R1 AS ERROR POINTER
2369  010052  021137  002076          1$:     CMP     (R1),R23C79     ;CURSOR TO LOWER RIGHT?
2370  010056  001411                          BEQ     C2XT            ;YES-NOT AN ERROR.
2371  010060  104012                          ERROR   12              ;NO-ISSUE ERROR MESSAGES
2372  010062  012746  020040                  MOV     #20040,-(SP)    ;;PUSH #20040 ON STACK
2373  010066  012177  005662                  MOV     (R1)+,@RBBUF    ;LOAD FAILING POS.
2374  010072  004037  017342                  JSR     R0,CURER        ;ISSUE CURSOR ERROR
2375  010076  020102                          CMP     R1,R2           ;DONE WITH ERRORS?
2376  010100  103764                          BLO     1$              ;NO, DUMP ANOTHER.
2377  010102  000240          C2XT:   NOP                             ;EXIT TEST
2378
2379                          ;*****************************************************************
2380                                  ;ROUTINE TO INSURE PROPER OPERATION OF CARRIAGE RETURN
2381                                  ;AND LINE FEED DURING NORMAL MODE. INITIALLY THE CURSOR IS
```

```
                                        ;SET(VIA D.C.A.) TO ROW0,COL 20 AND A LINE FEEL IS ISSUED
                                        ;THE CURSOR POSITION IS THEN READ AND MUST BE ROW1,COL20.
                                        ;A CARRIAGE RETURN IS THEN ISSUED AND CURSOR POSITION VERIFIED
                                        ;TO BE ROW1,COL0.

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

                                    ;*******************************************************************
010104  000004                      TST16:  SCOPE
010106  012737  000005  001156              MOV     #5,$TIMES       ;;DO 5 ITERATIONS
010114  012737  010122  001106              MOV     #NWLN,$LPADR    ;;SET SCOPE LOOP ADDRESS

010122  004037  016272              NWLN:   JSR     R0,RESETV       ;RESET THE UNIT AND ENTER MAINT.MODE
010126  013701  016262                      MOV     TBBUF,R1
010132  004037  020746                      JSR     R0,LDBUF        ;LOAD XMIT BUFFER WITH-
010136     033     131     040              .BYTE   .ESC,.Y,.R00,.C20
010141     064
010142     012     033     117              .BYTE   .LNFED,.ESC,.O,.RDCUR,.BEL,0
010145     131     007     000
010150  012737  000011  016270              MOV     #9.,XMCNT       ;SETUP TO XMIT 9 CHARACTERS
010156  004037  016676                      JSR     R0,XMREC        ;GO DO IT
010162  000402                              BR      30$             ;NORMAL EXIT.
010164  104011                              ERROR   11              ;TRANSMISSION CAUSED VT61 TO FAIL/HANG
010166  000454                              BR      4$              ;EXIT TEST
010170  023777  002172  005556  30$:       CMP     R01C20,@RBBUF   ;CHECK CURSOR POS. S/B ROW 1, COL 20.
010176  001412                              BEQ     3$
010200  005037  001124                      CLR     $GDDAT
010204  013737  002010  001126              MOV     LNFED,$BDDAT
010212  104001                              ERROR   1               ;ISSUE IT
010214  013746  002172                      MOV     R01C20,-(SP)    ;;PUSH R01C20 ON STACK
010220  004037  017342                      JSR     R0,CURER        ;SETUP AND ISSUE CURSOR ERROR
010224  013701  016262              3$:     MOV     TBBUF,R1
010230  013721  002006                      MOV     CARRT,(R1)+     ;LOAD XMIT BUFFER WITH
010234  013721  002110                      MOV     ESCOI,(R1)+     ;CARRIAGE RETURN, READ CURSOR
010240  013721  002104                      MOV     RDCUR,(R1)+     ;POSITION
010244  012737  000004  016270              MOV     #4,XMCNT        ;SET UP TO TRANSMIT 4 CHARACTERS
010252  004037  016676                      JSR     R0,XMREC        ;GO DO IT
010256  000402                              BR      40$             ;NORMAL EXIT.
010260  104011                              ERROR   11              ;TRANSMISSION CAUSED VT61 TO FAIL/HANG
010262  000416                              BR      4$              ;EXIT TEST
010264  023777  002170  005462  40$:       CMP     R01C00,@RBBUF   ;CHECK CURSOR POS. S/B ROW1, COL C.
010272  001412                              BEQ     4$              ;EXIT TEST IF GOOD.
010274  005037  001124                      CLR     $GDDAT
010300  013737  002006  001126              MOV     CARRT,$BDDAT
010306  104001                              ERROR   1               ;ISSUE IT
010310  013746  002170                      MOV     R01C00,-(SP)    ;;PUSH R01C00 ON STACK
010314  004037  017342                      JSR     R0,CURER        ;SET UP AND ISSUE CURSOR ERROR
010320  000240                      4$:     NOP

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

                                        ;ROUTINE TO VERIFY PROPER OPERATION OF ERASE TO END-OF-
                                        ;SCREEN. SCREEN IS WRITTEN TO 1920 INCREMENTING CHAR.
                                        ;ERASE TO END OF SCREEN IS THEN ISSUED AND THE
                                        ;ENTIRE SCREEN IS READ VERIFYING THAT IT IS ALL NULLS.
```

```
                                        ;*****************************************************************
                                        ;*****************************************************************
010322  000004                  TST17:  SCOPE
010324  012737  000003  001156          MOV     #3,$TIMES       ;;DO 3 ITERATIONS
010332  012737  010340  001106          MOV     #ERSE,$LPADR    ;;SET SCOPE LOOP ADDRESS

010340  004037  016272          ERSE:   JSR     R0,RESETV       ;RESET THE UNIT -SET MAINT. MODE.
010344  005077  005404                  CLR     @RBBUF          ;CLEAR THE CHECK LOCATION.
010350  004037  020250                  JSR     R0,DATSC        ;FILL THE SCREEN.
010354  013701  016262                  MOV     TBBUF,R1
010360  004037  020746                  JSR     R0,LDBUF        ;LOAD XMIT BUFFER WITH:
010364     033     110     033          .BYTE   .ESC,.CHOM,.ESC,.EOS,.ESC,.0,.XMTAL,0
010367     112     033     117
010372     126     000
010374  113737  002166  001125          MOVB    ESCN,$GDDAT+1
010402  113737  002030  001124          MOVB    EOS,$GDDAT      ;LOAD ERROR WITH ERASE TO EOS
010410  005037  001126                  CLR     $BDDAT
010414  005077  005334                  CLR     @RBBUF
010420  012737  000007  016270          MOV     #7,XMCNT        ;SET UP TO XMIT 7 BYTES
010426  004037  016676                  JSR     R0,XMREC        ;XMIT AND WAIT FOR REC. DONE
010432  000402                          BR      5$
010434  104011                          ERROR   11              ;ESC ERROR
010436  000413                          BR      ERSXT           ;EXIT TEST
010440  127737  005310  002224  5$:     CMPB    @RBBUF,ZERO     ;VT61 XMITTED SOM/EOM ONLY?
010446  001407                          BEQ     ERSXT           ;YES-EXIT TEST.
010450  104001                          ERROR   1               ;NO-ERASE TO END OF SCREEN
010452  004037  016654                  JSR     R0,CLREG        ;GO CLEAR ERROR STORAGE
010456  117737  005272  001126          MOVB    @RBBUF,$BDDAT
010464  104004                          ERROR   4               ;ISSUE DATA ERROR
010466  000240                  ERSXT:  NOP


                                        ;*****************************************************************
                                                ;ROUTINE TO SET UP END OF PASS INDICATION.
                                                ;SELF TEST(ESC O T) IS ISSUED TO THE UNIT UNDER TEST
                                                ;AND AN ERROR IS ISSUED IF THE UNIT CANNOT RESPOND AFTER
                                                ;SELF TEST IS COMPLETE. IF SELF TEST IS SUCCESSFUL THE
                                                ;SCREEN IS WRITTEN TO 23 LINES OF INCREMENTING CHARACTERS
                                                ;AND 23 LINES OF INCREMENTING CHAR. IN REVERSE VIDEO.
                                                ;THE IDENT IS THEN CHECKED AND IF A COPIER IS PRESENT A
                                                ;COPY SCREEN  COMMAND IS ISSUED(NOTE: THIS COMMAND WILL CAUSE
                                                ;THE UNIT TO BE "BUSY" AND NOT RESPOND TO ANY FURTHER COMMANDS
                                                ;UNTIL THE SCREEN HAS BEEN COMPLETELY COPIED.)
                                        ;*****************************************************************

                                        ;*****************************************************************
010470  000004                  TST20:  SCOPE
010472  012737  000001  001156          MOV     #1,$TIMES       ;;DO 1 ITERATION
010500  012737  010506  001106          MOV     #LSTST,$LPADR   ;;SET SCOPE LOOP ADDRESS

010506                          LSTST:
010506  013746  002224                  MOV     ZERO,-(SP)      ;;PUSH ZERO ON STACK
010512  013746  002156                  MOV     TSTER,-(SP)     ;;PUSH TSTER ON STACK
```

```
2494  010516  013746  002110                    MOV     ESCO,-(SP)      ;;PUSH ESCO ON STACK
2495  010522  004037  014370                    JSR     R0,TESC         ;TRANSMIT IT.
2496  010526  004037  016402                    JSR     R0,GETON        ;GO LOOK FOR A XON.
2497  010532  000407                            BR      1$              ;VT61RESPONDED-NOT HUNG
2498  010534  013737  001762  001124            MOV     VJCSR,$GDDAT    ;LOAD THE ADDRESS
2499  010542  013737  001772  001126            MOV     VECT,$BDDAT     ;LOAD THE VECTOR
2500  010550  104010                            ERROR   10              ;REPORT SELF TEST FAILURE
2501  010552  004037  016272          1$:       JSR     R0,RESETV       ;RESET AND SET MAINT. MODE.
2502  010556  005037  002244                    CLR     BUBCT           ;SET UP HALF-SCREEN FLAG.
2503  010562  042737  077577  002260  2$:       BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL.
2504  010570  013701  016262                    MOV     TBBUF,R1        ;SET UP BEG. OF XMIT BUFFER
2505  010574  012703  000500                    MOV     #320.,R3        ;FILL BUFFER WITH INCREMENTING CHAR.
2506  010600  004037  020222                    JSR     R0,BLDINC
2507  010604  012737  001001  002262            MOV     #1001,BLKM      ;SET XMIT TO SOM- DATA -EOM.
2508  010612  012737  001700  016270            MOV     #960.,XMCNT     ;SEND 12 LINE TO VT61
2509  010620  052777  040000  171134            BIS     #XENA,@VJCSR    ;ENABLE XMIT INTERRUPTS
2510  010626  012746  000001                    MOV     #XMDNE,-(SP)    ;;PUSH #XMDNE ON STACK
2511  010632  012746  000012                    MOV     #10.,-(SP)      ;;PUSH #10. ON STACK
2512  010636  004037  021654                    JSR     R0,WTBGND       ;LOOK FOR XMDNE.
2513  010642  000430                            BR      ENDSEL          ;NOT FOUND-EXIT.
2514  010644  005737  002244                    TST     BUBCT           ;DONE WITH SCREEN?
2515  010650  001007                            BNE     3$              ;YES-EXIT
2516  010652  012703  005672                    MOV     #SETREV,R3      ;NO-ISSUE ENTER REVERSE VIDEO
2517  010656  004037  017222                    JSR     R0,LDXMIT       ;ESCAPE SEQUENCE.
2518  010662  005237  002244                    INC     BUBCT           ;INCREMENT SCREEN HALF FLAG.
2519  010666  000735                            BR      2$              ;AND ISSUE SECOND HALF IN REV. VIDEO.
2520  010670  032737  000001  002160  3$:       BIT     #BIT00,IDENT    ;IDENT = COPIER?
2521  010676  001412                            BEQ     ENDSEL          ;NO
2522  010700  013746  002224                    MOV     ZERO,-(SP)      ;;PUSH ZERO ON STACK
2523  010704  012746  000135                    MOV     #.CPYSC,-(SP)   ;;PUSH #.CPYSC ON STACK
2524  010710  013746  002166                    MOV     ESCN,-(SP)      ;;PUSH ESCN ON STACK
2525  010714  004037  014370                    JSR     R0,TESC
2526  010720  004037  020754                    JSR     R0,CKABRT       ;CHECK FOR A PERIPHERAL ABORT.
2527  010724  105737  002232          ENDSEL:   TSTB    MODE            ;IF IN MAN MODE DO NOT ENTER EOP.
2528  010730  001402                            BEQ     ENDPS
2529  010732  000137  003332                    JMP     ASTRT
2530  010736  042777  000100  171016  ENDPS:    BIC     #RENA,@VJCSR    ;CLEAR REC.INT. BEFORE NEXT UNIT SELECT.
2531                                            ;*****************************************************************
2532
2533                                            .SBTTL  END OF PASS ROUTINE
2534
2535                                            ;*INCREMENT THE PASS NUMBER ($PASS)
2536                                            ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
2537                                            ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
2538                                            ;*IF SW12=1 INHIBIT TRACE TRAP
2539                                            ;*IF THERES A MONITOR GO TO IT
2540                                            ;*IF THERE ISN'T JUMP TO MODCA
2541
2542  010744                                    $EOP:
2543  010744  000004                                   SCOPE
2544  010746  005037  001102                           CLR     $TSTNM          ;;ZERO THE TEST NUMBER
2545  010752  005037  001156                           CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
2546  010756  005237  001100                           INC     $PASS           ;;INCREMENT THE PASS NUMBER
2547  010762  042737  100000  001100                   BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
2548  010770  005327                                   DEC     (PC)+           ;;LOOP?
2549  010772  000001                    $EOPCT: .WORD   1
```

```
                                                BGT     $DOAGN          ;;YES
                                                MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
                                        $ENDCT: .WORD   1
                                                $EOPCT
                                                TYPE    ,$ENDMG         ;;TYPE "END PASS #"
                                                MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
                                                TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
                                        $GET42:
                                                MOV     @#42,R0         ;;GET MONITOR ADDRESS
                                                BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
                                                RESET                   ;;MONITOR,CLEAR WORLD
                                                CLR     -(SP)           ;;INSURE THE "T" BIT IS CLEAR
                                                MOV     #$ENDAD,-(SP)   ;;SETUP FOR AN RTI OR RTT
                                                BR      $RTRN           ;;GO DO AN RTI OR RTT TO LOAD THE PSW
                                                                        ;;WITH A CLEARED "T" BIT

                                                MOV     @#42,R0         ;;GET MONITOR ADDRESS
                                                BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
                                                RESET                   ;;CLEAR THE WORLD
                                        $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
                                                NOP                     ;;SAVE ROOM
                                                NOP                     ;;FOR
                                                NOP                     ;;ACT11
                                        $DOAGN:
                                                CLR     -(SP)           ;;RESERVE A STACK LOC. FOR THE PS
                                                MOV     @#34,-(SP)      ;;SETUP THE TRAP VECTOR
                                                MOV     #1$,@#34        ;; TO GET THE PS
                                                TRAP
                                        1$:     TST     (SP)+           ;;CLEAN OFF THE USED PC
                                                MOV     (SP)+,2(SP)     ;;SAVE OFF THE PS
                                                MOV     (SP)+,@#34      ;;RESTORE TRAP VECTOR
                                                BIC     #20,(SP)        ;;CLEAR THE "T" BIT
                                                BIT     #BIT12,@SWR     ;; RUN WITH TRACE TRAP?
                                                BNE     2$              ;;BR IF NO
                                                COM     $TBIT           ;;IS IT TIME FOR TRACE TRAP
                                                BMI     2$              ;;BR IF NO
                                                BIS     #20,(SP)        ;;SET TRACE TRAP
                                        2$:     MOV     #$LOOP,-(SP)    ;;JUMP TO START OF TEST
                                        $RTRN:  RTI                     ;;RETURN--THIS IS CHANGED TO
                                                                        ;;AN "RTT" IF "RTT" IS LEGAL
                                                                        ;;INSTRUCTION
                                        $LOOP:
                                                JMP     @#MODCA         ;;RETURN
                                        $TBIT:  0
                                        $ENDMG: .ASCIZ  <15><12>/END PASS #/
                                        $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
                                        ;*****************************************************************
                                                ;ROUTINE TO ECHO THE KEYBOARD. KEYS FOR TAB,BELL,CARRIAGE
                                                ;AND LINE FEED ECHO A MNEMONIC,NON-DISPLAY CHAR. ECHO OCTAL
                                                ;EQUIVALENTS AND DISPLAY CHAR. ECHO THEMSELVES.
                                                ;(EXAMPLES-CHAR.,SPACE, ESC, SPACE OR 037, SPACE.) A
                                                ;CONTROL C (003) WILL CAUSE A TEST EXIT.
```

```
                                        ;*****************************************************************
                                        ;*****************************************************************
011174  000004                  TST21:  SCOPE
011176  012737  000001  001156          MOV     #1,$TIMES       ;;DO 1 ITERATION
011204  012737  011212  001106          MOV     #KYBD,$LPADR    ;;SET SCOPE LOOP ADDRESS

011212  004037  017262          KYBD:   JSR     R0,RESPTR
011216  012702  027122                  MOV     #DKYBD,R2       ;LOAD MESSAGE ADDRESS INR2
011222  004037  020316                  JSR     R0,DSMES        ;DISPLAY KEYBOARD MESSAGE
011226  012703  027510                  MOV     #DCNTZ,R3       ;ISSUE CONTROL C EXIT MESSAGE
011232  004037  017222                  JSR     R0,LDXMIT
011236  012703  011474                  MOV     #EXMAIN,R3
011242  004037  017222                  JSR     R0,LDXMIT       ;ISSUE EXIT MAINTENANCE MODE.
011246  042737  077577  002260  KYSTRT: BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL.
011254  105777  021040                  TSTB    @ABUFP          ;SEE IF A CHAR. RECEIVED
011260  001001                          BNE     11$             ;YES-GO PROCESS IT
011262  000001                          WAIT                    ;WAIT FOR A CH.
011264  117701  021030          11$:    MOVB    @ABUFP,R1       ;GET RAW RECEIVED DATA
011270  004037  021600                  JSR     R0,EXTST        ;CHECK FOR EXIT CONDITIONS
011274  000402                          BR      10$             ;NO EXIT -CONTINUE.
011276  000137  003332                  JMP     ASTRT           ;EXIT TEST 4
011302  105077  021012          10$:    CLRB    @ABUFP          ;CLEAR CHAR FROM BUFFER
011306  032737  000400  002260          BIT     #ESC,VSTAT      ;CHAR.=ESC(033)?
011314  001405                          BEQ     12$             ;NO
011316  005037  015762                  CLR     ESAMB           ;YES - RESET ESC ASSEMBLY FLAG
011322  012703  027115                  MOV     #DESC,R3        ;LOAD ESC MESSAGE ADDRESS
011326  000454                          BR      KYBXMT
011330  120127  000041          12$:    CMPB    R1,#41          ;CHAR. LESS THAN 41 OR
011334  103415                          BLO     2$              ;HIGHER THAN 176, GO ECHO
011336  120127  000176                  CMPB    R1,#176         ;OCTAL EQUIVALENT
011342  101012                          BHI     2$
011344  110177  004716                  MOVB    R1,@TBUFP       ;LOAD CHAR. IN XMIT BUFF.
011350  004037  017152                  JSR     R0,XMIT1        ;GO XMIT IT
011354  112777  000040  004704          MOVB    #40,@TBUFP      ;LOAD A SPACE
011362  004037  017152                  JSR     R0,XMIT1        ;AND XMIT IT.
011366  000727                          BR      KYSTRT
011370  120137  002004          2$:     CMPB    R1,BEL          ;CHAR.=BELL?
011374  001003                          BNE     3$
011376  012703  027371                  MOV     #DBELL,R3       ;LOAD BELL MESSAGE ADDRESS
011402  000426                          BR      KYBXMT
011404  120137  002012          3$:     CMPB    R1,TAB          ;CHAR. =TAB?
011410  001003                          BNE     4$
011412  012703  027352                  MOV     #DTAB,R3        ;YES-ECHO 'TAB'
011416  000420                          BR      KYBXMT
011420  123701  002006          4$:     CMPB    CARRT,R1        ;CHAR.=CARRIAGE RETURN?
011424  001003                          BNE     5$
011426  012703  027357                  MOV     #DCR,R3         ;YES - ECHO 'C/R'.
011432  000412                          BR      KYBXMT
011434  120137  002010          5$:     CMPB    R1,LNFED        ;CHAR.=LINE FEED?
011440  001003                          BNE     6$              ;NO CHECK FOR CONTROL Z
011442  012703  027364                  MOV     #DLF,R3         ;YES - ECHO 'L/F'.
011446  000404                          BR      KYBXMT
011450  004037  020412          6$:     JSR     R0,BINOCT       ;CONVERT BINARY TO OCTAL
011454  012703  002220                  MOV     #SVER1,R3
011460  042737  077577  002260  KYBXMT: BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL.
```

```
                                        JSR     R0,LDXMIT       ;GO XMIT BUFFER
                                        BR      KYSTRT          ;WAIT FOR NEXT CHAR.

                                        ;SEQUENCE TO EXIT MAINTENANCE MODE.
011474  033     117     141     EXMAIN: .BYTE   .ESC,.O,.DMAIN,0
011477  000

                                ;***************************************************************
                                        ;ROUTINE TO UTILIZE THE VT61 AS A PRINTER CONTROLLER.
                                        ;AFTER TEST MESSAGE IS DISPLAYED, THE TEST WAITS
                                        ;FOR A C/R BEFORE ACTUALLY ENTERING TEST. A PATTERN
                                        ;OF INCREMENTING, ROLLING CHAR. WILL BE OUTPUTTED UNTIL A
                                        ;CONTROL C(003) IS RECEIVED.

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

                                ;***************************************************************
011500  000004                  TST22:  SCOPE
011502  012737  000001  001156          MOV     #1,$TIMES       ;;DO 1 ITERATION
011510  012737  011516  001106          MOV     #TPRNT,$LPADR   ;;SET SCOPE LOOP ADDRESS

011516  012702  027554          TPRNT:  MOV     #DPRNT,R2       ;LOAD PRINTER MESSAGE ADDRESS
011522  004037  020316                  JSR     R0,DSMES        ;AND ISSUE IT
011526  012703  011474                  MOV     #EXMAIN,R3
011532  004037  017222                  JSR     R0,LDXMIT       ;ISSUE EXIT MAINTENANCE MODE.
011536  004037  020546                  JSR     R0,GTCR         ;GO SET CARRIAGE RETURN
011542                          3$:
011542  013746  002224                  MOV     ZERO,-(SP)      ;;PUSH ZERO ON STACK
011546  013746  002150                  MOV     EPNT,-(SP)      ;;PUSH EPNT ON STACK
011552  013746  002166                  MOV     ESCN,-(SP)      ;;PUSH ESCN ON STACK
011556  004037  014370                  JSR     R0,TESC
011562  013701  016262                  MOV     TBBUF,R1        ;LOAD R1 WITH XMIT BUFFER
011566  012705  000041          4$:     MOV     #41,R5          ;R5=1ST CHAR
011572  042737  077577  002260  5$:     BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL.
011600  013701  016262                  MOV     TBBUF,R1
011604  012703  000204                  MOV     #132.,R3                ;R3= LINE WIDTH
011610  004037  020226                  JSR     R0,BLDINA       ;GO BUILD A SLIDING PATTERN.
011614  013721  002006                  MOV     CARRT,(R1)+     ;LOAD A C/R AND L/F
011620  013721  002010                  MOV     LNFED,(R1)+
011624  012737  000206  016270          MOV     #134.,XMCNT     ;SET UP TO XMIT BY BYTES.
011632  052777  040000  170122          BIS     #XENA,@VJCSR
011640  032737  000001  002260          BIT     #XMDNE,VSTAT    ;WAIT FOR XMIT DONE
011646  001774                          BEQ     .-6
011650  004037  021600                  JSR     R0,EXTST        ;CHECK FOR EXIT REQUEST.
011654  000402                          BR      6$              ;NO-CONTINUE
011656  000137  003332                  JMP     ASTRT           ;YES-EXIT TEST!!
011662  004037  020754          6$:     JSR     R0,CKABRT       ;CHECK FOR A PERIPHERAL ABORT.
011666  122705  000177                  CMPB    #177,R5         ;EXCEEDED PATT. LIMIT?
011672  001337                          BNE     5$              ;NO
011674  000734                          BR      4$              ;YES RESET IT

                                ;***************************************************************
                                        ;ROUTINE TO LOOP DATA/COMMANDS FROM THE VT61 BACK TO
                                        ;THE VT61. DATA TRANSMISSIONS RESULTING FROM A ESC
                                        ;SEQUENCE WILL ALSO BE LOOPED AND WILL ENTER THE SCREEN
```

```
                                        ;AT THE CURSOR POSITION.THIS TEST CAN BE USED TO SIMULATE,
                                        ;OR CREATE, SPECIFIC SCREEN PATTERNS AND OPERATIONS.
                                ;*******************************************************************

                                ;*******************************************************************
011676  000004                  TST23:  SCOPE
011700  012737  000001  001156          MOV     #1,$TIMES       ;;DO 1 ITERATION
011706  012737  011714  001106          MOV     #LPTST,$LPADR   ;;SET SCOPE LOOP ADDRESS

011714  004037  017262          LPTST:  JSR     R0,RESPTR       ;RESET POINTERS
011720  012702  027377                  MOV     #DLOOP,R2       ;LOAD LOOP MESSAGE ADDRESS
011724  004037  020316                  JSR     R0,DSMES        ;DISPLAY IT
011730  012703  011474                  MOV     #EXMAIN,R3
011734  004037  017222                  JSR     R0,LDXMIT       ;ISSUE EXIT MAINTENANCE MODE.
01174C  004037  021350                  JSR     R0,LOOP         ;GO LOOP VT61
011744  000137  003332                  JMP     @STRT           ;ENTER MAN MODE VIA SCOPE ROUTINE.

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

                                        ;PRODUCTION KEYBOARD TEST.  ALL KEYS MUST BE DEPRESSED
                                        ;IN THE SEQUENCE INDICATED ON THE SCREEN.  ALL ERRORS
                                        ;OR MISTAKES ARE DISPLAYED IN OCTAL POSITIONAL FORMAT AND THE
                                        ;CORRECT KEY POSITION IN THE ROW  IS DISPLAYED IN DECIMAL.
                                        ;THIS TEST IS RUN IN MAINTENANCE MODE, THEREFORE THE KEYS
                                        ;WILL ECHO THEIR POSITION ,NOT THEIR INDICATED MNEMONIC. IO
                                        ;ERRORS WILL CAUSE AN AUTOMATIC EXIT FROM TEST.
                                ;*******************************************************************

                                ;*******************************************************************
011750  000004                  TST24:  SCOPE
011752  012737  000001  001156          MOV     #1,$TIMES       ;;DO 1 ITERATION
011760  012737  011766  001106          MOV     #PDKBD,$LPADR   ;;SET SCOPE LOOP ADDRESS


011766  012702  027772          PDKBD:  MOV     #DKBD   ,R2
011772  004037  020316                  JSR     R0      ,DSMES  ;DISPLAY KEYBOARD TEST MESSAGE.
011776  005037  002244                  CLR     @UBCT           ;CLEAR ERROR COUNT LOCATION.
012002  005005                          CLR     R5

012004  016504  012130          DOAROW: MOV     DTTBL(R5),R4    ;SET UP 'GOOD' CHAR. POINTER
012010  016503  012102                  MOV     MSTBL(R5),R3
012014  001414                          BEQ     FEXIT           ;MESSAGE WAS ZERO-EXIT.
012016  100421                          BMI     CLMAIN          ;IF MESSAGE IS -1,CLEAR MAINT. MODE.
012020  004037  017222                  JSR     R0,LDXMIT       ;ISSUE 'ROW OR FUNCTION' MESSAGE.
012024  004037  021102                  JSR     R0,CKKBD        ;GO CHECK IT.
012030  123727  002244  000012          CMPB    @UBCT,#10.      ;TEN ERROR EXIT?
012036  103401                          BLO     1$              ;NO-CONTINUE.
012040  000402                          BR      FEXIT           ;YES-EXIT TEST.
012042  005725                  1$:     TST     (R5)+           ;INCREMENT OFFSET.
012044  000757                          BR      DOAROW          ;NO-DO NEXT ROW/FUNCTION.
012046  012702  027541          FEXIT:  MOV     #DEXT,R2
012052  004037  020316                  JSR     R0,DSMES        ;ISSUE EXIT MESSAGE
012056  000137  003332                  JMP     @STRT
012062  012703  012076          CLMAIN: MOV     #RSMAIN,R3      ;SET UP TO EXIT MAINT. MODE.
012066  004037  017222                  JSR     R0,LDXMIT
012072  005725                          TST     (R5)+           ;INCREMENT OFFSET.
```

```
012074 000743                          BR      DOAROW          ;NOW TEST CONTROL AND SHIFT FUNCTIONS.
012076     033     117     141 RSMAIN: .BYTE   .ESC,.0,.DMAIN,0
012101     000
                                       ;TABLE OF MESSAGE ADDRESSES.
012102 030175 030302 030337 MSTBL:  .WORD   DTOP,DSEC,DTHRD,DBOT
012110 030466
012112 030544 030570 177777         .WORD   DSPCE,DKPD,-1,DCONT,DLSHFT,DRSHFT,0
012120 030416 030123 030227
012126 000000
012130 030771 031012 031032 DTTBL:  .WORD   ROW1,ROW2,ROW3,ROW4,SPCB
012136 031050 031072
012142 031074 000000 031066         .WORD   KYPD,0,CNTRA,SHFTA,SHFTA
012150 031070 031070
                                   ;*****************************************************************
                                   ;SUBROUTINE TO ALLOW SETUP FROM MULTIPLE ENTRIES
                                   ;*****************************************************************
012154                             SETA:
012154 012706 001100                    MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
012160 005026                           CLR     (R6)+           ;;CLEAR MEMORY LOCATION
012162 022706 001126                    CMP     #$BDDAT,R6      ;;DONE?
012166 001374                           BNE     .-6             ;;LOOP BACK IF NO
012170 012706 001100                    MOV     #STACK,SP       ;;SETUP THE STACK POINTER
012174 012737 021770 000020             MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
012202 012737 000340 000022             MOV     #340,@#IOTVEC+2 ;;LEVEL 7
012210 012737 022244 000030             MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
012216 012737 000340 000032             MOV     #340,@#EMTVEC+2 ;;LEVEL 7
012224 012737 023622 000034             MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
012232 012737 000340 000036             MOV     #340,@#TRAPVEC+2;LEVEL 7
012240 012737 023454 000024             MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
012246 012737 000340 000026             MOV     #340,@#PWRVEC+2 ;;LEVEL 7
012254 013737 011000 010772             MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
012262 005037 001156                    CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
012266 005037 001160                    CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
012272 112737 000001 001115             MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
012300 012737 011144 000014             MOV     #$RTRN,@#TBITVEC ;;SET "T" BIT VECTOR TO $RTRN
012306 012737 000340 000016             MOV     #340,@#TBITVEC+2 ;;LEVEL 7
012314 012737 000002 011144             MOV     #RTI,$RTRN      ;;SET $RTRN TO A RTI
012322 012737 012350 000010             MOV     #65$,@#RESVEC   ;;TRY TO DO A RTT
012330 005046                           CLR     -(SP)           ;;DUMMY PS
012332 012746 012340                    MOV     #64$,-(SP)      ;;AND PC
012336 000006                           RTT                     ;;TRY THE RTT
012340 012737 000006 011144 64$:        MOV     #RTT,$RTRN      ;;RTT IS LEGAL--SET $RTRN TO A RTT
012346 000402                           BR      66$
012350 062706 000010        65$:        ADD     #10,SP          ;;RTT ILLEGAL--CLEAN OFF THE STACK
012354 012737 000012 000010 66$:        MOV     #RESVEC+2,@#RESVEC ;;RESTORE TRAP CATCHER
012362 005037 011152                    CLR     $TBIT           ;;CLEAR "T" BIT SWITCH
012366 012737 012366 001106             MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
012374 012737 012374 001110             MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
012402 013746 000004                    MOV     @#4,-(SP)       ;;SAVE ERROR VECTOR
012406 013746 000006                    MOV     @#6,-(SP)
```

```
2830  012412  012737  012426  000004          MOV     #67$,4          ;;SET UP TIME OUT VECTOR
2831  012420  005777  166512                  TST     @SWR            ;;TRY TO REFERENCE HARDWARE SWR
2832  012424  000407                          BR      68$             ;;BRANCH IF NO TIMEOUT TRAP OCCURS
2833  012426  012737  000176  001136  67$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
2834  012434  012737  000174  001140          MOV     #DISPREG,DISPLAY ;;POINT TO SOFTWARE DISPLAY REG
2835  012442  022626                          CMP     (SP)+,(SP)+     ;;RESTORE STACK
2836  012444  012637  000006          68$:    MOV     (SP)+,@#6       ;;RESTORE ERROR VECTOR
2837  012450  012637  000004                  MOV     (SP)+,@#4
2838  012454  005227  177777                  INC     #-1             ;;FIRST TIME?
2839  012460  001017                          BNE     69$             ;;BRANCH IF NO
2840  012462  022737  011052  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
2841  012470  001413                          BEQ     69$             ;;BRANCH IF NO
2842  012472  104400  012500                  TYPE    ,70$            ;;TYPE ASCIZ STRING
2843  012476  000410                          BR      69$             ;;GET OVER THE ASCIZ
2844                          ;;70$:  .ASCIZ  <200>#MD-11-DZVTJ-A#<200>
2845  012520                  69$:
2846  012520  104400  023726                  TYPE    ,STUPM  ;ISSUE SET-UP MESSAGE.
2847  012524  012737  012544  000010          MOV     #TRPA,@#10      ;AND VECTOR
2848  012532  000230                          SPL     0               ;PROCESSOR IS 11/45?
2849  012534  012737  000004  020216          MOV     #4,PMULT        ;YES-DELAY MULTIPLIER = 4
2850  012542  000416                          BR      RTRP
2851
2852  012544  022626          TRPA:   POP2SP                          ;NO
2853  012546  012737  012570  000010          MOV     #TRPB,@#10      ;RELOAD TRAP ADDRESS
2854  012554  006737  002216                  SXT     CHRD            ;PROCESSOR IS 11/40 OR 35?
2855  012560  012737  000002  020216          MOV     #2,PMULT        ;YES-DELAY MULTIPLIER=2
2856  012566  000404                          BR      RTRP
2857
2858  012570  022626          TRPB:   POP2SP
2859  012572  012737  000001  020216          MOV     #1,PMULT        ;PROCESSOR MUST BE 11/05
2860  012600  012737  000012  000010  RTRP:   MOV     #12,@#10        ;RESTORE TRAP CATCHER
2861  012606  105737  002232                  TSTB    MODE            ;CHECK MODE FOR CORRECT EXIT.
2862  012612  001402                          BEQ     70$
2863  012614  000137  002332                  JMP     MANSA           ;EXIT TO MANUAL SELECT
2864  012620  000137  002276          70$:    JMP     AUTOA           ;EXIT TO AUTO MODE.
2865                  ;*****************************************************************
2866                  ;THIS SUBROUTINE WILL VERIFY EACH ADDRESS AS IT IS
2867                  ;ENTERED VIA THE KEYBOARD WHEN IN MANUAL MODE.
2868                  ;*****************************************************************
2869
2870  012624  020227  160000          TMNAD:  CMP     R2,#160000      ;INSURE ADDRESS IS IN RANGE.
2871  012630  103407                          BLO     BDEXT           ;ITS NOT-TYPE A ? AND EXIT.
2872  012632  012737  012646  000004          MOV     #BDEXTA,@#4     ;SET UP TRAP EXIT.
2873  012640  005712                          TST     @R2             ;CHECK THE ADDRESS.
2874  012642  000240                          NOP
2875  012644  000405                          BR      ALEXT           ;IF WE GOT THIS FAR ITS OK.
2876
2877  012646  022626          BDEXTA: POP2SP                          ;ADDRESS TRAPPED-PURGE IT, TYPE A
2878  012650  104400  027770          BDEXT:  TYPE    ,QMRK           ;QUESTION MARK, RESTORE TRAP
2879  012654  012700  002346                  MOV     #BLDADD,R0      ;LOCATION, AND EXIT TO BUILD THE
2880  012660  012737  000006  000004  ALEXT:  MOV     #6,@#4          ;NEXT ENTRY.
2881  012666  000200                          RTS     R0
2882
2883                  ;*****************************************************************
2884                  ;THIS ROUTINE MAPS ALL POSSIBLE DJ11 ADDRESSES AND STORES
2885                  ;THEM IN A TABLE (INTAB). ALL ADDRESSES WHICH DO NOT
```

```
                                        ;RESULT IN TIMEOUTS ARE STORED.
                                ;*****************************************************************
                                
012670  012701  000300          TRPVEC: MOV     #300,R1         ;START AT BEG. OF FLOATING VECTORS
012674  012702  000302                  MOV     #302,R2         ;
012700  012703  000004                  MOV     #4,R3           ;R3 CONTAINS IOT TRAP INST.
012704  010221                  1$:     MOV     R2,(R1)+        ;START LOADING ADDRESSES
012706  010321                          MOV     R3,(R1)+        ;LOAD THE TRAP
012710  062702  000004                  ADD     #4,R2           ;ASSUME 4 REGISTERS PER INTERFACE
012714  020127  001000                  CMP     R1,#1000        ;DONE?
012720  002771                          BLT     1$              ;NO CONTINUE LOADING TRAPS
012722  005037  000006                  CLR     @#6
012726  012737  012766  000004          MOV     #TPENT,@#4      ;SET UP TIME-OUT TRAP ADDRESS
012734  005001                          CLR     R1              ;CLEAR THE TABLE POINTER
012736  012705  001570                  MOV     #INTAB,R5       ;R5=DESTINATION TABLE
012742  013702  001756          FADD:   MOV     STRTAB,R2       ;PUT THE ADDRESS TO BE TESTED IN R2
012746  023702  001760          TRPE:   CMP     ENDTAB,R2       ;HAVE WE EXCEEDED END OF TABLE ADDRESS?
012752  103407                          BLO     TBLCK           ;YES GET NEXT BASE ADDRESS.
012754  005712                          TST     @R2             ;ADDRESS THE DEVICE IF POSSIBLE
012756  010225                          MOV     R2,(R5)+        ;IF WE GOT THIS FAR THERE IS A DEVICE THERE-SAVE IT
012760  062702  000010          FADD1:  ADD     #10,R2          ;INCREMENT TO THE NEXT POSSIBLE ADDRESS
012764  000770                          BR      TRPE            ;GO TEST THE NEXT ADDRESS
012766  022626                  TPENT:  POP2SP                  ;RESTORE THE STACK AND TEST
012770  000773                          BR      FADD1           ;NEXT ADDRESS
012772  005015                  TBLCK:  CLR     (R5)            ;SET UP TABLE TERMINATOR OF  ZEROS.
012774  000200                          RTS     R0
                                ;*****************************************************************
                                        ;THIS ROUTINE WILL INSURE THAT THE DEVICE(DJ11)
                                        ;WILL INTERRUPT WHEN XMIT INT. ENABLE BIT IS SET.
                                ;*****************************************************************

012776  005046                  CDEV:   CLR     -(SP)           ;CLEAR THE PSW,LSI11 STYLE.
013000  012746  013006                  MOV     #100$,-(SP)
013004  000002                          RTI
013006  012737  000004  000004  100$:   MOV     #4,@#4          ;INSTALL IOT TRAP INST. AT LOCATION 4.
013014  012737  013132  000020          MOV     #TDEV,@#IOTVEC  ;SET UP IOT TRAP EXIT ADDRESS
013022  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;SET PSW TO 7-ALLOW NO OTHER INTERRUPTS
013030  000005                          RESET                   ;INSURE ALL XMIT FLAGS HIGH.
013032  012703  001550                  MOV     #VVECT,R3       ;VECTOR STORAGE ADDRESS SET
013036  012702  001570                  MOV     #INTAB,R2       ;PRIMARY DEVICE TABLE ADDRESS SET
013042  012704  001630                  MOV     #DJLNE,R4       ;DJ11 LINE TABLE SET.
013046  012705  001560                  MOV     #DJTBL,R5       ;DEVICE TABLE ADDRESS SET.
013052  012701  001762          CDEVA:  MOV     #VJCSR,R1       ;VT61 DEVICE ADDRESS SET.
013056  005712                          TST     (R2)            ;CHECKED ALL DEVICES?
013060  001574                          BEQ     AOUT            ;YES-EXIT
013062  100403                          BMI     1$              ;INSURE ADDRESS IS IN PROPER RANGE(17XXXX)

013064  062702  000002                  ADD     #2,R2           ;ADDRESS IS DEFINITELY NOT GOOD -PURGE
013070  000770                          BR      CDEVA           ;AND LOOK FOR ANOTHER.
013072  004037  014076          1$:     JSR     R0,LDADD        ;LOAD NEXT ADDRESSES TO BE CHECKED
013076  012701  001200                  MOV     #1200,R1        ;NOW USE R1 AS FAILSAFE COUNTER
013102  052777  177777  166656          BIS     #177777,@VXTCR  ;ENABLE ALL LINES TO BE SCANNED.
013110  052777  000400  166644          BIS     #XSCN,@VJCSR    ;ENABLE THE XMITTER SCANNER.
013116  052777  040000  166636          BIS     #XENA,@VJCSR    ;SET XMIT ENABLE
013124  005301                          DEC     R1              ;IF DEVICE DOES NOT INTERRUPT WITHIN
013126  001376                          BNE     .-2             ;APPROX. 200US IT IS NOT A DJ11.
```

```
                                        BR      CDEVA           ;THEREFORE, GO TRY ANOTHER DEVICE.
013132  042777  040000  166622  TDEV:   BIC     #XENA,@VJCSR    ;CLEAR XMIT ENABLE.
013140  042777  000400  166614          BIC     #XSCN,@VJCSR    ;DISABLE THE XMIT SCANS.
013146  005077  166614                  CLR     @VXTCR          ;CLEAR XMIT LINE SCAN REG.
013152  162716  000010                  SUB     #10,(R6)        ;RESET TO RECEIVER VECTOR ADDRESS
013156  012613                          MOV     (R6)+,(R3)      ;STORE IT IN VECTOR TABLE(VVECT).
013160  005726                          TST     (R6)+           ;POP THE OLD PSW AND DISCARD
013162  022626                          POP2SP                  ;POP THE ADD. AND PSW PRIOR TO INTERRUPT.
                                ;*****************************************************************
                                ;THIS ROUTINE IS A QUICK TEST OF ANY DJ11 ENCOUNTERED
                                ;A DATA PATTERN WILL BE RUN ON ALL ENTRIES IN INTAB
                                ;*****************************************************************
013164  005046                          CLR     -(SP)           ;CLEAR THE PSW,LSI11 STYLE.
013166  012746  013174                  MOV     #100$,-(SP)
013172  000002                          RTI
013174  012301                  100$:   MOV     (R3)+,R1        ;GET THE RECEIVE VECTOR ADDRESS
013176  012721  014730                  MOV     #RECAD,(R1)+    ;AND STORE SAME.
013202  012721  000340                  MOV     #340,(R1)+      ;SET RECEIVE PSW TO 7.
013206  012721  015022                  MOV     #TSMAD,(R1)+    ;STORE THE XMIT VECTOR ADDRESS
013212  012711  000340                  MOV     #340,(R1)       ;SET XMIT PSW TO 7.
013216  012737  000001  001774          MOV     #1,TSTLNE       ;SET UP TO TEST LINE 0 FIRST.
013224  005037  020072                  CLR     HDFLG
013230  012737  000001  002246  LNECK:  MOV     #BIT00,TPREG    ;TPREG IS NOW DATA PATTERN OF 1.
013236  005001                          CLR     R1              ;SET UP FAILSAFE DELAY.
013240  052777  000010  166514          BIS     #MCLR,@VJCSR    ;CLEAR SILO AND UARTS.
013246  000240                          NOP                     ;CLEAR PROPOGATION TIME
013250  013777  001774  166510          MOV     TSTLNE,@VXTCR   ;LOAD THE LINE TO BE CHECKED.
013256  052777  000401  166476          BIS     #SCAN,@VJCSR    ;ENABLE XMIT AND RECIEVE SCANNERS.
013264  052777  000100  166470          BIS     #RENA,@VJCSR    ;SET RECEIVE ENABLE.
013272  052777  040004  166462          BIS     #TCOMB,@VJCSR   ;ENABLE XMIT INT. AND MAINT MODE.
013300  105737  002246          1$:     TSTB    TPREG           ;SEE IF COMPLETE PATTERN XMITTED.
013304  001436                          BEQ     GDAD            ;YES GO STORE THIS ADDRESS
013306  005301                          DEC     R1              ;CYCLE TIMEOUT DELAY
013310  001373                          BNE     1$              ;NOT YET 'TIMEOUT',KEEP CYCLING.
013312  042777  040000  166442          BIC     #XENA,@VJCSR    ;CLEAR INTERRUPT ENABLES ON BAD UNITS.
013320  042777  000100  166434          BIC     #RENA,@VJCSR
013326  005737  020072                  TST     HDFLG           ;ADDRESSES ISSUED?
013332  001010                          BNE     2$              ;YES-ISSUE LINES.
013334  104400  024202                  TYPE    ,DLERR          ;ISSUE DJ11 FAILURE MESSAGE.
013340  013746  001762                  MOV     VJCSR,-(SP)     ;;SAVE VJCSR FOR TYPEOUT
                                                                ;;TYPE BD. ADDRESS
013344  104402                          TYPOS                   ;;GO TYPE--OCTAL ASCII
013346     006                          .BYTE   6               ;;TYPE 6 DIGIT(S)
013347     001                          .BYTE   1               ;;TYPE LEADING ZEROS
013350  104400  001167                  TYPE    ,$CRLF
013354  004037  020510          2$:     JSR     R0,CONVLN       ;CONVERT BINARY LINE TO OCTAL LINE.
013360  000337  001776                  SWAB    OCTLNE          ;MOVE  OCTAL LINE # INTO POSITION.
013364  013746  001776                  MOV     OCTLNE,-(SP)    ;;SAVE OCTLNE FOR TYPEOUT
                                                                ;;TYPE A BAD LINE
013370  104402                          TYPOS                   ;;GO TYPE--OCTAL ASCII
013372     003                          .BYTE   3               ;;TYPE 3 DIGIT(S)
013373     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
013374  005237  020072                  INC     HDFLG
013400  000402                          BR      BDAD            ;GO TRY ANOTHER LINE OR ADDRESS.
013402  013724  001774          GDAD:   MOV     TSTLNE,(R4)+    ;STORE THE GOOD LINE
013406  006337  001774          BDAD:   ASL     TSTLNE          ;SET UP TO CHECK NEXT LINE.
```

```
 2998  013412  001306                        BNE     LNECK           ;IF NOT ZERO-GO CHECK IT.
 2999  013414  012724  177777                MOV     #-1,(R4)+       ;STORE DJ11 LINE SEPARATOR.
 3000  013420  013725  001762                MOV     VJCSR,(R5)+     ;STORE THE DJ11 ADDRESS.
 3001  013424  005037  020072                CLR     HDFLG           ;SET UP TO TYPE AN ADDRESS.
 3002  013430  042777  000401  166324        BIC     #SCAN,@VJCSR    ;DISABLE REC. AND XMIT SCANNERS.
 3003  013436  052777  000010  166316        BIS     #MCLR,@VJCSR    ;CLEAR SILO AND UARTS
 3004  013444  104400  001167                TYPE    ,$CRLF
 3005  013450  000600                        BR      CDEVA           ;;CHECK ANOTHER DJ11
 3006  013452  005015                AOUT:   CLR     (R5)            ;SET A ZERO TABLE TERMINTATOR.
 3007  013454  005014                        CLR     (R4)            ;SET THE LINE TABLE TERMINATOR
 3008  013456  012737  000006  000004        MOV     #6,@#4          ;RESTORE LOCATION 4 TO HALT CONDITION
 3009  013464  005037  000006                CLR     @#6             ;TO CATCH ERRORS AND ILLEGAL INTERRUPTS.
 3010  013470  012737  021770  000020        MOV     #$SCOPE,@#IOTVEC        ;RELOAD IOT VECTOR FOR SCOPE
 3011  013476  012737  000340  000022        MOV     #340,@#IOTVEC+2 ;LOOP.
 3012  013504  012701  000300                MOV     #300,R1
 3013  013510  012702  000302                MOV     #302,R2
 3014  013514  010221                1$:     MOV     R2,(R1)+
 3015  013516  005021                        CLR     (R1)+           ;RESTORE HALTS TO ALL LOCATIONS CONTAINING IOTS
 3016  013520  062702  000004                ADD     #4,R2
 3017  013524  020127  001000                CMP     R1,#1000        ;TO LOCATION 1000
 3018  013530  103771                        BLO     1$
 3019  013532  000005                        RESET                   ;CLEAR ALL FLAGS
 3020  013534  000200                        RTS     R0
 3021
 3022                                ;*****************************************************************
 3023                                ;INITIALIZATION ROUTINE FOR AUTO SELECTION. THIS ROUTINE
 3024                                ;WILL INSURE THAT ALL DJ11S IN DJTBL HAVE A VT61 CONNECTED
 3025                                ;ALL UNITS WHICH EITHER DO NOT OR DO NOT RESPOND WILL BE PURGED.
 3026                                ;*****************************************************************
 3027  013536  012702  001560        INITA:  MOV     #DJTBL,R2       ;R2 POINTS TO DJ11 ADDRESS TABLE
 3028  013542  012703  001630                MOV     #DJLNE,R3       ;POINTER TO DJ11 LINES.
 3029  013546  012701  001762        11$:    MOV     #VJCSR,R1       ;POINTER TO VT61 DJ11
 3030  013552  052777  000010  166202        BIS     #MCLR,@VJCSR    ;CLEAR SILO AND UARTS.
 3031  013560  000240                        NOP                     ;CLEAR PROPOGATION TIME
 3032
 3033  013562  005712                        TST     (R2)            ;SEE IF ALL CHECKED
 3034  013564  001456                        BEQ     INTXT           ;YES-EXIT
 3035  013566  004037  014076                JSR     R0,LDADD        ;NO-GO LOAD THE ADDRESSES
 3036  013572  022713  177777        12$:    CMP     #-1,(R3)        ;AT A LINE TABLE TERMINATOR?
 3037  013576  001004                        BNE     13$             ;NO-CONTINUE TESTING THIS ADDRESS.
 3038  013600  005723                        TST     (R3)+           ;IT IS-BUMP THE POINTER AND GET NEXT
 3039  013602  005077  166154                CLR     @VJCSR          ;SHUT DOWN DJ11 AFTER SAMPLING COMPLETE.
 3040  013606  000757                        BR      11$             ;DJ11 ADDRESS(IF ANY).
 3041  013610  011337  001774        13$:    MOV     (R3),TSTLNE     ;LOAD LINE TO BE TESTED.
 3042  013614  004037  020510                JSR     R0,CONVLN       ;CONVERT BINARY LINE TO OCTAL LINE.
 3043  013620  012377  166142                MOV     (R3)+,@VXTCR    ;ENABLE LINE TO BE SCANNED.
 3044  013624  052777  000401  166130        BIS     #SCAN,@VJCSR    ;ENABLE XMIT AND REC SCANNERS.
 3045  013632  004037  016452                JSR     R0,ZFLAG        ;ISSUE ESCZ AND LOOK FOR RESPONSE.
 3046  013636                        2$:
 3047  013636  012637  002216                MOV     (SP)+,CHRD      ;;POP STACK INTO CHRD
 3048  013642  100414                        BMI     5$              ;TIMEOUT OCCURRED NO CHARACTER
 3049  013644  123727  002216  000140        CMPB    CHRD,#140       ;CHECK IDENT FOR VT61 IDENTIFIERS
 3050  013652  103410                        BLO     5$              ;NOT A VT61-SET UP TO PURGE ADDRESS
 3051  013654  123727  002216  000172        CMPB    CHRD,#172       ;IDENTS ARE SMALL A THRU Z
 3052  013662  101004                        BHI     5$              ;NOT A VT61-PURGE
 3053  013664                        4$:
```

```
3054  013664  012637  002216                    MOV     (SP)+,CHRD      ;;POP STACK INTO CHRD
3055  013670  001375                            BNE     4$
3056  013672  000737                            BR      12$             ;TEST ANOTHER ADDRESS
3057  013674                            5$:
3058  013674  012637  002216                    MOV     (SP)+,CHRD      ;;POP STACK INTO CHRD
3059  013700  001375                            BNE     5$
3060  013702  162703  000002                    SUB     #2,R3           ;RESET LINE POINTER.
3061  013706  010346                            MOV     R3,-(SP)        ;;PUSH R3 ON STACK
3062  013710  012746  000001                    MOV     #1,-(SP)        ;;PUSH #1 ON STACK
3063  013714  004037  014316                    JSR     R0,BBLUP
3064  013720  000724                            BR      12$             ;TRY ANOTHER DJ11 LINE.
3065  013722  022737  177777  001630  INTXT:   CMP     #-1,DJLNE       ;GOOD LINE FROM 1ST ADDRESS?
3066  013730  001021                            BNE     EXINT           ;YES-BEGIN TESTING.
3067  013732  022737  177777  001632            CMP     #-1,DJLNE+2     ;SEE IF SECOND DJ HAS GOOD LINES.
3068  013740  001403                            BEQ     NOUNIT          ;NO GOOD LINES ON 2ND DJ11.
3069  013742  005737  001632                    TST     DJLNE+2         ;ZERO TERMINATOR FOUND?
3070  013746  001012                            BNE     EXINT           ;NO-FIRST DJ11 HAS GOOD LINES.
3071  013750  104400  024113            NOUNIT: TYPE    ,NOVT           ;NO-ISSUE NO VT61 MESSAGE.
3072  013754  012737  005670  020220            MOV     #3000.,DCOUNT   ;SET DELAY TO 30 SEC.
3073  013762  004037  020156                    JSR     R0,DELAY        ;AND DO IT.
3074  013766  062700  000004                    ADD     #4,R0           ;SET UP 'NO VT61 FOUND' EXIT
3075  013772  000200                            RTS     R0
3076  013774  012702  001560            EXINT:  MOV     #DJTBL,R2       ;LOAD AND ISSUE GOOD ADDRESSES
3077  014000  005712                            TST     (R2)            ;INSURE A GOOD ADDRESS.
3078  014002  001762                            BEQ     NOUNIT          ;NONE FOUND-EXIT
3079  014004  012703  001630                    MOV     #DJLNE,R3       ;LOAD TABLE ADDRESS OF GOOD LINES.
3080  014010  104400  024042                    TYPE    ,DUNTST         ;OF RESPONSIVE VT61S.
3081  014014                            1$:
3082  014014  012246                            MOV     (R2)+,-(SP)     ;;SAVE (R2)+ FOR TYPEOUT
3083                                            ;;TYPE AN ADDRESS
3084  014016  104402                            TYPOS                   ;;GO TYPE--OCTAL ASCII
3085  014020     006                            .BYTE   6               ;;TYPE 6 DIGIT(S)
3086  014021     001                            .BYTE   1               ;;TYPE LEADING ZEROS
3087  014022  104400  001167                    TYPE    ,$CRLF
3088  014026  012337  001774            2$:     MOV     (R3)+,TSTLNE    ;LOAD A LINE TO PRINT
3089  014032  022737  177777  001774            CMP     #-1,TSTLNE      ;IF LINE IS A TERMINATOR, DO NOT
3090  014040  001411                            BEQ     3$              ;DISPLAY - GO SET UP NEXT DJ11.
3091  014042  004037  020510                    JSR     R0,CONVLN       ;CONVERT IT TO A OCTAL #.
3092  014046  000337  001776                    SWAB    OCTLNE          ;MOVE IST INTO PRINT POSITION.
3093  014052  013746  001776                    MOV     OCTLNE,-(SP)    ;;SAVE OCTLNE FOR TYPEOUT
3094                                            ;;TYPE A GOOD LINE
3095  014056  104402                            TYPOS                   ;;GO TYPE--OCTAL ASCII
3096  014060     003                            .BYTE   3               ;;TYPE 3 DIGIT(S)
3097  014061     000                            .BYTE   0               ;;SUPPRESS LEADING ZEROS
3098  014062  000761                            BR      2$              ;NO-TYPE ANOTHER LINE.
3099  014064  104400  001167            3$:     TYPE    ,$CRLF
3100  014070  005712                            TST     (R2)            ;AT END OF GOOD UNITS?
3101  014072  001350                            BNE     1$              ;NO PRINT ANOTHER ADDRESS.
3102  014074  000200                            RTS     R0
3103
3104                            ;*****************************************************************
3105                                    ;SUBROUTINE TO LOAD 4 ADDRESSES FROM THE LOCATION AT (R2).
3106                                    ;TO A LOCATION POINTED TO BY R1.EXITS WITH R2 INC. BY 2.
3107                            ;*****************************************************************
3108
3109
```

```
                                                
014076  012211                  LDADD:  MOV     (R2)+,(R1)      ;LOAD THE ADDRESS
014100  012111                  1$:     MOV     (R1)+,(R1)      ;STORE AN ADDRESS
014102  062711  000002                  ADD     #2,(R1)         ;INCREMENT THE ADDRESS
014106  020127  001770                  CMP     R1,#VXBUF       ;LOADED 4?
014112  002772                          BLT     1$              ;NO LOAD ANOTHER
014114  000200                          RTS     R0              ;YES-EXIT
                                        
                                ;***************************************************************
                                        ;ROUTINE TO RECEIVE CHARACTER(S).  ENTERED WITH
                                        ;NUMBER OF CHARACTERS TO RECEIVE ON THE STACK.
                                        ;ROUTINE EXITS WITH CARACTER(S) ON STACK.  IF A
                                        ;PROGRAM TIME-OUT (100 M.S.) OCCURS BEFORE A CHARACTER
                                        ;IS RECEIVED ROUTINE EXITS WITH -1 ON STACK. FORMAT
                                        ;FOR DATA IS (BYTE2, BYTE1) ETC.  A WORD OF ZEROS TERMINATES
                                        ;DATA STRING ON THE STACK. SOM/EOM, IF SENT, ARE RECEIVED
                                        ;BUT NOT STORED.
                                        
                                ;***************************************************************
                                        
014116                          RECTM:
014116  012637  002256                  MOV     (SP)+,ROSVE     ;;POP STACK INTO ROSVE
014122  012637  002244                  MOV     (SP)+,@#BUBCT   ;;POP STACK INTO @#BUBCT
014126  013746  002224                  MOV     ZERO,-(SP)      ;;PUSH ZERO ON STACK
014132  005037  002216          1$:     CLR     CHRD            ;CLEAR CHARACTER STORAGE LOCATION.
014136  005037  002254                  CLR     DLAY            ;SET UP FAILSAFE DELAY
014142  032777  000200  165612  3$:     BIT     #RECDN,@VJCSR   ;SEE IF DONE FLAG SET
014150  001007                          BNE     4$
014152  005337  002254                  DEC     DLAY            ;DECREMENT FAILSAFE CNTR.
014156  001371                          BNE     3$              ;NOT AT ZERO-CONTINUE WAITING.
014160  012737  177777  002216  31$:    MOV     #-1,CHRD        ;SET UP FOR FAILSAFE EXIT.
014166  000446                          BR      RECEX           ;EXIT ROUTINE.
014170  017737  165570  002216  4$:     MOV     @VRBUF,CHRD     ;STORE THIS CHARACTER.
014176  042737  170200  002216          BIC     #170200,CHRD    ;STRIP ALL BUT CHAR. AND LINE #.
014204  123737  001777  002217          CMPB    OCTLNE+1,CHRD+1 ;RECEIVED FROM CORRECT LINE?
014212  001347                          BNE     1$              ;NO-IGNORE THIS CHAR.
014214  122737  000057  002216          CMPB    #SLSH,CHRD      ;RECEIVED A IDENT SLASH(57)?
014222  001007                          BNE     41$             ;NO-STORE A CHARACTER.
014224  105337  002245                  DECB    BUBCT+1         ;DECREMENT ALLOWABLE SLASH COUNT.
014230  001753                          BEQ     31$             ;COUNT EQUAL ZERO-SET UP ERROR EXIT.
014232  123727  002245  000213          CMPB    BUBCT+1,#139.   ;RECEIVED FIRST SLASH?
014240  103734                          BLO     1$              ;YES-IGNORE THIS ONE.
014242  122737  000002  002216  41$:    CMPB    #SOM,CHRD       ;IS CHAR. ACTUALLY SOM?
014250  001003                          BNE     5$              ;NO
014252  105237  002244                  INCB    BUBCT           ;YES -SET UP TO RECEIVE EOM ALSO
014256  000725                          BR      1$              ;AND RECEIVE NEXT CHAR.
014260  122737  000004  002216  5$:     CMPB    #EOM,CHRD       ;CHAR. = EOM?
014266  001410                          BEQ     RECEXA          ;YES- DO NOT PUSH IT ON STACK
014270  105337  002244                  DECB    BUBCT           ;DECREMENT CHARACTER COUNT.
014274  001403                          BEQ     RECEX           ;COUNT=0. EXIT WERE DONE.
014276  013746  002216                  MOV     CHRD,-(SP)      ;;PUSH CHRD ON STACK
014302  000713                          BR      1$              ;GO READ AGAIN.
                                        
014304                          RECEX:
014304  013746  002216                  MOV     CHRD,-(SP)      ;;PUSH CHRD ON STACK
014310                          RECEXA:
```

```
3166  014310  013746  002256                MOV     ROSVE,-(SP)     ;;PUSH ROSVE ON STACK
3167  014314  000200                        RTS     R0
3168
3169
3170                                ;*****************************************************************
3171                                        ;THIS ROUTINE WILL 'BUBBLE UP' XX WORDS TO
3172                                        ;ELIMINATE NON-RESPONSIVE ADDRESSES.  ENTERED
3173                                        ;WITH ADDRESS TO BE 'BUBBLED' TO ON THE STACK.  LOCATIONS
3174                                        ;ELIMINATED WILL BE FILLED WITH ZEROS.  THE STACK MUST ALSO
3175                                        ;BE LOADED WITH THE NUMBER OF POSITIONS TO BUBBLE.
```

```
                                        ;*****************************************************************
                                        BBLUP:
014316 012637 002256                            MOV     (SP)+,ROSVE     ;;POP STACK INTO ROSVE
014322 012637 002244                            MOV     (SP)+,BUBCT     ;;POP STACK INTO BUBCT
014326 012637 002242                            MOV     (SP)+,TOADD     ;;POP STACK INTO TOADD
014332 010446                                   MOV     R4,-(SP)        ;;PUSH R4 ON STACK
014334 013704 002242                    2$:     MOV     @#TOADD,R4      ;PUT LAST GOOD DJ11 ADDRESS IN R4
014340 012437 002216                            MOV     (R4)+,CHRD      ;MOVE NEXT WORD TO CHRD FOR STORAGE
014344 012464 177774                    1$:     MOV     (R4)+,-4(R4)    ;BUBBLE UP DATA.
014350 001375                                   BNE     1$              ;BUBBLE UNTIL ZERO BYTE MOVED.
014352 005337 002244                            DEC     BUBCT           ;SUBTRACT ONE FROM BUBBLE COUNT.
014356 001366                                   BNE     2$              ;IF BUBBLE COUNT NOT ZERO - DO AGAIN.
014360                                  3$:
014360 012604                                   MOV     (SP)+,R4        ;;POP STACK INTO R4
014362 013746 002256                            MOV     ROSVE,-(SP)     ;;PUSH ROSVE ON STACK
014366 000200                                   RTS     R0              ;YES-EXIT

                                        ;*****************************************************************
                                                ;THIS ROUTINE OUTPUTS THE ESC SEQNENCE FOUND ON
                                                ;THE STACK. A WORD OF ZEROS MUST TERMINATE THE SEQUENCE.
                                                ;FORMAT FOR STACK WORD IS SEQ-ESC, IE-XXX033.
                                        ;*****************************************************************

014370                                  TESC:
014370 012637 002256                            MOV     (SP)+,ROSVE     ;;POP STACK INTO ROSVE
014374 010437 002244                            MOV     R4,BUBCT        ;SAVE R4.
014400 112777 000002 165362                     MOVB    #SOM,@VXBUF     ;SEND A START OF MESSAGE.
014406 012705 177777                    1$:     MOV     #-1,R5          ;ALL ONES TO THE CHECK LOCATION.
014412 012604                                   MOV     (R6)+,R4        ;GET COMMAND FROM STACK.
014414 001415                                   BEQ     3$              ;IF ZERO TERMINATOR FOUND-EXIT.
014416 110405                                   MOVB    R4,R5           ;LOAD CHECK BYTE.
014420 105704                           2$:     TSTB    R4              ;CHECK BYTE FOR A ZERO.
014422 001406                                   BEQ     20$             ;IF ZERO+DO NOT XMIT IT.
014424 032777 100000 165330                     BIT     #TRDY,@VJCSR
014432 001774                                   BEQ     .-6             ;WAIT FOR XMIT READY BIT
014434 110477 165330                            MOVB    R4,@VXBUF       ;XMMIT A BYTE.
014440 000304                           20$:    SWAB    R4              ;GET THE OTHER BYTE .
014442 120405                                   CMPB    R4,R5           ;IF GOOD COMPARE WE HAVE CHECKED BOTH
014444 001760                                   BEQ     1$              ;BYTES SO POP ANOTHER WORD.
014446 000764                                   BR      2$              ;GO XMIT ANOTHER BYTE
014450 032777 100000 165304             3$:     BIT     #TRDY,@VJCSR    ;SEE IF READY SET
014456 001774                                   BEQ     .-6
014460 012777 000004 165302                     MOV     #EOM,@VXBUF     ;SEND A EOM.
```

```
3220  014466  032777  100000  165266          BIT     #TRDY,@VJCSR    ;SEE IF READY SET
3221  014474  001774                          BEQ     .-6
3222  014476  013704  002244                  MOV     BUBCT,R4        ;RESTORE R4.
3223  014502  013746  002256                  MOV     ROSVE,-(SP)     ;;PUSH ROSVE ON STACK
3224  014506  000200                          RTS     R0
3225
3226                                  ;*****************************************************************
3227                                          ;ROUTINE TO READ A CHARACTER FROM THE CONSOLE.
3228                                          ;EXITS WITH CHARACTER ON THE STACK.
3229                                  ;*****************************************************************
3230
3231  014510                          CONRD:
3232  014510  012637  002256                  MOV     (SP)+,ROSVE     ;;POP STACK INTO ROSVE
3233  014514  032777  000200  164420          BIT     #RECDN,@$TKS    ;LOOK FOR DONE BIT
3234  014522  001774                          BEQ     .-6             ;WAIT FOR IT
3235  014524  117746  164414                  MOVB    @$TKB,-(R6)     ;PUSH CHARACTER TO STACK
3236  014530  042716  000200                  BIC     #200,(R6)       ;STRIP ANY PARITY BIT.
3237  014534  013746  002256                  MOV     ROSVE,-(SP)     ;;PUSH ROSVE ON STACK
3238  014540  000200                          RTS     R0
3239
3240                                  ;*****************************************************************
3241                                          ;MANUAL TEST SELECT MONITOR
3242                                          ;SELECTS TESTS TO BE EXECUTED FROM THOSE ENTERED IN
3243                                          ;INITIAL DIALOGUE. IF TEST 377 WAS REQUESTED THE TESTS WILL
3244                                          ;REPEAT INFINITELY.
3245                                  ;*****************************************************************
3246
3247  014542  105737  002232          MONIT:  TSTB    MODE            ;TEST MODE SWITCH
3248  014546  001012                          BNE     1$              ;MANUAL MODE
3249  014550  023737  002234  002236          CMP     FTLCNT,ALWCNT   ;COMPARE FATAL XMITS WITH ALLOWED.
3250  014556  103405                          BLO     100$            ;FATALS LESS THAN ALLOWED-CONTINUE.
3251  014560  104400  025741                  TYPE    ,DABRT          ;ISSUE ABORT MESSAGE.
3252  014564  000005                  200$:   RESET                   ;CLEAR ALL INTERFACE FLAGS.
3253  014566  000137  012154                  JMP     SETA            ;SET UP TO RESTART TEST.
3254  014572  000200                  100$:   RTS     R0              ;AUTO MODE
3255  014574  005726                  1$:     TST     (R6)+           ;POP THE STACK
3256  014576  022626                          POP2SP                  ;POP SCOPE RETURN AND VECTOR
3257  014600  005037  002234                  CLR     FTLCNT          ;DO NOT INC. FATAL COUNT IN MANUAL MODE.
3258  014604  032777  000200  164330  10$:    BIT     #RECDN,@$TKS    ;CONSOLE ACTIVE?
3259  014612  001407                          BEQ     11$
3260  014614  117701  164324                  MOVB    @$TKB,R1        ;STORE INPUT BUFFER
3261  014620  042701  000200                  BIC     #200,R1         ;CLEAR THE PARITY BIT
3262  014624  122701  000003                  CMPB    #3,R1           ;CHAR. EQUAL ESC. C?
3263  014630  001755                          BEQ     200$            ;YES-EXIT
3264  014632  117701  165372          11$:    MOVB    @TSTPTR,R1      ;GET THE NEXT TEST #
3265  014636  001005                          BNE     2$              ;NOT AT END OF LIST
3266  014640  042777  000100  165114  12$:    BIC     #RENA,@VJCSR    ;CLEAR REC. INTERRUPTS BEFORE NEXT UNIT SELECT.
3267  014646  000137  002752                  JMP     MODCA           ;END OF LIST-GO SET UP NEXT 61
3268  014652  100004                  2$:     BPL     3$              ;WAS TEST REPEAT REQUESTED?
3269  014654  012737  001570  002230          MOV     #INTAB,TSTPTR   ;YES-RESET TEST POINTER
3270  014662  000750                          BR      10$             ;AND GET FIRST TEST SELECTED
3271  014664  005301                  3$:     DEC     R1              ;ADJUST OFFSET
3272  014666  006301                          ASL     R1              ;USE TEST # TO FORM ADDRESS OFFSET
3273  014670  016137  023656  014726          MOV     TSTADD(R1),JMPADD+2     ;LOAD NEW ADDRESS
3274  014676  062737  000002  014726          ADD     #2,JMPADD+2     ;BYPASS INITIAL SCOPE LOOP
3275  014704  005237  002230                  INC     TSTPTR          ;INCREMENT TEST POINTER
```

```
 3276  014710  005037  177776            CLR     PSW             ;SET NON-INT. PRIORITY TO ZERO
 3277  014714  005046                    CLR     -(SP)           ;CLEAR THE PSW,LSI11 STYLE.
 3278  014716  012746  014724            MOV     #JMPADD,-(SP)
 3279  014722  000002                    RTI
 3280  014724  000137  014724  JMPADD:   JMP     JMPADD          ;EXIT TO NEXT SELECTED TEST
 3281                                    ;*************************************************************
 3282                                    ;*************************************************************
 3283                                    ;FOLLOWING ROUTINES ARE INTERRUPT HANDLERS FOR THE
 3284                                    ;DJ11 QUICK-TEST.
 3285                                    ;*************************************************************
 3286                                    ;*************************************************************
 3287  014730  117737  165030  002216  RECAD:  MOVB    @VRBUF,CHRD     ;GET THE RECEIVED CHAR.
 3288  014736  042737  000200  002216           BIC     #200,CHRD       ;CLEAR ANY PARITY.
 3289  014744  123737  002246  002216           CMPB    TPREG,CHRD      ;COMPARE RECEIVED TO XMITTED
 3290  014752  001407                           BEQ     UPD4            ;AND UPDATE PATTERN IF OK.
 3291  014754  042777  040004  165000  TOFF:    BIC     #TCOMB,@VJCSR   ;DATA ERROR OCCURED OR WE ARE DONE
 3292  014762  042777  000100  164772           BIC     #RENA,@VJCSR    ;EITHER WAY-EXIT.
 3293  014770  000002                  REEX:    RTI
 3294  014772  052777  040000  164762  UPD4:    BIS     #XENA,@VJCSR    ;ENABLE XMIT INT.
 3295  015000  106337  002246                   ASLB    TPREG           ;UPDATE DATA PATTERN.
 3296  015004  032737  000200  002246           BIT     #BIT07,TPREG    ;ROTATED TO PARITY BIT?
 3297  015012  001766                           BEQ     REEX            ;NO-CONTINUE TESTING
 3298  015014  005037  002246                   CLR     TPREG           ;YES-SET UP COMPLETE FLAG
 3299  015020  000755                           BR      TOFF            ;AND EXIT.
 3300  015022  113777  002246  164740  TSMAD:   MOVB    TPREG,@VXBUF    ;XMIT DATA
 3301  015030  042777  040000  164724           BIC     #XENA,@VJCSR    ;CLEAR XMIT INT. UNTIL LAST BIT REC.
 3302  015036  000002                           RTI
 3303
 3304                                    ;*************************************************************
 3305                                            ;RECEIVE INTERRUPT ROUTINE. ROUTINE WILL TRAP ALL ESC
 3306                                            ;FUNCTIONS AND WILL SET FLAGS IN VSTAT FOR SOM, EOM, XON,
 3307                                            ;XOFF, AND OTHER SPECIAL FUNCTIONS(SEE VSTAT TABLE DEFINITION).
 3308                                            ;ALL INTERFACE STATUS ERRORS WILL BE REPORTED. MAXIMUM EXECUTION
 3309                                            ;TIME FOR THIS ROUTINE IS 200 MICRO SECONDS. AV. = 100.
 3310                                            ;UPON RECEIPT ON XON, XMTKIL BIT IS CHECKED IN VSTAT
 3311                                            ;AND IF SET, WILL BE CLEARED AND XMIT INT. ENABLE SET.
 3312                                            ;LOCATION ESAMB IS USED FOR ESC ASSEMBLY FLAGS. IE. BIT
 3313                                            ;00 SET MEANS A033 WAS RECEIVED, BIT 01 SET MEANS AN ESCP
 3314                                            ;SEQUENCE IS BEING ASSEMBLED. BIT 03
 3315                                            ;SET INDICATES AND ESCAPE O SEQUENCE IS BEING ASSEMBLED.
 3316                                    ;*************************************************************
 3317
 3318  015040                  INTRC:
 3319  015040  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 3320  015042  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 3321  015044  017701  164714                   MOV     @VRBUF,R1       ;USE R1 FOR  STORAGE OF STATUS AND CH.
 3322  015050  010102                           MOV     R1,R2           ;SET UP LINE CHECK LOCATION.
 3323  015052  042702  170377                   BIC     #170377,R2      ;CLEAR ALL BUT LINE BITS.
 3324  015056  023702  001776                   CMP     OCTLNE,R2       ;COMPARE CHAR LINE TO LINE UNDER TEST.
 3325  015062  001404                           BEQ     13$             ;YES-EXAMINE IT.
 3326  015064  052737  000010  002260           BIS     #ILLNE,VSTAT    ;NO-SET ILLEGAL LINE FLAG AND EXIT.
 3327  015072  000567                           BR      ERLNE
 3328  015074  042701  000200          13$:     BIC     #200,R1         ;STRIP PARITY BIT.
 3329  015100  032737  000100  002260           BIT     #TXSUM,VSTAT    ;CHECKSUM CALCULATION REQUESTED?
 3330  015106  001403                           BEQ     11$             ;NO
 3331  015110  010105                           MOV     R1,R5           ;YES-STORE CHAR. AND
```

```
3332  015112  004037  020700              JSR     R0,CALCK        ;CALCULATE THE CHECKSUM.
3333  015116  005237  032323      11$:    INC     ABUFP           ;INCREMINT THE RAW DATA POINTER
3334  015122  023727  032320  032404      CMP     ABUFP,#ABBUF+50.        ;AT THE END OF BUFFER?
3335  015130  001003                      BNE     12$             ;NO
3336  015132  012737  032322  032320      MOV     #ABBUF,ABUFP    ;YES-RESET IT
3337  015140  110177  015154      12$:    MOVB    R1,@ABUFP       ;STORE THE RAW DATA
3338  015144  001505                      BEQ     6$              ;IF CHAR. IS NULL-GO STORE IT
3339  015146  032737  000013  015762      BIT     #BIT00+BIT01+BIT03,ESAMB        ;ESC OR ESC 0?
3340  015154  001152                      BNE     ASESC           ;YES-KEEP ASSEMBLING
3341  015156  120137  002166              CMPB    R1,ESCN         ;BYTE = ESCN?
3342  015162  101076                      BHI     6$              ;NO-PROBABLY A DISPLAY CH.-STORE IT.
3343  015164  001007                      BNE     1$              ;NO-DECODE FOR XON,XOFF,SOM,EOM
3344  015166  012737  000001  015762      MOV     #1,ESAMB        ;YES SET ESC ASSEMBLY FLAG.
3345  015174  052737  000400  002260      BIS     #ESC,VSTAT      ;SET ESC RECEIVED FLAG
3346  015202  000515                      BR      RSTER           ;AND EXIT
3347  015204  120127  000023      1$:     CMPB    R1,#XOFF        ;SEE IF RECEIVED BYTE WAS XOFF
3348  015210  001004                      BNE     2$              ;NO
3349  015212  052737  100000  002260      BIS     #RXOFF,VSTAT    ;YES, SET XOFF IN STATUS REG.
3350  015220  000506                      BR      RSTER           ;EXIT
3351  015222  120127  000021      2$:     CMPB    R1,#XON         ;SEE IF BYTE WAS XON
3352  015226  001016                      BNE     3$              ;NO
3353  015230  042737  100000  002260      BIC     #RXOFF,VSTAT    ;YES, CLEAR XOFF IN VSTAT.
3354  015236  032737  000200  002260      BIT     #XMKIL,VSTAT    ;CHECK XMIT KILL BIT.
3355  015244  001474                      BEQ     RSTER           ;NOT SET, EXIT
3356  015246  052777  040000  164506      BIS     #XENA,@VJCSR    ;SET XMIT INT. ENABLE.
3357  015254  042737  000200  002260      BIC     #XMKIL,VSTAT    ;CLEAR THE XMIT KILLED FLAG
3358  015262  000465                      BR      RSTER           ;EXIT
3359  015264  120127  000002      3$:     CMPB    R1,#SOM         ;SEE IF BYTE WAS SOM
3360  015270  001004                      BNE     4$              ;NO
3361  015272  052737  040000  002260  31$:    BIS     #RSOM,VSTAT     ;YES, SET SOM IN VSTAT.
3362  015300  000456                      BR      RSTER           ;EXIT
3363
3364  015302  120127  000004      4$:     CMPB    R1,#EOM         ;WAS BYTE EOM?
3365  015306  001012                      BNE     5$              ;NO
3366  015310  052737  020000  002260      BIS     #REOM,VSTAT     ;NOW SET EOM IN VSTAT.
3367  015316  013737  015754  015760      MOV     RBBUF,RBUFP     ;RESET THE BUFFER POINTER.
3368  015324  042737  000100  002260      BIC     #TXSUM,VSTAT    ;CLEAR CHECKSUM REQUEST BIT.
3369  015332  000441                      BR      RSTER           ;AND EXIT
3370  015334  123701  002006      5$:     CMPB    CARRT,R1        ;CHAR. =CARRIAGE RETURN?
3371  015340  001403                      BEQ     51$             ;YES-GO SET END OF LINE FLAG
3372  015342  123701  002010              CMPB    LNFED,R1        ;CHAR.= LINEFEED?
3373  015346  001004                      BNE     6$              ;NO- GO STORE IT
3374  015350  052737  001000  002260  51$:    BIS     #EPL,VSTAT      ;SET END OF LINE INDICATOR
3375  015356  000427                      BR      RSTER
3376
3377  015360  023737  015760  015756  6$:     CMP     RBUFP,REBUF     ;IS CIRCULAR BUFFER FILLED?
3378  015366  001003                      BNE     61$             ;NO
3379  015370  013737  015754  015760      MOV     RBBUF,RBUFP     ;YES, RESET POINTER TO BEGINNING
3380  015376  032737  000020  002260  61$:    BIT     #COMGP,VSTAT    ;RECEIVING GRAPHICS CHAR.?
3381  015404  001402                      BEQ     7$              ;NO
3382  015406  162701  000137              SUB     #137,R1         ;YES-SUBTRACT 137 FROM RECEIVED CHAR.
3383
3384  015412  032737  000040  002260  7$:     BIT     #REVID,VSTAT    ;REVERSE VIDEO MODE?
3385  015420  001402                      BEQ     70$             ;NO STORE RECEIVED BYTE.
3386  015422  052701  000200              BIS     #200,R1         ;YES-FORCE BIT7 AS REV. VIDEO IND.
3387  015426  110177  000326      70$:    MOVB    R1,@RBUFP       ;STORE BYTE AND
```

```
3388  015432  005237  015760                    INC     RBUFP           ;INCREMENT POINTER.
3389
3390  015436  032701  070000          RSTER:    BIT     #70000,R1       ;CHECK FOR STATUS ERROR
3391  015442  001414                            BEQ     RECXT           ;NO, EXIT ROUTINE
3392
3393  015444  052737  004000  002260            BIS     #RSTT,VSTAT     ;SET STATUS ERROR FLAG IN VSTAT
3394  015452  027727  000330  177777  ERLNE:    CMP     @STTEP,#-1      ;IS ERROR TABLE FULL?
3395  015460  001405                            BEQ     RECXT           ;YES, EXIT ROUTINE
3396  015462  010177  000320                    MOV     R1,@STTEP       ;NO, STORE STATUS ERR. AND CHECK
3397  015466  062737  000002  016006            ADD     #2,STTEP        ;INCREMENT STATUS ERR. POINTER
3398
3399  015474                          RECXT:
3400  015474  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
3401  015476  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
3402  015500  000002                            RTI                     ;EXIT
3403  015502  032737  000002  015762  ASESC:    BIT     #2,ESAMB        ;ASSEMBLING ESC P?
3404  015510  001063                            BNE     AESCP           ;YES-GO GET LAST CH,
3405  015512  032737  000010  015762            BIT     #BIT03,ESAMB    ;ASSEMBLING ESC 0?
3406  015520  001062                            BNE     AESCO           ;YES
3407  015522  122701  000120                    CMPB    #120,R1         ;CH.= A P?
3408  015526  001004                            BNE     10$             ;NO KEEP CHECKING
3409  015530  052737  000002  015762            BIS     #BIT01,ES  3    ;YES-SET ESCP ASSEMBLY FLAG
3410  015536  000737                            BR      RSTER           ;AND EXIT
3411  015540  122701  000077          10$:      CMPB    #77,R1          ;CHAR.IS AN ESC ? ?
3412  015544  001403                            BEQ     110$            ;YES-FAKE AN ESC O.
3413  015546  122701  000117                    CMPB    #117,R1         ;CHAR = O?
3414  015552  001004                            BNE     11$             ;NO
3415  015554  052737  000010  015762  110$:     BIS     #BIT03,ESAMB    ;YES SET ESC O ASSEMBLY FLAG
3416  015562  000725                            BR      RSTER           ;AND EXIT
3417  015564  123701  002104          11$:      CMPB    RDCUR,R1        ;BYTE= CURSOR POSITION?
3418  015570  001004                            BNE     1$              ;NO-
3419  015572  052737  000004  002260            BIS     #CURPOS,VSTAT   ;YES-SET RECEIVED CURSOR POSITION.
3420  015600  000424                            BR      CESAM
3421  015602  122701  000057          1$:       CMPB    #SLSH,R1        ;BYTE=TERMINAL ID ESC?
3422  015606  001004                            BNE     2$              ;NO-CHECK FOR GRAPHICS SEQUENCE.
3423  015610  052737  000002  002260            BIS     #TRMID,VSTAT    ;YES-SET TERM. IDENT FLAG IN VSTAT
3424  015616  000415                            BR      CESAM
3425  015620  122701  000106          2$:       CMPB    #CKGP,R1        ;RECEIVED GRAPHICS CHAR. SEQUENCE?
3426  015624  001004                            BNE     3$              ;NO
3427  015626  052737  000020  002260            BIS     #COMGP,VSTAT    ;YES-SET GRAPHICS DATA FLAG.
3428  015634  000406                            BR      CESAM
3429  015636  122701  000107          3$:       CMPB    #NCKGP,R1       ;RECEIVED RESET GRAPHICS SEQ.?
3430  015642  001003                            BNE     CESAM           ;NO
3431  015644  042737  000020  002260            BIC     #COMGP,VSTAT    ;YES-SET NORMAL CHAR. RECEIVE.
3432  015652  005037  015762          CESAM:    CLR     ESAMB           ;CLEAR ASSEMBLY FLAG.
3433  015656  000667                            BR      RSTER           ;AND EXIT.
3434
3435  015660  110137  016012          AESCP:    MOVB    R1,STRP         ;STORE ANY UNCHECKED FOR  ESC. P
3436  015664  000772                            BR      CESAM
3437
3438  015666  123701  002052          AESCO:    CMPB    EEMP,R1         ;BYTE=ESC O  -REV. VIDEO- ?
3439  015672  001004                            BNE     1$              ;NO
3440  015674  052737  000040  002260            BIS     #REVID,VSTAT    ;YES-SET REVERSE VIDEO MODE IN VSTAT.
3441  015702  000763                            BR      CESAM
3442
3443  015704  123701  002054          1$:       CMPB    DEMP,R1         ;BYTE=ESC O DISABLE REV. VIDEO MODE?
```

```
3444  015710  001004                        BNE     2$                ;NO
3445  015712  042737  000040  002260        BIC     #REVID,VSTAT      ;YES-CLEAR REVERSE VIDEO MODE IN VSTAT.
3446  015720  000754                        BR      CESAM
3447  015722  122701  000171        2$:     CMPB    #CPABRT,R1        ;COPIER ABORT?
3448  015726  001403                        BEQ     3$                ;YES-SET ABORT FLAG IN VSTAT
3449  015730  122701  000172                CMPB    #PRABRT,R1        ;PRINTER ABORT?
3450  015734  001004                        BNE     4$                ;NO
3451  015736  052737  010000  002260  3$:   BIS     #PABRT,VSTAT      ;YES-SET THE ABORT FLAG.
3452  015744  000742                        BR      CESAM             ;AND EXIT.
3453  015746  1:0137  016010        4$:     MOVB    R1,STRO           ;STORE ESCAPE 0 COMMAND
3454  015752  000737                        BR      CESAM
3455
3456  015754  000000                RBBUF:  .WORD                     ;ADDRESS OF STAT OF BUFFER
3457  015756  000000                REBUF:  .WORD                     ;ADDRESS OF END OF BUFFER.
3458  015760  000000                RBUFP:  .WORD                     ;READ BUFFER POINTER.
3459  015762  000000                ESAMB:  .WORD   0                 ;ESCAPE SEQ.ASSEMBLY AREA
3460
3461  015764                        STTER:
3462  015764  000000                        0
3463  015766  000000                        0
3464  015770  000000                        0
3465  015772  000000                        0
3466  015774  000000                        0
3467  015776  000000                        0
3468  016000  000000                        0
3469  016002  000000                        0
3470  016004  177777                        .WORD   -1                ;STATUS REGISTER DELIMITER.
3471  016006  000000                STTEP:  .WORD                     ;STATUS ERROR POINTER.
3472  016010  000000                STRO:   .WORD   0                 ;ESCAPE 0 STORAGE
3473  016012  000000                STRP:   .WORD                     ;ESCAPE P STORAGE
3474
3475                                ;*****************************************************************
3476                                ;TRANSMIT INTERRUPT ROUTINE. IF XOFF BIT IS SET
3477                                ;IN VSTAT TRANMISSION WILL NOT OCCUR AND THIS ROUTINE
3478                                ;WILL RESET XMIT INT. ENABLE.  IF AFTER TRANSMISSION
3479                                ;OF THE CHARACTER DURING THIS INTERRUPT CYCLE, THE
3480                                ;XMIT COUNT (XMCNT) IS EQUAL TO ZERO,
3481                                ;THE XMIT DONE BIT WILL BE SET IN VSTAT AND XMIT
3482                                ;INT ENABLE BIT WILL CLEARED. TRANSMIT COUNT(XMCNT) MUST BE
3483                                ;SET TO THE NUMBER OF BYTE/CHARACTER TO TRANSMIT.
3484                                ;IF LOCATION BLKM IS SET TO 1001,A SOM WILL PRECEED THE
3485                                ;DATA AND A EOM WILL FOLLOW IT. IF XMZER IS SET TO NON-
3486                                ;ZERO, ALL DATA(INCLUDING ZEROS) WILL BE XMITTED.
3487                                ;*****************************************************************
3488  016014  005737  002260        INTXM:  TST     VSTAT             ;HAS 61 TRANSMITTED XOFF?
3489  016020  100004                        BPL     NOKIL             ;NO XMIT ANOTHER
3490  016022  052737  000200  002260        BIS     #XMKIL,VSTAT      ;SET XMIT KILLED BIT IN VSTAT
3491  016030  000510                        BR      KIENA             ;GO KILL XMIT ENABLE
3492
3493  016032  105737  002263        NOKIL:  TSTB    BLKM+1            ;SOM/EOM TRANSMIT?
3494  016036  001406                        BEQ     NOSOM             ;NO
3495  016040  112777  000002  163722        MOVB    #SOM,@VXBUF       ;YES-ISSUE START OF MESSAGE.
3496  016046  105037  002263                CLRB    BLKM+1            ;AND CLEAR SOM FLAG.
3497  016052  000002                        RTI
3498  016054  005737  016270        NOSOM:  TST     XMCNT             ;XMITTED THE BUFFER?
3499  016060  001006                        BNE     100$              ;NO-XMIT A NORMAL CHAR.
```

```
3500  016062  112777  000004  163700          MOVB    #EOM,@VXBUF     ;YES SEND EOM AND EXIT
3501  016070  105037  002262                  CLRB    BLKM
3502  016074  000452                          BR      2$
3503  016076  105777  000164          100$:   TSTB    @TBUFP          ;CHECK FOR CH.= ZERO. IF SO DO NOT XMIT
3504  016102  001016                          BNE     1$              ;OR COUNT BYTE. OR ARE WE
3505  016104  005737  021652                  TST     XMZER           ;XMITTING ZEROS?
3506  016110  001023                          BNE     22$             ;YES-XMIT NEXT BYTE
3507  016112  023737  016266  016264          CMP     TBUFP,TEBUF     ;AT END OF BUFFER?
3508  016120  001004                          BNE     10$             ;NO
3509  016122  013737  016262  016266          MOV     TBBUF,TBUFP     ;YES-RESET BUFFER POINTER
3510  016130  000740                          BR      NOKIL
3511  016132  005237  016266          10$:    INC     TBUFP
3512  016136  000735                          BR      NOKIL           ;LOOK FOR NON-ZERO BYTE TO TRANSMIT.
3513
3514  016140  032737  002000  002260  1$:     BIT     #CKSUM,VSTAT    ;CHECKSUM REQUESTED?
3515  016146  001404                          BEQ     22$
3516  016150  117705  000112                  MOVB    @TBUFP,R5       ;YES,LOAD THE BYTE
3517  016154  004037  020700                  JSR     R0,CALCK        ;AND CALCULATE THE NEW CHECKSUM.
3518  016160  117777  000102  163602  22$:    MOVB    @TBUFP,@VXBUF   ;TRAMSMIT A CHARACTER
3519  016166  023737  016266  016264          CMP     TBUFP,TEBUF     ;AT END OF CIRCULAR BUFFER?
3520  016174  001004                          BNE     11$             ;NO
3521  016176  013737  016262  016266          MOV     TBBUF,TBUFP     ;YES, RESET IT TO START.
3522  016204  000402                          BR      12$             ;BY-PASS INCREMENT BUFF. POINTER
3523  016206  005237  016266          11$:    INC     TBUFP           ;INCREMENT BUFFER POINTER.
3524
3525  016212  005337  016270          12$:    DEC     XMCNT           ;DECREMENT THE TRANSMIT COUNT
3526  016216  001401                          BEQ     2$              ;YES,CLEANUP,REQUEST ERRORS AND EXIT.
3527  016220  000002                          RTI                     ;NO, CONTINUE
3528  016222  105737  002262          2$:     TSTB    BLKM            ;SOM/EOM XMIT?
3529  016226  001014                          BNE     TXEX            ;YES-DO NOT SET XMDNE UNTIL EOM SENT.
3530  016230  052737  000001  002260          BIS     #XMDNE,VSTAT    ;SET THE DONE BIT IN VSTAT.
3531  016236  042737  002000  002260          BIC     #CKSUM,VSTAT    ;CLEAR THE CHECKSUM FLAG WHEN DONE.
3532  016244  013737  016262  016266          MOV     TBBUF,TBUFP     ;RESET BUFFER POINTER.
3533  016252  042777  040000  163502  KIENA:  BIC     #XENA,@VJCSR    ;CLEAR XMIT. INT. ENABLE
3534  016260  000002                  TXEX:   RTI
3535
3536
3537  016262  000000                  TBBUF:  .WORD                   ;CONTAINS INITIAL ADDRESS
3538  016264  000000                  TEBUF:  .WORD                   ;CONTAIN LAST ADDRESS
3539  016266  000000                  TBUFP:  .WORD                   ;CONTAINS CURRENT LOCATION
3540
3541  016270  000000                  XMCNT:  .WORD   0               ;LOADED WITH NUMBER OF XMITS.
3542                                  ;*****************************************************************
3543
3544
3545                                          ;SUBROUTINE TO ISSUE RESET TO THE VT61, ENTERS MAINTENANCE MODE
3546                                          ;AND FORCES LINEAR ADDRESSING.
3547                                  ;*****************************************************************
3548
3549  016272  113737  001102  002264  RESETV: MOVB    $TSTNM,TSTNM    ;LOAD THE TEST NUMBER IN ERROR PRINT AREA.
3550  016300  013746  002224                  MOV     ZERO,-(SP)      ;;PUSH ZERO ON STACK
3551  016304  013746  002164                  MOV     RESET,-(SP)     ;;PUSH RESET ON STACK
3552  016310  013746  002110                  MOV     ESCO,-(SP)      ;;PUSH ESCO ON STACK
3553  016314  004037  014370                  JSR     R0,TESC         ;GO XIMT IT
3554  016320  004037  016402                  JSR     R0,GETON        ;GO LOOK FOR XON.
3555  016324  000405                          BR      1$              ;FOUND IT.
```

```
3556  016326  005237  002234            INC     FTLCNT          ;ADD 1 TO FATAL XMIT COUNT.
3557  016332  010037  001120            MOV     R0,$GDADR       ;NO XON ISSUE XON ERROR
3558  016336  104017                    ERROR   17
3559  016340                    1$:
3560  016340  013746  002224            MOV     ZERO,-(SP)      ;;PUSH ZERO ON STACK
3561  016344  013746  002034            MOV     EMAIN,-(SP)     ;;PUSH EMAIN ON STACK
3562  016350  013746  002110            MOV     ESCO,-(SP)      ;;PUSH ESCO ON STACK
3563  016354  013746  002044            MOV     DRECT,-(SP)     ;;PUSH DRECT ON STACK
3564  016360  013746  002110            MOV     ESCO,-(SP)      ;;PUSH ESCO ON STACK
3565  016364  004037  014370    2$:     JSR     R0,TESC
3566  016370  005037  002260            CLR     VSTAT           ;CLEAR INT. FLAGS AFTER TERMINAL RESET
3567  016374  005037  020072            CLR     HDFLG           ;CLEAR PRINT HEADER FLAG.
3568  016400  000200                    RTS     R0
3569
3570                    ;*****************************************************************
3571                            ;SUBROUTINE TO WAIT FOR AN XON. NO XON EXIT IS PC +2.
3572                    ;*****************************************************************
3573
3574  016402  012737  000454  002244    GETON:  MOV     #300.,BUBCT     ;SET UP TO LOOK FOR 3 SEC.
3575  016410  105077  013704            CLRB    @ABUFP
3576  016414  127727  013700  000021    1$:     CMPB    @ABUFP,#XON     ;RECEIVED A XON?
3577  016422  001412                    BEQ     GOTON           ;YES-EXIT.
3578  016424  012737  000001  020220            MOV     #1,DCOUNT       ;NO-DELAY 10 M.S.
3579  016432  004037  020156            JSR     R0,DELAY
3580  016436  005337  002244            DEC     BUBCT           ;AT END OF DELAY?
3581  016442  001364                    BNE     1$              ;NO
3582  016444  062700  000002            ADD     #2,R0           ;YES-SET UP ERROR EXIT.
3583  016450  000200            GOTON:  RTS     R0
3584
3585                    ;*****************************************************************
3586
3587                            ;SUBROUTINE TO ISSUE ESCZ AND LOOK FOR A RESPONSE-EITHER
3588                            ;A -1 OR THE RETURNED IDENT. THE -1 INDICATES NO
3589                            ;RESPONSE FROM THE UNIT UNDER TEST.
3590                    ;*****************************************************************
3591
3592  016452            ZFLAG:
3593  016452  012637  016510            MOV     (SP)+,ROSV1     ;;POP STACK INTO ROSV1
3594  016456  013746  002224            MOV     @#ZERO,-(SP)    ;;PUSH @#ZERO ON STACK
3595  016462  013746  002162            MOV     @#ESCZ,-(SP)    ;;PUSH @#ESCZ ON STACK
3596  016466  004037  014370            JSR     R0,TESC         ;GO ISSUE ESZ SEQUENCE
3597  016472  012746  106003            MOV     #106003,-(SP)   ;;PUSH #106003 ON STACK
3598  016476  004037  014116            JSR     R0,RECTM        ;GO READ THE CHARACTER
3599  016502  013746  016510            MOV     ROSV1,-(SP)     ;;PUSH ROSV1 ON STACK
3600  016506  000200                    RTS     R0
3601  016510  000000            ROSV1:  .WORD   0
3602
3603                    ;*****************************************************************
3604                            ;ROUTINE TO CHECK SOFTWARE STATUS REGISTER (VSTAT)
3605                            ;RECEIVE FLAGS ONLY.  ENTERED WITH ANTICIPATED
3606                            ;STATUS WORD ON THE STACK.
3607                    ;*****************************************************************
3608
3609  016512            CKSFT:
3610  016512  012637  002256            MOV     (SP)+,ROSVE     ;;POP STACK INTO ROSVE
3611  016516  010137  002220            MOV     R1,SVER1        ;SAVE R1
```

```
 3612  016522  010237  002222                MOV     R2,SVER2        ;SAVE R2
 3613
 3614  016526  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 3615  016530  013702  002260                MOV     VSTAT,R2        ;SET R2 EQUAL TO VSTAT
 3616
 3617  016534  042702  003566                BIC     #003566,R2      ;CLEAR NON-ERROR BITS
 3618  016540  020102                        CMP     R1,R2           ;COMPARE ANTICIPATED TO ACTUAL.
 3619  016542  001432                        BEQ     NOER            ;NO UNUSAL BITS EXIT
 3620
 3621  016544  010137  001124                MOV     R1,$GDDAT       ;MOVE GOOD STATUS TO MESSAGE
 3622  016550  013737  002260  001126        MOV     VSTAT,$BDDAT    ;MOVE BAD STATUS TO MESSAGE
 3623  016556  104003                        ERROR   3               ;ISSUE ERROR MESSAGE.
 3624
 3625
 3626                          ;*****************************************************************
 3627                                  ;ROUTINE TO PRINT THE STATUS REGISTER IN THE FOLLOWING
 3628                                  ;FORMAT: STATUS BITS (XXX 000), CHARACTER TRANSFERRED (000  X X)
 3629                          ;*****************************************************************
 3630
 3631  016560  012701  015764                MOV     #STTER,R1       ;SET R1 EQUAL TO FIRST ENTRY
 3632  016564  013702  016006                MOV     STTEP,R2        ;SET R2 EQUAL LAST ENTRY
 3633  016570  020102                1$:     CMP     R1,R2           ;ARE THEY EQUAL
 3634  016572  001416                        BEQ     NOER            ;YES-RESET POINTERS AND EXIT.
 3635  016574  004037  016654                JSR     R0,CLREG        ;CLEAR ERROR PRINT LOC.
 3636  016600  013737  001762  001120        MOV     VJCSR,$GDADR    ;LOAD ADDRESS
 3637  016606  017737  163150  001124        MOV     @VJCSR,$GDDAT   ;LOAD CSR
 3638  016614  112137  001126          2$:   MOVB    (R1)+,$BDDAT    ;MOVE CHARACTER AND
 3639  016620  112137  001123                MOVB    (R1)+,$BDADR+1  ;STATUS BITS TO ERROR REGISTERS.
 3640  016624  104002                        ERROR   2               ;ISSUE ERROR MESSAGE
 3641  016626  000760                        BR      1$              ;DO AGAIN
 3642  016630  013701  002220          NOER: MOV     SVER1,R1        ;RESTORE R1 AND
 3643  016634  013702  002222                MOV     SVER2,R2        ;R2.
 3644  016640  012737  015764  016006        MOV     #STTER,STTEP    ;RESET STATUS ERROR POINTER.
 3645  016646  013746  002256                MOV     ROSVE,-(SP)     ;;PUSH ROSVE ON STACK
 3646  016652  000200                        RTS     R0              ;EXIT
 3647
 3648                          ;*****************************************************************
 3649                                  ;SUBROUTINE TO CLEAR ERROR/DATA OUTPUT LOCATIONS. NEEDED
 3650                                  ;ONLY WHEN DISPLAYING BYTES IN WORD LOCATIONS.
 3651                          ;*****************************************************************
 3652
 3653  016654  005037  001120          CLREG:  CLR     $GDADR
 3654  016660  005037  001122                  CLR     $BDADR
 3655  016664  005037  001124                  CLR     $GDDAT
 3656  016670  005037  001126                  CLR     $BDDAT
 3657  016674  000200                          RTS     R0
 3658                          ;*****************************************************************
 3659                                  ;SUBROUTINE TO TRANSMIT THE BUFFER AND WAIT FOR XMIT DONE
 3660                                  ;AND END OF RECEIVE MESSAGE. SUBROUTINE WILL LOOP IF LOCATION
 3661                                  ;RECITT IS PRE-LOADED WITH A NUMBER HIGHER THAN1(IE. MULTIPLE
 3662                                  ;RECEIVES CAN BE ACCOMPLISHED WITH ONLY ONE ENTRY TO SUB-
 3663                                  ;ROUTINE).WDSTOR AND BYSTOR ARE THE WORD(CURSOR POS.) AND BYTE
 3664                                  ;STORAGE LOCATIONS,RESPECTIVELY.DEFAULT STORAGE IS THE REC. BUFFER.
 3665
 3666                          ;*****************************************************************
 3667
```

```
3668  016676                                XMREC:
3669  016676  010546                              MOV     R5,-(SP)        ;PUSH R5 ON STACK
3670  016700  012737  001001  002262              MOV     #1001,BLKM      ;SET UP FOR A SOM/EOM TRANSMIT.
3671  016706  042737  077577  002260              BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL.
3672  016714  013701  017150                      MOV     BYSTOR,R1       ;LOAD THE STORAGE POINTERS
3673  016720  013702  017146                      MOV     WDSTOR,R2
3674  016724  052777  040000  163030              BIS     #XENA,@VJCSR    ;SET INTERRUPT ENABLES
3675  016732  042737  061466  002260  XMITT:      BIC     #61466,VSTAT    ;CLEAR SOM,EOM,EPL,ESC,REV.VID., PARA. DELIM.,IDENT,CUR.
3676  016740  005037  002254          1$:         CLR     DLAY            ;SET UP TIME OUT DELAY.
3677  016744  032737  000001  002260              BIT     #XMDNE,VSTAT    ;IS XMIT DONE?
3678  016752  001015                              BNE     3$              ;YES-LOOK FOR RECEIVE DONE.
3679  016754  032737  020000  002260  2$:         BIT     #REOM,VSTAT     ;RECEIVED AN EOM?
3680  016762  001401                              BEQ     20$             ;NO
3681  016764  000435                              BR      CKSTR           ;YES-GO HANDLE DATA
3682  016766  032737  100000  002260  20$:        BIT     #RXOFF,VSTAT    ;NO- IS XOFF SET?
3683  016774  001761                              BEQ     1$              ;NO-STILL TRANSMITTONG.
3684  016776  005337  002254                      DEC     DLAY            ;YES- RUN DELAY
3685  017002  001364                              BNE     2$              ;WAITING FOR XON
3686  017004  000416                              BR      XMAD2           ;NO XON-REPORT VT61 FAILURE.
3687
3688  017006  013705  032320          3$:         MOV     ABUFP,R5        ;LOAD CH. RECEIVED FLAG.
3689  017012  005037  002254                      CLR     DLAY            ;SET UP RECEIVE DELAY.
3690  017016  032737  020000  002260  4$:         BIT     #REOM,VSTAT     ;RECEIVE END OF MESSAGE?
3691  017024  001015                              BNE     CKSTR           ;YES-CHECK DATA STORAGE POINTERS
3692  017026  020537  032320                      CMP     R5,ABUFP        ;RECEIVED ANOTHER CHARACTER?
3693  017032  001365                              BNE     3$              ;YES-RESET CH. FLAG AND DELAY
3694  017034  005337  002254          5$:         DEC     DLAY            ;RUN DELAY
3695  017040  001366                              BNE     4$              ;AND KEEP LOOKING FOR EOM.
3696  017042  062700  000002          XMAD2:      ADD     #2,R0           ;TIME OUT OCCURRED-SET UP ERROR EXIT.
3697  017046  005237  002234                      INC     FTLCNT          ;INCREMENT FATAL XMIT COUNT.
3698  017052  004037  017262                      JSR     R0,RESPTR       ;AND REST ALL INTERRUPT POINTERS.
3699  017056  000422                              BR      CKVST
3700  017060  020102                  CKSTR:      CMP     R1,R2           ;STORAGE POINTERS CLEARED?
3701  017062  001413                              BEQ     CHKITT          ;YES--LEAVE DATA IN REC. BUFFER.
3702  017064  032737  000004  002260              BIT     #CURPOS,VSTAT   ;RECEIVED A CURSOR POSITION?
3703  017072  001403                              BEQ     STRBYT          ;NO-GO STORE A BYTE.
3704  017074  017722  176654                      MOV     @RBBUF,(R2)+    ;YES,STORE IT.
3705  017100  000404                              BR      CHKITT          ;AND CHECK ITERATION COUNT.
3706  017102  005701                  STRBYT:     TST     R1              ;STORING A CHAR?
3707  017104  001402                              BEQ     CHKITT          ;NO
3708  017106  117721  176642                      MOVB    @RBBUF,(R1)+    ;STORE A RECEIVED BYTE
3709  017112  005337  017144          CHKITT:     DEC     RECITT          ;DONE RECEIVING?
3710  017116  001305                              BNE     XMITT           ;NO-LOOP SUBROUTINE
3711  017120  004037  021744                      JSR     R0,CKOFF        ;SEE IS XOFF IS UP.
3712  017124                          CKVST:
3713  017124  012746  060001                      MOV     #60001,-(SP)    ;;PUSH #60001 ON STACK
3714  017130  004037  016512                      JSR     R0,CKSFT
3715  017134  004037  017262                      JSR     R0,RESPTR       ;RESET INTERRUPT POINTERS.
3716  017140  012605                              MOV     (SP)+,R5        ;;POP STACK INTO R5
3717  017142  000200          XMXT:       RTS     R0              ;EXIT SUBROUTINE.
3718  017144  000000          RECITT:     .WORD   0               ;RECEIVE ITERATION COUNT.
3719  017146  000000          WDSTOR:     .WORD   0               ;WORD STORAGE POINTER
3720  017150  000000          BYSTOR:     .WORD   0               ;BYTE STORAGE POINTER
3721
3722                          ;*****************************************************************
3723                                  ;SUBROUTINE TO  XMIT THE BYTE AT TBUFP.
```

```
                                        ;*****************************************************************
                                        ;
017152  042737  000001  002260  XMIT1:  BIC     #1,VSTAT        ;CLEAR XMIT DONE FLAG
017160  012737  000001  016270          MOV     #1,XMCNT        ;SET UP TO XMIT 1 BYTE
017166  052777  040000  162566          BIS     #XENA,@VJCSR
017174                          1$:
017174  012746  000001                  MOV     #XMDNE,-(SP)    ;;PUSH #XMDNE ON STACK
017200  012746  000001                  MOV     #1,-(SP)        ;;PUSH #1 ON STACK
017204  004037  021654                  JSR     R0,WTBGND       ;LOOK FOR XMIT DONE
017210  000401                          BR      FTLEXT          ;HUNG TRANSMIT-CLEAR FLAGS AND EXIT
017212  000402                          BR      NORXT           ;NORMAL EXIT.
017214  005037  002260          FTLEXT: CLR     VSTAT           ;CLEAR ANY FLAGS
017220  000200                  NORXT:  RTS     R0              ;AND EXIT
                                        ;
                                        ;*****************************************************************
                                                ;SUBROUTINE TO ISSUE A BYTE AT A TIME UNTIL A ZERO
                                                ;BYTE IS ENCOUNTERED.
                                        ;*****************************************************************
                                        ;
017222  112777  000002  177036  LDXMIT: MOVB    #SOM,@TBUFP     ;SEND THE START OF MESSAGE.
017230  000403                          BR      2$
017232  112377  177030          1$:     MOVB    (R3)+,@TBUFP    ;MOVE A BYTE TO XMIT BUFFER
017236  001403                          BEQ     LDOUT           ;IF A ZERO BYTE-EXIT
017240  004037  017152          2$:     JSR     R0,XMIT1        ;GO XMIT A BYTE
017244  000772                          BR      1$              ;XMIT AGAIN.
017246  112777  000004  177012  LDOUT:  MOVB    #EOM,@TBUFP     ;SEND THE END OF MESSAGE.
017254  004037  017152                  JSR     R0,XMIT1
017260  000200                          RTS     R0
                                        ;
                                        ;*****************************************************************
                                                ;ROUTINE TO RESET ALL INTERRUPT POINTERS.
                                        ;*****************************************************************
                                        ;
017262  042777  040000  162472  RESPTR: BIC     #XENA,@VJCSR    ;CLEAR INTERRUPT ENABLES
017270  013737  015754  015760          MOV     RBBUF,RBUFP     ;RESET RECEIVE BUF POINTER
017276  013737  016262  016266          MOV     TBBUF,TBUFP     ;RESET XMMT BUF POINTER
017304  012737  015764  016006          MOV     #STTER,STTEP    ;RESET RECEIVE STATUS ERR POINTER
017312  005037  016270                  CLR     XMCNT           ;CLEAR TRANSMIT COUNT
017316  005037  015762                  CLR     ESAMB           ;CLEAR ESC ASSEMBLY FLAGS
017322  012737  000001  017144          MOV     #1,RECITT       ;RESET REC. ITERATION COUNT
017330  005037  017146                  CLR     WDSTOR          ;CLEAR STORAGE POINTERS
017334  005037  017150                  CLR     BYSTOR
017340  000200                          RTS     R0
                                        ;
                                        ;*****************************************************************
                                                ;SUBROUTINE TO ISSUE CURSOR POSITION ERROR. GOOD
                                                ;LINE/COLUMN MUST BE A WORD ON STACK. ERROR
                                                ;POSITION IS EXPECTED TO BE @ RBBUF.
                                        ;*****************************************************************
                                        ;
017342                          CURER:
017342  012637  002256                  MOV     (SP)+,ROSVE     ;;POP STACK INTO ROSVE
017346  012637  002216                  MOV     (SP)+,CHRD      ;;POP STACK INTO CHRD
017352  162737  020040  002216          SUB     #20040,CHRD     ;EXTRACT MOD 40 FROM GOOD POSITION
017360  004037  016654                  JSR     R0,CLREG
```

```
3780  017364  113737  002217  001124          MOVB    CHRD+1,$GDDAT   ;LOAD MESSAGE WITH GOOD
3781  017372  113737  002216  001120          MOVB    CHRD,$GDADR     ;LINE AND COLUMN
3782  017400  017737  176350  002216          MOV     @RBBUF,CHRD     ;LINE AND COLUMN.
3783  017406  162737  020040  002216          SUB     #20040,CHRD     ;EXTRACT MOD 40 FROM BAD POSITION.
3784  017414  113737  002217  001126          MOVB    CHRD+1,$BDDAT   ;LOAD MESSAGE WITH BAD
3785  017422  113737  002216  001122          MOVB    CHRD,$BDADR     ;LINE AND COLUMN.
3786  017430  104006                          ERROR   6               ;ISSUE ERROR
3787  017432  013746  002256                  MOV     ROSVE,-(SP)     ;;PUSH ROSVE ON STACK
3788  017436  000200                          RTS     R0
3789
3790                                  ;*****************************************************************
3791
3792                                  ;*****************************************************************
3793                                          ;SUBROUTINE TO DECREMENT CUSROR POSITION IN A
3794                                          ;LINEAR SEQUENCE. (IE. ROW 20, COL 1 ;ROW 20 COL0 ;ROW 17, COL 157).
3795                                  ;*****************************************************************
3796
3797  017440  123727  017545  000040  CMPOS:  CMPB    LNRW+1,#40      ;AT LEFT EDGE OF ROW?
3798  017446  001403                          BEQ     1$              ;YES, GO ADJUST COL., ROW.
3799  017450  105337  017545                  DECB    LNRW+1          ;NO, DECREMENT COL. AND EXIT
3800  017454  000200                          RTS     R0
3801  017456  123727  017544  000040  1$:     CMPB    LNRW,#40        ;AT ROW 0?
3802  017464  001405                          BEQ     2$              ;YES, NO DECREMENT POSSIBLE-EXIT.
3803  017466  105337  017544                  DECB    LNRW            ;NO, DECREMENT ROW AND
3804  017472  112737  000157  017545          MOVB    #157,LNRW+1     ;SET COL. TO RIGHT EDGE.
3805  017500  000200                  2$:     RTS     R0
3806
3807                                  ;*****************************************************************
3808                                          ;SUBROUTINE TO INCREMENT CURSOR POSITION IN A LINEAR
3809                                          ;SEQUENCE (IE. ROW 10, COL 78, ROW 10. COL 79, ROW 11 , COL 0).
3810                                  ;*****************************************************************
3811
3812  017502  123727  017545  000157  CPPOS:  CMPB    LNRW+1,#157     ;AT RIGHT EDGE OF ROW
3813  017510  001403                          BEQ     1$              ;YES, ADJUST ROW AND COLUMN.
3814  017512  105237  017545                  INCB    LNRW+1          ;NO, INCREMENT COL. COUNT
3815  017516  000200                          RTS     R0              ;AND EXIT
3816  017520  123727  017544  000067  1$:     CMPB    LNRW,#67        ;AT BOTTOM ROW?
3817  017526  001405                          BEQ     2$              ;YES, NO INCREMENT POSSIBLE-EXIT.
3818  017530  105237  017544                  INCB    LNRW            ;NO, INCREMENT ROW COUNT AND
3819  017534  112737  000040  017545          MOVB    #40,LNRW+1      ;SET COL. TO LEFT EDGE.
3820  017542  000200                  2$:     RTS     R0
3821
3822  017544  000000                  LNRW:   .WORD   0               ;CONTAINS UPDATED CURSOR POSITION.
3823                                  ;*****************************************************************
3824
3825                                          ;SUBROUTINE TO XMIT, RECEIVE AND COMPARE. DATA ERRORS
3826                                          ;ARE REPORTED FROM SUBROUTINE. IF THE TRANSMIT OR
3827                                          ;RECEIVE LOOPS 'TIME OUT', EXIT FROM SUBROUTINE WILL
3828                                          ;BE NORMAL EXIT +2. SUBROUTINE ENTERED WITH (R1)=
3829                                          ;GOOD DATA BUFFER, (R2)=RECEIVE DATA BUFFER AND
3830                                          ;R3=COMPARE COUNT. IF THE VT61 DOES NOT HANG,THE ROUTINE
3831                                          ;WILL WAIT FOR END OF REC. MESSAGE(EOM).
3832
3833
3834                                  ;*****************************************************************
3835
```

```
3836  017546                          XRCMP:
3837  017546  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
3838  017550  005004                          CLR     R4              ;USE R4 A RECEIVE COUNTER.
3839  017552  012737  001001  002262          MOV     #1001,BLKM      ;SET UP FOR A SOM/EOM TRANSMIT.
3840  017560  042737  077577  002260          BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL.
3841  017566  052777  040000  162166          BIS     #XENA,@VJCSR    ;SET INTERRUPT ENABLES.
3842  017574  005037  020072                  CLR     HDFLG           ;CLEAR ERROR 13 PRINT FLAG
3843  017600  012705  032270                  MOV     #TCRLB+450,R5   ;R5 IS ERROR STORAGE POINTER
3844  017604  005037  002254          1$:     CLR     DLAY            ;SET UP TIME OUT DELAY
3845  017610  032737  000001  002260          BIT     #XMDNE,VSTAT    ;XMIT DONE?
3846  017616  001014                          BNE     XREC            ;YES-GO RECEIVE
3847  017620  023737  015754  015760  2$:     CMP     RBBUF,RBUFP     ;HAS RECEIVE OPERATION BEGUN?
3848  017626  103410                          BLO     XREC            ;YES-GO RECEIVE
3849  017630  032737  100000  002260          BIT     #RXOFF,VSTAT    ;XMIT XOFF SET?
3850  017636  001762                          BEQ     1$              ;NO-KEEP LOOKING FOR XMIT DONE?
3851  017640  005337  002254                  DEC     DLAY            ;YES RUN DELAY AND LOOK
3852  017644  001365                          BNE     2$              ;FOR XON OR RECEIVED CH.
3853  017646  000432                          BR      XRERR           ;TRANSMIT TIMEOUT-SET UP ERROR EXIT
3854
3855  017650  005037  002254          XREC:   CLR     DLAY            ;SET UP TIME OUT DELAY
3856  017654  020237  015760          1$:     CMP     R2,RBUFP        ;INSURE COMPARE POINTER
3857  017660  103410                          BLO     2$              ;LESS THAN RECEIVE POINTER
3858  017662  032737  020000  002260          BIT     #REOM,VSTAT     ;RECEIVE EOM?
3859  017670  001030                          BNE     XREXT           ;YES-SET UP TO EXIT
3860  017672  005337  002254                  DEC     DLAY            ;RUN TIMEOUT DELAY
3861  017676  001416                          BEQ     XRERR           ;TIME OUT OCCURRED-ERROR EXIT
3862  017700  000765                          BR      1$              ;RETURN TO CHECK RECEIVE COUNT
3863  017702  005204                  2$:     INC     R4              ;ADD 1 TO RECEIVE COUNTER.
3864  017704  122122                          CMPB    (R1)+,(R2)+     ;COMPARE CHARACTERS
3865  017706  001407                          BEQ     4$              ;EQUAL-COMPARE AGAIN
3866  017710  020527  032320                  CMP     R5,#TCRLB+500   ;ALLREADY STORED 50 ERRORS?
3867  017714  103004                          BHIS    4$              ;YES-BYPASS STORAGE
3868  017716  114125                          MOVB    -(R1),(R5)+     ;STORE GOOD DATA
3869  017720  114225                          MOVB    -(R2),(R5)+     ;STORE BAD DATA
3870  017722  010425                          MOV     R4,(R5)+        ;LOAD RECEIVE COUNT
3871  017724  132122                          BITB    (R1)+,(R2)+     ;RESET POINTERS AND
3872  017726  005303                  4$:     DEC     R3              ;CHECK COMPARE COUNT
3873  017730  001410                          BEQ     XREXT           ;ALL DONE-EXIT
3874  017732  000746                          BR      XREC            ;COMPARE ANOTHER
3875  017734  062700  000002          XRERR:  ADD     #2,R0           ;SET UP ERROR EXIT
3876  017740  005237  002234                  INC     FTLCNT          ;INCREMENT FATAL XMIT COUNT.
3877  017744  004037  017262                  JSR     R0,RESPTR       ;RESET INTERRUPT POINTERS.
3878  017750  000440                          BR      XROUT
3879  017752                          XREXT:
3880  017752  012746  020000                  MOV     #REOM,-(SP)     ;;PUSH #REOM ON STACK
3881  017756  012746  000004                  MOV     #4,-(SP)        ;;PUSH #4 ON STACK
3882  017762  004037  021654                  JSR     R0,WTBGND
3883  017766  000431                          BR      XROUT           ;NO EOM-ISSUE ERROR AND EXIT.
3884  017770  162705  032270                  SUB     #TCRLB+450,R5   ;NOW EXTRACT ERROR COUNT-IF ANY.
3885  017774  010501                          MOV     R5,R1           ;AND STORE IT IN R1
3886  017776  012705  032270                  MOV     #TCRLB+450,R5   ;RELOAD ERROR POINTER
3887  020002  005701                          TST     R1              ;TEST FOR ERRORS
3888  020004  001422                          BEQ     XROUT           ;NO-CHECK STATUS AND EXIT
3889  020006  005737  020072                  TST     HDFLG           ;;DATA ERROR HEADER PRINTED?
3890  020012  001003                          BNE     1$              ;YES-BYPASS HEADER PRINT
3891  020014  104012                          ERROR   12              ;PRINT DATA ERROR HEADER
```

```
 3892  020016  005237  020072          INC     HDFLG           ;SET HEADER PRINT FLAG
 3893  020022  004037  016654  1$:     JSR     R0,CLREG        ;ERROR WAS LEGTIMATE. LOAD
 3894  020026  112537  001124          MOVB    (R5)+,$GDDAT    ;ERROR MESSAGE AND ISSUE
 3895  020032  112537  001126          MOVB    (R5)+,$BDDAT    ;IT.
 3896  020036  012537  001120          MOV     (R5)+,$GDADR    ;LOAD RECEIVE COUNT
 3897  020042  104004                  ERROR   4               ;ISSUE DATA COMPARE ERROR
 3898  020044  162701  000004          SUB     #4,R1           ;DECREMENT ERROR COUNT
 3899  020050  001364                  BNE     1$              ;PRINT ANOTHER IF NOT AT ZERO
 3900  020052  004037  021744  XROUT:  JSR     R0,CKOFF        ;SEE IS XOFF IS UP.
 3901  020056  012746  060001          MOV     #60001,-(SP)    ;;PUSH #60001 ON STACK
 3902  020062  004037  016512          JSR     R0,CKSFT        ;CHECK FOR VSTAT /STATUS ERR.
 3903  020066  012604                  MOV     (SP)+,R4        ;;POP STACK INTO R4
 3904  020070  000200                  RTS     R0              ;EXIT SUBROUTINE
 3905
 3906  020072  000000  HDFLG:  0                               ;INHIBIT PRINT FLAG.
 3907
 3908                  ;*****************************************************************
 3909
 3910                          ;SUBROUTINE TO CREATE A 'RULER' IN LOCATIONS 200
 3911                          ;TO 317.
 3912
 3913                  ;*****************************************************************
 3914
 3915  020074  012701  032020  CRRUL:  MOV     #TCRLB+200,R1   ;LOAD STARTING ADDRESS
 3916  020100  012702  130461          MOV     #130461,R2      ;LOAD INITIAL RULER ASCII CODES.
 3917  020104  110221  1$:     MOVB    R2,(R1)+        ;STORE A RULER BYTE IN XMIT BUF.
 3918  020106  022701  032140          CMP     #TCRLB+320,R1   ;RULER COMPLETE?
 3919  020112  103001                  BHIS    2$              ;NO
 3920  020114  000200                  RTS     R0              ;AND EXIT.
 3921  020116  105202  2$:     INCB    R2              ;INCREMENT ASCII BYTE
 3922  020120  122702  000272          CMPB    #272,R2         ;END OF REVERSE VIDEO?
 3923  020124  001003                  BNE     3$              ;NO-SEE IF END OF NORMAL.
 3924  020126  012702  030660          MOV     #030660,R2      ;SET UP TO ISSUE REVERSE 0.
 3925  020132  000405                  BR      5$
 3926  020134  122702  000072  3$:     CMPB    #72,R2          ;END OF NORMAL VIDEO?
 3927  020140  001361                  BNE     1$              ;NOT AT END OF A VIDEO STRING.
 3928  020142  012702  130460          MOV     #130460,R2              ;YES-SET UP TO ISSUE NORMAL 0.
 3929  020146  110221  5$:     MOVB    R2,(R1)+        ;DO IT
 3930  020150  105202                  INCB    R2              ;SET BYTE TO NEXT ASCII CODE
 3931  020152  000302                  SWAB    R2              ;REVERSE VIDEO MODE.
 3932  020154  000753                  BR      1$              ;BEGIN NEXT STRING
 3933
 3934                  ;*****************************************************************
 3935                          ;SUBROUTINE TO DELAY 10 M.S. TIME THE NUMBER INLOCATION
 3936                          ;DCOUNT. THE PROCESSOR TYPE PRE-DETERMINES THE # OF LOOPS
 3937                          ;REQUIRED TO DELAY 10 M.S. FOR ONE ITERATION. LOCATION
 3938                          ;PMULT IS PRE-LOADED WITH : 11/45 = 4, 11/40 = 2
 3939                          ;AND 11/10 =1.
 3940                  ;*****************************************************************
 3941
 3942  020156          DELAY:
 3943  020156  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 3944  020160  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 3945  020162  013702  020216  1$:     MOV     PMULT,R2        ;LOAD PROCESSOR MULTIPLIER
 3946  020166  012701  002570  2$:     MOV     #1400.,R1       ;LOAD 10 M.S. DELAY
 3947  020172  005301                  DEC     R1              ;RUN BASIC DELAY
```

```
3948  020174  001376                        BNE     .-2
3949  020176  005302                        DEC     R2              ;RUN MULTIPLIER DELAY
3950  020200  001372                        BNE     2$
3951  020202  005337  020220                DEC     DCOUNT          ;RUN ITERATION COUNT
3952  020206  001365                        BNE     1$
3953  020210  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
3954  020212  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
3955  020214  000200                        RTS     R0
3956
3957  020216  000000              PMULT:    0                       ;PROCESSOR MULTIPLIER
3958  020220  000000              DCOUNT:   0                       ;ITERATION COUNT
3959
3960                              ;*****************************************************************
3961
3962                                      ;SUBROUTINE TO GENERATE A INCREMENTING PATTERN AT
3963                                      ;(R1)+.  ENTER WITH R3 EQUAL TO # OF CH. TO CREATE.
3964                                      ;R5 IS UTILIZED AS A WORK REGISTER.
3965
3966                              ;*****************************************************************
3967
3968  020222  012705  000041      BLDINC:   MOV     #41,R5          ;LOAD R5 WITH INITIAL CH.
3969  020226  110521              BLDINA:   MOVB    R5,(R1)+        ;MOVE A CH. TO BUFFER
3970  020230  005303                        DEC     R3              ;DECREMENT BYTE COUNT
3971  020232  001001                        BNE     2$              ;NOT DONE-UPDATE PATTERN
3972  020234  000200                        RTS     R0              ;EXIT-DONE.
3973  020236  105205              2$:       INCB    R5              ;UPDATE CH. PATTERN
3974  020240  122705  000177              CMPB    #177,R5         ;PATTERN EXCEEDED MAX?
3975  020244  001766                        BEQ     BLDINC          ;YES-RESET IT.
3976  020246  000767                        BR      BLDINA          ;NO-ISSUE CURRENT PATTERN.
3977
3978
3979                              ;*****************************************************************
3980
3981                                      ;SUBROUTINE TO FILL THE SCREEN WITH INCREMENTING DATA
3982                              ;*****************************************************************
3983
3984  020250  042737  077577  002260  DATSC:   BIC     #77577,VSTAT            ;CLEAR INTERRUPT FLAGS.
3985  020256  013701  016262                    MOV     TBBUF,R1
3986  020262  012703  000500                    MOV     #320.,R3        ;FILL XMIT BUFFER WITH INCRE-
3987  020266  004037  020222                    JSR     R0,BLDINC       ;MENTING PATTERN
3988  020272  012737  003600  016270  10$:      MOV     #TOTCH,XMCNT    ;SET UP TO XMIT 1920 BYTES
3989  020300  052777  040000  161454            BIS     #XENA,@VJCSR
3990
3991  020306  032737  000001  002260  1$:       BIT     #XMDNE,VSTAT    ;XMIT DONE?
3992  020314  001774                        BEQ     .-6             ;NO
3993
3994                              ;*****************************************************************
3995
3996                                      ;SUBROUTINE TO RESET VT61 AND DISPLAY MESSAGE
3997                                      ;POINTED TO BY R2.
3998
3999                              ;*****************************************************************
4000
4001  020316  004037  016272      DSMES:    JSR     R0,RESETV       ;RESET THE UNIT AND WAIT FOR XON.
4002  020322  042737  077577  002260            BIC     #77577,VSTAT    ;CLEAR ALL FLAGS EXCEPT XOFF AND XMKIL.
4003  020330  012737  000005  016270            MOV     #5,XMCNT        ;PRE-LOAD XMIT COUNT.
```

```
 4004  020336  013701  016262                    MOV     TBBUF,R1        ;LOAD XMIT BUFFER WITH:
 4005  020342  012721  000002                    MOV     #SOM,(R1)+      ;START OF MESSAGE
 4006  020346  013721  002110                    MOV     ESCO,(R1)+
 4007  020352  013721  002044                    MOV     DRECT,(R1)+     ;DISABLE RECTANGULAR MODE
 4008  020356  005237  016270          1$:       INC     XMCNT           ;INCREMENT TRANSMIT COUNT
 4009  020362  112221                            MOVB    (R2)+,(R1)+     ;DISPLAY MESSAGE
 4010  020364  001374                            BNE     1$
 4011  020366  112711  000004                    MOVB    #EOM,(R1)       ;TERMINATE WITH END OF MESSAGE.
 4012  020372  052777  040000  161362            BIS     #XENA,@VJCSR    ;XMIT IT AND WAIT FOR
 4013  020400  032737  000001  002260  2$:       BIT     #XMDNE,VSTAT    ;DONE
 4014  020406  001774                            BEQ     2$
 4015  020410  000200                            RTS     R0
 4016
 4017                                    ;*****************************************************************
 4018
 4019                                            ;SUBROUTINE TO CONVERT A BINARY CHARACTER
 4020                                            ;TO 3 OCTAL CHARACTERS. R1 CONTAINS BINARY
 4021                                            ;NUMBER. RESULT IS STORED IN LOCATIONS SVER1,
 4022                                            ;SVER2
 4023
 4024                                    ;*****************************************************************
 4025  020412                            BINOCT:
 4026  020412  010546                            MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 4027  020414  012705  000002                    MOV     #2,R5           ;LOAD ITERATION COUNT
 4028  020420  000403                            BR      2$              ;BYPASSS SHIFTS FOR 1ST CONVERSION
 4029  020422  106201                    1$:     ASRB    R1
 4030  020424  106201                            ASRB    R1              ;SHIFT A CHAR INTO POSITION
 4031  020426  106201                            ASRB    R1
 4032  020430  110165  002220          2$:       MOVB    R1,SVER1(R5)    ;STORE THE BINARY OFFSET
 4033  020434  142765  000370  002220            BICB    #370,SVER1(R5)  ;CLEAR NON ESSENTIAL BITS
 4034  020442  152765  000060  002220            BISB    #60,SVER1(R5)   ;CONVERT OFFSET TO OCTAL
 4035  020450  005305                            DEC     R5              ;DECREMENT CONVERSION COUNT
 4036  020452  100363                            BPL     1$              ;NOT DONE CONVERT ANOTHER
 4037  020454  112737  000040  002223            MOVB    #40,SVER2+1     ;LOAD A SPACE
 4038  020462  012605                            MOV     (SP)+,R5        ;;POP STACK INTO R5
 4039  020464  000200                            RTS     R0
 4040
 4041                                    ;*****************************************************************
 4042                                            ;SUBROUTINE TO CONVERT AN OCTAL CHAR. TO BINARY.  REG
 4043                                            ;R1 CONTAINS OCTAL AND REG R2 IS BINARY ASSEMBLY AREA.
 4044                                    ;*****************************************************************
 4045
 4046  020466  042701  177770           OCTBIN: BIC     #177770,R1      ;EXTRACT OCTAL COMPONENT
 4047  020472  005702                            TST     R2              ;FIRST CONVERSION?
 4048  020474  001403                            BEQ     NOSHFT          ;YES - DO NOT SHIFT
 4049  020476  006302                            ASL     R2              ;NO - SHIFT PREVIOUS CHAR.
 4050  020500  006302                            ASL     R2
 4051  020502  006302                            ASL     R2
 4052  020504  060102                    NOSHFT: ADD     R1,R2           ;ADD CURRENT CHAR.
 4053  020506  000200                            RTS     R0
 4054                                    ;*****************************************************************
 4055                                            ;SUBROUTINE TO CONVERT A BINARY POSITION TO A OCTAL #.
 4056                                    ;*****************************************************************
 4057  020510  005037  001776           CONVLN: CLR     OCTLNE
 4058  020514  013737  001774  002246            MOV     TSTLNE,TPREG    ;LOAD OCTLNE WITH OCTAL EQUIVALENT
 4059  020522  032737  000001  002246  101$:     BIT     #BIT00,TPREG    ;OF TSTLNE IN BITS 8 THRU 11.
```

```
 4060  020530  001005                        BNE     102$
 4061  020532  006037  002246                ROR     TPREG
 4062  020536  105237  001777                INCB    OCTLNE+1
 4063  020542  000767                        BR      101$
 4064  020544  000200              102$:     RTS     R0
 4065
 4066                                ;*****************************************************************
 4067                                        ;ROUTINE TO WAIT FOR C/R FROM VT61 UNDER TEST
 4068                                ;*****************************************************************
 4069
 4070  020546  032777  000200  161206  GTCR:   BIT     #RECDN,@VJCSR   ;WAIT FOR REVEIVE DONE
 4071  020554  001774                        BEQ     .-6
 4072  020556  127737  161202  002006        CMPB    @VRBUF,CARRT    ;CHAR = CARRIAGE RETURN?
 4073  020564  001370                        BNE     GTCR            ;NO-KEEP LOOKING
 4074  020566  000200                        RTS     R0              ;YES-EXIT
 4075
 4076
 4077                                ;*****************************************************************
 4078                                        ;SUBROUTINE TO GET A CHARACTER (NUMERIC) FROM THE
 4079                                        ;CONSOLE.  IF OTHER THAN A NUMERIC IS TYPED A
 4080                                        ;"?" WILL BE ECHOED.
 4081                                ;*****************************************************************
 4082
 4083  020570  004037  014510        GTNUM:  JSR     R0,CONRD        ;GET A CHAR
 4084  020574  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 4085  020576  122701  000054                CMPB    #54,R1          ;CHAR. =COMMA?
 4086  020602  001411                        BEQ     1$              ;YES-GO PRINT IT
 4087  020604  123701  002006                CMPB    CARRT,R1        ;CHAR. = CARRIAGE RETURN?
 4088  020610  001406                        BEQ     1$
 4089  020612  120127  000060                CMPB    R1,#60
 4090  020616  103421                        BLO     QUST            ;IF CHAR. IS LESS THAN 60
 4091  020620  120127  000067                CMPB    R1,#67          ;OR MORE THAN 67, TYPE
 4092  020624  101016                        BHI     QUST            ;A QUESTION MARK
 4093  020626  110137  020676        1$:     MOVB    R1,TYPNUM
 4094  020632  104400  020676                TYPE    ,TYPNUM
 4095  020636  123701  002010                CMPB    LNFED,R1
 4096  020642  001406                        BEQ     GTEXT
 4097  020644  123701  002006                CMPB    CARRT,R1        ;IF CHAR. - C/R SET UP TO ISSUE
 4098  020650  001003                        BNE     GTEXT           ;LINE FEED BEFORE EXITING.
 4099  020652  113701  002010                MOVB    LNFED,R1
 4100  020656  000763                        BR      1$
 4101  020660  000200              GTEXT:    RTS     R0              ;GOOD CHAR., EXIT
 4102  020662  112737  000077  020676  QUST:   MOVB    #77,TYPNUM
 4103  020670  104400  020676                TYPE    ,TYPNUM         ;TYPE QUESTION MARK AND
 4104  020674  000735                        BR      GTNUM           ;KEEP LOOKING.
 4105  020676     000              TYPNUM:  .BYTE   0
 4106  020677     000                       .BYTE   0
 4107
 4108
 4109                                ;*****************************************************************
 4110                                        ;SUBROUTINE TO CALCULATE CHECKSUM ON THE LOWER
 4111                                        ;BYTE OF R5.  R4 IS STORAGE FOR THE CHECKSUM
 4112                                        ;CHARACTER. ALGORITHM FOR CHECKSUM IS ROTATE
 4113                                        ;CURRENT ONE PLACE LEFT AND XOR NEW CHAR. CHECKSUM
 4114                                        ;IS THE LOWER 7 BITS OF R4
 4115
```

```
                                        ;*****************************************************************
                                        ;
020700  042705  177400          CALCK:  BIC     #177400,R5      ;CLEAR UPPER BYTE OF R5
020704  120527  000021                  CMPB    R5,#XON         ;CHAR. =XON?
020710  001415                          BEQ     NOCALC          ;YES DO NOT CALCULATE CHECKSUM
020712  120527  000023                  CMPB    R5,#XOFF        ;CHAR =XOFF?
020716  001412                          BEQ     NOCALC          ;YES DO NOT CALCULATE CHECKSUM
                                        ;
020720  000241                          CLC                     ;INSURE CARRY BIT INITIALLLY CLEAR
020722  105704                          TSTB    R4              ;SET UP TO ROTATE R4
020724  100001                          BPL     1$              ;A FULL 8 BYTES
                                        ;
020726  000261                          SEC                     ;R4 WAS NEG. SO ROTATE A ONE
020730  106104                  1$:     ROLB    R4              ;INTO LOW ORDER BIT.
020732  010403                          MOV     R4,R3
020734  040503                          BIC     R5,R3           ;NOT A AND B
020736  040405                          BIC     R4,R5           ;NOT B AND A
020740  050305                          BIS     R3,R5           ;ORED
020742  010504                          MOV     R5,R4           ;EQUAL NEW CHECKSUM
020744  000200                  NOCALC: RTS     R0
                                        ;*****************************************************************
                                        ;
                                                ;SUBROUTINE TO LOAD XMIT BUFFER FROM R0 THRU R1
                                        ;*****************************************************************
                                        ;
020746  112021                  LDBUF:  MOVB    (R0)+,(R1)+     ;LOAD A BYTE
020750  001376                          BNE     .-2             ; UNTIL ZERO BYTE FOUND.
020752  000200                          RTS     R0
                                        ;*****************************************************************
                                                ;SUBROUTINE TO CHECK THE VT61 FOR A PERIPHERAL ABORT.
                                        ;*****************************************************************
                                        ;
020754  032737  010000  002260  CKABRT: BIT     #PABRT,VSTAT    ;ABORT FLAG RECEIVED?
020762  001446                          BEQ     2$              ;NO-EXIT
020764  010037  001124                  MOV     R0,$GDDAT
020770  162737  000004  001124          SUB     #4,$GDDAT       ;POINT ERR PC TO MAIN ROUTINE.
020776  013737  002260  001126          MOV     VSTAT,$BDDAT
021004  104020                          ERROR   20              ;ISSUE PERIPHERAL ABORT ERROR
                                        ;
021006  013701  016262                  MOV     TBBUF,R1
021012  004037  020746                  JSR     R0,LDBUF        ;LOAD THE XMIT BUFFER WITH:
021016      033     117     137         .BYTE   .ESC,.0,.IABT,.ESC,.0,.RABT
021021      033     117     140
021024      033     117     145         .BYTE   .ESC,.0,.UNLKKB,.BEL,0,0
021027      007     000     000
021032  012737  000007  016270          MOV     #7,XMCNT        ;SET UP TO XMIT 7 BYTES.
021040  004037  016676                  JSR     R0,XMREC        ;XMIT AND RECEIVE.
021044  000240                          NOP
021046  123727  016010  000170          CMPB    STR0,#NABRT     ;ABORT FLAG CLEARED?
021054  001411                          BEQ     2$              ;YES-EXIT
021056  010037  001124                  MOV     R0,$GDDAT       ;NO-SET UP AND ISSUE A CANT
021062  162737  000004  001124          SUB     #4,$GDDAT       ;CLEAR ABORT FLAG ERROR MESSAGE.
021070  013737  002260  001126          MOV     VSTAT,$BDDAT
021076  104021                          ERROR   21
021100  000200                  2$:     RTS     R0
                                        ;
```

```
                                ;*****************************************************************
                                ;SUBROUTINE TO COMPARE RECEIVED KEYBOARD DATA WITH
                                ;DATA EXPECTED. ERRORS ARE REPORTED AS POSITIONAL
                                ;ERROR ONLY.
                                ;*****************************************************************

021102  105077  011212          CKKBD:  CLRB    @ABUFP          ;CLEAR RECEIVE BYTE
021106  005037  002216                  CLR     CHRD            ;CLEAR INPUT STORAGE.
021112  105777  011202          KBDLP:  TSTB    @ABUFP          ;WAIT FOR A INPUT.
021116  001775                  1$:     BEQ     .-4

021120  117737  011174  002216          MOVB    @ABUFP  ,CHRD   ;STORE IT AND
021126  105077  011166                  CLRB    @ABUFP          ;CLEAR THE INPUT AREA.
021132  123714  002216                  CMPB    CHRD    ,(R4)   ;RECEIVED EQUAL EXPECTED?
021136  001500                          BEQ     GDSTRK          ;NO-UPDATE POINTERS.
021140  005237  002244                  INC     BUBCT           ;INCREMENT ERROR COUNT.
021144  023727  002244  000012          CMP     BUBCT,#10.      ;COUNT = 10?
021152  103075                          BHIS    CNTF            ;YES-EXIT SUBROUTINE.
021154  010401                          MOV     R4,R1
021156  166501  012130                  SUB     DT1BL(R5),R1    ;EXTRACT KEY POSITION FROM ROW LOC.
021162  005201                          INC     R1              ;CONVERT LOGICAL POS. TO ACTUAL.
021164  004037  020412                  JSR     R0,BINOCT       ;GET KEY POSITION IN OCTAL.
021170  113737  002222  002220          MOVB    SVER2,SVER1     ;RE-ASSEMBLE OCTAL BYTES.
021176  123727  002221  000060          CMPB    SVER1+1,#60     ;POSITION LESS THAN 8?
021204  001413                          BEQ     LDPOS           ;YES-GO LOAD IT.
021206  123727  002220  000062          CMPB    SVER1,#62       ;POSITION GREATER THAN 8 AND LESS THAN12?
021214  103404                          BLO     BOROW           ;YES-SET UP TO BORROW.
021216  162737  000002  002220          SUB     #2,SVER1        ;NO-JUST SUBTRACT 2.
021224  000403                          BR      LDPOS
021226  162737  000370  002220  BOROW:  SUB     #370,SVER1      ;SUBTRACT AND BORROW.
021234  113737  002220  030731  LDPOS:  MOVB    SVER1,KYSTRK+1  ;LOAD THE CONVERTED DECIMAL #.
021242  113737  002221  030730          MOVB    SVER1+1,KYSTRK
021250  012703  030653          DMPOCT: MOV     #DKBERR ,R3
021254  004037  017222                  JSR     R0      ,LDXMIT ;ISSUE BODY OF KEYBOARD ERROR.
021260  111401                          MOVB    (R4)    ,R1
021262  004037  020412                  JSR     R0      ,BINOCT
021266  012703  002220                  MOV     #SVER1  ,R3
021272  004037  017222                  JSR     R0      ,LDXMIT ;CONVERT AND ISSUE GOOD CHAR.
021276  012703  030762                  MOV     #DSPC6,R3
021302  004037  017222                  JSR     R0,LDXMIT       ;INSERT 6 SPACES IN MESSAGE.
021306  113701  002216                  MOVB    CHRD    ,R1
021312  004037  020412                  JSR     R0      ,BINOCT
021316  012703  002220                  MOV     #SVER1  ,R3
021322  004037  017222                  JSR     R0      ,LDXMIT ;CONVERT AND ISSUE RECEIVED CHAR.
021326  012703  001167                  MOV     #$CRLF  ,R3
021332  004037  017222                  JSR     R0      ,LDXMIT ;ISSUE C/R AND L/F.
021336  000665                          BR      KBDLP           ;LOOK FOR SAME KEY AGAIN.

021340  005204          GDSTRK: INC     R4              ;INCREMENT KEYBOARD ROW COUNTER.
021342  105714                  TSTB    (R4)            ;REACHED END OF ROW?
021344  001262                  BNE     KBDLP           ;NO-LOOK FOR NEXT INPUT
021346  000200          CNTF:   RTS     R0              ;YES-EXIT.

                                ;*****************************************************************
```

```
 4228                                           ;SUBROUTINE TO LOOP DATA THROUGH HOST COMPUTER. ALL
 4229                                           ;FUNCTIONS ARE ALLOWED, BUT BLOCK TRANSMITS WHICH
 4230                                           ;EXCEED 552 BYTES WILL RESULT IN THE TERMINATION
 4231                                           ;OF THE OPERATION AFTER 552 RECEIVED BYTES.
 4232
 4233                                         ;*****************************************************************
 4234
 4235  021350  005237  021652          LOOP:   INC     XMZER                   ;SET UP TO XMIT NULLS.
 4236  021354  012737  032320  015756          MOV     #TCRLB+500,REBUF                ;RESET BUFFER POINTERS
 4237  021362  012737  031120  016262          MOV     #RCRLB,TBBUF
 4238  021370  004037  017262                  JSR     R0,RESPTR               ;RELOAD ALL INTERRUPT POINTERS
 4239  021374  042737  077577  002260          BIC     #77577,VSTAT            ;CLEAR ALL FLAGS BUT XOFF AND XMKIL.
 4240  021402  013704  015760          LOOPT:  MOV     RBUFP,R4                ;SET UP RECEIVE FLAG
 4241  021406  032737  000001  002260  LOOPTA: BIT     #XMDNE,VSTAT            ;XMIT COMPLETE?
 4242  021414  001407                  LOOPR   BEQ     LOOPR                   ;NO
```

```
 4243  021416  042737  000001  002260          BIC     #XMDNE,VSTAT    ;YES RESET FLAG
 4244  021424  013737  015754  015760          MOV     RBBUF,RBUFP     ;RESET THE REC. BUFFER POINTER
 4245  021432  000763                          BR      LOOPT
 4246  021434  032737  001400  002260  LOOPR:  BIT     #EPL+ESC,VSTAT  ;RECEIVED AN ESC OR EPL?
 4247  021442  001004                          BNE     LPSTR           ;YES-GO CHECK IT
 4248  021444  023704  015760                  CMP     RBUFP,R4        ;RECEIVED A DISPLAY CHAR?
 4249  021450  001756                          BEQ     LOOPTA          ;NO-LOOP
 4250  021452  000426                          BR      BUMPCT
 4251  021454  117777  010640  174276  LPSTR:  MOVB    @ABUFP,@RBUFP   ;YES LOAD IT IN THE BUFFER
 4252  021462  005237  015760                  INC     RBUFP           ;AND INCREMENT BUFFER POINTER
 4253  021466  005037  015762                  CLR     ESAMB           ;CLEAR ESC ASSEMBLY WORD
 4254  021472  042737  001400  002260          BIC     #EPL+ESC,VSTAT  ;CLEAR THE FLAGS
 4255  021500  005237  016270                  INC     XMCNT           ;INCREMENT XMIT COUNT
 4256  021504  123777  002166  010606          CMPB    ESCN,@ABUFP     ;CHAR. A ESC(033)?
 4257  021512  001733                          BEQ     LOOPT           ;YES WAIT FOR NEXT PART OF FUNCTION
 4258  021514  113777  002010  174236          MOVB    LNFED,@RBUFP    ;CHAR. WAS EPL ADD A LINE FEED.
 4259  021522  005237  015760                  INC     RBUFP
 4260  021526  000407                          BR      FRCECT          ;AND ISSUE THEM.
 4261  021530  023727  016270  000764  BUMPCT: CMP     XMCNT,#500.     ;BUFFER ABOUT FILLED?
 4262  021536  103403                          BLO     FRCECT          ;NO
 4263  021540  005337  015760          1$:     DEC     RBUFP           ;YES-RESET THE RECEIVE POINTER
```

```
4264  021544  000716                        BR      LOOPT
4265  021546  005237  016270                FRCECT: INC     XMCNT           ;INCREMENT THE XMIT COUNT
4266  021552  023727  016270  000002              CMP     XMCNT,#2        ;FIRST CHAR TO XMIT?
4267  021560  101003                              BHI     XMWT            ;NO
4268  021562  052777  040000  160172              BIS     #XENA,@VJCSR    ;YES-SET THE XMIT ENABLE
4269  021570  004037  021600                XMWT:   JSR     R0,EXTST        ;LOOK FOR END OF TEST COMMAND.
4270  021574  000702                              BR      LOOPT           ;NONE FOUND.
4271  021576  000200                              RTS     R0              ;AND EXIT
4272
4273                                        ;***************************************************************
4274                                        ;SUBROUTINE TO CHECK FOR END OF TEST COMMAND. THE CONTROL
4275                                        ;C KEY EXITS ALL TESTS.
4276                                        ;***************************************************************
4277
4278  021600  127727  010514  000003  EXTST:  CMPB    @ABUFP,#3       ;LOOK FOR CONTROL C.
4279  021606  001020                          BNE     NOROUT
4280
4281  021610  012737  031617  015756  ABSXT:  MOV     #RCRLB+477,REBUF        ;RESET THE BUFFERS
4282  021616  012737  031620  016262          MOV     #TCRLB,TBBUF
4283  021624  004037  017262                  JSR     R0,RESPTR       ;RESET ALL POINTERS
4284  021630  012702  027541                  MOV     #DEXT,R2
4285  021634  004037  020316                  JSR     R0,DSMES        ;ISSUE EXIT MESSAGE
4286  021640  005037  021652                  CLR     XMZER           ;CLEAR THE ZERO TRANSMIT FLAG.
4287  021644  062700  000002                  ADD     #2,R0           ;SET UP TEST EXIT.
4288  021650  000200                NOROUT:   RTS     R0              ;EXIT SUBROUTINE.
4289
4290  021652  000000                XMZER:    .WORD   0
4291                                        ;***************************************************************
4292                                        ;SUB-ROUTINE TO LOOK FOR VSTAT BIT ON THE STACK
4293                                        ;DELAY FACTOR IS FIRST WORD ON THE STACK AND VSTAT BIT
4294                                        ;IS THE SECOND. MIN. DELAY IS 4 U.S FOR A MOS 11/45.
4295                                        ;***************************************************************
4296
4297  021654                                WTBGND:
4298  021654  012637  002256                          MOV     (SP)+,ROSVE     ;;POP STACK INTO ROSVE
4299  021660  012637  021742                          MOV     (SP)+,VDLAY     ;;POP STACK INTO VDLAY
4300  021664  012637  021740                          MOV     (SP)+,VBIT      ;;POP STACK INTO VBIT
4301  021670  005037  002254                1$:       CLR     DLAY
4302  021674  033737  021740  002260        2$:       BIT     VBIT,VSTAT      ;SENSED THE CONDITION?
4303  021702  001012                                  BNE     FNDBT           ;YES-EXIT.
4304  021704  005337  002254                          DEC     DLAY            ;NO-RUN DELAY.
4305  021710  001371                                  BNE     2$
4306  021712  005337  021742                          DEC     VDLAY           ;DELAY FACTOR EXPIRED?
4307  021716  001364                                  BNE     1$              ;NO-LOOP
4308  021720  104011                                  ERROR   11              ;DELAY EXPIRED-ISSUE HUNG NIT
4309  021722  005237  002234                          INC     FTLCNT          ;INCREMENT FATAL XMIT COUNT.
4310  021726  000401                                  BR      TIMEXT
4311  021730  005720                FNDBT:    TST     (R0)+           ;SET UP FOR NORMAL EXIT
4312  021732                          TIMEXT:
4313  021732  013746  002256                          MOV     ROSVE,-(SP)     ;;PUSH ROSVE ON STACK
4314  021736  000200                                  RTS     R0
4315  021740  000000                VBIT:     0
4316  021742  000000                VDLAY:    0
4317                                        ;***************************************************************
4318                                        ;SUBROUTINE TO LOOK FOR XOFF BEFORE EXITING A RECEIVE ROUTINE.
4319                                        ;***************************************************************
```

```
                                ;*****************************************************************

                                .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

021770                          $SCOPE:
021770  004037  014542          1$:     JSR     R0,MONIT
021774  032777  040000  157134          BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
022002  001111                  2$:     BNE     $OVER           ;;YES IF SW14=1
                                ;*****START OF CODE FOR THE XOR TESTER*****
022004  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
022006  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
022012  012737  022032  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
022020  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
022024  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
022030  000463                          BR      $SVLAD          ;;GO TO THE NEXT TEST
022032  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
022034  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
022040  000423                          BR      7$              ;;LOOP ON THE PRESENT TEST
022042                          6$:;*****END OF CODE FOR THE XOR TESTER*****
022042  032777  000400  157066          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
022050  001404                          BEQ     2$              ;;BR IF NO
022052  127737  157060  001102          CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
022060  001462                          BEQ     $OVER           ;;BR IF YES
022062  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
022066  001421                          BEQ     3$              ;;BR IF NO
022070  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
022076  101015                          BHI     3$              ;;BR IF NO
022100  032777  001000  157030          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
022106  001404                          BEQ     4$              ;;BR IF NO
022110  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
022116  000443                          BR      $OVER
022120  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
022124  005037  001156                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
022130  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
022132  032777  004000  156776  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
022140  001011                          BNE     1$              ;;BR IF YES
```

```
4376  022142  005737  001100                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
4377  022146  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
4378  022150  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
4379  022154  023737  001156  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
4380  022162  002021                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
4381  022164  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
4382  022172  013737  022242  001156          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
4383  022200  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
4384  022204  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
4385  022210  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
4386  022214  005037  001160                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
4387  022220  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
4388  022226  013777  001102  156704  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
4389  022234  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
4390  022240  000002                          RTI                     ;;FIXES PS
4391  022242  000005                  $MXCNT: 5                       ;;MAX. NUMBER OF ITERATIONS
4392                                  ;*****************************************************************
4393
4394                                  .SBTTL  ERROR HANDLER ROUTINE
4395
4396                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
4397                                  ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
4398                                  ;*AND GO TO $ERRTYP ON ERROR
4399                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
4400                                  ;*SW15=1         HALT ON ERROR
4401                                  ;*SW13=1         INHIBIT ERROR TYPEOUTS
4402                                  ;*SW10=1         BELL ON ERROR
4403                                  ;*SW09=1         LOOP ON ERROR
4404                                  ;*CALL
4405                                  ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
4406
4407  022244                          $ERROR:
4408  022244  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
4409  022250  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
4410  022252  013777  001102  156660          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
4411  022260  032777  002000  156650          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
4412  022266  001402                          BEQ     1$              ;;NO - SKIP
4413  022270  104400  001162                  TYPE    ,$BELL          ;;RING BELL
4414  022274  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
4415  022300  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
4416  022304  162737  000002  001116          SUB     #2,$ERRPC
4417  022312  117737  156600  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
4418  022320  032777  020000  156610          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
4419  022326  001004                          BNE     20$             ;;SKIP TYPEOUTS
4420  022330  004737  022626                  JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
4421  022334  104400  001167                  TYPE    ,$CRLF
4422  022340                          20$:
4423  022340  005777  156572          2$:     TST     @SWR            ;;HALT ON ERROR
4424  022344  100006                          BPL     3$              ;;SKIP IF CONTINUE
4425  022346  000000                          HALT                    ;;HALT ON ERROR!
4426  022350  022737  011052  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
4427  022356  001001                          BNE     3$              ;;BRANCH IF NO
4428  022360  000000                          HALT                    ;;YES
4429  022362  032777  001000  156546  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
4430  022370  001402                          BEQ     4$              ;;BR IF NO
4431  022372  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
```

```
4432  022376  005737  001160          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
4433  022402  001402                          BEQ     5$              ;;BR IF NONE
4434  022404  013716  001160                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
4435  022410                          5$:
4436  022410  000002                          RTI                     ;;RETURN
4437                                  ;*****************************************************************
4438
4439                                  .SBTTL  TYPE ROUTINE
4440
4441                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
4442                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
4443                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
4444                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
4445                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
4446                                  ;*
4447                                  ;*CALL:
4448                                  ;*1) USING A TRAP INSTRUCTION
4449                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
4450                                  ;*OR
4451                                  ;*      TYPE
4452                                  ;*      MESADR
4453                                  ;*
4454
4455  022412  105737  001155          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
4456  022416  100002                          BPL     1$              ;;BR IF YES
4457  022420  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
4458  022422  000407                          BR      3$              ;;LEAVE
4459  022424  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
4460  022426  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
4461  022432  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
4462  022434  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
4463  022436  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
4464  022440  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
4465  022442  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
4466  022446  000002                          RTI                     ;;RETURN
4467  022450  122716  000011          4$:     CMPB    #THT,(SP)       ;;BRANCH IF <HT>
4468  022454  001426                          BEQ     8$
4469  022456  122716  000200                  CMPB    #TCRLF,(SP)     ;;BRANCH IF NOT <CRLF>
4470  022462  001004                          BNE     5$
4471  022464  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
4472  022466  104400                          TYPE                    ;;TYPE A CR AND LF
4473  022470  001167                          $CRLF
4474  022472  000757                          BR      2$              ;;GET NEXT CHARACTER
4475  022474  004737  022556          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
4476  022500  123726  001154          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
4477  022504  001352                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
4478  022506  013746  001152                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
4479                                                                  ;;AND THE NULL CHAR.
4480  022512  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
4481  022516  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
4482  022520  004737  022556                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
4483  022524  105337  022622                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
4484  022530  000770                          BR      7$              ;;LOOP
4485
4486                                  ;HORIZONTAL TAB PROCESSOR
4487
```

```
4488  022532  112716  000040          9$:     MOVB    #40,(SP)        ;;REPLACE TAB WITH SPACE
4489  022536  004737  022556          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
4490  022542  132737  000007  022622          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
4491  022550  001372                          BNE     9$              ;;TAB STOP
4492  022552  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
4493  022554  000726                          BR      2$              ;;GET NEXT CHARACTER
4494  022556  105777  156364          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
4495  022562  100375                          BPL     $TYPEC
4496  022564  116677  000002  156356          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
4497  022572  122766  000015  000002          CMPB    #15,2(SP)       ;;BRANCH IF
4498  022600  001003                          BNE     1$              ;;NOT <CR>
4499  022602  105037  022622                  CLRB    $CHARCNT        ;;
4500  022606  000406                          BR      $TYPEX          ;;EXIT
4501  022610  122766  000012  000002  1$:     CMPB    #12,2(SP)       ;;BRANCH IF
4502  022616  002002                          BGE     $TYPEX          ;;<LF>
4503  022620  105227                          INCB    (PC)+           ;;INC SPACE
4504  022622  000000          $CHARCNT:.WORD  0                       ;;COUNT
4505  022624  000207          $TYPEX: RTS     PC
4506                          ;;              EQUATES
4507          000011          THT=11
4508          000200          TCRLF=200
4509
4510                          ;*******************************************************************
4511
4512                          .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
4513
4514                          ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
4515                          ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
4516                          ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
4517
4518  022626          $ERRTYP:
4519  022626  104400  001167          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
4520  022632  010046                  MOV     R0,-(SP)        ;;SAVE R0
4521  022634  005000                  CLR     R0              ;;PICKUP THE ITEM INDEX
4522  022636  153700  001114          BISB    @#$ITEMB,R0
4523  022642  001004                  BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
4524                                                          ;;TYPE THE PC OF THE ERROR
4525  022644  013746  001116          MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
4526                                                          ;;ERROR ADDRESS
4527  022650  104401                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
4528  022652  000445                  BR      10$             ;;GET OUT
4529  022654  005300          1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
4530  022656  006300                  ASL     R0              ;;      WORK FOR THE ERROR TABLE
4531  022660  006300                  ASL     R0
4532  022662  006300                  ASL     R0
4533  022664  062700  001172          ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
4534  022670  012037  022700          MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
4535  022674  001404                  BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
4536  022676  104400                  TYPE                    ;;TYPE THE "ERROR MESSAGE"
4537  022700  000000          2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
4538  022702  104400  001167          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
4539  022706  012037  022716  3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
4540  022712  001404                  BEQ     5$              ;;SKIP TYPEOUT IF 0
4541  022714  104400                  TYPE                    ;;TYPE THE "DATA HEADER"
4542  022716  000000          4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
4543  022720  104400  001167          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
```

```
4544  022724  010146                5$:     MOV     R1,-(SP)        ;;SAVE R1
4545  022726  012001                        MOV     (R0)+,R1        ;;PICKUP "DATA TABLE" POINTER
4546  022730  001415                        BEQ     9$              ;;BR IF NO DATA TO BE TYPED
4547  022732  012000                        MOV     (R0)+,R0        ;;PICKUP "DATA FORMAT" POINTER
4548  022734  105720                6$:     TSTB    (R0)+           ;;"OCTAL" OR "DECIMAL"
4549  022736  001003                        BNE     7$              ;;BR IF DECIMAL
4550  022740  013146                        MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
4551  022742  104401                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
4552  022744  000402                        BR      8$
4553  022746                        7$:
4554  022746  013146                        MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
4555  022750  104404                        TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
4556  022752  005711                8$:     TST     (R1)            ;;IS THERE ANOTHER NUMBER?
4557  022754  001403                        BEQ     9$              ;;BR IF NO
4558  022756  104400  022776                TYPE    ,11$            ;;TYPE TWO(2) SPACES
4559  022762  000764                        BR      6$              ;;LOOP
4560
4561  022764  012601                9$:     MOV     (SP)+,R1        ;;RESTORE R1
4562  022766  012600                10$:    MOV     (SP)+,R0        ;;RESTORE R0
4563  022770  104400  001167                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
4564  022774  000207                        RTS     PC              ;;RETURN
4565  022776  020040     000        11$:    .ASCIZ  /  /            ;;TWO(2) SPACES
4566          023002                        .EVEN
4567                                ;*****************************************************************
4568
4569                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
4570
4571                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
4572                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
4573                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
4574                                ;*CALL:
4575                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
4576                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
4577                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
4578                                ;*      .BYTE   M               ;;M=1 OR 0
4579                                ;*                                      ;;1=TYPE LEADING ZEROS
4580                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
4581                                ;*
4582                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
4583                                ;*$TYPOS OR $TYPOC
4584                                ;*CALL:
4585                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
4586                                ;*      TYPON                   ;;CALL FOR TYPEOUT
4587                                ;*
4588                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
4589                                ;*CALL:
4590                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
4591                                ;*      TYPOC                   ;;CALL FOR TYPEOUT
4592
4593  023002  017646  000000        $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
4594  023006  116637  000001  023225        MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
4595  023014  112637  023227                MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
4596  023020  062716  000002                ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
4597  023024  000406                        BR      $TYPON
4598  023026  112737  000001  023225  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
4599  023034  112737  000006  023227        MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
```

```
 4600  023042  112737  000005  023224  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
 4601  023050  010346                  MOV     R3,-(SP)        ;;SAVE R3
 4602  023052  010446                  MOV     R4,-(SP)        ;;SAVE R4
 4603  023054  010546                  MOV     R5,-(SP)        ;;SAVE R5
 4604  023056  113704  023227          MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 4605  023062  005404                  NEG     R4
 4606  023064  062704  000006          ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 4607  023070  110437  023226          MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 4608  023074  113704  023225          MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 4609  023100  016605  000012          MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 4610  023104  005003                  CLR     R3              ;;CLEAR THE OUTPUT WORD
 4611  023106  006105          1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 4612  023110  000404                  BR      3$              ;;GO DO MSB
 4613  023112  006105          2$:     ROL     R5              ;;FORM THIS DIGIT
 4614  023114  006105                  ROL     R5
 4615  023116  006105                  ROL     R5
 4616  023120  010503                  MOV     R5,R3
 4617  023122  006103          3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 4618  023124  105337  023226          DECB    $OMODE          ;;TYPE THIS DIGIT?
 4619  023130  100016                  BPL     7$              ;;BR IF NO
 4620  023132  042703  177770          BIC     #177770,R3      ;;GET RID OF JUNK
 4621  023136  001002                  BNE     4$              ;;TEST FOR 0
 4622  023140  005704                  TST     R4              ;;SUPPRESS THIS 0?
 4623  023142  001403                  BEQ     5$              ;;BR IF YES
 4624  023144  005204          4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 4625  023146  052703  000060          BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 4626  023152  052703  000040          BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 4627  023156  110337  023222  5$:     MOVB    R3,8$           ;;SAVE FOR TYPING
 4628  023162  104400  023222          TYPE    ,8$             ;;GO TYPE THIS DIGIT
 4629  023166  105337  023224  7$:     DECB    $OCNT           ;;COUNT BY 1
 4630  023172  003347                  BGT     2$              ;;BR IF MORE TO DO
 4631  023174  002402                  BLT     6$              ;;BR IF DONE
 4632  023176  005204                  INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 4633  023200  000744                  BR      2$              ;;GO DO THE LAST DIGIT
 4634  023202  012605          6$:     MOV     (SP)+,R5        ;;RESTORE R5
 4635  023204  012604                  MOV     (SP)+,R4        ;;RESTORE R4
 4636  023206  012603                  MOV     (SP)+,R3        ;;RESTORE R3
 4637  023210  016666  000002  000004  MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 4638  023216  012616                  MOV     (SP)+,(SP)
 4639  023220  000002                  RTI                     ;;RETURN
 4640  023222     000          8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 4641  023223     000                  .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 4642  023224     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
 4643  023225     000          $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
 4644  023226  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
 4645                          ;******************************************************************
 4646
 4647                          .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
 4648
 4649                          ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 4650                          ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 4651                          ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 4652                          ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 4653                          ;*REPLACED WITH SPACES.
 4654                          ;*CALL:
 4655                          ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
```

```
                                        ;*      TYPDS                   ;;GO TO THE ROUTINE
                                        $TYPDS:
023230                                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
023230  010046                                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
023232  010146                                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
023234  010246                                 MOV     R3,-(SP)        ;;PUSH R3 ON STACK
023236  010346                                 MOV     R5,-(SP)        ;;PUSH R5 ON STACK
023240  010546                                 MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
023242  012746  020200                         MOV     20(SP),R5       ;;GET THE INPUT NUMBER
023246  016605  000020                         BPL     1$              ;;BR IF INPUT IS POS.
023252  100004                                 NEG     R5              ;;MAKE THE BINARY NUMBER POS.
023254  005405                                 MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
023256  112766  000055  000001                 CLR     R0              ;;ZERO THE CONSTANTS INDEX
023264  005000                        1$:      MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
023266  012703  023444                         MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
023272  112723  000040                 2$:      CLR     R2              ;;CLEAR THE BCD NUMBER
023276  005002                                 MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
023300  016001  023434                 3$:      SUB     R1,R5           ;;FORM THIS BCD DIGIT
023304  160105                                 BLT     4$              ;;BR IF DONE
023306  002402                                 INC     R2              ;;INCREASE THE BCD DIGIT BY 1
023310  005202                                 BR      3$
023312  000774                         4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
023314  060105                                 TST     R2              ;;CHECK IF BCD DIGIT=0
023316  005702                                 BNE     5$              ;;FALL THROUGH IF 0
023320  001002                                 TSTB    (SP)            ;;STILL DOING LEADING 0'S?
023322  105716                                 BMI     7$              ;;BR IF YES
023324  100407                                 ASLB    (SP)            ;;MSD?
023326  106316                         5$:     BCC     6$              ;;BR IF NO
023330  103003                                 MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
023332  116663  000001  177777                 BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
023340  052702  000060                 6$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
023344  052702  000040                 7$:     MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
023350  110223                                 TST     (R0)+           ;;JUST INCREMENTING
023352  005720                                 CMP     R0,#10          ;;CHECK THE TABLE INDEX
023354  020027  000010                         BLT     2$              ;;GO DO THE NEXT DIGIT
023360  002746                                 BGT     8$              ;;GO TO EXIT
023362  003002                                 MOV     R5,R2           ;;GET THE LSD
023364  010502                                 BR      6$              ;;GO CHANGE TO ASCII
023366  000764                         8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
023370  105726                                 BPL     9$              ;;BR IF NO
023372  100003                                 MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
023374  116663  177777  177776                 CLRB    (R3)            ;;SET THE TERMINATOR
023402  105013                         9$:     MOV     (SP)+,R5        ;;POP STACK INTO R5
023404  012605                                 MOV     (SP)+,R3        ;;POP STACK INTO R3
023406  012603                                 MOV     (SP)+,R2        ;;POP STACK INTO R2
023410  012602                                 MOV     (SP)+,R1        ;;POP STACK INTO R1
023412  012601                                 MOV     (SP)+,R0        ;;POP STACK INTO R0
023414  012600                                 TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
023416  104400  023444                         MOV     2(SP),4(SP)     ;;ADJUST THE STACK
023422  016666  000002  000004                 MOV     (SP)+,(SP)
023430  012616                                 RTI                     ;;RETURN TO USER
023432  000002
023434  023420                         $DTBL:  10000.
023436  001750                                 1000.
023440  000144                                 100.
023442  000012                                 10.
```

```
4712  023444  000004                  $DBLK:  .BLKW   4
4713                                  ;*****************************************************************
4714
4715                                  .SBTTL  POWER DOWN AND UP ROUTINES
4716
4717                                  ;POWER DOWN ROUTINE
4718  023454  012737  023604  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
4719  023462  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
4720  023470  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
4721  023472  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
4722  023474  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
4723  023476  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
4724  023500  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
4725  023502  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
4726  023504  010637  023610                MOV     SP,$SAVR6       ;;SAVE SP
4727  023510  012737  023522  000024        MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
4728  023516  000000                        HALT
4729  023520  000776                        BR      .-2             ;;HANG UP
4730
4731                                  ;POWER UP ROUTINE
4732  023522  013706  023610          $PWRUP: MOV     $SAVR6,SP       ;;GET SP
4733  023526  005037  023610                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
4734  023532  005237  023610          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
4735  023536  001375                        BNE     1$              ;;OF  WORD
4736  023540  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
4737  023542  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
4738  023544  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
4739  023546  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
4740  023550  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
4741  023552  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
4742  023554  012737  023454  000024        MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
4743  023562  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
4744  023570  104400                        TYPE                    ;REPORT THE POWER FAILURE
4745  023572  023612          $PWRMG: .WORD   $POWER                  ;;POWER FAIL MESSAGE POINTER
4746  023574  042766  000020  000002  BIC     #20,2(SP)               ;;CLEAR "T" BIT
4747  023602  000002                        RTI
4748  023604  000000          $ILLUP: HALT                            ;;THE POWER UP SEQUENCE WAS STARTED
4749  023606  000776                        BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
4750  023610  000000          $SAVR6: 0                               ;;PUT THE SP HERE
4751  023612  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
4752  023620  000122
4753                                          .EVEN
4754                                  ;*****************************************************************
4755
4756                                  .SBTTL  TRAP DECODER
4757
4758                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
4759                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
4760                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
4761                                  ;*GO TO THAT ROUTINE.
4762
4763  023622  010046          $TRAP:  MOV     R0,-(SP)                ;;SAVE R0
4764  023624  016600  000002          MOV     2(SP),R0                ;;GET TRAP ADDRESS
4765  023630  005740                  TST     -(R0)                   ;;BACKUP BY 2
4766  023632  111000                  MOVB    (R0),R0                 ;;GET RIGHT BYTE OF TRAP
4767  023634  006300                  ASL     R0                      ;;POSITION FOR INDEXING
```

```
 4768  023636  016000  023644                  MOV     $TRPAD(RO),RO   ;;INDEX TO TABLE
 4769  023642  000200                          RTS     RO              ;;GO TO ROUTINE
 4770
 4771
 4772                                  .SBTTL  TRAP TABLE
 4773
 4774                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
 4775                                  ;*BY THE "TRAP" INSTRUCTION.
 4776
 4777                                  ;       ROUTINE
 4778                                  ;       -------
 4779  023644                          $TRPAD:
 4780  023644  022412                          $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
 4781  023646  023026                          $TYPOC  ;;CALL=TYPOC    TRAP+1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
 4782  023650  023002                          $TYPOS  ;;CALL=TYPOS    TRAP+2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
 4783  023652  023042                          $TYPON  ;;CALL=TYPON    TRAP+3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
 4784  023654  023230                          $TYPDS  ;;CALL=TYPDS    TRAP+4(104404)  TYPE DECIMAL NUMBER (WITH SIGN)
 4785
 4786  023656  003332  003724  004132  TSTADD: TST1,TST2,TST3
 4787
 4788  023664  004326  004460  004674          TST4,TST5,TST6
 4789
 4790  023672  005124  005700  006360          TST7,TST10,TST11
 4791
 4792  023700  006572  006746  007174          TST12,TST13,TST14
 4793
 4794  023706  007416  010104  010322          TST15,TST16,TST17
 4795
 4796  023714  010470  011174  011500          TST20,TST21,TST22
 4797
 4798  023722  011676  011750                  TST23,TST24
 4799
 4800
 4801
 4802  023726  042523  020124  052126  STUPM:  .ASCII  /SET VT61S TO  FULL DUPLEX, /<15><12>
 4803  023734  030466  020123  047524
 4804  023742  020040  052506  046114
 4805  023750  042040  050125  042514
 4806  023756  026130  006440    012
 4807  023763     071  030060  041060          .ASCIZ  /9600BAUD, REMOTE,PARITY MATCHED TO INTERFACE/<15><12>
 4808  023770  052501  026104  051040
 4809  023776  046505  052117  026105
 4810  024004  040520  044522  054524
 4811  024012  046440  052101  044103
 4812  024020  042105  052040  020117
 4813  024026  047111  042524  043122
 4814  024034  041501  006505  000012
 4815
 4816
 4817
 4818  024042  005015  042101  051104  DUNTST: .ASCIZ  <15><12>/ADDRESSES WITH RESPONSIVE VT61S ARE:/<15><12>
 4819  024050  051505  042523  020123
 4820  024056  044527  044124  051040
 4821  024064  051505  047520  051516
 4822  024072  053111  020105  052126
 4823  024100  030466  020123  051101
```

```
4824  024106  035105  005015     000
4825  024113     116  020117  052126  NOVT:   .ASCIZ  /NO VT61 RESPONDED TO ESCZ SEQ. AUTO RETRY IN 30 SEC./<15><12>
4826  024120  030466  051040  051505
4827  024126  047520  042116  042105
4828  024134  052040  020117  051505
4829  024142  055103  051440  050505
4830  024150  020056  052501  047524
4831  024156  051040  052105  054522
4832  024164  044440  020116  030063
4833  024172  051440  041505  006456
4834  024200  000012
4835
4836
4837  024202  005015  045104  030461  DLERR:  .ASCIZ  <15><12>/DJ11 FAILED AT ADDRESS/
4838  024210  043040  044501  042514
4839  024216  020104  052101  040440
4840  024224  042104  042522  051523
4841  024232     000
4842
4843  024233     115  047101  040525  DMANA:  .ASCII  /MANUAL TEST SELECTED -/<15><12>
4844  024240  020114  042524  052123
4845  024246  051440  046105  041505
4846  024254  042524  020104  006455
4847  024262     012
4848  024263     105  052116  051105          .ASCIZ  /ENTER ADDRESSES OF VT615 TO BE TESTED/<15><12>
4849  024270  040440  042104  042522
4850  024276  051523  051505  047440
4851  024304  020106  052126  030466
4852  024312  020123  047524  041040
4853  024320  020105  042524  052123
4854  024326  042105  005015     000
4855
4856  024333     105  052116  051105  DMANB:  .ASCIZ  /ENTER TESTS TO BE RUN/<15><12>
4857  024340  052040  051505  051524
4858  024346  052040  020117  042502
4859  024354  051040  047125  005015
4860  024362     000
4861  024363     105  052116  051105  DMANL:  .ASCIZ  /ENTER LINES TO BE TESTED IN BINARY FORMAT(I.E.0=1,10=2000)/<15><12>
4862  024370  046040  047111  051505
4863  024376  052040  020117  042502
4864  024404  052040  051505  042524
4865  024412  020104  047111  041040
4866  024420  047111  051101  020131
4867  024426  047506  046522  052101
4868  024434  044450  042456  030056
4869  024442  030475  030454  036460
4870  024450  030062  030060  006451
4871  024456  000012
4872
4873  024460  047101  042440  041523  EM1:    .ASCIZ  /AN ESC SEQ. TO THE VT61  FAILED - OCTAL EQUIV. IS:/<15><12>
4874  024466  051440  050505  020056
4875  024474  047524  052040  042510
4876  024502  053040  033124  020061
4877  024510  043040  044501  042514
4878  024516  020104  020055  041517
4879  024524  040524  020114  050505
```

```
4880  024532  044525  027126  044440
4881  024540  035123  005015     000
4882  024545     124  051505  021524  DH1:    .ASCIZ  /TEST#  ERR PC  BYTE 1+2 BYTE 3+4/<15><12>
4883  024552  020040  051105  020122
4884  024560  041520  020040  054502
4885  024566  042524  030440  031053
4886  024574  041040  052131  020105
4887  024602  025463  006464  000012
4888
4889  024610  042522  042503  053111  EM2:    .ASCIZ  /RECEIVE STATUS ERROR./<15><12>
4890  024616  020105  052123  052101
4891  024624  051525  042440  051122
4892  024632  051117  006456  000012
4893  024640  042101  027104  020040  DH2:    .ASCIZ  /ADD.  STAT.  ERR.BITS  CHAR./<15><12>
4894  024646  052123  052101  020056
4895  024654  042440  051122  041056
4896  024662  052111  020123  041440
4897  024670  040510  027122  005015
4898  024676     000
4899
4900  024677     123  043117  053524  EM3:    .ASCIZ  /SOFTWARE (VSTAT) STATUS ERROR./<15><12>
4901  024704  051101  020105  053050
4902  024712  052123  052101  020051
4903  024720  052123  052101  051525
4904  024726  042440  051122  051117
4905  024734  006456  000012
4906  024740  050040  051501  021523  DH3:    .ASCIZ  / PASS#,  TEST#,  EXP.STAT,  ACT.STAT/<15><12>
4907  024746  020054  052040  051505
4908  024754  021524  020054  042440
4909  024762  050130  051456  040524
4910  024770  026124  020040  041501
4911  024776  027124  052123  052101
4912  025004  005015     000
4913
4914  025007     107  027104  042040  EM4:    .ASCIZ  /GD. DATA DOES NOT MATCH REC. DATA/<15><12>
4915  025014  052101  020101  047504
4916  025022  051505  047040  052117
4917  025030  046440  052101  044103
4918  025036  051040  041505  020056
4919  025044  040504  040524  005015
4920  025052     000
4921  025053     124  051505  021524  DH4:    .ASCIZ  /TEST# ,REC.CNT.,GD. DATA, REC. DATA/<15><12>
4922  025060  026040  042522  027103
4923  025066  047103  027124  043454
4924  025074  027104  042040  052101
4925  025102  026101  051040  041505
4926  025110  020056  040504  040524
4927  025116  005015     000
4928          025122                          .EVEN
4929
4930  025122  054502  042524  020123  EM5:    .ASCIZ  /BYTES EXPECTED DOES NOT EQUAL BYTES RECEIVED/<15><12>
4931  025130  054105  042520  052103
4932  025136  042105  042040  042517
4933  025144  020123  047516  020124
4934  025152  050505  040525  020114
4935  025160  054502  042524  020123
```

```
4936  025166  042522  042503  053111
4937  025174  042105  005015     000
4938  025201     102  052131  051505  DH5:    .ASCIZ  /BYTES EXP., BYTES REC./<15><12>
4939  025206  042440  050130  026056
4940  025214  041040  052131  051505
4941  025222  051040  041505  006456
4942  025230  000012
4943
4944  025232  052503  051522  051117  EM6:    .ASCIZ  /CURSOR POSITIONING ERROR/<15><12>
4945  025240  050040  051517  052111
4946  025246  047511  044516  043516
4947  025254  042440  051122  051117
4948  025262  005015     000
4949  025265     107  020104  044514  DH6:    .ASCIZ  /GD LINE  GD COL.   BD LINE  BD COL/<15><12>
4950  025272  042516  020040  042107
4951  025300  041440  046117  020056
4952  025306  020040  042102  046040
4953  025314  047111  020105  041040
4954  025322  020104  047503  006514
4955  025330  000012
4956
4957  025332  044504  042522  052103  EM7:    .ASCIZ  /DIRECT CURSOR ADDRESSING FAILURE/<15><12>
4958  025340  041440  051125  047523
4959  025346  020122  042101  051104
4960  025354  051505  044523  043516
4961  025362  043040  044501  052514
4962  025370  042522  005015     000
4963  025375     120  051501  021523  DH7:    .ASCIZ  /PASS#  TEST #  ERROR PC   /<15><12>
4964  025402  020040  042524  052123
4965  025410  021440  020040  051105
4966  025416  047522  020122  041520
4967  025424  020040  006440  000012
4968  025432  040520  051523  020043  DH10:   .ASCIZ  /PASS#  TEST#  BD.ROW  BD.COL/<15><12>
4969  025440  052040  051505  021524
4970  025446  020040  042102  051056
4971  025454  053517  020040  042102
4972  025462  041456  046117  005015
4973  025470     000
4974
4975  025471     114  051501  020124  EM11:   .ASCIZ  /LAST TRANSMISSION TO VT61 CAUSED UNIT TO FAIL-HANG./<15><12>
4976  025476  051124  047101  046523
4977  025504  051511  044523  047117
4978  025512  052040  020117  052126
4979  025520  030466  041440  052501
4980  025526  042523  020104  047125
4981  025534  052111  052040  020117
4982  025542  040506  046111  044055
4983  025550  047101  027107  005015
4984  025556     000
4985
4986  025557     126  033124  020061  EM12:   .ASCIZ  /VT61 UNDER TEST FAILED- ERROR DATA FOLLOWS/<15><12>
4987  025564  047125  042504  020122
4988  025572  042524  052123  043040
4989  025600  044501  042514  026504
4990  025606  042440  051122  051117
4991  025614  042040  052101  020101
```

```
MAINDEC-11-DZVTJ-A      MACY11 27(732)  25-SEP-76  09:05  PAGE 96
DZVTJ.P11       TRAP TABLE

 4992  025622  047506  046114  053517
 4993  025630  006523  000012
 4994
 4995  025634  052126  030466  043040  EM10:   .ASCIZ  /VT61 FAILED SELF TEST FUNCTION/<15><12>
 4996  025642  044501  042514  020104
 4997  025650  042523  043114  052040
 4998  025656  051505  020124  052506
 4999  025664  041516  044524  047117
 5000  025672  005015     000
 5001
 5002
 5003  025675     120  051501  021523  DH12:   .ASCIZ  /PASS#,  TEST#, GD.CKSUM, BD.CKSUM/<15><12>
 5004  025702  020054  052040  051505
 5005  025710  021524  020054  042107
 5006  025716  041456  051513  046525
 5007  025724  020054  042102  041456
 5008  025732  051513  046525  005015
 5009  025740     000
 5010
 5011  025741     124  051505  044524  DABRT:  .ASCIZ  /TESTING ABORTED-TOO MANY FATAL XMITS/<15><12>
 5012  025746  043516  040440  047502
 5013  025754  052122  042105  052055
 5014  025762  047517  046440  047101
 5015  025770  020131  040506  040524
 5016  025776  020114  046530  052111
 5017  026004  006523  000012
 5018
 5019  026010  052126  030466  051040  EM13:   .ASCIZ  /VT61 RECEIVER CHECKSUM COMPARE ERROR/<15><12>
 5020  026016  041505  044505  042526
 5021  026024  020122  044103  041505
 5022  026032  051513  046525  041440
 5023  026040  046517  040520  042522
 5024  026046  042440  051122  051117
 5025  026054  005015     000
 5026
 5027  026057     126  023124  020061  EM14:   .ASCIZ  /VT61 TRANSMITTER CHECKSUM COMPARE ERROR/<15><12>
 5028  026064  051124  047101  046523
 5029  026072  052111  042524  020122
 5030  026100  044103  041505  051513
 5031  026106  046525  041440  046517
 5032  026114  040520  042522  042440
 5033  026122  051122  051117  005015
 5034  026130     000
 5035
 5036          026132                  .EVEN
 5037  026132  047125  052111  052440  DVUNIT: .ASCII  /UNIT UNDER TEST /<15><12>
 5038  026140  042116  051105  052040
 5039  026146  051505  020124  005015
 5040  026154  041522  051123  020040          .ASCIZ  /RCSR   VECT.   LINE  IDENT/<15><12>
 5041  026162  053040  041505  027124
 5042  026170  020040  046040  047111
 5043  026176  020105  044440  042504
 5044  026204  052116  005015     000
 5045  026211     040  041522  051123  DH11:   .ASCIZ  / RCSR   VECT./<15><12>
 5046  026216  020040  053040  041505
 5047  026224  027124  005015     000
```

```
5048  026231     120  044522  052116  DPRTR:  .ASCIZ  /PRINTER IS ATTACHED/<15><12>
5049  026236  051105  044440  020123
5050  026244  052101  040524  044103
5051  026252  042105  005015     000
5052  026257     103  050117  042511  DCOPYR: .ASCIZ  /COPIER IS ATTACHED/<15><12>
5053  026264  020122  051511  040440
5054  026272  052124  041501  042510
5055  026300  006504  000012
5056  026304  047530  043106  052040  EM15:   .ASCIZ  /XOFF TO VT61 FAILED TO HALT BLOCK XMIT/<15><12>
5057  026312  020117  052126  030466
5058  026320  043040  044501  042514
5059  026326  020104  047524  044040
5060  026334  046101  020124  046102
5061  026342  041517  020113  046530
5062  026350  052111  005015     000
5063  026355     130  047117  052040  EM16:   .ASCIZ  /XON TO VT61 FAILED TO RESTART BLOCK XMIT/<15><12>
5064  026362  020117  052126  030466
5065  026370  043040  044501  042514
5066  026376  020104  047524  051040
5067  026404  051505  040524  052122
5068  026412  041040  047514  045503
5069  026420  054040  044515  006524
5070  026426  000012
5071  026430  047516  054040  047117  EM17:   .ASCIZ  /NO XON RECEIVED WITHIN 3 SEC. AFTER A RESET/<15><12>
5072  026436  051040  041505  044505
5073  026444  042526  020104  044527
5074  026452  044124  047111  031440
5075  026460  051440  041505  020056
5076  026466  043101  042524  020122
5077  026474  020101  042522  042523
5078  026502  006524  000012
5079  026506  040514  052123  050040  EM20:   .ASCIZ  /LAST PERIPHERAL OPERATION ABORTED/<15><12>
5080  026514  051105  050111  042510
5081  026522  040522  020114  050117
5082  026530  051105  052101  047511
5083  026536  020116  041101  051117
5084  026544  042524  006504  000012
5085  026552  047503  046125  020104  EM21:   .ASCIZ  /COULD NOT CLEAR LAST ABORT FLAG./<15><12>
5086  026560  047516  020124  046103
5087  026566  040505  020122  040514
5088  026574  052123  040440  047502
5089  026602  052122  043040  040514
5090  026610  027107  005015     000
5091  026615     123  046517  047440  EM22:   .ASCIZ  /SOM OR EOM NOT RECEIVED DURING MAINT. MODE TRANSMIT/<15><12>
5092  026622  020122  047505  020115
5093  026630  047516  020124  042522
5094  026636  042503  053111  042105
5095  026644  042040  051125  047111
5096  026652  020107  040515  047111
5097  026660  027124  046440  042117
5098  026666  020105  051124  047101
5099  026674  046523  052111  005015
5100  026702     000
5101  026703     114  047111  020105  EM23:   .ASCIZ  /LINE FEED OR CURSOR RIGHT ISSUED FROM ROW 23 DID NOT CAUSE SCREEN TO SC
5102  026710  042506  042105  047440
5103  026716  020122  052503  051522
```

```
5104  026724  051117  051040  043511
5105  026732  052110  044440  051523
5106  026740  042525  020104  051106
5107  026746  046517  051040  053517
5108  026754  031040  020063  044504
5109  026762  020104  047516  020124
5110  026770  040503  051525  020105
5111  026776  041523  042522  047105
5112  027004  052040  020117  041523
5113  027012  047522  046114  005015
5114  027020     000
5115  027021     120  051501  020123  DH13:   .ASCIZ  /PASS ,   TEST ,   VSTAT/<15><12>
5116  027026  020054  020040  042524
5117  027034  052123  026040  020040
5118  027042  053040  052123  052101
5119  027050  005015     000
5120  027053     120  051501  026123  DH14:   .ASCIZ  /PASS,   TEST,   ERR PC,   VSTAT/<15><12>
5121  027060  020040  052040  051505
5122  027066  026124  020040  042440
5123  027074  051122  050040  026103
5124  027102  020040  053040  052123
5125  027110  052101  005015     000
5126
5127  027115     105  041523  000040  DESC:   .ASCIZ  /ESC /
5128
5129
5130
5131  027122  042513  041131  040517  DKYBD:  .ASCII  /KEYBOARD TEST/<15><12>
5132  027130  042122  052040  051505
5133  027136  006524     012
5134  027141     113  054505  052123          .ASCII  /KEYSTROKES ECHO:/<15><12>
5135  027146  047522  042513  020123
5136  027154  041505  047510  006472
5137  027162     012
5138  027163     101  042040  051511          .ASCII  /A DISPLAY CHAR. = A DISPLAY CHAR./<15><12>
5139  027170  046120  054501  041440
5140  027176  040510  027122  036440
5141  027204  040440  042040  051511
5142  027212  046120  054501  041440
5143  027220  040510  027122  005015
5144  027226  031463  036440  042440          .ASCII  /33 = ESC/<15><12>
5145  027234  041523  005015
5146  027240  032461  036440  041440          .ASCII  /15 = C-R/<15><12>
5147  027246  051055  005015
5148  027252  031061  036440  046040          .ASCII  /12 = L-F/<15><12>
5149  027260  043055  005015
5150  027264  033460  036440  041040          .ASCII  /07 = BELL/<15><12>
5151  027272  046105  006514     012
5152  027277     061  020060  020075          .ASCII  /10 = TAB/<15><12>
5153  027304  040524  006502     012
5154  027311     116  047117  042055          .ASCIZ  /NON-DISPLAY CHAR.= OCTAL EQUIV/<15><12>
5155  027316  051511  046120  054501
5156  027324  041440  040510  027122
5157  027332  020075  041517  040524
5158  027340  020114  050505  044525
5159  027346  006526  000012
```

```
                                                
                                                
027352  040524  020102     000  DTAB:   .ASCIZ  /TAB /
027357     103  051055  000040  DCR:    .ASCIZ  /C-R /
027364  026514  020106     000  DLF:    .ASCIZ  /L-F /
027371     102  046105  020114  DBELL:  .ASCIZ  /BELL /
027376     000
                                                
027377     114  047517  020120  DLOOP:  .ASCII  /LOOP TEST - LOOP COMMANDS AND DATA THRU/<15><12>
027404  042524  052123  026440
027412  046040  047517  020120
027420  047503  046515  047101
027426  051504  040440  042116
027434  042040  052101  020101
027442  044124  052522  005015
027450  047510  052123  041040          .ASCII  /HOST BACK TO VT61 UNDER TEST. /<15><12>
027456  041501  020113  047524
027464  053040  033124  020061
027472  047125  042504  020122
027500  042524  052123  020056
027506  005015
027510  047503  052116  047522  DCNTZ:  .ASCIZ  /CONTROL C  EXITS TEST./<15><12>
027516  020114  020103  042440
027524  044530  051524  052040
027532  051505  027124  005015
027540     000
                                                
027541     105  044530  020124  DEXT:   .ASCIZ  /EXIT TEST./
027546  042524  052123  000056
                                                
027554  051120  047111  042524  DPRNT:  .ASCII  /PRINTER TEST -/<15><12>
027562  020122  042524  052123
027570  026440  005015
027574  031461  020062  047503          .ASCII  /132 COLUMNS OF A SLIDING PATTERN WILL BE/
027602  052514  047115  020123
027610  043117  040440  051440
027616  044514  044504  043516
027624  050040  052101  042524
027632  047122  053440  046111
027640  020114  042502
027644  047503  052116  047111          .ASCII  /CONTINOUSLY OUTPUTTED TO PRINTER/<15><12>
027652  052517  046123  020131
027660  052517  050124  052125
027666  042524  020104  047524
027674  050040  044522  052116
027702  051105  005015
027706  040503  027122  051040  DCRST:  .ASCIZ  /CAR. RET. TO START/<15><12>
027714  052105  020056  047524
027722  051440  040524  052122
027730  005015     000
                                                
027733     114  051501  020124  DEVERR: .ASCIZ  /LAST XMIT CAUSED VT61 HANG/<15><12>
027740  046530  052111  041440
027746  052501  042523  020104
027754  052126  030466  044040
027762  047101  006507  000012
027770  000077                  QMRK:   .ASCIZ  /?/
```

```
5216  027772  051120  042117  041525  DKBD:   .ASCII  /PRODUCTION KEYBOARD TEST. 10 ERRORS CAUSES TEST EXIT./<15><12>
5217  030000  044524  047117  045440
5218  030006  054505  047502  051101
5219  030014  020104  042524  052123
5220  030022  020056  030061  042440
5221  030030  051122  051117  020123
5222  030036  040503  051525  051505
5223  030044  052040  051505  020124
5224  030052  054105  052111  006456
5225  030060     012
5226  030061     104  050105  042522          .ASCIZ  /DEPRESS KEYS FROM LEFT TO RIGHT/<15><12>
5227  030066  051523  045440  054505
5228  030074  020123  051106  046517
5229  030102  046040  043105  020124
5230  030110  047524  051040  043511
5231  030116  052110  005015     000
5232  030123     104  050105  042522  DLSHFT: .ASCIZ  /DEPRESS LEFT SHIFT KEY AND THE "A" KEY /<15><12>
5233  030130  051523  046040  043105
5234  030136  020124  044123  043111
5235  030144  020124  042513  020131
5236  030152  047101  020104  044124
5237  030160  020105  040442  020042
5238  030166  042513  020131  005015
5239  030174     000
5240  030175     104  050105  042522  DTOP:   .ASCIZ  /DEPRESS KEYS IN TOP ROW/<15><12>
5241  030202  051523  045440  054505
5242  030210  020123  047111  052040
5243  030216  050117  051040  053517
5244  030224  005015     000
5245
5246  030227     104  050105  042522  DRSHFT: .ASCIZ  /DEPRESS RIGHT SHIFT KEY AND THE "A" KEY /<15><12>
5247  030234  051523  051040  043511
5248  030242  052110  051440  044510
5249  030250  052106  045440  054505
5250  030256  040440  042116  052040
5251  030264  042510  021040  021101
5252  030272  045440  054505  006440
5253  030300  000012
5254  030302  042504  051120  051505  DSEC:   .ASCIZ  /DEPRESS KEYS IN SECOND ROW/<15><12>
5255  030310  020123  042513  051531
5256  030316  044440  020116  042523
5257  030324  047503  042116  051040
5258  030332  053517  005015     000
5259
5260  030337     104  050105  042522  DTHRD:  .ASCIZ  /DEPRESS KEYS IN THIRD ROW BEGINNING WITH 'A'/<15><12>
5261  030344  051523  045440  054505
5262  030352  020123  047111  052040
5263  030360  044510  042122  051040
5264  030366  053517  041040  043505
5265  030374  047111  044516  043516
5266  030402  053440  052111  020110
5267  030410  040447  006447  000012
5268  030416  042504  051120  051505  DCONT:  .ASCIZ  /DEPRESS CONTROL KEY ,AND THE "A" KEY /<15><12>
5269  030424  020123  047503  052116
5270  030432  047522  020114  042513
5271  030440  020131  040454  042116
```

```
5272  030446  052040  042510  021040
5273  030454  021101  045440  054505
5274  030462  006440  000012
5275  030466  042504  051120  051505  DBOT:   .ASCIZ  /DEPRESS KEYS IN FORTH ROW EXCEPT SHIFT KEYS/<15><12>
5276  030474  020123  042513  051531
5277  030502  044440  020116  047506
5278  030510  052122  020110  047522
5279  030516  020127  054105  042503
5280  030524  052120  051440  044510
5281  030532  052106  045440  054505
5282  030540  006523  000012
5283  030544  042504  051120  051505   DSPCE: .ASCIZ  /DEPRESS SPACE BAR/<15><12>
5284  030552  020123  050123  041501
5285  030560  020105  040502  006522
5286  030566  000012
5287
5288  030570  042504  051120  051505  DKPD:   .ASCIZ  /DEPRESS KEYPAD KEYS,LEFT TO RIGHT, TOP TO BOTTOM/<15><12>
5289  030576  020123  042513  050131
5290  030604  042101  045440  054505
5291  030612  026123  042514  052106
5292  030620  052040  020117  044522
5293  030626  044107  026124  052040
5294  030634  050117  052040  020117
5295  030642  047502  052124  046517
5296  030650  005015     000
5297
5298  030653     113  054505  047502  DKBERR: .ASCII  /KEYBOARD ERROR,KEY POSITION IN ROW SHOULD BE /
5299  030660  051101  020104  051105
5300  030666  047522  026122  042513
5301  030674  020131  047520  044523
5302  030702  044524  047117  044440
5303  030710  020116  047522  020127
5304  030716  044123  052517  042114
5305  030724  041040  020105
5306  030730  020040  005015          KYSTRK: .ASCII  /  /<15><12>
5307  030734  041517  040524  020114          .ASCIZ  /OCTAL GD, OCTAL BAD/<15><12>
5308  030742  042107  020054  041517
5309  030750  040524  020114  040502
5310  030756  006504  000012
5311  030762  020040  020040  020040  DSPC6:  .ASCIZ  /      /
5312  030770     000
5313
5314  030771      036      076      020  ROW1:   .BYTE   36,76,20,13,32,12,54,44,14,41,71,57,63.64,3,114,0
5315  030774      013      032      012
5316  030777      054      044      014
5317  031002      041      071      057
5318  031005      063      064      003
5319  031010      114      000
5320
5321  031012      026      056      030  ROW2:   .BYTE   26,56,30,73,52,22,55,34,24,31,51,77,62,61,2,0
5322  031015      073      052      022
5323  031020      055      034      024
5324  031023      031      051      077
5325  031026      062      061      002
5326  031031      000
5327
```

```
5328  031032  046  040  053  ROW3:   .BYTE   46,40,53,23,72,42,45,74,11,21,47,27,66,0
5329  031035  023  072  042
5330  031040  045  074  011
5331  031043  021  047  027
5332  031046  066  000
5333
5334  031050  016  070  060  ROW4:   .BYTE   16,70,60,50,33,43,25,35,75,65,37,115,67,0
5335  031053  050  033  043
5336  031056  025  035  075
5337  031061  065  037  115
5338  031064  067  000
5339
5340  031066  001  000       CNTRA:  .BYTE   01,0
5341
5342  031070  101  000       SHFTA:  .BYTE   101,0
5343
5344  031072  015  000       SPCB:   .BYTE   15,0
5345
5346  031074  113  004  103  KYPD:   .BYTE   113,04,103,104,1,112,101,102,6,7,106,100,5
5347  031077  104  001  112
5348  031102  101  102  006
5349  031105  007  106  100
5350  031110  005
5351  031111  010  105  107       .BYTE   10,105,107,110,17,111,0
5352  031114  110  017  111
5353  031117  000
5354
5355                            .EVEN
5356  031120  000500            RCRLB:  .BLKB   500     ;RECEIVE CIRCULAR BUFFER
5357
5358  031620  000500            TCRLB:  .BLKB   500             ;TRANSMIT CIRCULAR BUFFER
5359  032320  000000            ABUFP:  .WORD   0
5360  032322  000062            ABBUF:  .BLKB   50.
5361  032404  000000                    0
5362          000001                    .END
```

```
ABBUF   032322          1504    3334    3336    5360#
ABSXT   021610          4281#
ABUFP   032320          1504*   1781    1813    1816    1828    1831    2040    2041    2048    2621    2624    2628*   3333*
                        3334    3336*   3337*   3575*   3576    3688    3692    4179*   4181    4184    4185*   4251    4256
                        4278    5359#
AESCO   015666          3406    3438#
AESCP   015660          3404    3435#
ALEXT   012660          2875    2880#
ALWCNT  002236          1306#   1607    1867    2308    3249
AOUT    013452          2930    3006#
ASESC   015502          3340    3403#
ASTRT   003332          1507    1520#   2529    2627    2707    2733    2770
AUTO    002266           793    1325#
AUTOA   002276          1327#   1331    1428    2864
BASC3   004476          1753    1754#
BBLUP   014316          3063    3178#
BCLR  = 000020          1051#
BDAD    013406          2995    2997#
BDEXT   012650          2871    2878#
BDEXTA  012646          2872    2877#
BEL     002004          1097#   1526    2643
BINOCT  020412          2659    4025#   4194    4208    4214
BIT0  = 000001           757#
BIT00 = 000001           747#    757     1496    2520    2964    3339    4059
BIT01 = 000002           746#    756     1499    1571    3339    3409
BIT02 = 000004           745#    755
BIT03 = 000010           744#    754     3339    3405    3415
BIT04 = 000020           743#    753
BIT05 = 000040           742#    752
BIT06 = 000100           741#    751
BIT07 = 000200           740#    750     1406    3296
BIT08 = 000400           739#    749     4359
BIT09 = 001000           738#    748     4367    4429
BIT1  = 000002           756#
BIT10 = 002000           737#    4411
BIT11 = 004000           736#    4374
BIT12 = 010000           735#    2583
BIT13 = 020000           734#    4418
BIT14 = 040000           733#    4345
BIT15 = 100000           732#
BIT2  = 000004           755#
BIT3  = 000010           754#
BIT4  = 000020           753#
BIT5  = 000040           752#
BIT6  = 000100           751#
BIT7  = 000200           750#
BIT8  = 000400           749#
BIT9  = 001000           748#
BLDADA  002350          1349#   1371
BLDADD  002346          1347#   1354    2879
BLDINA  020226          2698    3969#   3976
BLDINC  020222          1722    1763    1998    2184    2506    3968#   3975    3987
BLDLNA  002470          1374#   1379
BLDLNE  002460          1365    1368    1372#
BLDTST  002534          1388#
BLKM    002262          1316#   1454*   1755*   1992*   2026*   2335*   2507*   3493    3496*   3501*   3528    3670*   3839*
```

```
SDROW   021226          4199    4202#
SPTVEC= 000014           764#
SUBCT   002244          1309#   2502*   2514    2518*   2754*   2763    3132*   3148*   3150    3154*   3158*   3180*   3187*
                        3202*   3222    3574*   3580*   4188*   4189
SUMPCT  021530          4250    4261#
SYSTOR  017150          2190*   3672    3720#   3765*
CALCK   020700          2006    3332    3517    4118#
CARRT   002006          1099#   2414    2425    2651    2699    3370    4072    4087    4097
CDEV    012776          1328    1362    1366    2917#
CDEVA   013052          2928#   2934    2942    3005
CDWN    002022          1115#
CESAM   015652          3420    3424    3428    3430    3432#   3436    3441    3446    3452    3454
CHKITT  017112          3701    3705    3707    3709#
CHOM    002016          1109#   1876    2000    2337
CHRD    002216          1298#   1468*   1483*   1484*   1485    1560*   1561    1564*   1611*   2854*   3047*   3049    3051
                        3054*   3058*   3134*   3140*   3142*   3143*   3144    3146    3152    3156    3160    3164    3184*
                        3287*   3288*   3289    3777*   3778*   3780    3781    3782*   3783*   3784    3785    4180*   4184*
                        4186    4213
CKABRT  020754          2526    2708    4148#
CKDAT   005474          1896    1918#
CKEOM   004040          1638    1642#
CKEXT   006354          2013    2038    2047    2054    2061    2067#
CKGP    = 000106        1247#   3425
CKKBD   021102          2762    4179#
CKLIN   004344          1717    1719#
CKMEM   005534          1919    1929#
CKMNT   003742          1625    1627#
CKOFF   021744          3711    3900    4321#
CKSCRA  007212          2237    2239#
CKSCRB  007302          2255    2257#
CKSFT   016512          1775    1786    1937    2021    2060    3609#   3714    3902
CKSRC   006222          2039#   2049
CKSTR   017060          3681    3691    3700#
CKSUM = 002000          1030#   2007    3514    3531
CKSUMA  005716          1988    1990#
CKSUMB  006122          2015    2023#
CKVST   017124          3699    3712#
CLFT    002024          1118#   2144
CLMAIN  012062          2760    2771#
CLRCK   002046          1146#
CLREG   016654          1598    1921    2016    2055    2159    2466    3635    3653#   3779    3893
CLTCK   002050          1149#
CMPOS   017440          2327    3797#
CNTF    021346          4190    4224#
CNTRA   031066          2789    5340#
COMGP = 000020          1036#   3380    3427    3431
CONRD   014510          3231#   4083
CONVLN  020510          1458    2987    3042    3091    4057#
CPABRT= 000171          1249#   3447
CPPOS   017502          2359    3812#
CRBUF   002002          1089#
CRCSR   002000          1088#
CRRUL   020074          3915#
CRT     002020          1112#   2105
CUHME   002210          1273#   1684    1687    2325    2331
CUP     002026          1121#   2117
```

```
CURER   017342          1688    1706    1737    2115    2276    2374    2412    2428    3775#
CURPOS= 000004          1038#   3419    3702
CURS1   004150          1671    1673#
CURS1A  006376          2082    2084#
CURS1B  006610          2138    2140#
CURS2   007434          2296    2298#
CURIXT  006570          2102    2119    2126#
C2CK    010040          2351    2355    2366#
C2XT    010102          2310    2324    2345    2367    2370    2377#
DABRT   025741          3251    5011#
DAPNT   002064          1167#
DATSC   020250          2448    3984#
DBELL   027371          2645    5164#
DBOT    030466          2781    5275#
DCNTZ   027510          2616    5180#
DCONT   030416          2783    5268#
DCOPYR  026257          1498    5052#
DCOUNT  020220          1553*   1645*   1767*   1808*   1814*   1829*   3072*   3578*   3951*   3958#
DCR     027357          2653    5162#
DCRAD   002074          1180#   1245    1691    1758    2314
DCRST   027706          5205#
DDISP = 177570           679#    822
DELAY   020156          1554    1646    1768    1809    1815    1830    3073    3579    3942#
DELIM   003532          1538    1568#
DEMP    002054          1155#   3443
DESC    027115          2632    5127#
DEVERR  027733          5210#
DEXT    027541          2768    4284    5186#
DF0     001442           859     866     881     993#
DF1     001446           910     995#
DF2     001450           888     997#
DF3     001474           895    1003#
DF4     001534           916     923     946     953     960     988    1011#
DF5     001537          1012#
DF6     001543           874     902     930     937     967     974     981    1014#
DH1     024545           857    4882#
DH10    025432           900    4968#
DH11    026211           908    5045#
DH12    025675           928     935    5003#
DH13    027021           944     951    5115#
DH14    027053           965     972    5120#
DH2     024640           864    4893#
DH3     024740           872     979    4906#
DH4     025053           879    4921#
DH5     025201           886    4938#
DH6     025265           893    4949#
DH7     025375           914     921     958     986    4963#
DISPLA  001140           822#   2834*   4388*   4410*
DISPRE  000174           780#   2834
DJAPT   001752          1073#   1418*   1422    1434*
DJLNE   001630          1067#   1373    1420    2926    3028    3065    3067    3069    3079
DJTBL   001560          1065#   1363    1418    2927    3027    3076
DKBD    027772          2752    5216#
DKBERR  030653          4205    5298#
DKPD    030570          2783    5288#
DKYBD   027122          2614    5131#
```

```
DLAY    002254          1313#  1636*  1639*  1831*  1897*  2039*  2043*  2347*  2352*  3135*  3138*  3676*  3684*
                        3689*  3694*  3844*  3851*  3855*  3860*  4301*  4304*  4321*  4324*
DLERR   024202          2980   4837#
DLF     027364          2657   5163#
DLOOP   027377          2728   5167#
DLSHFT  030123          2783   5232#
DMAIN   002036          1135#
DMANA   024233          1344   4843#
DMANB   024333          1388   4856#
DMANL   024363          1372   4861#
DMPOCT  021250          4205#
DOAROW  012004          2757#  2767   2774
DPNT    002152          1228#
DPRNT   027554          2683   5189#
DPRTR   026231          1501   5048#
DRECT   002044          1143#  1734   3563   4007
DRSHFT  030227          2783   5246#
DSEC    030302          2721   5254#
DSMES   020316          2615   2684   2729   2753   2769   4001#  4285
DSPCE   030544          2783   5283#
DSPC6   030762          4211   5311#
DSWR  = 177570           678#   821
DTAB    027352          2649   5161#
DTHRD   030337          2781   5260#
DTOP    030175          2781   5240#
DTTBL   012130          2757   2787#  4192
DT0     001428           858    990#
DT1     001434           887    909    992#
DT2     001452           865    894    999#
DT3     001464           915    922   1001#
DT4     001500           873    901    929    936    966    973    980   1005#
DT5     001512           880   1007#
DT6     001524           945    952    959    987   1009#
DUNTST  024042          3080   4818#
DVUNIT  026132          1472   5037#
EAPNT   002062          1165#
EEMP    002052          1153#  3438
EINST   002066          1170#
EMAIN   002034          1133#  3561
EMTVEC= 000030           767#  2804*  2805*
EM1     024460           856   4873#
EM10    025634           907   4995#
EM11    025471           913   4975#
EM12    025557           920   4986#
EM13    026010           927   5019#
EM14    026057           934   5027#
EM15    026304           943   5056#
EM16    026355           950   5063#
EM17    026430           957   5071#
EM2     024610           863   4889#
EM20    026506           964   5079#
EM21    026552           971   5085#
EM22    026615           978   5091#
EM23    026703           985   5101#
EM3     024677           871   4900#
EM4     025007           878   4914#
```

```
EM5     025122            985    4930#
EM6     025232            892    4944#
EM7     025332            899    4957#
ENDPS   010736           2528    2530#
ENDSEL  010724           2513    2521    2527#
ENDTAB  001760           1077#   2902
ENSRT   007166           2211    2218#
EOM   = 000004           1295#   1634    1870    2005    2048    3156    3219    3364    3500    3749    4011
EOS     002030           1124#   2455
EPL   = 001000           1031#   3374    4246    4254
EPNT    002150           1226#   2690
ERLNE   015452           3327    3394#
ERPL    002070           1172#
ERRVEC= 000004            760#   4350    4351*   4353*   4356*
ERSE    010340           2443    2446#
ERSXT   010466           2462    2464    2469#
ESAMB   015762           2631*   3339    3344*   3403    3405    3409*   3415*   3432*   3459#   3762*   4253*
ESC   = 000400           1032#   2629    3345    4246    4254
ESCN    002166           1261#   1579    1875    1999    2109    2143    2336    2454    2524    2691    3341    4256
ESCO    002110           1192#   1253    1582    1732    1760    1806    1878    2001    2002    2338    2494    3552    3562
                         3564    4006
ESCOI = 002110           1253#   1675    1693    2145    2154    2415
ESCP    002154           1242#   1254    1585    2206
ESCPI = 002154           1254#
ESCYI = 002074           1245#
ESCZ    002162           1255    1257#   1630    1632    3595
ESCZI = 002162           1255#   1877
ESSEQ   002252           1312#   1542*   1544    1601    1602
ESTEX   003712           1536    1610#
ESTST   003350           1524    1526#
EXINT   013774           3066    3070    3076#
EXIT3   004670           1780    1782    1787#
EXMAIN  011474           2618    2666#   2685    2730
EXMNT   004130           1652    1657#
EXTST   021600           2625    2705    4269    4278#
FADD    012742           2901#
FADD1   012760           2906#   2909
FEXIT   012046           2759    2765    2768#
FNDBT   021730           4303    4311#
FRCECT  021546           4260    4262    4265#
FTEX1   003716           1608    1612#
FTLCNT  002234           1305#   1502*   1606*   1607    1867    1899*   2045*   2308    3249    3257*   3556*   3697*   3876*
                         4309*
FTLEXT  017214           3733    3735#
GCMD    003406           1534#   1576    1580    1583    1586    1589    1596
GDAD    013402           2973    2996#
GDCURP  006560           2103    2122#
GDSCRL  007414           2253    2270    2272    2277#
GDSTRK  021340           4187    4221#
GETON   016402           2496    3554    3574#
GNS   = ****** U          779    2844    4780    4781    4782    4783    4784
GOTON   016450           3577    3583#
GTCR    020546           2687    4070#   4073
GTEXT   020660           4096    4098    4101#
GTNUM   020570           1349    1375    1392    4083#   4104
HDFLG   020072           2963#   2978    2994*   3001*   3567*   3842*   3889    3892*   3906#
```

```
IABT     002056          1158#
IDENT    002160          1252#   1464*   1466*   1470*   1490    1496    1493    1561    2520
ILLNE =  000010          1037#   3326
INAG     007022          2189#   2213
INITA    013536          1329    1367    3027#
INRPL    006764          2178    2180#
INRXT    007172          2202    2210    2220#
INTAB    001570          1066#   1346    1389    1508    2900    2925    3269
INTRC    015040          1436    3318#
INTXM    016014          1438    3488#
INTXT    013722          3034    3065#
INXMT    003434          1543#   1592
IOTVEC=  000020           765#   2802*   2803*   2921*   2922*   3010*   3011*
ITSUMA   006344          1993    2063#
JMPADD   014724          3273*   3274*   3278    3280#
KBDLP    021112          4181#   4219    4223
KIENA    016252          3491    3533#
KYBD     011212          2611    2613#
KYBXMT   011460          2633    2646    2650    2654    2658    2661#
KYPD     031074          2789    5346#
KYSTRK   030730          4203*   4204*   5306#
KYSTRT   011246          2620#   2642    2663
LDADD    014076          1433    2935    3035    3111#
LDBUF    020746          2028    2086    2192    2241    2260    2396    2450    4141#   4156
LDOUT    017246          3746    3749#
LDPOS    021234          4197    4201    4203#
LDXMIT   017222          1945    1994    2187    2212    2517    2617    2619    2662    2686    2731    2761    2772    3743#
                         4206    4210    4212    4216    4218
LINXT    004456          1729    1731    1738#
LKKB     002040          1138#
LNECK    013230          2964#   2998
LNEPT    001754          1074#   1420*   1440    1443*   1503*
LNFED    002010          1101#   1355    1380    1400    1880    2409    2655    2700    3372    4095    4099    4258
LNRW     017544          2312*   2328    2331*   2363    3797    3799*   3801    3803*   3804*   3812    3814*   3816    3818*
                         3819*   3822#
LOOP     021350          2732    4235#
LOOPR    021434          4242    4246#
LOOPT    021402          4240#   4245    4257    4264    4270
LOOPTA   021406          4241#   4249
LPSTR    021454          4247    4251#
LPTST    011714          2725    2727#
LSTST    010506          2489    2491#
MAINT =  000004          1053#
MANS     002320           794    1342#
MANSA    002332          1344#   1364    1369    1432    2863
MCLR  =  000010          1052#   1459    2966    3003    3030
MEMA     005150          1855#   1946
MEMB     005204          1861#   1868
MEMC     005356          1891#   1908    1910    1913    1917
MEMD     005424          1894    1902#
MEMXT    005676          1869    1885    1901    1940    1969#
MEMI     005142          1851    1853#
MODCA    002752          1440#   2593    3267
MODCK    002626          1330    1398    1405    1418#   1431
MODCO    002650          1421#   1445
MODE     002232          1304#   1325*   1342*   1404*   1426    1506    2527    2861    3247
```

```
MONIT   014542          3247*   4344
MPATT = 005656          1855    1888    1939    1956    1959*
MSTBL   012102          2758    2781*
MSTRT   000204           794*
NABRT = 000170          1251*   4164
NCKGP = 000107          1248*   3429
NOCALC  020744          4120    4122    4135*
NOER    016630          3619    3634    3642*
NOKIL   016032          3489    3493*   3510    3512
NOROUT  021650          4279    4288*
NORXT   017220          3734    3736*
NOSHFT  020504          4048    4052*
NOSOM   016054          3494    3498*
NOUNIT  013750          3068    3071*   3078
NOVT    024113          3071    4825*
NWLN    010122          2392    2394*
OCTBIN  020466          1370    1382    1407    4046*
OCTLNE  001776          1087*   1483    2988*   2989    3092*   3093    3144    3324    4057*   4062*
OFFLP   004774          1813*   1819
ONE     002240          1307*
ONLP    005052          1826*   1834
ONOFA   005034          1817    1823*
ONOFLP  004744          1808*   1832
ONOFXT  005122          1827    1837*
ONOF61  004712          1800    1802*
PABRT = 010000          1028*   3451    4148
PATGN   005624          1864    1903    1948*
PC    =%000007           691*   2548*   2551*   2570*   4420*   4475*   4482*   4489*   4503*   4505*   4564*
PDKBD   011766          2749    2752*
PIRQ  = 177772           677*
PIRQVE= 000240           771*
PMULT   020216          2849*   2855*   2859*   3945    3957*
POPIT   003522          1563*   1609
POP2SP= 022626          1018*   2852    2858    2877    2908    2949    3256
PRABRT= 000172          1250*   3449
PRESC   002250          1311*   1529*   1533    1547    1579*   1582*   1585*   1599
PR0   = 000000           694*
PR1   = 000040           695*
PR2   = 000100           696*
PR3   = 000140           697*
PR4   = 000200           698*
PR5   = 000240           699*
PR6   = 000300           700*
PR7   = 000340           701*
PS    = 177776           674*    675
PSW   = 177776           675*   3276*
PUSH2S= 024646          1019*
PWRVEC= 000024           766*   2808*   2809*   4718*   4719*   4727*   4742*   4743*
QMRK    027770          2878    5215*
QUST    020662          4090    4092    4102*
RABT    002142          1214*
RBBUF   015754          1451*   1730    1890    1892    2014    2018    2053    2057    2152    2161    2254    2271    2274*
                        2356    2357    2373*   2406    2422    2447*   2457*   2463    2467    3367    3379    3456*   3704
                        3708    3758    3782    3847    4244
RBUFP   015760          1892*   1893    2346    2348    2356*   3367*   3377    3379*   3387*   3388*   3458*   3758*   3847
                        3856    4240    4244*   4248    4251*   4252*   4258*   4259*   4263*
```

```
RCRLB   031120          1449    1451    1683    1700    2113*   4237    4281    5356#
RCUR    002102          1185#   1723
RDCUR   002104          1188#   1676    1694    1724    2416    3417
REBUF   015756          1449*   3377    3457#   4236*   4281*
RECAD   014730          2958    3287#
RECDN =  000200         1049#   3136    3233    3258    4070
RECEX   014304          3141    3159    3163#
RECEXA  014310          3157    3165#
RECITT  017144          2097*   2189*   2257*   3709*   3718#   3763*
RECTM   014116          3130#   3598
RECXT   015474          3391    3395    3399#
REEX    014770          3293#   3297
RENA  =  000100         1050#   1505    1527    1612    2530    2970    2977    3266    3292
REOM  =  020000         1027#   1643    1651    1826    1895    2010    2050    2051    2350    3366    3679    3690    3858
                        3880
RESET   002164          1259#   3551
RESETV  016272          1627    1674    1719    1757    1802    1853    1990    2085    2141    2180    2239    2299    2394
                        2446    2501    3549#   4001
RESPTR  017262          1453    1787    1856    1872    1991    2067    2311    2330    2613    2727    3698    3715    3757#
                        3877    4238    4283
RESVEC=  000010           761#  2817*   2824*
REVID =  000040         1035#   3384    3440    3445
ROW1    030771          2787    5314#
ROW2    031012          2787    5321#
ROW3    031032          2787    5328#
ROW4    031050          2787    5334#
RRDY  =  100000         1057#   1555
RSCN  =  000001         1054#
RSMAIN  012076          2771    2775#
RSOM  =  040000         1026#   1637    1649    3361
RSTER   015436          3346    3350    3355    3358    3362    3369    3375    3390#   3410    3416    3433
RSTT  =  004000         1029#   3393
RTRP    012600          2850    2856    2860#
RXOFF =  100000         1025#   1769    3349    3353    3682    3849    4322
R0    =%000000           682#  1327*   1328*   1329*   1345*   1349*   1352*   1359*   1362*   1366*   1367*   1370*   1375*
                        1382*   1392*   1407*   1433*   1453*   1458*   1463*   1549*   1554*   1558*   1598*   1627*   1629*
                        1631*   1633*   1635*   1646*   1674*   1679*   1688*   1696*   1706*   1719*   1722*   1726*   1737*
                        1757*   1763*   1768*   1775*   1779*   1786*   1787*   1802*   1807*   1809*   1811*   1815*   1824*
                        1830*   1853*   1856*   1859*   1864*   1866*   1871*   1872*   1882*   1903*   1921*   1937*   1945*
                        1952*   1957*   1990*   1991*   1994*   1998*   2006*   2012*   2016*   2021*   2028*   2037*   2055*
                        2060*   2067*   2085*   2086*   2099*   2115*   2141*   2148*   2159*   2180*   2184*   2187*   2192*
                        2199*   2212*   2239*   2241*   2250*   2260*   2267*   2276*   2299*   2302*   2307*   2311*   2323*
                        2327*   2330*   2344*   2359*   2374*   2394*   2396*   2402*   2412*   2418*   2428*   2446*   2448*
                        2450*   2459*   2466*   2495*   2496*   2501*   2506*   2512*   2517*   2525*   2526*   2559*   2567*
                        2570    2613*   2615*   2617*   2619*   2625*   2639*   2641*   2659*   2662*   2684*   2686*   2687*
                        2692*   2698*   2705*   2708*   2727*   2729*   2731*   2732*   2753*   2761*   2762*   2769*   2772*
                        2879*   2881*   2911*   2935*   2987*   3020*   3035*   3042*   3045*   3063*   3073*   3074*   3075*
                        3091*   3102*   3116*   3167*   3192*   3224*   3238*   3254*   3332*   3517*   3553*   3554*   3557
                        3565*   3568*   3579*   3582*   3583*   3596*   3598*   3600*   3635*   3646*   3657*   3696*   3698*
                        3711*   3714*   3715*   3717*   3732*   3736*   3747*   3750*   3751*   3766*   3779*   3788*   3800*
                        3805*   3815*   3820*   3875*   3877*   3882*   3893*   3900*   3902*   3904*   3920*   3955*   3972*
                        3987*   4001*   4015*   4039*   4053*   4064*   4074*   4083*   4101*   4135*   4141    4143*   4150
                        4156*   4162*   4166    4170*   4194*   4206*   4208*   4210*   4212*   4214*   4216*   4218*   4224*
                        4238*   4269*   4271*   4283*   4285*   4287*   4288*   4311    4314*   4326*   4344*   4459    4460*
                        4461    4464*   4520    4521*   4522*   4529*   4530*   4531*   4532*   4533*   4534    4539    4545
                        4547*   4548    4562*   4659    4669*   4673    4689    4690    4703*   4720    4741*   4763    4764*
```

```
                    4765    4766*   4767*   4768*   4769*
ROSVE   002256      1314#   3131*   3166    3179*   3191    3201*   3223    3232*   3237    3610*   3645    3776*   3787
                    4298*   4313
ROSV1   016510      3593*   3599    3601#
ROOC08  002206      1272#
ROOC11  002202      1269#
ROOC20  002204      1271#
ROOC80  002212      1274#
RO1C00  002170      1262#   1730    1736    2422    2427
RO1C20  002172      1263#   2406    2411
R1    =%000001       683#   1350    1355    1376    1380    1393    1400    1421*   1526*   1534    1642*   1647*   1673*
                    1675*   1676*   1683*   1684    1690*   1691*   1692*   1693*   1694*   1700*   1702    1720*   1723*
                    1724*   1754*   1758*   1759*   1760*   1761*   1874*   1875*   1876*   1877*   1878*   1879*   1880*
                    1890*   1893    1907    1915    1920*   1922    1923    1924    1926    1996*   1999*   2000*   2001*
                    2002*   2003*   2027*   2084*   2103*   2107    2114    2116    2140*   2142*   2143*   2144*   2145*
                    2146*   2181*   2182*   2185*   2191*   2240*   2259*   2313*   2314*   2315    2316*   2317*   2332*
                    2333    2336*   2337*   2338*   2339*   2346*   2348    2368*   2369    2373    2375    2395*   2413*
                    2414*   2415*   2416*   2449*   2504*   2624*   2634    2636    2638    2643    2647    2651    2655
                    2693*   2696*   2699*   2700*   2889*   2892*   2893*   2895    2899*   2928*   2936*   2940*   2957*
                    2958*   2959*   2960*   2961*   2965*   2974*   3012*   3014*   3015*   3017    3029*   3111*   3112*
                    3113*   3114    3260*   3261*   3262    3264*   3271*   3272*   3273    3319    3321*   3322    3328*
                    3331    3337    3341    3347    3351    3359    3364    3370    3372    3382*   3386*   3387    3390
                    3396    3401*   3407    3411    3413    3417    3421    3425    3429    3435    3438    3443    3447
                    3449    3453    3611    3614*   3618    3621    3631*   3633    3638    3639    3642*   3672*   3700
                    3706    3708*   3854    3868    3871    3885*   3887    3898*   3915*   3917*   3918    3929*   3943
                    3946*   3947*   3954*   3969*   3985*   4004*   4005*   4006*   4007*   4009*   4011*   4029*   4030*
                    4031*   4032    4046*   4052    4084*   4085    4087    4089    4091    4093    4095    4097    4099*
                    4141*   4155*   4191*   4192*   4193*   4207*   4213*   4544    4545*   4550    4554    4556    4561*
                    4660    4673*   4674    4678    4702*   4721    4740*
R12C00  002176      1265#
R2    =%000002       684#   1347*   1353    1357    1360    1374*   1378    1384    1391*   1395    1402    1422*   1424
                    1434    1533*   1593*   1595*   1860*   1861*   1887*   1902*   1909    1916    1929    1932    2040*
                    2041    2104*   2107    2113    2116    2300*   2303*   2315*   2325    2328*   2333*   2361    2363*
                    2366    2375    2614*   2683*   2728*   2752*   2768*   2870    2873    2890*   2892    2894*   2901*
                    2902    2904    2905    2906*   2925*   2929    2933*   3013*   3014    3016*   3027*   3033    3076*
                    3077    3082    3100    3111    3320    3322*   3323*   3324    3400*   3612    3615*   3617*   3618
                    3632*   3633    3643*   3673*   3700    3704*   3856    3864    3869    3871    3916*   3917    3921*
                    3922    3924*   3926    3928*   3929    3930*   3931*   3944    3945*   3949*   3953*   4009    4047
                    4049*   4050*   4051*   4052*   4284*   4661    4672*   4676*   4679    4686*   4687*   4688    4693*
                    4701*   4722    4739*
R22C00  002174      1264#
R23C00  002200      1267#   1759    2275
R23C78  002214      1275#   2312    2316
R23C79  002076      1182#   1692    1702    1705    2369
R3    =%000003       685#   1346*   1353*   1360*   1361*   1373*   1378*   1384*   1385*   1386*   1389*   1395*   1402*
                    1403*   1423*   1435    1436*   1437*   1438*   1439*   1534*   1537    1542    1568    1571    1574
                    1591    1593    1594*   1595    1721*   1762*   1766*   1771*   1889*   1912    1914*   1915*   1916*
                    1918    1926    1944*   1993*   1997*   2105*   2110    2117    2183*   2186*   2211*   2505*   2516*
                    2616*   2618*   2632*   2645*   2649*   2653*   2657*   2660*   2685*   2697*   2730*   2758*   2771*
                    2891*   2893    2924*   2947*   2957    3028*   3036    3038    3041    3043    3060*   3061    3079*
                    3088    3745    3872*   3970*   3986*   4130*   4131*   4133    4205*   4209*   4211*   4215*   4217*
                    4601    4610*   4616*   4617*   4620*   4625*   4626*   4627    4636*   4662    4670*   4671*   4685*
                    4688*   4697*   4698*   4700*   4723    4738*
R4    =%000004       686#   1390*   1396*   1397    1530*   1540    1545    1551    1578*   1588*   1812*   1818*   1825*
                    1833*   1855*   1865    1888*   1906*   1907    1914    1948*   1949    1951*   1953*   1954    1956*
                    2004*   2014    2017    2024*   2053    2056    2757*   2926*   2996*   2999*   3007*   3182    3183*
```

```
                    3184    3185*   3190*   3202    3205*   3207    3208    3212    3213*   3214    3222*   3837    3838*
                    3863*   3870    3903*   4125    4129*   4130    4132    4134*   4186    4191    4207    4221*   4222
                    4240*   4248    4602    4604*   4605*   4606*   4607    4608*   4622    4624*   4632*   4635*   4724
                    4737*
R5      =%000005     687#    1813*   1816    1828*   1831    1854*   1855    1888    1904    1938*   1939    1942    1956
                    2005*   2188*   2203    2205    2208    2209    2694*   2709    2755*   2757    2758    2766    2773
                    2900*   2905*   2910*   2927*   3000*   3006*   3204*   3207*   3214    3331*   3516*   3669    3688*
                    3692    3716*   3843*   3866    3868*   3869*   3870*   3884*   3895    3886*   3894    3895    3896
                    3968*   3969    3973*   3974    4026    4027*   4032*   4033*   4034*   4035*   4038*   4118*   4119
                    4121    4131    4132*   4133*   4134    4192    4603    4609*   4611*   4613*   4614*   4615*   4616
                    4634*   4663    4665*   4667*   4674*   4678*   4693    4699*   4725    4736*
R6      =%000006     688#    690     2797*   2798*   2799    2946*   2947    2948    3205    3235*   3236*   3255
R7      =%000007     689#    691
SCAN    = 000401    1056#   1462    2969    3002    3044
SEQ4      003620    1541    1591#
SETA      012154    1326    1343    2796#   3253
SETREV    005672    1944    1967#   2516
SHFTA     031070    2789    5342#
SLSH    = 000057    1246#   3146    3421
SOM     = 000002    1294#   1628    1858    2301    3152    3203    3359    3495    3743    4005
SP      =%000006     690#   1446*   1447*   1464    1468    1473*   1478*   1485*   1490*   1532*   1544*   1547*   1560
                    1564    1611    1687*   1705*   1736*   1774*   1777*   1778*   1785*   1804*   1805*   1806*   1936*
                    2010*   2011*   2020*   2035*   2036*   2059*   2114*   2275*   2321*   2322*   2342*   2343*   2372*
                    2411*   2427*   2492*   2493*   2494*   2510*   2511*   2522*   2523*   2524*   2555*   2562*   2563*
                    2575*   2576*   2579    2580*   2581    2582*   2587*   2588*   2689*   2690*   2691*   2801*   2818*
                    2819*   2823*   2828*   2829*   2835    2836    2837    2917*   2918*   2954*   2955*   2981*   2989*
                    3047    3054    3058    3061*   3062*   3082*   3093*   3131    3132    3133*   3160*   3164*   3166*
                    3179    3180    3181    3182*   3190    3191*   3201    3223*   3232    3237*   3277*   3278*   3319*
                    3320*   3400    3401    3550*   3551*   3552*   3560*   3561*   3562*   3563*   3564*   3593    3594*
                    3595*   3597*   3599*   3610    3614    3645*   3669*   3713*   3716    3730*   3731*   3776    3777
                    3787*   3837*   3880*   3881*   3901*   3903    3943*   3944*   3953    3954    4026*   4038    4084
                    4298    4299    4300    4313*   4350*   4353    4355    4356    4384    4385    4389*   4415    4431*
                    4434*   4459*   4460    4461*   4463    4464    4465*   4467    4469    4471    4476    4478*   4480*
                    4488*   4492    4496    4497    4501    4520*   4525*   4544*   4550*   4554*   4561    4562    4593*
                    4594    4595    4596*   4601*   4602*   4603*   4609    4634    4635    4636    4637*   4638*   4659*
                    4660*   4661*   4662*   4663*   4664*   4665    4668*   4681    4683*   4685    4695    4697    4699
                    4700    4701    4702    4703    4705*   4706*   4720*   4721*   4722*   4723*   4724*   4725*   4726
                    4732*   4736    4737    4738    4739    4740    4741    4746*   4763*   4764
SPCB      031072    2787    5344#
SPTN      003612    1539    1588#
STACK   = 001100     671#   2801
START     000200     793#
STKLMT= 177774       676#
STRBYT    017102    3703    3706#
STRO      016010    3453*   3472#   4164
STRP      016012    3435*   3473#
STRTAB    001756    1076#   2901
STTEP     016006    3394    3396*   3397*   3471#   3632    3644*   3760*
STTER     015764    3461#   3631    3644    3760
STUPM     023726    2846    4802#
SVER1     002220    1299#   2660    3611*   3642    4032*   4033*   4034*   4195*   4196    4198    4200*   4202*   4203
                    4204    4209    4215
SVER2     002222    1300#   3612*   3643    4037*   4195
SWR       001136     821#   2583    2831    2833*   4345    4359    4361    4367    4374    4411    4418    4423    4429
SWREG     000176     781#   2833
SWO     = 000001     729#
```

```
SW00  = 000001          719#    729
SW01  = 000002          718#    728
SW02  = 000004          717#    727
SW03  = 000010          716#    726
SW04  = 000020          715#    725
SW05  = 000040          714#    724
SW06  = 000100          713#    723
SW07  = 000200          712#    722
SW08  = 000400          711#    721
SW09  = 001000          710#    720
SW1   = 000002          728#
SW10  = 002000          709#
SW11  = 004000          708#
SW12  = 010000          707#
SW13  = 020000          706#
SW14  = 040000          705#
SW15  = 100000          704#
SW2   = 000004          727#
SW3   = 000010          726#
SW4   = 000020          725#
SW5   = 000040          724#
SW6   = 000100          723#
SW7   = 000200          722#
SW8   = 000400          721#
SW9   = 001000          720#
TAB     002012         1103#   2647
TBBUF   016262         1452*   1673    1690    1720    1754    1874    1996    2027    2084    2140    2181    2186    2191
                       2240    2259    2313    2332    2366    2368    2395    2413    2449    2504    2693    2696    3509
                       3521    3532    3537#   3759    3985    4004    4155    4237*   4282*
TBITVE= 000014          762#   2814*   2815*
TBLCK   012772         2903    2910#
TBUFP   016266         1628*   1630*   1632*   1634*   1810*   1823*   1858*   1865*   1870*   2301*   2306*   2638*   2640*
                       3503    3507    3509*   3511*   3516    3518    3519    3521*   3523*   3532*   3539#   3743*   3745*
                       3749*   3759*
TCOMB = 040004         1055#   2971    3291
TCRLB   031620         1450    1452    1889    1912    1918    1920    2098    2104    2190    2203    2258    2274    2361
                       3843    3866    3884    3886    3915    3918    4236    4282    5358#
TCRLF = 000200         4469    4508#
TCUCH   002120         1199#   2146    2157
TDATA   007162         2203    2216#
TDEV    013132         2921    2943#
TEBUF   016264         1450*   3507    3519    3538#
TESC    014370         1549    1807    2495    2525    2692    3200#   3553    3565    3596
TFUNCT  007154         2205    2215#
THT   = 000011         4467    4507#
TIMEXT  021732         4310    4312#
TKVEC = 000060          769#
TMNAD   012624         1352    1359    2870#
TOADD   002242         1308#   3181*   3183
TOFF    014754         3291#   3299
TOTCH = 003600         1059#   1860    1909    1929    1931    2300    3988
TOTC1 = 003601         1060#
TPENT   012766         2898    2908#
TPREG   002246         1310#   2964*   2972    3289    3295*   3296    3298*   3300    4058*   4059    4061*
TPRNT   011516         2681    2683#
TPVEC = 000064          770#
```

```
TRAPVE= 000034           768#   2806*  2807*
TRDY  = 100000          1046#   3210   3217   3220
TRMID = 000002          1039#   3423
TRPA    012544          2847    2852#
TRPB    012570          2853    2858#
TRPE    012746          2902#   2907
TRPVEC  012670          1327    1345   2889#
TRTVEC= 000014           763#
TSMAD   015022          2960    3300#
TSTADD  023656          3273    4786#
TSTER   002156          1244#   2493
TSTLNE  001774          1086#   1440*  1441   1461   2962*  2968   2996   2997*  3041*  3088*  3089   4058
TSTNM   002264           990    1001   1005   1007   1009   1317#  1455*  1528*  3549*
TSTPTR  002230          1303#   1429   1508*  3264   3269*  3275*
TST1    003332          1522#   4786
TST10   005700          1986#   4790
TST11   006360          2080#   4790
TST12   006572          2136#   4792
TST13   006746          2176#   4792
TST14   007174          2235#   4792
TST15   007416          2294#   4794
TST16   010104          2390#   4794
TST17   010322          2441#   4794
TST2    003724          1623#   4786
TST20   010470          2487#   4796
TST21   011174          2609#   4796
TST22   011500          2679#   4796
TST23   011676          2723#   4798
TST24   011750          2747#   4798
TST3    004132          1669#   4786
TST4    004326          1715#   4788
TST5    004460          1751#   4788
TST6    004674          1798#   4788
TST7    005124          1849#   4790
TXEX    016260          3529    3534#
TXRCK   002126          1204#   2003
TXSUM = 000100          1034#   2025   3329   3368
TXTCK   002134          1209#
TYPDS = 104404          2556    4555   4784#
TYPE  = 104400          1344    1372   1388   1471   1472   1495   1498   1501   2554   2557   2842   2846   2878
                        2980    2986   3004   3071   3080   3087   3099   3251   4094   4103   4413   4421   4472
                        4519    4536   4538   4541   4543   4558   4563   4628   4704   4744   4780#
TYPNUM  020676          4093*   4094   4102*  4103   4105#
TYPOC = 104401          4527    4551   4781#
TYPON = 104403          4763#
TYPOS = 104402          1475    1480   1487   1492   2983   2991   3084   3095   4782#
TYP6    002226          1302#
T1ERR   003634          1562    1598#
UNLKKB  002042          1140#
UPD4    014772          3290    3294#
VBIT    021740          4300*   4302   4315#
VDLAY   021742          4299*   4306*  4316#
VECPT   001750          1072#   1419*  1423   1444*  1478
VECT    001772          1085#   1435*  2499
VJCSR   001762          1081#   1421   1459*  1462*  1473   1505*  1527*  1612*  1765*  2009*  2034*  2320*  2341*
                        2498    2509*  2530*  2702*  2928   2938*  2939*  2943*  2944*  2966*  2969*  2970*  2971*
```

```
                    2976*  2977*  2981   3000   3002*  3003*  3029   3030*  3039*  3044*  3136   3210   3217
                    3220   3266*  3291*  3292*  3294*  3301*  3356*  3533*  3636   3637   3674*  3728*  3757*
                    3841*  3989*  4012*  4070   4268*
VRBUF   001764      1082#  1555   3142   3287   3321   4072
VSTAT   002260      1315#  1457*  1637   1643   1649   1651   1654   1756*  1769   1803*  1820   1826   1835
                    1857*  1886*  1895   1995*  2007*  2023*  2025*  2050*  2051   2298*  2319*  2334*  2350
                    2503*  2620*  2629   2661*  2695*  2703   3326*  3329   3345*  3349*  3353*  3354   3357*
                    3361*  3366*  3368*  3374*  3380   3384   3393*  3419*  3423*  3427*  3431*  3440*  3445*
                    3451*  3488   3490*  3514   3530*  3531*  3566*  3615   3622   3671*  3675*  3677   3679
                    3682   3690   3702   3726*  3735*  3840*  3845   3849   3858   3984*  3991   4002*  4013
                    4148   4152   4168   4239*  4241   4243*  4246   4254*  4302   4322
VVECT   001550      1064#  1419   2924
VXBUF   001770      1084#  3114   3203*  3212*  3219*  3300*  3495*  3500*  3518*
VXTCR   001766      1083#  1461*  2937*  2945*  2968*  3043*
WDSTOR  017146      2098*  2258*  3673   3719#  3764*
WTBGND  021654      1779   2012   2037   2323   2344   2512   3732   3882   4297#
XENA  = 040000      1047#  1765   2009   2034   2320   2341   2509   2702   2939   2943   2976   3294   3301
                    3356   3533   3674   3728   3757   3841   3989   4012   4268
XMAD2   017042      3686   3696#
XMCNT   016270      1677*  1695*  1725*  1764*  1881*  2008*  2033*  2096*  2147*  2198*  2249*  2266*  2318*
                    2340*  2401*  2417*  2458*  2508*  2701*  3498   3525*  3541#  3727*  3761*  3988*  4003*
                    4008*  4161*  4255*  4261   4265*  4266
XMDNE = 000001      1040#  1777   2035   2321   2342   2510   2703   3530   3677   3730   3845   3991   4013
                    4241   4243
XMITT   016732      3675#  3710
XMIT1   017152      1629   1631   1633   1635   1811   1824   1859   1866   1871   2302   2307   2639   2641
                    3726#  3747   3750
XMKIL = 000200      1033#  3354   3357   3490
XMREC   016676      1679   1696   1726   1882   2099   2148   2199   2250   2267   2402   2418   2459   3668#
                    4162
XMTAL   002112      1194#  1761   1805   1879   2339
XMWT    021570      4267   4269#
XMXT    017142      3717#
XMZER   021652      1456*  3505   4235*  4286*  4290#
XOFF  = 000023      1296#  1810   3347   4121
XON   = 000021      1297#  1781   1823   3351   3576   4119
XRCMP   017546      3836#
XREC    017650      3846   3848   3855#  3874
XRERR   017734      3853   3861   3875#
XREXT   017752      3859   3873   3879#
XROUT   020052      3878   3883   3888   3900#
XSCN  = 000400      1048#  2938   2944
ZERO    002224      1301#  1361   1532   1804   2271   2463   2492   2522   2689   3133   3550   3560   3594
ZERST   003376      1531#  1566
ZFLAG   016452      1463   1558   3045   3592#
$BDADR  001122       815#   999   3639*  3654*  3785*
$BDDAT  001126       817#   990    992    999   1005   1007   1601*  1604*  1654*  1734*  1923*  1932*  2018*
                    2057*  2111*  2156*  2157*  2161*  2205*  2409*  2425*  2456*  2467*  2499*  2799   3622*
                    3638*  3656*  3784*  3895*  4152*  4168*
$BELL   001162       833#  4413   4437
$CHARC  022622      4483*  4490   4499*  4504#
$CMTAG  001100       803#  2797   2804   2810   2811   2812
$CM3  = 000000       831#
$CRLF   001167       835#  1471   1495   2986   3004   3087   3099   4217   4421   4437   4473   4510   4519
                    4538   4543   4563
$DBLK   023444      4670   4704   4712#
```

```
$DOAGN  011062          2550    2560    2568    2574#
$DTBL   023434          4673    4708#
$ENDAD  011052           788    2563    2570#   2840    4426
$ENDCT  011000          2552#   2810
$ENDMG  011154          2554    2595#
$ENULL  011171          2557    2598#
$EOP    010744          2542#
$EOPCT  010772          2549#   2553    2810*
$ERFLG  001103           806#   4334    4363    4365    4371*   4392    4408*   4437
$ERMAX  001115           812#   2813*   4365    4387*   4392
$ERROR  022244          2804    4407#
$ERRPC  001116           813#    990    1001    4415*   4416*   4417    4437    4525
$ERRTB  001172           853#   4533
$ERRTY  022626          4420    4518#
$ERTTL  001112           810#   4414*   4437
$ESCAP  001160           832#   2812*   4386*   4432    4434    4437
$FILLC  001154           829#   4476    4510
$FILLS  001153           828#   4510
$GDADR  001120           814#    999    1007    1009    1820*   1835*   1924*   3557*   3636*   3653*   3781*   3896*
$GDDAT  001124           816#    990     992     999    1005    1007    1599*   1600*   1653*   1732*   1733*   1922*   1931*
                        2017*   2056*   2109*   2110*   2154*   2155*   2160*   2206*   2408*   2424*   2454*   2455*   2498*
                        3621*   3637*   3655*   3780*   3894*   4150*   4151*   4166*   4167*
$GET42  011022          2558#
$HD   = 000000           654
$ICNT   001104           807#   4378*   4379    4381*   4391
$ILLUP  023604          4718    4748#
$ITEMB  001114           811#   4417*   4437    4522
$LF     001170           836#   4437    4510
$LOOP   011146          2588    2592#
$LPADR  001106           808#   1524*   1625*   1671*   1717*   1753*   1800*   1851*   1988*   2082*   2138*   2178*   2237*
                        2296*   2392*   2443*   2489*   2611*   2681*   2725*   2749*   2826*   4369*   4384*   4389    4391
$LPERR  001110           809#   2827*   4369    4385*   4391    4431
$MAIL = ****** U        1525    1626    1672    1718    1754    1801    1852    1989    2083    2139    2179    2238    2297
                        2393    2444    2490    2612    2682    2726    2750    2838    4384    4423    4461
$MXCNT  022242          4382    4391#
$NULL   001152           827#   4478    4510
$NWTST= 000000          1521#   1622#   1668#   1714#   1750#   1797#   1848#   1985#   2079#   2135#   2175#   2234#   2293#
                        2389#   2440#   2486#   2608#   2678#   2722#   2746#
$OCNT   023224          4600*   4629*   4642#
$OMODE  023226          4595*   4599*   4604    4607*   4618*   4644#
$OVER   022226          4346    4362    4370    4380    4388#
$PASS   001100           804#   1001    1005    1009    2546*   2547*   2555    2594    4376    4392
$POWER  023612          4745    4751#
$PWRDN  023454          2808    4718#   4742
$PWRMG  023572          4745#
$PWRUP  023522          4727    4732#
$QUES   001166           834#   4437    4510
$RDCHR= ****** U        4785
$RDDEC= ****** U        4785
$RDLIN= ****** U        4785
$RDOCT= ****** U        4785
$RTRN   011144          2564    2589#   2814    2816*   2821*
$R2A  = ****** U        4785
$SAVRE= ****** U        4785
$SAVR6  023610          4726*   4732    4733*   4734*   4750#
$SCOPE  021770          2802    3010    4343#
```

```
$SETUP= 000037          1320#   2544    2802    2804    2806    2808    2810    2811    2812    2814    2826    2840    4426
$STUP = 177777          1320#
$SVLAD  022200          4354    4383#
$SVPC = 000200           786#    791
$SWR  = 177400           644#    654     659     660     661     662     663     664     665     666     831     832     833
                        1523    1624    1670    1716    1752    1799    1850    1987    2081    2137    2177    2236    2295
                        2391    2442    2488    2538    2545    2559    2575    2594    2610    2680    2724    2748    2811
                        2812    2814    2826    2827    4335    4336    4337    4338    4339    4345    4357    4359    4360
                        4363    4364    4365    4372    4373    4374    4385    4388    4391    4399    4400    4401    4402
                        4403    4411    4418    4423    4429    4437    4746
$SWRMK= 000000           666     667    4339    4340    4361
$TBIT   011152          2585*   2594#   2825#
$TIMES  001156           831#   1523*   1624*   1670*   1716*   1752*   1799*   1850*   1987*   2081*   2137*   2177*   2236*
                        2295*   2391*   2442*   2488*   2545*   2610*   2680*   2724*   2748*   2811*   4372*   4379    4382*
                        4391
$TKB    001144           824#   3235    3260
$TKS    001142           823#   3233    3258
$TN   = 000025           644#    654    1521    1523#   1622    1624#   1668    1670#   1714    1716#   1750    1752#   1797
                        1799#   1848    1850#   1985    1987#   2079    2081#   2135    2137#   2175    2177#   2234    2236#
                        2293    2295#   2389    2391#   2440    2442#   2486    2488#   2608    2610#   2678    2680#   2722
                        2724#   2746    2748#
$TPB    001150           826#   4496*   4510
$TPFLG  001155           830#   4455    4510
$TPS    001146           825#   4494    4510
$TRAP   023622          2806    4763#
$TRP  = 000005          4771#   4781#   4782#   4783#   4784#   4785#
$TRPAD  023674          4768    4779#
$TSTNM  001102           805#   1528    2544*   3549    4334    4361    4383*   4388    4392    4410    4437
$TYPBN= ****** U        4785
$TYPDS  023230          4658#   4784
$TYPE   022412          4455#   4771    4780
$TYPEC  022556          4475    4482    4489    4494#   4495
$TYPEX  022624          4500    4502    4505#
$TYPOC  023026          4598#   4781
$TYPON  023042          4597    4600#   4783
$TYPOS  023002          4593#   4782
$XTSTR  022004          4348#
$DFILL  023225          4594*   4598*   4608    4643#
$40CAT= ****** U        4345    4420
.     = 032406           775#    779#    786     787#    789#    791#    792#    802#    837    1016#   1064#   1065#   1066#
                        1067#   1556    1565    1959    2052    2594    2599    2704    2800    2826    2827    2941    3211
                        3218    3221    3234    3948    3992    4071    4142    4182    4391    4392    4437    4510    4566#
                        4712#   4729    4749    4928#   5036#   5356#   5358#   5360#
.BEL  = 000007          1096#   2093    2246    2399    4159
.CARRT= 000015          1098#
.CDWN = 000102          1114#   2087    2196    2242
.CHOM = 000110          1108#   2193    2246    2264    2451
.CLFT = 000104          1117#   2090    2242
.CLRCK= 000133          1145#   2063
.CLTCK= 000134          1148#   2029
.CPYSC= 000135          1230#   2523
.CRT  = 000103          1111#   2087
.CUP  = 000101          1120#   2093
.C00  = 000040          1281#   2242
.C03  = 000043          1282#
.C08  = 000050          1283#
```

```
.C11   = 000053          1284#
.C20   = 000064          1285#   2397
.C21   = 000065          1286#
.C40   = 000110          1287#
.C79   = 000157          1288#   2261
.DAPNT= 000171           1166#
.DCRAD= 054433           1179#
.DEMP  = 000152          1154#
.DISAC= 000137           1232#
.DMAIN= 000141           1134#   2666    2775
.DPNT  = 000130          1227#
.DRECT= 000103           1142#   2063
.EAPNT= 000131           1164#
.EEMP  = 000112          1152#   1967
.EINST= 000111           1169#   2215    2218
.EMAIN= 000101           1132#
.ENAC  = 000136          1231#
.EOS   = 000112          1123#   2451
.EPNT  = 000127          1225#
.ERPL  = 000151          1171#   2215
.ESC   = 000033          1240#   1967    2029    2063    2087    2090    2093    2193    2196    2218    2242    2246    2261
                         2264    2397    2399    2451    2666    2775    4157    4159
.ESCZ  = 055033          1256#
.IABT  = 000137          1157#   4157
.LKKB  = 000105          1137#
.LNFED= 000012           1100#   2246    2399
.0     = 000117          1191#   1967    2029    2063    2087    2090    2093    2193    2196    2246    2261    2264    2399
                         2451    2666    2775    4157    4159
.P     = 000120          1241#   2218
.PSCRN= 000150           1233#
.RABT  = 000140          1213#   4157
.RDCUR= 000131           1187#   2087    2090    2093    2261    2399
.RESET= 000122           1258#
.R00   = 000040          1276#   2397
.R00C1= 025440           1268#
.R00C2= 032040           1270#
.R01   = 000041          1277#
.R12   = 000054          1278#
.R22   = 000066          1279#
.R23   = 000067          1280#   2242    2261
.R23CO= 020067           1266#
.R23C7= 057467           1181#
.TAB   = 000011          1102#
.TCUCH= 000127           1198#   2193    2196    2246    2264
.TSTER= 000124           1243#
.TXRCK= 000135           1203#
.TXTCK= 000136           1208#   2029
.UNLKK= 000145           1139#   4159
.XMTAL= 000126           1193#   2029    2451
.Y     = 000131          1186#   2242    2261    2397
```

```
COMMEN   772#
ENDCOM   772#
ERROR    672#  1605  1655  1681  1686  1698  1704  1728  1735  1773  1784  1821  1836  1884  1900
        1925  1933  2019  2046  2058  2101  2112  2150  2158  2162  2201  2207  2252  2256  2269
        2273  2354  2371  2404  2410  2420  2426  2461  2465  2468  2500  3558  3623  3640  3786
        3891  3897  4153  4169  4308
ESCAPE   772#
MULT     772#
NEWTST   772#  1520  1622  1668  1714  1750  1797  1848  1985  2079  2135  2175  2234  2293  2389
        2440  2486  2608  2678  2722  2746
POP      772#  1464  1467  1559  1563  1610  3046  3053  3057  3130  3132  3178  3180  3181  3189
        3200  3231  3399  3592  3609  3614  3716  3775  3777  3903  3953  4038  4084  4297  4299
        4300  4699  4736
PUSH     772#  1531  1543  1547  1687  1705  1736  1774  1776  1778  1785  1804  1805  1806  1935
        2010  2011  2020  2035  2036  2059  2114  2275  2321  2322  2342  2343  2372  2411  2427
        2491  2493  2494  2510  2511  2522  2523  2524  2688  2690  2691  3061  3062  3133  3160
        3163  3165  3182  3191  3223  3237  3318  3550  3551  3552  3559  3561  3562  3563  3564
        3594  3597  3599  3645  3668  3712  3729  3731  3787  3836  3879  3881  3901  3942  4025
        4312  4658  4720
SCOPE    673#  1522  1623  1669  1715  1751  1798  1849  1986  2080  2136  2176  2235  2294  2390
        2441  2487  2543  2609  2679  2723  2747
SETTRA  4771#  4781  4782  4783  4784
SETUP    772#  2796
SKIP     772#
SLASH    772#
SPACE    772#
STARS    772#   782   795   837  1021  1023  1042  1044  1061  1063  1069  1071  1078  1080  1092
        1094  1104  1106  1127  1129  1160  1162  1175  1177  1217  1219  1221  1223  1236  1238
        1290  1293  1320  1323  1333  1340  1410  1416  1510  1511  1518  1521  1614  1620  1622
        1658  1665  1668  1708  1712  1714  1741  1748  1750  1789  1795  1797  1839  1846  1848
        1971  1983  1985  2069  2077  2079  2128  2133  2135  2165  2173  2175  2222  2232  2234
        2280  2291  2293  2379  2387  2389  2432  2438  2440  2473  2484  2486  2531  2599  2606
        2608  2669  2676  2678  2713  2720  2722  2735  2744  2746  2792  2794  2865  2868  2883
        2887  2912  2915  2950  2953  3022  3026  3104  3107  3118  3128  3170  3176  3194  3198
        3226  3229  3240  3245  3281  3285  3304  3316  3475  3487  3542  3547  3570  3572  3585
        3590  3603  3607  3626  3629  3648  3651  3658  3666  3722  3724  3738  3741  3753  3755
        3769  3773  3790  3792  3795  3807  3810  3823  3834  3908  3913  3934  3940  3960  3966
        3979  3982  3994  3999  4017  4024  4041  4044  4054  4056  4066  4068  4077  4081  4109
        4116  4136  4139  4144  4146  4172  4176  4226  4233  4273  4276  4291  4295  4317  4319
        4328  4392  4437  4510  4567  4645  4713  4754
SWRSU   2828#
TRMTRP  4771#
TYPBIN   772#
TYPDEC   772#  2555  4553
TYPNAM   644#  2838
TYPNUM   772#
TYPOCS   772#  1473  1478  1485  1490  2981  2989  3081  3093
TYPOCT   772#  4525  4550
TYPTXT   772#
$$CMRE   795#
$$CMTM   795#
$$ESCA   772#
$$NEWT   772#  1521  1622  1668  1714  1750  1797  1848  1985  2079  2135  2175  2234  2293  2389
        2440  2486  2608  2678  2722  2746
$$SET   4771#  4781  4782  4783  4784
$$SETU  2814#
```

```
$$SKIP    772#
.EQUAT    644#    667
.HEADE    644#
.SETUP    644#   1320
.SWRHI    644#    654
.SWRLO    644#    667#
.SACTI    644#    782
.SCATC    644#    772
.SCMTA    644#    795
.SEOP     644#   2531
.SERRO    644#   4392
.SERRT    644#   4510
.SPOWE    644#   4713
.SSAVE    644#
.SSCOP    644#   4328
.STRAP    644#   4754
.STYPD    644#   4645
.STYPE    644#   4437
.STYPO    644#   4567
```

```
ADD     1443  1444  1503  1938  2823  2894  2906  2933  3016  3074  3113  3274  3397  3582  3696
        3875  4052  4287  4465  4533  4596  4606  4678
ASL     2997  3272  4049  4050  4051  4530  4531  4532  4767
ASLB    3295  4683
ASRB    4029  4030  4031
BCC     4684
BEQ     1358  1364  1381  1398  1497  1500  1507  1535  1539  1546  1569  1572  1575  1592  1603
        1650  1685  1703  1731  1782  1817  1862  1908  1910  1919  1930  1940  2015  2052  2054
        2108  2153  2204  2210  2255  2272  2304  2326  2367  2370  2407  2423  2464  2521  2528
        2560  2568  2630  2704  2759  2841  2862  2930  2973  3034  3068  3078  3090  3149  3157
        3159  3206  3209  3211  3215  3218  3221  3234  3259  3263  3290  3297  3325  3330  3338
        3355  3371  3381  3385  3391  3395  3412  3448  3494  3515  3526  3577  3619  3634  3680
        3683  3701  3703  3707  3746  3798  3802  3813  3817  3850  3861  3865  3873  3888  3975
        3992  4014  4048  4071  4086  4088  4096  4120  4122  4149  4165  4182  4187  4197  4242
        4249  4257  4323  4360  4362  4364  4368  4377  4409  4412  4430  4433  4468  4535  4540
        4546  4557  4623
BGE     4380  4502
BGT     2550  4630  4692
BHI     2637  3052  3342  4092  4267  4366
BHIS    1608  1913  3867  3919  4190
BIC     1527  1803  1857  1886  1948  1995  2023  2050  2298  2319  2334  2503  2530  2547  2582
        2620  2661  2695  2943  2944  2976  2977  3002  3143  3236  3261  3266  3288  3291  3292
        3301  3323  3328  3353  3357  3368  3431  3445  3531  3533  3617  3671  3675  3726  3757
        3840  3984  4002  4046  4118  4131  4132  4239  4243  4254  4620  4746
BICB    4033
BIS     1459  1462  1505  1612  1765  1906  2007  2009  2025  2034  2320  2341  2509  2587  2702
        2937  2938  2939  2966  2969  2970  2971  3003  3030  3044  3294  3326  3345  3349  3356
        3361  3366  3374  3386  3393  3409  3415  3419  3423  3427  3440  3451  3490  3530  3674
        3728  3841  3989  4012  4133  4268  4625  4626  4686  4687
BISB    4034  4522
BIT     1496  1499  1555  1637  1643  1649  1651  1769  1826  1895  2051  2350  2520  2583  2629
        2703  3136  3210  3217  3220  3233  3258  3296  3329  3339  3354  3380  3384  3390  3403
        3405  3514  3677  3679  3682  3690  3702  3845  3849  3858  3991  4013  4059  4070  4148
        4241  4246  4302  4322  4345  4359  4367  4374  4411  4418  4429
BITB    3871  4490
BLO     1538  1868  1927  1955  2309  2349  2376  2635  2764  2871  2903  3018  3050  3151  3250
        3848  3857  4090  4199  4262
BLOS    2362
BLT     2896  3115  4481  4631  4675  4691
BMI     1536  1541  1950  2586  2760  2931  3048  4682
BNE     1351  1356  1377  1394  1401  1425  1427  1442  1469  1556  1562  1565  1633  1640  1644
        1648  1652  1770  1772  1819  1827  1832  1834  1894  1896  1898  1905  1943  2042  2044
        2049  2118  2351  2353  2358  2515  2584  2622  2644  2648  2652  2656  2710  2800  2839
        2941  2975  2979  2998  3037  3055  3059  3066  3070  3101  3137  3139  3145  3147  3153
        3186  3188  3248  3265  3335  3340  3343  3348  3352  3360  3365  3373  3378  3404  3406
        3408  3414  3418  3422  3426  3430  3439  3444  3450  3499  3504  3506  3508  3520  3529
        3581  3678  3685  3691  3693  3695  3710  3846  3852  3859  3890  3899  3923  3927  3948
        3950  3952  3971  4010  4060  4073  4098  4142  4223  4247  4279  4303  4305  4307  4325
        4346  4375  4419  4427  4462  4470  4477  4491  4498  4523  4549  4621  4680  4735
BPL     1430  1465  1552  1941  3268  3489  4036  4126  4424  4456  4495  4619  4666  4696
BR      1331  1354  1369  1371  1379  1383  1399  1405  1408  1431  1445  1566  1576  1580  1583
        1586  1589  1596  1609  1680  1682  1697  1699  1727  1729  1780  1883  1885  1901  1917
        2013  2038  2047  2061  2100  2102  2119  2149  2151  2200  2202  2213  2251  2253  2268
        2270  2324  2329  2345  2355  2360  2364  2403  2405  2419  2421  2460  2462  2497  2513
        2519  2564  2626  2633  2642  2646  2650  2654  2658  2663  2706  2711  2765  2767  2774
        2822  2832  2843  2850  2856  2875  2907  2909  2934  2942  2995  3005  3040  3056  3064
```

F10


```
                3098  3141  3155  3161  3216  3270  3299  3327  3346  3350  3358  3362  3369  3375  3410
                3416  3420  3424  3428  3433  3436  3441  3446  3452  3454  3491  3502  3510  3512  3522
                3555  3641  3681  3686  3699  3705  3733  3734  3744  3748  3853  3862  3874  3878  3883
                3925  3932  3976  4028  4063  4100  4104  4201  4219  4245  4250  4260  4264  4270  4310
                4348  4354  4357  4370  4373  4458  4474  4484  4493  4500  4528  4552  4559  4597  4612
                4633  4677  4694  4729  4749
CLC             4124
CLR             1325  1347  1374  1386  1390  1391  1446  1454  1455  1456  1457  1466  1502  1529  1530
                1636  1756  1854  1887  1891  2004  2024  2039  2111  2156  2188  2347  2408  2424  2447
                2456  2457  2502  2544  2545  2562  2575  2631  2754  2755  2798  2811  2812  2818  2825
                2897  2899  2910  2917  2945  2954  2963  2965  3001  3006  3007  3009  3015  3039  3134
                3135  3257  3276  3277  3298  3432  3566  3567  3653  3654  3655  3656  3676  3689  3735
                3761  3762  3764  3765  3838  3842  3844  3855  4057  4180  4253  4286  4301  4321  4372
                4386  4521  4610  4669  4672  4733
CLRB            1403  1470  2185  2628  3496  3501  3575  4179  4185  4371  4499  4698
CMP             1397  1441  1607  1684  1702  1730  1816  1831  1867  1893  1909  1912  1918  1926  1929
                2041  2107  2116  2117  2203  2209  2308  2325  2348  2361  2366  2369  2375  2406  2422
                2799  2835  2840  2870  2895  2902  3017  3036  3065  3067  3089  3114  3249  3324  3334
                3377  3394  3507  3519  3618  3633  3692  3700  3847  3856  3866  3918  4189  4248  4261
                4266  4355  4379  4426  4690
CMPB            1350  1355  1376  1380  1393  1400  1537  1561  1568  1571  1574  1781  1904  1907  1942
                1954  2014  2048  2053  2152  2254  2271  2357  2463  2634  2636  2643  2647  2651  2655
                2709  2763  3049  3051  3144  3146  3150  3152  3156  3214  3262  3289  3341  3347  3351
                3359  3364  3370  3372  3407  3411  3413  3417  3421  3425  3429  3438  3443  3447  3449
                3576  3797  3801  3812  3816  3864  3922  3926  3974  4072  4085  4087  4089  4091  4095
                4097  4119  4121  4164  4186  4196  4198  4256  4278  4361  4365  4467  4469  4476  4497
                4501
COM             2585
DEC             1639  1647  1771  1818  1833  1861  1897  2043  2303  2352  2548  2940  2974  3138  3187
                3271  3525  3580  3684  3694  3709  3851  3860  3872  3947  3949  3951  3970  4035  4263
                4304  4306  4324  4529
DECB            3148  3158  3799  3803  4480  4483  4618  4629
EMT              672
HALT             779  4425  4428  4457  4728  4748
INC             1396  1606  1899  1902  2045  2182  2518  2546  2838  2994  3275  3333  3388  3511  3523
                3556  3697  3863  3876  3892  4008  4188  4193  4221  4235  4252  4255  4259  4265  4309
                4378  4414  4624  4632  4676  4734
INCB            1953  3154  3814  3818  3921  3930  3973  4062  4383  4408  4503
IOT              673
JMP              793   794  1326  1330  1343  1365  1368  1428  1432  1869  1946  2310  2529  2593  2627
                2707  2733  2770  2863  2864  3253  3267  3280
JSR             1327  1328  1329  1345  1349  1352  1359  1362  1366  1367  1370  1375  1382  1392  1407
                1433  1453  1458  1463  1549  1554  1558  1598  1627  1629  1631  1633  1635  1646  1674
                1679  1688  1696  1706  1719  1722  1726  1737  1757  1763  1768  1775  1779  1786  1787
                1802  1807  1809  1811  1815  1824  1830  1853  1856  1859  1864  1866  1871  1872  1882
                1903  1921  1937  1945  1990  1991  1994  1998  2006  2012  2016  2021  2028  2037  2055
                2060  2067  2085  2086  2099  2115  2141  2148  2159  2180  2184  2187  2192  2199  2212
                2239  2241  2250  2260  2267  2276  2299  2302  2307  2311  2323  2327  2330  2344  2359
                2374  2394  2396  2402  2412  2418  2428  2446  2448  2450  2459  2466  2495  2496  2501
                2506  2512  2517  2525  2526  2570  2613  2615  2617  2619  2625  2639  2641  2659  2662
                2684  2686  2687  2692  2698  2705  2708  2727  2729  2731  2732  2753  2761  2762  2769
                2772  2935  2987  3035  3042  3045  3063  3073  3091  3332  3517  3553  3554  3565  3579
                3596  3598  3635  3698  3711  3714  3715  3732  3747  3750  3779  3877  3882  3893  3900
                3902  3987  4001  4083  4156  4162  4194  4206  4208  4210  4212  4214  4216  4218  4238
                4269  4283  4285  4344  4420  4475  4482  4489
MOV             1342  1346  1353  1360  1361  1373  1378  1384  1385  1389  1418  1419  1420  1421  1422
```

```
                1423  1434  1435  1436  1437  1438  1439  1440  1447  1449  1450  1451  1452  1461  1464
                1468  1473  1478  1483  1485  1490  1504  1508  1523  1524  1526  1532  1533  1534  1542
                1544  1547  1553  1560  1564  1578  1579  1582  1585  1588  1599  1601  1611  1624  1625
                1642  1645  1653  1654  1670  1671  1673  1675  1677  1683  1687  1690  1691  1692  1693
                1694  1695  1700  1705  1716  1717  1720  1721  1723  1724  1725  1732  1734  1736  1752
                1753  1754  1755  1758  1759  1760  1761  1762  1764  1766  1767  1774  1777  1778  1785
                1799  1800  1804  1805  1806  1808  1812  1813  1814  1820  1825  1828  1829  1835  1850
                1851  1855  1860  1874  1875  1876  1877  1878  1879  1880  1881  1888  1889  1890  1892
                1916  1920  1924  1931  1932  1936  1944  1956  1987  1988  1992  1993  1996  1997  2005
                2008  2010  2011  2020  2026  2027  2033  2035  2036  2040  2059  2081  2082  2084  2096
                2097  2098  2103  2104  2105  2113  2114  2137  2138  2140  2142  2145  2146  2147  2154
                2177  2178  2181  2183  2186  2189  2190  2191  2198  2205  2206  2211  2236  2237  2240
                2249  2257  2258  2259  2266  2274  2275  2295  2296  2300  2312  2313  2314  2315  2316
                2318  2321  2322  2328  2331  2332  2333  2335  2336  2337  2338  2339  2340  2342  2343
                2346  2356  2363  2368  2372  2373  2391  2392  2395  2401  2409  2411  2413  2414  2415
                2416  2417  2425  2427  2442  2443  2449  2458  2488  2489  2492  2493  2494  2498  2499
                2504  2505  2507  2508  2510  2511  2516  2522  2523  2524  2551  2555  2559  2563  2567
                2576  2577  2580  2581  2588  2610  2611  2614  2616  2618  2632  2645  2649  2653  2657
                2660  2680  2681  2683  2685  2689  2690  2691  2693  2694  2696  2697  2699  2700  2701
                2724  2725  2728  2730  2748  2749  2752  2757  2758  2768  2771  2797  2801  2802  2803
                2804  2805  2806  2807  2808  2809  2810  2814  2815  2816  2817  2849  2821  2824  2826
                2827  2828  2829  2830  2833  2834  2836  2837  2847  2849  2853  2855  2859  2860  2872
                2879  2880  2889  2890  2891  2892  2893  2898  2900  2901  2905  2918  2920  2921  2922
                2924  2925  2926  2927  2928  2936  2947  2955  2957  2958  2959  2960  2961  2962  2964
                2968  2981  2989  2996  2999  3000  3008  3010  3011  3012  3013  3014  3027  3028  3029
                3041  3043  3047  3054  3058  3061  3062  3072  3076  3079  3082  3088  3093  3111  3112
                3131  3132  3133  3140  3142  3160  3164  3166  3179  3180  3181  3182  3183  3184  3185
                3190  3191  3201  3202  3204  3205  3219  3222  3223  3232  3237  3269  3273  3278  3319
                3320  3321  3322  3331  3336  3344  3367  3379  3396  3400  3401  3509  3521  3532  3550
                3551  3552  3557  3560  3561  3562  3563  3564  3574  3578  3593  3594  3595  3597  3599
                3610  3611  3612  3614  3615  3621  3622  3631  3632  3636  3637  3642  3643  3644  3645
                3669  3670  3672  3673  3688  3704  3713  3716  3727  3730  3731  3758  3759  3760  3763
                3776  3777  3782  3787  3837  3839  3843  3870  3880  3881  3885  3886  3896  3901  3903
                3915  3916  3924  3928  3943  3944  3945  3946  3953  3954  3968  3985  3986  3988  4003
                4004  4005  4006  4007  4026  4027  4038  4058  4084  4130  4134  4150  4152  4155  4161
                4166  4168  4191  4205  4209  4211  4215  4217  4236  4237  4240  4244  4281  4282  4284
                4298  4299  4300  4313  4350  4351  4353  4356  4369  4381  4382  4384  4385  4388  4389.
                4410  4415  4431  4434  4459  4460  4464  4478  4520  4525  4534  4539  4544  4545  4547
                4550  4554  4561  4562  4593  4601  4602  4603  4609  4616  4634  4635  4636  4637  4638
                4659  4660  4661  4662  4663  4664  4665  4670  4673  4693  4699  4700  4701  4702  4703
                4705  4706  4718  4719  4720  4721  4722  4723  4724  4725  4726  4727  4732  4736  4737
                4738  4739  4740  4741  4742  4743  4763  4764  4768
MOVB            1395  1402  1404  1528  1593  1595  1628  1630  1632  1634  1676  1810  1823  1858  1865
                1870  1914  1915  1922  1923  1999  2000  2001  2002  2003  2017  2018  2056  2057  2109
                2110  2143  2144  2157  2160  2161  2301  2306  2317  2454  2455  2467  2624  2638  2640
                2813  3203  3207  3212  3235  3260  3264  3287  3300  3337  3387  3435  3453  3495  3500
                3516  3518  3549  3638  3639  3708  3743  3745  3749  3780  3781  3784  3785  3804  3819
                3868  3869  3894  3895  3917  3929  3969  4009  4011  4032  4037  4093  4099  4102  4141
                4184  4195  4203  4204  4207  4213  4251  4258  4387  4417  4461  4488  4496  4594  4595
                4598  4599  4600  4604  4607  4608  4627  4668  4671  4685  4688  4697  4766
NEG             4605  4667
NOP             1460  1657  1707  1738  1837  1969  2126  2164  2220  2277  2377  2429  2469  2571  2572
                2573  2874  2967  3031  4163
RESET           2561  2569  2923  3019  3252
ROL             4611  4613  4614  4615  4617
ROLB            4129
```

```
ROR     4061
RTI     1448  2589  2816  2919  2956  3279  3293  3302  3402  3497  3527  3534  4390  4436  4466
        4639  4707  4747
RTS     1952  1957  2881  2911  3020  3075  3102  3116  3167  3192  3224  3238  3254  3568  3583
        3600  3646  3657  3717  3736  3751  3766  3788  3800  3805  3815  3820  3904  3920  3955
        3972  4015  4039  4053  4064  4074  4101  4135  4143  4170  4224  4271  4288  4314  4326
        4505  4564  4769
RTT     2820  2821
SEC     4128
SPL     2848
SUB     2946  3060  3382  3778  3783  3884  3898  4151  4167  4192  4200  4202  4416  4674
SWAB    1484  1594  1600  1604  1733  1951  2155  2988  3092  3213  3931
SXT     2854
TRAP    2578  4771  4781  4782  4783  4784
TST     1357  1363  1424  1426  1540  1545  1551  1591  1939  1949  2208  2514  2579  2766  2773
        2831  2873  2904  2929  2948  2978  3033  3038  3069  3077  3100  3255  3488  3498  3505
        3706  3887  3889  4047  4311  4352  4376  4423  4432  4463  4471  4492  4556  4622  4679
        4689  4765
TSTB    1429  1506  1602  2527  2621  2861  2972  3208  3247  3493  3503  3528  4125  4181  4222
        4363  4455  4494  4548  4681  4695
WAIT    2623
.ASCII   834   835  4802  4843  5037  5131  5134  5138  5144  5146  5148  5150  5152  5167  5174
        5189  5192  5199  5216  5298  5306
.ASCIZ   833   836  2595  2845  4565  4751  4807  4818  4825  4837  4848  4856  4861  4873  4882
        4889  4893  4900  4906  4914  4921  4930  4938  4944  4949  4957  4963  4968  4975  4986
        4995  5003  5011  5019  5027  5040  5045  5048  5052  5056  5063  5071  5079  5085  5091
        5101  5115  5120  5127  5154  5161  5162  5163  5164  5180  5186  5205  5210  5215  5226
        5232  5240  5246  5254  5260  5268  5275  5283  5288  5307  5311
.BLKB   5356  5358  5360
.BLKW   1064  1065  1066  1067  4712
.BYTE    805   806   811   812   827   828   829   830   993   995   997  1003  1011  1012  1014
        1476  1477  1481  1482  1488  1489  1493  1494  1967  2029  2063  2087  2090  2093  2193
        2196  2216  2218  2242  2246  2261  2264  2397  2399  2451  2598  2666  2775  2984  2985
        2992  2993  3085  3086  3096  3097  4105  4106  4157  4159  4640  4641  4642  4643  5314
        5321  5328  5334  5340  5342  5344  5346  5351
.ENABL   644
.END    5362
.ENDC    649   663   665   666   667   672   758   772   782   783   789   791   796   803   805
         831   832   833   834   838  1022  1024  1043  1045  1062  1064  1070  1072  1079  1081
        1093  1095  1105  1107  1128  1130  1161  1163  1176  1178  1218  1220  1222  1224  1237
        1239  1291  1294  1320  1321  1324  1334  1341  1411  1417  1477  1478  1482  1483  1489
        1490  1494  1495  1511  1512  1519  1522  1523  1524  1525  1615  1621  1623  1624  1625
        1626  1659  1666  1669  1670  1671  1672  1709  1713  1715  1716  1717  1718  1742  1749
        1751  1752  1753  1754  1790  1796  1798  1799  1800  1801  1840  1847  1849  1850  1851
        1852  1972  1984  1986  1987  1988  1989  2070  2078  2080  2081  2082  2083  2129  2134
        2136  2137  2138  2139  2166  2174  2176  2177  2178  2179  2223  2233  2235  2236  2237
        2238  2281  2292  2294  2295  2296  2297  2380  2388  2390  2391  2392  2393  2433  2439
        2441  2442  2443  2444  2474  2485  2487  2488  2489  2490  2532  2535  2537  2538  2539
        2541  2544  2550  2553  2554  2558  2566  2593  2594  2595  2598  2599  2600  2607  2609
        2610  2611  2612  2670  2677  2679  2680  2681  2682  2714  2721  2723  2724  2725  2726
        2736  2745  2747  2748  2749  2750  2793  2795  2801  2802  2804  2806  2808  2810  2811
        2812  2814  2826  2828  2838  2842  2845  2866  2869  2884  2888  2913  2916  2951  2954
        2985  2986  2993  2994  3023  3027  3086  3087  3097  3098  3105  3108  3119  3129  3171
        3177  3195  3199  3227  3230  3241  3246  3283  3287  3305  3317  3476  3488  3543  3548
        3571  3573  3586  3591  3604  3608  3627  3630  3649  3652  3659  3667  3723  3725  3739
        3742  3754  3756  3770  3774  3791  3793  3796  3808  3811  3824  3835  3909  3914  3935
```

```
                3941    3961    3967    3980    3983    3995    4000    4018    4025    4042    4045    4055    4057    4067    4069
                4078    4082    4110    4117    4137    4140    4145    4147    4173    4177    4227    4234    4274    4277    4292
                4296    4318    4320    4329    4335    4340    4345    4347    4358    4361    4362    4363    4365    4367    4374
                4378    4383    4384    4388    4391    4392    4393    4399    4408    4415    4420    4421    4422    4423    4429
                4436    4437    4438    4461    4511    4529    4567    4568    4646    4714    4726    4736    4746    4747    4754
                4755    4764    4767    4780    4781    4782    4783    4784    4785
.EQUIV           672     673     675     690     691     720     721     722     723     724     725     726     727     728     729
                 748     749     750     751     752     753     754     755     756     757
.EVEN           1016    1958    2845    4566    4753    4928    5036    5355
.IF              645     662     663     664     665     666     667     670     730     758     782     787     789     795     802
                 804     831     832     833     837     838    1021    1023    1042    1044    1061    1063    1069    1071    1078
                1080    1092    1094    1104    1106    1127    1129    1160    1162    1175    1177    1217    1219    1221    1223
                1236    1238    1290    1293    1320    1323    1333    1340    1410    1416    1476    1477    1481    1482    1488
                1489    1493    1494    1510    1511    1518    1521    1523    1524    1525    1614    1620    1622    1624    1625
                1626    1658    1665    1668    1670    1671    1672    1708    1712    1714    1716    1717    1718    1741    1748
                1750    1752    1753    1754    1789    1795    1797    1799    1800    1801    1839    1846    1848    1850    1851
                1852    1971    1983    1985    1987    1988    1989    2069    2077    2079    2081    2082    2083    2128    2133
                2135    2137    2138    2139    2165    2173    2175    2177    2178    2179    2222    2232    2234    2236    2237
                2238    2280    2291    2293    2295    2296    2297    2379    2387    2389    2391    2392    2393    2432    2438
                2440    2442    2443    2444    2473    2484    2486    2488    2489    2490    2531    2535    2536    2537    2538
                2540    2541    2543    2549    2552    2554    2558    2559    2575    2593    2594    2595    2599    2606    2608
                2610    2611    2612    2669    2676    2678    2680    2681    2682    2713    2720    2722    2724    2725    2726
                2735    2744    2746    2748    2749    2750    2792    2794    2797    2801    2802    2804    2806    2808    2810
                2811    2812    2814    2826    2838    2840    2844    2865    2868    2883    2887    2912    2915    2950    2953
                2984    2985    2992    2993    3022    3026    3085    3086    3096    3097    3104    3107    3118    3128    3170
                3176    3194    3198    3226    3229    3240    3245    3281    3285    3304    3316    3475    3487    3542    3547
                3570    3572    3585    3590    3603    3607    3626    3629    3648    3651    3658    3666    3722    3724    3738
                3741    3753    3755    3769    3773    3790    3792    3795    3807    3810    3823    3834    3908    3913    3934
                3940    3960    3966    3979    3982    3994    3999    4017    4024    4041    4044    4054    4056    4066    4068
                4077    4081    4109    4116    4136    4139    4144    4146    4172    4176    4226    4233    4273    4276    4291
                4295    4317    4319    4328    4334    4339    4344    4345    4357    4359    4360    4361    4363    4364    4365
                4374    4376    4384    4385    4390    4391    4392    4398    4408    4411    4418    4420    4421    4423    4426
                4429    4436    4437    4461    4510    4528    4544    4567    4645    4713    4726    4736    4744    4746    4751
                4754    4763    4767    4771    4781    4782    4783    4784    4785
.IFF             663     665     666     667     670     783     789     791     796     802     804     831     838    1021    1023
                1042    1044    1061    1063    1069    1071    1078    1080    1092    1094    1104    1106    1127    1129    1160
                1162    1175    1177    1217    1219    1221    1223    1236    1238    1290    1293    1320    1323    1333    1340
                1410    1416    1477    1478    1482    1489    1493    1494    1511    1518    1522    1523    1524    1614    1620
                1623    1624    1658    1665    1669    1670    1708    1712    1715    1716    1741    1748    1751    1752    1789
                1795    1798    1799    1800    1839    1846    1849    1850    1851    1971    1983    1986    1987    2069    2077
                2080    2081    2128    2133    2136    2137    2165    2173    2176    2177    2222    2232    2235    2236    2280
                2291    2294    2295    2296    2379    2387    2390    2391    2432    2438    2441    2442    2473    2484    2487
                2488    2489    2532    2540    2544    2549    2552    2594    2599    2606    2609    2610    2611    2669    2676
                2679    2680    2681    2713    2720    2723    2724    2725    2735    2744    2747    2748    2749    2792    2794
                2801    2865    2868    2883    2887    2912    2915    2950    2953    2985    2986    2993    3022    3026    3086
                3087    3097    3104    3107    3118    3128    3170    3176    3194    3198    3226    3229    3240    3245    3281
                3285    3304    3316    3475    3487    3542    3547    3570    3572    3585    3590    3603    3607    3626    3629
                3648    3651    3658    3666    3722    3724    3738    3741    3753    3755    3769    3773    3790    3792    3795
                3807    3810    3823    3834    3908    3913    3934    3940    3960    3966    3979    3982    3994    3999    4017
                4024    4041    4044    4054    4056    4066    4068    4077    4081    4109    4116    4137    4140    4144    4146
                4172    4176    4226    4233    4273    4276    4291    4295    4317    4319    4329    4358    4361    4362    4365
                4391    4393    4398    4411    4436    4437    4438    4511    4528    4544    4568    4646    4714    4746    4755
                4764
.IFT            2845    4373    4421
.IFTF           2845    4371    4420
.IIF             644     649     654     659     660     661     663     666     667     779     837    1474    1479    1486    1491
```

```
                2537  2544  2545  2556  2594  2599  2802  2804  2810  2811  2812  2826  2827  2840  2982
                2990  3093  3094  4335  4336  4337  4338  4339  4340  4372  4373  4388  4391  4392  4399
                4400  4401  4402  4403  4426  4437  4510  4526  4551  4555  4780  4781  4782  4783  4784
.IRP            1320  1464  1468  1521  1532  1544  1547  1560  1564  1611  1622  1668  1687  1705  1714
                1736  1750  1774  1777  1778  1785  1797  1804  1805  1806  1848  1936  1985  2010  2011
                2020  2035  2036  2059  2079  2114  2135  2175  2234  2275  2293  2321  2322  2342  2343
                2372  2389  2411  2427  2440  2486  2492  2493  2494  2510  2511  2522  2523  2524  2608
                2678  2689  2690  2691  2722  2746  3047  3054  3058  3061  3062  3131  3132  3133  3160
                3164  3166  3179  3180  3181  3182  3190  3191  3201  3223  3232  3237  3319  3400  3550
                3551  3552  3560  3561  3562  3563  3564  3593  3594  3597  3599  3610  3614  3645  3669
                3713  3716  3730  3731  3776  3777  3787  3837  3880  3881  3901  3903  3943  3953  4026
                4038  4084  4298  4299  4300  4313  4344  4659  4699  4720  4736
.LIST            643   644   666   772   779   831  1021  1022  1023  1024  1042  1043  1044  1045  1061
                1062  1063  1064  1069  1070  1071  1072  1078  1079  1080  1081  1092  1093  1094  1095
                1104  1105  1106  1107  1127  1128  1129  1130  1160  1161  1162  1163  1175  1176  1177
                1178  1217  1218  1219  1220  1221  1222  1223  1224  1236  1237  1238  1239  1290  1291
                1293  1294  1320  1321  1323  1324  1333  1334  1340  1341  1410  1411  1416  1417  1511
                1512  1518  1519  1521  1523  1614  1615  1620  1621  1622  1624  1658  1659  1665  1666
                1668  1670  1708  1709  1712  1713  1714  1716  1741  1742  1748  1749  1750  1752  1789
                1790  1795  1796  1797  1799  1839  1840  1846  1847  1848  1850  1971  1972  1983  1984
                1985  1987  2069  2070  2077  2078  2079  2081  2128  2129  2133  2134  2135  2137  2165
                2166  2173  2174  2175  2177  2222  2223  2232  2233  2234  2236  2280  2281  2291  2292
                2293  2295  2379  2380  2387  2388  2389  2391  2432  2433  2438  2439  2440  2442  2473
                2474  2484  2485  2486  2488  2544  2599  2600  2606  2607  2608  2610  2669  2670  2676
                2677  2678  2680  2713  2714  2720  2721  2722  2724  2735  2736  2744  2745  2746  2748
                2792  2793  2794  2795  2828  2840  2845  2865  2866  2868  2869  2883  2884  2887  2888
                2912  2913  2915  2916  2950  2951  2953  2954  3022  3023  3026  3027  3104  3105  3107
                3108  3118  3119  3128  3129  3170  3171  3176  3177  3194  3195  3198  3199  3226  3227
                3229  3230  3240  3241  3245  3246  3281  3282  3283  3285  3286  3287  3304  3305  3316
                3317  3475  3476  3487  3488  3542  3543  3547  3548  3570  3571  3572  3573  3585  3586
                3590  3591  3603  3604  3607  3608  3626  3627  3629  3630  3648  3649  3651  3652  3658
                3659  3666  3667  3722  3723  3724  3725  3738  3739  3741  3742  3753  3754  3755  3756
                3769  3770  3773  3774  3790  3791  3792  3793  3795  3796  3807  3808  3810  3811  3823
                3824  3834  3835  3908  3909  3913  3914  3934  3935  3940  3941  3960  3961  3966  3967
                3979  3980  3982  3983  3994  3995  3999  4000  4017  4018  4024  4025  4041  4042  4044
                4045  4054  4055  4056  4057  4066  4067  4068  4069  4077  4078  4081  4082  4109  4110
                4116  4117  4144  4145  4146  4147  4172  4173  4176  4177  4226  4227  4233  4234  4273
                4274  4276  4277  4291  4292  4295  4296  4317  4318  4319  4320  4339  4426  4771  4780
                4781  4782  4783  4784  4785
.MACRO           667   795  2814  4771
.MCALL           644   772  2828
.NLIST           642   644   666   772   779   831  1021  1022  1023  1024  1042  1043  1044  1045  1061
                1062  1063  1064  1069  1070  1071  1072  1078  1079  1080  1081  1092  1093  1094  1095
                1104  1105  1106  1107  1127  1128  1129  1130  1160  1161  1162  1163  1175  1176  1177
                1178  1217  1218  1219  1220  1221  1222  1223  1224  1236  1237  1238  1239  1290  1291
                1293  1294  1320  1321  1323  1324  1333  1334  1340  1341  1410  1411  1416  1417  1511
                1512  1518  1519  1521  1523  1614  1615  1620  1621  1622  1624  1658  1659  1665  1666
                1668  1670  1708  1709  1712  1713  1714  1716  1741  1742  1748  1749  1750  1752  1789
                1790  1795  1796  1797  1799  1839  1840  1846  1847  1848  1850  1971  1972  1983  1984
                1985  1987  2069  2070  2077  2078  2079  2081  2128  2129  2133  2134  2135  2137  2165
                2166  2173  2174  2175  2177  2222  2223  2232  2233  2234  2236  2280  2281  2291  2292
                2293  2295  2379  2380  2387  2388  2389  2391  2432  2433  2438  2439  2440  2442  2473
                2474  2484  2485  2486  2488  2544  2599  2600  2606  2607  2608  2610  2669  2670  2676
                2677  2678  2680  2713  2714  2720  2721  2722  2724  2735  2736  2744  2745  2746  2748
                2792  2793  2794  2795  2828  2840  2845  2865  2866  2868  2869  2883  2884  2887  2888
                2912  2913  2915  2916  2950  2951  2953  2954  3022  3023  3026  3027  3104  3105  3107
```

```
            3108    3118    3119    3128    3129    3170    3171    3176    3177    3194    3195    3198    3199    3226    3227
            3229    3230    3240    3241    3245    3246    3281    3282    3283    3285    3286    3287    3304    3305    3316
            3317    3475    3476    3487    3488    3542    3543    3547    3548    3570    3571    3572    3573    3585    3586
            3590    3591    3603    3604    3607    3608    3626    3627    3629    3630    3648    3649    3651    3652    3658
            3659    3666    3667    3722    3723    3724    3725    3738    3739    3741    3742    3753    3754    3755    3756
            3769    3770    3773    3774    3790    3791    3792    3793    3795    3796    3807    3808    3810    3811    3823
            3824    3834    3835    3908    3909    3913    3914    3934    3935    3940    3941    3960    3961    3966    3967
            3979    3980    3982    3983    3994    3995    3999    4000    4017    4018    4024    4025    4041    4042    4044
            4045    4054    4055    4056    4057    4066    4067    4068    4069    4077    4078    4081    4082    4109    4110
            4116    4117    4144    4145    4146    4147    4172    4173    4176    4177    4226    4227    4233    4234    4273
            4274    4276    4277    4291    4292    4295    4296    4317    4318    4319    4320    4339    4426    4771    4780
            4781    4782    4783    4784    4785
.PAGE        795     837
.REM           1
.REPT        779    1021    1023    1042    1044    1061    1063    1069    1071    1078    1080    1092    1094    1104    1106
            1127    1129    1160    1162    1175    1177    1217    1219    1221    1223    1236    1238    1290    1293    1320
            1323    1333    1340    1410    1416    1511    1518    1614    1620    1658    1665    1708    1712    1741    1748
            1789    1795    1839    1846    1971    1983    2069    2077    2128    2133    2165    2173    2222    2232    2280
            2291    2379    2387    2432    2438    2473    2484    2599    2606    2669    2676    2713    2720    2735    2744
            2792    2794    2865    2868    2883    2887    2912    2915    2950    2953    3022    3026    3104    3107    3118
            3128    3170    3176    3194    3198    3226    3229    3240    3245    3281    3285    3304    3316    3461    3475
            3487    3542    3547    3570    3572    3585    3590    3603    3607    3626    3629    3648    3651    3658    3666
            3722    3724    3738    3741    3753    3755    3769    3773    3790    3792    3795    3807    3810    3823    3834
            3908    3913    3934    3940    3960    3966    3979    3982    3994    3999    4017    4024    4041    4044    4054
            4056    4066    4068    4077    4081    4109    4116    4144    4146    4172    4176    4226    4233    4273    4276
            4291    4295    4317    4319
.SBTTL       655     668     773     784     797     839    2533    4330    4394    4439    4512    4569    4647    4715    4756
            4772
.TITLE       644
.WORD        779     780     781     790     804     807     808     809     810     813     814     815     816     817     818
             819     820     821     822     990     992     999    1001    1005    1007    1009    1072    1073    1074    1076
            1077    1081    1082    1083    1084    1085    1086    1087    1088    1089    1097    1099    1101    1103    1105
            1109    1112    1115    1118    1121    1124    1128    1133    1135    1138    1140    1143    1146    1149    1153
            1155    1158    1161    1165    1167    1170    1172    1176    1180    1182    1183    1185    1188    1189    1192
            1194    1195    1197    1199    1200    1202    1204    1205    1207    1209    1210    1212    1214    1215    1218
            1226    1228    1242    1244    1252    1257    1259    1261    1262    1263    1264    1265    1267    1269    1271
            1272    1273    1274    1275    1298    1299    1300    1301    1302    1303    1304    1305    1306    1307    1308
            1309    1311    1312    1313    1314    1315    1316    1317    1960    1961    1962    1963    1964    1965    2122
            2123    2124    2125    2215    2549    2552    2781    2783    2787    2789    3456    3457    3458    3459    3470
            3471    3472    3473    3537    3538    3539    3541    3601    3718    3719    3720    3822    4290    4504    4537
            4542    4644    4745    5359
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

```
*,DZVTJ.SEQ/SOL/CRF/PAGNUM/NL:TOC=DZVTJ.P11
RUN-TIME: 51 48 10 SECONDS
RUN-TIME RATIO: 222/111=2.0
CORE USED:  21K  (41 PAGES)
```

Spooler runtime 19 Seconds, 81 KCS, 586 disk reads, 0 disk writes, 127 pages