intel®

APPLICATION NOTE

AP-243

Debugging with Intel on the VAX*

BRIAN VALENTINE AND STEVE CAPIE
DSO APPLICATIONS ENGINEERING

I. INTRODUCTION

The VAX*/VMS mainframe computer has always been popular for large team software development. Its ability to serve many users and the quality software tools available make it a natural choice in the engineering lab. The introduction of quality microprocessor development languages, such as the 8086, 80186 and 80286 series from Intel Corporation, has increased the versatility of the VAX, further strengthening its position as the large team software development standard machine.

However, microprocessor development is not solely software development. At some time, the VAX developed software must be integrated and debugged in the target microprocessor environment. There are no suitable VAX tools for this integration/debug phase, so the developer must turn to a development system supplied by the microprocessor manufacturer. Intel, for example, provides a Series IV development system, coupled with a sophisticated in-circuit emulator and integrated instrumentation system, to solve this difficult integration/debug phase. Consequently, software developers must use both a VAX computer and microcomputer development system (MDS) workstation to perform their microprocessor development.

Currently VAX software developers need two keyboards to perform their work efficiently:a VAX keyboard for software generation and a microcomputer development system workstation for debugging and in-circuit emulation. Often these two keyboards are not located in the same working environment. While the microcomputer development system workstation is generally placed in an engineering laboratory, the VAX keyboard is typically found in the software developer's office. Having to work in both environments creates two problems for the software developers. First, debugging or using an in-circuit emulator requires him to leave his office, second, he's required to learn two keyboard environments.

One solution to both these problems is to allow software developers to perform debugging and in-circuit emulation from their offices using their VAX keyboard. With Intel's new VAX program called CONNECT, this is now possible. CONNECT allows software developers to use their VAX keyboards as a virtual MDS workstation. Software developers can now run PSCOPE, ICE ™ and I2ICE ™ debuggers from their VAX keyboard.

Some of the highlights of CONNECT include:
1. A user friendly human interface
   — help command
   — informative and easy to understand system messages

2. A flexible menu for selecting MDS workstation.
3. The support of a VAX environment with multiple VAX/MDS workstations (performs keyboard lock out).
4. Source code is supplied with package; therefore, CONNECT may be customized to your needs.

With CONNECT, software developers can now do the following:
1. Spend less time walking from keyboard to keyboard, and spend more time developing software.
2. By only using their fingertips, access any MDS workstation connected to the VAX.
3. Perform the entire development process in the comfort of their own office.

CONNECT is another innovative feature to add to the Intel-VAX microprocessor development environment. Intel already provides cross software which includes compilers, linkers and locators. In addition, for file transfer between the VAX environment and the MDS environment, Intel provides a serial communication link (ACL) and an Ethernet communication link (NVL).

The remainder of this application note will cover the following material:
II. CONNECT environment
    A. Hardware considerations
    B. Software considerations
III. CONNECT SETUP - for the VAX system manager
    A. How to create CONNECT:CONNECT.dat configuration file
IV. USING CONNECT—for the VAX software developer
    A. How to run program CONNECT with qualifiers
    B. How to choose and select MDS workstation
    C. How to use VAX keyboard as a virtual MDS workstation
    D. Additional CONNECT commands

II. CONNECT ENVIRONMENT

A. Hardware considerations

The VAX system manager must connect all desired MDS workstations to the VAX, using RS-232 serial cables. These cables are connected from any VAX serial port to serial channel 1 on the MDS.

MDS is used in this applications note as an abbreviation for Microcomputer Development System
MDS™ a trademark of Mohawk Data Sciences
*VAX is a trademark of Digital Equipment Corporation

Example: Connect an RS-232 cable from port TTA1:on the VAX to serial channel 1 on the MDS Series IV development system.

Each MDS to be used with the CONNECT requires a serial cable from serial channel 1 to a port on the VAX. The RS-232 connection is defined as:

Pins		Pins
1	←→	1
2	←→	2
3	←→	3
4	←→	4
5	←→	5
6	←→	6
7	←→	7
20	←→	20

NOTE:
If there are problems with CONNECT communicating to the MDS, it is likely a serial cable problem. Depending on the jumper configuration of the MDS serial channel 1, a different configuration of the serial cable may be needed. The serial cable above will work with a factory standard jumper configuration of serial channel 1 on the MDS. Consult the Intellec Series II Microcomputer Development System Hardware Reference Manual #980556-002 for more information on the serial channel 1 configurations for a Series II, III or IV.

B. Software considerations

Now that a serial cable is installed linking the MDS to the VAX, all normal console input and output of the MDS should be redirected to serial channel 1.

To redirect console input and output of a Series II or Series III to serial channel 1 perform the following steps:
1. With the ISIS prompt (-) on the screen type:

```
-SPEED 300 | 600 | 1200 | 2400 | 4800 | 9600
-CONSOL :TI:,:TO:
```

SPEED is used to set the baudrate of serial channel 1 while CONSOL is used to redirect console input and console output to serial channel 1 on the Series II or III. Running CONNECT, while if there is a need to restore the console input and output back to the MDS screen, enter the following command:

```
-CONSOL :CI:,:CO:
```

NOTE:
If at any time the Series II or III encounters an error that requires a warm boot (Example:ERROR 30), the console will be automatically switched back to the

normal console input and output. Consequently, any time a warm boot occurs, the console input and output must be redirected to serial channel 1.

To redirect console input and output of a Series IV to serial channel 1, run the STTY program with the baudrate no greater than 2400 baud. See the Intellec Series IV Operating and Programming Guide #121753-004 for more detail on the STTY program. Example SIV/ STTY command:

```
>STTY BAUDRATE (2400)
CONFIG(VT100.CFG) REMOTE
```

When running CONNECT, if there is a need to restore the console input and output on the Series IV, enter the following command:

```
>STTY LOCAL
```

Please note that while the Series II or Series III is running CONSOL and the Series IV is running STTY, no user may work at these workstations.

NOTE:
All VAX serial ports that are used to connect the MDS to the VAX should be configured at a constant baud rate. The autobaud feature should be disabled. The suggested baudrate is 2400 baud for Series IV and 9600 baud for Series II.

III. CONNECT SETUP - FOR THE VAX* SYSTEM MANAGER

A. How to create CONNECT:CONNECT.DAT configuration file

(Only a System Manager needs to read this)

NOTE:
CONNECT requires a system logical name to be assigned to the directory on the VAX where the CONNECT image and configuration files will be stored. The logical name must be "CONNECT:". Also, see APPENDIX A for a listing of the CONNECT. CLD command definition file. This file should be added to the system-wide DCL command file. If this is done, any VAX user may access and execute CONNECT.

A system manager must create an ASCII file named "CONNECT:CONNECT.DAT". This file contains all the configuration information of the MDS workstations connected to the VAX. This file is used to determine what type of MDS systems have been connected to the VAX. eg. Series II, III, etc. One of the features of CONNECT is that a user may select differ-

ent workstations to use during debugging; CONNECT uses the configuration file to determine which port on the VAX the corresponding workstation is connected.

The file consists of the following configuration parameters: system, id, ice, port.

"System" refers to the type of MDS workstation. eg. Series 2/3/4.

"Id" refers to the identification number of the workstation (system manager chooses this). This ID number is any number from 00 to 99. Each system entered in the configuration file must have a unique ID number. It is suggested that when building the configuration file, the first system entered is given id = 00, the second system id = 01, etc.

"Ice" refers to the type of ICE module connected to the MDS.

"Port" refers to the serial address to which the workstation is connected to the VAX (e.g. TTA0:). For example, if you connected a Series IV to the port TTA7:on the VAX, you would enter port = tta7:in the configuration file.

NOTE:
The maximum line length for a line in the CONNECT.DATconfiguration file is 128 characters.

These four configuration parameters accompanied by their input values form a single MDS workstation record.

Each parameter and its input value are specified on a SINGLE LINE of input in the file. Each record must contain both the **system** and **port** configuration parameters. Both id and ice are optional. THE **SYSTEM PARAMETER MUST ALWAYS OCCUR FIRST IN EACH RECORD**. to add comments in the file, enter a semicolon in the first position of a new line. The syntax for each parameter and it's input value is as follows:

The system manager should now perform the following steps:

1. Go into a VAX editor (e.g. EDT) and create a new file called CONNECT:CONNECT.DAT

2. Follow the rules in forming the MDS workstation records. If there is a Series III with an ICE-51 connected to the VAX and a Series IV with an ICE-86, the configuration file created could look as follows (the semicolons are in column zero):

```
; This series 3 is in SJC's office
; Has a 557 board (224K addition)
system : 3
id : 00
ice : 51
port : ttb0
; This series IV is in LAB 2
; Note the next record uses the
; abbreviations for the parameters
s : 4
id : 01
ic : 86
p : ttb1
```

3. After the configuration file has been created, exit the editor. Now it is time to check the data entered to be sure it is in a form CONNECT can understand. To do this, run the program VCONFIG supplied with the CONNECT package. Example: RUN CONNECT:VCONFIG

4. If any errors were found, VCONFIG will print the line number, the line with the error, and a message explaining what error occurred. Fix the error by re-editing the CONNECT:CONNECT.DAT and re-run VCONFIG.

5. If no errors were detected, then VCONFIG will report

   NO ERRORS DETECTED IN ASCII FILE. SYSTEM SUCCESSFULLY CONFIGURED !

6. Be sure to set world read protection on the CONNECT:CONNECT.DAT file.

```
S(ystem)  : [ 2 | two   |    ii ]        → REQUIRED ←
            [ 3 | three| iii ]
            [ 4 | four|   iv]

ID        : [ 0..9 ] [ 0..9 ]            → optional ←

IC(e)     : [ 42 | 44 | 49 | 51 | 85 | 86 | 88 ]    → optional ←
            [ i2ice ] − [ 86 | 88 | 186 | 188 | 286 ]

P(ort)    : tt[ a..z ] [ 0..9 ]          → REQUIRED ←
```
NOTE:
IF the MDS configuration cannot be specified with the given parameters, it is suggested that the user assign the SYSTEM, ID and PORT parameters. Therefore, when a development engineer connects to the system with CONNECT, the ID number may be specified. Also, if the system is not a II, III, or IV, any number type may be assigned to the system (e.g. II, II or IV), but when the system is connected to with CONNECT, the ID number must be used instead of the system type.

IV. USING CONNECT - FOR THE VAX SOFTWARE DEVELOPER

A. How to run the program CONNECT

NOTE:

The CONNECT.CLD file must be installed on the VAX before CONNECT will run.

With the dollar sign ('$') prompt present, type in the following format to run CONNECT:

```
$CONNECT [/baudrate = (SPEED)]
[/macrofile = (FILENAME)] [/[NO]VT100]
```

SPEED is one of the following: 300, 600, 1200, 2400, 4800, 9600

FILENAME is any legal filename on the VAX

If the baudrate qualifier is omitted, then the default is set to 2400 baud. If the NOVT100 qualifier is omitted, then the default is set to VT100. The macrofile qualifier is explained later.

B. How to choose and select workstations when running CONNECT

When running the program CONNECT, the following header will appear:

CONNECT   Version (X$$$)

At this point CONNECT checks the validity of the CONNECT:CONNECT.DAT file. This is done by running the VCONFIG program. If an error was found, CONNECT will display to the user:

AN ERROR WAS FOUND IN THE FILE

CONNECT:CONNECT.DAT PLEASE REPORT THIS

TO THE SYSTEM MANAGER

At this point, the system manager must find and correct any error(s) in the file CONNECT:-CONNECT.DAT. Otherwise, the same error message will be printed to the screen each time CONNECT is run.

If the user typed in the macrofile qualifier, then the development system search parameters are read from the file entered for the qualifier. Otherwise, the user is prompted with the CONNECT menu screen. The screen is as follows:

```
--------------------> CONNECT MENU <--------------------

The CONNECT Options are the following:
     E(xit)
     H(elp)
     S(ystem)=attribute, IC(e)=attribute, ID=attribute

To select MDS workstation, enter any combination of the following MDS search
parameters:

Please enter MDS Option:

         The MDS options available to the user are the following:

            S(ystem) =     attribute Distinguishes among a Series 2,3 or 4
            ID       = attribute     Identifies a workstation with an id
number
            IC(e)    = attribute     Chooses between an ICE and I2ICE
workstation

         The attributes for each option are as follows:

            S(system)=[2|two|ii]
                     =[3|three|iii]
                     =[4|four|iv]
                     =[8bit] (Either a Series 2 or Series 3)
                     =[16bit] (Either a Series 3 or Series 4)

            ID       =[0..9][0..9]

            IC(e)    =[42|44|49|51|85|86|88]
                     =[i2ice]-[86|88|186|188|286]
```

None, one, two or any possible combinations of these three options may be used. When two or more of these options are used, THE COMMA IS REQUIRED AS THE DELIMITER. Any option which is omitted will be treated as a wildcard!

<EXAMPLES OF MDS OPTION INPUT FROM THE CONSOLE INPUT>

1. If the user wants to connect to any available MDS workstation with an ICE-86 workstation:

    Please enter MDS Option:ic = 86

2. If the user wants to connect to the Series II with an ICE-51 workstation:

    Please enter MDS Option:s = ii, ice = 51

3. If the user wants to connect to any 8-bit workstation.

    Please enter MDS Option:s = 8bit

4. If the user wants to connect to a Series IV with an I2ICE-186.

    Please enter MDS Option:s = four, ic = i2ice-186

5. If the user wants to connect to any available MDS workstation.

    Please enter MDS Option:(cr)

6. If the user wants to use a particular system, to continue a previous debug session, he should specify the station ID number:

    Please enter the MDS option:id = 10

Notice that whether the search parameters are read from the user interactively or from the macrofile, there should be NO LEADING, TRAILING, OR ANY SPACES WHAT-SO-EVER ON THE MDS OPTION LINE (NO EXCEPTIONS!!!).

If the input is expected from the macrofile (specified with the MACROFILE qualifier when running CONNECT), CONNECT ONLY LOOKS AT THE VERY FIRST LINE IN THE FILE. The first line MUST END WITH A CARRIAGE RETURN. If the carriage return is not included CONNECT will be unable to properly interpret the line and will print an error to the screen. If the first line in the macrofile only contains the End Of File (EOF) marker, CONNECT will display the error:

```
ERROR IN GETTING LINE FROM
(FILENAME)
<FILENAME> ONLY CONTAINS THE EOF
MARKER
EITHER PUT THE MDS SEARCH
PARAMETER(S) INTO THE FILE OR DELETE
```

```
IT ENTIRELY!
<<CONNECT TERMINATED>>
```

<EXAMPLE OF MDS OPTION INPUT FROM MACROFILE>

```
ic=i2ice-86,s=four(cr)
[EOF]
```

The previous example would be used if the user wants a series IV with an i2ice-86 workstation.

At this point, CONNECT interprets the line (from either console input or from the macrofile). If any errors were found, CONNECT will show the error, show the user what he/she typed, and give him/her the correct syntax of the incorrect MDS search parameter. If there were no errors, then one of the five following conditions (and possibly displayed messages) will occur:

1. The MDS options could not be met. The message is:

    NO WORKSTATION WAS FOUND THAT MET YOUR MDS OPTION REQUIREMENT(S).

    This means that the system specified by the user was not found in the CONNECT.DAT file. The system manager must be notified to add the system to the configuration file.

2. The particular device chosen (i.e. id number) is currently being used. The message is

    PARTICULAR DEVICE IS ALREADY IN USE, PLEASE PICK ANOTHER.

    This indicates that some other user is currently using the system specified.

3. If the first workstation found is busy, CONNECT continues its search. If no other workstation is found, then the message is:

    ALL DEVICES THAT MEET YOUR MDS OPTION REQUIREMENT(S) ARE CURRENTLY BEING USED.

4. A workstation was found, but its port address does not exist on the VAX system. The message is

    THE SPECIFIED DEVICE (TT**) DOES NOT EXIST IN THE HOST SYSTEM. PLEASE TELL SYSTEM MANAGER OF THE ERROR IN CONNECT:CONNECT.DAT ((CONNECT TERMINATED))

    This last message is fatal and will therefore exit CONNECT. All other messages will re-prompt the user with the MDS option line.

5. A workstation was found and a connection can be made!!!

    The MDS workstation is now ready for use.

CONNECT will display to the screen the MDS characteristics of your workstation and prompt you for verification.

```
        --------MDS Workstation
     Characteristics--------

System is a Series     (ALWAYS APPEARS)
Identification number is _____
ICE workstation is _____

ARE THESE CHARACTERISTICS ACCEPTABLE
(Y/N):
```

In response to the question on the screen, pressing any key other than 'Y' will re-display the entire CON-NECT menu screen. Otherwise, you will be connected via serial channel 1 to the debug workstation. If your screen remains blank, do not worry. If no data is transferred back to your terminal from the workstation within five seconds, the following error message will be displayed:

```
TIME OUT ERROR: THOUGH  THE  PORT  ADDRESS
               EXISTS  <TT**>,  THERE  IS
               NO  MDS  WORKSTATION  CON-
               NECTED TO THE VAX SYSTEM.

NOTE: IF YOU KNOW FOR A FACT THAT THERE IS A
      MDS WORKSTATION CONNECTED TO THAT
      PORT THEN EITHER THE SERIES IV IS NOT
      FUNCTIONING IN STTY PROGRAM
                     OR
      SERIES II/III IS NO LONGER IN CONSOL
      MODE

PLEASE REPORT THIS TO SYSTEM
MANAGER!!!

<<CONNECT TERMINATED>>
```

Otherwise, the Series II/III prompt or the Series IV prompt and menu line will be displayed.

C. How to use VAX keyboard as a virtual MDS workstation

Now that you are connected to the MDS workstation, any key you press or command you type will be interpreted by that workstation. Therefore, your VAX terminal now looks and acts like the console of the MDS.

D. Additional CONNECT commands

The following are additional CONNECT commands which are supported while the VAX keyboard is acting as a virtual MDS workstation:

vbreak or control-b	Breaks connection with MDS workstation and prompts you with the CONNECT menu
vdev	Tells you the system and, if possible, the ID number and ICE workstation
vhelp	Explains all the CONNECT commands listed here
vquit or control-p	Exits the program CONNECT

APPENDIX A
Source code modules for CONNECT

CONNECT is written using the VAX/VMS C Compiler. It is included in the Network/Series IV toolbox or can be purchased from Intel's Insite Users Program Library.

Insite Part Number: CONNECT, Part #AD26

Insite Mailing Address:
  Intel Corporation
  3065 Bowers Avenue
  Santa Clara, CA 95051
  Attn:Insite User's Program Library GR1-2-78
  Telephone:408-987-7256

CONNECT C source code modules:

CONNECT.C	—main module for CONNECT
VSUBMIT.C	—reads macro file
CONFIGCH.C	— checks errors of CONNECT.DAT configuration file
CCONNECT.C	— searches the configuration file for user's MDS

INTERACT.C	— handles the communications of VAX to MDS
STRUTIL.C	—string parsing routines

VCONFIG C source modules:

VCONFIG.C	— checks for errors in the CONNECT.DAT configuration file
STRUTIL.C	—string parsing routines

Executables:
  CONNECT.EXE
  VCONFIG.EXE

CONNECT command definition file:
  CONNECT.CLD

Command files to generate CONNECT:

V.COM	—generates CONNECT.EXE
VCONFIG.COM	—generates VCONFIG.EXE

Sample CONNECT configuration file:
  CONNECT.DAT

Sample CONNECT macro file:
  CONNECT.MAC

Series IV Keys Defined on VAX Terminal

The following shows you how to simulate the function keys and toggle key on the Series IV while running CONNECT on your VAX terminal:

Control sequence on VAX terminal	Series IV equivalent key
control-f (unshifted) 0	F0 (unshifted)
control-f (unshifted) 1	F1 (unshifted)
control-f (unshifted) 2	F2 (unshifted)
control-f (unshifted) 3	F3 (unshifted)
control-f (unshifted) 4	F4 (unshifted)
control-f (unshifted) 5	F5 (unshifted)
control-f (unshifted) 6	F6 (unshifted)
control-f (unshifted) 7	F7 (unshifted)
control-f (shifted) 0	F0 (shifted) ON LINE HELP
control-f (shifted) 1	F1 (shifted) ON LINE HELP
control-f (shifted) 2	F2 (shifted) ON LINE HELP
control-f (shifted) 3	F3 (shifted) ON LINE HELP
control-f (shifted) 4	F4 (shifted) ON LINE HELP
control-f (shifted) 5	F5 (shifted) ON LINE HELP
control-f (shifted) 6	F6 (shifted) ON LINE HELP
control-f (shifted) 7	F7 (shifted) ON LINE HELP
control-f t(oggle)	toggle switch

APPENDIX B

```
\* Intel Corporation 1984
   VT100 configuration file

   This file contains the VT100 configuration information for the program
   STTY *\

\* Default configuration for NRM console or Series IV "secondary console"
   (serial channel 1) : *\

AV=25;              \* NUMBER OF LINE (DECIMAL) *\
AW=F;               \* WRAPPER                  *\
AX=F;               \* X FIRST                  *\
AI=T;               \* INVISIBLE ATTRIBUTES     *\
AC=T;               \* CHARACTER ATTRIBUTES     *\
AO=20;              \* OFFSET FOR CURSOR ADDR    *\
AFBK=20;            \* BLANK CHARACTER          *\
AFIG=;              \* IGNORE CHARACTER         *\
AFKM=FF;            \* KEY MASK                 *\
AFPM=FF;            \* PRINT MASK               *\
AFDC=T;             \* DC1 - DC3 PROTOCOL        *\
AFHG=T;             \* HANG UP MODEM ON SW LOCAL *\
AFLO=F;             \* LOGOFF ON CONSOLE SWITCH   *\
AFSE=T;             \* SLOW LINE EDIT MODE       *\
AFTX=04;            \* TRANSMIT READY MASK VALUE *\

\* The following commands are used to convert keyboard sequences on your
   terminal to Series IV codes. *\

AFCU=1B41;                 \* CURSOR UP                      *\
AFCD=1B42;                 \* CURSOR DOWN                    *\
AFCR=1B43;                 \* CURSOR RIGHT                   *\
AFCL=1B44;                 \* CURSOR LEFT                    *\
AFCH=0F;                   \* CURSOR HOME                    *\
AFXZ=1B5B324B;             \* DELETE LINE                    *\
AFXA=1B5B4B;               \* DELETE RIGHT                   *\
AFXU=15;                   \* UNDO                           *\
AR=7F;                     \* RUBOUT                         *\
AFSS=13;                   \* SCREEN SUSPEND                 *\
AFSR=11;                   \* SCREEN RESUME                  *\
AFCA=03;                   \* COMMAND ABORT                  *\
AFDB=04;                   \* DEBUG                          *\
AFJA=1B4F53;               \* JOB ABORT                      *\
AB=1B;                     \* ESCAPE                         *\
AFSO=;                     \* SCREEN ON/OFF TOGGLE           *\
AFC1=;                     \* CLEAR SCROLL PART OF SCREEN *\
AFC2=;                     \* CLEAR MESSAGE PART OF SCREEN *\
AFC3=;                     \* CLEAR PROMPT PART OF SCREEN *\
AFCC=;                     \* CAUSE INTERRUPT 3 (NMI)     *\
```

231479-1

```
\* The following commands are output sequences which convert screen control
   character sequences from the Series IV sequence to the sequence expected
   by your terminal *\


AFMU=1B41;                  \* CURSOR UP *\
AFMD=1B42;                  \* CURSOR DOWN *\
AFMR=1B43;                  \* CURSOR RIGHT *\
AFML=1B44;                  \* CURSOR LEFT *\
AFMH=1B48;                  \* CURSOR HOME *\
AFMB=0D;                    \* RETURN (LF) *\
AFER=1B5B4A;                \* ERASE FROM CURRENT POSITION TO END OF SCREEN *\
AFEL=1B5B4B;                \* ERASE FROM CURRENT POSITION TO END OF LINE *\
AFAC=1B20;                  \* CURSOR CONTROL LEAD IN -- CONNECT EXPECTS 20H *\

AFAT=;                      \* THERE WILL BE NO INVERSE VIDEO ON ANY OF THE *\
AFRV=;                      \* VAX TERMINALS.                             *\
AFNV=;

AFIL=;                      \* INSERT LINE *\
AFDL=;                      \* DELETE LINE *\

\* ---------------------------------------------------------------------- *\
\* THE FOLLOWING UNSHIFTED/SHIFTED FUNCTION CODES AND TOGGLE SWITCH "MUST"
   REMAIN ON ALL VAX TERMINALS.  ONCE AGAIN, DO NOT CHANGE ANY OF THESE CODES *\

AFL0=80;                            \* UNSHIFTED FUNCTION KEY 0 *\
AFL1=81;                            \* UNSHIFTED FUNCTION KEY 1 *\
AFL2=82;                            \* UNSHIFTED FUNCTION KEY 2 *\
AFL3=83;                            \* UNSHIFTED FUNCTION KEY 3 *\
AFL4=84;                            \* UNSHIFTED FUNCTION KEY 4 *\
AFL5=85;                            \* UNSHIFTED FUNCTION KEY 5 *\
AFL6=86;                            \* UNSHIFTED FUNCTION KEY 6 *\
AFL7=87;                            \* UNSHIFTED FUNCTION KEY 7 *\
AFU0=90;                            \* SHIFTED FUNCTION KEY 0 *\
AFU1=91;                            \* SHIFTED FUNCTION KEY 1 *\
AFU2=92;                            \* SHIFTED FUNCTION KEY 2 *\
AFU3=93;                            \* SHIFTED FUNCTION KEY 3 *\
AFU4=94;                            \* SHIFTED FUNCTION KEY 4 *\
AFU5=95;                            \* SHIFTED FUNCTION KEY 5 *\
AFU6=96;                            \* SHIFTED FUNCTION KEY 6 *\
AFU7=97;                            \* SHIFTED FUNCTION KEY 7 *\
AFTS=8A;                            \* TOGGLE SCREEN       *\

\* END OF COMMANDS WHICH CANNOT BE CHANGED FOR ANY VAX TERMINAL *\
\* ---------------------------------------------------------------------- *\
                                                          231479-2
```