6.1 INSTALLATION WITH HPC MOLE

SETUP

The setup requires that two RS—232 serial lines to the VAX be available.
The first line connects to the users terminal. The second line connects
to the MOLE (to J2 on MOLE if line is configured to connect to a terminal,
else to J1 or J3; if J1 is used be sure that jumper W2 on Brain Board
is set to position 1—3).

The line connected to the MOLE must be one that the UNIX init process
will not getty on and that is user accessible. In the following,
replace /dev/tty?? with the appropriate line:

If necessary, the line must be setup by entering the following when
logged in as root or by superuser (assume UNIX 4.3 bsd):

1. Edit the file /etc/ttys and change the line corresponding to
   /dev/tty?? to set getty off, 9600 baud. The following is
   actual line from file /etc/ttys used in testing download:

   ttyj2   "/etc/getty std.9600"   vt100   off   # Bldg.7c rm.364

2. Exit the editor and enter the command:

   kill —HUD 1

3. Enter the command:

   chmod 666 /dev/tty??

User should now be able to connect to MOLE for terminal emulation
(assume MOLE reset with 9600 baud lines to terminal and MOLE):

```
% cu —s 9600 —l /dev/tty??              # Connect to MOLE through /dev/tty??
<carriage return>

EXEC,REV:B 14 JAN 87                    # MOLE response
X>
```

DETAILS OF DOWNLOAD

Following are download details, which are necessary to understand the
download setup and examples which are on following pages.

The MOLE Brain Board contains 64K memory, called MM, some of which is available
to the user. The FORMAT 200/3FFF command limits the B0 to B3 buffers so that
the range 4000/BFFF is free for the user. Download should always be into the
range 4000/BFFF so MOLE system memory is not overwritten.

LOAD 0,xxxx command loads a hex file from the console port into MM memory;
xxxx is an offset which must be specified to load the program into the
MM range 4000/BFFF, e.g. program at 8000/FFFF will load into MM 4000/BFFF
if a LOAD 0,C000 is specified.

Since the size of MM memory 4000/BFFF is 32K, programs larger than this cannot
be loaded all at once. The program must be loaded in two (or more) parts.
The cpyhex program allows a range of a hex file to be loaded to MM, e.g.

cpyhex 0/7fff file.hex

will list to standard output a hex file which only contains data in range
0/7fff.

A further complication is that user memory must be loaded from shared
memory. Blocks of memory from MM must be moved to Shared Memory, and
then to user memory. Programs less than 16K in size need not have
user memory, therefore only a move from MM to SH is required.

The MOVE command is used to move from MM to Shared Memory, e.g.
MOVE MM,4000/7FFF,SH,C000 moves 4000/7FFF of MM to C000/FFFF of SH.

The XMOVE command is used to move from Shared Memory to User Memory,e.g.
XMOVE C000/FFFF,4000 moves data at C000/FFFF (e.g. SH) to 4000/7FFF (e.g.
user memory). See HPC Personality Board Manual for more details.

Finally, the UNIX utility cu provides terminal emulation with the MOLE,
e.g. cu —s 9600 —l /dev/tty?? connects to /dev/tty?? at 9600 baud. Other
terminal emulation utilities could be used; however only cu has been
tested with the following procedure.

DOWNLOAD SETUP

In the future, automated methods of download will be provided; however, currently, the only way to download to the MOLE from the VAX is using the Intel Hex download on the MOLE Brain Board.

The following procedure was tested with MOLE Brain Board Rev.B firmware, and HPC MOLE Personality Board Monitor Rev.E firmware. The line setup for both lines was 9600 baud, no parity, 8 data bits, 1 stop bit. (If the terminal emulation or download does fail at 9600, try a slower baud rate, e.g. 4800).

It is assumed that the user has a file.hex file to download to the HPC MOLE (the linker outputs a .hex file by default).

The following commands are entered once after resetting MOLE Brain Board: (<cr> is carriage return, <shift lock on> is to force upper case for MOLE, <shift lock off> is allow lower case for UNIX).

```
% cu -s 9600 -l /dev/tty??          # Connect to MOLE through /dev/tty??
<shift lock on>                      # Talk to MOLE in upper case
<cr>                                 # Enter carriage return
EXEC,REV:B 14 JAN 87
X>PSWD                               # Enable FORMAT and LOAD commands

X>FORMAT 200/3FFF                    # Limit size of buffers B0-B3

BUFFER REGION IS 0200 TO 3FFF
X>M                                  # Go to monitor

HPC  MONITOR, REV: E , 03 NOV 87
CHIP NUMBER (DEFAULT = HPC16083):    # Answer questions as appropriate
SINGLE CHIP EMULATION (Y/N, <cr>=YES):
ROMLESS MODE (Y/N, <cr>=NO):
ERROR MESSAGE (Y/N, <cr>=YES):

M>                                   # In monitor
```

The following examples are for each time the user wishes to download.

16K DOWNLOAD EXAMPLE
_____

The first example is for download just to Shared Memory on MOLE.
Shared Memory is enabled in range C000 to FFFF (default) and all
program (in file.hex) falls in that range.

```
                                        # Assume in MOLE monitor
M>R                                     # Reset chip

CHIP IS RESET

M>e                                     # Go to EXEC

X>LOAD 0,8000                           # Wait for load, offset 8000
                                        #  so C000/FFFF of hex file go to
                                        #  4000/7FFF of MM
~.                                      # Exit cu
<Shift lock off>                        # Talk to UNIX in lower case
% cp file.hex /dev/tty??                # Send file to MOLE
% cu -s 9600 -l /dev/tty??              # Connect to MOLE
<shift lock on>                         # Talk to MOLE in upper case
<cr><cr>
LOAD DONE                               # See load done, address should
ADDRESS LAST BYTE = xxxx                # be between 4000 and 7FFF
X>MOVE MM,4000/7FFF,SH,C000             # Move data to Shared Memory

X>M                                     # DONE; Return to Monitor
```

64K DOWNLOAD EXAMPLE

The following example is for a 64K program. Shared Memory is in
range C000 to FFFF and user memory in range 200 to BFFF. Program
will be placed at addresses indicated in file.hex file.

```
M>R                                     # Reset chip

CHIP IS RESET

M>@                                     # Go to EXEC

X>LOAD 0,4000                           # Setup to load with offset 4000
                                        #  so 0/7FFF of hex file go to
                                        #  4000/BFFF in MM

~.                                      # Exit cu
<shift lock off>                        # Talk to UNIX in lower case
% cpyhex 0/7fff file.hex>/dev/tty??     # Send only 0/7FFF of .hex to MOLE
% cu -s 9600 -l /dev/tty??              # Connect to MOLE
<shift lock on>                         # Talk to MOLE in upper case
<cr><cr>
LOAD DONE                               # See load done, address between
ADDRESS LAST BYTE = xxxx                #  4000 and BFFF
X>MOVE MM,4000/7FFF,SH,C000             # Move 1st 16K block to Shared Memory
                                        #  (Program 200/3FFF)

X>M

M>XMOVE C200/FFFF,200                   # Move 1st block to User Memory
                                        #  (below 200 is not program area)

M>@

X>MOVE MM,8000/BFFF,SH,C000             # Move 2nd 16K block to Shared Memory
                                        #  (Program 4000/7FFF)

X>M

M>XMOVE C000/FFFF,4000                  # Move 2nd Block to User Memory

M>@

X>LOAD 0,C000                           # Setup to load with offset C000
                                        #  so 8000/FFFF of hex file go to
                                        #     4000/BFFF in MM

~.                                      # Exit cu
<shift lock off>                        # Talk to UNIX in lower case
% cpyhex 8000/ffff file.hex>/dev/tty??  # Send only 8000/FFFF to MOLE
% cu -s 9600 -l /dev/tty??              # Connect to MOLE
<shift lock on>                         # Talk to MOLE in upper case
<cr><cr>
LOAD DONE                               # See load done, address between
ADDRESS LAST BYTE = xxxx                #  4000 and BFFF
X>MOVE MM,4000/7FFF,SH,C000             # Move 3th 16K block to Shared Memory
                                        #  (Program 8000/BFFF)

X>M
```

```
M>XMOVE C000/FFFF,8000              # Move 3th block to user memory

M>@

X>MOVE MM,8000/BFFF,SH,C000         # Move 4th 16K block to Shared Memory
                                    #  (Program C000/FFFF)
X>M                                 # DONE; return to Monitor
```