The purpose of this section is to introduce the QuickHDL software.
Five lessons are presented to help you become familiar with the QuickHDL system.
Assumptions:

- You should be familiar with the graphical interface on your workstation: OpenWindows or OSF/Motif on Sun SPARCstations, or OSF/Motif user environment on Hewlett-Packard workstations.
- You should have a working knowledge of VHDL, Verilog, or both.

Although QuickHDL is an excellent tool to use while learning the concepts and practices of these languages, this manual is not written to teach you to create VHDL or Verilog HDL designs.

1 Lesson One

The goals of Lesson One are:

- Create a library.
- Compile a VHDL file.
- Start the simulator.
- Understand the basic simulation windows, mouse, and menu conventions.
- Run the simulator.
- List some signals.
- Use the waveform display.
- Force the value of a signal.
- Single-step through a simulation run.
- Set a breakpoint.
- Quit the simulator.

To begin, follow the steps below. (Note that the prompt and the Return that ends a command line have been omitted from the examples for simplicity.)

1. Create a new directory and go there by entering the UNIX shell commands. (We recommend you create a “training” directory if you don’t already have one, and place this new directory named “qhdl_t” within it:
   mkdir <directory_name> (mkdir qhdl_t)
   cd <directory_name> (cd qhdl_t)
2. Copy all files from the /cad/mentor/shared/pkgs/quickhdl/examples directory into your current directory by entering the command (note the period specifying the current directory):
   cp -r /cad/mentor/shared/pkgs/quickhdl/examples/* .
3. Before you compile any VHDL code, you need a design library to hold the compilation results. To create a new design library, enter the following:
   /cad/mentor/bin/qhlib work
   This creates a subdirectory named work. This subdirectory contains a special file named _info. Do not create these using UNIX shell commands-always use the qhlib command.
4. Compile the file counter.vhd by entering the command:
   /cad/mentor/bin/qvhcom counter.vhd
   NOTE: If you had not created the “work” library in the previous step, this compile command would automatically create it prior to compiling.
5. Start the simulator by entering the command:
   /cad/mentor/bin/qhsim
   NOTE: Commands previous to this point (at the shell level) are case-sensitive. Within simulator, commands are not case-sensitive.

   The Startup window comes up. This Startup window allows you to select the library and the top-level design unit to simulate. You can also select the resolution limit for this simulation. By default, the following will appear in the Startup window for this simulation run:

   - Simulator resolution: ns
   - Library: work

- Entity, Configuration, or Module: Entity counter
- Architecture: only

6. Click Load to accept these settings.
7. Bring up all the simulation windows (except the Schematic window) by entering the following line at the prompt within the QuickHDL window: view *

   The simulator uses nine windows (the first of which is displayed at invocation): QuickHDL (main), Source, Structure, List, Wave, Signals, Process, Dataflow, and Variables.

   When you start the simulator, it reads the window positions in the quickhdl.ini file. When you quit the simulator it saves the window positions in the quickhdl.ini file. (Chapter 9 describes this file.)

   The Signals window displays the names and current values of the signals in the current region. (The current region is that part of your design that you have selected in the Structure window.)

   Collapsed levels of hierarchy are indicated by plus boxes. Click on one to expand that level. When no lower levels exist, the box is empty. A minus box indicates an expanded item.

   The Structure window displays the structural hierarchy in your design. The source code for the selected design region appears in the Source window. Packages are always shown immediately under the top level. Just as with the Signals window, collapsed levels of hierarchy are indicated by plus boxes.

   The Source window shows the VHDL source for the current design region. An arrow in the window points to the next VHDL statement to be executed.

   You can set a breakpoint by clicking near the line numbers at the left of the windowpane.

   The Process window has two modes. When Active is selected, the window displays a list of the processes that are scheduled to execute in this simulation cycle. When In Region is selected, the window lists all processes in the current region.

   The Variable window displays generics, constants, and variables that are visible in the current process. Just as with the Signals and Structure

windows, collapsed levels of hierarchy are indicated by plus boxes.
The List window displays a tabular listing of values for signals you have selected.
The Wave window displays waveforms, names, and current values for signals you have selected. The OpenWindows version is described first.
When you're using the Wave window in the OpenWindows environment, you can:

- Scroll by clicking on the elevator arrows to move 1 time tick,
- Click on the scrollbar cable to move one window width,
- Use the cable anchor to move to the beginning or end.

You can zoom into an area by highlighting it with the middle mouse button, or using the Zoom dialog box.
All the same zoom functions are available with the Motif version of the window. With the Motif version, click on the scroll bar arrows to move one time tick, and on the slider groove to move one window width. Use the slider to move in larger increments. Use the middle mouse button and drag to zoom in on an area of the waveform display.
The Dataflow window allows you to trace signals throughout your design.
The Dataflow window shows either:

- A signal in the center of the window with all the processes that drive the signal on the left, and all the processes that read the signal on the right side of the window, or
- A process with all the signals read by the process shown as inputs on the left of the window, and all the signals driven by the process on the right side of the window.

8. List the top-level signals, by choosing the List >
Signals in region menu item from the Signals window.

9. Add top-level signals to the Wave window, by choosing the
Wave >
Signals in region menu item from the Signals window.

10. Apply stimulus to the clock input by moving the pointer to the QuickHDL window and entering the following command at the prompt: force clk 1 50, 0 100 -repeat 100
    Because the force command is a simulation command, it is case-insensitive. It is interpreted by the simulator to mean:

    - Force clk to the value 1 at 50 ns after the current time
    - Then to 0 at 100 ns after the current time
    - Repeat this cycle every 100 ns

    You will see the effects of this force command as soon as you tell the simulator to run.
    Next you will exercise different Run functions. They are found under the Run menu button (OpenWindows) or Run in the Motif menu bar.

11. Choose the Run > 100 ns menu item. This causes the simulatinto run and then stop after 100 ns.

12. Choose Run > Time'High. This causes the simulator to run forever. To stop the run, go to the next step.

13. Click Break menu button to stop the simulation.
    The arrow in the Source window points to the next VHDL statement to be executed. Next, you will set a breakpoint in the function on line 18.

14. Move the pointer to the Source window. Using the vertical scrollbar, scroll until line 18 is visible. Click at or near line number 18 to set the breakpoint.
    You should see a dot next to the line number where the breakpoint is set. This breakpoint can be toggled on and off by clicking it.

15. Choose Continue in the Run menu in the main window to resume the run that you interrupted.
    The simulator will hit the breakpoint, as shown by an arrow in the Source window and by a message in the QuickHDL window. Also note that the parameters and variables within the function are displayed in the Variables window.

16. Click Step to single-step through the simulation.
    Notice that the values change in the Variables window. You can keep clicking Step if you wish.

17. In the Wave window, use cursors to:

    - Probe for values: signal values update whenever you move the cursor.
      You can click at a time or transition in the Wave window and the cursor will snap to that position.
    - Find the time that a signal changes by clicking on the signal.
      The cursor will snap to the edge.
    - Measure time intervals. Add a second cursor and position the two cursors to view the time between them.
      Experiment with scrolling, zooming, and using the cursors until you are satisfied.

18. Then, quit the simulator by entering the command:
    quit -force
    This command exits the simulator without saving data.

2 Lesson Two

In Lesson Two, you will actually debug something! The goals of this lesson are:

- Show an example of a VHDL testbench-a VHDL architecture that instantiates the VHDL design units to be tested, provides simulation stimuli, and checks the results.
- Map a logical library name to an actual library.
- Change the default run length.
- Recognize assertion messages in the command window.
- Change the assertion break level.
- Restart the simulation run using the Restart command.
- Examine composite types displayed in the Variables window.
- Change the value of a variable.

- Use a strobe to trigger lines in the List window.
- Change the radix of signals displayed in the List window.

This lesson uses a VHDL package that you will compile into a different library.

1. Set your working directory to the tutorial directory (qhdl_t) that you created.
2. To create the new library, enter the following command at the shell prompt:
   /cad/mentor/bin/qhlib library_2
3. Compile the package by entering the next command at the shell prompt:
   /cad/mentor/bin/qvhcom -work library_2 stdlogic.vhd

   The other VHDL files expect the stdlogic package to be in a library having the logical name IEEE. To set this up, you can edit the [Library] section of the quickhdl.ini file, or you can create a logical library name with the qhmap command. (Normally, you use the precompiled IEEE library that is supplied with QuickHDL Pro.)
   For example:
   /cad/mentor/bin/qhmap ieee library_2
4. Compile the other files by entering the following sequence of commands:
   /cad/mentor/bin/qvhcom gates.vhd
   /cad/mentor/bin/qvhcom adder.vhd
   /cad/mentor/bin/qvhcom testadder.vhd
   The command could also list all the files on one line:
   /cad/mentor/bin/qvhcom gates.vhd adder.vhd testadder.vhd
5. Start the simulator by entering the following command at the shell prompt:
   /cad/mentor/bin/qhsim
   Whenever you invoke the simulator without specifying a design, the Startup window is displayed.
6. Within the Startup window, check that the simulator resolution is ns.

7. Within the Startup window, look in the Entity, Configuration, or Module scrollbox and select the design unit named Config test_adder_structural.
   This design unit is a configuration, so no architectures are displayed in the architecture scrollbox at the bottom of the window.
8. Within the Startup window, click Load to accept the settings.
9. To open all the simulation windows (except the Schematic window), enter the following command in the main window at the prompt:
   view *
10. To list the top-level signals, enter the command:
    list /*
11. To add top-level signals to the Wave window, enter the command:
    wave /*
12. To change the length of the default simulation run in the QuickHDL window, click on the Prop... menu button.
    This brings up the Properties window.
13. Move the pointer to the Default Run field and enter 1000. Do not change the other settings.
14. Click Apply (OK in Motif) to accept the new settings and then remove the dialog box.
    Next, you will run the simulator.
15. In the QuickHDL window, choose the Run > 1000 ns menu item. A message in the QuickHDL window will notify you that there was an assertion error. Lets find out whats wrong. To track down the assertion message, perform the steps listed below.
16. In the QuickHDL window, again choose the Prop > Options menu item.
17. Change the selection for Break on Assertion to Error and click Apply. Then dismiss the Properties dialog box
    This will cause the simulator to stop at the VHDL statement after the assertion is displayed.

18. To restart the simulation, in the QuickHDL window, select File and then click Restart design. Then press Restart on the Restart dialog box.
19. In the QuickHDL window, choose the Run > 1000 ns menu item. Notice that the arrow in the Source window is pointing to the statement after the assertion (assert false).
    In the Variables window (under "Loop"), note that i = 6. The simulation stopped in the sixth iteration of the test pattern's loop. Resize the Variables window so that you can see everything.
20. Expand the variable named test_patterns by clicking the [+].
21. Also expand the sixth record in the array, that is, test_patterns(6), by clicking the [+].
    The assertion shows that the signal sum does not equal the sum field in test_patterns(6). Note that the sum of the inputs a, b, and cin should be equal to the output sum. But there is an error in the test vectors. To correct this error, you need to restart the simulation and modify the initial value of the test vectors.
22. In the main window, type:
    restart -f
    The -f option forces the simulator to restart without displaying the confirmation dialog box.
23. In the Variables window, repeat the instructions in step 21 to expand test_pattern(6). Then highlight the sum record by clicking on the variable name (sum, not the box before it), and then click the Change button.
24. Select the last four bits in the value field 1000 by dragging the pointer across them. Replace them with 0111.
25. Click Change.
26. Select Run and 1000 ns. At this point, the simulation will run without any errors.

Next, you will learn how to change the triggering for the List window. By default, a new line is displayed in the List window for each time unit at which a listed signal changes. The following steps will change this so that values are only listed every 100 ns.

27. In the List window, select Triggers (OpenWindows) or Properties and then Triggers (Motif). This displays the Trigger dialog window.
28. To disable triggering on signals, deselect Trigger on: Signals.
29. To enable the strobe, select Trigger on: Strobe.
30. Enter 100 in the Strobe period field.
31. Enter 70 in the First strobe at field.
32. Click the appropriate button to tell the simulator you accept the settings.
    Notice that the list window contents changes to only show data for times that are increments of 100 ending in "70" (for example 870).
    Your next action will be to change the radix for a, b, and sum to decimal.
33. In the List window, select Signals (OpenWindows) or Properties and then Signals (Motif). This brings up the List Signals dialog.
34. Select signal a in the "Signals:" area of the List Signals dialog box by clicking on the signal label a. Then click Decimal in the Radix field and Apply the dialog box.
    The radix of the signal values displayed in the List window changes to decimal values.
35. Select signal b and change its radix to Decimal as in the previous step.
36. Select signal sum and change its radix to Decimal.
    The values a, b, and sum displayed in the List windows should all be displayed in decimal radix.
    This brings you to the end of Lesson Two, but feel free to experiment further with the menu system.
37. When you are ready to end the simulation session, quit the simulator without saving data by entering the following command in the main window:
    quit -force

3 Lesson Three

The goals of Lesson Three are to:

- Run a batch simulation
- Execute a macro file
- View a saved simulation

1. Set your working directory to the tutorial directory.
2. For this lesson, you will use a macro file that provides stimulus for the counter. Make sure that the file "stim.do" exists in your tutorial directory.
   If not, you need to copy this macro file from the installation directory by entering the command:
   cp /cad/mentor/shared/pkgs/quickhdl/examples/stim.do .
3. Create a new batch file named "batch1" using a text editor, and enter the following on separate lines in this file:
   list *
   do stim.do
   write list counter.lst
   Save this file (as batch1).
4. To run the batch simulation, enter the following command:
   /cad/mentor/bin/qhsim -wav saved.wav counter < batch1
   NOTE: You will get the message produced by the HDL code:
   !!! Error: Counting to 100 failed. Count is 01100100
   This command invokes the simulator on a design unit called counter that was included in the examples directory. The -wav option instructs the simulator to save the simulation results to log file saved.wav. The batch file you created specified that values are to be listed in decimal; it also executed a stimulus file containing stimuli and wrote the results to a file named counter.lst.
5. Since you saved the simulation results in saved.wav, you can view the simulation results by starting QHSIM with its -view option.
   /cad/mentor/bin/qhsim -view saved.wav

6. When the QuickHDL window is displayed, open the windows indicated in the following command (or select the View menu with the right mouse button and open them individually):
   view list wave
   Notice that you cannot open the Process or Variables windows. You are looking at a saved simulation, not examining one interactively. The logfile saved in saved.wav was used to reconstruct the current windows.
7. Now that you have the windows open, put signals in them:
   wave *
   list *
8. Use the available QuickHDL windows to experiment with the saved simulation results. Did the count actually reach 100 (change the radix in the List window to Decimal).
9. Quit when you are ready.
   quit -f

4 Lesson Four

The goals of Lesson Four are to:

- Specify the design unit to be simulated on the command line.
- Edit the quickhdl.ini file.
- Execute commands at startup.

1. Set your working directory to the tutorial directory.
2. In this lesson, you will use the existing "startup.do" macro file
   that provides
   startup information. If this file does not exist, you need to copy this macro file from the installation directory by entering the command (note the period specifying the current directory):
   cp /cad/mentor/shared/pkgs/quickhdl/examples/startup.do .
3. Next, edit the system initialization file in the tutorial directory and uncomment the following line in the [vsim] section of the quickhdl.ini file:
   Startup = do startup.do

4. Take a look at the “startup.do” file. It uses the predefined variable $entity to do different things at startup for different designs.
5. Start the simulator and specify the top-level design unit to be simulated by issuing the following command at the shell prompt:
   /cad/mentor/bin/qhsim counter
   Notice that the simulator loads the design unit without displaying the Startup window (as in Step 7 on page 1-9). This is handy if you are simulating the same design unit over and over. Also notice that all the windows are open. This is because the view * command is included in the macro that you executed at startup.
6. Quit the simulator by entering the following command at the prompt in the main window:
   quit -f

5 Lesson Five

The goals of Lesson Five are to:

- Use qvhcom to compile VHDL components and qvlcom to compile the Verilog components that are contained in the same design.
- Run the QuickHDL simulator on a VHDL design that contains the compiled Verilog and VHDL components.
- Use the Structure window to view the comparative hierarchies of these files.
- Use the Signals and Wave windows to list some signals.
- Run a simulation and observe the results as displayed waveforms.
- Quit the simulator.

1. 1.Set your working directory to the tutorial directory (qhdl_t).
2. Verify that the “mixedHDL” directory exists. If not, copy it into your current directory by entering the following command (note the period specifying the current directory):
   cp /cad/mentor/shared/pkgs/quickhdl/examples/mixedHDL/* .

3. Now set your working directory to the “mixedHDL” directory.
4. Run the compile script, located in the current directory:
   compile
   This script executes the qvhcom and qvlcom commands, which compile the VHDL and Verilog components of the design. The script also uses the qhlib command to search for a subdirectory called work and creates one if none is found.
5. Start the simulator on the “top.vhd” by entering the command:
   /cad/mentor/bin/qhsim top
   As before, the Main QuickHDL window comes up, from which you can call any or all of the other windows.
6. Bring up the Structure window by entering the following line at the prompt in the QuickHDL window:
   view structure
   The Structure window displays the structural hierarchy in your design. The VHDL or Verilog primary design units (see Chapter 2) appear in the Source window. Collapsed levels of hierarchy are indicated by a box containing a plus (+) for VHDL, a circle containing a plus for Verilog. Click on one to expand that level. When no lower levels exist, the box is empty. A minus (-) in a box or circle indicates an item that has been expanded.
7. Bring up the Signals and Wave windows:
   view signals wave
8. In the Signals window, choose the List > Signals in region menu item.
9. In the Signals window, choose the Wave > Signals in region menu item.
10. Move the pointer back to the main QuickHDL window. Enter the following command to run a simulation for 400 ns on this design and then stop:
    run 400

11. In the Wave window, expand and contract signal hierarchy in the signal list on the left. Also add cursors on the waveforms to examine signal deltas. Move these cursors to different transitions.
12. Quit the simulator by entering the command:
    quit -f
    This command exits the simulator without saving data.