ECE 426

CADENCE DESIGN SYSTEM
Tutorial

rev. 1/96

SUMMARY OF ENVIRONMENT

For this tutorial, students will be using the Logic Work Bench (LWB), which is a part of the Cadence Design System. LWB is a digital design and simulation environment that lets you use multiple simulation engines in the same design. By allowing multiple simulation engines, the designer can start with a behavioral description of a circuit (say in Verilog) and then move piece by piece to a schematic layout of the circuit. For this tutorial, the students will only be using LWB to create schematics and then move to the simulation stage.

The Logic Work Bench design environment is an integrated package of several related design tools. These tools each fill a different niche in the design process and interact closely with one another to simplify the many steps in creating and testing designs. The tools the student will be using consist of:

A. Setup
B. Concept
C. Hierarchy Manager
D. Valid Compiler
E. OpenSim
F. WaveEdit
G. WaveView

LWB graphically shows the tools needed for each step in the design process. To help understand the following tutorial, the circuit design process for LWB goes as follows:

1. The Setup tool creates a design directory and allows the user to set the design parameters for the different tools.

2. The Schematic Editor (Concept) helps the user to create an initial circuit design.

3. Since the circuit could consist of several layers (a CPU schematic contains a counter which contains several registers which is made up of a DFF and an AND gate), the Hierarchy Manager provides a way of moving through the different levels of a design to make edits. The Hierarchy Manager allows the user to select the simulation engine for each part of the design.

4. VALID compiles the schematic into a form understandable by the various simulators. VALID is never invoked directly in the LWB design process, but several of the tools (Hierarchy Manager and OpenSim) invoke the VALID compiler to verify the validity of a design and to compile it into format used by the various simulators.

5. WaveEdit creates a set of simulation directives. The simulation directives include the initial state of the circuit and the waveforms for the various input signals.

6. The simulation is started using OpenSim.

7. WaveView displays the outputs or state transitions within the circuit as the simulation runs.

Additional Documentation:
The Cadence Systems' tools all come with online documentation. Throughout this tutorial, there will be pointers for more information inside the OpenBook documentation.

SYSTEM SETUP

Currently there are binaries available for Solaris and SUN OS 4.1.3. You need to set your path to the binaries of the operating system you plan to use. Both are available in the riggs SUN lab.

For the Solaris operating system.
Add the following line to your .cshrc file:

```
set path=(/apps/ece/cadence/tools/bin $path)
setenv CDS_INST_DIR /apps/ece/cadence
```

Note: .cshrc is a hidden file in your home directory used to configure your login session. Enter "ls -a" to see all hidden files.

If you are wan to run Cadence from a SUN OS 4.1.3 machine, you need to add the following lines to your .cshrc file.

```
set path=(/apps/ece/cadence2/tools/bin $path)
setenv CDS_INST_DIR /apps/ece/cadence
```

TOOL SUMMARY

CONCEPT

Overview

Concept is a drawing package similar to AutoCAD which can be used to create logic drawings (schematics) and body drawings (shapes or symbols representing parts or functional design blocks). In addition to creating and modifying drawings, Concept organizes and maintains a design database, allowing users to incorporate old designs as functional blocks in newer designs. The Concept package also interacts closely with the other Cadence products allowing for a smooth transition from the design stage to the simulation and testing stage of the design process.

Place in Design Process

Concept will be the first stage in the design process. Concept forces the designer to use a hierarchial design process. Simple circuits can be created and stored in the design database. After these circuits have been tested, the designer can associate a body drawing with these simple circuits which can then be used to create more complex circuits. This tutorial will use a small library of pregenerated parts to build some simple circuits.

To Find Out More

System Design Solutions -> Logic Work Bench -> Concept Schematic User Guide .

HIERARCHY MANAGER

Overview

The Hierarchy Manager (HM) lets you work with a design in the context of the design schematic hierarchy. You can use the HM to

- Browse the schematic hierarchy
- Trace signals throughout the design
- Find components
- Create design configurations with different simulation models

You can use the Hierarchy Manager to work interactively with the graphical representation of your design hierarchy and an editor. For example, you can use the Hierarchy Manager to quickly locate a drawing or component in your design, then use the Concept schematic editor to edit the drawing.

Place in Design Process

The HM will typically be used after the first cut of a design is completed. The user will need to review a component included in their design or edit a section of the design deeper in the design hierarchy. Once the edits to the design are completed and the user is ready to simulate their design, the HM is used to create a simulation configuration file.

To Find Out More

System Design Solutions -> Logic Work Bench -> Hierarchy Manager User Guide

WAVE EDIT

Overview

WaveEdit is a Simulation Control Language (SCL) generator. This tool can be used to select input signals from a Concept drawing to include in the simulation. The selected signals are displayed on a simulation wave form display. For each input signal selected, the user can graphically specify where on the timeline a state change will occur. The user can specify a single period for a signal and then mark it as a clock signal and WaveEdit will ensure the clock behavior of the signal. When the user indicates a state change, they are given a list of all possible states the signal can assume. For a single wire, the options include 0, 1, undefined, and high impedance states. For a bus, the user enters the signal state in hexadecimal and the system automatically determines which wires on the bus need to be to 1's and which need to be set to 0's.

Once the user has fully specified input signals for their simulation, a script for controlling the simulation needs to be created. The simulation script is written in SCL (Simulation Control Language), and is used by OpenSim to initialize a circuit and to set points where a state change occurs. WaveEdit automatically creates this SCL script file based on the signal diagrams created by the user. The user just needs to specify the filename for the script file.

Place in Design Process

Once the drawing is complete and the simulation engine has been configured, WaveEdit will need to be invoked. With WaveEdit, the user will create the SCL file which will control the simulation. After this file is created, WaveEdit can be closed. The user will only need to restart WaveEdit if they need to change the input signal behavior for their simulation.

To Find Out More

System Design Solutions -> Logic Work Bench -> OpenWaves Reference -> Using WaveEdit

OPENSIM

Overview

OpenSim is the multiple simulation engine provided by Cadence. This tool allows the user to simulate designs with different simulation engines. For this tutorial, the RapidSim simulation engine will be the only one used. OpenSim compiles a design using the VALID compiler into a format that can be used by the RapidSim simulator. The user can check for any design errors or configuration problems at this point. The user then loads timing information and the SCL script file created by WaveEdit. Then the user issues the SCL step command to step through the simulation and watch the results on the WaveView tool. If the user is familiar with the SCL commands, they can modify the simulation behavior at any point in the simulation (However, I recommend that the user create another SCL script file using WaveEdit, then rerun the simulation).

Place in Design Process

This tool is the last step in the design process. It combines all of the output from the other tools into the final simulation. The user will mostly interact with the VALID compiler through OpenSim, then review any errors. Users will then make any fixes in Concept and recompile through OpenSim. Once this is done, the user's only other major interaction with OpenSim is stepping through the simulation. Between OpenSim and WaveEdit, all of the details of setting up and running a simulation are hidden.

To Find Out More

System Design Solutions -> Logic Work Bench -> OpenSim Simulation Control Reference

Read at least first 5 pages of
System Design Solutions -> Logic Work Bench -> RapidSim BP Reference Manual

WAVE VIEW

Overview

WaveView is tool closely related to WaveEdit. The screen layout and command options are almost exactly the same as WaveEdit. The main difference is that while WaveEdit is used to create input signal wave forms for the simulation, WaveView is used to display signal states versus time as the simulation runs. WaveView can be used to watch any wire or bus states throughout the simulation. At the end of the simulation, WaveView can save the simulation results to file or print the results to a postscript file or printer.

Place in Design Process

After OpenSim has been started and the simulation has been initialized with the SCL script file created by WaveEdit, the user will select wires and buses to monitor from the Concept Schematic. These selections will automatically be displayed on the WaveView screen. As the user steps through the simulation, the selected signals will be displayed and any state changes will be shown. At the end of the simulation, the user will save and/or print the results.

To Find Out More

System Design Solutions -> Logic Work Bench -> OpenWaves Reference -> Using WaveView

DESIGN EXAMPLE

Setting up a Cadence working directory

1. Create a directory to hold all of your schematics and simulation results.
   md my_proj_dir (does not matter what name is used)

2. Go to this directory.
   cd my_proj_dir

NOTE: It is important that you are in the project directory whenever you start up LWB.

3. Start the Logic Work Bench.
   lwb

4. It will ask you if you want to configure this new directory. Say yes (option 1).
   Configure it with the "top down" choice (option 1).
   Then quit the setup menu (option q)

5. You now need to edit master.local file that was created.
   Type the following lines into master.local:

```
file_type = master_library;
'sim'           '/apps/ece/cadence/lib/sim/sim.lib';
'standard'      '/apps/ece/cadence/lib/standard/standard.lib';
'wlsttl'        '/apps/ece/cadence/lib/wlsttl/lsttl.lib';
end.
```

   Save the file.

Note: The directories for these libraries keep changing as ECO reorganizes it servers. Check with ECO to discover if these directories have been changed and edit the file accordingly.

Setting up the Tools

1. Start Logic Work Bench by typing the "lwb" command "inside" the newly created project directory.

LWB creates the following files automatically:

framework.env@	lwb_td.log	pman.pro@	voices.log
lwb.log	pman.log	tdformserv.log	

2. Once LWB is started, you will see the graphical representation of the design process. The flow that this tutorial uses goes as follows:

Setup:	to setup all of the tools for our use
Sch Edit:	starts Concept Schematic Editor
Explore:	starts Hierarchy Manager
WaveEdit:	starts the Wave Edit Tool
SimControl:	starts OpenSim Simulation Control
Wave View:	starts the Wave View Tool

3. Select the "Setup" button on the LWB design flow (left click). You will be prompted to enter a design name:

enter: project <RETURN>

4. Select the "Config" button. This is used to set the default simulator to be used with any designs. Many of the predefined components that will be used in this tutorial are defined here.

enter: _sim <RETURN>

This sets RapidSim as the default simulator. Other settings are:

_sim	(for RapidSim simulations)
_backplane	(for Verilog simulations)

5. Select the "Global" button. A dialog should appear with 4 main options:

Master Library
Library
Root Drawing
Use

You will click on the box beside each one in turn and fill out the following dialog boxes:

Master Library: <click on the blank line in the box>

enter: master.local <click on DONE>

Library:

click on <Add> three times, there are now 4 entry blanks

Fill out each blank with the following text "in the order" listed

sim

wlsttl

standard

<click on DONE>

The library entries specify the default libraries to search for schematic or simulation building blocks (parts). If a schematic indicates a part named 3AND (a 3 input AND gate), then the system first searches the sim library, then the wlsttl library and so on until it finds a part with the given name. The "standard" library is automatically available, but by putting it at the end of this list, it will be searched last for component definitions when compiling the schematic.

Root Drawing:

enter: project <Use **this** word specifically! It will be explained later>

The compiler needs a root drawing to be specified if it is to successfully compile any other designs in the project directory.

Use:

enter: userid.wrk (userid should be login name)

<click on DONE>

From the main Global Dialog box,

<click on APPLY>

<click on QUIT>

From these operations, the global.cmd file will be created.

6. Select the "Compiler" button. This setup form will be used to create a default configuration file for the Valid Compiler. This file is necessary for the compiler to run. The Hierarchy Manager and the OpenSim simulator use the Valid Compiler.

Make sure "logic" is the current compile type

Click on <APPLY> and then <QUIT> as in step 5.

7. Select the “Concept” button. This form defines the parameters used in your drawing.

Using the scrollbar on the right, scroll down until you see “Plotter Type”.

Click in the box and replace “versatek” with “ps”. Now your printouts should go to the Riggs printer.

Click on <APPLY> and then <QUIT> as in step 6.

NOTE: Step 8 is for Verilog simulation. Skip this step for RAPIDSIM simulation.

8. Select the “Verilog” button. A pop-up allows you to select their formatter or simulator, select “simulator”, another popup gives several choices - select “VLOG1”. A form appears for setting the Verilog simulator runtime options.

Click the box marked “Upper Case”

Click the box marked “Library Paths”. A pop-up appears - use it to enter the following path: “/apps/ece/veriloglib/verilogTTL/74LSTTL”

Now exit the Setup application.

Running Concept

Now Concept will be used to create a simple 8-bit register from the tutorial component library selected earlier.

Note: Anytime the words **select**, **place**, or **click** are used, then the user should click the left mouse button to perform the specified action.

1. Start Concept from the LWB flow chart by selecting the SchEdit icon. The Concept screen should start with a blank View Area labeled "DWG: PROJECT.LOGIC.1.1" on the info bar below the View Area.

2. This drawing is considered the root drawing of a design and the Valid compiler will always start with this drawing and then go to the currently active schematic. Therefore, we need to save this blank drawing so the compiler can find it.

    enter: write

You will prompted to save <userid.wrk>PROJECT.LOGIC.1.1 Enter: "y"
The "write" command does the following:

    Creates a new subdirectory in the project directory with the name "project"
        (if it does not already exist) to act as the design directory
    Saves all files for the current drawing into the design directory.
    Creates the file "userid.wrk" in the current dir (if it did not already exist) which Concept
        uses to track all of the drawings a designer has in the project directory.
        Note: the file should now have an entry for "PROJECT" in it.

3. Now lets create our first drawing. It is to be a one bit register. Later we will save this drawing, and use it to create an eight bit register. Start editing a new drawing called “reg1” with the following command (type in the command area below the View Area):

    enter: edit reg1

A listbox will popup:

    select: logic (should be at the top of the list)
    click on <OK>

This tells Concept that the new drawing “reg1” will use **logic** (AND gates, OR gates, etc).
Notice the DWG label in the Info Bar changes to: DWG: REG1.LOGIC.1.1

Now save this new drawing:

    enter: write <RETURN>

Again, this will create a REG directory and place another entry in the userid.wrk file.

The write command followed by a drawing name has the following effects:

write new_name <RETURN>

Creates a new design directory with the new name

Copies the current drawing into the new directory

Displays the newly created drawing

Note: This is good for maintaining different versions of the same drawing or copying another design to make changes without affecting the original.

4. Starting the library browser. To easily access libraries of parts and basic circuit building blocks, Concept provides a library Browser window. This window can be used to list the parts in any number of libraries or work files (filename.wrk) From this window, the user can edit a specific part or add the part to the current drawing. So lets start the Library Browser:

Click on the <Add Part> button, **or** enter: dir<return> on the command line.

A single column list box should appear with "userid.wrk" as the list label.

REG1 and PROJECT should appear in the list box.

4. Adding libraries to the browser. More than one library can be opened at a time; they will simply be cascaded out to the right.

Click on "Commands" button at the top of the list box.

Click on "Add Library"

Another column holding the userid.wrk listing should appear.

Click on the list label "userid.wrk" .

A list of available libraries (the ones specified earlier) appears in the listing.

Select "wlsttl" from the listing.

A list box will appear with ttl parts listed in it.

Note the 3-buttons at the base of the list box for the first column:

Add

Edit

PPT (will not be used)

The "Add" button indicates that every part selected from any of the libraries in the Browser are to be added to the current drawing.

The "Edit" button indicates that any part selected from the Browser is to become the current drawing, and immediately available for editing.

5. Before we begin to build the 1 and 8 bit registers, there are a few useful functions to point out:

Copy	(Big Button)	Note that the buttons
Delete	(Big Button)	are on the left side of
Move	(Big Button)	the window for these.
Undo	(Small Button)	
Redo	(Small Button)	

These functions are used by clicking on the respective button then clicking left mouse button on the component or wire that needs to be moved, copied, or deleted. For the move and copy command, clicking the left-mouse button again will place the part in the current mouse location.

Undo will cause the schematic to revert to the drawing before the last command was issued. This button can be used to back several commands. Redo undoes undo.

6. Lets set up the page we will be using, and explore how to see multiple choices for an element.
   Open a third library called "standard", just like you opened the wlsttl library. See step 4.
   Click on "Add" button in the Library Browser
   Click on A SIZE PAGE in the Component Browser
   As you move the mouse back to the Concept window, a standard page border will appear.
      Clicking the **middle** mouse button will show four possible directions.
      Many of the elements you use will have more than one rotation. It is a good idea to click through the possible choices to determine which one fits best.
      Note that two of the borders are "portrait" style, the other two are "landscape"
      You should **always** put a border around your drawings for 426 assignments.
   Pick one of the two portrait styles, and left-click on it while Title, Date, Engineer, & Page are visible.
   To see the whole page border at one time, hold the left mouse button down and draw a letter "W" over the part of the border that is visible. Concept has a built-in
process
      called **strokes** that will actually recognize these drawn alphabet characters. The letter "W" is the shortcut stroke for "zoom fit". The entire border should be
visible
      Note the section below that discusses other **strokes**.

7. Now let's build a 1-bit Register.

Click on LS74 in the Component Browser (DFF). Try the middle mouse button for direction.

Also try the **right** mouse button. Some elements can have different representations.

Place the DFF on the screen in the center of the border. Now is a good time to play with the Copy, Delete, and Move buttons. Make a few copies, move them around, get familiar with how the buttons work. Making a "Z" stroke will zoom in to a box that is the exact size of the Z drawn, see **zoom** section. Try drawing a "Z" over the DFF that's about twice its size. Refer to the **strokes** section.

Click on LS08 (2AND) and place it to the left of the DFF (see diagram below)

Click on LS04 (INVERTER) and place it below and to the left of the 2AND gate

8. Lets draw some wires to connect the components together. There are two ways to do this:

The **wire** and **route** commands are subtly different in their use.

Click on the **wire** button.

left button	selects a pin
left again	sets a corner for the wire
middle button	changes the way the wire is routed (try it!)
right button	ends a wire (system tries to connect wire to nearest pin, so if the wire is not going to be connected to anything, place a corner where you want the wire to end, (left button) then click on the right button.

Place a wire from the AND output to the clock pin on the DFF (the triangle) using wire.

Click on the **route** button.

left button	selects a pin
left again	ends a wire in space.
right button	ends a wire and performs the routing to the nearest pin.

Place a wire from the INV output to the Clear pin on the DFF (on the bottom) using route.

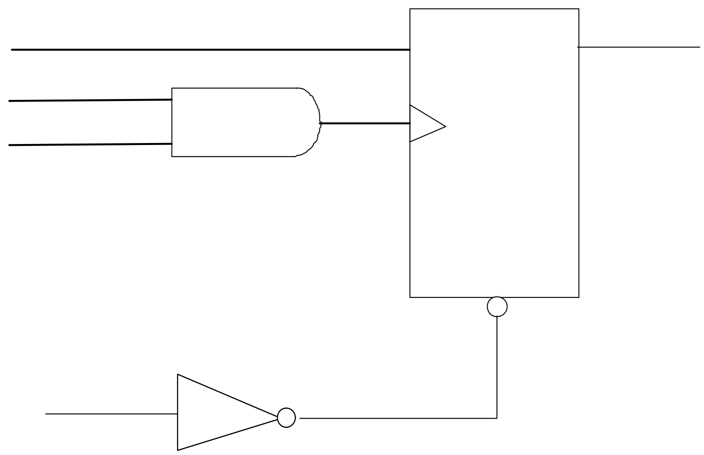
Add wires to the 2 inputs of the AND gate using route or wire.

Add a wire to the single input for the inverter.

Draw a wire to the D pin and draw a wire from the Q pin.

Try moving the AND gate. Note what happens to the connected wires. Try the DFF.

The schematic should look something like the following:



9. Adding signal names to the important wires.

All wires can have the SIGNAME attribute. The SIGNAME is used to connect wires without having to actually draw the connection. Two wires that have the same SIGNAME will both see the exact same signal during asimulation. Also, SIGNAME attribute can be used to indicate a signal vector (a bus).

Click on SIGNAME

Click on the D wire and on the command line, enter D<7..0>\I <RETURN>

Click on first AND input and on command line, enter: ENABLE\I

Click on second AND input and on command line, enter: CLK\I

Click on INV input and on command line, enter: RESET\I

Click on Q-output for DFF and on command line, enter: Q<7..0>\I

Click on semi-colon icon to exit Signal Naming mode.

Add SIZE attribute to DFF.

1) on command line, enter: attribute <RETURN>
2) click on the middle of the DFF

A form will appear showing all current attributes for that drawing element.‘

3) click on the ADD button on the Attribute Form
4) under the Name entry space, type SIZE in all caps
5) under value entry space, type 8
6) click on DONE button

An 8 will appear above the DFF to show this attribute has been added

Change 1-bit DFF to an 8-bit DFF

This shortcut can be used to change the size of any device that uses the SIZE attribute. In this way, an 8-bit DFF could be created without having to layout 8 separate bit registers connected to the same clock.

Every element can have 0 or more attributes associated with them. However, the element must be designed to take advantage of a given attribute. Most of the predefined parts will take advantage of the SIZE attribute. For most elements, the SIZE attribute is used to change the inputs to a device into a bus. For a DFF, setting the size attribute = 8 means that there is an 8-bit bus at the D-input and an 8-bit bus at the Q-output. For a 2 input AND gate, a SIZE = 2 will result in there being a 2-bit bus on each of the 2 inputs and a 2bit bus on the output. Always add a signal name to any wire you wish to monitor or control while simulating.

Save drawing, enter: write <RETURN> and type "yes" when prompted

The first Concept Schematic design is completed.

Zoom features

Click on the ZOOM button in the first vertical button bar

Click on the upper left-hand area near the components

Hold the button down and move the mouse until the box contains all of the components placed on screen and click the left button again.

Type the following commands and observe the View Area:

zoom fit	zoom in	zoom out
zoom .5	zoom 1.5	zoom left
zoom right	zoom up	zoom down

The zoom “directions" can be duplicated by clicking on "Zoom" then typing arrow keys

Note: zoom value — where x.xx is the amount of zoom
value > 1.0 then zoom in, value < 1.0 then zoom out

Stroke features

The strokes are quick and easy shortcuts that are performed by holding down the left mouse button and “drawing out” the desired alphabet character.

- “W” drawn in the window will cause a “zoom fit” to be executed.
- “Z” drawn over a specific area in the window causes a “zoom in” that will cover the exact area specified by the drawn letter. It is a good idea to draw this letter bigger than the area you wish to zoom into.
- “m” drawn over an object will enable a move.
- “/\” drawn like an upside-down v will delete the nearest object.
- “V” drawn over an object will allow you to change its version or representation.
- “c” drawn will copy the nearest object.
- “R” will put you in route mode
- “S” will select an object

To see more about the strokes command, read /apps/ece/tools/fet/concept/README.strokes.

*Printing in Concept***:**

From the command area enter the following two commands:

set pl ps
ha

"set pl ps" is short for set plotter to postscript printer.
"ha" is short for hardcopy - the Cadence command for printing.

Running the Hierarchy Manager

Now we will use the Hierarchial Manager to move through the new design and setup the design to be simulated.

1. Click on the Explore icon in the LWB Design Flow

2. For this simple tutorial, all that needs to be done is to load the design and default simulation configuration into the HM.

    a. on the command line enter: design reg1
    b. enter: load _sim
    c. enter: update (will have to wait a few seconds)

3. The HM can communicate directly with Concept and control which drawings are displayed. This allows the user to select any part displayed in the HM and popup the correct schematic in Concept. To do this:

    a. click on the "Edit" button in the vertical button bar
    b. click on one of the PART.SIM.1.1 components that make up the REG1.LOGIC hierarchy.
    c. Switch to the Concept View - the simulation part definition for the selected component is displayed. Zoom in on the diagram in the center of the View Area and the DELAY and SIZE attributes for the part are visible.
    d. Switch back to HM and click on REG1 - the Concept View area will return to the Register schematic.

If the _backplane configuration had been specified instead of _sim, then editing the component parts would have popped up a text editor with the appropriate Verilog file loaded.

Note the (In Context) notice beside the design name. This indicates that the HM and Concept are sharing data about the current design. The (In Context) notice should appear on all of the tools that are started - to indicate they are all dealing with the same drawing.

Running WaveEdit

Now that the drawing is complete and the simulation engine has been configured, the simulation script for controlling the simulation needs to be created. The simulation script is written in SCL (Simulation Control Language), and is used to initialize a circuit and drive the circuit inputs. The SCL can quickly be created with WaveEdit.

<u>Add Signals to Wave Edit's Signal Graph</u>
1. Click on the Wave Edit icon from the LWB flow.

For the following process to work, OpenSim **must** be running:
2. Click on the "OpenSim" icon from the LWB flow.
3. Click on the "Add" button (2nd button from the top). On the command line, "Name of signal to add>" appears.
4. Switch to Concept but make sure the command line area on WaveEdit is visible.
5. From Concept, click on all of the signals defined with the SIGNAME property in the following order:

CLK
D<7..0>
ENABLE
RESET
Q<7..0>

After clicking on "Add", the user can just type the signal names on the prompt if they wish.

6. Switch to WaveEdit
7. Click on the <CR> button at the bottom of the button bar to leave the Add Signal mode.
8. Notice that the selected signals are listed to the left of the View Area in the top left corner of WaveEdit.

<u>Draw a Clock Signal</u>
1. Now select the "Draw" button.
2. Click on the CLK signal. A popup box appears with the buttons for :

0: low binary state
1: high binary state
U: undefined state
Z: high impedance (open circuit) state

3. Click on 0, then the "Done" button - wait for the box to reappear
4. Click on 1, then the "Done" - the box disappears and the user should place where the state change will occur on the Signal Timing Graph. Click on the first dotted line (.2 usec).

5. The popup box will return. Click on 0, then the "Done" button and place the state change at the next dotted line (.4 usecs).
6. When popup box returns, click on Cancel.
7. Click on the <CR> button to exit Draw mode.
8. Click on the Clock button, then click on the CLK signal.
9. In the command area, the user is prompted if the given period is acceptable, click on the <CR> button. Wave Edit automatically generates a Clock signal for the single period defined. The user could have redefined the clock period simply by typing a new one at the prompt.

Panning
1. In WaveEdit, the arrow keys cause the Signal Timing Graph to Pan left, right, up or down. There are menu buttons which accomplish the same thing.

Draw Typical Wire Signal:
1. Click on "Draw" and select ENABLE. Define the ENABLE signal to start at 0 and switch to 1 at the .8 usec point.
2. Now draw the RESET signal starting at 0, switch to 1 at 1.5 usecs then set to 0 at 2.0 usecs. May need to PAN LEFT or PAN RIGHT to view the .

Draw a Bus Signal:
1. Note that the system is still in the Draw state so click on D<7..0>. A keypad appears so the user can enter the signal state in Hex. The two hex digits are shown at the bottom of the keypad as are the bits that are defined by a hex digit. The reset button can be used to enter a new number.
2. Enter 0F in hex (15 in decimal) on the keypad and click on Done. Then enter 04 and click on Done.
3. Place the state change at the 1.2 usec point. Notice that Wave Edit prints the Hex values for bus signals on the graph.

Saving SCL:
Now that all of the signals have been defined, save the signal graph as an SCL file:
1. Click on "File" -> "Write" -> "Stimulus" .
2. Enter the filename: reg1_sim.scl

Exiting:
The user may exit by clicking on "File" -> "Exit" or leave WaveEdit up to modify the simulation behavior after the first run.

Running OpenSim

Now that all of the input to the simulator is ready, a simple simulation can be ran.

Setting up the Simulation

1. Click on "SimControl" from the LWB flow to start OpenSim.

At the top are several labels:

Resolution: displays how large a single simulation step is (currently 1 ns). Note that WaveEdit displayed signals on a microsecond scale and 1 usec = 1000 nsec.
Sim Time: how long has the simulation been running
Design: current design being simulated
Config: simulation configuration file being used to control which simulators are used (default RapidSim file is being used)

2. Need to Compile the schematic into a format which can be simulated (ie. run the Valid Compiler):

click on: "Load" -> "Load_Design"

This runs the Valid Compiler and then RapidSim is called by OpenSim. This step may take a few minutes.

3. Always check for errors: if a Fatal Error occurs in the Valid Compiler, OpenSim does not precisely state the problem. Therefore the user must run another tool to get an error report from the Valid Compiler. This listing states all fatal errors that occurred in the latest compile.

a. switch to the cmdtool which you used to log into a Cadence machine.
b. Change to the project directory (do not cd into the subdirectories for the designs).
c. type: comperr
d. when comperr completes, it will have created a file called "cmplst.dat"
e. view this file and hope that you can figureout the problem from there.

4. Once the Compiler and RapidSim complete successfully, the user must load the component timing information.

click on: "Load" -> "Load_Timing"

5. Start WaveView by clicking on the WaveView icon on the LWB design flow. See the Running WaveView page for more information about the operations that can be performed using this tool.

6. Start the simulation:

click on: "Start/Stop" -> "StartSim"

7. Next need to load the Simulation Control Language script file created by WaveEdit into OpenSim. This file sets the initial state, sets which signals to monitor in waveview, and sets when input signals change state.

click on: "Others" -> "Script"

At the Script prompt in the command area:

type (with the quotes): "reg1_sim.scl"

Obviously, any script file name could have been used.

Stepping Through the Simulation

The command area in the OpenSim window has the prompt SCL> .
This means that the user can type any SCL command at this prompt and have OpenSim execute it. Most of the SCL commands are stored in the script file created by WaveEdit. However, the user will progress through the simulation with the "Step" command. The command has the format:

s xxx

Where xxx is the number of steps in units of the Simulation Resolution specified in the Info Area at the top of the OpenSim window. Therefore, with a resolution of 1 nanosecond,

s 1000

advances the simulation 1 microsecond.

With the WaveView signal graph visible, enter the following series of step commands at the SCL prompt on OpenSim:

s 200
s 200
s 400
s 200
s 500
s 1000

Now you have completed simulating the interesting points in the Simulation with WaveEdit.

The user could now go back and edit the SCL file. Reload the file as specified above. Restart the simulation. Finally, step through the simulation to see the new results.

Restarting the Simulation

Click on: "Start/Stop" -> "StopSim"

This should reset the simulation time to 0.