14:332:479 Concepts in VLSI Design
16:332:574 CAD Digital VLSI Design
Instructions for Using the Cadence System

M. L. Bushnell and Tezaswi Raja

December 18, 2005

1 Help

In order to use the Cadence system, you must copy the .login and .cshrc files from Bushnell's directory (~ bushnell) into your top directory. If you do not do this, Cadence will not work at all for you. You can get the entire on-line manual set from Cadence at any time by typing **cdsdoc** (this takes quite a while to load). Some of the manuals tell you to type **openbook**, which no longer works. Please copy over these files from ~ bushnell:

- .login
- .cshrc
- .cdsenv
- .cdsinit
- cds.lib

These files set up the various libraries that you will be using. For your designs, please use parts ONLY from the *Rutgers_Digital_Parts* and *Rutgers_Analog_Parts* libraries.

2 Selecting the Proper Cadence Framework

There are a number of commands to activate the Cadence tools.

- If you type:
  icde &
  This activates the design environment, in which the library browser and the Schematic Composer are available. The layout editor is not available, and the best simulation tools are not available. This mode of operation is not recommended.

- If you type:
  icms &
  This activates the mixed-signal design environment, in which the library browser, the schematic composer, the Spectre analog simulator, and the Verilog-XL logic/switch-level simulator are available. Use this command to enter schematics and perform logic and analog simulations.

- If you type:
  layoutPlus &
  This activates the layout design environment, in which the library browser, the schematic composer, and the virtuoso layout editor are available. Use this command for your layout editing activities, and for running the parasitic circuit extractor.

- If you type:
  icfb &
  This activates all of the layout and mixed-signal tools.

- If you type:
  seultra &
  This activates all of the Silicon Ensemble (standard cell place-and-route) tools.

- If you type:
  sesi &
  This activates all of the Silicon Ensemble (transistor place-and-route) tools.

3 Schematic Composer and Virtuoso Layout Editor

There are two excellent tutorials in the Cadence manual set on these two tools. The manuals are called *Cadence Virtuoso Schematic Composer Tutorial* and *Cadence Cell Design Tutorial*. Please use them to learn the tools. There are signal conventions that must be followed for power and ground in both of these tools. For Schematic Composer, both at the logic and at the switch levels, all V_{DD} signals should be named and labeled **VDD!**, and all GND signals should be named and labeled **GND!**, for the logic simulator to work well. For Virtuoso, all V_{DD} signals should be labeled **VDD** (not **VDD!**), and all GND signals should be labeled **GND** (not **GND!**), for Spectre analog simulation to work. If you do not follow this rule, then the Spectre netlister will abort on your circuit. This means that during analog simulation, **VDD** and **GND** will show up in Spectre as normal voltage sources, rather than global voltage sources.

4 Logic/Switch-Level Simulation

The Cadence Schematic Composer, Virtuoso Layout Editor, and Verilog-XL simulator are outstanding products, and reliable nearly all of the time. However, the interface between Schematic Composer and Verilog-XL is a disgraceful mess of software engineering, and riddled with bugs. We apologize for the difficulty of using this software. This memo gives instructions that makes it possible for you to use it.

1. *Schematic Parts.* Only use logic gates and transistors from the *Rutgers_Digital_Parts* library, as these have the necessary Verilog views for successful simulation.

2. *Design of Schematic* (switch-level simulation only). In Schematic Composer, when you enter a switch-level schematic, assign *VDD* the net name *VDD!*. Assign *VSS* the net name *VSS!*. You may have as many of these nets as you like.

3. When the schematic is done, use the *check and save* command on it. Fix every floating or disconnected net. Fix all nets with conflicting names. If you have two wires crossing and connected by a single solder dot, Composer will complain. This is o.k. if you really meant to connect the wires. Otherwise, delete the solder dot.

4. Use the *Tools-Simulation-Verilog-XL* menu selection.

5. If the run directory for simulation proposed in the form is o.k., then click *OK*. Otherwise, type in a different run directory and click *OK*.

6. In the *Setup-Netlist* menu of Verilog-XL, set the Netlisting mode to *Incremental* if that is not already set, and delete the *Symbol* view from the types of netlists to include in the Verilog run deck. Then, click on *OK*.

7. In the *Setup* menu of Verilog-XL, select the *Record Signals* menu item. In the form that pops up, select *Design Selections*. This is the only option that seems to work. Then, click on your Schematic Composer window containing the schematic. Select the first signal pin that you want to appear in the simulation output. For all additional pins that you want to appear in simulation output, shift-click

on them. If you make a mistake in selecting signals, then click on empty space and start the selection process over.

8. Once you have selected all pins, click on the window for the form for signal selections. Then, click *OK* on the form.

9. In the *Stimulus* menu in the simulation window, click on *Verilog*. In the resulting form, tell the system to create a new Verilog stimulus file for you. Select *testfixture.verilog* as the stimulus file.

10. If at any any point in this process, the system complains that no design is selected for Verilog-XL simulation, go to the highest-level *Command Interpreter Window* (CIW). Select *Tools-Verilog-Integration-Verilog-XL*. Select the run directory proposed by the interface. That will usually fix the design selection problem.

11. Realize that nearly all of the buttons in the Verilog simulation window will not work for you. The only interesting ones are *Start Interactive*, *Finish Interactive*, *Continue*, and *View Waveforms*. Click on the *Start Interactive* button.

12. The interface will *netlist* your design, which means that it will convert the design from the graphical schematic composer layout into the textual Verilog hardware description language. Look in the Command Interpreter Window and see if there are any errors from netlisting. In particular, check that a simulation view (*functional*, *schematic*, *cmos_sch*, *behavioral*) has been included in the table for each cell that you instantiated in your design. There should also be an entry for your entire design. If this is not happening, get help from Bushnell.

13. After netlisting has succeeded, the CIW should have a line indicating that the system has successfully gathered the probes for the signals that you want to look at. If this did not happen, see Bushnell.

14. Next, in the Verilog-XL window, Verilog will compile the Verilog files for your simulation, your probes, and your testfixture stimulus into binary code to link into the simulator. Look for any errors in the compilation and see Bushnell if you do not know how to fix them.

15. Once compilation succeeded, then click on *Continue* to finish the simulation. You have now simulated your design, but with the default test fixture. This is not useful, so we will now edit the *testfixture.verilog* file in the simulation run directory with your favorite text editor. The file must be changed to look like this:

```
// Verilog stimulus file.
// Please do not create a module in this file.

// Default Verilog stimulus.

parameter period = 1000;

initial
begin
$dumpfile("mysigfile.vcd");
$dumpvars;

   A = 1'b0;
   B = 1'b0;
   C = 1'b0;
   #1000000 $finish;
end

  always #(period/2) A = !A;
  always #(period) B = !B;
  always #(period*2) C = !C;
```

Lines beginning with // are comments. The line

```
parameter period = 1000;
```

sets the clock period to 1000 nanosec. Do not change this. There are four processes in this file, which are very similar to VHDL processes. The *initial* process starts at the beginning of simulation and executes only once. The three *always* processes execute all of the time. A process can either be a single statement or a *begin-end* block containing many statements. The *$dumpfile("mysigfile.vcd");* statement is critical and causes all simulation waveforms to be saved in the *mysigfile.vcd* file. Without this, you will get nothing from Verilog-XL. The *$dumpvars;* statement is also critical – without it, you will not see simulation results from any signal. Regardless of what the manuals say or what the Verilog-XL interface implies, these two statements in the *testfixture.verilog* file are the ONLY way to get the simulator to display useful information to you. The three signal assignments to *A*, *B*, and *C* in the initial process initialize all three signals to binary 0 (the last 0 in the expression). You should change the signal names and signal values to whatever you want your signals initialized to. You must initialize all of your input signals (but you do not need to initialize *VDD!* and *VSS!* for switch-level simulation). The statement *#1000000 $finish;* delays the initial process by 1000000 nanosec. and then terminates the simulation. Without this, you will not get meaningful simulation output. The statement *always #(period/2) A = !A;* complements signal A after a delay of period/2 and runs continuously. The statement *always #(period) B = !B;* complements signal B after a delay of period and runs continuously. The statement *always #(period*2) C = !C;* complements signal *C* after a delay of period * 2 and runs continuously. This is by far the most convenient way to cover all signal combinations. Please change the signal names to the names used in your design, and adjust the clock periods for signal complementation to obtain the waveforms that you want. Save the *testfixture.verilog* file, and also write it into a backup file *testfixture.verilogsave*. This system loves to destroy your *testfixture.verilog* file whenever it netlists your design, so you must keep a backup copy of this file.

16. Switch-level simulation only. There is an incredibly stupid bug in this interface that affects switch-level simulation. Whenever netlisting occurs, the system destroys the power supply setting for *VSS*, and you must restore it. In the run directory for simulation, edit the file *hdlFilesDir/cds_globals.v* and make sure that the following declarations appear:

    ```
    supply1 VDD_;
    supply0 VSS_;
    ```

    Save the file, as it will be destroyed every time your design is netlisted. The netlister changes the *VSS_* declaration to *wire VSS_;*, which causes the *VSS* bus to float in the high-impedance state. This leads to incorrect simulation. Similar errors occur for the power supplies *PWR* and *GND*, and you can correct these in the same way by editing the *cds_globals.v* file.

17. Now that you have corrected the *VSS* power supply and the testfixture, we are ready for the real simulation. Click in the Verilog-XL window on the *Start Interactive* button again. However, when it asks you in a form whether you want the design netlisted again, just say *NO*. Otherwise, the interface will again destroy the *testfixture.verilog* and the *cds_globals.v* files.

18. Once the Verilog files have been compiled, click on *Continue Simulation*. After that completes, check that you got more than 0 simulation events or more than 0 accelerated simulation events. If you did not, then get help from Bushnell.

19. Click on the *View Waveforms* button to run Signalscan. A window should pop up and you can see your simulation results. You can also print this graphical display. Signalscan works just fine. However, occasionally it will be unable to find the waveforms, because the main design was not selected. This appears to be a random bug. You can open the file that you saved your waveforms in (as specified in the *testfixture.verilog* file), and read the waveforms into Signalscan. It will then work correctly. Here is the procedure from Signalscan:

(a) Click on the button *Open-Simulation-File*. In the list of files in the directory, double-click on the simulation runtime directory that you are using. The directory is added to the location of the waveform file. Click to the right of the directory in the entry blank, and type in *mysigfile.vcd*, which is where you told Verilog-XL to save your simulation results.

(b) A window will pop up, and the system will ask if you wish to convert the files. Check on the *Include Sequence Time Information* and then the file conversion will proceed.

(c) Back in your main Signalscan menu, click on *Des.Brows:1*.

(d) A window will open up. In the *Instances in Current Context*, click on your waveform file that you just read in. For the instance, select *test*.

(e) All simulation variables will display in the bottom part of the window. Select these in the order in which you want them displayed, and they will appear in the window to your left.

(f) When you have all of the variables that you want, click on *Control-Add Variable and Close Window*. This will add all of the timing waveforms to your main timing window. After this, you can proceed as if Signalscan correctly read in your waveforms.

Remember that all other buttons in the Verilog-XL simulator interface window have not been shown to be useful, and may actually cause problems. Do not use them.

5 Using the Cadence Spectre Analog Simulator with the Schematic Composer

It appears that the Spectre analog simulation environment is significantly nicer than the Cadence digital simulation environment.

1. *Schematic Parts*. Only use analog devices and transistors from the *Rutgers_Analog_Parts* library, as these have the necessary models for successful simulation. Note that several of us have had repeated problems doing the analog simulation with the *nmos* and *pmos* parts. For reliability in your simulation, use parts *nmos4* and *pmos4*, instead, because these always work. The fourth electrode is the bulk terminal, which must be correctly set for correct analog simulation.

2. Enter your design by typing *icms &* and using Schematic Composer to create the analog circuit with parts from the *Rutgers_Analog_Parts* library.

3. When you want to simulate, click on *Tools – Analog Environment*. Click on *Setup – Design*. Select the design name to simulate, and then click on *OK*.

4. Click on *Setup – Simulator/Directory/Host*. Select only *spectre* (not *spectreS*). Click on *OK*.

5. This step is needed only for the transistors in your analog circuit. Click on *Setup – Model Libraries*. Click on *Browse*. Go to the directory */caip/u21/cadence/local/models/spectre/nom/*. There you will find the various transistor models *tsmc25N* and *tsmc25P* for the various technology files. Select the first model that you need for simulation and click on *Apply*. Click on *Add* in the other form that pops up for this model. Continue this process of clicking on *Apply* and then *Add* until you have selected both of the transistor models that you need to use for your analog simulation. When you have everything, click on *OK* in the main form.

6. Click on *Setup – Stimuli*. In the form, setup your stimulus for the one analog input. If the design has multiple pins, it will prompt you for the pin to which you are applying each stimulus (a voltage source or a current source). Then click on *Global Sources*. Make sure that you have selected $2.5\,V$ for V_{DD}!. Otherwise, your simulation will be wrong. *GROUND!* is always set to $0\,V$.

7. Click on *Analyses – Choose*. Fill out the form for each analysis that you want and then click on *Apply*. Finally, click on *OK* when you have completed describing all of your analyses. If you want transient analysis, choose the *Liberal* mode unless you feel like waiting a few days for your analysis to be computed.

8. Click on *Outputs – To Be Plotted.* Click on *Select on Schematic.* Click on every *WIRE*, and not the *PIN*, of the circuit nodes that you want to be plotted. When you have selected everything that you want plotted, look in the simulation control box to verify that the simulator understood your selections. Then go on.

9. Click on *Simulation* and then on *Netlist and Run* (or only on *Run* if you have not changed the netlist since the last simulation). Ignore the form that pops up with the fine print and just click on *OK.*

This simulator does not like to stop simulating, even if you interrupt it and ask it to stop. You may have to kill it from another window with a *kill -9 <process id>* command.

6 Layout Generation from a Verilog/VHDL File via Synopsys Using Silicon Ensemble

Please follow these steps exactly.

1. Start Synopsys **design_analyzer**

2. Change target library in Defaults to stdcells.db

3. Change link library in Defaults to stdcells.db

4. Clear the symbol library blank

5. Read in Verilog/VHDL into Synopsys.

6. Optimize the hardware – this will translate the behavioral design into a logic-level netlist using the stdcells library (which is different from gtech)

7. Write out the logic-level netlist by changing the format to Verilog from db format, and changing the netlist name from whatever.db to whatever.v

8. Type **seultra &** to bring up the Silicon Ensemble tool.

9. Import the stdcells.lef file from /caip/u21/cadence/local/lib/stdcells.lef – use the *FILE – IMPORT – LEF – SELECTION* form. The tool will say that this will destroy the .rpt file, and that is fine. Check in the commentary window for the message *The database created successfully.*

10. Import the stdcells.v file from /caip/u21/cadence/local/lib/stdcells.v – use the *FILE – IMPORT – VERILOG – Verilog Source Files* form. Clear the Verilog Top Module and Verilog Reference Libraries blanks in the form. Enter a directory name for the *Compiled Verilog Output Library* blank in the form. DO NOT change the names of the **vdd!** and **gnd!**. Click OK on the form. The tool will say that this will destroy the .rpt file, and that is fine. Check in the commentary window for the message *End importing Verilog.*

11. Import the Verilog netlist produced by the Synopsys system. – use the *FILE – IMPORT – VERILOG – Verilog Source Files* form. Fill in the *Verilog Top Module* blank with the name of the top module entity in your Verilog design. Copy the *Compiled Verilog Output Library* blank contents in the form into the *Compiled Verilog Reference Libraries* blank. Click on *OK* in the *ADD/OVERWRITE* warning form (it does not overwrite, it will instead add the new module). Check in the commentary window for the message *End importing Verilog.*

12. Click on *FLOORPLAN – INITIALIZE FLOORPLAN.* Change *Left/Right IO To Core Distance* to 7.0 microns. Change *Top/Bottom IO To Core Distance* to 7.0 microns. Click on *Abut Rows.* Click on *OK.*

13. Click on *ROUTE – PLAN POWER – Add Rings.* Change *Core Ring Width for Horizontal metal1* to 0.9 microns. Change *Core Ring Width for Vertical metal2* to 0.9 microns. Click on *OK.*

14. If you get an error *Cannot create rings ... because of no space* go to *Floorplan – Resize Floorplan – Horizontal 100 %*. Repeat this for *Floorplan – Resize Floorplan – Vertical 100 %*. This increases your layout module area.

15. Click on *Place – IOs – OK* to place the inputs and outputs.

16. Click on *Place – Cells – OK* to place the required standard cells. If you get the message *Impossible to Place Cells without Overlaps*, then you need to resize the floor plan to increase the available area.

17. Use the *Floorplan – compact Floorplan – OK* command to compact the floor plan. This has been shown to be highly effective.

18. Click on *Route – Route Power – Follow Pins* changing the primary Metal 1 Width in the form to 0.90. Click *OK*.

19. Click on *Route – WRoute – OK* to finish the final detailed routing.

20. Click on *File – Export – DEF*. In the option form give the file name for the .def file.

21. Exit the tool from the *File* menu.

22. Run **icfb &**

23. In the *Files – Import – DEF* menu, fill the form with the name of your .def file. Specify also the existing library name where the layout cell for standard cell place-and-route will be created. Also specify a cell name for the new layout cell. Also, give *autoRouted* for the cell view name in the same form.

24. In the *Files – Open – Cell* menu, open the cell that you created, and a graphical display tool should come up and show you the layout.

25. In the main menu choose the *Tools – Layout* option to bring up the normal Virtuoso layout tool.

26. In the *Edit – Search* menu, change the *search for* blank to *inst in current cellView*. Click on *Add Criteria* and change the first blank to *view name == abstract*. Change the bottom line to *Replace view name ->layout*. Click on *Apply* and *Replace All*. You should now have a layout cell with your standard cell logic, such that you can place this cell in a manually-created design.

7 Layout Generation from a Transistor Schematic Using Virtuoso Layout XL

7.1 Introduction

This procedure lets you generate a layout of a schematic using the *Virtuoso XL* tool. For using this procedure you need

- A transistor schematic in Schematic Composer with the connections already made.

7.2 Setting Up Rules Files

Before you can successfully use *Virtuoso XL*, you need to establish a rules file in your account for the tool to use. The file should be in the directory in your account where you will be running *Virtuoso XL*. Please type this command to UNIX to copy the tsmc 0.25 micron rules files into your directory:

```
cp /caip/u21/cadence/local/lib/tsmc.25rules .
```

7.3 Procedure

7.3.1 Starting *Virtuoso XL* (VXL)

- Open the schematic and Check and Save it.
- Now in the tools bar click *Tools* -> *Design Synthesis* -> *Layout XL*
- Make sure the *Create New* button is checked in the pop up menu for layout generation. Another window pops up titled as the layout.

7.3.2 Layout Window

- In the layout window click on *Design* -> *Gen from Source.* The *Layout Generation Form* (LGF) appears.
- In the LGF deselect *Boundary* in generation options. Select a metal layer for every I/O pin from the cyclic field alongside the pins table. When all pins have layers selected, click *OK.*
- In the *Command Interpreter Window* (CIW) you get a warning that "Terminal B not found." This is true as we don't have a bulk terminal in our cell layout. You can safely ignore it.
- Now we need to create a boundary for the layout. Select the prBound Layer in the *Layer Select Window* (LSW) and in the menu *Create* -> *Rectangle.* Draw a rectangle covering all the cells in the layout.
- Now we place the cells within the boundary using *Edit* -> *Place from Schematic.* This puts the cells within the boundary.
- Now we need to place the pins. In the menu *Place* -> *Pin Placement.* Select the 'on left boundary edge in the cyclic field and select the pins you want in the left side of the table (multiple select with the ctrl key). Now click on the > arrow icon. This means that these pins will be placed on the left boundary. Now select another side in the cyclic field and repeat the procedure until there are no pins left in the 'Unplaced pins field.'

  Click on *OK* (you will see the pins placed on the boundary as you requested them.)
- Now click on *Connectivity* -> *Show Incomplete Nets.* Click on *Select All* and *OK.* You will see the unconnected sections of the routes in colored guide lines. These are the connections to be routed by the router.

  Now click on *Design* -> *Save.*

7.3.3 Starting the Router

- In the menu click on *Route* -> *Export to Router.* The *Export to Router* form appears.
- Fill in the Export Cellview with the name of your own layout.
- Check *Use Rules* file: Click on *Set File* and select the translation rules file.
- Click *Set directory* and select the directory in which you want the intermediate files and session files to be dumped. (Create just one named VXL and use it for every layout). Delete the files periodically to save space.
- Click on *Cadence Chip Assembly Router* in the *Routers* part of the form. Click on the *Start Router with Options* button.
- Click *OK.* This will start the router window.

7.3.4 Router Window

Power Rings Generation: In the command window enter these commands for selecting your power route.

- rule net vdd(width 0.9)
- rule net gnd(width 0.9)
- direction metal1 horizontal
- direction metal2 vertical
- proute ring(net vdd gnd(ordered))(primary_layer metal1 metal2)

This will generate the supply trunks for the nets V_{DD} and GND. Change the names and widths of the supply trunks according to your layout.

Global Routing: In the command window type:

- route 15; clean 10

  The syntax is route <no. of route attempts>; clean <no. of clean attempts> . The Router starts routing and takes a few seconds for a large circuit.

7.3.5 Saving and Importing Back to Layout

- Once the layout routing is complete we need to Import it back to the layout window. In the router window click on *File -> Write -> Session.* Save the *.ses* file and click *OK*. This should automatically start the Import from Router to Layout. If it doesn't, then use the *Route -> Import from Router* in the layout window to do the same.

By the end of this procedure you have a layout of the schematic with all routes performed.

7.4 Known Problems and Solutions

- *DRC errors after layout generation*: There are some cases where you get design rule errors after the layout. This is because the cells are placed too close to one another. So it is advisable to perform a DRC check before you export the layout to the router and remove the placement errors. In this case, increase the boundary area width and height to correct the overlap problem, since you have overconstrained the layout system by giving it too little area for placing the standard cells.
- Some nets are not routed: If the routing is not 100% successful, increase the routing attempts or if they are too small route them yourself to save time.

8 Layout Plotting Guidance

Many students have trouble plotting their layouts on the color printer c633gn, because they do not realize that a color layout plot has an enormous number of pixels for a large chip. Therefore, the disk files become huge, and they may cause the printing system (which is flaky anyway) to choke. So, use c633gn for small to moderate sized layouts, and use c633plotter (the large ink jet plotter) for large layouts.

This procedure will help you to plot a huge layout:

1. Cadence will plot a maximum of 30 pages at once. So, if you want to plot a larger layout than that, you must break it up into parts and plot each part separately.

2. First select the plotting scale that you want to use. For example, if you want to plot a 1 meter wide hard copy of a 1000 micron layout, the scale is 1000.

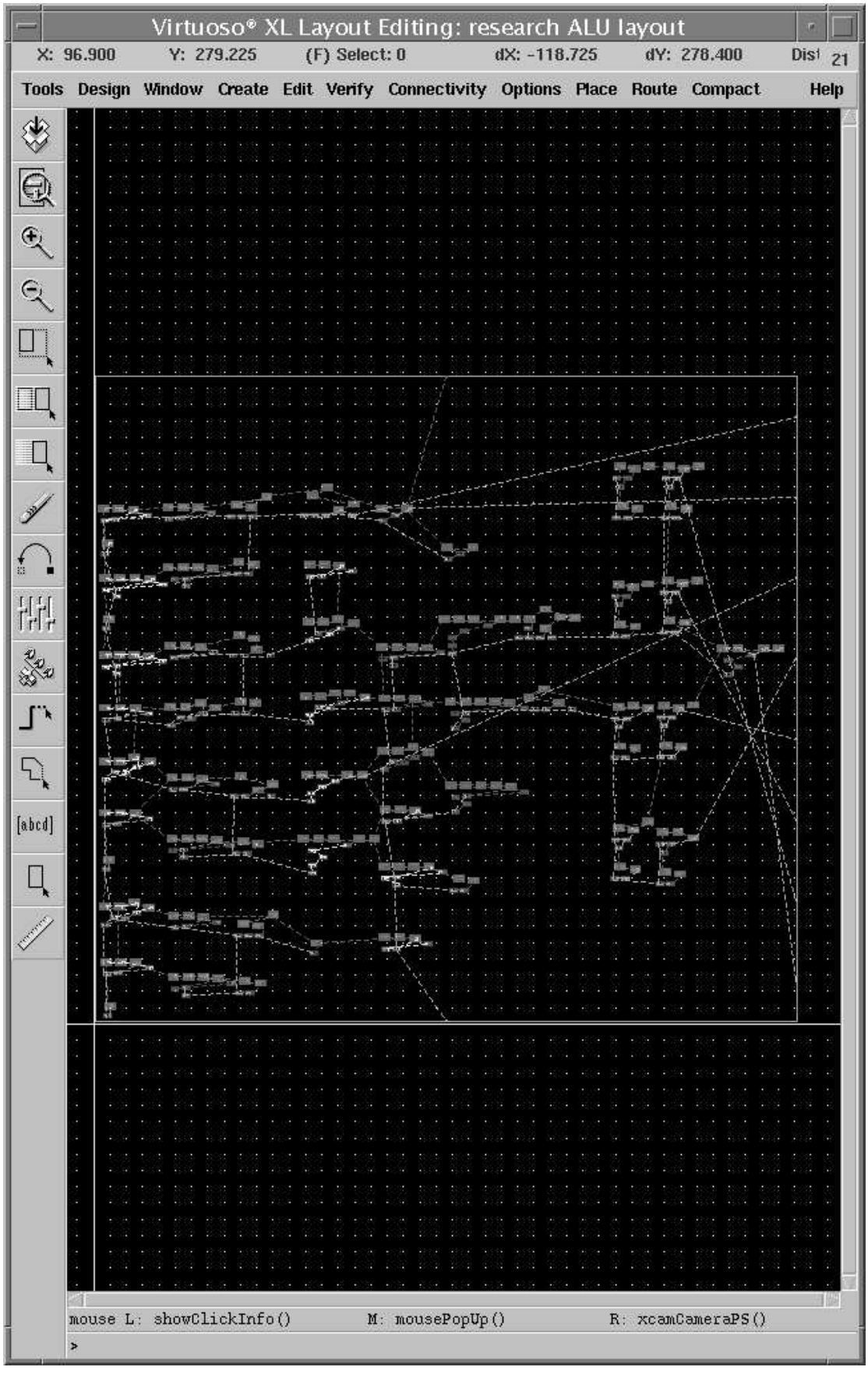
3. If the plot is more than 30 pages, then you need to plot more than once. Make sure that you use the same scale for each plot.

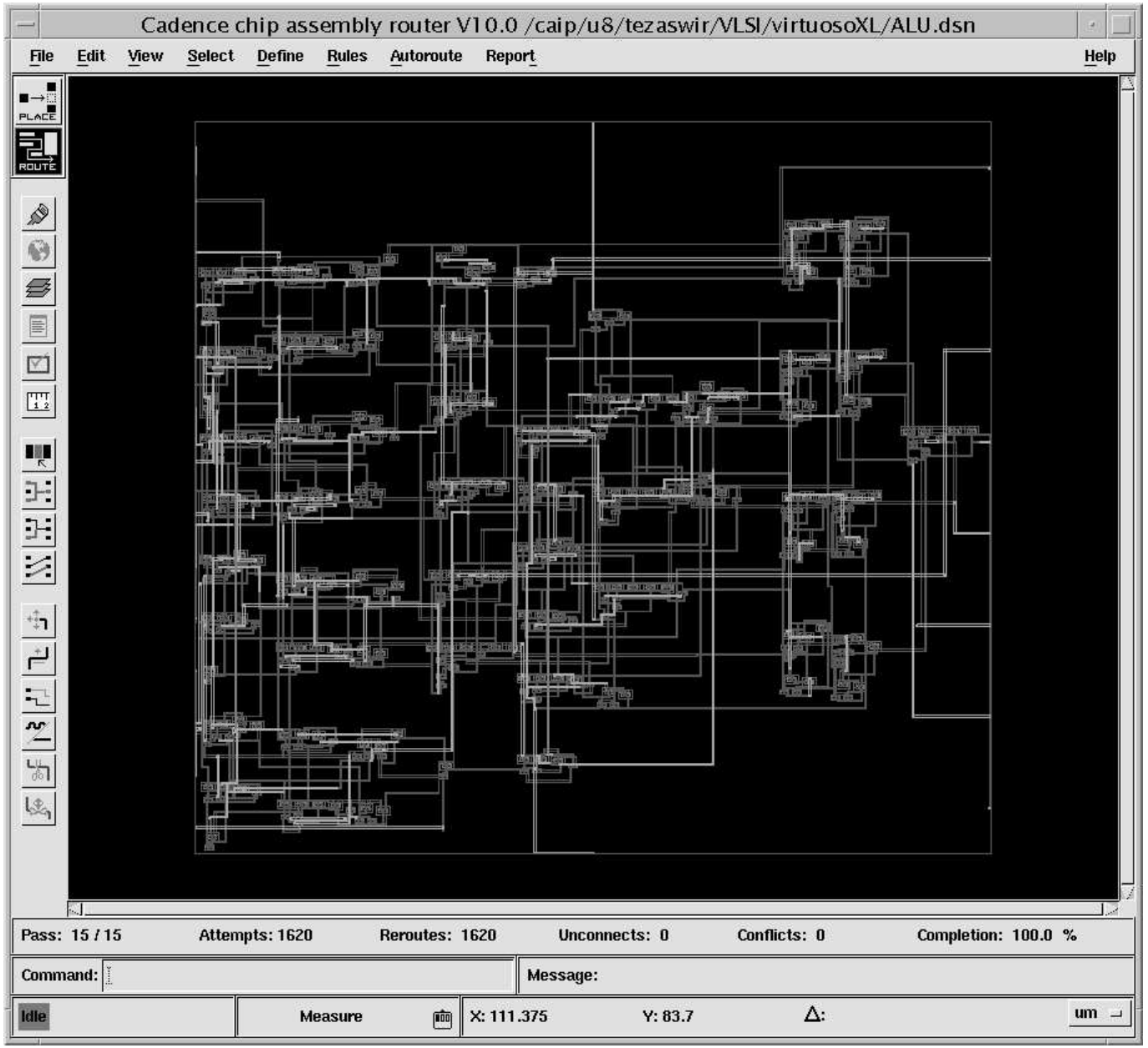
4. It is strongly recommended that you first plot your huge layout into a file. Select the "plot into a file" option in the plot dialog window. Give a name like abc.ps. Once you have abc.ps, use **lpr -Pc633gn abc.ps** to plot it.

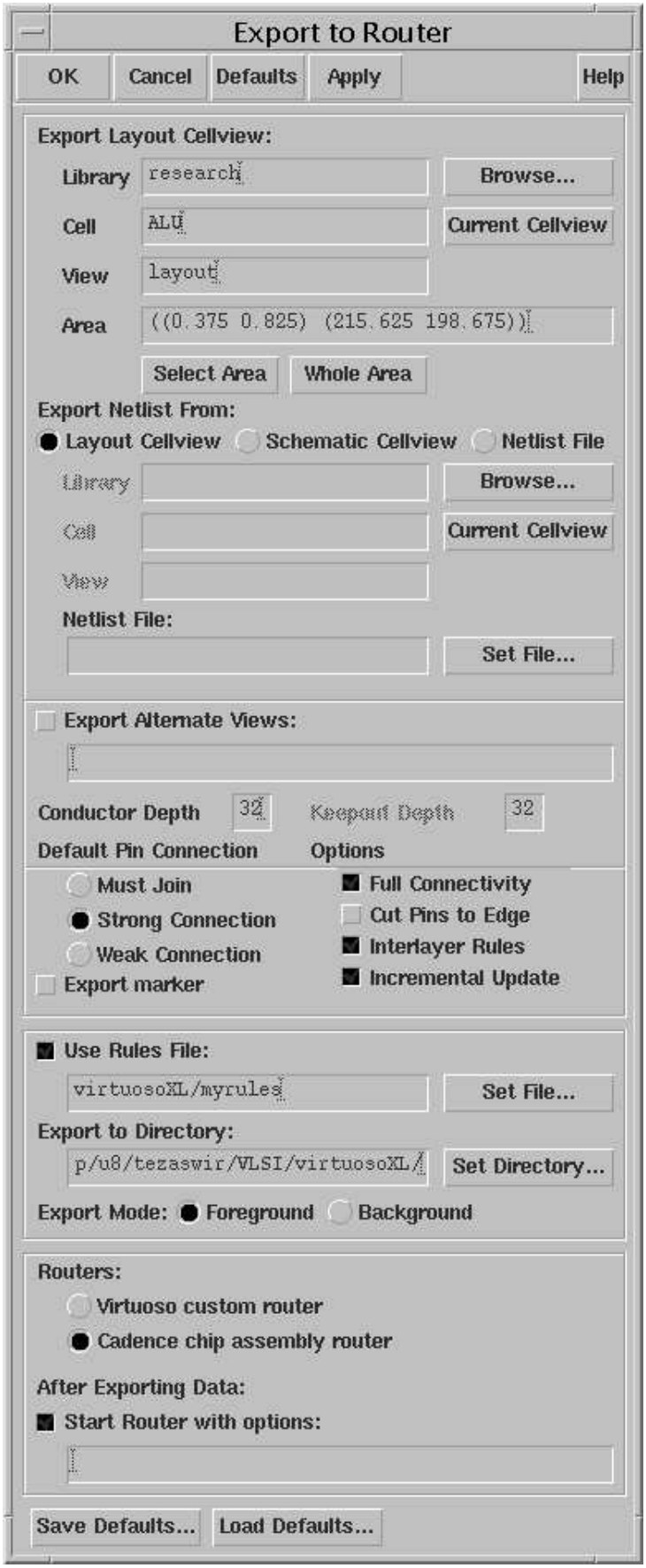
5. It saves a lot of time to plot a layout into a .ps file. It takes about 5 minutes to plot a .ps file of 100 MBytes, while it takes about 20 minutes or more to send the same layout directly to the c633gn printer. You cannot do anything but wait during this period, so write the plot to a postscript file.

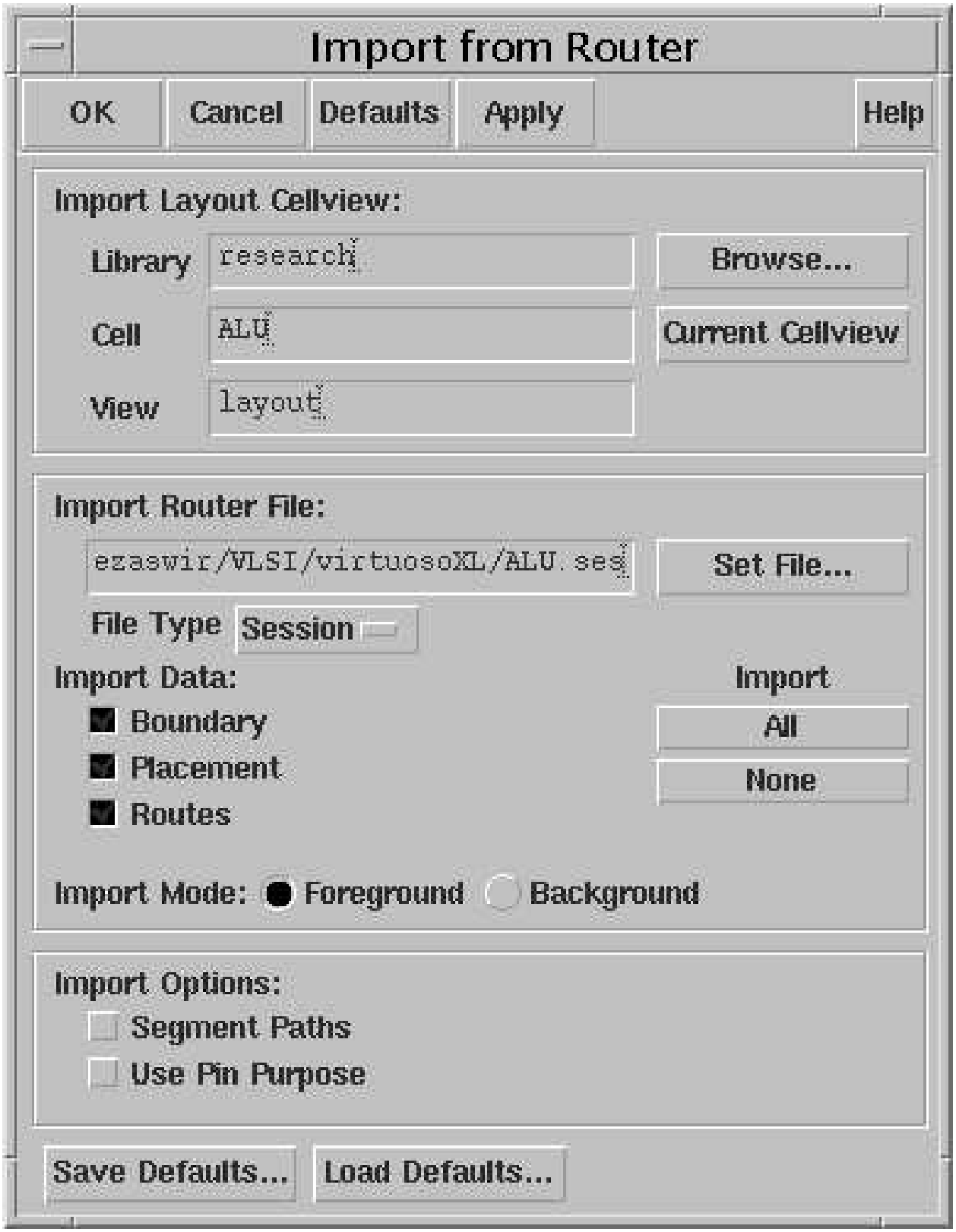
6. Please plot huge layouts overnight. Printer c633gn can process 1 MByte to 1.5 MByte per minute. It takes 11 hours to plot an 800 MByte .ps file. If you plot a 100 MByte file during the day, all jobs from other people behind you will be blocked for 1 to 2 hours!

7. Please arrange with the operations staff for a common directory for huge .ps files. Everyone can have write access to the directory. You can then plot huge .ps files into this directory without worrying about your own quota. The system administrator can clean this common directory periodically.


Virtuoso® XL Layout Editing: research ALU layout
X: 96.900
Y: 279.225
(F) Select: 0
dX: -118.725
dY: 278.400
Dist
21
Tools
Design
Window
Create
Edit
Verify
Connectivity
Options
Place
Route
Compact
Help
[abcd]
mouse L: showClickInfo()
M: mousePopUp()
R: xcamCameraPS()
>


Cadence chip assembly router V10.0 /caip/u8/tezaswir/VLSI/virtuosoXL/ALU.dsn
File
Edit
View
Select
Define
Rules
Autoroute
Report
Help
PLACE
ROUTE
Pass: 15 / 15
Attempts: 1620
Reroutes: 1620
Unconnects: 0
Conflicts: 0
Completion: 100.0 %
Command:
Message:
Idle
Measure
X: 111.375
Y: 83.7
Δ:
um


Export to Router
OK
Cancel
Defaults
Apply
Help
Export Layout Cellview:
Library
research
Browse...
Cell
ALU
Current Cellview
View
layout
Area
((0.375 0.825) (215.625 198.675))
Select Area
Whole Area
Export Netlist From:
Layout Cellview
Schematic Cellview
Netlist File
Library
Browse...
Cell
Current Cellview
View
Netlist File:
Set File...
Export Alternate Views:
Conductor Depth
32
Keepout Depth
32
Default Pin Connection
Options
Must Join
Strong Connection
Weak Connection
Export marker
Full Connectivity
Cut Pins to Edge
Interlayer Rules
Incremental Update
Use Rules File:
virtuosoXL/myrules
Set File...
Export to Directory:
p/u8/tezaswir/VLSI/virtuosoXL/
Set Directory...
Export Mode:
Foreground
Background
Routers:
Virtuoso custom router
Cadence chip assembly router
After Exporting Data:
Start Router with options:
Save Defaults...
Load Defaults...


Import from Router
OK
Cancel
Defaults
Apply
Help
Import Layout Cellview:
Library
research
Browse...
Cell
ALU
Current Cellview
View
layout
Import Router File:
ezaswir/VLSI/virtuosoXL/ALU.ses
Set File...
File Type
Session
Import Data:
Import
Boundary
All
Placement
None
Routes
Import Mode:
Foreground
Background
Import Options:
Segment Paths
Use Pin Purpose
Save Defaults...
Load Defaults...