Tutorial PnR: Place and Route

Created for the MSU VLSI program by Pete Semig, Fall 2004
Updated: S05, S06

Document Contents

Introduction

This document will provide students with the methodology for performing ***place and route*** (P&R) with the Synopsys and Cadence tools. We will use a standard cell library created by Illinois Institute of Technology that is compatible with the MOSIS AMI C5N process. Students will learn how to perform synthesis with Synopsys Design Vision (aka Design Compiler) and P&R with Cadence's Silicon Ensemble.

Environment Setup

The setup for this tutorial is extremely important. Please execute the following series of commands very carefully:

1. Open a terminal window and go to your class directory. For example,

```
     cd  /egr/courses/personal/ece410/<username>/
```

2. Create two folders called synopsys and se, and then go to the synopsys directory.

```
     mkdir  synopsys
     mkdir  se
     cd  synopsys
```

3. Copy some files you will need for this tutorial to your directory with the following command.

```
     cp  /egr/courses/personal/ece410/resources/pnr/*.*  .
```

Note: you must include the last '.', with a space between the last '*' and the last '.'.
This will copy *pnr_setup.tar* and *adder8.v* to your */synopsys* directory.

4. Expand the .tar file

```
     tar  xvf  pnr_setup.tar
```

This will create directories *WORK*, *db*, *lef*, *map*, *muxcase* and *v* under */synopsys*. In these six directories you will find all the files needed to perform place & route. From here on, references to these directories will only include the directory name, not the absolute path. For example, if you are requested to copy a file from the *db* directory, that is the same as the */egr/courses/personal/ece410/<username>/synopsys/db* directory.

5. Copy the standard cell library you will need for logic synthesis to your directory with the following commands.

```
   cd  /egr/courses/personal/ece410/resources/iit_stdcells/lib/ami05
   cp  -R  IIT_stdcells_ami05  /egr/courses/personal/ece410/<username>/cadence
```

This completes the environment setup.

Synthesis with Synopsys Design Vision

This tutorial will guide you through construction of a multiplexer based on provided VHDL code. The first step is to synthesize the code using the Design Compiler tool from Synopsys.

1. Move to the directory where the multiplexer files are stored.

```
cd  /egr/courses/personal/ece410/<username>/synopsys/muxcase
```

2. Type **source $SOFT/synopsys**

3. Type **design_vision &** to start the Design Vision tool. The window in Figure 1 will open.

4. To tell Design Vision that we will be importing a VHDL file, type the following command at the *design vision-t>* prompt:

```
set hdlin_enable_presto_for_vhdl true
```

Note: you should skip this step if a Verilog (the default) file is used.

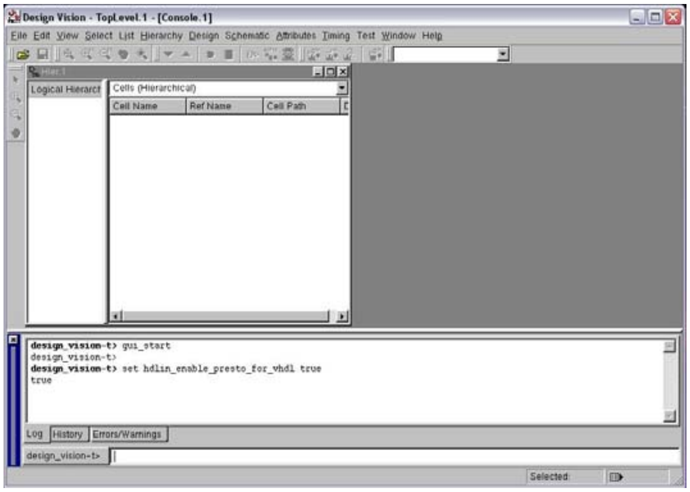
5. Select **File** >> **Setup**. The window in Figure 2 will open.

6. Delete the contents in the *Link library*, *Target library*, and *Symbol library* boxes.

7. Click on the ellipsis (...) next to *Link library*.

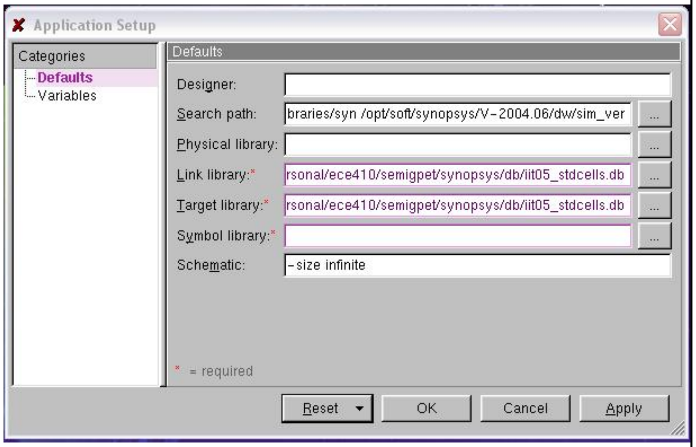
8. Select **Add** and locate the *iit05_stdcells.db* file from your *db* directory. Select **OK**.

9. Repeat steps 7-8 for *Target library*. Leave the *Symbol library* box blank. Select **OK** to close the setup window.


Design Vision - TopLevel.1 - [Console.1]
File Edit View Select List Hierarchy Design Schematic Attributes Timing Test Window Help
Hier.1
Logical Hierarch
Cells (Hierarchical)
Cell Name
Ref Name
Cell Path
Log
History
Errors/Warnings
Selected:

```
design_vision-t> gui_start
design_vision-t>
design_vision-t> set hdlin_enable_presto_for_vhdl true
true
```

Figure 1: Design Vision window with VHDL enabling command.



Designer:

Search path: braries/syn /opt/soft/synopsys/V-2004.06/dw/sim_ver

Physical library:

Link library:* rsonal/ece410/semigpet/synopsys/db/iit05_stdcells.db

Target library:* rsonal/ece410/semigpet/synopsys/db/iit05_stdcells.db

Symbol library:*

Schematic: -size infinite

* = required

Figure 2: Setup Link and Target Libraries.

10. In the *Design Vision* window, select **File >> Analyze**.

11. In the window that opens (Fig. 3), click **Add**. Specify the *muxcase.vhd* file from your *muxcase/source* directory.

12. Check the *Create new library if it does not exist* box. Click **OK** to close the Analyze Designs window. A window will pop up saying the library already exists. Click **OK**.

13. Make sure that the bottom of your *Design Vision* window reports successful compilation as shown in Figure 4.

14. In the *Design Vision* window, select **File** >> **Elaborate** to open the window in Figure 5.

15. Drop down the *Library* menu and select **WORK**.

16. Check *Reanalyze out-of-date libraries*. Click **OK**.

17. In the *Design Vision* window, select **MUXCASE** in the *Logical Hierarchy* sub-window (Fig. 6).

18. Select **Hierarchy** >> **Uniquify** >> **Hierarchy**

19. If you get an error message, click **Close**.

20. In the *Uniquify Hierarchy* window click **OK**. This window may be hidden behind other windows.

21. In the *Design Vision* window select **Design** >> **Compile Design**.

22. Select **OK** to compile the design and return to the *Design Vision* window.

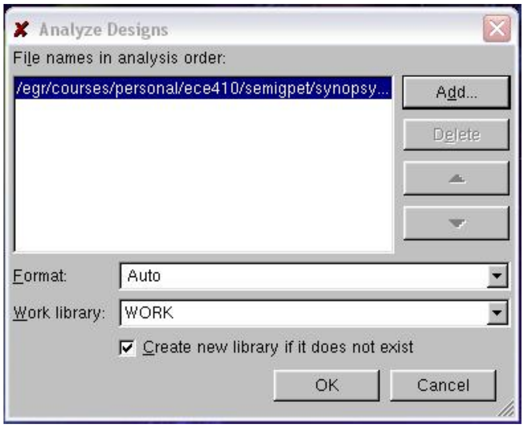
23. Select **Design** >> **Check Design**. Select **OK** to check the design.

24. Select **File** >> **Save As** and save the file as *muxsyn.v* in your *v* directory.

25. Re-save the file as *muxsyn.db* in your *db* directory. Be sure to select the **DB(db)** Format when saving the file.

26. To view the schematic that the Synopsys tools have created from the VHDL file, right-click on *MUXCASE* in the *Logical Hierarchy* sub-window and select **Schematic View**. This is not a necessary design step, but it is interesting to see.

27. Select **File >> Exit** to exit *Design Vision*.



Analyze Designs

File names in analysis order:

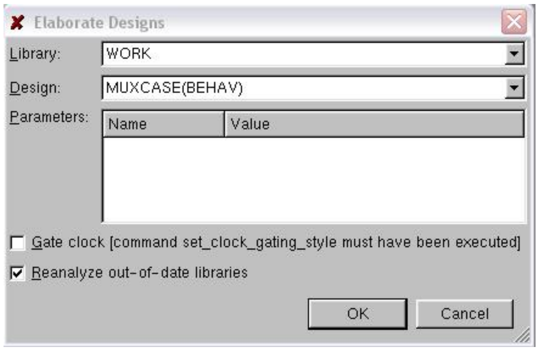
/egr/courses/personal/ece410/semigpet/synopsy...

Format: Auto

Work library: WORK

☑ Create new library if it does not exist

Figure 3: Analyze Designs window.



Elaborate Designs

Library: WORK

Design: MUXCASE(BEHAV)

Parameters: Name Value

☐ Gate clock [command set_clock_gating_style must have been executed]

☑ Reanalyze out-of-date libraries

Figure 5: Elaborate window.

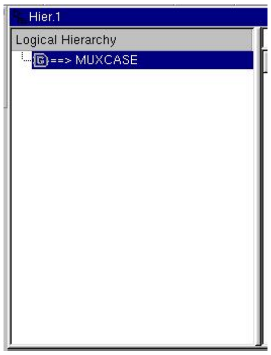
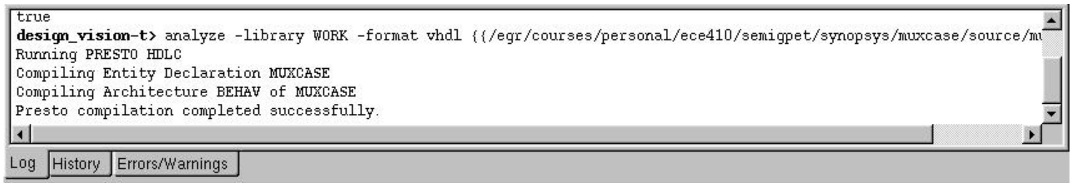


Figure 6: Select MUXCASE.



```
true
design_vision-t> analyze -library WORK -format vhdl {{/egr/courses/personal/ece410/semigpet/synopsys/muxcase/source/m
Running PRESTO HDLC
Compiling Entity Declaration MUXCASE
Compiling Architecture BEHAV of MUXCASE
Presto compilation completed successfully.
```

Figure 4: Design Vision report of successful compilation.

Place & Route with Cadence Silicon Ensemble

***NOTE: Some students have reported problems running* Gnome *desktop environment with Silicon Ensemble. It is recommended that you use the* Common Desktop Environment.**

GETTING STARTED

1. Create a folder under /egr/courses/personal/ece410/<username>/se called **mux** and go to that directory.

```
cd  /egr/courses/personal/ece410/<username>/se
mkdir  mux
cd  /egr/courses/personal/ece410/<username>/se/mux
```

2. Type **source $SOFT/cadence**

3. To launch Silicon Ensemble, type **seultra -m=200 &**

SETUP

4. Select **File** >> **Import** >> **LEF**

5. Select *iit05_stdcells.lef* from your *synopsys/lef* directory.

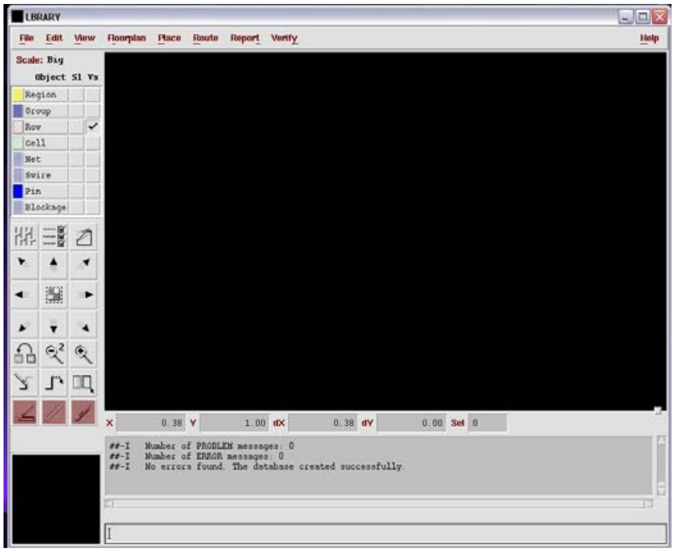
6. Click **OK**. Figure 7 shows the Silicon Ensemble program window with a message in the bottom feedback window that LEF importing was successful.

7. To add Verilog source files, select **File** >> **Import** >> **Verilog** and click **Browse** (near *Verilog Source Files*).

8. In the *MultiBrowse* window (Fig. 8), select *muxsyn.v* from your *synopsys/v* directory and click **Add**.

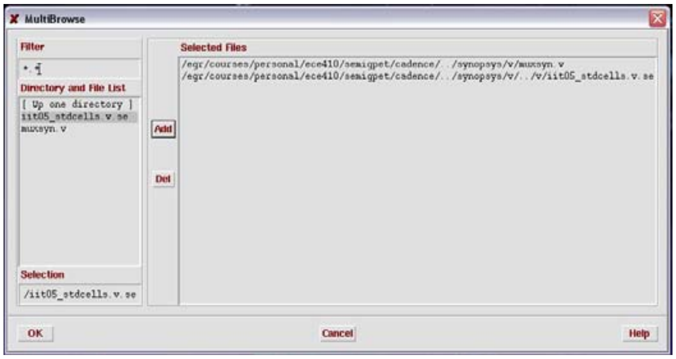
9. Also add *iit05_stdcells.v.se* from the same directory. The trick here is to change the *Filter* field from ***.v** to ***.*** and press ***Enter***.

10. Click **OK** to exit the *MultiBrowse* window. An *Import Verliog* window will open (Fig. 9).



```
##-I   Number of PROBLEM messages: 0
##-I   Number of ERROR messages: 0
##-I   No errors found. The database created successfully.
```

Figure 7: Import LEF into Silicon Ensemble.



Filter: *.*

Directory and File List:
```
[ Up one directory ]
iit05_stdcells.v.se
muxsyn.v
```

Selected Files:
```
/egr/courses/personal/ece410/semiqpet/cadence/../synopsys/v/muxsyn.v
/egr/courses/personal/ece410/semiqpet/cadence/../synopsys/v/../v/iit05_stdcells.v.se
```

Selection: /iit05_stdcells.v.se

Figure 8: Add Verilog Source Files.

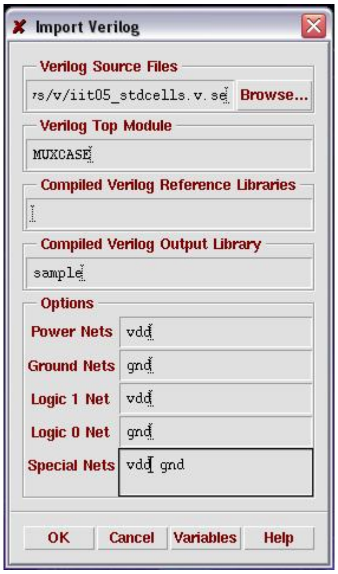
11. Enter **MUXCASE** for *Verilog Top Module*. This value is determined by examining the *entity* definition within muxcase.vhd (or the *module* definition within a Verilog file).

12. Delete everything in the *Compiled Verilog Reference Libraries* box.

13. Type **sample** in the *Compiled Verilog Output Library* box. You can actually type any name you desire in this box.

14. Delete all exclamation points (**!**) in the *Options* section.

15. Select **OK** to close the *Import Verilog* window.



Import Verilog

Verilog Source Files: rs/v/iit05_stdcells.v.se

Verilog Top Module: MUXCASE

Compiled Verilog Reference Libraries:

Compiled Verilog Output Library: sample

Options

Power Nets: vdd

Ground Nets: gnd

Logic 1 Net: vdd

Logic 0 Net: gnd

Special Nets: vdd gnd

Figure 9: Import Verilog Dialog.

FLOORPLANNING

16. From the main window, select **Floorplan**>> **Initialize Floorplan**

17. In the *I/O To Core Distance* section of the new window (Fig. 10), input **12 *microns*** for both fields.

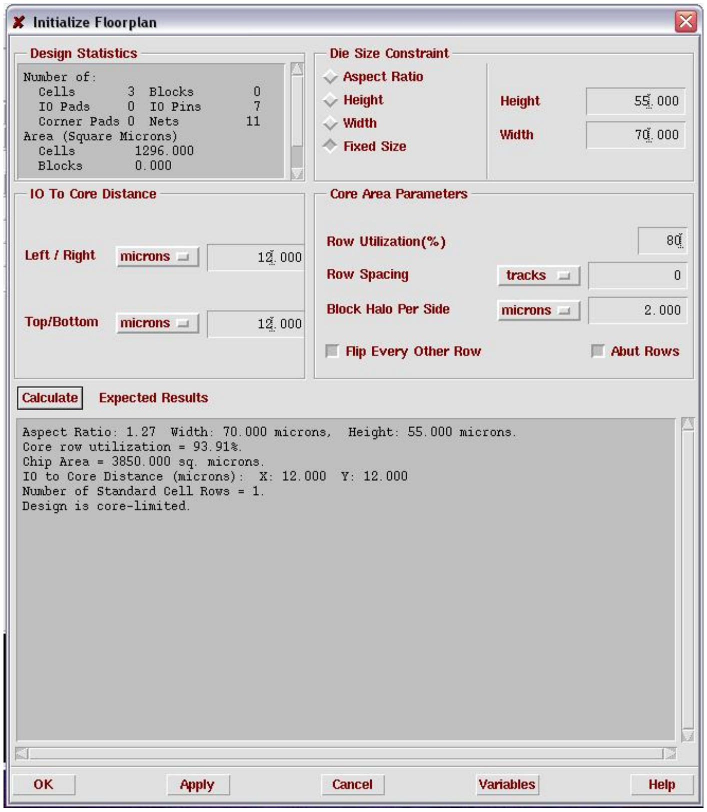
18. Select **Fixed Size** in the *Die Size Constraint* section and input **55** for the *Height* and **70** for the *Width*.

The actual die size and area are reported in the message window after initializing the floorplan (Fig. 11). You may have to scroll up to see the size and area. The area is also reported in the file *se/LIBRARY.summary.* Look for a line starting with *Area of chip:* around line 50. The width and height units are in nanometers. The area units, square DBU, are equivalent to square nanometers in our design.

19. Under *Core Area Parameters* change the *Row Utilization* to **80**% and make sure *Flip Every Other Row* and *Abut Rows* are checked.

20. Click the **Calculate** button. You should see some statistics such as row utilization and number of rows. Then click **OK** to close the floorplan window.

21. Back in the main window (Fig. 12), select **Place** >> **Ios** then click **OK**.



Initialize Floorplan

Design Statistics

```
Number of:
  Cells       3  Blocks        0
  IO Pads     0  IO Pins       7
  Corner Pads 0  Nets         11
Area (Square Microns)
  Cells        1296.000
  Blocks       0.000
```

Die Size Constraint

◇ Aspect Ratio
◇ Height
◇ Width
◆ Fixed Size

Height: 55.000

Width: 70.000

IO To Core Distance

Left / Right: microns 12.000

Top/Bottom: microns 12.000

Core Area Parameters

Row Utilization(%): 80

Row Spacing: tracks 0

Block Halo Per Side: microns 2.000

☑ Flip Every Other Row ☑ Abut Rows

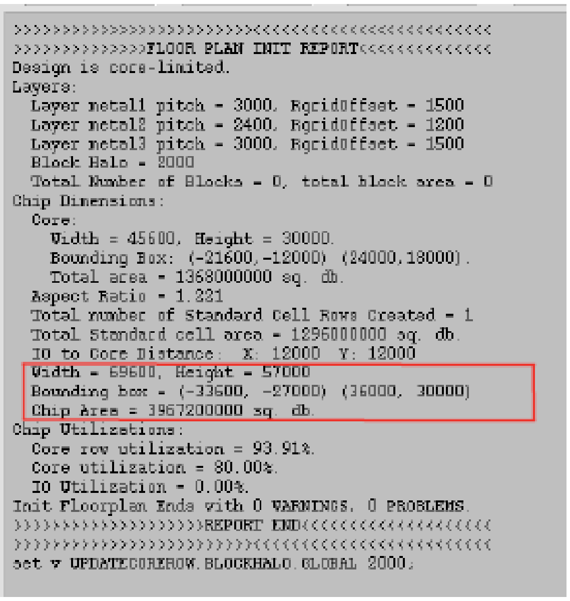
Expected Results

```
Aspect Ratio: 1.27  Width: 70.000 microns,  Height: 55.000 microns.
Core row utilization = 93.91%.
Chip Area = 3850.000 sq. microns.
IO to Core Distance (microns):  X: 12.000  Y: 12.000
Number of Standard Cell Rows = 1.
Design is core-limited.
```

Figure 10: Initialize Floorplan.

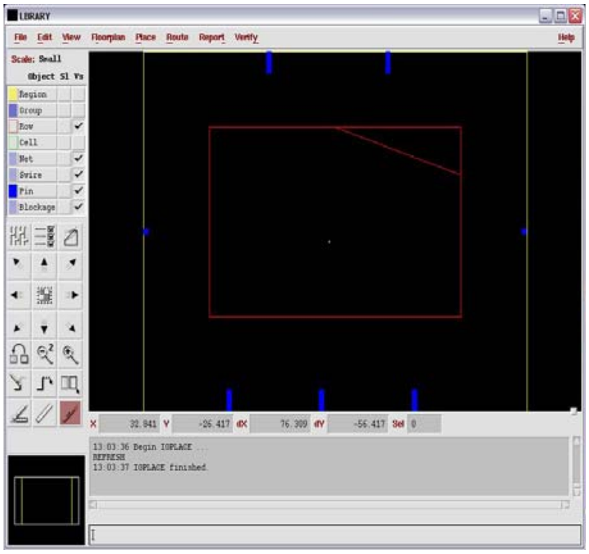
22. To create power and ground rings, select **Route** >> **Create Ring**.

23. In the window that opens (Fig 13), make sure *Core ring(s)* is selected and change all values of *Width*, *Spacing*, and *Offset* to **1.5**. Click **OK**. The layout should now look like Figure 14.



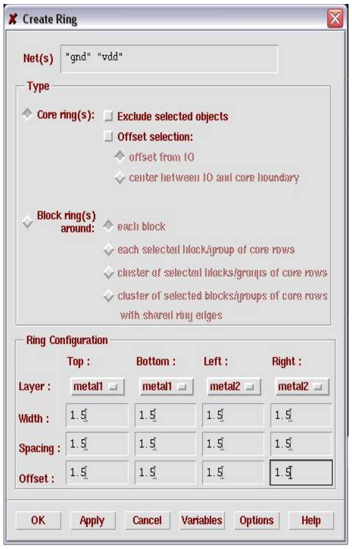
```
>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<
>>>>>>>>>>>>>>FLOOR PLAN INIT REPORT<<<<<<<<<<<<<<
Design is core-limited.
Layers:
  Layer metal1 pitch = 3000. RgridOffset = 1500
  Layer metal2 pitch = 2400. RgridOffset = 1200
  Layer metal3 pitch = 3000. RgridOffset = 1500
  Block Halo = 2000
  Total Number of Blocks = 0, total block area = 0
Chip Dimensions:
  Core:
    Width = 45600, Height = 30000.
    Bounding Box: (-21600,-12000) (24000,18000).
    Total area = 1368000000 sq. db.
  Aspect Ratio = 1.221
  Total number of Standard Cell Rows Created = 1
  Total Standard cell area = 1296000000 sq. db.
  IO to Core Distance:  X: 12000  Y: 12000
  Width = 69600, Height = 57000
  Bounding box = (-33600, -27000) (36000, 30000)
  Chip Area = 3967200000 sq. db.
Chip Utilizations:
  Core row utilization = 93.91%.
  Core utilization = 80.00%.
  IO Utilization = 0.00%.
Init Floorplan Ends with 0 WARNINGS. 0 PROBLEMS.
>>>>>>>>>>>>>>>>>>>>>REPORT END<<<<<<<<<<<<<<<<<<<
>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<
set v UPDATECOREROW.BLOCKHALO.GLOBAL 2000;
```

Figure 11: Initialize Floorplan feedback in message window.


LIBRARY
File Edit View Floorplan Place Route Report Verify Help
Scale: Small
Object Sl Vs
Region
Group
Row
Cell
Net
Swire
Pin
Blockage
X 32.941 Y -26.417 dX 76.309 dY -56.417 Sel 0

```
13:03:36 Begin IOPLACE ...
REFRESH
13:03:37 IOPLACE finished.
```

Figure 12: Place Inputs and Outputs.



Net(s) "gnd" "vdd"

Type

● Core ring(s): □ Exclude selected objects
□ Offset selection:
● offset from IO
○ center between IO and core boundary

○ Block ring(s) around: ● each block
○ each selected block/group of core rows
○ cluster of selected blocks/groups of core rows
○ cluster of selected blocks/groups of core rows with shared ring edges

Ring Configuration

	Top :	Bottom :	Left :	Right :
Layer :	metal1	metal1	metal2	metal2
Width :	1.5	1.5	1.5	1.5
Spacing :	1.5	1.5	1.5	1.5
Offset :	1.5	1.5	1.5	1.5

Figure 13: Create Vdd and Gnd Rings.

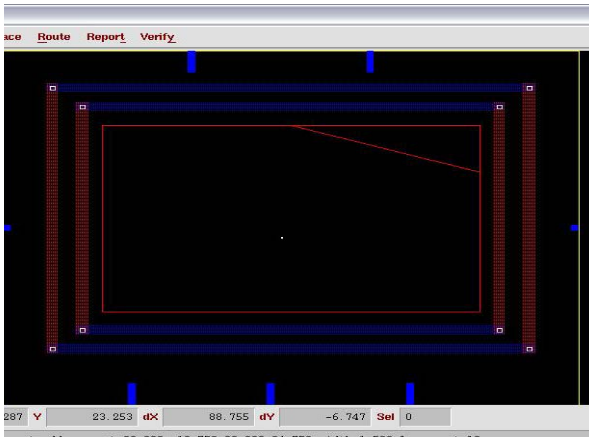


Figure 14: Vdd and Gnd Rings.

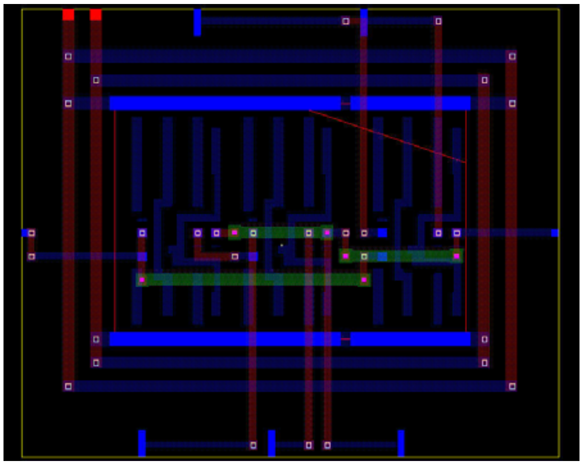


Figure 15: Layout after placing cells.

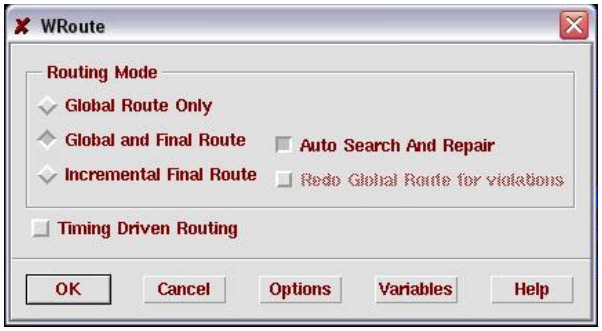
PLACE AND ROUTE

24. We can now add cells to the floorplan. Select **Place** >> **Cells**.

25. In the window that opens, all boxes should be unchecked. Click **OK** to close this window. Back in the main window, you will see some metal layers from the added cells (Fig. 15). Since all the layers are not defined in *Silicon Ensemble*, you won't see all the layers. You'll see the whole design later in *Virtuoso*.

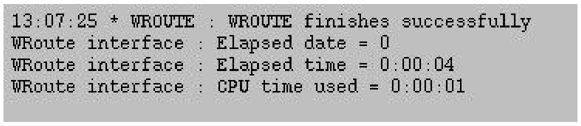
26. To route the cells to the power and ground rings, select **Route** >> **Connect Ring**. Click **OK**.

27. Next we have the tools route all the signals. Select **Route** >> **WRoute**.



○ Global Route Only
● Global and Final Route
○ Incremental Final Route
□ Auto Search And Repair
□ Redo Global Route for violations
□ Timing Driven Routing

Figure 16: Wire Routing.



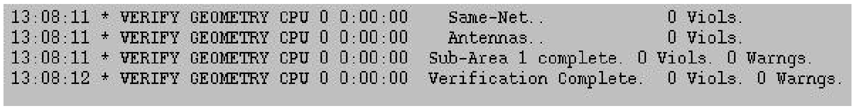
```
13:07:25 * WROUTE  : WROUTE finishes successfully
WRoute interface : Elapsed date = 0
WRoute interface : Elapsed time = 0:00:04
WRoute interface : CPU time used = 0:00:01
```

Figure 17: Successful WROUTE.

28. In the *WRoute* window (Fig 16), make sure *Global and Final Route* and *Auto Search And Repair* are checked. Click **OK** and you should see the results shown in Figure 17 at the bottom of the main window.

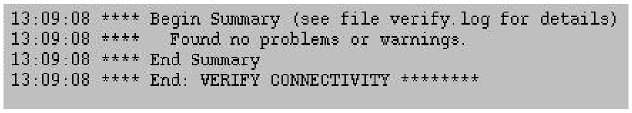
29. Now we need to run a couple of checks. Complete the following steps and check the bottom of the main window for the verifications shown in Figures 18-19.

Select **Verify** >> **Geometry**. Click **OK**.
Select **Verify** >> **Connectivity**. Click **OK**.



```
13:08:11 * VERIFY GEOMETRY CPU 0 0:00:00    Same-Net..            0 Viols.
13:08:11 * VERIFY GEOMETRY CPU 0 0:00:00    Antennas..            0 Viols.
13:08:11 * VERIFY GEOMETRY CPU 0 0:00:00  Sub-Area 1 complete.  0 Viols.  0 Warngs.
13:08:12 * VERIFY GEOMETRY CPU 0 0:00:00  Verification Complete.   0 Viols.  0 Warngs.
```

Figure 18: Verify Geometry.



```
13:09:08 **** Begin Summary (see file verify.log for details)
13:09:08 ****   Found no problems or warnings.
13:09:08 **** End Summary
13:09:08 **** End: VERIFY CONNECTIVITY ********
```

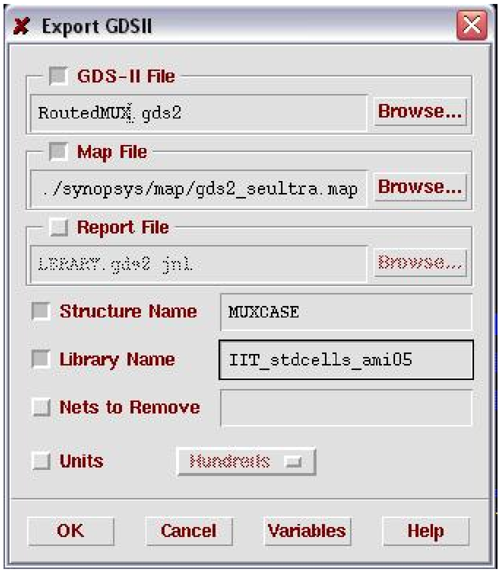
Figure 19: Verify Connectivity.

EXPORT LAYOUT

30. The final step here is to export the layout to a format that can be read by the Cadence layout tools. Select **File** >> **Export** >> **GDSII**.

31. In the window that opens (Fig. 20), check the *GDS-II File* box and enter **RoutedMUX.gds2.**

32. Check *Map File*, and Browse to *synopsys/map* and select the *gds2_seultra.map* file.



□ GDS-II File: RoutedMUX.gds2
□ Map File: ./synopsys/map/gds2_seultra.map
□ Report File: LIBRARY.gds2.jnl
□ Structure Name: MUXCASE
□ Library Name: IIT_stdcells_ami05
□ Nets to Remove:
□ Units: Hundreds

Figure20: Export GDSII.

33. Check *Structure Name* and make sure **MUXCASE** is entered in that box.

34. Check *Library Name* and make sure **IIT_stdcells_ami05** is in that box.

35. Click **OK**. The export might take a minute. When you see *Process Complete* in the main window, you are done.

36. Exit *Silicon Ensemble*. If it asks you to save your database, select **Yes**.

Importing the GDSII File into a Virtuoso Library

1. Go to the following directory:
/egr/courses/personal/ece410/<username>/cadence

2. The Cadence tools should already be sourced so launch them by entering the command **icfb &.**

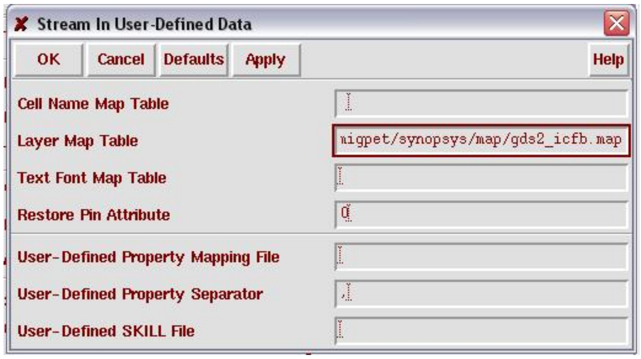
3. First we need to connect with the IIT_stdcells_ami05 library. You should have already copied the directory into your Cadence directory (last step of Environment Setup section). In the Library Manager, select **File >> New >> Library**.

4. Under Name, type **IIT_stdcells_ami05**. Under Path, put a period (**.**). Click **OK**. You should now see the library in the Library Manager. Make sure that you can open some cell layouts before you continue.

5. In the *Command Interpreter Window* (*CIW*) select **File >> Import >> Stream.**

6. Click the *Options* button and check *Snap XY to Grid Resolution*. Click **OK**.

7. Click the *User-Defined Data* button and, in *LayerMap Table*, type (Fig 21) **/egr/courses/personal/ece410/<username>/synopsys/map/gds2_icfb.map**. Click **OK**.



Cell Name Map Table
Layer Map Table nigpet/synopsys/map/gds2_icfb.map
Text Font Map Table
Restore Pin Attribute 0
User-Defined Property Mapping File
User-Defined Property Separator ,
User-Defined SKILL File

Figure 21: Layer Map Table

8. In *Input File*, type **/egr/courses/personal/ece410/<username>/se/mux/RoutedMUX.gds2**

9. In *Library Name*, type **IIT_stdcells_ami05**.

10. Click **OK**. You will probably get a warning. Click **OK** if this happens.

11. In the *Library Manager*, open the layout view of the **MUXCASE** cell in the *IIT_stdcells_ami05* library. (If you cannot see the **MUXCASE** cell, select **View >> Refresh** to refresh the window). You will see some routing and three mux cells (Fig 22).

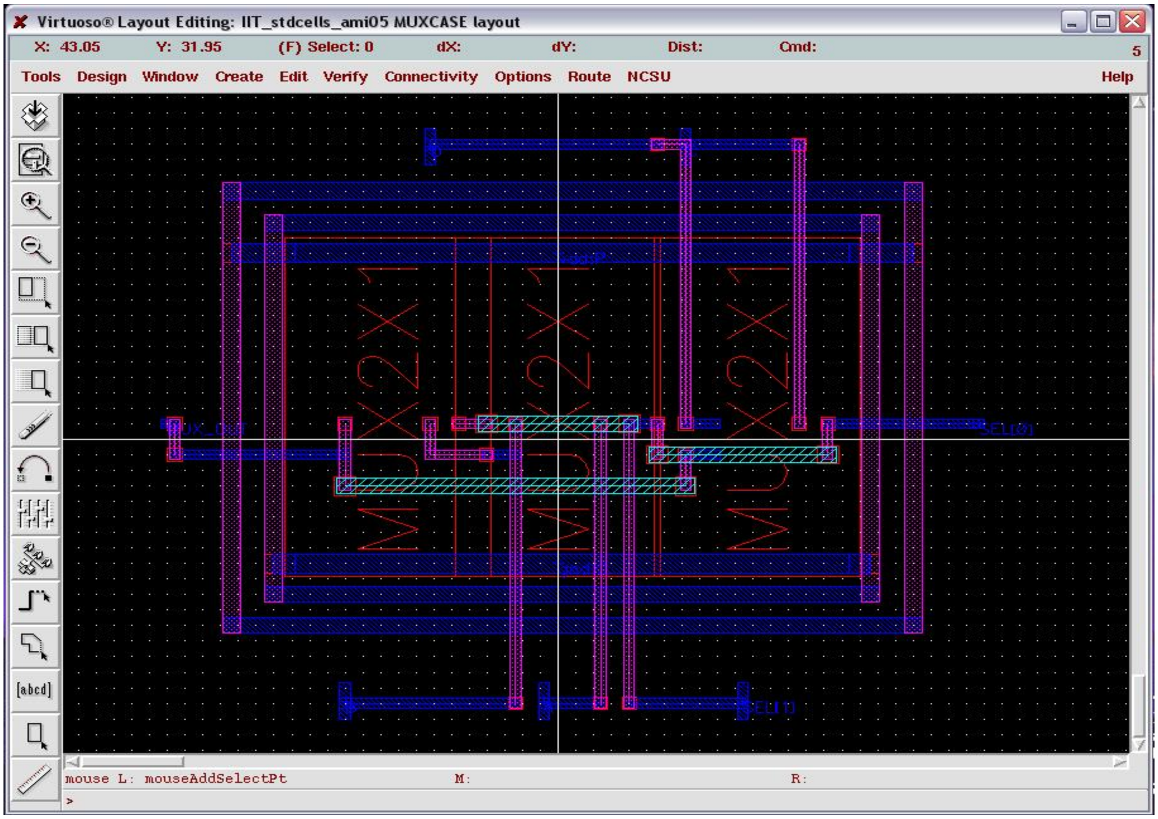
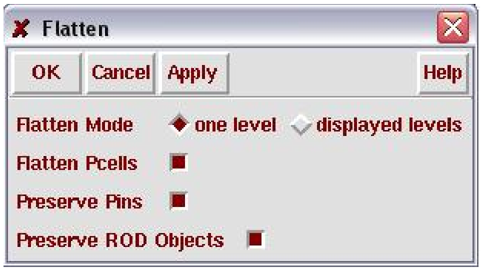


Figure 22: Initial Virtuoso Layout

12. Now we need to ’flatten’ the mux instances. First use the mouse to select the entire design. You do this by left-clicking in the upper left corner and dragging a box to the lower right corner.

13. With everything selected, go to **Edit >> Hierarchy >> Flatten** in the Virtuoso window.

14. Check *one level*, *Flatten Pcells*, *Preserve Pins*, and *Preserve ROD Objects* as in Figure 23. Click **OK**.



Flatten Mode ◆ one level ◇ displayed levels

Flatten Pcells ■

Preserve Pins ■

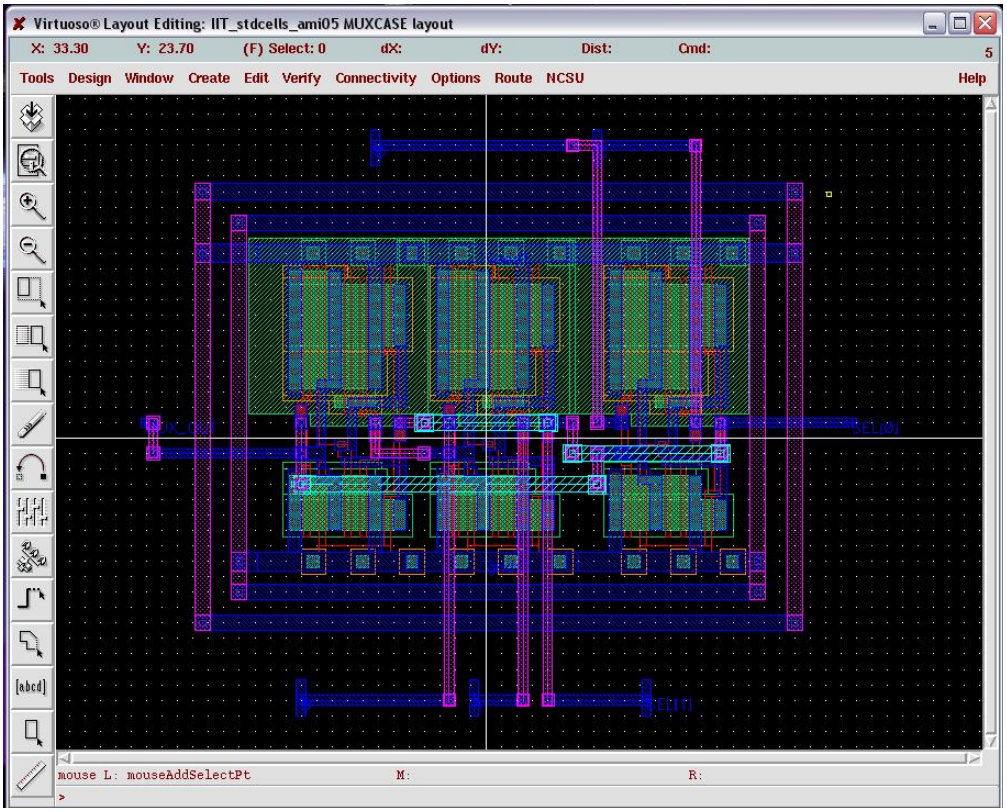
Preserve ROD Objects ■

Figure 23: Flatten Cells

15. Now you should see the entire design, as shown in Figure 24.

16. Run a *DRC* check. You will find that there are several errors. Each error corresponds to a pin and we need to redo the pins. You should now see the name and location of each of the pins.

17. Place new *Metal1* pin for **vdd!, gnd!, A, B, C, D, MUX_OUT, S0,** and **S1**. Be sure to make **A, B, C, D, S0,** and **S1** inputs. Make **MUX_OUT** an output. Make **vdd!** and **gnd!** jumpers.


Virtuoso® Layout Editing: IIT_stdcells_ami05 MUXCASE layout

X: 33.30 Y: 23.70 (F) Select: 0 dX: dY: Dist: Cmd: 5

mouse L: mouseAddSelectPt M: R:

Figure 24: Final Circuit

After you place the new pins, be sure to delete the old text pins. The text will not disappear after deleting, but it will disappear after re-running the *DRC*.

18. Re-run the *DRC* to verify no errors.

We're now ready to use the Cadence tools to extract a netlist of the active devices, parasitics capacitances and interconnections. The extracted netlist can be used to simulate the performance of the circuit with results that are generally very close to those obtained after fabrication.

Simulating from an Extracted Netlist

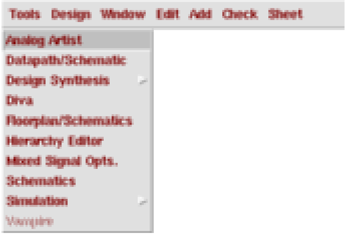
To verify functionality of the design, we must extract the layout and simulate the extracted netlist.

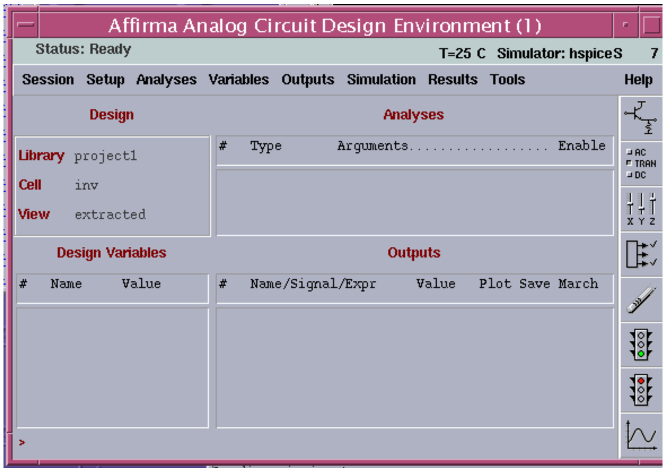
STEP 1. Extract Netlist

Extract the netlist for the MUXCASE layout and load the extracted cellview. This process is described in detail in Tutorial B between DRC and LVS procedures.

STEP 2. Start Analog Environment

- With the extracted view open, in the Virtuoso Layout Editing window select Tools => Analog Environment to open the Affirma Analog Circuit Design Environment window.




Affirma Analog Circuit Design Environment (1)

Status: Ready T=25 C Simulator: hspiceS 7

Session Setup Analyses Variables Outputs Simulation Results Tools Help
Design
Analyses
# Type Arguments................. Enable
Design Variables
# Name Value
Outputs
# Name/Signal/Expr Value Plot Save March

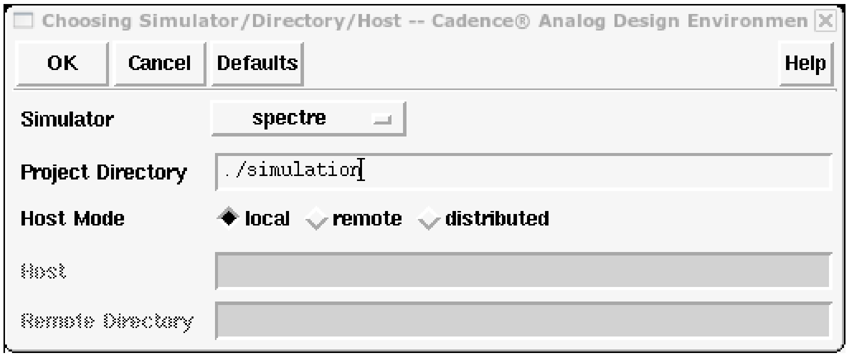
Library project1

Cell inv

View extracted

STEP 3. Setup Analog Simulator

- In Affirma Analog Circuit Design Environment, click on Setup => Simulator/Directory/Host.
- Choose **spectre** as the Simulator.
- Enter a path for your simulations files and results. You may set this to any valid path, but you might find it useful to keep all simulations in one directory. For example 410 students might use */egr/courses/personal/ece410/<username>/se/simulation/muxcase*. If you need to run multiple simulations on the same cell, you can even use different paths for each simulation.


Choosing Simulator/Directory/Host -- Cadence® Analog Design Environmen

Simulator spectre

Project Directory ./simulation

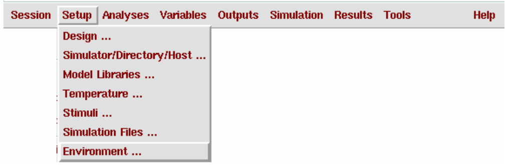
Host Mode ● local ○ remote ○ distributed

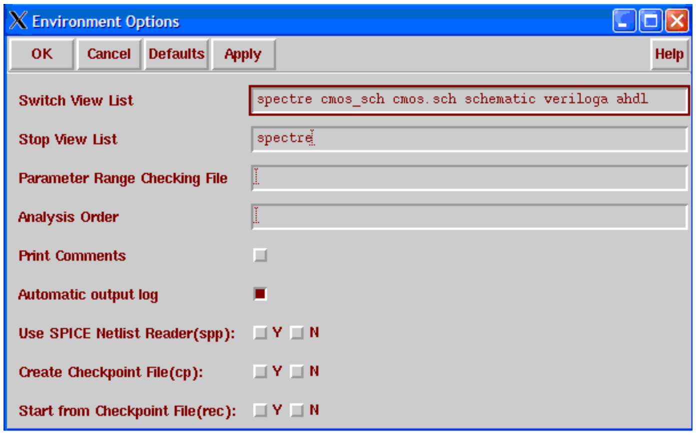
Host

Remote Directory

STEP 4. Setup Analog Environment to use Extracted View

- In Affirma Analog Circuit Design Environment, click on **Setup => Environment.** The Environment Options dialog box (below) will pop up.


Session Setup Analyses Variables Outputs Simulation Results Tools Help
Design ...
Simulator/Directory/Host ...
Model Libraries ...
Temperature ...
Stimuli ...
Simulation Files ...
Environment ...


Environment Options

Switch View List spectre cmos_sch cmos.sch schematic veriloga ahdl

Stop View List spectre

Parameter Range Checking File

Analysis Order

Print Comments □

Automatic output log ■

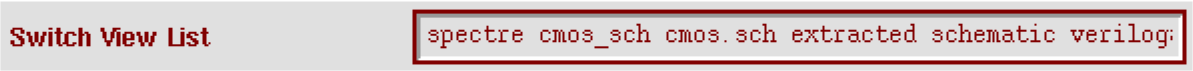
Use SPICE Netlist Reader(spp): □ Y □ N

Create Checkpoint File(cp): □ Y □ N

Start from Checkpoint File(rec): □ Y □ N

- In the Environment Options window, under the line *Switch View List*. type the word "**extracted**" before the word "schematic" (see below).

This entry is an ordered list of cell views which contain information that can be simulated. The simulator (in fact the netlister) will search until it finds one of these cellviews. The default entry does not contain an *extracted* cellview. As a result of this modification, the simulator will use the extracted cellview instead of the schematic cellview to include the effect of parasitic capacitance in the simulation.



Switch View List spectre cmos_sch cmos.sch extracted schematic verilog

- Make sure to check the Y for Use SPICE Netlist Reader(spp), since the default transistor model files are written in SPICE syntax. Click OK once you have done this.

STEP 5. Simulation

- A stimulus file for the MUXCASE cell has been created for you. Copy this file from /egr/courses/personal/ece410/resources/stimulus.txt to an appropriate location in your directory.
- Use this stimulus file to simulate the cell and verify functionality. Refer to Tutorial A for information about running this simulation.
- Be sure to run the simulation for 360ns.
- When you choose the signals to plotted, the extracted view of the mux should come to the forefront. Select the signals to be plotted by clicking on the corresponding input and output nets. The names of the nets should appear in the "Outputs" box in the Affirma Analog Circuit Design Environment window.
- Verify the circuit functions like you expect a multiplexer to operate.
- Save your simulations results to include in your report.

THE END