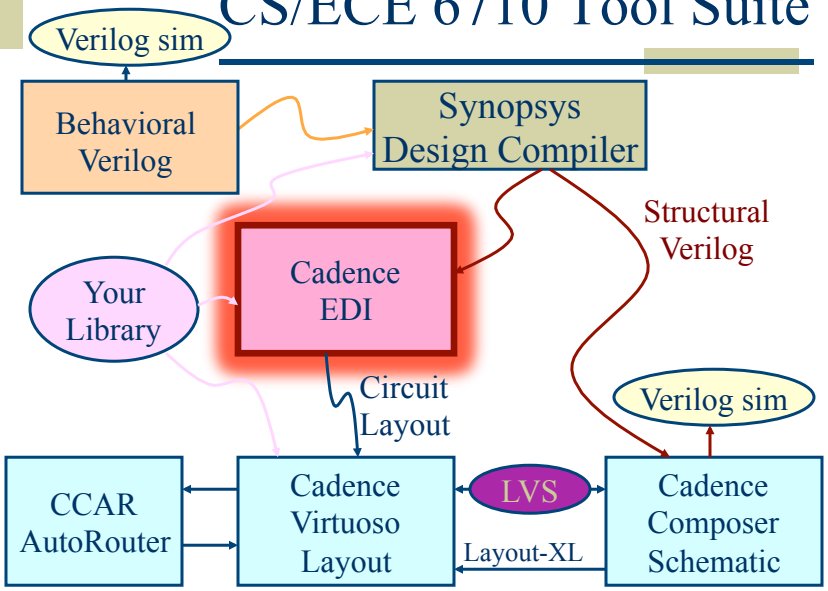
Place & Route

Cadence Innovus place & route

CS/ECE 6710 Tool Suite



In the CAD Book

- Chapter 11 on SOC Encounter Place and Route
  - That's an old name – In 2015 they changed it to Cadence EDI (Encounter Design Implementation)
  - Now that's an old name – it's now called Cadence Innovus…
  - Need additional information about your cells
- Specifically, a .lef (physical place &route) file
  - This basically describes the abstract views of your cells in a way that place and route understands…

Encounter Innovus Place & Route

1. Import Design
2. Floorplan
3. Power plan
4. Place cells
5. Synthesize clock tree?
6. Route signal nets
7. Verify results
8. Write out results

Convert structural Verilog
(From Synopsys)
Into physical layout

Innovus Usage

- Need…
  - Structural Verilog <design>.v (from Synopsys)
  - Structural Verilog timing, <design>.sdc (from Synopsys)
  - Library timing information <library>.lib (from Liberate)
  - Library layout information <library>.lef (from Abstract)
  - 6710.tcl file with Innovus variable settings (from /uusoc/facility/cad_common/local/class/6710/F17/cadence/Innovus
  - mmmc.tcl file with timing information (multi-mode multi-corner timing)
- Make a new dir for Innovus... (I call mine INN)
- Call with cad-inn

cad-innFlow

1. Import Design
   - .v, .sdc, .lib, .lef, mmmc.tcl
2. Source 6710.tcl file to get things set up
3. Floorplan
   - Choose physical size, ratio, utilization percentage, etc.
4. Power plan
   - rings, stripes, row-routing (sroute)

cad-edi Flow

5. Placement
   - place cells in the rows
   - …with optimization step
6. Synthesize clock tree?
   - use your buf and inv footprint cells
   - maybe not needed – Synopsys already did this…
7. Global routing
   - NanoRoute
   - …with optimization step

cad-edi Flow

8. Add filler cells
   - Fill in the spots in the row with no cells
   - Adds NWELL for continuity
9. Write out results
   - <name>.def can be imported as layout
   - <name>_innovus.v is the placed and routed Verilog dscription
   - Write out timing information if desired. .spef, .sdc, _innovus.lib

To start…

```
[elb@lab2-20 INN]$ ls
6710.tcl  CAD6.lef  CAD6.lib  controller_struct.sdc  controller_struct.v  mmmc.tcl
[elb@lab2-20 INN]$
```

The set of files needed…

```
#
# set the name of your .lib file (e.g. Lib6710_01.lib)
# You can create multiple library sets if you have multiple libraries
# such as fast, slow, and typ
# If you have multiple .lib files put them in a [list lib1 lib2] structure
create_library_set -name typical_lib \
   -timing {CAD6.lib}
# Specify the .sdc timing constraint file to use
# This file comes from Synopsys synthesis. (e.g. design_struct.sdc)
create_constraint_mode -name typical_constraint \
   -sdc_files {controller_struct.sdc}
#
#################################################################
```

mmmc.tcl timing description

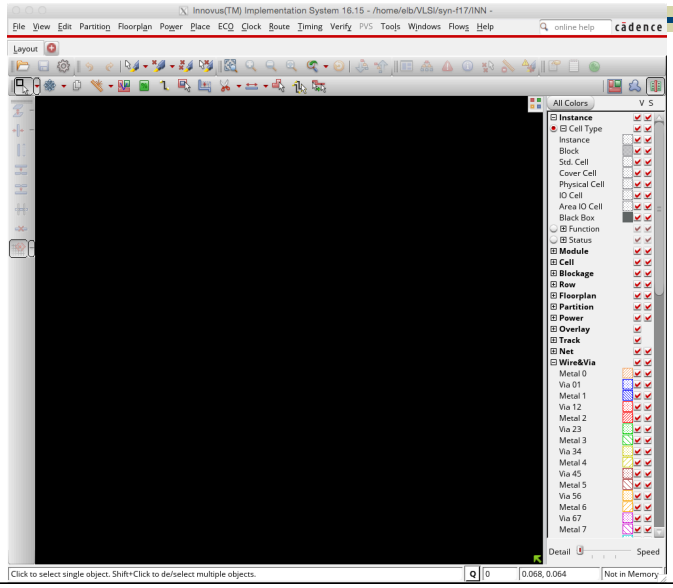
mmmc.tcl

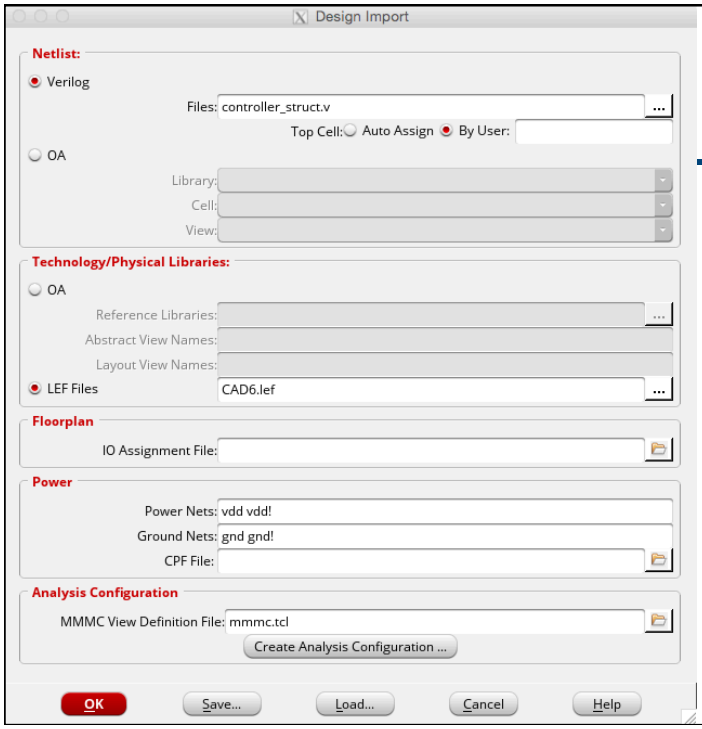
```
# set the name of your .lib file (e.g. Lib6710_01.lib)
# You can create multiple library sets if you have multiple libraries
# such as fast, slow, and typ
# If you have multiple .lib files put them in a [list lib1 lib2] structure
create_library_set -name typical_lib \
   -timing {!!your-lib-file!!.lib}
# Specify the .sdc timing constraint file to use
# This file comes from Synopsys synthesis. (e.g. design_struct.sdc)
create_constraint_mode -name typical_constraint \
   -sdc_files {!!your-sdc-file!!.sdc}
…
```

mmmc.tcl

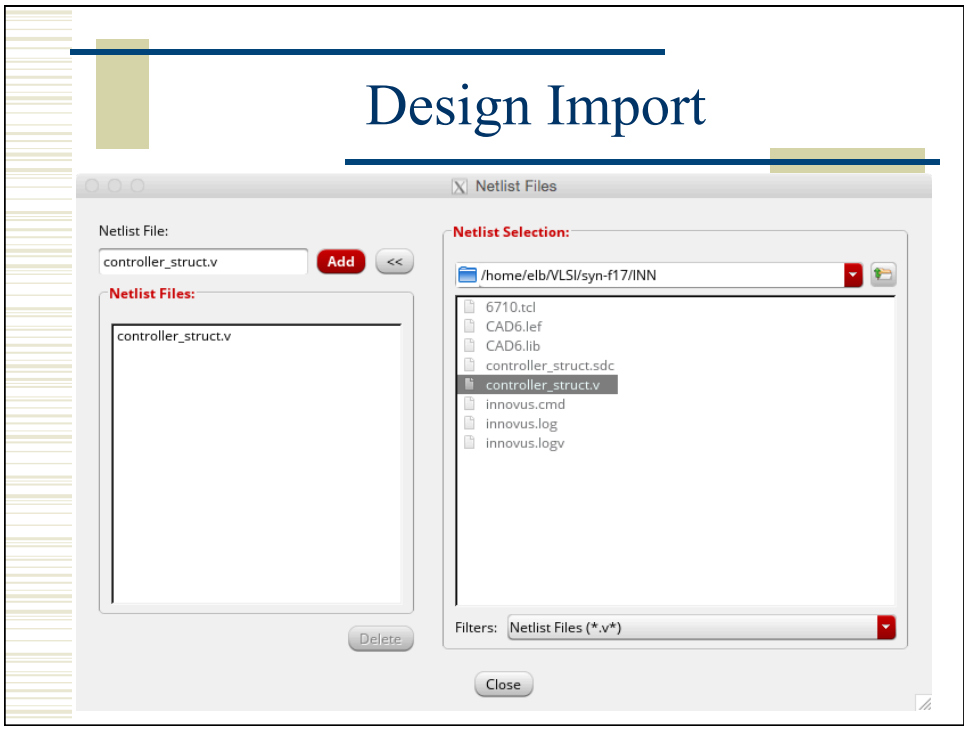
```
#################################################################
# Below here you shouldn't have to change, unless you're doing
# something different than the basic EDI run...
#################################################################
# Create an RC_corner that has specific capacatance info.
create_rc_corner -name typical_rc\
…
# Define delay corners and analysis views.
create_delay_corner -name typical_corner \
    -library_set {typical_lib} \
    -rc_corner {typical_rc}
create_analysis_view -name typical_view \
    -constraint_mode {typical_constraint} \
    -delay_corner {typical_corner}
# Now define which analysis view to use for setup and for hold.
set_analysis_view -setup {typical_view} -hold {typical_view}
```

cad-inn gui





Design Import



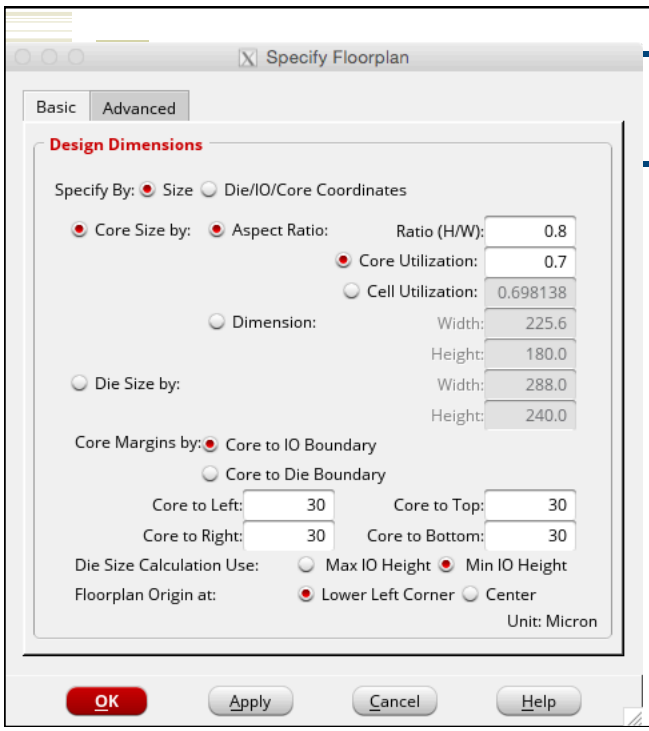
Result of Successful Import

```
*** Summary of all messages that are not suppressed in this session:
Severity  ID              Count  Summary
WARNING   IMPFP-3961          3  The techSite '%s' has no related standar...
WARNING   IMPTS-11            2  cell '%s' may have cyclic timing arc dec...
WARNING   IMPEXT-2766         3  The sheet resistance for layer %s is not...
WARNING   IMPEXT-2773         1  The via resistance between layers %s and...
WARNING   IMPEXT-2776         2  The via resistance between layers %s and...
WARNING   TCLNL-330           1  set_input_delay on clock root '%s' is no...
WARNING   TECHLIB-436        20  Attribute '%s' on '%s' pin '%s' of cell ...
*** Message Summary: 32 warning(s), 0 error(s)

innovus 1>
```

Source 6710.tcl

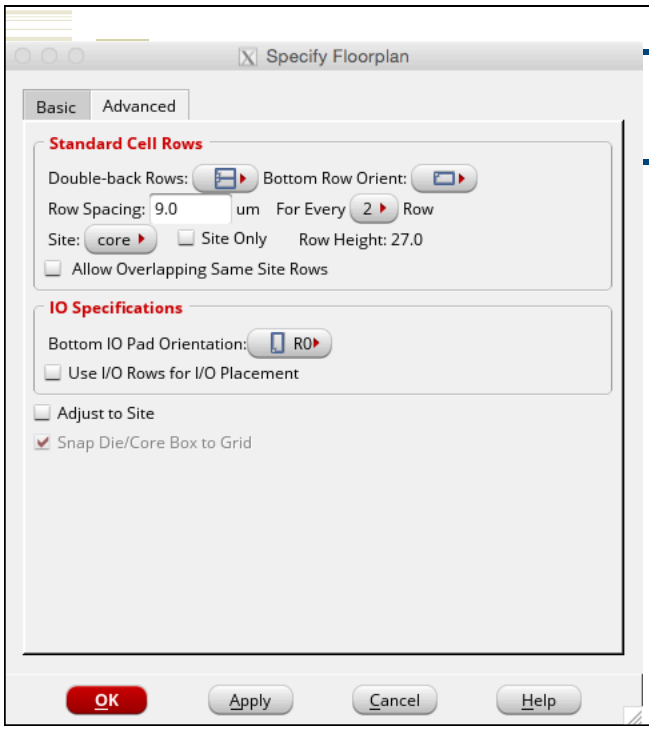
Type command to the Innovus command line

```
innovus 1> source 6710.tcl
Applying the recommended capacitance filtering threshold values for 250nm process node: total_c_th=5, re
lative_c_th=0.03 and coupling_c_th=3.
        These values will be used by all post-route extraction engines, including tQuantus, IQRC and Qua
ntus QRC extraction.
        For postRoute effortLevel low extraction, the coupling_c_th value will be used only when the -ca
pFilterMode parameter of the setExtractRCMode command is set to relAndCoup or relOrCoup. Default value o
f -capFilterMode parameter in postRoute extraction effortLevel low is relOnly.
        The accuracy mode for postRoute effortLevel low extraction will be set to 'standard'.
        Default value for EffortLevel(-effortLevel option of the setExtractRCMode) in postRoute extracti
on mode is 'low'.
Updating process node dependent CCOpt properties for the 250nm process node.
Applying the recommended capacitance filtering threshold values for 250nm process node: total_c_th=5, re
lative_c_th=0.03 and coupling_c_th=3.
        These values will be used by all post-route extraction engines, including tQuantus, IQRC and Qua
ntus QRC extraction.
        For postRoute effortLevel low extraction, the coupling_c_th value will be used only when the -ca
pFilterMode parameter of the setExtractRCMode command is set to relAndCoup or relOrCoup. Default value o
f -capFilterMode parameter in postRoute extraction effortLevel low is relOnly.
        The accuracy mode for postRoute effortLevel low extraction will be set to 'standard'.
        Default value for EffortLevel(-effortLevel option of the setExtractRCMode) in postRoute extracti
on mode is 'low'.
Updating process node dependent CCOpt properties for the 250nm process node.
typical_view
innovus 2>
```



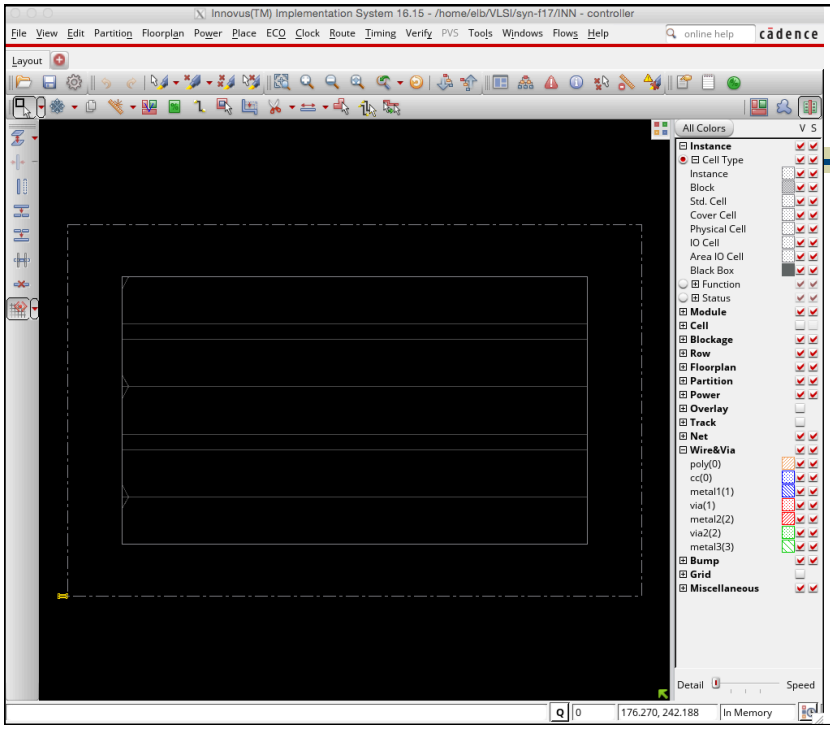
Floorplan

Specify -> Floorplan

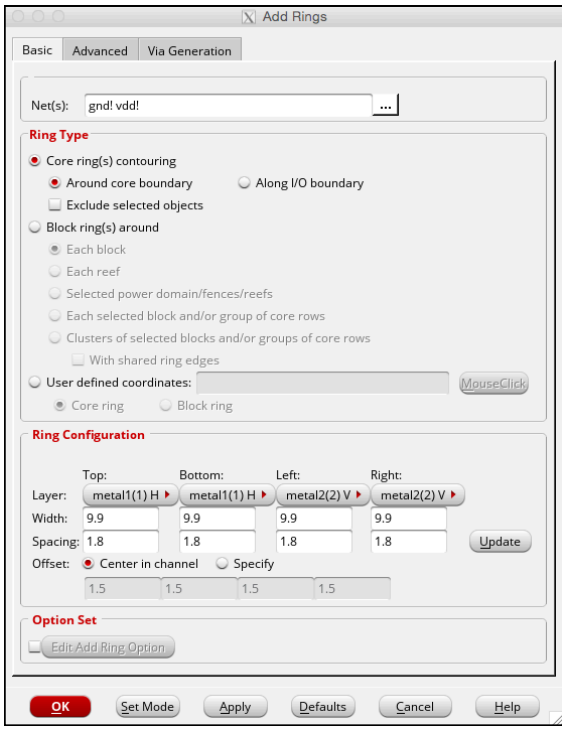


Floorplan

Specify -> Floorplan

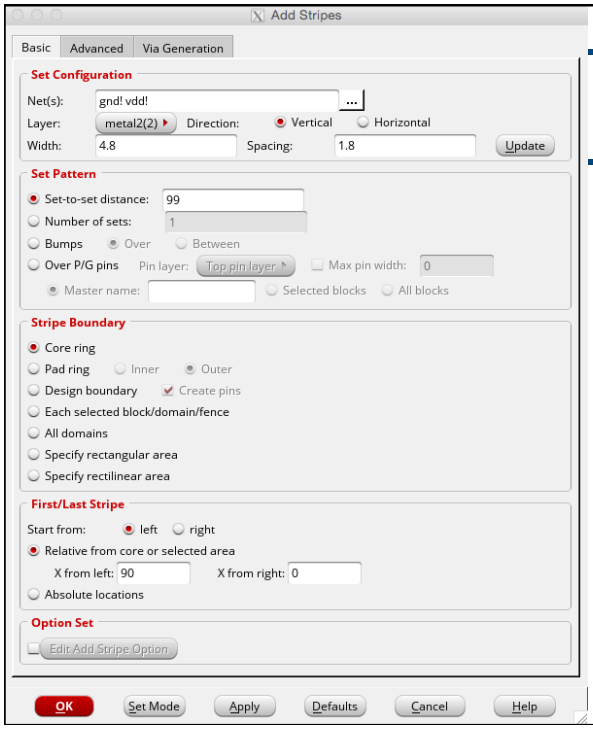


Floor plan

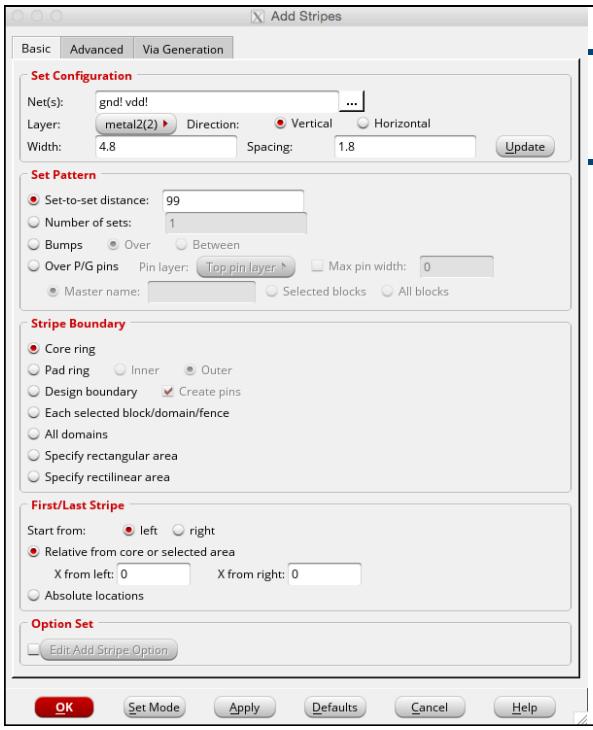


Power Rings

Power -> Power Planning

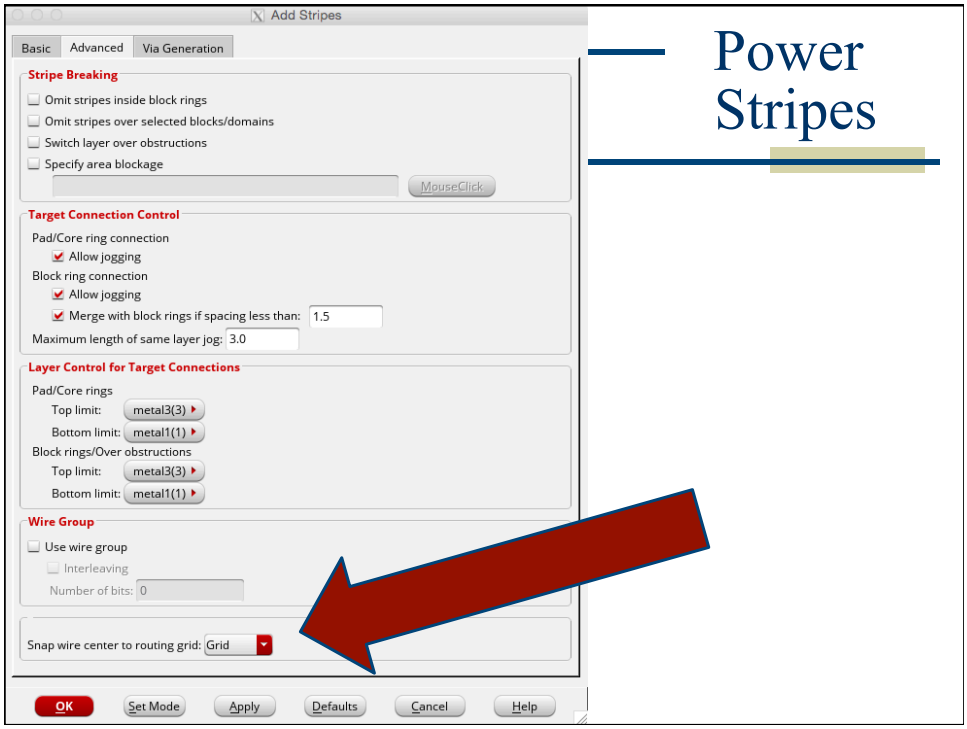


Power Stripes

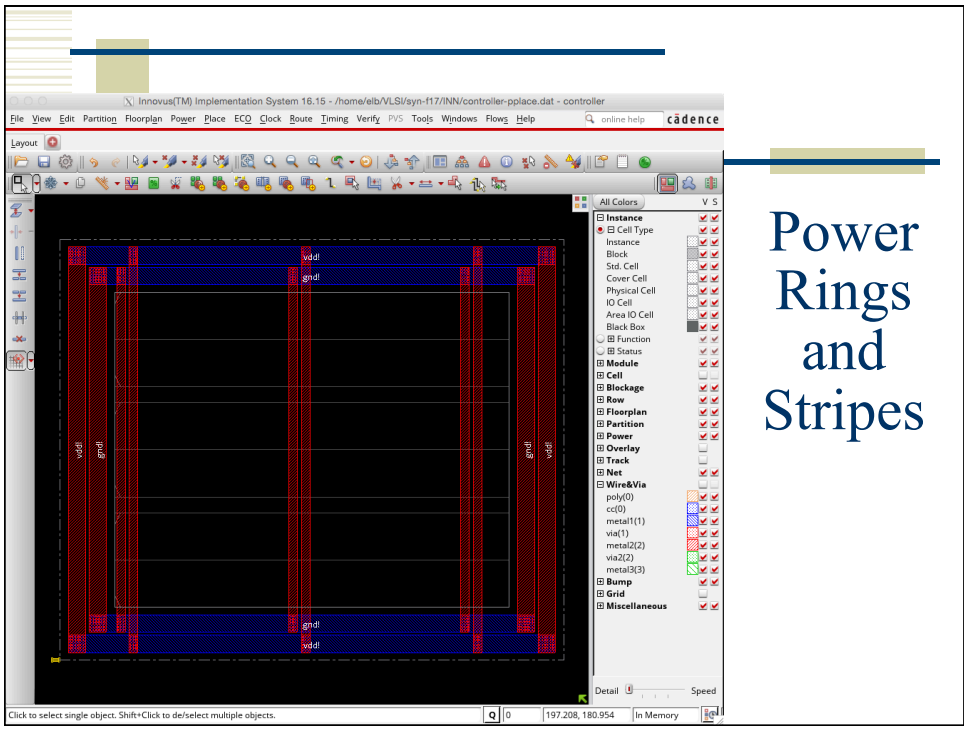


Power Stripes

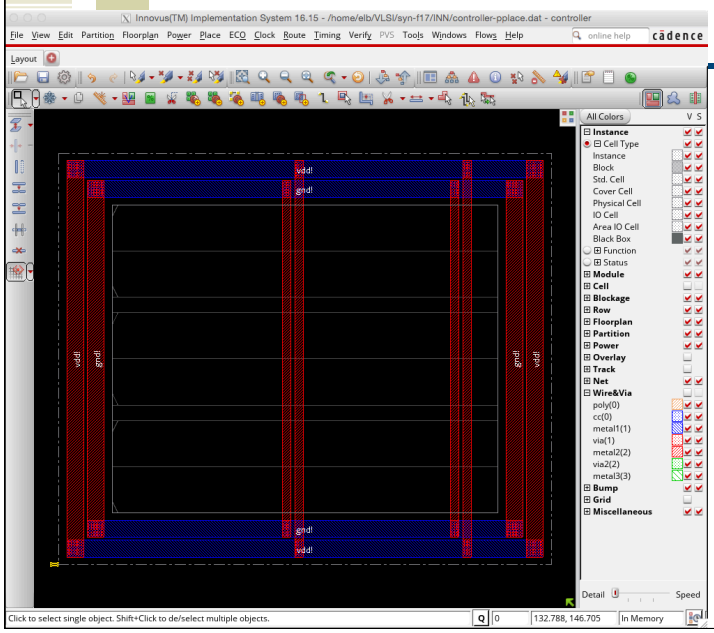
Annoying… This will start the stripes from 0 offset…



Power Stripes



Power Rings and Stripes

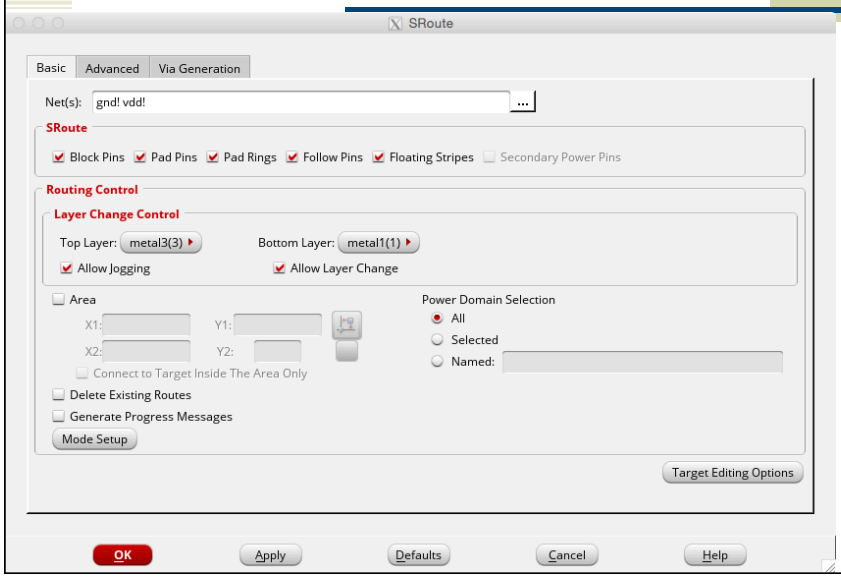


Power Rings and Stripes

Select and remove the leftmost power stripe…

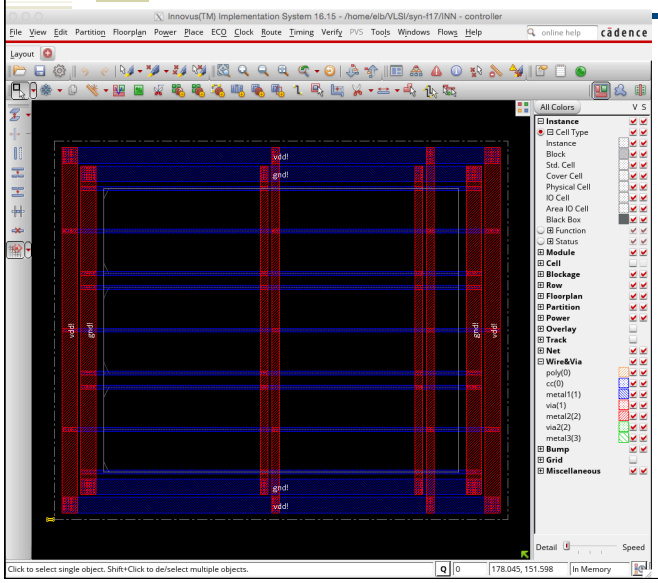
Route -> Sroute

Sroute to connect things up



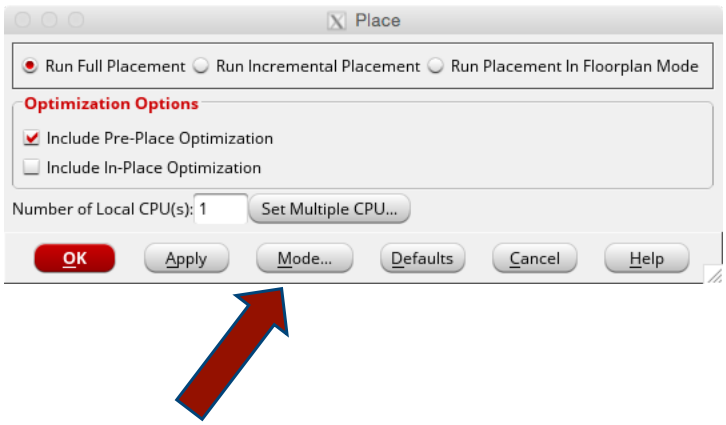
Route -> Sroute

Sroute to connect rows to power

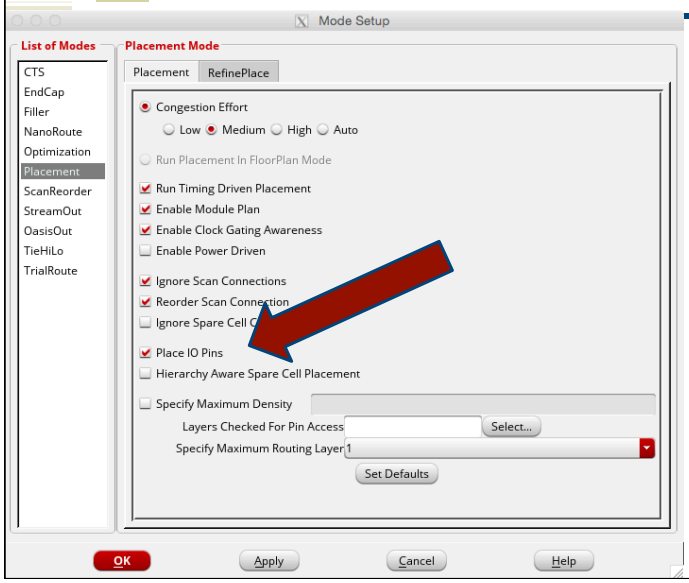


Place -> Place Standard Cell...

Place cells

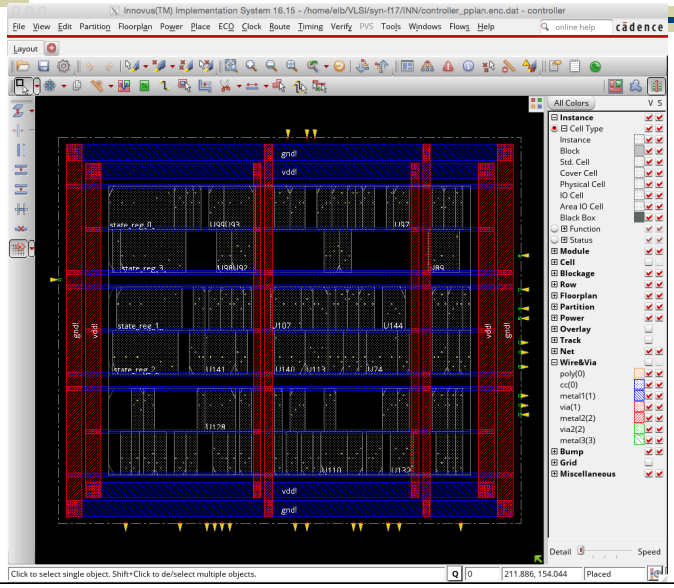


Place cells



Make sure to have your I/O signals placed!

Placed Cells…



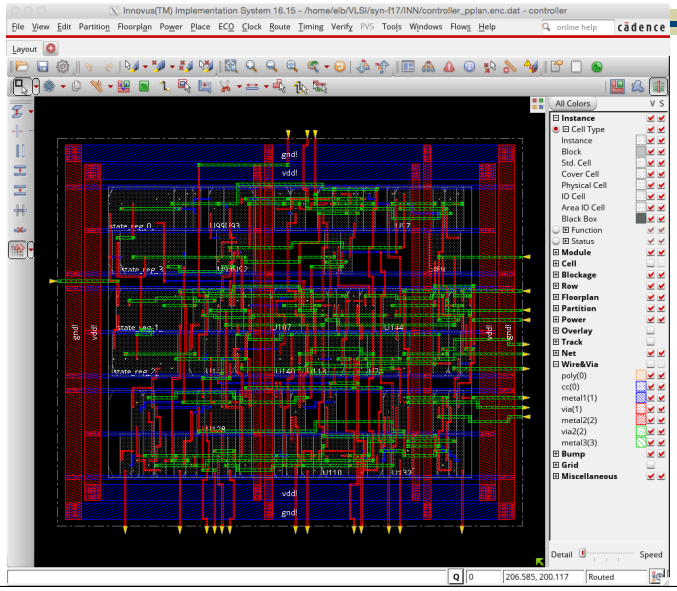
Clock Tree Synthesis…

- Probably don't have to do this explicitly
  - Synopsys has already generated a clock tree during synthesis…
  - Innovus can make a new one for you, and might have better information because of floorplan and placement…
  - But, seems like it can't be done at the GUI. Only through scripts…

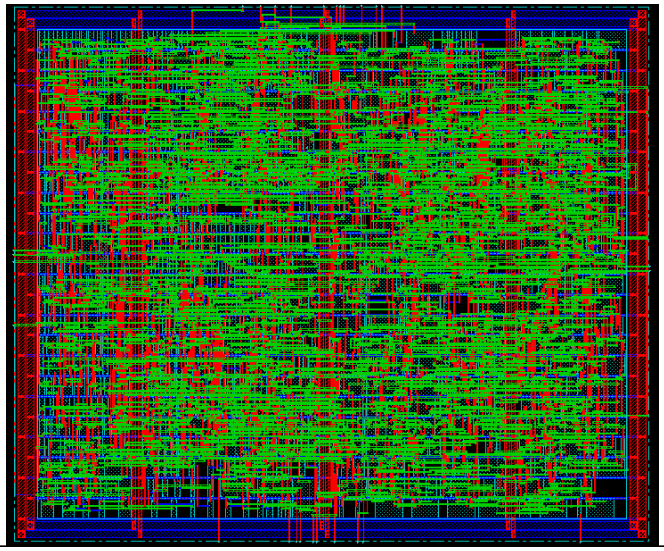
NanoRoute

Route -> NanoRoute -> Route

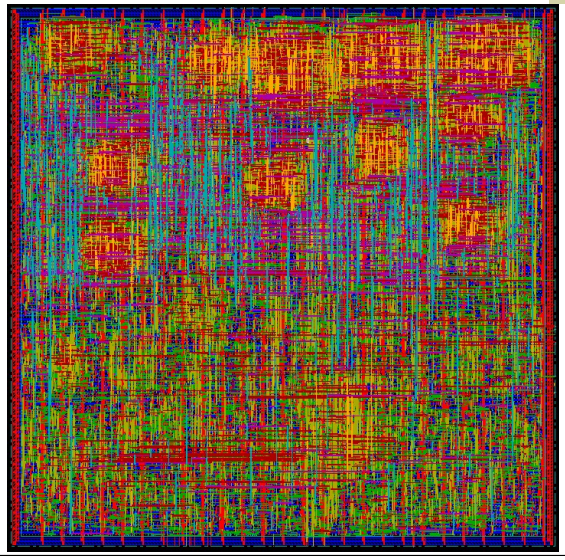
Routed circuit



Routed circuit – another example..

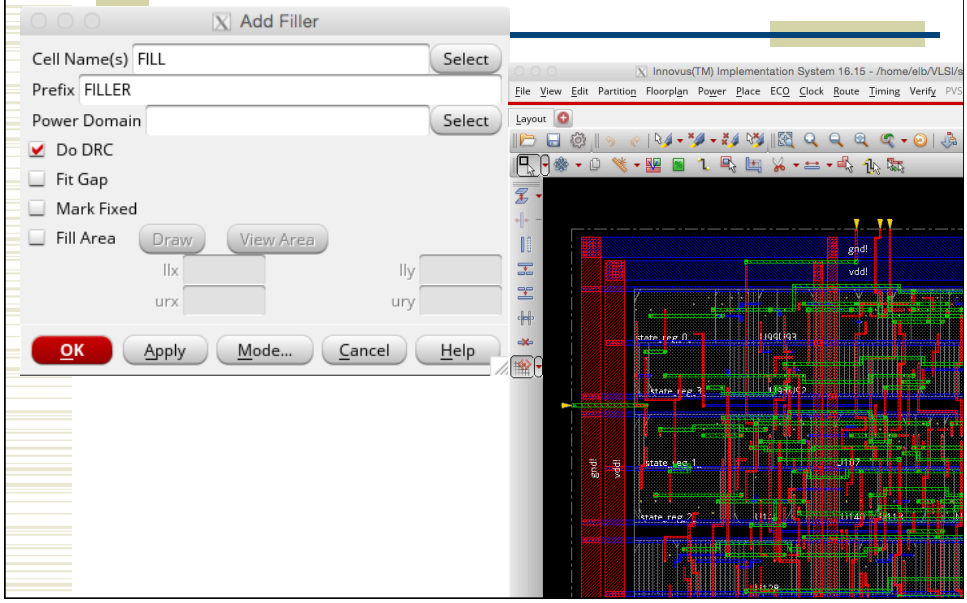


Routed circuit – yet another example..



Place -> Physical Cell -> Add Filler

Add Filler Cells



Verify connectivity

Verify Connectivity

Net Type
- All
- Regular Only
- Special Only

Nets
- All
- Selected
- Named:

Check
- Open
- UnConnected Pin
- Unrouted Net
- Connectivity Loop
- DanglingWire (Antenna)
- Weakly Connected Pin
- Geometry Loop
- Geometry Connectivity
- Keep Previous Results
- Divide Power Net
- Soft PG Connect
- Raw Violations Mark
- Use Virtual Connection
- TSV Die Abstract File

Verify Connectivity Report: controller.conn.rpt

Report Limits
- Error: 1000
- Warning: 50

Set Multiple CPU...

OK Apply Cancel Help

```
VERIFY_CONNECTIVITY use new engine.

******** Start: VERIFY CONNECTIVITY ********
Start Time: Fri Oct 13 15:41:54 2017

Design Name: controller
Database Units: 1000
Design Boundary: (0.0000, 0.0000) (288.0000, 240.0
Error Limit = 1000; Warning Limit = 50
Check all nets

Begin Summary
  Found no problems or warnings.
End Summary

End Time: Fri Oct 13 15:41:54 2017
Time Elapsed: 0:00:00.0

******** End: VERIFY CONNECTIVITY ********
  Verification Complete : 0 Viols.  0 Wrngs.
  (CPU Time: 0:00:00.0  MEM: 0.000M)

innovus 5>
```

Verify DRC (only wires!)

Verify DRC

Basic | Advanced

Verification Area
- Entire area
- Specify

X1: 0 Y1: 0
X2: 0 Y2: 0

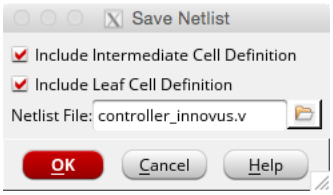
Layer Range: Bottom Layer: metal1(1) Top Layer: metal3(3)

Disable Rules
- Color
- Cut Spacing
- Enclosure
- EOL Spacing
- Min Area
- Min Cut
- Min Step
- Protrusion

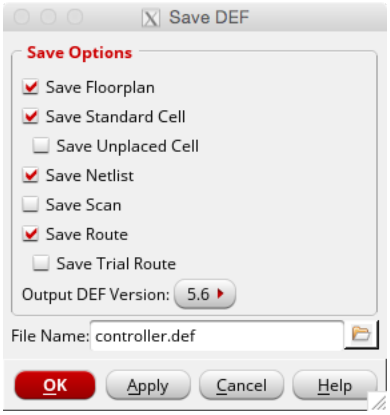
OK Apply Cancel Help

```
innovus 5> #-report controller.drc.rpt              # str
 *** Starting Verify DRC (MEM: 1545.3) ***

  VERIFY DRC ...... Starting Verification
  VERIFY DRC ...... Initializing
  VERIFY DRC ...... Deleting Existing Violations
  VERIFY DRC ...... Creating Sub-Areas
  VERIFY DRC ...... Using new threading
  VERIFY DRC ...... Sub-Area: {0.000 0.000 288.000 240.00
  VERIFY DRC ...... Sub-Area : 1 complete 0 Viols.

  Verification Complete : 0 Viols.

 *** End Verify DRC (CPU: 0:00:00.0  ELAPSED TIME: 0.00


innovus 5>
```

Write Results...



Design -> Save -> Netlist
(structural Verilog)



Design -> Save -> DEF
(layout information)

Innovus Scripting

- Usual warnings – know what's going on!
- Use top.tcl as a starting point
  - And the other .tcl files it calls...
- Innovus has a floorplanning stage that you may want to do by hand
  - write another script to read in the floorplan and go from there...
- Use innovus.cmd to see the text versions of what you did in the GUI...

Innovus Scritping Usage

- Need structural Verilog, struct.sdc, library.lib, library.lef
- Make a new dir for Innovus... (I call mine INN)
- Make an mmmc.tcl file with timing/library info
- <design>.globals has design-specific settings
  - use basename.globals as starting point.
- Usual warnings about scripting…
  - top.tcl and other *.tcl are in the class directory as starting points
  - /uusoc/facility/cad_common/local/class/6710/F17/cadence/Innovus
- Call with cad-inn, but this time source scripts instead of using GUI

Innovus Scripting Starting Point

```
[elb@lab2-21 INN]$ ls
6710.tcl  CAD6.lib           controller_struct.sdc  cts.tcl    mmmc.tcl   pplan.tcl  top.tcl
CAD6.lef  controller.globals controller_struct.v    fplan.tcl  place.tcl  route.tcl  verify.tcl
```

Note the same six files as before, but now adding <basename>.globals, and all the other .tcl files from /uusoc/facility/cad_common/local/class/6710/F17/cadence/Innovus

<basename>.globals

```
#
# Set the name of your structural Verlog file
# This comes from Synopsys synthesis
set init_verilog {!!your-file-name.v!!}
# Set the name of your top module
set init_design {!!your-top-module-name!!}
# Set the name of your .lef file
# This comes from ELC
set init_lef_file {!!your-file-name.lef!!}
…
```

<basename>.globals

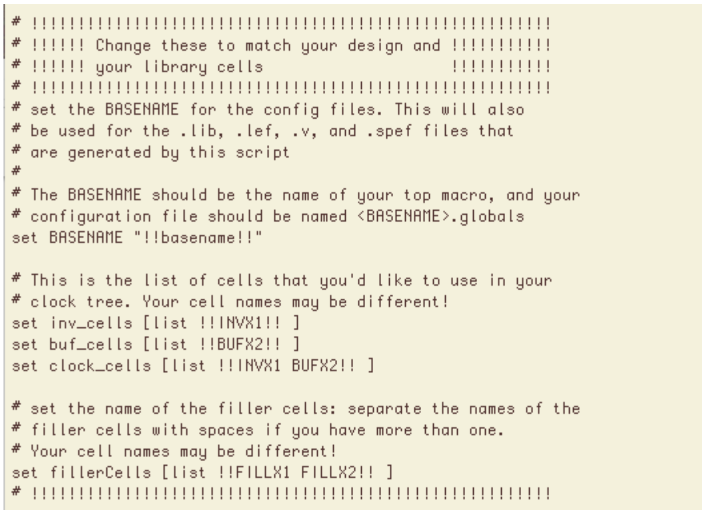
```
##################################################
#
# Set the name of your structural Verlog file
# This comes from Synopsys synthesis
set init_verilog {controller_struct.v}
# Set the name of your top module
set init_design {controller}
# Set the name of your .lef file
# This comes from Abstract
set init_lef_file {CAD6.lef}

##################################################
```

<basename>.globals

```
##################################################################
# below here you probably don't have to change anything
##################################################################
# Set the name of your "muli-mode-multi-corner data file
# You don't need to change this unless you're using a
# different mmmc.tcl file.
set init_mmmc_file {mmmc.tcl}
# Some helpful input mode settings
set init_import_mode {-treatUndefinedCellAsBbox 0 -keepEmptyModule 1 }
# Set the names of your gnd and power nets
set init_gnd_net {gnd!}
set init_pwr_net {vdd!}
```

---

top.tcl



```
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# !!!!!! Change these to match your design and !!!!!!!!!!!
# !!!!!! your library cells                    !!!!!!!!!!!
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# set the BASENAME for the config files. This will also
# be used for the .lib, .lef, .v, and .spef files that
# are generated by this script
#
# The BASENAME should be the name of your top macro, and your
# configuration file should be named <BASENAME>.globals
set BASENAME "!!basename!!"

# This is the list of cells that you'd like to use in your
# clock tree. Your cell names may be different!
set inv_cells [list !!INVX1!! ]
set buf_cells [list !!BUFX2!! ]
set clock_cells [list !!INVX1 BUFX2!! ]

# set the name of the filler cells: separate the names of the
# filler cells with spaces if you have more than one.
# Your cell names may be different!
set fillerCells [list !!FILLX1 FILLX2!! ]
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

top.tcl

```
# The following variables are used in fplan.tcl...
#
# These set the percent utilization target (how dense should
# the cells be placed), and the gap for routing between rows.
# These are good starting values for small macros. Larger or
# more complex macros will likely need a lower usepct or
# larger rowgap or both.
#
# Note that rowgap and coregap should be divisible by the basic
# grid unit of 0.3 that our ON Semi C5N 600nm process uses.
#
set usepct 0.70   ;# percent utilization in placing cells
set rowgap 9      ;# gap (microns) between pairs of std cell rows
set aspect 0.8    ;# aspect ratio of overall cell (1 is square,
                   # <1 is landscape, >1 is portrait
```

top.tcl

```
##################################################################
# You may not have to change things below this line - but check!
#
# You may want to do floorplanning by hand in which case you
# have some modification to do!
##################################################################

# Set some of the power and stripe parameters - you can change
# these if you like - in particular check the stripe space (sspace)
# and stripe offset (soffset)! These values should be divisible by 0.3
# so that they'll fall on the lambda grid
set pwidth 9.9     ;# power rail width
set pspace 1.8     ;# power rail space
set swidth 4.8     ;# power stripe width
set sspace 210     ;# power stripe spacing
set soffset 207    ;# power stripe offset to first stripe
```

top.tcl

```
#
# Set the flag for EDIto automatically figure out buf, inv, etc.
set dbgGPSAutoCellFunction 1

# Import design and floorplan
# If the config file is not named $basename.globals, edit this line.
source $BASENAME.globals
init_design
```

.globals file

- Same .globals file that we saw before with the walk-through
- Start with basename.globals and mmmc.tcl from the /uusoc/facility/cad_common/local/class/6710/F17/cadence/Innovus directory
- This describes the .lib, .lef, etc. information

top.tcl

```
# source the files that operate on the circuit
source fplan.tcl  ;# create the floorplan (might be done by hand...)
source pplan.tcl  ;# create the power rings and stripes
source place.tcl  ;# Place the cells and optimize (pre-CTS)
source cts.tcl    ;# Create the clock tree, and optimize (post-CTS)
source route.tcl  ;# Route the design using nanoRoute
source verify.tcl ;# Verify the design and produce output files
exit
```

fplan.tcl

```
puts "-------------Floorplanning---------------"
#
# Make a floorplan - this works fine for projects that are all
# standard cells and include no blocks that need hand placement...
setDrawView fplan
setFPlanRowSpacingAndType $rowgap 2
floorPlan -site core -r $aspect $usepct \
        $coregap $coregap $coregap $coregap
fit

#
# Save design so far
saveDesign ${BASENAME}_fplan.enc
saveFPlan  ${BASENAME}.fp
puts "--------------Floorplanning done----------
```

pplan.tcl

```
puts "-------------Power Planning----------------"
puts "-------Making power rings------------------"
#
# Make power and ground rings - $pwidth microns wide
# with $pspace spacing between them and centered in the channel
addRing -spacing_bottom $pspace \
        -width_left $pwidth \
        -width_bottom $pwidth \
        -width_top $pwidth \
        -spacing_top $pspace \
        -layer_bottom metal1 \
        -center 1 \
        -stacked_via_top_layer metal3 \
       ...
```

pplan.tcl

```
puts "------making power stripes-----------------"
# Make Power Stripes. This step is optional. If you keep it
# in remember to check the stripe spacing
# (set-to-set-distance = $sspace) and stripe offset
# (xleft-offset = $soffset))
addStripe -block_ring_top_layer_limit metal3 \
         -max_same_layer_jog_length 3.0 \
         -snap_wire_center_to_grid Grid \
         -padcore_ring_bottom_layer_limit metal1 \
         …
# Use the special-router to route the vdd! and gnd! nets
sroute -allowJogging 1

# Save the design so far
saveDesign ${BASENAME}_pplan.enc
puts "-------------Power Planning done---------"
```

top.tcl

Read the script...

- place
- pre-CTS optimization
- clock tree synthesis
- post-CTS optimization
- routing
- post-ROUTE optimization
- add filler
- write out results

Report Files

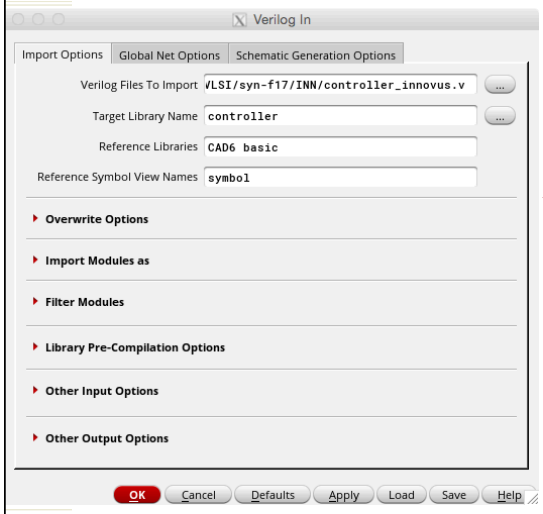
- <topname>_Conn_regular.rpt
- <topname>_Conn_special.rpt
- <topname>_Geom.rpt

- Want 0 violations in all
  - If you have 1 or 2 in the geometry you might be able to fix them easily in Virtuoso...

Read back to Virtuoso

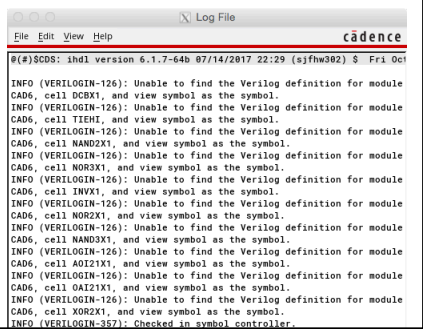
- Make a new library to hold the placed and routed version
- Commands to read Verilog and DEF are in the CIW, not Library Manager…
  - Once you have both schematic and layout, you can DRC-Extract-LVS to make sure it's all OK!

Import Verilog

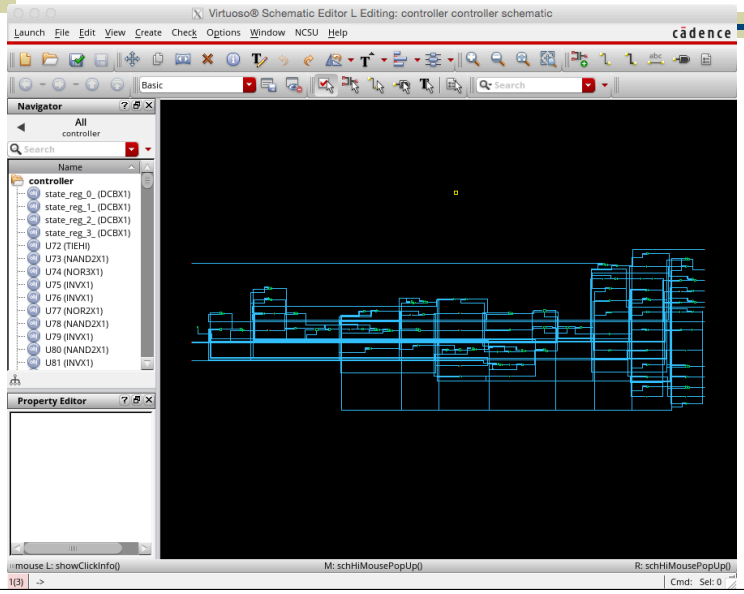


In CIW
File -> Import -> Verilog

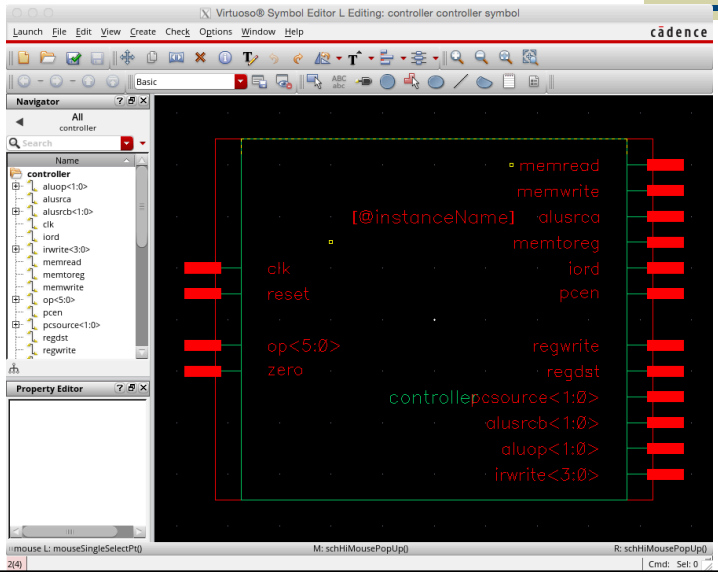
Make SURE you import The Verilog from Innovus!

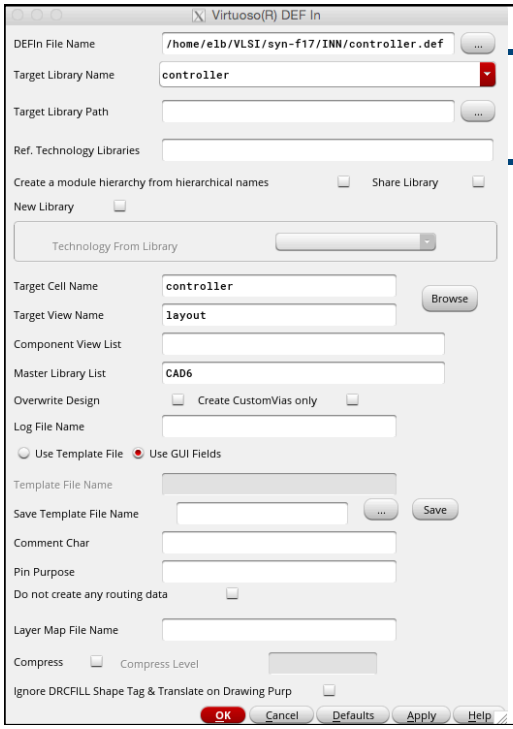


Schematic view



Symbol view

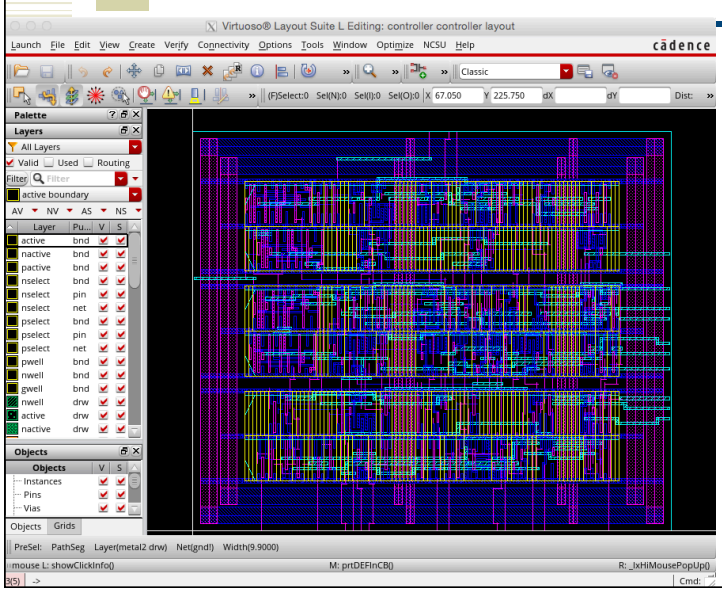




Read DEF

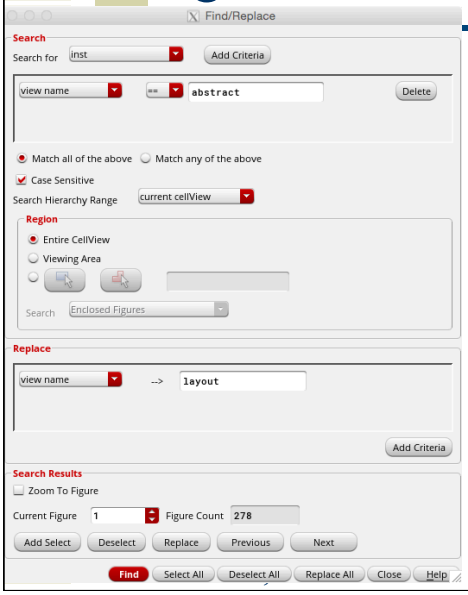
File -> Import -> DEF

Resulting layout view



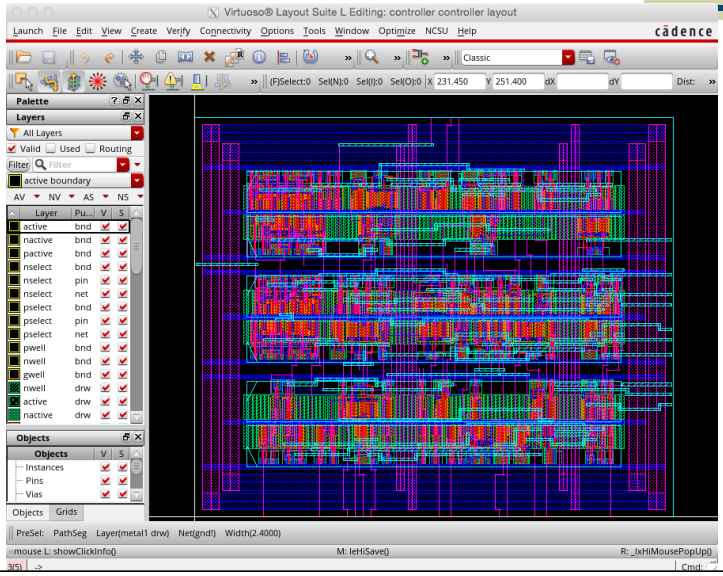
Problem: all cells are abstract views!

Change abstract to layout cellviews

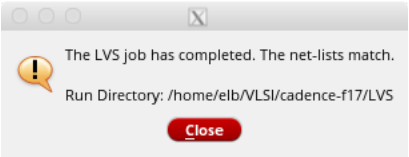


Edit -> Search

Change abstract to layout cellviews



LVS Result



Yay!!!

Summary

- Behavioral -> Structural -> Layout
- Can be automated by scripting, but make sure you know what you're doing
  - Synopsys documentation for design_compiler
  - innovus.cmd (and documentation) for Innovus
- End up with placed and routed core layout
  - or BLOCK for later use...