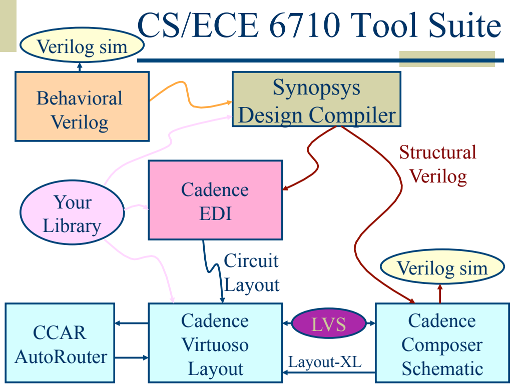
Synthesis and Place & Route

Synopsys design compiler

Cadence Encounter Digital Implementation System (EDI)

CS/ECE 6710 Tool Suite



Design Compiler

- Synthesis of behavioral to structural
- Three ways to go:
  1. Type commands to the design compiler shell
     - Start with syn-dc and start typing
  2. Write a script
     - Use syn-script.tcl as a starting point
  3. Use the Design Vision GUI
     - Friendly menus and graphics...

Design Compiler – Basic Flow

1. Define environment
   - target libraries – your cell library
   - synthetic libraries – DesignWare libraries
   - link-libraries – libraries to link against
2. Read in your structural Verilog
   - Usually split into analyze and elaborate
3. Set constraints
   - timing – define clock, loads, etc.

Design Compiler – Basic Flow

4. Compile the design
   - compile or compile_ultra
   - Does the actual synthesis
5. Write out the results
   - Make sure to change_names
   - Write out structural verilog, report, ddc, sdc files

beh2str – the simplest script!

```
# beh2str script
set target_library [list [getenv "LIBFILE"]]
set link_library [concat [concat "*" $target_library] $synthetic_library]
read_file -f verilog [getenv "INFILE"]
#/* This command will fix the problem of having */
#/* assign statements left in your structural file. */
set_fix_multiple_port_nets -all -buffer_constants
compile -ungroup_all
check_design
#/* always do change_names before write... */
redirect change_names { change_names -rules verilog -hierarchy -
    verbose }
write -f verilog -output [getenv "OUTFILE"]
quit
```

.synopsys_dc.setup

```
set SynopsysInstall [getenv "SYNOPSYS"]

set search_path [list . \
[format "%s%s"  $SynopsysInstall /libraries/syn] \
[format "%s%s"  $SynopsysInstall /dw/sim_ver] \
]
define_design_lib WORK -path ./WORK
set synthetic_library [list dw_foundation.sldb]
set synlib_wait_for_design_license [list "DesignWare-Foundation"]
set link_library [concat  [concat  "*" $target_library] $synthetic_library]
set symbol_library [list generic.sdb]
```

What beh2str leaves out...

- Timing!
  - No clock defined so no target speed
  - No input drive defined so assume infinite drive
  - No output load define so assume something

syn-script.tcl

- /uusoc/facility/cad_common/local/class/6710/F11/synopsys

```
#/* search path should include directories with memory .db files */
#/* as well as the standard cells */
set search_path [list . \
[format "%s%s" SynopsysInstall /libraries/syn] \
[format "%s%s" SynopsysInstall /dw/sim_ver] \
!!your-library-path-goes-here!!]
#/* target library list should include all target .db files */
set target_library [list !!your-library-name!!.db]
#/* synthetic_library is set in .synopsys_dc.setup to be */
#/* the dw_foundation library. */
set link_library [concat [concat "*" $target_library] $synthetic_library]
```

syn-script.tcl

```
#/* below are parameters that you will want to set for each design */
#/* list of all HDL files in the design */
set myFiles [list !!all-your-structural-Verilog-files!! ]
set fileFormat verilog          ;# verilog or VHDL
set basename !!basename!!       ;# Name of top-level module
set myClk !!clk!!               ;# The name of your clock
set virtual 0                   ;# 1 if virtual clock, 0 if real clock
#/* compiler switches... */
set useUltra 1                   ;# 1 for compile_ultra, 0 for compile
                                #mapEffort, useUngroup are for
                                #non-ultra compile...
set mapEffort1 medium           ;# First pass - low, medium, or high
set mapEffort2 medium           ;# second pass - low, medium, or
   high
set useUngroup 1                ;# 0 if no flatten, 1 if flatten
```

syn-script.tcl

```
#/* Timing and loading information */
set myPeriod_ns !!10!!          ;# desired clock period (sets speed goal)
set myInDelay_ns !!0.25!!       ;# delay from clock to inputs valid
set myOutDelay_ns !!0.25!!      ;# delay from clock to output valid
set myInputBuf !!INVX4!!        ;# name of cell driving the inputs
set myLoadLibrary !!Lib!!       ;# name of library the cell comes from
set myLoadPin !!A!!             ;# pin that outputs drive

#/* Control the writing of result files */
set runname struct ;# Name appended to output files
```

syn-script.tcl

```
#/* the following control which output files you want. They */
#/* should be set to 1 if you want the file, 0 if not */
set write_v 1    ;# compiled structural Verilog file
set write_ddc 0  ;# compiled file in ddc format
set write_sdf 0  ;# sdf file for back-annotated timing sim
set write_sdc 1  ;# sdc constraint file for place and route
set write_rep 1  ;# report file from compilation
set write_pow 0 ;# report file for power estimate
```

syn-script.tcl

```
# analyze and elaborate the files
analyze -format $fileFormat -lib WORK $myfiles
elaborate $basename -lib WORK -update
current_design $basename
# The link command makes sure that all the required design
# parts are linked together.
# The uniquify command makes unique copies of replicated modules.
link
uniquify
# now you can create clocks for the design
if { $virtual == 0 } {
    create_clock -period $myPeriod_ns $myClk
    } else {
    create_clock -period $myPeriod_ns -name $myClk
    }
```

syn-script.tcl

```
# Set the driving cell for all inputs except the clock
# The clock has infinite drive by default. This is usually
# what you want for synthesis because you will use other
# tools (like SOC Encounter) to build the clock tree (or define it by hand).
set_driving_cell -library $myLoadLibrary -lib_cell $myInputBuf \
    [remove_from_collection [all_inputs] $myClk]
# set the input and output delay relative to myclk
set_input_delay $myInDelay_ns -clock $myClk \
    [remove_from_collection [all_inputs] $myClk]
set_output_delay $myOutDelay_ns -clock $myClk [all_outputs]
# set the load of the circuit outputs in terms of the load
# of the next cell that they will drive, also try to fix hold time issues
set_load [load_of  [format “%s%s%s%s%s” $myLoadLibrary \
                               "/" $myInputBuf "/" $myLoadPin]] [all_outputs]
set_fix_hold $myClk
```

syn-script.tcl

```
# now compile the design with given mapping effort
# and do a second compile with incremental mapping
# or use the compile_ultra meta-command
if { $useUltra == 1 } {
    compile_ultra
    } else {
    if { $useUngroup == 1 } {
          compile -ungoup_all -map_effort $mapEffort1
          compile -incremental_mapping -map_effort $mapEffort2
          } else {
          compile -map_effort $mapEffort1
          compile -incremental_mapping -map_effort $mapEffort2
          }
    }
```

syn-script.tcl

```
# Check things for errors
check_design
report_constraint -all_violators
set filebase [format "%s%s%s" $basename "_"  $runname]
# structural (synthesized) file as verilog
if { $write_v == 1 } {
    set filename [format "%s%s" $filebase ".v"]
    redirect change_names { change_names -rules verilog \
                                  -hierarchy -verbose }
    write -format verilog -hierarchy -output $filename
}
# write the rest of the desired files... then quit
```

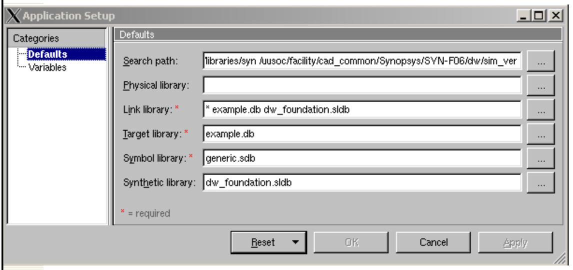
Using Scripts

- Modify syn-script.tcl or write your own
- syn-dc –f scriptname.tcl
- Make sure to check output!!!!

Using Design Vision

- You can do all of these commands from the design vision gui if you like
- syn-dv
- Follow the same steps as the script
  - Set libraries in your own .synopsys_dc.setup
  - analyze/elaborate
  - define clock and set constraints
  - compile
  - write out results

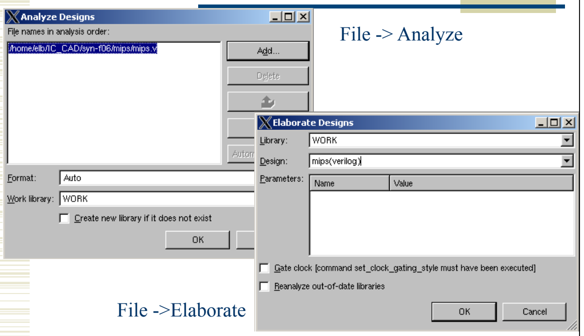
Setup



Search path: libraries/syn /uusoc/facility/cad_common/Synopsys/SYN-F06/dw/sim_ver
Physical library:
Link library: * * example.db dw_foundation.sldb
Target library: * example.db
Symbol library: * generic.sdb
Synthetic library: dw_foundation.sldb
* = required

File ->Setup

analyze/elaborate



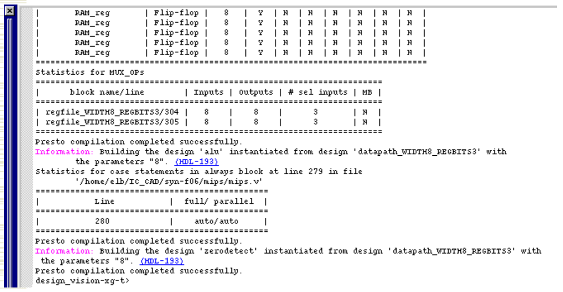
File names in analysis order: /home/elb/IC_CAD/syn-f06/mips/mips.v
Format: Auto
Work library: WORK
☐ Create new library if it does not exist

Library: WORK
Design: mips(verilog)
Parameters:
☐ Gate clock [command set_clock_gating_style must have been executed]
☐ Reanalyze out-of-date libraries

File -> Analyze

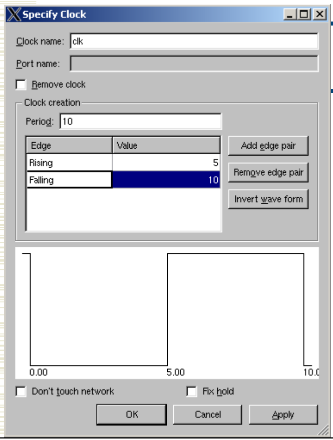
File ->Elaborate

Look at results...



```
|       RAM_reg        | Flip-flop |    8    |  Y  | N  | N  | N  | N  | N  | N  |
|       RAM_reg        | Flip-flop |    8    |  Y  | N  | N  | N  | N  | N  | N  |
|       RAM_reg        | Flip-flop |    8    |  Y  | N  | N  | N  | N  | N  | N  |
|       RAM_reg        | Flip-flop |    8    |  Y  | N  | N  | N  | N  | N  | N  |
|       RAM_reg        | Flip-flop |    8    |  Y  | N  | N  | N  | N  | N  | N  |
===================================================================================
Statistics for MUX_OPs
======================================================================
|      block name/line        | Inputs | Outputs | # sel inputs | MB |
======================================================================
| regfile_WIDTH8_REGBITS3/304 |    8   |    8    |      3       | N  |
| regfile_WIDTH8_REGBITS3/305 |    8   |    8    |      3       | N  |
======================================================================
Presto compilation completed successfully.
Information: Building the design 'alu' instantiated from design 'datapath_WIDTH8_REGBITS3' with
        the parameters "8". (HDL-193)
Statistics for case statements in always block at line 279 in file
        '/home/elb/IC_CAD/syn-f06/mips/mips.v'
===============================================
|           Line           |  full/ parallel  |
===============================================
|           280            |    auto/auto     |
===============================================
Presto compilation completed successfully.
Information: Building the design 'zerodetect' instantiated from design 'datapath_WIDTH8_REGBITS3' with
 the parameters "8". (HDL-193)
Presto compilation completed successfully.
design_vision-xg-t>
```

Define clock



Clock name: clk
Port name:
☐ Remove clock
Period: 10

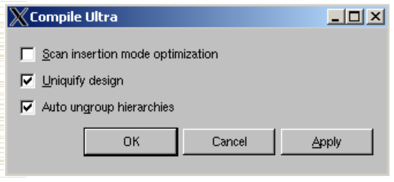
Edge	Value
Rising	5
Falling	10

☐ Don't touch network
☐ Fix hold

attributes -> specify clock

Also look at other attributes...

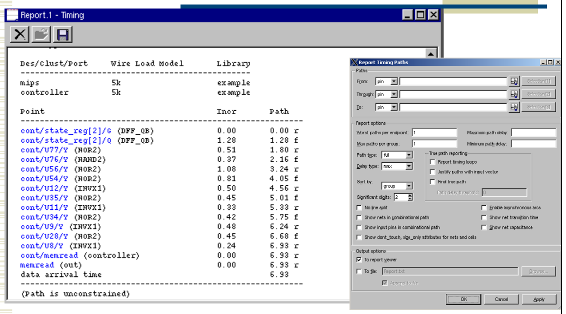
Compile



☐ Scan insertion mode optimization
☑ Uniquify design
☑ Auto ungroup hierarchies

Design -> Compile Ultra

Timing Reports

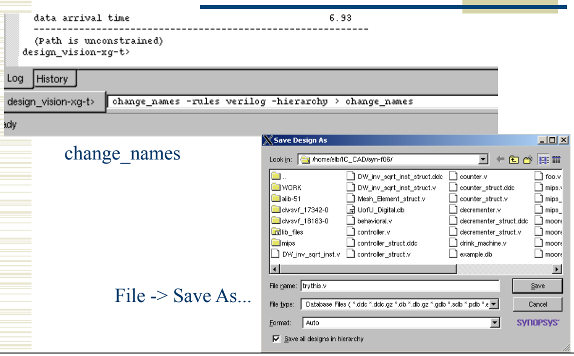


```
Des/Clust/Port     Wire Load Model       Library
------------------------------------------------
mips               5k                    example
controller         5k                    example

Point                                    Incr       Path
-----------------------------------------------------------
cont/state_reg[2]/G (DFF_QB)             0.00       0.00 r
cont/state_reg[2]/Q (DFF_QB)             1.28       1.28 f
cont/U77/Y (NOR2)                        0.51       1.80 r
cont/U76/Y (NAND2)                       0.37       2.16 f
cont/U56/Y (NOR2)                        1.08       3.24 r
cont/U54/Y (NOR2)                        0.81       4.05 f
cont/U12/Y (INVX1)                       0.50       4.56 r
cont/U35/Y (NOR2)                        0.45       5.01 f
cont/U11/Y (INVX1)                       0.33       5.33 r
cont/U34/Y (NOR2)                        0.42       5.75 f
cont/U9/Y (INVX1)                        0.48       6.24 r
cont/U28/Y (NOR2)                        0.45       6.68 f
cont/U8/Y (INVX1)                        0.24       6.93 r
cont/memread (controller)                0.00       6.93 r
memread (out)                            0.00       6.93 r
data arrival time                                   6.93
-----------------------------------------------------------
(Path is unconstrained)
```

From: pin
Through: pin
To: pin
Worst paths per endpoint: 1
Max paths per group: 1
Maximum path delay:
Minimum path delay:
Path type: full
Delay type: max
Sort by: group
Significant digits: 2
☐ Report timing loops
☐ Justify paths with input vector
☐ Find true path
Path delay threshold: 0
☐ No line split
☐ Show nets in combinational path
☐ Show input pins in combinational path
☐ Show dont_touch, size_only attributes for nets and cells
☐ Enable asynchronous arcs
☐ Show net transition time
☐ Show net capacitance
☑ To report viewer
☐ To file: Report.txt
☑ Append to file

Timing -> Report Timing Path

Write Results



```
  data arrival time                                        6.93
  ------------------------------------------------------------
  (Path is unconstrained)
design_vision-xg-t>
```

```
design_vision-xg-t> change_names -rules verilog -hierarchy > change_names
```

Save Design As

Look in: /home/elb/IC_CAD/syn-f06/

..	DW_inv_sqrt_inst_struct.ddc	counter.v	foo.v
WORK	DW_inv_sqrt_inst_struct.v	counter_struct.ddc	mips.v
alib-51	Mesh_Element_struct.v	counter_struct.v	mips_
dwsvf_17342-0	UofU_Digital.db	decrementer.v	mips_
dwsvf_18183-0	behavioral.v	decrementer_struct.ddc	moore
lib_files	controller.v	decrementer_struct.v	moore
mips	controller_struct.ddc	drink_machine.v	moore
DW_inv_sqrt_inst.v	controller_struct.v	example.db	moore

File name: trythis.v

File type: Database Files (*.ddc *.ddc.gz *.db *.db.gz *.gdb *.sdb *.pdb *.e

Format: Auto

[✓] Save all designs in hierarchy

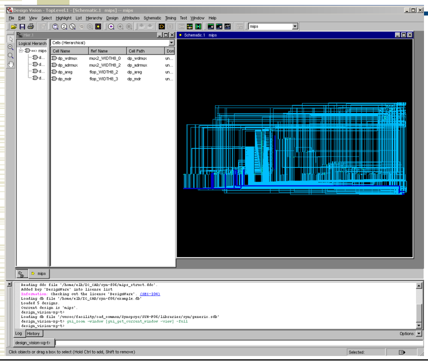
change_names

File -> Save As...

Or, use syn-dv after script...

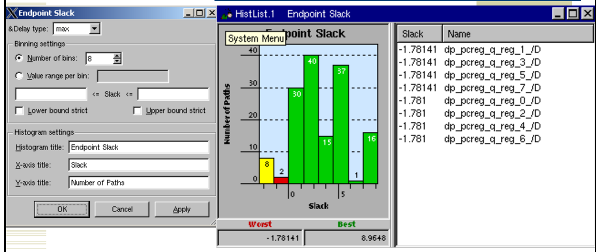
- syn-dc –f mips.tcl
- results in .v, .ddc, .sdc, .rep files
- Read the .ddc file into syn-dv and use it to explore timing...

syn-dv with mips_struct.v



File -> Read

Endpoint slack...



Endpoint Slack

&Delay type: max

Binning settings

(●) Number of bins: 8

( ) Value range per bin:

<= Slack <=

[ ] Lower bound strict [ ] Upper bound strict

Histogram settings

Histogram title: Endpoint Slack

X-axis title: Slack

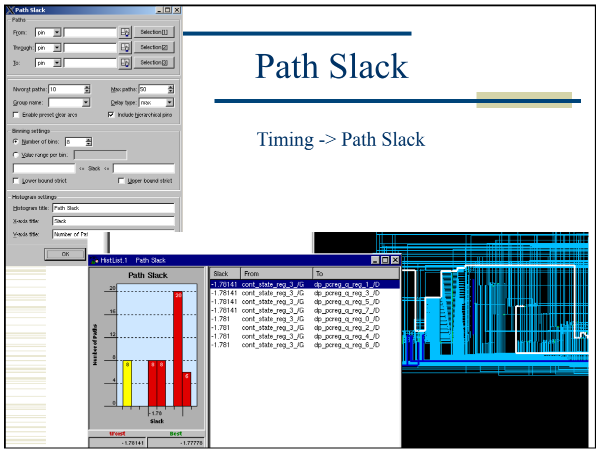
Y-axis title: Number of Paths

Worst -1.78141 Best 8.9648

Slack	Name
-1.78141	dp_pcreg_q_reg_1_/D
-1.78141	dp_pcreg_q_reg_3_/D
-1.78141	dp_pcreg_q_reg_5_/D
-1.78141	dp_pcreg_q_reg_7_/D
-1.781	dp_pcreg_q_reg_0_/D
-1.781	dp_pcreg_q_reg_2_/D
-1.781	dp_pcreg_q_reg_4_/D
-1.781	dp_pcreg_q_reg_6_/D

Timing -> Endpoint Slack

Path Slack



Path Slack

Paths

From: pin

Through: pin

To: pin

Nworst paths: 10 Max paths: 50

Group name: Delay type: max

[ ] Enable preset clear arcs [✓] Include hierarchical pins

Binning settings

(●) Number of bins: 8

( ) Value range per bin:

<= Slack <=

[ ] Lower bound strict [ ] Upper bound strict

Histogram settings

Histogram title: Path Slack

X-axis title: Slack

Y-axis title: Number of Pat

Worst -1.78141 Best -1.77778

Slack	From	To
-1.78141	cont_state_reg_3_/G	dp_pcreg_q_reg_1_/D
-1.78141	cont_state_reg_3_/G	dp_pcreg_q_reg_3_/D
-1.78141	cont_state_reg_3_/G	dp_pcreg_q_reg_5_/D
-1.78141	cont_state_reg_3_/G	dp_pcreg_q_reg_7_/D
-1.781	cont_state_reg_3_/G	dp_pcreg_q_reg_0_/D
-1.781	cont_state_reg_3_/G	dp_pcreg_q_reg_2_/D
-1.781	cont_state_reg_3_/G	dp_pcreg_q_reg_4_/D
-1.781	cont_state_reg_3_/G	dp_pcreg_q_reg_6_/D

Timing -> Path Slack

Encounter Digital Implementation System

- Need structural Verilog, .sdc, library.lib, library.lef
- make a new dir for edi...
- <design>.conf is also very helpful
  - use UofU_soc.conf as starting point.
  - And Default.view
- Usual warnings about scripting... UofU_edi.tcl is the generic script
  - .../local/class/6710/F11/cadence/EDI

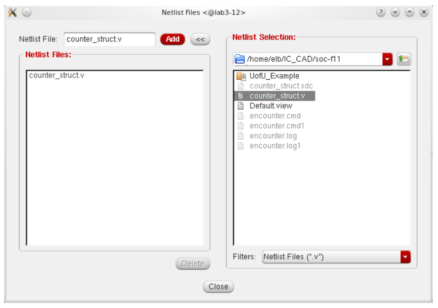
cad-edi Flow

1. Import Design
   - .v, .sdc, .lib, .lef – can put this in a file.conf and Default.view
2. Power plan
   - rings, stripes, row-routing (sroute)
3. Placement
   - place cells in the rows
4. Timing optimization – preCTS

cad-edi Flow

5. Synthesize clock tree
   - use your buf or inv footprint cells
6. timing optimization – postCTS
7. global routing
   - NanoRoute
8. timing optimization – postRoute
9. Add filler cells
10. Write out results
    - .def, _soc.v, .spef, .sdc, .lef

Design Import


Netlist Files <@lab3-12>

Netlist File: counter_struct.v

Netlist Files:
counter_struct.v

Netlist Selection:
/home/elb/IC_CAD/soc-f11

UofU_Example
counter_struct.sdc
counter_struct.v
Default.view
encounter.cmd
encounter.cmd1
encounter.log
encounter.log1

Filters: Netlist Files (*.v*)

Using a conf file

- Put the load information into a .conf file
- Load it up without having to re-type
- Also need a Default.view file

UofU_edi.conf

```
#
global rda_Input
#
#########################################################
# Here are the parts you need to update for your design
#########################################################
#
# Your input is structural verilog. Set the top module name
# and also give the .sdc file you used in synthesis for the
# clock timing constraints.
set rda_Input(ui_netlist)       {!!filename!!.v}
set rda_Input(ui_topcell)       {!!TopCellName!!}
set rda_Input(ui_timingcon_file) {!!filename!!.sdc}
```

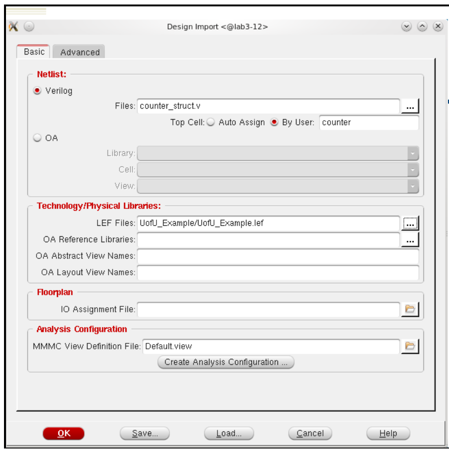
UofU_edi.conf

```
#
# Leave min and max empty if you have only one timing library
# (space-separated if you have more than one)
set rda_Input(ui_timelib)       {!!filename!!.lib}
set rda_Input(ui_timelib,min)   {}
set rda_Input(ui_timelib,max)   {}
#
#
# Set the name of your lef file or files
# (space-separated if you have more than one).
set rda_Input(ui_leffile) {!!filename!!.lef}
```

Default.view

```
# Version:1.0 MMMC View Definition File
# Do Not Remove Above Line
create_rc_corner -name typical -preRoute_res {1.0} -preRoute_cap {1.0} -
    preRoute_clkres {0.0} -preRoute_clkcap {0.0} -postRoute_res {1.0} -
    postRoute_cap {1.0} -postRoute_xcap {1.0} -postRoute_clkres {0.0} -
    postRoute_clkcap {0.0}
create_library_set -name 6710_typical -timing {!!your-lib-name.lib!!}
create_constraint_mode -name typical -sdc_files {!!your-cell-name.sdc!!}
create_delay_corner -name typical -library_set {6710_typical} -rc_corner {typical}
create_analysis_view -name typical -constraint_mode {typical} -delay_corner
    {typical}
set_analysis_view -setup {typical} -hold {typical}
```

Design Import


Design Import <@lab3-12>

Basic | Advanced

Netlist:
[✓ Verilog]
Files: counter_struct.v
Top Cell: Auto Assign [✓ By User:] counter
OA
Library:
Cell:
View:

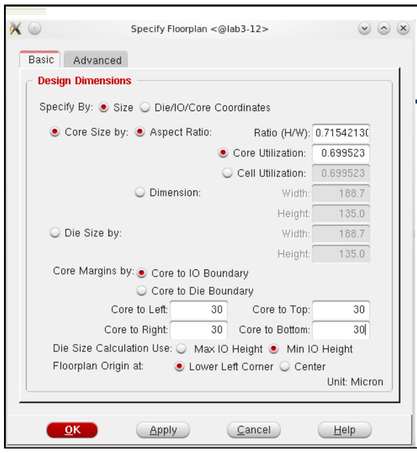
Technology/Physical Libraries:
LEF Files: UofU_Example/UofU_Example.lef
OA Reference Libraries:
OA Abstract View Names:
OA Layout View Names:

Floorplan
IO Assignment File:

Analysis Configuration
MMMC View Definition File: Default.view
Create Analysis Configuration ...

OK | Save... | Load... | Cancel | Help

Floorplan


Specify Floorplan <@lab3-12>

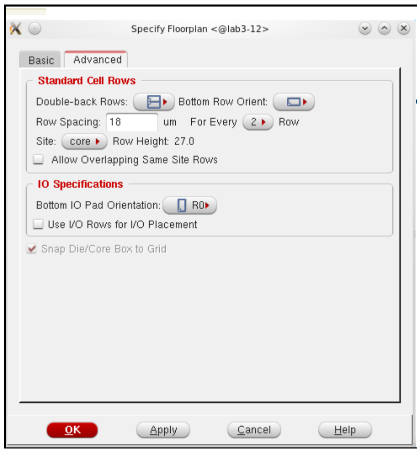
Basic | Advanced

Design Dimensions
Specify By: [✓ Size] Die/IO/Core Coordinates
[✓ Core Size by:] [✓ Aspect Ratio:] Ratio (H/W): 0.71542130
[✓ Core Utilization:] 0.699523
Cell Utilization: 0.699523
Dimension: Width: 188.7 Height: 135.0
Die Size by: Width: 188.7 Height: 135.0
Core Margins by: [✓ Core to IO Boundary] Core to Die Boundary
Core to Left: 30 Core to Top: 30
Core to Right: 30 Core to Bottom: 30
Die Size Calculation Use: Max IO Height [✓ Min IO Height]
Floorplan Origin at: [✓ Lower Left Corner] Center
Unit: Micron

OK | Apply | Cancel | Help

Specify -> Floorplan

Floorplan


Specify Floorplan <@lab3-12>

Basic | Advanced

Standard Cell Rows
Double-back Rows: Bottom Row Orient:
Row Spacing: 18 um For Every 2 Row
Site: core Row Height: 27.0
Allow Overlapping Same Site Rows

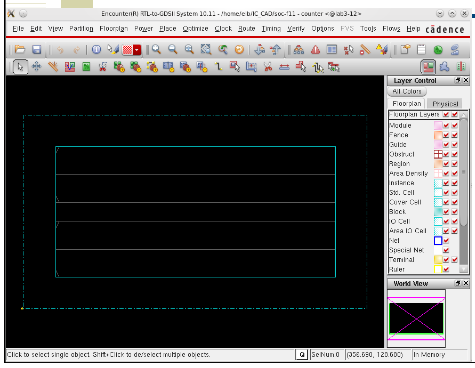
IO Specifications
Bottom IO Pad Orientation: R0
Use I/O Rows for I/O Placement

[✓ Snap Die/Core Box to Grid]

OK | Apply | Cancel | Help

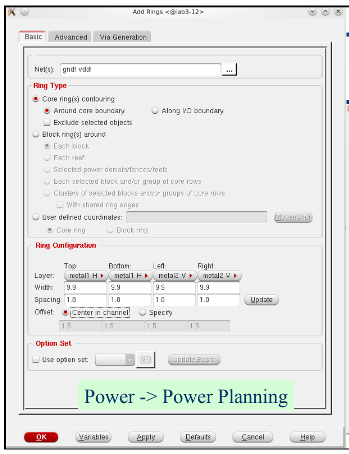
Specify -> Floorplan

Floorplan


Encounter(R) RTL-to-GDSII System 10.11 - /home/elb/IC_CAD/soc-f11 - counter <@lab3-12>
File Edit View Partition Floorplan Power Place Optimize Clock Route Timing Verify Options PVS Tools Flows Help cādence
Layer Control
All Colors
Floorplan Physical
Floorplan Layers
Module
Fence
Guide
Obstruct
Region
Area Density
Instance
Std. Cell
Cover Cell
Block
IO Cell
Area IO Cell
Net
Special Net
Terminal
Ruler
World View
Click to select single object. Shift+Click to de/select multiple objects.
Q
SelNum:0
(356.690, 128.680)
In Memory

Specify -> Floorplan

Power Rings and Stripes


Add Rings <@lab3-12>

Basic | Advanced | Via Generation

Net(s): gnd! vdd!

Ring Type
[✓ Core ring(s) contouring]
[✓ Around core boundary] Along I/O boundary
Exclude selected objects
Block ring(s) around
Each block
Each reef
Selected power domain/fences/reefs
Each selected block and/or group of core rows
Clusters of selected blocks and/or groups of core rows
With shared ring edges
User defined coordinates: MouseClick
Core ring Block ring

Ring Configuration

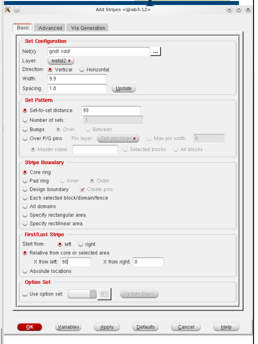
	Top:	Bottom:	Left:	Right:
Layer:	metal1 H	metal1 H	metal2 V	metal2 V
Width:	9.9	9.9	9.9	9.9
Spacing:	1.8	1.8	1.8	1.8
Offset:	1.5	1.5	1.5	1.5

Update
Offset: [✓ Center in channel] Specify

Option Set
Use option set: Update Basic

Power -> Power Planning

OK | Variables | Apply | Defaults | Cancel | Help


Add Stripes <@lab3-12>

Basic | Advanced | Via Generation

Set Configuration
Net(s): gnd! vdd!
Layer: metal2
Direction: [✓ Vertical] Horizontal
Width: 9.9
Spacing: 1.8 Update

Set Pattern
[✓ Set-to-set distance:] 99
Number of sets: 1
Bumps Over Between
Over P/G pins Pin layer: Top pin layer Max pin width: 0
Master name: Selected blocks All blocks

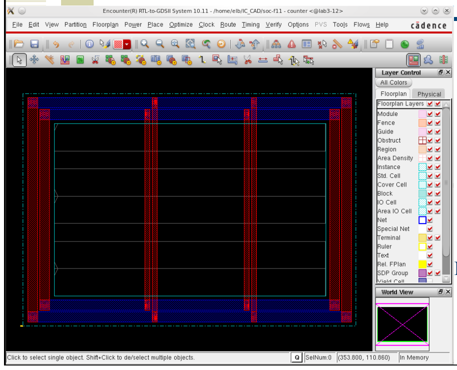
Stripe Boundary
[✓ Core ring]
Pad ring Inner Outer
Design boundary Create pins
Each selected block/domain/fence
All domains
Specify rectangular area
Specify rectilinear area

First/Last Stripe
Start from: [✓ left] right
[✓ Relative from core or selected area]
X from left: 90 X from right: 0
Absolute locations

Option Set
Use option set: Update Basic

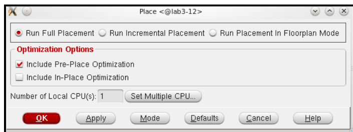
OK | Variables | Apply | Defaults | Cancel | Help

Sroute to connect things up


Encounter(R) RTL-to-GDSII System 10.11 - /home/elb/IC_CAD/soc-f11 - counter <@lab3-12>
File Edit View Partition Floorplan Power Place Optimize Clock Route Timing Verify Options PVS Tools Flows Help
cadence
Layer Control
All Colors
Floorplan Physical
Floorplan Layers
Module
Fence
Guide
Obstruct
Region
Area Density
Instance
Std. Cell
Cover Cell
Block
IO Cell
Area IO Cell
Net
Special Net
Terminal
Ruler
Text
Rel. FPlan
SDP Group
World View
Click to select single object. Shift+Click to de/select multiple objects.
SelNum:0
(353.800, 110.860)
In Memory

Route -> Sroute

Place cells



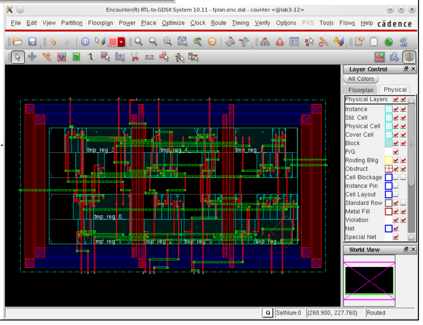
[✓ Run Full Placement] Run Incremental Placement Run Placement In Floorplan Mode

Optimization Options

- [✓] Include Pre-Place Optimization
- [ ] Include In-Place Optimization

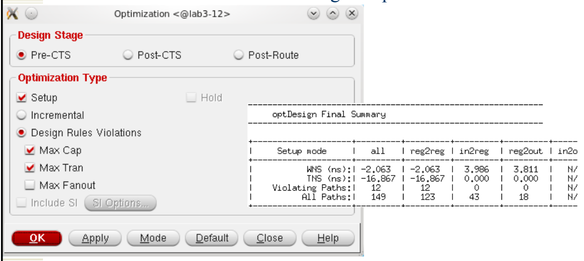
Number of Local CPU(s): 1 Set Multiple CPU...

Place -> Place cells...



pre-CTS timing optimization

Timing -> Optimization



Design Stage

[✓ Pre-CTS] Post-CTS Post-Route

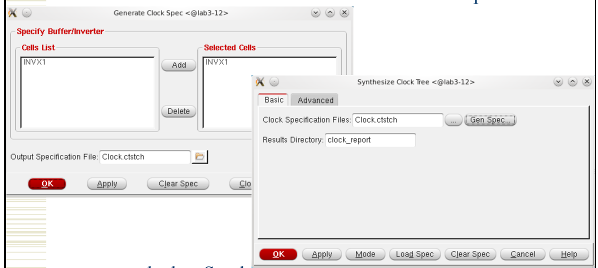
Optimization Type

- [✓] Setup
- [ ] Hold
- Incremental
- [✓ Design Rules Violations]
  - [✓] Max Cap
  - [✓] Max Tran
  - [ ] Max Fanout
- [ ] Include SI SI Options...

```
------------------------------------------------------------
     optDesign Final Summary
------------------------------------------------------------

+--------------------+---------+---------+---------+---------+-----
|     Setup mode     |   all   | reg2reg | in2reg  | reg2out | in2o
+--------------------+---------+---------+---------+---------+-----
|           WNS (ns):| -2.063  | -2.063  |  3.986  |  3.811  |   N/
|           TNS (ns):| -16.867 | -16.867 |  0.000  |  0.000  |   N/
|    Violating Paths:|   12    |   12    |    0    |    0    |   N/
|          All Paths:|   149   |   123   |   43    |   18    |   N/
+--------------------+---------+---------+---------+---------+-----
```

Clock Tree Synthesis

clock -> create clock tree spec



Specify Buffer/Inverter

Cells List	Selected Cells
INVX1	INVX1

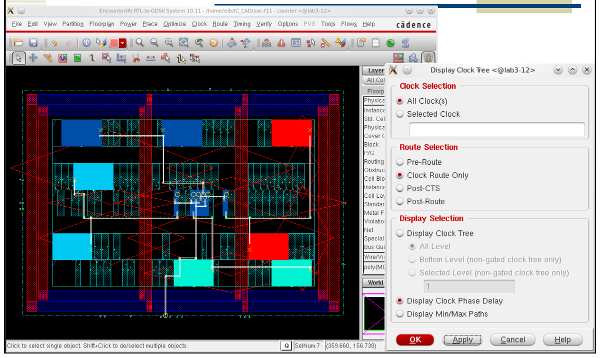
Output Specification File: Clock.ctstch

Clock Specification Files: Clock.ctstch

Results Directory: clock_report

clock ->Synthesize clock tree

Display Clock Tree


Encounter(R) RTL-to-GDSII System 10.11 - /home/elb/IC_CAD/soc-f11 - counter <@lab3-12>
File Edit View Partition Floorplan Power Place Optimize Clock Route Timing Verify Options PVS Tools Flows Help
cadence
Click to select single object. Shift+Click to de/select multiple objects.
SelNum:7
(359.660, 156.730)
Display Clock Tree <@lab3-12>
OK
Apply
Cancel
Help

Clock Selection

- [✓ All Clock(s)]
- Selected Clock

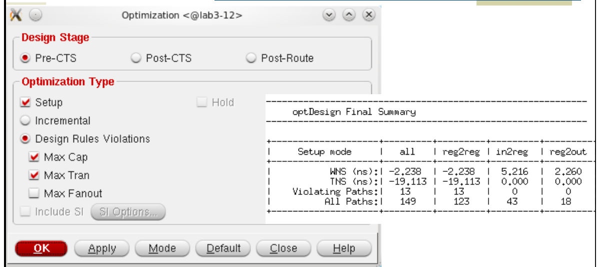
Route Selection

- Pre-Route
- [✓ Clock Route Only]
- Post-CTS
- Post-Route

Display Selection

- Display Clock Tree
  - [✓ All Level]
  - Bottom Level (non-gated clock tree only)
  - Selected Level (non-gated clock tree only) 1
- [✓ Display Clock Phase Delay]
- Display Min/Max Paths

post-CTS optimization



Design Stage

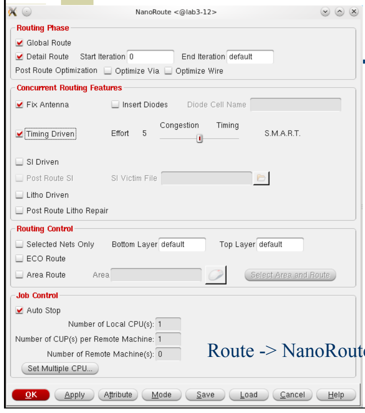
[✓ Pre-CTS] Post-CTS Post-Route

Optimization Type

- [✓] Setup
- [ ] Hold
- Incremental
- [✓ Design Rules Violations]
  - [✓] Max Cap
  - [✓] Max Tran
  - [ ] Max Fanout
- [ ] Include SI SI Options...

```
------------------------------------------------------------
     optDesign Final Summary
------------------------------------------------------------

+--------------------+---------+---------+---------+---------
|     Setup mode     |   all   | reg2reg | in2reg  | reg2out
+--------------------+---------+---------+---------+---------
|           WNS (ns):| -2.238  | -2.238  |  5.216  |  2.260
|           TNS (ns):| -19.113 | -19.113 |  0.000  |  0.000
|    Violating Paths:|   13    |   13    |    0    |    0
|          All Paths:|   149   |   123   |   43    |   18
+--------------------+---------+---------+---------+---------
```

NanoRoute



NanoRoute <@lab3-12>

Routing Phase
- [✓] Global Route
- [✓] Detail Route — Start Iteration: 0 — End Iteration: default
- Post Route Optimization — [ ] Optimize Via — [ ] Optimize Wire

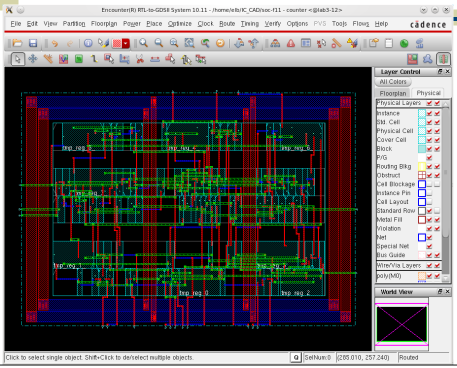
Concurrent Routing Features
- [✓] Fix Antenna — [ ] Insert Diodes — Diode Cell Name:
- [✓] Timing Driven — Effort 5 — Congestion / Timing — S.M.A.R.T.
- [ ] SI Driven
- [ ] Post Route SI — SI Victim File:
- [ ] Litho Driven
- [ ] Post Route Litho Repair

Routing Control
- [ ] Selected Nets Only — Bottom Layer: default — Top Layer: default
- [ ] ECO Route
- [ ] Area Route — Area: — Select Area and Route

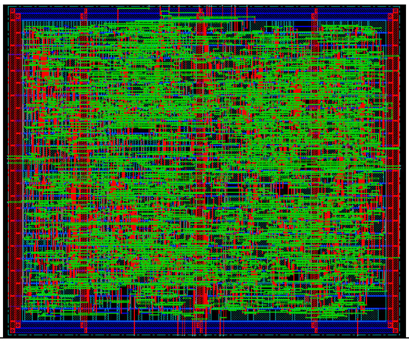
Job Control
- [✓] Auto Stop
- Number of Local CPU(s): 1
- Number of CUP(s) per Remote Machine: 1
- Number of Remote Machine(s): 0
- Set Multiple CPU...

Route -> NanoRoute -> Route

Routed circuit

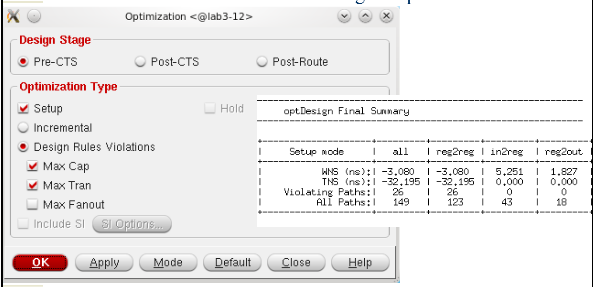


Routed circuit



postRoute optimization

Timing -> Optimization



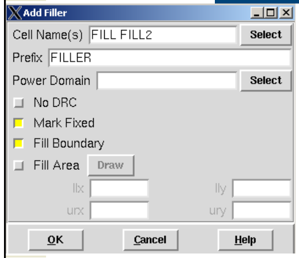
Optimization <@lab3-12>

Design Stage
[● Pre-CTS] ○ Post-CTS ○ Post-Route

Optimization Type
- [✓] Setup — [ ] Hold
- ○ Incremental
- ● Design Rules Violations
  - [✓] Max Cap
  - [✓] Max Tran
  - [ ] Max Fanout
- [ ] Include SI — SI Options...

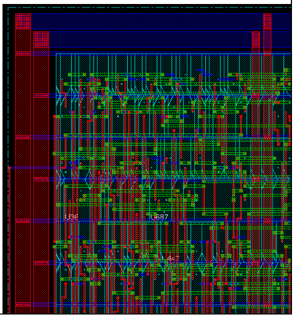
```
---------------------------------------------------------------
     optDesign Final Summary
---------------------------------------------------------------

+--------------------+----------+----------+----------+----------+
|     Setup mode     |   all    | reg2reg  |  in2reg  | reg2out  |
+--------------------+----------+----------+----------+----------+
|           WNS (ns):| -3.080   | -3.080   |  5.251   |  1.827   |
|           TNS (ns):| -32.195  | -32.195  |  0.000   |  0.000   |
|    Violating Paths:|   26     |   26     |    0     |    0     |
|          All Paths:|   149    |   123    |   43     |   18     |
+--------------------+----------+----------+----------+----------+
```

Add Filler



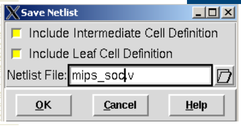
Add Filler

- Cell Name(s): FILL FILL2 — Select
- Prefix: FILLER
- Power Domain: — Select
- [ ] No DRC
- [✓] Mark Fixed
- [✓] Fill Boundary
- [ ] Fill Area — Draw
- llx: — lly:
- urx: — ury:



Place -> Filler -> Add...

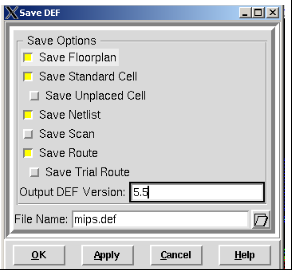
Write Results...



Save Netlist

- [✓] Include Intermediate Cell Definition
- [✓] Include Leaf Cell Definition
- Netlist File: mips_soc.v

Design -> Save -> Netlist



Save DEF

Save Options
- [✓] Save Floorplan
- [✓] Save Standard Cell
  - [ ] Save Unplaced Cell
- [✓] Save Netlist
- [ ] Save Scan
- [✓] Save Route
  - [ ] Save Trial Route
- Output DEF Version: 5.5
- File Name: mips.def

Design -> Save -> DEF

Encounter Scripting

- Usual warnings – know what's going on!
- Use top.tcl as a starting point
  - And the other .tcl files it calls...
- EDI 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 encounter.cmd to see the text versions of what you did in the GUI...

top.tcl

```
# set the basename for the config and floorplan files. This
# will also be used for the .lib, .lef, .v, and .spef files...
set basename “mips"


# set the name of the filler cells - you don't need a list
# if you only have one
set fillerCells FILL
#set fillerCells [list FILL FILL2]
```

top.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 lowered usepct or
#  larger rowgap or both.
set usepct 0.65 ;# percent utilization in placing cells
set rowgap 30   ;# gap between pairs of std cell rows

# "aspect" sets the shape of the floorplan: less than 1.0
# is landscape, greater than 1.0 is portrait, 1.0 is square
set aspect 0.60  ;# aspect ratio of overall cell
```

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)!
set pwidth 9.9     ;# power rail width
set pspace 1.8     ;# power rail space
set swidth 4.8     ;# power stripe width
set sspace 123     ;# power stripe spacing
set soffset 120    ;# power stripe offset to first stripe
set coregap 30.0  ;# gap between the core and the power rails
```

top.tcl

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

# Import design and floorplan
# If the config file is not named $basename.conf, edit this line.
loadConfig $basename.conf 0
commitConfig
```

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 "-------------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 \
       ...
#
# 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

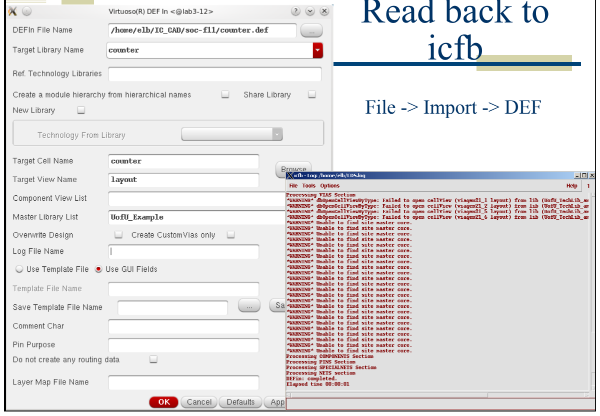
Read the script...
- place
- pre-CTS optimization
- clock tree synthesis
- post-CTS optimization
- routing
- post-ROUTE optimization
- add filler
- write out results

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 icfb

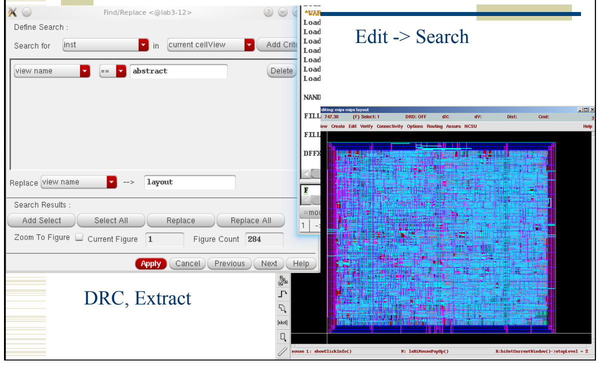


DEFIn File Name: /home/elb/IC_CAD/soc-f11/counter.def
Target Library Name: counter
Ref. Technology Libraries:
Create a module hierarchy from hierarchical names ☐ Share Library ☐
New Library ☐
Technology From Library
Target Cell Name: counter
Target View Name: layout
Component View List:
Master Library List: UofU_Example
Overwrite Design ☐ Create CustomVias only ☐
Log File Name:
Use Template File [✓ Use GUI Fields]
Template File Name:
Save Template File Name:
Comment Char:
Pin Purpose:
Do not create any routing data ☐
Layer Map File Name:

```
Processing VIAS Section
*WARNING* dbOpenCellViewByType: Failed to open cellView (viagen21_1 layout) from lib (UofU_TechLib_am
*WARNING* dbOpenCellViewByType: Failed to open cellView (viagen21_2 layout) from lib (UofU_TechLib_am
*WARNING* dbOpenCellViewByType: Failed to open cellView (viagen21_5 layout) from lib (UofU_TechLib_am
*WARNING* dbOpenCellViewByType: Failed to open cellView (viagen21_6 layout) from lib (UofU_TechLib_am
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
*WARNING* Unable to find site master core.
Processing COMPONENTS Section
Processing PINS Section
Processing SPECIALNETS Section
Processing NETS section
DEFin: completed.
Elapsed time 00:00:01
```

File -> Import -> DEF

Change abstract to layout cellviews



Find/Replace <@lab3-12>

Define Search :

Search for: inst in current cellView

view name == abstract

Replace view name --> layout

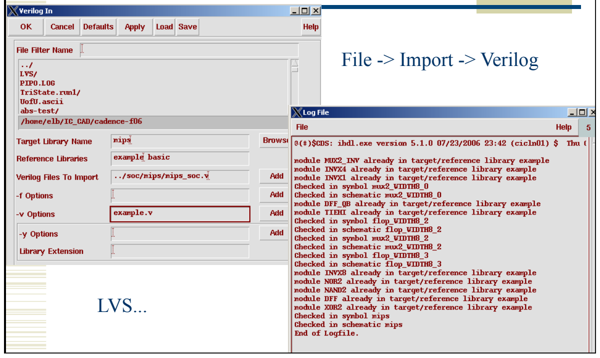
Search Results :

Zoom To Figure ☐ Current Figure 1 Figure Count 284

Edit -> Search

DRC, Extract

Import Verilog



Verilog In

File Filter Name

```
../
LVS/
PIPO.LOG
TriState.run1/
UofU.ascii
abs-test/
/home/elb/IC_CAD/cadence-f06
```

Target Library Name: mips

Reference Libraries: example basic

Verilog Files To Import: ../soc/mips/mips_soc.v

-f Options:

-v Options: example.v

-y Options:

Library Extension:

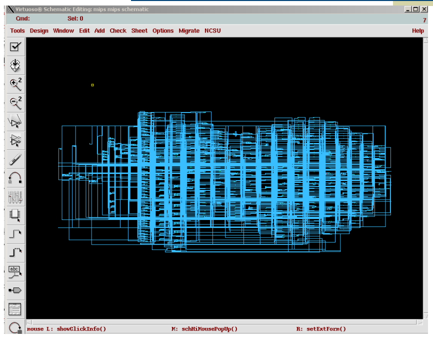
Log File

```
@(#)$CDS: ihdl.exe version 5.1.0 07/23/2006 23:42 (cicln01) $  Thu (

module MUX2_INV already in target/reference library example
module INVX4 already in target/reference library example
module INVX1 already in target/reference library example
Checked in symbol mux2_WIDTH8_0
Checked in schematic mux2_WIDTH8_0
module DFF_QB already in target/reference library example
module TIEHI already in target/reference library example
Checked in symbol flop_WIDTH8_2
Checked in schematic flop_WIDTH8_2
Checked in symbol mux2_WIDTH8_2
Checked in schematic mux2_WIDTH8_2
Checked in symbol flop_WIDTH8_3
Checked in schematic flop_WIDTH8_3
module INVX8 already in target/reference library example
module NOR2 already in target/reference library example
module NAND2 already in target/reference library example
module DFF already in target/reference library example
module XOR2 already in target/reference library example
Checked in symbol mips
Checked in schematic mips
End of Logfile.
```

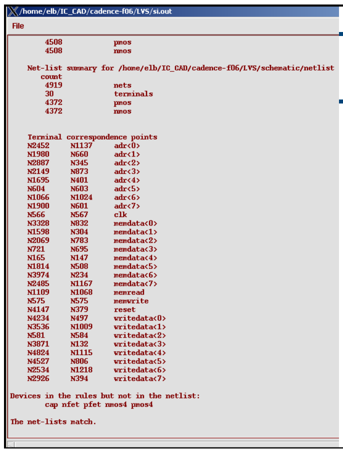
File -> Import -> Verilog

LVS…

Schematic view



LVS Result



/home/elb/IC_CAD/cadence-f06/LVS/si.out

```
        4508            pmos
        4508            nmos

    Net-list summary for /home/elb/IC_CAD/cadence-f06/LVS/schematic/netlist
       count
        4919            nets
        30              terminals
        4372            pmos
        4372            nmos


    Terminal correspondence points
    N2452     N1137     adr<0>
    N1980     N660      adr<1>
    N2887     N345      adr<2>
    N2149     N873      adr<3>
    N1695     N401      adr<4>
    N604      N603      adr<5>
    N1066     N1024     adr<6>
    N1900     N601      adr<7>
    N566      N567      clk
    N3328     N832      memdata<0>
    N1598     N304      memdata<1>
    N2069     N783      memdata<2>
    N721      N695      memdata<3>
    N165      N147      memdata<4>
    N1814     N508      memdata<5>
    N3974     N234      memdata<6>
    N2485     N1167     memdata<7>
    N1109     N1068     memread
    N575      N575      memwrite
    N4147     N379      reset
    N4234     N497      writedata<0>
    N3536     N1009     writedata<1>
    N581      N584      writedata<2>
    N3871     N132      writedata<3>
    N4824     N1115     writedata<4>
    N4527     N806      writedata<5>
    N2534     N1218     writedata<6>
    N2926     N394      writedata<7>

Devices in the rules but not in the netlist:
        cap nfet pfet nmos4 pmos4

The net-lists match.
```

Yay!

Summary

- Behavioral -> Structural -> Layout
- Can be automated by scripting, but make sure you know what you're doing
  - on-line tutorials for TCL
    - Google "tcl tutorial"
  - Synopsys documentation for design_compiler
  - encounter.cmd (and documentation) for EDI
- End up with placed and routed core layout
  - or BLOCK for later use…