# Synthesis with Design Vision

Install and run Xming and Xming-fonts. Both are available for free downloading from [http://uits.uark.edu/prodsysdb/index.php?method=alpha&value=X](http://uits.uark.edu/prodsysdb/index.php?method=alpha&amp;value=X).

Log into turing.uark.edu with GitHub, setup Connection/SSH/X11 enable X11 with ‘localhost:0.0’. Create working directories:

$tcsh

$cp /home/smli/cshrc.dat . //added

$source cshrc.dat //added

$synopsys

$ mkdir ‐p Project\_Name/{src,syn,sim}

--Copy your design into the src directory.

Create .synopsys\_dc.setup file for default library assignments.

$ cd Project\_Name/syn

$ nano .synopsys\_dc.setup

set link\_path

{/mscad/apps/Linux/synopsys/SAED\_EDK90nm/Digital\_Standard\_Cell\_Library/synopsys/models/saed90nm\_typ.db}

set target\_library

{/mscad/apps/Linux/synopsys/SAED\_EDK90nm/Digital\_Standard\_Cell\_Library/synopsys/models/saed90nm\_typ.db}

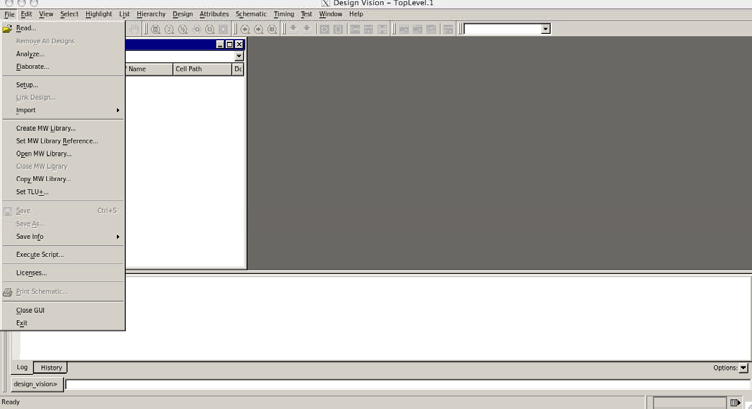
set symbol\_library

{/mscad/apps/Linux/synopsys/SAED\_EDK90nm/Digital\_Standard\_Cell\_Library/synopsys/icons/saed90nm.sdb}

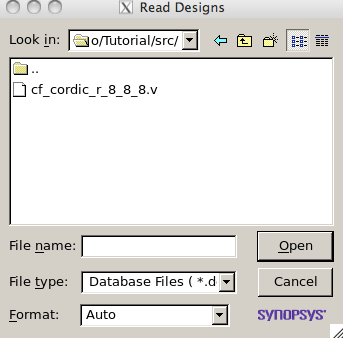
Save the file.

Start Design Vision:design

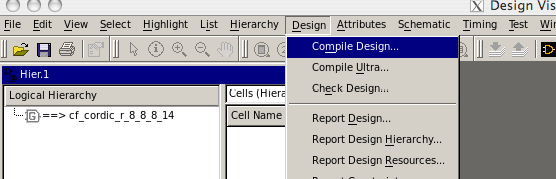
$ design\_vision

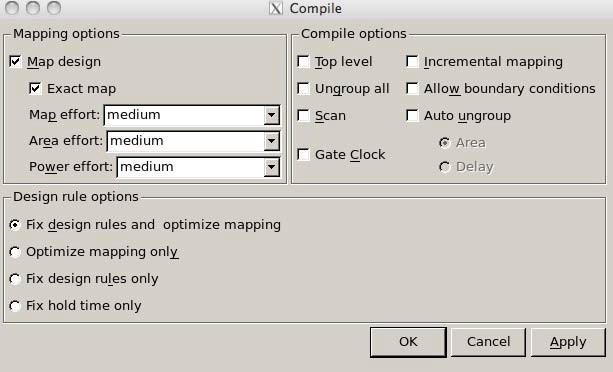


# 1. In Design Vision, Click on File->Read



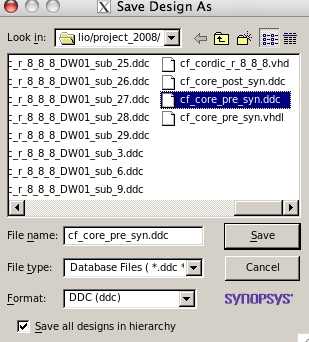
# 2. Design > Compile Design...



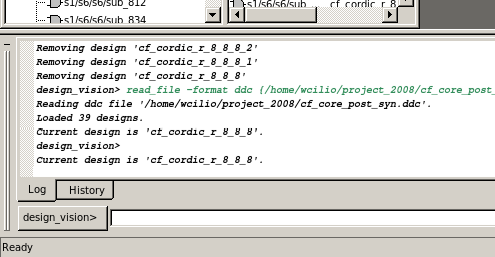


For this tutorial we will leave the defaults, but you can change them to better fit your design constraints such as area, and power. Click on OK after you are ready.

# 3. Save your file as .v (or .ddc) so that we can use it in ModexitelSim later on, give it a different name so we know is the synthesized design. In this tutorial it will be called top\_entity.v.



The next steps are to set your constraints. Use the console at the button of the window to input the constraints. Change clock\_c for the name of your clock.



* Timing Constraints
  + create\_clock -name "CLK\_0" -period 30 -waveform { 0.000 15.000 }

{ clock\_c }//creates clock period and duty cycle.

* + set\_clock\_uncertainty 0.14 CLK\_0 //Clock Skew
  + set\_input\_delay 2.0 -max -clock CLK\_0 [remove\_from\_collection [all\_inputs] clock\_c] //input delay except for clock.
  + set\_output\_delay 0.5 -max -clock CLK\_0 [all\_outputs]
* Design Constraints

o set\_max\_area 1000.000000

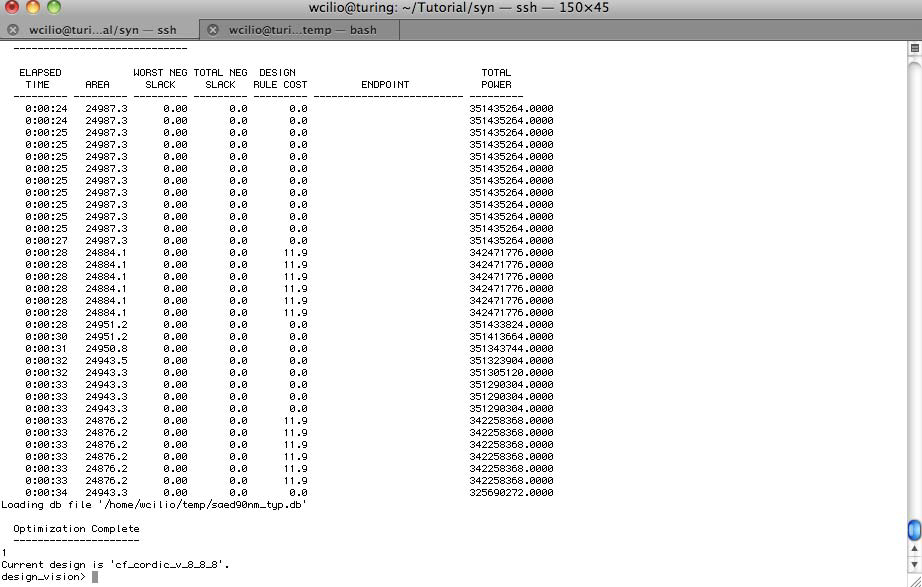
* + set\_max\_dynamic\_power 0.2 uW
  + set\_max\_leakage\_power 0.005 uW
  + set\_max\_total\_power 0.5 uW

You can also compile your design from the console with the following commands:

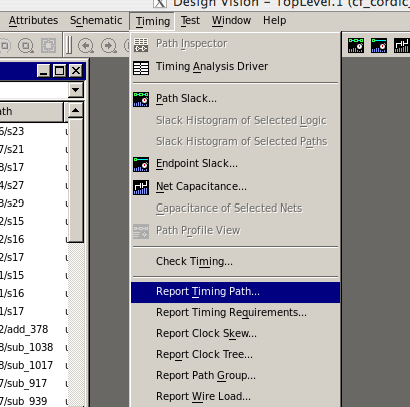
current\_design cf\_cordic\_r\_8\_8\_8 //this is the name of the top level

compile ‐map\_effort high ‐area\_effort high //select different efforts just to see what fits best.

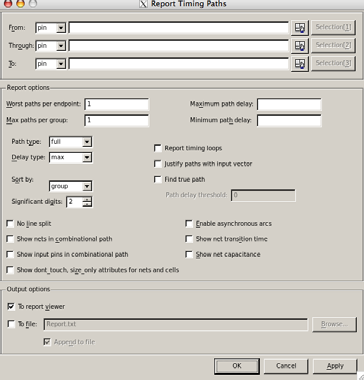
Look at your console and you can see what the compiler is doing as well as a final report.



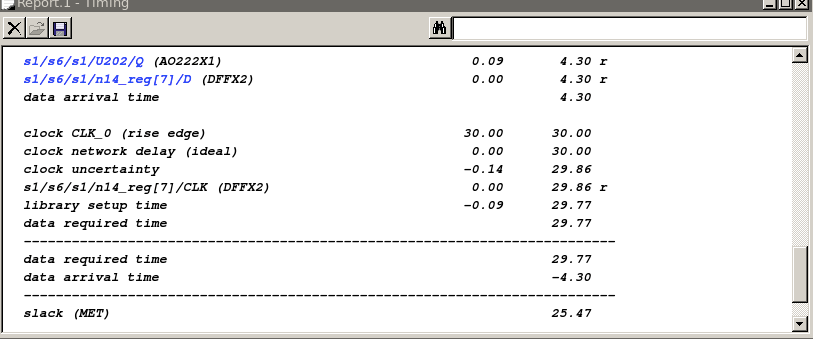
Go to Timing7Report Timing Path



Click ok on the new window.



A report is generated for the critical path. If the timing requirement is met you will have zero or positive slack. If the slack is not met, then try making your clock slower.



Now save your design one more time using SAVE AS so only one file is created. You can overwrite the old files.

You can also get the constraints saved into a file for later use with PrimeTime

write\_sdc <filename.sdc> //TopLevel.sdc in my case

# Post-Synthesis ModelSim Simulation

We need some Verilog files so that ModelSim takes into consideration the delay in your gates.

This process has created the Verilog files that include the delays for all the standard cells for the process. They are in the following directory.

/mscad/apps/Linux/synopsys/SAED\_EDK90nm/Digital\_Standard\_Cell\_Library/ve rilog/

Make sure you copy the whole directory along with your synthesized code to your ModelSim directory or project directory.

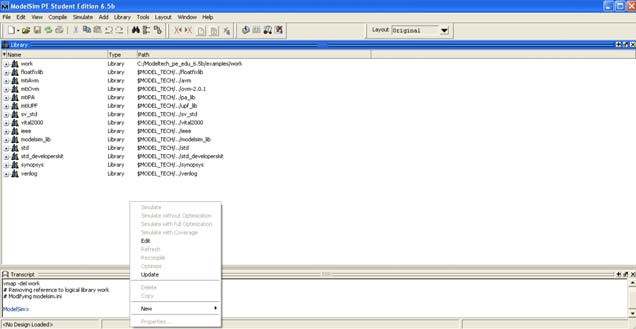
Open ModelSim

$tcsh

$modelsim

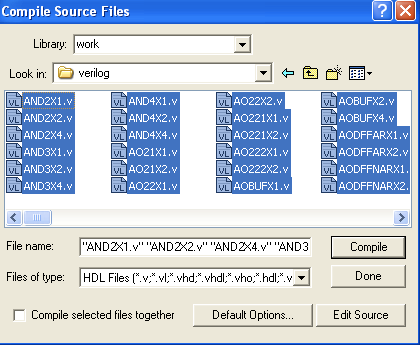
$vsim

Create a library called work\_syn.



Click Compile 7 Compile

First, we need to compile the gate delay Verilog code. Make sure the library field is set to work\_syn. Select all of the gates that are contained in the directory we downloaded from Turing and click the compile button.

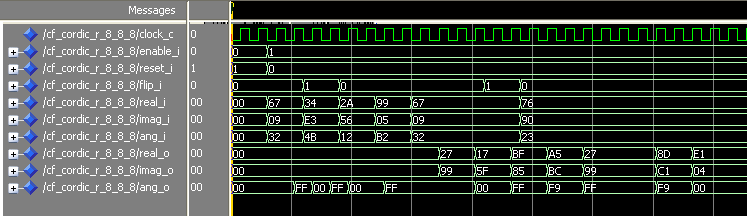


This process will take some time. Keep looking at the command window (bottom of ModelSim window) to know when it is done.

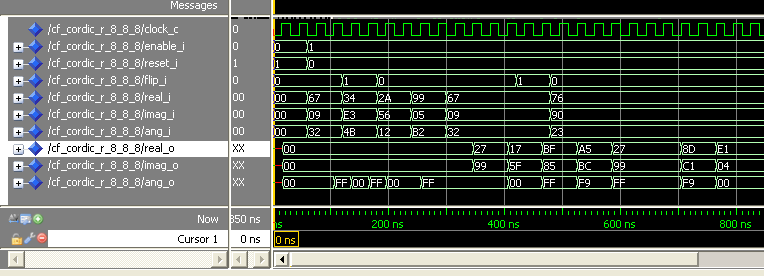
Once it is done, compile the synthesized code you saved from Design Vision.

Now, we can run a simulation on the new code using the same testbench or .DO file used for the pre-synthesis.

We can look at the results now: Before Synthesis:



After Synthesis:



We can see that the delay has been taken into account because of the delay on the waveform. Zoom in and zoom out so that you can see the delay.

From the before and after synthesis we can see that the outputs are the same.

Therefore, we can say that the synthesis is equivalent to our behavioral code.