ECE 394 ASIC & FPGA Design
Synopsys Design Compiler and Design Analyzer Tutorial

A. Setting Up the Environment

a. Create a new folder (i.e. synopsys) under your ece394 directory
~/ece394 % mkdir synopsys

b. Copy the environment and setup files to this folder from /vol/ece394
~/ece394 % cd synopsys
~/ece394/synopsys % cp /vol/ece394/.synopsys_dc.setup ./
~/ece394/synopsys % cp /vol/ece394/synopsys.env ./
~/ece394/synopsys % ls –la
Now you must be able to see both of the files in your directory.

c. Setup your environment
~/ece394/synopsys % source synopsys.env

d. Copy your .vhd files to /synopsys folder (this may change depending on where you keep your design files and their names) I'll go with the full adder example in class.
~/ece394/synopsys % cp ../my_vhdfiles/HA.vhd ./
~/ece394/synopsys % cp ../my_vhdfiles/OR.vhd ./
~/ece394/synopsys % cp ../my_vhdfiles/FA.vhd ./

At this point you can either choose to use the Design Compiler (the one without the graphic interface) or you can use Design Analyzer (the one with the GUI). In either case you can execute the compiler commands that I tried to cover in class one by one (or in Design Analyzer case use the GUI menus to perform those executions) or create a script file (ie synth.script)

B. Creating Your Script File

a. Choose your favorite text editor program (i.e. xemacs)
~/ece394/synopsys % xemacs &

b. Insert the compiler commands that you want to use. Here only the most basic and widely used commands will be covered. These commands you should be enough for the scope of this course but if you are really interested in the tool or need to setup a different constraint in another project I'd recommend a google search on them.

Analyze all your vhdl files that are hierarchically related to your top level design that you are synthesizing.

analyze –format vhdl HA.vhd
analyze –format vhdl OR.vhd
analyze –format vhdl FA.vhd

Elaborate just your top level entity

elaborate full_adder –arch structural

Set your current design to your top level design (this tells DC which module you want it to work on)

current_design adder

Uniquify your components (explained in class)

uniquify

Setting up the operating conditions of the environment:

set_operating_conditions –library “lsi_10k” “parameter”

where parameter can be

WCCOM	BCCOM	worst/best case commercial
WCIND	BCIND	worst/best case industrial
WCMIL	BCMIL	worst/best case military

Setting the input and output drive strengths. For set drive the value entered before the input pin name corresponds to the port drive resistance (>=0) where a bigger value corresponds to a pin harder to drive (so 0 means this pin is highly unresistive). For the set_load the value specifies the capacitance value of the output pin.

set_drive 0 “in1”
set_drive 2 “in1”
set_load 0.5 “sum”

Setting up timing and area constraints:
Create a clock. Since we don’t have a clock in this design these commands are just given FYI.

create_clock –period 10 –name CLK
set_clock_skew –ideal “CLK” for ideal clocking
set_clock_skew –propagated “CLK”
propagate clock delays through network
set_clock_skew –uncertainty 0.2 “CLK”

Input and Output delays relative to a clock. For example set_input_delay command defines the time a data arrives at an input port relative to a clock edge (ie there is extra logic between the flip flop driving the input of this module, so it takes extra time to have a valid input).

set_input_delay 3 –clock “CLK” “in1”
set_input_delay 5 –clock “CLK” “in2”
set_output_delay 5 –clock “CLK” “sum”

Setting up delay requirements for a specific path:

set_max_delay 10 –from “in1” –to “sum”
set_max_delay 10 –from “in2” –to “sum”

Specify a signal not to be optimised (ie clocks will optimised by clock tree insertion, so DC shouldn’t try to optimise it for now)

set_dont_touch_network <object list>

Setting up area constraint. Usually the value given for area optimisation is 0, which means after meeting the timing constraints, minimize the area as much as possible.

set_max_area 0

Compilation. Map effort high parameter tries to optimise the logic in the critical path to meet the timing constraints. Use it only when your initial compile does not meet your specifications because high effort increases compilation time drastically. Scan parameter replaces all your flipflops with Scan flipflops for testing and verification purposes (to generate the scan chain)

compile
OR **compile – scan**

OR **compile –map_effort high -scan**

Write the generated netlist to a file

write –format vhdl –hierarchy –output FA.svhd

Creating your timing and area reports. (area_log and timing_log are just file names) In report timing you can set the number of paths you want to be reported (in this example 3 worst delay paths)

report_area > area_log
report_timing -max_paths 3 > timing_log

Close the compiler

quit

c. Save your script file (i.e. synth.script)

C. Synthesis with Design Compiler Shell

a. Since you have set up your environment and created your script file all you have to do is to type

~/ece394/synopsys % dc_shell –f synth.script >& log_file

log_file will keep the log of your synthesis, for example if something goes wrong(i.e. cannot locate one of your files, a typo in your script or vhdl files etc.) you can see it in that log file.

b. You can find the timing and area reports of your synthesis in the area_log and timing_log files, where you can see if your circuit meets the timing constraints or not and determine the speed-area pair which optimizes your design in the design curve (01.21.05 lecture slide 9)

c. Also if you just type

~/ece394/synopsys % dc_shell

the DC shell command line will open for you to type the commands one by one and if you type a command (i.e. set_max_area) without its parameters, you'll get an error message and the shell will tell you what kind of parameters you can use and the syntax for the command.

D. Synthesis with Design Analyzer GUI

a. Type the line below to start the GUI.

~/ece394/synopsys % design_analyzer

The design analyzer window will pop

Note: the directory where you invoke design_analyzer would contain the `command.log` *file which lists the commands executed.*

b. Verify that your .synopsys_dc.setup file was executed by selecting **Setup -> Defaults**. You should be able to see link library is * lsi_10k.db; target library is lsi_10k.db and symbol library is class.sdb.

Close the setup menu window by selecting Cancel, please DO NOT close any windows in the Design Analyzer by using the "close" window command of the native windowing environment! ALWAYS use the Cancel buttons provided.

c. Select **Setup -> Command Window** to bring up the window that gives you access to dc_shell and also immediate feedback on the progress of your synthesis session. Resize it and drag it to the appropriate place in your display. All commands entered via the menus of the Design Analyzer are echoed, so you can learn how to write dc_shell commands that you don't know from this.

d. **Select File -> Analyze**, go to your ece394/synopsys directory to see the appropriate files. Select the appropriate format for **File Format**.

To analyze more than one HDL file: click on the first file with the left mouse button, and select the other files using the middle mouse button. Or simply type them all in the "File Name" box, separating the files with spaces (not commas).

e. Inspect the messages in the Analyze window, correct any syntax errors in your HDL files and do the analyze again, otherwise, cancel the Analyze window and proceed.

f. Select **File -> Elaborate**, select **full_adder from the Library box**, select your "top level design" from the Design box, and click OK. Elaboration brings all the associated lower level blocks into the Design Compiler automatically (by following the dependency of the instantiations within the HDL code). After the elaboration is done, cancel the Elaborate window.

g. On the left side of the Design Analyzer window are the View buttons.

The top 4 buttons select the type of view: Design, Symbol, Schematic or Text. The bottom 2 buttons are used to traverse the hierarchy of a design. Select the icon for your top level design block, say full_adder, by clicking on it, the border of icon is shown as a dashed line instead of a solid line, the design is now full_adder, as seen in the lower left corner of the Design Analyzer window.

h. Double click on the adder icon and this will produce the Symbol View.

The Symbol View is convenient for applying attributes and constraints to a design. Click on the appropriate port and select from the **Attributes** menu your desired constraints.
For example if you select in1 and sum then select **Attributes>Optimisation Constraints>Timing Constraints** a new window will pop up where you'll see a *from* field which contains in1 signals and a *to* field which contains sum field. If you fill maximum delay>rise 10 and select same Rise and Fall, it'll equivalent to say **set_max_delay 10 –from in1 –to sum.**
If your design has a clock port, you have to select your clock port and select Dont Touch Network in the **Attributes>Clocks>Specify** window, Synopsys does not effectively synthesize clock tree. You can also set your clock period in the same window.
Also you can set your input/output delays, drive strengths/loads, operating conditions and wire load models from this menu.
Attributes>Operating Environment>…

Please see the Design Compiler part for properties of this 6 commands.

i. If you go to **Attributes>Optimisation Constraints>Design Constraints** you can specify the maximum area and maximum fanout constraint.

j. At this point you may save your design as an unmapped db format, select **File -> Save As,** navigate to the ./db directory in the Directory menu, and name your design as control_unmap.db , choose DB as the File Format. When a design is saved as a .db file, the design plus all attributes are saved.
The equivalent dc_shell command will be:
write -format db -hierarchy –output FA.db

k. Next time you want to retrieve the already elaborated but unmapped design, you could select **File -> Read** with DB as file format. You can enter **reset_design** in the **command window** to remove all attributes and apply new attributes if needed.

l. Compilation and optimization:
Select Tools -> Design Optimization, the default Map Design setting is Medium. Then click on OK, it might take a few hours or even more than a day to compile. Do not apply other options in your first compile run.

m. When the optimization is complete, a Cancel button located at the bottom of the Compile Log Window will become “grayed in" indicating that you can cancel the window and view your results.

n. Inspection of results:

Notice the changes in any icon of your design, it is now optimized to gate level, double click on the icon to reach the Symbol View, then click on the Schematic View button to inspect the design. Select **View -> Zoom In** to zoom in.

o. You can select pin(s) or port(s) from the design, then select from **Analysis -> Highlight** and a type of highlighting to see the path to the pin or port. Control-T is a shortcut to see the critical path.

p. Select a design block (the top level design if your design is hierarchical), do **Analysis -> Report**, click on **Area, Timing** , you could direct the output to a file for later reference. Inspect the **Report Output** window, use the mouse to select a line, click on the Next button, the item(s) in the corresponding schematic will be "selected" automatically. The equivalent dc_shell commands will be:

report_area
report_timing

q. Inspect the timing report, each **Incr** entry indicates the delay from the previous point to the current point, and the **Path** entry indicates the total delay from the input external delay to the current point. You can detect any suspicious path with exceptional long delay through this inspection. The most important thing is to check the **slack**, which is the **required delay** minus the **actual delay**, if it

reports **MET**, your design has met the timing constraints, if it reports **VIOLATED** , you should go back to your HDL code and re-write it to improve timing. Then go back and re-analyze -elaborate the block and compile the whole design again.

r. Save your design by selecting **File -> Save As**, navigate to the .db directory in the Directory menu and choose DB as the File Format, it is recommended to use the Save All Designs in Hierarchy option.

s. You might want to read in another design without quitting Design Analyzer, you could first remove the current design by selecting the design to be removed, then **Edit -> Delete** . This would remove the design from the Design Compiler memory, it would not remove any physical design files.

t. To quit the Design Analyzer, select File -> Quit and click OK.

E. Post-Synthesis Simulation: ModelSim simulations with the netlist generated by the Synopsys tools. During the period when you were adding your .vhd files to your library (either using the ModelSim GUI or by typing vcom ***.vhd) instead of adding the .vhd file if you specify the netlist file you write (ie FA.svhd) the simulations will be carried on using the netlist not the .vhd file, so in this way you can see the delays of different paths and the result converging in time rather time taking the correct value at the same instant the inputs change!

F. Further Questions about the Tools

For your further questions about the Synopsys Design Compiler and/or Design Analyzer or if you want to comment on this tutorial, you can email Min or me.

There is a sample script file for .vhd files for the full adder example. You can copy it from /vol/ece394/

~/ece394/synopsys % cp /vol/ece394/synth.script ./