National Tsing Hua University
Department of Electrical Engineering
EE429200 IC Design Laboratory, Fall 2021

Lab 06-1: Design Compiler
Assigned on Oct 21, 2021
Due day in Oct 28, 2021

Objective

In this lab, you will learn:

1. Synthesis design with ***Design Compiler***.
2. Understand and run the gate-level simulation.

Demo checklist:

- ☐ Record the result of gated/non-gated clk and show TA.
- ☐ Answer the following questions
    1. What's the purpose of adding ***$sdf_annotate*** into testbench when doing gate-level simulation?
    2. Why do we need this file "***saed32nm_hvt.v***" in the gate-level simulation, but pre simulation does not?
    3. What's the difference between gated/ non-gated clk? (power, area, timing …)

Environment Setup

Copy lab file packages from ee4292. Decompress the package and enter it. You can check the file list in Appendix.

```
$ cp ~ee4292/iclab2021/lab06.zip .
$ unzip lab06.zip
$ cd lab06/lab06_part1
```

Note: please run simulation in the ***sim*** directory to maintain a fine data management.

Description

In lab6 part 1, we are going to **synthesize our CPU with Design Compiler (DC)**. Rather than using the graphical user interface(GUI), we usually use the tool command language(TCL) script to synthesize. Please first read the synthesis tutorial to understand the synthesis process and then finish the scripts. After you synthesized the CPU with ***Design Compiler***, a netlist will be generated. We want to run the simulation again to make sure the function of netlist still correct. This process is called **gate-level simulation**.

Action Items

I. Check synthesizability.

1. Use ***spyglass*** to check the design.

2. After checking the design, you need to run the simulation to ensure the function correctness.

II. Synthesize the design with Design Compiler.

There are two methods to synthesize the design. **Method-1** is using TCL script in terminal, **Method-2** is using GUI to do synthesize. However, we usually use TCL script to do synthesis, so we recommend you to use Method-1.

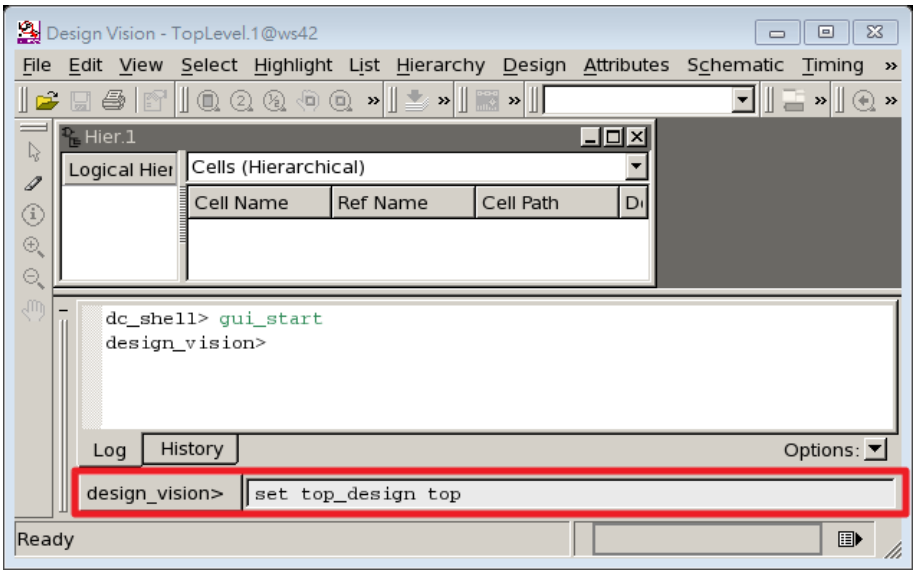
Use following command to open Design Compiler:

$ dc_shell (for **Method-1**, open Design Compiler Command Line termeinal)

$ dv & (for **Method-2**, open Design Compiler GUI)

Method-1: Continue from the provided tcl files, and finish them as explained in synthesis tutorial.

Method-2: To avoid typos in the scripts, besides directly write the scripts in tcl files, you can enter the commands in the DC_shell or GUI of DC line by line.



III. TCL scripts.

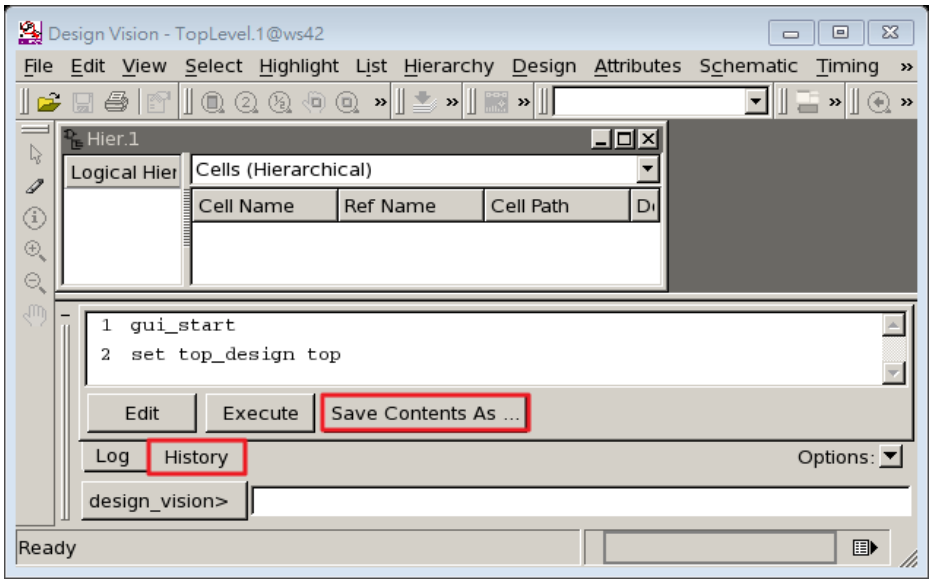
1. If you use the Method-1 in Action II, use the command below to synthesis with DC.

   $ dc_shell –f synthesis.tcl | tee logfile

Note 1: the "**| tee**" is a linux command to save the outputs of the program into a text file.

Note 2 : You can change "**logfile**" to any name you want, typically we use the name `da.log`

2. If you use the Method-2 in Action II, you have already finish the whole synthesis process. You can save the commands you typed as a tcl script by "**History > Save Contents As…**" After that, you can simply change the synthesis setting by modifying the script, rather than keying the commands again.



IV. Gate-level Simulation.

After synthesizing, a netlist and corresponding cell delay information is stored under "**/syn/netlist/**". You should include these files into the simulation process.

/syn/netlist/top_pipe_syn.v : Netlist generated by Design Compiler.
/syn/netlist/top_pipe_syn.sdf : Delay information inside the netlist, this is also generated by Design Compiler.

1. Please read the provided filelist(**sim_gate.f**) and the testbench (**test_top.v**). Understand their contents and meaning.
2. Use ***nWave*** to observe the gate-level simulation waveform, what's the difference between this waveform (**lab6_gate.fsdb**) and the presim waveform (**lab6.fsdb**)?
3. Please make sure the function is still correct.

V. Explore Design.

Design Compiler is a very strong synthesis tool, it can generate an optimized netlist to meet your requirement according to the constraints you set. Now, we're going to explore the design by setting different kinds of constraints and modifying the RTL codes itself, please do the following experiment and record the results.

Record your result as following table:

Setting	Without gated-clock		With gated-clock	
ALU With AND(&)	Timing		Timing	
	Area		Area	
	Area (alu)		Area (alu)	
	Power (Dynamic)		Power (Dynamic)	
	Power (Leakage)		Power (Leakage)	
	Power (Total)		Power (Total)	

NOTE[1]: Set/Unset gate_clock option in the first compiler_ultra command to enable/disable gated-clock generation.

NOTE[2]: Notice that there are two compile_ultra commands inside 1_compiler.tcl, the second one with parameter ***-incremental*** will further improve the synthesis performance of your circuit.

Appendix

Directory	Filename	Description
source	CPU_define.v	Definition file
source	top_pipe.v	Top module wrapper for this CPU
source	top.v	Top module for this CPU
source	IF_stage.v	RTL for IF stage
source	IF_ID.v	RTL for IF to ID flip flop
source	ID_stage.v	RTL for ID stage
source	controller.v	RTL for control signal
source	regfile.v	RTL for register file
source	ID_EXE.v	RTL for ID to EXE flip flop
source	EXE_stage.v	RTL for EXE stage
source	alu.v	RTL for alu module
source	PC.v	RTL for program counter
source	dsram.v	RTL for memory
sim	test_top.v	Testbench
sim	instruction.txt	Instruction file
sim	sim_pre.f	Simulation script
sim	sim_gate.f	Simulation script
sim	hdl.f	hdl source filelist
sim	Makefile	script to do the actions items
syn	.synopsys_dc.setup	DC setup file
syn	*.tcl	Synthesis scripts
syn	run_dc.bat	batch file to run Design Compiler

Make file descriptions:

A Makefile is provided to ease the pain of typing lots of linux commands.
You can understand and modify this file for your future projects.
(You can use it or not depending on your personal will.)

Understand it before you use it !!!
Understand it before you use it !!!
Understand it before you use it !!!

- make presim : run CPU simulation and save the result in /sim/register.txt
- make gatesim : run CPU gate-level simulation and save the result in /sim/register.txt
  (Note : you should run synthesis first before gate-level simulation!!)
- make syn " run *Design Compiler* to synthesize the CPU into netlist under /syn
- make veryclean : clean all the temporary files under /sim, /source and syn
    - make simclean : clean temporary files under /sim
    - make hdlclean : clean temporary files under /source
    - make synclean : clean temporary files under /syn