Organization of Digital Computers Lab
EECS 112L

Lab 1 - Setting up the CAD Tools

University of California, Irvine Winter

2018

1 Goal

In this Lab, you are going to set up the required tools for hardware design. First you should connect to a server, then you will clone lab files from the GitHub, and at last, you will simulate and synthesize a simple version of single cycle RISC-V processor.

1.1 Connecting to a Linux Server

The CAD tools that we are going to work with during this course are installed on the server. So first we need to connect to one of the EECS machines. In order to be able to connect to a Linux machine, one way is use SSH protocol. If you are using the Windows OS download the MobaXterm and If you have Mac, use Xquarts. Find details about using these two SSH protocols in the Top Hat Chapter 2.
These are a list of EECS servers. Use your UCINetID and Password to connect to either of these machines.

zuma.eecs.uci.edu
laguna.eecs.uci.edu
crystalcove.eecs.uci.edu
bondi.eecs.uci.edu

1.2 Using GitHub

Git is a distributed revision control system which is fast with a rich set of commands. We are going to put all necessary files for this course in the UCI GitHub. So you need to know how to use it. Generate the ssh key and clone the repository to some local folder on the server. You can find more information about GitHub in the Top Hat. Watch the video and follow the instructions.

1.3 Directory System

After cloning you will see that there are 4 directories in the lab1 folder. The design folder contains all System Verilog design sources. All the files that we need for simulation are in the sim folder and synthesis files are in the syn folder. you will find test benches and other files that we need for verifying our design in the verify folder. During this course, you should manage your project files in the same way.

1.4 Compile and Simulation

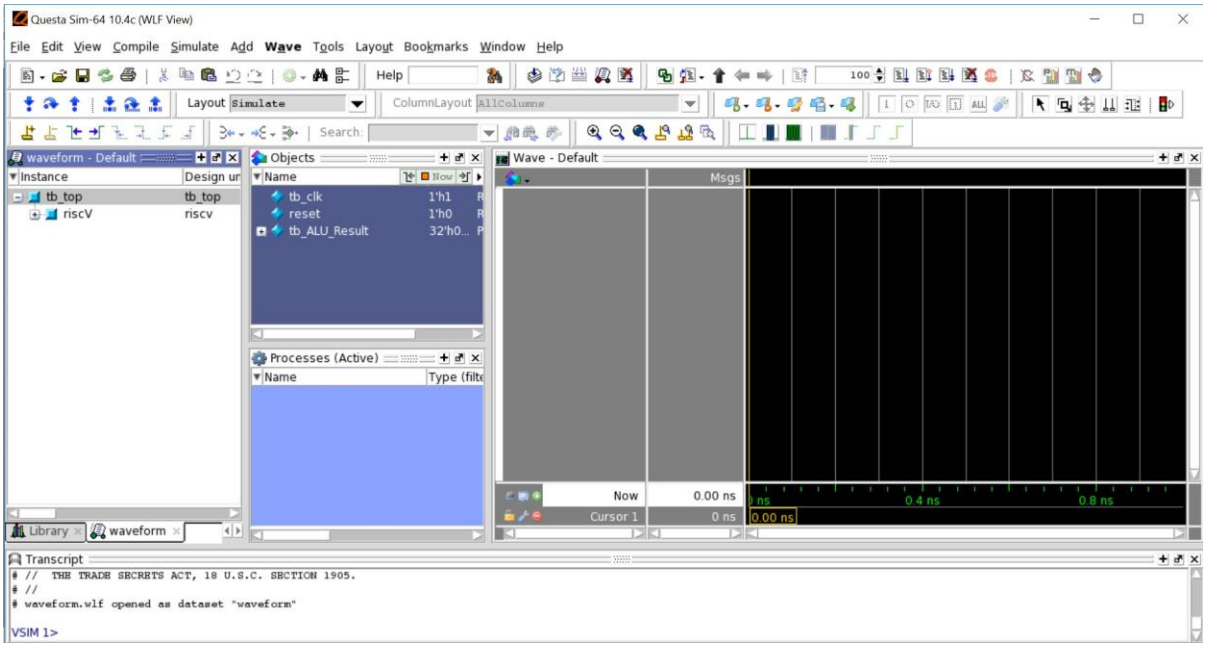
In this section, you are going to use the Questasim software on the server to compile and simulate the single cycle processor code.
Go to the folder location in your online account in the server, where the files are uploaded. Use the following Linux commands to do the compilation, optimization, simulation and to view the waveform. Note: All these Linux commands should be executed in the folder location where these files are uploaded to the server.

1. `source setup.csh`
   This source command will load the file into the command prompt. It reads and executes the commands from the file setup.csh
2. `source pre compile.csh`
3. Compile, Optimize and Simulate the System Verilog design sources
   - `vlog -64 -sv -f rtl.cfg`
     This command will compile the System Verilog design source files. If you open the rtl.cfg you will see the list of .sv files. The rtl.cfg is in the design folder so change directory to the design folder or run `vlog -64 -sv -f $design/rtl.cfg .`
     - `vlog` is the compile command
     - `-64` represents that vlog uses 64-bit executable
     - `-sv` is used to enable the System Verilog features and keywords
     - `-f` specifies the argument file with command lines arguments, which allows to use the complex arguments once again without retyping
     - `rtl.cfg` contains all .sv source files
   - `vlog -64 -sv -f tb.cfg -work work`
     This command will compile the test bench source. this file is in the verif folder. So change directory to the verif folder run `vlog -64 -sv -f $verif/tb.cfg -work work`.
     - `rtl.cfg` contains system verilog test bench source file
     - `-work work` The first work is a command option which specifies a logical name or path- name of a library that is to be mapped to the logical library work. Here the name of this specified library is work

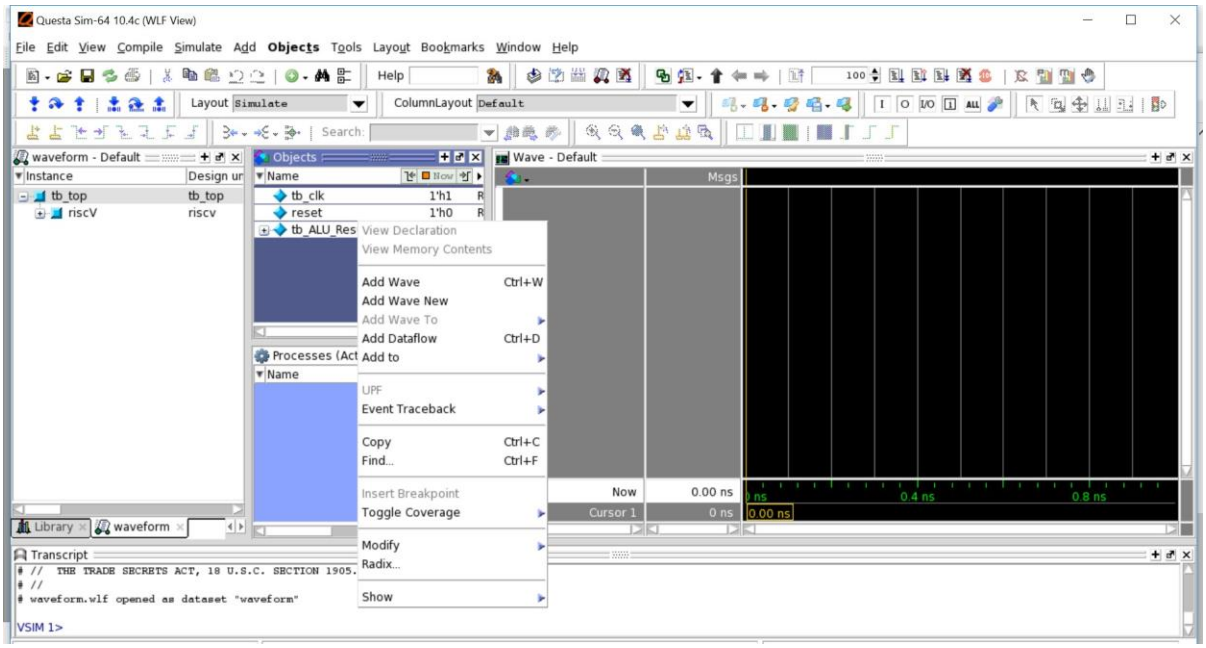
- `vopt -64 tb top -o _tb _top opt +acc -work work`
  This command will optimize the design.
  - `vopt` is used to do the global optimization on the design after the compilation has been done
  - `-o` allows us to designate the name of the optimized design file
  - `tb_ top opt` this is the name that we choose for the optimized design
  - `+acc` provides visibility into the design for debugging purposes
- `vsim -64 -c tb top opt -do sim.do`
  This command will simulate the design.
  - `vsim` is used to simulate the optimized design
  - `-c` specifies that the simulator will run in command-line mode
  - `-do sim.do` tells the vsim to use the commands specified in the do file

You can also run this command : `source sim/run`. The run file contains all of these 4 prevoius commands.

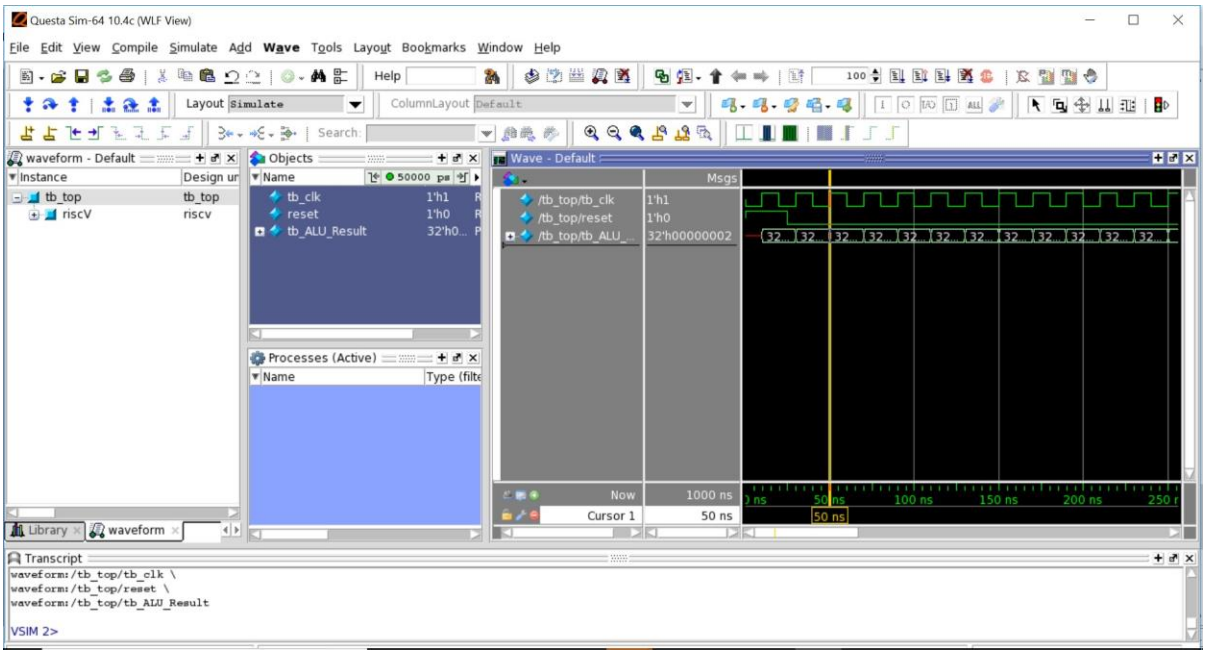
4. View the waveform
   run `vsim -64 -gui -view waveform.wlf`
   - `vsim` is also used to view the results of the previous simulation run when `-view` switch is invoked
   - `waveform.wlf` is the simulation file. It is opened to view the waveform. After running the command, a window will pop up. That window is shown below.



To add the signals to the wave window right click on them (in the object window) and choose ”Add Wave”.



Now you can see the signals in the wave window. Zoom out to see the ALU Result



You can also find a video about the simulation process in the Top Hat.

1.5 Synthesis

The process of converting the high-level RTL code into pure logic gates is called logic synthesis. Synthesis requires EDA (electronic design automation) tools to e↵ectively turn the complex RTL designs to equivalent logic gates. One of the major vendors of logic synthesis tool for ASIC is Synopsys with the tool Design Compiler (DC). This software is also installed on the server. Go to the folder that you clone the lab1 files.

1. `cd syn`
   Change directory to the syn folder
2. `source /ecelib/linware/synopsys15/env/dc.csh`
   This command will add the Synopsys toolset to your PATH variable
3. `dc_shell-t -f synth.tcl`
   This command will invoke the Synopsys Design Compiler (the synthesizer tool) and pass the synth.tcl script to run. This may take several minutes. At the end, you will see some reports about your design like timing and area reports.
4. `dc_shell-t -guil`
   Run this command to open the graphical User Interface (GUI)