Organization of Digital Computers Lab
EECS 112L

Lab 1 - Setting up the CAD Tools

University of California, Irvine

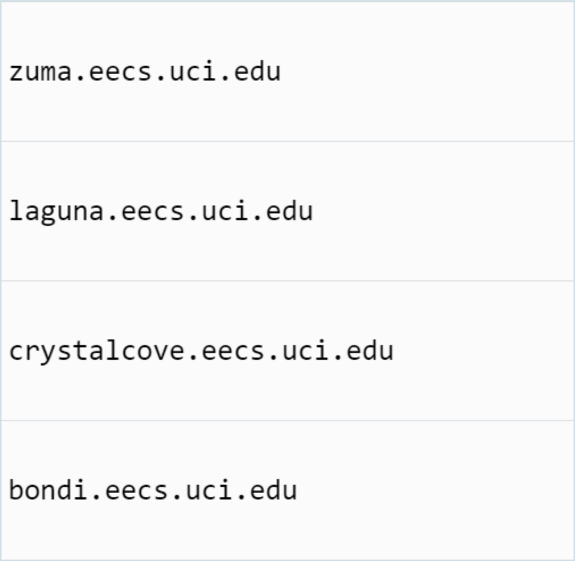
Winter 2018

1 Gole

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 necessary CAD tools for this course are installed on the server. At first, you need to connect to one of the EECS machines. In order to connect to a Linux machine, one way is to use SSH protocol. For Windows clients, download the MobaXterm. For Mac clients, 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 testbenches and other files that we need for verifying our design in the verify folder. Follow the same pattern for your all of your projects during this course.

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.
Use the following Linux commands to do the compilation, optimization, simulation and to view the waveform.

1. `source setup.csh`
   This command sets the environment.

   Note: All these Linux commands should be executed in the folder where the input files are uploaded. For example, you should cd (Change Directory) to the sim folder to run the `source setup.csh` command.

2. `source pre_compile.csh`

3. Compile, Optimize and Simulate the System Verilog design sources

   - `vlog -64 -sv -f rtl.cfg`
     This command compiles 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 compiles the testbench source. This file is in the verif folder. So change directory to the verif folder or 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 pathname of a library that is to be mapped to the logical library work. Here the name of this specified library is work

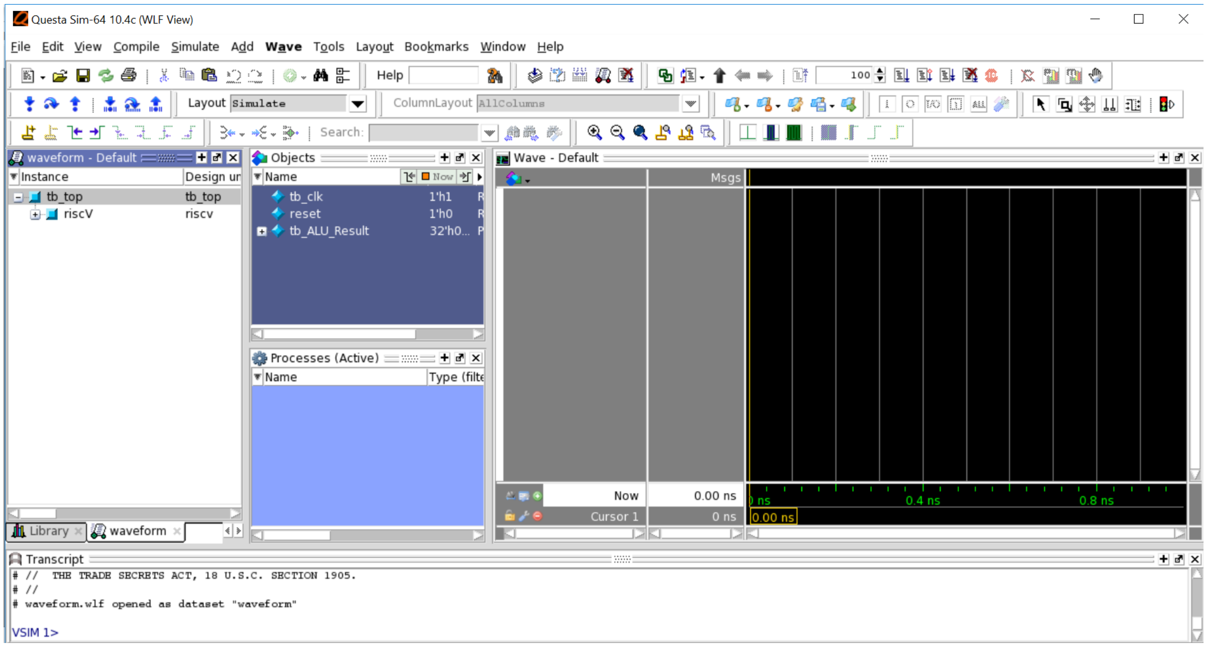
- `vopt -64 tb_top -o tb_top_opt +acc -work work`
  This command optimizes the design.
  - `vopt` is used to do the global optimization on the design after the compilation has been done
  - `-o` allows us to designate the name of the optimized design file
  - `tb_top_opt` this is the name that we choose for the optimized design
  - `+acc` provides visibility into the design for debugging purposes

- `vsim -64 -c tb_top_opt -do sim.do`
  This command simulates 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 and executes them consecutively.

4. View the waveform
   run `vsim -64 -gui -view waveform.wlf`

   - `vsim` is also used to view the results of the simulation 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.



Instance	Design ur
tb_top	tb_top
riscV	riscv

Name	Now	
tb_clk	1'h1	R
reset	1'h0	R
tb_ALU_Result	32'h0...	P

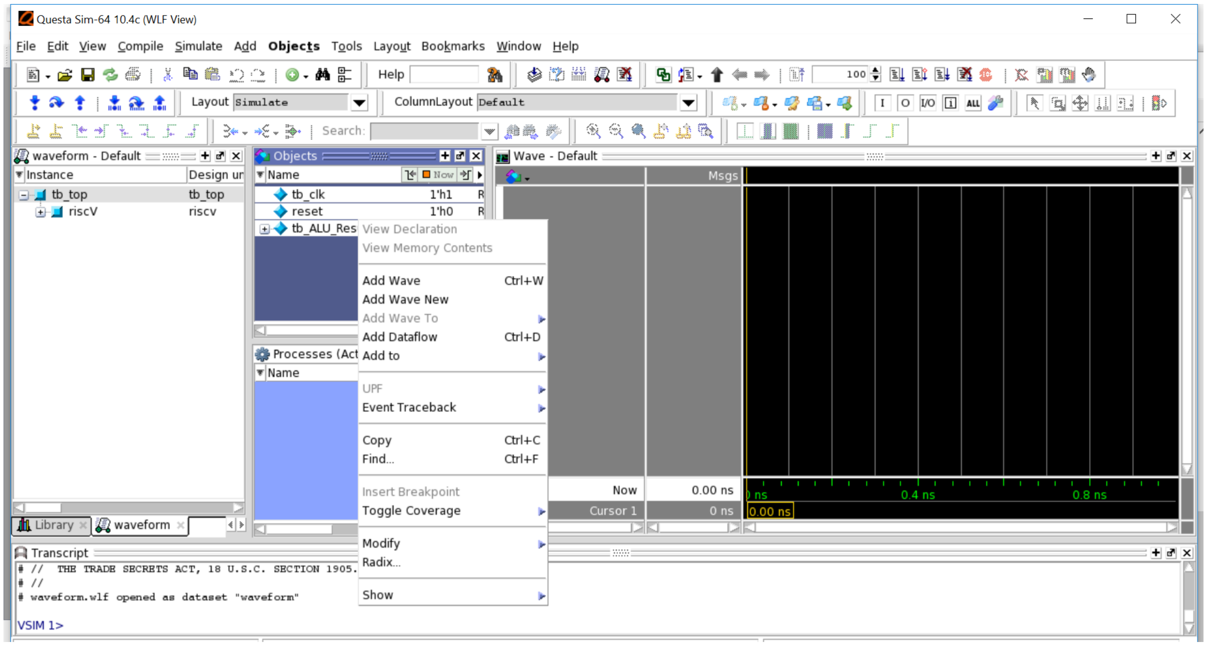
Name	Type (filte

Now	0.00 ns
Cursor 1	0 ns

```
# //  THE TRADE SECRETS ACT, 18 U.S.C. SECTION 1905.
# //
# waveform.wlf opened as dataset "waveform"

VSIM 1>
```

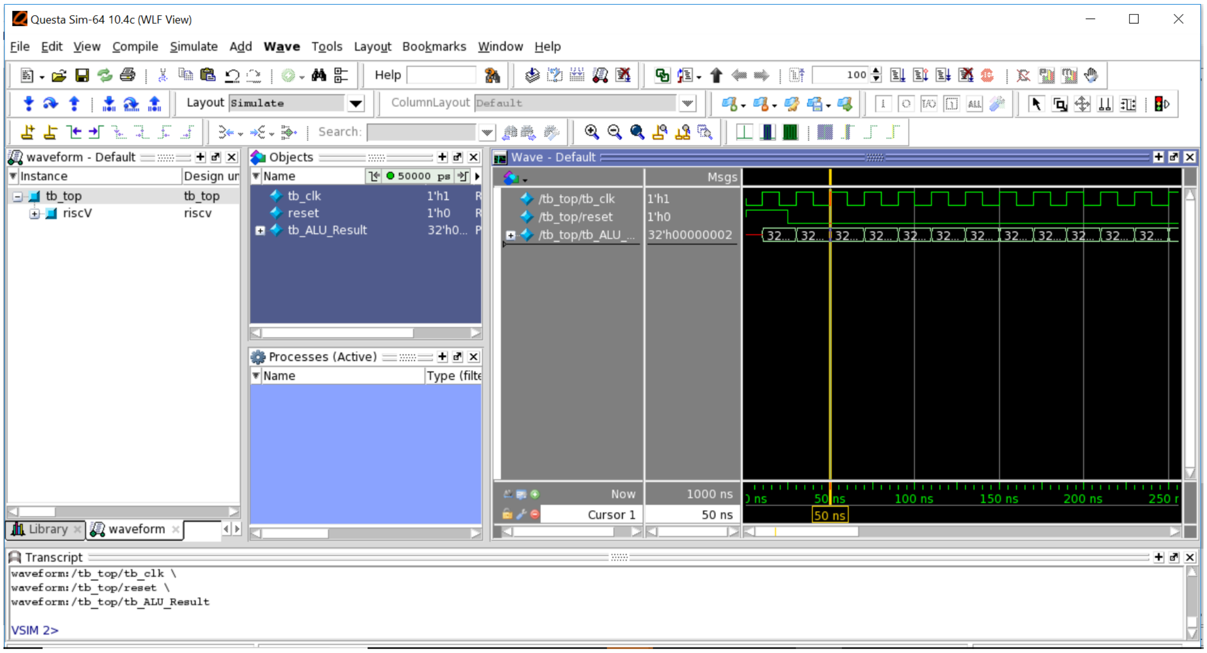
To add the signals to the wave window right click on them (in the object window) and choose "Add Wave".



```
# //  THE TRADE SECRETS ACT, 18 U.S.C. SECTION 1905.
# //
# waveform.wlf opened as dataset "waveform"

VSIM 1>
```

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



```
waveform:/tb_top/tb_clk \
waveform:/tb_top/reset \
waveform:/tb_top/tb_ALU_Result

VSIM 2>
```

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