Lab 0

Tool Introduction Lab Manual

Due date

Aug 26, 2024 11:59 PM CST

Table of content

Academic Integrity

The following actions are strictly prohibited and violate the honor code. The minimum penalty for plagiarism is a grade of zero and a report to the Aggie honor system office.

- Uploading assignments to external websites or tutoring websites
- Copying solutions from external websites or tutoring websites
- Copying code from a classmate or unauthorized sources and submitting it as your work.
- Sharing your solutions with a classmate.

Introduction

The objective of this lab is to familiarize you with the verification software and tools that we will use this semester.

Design Under Test

Your first design under test (DUT) is a module that processes an input address and extracts various fields from it, such as index, tag, and block offset for address decoding in a cache memory system.

The module takes an input address and processes it based on the control signals and parameterized bit ranges, providing the extracted index, tag, and block offset portions of the address as outputs.

Environment Setup

We will use the Cadence Xcelium verification platform for our labs. Cadence Xcelium is installed on ECEN Unix servers. As a student in this course, you should have an ECEN Unix account (regardless if you are in CSCE or ECEN).

If you do not have an ECEN account, contact TAMU help desk central as soon as possible.

Login to the Linux server

Note 1: If you are using macOS, install xQuartz to enable graphical windowing.

Note 2: Texas A&M's Virtual Private Network (VPN) allows you to access the university's network from off-campus. If you are connecting to the ECEN server from outside the TAMU network, establish a VPN connection before starting SSH. Follow the instructions for VPN at https://connect.tamu.edu.

Start SSH with X11 Forwarding enabled (-X). X11 forwarding allows you to run graphical applications such as Cadence tools from the remote server and have their graphical user interfaces (GUIs) displayed on your local machine.

All labs require simulation and debug software installed on a Linux server. Use a secure shell to remote login to the Olympus server (olympus.ece.tamu.edu). Once logged in, run the command load-csce-616.

```
ssh -Y <netid>@olympus.ece.tamu.edu
load-csce-616
```

1. **If you are using a TAMU Linux server for the first time, you must set up your home directory first**. Follow the steps here: https://tamuengr.atlassian.net/l/cp/uNUrkbBa
2. Accept the assignment's repository on GitHub Classroom: https://classroom.github.com/a/Oy-z5Kek . Link your GitHub account to the classroom. If you cannot find your name, you accidentally linked the account to an incorrect name, or you want to change the account linked to your name, please message the course instructor on Discord.
3. Clone your lab repository on the Linux server.
4. Setup your environment

```
source setupX.bash
```

5. Check the Cadence tool version.

```
xrun -version
```

To confirm, you should see:

```
TOOL:   xrun    22.03-s012
```

6. Change the directory to work (cd work)

   Read the content of the lab directory and understand the file structure.

- design/common: this directory contains component design files
- sim: the directory that contains files to control simulation and store results.
- tb: the directory that contains testbench files. There is only one testbench for this lab.

Getting Started

Part 1: Run a simulation using the provided testbench

In this section, you will learn how to run a simulation on a testbench using Cadence tools.

1. Go to the directory sim/ and find the run.f file. This file launches and controls the simulation. Open run.f with a text editor. Read the comments in the file to understand what each command does.
2. Return to the terminal. Type the following command to launch Cadence Xcelium.

```
xrun -f run.f
```

The xrun command compiles and elaborates files specified in the file run.f.

Two windows will pop out, one is the console, and the other is the design browser.

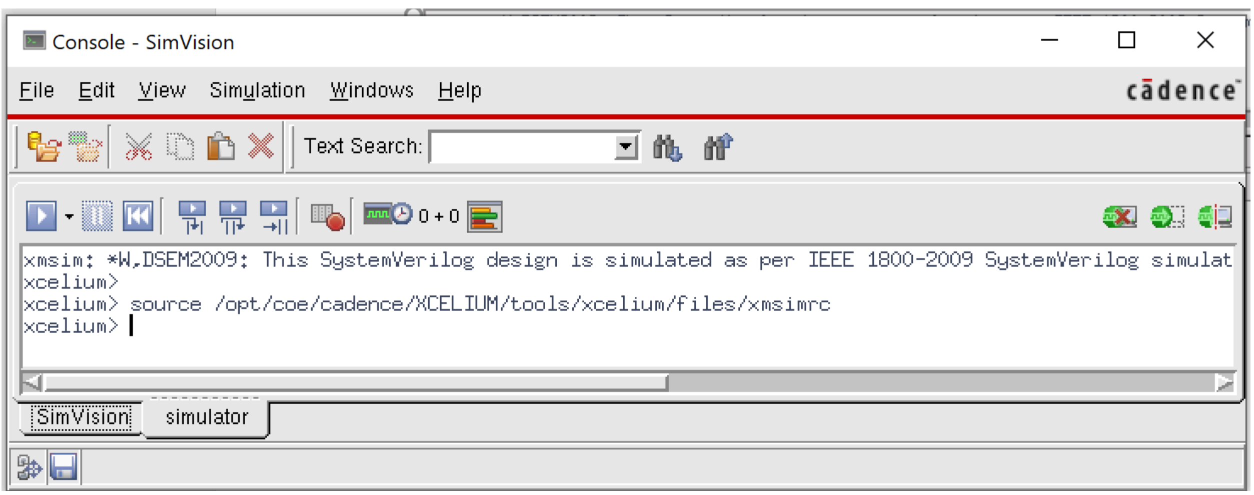

Figure 2-1 Console

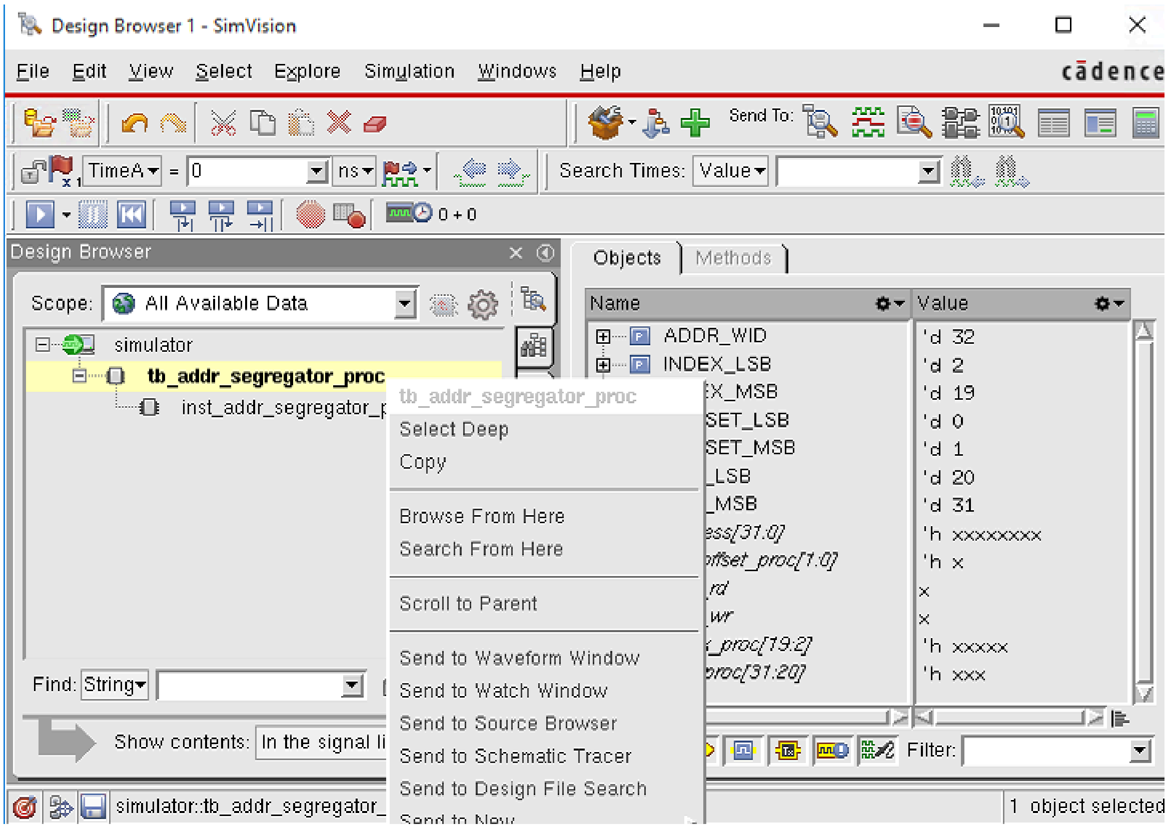

Figure 2-2 Design browser

1. Select tb_addr_segregator_proc in the design browser; you can see the hierarchy of the testbench and its instances (Fig.2-2). Right-click on any instance and choose Send to Waveform Window. All the signals of that instance will be added to the waveform viewer. You can also choose the signals you need individually in the Object window on the right and send them to the waveform window.

2. In the console window, type run to start a full simulation or specify a run duration for the simulation.

   e.g.

```
run 100ns
```

   The waveform will display (Fig. 2-3).

3. Store the waveform by clicking File -> Export. Choose All recorded variables and click OK. The waveforms will be saved to the file you designated (Fig. 2-4). You can load the waveform by clicking File -> Open Database.

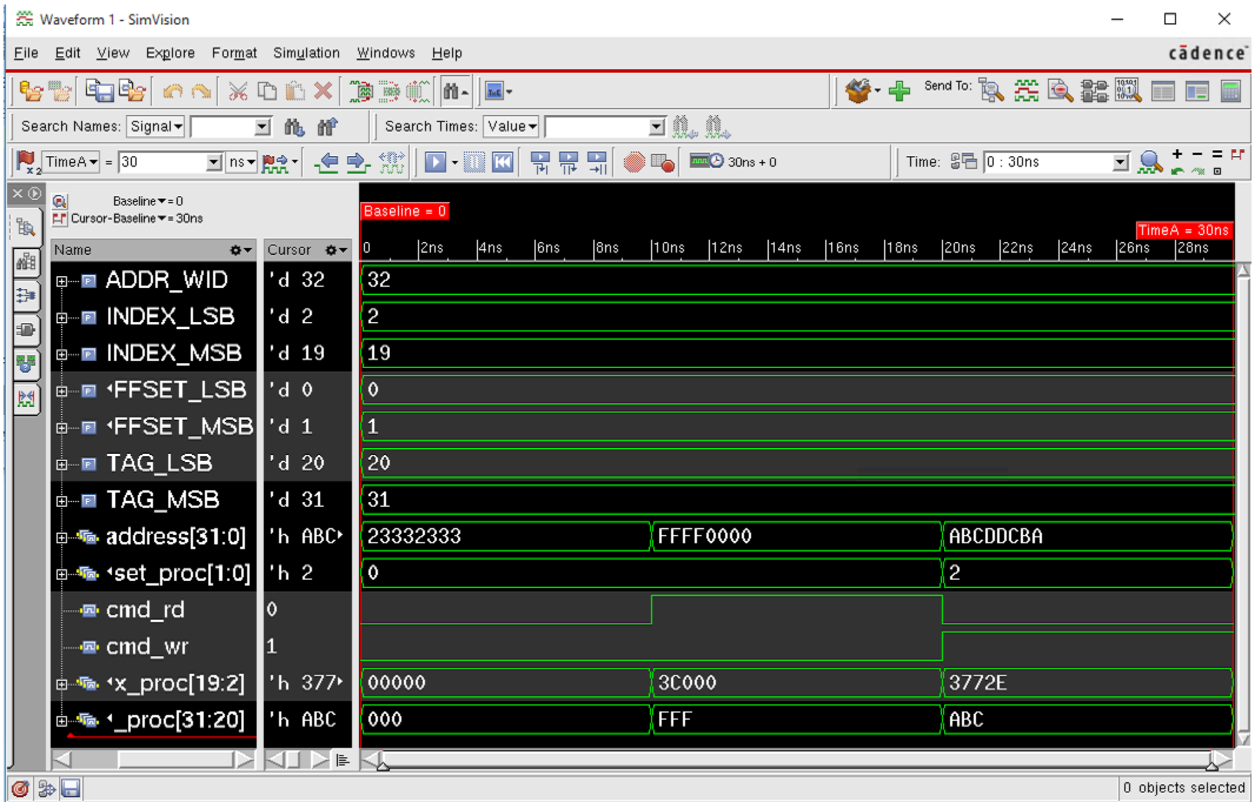

Figure 2-3 Waveform viewer

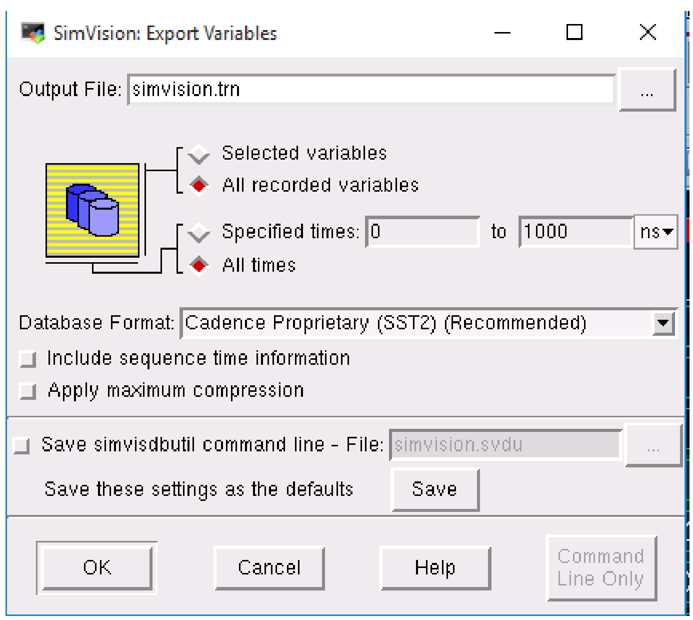

Figure 2-4 Save waveforms

4. To view an existing waveform database (without rerunning a simulation), follow the steps below:

   a. Type the following command on the UNIX terminal: simvision

b. Click on File -> Open Database

c. Choose your database. The default one would be waves.shm

d. After the database is loaded, click on waves -> tb_addr_segregator_proc. All the signals in this hierarchy will be shown in the window below. Clicking on a signal will add the signal to the waveform display.

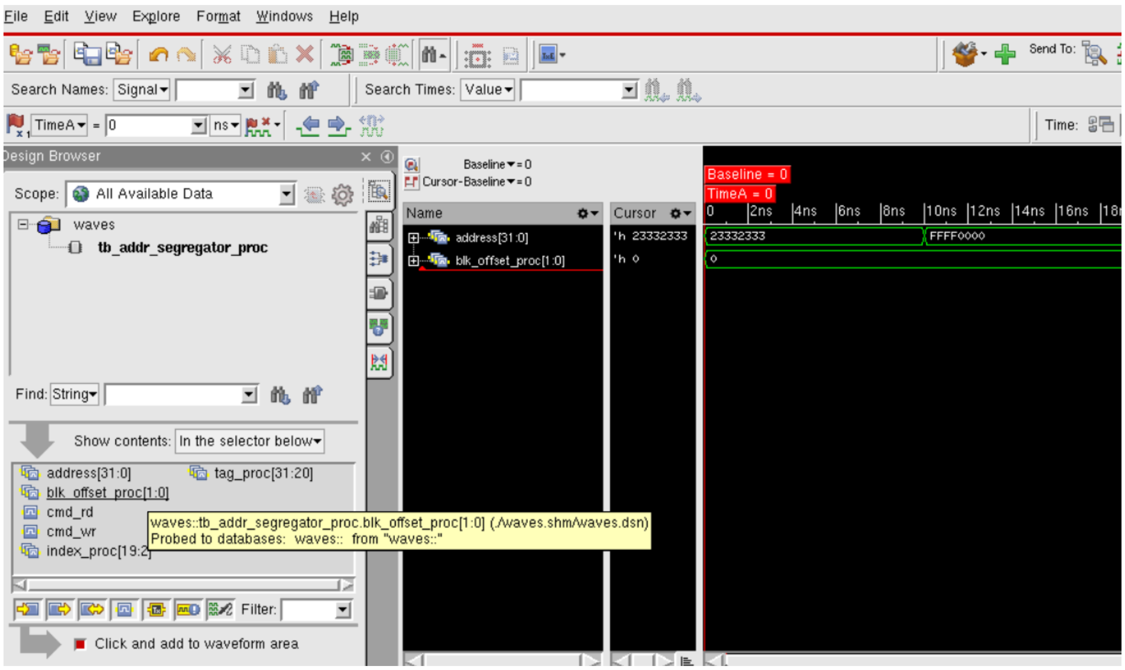

Figure 2-5 Waveform Viewer

Part 2: Create your testbench

Open the testbench file tb/tb_addr_segregator_proc.sv and read it.

The first part of this testbench includes parameter and variable declarations, with all signals and parameters used in the testbench.

The second part is component instantiation. Any new instances are added here.

The third part contains blocks to control simulation. You can add test cases to the initial block for your simulation.

To-do

1. Update the testbench tb_addr_segregator_proc.sv by adding two new test cases.

   Testcase 4: Make a write request to the address 0xFEEDC0DE for 30 time units.

   Testcase 5: Make a read request to the address 0xC00010FF for 60 time units.

2. Run the simulation and capture the full waveform in an image file cache_waveform.png

3. Export the waveform database to the file cache_waveform

Deliverables

Commit and push all your changes to your remote repository.

Your repository must include the following:

1. design directory
2. sim directory containing the exported waveform database file cache_waveform
3. tb directory containing the updated testbench tb_addr_segregator_proc.sv
4. The waveform image cache_waveform.png showing the simulation from the updated testbench

Important note: To get full credit, you must upload all the required files and directories and strictly name your files according to the requirements.