Computer-Aided VLSI System Design
Lab1: 4-Bit ALU with Two Instructions

TA: 黃佳穎 r13943003@ntu.edu.tw

Introduction

The arithmetic logic unit (ALU) is one of components of a computer processor. The ALU has math, logic, and some designed operations in the computer. In this lab, you will learn:

1. A synthesizable Verilog HDL code of ALU and the corresponding test bench.
2. How to run VCS simulator.

Data Preparation

1. Upload your files (Lab1.tar) to your work directory.
2. Decompress Lab1.tar with following command:

```
tar -xvf Lab1.tar
```

3. Lab1 files are shown as below:

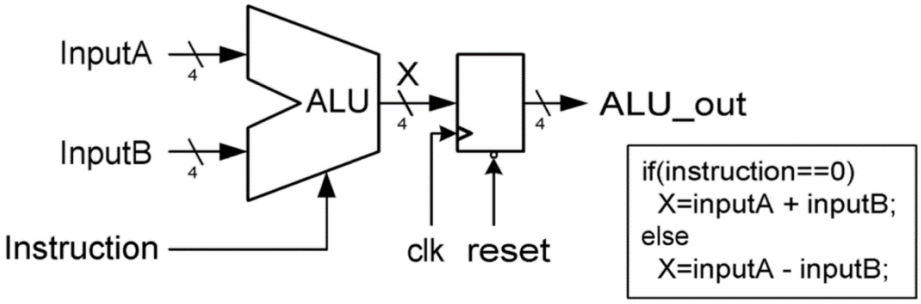
Files/Folder	Description
Lab1_alu.v	RTL code for this ALU
Lab1_test_alu.v	Test bench for this ALU
Lab1_alu_run.f	Command-line file
Lab1_test_alu_file.v	Test bench with file I/O
program.txt	Input stimulus
program_out.txt	Correct answers

Environment Setup

1. Source the default **cshrc** file:

```
source /usr/cad/synopsys/CIC/vcs.cshrc
```

Module Description



This RTL describes a simple ALU with two 4-bit input signals and two instructions. One instruction is summation, and the other one is subtraction.

1. Type this command to enter Lab1 directory:

```
cd 114_1
```

2. View the ALU module by the following command. (You can also view this file by other text editors. The file can be also modified by these text editors.)

```
gedit Lab1_alu.v &
```

3. Your module looks like this

```
module ALU(alu_out,instruction,inputA,inputB,clk,reset);
output [3:0] alu_out;
input [3:0] inputA,inputB;
input instruction;
input clk,reset;

…
```

Correct Syntax Error by VCS

1. Type this command:

```
vcs -full64 Lab1_alu.v +v2k -R
```

VCS will report one error because the signal X is not declared. Please help to add the declaration for the file Lab1_alu.v .

2. Correct the code by declaration of the signal X:

- Declare signal X:

```
reg [3:0]   alu_out;
reg [3:0]   X;          // add this line
```

3. Redo step. 1, and no error occurs now.

Run Simulation with a Test Bench by VCS

In this lab, we use file I/O to run the simulation, you can find out how test bench works with file I/O by open Lab1_test_alu_file.v and view the following two lines:

```
$ readmemb("program.txt", program);          // Input stimulus
$ readmemb("program_out.txt", answer);       // Correct answers
```

1. Type this command

```
vcs -full64 Lab1_test_alu_file.v Lab1_alu.v +v2k -R
```

2. 224 errors are found after simulation. Please read the file alu_out.txt to find out what happens. Please modify the file Lab1_alu.v to fix these errors.
   - Correct the computing error:

```
If (!instruction)
    X=inputA+inputB;
else
    X=inputA-inputB;
```

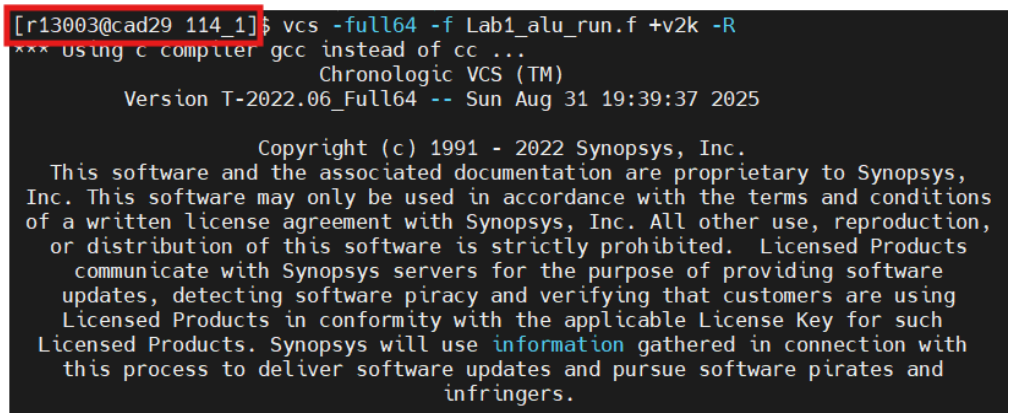
3. Run the command in step 1 again. The simulation result is correct now.
4. Instead of the command in step 1, you can also use the command-line file to run simulation by typing this command

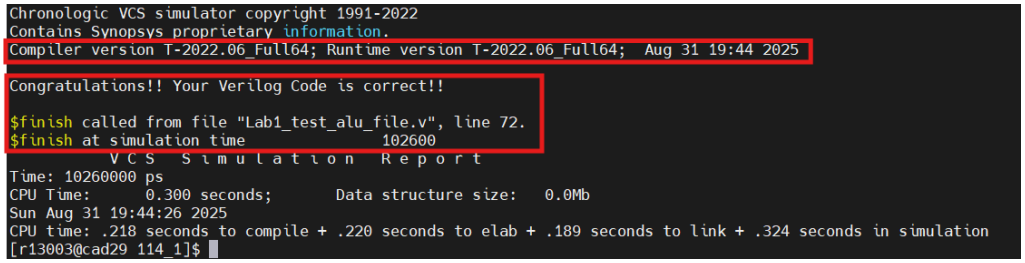
```
vcs -full64 -f Lab1_alu_run.f +v2k -R
```

Checkpoints

Please check with TAs before leaving this lab to make sure the following goals are accomplished and to get credits.

1. Please fix the declaration error.
2. Please fix the simulation errors.
3. Show your console:





Submission

1. **Due Tuesday, Sep. 16, 20:10. No delay is allowed.**

2. Selected students need to take the snapshot of the result shown in the previous section, record it into a PDF file and submit it to NTU COOL.
   Title: studentID_lab1 (E.g. r13943003_lab1.pdf)