Computer-Aided VLSI System Design
Lab1: 4-Bit ALU with 2 Instructions

TA: 羅宇呈 f08943129@ntu.edu.tw

Introduction

The Arithmetic logic unit (ALU) is one of the 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 NC-Verilog 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:

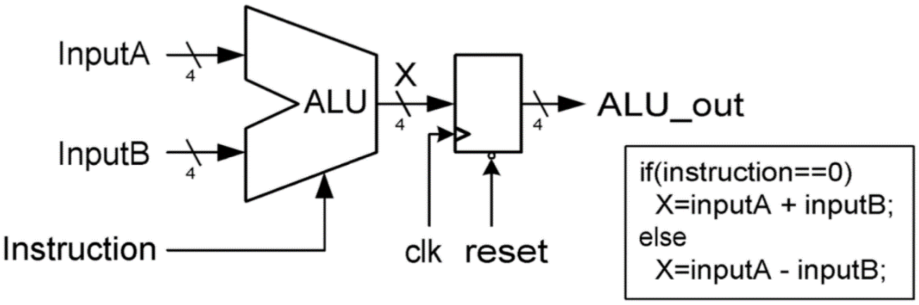
Files/Folder	Description
Lab1_alu.v	RTL code for this ALU
Lab1_test_alu.v	Test bench for this ALU
Lab1_alu_run.f	Command-line file
Lab1_test_alu_file.v	Test bench with file I/O
program.txt	Input stimulus
program_out.txt	Correct answers

Environment Setup

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

```
source /usr/cadence/cshrc
```

Module Description



```
if(instruction==0)
  X=inputA + inputB;
else
  X=inputA - inputB;
```

This RTL describes a simple ALU with two 4-bit input signals and 2instructions. One instruction is summation, and the other one is subtraction.

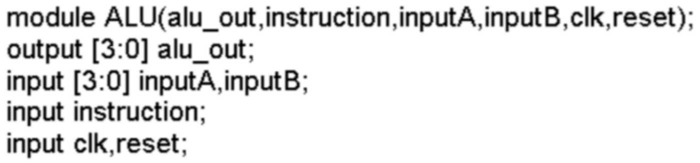
1. Type this command to enter Lab1 directory:

```
cd 111_1
```

2. View the ALU module by the following command. (You can also modify the file by this editor)

```
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 NC-Verilog

1. Type this command:

```
ncverilog Lab1_alu.v
```

NC-Verilog 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 NC-Verilog

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

```
ncverilog Lab1_test_alu_file.v Lab1_alu.v
```

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;
```

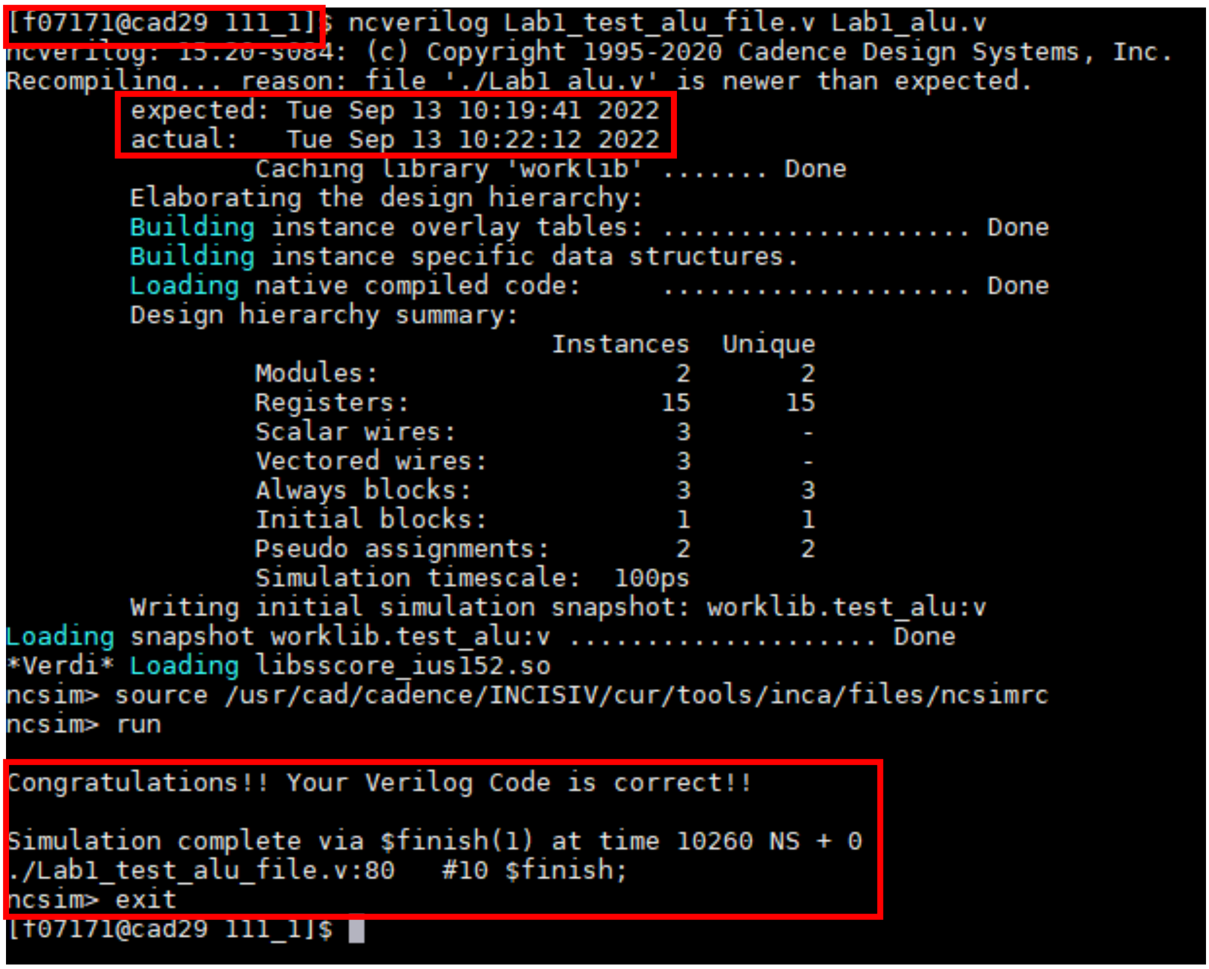
3. Run the command in step 1 again. The simulation result is correct now.
4. Instead of the command in step 1, you can also use the command-line file to run simulation by typing this command

```
ncverilog -f Lab1_alu_run.f
```

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:



```
[f07171@cad29 111_1]$ ncverilog Lab1_test_alu_file.v Lab1_alu.v
ncverilog: 15.20-s084: (c) Copyright 1995-2020 Cadence Design Systems, Inc.
Recompiling... reason: file './Lab1_alu.v' is newer than expected.
        expected: Tue Sep 13 10:19:41 2022
        actual:   Tue Sep 13 10:22:12 2022
                Caching library 'worklib' ....... Done
        Elaborating the design hierarchy:
        Building instance overlay tables: .................... Done
        Building instance specific data structures.
        Loading native compiled code:     .................... Done
        Design hierarchy summary:
                                  Instances  Unique
                Modules:                  2       2
                Registers:               15      15
                Scalar wires:             3       -
                Vectored wires:           3       -
                Always blocks:            3       3
                Initial blocks:           1       1
                Pseudo assignments:       2       2
                Simulation timescale:  100ps
        Writing initial simulation snapshot: worklib.test_alu:v
Loading snapshot worklib.test_alu:v .................... Done
*Verdi* Loading libsscore_ius152.so
ncsim> source /usr/cad/cadence/INCISIV/cur/tools/inca/files/ncsimrc
ncsim> run

Congratulations!! Your Verilog Code is correct!!

Simulation complete via $finish(1) at time 10260 NS + 0
./Lab1_test_alu_file.v:80   #10 $finish;
ncsim> exit
[f07171@cad29 111_1]$ 
```

Submission

1. **Due Tuesday, Mar. 10th, 23:59**

2. Take the snapshot of the result shown in previous section, record them to a single word file and submit to ntu cool
   Title: CVSD_Lab1_studentID (E.g. CVSD_Lab1_F08943129.docx)