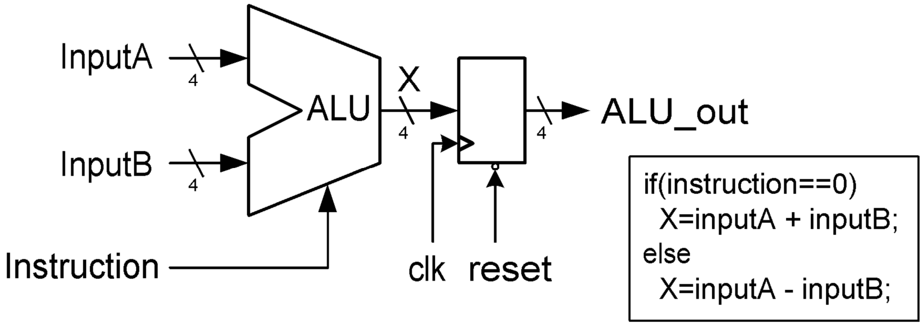
Verilog Lab 1 – **4-Bit ALU with 2 Instructions**

I. 4-Bit ALU with 2 Instructions:



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

II. Objectives:

In this lab, you will learn:

1. A synthesizable Verilog HDL code and the corresponding test bench.
2. How to run NC-Verilog simulator?

III. Lab Files:

1. Upload your files (Lab1.tar) into your work directory by FTP.
2. Type this command to extract your files.

   tar -xvf Lab1.tar

3. Lab1 files are shown as below:

Filename	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

4. Your ALU 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;

…
```

5. Type this command to enter Lab1 directory:

   cd Lab1

IV. Environment Setup:

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

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

V. Module Description:

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

VI. Check Verilog Code by NC-Verilog:

1. Type this command

   ```
   ncverilog    Lab1_alu.v
   ```

2. 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 .
   - Type this command to enter text editor

     ```
     gedit    Lab1_alu.v
     ```

   - Declare signal X:

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

3. Run the command in step 1 again. No error should occur now.

VII. Run simulation with a test bench by NC-Verilog:

1. Type this command

   ```
   ncverilog    Lab1_test_alu.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.
   - Type this command to enter text editor

     ```
     gedit Lab_alu.v
     ```

   - Correct the computing

     ```
     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    Lab_alu_run.f
   ```

VIII. Run simulation with file I/O:

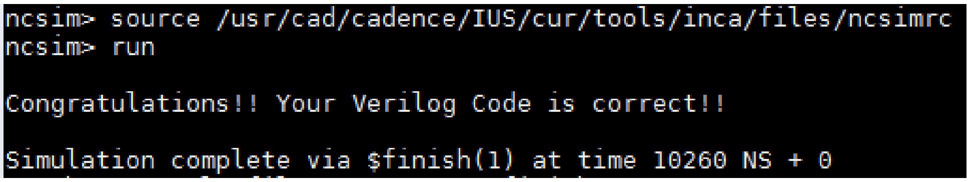
Using the test bench Lab1_test_alu_file.v for simulation. Can you find out how this test bench works with file I/O?

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

IX. 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 in Section VI.
2. Please fix the simulation errors in Section VII.



```
ncsim> source /usr/cad/cadence/IUS/cur/tools/inca/files/ncsimrc
ncsim> run

Congratulations!! Your Verilog Code is correct!!

Simulation complete via $finish(1) at time 10260 NS + 0
```

END of LAB

Creator:
1st Edition: Chao-Tsung Huang, 2002
2nd Edition: Yu-Lin Chang, 2004
3rd Edition: Yu-Lin Chang, 2005
4th Edition: En-Jui Chang, 2010
5th Edition: En-Jui Chang, 2011
6th Edition: Yu-Min Lin,2013
7th Edition: Chou Ching Yao,2015