Computer Organization, Spring 2020

Lab 2: 32-bit ALU

Due: 2020/04/16

1. Goal

The goal of this LAB is to implement a 32-bit ALU (Arithmetic Logic Unit). ALU is the basic computing component of a CPU. Its operations include AND, OR, addition, subtraction, etc. This series of LABs will help you understand the CPU architecture. LAB 3 will be reused; you will use this module in the later LABs.

2. HW Requirement

(1) Please use ModelSim/ISE as you HDL simulator.
(2) Please attach your names and student IDs as comment at the top of each file.
(3) Please use the Testbench we provide you.
(4) The names of top module and IO ports must be named as follows:
Top module: alu.v

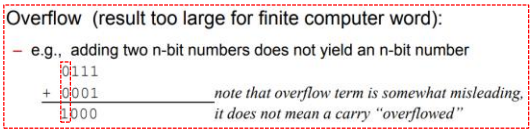
```
module alu(
          rst_n,          // negative reset            (input)
          src1,           // 32 bits source 1          (input)
          src2,           // 32 bits source 2          (input)
          ALU_control,    // 4 bits ALU control input  (input)
          result,         // 32 bits result            (output)
          zero,           // 1 bit when the output is 0, zero must be set (output)
          cout,           // 1 bit carry out           (output)
          overflow        // 1 bit overflow            (output)
          );
```

ALU starts to work when the signal rst_n is 1, and then catches the data from src1 and src2.
In order to have a good coding style, please obey the rules below:
One module in one file.
Module name and filename must be the same.
For example: The file “alu.v” only contains the module “alu”.

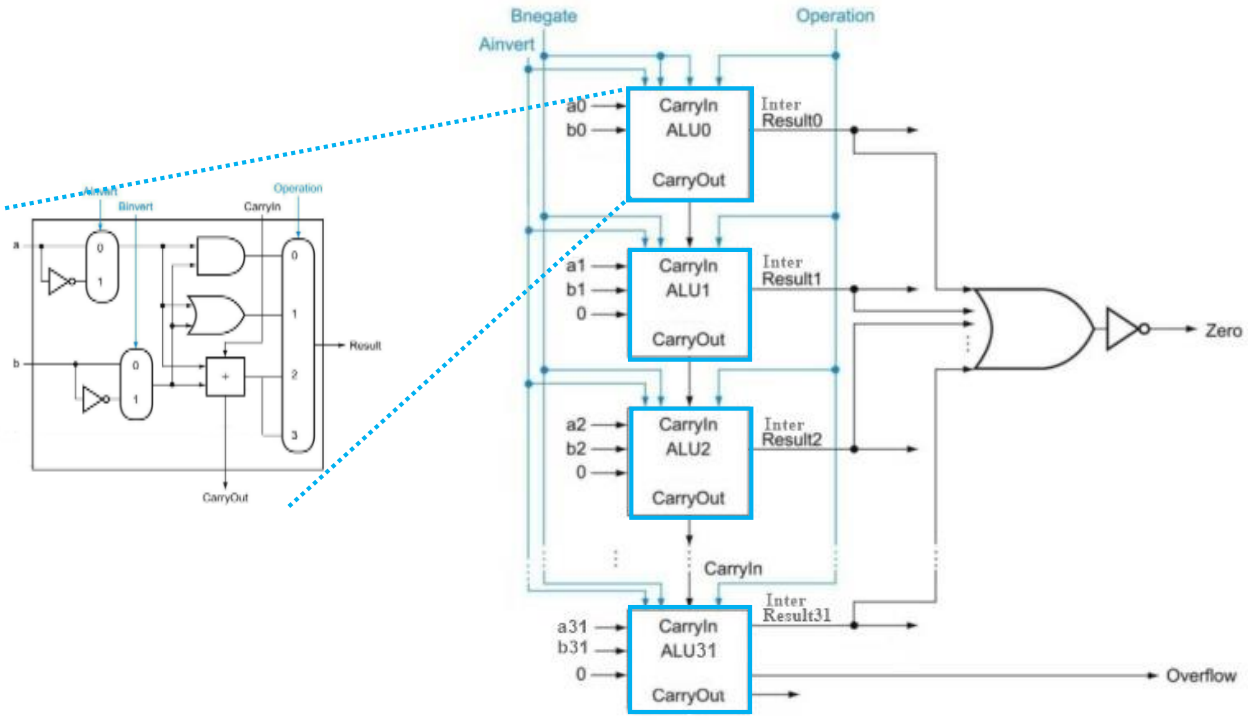
(5) Basic instruction set (70%)

ALU action	Function	ALU control input
and	AND	0000
or	OR	0001
add	Addition	0010
sub	Subtract	0110
slt	Set less than	0111
nor	NOR	1100
nand	NAND	1101

(6) ZCY three flags: zero, carry out, and overflow (30%)
zero: must be set when the output is 0
cout: must be set when carry out is 1
overflow: must be set when overflow happens



3. Architecture Diagram



4. Grade

(1) Basic instructions score: 70 points.
(2) ZCY three flags score: 30 points.
(3) Late submission: 10 percent penalty per day
(4) No plagiarism, or you will get 0 point.

5. Hand in

(1) Zip your folder and name it as "ID.zip" (e.g. 0816001.zip) before uploading to newe3. Other filenames and formats such as *.rar and *.7z are NOT accepted! Multiple submissions are accepted, and the version with the latest time stamp will be graded.
(2) Please include ONLY Verilog source codes (*.v) and your report (*.docx or *.pdf) in the zipped folder.
(3) Rename your report to "Report_ID.docx" (e.g. Report_0816001.docx) or "Report_ID.pdf" (e.g. Report_0816001.pdf).

6. How to test

The function of testbench is to read input data automatically and output erroneous data. Please put all the .txt files and project in the same folder, after simulation finishes, you will get some information.

Partial error:

```
VSIM 49> run -all
# ******************************************************
# *              PATTERN RESULT TABLE                  *
# ******************************************************
# * PATTERN *              Result              * ZCV *
# ******************************************************
# *   6     *  User   :  00000001              * 000 *
# *         *  Correct:  00000001              * 001 *
# ******************************************************
```

Or All cases pass:

```
VSIM 55> run -all
# ******************************************************
# *              PATTERN RESULT TABLE                  *
# ******************************************************
# * PATTERN *              Result              * ZCV *
# ******************************************************
# *     Congratulation! All result are correct!      *
# ******************************************************
```

7. Q&A

For any questions regarding Lab 2, please contact

張祐銘 yumingchang.cs03@g2.nctu.edu.tw

賴柏宏 bhbruce.cs07g@nctu.edu.tw

鄭俊賢 petertay1996.cs08g@nctu.edu.tw