Lab 2 ALU

ALU is used to operate some calculations for processor. This lab requires you to simulate how ALU doing these operations. The required ALU must have the following operations: add, sub, shift-left and shift-right. “add” instruction will add the values of two registers A and B, and then store in the C register. “sub” instruction will subtract the value in register B from the value in register A, and then store the result in the register C. And shift instructions (including “shift left” and “shift right”) will shift the value in register A and then store in register B.

Instruction format

Instruction	OP			
ADD	00100000	Rs	Rt	Rd
SUB	00010000	Rs	Rt	Rd
SHIFT LEFT	00001000	Rs	shmt	Rd
SHIFT RIGHT	00000100	Rs	shmt	Rd

`ADD: Rd = Rs + Rt`

`SUB: Rd = Rs - Rt`

`SHIFT LEFT:` shift the value in Rs left shmt bits and then stores the result in Rd.

`SHIFT RIGHT:` shift the value in Rs right shmt bits arithmetically and then stores the result in Rd.

I/O Pins

```
Module name: ALU
  Input：
    ALUCtl[1:0]
    A[31:0]  B[31:0]
  Output：
    ALUout[31:0]
```

Notice:

1. The A here is the value of register Rs, and B is the value of register Rt in “ADD” and “SUB” instructions and shmt in shift instructions.
2. Control signals of ALUCtl for the instructions above:

	ADD	SUB	SHIFT LEFT	SHIFT RIGHT
instructions	2’b00	2’b01	2’b10	2’b11

3. You need to complete another module called decoder to transform instructions into the format of ALU.

Test Instructions

Initial values of register files R0~R3 are 1,2,3,4

```
00001000 00000000 00000010 00000000 (SL R0 2 R0)
00000100 00000011 00000010 00000011 (SR R3 2 R3)
00010000 00000000 00000001 00000010 (SUB R0 R1 R2)
00100000 00000010 00000011 00000001 (ADD R2 R3 R1)
```

※ We will have multiple hidden testcases to test your ALU, and you need to pass all of the cases to get the full grade.

Final result

R0	R1	R2	R3
4	3	2	1

Requirements

1. Note that the values in registers may be negative (2's complement), but the shift amount shmt in shift instructions would be a non-negative integer.
2. Environment: Xilinx v13.2, and one person per group.
3. Requested files: codes, report. You don't need to submit the whole xilinx project.
4. ALU's Input / Output pin names must be the same in I/O pins
5. Deadline : **2011/11/03**
6. If you have any question, please contact with TAs:

電資 712
廖世滄 ce752014liao@gmail.com
陳玠竹 testbibibird@gmail.com
電資 713
蘇宏彥 lionking.cs00g@nctu.edu.tw
朱浩賢 yosa21erod@gmail.com

Handout details：

1. Please zip all of the requested files (including codes, reports…) into one file (.rar, .zip), and filename format is (your student ID)_(index of this lab)_v(your homework version). Ex: 9817000_2_v1.zip. Note that do not paste your codes on your documentation. Please submit the original .v files.
2. Please submit your homework before due time (23:59:59 on due day). The submission time is based on e3 platform.
3. If any violation of the rules above is found, -5 grades per violation.

Late submission：

1. Your grade will be 10% discount for late submission for 1~2 days, 15% discount for 3~4 days, 20% discount for 5~6 days and 40% discount for more than 6 days.
2. **Notice: No cheating! Or you grade will be 0!**