LAB 1

Digital System Design - 2020 spring

Sungkyunkwan University

IDEA Lab

Goals

- Practice designing **combinational logic circuits** in Verilog HDL

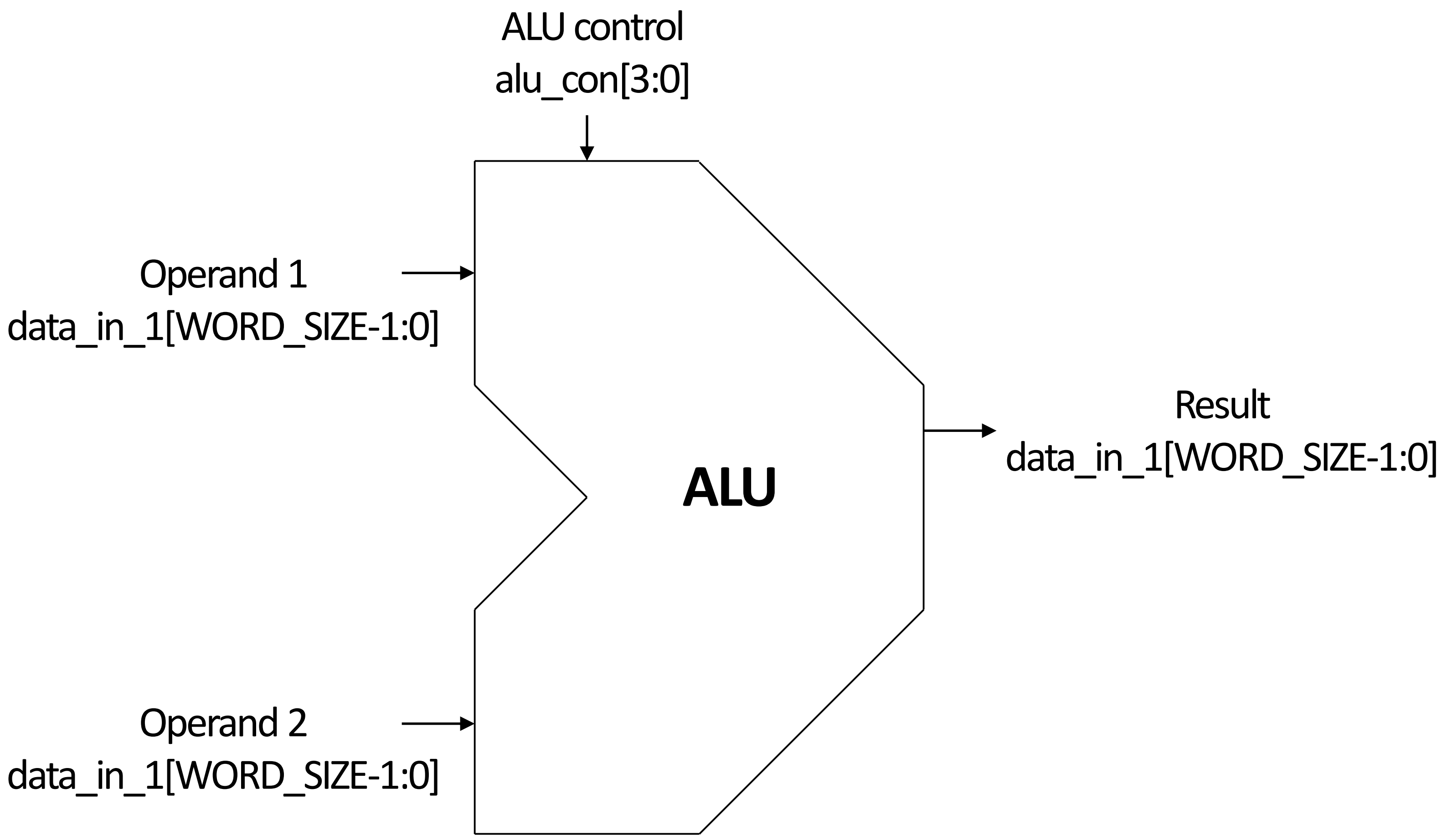
- Practice designing a **hierarchical structural model** of combinational logic circuits

Objectives

- Design submodules (Adder, Subtractor, Bitwise Operator)

- **Design a ALU using submodules**

Block Diagram



Specification

- **ALU must support following operations**
  - Bitwise OR / AND / NOR
  - Signed Addition
  - Signed Subtraction

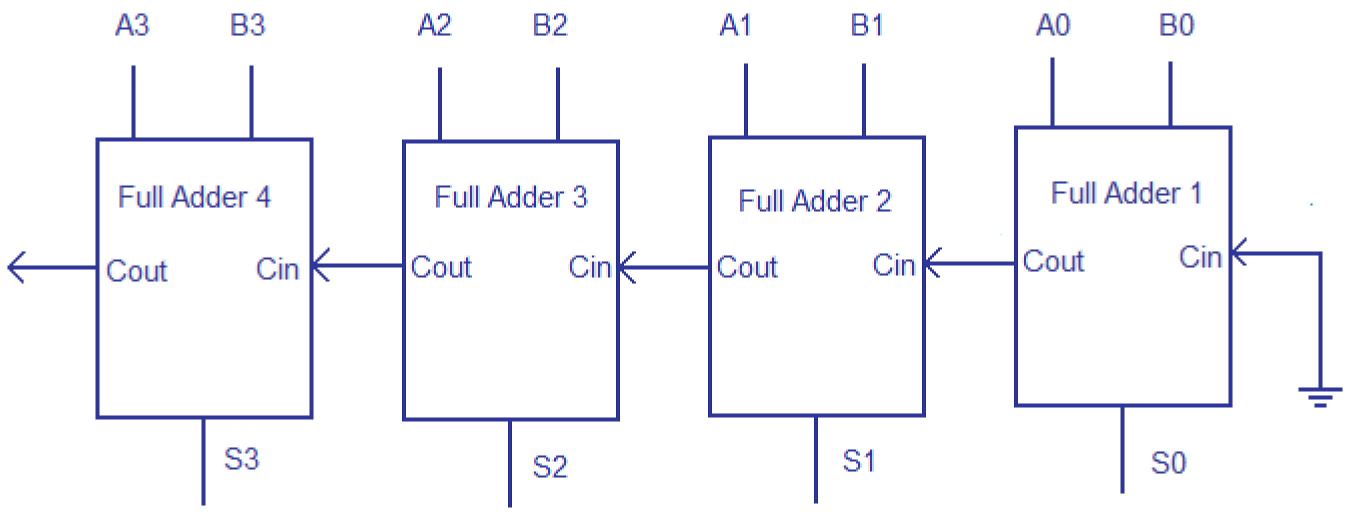
Opcode Table

Opcode	Instruction
0010	Signed Addition
0110	Signed Subtraction
0000	Bitwise AND
0001	Bitwise OR
1111	Bitwise NOR

* subtraction operates data_in_1 – data_in_2

Opcode Table

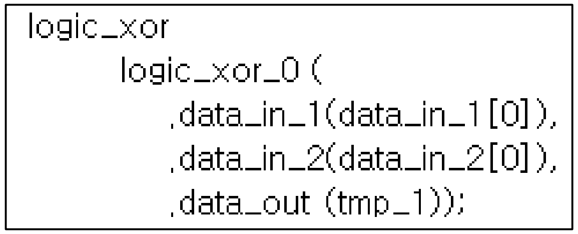
- **Adder and Subtractor should be coded using Ripple carry adder which can be simply realized.**



4 bit ripple carry adder

Restriction

- You must use **submodules** that you will design to implement a complete ALU.
  - Hierarchical Structural Model
    - Use Only Gate modules what we give to you in the code in “logic_gate.v”
    - logic_or, logic_and, logic_xor
    - Ex)



```
logic_xor
        logic_xor_0 (
            .data_in_1(data_in_1[0]),
            .data_in_2(data_in_2[0]),
            .data_out (tmp_1));
```

  - No Behavioral Description(including concatenation)

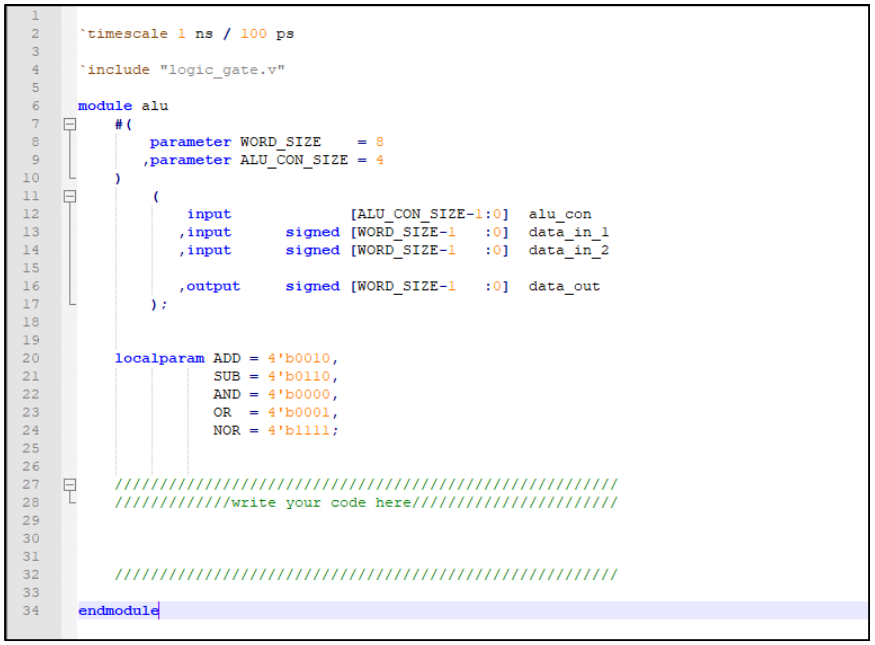
Ex) ~~S <= a+b;~~ ~~S <= a * b;~~ ~~S <= {a, b};~~

you can only use statements below!

```
assign A = B;
assign A = ~B;
assign A = (B==1)?C:D;
```

Restriction

- You can only modify “alu.v” file



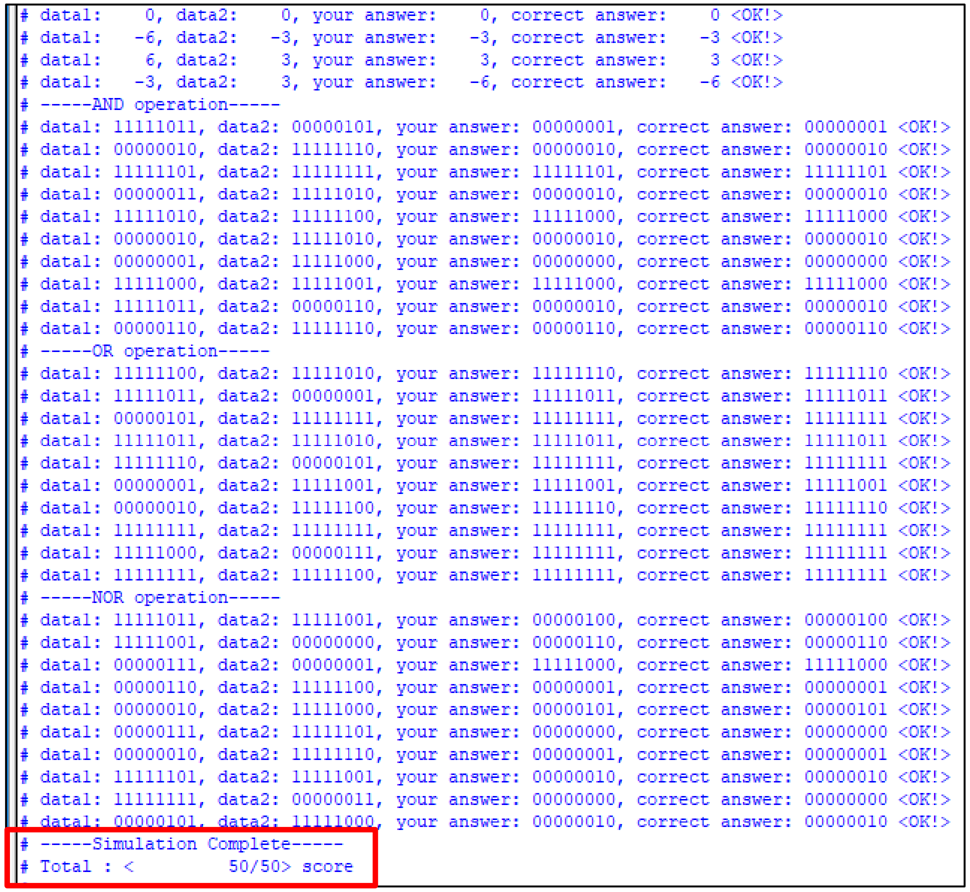
```

`timescale 1 ns / 100 ps

`include "logic_gate.v"

module alu
    #(
        parameter WORD_SIZE    = 8
       ,parameter ALU_CON_SIZE = 4
    )
        (
            input              [ALU_CON_SIZE-1:0]  alu_con
           ,input       signed [WORD_SIZE-1    :0]  data_in_1
           ,input       signed [WORD_SIZE-1    :0]  data_in_2

           ,output      signed [WORD_SIZE-1    :0]  data_out
        );


    localparam ADD = 4'b0010,
               SUB = 4'b0110,
               AND = 4'b0000,
               OR  = 4'b0001,
               NOR = 4'b1111;


    ////////////////////////////////////////////////////////
    /////////////write your code here//////////////////////



    ////////////////////////////////////////////////////////

endmodule
```

- You can use any other wires but not registers.

IDEA Lab

Test bench

- Result of test bench
  - it test 10 random input cases with each operation
  - if the whole cases are correct, you will get 50/50 score



```
# data1:    0, data2:    0, your answer:    0, correct answer:    0 <OK!>
# data1:   -6, data2:   -3, your answer:   -3, correct answer:   -3 <OK!>
# data1:    6, data2:    3, your answer:    3, correct answer:    3 <OK!>
# data1:   -3, data2:    3, your answer:   -6, correct answer:   -6 <OK!>
# -----AND operation-----
# data1: 11111011, data2: 00000101, your answer: 00000001, correct answer: 00000001 <OK!>
# data1: 00000010, data2: 11111110, your answer: 00000010, correct answer: 00000010 <OK!>
# data1: 11111101, data2: 11111111, your answer: 11111101, correct answer: 11111101 <OK!>
# data1: 00000011, data2: 11111010, your answer: 00000010, correct answer: 00000010 <OK!>
# data1: 11111010, data2: 11111100, your answer: 11111000, correct answer: 11111000 <OK!>
# data1: 00000010, data2: 11111010, your answer: 00000010, correct answer: 00000010 <OK!>
# data1: 00000001, data2: 11111000, your answer: 00000000, correct answer: 00000000 <OK!>
# data1: 11111000, data2: 11111001, your answer: 11111000, correct answer: 11111000 <OK!>
# data1: 11111011, data2: 00000110, your answer: 00000010, correct answer: 00000010 <OK!>
# data1: 00000110, data2: 11111110, your answer: 00000110, correct answer: 00000110 <OK!>
# -----OR operation-----
# data1: 11111100, data2: 11111010, your answer: 11111110, correct answer: 11111110 <OK!>
# data1: 11111011, data2: 00000001, your answer: 11111011, correct answer: 11111011 <OK!>
# data1: 00000101, data2: 11111111, your answer: 11111111, correct answer: 11111111 <OK!>
# data1: 11111011, data2: 11111010, your answer: 11111011, correct answer: 11111011 <OK!>
# data1: 11111110, data2: 00000101, your answer: 11111111, correct answer: 11111111 <OK!>
# data1: 00000001, data2: 11111001, your answer: 11111001, correct answer: 11111001 <OK!>
# data1: 00000010, data2: 11111100, your answer: 11111110, correct answer: 11111110 <OK!>
# data1: 11111111, data2: 11111111, your answer: 11111111, correct answer: 11111111 <OK!>
# data1: 11111000, data2: 00000111, your answer: 11111111, correct answer: 11111111 <OK!>
# data1: 11111111, data2: 11111100, your answer: 11111111, correct answer: 11111111 <OK!>
# -----NOR operation-----
# data1: 11111011, data2: 11111001, your answer: 00000100, correct answer: 00000100 <OK!>
# data1: 11111001, data2: 00000000, your answer: 00000110, correct answer: 00000110 <OK!>
# data1: 00000111, data2: 00000001, your answer: 11111000, correct answer: 11111000 <OK!>
# data1: 00000110, data2: 11111100, your answer: 00000001, correct answer: 00000001 <OK!>
# data1: 00000010, data2: 11111000, your answer: 00000101, correct answer: 00000101 <OK!>
# data1: 00000111, data2: 11111101, your answer: 00000000, correct answer: 00000000 <OK!>
# data1: 00000010, data2: 11111110, your answer: 00000001, correct answer: 00000001 <OK!>
# data1: 11111101, data2: 11111001, your answer: 00000010, correct answer: 00000010 <OK!>
# data1: 11111111, data2: 00000011, your answer: 00000000, correct answer: 00000000 <OK!>
# data1: 00000101, data2: 11111000, your answer: 00000010, correct answer: 00000010 <OK!>
# -----Simulation Complete-----
# Total : <          50/50> score
```

Submission

- Submit your ***.v file** and **report** on iCampus
- Report must include questions below
  - 1. Explain your code.
  - 2. If AND-gate has 100ps delay, OR-gate has 50ps delay, and XOR-gate has 150ps delay, calculate the critical path delay of your ALU.
  - 3. Explain the disadvantage of Ripple Carry Adder.

  format : name_ID.v & name_ID.pdf

  ex) 홍길동_2020000000.v, 홍길동_2020000000.pdf

Grade

- **Function test (50)**
- **Work correctly even if WORD_SIZE is changed (20)**
- **Coding style(readability&clarity) (30)**
- **Report (50)**
- **You don't need to write comment in your code**