CSE 331 Computer Organization

Project 2 – ALU with Structural Verilog

Due date: November 16, Friday 23:59 (Moodle)

Write a structural Verilog on Altera Quartus II tool to implement a 32-bit ALU. Only structural Verilog is allowed, dataflow and behavioral Verilog is not allowed. This means you cannot use `assign`, `if-else`, `always`, `?:` and etc.

Use hierarchy in your project. For instance, you can design a 2X1 MUX and using three 2X1 MUXes you can design a 4X1 MUX etc.

The primitive gates that you can use in structural Verilog are listed below:

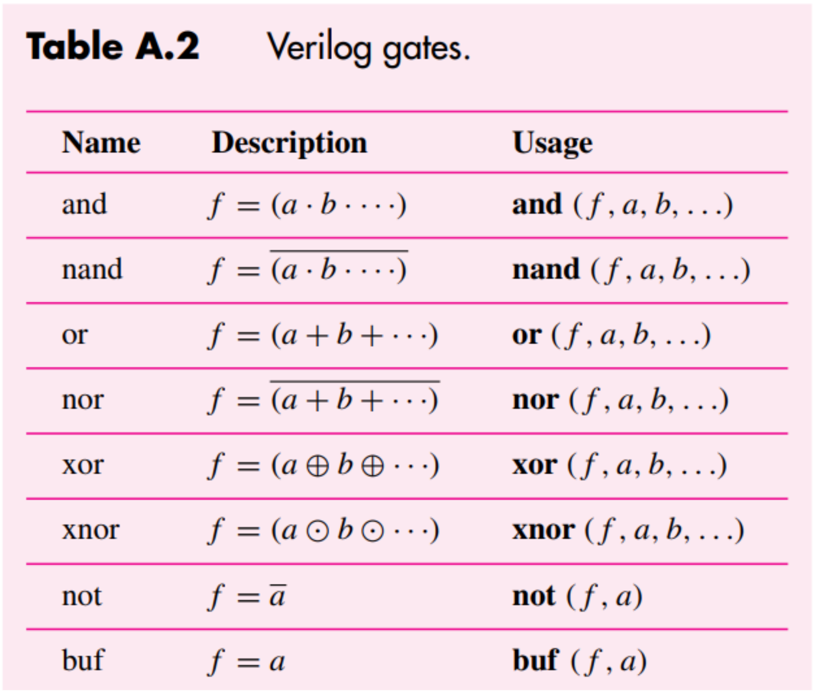

Table A.2 Verilog gates.

Name	Description	Usage
and	$f = (a \cdot b \cdot \cdots)$	**and** $(f, a, b, \ldots)$
nand	$f = \overline{(a \cdot b \cdot \cdots)}$	**nand** $(f, a, b, \ldots)$
or	$f = (a + b + \cdots)$	**or** $(f, a, b, \ldots)$
nor	$f = \overline{(a + b + \cdots)}$	**nor** $(f, a, b, \ldots)$
xor	$f = (a \oplus b \oplus \cdots)$	**xor** $(f, a, b, \ldots)$
xnor	$f = (a \odot b \odot \cdots)$	**xnor** $(f, a, b, \ldots)$
not	$f = \overline{a}$	**not** (f, a)
buf	$f = a$	**buf** (f, a)

The ALU will get two 32-bit numbers **A** and **B**, and a 3-bit **S** (select) signal as inputs and a 32-bit **R** signal as output. The ALU perform the following operations:

ALU select (S)	Operation
000	R = A AND B
001	R = A OR B
010	R = A + B
011	R = A XOR B
100	R = A - B
101	R = A >> B (arithmetic shift right)
110	R = A << B (shift left)
111	R = A NOR B

You should also write a Verilog test bench to simulate your design with Modelsim. In your tests, you should test all eight scenarios shown by the above table for three different (A, B) input pairs. So you will test 3 X 8 = 24 different cases.

You should write a report (25%) including:

1. Your schematic designs for all modules.
2. Your Verilog modules and their description.
3. Modelsim Simulation results.
4. If not compiling or partial working the explanation of which parts work which parts do not.

You will submit your report, your full project as a zip file to Moodle.

Rules:

1. Behavioral or Dataflow Verilog are not allowed.
2. Not compiling or not simulating solutions can at most get 25pts.
3. You have to use Quartus II tool referred in Moodle.
4. Each day of late submission will get 25 point loss.
5. Write 32-bit adder and shifters as modules.
6. The name of your top module should be alu32.
7. Implementation of Zero and Overflow bit gets you extra 15pts only if all your design is working perfectly.

Hint: Start with drawing schematic on paper for each module. Do not hesitate to write 32 lines of logic expressions for each bit of one or two 32-bit numbers whenever required.

Honor code: It is not a group project. Do not take any code from Internet. Any cheating means at least -100 for both sides. Do not share your codes and design to any one in any circumstance. Be honest and uncorrupt <u>not to win</u> but because <u>it is RIGHT</u>!