NANYANG TECHNOLOGICAL UNIVERSITY

CZ3001

LAB 1 (50 POINTS)

REPORT DUE 22/09/2013, 11:59PM

I. ARITHMETIC LOGIC UNIT (ALU) SPECIFICATIONS

An arithmetic logic unit is the main piece of hardware in a CPU that performs the computation portion of all the instructions. For this first assignment, we will build, in Verilog an ALU that performs the computation for eight arithmetic and logical instructions. The ALU should be entirely combinational logic, with no storage or sequential operation.

The eight computation types are: ADD, SUB, AND, OR, SLL, SRL, SRA, and RL. The operation of the instructions is described in Table 1.

Table 1 - Description of ALU Operations

Computation Type	Equation	Description
ADD	A+B	Addition (A + B) in 2's Complement format
SUB	A-B	Subtraction (A – B) in 2's Complement format
AND	A&B	Logical (bit-wise) AND of A, B
OR	A|B	Logical (bit-wise) OR of A, B
SLL	A<<Imm	Shift left logical, operand A shift to left by 4-bit immediate (Imm), with 0 shift in
SRL	A>>Imm	Shift right logical, operand A shift to right by 4-bit immediate (Imm), with 0 shift in
SRA	A>>Imm (with A[15] shifted in)	Shift right arithmetic, operand A shift to right by 4-bit immediate, with sign-bit preserved
RL	A rot Imm	Rotate left, operand A rotated to left by 4-bit immediate input

The interface to the ALU consists of 2 16-bit data inputs, A and B, one 3-bit input op to signify which of the 8 instructions should be performed, one 4-bit input imm used with the shift and rotate instructions to determine the shift (or rotate) amount, and one 16-bit output 'Out'. The information is also listed in Table 2. The encoding of the 8 instructions is listed in Table 3.

Table 2 - Port List Specification

Port Name	Port Direction	Size	Description
A	Input	16-bit	First operand
B	Input	16-bit	Second operand for ADD, SUB, AND & OR instructions
op	Input	3-bit	Selects which operation to be performed
imm	Input	4-bit	Selects shift (rotate) amount for the SLL, SRL, SRA, RL instructions
Out	Output	16-bit	Output of the operation

Table 3 - ALU operation encoding

Operation	'op' value
AND	000
SUB	001
AND	010
OR	011
SLL	100
SRL	101
SRA	110
RL	111

II. ALU IMPLEMENTATION

For this assignment, you will implement the ALU as specified above in Verilog. You must follow the same port naming, size and order conventions as listed above. You should create your own test bench for the ALU and test appropriate input combinations in order to prove to yourself that your ALU is operating correctly according to the specifications. The grading for this assignment will be as follows:

50% - A written report describing how you implemented the ALU, with annotated simulations of the test cases you attempted in order to verify the operation.
50% - Verilog correctness; we will use a testbench and test-cases that will not be provided in advance to test the correctness of your output.

III. REFERENCE LINK FOR VERILOG

1. http://www.asic-world.com/verilog/
2. http://web.engr.oregonstate.edu/~traylor/ece474/lecture_verilog/beamer/verilog_intro.pdf
   http://web.engr.oregonstate.edu/~traylor/ece474/lecture_verilog/beamer/verilog_modules.pdf
   http://web.engr.oregonstate.edu/~traylor/ece474/lecture_verilog/beamer/verilog_data_types.pdf
   http://web.engr.oregonstate.edu/~traylor/ece474/lecture_verilog/beamer/verilog_number_literals.pdf
   http://web.engr.oregonstate.edu/~traylor/ece474/lecture_verilog/beamer/verilog_operators.pdf
3. File Handling: http://www.asic.co.in/Index_files/verilog_files/File_IO.htm