CSE/ELE 664 – Intro. To SoC Design, Fall 2021
Project part1 – 8-bit ALU Design

CONTENT

In this part of the project, you will design an 8-bit ALU at RT level using Verilog, and test it using ModelSim Simulation. This ALU will be used as one of the components in the 8-bit microcontroller that you will design in the first course project.

1. 8-bit ALU

The 8-bit ALU has two 8-bit inputs, A[7:0] and B[7:0]. It supports 7 operations: addition, subtraction, NOR, shift left, shift right, load value, and reset value. It has two sets of 2-bit control signals, ALU_sel and load_shift. It has an 8-bit output: result[7:0], and two one bit outputs: carry (cout) and zero (zout). The output “cout” will be asserted when there is carry or borrow during the addition or subtraction. The output “zout” will be asserted when the result[7:0] is 0. The following table specifies the input/output and operations of the ALU.

A. Inputs and outputs list

The table for inputs and output is as followed. × means not-care

A[7:0	B[7:0]	ALU_sel[1:0]	load_shift[1:0]	result[7:0]	Function
A[7:0]	B[7:0]	10	xx	A+B	ADD
A[7:0]	B[7:0]	11	xx	A-B	SUB
A[7:0]	B[7:0]	01	xx	A NOR B	NOR
A7 A6 … A1 A0	×	00	11	0 A7 … A2 A1	SHR
A7 A6 … A1 A0	×	00	01	A6 A5 … A0 0	SHL
A[7:0]	×	00	10	A	LD
×	×	00	00	0	RST

B. Names of inputs and outputs

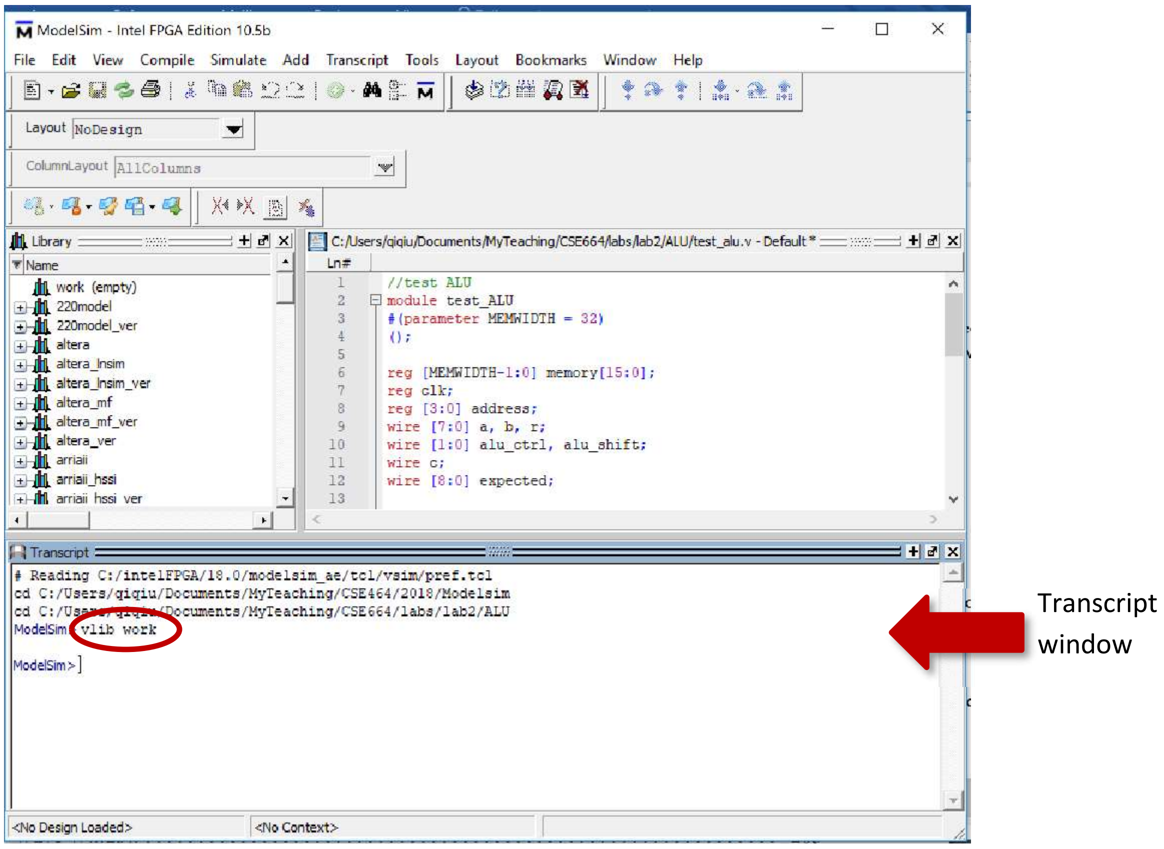
The following is the names of inputs and outputs for ALU_8bit. You MUST use these names in your design.

	Data Inputs		Control Inputs		Outputs		
name	A[7:0]	B[7:0]	ALU_sel[1:0]	load_shift[1:0]	result[7:0]	cout	zout

2. Testing and Simulation

Download the ALU.zip to your working directory. Unzip it and you will see a directory called ALU. There are three files in it: ALU.v, test_alu.v and test_alu.txt. ALU.v gives an empty design of ALU module, which you need to complete. The file test_alu.v gives the test bench for this module, and test_alu.txt is the test vector that will be loaded into test_alu.v.

Start ModelSim – Intel FPGA edition. Use **File > Change Directory** to change your working directory to your_directory/ALU/. Open ALU.v and complete the design. Take a look of test_alu.v. Do NOT change test_alu.v.



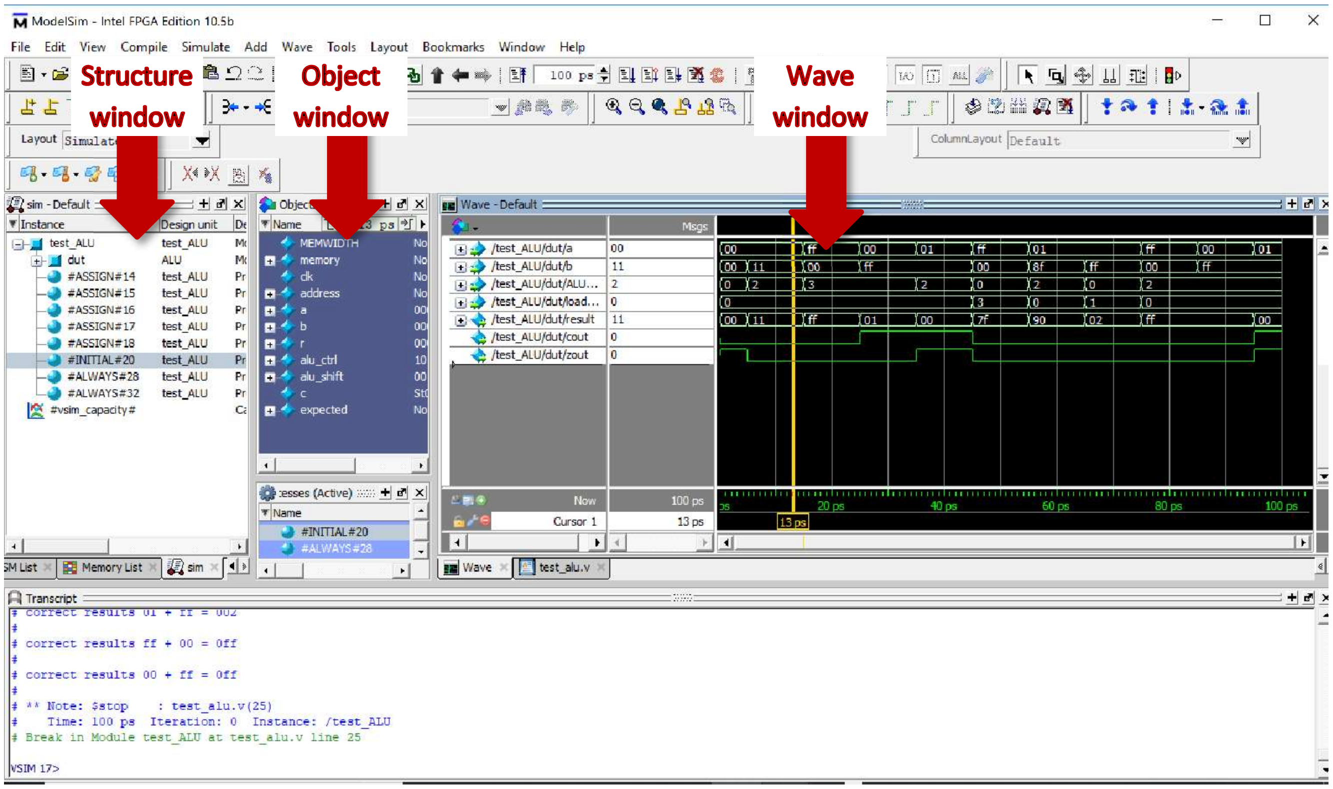
```
//test ALU
module test_ALU
#(parameter MEMWIDTH = 32)
();

reg [MEMWIDTH-1:0] memory[15:0];
reg clk;
reg [3:0] address;
wire [7:0] a, b, r;
wire [1:0] alu_ctrl, alu_shift;
wire c;
wire [8:0] expected;
```

```
# Reading C:/intelFPGA/18.0/modelsim_ae/tcl/vsim/pref.tcl
cd C:/Users/qiqiu/Documents/MyTeaching/CSE464/2018/Modelsim
cd C:/Users/qiqiu/Documents/MyTeaching/CSE664/labs/lab2/ALU
ModelSim> vlib work

ModelSim>
```

In the transcript window, type in the command “vlib work”. This will create a work library in your current directory. Then type in the command: “vlog ALU.v”, “vlog test_alu.v”. These commands compiles the Verilog file. Finally type in the command: “vsim test_ALU”, where test_ALU is the top level module of the testbench, to start simulation. The structure window, object window, wave window will appear. Navigate down the structure of the design and drag signals that you want to monitor to the wave window one by one. Use **File > Save Format** to save the selected signal to wave.do. Next time when you re-open the ModelSim in the same directory, you only need to run “run wave.do” in the transcript window to reload all the selected signals.



```
# correct results 01 + ff = 002
#
# correct results ff + 00 = 0ff
#
# correct results 00 + ff = 0ff
#
# ** Note: $stop    : test_alu.v(25)
#    Time: 100 ps  Iteration: 0  Instance: /test_ALU
# Break in Module test_ALU at test_alu.v line 25

VSIM 17>
```

Run the simulation by typing “run 100”, and observe the display for any possible errors.

3. What to Turn In

Complete alu.v.

Modify the test_alu.v and test_alu.txt to achieve the following:

1. Add test vectors to test all functions, including NOR, SHR, SHL, LD
2. Make zout and cout both high at time 51 ps;
3. Make zout low and cout high at time 101 ps;
4. Make zout high and cout low at time 151 ps;
5. Test an addition at time 31 ps followed by a shift left.
6. Test an subtraction at time 81 ps followed by a NOR.

Hand in all three files: alu.v, test_alu.v and test_alu.txt. No report is needed.