IC Design Lab

HW 1 Verilog

- Due on 02/24, 23:59, 10% penalty for each day of delay
- If you have any questions, please contact r10943003@ntu.edu.tw, and specify [ICDLab] before your title

I. Objectives

In this homework, we will practice a synthesizable Verilog HDL code and try to use NC-verilog to verify our RTL code. Besides, it will be requested to use nWave to double-check output result.

II. Design Files

Copy files from course website and check if you have these files：

1. HW1_alu.v
2. HW1_ alu_tb.v
3. data1.dat
4. data2.dat
5. cmd.dat
6. out_golden.dat

III. Specifications

Please design an ALU with I/O register, which conforms to the specification with the block diagram of an 8-bit ALU with I/O register given in Fig. 1 and the 3-bit instruction set in the table.1 below.

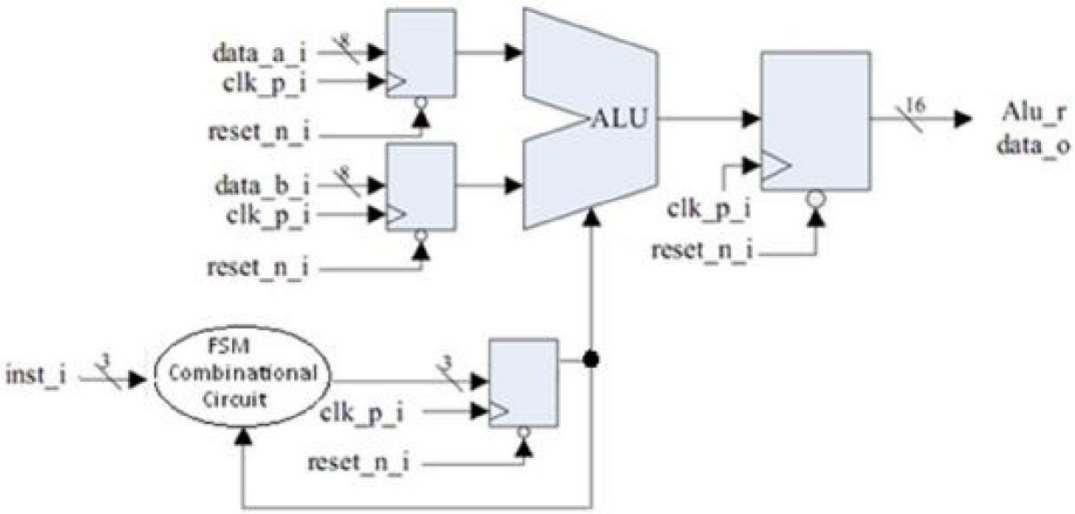

Figure1. 8-bit ALU

Comparing to the Lab1, the function has been changed, so you have to redesign the RTL code from Lab1_alu.v. Besides, we add a new condition to the ALU. In this homework, the function of the ALU to be executed depends on the **input instruction** and the **current state**. So, you should implement this ALU in the form of a Finite State Machine. When receiving a new input instruction, the ALU would consider the current state. If the state specified by this input instruction is allowed to be the next state (see the table.2), then the next state would be this new instruction. Otherwise the state will not change, i.e., the ALU will maintain the previous operation. We give an example in the table.3 and Fig.2.

State	Operation	Notes
000	Unsigned Addition	data_o=data_a_i+data_b_i
001	Unsigned Subtraction	data_o=data_b_i-data_a_i
010	Unsigned Multiplication	data_o= data_a_i*data_b_i
011	data_a_i Divide by 2	data_o=data_a_i>>1
100	Subtraction & NOT	data_o=~ (data_b_i-data_a_i)
101	XOR	data_o= data_a_i⊕data_b_i
110	Absolute Value	data_o=| data_b_i-data_a_i |

Table1. Function Operator

*for state 001, you need to consider sign extension

Instruction	Operation	Allowed Next State
000	Unsigned Addition	Don't care
001	Unsigned Subtraction	Don't care
010	Unsigned Multiplication	000,001,011,101,111
011	data_a_i Divide by 2	000,001,110,111
100	Subtraction & NOT	001,101,111
101	XOR	000,001,010,111
110	Absolute Value	001,101
111	Maintain the Previous Operation	Same with previous operation

Table2. Function Priority Condition

Number	Input Instruction	Current State	Next State	Operation
1	000	000	000	Unsigned Addition
2	101	000	101	XOR
3	110	101	101	XOR

4	111	101	101	XOR
5	010	101	010	Unsigned Multiplication

Table3. Operation Example

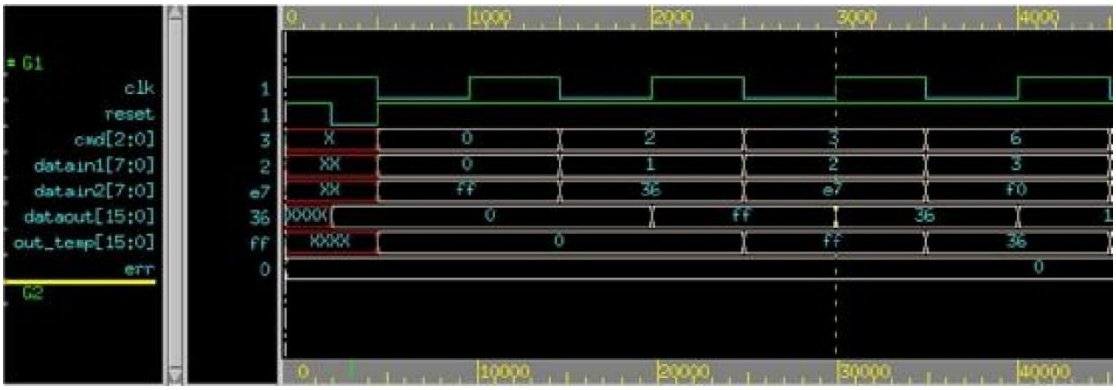

Figure2. Waveform Example

IV. Attention

1. Plagiarism is not allowed.
2. An unfinished Verilog RTL template program is provided (HW1_alu.v). Please complete the program to achieve the above functions.
3. Please use the provided testbench to test the basic functions of your RTL code. We'll score your ALU using this testbench.
4. In your design, please follow the port name and port order as used in the given Verilog template.

V. Grading

1. HW1_alu.v design using Verilog (80%)

2. Report(StudentID_HW1.pdf)

2.1 FSM diagram (15%) : Plot the state diagram of your design.

2.2 Discussion (5%) : Discuss about your design. For example, introduce your design, how do you do the FSM.

VI. Notification

1. You should compress your design files into a single ZIP file, and then upload your ZIP file to NTU COOL before 02/24 23:59.

1.1 Files should be compressed as follows:
ICDLAB_HW1_ StudentID.zip
- ICDLAB_HW1_StudentID
  - HW1_alu.v
  - StudentID_HW1.pdf

Examples:
ICDLAB_HW1_r10943003.zip
- ICDLAB_HW1_r10943003
  - HW1_alu.v
  - R10943003_HW1.pdf

1.2 If you want to modify your code, please submit the new file with the name ICDLAB_HW1_ StudentID_v2.zip