Project #2:
ALU & Register File

Dongmoon Min
(Sungkyunkwan University)
dongmoon.min@skku.edu

Projects of This Semester

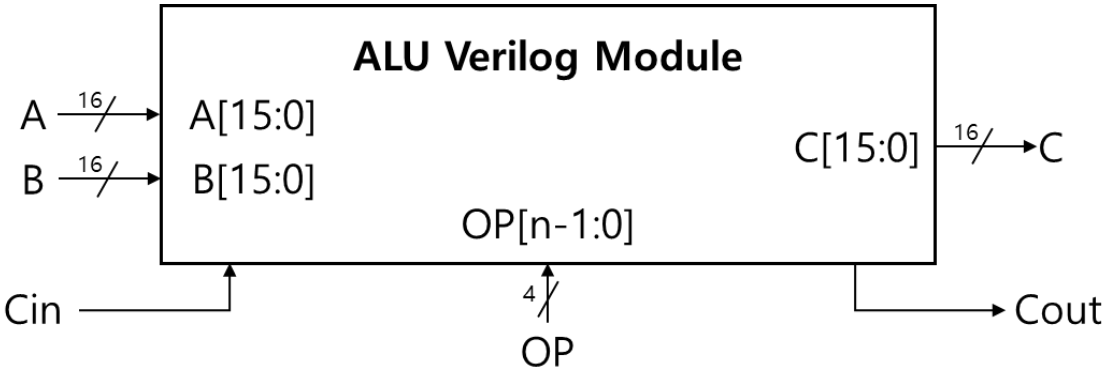
	What we will make
~~Project #1~~	~~010 Detector~~
Project #2	ALU & Register File
Project #3	Single-Cycle CPU

- Language: Verilog
- Framework: AMD Vivado, Icarus Verilog

Arithmetic & Logic Unit

Arithmetic & Logic Unit (ALU)

- Goal: Implement a 16-bit arithmetic & logic unit
  - Input: 16-bit A, 16-bit B, 1-bit Cin, 4-bit OP
  - Output: 16-bit C, 1-bit Cout



Category	Operation	Description	Opcode
Arithmetic	Add	C <- A + B (You must take care of Cin & Cout.)	OP_ADD
	Subtract	C <- A – B (You must take care of Cin & Cout.)	OP_SUB
Bitwise Boolean Operation	Identity	C <- A	OP_ID
	NAND	C <- A nand B	OP_NAND
	NOR	C <- A nor B	OP_NOR
	XNOR	C <- A xnor B	OP_XNOR
	NOT	C <- ~A	OP_NOT
	AND	C <- A and B	OP_AND
	OR	C <- A or B	OP_OR
	XOR	C <- A xor B	OP_XOR
Shifting	Logical right shift	C <- A >> 1	OP_LRS
	Arithmetic right shift	C <- A >>> 1	OP_ARS
	Rotate right	rotate each bit of A right by 1 bit LSB becomes MSB	OP_RR
	Logical left shift	C <- A << 1	OP_LLS
	Arithmetic left shift	C <- A <<< 1	OP_ALS
	Rotate left	rotate each bit of A left by 1 bit MSB becomes LSB	OP_RL

Cin & Cout

- Cin and Cout are only used for arithmetic operations.
- Cout = 0 for other operations
  - e.g., Bitwise Boolean Operation, Shifting

- Add: C = A + B + Cin
  - Cout = 1 when overflow occurred (i.e. C[16] = 1)
- Subtract: C = A – (B + Cin)
  - Cout = 1 when underflow at the largest bit occurred (i.e. A < B + Cin)

Important Announcement

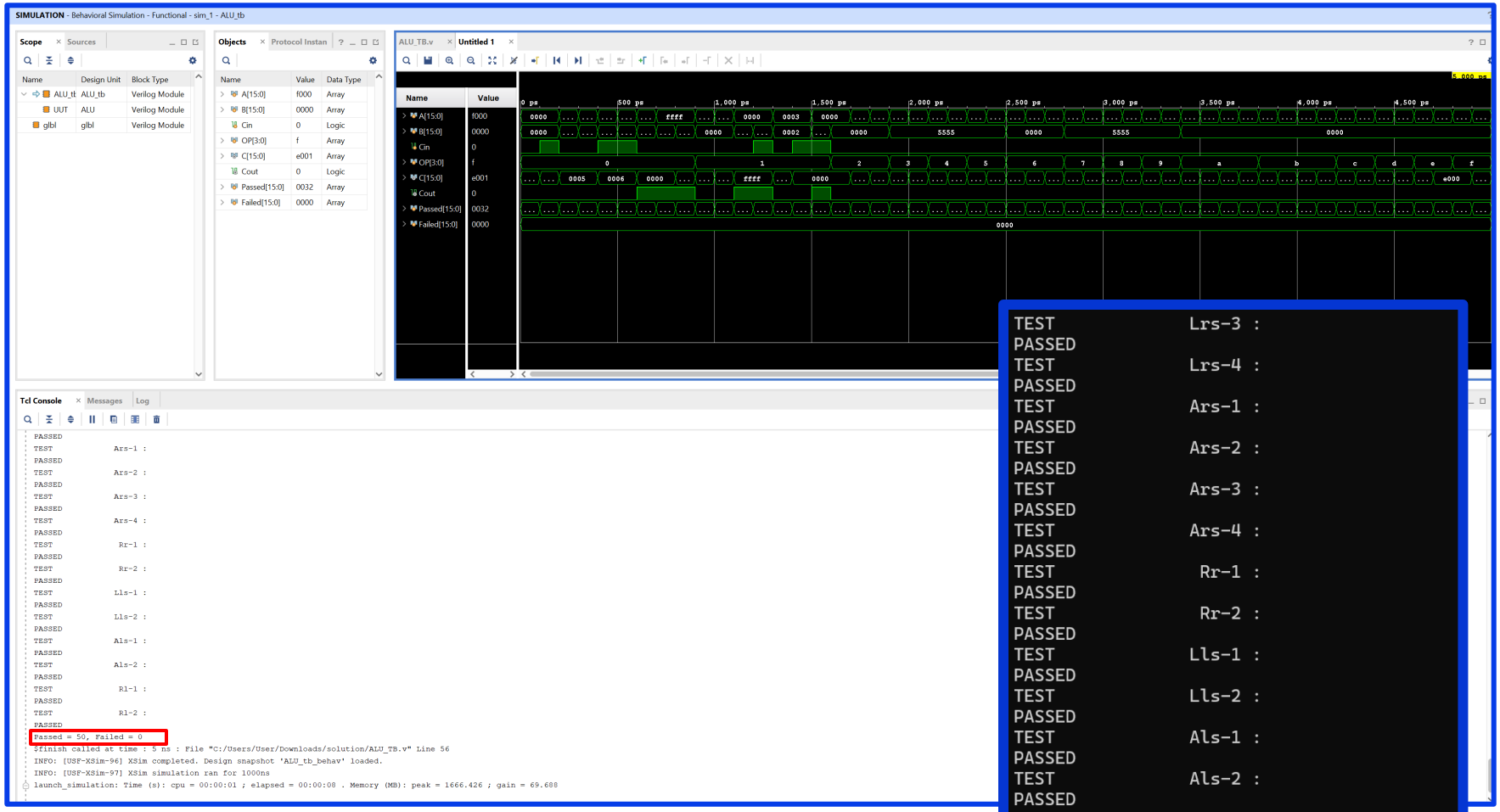
- You can change "FILLME in ALU.v" file only.
  - If you change others (i.e., module name, input, output, ALU_TB.v), we will give you zero score for your project!

- You should use OP codes defined in ALU_TB.v for implementing your ALU in ALU.v.

```
1   `timescale 100ps / 100ps
2
3   // Arithmetic
4   `define OP_ADD  4'b0000
5   `define OP_SUB  4'b0001
6   //  Bitwise Boolean operation
7   `define OP_ID   4'b0010
8   `define OP_NAND 4'b0011
9   `define OP_NOR  4'b0100
10  `define OP_XNOR 4'b0101
11  `define OP_NOT  4'b0110
12  `define OP_AND  4'b0111
13  `define OP_OR   4'b1000
14  `define OP_XOR  4'b1001
```

```
15  // Shifting
16  `define OP_LRS  4'b1010
17  `define OP_ARS  4'b1011
18  `define OP_RR   4'b1100
19  `define OP_LLS  4'b1101
20  `define OP_ALS  4'b1110
21  `define OP_RL   4'b1111
22
```

Expected Result



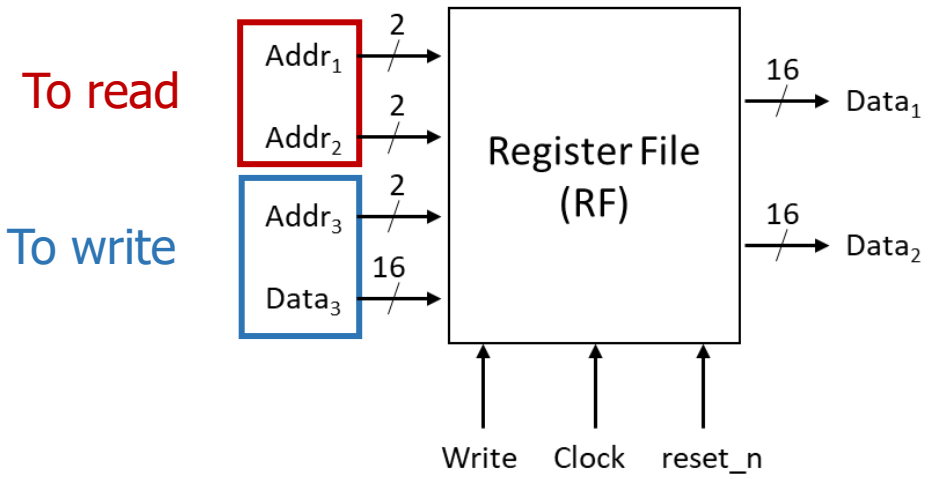

```
TEST            Lrs-3 :
PASSED
TEST            Lrs-4 :
PASSED
TEST            Ars-1 :
PASSED
TEST            Ars-2 :
PASSED
TEST            Ars-3 :
PASSED
TEST            Ars-4 :
PASSED
TEST             Rr-1 :
PASSED
TEST             Rr-2 :
PASSED
TEST            Lls-1 :
PASSED
TEST            Lls-2 :
PASSED
TEST            Als-1 :
PASSED
TEST            Als-2 :
PASSED
TEST             Rl-1 :
PASSED
TEST             Rl-2 :
PASSED
Passed = 50, Failed = 0
ALU_TB.v:55: $finish called at 5000 (1ps)
```

You should pass all 50 cases!

Register File

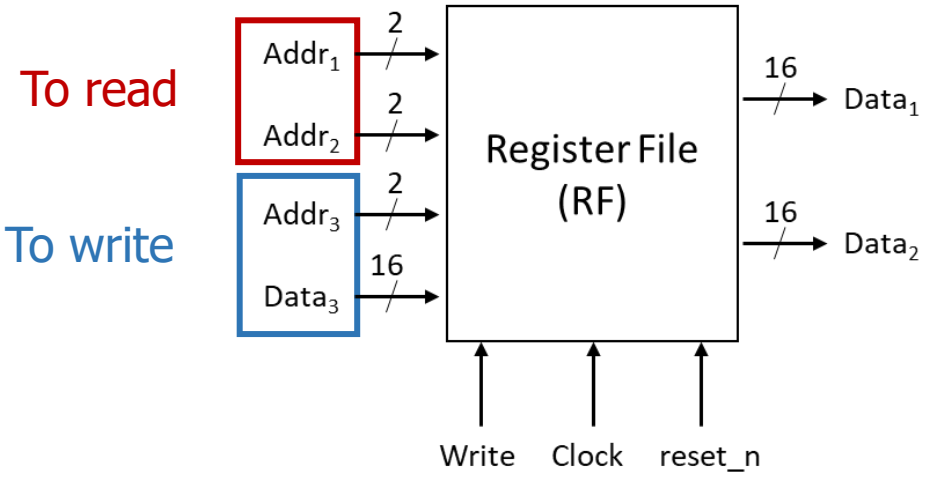
Register File

- Goal: Implement a 16-bit 2-read / 1-write register file
  - Input: write (1-bit), clock (1-bit), reset (1-bit), three addresses (each 2-bit), one data (16-bit)
  - Output: two read data (each 16 bits)



Register File

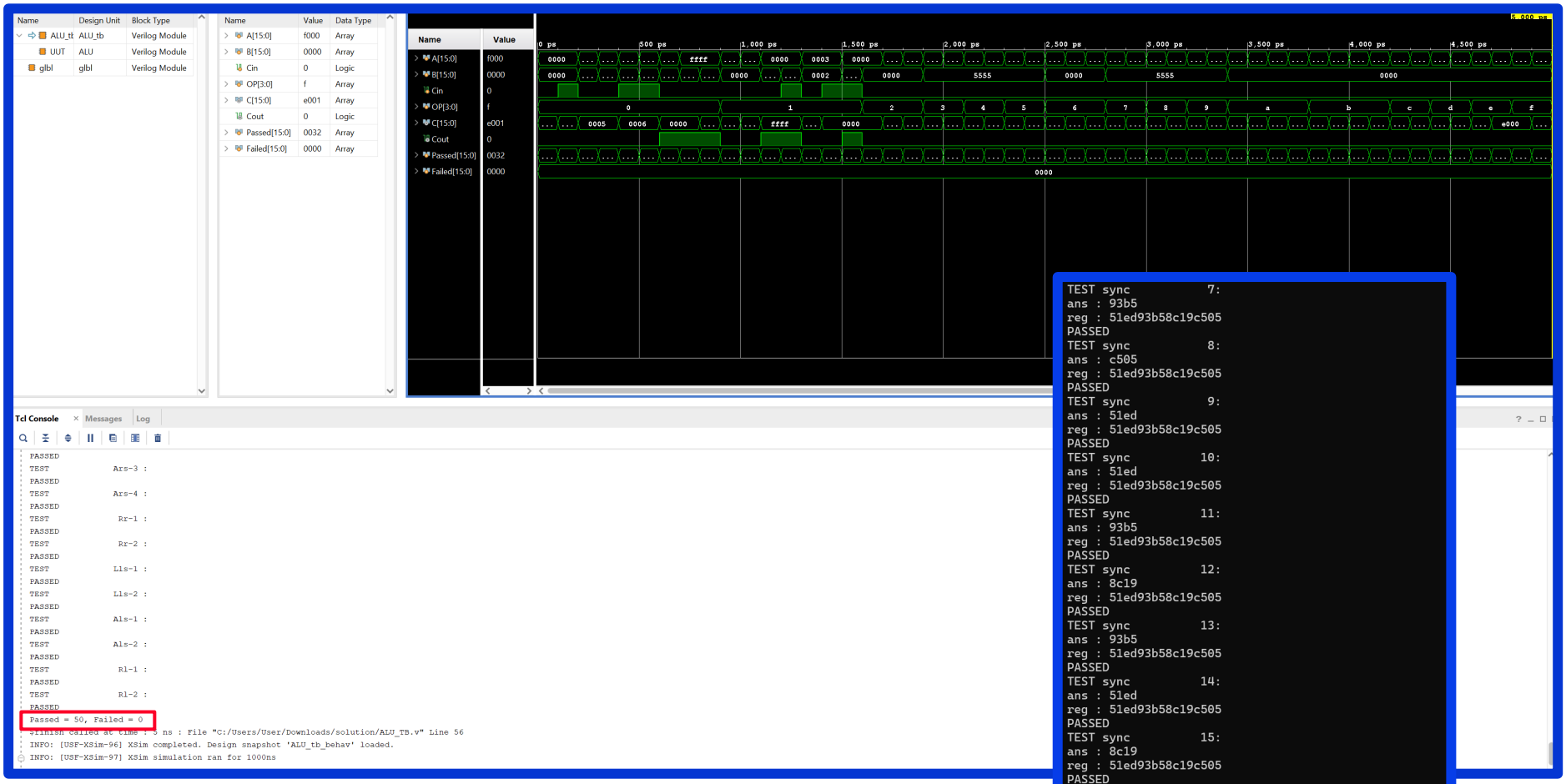
- Operation
  - In rising edge of clock,
    - when reset is high, set all the register values to 0.
    - when write is high, write $data_3$ to the register of $addr_3$.
  - Read $data_1$ and $data_2$ by using $addr_1$ and $addr_2$, respectively, regardless of write & clock



Important Announcement

- You can change "FILLME in RF.v" file only.
  - If you change others (i.e., module name, input, output, RF_TB.v), we will give you zero score for your project!

Expected Result

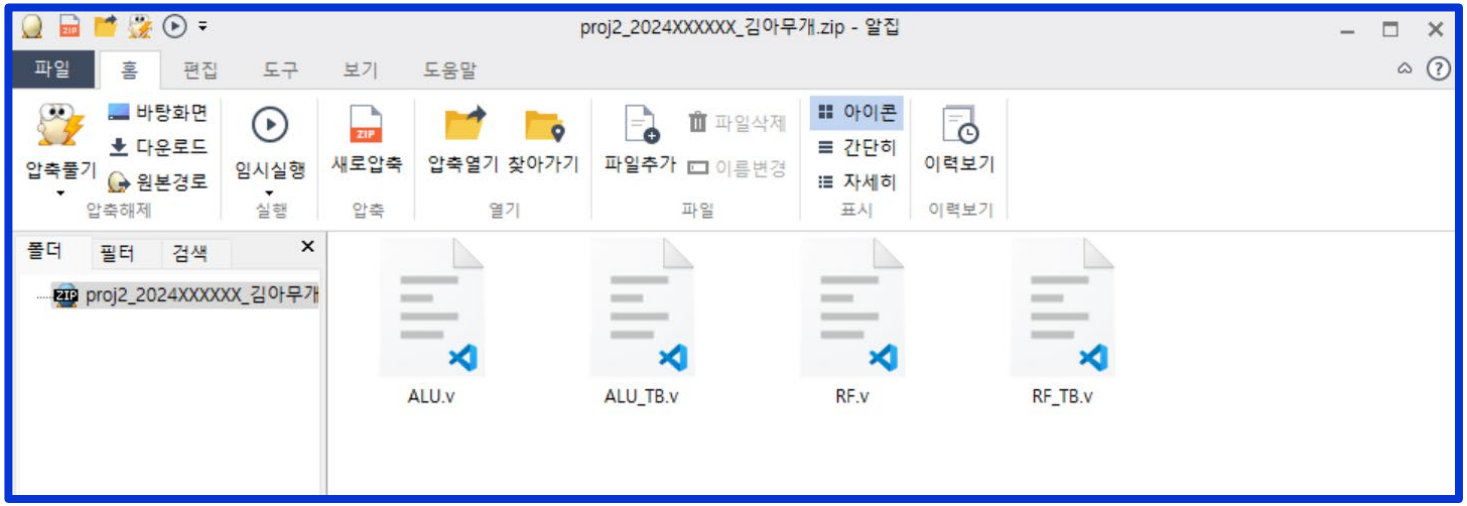


You should pass all 50 cases!

Submission

- Deadline: ~ 5/11 23:59

- Submit your **Verilog codes only** (i.e., ALU.v, ALU_TB.v, RF.v, RF_TB.v) by compressing them to zip file (*.zip).
  - Name of file: proj2_2024XXXXXX_김아무개.zip



Again!!!

- You should change only “FILLME of ALU.v, RF.v files”.
  - If you change others (i.e., module name, input, output, testbench), we will give you zero score for your project!