University of Jordan

Department of Computer Engineering

Digital Logic Laboratory

Lab Project

Due Date (10/5/2021 @ 11:59 PM)

The purpose of this project is to design a 4-bit Arithmetic Logic Unit (ALU) with registered inputs and outputs. The ALU should perform the following logical operations: AND, OR, XOR, XNOR, NOT, Rotate Left (RL), Rotate Right (RR), Shift Left (SL), and Shift Right (SR). In addition, the ALU should perform the following unsigned arithmetic operations: addition, subtraction, and multiplication.

The block diagram of the ALU is shown in Figure 1. Notice that the ALU inputs and outputs are registered using D-type flip-flops. Moreover, 7-segment displays and light-emitting diodes (LEDs) are used to display the registered inputs and outputs.

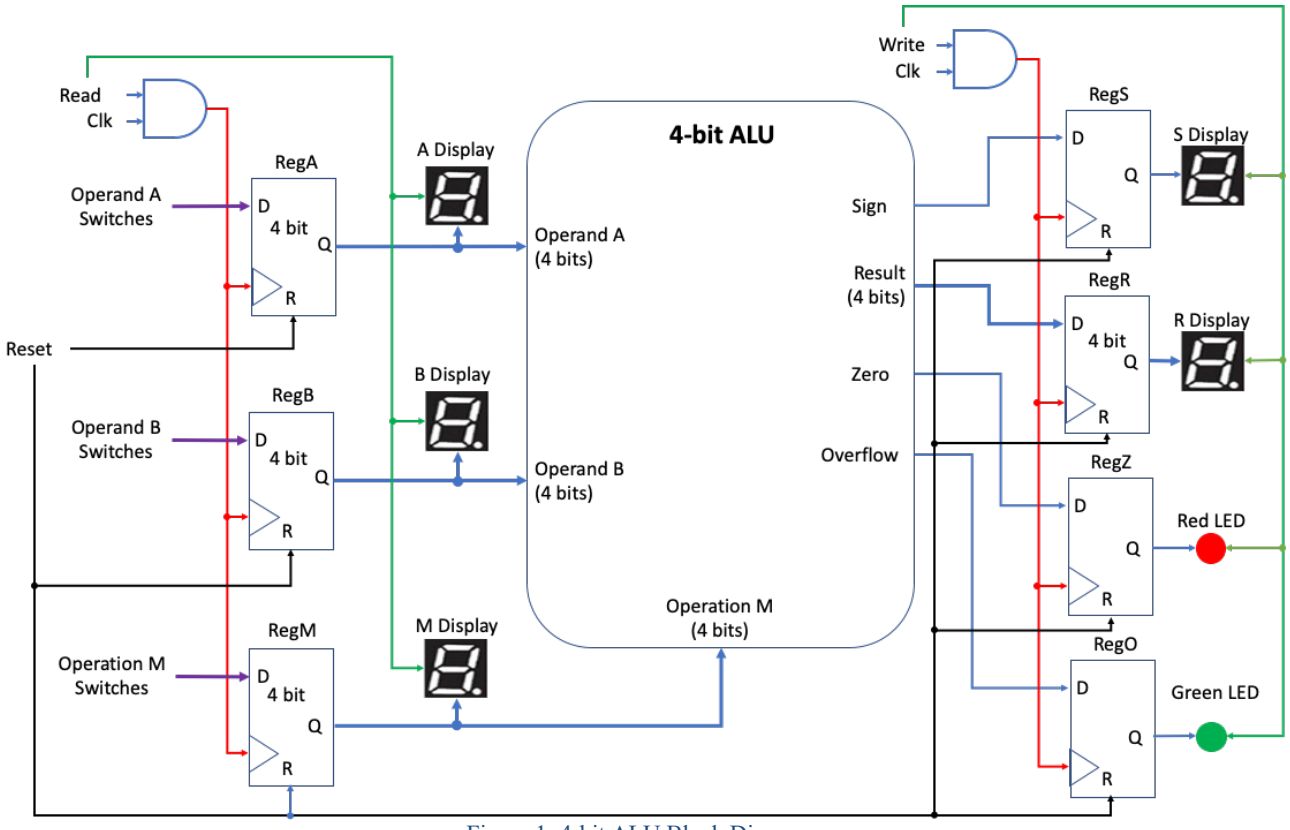

Figure 1: 4-bit ALU Block Diagram

- **ALU Inputs**

The ALU has two 4-bit data inputs (i.e. Operand A ($A_3A_2A_1A_0$) and operand B ($B_3B_2B_1B_0$)) and one 4-bit control input (i.e. M ($M_3M_2M_1M_0$)). The arithmetic/logical operation on A and B is determined according to the value of M as shown in Table 1.

Table 1: ALU Operations

M_3	M_2	M_1	M_0	Operation
0	0	0	0	A AND B
0	0	0	1	A OR B
0	0	1	0	A XOR B
0	0	1	1	A XNOR B
0	1	0	0	RL A by one bit
0	1	0	1	RR A by one bit
0	1	1	0	SL A by one bit
0	1	1	1	SR A by one bit
1	0	0	0	RL B by one bit
1	0	0	1	RR B by one bit
1	0	1	0	SL B by one bit
1	0	1	1	SR B by one bit
1	1	0	0	Unsigned Subtract (A – B)
1	1	0	1	Unsigned Add
1	1	1	0	Unsigned Multiply
1	1	1	1	NOT A

Each one of the inputs is registered using a 4-bit register. For example, operand A is registered using the 4-bit register: RegA. Similarly, RegB and RegM are used to register operand B and operation M respectively. The input registers operation is controlled by the Reset, Clk, and Read signals. The Reset signal is asynchronous active high signal; hence, when the Reset signal is active all input registered are set to 0000. When Reset is not active, the input registers values are updated at the positive edge of the Clk signal provided that Read signal is set to 1.

- **ALU Outputs**

After the ALU module finishes execution, the 4-bit result will be registered displayed in RegR at the positive edge of the Clk signal provided that the Write signal is set to 1. In addition, the ALU module generates three 1-bit flags: **Sign, Zero, and Overflow**. The flags will also be registered at the same time with the result in RegS, RegZ, and RegO respectively. Table *2* describes the functionality of the flags. **Notice that all output registers are set to Zero when Reset is active.**

Table 2: Zero and Overflow Flags Functionality

Sign	**=1 The sign of the result is negative (only applicable with subtraction operation)**
	=0 The sign of the result is positive or irrelevant
Zero	**=1 The result of the performed operation is 0**
	=0 The result of the performed operation is not Zero
Overflow	**=1 The performed operation produces an overflow (i.e. only addition or multiplication). This happens when the correct answer requires more than the 4-bit result available.**
	=0 The performed operation does not produce an overflow

- **Inputs and Outputs Display**

The **registered** inputs (i.e. RegA, RegB, and RegM) are displayed on three 7-segment displays as shown in Figure 1 (i.e. A Display, B Display, and M Display). Similarly, two 7-segment displays (i.e. R Display and S Display) are used to display the **registered** result of the ALU (i.e. RegR) with its corresponding sign when applicable. On the other hand, the **registered** Zero and Overflow flags are displayed using a red led and a green led respectively. When the flag is 1 **and** the Write signal is 1, the led turns ON, otherwise it turns OFF. Table 3 show the details of how the input data and output result are displayed on the 7-segment displays.

Table 3: Inputs and Outputs Display Details

7-segment Display	Displayed Value
A Display	The value of RegA in Hexadecimal format (i.e. 0 to F). The Display is ON only when the Read signal is set; otherwise it is OFF.
B Display	The value of RegB in Hexadecimal format. The Display is ON only when the Read signal is set; otherwise it is OFF.
M Display	The value of RegM in Hexadecimal format. The Display is ON only when the Read signal is set; otherwise it is OFF.
S Display	' – ' when operation is unsigned subtraction **and** (RegA < RegB) **and** the Write signal is set; otherwise it is OFF.
R Display	The value of RegR in Hexadecimal format. The Display is ON only when the Write signal is set; otherwise it is OFF.

- **Important Notes:**

1. When the selected operation is unsigned subtraction (i.e. RegM = 1100), RegR should have the value of (RegB – RegA) if RegA < RegB and the value of (RegA – RegB) when RegA ≥ RegB. For example, if RegA = 0101 and RegB = 0111, then RegR should be 0010.
2. When the selected operation is unsigned multiplication (i.e. RegM = 1110), RegR should have the least significant 4-bit of the answer. For example, if RegA = 0101 and RegB = 0111, then RegR should be 0011 because the full answer is "0010 0011".

- **Submission Guidelines:**

1. Students should implement the project using Verilog HDL. Schematic designs are not allowed.
2. Students are allowed to work in teams of two or three and students from different sections CANNOT work in the same team. It is preferable that the team members are from different departments (i.e. computer, electrical, and mechatronics).
3. Each team should be able to show a demo of the working project to their lab engineer/instructor and answer all their questions during specific time that will be announced. Students are expected to answer all the questions about all parts of the project

during the discussion.

4. Each team should submit a written report that describes the functionality, inputs and outputs of each defined module. In addition, you need to describe how the modules are connected together to form the top-level design. Moreover, the report should include snapshots of multiple simulation waveforms that demonstrate the correct functionality of the system in each of the following scenarios:

    a. When Read = 0: A Display, B Display, and M Display are turned OFF.

    b. When Write = 0: S Display, R Display, Zero LED, and Overflow LED are turned OFF.

    c. When Reset is active with Read = 1and Write = 1, all registers are set to 0. Hence, all 7-segment displays show the value 0 and all LEDs are OFF.

    d. For each of the 16 operations, show multiple cases (at least four cases) with correct input and output displays and LEDs. For addition and multiplication, the simulation should include some cases with Overflow LED ON.

5. Each team should choose a leader, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives. The roles, goals, and tasks should be documented in the report and submitted with the project files.

6. Please make sure to follow these rules when you submit your project:

    a. Place all of your project files (.v, .vwf, and report) in one folder.

    b. Compress the submission folder into a .zip or .rar format.

    c. The name of the compressed file should be Team_[TeamNumber].rar OR Team_[TeamNumber].zip. For example, if your team number is 15 then your submitted file should be named as Team_15.rar OR Team_15.zip.

    d. Each team should submit the compressed file through Teams before the deadline.

    e. If you want to update your submission, you need to delete the old submission before turning in the new one and this has to be done before the deadline.

    f. No late submission will be accepted under any circumstances.

7. Cheating will not be tolerated and will result in zero grade.