Cheating Policy: Each student involved in cheating will be given -100.
Grading Policy: The resultant design must execute on FPGA to get a grade greater than 30. Each operation brings points if it executes accurately:

- Accurate ADD (20pts)
- Accurate SUB (20pts)
- Accurate AND (15pts)
- Accurate XOR (15pts)
- Accurate Design (30pts)
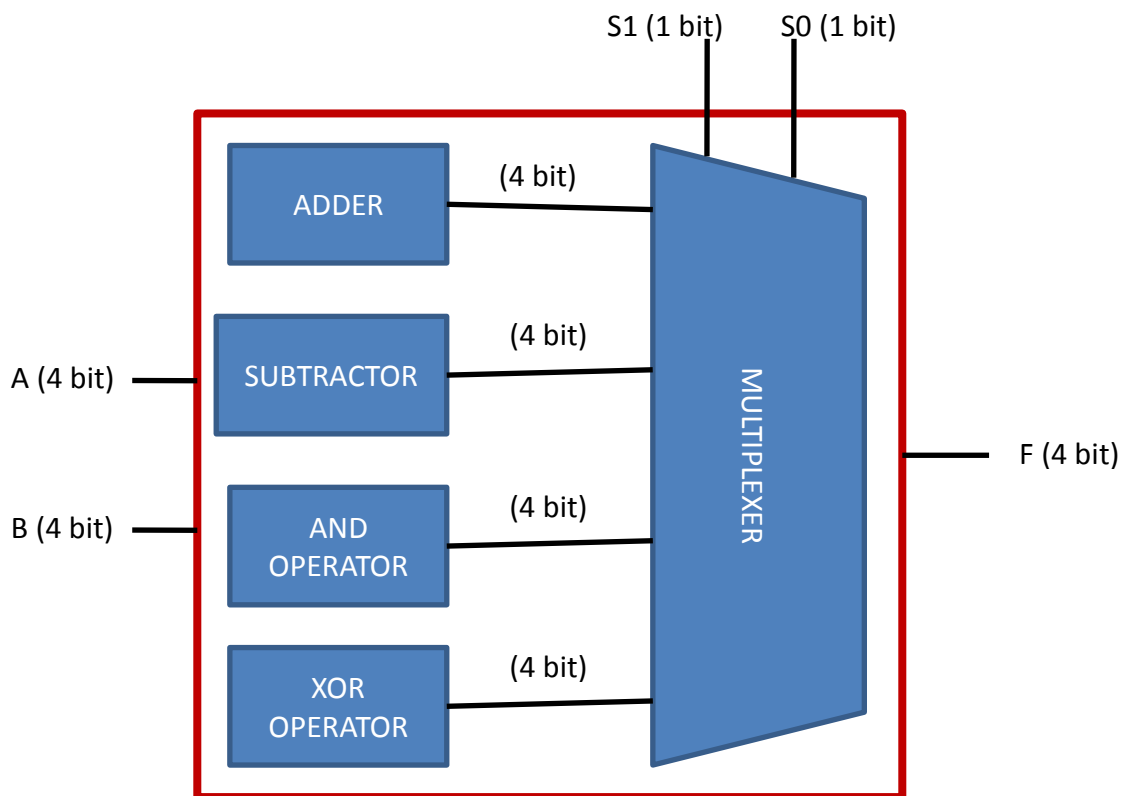
In this project, you are going to implement a simple ALU using Verilog. ALU is capable of performing 4 basic operations as described below and depicted above:

1. Adder: It takes the values A and B (4 bits) as input. Then, it sums up A and B to find the result F (4 bits). Note that F can be 5 bits as well. Just ignore the fifth bit.
2. Subtractor: It takes the values A and B (4 bits) as input. Then, it subtracts B from A to find the result F (4 bits).
3. AND Operation: It takes the values A and B (4 bits) as input. Then, it performs AND operation using A and B to find the result F (4 bits).
4. XOR Operation: It takes the values A and B (4 bits) as input. Then, it performs XOR operation using A and B to find the result F (4 bits).

You can only use structural Verilog to implement this project. You can use logic gates AND, OR, NOT, XOR ONLY. Behavioral Verilog is not allowed.

You must implement a multiplexer which determines which operation ALU will do. Multiplexer takes S (2 bits – S1 and S0).

- If S is 0 (00), adder result will be seen on LEDs.
- If S is 1 (01), subtraction result will be seen on LEDs.
- If S is 2 (10), AND operation result will be seen on LEDs.
- If S is 3 (11), XOR operation result will be seen on LEDs.

Values A and B taken from user using toggle switch on the DE0 board. SW0-SW3 are going to be used to take the value of A. Similarly, SW4-SW7 are going to be used to get value of B. The last two switches SW8 and SW9 are going to be used to determine which operation ALU will do. The result is going to be shown on LEDs.

[BONUS – 30 Points] The result value (F) is going to be displayed on Seven Segment Display instead of LEDs. The result will be hexadecimal instead of binary.

Sample program for controlling LEDs with switches on DE0 board:

```
module toggleswitch_led_tutorial (led[9:0], sw[9:0]);
        input [9:0] sw;
        output [9:0] led;

        assign led = sw;        // each toggle switch is assigned to corresponding led.
                                // i.e. led[0] = sw[0]
endmodule
```
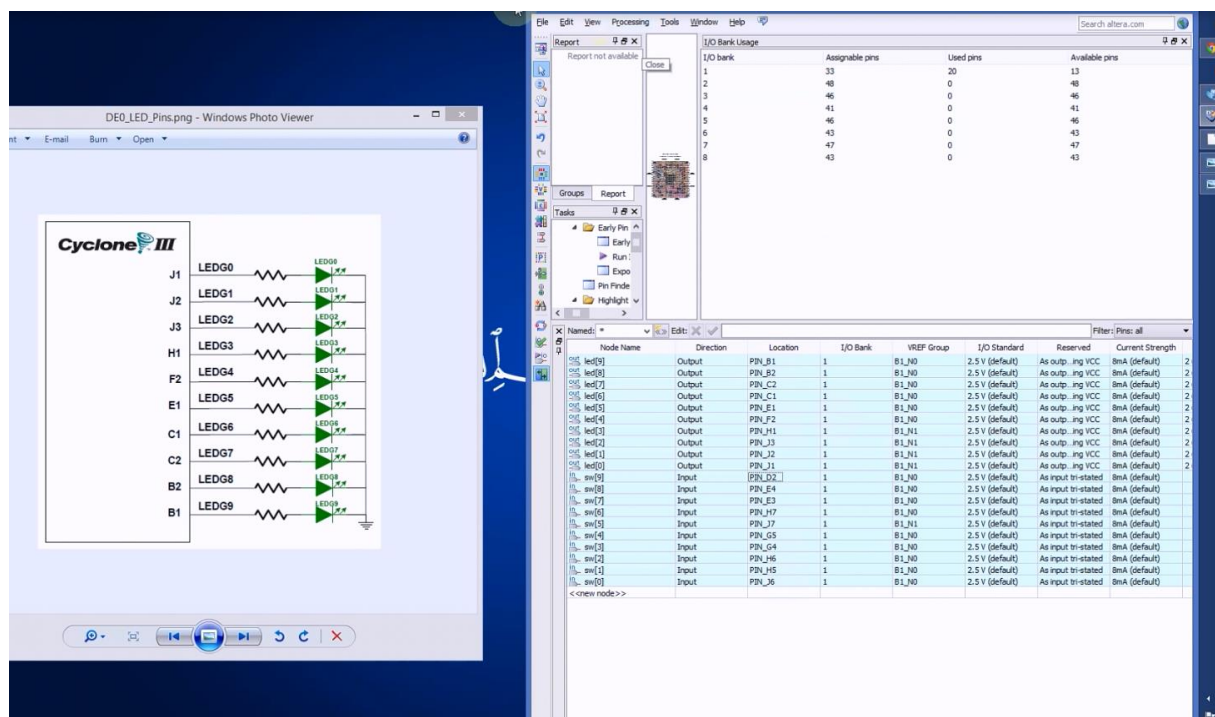

Screenshot from the tutorial videos, which shows the pin planning.

Watch the video tutorials of your TA uploaded to Moodle! Check the DE0 Manual on Moodle.

Come to the PS next Tuesday (October 6). TA will explain the project in detail, teach structural Verilog and you can also ask anything about the project during this PS.

The grading will be done by demo performed by the students. The demo hours will be announced on Moodle.

No late submissions even 1 minute.