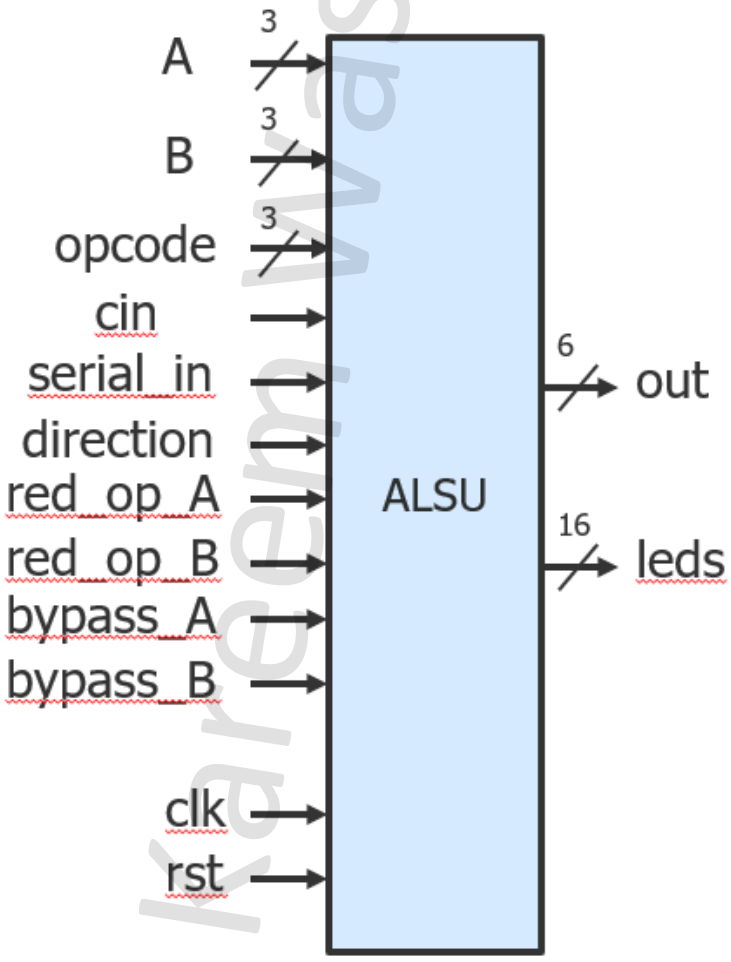
Assignment 4

Design the following circuits using Verilog **and create a testbench** for each design to check its functionality. **Create a do file for the question.**

1) ALSU is a logic unit that can perform logical, arithmetic, and shift operations on input ports

- Input ports A and B have various operations that can take place depending on the value of the opcode.
- Each input bit except for the clk and rst will be sampled at the rising edge before any processing so a D-FF is expected for each input bit at the design entry.
- The output of the ALSU is registered and is available at the rising edge of the clock.



Inputs

Each input bit except for the clk and rst will have a DFF in front of its port. Any processing will take place from the DFF output.

Input	Width	Description
clk	1	Input clock
rst	1	Active high asynchronous reset
A	3	Input port A
B	3	Input port B
cin	1	Carry in bit, only valid to be used if the parameter FULL_ADDER is "ON"
serial_in	1	Serial in bit, used in shift operations only
red_op_A	1	When set to high, this indicates that reduction operation would be executed on A rather than bitwise operations on A and B when the opcode indicates AND and XOR operations
red_op_B	1	When set to high, this indicates that reduction operation would be executed on B rather than bitwise operations on A and B when the opcode indicates AND and XOR operations
opcode	3	Opcode has a separate table to describe the different operations executed
bypass_A	1	When set to high, this indicates that port A will be registered to the output ignoring the opcode operation
bypass_B	1	When set to high, this indicates that port B will be registered to the output ignoring the opcode operation
direction	1	The direction of the shift or rotation operation is left when this input is set to high; otherwise, it is right.

Outputs and parameters

Output	Width	Description
leds	16	When an invalid operation occurs, all bits blink (bits turn on and then off with each clock cycle). Blinking serves as a warning; otherwise, if a valid operation occurs, it is set to low.
out	6	Output of the ALSU

Parameter	Default value	Description
INPUT_PRIORITY	A	Priority is given to the port set by this parameter whenever there is a conflict. Conflicts can occur in two scenarios, red_op_A and red_op_B are both set to high or bypass_A and bypass_B are both set to high. Legal values for this parameter are A and B
FULL_ADDER	ON	When this parameter has value "ON" then cin input must be considered in the addition operation between A and B. Legal values for this parameter are ON and OFF

Opcodes & Handling invalid cases

Opcode	Operation
000	AND
001	XOR
010	Addition
011	Multiplication
100	Shift output by 1 bit
101	Rotate output by 1 bit
110	Invalid opcode
111	Invalid opcode

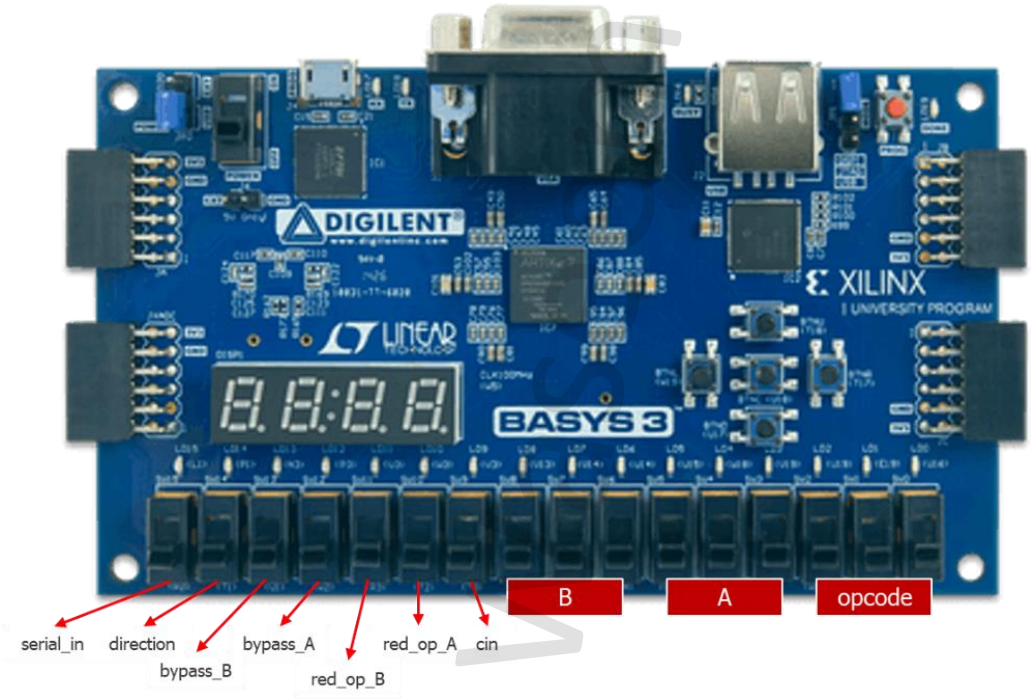
<u>Invalid cases</u>

1. Opcode bits are set to 110 or 111
2. red_op_A or red_op_B are set to high and the opcode is not AND or XOR operation

<u>Output when invalid cases occurs</u>

1. leds are blinking
2. out bits are set to low, but if the bypass_A or bypass_B are high then the output will take the value of A or B.

You are required to write the constraint file for the ALSU done in assignment 5. Connect the inputs A, B and opcode to the switches as shown on the board below



- "clk" is connected to W5 pin as suggested in the board's reference manual with frequency 100 MHz
- "rst" is connected to button U18
- "leds" are connected to the LEDs on the board

You are required to set a debug core to be able to debug any input or output for the design.

Note:

- You can use in the testbench the system function "$urandom_range" which returns randomized unsigned integer within a range if you want to strict the opcode to have valid opcodes only
- signal_in = $urandom_range(5,15); //randomized between 5 and 15
- Bit stream generation will not be successful since the output bits are not connected yet (You will connect them to the seven segment display in assignment 5)

Deliverables:

1) The assignment should be submitted as a PDF file with this format <your_name>_Assignment4 for example Kareem_Waseem_Assignment4.
2) Snippets from the waveforms captured from QuestaSim for the design with inputs assigned values and output values visible.
3) Snippets from the schematic after the elaboration & synthesis

4) Snippet from the utilization & timing report & after the synthesis and implementation.
5) Snippet of the “Messages” tab showing no critical warnings or errors after running elaboration, synthesis, and implementation.

Note that your document should be organized as follows:

1. RTL code
2. Testbench code
3. Do file
4. QuestaSim Snippets
5. Constraint File showing the debug core added in the end of the file.
6. Elaboration (“Messages” tab & Schematic snippets)
7. Synthesis (“Messages” tab, Utilization report, timing report & Schematic snippets)
8. Implementation (“Messages” tab, Utilization report, timing report & device snippets)

Note that your document should be organized. I am expecting the Verilog code, and the waveforms snippets