Finite State Machine

Design the following circuits using Verilog **and create a testbench** for each design to check its functionality using QuestaSim. Use a **do file** for question 3.

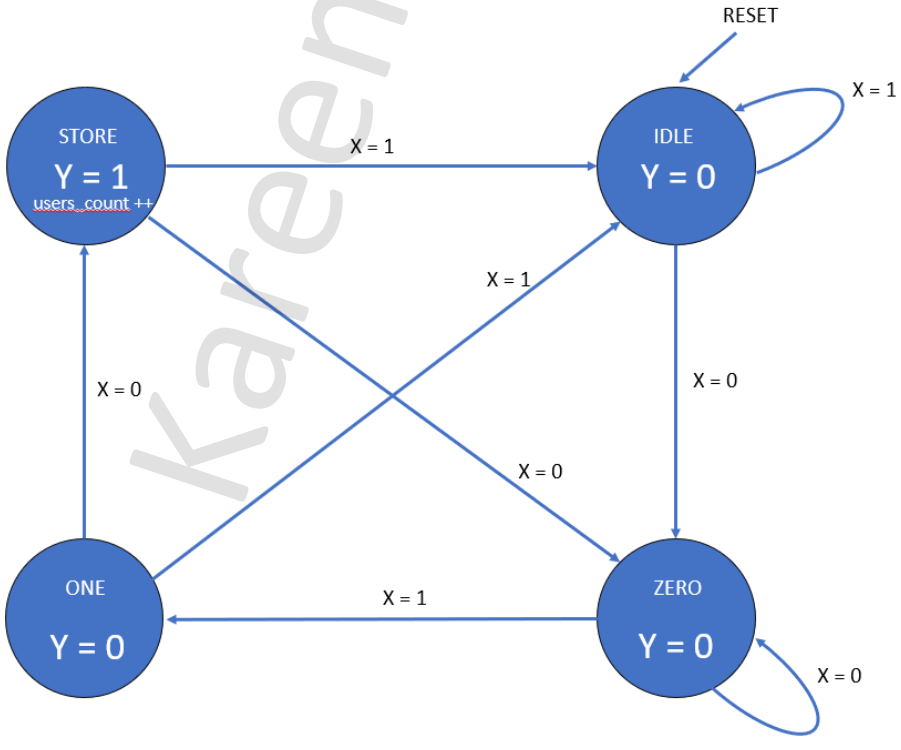
Use **Vivado** to go through the **design flow** running elaboration, synthesis, implementation for all the designs making sure that there are no design check errors during the design flow. Check the deliverables by the end of the document

Requirements:

1- Design Moore FSM that detects "010" non-overlapped pattern. (Draw state transition diagram)
2- Store how many time the pattern was detected.

Ports:

Name	Type	Size	Description
x	Input	1 bit	Input sequence
clk			Clock
rst			Active high asynchronous reset
y	Output	1 bit	Output that is HIGH when the sequence 010 is detected
count		10 bits	Outputs the number of time the pattern was detected



2) Suppose that you are working as a Digital design Engineer in Tesla Company. It is required to design a control unit using Moore FSM for Self-driving cars on highways that controls the acceleration of the car as well as the door locking mechanism with the following specifications.

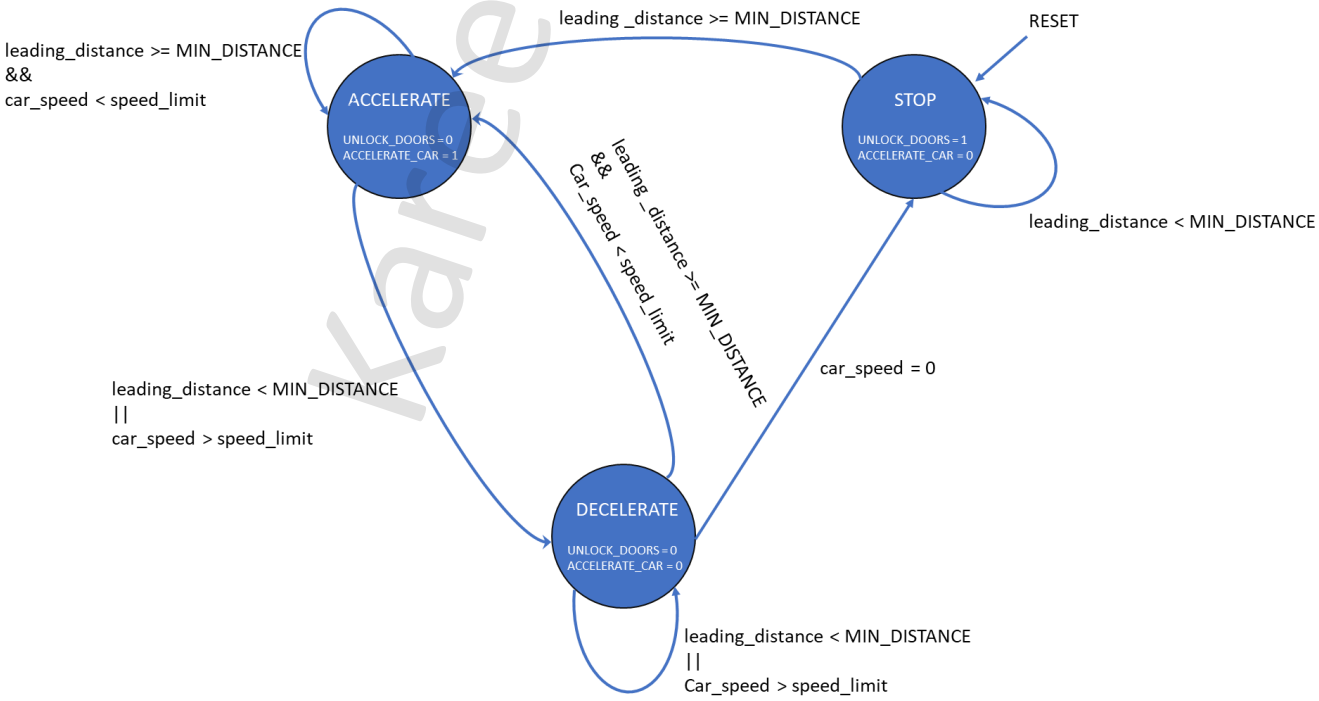
>> Consider creating realistic testbench for this design <<

Ports

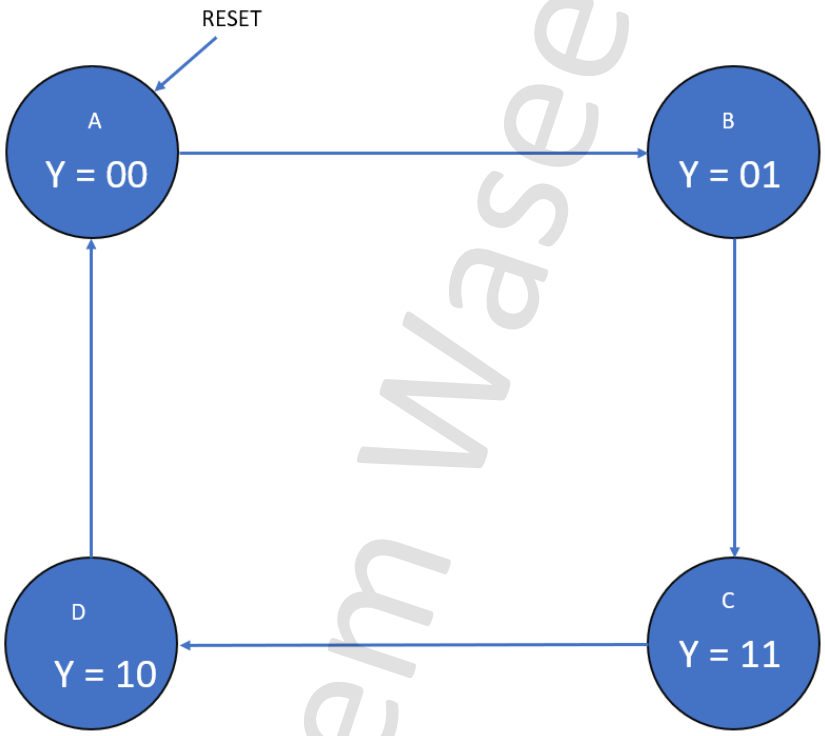
Name	Type	Size	Description
speed_limit	Input	8 bits	Allowable speed limit of the highway
car_speed		8 bits	Current car speed
leading_distance		7 bits	Distance between the car and the vehicle/object in front of it
clk		1 bit	Clock
rst		1 bit	Active high asynchronous reset
unlock_doors	Output	1 bit	Signal that unlock the car doors when HIGH
accelerate_car			Signal that control the flow of the fuel to the engine to accelerate the car when HIGH

Parameters

- MIN_DISTANCE: Minimum distance between two vehicles, default = 7'd40 //40 meters



3) Implement 2-bit gray counter using Moore FSM. Test the following by instantiating the gray counter done in the assignment 4 (extra) and taking it as a reference design. When the reset is deasserted, the states will continue to transition from each state to the neighboring state with each clock cycle.



FSM Encoding:

- A = 2'b00
- B = 2'b01
- C = 2'b10
- D = 2'b11

Inputs:

- clk
- rst (active high async)

Outputs:

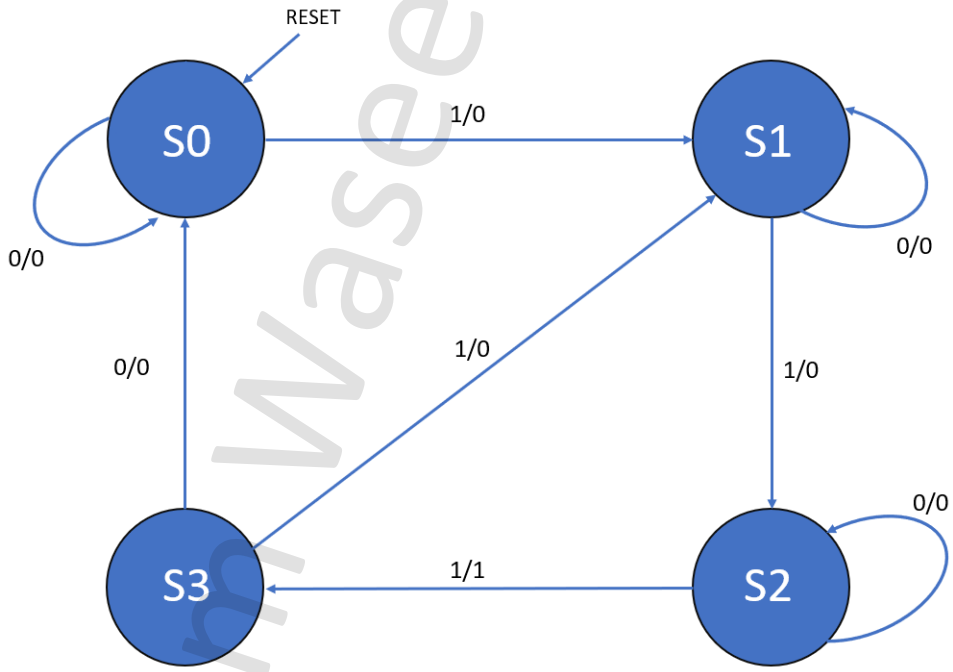
- y, 2-bit output

Create a constraint file and go through the design flow where the reset is connected to a button and the y output is connected to two leds and then generate a bitstream file

4) Create a sequence detector using a Mealy state machine. There is one input (in) and one output (y) in the Mealy state machine. If and only if the total number of 1s received is divisible by 3, the output yout is 1. The rst of the design is active high async.

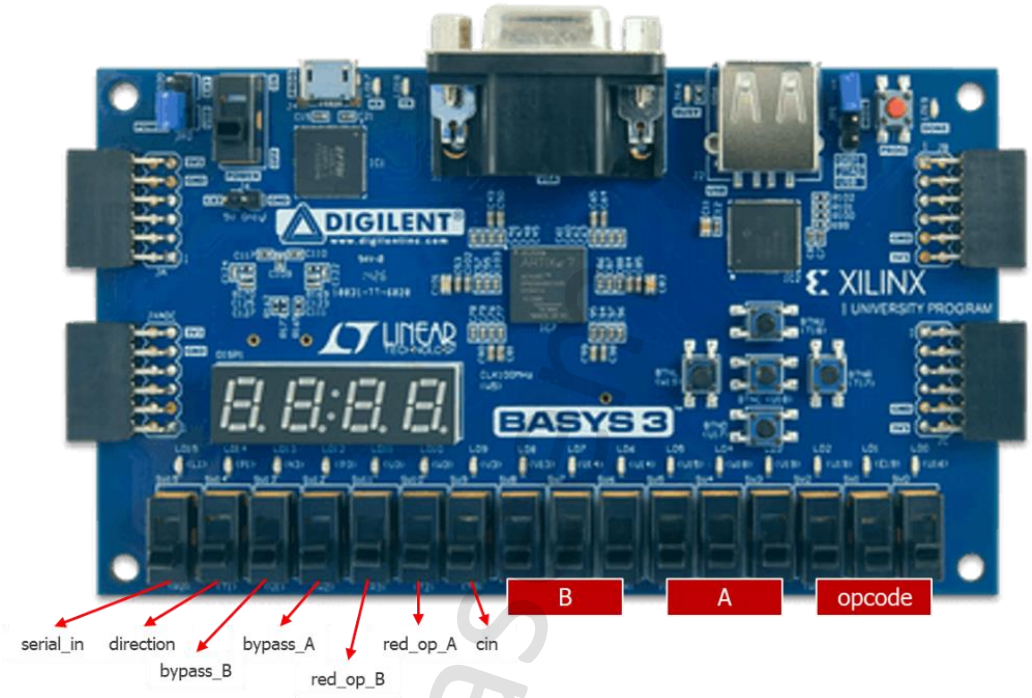
FSM Encoding:

- S0 = 2'b00
- S1 = 2'b01
- S2 = 2'b10
- S3 = 2'b11



Create a constraint file and go through the design flow where the input in is connected to a switch, output y is connected to a led and the reset is connected to a button and then generate a bitstream file

5) 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
- "rst" is connected to button U18
- "leds" are connected to the LEDs on the board

Also, use the IP catalog to generate following 2 IPs:

- multilper to be used when the opcode is 3
- adder to be used when the opcode is 2.
    - Adder IP takes in the A and B and the carry in as inputs. Use a generate if block to instantiate the IP connecting its ports to the ALSU A, B and cin ports when the FULL_ADDER parameter is ON, else connect only A and B and connect the carry in port of the Adder IP to zero

Bonus (Extra):

The output "out" shall be displayed on the seven segment display

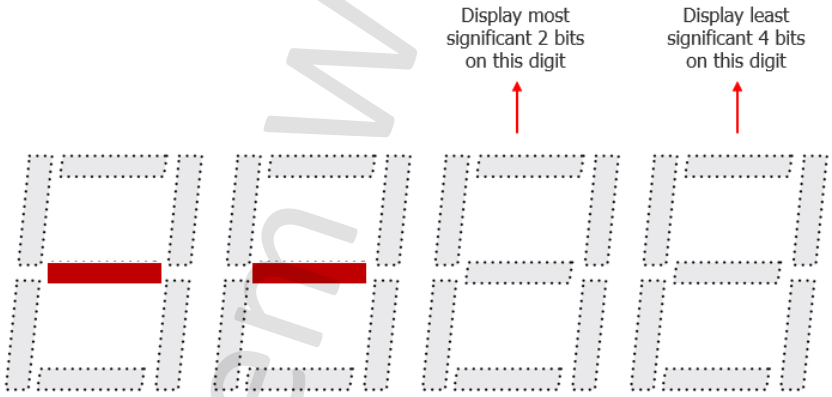
Adjustments to the outputs:

- "out" port will not be output anymore and it will be an internal register
- The output ports are shown below where two outputs are added to be connected to the seven segment display
- You can learn more on how to drive the seven segment display in Basys reference manual and in this link

Output	Width	Description
leds	16	All bits are blinking when an invalid operation takes place. Acts as a warning
anode	4	Output to drive the anodes of the seven segment display
cathode	7	Output to drive the cathodes of the seven segment display

Seven segment display:

1. The least significant 4 bits of the internal register “out” which is the ALSU output will be displayed on the right most digit
2. The most significant 2 bits of the internal register “out” will be displayed on the second digit on the right
3. The 2 digits on the left will be displaying a dash in the middle of them as they are not used
4. If invalid case occurs then “E404” will be displayed on the 4 digits



Deliverables for questions 1 to 4:

1) The assignment should be submitted as a PDF file with this format <your_name>_Assignment6 for example Kareem_Waseem_Assignment6
2) Snippets from the waveforms captured from QuestaSim for each design with inputs assigned values and output values visible.
3) Snippets from the schematic after the elaboration
4) Snippet from the schematic after the synthesis
5) Snippet from the utilization report
6) Snippet from the device after implementation
7) Snippet of the constraint file for questions 3, 4, and 5
8) Snippet for successful bitstream generation for questions 3 and 4

Deliverables for questions 5: Same as the above but you do not have to run Questasim again for simulations, only snippets of the Verilog code after adding the IP catalog as well as add the snippets from points from 3 to 8

Note that your document should be organized as 5 sections corresponding to each design above, and in each section, I am expecting the Verilog code, and the snippets above.