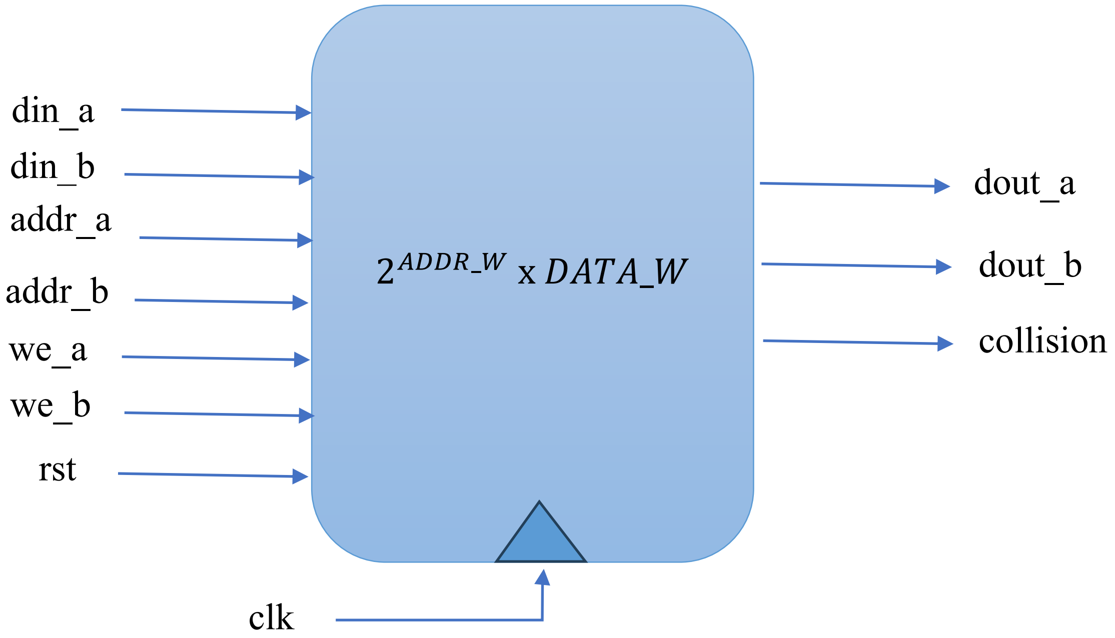
Assignment 6

Q1. Parameterized Dual-Port RAM with Collision Detection



You have to design and implement parameterized **synchronous dual-port RAM** in Verilog with **write collision detection**.

Specifications:

- **Parameters:**
  - DATA_W (default: 16) – Data width
  - ADDR_W (default: 4) – Address width (16 locations)
- **Features:**
  - Two independent ports (A and B) with:
    - Separate address, data, and control signals

- Write collision detection when both ports attempt to write to the same address in the same cycle. Make a priority for port A.

Inputs

Signal	Width	Description
clk	1	Common clock for both ports
addr_a	ADDR_W	Address for Port A
addr_b	ADDR_W	Address for Port B
din_a	DATA_W	Data input for Port A
din_b	DATA_W	Data input for Port B
we_a	1	Write enable for Port A
we_b	1	Write enable for Port B
rst	1	Asynchronous reset

Outputs

Signal	Width	Description
dout_a	DATA_W	Data output from Port A
dout_b	DATA_W	Data output from Port B
collision	1	High if both ports write to same address

Testbench Requirements:

1. Initialize RAM with known values using **$readmemh**.
2. Perform independent reads/writes from both ports.
3. Create scenarios where both ports:
    - Write to different addresses (no collision)
    - Write to the same address (trigger collision)

Q2. FIFO with Almost-Full and Almost-Empty Flags

Specifications:

- **Parameters:**
    - DATA_W (default: 16) – Data width
    - DEPTH (default: 64) – Number of elements in the FIFO
- **Features:**
    - Synchronous read/write operation
    - Status flags:
        - full: FIFO is completely filled
        - almost_full: Only 2 or fewer slots left
        - empty: FIFO is completely empty
        - almost_empty: Only 2 or fewer elements remaining
    - Overflow and underflow detection

Inputs

Signal	**Width**	**Description**
clk	1	System clock
reset_n	1	Active-low synchronous reset
wr_en	1	Write enable
rd_en	1	Read enable
din	DATA_W	Data input

Outputs

Signal	**Width**	**Description**
dout	DATA_W	Data output
full	1	High when FIFO is full
empty	1	High when FIFO is empty
almost_full	1	High when ≤ 2 locations are free
almost_empty	1	High when ≤ 2 elements remain

Q3. Arithmetic Logic Unit

```
module ALU (A,B,opcode,clk,rst,result);
input [1:0] A,B,opcode;
input clk,rst;
output [3:0] result;


always @(posedge clk or posedge rst) begin
   if (!rst) begin
      result <= 0;
   end
   else
      case (opcode)
         0: result <= A + B;
         1: result <= A * B;
         0: result <= A | B;
         3: result <= A ^ B;
      endcase
   end
```

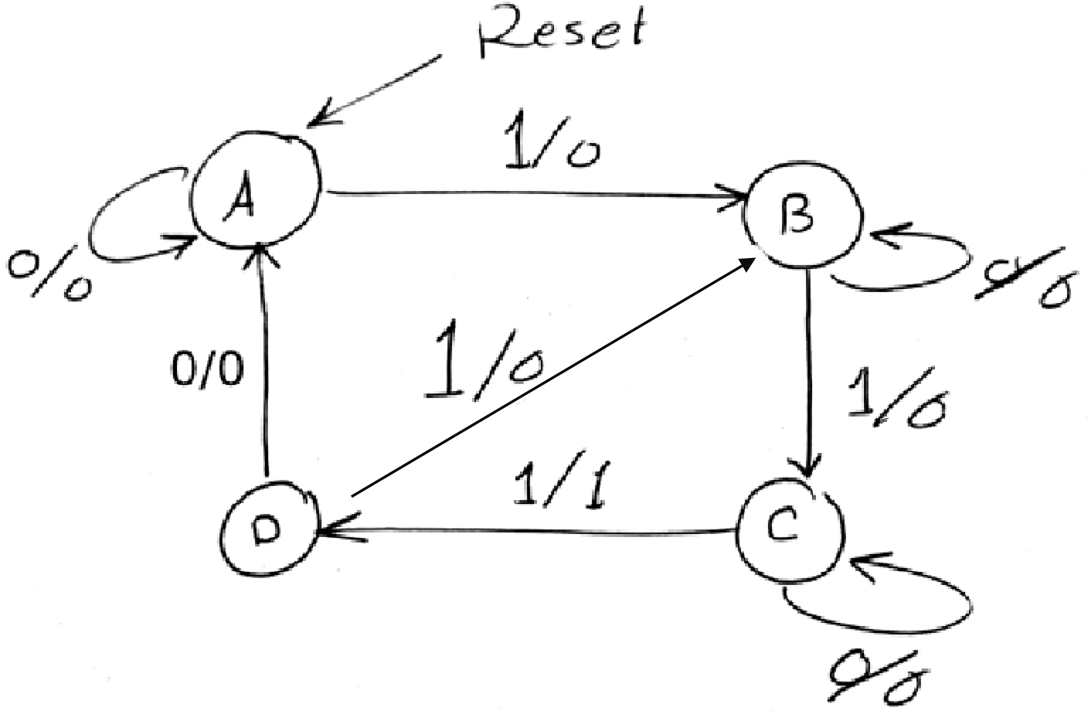
First Part: There are 3 linting errors please correct them and document them.

Second Part: Write an exhaustive test bench and test all input cases.

Third Part:

- Do Vivado flow as explained in session 6, reach out to implementation on any FPGA Board(Ensure that you have chosen a board having enough number of i/o ports) and make sure that it’s successful implemented.
- Repeat the whole flow after using IP Catalog to replace the addition and multiplication with adder and multiplier as shown in session 6.

Q4. FSM



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

Use 3 differnet encodings (Binary, Gray and One Hot) and compare between them.

Submission Guidelines:

- The submitted file must be a PDF file.
- Name the file in the following format: StudentName_assignment6.pdf
- The PDF must include:

1- Verilog design codes.
2- Testbenches and the DO file.
3- Waveform snippets from QuestaSim showing test results.

Vivado Deliverables:

4- Snippets from the schematic after the elaboration, synthesis, Implementation and after setting up debug cores.
5- Snippets of FPGA device after implementation.
6- Snippet from the utilization & timing report & after the synthesis and implementation.
7- Snippets from IP Catalog when you modify it to insert it into your design.
8- Snippet of the "Messages" tab showing no critical warnings or errors after running elaboration, synthesis, and implementation.

Gentle Note:

Please connect inputs to switches, outputs to leds and rst to button to ensure a successful vivado operation.