CS224_2021_quiz2_B

Points: 6/20

1. Design 2-bit register. Register is loaded only when the load signal (ld) is high. Data to load comes on input wires 'din'. The register has an input 'incr' for increment function. The value is incremented when incr is high. The register works only if it is enabled by another input 'en'.

   The register increments the contents and when the value reaches all 1s the content becomes all 0s. However, to identify that the register counter has saturated, we have to indicate using an output signal 'overflow'.

   Write the Verilog code and paste it in the space below.
   (4/6 Points)

```
`timescale 1ns / 1ps

module bq1(
    input [1:0] din,
    input ld,
    input incr,
    input en,
    output reg overflow,
    output reg [1:0] q,
         input clk
    );

always @(posedge clk)
    begin
     if (en)
     begin
        if(ld)
            q = din;
          if(incr)
            q=q+1;
            if (q==3)
             begin
                q=0; overflow=1;
             end
             else
              overflow=0;
        end
    end



endmodule
```

“q become 0 as soon as it become 11 so we never see q = 11”

2. Write testbench to test your design and demonstrate the working using proper test cases. Paste output of monitor command in the space below.

   Apart from clock and initialisation, other inputs are given below.

   ld=0; en=0; din=0;

   #10; ld=1; en=1;
   #10; din=1;
   #10; incr=1;
   #10; ld=0;
   #10; din=0;
   #60;
   $stop;
   (2/4 Points)

```
time=                    0 din= 00, ld = 0 ,incr = 0, en =0, q= xx, overflow = x
time=                   10 din= 00, ld = 1 ,incr = 0, en =1, q= xx, overflow = x
time=                   15 din= 00, ld = 1 ,incr = 0, en =1, q= 00, overflow = 0
time=                   20 din= 01, ld = 1 ,incr = 0, en =1, q= 00, overflow = 0
time=                   25 din= 01, ld = 1 ,incr = 0, en =1, q= 01, overflow = 0
time=                   30 din= 01, ld = 1 ,incr = 1, en =1, q= 01, overflow = 0
time=                   35 din= 01, ld = 1 ,incr = 1, en =1, q= 10, overflow = 0
time=                   40 din= 01, ld = 0 ,incr = 1, en =1, q= 10, overflow = 0
time=                   45 din= 01, ld = 0 ,incr = 1, en =1, q= 00, overflow = 1
time=                   50 din= 00, ld = 0 ,incr = 1, en =1, q= 00, overflow = 1
time=                   55 din= 00, ld = 0 ,incr = 1, en =1, q= 01, overflow = 0
time=                   65 din= 00, ld = 0 ,incr = 1, en =1, q= 10, overflow = 0
time=                   75 din= 00, ld = 0 ,incr = 1, en =1, q= 00, overflow = 1
time=                   85 din= 00, ld = 0 ,incr = 1, en =1, q= 01, overflow = 0
time=                   95 din= 00, ld = 0 ,incr = 1, en =1, q= 10, overflow = 0
time=                  105 din= 00, ld = 0 ,incr = 1, en =1, q= 00, overflow = 1
```

“Does not count till 3”

3. Design 4-bit register using the above 2-bit registers in cascade. This will make a 4-bit up-counter, using two 2-bit up-counters. The overflow output indicates that the 4-bit counter has saturated.
   The inputs to 4-bit module are: clk, ld, incr, din.
   The outputs are: 4-bit q , 1-bit overflow

   Write the Verilog code and paste it in the space below.
   (0/6 Points)

```
`timescale 1ns / 1ps

module bq2(
    input clk,
    input ld,
    input incr,
    input [3:0] din,
    output reg [3:0] q,
    output reg overflow
    );

         reg[1:0] reg1;
         reg[1:0] reg2;




         always @(posedge clk)
    begin

                        if(reg1 == 2'b11&& reg2 == 2'b11 && incr)
                        begin
                                q= 4'b000;
                                overflow = 1;
                        end
                        else if(incr)
                        begin
                                reg1 = reg1+1;
                                if(reg1 == 3)
                                begin
                                reg1 = 0;
                                        reg2 = reg2 + 1;
                                        if(reg2 ==3)
                                        begin
                                        overflow = 1;
                                        reg2 = 0;
                                        reg1 = 0;
                                        end
                                end
                        end
                        q = {reg2, reg1};
        end
endmodule
```

4. Write testbench to test your design and demonstrate the working using proper test cases. Paste output of monitor command in the space below.

   Apart from clock and initialisation, other inputs are given below.

```
ld=0; incr=0; din=0;

#10; ld=1;
#10; din=0;
#10; incr=1;
#10; ld=0;
#10; din=0;
#160;
$stop;
```

(-/4 Points)

Enter your answer