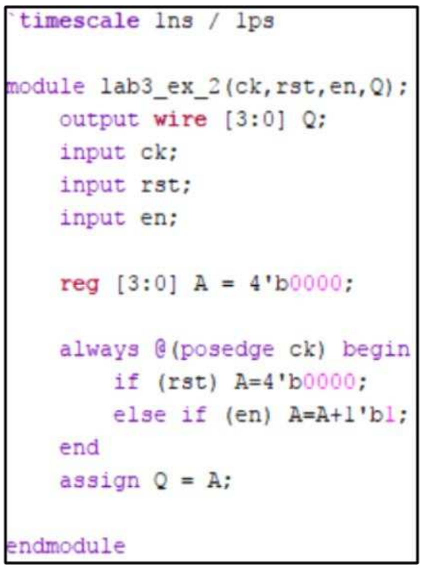
4 bit counter

Serge Hanssens

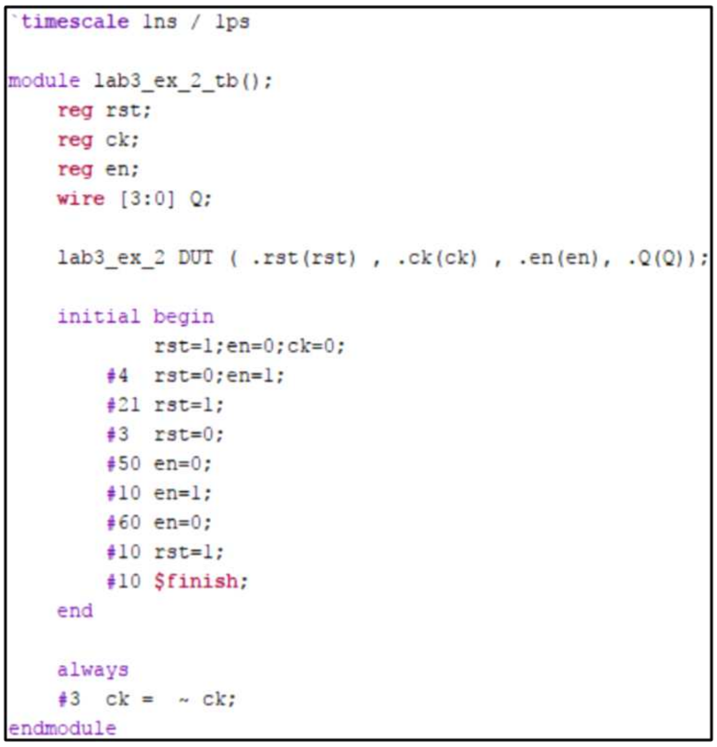
We will now design a 4-bit counter. The counter has a clock input “ck”, a reset signal “rst” and an enable signal “en”. It generates a 4 bit output “Q”. On the positive edge, ck, the counter counts if en=1. If reset is high, the counter will synchronously reset. (so a reset happens on the edge of the clock when rst=1).

We need to make a process that is clock-edge sensitive. If “rst”=1, Q has to go to zero. If “en”=1, Q=Q+1.

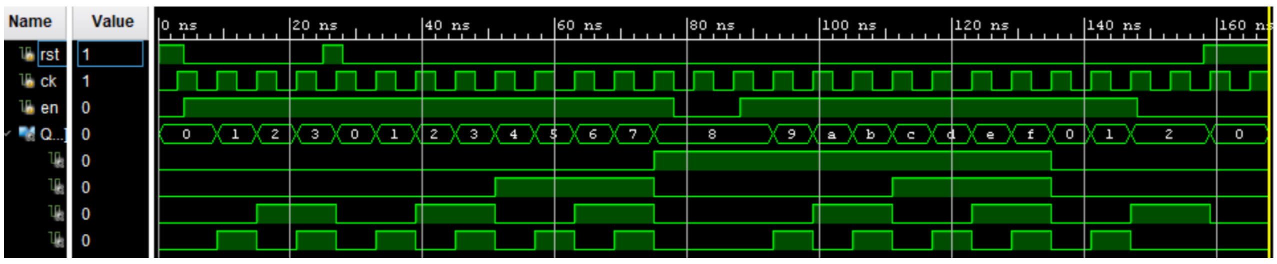
Create a testbench and synthesize the design. Look at the RTL and technology netlist.

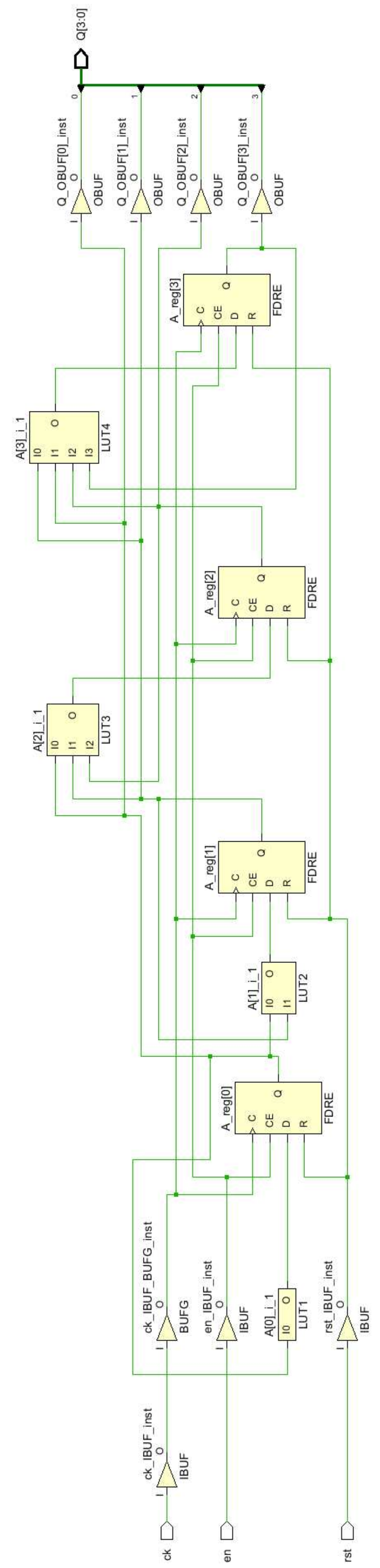


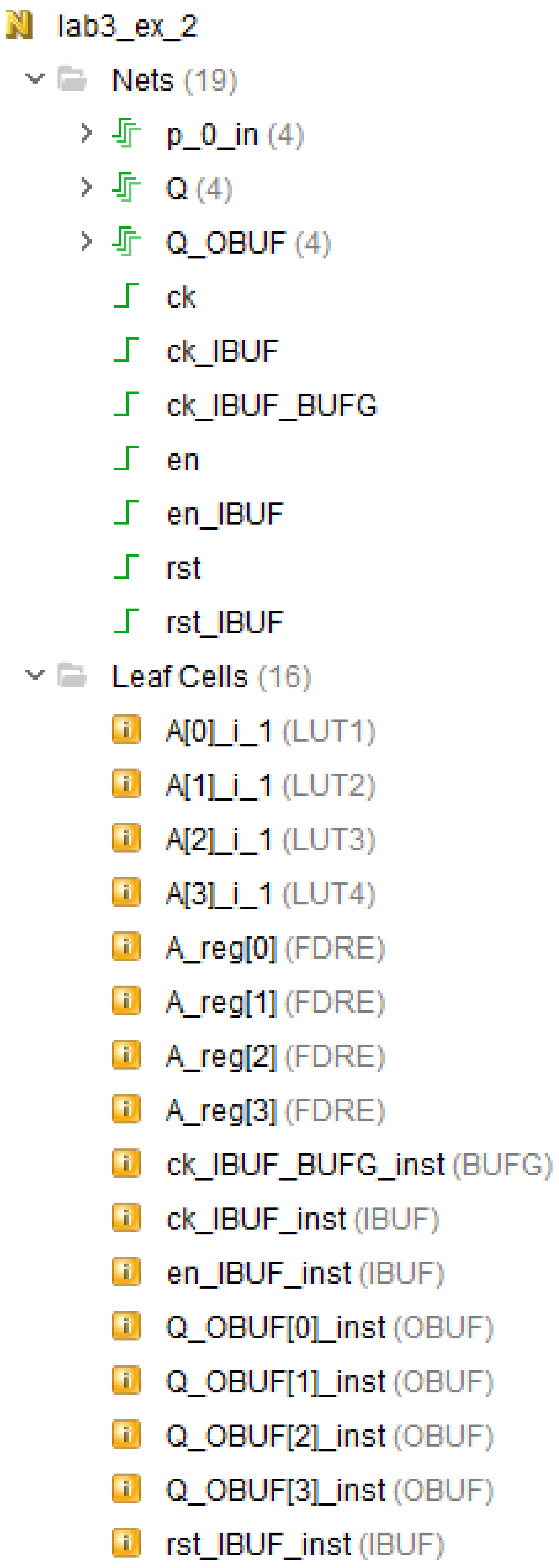
```
`timescale 1ns / 1ps

module lab3_ex_2(ck,rst,en,Q);
    output wire [3:0] Q;
    input ck;
    input rst;
    input en;

    reg [3:0] A = 4'b0000;

    always @(posedge ck) begin
        if (rst) A=4'b0000;
        else if (en) A=A+1'b1;
    end
    assign Q = A;

endmodule
```



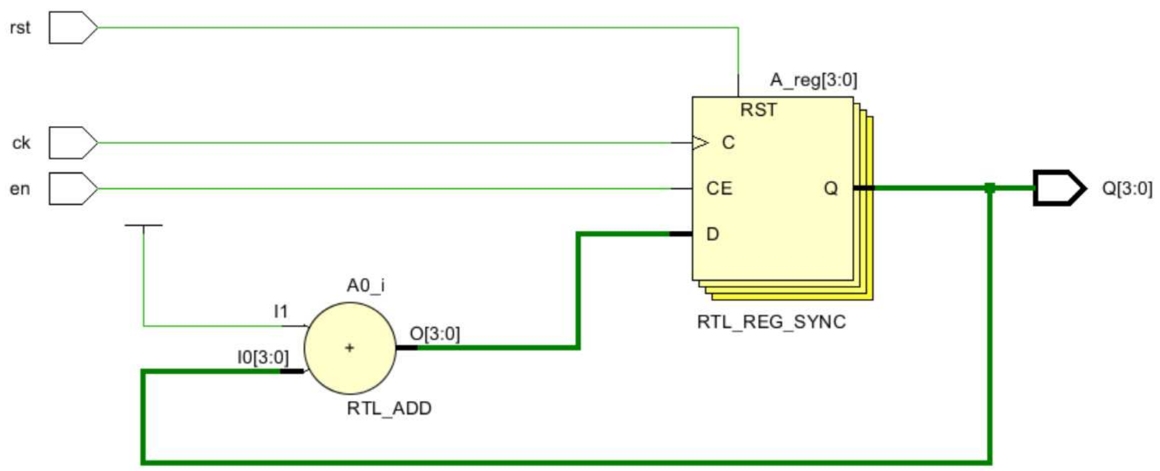
```
`timescale 1ns / 1ps

module lab3_ex_2_tb();
    reg rst;
    reg ck;
    reg en;
    wire [3:0] Q;

    lab3_ex_2 DUT ( .rst(rst) , .ck(ck) , .en(en), .Q(Q));

    initial begin
            rst=1;en=0;ck=0;
        #4  rst=0;en=1;
        #21 rst=1;
        #3  rst=0;
        #50 en=0;
        #10 en=1;
        #60 en=0;
        #10 rst=1;
        #10 $finish;
    end

    always
    #3  ck =  ~ ck;
endmodule
```

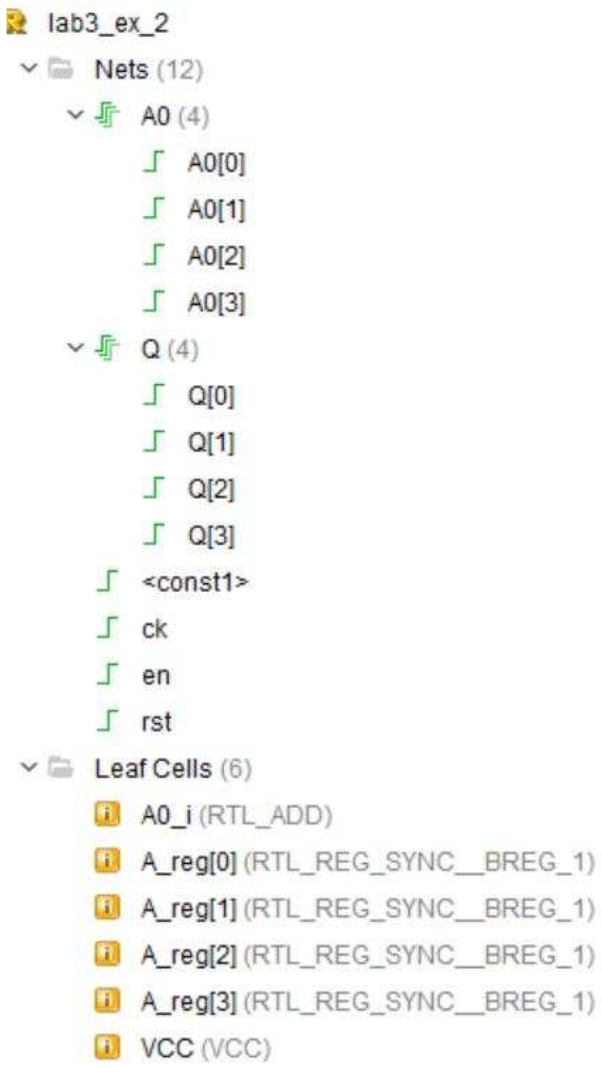





lab3_ex_2
- Nets (19)
  - p_0_in (4)
  - Q (4)
  - Q_OBUF (4)
  - ck
  - ck_IBUF
  - ck_IBUF_BUFG
  - en
  - en_IBUF
  - rst
  - rst_IBUF
- Leaf Cells (16)
  - A[0]_i_1 (LUT1)
  - A[1]_i_1 (LUT2)
  - A[2]_i_1 (LUT3)
  - A[3]_i_1 (LUT4)
  - A_reg[0] (FDRE)
  - A_reg[1] (FDRE)
  - A_reg[2] (FDRE)
  - A_reg[3] (FDRE)
  - ck_IBUF_BUFG_inst (BUFG)
  - ck_IBUF_inst (IBUF)
  - en_IBUF_inst (IBUF)
  - Q_OBUF[0]_inst (OBUF)
  - Q_OBUF[1]_inst (OBUF)
  - Q_OBUF[2]_inst (OBUF)
  - Q_OBUF[3]_inst (OBUF)
  - rst_IBUF_inst (IBUF)

RTL Schematic & RTL Netlist:





lab3_ex_2
- Nets (12)
  - A0 (4)
    - A0[0]
    - A0[1]
    - A0[2]
    - A0[3]
  - Q (4)
    - Q[0]
    - Q[1]
    - Q[2]
    - Q[3]
  - <const1>
  - ck
  - en
  - rst
- Leaf Cells (6)
  - A0_i (RTL_ADD)
  - A_reg[0] (RTL_REG_SYNC__BREG_1)
  - A_reg[1] (RTL_REG_SYNC__BREG_1)
  - A_reg[2] (RTL_REG_SYNC__BREG_1)
  - A_reg[3] (RTL_REG_SYNC__BREG_1)
  - VCC (VCC)