**实验报告**

实验题目： 寄存器和计数器 日期： 2018 年 11 月 9 日

姓名: 罗晏宸 学号: PB17000297 成绩:\_\_\_\_\_\_\_\_\_\_\_\_

**实验目的：**

1. 复习寄存器与计数器的逻辑电路基础
2. 了解具有附加控制信号的寄存器与计数器的行为
3. 设计寄存器与计数器并加以验证

**实验内容**（截图、照片与代码）

Lab6\_1\_1

仿真代码Register\_with\_synch\_reset\_load\_behavior\_tb.v

`timescale 1ns / 1ps

module Register\_with\_synch\_reset\_load\_behavior\_tb(

);

reg [3:0] D;

reg reset, load;

wire [3:0] Q;

wire Clk;

Register\_with\_synch\_reset\_load\_behavior DUT(.D(D),.Clk(Clk),.reset(reset),.load(load),.Q(Q));

GenerateClock Clock (Clk);

initial

begin

    load = 0; reset = 0; D = 4'b0000;

    #20 D = 4'b0101;

        #40 load = 1;

        #20 load = 0; D = 4'b1001;

        #40 load = 1;

        #20 load = 0;

        #15 reset = 1;

        #40 load = 1;

        #20 load = 0;

        #25 reset = 0;

        #40 load = 1;

end

endmodule

仿真代码GenerateClock.v

`timescale 1ns / 1ps

module GenerateClock(output reg Clk);

parameter tON = 5,tOFF = 5;

initial

 begin

        Clk = 0;

        forever

        begin

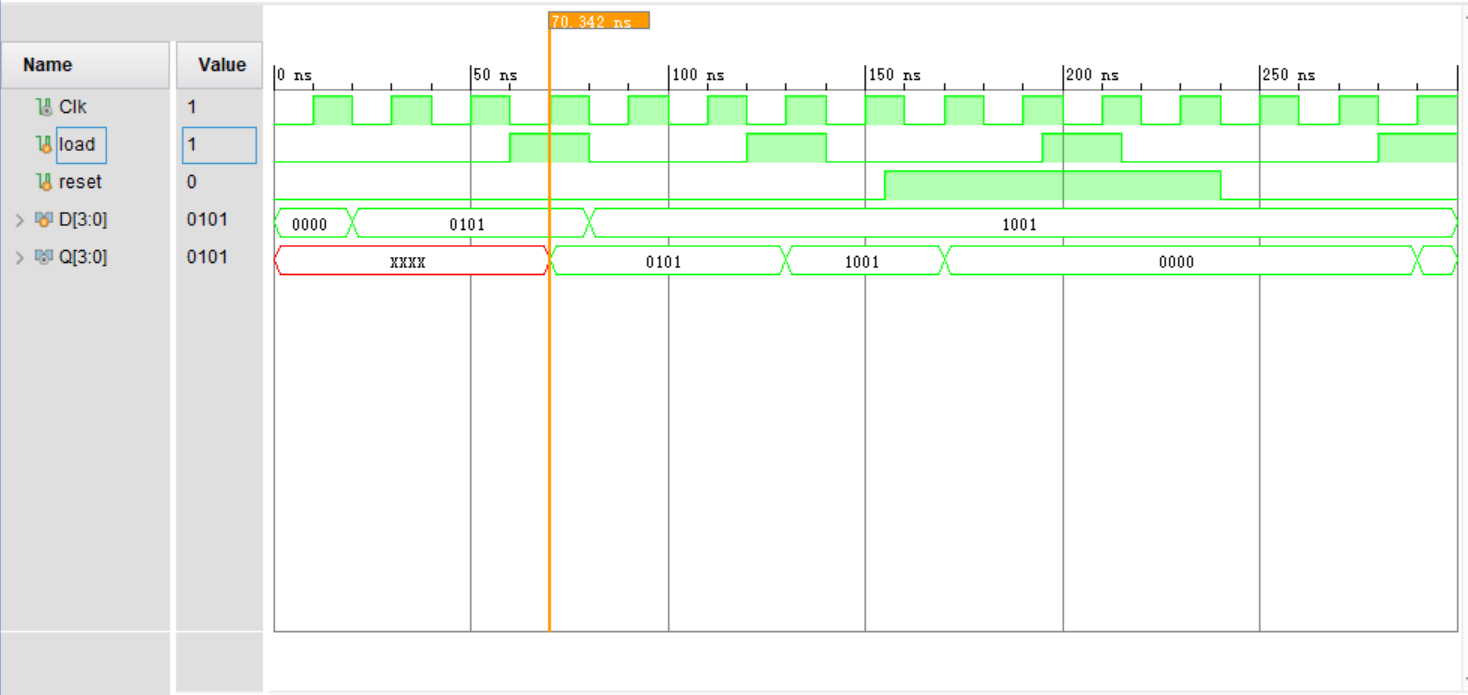
            # tOFF Clk = 1;

            # tON Clk = 0;

        end

end

endmodule



截图 1-lab6\_1\_1仿真

照片 1-lab6\_1\_1下载

“照片 1-lab6\_1\_1下载”的说明：本工程使用行为级建模设计了一个包含同步的重置和载入信号的4位寄存器，图中开关switch[0], switch[2], switch[5]与switch[15]拨至开，其余开关拨至关；led[0]与led[2]亮，其余LED灯均灭，图片表示的是向寄存器中置数(0101)­­B的过程

Lab6\_2\_2

代码Counter\_8bits.v

`timescale 1ns / 1ps

module Counter\_8bits(input Enable, input Clock, input Clear\_n, output [7:0] Q);

    wire [7:0] temp;

    assign temp[0] = Enable;

    T\_ff\_enable\_behavior TFF0 (Clock, Clear\_n, temp[0], Q[0]);

    assign temp[1] = Q[0] & temp[0];

    T\_ff\_enable\_behavior TFF1 (Clock, Clear\_n, temp[1], Q[1]);

    assign temp[2] = Q[1] & temp[1];

    T\_ff\_enable\_behavior TFF2 (Clock, Clear\_n, temp[2], Q[2]);

    assign temp[3] = Q[2] & temp[2];

    T\_ff\_enable\_behavior TFF3 (Clock, Clear\_n, temp[3], Q[3]);

    assign temp[4] = Q[3] & temp[3];

    T\_ff\_enable\_behavior TFF4 (Clock, Clear\_n, temp[4], Q[4]);

    assign temp[5] = Q[4] & temp[4];

    T\_ff\_enable\_behavior TFF5 (Clock, Clear\_n, temp[5], Q[5]);

    assign temp[6] = Q[5] & temp[5];

    T\_ff\_enable\_behavior TFF6 (Clock, Clear\_n, temp[6], Q[6]);

    assign temp[7] = Q[6] & temp[6];

    T\_ff\_enable\_behavior TFF7 (Clock, Clear\_n, temp[7], Q[7]);

endmodule

代码T\_ff\_enable\_behavior.v

`timescale 1ns / 1ps

module T\_ff\_enable\_behavior(input Clk, input reset\_n, input T, output Q);

    wire D;

    D\_ff\_negedge\_behavior DFF (D,reset\_n,Clk,Q);

    assign D = (T & ~Q) | (~T & Q);

endmodule

代码D\_ff\_negedge\_behavior.v

`timescale 1ns / 1ps

module D\_ff\_negedge\_behavior (input D, input reset\_n, input Clk, output reg Q);

    always @ (negedge Clk)

        if (!reset\_n)

                Q <= 1'b0;

        else

            if(~Clk)

                Q <= D;

endmodule

仿真代码Counter\_8bits\_tb.v

`timescale 1ns / 1ps

module Counter\_8bits\_tb();

    wire Clock;

    wire [7:0] Q;

    reg Enable, Clear\_n;

    GenerateClock CLOCK (.Clk(Clock));

    Counter\_8bits DUT (.Enable(Enable), .Clock(Clock), .Clear\_n(Clear\_n), .Q(Q));

    initial

    begin

        Enable = 0; Clear\_n = 0;

        #20 Enable = 1;

        #20 Clear\_n = 1;

        #80 Enable = 0;

        #80 Enable = 1;

    end

endmodule

仿真代码GenerateClock.v

`timescale 1ns / 1ps

module GenerateClock(output reg Clk);

parameter tON = 5,tOFF = 5;

initial

 begin

        Clk = 0;

        forever

        begin

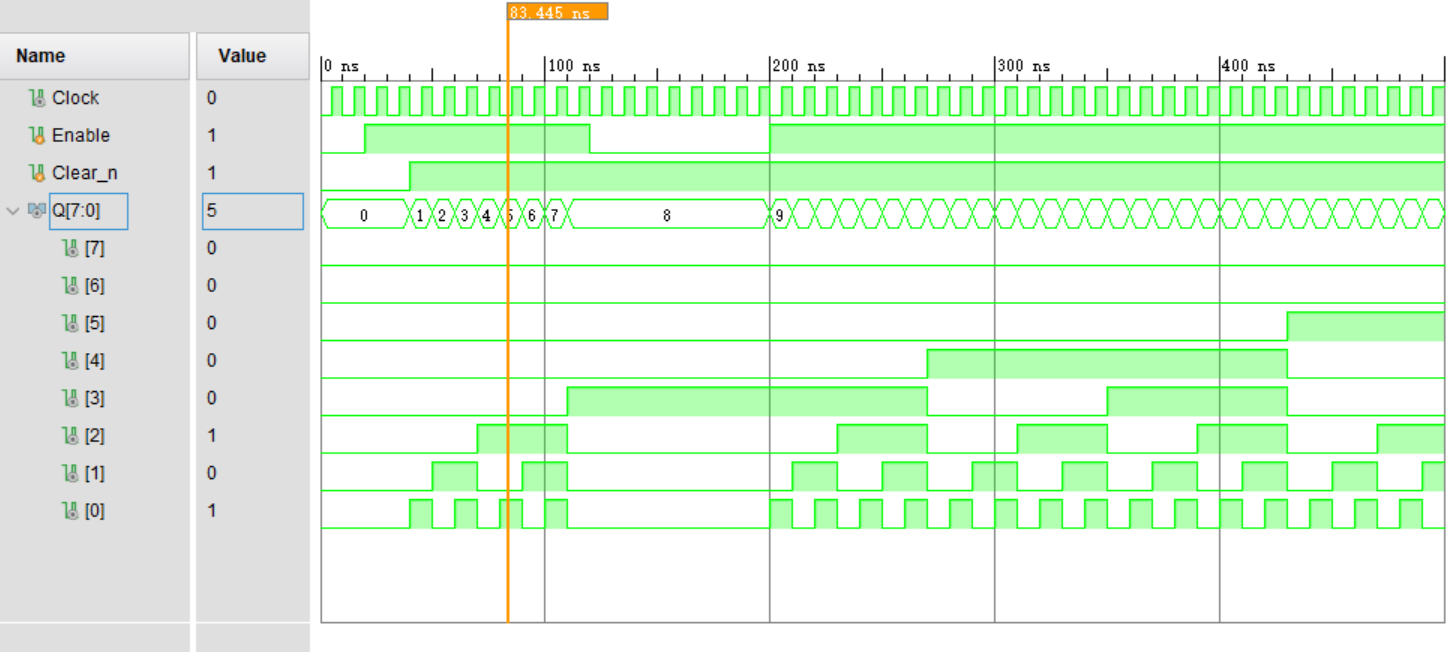
            # tOFF Clk = 1;

            # tON Clk = 0;

        end

end

endmodule



截图 2-lab6\_2\_2仿真

照片 2-lab6\_2\_2下载

“照片 2-lab6\_2\_2下载”的说明：本工程使用行为级建模设计了一个基于D触发器构建T触发器的8位计数器，图中开关switch[0], switch[1]拨至开，其余开关拨至关；led[0]与led[2]亮，其余LED灯均灭，图片表示的是计数到(00000101)**B**的过程

Lab6\_2\_3

代码Reduce\_Counter\_4bits.v

`timescale 1ns / 1ps

module Reduce\_Counter\_4bits(input Clock, input Enable, input Clear, input Load, output [3:0] Q);

reg [3:0] count;

wire cnt\_done;

assign cnt\_done = ~| count;

assign Q = count;

always @(posedge Clock)

    if (Clear)

        count <= 0;

    else

        if (Enable)

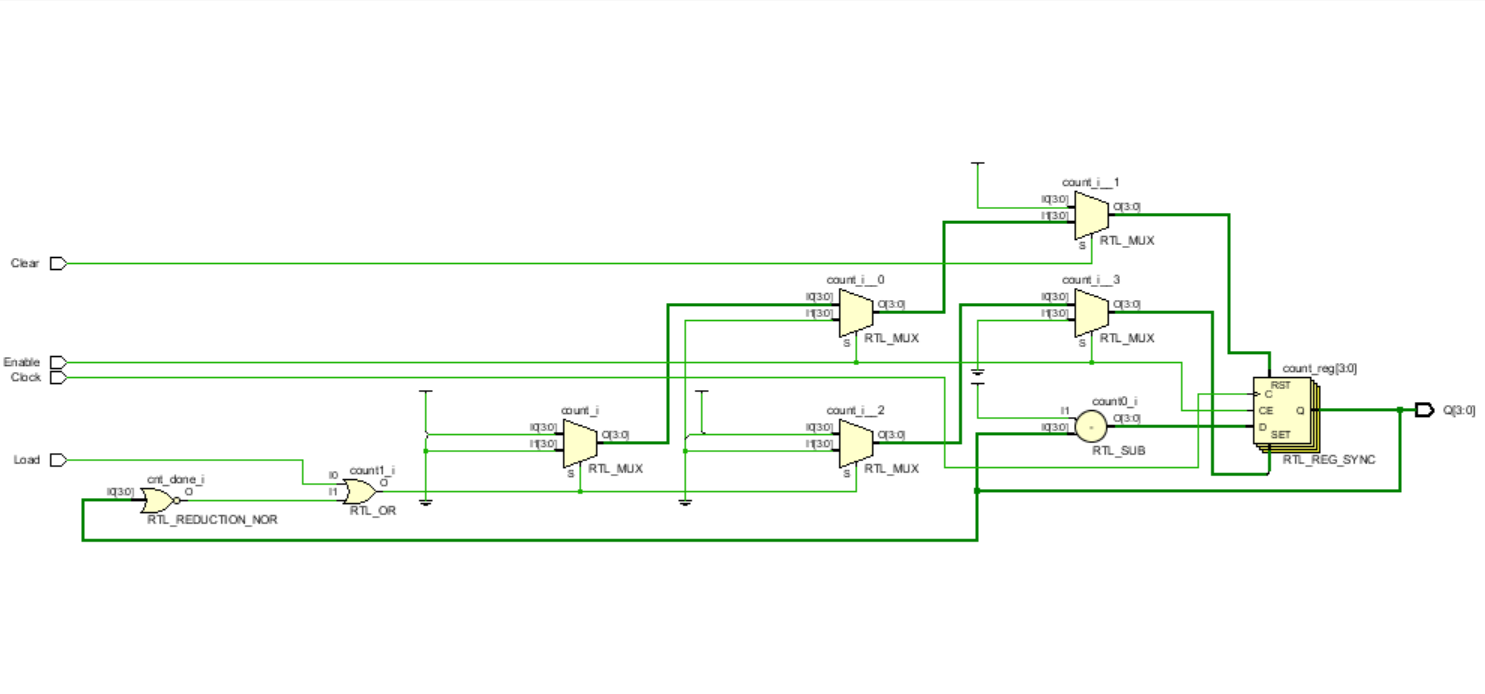
            if (Load | cnt\_done)

                count <= 4'b1010; // decimal 10

            else

                count <= count - 1;

endmodule



截图 3-lab6\_2\_3原理图

仿真代码Reduce\_Counter\_4bits\_tb.v

`timescale 1ns / 1ps

module Reduce\_Counter\_4bits\_tb();

    wire Clk;

    wire [3:0] Q;

    reg Enable, Clear, Load;

    Reduce\_Counter\_4bits DUT (.Clock(Clk), .Enable(Enable), .Clear(Clear), .Load(Load), .Q(Q));

    GenerateClock CLOCK (Clk);

    initial

    begin

        Enable = 0; Clear = 0; Load = 0;

        #20 Enable = 1;

        #20 Clear = 1;

        #20 Clear = 0;

        #20 Load = 1;

        #10 Load = 0;

        #80 Enable = 0;

        #40 Enable = 1;

    end

endmodule

仿真代码GenerateClock.v

`timescale 1ns / 1ps

module GenerateClock(output reg Clk);

parameter tON = 5,tOFF = 5;

initial

 begin

        Clk = 0;

        forever

        begin

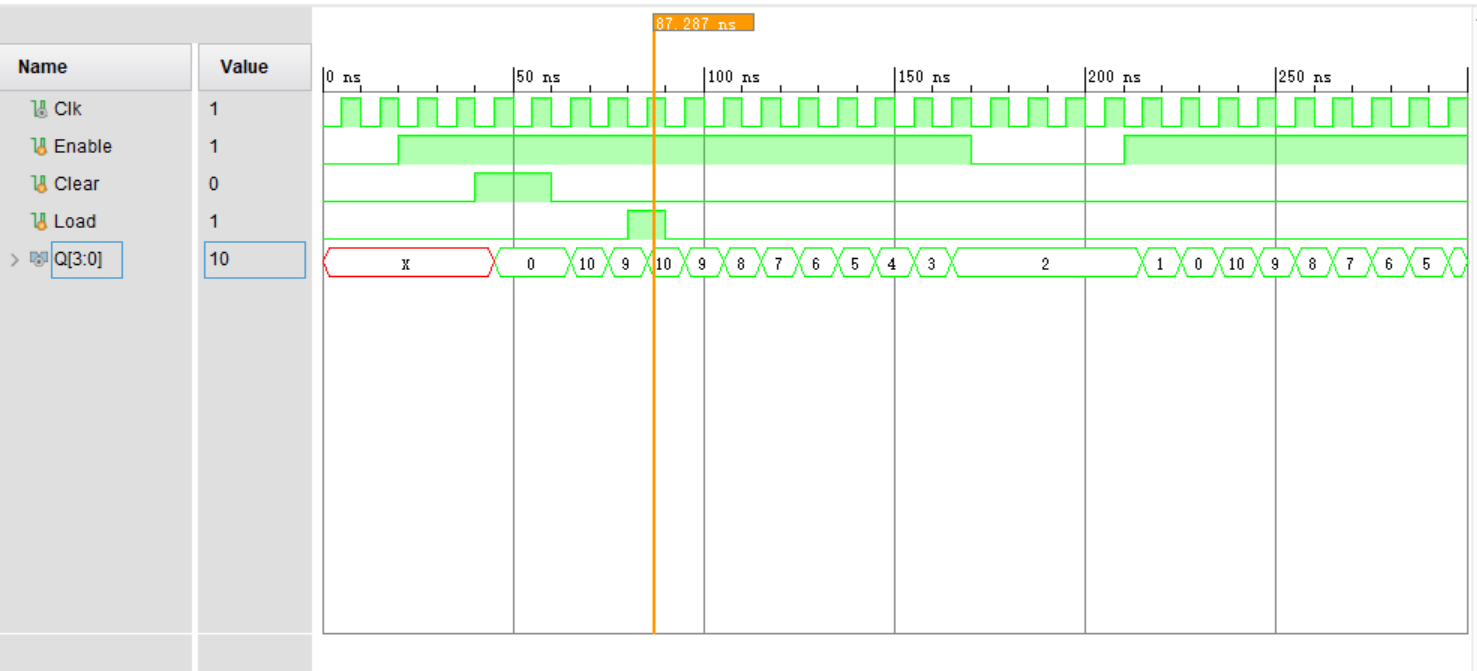
            # tOFF Clk = 1;

            # tON Clk = 0;

        end

end

endmodule



截图 4-lab\_2\_3仿真