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）

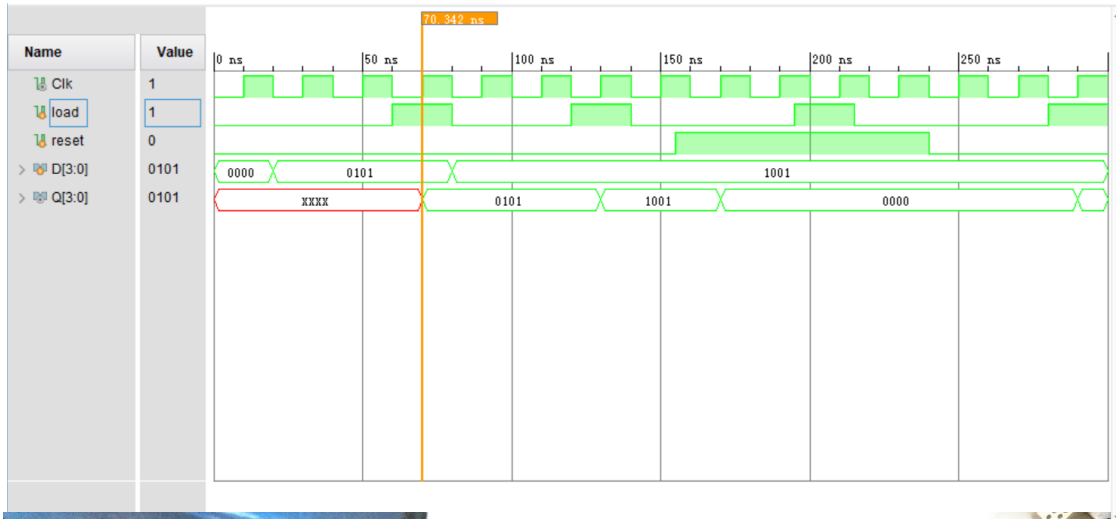
Lab6_1_1

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
```



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

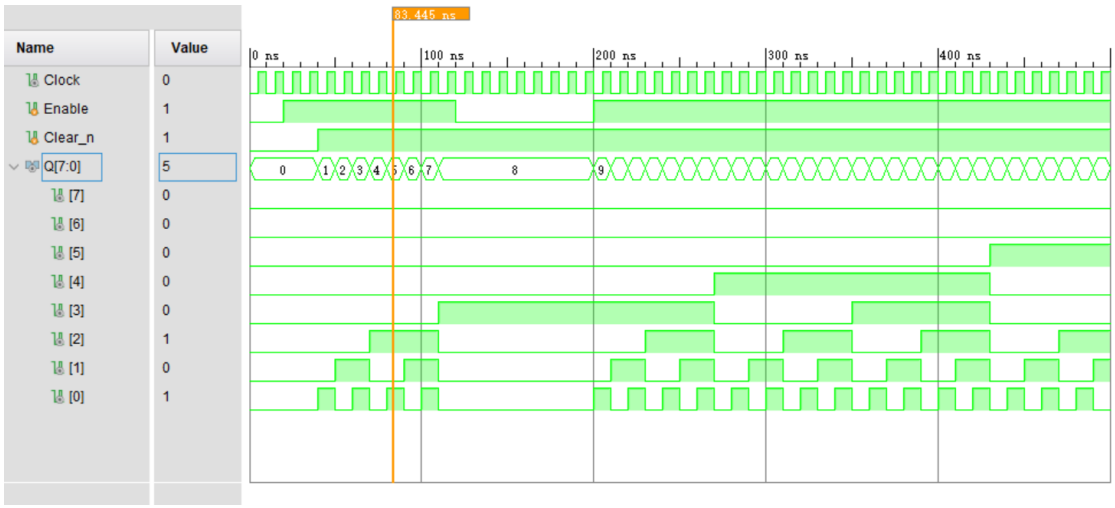
```
`timescale 1ns / 1ps
module D_ff_negedge_behavior (input D, input reset_n, input Clk, output
reg Q);
    always @ (negedge Clk)
        if (!reset_n)
                Q <= 1'b0;
        else
            if(~Clk)
                Q <= D;
endmodule
```

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

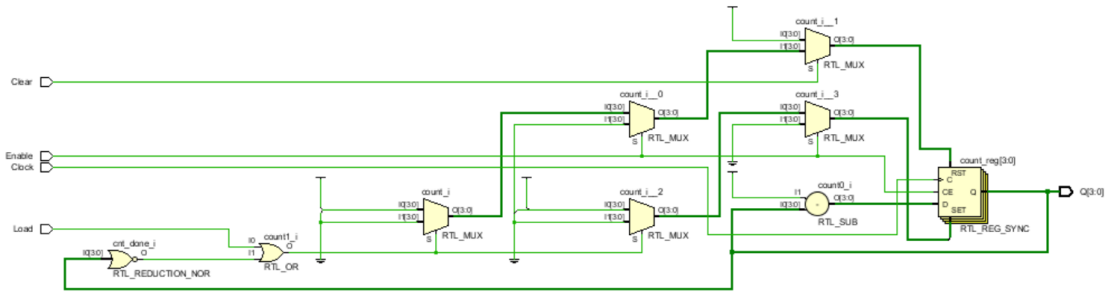
照片 2-lab6_2_2 下载

“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
```



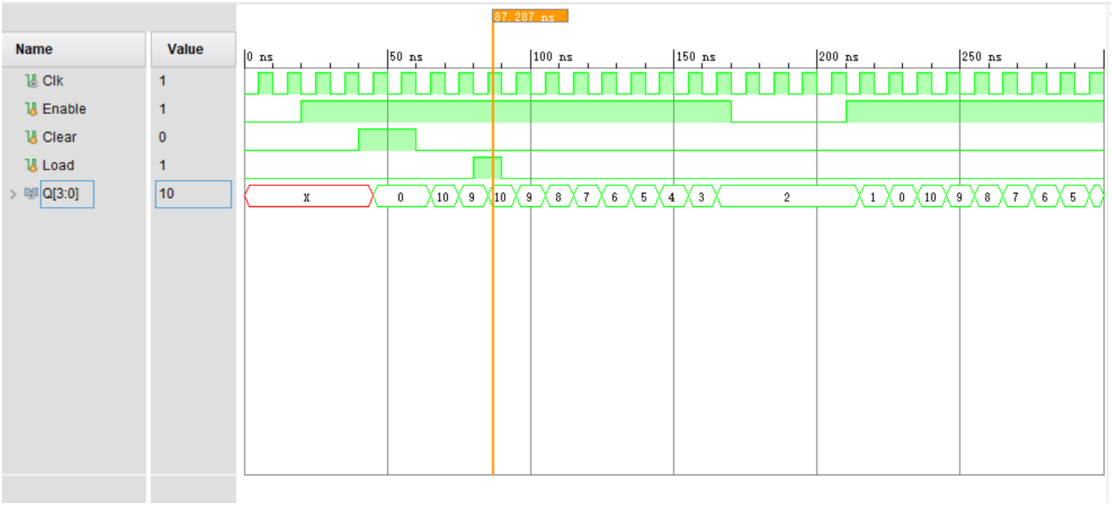
截图 3-lab6_2_3 原理图

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
```


87.287 ns
Name
Value
0 ns
50 ns
100 ns
150 ns
200 ns
250 ns
Clk
1
Enable
1
Clear
0
Load
1
Q[3:0]
10
X
0
10
9
10
9
8
7
6
5
4
3
2
1
0
10
9
8
7
6
5

截图 4-lab_2_3 仿真