

同济大学计算机系

数字逻辑课程实验报告



学 号 2252707

姓 名 陈艺天

专 业 计算机科学与技术

授课老师 张冬冬

1. 同步 D 触发器

一、实验内容

在本次实验中，我们将使用 Verilog HDL 语言实现同步 D 触发器的设计和仿真。

二、模块建模

```
module Synchronous_D_FF(  
    input CLK,  
    input D,  
    input RST_n,  
    output reg Q1,  
    output reg Q2  
);  
  
always @(posedge CLK) begin  
    if(!RST_n) begin  
        Q1=0;  
        Q2=1;  
    end  
    else begin  
        Q1 = D;  
        Q2 = !Q1;  
    end  
end  
endmodule
```

三、测试模块建模

```
module Synchronous_D_FF_tb;  
    reg CLK;  
    reg D;  
    reg RST_n;  
    wire Q1;  
    wire Q2;  
    Synchronous_D_FF inst(CLK,D,RST_n,Q1,Q2);  
    parameter t=20;  
  
    initial begin  
        CLK=0;  
        repeat(t) begin  
            #40 CLK=~CLK;
```

```

        end
    end

    initial begin
        D=1;
        #40 D=0;
        #40 D=1;
        #40 D=0;
        #40 D=1;
    end

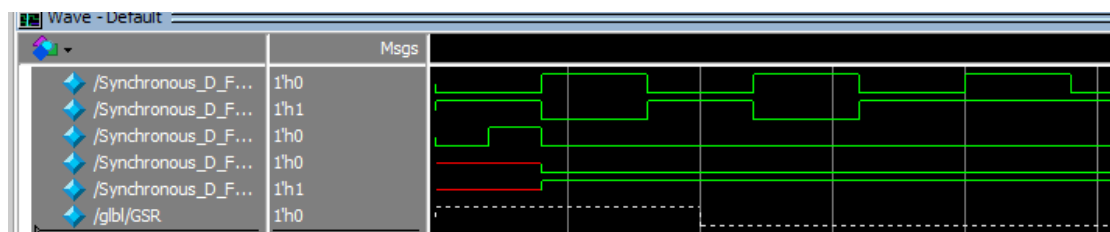
    initial begin
        RST_n=0;
        #20 RST_n=1;
        #20 RST_n=0;
        #40 RST_n=0;
    End

endmodule

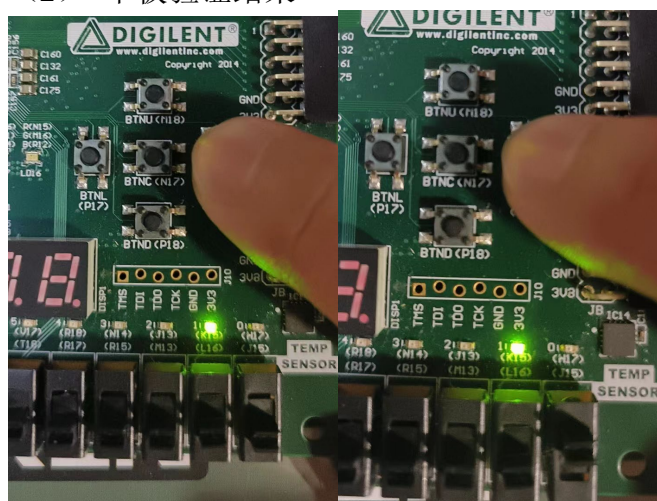
```

四、实验结果

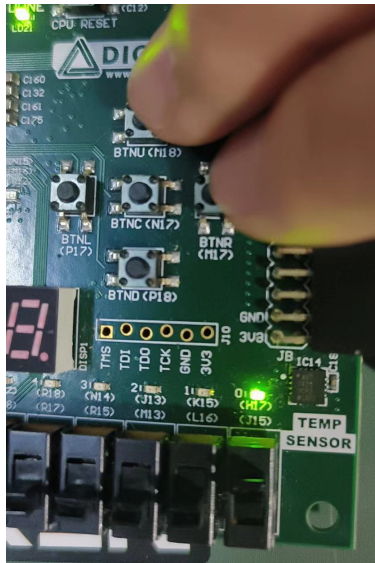
(1) Modelsim 仿真图



(2) 下板验证结果



此时 RST_n 为低电平，
无论 D 是低电平还是
高电平，时钟上升沿到
来时，始终是 0 态



此时 RST_n 为高电平, Q 的状态在时钟上升沿到来时取决于 D 的值。此时 D 为高电平, Q 为 1。

2. 异步 D 触发器

一、 实验内容

在本次实验中,我们将使用 Verilog HDL 语言实现异步 D 触发器的设计和仿真。

二、 模块建模

```
module Asynchronous_D_FF(
    input CLK,
    input D,
    input RST_n,
    output reg Q1,
    output reg Q2
);

    always @(posedge CLK or negedge RST_n)
    begin
        if (!RST_n) begin
            Q1 = 0;
            Q2 = 1;
        end
        else begin
            Q1 = D;
            Q2 = !Q1;
        end
    end
end
```

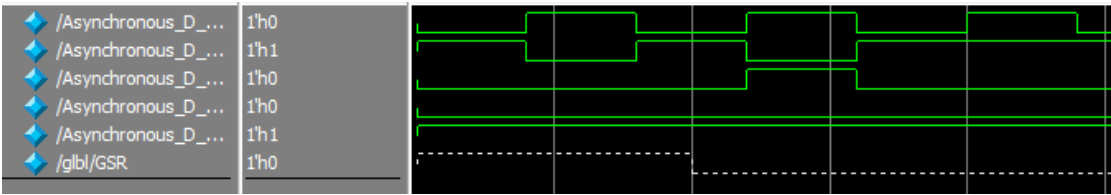
```
endmodule
```

三、 测试模块建模

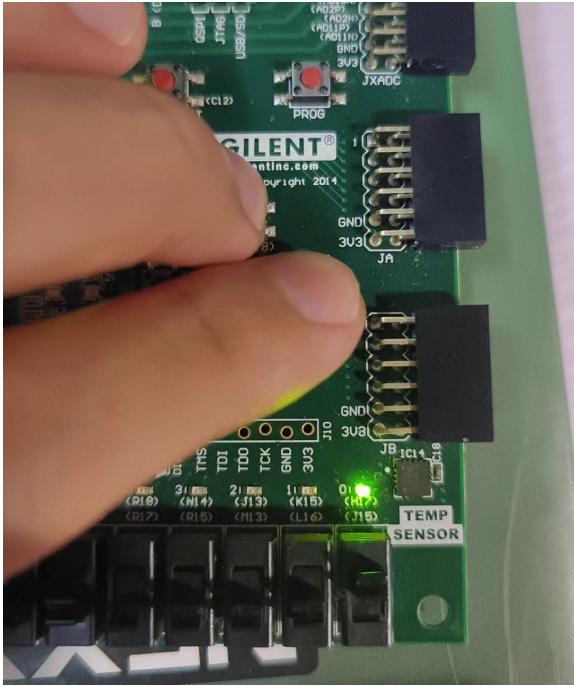
```
module Asynchronous_D_FF_tb;  
    reg CLK;  
    reg D;  
    reg RST_n;  
    wire Q1;  
    wire Q2;  
  
    Asynchronous_D_FF inst(CLK,D,RST_n,Q1,Q2);  
    parameter t=20;  
  
    initial begin  
        CLK=0;  
        repeat(t) begin  
            #40 CLK=~CLK;  
        end  
    end  
  
    initial begin  
        RST_n=0;  
        D=1;  
        #40 D=0;  
        #40 D=1;  
        #40 D=0;  
        #40 D=1;  
        #40 RST_n=0;  
    end  
  
    initial begin  
        #60 RST_n=0;  
        #60 RST_n=1;  
        #40 RST_n=0;  
    end  
  
endmodule
```

四、 实验结果

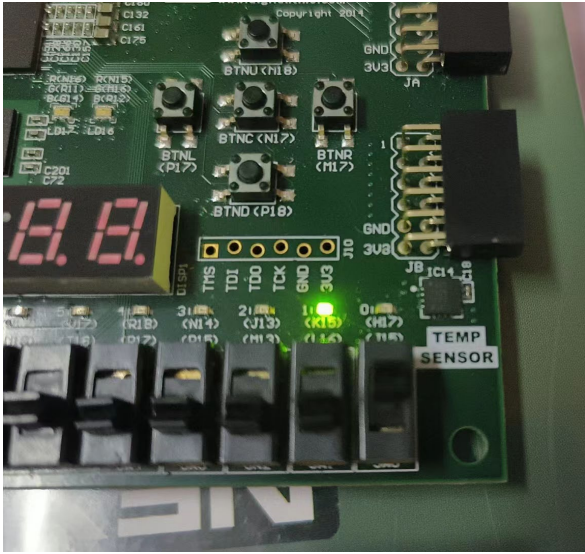
(1) Modelsim 仿真图



(2) 下板实验结果



此时置 RST_n 为高电平，通过 D 输入高电平，在时钟上升沿置为 1 态



此时置 RST_n 为低电平，无需时钟上升沿到来，置为 0 态

3. 异步 JK 触发器

一、 实验内容

在本次实验中，我们将使用 Verilog HDL 语言实现异步 JK 触发器的设计和仿真。

二、 模块建模

```
module JK_FF(  
    input CLK,  
    input J,  
    input K,  
    input RST_n,  
    output reg Q1,  
    output reg Q2  
);  
  
    always @(posedge CLK or negedge RST_n) begin  
        if (!RST_n) begin  
            Q1=0;  
            Q2=1;  
        end  
        else begin  
            if(J&K) begin  
                Q1=~Q1;  
                Q2=~Q2;  
            end  
            else if(~J&~K)  
                ;  
            else begin  
                Q1=J;  
                Q2=K;  
            end  
        end  
    end  
end  
  
endmodule
```

三、 测试模块建模

```
initial begin
    CLK=0;
    repeat(t) begin
        #20 CLK=~CLK;
    end
end
```

```
initial begin
    J=0;
    #40 J=0;
    #40 J=1;
    #40 J=1;

    #40 J=0;
    #40 J=0;
    #40 J=1;
    #40 J=1;
end
```

```
initial begin
    K=0;
    #40 K=1;
    #40 K=0;
    #40 K=1;

    #40 K=0;
    #40 K=1;
    #40 K=0;
    #40 K=1;
end
```

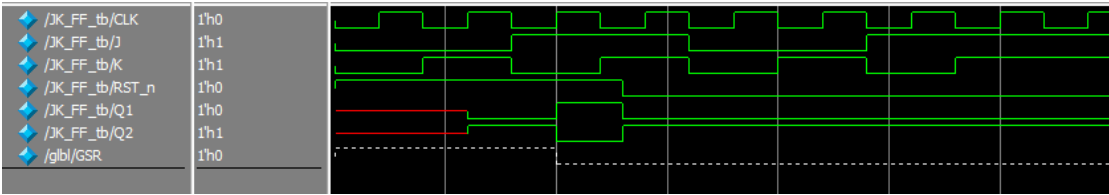
```
initial begin
    RST_n=1;

    #130 RST_n=0;
end
```

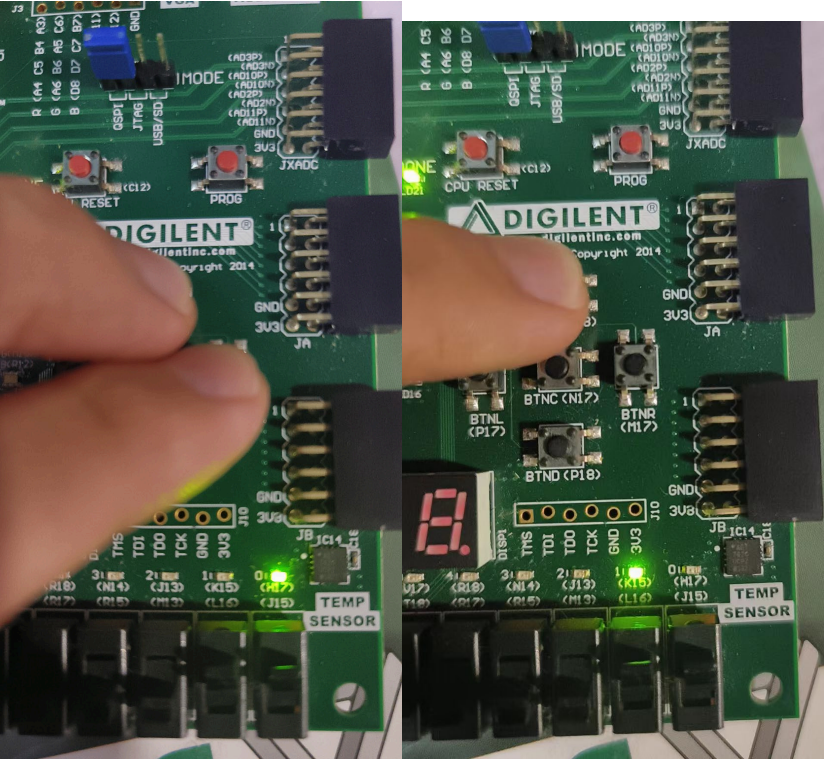
```
endmodule
```


四、 实验结果

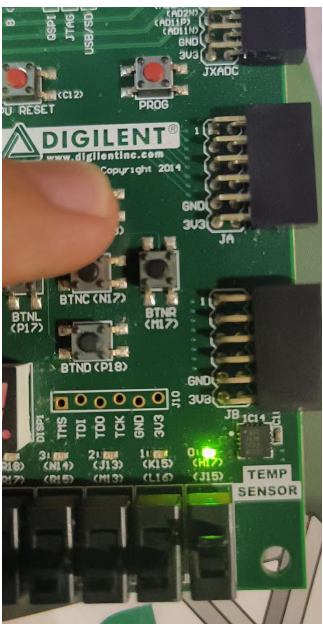
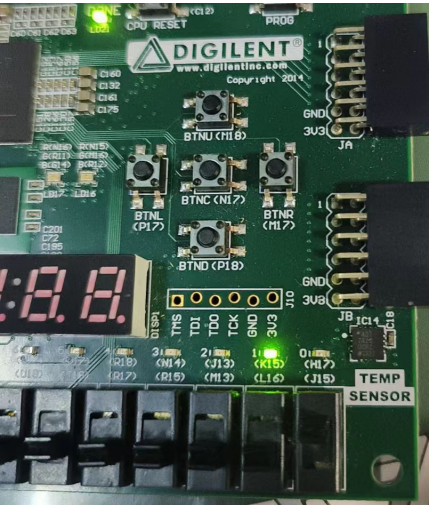
(1) Modelsim 仿真图



(2) 下板验证



J=K=1，时钟上升沿到来时，进行状态的翻转



J=1,K=0，时钟上升沿到来时置 1 态。

Rst 为低电平，置 0 态

4. 3-8 译码器

一、 实验内容

在本次实验中，我们将使用 Verilog HDL 语言实现 PC 寄存器的设计和仿真。

二、 模块建模

```
module A_D_FF(  
    input clk,  
    input rst,  
    input ena,  
    input D,  
    output reg Q1  
);  
  
always @(posedge clk or posedge rst)  
begin  
    if (rst) begin  
        Q1 = 0;  
    end  
    else begin  
        if(ena)  
            Q1 = D;  
        else  
            ;  
        end  
    end  
end  
  
endmodule  
  
module pcreg(  
    input clk,  
    input rst,  
    input ena,  
    input [31:0] data_in,  
    output [31:0] data_out  
);  
  
    A_D_FF A1(clk,rst,ena,data_in[0],data_out[0]);  
    A_D_FF A2(clk,rst,ena,data_in[1],data_out[1]);
```

```

A_D_FF A3(clk,rst,ena,data_in[2],data_out[2]);
A_D_FF A4(clk,rst,ena,data_in[3],data_out[3]);

A_D_FF A5(clk,rst,ena,data_in[4],data_out[4]);
A_D_FF A6(clk,rst,ena,data_in[5],data_out[5]);
A_D_FF A7(clk,rst,ena,data_in[6],data_out[6]);
A_D_FF A8(clk,rst,ena,data_in[7],data_out[7]);

A_D_FF A9(clk,rst,ena,data_in[8],data_out[8]);
A_D_FF A10(clk,rst,ena,data_in[9],data_out[9]);
A_D_FF A11(clk,rst,ena,data_in[10],data_out[10]);
A_D_FF A12(clk,rst,ena,data_in[11],data_out[11]);

A_D_FF A13(clk,rst,ena,data_in[12],data_out[12]);
A_D_FF A14(clk,rst,ena,data_in[13],data_out[13]);
A_D_FF A15(clk,rst,ena,data_in[14],data_out[14]);
A_D_FF A16(clk,rst,ena,data_in[15],data_out[15]);

A_D_FF A17(clk,rst,ena,data_in[16],data_out[16]);
A_D_FF A18(clk,rst,ena,data_in[17],data_out[17]);
A_D_FF A19(clk,rst,ena,data_in[18],data_out[18]);
A_D_FF A20(clk,rst,ena,data_in[19],data_out[19]);

A_D_FF A21(clk,rst,ena,data_in[20],data_out[20]);
A_D_FF A22(clk,rst,ena,data_in[21],data_out[21]);
A_D_FF A23(clk,rst,ena,data_in[22],data_out[22]);
A_D_FF A24(clk,rst,ena,data_in[23],data_out[23]);

A_D_FF A25(clk,rst,ena,data_in[24],data_out[24]);
A_D_FF A26(clk,rst,ena,data_in[25],data_out[25]);
A_D_FF A27(clk,rst,ena,data_in[26],data_out[26]);
A_D_FF A28(clk,rst,ena,data_in[27],data_out[27]);

A_D_FF A29(clk,rst,ena,data_in[28],data_out[28]);
A_D_FF A30(clk,rst,ena,data_in[29],data_out[29]);
A_D_FF A31(clk,rst,ena,data_in[30],data_out[30]);
A_D_FF A32(clk,rst,ena,data_in[31],data_out[31]);

```

```
endmodule
```

三、 测试模块建模

```
module pcreg_tb;
```

```

reg clk;
reg rst;
reg ena;
reg [31:0] data_in;
wire [31:0] data_out;
pcreg ins(clk,rst,ena,data_in,data_out);

parameter t=40;

initial begin
    clk=0;
    repeat(t) begin
        #40 clk=~clk;
    end
end

initial begin
    data_in=32'b1000_0010_1111_0101;
    #80 data_in=32'b1001_0010_1111_1000;
    #80 data_in=32'b1010_0010_1001_0111;
    #80 data_in=32'b1100_0110_1111_1101;
    #80 data_in=32'b1000_0111_1101_1001;
    #80 data_in=32'b1010_0010_0111_1111;
    #80 data_in=32'b1000_0110_1110_0000;
    #80 data_in=32'b1000_0010_1111_0101;
    #80 data_in=32'b1000_0010_0111_0001;
    #80 data_in=32'b1000_0010_1101_0111;
end

initial begin
    rst = 0 ;
    #40 rst=1;
    #160 rst=0;
end

initial begin
    ena=1;
    #80 ena=0;
    #80 ena=1;
end
endmodule

```

四、 实验结果

(1) Modelsim 仿真图

