

中国科学技术大学计算机学院
《数字电路实验》报告



实验题目：使用 Vivado 进行仿真

学生姓名：陈鸿绪

学生学号：PB21000224

完成日期：11.3

计算机实验教学中心制

2020 年 09 月

【实验题目】

使用 Vivado 进行仿真

【实验目的】

学会 Vivado 软件的下载、安装及使用；

学会使用 Verilog 编写仿真文件；

学会使用 Verilog 进行仿真，查看并分析波形文件；

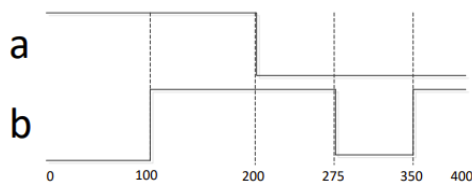
【实验环境】

实验平台：vlab.ustc.edu.cn

实验软件：Vivado

【实验练习】

T1. 请编写 Verilog 仿真文件，生成如下图所示的波形，并在 Vivado 中进行仿真。



[解]：首先观察 a、b 的波形，判断各自高电平、低电平相应的持续时间，发现：a 在 0—200 处于高电平，200—400 处于低电平；B 在 0—100 和 275—350 两个时间段处于低电平，100—275 和 350—400 两个时间端处于高电平。进而写出 Verilog 仿真代码：

```
`timescale 1ns / 1ps
```

```
module test_bench();
```

```
reg a,b;
```

```

initial
begin
    a=1;

    #200 a=0;

    #200 $stop;
end

initial
begin
    b=0;

    #100 b=1;

    #175 b=0;

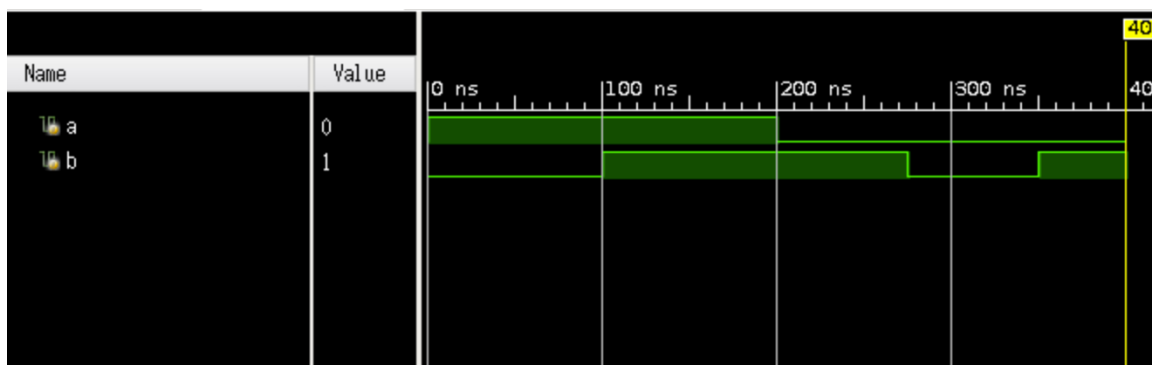
    #75 b=1;

    #50 $stop;
end

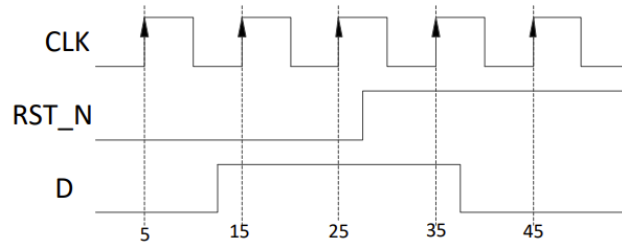
endmodule

```

在 Vivado 里进行仿真后得到波形如下截图所示：



T2. 请编写 Verilog 仿真文件，生成如下图所示的波形，并在 Vivado 中进行仿真。



[解]：对时钟信号 clk，我们可以使用 `always #5 clk=~clk;` 语句让其周期性循环，对 RST_N 与 D 两个信号，认为其上升沿与下降沿都在时钟 1/4 周期的整数倍位置，所以得到如下 Verilog 仿真代码：

```
`timescale 1ns / 1ps

module test_bench();

reg rst_n,D,clk;

initial clk=0;

always #5 clk=~clk;

initial
begin
    rst_n=0;

    #27.5 rst_n=1;

    #27.5 $stop;

end

initial
begin
```

```

        D=0;

        #12.5 D=1;

        #25 D=0;

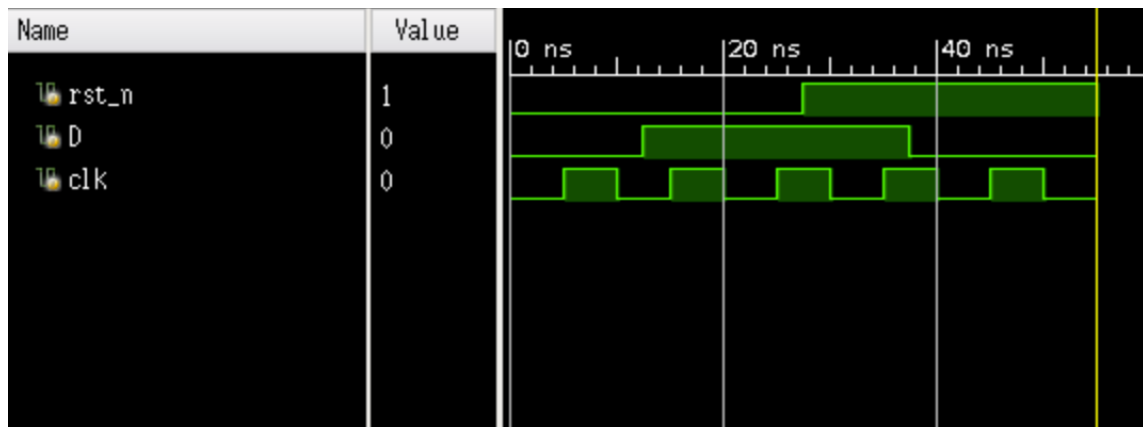
        #17.5 $stop;

    end

endmodule

```

在 Vivado 里进行仿真后得到波形如下截图所示：



T3. 利用题目 2 中的信号作为以下代码的输入，在 Vivado 中对其仿真，并观察仿真波形。

```

module d_ff_r(
    input clk, rst_n, d,
    output reg q);
    always@(posedge clk)
        begin
            if(rst_n==0)
                q <= 1'b0;

```

```

        else q <= d;
    end

endmodule

```

[解]：由题目中所给代码，可知这是一个同步复位触发器，首先新建 Verilog 设计文件，将题目中所给代码写入其中，再新建 Verilog 仿真文件，由于信号输入是题目二中的信号，所以我们有如下 Verilog 仿真文件代码：

```

`timescale 1ns / 1ps

module test_bench();

reg rst_n, d, clk;

wire q;

d_ff_r u1(.clk(clk),.rst_n(rst_n),.d(d),.q(q));

initial clk=0;

always #5 clk=~clk;

initial

    begin

        rst_n=0;

        #27.5 rst_n=1;

        #27.5 $stop;

    end

initial

begin

```

```

        d=0;

        #12.5 d=1;

        #25 d=0;

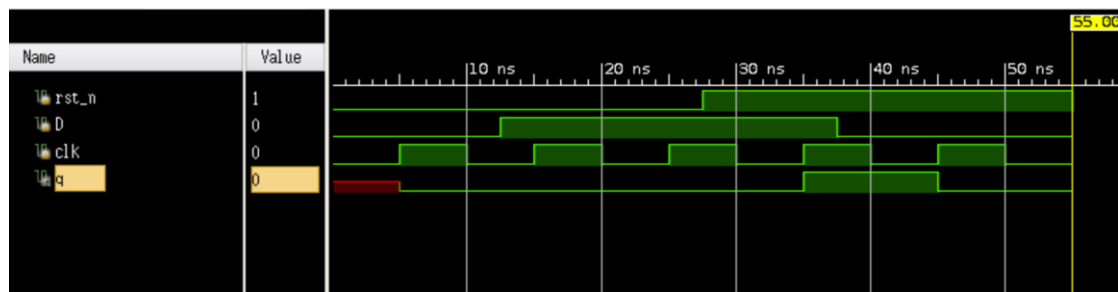
        #17.5 $stop;

    end

endmodule

```

在 Vivado 里进行仿真后得到波形如下截图所示：



不难发现，q 在初始阶段 0—5 时间段是处于高阻态的，这与 D 触发器初始阶段没有设置 q 初始值有关系。

T4. 设计一个 3-8 译码器，编写仿真测试文件，在 Vivado 中对其进行仿真。要求仿真时遍历所有的输入情况组合，给出源代码和仿真截图。

[解]：首先新建 Verilog 设计文件，将设计出的 3-8 译码器代码写入文件，该译码器可以使用 case 语句实现，代码如下：

```

`timescale 1ns / 1ps

module decode(

input [2:0] a,

output reg [7:0] o

```

```

    );
always@(*)
    begin
        case(a)
            3'b000: o=8'b00000001;
            3'b001: o=8'b00000010;
            3'b010: o=8'b00000100;
            3'b011: o=8'b00001000;
            3'b100: o=8'b00010000;
            3'b101: o=8'b00100000;
            3'b110: o=8'b01000000;
            3'b111: o=8'b10000000;
            default: o=8'b0;
        endcase
    end
endmodule

```

题中要求需要遍历输入组合情况，所以可以用 for 语句，让输入从 0 不断增 1，一直到 7 为止。具体 Verilog 仿真代码如下：

```

`timescale 1ns / 1ps

module test_bench();

    reg [2:0] a;

    wire [7:0] o;

```



```

integer i;

decode ul(.a(a),.o(o));

initial
    begin
        a=3'b000;

        for(i=0;i<7;i=i+1)

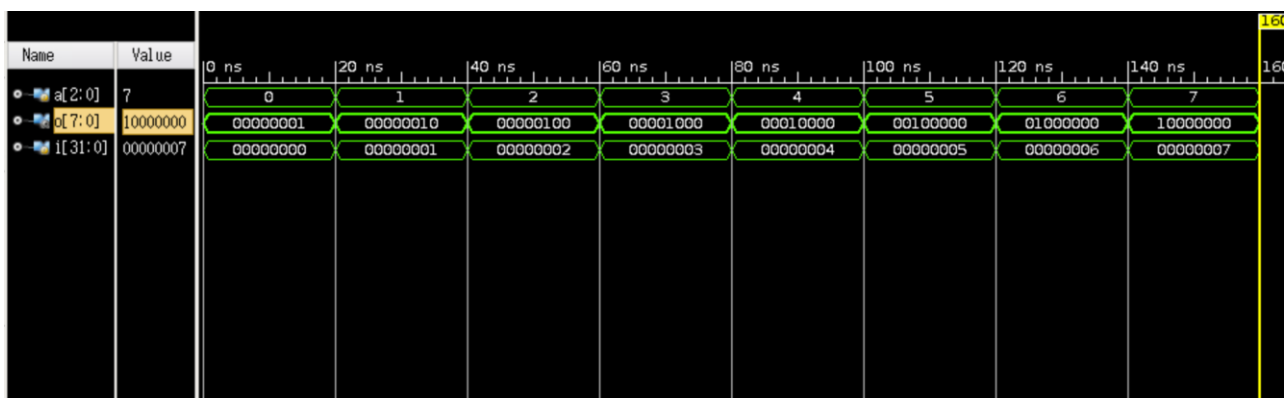
            #20 a=a+3'b001;

            #20 $stop;

        end
    endmodule

```

在 Vivado 里进行仿真后得到波形如下截图所示：



【总结与思考】

1. 收获巨大，我学会了如何使用 Vivado 进行 Verilog 仿真
2. 难易程度：一般，主要是操作方面可能一开始比较迷惑
3. 任务量适中
4. 改进建议：可以适当拓展 Vivado 操作知识