

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：舒文炫_____

学生学号：PB18000029_____

完成日期：2021.11.19_____

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】

- 熟悉 Vivado 软件的下载、安装及使用
- 学习使用 Verilog 编写仿真文件
- 学习使用 Verilog 进行仿真，查看并分析波形文件

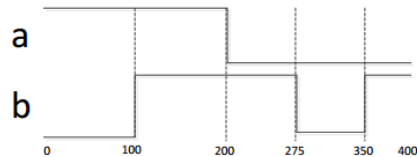
【实验环境】

- PC, win10 操作系统
- Vivado2020 版

【实验练习】

题目 1

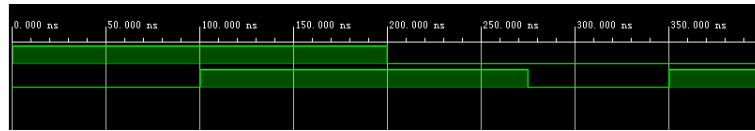
请编写 Verilog 仿真文件，生成如下图所示的波形，并在 Vivado 中进行仿真。



图中 a 波形前 200 个时间单位为 1，后 200 为 0，b 波形 100 时间单位为 0，后面 175 个单位为 1，之后又变成 0，所以仿真文件对应可写成如下，用 # 进行这个时延控制

```
> module test_bench0;  
  reg a, b;  
  initial  
  begin  
    a=1'b1;  
    #200 a=1'b0;  
    #200 $stop;  
  end  
  initial  
  begin  
    b=1'b0;  
    #100 b=1'b1;  
    #175 b=1'b0;  
    #75 b=1'b1;  
  end  
endmodule
```

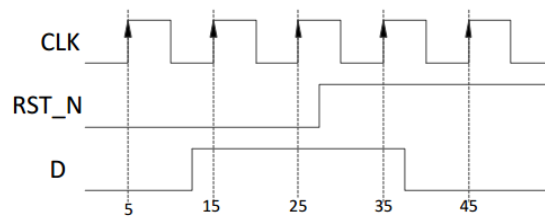
进行仿真得到波形如下：



可以看到这个波形与题目要求相同

题目 2

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。



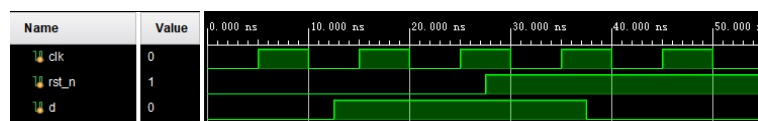
这里 clk 信号有周期性可以考虑使用 forever 语句，里面通过 #5，表示每隔 5 个时间单位变化一次，RST_N 和 D 信号和题目 1 相同做法得到仿真文件如下

```

22
23 module test_bench0;
24     reg clk, rst_n, d;
25     initial
26     begin
27         d = 1'b1;
28         #12.5 d = 1'b0;
29         #25 d = 1'b0;
30         #17.5 $stop;
31     end
32     initial
33     begin
34         clk = 1'b0;
35         forever #5 clk = ~clk;
36     end
37     initial
38     begin
39         rst_n = 1'b0;
40         #27.5 rst_n = 1'b1;
41     end
42 endmodule
43

```

仿真后得到波形如下



与题目要求相同

题目 3

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

```

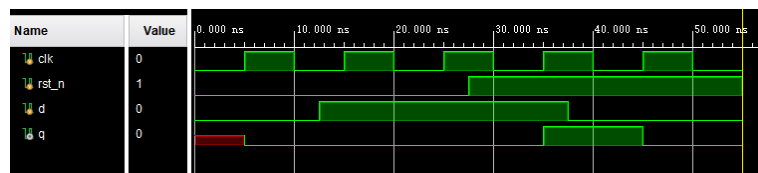
将上述代码输入到 vivado 里面，然后将题目 2 中写的仿真文件稍加修改，以调用该模块

```

module test_bench0;
    reg clk, rst_n, d;
    wire q;
    d_ff_x inst_1( clk(clk), rst_n(rst_n), d(d), q(q));
    initial
    begin
        d=1'b0;
        #12.5 d=1'b1;
        #25 d=1'b0;
        #17.5 $stop;
    end
    initial
    begin
        clk=1'b0;
        forever #5 clk=~clk;
    end
    initial
    begin
        rst_n=1'b0;
        #27.5 rst_n=1'b1;
    end
endmodule

```

得到的波形如下



题目 4

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。

3-8 译码器用 verilog 来写很容易，给个三位的信号，然后一个八位的独热码输出，这个可以很容易用 case 代码块实现，具体代码如下，这是输入高电平有效，输出低电平有效的 3-8 译码器

```

22
23 module p4(
24     input [2:0] i,
25     output reg [7:0] y
26 );
27 always@(*)
28 begin
29     case(i)
30         3'b000: y = 8'b11111110;
31         3'b001: y = 8'b11111101;
32         3'b010: y = 8'b11111011;
33         3'b011: y = 8'b11110111;
34         3'b100: y = 8'b11101111;
35         3'b101: y = 8'b11011111;
36         3'b110: y = 8'b10111111;
37         3'b111: y = 8'b01111111;
38     endcase
39 end
40 endmodule
41

```

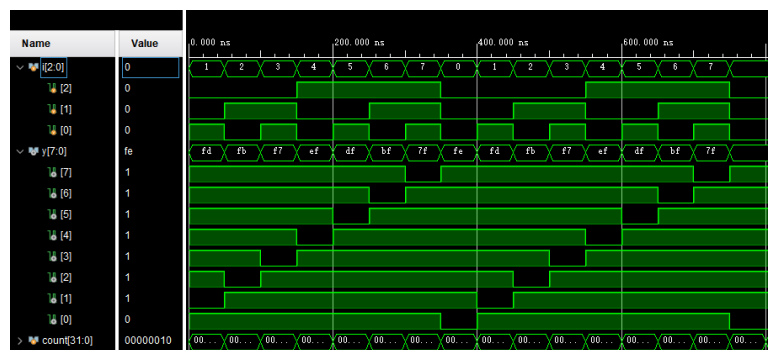
编写仿真测试文件,需要遍历所有状态,我使用了 count 内部变量,用 while 循环, $count < 16$ 进行循环, 所以这里循环 16 次, 每次循环 i+1, 这样就能遍历所有情况了, 代码如下

```

module test_bench_10:
    reg [2:0] i;
    wire [7:0] y;
    p4 inst_1(i(i), y(y));
    integer count;
    initial
    begin
        i=0;
        count=0;
        while(count<16)
        begin
            i=i+1; #50;
            count=count+1;
        end
    end
endmodule

```

输出波形如下



可以看出来我们确实实现了 3-8 译码器

【总结与思考】

1. 请总结本次实验的收获

本次实验是第一次真正上手 vivado, 进行 verilog 代码编写, 仿真, 让我初步掌握了这一语言的基本语法, 为后续实验进行打下基础

2. 请评价本次实验的难易程度

难度不高，将实验步骤过一遍基本练习就没有什么问题了

3. 请评价本次实验的任务量

任务量不高，大概可以在三四个小时内完成，主要的时间还是 vivado 自己仿真花的时间

4. 请为本次实验提供改进建议

感觉本次实验设计挺好，步骤详细，可以学到很多