

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\_\_\_\_\_

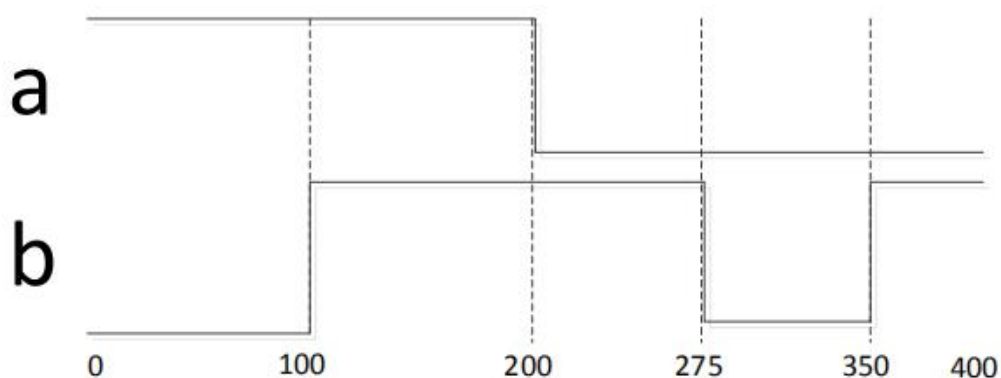
学生姓名：\_\_\_\_\_牛庆源\_\_\_\_\_

学生学号：\_\_\_\_\_PB21111733\_\_\_\_\_

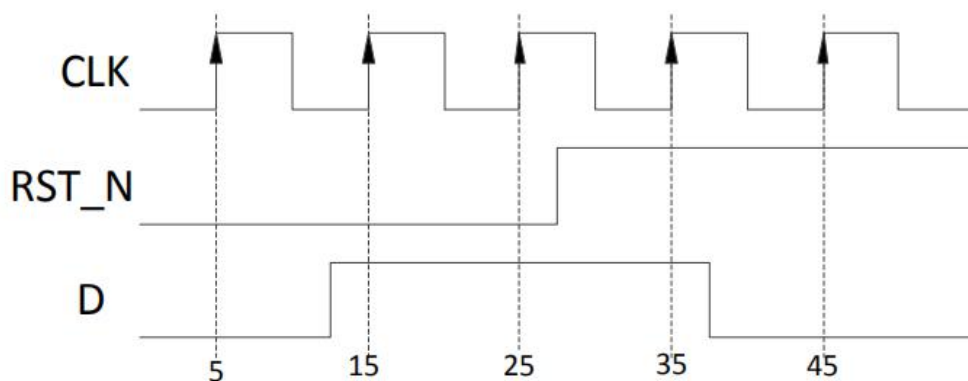
完成日期：\_\_\_\_\_2022. 11. 10\_\_\_\_\_

### 【实验题目】

1. 编写 verilog 仿真文件，生成指定波形，使用 Vivado 仿真。



2. 编写 verilog 仿真文件，生成指定波形，使用 Vivado 仿真。



3. 利用题目 2 中的信号作为一段给定代码的输入，使用 Vivado 仿真。  
并观察仿真波形。
4. 设计一个 3-8 译码器，编写仿真测试文件，在 Vivado 中进行仿真，  
要求遍历所有的输入组合。

### 【实验目的】

1. 熟悉如何编写 verilog 仿真文件，以及了解并利用 initial, \$以及#语句。
2. 熟悉如何使用 vivado 进行仿真，并可以通过仿真判断模块是否正

常运作，或者观察模块的运行特性。

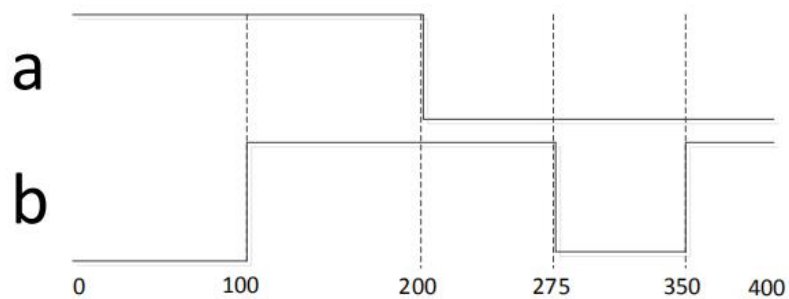
### 【实验环境】

vlab.ustc.edu.cn

### 【实验练习】

#### 题目一

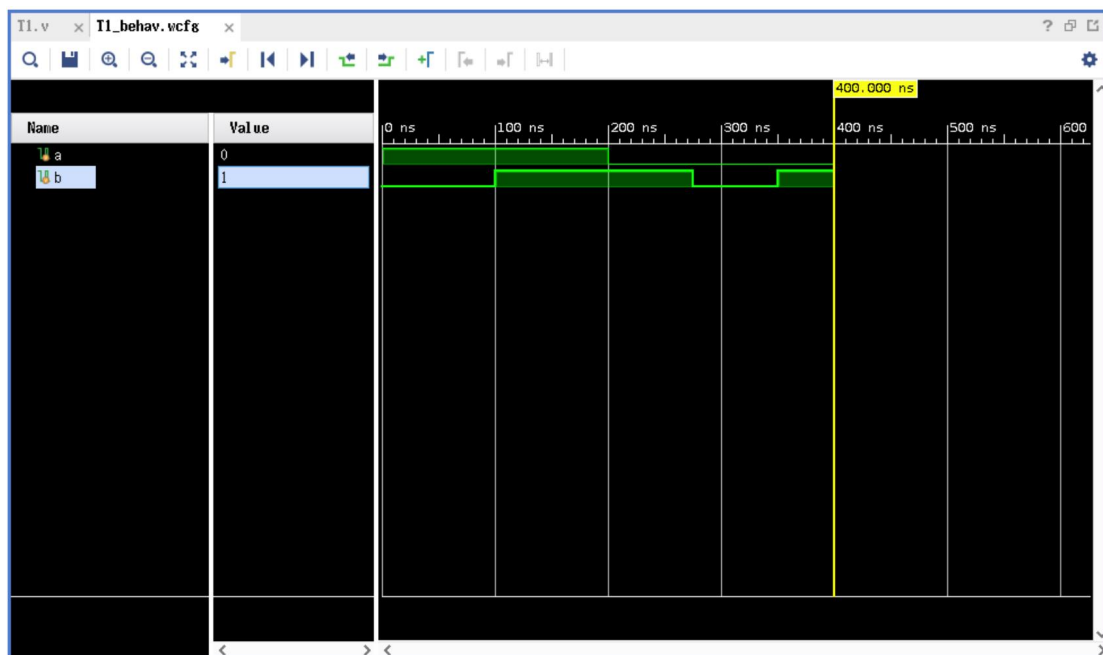
编写 verilog 仿真文件，生成下图波形，使用 Vivado 仿真。



观察波形图可得，信号 a 和 b 在时间为  
0-100, 100-200, 200-275, 275-350, 350-400 内与之前的状态相比有发  
生改变，于是有如下仿真代码：

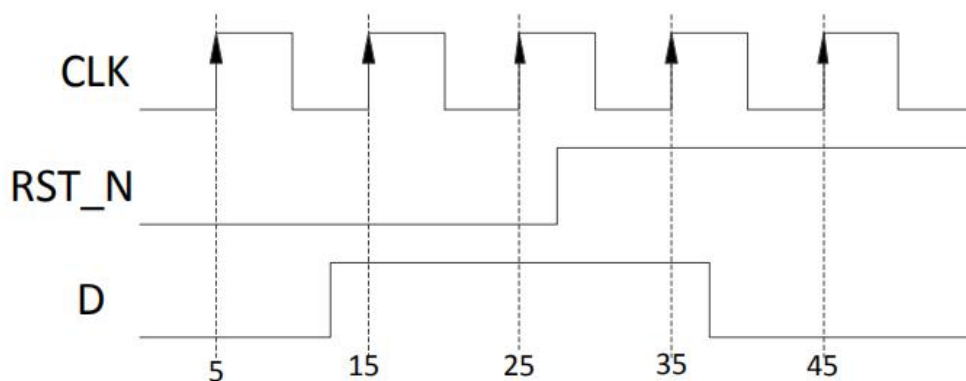
```
23 module T1( );  
24     reg a,b;  
25     initial  
26     begin  
27         a = 1'b1; b = 1'b0;  
28         #100 a = 1'b1; b = 1'b1;  
29         #100 a = 1'b0; b = 1'b1;  
30         #75 a = 1'b0; b = 1'b0;  
31         #75 a = 1'b0; b = 1'b1;  
32         #50 $finish;  
33     end  
34 endmodule  
35
```

产生的仿真波形为：



## 题目二

编写 verilog 仿真文件，生成指定波形，使用 Vivado 仿真。



观察波形可得，时钟信号每 5 个单位时间变化一次，可以使用 always 语句每 5 个单位时间将 initial 为 0 的时钟信号翻转一次。

对于剩下两个信号，分别用 initial 语句赋初值和经过某一个时间间隔后变化的值即可。

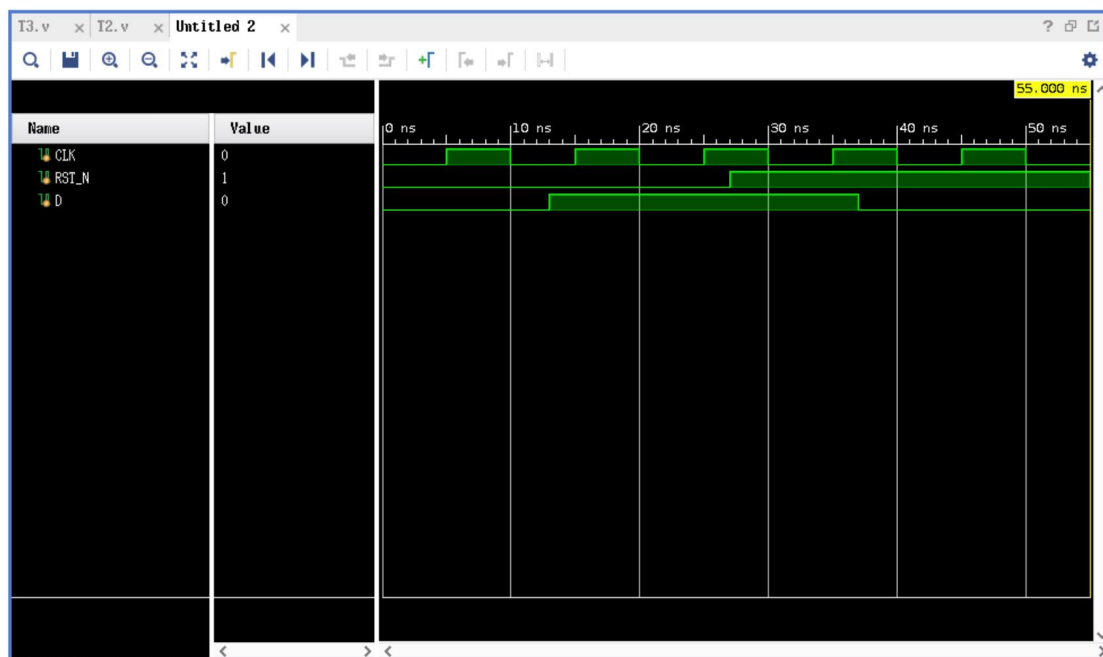
注意时钟信号在 55 处停止，所以在 55 处停止仿真。代码如下：

```

23 module T2( );
24     reg CLK, RST_N, D;
25     initial CLK = 0;
26     always #5 CLK = ~CLK;
27     initial
28     begin
29         RST_N = 0;
30         #27 RST_N = 1;
31         #28 $stop;
32     end
33     initial
34     begin
35         D = 0;
36         #13 D = 1;
37         #24 D = 0;
38     end
39
40 endmodule

```

产生的仿真波形为：



### 题目三

用题目二中的信号作为以下代码的输入，进行仿真并观察波形。

```

module T3(
input clk, rst_n, d,
output reg q);
always@(posedge clk)

```

```

begin
    if(rst_n == 0)
        q <= 1'b0;
    else
        q <= d;
    end
end
endmodule

```

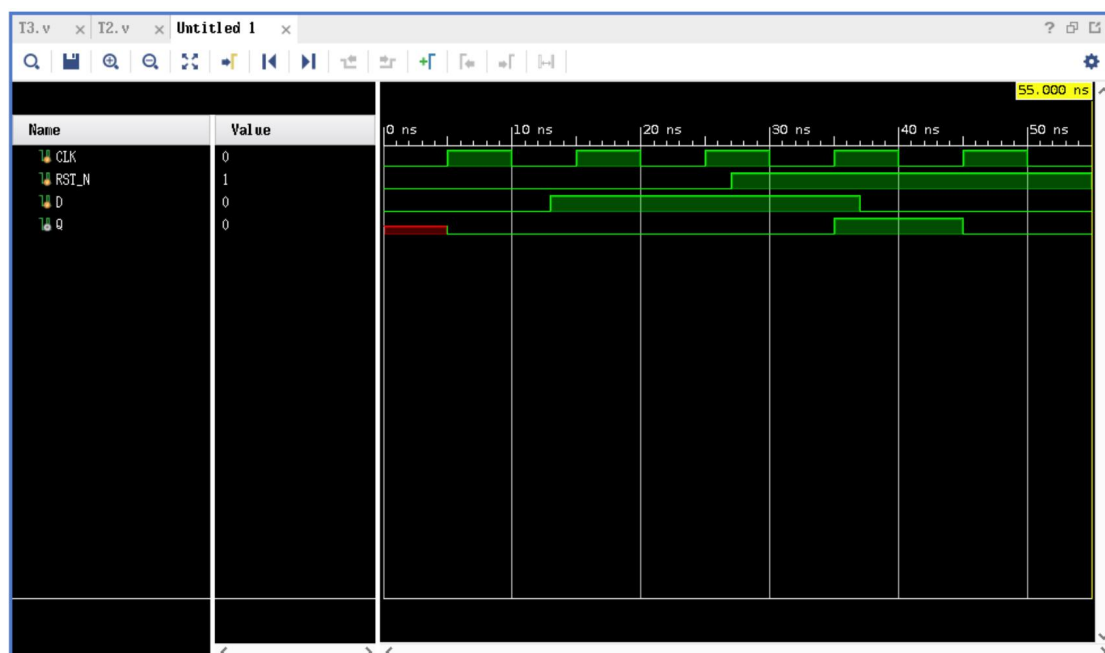
使用模块例化的方式，在题目二中加入线网类型 wire 的 Q，例化题目所给的模块，进行仿真即可：

```

23 module T2( );
24     reg CLK, RST_N, D;
25     wire Q;
26     T3 T3(CLK, RST_N, D, Q);
27     initial CLK = 0;
28     always #5 CLK = ~CLK;
29     initial
30     begin
31         RST_N = 0;
32         #27 RST_N = 1;
33         #28 $stop;
34     end
35     initial
36     begin
37         D = 0;
38         #13 D = 1;
39         #24 D = 0;
40     end
41
42 endmodule

```

仿真结果如下：



观察到在时钟信号第一个上升沿到来之前，输出信号 Q 为不确定态 x，在时钟信号第一个上升沿开始 Q 开始在每一个时钟上升沿根据 RST\_N 和 D 的值发生变化，观察波形可知，RST\_N 信号为同步复位信号，低电平有效，当 RST\_N 为 0 时，无论 D 如何变化，Q 依然为 0。当 RST\_N 为 1 时，在时钟上升沿，Q 变为 D 信号，并保持一个时钟周期，之后再更新状态。

## 题目四

设计一个 3-8 译码器，编写仿真测试文件，在 Vivado 中进行仿真，要求遍历所有的输入组合。

当使能信号 S<sub>2</sub>S<sub>1</sub>S<sub>0</sub> 为 1 0 0 时，三位信号 A 输入有效，将输出的八位信号 Y 的某一位变为 0（低电平有效）。使能信号无效时，输出为八位 1。Verilog 代码如下：

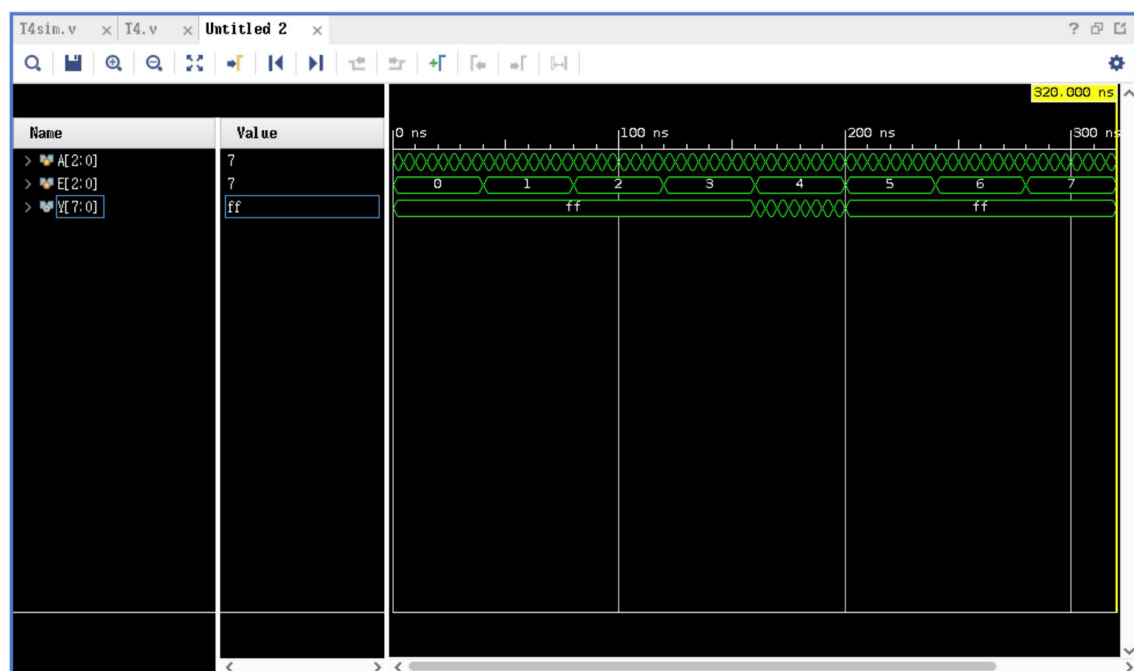
```
23 module T4(  
24     input [2:0] A, E,  
25     output reg [7:0] Y);  
26 always@(*)  
27 begin  
28     if(E[0] || E[1] || !E[2])  
29         Y = 8'b1111_1111;  
30     else  
31         case(A)  
32             3'b000: Y = 8'b0111_1111;  
33             3'b001: Y = 8'b1011_1111;  
34             3'b010: Y = 8'b1101_1111;  
35             3'b011: Y = 8'b1110_1111;  
36             3'b100: Y = 8'b1111_0111;  
37             3'b101: Y = 8'b1111_1011;  
38             3'b110: Y = 8'b1111_1101;  
39             3'b111: Y = 8'b1111_1110;  
40         endcase  
41     end  
42 endmodule  
43
```

要求遍历所有的输入组合，即遍历六位输入 00\_0000 到 11\_1111。可以分别对 A 和 E 进行遍历，A 信号初始为 000，每过 5 个单位时间

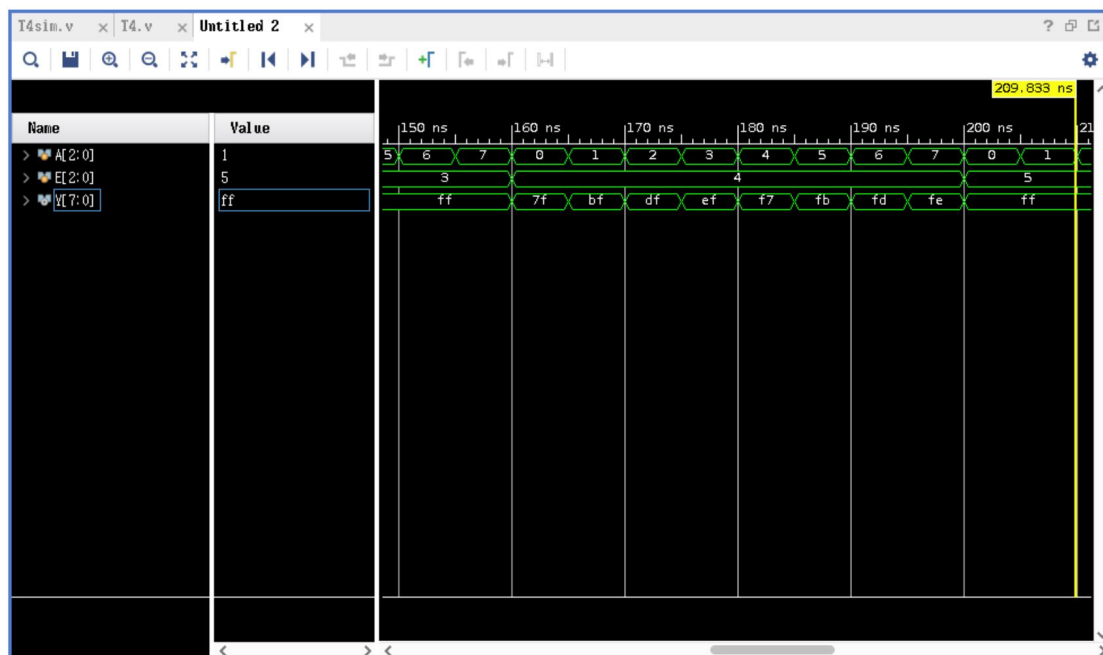
变为 A+1, E 信号, 初始为 000, 每次 A 的八种情况遍历结束后 E 变为 E+1, 即每  $5 * 8 = 40$  个单位时间变化一次。仿真在遍历完所有的输入信号后结束, 即在  $40 * 8 = 360$  时间后结束, 写出仿真代码:

```
23 module T4sim( );
24     reg [2:0] A, E;
25     wire [7:0] Y;
26     initial A = 3'b000;
27     initial E = 3'b000;
28     T4 T4(A, E, Y);
29     always #5 A = A + 1;
30     always #40 E = E + 1;
31     initial #320 $stop;
32 endmodule
```

用 vivado 进行仿真得到结果: (第一张为所有情况遍历的大图, 细节看不太清除; 第二张为使能信号有效以及前后部分的细节展示)







### 【总结与思考】

1. 初步编写了 verilog 仿真文件，对仿真文件中的一些关键字和语句有了初步了解。
2. 进行了 vivado 仿真实验，也通过仿真判断了模块是否正常运作，同时有了更直观的观察模块特性的方法。