

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：__宋玮_____

学生学号：PB20151793_____

完成日期：__2021.11.16_____

计算机实验教学中心制

【实验题目】

实验 5：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】

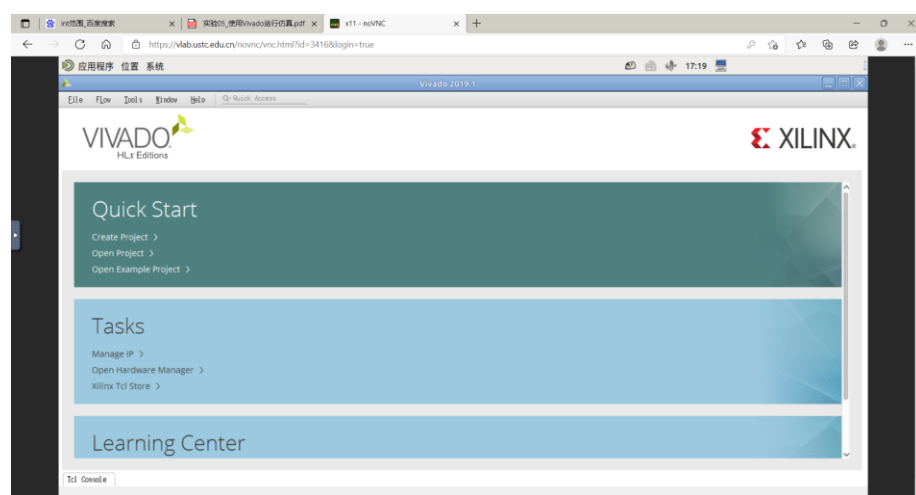
PC 一台

vlab.ustc.edu.cn

Vivado 工具

【实验过程】

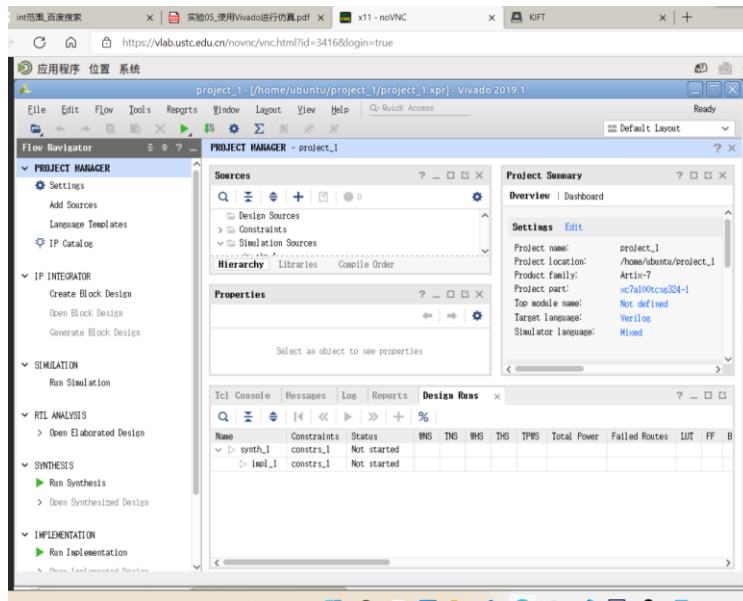
Step1. 下载并安装 Vivado 环境



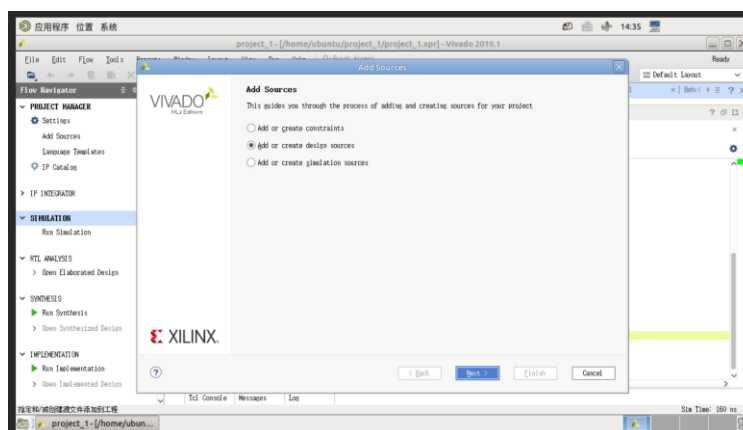
Step2. 建立 Vivado 工程

单击“Create New Project”以启动工程向导，按照向导提示建立工程，一直点击 Next 完成工程的创建。 需要注意：工程路径应为不含空格的纯英文路径、“Default Part”页面选择 xc7a100tcsg324-1

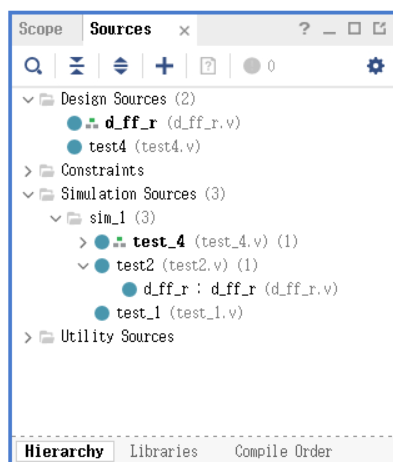
型号的器件。



Step3. 添加 Verilog 设计文件

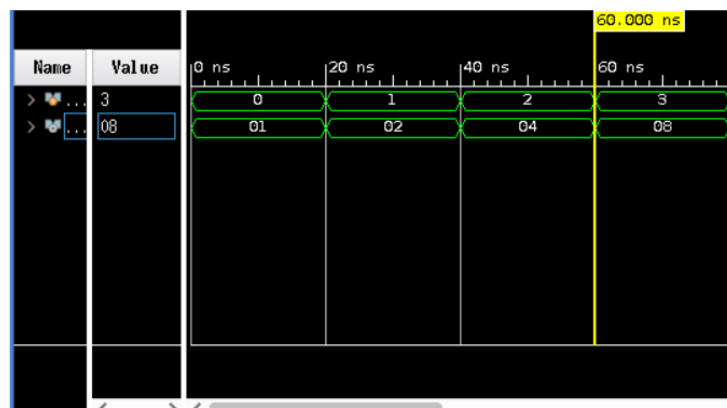


Step4. 添加仿真文件



Step5. 波形仿真

点击“Run Simulation”运行仿真工具。



Step6. Verilog 仿真文件常用语法

initial: 该关键字与 `always` 同为过程语句关键字，但与 `always` 不同的是，`initial` 语句只执行一次，`initial` 语句在模拟开始时执行，其语法结构为：

`initial` （时序控制） 过程语句

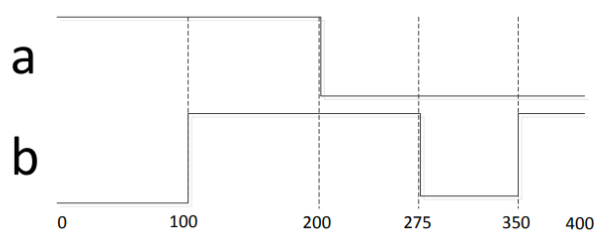
时序控制: 一般用在 `always`、`initial` 关键字后面，或者过程语句内部，常用的时序控制语句有时延控制、电平敏感事件控制和边沿触发事件控制三种。时延控制语句形式为“`#n`”，用于实现 `n` 个时间单位的延时，常用在过程语句内部；电平敏感事件控制语法格式为“`always @ (a,b,c) ...`”，`a,b,c` 三个信号的电平变化时会执行其后的过程语句，一般用来实现组合逻辑，更简洁的一种写法是“`always @ (*) ...`”；边沿敏感事件控制语法格式为“`always@(posedge clk or negedge rst_n)...`”，表示在“`clk`”信号的上升沿或“`rst_n`”信号的下降沿时执行，需要用到表示边沿事件的“`posedge`”“`negedge`”两个关键字。

循环控制：在过程语句中可以通过循环语句实现循环控制，主要包括 forever、repeat、while、for 四种，具体用法可以在上述代码中学习。

系统函数：在 Verilog 仿真文件中支持调用一些系统函数，以提高仿真效率，调用格式为：\$函数名，如\$random 表示生成一个随机数，\$finish 表示仿真结束，\$stop 表示停止仿真，\$fopen、\$fclose 用于打开和关闭文件，\$fwrite、\$fwriteb 等表示写入文件，Verilog 语法中支持的系统函数有很多，本文作为 Verilog 语法的入门介绍，不再一一列举。

【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。



```
module test_1();  
  
reg a, b;  
  
initial  
  
begin  
  
    a=1'b1; b=1'b0;
```

```

#100 a=1'b1; b=1'b1;

#100 a=1'b0; b=1'b1;

#75 a=1'b0; b=1'b0;

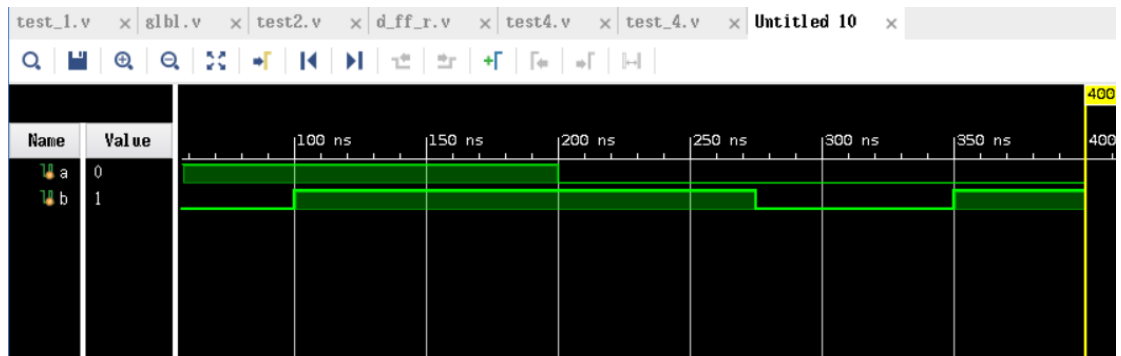
#75 a=1'b0; b=1'b1;

#50 $finish;

end

endmodule

```



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。

```

`timescale 1ns / 1ps

module test2();

reg clk;

reg rst_n;

reg d;

initial

```

```
begin

clk=1'b0;

rst_n=1'b0;

d=1'b0;

#5 clk=1'b1; rst_n=1'b0;d=1'b0;

#5 clk=1'b0; rst_n=1'b0;d=1'b0;

#3 clk=1'b0; rst_n=1'b0;d=1'b1;

#2 clk=1'b1; rst_n=1'b0;d=1'b1;

#5 clk=1'b0; rst_n=1'b0;d=1'b1;

#5 clk=1'b1; rst_n=1'b0;d=1'b1;

#3 clk=1'b1; rst_n=1'b1;d=1'b1;

#2 clk=1'b0; rst_n=1'b1;d=1'b1;

#5 clk=1'b1; rst_n=1'b1;d=1'b1;

#3 clk=1'b1; rst_n=1'b1;d=1'b0;

#2 clk=1'b0; rst_n=1'b1;d=1'b0;

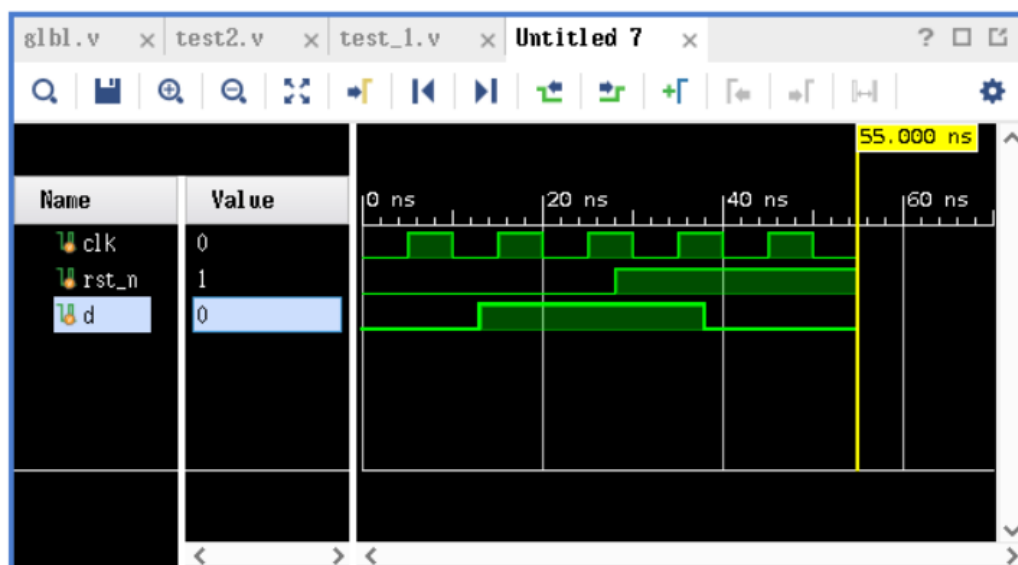
#5 clk=1'b1; rst_n=1'b1;d=1'b0;

#5 clk=1'b0; rst_n=1'b1;d=1'b0;

#5 $finish;

end

endmodule
```



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
```

仿真文件如下：


```

`timescale 1ns / 1ps

module test2();

reg clk;

reg rst_n;

reg d;

wire q;

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

initial

begin

clk=1'b0;

rst_n=1'b0;

d=1'b0;

#5 clk=1'b1; rst_n=1'b0;d=1'b0;

#5 clk=1'b0; rst_n=1'b0;d=1'b0;

#3 clk=1'b0; rst_n=1'b0;d=1'b1;

#2 clk=1'b1; rst_n=1'b0;d=1'b1;

#5 clk=1'b0; rst_n=1'b0;d=1'b1;

#5 clk=1'b1; rst_n=1'b0;d=1'b1;

#3 clk=1'b1; rst_n=1'b1;d=1'b1;

#2 clk=1'b0; rst_n=1'b1;d=1'b1;

#5 clk=1'b1; rst_n=1'b1;d=1'b1;

#3 clk=1'b1; rst_n=1'b1;d=1'b0;

```

```

#2 clk=1'b0; rst_n=1'b1;d=1'b0;

#5 clk=1'b1; rst_n=1'b1;d=1'b0;

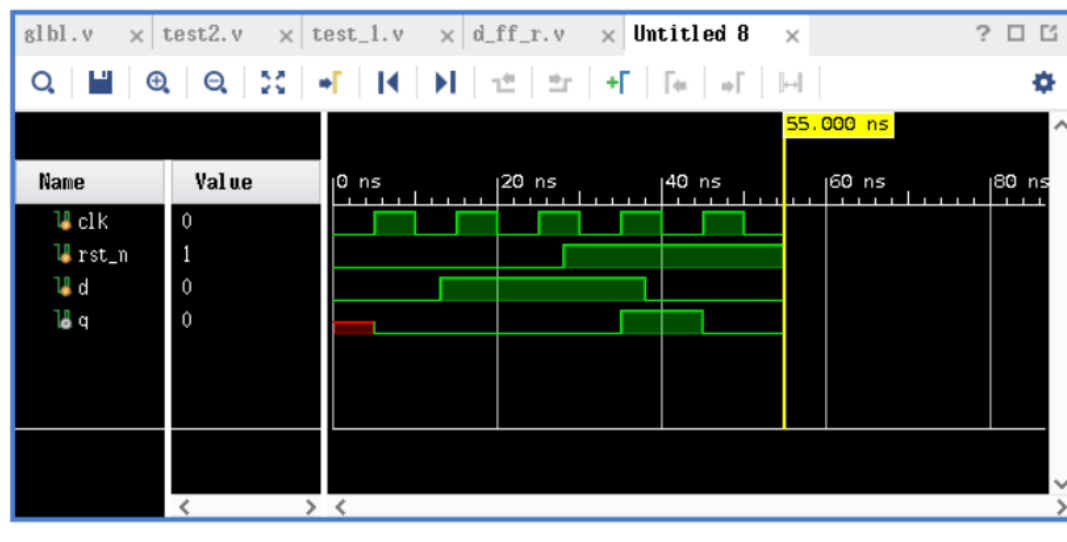
#5 clk=1'b0; rst_n=1'b1;d=1'b0;

#5 $finish;

end

endmodule

```



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。

(1) 设计文件:

```

module test4(
    input [2:0] a,

```

```

output reg [7:0] y

);

always@(*)
begin
    if(a==3'b000) y[7:0]=8'b00000001;
    else if(a==3'b001) y[7:0]=8'b00000010;
    else if(a==3'b010) y[7:0]=8'b00000100;
    else if(a==3'b011) y[7:0]=8'b00001000;
    else if(a==3'b100) y[7:0]=8'b00010000;
    else if(a==3'b101) y[7:0]=8'b00100000;
    else if(a==3'b110) y[7:0]=8'b01000000;
    else y[7:0]<=8'b10000000;
end

endmodule

```

(2) 仿真文件:

```

`timescale 1ns / 1ps

module test_4( );

    reg [2:0] a;

    wire [7:0] y;

    test4 test4(.a(a),.y(y));

    initial

```

```

begin

    a=3'b000;

#20 a=3'b001;

#20 a=3'b010;

#20 a=3'b011;

#20 a=3'b100;

#20 a=3'b101;

#20 a=3'b110;

#20 a=3'b111;

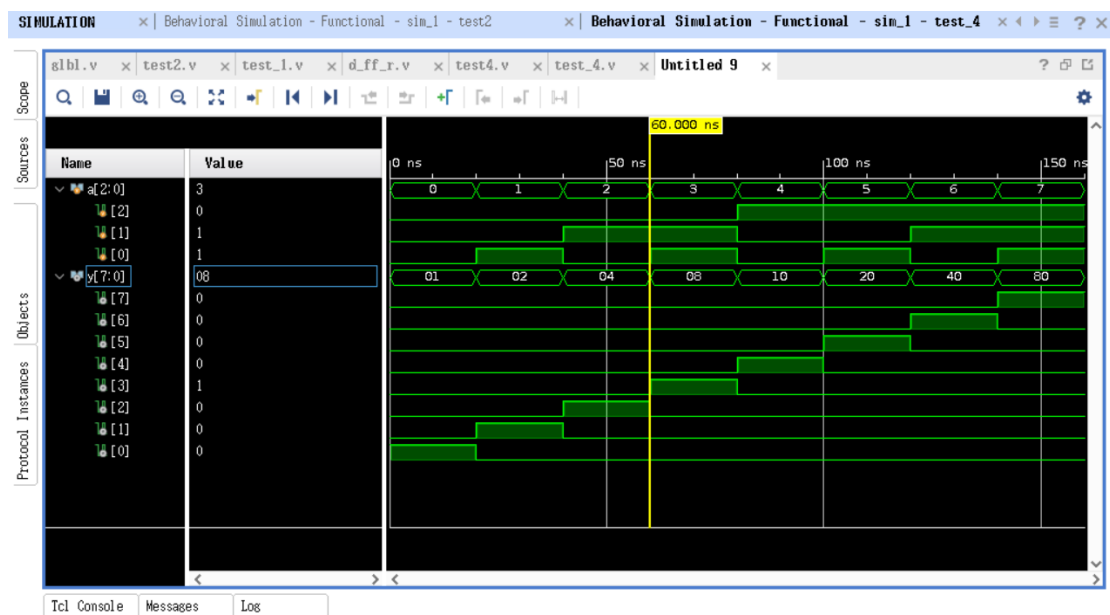
#20 $finish;

end

endmodule

```

(3) 仿真截图:



【总结与思考】

通过本次实验，初步了解并掌握了利用 vivado 进行仿真的过程与方法。实验难度适中，实验用时不长，无改进建议。