

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：王章瀚

学生学号：PB18111697

完成日期：2019/10/28

计算机实验教学中心制

2019 年 09 月

1 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

2 实验环境

PC 一台
Windows 或 Linux 操作系统
Vivado 工具
vlab.ustc.edu.cn（包含 Vivado 下载安装及使用教程）

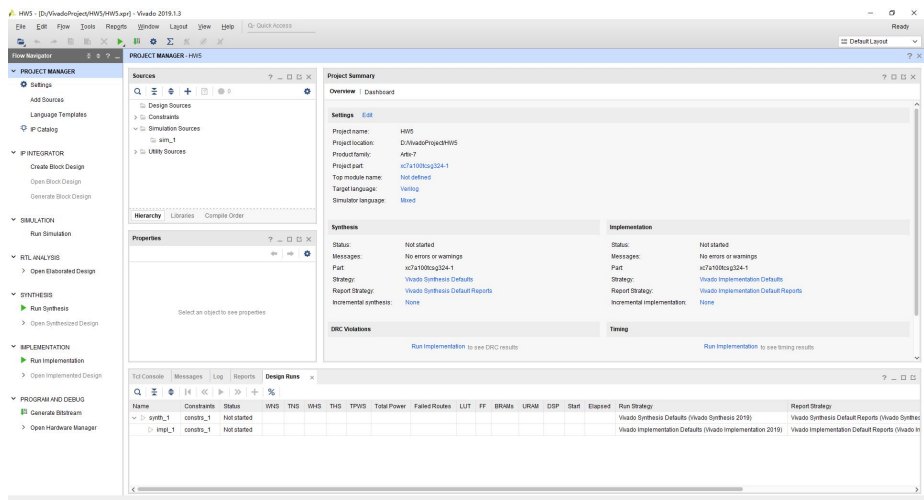
3 实验过程

3.1 下载并安装 Vivado 环境

本人利用校园网直接从赛灵思官方网站下载成功。

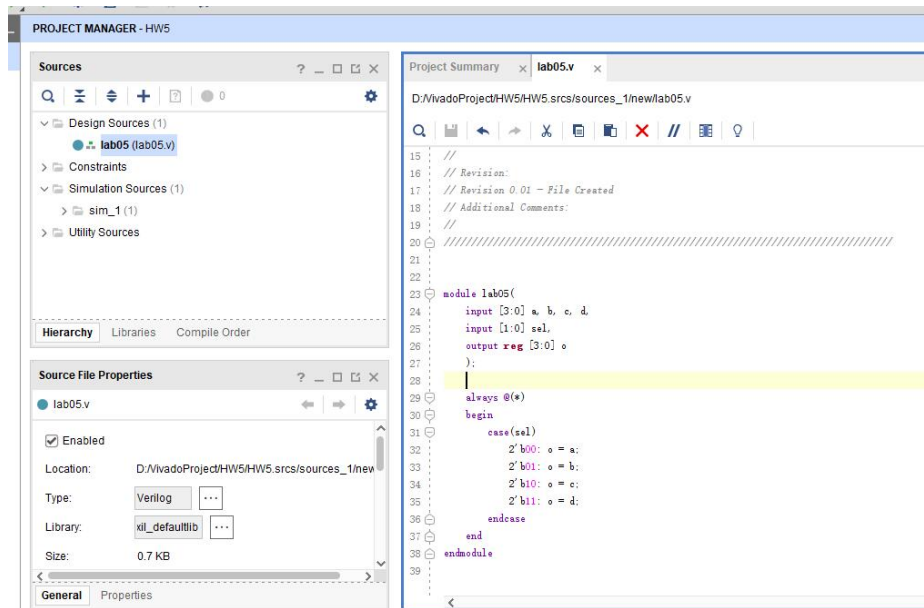
3.2 建立 Vivado 工程

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



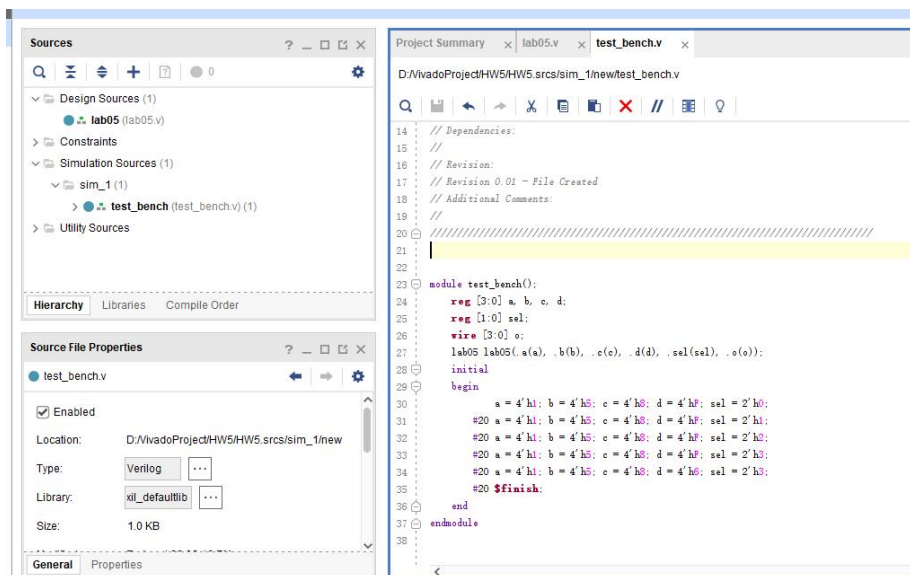
3.3 添加 Verilog 设计文件

最终创建得到一个 4bit 的四选一选择器 module 的 design 文件



3.4 添加仿真文件

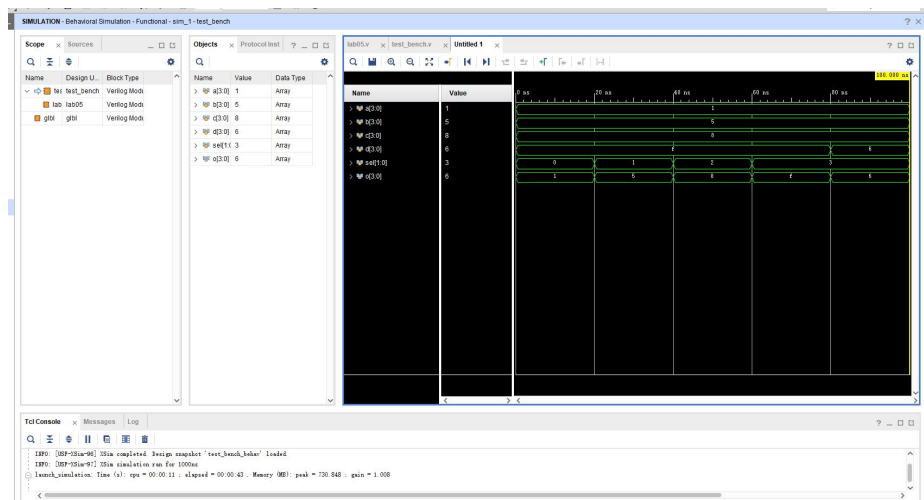
添加结果如下：



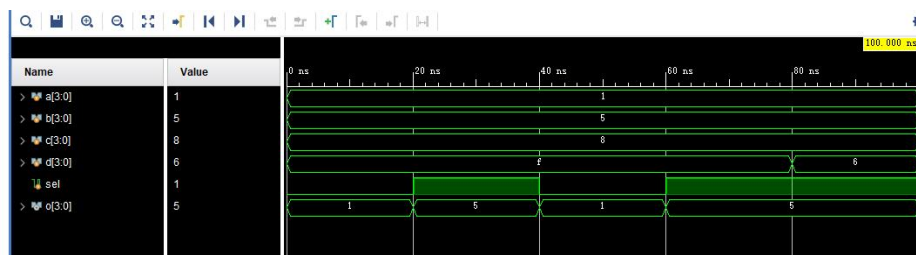
由上述仿真代码可以看出，Verilog 仿真文件与 Verilog 设计文件有些不同。第一，仿真文件不需要输入输出信号，所有的信号都是模块的内部信号。第二，在仿真文件内对被测试模块进行实例化，并对被测试模块构造输入信号。第三，仿真文件只用于仿真，最终不会被综合成电路，会经常用到“initial”等 Verilog 设计文件中不会用到的关键字或语法，这些语法很多是不可综合的。

3.5 波形仿真

仿真结果如下：



将”input [1:0] sel”,改成”input sel” 后,会得到如下奇怪的波形,此时容易查出错误。



3.6 Verilog 仿真文件常用语法

经学习,对原代码注释如下:

```

1  module test_bench2();
2  reg clk,rst_n,clk_a;
3  reg [7:0] r1,r2,r3;
4  integer i;
5  initial clk = 0;
6  always #5 clk = ~clk;
7
8  // $stop的使用
9  initial
10 begin

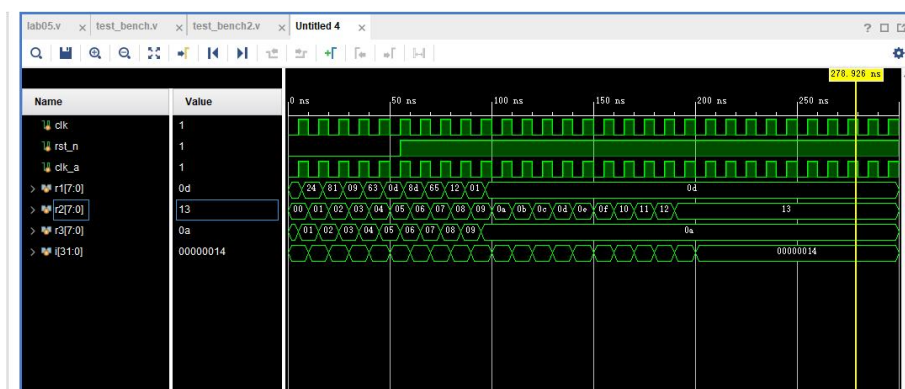
```

```

11         rst_n = 0;
12         #55 rst_n = 1;
13         #245 $stop;
14     end
15
16     // forever是循环控制的一种，表示一直执行
17     initial
18     begin
19         clk_a = 0;
20         forever #5 clk_a = ~clk_a;
21     end
22
23     // repeat是循环控制的一种，括号内是执行次数
24     // $random%256表示在256内产生随机数
25     initial
26     begin
27         r1 = 0;
28         repeat(10)
29         begin
30             @(posedge clk);
31             #2 r1 = $random%256;
32         end
33     end
34
35     // for是循环控制的一种，类似于C语言的for
36     initial
37     begin
38         for(i=0;i<20;i=i+1)
39         begin
40             r2 = i;#10;
41         end
42     end
43
44     // while是循环控制的一种，类似于C语言的while
45     initial
46     begin
47         r3=0;
48         while(r3<10)
49         begin
50             @(posedge clk);
51             r3 = r3 +1;
52         end
53     end
54 endmodule
55

```

仿真结果如下：



其中有些要点：

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。

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。

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。

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。

4 实验练习

4.1 题目 1

代码如下：

```
1 module e1_sim();
2   reg a, b;
3
4   initial
```

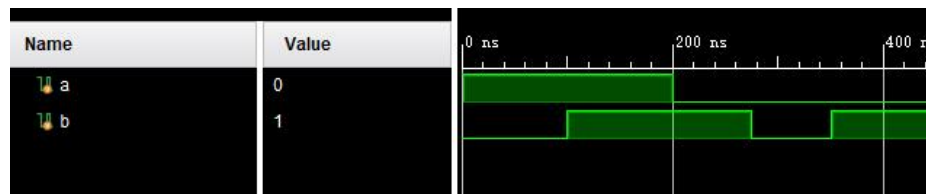
```

5      begin
6          a = 1;
7          #200;
8          a = 0;
9      end

10
11      initial
12      begin
13          b = 0;
14          #100;
15          b = 1;
16          #175;
17          b = 0;
18          #75;
19          b = 1;
20      end
21  endmodule
22

```

仿真结果如下：



4.2 题目 2

代码如下：

```

1  module e2_sim();
2  reg CLK, RST_N, D;
3
4  initial
5  begin
6      CLK = 0;
7      forever #5 CLK = ~CLK;
8  end
9
10 initial
11 begin

```



```

12     RST_N = 0;
13     #27;
14     RST_N = 1;
15 end
16
17 initial
18 begin
19     D = 0;
20     #13;
21     D = 1;
22     #25;
23     D = 0;
24 end
25 endmodule
26

```

仿真结果如下：



4.3 题目 3

仿真文件代码修改如下：

```

1  module e2_sim();
2  reg CLK, RST_N, D;
3
4  // 修改处
5  d_ff_r d_ff_r_init(CLK, RST_N, D);
6
7  initial
8  begin
9  CLK = 0;

```

```

10     forever #5 CLK = ~CLK;
11     end
12
13     initial
14     begin
15         RST_N = 0;
16         #27;
17         RST_N = 1;
18     end
19
20     initial
21     begin
22         D = 0;
23         #13;
24         D = 1;
25         #25;
26         D = 0;
27     end
28 endmodule
29

```

仿真结果如下：



4.4 题目 3

仿真文件代码修改如下：

```

1 module decoder_3_8(
2     input [2:0] in ,
3     output reg [7:0] out
4 );
5
6 always @(*)

```

```

7      begin
8          case(in)
9              3'b000: out = 8'b0000_0001;
10             3'b001: out = 8'b0000_0010;
11             3'b010: out = 8'b0000_0100;
12             3'b011: out = 8'b0000_1000;
13             3'b100: out = 8'b0001_0000;
14             3'b101: out = 8'b0010_0000;
15             3'b110: out = 8'b0100_0000;
16             3'b111: out = 8'b1000_0000;
17             default: out = 8'b0000_0000;
18         endcase
19     end
20
21 endmodule
22

```

3-8 译码器

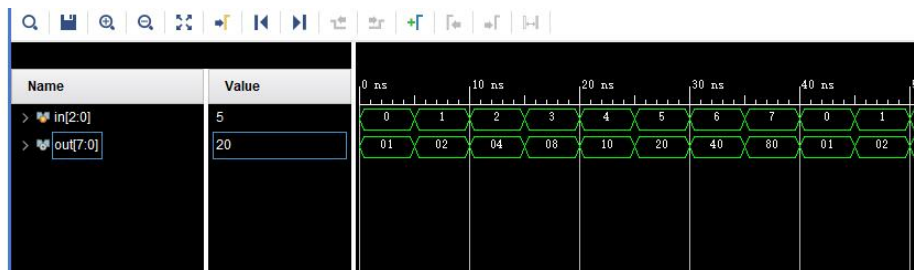
```

1      module decoder_3_8_sim();
2          reg [2:0] in;
3          wire [7:0] out;
4
5          decoder_3_8 decoder_3_8_inst(in, out);
6
7          initial
8          begin
9              in = 3'b000;
10             forever #5 in = in + 1;
11         end
12
13     endmodule
14

```

对应仿真文件

仿真结果如下：



5 总结与思考

5.1 本次实验的收获

在本次实验中，收获较大，学会了使用 Vivado 的仿真。

5.2 评价本次实验的难易程度

本次实验内容难度适中。

5.3 评价本次实验的任务量

本次实验任务量合理。

5.4 为本次实验提供改进建议

建议详述语法方面的问题。