

Vivado安装与使用

② 主讲人: 蔡建

德以明理 学以精Z





目录一

CONTENTS

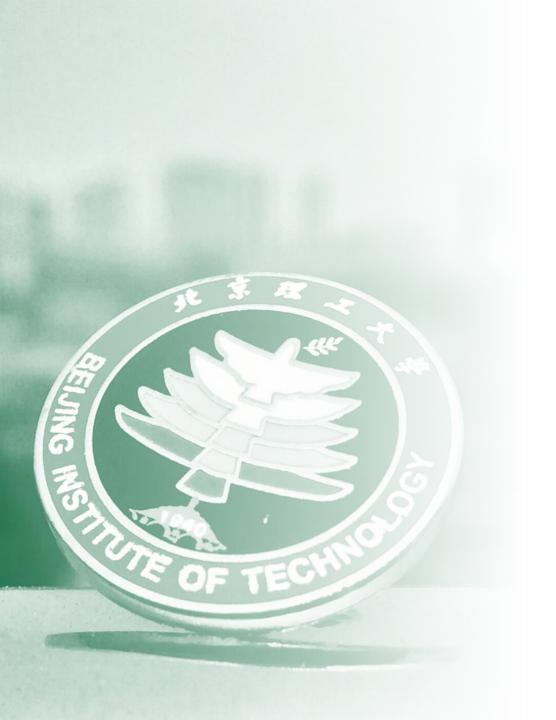
1 Vivado安装

2 Vivado基本功能

3 "三人表决器"设计示例

4 Vivado仿真

5 总结



1 Vivado安装



https://china.xilinx.com/support/download.html

Version

2020.2

2020.1

2019.2

Vivado 存档

ISE 存档

CAE 供应商库

Vivado Design Suite - HLx Editions Update 1 - 2019.2

重要信息

这是一个常见的更新程序。如果您已为 Vivado 运行了此更新程序,则无需为 Vitis 重新运行,反之亦然。

Vivado® Design Suite 2019.2.1 现已推出, 支持:

• 其它 Zynq® UltraScale+™ RFSoC 器件: - (XCZU46DR, XCZU47DR, XCZU48DR, XCZU49DR)

对于使用这些器件的客户, Xilinx 建议您安装 Vivado 2019.2.1。对于其它器件,请继续使用 Vivado 2019.2。

注: 只有 Google Chrome 和 Microsoft Edge 网络浏览器支持下载验证。

下载内容包

Vivado Design Suite HLx Editions (All Editions)

下载类型

Last Updated

答案

2019.x - Vivado 已知问题

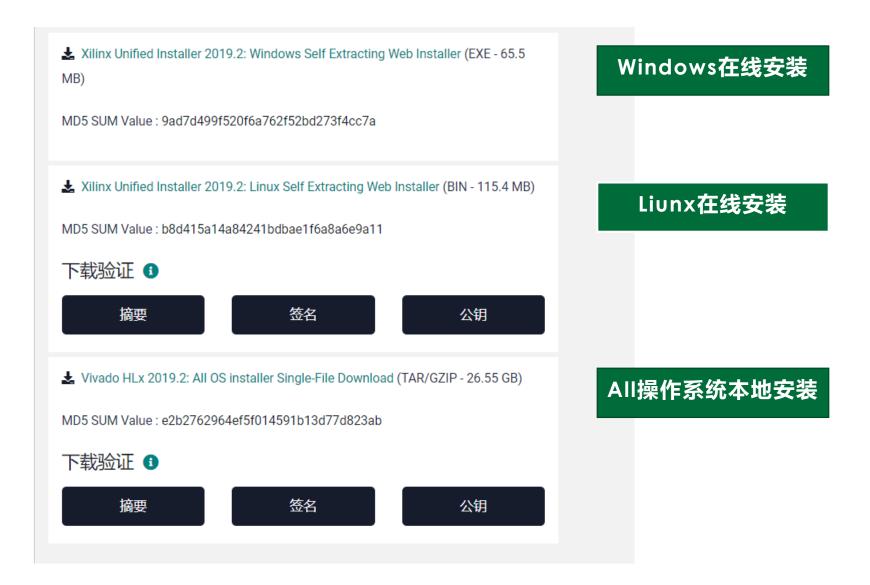
技术文档

发布说明 OS 支持更新

2019-12-11

下载





下载

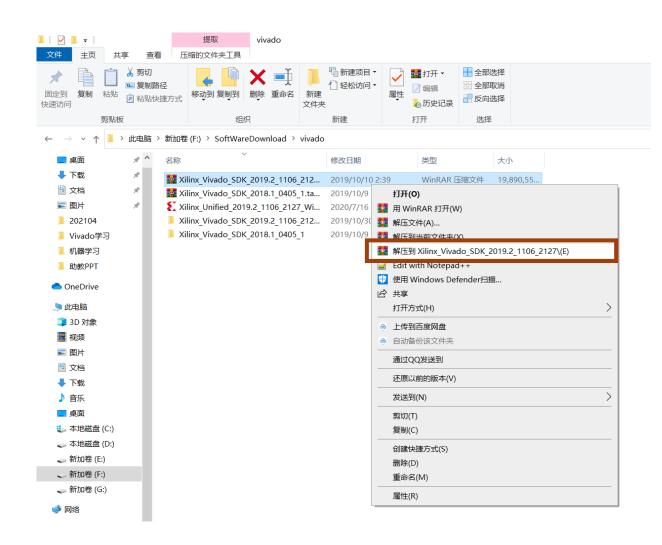


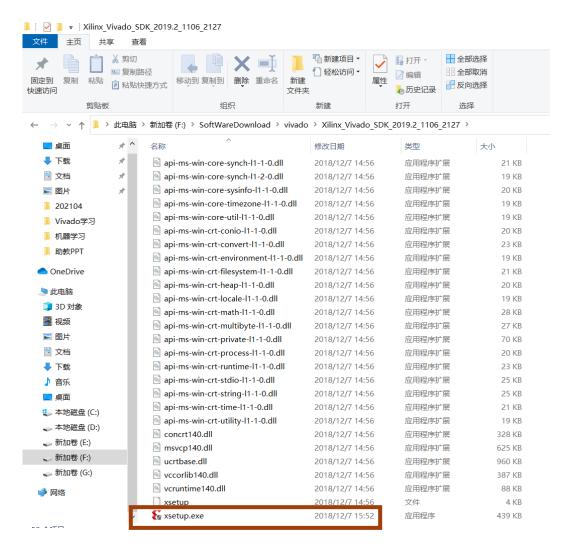
| E XILINX |
|---|
| 登录 |
| 电子邮件地址 |
| I |
| 密码 |
| |
| |
| · 登录 |
| |
| |
| 创建账号 |
| 忘记/重设密码? |
| 重新发送账号激活邮件? |
| 我们正在不断提升网站的安全性。 现使用电子邮件地址登录帐户。 了解更多 |

下载中心 - 姓名与地址验证

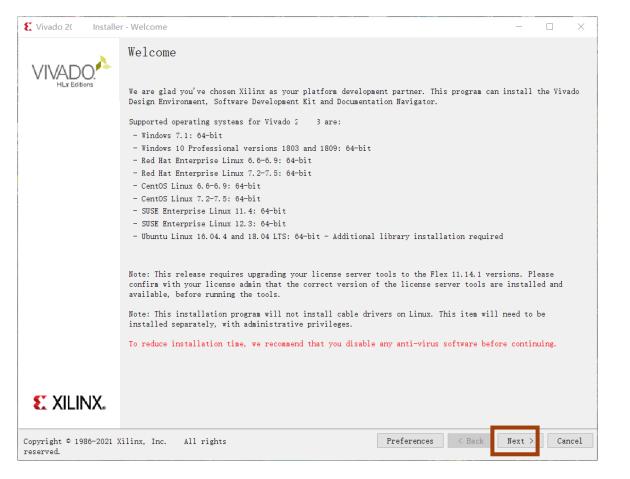
| 美国政府出口批准 • 美国政府出口法律法规规定在满足您的下载需求前须验证您的名、姓、公司名称和收货地址。请提供正确完整的信息。 • 若地址包含带有非罗马字符(如如沉音符、波浪符或冒号)的邮政信箱和姓名,将 无法通过美国出口合规系统。 | | | | | | | |
|--|---------|--|--|--|--|--|--|
| 姓 (中文) | 名 (中文) | | | | | | |
| 名 (英文)* | 姓 (英文)* | | | | | | |
| Business E-mail* | | | | | | | |
| 1293234317@qq.com 公司名称(中文) | | | | | | | |
| 北京理工大学 | | | | | | | |
| 公司名称 (英文)* | | | | | | | |
| Beijing Institute of Technology | | | | | | | |

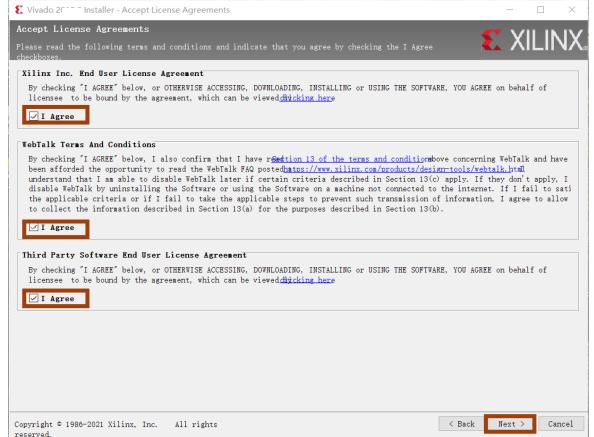




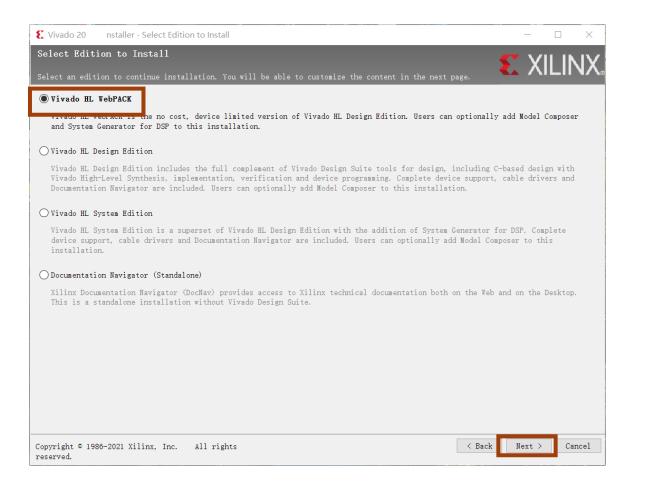


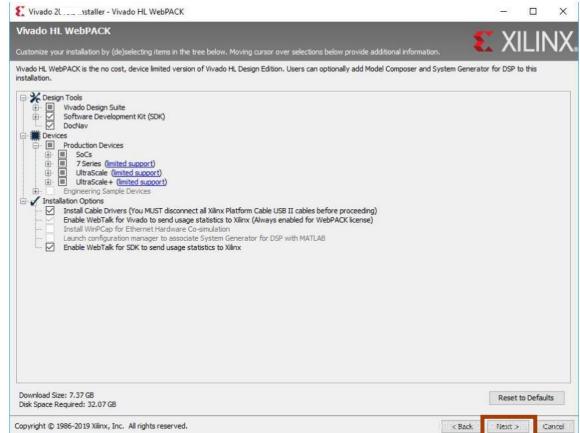




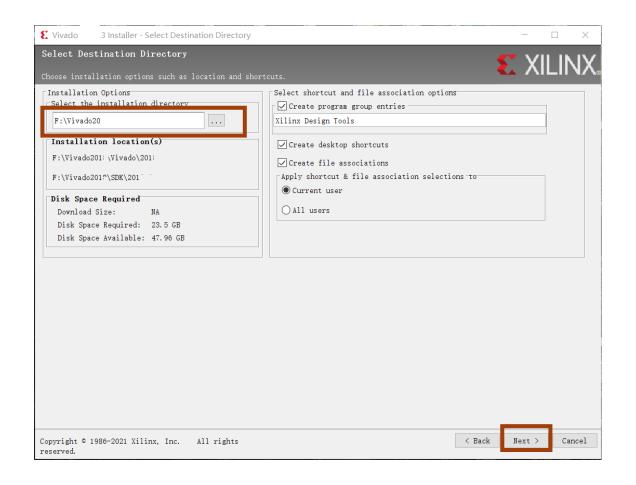


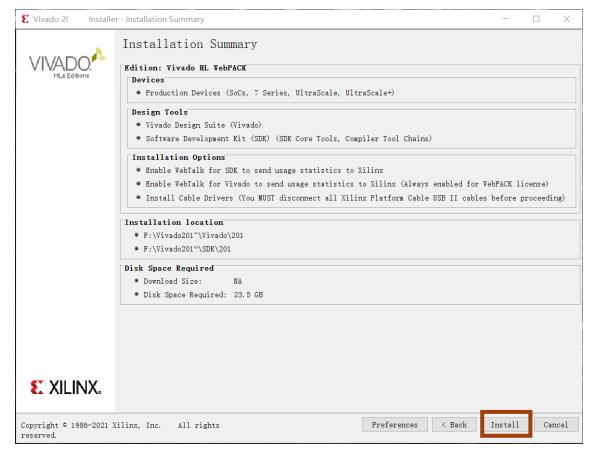




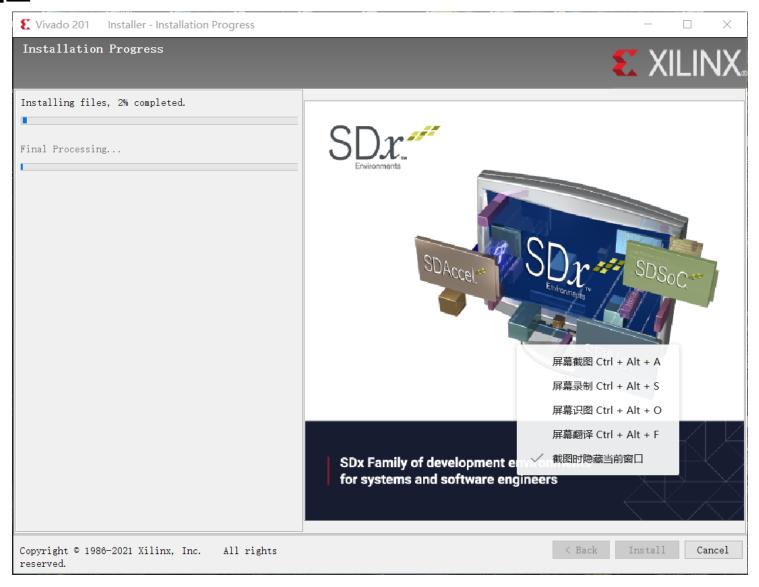










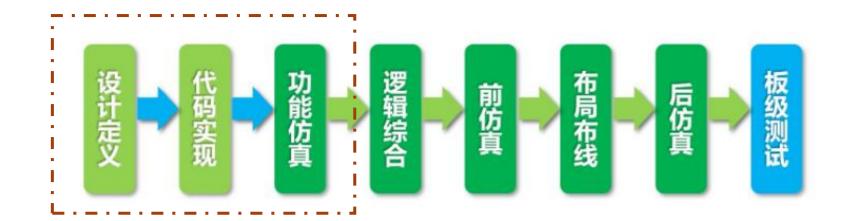




2 Vivado基本功能

硬件设计流程









设计定义

- 芯片器件的选择
- 需要实现什么功能
- 逻辑设计、系统/模块设计
- 方案验证

代码实现

- 将划分好的各功能模块用 硬件描述语言表达出来
- 用Verilog搭硬件电路

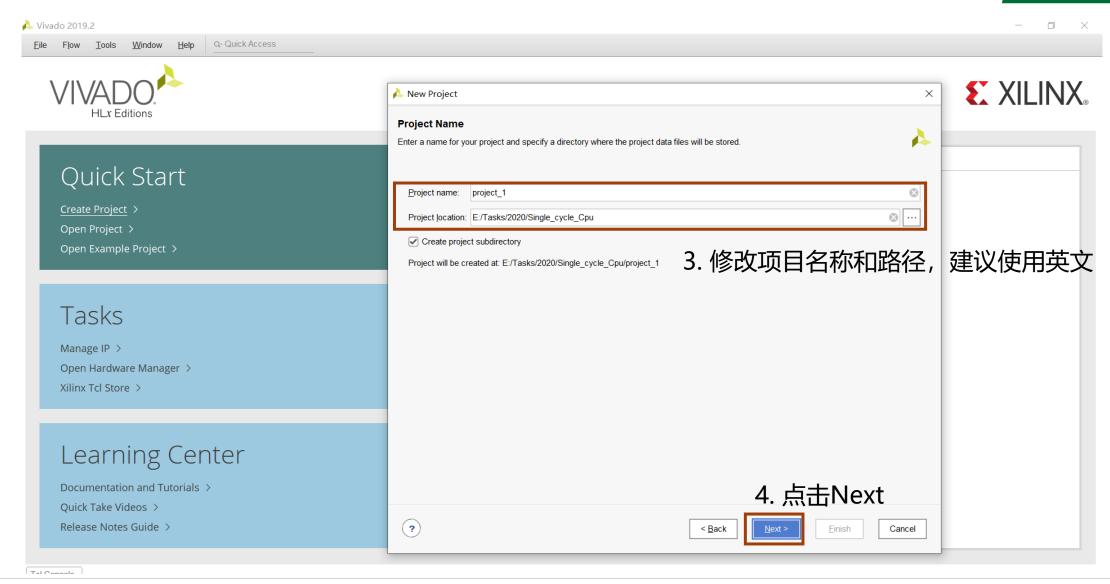
功能仿真

- 即行为仿真,验证你设计 的电路行为是否符合预期
- 对所设计的电路进行逻辑 功能验证



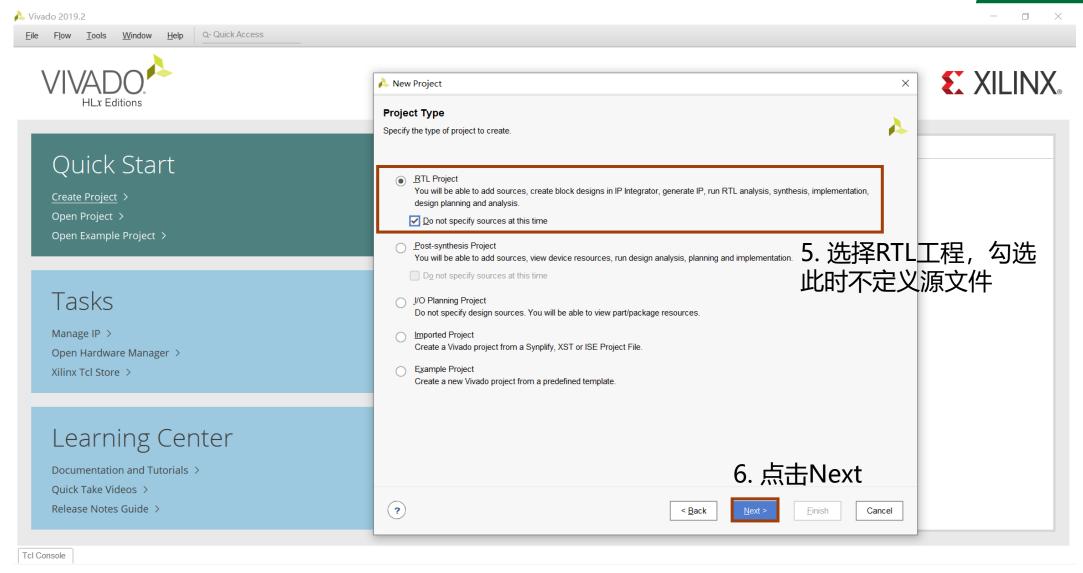




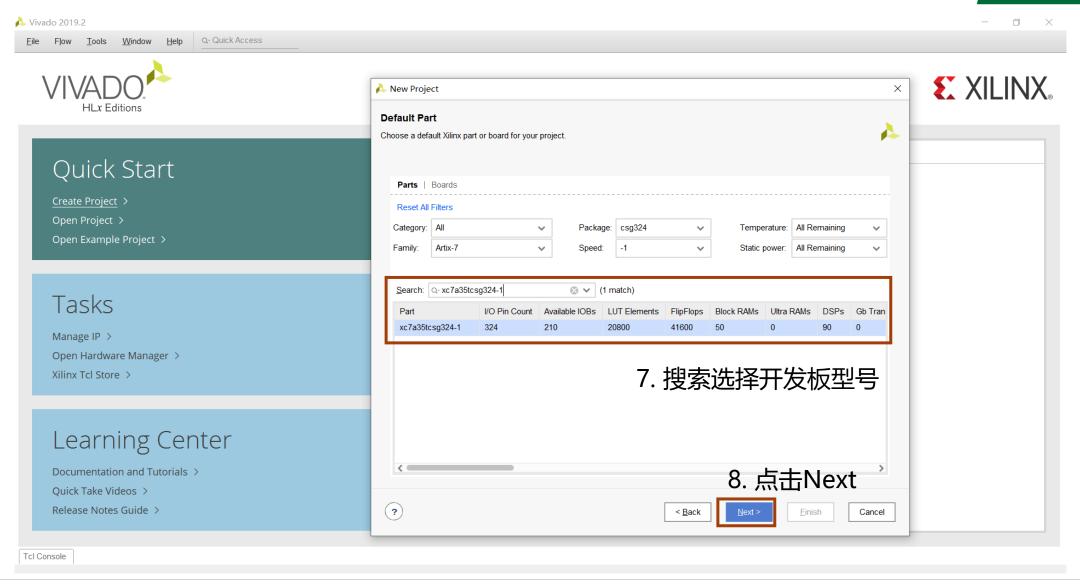




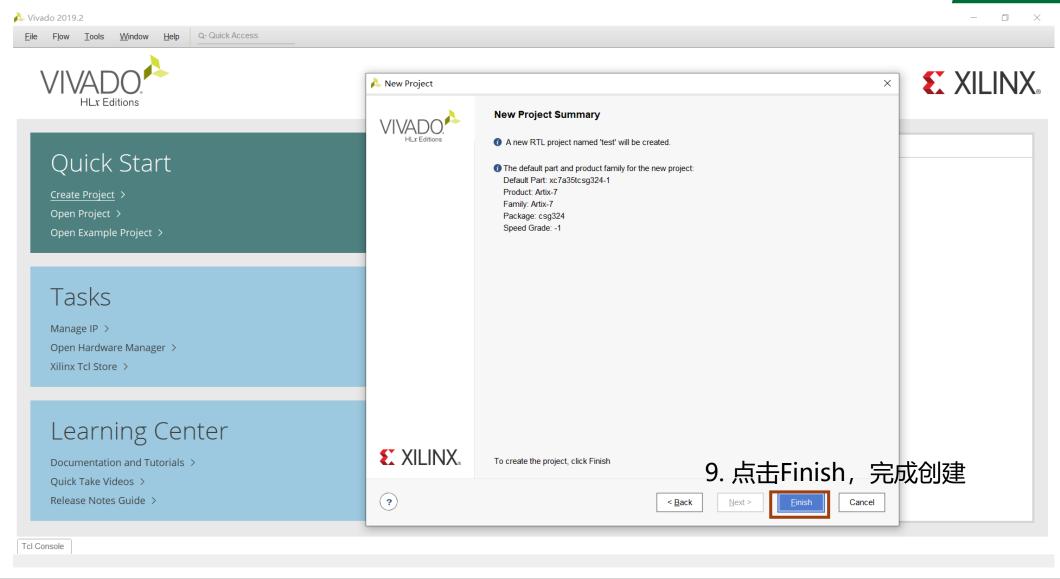






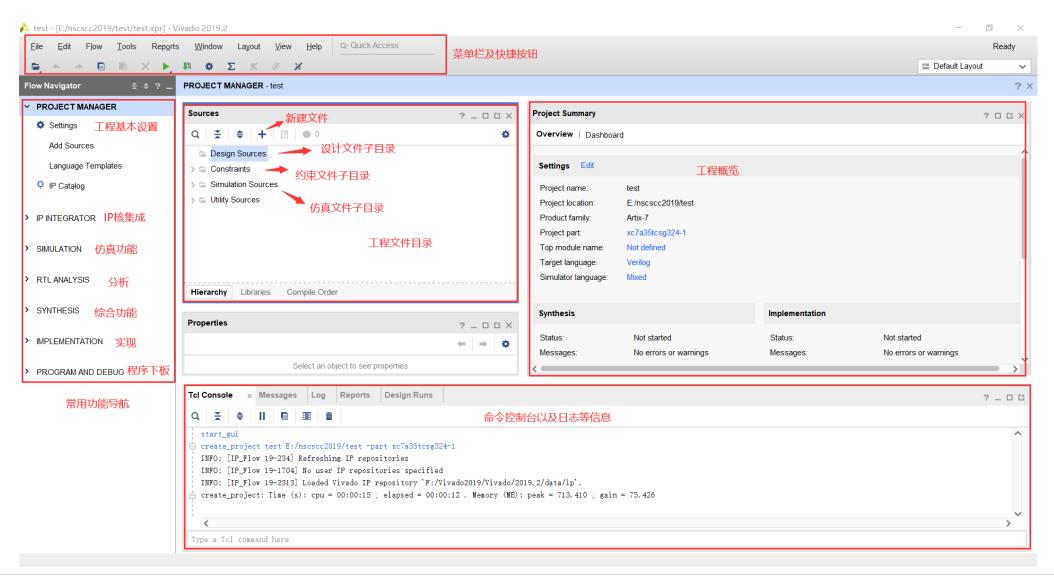




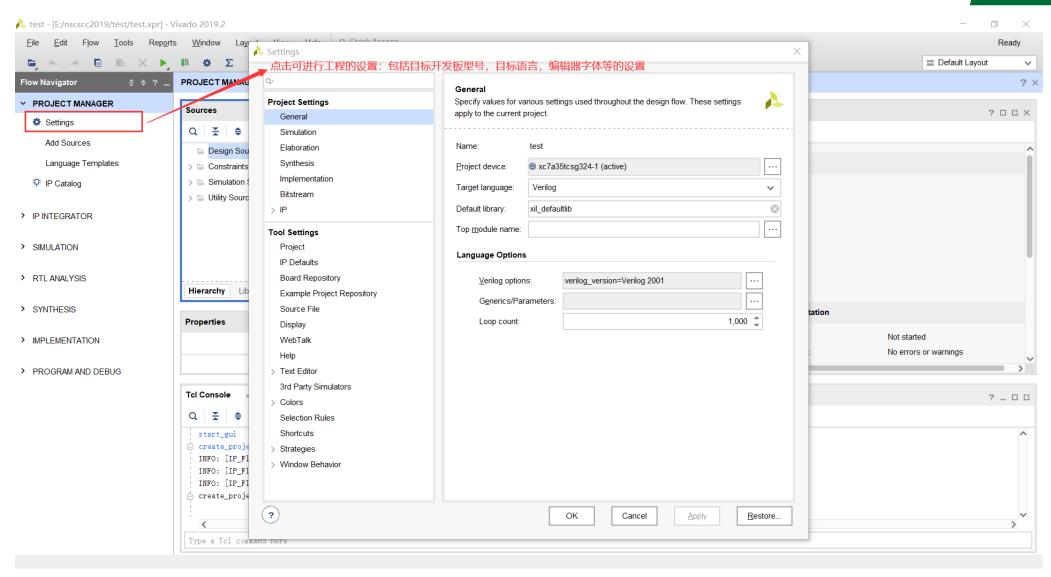




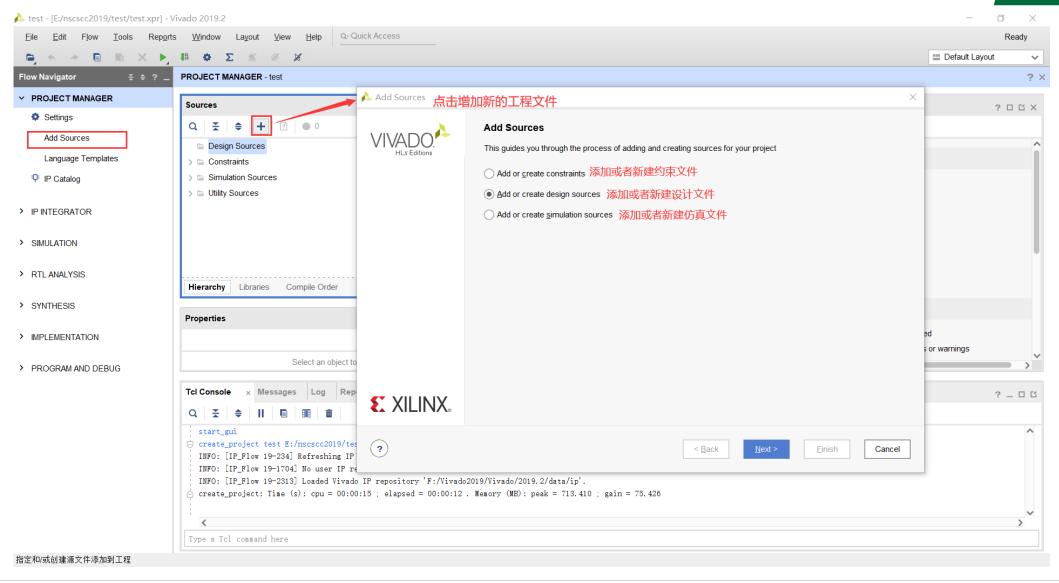














3 "三人表决器"设计示例

三人表决器





问题提出:小A、小B、小C三人需要对某一个提案进行投票表决,每个人可以投支持票或者反对票。三个人参与表决,少数服从多数,若有两人投支持票,则提案通过,否则否决。请设计一个数字组合逻辑电路实现上述三人表决的功能。

逻辑抽象原问题

对于三人的投票选择: 对于投票结果F:

投支持票——输入逻辑1 通过提案——输出逻辑1

对应到开发板上:每个人可以通过拨动开关来投票;通过一个LED灯的亮灭来表示表决的结果。

如何设计中间逻辑电路?

列真值表求解布尔表达式



| Α | В | С | F |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

| C AB | 00 | 01 | 11 | 10 |
|------|----|----|----|----|
| 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 |

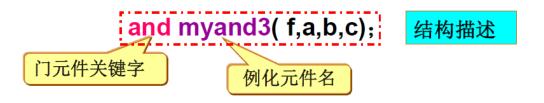
F = AB + BC + AC

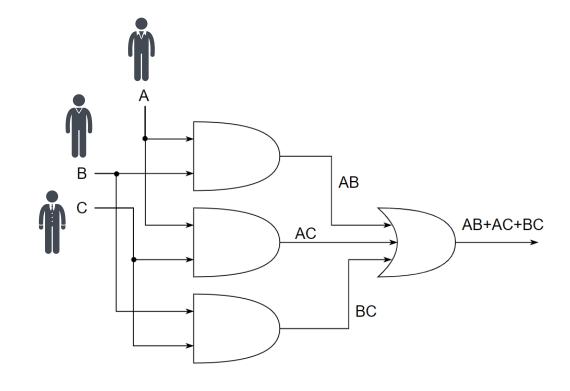
三个与门;两个或门

Verilog结构化描述



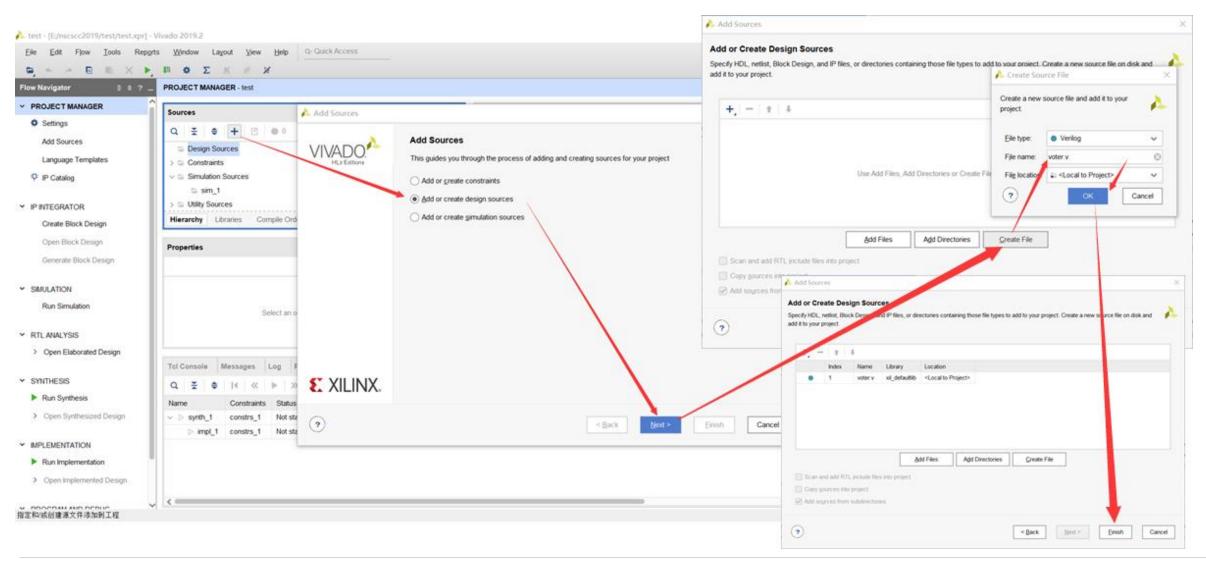
```
≡ voter.v
     `timescale 1ns / 1ps
  2
     module voter(
        // 输入端口声明
        input a,
        input b,
        input c,
        // 输出端口声明
  9
        output f
 10
 11
     wire ab, ac, bc; // 声明中间信号
 13
     // 结构化描述组合电路
     and and1(ab, a, b); // a信号与b信号经过"与元件"后输出信号ab
     and and2(ac, a, c);
 17
     and and3(bc, b, c);
 18
     or or1(f, ab, ac, bc); // ab, ac, bc经过"或元件"后输出最终表决结果
 20
     endmodule
 22
```





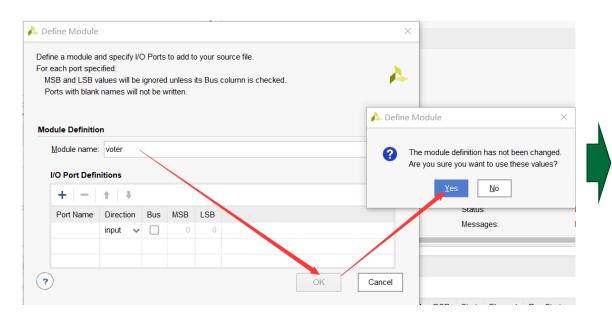
加入新的设计文件

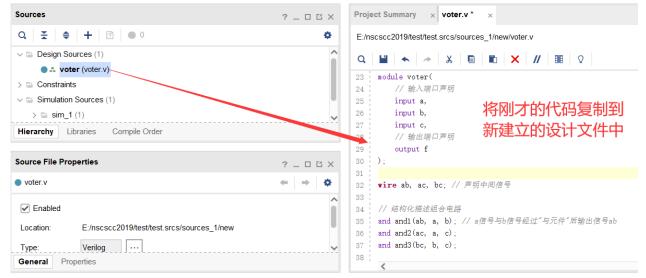




加入新的设计文件









4 Vivado仿真

Testbench



专门为测试特定的模块撰写的测试代码,用于在正式上板前模拟板上的相关激励(或者数据)以达到对测试代码进行效果验证的功能,测试代码的正确性可以通过仿真波形来获得。



输入仿真

- 生成周期性的时钟信号来驱动时序逻辑。
- 生成复位信号使系统中各部分回到初始状态然后同步进行。
- 生成其它输入激励验证模块正确性。

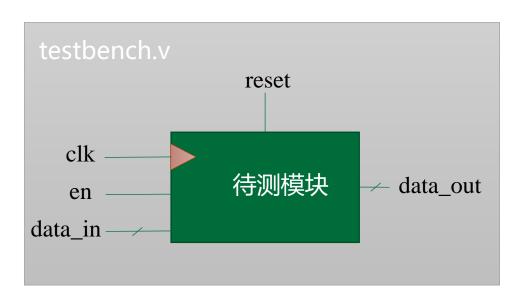


例化待测模块

- 例化带测试模块,传入相应的仿真激励。
- 模拟实际环境的输入激励和输出校验对模块进行 仿真测试。

Testbench要素及模板





testbench需要根据待测模块的输入设计激励

testbench.v 时间单位和时间精度 timescale 1ns / 1ps module testbench(); reg clk; reg rst n; 待测试模块input对应 reg en; 的端口应当申明为reg reg data in; // 定义周期性的时钟信号 parameter CYCLE = 10 initial clk = 1'b0; // 定义仿真刚开始时的时钟信号值 always #(CYCLE) clk = ~clk // 反转每隔CYCLE ns的时间时钟信号发生一次翻转 ... // 待续

Testbench要素及模板



testbench.v 续

```
// 定义复位信号
initial begin
    rst_n=1'b0;
#20
    rst_n=1'b1; // 仿真开始20ns后发生复位
end

// 定义使能信号
initial begin
    #20
    en = 1'b1; // 仿真开始20ns后使能信号有效
#1000
    en = 1'b0; // 仿真开始1020ns后使能信号无效
end
```

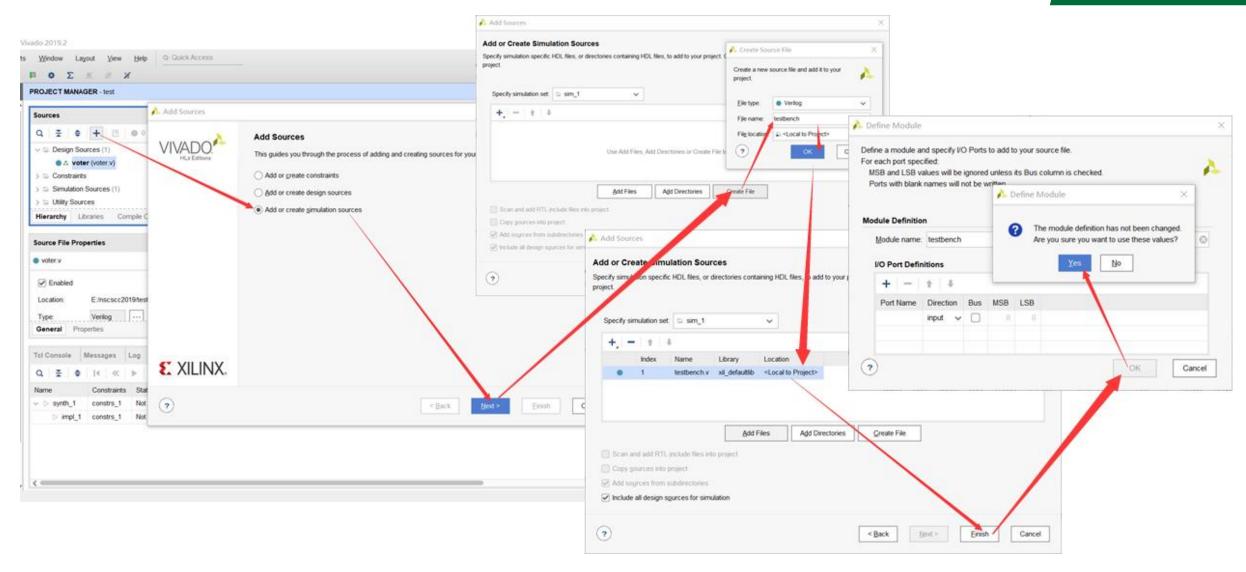
initial、always 语句块是并行触 发的。

initial、#语句不可综合,但是方便仿真。

endmodule

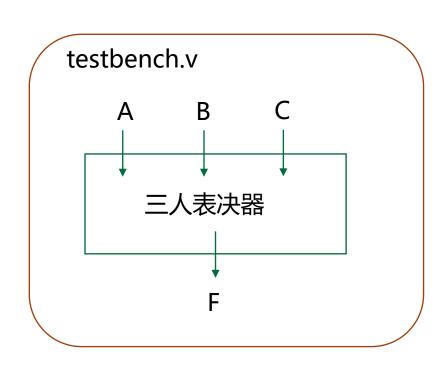
新建仿真文件





为 "三人表决器"编写testbench



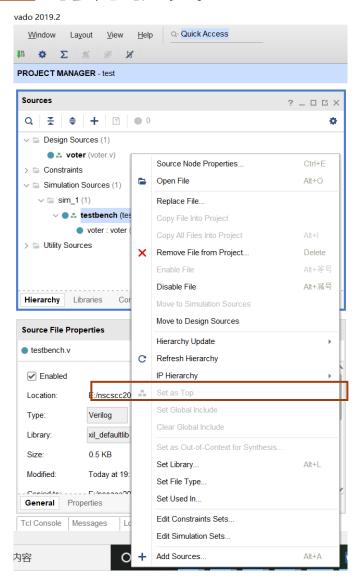


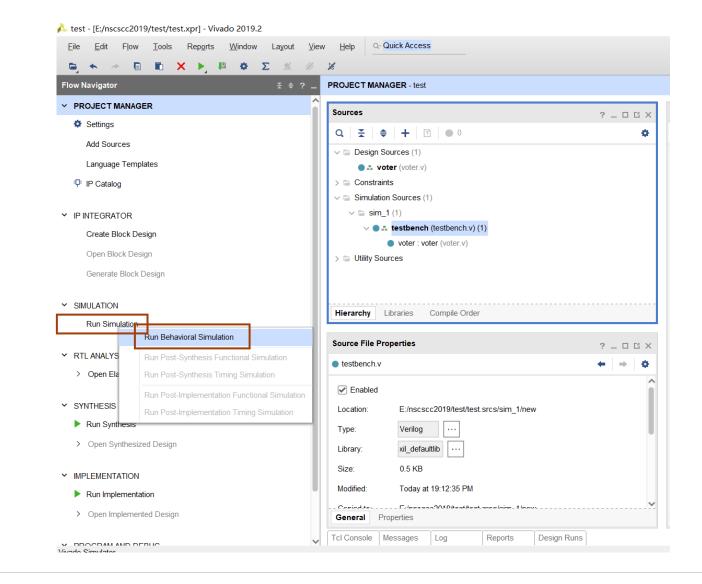
确立缺失的激励:没有时序逻辑,无需时钟信号与复位信号。只需建立三人的选择信号。

```
testbench.v
                                                                                                                           _ o a x
                                         E:/nscscc2019/test/test.srcs/sim_1/new/testbench.v
voter.v
                                                ★ → ¾ ■ ■ X // ■ ♀
E:/nscscc2019/test/test.srcs/sources_1/nev
                                                  timescale 1ns / 1ps
                                                 module testbench(
       timescale 1ns / 1ps
                                                    reg A, B, C;
                                                    initial begin
     module voter(
                                                      A = 1'b0:
                                                       #10 A = 1'b1:
          // 输入端口声明
                                                       #20 A = 1'b0;
          input a.
                                                       #30 A = 1'b1
         input b,
          input c,
                                                    initial begin
         // 输出端口声明
                                         15
                                                       B = 1'b0:
          output f
                                                       #5 B = 1'b1:
                                                       #10 B = 1'b0
10 );
                                                       #25 B = 1'b1;
     wire ab, ac, bc; // 声明中间信号
                                                    initial begin
                                                      C = 1'b1;
      // 结构化描述组合电路
                                                       #15 C = 1'b0;
     and and1(ab, a, b); // a信号与b
                                                       #20 C = 1'b1;
     and and2(ac, a, c):
     and and3(bc, b, c):
                                                    wire F:
                                                    voter voter
19 or or1(f, ab, ac, bc); // ab,
                                                       .c(C).
      endmodule
                                                       .f(F)
22
                                        34
                                         35
                                                 endmodule
                                         36
```

行为仿真



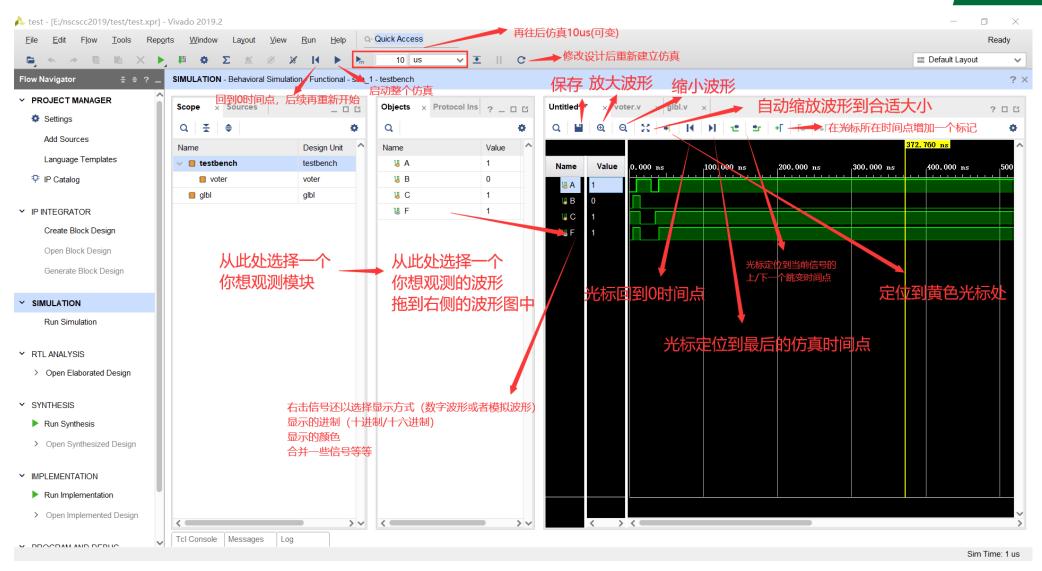






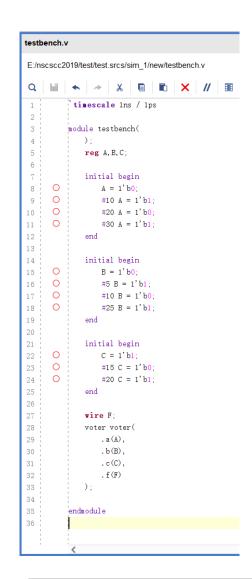
仿真界面

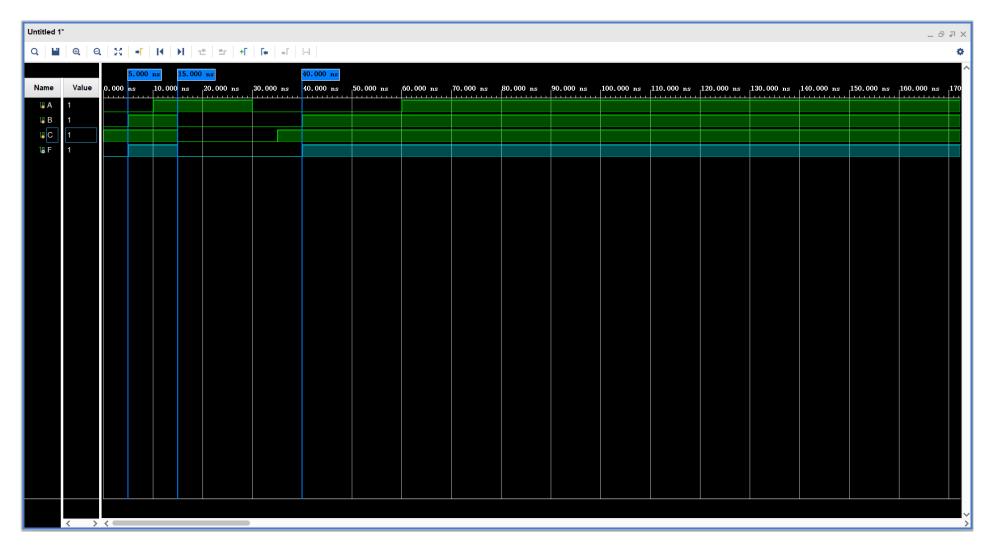




分析仿真波形

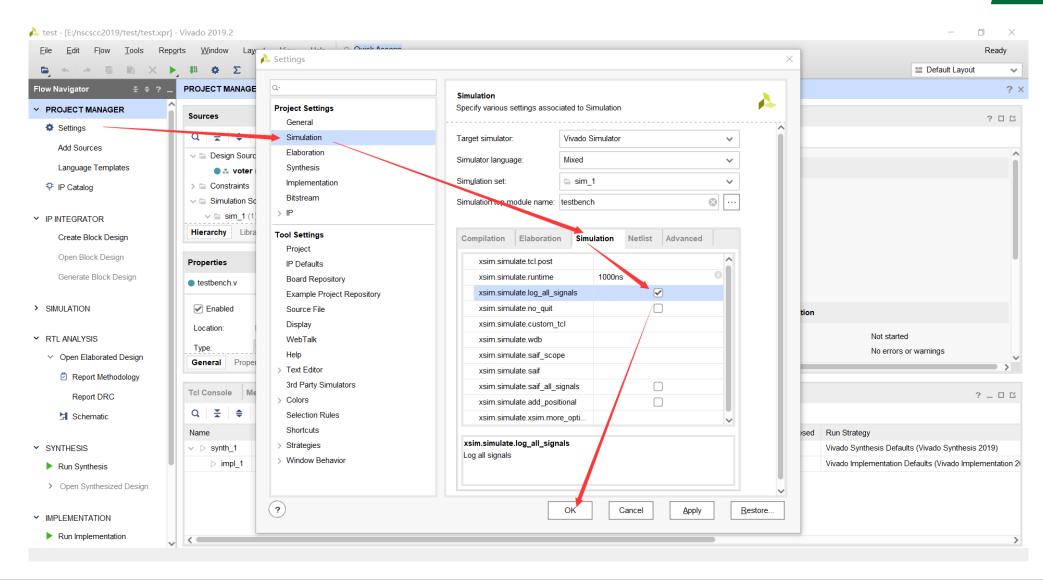






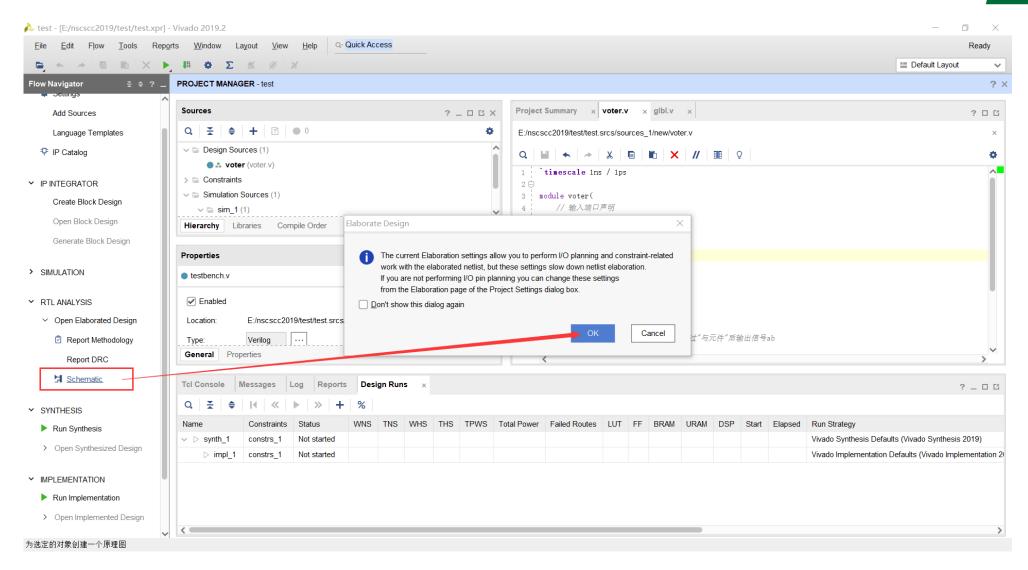
Log所有信号





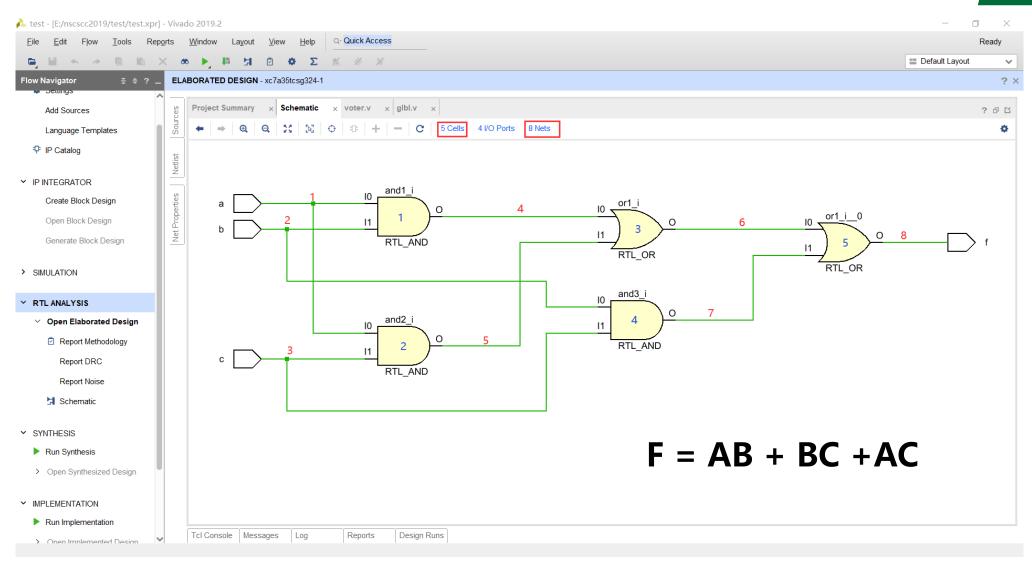
查看原理图 (Schematic)





查看原理图 (Schematic)







5 总结

Tips

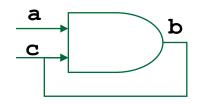


- ◆ 除testbench外设计文件不要使用#、initial等不可综合语句(仅仿真不上板随意)
- ◆ 注意仿真时 Z 表示对应连线浮空(缺少连接), X 则表示当前信号值不确定,往往是由于 Z 造成的。
- ◆ 同学们可以自行学习一些Verilog系统函数\$readmemh、\$fscanf、\$display、

\$finish等来帮助更好地完成仿真。



- □ 仿真波形跑到一半不动了: 检查设计中是否存在逻辑回环。
- □ 出现x不确定的信号值:检查是否多驱动,是否有浮空的连线。
- ◆ 良好的代码书写习惯
 - □ 命名规律
 - □ 适当缩进
 - □信号对齐

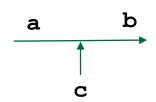


```
module wrong(b)
  output b;

wire a,c;

assign b = a&c;
  assign c = b;

endmodule
```



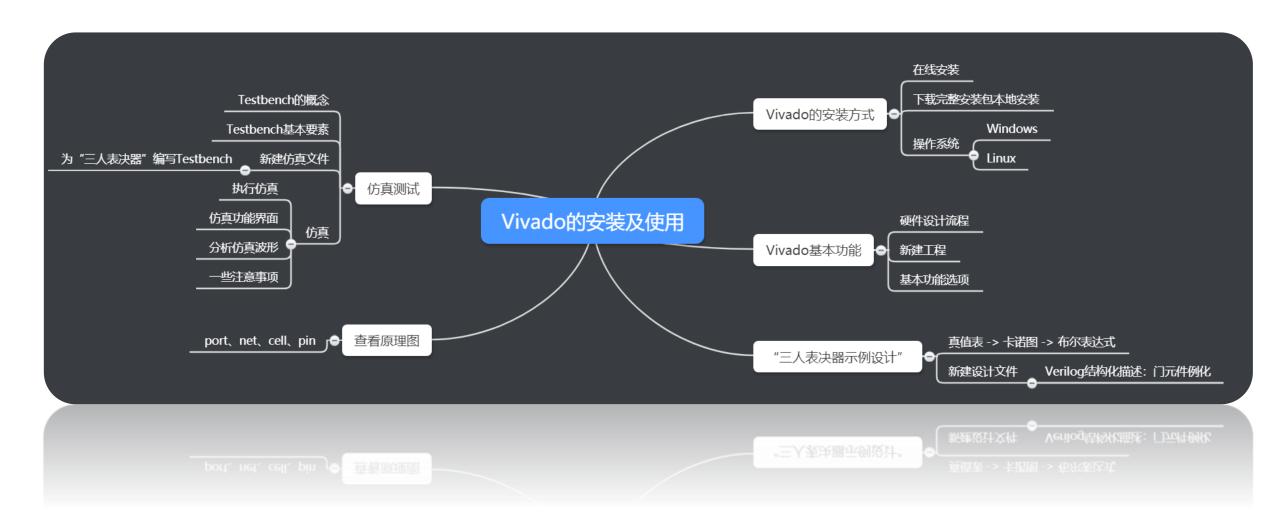
```
module wrong(b)
  output b;

wire a,c;

assign b = a;
  assign b = c;
endmodule
```

小结







感谢各位

② 主讲人: 蔡建

山 计算机学院

德以明理 学以特Z