实验一 基本门电路——Vivado 开发环境及 Nexys4 开发板的熟悉

实验目的

熟悉 Vivado 开发环境，掌握设计流程，包括如何用 Verilog 语言设计电路、仿真、综合、实现与下载。

实验内容

1. 拨码开关与 LED 灯

利用 Verilog HDL 语言，在 Vivado 中创建简单的 16 位拨码开关的输入和 16 位 LED 灯的输出电路,对设计进行仿真验证，并将设计下载 N4 开发板。

2. 门电路设计

使用 Verilog HDL 语言的数据流描述方法设计一个数据宽度可在 1~32 之间变化的 2 输入与门 andgate（或门、非门、与非门等基本门电路），利用仿真来验证设计。并将该与门封装成可变数据宽度参数的 IP 核。

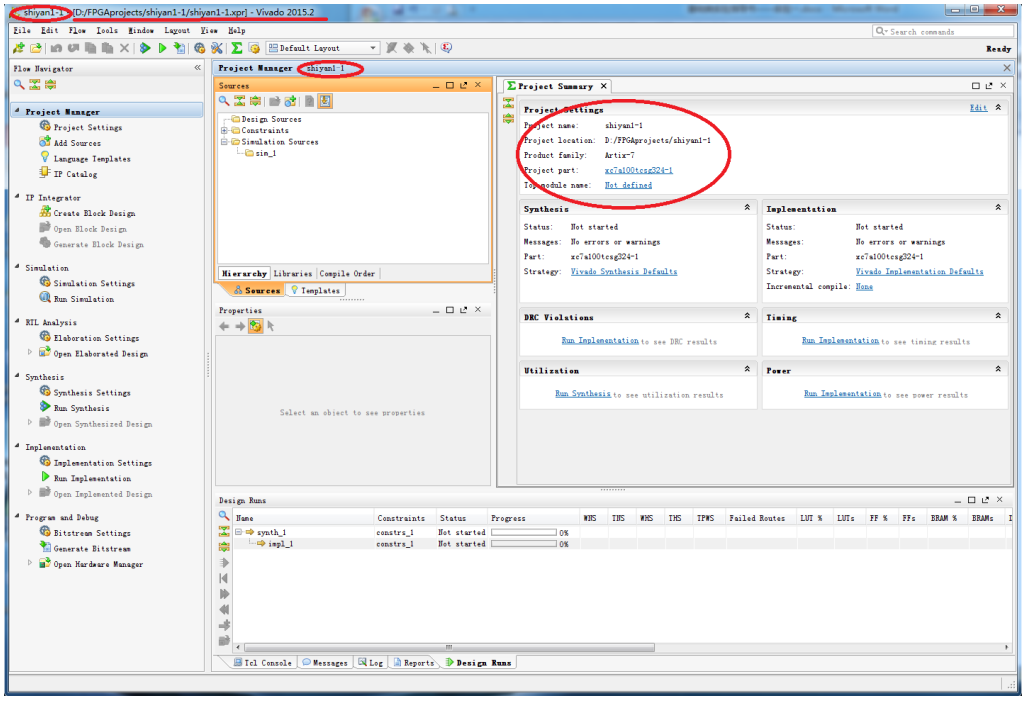
与门真值表如下：

输入		输出
a	*b*	*c*
0	0	0
0	1	0
1	0	0
1	1	1

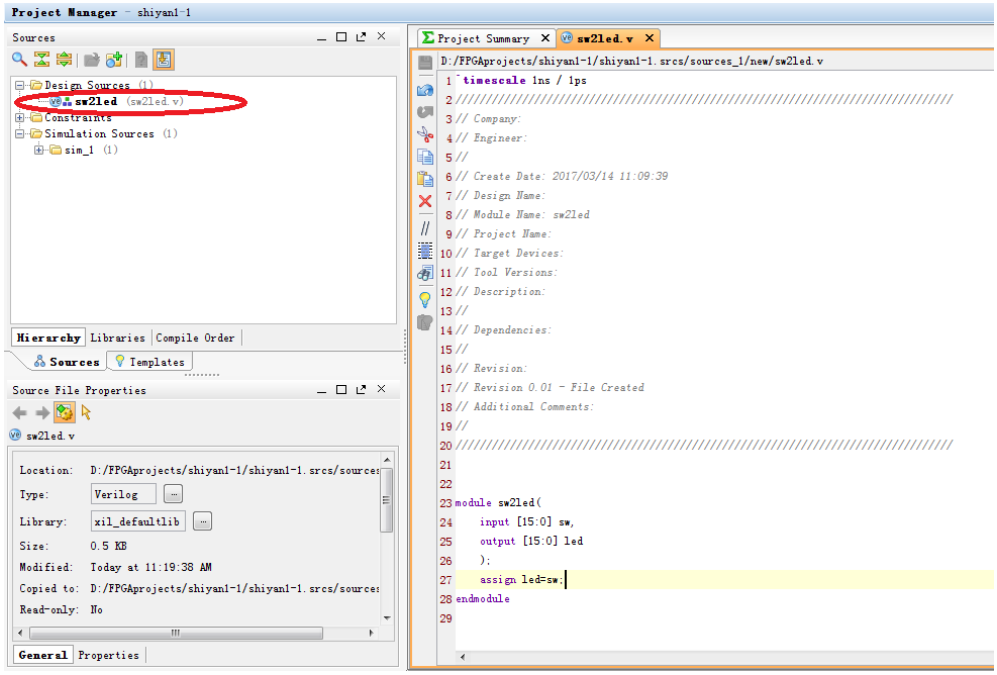
实验过程

一、拨码开关与 LED 灯

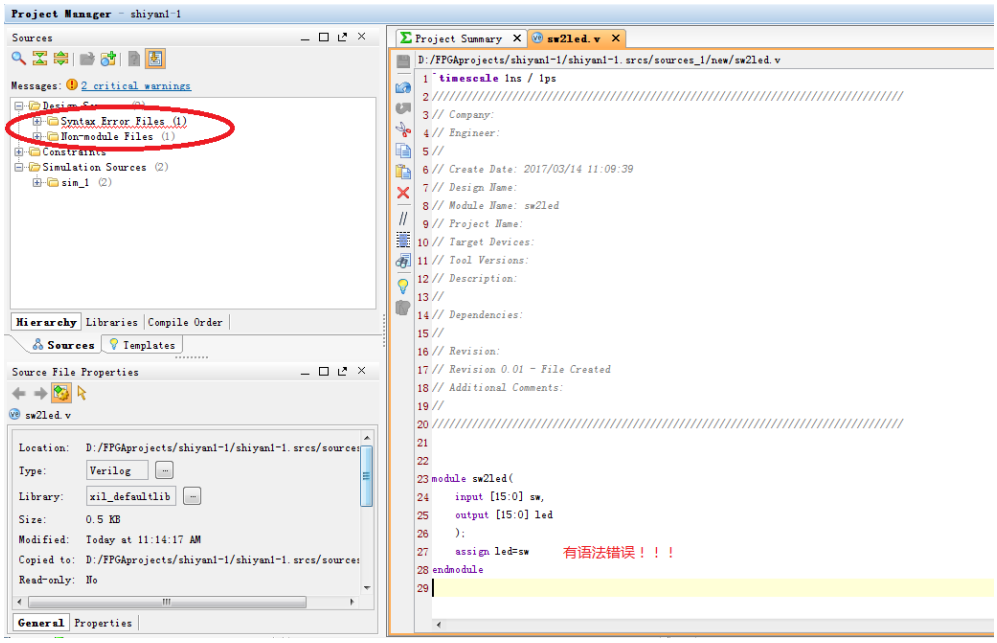
1、打开 vivado 开发环境，新建工程 shiyan1-1，得到空白的 Vivado 工程界面（注意：工程名称、路径不能含中文或特殊字符，N4 板卡型号为 XC7A100TCSG324-1）。



2、新建设计文件(sw2led.v)，并输入设计代码，实现 16 位拨码开关的输入和 16 位 LED 灯的输出电路模块。

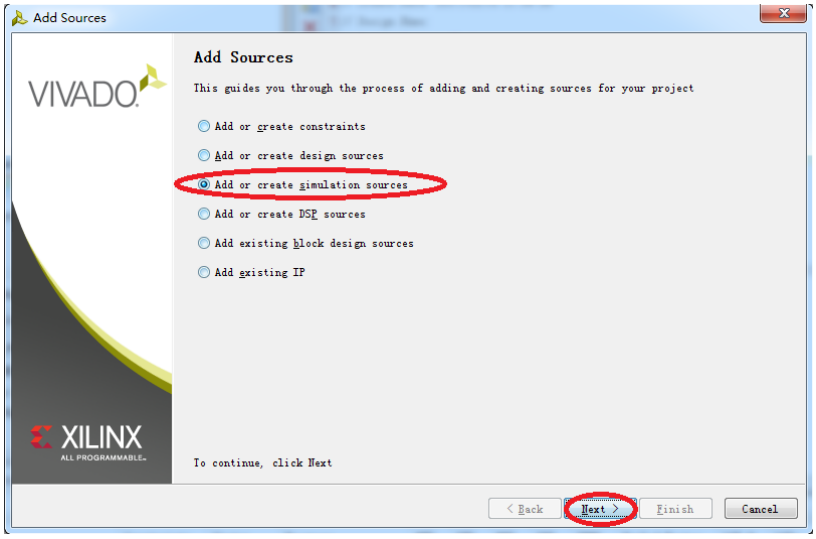


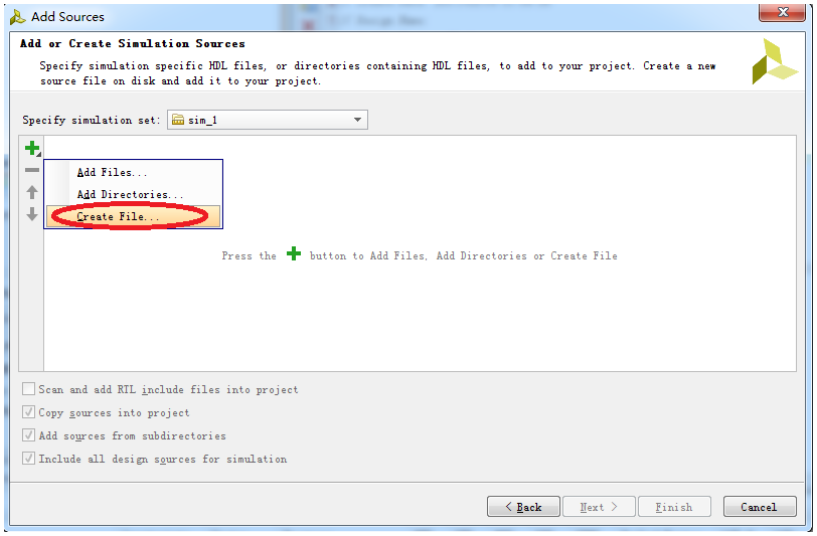
注意：当设计文件有语法错误的时候点击保存后，工程管理窗口内会出现语法错误的文件夹如下图，应仔细检查语法错误！！！



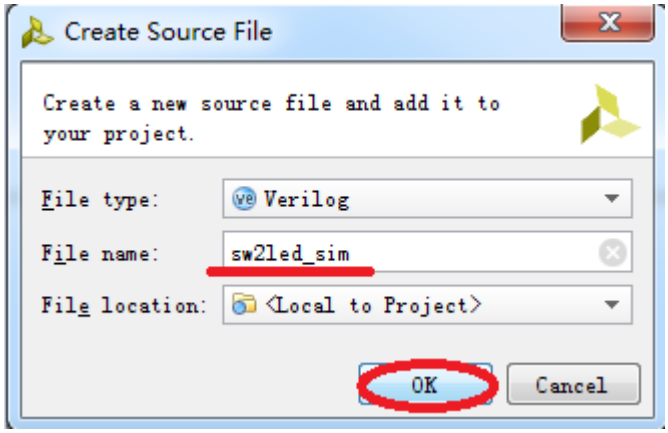
3、编写仿真文件(sw2led_sim.v)，验证电路设计的正确性。如不仿真验证，则可直接添加约束文件综合、实现、生成 BIT 流下载到开发板。

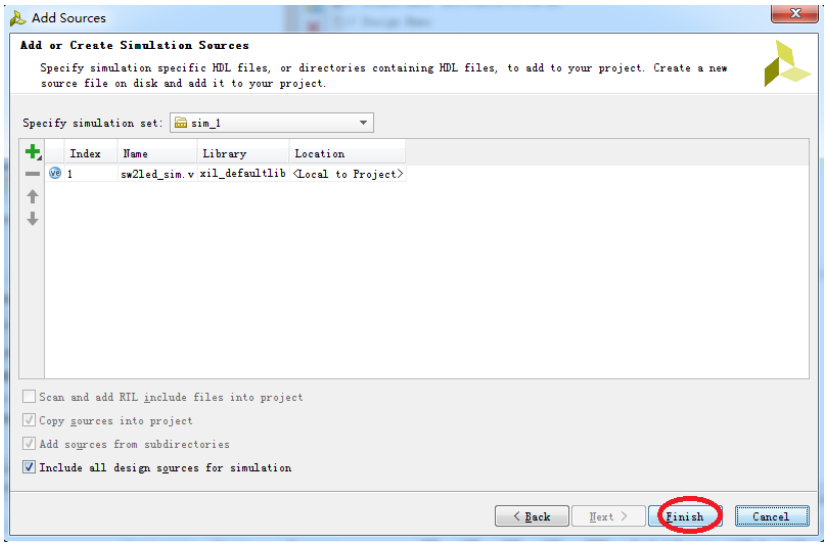
1） 添加仿真文件：



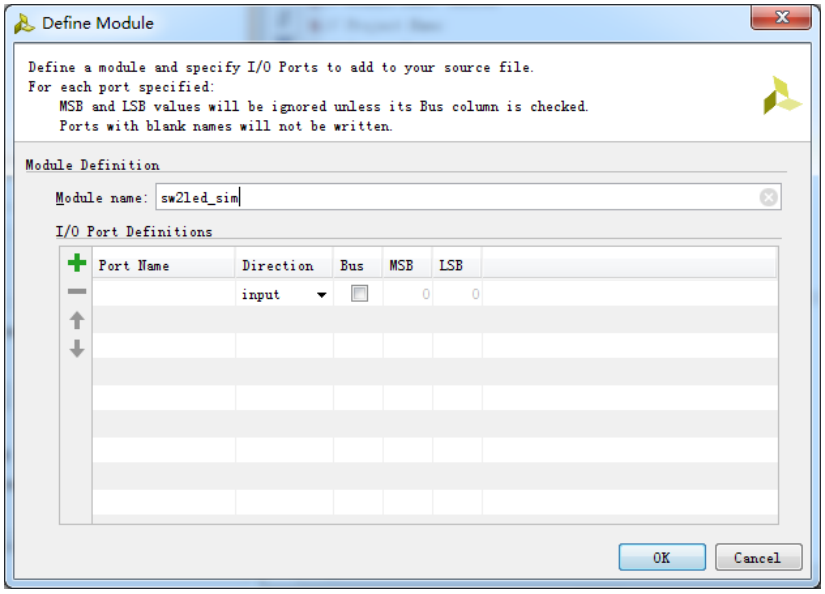


2） 输入仿真文件名称：

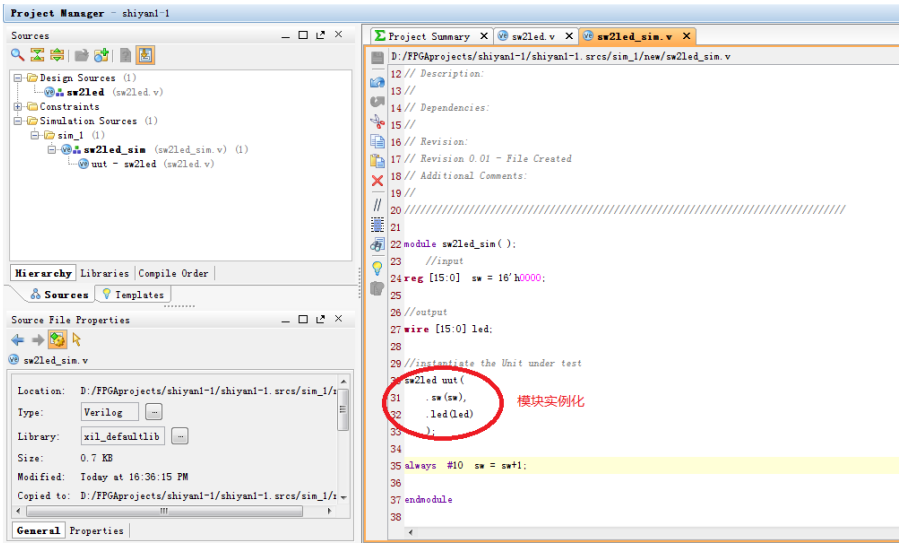




3） 仿真模块不需要定义输入输出端口：

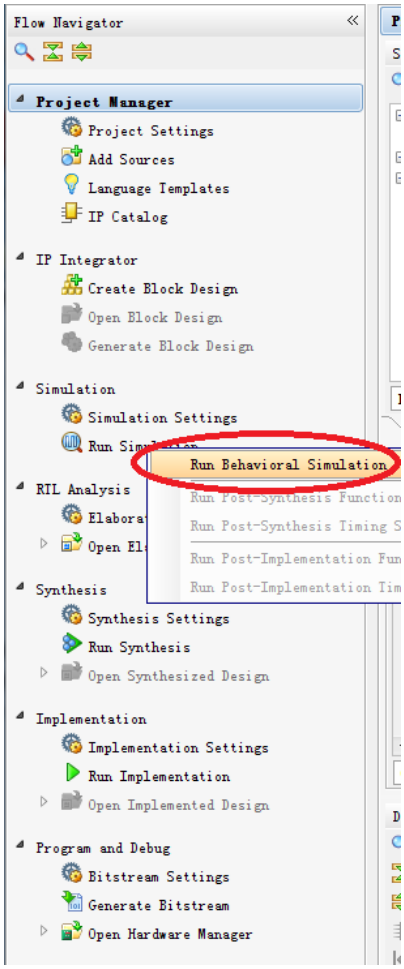


4） 编写仿真代码：

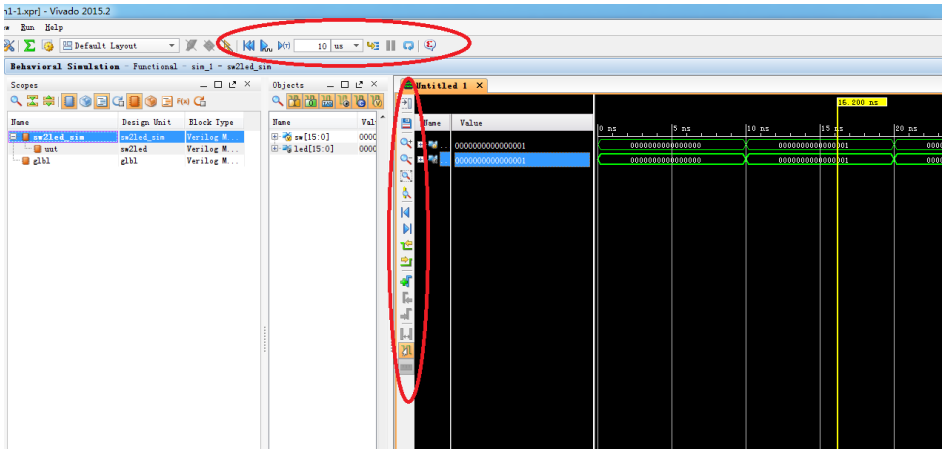


仿真文件中，不需要定义输入输出端口，但一定要对仿真的模块进行实例化（本次实验是对 sw2led 模块进行实例化），因此，要定义传给实例化模块的输入输出变量，并给出输入变量的初始值和变化值，以便观察输出值。

5） 在导航栏里点击仿真（行为仿真）：



6） 观看仿真结果(红框卷住的按钮可以对仿真视图进行调整)：



由于我们设计的电路就是将 sw 的值传给 led，因此，可以看到，led[15:0]的值始终等于 sw[15:0]的值，根据仿真文件的代码可以看到 sw 的值每隔 10ns 增加 1，led 的值也是每隔 10ns 增加 1。

4、综合、实现、添加约束文件。

根据 N4 开发板的设备管脚连接图分配输入输出端口管脚：

信号	管脚	信号	管脚
led[15]	V11	sw[15]	V10
led[14]	V12	sw [14]	U11
led[13]	V14	sw [13]	U12
led[12]	V15	sw [12]	H6
led[11]	T16	sw [11]	T13
led[10]	U14	sw [10]	R16
led[9]	T15	sw [9]	U8
led[8]	V16	sw [8]	T8
led[7]	U16	sw [7]	R13
led[6]	U17	sw [6]	U18
led[5]	V17	sw [5]	T18
led[4]	R18	sw [4]	R17
led[3]	N14	sw [3]	R15
led[2]	J13	sw [2]	M13
led[1]	K15	sw [1]	L16
led[0]	H17	sw [0]	J15

5、生成 bit 流，下载到开发板，观察开发板运行情况。

二、门电路设计与 IP 核封装

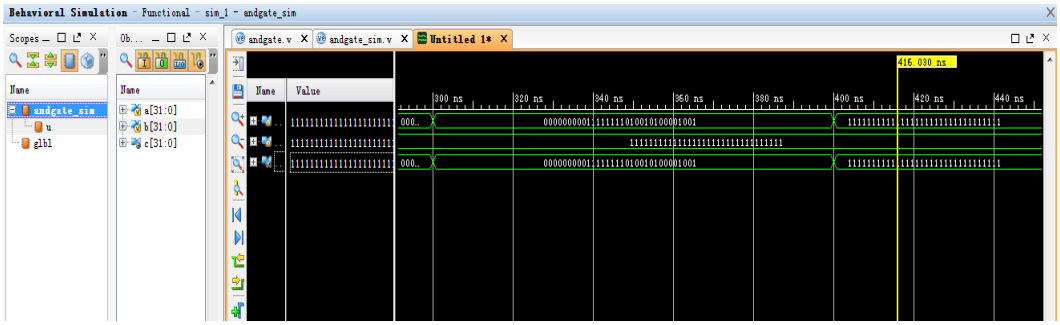
1、新建工程 shiyan1-2，在 shiyan1-2 中新建设计文件(andgate.v)，并输入设计代码，实现一个数据宽度可在 1~32 之间变化的 2 输入与门的电路模块。

```
module andgate
#(parameter WIDTH=8)          //指定数据宽度参数，缺省值是8
   (
    input [(WIDTH-1):0] a,      // 出具位数由参数WIDTH决定
    input [(WIDTH-1):0] b,
    output [(WIDTH-1):0] c
    );

    assign c = a & b;       //  2输入与门
endmodule
```

2、编写仿真文件，并进行仿真。

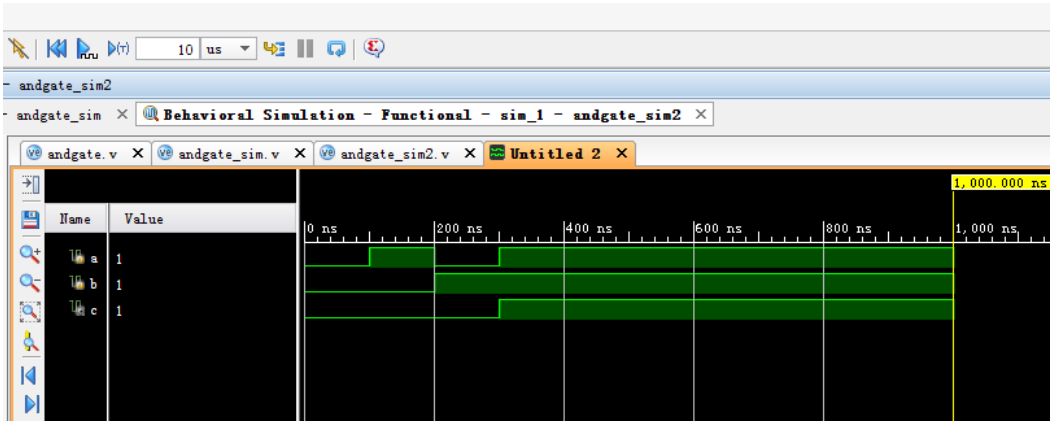
a) 仿真 32 位的 2 输入与门(andgate_sim.v)，情况如下：

```
module andgate_sim(

    );
    // input
    reg [31:0] a=32'h00000000;
    reg [31:0] b=32'h00000000;

    //outbut
    wire [31:0] c;
    andgate #(32) u(a,b,c);     // 实例化与门的时候，设定宽度为32

    initial begin
    #100   a=32'hffffffff;
    #100   begin a=32'h00000000;b=32'hffffffff;end
    #100   a = 32'h007fa509;
    #100   a=32'hffffffff;
   end
endmodule
```



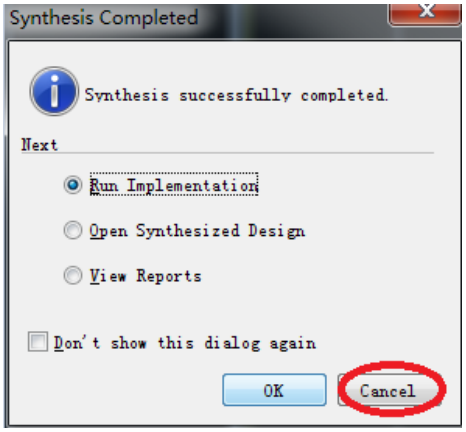
b）仿真 1 位的 2 输入与门(andgate_sim2.v)，情况如下：

```
module andgate_sim2( );
// input
reg a=0;
reg b=0;
//outbut
wire c;
andgate #(1) u(a,b,c); // 实例化与门的时候，设定宽度为1
initial begin
#100 a=1;
#100 begin a=0;b=1;end
#100 a=1;
end
endmodule
```

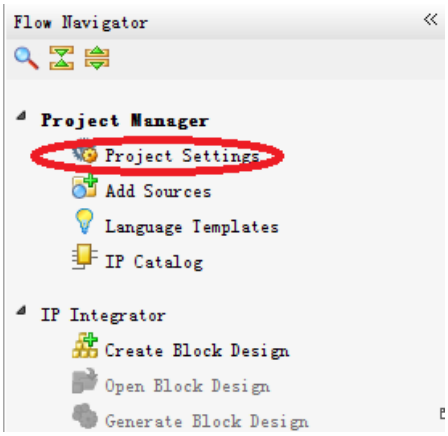


3、验证无误后，可以封装 IP 核：

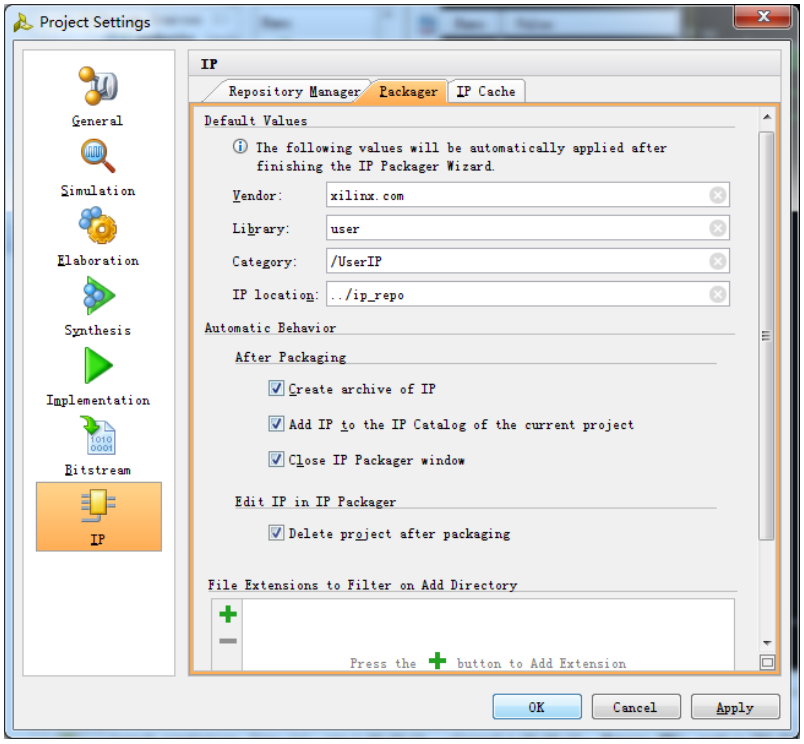
1）对仿真正确的 andgate 模块进行综合，综合结束后在出现的对话框中选择 Cancel。



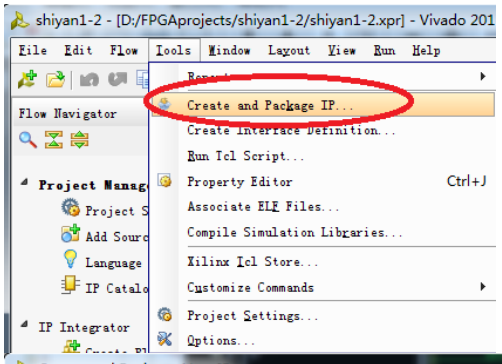
2）在 Flow Navigator 中选择 Project Settings。

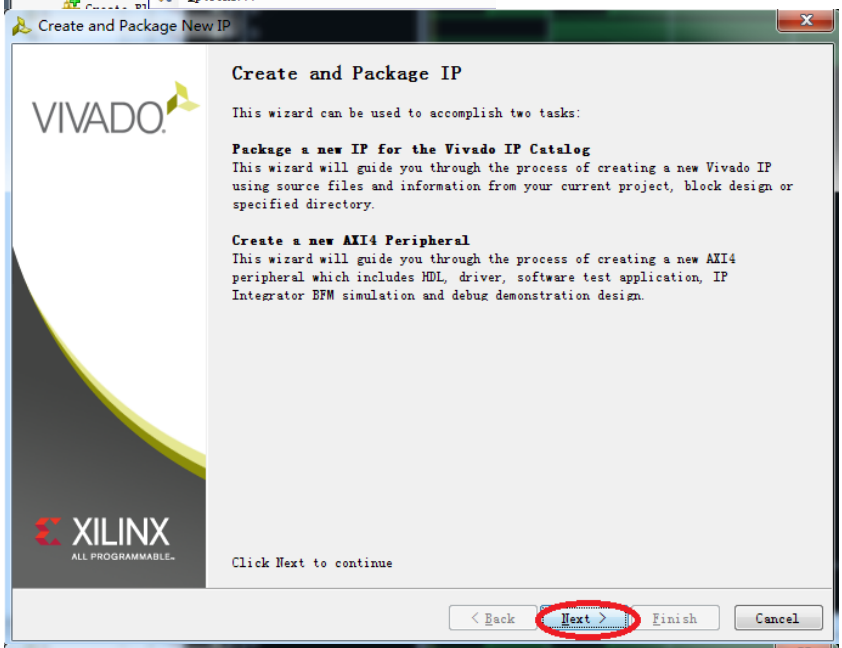


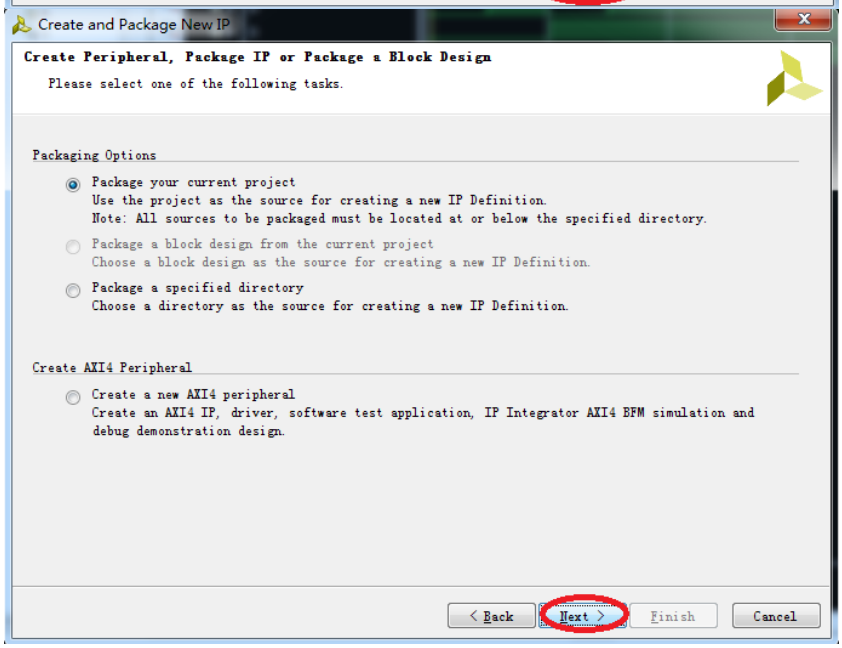
3）在 Project Settings 对话框中选择 IP，并进入 Packager 选项卡，如图进行设置。设置好后，点击 Apply，然后点击 OK。记住这里设置的各个属性。



4）在 Vivado 的菜单栏中选择 Tools->Create and Package IP…。在弹出的窗口中点击 Next。在之后弹出的窗口中如图 2-39 所示设置封装选项。点击 Next。

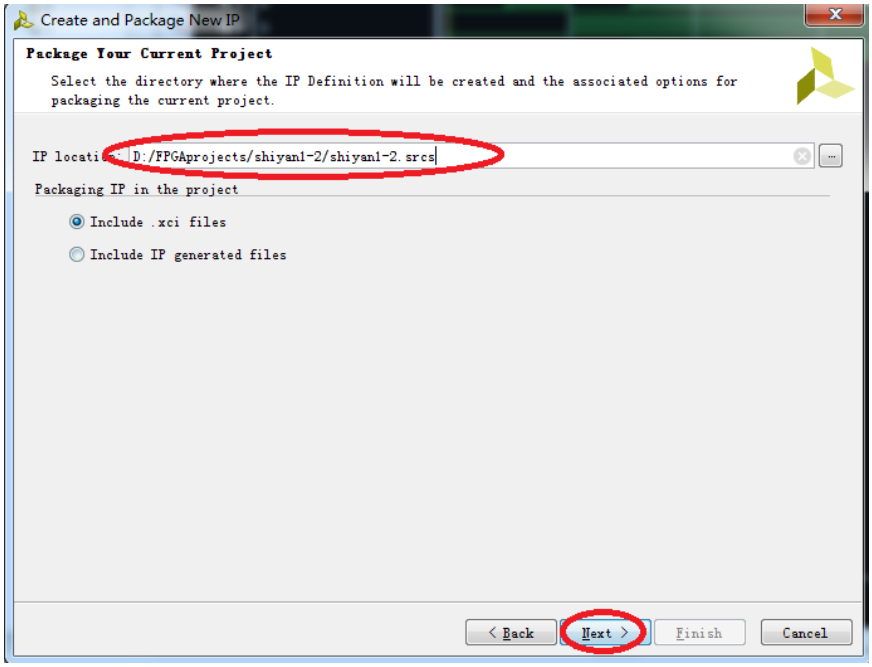




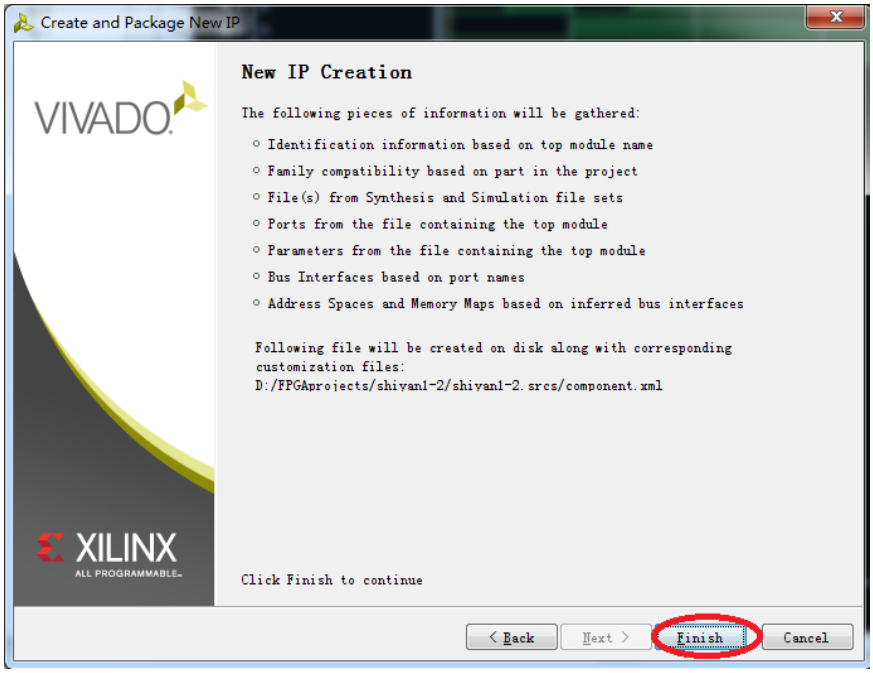


5）设置 IP Location，可以使用默认路径，不做修改，不过我们知道了封装后的 IP 放在了 D:/FPGAprojects/shiyan1-2/shiyan1-

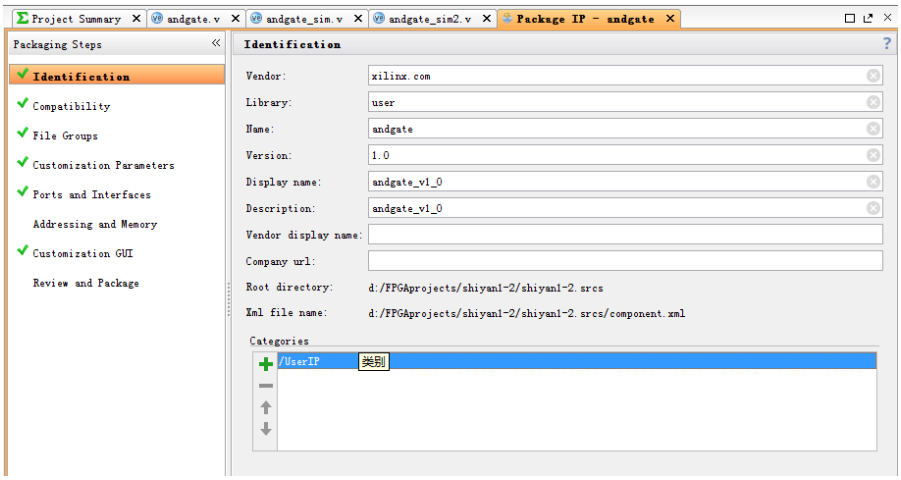
2. srcs 这个文件夹中，点击 Next。



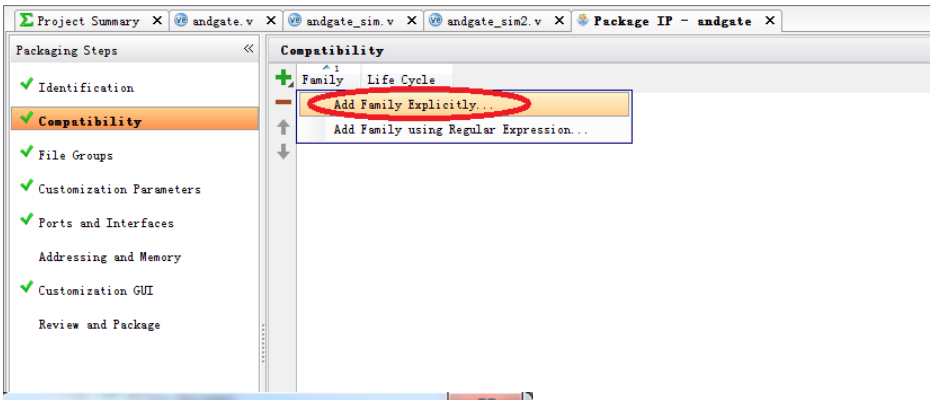
6）随后点击 Finished，进行 IP 核封装的具体设置。

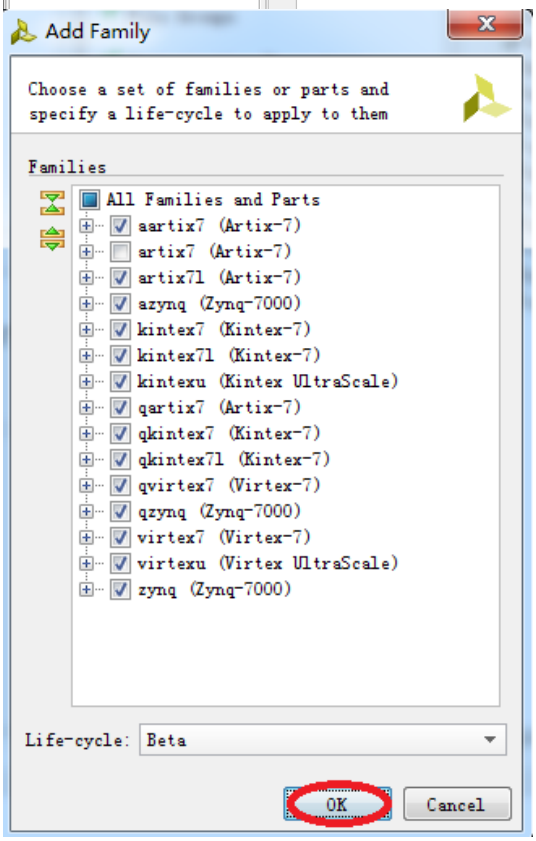


7）Identification 设置，可以就是用默认设置，不做修改。

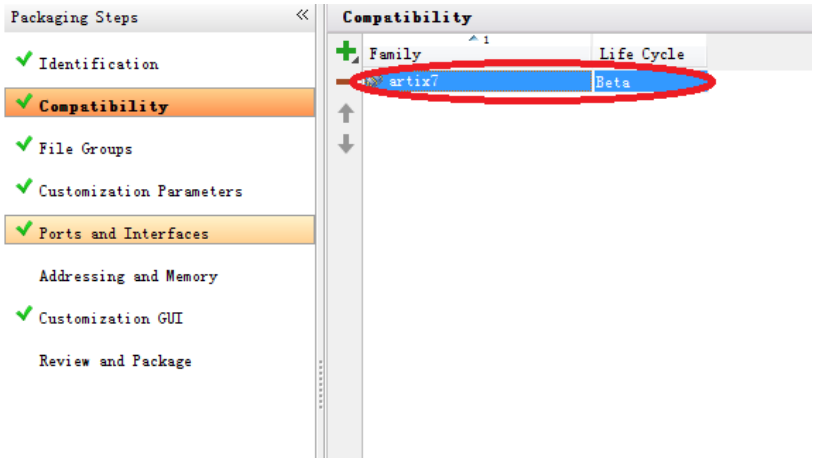


8）Compatibility 设置，添加 IP 核支持的芯片家族，点击 Add Family Explicitly,选中除 artix7 之外的所有芯片家族（因为 artix7 系列已经有了），然后点击 OK。这样就设置完了 compatibility。

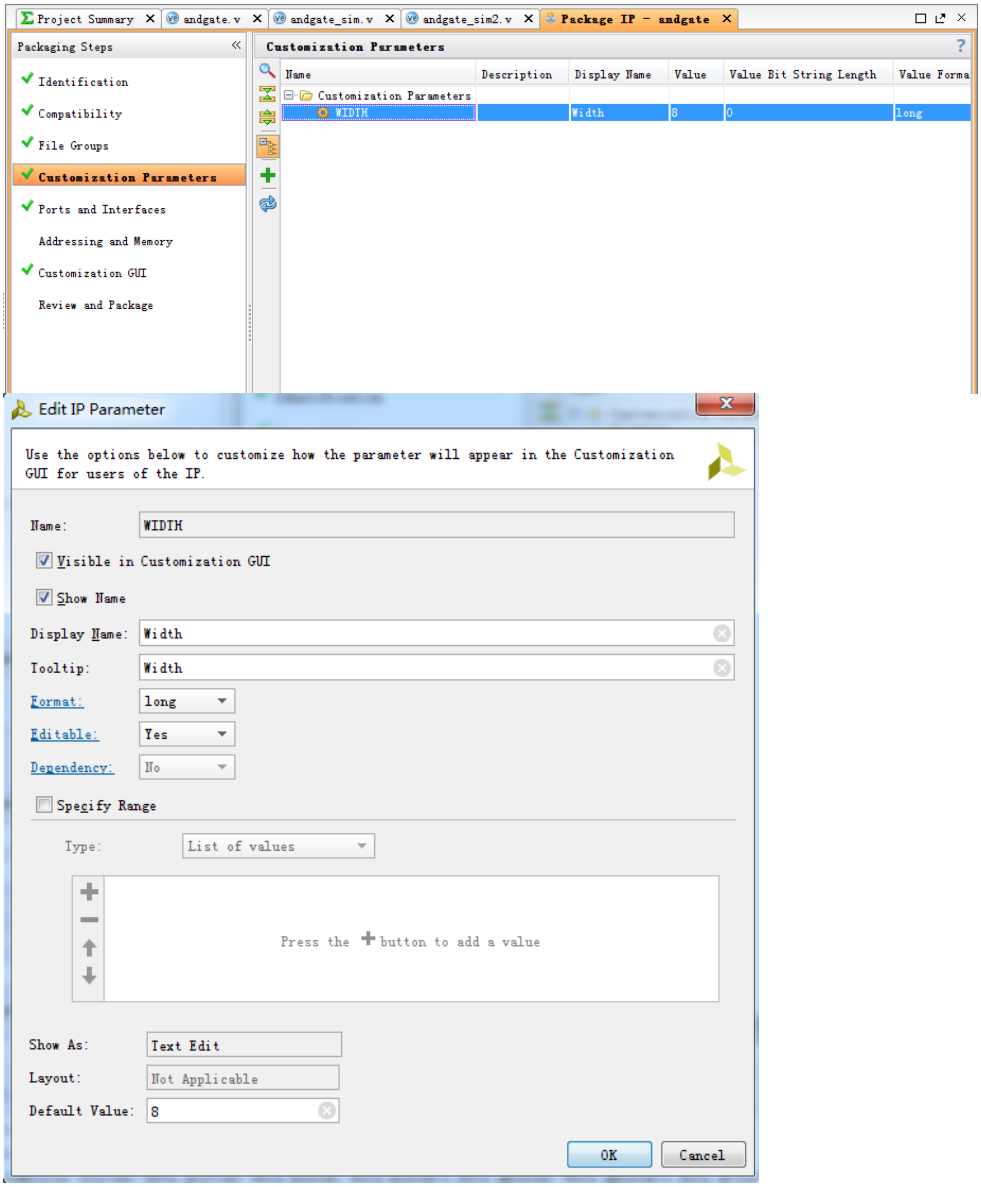




在这一步中，只要保证 artix7 系列被选中即可。

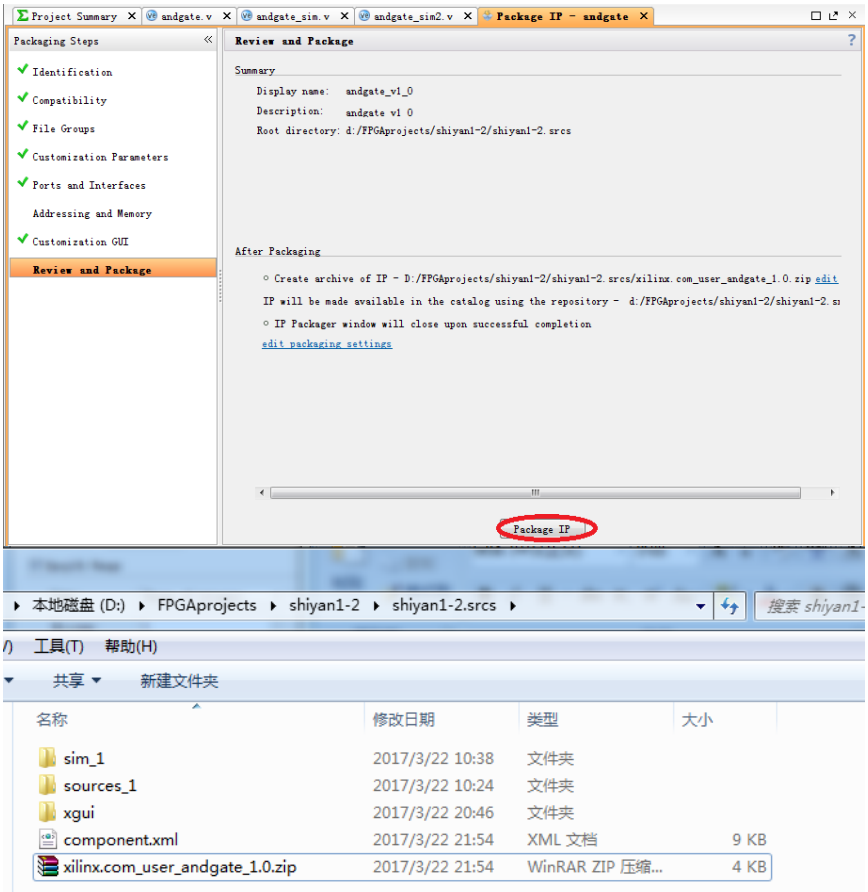


9）接下来设置 Customization Parameters。双击 WIDTH（图中高亮部分），弹出 Edit IP Parameter 对话框，设置参数后点击 OK。



10） 接下来我们到 Review and Packaging，点击 Package IP。andgate 的 IP 核就生成了，在 D:/FPGAprojects/shiyan1-2/shiyan1-2.srcs 路径下，xilinx.com_user_andgate_1.0.zip 这

个文件中。



4、用以上方法分别设计或门、非门、与非门等基本门电路并封装成 IP 核。