

《计算机数字逻辑设计》

第一次实验

1. 实验一 实验平台使用	3
一、实验目的	3
二、实验环境与仪器设备	3
三、实验要求	3
四、熟悉实验板	4
五、实验内容与操作步骤	5
六、实验板观察与验证	17
七、实验数据处理及其他要求	24
2. 实验二 组合逻辑电路设计——数据选择器	25
一、实验目的	25
二、实验环境与仪器设备	25
三、实验要求	25
四、实验原理	25
五、实验任务	27
六、实验数据处理及其他要求	28
3. 附录 A Verilog 简明手册	29
4. 附录 B EGO1 板硬件约束资源	31
5. 附录 C 操作常见问题及注意事项	38

实验板安全使用规范

更改跳帽位置或使用扩展接口扩展电路应用前请关闭电路板总开关，避免损坏器件。

电路板建议在绝缘平台上使用，不用金属或其他导电物触碰电路板线路，否则可能引起电路板损坏。

电路使用时应防止静电。秋冬季时，操作电路板前，手或纤维手套应先接触接地金属物。

液晶显示器件或模块结雾时，不要通电工作，防止电极化学反应，产生断线。

电源正负极、输入/输出端口定义时需谨慎，避免因接反引起开发板的损坏。

保持电路板的表面清洁。

小心轻放，避免不必要的硬件损伤

1. 实验一 实验平台使用

一、实验目的

- 1、掌握 Vivado 软件安装与使用，熟悉在 Vivado 环境下利用 Verilog HDL 进行电路设计、仿真、综合和下载的全过程。
- 2、掌握 Verilog HDL 基本运用，能完成电路设计和 Test Bench 程序的编写。
- 3、了解实验开发板的使用，通过观察和数据记录理解仿真和 FPGA 实现的差异。

二、实验环境与仪器设备

- 1、一台内存 4GB 以上，装有 64 位 Windows 操作系统和的 PC 机。
- 2、Vivado2017.4 或以上版本
- 3、Xilinx 实验板（EGO1）

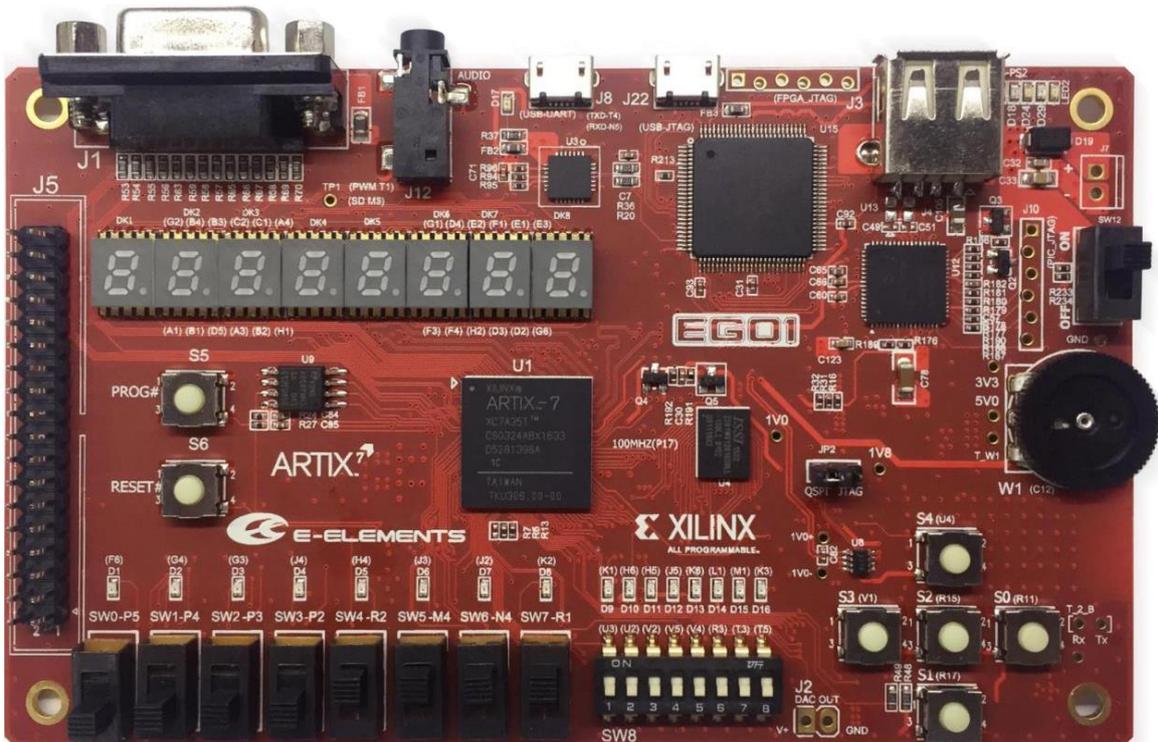
三、实验要求

- 1、实验预习阶段：做好实验内容预习及准备工作，并提交预习报告。复习逻辑门电路的工作原理和运算规则；学习 Verilog HDL 基本语法和电路设计方法；熟悉实验操作流程，拟定实验步骤（利用 Visio 等绘图工具绘制）。
- 2、根据实验内容，在 Vivado 环境中完成设计、仿真与硬件下载，记录并分析讨论实验结果的正确性，最后给出测试结论。
- 3、实验完毕，写出实验报告。实验报告必须包含的内容如下（可以根据实验任务增加其他内容）：
 - （1）实验目的
 - （2）实验内容
 - （3）实验环境
 - （4）实验预习内容：包括实验原理及设计思路说明、实验步骤设计（利用 Visio 等绘图工具绘制）、逻辑设计过程（组合电路包括：真值表、卡诺图、逻辑表达式等；时序电路包括：输出函数表达式、激励函数表达式、次态方程组、状态表或状态图等）、Verilog 设计源文件和仿真源文件、实验验证方案等。
 - （5）实验过程记录
 - （6）实验体会与小结

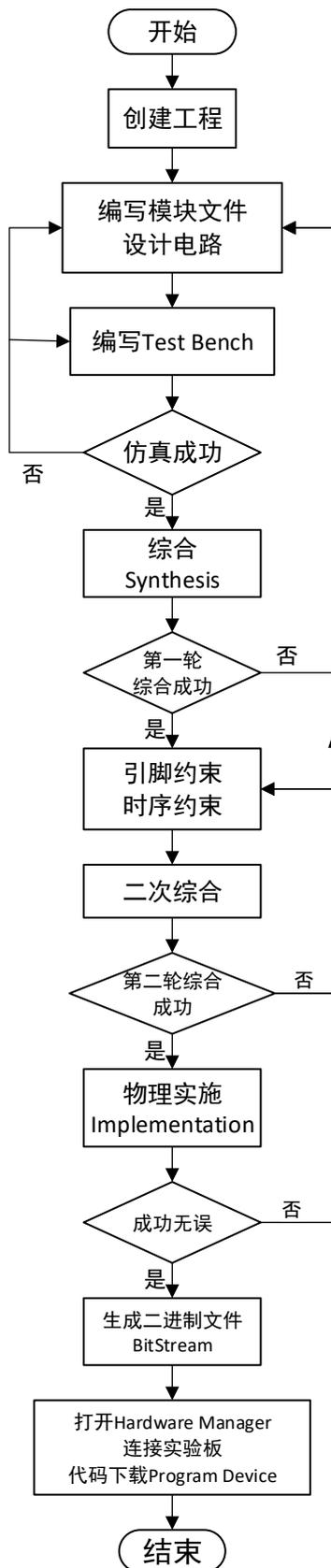
四、熟悉实验板

功能特性：

- 采用 Xilinx Artix-7 XC7A35T芯片
- 配置方式：USB-JTAG/SPI Flash
- 高达100MHz 的内部时钟速度
- 存储器：
 - 2Mbit SRAM
 - N25Q064A SPI Flash（样图旧款为N25Q032A，**千万注意**）
- 通用IO：
 - Switch : x8
 - LED: x16
 - Button: x5
 - DIP: x8
- 通用扩展IO: 32pin
- 音视频/显示：
 - 7段数码管: x8
 - VGA视频输出接口
 - Audio音频接口
- 通信接口：
 - UART: USB转UART
 - Bluetooth: 蓝牙模块
- 模拟接口：
 - DAC: 8-bit分辨率
 - XADC: 2路12bit 1MSPS ADC



五、实验内容与操作步骤



本实验指导书以 Vivado 2018.1 为例，进行说明。其他版本的 Vivado 软件的使用过程类似。

1、创建工程

打开 vivado，在图 1.1 中点击 create project 新建工程；在图 1.2 中选择自己的工作区文件夹，输入合适的工程名；图 1.3 所示的工程类型可以默认，即 RTL 项目，不包含指定代码；图 1.4 所示的器件选择需要跟硬件一致。本课程采用 Family: Artix-7, Package: csg324, 具体型号为 xc7a35tcsg324-1；最后核对相应工程信息，如图 1.5 所示。检查无误后点击 Finish 即可进入工程操作界面，如图 1.6 所示。

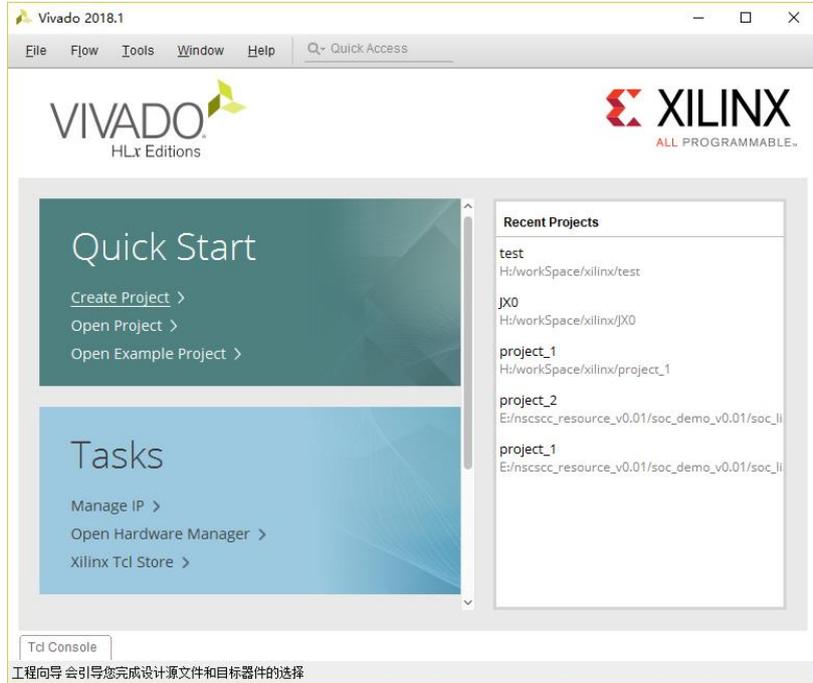


图 1.1

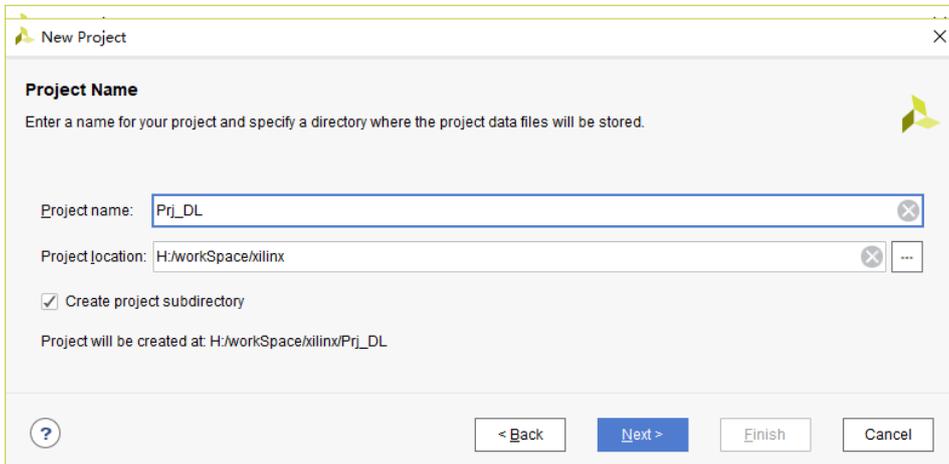


图 1.2

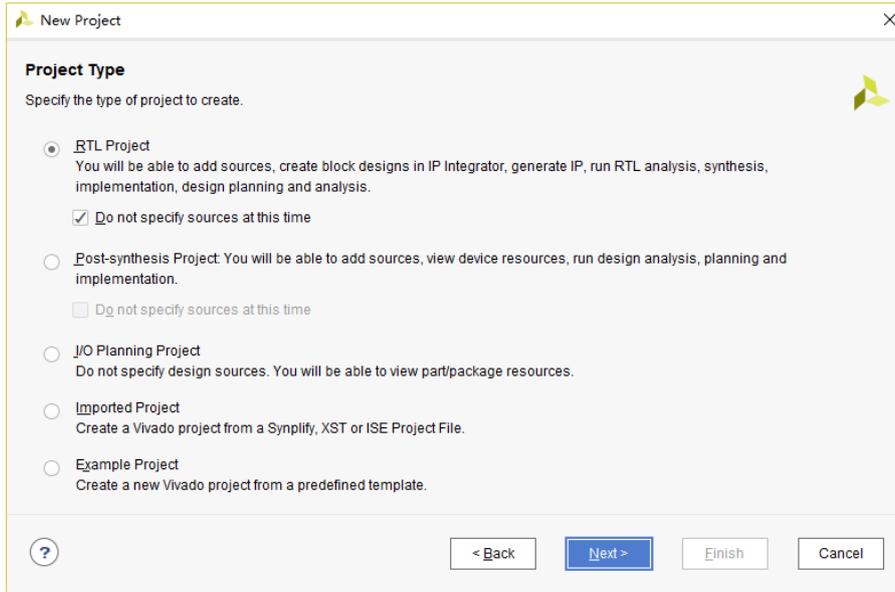


图 1.3

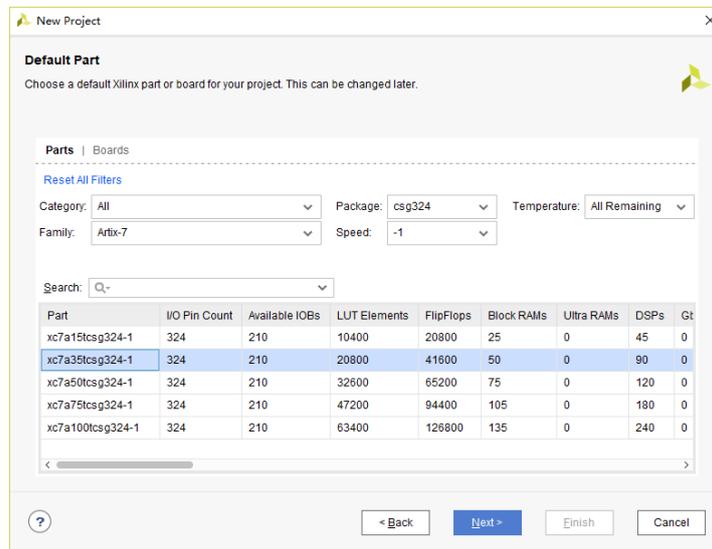


图 1.4

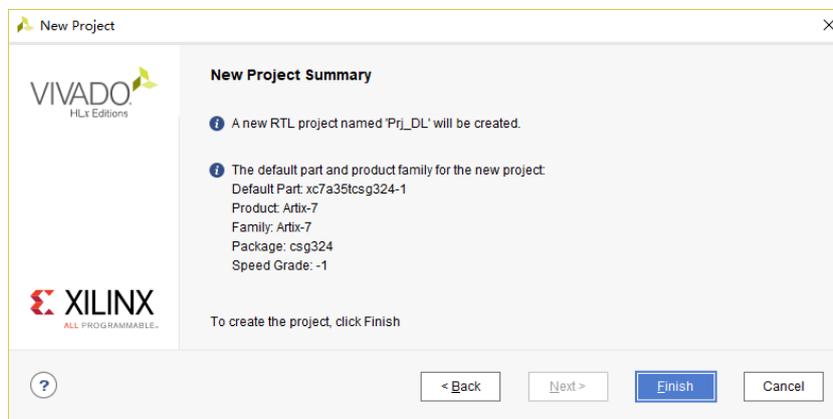


图 1.5

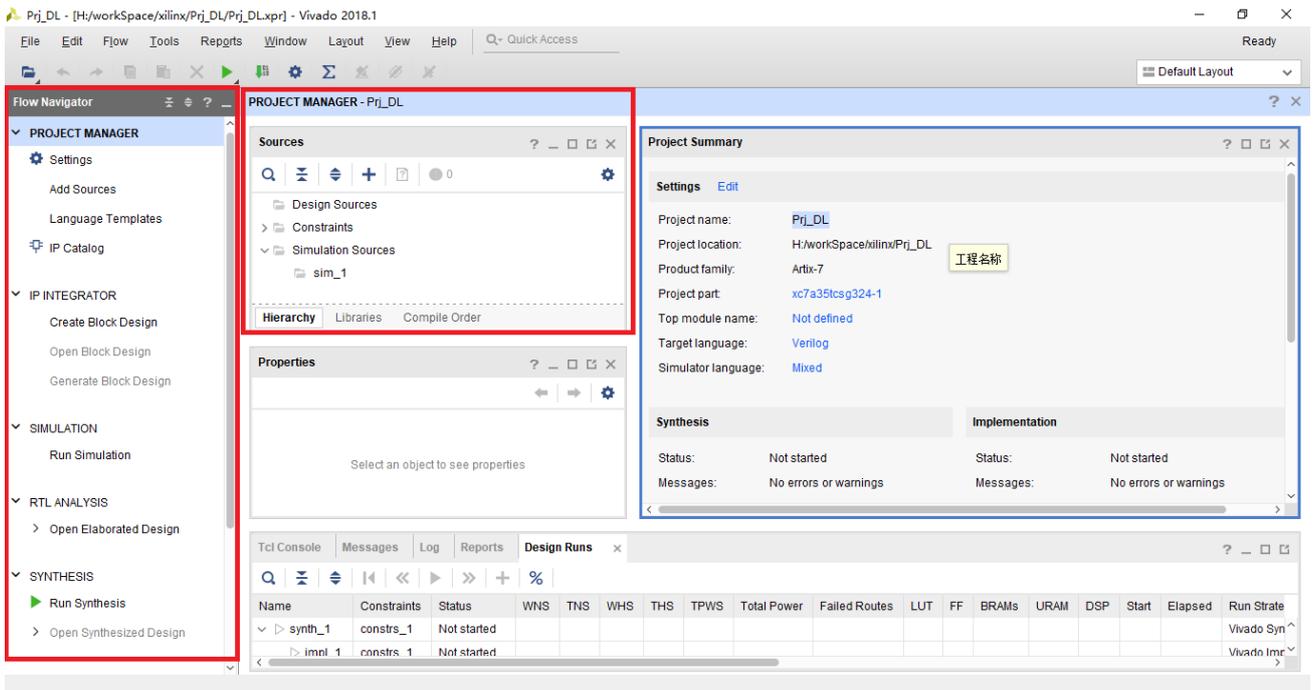


图 1.6

2、创建功能电路

创建功能电路有两种方法：原理图法和 Verilog HDL 描述法。本次实验以包含 1 个与门、1 个或门和 1 个异或门的电路模块为例，介绍利用 Verilog HDL 描述设计电路的方法。电路的原理图如下：

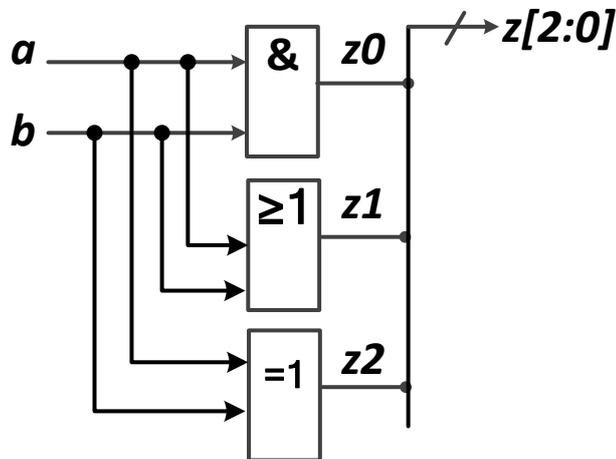
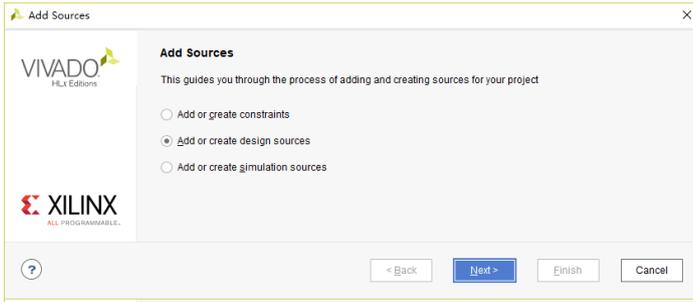
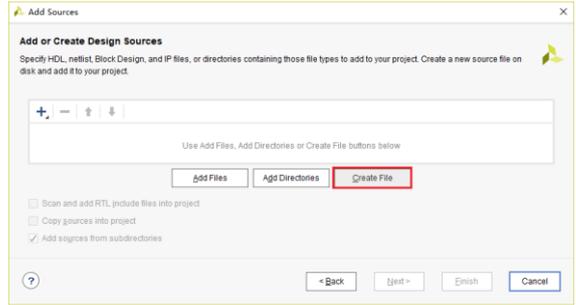


图 1.7

A) 点击左边 Flow Navigator 中，PROJECT MANAGER 中 Add Sources，（或者点击 Source 窗口中“Add Sources”按钮或快捷键 Alt+A），在图 1.8(a)对话框中选中 design sources 类型后点击 next，在图 1.8(b)对话框中点击 create 按钮，在图 1.9 对话框中设定文件名后，图 1.8(b)对话框会自动变成图 1.10 所示，点击 Finish 后会自动出现图 1.11 示意的引脚配置，本实验先做原理仿真，可以直接点击 OK 并确认跳过。操作区的源码列表自动 update 后如图 1.12。



(a)



(b)

图 1.8

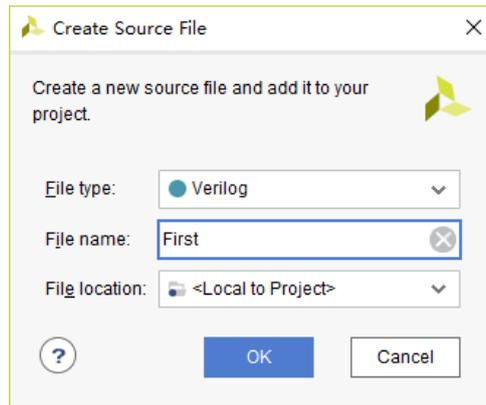


图 1.7

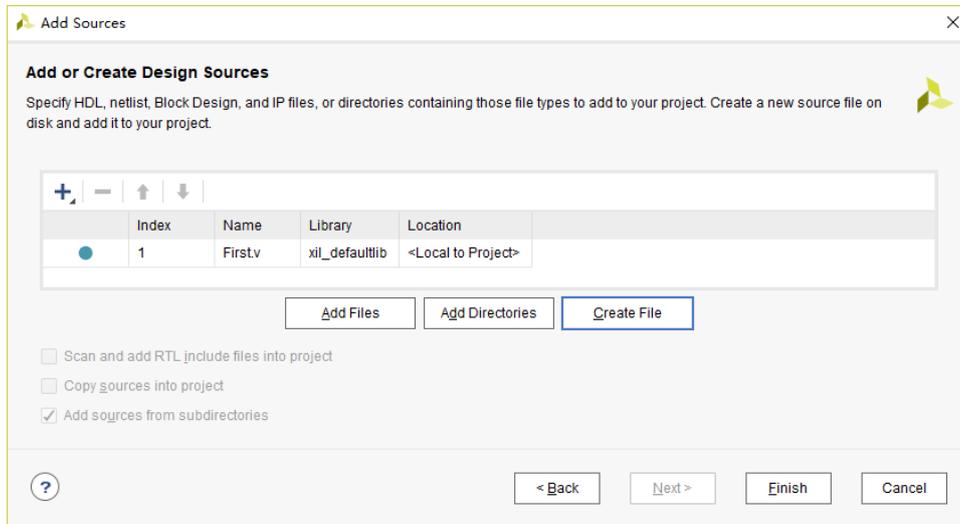


图 1.8

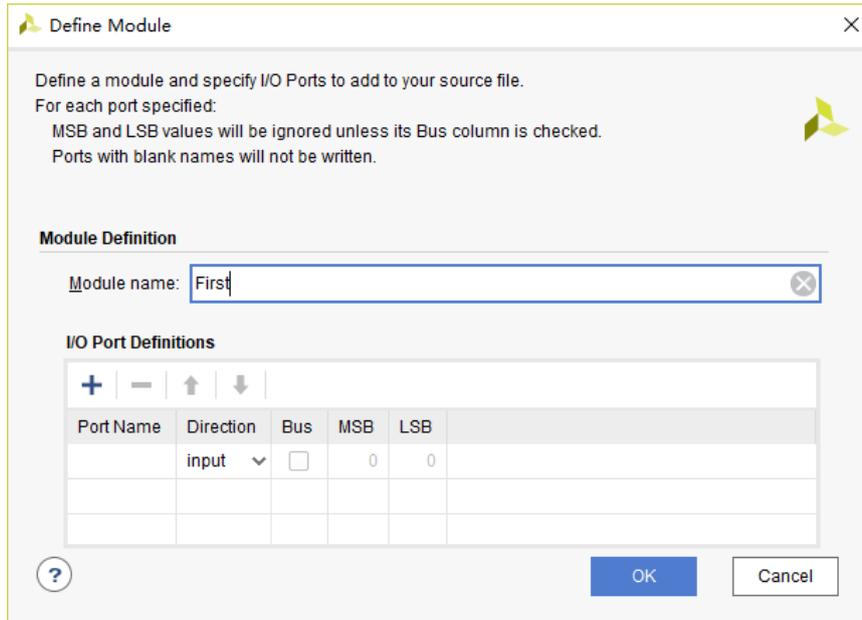


图 1.9

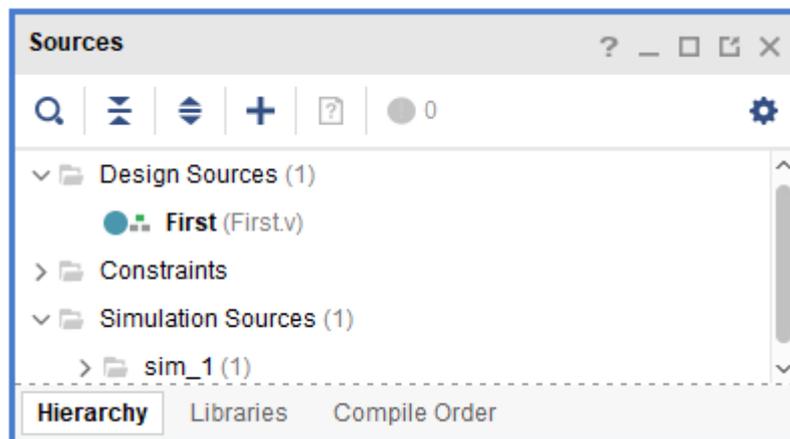


图 1.10

B) 点击 First.v, 编辑 Verilog 文件, 保存。

```

module First( input wire a, input wire b, output wire [2:0] z );
    assign z[0] = a&b;
    assign z[1] = a|b;
    assign z[2] = a^b;
endmodule

```

C) 点击 Flow Navigator 中的 RTL ANALYSIS 的 Open Elaborated Design, 如图 1.13; 自动生成器件原理图, 如图 1.14。

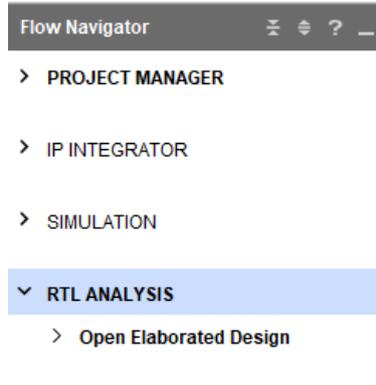


图 1.11

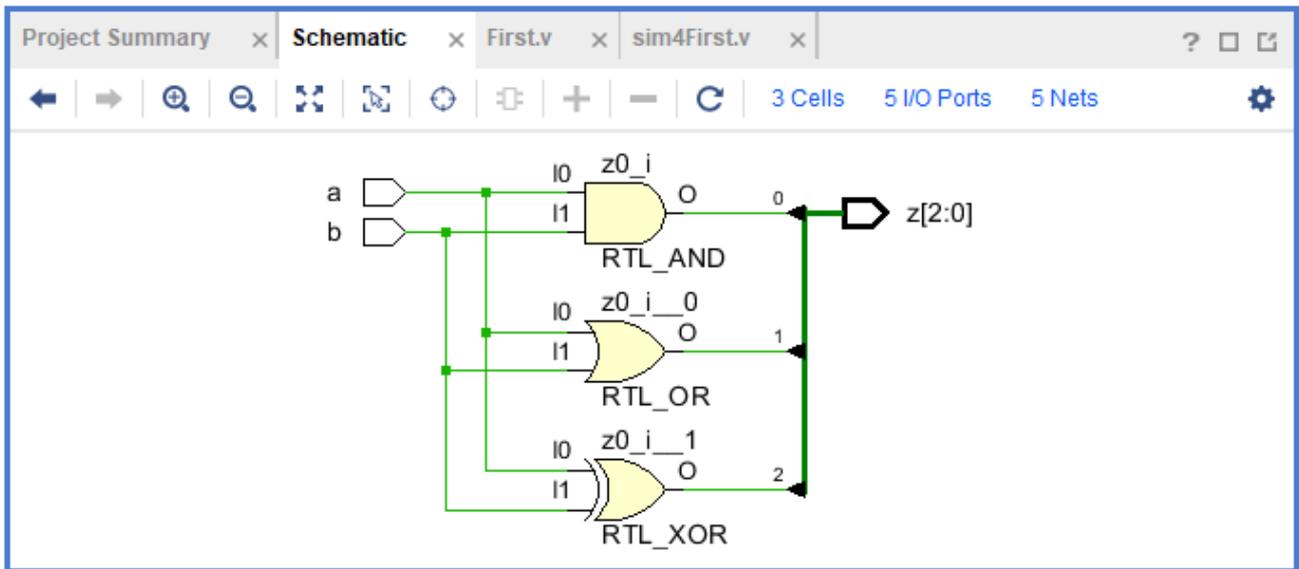


图 1.12

3、添加仿真激励程序

依照第 2 步 A) 所述，点击左边 Flow Navigator 中，PROJECT MANAGER 中 Add Sources，（或者点击 Source 窗口中“Add Sources”按钮或快捷键 Alt+A），在图 1.15 对话框中选中 simulation sources 类型后点击 next，在图 1.16 对话框中点击 create 按钮，在图 1.17 对话框中设定文件名后，图 1.16 对话框会自动变成图 1.18 所示，点击 Finish 后会自动出现图 1.19 示意的引脚配置，本实验只做原理仿真，可以直接点击 OK 并确认跳过。操作区的源码列表自动 update 后如图 1.20。

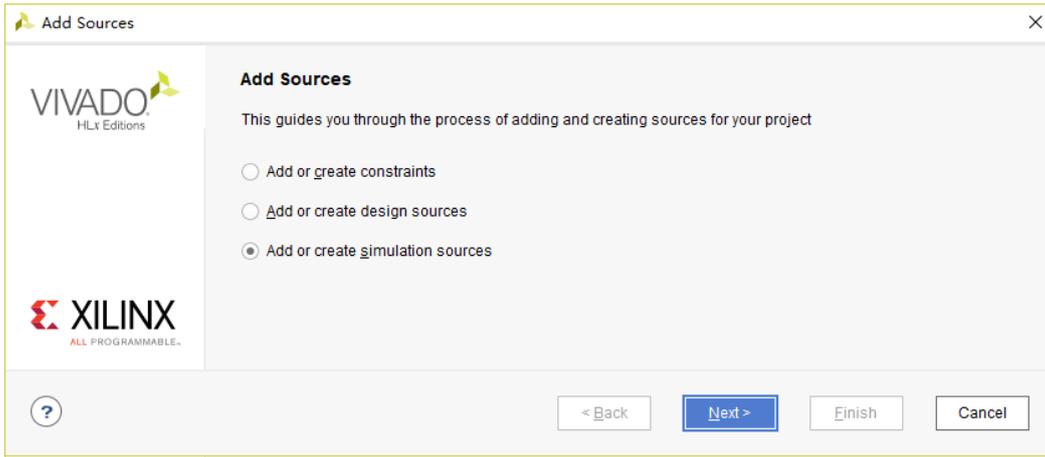


图 1.13

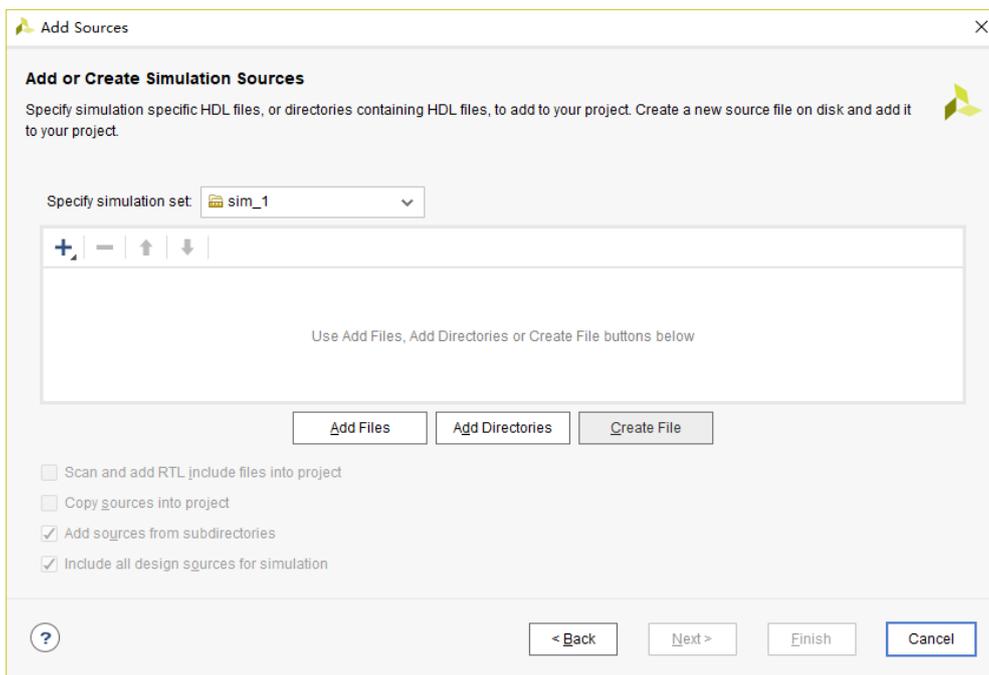


图 1.14

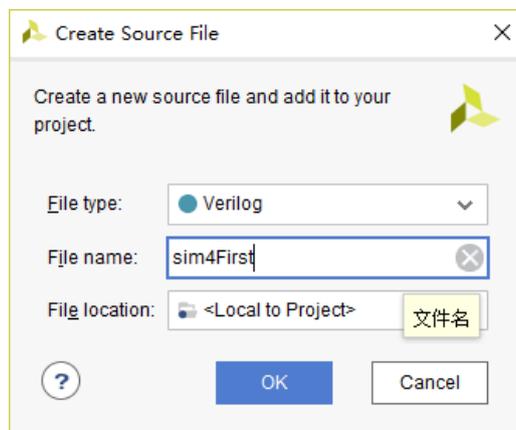


图 1.15

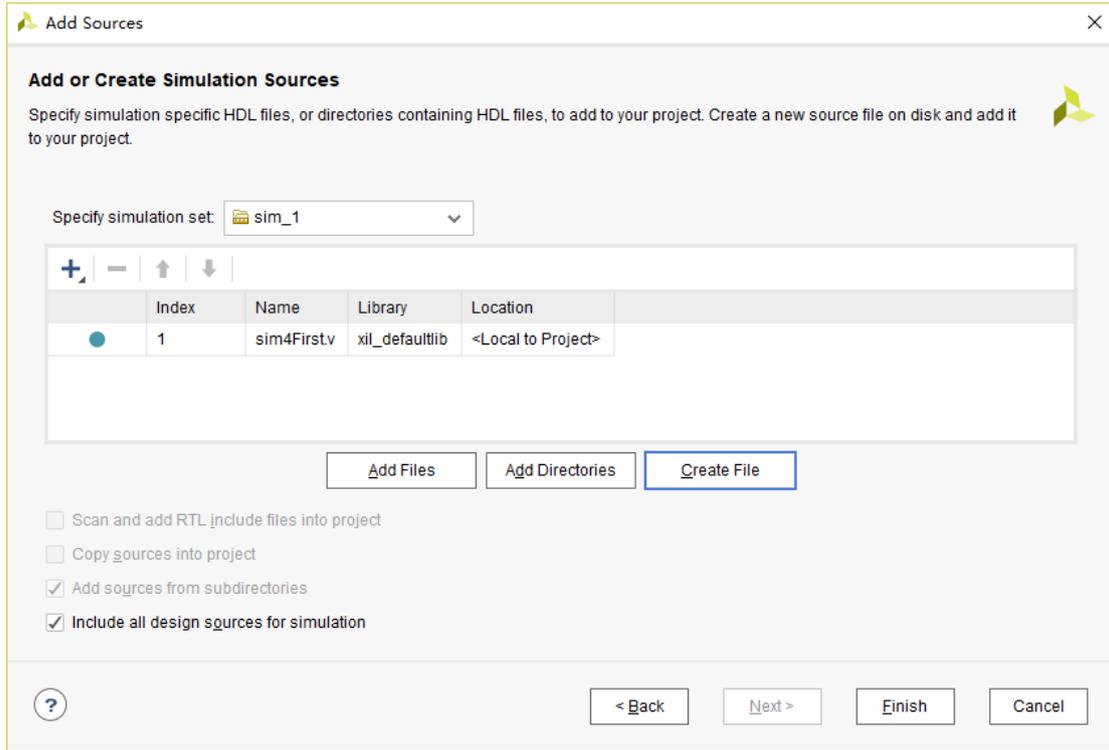


图 1.16

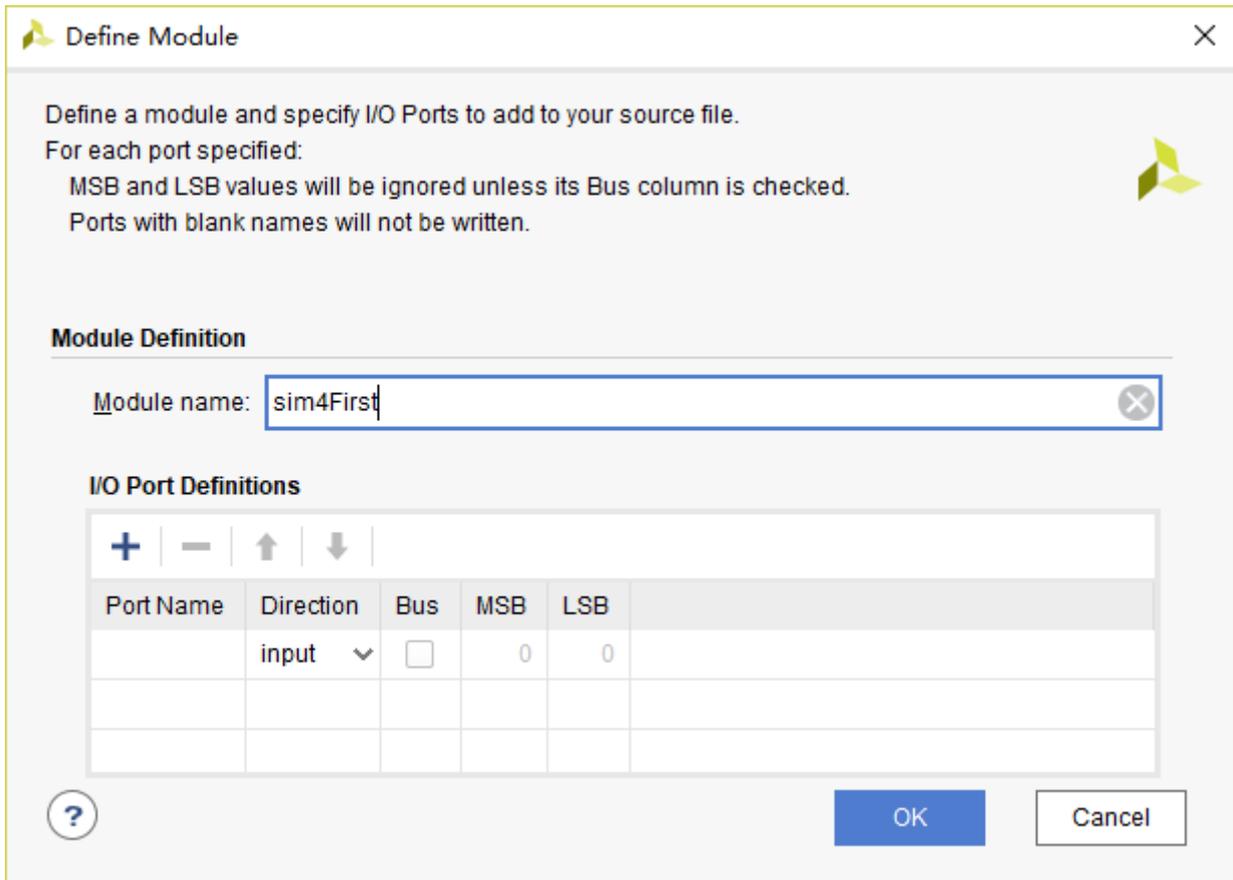


图 1.17



图 1.18

编辑 sim4First.v，加入仿真输入信号和时序。

```

`timescale 1ns / 1ps
module sim4First( );
    reg a;
    reg b;
    wire [2:0] z;
    First uut( .a(a), .b(b), .z(z) );
    always begin
        a = 0; b = 0; #100;
        a = 0; b = 1; #100;
        a = 1; b = 0; #100;
        a = 1; b = 1; #100;
    end
endmodule

```

保存后，操作区的源码列表会自动 update，得结构如图 1.21。

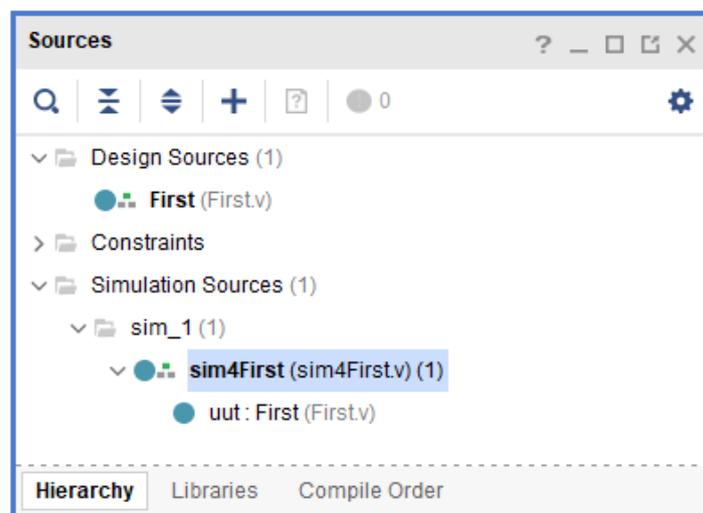


图 1.19

4、观察记录仿真波形

点击 Flow Navigator – SIMULATION – Run Simulation，如图 1.22；进入仿真状态，如图 1.23。注意此时工具栏的变化，如图 1.24。

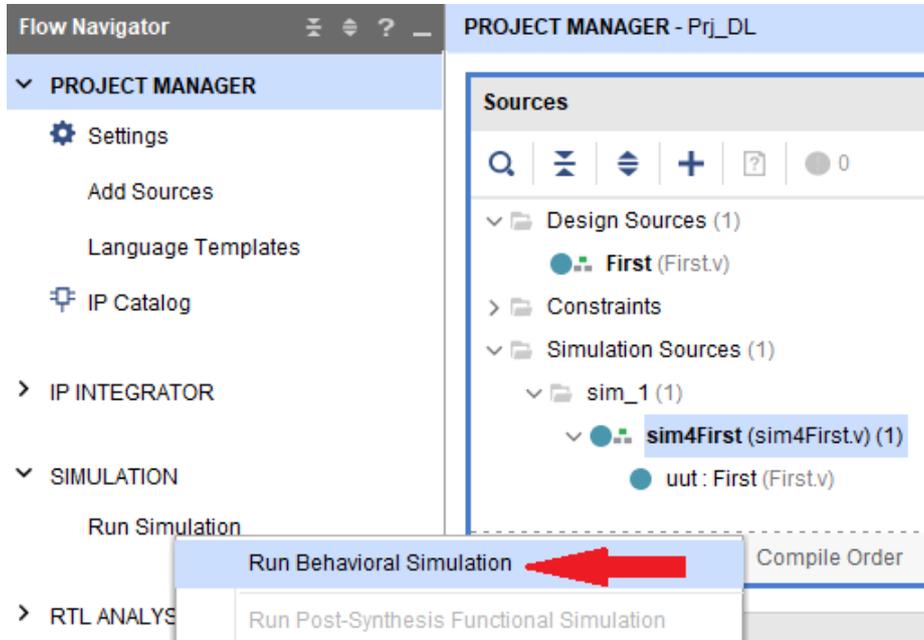


图 1.20

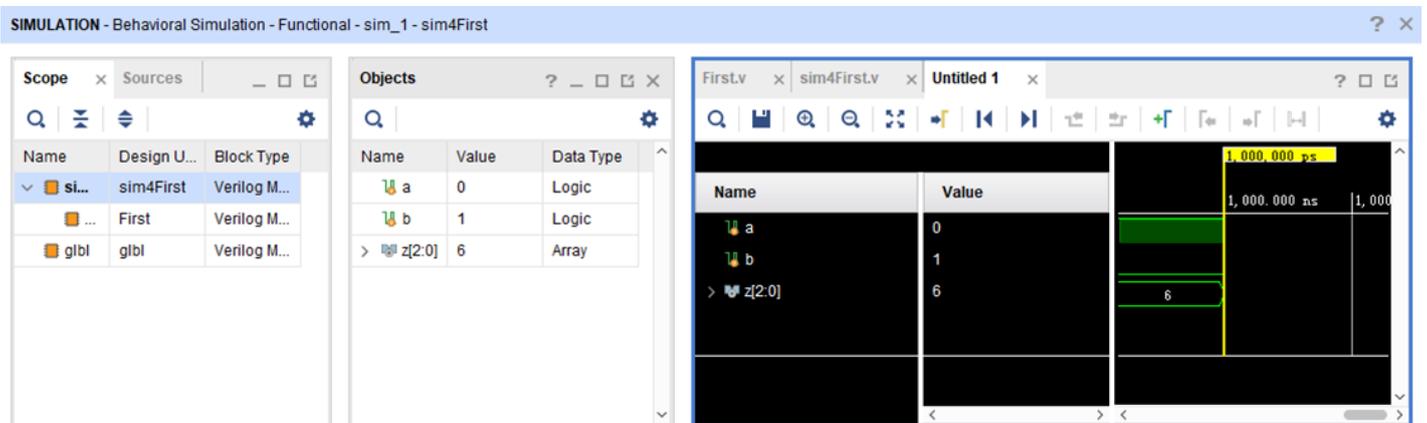


图 1.21



图 1.22

点击  按钮将仿真时间点切换至开始。由于仿真文件中我们加了时间间隔为#100ns，为仿真的简便起见，我们将运行时间片设为 100ns，如图 1.25。

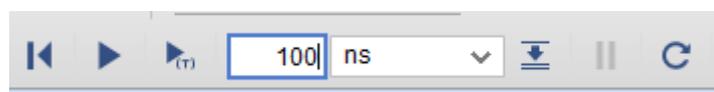


图 1.23

为了观察方便，运行前点击  将波形时间窗调整至合适的尺度，如图 1.26。

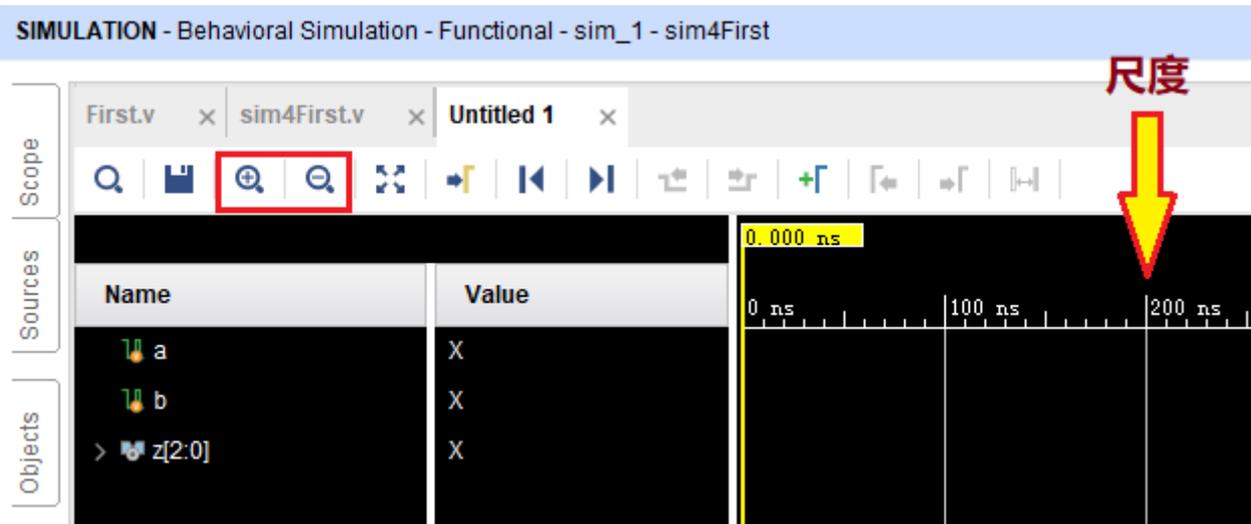


图 1.24

点击 ，观察输出波形，局部效果如图 1.27。

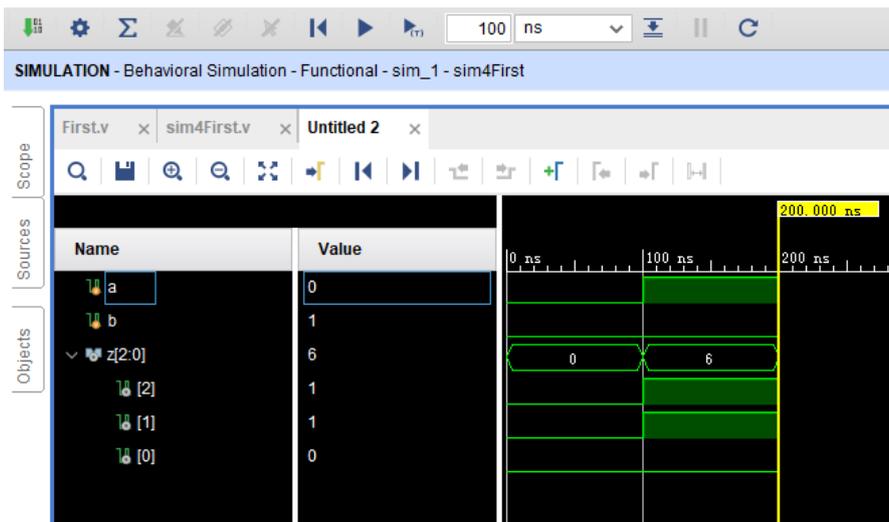


图 1.25

分析波形，记录输出引脚状态至表 1.1 中，检验真值表与逻辑电路的对应关系。

表 1.1 真值表

输 入		输 出		
引脚 a	引脚 b	引脚 z[2]	引脚 z[1]	引脚 z[0]
0	0			
0	1			
1	0			
1	1			

六、实验板观察与验证

1、添加硬件约束

为了在实验板上验证电路的功能，需要为电路的输入输出配置合适的开发板资源，这个工作通过添加硬件约束来实现。有两种方法可以添加硬件约束。一是利用 Vivado 环境提供的 IO planning 功能；二是直接新建 XDC 的硬件约束文件，手动输入约束命令。

a、方法一：利用 IO planning 添加硬件约束

先来看第一种方法，利用 IO planning 功能，用可视化的方法添加硬件约束。

a.1、点击 Flow Navigator 中 Synthesis 中的 Run Synthesis，先对工程进行综合，见图 1.28。

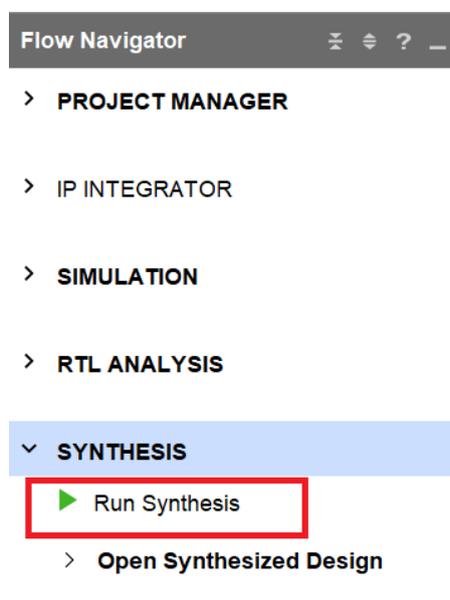


图 1.26

a.2、综合完成之后，选择 Open Synthesized Design，打开综合结果，见图 1.29。

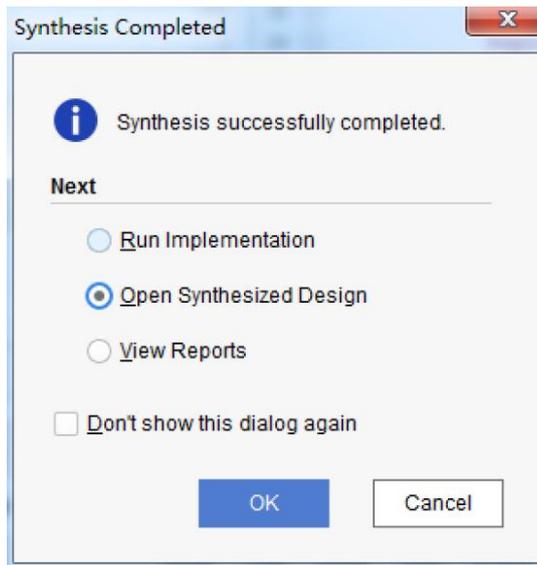


图 1.27

a.3、此时应看到下图 1.30 所示的界面。如果没出现如下界面，在图示位置的 layout 中选择 IO Planning 子菜单项即可。

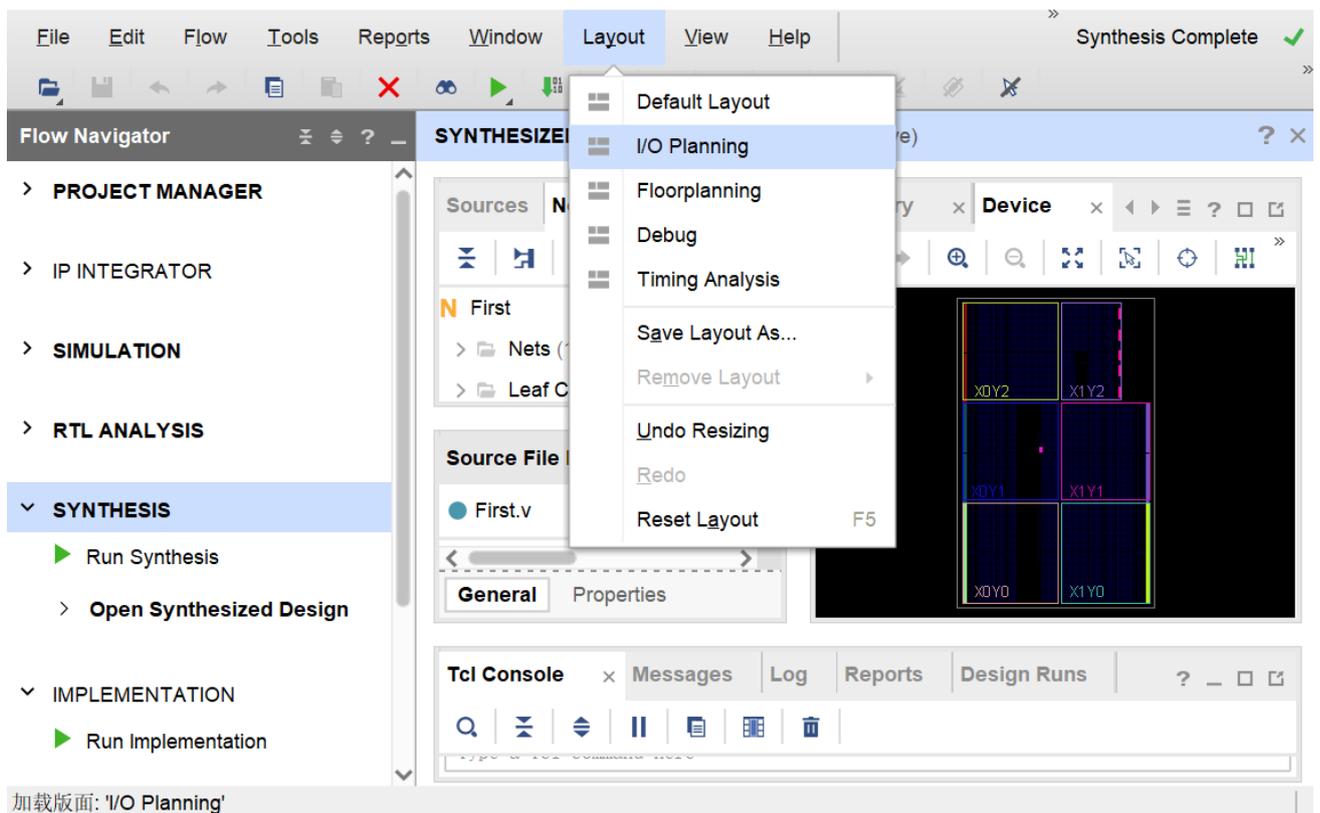


图 1.28

a.4、在图 1.30 所示界面的右下方区域，把选项卡中切换到 I/O ports 一栏，并在对应的信号后，输入对应的 FPGA 管脚标号（或将信号拖拽到右上方 Package 图中对应的管脚上），并指定 I/O std，如图 1.31 所示。具体的 FPGA 约束管脚和 IO 电平标准，可参考实验板 EG01 的用户手册或原理图）。

SYNTHESIZED DESIGN * - xc7a35tcs324-1 (active)

Tcl Console Messages Log Reports Design Runs Package Pins I/O Ports x

Q [] [] [] [] [] []

Name	Direction	Neg ...	Package Pin	F...	...	I/O Std	Vcco
z (3)	OUT			<input checked="" type="checkbox"/>	35	LVC MOS33*	3.300
z[2]	OUT		J3	<input checked="" type="checkbox"/>	35	LVC MOS33*	3.300
z[1]	OUT		J2	<input checked="" type="checkbox"/>	35	LVC MOS33*	3.300
z[0]	OUT		K2	<input checked="" type="checkbox"/>	35	LVC MOS33*	3.300
a	IN		P5	<input checked="" type="checkbox"/>	34	LVC MOS33*	3.300
b	IN		P4	<input checked="" type="checkbox"/>	34	LVC MOS33*	3.300

图 1.29

a.5、完成之后，点击左上方工具栏中的保存按钮，工程提示新建 XDC 文件或选择工程中已有的 XDC 文件。本次实验，选择 Create a new file，输入 File name，点击 OK，如图 1.32 所示。至此，完成硬件约束的添加操作。

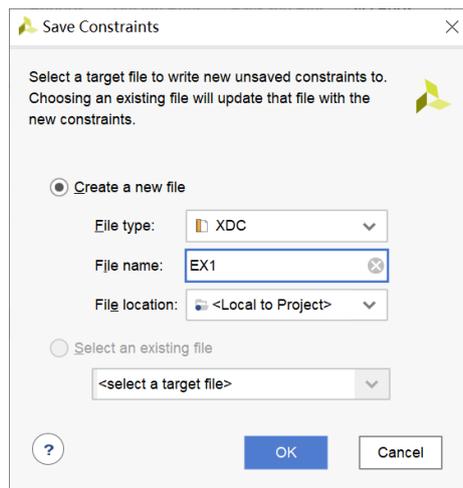


图 1.30

a.6、此时，在 Sources 下 Constraints 中会找到新建的 XDC 文件，见图 1.33。

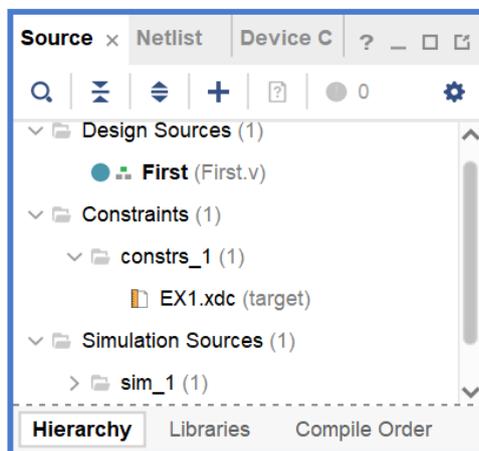


图 1.31

b、方法二：新建或添加约束文件，完成硬件约束

如何利用第二种方法添加约束文件。

b.1、打开 Add Sources 界面，如图 1.34 所示。选择第一项 Add or Create Constraints 一项，点击 Next，进入图 1.35 所示的界面。

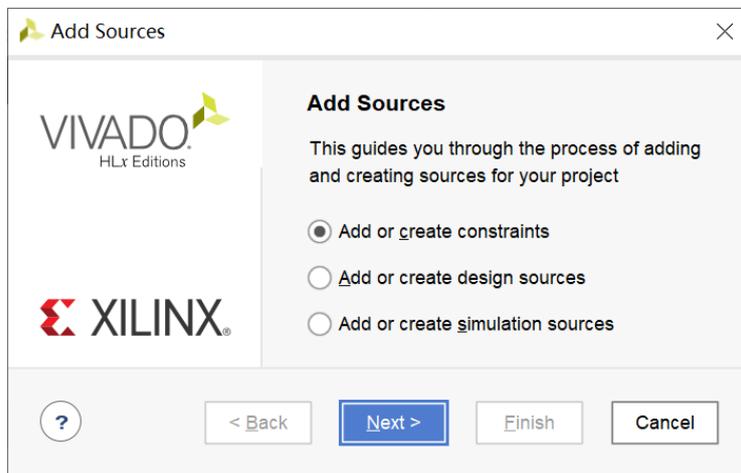


图 1.32

b.2、在图 1.35 所示的界面上，点击 Create File，新建一个 XDC 文件，输入 XDC 文件名，点击 OK。点击 Finish，如图 1.36 所示。

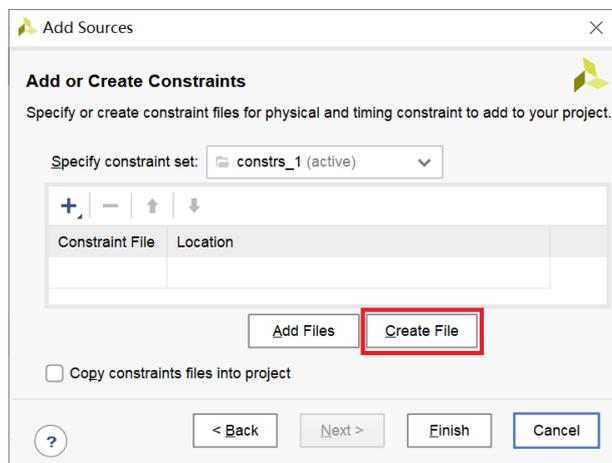


图 1.33

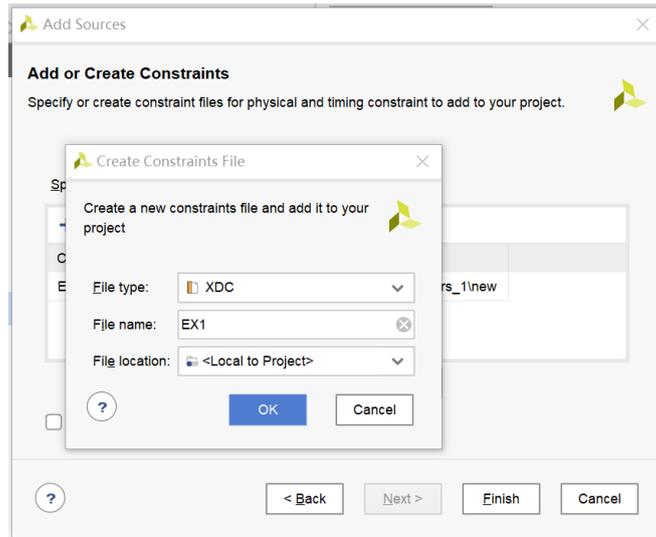


图 1.34

b.3、双击打开新建好的 XDC 文件，并按照如图 1.37 所示规则，输入相应的 FPGA 管脚约束信息和电平标准。

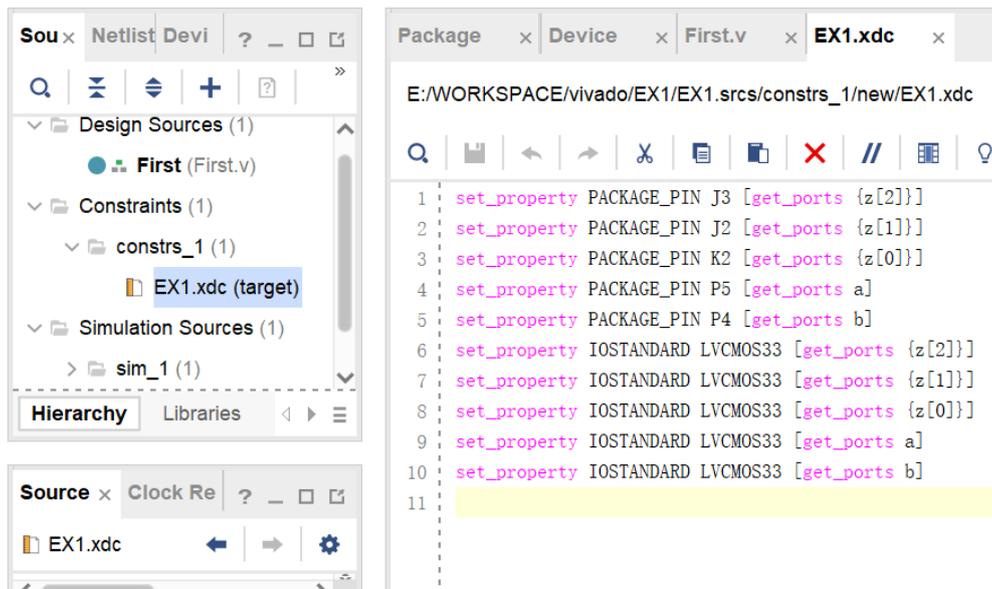


图 1.35

2、添加时序约束

本实验电路属于简单的组合逻辑电路，因此可以跳过时序约束设计。

若要实现时序约束，按以下步骤完成。

a、在综合完成后打开选择 Open Synthesis Design, 或者从 Flow Navigator 中选择 Open Synthesis Design。

b、在 Flow Navigator 中选择 Synthesis > Synthesized Design > Edit Timing Constraints，打开时序约束界面，为电路添加时钟并配置引脚，进而完成时序约束。

3、电路生成

如图 1.38 所示，依次进行网表综合 SYNTHESIS、实现 IMPLEMENTATION 并生成二进制流文件后，即完成硬件电路的生成操作。



图 1.36

4、实验板使用

a、用实验板配套的 USB 下载线将 EGO1 实验板连接至电脑后，打开实验板电源。

b、点击 Flow Navigator 中 Open Hardware Manager 一项，进入硬件编程管理（HARDWARE MANAGER），选择 Auto Connect 连接到板卡，如下图 1.39 所示。【也可在 Flow Navigator 中展开 Hardware Manager，点击 Open Target，选择 Auto Connect 连接到板卡】

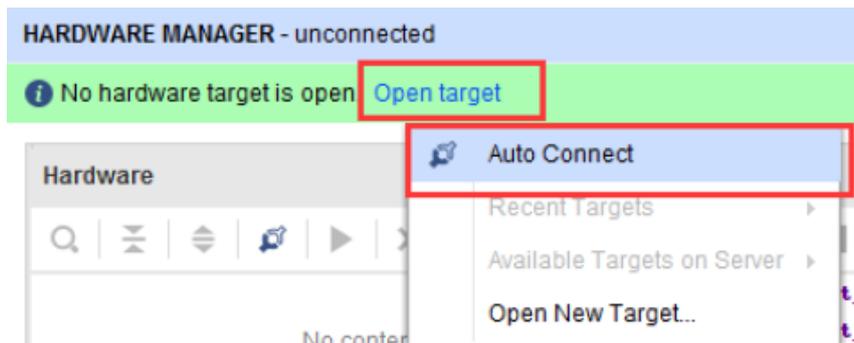


图 1.37

c、连接成功后，在目标芯片上右击，选择“Program Device”。在弹出的对话框中“Bitstream File”一栏已经自动加载本工程生成的比特流文件，点击“Program”对 FPGA 芯片进行编程，如图 1.40。成功下载后，目标板芯片会显示“Programmed”，如图 1.41。

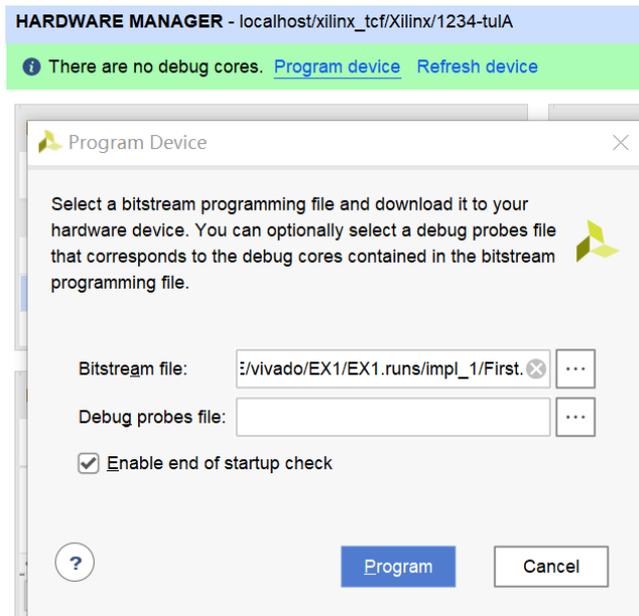


图 1.38

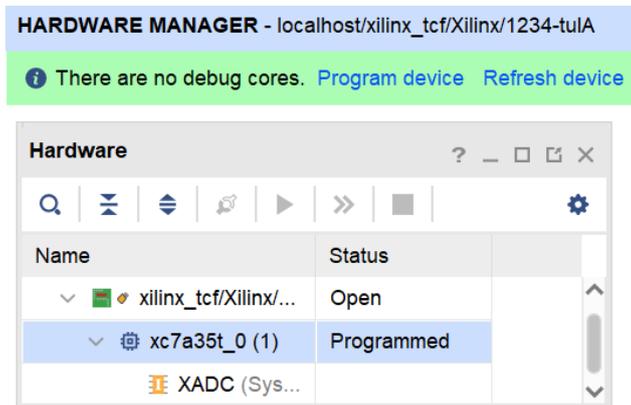


图 1.39

d、如图 1.42 所示，拨动开关 SW7 和 SW6，观察 LD2 灯组中第 2~第 0 个 LED 的状态



图 1.40

七、实验数据处理及其他要求

- 1、分别测试三个门的逻辑关系，**记录**仿真波形，**整理**实验数据并**填写**真值表，**分析验证**实验结果是否符合逻辑表达式功能。
- 2、记录实验中出现的**问题**并分析其原因。
- 3、总结 FPGA 实施方案，记录硬件引脚约束（如下表，自行补充）。

输入				输出			
名称	FPGA 管脚	EGO1 器件 (丝印)	作用	名称	FPGA 管脚	EGO1 器件 (丝印)	作用
a	P5	SW7	输入 a 电平	Z(0)	K2	LD2(0)	a&b
b				Z(1)			
				Z(2)			

- 4、扩展要求：构造其他常见复合门（与非、或非、与或非、同或），重复上述实验过程。

2. 实验二 组合逻辑电路设计——数据选择器

一、实验目的

- 1、深化对数据选择器工作原理和逻辑功能的理解，了解其如何对输入信号进行选择 and 传输。
- 2、掌握组合逻辑电路的设计方法，能利用 Verilog HDL，采用结构描述法、数据流描述法和行为描述法这三种进行数据选择器的电路建模。
- 3、掌握数据选择器的使用方法。
- 4、掌握在 Vivado 环境下，用 Verilog HDL 进行电路建模、仿真验证并利用 FPGA 进行板级验证的硬件电路设计方法。

二、实验环境与仪器设备

- 1、一台内存 4GB 以上，装有 64 位 Windows 操作系统和的 PC 机。
- 2、Xilinx 实验板（EGO1）一个。
- 3、Vivado 2017.4 以上版本软件以及 XUP library IP 包。

三、实验要求

1、实验预习阶段

预习实验内容及准备工作，并提交预习报告。复习 MUX（数据选择器）工作原理；拟定实验步骤及操作流程（利用 Visio 等绘图工具绘制），整理逻辑电路设计思路，画出真值表、卡诺图，写出逻辑表达式等；利用 Verilog HDL 设计源文件和仿真文件；设计实验验证方法和数据分析等。

2、实验实施阶段

利用 Verilog HDL 设计相关电路，在 Vivado 中完成设计与仿真，记录并分析讨论实验结果的正确性，最后给出测试结论。

3、实验工作总结阶段

实验完毕，写出实验报告。

四、实验原理

1、“2 选 1”数据选择器原理

图 2.1 所示为一个“2 选 1”的数据选择器，a、b 为信号输入端，c 为选择控制端，y 为数据输出端。利用表 2.1 所示真值表，可知：当 $c=0$ 时 $y=a$ ；当 $c=1$ 时 $y=b$ 。从而可得 y 的逻辑表达式：

$$y = \bar{c}a + cb \quad \text{式 2.1}$$

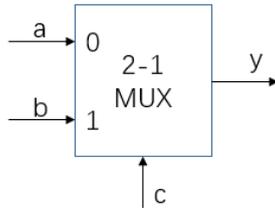


图 2.1 “2 选 1” 数据选择器原理框图

表 2.1 “2 选 1” 数据选择器真值表

c	y
0	a
1	b

以下给出其参考 Verilog HDL 源程序:

```

module mux21( input a,input b,input c,output y);
    assign y = c ? b : a;
endmodule

```

2、设计“4 选 1”数据选择器

“4 选 1”数据选择器的原理框图如图 2.2 所示，其真值表如表 2.2 所示。

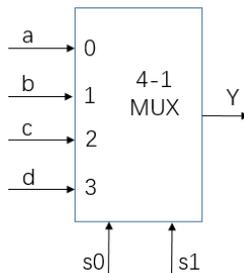


图 2.2 “4 选 1”数据选择器原理框图

表 2.2 “4 选 1”数据选择器真值表

s1	s0	Y
0	0	a
0	1	b
1	0	c
1	1	d

为简便起见，表中用总线 I[3:0]来命名输入信号 d-a。可知 Y 的逻辑表达式为：

$$Y = \bar{S}_1 \bar{S}_0 a + \bar{S}_1 S_0 b + S_1 \bar{S}_0 c + S_1 S_0 d \quad \text{式 2.2}$$

方法一：根据“4 选 1”数据选择器原理，用 Verilog HDL 采用结构描述法、数据流描述法或行为描述法进行电路建模。

方法二：采用模块化设计方法（迭代法）利用“2 选 1”数据选择器构造“4 选 1”数据选择器电路结构可按下图设计：

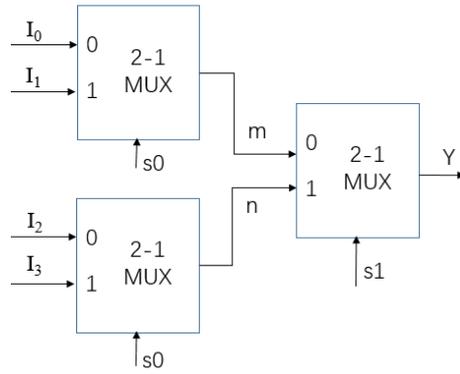


图 2.3 利用迭代法设计的“4 选 1”数据选择器原理框图

以下给出其参考 Verilog HDL 源程序：

```

module mux41(input [3:0] I, input [1:0] S, output Y);
    wire m,n;
    mux21 M1(.a(I[0]), .b(I[1]), .c(S[0]), .y(m));
    mux21 M2(.a(I[2]), .b(I[3]), .c(S[0]), .y(n));
    mux21 M3(.a(m), .b(n), .c(S[1]), .y(Y));
endmodule

```

五、实验任务

1、在结构描述法、数据流描述法或行为描述法三种方法中，选择一种方法进行电路建模，设计“2 选 1”数据选择器，完成仿真验证和 FPGA 上板验证。

2、在结构描述法、数据流描述法或行为描述法三种方法中，选择一种方法进行电路建模，设计“4 选 1”数据选择器，完成仿真验证和 FPGA 上板验证。

3、采用迭代法，利用实验一中设计的多功能门电路模块和本次实验设计的“4 选 1”数据选择器，设计一个具有多功能逻辑运算单元 LU，完成仿真验证和 FPGA 上板验证。该逻辑运算单元的功能框图如下图 2.4 所示，其功能要求如下表 2.3 所示。（提示：此实验任务需要根据 LU 的功能要求，修改实验一中设计的多功能门电路模块的功能。）

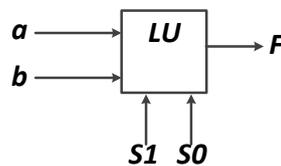


图 2.4 多功能逻辑运算单元 LU 原理框图

表 2.3 多功能逻辑运算单元 LU 真值表

s1	s0	Y
0	0	$a \cdot b$
0	1	$a+b$
1	0	$a \oplus b$
1	1	0

六、实验数据处理及其他要求

- 1、编写 Verilog HDL 源代码，完成电路建模。
- 2、编译文件并查看 RTL 视图，说明其与原理框图的差异。
- 3、编写 Test Bench，进行仿真测试，观察输出波形，整理实验数据，分析实验结果验证电路功能是否满足设计目标。
- 4、编写硬件约束文件，利用实验板 EG01 验证电路功能。
- 5、分析实验中出现的的问题的原因。

3. 附录A Verilog 简明手册

Verilog Cheat Sheet ——— S Winberg and J Taylor

Comments

// One-liner

/* Multiple

lines */

Numeric Constants

// The 8-bit decimal number 106:

8'b_0110_1010 // Binary

8'o_152 // Octal

8'd_106 // Decimal

8'h_6A // Hexadecimal

"j" // ASCII

78'bZ // 78-bit high-impedance

Too short constants are padded with zeros on the left. Too long constants are truncated from the left.

Nets and Variables

wire [3:0]w; // Assign outside always blocks

reg [1:7]r; // Assign inside always blocks

reg [7:0]mem[31:0];

integer j; // Compile-time variable

genvar k; // Generate variable

Parameters

parameter N = 8;

localparam State = 2'd3;

Assignments

assign Output = A * B;

assign {C, D} = {D[5:2], C[1:9], E};

Operators

// These are in order of precedence...

// Select

A[N] A[N:M]

// Reduction

&A ~&A |A ~|A ^A ~^A

// Compliment

!A ~A

// Unary

+A -A

// Concatenate

{A, ..., B}

// Replicate

{N{A}}

// Arithmetic

A*B A/B A%B

A+B A-B

// Shift

A<<B A>>B

// Relational

A>B A<B A>=B A<=B

A==B A!=B

// Bit-wise

A&B

A^B A~^B

A|B

// Logical

A&&B

A||B

// Conditional

A ? B : C

Module

module MyModule

#(parameter N = 8) // Optional parameter

(input Reset, Clk,

output [N-1:0]Output);

// Module implementation

endmodule

Module Instantiation

// Override default parameter: setting N = 13

MyModule #(13) MyModule1(Reset, Clk, Result);

Case

```
always @(*) begin
  case(Mux)
    2'd0: A = 8'd9;
    2'd1,
    2'd3: A = 8'd103;
    2'd2: A = 8'd2;
  default::;
  endcase
end
always @(*) begin
  casex(Decoded)
    4'b1xxx: Encoded = 2'd0;
    4'b01xx: Encoded = 2'd1;
    4'b001x: Encoded = 2'd2;
    4'b0001: Encoded = 2'd3;
  default: Encoded = 2'd0;
  endcase
end
```

Synchronous

```
always @(posedge Clk) begin
  if(Reset) B <= 0;
  else B <= B + 1'b1;
end
```

Loop

```
always @(*) begin
  Count = 0;
  for(j = 0; j < 8; j = j+1)
    Count = Count + Input[j];
end
```

Function

```
function [6:0]F;
  input [3:0]A;
  input [2:0]B;
  begin
    F = {A+1'b1, B+2'd2};
  end
endfunction
```

Generate

```
genvar j;
wire [12:0]Output[19:0];
generate
  for(j = 0; j < 20; j = j+1)
  begin: Gen_Modules
    MyModule #(13) MyModule_Instance(
      Reset, Clk,
      Output[j]
    );
  end
endgenerate
```

State Machine

```
reg [1:0]State;
localparam Start = 2'b00;
localparam Idle = 2'b01;
localparam Work = 2'b11;
localparam Done = 2'b10;

reg tReset;

always @(posedge Clk) begin
  tReset <= Reset;
  if(tReset) begin
    State <= Start;
  end else begin
    case(State)
      Start: begin
        State <= Idle;
      end
      Idle: begin
        State <= Work;
      end
      Work: begin
        State <= Done;
      end
      Done: begin
        State <= Idle;
      end
    default::;
  endcase
end
end
```

4. 附录 B EGO1 板硬件约束资源

Vivado 的约束文件编写采用 Tcl 语言规则。Tcl（读作 tickle）诞生于 80 年代的加州大学伯克利分校，作为一种简单高效可移植性好的脚本语言，目前已经广泛应用在几乎所有的 EDA 工具中。Tcl 的最大特点就是其语法格式极其简单甚至可以说僵化，采用纯粹的 [命令 选项 参数] 形式，是名副其实的“工具命令语言”（Tcl 的全称 Tool Command Language）。

```
set myVar "Hello World!" //设置一个名为 myVar 的变量，其值为 Hello World!
```

在 Vivado 中使用 Tcl 最基本的场景就是对网表上的目标进行遍历、查找和定位，这也是对网表上的目标进行约束的基础。要掌握这些则首先需要理解 Vivado 对目标的分类。

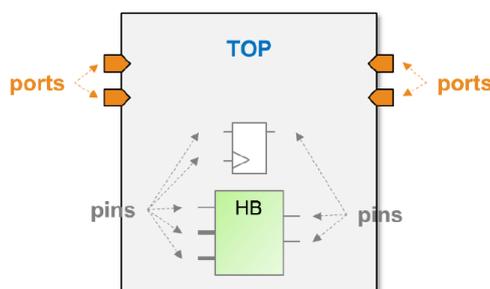


图 4.1

如上图所示，设计顶层的 I/O 称作 ports，其余底层模块或是门级网表上的元件端口都称作 pins。而包括顶层在内的各级模块，blackbox 以及门级元件，都称作 cells。连线称作 nets，加上 XDC 中定义的 clocks，在 Vivado 中一共将网表文件中的目标定义为五类。要选取这五类目标，则需用相应的 get_*命令，例如 get_pins 等等。

Tcl 在搜索网表中的目标时，除了上述根据名字条件直接搜索的方式，还可以利用目标间的关系，使用 -of_objects（Tcl 中可以简写为 -of）来间接搜索特定目标。Vivado 中定义的五类目标间关系如下图所示。

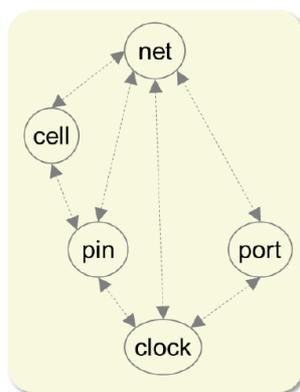


图 4.2

以下图的设计来举例，

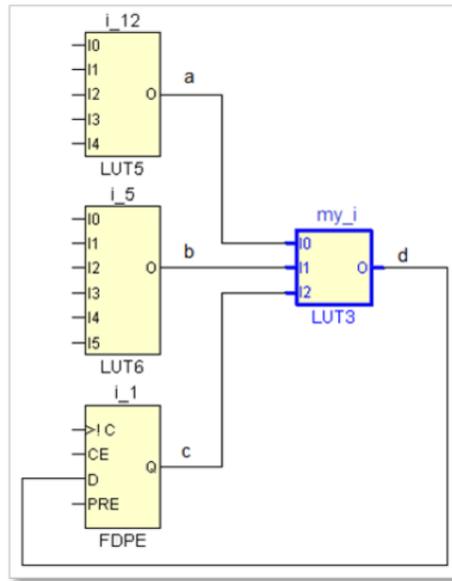


图 4.3

```
get_pins -of [get_cells my_i]           //返回 my_i/I0 my_i/I1 my_i/I2 my_i/O
get_nets -of [get_cells my_i]         //返回 a b c d
get_cells -of [get_nets -of [get_cells my_i]] //返回 i_12 i_5 i_1 my_i
```

需要注意的是：

XDC 在本质上就是 Tcl 语言，但其仅支持基本的 Tcl 语法如变量、列表和运算符等等，对其它复杂的循环以及文件 I/O 等语法可以通过在 Vivado 中 source 一个 Tcl 文件的方式来补充。XDC 的基本语法可以分为时钟约束、I/O 约束以及时序例外约束三大类。

以下为实验板自带的系统引脚 I/O 约束，设计者可以结合自己的电路修改引脚命名。

////////////////////////////////////系统时钟和复位////////////////////////////////////

```
set_property -dict {PACKAGE_PIN P17 IOSTANDARD LVCMOS33} [get_ports sys_clk_in ]
set_property -dict {PACKAGE_PIN P15 IOSTANDARD LVCMOS33} [get_ports sys_rst_n ]
```

////////////////////////////////////串口////////////////////////////////////

```
set_property -dict {PACKAGE_PIN N5 IOSTANDARD LVCMOS33} [get_ports PC_Uart_rxd]
set_property -dict {PACKAGE_PIN T4 IOSTANDARD LVCMOS33} [get_ports PC_Uart_txd]
```

////////////////////////////////////蓝牙////////////////////////////////////

```
set_property -dict {PACKAGE_PIN L3 IOSTANDARD LVCMOS33} [get_ports BT_Uart_rxd]
set_property -dict {PACKAGE_PIN N2 IOSTANDARD LVCMOS33} [get_ports BT_Uart_txd]
```

```
set_property -dict {PACKAGE_PIN D18 IOSTANDARD LVCMOS33} [get_ports {bt_ctrl_o[0]}]
set_property -dict {PACKAGE_PIN M2 IOSTANDARD LVCMOS33} [get_ports {bt_ctrl_o[1]}]
set_property -dict {PACKAGE_PIN H15 IOSTANDARD LVCMOS33} [get_ports {bt_ctrl_o[2]}]
```

```
set_property -dict {PACKAGE_PIN C16 IOSTANDARD LVCMOS33} [get_ports {bt_ctrl_o[3]}]
set_property -dict {PACKAGE_PIN E18 IOSTANDARD LVCMOS33} [get_ports {bt_ctrl_o[4]}]
```

```
set_property -dict {PACKAGE_PIN C17 IOSTANDARD LVCMOS33} [get_ports bt_mcu_int_i]
```

```
//////////音频接口//////////
```

```
set_property -dict {PACKAGE_PIN T1 IOSTANDARD LVCMOS33} [get_ports audio_pwm_o]
set_property -dict {PACKAGE_PIN M6 IOSTANDARD LVCMOS33} [get_ports audio_sd_o]
```

```
//////////iic//////////
```

```
set_property -dict {PACKAGE_PIN F18 IOSTANDARD LVCMOS33} [get_ports pw_iic_scl_io]
set_property -dict {PACKAGE_PIN G18 IOSTANDARD LVCMOS33} [get_ports pw_iic_sda_io]
```

```
//////////XADC 模数转换//////////
```

```
set_property -dict {PACKAGE_PIN B12 IOSTANDARD LVCMOS33} [get_ports XADC_AUX_v_n ]
set_property -dict {PACKAGE_PIN C12 IOSTANDARD LVCMOS33} [get_ports XADC_AUX_v_p ]
set_property -dict {PACKAGE_PIN K9 IOSTANDARD LVCMOS33} [get_ports XADC_VP_VN_v_n]
set_property -dict {PACKAGE_PIN J10 IOSTANDARD LVCMOS33} [get_ports XADC_VP_VN_v_p]
```

```
////////// 5 个按键//////////位置以印刷板丝印为准//////////
```

```
set_property -dict {PACKAGE_PIN R11 IOSTANDARD LVCMOS33} [get_ports {btn_pin[0]}]
set_property -dict {PACKAGE_PIN R17 IOSTANDARD LVCMOS33} [get_ports {btn_pin[1]}]
set_property -dict {PACKAGE_PIN R15 IOSTANDARD LVCMOS33} [get_ports {btn_pin[2]}]
set_property -dict {PACKAGE_PIN V1 IOSTANDARD LVCMOS33} [get_ports {btn_pin[3]}]
set_property -dict {PACKAGE_PIN U4 IOSTANDARD LVCMOS33} [get_ports {btn_pin[4]}]
```

```
//////////拨码开关 sw0~sw7//////////位置以印刷板丝印为准//////////
```

```
set_property -dict {PACKAGE_PIN P5 IOSTANDARD LVCMOS33} [get_ports {sw_pin[0]}]
set_property -dict {PACKAGE_PIN P4 IOSTANDARD LVCMOS33} [get_ports {sw_pin[1]}]
set_property -dict {PACKAGE_PIN P3 IOSTANDARD LVCMOS33} [get_ports {sw_pin[2]}]
set_property -dict {PACKAGE_PIN P2 IOSTANDARD LVCMOS33} [get_ports {sw_pin[3]}]
set_property -dict {PACKAGE_PIN R2 IOSTANDARD LVCMOS33} [get_ports {sw_pin[4]}]
set_property -dict {PACKAGE_PIN M4 IOSTANDARD LVCMOS33} [get_ports {sw_pin[5]}]
set_property -dict {PACKAGE_PIN N4 IOSTANDARD LVCMOS33} [get_ports {sw_pin[6]}]
set_property -dict {PACKAGE_PIN R1 IOSTANDARD LVCMOS33} [get_ports {sw_pin[7]}]
```

```
//////////拨码开关 sw8~sw15//////////位置以印刷板丝印为准//////////
```

```
set_property -dict {PACKAGE_PIN U3 IOSTANDARD LVCMOS33} [get_ports {dip_pin[0]}]
set_property -dict {PACKAGE_PIN U2 IOSTANDARD LVCMOS33} [get_ports {dip_pin[1]}]
set_property -dict {PACKAGE_PIN V2 IOSTANDARD LVCMOS33} [get_ports {dip_pin[2]}]
set_property -dict {PACKAGE_PIN V5 IOSTANDARD LVCMOS33} [get_ports {dip_pin[3]}]
set_property -dict {PACKAGE_PIN V4 IOSTANDARD LVCMOS33} [get_ports {dip_pin[4]}]
set_property -dict {PACKAGE_PIN R3 IOSTANDARD LVCMOS33} [get_ports {dip_pin[5]}]
set_property -dict {PACKAGE_PIN T3 IOSTANDARD LVCMOS33} [get_ports {dip_pin[6]}]
```

set_property -dict {PACKAGE_PIN T5 IOSTANDARD LVCMOS33} [get_ports {dip_pin[7]}]

//////////////////////////////////LED0~LED15//////////////////////////////////位置以印刷板丝印为准//////////////////////////////////

set_property -dict {PACKAGE_PIN F6 IOSTANDARD LVCMOS33} [get_ports {led_pin[0]}]
set_property -dict {PACKAGE_PIN G4 IOSTANDARD LVCMOS33} [get_ports {led_pin[1]}]
set_property -dict {PACKAGE_PIN G3 IOSTANDARD LVCMOS33} [get_ports {led_pin[2]}]
set_property -dict {PACKAGE_PIN J4 IOSTANDARD LVCMOS33} [get_ports {led_pin[3]}]
set_property -dict {PACKAGE_PIN H4 IOSTANDARD LVCMOS33} [get_ports {led_pin[4]}]
set_property -dict {PACKAGE_PIN J3 IOSTANDARD LVCMOS33} [get_ports {led_pin[5]}]
set_property -dict {PACKAGE_PIN J2 IOSTANDARD LVCMOS33} [get_ports {led_pin[6]}]
set_property -dict {PACKAGE_PIN K2 IOSTANDARD LVCMOS33} [get_ports {led_pin[7]}]

set_property -dict {PACKAGE_PIN K1 IOSTANDARD LVCMOS33} [get_ports {led_pin[8]}]
set_property -dict {PACKAGE_PIN H6 IOSTANDARD LVCMOS33} [get_ports {led_pin[9]}]
set_property -dict {PACKAGE_PIN H5 IOSTANDARD LVCMOS33} [get_ports {led_pin[10]}]
set_property -dict {PACKAGE_PIN J5 IOSTANDARD LVCMOS33} [get_ports {led_pin[11]}]
set_property -dict {PACKAGE_PIN K6 IOSTANDARD LVCMOS33} [get_ports {led_pin[12]}]
set_property -dict {PACKAGE_PIN L1 IOSTANDARD LVCMOS33} [get_ports {led_pin[13]}]
set_property -dict {PACKAGE_PIN M1 IOSTANDARD LVCMOS33} [get_ports {led_pin[14]}]
set_property -dict {PACKAGE_PIN K3 IOSTANDARD LVCMOS33} [get_ports {led_pin[15]}]

//////////////////////////////////8 个数码管位选信号//////////////////////////////////

set_property -dict {PACKAGE_PIN G2 IOSTANDARD LVCMOS33} [get_ports {seg_cs_pin[0]}]
set_property -dict {PACKAGE_PIN C2 IOSTANDARD LVCMOS33} [get_ports {seg_cs_pin[1]}]
set_property -dict {PACKAGE_PIN C1 IOSTANDARD LVCMOS33} [get_ports {seg_cs_pin[2]}]
set_property -dict {PACKAGE_PIN H1 IOSTANDARD LVCMOS33} [get_ports {seg_cs_pin[3]}]
set_property -dict {PACKAGE_PIN G1 IOSTANDARD LVCMOS33} [get_ports {seg_cs_pin[4]}]
set_property -dict {PACKAGE_PIN F1 IOSTANDARD LVCMOS33} [get_ports {seg_cs_pin[5]}]
set_property -dict {PACKAGE_PIN E1 IOSTANDARD LVCMOS33} [get_ports {seg_cs_pin[6]}]
set_property -dict {PACKAGE_PIN G6 IOSTANDARD LVCMOS33} [get_ports {seg_cs_pin[7]}]

//////////////////////////////////数码管段选信号//////////////////////////////////

set_property -dict {PACKAGE_PIN B4 IOSTANDARD LVCMOS33} [get_ports {seg_data_0_pin[0]}]
set_property -dict {PACKAGE_PIN A4 IOSTANDARD LVCMOS33} [get_ports {seg_data_0_pin[1]}]
set_property -dict {PACKAGE_PIN A3 IOSTANDARD LVCMOS33} [get_ports {seg_data_0_pin[2]}]
set_property -dict {PACKAGE_PIN B1 IOSTANDARD LVCMOS33} [get_ports {seg_data_0_pin[3]}]
set_property -dict {PACKAGE_PIN A1 IOSTANDARD LVCMOS33} [get_ports {seg_data_0_pin[4]}]
set_property -dict {PACKAGE_PIN B3 IOSTANDARD LVCMOS33} [get_ports {seg_data_0_pin[5]}]
set_property -dict {PACKAGE_PIN B2 IOSTANDARD LVCMOS33} [get_ports {seg_data_0_pin[6]}]
set_property -dict {PACKAGE_PIN D5 IOSTANDARD LVCMOS33} [get_ports {seg_data_0_pin[7]}]

set_property -dict {PACKAGE_PIN D4 IOSTANDARD LVCMOS33} [get_ports {seg_data_1_pin[0]}]
set_property -dict {PACKAGE_PIN E3 IOSTANDARD LVCMOS33} [get_ports {seg_data_1_pin[1]}]

```

set_property -dict {PACKAGE_PIN D3 IOSTANDARD LVCMOS33} [get_ports {seg_data_1_pin[2]}]
set_property -dict {PACKAGE_PIN F4 IOSTANDARD LVCMOS33} [get_ports {seg_data_1_pin[3]}]
set_property -dict {PACKAGE_PIN F3 IOSTANDARD LVCMOS33} [get_ports {seg_data_1_pin[4]}]
set_property -dict {PACKAGE_PIN E2 IOSTANDARD LVCMOS33} [get_ports {seg_data_1_pin[5]}]
set_property -dict {PACKAGE_PIN D2 IOSTANDARD LVCMOS33} [get_ports {seg_data_1_pin[6]}]
set_property -dict {PACKAGE_PIN H2 IOSTANDARD LVCMOS33} [get_ports {seg_data_1_pin[7]}]

```

//////////////////////////////////VGA 行同步场同步信号//////////////////////////////////

```

set_property -dict {PACKAGE_PIN D7 IOSTANDARD LVCMOS33} [get_ports vga_hs_pin]
set_property -dict {PACKAGE_PIN C4 IOSTANDARD LVCMOS33} [get_ports vga_vs_pin]

```

//////////////////////////////////VGA 红绿蓝信号//////////////////////////////////

```

set_property -dict {PACKAGE_PIN F5 IOSTANDARD LVCMOS33} [get_ports {vga_data_pin[0]}]
set_property -dict {PACKAGE_PIN C6 IOSTANDARD LVCMOS33} [get_ports {vga_data_pin[1]}]
set_property -dict {PACKAGE_PIN C5 IOSTANDARD LVCMOS33} [get_ports {vga_data_pin[2]}]
set_property -dict {PACKAGE_PIN B7 IOSTANDARD LVCMOS33} [get_ports {vga_data_pin[3]}]
set_property -dict {PACKAGE_PIN B6 IOSTANDARD LVCMOS33} [get_ports {vga_data_pin[4]}]
set_property -dict {PACKAGE_PIN A6 IOSTANDARD LVCMOS33} [get_ports {vga_data_pin[5]}]
set_property -dict {PACKAGE_PIN A5 IOSTANDARD LVCMOS33} [get_ports {vga_data_pin[6]}]
set_property -dict {PACKAGE_PIN D8 IOSTANDARD LVCMOS33} [get_ports {vga_data_pin[7]}]
set_property -dict {PACKAGE_PIN C7 IOSTANDARD LVCMOS33} [get_ports {vga_data_pin[8]}]
set_property -dict {PACKAGE_PIN E6 IOSTANDARD LVCMOS33} [get_ports {vga_data_pin[9]}]
set_property -dict {PACKAGE_PIN E5 IOSTANDARD LVCMOS33} [get_ports {vga_data_pin[10]}]
set_property -dict {PACKAGE_PIN E7 IOSTANDARD LVCMOS33} [get_ports {vga_data_pin[11]}]

```

//////////////////////////////////DAC 数模转换//////////////////////////////////

```

set_property -dict {PACKAGE_PIN R5 IOSTANDARD LVCMOS33} [get_ports dac_ile]
set_property -dict {PACKAGE_PIN N6 IOSTANDARD LVCMOS33} [get_ports dac_cs_n]
set_property -dict {PACKAGE_PIN V6 IOSTANDARD LVCMOS33} [get_ports dac_wr1_n]
set_property -dict {PACKAGE_PIN R6 IOSTANDARD LVCMOS33} [get_ports dac_wr2_n]
set_property -dict {PACKAGE_PIN V7 IOSTANDARD LVCMOS33} [get_ports dac_xfer_n]

```

```

set_property -dict {PACKAGE_PIN T8 IOSTANDARD LVCMOS33} [get_ports {dac_data[0]}]
set_property -dict {PACKAGE_PIN R8 IOSTANDARD LVCMOS33} [get_ports {dac_data[1]}]
set_property -dict {PACKAGE_PIN T6 IOSTANDARD LVCMOS33} [get_ports {dac_data[2]}]
set_property -dict {PACKAGE_PIN R7 IOSTANDARD LVCMOS33} [get_ports {dac_data[3]}]
set_property -dict {PACKAGE_PIN U6 IOSTANDARD LVCMOS33} [get_ports {dac_data[4]}]
set_property -dict {PACKAGE_PIN U7 IOSTANDARD LVCMOS33} [get_ports {dac_data[5]}]
set_property -dict {PACKAGE_PIN V9 IOSTANDARD LVCMOS33} [get_ports {dac_data[6]}]
set_property -dict {PACKAGE_PIN U9 IOSTANDARD LVCMOS33} [get_ports {dac_data[7]}]

```

//////////////////////////////////PS2//////////////////////////////////

```

set_property -dict {PACKAGE_PIN K5 IOSTANDARD LVCMOS33} [get_ports ps2_clk ]
set_property -dict {PACKAGE_PIN L4 IOSTANDARD LVCMOS33} [get_ports ps2_data ]

```

//////////////////////////////////SDRAM//////////////////////////////////

```
set_property -dict {PACKAGE_PIN L15 IOSTANDARD LVCMOS33} [get_ports {sram_addr[18]}]
set_property -dict {PACKAGE_PIN L16 IOSTANDARD LVCMOS33} [get_ports {sram_addr[17]}]
set_property -dict {PACKAGE_PIN L18 IOSTANDARD LVCMOS33} [get_ports {sram_addr[16]}]
set_property -dict {PACKAGE_PIN M18 IOSTANDARD LVCMOS33} [get_ports {sram_addr[15]}]
set_property -dict {PACKAGE_PIN R12 IOSTANDARD LVCMOS33} [get_ports {sram_addr[14]}]
set_property -dict {PACKAGE_PIN R13 IOSTANDARD LVCMOS33} [get_ports {sram_addr[13]}]
set_property -dict {PACKAGE_PIN M13 IOSTANDARD LVCMOS33} [get_ports {sram_addr[12]}]
set_property -dict {PACKAGE_PIN R18 IOSTANDARD LVCMOS33} [get_ports {sram_addr[11]}]
set_property -dict {PACKAGE_PIN T18 IOSTANDARD LVCMOS33} [get_ports {sram_addr[10]}]
set_property -dict {PACKAGE_PIN N14 IOSTANDARD LVCMOS33} [get_ports {sram_addr[9]}]
set_property -dict {PACKAGE_PIN P14 IOSTANDARD LVCMOS33} [get_ports {sram_addr[8]}]
set_property -dict {PACKAGE_PIN N17 IOSTANDARD LVCMOS33} [get_ports {sram_addr[7]}]
set_property -dict {PACKAGE_PIN P18 IOSTANDARD LVCMOS33} [get_ports {sram_addr[6]}]
set_property -dict {PACKAGE_PIN M16 IOSTANDARD LVCMOS33} [get_ports {sram_addr[5]}]
set_property -dict {PACKAGE_PIN M17 IOSTANDARD LVCMOS33} [get_ports {sram_addr[4]}]
set_property -dict {PACKAGE_PIN N15 IOSTANDARD LVCMOS33} [get_ports {sram_addr[3]}]
set_property -dict {PACKAGE_PIN N16 IOSTANDARD LVCMOS33} [get_ports {sram_addr[2]}]
set_property -dict {PACKAGE_PIN T14 IOSTANDARD LVCMOS33} [get_ports {sram_addr[1]}]
set_property -dict {PACKAGE_PIN T15 IOSTANDARD LVCMOS33} [get_ports {sram_addr[0]}]
```

```
set_property -dict {PACKAGE_PIN V15 IOSTANDARD LVCMOS33} [get_ports sram_ce_n]
set_property -dict {PACKAGE_PIN R10 IOSTANDARD LVCMOS33} [get_ports sram_lb_n]
set_property -dict {PACKAGE_PIN T16 IOSTANDARD LVCMOS33} [get_ports sram_oe_n]
set_property -dict {PACKAGE_PIN R16 IOSTANDARD LVCMOS33} [get_ports sram_ub_n]
set_property -dict {PACKAGE_PIN V16 IOSTANDARD LVCMOS33} [get_ports sram_we_n]
```

```
set_property -dict {PACKAGE_PIN T10 IOSTANDARD LVCMOS33} [get_ports {sram_data[15]}]
set_property -dict {PACKAGE_PIN T9 IOSTANDARD LVCMOS33} [get_ports {sram_data[14]}]
set_property -dict {PACKAGE_PIN U13 IOSTANDARD LVCMOS33} [get_ports {sram_data[13]}]
set_property -dict {PACKAGE_PIN T13 IOSTANDARD LVCMOS33} [get_ports {sram_data[12]}]
set_property -dict {PACKAGE_PIN V14 IOSTANDARD LVCMOS33} [get_ports {sram_data[11]}]
set_property -dict {PACKAGE_PIN U14 IOSTANDARD LVCMOS33} [get_ports {sram_data[10]}]
set_property -dict {PACKAGE_PIN V11 IOSTANDARD LVCMOS33} [get_ports {sram_data[9]}]
set_property -dict {PACKAGE_PIN V10 IOSTANDARD LVCMOS33} [get_ports {sram_data[8]}]
set_property -dict {PACKAGE_PIN V12 IOSTANDARD LVCMOS33} [get_ports {sram_data[7]}]
set_property -dict {PACKAGE_PIN U12 IOSTANDARD LVCMOS33} [get_ports {sram_data[6]}]
set_property -dict {PACKAGE_PIN U11 IOSTANDARD LVCMOS33} [get_ports {sram_data[5]}]
set_property -dict {PACKAGE_PIN T11 IOSTANDARD LVCMOS33} [get_ports {sram_data[4]}]
set_property -dict {PACKAGE_PIN V17 IOSTANDARD LVCMOS33} [get_ports {sram_data[3]}]
set_property -dict {PACKAGE_PIN U16 IOSTANDARD LVCMOS33} [get_ports {sram_data[2]}]
set_property -dict {PACKAGE_PIN U18 IOSTANDARD LVCMOS33} [get_ports {sram_data[1]}]
```

```
set_property -dict {PACKAGE_PIN U17 IOSTANDARD LVCMOS33} [get_ports {sram_data[0]}]
```

```
////////////////////////////////////32 个 pmod 接口////////////////////////////////////
```

```
set_property -dict {PACKAGE_PIN B16 IOSTANDARD LVCMOS33} [get_ports {exp_io[0]} ]  
set_property -dict {PACKAGE_PIN A15 IOSTANDARD LVCMOS33} [get_ports {exp_io[1]} ]  
set_property -dict {PACKAGE_PIN A13 IOSTANDARD LVCMOS33} [get_ports {exp_io[2]} ]  
set_property -dict {PACKAGE_PIN B18 IOSTANDARD LVCMOS33} [get_ports {exp_io[3]} ]  
set_property -dict {PACKAGE_PIN F13 IOSTANDARD LVCMOS33} [get_ports {exp_io[4]} ]  
set_property -dict {PACKAGE_PIN B13 IOSTANDARD LVCMOS33} [get_ports {exp_io[5]} ]  
set_property -dict {PACKAGE_PIN D14 IOSTANDARD LVCMOS33} [get_ports {exp_io[6]} ]  
set_property -dict {PACKAGE_PIN B11 IOSTANDARD LVCMOS33} [get_ports {exp_io[7]} ]  
set_property -dict {PACKAGE_PIN E15 IOSTANDARD LVCMOS33} [get_ports {exp_io[8]} ]  
set_property -dict {PACKAGE_PIN D15 IOSTANDARD LVCMOS33} [get_ports {exp_io[9]} ]  
set_property -dict {PACKAGE_PIN H16 IOSTANDARD LVCMOS33} [get_ports {exp_io[10]} ]  
set_property -dict {PACKAGE_PIN F15 IOSTANDARD LVCMOS33} [get_ports {exp_io[11]} ]  
set_property -dict {PACKAGE_PIN H14 IOSTANDARD LVCMOS33} [get_ports {exp_io[12]} ]  
set_property -dict {PACKAGE_PIN E17 IOSTANDARD LVCMOS33} [get_ports {exp_io[13]} ]  
set_property -dict {PACKAGE_PIN K13 IOSTANDARD LVCMOS33} [get_ports {exp_io[14]} ]  
set_property -dict {PACKAGE_PIN H17 IOSTANDARD LVCMOS33} [get_ports {exp_io[15]} ]
```

```
set_property -dict {PACKAGE_PIN B17 IOSTANDARD LVCMOS33} [get_ports {exp_io[16]} ]  
set_property -dict {PACKAGE_PIN A16 IOSTANDARD LVCMOS33} [get_ports {exp_io[17]} ]  
set_property -dict {PACKAGE_PIN A14 IOSTANDARD LVCMOS33} [get_ports {exp_io[18]} ]  
set_property -dict {PACKAGE_PIN A18 IOSTANDARD LVCMOS33} [get_ports {exp_io[19]} ]  
set_property -dict {PACKAGE_PIN F14 IOSTANDARD LVCMOS33} [get_ports {exp_io[20]} ]  
set_property -dict {PACKAGE_PIN B14 IOSTANDARD LVCMOS33} [get_ports {exp_io[21]} ]  
set_property -dict {PACKAGE_PIN C14 IOSTANDARD LVCMOS33} [get_ports {exp_io[22]} ]  
set_property -dict {PACKAGE_PIN A11 IOSTANDARD LVCMOS33} [get_ports {exp_io[23]} ]  
set_property -dict {PACKAGE_PIN E16 IOSTANDARD LVCMOS33} [get_ports {exp_io[24]} ]  
set_property -dict {PACKAGE_PIN C15 IOSTANDARD LVCMOS33} [get_ports {exp_io[25]} ]  
set_property -dict {PACKAGE_PIN G16 IOSTANDARD LVCMOS33} [get_ports {exp_io[26]} ]  
set_property -dict {PACKAGE_PIN F16 IOSTANDARD LVCMOS33} [get_ports {exp_io[27]} ]  
set_property -dict {PACKAGE_PIN G14 IOSTANDARD LVCMOS33} [get_ports {exp_io[28]} ]  
set_property -dict {PACKAGE_PIN D17 IOSTANDARD LVCMOS33} [get_ports {exp_io[29]} ]  
set_property -dict {PACKAGE_PIN J13 IOSTANDARD LVCMOS33} [get_ports {exp_io[30]} ]  
set_property -dict {PACKAGE_PIN G17 IOSTANDARD LVCMOS33} [get_ports {exp_io[31]} ]
```

5. 附录 C 操作常见问题及注意事项

1、新建工程时没有选择合适的型号怎么办？

答：可以在 Flow Navigator 的 Project Manager 下 settings 中重新选择。

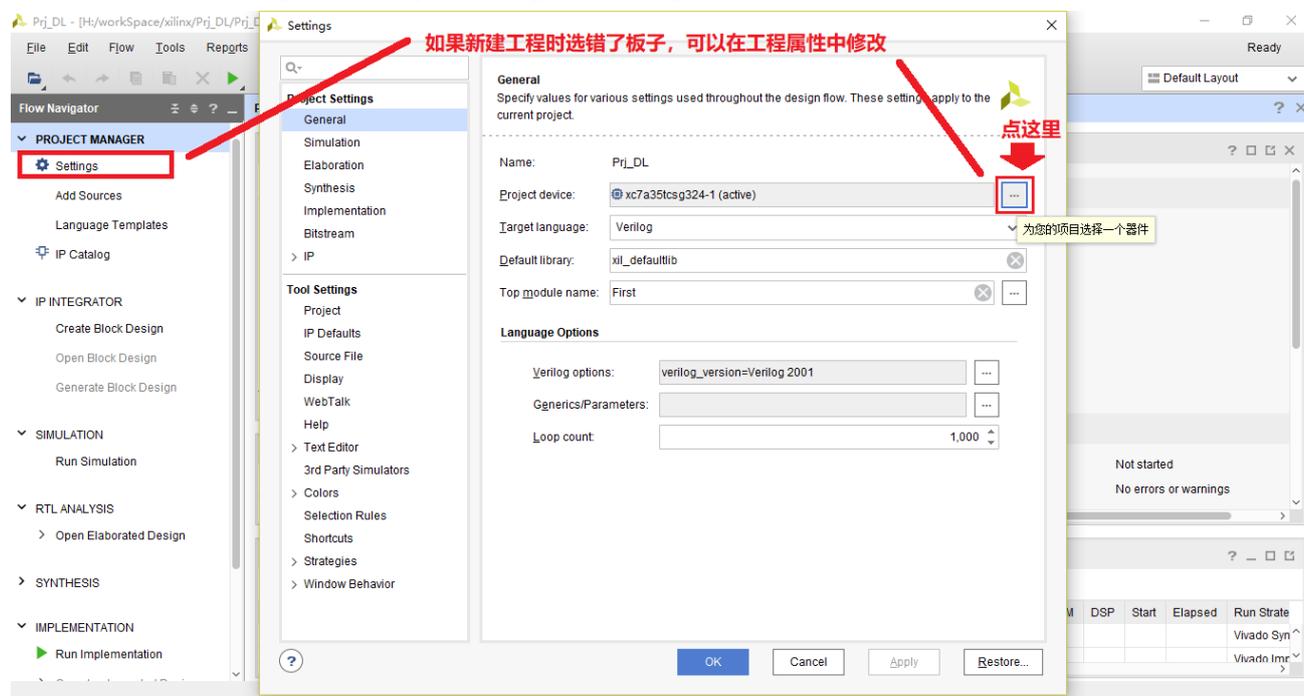


图 5.1