|  |
| --- |
| 电子科技大学计算机科学与工程学院 |
| 计算机组成原理 |
| [实验指导书] |
|  |
|  |
|  |

目 录

[前 言 1](#_Toc433292813)

[1.1 硬件环境 2](#_Toc433292814)

[1.2软件环境 3](#_Toc433292815)

[1.3实验主要仪器设备连接框图 3](#_Toc433292816)

[2. ISE Design Suite 14.7集成开发环境 4](#_Toc433292817)

[2．1 创建工程（Project） 4](#_Toc433292818)

[2．2 模块（Module）的创建 6](#_Toc433292819)

[2．3 仿真（Simulation） 8](#_Toc433292820)

[3. 下载到开发板 11](#_Toc433292821)

[3．1 约束（Constraints） 11](#_Toc433292822)

[3．2实现（Implementation） 11](#_Toc433292823)

[3．3流代码的生成与下载 12](#_Toc433292824)

[5. 附录：Spartan6/XC6SLX45开发板用户接口 13](#_Toc433292825)

# 前 言

《计算机组成原理》是计算机科学专业的一门重要专业基础课。在该课程中的理论学习中系统地阐述了计算机各组成部件的工作原理、逻辑实现和设计方法及将各部件连接成整机的方法，计算机硬件与底层软件的接口，培养了学生对计算机硬件系统的分析、开发与设计的基本技能能力。

计算机组成原理实验是学习《计算机组成原理》这门课程的重要组成部分，通过实验让同学们加深对所学知识的理解和掌握，培养和锻炼同学们的动手能力。

计算机组成原理实验硬件平台通常采用采用FPGA数字电路开发板，软件采用Verilog HDL硬件描述语言。

本实验指导书的目的是指导同学们学习和掌握FPGA数字电路开发板的使用、Verilog HDL硬件描述语言开发具体电路部件的一般步骤和方法。

本实验的先修课程：《数字逻辑》及《Verilog HDL硬件描述语言》。

对于未修先修课程的学生实验指导教师应对其进行相关知识的培训，培训建议4学时。

**1. 实验开发环境**

## 1.1 硬件环境

本实验所需的硬件主要有：PC计算机和FFPEG实验开发板，具体要要求如下：

1. IBM PC及兼容计算机：内存≥1GB，硬盘空间≥30GB；
2. FPGA数字电路开发平台： Anvyl（燧石TM）开发板，型号Spartan6-XC6SLX45；
3. USB连接线：用于软件下载。



图1-1 Anvyl（燧石TM）Spartan6-XC6SLX45开发板

Anvyl（燧石TM）Spartan6/XC6SLX45开发板的主要特点：

* 通用输入接口
* 14个LED：8个红色，2组“红绿灯”模式
* 8个滑动开关
* 4个按键
* 6个7段数码管
* 2组拨码开关（4个/组）
* 音频接口：I2S Codec接口
* 视频接口
* HDMI输出接口
* 12比特VGA接口（800×600），使用无源电阻网路
* USB接口
* USB1：JTAG编程及UART
* USB2/3：USB设备接口，可接HID设备或U盘
* 片外存储器

DDR2：256MB SRAM：2MB QSPI Falsh：32MB

* OLED：128×32蓝色显示
* LCD液晶屏：480×272全彩色液晶触摸屏
* Clock：100MHz单端CMOS晶振

注：

Anvyl（燧石TM）Spartan6/XC6SLX45开发板的用户接口参见附录1。

## 1.2软件环境

1. 操作系统：Windows XP、Windows 7；
2. 开发平台：Xilinx ISE Design Suite 14.7集成开发系统；
3. 下载软件：digilent.adept.system\_v2.10.2.exe（由FPGA开用板厂家提供，用于将Xilinx开发生成的流代码bit文件下载到FPGA开发板上）；
4. 编程语言：Verilog HDL硬件描述语言。

## 1.3实验主要仪器设备连接框图

实验主要仪器设备的连接框图如图1-2所示。

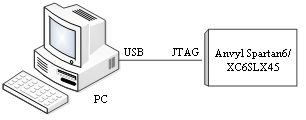


图2-1 主要实验仪器设备连接图

通过USB连接线将实验开发板与PC计算机连接，USB线的一端接在计算机上任意一USB接口上，另一端接在实验板的JTAG（Joint Test Action Group；联合测试行动小组）接口上。

# 2. ISE Design Suite 14.7集成开发环境

为了便于同学们使用本指导书，以设计一个交通信号的红绿灯控制为例（如图2-1所示），介绍ISE Design Suite 14.7集成开发环境的使用。

图2-1 红绿灯控制器

Switch0

Switch1

Red\_Led

Green\_Led

Yellow\_Led

R

G

Y

图2-1 红绿灯控制电路

该控制电路的功能：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Switch1** | **Switch0** | **Red\_Led** | **Green\_Led** | **Yellow\_Led** |
| 0 | 0 | 灭 | 亮 | 灭 |
| 0 | 1 | 亮 | 灭 | 灭 |
| 1 | X | 灭 | 灭 | 亮 |

具体实现时，各引脚与FPGA开发板上电路连接的关系如下：

* Switch0：SW0
* Switch1：SW0
* Red\_Led：LD9（红灯）
* Green\_Led：LD10（绿灯）
* Yellow\_Led：LD11（黄灯）

## 2．1 创建工程（Project）

启动ISE软件，然后选择菜单File→New Project，弹出New Project Wizard对话框，在对话框中输入工程名：Traffic\_Controller，并指定工作路径： D:\Computer\_Example\Traffic\_Controller，如图2-2所示。

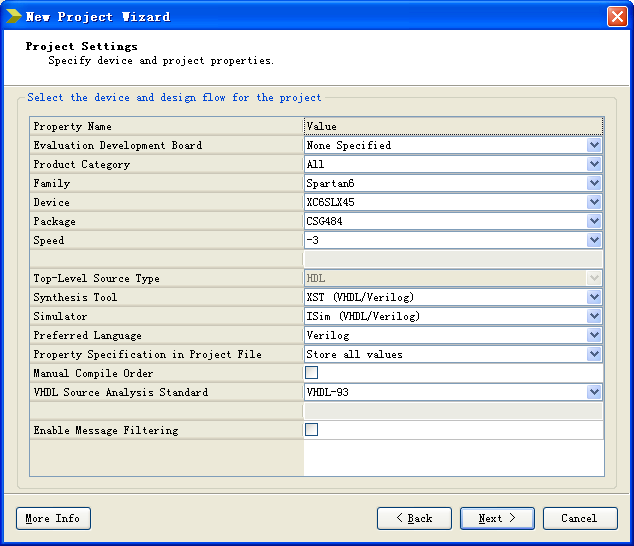


图2-2 New Project Wizard对话框

在图2-2中输入完工程名和工作路径后点击Next进入下一页：Project Settings，设置设备和工程属性，Family属性选用Spartan6 XC6SLX45芯片，Device属性选用CSG484封装，这是本实验开发板所用的芯片，如图2-3所示。

设置好后，在图2-3中点击Next进入下一页：Project Summary（如图2-4所示），这里显示了新建工程的信息，确认无误后，点击Finish按钮就可以建立一个新的工程了。

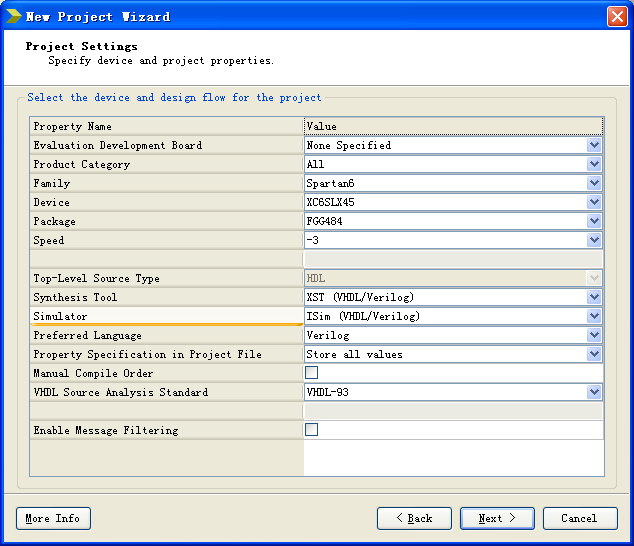


图2-3 Project Settings对话框

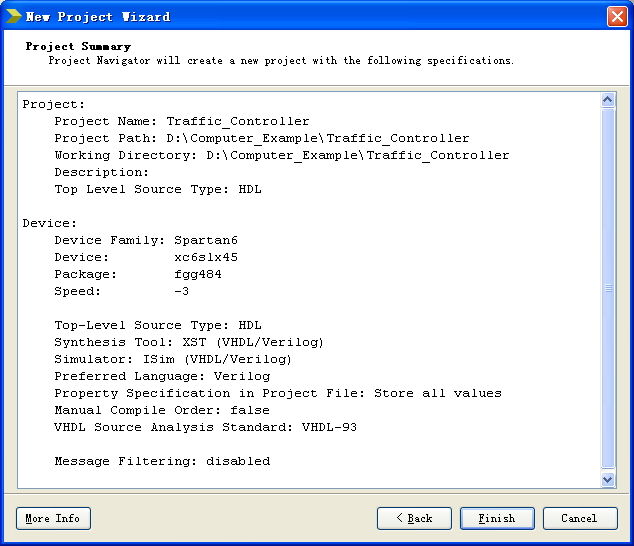
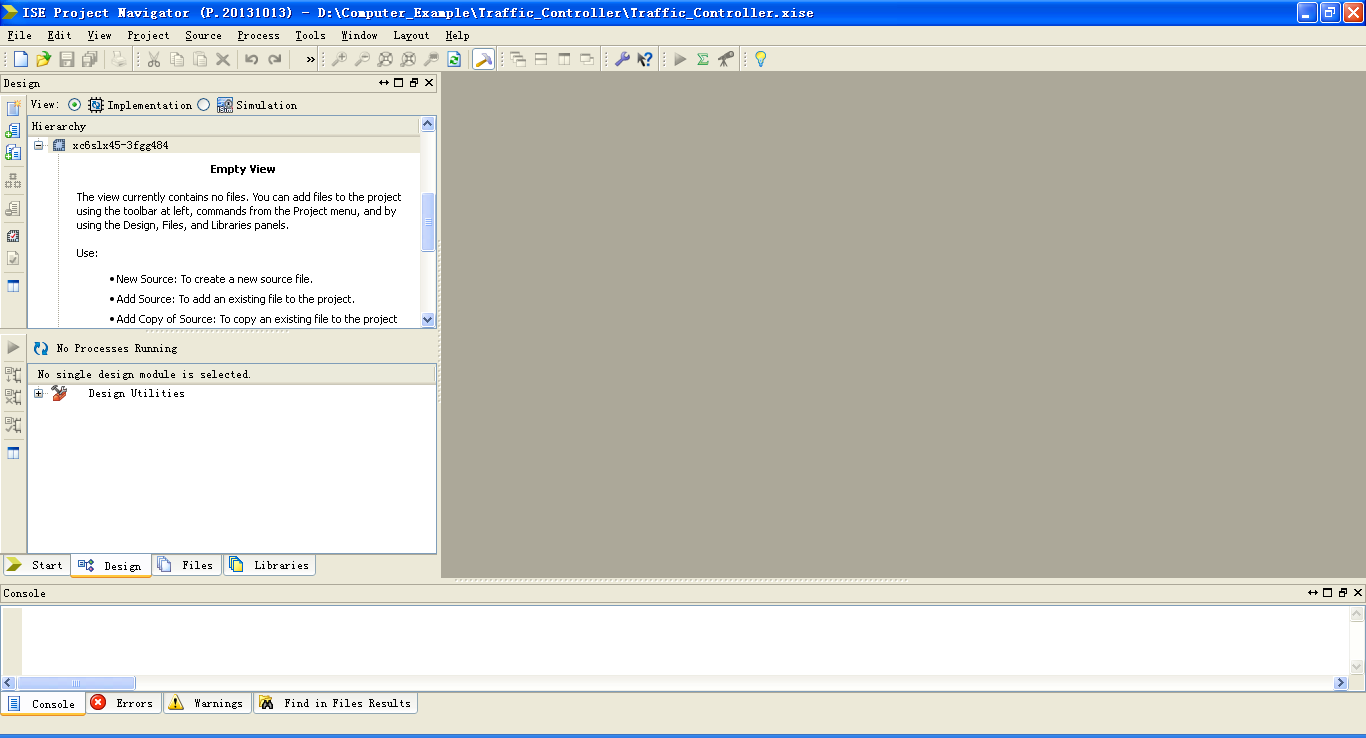


图2-4 Project Summary对话框

## 2．2 模块（Module）的创建

ISE集成开发环境如图2-5所示，主要分为4个区：工程管理区、过程管理区、源代码编辑区和信息显示区。



**信息显示区**

图2-5 ISE集成开发环境

在工程管理区任意位置单击鼠标右键，在弹出的菜单中选择New Source命令，会弹出如图2-6所示的New Source Wizard对话框：Select Source Type。

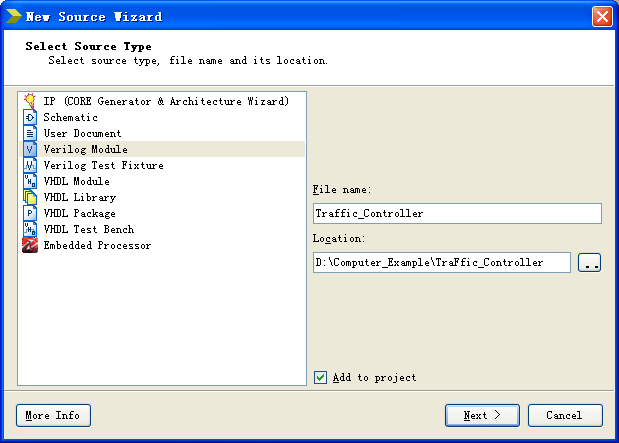


图2-6 Select Source Type对话框

在图2-6中选择Verilog Module，并输入Verilog文件名： Controller，代码存放位置（Location）为D:\Computer\_Example\Code，然后点击Next按钮进入端口定义对话框：Define Module，如图2-7所示。

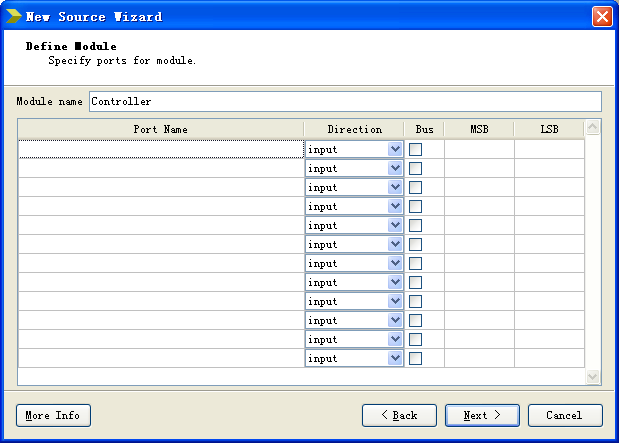


图2-7 Define Module对话框

在图2-7中， Module name栏用于输入模块名，这里是Controller，下面的列表框用于端口的定义。Port Name表示端口名称，Direction表示端口方向(可选择为input、output或inout)，MSB表示信号最高位，LSB表示信号最低位，对于单信号的MSB和LSB不用填写。端口定义这一步略过，在源程序中自行添加。定义完端口后，点击Next按钮进入下一步，然后点击Finish按钮完成模块创建。

此时开发环境的工程管理区中将有一个Controller（Controller.v）模块，双击该模块，然后在源代码编辑区编写实现该模块的代码：

module Controller(

input switch0,switch1,

output R,G,Y

);

assign Y = switch1;

assign R = switch0 & ~ switch1;

assign G = ~switch0 & ~ switch1;

endmodule

代码输入完成后，首先检查Verilog HDL语法：在工程管理区选中要检查的模块，在过程管理区双击Synthesize – XST→Check Syntax。如果有语法错误，会在信息显示区给出指示，请检查调试。如果没有语法错误，在模拟仿真前要进行Verilog HDL代码综合。

接下来进行综合（Synthesize）。所谓综合，就是将HDL语言、原理图等设计输入翻译成由与、或、非门和RAM、触发器等基本逻辑单元的逻辑连接（网表），并根据目标和要求（约束条件）优化生成的RTL（Real Time Logistics）层连接。在工程管理区的View中选择Implementation，并选中要综合的模块Controller，然后在过程管理区双击Synthesize-XST，就开始综合过程，如图2-8所示。

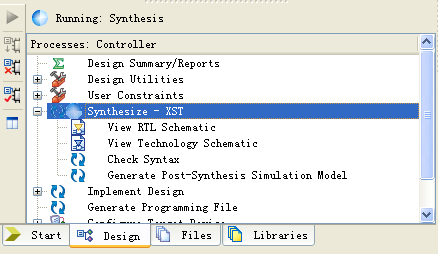


图2-8 模块（Module）的综合（Synthesize）

综合错误会在信息显示区给出指示，如果代码过于行为化而不能转换成电路，称为不可综合代码，表示ISE无法转换成对应的电路，需要修改你的设计代码。如果综合通过，表示你设计的代码可以转换成RTL层描述，称为可综合代码。你可以通过View RTL Schematic查看综合后的RTL层电路，如图2-9所示。

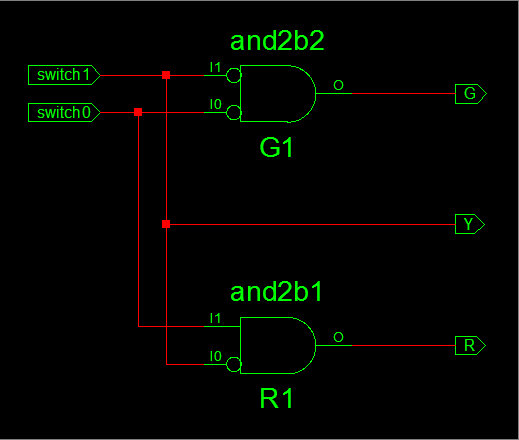


图2-9 模块Controller综合结果

在图2-9中显示出了综合后的电路图。在此可以比较该图与原设计是否一致，若不一此则说明模块设计代码有错，需对模块代码进行修改；若一致则可以进行后面的操作和设计。

## 2．3 仿真（Simulation）

仿真并不是设计过程必须的步骤。为了尽量减少设计中的错误，在将所做设计下载到开发板上进行板级验证时，对所做设计进行仿真是必要的。

在工程管理区任意位置单击鼠标右键，在弹出的菜单中选择New Source命令，会弹出如图2-10所示的New Source Wizard对话框：Select Source Type。在图2-10选择Verilog Test Fixture，输入测试文件名：Controller\_tb，单击Next按钮，进入下一个对话框，如图2-11所示。

在图2-11中工程中所有的模块名都会显示出来（由于目前工程中只有一个模块，故只显示了一个模块名），选择要测试的模块Controller，点击Next按钮，再点击Finish按钮，ISE会在源代码编辑区自动生成测试模块的代码，如图2-12所示。

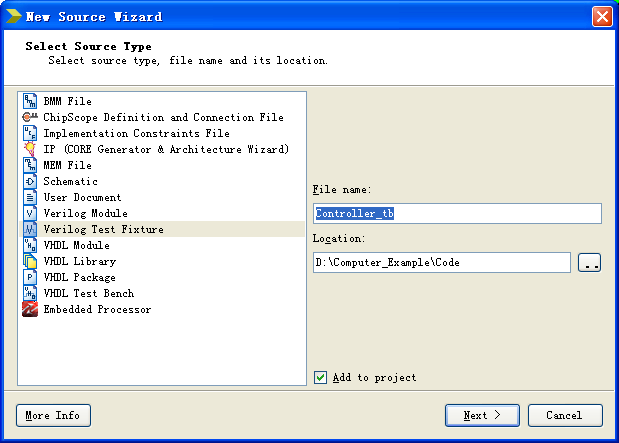
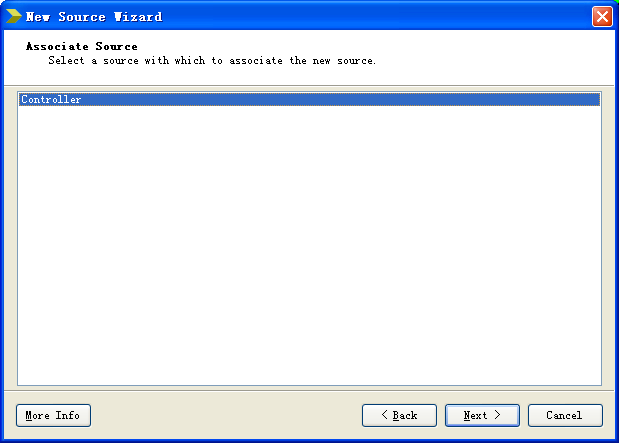
 

图2-10 创建仿真模块（1） 图2-11 创建仿真模块（2）

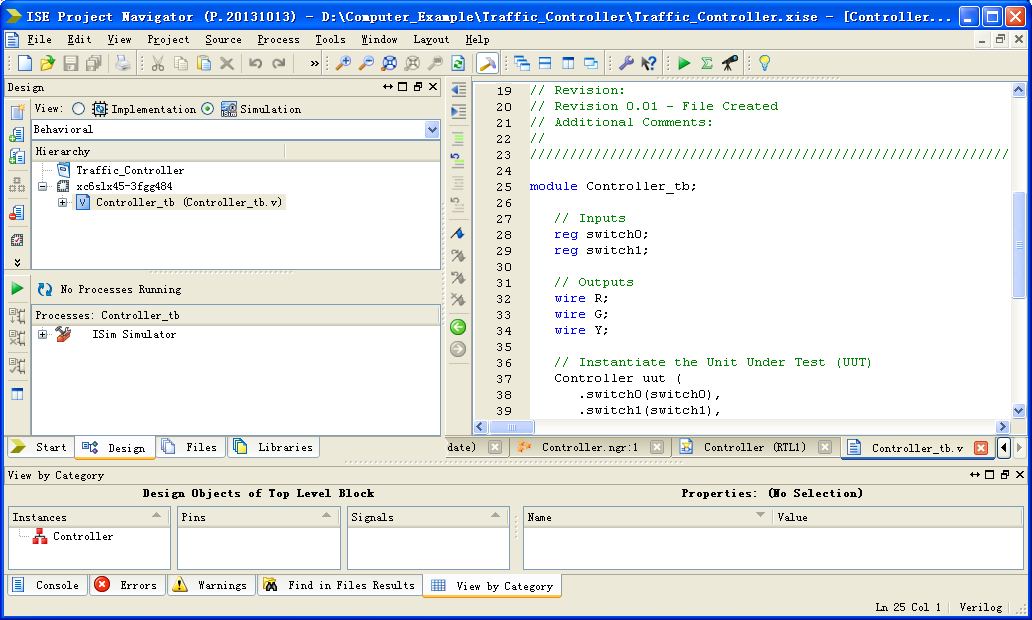


图2-12 Verilog Test Fixture – Controller\_tb

在图2-12中示了刚生成的Controller\_tb模块，在该模块中添加如下测试代码：

initial begin

// Initialize Inputs

switch0 = 0;

switch1 = 0;

// Wait 100 ns for global reset to finish

#100;

switch1 = 0; switch0 = 1;

#100;

switch1 = 1; switch0 = 0;

#100;

switch1 = 1; switch0 = 1;

end

完成测试文件编辑后，确认工程管理区中View选项设置为Simulation，并选中Controller\_tb模块。 这时在过程管理区会显示与仿真有关的进程，如图2-13所示。

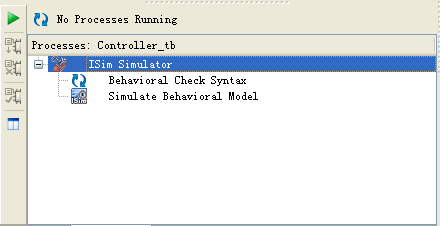


图2-13 选择仿真行为

在图2-13中右键单击其中的Simulate Behavioral Model项，选择弹出菜单中的Process Properties项，会弹出如图2-14所示的属性设置对话框，其中Simulation Run Time就是仿真时间的设置，可将其修改为任意时长，由于测试模块Controller\_tb中测试时间定义为300ns，故在图2-14仿真测试设置为600ns（注：大于测试时间即可）。

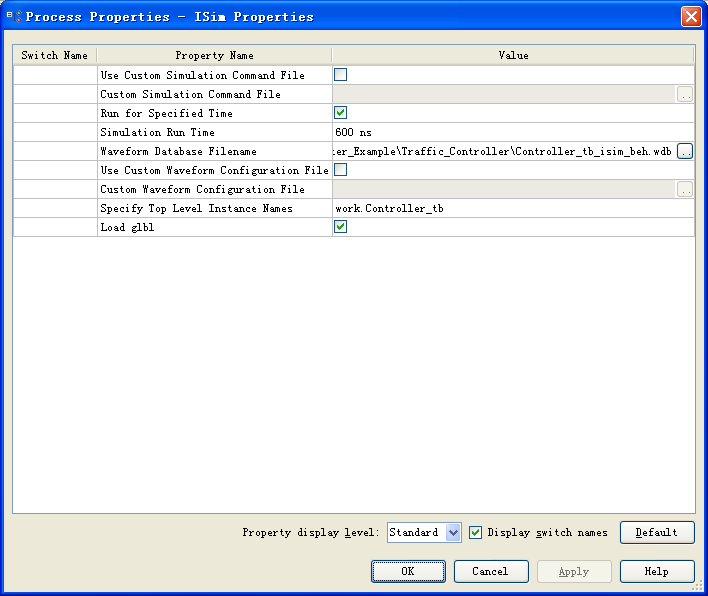


图2-14 Process Properties对话框

仿真参数设置完后，就可以进行仿真。首先在工程管理区选中测试代码，然后在过程管理区双击Simulate Behavioral Model，SE将启动ISE Simulator，可以得到仿真结果，如图2-15所示。

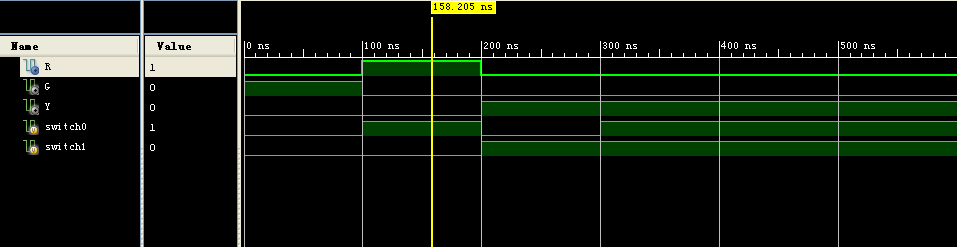


图2-15 模块Controller仿真效果

# 3. 下载到开发板

设计完成后，为了在FPGA开发板上验证所做设计，需要将所做设计下载到开发板运行。

## 3．1 约束（Constraints）

要实现设计，还需要为模块中的输入/输出信号添加管脚约束。这就需要在工程中添加UCF（User Constraints File）文件。在工程管理区单击鼠标右键，点击New Source，弹出如图2-6所示对话框，在类型中选择Implementation Constraints File，输入文件名：Controller\_ucf，然后依次点击“Next”按钮、“Finish”按钮，系统会生成一个空白的约束文件并打开。我们就可以为设计添加各种约束。在Controller\_ucf输入如下代码：

#switches

NET switch0 LOC = V5;

NET switch1 LOC = U4;

#Leds

NET R LOC = T8;

NET G LOC = R7;

NET Y LOC = U6;

约束完成后，下一个步骤就是实现(Implementation)。

## 3．2实现（Implementation）

所谓实现，是指将综合输出的逻辑网表翻译成所选器件的底层模块和硬件原语，将设计映射到器件结构上，进行布局布线，达到在选定器件上实现设计的目的。

在ISE中，执行实现过程，会自动执行翻译、映射和布局布线过程：也可单独执行。在过程管理区双击Implementation Design选项，就可以自动完成实现的3个步骤，如图2-16所示。

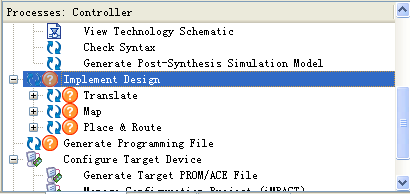


图2-16 模块实现过程

完成Place & Route后，实现过程就全部完成了。

## 3．3流代码的生成与下载

实现设计（Implementation Design）完成后，在图2-16中双击Generate Programming File项生成二进制比特文件Controller.bit。

将Anyvl开发板与计算机通过USB线连接好，USB连接线的一端连接到开发板的JTAG（开发板的J12接口），另一端连接到计算机的任意一USB接口即可。开发板加电，启动Digilent Adept软件，系统开始自动连接FPGA设备，成功检测到设备后，会显示出JTAG链上所用芯片，如图2-17所示。

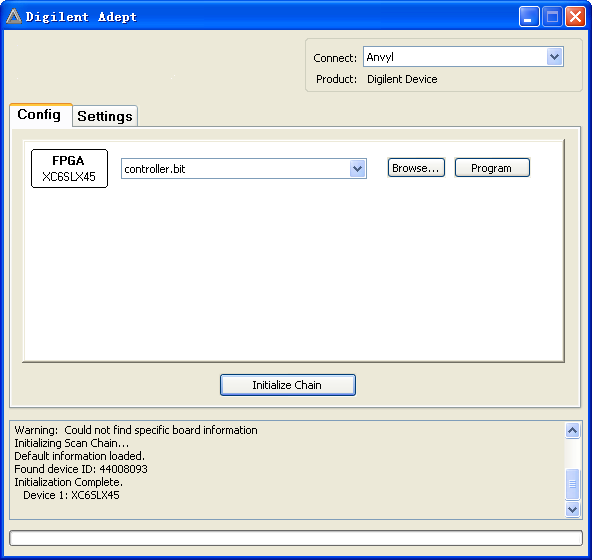


图2-17 Digilent Adept软件

在图2-17中点击Browse按钮，找到刚才生成的二进制比特文件Controller.bit，然后点击Program按钮开始下载，下载成功后会在图2-17下面的状态栏中显示Programming Successful。同学们可在开发板上拨动开关SW0和SW1开关，并观察LD9、LD10及LD11的显示，观察的结果与设计是否一致。

# 附录：Spartan6/XC6SLX45开发板用户接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **实验板器件** | **实验板标注** | **接口引脚** | **实验板器件** | **实验板标注** | **接口引脚** |
| 按钮 | BTN0 | E6 | 红绿黄灯  （2） | LD12(G) | T7 |
| BTN1 | D5 | LD13(Y) | W4 |
| BTN2 | A3 | LD14(R) | U8 |
| 指示灯 | LD0 | W3 | 拨位开关 | SW0 | V5 |
| LD1 | Y4 | SW1 | U4 |
| LD2 | Y1 | SW2 | V3 |
| LD3 | Y3 | SW3 | P4 |
| LD4 | AB4 | SW4 | R4 |
| LD5 | W1 | SW5 | P6 |
| LD6 | AB3 | SW6 | P5 |
| LD7 | AA4 | SW7 | P8 |
| 红绿黄灯  （1） | LD9(G) | R7 |  | | |
| LD10(Y) | U6 |
| LD11(R) | T8 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **7段数码管** | | | | |
| 公共部分 | 信号 | 引脚 | 信号 | 引脚 |
| AA | AA21 | AE | AB19 |
| AB | AA22 | AF | P20 |
| AC | Y22 | DP | P15 |
| AD | N15 |  |  |
| Disp1 | 高位选择 | M17 | 低位选择 | P16 |
| Disp2 | 高位选择 | M17 | 低位选择 | P16 |
| Disp3 | 高位选择 | 灭 | 低位选择 |  |