

中山大学数据科学与计算机学院本科生实验报告

(2017 学年春季学期)

课程名称：计算机组成原理实验

任课教师：郭雪梅

助教：李声涛、王绍菊

年级&班级	1518 班	专业(方向)	移动信息工程
学号	15352408	姓名	张稼伟
电话	13531810182	Email	709075442@qq.com
开始日期	2017.5.5	完成日期	2017.5.6

一、实验题目

通过 Vivado 设计 3-8 译码器，并将其烧录在 Basys3 开发板上。通过拨码开关和 led 控制显示。

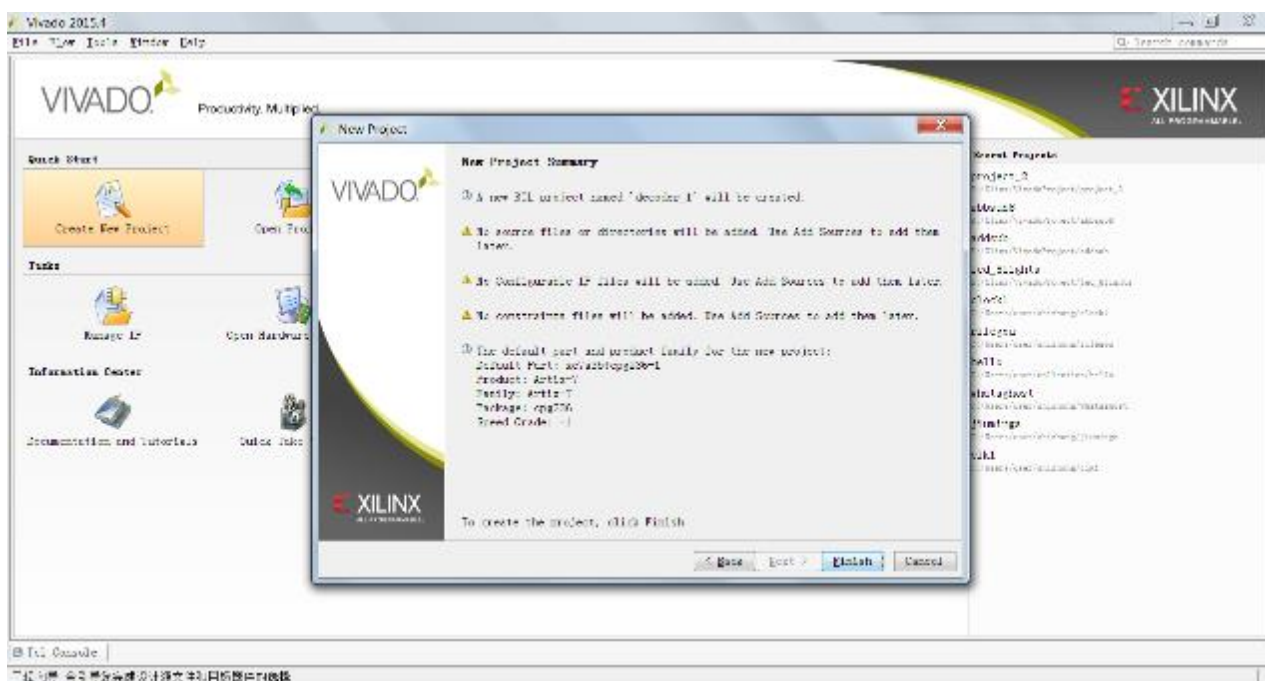
二、实验目的

1. 熟练使用 vivado 的各种功能。
2. 练习 verilog 编程。
3. 练习使用 Basys3 开发板。

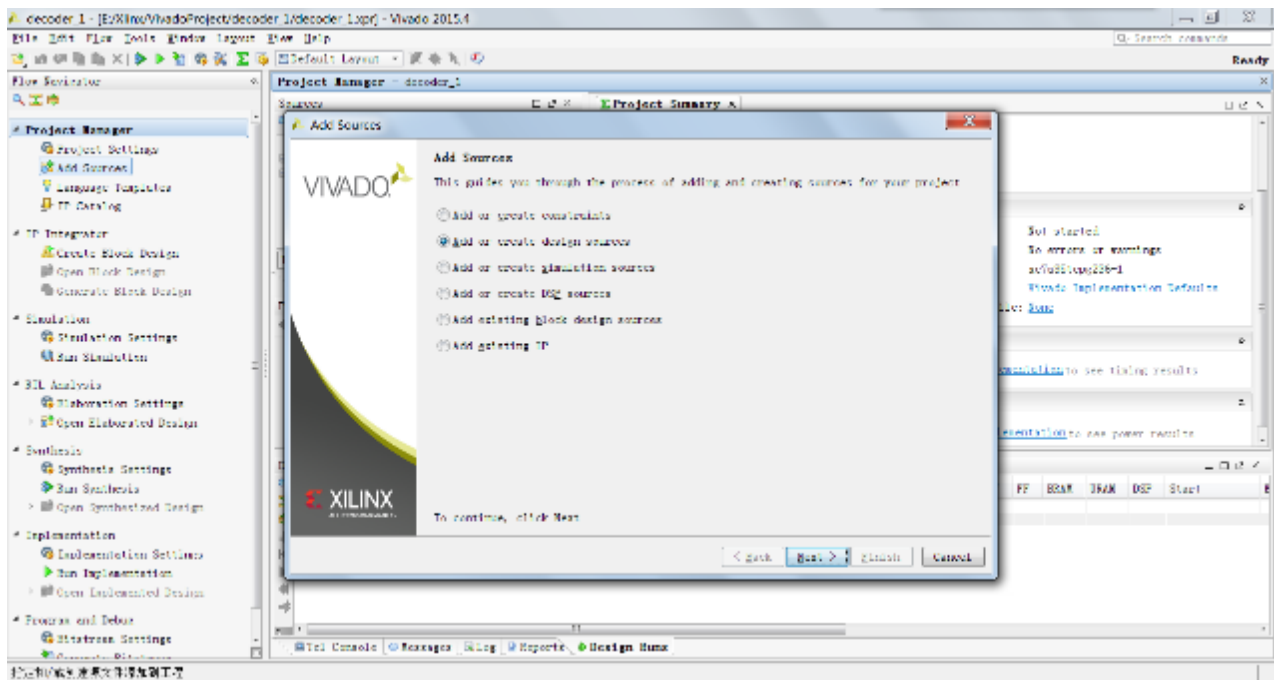
三、实验内容

1. 实验步骤

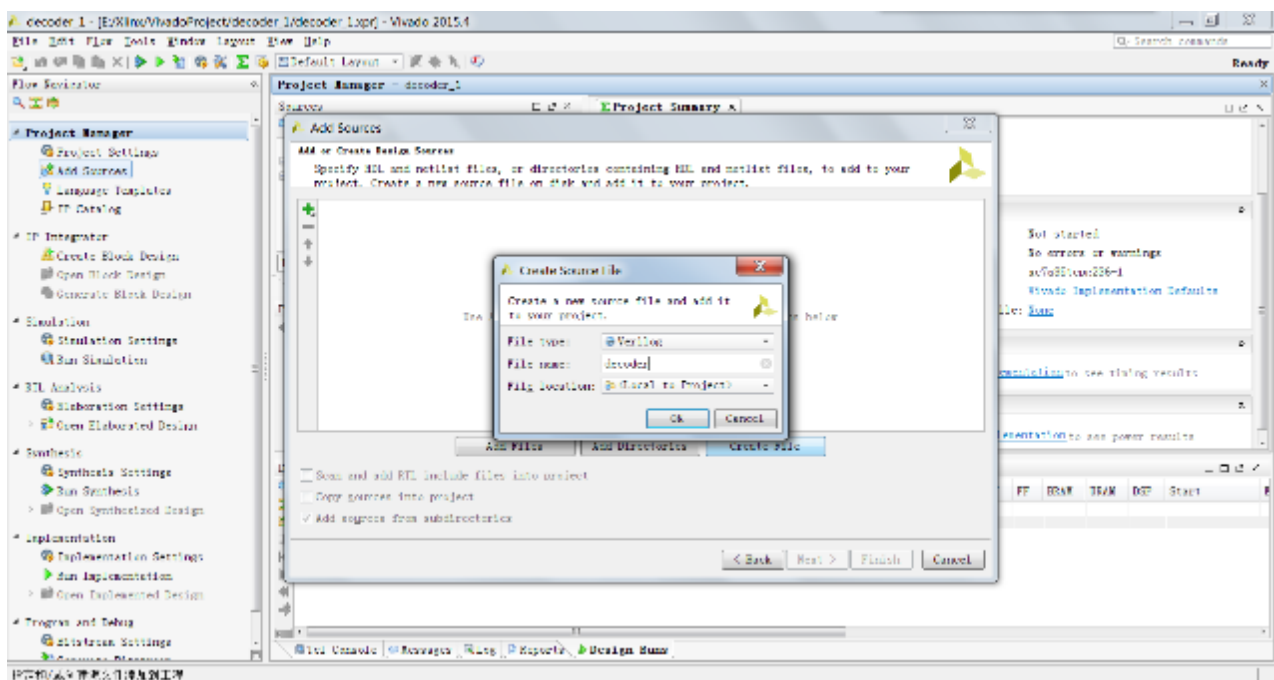
(1) 打开 Vivado，新建一个 RTL 项目。



(2) 点击左侧 Project Manager 下的 Add Sources, 选择 Add or create design sources 来添加源文件。



(2) 选择 create file 创建新的源文件。



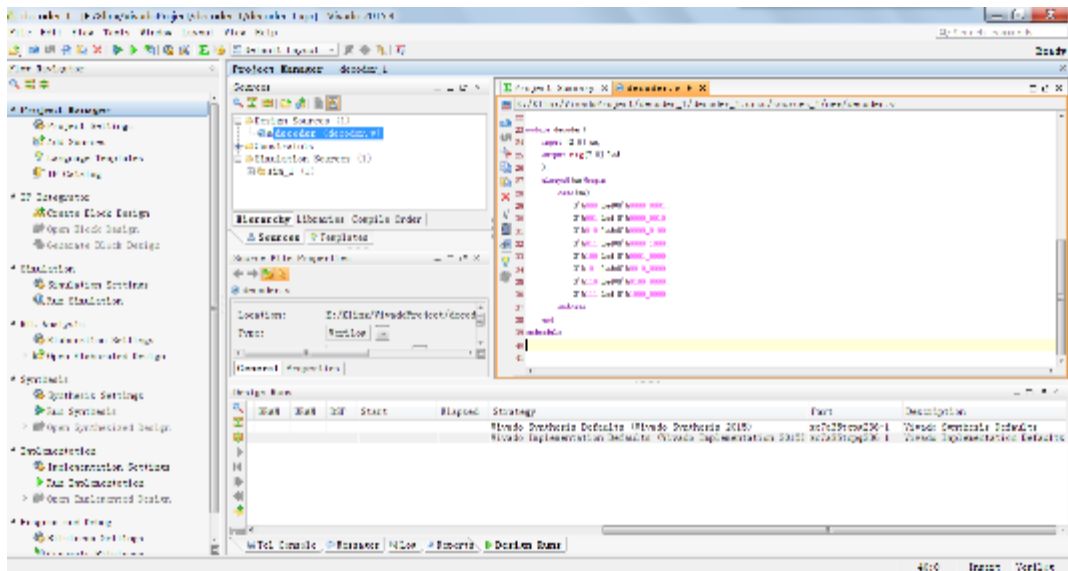
(3) 按照实验教程输入代码。

```
timescale 1ns / 1ps
module decoder(
    input [2:0] sw,
    output reg[7:0] led
); //定义输入输出
```

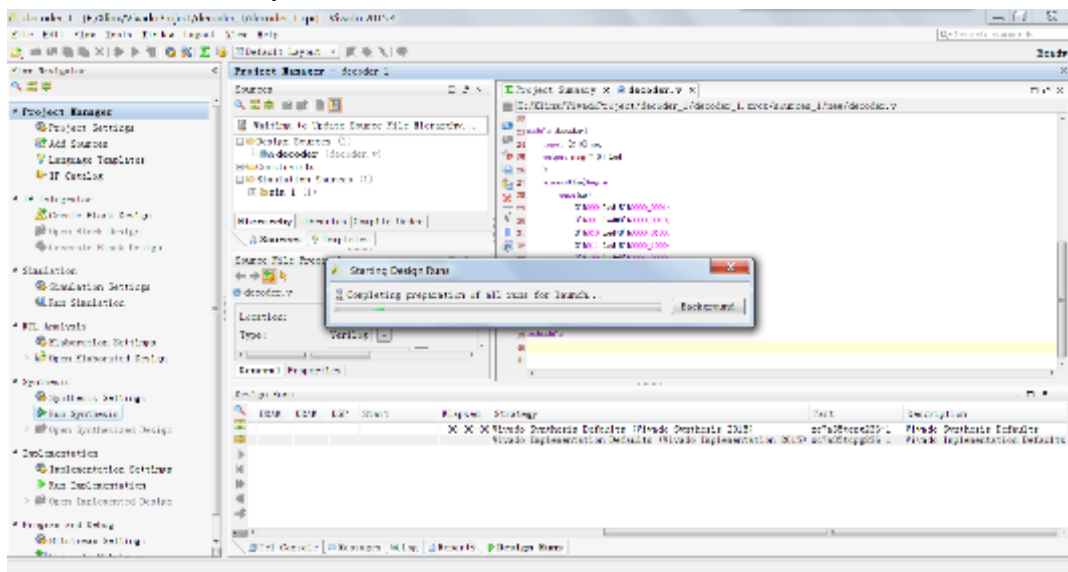
```

always@(sw)begin
    //case 语句来实现译码
    case(sw)
        3'b000:led=8'b0000_0001;
        3'b001:led=8'b0000_0010;
        3'b010:led=8'b0000_0100;
        3'b011:led=8'b0000_1000;
        3'b100:led=8'b0001_0000;
        3'b101:led=8'b0010_0000;
        3'b110:led=8'b0100_0000;
        3'b111:led=8'b1000_0000;
    endcase
end
endmodule

```



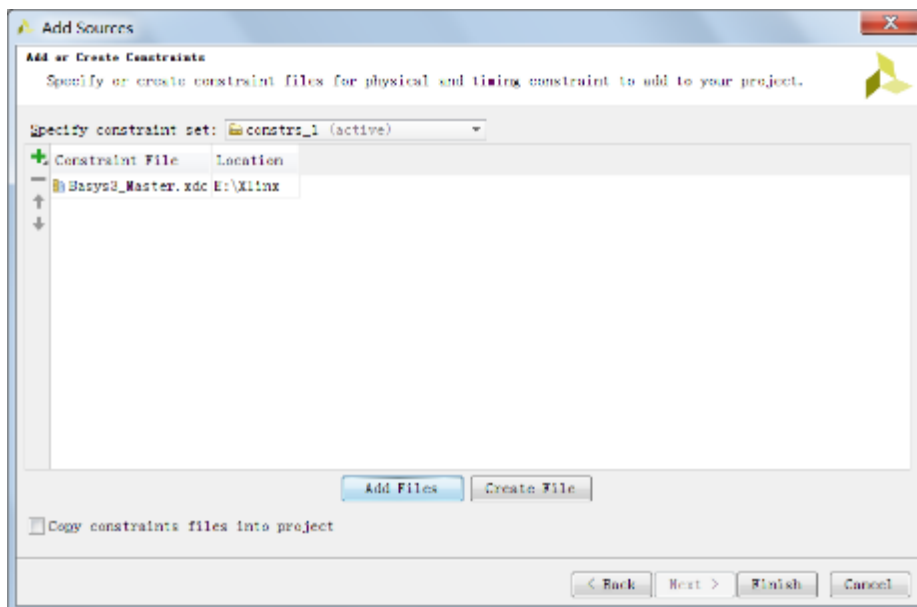
(4) 点左侧 Run Synthesis 综合设计文件。



(5) 综合完成后点左侧 Add source, 选择 Add or create constraints 添加约束文件。



(6) 添加给出的约束文件。



(7) 做 FPGA 的管脚分配。

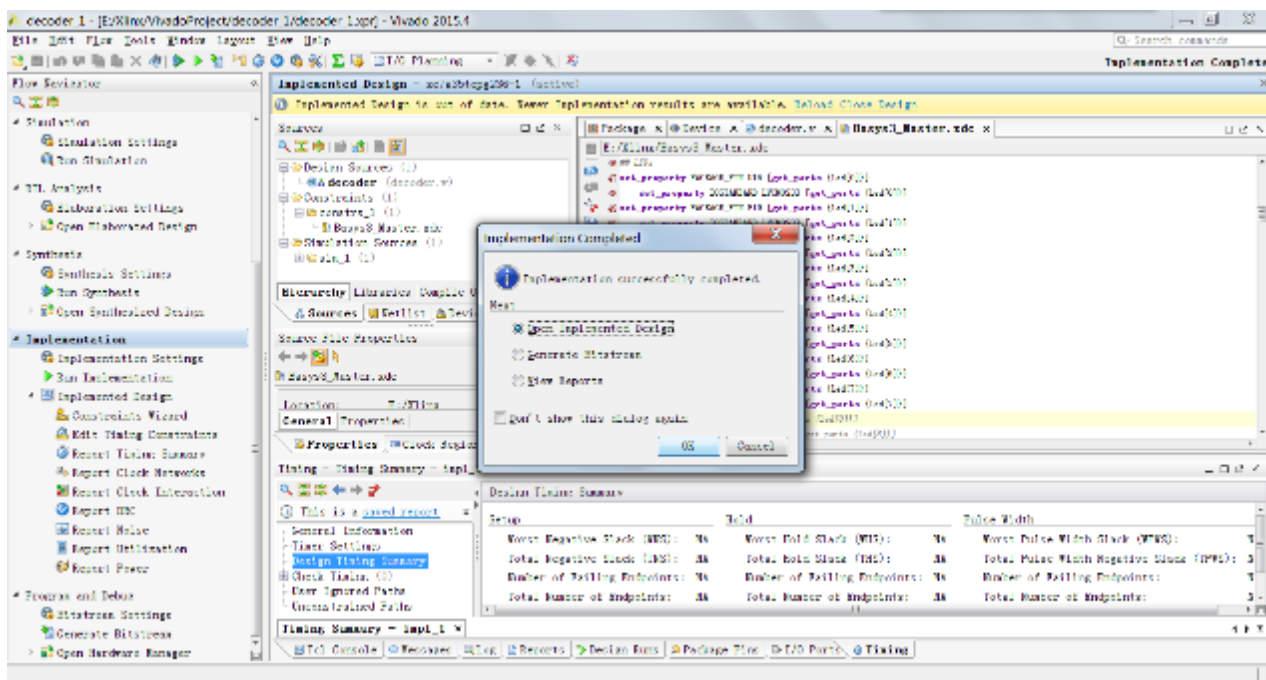
```
## Switches
set_property PACKAGE_PIN V17 [get_ports {sw[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {sw[0]}]
set_property PACKAGE_PIN V16 [get_ports {sw[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {sw[1]}]
set_property PACKAGE_PIN W16 [get_ports {sw[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {sw[2]}]
set_property PACKAGE_PIN W17 [get_ports {sw[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {sw[3]}]
set_property PACKAGE_PIN W15 [get_ports {sw[4]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {sw[4]}]
set_property PACKAGE_PIN V15 [get_ports {sw[5]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {sw[5]}]
set_property PACKAGE_PIN W14 [get_ports {sw[6]}]
```

```

46 ## LEDs
47 set_property PACKAGE_PIN U16 [get_ports {led[0]}]
48 set_property IOSTANDARD LVCMOS33 [get_ports {led[0]}]
49 set_property PACKAGE_PIN E19 [get_ports {led[1]}]
50 set_property IOSTANDARD LVCMOS33 [get_ports {led[1]}]
51 set_property PACKAGE_PIN U19 [get_ports {led[2]}]
52 set_property IOSTANDARD LVCMOS33 [get_ports {led[2]}]
53 set_property PACKAGE_PIN V19 [get_ports {led[3]}]
54 set_property IOSTANDARD LVCMOS33 [get_ports {led[3]}]
55 set_property PACKAGE_PIN W18 [get_ports {led[4]}]
56 set_property IOSTANDARD LVCMOS33 [get_ports {led[4]}]
57 set_property PACKAGE_PIN U15 [get_ports {led[5]}]
58 set_property IOSTANDARD LVCMOS33 [get_ports {led[5]}]
59 set_property PACKAGE_PIN U14 [get_ports {led[6]}]
60 set_property IOSTANDARD LVCMOS33 [get_ports {led[6]}]
61 set_property PACKAGE_PIN V14 [get_ports {led[7]}]
62 set_property IOSTANDARD LVCMOS33 [get_ports {led[7]}]
63 set_property PACKAGE_PIN V13 [get_ports {led[8]}]
64 set_property IOSTANDARD LVCMOS33 [get_ports {led[8]}]
65 set_property PACKAGE_PIN V3 [get_ports {led[9]}]

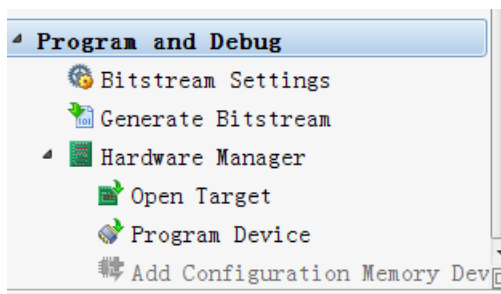
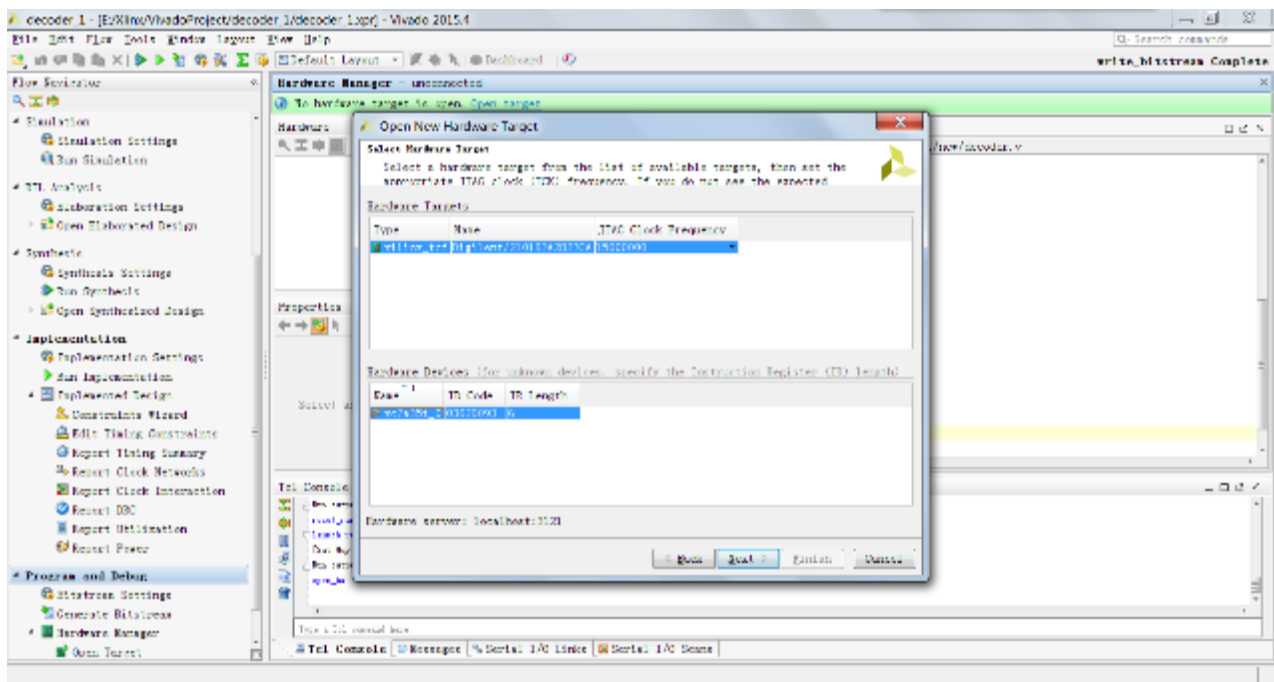
```

(8) 点击 Run Implementation

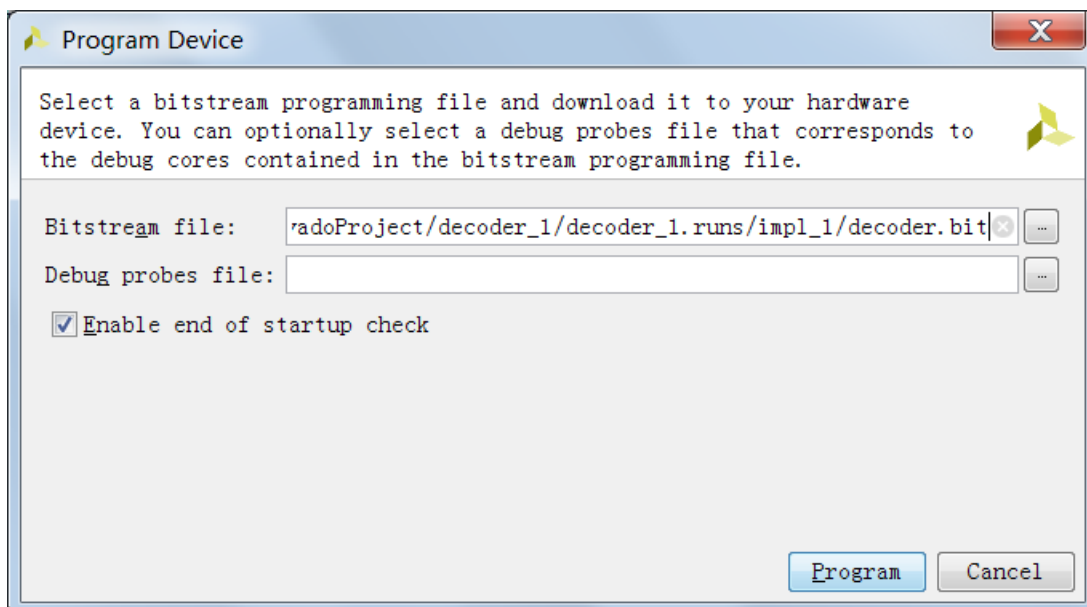


(9) 点击 Generate Bitstream 产生比特流

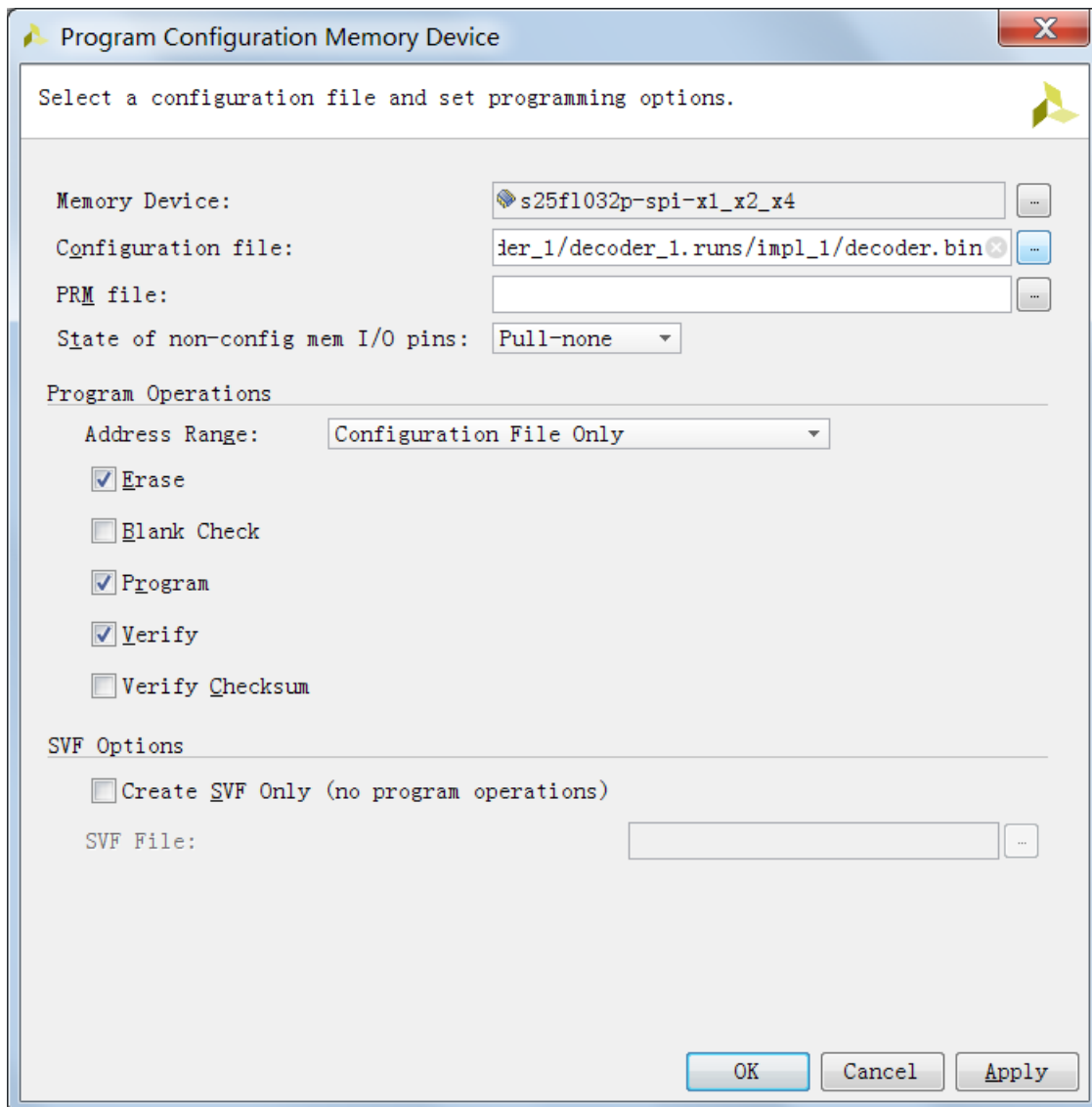
(10) 板子连接电脑，点击 Program and Debug 下的 Hardware Manager 的 Open Target，选择板子。点击 Program Device，选择板子。



(11) 选择之前生成的比特流文件。



(12) 烧录成功后板子按照要求工作。



(17)完成后与之前工作状态相同

2. 实验原理

输入信号 wire [2:0]sw, 输出信号 reg [7:0]led。

这里的 led 必须为 reg 型, 因为在 always 中不断改变。

这里使用 case 语句实现 3-8 译码器。

sw 的电平是该组合逻辑的触发信号。每种 SW 信号对应一种 led 亮灯模式。

比如将拨码开关都关闭, 对应 000, 那么输出为 0000_0000, led0 亮。

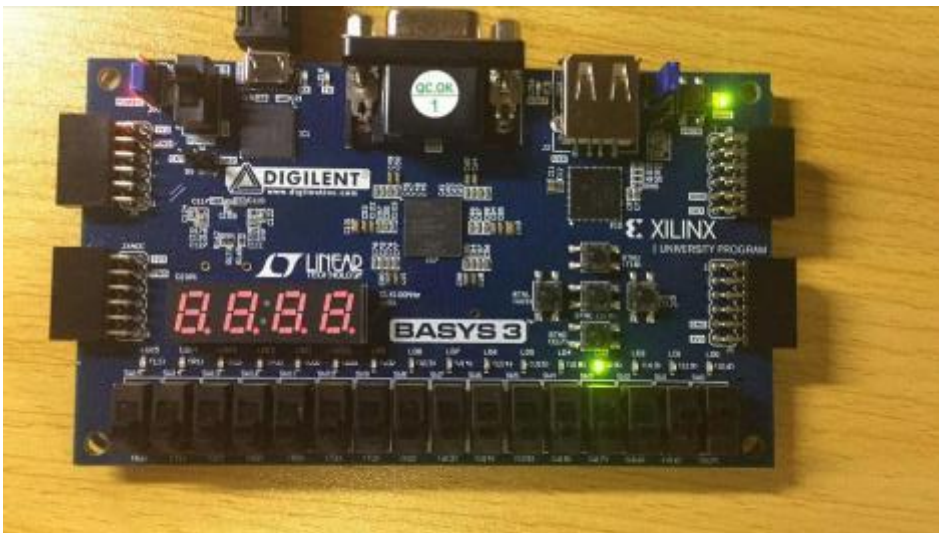
四、实验结果

每种SW信号对应一种led亮灯模式。

001



011



101





五、实验感想

这个实验的重点是教会我们如何将 vivado 中写好的代码烧录进 basys3 的开发板中去运行，3-8 译码器的代码设计比较简单，直接 8 个 case 判断 8 种输入应该生成什么样的输出即可。烧录进开发板的操作其实我们以前也做过，不过这次是在约束文件中分配好了管脚，而之前是综合实现之后再一个界面慢慢一个一个加。以前我不知道这样做完后会生成这个约束文件，现在知道了之后以后就不用每次新建工程都去手动添加了，方便不少。