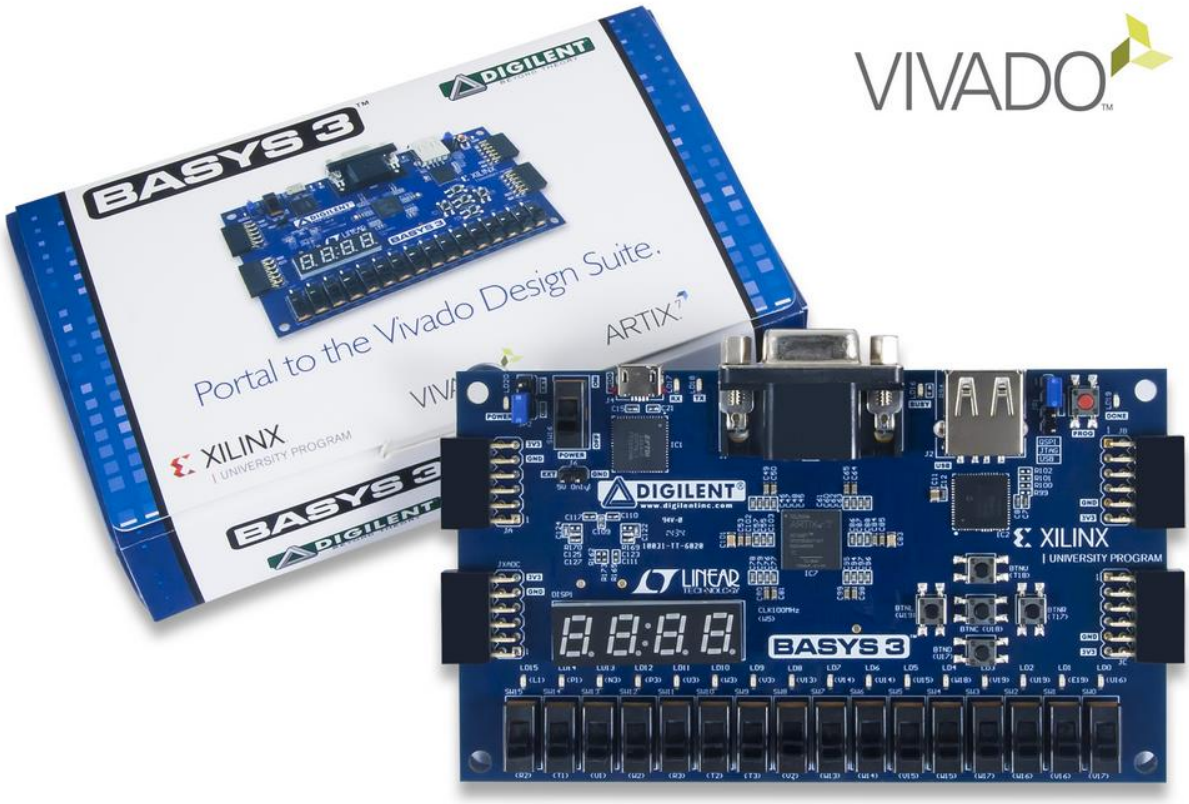
上海浦东新区张东路 1387 号 45 幢
位于母公司美国国家仪器 NI 中国总部大楼内
电话：+86-21-58893151
邮箱：china.sales@digilent.com.cn

“你的 Basys 3 第一个入门实验”官方指导手册



目　录

第一章 Basys3 硬件电路

Basys3 是围绕着一个 Xilinx Artix®-7 FPGA 芯片 XC7A35T-1CPG236C 搭建的，它提供了完整、随时可以使用的硬件平台，并且它适合于从基本逻辑器件到复杂控制器件的各种主机电路。Basys3 板上集成了大量的 I/O 设备和 FPGA 所需的支持电路，让您能够构建无数的设计而不需要其他器件。

主要规格/特殊功能

产品规格：

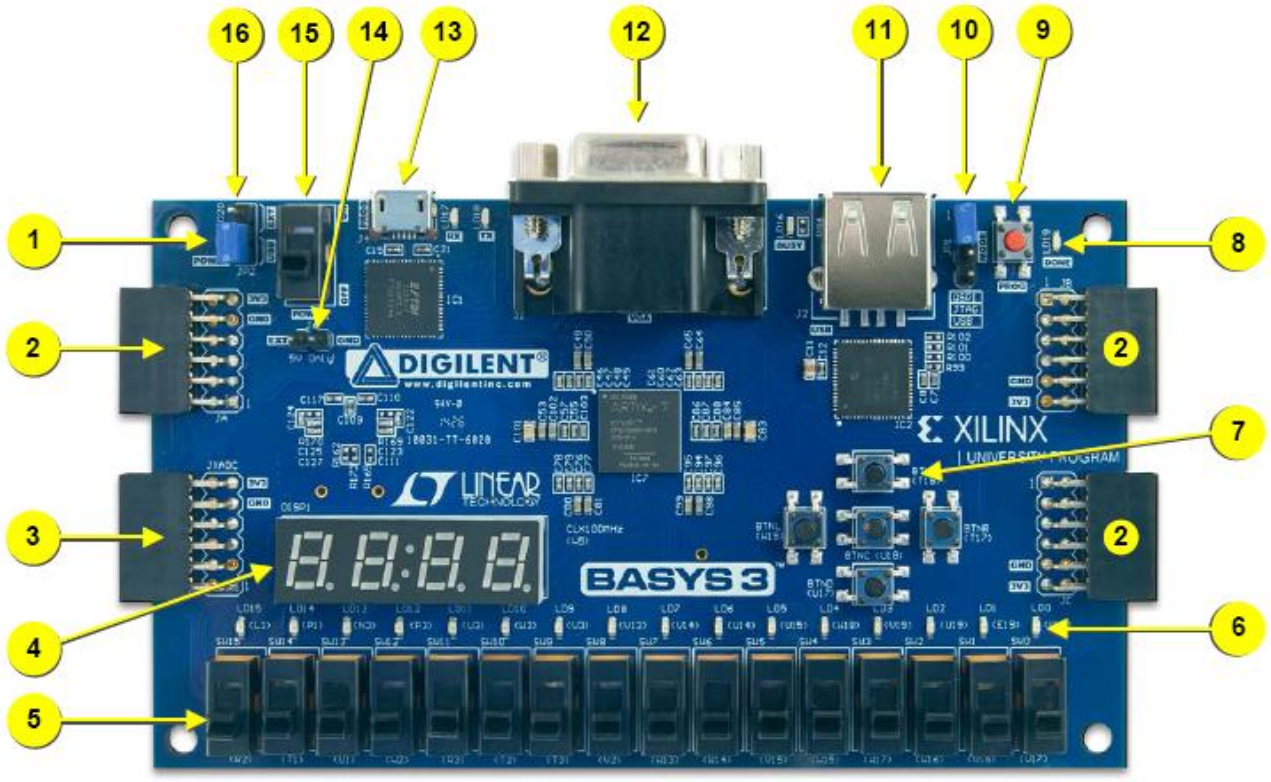
Basys3为想要学习FPGA和数字电路设计的用户提供一个理想的电路设计平台。Basys3板提供完整的硬件存取电路,可以完成从基本逻辑到复杂控制器的设计。四个标准扩展连接器配合用户设计的电路板，或Pmods（Digilent设计的A / D和D / A转换，电机驱动器，传感器输入等）其他功能。扩展信号的8针接口均采用ESD保护，附带的USB电缆，提供电源和编程接口，因此不需要额外配置电源或其他编程电缆，使之成为了入门或复杂数字电路系统设计的完美低成本平台。

关键特性：

- 33,280 个逻辑单元，六输入LUT结构
- 1,800 Kbits 快速RAM块
- 5个时钟管理单元，均各含一个锁相环 (PLL)
- 90个DSP slices
- 内部时钟最高可达450MHz
- 1个片上模数转换器 (XADC)

外围设备：

- 16个拨键开关
- 16个LED
- 5个按键开关
- 4位7段数码管
- 3个Pmod连接口
- 一个专用AD信号Pmod接口
- 12位的VGA输出接口
- USB-UART桥
- 串口flash
- 用于FPGA编程和通信的USB-JTAG口
- 可连接鼠标、键盘、记忆棒的USB口



序号	描述	序号	描述
1	电源指示灯	9	FPGA配置复位按键
2	Pmod连接口	10	编程模式跳线柱
3	专用模拟信号Pmod连接口	11	USB连接口
4	4位7段数码管	12	VGA连接口
5	16个拨键开关	13	UART/JTAG共用USB接口
6	16个LED	14	外部电源接口
7	5个按键开关	15	电源开关
8	FPGA编程指示灯	16	电源选择跳线柱

1.1 电源电路

Basys3 开发板可以通过 2 种方式进行供电，一种是通过 J4 的 USB 端口供电；另一种是通过 J6 的接线柱进行供电（5V）。通过 JP2 跳线帽的不同选择进行供电方式的选择。电源开关通过 SW16 进行控制，LD20 为电源开关的指示灯。电源的电路如下图所示：

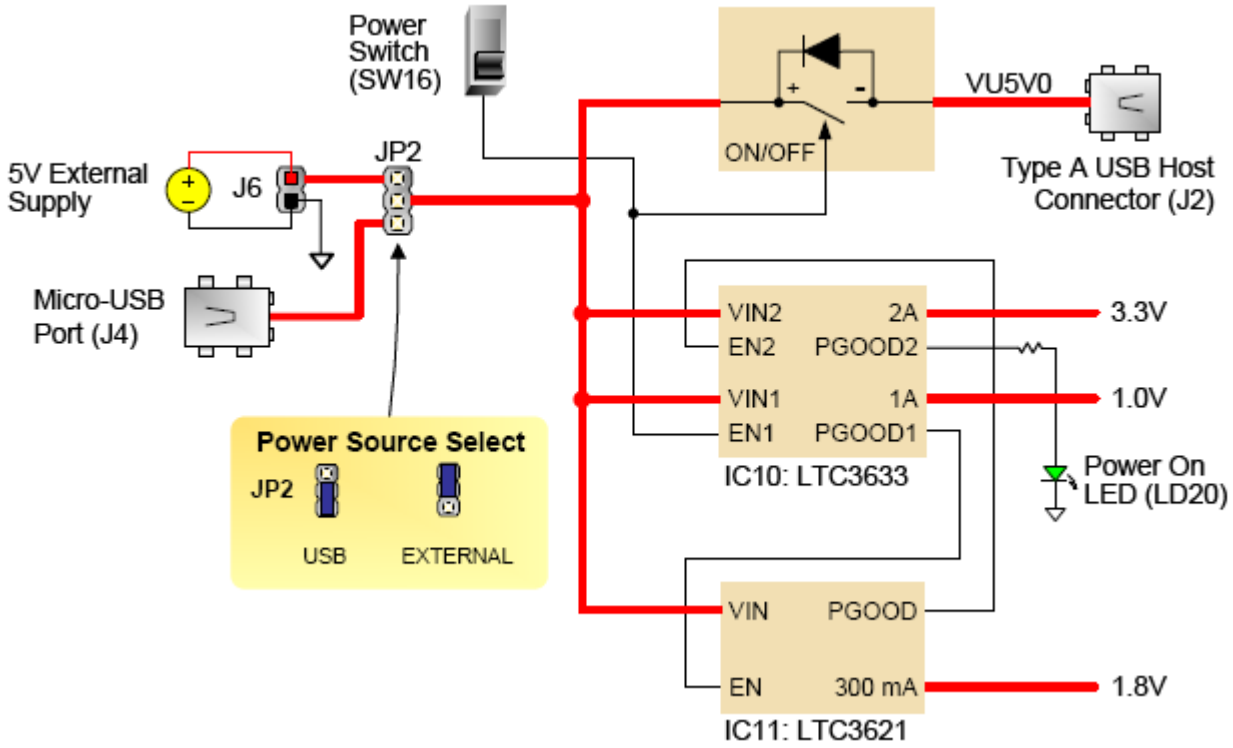


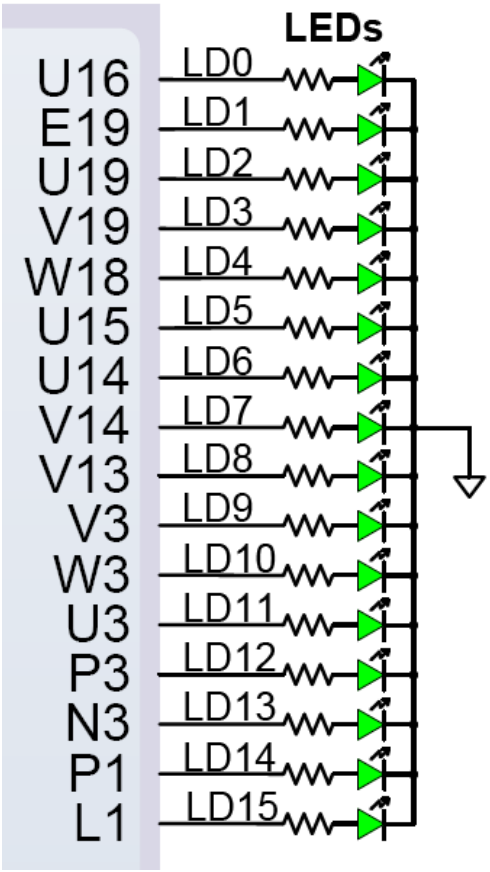
图 1 USB 接口电路

说明，如果选用外部电源（即 J6）那么应该保证：1，电源电压在 4.5V-5.5V 范围内；2，至少能提供 1A 的电流。

注意：只有在特别情况下电源电压才可以使用 3.6V 电压。

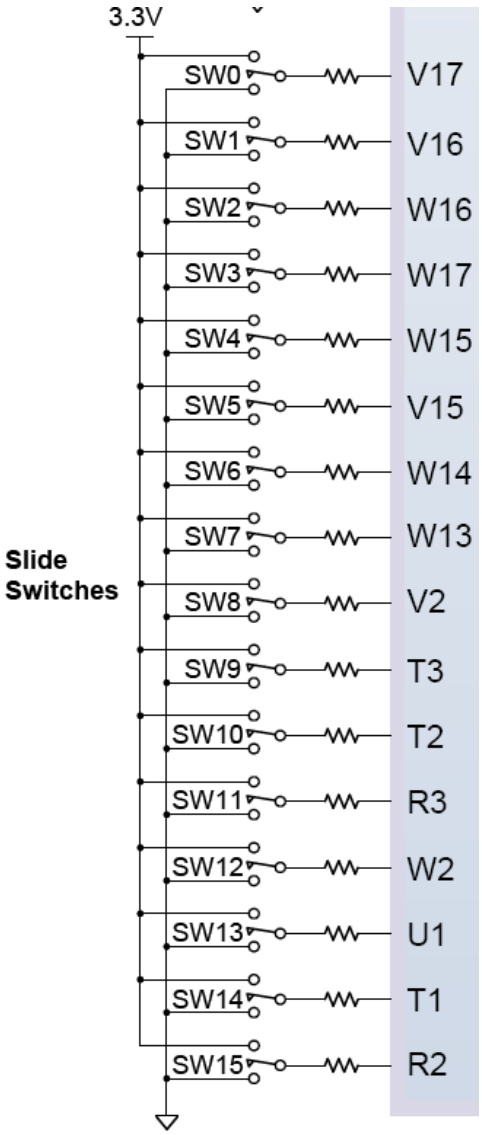
1.2 LED 灯电路

LED 部分的电路如图所示。当 FPGA 输出为高电平时，相应的 LED 点亮；否则，LED 熄灭。板上配有 16 个 LED，在实验中灵活应用，可用作标志显示或代码调试的结果显示，既直观明了又简单方便。



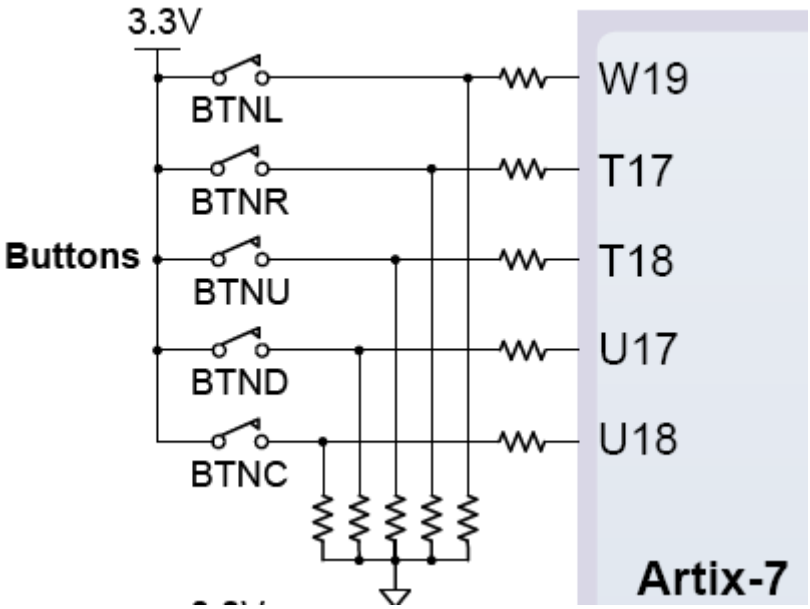
1.3 拨码开关电路

拨码开关的电路如图所示。在使用这个 16 位拨码开关时请注意一点，当开关打到下档时，表示 FPGA 的输入为低电平。



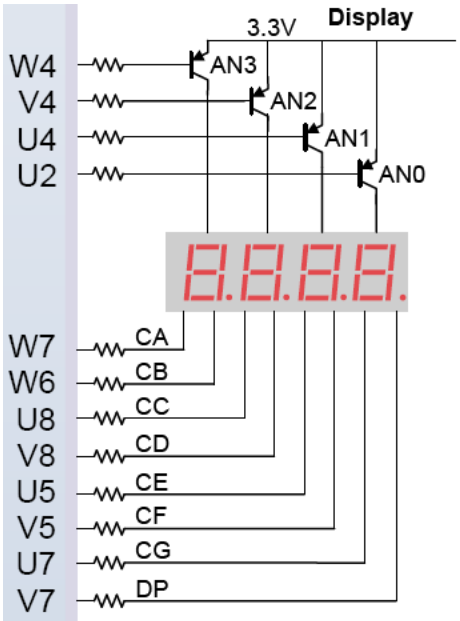
1.4 按键电路

按键部分的电路如图所示。板上配有 5 个按键，当按键按下时，表示 FPGA 的相应输入脚为高电平。在学习过程中，我们建议每个工程都有一个复位输入，这对代码调试将大有好处。



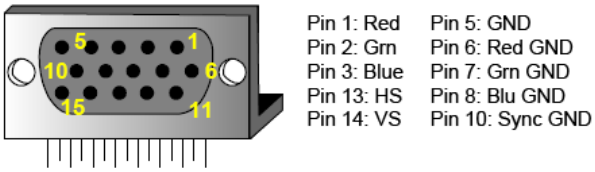
1.5 数码管电路

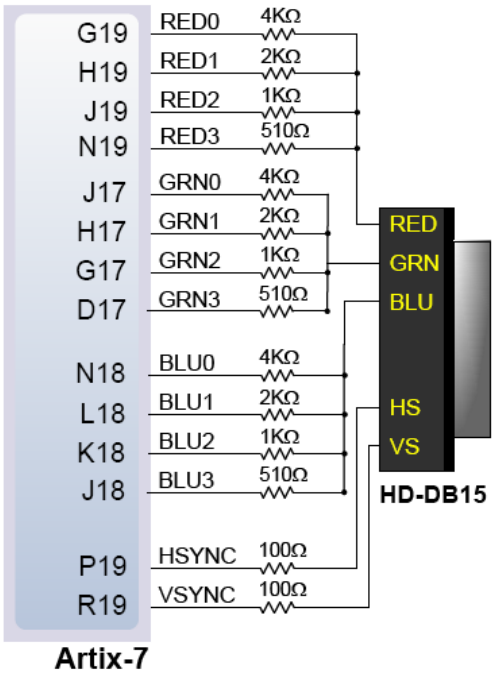
数码管显示部分的电路如图所示。我们使用的是一个四位带小数点的七段共阳数码管，当我们相应的输出脚为低电平时，该段位的 LED 点亮。位选位也是低电平选通。



1.6 VGA 显示电路

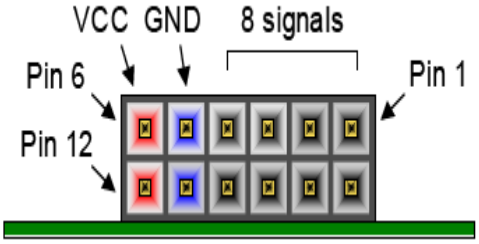
VGA 视频显示部分的电路如图所示。我们所用的电阻搭的 12bit(2^{12} 色)电路，由于没有采用视频专用 DAC 芯片，所以色彩过渡表现不是十分完美。





1.7　IO 扩展电路

4 个标准的扩展连接器（其中一个为专用 AD 信号 Pmod 接口）允许设计使用面包板、用户设计的电路或 Pmods 扩展 Basys3 板板，（Pmods 是价格便宜的模拟和数字 I / O 模块，能提供一个 A/ D&D / A 转换，电机驱动器，传感器 投入和许多其他功能）。8 针连接器上的信号免受 ESD 损害和短路损害，从而确保了在任何环境中的 使用寿命更长。



Pmod JA	Pmod JB	Pmod JC	Pmod XDAC
JA1: J1	JB1: A14	JC1: K17	JXADC1: J3
JA2: L2	JB2: A16	JC2: M18	JXADC2: L3
JA3: J2	JB3: B15	JC3: N17	JXADC3: M2
JA4: G2	JB4: B16	JC4: P18	JXADC4: N2
JA7: H1	JB7: A15	JC7: L17	JXADC7: K3
JA8: K2	JB8: A17	JC8: M19	JXADC8: M3
JA9: H2	JB9: C15	JC9: P17	JXADC9: M1
JA10: G3	JB10: C16	JC10: R18	JXADC10: N1

1.ψ FPGA 调试及配置电路

上电后，Basys3 板上必须配置 FPGA，然后才能执行任何有用功能。在配置过程中，一“Bit”文件转移到 FPGA 内存单元中实现逻辑功能和电路互连。借助赛灵思免费的 Vivado 软件可以通过 VHDL，Verilog 语言，或基于原理图的源文件创建.bit 文件。

编程下载：

下载程序 3 种方式：

- 用Vivado通过JTAG方式下载.bit文件到FPGA芯片。
- 用Vivado通过QSPI方式下载.bit文件到Flash芯片，实现掉电不易失。
- 用U盘或移动硬盘通过J2的USB端口下载.bit文件到FPGA芯片（建议将.bit文件放到U盘根目录下，且只放1个），该U盘应该是FAT32文件系统。

注意：1、下载方式通过JP1的短路帽进行选择；
　　2、系统默认主频率为100MHz

引脚分配表格：

LED	PIN	CLOCK	PIN	SWITCH	PIN	BUTTON	PIN	Seven-segment digital tube	PIN
LD0	U16	MRCC	W5	SW0	V17	BTNU	T18	AN0	U2
LD1	E19			SW1	V16	BTNR	T17	AN1	U4
LD2	U19			SW2	W16	BTND	U17	AN2	V4
LD3	V19			SW3	W17	BTNL	W19	AN3	W4
LD4	W18			SW4	W15	BTNC	U18	CA	W7
LD5	U15			SW5	V15			CB	W6
LD6	U14			SW6	W14			CC	U8
LD7	V14			SW7	W13			CD	V8
LD8	V13	USB(J2)	PIN	SW8	V2			CE	U5
LD9	V3	PS2_CLK	C17	SW9	T3			CF	V5
LD10	W3	PS2_DAT	B17	SW10	T2			CG	U7
LD11	U3			SW11	R3			DP	V7
LD12	P3			SW12	W2				
LD13	N3			SW13	U1				
LD14	P1			SW14	T1				
LD15	L1			SW15	R2				

VGA	PIN	JA	PIN	JB	PIN	JC	PIN	JXADC	PIN
RED0	G19	JA0	J1	JB0	A14	JC0	K17	JXADC0	J3
RED1	H19	JA1	L2	JB1	A16	JC1	M18	JXADC1	L3
RED2	J19	JA2	J2	JB2	B15	JC2	N17	JXADC2	M2
RED3	N19	JA3	G2	JB3	B16	JC3	P18	JXADC3	N2
GRN0	J17	JA4	H1	JB4	A15	JC4	L17	JXADC4	K3
GRN1	H17	JA5	K2	JB5	A17	JC5	M19	JXADC5	M3
GRN2	G17	JA6	H2	JB6	C15	JC6	P17	JXADC6	M1
GRN3	D17	JA7	G3	JB7	C16	JC7	R18	JXADC7	N1
BLU0	N18								
BLU1	L18								
BLU2	K18								
BLU3	J18								
HSYNC	P19								
YSYNC	R19								

第二章 电路实验

这一章我们将通过硬件描述语言HDL——Verilog，对Basys3上的各模块进行逻辑验证，实现相应的实验目标。事不宜迟，让我们马上进入这一阶段的学习。

2　七段数码管显示实验

利用开发板 4 个 7 段数码管依次显示数字“1234”和“4321”，通过判断拨键开关 SW0 的状态进行选择数码管是顺序显示数字还是逆序显示数字。亦可使用按键开关控制模式转换。

如：当 SW0=1 时显示“1234”;

当 SW0=0 时显示“4321”;

要求：1.给出行为仿真波形

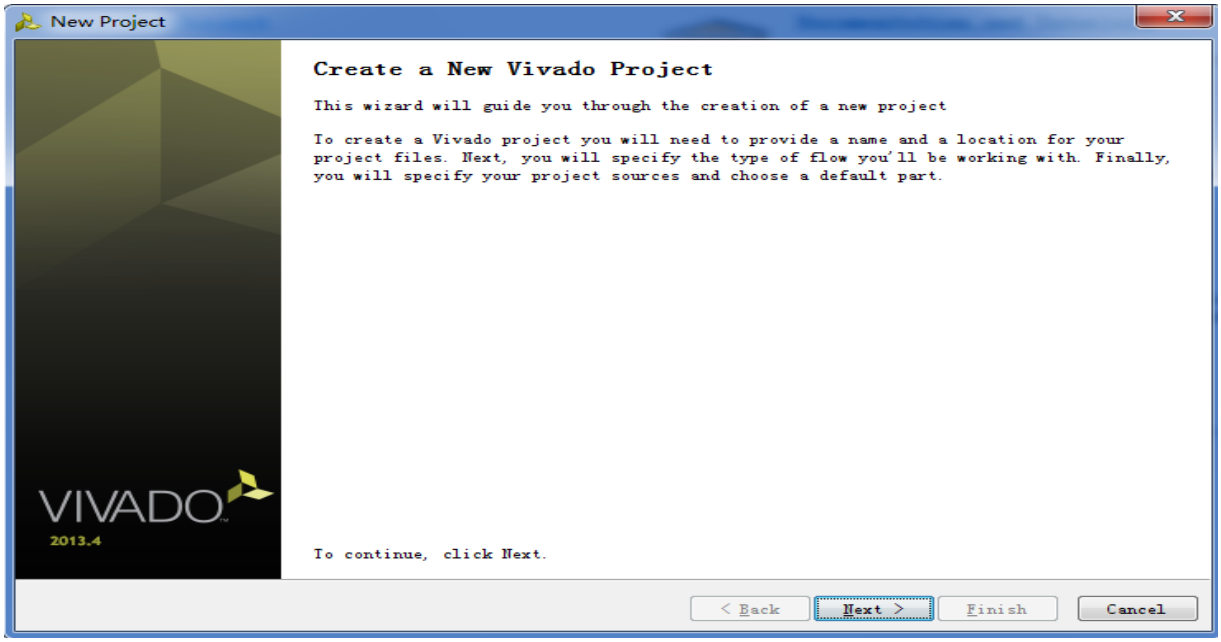
2.将程序烧录到 ROM 里面

3.开发板演示

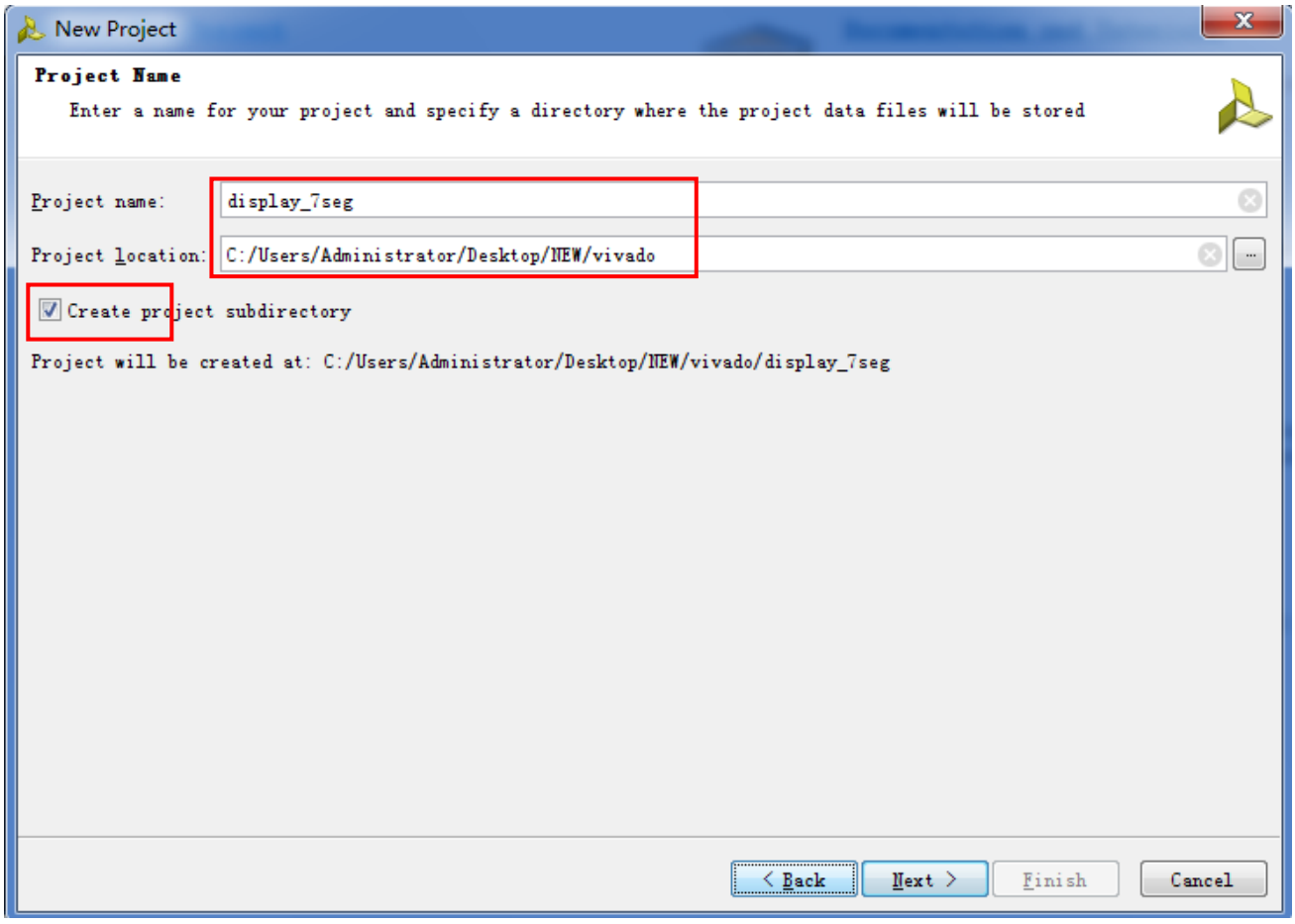
操作步骤详解：

第一步：打开 Vivado，相信大家都知道怎样打开一个软件，稍等软件启动后，进入如下界面：

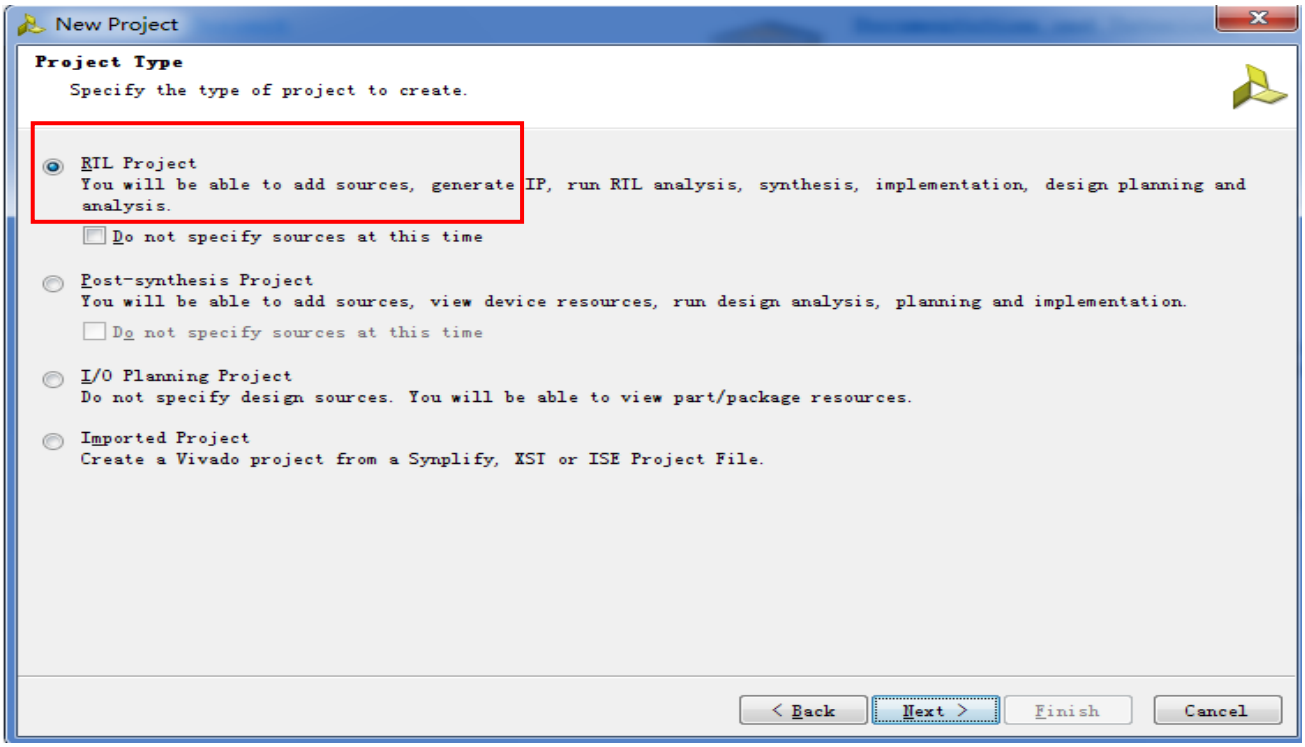
第二步：点Create New Project …进入新建工程向导



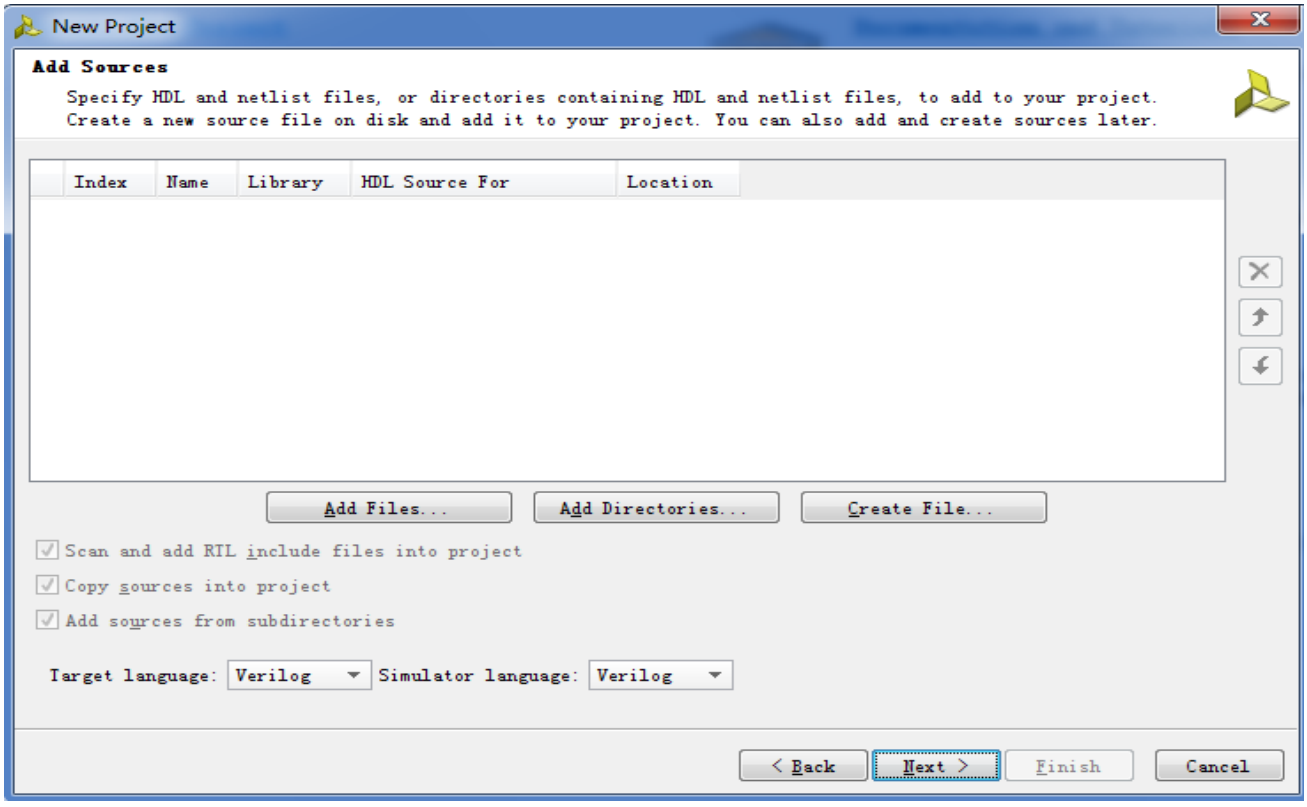
第三步：在弹出窗口填写Project名称，此处为“display_7seg”，存储地址为桌面文件夹“NEW”下的子文件夹“vivado”。



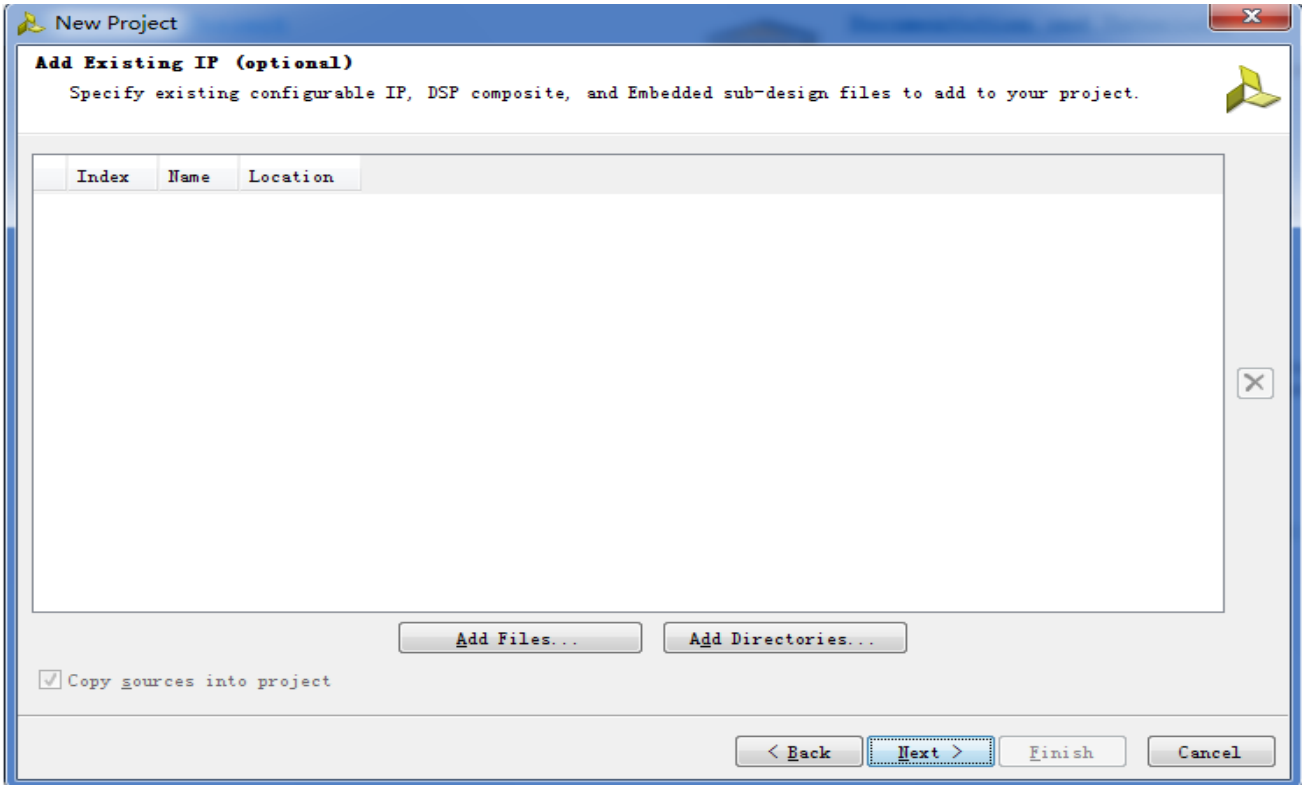
在弹出窗口按下图设置：



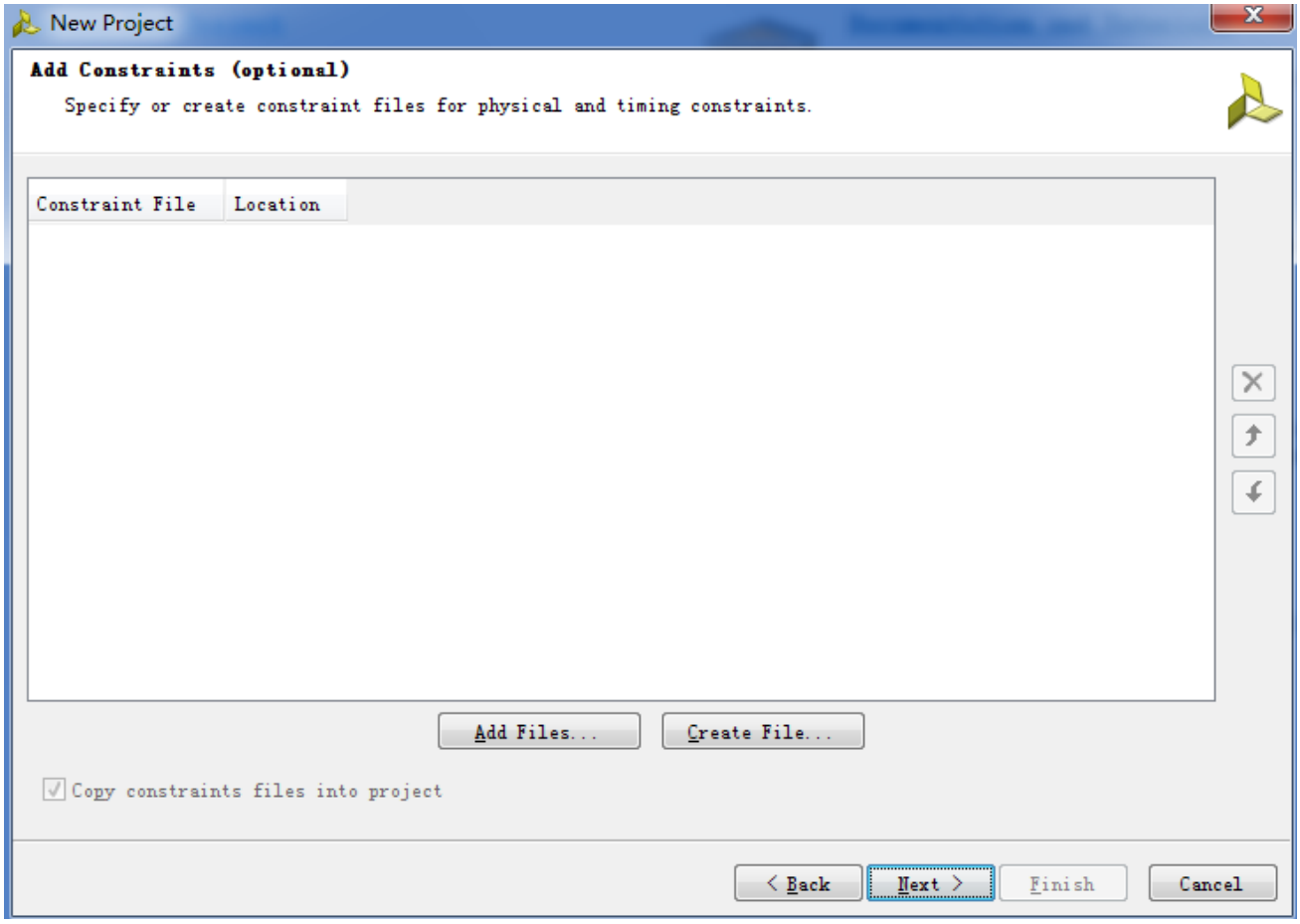
由于我们是新建工程，所以此处默认没有可以添加的源文件，并且设置编程语言和仿真语言均设置为Verilog。点击Next



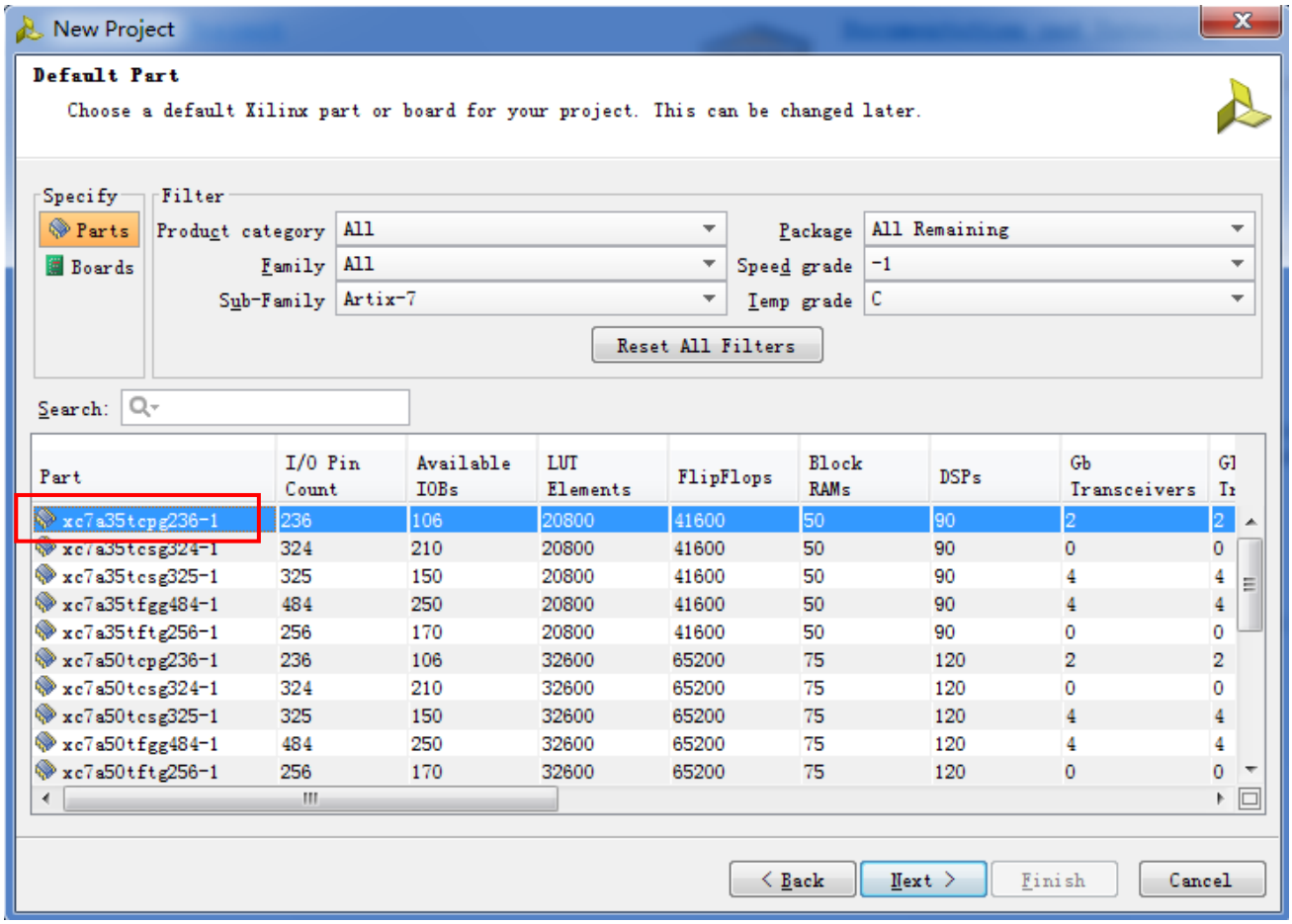
也没有可以添加的IP，所以不添加IP，点击Next如下图：



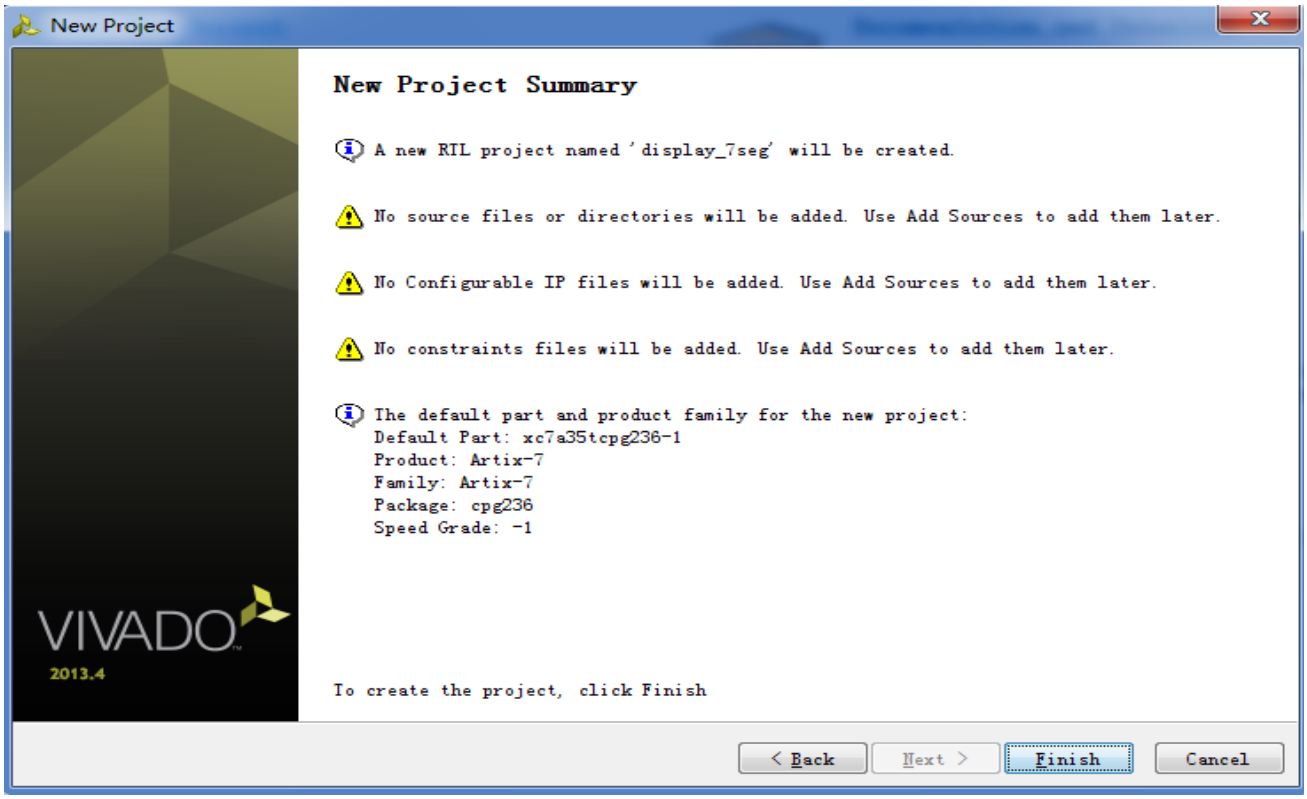
也没有可以添加的约束文件，所以不添加，点击Next如下图：



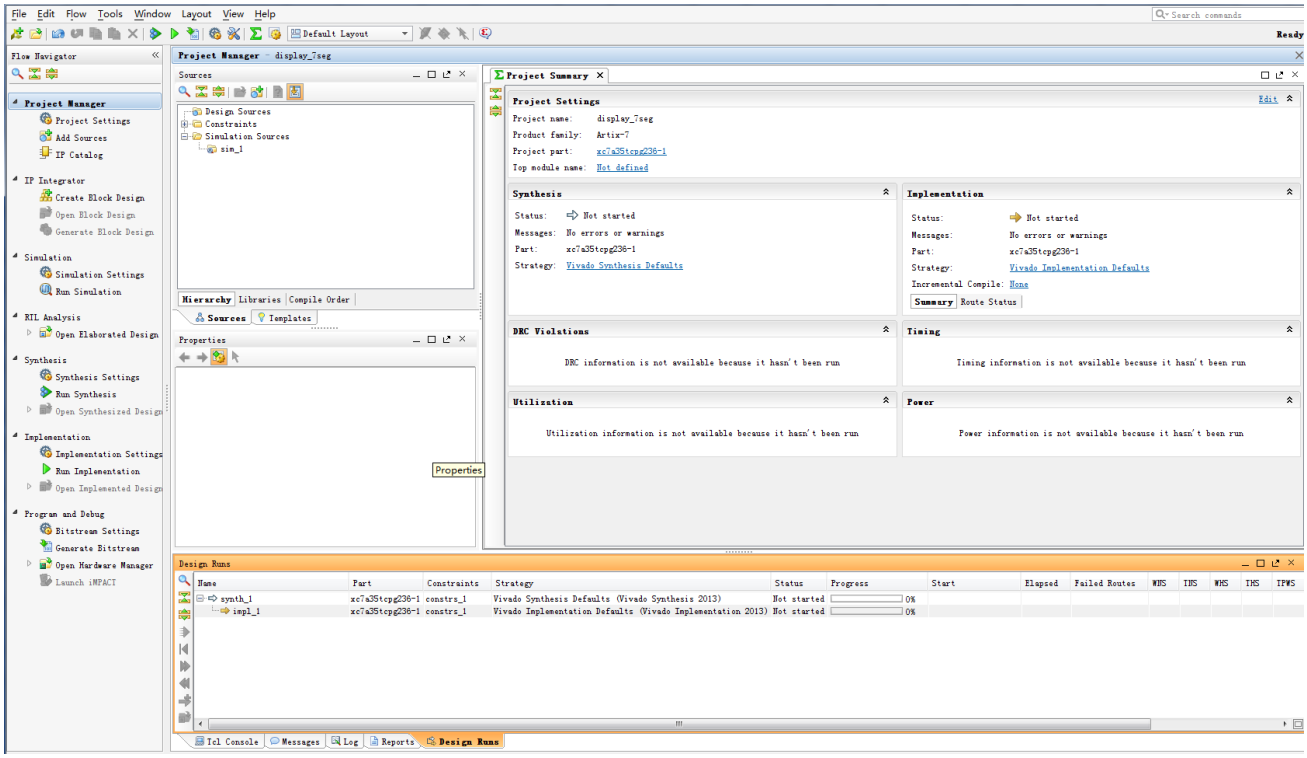
选择FPGA芯片，如下图选择xc7a35tcpg236-1



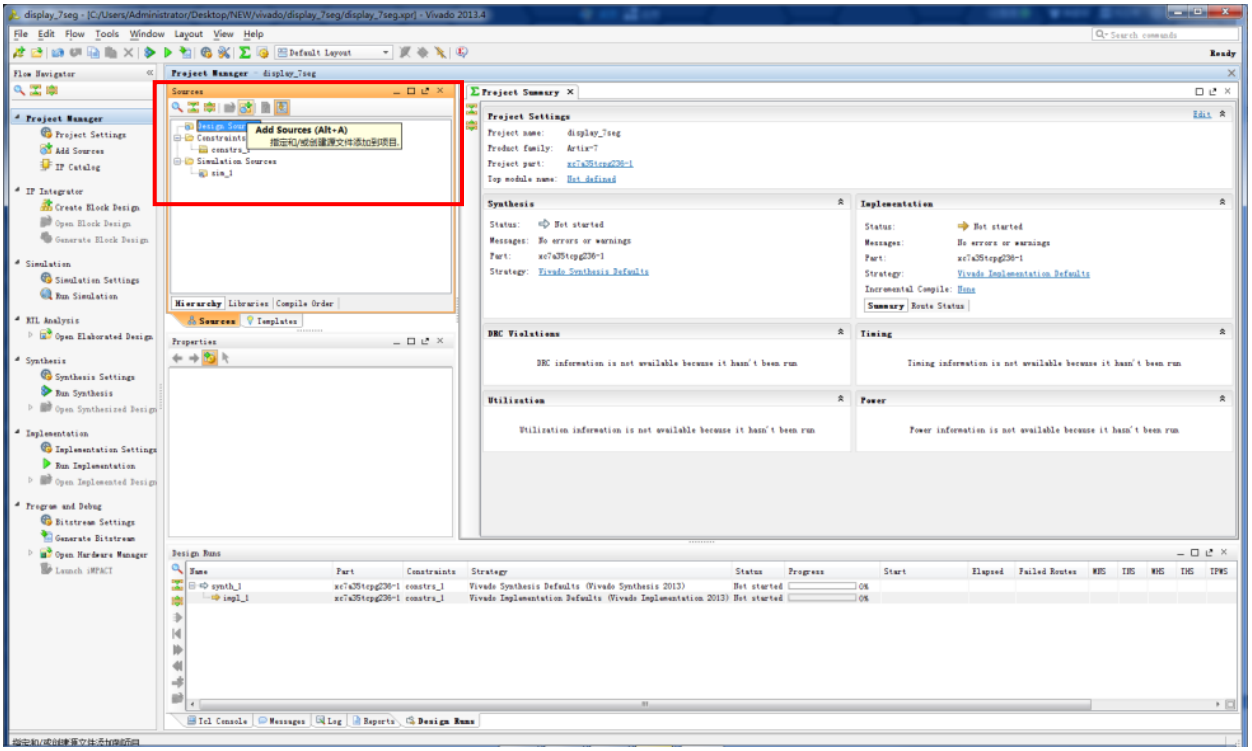
完成设置，如下图：



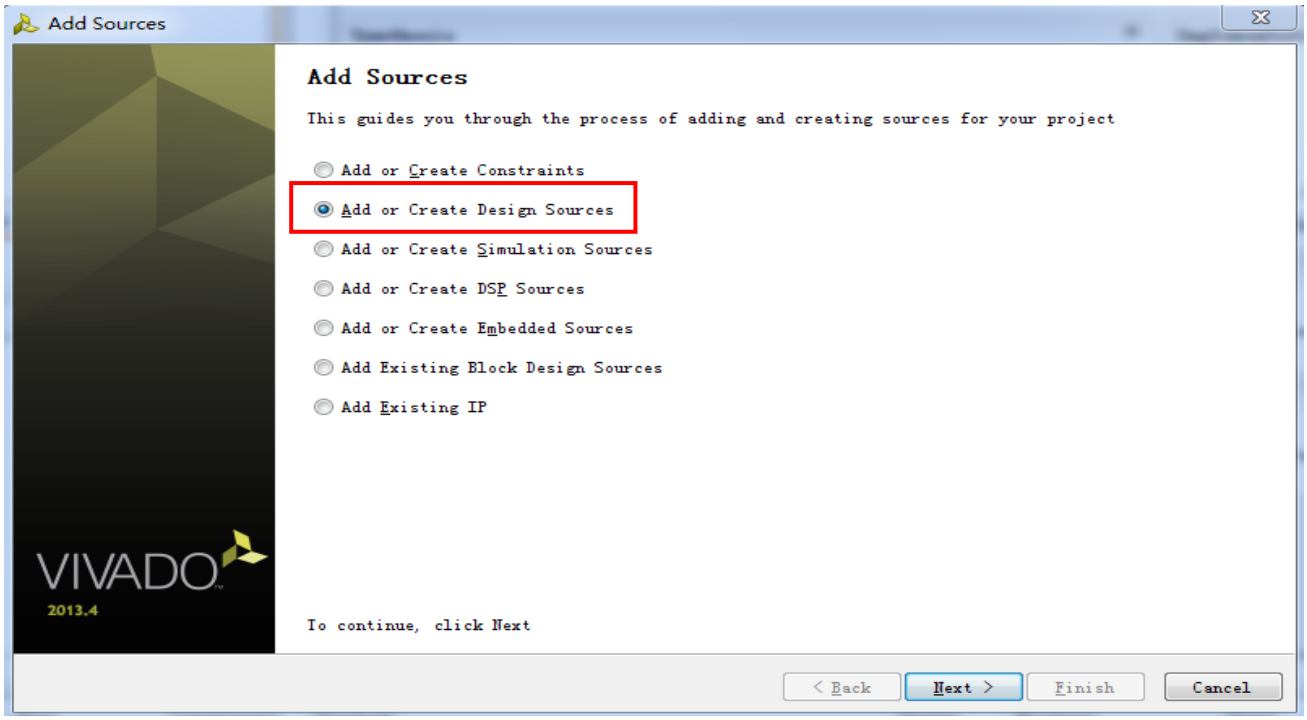
弹出如下窗口：



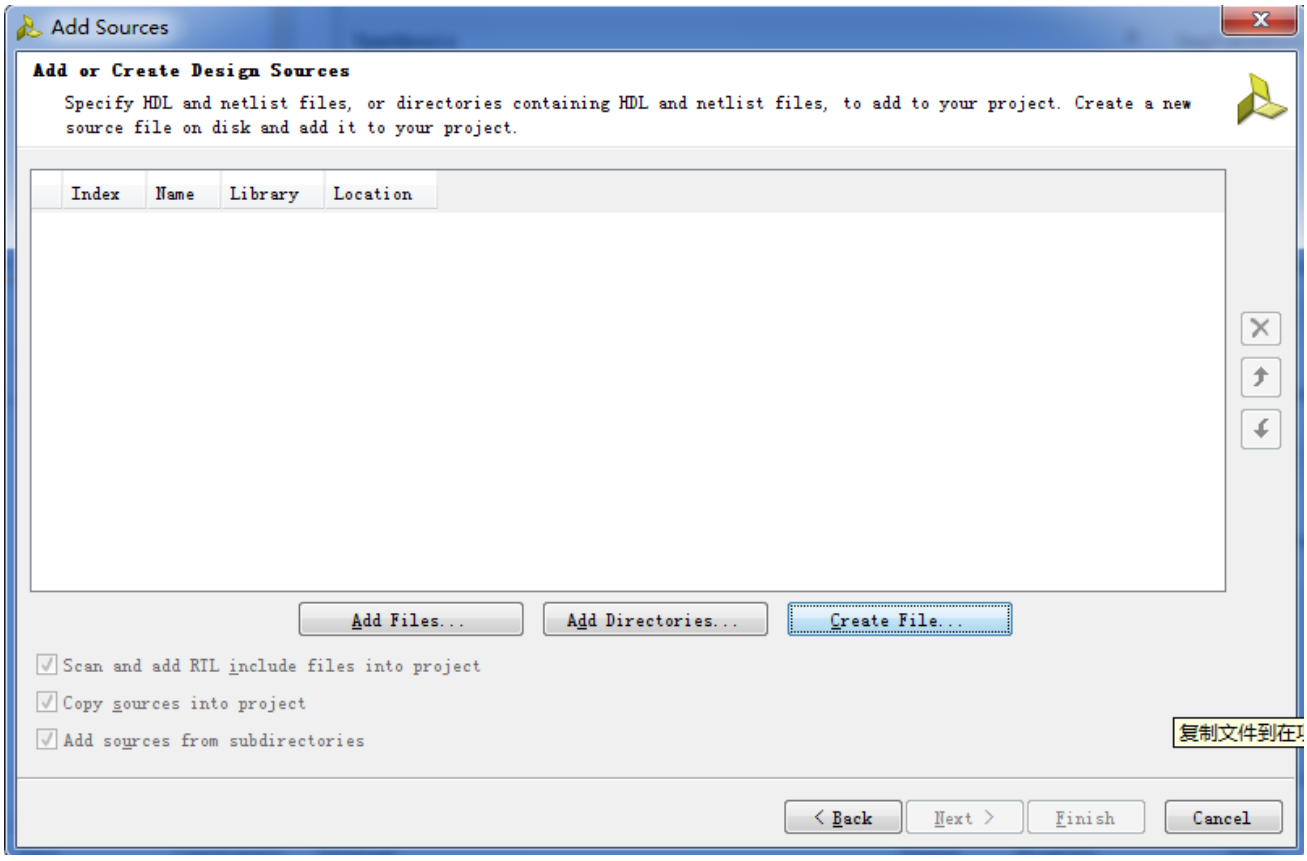
第四步：新建（添加）源文件



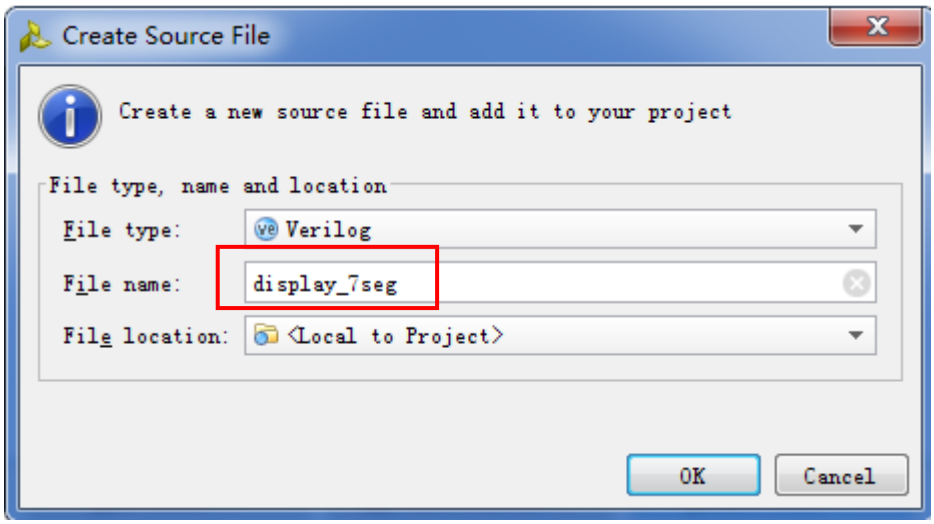
如上图，右击Design sourse选择Add Sourse，弹出下图窗口：



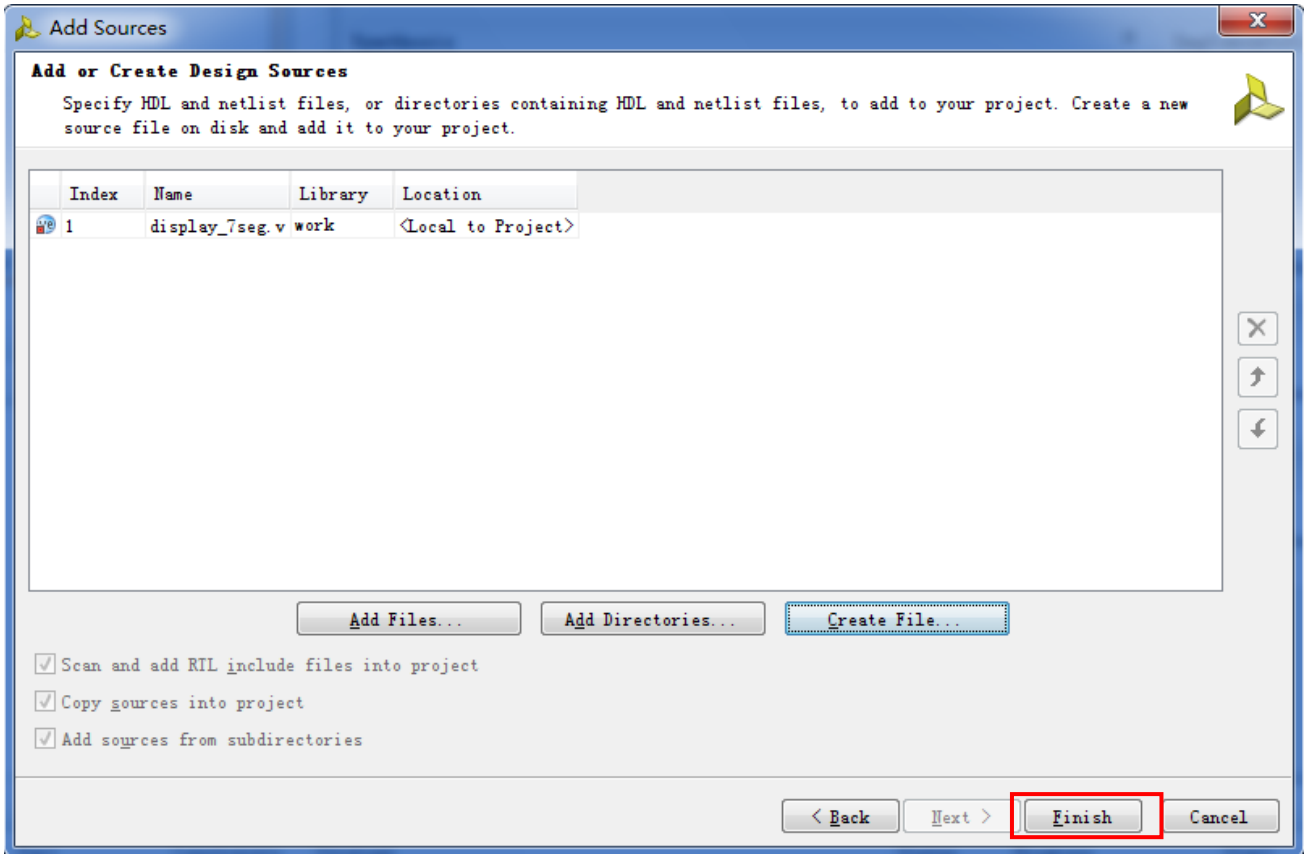
如上图选择第二个选项。弹出下图窗口：



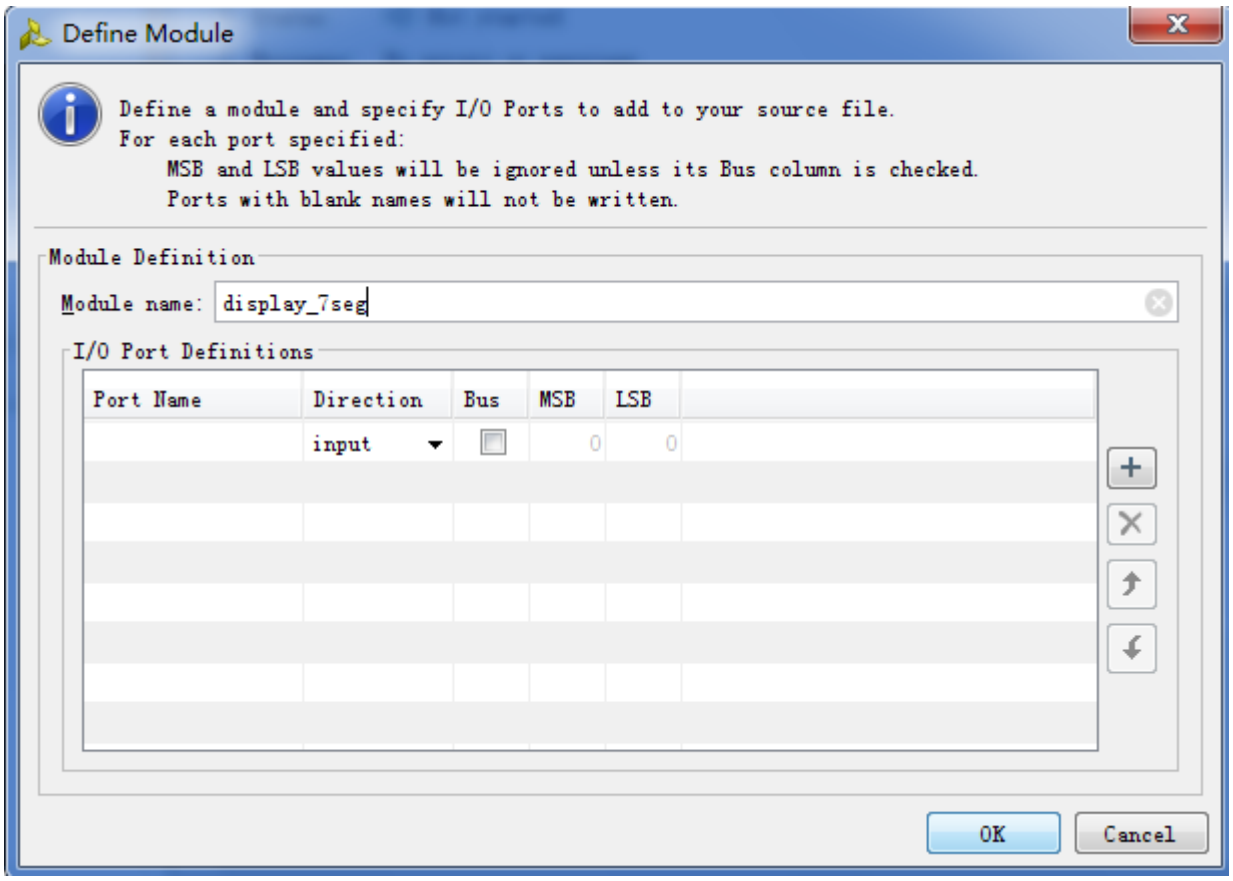
在上图中选择Create File…在弹出下面窗口填写新建源文件名称为display_7seg

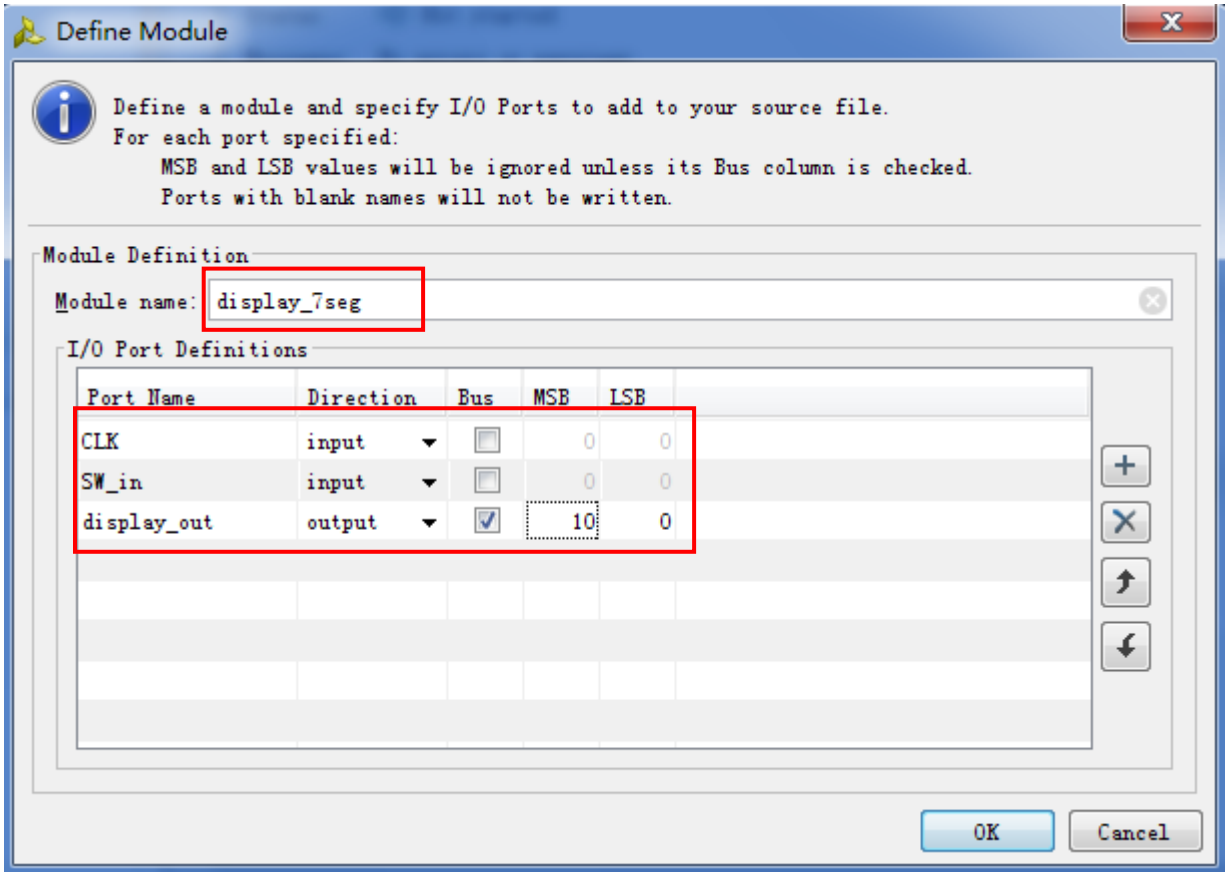


点击OK，弹出下图：

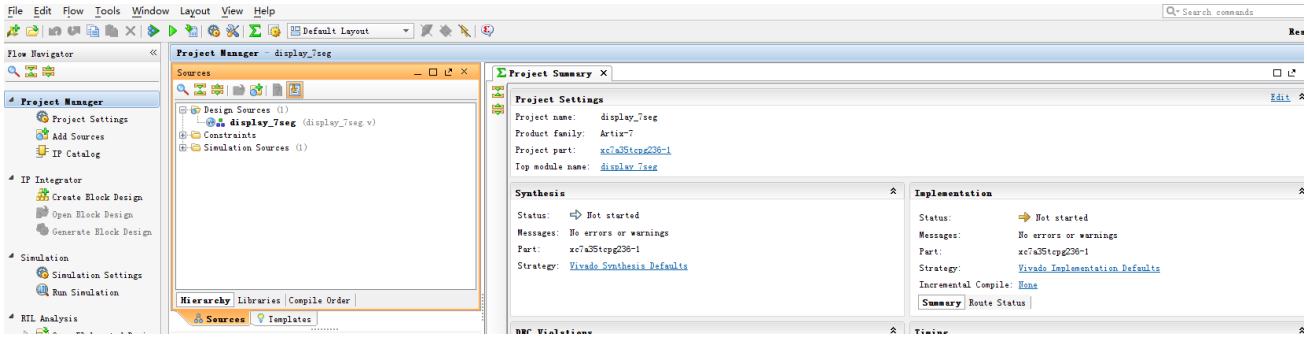


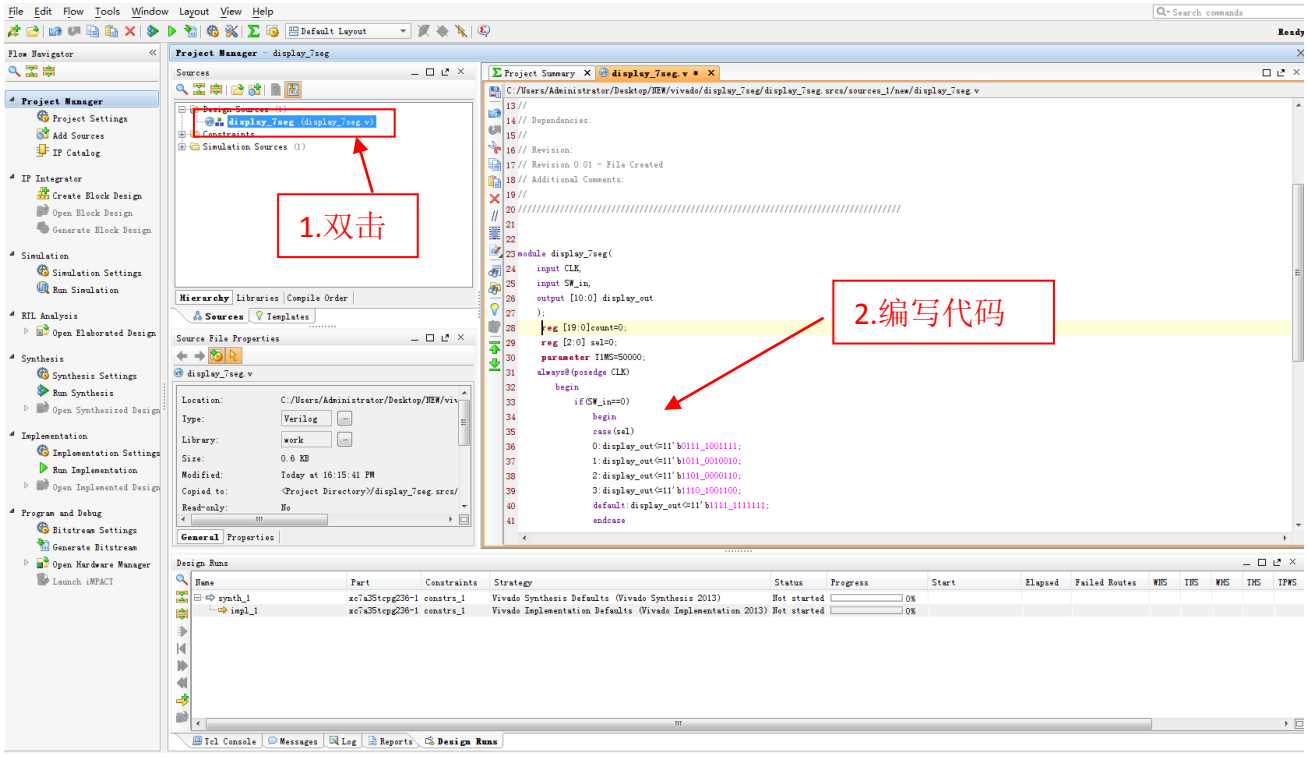
在上图中点击Finish在弹出窗口填写模块名称和端口，如下2张图所示


Define Module
Define a module and specify I/O Ports to add to your source file.
For each port specified:
MSB and LSB values will be ignored unless its Bus column is checked.
Ports with blank names will not be written.
Module Definition
Module name: display_7seg
I/O Port Definitions
Port Name
Direction
Bus
MSB
LSB
input
0
0
OK
Cancel


Define Module
Define a module and specify I/O Ports to add to your source file.
For each port specified:
MSB and LSB values will be ignored unless its Bus column is checked.
Ports with blank names will not be written.
Module Definition
Module name: display_7seg
I/O Port Definitions
Port Name
Direction
Bus
MSB
LSB
CLK
input
SW_in
input
display_out
output
10
0
OK
Cancel

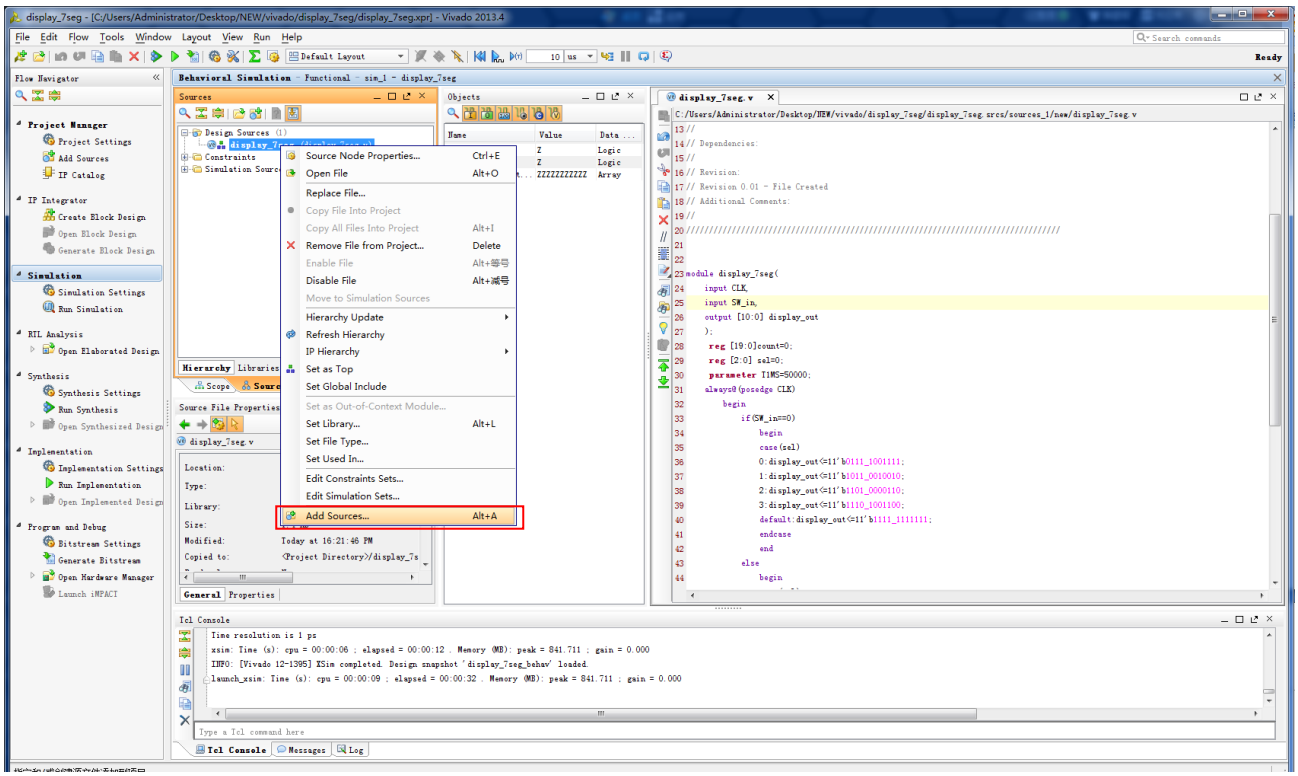
设置完成，弹出下图：



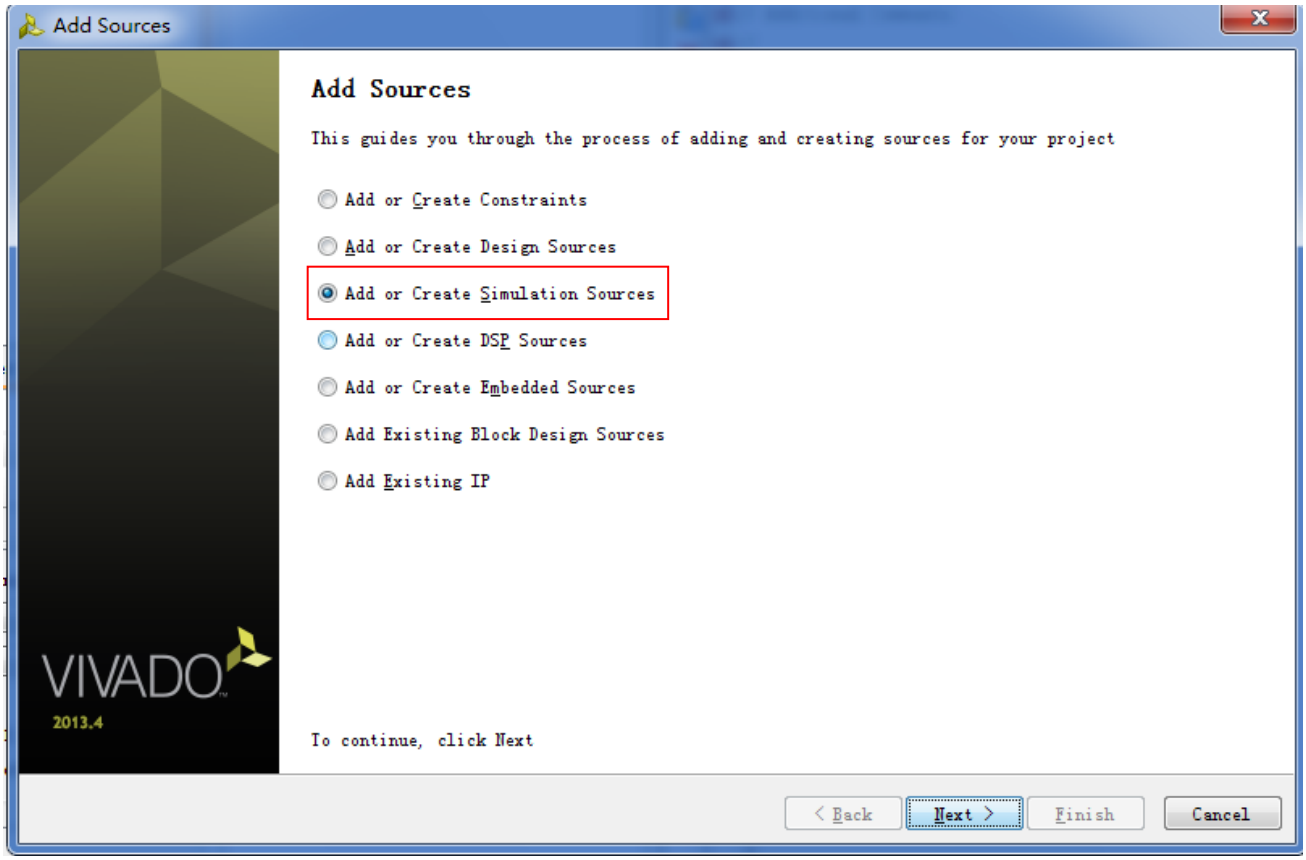


第五步：新建（添加）仿真文件

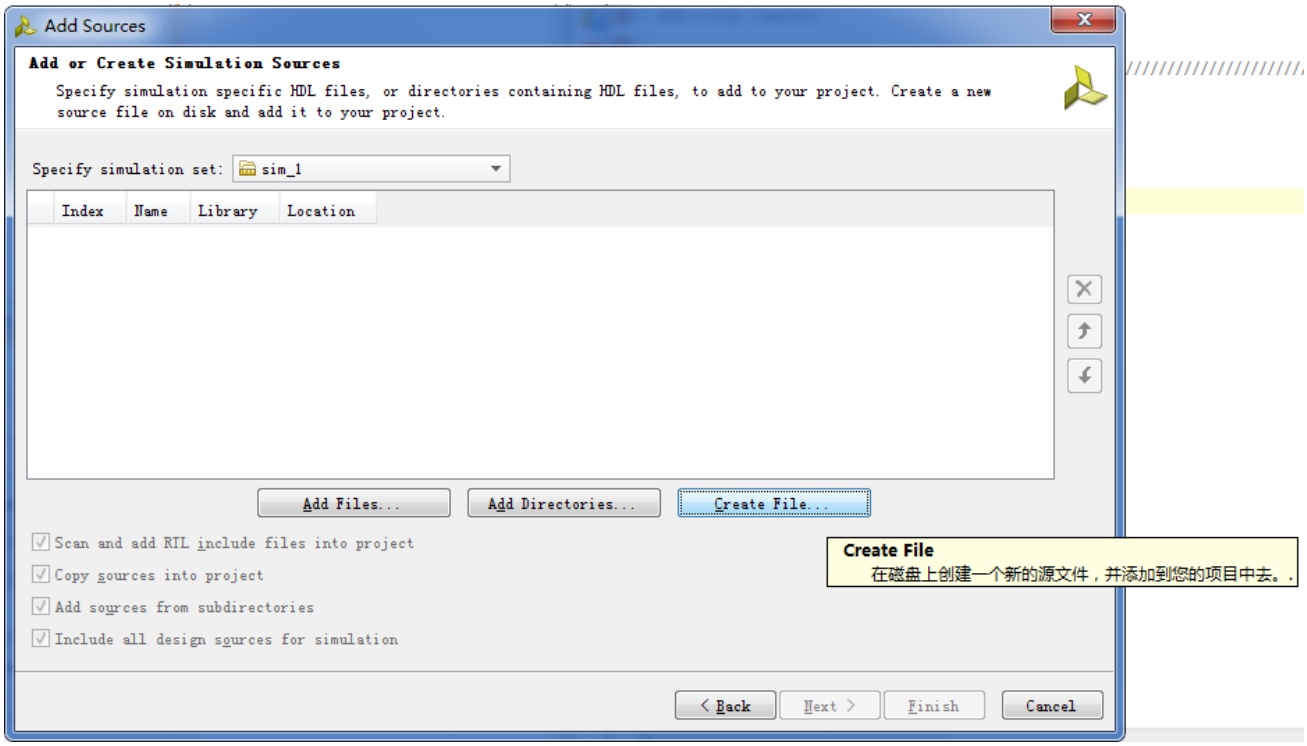
如下图，右击源文件选择Add Sourse



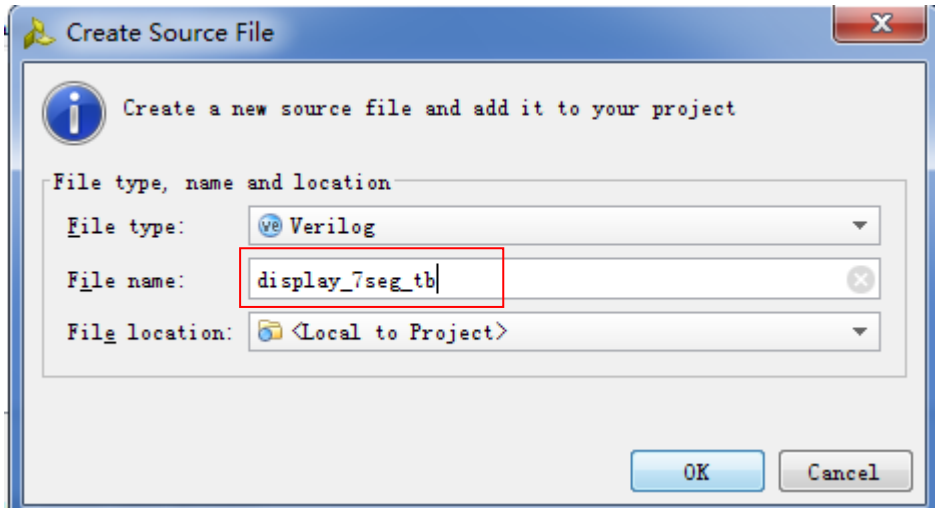
如下图，选择第三个新建仿真文件：



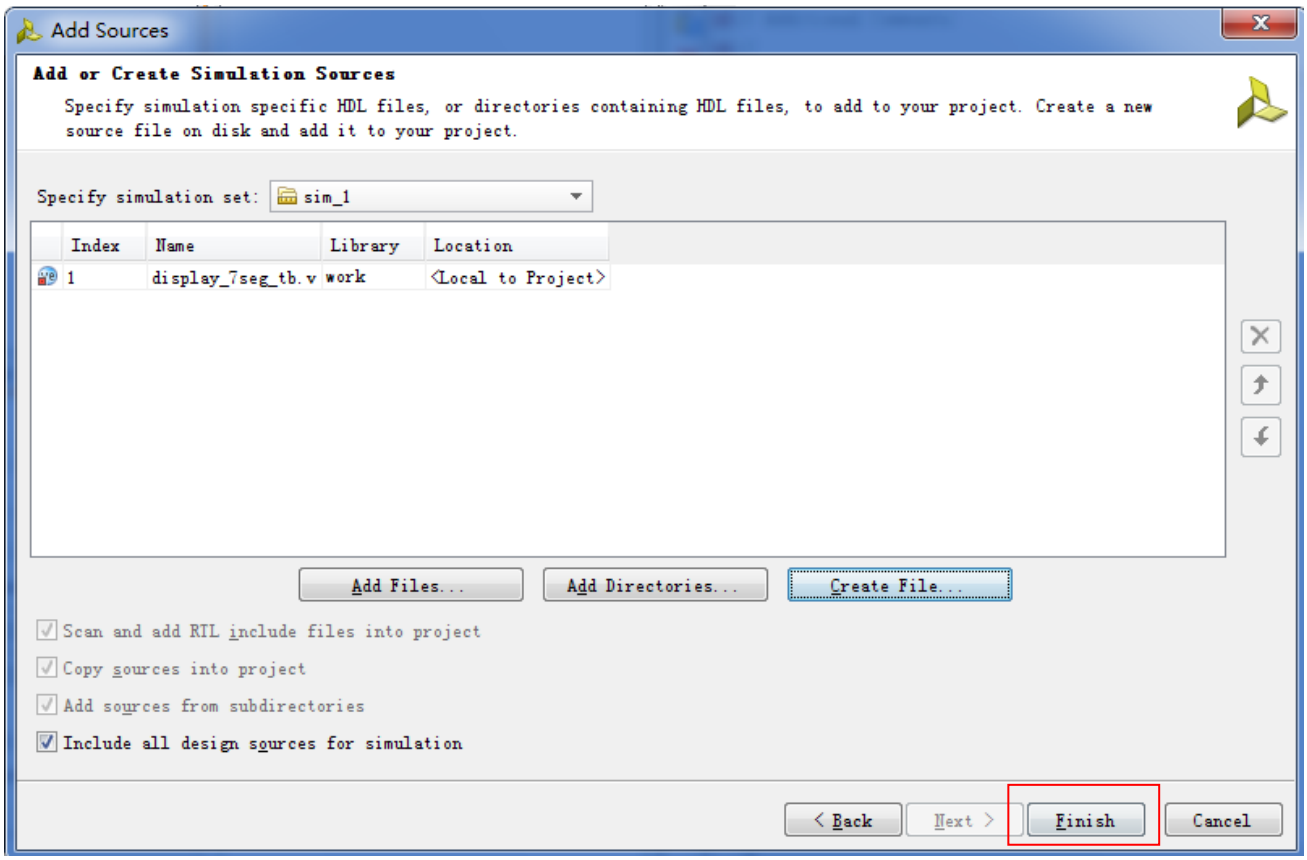
在下图的弹出窗口选择Create File…

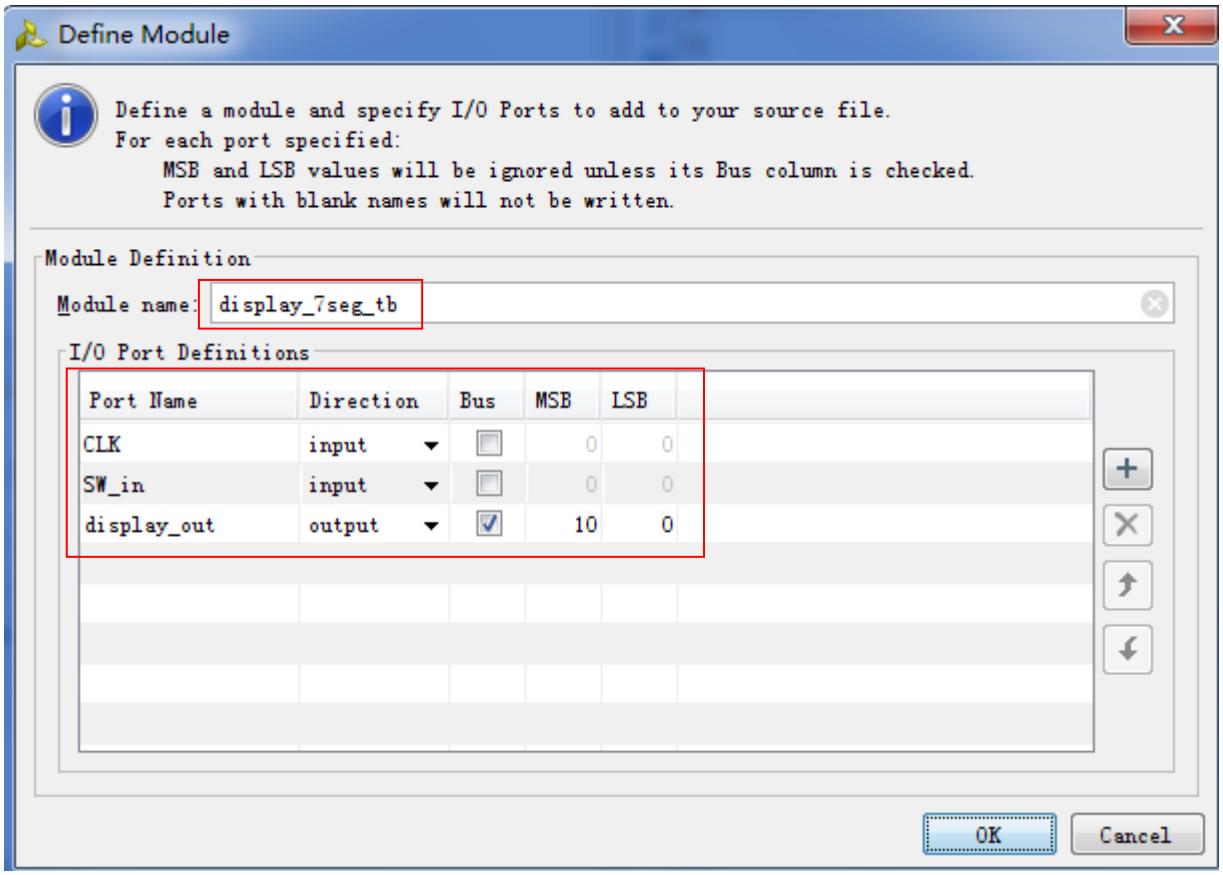


在下图的弹出窗口，设置文件名为display_7seg_tb

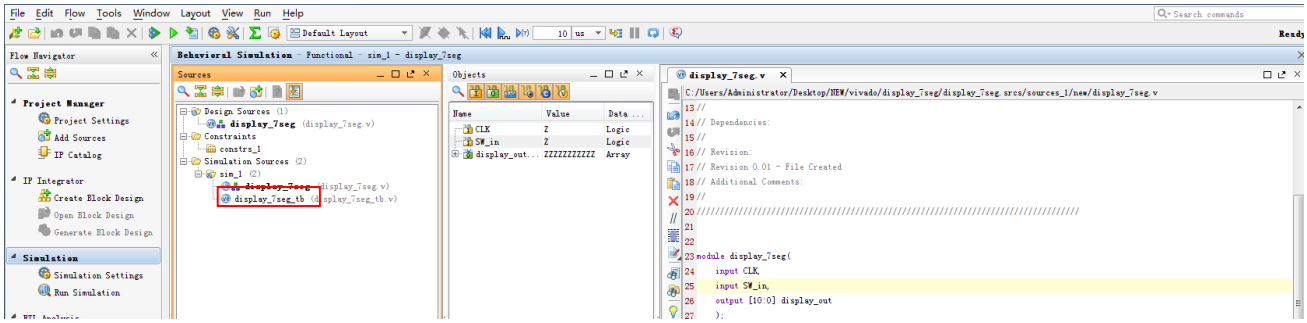


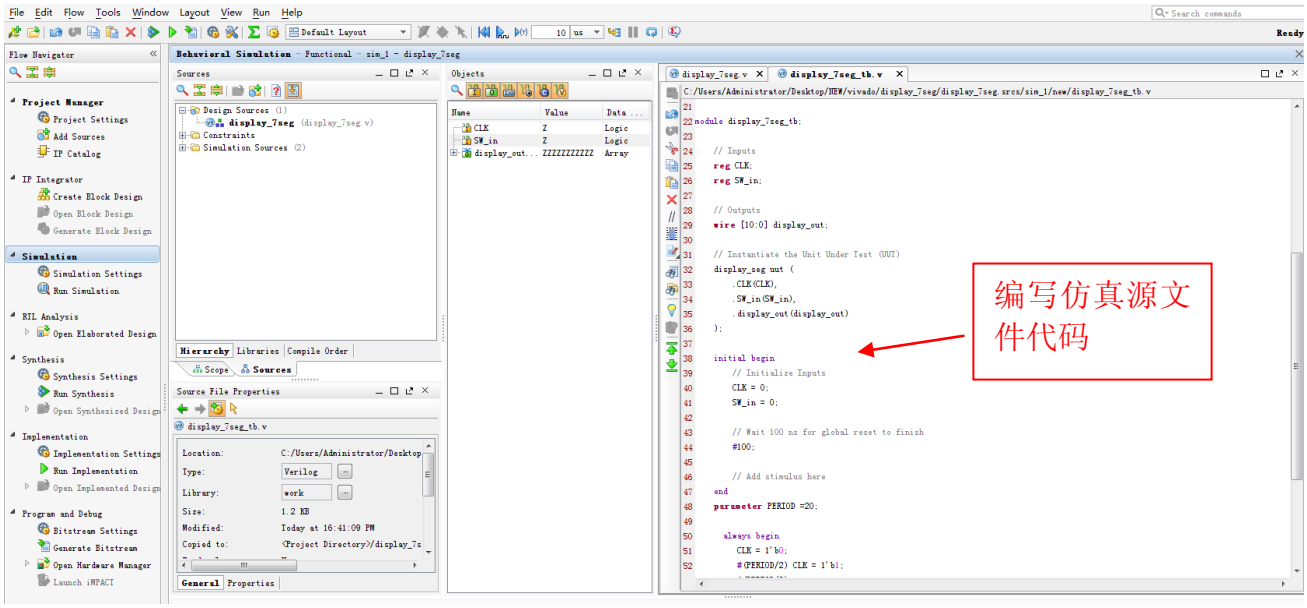
在接下来的2张图中的编写直接参考源文件的步骤


Add Sources
Add or Create Simulation Sources
Specify simulation specific HDL files, or directories containing HDL files, to add to your project. Create a new source file on disk and add it to your project.
Specify simulation set: sim_1
Index
Name
Library
Location
1
display_7seg_tb.v
work
<Local to Project>
Add Files...
Add Directories...
Create File...
Scan and add RTL include files into project
Copy sources into project
Add sources from subdirectories
Include all design sources for simulation
< Back
Next >
Finish
Cancel

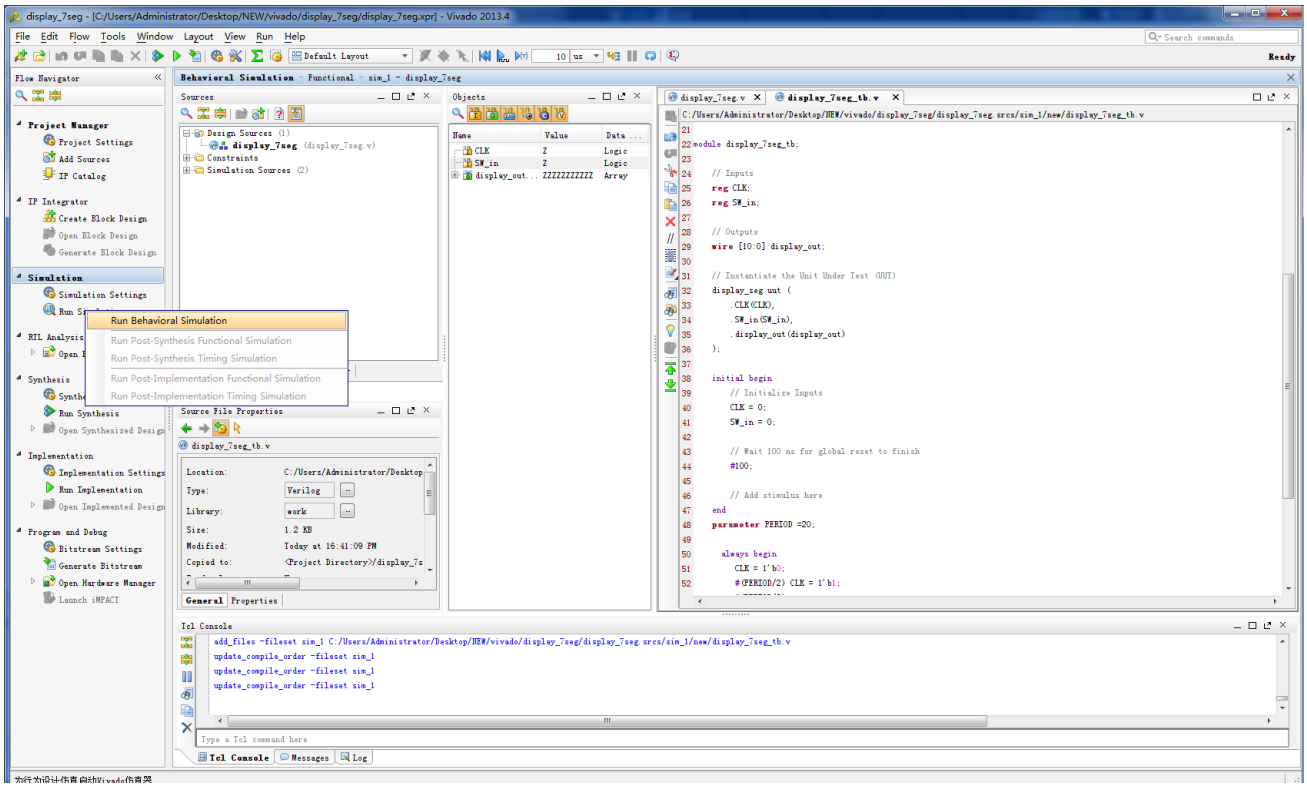

Define Module
Define a module and specify I/O Ports to add to your source file.
For each port specified:
MSB and LSB values will be ignored unless its Bus column is checked.
Ports with blank names will not be written.
Module Definition
Module name: display_7seg_tb
I/O Port Definitions
Port Name
Direction
Bus
MSB
LSB
CLK
input
0
0
SW_in
input
0
0
display_out
output
10
0
OK
Cancel

如下2张图，编写仿真源文件代码：

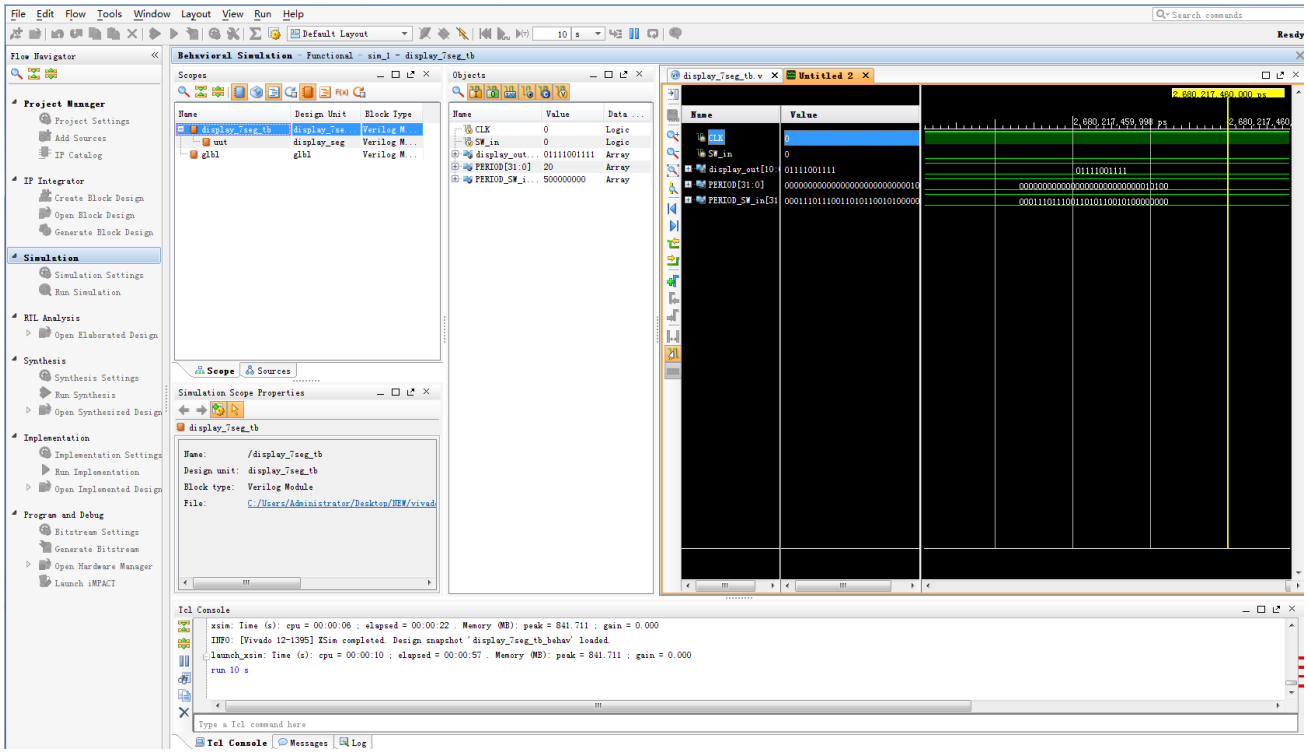


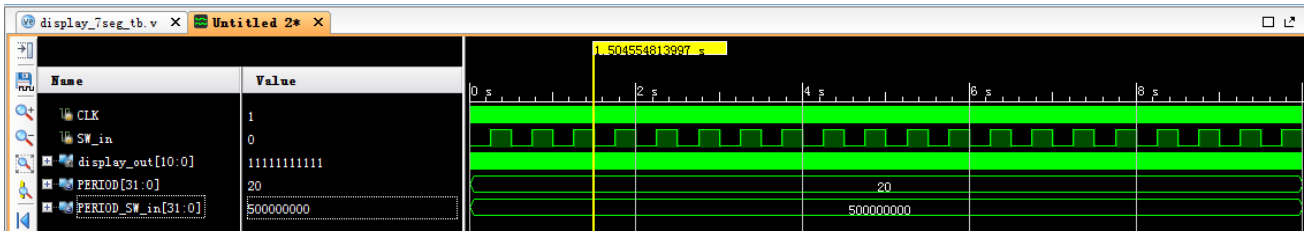


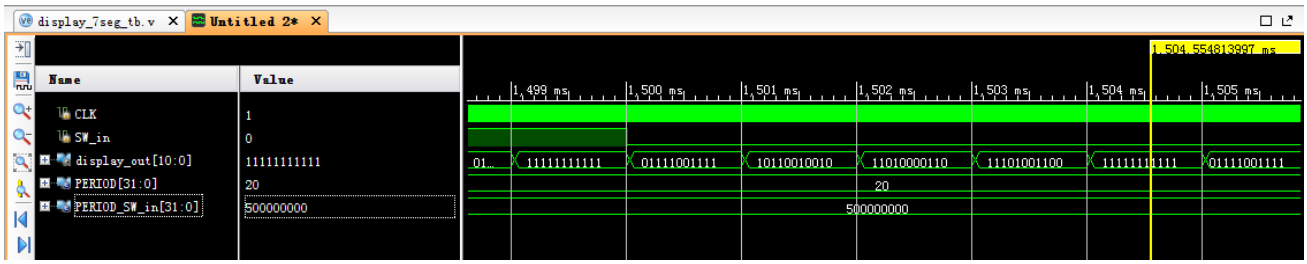
运行仿真：



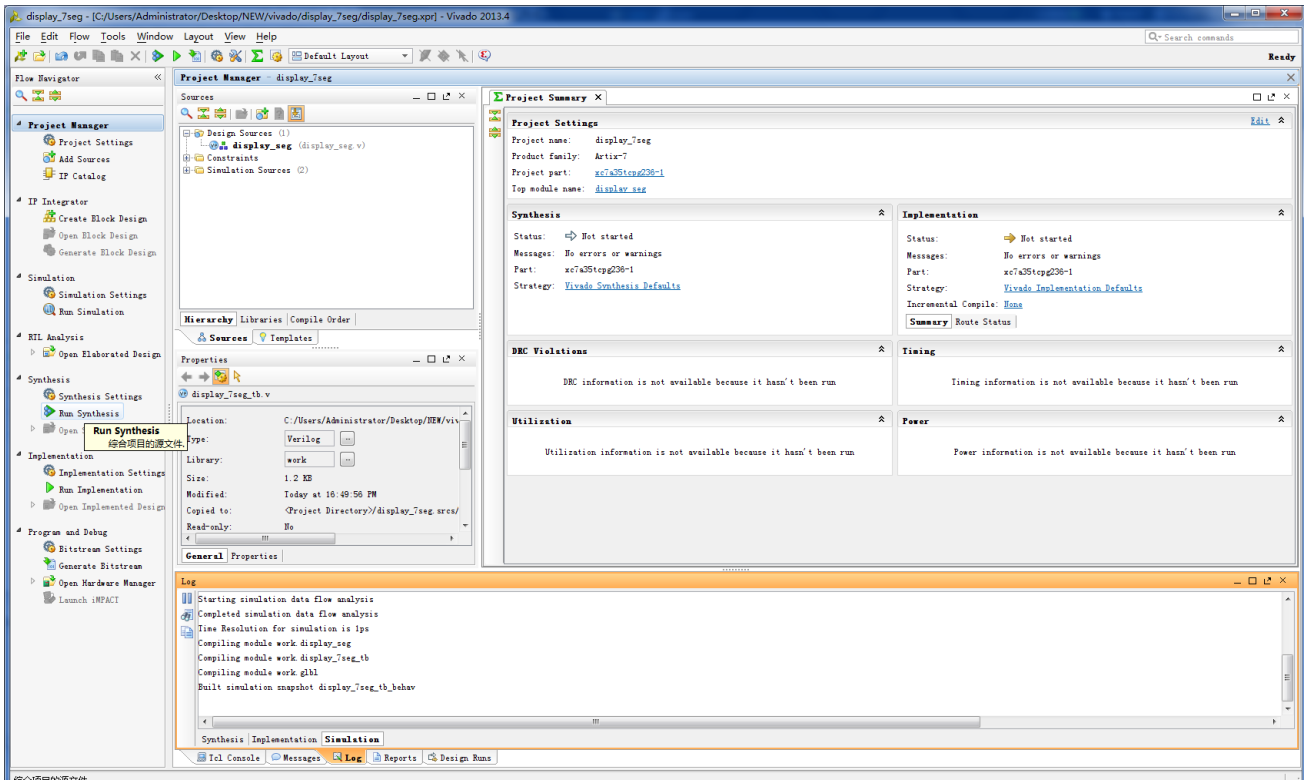
如上图右击Run Simulation，选择Run Behavioral Simulation，弹出仿真波形窗口如下3张图：



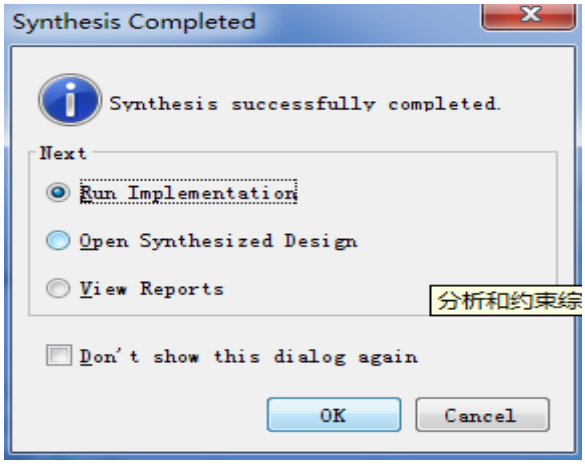




第六步：综合

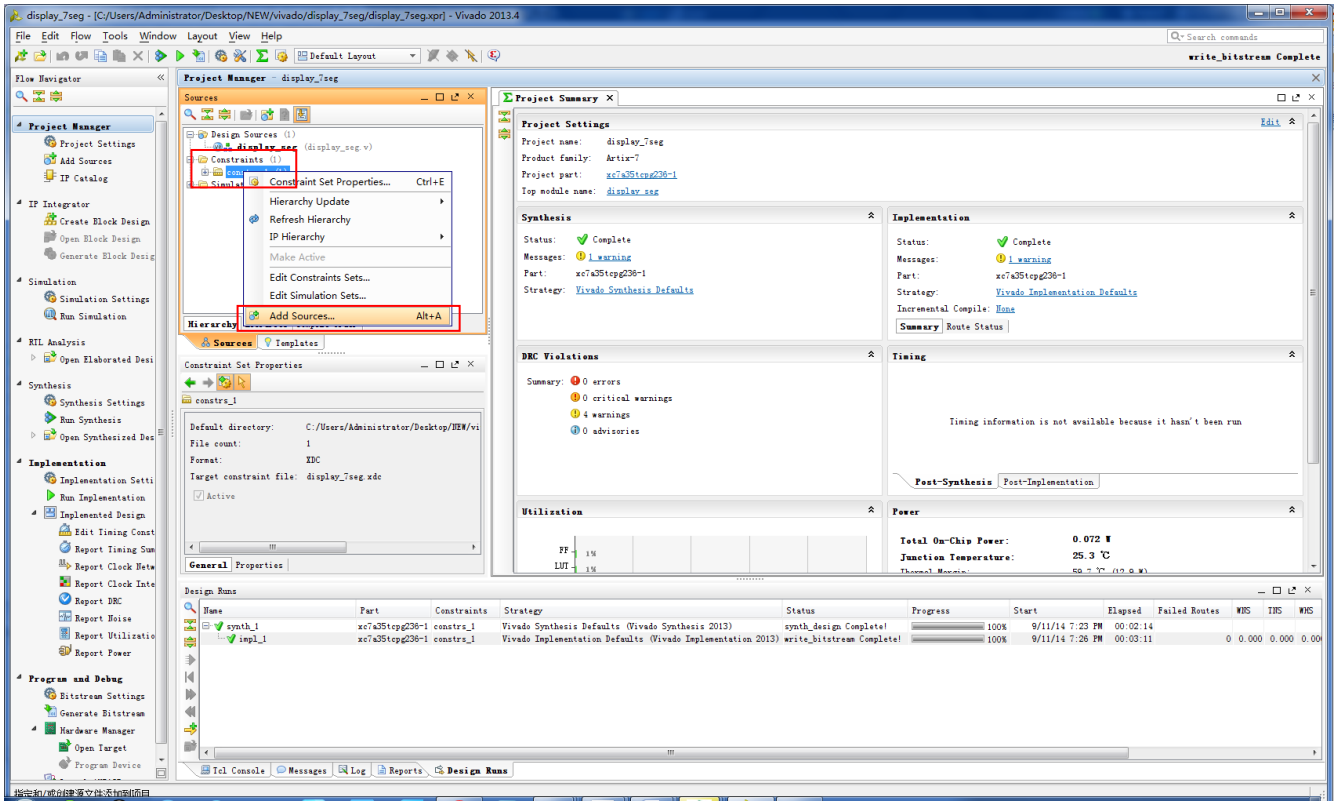


如上图单击Run Synthesis，完成后弹出下面窗口：

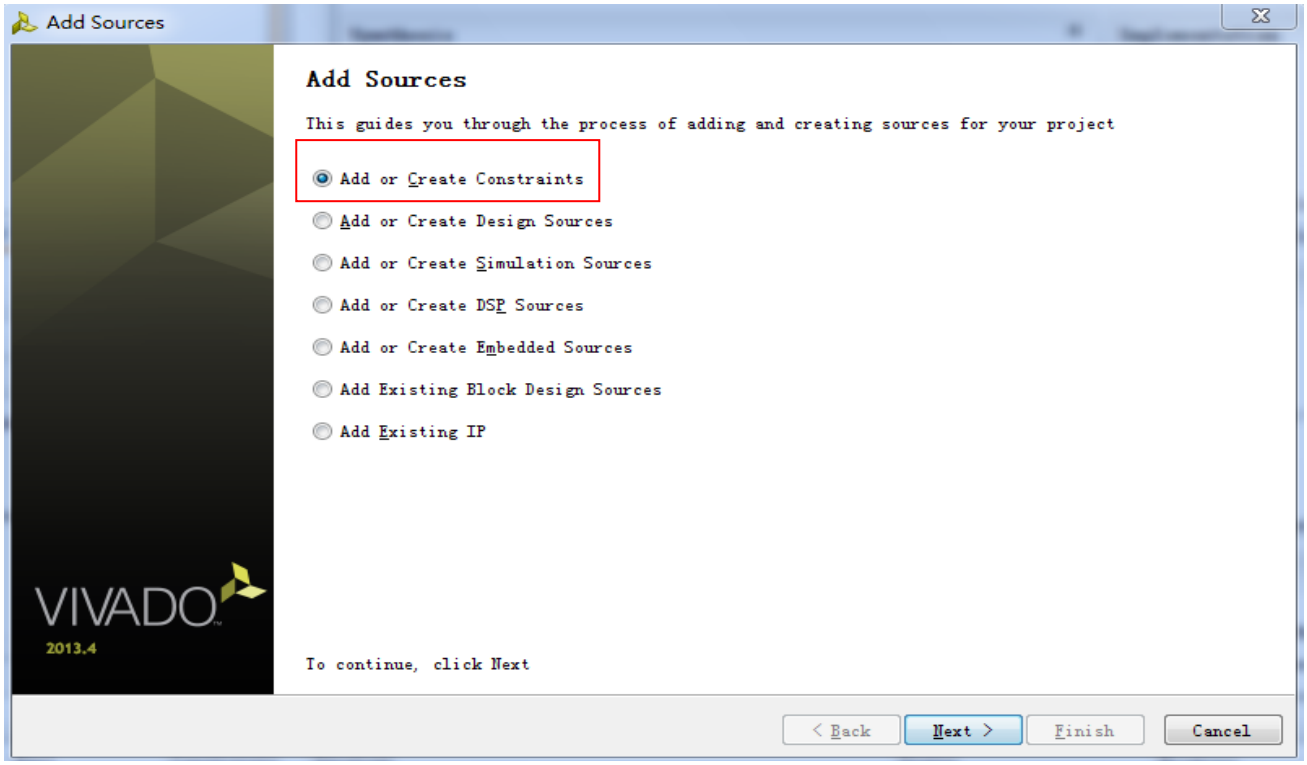


在上图中点击关闭

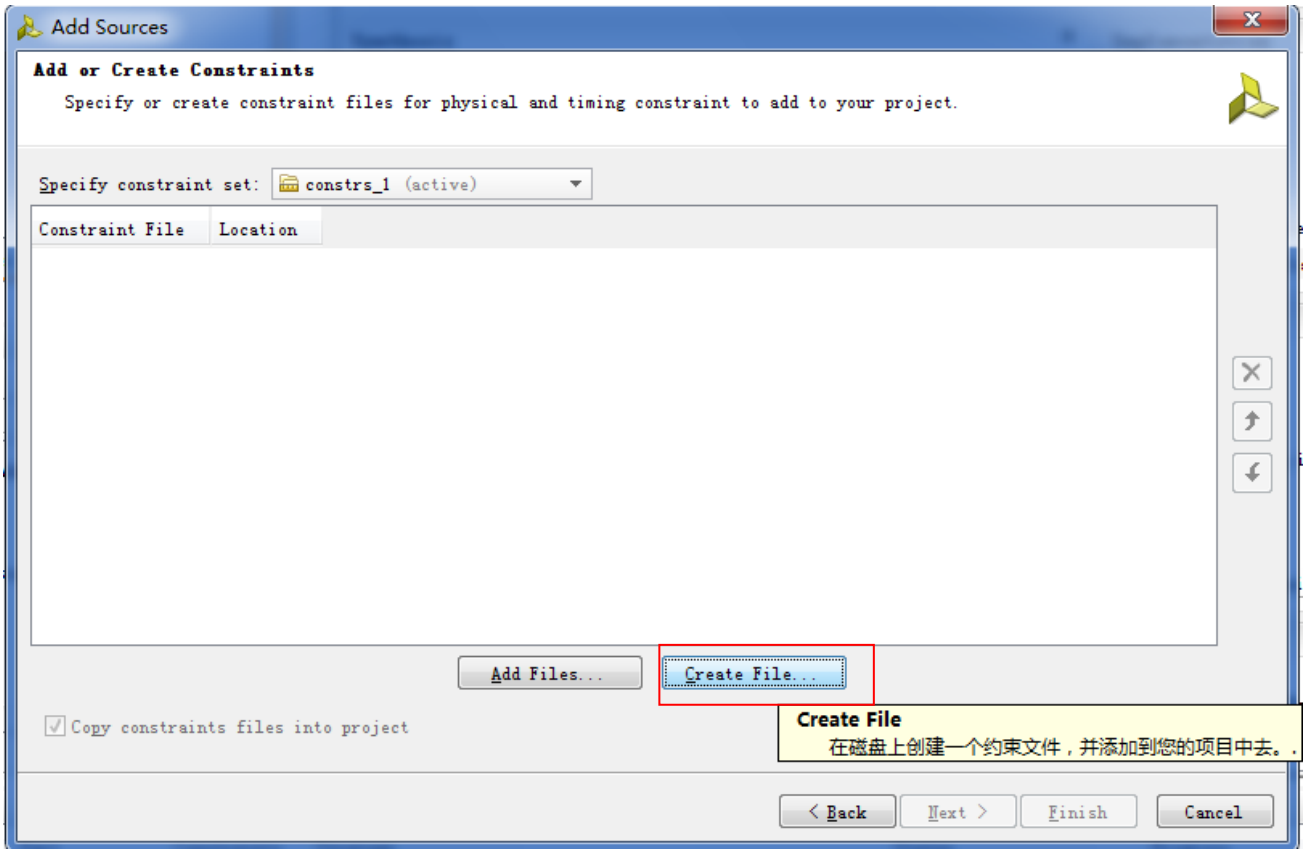
第七步：分配引脚



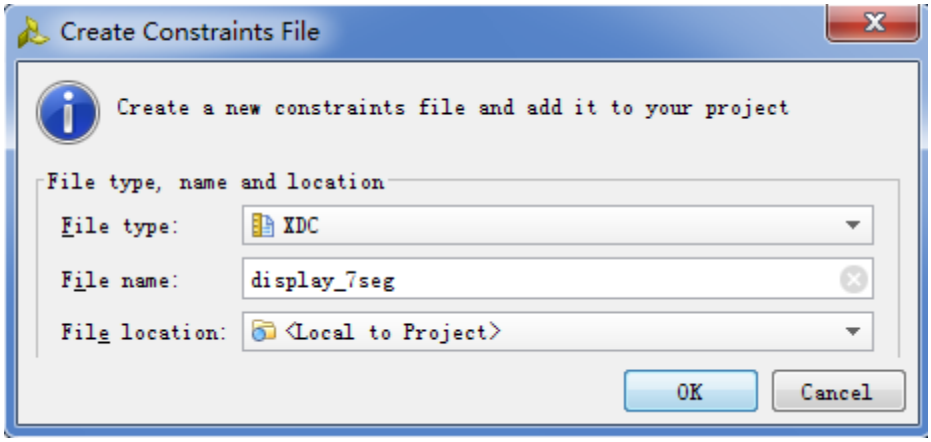
如上图，右击约束子目录下文件夹，选择Add Sourse...弹出下列窗口：



如上图选择第一个



如上图选择Create File…弹出下面的窗口，填写约束文件的文件名display_7seg

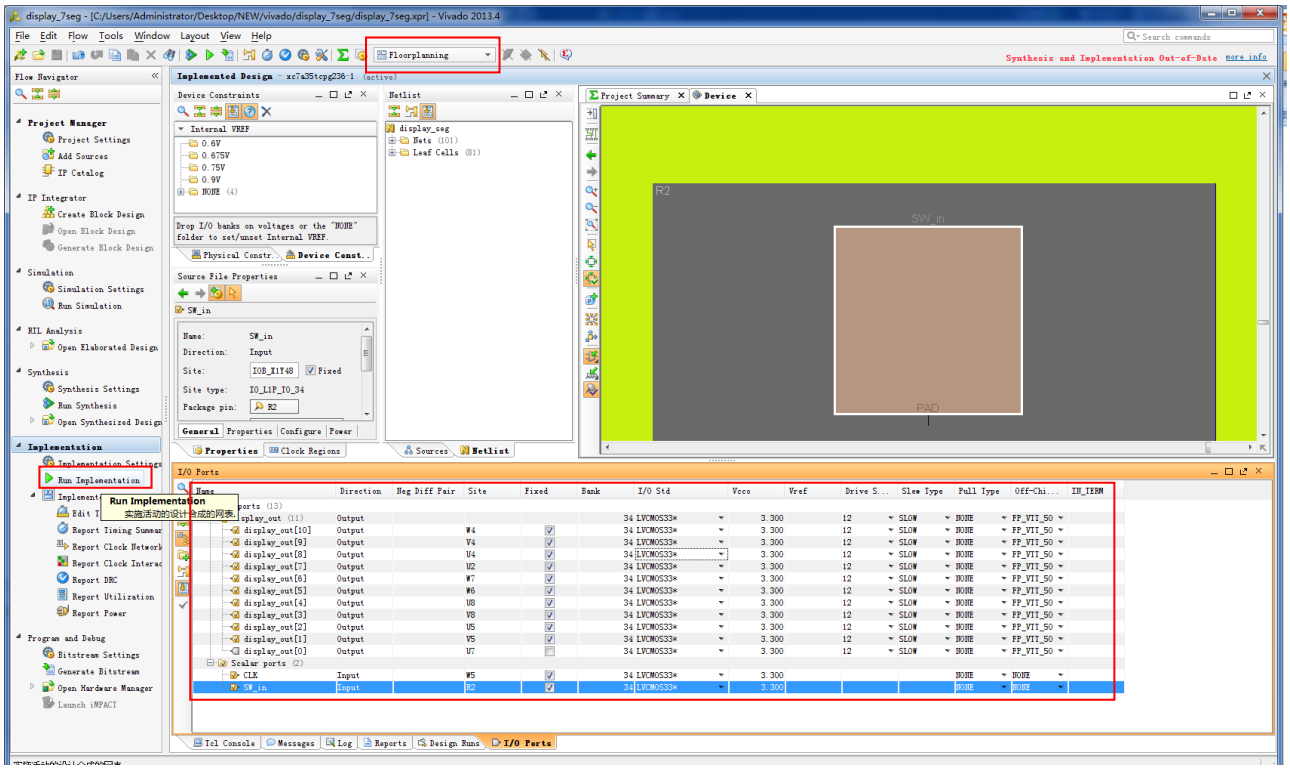


编写约束文件如下：

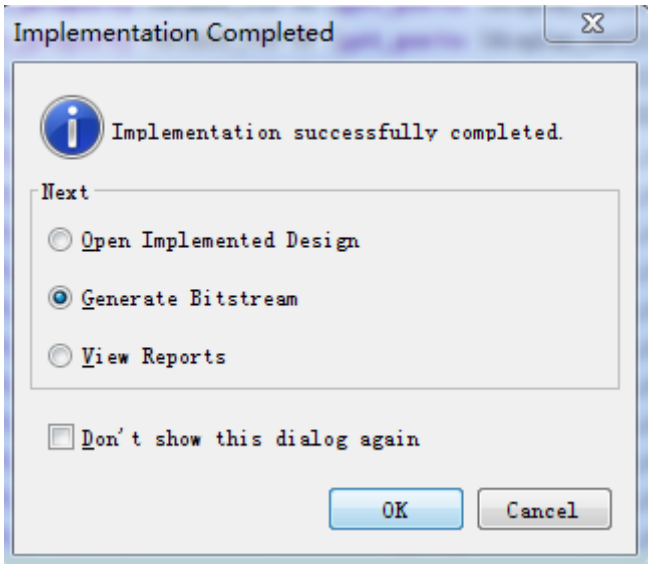
C:/Users/Administrator/Desktop/NEW/vivado/display_7seg/display_7seg.srcs/constrs_1/new/display_7seg.xdc

```
set_property PACKAGE_PIN W5 [get_ports CLK]
set_property PACKAGE_PIN V17 [get_ports SW_in]
set_property IOSTANDARD LVCMOS33 [get_ports SW_in]
set_property IOSTANDARD LVCMOS33 [get_ports CLK]
set_property PACKAGE_PIN W4 [get_ports {display_out[10]}]
set_property PACKAGE_PIN V4 [get_ports {display_out[9]}]
set_property PACKAGE_PIN U4 [get_ports {display_out[8]}]
set_property PACKAGE_PIN U2 [get_ports {display_out[7]}]
set_property PACKAGE_PIN W7 [get_ports {display_out[6]}]
set_property PACKAGE_PIN W6 [get_ports {display_out[5]}]
set_property PACKAGE_PIN U8 [get_ports {display_out[4]}]
set_property PACKAGE_PIN V8 [get_ports {display_out[3]}]
set_property PACKAGE_PIN U5 [get_ports {display_out[2]}]
set_property PACKAGE_PIN V5 [get_ports {display_out[1]}]
set_property PACKAGE_PIN U7 [get_ports {display_out[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[9]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[8]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[10]}]
```

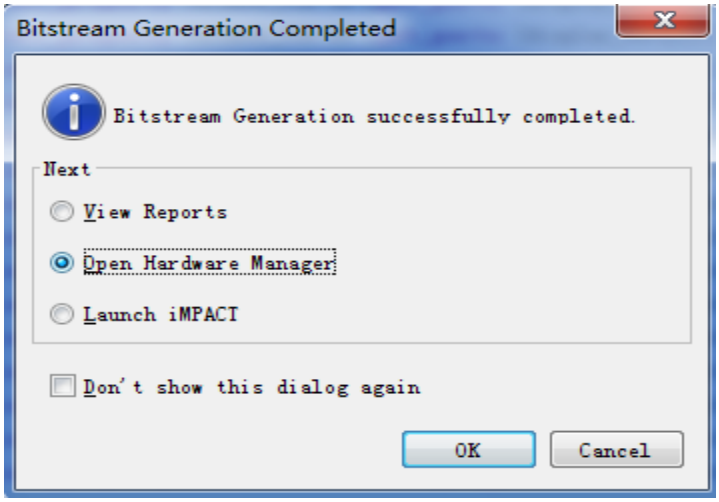
在Floorplanning窗口的视图如下：



在上图中单击Run implementation，运行完成后弹出下面窗口：

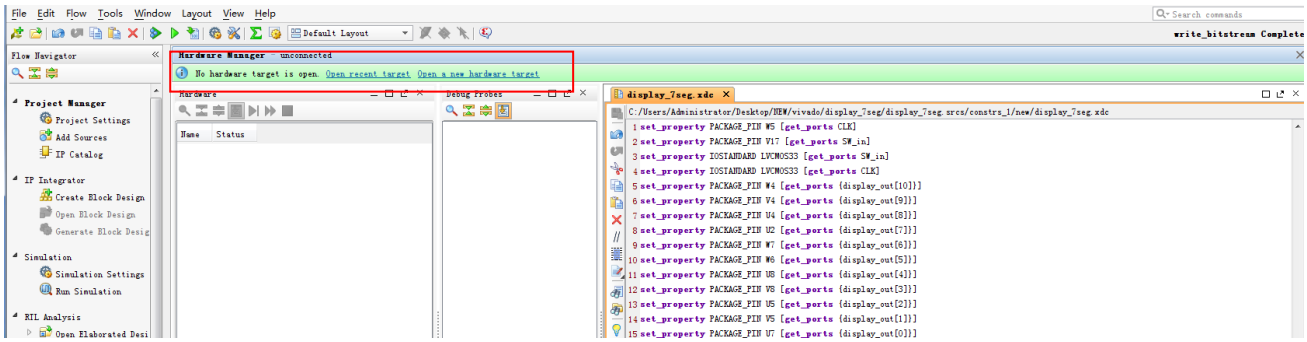


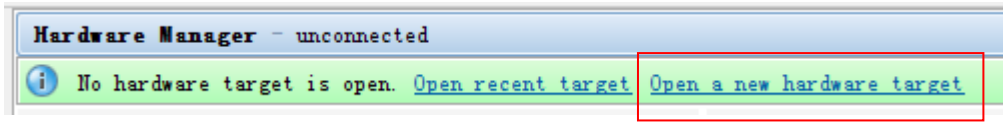
上图，选择Generate Bitstream，运行完成弹出下图：

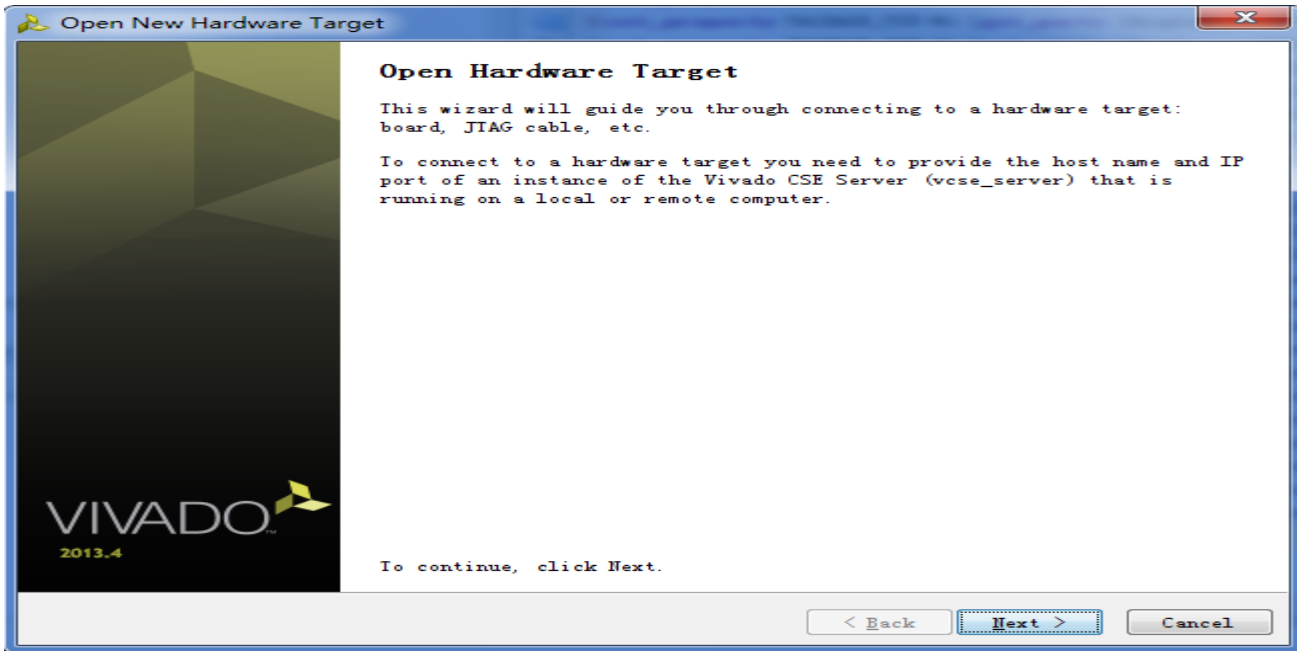


运行完成，在上图中选择Open Hardware Manager，点击下图中的Open a new hardware target

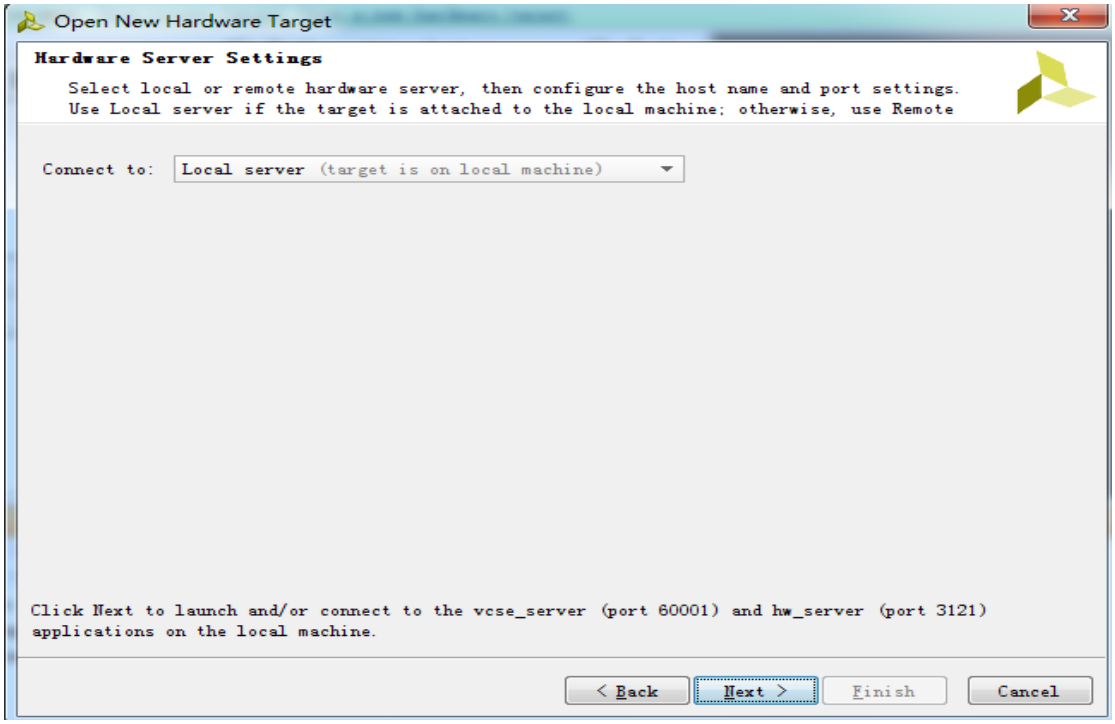
第八步：下载程序到FPGA



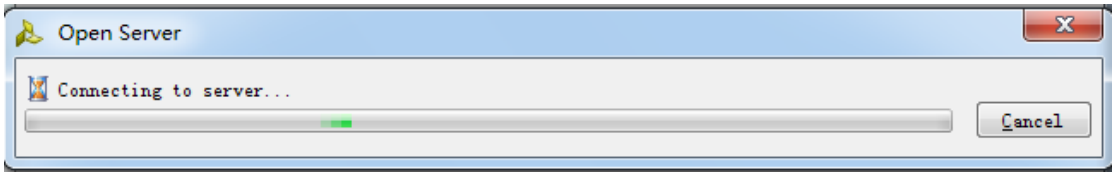


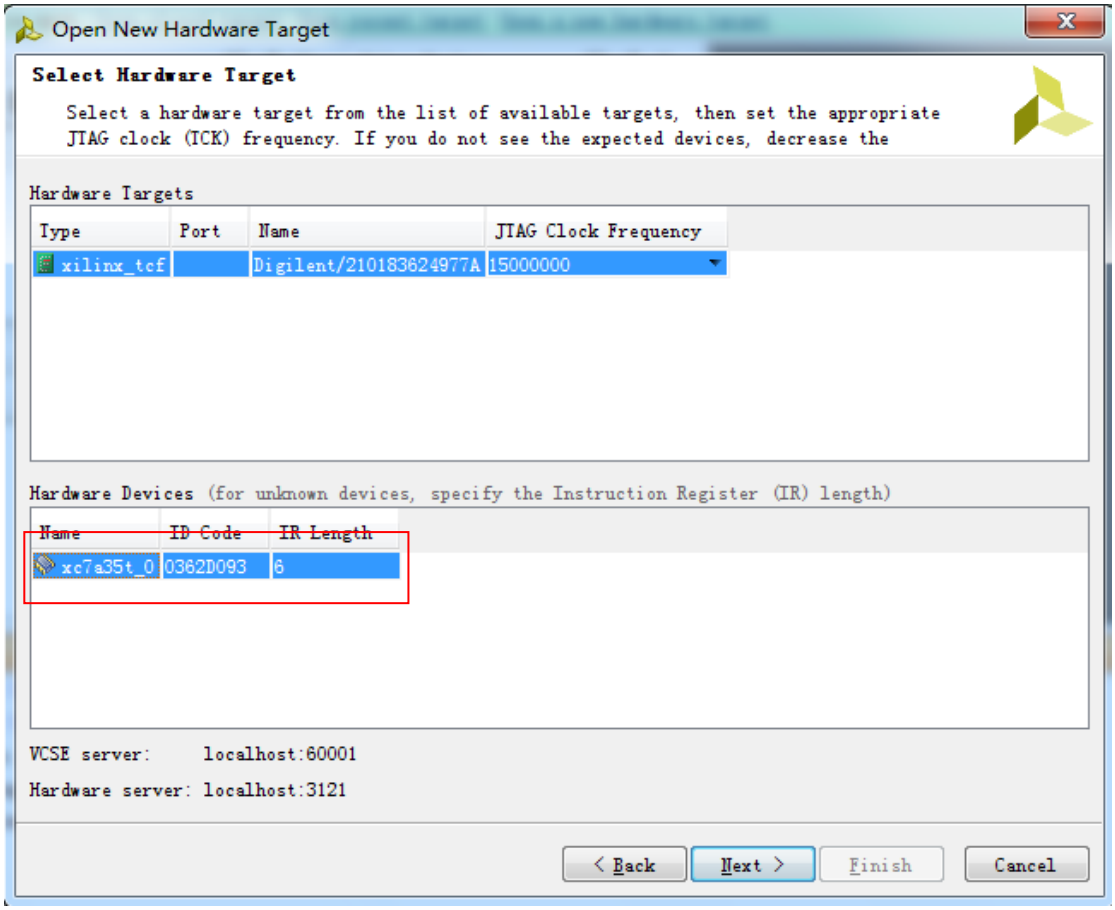


在上图中，单击Next

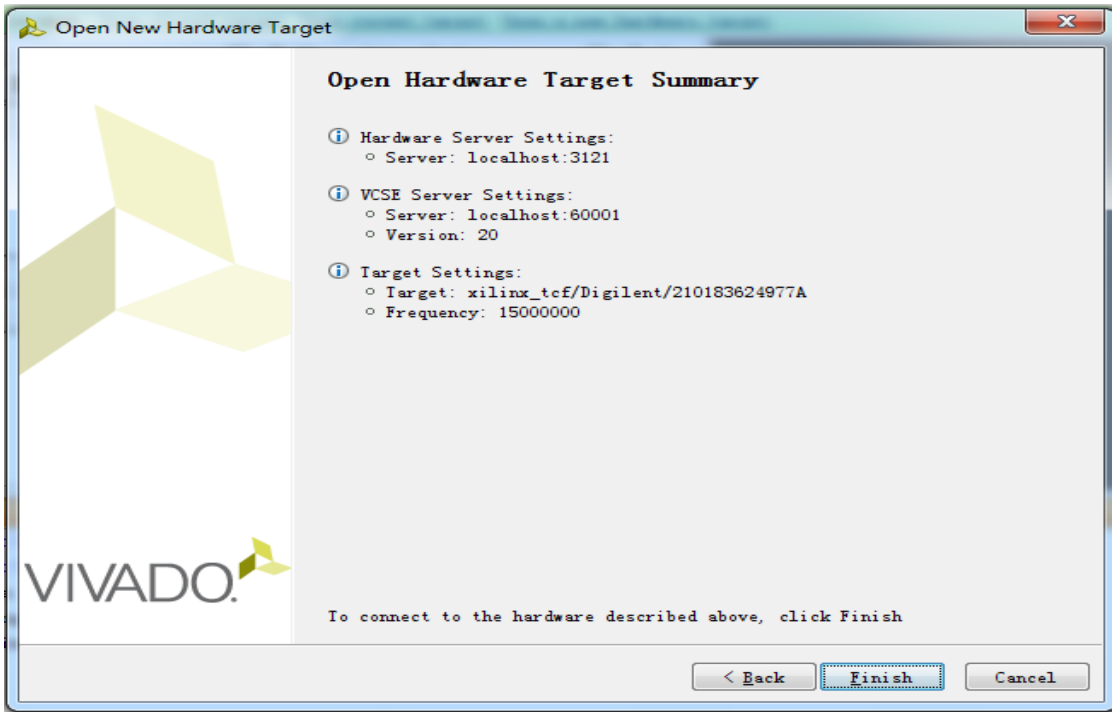


在上图中，单击Next

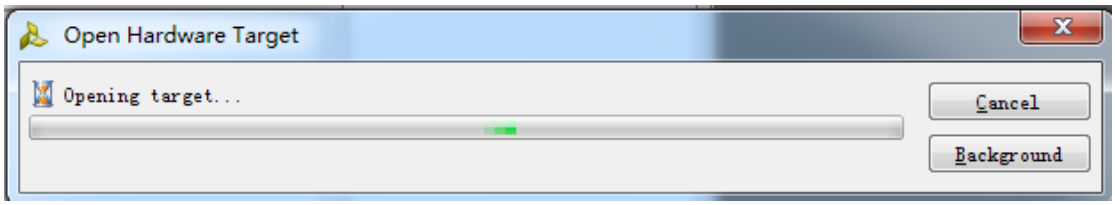




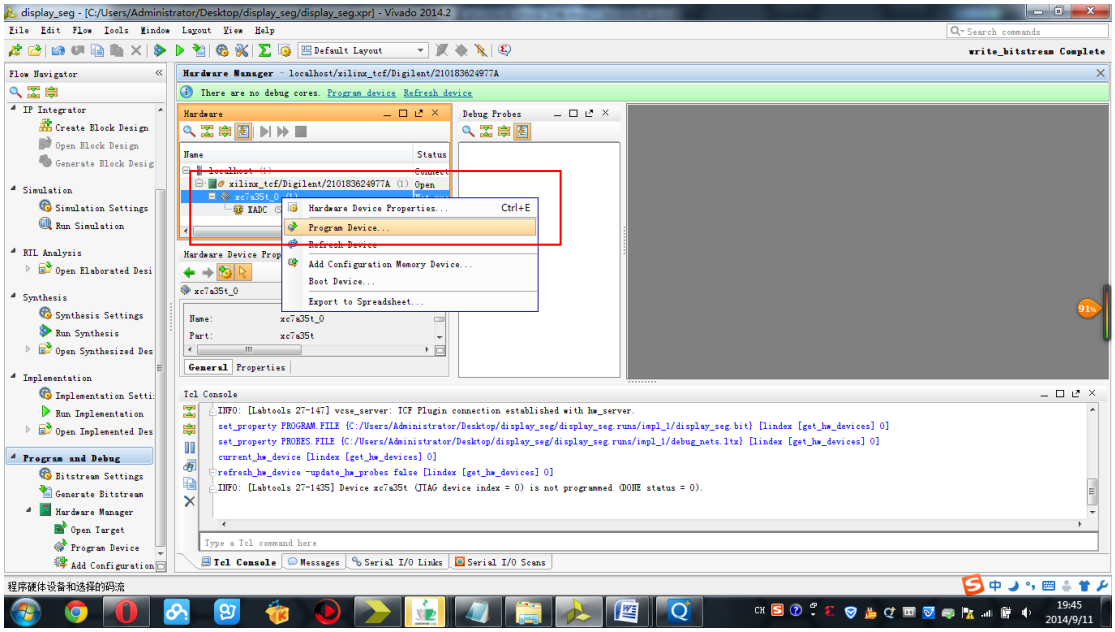
如上图，vivado 已经连接我们的开发板，并且已经识别我们的 FPHA 芯片型号。



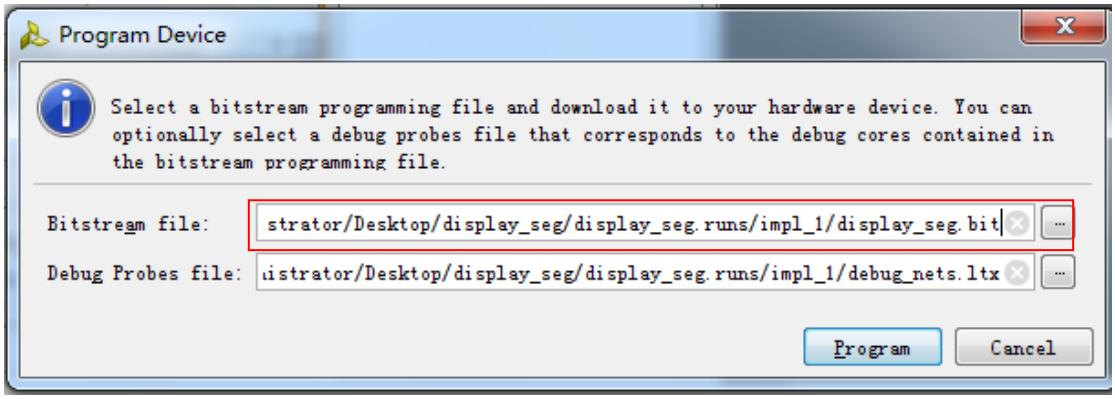
点击 Finish



在弹出的窗口（如下图）右击 FPGA 芯片选择 Program Device

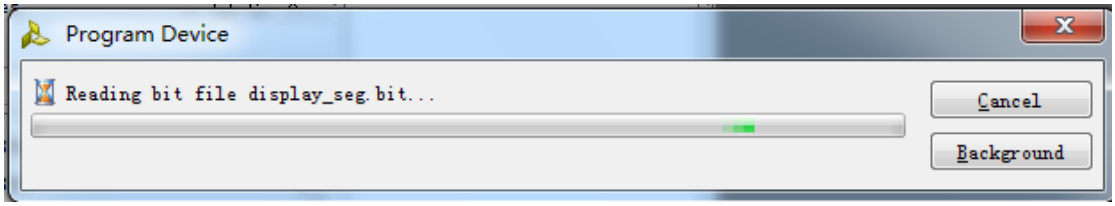


选择.bit 文件，在工程目录下选择前面生成的.bit 文件：



（上图中实为 display_7seg.bit），对于 Debug Probes file 框，如果是第一次操作应该是空白的，保持空白即可。

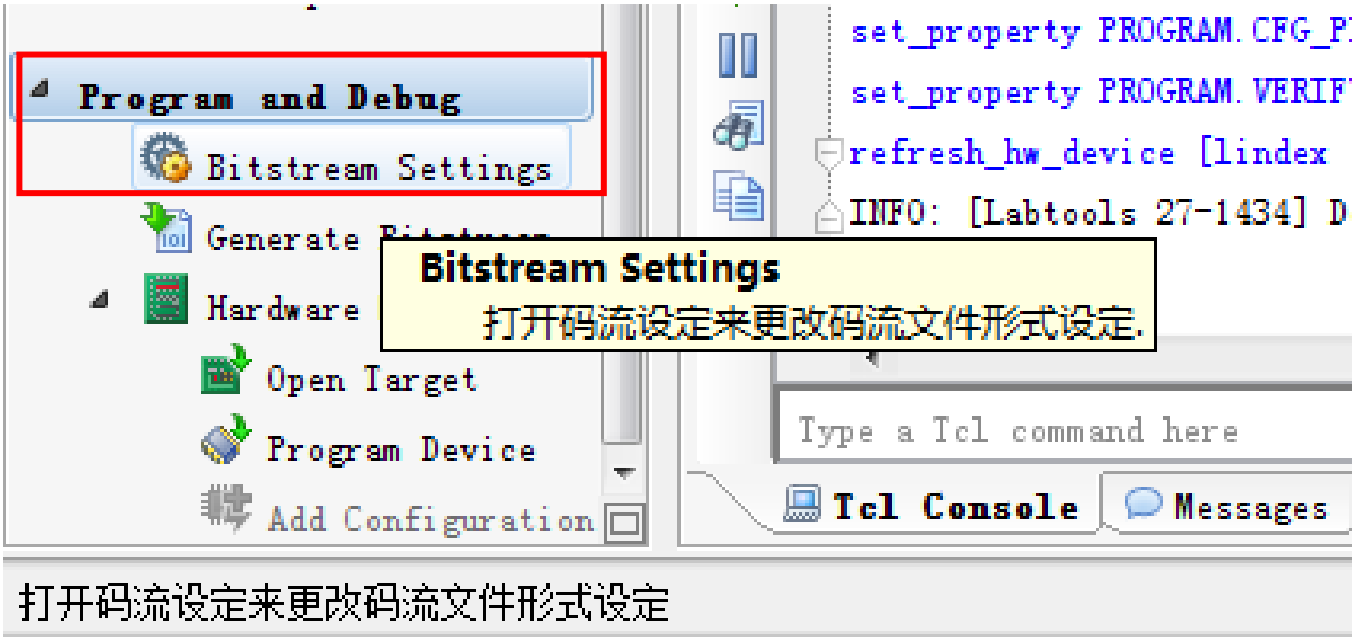
点击 program



下载完成OK，开发板演示即可。

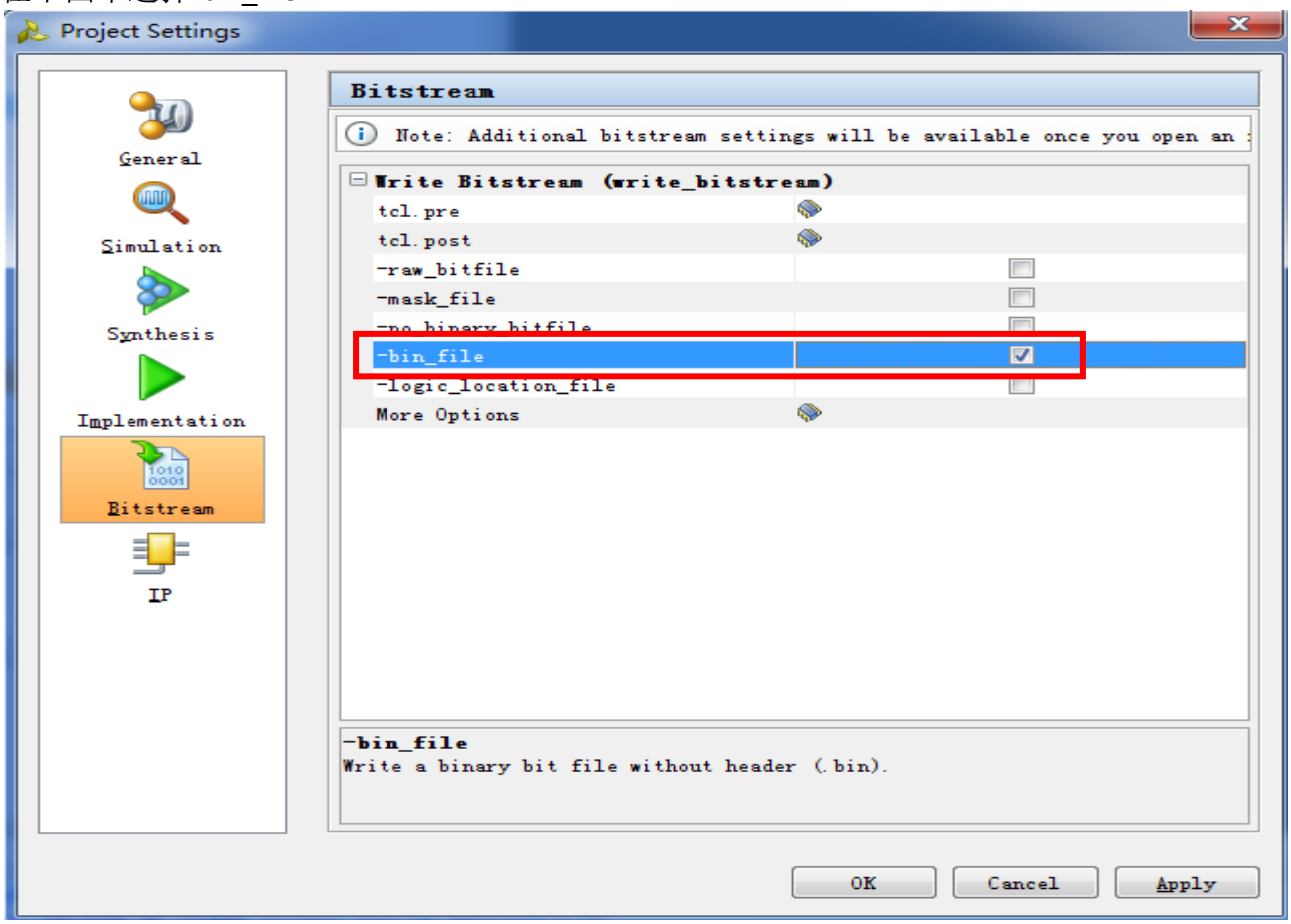
接下来介绍如何将程序烧录到 ROM 里，这样程序就能掉电不丢失。

首先，打开 Bitstream Settings，如下图所示单击：

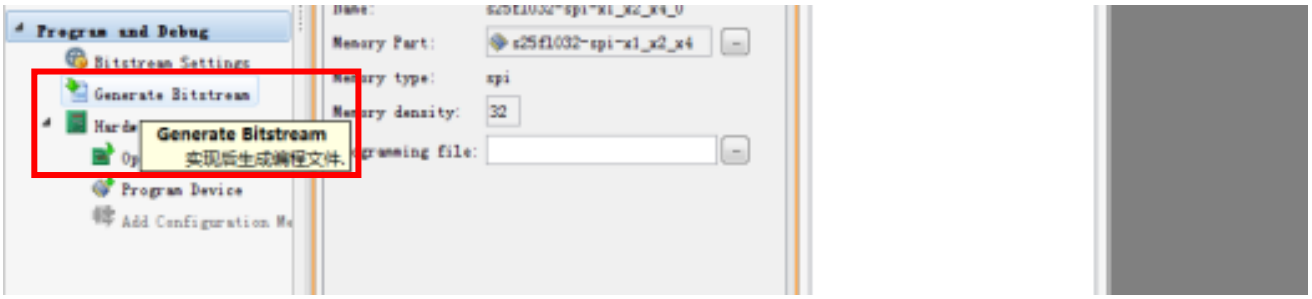


打开码流设定来更改码流文件形式设定

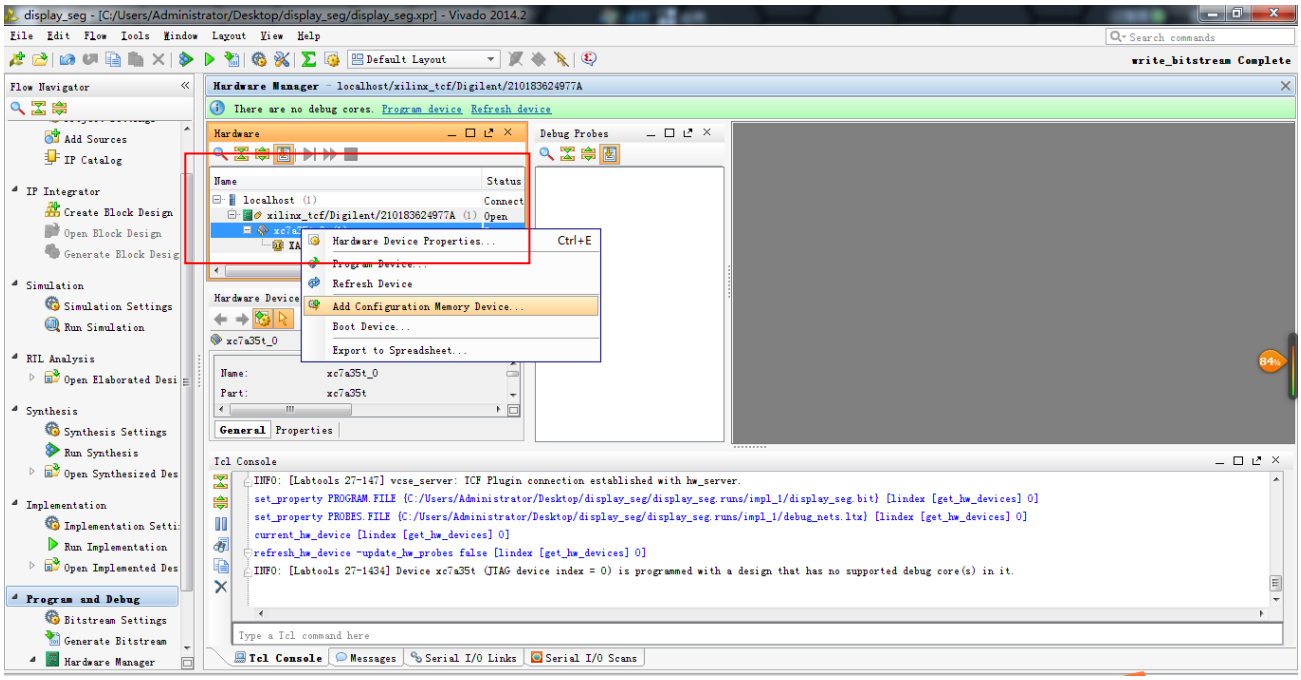
在下图中选择-bin_file



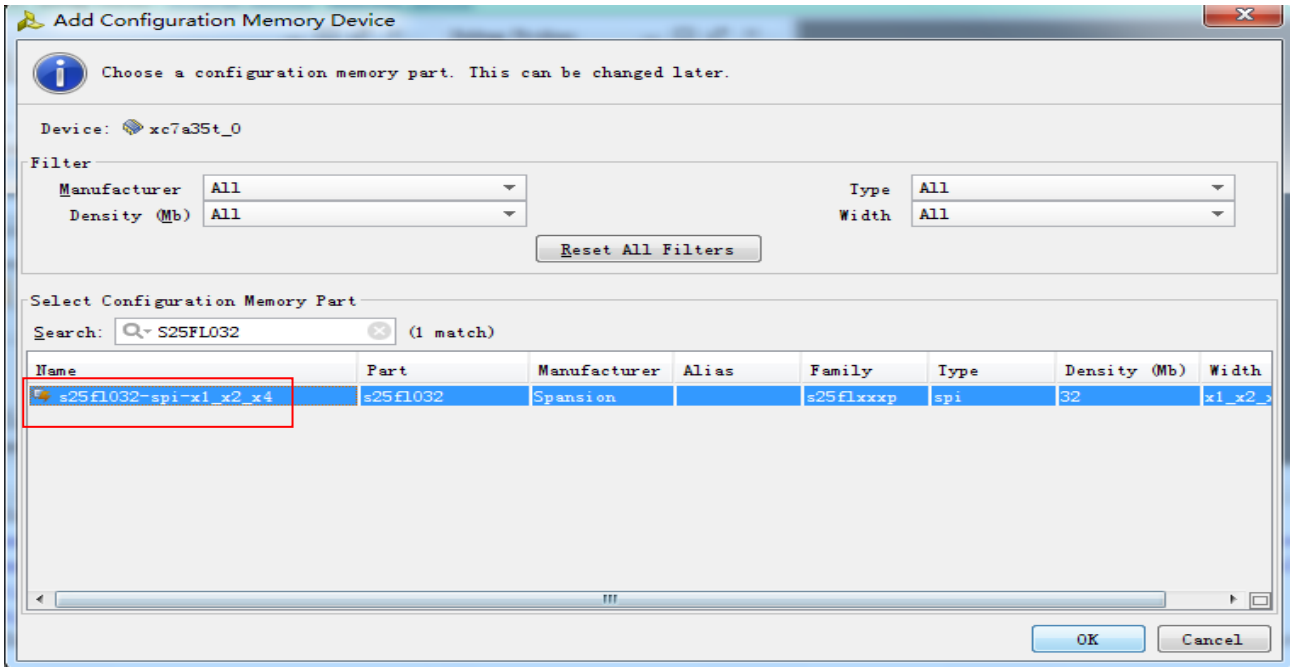
重新生成.bit 文件：

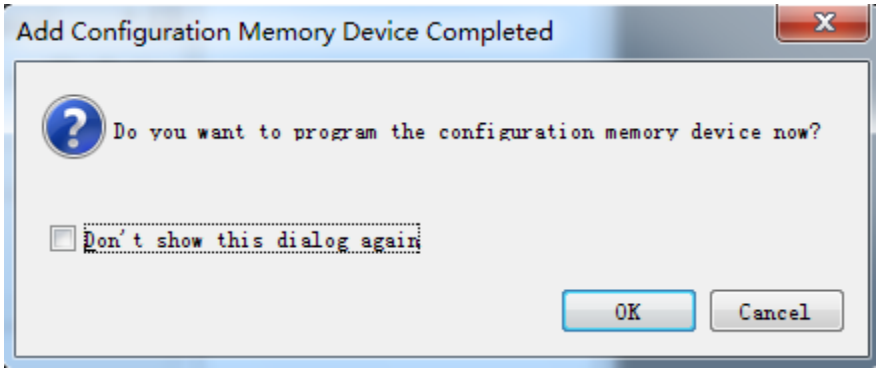


.bit 文件重新生成之后，右击 FPGA 芯片选择 Add Configuration Memory Device



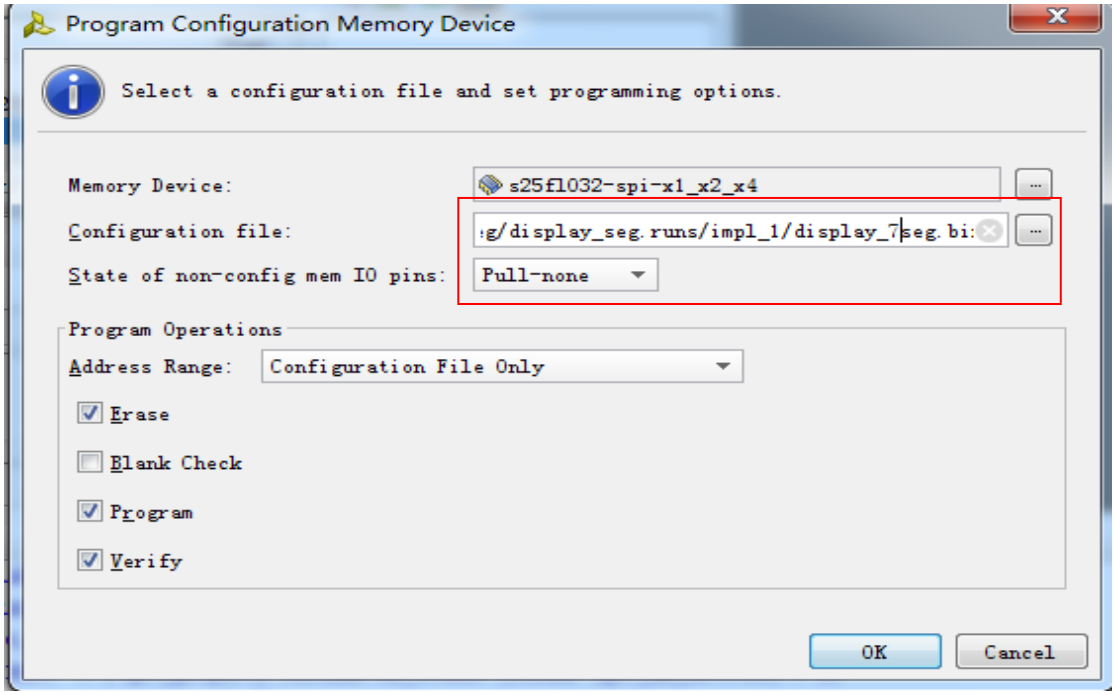
选择 flash 芯片信号，如下图：

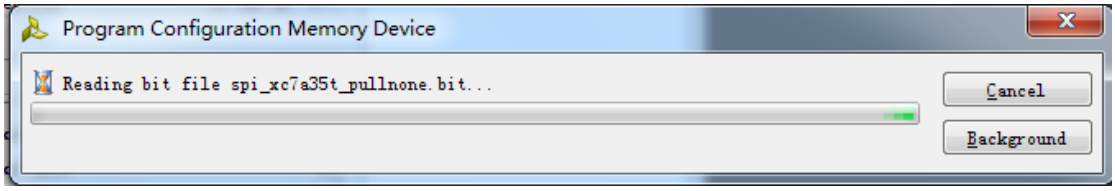


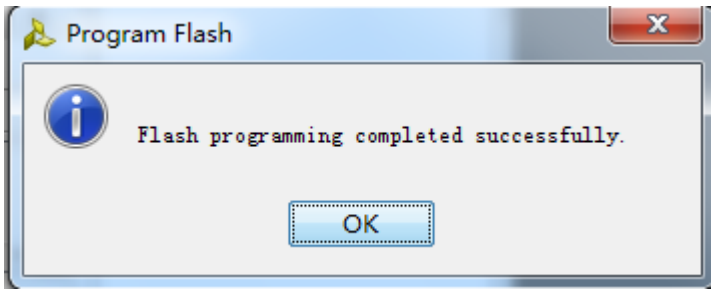


上图，点击 OK

在弹出的窗口选择配置文件为前面生成的.bit 文件

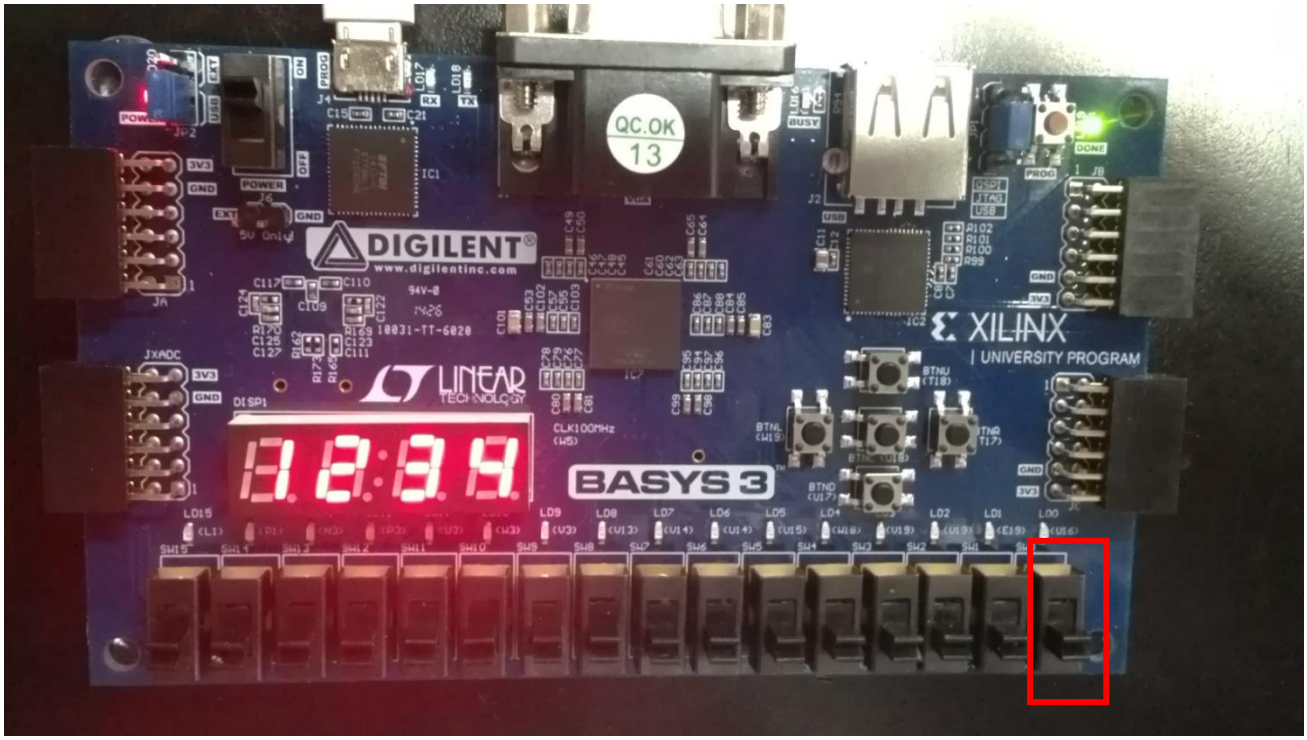


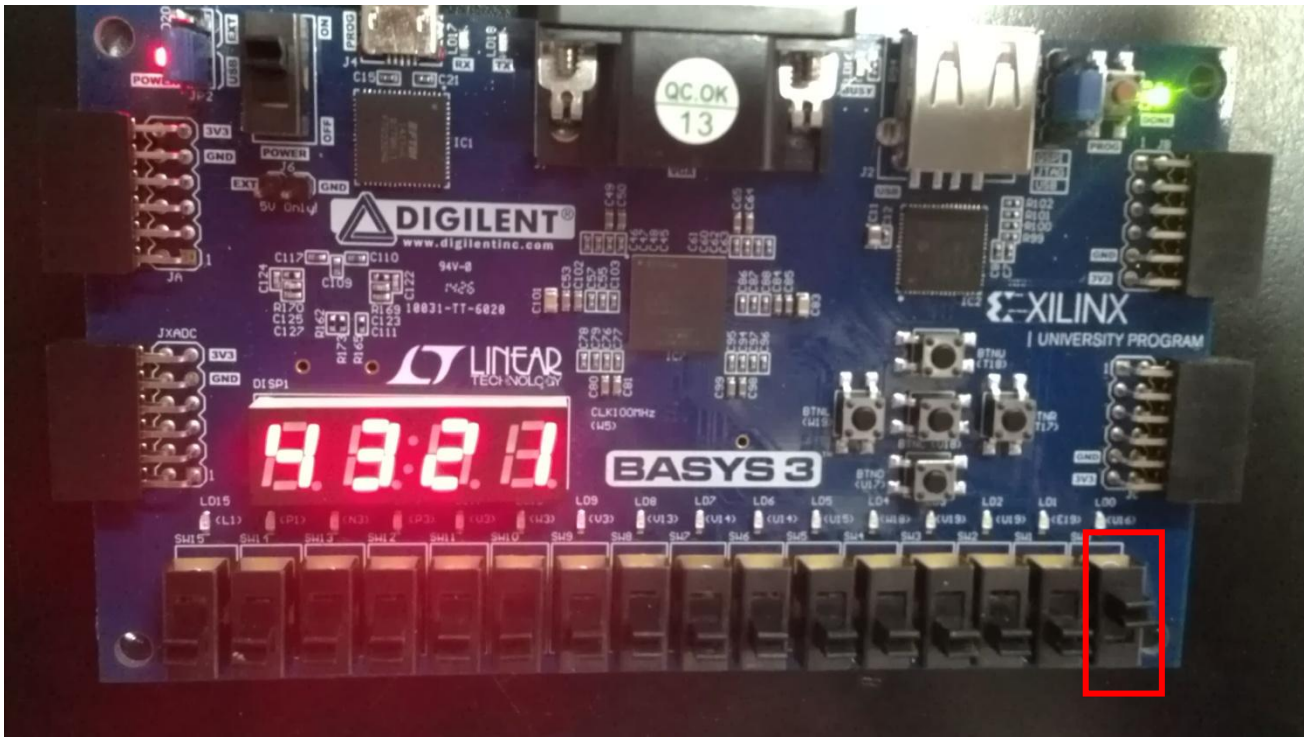




点击 OK，完成 Flash programming

开发板演示：


DIGILENT
www.digilentinc.com
XILINX
UNIVERSITY PROGRAM
LINEAR TECHNOLOGY
BASYS 3
QC.OK
13


DIGILENT
www.digilentinc.com
XILINX
UNIVERSITY PROGRAM
LINEAR TECHNOLOGY
BASYS 3
QC.OK
13

附　录

引脚分配约束文件：

```
set_property PACKAGE_PIN W5 [get_ports CLK]
set_property PACKAGE_PIN V17 [get_ports SW_in]
set_property IOSTANDARD LVCMOS33 [get_ports SW_in]
set_property IOSTANDARD LVCMOS33 [get_ports CLK]
set_property PACKAGE_PIN W4 [get_ports {display_out[10]}]
set_property PACKAGE_PIN V4 [get_ports {display_out[9]}]
set_property PACKAGE_PIN U4 [get_ports {display_out[8]}]
set_property PACKAGE_PIN U2 [get_ports {display_out[7]}]
set_property PACKAGE_PIN W7 [get_ports {display_out[6]}]
set_property PACKAGE_PIN W6 [get_ports {display_out[5]}]
set_property PACKAGE_PIN U8 [get_ports {display_out[4]}]
set_property PACKAGE_PIN V8 [get_ports {display_out[3]}]
set_property PACKAGE_PIN U5 [get_ports {display_out[2]}]
set_property PACKAGE_PIN V5 [get_ports {display_out[1]}]
set_property PACKAGE_PIN U7 [get_ports {display_out[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[9]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[8]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {display_out[10]}]
```

参考源文件代码：

```
module display_7seg(
        input CLK,
        input SW_in,
        output reg[10:0] display_out
        );
         reg [19:0]count=0;
         reg [2:0] sel=0;
         parameter T1MS=50000;
         always@(posedge CLK)
            begin
                if(SW_in==0)
                    begin
                    case(sel)
                    0:display_out<=11'b0111_1001111;
                    1:display_out<=11'b1011_0010010;
                    2:display_out<=11'b1101_0000110;
                    3:display_out<=11'b1110_1001100;
                    default:display_out<=11'b1111_1111111;
                    endcase
                    end
                else
                    begin
                    case(sel)
                    0:display_out<=11'b1110_1001111;
                    1:display_out<=11'b1101_0010010;
                    2:display_out<=11'b1011_0000110;
                    3:display_out<=11'b0111_1001100;
                    default:display_out<=11'b1111_1111111;
                    endcase
                    end
            end
            always@(posedge CLK)
            begin
                count<=count+1;
                if(count==T1MS)
                    begin
                    count<=0;
                    sel<=sel+1;
                    if(sel==4)
                        sel<=0;
                    end
            end
endmodule
```