实验操作手册

安全说明：

1、所有的连接线严禁在设备上电的状态下进行拔出插操作。

2、光纤头或激光器头不要对着人的眼睛，以免发生危险。

3、光纤通信模块的 TE1 至 TE6 为数字信号输出、输入端口，最高电平不能大于 3.3V，最低电平不能小于 0V。

其它说明：

1、FPGA 开发的工程所在的路径必须为全英文的路径，工程所在的文件夹里面新建的文件夹及文件名也必须是全英文的（如里需要被 Quartus 调用），即不能有中文字符。

2、程序代码中的字符必须是全英文的，即不能有中文字符。

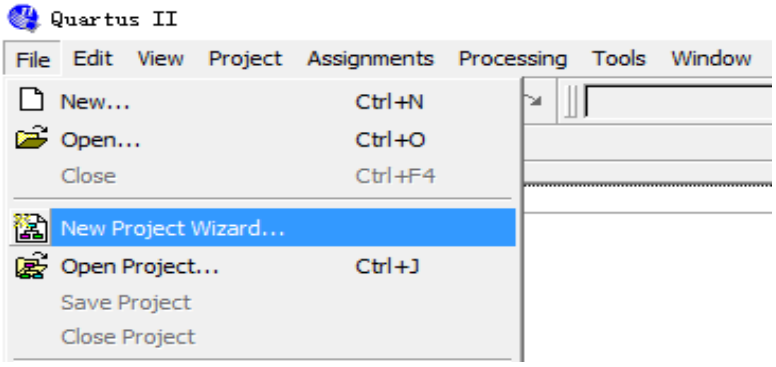
//后面的注释内容可以是中文的，如下图。

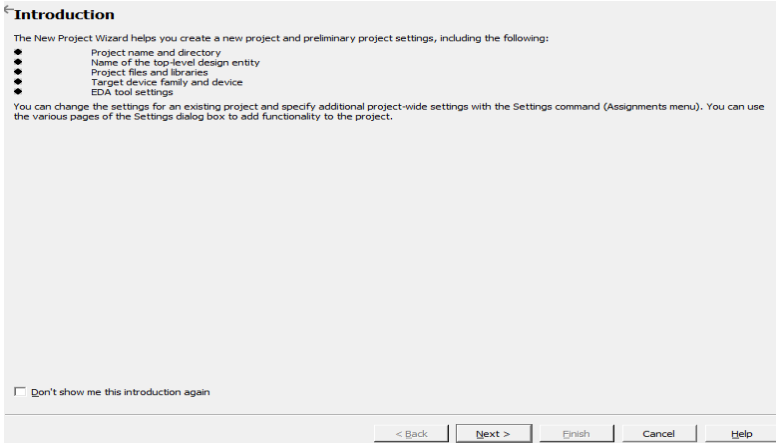
```
always @(posedge clkin_7p68m_pin90_z)        //数据放到CH1和CH2上输出，可用示波器时行观测
begin
```

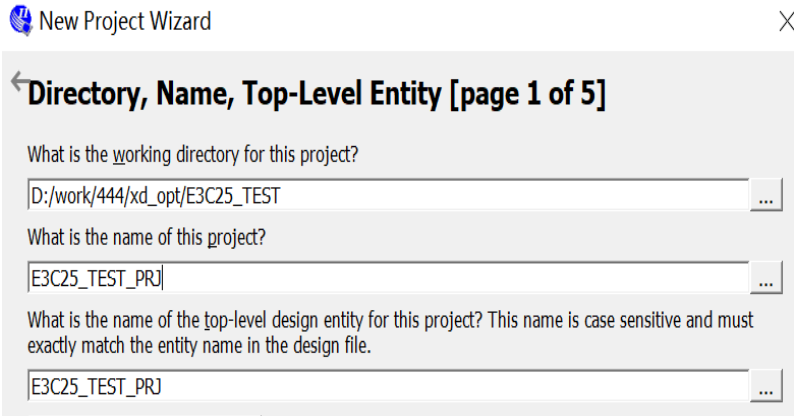
Quartus_11 操作方法

一、新建工程

1、FPGA 的工程建立过程

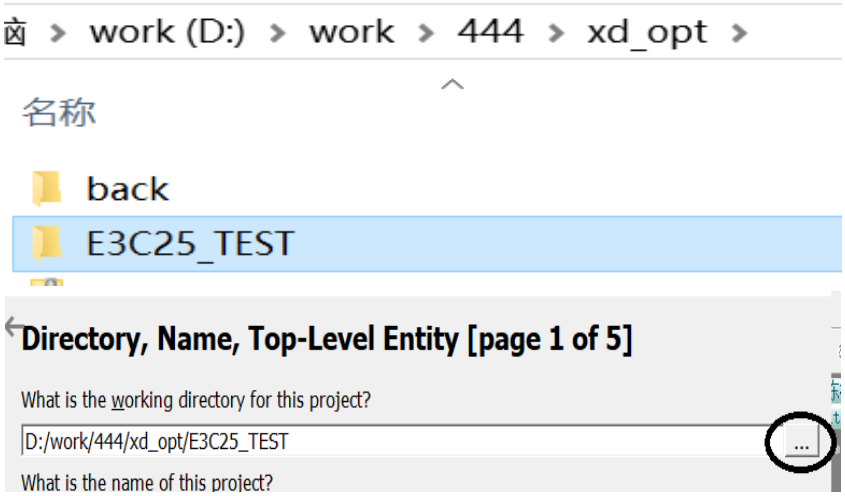






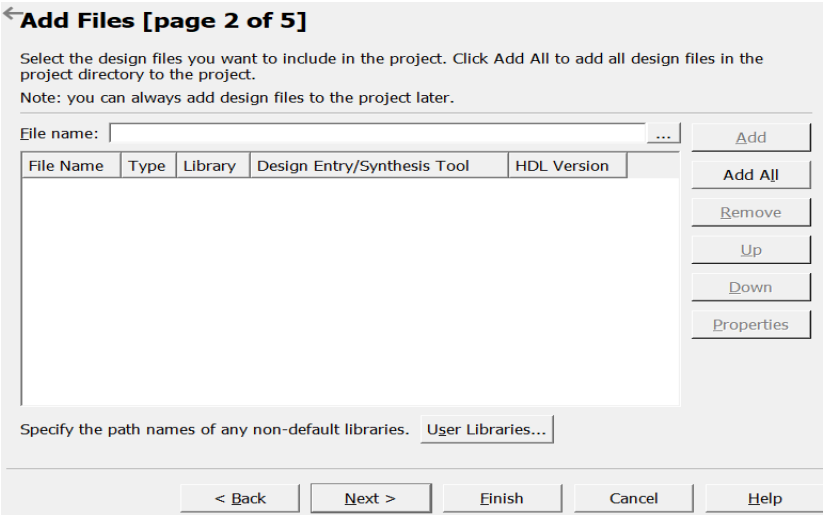
第一行，选译项目用的文件夹（可以在电脑中先建立一个文件夹，如“E3C25_TEST”）。**注：整个文件夹路径不能有中文字符，文件名或工程名也不能有中文字符。**

下图为在电脑中建立的文件夹“E3C25_TEST”



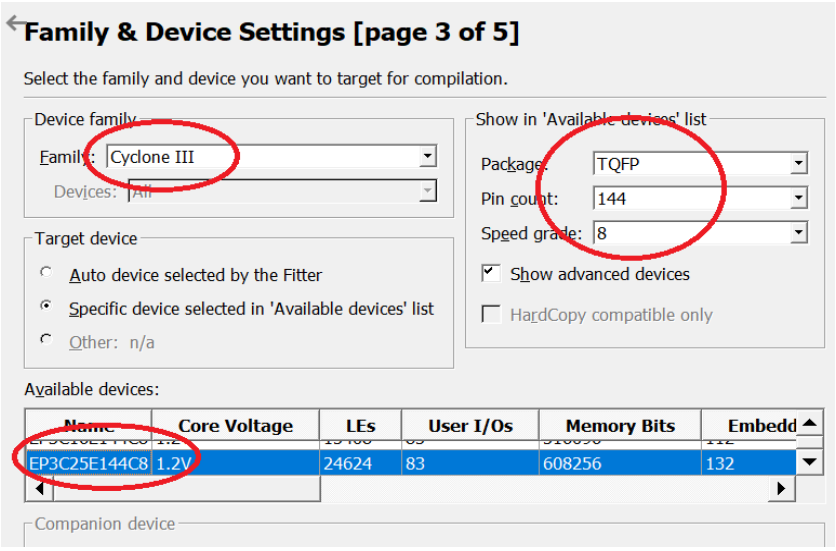
第二行给项目工程叫一个名字（直接在框中输入名字），上图为工程名为“E3C25_TEST_PRJ”

NEXT：

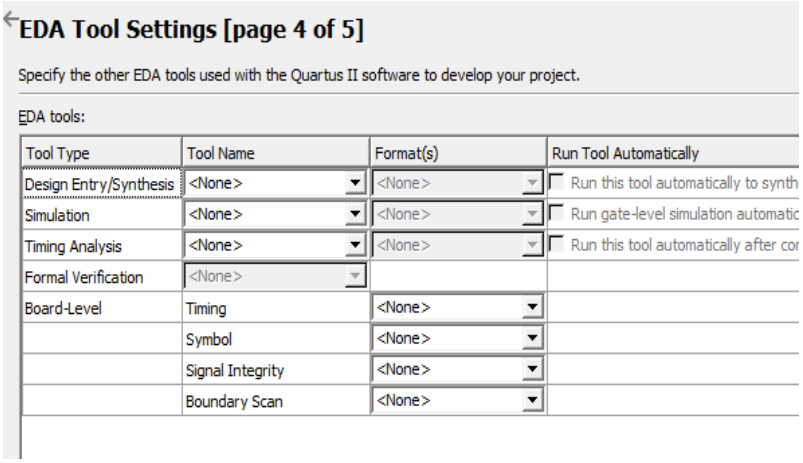


这里不加文件，直接 NEXT。

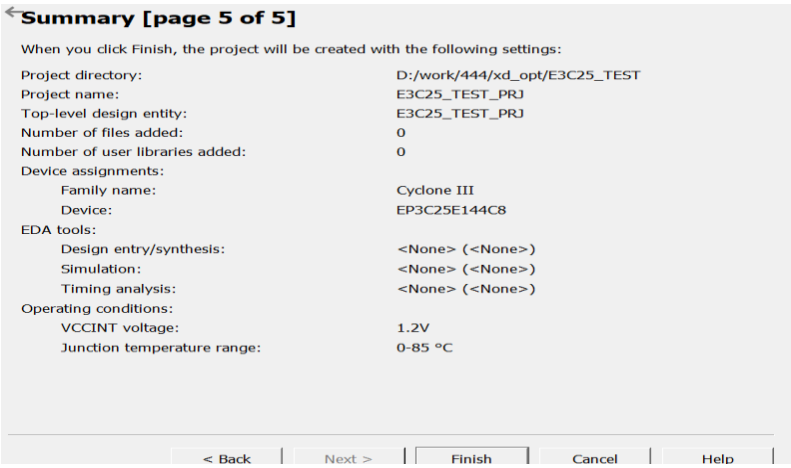
选择器件，NEXT：EP3C25E144C8



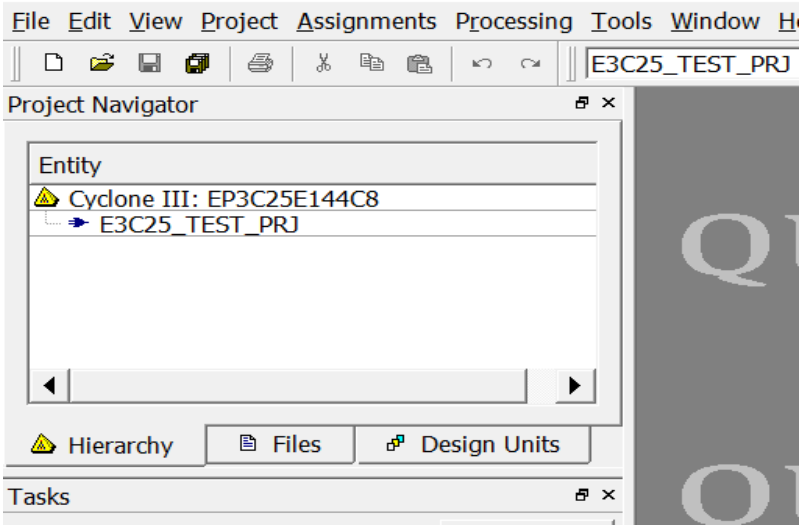
下面，直接 NEXT



NEXT：

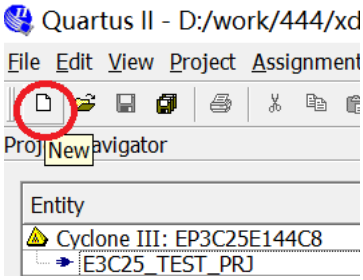


直接 finish。可以看到，如下图：

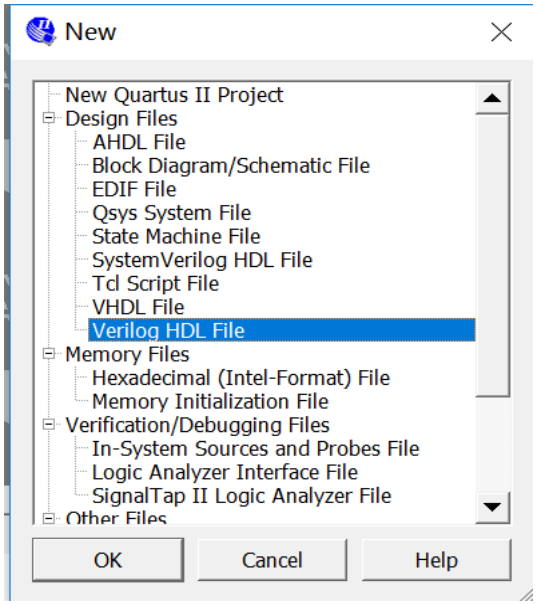


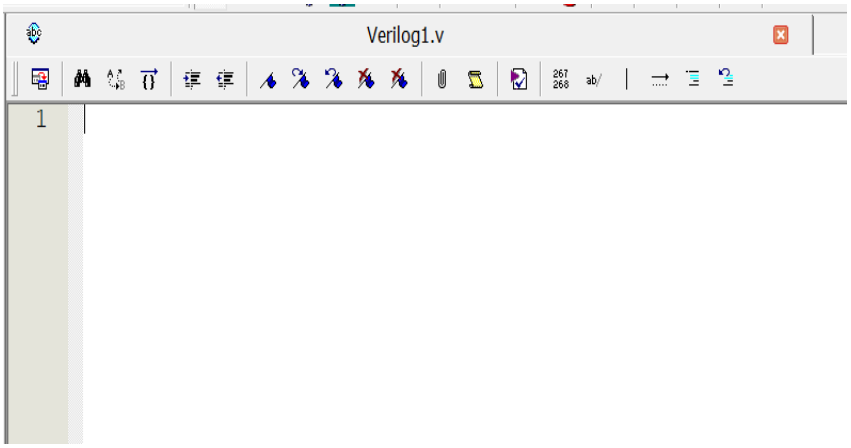
二、设计顶层文件

1、新建一个文件



新建一个文件，有原理图（Block Diagram/Schematic File）或语言（Verilog HDL File）





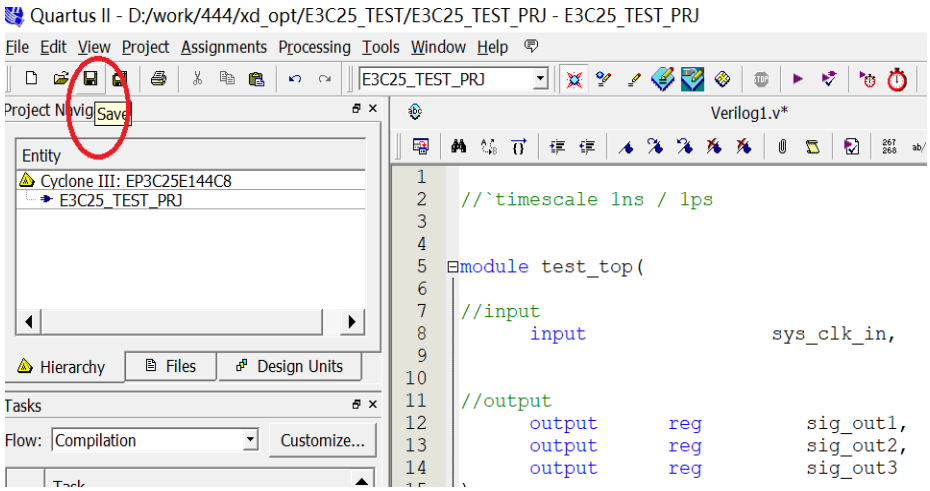
2、编写代码

编写代码，如下：

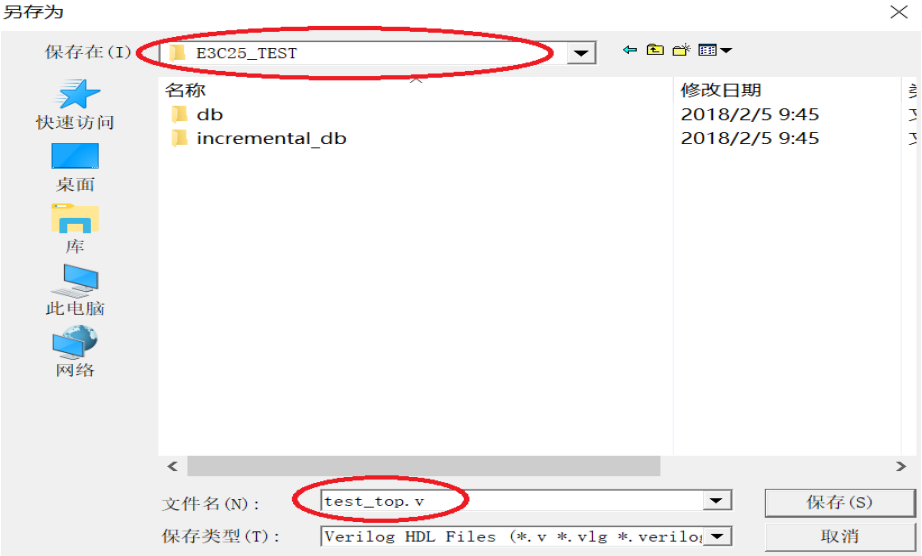
```
//`timescale 1ns / 1ps
module test_top(
//input
    input                   sys_clk_in,
//output
    output              sig_out1,
    output              sig_out2,
    output              sig_out3
);
//将信号输出到脚上
assign   sig_out1            = count[3];
assign   sig_out2            = count[4];
assign   sig_out3            = count[5];
//////////////////////////////
reg    [7:0]   count;
```

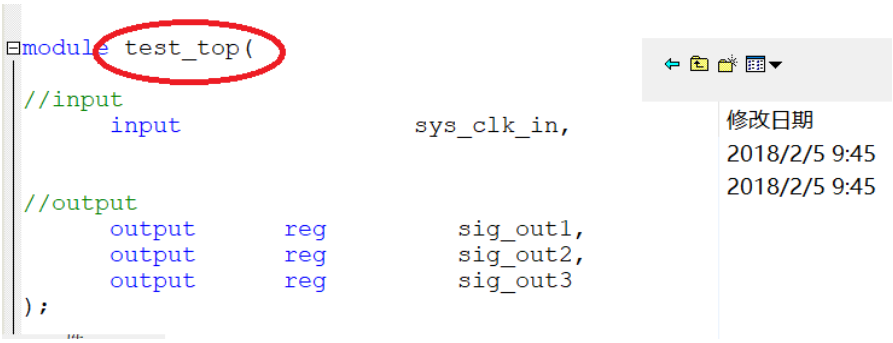
```
always@(posedge sys_clk_in)        begin                    //
      count <= count + 1'b1;
end
endmodule
```

3、保存这个文件



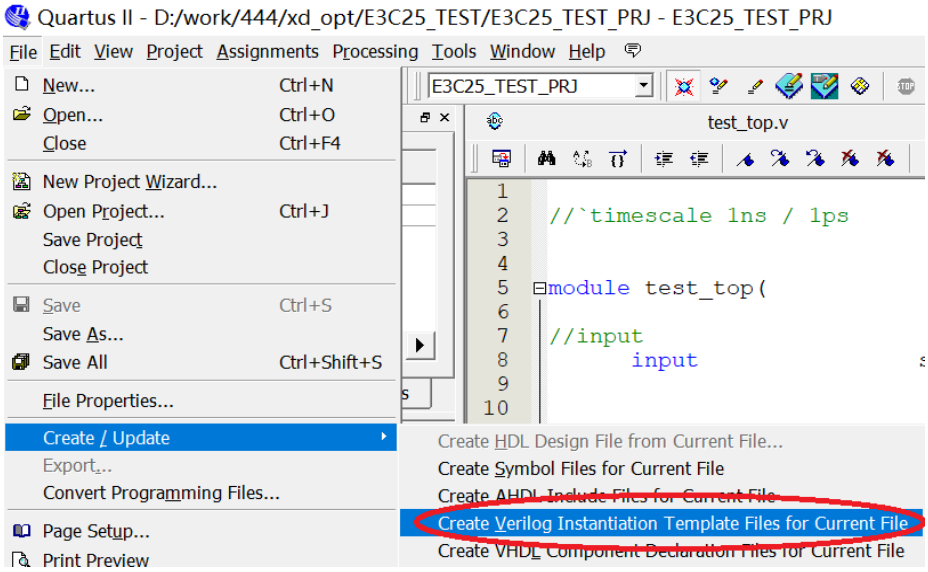
下图中的保存文件的路径一定是你工程所在的路径，文件名与上图中文件中的 **module** 后面的“**test_top**”一致。





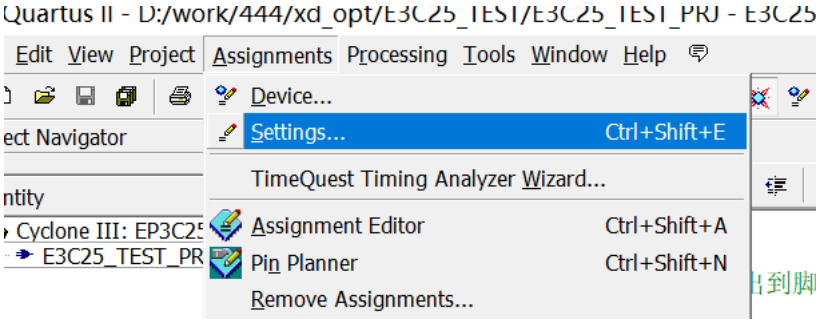
4、编译代码

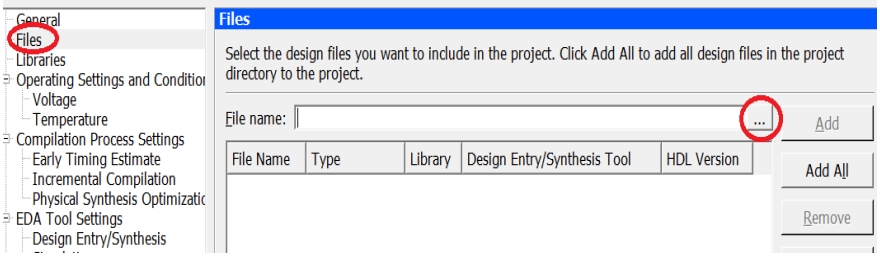
编译一下当前代码有没有语法错误，如果有错，则需要根据提示进行修改。

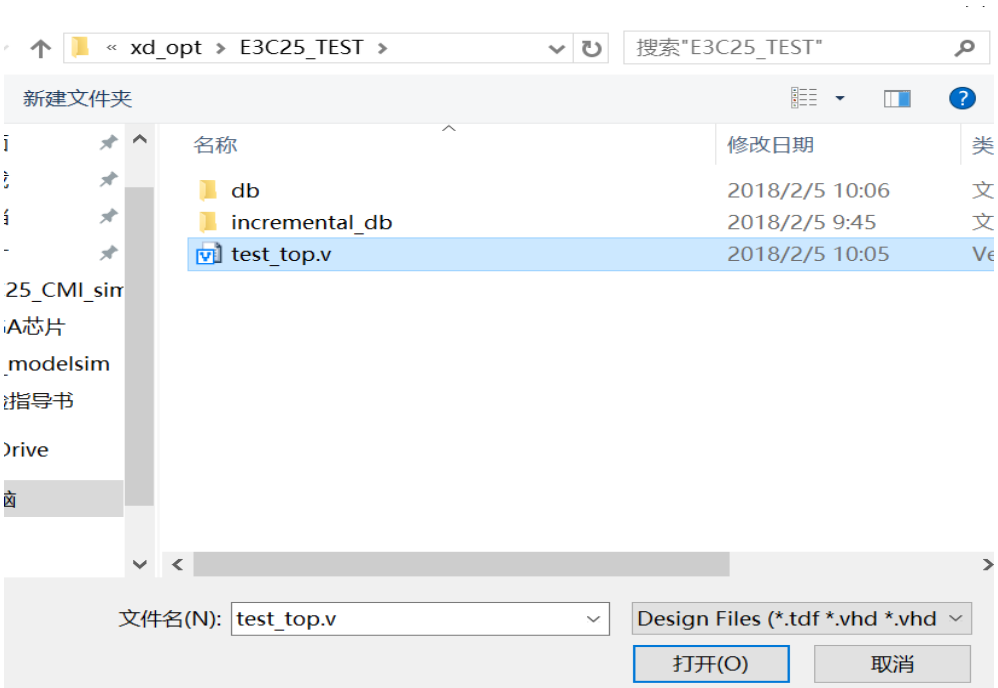


5、将文件加入到本工程中

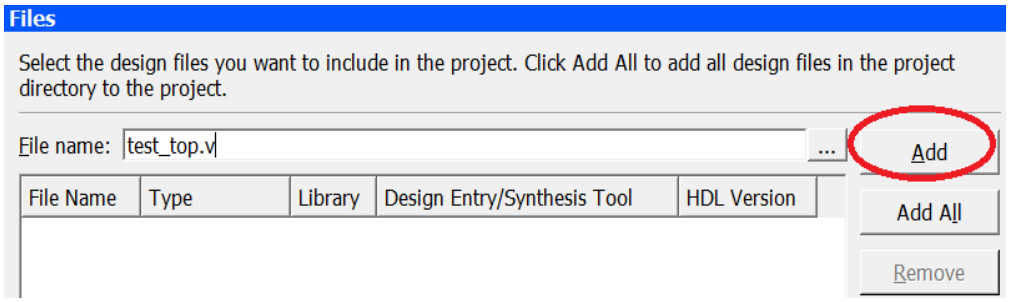
加该文件到工程中：

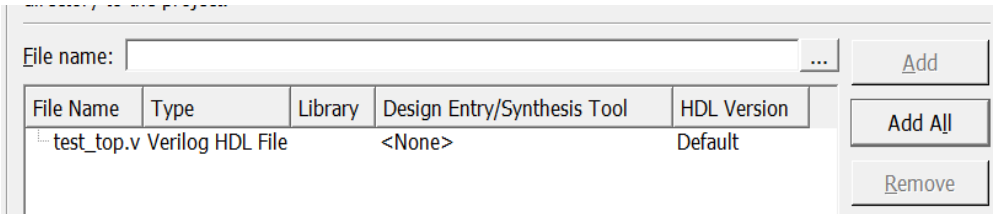




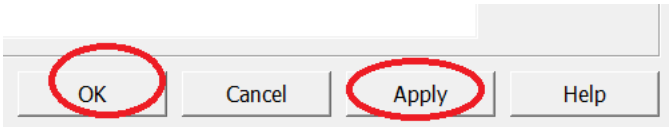


如下图，点“Add”



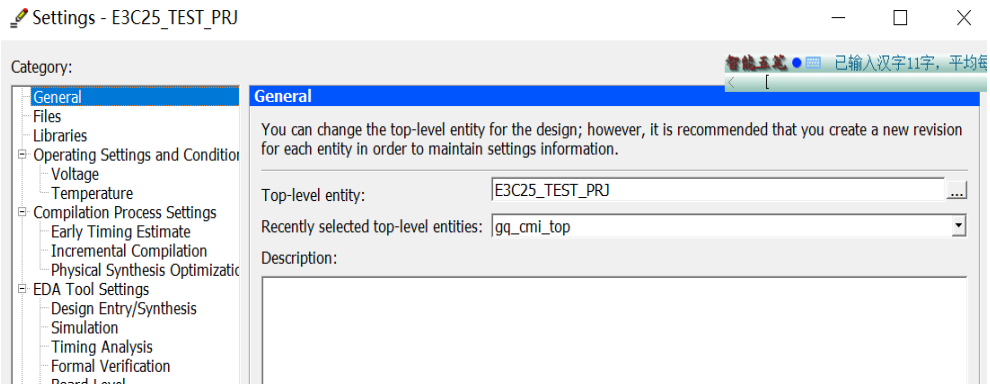


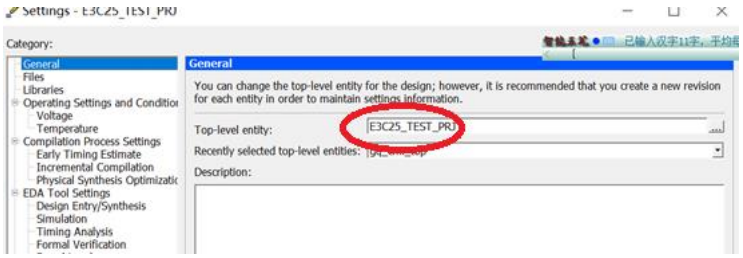
再点“Apply”后，最后点“OK”，如下图：



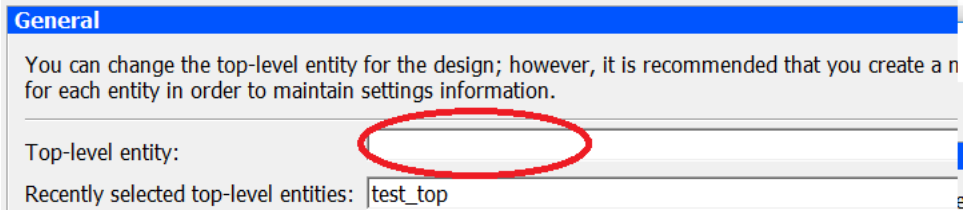
6、将该文件设置为本工程顶层文件

将该文件设置成顶层文件

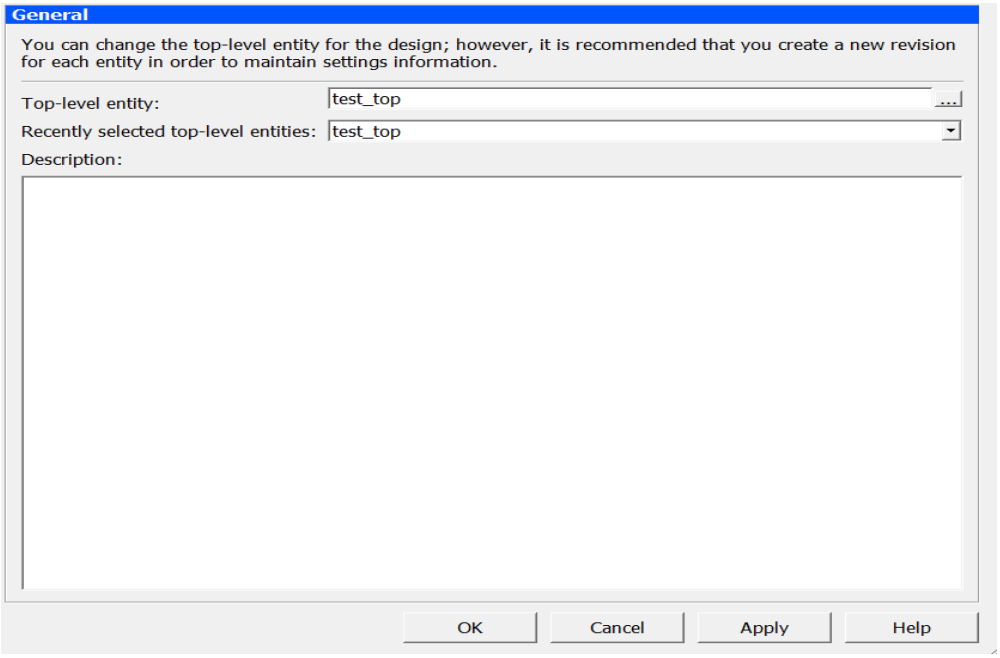




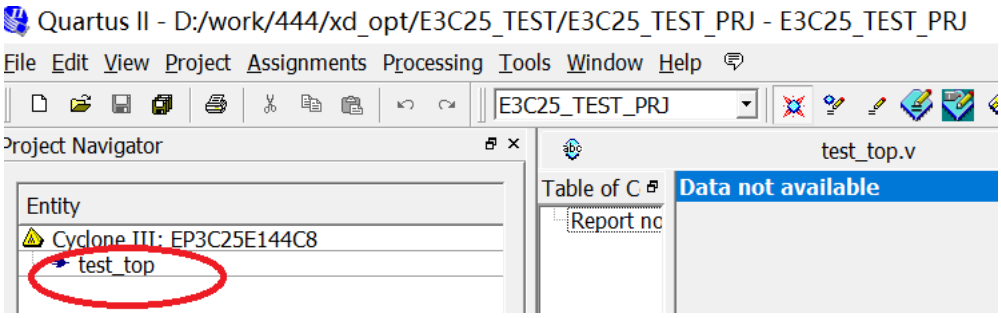
将上图中 Top – level entity:　后面的“＊＊＊”文字删除。



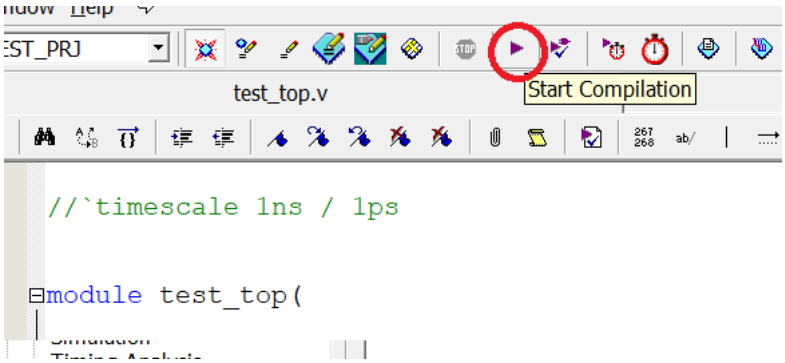
直接 Top – level entity:　后输入新的顶层文件名“test_top”。



再点“Apply”后，最后点“OK”。



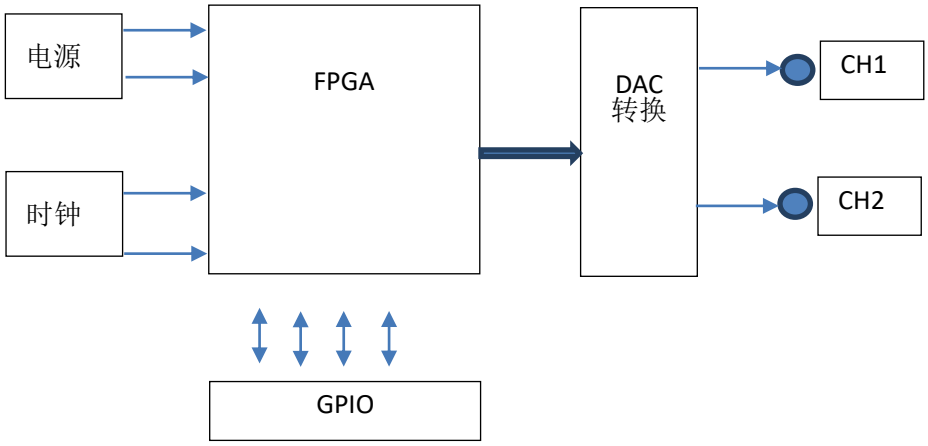
7、编译工程



如果有错误，根据提示进行修改。

三、管脚定义

FPGA 的外部结构：

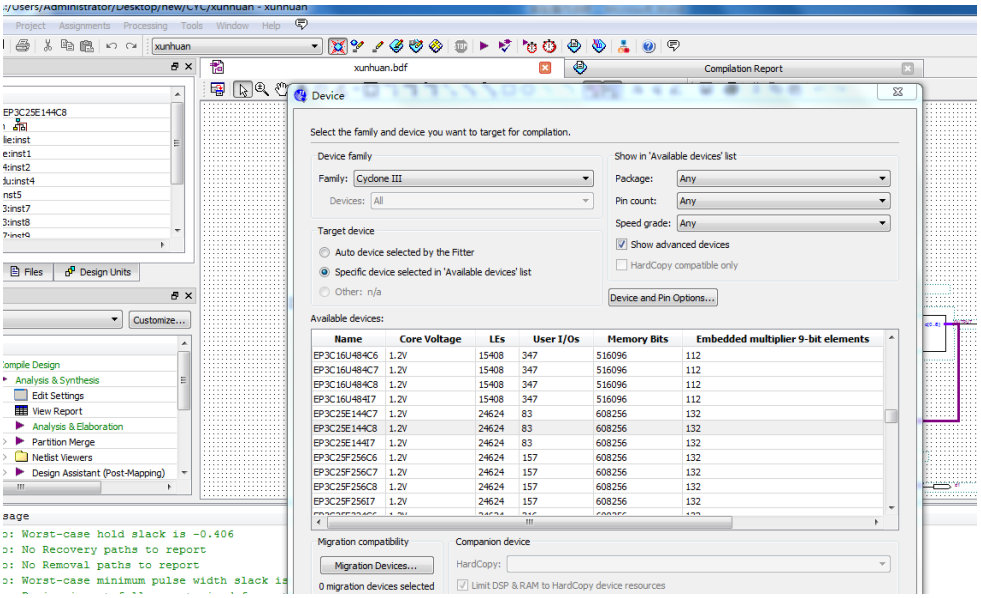


注意：GPIO 上的信号是 3.3V 电平的，所以如果用其它模块产生的信号输入到 FPGA 的 GPIO 脚上，请确认输入信号的电平是否相配置（高电平不能高于 3.3V）

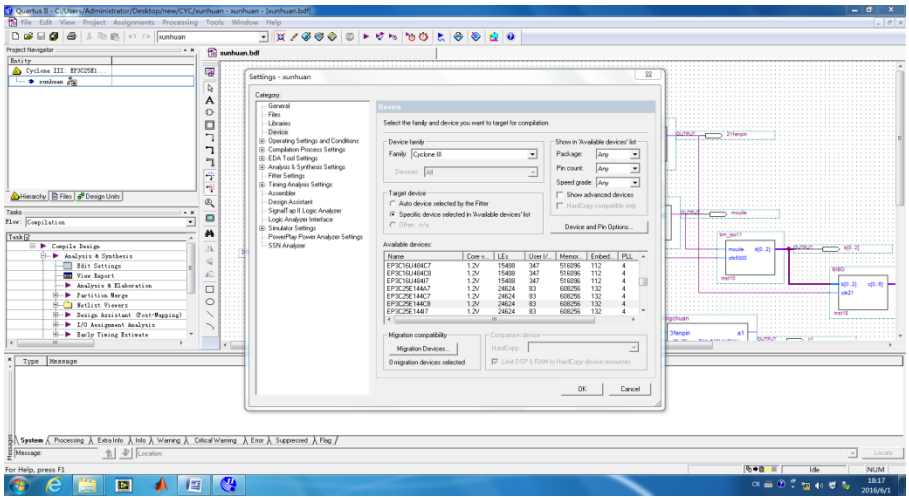
		缺口标记		
GND	保留	PIN_100	PIN_98	保留
5V	保留	PIN_103	PIN_99	保留

缺口管脚为 98，99，100，103。这 4 个管脚是板子设计时留下的专门为用户开发使用的管脚。所以在下载程序到硬件时，我们可以在这 4 个管脚中任选使用。如果在前面设置的时候未选定硬件，可以重新进行选定，进而重新设置引脚。选择硬件：菜单 Assignments 下选择 Devise。

11 版本: Cyclone III – EP3C25E144C8



9 版本：Cyclone III – EP3C25E144C8

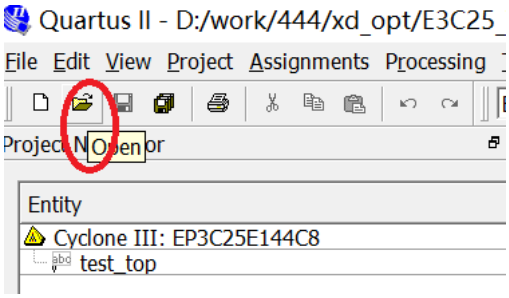


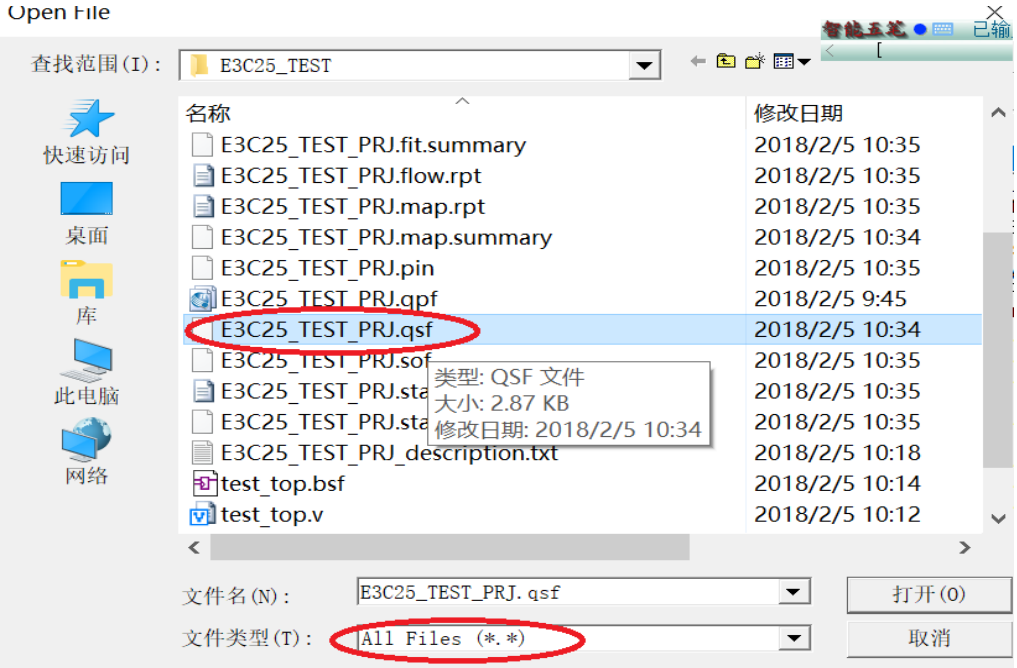
二次开发管脚：

选择时：点击 I/O Bank，然后再左边的下拉列表中选择可选的管脚。

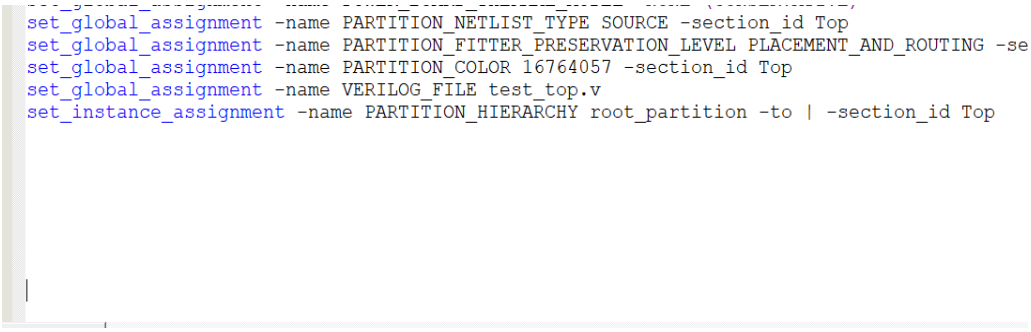
1、方法 A

直接修改“＊.qsf”文件，打开





在该文件最后进行增加信号与管脚的对应关系：



```
set_global_assignment -name PARTITION_NETLIST_TYPE SOURCE -section_id Top
set_global_assignment -name PARTITION_FITTER_PRESERVATION_LEVEL PLACEMENT_AND_ROUTING -se
set_global_assignment -name PARTITION_COLOR 16764057 -section_id Top
set_global_assignment -name VERILOG_FILE test_top.v
set_instance_assignment -name PARTITION_HIERARCHY root_partition -to | -section_id Top
```

在上图的下面空白处，增加以下代码：

set_location_assignment PIN_98 -to sig_out1

set_location_assignment PIN_99 -to sig_out2

set_location_assignment PIN_100 -to sig_out3

set_location_assignment PIN_89 -to sys_clk_in

```
set_global_assignment -name PARTITION_NETLIST_TYPE SOURCE -section_id
set_global_assignment -name PARTITION_FITTER_PRESERVATION_LEVEL PLACEM
set_global_assignment -name PARTITION_COLOR 16764057 -section_id Top
set_global_assignment -name VERILOG_FILE test_top.v



set_location_assignment PIN_98 -to sig_out1
set_location_assignment PIN_99 -to sig_out2
set_location_assignment PIN_100 -to sig_out3
#set_location_assignment PIN_103 -to sys_clk_in

set_location_assignment PIN_89 -to sys_clk_in
```

（注：＃表示注释掉，这行无效）

sig_out1、sig_out2、sig_out3、sys_clk_in 这是信号的名称，名字随便取，要与“test_top”程序中定义的输入、输出信号的名字一样即可（如下图是 test_top 中的输入和输出）。

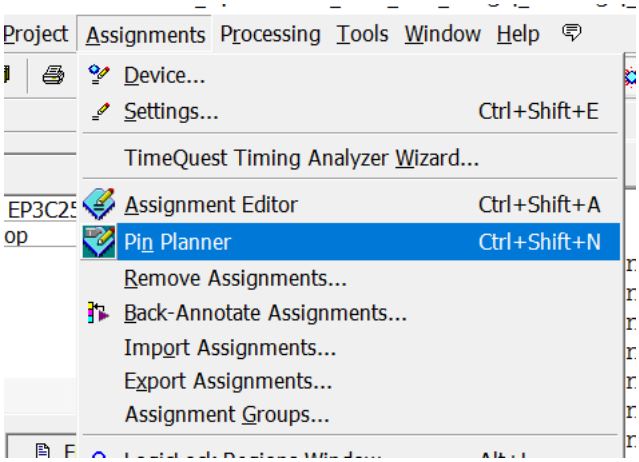
PIN_90、PIN_98……这些才是对应 FPGA 的管脚。

```
4
5  module test_top(
6
7   //input
8         input                sys_clk_in,        //pin103_te6
9
10
11  //output
12        output               sig_out1,          //pin98_te3
13        output               sig_out2,          //pin99_te4
14        output               sig_out3           //pin100_te5
15  );
16
17
```

修改完成后，保存 E3C25_TEST_PRJ.qsf 文件。

2、方法 B

管脚定义也可以通过” Assignments-Pin Planner”进行配置：下图为（11 版本）图。

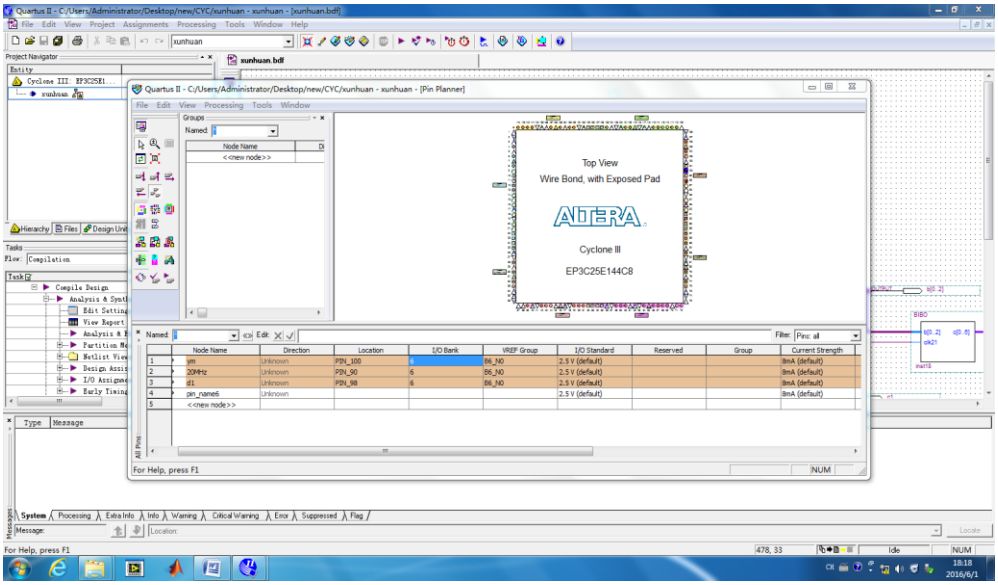


Node Name 是信号名称，Location 对应的管脚

Named: * Edit:

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	rrent S
sig_out1	Output	PIN_98	6	B6_N0	3.3-V LVTTL		8mA (.
sig_out2	Output	PIN_99	6	B6_N0	3.3-V LVTTL		8mA (.
sig_out3	Output	PIN_100	6	B6_N0	3.3-V LVTTL		8mA (.
sys_clk_in	Input	PIN_89	5	B5_N0	3.3-V LVTTL		8mA (.
<<new node>>							

9 版本: Assignments 菜单-Pins



注意: 在这里我们只对我们要求的管脚进行设置;

(1)系统时钟 20Mhz—90 管脚;

(2)编码译码管脚在 98，99，100，103 这几个管脚中任选，但要记住自己所选的管脚，下载到硬件中测试使用。

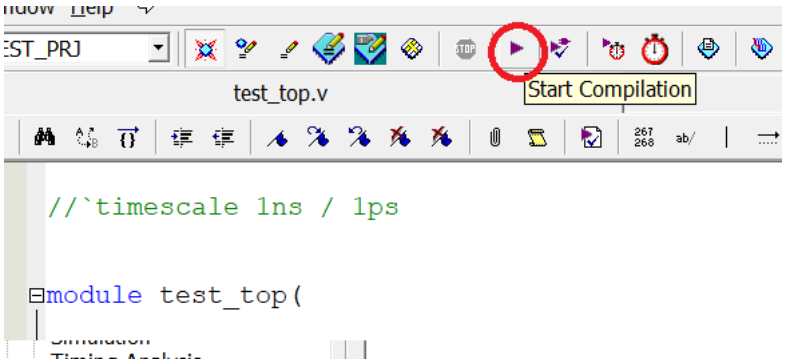
选择时：点击 I/O Bank，然后再左边的下拉列表中选择可选的管脚。

以上 2 种方式修改管脚定义后，都会在另一种方式进行了更新，就是用任何一种方法定义或修改就可以了。

四、整个工程编译

再次编译一次工程。

注意：选择设定管脚以后要再次进行编译，生存新的 sof 文件。否则设置不成功。



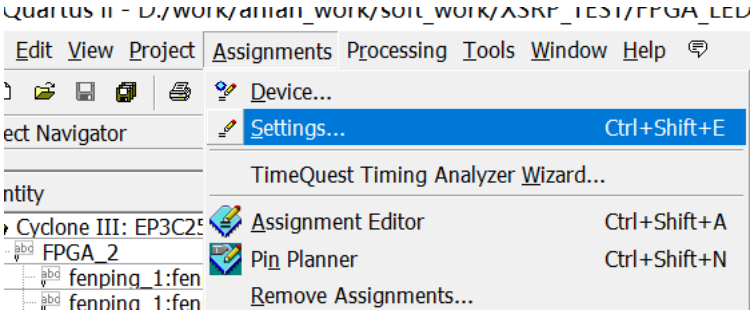
如果有错就根据提供修改。

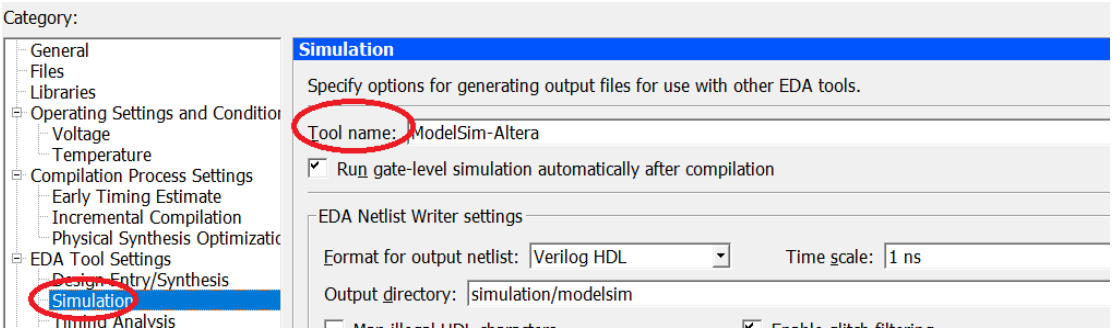
编译成功后，系统会在工程所在文件夹中生成 E3C25_TEST_PRJ.sof 供下载器下载到 FPGA 的硬件中。

五、modelsim 仿真

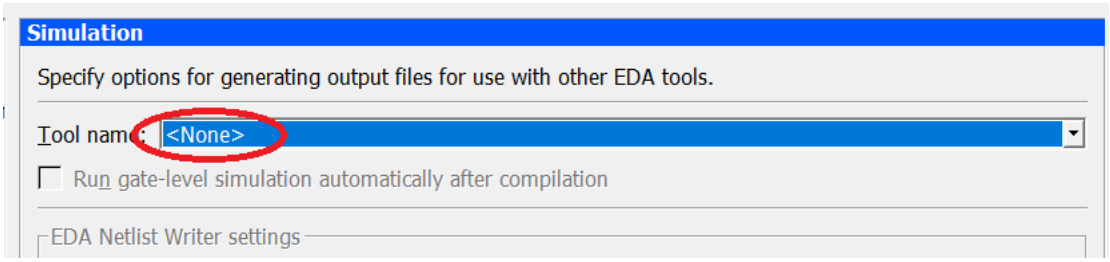
第五节可以不做，如果不做可直接到第六节。

1、配置 modelsim

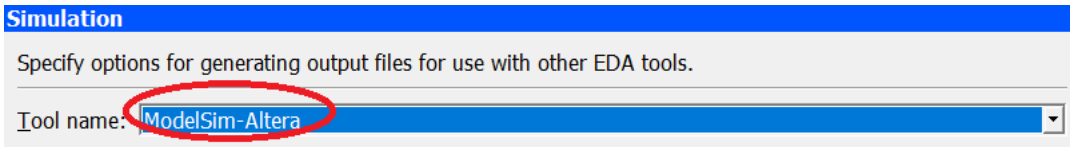




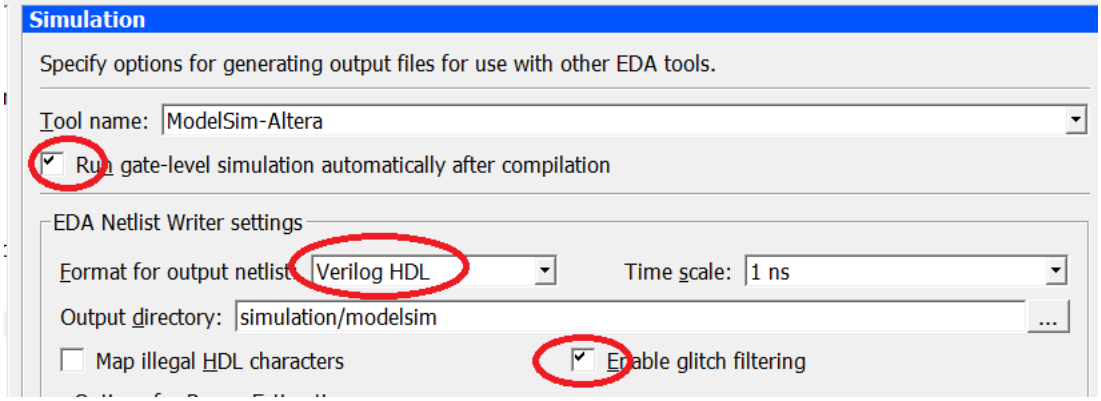
不开仿真功能：



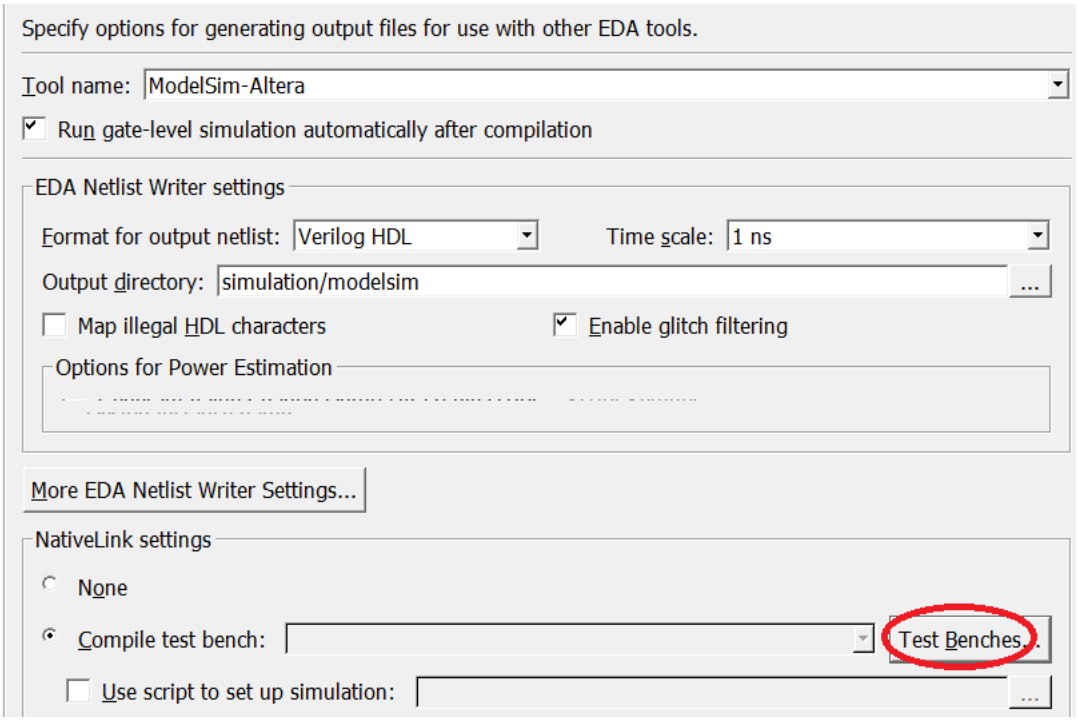
开仿真功能：

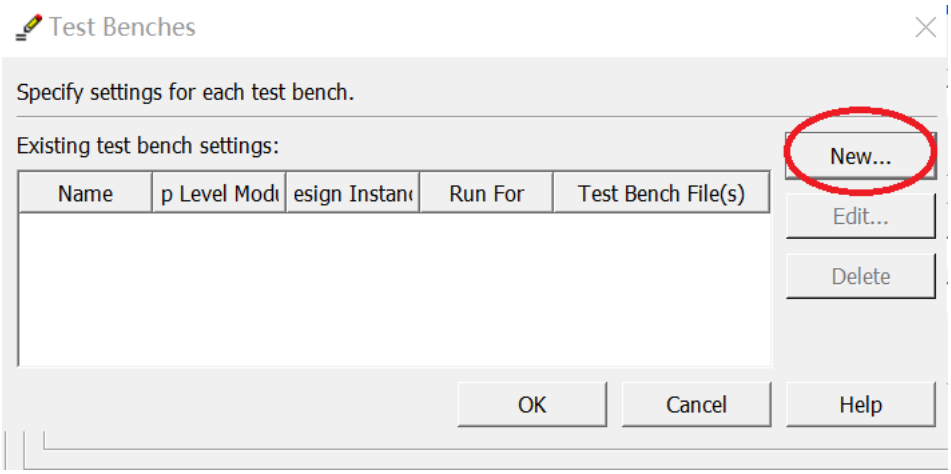


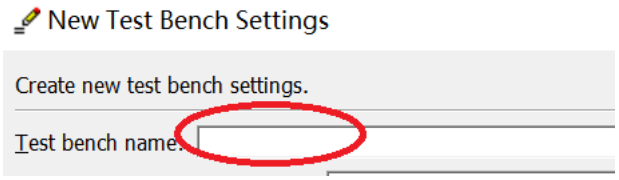
以下是开仿真的配置：



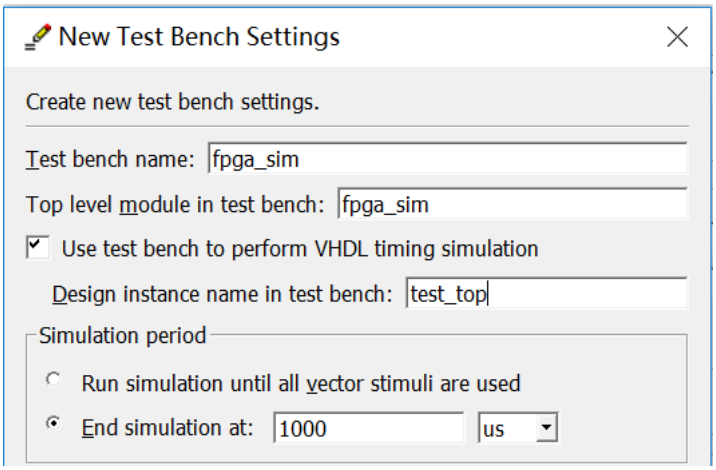
2、关联 test bench files







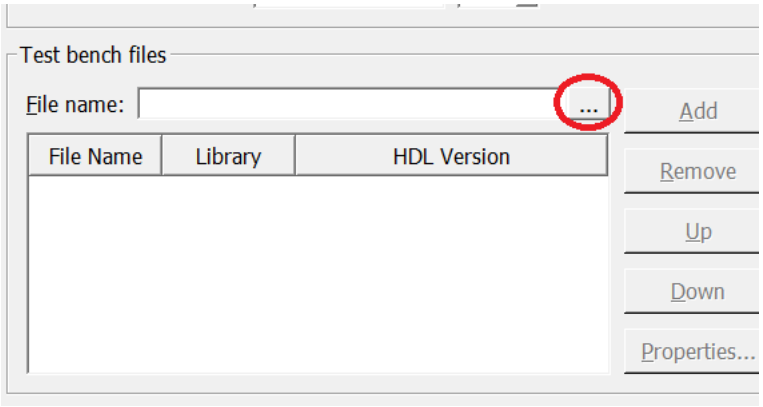
如：posedge_detection_tb、或 fpga_sim 等任意名字。也就是后面要加的 test bench files 中的文件名。



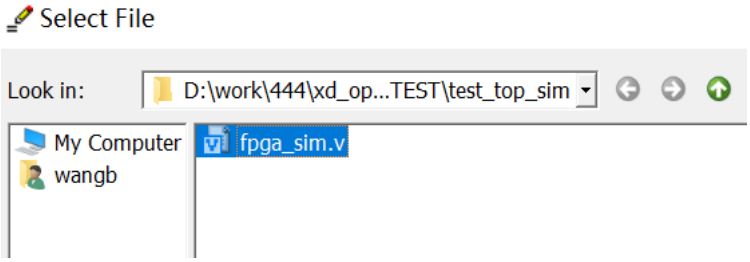
“test_top”是要仿真的模块名称，这例子中是工程顶层文件（模块）。

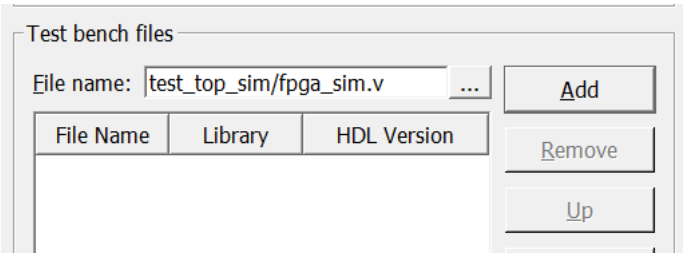
“1000”us 是仿真时间长度。

在 test bench files 中，加入仿真文件：

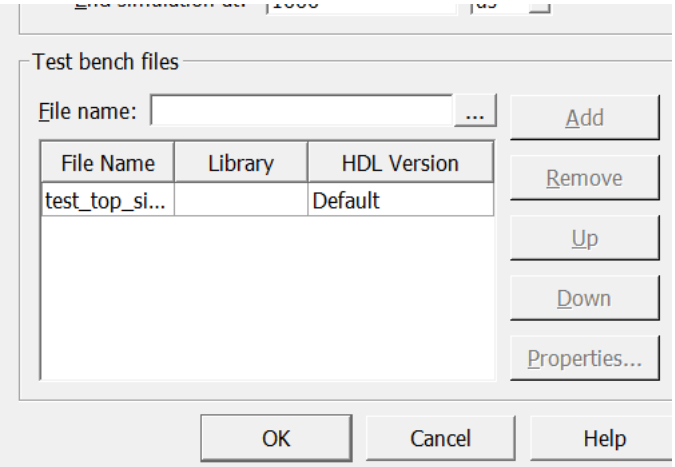


本例中是将 fpga_sim.v 文件下在工程所在的文件夹下，又增加了一层文件夹“test_top_sim”,将 fpga_sim.v 文件放在其中，新建文件夹的目的是将文件分类，这样就比较好找文件，用户也可以将 fpga_sim.v 直接放在工程所在的文件夹下大，这都是可以的。

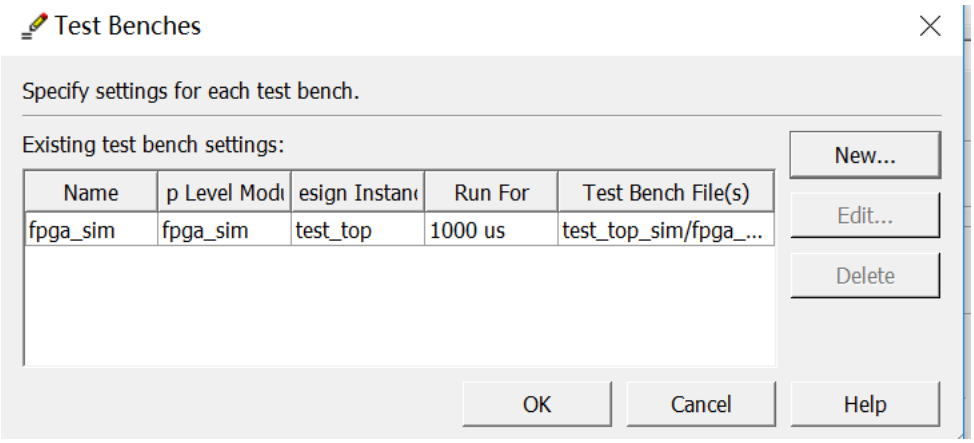




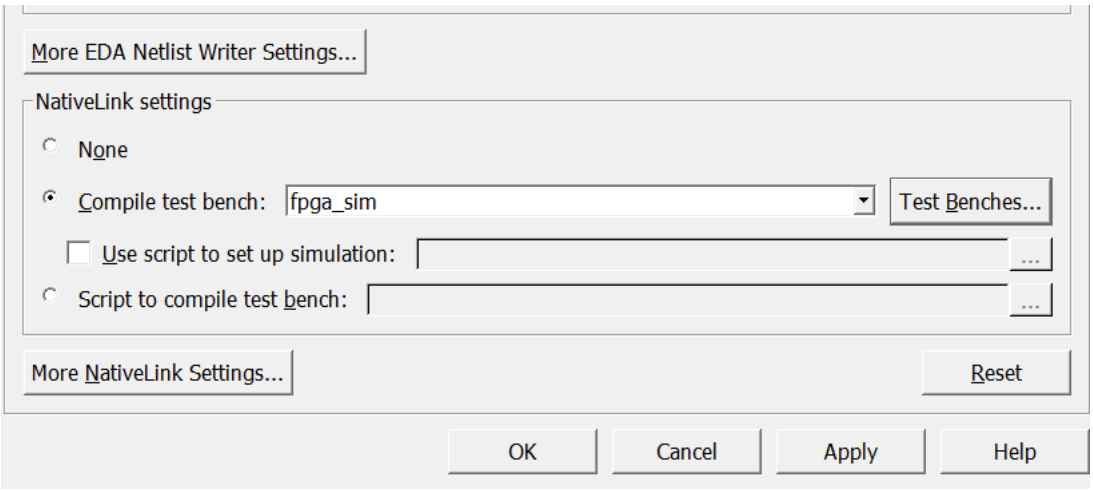
点“Add”



点“OK”



点“OK”



再点“Apply”后，最后点“OK”。

3、设计 test bench files

编写 fpga_sim.V 代码（这一步可以放在前面做，先生成 fpga_sim.v）

从本项目可以看出，这个 test_top 的模块，有 1 个输入，3 个输出。如下图：

1 个输入为工作时钟信号。

3 个输出分别是不同分频信号。

```
module test_top(

//input
        input                   sys_clk_in,         //pin103_te6


//output
        output                  sig_out1,           //pin98_te3
        output                  sig_out2,           //pin99_te4
        output                  sig_out3            //pin100_te5
);
```

设计 fpga_sim.v 文件

先打开这个文件，可以看出里面的结构，主要是生成 sys_clk 和 reset_n 这 2 个信号(其中 reset_n 未使用，可以去掉)。

```
`timescale 10ns/10ps
module fpga_sim;
    reg                     sys_clk;
    reg                     reset_n;    //未使用
    wire                    test_s1;
    wire                    test_s2;
    wire                    test_s3;
test_top    test_top (
    //input
    .sys_clk_in             (sys_clk),
    //output
    .sig_out1               (test_s1),
    .sig_out2               (test_s2),
    .sig_out3               (test_s3)

);
parameter clkper = 100;
initial begin
    sys_clk = 1'b0;
    #0;
    reset_n   =   1'b1;
    #100;
    reset_n   =   1'b0;

    #150;
    reset_n   =   1'b1;
```

end

编定完代码后保存文件。

注：关于仿真时钟的频率

以下中代码中的重要参数

```
//////////////////////////////////////////////////////////
`timescale 10ns/10ps
parameter clkper = 100;
always   begin
    #(clkper / 2) sys_clk = ~sys_clk;
End
//////////////////////////////////////////////////////////
```

`timescale 10ns/10ps

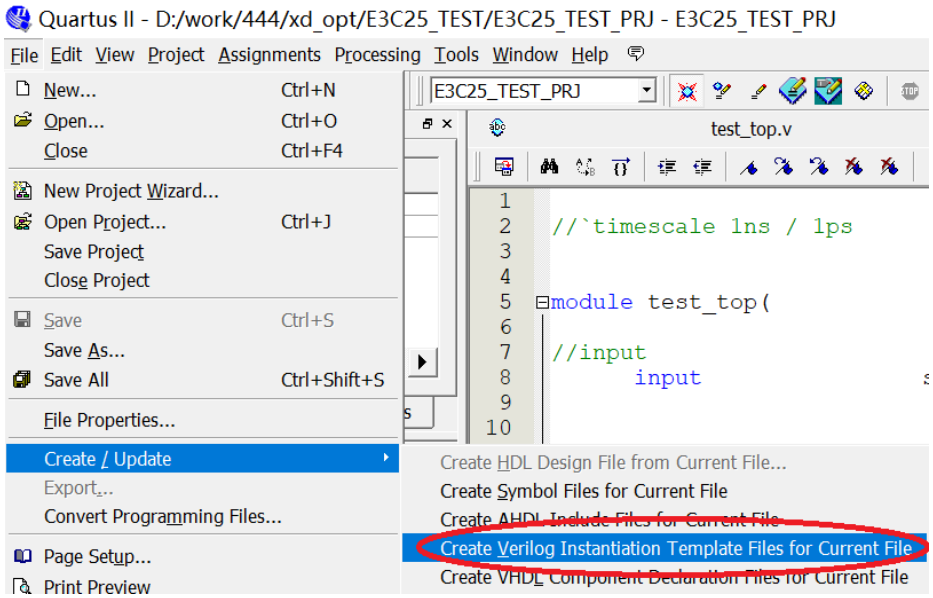
表示时延单位为 10ns, 时延精度为 10ps。

Clkper =100，表示 sys_clk 信号周期为 100＊10ns = 1000ns，即速率为 1MHz 的速率。

clkper / 2　表示每 500 ns，sys_clk 翻转一次。

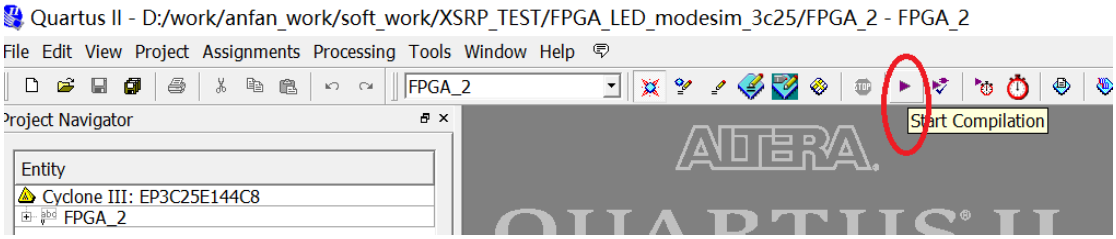
4、编译测试文件

编译当前的 fpga_sim.v 文件，如果有错语法就需要根据提示修改。

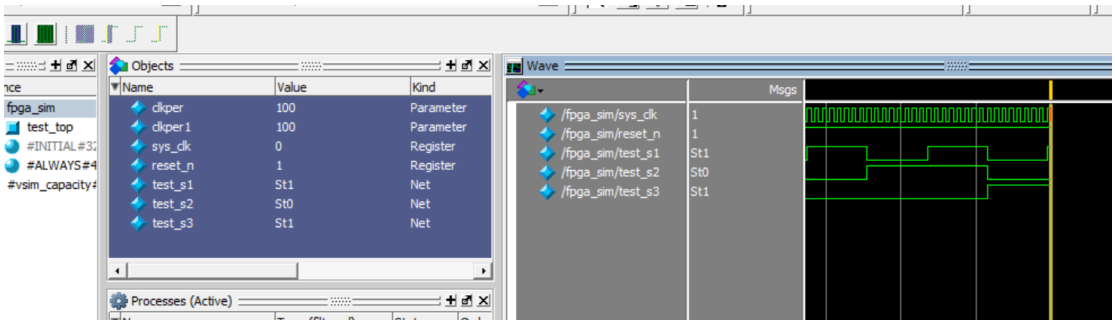


5、编译整个工程

编译整个工程。

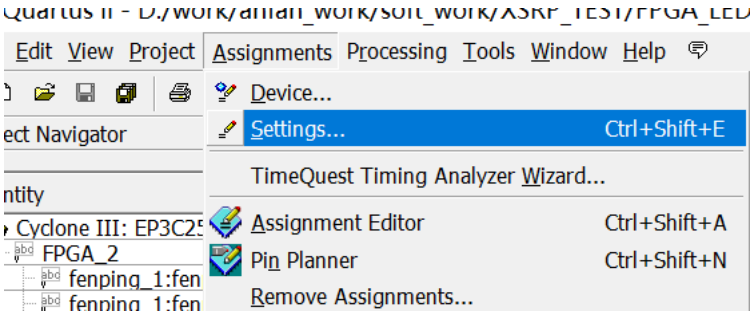


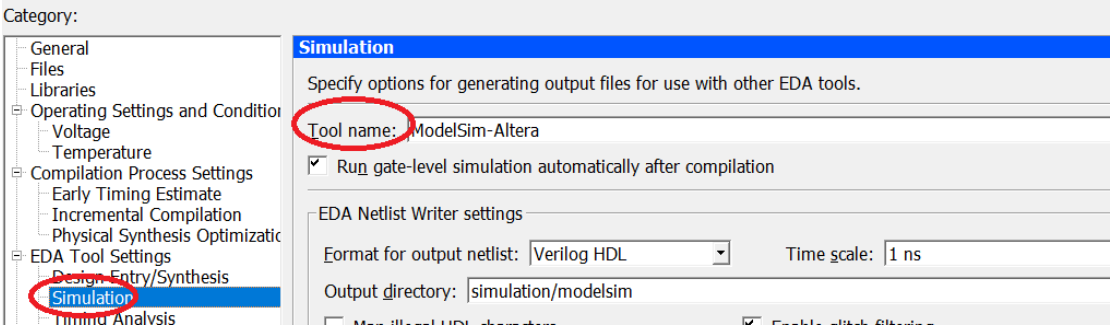
可以看出，各信号的波形了。



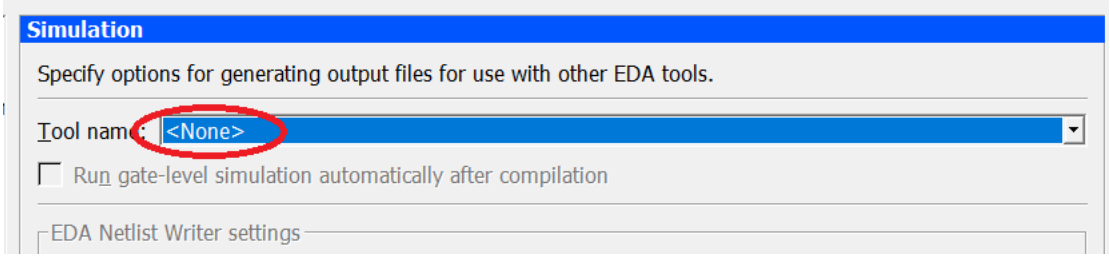
5、关闭仿真

如果在后面的设计工作中，不需要仿真，可以关闭仿真功能，即如下图，将Tool name:<NOne>即可。



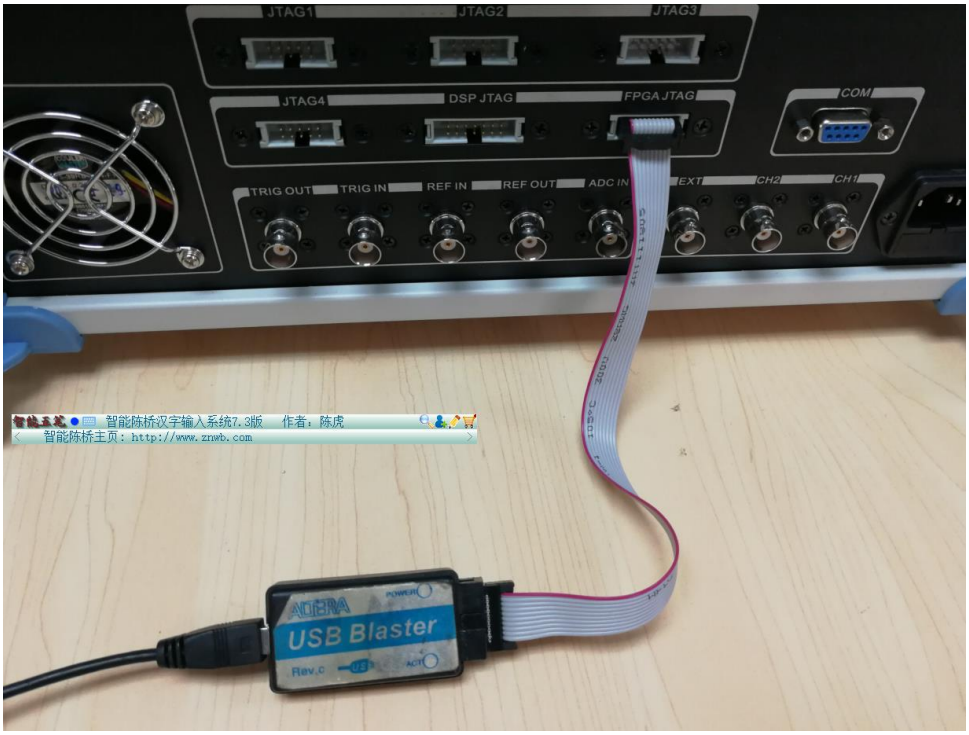


不开仿真功能：

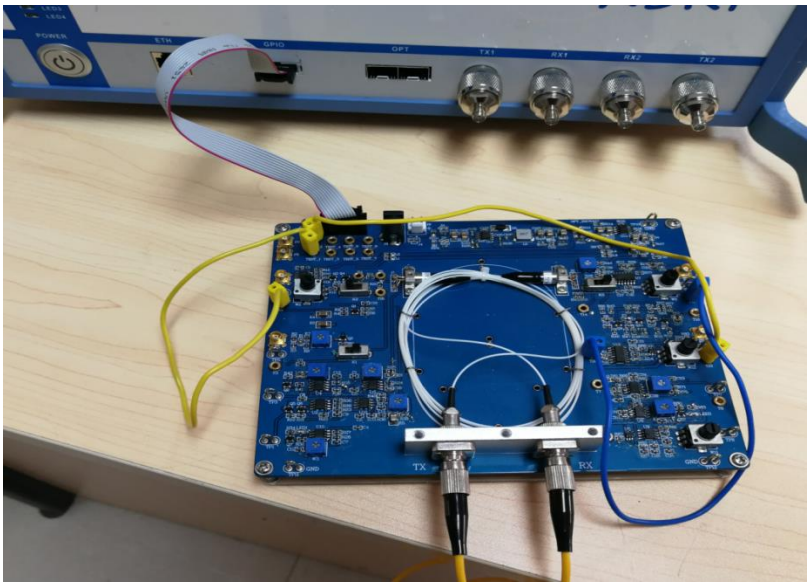


六、程序下载

准备工作：FPGA 编程时，FPGA 下载器的 USB 端接电脑的 USB 接口，另一端的 10pin 的连接线接 XSRP 后面的“FPGA　JTAG”（“FPGA　JTAG”对应设备内部的 EP3C25E144 的 FPGA 芯片）。

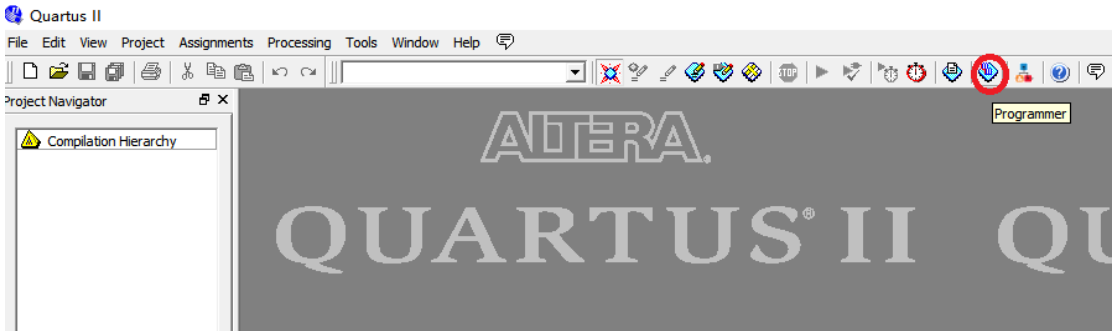


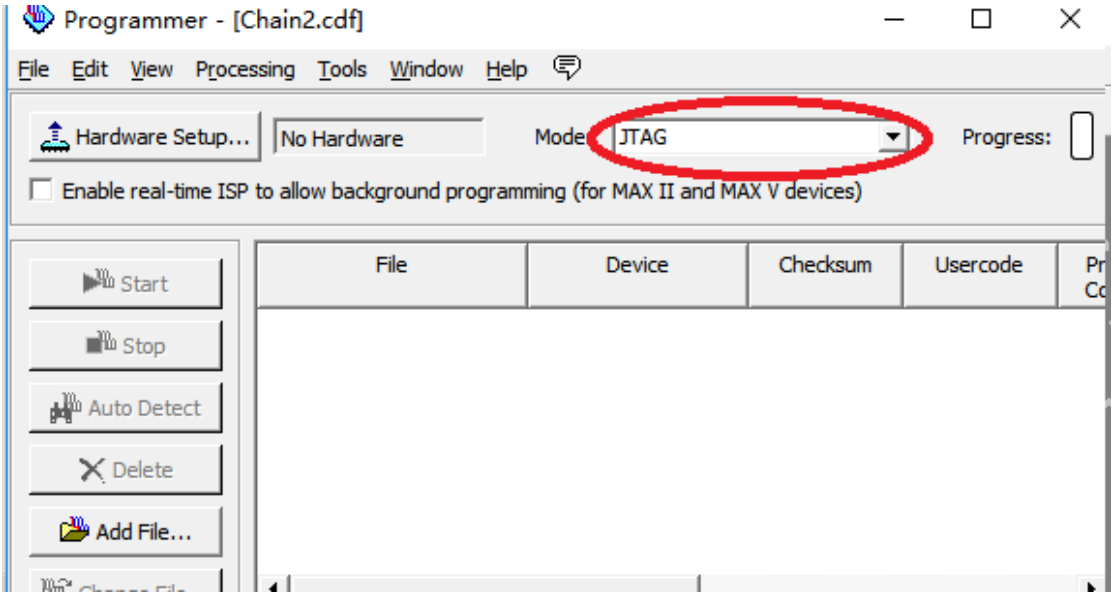
光通信模块的排线接口“J3”与 XSRP 前面板的“GPIO”接口用 10pin 的排线连接。



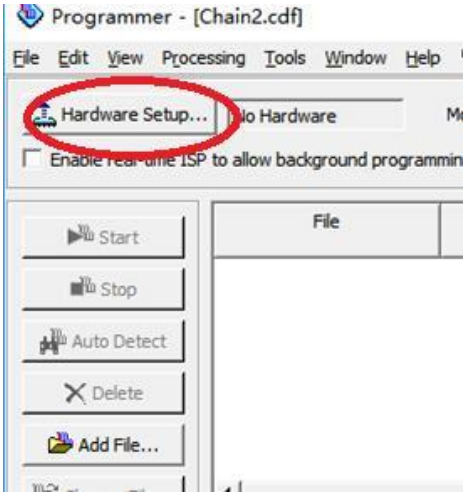
线缆连接完成这后，再给 XSRP 上电。

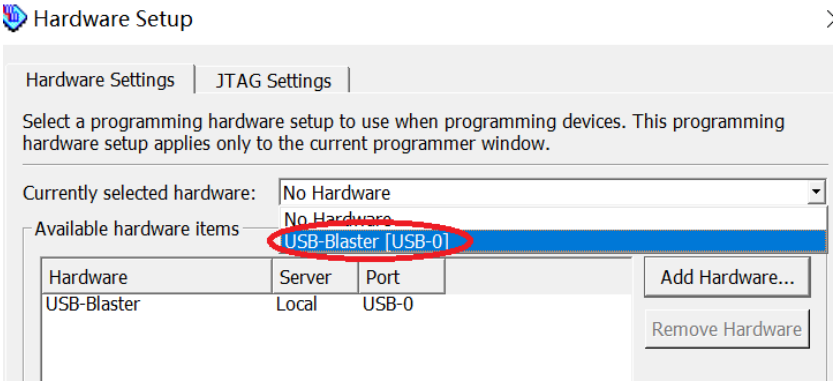
1、点下载程序





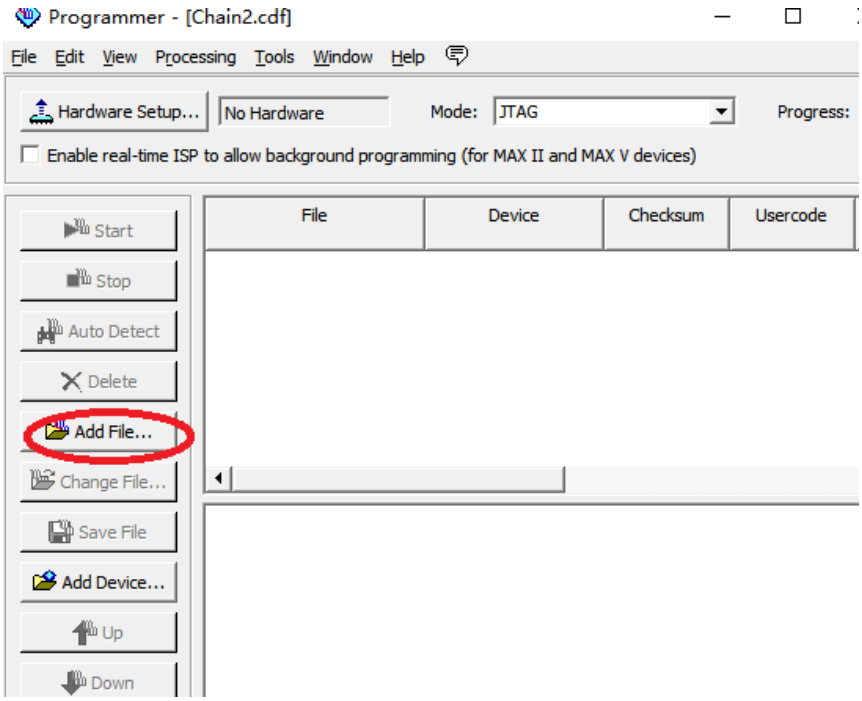
2、查找下载器



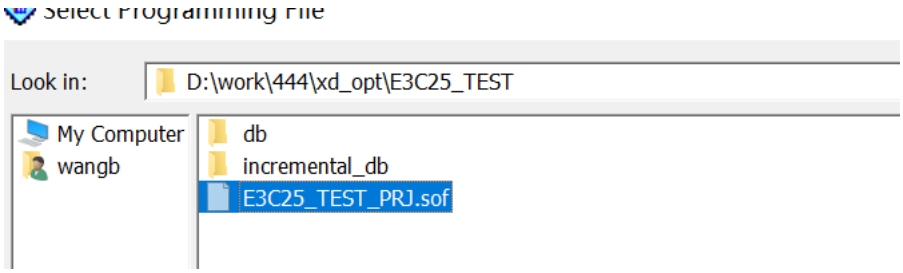


上图中要出现 USB 仿真器，并选中。

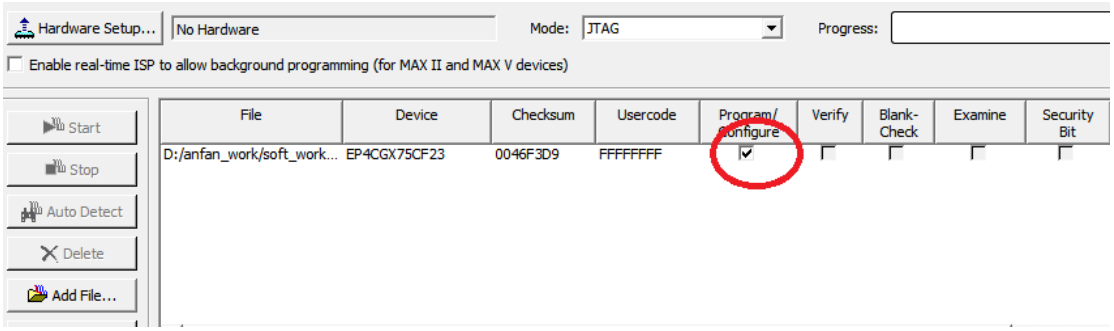
3、加入要下载的文件



如下图，找到当前工程生成的*.sof 文件

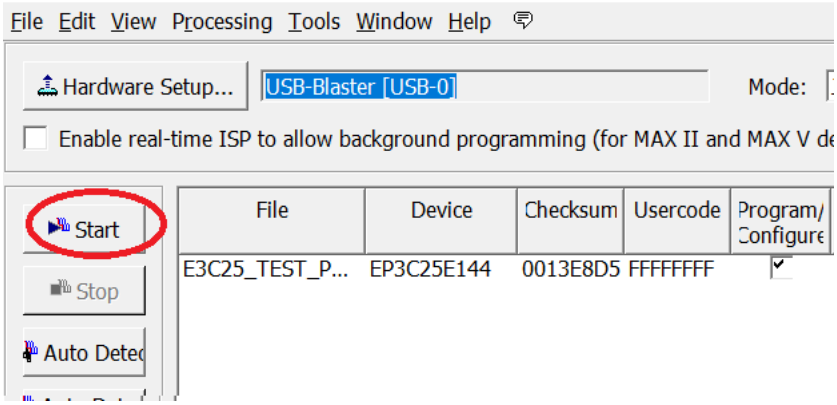


如下图，找钩。

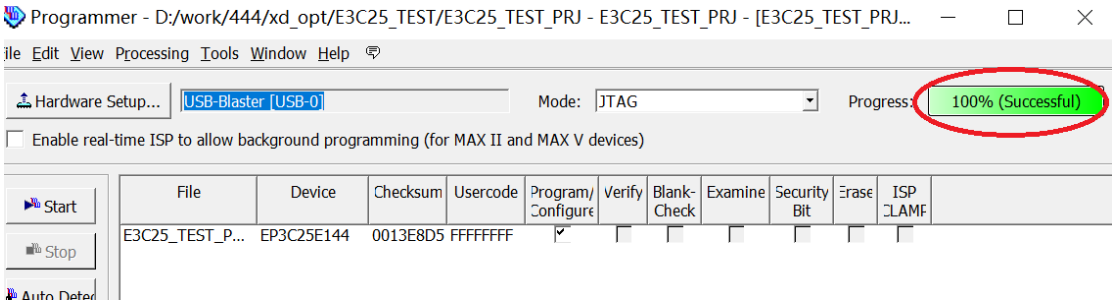


4、下载

点“Start”就可以点了。程序就会下载



下载过程中，下图会显示进度条，直到 100%完成。



下载：（一定要在 11 版本中进行下载，否则不能生成最新的 sof 文件）

Tools-Programmer。

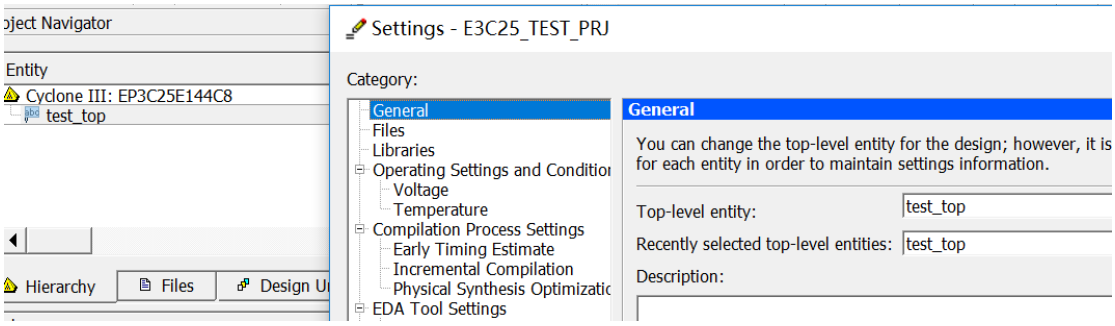
选择最新生成的 sof 文件，若有其他文件，可以右击文件-Delete，选择删除。

5、观察波形

利用示波器，观察波形输出。

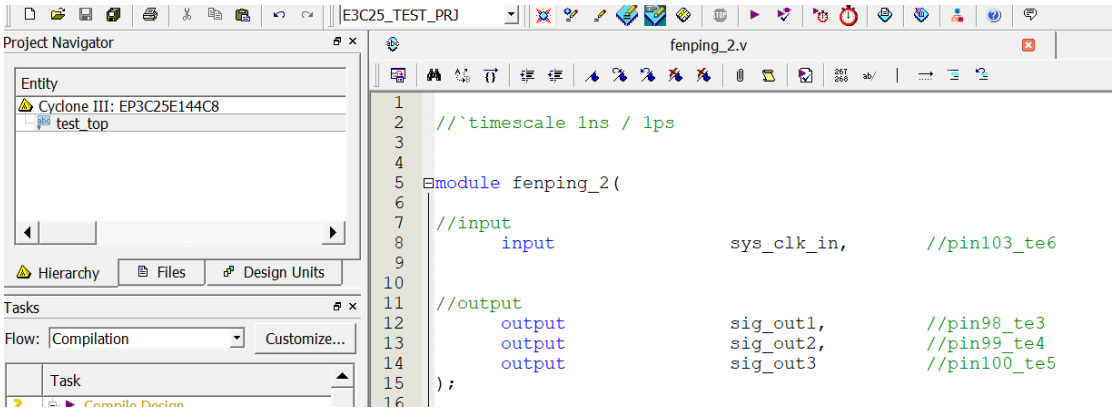
七、更换顶层文件

如果用户编写了另一个＊.v 文件（如 fenping_2.v），想把这个文件当顶层文件。如：原来的顶层文件为 test_top.v。



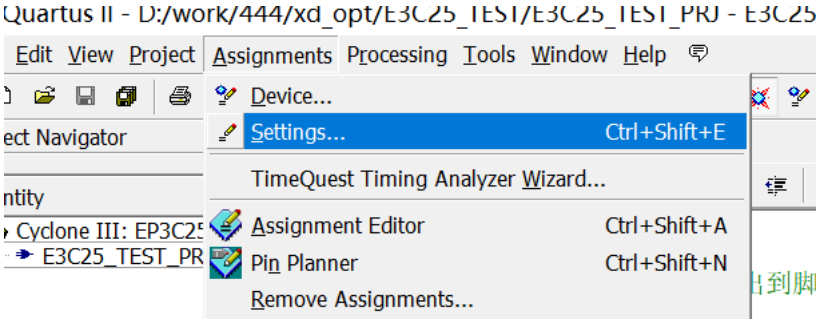
1、 先把 fenping_2.v 编写好。

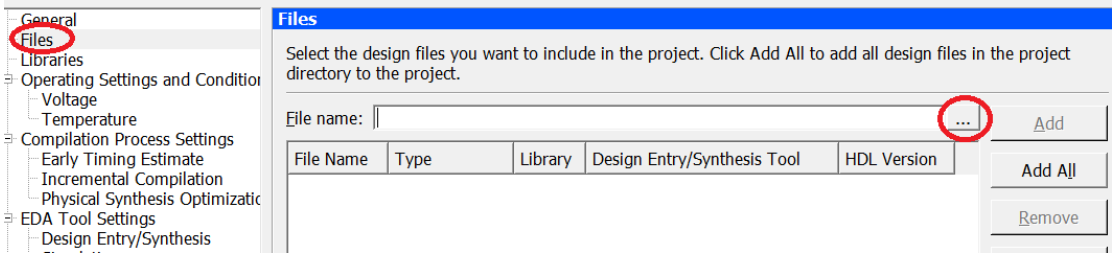
如下图：

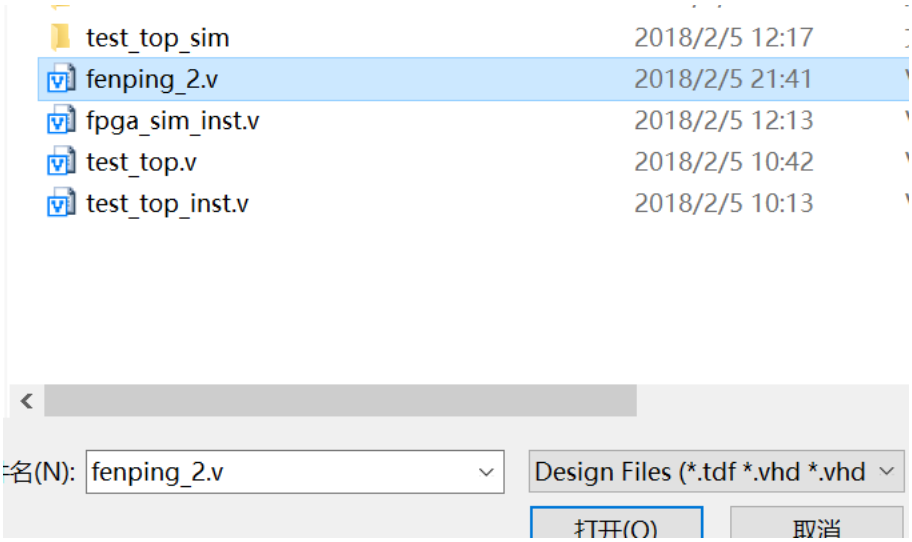


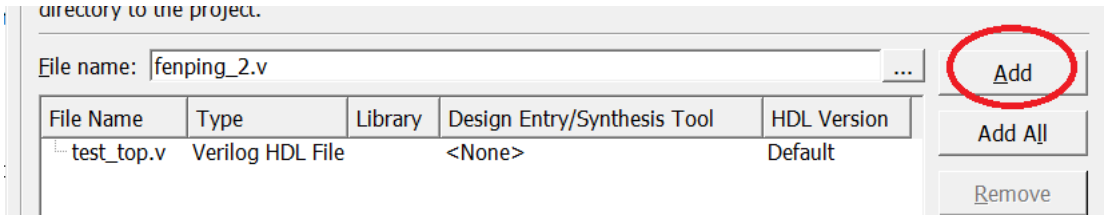
2、将文件加入到本工程中

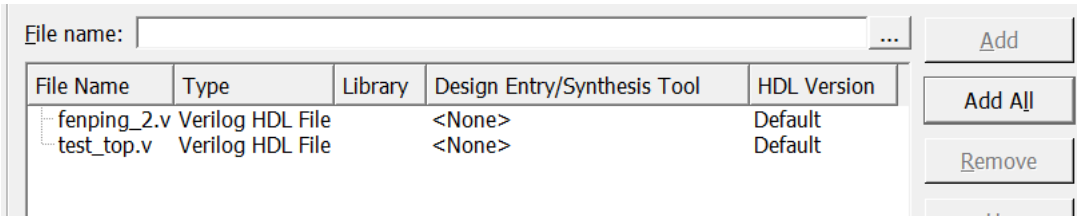
加该文件到工程中：





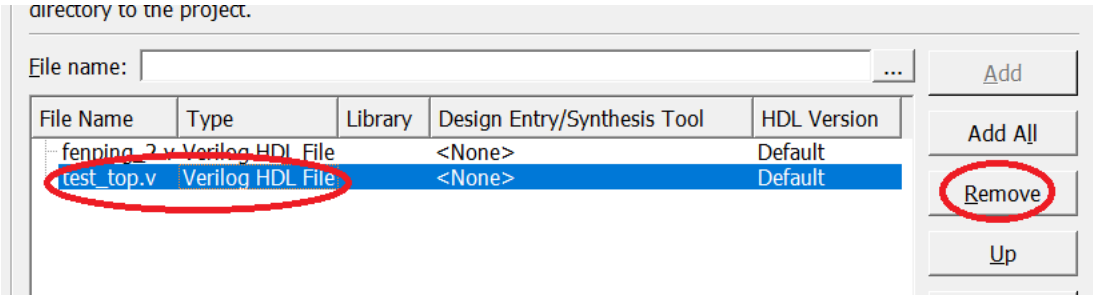




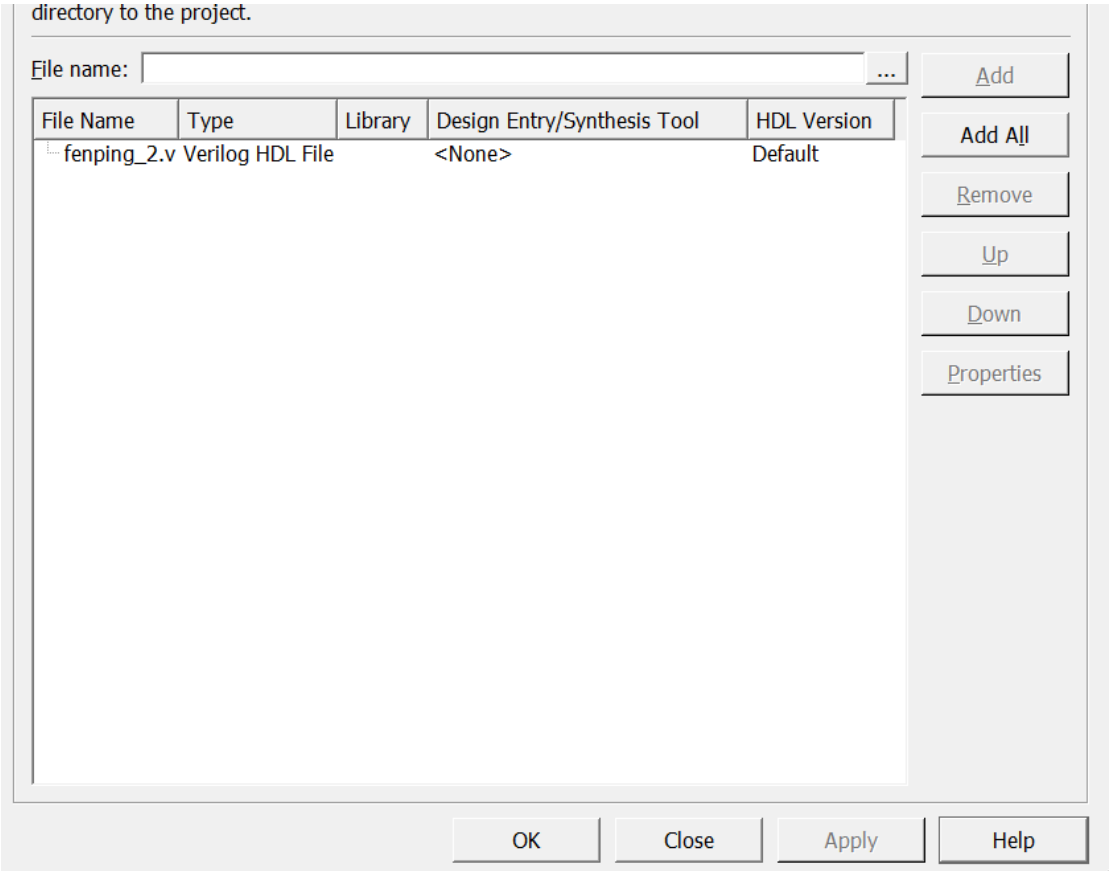


你可以根据需要求，原文件可以删除或保留在该项目中。

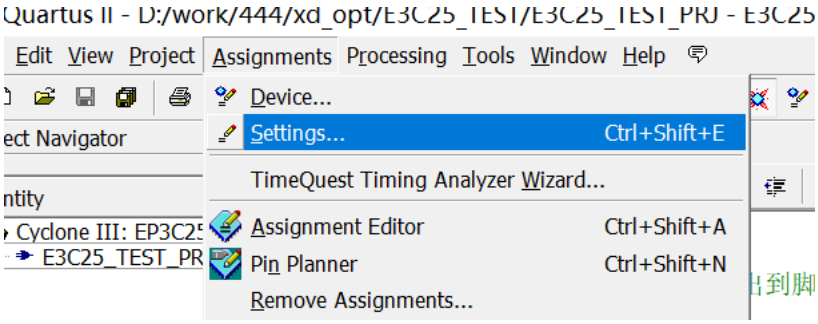
下张图是将文件从项目中移出（删除）：

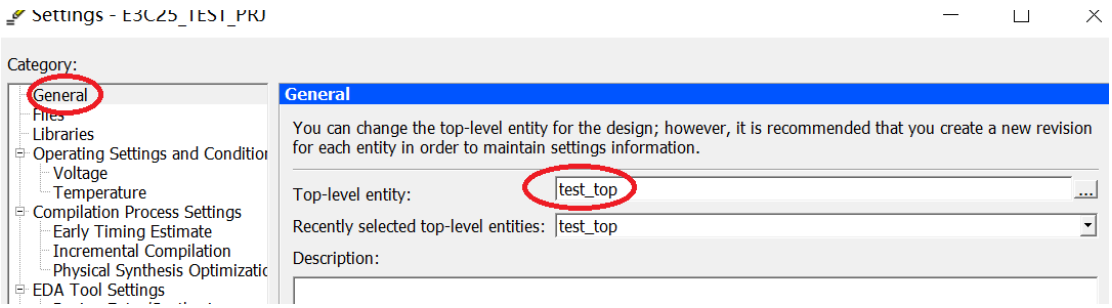


然后点“Apply”，再点“OK”

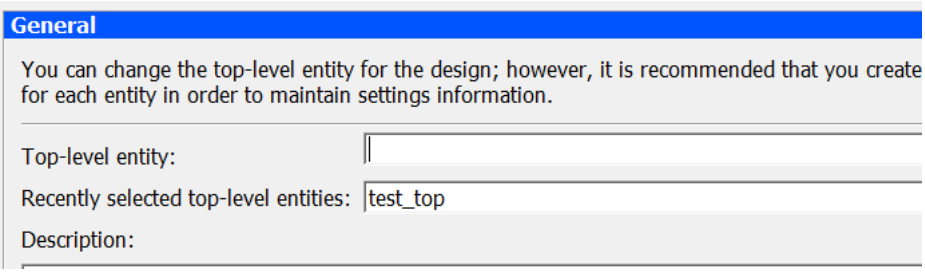


3、更换顶层文件操作



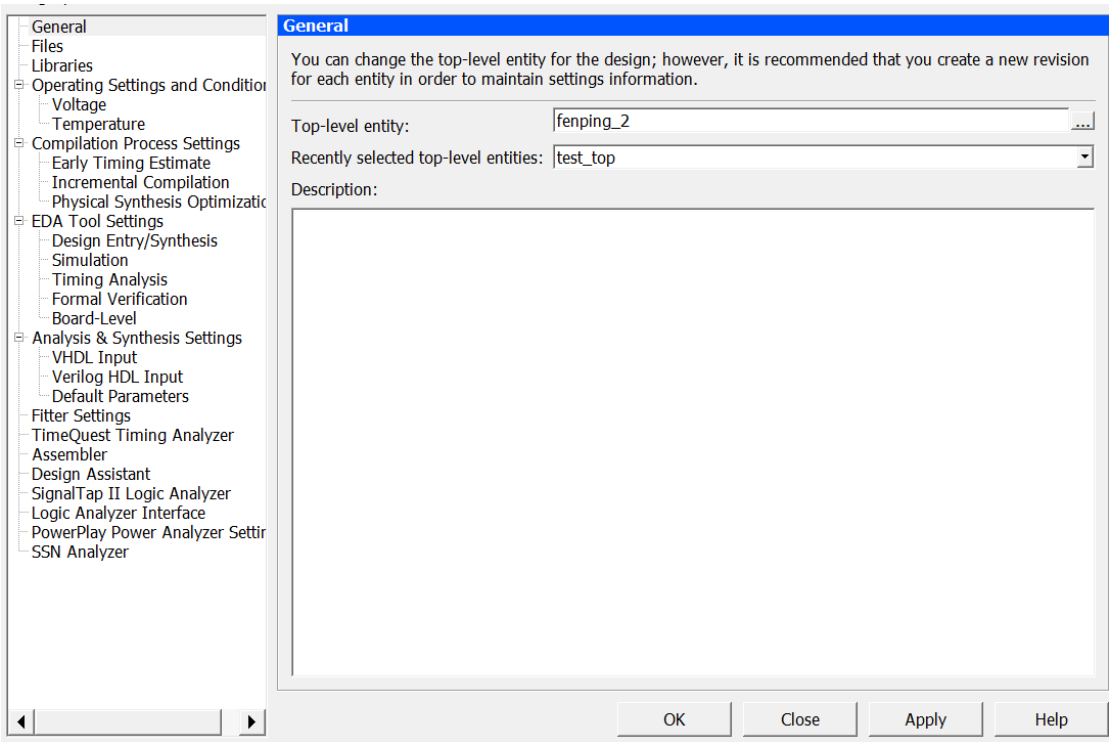


删除 Top – level entity:　后面的“tesp_top”文字

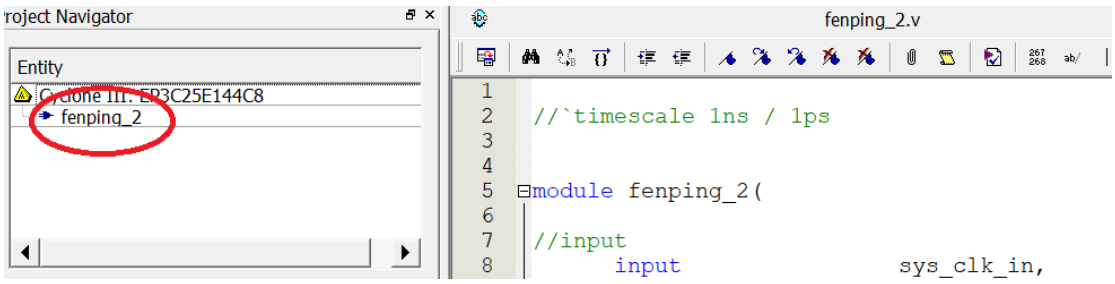


里面再输入：“fenping_2”

然后点“Apply”，再点“OK”



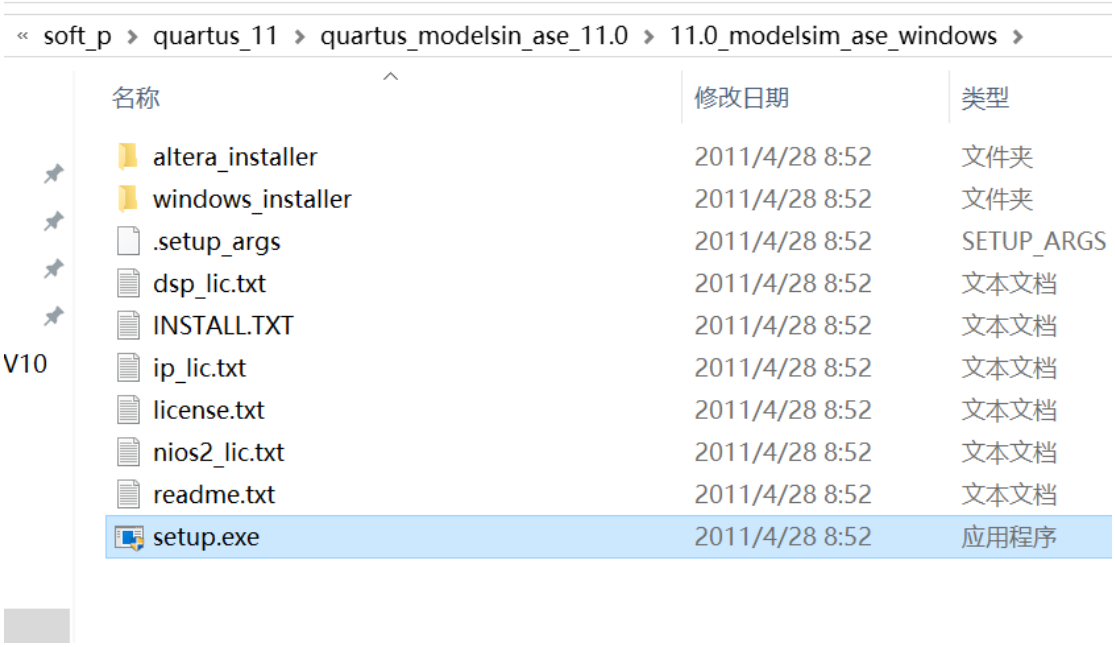
这时顶层文件已改为“fenping_2”了。

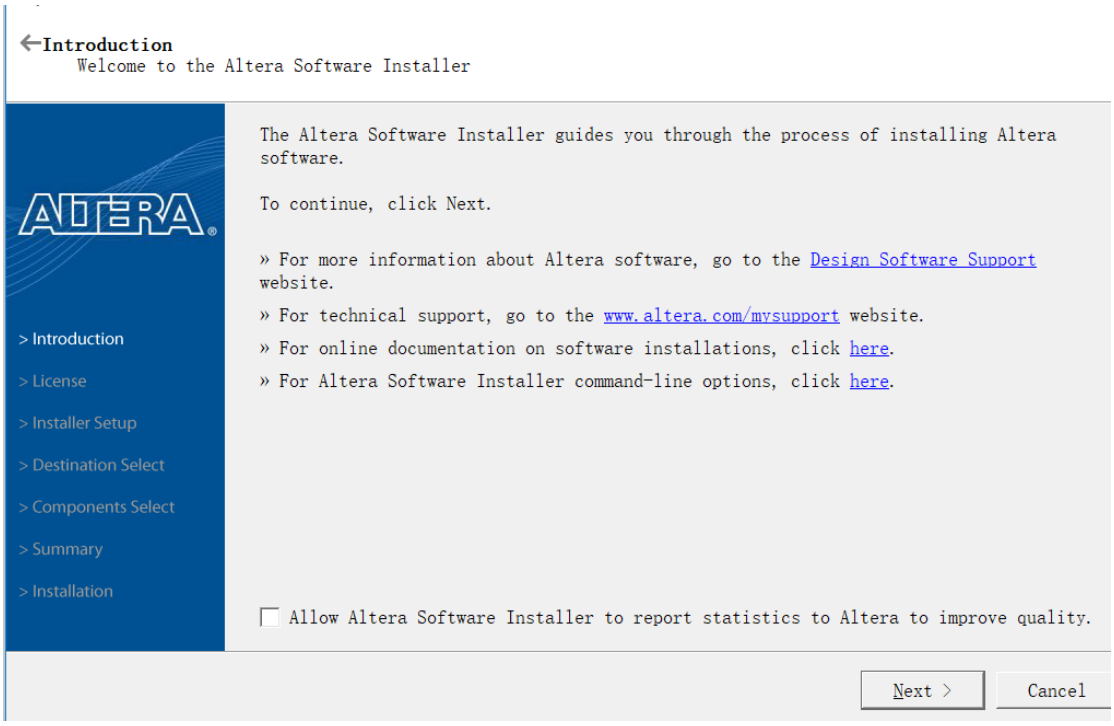


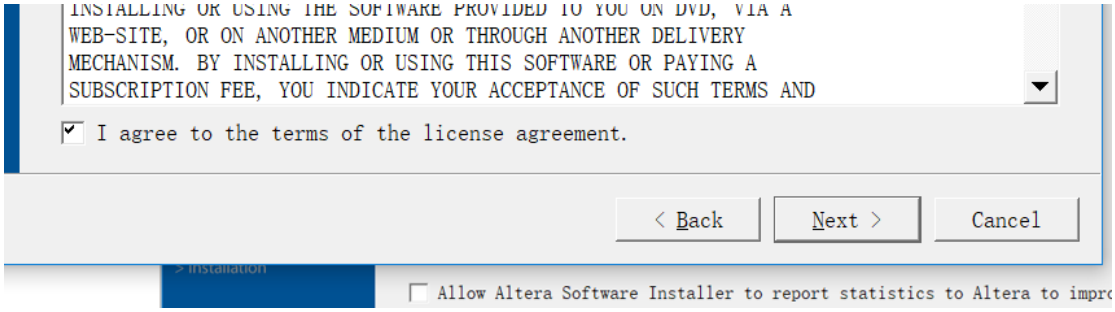
附 1、安装 quartus_11

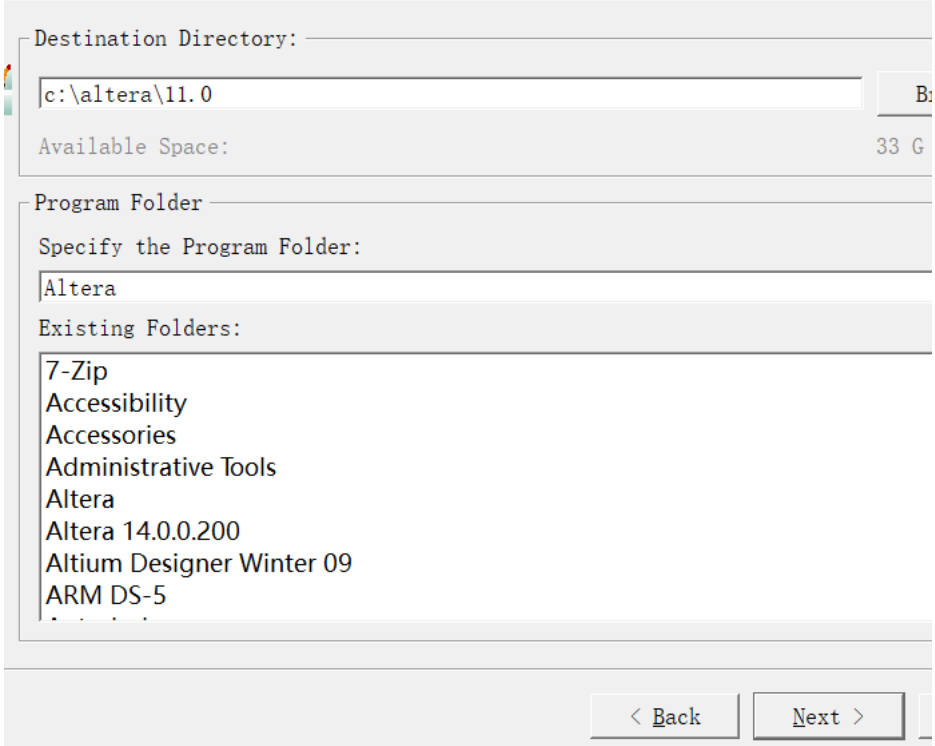
见相关文档，这里不详细描述。

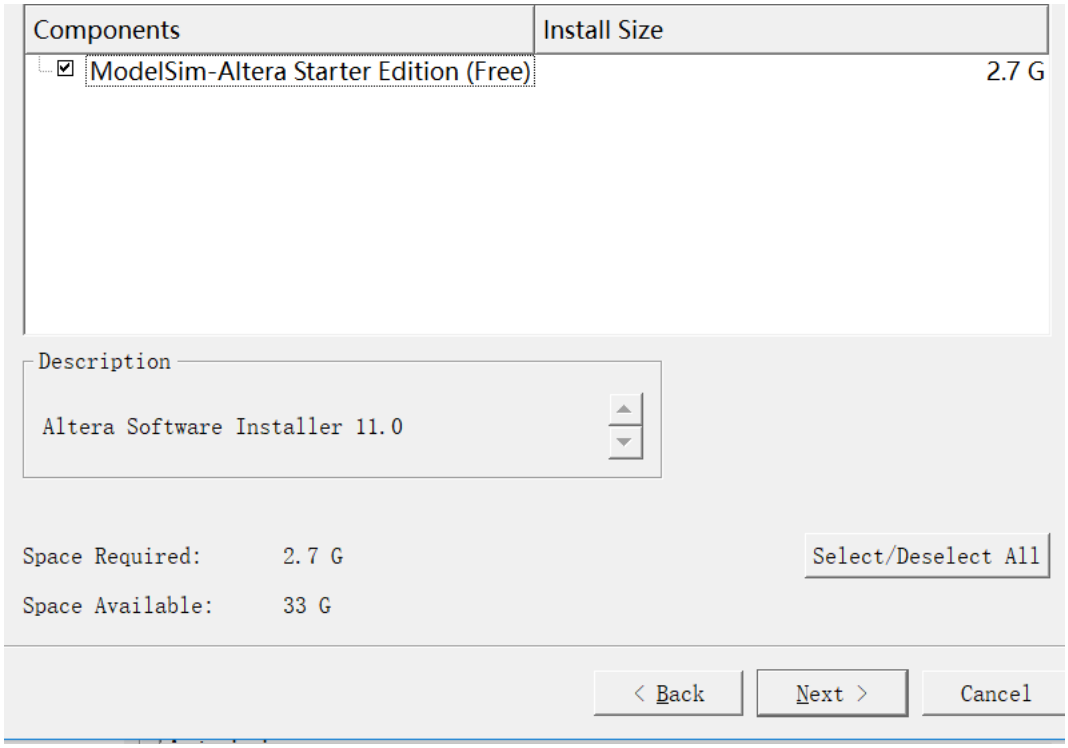
附 2、安装 modelsim



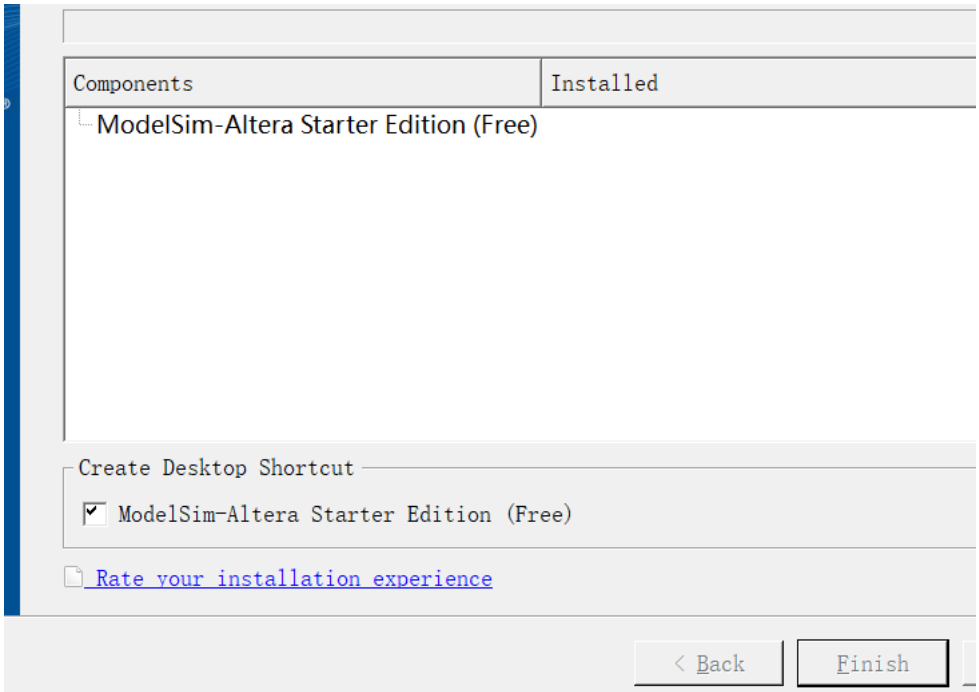





Destination Directory:
c:\altera\11.0
Available Space:
33 G
Program Folder
Specify the Program Folder:
Altera
Existing Folders:
7-Zip
Accessibility
Accessories
Administrative Tools
Altera
Altera 14.0.0.200
Altium Designer Winter 09
ARM DS-5
< Back
Next >

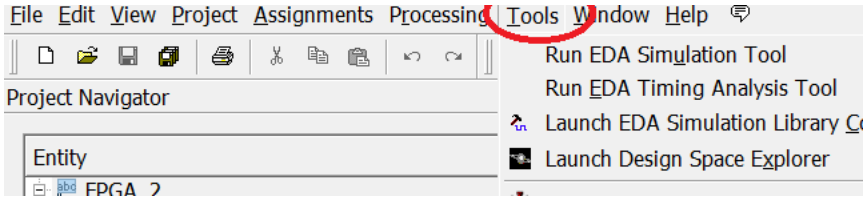

Components
Install Size
ModelSim-Altera Starter Edition (Free)
2.7 G
Description
Altera Software Installer 11.0
Space Required:
2.7 G
Space Available:
33 G
Select/Deselect All
< Back
Next >
Cancel

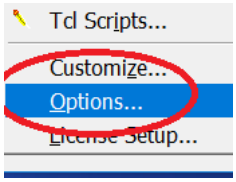
安装之中。。。。。

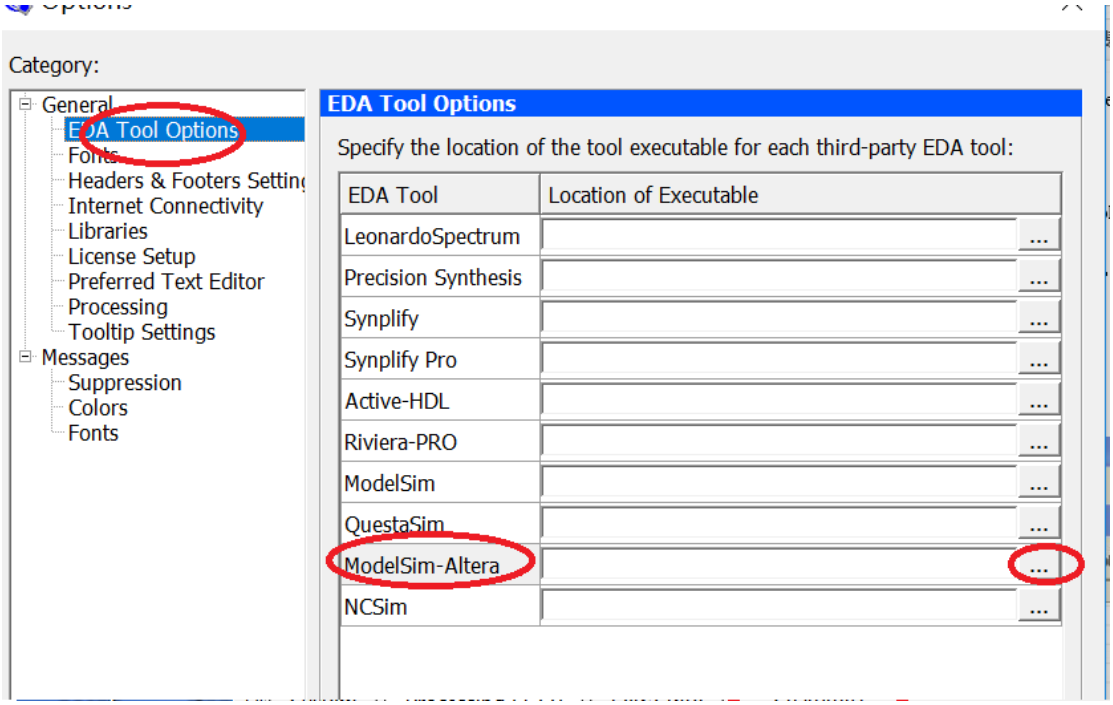


附 3、Modelsim 关联配置

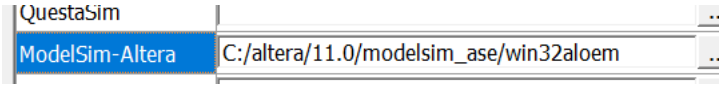
主要目的是让 Quartus 能调用 ModelSim







选中 modelsim 的位置：



附 4、EP3c25e144C8 管脚连接说明

FPGA　PIN_98　　说明：PIN_98 与光通信模块的 TE3 测试孔连接。

FPGA　PIN_99　　说明：PIN_99 与光通信模块的 TE4 测试孔连接。

FPGA　PIN_100　说明：PIN_100 与光通信模块的 TE5 测试孔连接。

FPGA　PIN_103　　说明：PIN_103 与光通信模块的 TE6 测试孔连接。

FPGA　PIN_87　说明：PIN_87 与内部的 12MHz 晶振连接，只能做信号输入。

FPGA　PIN_88　说明：PIN_88 与内部的 12MHz 晶振连接，只能做信号输入。

FPGA　PIN_89　说明：PIN_89 与内部的 7.68MHz 晶振连接，只能做信号输入。

FPGA　PIN_132 说明：PIN_132 与内部的 7.68MHz 晶振连接，只能做信号输入。

附 5、其它说明

做二次开发时，光纤通信模块的 TE1，有 7.68MHz 时钟输出，TE1 内部也是连接 7.68MHz 时钟，用户可以使用这一时钟信号。

附 6、内、外部时钟选择

```
set_global_assignment -name PARTITION_NETLIST_TYPE SOURCE -section_id
set_global_assignment -name PARTITION_FITTER_PRESERVATION_LEVEL PLACEM
set_global_assignment -name PARTITION_COLOR 16764057 -section_id Top
set_global_assignment -name VERILOG_FILE test_top.v



set_location_assignment PIN_98 -to sig_out1
set_location_assignment PIN_99 -to sig_out2
set_location_assignment PIN_100 -to sig_out3
#set_location_assignment PIN_103 -to sys_clk_in

set_location_assignment PIN_89 -to sys_clk_in
```

1、内部时钟：

set_location_assignment PIN_89 -to sys_clk_in

sys_clk_in 这个时钟信号从 XSRP 内部 89 脚入。

2、外部时钟：

将这行代码 sys_clk_in 对应的管脚改为 PIN_103，如下：

set_location_assignment PIN_103 -to sys_clk_in

sys_clk_in 这个时钟信号从光通信模块的 TE6 测试孔输入。

3、验证外部时钟：

为了验证外部输入时钟是否可以工作，利用“附 5”的说明，可将 TE1 测试孔与 TE6 测试孔用香蕉插头连接线连接，程序下载后，利用示波器进行波形测试。