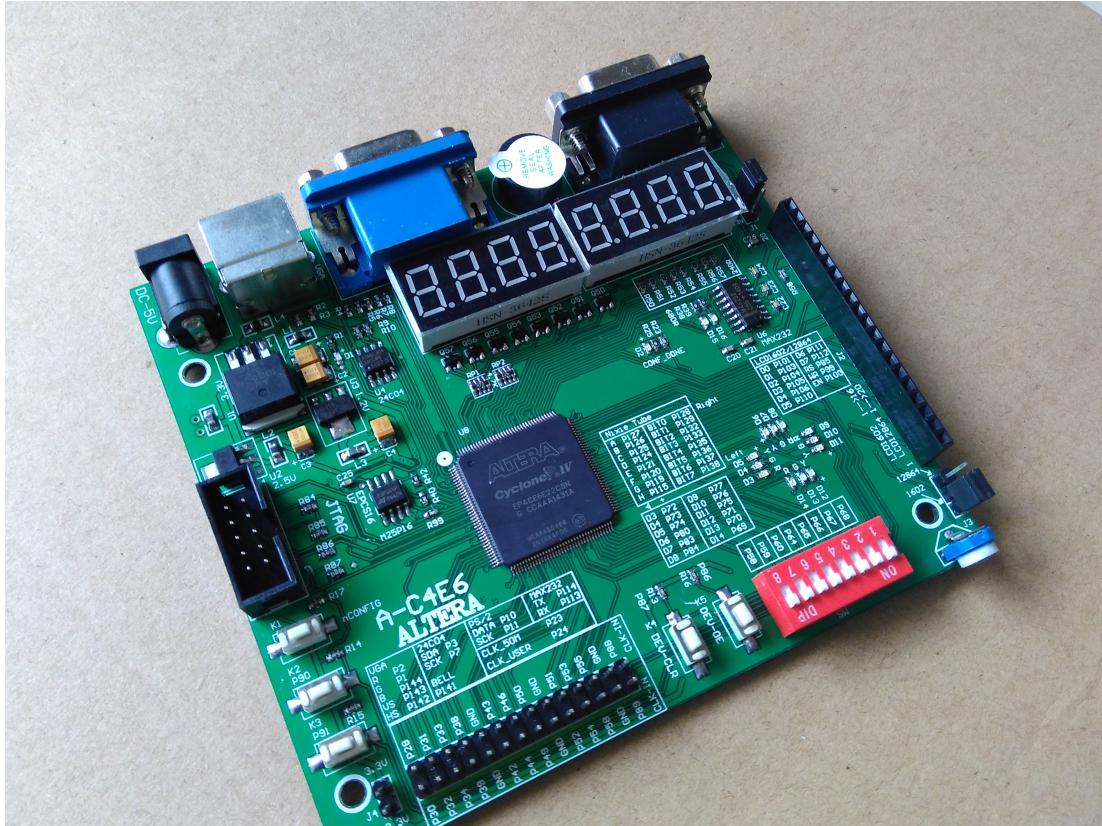


Email:906606596@qq.com

You need use QuartusII 11.0 or high version

Introduction how to use QuartusII , you need chose FPGA chip Cyclone IV EP4CE6E22C8N.
NOT EP2C8Q208C8N



First, open the QuartusII9.0SP2 software development. You have three ways to open.

The first: the following chart, on the desktop shortcut icon. Double-click the.





Second: as shown in the following figure, can be found in the start menu.



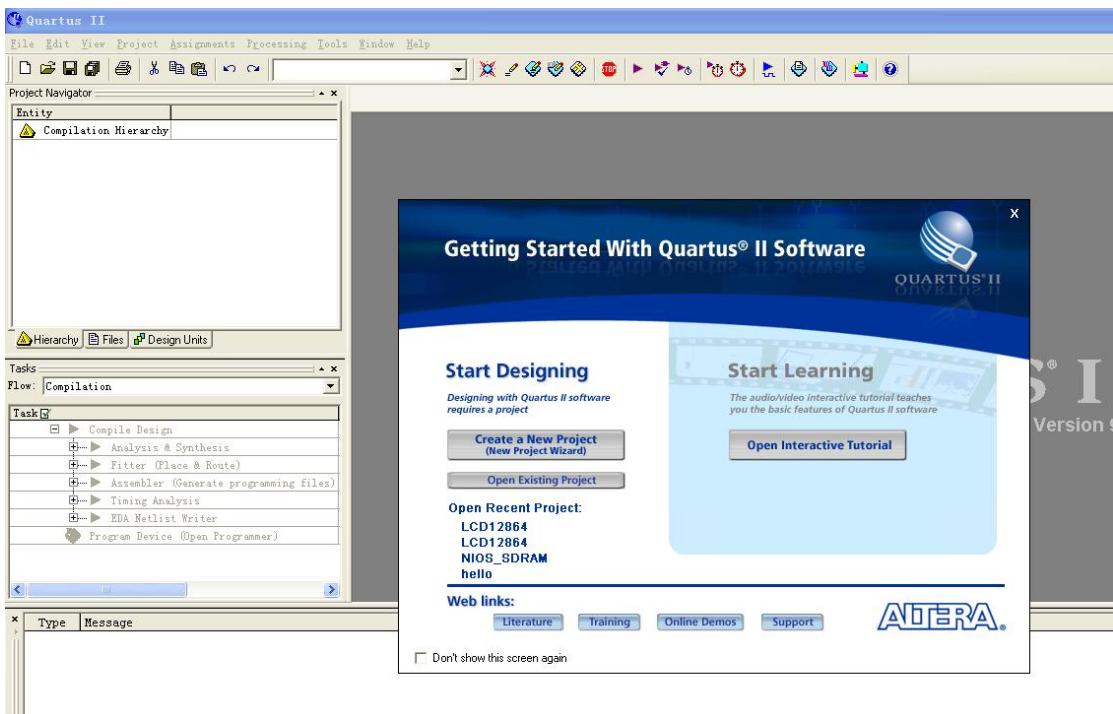
第三：如下图，在开始菜单里面也可以找到。



Email:906606596@qq.com



如下图：打开QII软件了



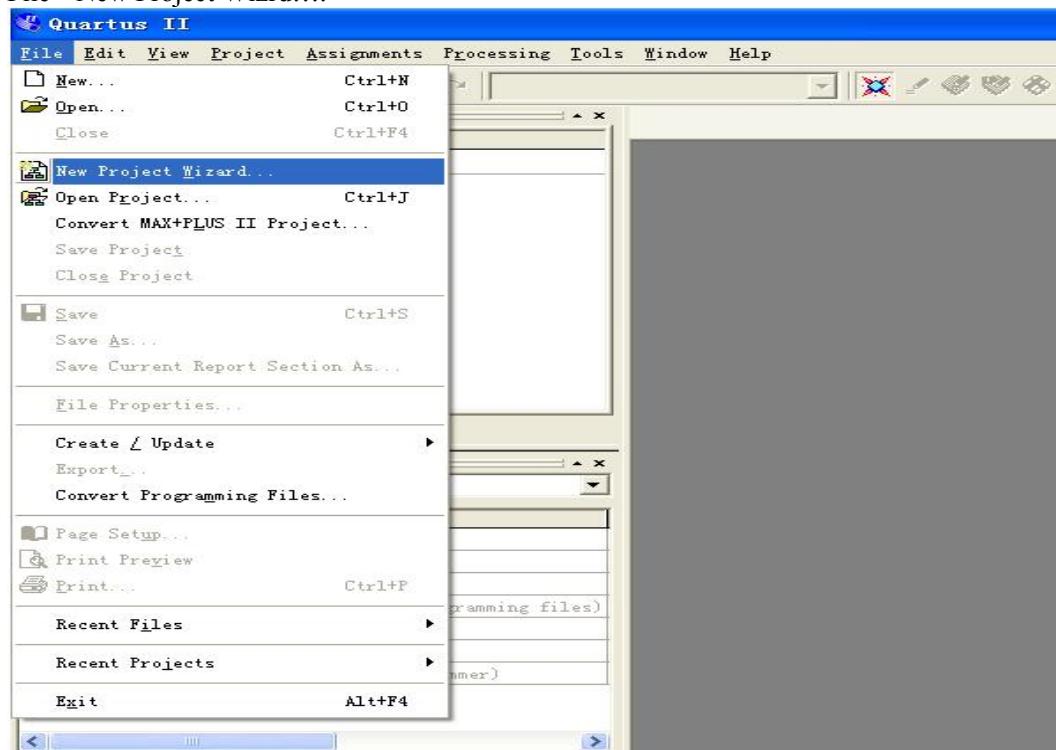
The following diagram: we can put this off.



Email:906606596@qq.com



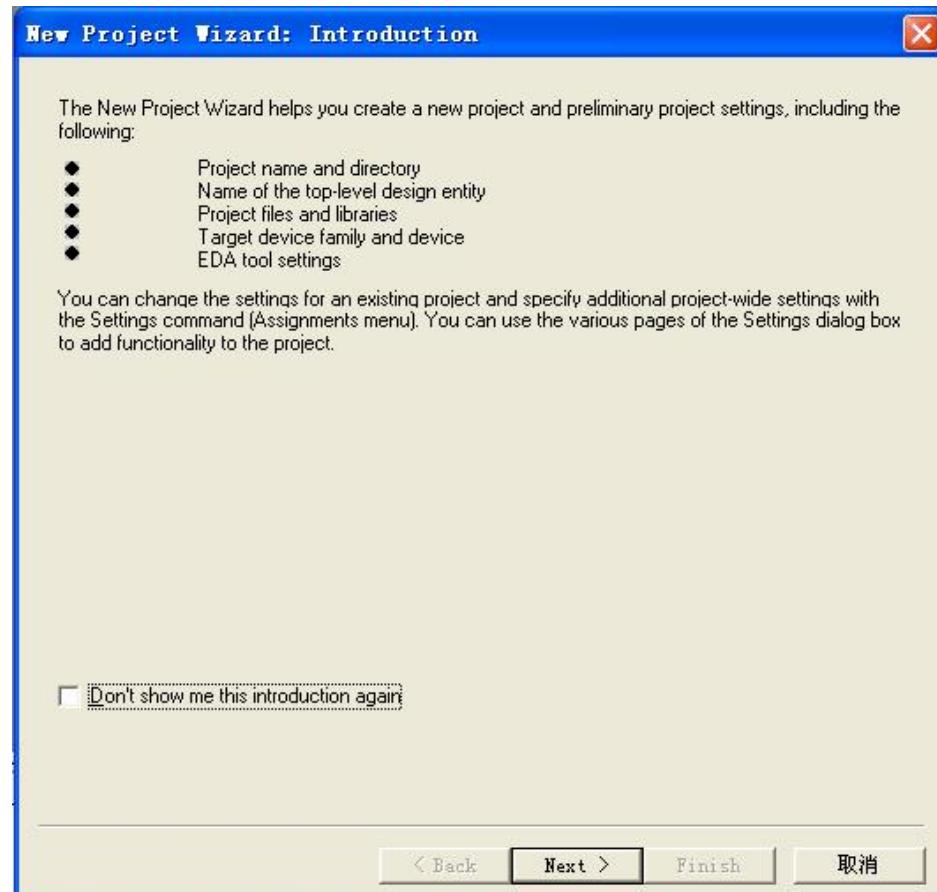
File->New Project Wizrd....



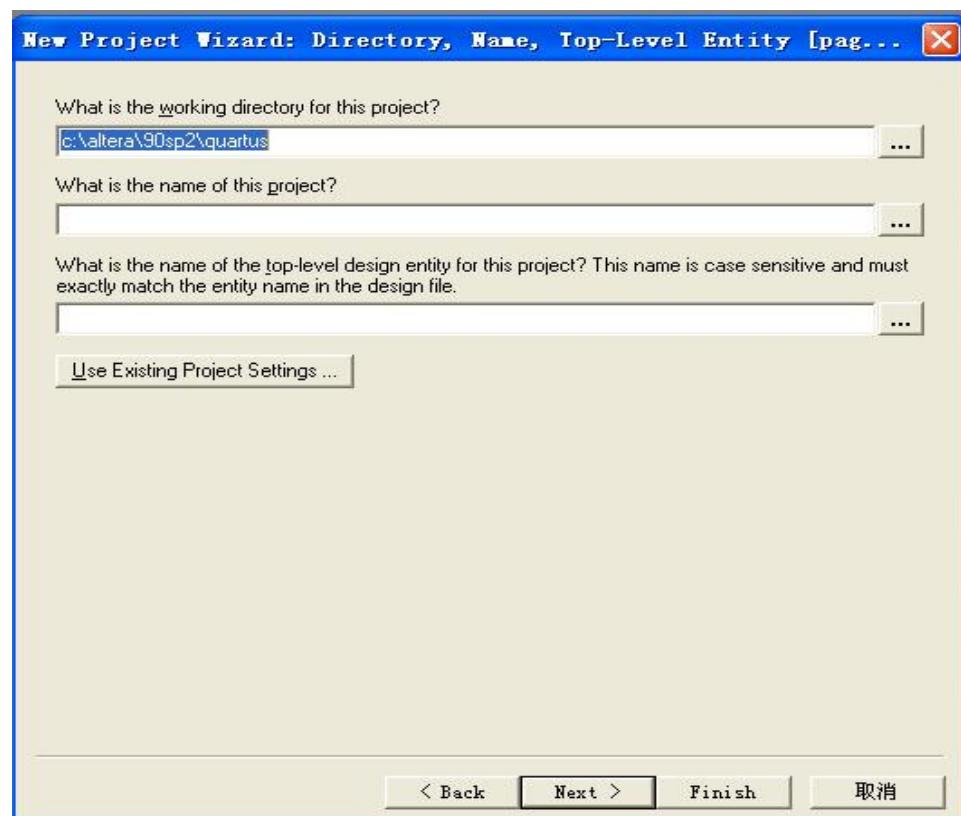
Next



Email:906606596@qq.com



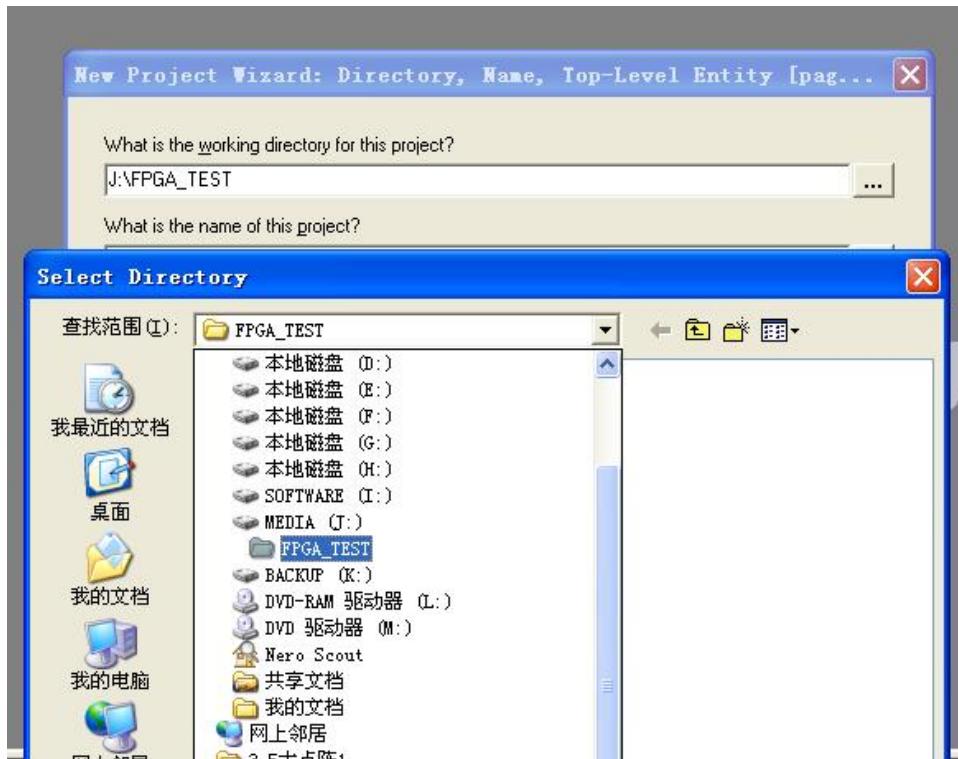
如下图：这里我们要设置工程路径，工程名和顶层名。点第一排三个点点那里



The project path J:\FPGA_TEST

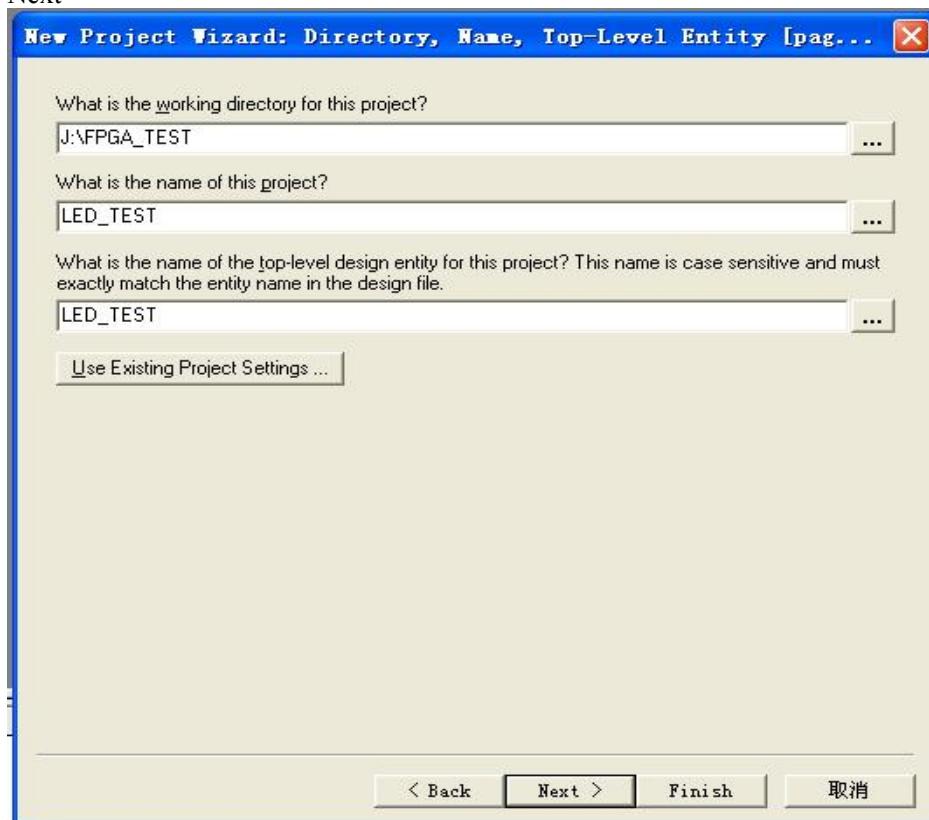


Email:906606596@qq.com



The following diagram: project name and the name can be the same, also can not be the same, we write here is the same.

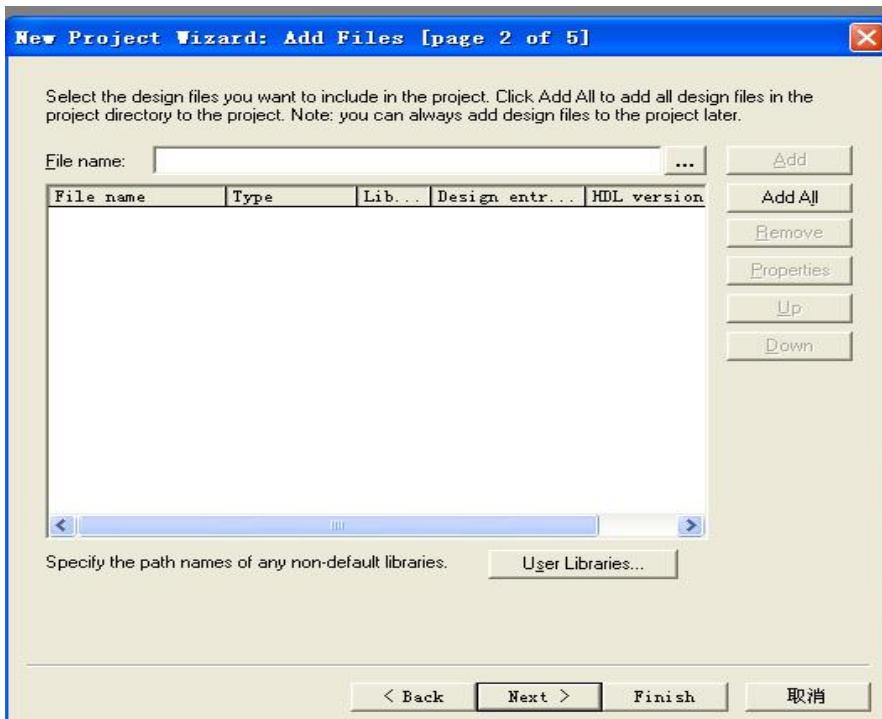
Next



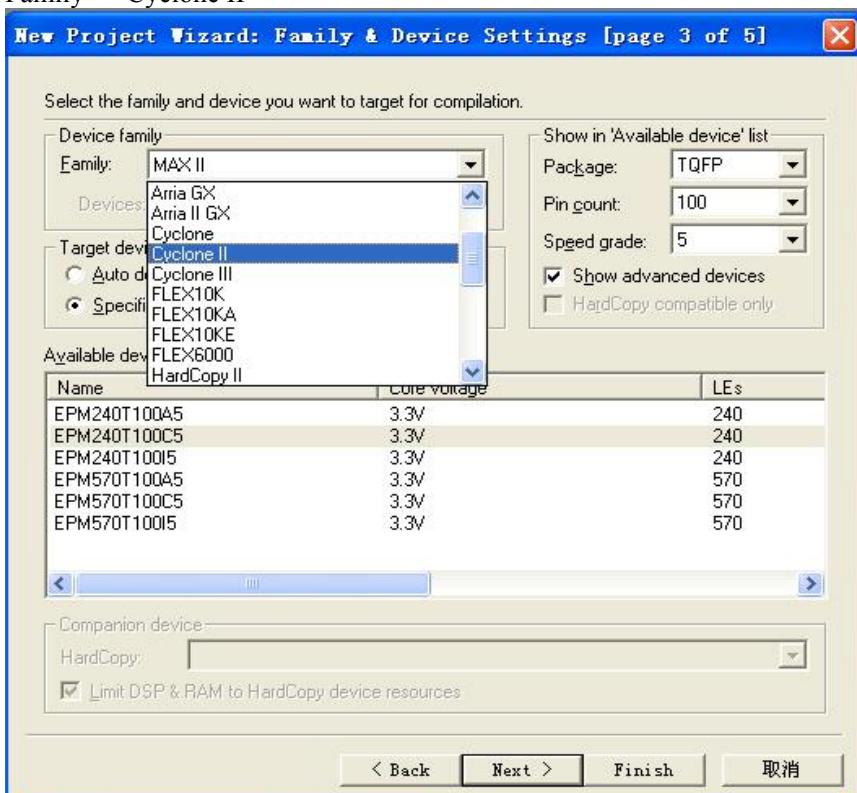
Next



Email:906606596@qq.com



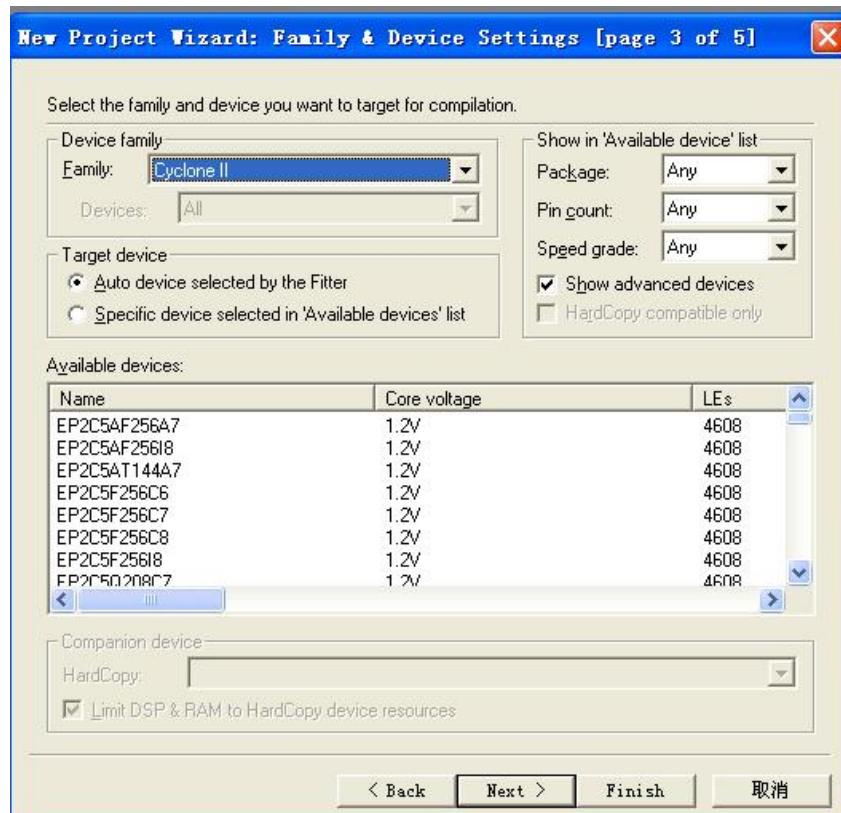
The following diagram: Here we need to set up FPGA, and we set the corresponding chip development board.. ° We use EP2C8Q208C8N as an example. Note that you are using the A-C2FB model, EP2C5T144C8N Family Cyclone II



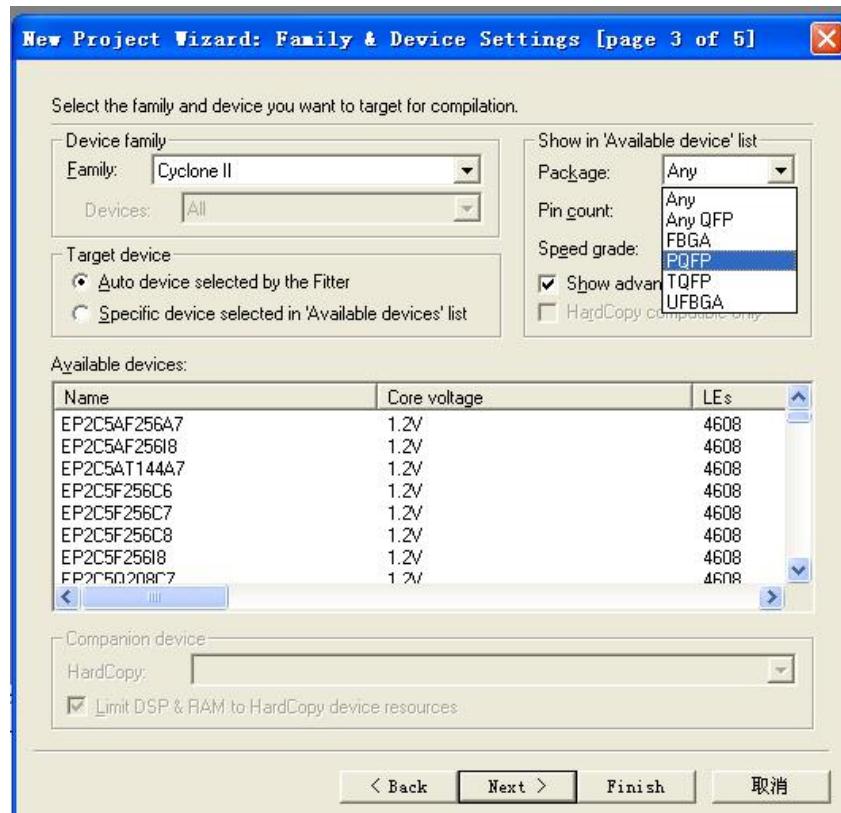
如下图：



Email:906606596@qq.com



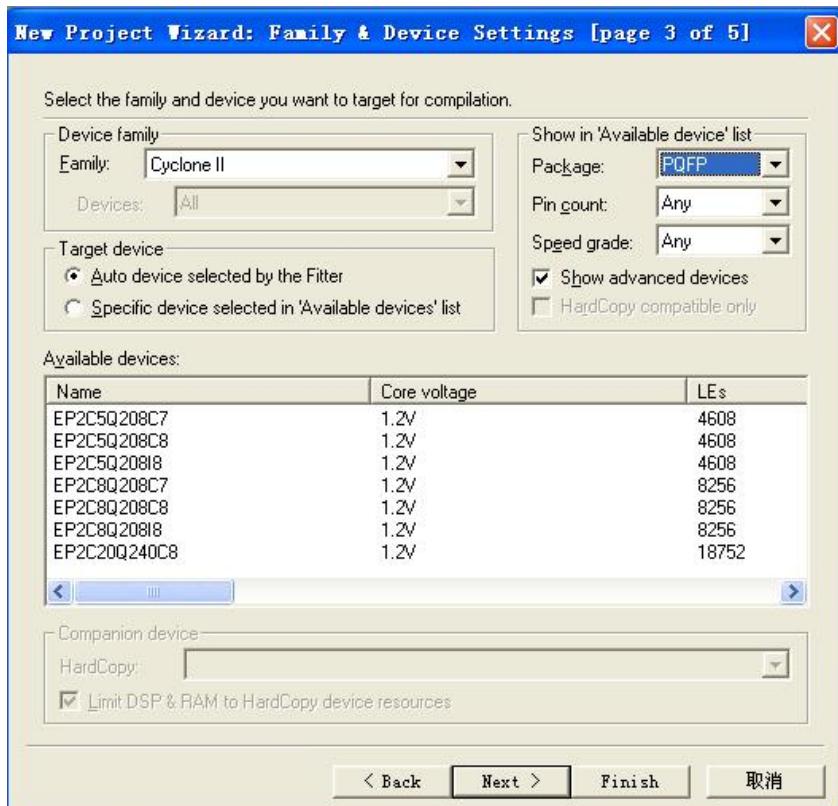
208 -> PQFP T144->TQFP



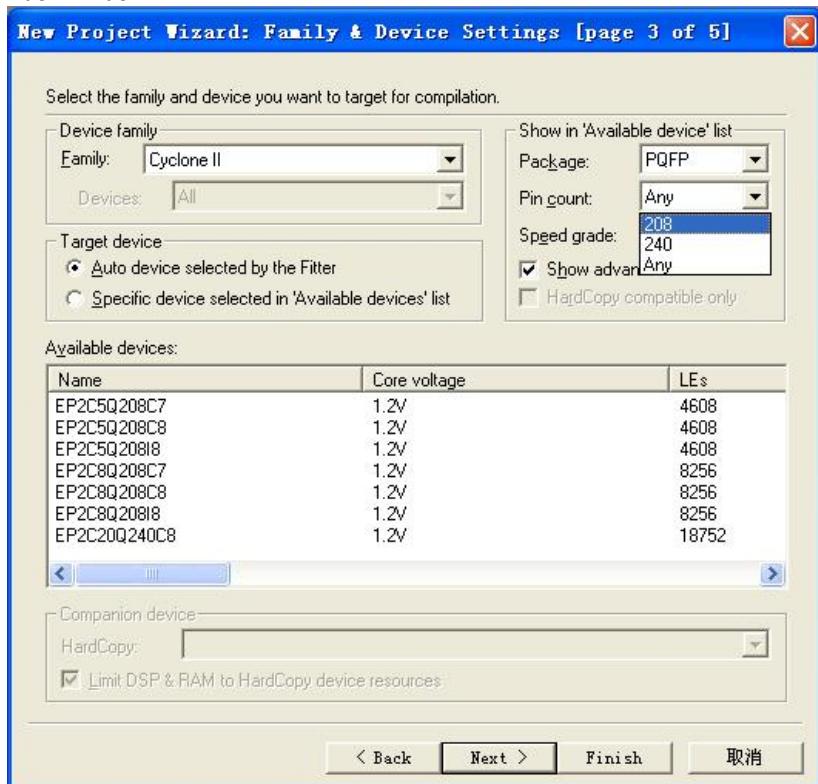
如下图：



Email:906606596@qq.com



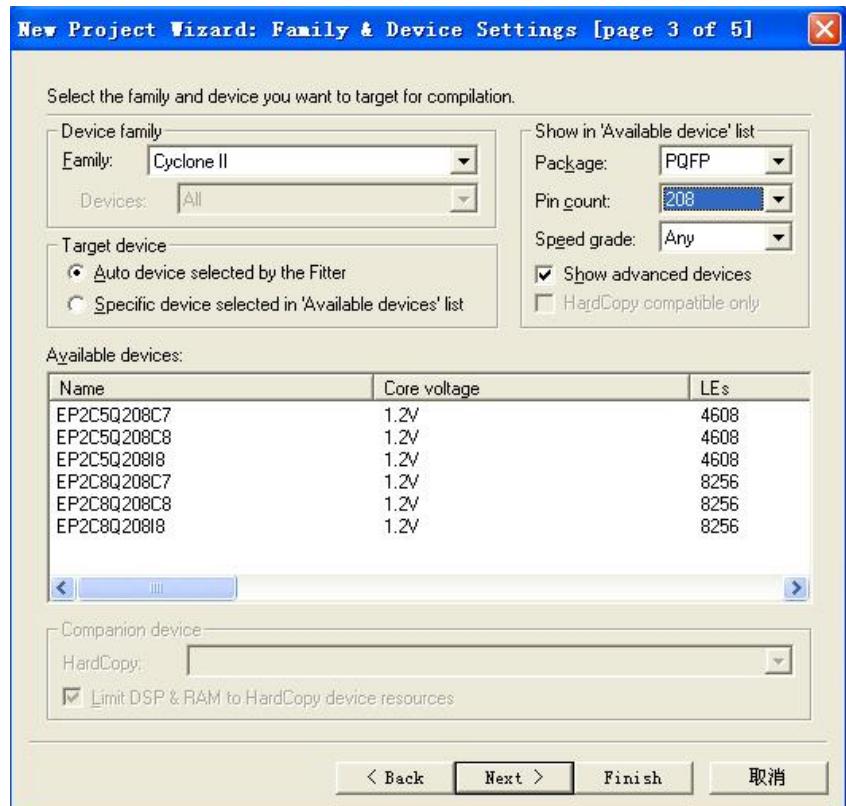
208 -> 208 144->144



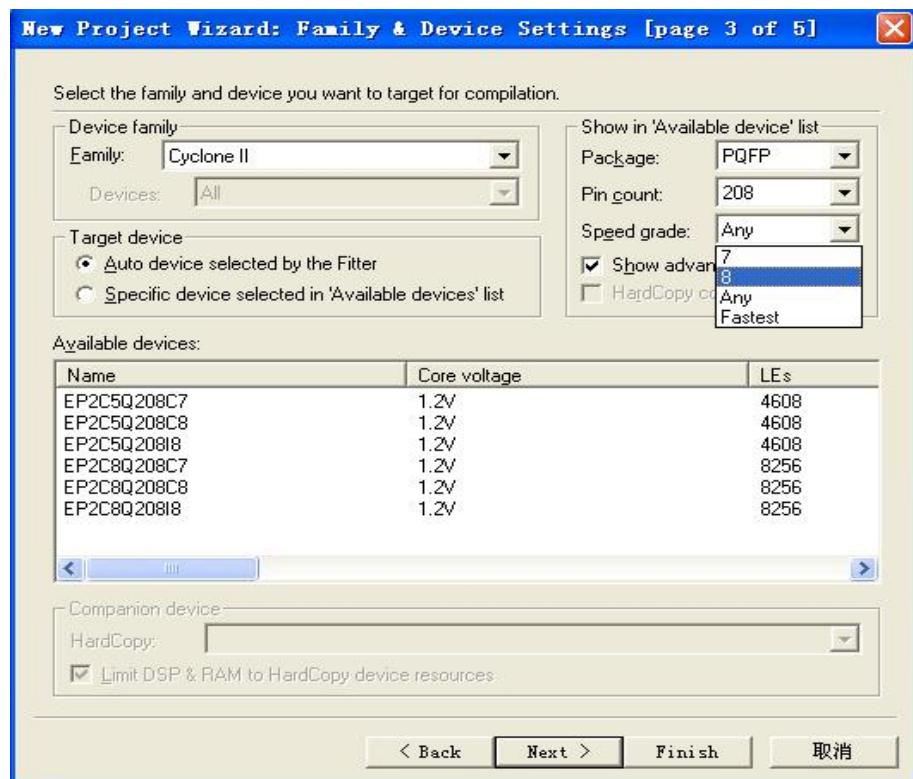
如下图：



Email:906606596@qq.com



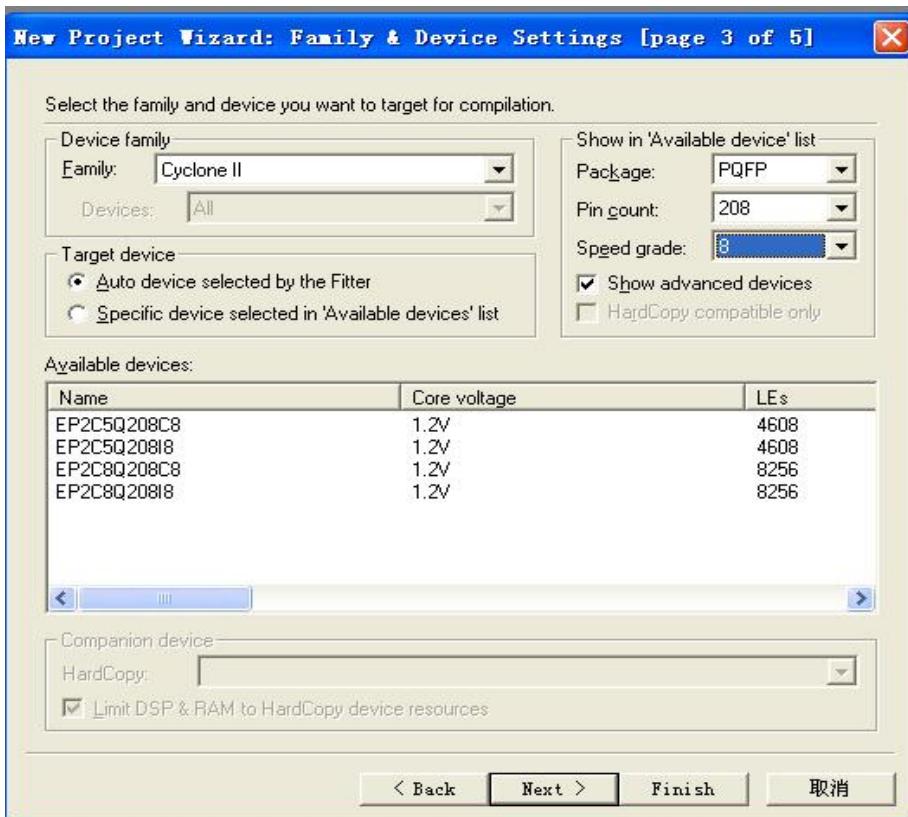
如下图：选择速度等级，我们这里选择8



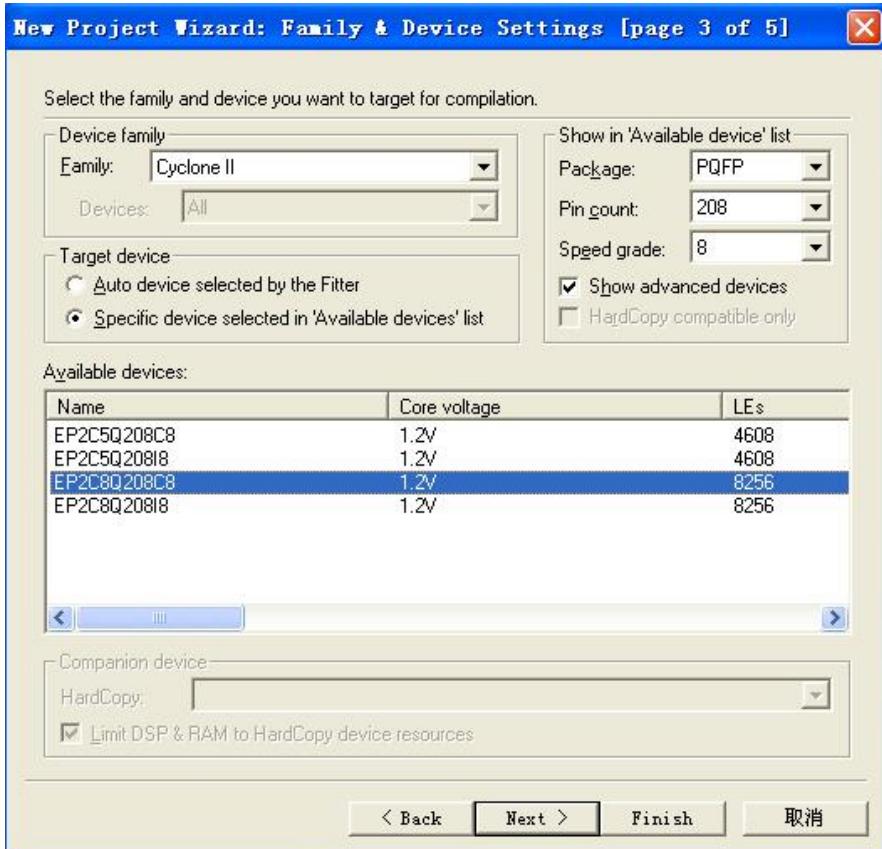
如下图：



Email:906606596@qq.com



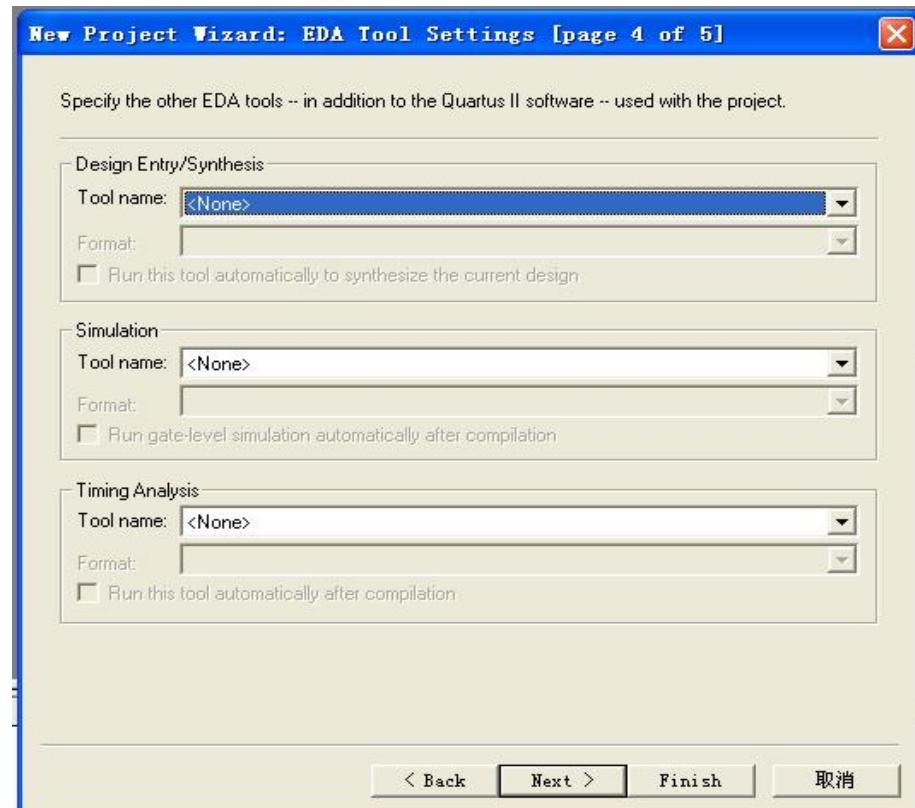
如下图：选择我们最后设置好的芯片。然后点击Next



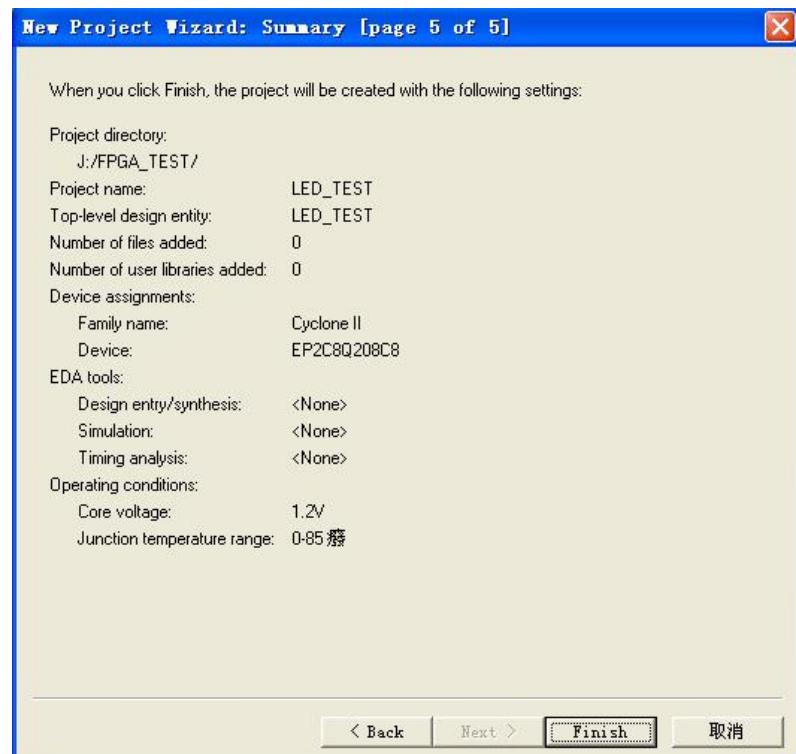
如下图：然后点击Next



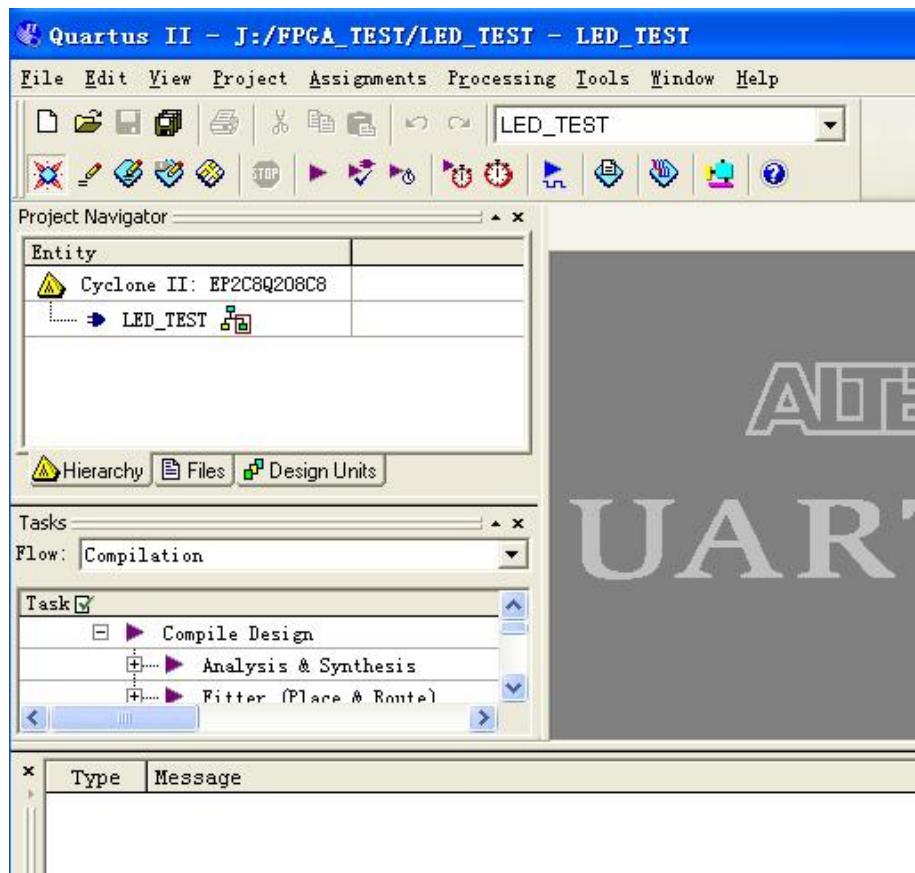
Email:906606596@qq.com



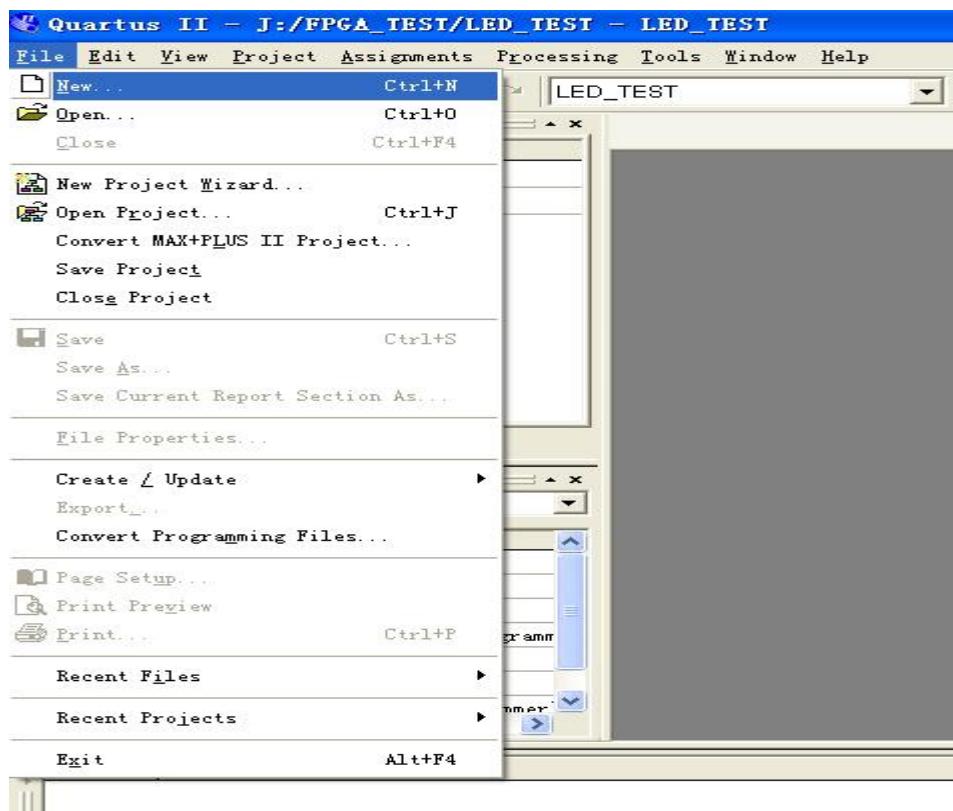
如下图：然后点击Finish



Email:906606596@qq.com



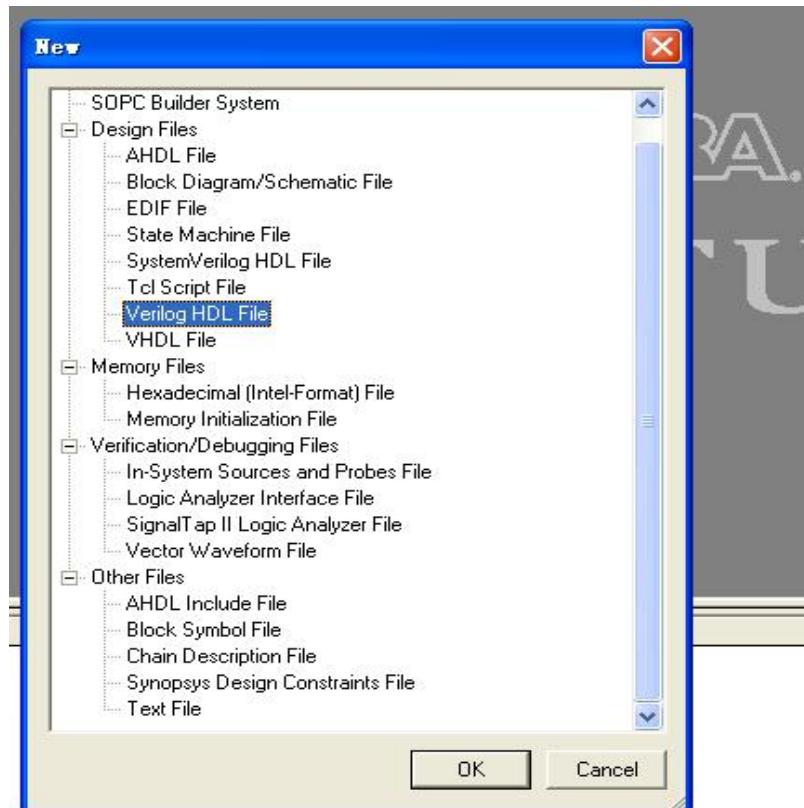
如下图：File->New



如下图：If the Verilog language you use, select the Verilog HDL File, and then click OK

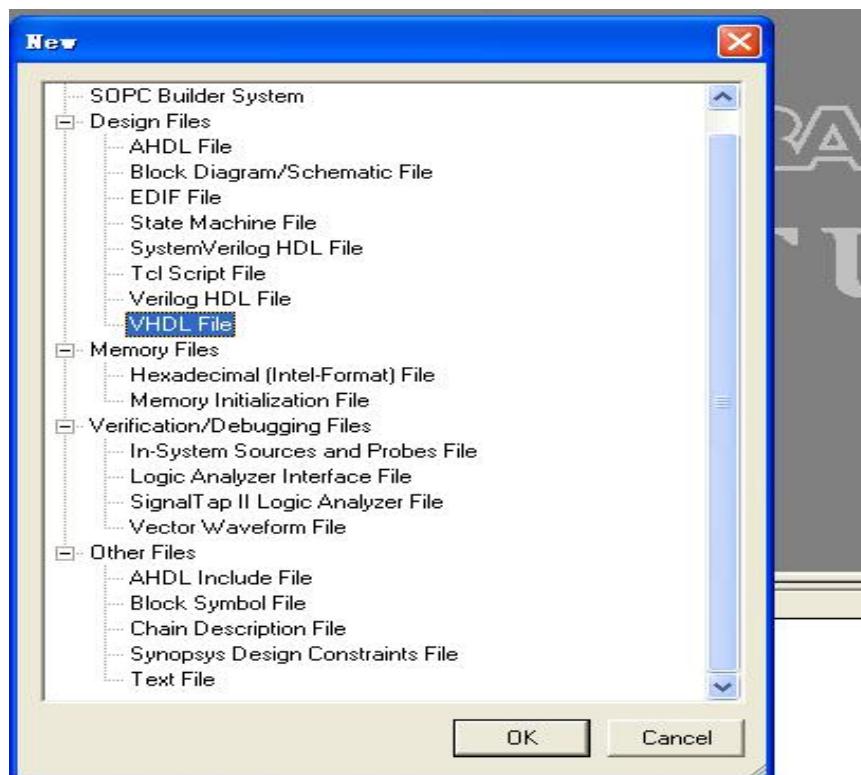


Email:906606596@qq.com

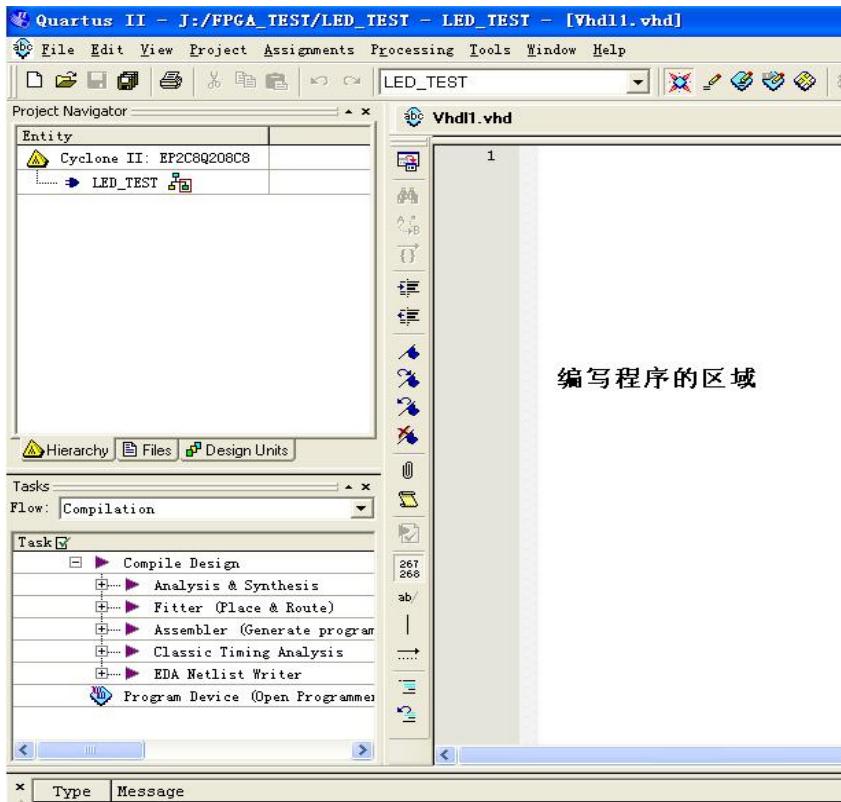


The following diagram: if VHDL language you use, select VHDL File and click OK

Here we use VHDL as a demonstration



Email:906606596@qq.com



```
1 --深圳市21EDA电子
2 --开发板型号: A-C8V4
3 --www.21eda.com
4 --学习LED控制。LED灯对应的脚为0时，点亮
5 --视频教程适合我们21EDA电子的所有学习者
6
7 LIBRARY IEEE;
8 USE IEEE.STD_LOGIC_1164.ALL;
9 USE IEEE.std_logic_unsigned.ALL;
10 ENTITY LEDA IS
11 PORT(
12     clk:in STD_LOGIC; --System clock
13     led1:out STD_LOGIC_VECTOR(7);
14 );
15 END LEDA;
16 ARCHITECTURE light OF LEDA IS
17 SIGNAL clk1,CLK2:std_logic;
18 BEGIN
19 P1:PROCESS (clk)
20 VARIABLE count:INTEGER RANGE 0 TO 999;
21 BEGIN
22 IF clk'EVENT AND clk='1' THEN
23 IF count<=4999999 THEN
24     clk1<='0';
25     count:=count+1;
26 ELSIF count>=4999999 AND count<=5000000 THEN
27     clk1<='1';
28     count:=count+1;
29 ELSE count:=0;
30 END IF;
31 END IF;
32 END PROCESS P1;
33 CLK2<=clk1;
34 END light;
```

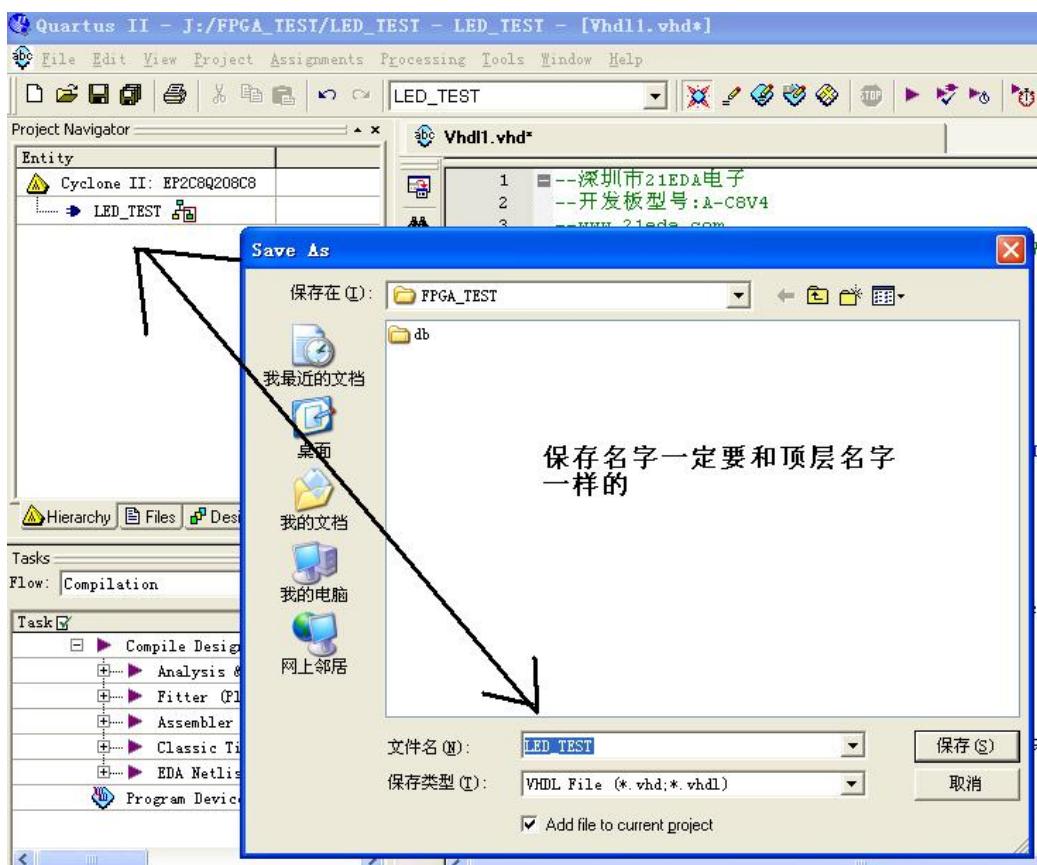
如下图：我们的程序编写了，要保存 File -> Save



The screenshot shows the Quartus II interface with the project 'LED_TEST' open. The left pane displays the 'File' menu, which includes options like 'New...', 'Open...', 'Save', and 'Exit'. The right pane shows the VHDL source code for the entity 'LEDA'. The code is as follows:

```
1 --深圳市21EDA电子
2 --开发板型号:A-
3 --www.21eda.com
4 --学习LED控制。
5 --视频教程适合初
6
7 LIBRARY IEEE;
8 USE IEEE.STD_LOGIC_1164.all;
9 USE IEEE.STD.TEXTIO.all;
10 ENTITY LEDA IS
11     PORT(
12         clk: IN STD_LOGIC;
13         led1: OUT STD_LOGIC);
14
15 END LEDA;
16 ARCHITECTURE behavioral OF LEDA IS
17     SIGNAL clk1, CLK: STD_LOGIC;
18 BEGIN
19     P1: PROCESS (clk)
20     VARIABLE count: INTEGER;
21     BEGIN
22         IF clk'EVENT AND clk='1' THEN
23             IF count = 1000000 THEN
24                 clk1<= NOT clk1;
25             COUNT := 0;
26         ELSIF count > 1000000 THEN
27             COUNT := 0;
28         ELSE
29             COUNT := COUNT + 1;
30         END IF;
31     END IF;
32     END PROCESS P1;
33     clk1<= clk;
34     LED1 <= clk1;
35 END behavioral;
```

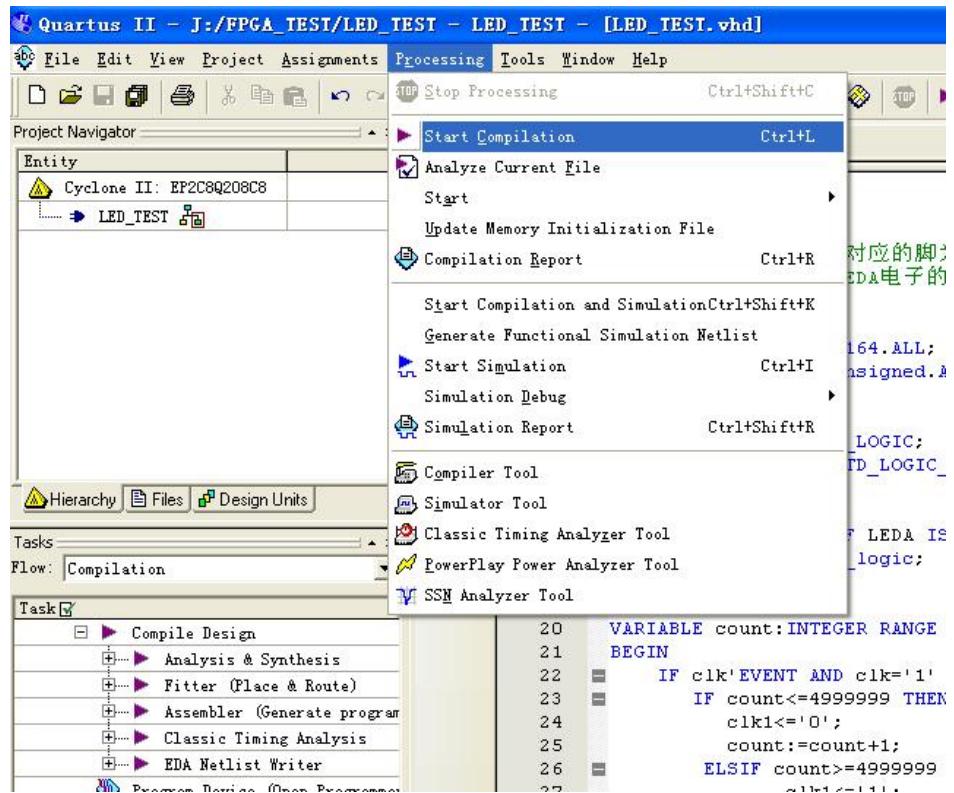
如下图：



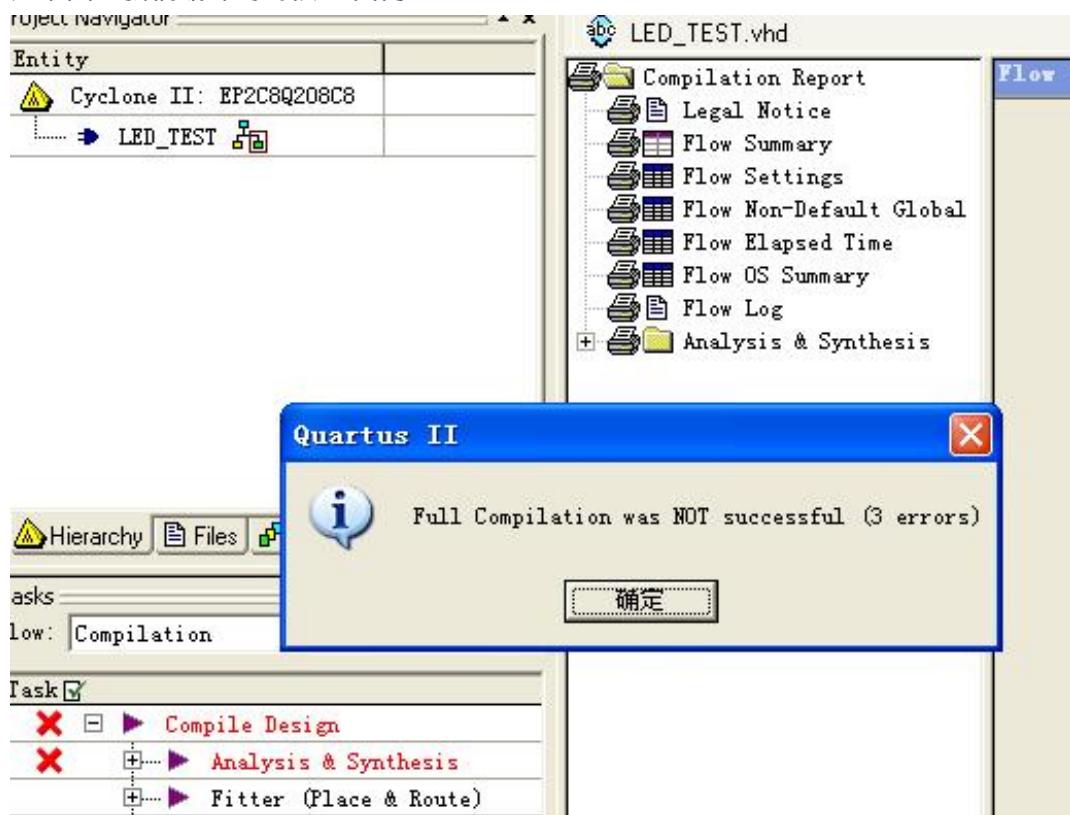
如下图：保存了之后，就要编译，



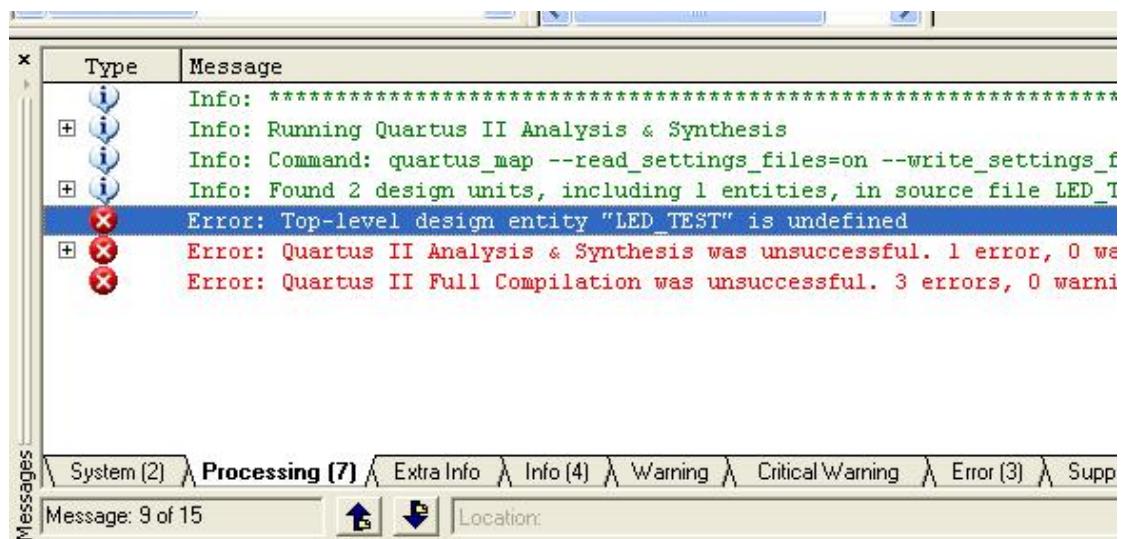
Email:906606596@qq.com



如下图：我们编译的时候 出错了。



如下图：说明顶层名和实体名字不一样，这个是初学者容易忘记的东西，这要注意哦



下面我们来修改。

如下图：实体名不对，高亮显示的

```
6
7 LIBRARY IEEE;
8 USE IEEE.STD_LOGIC_1164.ALL;
9 USE IEEE.std_logic_unsigned.ALL;
10 ENTITY LEDA is
11   PORT(
12     clk:in STD_LOGIC;  --System Clock
13     led1:out STD_LOGIC_VECTOR(7 DOWNTO 0);
14
15   END LEDA ;
16   ARCHITECTURE light OF LEDA IS
17     SIGNAL clk1,CLK2:std_logic;
18   BEGIN
19   PROCESS (clk)
20     VARIABLE count:INTEGER RANGE 0 TO 999999;
21     BEGIN
```

如下图：实体名我们改成LED_TEST，高亮显示的

```
5 --视频教程适合我们21EDA电子的所有朋友
6
7 LIBRARY IEEE;
8 USE IEEE.STD_LOGIC_1164.ALL;
9 USE IEEE.std_logic_unsigned.ALL;
10 ENTITY LED_TEST is
11   PORT(
12     clk:in STD_LOGIC;  --System Clock
13     led1:out STD_LOGIC_VECTOR(7 DOWNTO 0);
14
15   END LEDA ;
16   ARCHITECTURE light OF LEDA IS
17     SIGNAL clk1,CLK2:std_logic;
18   BEGIN
```



如下图：实体名不对，高亮显示的

```

      led1:out STD_LOGIC_VECTOR('7 DOWNTO 0)); --LED output是
--我们这里实际
5   END LEDA ;
6   ARCHITECTURE light OF LEDA IS
7     SIGNAL clk1,CLK2:std_logic;
8   BEGIN
9     P1:PROCESS (clk)
10    VARIABLE count:INTEGER RANGE 0 TO 9999999;
11    BEGIN
12      IF clk'EVENT AND clk='1' THEN --当时
13        IF count<=4999999 THEN
14          clk1<='0';
15          count:=count+1; --当
16        ELSIF count>=4999999 AND count<=9999999 THEN
17          clk1<='1';

```

如下图：实体名我们改成LED TEST，高亮显示的

```

12           clk:in STD_LOGIC; --System Clk
13           led1:out STD_LOGIC_VECTOR(7 DOWNTO 0));
14
15      END LED_TEST ;
16
17  ■ ARCHITECTURE light OF LEDA IS
18      SIGNAL clk1,CLK2:std_logic;
19
20  ■ BEGIN
21      ■ P1:PROCESS (clk)
22          VARIABLE count:INTEGER RANGE 0 TO 9999999;
23          BEGIN
24              ■ IF clk'EVENT AND clk='1' THEN
25                  ■ IF count<=4999999 THEN
26                      clk1<='0';
27                      count:=count+1;
28                  ■ ELSIF count>=4999999 AND count<=9999999 THEN
29

```

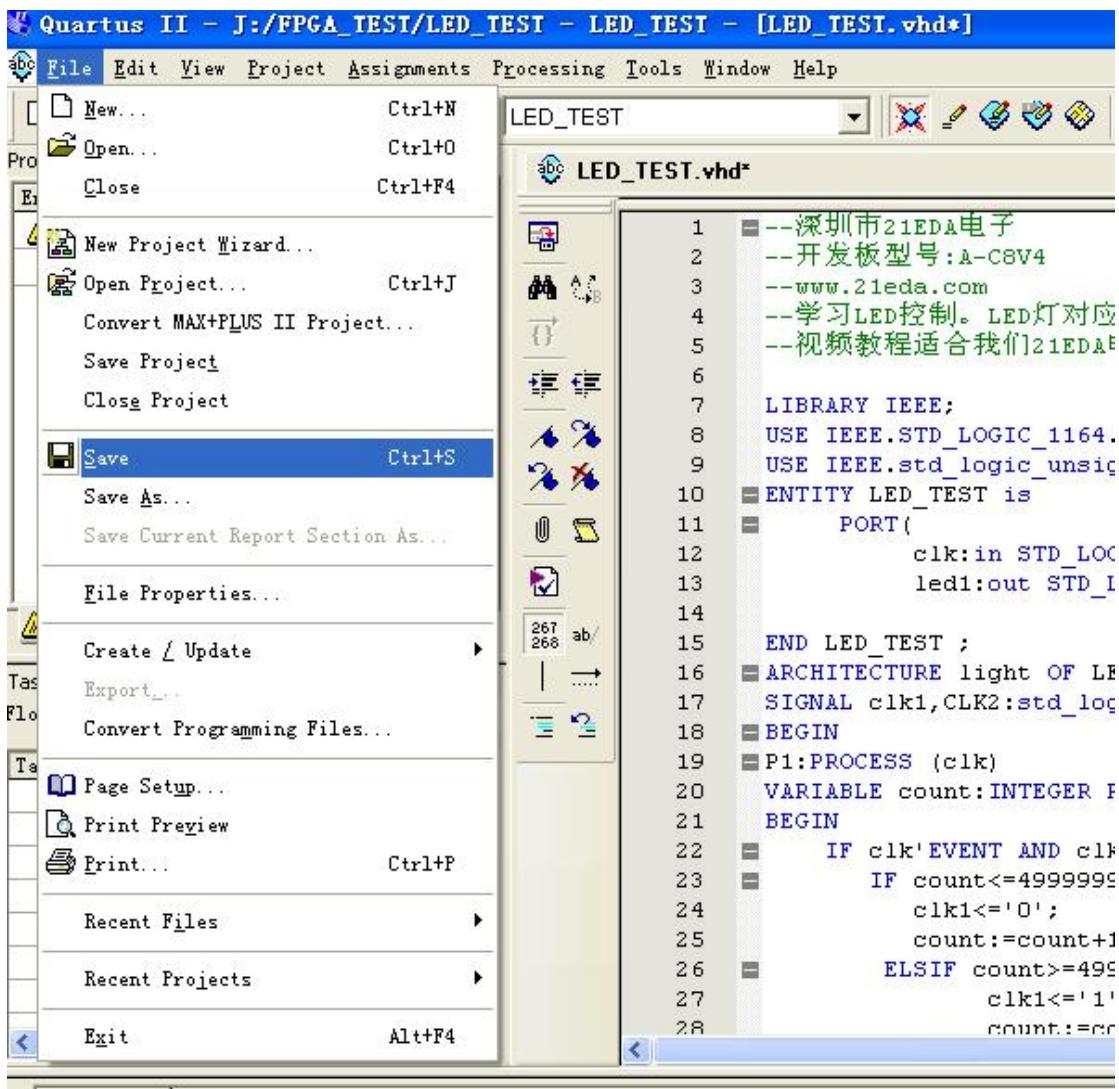
如下图：还有结构体那里，高亮显示的



Email:906606596@qq.com

```
END LED_TEST ;
ARCHITECTURE light OF LED_TEST IS
SIGNAL clk1,CLK2:std_logic;
BEGIN
P1:PROCESS (clk)
VARIABLE count:INTEGER RANGE 0 TO 9999999;
BEGIN
IF clk'EVENT AND clk='1' THEN
IF count<=4999999 THEN
clk1<='0';
count:=count+1;
ELSIF count>=4999999 AND count<=9999999 THEN
clk1<='1';
--当
--当
--
--
--
```

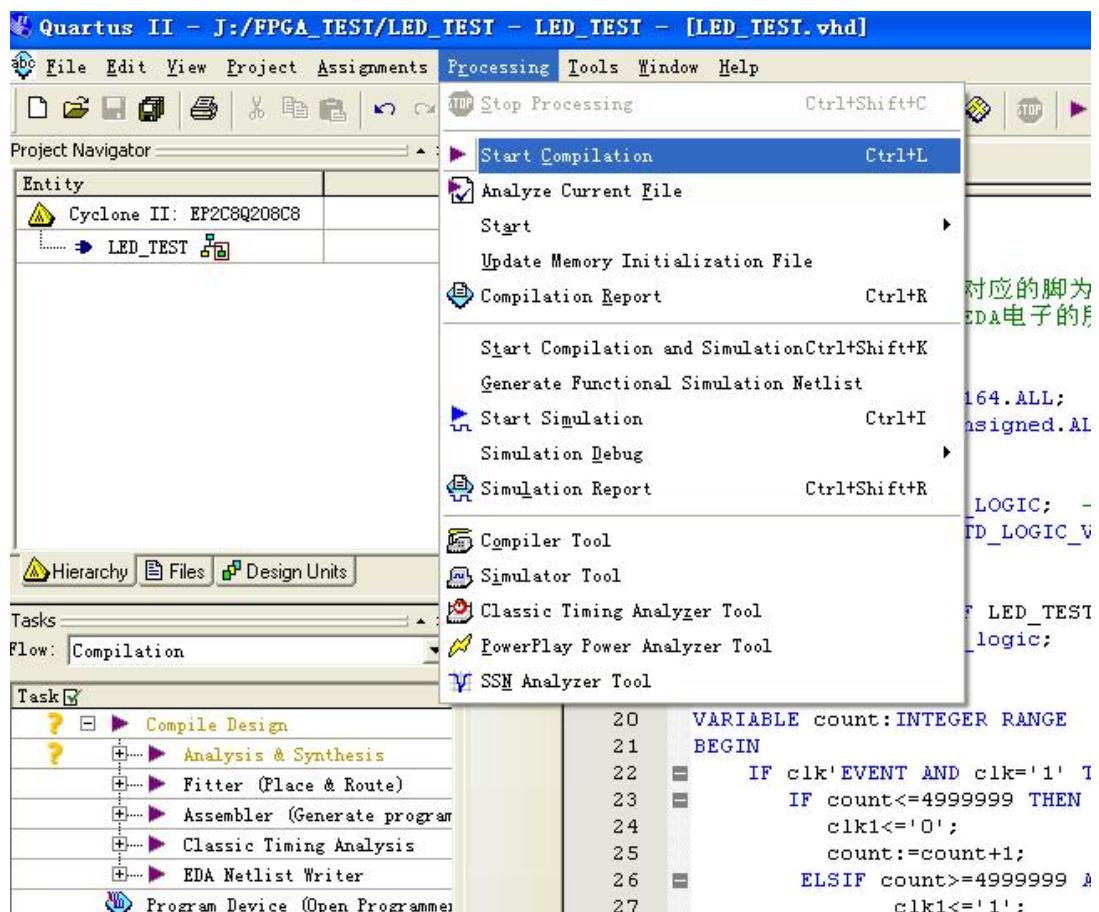
好了，我们该了之后需要保存 如下图：



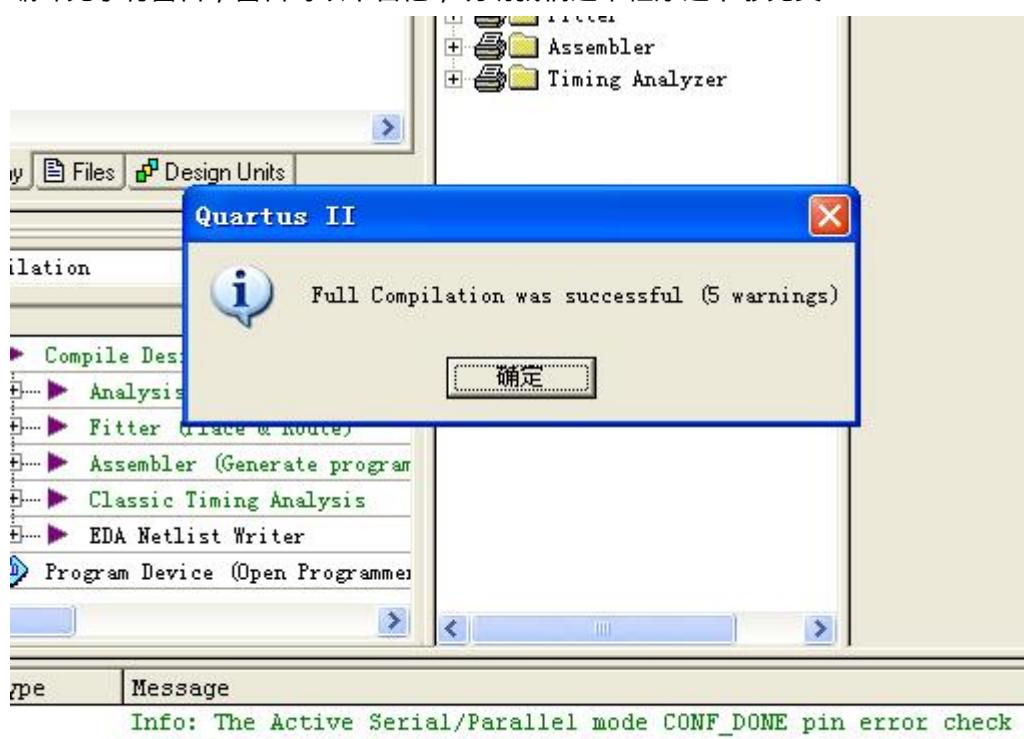
保存了之后，还要编译一次，如下图



Email:906606596@qq.com



编译完了有警告，警告可以不管他，说明我们这个程序还不够完美。

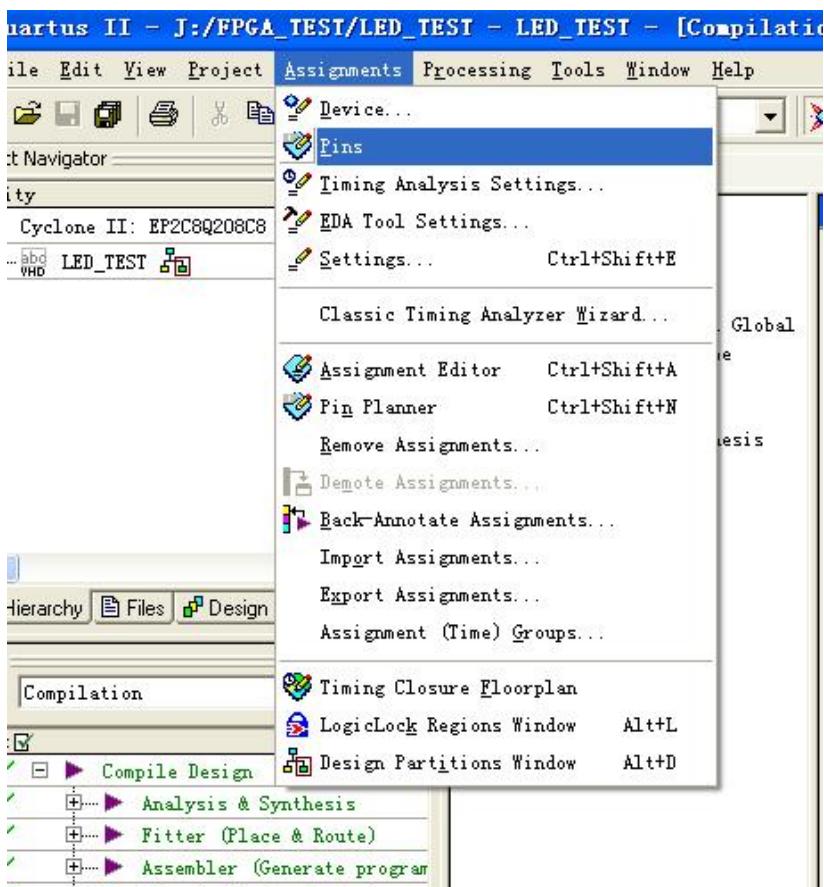


Email:906606596@qq.com

如下图：是整个程序用的资源。

Flow Status	Successful - Mon Jul 18 22:39:28 2011
Quartus II Version	9.0 Build 235 06/17/2009 SP 2 SJ Web Edition
Revision Name	LED_TEST
Top-level Entity Name	LED_TEST
Family	Cyclone II
Device	EP2C8Q208C8
Timing Models	Final
Met timing requirements	Yes
Total logic elements	75 / 8,256 (< 1 %)
Total combinational functions	75 / 8,256 (< 1 %)
Dedicated logic registers	39 / 8,256 (< 1 %)
Total registers	39
Total pins	9 / 138 (7 %)
Total virtual pins	0
Total memory bits	0 / 165,888 (0 %)
Embedded Multiplier 9-bit elements	0 / 36 (0 %)
Total PLLs	0 / 2 (0 %)

如下图：现在我们来分配I/O口

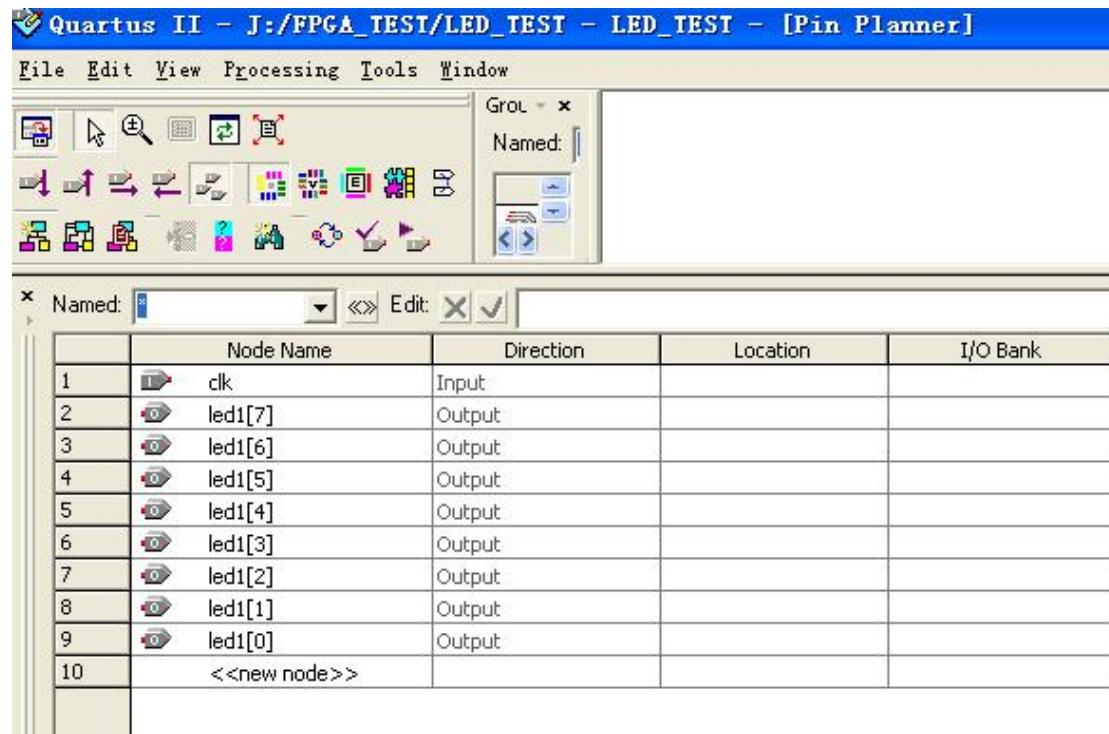


如下图：这里是我们还没有分配的时候

如果你对I/O的分配对应关系还不熟悉，可以看一下I/O的对应PDF文件。



Email:906606596@qq.com



如下图：这个我们不用管他



I/O Standard	Reserved	Group	Current Strength
3.3-V LVTTL (default)			24mA (default)
3.3-V LVTTL (default)		led1[7..0]	24mA (default)
3.3-V LVTTL (default)		led1[7..0]	24mA (default)
3.3-V LVTTL (default)		led1[7..0]	24mA (default)
3.3-V LVTTL (default)		led1[7..0]	24mA (default)
3.3-V LVTTL (default)		led1[7..0]	24mA (default)
3.3-V LVTTL (default)		led1[7..0]	24mA (default)
3.3-V LVTTL (default)		led1[7..0]	24mA (default)
3.3-V LVTTL (default)		led1[7..0]	24mA (default)

如下图：我们这里的时钟是分配的23脚。 (50MHz输入)



Email:906606596@qq.com

	Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Re:
1	clk	Input				3.3-V LVTTL (default)	
2	led1[7]	Output	PIN_12	IOBANK_1	Row I/O	LVDS12n	
3	led1[6]	Output	PIN_13	IOBANK_1	Row I/O	VREFB1N0	
4	led1[5]	Output	PIN_14	IOBANK_1	Row I/O	LVDS8p, DPLK0/DQS0L/CQ1L	
5	led1[4]	Output	PIN_15	IOBANK_1	Row I/O	LVDS8n	
6	led1[3]	Output	PIN_23	IOBANK_1	Dedicated Clock	CLK0, LVDSCLK0p, Input	
7	led1[2]	Output	PIN_24	IOBANK_1	Dedicated Clock	CLK1, LVDSCLK0n, Input	
8	led1[1]	Output	PIN_27	IOBANK_1	Dedicated Clock	CLK2, LVDSCLK1p, Input	
9	led1[0]	Output	PIN_28	IOBANK_1	Dedicated Clock	CLK3, LVDSCLK1n, Input	
10	<<new node>>						

	Node Name	Direction	Location
1	clk	Input	PIN_23
2	led1[7]	Output	
3	led1[6]	Output	
4	led1[5]	Output	
5	led1[4]	Output	
6	led1[3]	Output	
7	led1[2]	Output	
8	led1[1]	Output	
9	led1[0]	Output	
10	<<new node>>		

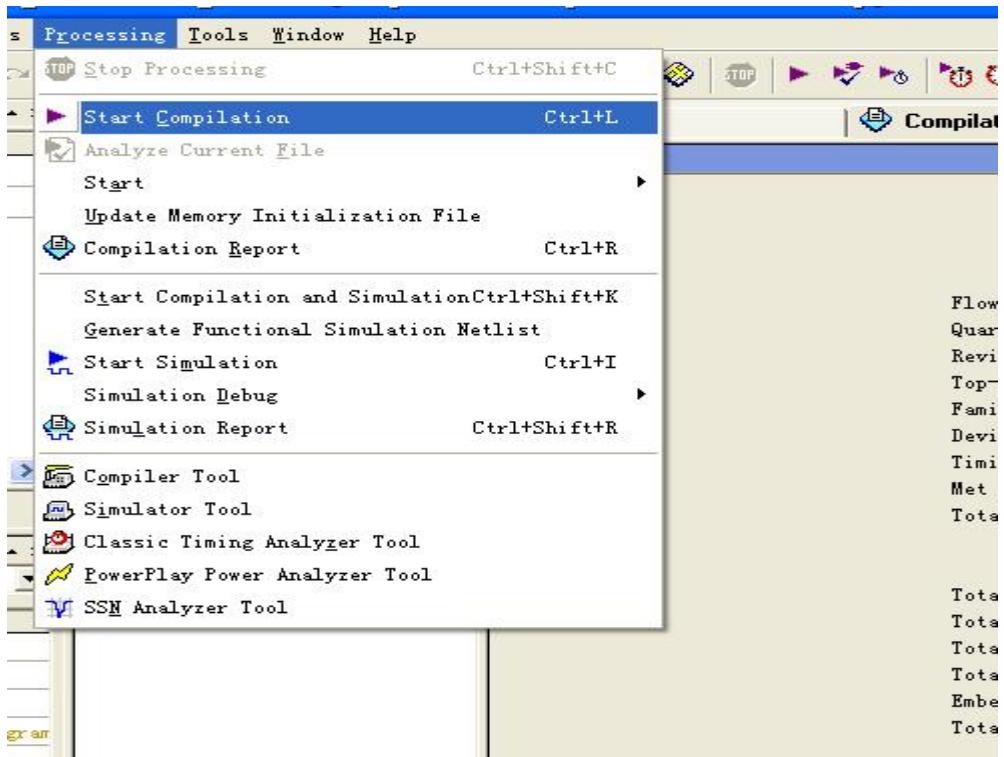
其它发光二极管的分配方法一样的。在location对应的框框里面双击，会出来管脚的。

	Node Name	Direction	Location
1	clk	Input	PIN_23
2	led1[7]	Output	
3	led1[6]	Output	
4	led1[5]	Output	PIN_118
5	led1[4]	Output	PIN_117
6	led1[3]	Output	PIN_116
7	led1[2]	Output	PIN_115
8	led1[1]	Output	PIN_114
9	led1[0]	Output	PIN_113
10	<<new node>>		

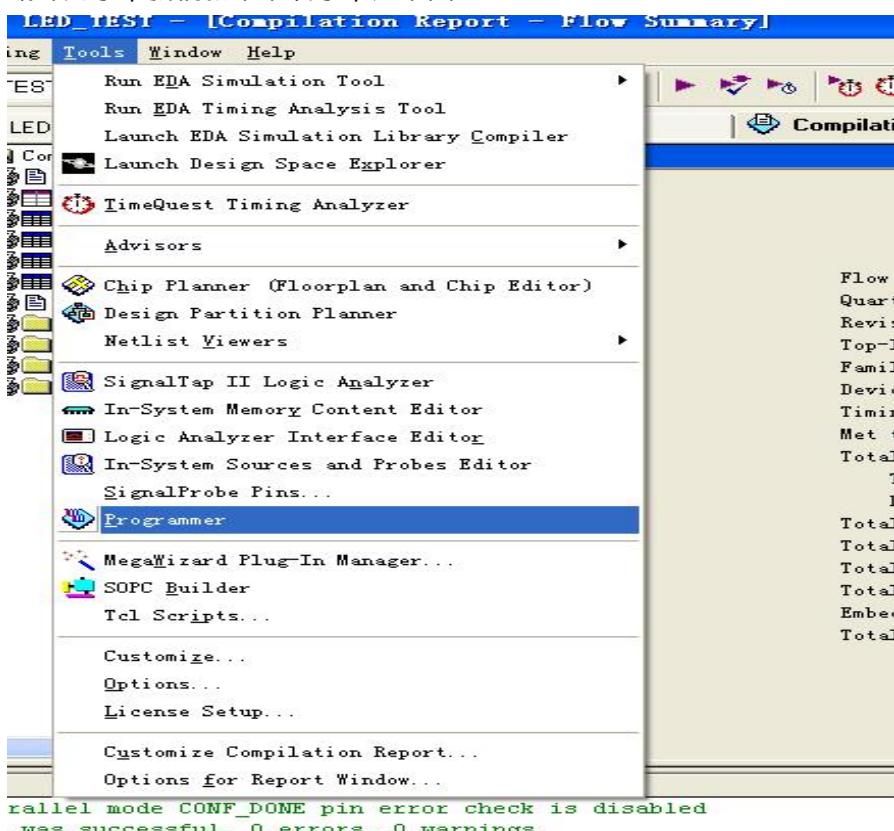
分配好了管脚，就要编译下，如下图：



Email:906606596@qq.com



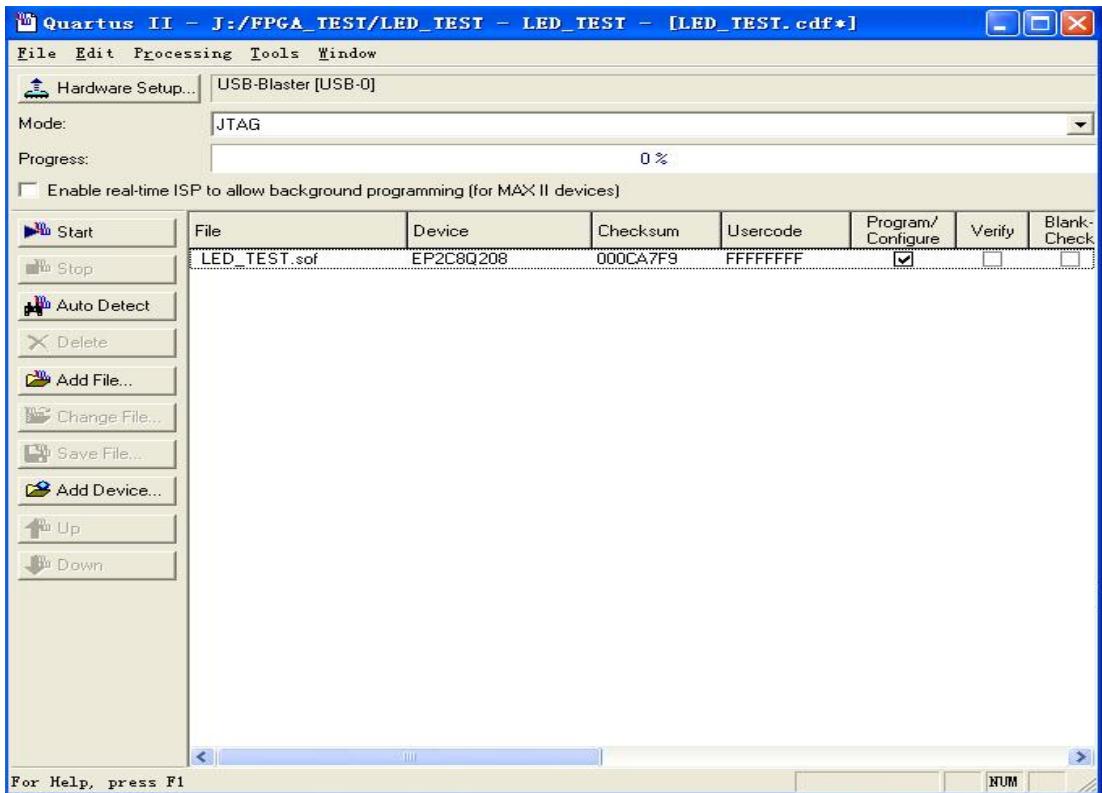
编译完了，我们就来下载了，如下图：



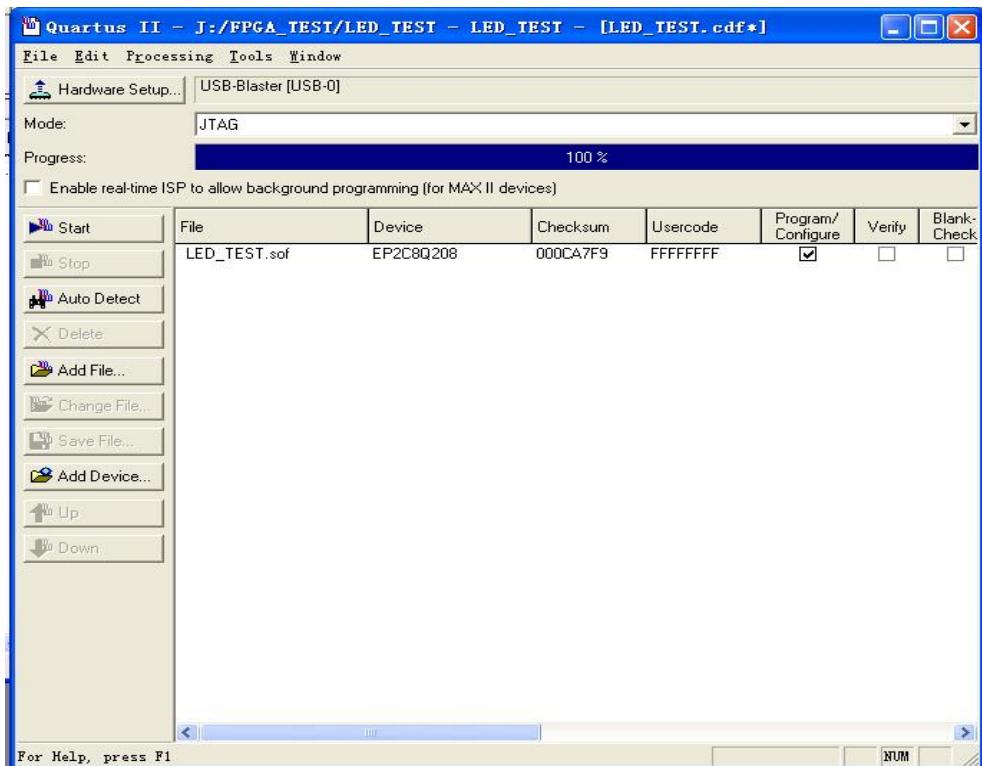
模式选择JTAG模式，我们这里用的USB下载器。当然你用并口的也是可以的。



Email:906606596@qq.com



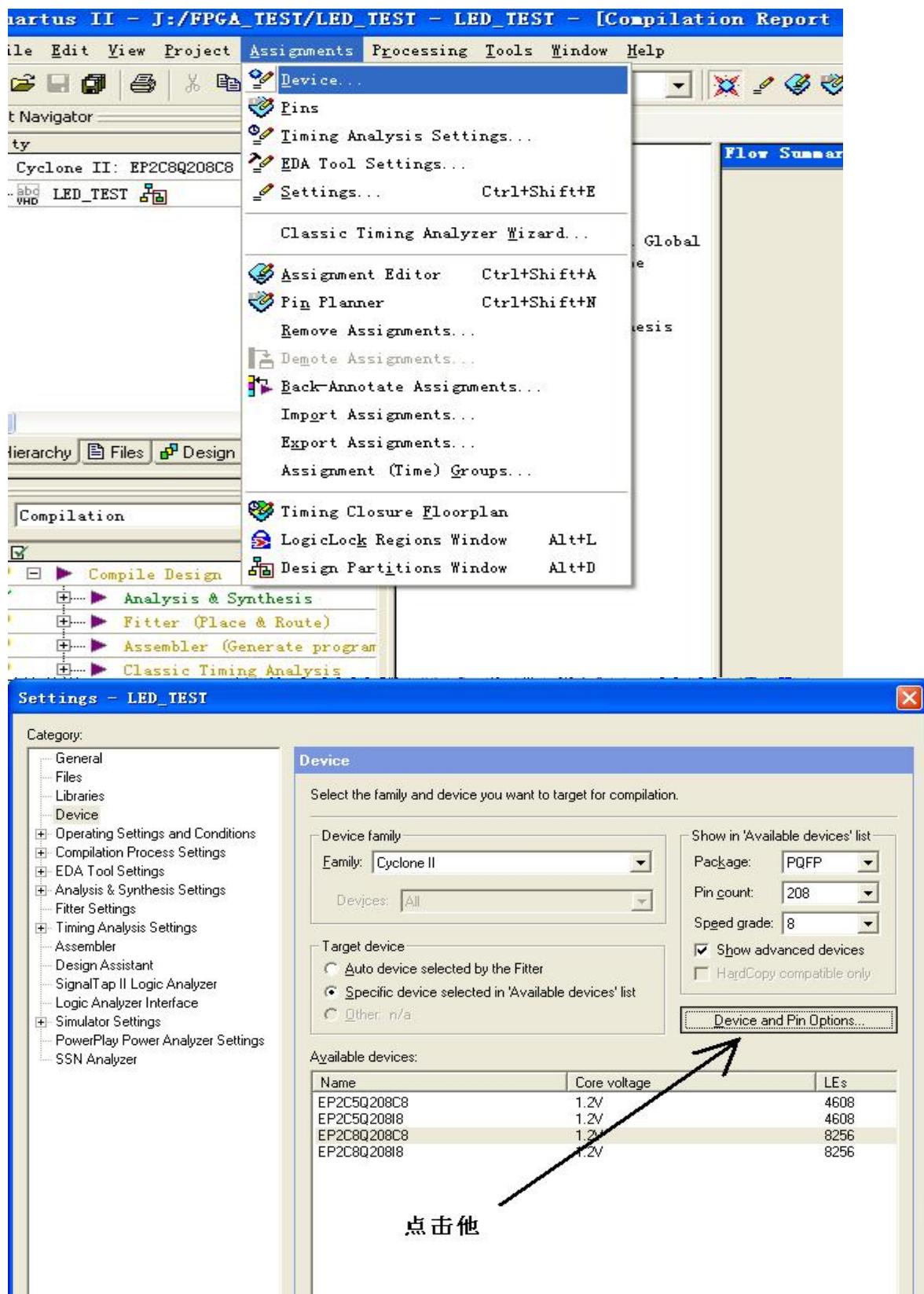
如下图：有的朋友可能会问了。为什么我下载进去蜂鸣器也在叫，数码管全部显示8了



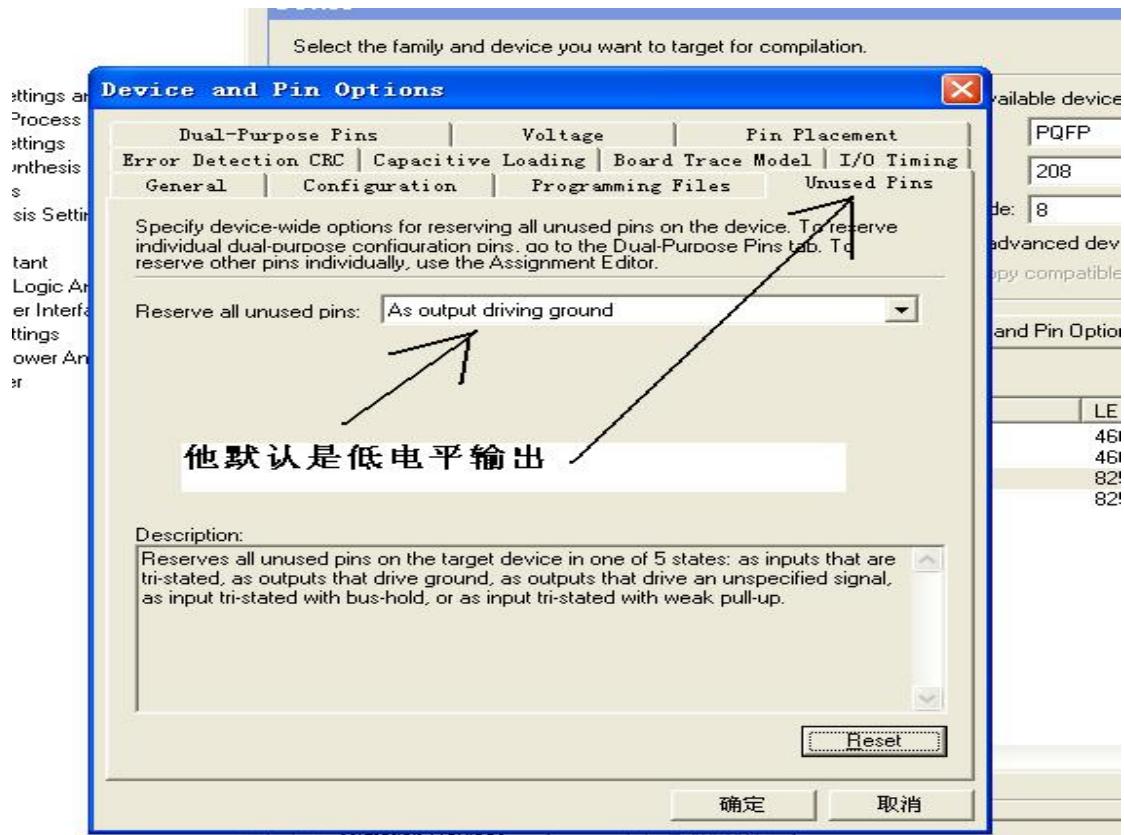
你不要着急，有个地方设置一下。下面一步一步来。



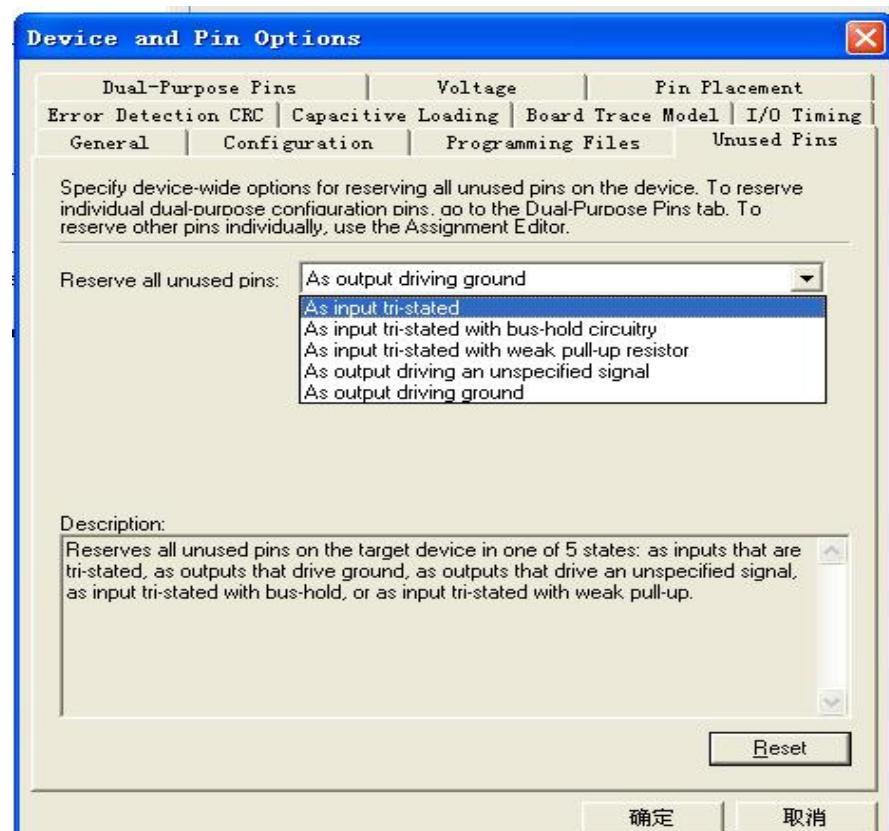
Email:906606596@qq.com



Email:906606596@qq.com



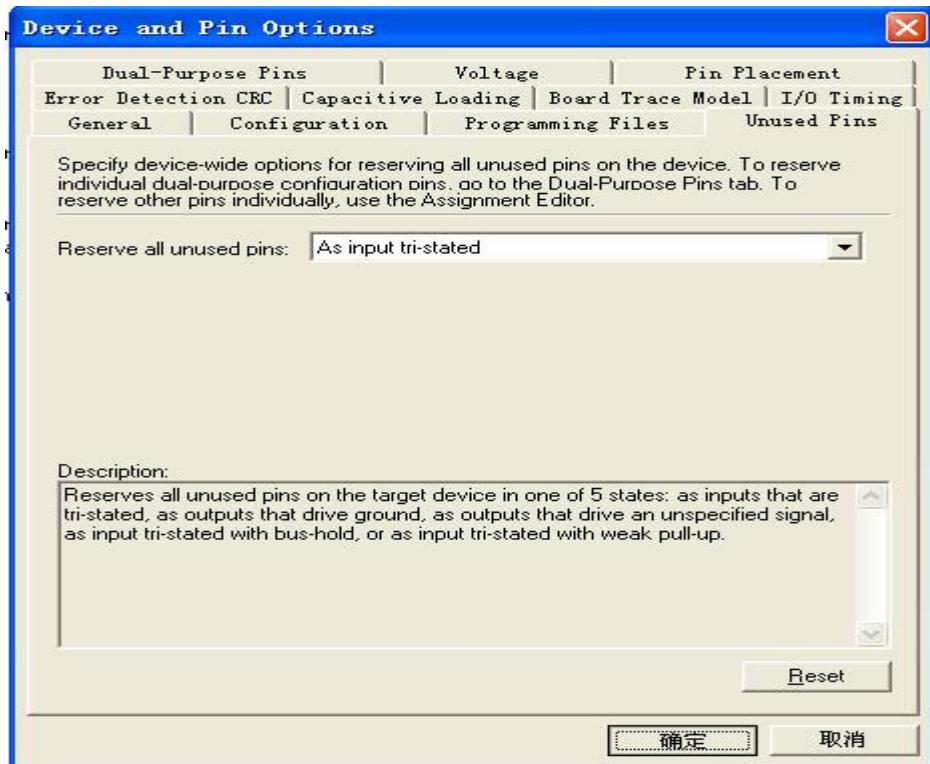
我们要修改一下，如下图，把它设置成三态输入。



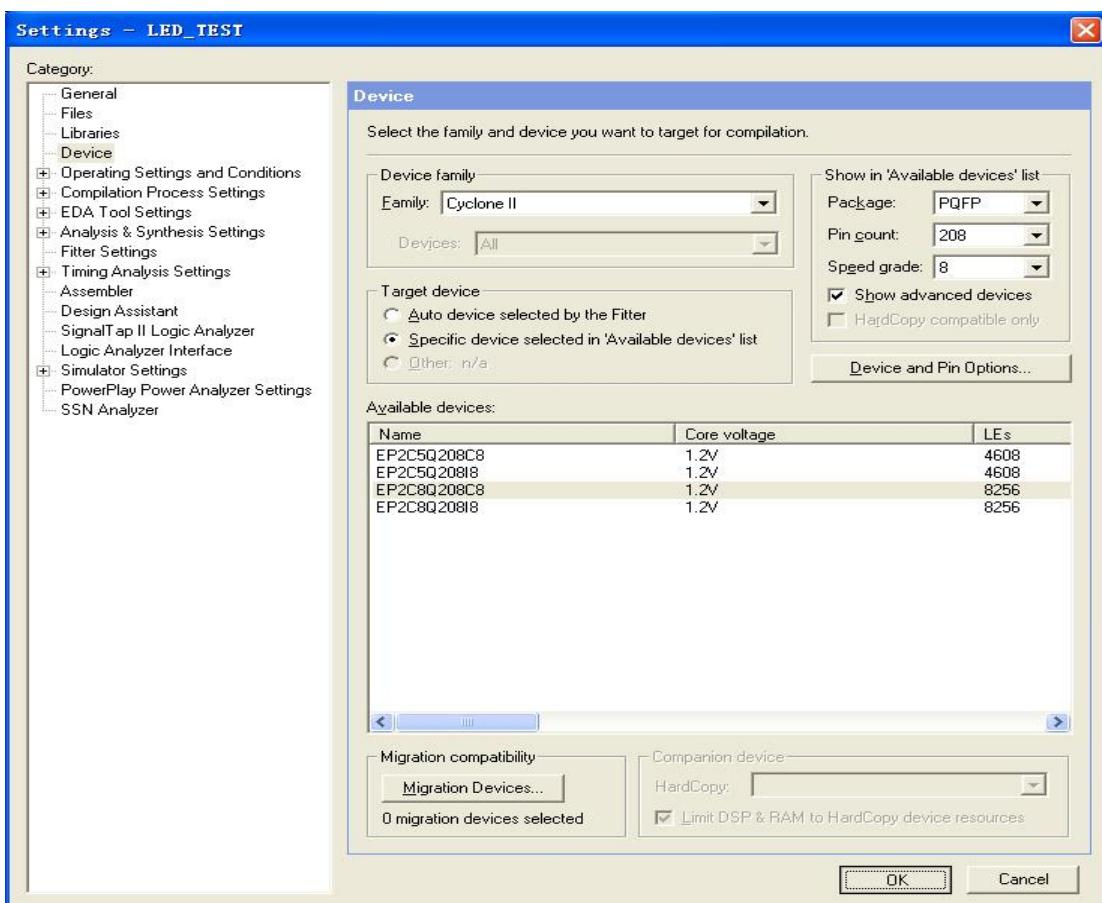
如下图点击确定。



Email:906606596@qq.com



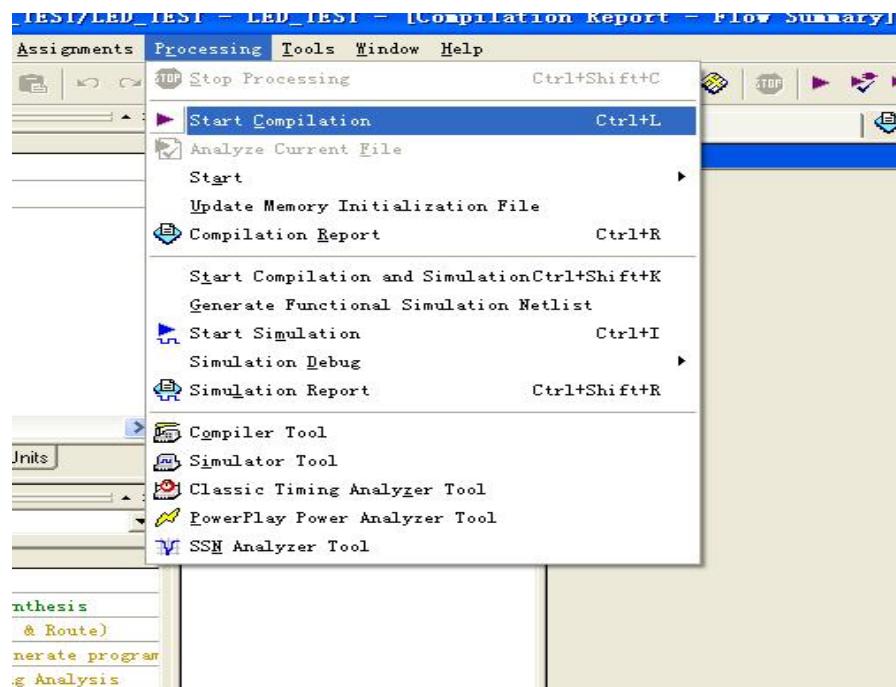
如下图点击确定。



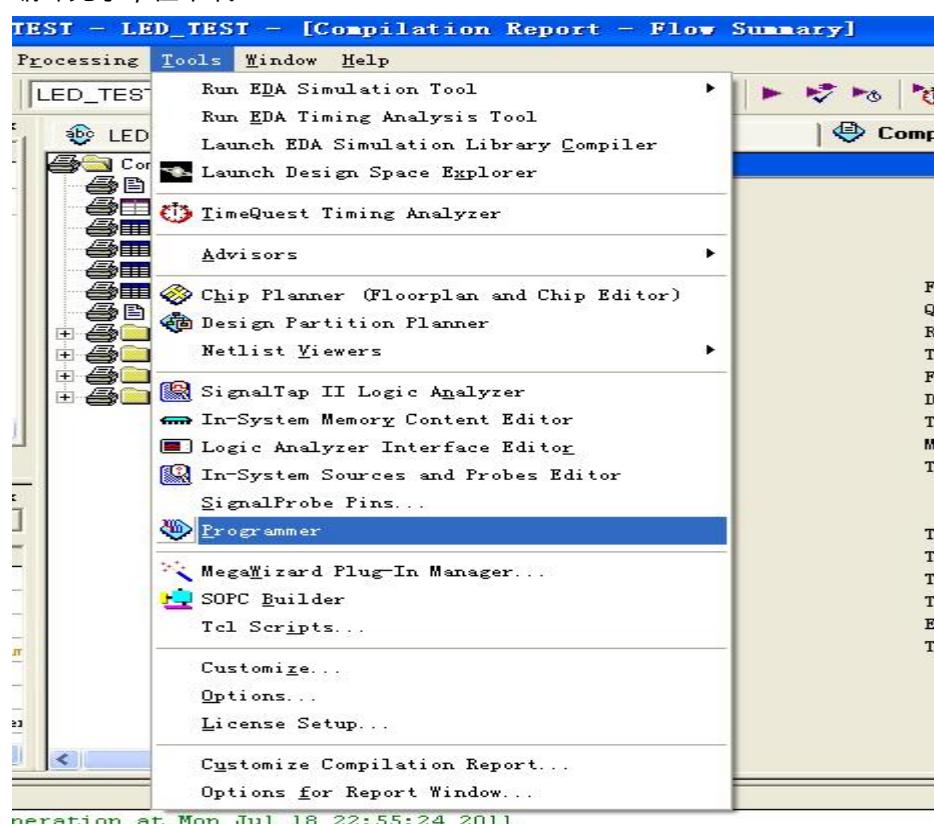
在编译一下 如下图



Email:906606596@qq.com



编译完了，在下载



现在就不会出现蜂鸣器也在叫，数码管全部显示8了

你就可以看到LED灯流水大灯的效果了。



Email:906606596@qq.com

