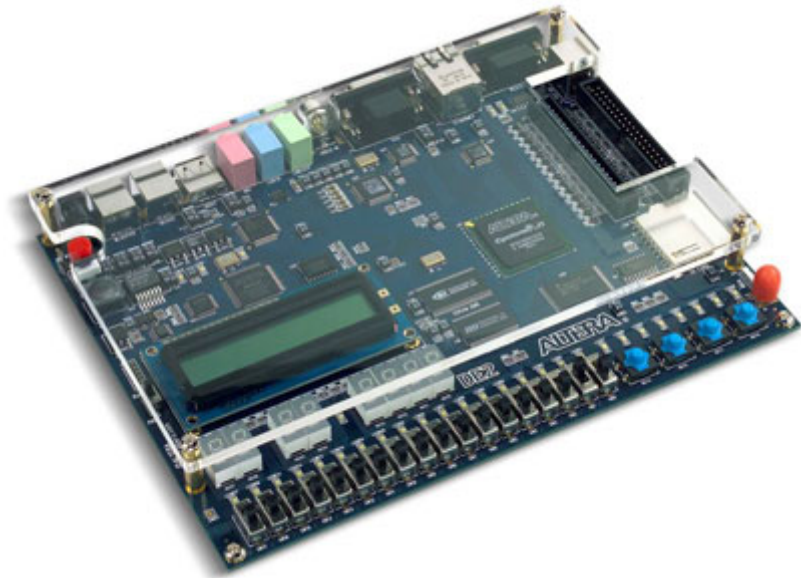


Altera DE2 使用教程



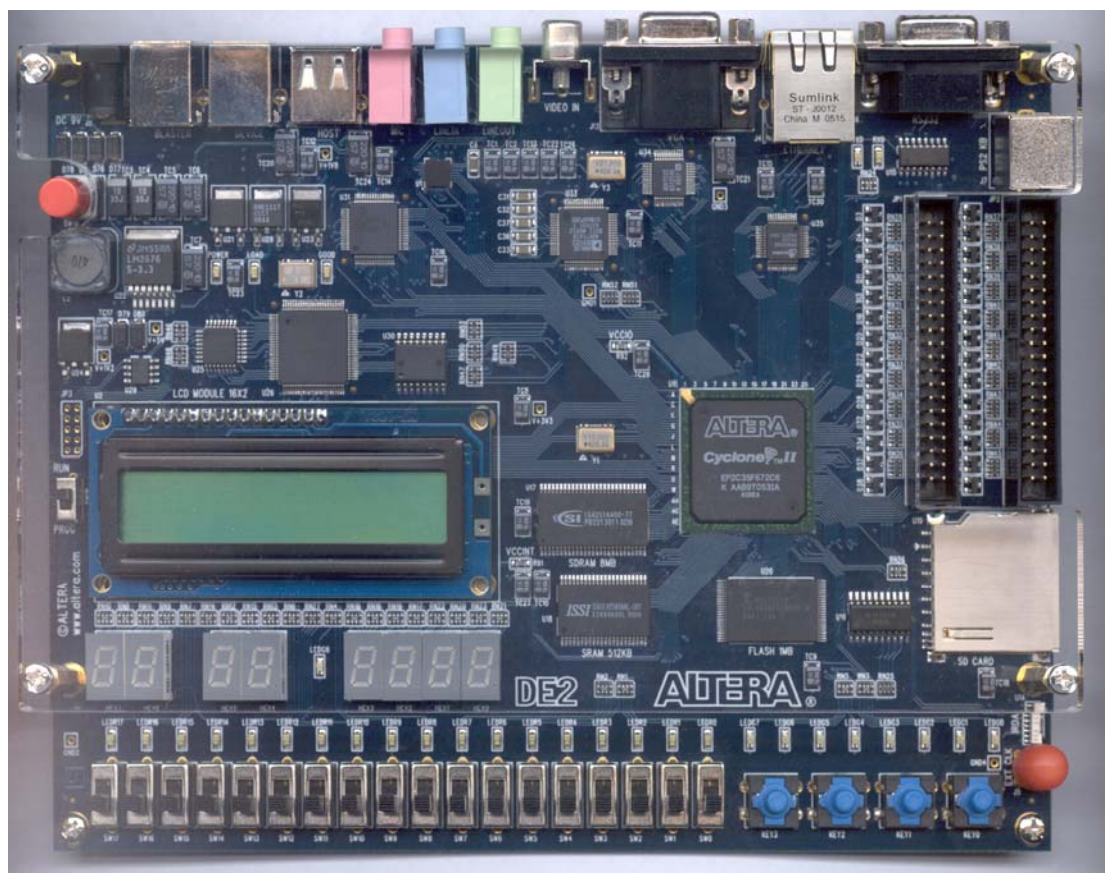
电子信息科学与技术系

目录

DE2 开发板资源:	3
FPGA简要介绍	5
Quaturs使用基础	7
用HDL写代码	22
管脚的输入方式	25
NIOSII 实验	25
FAQ	36

DE2 开发板资源:

<i>Table. DE2 Board Information</i>	
Feature	Description
FPGA	<ul style="list-style-type: none"> Cyclone II EP2C35F672C6 with EPCS16 16-Mbit serial configuration device
I/O Devices	<ul style="list-style-type: none"> Built-in USB-Blaster™ cable for FPGA configuration 10/100 Ethernet RS232 Video out (VGA 10-bit DAC) Video in (NTSC/PAL/multi-format) USB 2.0 (type A and type B) PS/2 mouse or keyboard port Line in/out, microphone in (24-bit Audio CODEC) Expansion headers (76 signal pins) Infrared port
Memory	<ul style="list-style-type: none"> 8-MBytes SDRAM, 512K SRAM, 4-MBytes flash SD memory card slot
Displays	<ul style="list-style-type: none"> 16 x 2 LCD display Eight 7-segment displays
Switches and LEDs	<ul style="list-style-type: none"> 18 toggle switches 18 red LEDs 9 green LEDs Four debounced push-button switches
Clocks	<ul style="list-style-type: none"> 50-MHz crystal for FPGA clock input 27-MHz crystal for video applications External SMA clock input



可以在目前开发板上做的实验包括
基本的 VHDL 以及 Verilog HDL 实验
接口实验
高级实验
NIOS II CPU 实验

开发套件清单

- 1 DE2 开发板一个
- 2 ByteBlasterII 下载电缆 1 根
- 3 5v 电源一个
- 4 光盘一张

FPGA 简要介绍

FPGA 是英文 Field Programmable Gate Array 的缩写,即现场可编程门阵列,它是在 PAL、GAL、EPLD 等可编程器件的基础上进一步发展的产物。它是作为专用集成电路(ASIC)领域中的一种半定制电路而出现的,既解决了定制电路的不足,又克服了原有可编程器件门电路数有限的缺点。FPGA 采用了逻辑单元阵列 LCA (Logic Cell Array) 这样一个新概念,内部包括可配置逻辑模块 CLB (Configurable Logic Block)、输出输入模块 IOB (Input Output Block) 和内部连线 (Interconnect) 三个部分。FPGA 的基本特点主要有: 1) 采用 FPGA 设计 ASIC 电路,用户不需要投片生产,就能得到合用的芯片。 2) FPGA 可做其它全定制或半定制 ASIC 电路的中试样片。 3) FPGA 内部有丰富的触发器和 I / O 引脚。 4) FPGA 是 ASIC 电路中设计周期最短、开发费用最低、风险最小的器件之一。 5) FPGA 采用高速 CHMOS 工艺,功耗低,可以与 CMOS、TTL 电平兼容。可以说, FPGA 芯片是小批量系统提高系统集成度、可靠性的最佳选择之一。

目前 FPGA 的品种很多,有 XILINX 的 XC 系列、TI 公司的 TPC 系列、ALTERA 公司的 FIEX 系列等。FPGA 是由存放在片内 RAM 中的程序来设置其工作状态的,因此,工作时需要对片内的 RAM 进行编程。用户可以根据不同的配置模式,采用不同的编程方式。加电时, FPGA 芯片将 EPROM 中数据读入片内编程 RAM 中,配置完成后, FPGA 进入工作状态。掉电后, FPGA 恢复成白片,内部逻辑关系消失,因此, FPGA 能够反复使用。FPGA 的编程无须专用的 FPGA 编程器,只须用通用的 EPROM、PROM 编程器即可。当需要修改 FPGA 功能时,只需换一片 EPROM 即可。这样,同一片 FPGA,不同的编程数据,可以产生不同的电路功能。因此, FPGA 的使用非常灵活。FPGA 有多种配置模式:并行主模式为一片 FPGA 加一片 EPROM 的方式;主从模式可以支持一片 PROM 编程多片 FPGA;串行模式可以采用串行 PROM 编程 FPGA;外设模式可以将 FPGA 作为微处理器的外设,由微处理器对其编程。

使用本学习板涉及的知识有:

- 1 数字电子技术(原理图方式)
- 2 VHDL 或者 verilog HDL 语言
- 3 Quartus 使用
- 4 NIOS II IDE 使用
- 5 SOPC builder

从容易操作和上手的角度出发,我们将采用简单的例子来学习这些工具的使用,目的是使学习着容易成功,增强信心.许多 FPGA 开发板做了很复杂,但对入门者来说不适合.因为复杂的功能很容易引起实验失败,过多的失败会引起初学者丧失兴趣.

在开始实验前,先作好如下准备:

- 安装好 Quartus6.0
- 安装好 NIOS II development kit 6.0
- 安装好 license

安装软件的版本一定要一致,否则会有问题,例如做 NIOSII 实验的时候,项目编译出错
注意:要确保安装的软件可以用,特别是 `license`,否则可能在生成 NIOS II 项目的时候 POF 文件不存在,无法下载到配置芯片.最好使用默认的安装路径.

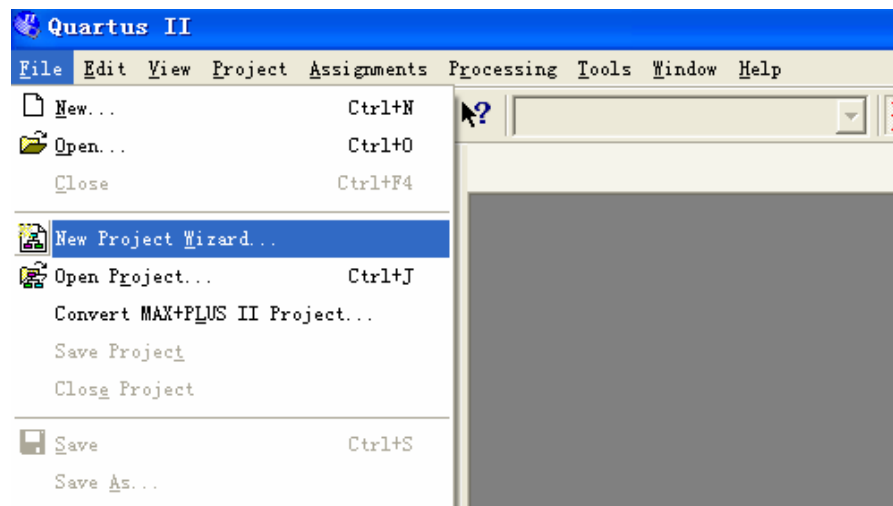
Quartus 使用基础

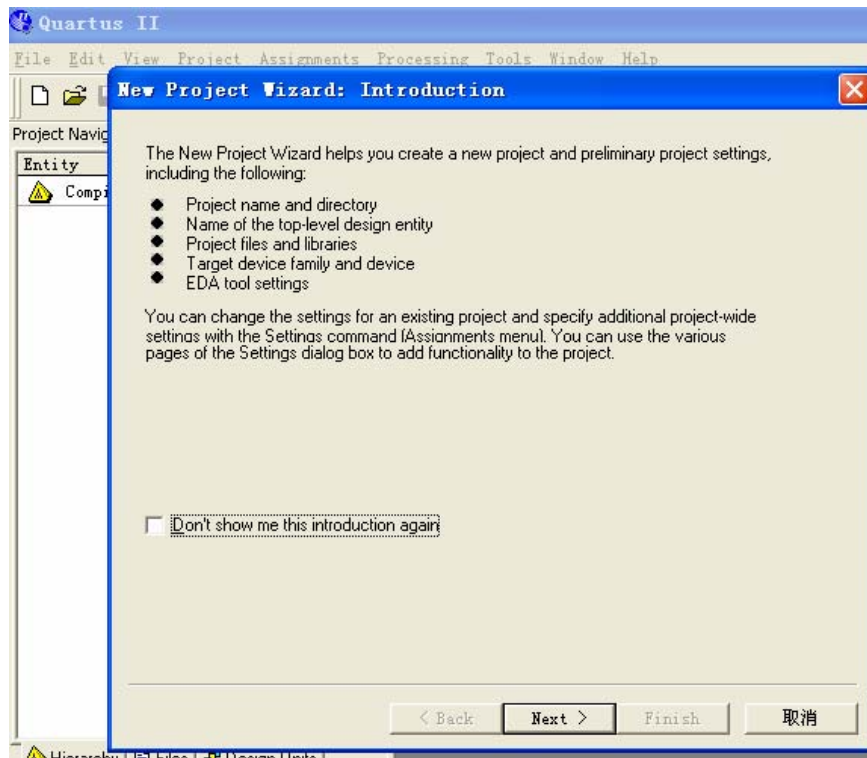
步骤:

- 1 编写代码
- 2 分配管脚
- 3 编译
- 4 下载
- 5 断电后重新上电,系统运行

本实验通过按键 SW0 控制 LEDR0 他们之间是反相关系。

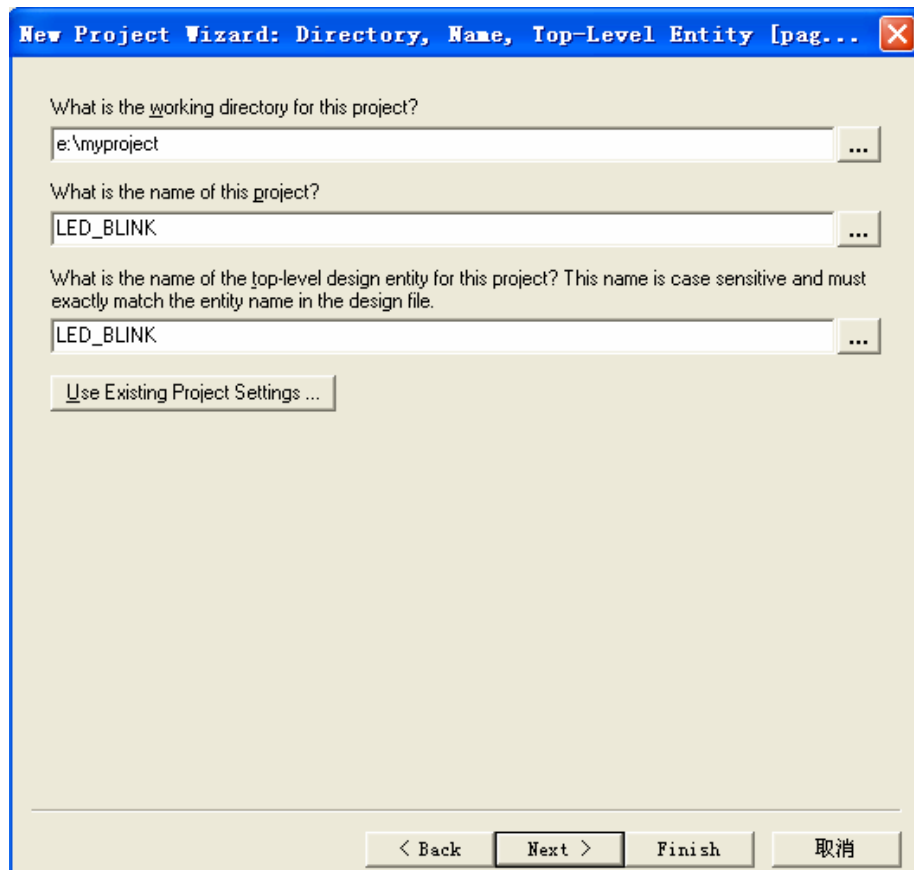
按如下操作



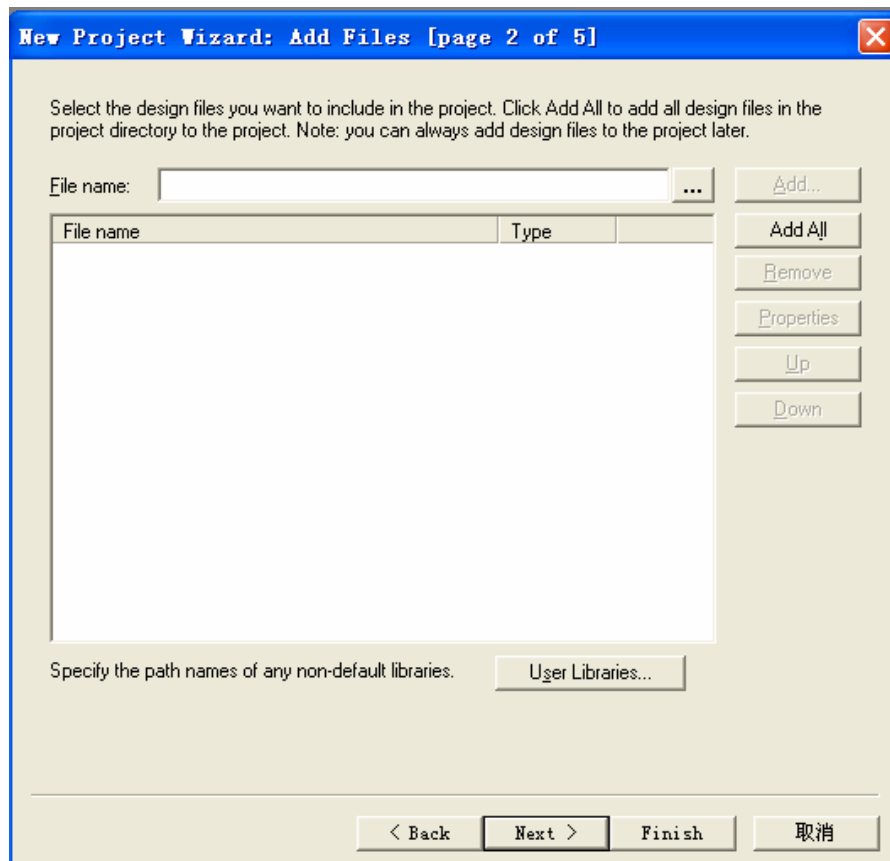


选择 next

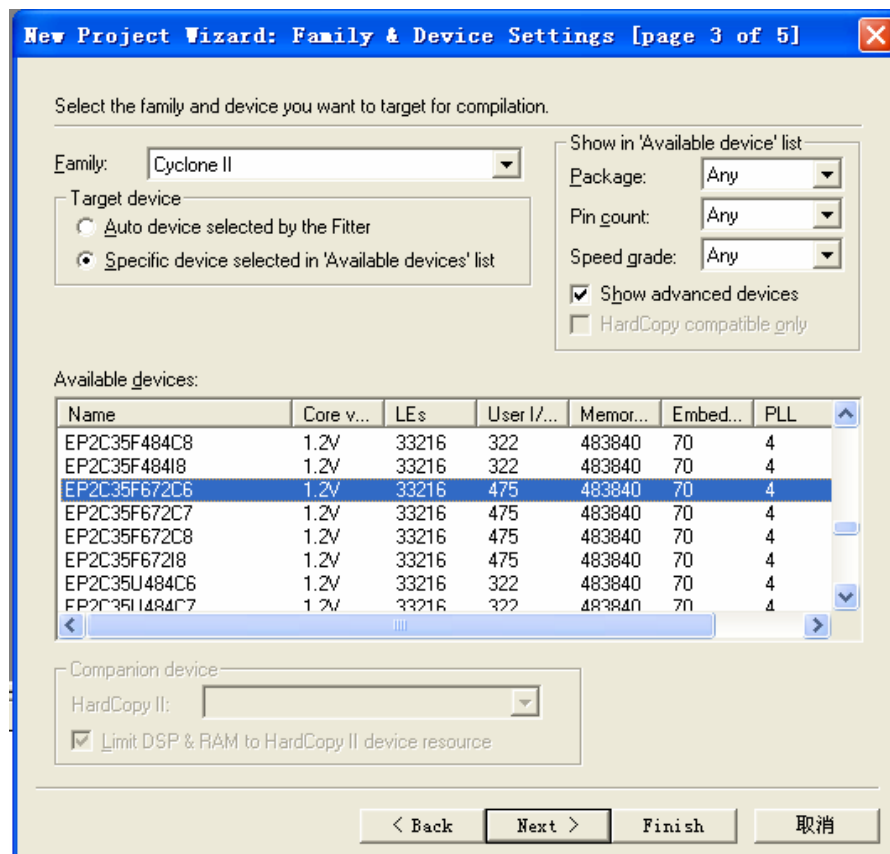
输入项目名称：注意项目不要建在 altera 默认位置，最好建立在自带 U 盘上，因系统重新启动后将会把硬盘所有分区全部复原，你所做工作将全部丢失！！



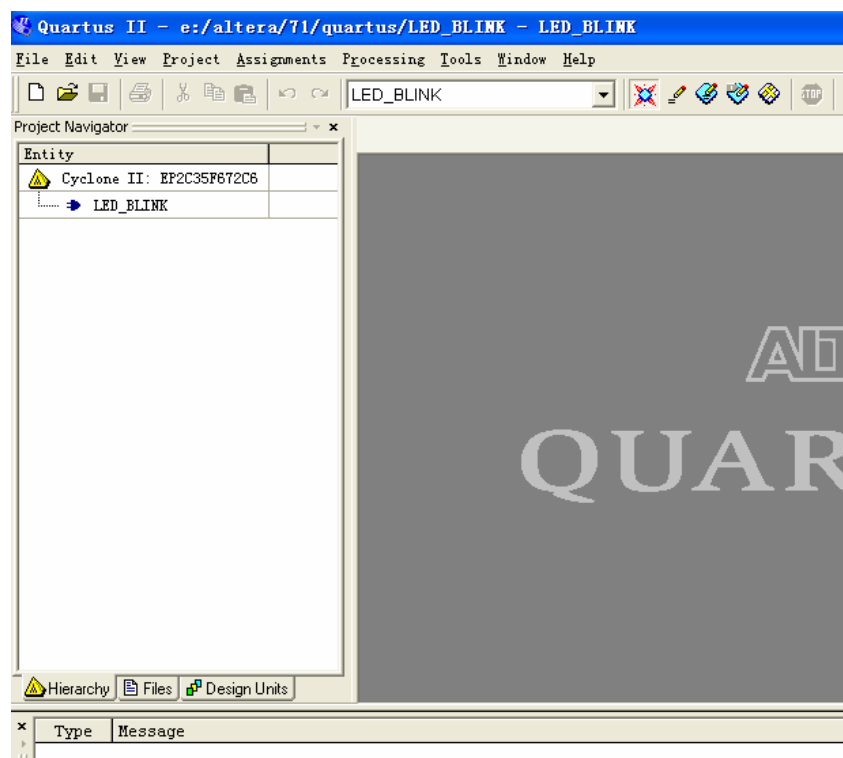
Filename 空,选 next



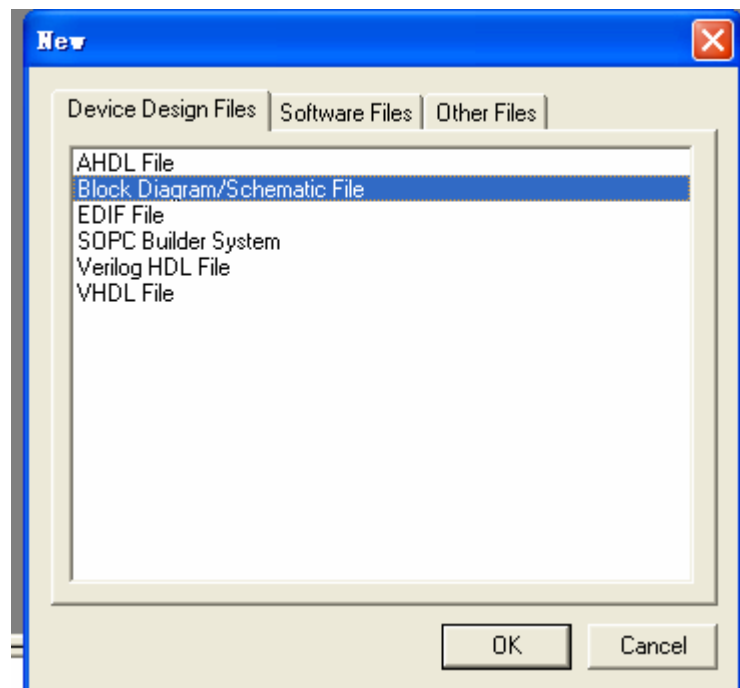
选择 next，出现下面对话框，选择器件：



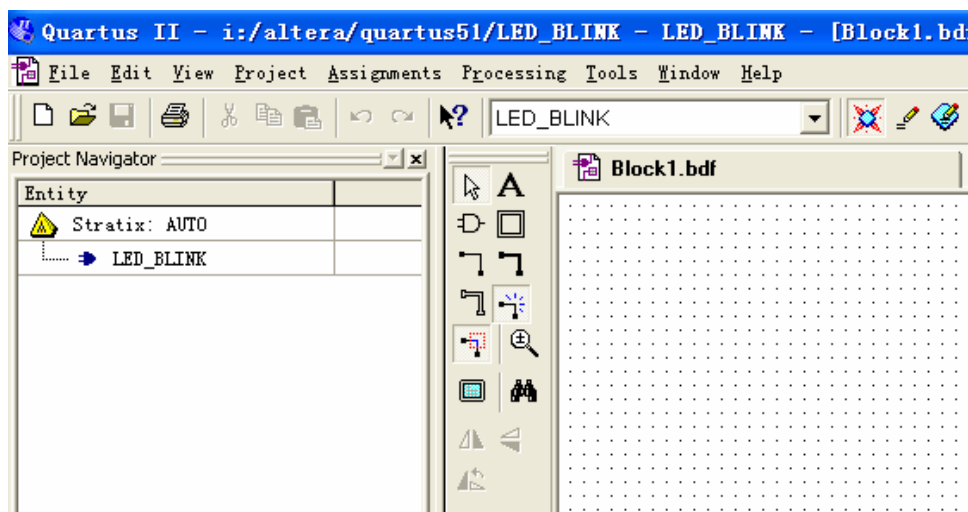
选择 finish，出现以下画面：



建立顶层图,名字必须和项目名一致.选择 file→new,出现如下界面

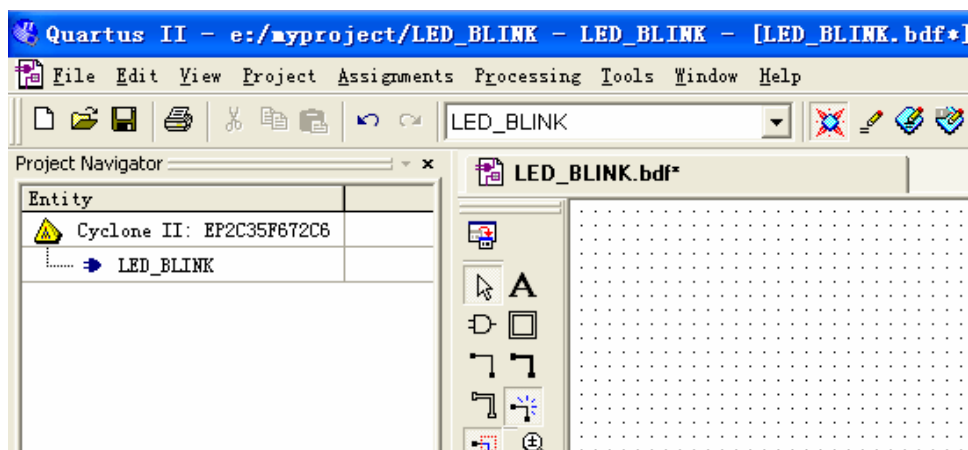


选择 Block diagram/schematic file 然后 ok



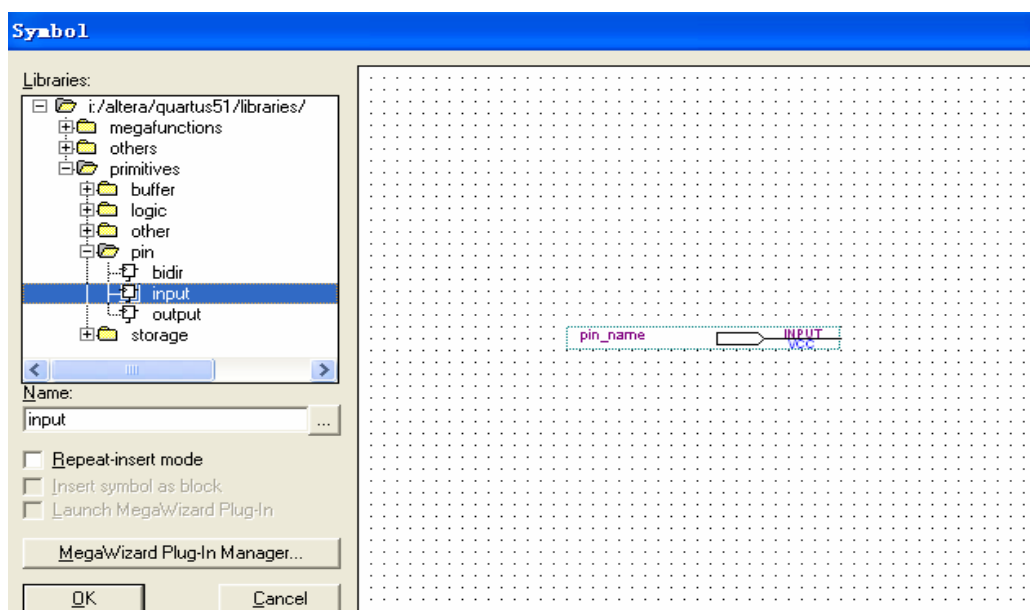
目前,顶层图名字为 block1.bdf

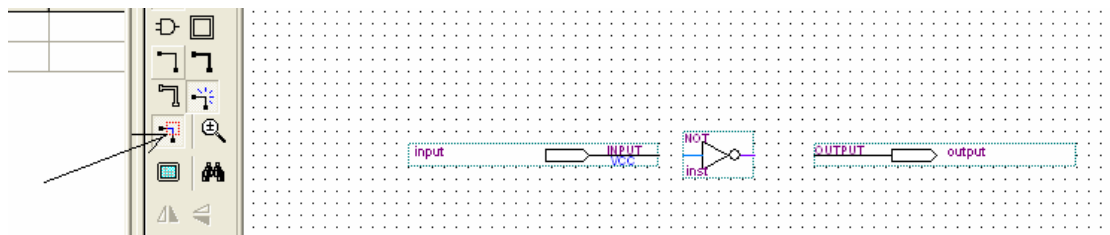
选择 file→save as 不要做任何改动



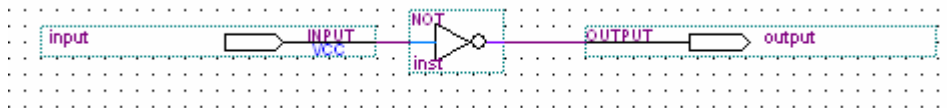
顶层图名字自动变为 LED_BLINK.bdf

在图纸上双击鼠标,
这里先放一个输入按钮

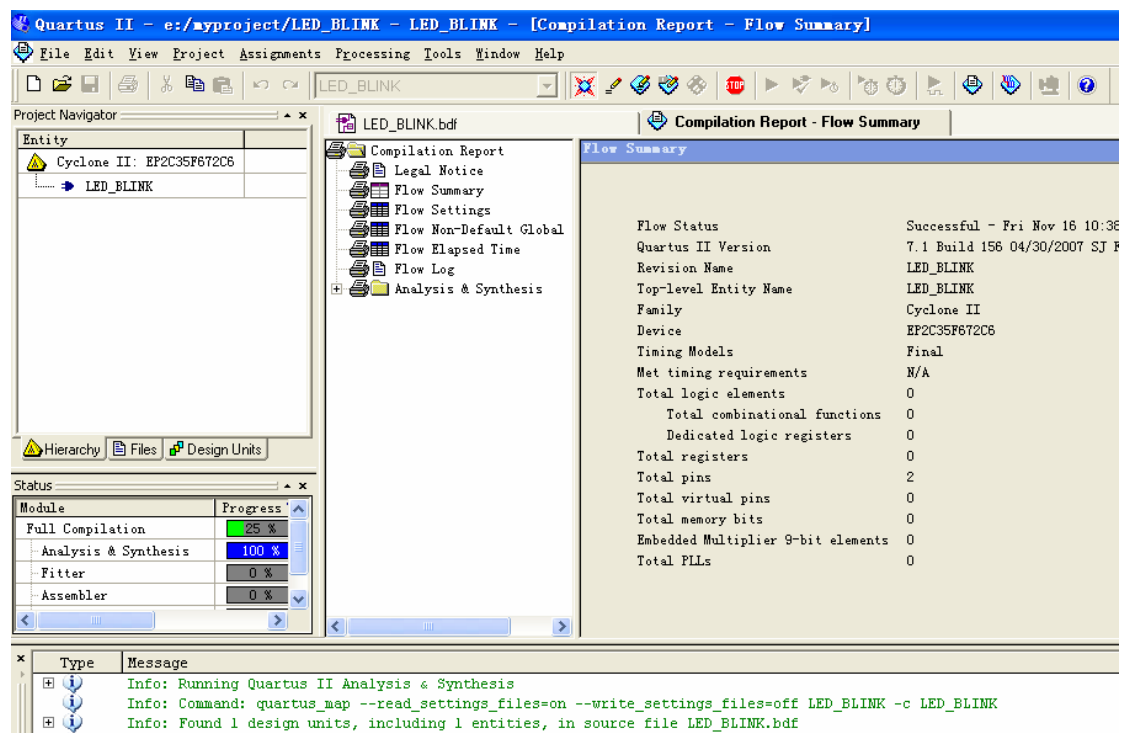




用鼠标选中上面箭头处,在图纸上连线



开始编译,



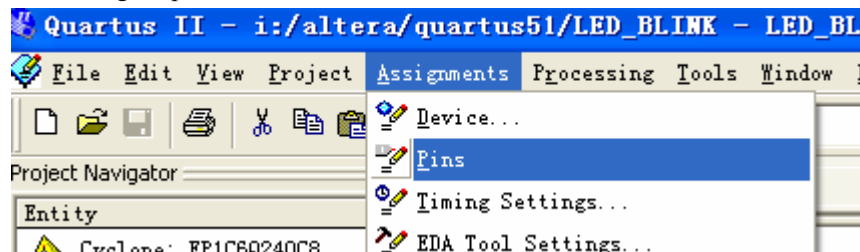
编译成功后开始分配管脚!

常用管脚分配关系可以参见DE2_pin_assignments.csv——非常重要!

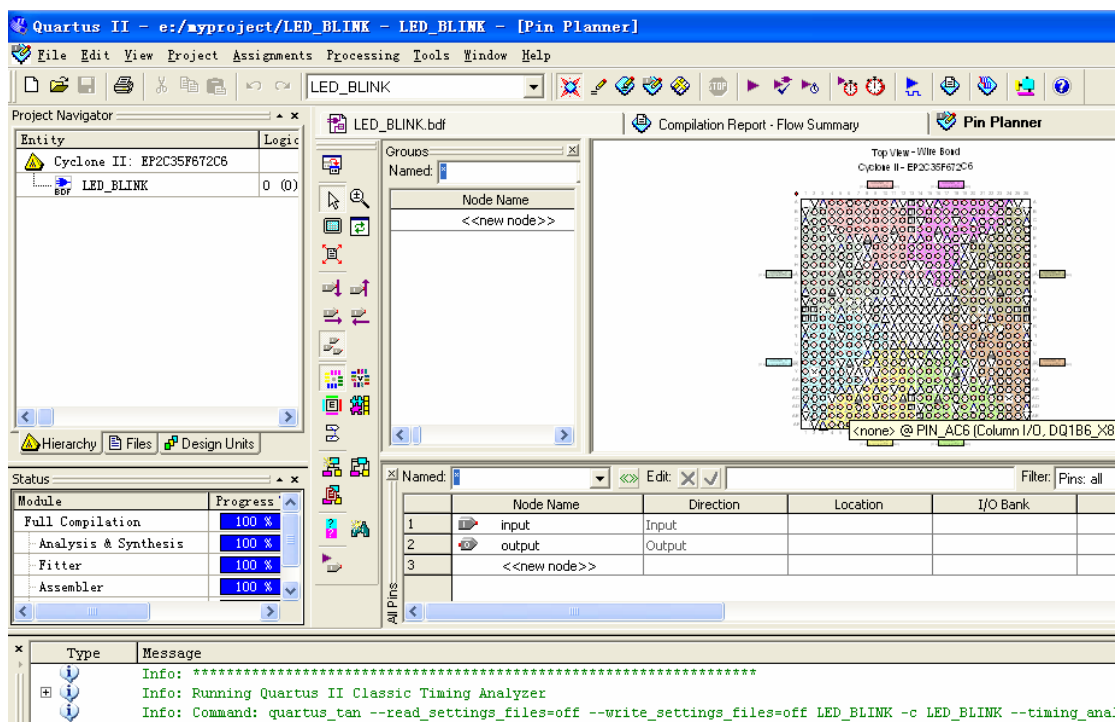
现在选用 SW0 作为输入和 LEDR0 作为输出

图纸设计结束,下面分配 FPGA 管脚,

选择 assign→pins

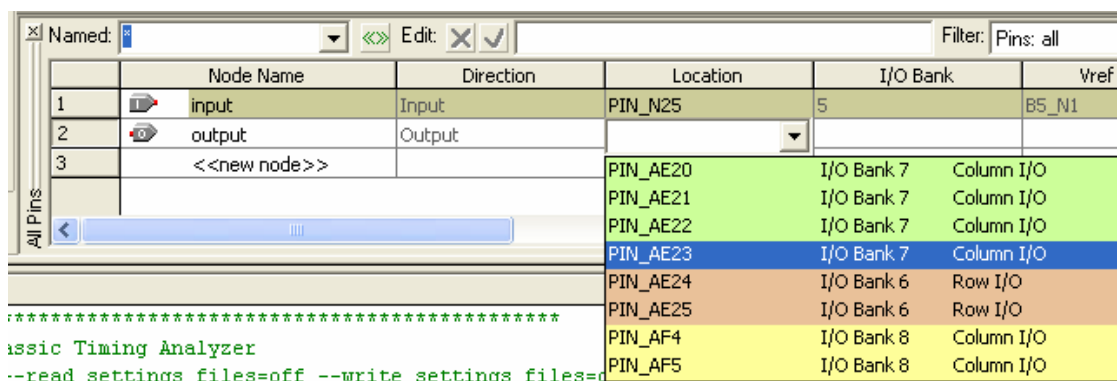


出现下面对话框：

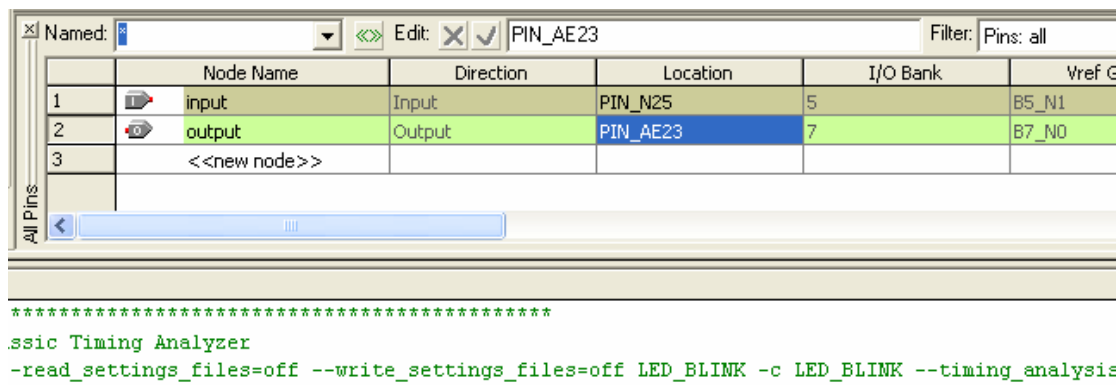


在第 1 行 input 选择对应的脚位是 PIN_N25

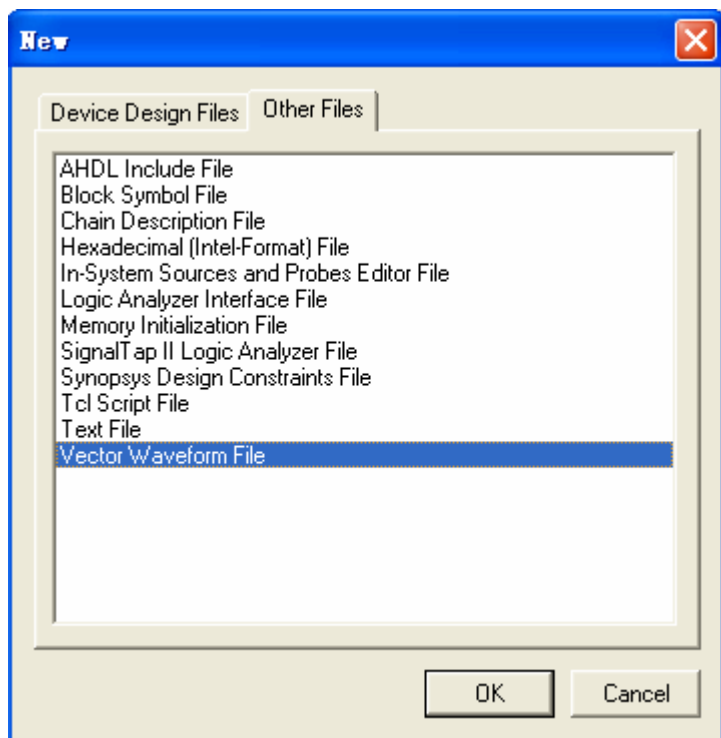
在第 2 行 output 选择对应的脚位是 PIN_AE23



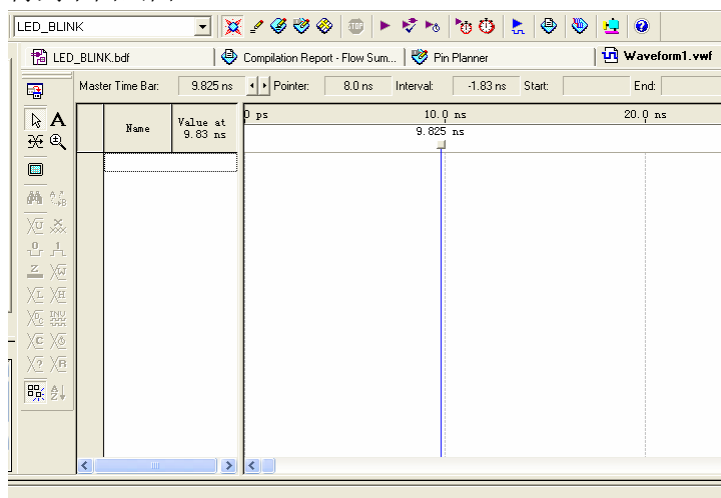
如下图



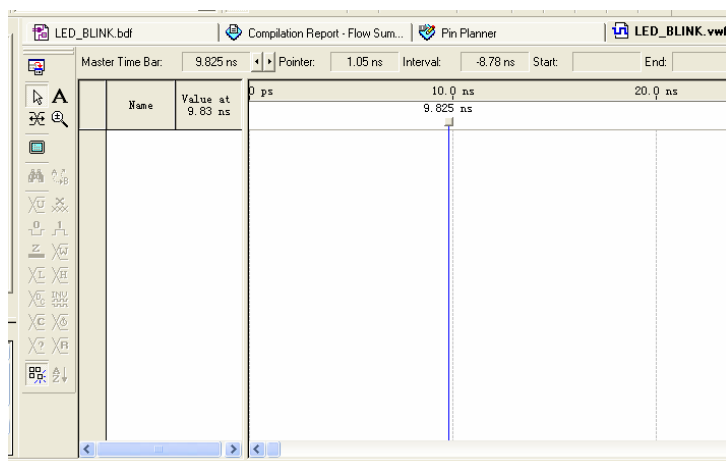
仿真，打开 File->new



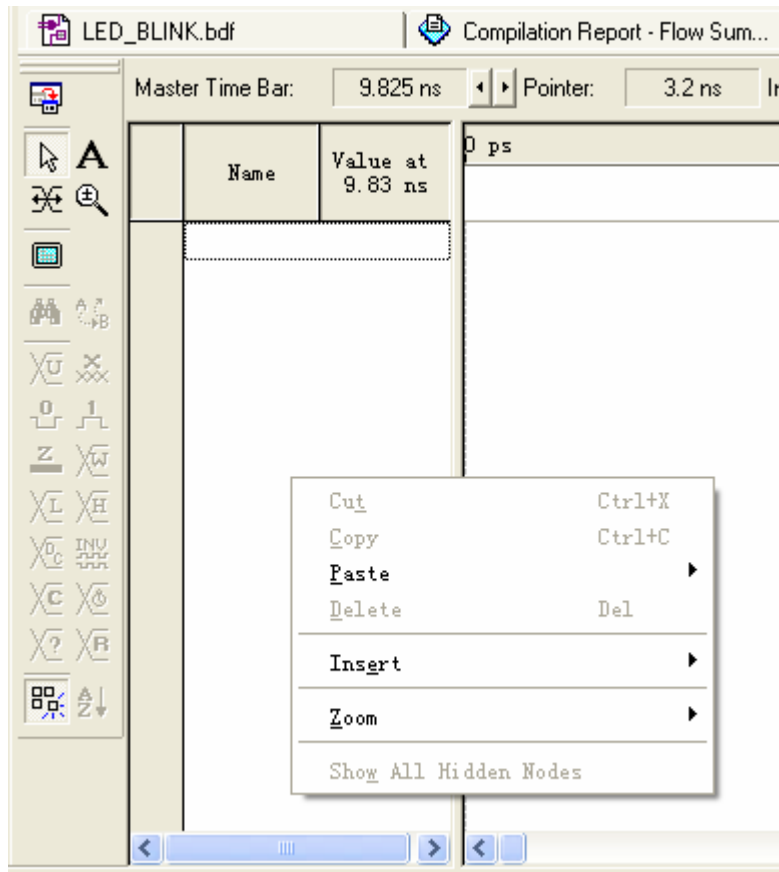
得到下面画面：



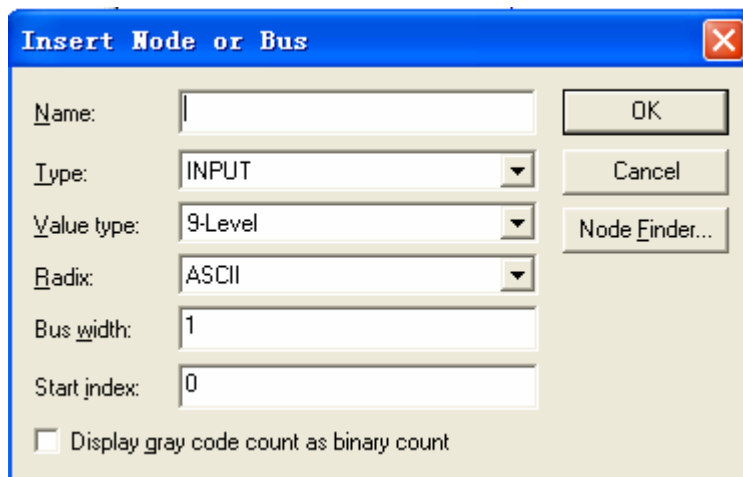
将波形文件改成与 Project 名字一致！！



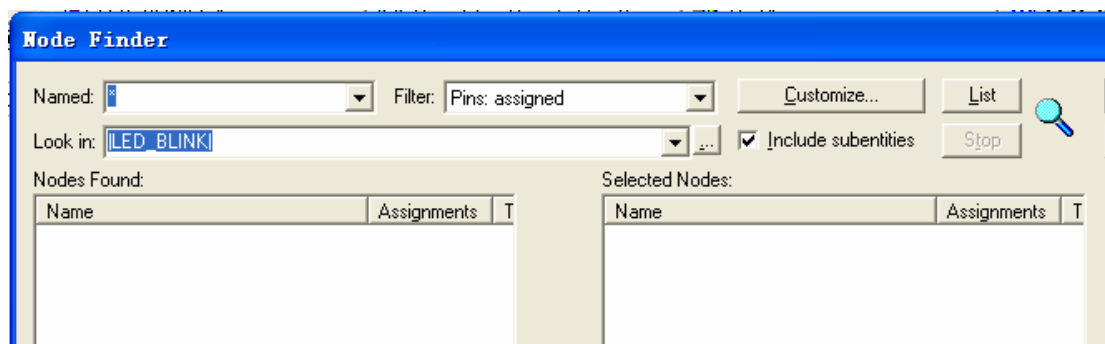
导入要仿真的信号量：右键：



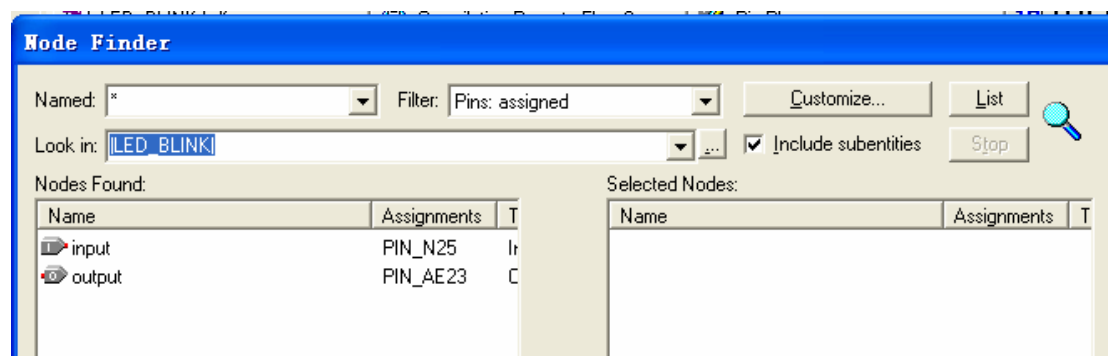
选择 Insert->Node or Bus



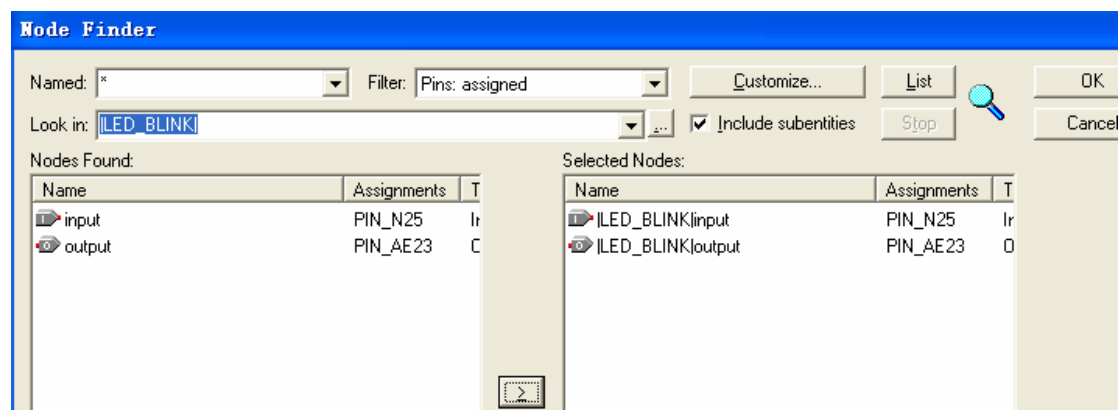
选择 右边 Node Finder



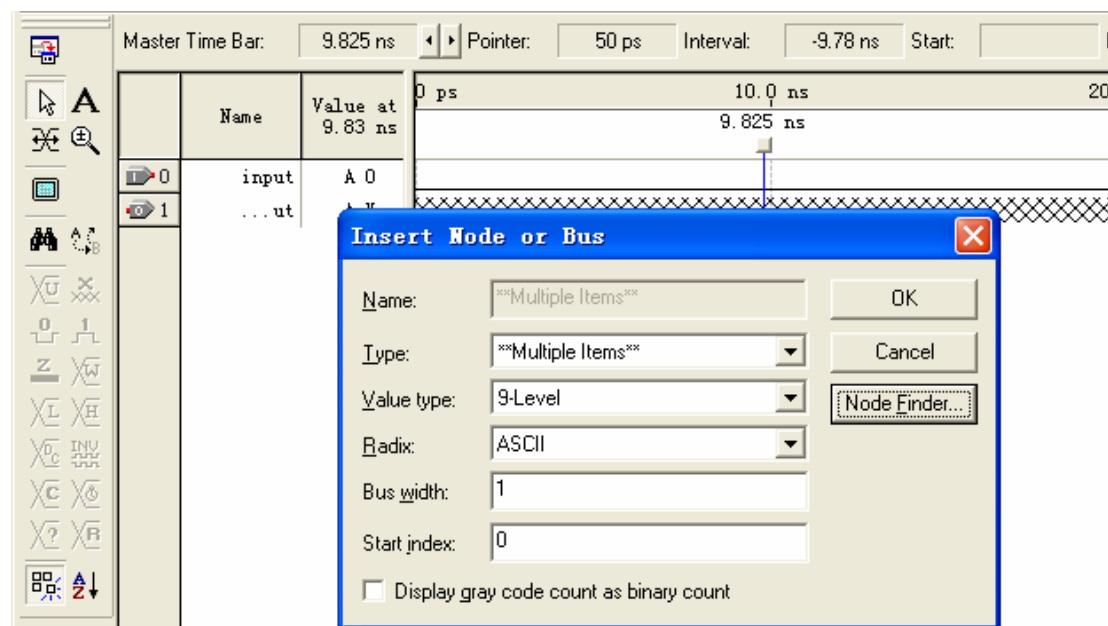
选择 List



将两个信号量导入右边

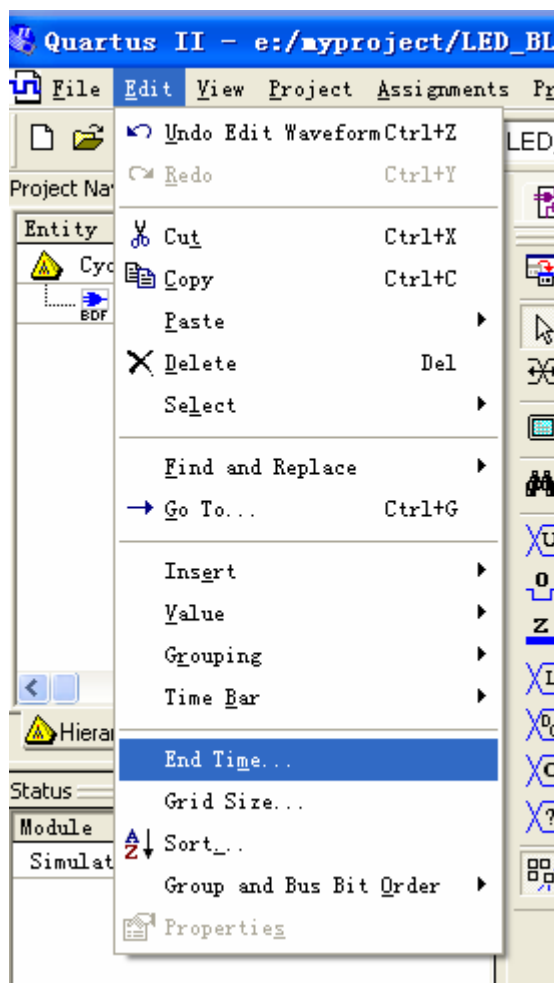


选择 OK，完成！

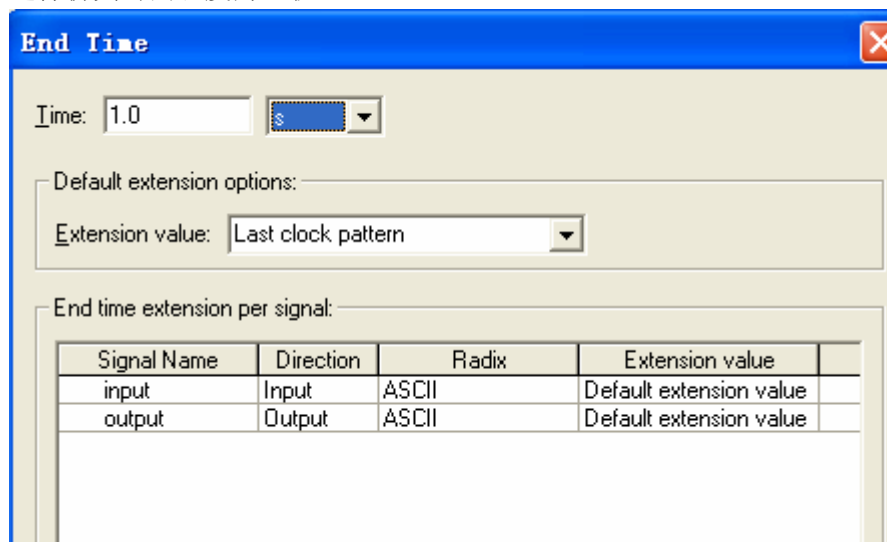


选择 Ok！

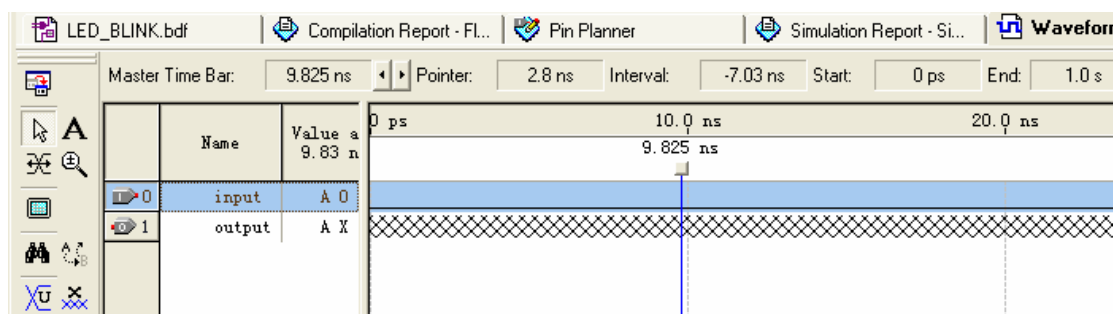
然后选择 Edit->Endtime



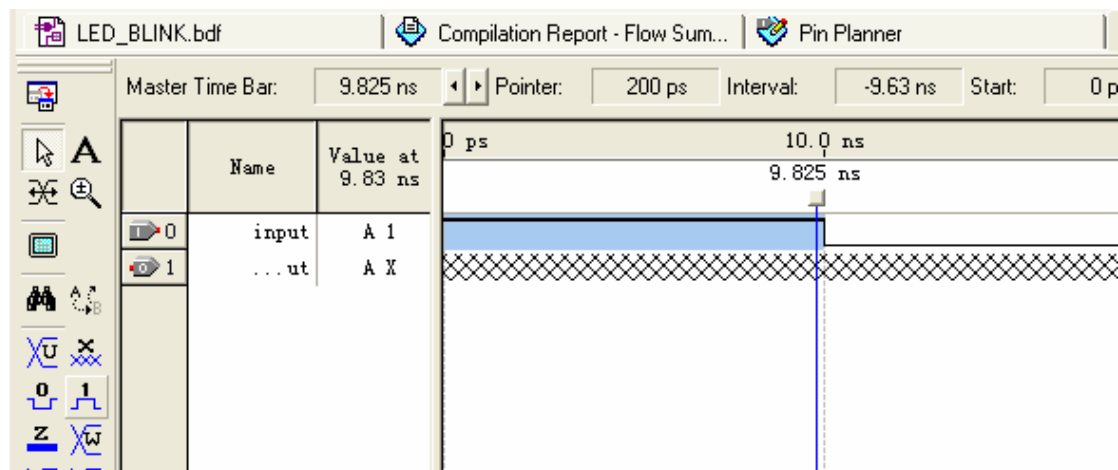
选择仿真时间长度为 1 秒



然后选择



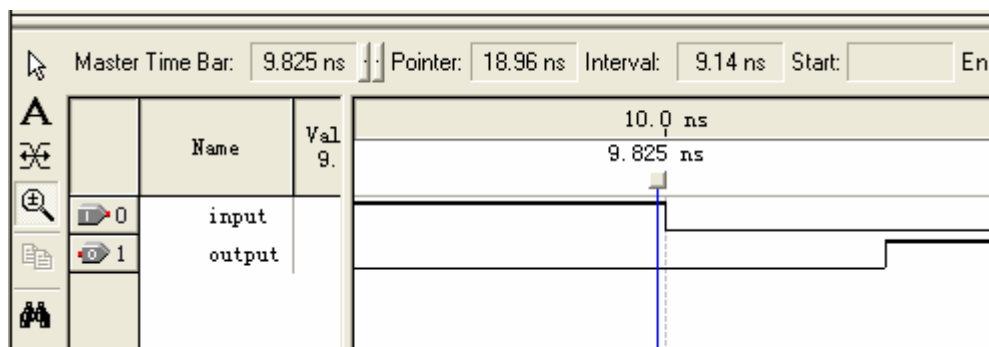
设定 input 输入



选择波形仿真按钮！



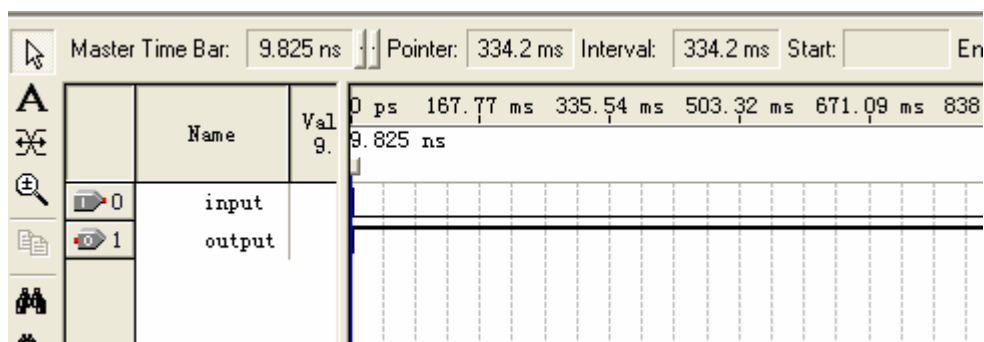
得到以下报告：



看到什么问题？非门将输入信号反相！！

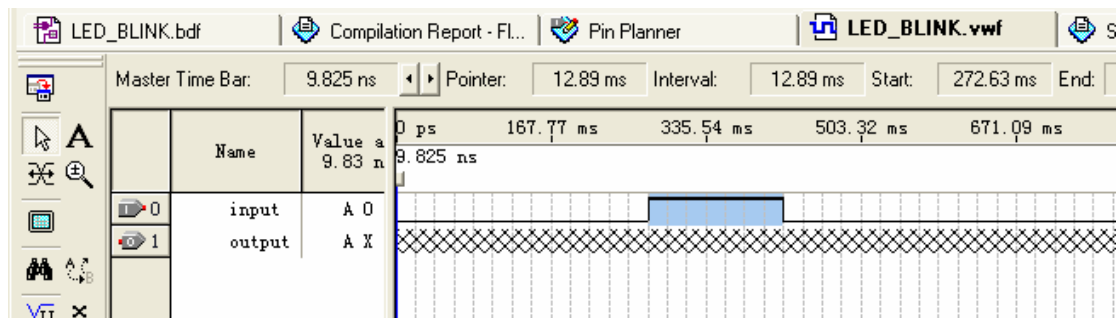
中间有段时间怎么没有反相？

我们这个时候用 Ctrl+W 快捷键再观察波形？

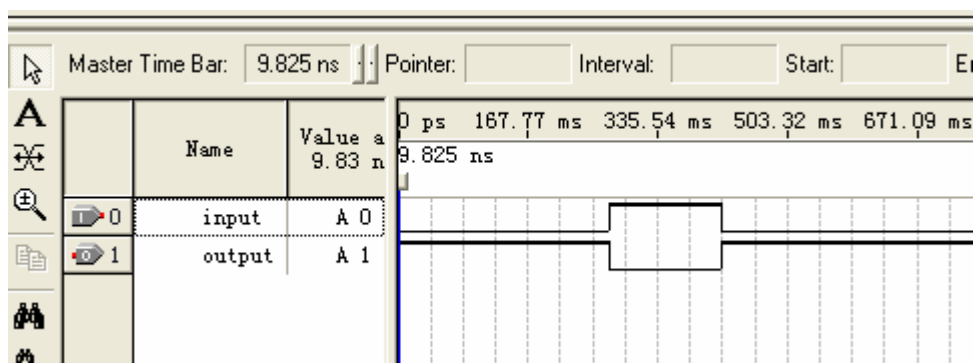


上述图形根本看不出反相，为什么？

我们在设定 input 的值之前也采用 Ctrl+W 快捷键，并设定 input 值，得到：



再选择波形仿真按钮！



很漂亮的波形，确实说明了 input 经过非门反相了！

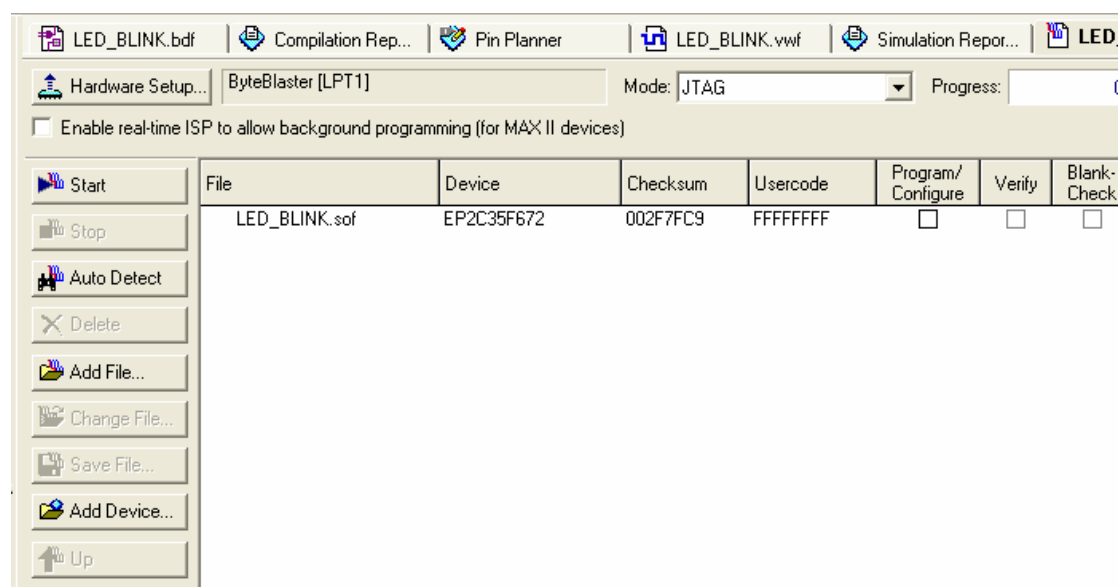
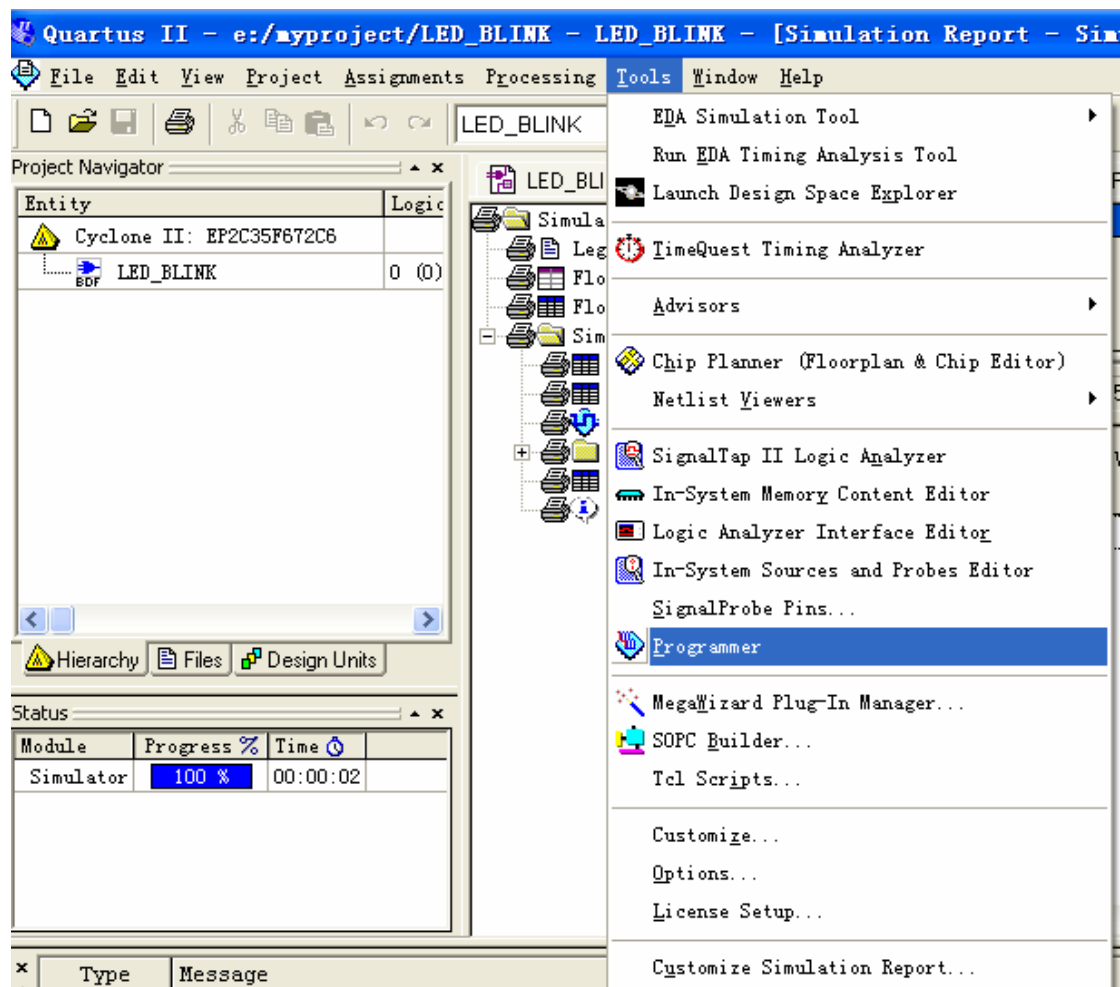
为什么在 ns 级中间会有一段没有反相，而在 s 级根本不存在？

你们所做的设计会出现很多问题，包括逻辑上的错误，要充分利用波形仿真工具，学会自己调试！这种能力才是最重要！不要一出现问题就问老师！

到这里设计完成

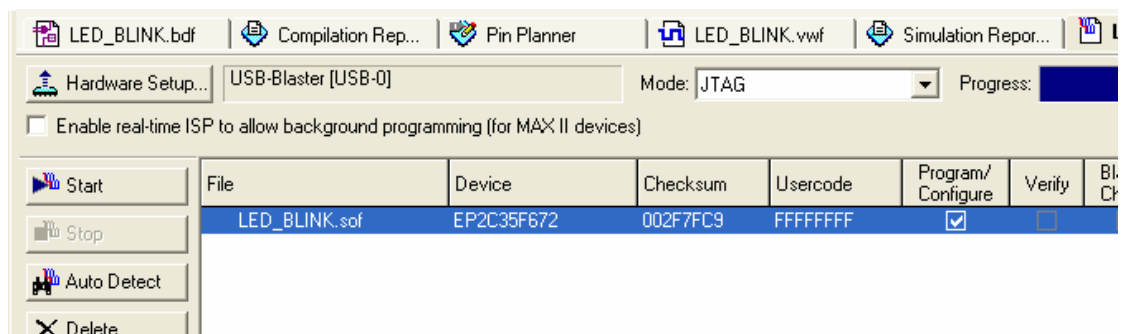
将计算机并口连接并口延长线,并口延长线连接 ByteBlasterII Cable,的一端, ByteBlasterII Cable,的另一端通过 10pin 小电缆连接开发板的 JTAG 口,开发板上电

在 QuartusII IDE 里面选择 tools→programmer



mode 选择 JTAG

然后在应文件的 **program config** 打勾



点 start 开始下载,下载完毕,在提示框提示成功

这时按 SW0,LED0 发光 说明程序运行了

用 HDL 写代码

所有任务如果有同学使用原理图完成外再使用 VHDL 或者 Verilog 来实现将加分，直至得到优！其他相关工程项目，竞赛也将优先参加！

实验代码很简单,这里主要掌握如何将代码放到开发环境

一个减法例子

计算a-b 在数码管显示结果

在本实验中,用s1,s2,s3,s4模拟外部数据a的输入

b 在程序中固定为1

```

/*
两个4位二进制数的减法，结果输出到数码管显示
C[0] a,C[1] b C[2] c,C[3] d,C[4] e,C[5] f,C[6] g
按钮SW3—SW0 表示四位二进制 全部按下表示0 不按表示1111
*/
module sub(a,c,en);

input[3:0] a;

output[6:0] c;
reg[6:0] c;
output en;
wire[3:0] b;
wire[3:0] c_tmp;

assign en=0;
assign b=1;
assign c_tmp=a-b;

always@(c_tmp)
begin
    case(c_tmp)
        4'b0000:
            c=7'b100_0000;
        4'b0001:
            c=7'b111_1001;
        4'b0010:
            c=7'b010_0100;
        4'b0011:
            c=7'b011_0000;
        4'b0100:
            c=7'b001_1001;
        4'b0101:
            c=7'b001_0010;
        4'b0110:
            c=7'b000_0010;
        4'b0111:
    
```

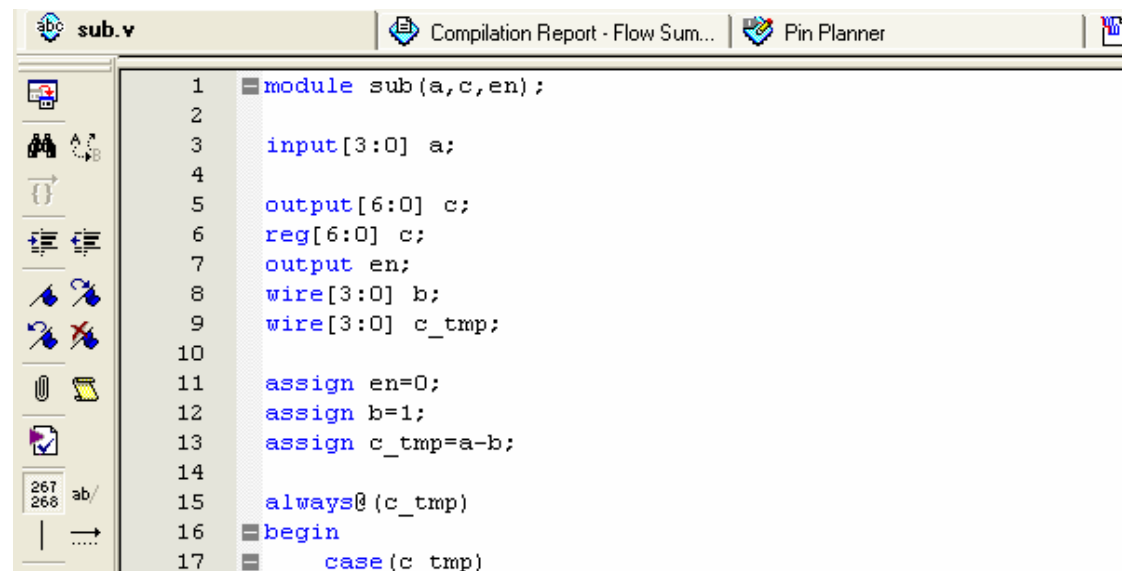
```
        c=7'b111_1000;
4'b1000:
        c=7'b000_0000;
4'b1001:
        c=7'b001_0000;
4'b1010:
        c=7'b000_1000;
4'b1011:
        c=7'b000_0011;
4'b1100:
        c=7'b100_0110;
4'b1101:
        c=7'b010_0001;
4'b1110:
        c=7'b000_0110;
4'b1111:
        c=7'b000_1110;
    endcase
end
endmodule
```

步骤,

在 quartus 中

创建一个工程,名字叫 SUB

用 file→new 创建一个顶层图, 选择 verilog HDL 类型 文件名字 sub.v















输入以上代码

和前面实验一样,设置 device 和 pin

Pin 的引脚如下

Named: PIN_AE23 Edit: ☐ ☒ Filter

	Node Name	Direction	Location	I/O Bank
1	 a[3]	Input	PIN_AE14	7
2	 a[2]	Input	PIN_P25	6
3	 a[1]	Input	PIN_N26	5
4	 a[0]	Input	PIN_N25	5
5	 c[6]	Output	PIN_V13	8
6	 c[5]	Output	PIN_V14	8
7	 c[4]	Output	PIN_AE11	8
8	 c[3]	Output	PIN_AD11	8
9	 c[2]	Output	PIN_AC12	8
10	 c[1]	Output	PIN_AB12	8
11	 c[0]	Output	PIN_AF10	8
12	 en	Output	PIN_AE23	7

编译并下载,验证程序的正确性.

管脚的输入方式

对新建的工程如果不准备每次输入管脚,可以打开对应的*.qsf 文件,找到# Pin & Location Assignments

将以下代码拷贝到新建的项目的.qsf 文件里.也可以将它拷贝到建立的 tcl 文件,并做适当的修改.

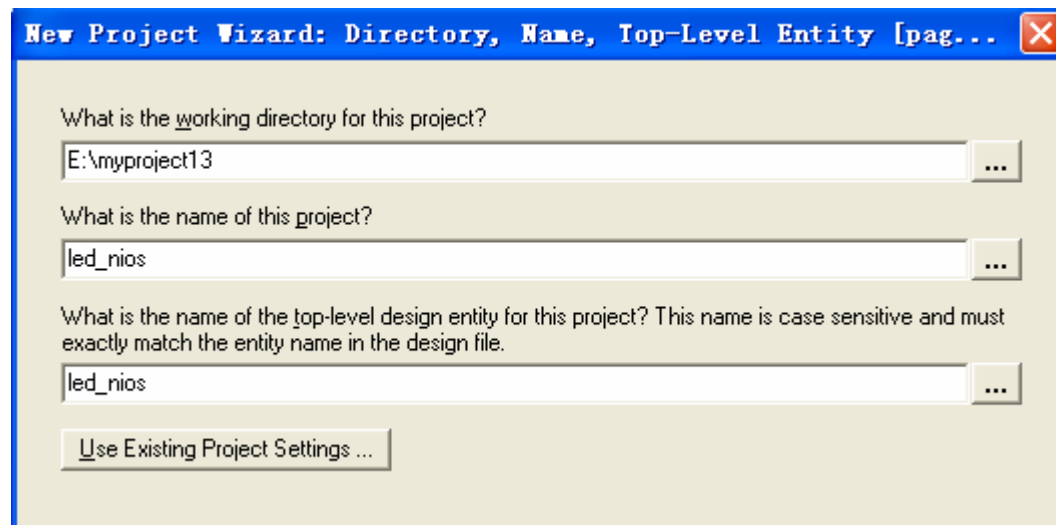
```
set_location_assignment PIN_N25 -to a[0]
set_location_assignment PIN_AE23 -to a[1]
set_location_assignment PIN_P25 -to a[2]
set_location_assignment PIN_AE14 -to a[3]
set_location_assignment PIN_AF10 -to c[0]
set_location_assignment PIN_AB12 -to c[1]
set_location_assignment PIN_AC12 -to c[2]
set_location_assignment PIN_AD11 -to c[3]
set_location_assignment PIN_AE11 -to c[4]
set_location_assignment PIN_V14 -to c[5]
set_location_assignment PIN_V13 -to c[6]
set_location_assignment PIN_AE23 -to en
```

NIOSII 实验

实验目的:做一个 NIOSII CPU 处理器,并在此处理器上编写一个简单的程序,让开发板上的四

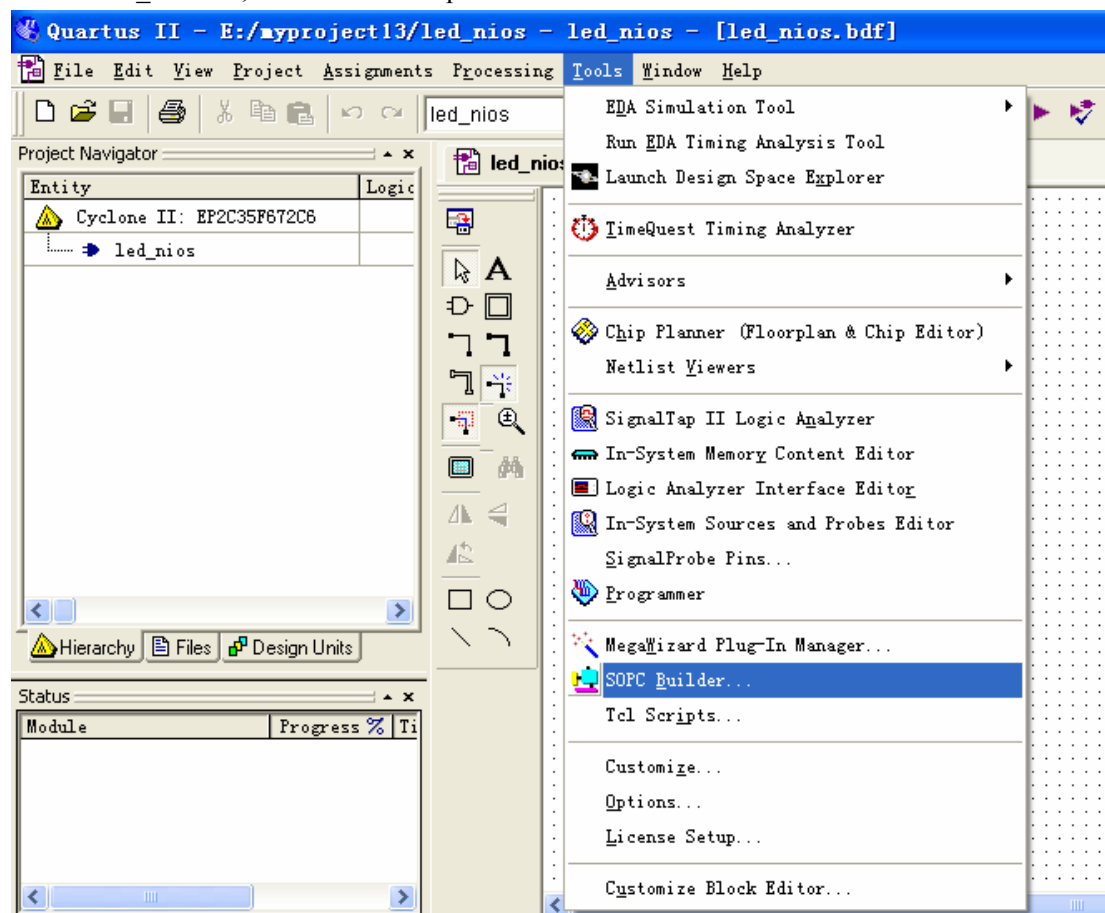
个跑马灯循环亮.

用 Quartus 建立一个工程, 然后 finish

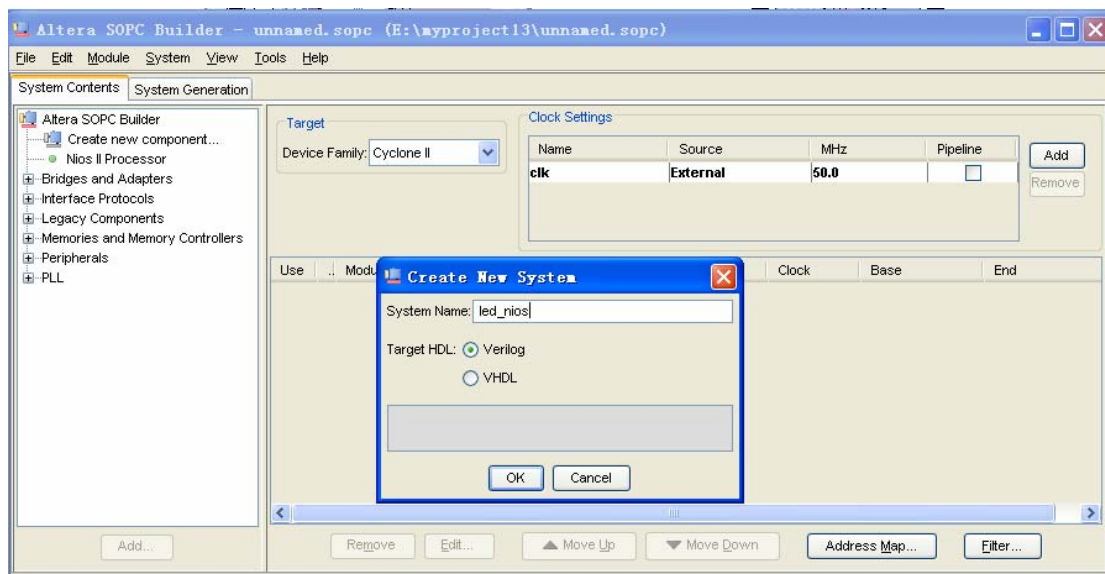


创建一个空的顶层图,类型为 block diagram/schematic file

另存为 led_nios.bdf, 选择 tools→sopc builder

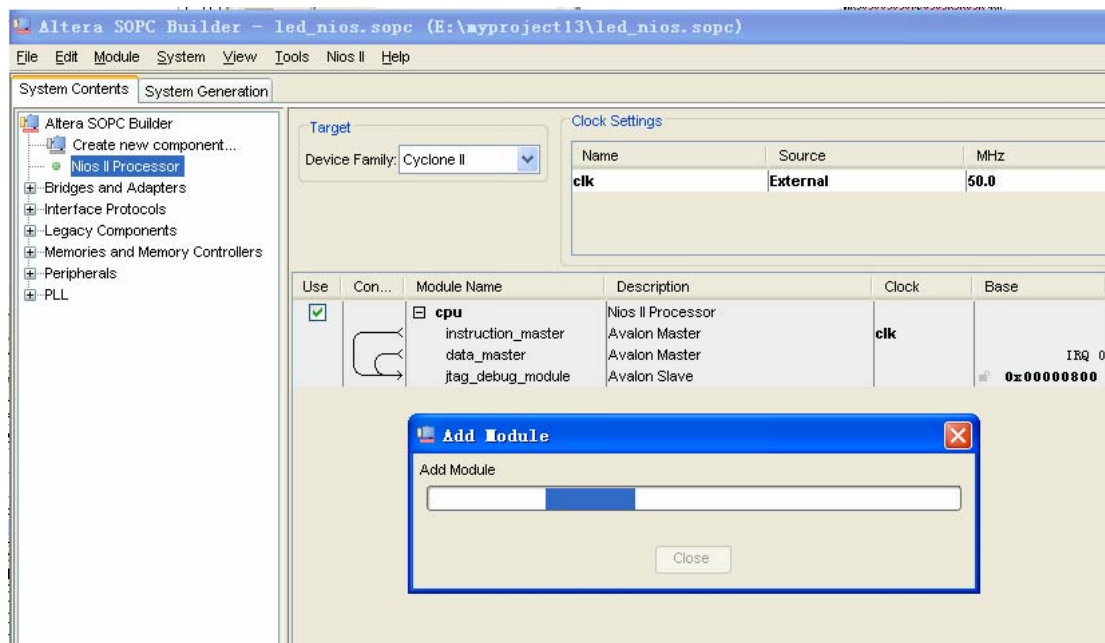


启动 socp builder

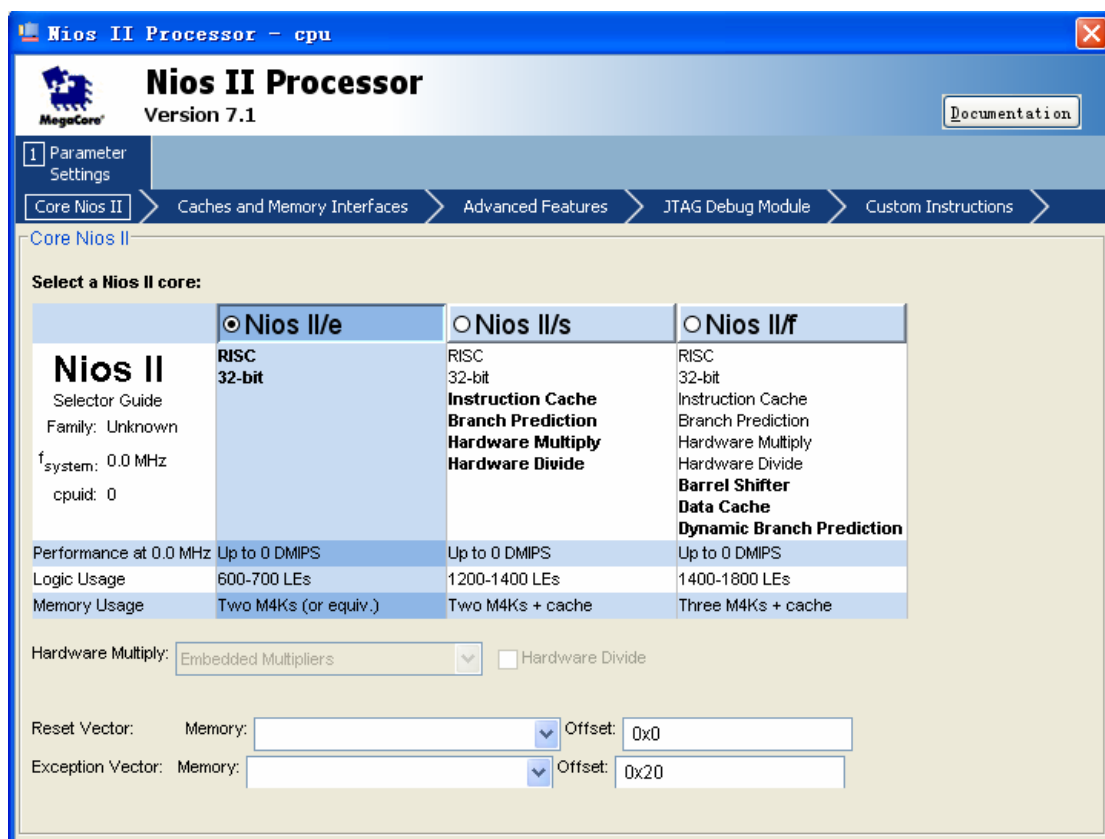


命名为 led_nios_cpu,确认

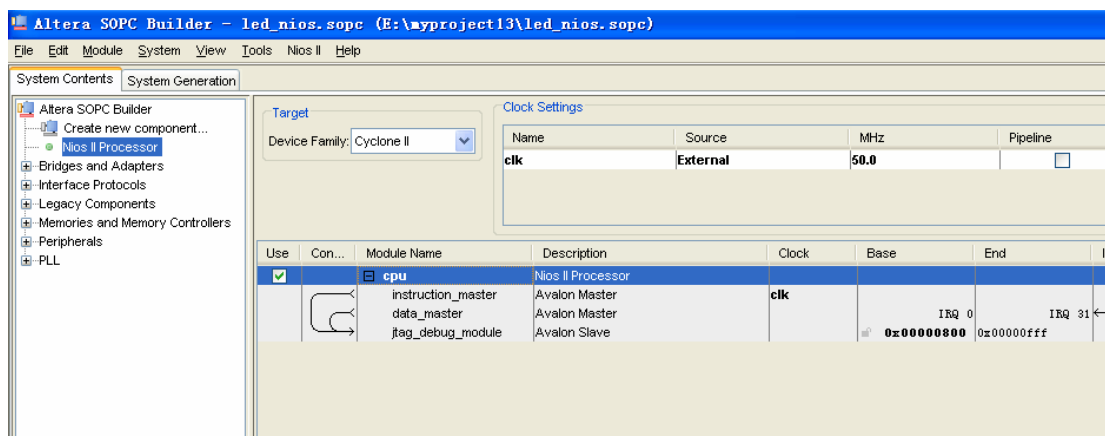
在 Avalon components 下双击 nios ii processor,出现如下



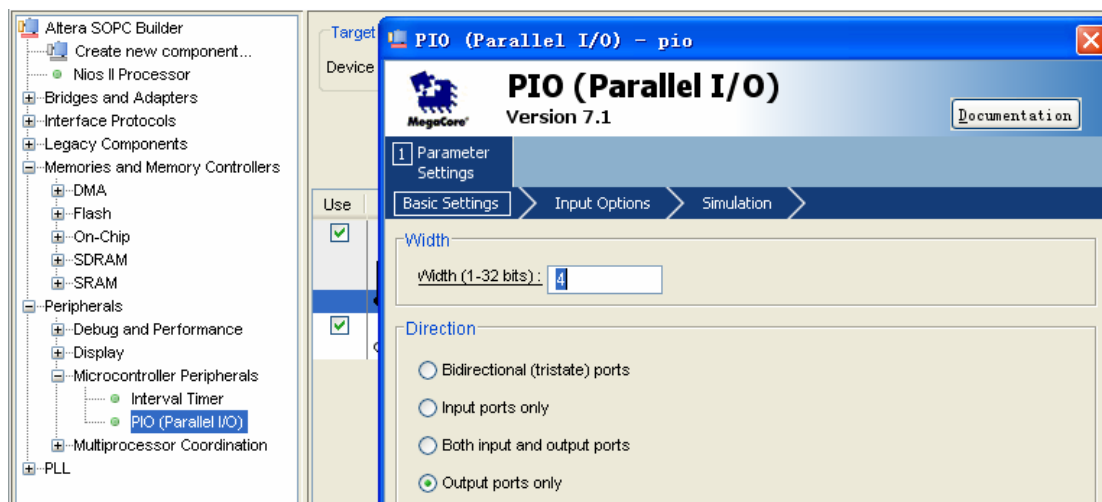
选择 NIOS II/e



按 finish,出现以下界面



然后在左侧 peripherals 下选择 microcontroller peripherals 再选择 PIO

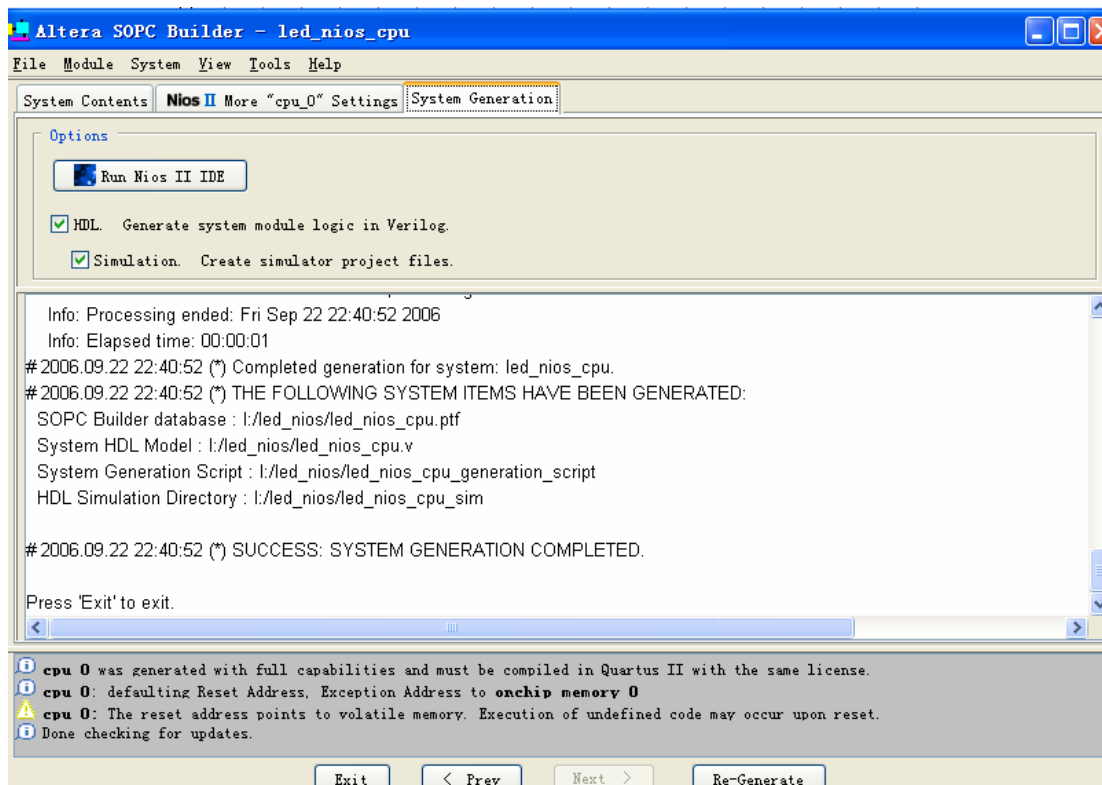


Width 设置为 4bit direction 为 output ports only, 然后 finish

在 system menu 下选择 auto assignment base address ,auto assignment IRQs, 如下图

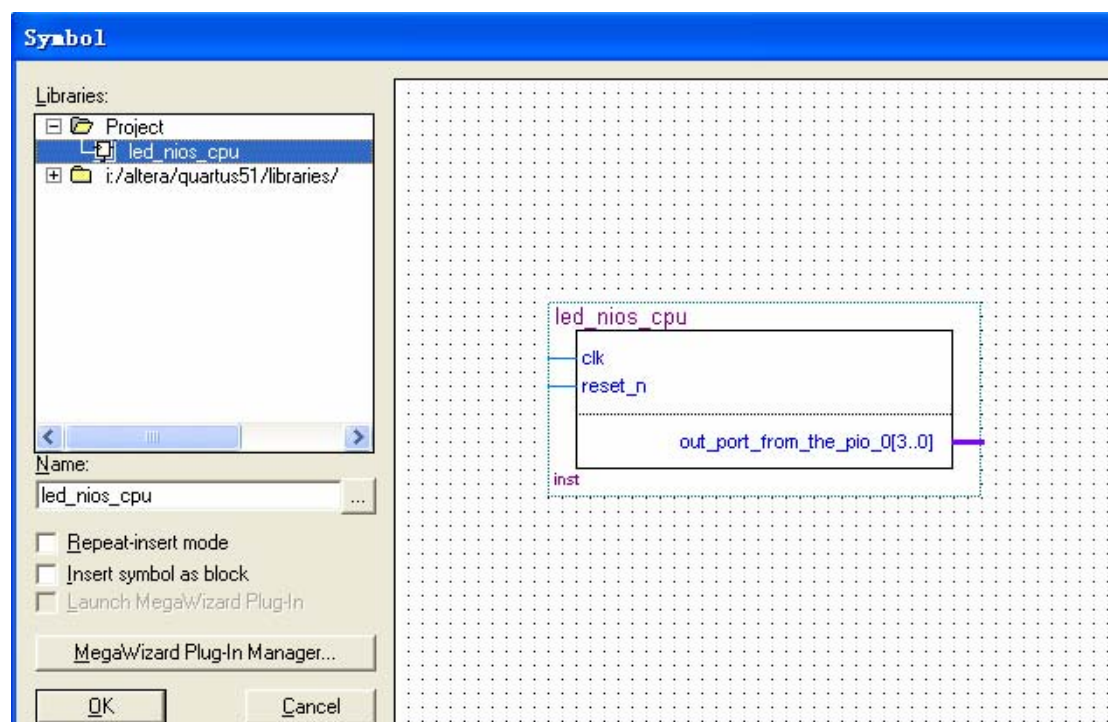


最后 generate 生成系统



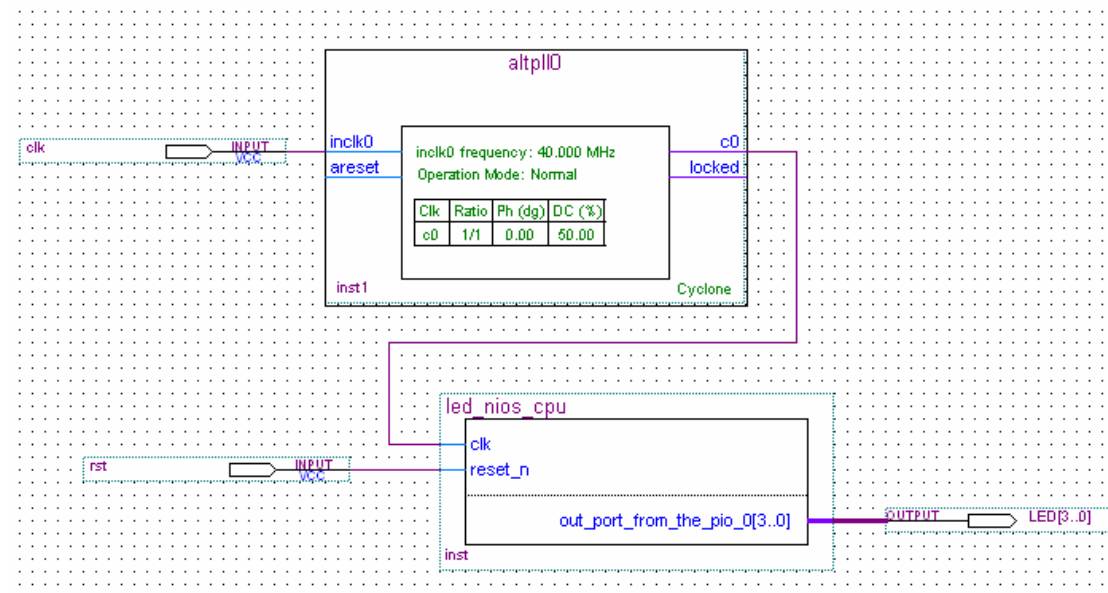
成功后 exit 退出 socp builder.现在回到 quartus IDE

在设计图空白的地方双击鼠标



在 symbol 的 library 的 project 下选择刚才生成的 led_nios_cpu,并确定
这时候原理图上出现了刚才设计的 nios 处理器.

选择输入频率 50M,完成设置.在原理图上连线,并放置管脚 rst,clk led[4..0]



4个LED发光管的FPGA对应关系如下
高电位发光

LEDR[0] PIN_AE23
LEDR[1] PIN_AF23
LEDR[2] PIN_AB21

LEDR[3] PIN_AC22

4个开关的对应如下

按下为低电平

SW[0] PIN_N25

SW[1] PIN_N26

SW[2] PIN_P25

SW[3] PIN_AE14

CLOCK_50 PIN_N2

然后设置器件，以及配置芯片类别，

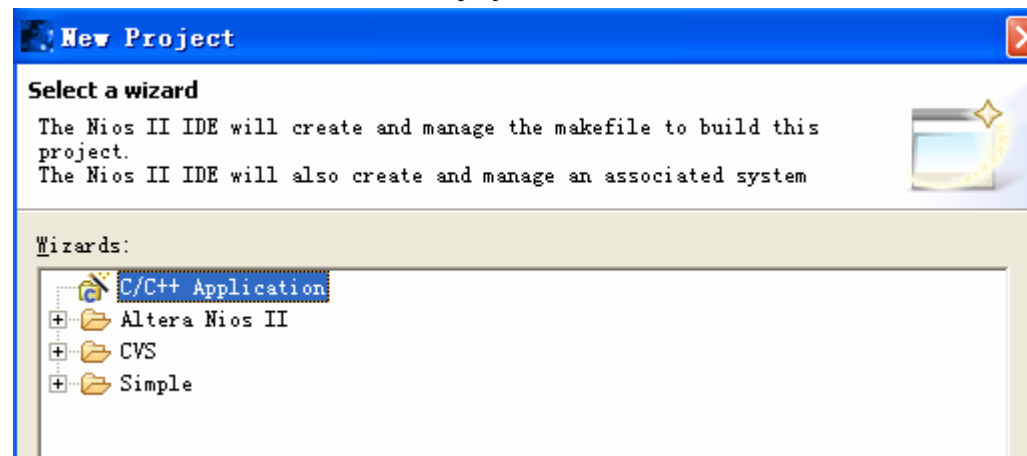
分配管脚

编译程序,通过下载端口下载到开发板.到这里,NIOSII 硬件生成完毕,

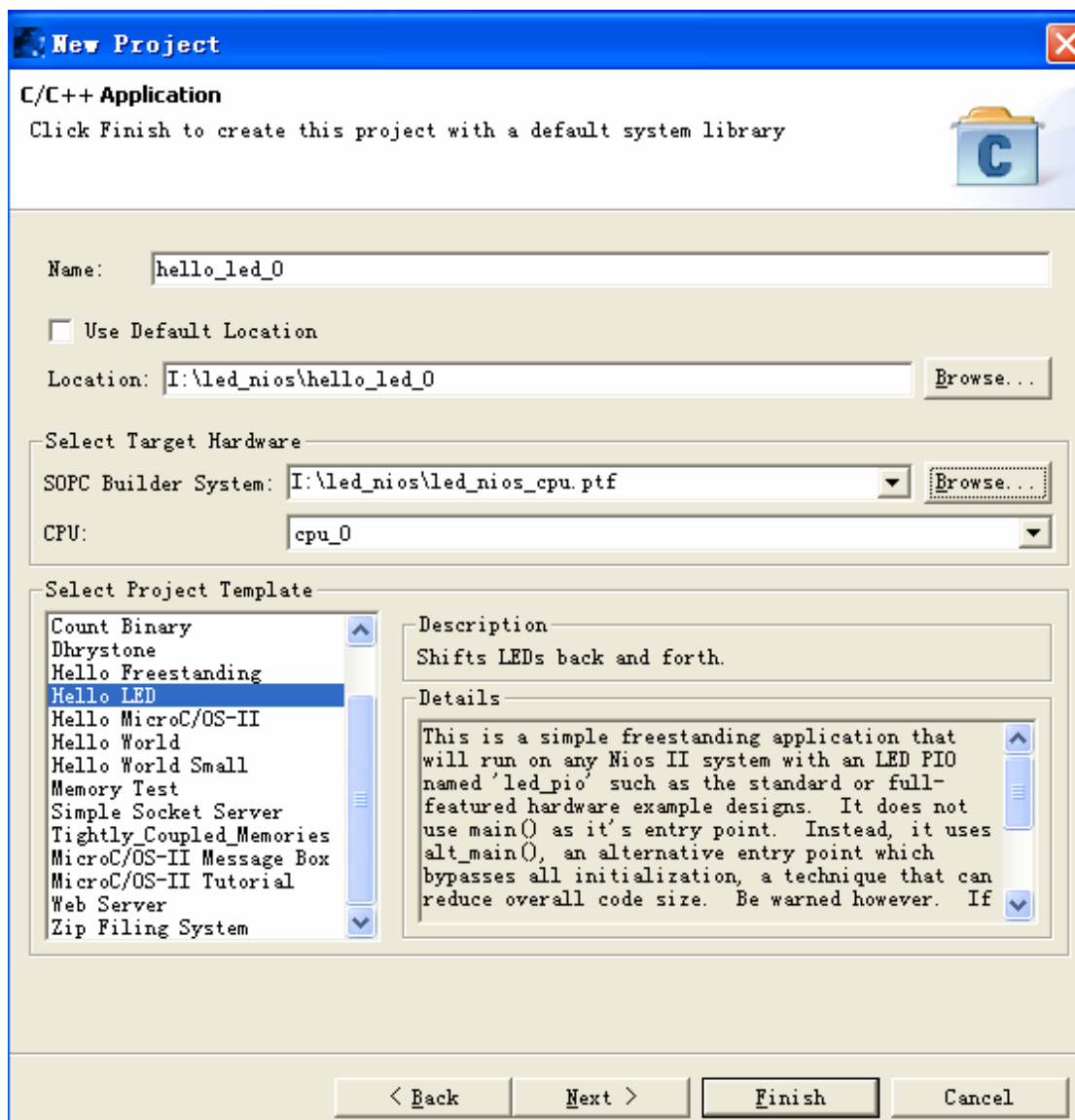
Quartus 不要关闭!!

以下编写在 NIOSII 硬件上面运行的程序

启动 NIOSII IDE, 选择 file→new→project



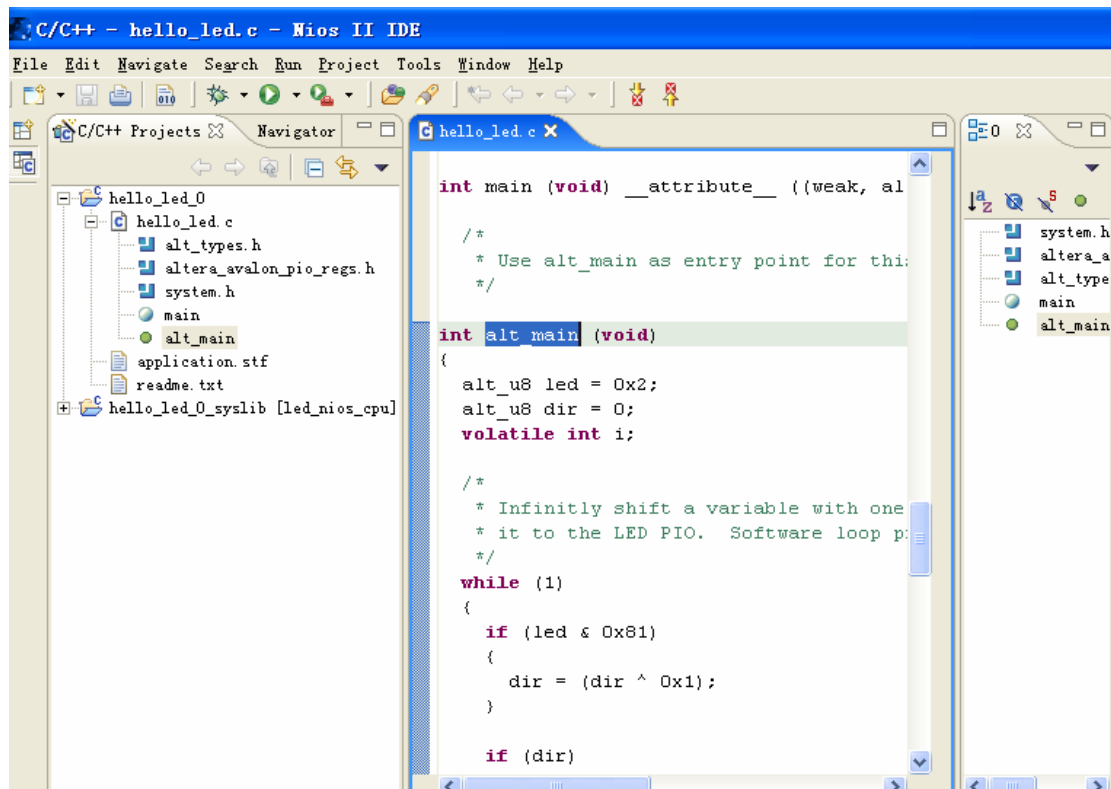
运行 C++ application 向导



选择一个例子工程 Hello_led,这样只要在这基础上适当修改就可以了.

在 select target hardware 中用 browse 到刚才做 niosii 工程的目录中找后缀是.ptf 的文件.

Finish 进入下一步



对 alt_main 函数进行修改

```
int alt_main (void)
{
    alt_u8 led = 0x2;
    volatile int i;

    /*
     * Infinitely shift a variable with one bit set back and forth, and write
     * it to the LED PIO.  Software loop provides delay element.
     */
    while (1)
    {
        i = 0;
        while (i<200000)
            i++;

        led=0x1;
        IOWR_ALTERA_AVALON_PIO_DATA(PIO_0_BASE, led);

        i = 0;
        while (i<200000)
            i++;
        for(i=0;i<300000;i++)
            led=0x2;
        IOWR_ALTERA_AVALON_PIO_DATA(PIO_0_BASE, led);
    }
}
```

```

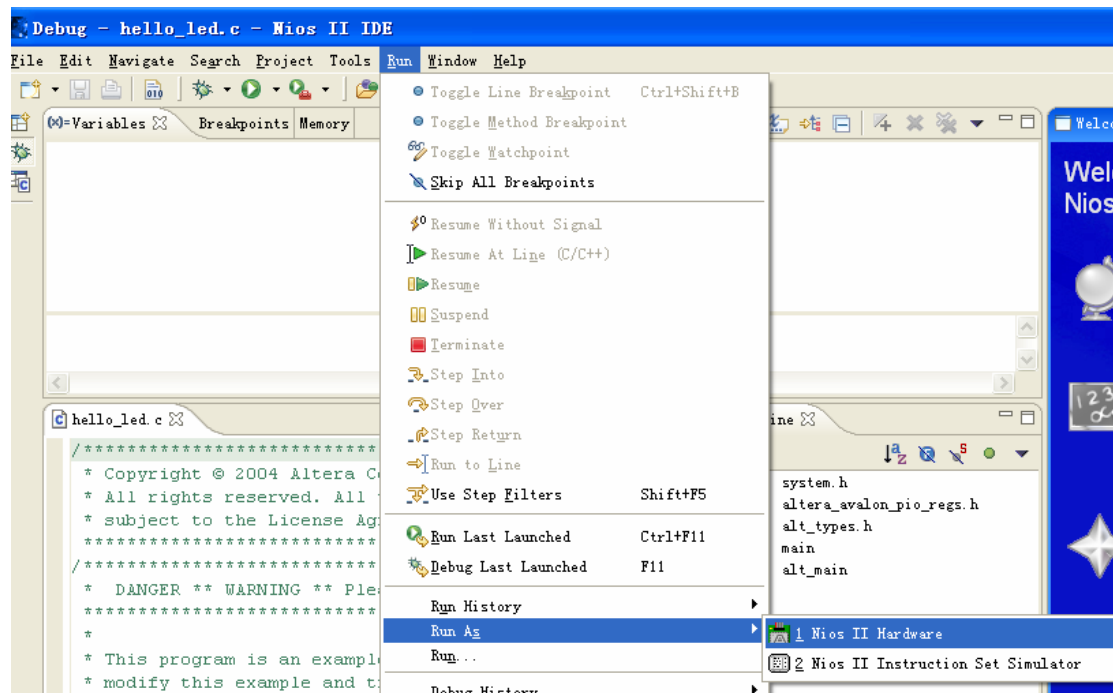
        i = 0;
    while (i<200000)
        i++;
    for(i=0;i<300000;i++)
        led=0x4;
    IOWR_ALTERA_AVALON_PIO_DATA(PIO_0_BASE, led);
        i = 0;
    while (i<200000)
        i++;
    for(i=0;i<300000;i++)
        led=0x8;
    IOWR_ALTERA_AVALON_PIO_DATA(PIO_0_BASE, led);

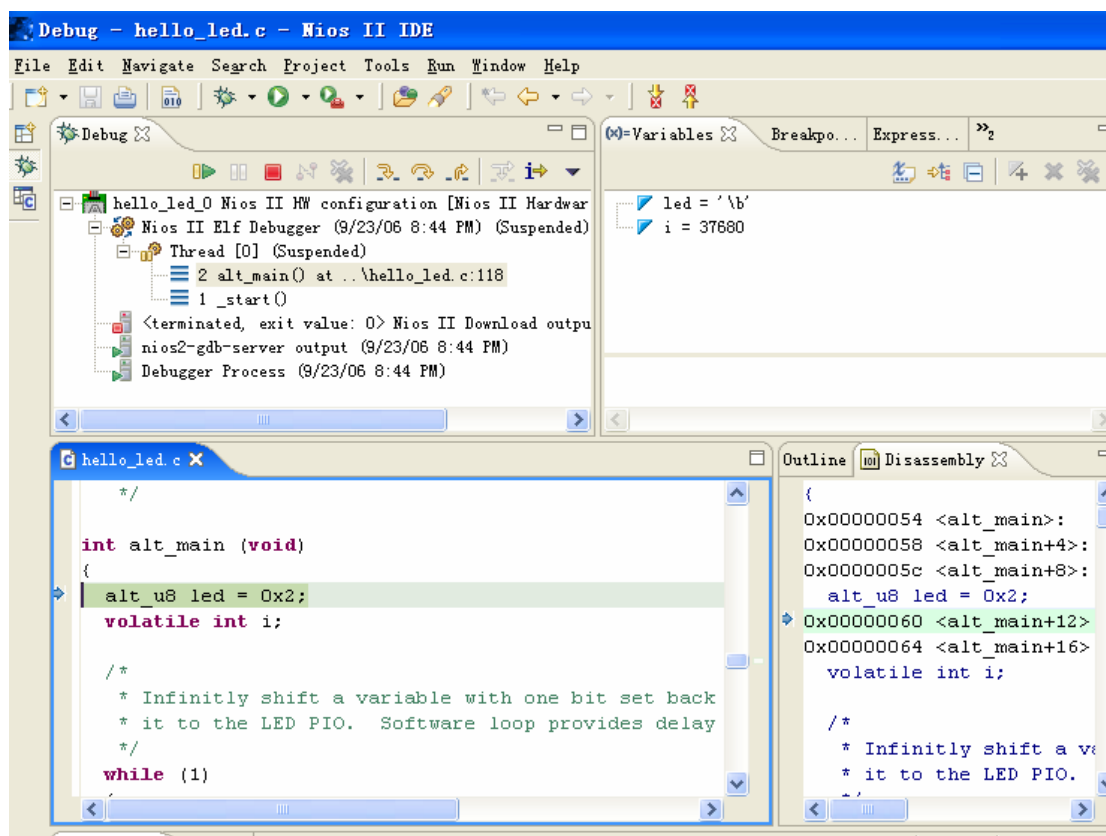
}

return 0;
}

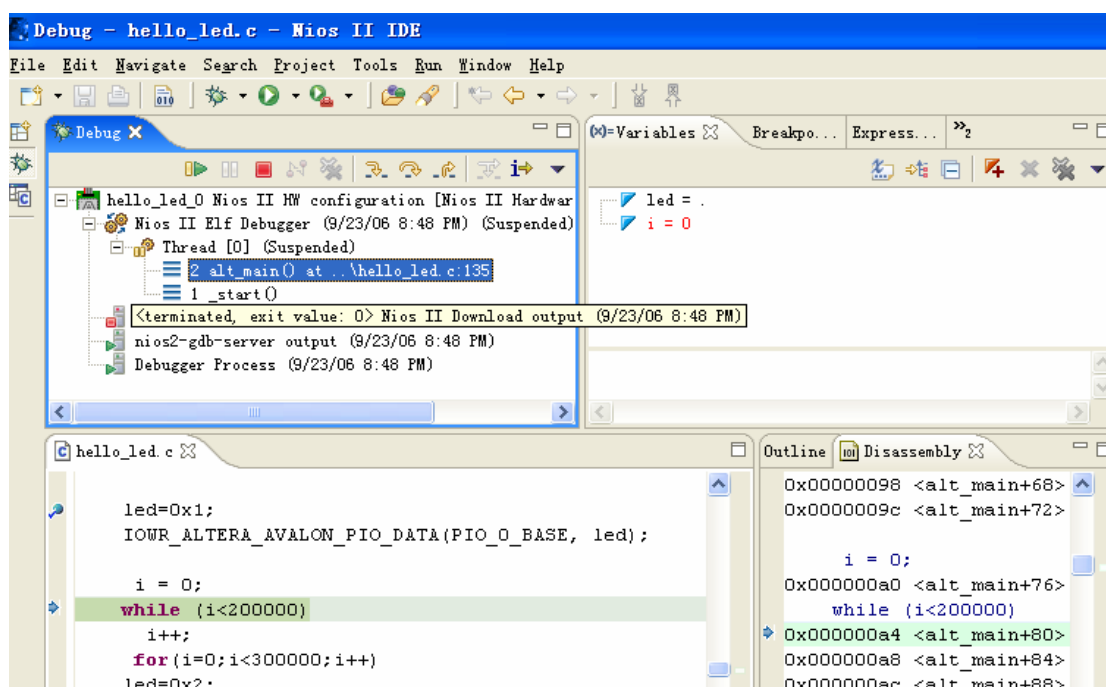
```

编译，并选择 run→run as→nios ii hardware 可以全速运行程序
选择 debug as→niosii hardware 可以调试程序





在调试状态，选择 step into, step over, step return 调试



在右侧变量区可以查看变量。到目前，调试在 jtag 方式，如果断电，程序将丢失，下面把这个程序烧写进开发板。回到 quartus，重新编译原来的项目，选择 AS 方式下载，并运行程序。可以发现 4 个 led 循环闪烁。

FPGA 学习资料:

<http://www.farsight.com.cn/FarsightBBS/dispbbs.asp?boardID=19&ID=483&page=1>

FAQ

1. 怎样在TCL脚本分配管脚 source <pin_assign>.tcl 有点问题:

```
# source stratix_pin_assign.tcl
```

```
couldn't read file "stratix_pin_assign.tcl": no such file or directory
```

答: From the Tools menu select Tcl Scripts, and then from the project folder choose the setup script for your particular development board, and click Run.

2. D:\TEST\nios_sst60 下载时出现错误:

Error: Can't configure device. Expected JTAG ID code 0x020010DD for device 1, but found JTAG ID code 0x020B40DD.

答: SOPC所选器件和开发板上的不一致。

3. 在NOIS II中Bulid例程hello_world都出现了错误, 错误提示为:

```
gdrive/c/altera/kits/nios2/components/altera_nios2/HAL/src/alt_busy_sleep.c:68: error: parse error before '/' token等 错误全部由alt_busy_sleep.c引起, 都是关于括号不匹配的问题, 而alt_busy_sleep.c 是 IDE 中的一个默认程序。这个错误同样出现在D:\TEST\DE2Project\software\nios2, 培训用的一个简单实验。
```

答: 找到system.h文件, 里面有个关于系统时钟频率的设置项, 应该是没有赋值, 你手动赋值。比如你用 50MHZ的时钟, 就设成 50000000

如果总是出现上述问题, 可能和软件有关系, 建议重新安装软件。注意quartus和NIOS安装版本一定要相同,不能混装。

4. 这个错误是什么原因引起, 把那个sdk_arm删除后, 又提示另一个地方出错。

答: 运行NIOS II IDE, 点Project->Clear, 重新Build,应该可以解决问题。

5. 在SOPC中Generate出现如下错误是怎么回事? Error: Generator program for module 'epcs_controller' did NOT run successfully. 只要在SOPC中加入epcs_controller就会出现此错误, 无法生成一个元件。

答: 可能和软件有关系, 建议重新安装软件 (这个问题是我刚学NIOS遇到的最头痛的一个问题, 问题的原因是Quartus和Nios安装的版本不一致)。

6. 在Nios II IDE中, 怎样打开一个已经存在的工程? 每次新建工程比较麻烦。

答: 指定一个Nios II IDE的工作目录, 就打开了那个目录下存在的工程。另外, 在Nios II中是可以建立多个工程的。

7. 在Quartus II 中编译出现如下错误怎么办?

```
Error: Can't place pins assigned to pin location Pin_AE24 (IOC_X65_Y2_N2)
```

答: 按F1 可以查看帮助, 出现这样的错误的原因是:

CAUSE: You assigned two or more pins to the specified location, but the Fitter cannot place all the pins in that location.删除这个管脚即可。有一种比较简便的方法，就是在工程目录中找到一个后缀为QSF的管脚配置文件，查找Pin_AE24 删除那行语句就行了。

8.如何在NIOS II IDE 下跟踪查看变量的定义或者函数的定义？

答：按住CTRL键，鼠标移动到变量或者函数名的地方，就可以发现这些地方高亮显示，单击就可以进入到变量或者函数定义的地方。

9. 在count_binary.c有这样一段程序，它是如何操作的？ `unsigned int data = segments[hex & 15] | (segments[(hex >> 4) & 15] << 8)`

答：segments[hex & 15]显示个位 0~F；(segments[(hex >> 4) & 15] << 8)显示十位数 0~F，个位 0~F，然后十位加一。

10. 这个错误是由什么引起？

答：提示LED_PIO_BASE没有声明，这是因为名字不一致引起的比如，在生成SOPC系统时，双击PIO(Parallel I/O)(在Avalon Modules -> Other 下)，为系统添加输出接口,你没有把该组件改 名 成 LED_PIO, 而是 保 留 了 原 始 的 名 字 : PIO_0 ; 但 你 又 通 过 IOWR_ALTERA_AVALON_PIO_DATA(LED_PIO_BASE, led);来向该组件写入数据，就会导致上述错误。解决办法：1.可以修改sopc系统，为该PIO改名为LED_PIO ;2.在hello_led.c的前面给LED_PIO_BASE赋值，如#define LED_PIO_BASE 0x00001800，后面的这个地址要与SOPC中的地址对应。

11. 如何在NIOS II中驱动外部芯片或设备？

答：在开发过程中，我们经常会使用外接一些芯片，或外接一些实用电路，比如AD芯片、串并转换芯片等等，那么如何在NIOS II中去使用这些芯片呢？我们在开发中会有多个选择：

1. 如果这个芯片是使用数据和地址总线的，并且补线时没有足够的单独引脚，那么这个芯片必须挂接在tri_stat总线上，这种用法和我们以往单片机或MCU类似，在SOPC Builder中直接定义用户逻辑，如果有额外的控制引脚，就需要用PIO来驱动，虽然公用数据地址线节省了引脚，但做控制时序费劲了。
2. 如果这个芯片单独使用数据和地址，那么我们会直接做成Avalone总线的Slave设备，在Sopc Builder中自己定义component。需要写HDL模块，自己定控制、状态、数据寄存器和控制位定义，这种方式控制灵活，易于实现复杂的控制时序。
3. 在Quartus II 工程中画电路模块（或用HDL写电路模块），这种方式是独立于SOPC定义的模块之外的，做一个纯电路的处理模块，有时候会比较容易实现某些灵活的处理。

12.在SOPC添加Avalon Tristate Bridge时，提示有如下错误，该如何解决？

Tri state bridge/tristate master requires a slave of type Avalon tristate.Please add a slave of type Avalon tristate.

Generate按钮为灰色，无法Generate

答：需要一个专门接三台桥的设备，把flash添加到sopc中就可以了。

13. 在做count_binary这个例子时，出现一个错误： error: 'BUTTON_PIO_IRQ' undeclared (first use in this function) BUTTON_PIO_IRQ的值如何给他定义？ 答：这个错误可能是在

sopc builder中定制的pio端口名称是否与程序中用的不一致，要和程序里的一致，把pio组件的名称就改为button_pio，如下图：

如果已经这样设置，还有可能是没有设置中断号，具体应该这样设置：

14.在Quartus II中编译时出现如下错误：

Error: Node instance "cpu_bht" instantiates undefined entity "cpu_bht_module"

点击错误的地方弹出一个对话框

这个错误是怎么回事？怎样解决？

答：可能是在SOPC中所选的CPU型号不正确，换一个试试。

15.怎样自动分配管脚？

答：在Quartus II自动分配管脚有一个方法，点击Assignments->Import Assignments,导入一个管脚分配文件，后缀为.csv .txt等，前提是在顶层原理图中各个输入输出的名称要和这个文件中的名称一致。

16.在NIOS II中编译时出现如下错误怎么解决？错误是不是由SOPC中的RAM引起？

region ram is full (count_binary.elf section .text). Region needs to be 24672 bytes larger.

address 0x80c1f8 of count_binary.elf section .rwddata is not within region ram

Unable to reach edge_capture (at 0x00800024) from the global pointer (at 0x0081419c) because the offset (-82296) is out of the allowed range, -32678 to 32767.

答：可能时RAM的大小不够，也有可能是中断地址(exception address) 的偏余量不够，设置大些就可以了。如果还是出现这个问题，加一个SDRAM试试。

17. 在Quartus II中，为什么编译一些样板工程都会出错？

Error: DDR timing cannot be verified until project has been successfully compiled.

Error: Evaluation of Tcl script auto_verify_ddr_timing.tcl unsuccessful

Error: Quartus II Shell was unsuccessful. 2 errors, 1 warning

答：在做样板工程时要注意一个问题：Due to the library paths that are coded into the Quartus settings for this project, if a user wishes to modify the hardware design they must first strip out any old paths from within the project settings file (qsf)

在qsf文件中找到类似语句

set_global_assignment-nameVHDL_FILE

C:/MegaCore/ddr_ddr2_sdram-v3.2.0/lib/auk_ddr_tb_functions.vhd"这里的目录是初始的目录，把它改为正确的目录。

18.在NIOS II IDE编译时出现如下错误是怎么回事？

system_description/alt_sys_init.c:75: error: 'ONCHIP_MEMORY_BASE' undeclared here (not in a function)

system_description/alt_sys_init.c:75: error: initializer element is not constant

system_description/alt_sys_init.c:75: error: (near initialization for `ext_flash.dev.write')

system_description/alt_sys_init.c:75: error: initializer element is not constant

system_description/alt_sys_init.c:75: error: (near initialization for `ext_flash.dev.read')

答：ONCHIP_MEMORY_BASE 没有赋值，在 alt_sys_init.c 程序的开头加上 #define ONCHIP_MEMORY_BASE 0x00000000 后面的这个地址要与SOPC中的对应。

19. 在NIOS II IDE编译时出现如下错误是怎么回事？

Pausing target processor: not responding.

Resetting and trying again: FAILED

Leaving target processor paused

答：以下是一位FAE的回答：

1.关于USB-Blaster在Nios II IDE下载时会发生偶发性错误，这种现象主要是IDE software与Nios II CPU透过USB-Blaster在做通讯时发生错误，若是确认FPG上配置没有错误，连续发生错误的机率应该是相当的低，您只需要重新下载即可。

2. 若您使用Nios II IDE 6.0，请尽量配合SOPC Builder 6.0 重新build您的system，并且使用Quartus II 6.0 重新compile您的project，以减少CPU与IDE software不兼容的情形。

20. 在 NIOS II IDE 中 工程 的 System Library 选项中的这几个选项代表什么意思？.text .rodata .rdata 与reset .exception这几个地址之间的关系是什么？

答：.text：代码区 .rodata:只读数据区，一般存放静态全局变量 .rdata:可读写变量数据区另外还有.bss:存放未被初始化的变量。

- .text — the actual executable code
- .rodata — any read only data used in the execution of the code
- .rdata — where read/write variables and pointers are stored
- heap — where dynamically allocated memory is located
- stack — where function call parameters and other temporary data is stored

21. 怎样在NIOSII中操作PIO，提供一种参考方法。

答：hello_led.c是这样写IO口的：

IOWR_ALTERA_AVALON_PIO_DATA(LED_PIO_BASE, led);

首先在altera_avalon_pio_regs.h找到定义

#include <io.h>

#define IORD_ALTERA_AVALON_PIO_DATA(base) IORD(base, 0)

#define IOWR_ALTERA_AVALON_PIO_DATA(base, data) IOWR(base, 0, data)

因此在NIOSII中可以调用#include <io.h>库函数IORD/IOWR来操作PIO。

在small\software\hello_led_0_syslib\Debug\system_description下的system.h

中,有以下内容:

#define LED_PIO_TYPE "altera_avalon_pio"

#define LED_PIO_BASE 0x00004000

其中LED_PIO_BASE（IO寄存器地址？）为 0x00004000 同SopcBuilder中设置一致！

(其实在SopcBuilder中有关NiosII的配置,就是通过system.h来传送给IDE的!)

最后用IOWR(0x00004000, 0, led);替代

IOWR_ALTERA_AVALON_PIO_DATA(LED_PIO_BASE, led);编译,下载到开发板上,运行成功!

22.怎样让SDRAM和FLASH, SRAM的地址公用?

答: sdram可以和sram,flash共用数据总线和地址总线.在sopc builder中添加SDRAM控制器时,在其share pins via tristate选择项上勾上Controller shares dq/dqm/addr IO pins选项,这样sdram可以和sram,flash共用数据总线和地址总线.但是ATERA不推荐这样做,因为这样会降低SDRAM控制器的性能,在FPGA芯片管脚资源比较紧张的时候迫不得已才这样做。

23.怎样使用DE2 板上的数码管, 绑定管脚时需要注意什么? 请举一个例子说明。

答: DE2 板上的数码管只用了 7 位, 小数点位默认为高电平。例如, 在做例程count_binary 时, 需要用到两个数码管, 以实现 00 到FF的计数, 在SOPC添加一个 16 位的PIO SEVEN_SEG[15..0], 绑定管脚时注意: 使SEVEN_SEG[0..6]绑定到数码管HEX0[6..0]位, SEVEN_SEG[8..14]绑定到数码管HEX1[6..0]位, SEVEN_SEG[7]和SEVEN_SEG[15]直接接VCC, 这样才能让两个数码管正常显示并计数。

24. 在NIOS II IDE 中调试, 编译通过的软件时, 出现了下面的提示, 是什么原因?

Using cable "ByteBlasterII [LPT1]", device 1, instance 0x00

Processor is already paused

Downloading 00000000 (0%)

Downloaded 57KB in 1.2s (47.5KB/s)

Verifying 00000000 (0%)

Verify failed

Leaving target processor paused

答: Verify failed这个问题说明板子的复位电路可能有问题, 或装载程序的外部SRAM或SDRAM和CPU 的连接或时序有问题,也有可能是软件偶尔的错误。若是软件偶尔的错误,可以先复位一下CPU,然后再下载程序;或者断电后重新下载.sof和NIOS II 程序

25.在练习C:\altera\kits\nios2_51\examples\verilog\niosII_cycloneII_2c35 的工程时, 里面有很多元件DE2 板上都没有对应的管脚, 比如网络元件lan91c111, 使用的IP核, 还有 24 位的ext_flash, 而DE2 板上的FLASH只有 22 位。是不是SOPC中只能让DE2 板上有对应管脚的元件存在, 其它的只能删除?

答: 这个目录下的example是针对altera自己的板子的, 和DE2 硬件不能对应, 只能作为参考, 或者做一些修改后用。

26.在SOPC中加了一个 200KB的onchip_memory ,为什么在Quartus II 编译时出现这个错误?

Error: Selected device has 105 RAM location(s) of type M4K RAM. However, the current design needs more than 105 to successfully fit

答: SOPC中的onchip_memory和M4K RAM根本就不是一个概念。Quartus II中编译出现这个错误, 是由于设计中用到了太多的M4K。

27. 关于sopc-builder 中reset address 的设置, 一直搞得不是很明白。

答: SOPC中的reset address 指定的是最终全部软件程序代码下载到的地方, 并且程序从reset address 启动。

SOPC中的exception address 指定的是系统异常处理代码存放的地方。如果exception address 和reset address 不一样, 那么程序从reset address 启动后将把放在reset address 处的系统异常处理代码拷贝到exception address 。

NIOS II软件中的text address指定的是程序运行的地方。如果text address和reset address 不一样,那么程序从reset address 启动后将把放在reset address 处的普通只读程序代码拷贝到text address 。NIOS II软件中的rodata address指定的是只读数据的存放地方。如果rodata address和reset address 不一样,那么程序从reset address 启动后将把放在reset address 处的只读数据拷贝到rodata address 。

NIOS II软件中的rwd data address指定的是可读写数据的存放地方。如果rwd data address和reset address 不一样,那么程序从reset address 启动后将初始化rwd data address 处的可读写数据。

28. 如何提高NIOS II系统的性能?

答: 主要可以从这几个方面入手:

- 1、使用fast CPU类型。
- 2、提高系统主频。
- 3、优先在SRAM中运行程序, SDRAM次之, 最后选择FLASH中运行。
- 4、使用片内RAM作为数据缓冲, 片外SRAM次之, 最后选SDRAM。
- 5、IO数据传输尽可能采用DMA。
- 6、对能并行处理的数据考虑使用多CPU协同处理。
- 7、典型算法做成用户指令, 有 256 条可以做, 足够你用的。
- 8、能用HDL模块来完成工作吗? 能, 就用HDL模块做成外设来完成吧
- 9、采用C2H。

29.如何优化NIOSII里的应用工程?

答: Optimize your Nios II application design!

1.Creat an Blank Project: Select Altera Nios II " C/C++ Application"

2. In main.c[your main fuctional c/c++ file]:

```
#include "system.h"
#include "alt_types.h"
#.....
int main (void) __attribute__ ((weak, alias ("alt_main")));
int alt_main (void)
{
.....
return 0;
}
```

3."System Library Properties" Options

Select "Clean EXIT"

Select "Small C Library"

Select "Reduced Device Drivers"

Spicify linker script position, to ext_flash, ext_ram or on-chip-ram

30. 关于verify failed的总结

1. SDRAM的时序不对

有时候不正确的pll clock phase shift for sdram_clk_out就会导致SDRAM不能正常工作:

2. SDRAM的连线不对,物理板子的连线问题

3. 在调试的时候,程序下载的空间不是非易丢失存储器(non-volatile memory)或者存储器的

空间不够也会导致这个错误

4. QuartusII的默认设置导致的错误

QuartusII默认将所有没有使用的IO口接地,这种时候可能导致某些元器件工作不正常;最好将不用的IO口设置为三态

5. USB-blaster坏了,或者JTAG通信的信号噪声太大

JTAG的端口需要一个弱上拉电阻来抗干扰

6. 确保你的sdram既连接到CPU的指令总线也连接到CPU的数据总线

31.关于DMA传输的几点说明:

1. memory 到 外设的传输,调用alt_dma_txchan_ioctl()时,有一个参数为alt_dma_tx_only_on等
2. 调用alt_dma_txchan_send函数时,在传输结束前就返回一个值,如果此值为负的话,说明发送请求失败。正确传输结束后,调用done函数。
3. 接收函数alt_dma_rxchan_prepare类似上面的1和2
4. 传输结束,有两种可能:数据传完或者end of packet(要预先使能)
5. Sopc builder中例化时要制定哪些可以访问DMA的主端口, DMA的avalon slave端口要接cpu。实际传输的最大数可以帮助确定设置的位数
6. alt_dma_txchan_ioctl用于控制dma的一些工作性质,使用多的话可以用信号量等来“抢占”
7. dma传输最小应该传4字或者其倍数。

32. 1. 建立CPU时,下面的那个HardWare Multiply里面可以选①Embedded Multipliers, ②Logic Elements, ③None,这三个选项有什么区别呀?

答: ①Embedded Multipliers,使用专门的内嵌硬件乘法单元(不可编程,仅能做乘法,且乘法速度最快),不是RAM。 ②Logic Elements,使用逻辑单元也就是FPGA中的查找表(速度较慢)。 ③None,那就是不要声长硬件乘法器了,这是只能通过软件模拟乘法,速度最慢。

33.下载是出现“not responding”错误的又一点发现:

答:在设计的过程中,经常使用板子上内部的两个晶振作为CPU或者有些信号的时钟,而且习惯用osc_27和osc_50命名。在绑定管脚时又经常导入一个CSV文件,没有手动一一绑定,而在那个CSV文件中晶振管脚名分别为CLOCK_27,CLOCK_50,这样系统的CPU根本就没有绑定管脚,不出错才怪。把名称改为CLOCK_27,CLOCK_50,重新编译下载就可以了。

34. 安装了nios2linux-1.4之后,为什么在New->Project时并没有出现Microtronx NIOS II选项呢?

答:这个问题好像难倒过很多人,其实在运行nios2linux-1.4安装的第一步,里面有个提示: Information Regarding the Installation Procedure

IMPORTANT: Please ensure that you specify the correct path for the Altera Nios II Kit installation directory and the Cygwin root directory. The typical Altera Nios II Kit directory is:c:\altera\kits\nios2

The typical Cygwin root directory is:

c:\altera\quartus50\bin\cygwin

问题的关键就是上面的路径不正确。对于Nios 5.1和6.0的路径分别为nios_51和nios_60,都不是nios2,所以安装后找不到模板工程很正常。但它的一些必要文件又确实是安装在了nios_51或nios_60中。我也遇到了过同样的问题,想到了一个简单的方法可以解决:

- 1.把文件夹nios_51 或nios_60 改为nios2, 启动Nios II, 就可以看到所需模板工程。
- 2.退出Nios II, 把文件夹名还原。
- 3.再次启动Nios II, 你会发现模板工程依然还在。

35. Nios II汉化解决方案

答: NIOS II IDE实际上是在eclipse平台上的一个应用插件, 而Eclipse 是一个开放源代码的软件开发项目, 专注于为高度集成的工具开发提供一个全功能的、具有商业品质的工业平台。可访问如下网站:

- 1.Eclipse官方网站<http://www.eclipse.org/>
- 2.中国Eclipse社区 <http://www.eclipseworld.org/bbs/index.php>

下面介绍如何汉化NIOS II (实际上只是汉化了eclipse)

- 1.打开NIOS II, 点help->about NIOS II IDE,会看到有关NIOS的版权信息。可以看到NIOS II 5.1 的版本上的Eclipse版本是Eclipse 3.0.1

- 2.在Eclipse官方网站上找到Eclipse 3.0.1 多国语言包NLpack-eclipse-SDK-3.0.x-win32.zip

下载地址:

[http://sunsite.informatik.rwth-a ...](http://sunsite.informatik.rwth-a...) SDK-3.0.x-win32.zip

- 3.在NIOS II的安装目录中, 你可以找到一个eclipse目录, 如: C:\altera\kits\nios2\bin\eclipse, 在这个目录下新建 2 个文件夹language和links

- 4.将NLpack1-eclipse-SDK-3.1.1a-win32.zip解压到language目录下

- 5.在links文件夹下用记事本新建一个文件,取名为link.link(必须是此扩展名)

- 6.在language.link 里输入代码如下: path=c:\altera\kits\nios2\bin\eclipse/language 保存此文件。

- 7.汉化完成。

说明, 这种方法只是汉化了eclipse, 对NIOS II没有汉化, 但这个软件已经大部分为中文了, 因为NIOS II 只是eclipse一个插件。对于我们新手来说这还是必要的。