

NiosII-LED-Demo

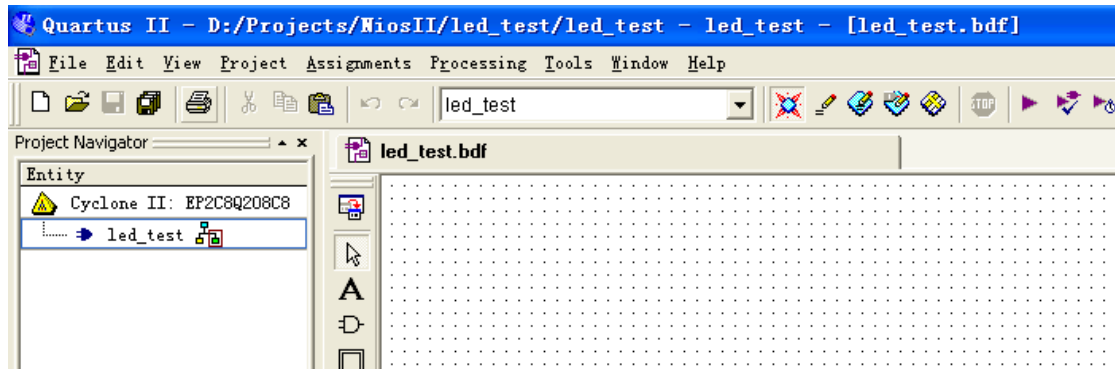
环境:

QuartusII 9.0

NiosII IDE 9.0

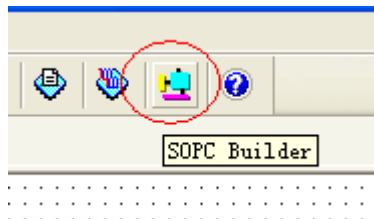
(注意, NiosII IDE 不要安装在带空格或汉字的文件夹下, 否则会出错)

1.新建工程, led_test

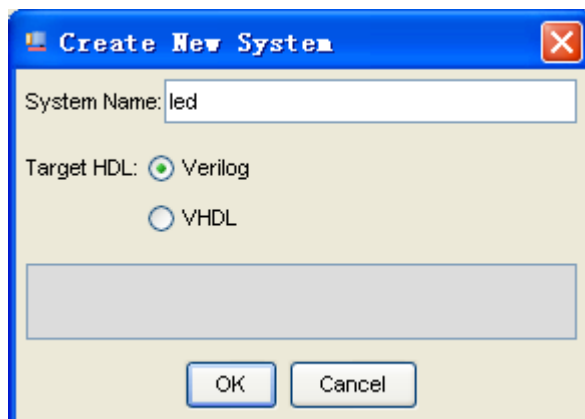


我这里是放在 `D:/Projects/NiosII/led_test` 文件夹下的。

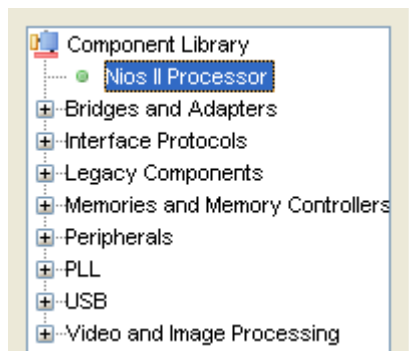
2.点击 SOPC Builder 按钮



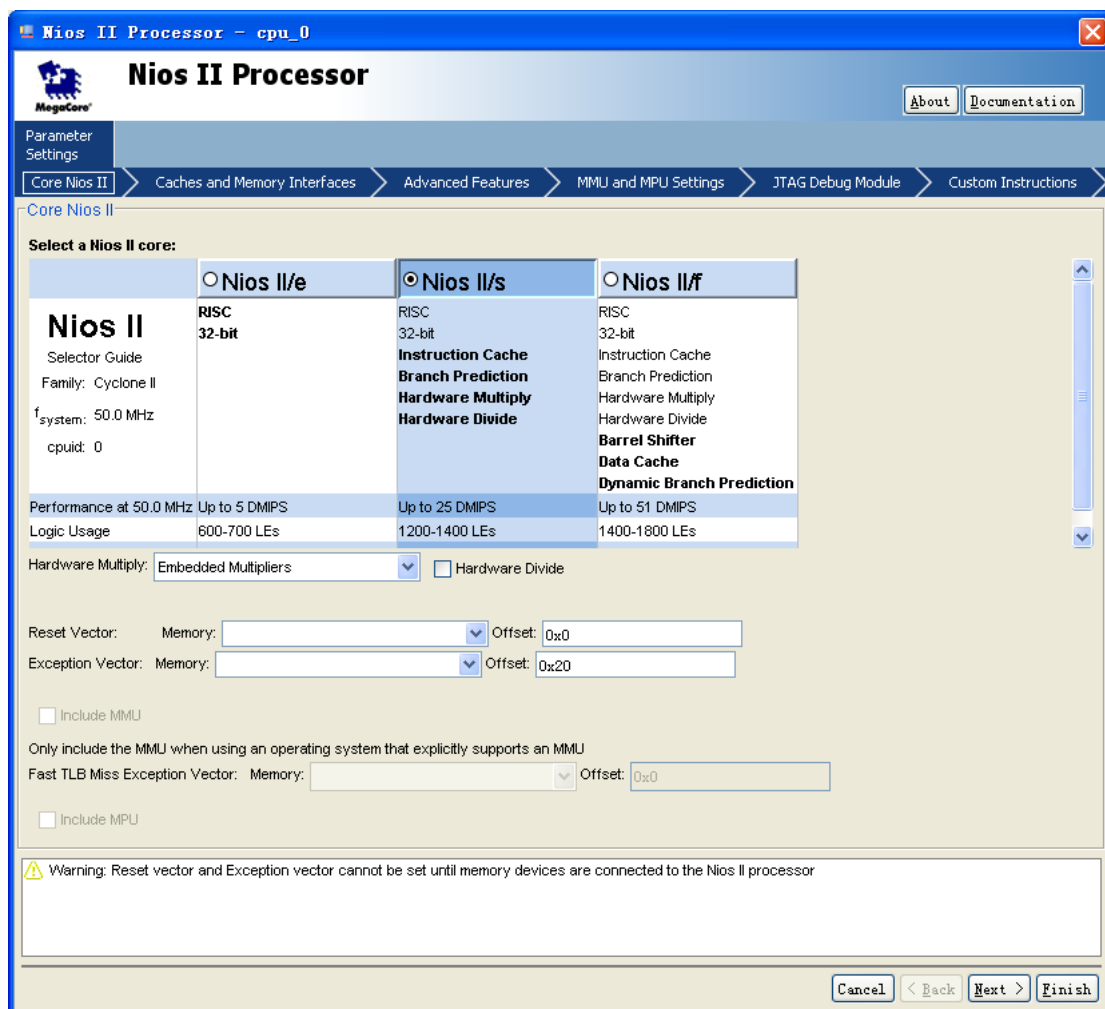
3.新建一个 System, 注意名称不要与工程名 led_test 相同!



4.首先添加 CPU ， 双击图中的 Nios II Processor.



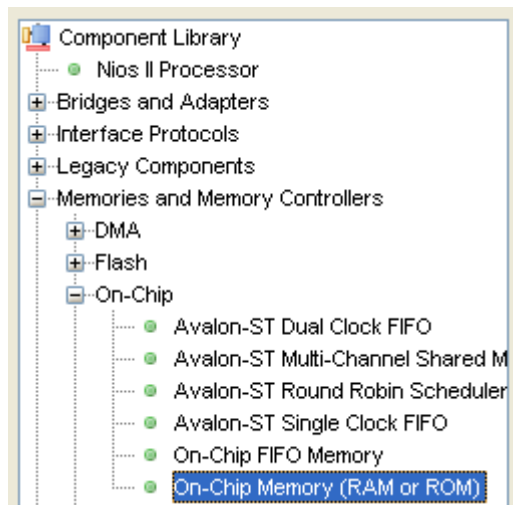
5.选择 CPU 类型为标准型 (s)， 其它的选项默认， 然后 Finish。



6.重命名 CPU， 将名称中的 “_0” 去掉。

Use	Con...	Module Name	Description
<input checked="" type="checkbox"/>		cpu	Nios II Processor
		instruction_master	Avalon Memory Mapped Master
		data_master	Avalon Memory Mapped Master
		jtag_debug_module	Avalon Memory Mapped Slave

7.添加片上 ROM，用来存放程序（重命名为 onchip_ROM）



Memory type

☐ RAM (Writable) ☒ ROM (Read-only)

☐ Dual-port access

Read During Write Mode:

Block type:

☒ Initialize memory content

Memory will be initialized from onchip_ROM.hex

Size

Data width:

Total memory size:

☐ Minimize memory block usage (may impact fmax)

Read latency

Slave s1: Slave s2:

Memory initialization

☒ Enable non-default initialization file

User-created initialization file: .hex

☐ Enable In-System Memory Content Editor feature

Instance ID:

8.添加片上 RAM(重命名为 onchip_RAM)

Memory type

☒ RAM (Writable) ☐ ROM (Read-only)

☐ Dual-port access

Read During Write Mode: DONT_CARE

Block type: Auto

☒ Initialize memory content

Memory will be initialized from onchip_memory2_0.hex

Size

Data width: 32

Total memory size: 4096 Bytes

☐ Minimize memory block usage (may impact fmax)

Read latency

Slave s1: 1 Slave s2: 1

Memory initialization

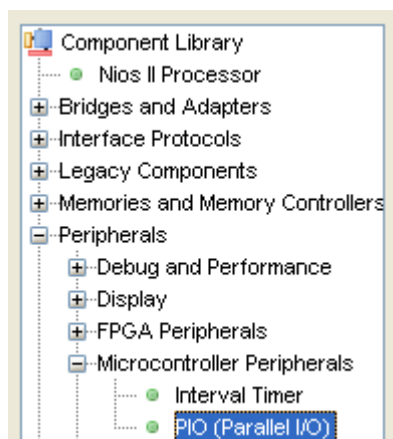
☐ Enable non-default initialization file

User-created initialization file: onchip_memory2_0.hex

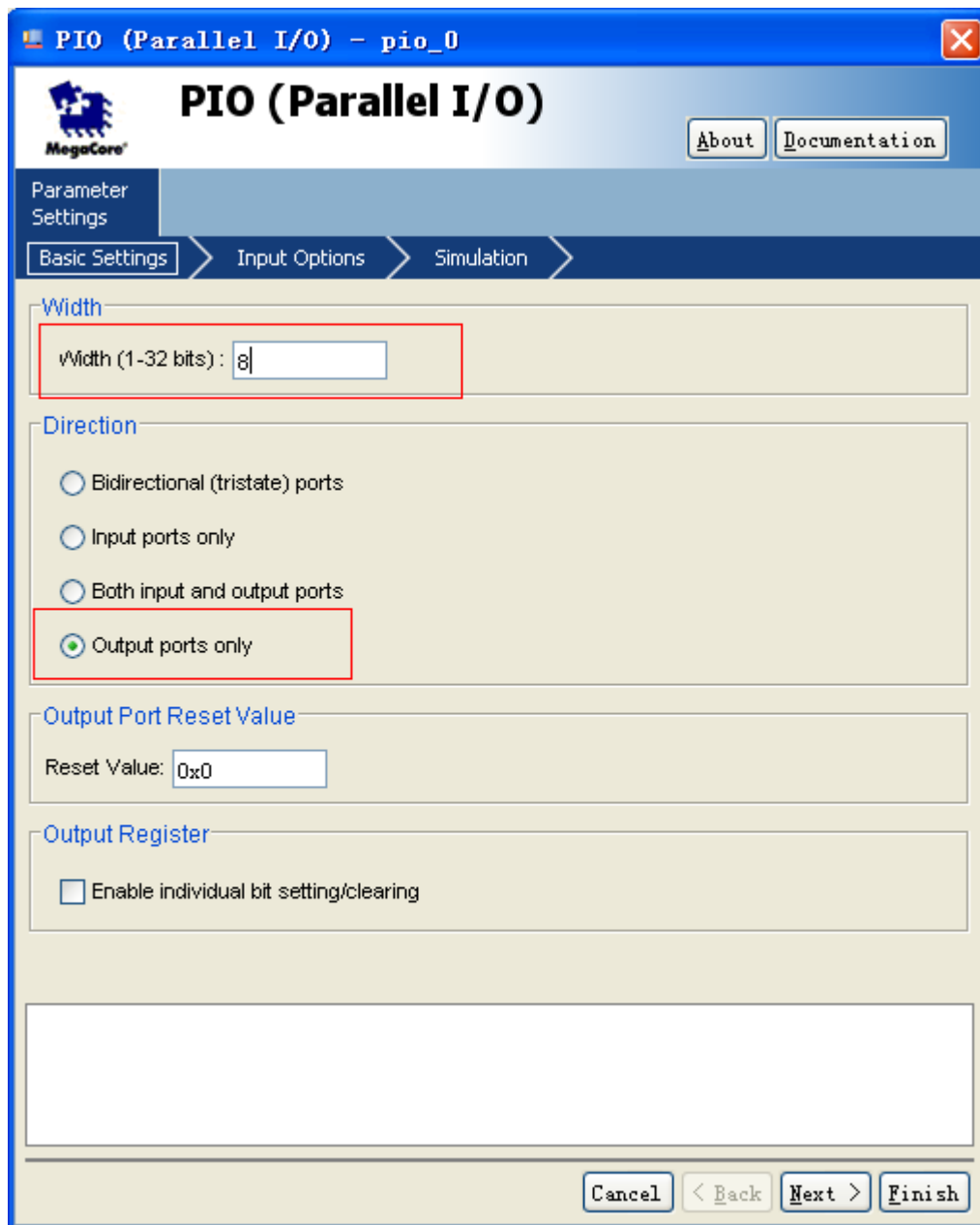
☐ Enable In-System Memory Content Editor feature

Instance ID: NONE

9.添加 PIO



10.选择 PIO 的位宽和接口方向，如下图。并将其重命名为 LED_PIO。



11.设备都添加成功了，应该如下图所示。

Use	Con...	Module Name	Des
<input checked="" type="checkbox"/>		<input type="checkbox"/> cpu	Nios I
		instruction_master	Avalc
		data_master	Avalc
		jtag_debug_module	Avalc
<input checked="" type="checkbox"/>		<input type="checkbox"/> onchip_ROM	On-C
		s1	Avalc
<input checked="" type="checkbox"/>		<input type="checkbox"/> onchip_RAM	On-CI
		s1	Avalc
<input checked="" type="checkbox"/>		<input type="checkbox"/> LED_PIO	PIO (F
		s1	Avalc

12.重新来设置 CPU 的选项，将 Reset Vector 设为 onchip_ROM，将 Exception Vector 设为 onchip_RAM。

Select a Nios II core:

	<input type="radio"/> Nios II/e	<input checked="" type="radio"/> Nios II/s	<input type="radio"/> Nios II/f
Nios II	RISC 32-bit	RISC 32-bit Instruction Cache Branch Prediction Hardware Multiply Hardware Divide	RISC 32-bit Instruction Cache Branch Prediction Hardware Multiply Hardware Divide Barrel Shifter Data Cache Dynamic Branch Prediction
Selector Guide			
Family: Cyclone II			
f _{system} : 50.0 MHz			
cpuid: 0			
Performance at 50.0 MHz	Up to 5 DMIPS	Up to 25 DMIPS	Up to 51 DMIPS
Logic Usage	600-700 LEs	1200-1400 LEs	1400-1800 LEs

Hardware Multiply: ☐ Hardware Divide

Reset Vector: Memory: Offset: 0x00002000

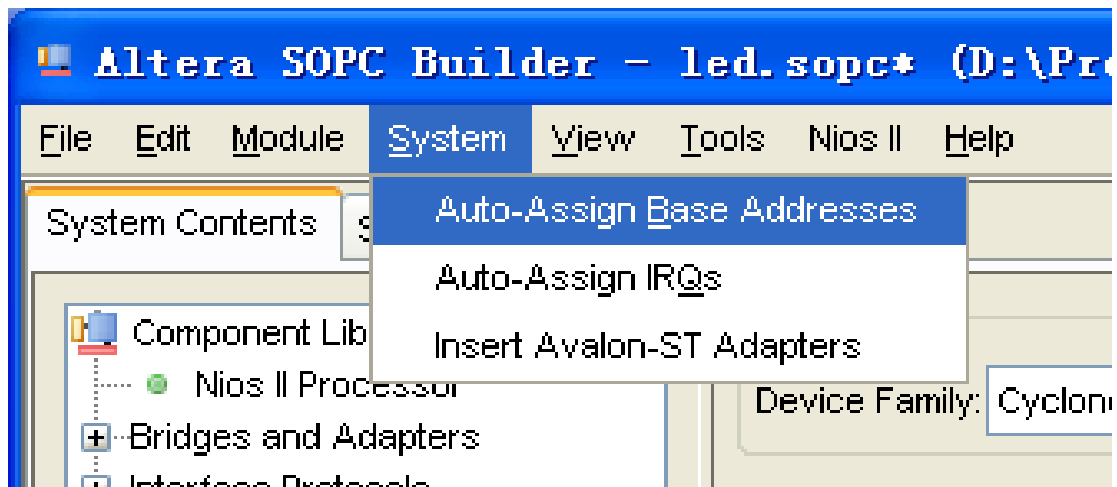
Exception Vector: Memory: Offset: 0x00003020

☐ Include MMU

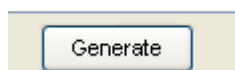
Only include the MMU when using an operating system that explicitly supports an MMU

Fast TLB Miss Exception Vector: Memory: Offset:

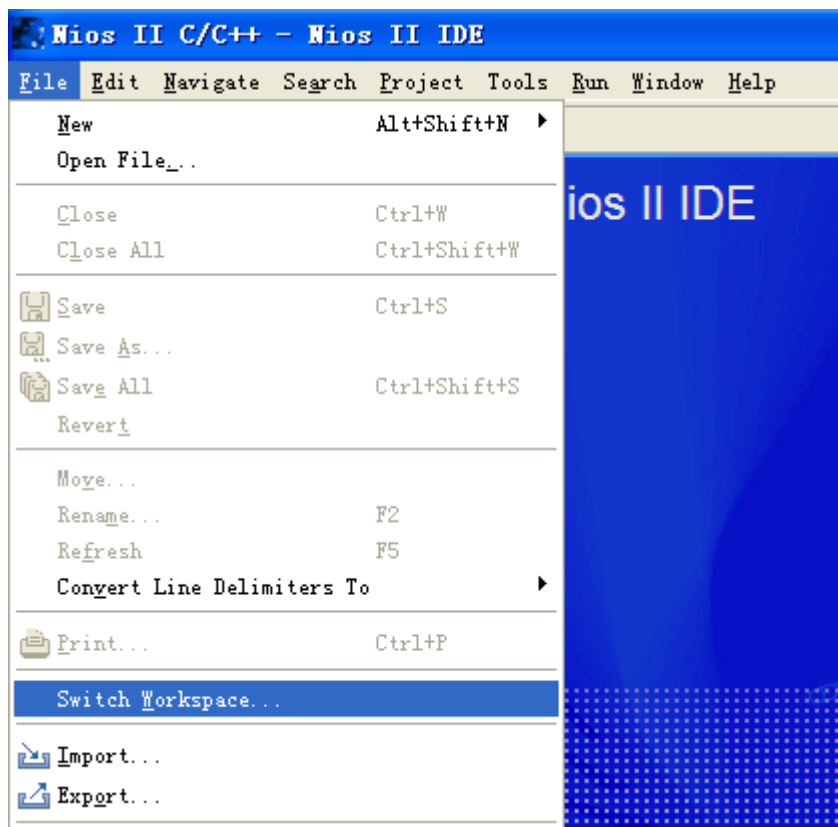
13. 点击运行 System 下的 Auto-Assign Base Addresses 和 Auto-Assign IRQs。

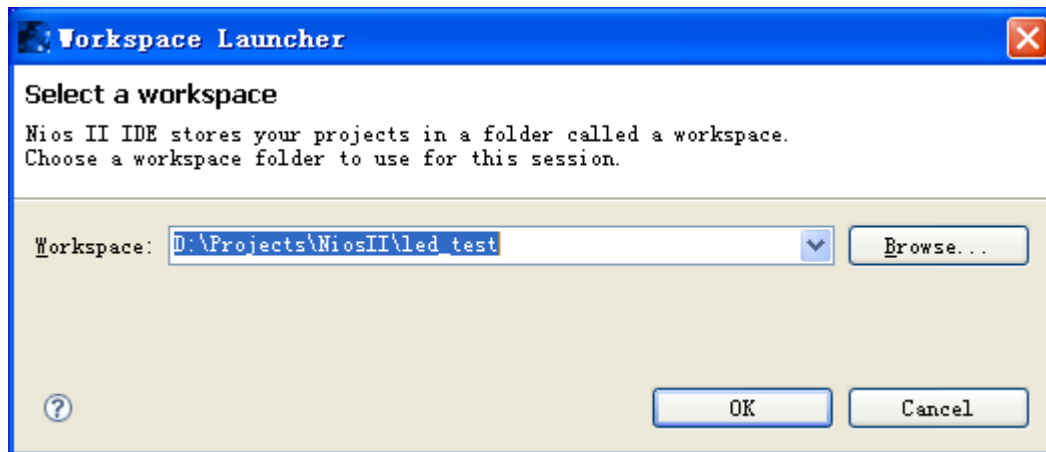


14. 点击软件下方的 Generate，生成 NiosII 核。
(这个时间比较长，2~5 分钟)

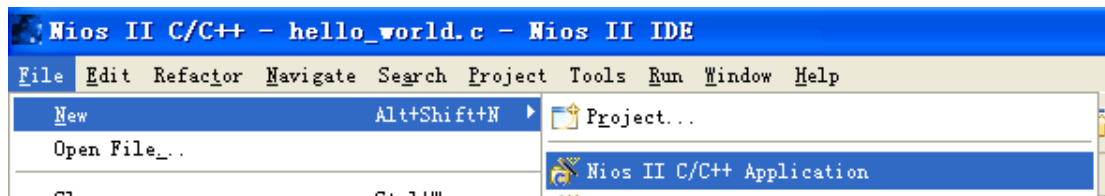


15. 启动 NiosII IDE，将工作目录切换到当前工程所在文件夹下。(D:/Projects/NiosII/led_test)

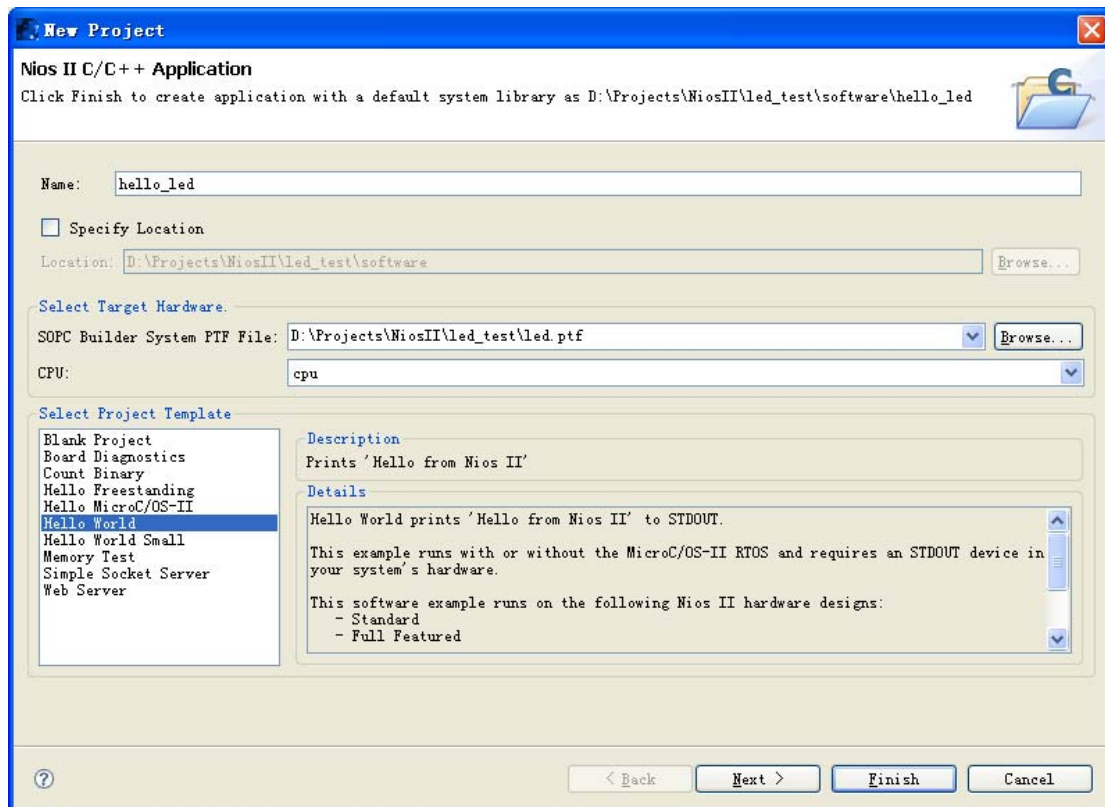




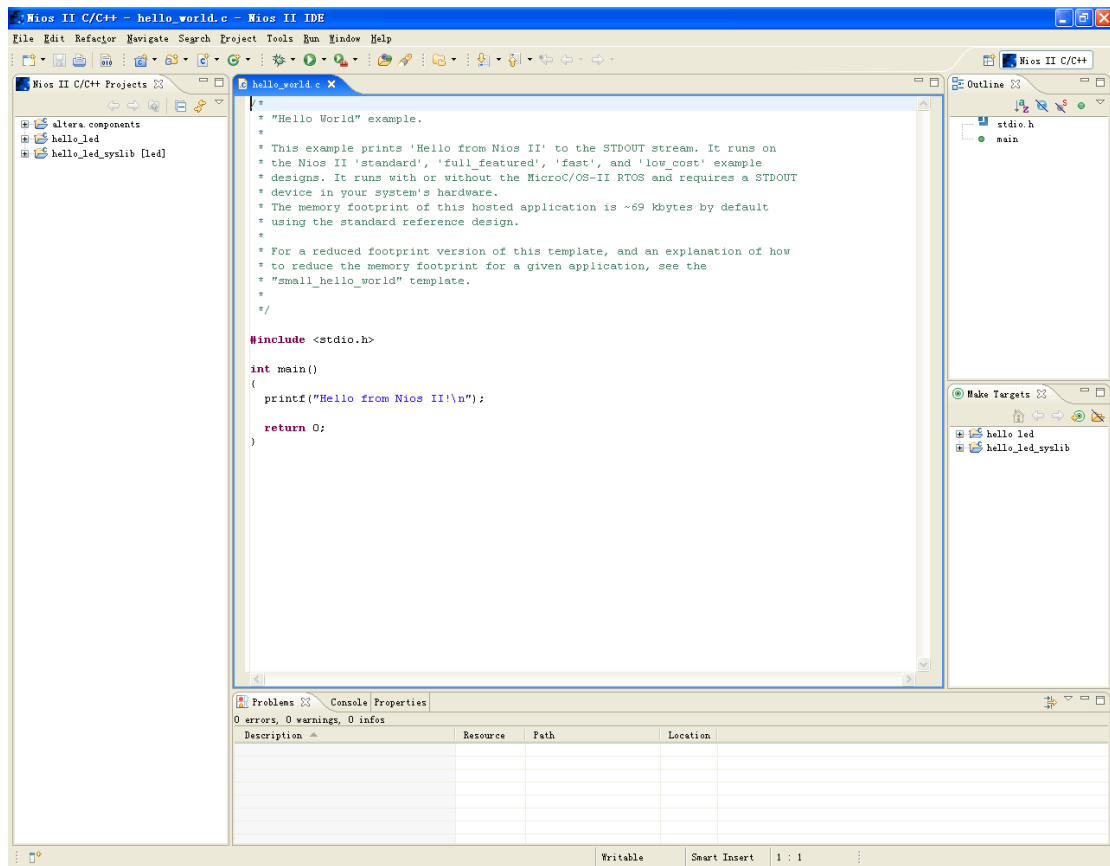
16.选择 File -> New -> NiosII C/C++ Application，新建应用程序。



17.选择“Hello World”模板，注意 PTF 文件应该是在 SOPC Builder 中生成的 led.ptf 文件。



18.新建完成后的界面应该如下图所示。



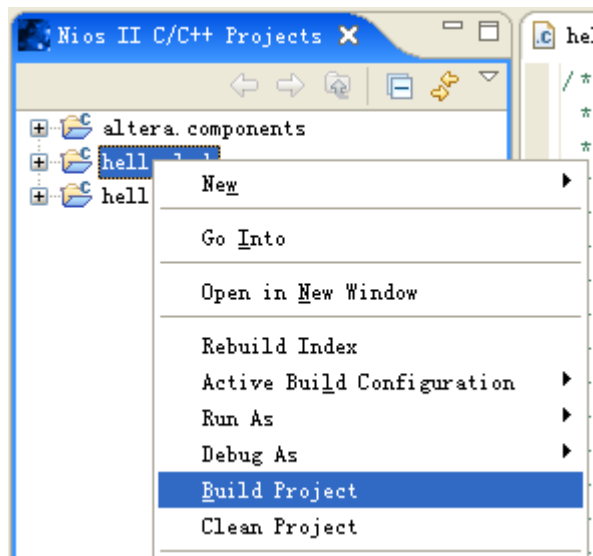
19.将 hello_world.c 中原来的程序删除，输入下面代码：

```
#include "stdio.h"
#include "system.h"
#include "altera_avalon_pio_regs.h"
#include "alt_types.h"

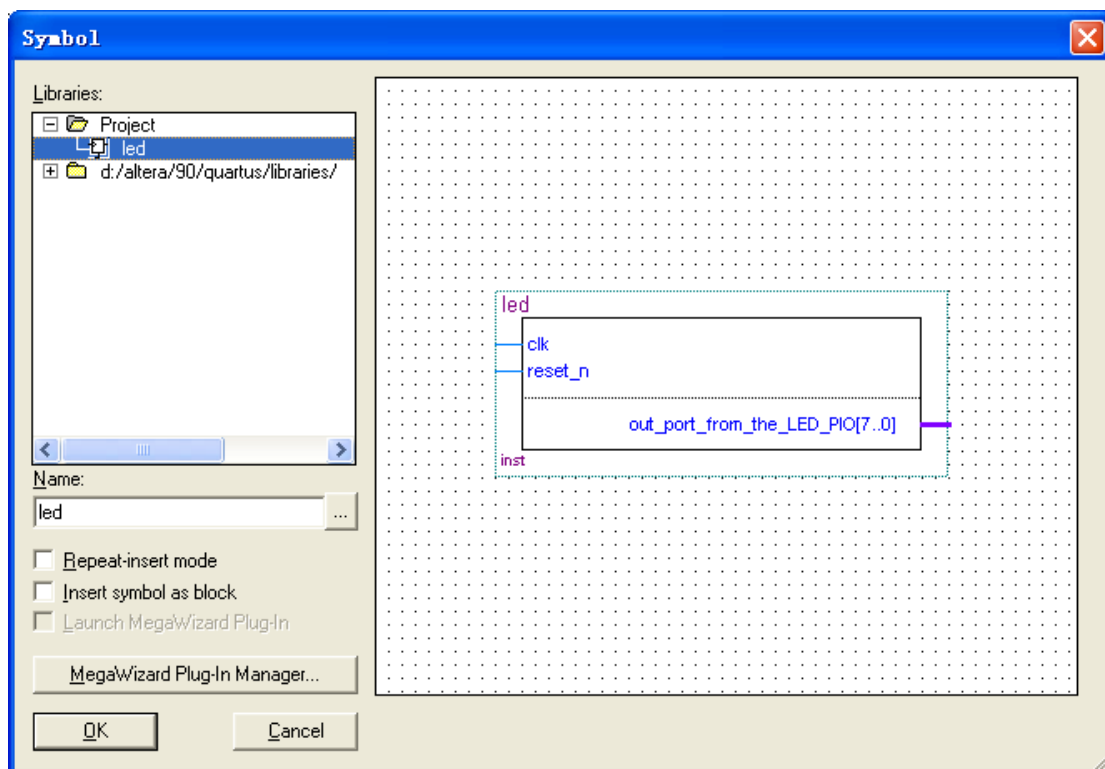
int main(void) __attribute__((weak, alias("alt_main")));

int alt_main(void)
{
    alt_u8 led = 0;
    alt_u32 i;
    while(1)
    {
        IOWR_ALTERA_AVALON_PIO_DATA(LED_PIO_BASE, led);
        for(i=0; i<500000; i++);
        led = ~led;
    }
}
```

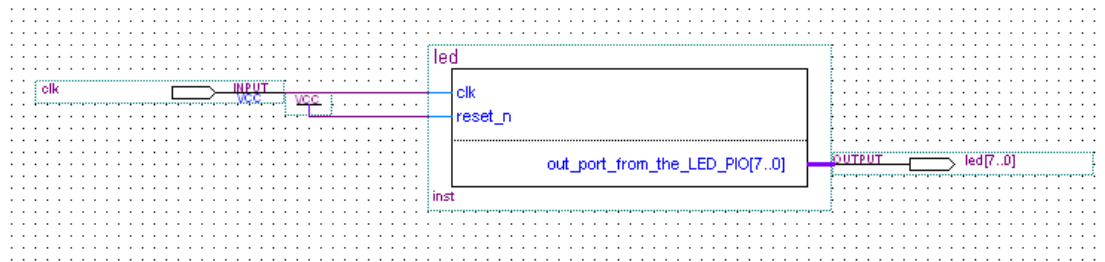
20.在 hello_led 工程上点右键，然后 Build Project。（这个过程也比较长，1~3 分钟）



21.回到 QuartusII 中，在顶层原理图中将生成的 NiosII 核添加进来。



22.添加 input 和 output



23.配置引脚，编译工程，并通过 JTAG 下载 led.sof 文件到 FPGA。

24.在 NiosII IDE 中，点击 hello_led 工程右键，Run as -> Nios II Hardware 。稍后就可以看到开发板上的 LED 闪烁。

