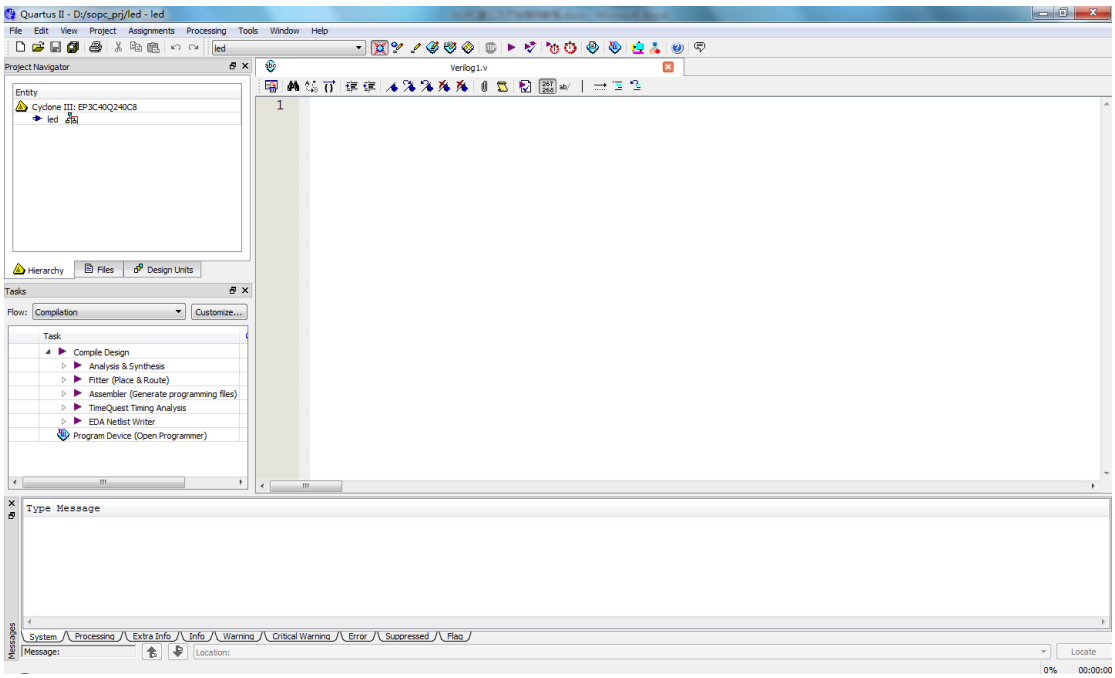
SOPC 建立与开发简明教程

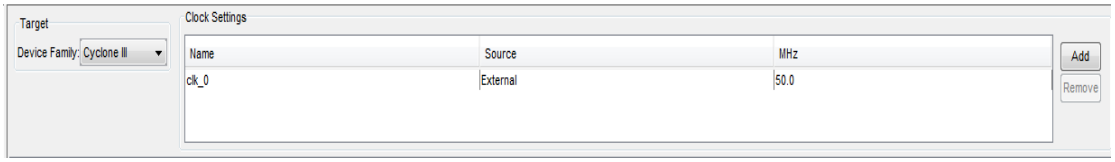
1， 新建一个 Quartus II 工程，并且新建一个 Verilog HDL 文件，本例程设置工程名为 sopc_prj，如下：



2， Tools->SOPC Builder,如图：

3， 设置系统名称和需要的硬件描述语言，本例程设置系统名称为 nios_cpu ，硬件描述语言选择 Verilog。

4， 正确设置器件系列和时钟频率，如下图：



5，添加组件。

- NIOS II CPU

  Library->processers->nios ii processer，选择 NIOS II CORE 等级为 NIOS II/f 全功能型；Debug Level 选为 Level 2，其它设置保持默认。右键点击生成的 CPU_0，Rename 为 CPU

- JTAG UART

  Library->Interface Protocals->serial->JTAG UART,设置默认，点击 Finish 确认添加。右键点击生成的 JTAG UART，Rename 为 jtag_uart

- On-Chip Ram

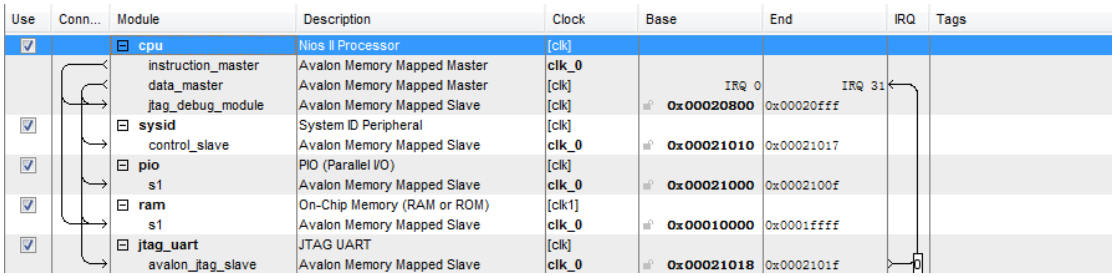
  Library->Memory and Memory Controller->On Chip->On Chip Memory,把 Total Memory Size,改为 65536。右键点击生成的 on-chip Ram，Rename 为 ram

- 添加 PIO

  Library->periperals->microcontroller periperals->PIO，Width 设置为 4，Direction 设置为 Biri（三态）。

- 添加 system ID。Library->peripherials->system ID peripherial,设置默认。右键点击生成的 sysid_0，Rename 为 sysid

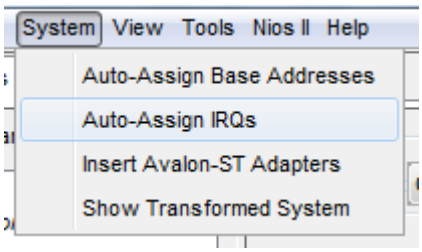
  添加的组件如下图:



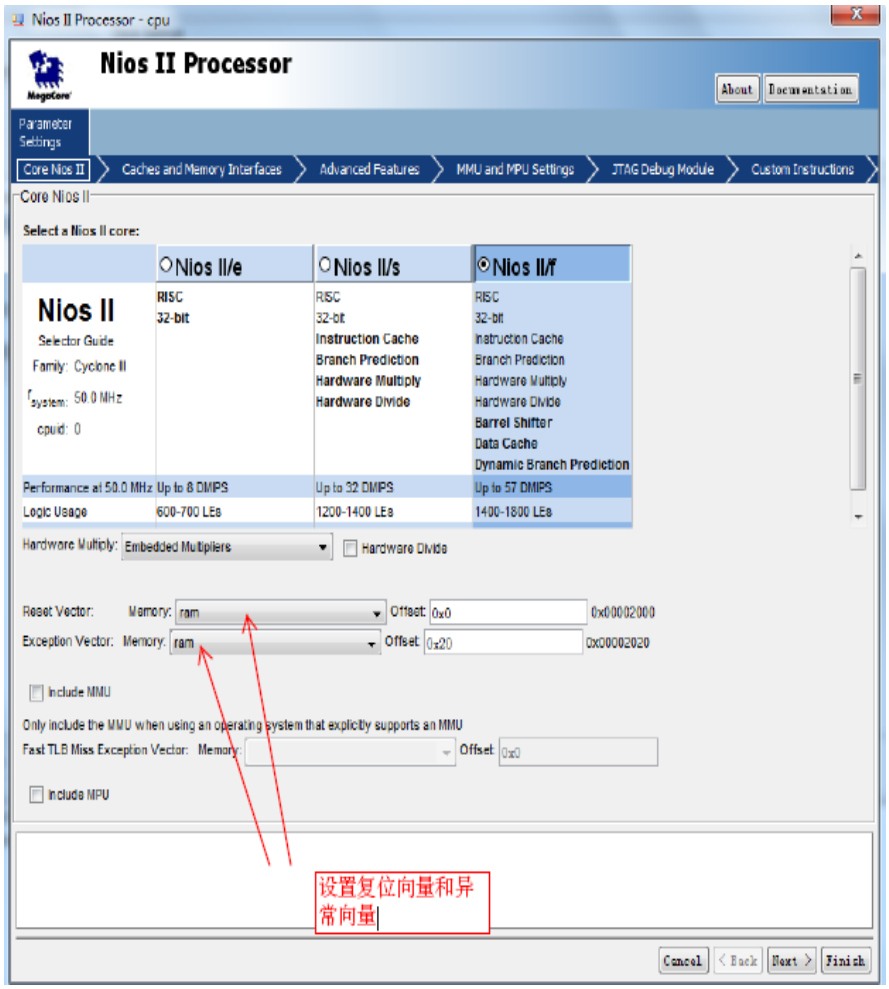
Use	Conn...	Module	Description	Clock	Base	End	IRQ	Tags
☑		cpu	Nios II Processor	[clk]				
		instruction_master	Avalon Memory Mapped Master	clk_0				
		data_master	Avalon Memory Mapped Master	[clk]			IRQ 0	IRQ 31
		jtag_debug_module	Avalon Memory Mapped Slave	[clk]	0x00020800	0x00020fff		
☑		sysid	System ID Peripheral	[clk]				
		control_slave	Avalon Memory Mapped Slave	clk_0	0x00021010	0x00021017		
☑		pio	PIO (Parallel I/O)	[clk]				
		s1	Avalon Memory Mapped Slave	clk_0	0x00021000	0x0002100f		
☑		ram	On-Chip Memory (RAM or ROM)	[clk1]				
		s1	Avalon Memory Mapped Slave	clk_0	0x00010000	0x0001ffff		
☑		jtag_uart	JTAG UART	[clk]				
		avalon_jtag_slave	Avalon Memory Mapped Slave	clk_0	0x00021018	0x0002101f		

6，设置

- 自动分配基址。设置界面如下：



- 设置复位向量和异常向量，如下图：



7， 点击 NEXT->Generate，开始生成系统。

8， 在系统生成后，会产生一个例化文件，如本教程建立的系统会产生名为 nios_cpu_inst.v 的例化文件，用户可以用此文件在 Quartus 中例化生成的系统，

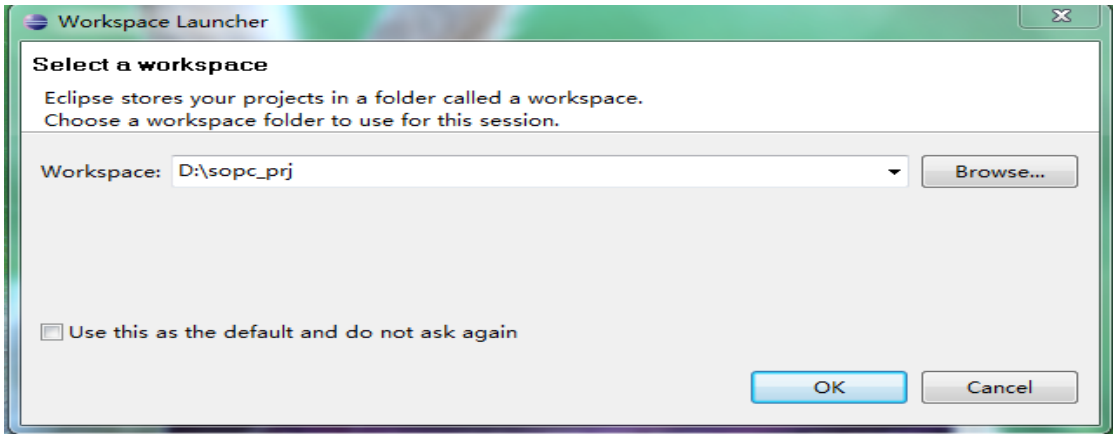
例化程序为：

```
module led(
            clk,
            reset,
            led
              );
             input clk,reset;
             output[3:0] led;
  //Example instantiation for system 'nios_cpu'
  nios_cpu nios_cpu_inst
    (
      .bidir_port_to_and_from_the_pio (led),
      .clk_0                          (clk),
      .reset_n                        (reset)
    );
Endmodule
```

之后分配管脚并编译整个系统，下载到开发板。

8，打开 NIOS II EDS 软件，首先设置 Workspace，本教程设置在工程目录下，这样 NIOS II EDS 软件会自动在工程目录

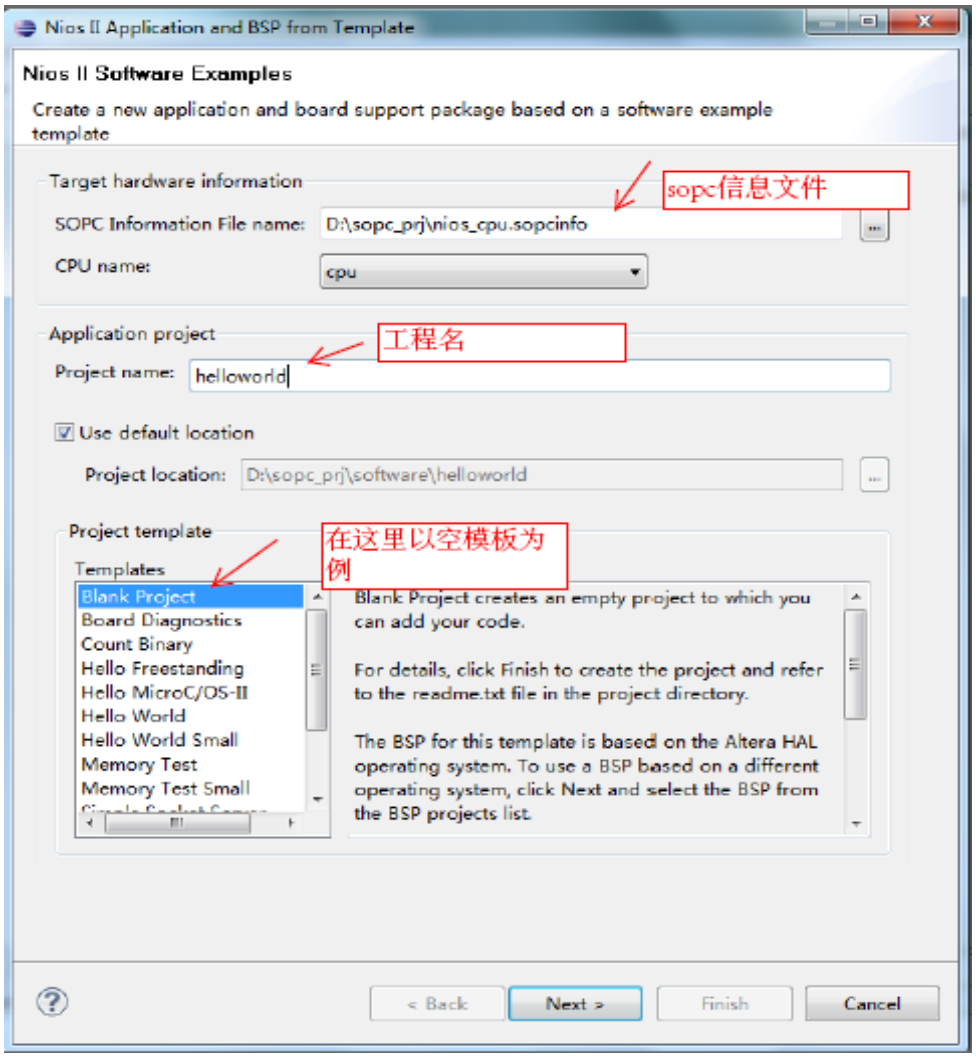
下生成一个名为 software 的文件夹，NIOS II 文件会保存在这里。如图：



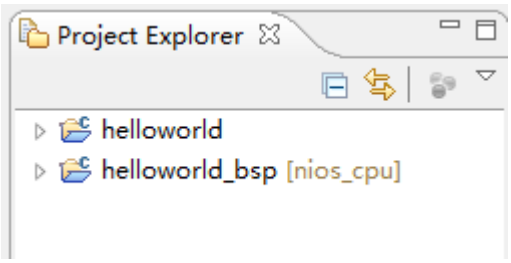
9，设置完成后点击 OK,进入环境。

10，创建 Hello World 程序。

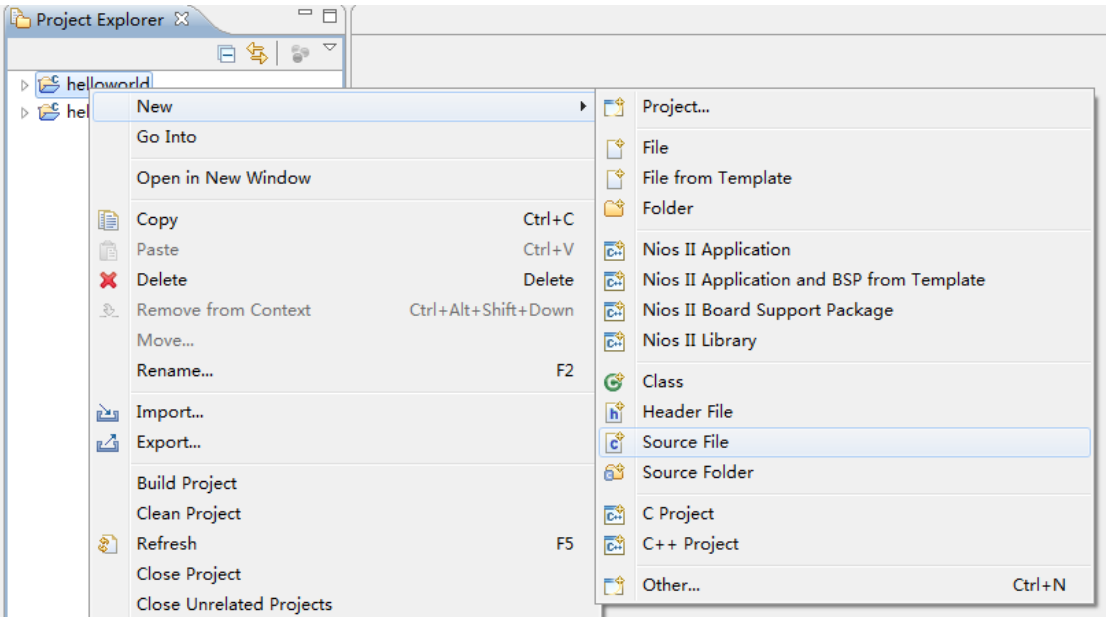
- File->New->NIOS II apllication and BSP from Template,按图示对弹出的对话框进行设置：



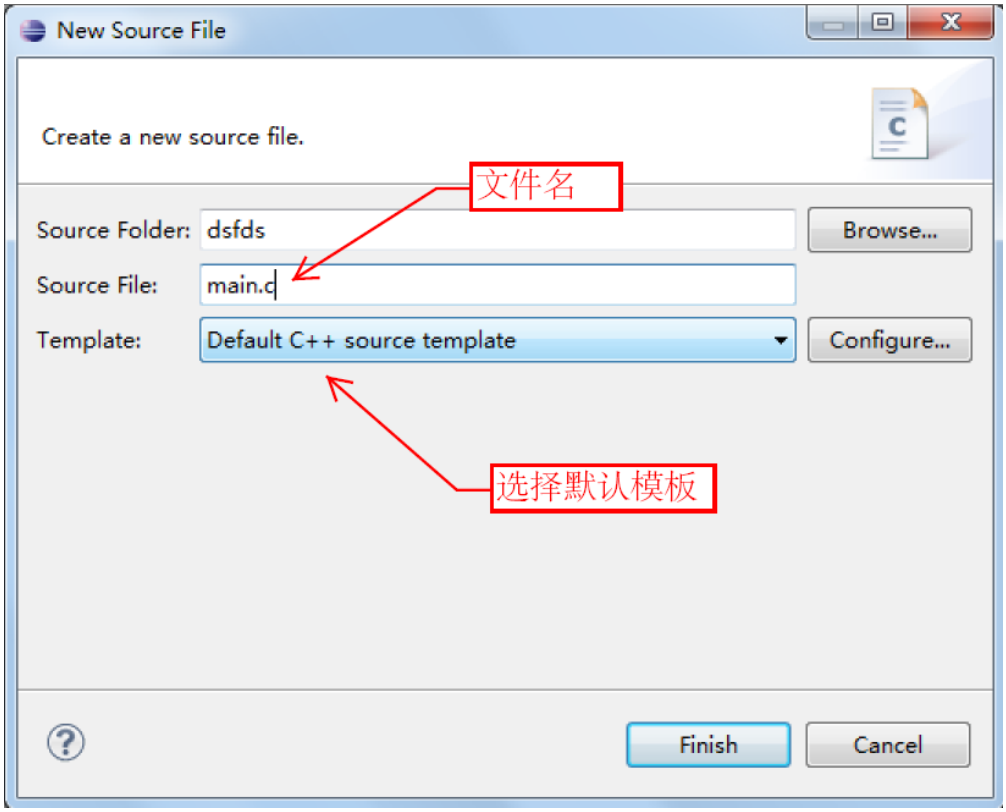
- NEXT->Finish,会发现在在左边的工程窗口出现两个工程文件，如下图：



- 右键 helloworld 工程，选择 New->Source file,如下：



之后设置文件名和模板：



- 在 main.c 中输入程序，代码如下：

```
#include <stdio.h>

Int main(void)

{
```

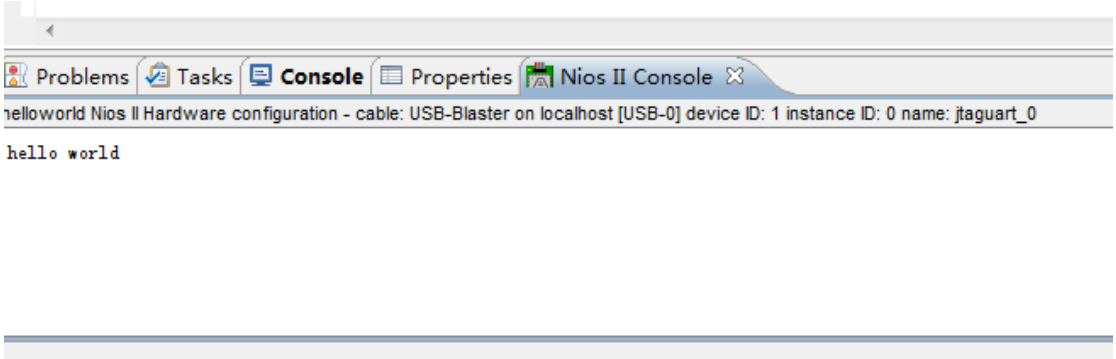
```
Printf("hello world");

Return 0;

}
```

11，硬件仿真

- 右键点击 heolloworld 工程->Build
- 连接开发板，右键 helloworld 工程，选择 Run As->NIOS II Hardware,程序执行效果如下图所示：



在自动弹出的 NIOS II Console 窗口中打印出“hello world”。