

# FPGA 综合实验指导书

## 一、Quartus II 安装说明

- (1) 运行 quart91 文件夹中的 setup.exe，安装 quartus91 软件。  
(安装路径 X:\altera\)
- (2) 安装破解器；(破解器和说明在 quart91 文件夹中)
- (3) 安装 USB 下载驱动。(驱动在 X:\altera\quartus\drivers)

## 二、基于 Quartus II 的电路设计（原理图输入法）

- 1、创建工程 File → New Project Wizard
- 2、设计输入 Block Diagram/Schematic File
- 3、编译（检查语法错误）Compilation
- 4、仿真（时序检查）Simulate
- 5、管脚分配 assignments editor
- 6、编译（检查系统设计错误）Compilation
- 7、下载 Programmer
- 8、实验验证

Quartus II 中每一项设计都对应一个工程（Project），Quartus II 中的工程是由有关的设计文件组成。为便于设计项目的存储，必须先建立一个文件夹(称为工作目录)，用来存放与此工程相关的所有文件。例中文件夹 E:\ZWY，此文件夹被默认为用户库（Work Library）。

当一个工程中需要多个文件时，这些文件必须放在同一个文件夹中，否则会出错。  
在打开文件时，必须先打开工程文件，才能对这个工程进行编译、仿真和下载。

【例 1-2-1】用原理图输入法设计一个与门。

步骤 1：进入 Windows 操作系统，打开 Quartus II 软件，Quartus II 界面如图 1-2-1 所示。

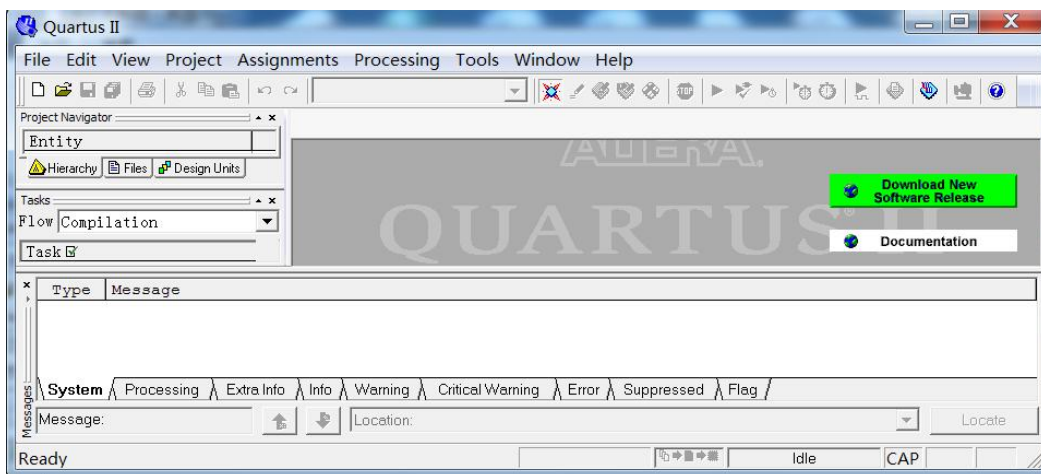


图 1-2-1 Quartus II 界面

### 步骤 2：创建工程

(1) 工程设置。选择 File → New Project Wizard 命令，如图 1-2-2 所示。在弹出的图 1-2-3 所示对话框中，选 Next，进入图 1-2-4 所示的工程设置页面，设置工程文件夹 E:\ZWY、工程名和顶层实体名均为 zand。设置完成后选 Next，进入目标器件选择页面。

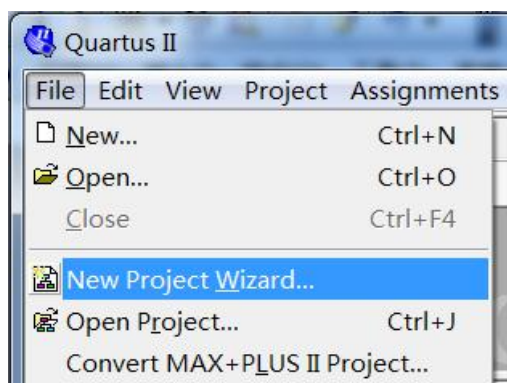


图 1-2-2

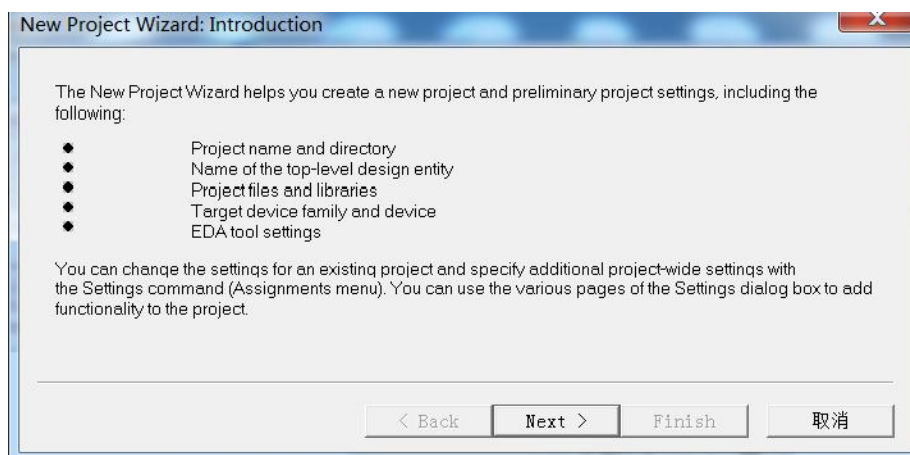


图 1-2-3 新建工程向导

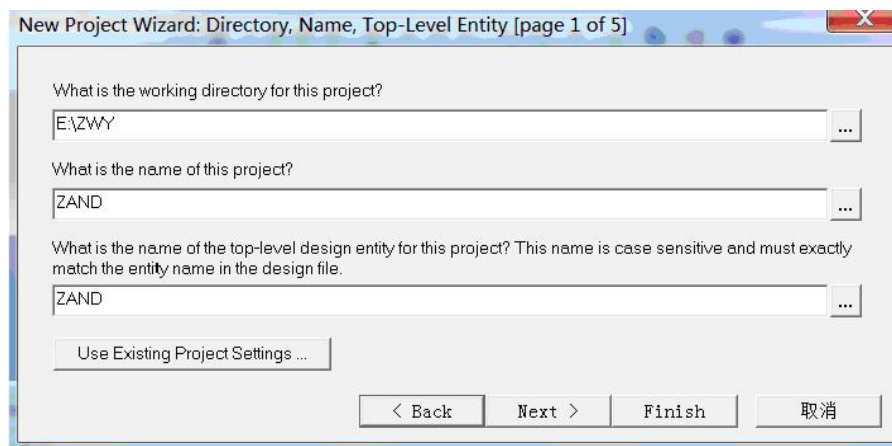


图 1-2-4 新建工程路径、工程名称、工程实体名

(2) 指定目标器件。设计人员可以在该步骤中指明本次设计的目标器件。按图 1-2-5 所示选择目标芯片。设置完毕，选择 Next。

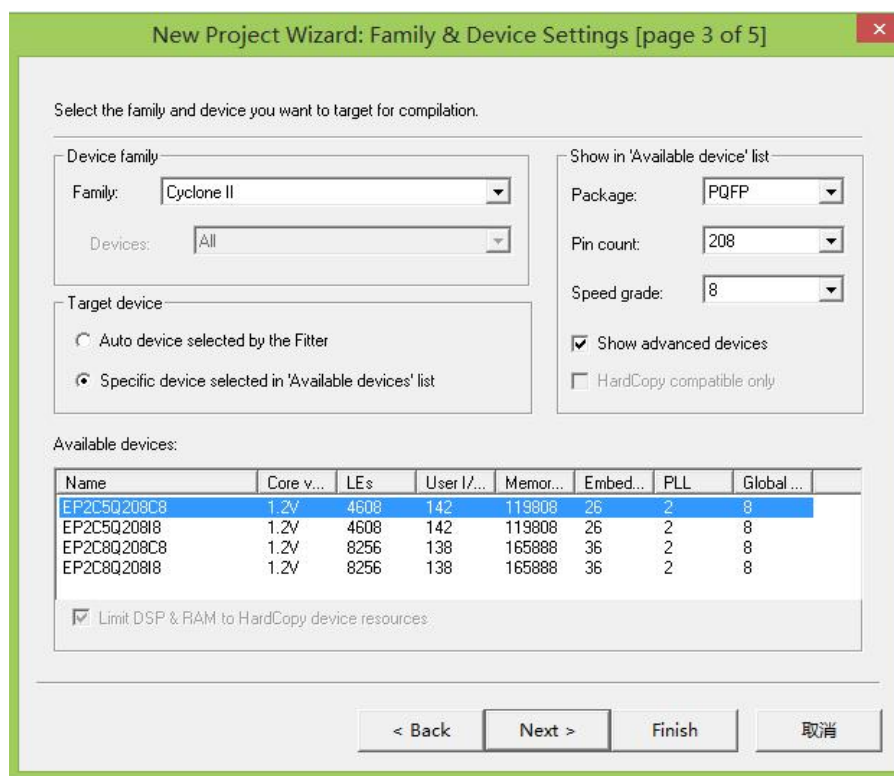


图 1-2-5 目标芯片选择

(3) 工程总结。图 1-2-6 工程设置情况总结，包括工程文件夹位置、工程名和顶层实体名、器件类型、综合器与仿真器选择等。设计人员在此可检查设置是否符合要求。若无问题，点击“Finish”结束工程的创建。若有不符合要求的情况，可点击“Back”退回修改。

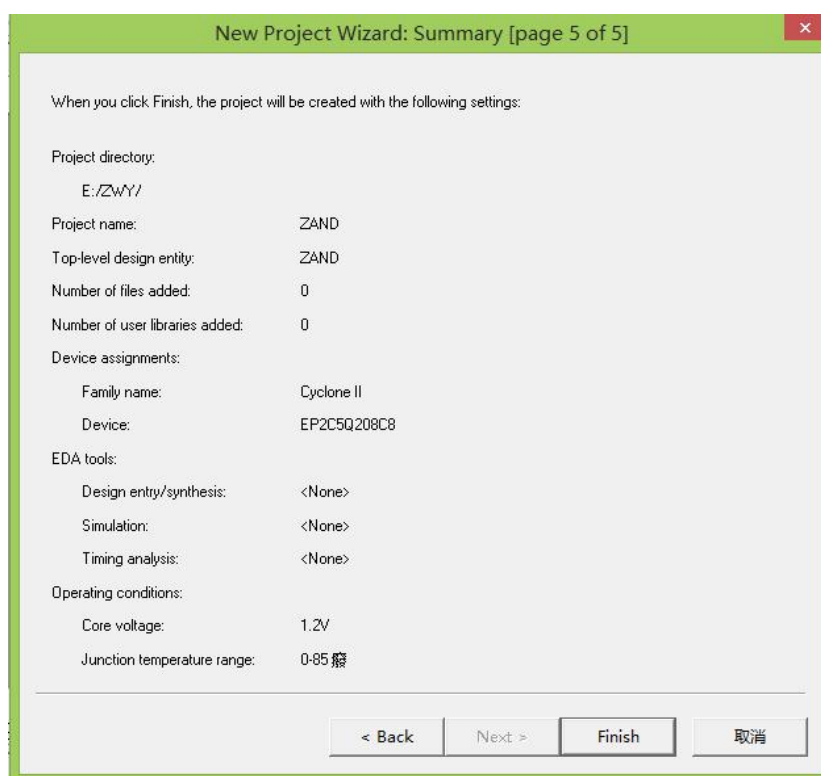


图 1-2-6 创建工程总结

### 步骤 3: 打开原理图编辑器

项目建立后, 便可进行具体设计, 为项目添加实际的设计文件。选择“File”菜单“New”, 弹出文件类型选择对话框如图 1-2-7 所示, 在“Design Files”栏中选择“Block Diagram/Schematic File”, 进入图 1-2-8 所示原理图文件（扩展名为 bdf）编辑界面。

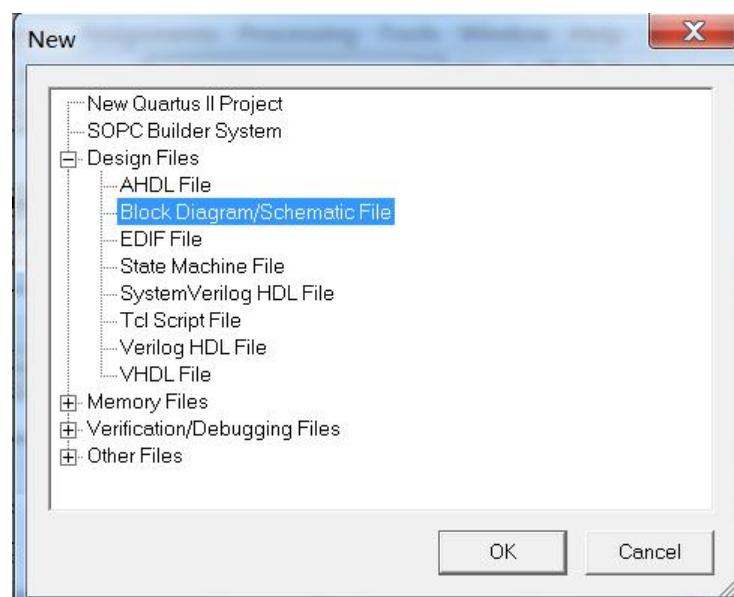


图 1-2-7 新建原理图文件（.bdf 文件）

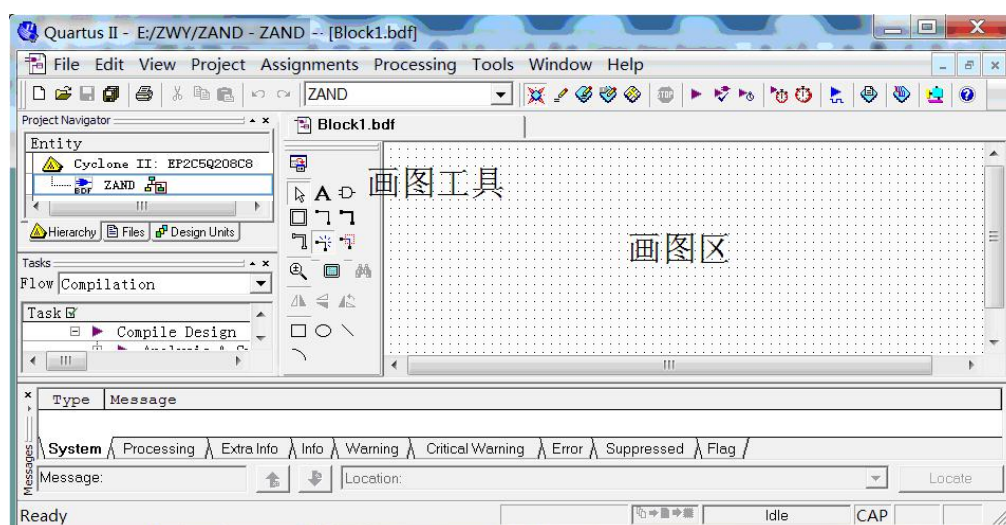


图 1-2-8 bdf 原理图文件编辑界面

### 步骤 4: 原理图文件编辑

#### (1) 元器件放置

在图 1-2-8 原理图文件编辑界面空白处双击鼠标左键, 弹出元件选择页面, 如图 1-2-9 所示。图中“Libraries”处列出元件库目录, 包括基本元件库、宏功能库和其它元件库。选择其中任一库, 如基本元件库, 双击所需的元件即可将元件调入文件。也可在页面“Name”处输入元件名, 如 and3（三输入与门）、not（非门）、input（输入端口）等, 并点击 OK。

若要放置相同的元件, 只要按住 Ctrl 键, 用鼠标拖动该元件。

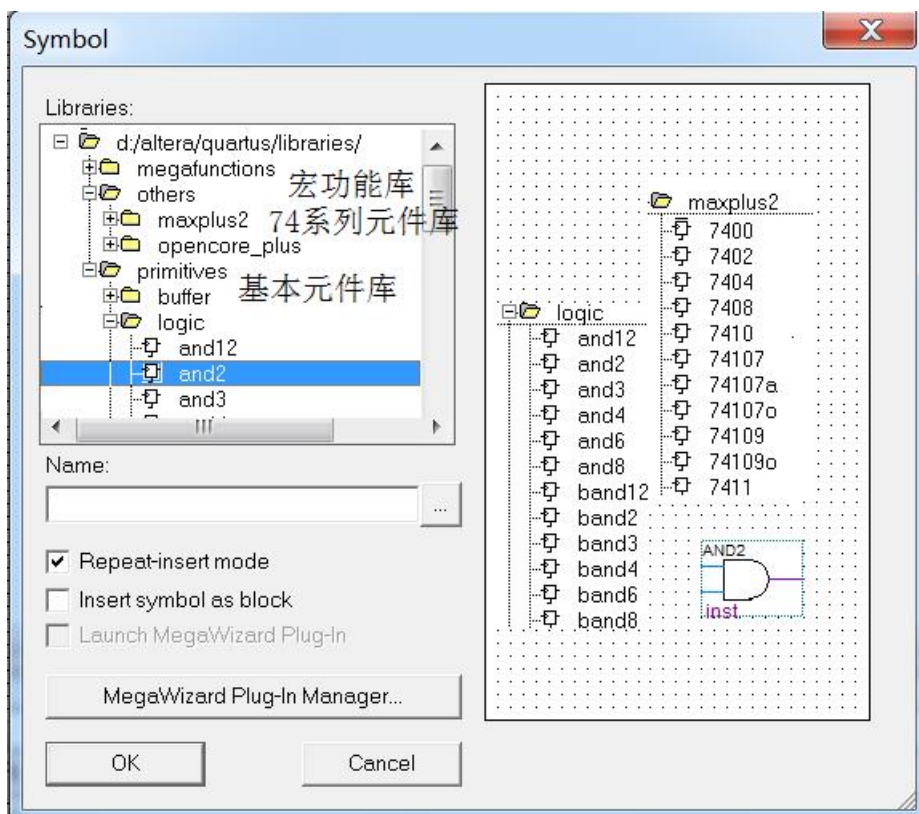


图 1-2-9 元件选择窗口

## (2) 在器件之间添加连线

把鼠标移到元件引脚附近，则鼠标光标自动由箭头变为“十”字，按住鼠标左键拖动，即可画出连线。如图 1-2-10。

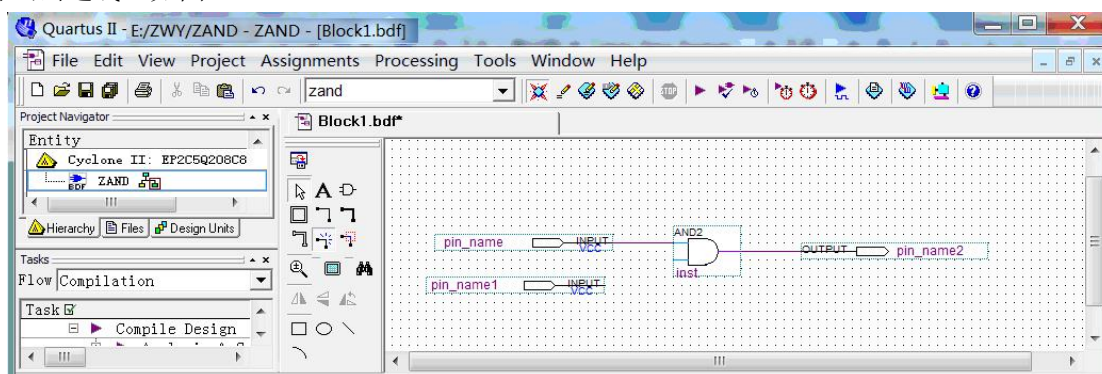


图 1-2-10 在器件之间添加连线

## 步骤 5: 给输入、输出引脚命名

电路图绘制完成后，给输入、输出引脚命名加以区别。例中将输入、输出引脚的“pin\_name”分别改为：a, b 和 y，如图 1-2-11 所示。



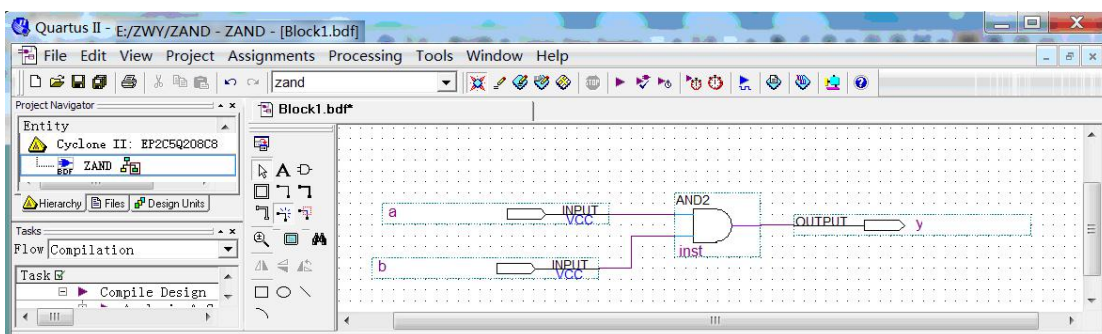


图 1-2-11 给输入输出引脚命名

#### 步骤 6: 保存原理图文件

在选择 File \Save As 保存原理图文件（文件名为 ZAND.bdf），将文件存入用户库，如图 1-2-12a 所示，并选择将文件加入当前工程，点击保存后，图 1-2-11 的原理图输入界面发生了变化，如图 1-2-12b 所示。

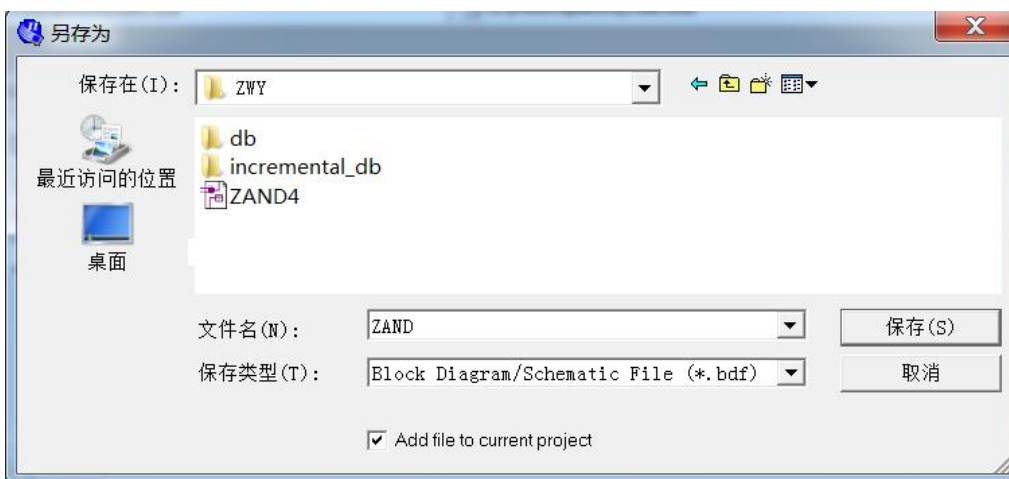


图 1-2-12a 将原理图文件保存并加入当前工程

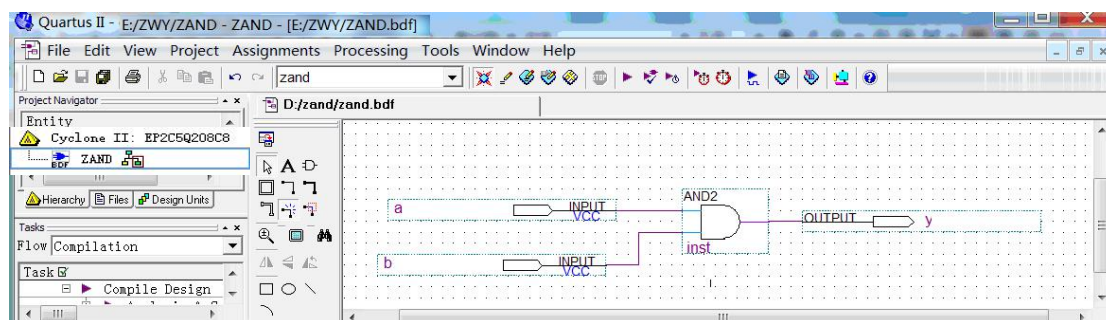


图 1-2-12b 将原理图文件保存并加入当前工程后的界面

#### 步骤 7: 编译

Quartus II 的编译器可完成对设计项目的检错、逻辑综合、结构综合等功能。选择“Processing”下的“Start Compilation”项，即可启动编译。编译过程中“Processing”窗口会显示相关信息，若发现问题，会以红色的错误标记条或深蓝色警告标记条加以提示。Warning 一般不影响编译通过，error 则必须排除。双击错误条文，光标将定位于错误处。

编译完成后，将会出现图 1-2-13 所示的编译结果报告。用户可以在窗口中查看项目编

译后的各种统计信息，包括资源使用情况、时序情况、适配情况等。

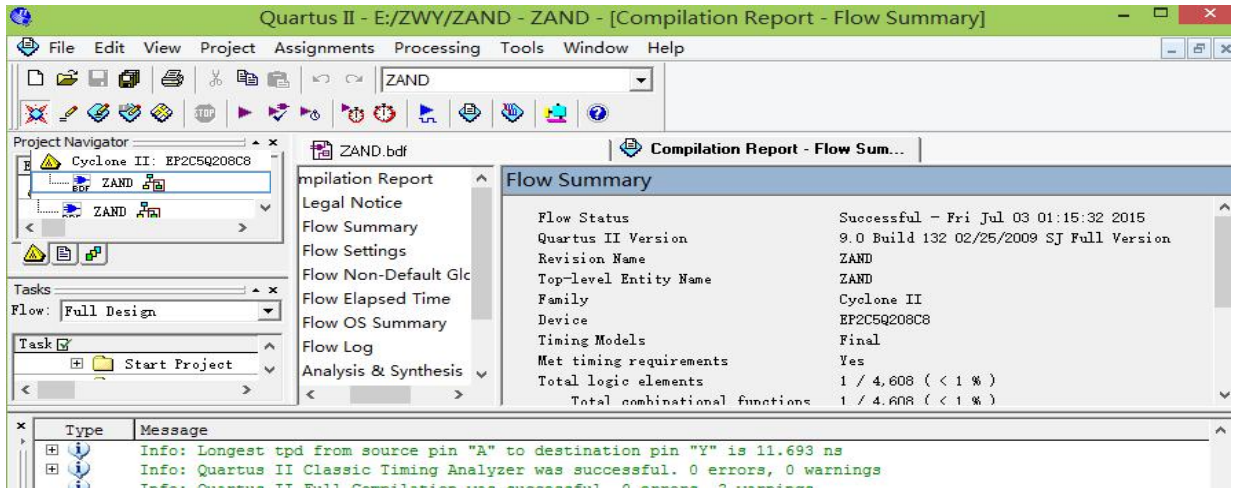


图 1-2-13 编译报告窗口

**步骤 8: 时序模拟**

工程编译完成后，可以进行功能和时序仿真测试，以验证设计结果是否满足设计要求。对工程进行仿真的步骤如下：

**(1) 新建 vwf 波形文件**

选择“File”菜单“New”，弹出文件类型选择对话框如图 1-2-14 所示，在“Verification Files”中选择 Vector Waveform File”，点击 OK 后，此时弹出图 1-2-15 所示 vwf 波形文件编辑界面，新建仿真波形文件（扩展名为 vwf）。

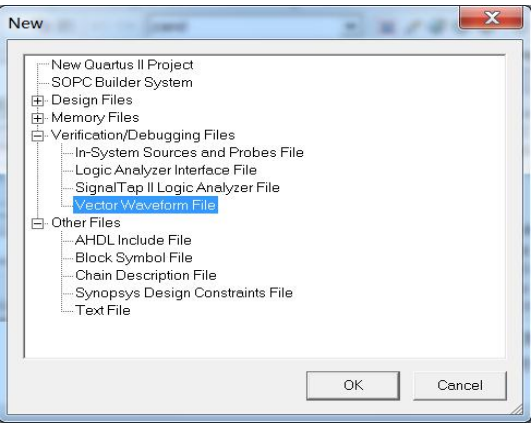


图 1-2-14 新建 vwf 波形文件

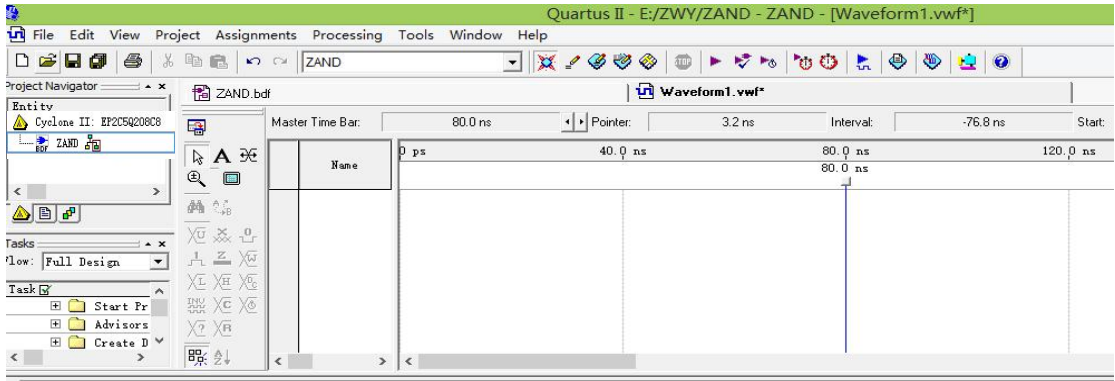


图 1-2-15 vwf 文件编辑界面

## (2) 确定仿真时间和网格宽度

为设置满足要求的仿真时间区域，选择“Edit”菜单下的“End Time”项，指定仿真结束时间。可通过“Edit”菜单下的“Grid Size”项指定网格宽度。例中将仿真结束时间设定为 20us（图 1-2-16），网格宽度设定为 40ns（图 1-2-17）。（必须 $\geq 40\text{ns}$ ）

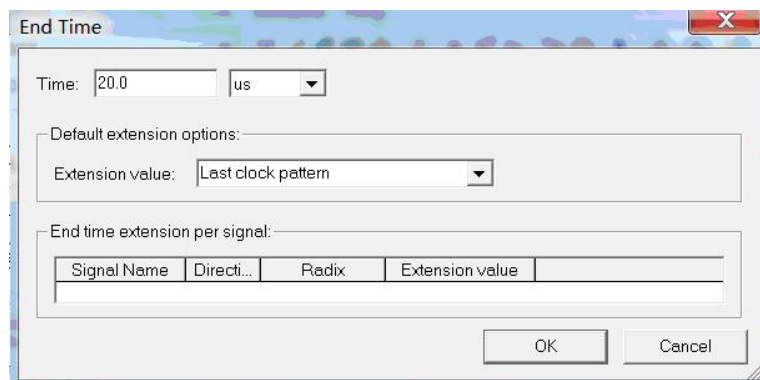


图 1-2-16 指定仿真结束时间

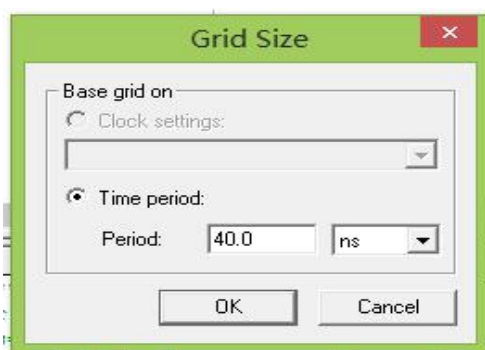


图 1-2-17 指定网格宽度

## (3) 编辑 vwf 文件

在图 1-2-18 vwf 波形文件编辑界面中，在端口列表名 name 下空白处点击右键，选择“Insert Node or Bus”，弹出图 1-2-19 所示对话框，点击“Node Finder”，弹出图 1-2-20 所示对话框；点击“List”找到设计中出现的输入输出端口；用图 1-2-20 中“>>”符号将全部或部分选中的端口调入仿真波形文件；点击图 1-2-20 所示 Node Finder 对话框中的 OK，再点击图 1-2-19 所示 Insert Node or Bus 对话框中的 OK。



图 1-2-18 vwf 波形文件编辑界面



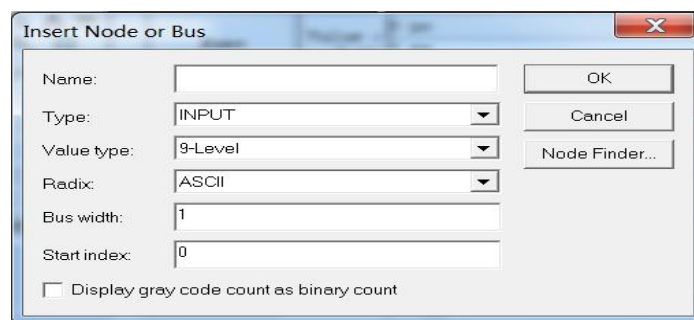


图 1-2-19 端口搜索

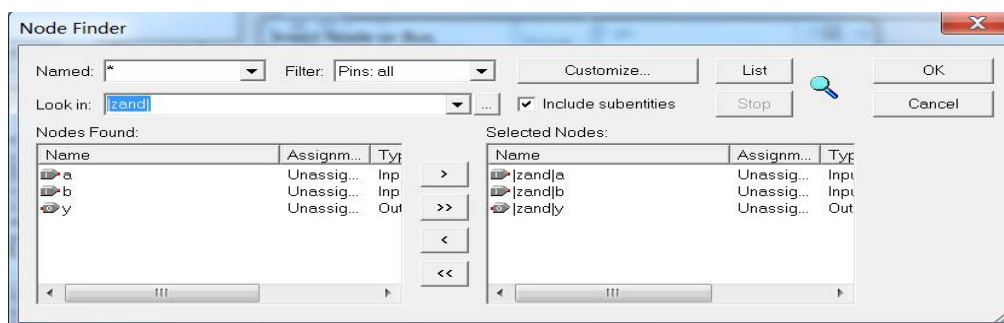
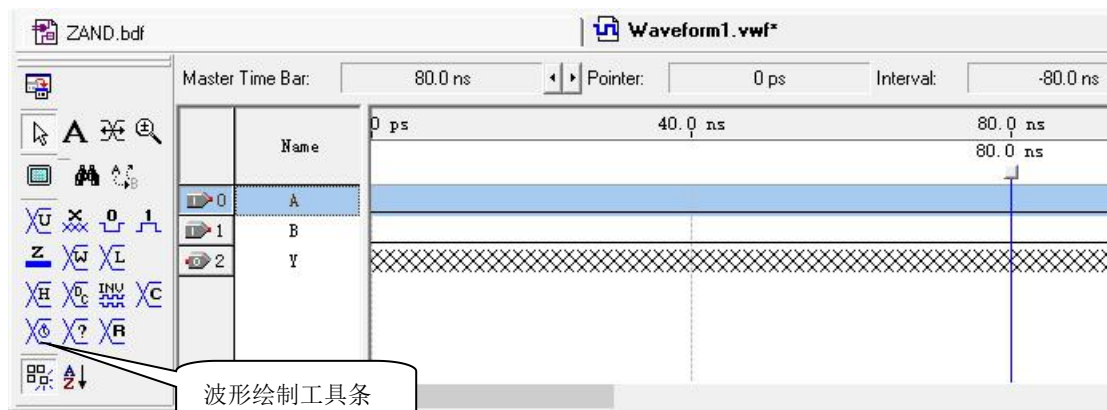
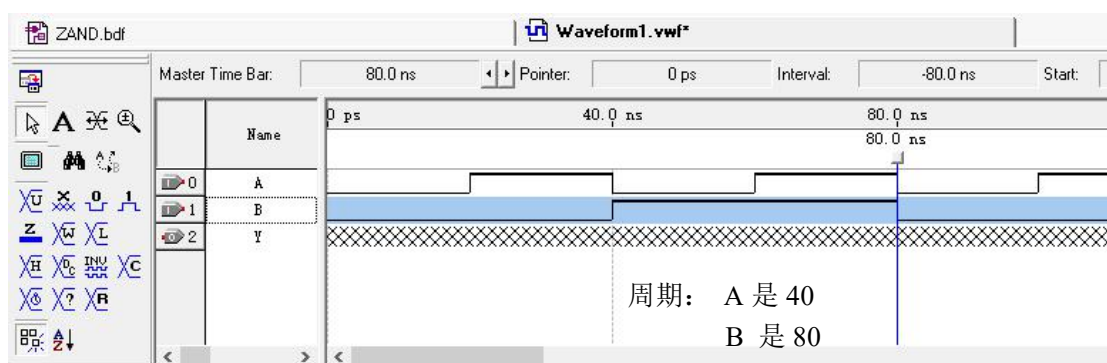


图 1-2-20 将输入输出端口调入仿真波形文件

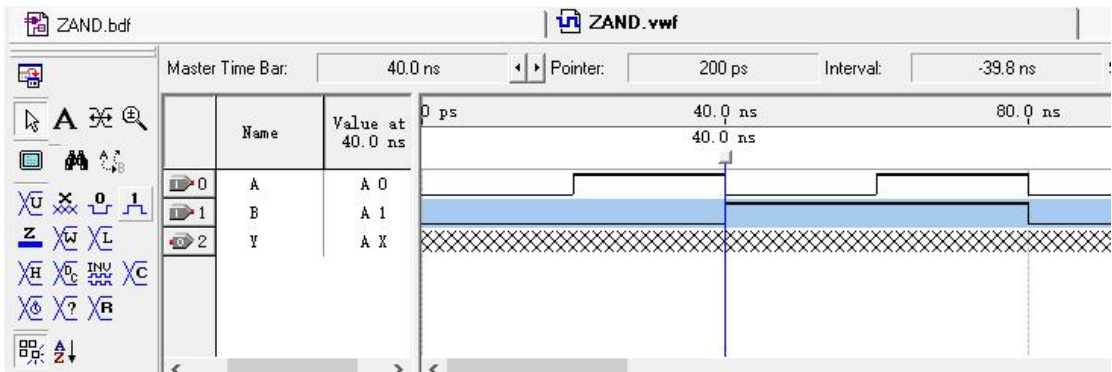
仿真前需要对输入端口进行赋值，利用图 1-2-21 中波形绘制工具来编辑输入端口 a，b 波形。单线信号赋值时，可用鼠标拖动选定区域，利用置 0、置 1 等按钮将区域赋值为低电平、高电平；总线信号赋值时，可利用专用的总线赋值按钮来完成；时钟信号赋值时，则应该选择专门的时钟信号设置按钮，在设置对话框内指明时钟信号的周期。编辑完成后选择 File \Save As 保存仿真波形文件，文件名为 ZAND.vwf，点击保存，将波形文件存入用户库。



(a)



(b)



(c)

图 1-2-21 波形编辑和保存

#### (4) 启动仿真

在“Processing”菜单下选择“Start Simulation”命令，或点击其快捷图标，即可启动工程仿真，如图 1-2-22 所示。仿真结束后可在 zand.vwf 文件中观察仿真结果，如图 1-2-23 所示。可见，例中仿真结果符合表 1-2-1 所列的功能（注意有竞争冒险现象和信号延迟现象）。

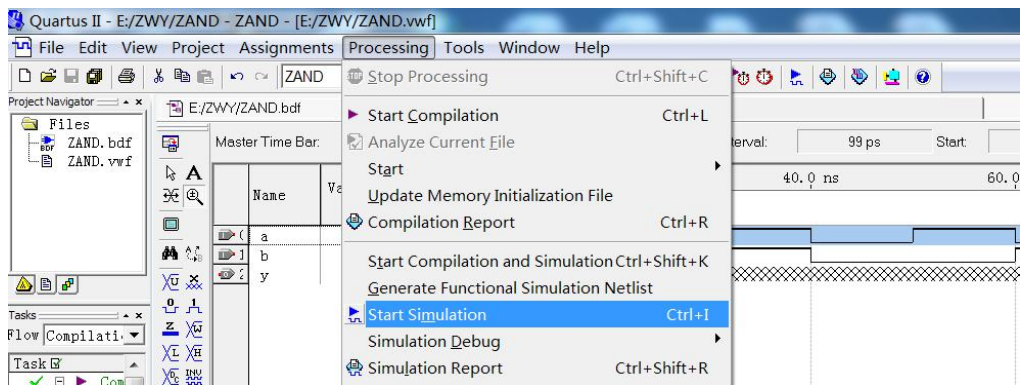


图 1-2-22 启动仿真

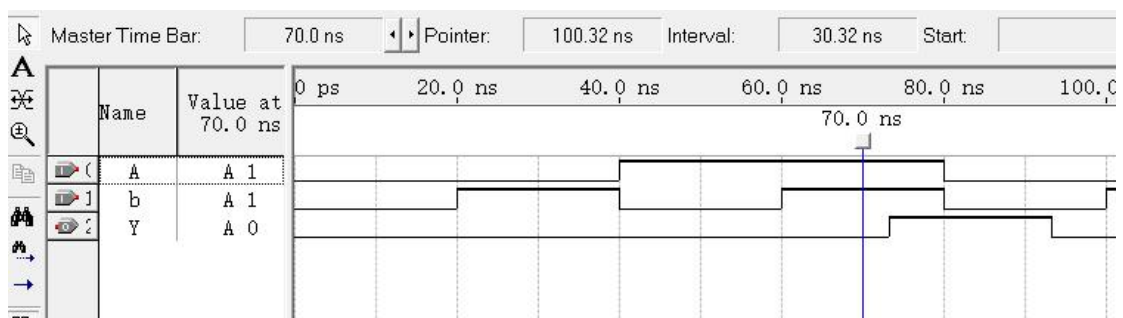


图 1-2-23 仿真结果（注意延迟时间）

表 1-2-1 与门真值表

a b	y
0 0	0
0 1	0
1 0	0
1 1	1

步骤 9：管脚分配

仿真正确后，就可以准备将设计下载至 PLD 目标芯片进行验证了。通过管脚分配将其输入输出端口与 PLD 器件的管脚建立对应关系。选择 Assignments 菜单中的 Pin 栏，显示项目的信号列表和目标芯片的管脚图，如图 1-2-24 所示。

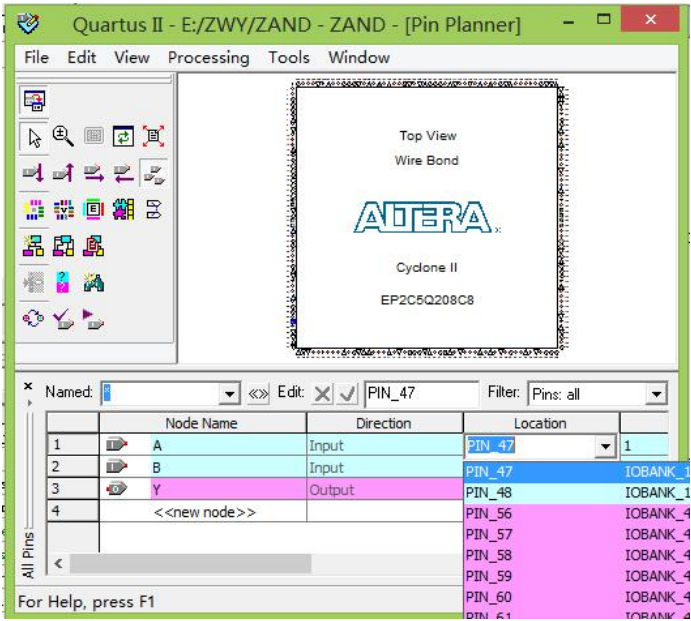


图 1-2-24 管脚分配界面

管脚分配时应注意所用实验箱的实际情况，由于实验箱已将 FPGA/CPLD 芯片的引脚与外部的开关、LED、数码管、接口设备等连接在一起，管脚分配时只能根据管脚与外部器件的对应表用**手动方式分配**。实验箱使用的 EP2C5Q208C8 管脚分配方案如表 1.2.2 所示。

表 1.2.2 EP2C5Q208C8 管脚分配方案

8 位数码管				拨码开关		发光二极管		时钟源	
a	164	MS1	170 左 1	S11	47	LED7	63	1Hz	3
b	165	MS2	171	S10	48	LED6	64	1KHz	5
c	168	MS3	173	S09	56	LED5	67		
d	169	MS4	175	S08	57	LED4	68		
e	176	MS5	182	S07	58	LED3	75		
f	179	MS6	185	S06	59	LED2	76		
g	180	MS7	187	S05	60	LED1	77		
dp	181	MS8	188	S04	61	LED0	80		
高电平亮		低电平有效		拨上 1，拨下 0		高电平亮			

FPGA 芯片外接输入端口，用于接时钟和单脉冲

I/O NO.	1	2	3	4	5	6	7	8	9	10
FPGA 引脚		131	206	34	31	27	14	11	6	3
I/O NO.		11	12	13	14	15	16	17	18	19
FPGA 引脚		208	205	201	32	28	15	12	8	4
I/O NO.		20	21	22	23	24	25	26	27	28
FPGA 引脚		207	203	200	33	30	24	13	10	5

管脚分配的过程：双击图 1-2-24 中“Location”下的空白格，输入端口 a,b 与开关相连，输出端口 y 与发光二极管相连，管脚分配情况如图 1-2-25 所示。

Named: <span style="border: 1px solid black; padding: 2px;">PIN_47</span> Filter: Pins: all				
	Node Name	Direction	Location	I/O Bank
1	A	Input	PIN_47	1
2	B	Input	PIN_48	1
3	Y	Output	PIN_63	4
4	<<new node>>			

图 1-2-25 管脚分配参考方案

管脚分配后，需要对工程再一次编译，以将管脚对应关系存入设计，并产生 zand.sof 文件。

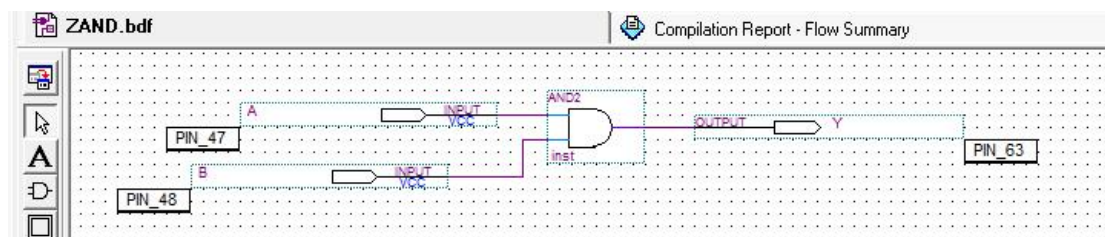


图 1-2-26 管脚分配后的原理图文件

#### 步骤 10: 下载（用 USB 口）

(1) 关闭实验箱上的电源，用 USB 下载器分别连接计算机和目标器件的 JTAG 口，打开实验箱电源，如图 1-2-27 所示。

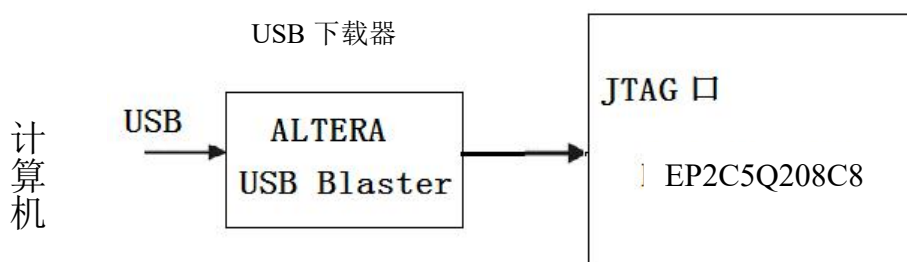


图 1-2-27 USB 下载器与开发板的连接图

(2) 在 Quartus II 软件中选择“Tools”菜单下的“Programmer”命令。在下载之前，首先需要进行硬件设置，如图 1-2-28 所示，点击界面中“Hardware Setup”按钮，在“Hardware Settings”项中，找到“Currently selected hardware”选项，选中“USB blaster”后，如图 1-2-29 所示，再点击“close”退出，完成硬件设置。在图 1-2-28 界面中，将编程模式确定为“JTAG”，并在“Program/Configure”复选框内打“√”，便可点击“Start”按钮开始下载，Quartus II 软件便将设计（zand.sof）载入 FPGA 目标器件中。

注意：若有提示下载不成功信息，应按以上各步检查是否设置正确，并检查计算机与实验箱硬件连接，排除故障后，再次尝试下载。

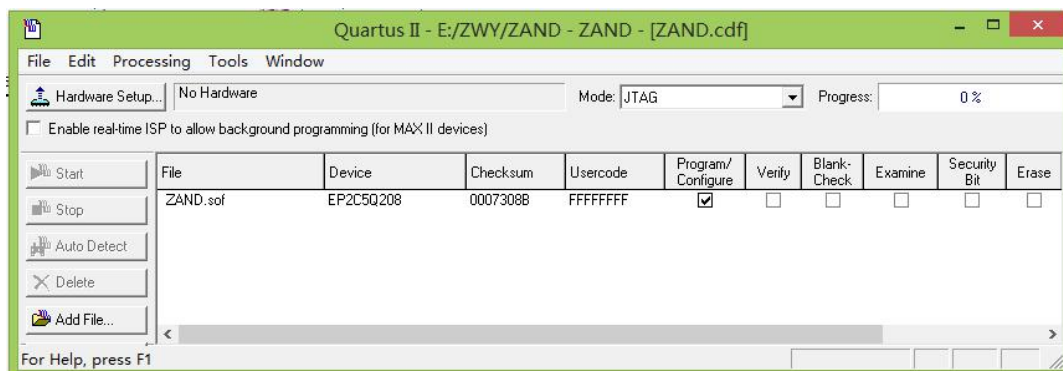




图 1-2-28 设置下载界面

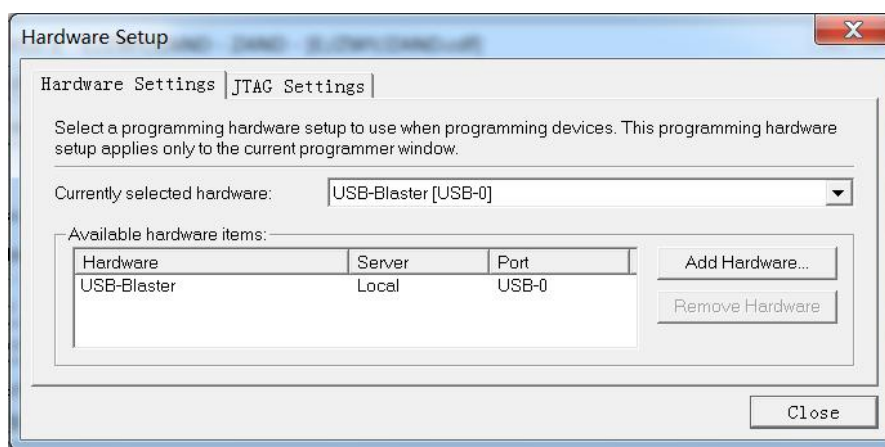
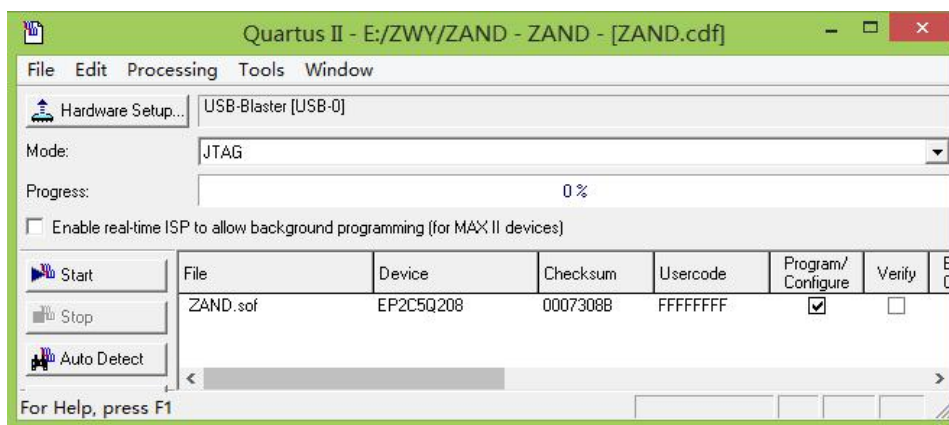


图 1-2-29 下载设备设置为“USB-Blaster”



下载设备设置为“USB-Blaster”

**步骤 11:** 设计文件下载至目标芯片后，根据步骤 9 管脚分配的结果，改变数据开关的电平，验证发光管的状态是否满足表 1-2-1。硬件系统示意图如图 1-2-30 所示，图中所示的 Y 对应于实验箱上的发光二极管 LED7，a，b 对应于实验箱上的电平开关 S11，S10。

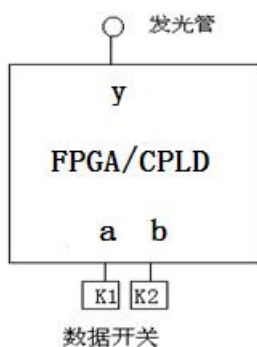


图 1-2-30 实验连接示意图

## 附一 为工程选定芯片和设置：（在编译之前做）

1 如图 1-3-1 所示，单击“Assignments”菜单，选择“Device...”，弹出“Setting”浮动窗口；  
做如图 1-3-2 所示的选择；

2 继续在“Setting”浮动窗口中，单击“Device and Pin Options”按钮，弹出“Device and Pin Options”浮动窗口；单击“Unused Pins”，将未使用的引脚设定为三态输入方式，如图 1-3-3 所示。无误后，选择“确定”按钮，然后选择图 1-3-2 中的“OK”按钮。

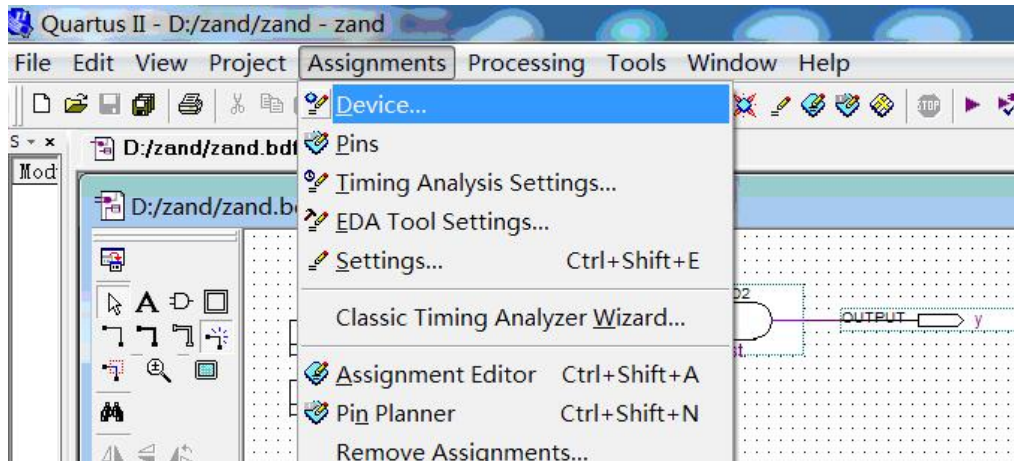


图 1-3-1

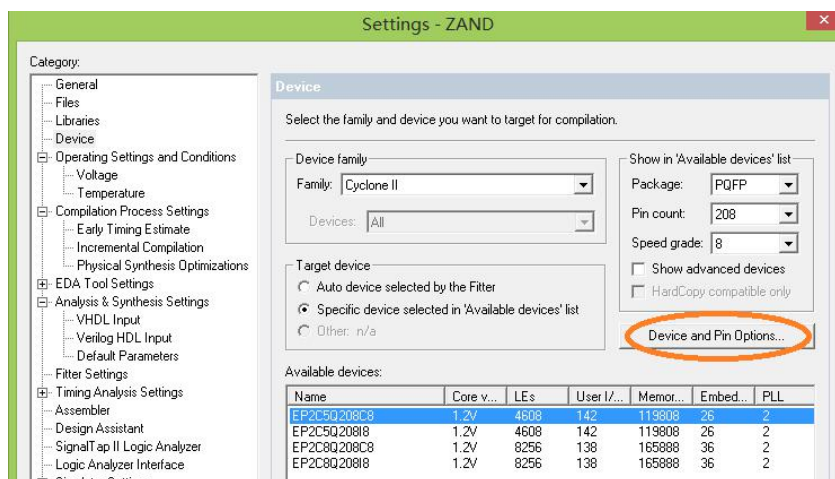
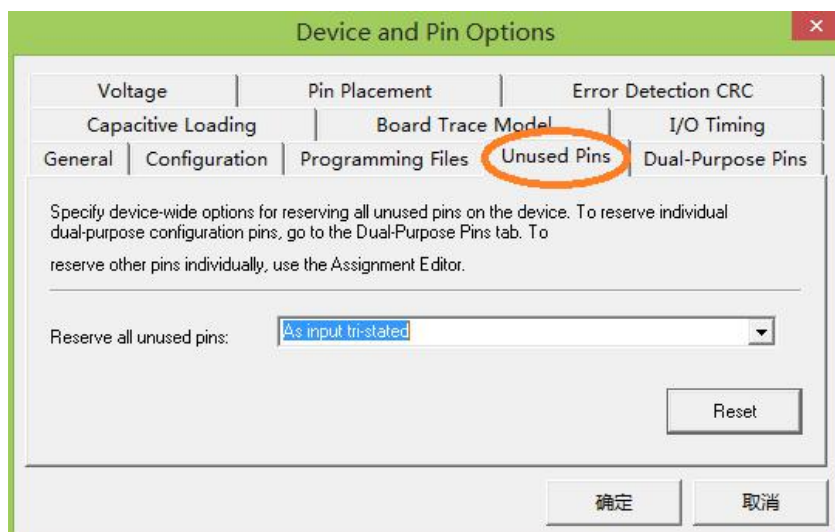


图 1-3-2



选 As input tri-stated

图 1-3-3

附二 在原理图设计文件的基础上建立一个元件符号，放在用户库（当前文件夹）中，供当前项目工程中的原理图文件调用（自己除外）。

生成元件符号的方法是：在“File”菜单中，使用 Create/Update 命令,从当前的原理图设计文件建立单元符号文件。

**步骤 1** 在当前项目工程中，选择“File”菜单“New”，弹出文件类型选择对话框如图 1-3-4 所示，在“Design Files”栏中选择“Block Diagram/Schematic File”，进入图 1-3-5 所示原理图文件（扩展名为 bdf）编辑界面。

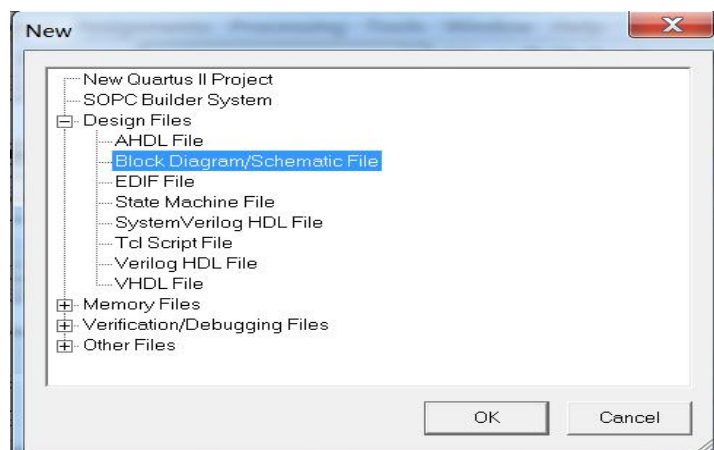


图 1-3-4 新建原理图文件（.bdf 文件）

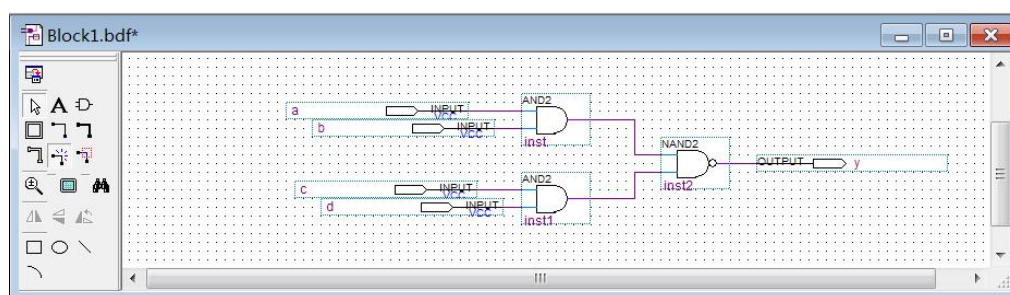


图 1-3-5a bdf 原理图文件编辑界面

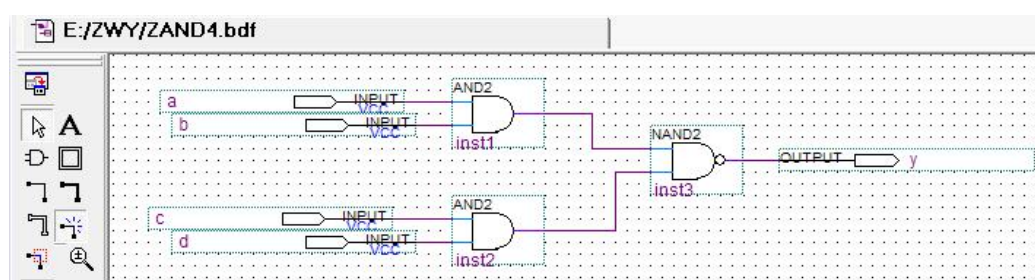


图 1-3-5b 将原理图文件保存

**步骤 2：** 保存原理图文件

在选择 File \Save As 保存原理图文件（文件名为 zand4.bdf），将文件存入用户库。如图 1-3-6 所示，点击保存。

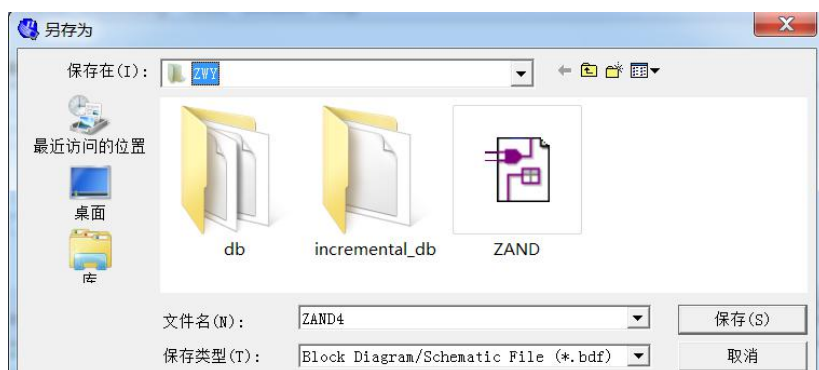


图 1-3-6 将原理图文件保存

### 步骤 3: 制作元件

(1) 在原理图编辑界面下，单击“File”菜单，选择“Create/Update”，做如图 1-3-7 所示的选择：

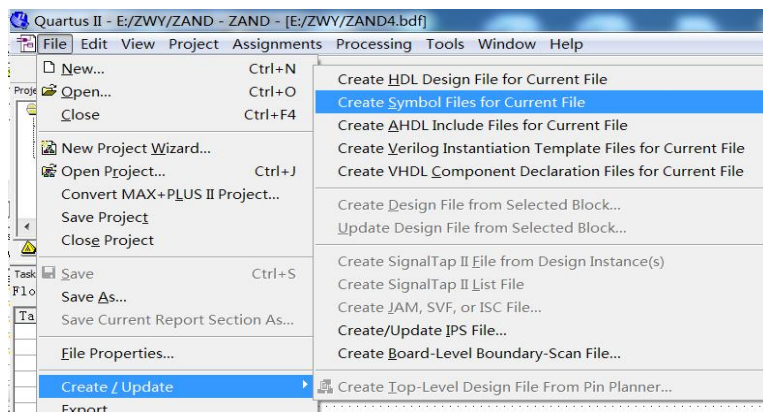


图 1-3-7

(2) 元件符号保存的位置选择如图 1-3-8 所示；

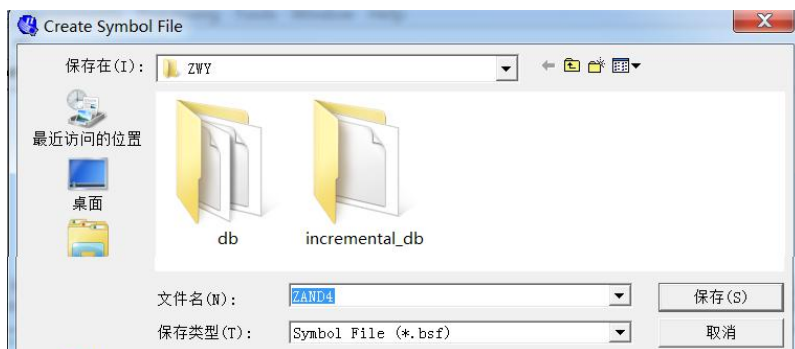


图 1-3-8

(3) 单击“保存”。

**步骤 4:** 生成的元件符号 zand4.bsf 被存入用户库，如图 1-3-9 所示。供当前项目工程中的原理图文件（自己除外）调用，调用方法与器件库中元件的调入方法相同。



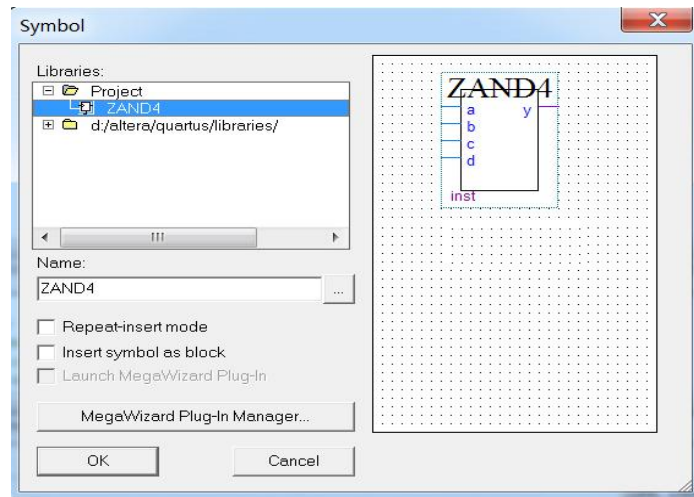


图 1-3-9

### 三、基于 Quartus II 的电路设计（HDL 文本输入法）

**【例 1-2-2】** 用 Verilog HDL 设计一个 3 线-8 线译码器。

**步骤 1~2:** 与例 1-2-1 原理图输入法相同，工程名与文本文件的名称相同。

**步骤 3:** 启动 File \ New 菜单，如图 1-4-1，选择 “Verilog HDL File”，点击 OK。

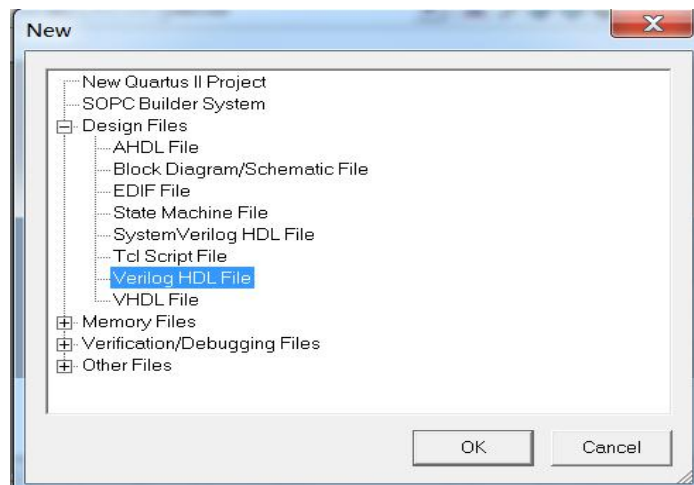


图 1-4-1 新建 Verilog HDL 文件

**步骤 4:** 在 HDL 编辑界面下，输入 Verilog HDL 文本文件 decoder.v。

```
module decoder(out,in);
output[7:0] out;
input[2:0] in;
reg[7:0] out;
always@(in)
begin
case(in)
3'd0:out=8'b00000001;
3'd1:out=8'b00000010;
3'd2:out=8'b00000100;
3'd3:out=8'b00001000;
3'd4:out=8'b00010000;
3'd5:out=8'b00100000;
```

```

3'd6:out=8'b01000000;
3'd7:out=8'b10000000;
endcase
end
endmodule

```

#### 步骤 5: 保存文本文件

在菜单 File \Save as 下保存文件名为 decoder.v（与 module 后的模块名相同，字母区分大小写），并选择将文件加入当前工程。

#### 步骤 6: 编译、仿真、管脚分配、下载等，与图形输入法相同。

**附1** Verilog HDL文件描述的功能模块可以生成一个元件符号，放在用户库（工作目录）中，供当前项目工程中的原理图文件调用。生成元件符号的方法是：在“File”菜单中，使用 Create/Update 命令,从当前的 Verilog HDL设计文件建立元件符号。

**步骤 1** 在当前项目工程中，选择“File”菜单“New”，弹出文件类型选择对话框如图 1-2-6 所示，在“Design Files”栏中选择“Verilog HDL File”，进入 HDL 编辑界面。

**步骤 2:** 在 HDL 编辑界面下，键入完整的 Verilog HDL 代码。

```

module  decody(out,in);
.....

```

#### 步骤 3: 保存设计文件

在菜单 File \Save 下保存文件为 decody.v。

**步骤 4:** 在 HDL 编辑界面下，选择菜单“File\Create/Update”，并选择“Create Symbol Files for Current File”子菜单，如图 1-4-2 所示。

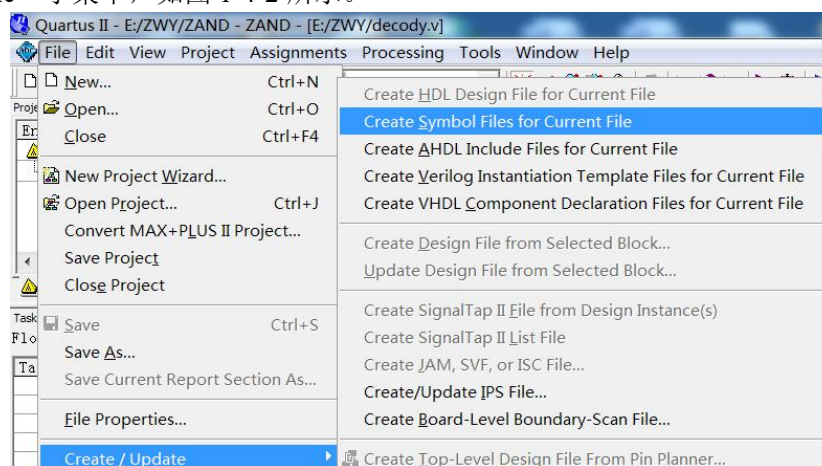


图 1-4-2 生成元件符号

生成的元件符号被放入用户库，如图 1-4-3 所示。供当前项目工程中的任何一个原理图文件调用，调用方法与器件库中元件的调入方法相同。

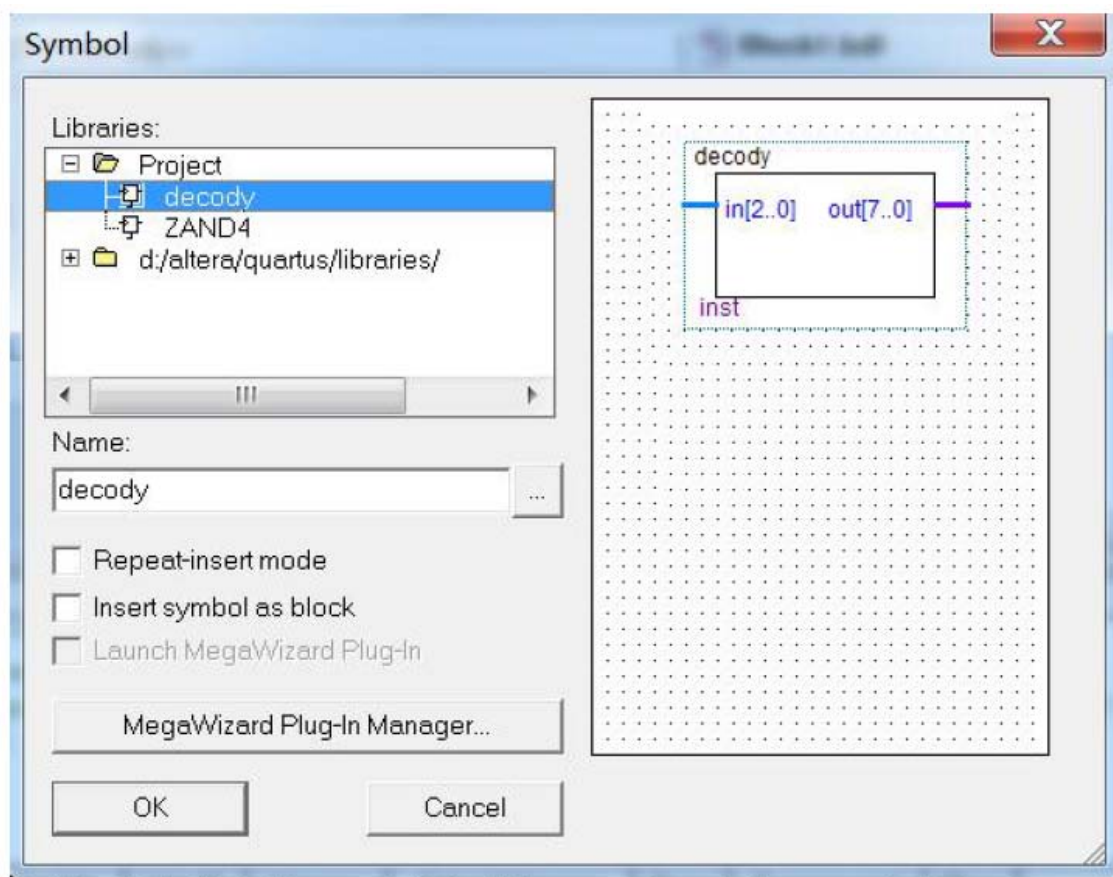


图 1-4-3 生成元件符号及调用

# 实验任务

1. 设计 3-8 译码器；
2. 设计十进制加法计数器，具有同步清零的功能；
3. 设计两位数相加的全加器；

要求：完成各个文件的设计输入、编译、仿真，管脚分配，下载，验证。（4 学时）