数字逻辑电路/计算机组成原理与系统结构

实验指导书

**2016.10**

目录

[第1章 QuartusII软件使用介绍 - 1 -](#_Toc466281520)

[1.1 QuartusⅡ图形输入法 - 1 -](#_Toc466281521)

[1.2 仿真验证 - 12 -](#_Toc466281522)

[1.3 QuartusⅡ文本输入法使用介绍 - 16 -](#_Toc466281523)

[1.4 层次化设计方法 - 18 -](#_Toc466281524)

[第2章 实验系统介绍 - 20 -](#_Toc466281525)

[2.1 实验系统组成 - 20 -](#_Toc466281526)

[2.2 实验硬件平台 - 21 -](#_Toc466281527)

[2.2.1实验硬件平台结构介绍 - 21 -](#_Toc466281528)

[2.2.2 实验硬件平台的资源简介 - 21 -](#_Toc466281529)

[2.2.3 实验硬件平台的连接 - 24 -](#_Toc466281530)

[2.3普通版实验软件使用介绍 - 24 -](#_Toc466281531)

[2.3.1 实验软件的安装 - 24 -](#_Toc466281532)

[2.3.2 本地实验 - 25 -](#_Toc466281533)

[2.3.2 远程实验 - 29 -](#_Toc466281534)

[2.4简化版实验软件使用介绍 - 30 -](#_Toc466281535)

[2.4.1 实验软件的安装 - 30 -](#_Toc466281536)

[2.4.2 实验软件的使用 - 30 -](#_Toc466281537)

[第3章 数字逻辑电路实验 - 34 -](#_Toc466281538)

[3.1 实验用资源介绍 - 34 -](#_Toc466281539)

[3.2 数字逻辑电路实验 - 34 -](#_Toc466281540)

[实验1 数据选择器 - 35 -](#_Toc466281541)

[实验2 全加器 - 36 -](#_Toc466281542)

[实验3 十进制数加法器 - 38 -](#_Toc466281543)

[实验4 译码器 - 39 -](#_Toc466281544)

[实验5 八位寄存器 - 40 -](#_Toc466281545)

[实验6 同步模4可逆计数器 - 42 -](#_Toc466281546)

[实验7 异步模8加1计数器 - 43 -](#_Toc466281547)

[实验8 节拍发生器 - 44 -](#_Toc466281548)

[3.3数字逻辑课程设计 - 46 -](#_Toc466281549)

[课题1 交通灯信号控制器设计 - 46 -](#_Toc466281550)

[课题2 数字密码锁电路设计 - 47 -](#_Toc466281551)

[课题3 自动升降电梯控制器设计 - 48 -](#_Toc466281552)

[第4章 计算机组成原理实验 - 50 -](#_Toc466281553)

[4.1 实验目的 - 50 -](#_Toc466281554)

[4.2实验用资源介绍 - 50 -](#_Toc466281555)

[4.3实验内容 - 50 -](#_Toc466281556)

[实验1 逻辑运算电路 - 50 -](#_Toc466281557)

[实验2 补码加法器 - 52 -](#_Toc466281558)

[实验3 移位器 - 53 -](#_Toc466281559)

[实验4 八位串/并进位运算器 - 54 -](#_Toc466281560)

[实验5 四位补码运算器 - 55 -](#_Toc466281561)

[实验6 控制器实验 - 57 -](#_Toc466281562)

[实验7 CPU综合实验 - 58 -](#_Toc466281563)

[第5章 计算机系统结构实验 - 61 -](#_Toc466281564)

[5.1模型计算机-Ⅰ系统结构设计 - 61 -](#_Toc466281565)

[5.1.1 实验目的 - 61 -](#_Toc466281566)

[5.1.2 实验说明 - 61 -](#_Toc466281567)

[5.1.3 实验要求 - 61 -](#_Toc466281568)

[5.1.4模型计算机-Ⅰ设计简介 - 62 -](#_Toc466281569)

[5.2基本模型机-Ⅱ系统结构设计 - 65 -](#_Toc466281570)

[5.2.1 实验目的 - 65 -](#_Toc466281571)

[5.2.2 实验说明 - 65 -](#_Toc466281572)

[5.2.3 实验要求 - 65 -](#_Toc466281573)

[5.2.4模型计算机-Ⅱ设计简介 - 66 -](#_Toc466281574)

[附录1 JYS实验系统硬件平台资源和FPGA引脚连接表 - 70 -](#_Toc466281575)

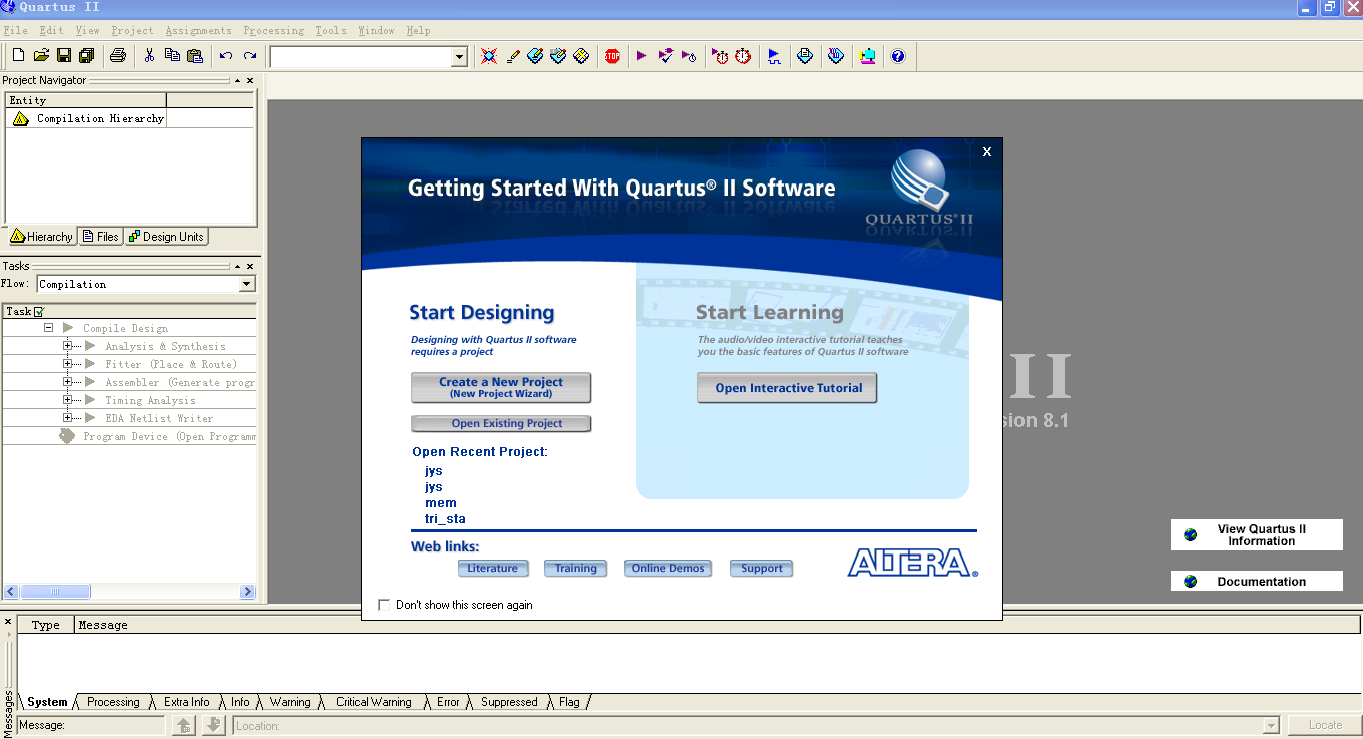
[附录2 QuartusⅡ库及库元件说明 - 72 -](#_Toc466281576)

# 第1章 QuartusII软件使用介绍

Altera公司的QuartusⅡ软件提供了可编程片上系统（SOPC）设计的一个综合开发环境，是进行SOPC设计的基础。QuartusⅡ集成环境包括以下内容：系统级设计、嵌入式软件开发、可编程逻辑器件设计、综合、布局和布线、验证和仿真等。本章将通过设计一个 **反相器** 的简单实例介绍QuartusII软件的使用过程。

## 1.1 QuartusⅡ图形输入法

在桌面上双击QuartusII图标（或从“程序”→“Altera”→Quartus），进入QuartusII项目管理器窗口如图1.1所示。



新建项目

打开原项目

图1.1 QuartusII管理器窗口

1.1.1 建立新项目

QuartusⅡ软件的工作对象是项目，一个项目（Project）是一个系统设计的总和，包含了所有的子设计文件和设计项目中的所有辅助文件，所以在进行一个逻辑设计时，首先要指定该设计的项目名称，对于每个新的项目应该建立一个单独的子目录，以后所有与该项目有关的文件都将存在这个子目录下。

1.单击图1.1中间的“新建项目”选项（单击鼠标左键，以后如无特殊说明，则含义不变），屏幕如图1.2所示。

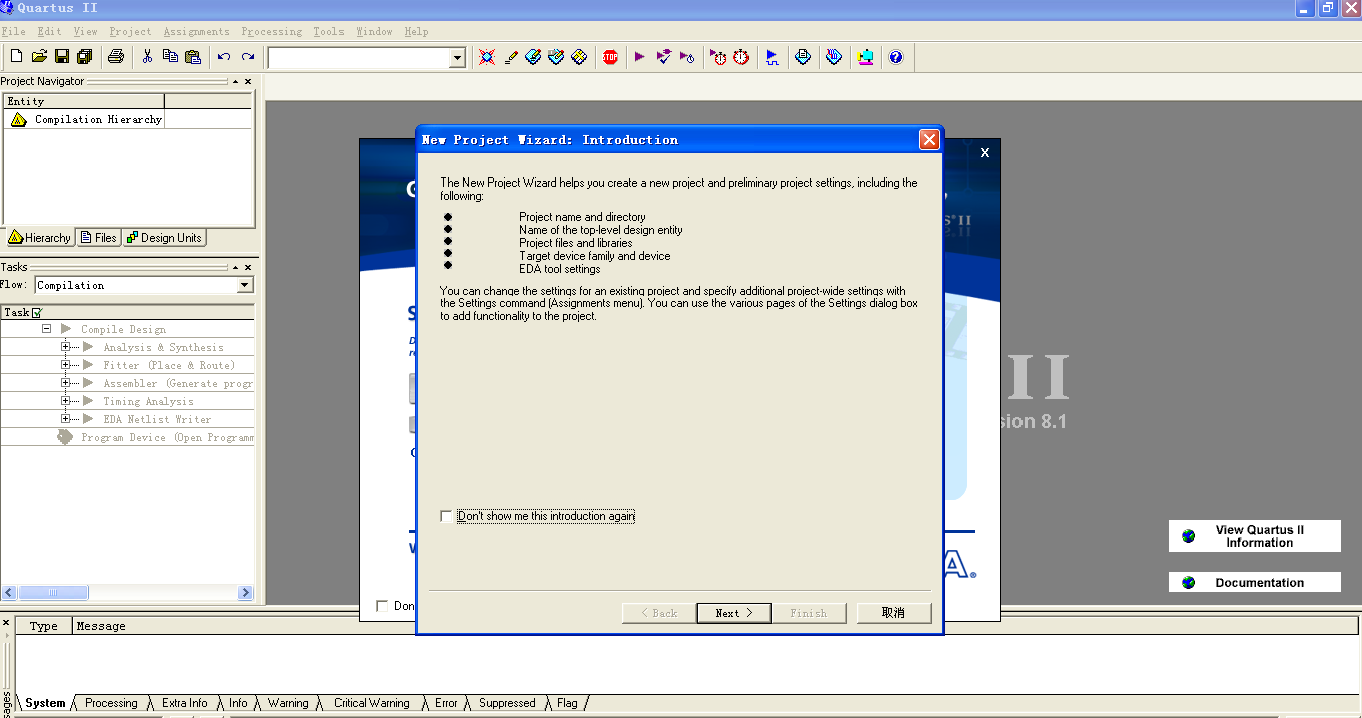


图1.2 建立新项目的对话框

2. 在图1.2中单击New Project Wizard窗口中的“NEXT”按钮，弹出如图1.3所示的对话框，从上向下分别键入新项目的文件夹名、项目名后（本例中，我们分别键入D：\MY\_EDA、MY\_NOT、MY\_NOT）。

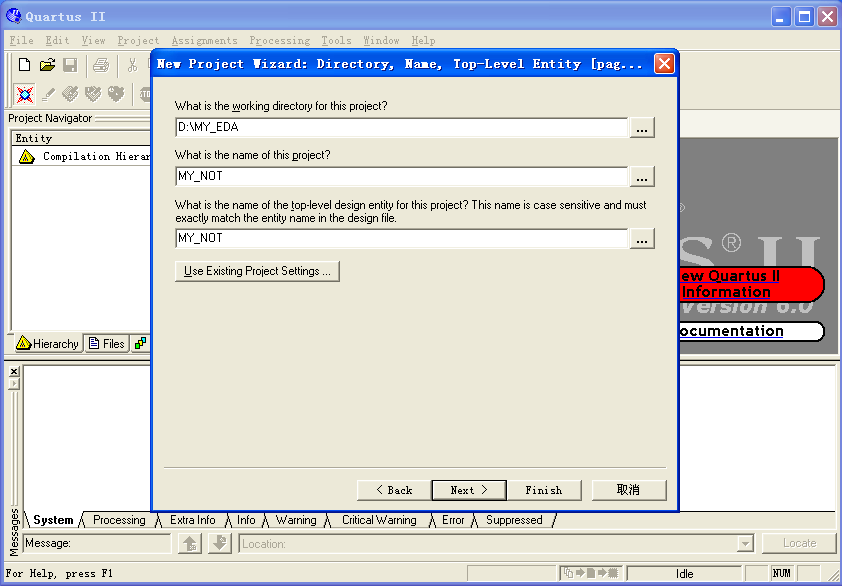


图1.3 指定项目名对话框

3. 在图1.3中，单击“Finish”按钮（如果文件夹不存在，则系统会提示用户，确认后系统会自动建立该文件夹），弹出如图1.4所示的对话框。

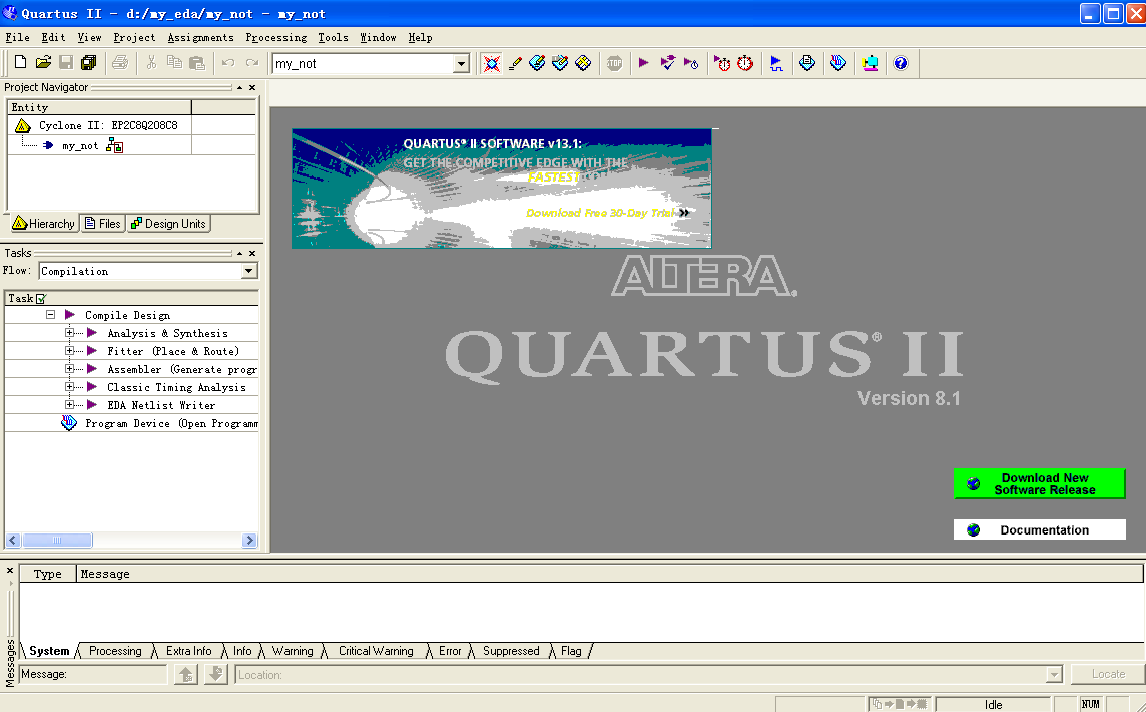


图1.4 确定项目名对话框

1.1.2建立原理图文件

1. 打开原理图编辑器。在图1.4中，单击“File”菜单→单击New选项（快捷键：Ctrl+N），弹出如图1.5所示的对话框。在该对话框中，双击（双击鼠标左键，以后如无特殊说明，则含义不变）“Block Diagram/Schematic File”选项,打开原理图编辑器，如图1.6所示。

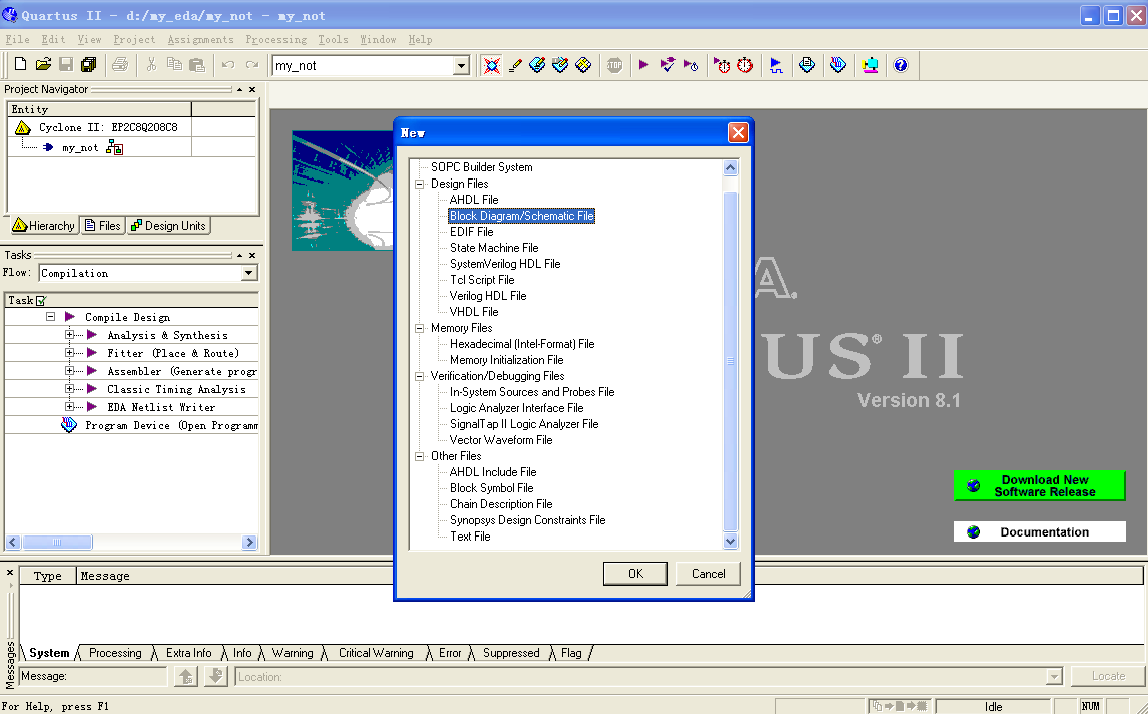


图1.5 选择图形编辑器对话框

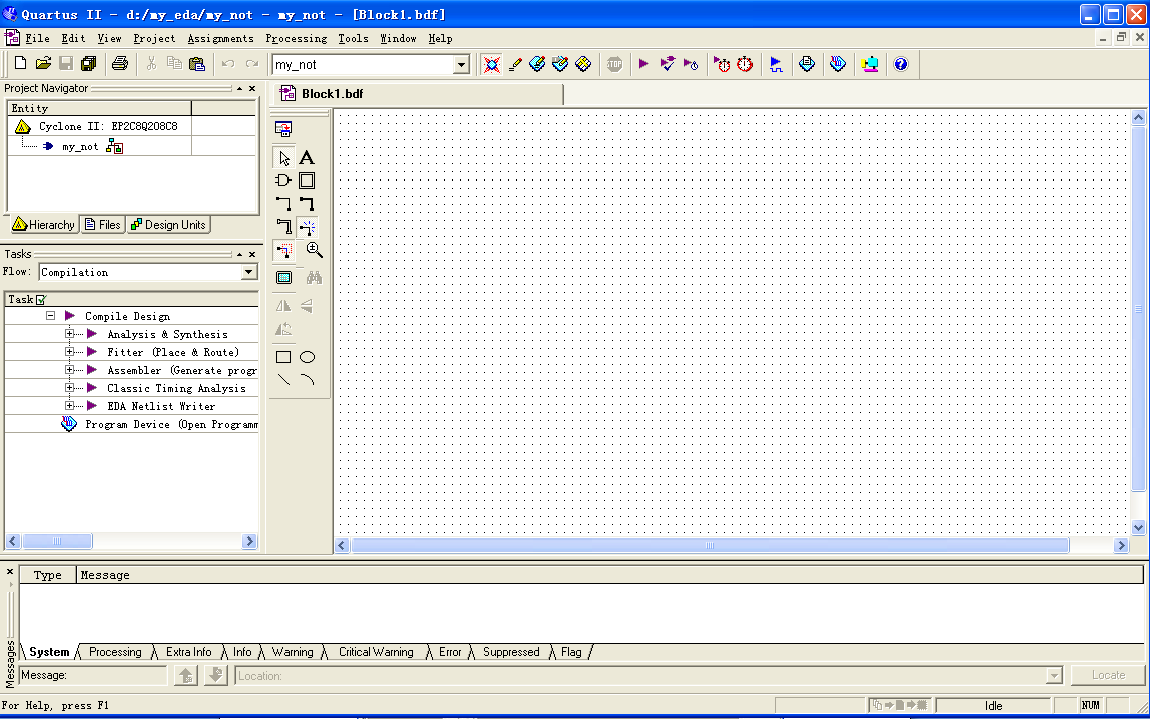


图1.6图形编辑器窗口

2. 添加元件。

(1)在图1.6中，双击图形编辑窗的空白处，弹出图1.7所示窗口。

(2)在图1.7的Name输入编辑框中输入not后，单击“Ok”按钮，弹出图1.7所示窗口。

(3)此时可以看到光标上站着被选的符号（反相器），将其移动到合适的位置单击鼠标左键，使其固定。

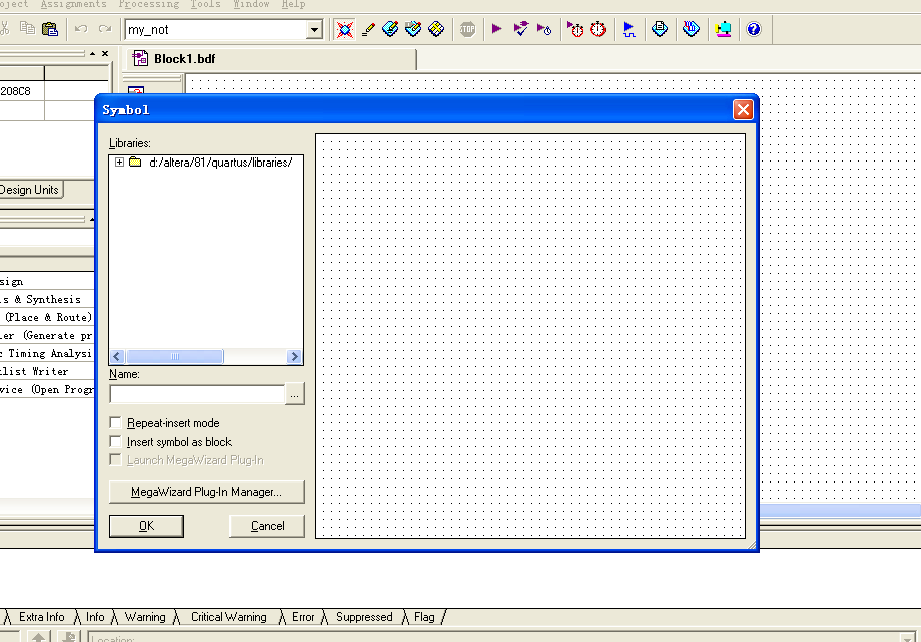


图1.7 添加反相器元件符号的窗口

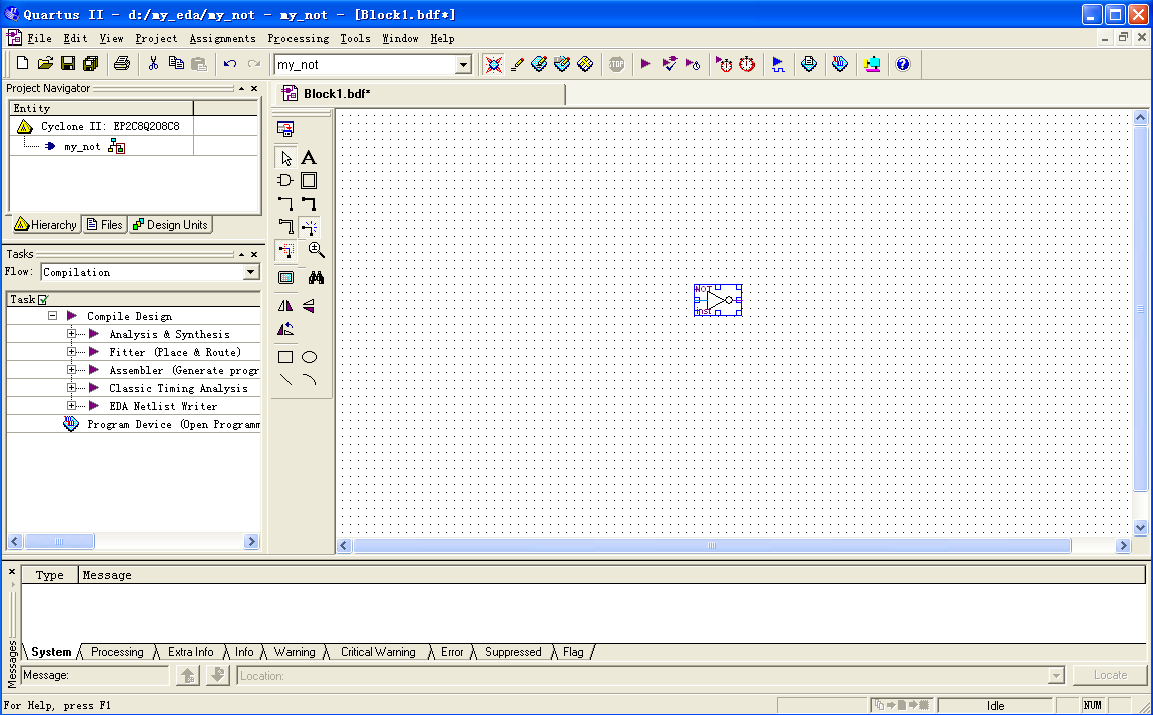


图1.8 放置反相器窗口

(4)重复(1)、(2)、(3)三个步骤，给图中放置一个input和一个output符号，如图1.9所示。

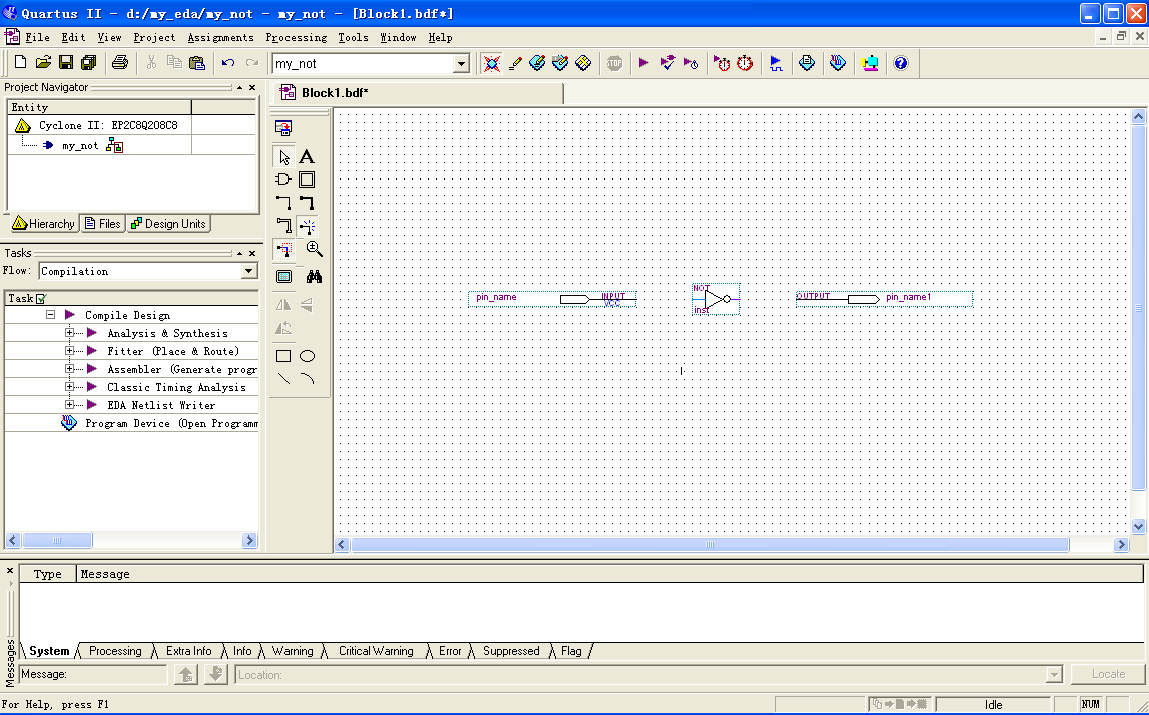


图1.9 放置了所有元件符号的屏幕

3.连线。

(1)如果需要连接两个端口，可将鼠标移到其中一个端口，这时鼠标自动变为现状，然后一直按住鼠标左键并将其拖到第二个端口，待连接点上出现蓝色的小方块后再释放鼠标左键，即可看到在两个端口之间有一条线生成，如图1.9 所示。

(2)重复步骤（1）的方法，将反相器和output连起来。完成所有连线的电路图如图1.10所示。

4.为输入/输出端口命名。

在图1.10中，双击pin\_name使其衬底变黑后，再键入信号名A并回车确认。输出端口的标记方法与此相同。本例中将输出信号命名为B。

5.保存文件。

(1)在图1.10中单击保存文件按钮，弹出如图1.11的对话框。

(13) 在图1.11中，缺省情况下“文件名（N）”的文本编辑框中为MY\_NOT,单击“保存”按钮即可已完成的设计，如图1.12所示。

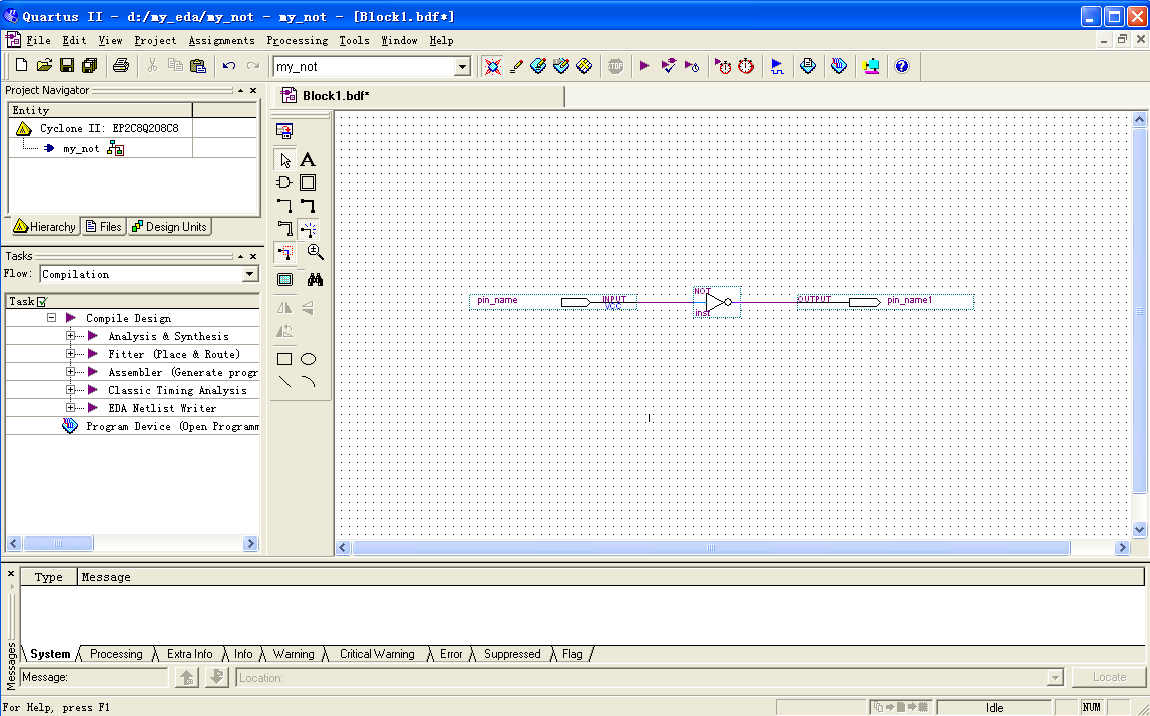


图1.10 完成全部连线的窗口

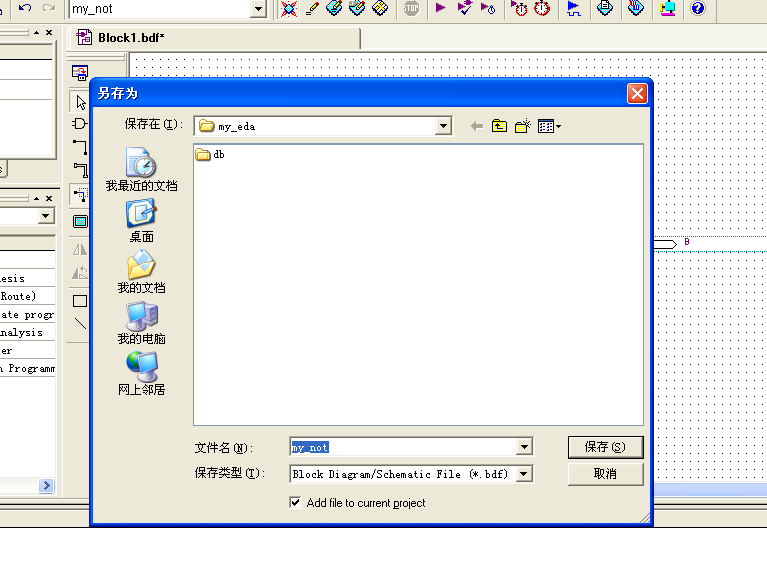


图1.11 欲保存文件前的窗口

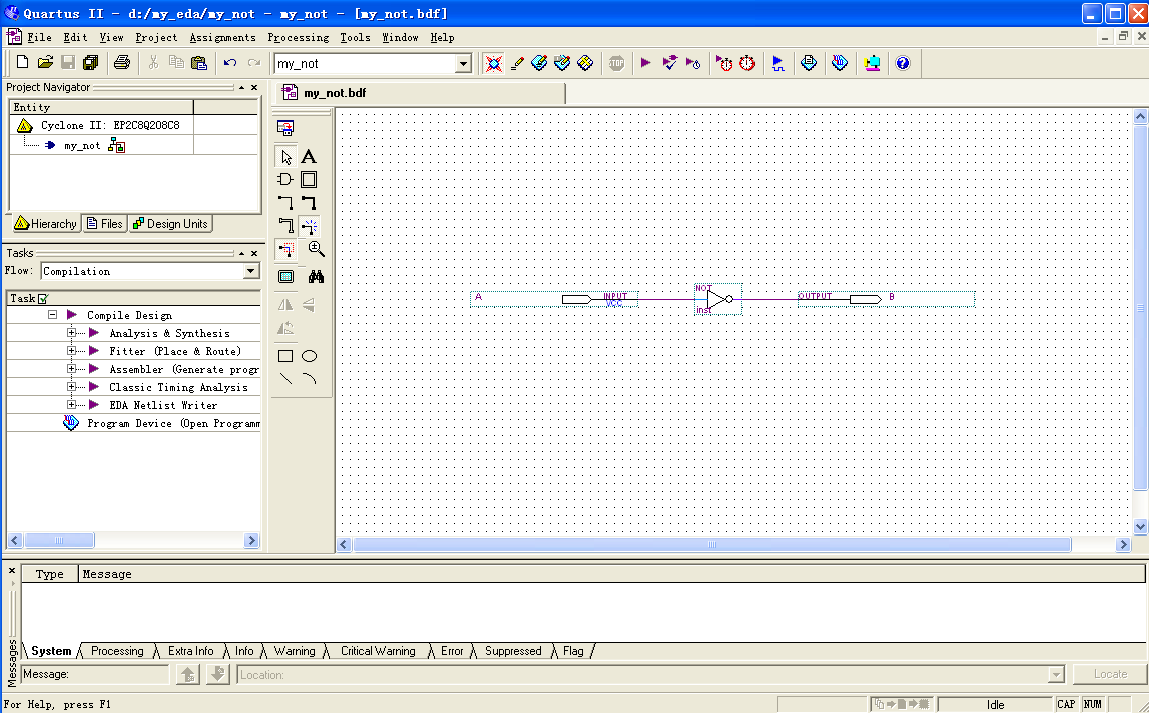


图1.12 文件已保存的窗口

1.1.3 目标器件选择

对于进行的设计项目通常需要设计人员自己指定欲使用的目标器件型号，选择目标器件的步骤如下：

1.单击图1.12中的Assignments菜单，选中Device……选项，如图6.13所示。

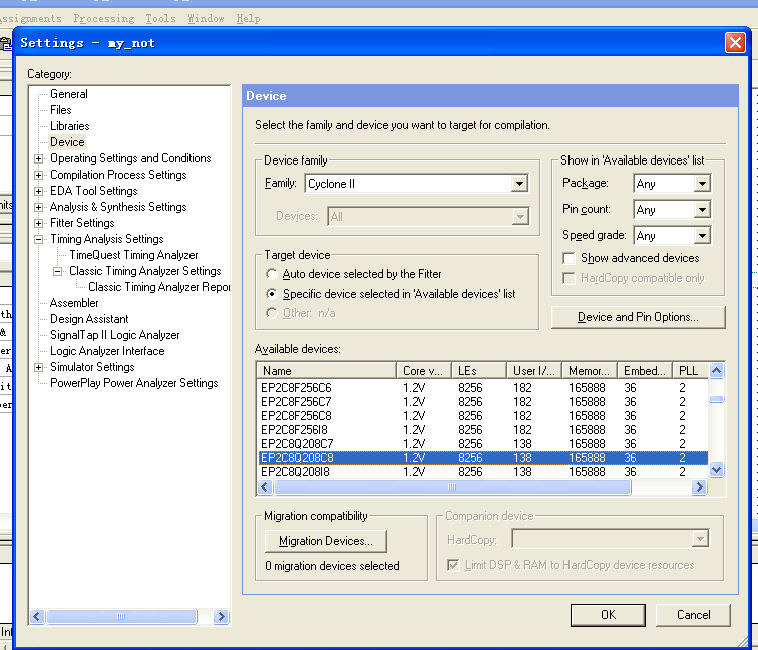


图1.13 选择器件的窗口

2.图1.13中，在Family列表中选择器件系列，在Available device框中选择器件型号，本实验使用的器件系列：Cyclong II，器件型号：EP2C8Q208C7，选择无误后单击“OK”按钮返回图1.12窗口。

3.在图1.13中，还可以单击“Device & Pin Options”按钮，在弹出的“Device & Pin Options”窗口中，单击“Unused Pins”对未使用引脚进行处理，如图1.14所示。

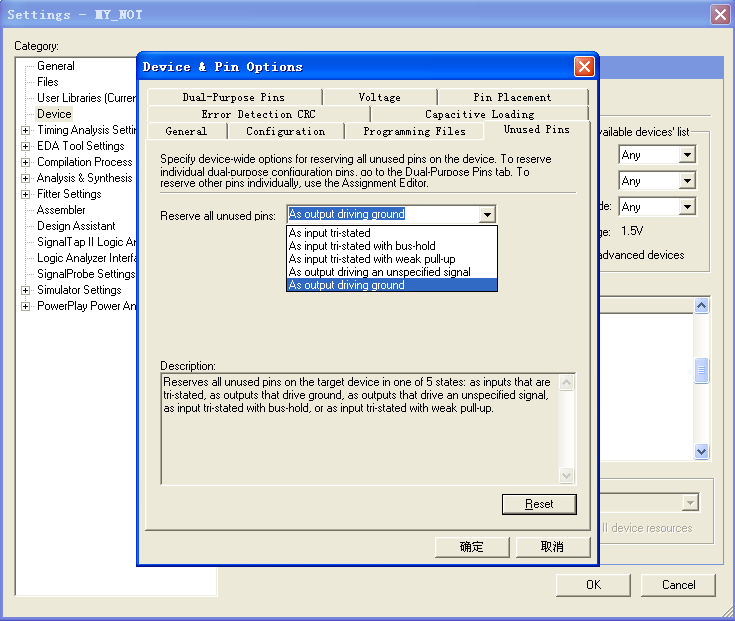


图1.14 未使用引脚的处理

3.在图1.14中选择“As input tri-stated”（避免未使用引脚对系统中其他元器件造成影响，保证系统可靠工作，一般可将未使用引脚设定为三态输入方式）后，单击“确定”按钮，返回图1.13所示窗口。

1.1.4设计项目的编译

QuartusⅡ软件的编译器包括多个独立的模块。各模块可以单独运行，也可以选择Processing－>Start Compilation命令启动全编译过程。

1. 在图1.12中，单击水平工具条上的编译按钮（形似向右的小三角），或者使用Processing菜单中的Start Compilation选项开始编译，并伴随着进度不断地变化屏幕，编译完成后的窗口如图1.15所示。

2. 在图1.15中单击“确定”按钮后，进入图1.16所示窗口。在该图中显示了编译时的各种信息，其中包括警告和出错信息。根据信息提示，如果有错，则再做相应的修改。重复步骤1，直到没有错误提示为止。

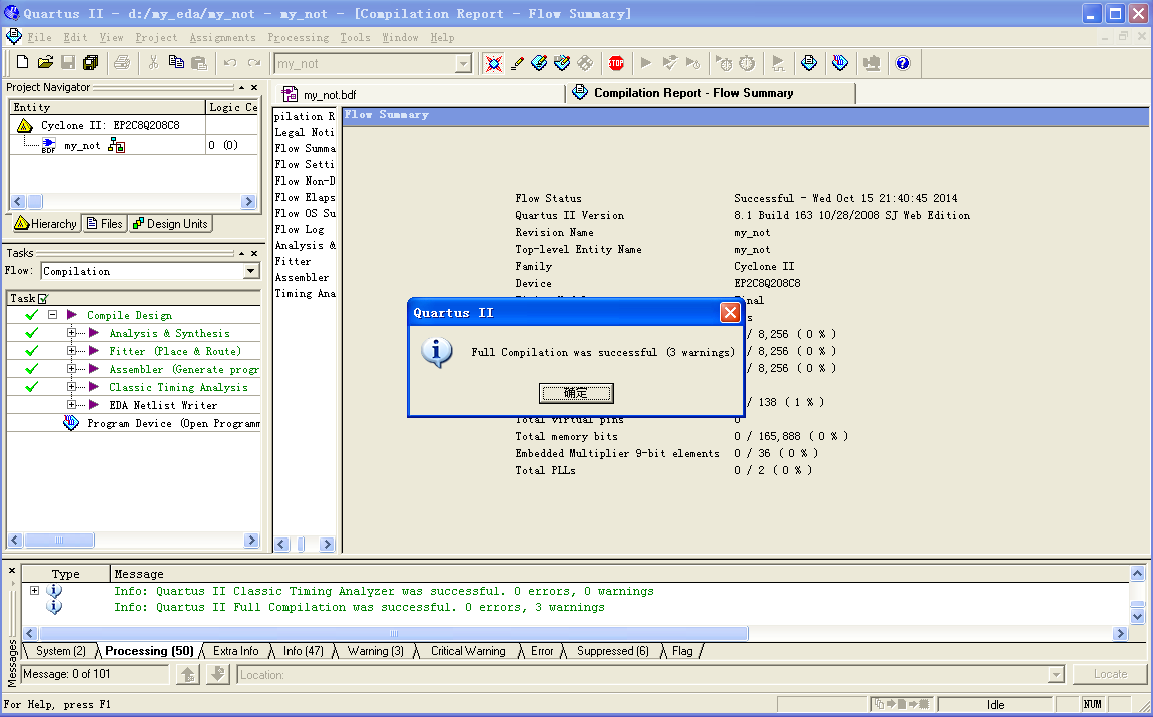


图1.15 编译完成的窗口

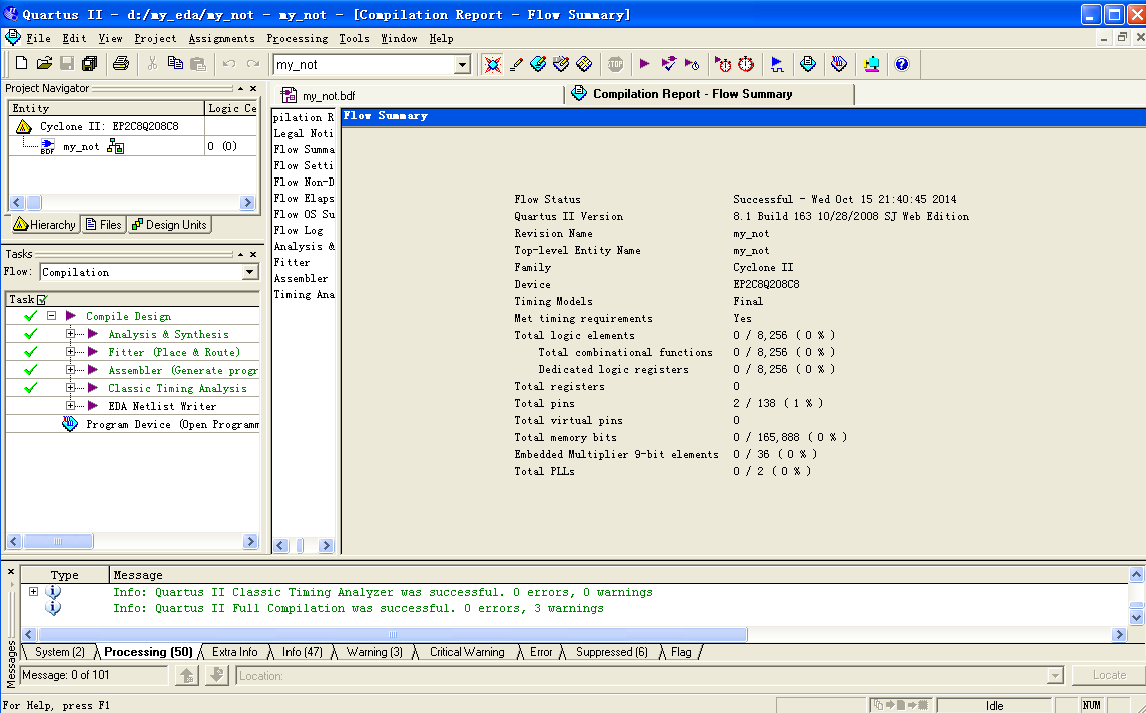


图1.16 编译完成后的信息窗口

1.1.5 引脚分配

在选择好合适的目标器件，完成设计的分析综合过程并得到工程的数据库文件之后，需要对设计中的输入、输出引脚指定具体的器件引脚号码，指定引脚号码称为引脚分配或引脚锁定。

1.在图1.12窗口中单击菜单Assignments命令，在弹出的下拉菜单中选择Pins选项，进入如图1.17所示引脚分配窗口。

2. 在图1.17中选中引脚“A”，双击“Location”列的蓝色矩形框（与本引脚处于同一行），在弹出的引脚列表中选择合适的引脚，也可直接键入引脚号码，如图1.18所示。

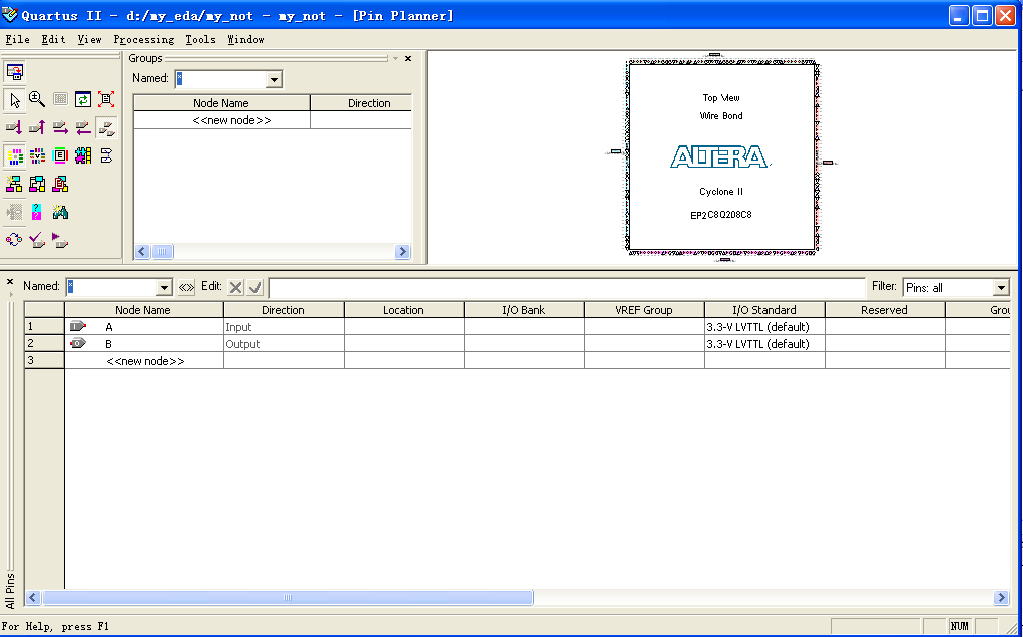


图1.17 引脚分配窗口

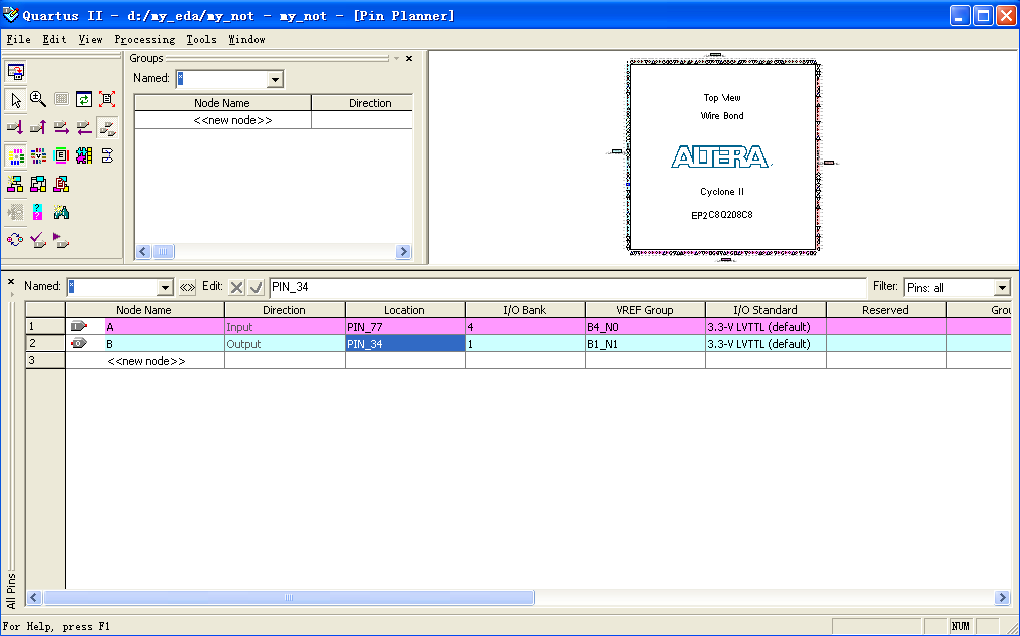


图1.18 指定引脚时的屏幕显示

3. 重复步骤2，完成所有引脚的指定，如图1.18所示，本例中输入A接开关K0(77pin）,输出B接发光二极管LD0（34pin）,保存引脚分配结果，退出引脚分配界面。

4. 引脚全部指定完毕之后，需重新进行一次编译，如正确将得到和图1.15的信息窗口。

1.1.6编程下载

使用QuartusⅡ软件完成设计项目的编译之后，将产生以目标器件的编程器对象文件（.pof）或SRAM对象文件（.sof）的形式保存的文件，QuartusⅡ软件的编程器（Programmer）使用该文件对器件进行编程或配置。

1. 在图1.12窗口中，单击下载按钮或是使用Tools－>Programmer命令，编程器窗口自动打开，如图1.19所示。

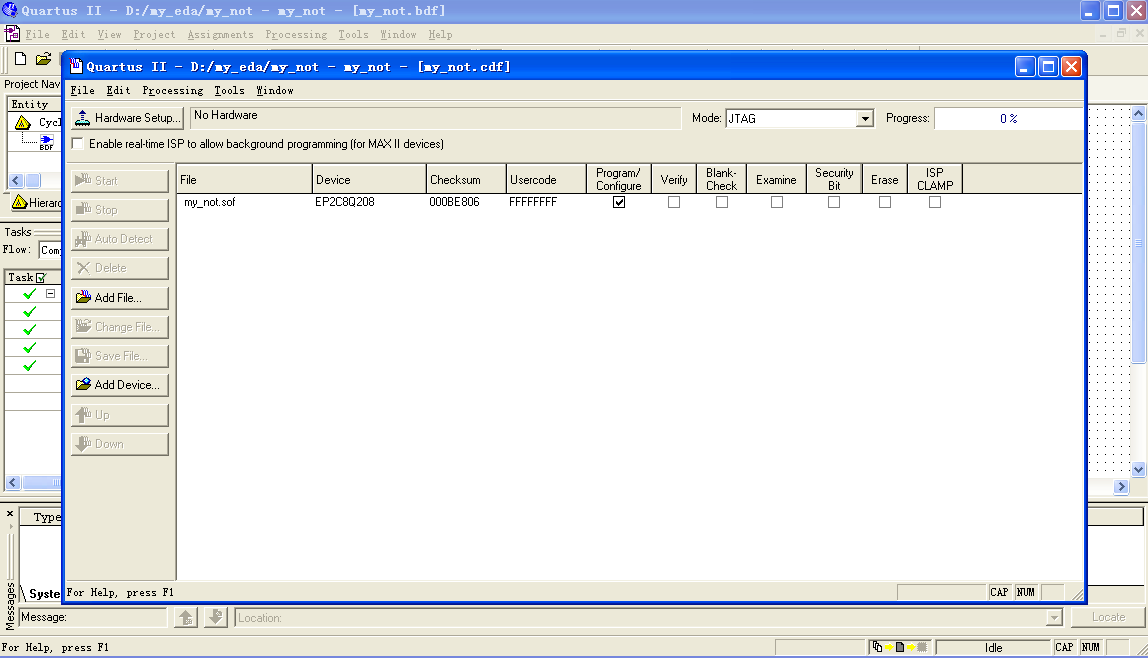


图1.19 编程器窗口

2.在图1.19窗口中，单击“Start”按钮，计算机就开始下载编程文件，开始下载后，屏幕上的进度条以百分数表示下载进度。下载完成后如图1.20所示。

注意：下载前应检查图1.19窗口中的欲下载的文件名、器件型号和下载模式是否正确、硬件编程器是否存在等，只有上述正确才可以开始下载。

本例中，编程文件为MY\_NOT.sof，它是由QuartusII在编译时自动生成的；以后的实验中下载模式全部选择JTAG。

3. 在图1.19窗口中，如系统没有找到硬件编程器，则需要自行添加。添加硬件编程器的方法是单击  按钮，按图1.21所示窗口设置即可。

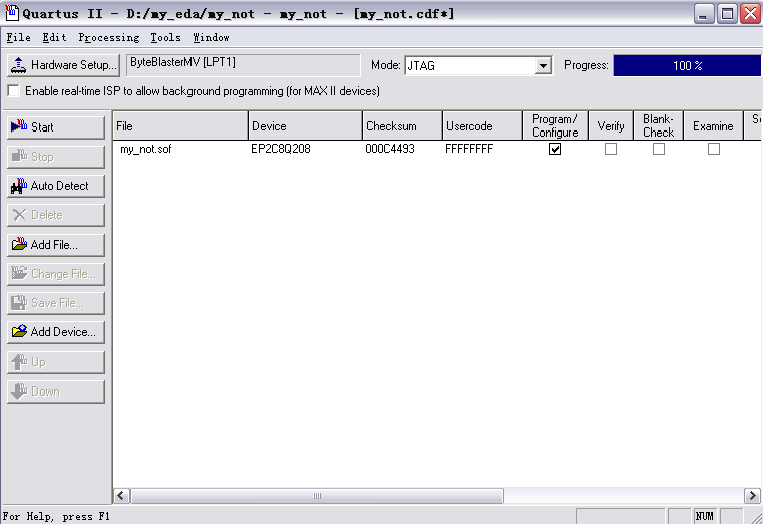


图1.20 下载完成窗口

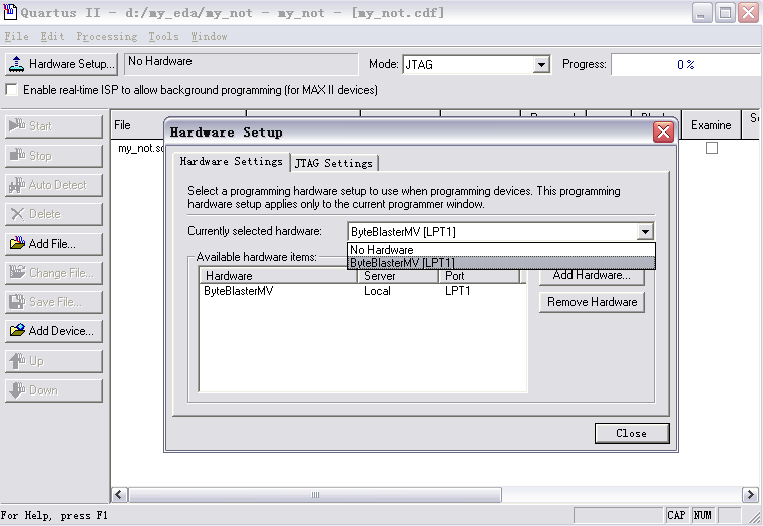


图1.21 “Hardware Setup”窗口

4. 图1.19中，点击“mode”下拉窗口，选择JTAG项即可，如图1.22所示。

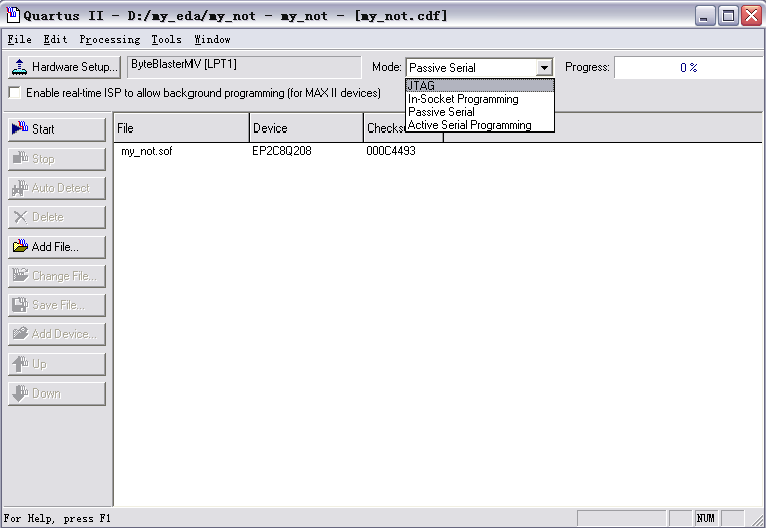


图1.22 下载模式选择窗口

5.拨动开关K0，即可通过发光二极管LD0验证设计结果。

1.1.7 符号图的生成

在层次化设计中，一个项目工程往往是另外一个项目工程的子项目，该子项目在总的工程项目中只是一个实现某种功能的符号图。因此，在完成一个子项目工程时，要产生一个可供顶层项目工程使用的符号图。

1. 在图1.12窗口中，单击“File”菜单，选择Create/Update－>Create Symbol File For Current File命令，如图1.23所示

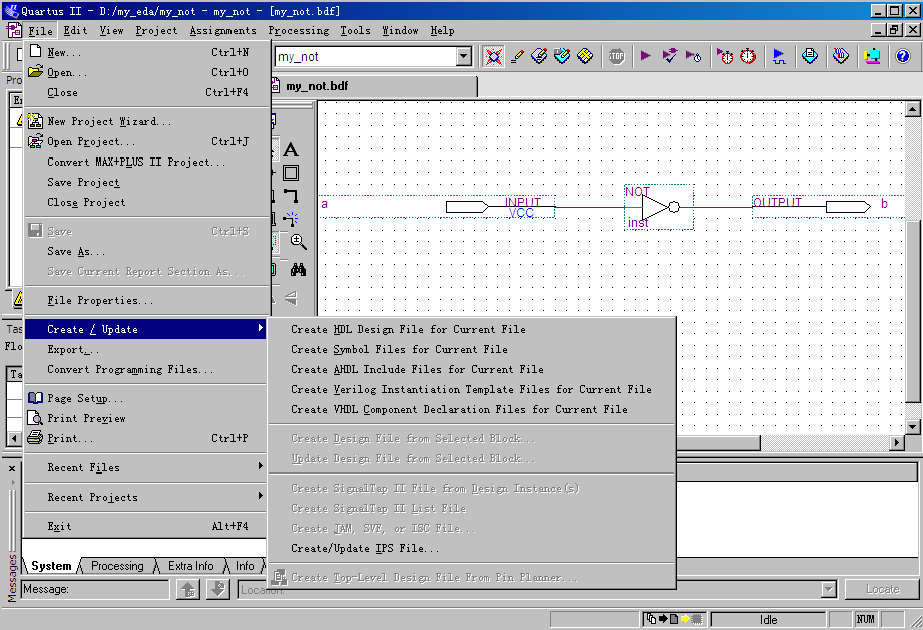


图1.23 选择符号图生成命令窗口

3. 按图1.23选择后弹出符号图保存窗口，确定后弹出生成成功的提示，如图1.24所示。

4. 在图1.24中，单击“确定”按钮，完成了该设计的符号图生成。

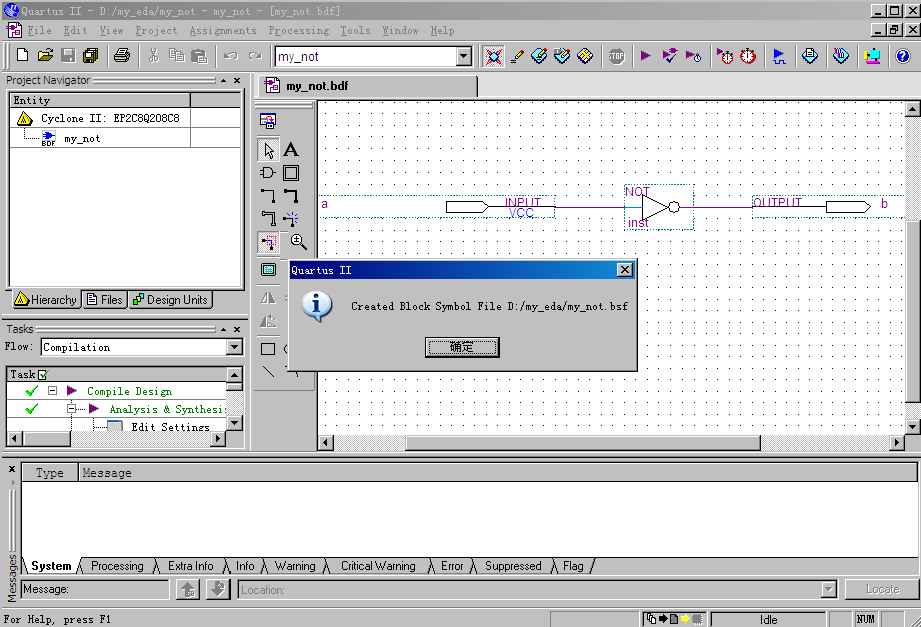


图1.24 符号图生成成功的提示窗口

1.2 仿真验证

仿真验证是EDA设计技术的重要特征。在完成了设计项目的输入、综合以及布局布线等步骤以后，则可以使用EDA仿真工具或QuartusⅡ仿真器对设计项目的功能与时序进行仿真，以检查设计结果正确如否。下面介绍在QuartusⅡ仿真器中对设计项目进行仿真验证的方法。

1. 建立仿真波形文件。

（1） 在图1.12中，单击File/New菜单命令，弹出New对话框，选中Other Files标签，如图1.25所示。

(2)在图1.25中选择Vector Waveform File选项后，单击“OK”按钮，进入图1.26所示波形编辑窗口。

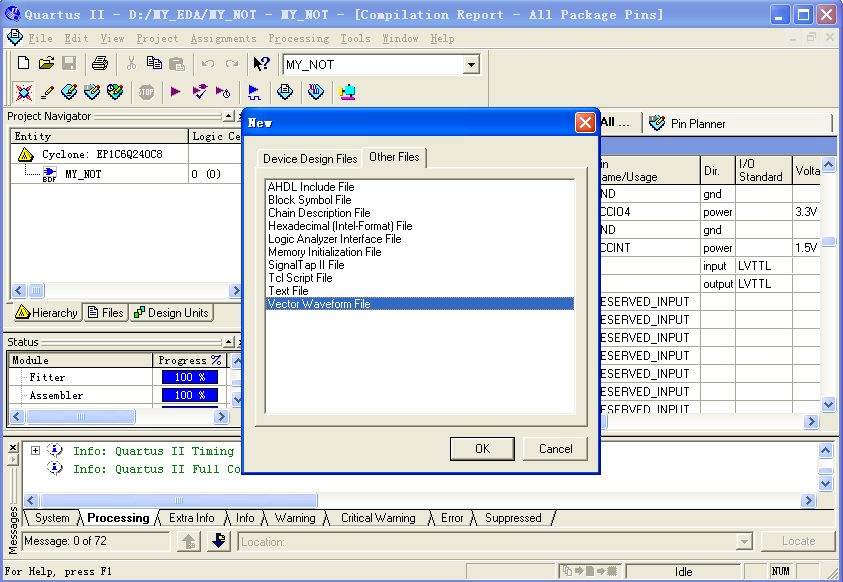


图1.25 Other Files标签

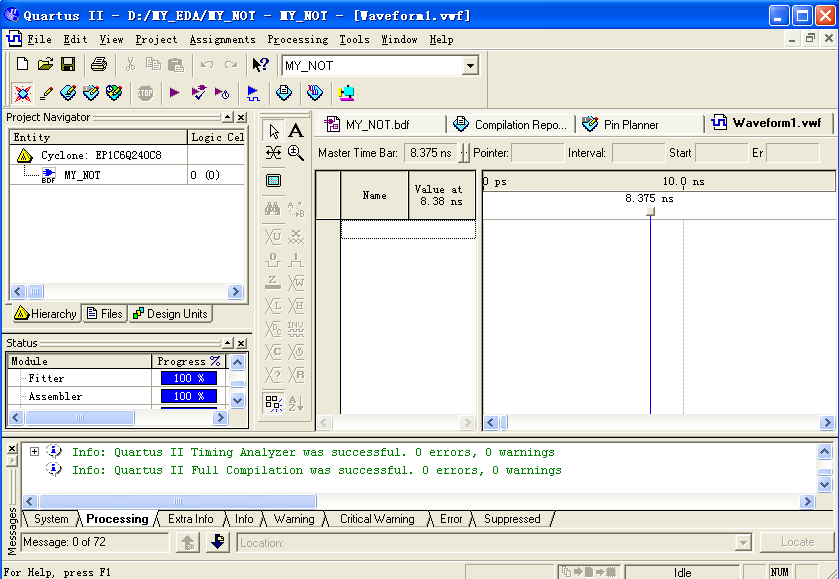


图1.26 编辑仿真文件窗口

(3)在图1.26窗口中，双击“Name”下方的空白处，弹出“Insert Nod or Bus”对话框，单击该对话框“Node Finder...”按钮后，进入Node Finder窗口，如图1.27所示。

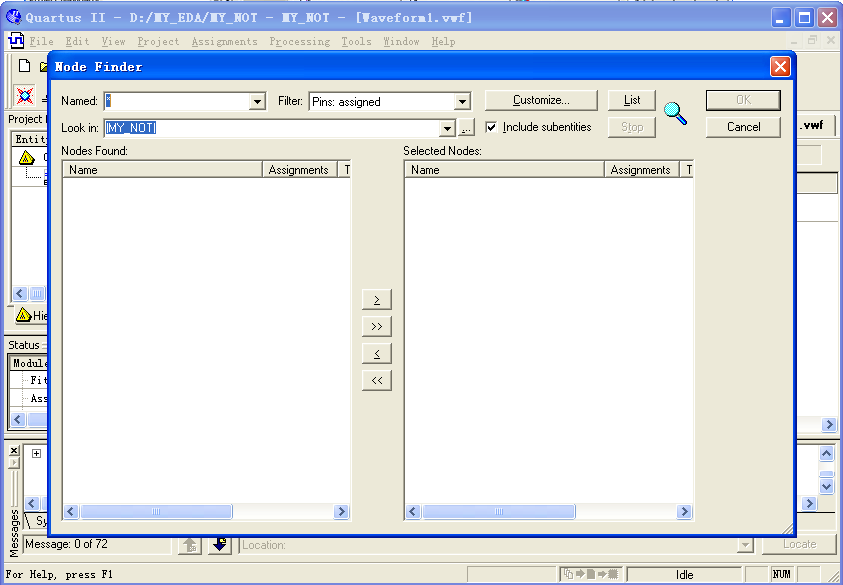


图1.27 “Node Finder”对话框

(4)在图1.27界面中单击“List”按钮，设计项目中的全部输入/输出管脚在左边窗口列出，如图1.28所示。

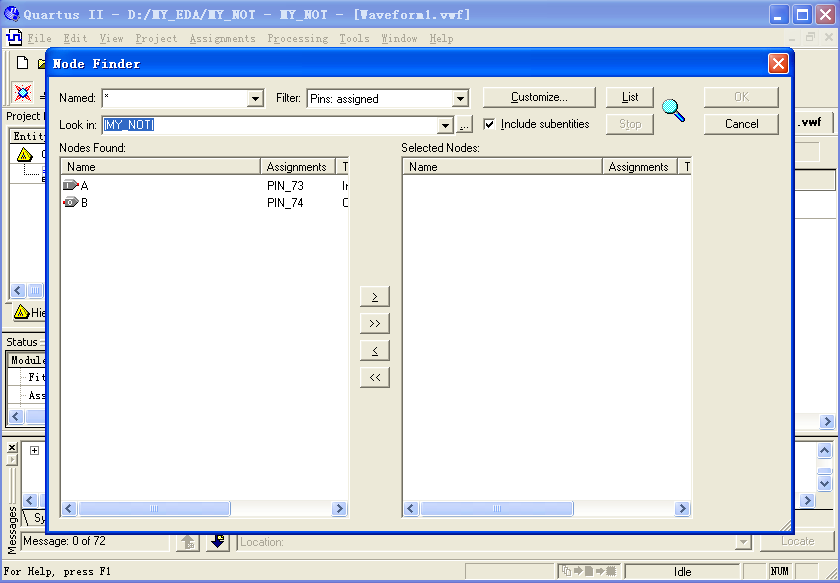


图1.28 “Node Finder”对话框

(5)在图1.24中，选中欲观察的输入或输出引脚，单击“≥“按钮，将其拷贝到右边窗口中。如果要将左边窗口中所有输入/输出都被拷贝到右边窗口中，可直接单击“>>”按钮，如图1.29所示。

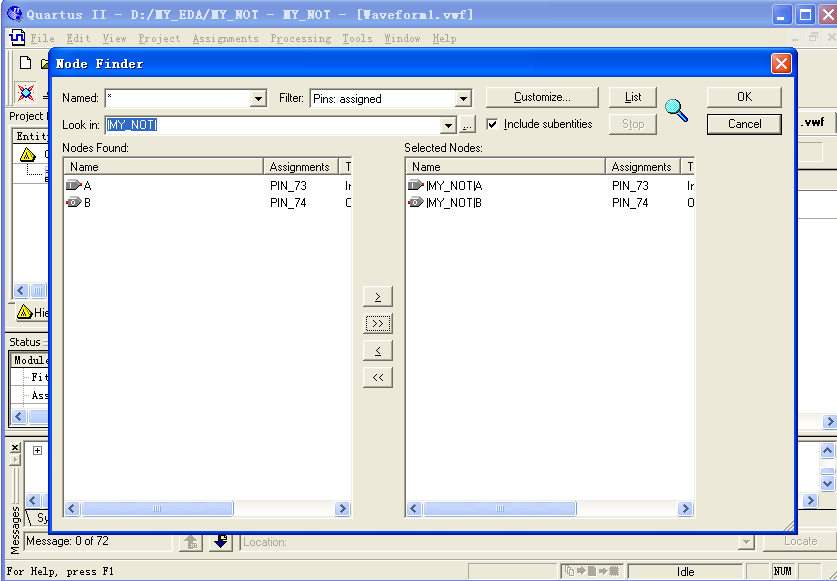


图1.29 选择欲观察的输入/输出对话框

(6) 在图1.29中，完成全部引脚选择后单击“OK”按钮，返回图1.30所示界面。

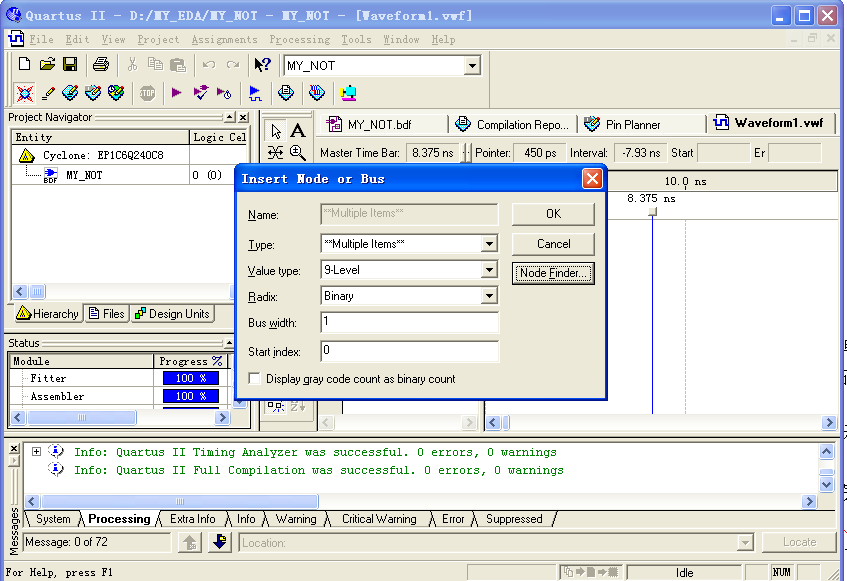


图1.30输入/输出引脚选择完成后的窗口

(7) 在图1.30窗口中，单击“OK”按钮，返回图1.31所示的波形编辑窗口。

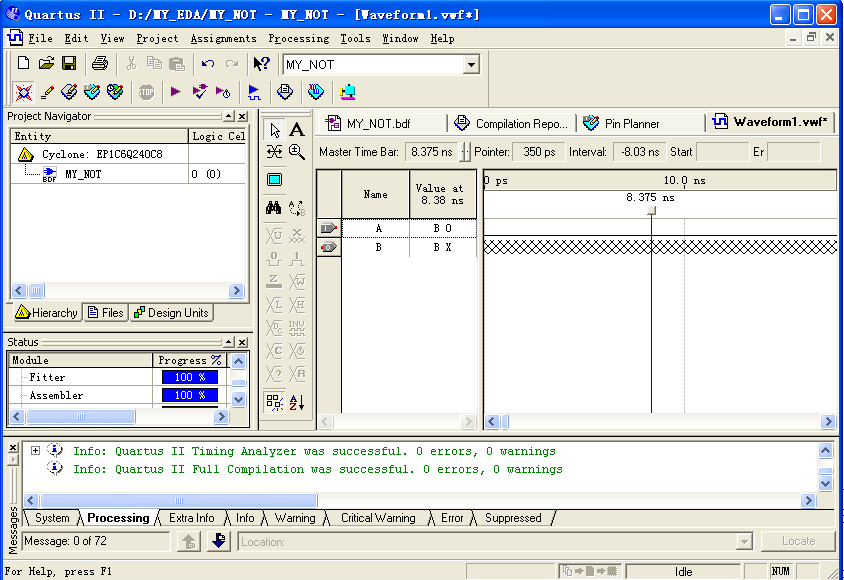


图1.31 波形编辑窗口

(8) 在图1.31中，单击“Name”栏下方的“A”，即选中该行编辑输入波形。再单击工具栏中的“”按钮，屏幕将弹出“Clock”对话框，此时可以修改信号的周期、相位和占空比，然后故单击“OK”按钮，输入信号A的波形添加完毕，如图1.32。

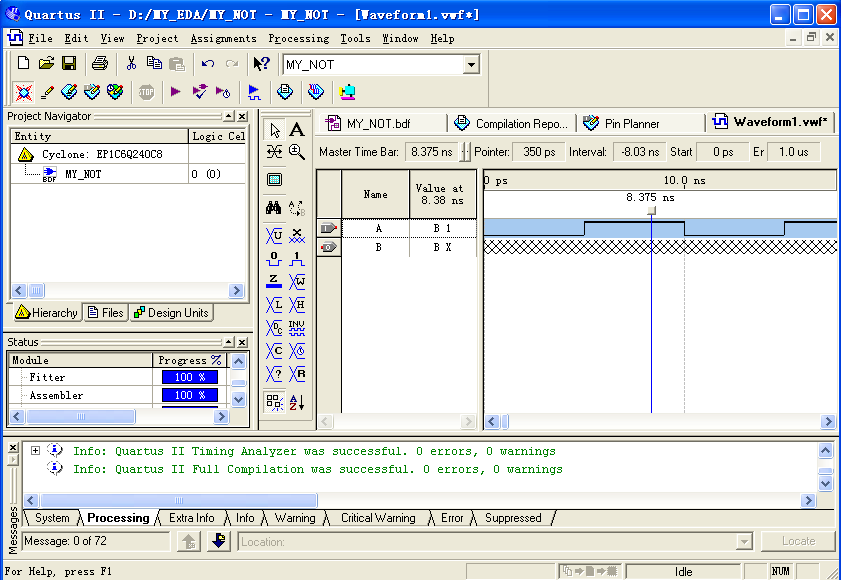


图1.32 完成仿真波形编辑的窗口

(9) 保存建立的波形文件。

2. 设计的仿真验证

(1) 仿真器设置。选择Assignments－>Settings...命令，在Settings对话框的Category列表中选择Simulator Settings，然后在弹出的界面中进行功能仿真或时序仿真等设置。

（2）启动仿真器。完成仿真器设置后，在图1.32中窗口中，单击仿真“”按钮或者使用Processing－>Start Compilation命令启动仿真器，并有进度提示。完成后会弹出一个“QuartusII”对话框，表明仿真是否成功，本例的仿真结果如图1.33所示。

(3)分析仿真结果。从图1.33中可以看出功能是正确的。

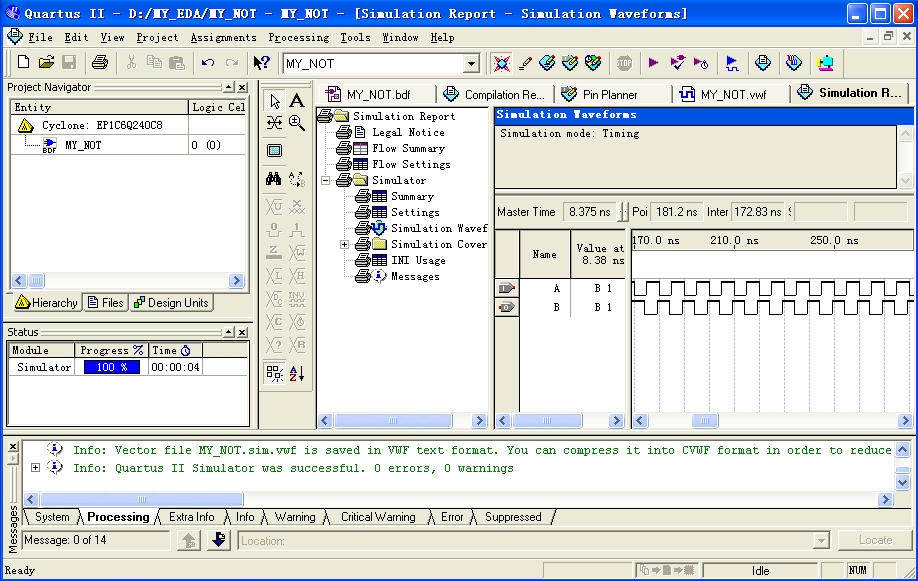


图1.33 仿真完成后的窗口

## 1.3 QuartusⅡ文本输入法使用介绍

QuartusⅡ的文本编辑器是一个非常灵活的编辑工具，用于以AHDL、VHDL和Verilog HDL语言形式以及Tcl脚本语言输入的文本型设计。下面介绍硬件描述语言（HDL）形式的文本输入方法。

1.4.1 建立新项目

在图1.4窗口中选择File－> New Project Wizar...命令，按前述原理图输入法中建立新项目的过程，建立一个文件夹名、项目名、顶层实体名分别为MY\_EDA、VHDL\_nor、VHDL\_nor的新项目，如图1.34所示。

1.4.2 建立文本文件

1. 打开文本编辑器

在创建好新项目后，选择File－>New...菜单，在弹出的文件选择窗口（如图1.5所示）中选择Device Design File标签下的VHDL File，单击OK，进入文本编辑器窗口，如图1.35所示。

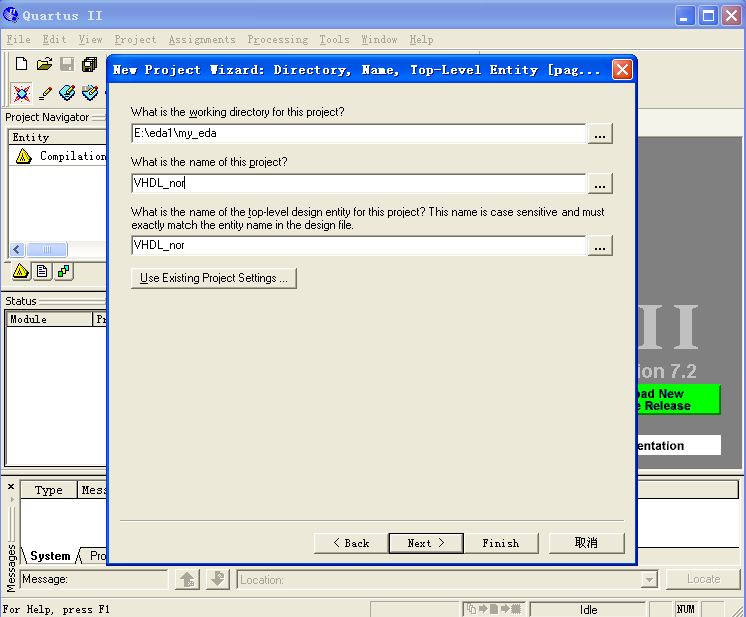


图1.34 建立新项目窗口

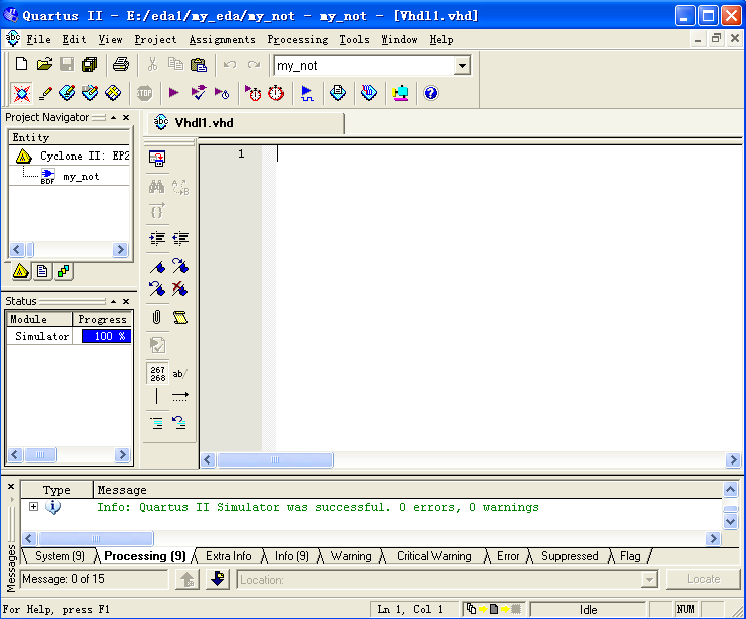


图1.35 空白的文本编辑器窗口

2.编辑文本文件

在对文本文件进行编辑时，文本编辑器窗口的标题名称后面将出现一个（\*），表明正在对当前文本进行编辑操作，存盘后星号消失。

在图1.35的文本编辑窗中可以直接输入HDL程序描述语句，此处我们实现一个二输入的“或非门”如图1.36所示。

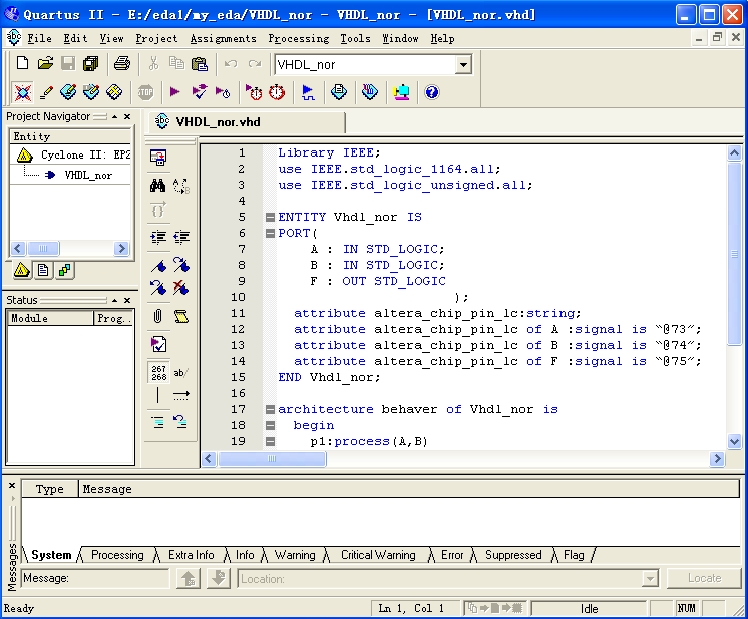


图1.36 完成文本输入后的窗口

3. 保存文本设计文件。

为了便于读者输入文本文件，下面给出VHDL语言的源文件“

LIBRARY IEEE;

USE IEEE.std\_logic\_1164.all;

USE IEEE.std\_logic\_unsigned.all;

ENTITY VHDL\_nor IS

PORT( A : IN STD\_LOGIC;

B : IN STD\_LOGIC;

F : OUT STD\_LOGIC );

END VHDL\_nor;

architecture behaver of VHDL\_nor is

begin

p1:process(A,B)

begin

if (A=‘0’and B=‘0’) then F <=‘1’;

else F <=‘0’;

end if;

end process p1;

end architecture behaver;

其余过程和原理图输入方法完全相同，大家可参照1.1节中的步骤完成仿真、器件选择、编程下载和在实验平台上进行验证。

## 1.4 层次化设计方法

前面几节中，我们都是使用单一的描述方法来实现各种电路功能，其所实现的功能相对来说很简单，像这样简单的工程项目，在实际中几乎不存在。一般情况下，一个实际的工程项目都是由很多功能模块互相连接而构成，其中部分模块还可能由子模块构成。因此层次化的设计方法在实际工程项目的实现中应用非常广泛。下面以1.1节和1.3节中设计的两个项目作为新项目的底层子模块说明层次化设计的基本步骤，设计项目本身没有实际意义。

1. 建立名为MIXED的工程项目并进入原理图编辑器窗口，如图1.37所示。

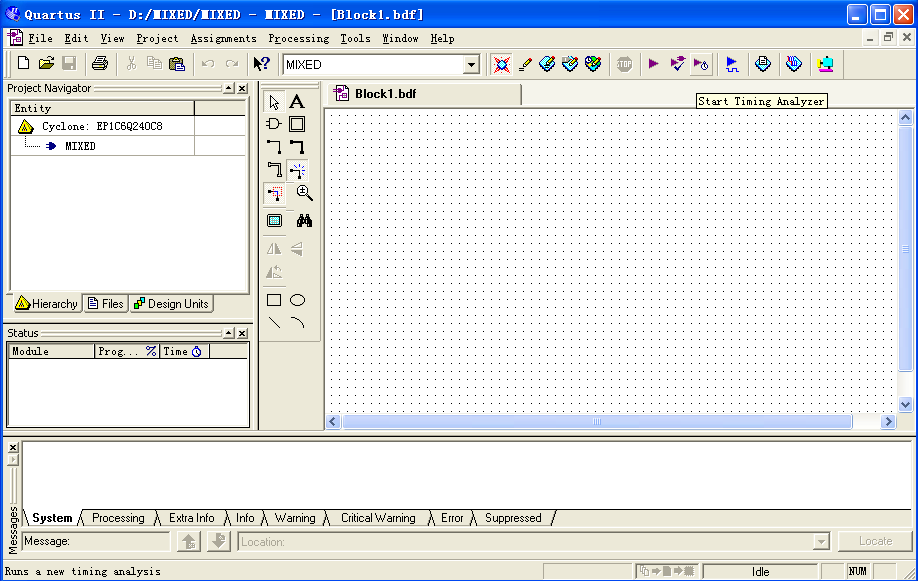


图1.37 建立MIXED工程项目窗口

2. 打开用户工作库。在图1.37中原理图编辑器的编辑窗口的空白处双击鼠标左键，从弹出的器件库选择窗口中点击Project打开用户工作库，可以看到前面生成的二个器件my\_not和my\_nor如图1.38所示。

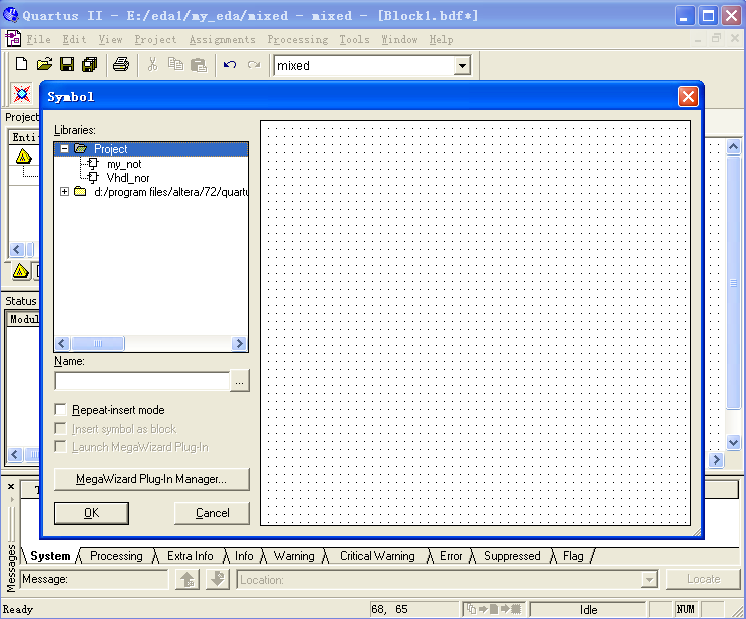


图1.38用户工作库窗口

3. 完成原理图编辑。在图1.38中，选择my\_not、my\_nor、input、output部件，完成原理图编辑，如图1.39所示。

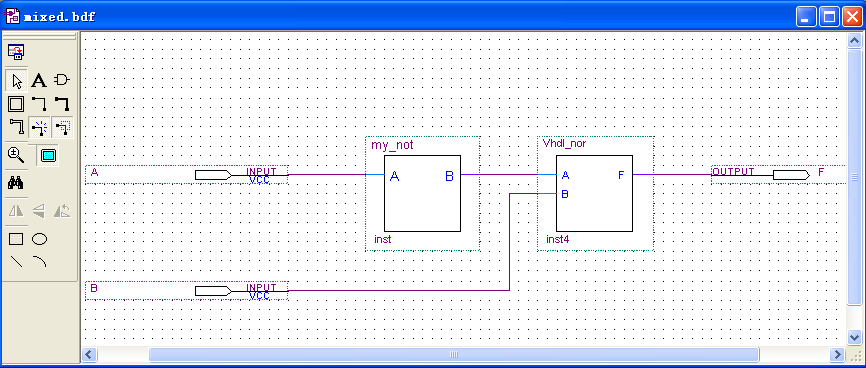


图1.39 完成后的mixed原理图

4.器件选择和器件引脚定义。根据前面所述方法，选择合适的器件系列和器件型号并完成引脚分配。

5.项目编译。启动编译器完成项目编译。

6.项目仿真。仿真结果如图1.40所示。

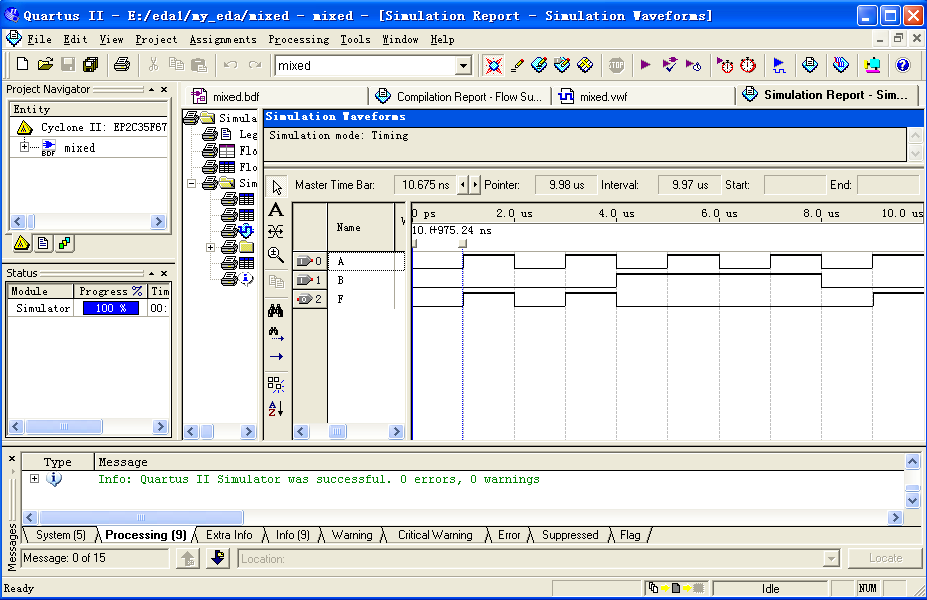


图1.40 mixed项目的仿真波形

7.下载并验证电路功能，A、B、C、D分别接逻辑开关，F接逻辑指示器，本节中的举例在于了解“自底而上”的设计过程，例题本身没有实用价值。

本章仅仅是QuartusII入门的基础，所涉及的内容很少。要做到灵活使用的目的，还需要大量的练习，并在使用中不断总结经验，提高自己的操作技能。

# 第2章 实验系统介绍

## 2.1 实验系统组成

EDA(Electronic Design Automation)是电子设计自动化的英文缩写。它是指以计算机为工具，在EDA软件平台上，根据设计者描述的源文件（可以是原理图文件、语言文件或波形图文件），自动完成系统的设计，包括编译、仿真、综合、优化、布局布线以及对特定器件的适配。设计者的任务仅限于利用软件的方式来完成对系统硬件功能的描述。尽管目标系统是硬件，但整个设计和修改过程如同完成软件设计一样方便。

在数字逻辑电路、计算机组成原理、计算机系统结构和现代数字系统设计中,现场可编程门阵列（FPGA）和电子设计自动化（EDA）技术的使用越来越广,掌握FPGA与EDA技术是从事电子系统设计、计算机硬件开发和研究人员的必备技能,也是电子技术、计算机类专业学生学习的重要内容。

为了提高教学质量，增强计算机组成原理实验系统的功能，降低实验成本，提高系统的灵活性，本实验系统采用先进的EDA技术，设计出一套基于FPGA的数字逻辑与计算机组成原理实验系统，采用该系统学习可利用FPGA和QUARTUSII软件快速、方便地设计出计算机的各种部件，利用硬件实验平台和软件实验平台完成实验调试。

该实验系统可用于数字逻辑电路、计算机组成原理和计算机系统结构等课程的实验教学，同时也适用于本科毕业设计、智能化设计、新产品开发等应用。

本实验系统主要由实验硬件平台和实验软件平台二大部分组成，其中实验软件平台由服务器管理软件、本地实验软件和远程实验软件组成。本地实验用于在本地PC机上进行实验，远程实验用于通过网络进行实验，服务器管理软件用于对远程实验进行管理，包括实验者身份验证、实验时间记录和实验硬件平台使用情况记录等。由如图2.1所示。

服务器管理软件

远程实验软件

本地实验软件

超大规模

集成电路

（FPGA）

**实验软件平台**

**实验硬件平台**

图2.1 实验系统结构框图

该实验系统可用于完成数字逻辑电路、计算机组成原理、计算机系统结构等课程的**实验设计**、**实验调试**和**模型机实验**。

下面分别对实验硬件平台和实验软件平台进行详细介绍。

## 2.2 实验硬件平台

### 2.2.1实验硬件平台结构介绍

硬件平台结构如图2.2所示，该平台主要包括发光二极管指示灯（Ⅰ区）、FPGA电路（Ⅱ区）、ROM、RAM存储器电路（Ⅲ区）、开关信号输入电路（Ⅳ区）、电源电路（Ⅴ区）和单片机及外围电路（Ⅵ区）等部分。同时图中给出了电源开关、实验模式选择开关、连续脉冲输入选择插座、单脉冲输入按键、单片机复位按键、FPGA复位按键（CPU复位）、FPGA配置插座、RS232串口插座等接口的相对位置，便于学生的实验操作。

### 2.2.2 实验硬件平台的资源简介

硬件实验平台资源包括平台监控资源和提供给用户的实验资源二部分，如图2.3所示。用户实验资源如下：

1. FPGA电路：使用超大规模集成电路FPGA芯片EP2C8Q208C8,作为用户完成相关实验的硬件资源。

2. 时钟：实验平台上分别设有一个多频率连续时钟发生器和一个单脉冲信号发生器。连续时钟发生器可提供2Hz、4Hz 、8Hz 、16Hz、32Hz 、64Hz 、128Hz、256Hz 、512Hz 、1024Hz 和2048Hz，占空比为1:1的脉冲，其频率可通过改变短路块位置选择；单脉冲信号发生器由单脉冲按键产生的宽度为20ms的无抖动负脉冲，常用于单脉冲输入以及存储器的读写信号。

3. RAM存储器：采用静态的2K×8位的RAM，用于存放指令和数据。

4. ROM存储器：用于存放系统的微程序。

5. 发光二极管指示灯：共4组（LR15 ~ LR0、LD15 ~ LD0、LA15 ~ LA2 、L23 ~ L0）70只发光二极管。其中LR15 ~ LR0、LD15 ~ LD0、LA15 ~ LA2共46只可由用户随意使用（LA1、 LA0由系统占用，用户不能使用）；L23 ~ L0用于显示微指令寄存器内容，即当前执行的微指令。

6. 开关信号输入：共24只拨动开关（K23 ~ K0），用于提供高低电平输入。

7. 复位信号输入：包括单片机复位和CPU复位二个按键。前者用于单片机的复位，后者用于FPGA的复位。

8. 接口插座：包括RS232串口插座、FPGA配置插座和RJ45网络接口插座三个。

RS232串口插座用于本实验平台和PC机之间的通信；

FPGA配置插座用于本地实验和远程实验时对FPGA进行配置；

RJ45网络接口插座用于远程实验时和INTERNET网络的连接。

**Ⅳ**

**ROM1**

**ROM2**

**ROM3**

**RAM**

**单片机**

**232串口插座**

**单片机复位 CPU复位 单脉冲**

**远程 本地**

**连续脉冲选择**

**…..**

**…..**

**..**

**..**

**..**

**..**

**..**

**..**

**FPGA**

**FPGA配置插座**

**……….**

**……….**

**电源开关**

**Ⅰ**

**Ⅱ**

**Ⅲ**

**Ⅴ**

**Ⅵ**

**实验模式选择**

**开放式计算机组成原理与结构**

**实验系统**

**短路块**

（和DL1、DL3电缆连接）

（和DL2电缆连接）

图2.2 硬件实验平台结构示意图

指示灯

**L7-0**

指示灯

**L15-8**

指示灯

**L23-16**

**FPGA**

**EP2C8Q208C8**

**单时钟**

**指示灯**

**LA15-0**

**ROM1**

**单片机**

**开关K23-16**

**开关K15-8**

**开关K7-0**

**远程模块**

**配置**

**接口**

**复位**

**连续时钟**

**指示灯**

**LD15-0**

**指示灯**

**LR15-0**

**RS232**

**接口**

**RJ45**

**接口**

**ROM2**

**ROM3**

**RAM**

2.3 硬件实验平台主要资源示意图

### 2.2.3 实验硬件平台的连接

为了能够顺利完成相关实验，使用者应根据本地实验还是远程实验对硬件实验平台和PC机及网络进行正确的连接。

系统共提供了二条独立的10线并口连接电缆DL1（FPGA配置电缆）、10线串口连接电缆DL2（PC机通信电缆）和一条固定于硬件实验平台左侧电缆盒内的10线连接电缆DL3（FPGA配置电缆）。

本地实验时，将DL1连接于PC机并口和实验硬件平台的FPGA配置插座上、DL2连接于PC机串口和实验平台的RS232串口插座上。

远程实验时，将DL3连接于实验平台的FPGA配置插座上，另外通过实验平台的RJ45插座将实验平台和网络连接。

## 2.3普通版实验软件使用介绍

### 2.3.1 实验软件的安装

双击**“计算机组成原理与系统结构.exe”**开始实验软件的安装，安装过程中只要在出现“NEXT”选项时点击即可自动完成安装。

安装成功后会在桌面上添加“计算机组成原理与系统结构.exe”快捷方式图标，在桌面上双击“计算机组成原理与系统结构.exe” 快捷方式图标即可进入实验软件平台，然后单击右下角的“开始”按钮进入图2.4所示界面，该平台包括本地实验（PC机和硬件平台直接连接）和远程实验（PC机和硬件平台通过网络连接）二部分。下面主要介绍本地实验软件的使用方法，远程实验软件的使用方法见“远程实验软件使用说明书”。

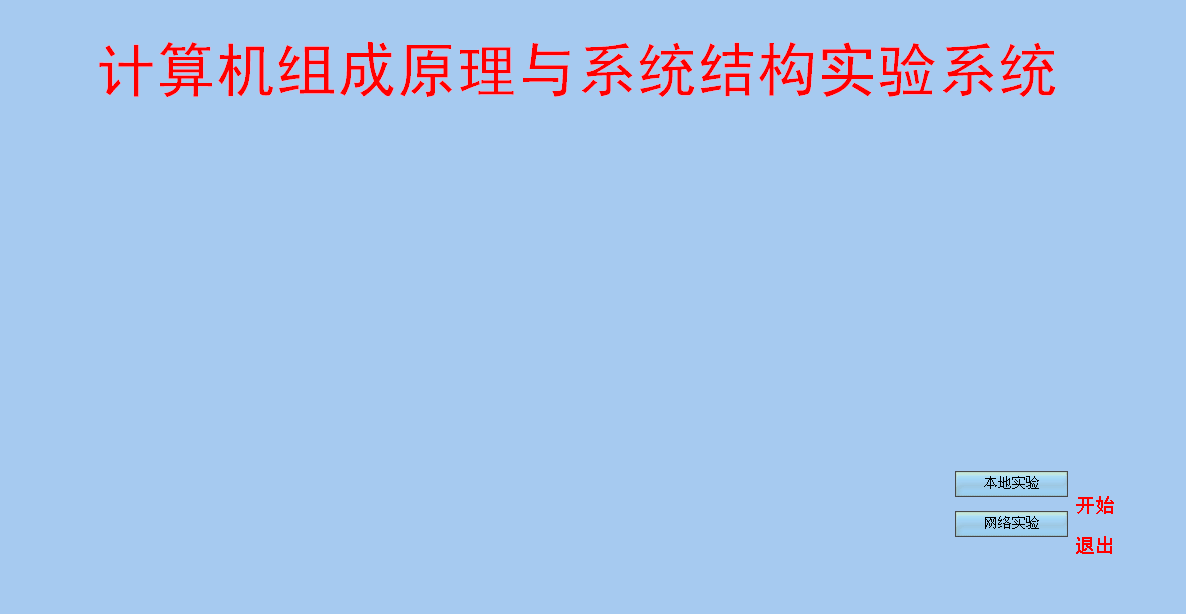


图2.4 实验程序开始界面

### 2.3.2 本地实验

单击桌面**实验程序**图标，进入图2-4实验程序开始界面，继续单击“本地实验”，进入图2.5所示的本地实验主界面。

图2.5中包括**文件**、**编辑、实验设计**、**实验调试**、**模型机实验**、**虚拟实验**、**设置**和**帮助**八个菜单项和33个快捷键，菜单项主要功能如下：

**◆ 实验设计：**点击该菜单项将进入Quartus II应用程序，完成数字逻辑电路、计算机组成原理、计算机系统结构和现代数字系统设计课程的实验设计。

**◆ 实验调试：**该菜单项包括“**电路调试、ROM写入、RAM写入**”三个子菜单，其中**电路调试**是检查和调试下载到实验硬件平台的实验设计结果是否正确；**ROM写入**和**RAM写入**是对实验硬件平台上的ROM存储器和RAM存储器根据需要进行数据写入。

**注意：**“电路调试”功能一般仅用于远程实验，本地实验可以直接在实验硬件平台上进行电路的检查和调试。

**◆ 模型机实验：**完成计算机工作过程演示、计算机组成原理和计算机系统结构模型机的设计。

**◆ 虚拟实验：**预留功能扩展菜单，暂无法使用。

**◆ 文件、编辑、设置和帮助菜单**：与一般应用软件的功能相同，不再详述。

下面详细介绍实验设计、实验调试和模型机实验三个菜单项的使用方法。

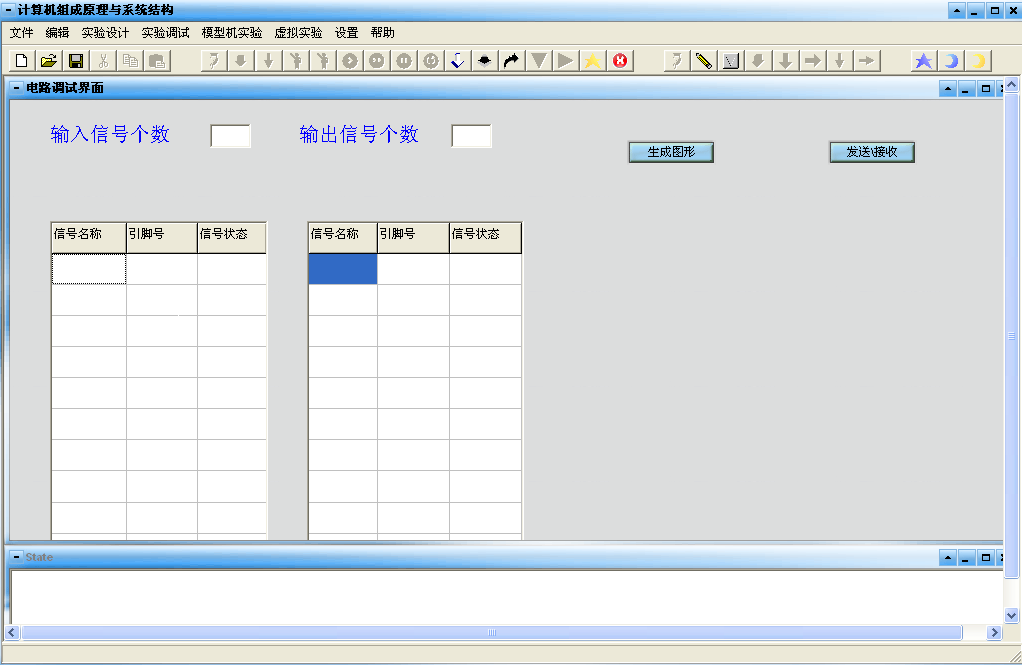
****

图2.5 实验程序主界面

**１. 实验设计**

（1）在图2.5中，点击“实验设计”进入Quartus II，如图2.6所示。

（2）在Quartus II（Quartus II使用方法参考第1章）下按要求完成相关的实验电路设计后，将设计结果下载到实验硬件平台的FPGA中，结束实验设计。

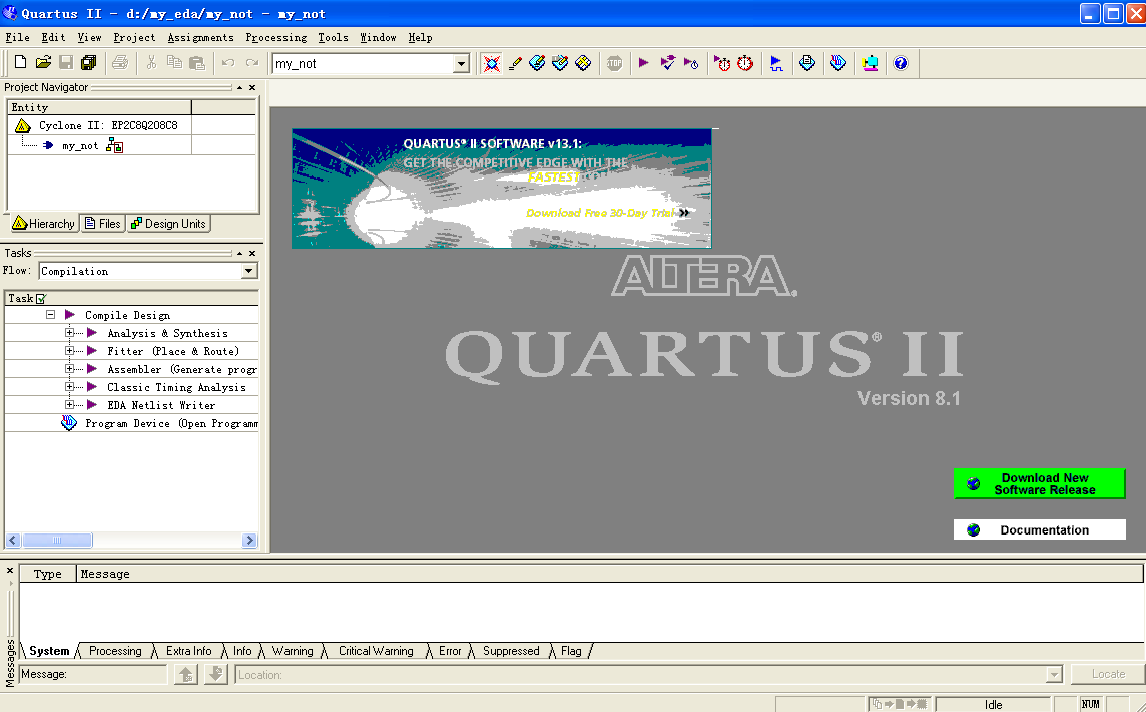
****

图2.6 Quartus II实验设计程序界面

**2. 实验调试**

“实验调试”包括“**电路调试、ROM写入、RAM写入**”三个子菜单，使用方法如下：

**（1）电路调试：**

◇ 单击“实验调试 🡪 电路调试”进入电路调试程序界面，如图2.7所示。



图2.7 电路调试界面

◇ 确定实验的输入和输出信号：将您所设计实验的输入信号个数和输出信号个数输入到图2.7上部的**输入信号个数**和**输出信号个数**窗口中，并在其下方表中添加输入信号和输出信号的信号名称，如图2.8所示。

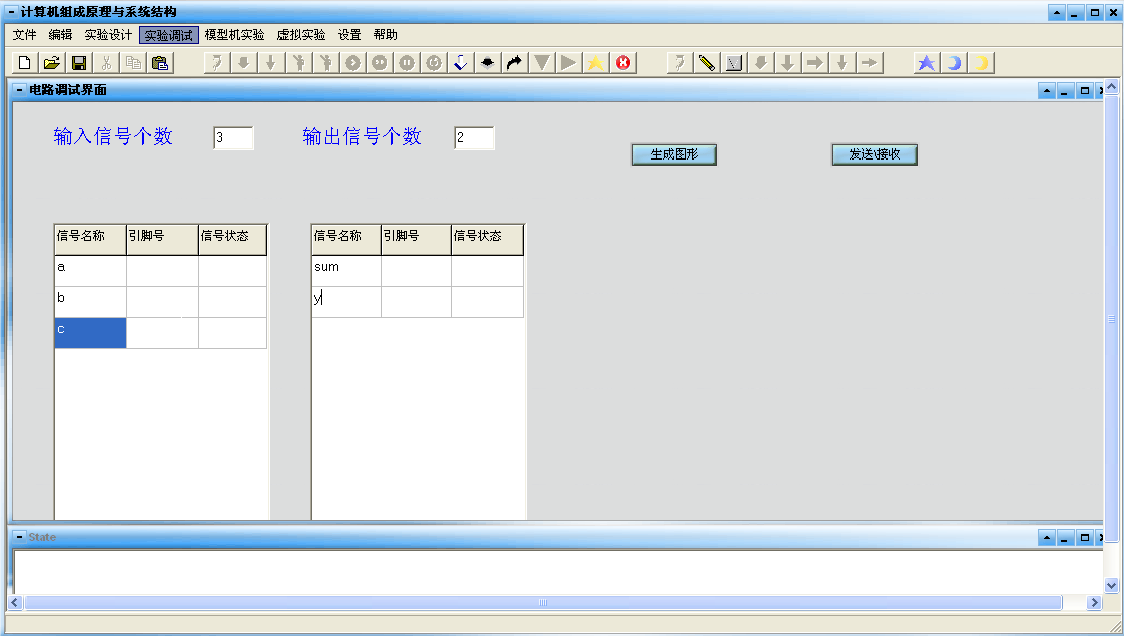


图2.8 确定信号个数和信号名称

点击图2.8中**“生成图形”**按钮即可产生所设计实验的图形框图模型，如图2.9所示。在图2.9的表格中填入实验设计电路的输入引脚号及其后的信号状态（输入信号赋值）后，点“**发送\接收”**按键即可完成对硬件电路输入信号的赋值，然后实验平台自动将该设计电路的输出信号值返回并在生成的框图模型中以引脚颜色红（代表1）、黑（代表0）显示出来，如图2.10所示。

**注意：“电路调试”**中输入和输出信号的引脚号仅限于使用FPGA器件EP2CIQ208C8的 **45、46、47、48、56、57、58、59、203、205**共10个引脚。

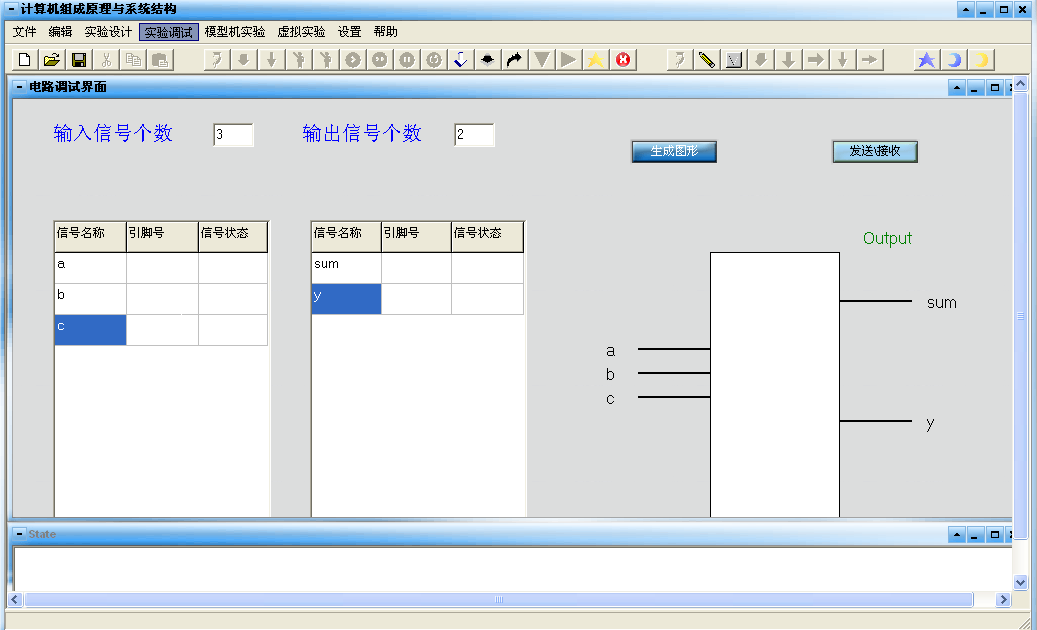


图2.9 确定信号个数和信号名称

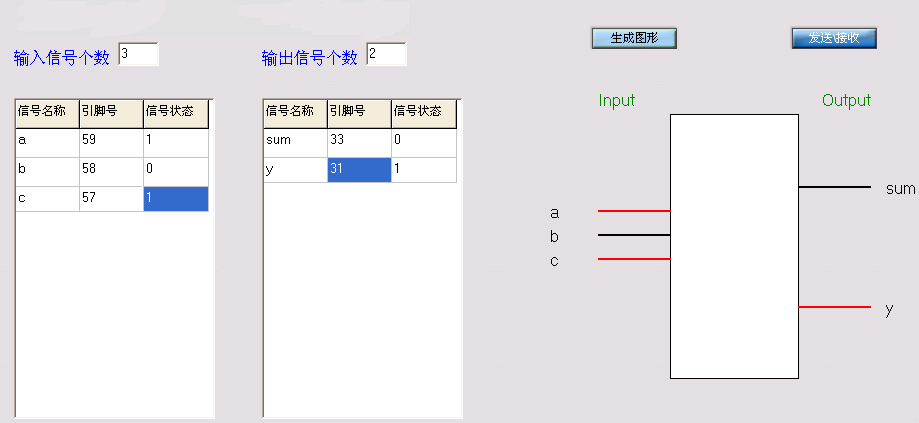


图2.10 电路调试程序界面

**（2）ROM写入、RAM写入**

**◆ 地址及数据输入：**点击菜单栏中的“**ROM写入🡪地址”**子菜单，将出现可以由电路调试界面切换到写ROM及RAM的界面，再继续点菜单栏的ROM、RAM的地址和数据以完成在相应输入框中输入起始地址及以此地址开始的一组数据，如图2.11和图2.12所示。

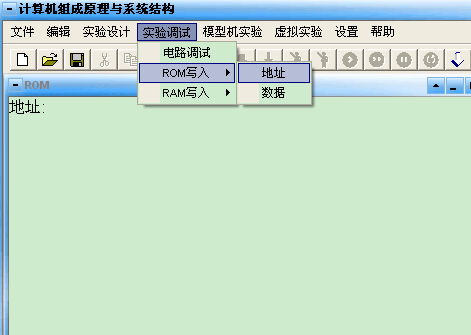


图 2.11 ROM地址写入界面

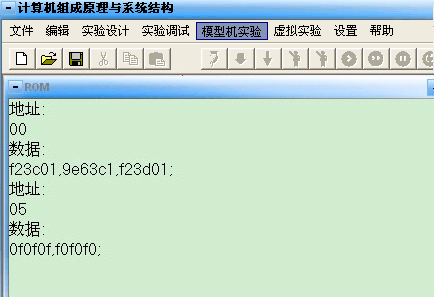


图 2.12 ROM数据写入界面

**◆ 数据写入ROM、RAM：**

数据输入结束后，在快捷键中选择按键，打开COM口（最右侧的红色按键为关闭串口），打开串口成功后再分别点击快捷键中的、按键，即可将数据下载到ROM、RAM中；快捷键为清空键，可将所有输入框和错误提示框信息清除。

**◆ ROM和RAM地址及数据写入规则：**输入规则请查看帮助

◇ ROM、RAM地址及RAM数据均以两位数据格式输入，如03,c2,范围00-ff；

◇ ROM数据以六位数据格式输入，如f23c01,范围000000-ffffff；

◇ ROM输入框每行输入8组数据回车换行，RAM输入框每行输入18组数据回车换行；

◇ 每组数据之间以 , 隔开，每组数据最后以 ; 结束；

**◆ 部分错误提示：**

◇ 地址与数据对应有误：查看是否是一个起始地址对应于一组数据；

◇ 地址输入有误：查看地址输入是否正确；

◇ 缺少分号：查看数据输入最后是否以 ：结束；

◇ 数据输入有误：查看每个数据的位数是否正确；

◇ 每行限输入8组（18组）数据：查看是否有某行数据输入超过规定个数。

◇地址冲突：查看后一组的起始地址是否已在前一组输入数据时对该地址已经赋值。）

**3. 模型机实验：**见第5章。

**注：快捷键功能介绍**

快捷键共分为四组，图2.13所示为右边三组的示意图（最左边一组与常用软件快捷键的功能相同，不再详述），图中第一组为**本地实验快捷键**，第二组为**远程实验快捷键**，第三组为**其它功能快捷键**。

各快捷键功能如下：

第一组，共16个快捷键，自左至右的功能分别为：

编译、下载指令、下载微指令、自动演示（连续）、自动演示（单步）、单步运行（组成原理）、连续运行（组成原理）、暂停、复位、写ROM、写RAM、清空、下载、单步运行（系统结构）、打开串口、关闭串口；

第二组，共8个快捷键，自左至右的功能分别为：

编译、选择配置文件、配置FPGA、下载机器码、下载微指令、单步运行（组成原理）、下载数据、单步运行（系统结构）；

第三组，共3个快捷键，自左至右的功能分别为：

硬件实验平台上的FPGA复位、RAM和ROM清零。

000.bmp

图2.13 快捷键示意图

### 2.3.2 远程实验

见远程实验说明书。

## 2.4简化版实验软件使用介绍

### 2.4.1 实验软件的安装

简化版实验软件为免安装软件，只要双击**“计算机组成原理与系统结构.exe”**即可进入实验软件平台，如图2.14所示界面，下面介绍实验软件的使用方法。



图2.14 实验程序开始界面

### 2.4.2 实验软件的使用

图2-14实验程序界面中，包括**文件**、**编辑、ROM/RAM读写、模型机实验**、**设置**和**帮助**八个菜单项和12个快捷键，菜单项的主要功能如下：

**1. ROM/RAM读写**

该菜单包括“**ROM/RAM读写、ROM/RAM读出、ROM/RAM写入**”三个子菜单，其中“**ROM/RAM读写”**子菜单的功能是在一个窗口中进行ROM/RAM的读和写操作、“**ROM/RAM读出”**子菜单的功能是仅对ROM/RAM进行读出操作、“**ROM/RAM写入”**子菜单的功能是仅对ROM/RAM进行写入操作。

**（1） ROM/RAM读写子菜单**

**◆** 点击“**ROM/RAM读写”**子菜单，将弹出图2.15所示窗口，最左边二列为RAM地址和其中的数据，中间四列为ROM1-ROM3的地址及数据。首次进入图2.15窗口时，系统会自动读出目前RAM和ROM中的数据。

**◆ RAM/ROM数据写入：**在图2.15中相应地址所对应的数据窗口中直接进行改写，所有需要写入的数据全部改写完毕后点击窗口右边的 **“写入RAM”** 按钮或**“写入ROM”** 按钮即可。

**◆ RAM/ROM数据读出：**直接点击**“读取RAM”** 按钮或**“读取ROM”** 按钮即可。

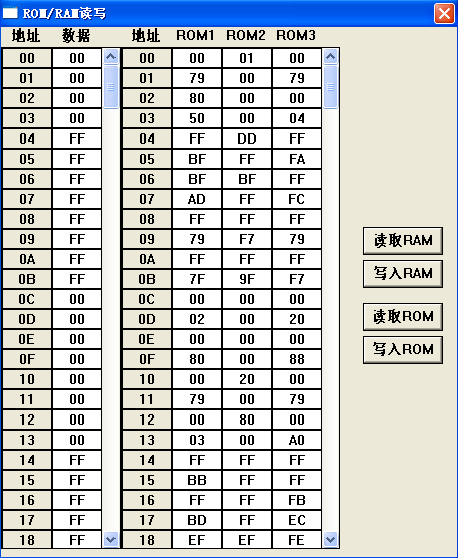
****

图2.15 ROM/RAM读写界面

**（2）ROM/RAM读出子菜单**

**◆** 单击程序菜单栏中的“ROM/RAM读出”子菜单；

**◆** 在弹出的对话框中分别输入要读出的单元的**【起始地址】**和**【回传单元数】**；

**◆** 单击**“读出RAM”**或 **“读出ROM” 按钮**，读出相应数据。

注意：① 起始地址：从该地址开始读取数据，为0到FF之间的十六进制数；

② 回传单元数：是从起始地址开始读出的单元个数，为0到FF之间的十六进制数；

③ 如果在回传单元数中输入0，则读取从起始地址开始一直到FF地址的数据。

**（3）ROM/RAM写入子菜单**

**◆** 单击程序菜单栏中的“ROM/RAM写入”子菜单；

**◆** 在弹出的对话框中按规定格式输入要写入数据的**地址**及要写入的**数据；**

**格式规定如下：**

地址: X1

数据: Y1 Y2 ... Yn

地址: X2

数据: Z1 Z2 ... Zn

...

**格式说明：**

数据Y1 写入 地址X1 、

数据Y2 写入 地址X1 + 1

...

数据Yn 写入 地址X1 + n - 1

数据Z1 写入 地址X2

数据Z2 写入 地址X2 + 1

...

数据Zn 写入 地址X2 + n -1

...

其中：

地址应为介于0到FF之间的十六进制数；

数据规定为：

如果是写入RAM，则每一数据为以两位十六进制表示的8位数据；

如果是写入ROM，则每一数据为一以六位十六进制表示的24位数据，其中前8位将写入ROM1，中间8位将写入ROM2，最后8位将写入ROM3。

**◆** 输入完毕后，单击**“写入RAM”**或 **“写入ROM” 按钮**进行写入。

**2. 模型机实验菜单**

单击**“模型机实验”**菜单弹出图2.16界面，图中上半部为一个典型模型计算机结构图，下半部的四个窗口分别用于汇编代码的输入、机器指令显示、微指令显示和编译结果及编译过程中的信息显示。

**◆** 汇编代码仅支持 MOVD MOVR ADD SUB INC IN OUT 七条指令；

**◆** 单击**“编译”**按钮，汇编代码对应的机器指令、微指令将在机器指令窗口、微指令窗口显示出来，同时在编译结果窗口显示编译是否成功；

**◆** 若编译成功，下载机器指令按钮将被激活，单击该按钮，编译生成的机器指令将被下载到下位机。

**3. 串口设置菜单**

单击**“串口设置”**菜单弹出一个设置界面，该界面中包括“串口号、校验位、停止位、数据位和波特率”五个参数的设置，一般情况下仅进行串口号的设置，其他参数需要和下位机配合进行设置，在此不要进行更改。

**4. 文件、编辑、设置和帮助菜单**

与一般应用软件的功能相同，不再详述。

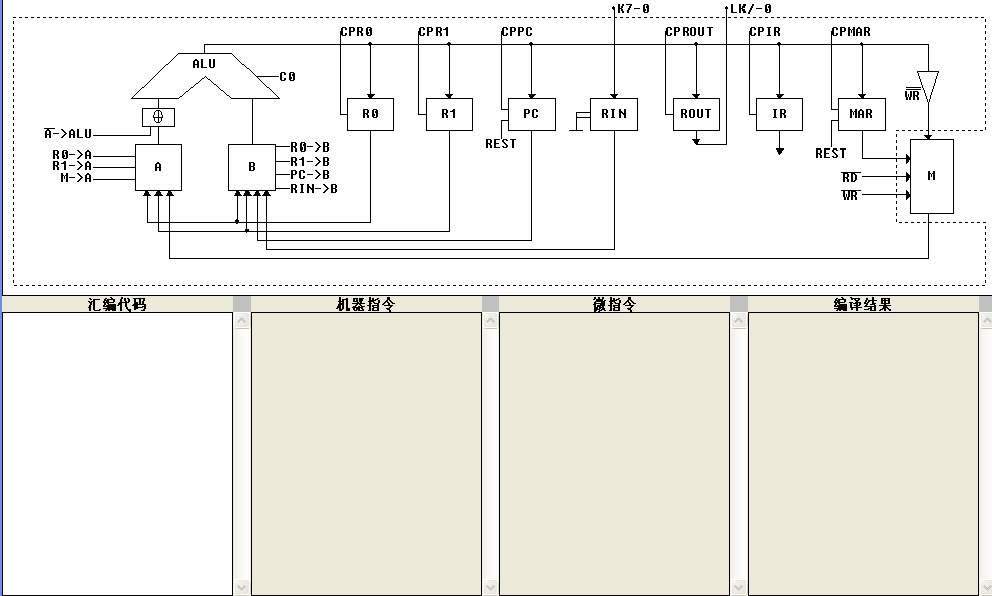
****

图2.16 模型机实验界面

**注意：**

* 本程序除串口设置外不支持十进制数，请使用十六进制数。
* 串口设置只支持十进制数，请在该处使用十进制数。

**◆** 单击**“下载微指令”**按钮将使微指令表下载到下位机，要查看微指令表请单击菜单栏的帮助->微指令表 ,这将自动打开jpg格式的微指令表，如需查看pdf以及html格式的微指令表请到程序目录下查看Form.pdf 以及Form.jpg文件。

**附：快捷键功能介绍**

该实验软件共有12个快捷键，分为四组，如图2.17所示。

5.bmp

图2.17 快捷键示意图

第一组：自左向右的功能为新建、打开和保存；

第二组：自左向右的功能为剪贴、复制、粘贴和清空；

第三组：自左向右的功能为编译、下载机器指令和下载微指令；

第四组：自左向右的功能为打开串口和关闭串口。

# 第3章 数字逻辑电路实验

## 3.1 实验用资源介绍

数字逻辑电路实验所用资源情况如图3.1所示。

● 输入开关：K23－16、K15－8和K7－0共3组；

● 发光管指示灯：LA15－2、LD15－0、LR15－0共3组；

● 时钟脉冲：连续时钟和单脉冲各1个；

● CPU复位：为**FPGA**的复位输入端。

● 配置接口：为**FPGA**的下载输入插口。

**FPGA**

**EP2C8Q208C8**

**单脉冲**

**指示灯**

**LA15-2**

**开关K23-16**

**开关K15-8**

**开关K7-0**

**下载**

**接口**

**CPU**

**复位**

**连续脉冲**

**指示灯**

**LD15-0**

**指示灯**

**LR15-0**

图3.1 **FPGA与**输入、输出资源的连接示意图

**说明： 输入开关、发光管显示和时钟脉冲所对应的管脚号见附录2。**

## 3.2 数字逻辑电路实验

**注意：**本地实验时，务必保证将DL1连接于PC机并口和实验平台的FPGA配置插座上、DL2连接于PC机串口和实验平台的RS232串口插座上。

### 实验1 数据选择器

1、实验目的

（1）了解数据选择器的工作原理；

（2）学习组合电路的基本设计方法；

（3）熟悉EDA工具软件的使用方法。

2、实验设备及器件

（1）操作系统为WINDOWS XP的计算机一台；

（2）数字逻辑与计算机组成原理实验系统一台；

（3）与门电路和或门电路若干。

3、实验内容及说明

数据选择器是指从多路数据输入中选择一路作为输出，本实验要求设计一个从三路输入中选择一路作为输出的三选一的数据选择器，其中每路输入和输出都为四位数据。图3.2所示为三选一数据选择器的框图，图中：A=a3a2a1a0，B=b3b2b1b0，C=c3c2c1c0，为三路四位数据输入端，Y=y3y2y1y0为一路四位数据输出端，CY、BY、AY为数据选择器的选择输入端。

AY

BY

Y

CY

B

C

A

数据选择器

图3.2 数据选择器框图

4、实验步骤

（1）原理图输入：根据图3.3电路，采用图形输入法在计算机上完成实验电路的原理图输入。

（2）管脚定义：根据图3.1硬件实验平台资源示意图和附录一 平台资源和FPGA引脚连接表完成原理图中输入、输出管脚的定义。

（3）原理图编译、适配和下载：在QuartusⅡ环境中选择**EP2C8Q208C8器件，**进行原理图的编译和适配，无误后完成下载。

（4）数据选择器的调试： 使用输入开关在数据选择器输入端预置任意数值，然后使AY、BY、CY分别有效（高电平有效，即开关向上），观察输出Y的值是否和相应的输入值相同。

（5）生成元件符号，以备以后使用。



图3.3 数据选择器电路原理图

### 实验2 全加器

1、实验目的

（1）学习组合电路的设计方法；

（2）了解全加器的构成和工作原理；

（3）熟悉EDA工具软件的使用方法。

2、实验设备及器件

（1）操作系统为WINDOWS XP的计算机一台；

（2）数字逻辑与计算机组成原理实验系统一台；

（3）二输入四与非门74LS00 和二输入四异或门74LS86。

3、实验内容及说明

本实验要求利用两输入与非门和异或门设计一个1位二进制全加器，其中a为被加数、b为加数，ci为低位来的进位，s为本位和，co为向高位的进位，图3.4为1位二进制全加器的框图，原理图如图3.5所示。

图3.4 1位二进制全加器框图

**b**

**ci**

**s**

**co**

**a**

**FADD**



图3.5 1位二进制全加器电路原理图

4、实验步骤

（1）原理图输入：根据图3.5电路，采用图形输入法在计算机上完成实验电路的原理图输入。

（2）管脚定义：根据图3.1硬件实验平台资源示意图和附录一 平台资源和FPGA引脚连接表完成原理图中输入、输出管脚的定义。

（3）原理图编译、适配和下载：在QuartusⅡ环境中选择**EP2C8Q208C8器件，**进行原理图的编译和适配，无误后完成下载。

（4）全加器的调试： 根据全加器的真值表，使用输入开关为全加器输入赋值，观察输出本位和S及向高位进位CO的值是否正确。

（5）生成元件符号。

5、选作内容

设计一个逻辑电路来判断四位BCD码中“1”的个数是否为奇数，是奇数输出结果为“1”，是偶数输出结果为“0”。

### 实验3 十进制数加法器

1、实验目的

（1）学习组合电路的设计方法；

（2）了解余三码的构成和十进制数加法器的构成和工作原理；

（3）熟悉EDA工具软件的使用方法。

2、实验设备及器件

（1）操作系统为WINDOWS XP的计算机一台；

（2）数字逻辑与计算机组成原理实验系统一台；

（3）4位二进制并行加法器74283和六非门7404。

3、实验内容及说明

本实验要求在掌握四位并行加法器74283使用方法和理解余3码运算法则的基础上，利用4位二进制并行加法器74283和六非门7404设计一个用余三码编码的1位十进制数加法器，并通过发光二极管显示加法结果。余三码编码的1位十进制数加法器原理图如图3.6所示，其中A4-A1和B4-B1为两个余三码编码表示的加数，CIN为低位来的进位，SUM4-SUM1为余三码编码表示的和数，COUT为向高位的进位。



图3.6 余三码编码的1位十进制数加法器原理图

4、实验步骤

（1）原理图输入：根据图3.6电路，采用图形输入法在计算机上完成实验电路的原理图输入。

（2）管脚定义：根据图3.1硬件实验平台资源示意图和附录一 平台资源和FPGA引脚连接表完成原理图中输入、输出管脚的定义。

（3）原理图编译、适配和下载：在QuartusⅡ环境中选择**EP2C8Q208C8器件，**进行原理图的编译和适配，无误后完成下载。

（4）加法器的调试： 根据余三码的运算规则，使用输入开关在余三码编码的1位十进制数加法器的输入端赋值，观察输出和数SUM及向高位进位COUT的值是否正确。

### 实验4 译码器

1、实验目的

（1）学习组合电路的设计方法；

（2）了解译码器的工作原理和构成；

（3）熟悉EDA工具软件的使用方法。

2、实验设备及器件

（1）操作系统为WINDOWS XP的计算机一台；

（2）数字逻辑与计算机组成原理实验系统一台；

（3）三输入与门和非门电路若干。

3、实验内容及说明

本实验要求完成一个3线－8线译码器的设计。其中i2-i0为译码器输入端，y7-y0为译码器输出端。图3.7为三线―八线译码器的框图，图8给出了三线―八线译码器的原理图。

i2 i1 i0

y7 y6 y5 y4 y3 y2 y1 y0

decoder

图3.7 3线―8线译码器框图

4、实验步骤

（1）原理图输入：根据图3.8电路，采用图形输入法完成实验电路的原理图输入。

（2）管脚定义：根据图3.1硬件实验平台资源示意图和附录一 平台资源和FPGA引脚连接表完成原理图中输入、输出管脚的定义。

将译码器的三个输入端分别定义在K2-0上。

将译码器的三个输出端分别定义在LD7-0上。

（3）原理图编译、适配和下载：在QuartusⅡ环境中选择**EP2C8Q208C8器件，**进行原理图的编译和适配，无误后完成下载。

（4）功能测试：改变K2-0的状态，译码器的输出则相应改变。

（5）生成元件符号。



图3.8 3线—8线译码器

### 实验5 八位寄存器

1、实验目的

（1）学习基本时序电路的设计方法；

（2）了解寄存器的工作原理和构成；

（3）熟悉EDA工具软件的使用方法。

2、实验设备及器件

（1）操作系统为WINDOWS XP的计算机一台；

（2）数字逻辑与计算机组成原理实验系统一台；

（3）基本D触发器8个。

3、实验内容及说明

本实验要求采用D触发器设计一个8位的寄存器，其中d7—d0、q7—q0分别为寄存器的输入和输出，cp为寄存器的时钟脉冲。

图3.9为8位寄存器的框图。

图3.10电路为8位寄存器的电路原理图。

cp

d7 d6 d5 d4 d3 d2 d1 d0

q7 q6 q5 q4 q3 q2 q1 q0

reg8

图3.9 8位寄存器的框图



图3.10 8位寄存器电路原理图

4、实验步骤

（1）原理图输入：根据图3.10电路，采用图形输入法在计算机上完成实验电路的原理图输入。

（2）管脚定义：根据图3.1硬件实验平台资源示意图和附录一 平台资源和FPGA引脚连接表完成原理图中输入、输出管脚的定义。

将寄存器的输出q7－q0分别锁定在LD7－0上。

将寄存器的输入d7－d0分别锁定在K7－0上。

将寄存器的输入脉冲cp锁定在单脉冲上。

（3）原理图编译、适配和下载：在QuartusⅡ环境中选择**EP2C8Q208C8器件，**进行原理图的编译和适配，无误后完成下载。

（4）功能测试：改变K7－0的状态，按动一次单脉冲键，LD7－0的显示将与K7－0相对应，若有错则重新调试。

（5）生成元件符号。

### 实验6 同步模4可逆计数器

1、实验目的

（1）学习同步时序电路的设计方法；

（2）了解可逆计数器的工作原理和设计实现；

（3）熟悉EDA工具软件的使用方法。

2、实验设备及器件

（1）操作系统为WINDOWS XP的计算机一台；

（2）数字逻辑与计算机组成原理实验系统一台；

（3）双D触发器 74LS74、二输入二与非门74LS00、三输入三与非门74LS10和二输入四异或门74LS86。

3、实验内容及说明

在掌握同步时序逻辑电路设计方法的基础上，要求采用D触发器、二输入与非门、三输入与非门和异或门设计一个可逆模4计数器，其框图如图3.11所示，其中CP为计数脉冲输入端，CON为可逆计数器的控制端（CON=1进行加计数，CON=0进行减计数），Q2Q1位计数输出端，Z为进位端。同步模4可逆计数器原理图如图3.12所示。

**CP**

**Z**

**Q1**

**Q2**

**CON**

**COUNTER4**

图3.11 同步模4可逆计数器框图



图3.12 同步模4可逆计数器原理图

4、实验步骤

（1）原理图输入：根据图3.12电路，采用图形输入法在计算机上完成实验电路的原理图输入。

（2）管脚定义：根据图3.1硬件实验平台资源示意图和附录一 平台资源和FPGA引脚连接表完成原理图中输入、输出管脚的定义。

将可逆计数控制端CON锁定在K0上。LD7－0上。

将计数脉冲输入端CP锁定在单脉冲上。

计数脉冲输出Q2Q1可锁定在LD1和LD0上，进位输出Z可锁定在LD2上。

（3）原理图编译、适配和下载：在QuartusⅡ环境中选择**EP2C8Q208C8器件，**进行原理图的编译和适配，无误后完成下载。

（4）功能测试：按动单脉冲键，LD1－0应该按00-01-10-11或11-10-01-00的规律显示，LD2显示进位输出，如有错则检查电路并按上述步骤重新进行。

5、选作内容

利用JK触发器设计一个可逆模4计数器。

### 实验7 异步模8加1计数器

1、实验目的

（1）学习异步时序电路的设计方法；

（2）了解异步计数器的工作原理和设计方法；

（3）熟悉EDA工具软件的使用方法。

2、实验设备及器件

（1）操作系统为WINDOWS XP的计算机一台；

（2）数字逻辑与计算机组成原理实验系统一台；

（3）D触发器和非门电路若干。

3、实验内容及说明

本实验要求设计一个异步模8加1计数器，其中CLK为计数脉冲输入，CLR为复位输入，q2-q0为计数器的输出。

图3.13为异步模8加1计数器的原理图。



图3.13 异步模8加1计数器原理图

4、实验步骤

（1）原理图输入：根据图3.13电路，采用图形输入法完成实验电路的原理图输入。

（2） 管脚定义：将原理图中的计数脉冲CLK定义在单脉冲键上；CLR定义在k0上；计数的输出端分别定义在LD2－0上。

（3）原理图编译、适配和下载：在QuartusⅡ环境中选择**EP2C8Q208C8器件，**进行原理图的编译和适配，无误后完成下载。

（4）功能测试：

按一次单脉冲键（132脚），计数器加1，由LED2-0显示计数值。

将计数脉冲定义在连续脉冲上（131脚），则计数器循环计数LED2-0循环显示。

调整连续时钟脉冲插座上短路块的位置（见图2.2）改变连续脉冲频率，则LED闪烁频率将随之改变。

（5）生成元件符号。

### 实验8 节拍发生器

1、实验目的

（1）了解节拍发生器的工作原理和设计过程；

（2）学习采用层次化进行数字电路设计的方法；

（3）熟悉EDA工具软件的使用方法。

2、实验设备及器件

（1）操作系统为WINDOWS XP的计算机一台；

（2）数字逻辑与计算机组成原理实验系统一台；

（3）前述实验中生成的译码器和模8计数器元件符号。

3、实验内容及说明

本实验采用层次化的方法进行设计，在顶层电路中直接调用实验7、实验4中完成的计数器元件符号和译码器元件符号作为底层电路元件使用，图3.14是该节拍发生器的顶层电路原理图。



图3.14 节拍发生器原理图

4、实验步骤

（1）原理图输入：根据图3.14电路，直接调用实验3、实验4中完成的计数器元件符号和译码器元件符号完成节拍发生器的顶层电路原理图设计。

（2）管脚定义：根据图3.1中的管脚连接示意图完成原理图中输入、输出的管脚定义。

将计数器的计数脉冲端cp接入连续脉冲上。

将计数器的复位端re接入k0上。

将译码器的输出分别定义在LD7－0上。

（3）原理图编译、适配和下载：在QuartusⅡ环境中选择**EP2C8Q208C8器件，**进行原理图的编译和适配，无误后完成下载。

（4）功能测试：若连接无误，则LD7－0将依次循环点亮。用跨接线改变计数脉冲，节拍的显示频率将被改变。

（5）生成元件符号。

## 3.3数字逻辑课程设计

### 课题1 交通灯信号控制器设计

**1.1目的**

1.了解交通灯控制器的工作原理及实现过程。

2. 学习数字系统设计中的自顶向下设计法及状态机的设计。

3. 加深利用EDA技术实现数字系统的体会。

RGY

RGY

SEN1

SEN1

方向2

方向

1

图3-15 交通情况简图

**1.2仪器与器材**

1．EDA开发软件 一套

2．微机 一台

3．实验开发系统 一台

4．其他器件与材料 若干

**1.3 课题说明**

设交叉路口的交通情况如图3.15所示。有两条通行道路，分别沿方向 l和方向2通行。每一方向都有红、黄、绿组成的交通信号灯。这些信号灯由相应的高有效信号( R1，G1，Y1，R2，G2，Y2)来驱动。此外，每个方向上都装有一个传感器，当相应方向有车辆要求通过时，传感器给出高电平有效的信号。

具体要求：

1. 当SENl和SEN2之一为l时，相应方向的绿灯应点亮。

2. 当SENl和SEN2同时为l或为0时，两个方向的交通灯应循环点亮。

3. 要求绿灯点亮时间为20s，黄灯点亮时间为5s。

4. 按倒计时的方法用数码管指示灯亮时间。

5. 在要求的某时间段内，两个方向仅要求黄灯闪动。

交通流向的四种情况和可能的亮灯情况如表3.l和表3.2所示。

表3-1 交通流向

|  |  |  |
| --- | --- | --- |
| SEN1 | SEN2 | 流 向 |
| 0 | 1 | 允许方向2通行 R1=G2=1 |
| 1 | 0 | 允许方向1通行 R2=G1=1 |
| 0 | 0 | 两个方向等时循环亮灯 |
| 1 | 1 | 两个方向等时循环亮灯 |

表3-2 可能亮灯情况

|  |  |  |
| --- | --- | --- |
| 情 况 | R1 G1 Y1 | R2 G2 Y2 |
| 1  2  3  4 | 1 0 0  1 0 0  0 1 0  0 0 1 | 0 1 0  0 0 1  1 0 0  1 0 0 |

**1.4课题设计及报告要求**

1．写出状态编码方案和状态机转移图。

2．画出顶层电路图及各模块的原理图（或HDL文件）。

3. 将设计结果下载到实验开发系统上进行验证。

4．写出测试结果及分析。

### 课题2 数字密码锁电路设计

**2.1目的**

1.了解数字密码锁的工作原理及实现过程。

2. 学习数字系统设计中的自顶向下设计法及状态机的设计。

3. 加深利用EDA技术实现数字系统的体会。

**2.2 实验仪器及器件**

1.EDA开发软件 一套

2.微机 一台

3.实验开发系统 一台

4.其他器件与材料 若干

**2.3 课题要求及说明**

设计一个简易的数字密码锁电路，该锁应在收到3位与规定码相符的十进制数码时打开，使相应指示灯点亮；若收到代码与规定不符或者开锁程序有误，表示错误的指示灯点亮。数字密码锁基本要求如下：

1.系统接通电源后，首先按动SETUP键后方投入运行。运行时标志开门的指示灯和警报灯、铃皆不工作，系统处于安锁状态。

2.开锁代码是3位十进制数，可按照用户的意愿调定。代码不足3位或超出3位时均不能开锁。

3.开锁程序由设计者确定，用户必须严格执行所规定的程序，方可开锁。

4.开锁代码和程序正确，表示数字锁打开的指示灯点亮。

5.允许用户在开锁过程中有1次错误(输入代码错误或开锁程序错误)，只要出错，表示错误的指示灯必定点亮。如果有两次错误，则报警器——喇叭鸣叫，以示情况异常。

6.开锁程序为：

(1) 按启动键(START)启动开锁程序，此时系统内部应处于初始状态。

(2) 依次键入3个十进制码。

(3) 按开门键(OPEN)准备开门。

若按上述程序执行且拔号正确，则开门继电器工作，绿灯亮；若密码输入错误或未按上述程序执行，则按动开门键(OPEN)后警报装置鸣叫(单频)，红灯亮。

(4) 开锁事务处理完毕后，应将门关上，按SETUP键，使系统重新进入安锁状态。(若在报警，按SETUP或START均不起作用，应另用一个内部的I-SETUP键才能使系统进入安锁状态)。

(5) 若按错号码，可在按OPEN键之前按START键重新启动开锁程序。

7.号码0～9、START、OPEN均用按键产生，并均有消抖和同步化电路。

数字密码锁框图如图3.16所示。

状 态 机

密 码 输 入 电 路

比 较 器

密码寄存及选择

en

0

1

2

3

4

5

6

7

8

9

right

i-setup

setup

start

open

clk

en

gree

red

jbao

kaimen

sel

图3.16 数字密码锁框图

**2.4 课题报告要求**

1.系统设计方案论述，画出顶层电路结构图及功能分割图。

2.各功能模块设计说明及原理图（或源程序）。

3.设计实现过程及引脚锁定情况。

4.实验结果记录（下载测试结果或仿真结果）、讨论及心得体会。

### 课题3 自动升降电梯控制器设计

**3.1目的**

1. 了解自动升降电梯控制器的工作原理及实现过程。

2. 学习数字系统设计中的自顶向下设计法及状态机的设计。

3. 加深利用EDA技术实现数字系统的体会。

**3.2 实验仪器及器件**

1.EDA开发软件 一套

2.微机 一台

3.实验开发系统 一台

4.其他器件与材料 若干

**3.3 课题要求及说明**

**1. 控制器功能要求**

设计一个4层楼的电梯控制器，该控制器可完成4层楼的载客服务，基本要求如下：

(1) 根据电梯的运行状态，按方向优先控制方案设计（在电梯运行过程中，电梯上升（或下降）途中，任何反方向下降（或上升）的外呼梯信号均不响应，但如果反向外呼梯信号前方无其它内、外呼梯信号时，则电梯响应该外呼梯信号）；

(2) 当有内呼梯或外呼梯信号到来时，电梯响应呼梯信号，到达该楼层时，电梯停止运行，电梯门打开，延时一定时间后自动关门；

(3) 电梯应具有最远反向外梯响应功能。例如：电梯在一楼，而同时有二层向下外呼梯，三层向下外呼梯，则电梯先去三楼响应三层向下外呼梯信号；

(4) 电梯未平层或运行时，开门按钮和关门按钮均不起作用，平层且电梯停止运行后，按开门按钮电梯门打开，按关门电梯门关闭；

(5) 电梯空闲时，停在任意一层；

(6) 具有超载报警功能，当超载时电梯不关门，直到超载信号清除。

**2. 主要信号说明**

输入信号：请求信号、楼层到达信号、楼层指示信号、超载和上下行指示信号等。

输出信号：开关门、电机驱动、显示及报警信号等。

**3.4 课题设计及报告要求**

1. 系统设计方案论述，标明设计中使用的信号名称、功能等，画出顶层电路结构图及功能分割图。

2. 各功能模块设计说明及原理图（或源程序）。

3. 设计实现过程及引脚锁定情况。

4. 实验结果记录（下载测试结果或仿真结果）、讨论及心得体会。

# 第4章 计算机组成原理实验

## 4.1 实验目的

通过组成原理实验应使学生掌握计算机的基本结构以及各部件的功能、设计方法、设计过程和调试过程，熟悉计算机的工作原理和工作过程。

## 4.2实验用资源介绍

实验所用资源如图4.1所示。

## 4.3实验内容

### 实验1 逻辑运算电路

1、实验内容及说明

本实验要求设计一个能实现1位逻辑乘ab、逻辑或a+b和半加(a⊕b)的逻辑运算电路。图4.2为实现上述逻辑功能的电路原理图，其中参与运算的两个1位二进制数为a和b，and、xor和or分别为与运算、异或运算和或运算控制输入端。

2、实验步骤

（1）原理图输入：根据图4.2所示电路，完成逻辑运算的电路原理图设计。

（2）管脚锁定：完成原理图中输入、输出的管脚锁定。

将a操作数锁定在K9上；将b操作数锁定在K8上；将输出e锁定在LD0上；将与运算控制输入and端锁定在K2上；将异或运算控制输入xor端锁定在K1上；将或运算控制输入or端锁定在K0上。

（3）原理图编译、适配和下载：在QuartusⅡ环境中选择**EP2C8Q208C8器件，**进行原理图的编译和适配，无误后完成下载。

（4）功能测试：利用输入开关及发光二极管LD测试逻辑运算部件的功能并记录测试结果。

（5）生成元件符号。

3、选做

利用一位逻辑运算的结果实现两个4位二进制数A（a3a2a1a0）和B(b3b2b1b0)的逻辑运算并生成元件符号。

**RAM**

**FPGA**

**EP2C8Q208C8**

WR

μRD

**L23~ 0**

CPμIR

**132**

**131**

**单时钟**

**指示灯**

**LA15-0**

**单片机**

**开关K23-16**

**开关K15-8**

**开关K7-0**

**配置**

**接口**

**CPU**

**复位**

**连续时钟**

**指示灯**

**LD15-0**

**指示灯**

**LR15-0**

**RS232**

**接口**

**ROM1~ 3**

**μIR23~ 0**

**μIR23~ 0**

**AD7 ~ 0**

**D’7~ 0**

RD

**A’7~0**

图4.1 计算机组成原理实验资源示意图



图4.2 逻辑运算电路原理图

### 实验2 补码加法器

  1、实验内容及说明

本实验要求采用一位全加器和异或门设计一个四位二进制补码加法器，操作数和运算结果均采用补码表示，该加法器要求可做补码加法，也可做补码减法运算。

在补码运祘中：

  （X)补 十（Y)补 ＝（X十Y)补

   （X)补 一（Y)补 ＝（X)补 十（（Y)补）补

    其中（（Y)补）补是将（Y)补连同符号位逐位取反末位加1。

图4.3是四位补码加法器的原理图，图中SUM3-SUM0表示四位和数，C4为向高位的进位，A3-A0为A操作数，B3-B0为B操作数，C0为低位来的进位， K为控制端，当K为高电平时，执行减法运算，K为低电平时，执行加法运算。

 2.实验步骤

（1）原理图输入：从元件库中选取相应器件实现四位补码加法器电路设计。

   （2）管脚锁定：将SUM3-SUM0、C4依次定义在LED指示灯LD5-LD0上，将K、C0、A3-A0、B3-B0依次定义在输入开关K9、K8、K7-K4、K3-K0上。

（3）原理图编译、适配和下载：在QuartusⅡ环境中选择**EP2C8Q208C8器件，**进行原理图的编译和适配，无误后完成下载。

（4）功能测试：利用输入开关 K9-K0改变K、C0和A、B操作数的值，看LED指示灯显示的结果是否正确并记录结果。

（5）生成元件符号。



图4.3 四位补码加法器原理图

### 实验3 移位器

1、实验内容及说明

本实验要求采用传送方式实现二进制数的移位电路。图4.4给出了可对四位二进制数实现左移1位（×2），右移1位（÷2）和直接传送功能的移位线路，这也是运算器的主要功能。

在LM（左移）的控制下可实现左移1位，空位补0。

在RM（右移）的控制下可实现右移1位，空位补0。

在DM（直送）的控制下可实现直接传送。

2、实验步骤

（1）用图形输入法完成图4.3逻辑电路输入。

（2）管脚锁定：将四位二进制数a3-a0定义在K3－0上；将4位输出q3-q0定义在LD3－0上；将LM定义在K8上，高电位有效；将DM定义在K9上，高电位有效；将RM定义在K10上，高电位有效，完毕后下载。

（3）设置K3－0为任意4位数，在LM、DM、RM的作用下分别观察LD3－0的显示，并分析其正确性。

（4）生成元件符号。



图4.4 移位器电路原理图

### 实验4 八位串/并进位运算器

1、实验內容及说明

本实验可通过使用二片74LS181运算器构成八位组间串行进位运算器，也可以借助先行进位发生器74LS182构成组间并行进位八位运算器。图4.5是由二片74LS181和74LS182构成的八位组间并行进位运算器电路图。

**F3 F2 F1 F0**

**F7 F6 F5 F4**

**PN**

**GN**

**B3N ~ B0N**

**C8**

**A3N ~ A0N**

**B7N ~ B4N**

**A7N ~ A4N**

**CN4 74LS181 CN**

**74LS181 CN**

**GN1 PN1 CX GN0 PN0**

**PN**

**74LS182 C1**

**GN**

**C0**

图4.5 八位组间并行进位运算器原理图

2、实验步骤

（1）原理图输入：从元件库中选取元件74LS181和74LS182并按照图4.5完成电路设计。

   （2）管脚锁定：将操作数A7N-A0N依次定义在K7-K0，B8N-B0N依次定义在K15-K8，C0定义在K16上，将C8定义在LR0上，F8-F0依次定义在LD7–LD0上。

（3）原理图编译、适配和下载：在QuartusⅡ环境中选择**EP2C8Q208C8器件，**进行原理图的编译和适配，无误后完成下载。

（4）功能测试：利用输入开关K16-K0和LED指示灯测试运算器的功能并记录测试结果。

### 实验5 四位补码运算器

**1、实验内容及说明**

本实验要求设计一个能够实现补码加法、减法、加1、左移、右移、直接传送等功能的四位补码运算器，图4.6为四位补码运算器的电路框图,图中，移位器具有左移、右移和直接传送功能。

实验时，图4.6中的元件可直接调用，其中R0、R1、R2四位寄存器可直接调用元件库中的4D寄存器74173（设计原理图时74173的G1N、G2N、MN、NN和CLR管脚接地），移位器可调用本章实验3中设计的移位器元件，四位加法器可调用本章实验2中设计的补码加法器元件。

D

B

A

C0

CPR1

CPR0

CPR2

4位加法器

R2

R0

R1

移位器

LM

DM

RM

图4.6 四位补码运算器电路框图

**2、实验步骤**

（1）原理图输入：调用寄存器、移位器、补码加法器等元件根据图4.6完成四位补码运算器电路设计、调试并生成元件符号。

（2）管脚锁定：将4位操作数A（a3a2a1a0）锁定在K7－4上、将4位操作数B(b3b2b1b0)锁定在K3－0上、将打入脉冲CPR0锁定在K8上、将打入脉冲CPR1锁定在K9上、将打入脉冲CPR2锁定在K10上、将LM锁定在K11上、将DM锁定在K12上、将RM锁定在K13上、将C0（低位的进位）锁定在K14上。

开关K由低电平拔向高电平再拔向低电平，相当于一个脉冲。

将存放结果的寄存器R2的输出端D(d3d2d1d0)分别锁定在LD3－0上。

（3）管脚锁定完毕后在QuartusⅡ中选择**EP2C8Q208C8器件，**进行原理图的编译和适配，无误后完成下载。

（4）功能测试：利用开关K与LED测试：加法、加1、左移、右移、直传等功能，并分析结果正确性。

### 实验6 控制器实验

**1、实验内容及说明**

目前控制器设计大都采用微程序设计方法，又称存储逻辑控制器。微程序控制器电路结构如图4.7所示。它由控制存储器ROM、微程序PC计数器和微指令寄存器IR构成。

图中实线部分均由课程设计平台提供，而虚线部分则由学生自行设计。

其中，微程序计数PC向控制存储器提供8位微地址，在控存读信号的作用下，读出一条长24位的微指令代码，并在打入命令CPIR的作用下，送入IR23-16、IR15-8、IR7-0。

L23-16、L15-8、L7-0用于显示微指令寄存器IR23-16、IR15-8、IR7-0的内容。

**A7–A0**

**L15-8**

**L23-16**

**L7-0**

**CPμIR**

μRD



FPGA CPPC

μPC

**μIR23 - 16** **μIR15 ~ 8**  **μIR 7 - 0**

**ROM3 ROM2 ROM1**

图4.7 微程序控制器框图

每当按一次脉冲键便产生一个负脉冲，该脉冲的作用是：

·作为读控存的命令。

·负脉冲当作CPIR将读出的微指令打入微指令寄存器IR。

·负脉冲的上升沿使PC＋1形成下一条微指令的地址。

·负脉冲反相后的上升沿作为寄存器打入脉冲。

微程序时序如图4.8。

P //

 //

图 4.8 微程序时序

**2、实验步骤**

（1）参考数字逻辑实验的异步模8计数器（实验7）设计过程，完成8位具有加1功能和清除功能的计数器设计并封装（作为微程序计数器μPC使用）。

（2）按图4.6完成微程序控制器的连线及引脚锁定。

\* 用单脉冲驱动μPC的计数脉冲CPPC、μRD、CPμIR。

\* 将μPC的8位输出锁定在A7－A0上。

\* 将μPC的复位端锁定在**CPU复位**引脚上。

\* 下载。

（3）写入微程序 (请参考本地实验调试中的ROM写入)

       从ROM的0号单元开始写入、写入的微程序代码随意确定。

（4）读出微指令

◆按CPU复位按键，清除微指令计数器。

         ◆按单脉冲键

         读出0号单元中的微指令代码送IR23-IR0同时显示在L23-L0上；

         将微指令计数器PC十1为读下条微指令做准备。

        ◆连续按单脉冲键

微指令连续从ROM中读出并显示在L23-L0上。

               注意：在读出过程中要和原先写入的代码比对看是否正确

### 实验7 CPU综合实验

**1、实验内容及说明**

CPU综合实验电路包括运算器电路和控制器电路。图4.9给出了CPU综合实验结构框图。图中的虚线框外电路由实验箱提供，虚线框内电路由学生自行设计，其中微程序控制器是由μPC、ROM3、ROM2、ROM1和μIR构成。

运算器由三个寄存器R0、R1、R2、移位器、加法器等构成，并组装在一起构成ALU算术逻辑运算部件，参照图4.6所示。

**2、实验步骤**

(1) 调用ALU模块、μPC模块及门电路按图4.9完成连线。

(2) 管脚定义：ALU的输入数据a3-a0依次锁定在μIR23-μIR420上，CPR0、CPR1、CPR2依次锁定在μIR7-μIR5上，LM、DM、RM、C0依次锁定在μIR4-μIR1上，P锁定在单脉冲按键上。Q3-Q0依次锁定在LD3-LD0上。

CPU

复位

P

CPμPC

CPμIR μRD

a3 a2 a1 a0 μIR CPR0 CPR1 CPR2 LM DM RM C0

ROM2 ROM1

A7 A6 A5 A4 A3 A2 A1 A0

μPC

&

&

&

Q3 Q2 Q1 Q0

ALU

a3 …… a0 b3……b0 CPR0 CPR1 CPR2 LM DM RM C0

μIR23- 20 μIR7 μIR6 μIR1 μIR0

图4.9 CPU综合实验结构框图

(3) 适配、下载

(4) 编制微程序

微指令可确定如下格式：

CPR0 CPR1 CPR2

a3a2a1a0 CPR0 CPR1 CPR2 LM DM RM C0

&

&

&

P

μIR23—μIR20 μIR7 μIR6 μIR5 μIR4 ……………… μIR0

将微指令格式分为两部分：前面部分μIR23～μIR20可设置数据，后面部分μIR7～μIR0可确定微命令，例：需要CPR0脉冲，该位为1，否则为0；备用位填0。

例题：编写一个0110＋1000的微程序。

寄存器分配：0110送R0、1000送R1、结果送R2。

操作步骤 微指令 说明

0 1 1 0→R0； 60 00 80H 存入控制存储器ROM的0单元。

↓

1 0 0 0→R1； 80 00 40H 存入控制存储器ROM的1单元。

↓

R0＋R1→R2； 00 00 08H 存入控制存储器ROM的2单元。

(5) 写入微程序。

将三条微指令依次写入ROM的0、1、2单元，读出并查看是否有错，若有错改之。

(6) 功能检查

按CPU复位键清μPC，使之指向控制存储器的0号单元。

每按一次单脉冲键，便执行一条微指令。

按第3次单脉冲键，微程序执行完毕。

运算结果应存放在R2中，并用LD3－0指示。

同学可编制多种多样的微程序来完成各种操作。

例 2×（R0＋R1）⇒ R2；

1/2（R0＋R1）⇒ R2；

……

# 第5章 计算机系统结构实验

众所周知，计算机整机概念是了解、学习计算机知识最重要的一个概念。要建立一个清晰的整机概念，就必须搞懂一条指令执行的全过程，在这一过程中，要完成哪些操作，需要提哪些控制指令，数据通路是如何确定的，才使数据沿着指定的数据通路从一个部件流向另一个部件，这些问题的解决对建立一个清晰的整机概念至关重要。而硬件实验平台提供了一个完整的模型机实验，它较好地展示了上述各个问题，学生通过该模型机实验，能迅速地清楚地建立起整机概念，对学习计算机原理有极大的帮助。下面以二个实例介绍模型机的基本结构及实现。

## 5.1模型计算机-Ⅰ系统结构设计

### 5.1.1 实验目的

在对数据选择器、移位器、寄存器、运算器、存储器及微程序控制器等计算机基本部件了解的基础上，完成一个简单计算机的设计，通过该设计可使学生建立清晰的整机概念和设计方法。

### 5.1.2 实验说明

由于计算机涉及的部件较多、结构原理比较复杂，对于一个初学者常会感到无从下手，下面给出计算机的设计与调试的一般步骤，以帮助同学们顺利完成设计。

1、拟定指令系统；

2、确定总体结构；

3、逻辑设计；

4、确定控制方式；

5、编制指令流程；

6、编制微程序；

7、调试。

注意：指令系统条数不要太多，（1）和（2）两步需要反复考虑才能确定。

### 5.1.3 实验要求

1、画出计算机的原理图；

2、写出调试方案；

3、写出测试方案；

4、评价设计的优缺点。

### 5.1.4模型计算机-Ⅰ设计简介

**1、模型机指令系统**

模型机指令系统比较简单，设置双操作数指令和单操作数指令，如表5.1所示。

表5.1 指令系统

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 双操作数指令 | | 单操作数指令 | | | |
| 加法指令 | ADD | 左移指令 | ROL | 输入指令 | IN |
| 减法指令 | SUB | 右移指令 | ROR | 输出指令 | OUT |
| 送数指令 | MOV | 加1指令 | INC | 转移指令 | JMP |

1. **指令格式**

采用双字节指令，字长16位

寄存器设置：

R0

R1

C：存放从内存取出的源操数

D：存放从内存取出的目的数

操作码

源操作数

目的操作数

寄存器号

寻址方式

寻址方式

寄存器号

通用寄存器

寻址方式：寄存器寻址；操作数在指定的寄存器中

立即数 ；操作数在指令的下一个单元

寄存器间址；操作数地址在寄存器中。

**3、模型机总体结构**

模型机总体结构如图5.1所示。

**（1）总体结构基本设置**

模型机基本字长8位，因此通用寄存器。R0，R1、暂存器C、D，输入寄存器Ri，输出寄存器RO，程序计数器PC，均为8位。指令寄存器IR因采用双字节所以为16位。地址寄存器8位，存储器M容量为256X8，移位器、ALU为8位，A、B选择器均为8位。

**（2）控制方式**

模型机控制器的控制方式采用微程序设计，统称存储逻辑控制器，亦称微程序控制器。主要由微指令寄存器、控制存储器构成，用它来存放微程序，微程序中的任何一条微指令读出至微指令寄存器都产生相应的控制指令，如DM、PC A等，这些控制指令就能勾通一条数据通路，数据信息就从起点传送到终点。

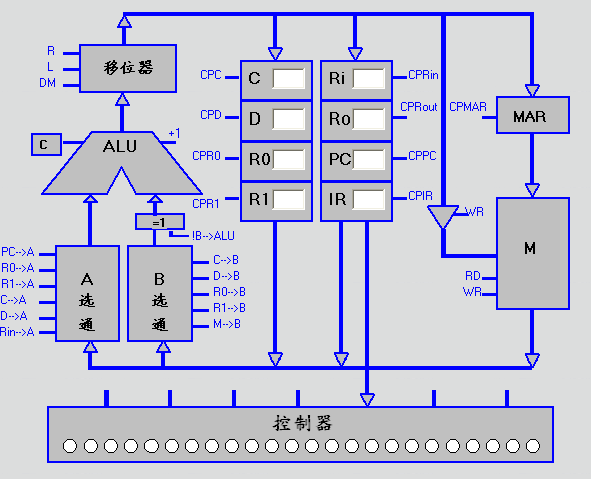


图5.1 组成原理模型机结构图

**（3）采用总线结构**

如图5.1所示：

A 各寄存器及存储器接收移位器经总线送来的数据。

B各寄存器及存储器经总线将数据送入运算器的选择器。

C信息在控制命令的控制下，经过选择器，送入ALU加工、处理，送入移位器。

**4、指令流程**

指令执行流程亦称指令执行过程，大都由四个阶段构成，即取指令，取源操作，取目的操作数，执行运算。但不同的指令不同的寻址方式，执令流程也不尽相同，需要的数据通路也不一样，因此需要的数据通路也不一样，当然控制命令也不一样，下面以ADDR0，R1为例的执行流程。

从以上流程可看出，因为源操作数、目的操作数均在寄存器，不需要访问内存，因此流程就非常简单，上面每一个流程需要一条数据通路，即需要一条微指令的控制命令勾通。

M ALU IR1

PC+1 PC

MAR

M ALU IR2

PC+1 PC

R0+R1  R1

PC MAR

取指令的第一个字节送指令寄存器

为取下一个指令字节做准备

取指令的第二个字了送指令寄存器

为取下条指令做准备

执行运算，并将结果送R1

将下条指令的地址送地址寄存器

**5、指令格式**

（1）模型机微指令格式

该模型机规定字长为24位，其格式如下：

A

B

Y

F

J

CP1

CP2

RD

WR

其中，A和B均为选择器字段，3位字长；Y为移位字段2位字长；F为反相控制，1位字长；J 为 进位控制，1位字长； CP1、CP2均为打入脉冲，3位字长； RD为读命令，1位字长；WR为写命令，1位字长。

**（2）字段定义**

A（3位） B（3位） Y（2位） F（1位）

000 R0 A 000 R0 B 00 R 1 B ALU

001 R1 A 001 R1 B 01 L

010 C A 010 C B 10 DM 0 B ALU

011 D A 011 D B

100 PC A 100 M B

101 Rin A

J （1位） CP1 (3位) CP2 (3位) RD（1位） WR （1位）

1 +1 000 CPR0 000 CPRin 0 读 0 写

0 001 CPR1 001 CPRout

010 CPC 010 CPPC

011 CPD 011 CPIR1

100 CPMAR 100 CPIR2

## 5.2基本模型机-Ⅱ系统结构设计

### 5.2.1 实验目的

在对数据选择器、移位器、寄存器、运算器、存储器及微程序控制器等计算机基本部件了解的基础上，完成一个简单计算机的设计，通过该设计可使学生建立清晰的整机概念和设计方法。

### 5.2.2 实验说明

由于计算机涉及的部件较多、结构原理比较复杂，对于一个初学者常会感到无从下手，下面给出计算机的设计与调试的一般步骤，以帮助同学们顺利完成设计。

1、拟定指令系统；

2、确定总体结构；

3、逻辑设计；

4、确定控制方式；

5、编制指令流程；

6、编制微程序；

7、调试。

注意：指令系统条数不要太多，（1）和（2）两步需要反复考虑才能确定。

### 5.2.3 实验要求

1、画出计算机的原理图；

2、写出调试方案；

3、写出测试方案；

4、评价设计的优缺点

### 5.2.4模型计算机-Ⅱ设计简介

**1、指令系统**

模型机指令系统比较简单，设置双操作数指令和单操作数指令，如表5.2所示。

表5.2 指令系统

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 双操作数指令 | | | 单操作数指令 | | |
| 指令 | 汇编符号 | 操作码 | 指令 | 汇编符号 | 操作码 |
| 数据传送 | MOVD | 0001 | 加1 | INC | 0101 |
| 寄存器间传送 | MOVR | 0010 | 输入 | IN | 0110 |
| 加法 | ADD |  | 输出 | OUT | 0111 |
| 减法 | SUB |  | 停机 | END | 1000 |

1. **指令格式**

指令基本字长8位，格式如下：

指令操作码字段 寄存器号字段 寄存器号字段

源操作数

目的操作数

其中：指令操作码字段占4位，寄存器号字段各占2位；00 选用R0，01 选用R1 ；单操作数指令中源操作数字段不用。

寻址方式：寄存器寻址，操作数地址在寄存器中；

立即数；操作数在指令的下一个单元，该方式只用于MOVD指令。

寻址方式隐含在指令操作码中。

**3、模型机总体结构**

模型机总体结构如图5.2所示。

**（1）总体结构基本设置**

模型机基本字长8位，因此通用寄存器R0，R1，输入专用寄存器Rin，输出专用寄存器Rout，指令寄存器IR，程序计数器PC，地址寄存器MAR均为8位。存储器M容量为256X8， ALU为8位，A、B选择器均为8位。

**（2）控制方式**

模型机控制器的控制方式采用微程序设计，统称存储逻辑控制器，亦称微程序控制器。主要由微程序控制器μPC、微指令寄存器μIR和控制存储器构成。微程序中的任何一条微指令读出至微指令寄存器就会产生相应的控制指令，如M A 、PC B等，这些控制指令就能勾通一条数据通路，数据信息就从起点传送到终点。

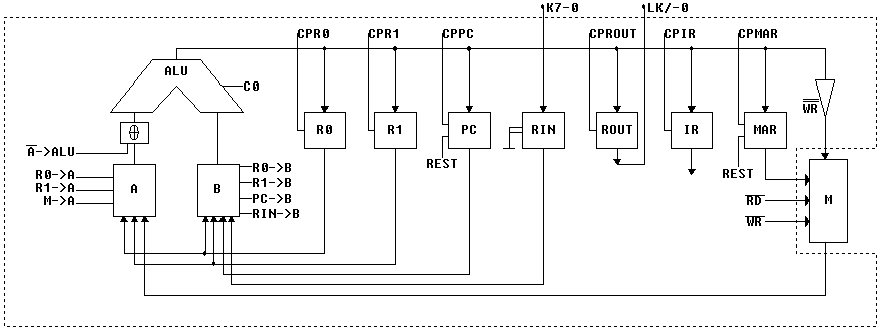


图5.2 模型机-Ⅱ结构图

**（3）采用总线结构**

如图5.2所示：

A 各寄存器及存储器接收总线送来的数据。

B各寄存器及存储器将数据送入运算器的选择器。

C信息在控制命令的控制下，经过选择器，送入ALU加工、处理后送入总线。

**4、指令流程**

指令执行流程亦称指令执行过程，大都由四个阶段构成，即取指令，取源操作，取目的操作数，执行运算。但不同的指令不同的寻址方式，执令流程也不尽相同，需要的数据通路也不一样，因此需要的数据通路也不一样，当然控制命令也不一样。

指令流程如图5.3所示。

流程中Ri是源操作数，Rj是目的操作数。

图5.3 指令流程图

MOVD MOVR ADD SUB INC IN OUT END

PC ALU Rj

M ALU

PC + 1 PC

PC μAR Ri Rj Ri+Rj Rj Rj-Ri Rj Rj+1 Rj Rin Rj Rj Rout

PC μAR

PC+1 PC

M ALU IR

从以上流程可以看出，因为源操作数、目的操作数均在寄存器，不需要访问内存，因此流程就非常简单，上面一个流程需要一条数据通路，即需要一条微指令的控制命令沟通。

**5、微指令格式与控制逻辑**

（1）模型机微指令格式

该模型机规定微指令字长为24位，其格式如图5.4所示，微指令字长共24位，共设8个有效字段，其它空位均可做备用位。

μIR23-2019 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

A选择门 B选择门 ALU输出分配 读写控制 顺序控制

图5.4 模型机微指令格式

图5.4中：

 A选择门字段共3位，用于产生A选择门所需要的控制命令。

B选择门字段共3位，用于产生B选择门所需要的控制命令。

ALU输出分配字段共5位，用于产生写入寄存器所需要的控制脉冲。

仃机状态字段共1位，用于确定是否仃机。

进位c字段共1位，用于确定进位C是1还是0。

RAM读写控制字段共2位，用于控制RAM是读还是写操作。

微程序顺序控制字段共2位，实現微程序的顺序控制、按指令操作码转移和按给定地址转移控制。

微指令中相应位为1该命令有效，为0该命令无效。编写微程序时，备用位均填0。

2 控制逻辑电路

（1）图5.4中μIR23----μIR0为微指令寄存器，微指令从控制存储器ROM读出至微指令寄存器后，通过直接或简单译码产生所需要的控制命令。

（2）根据指令寄存器IR的操作数字段和微指令寄存器μIR相关位的状态，可产生通用寄存器R0.R1写入信息所需要的控制命令,如图5.5所示。

IR7-- 4 IR3-- 0

X X X X

R0 🡪 A R0 🡪 B

R1 🡪 A R1 🡪 B

**00 01 10 11**

2-4译码器

**00 01 10 11**

2-4译码器

CPRj

**CPR0 CPR1**

图5.5 R0、R1写入信息控制命令产生电路

（3） 图5.6是微程序后继地址形成电路。主要由微程序计数器μPC等电路构成。该电路根据微指令寄存器μIR的顺序字段的不同状态产生不同的后继地址。

当J0为1时，微程序计时器加1形成后继地址。

当J1为1时，高4位由指令操作码确定，低4位补0形成后继地址。

当J2为1时，接收微指令寄存器μIR23---16给定的地址形成后继地址。

μPC7-- 4 μPC3-- 0

**AD7—AD0**

**J0**

**J1**

**J2**

IR7 IR 4

**......**

**......**

μIR23 **......** μIR 20 μIR19 **......** μIR 16

**CLK**

**RST**

图5.5 微程序后继地址形成电路

# 附录1 JYS实验系统硬件平台资源和FPGA引脚连接表

硬件实验平台上可提供给用户使用的资源主要包括输入开关、输出发光二极管指示灯、单脉冲和连续脉冲等；其中输入开关共有24个（K23 ~ K0），发光二极管指示灯共72只，其中LR15 ~ LR0、LD15 ~ LD0、LA15 ~ LA2共46只可由用户随意使用（**LA1、LA0由系统占用，用户不能使用**），L23 ~ L0用于显示微指令寄存器内容，即当前执行的微指令。

**1. 开关K23 ~ K0与FPGA芯片的引脚连接**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **开关号** | **K23** | **K22** | **K21** | **K20** | **K19** | **K18** | **K17** | **K16** | **K15** | **K14** | **K13** | **K12** |
| **引脚号** | **107** | **130** | **129** | **208** | **28** | **27** | **24** | **23** | **76** | **75** | **74** | **72** |
| **开关号** | **K11** | **K10** | **K9** | **K8** | **K7** | **K6** | **K5** | **K4** | **K3** | **K2** | **K1** | **K0** |
| **引脚号** | **70** | **69** | **68** | **67** | **88** | **87** | **86** | **84** | **82** | **81** | **80** | **77** |

**2. 发光二极管指示灯与FPGA芯片的引脚连接**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **发光管** | **LA15** | **LA14** | **LA13** | **LA12** | **LA11** | **LA10** | **LA9** | **LA8** | **LA7** | **LA6** | **LA5** | **LA4** | **LA3** | **LA2** | **LA1** | **LA0** |
| **引脚号** | **33** | **31** | **30** | **15** | **14** | **13** | **12** | **11** | **10** | **8** | **6** | **5** | **4** | **3** | **系统占用** | |
| **发光管** | **LD15** | **LD14** | **LD13** | **LD12** | **LD11** | **LD10** | **LD9** | **LD8** | **LD7** | **LD6** | **LD5** | **LD4** | **LD3** | **LD2** | **LD1** | **LD0** |
| **引脚号** | **141** | **139** | **138** | **137** | **135** | **134** | **133** | **128** | **44** | **43** | **41** | **40** | **39** | **37** | **35** | **34** |
| **发光管** | **LR15** | **LR14** | **LR13** | **LR12** | **LR11** | **LR10** | **LR9** | **LR8** | **LR7** | **LR6** | **LR5** | **LR4** | **LR3** | **LR2** | **LR1** | **LR0** |
| **引脚号** | **165** | **164** | **163** | **162** | **161** | **160** | **152** | **151** | **150** | **149** | **147** | **146** | **145** | **144** | **143** | **142** |

**3. RAM地址线与FPGA芯片的引脚连接**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **RAM地址线** | **A’7** | **A’6** | **A’5** | **A’4** | **A’3** | **A’2** | **A’1** | **A’0** |
| **引脚号** | **179** | **176** | **175** | **173** | **171** | **170** | **169** | **168** |

**4. RAM数据线（双向）与FPGA芯片的引脚连接**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **RAM数据线** | **D’7** | **D’6** | **D’5** | **D’4** | **D’3** | **D’2** | **D’1** | **D’0** |
| **引脚号** | **201** | **200** | **199** | **198** | **197** | **195** | **193** | **192** |

**5. ROM地址线与FPGA芯片的引脚连接**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ROM地址线** | **AD7** | **AD6** | **AD5** | **AD4** | **AD3** | **AD2** | **AD1** | **AD0** |
| **引脚号** | **191** | **189** | **188** | **187** | **185** | **182** | **181** | **180** |

**6. 微指令数据输入：**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **uIR23** | **uIR22** | **uIR21** | **uIR20** | **uIR19** | **uIR18** | **uIR17** | **uIR16** | **uIR15** | **uIR14** | **uIR13** | **uIR12** |
| **127** | **118** | **117** | **116** | **115** | **114** | **113** | **112** | **110** | **108** | **106** | **105** |
| **uIR11** | **uIR10** | **uIR9** | **uIR8** | **uIR7** | **uIR6** | **uIR5** | **uIR4** | **uIR3** | **uIR2** | **uIR1** | **uIR0** |
| **104** | **103** | **102** | **101** | **99** | **97** | **96** | **95** | **94** | **92** | **90** | **89** |

**7. 其他信号与FPGA芯片的引脚连接**

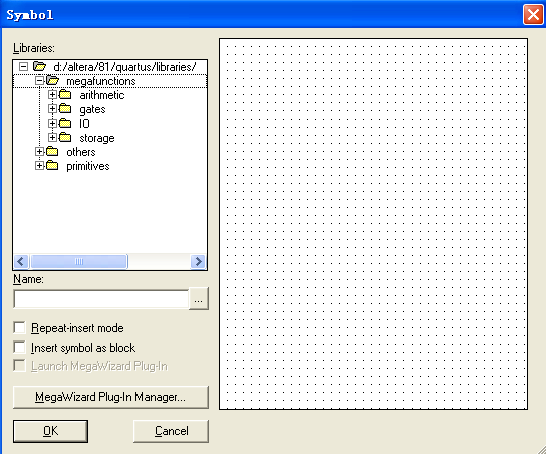
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **信号名** | μRD | CPμIR | rd | wr | **连续脉冲** | **单脉冲** | **CPU复位** |
| **引脚号** | **60** | **61** | **63** | **64** | **131** | **132** | **206** |

表中：μRD为ROM读信号、CPμIR为微指令寄存器时钟信号、rd为RAM读信号、wr为RAM写信号、连续脉冲可在2Hz、4Hz 、8Hz 、16Hz、32Hz 、64Hz 、128Hz、256Hz 、512Hz 、1024Hz 和2048Hz中选择，占空比为1:1、单脉冲为宽度20ms的负脉冲（无抖动，按单脉冲键）。

# 附录2 QuartusⅡ库及库元件说明

Quartus Ⅱ软件为实现不同的逻辑功能提供了大量的基本单元符号和宏功能模块，设计者可以在原理图编辑器中直接调用，非常方便设计者使用原理图输入法进行设计。Quartus Ⅱ软件为设计者提供了三个库：宏功能模块(megafunctions)库、其他 模块(others)库和基本单元符号(primitives)库。

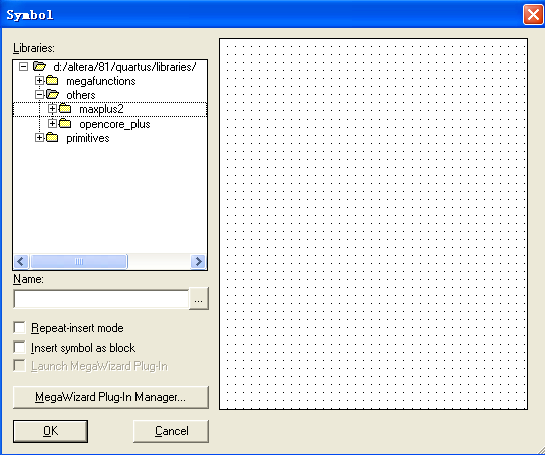
宏功能模块 (megafunctions)库如图附2.1所示，宏功能模块式参数化的模块，模块的各个参数由设计者为满足设计要求自行定制，只要修改模块参数，就可以得到满足需要的特定模块。宏功能模块包含算术运算类模块、门单元类模块、输入/输出类模块和存储器类模块。



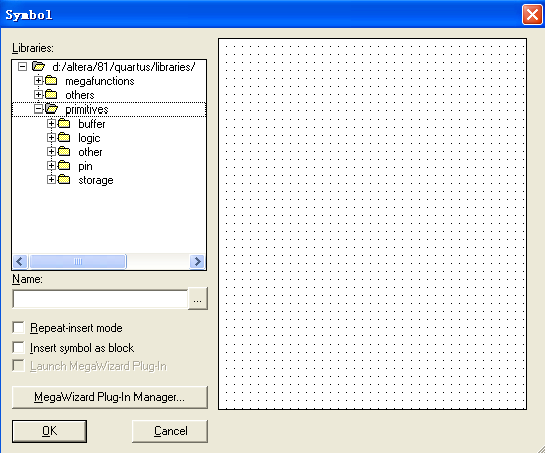
图附2.1 宏功能函数(megafunctions)库

其他模块(others)库是一个与maxplus2兼容的模块库，如图附2.2所示。包含74系列大部分器件符号和各种组合电路模块符号，在模块编辑器中可以查看内部的电路结构，例如，输入二选一数据选择器符号21mux，在模块编辑器中双击该符号，就会出现21mux的内部电路结构和说明。

基本单元符号(primitives)库如图附2.3所示，该库中包含缓冲器(buffer)、基本逻辑符号(logic)、引脚符号（pin）、触发器（storage）和其它功能模块（other）,其中最重要的是pin目录下的输入引脚和输出引脚，这两个引脚是任何原理图文件都要用到的引脚符号。



图附2.2 其他 (others)库



图附2.3 基本单元符号(primitives)库

此外，设计过程中，设计者还可以为设计项目创建专门为自己使用的模块符号，以方便在原理图编辑器中多次调用。