

数字逻辑电路与计算机组成原理

实验指导书

2016. 1

山东大学

目录

第 1 章 QuartusII 软件使用介绍	- 1 -
1.1 Quartus II 图形输入法	- 1 -
1.2 仿真验证.....	- 12 -
1.3 Quartus II 文本输入法使用介绍	- 16 -
1.4 层次化设计方法.....	- 18 -
第 2 章 实验系统介绍.....	- 20 -
2.1 实验系统组成.....	- 20 -
2.2 实验硬件平台	- 21 -
2.2.1 实验硬件平台结构介绍	- 21 -
2.2.2 实验硬件平台的资源简介	- 21 -
2.2.3 实验硬件平台的连接	- 24 -
2.3 实验软件介绍	- 24 -
2.3.1 实验软件的安装.....	- 24 -
2.3.2 本地实验.....	- 25 -
2.3.2 远程实验.....	- 29 -
第 3 章 数字逻辑电路实验.....	- 30 -
3.1 实验用资源介绍	- 30 -
3.2 数字逻辑电路实验	- 30 -
实验 1 数据选择器	- 31 -
实验 2 全加器	- 32 -
实验 3 十进制数加法器	- 34 -
实验 4 译码器	- 35 -
实验 5 八位寄存器	- 36 -
实验 6 同步模 4 可逆计数器	- 38 -
实验 7 异步模 8 加 1 计数器	- 39 -
实验 8 节拍发生器	- 40 -
3.3 数字逻辑课程设计	- 42 -
课题 1 交通灯信号控制器设计	- 42 -
课题 2 数字密码锁电路设计	- 43 -
课题 3 自动升降电梯控制器设计	- 44 -

第 4 章 计算机组成原理实验	- 46 -
4.1 实验目的.....	- 46 -
4.2 实验用资源介绍.....	- 46 -
4.3 实验内容.....	- 46 -
实验 1 逻辑运算电路.....	- 46 -
实验 2 补码加法器.....	- 48 -
实验 3 移位器.....	- 49 -
实验 4 八位串/并进位运算器.....	- 50 -
实验 5 四位补码运算器.....	- 51 -
实验 6 控制器实验.....	- 53 -
实验 7 CPU 综合实验.....	- 54 -
实验 8 模型机演示.....	- 56 -
第 5 章 模型机课程设计	- 58 -
5.1 模型机系统结构.....	- 58 -
5.1.1 模型机指令系统.....	- 58 -
5.1.2 模型机总体结构.....	- 59 -
5.1.3 指令流程.....	- 60 -
5.1.4 微指令格式.....	- 60 -
5.2 模型机数据通路设计.....	- 61 -
5.3 简单计算机的设计.....	- 62 -
附录 1 JYS 实验系统硬件平台资源和 FPGA 引脚连接表	- 63 -
附录 2 Quartus II 库及库元件说明	- 65 -
附录 3 JYS-X 数字实验扩展板使用说明	- 67 -

第 1 章 QuartusII 软件使用介绍

Altera 公司的 Quartus II 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。Quartus II 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 Quartus II 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。

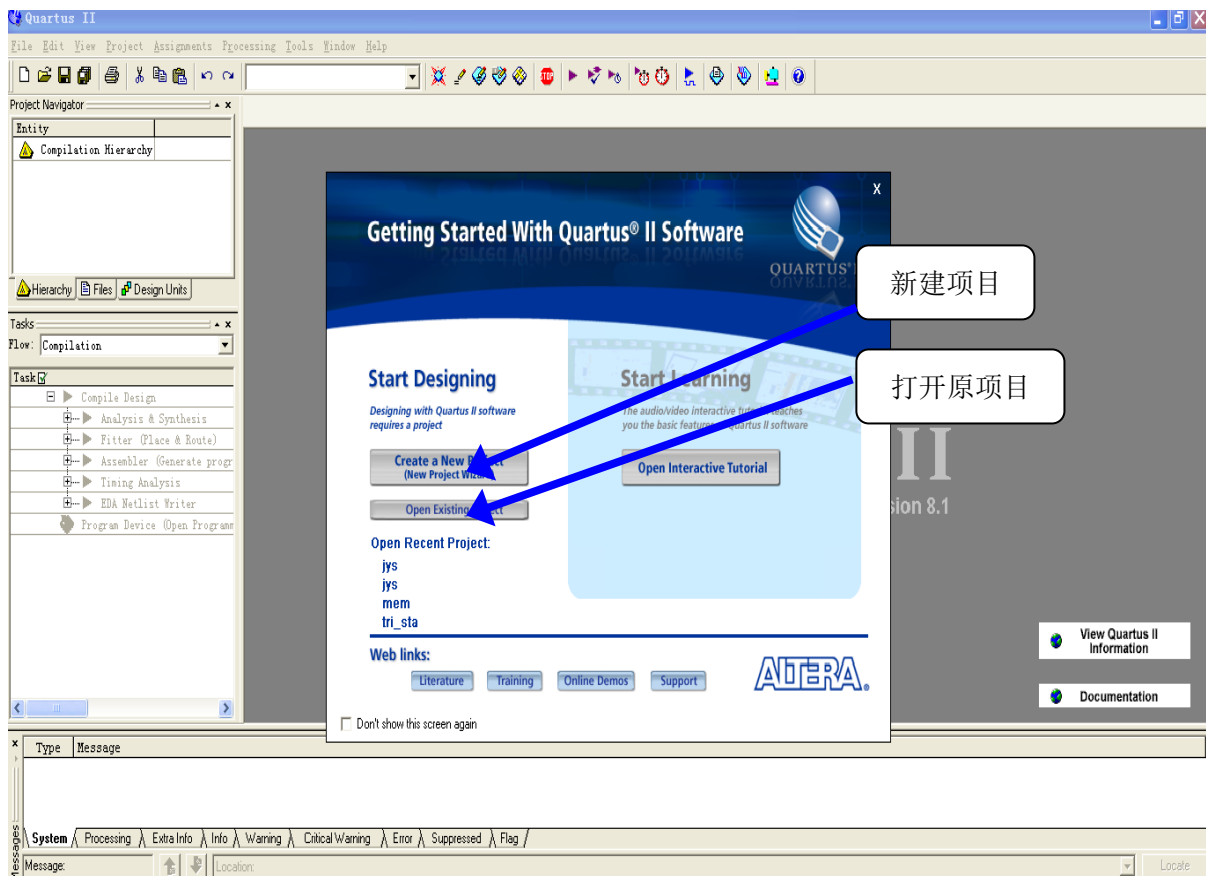


图 1.1 QuartusII 管理器窗口

1.1.1 建立新项目

Quartus II 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。

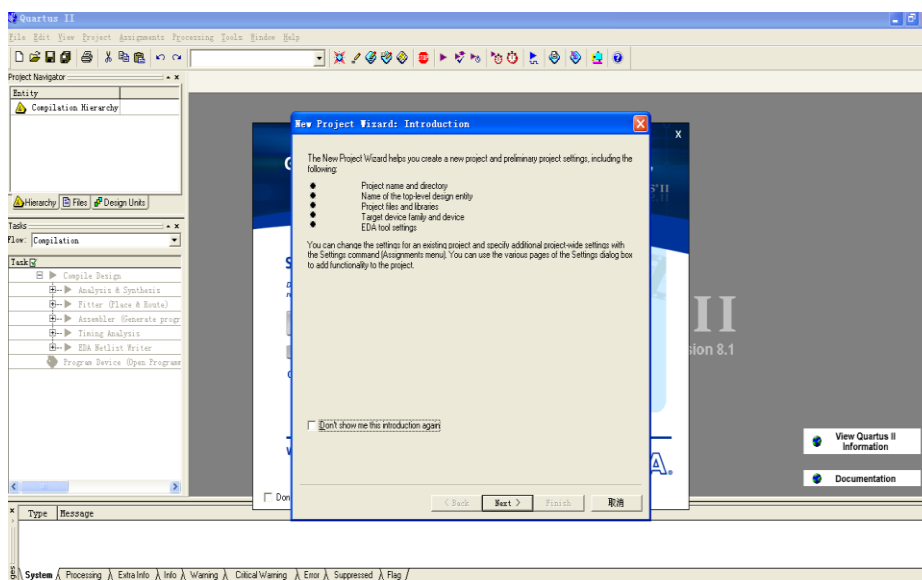


图 1.2 建立新项目的对话框

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）。

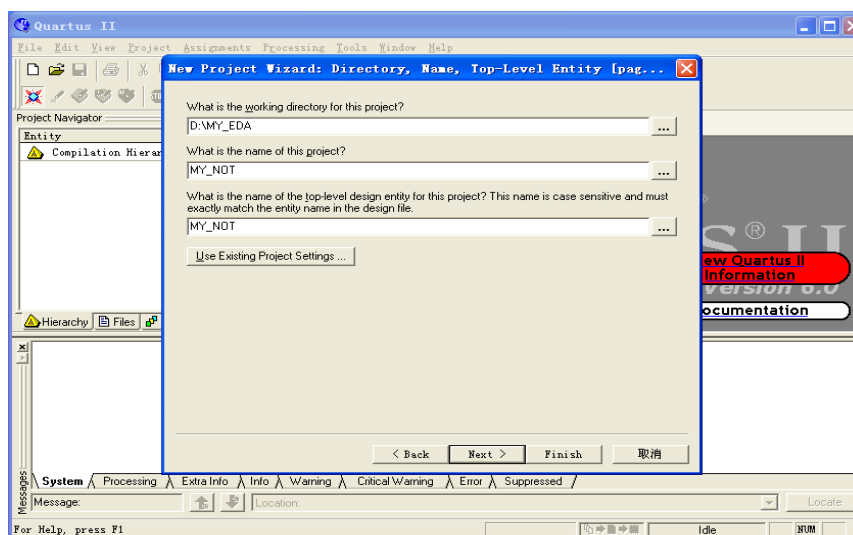


图 1.3 指定项目名对话框

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。

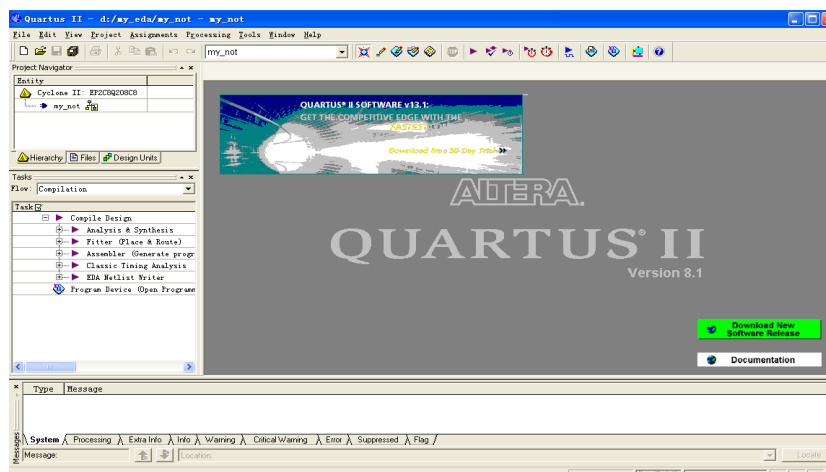


图 1.4 确定项目名对话框

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。

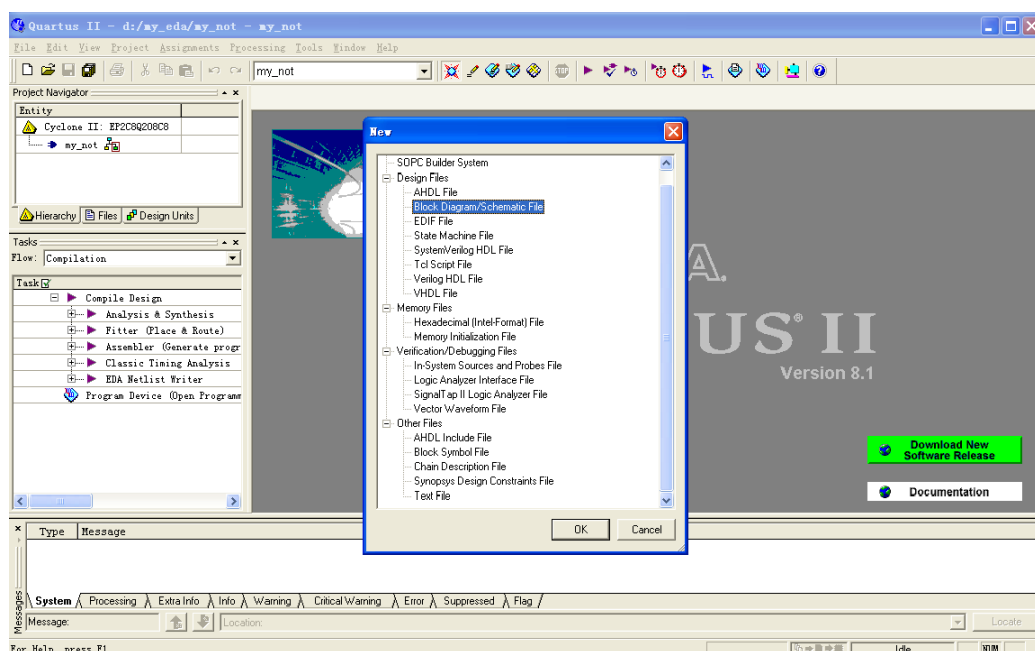


图 1.5 选择图形编辑器对话框

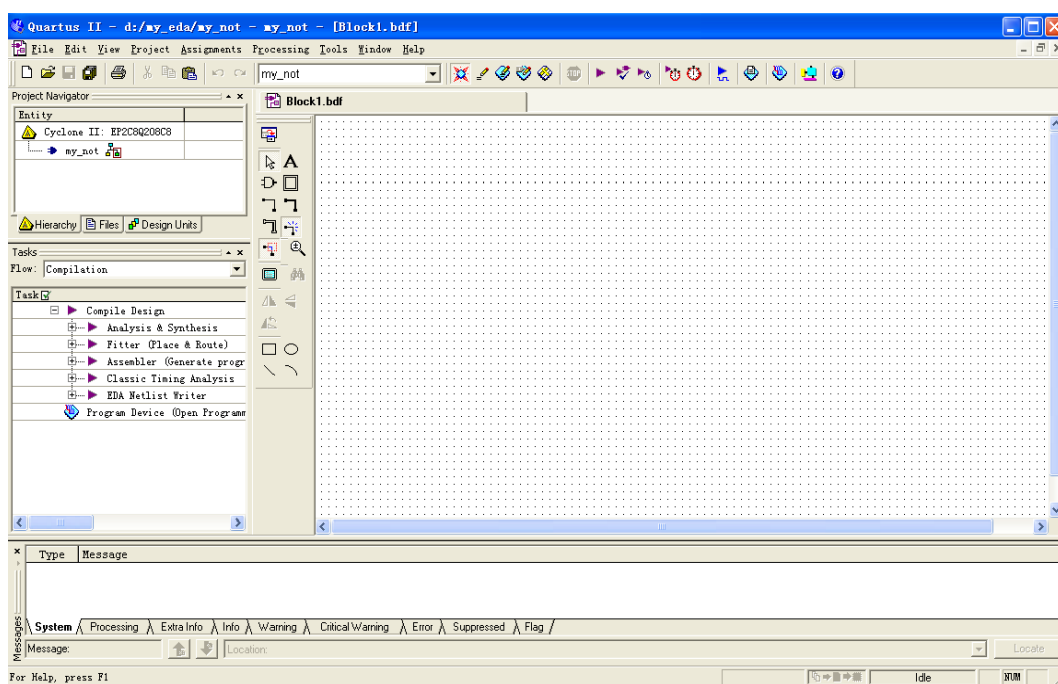


图 1.6 图形编辑器窗口

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。

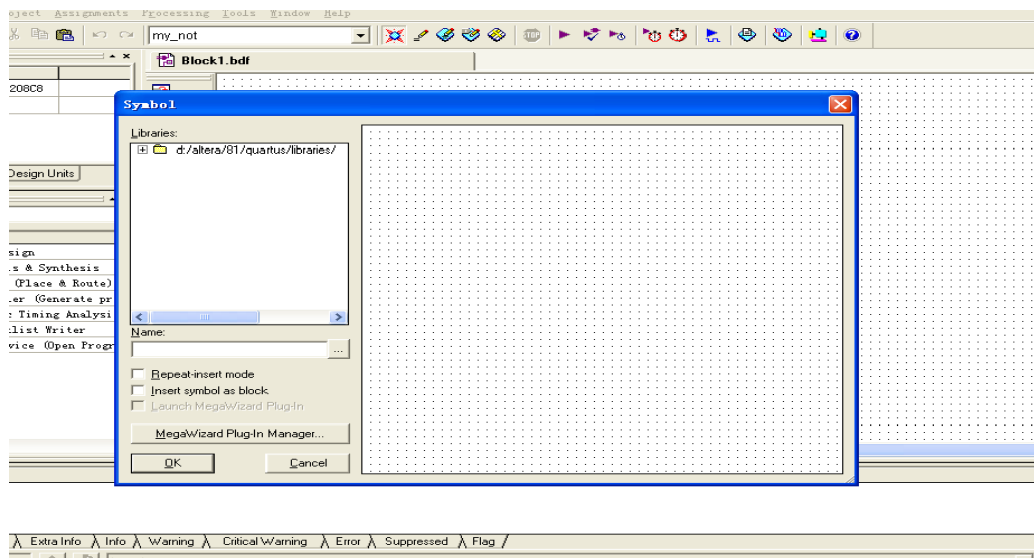


图 1.7 添加反相器元件符号的窗口

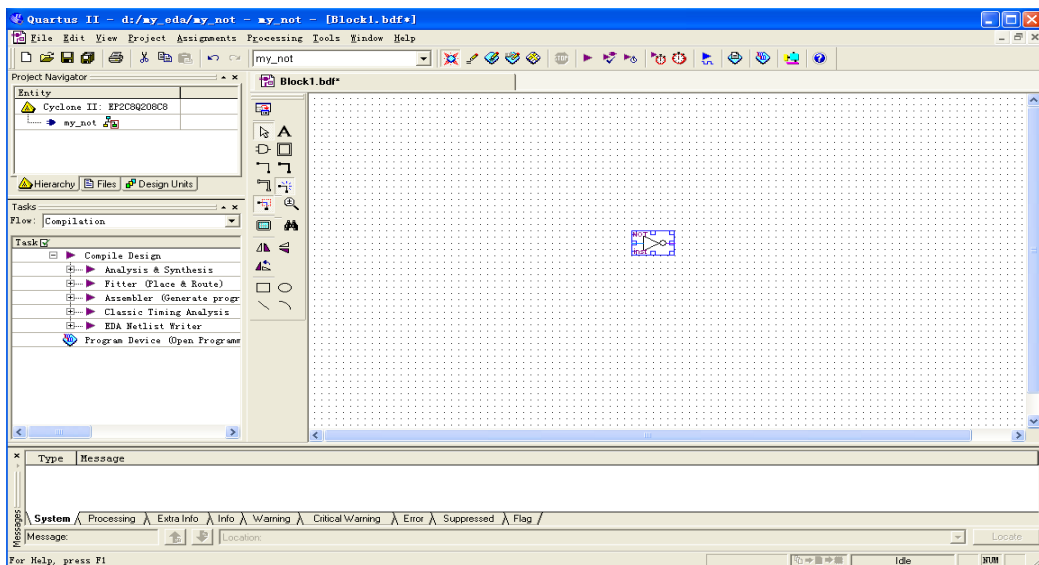


图 1.8 放置反相器窗口

(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。

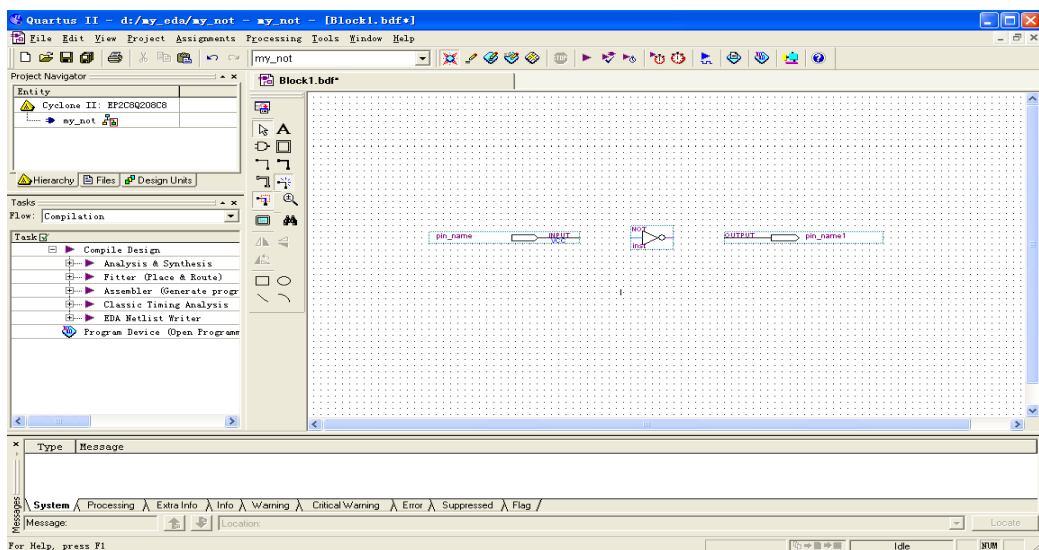



图 1.9 放置了所有元件符号的屏幕

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。

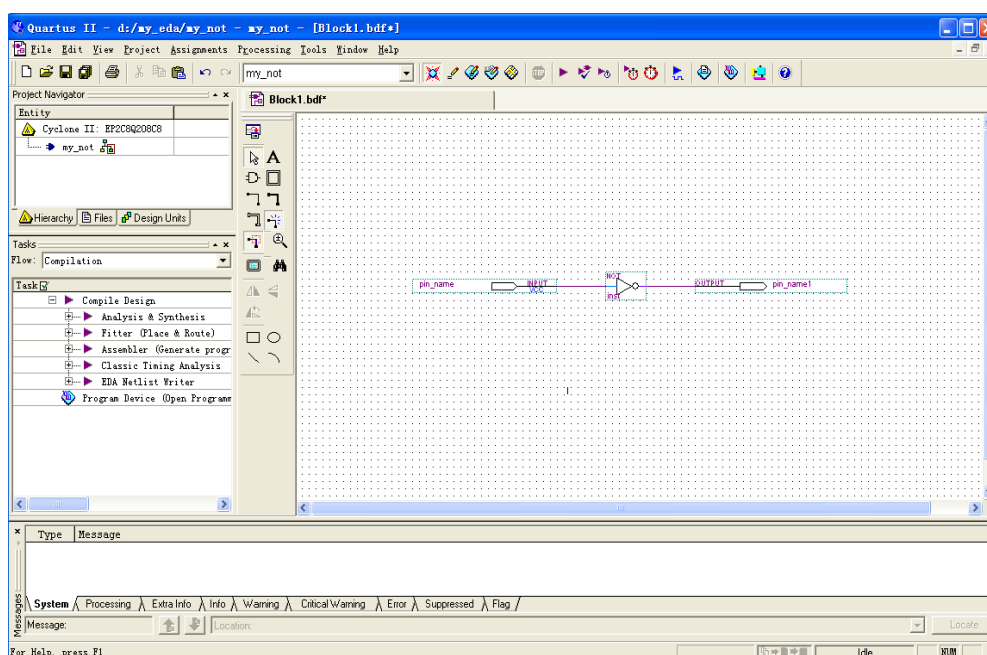


图 1.10 完成全部连线的窗口

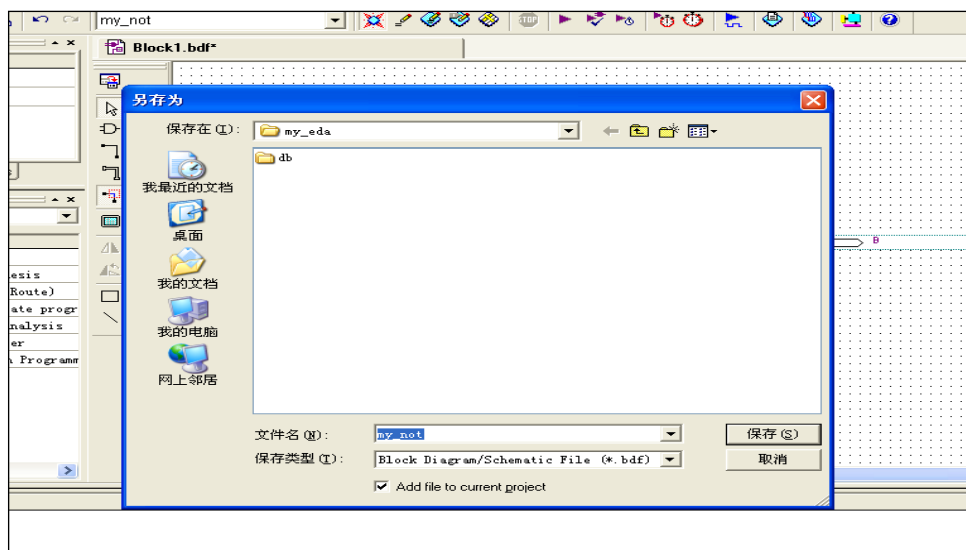


图 1.11 欲保存文件前的窗口

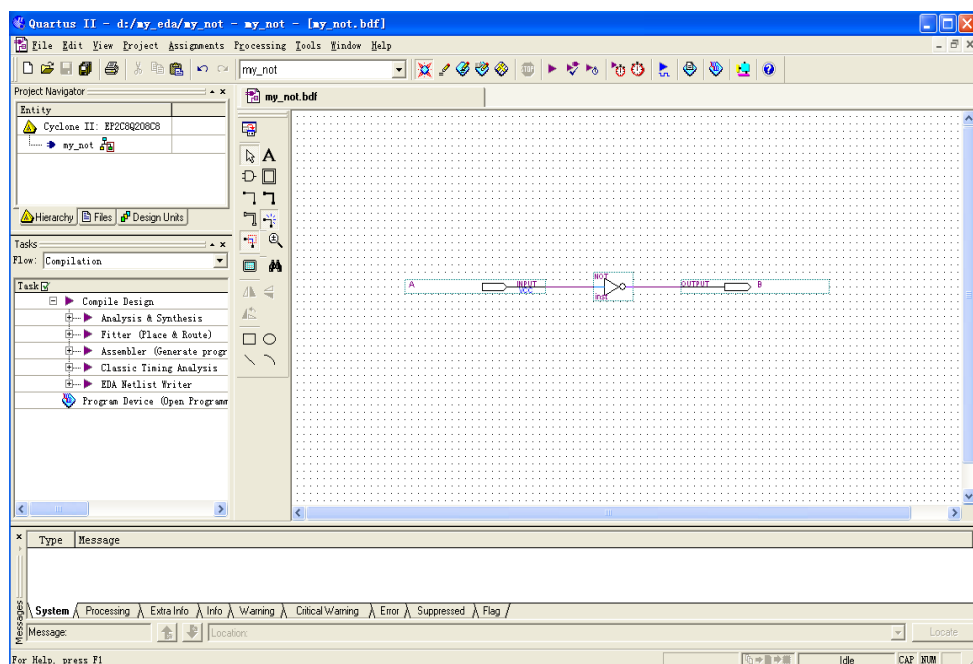


图 1.12 文件已保存的窗口

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。

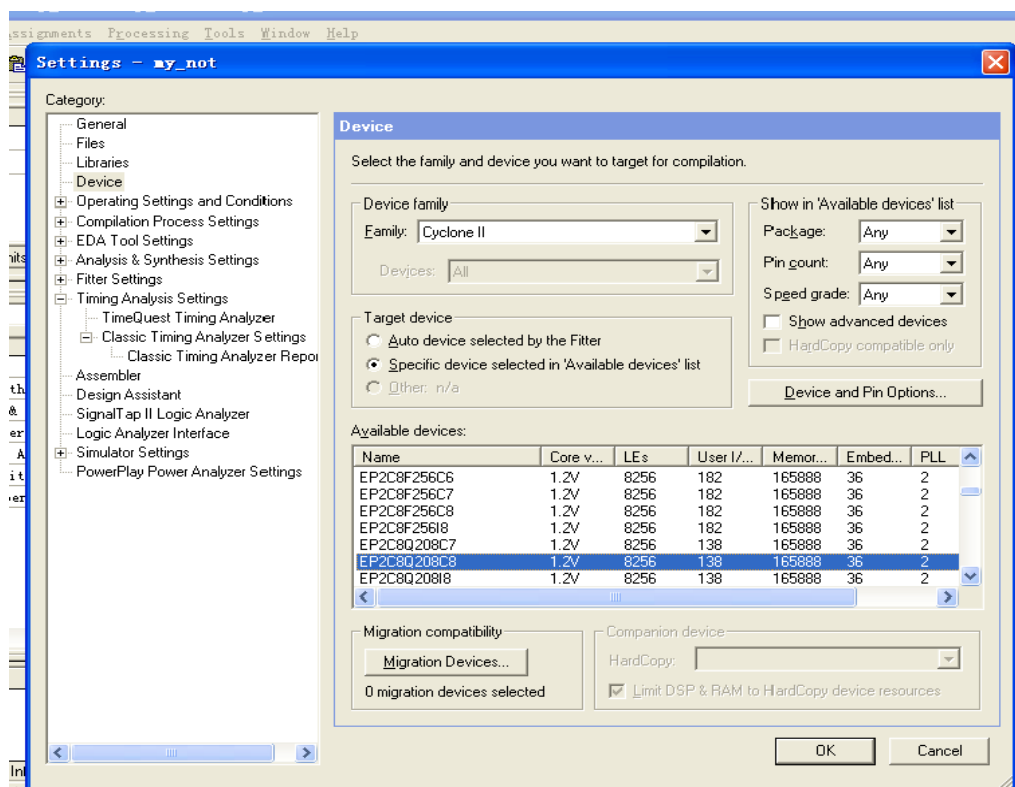


图 1.13 选择器件的窗口

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。

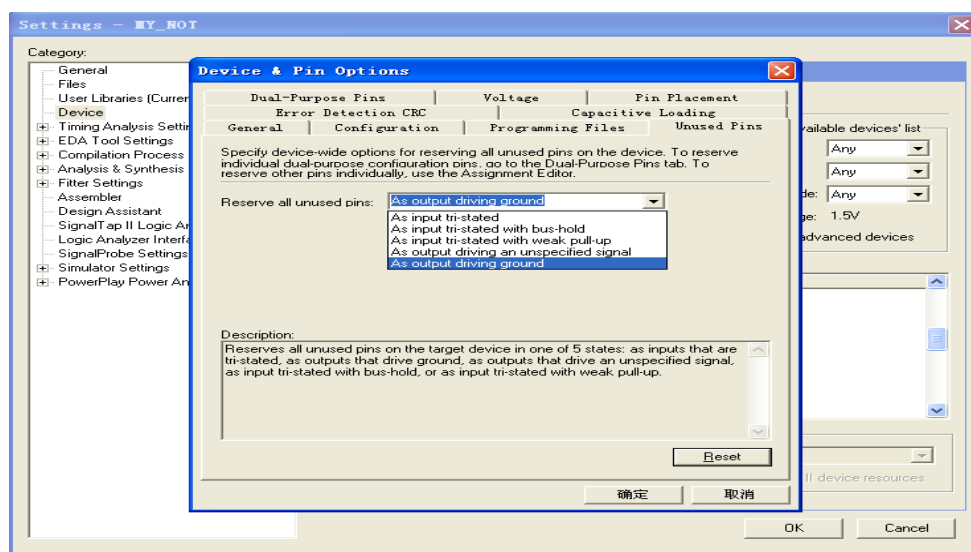



图 1.14 未使用引脚的处理

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

Quartus II 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。

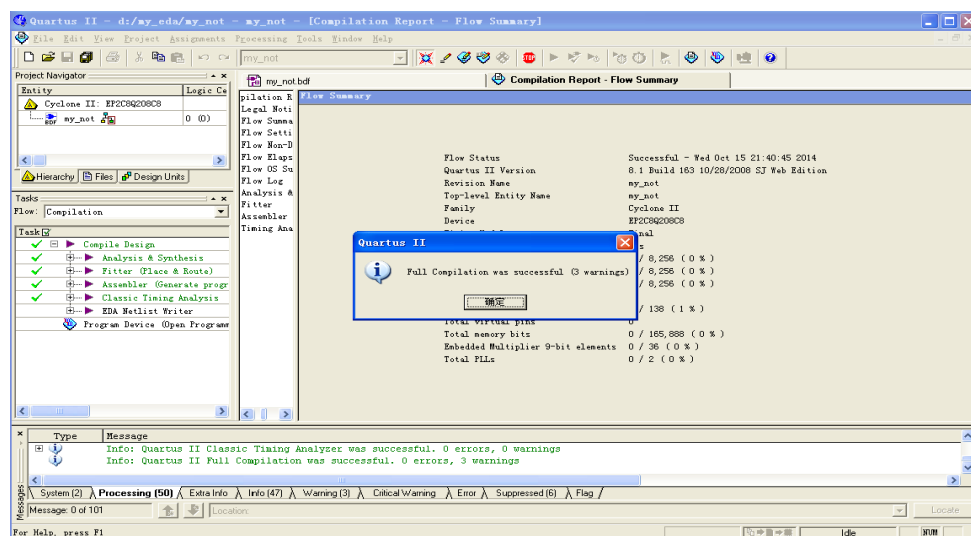


图 1.15 编译完成的窗口

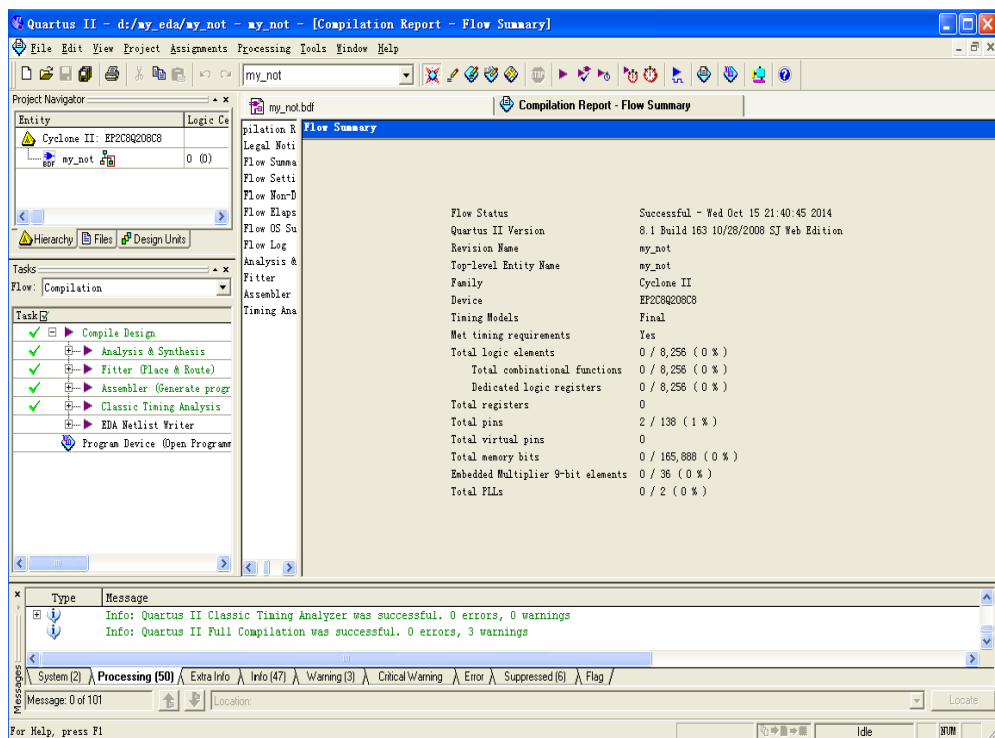


图 1.16 编译完成后的信息窗口

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。

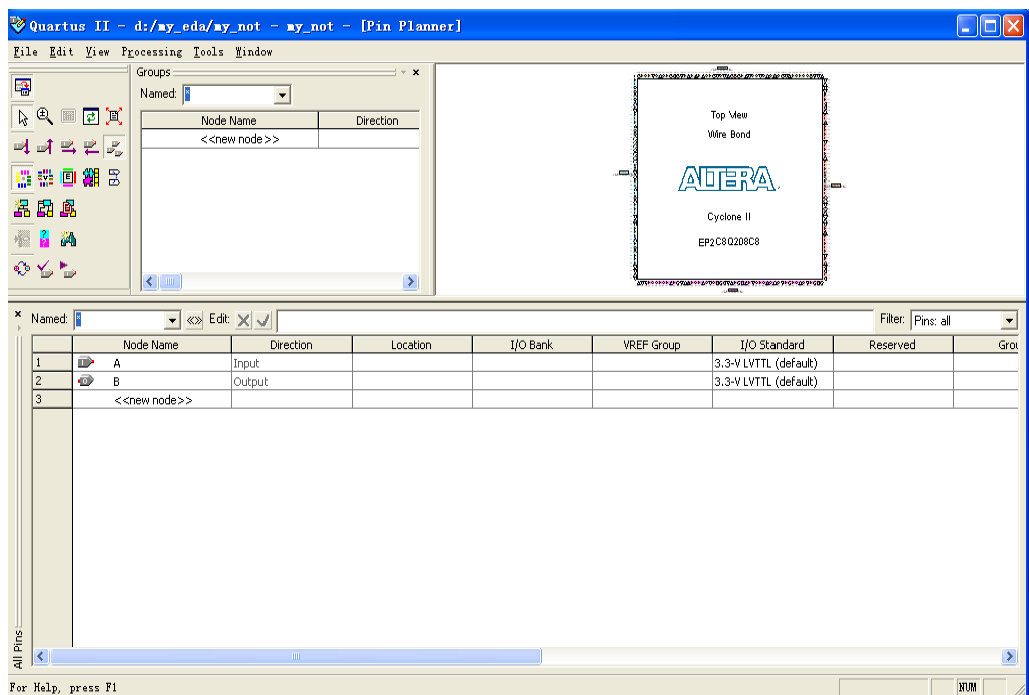


图 1.17 引脚分配窗口

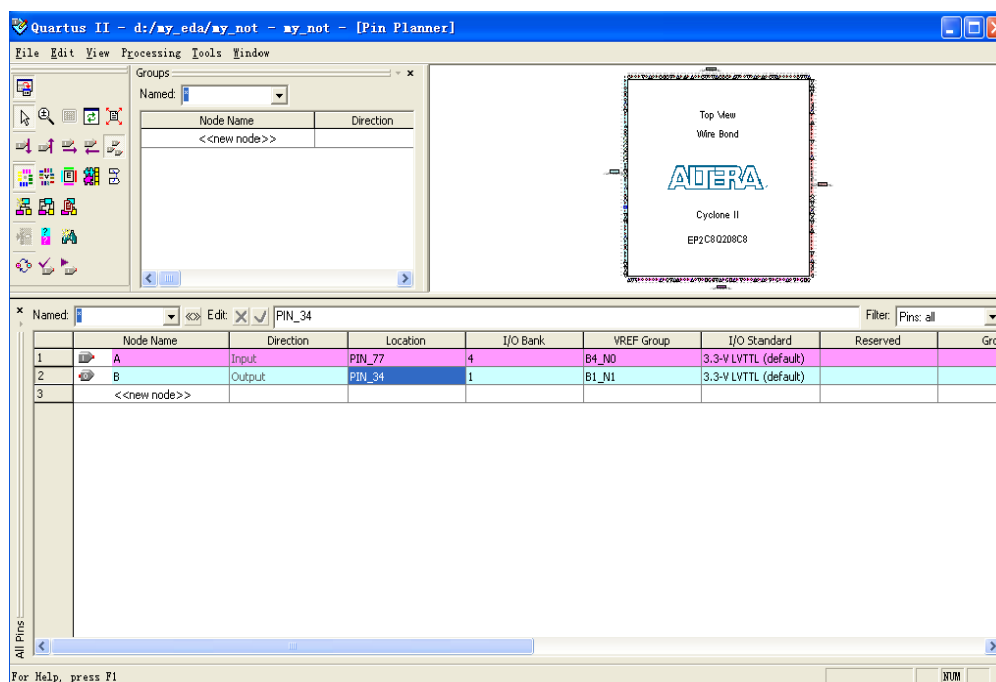



图 1.18 指定引脚时的屏幕显示

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 Quartus II 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，Quartus II 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。

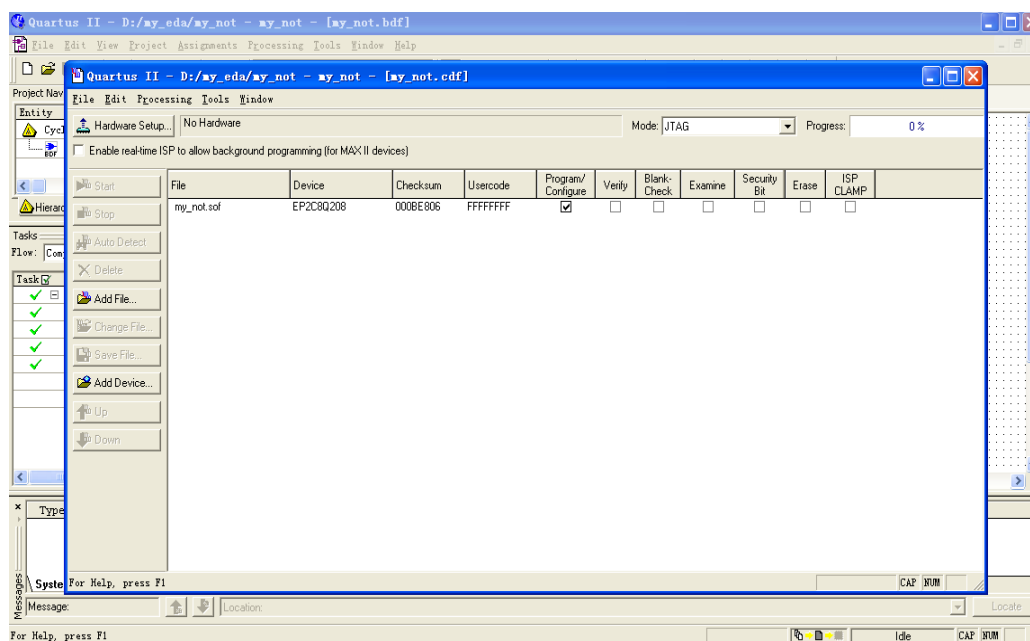



图 1.19 编程器窗口

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  Hardware Setup... 按钮, 按图 1.21 所示窗口设置即可。

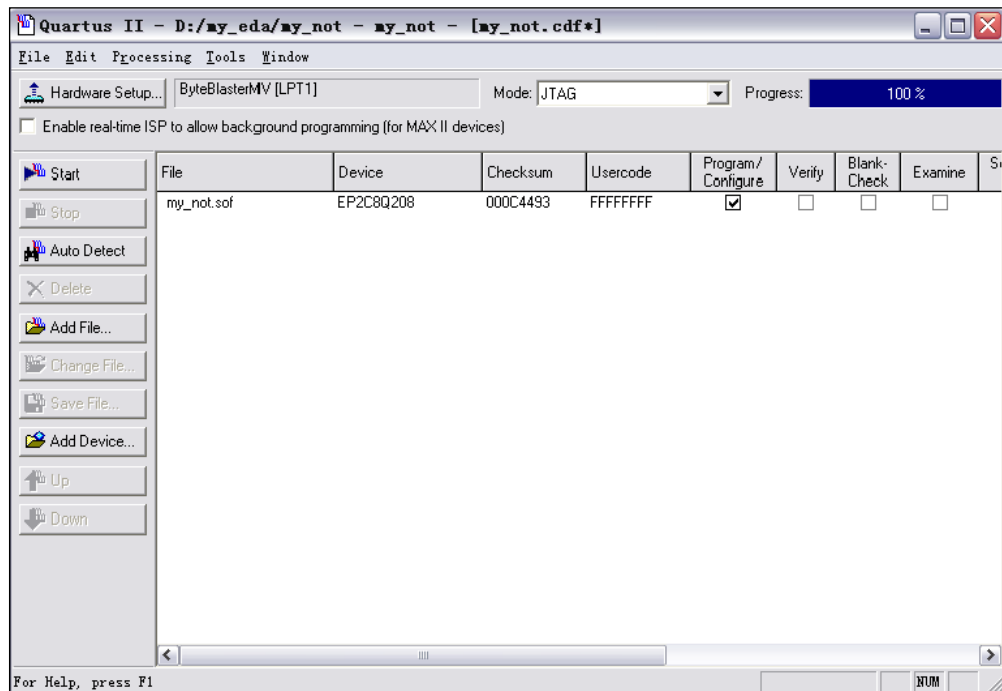


图 1.20 下载完成窗口

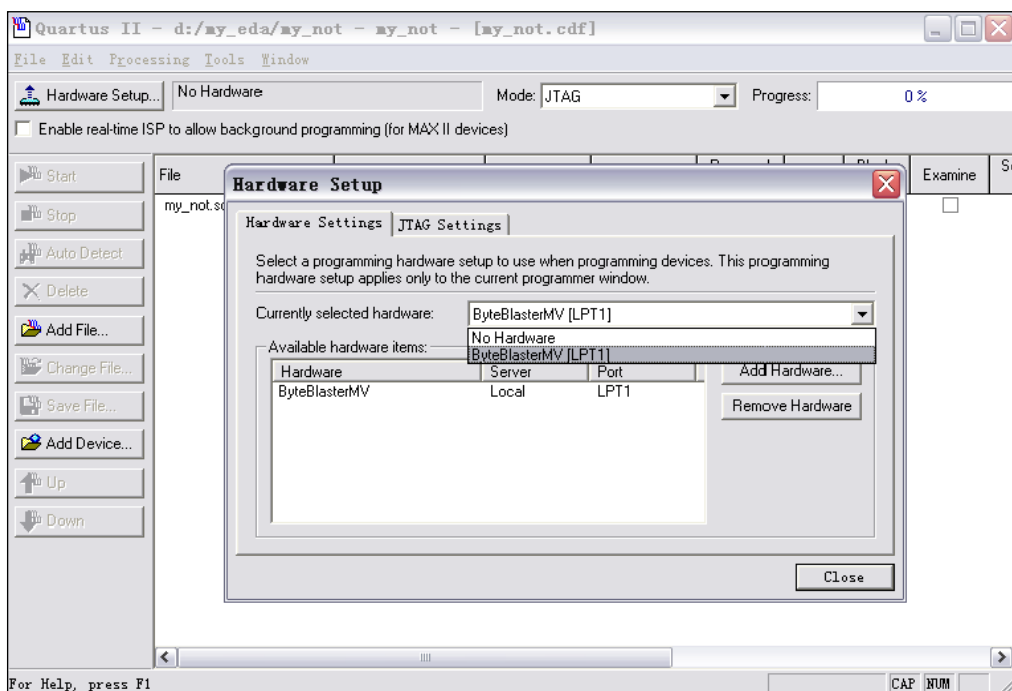


图 1.21 “Hardware Setup” 窗口

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

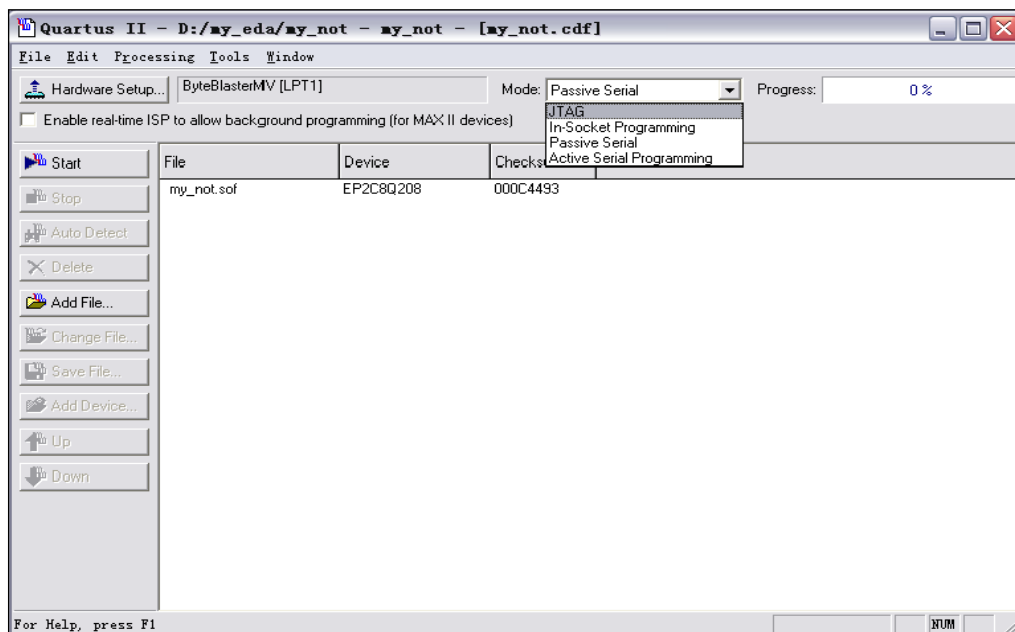


图 1.22 下载模式选择窗口

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

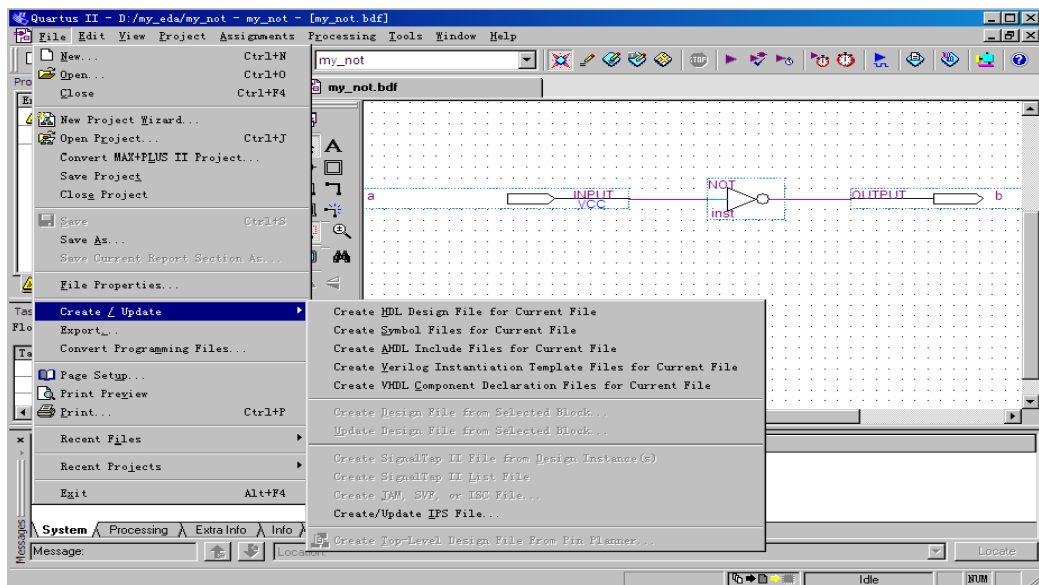


图 1.23 选择符号图生成命令窗口

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。

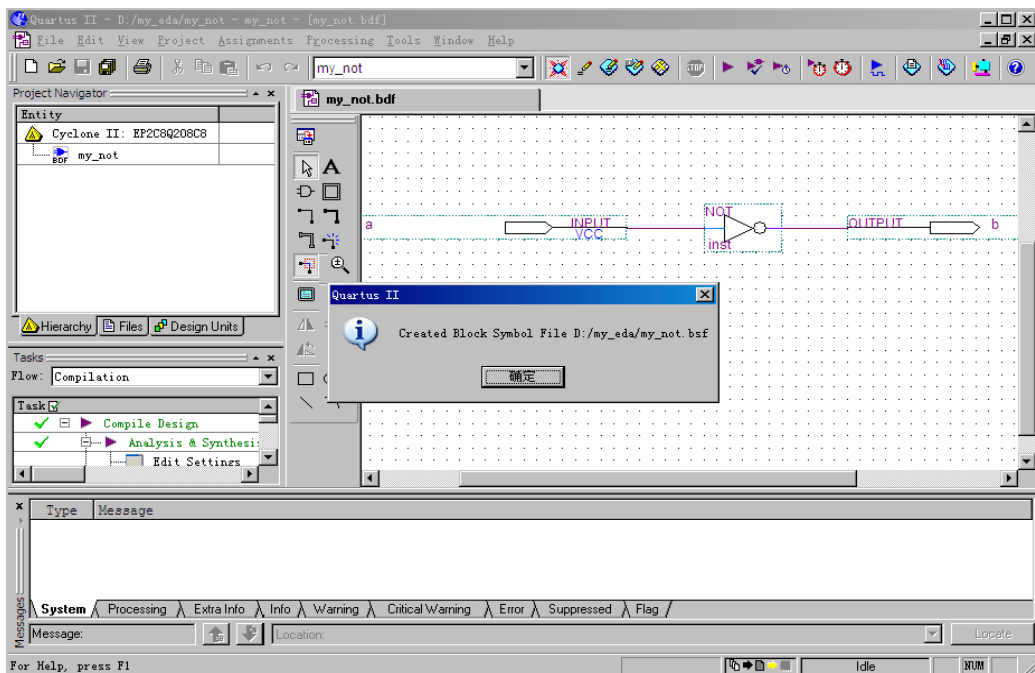


图 1.24 符号图生成成功的提示窗口

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 Quartus II 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与否。下面介绍在 Quartus II 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。

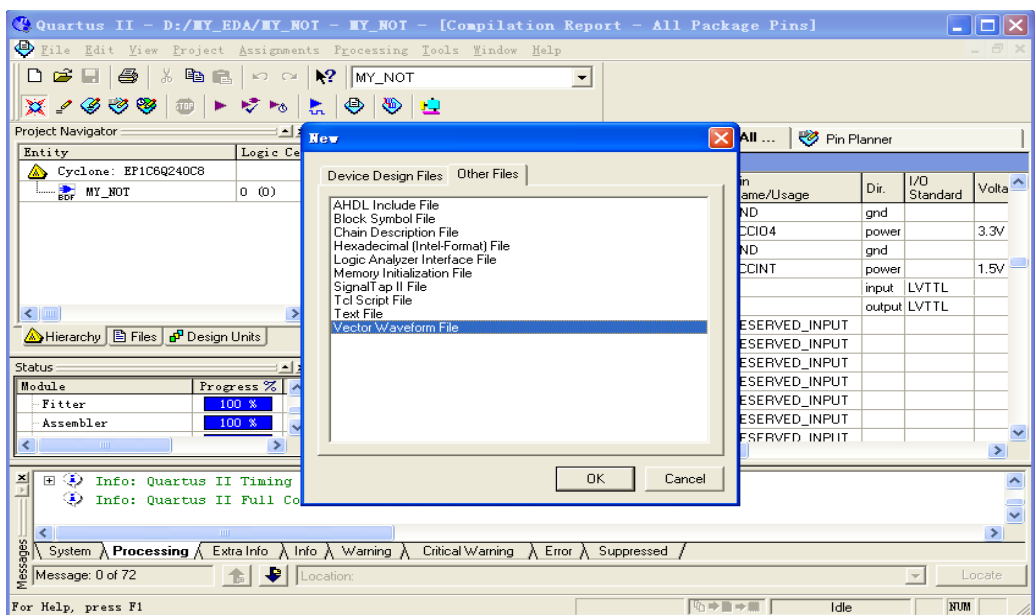


图 1.25 Other Files 标签

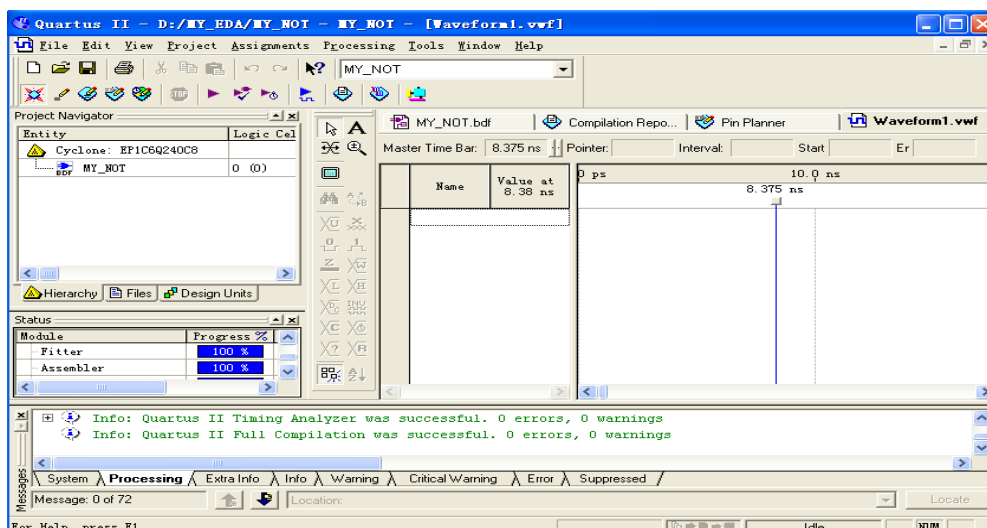


图 1.26 编辑仿真文件窗口

(3) 在图 1.26 窗口中，双击“Name”下方的空白处，弹出“Insert Node 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。

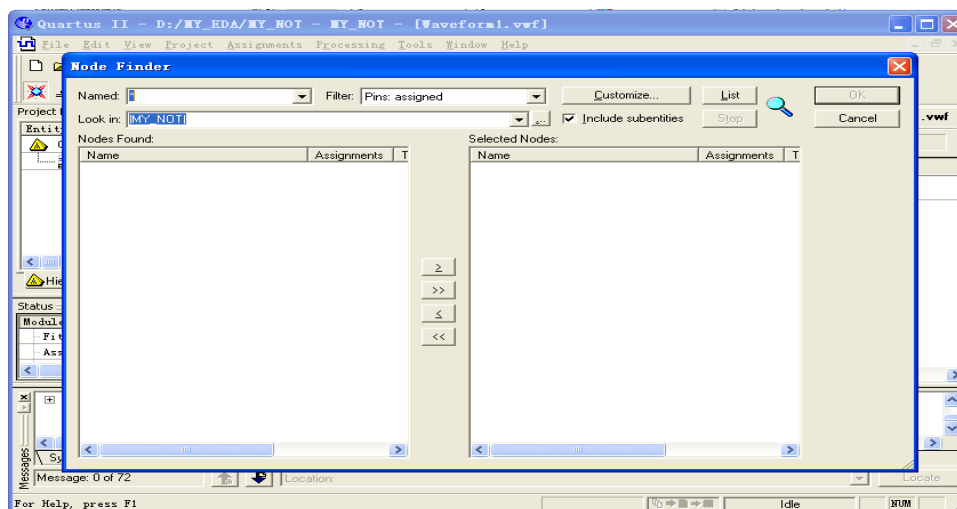


图 1.27 “Node Finder”对话框

(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。

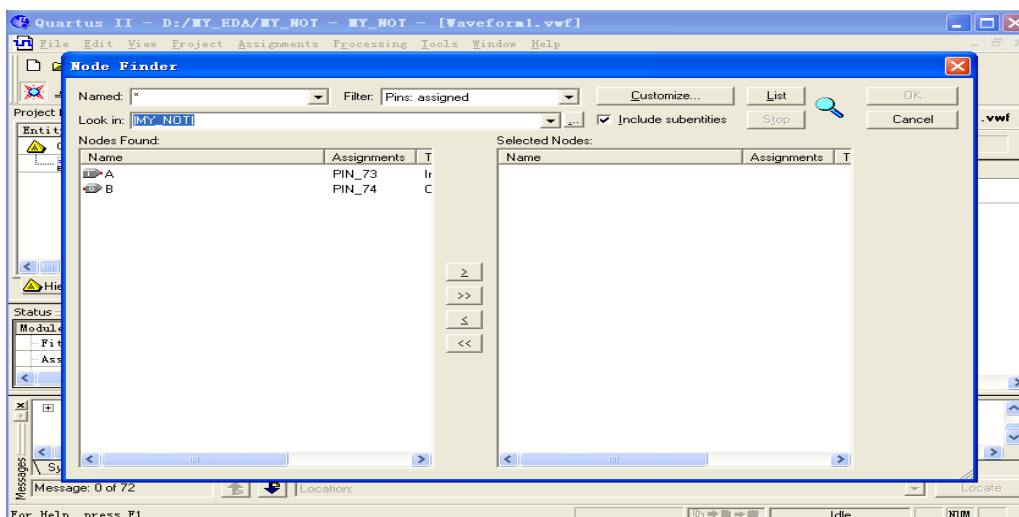


图 1.28 “Node Finder”对话框

(5) 在图 1.24 中，选中欲观察的输入或输出引脚，单击“ \geq ”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。

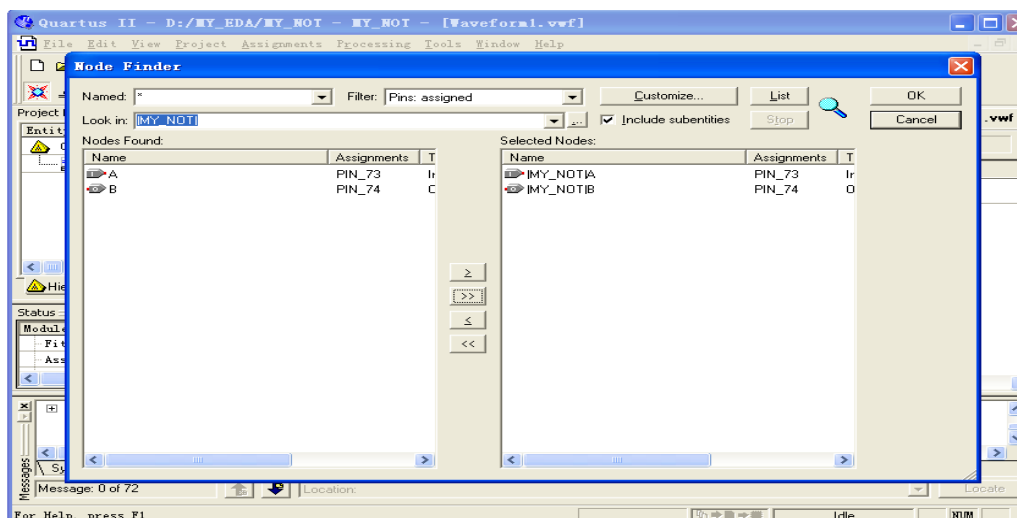


图 1.29 选择欲观察的输入/输出对话框

(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。

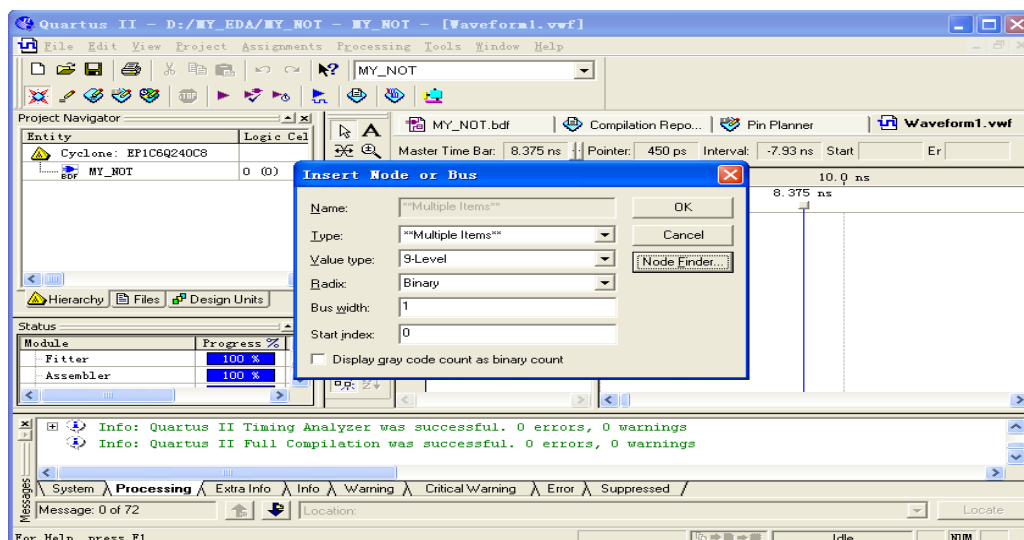


图 1.30 输入/输出引脚选择完成后的窗口

(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。

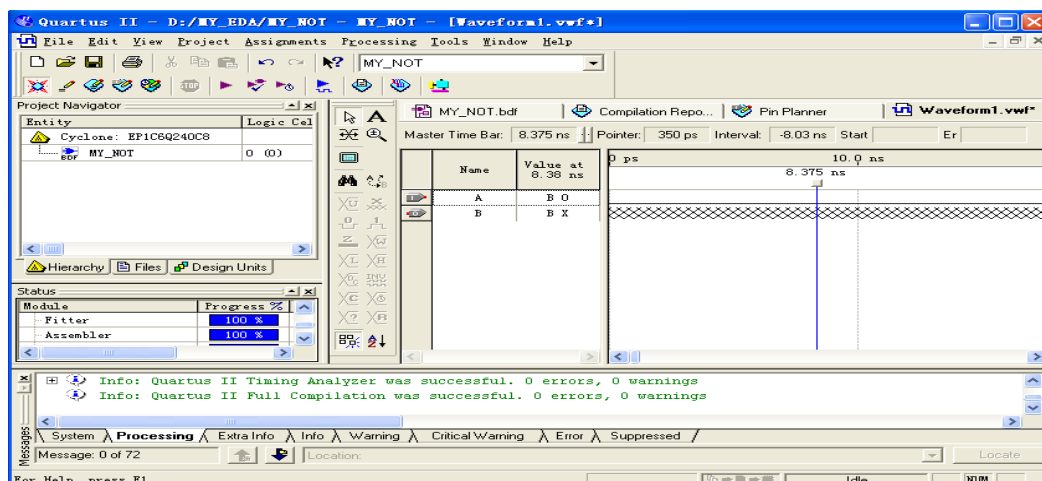


图 1.31 波形编辑窗口

(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

“X”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。

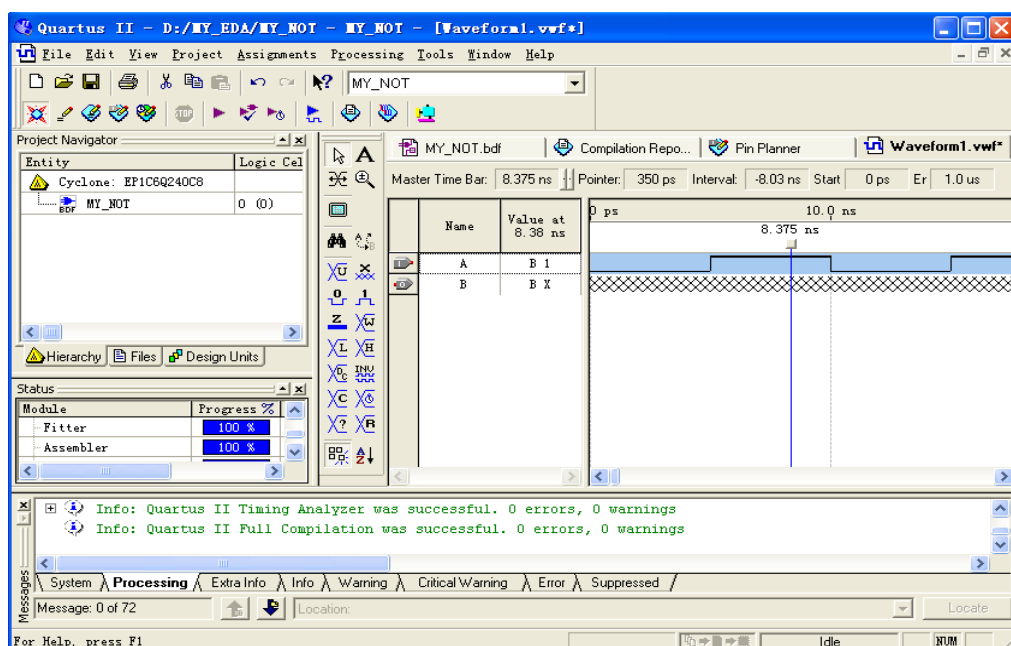


图 1.32 完成仿真波形编辑的窗口

(9) 保存建立的波形文件。

2. 设计的仿真实验

(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“运行”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。

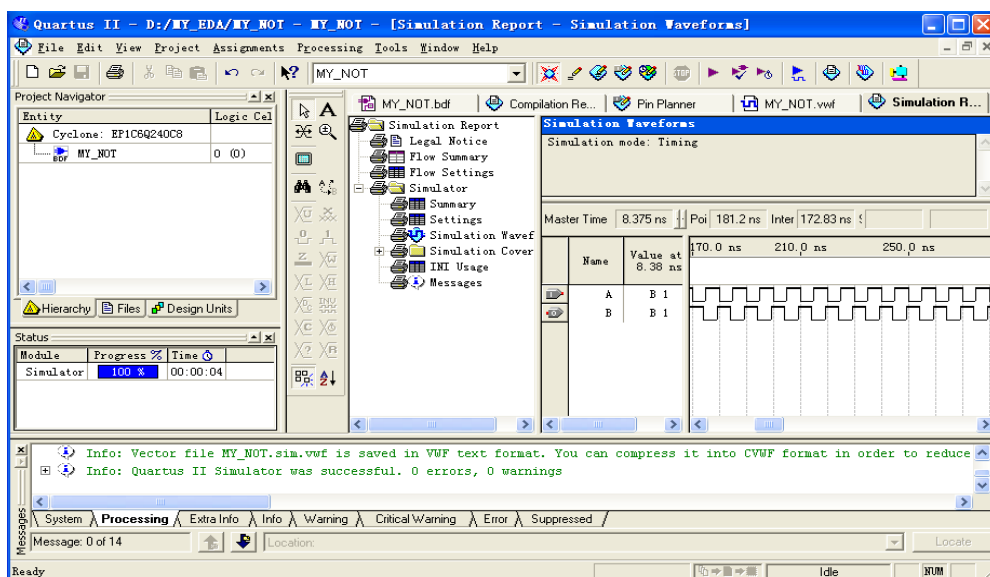


图 1.33 仿真完成后的窗口

1.3 Quartus II 文本输入法使用介绍

Quartus II 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 Wizard... 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。

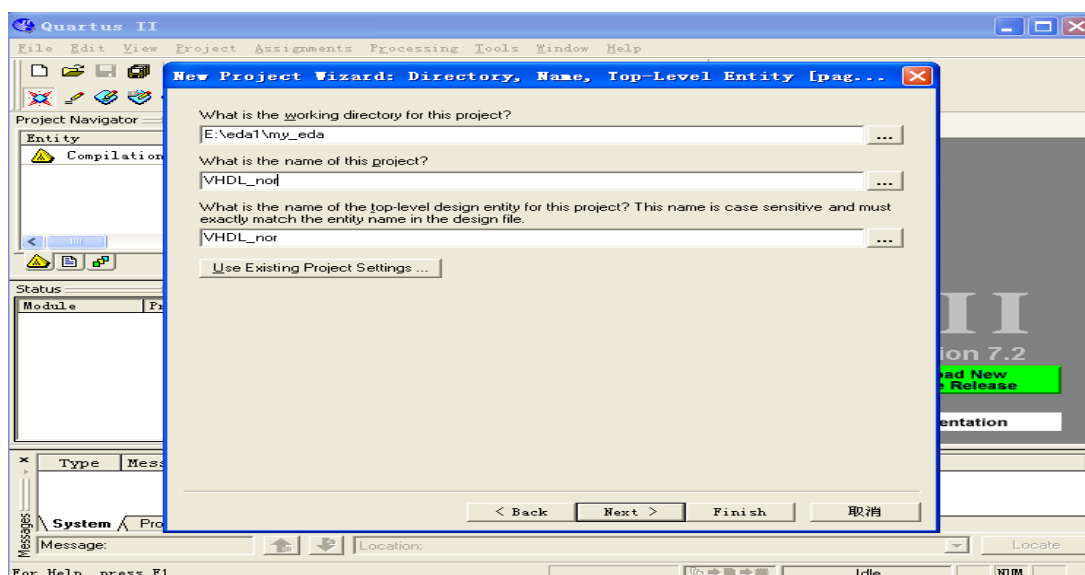


图 1.34 建立新项目窗口

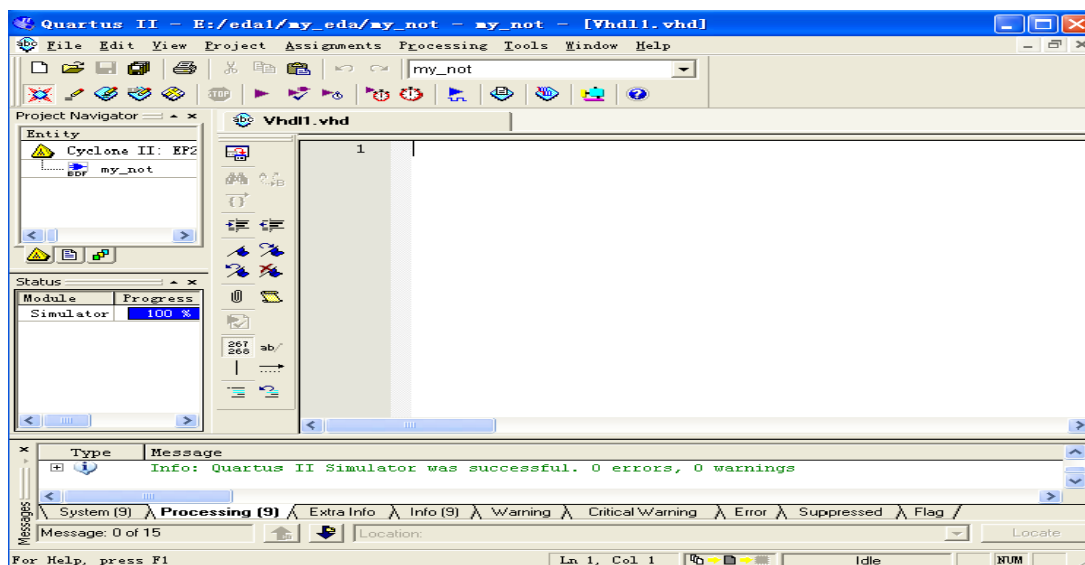


图 1.35 空白的文本编辑器窗口

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。

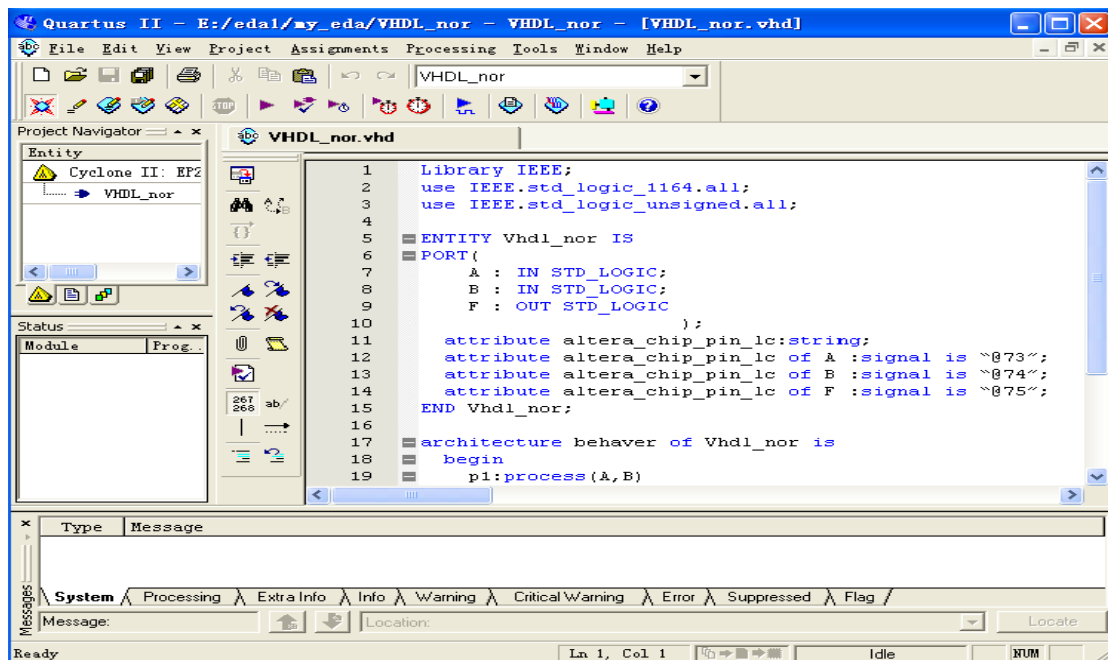


图 1.36 完成文本输入后的窗口

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 behavior 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 behavior;
```

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。

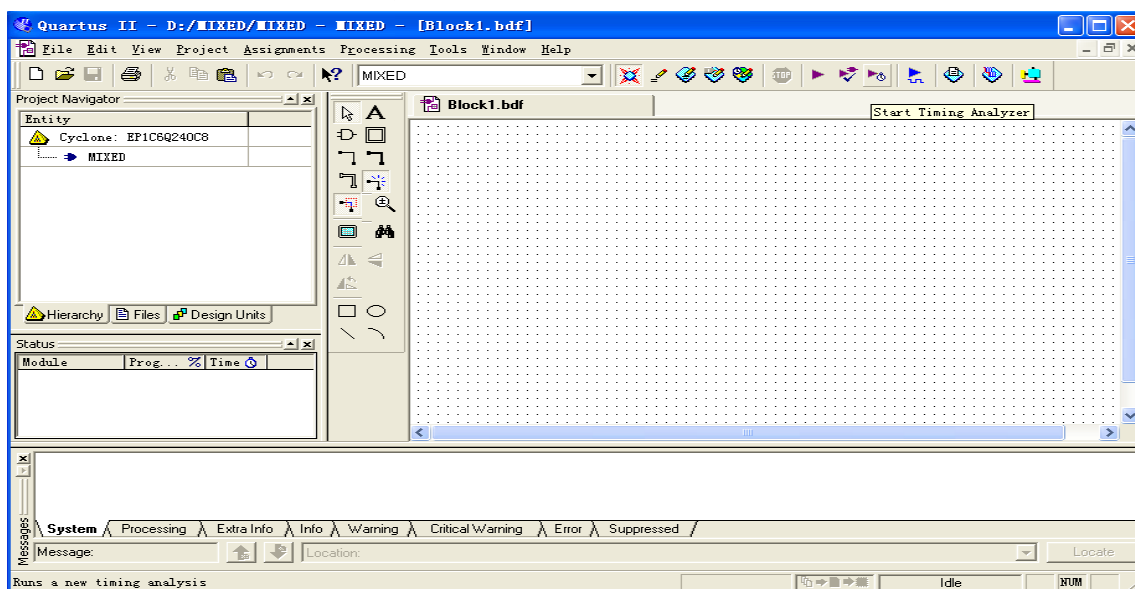


图 1.37 建立 MIXED 工程项目窗口

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。

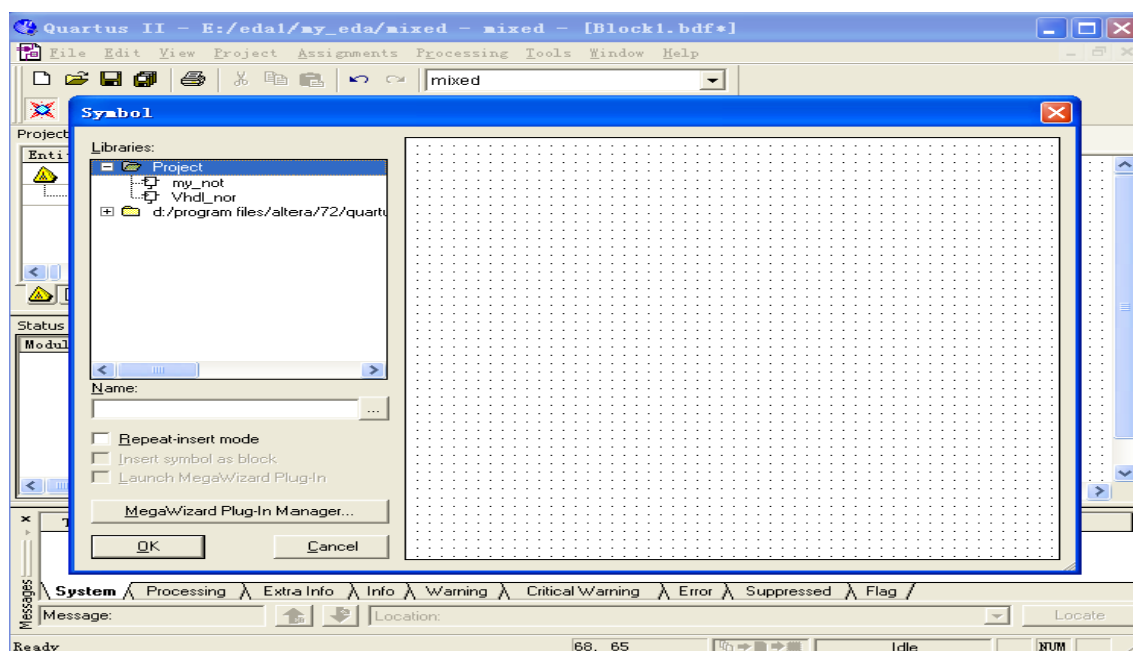


图 1.38 用户工作库窗口

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。

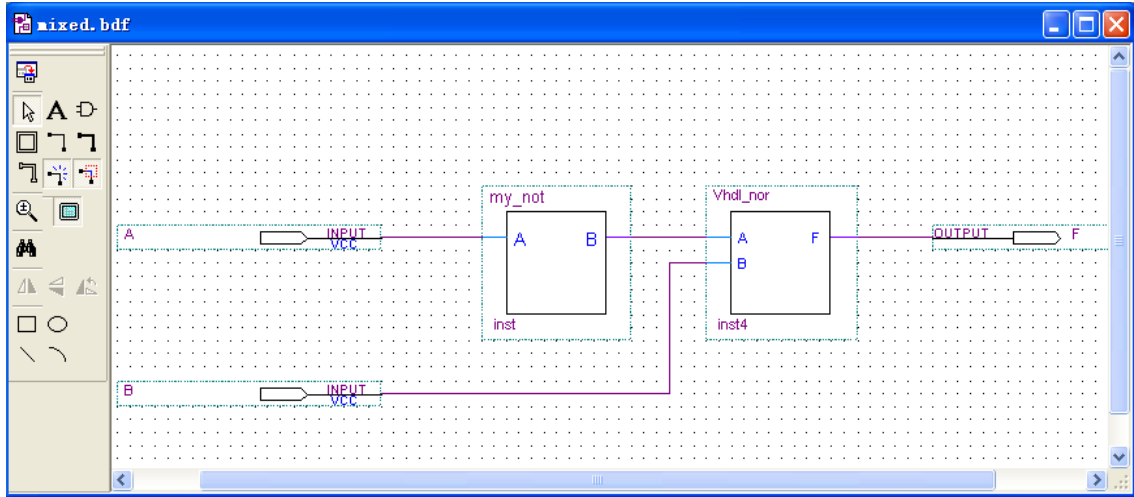


图 1.39 完成后的 mixed 原理图

- 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。

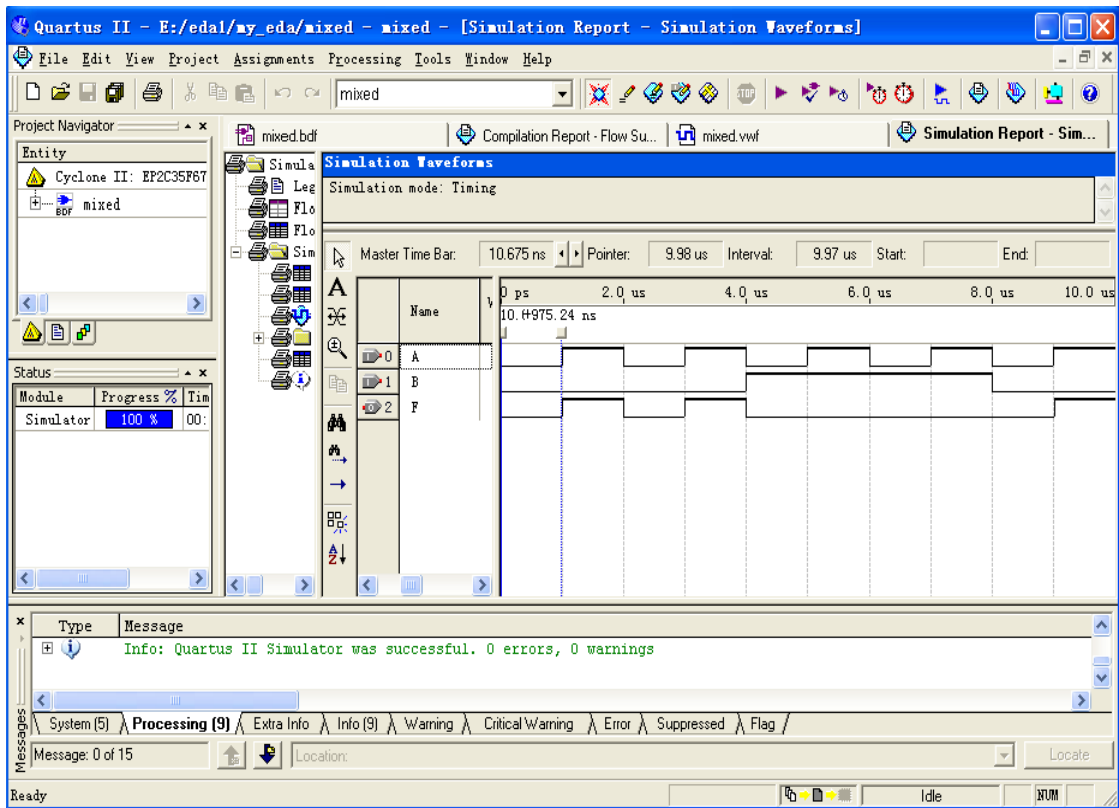


图 1.40 mixed 项目的仿真波形

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。

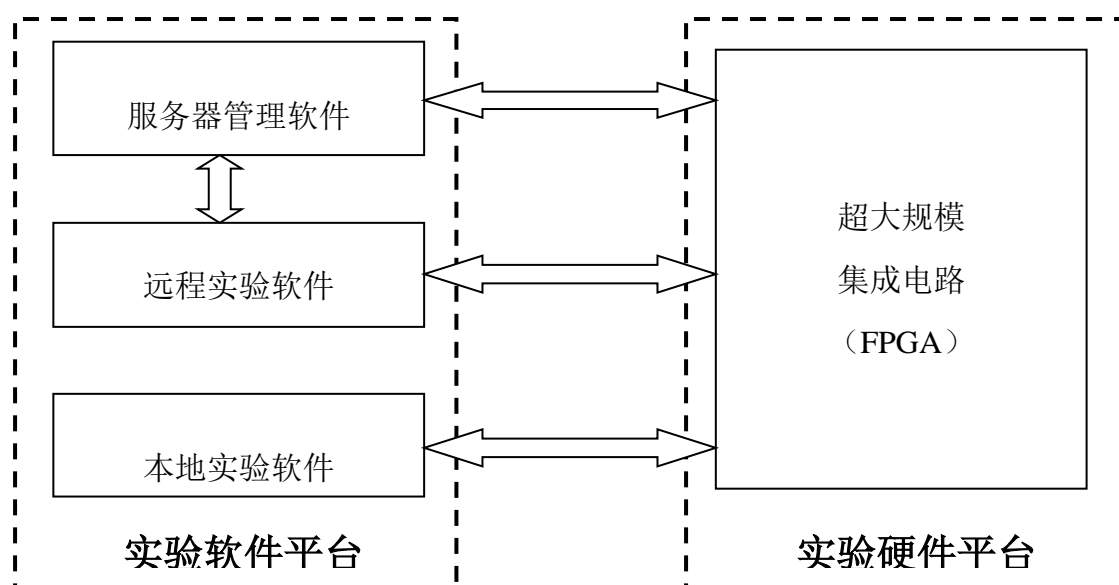


图 2.1 实验系统结构框图

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。

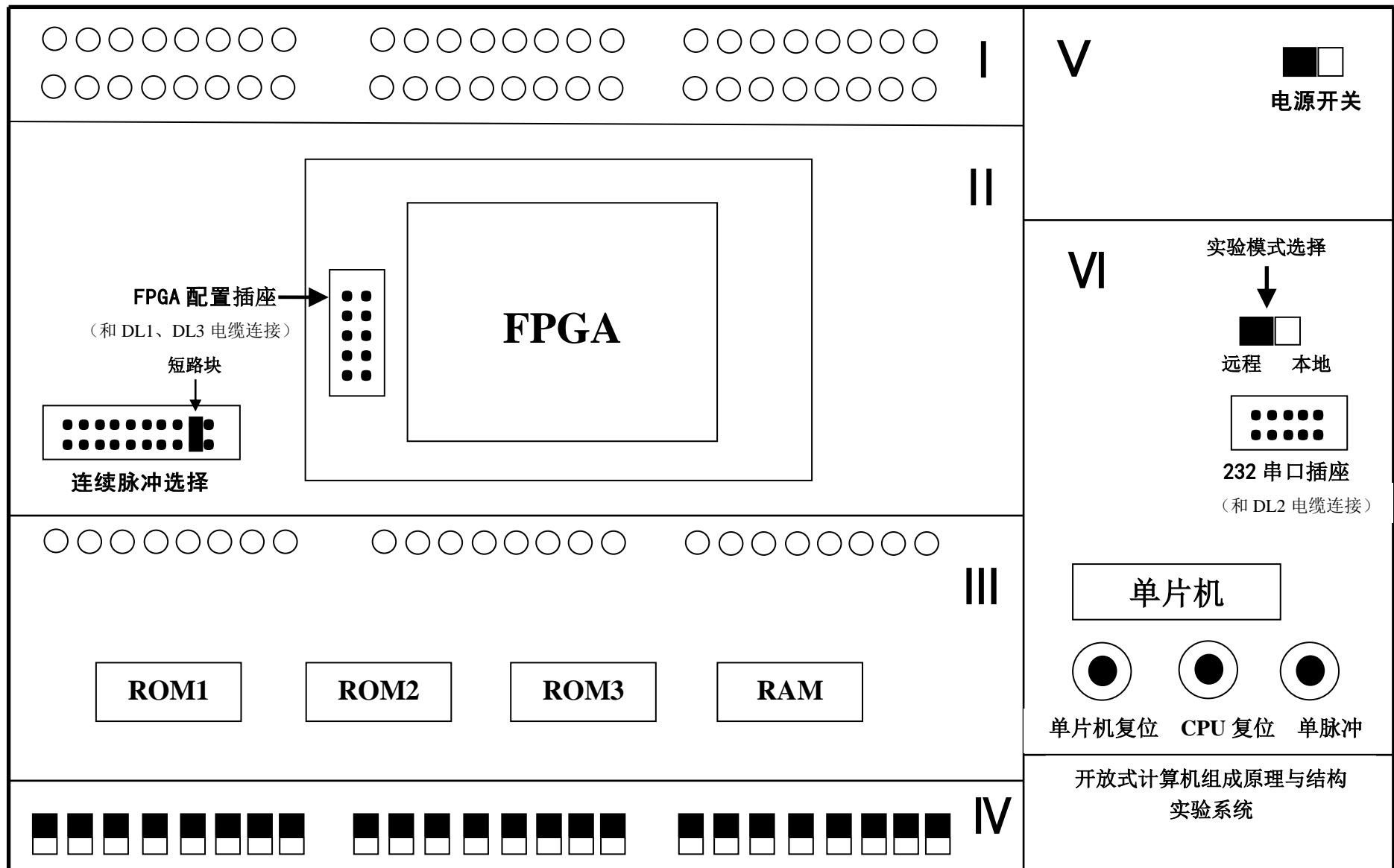
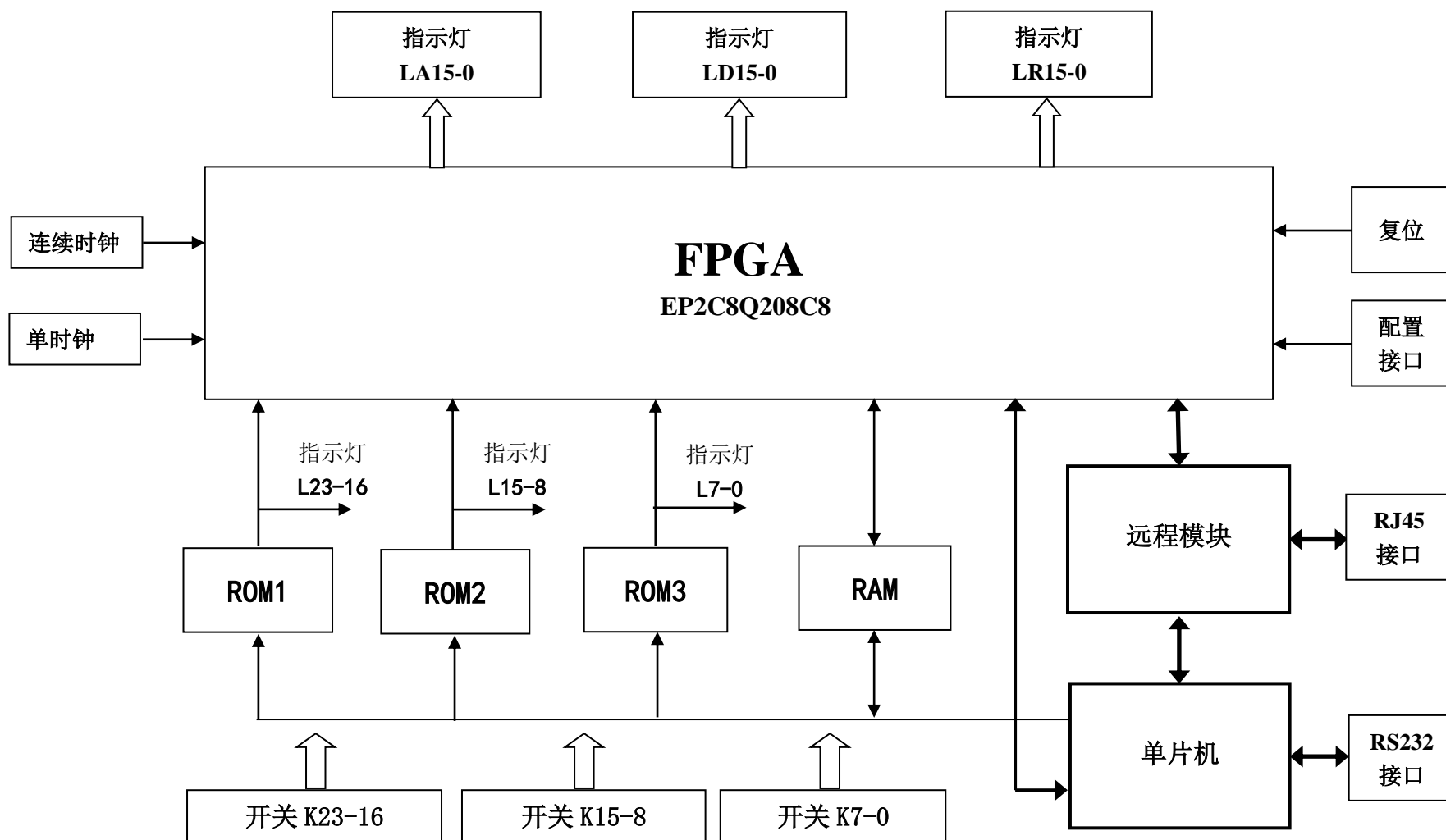


图 2.2 硬件实验平台结构示意图



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 实验软件介绍

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”。

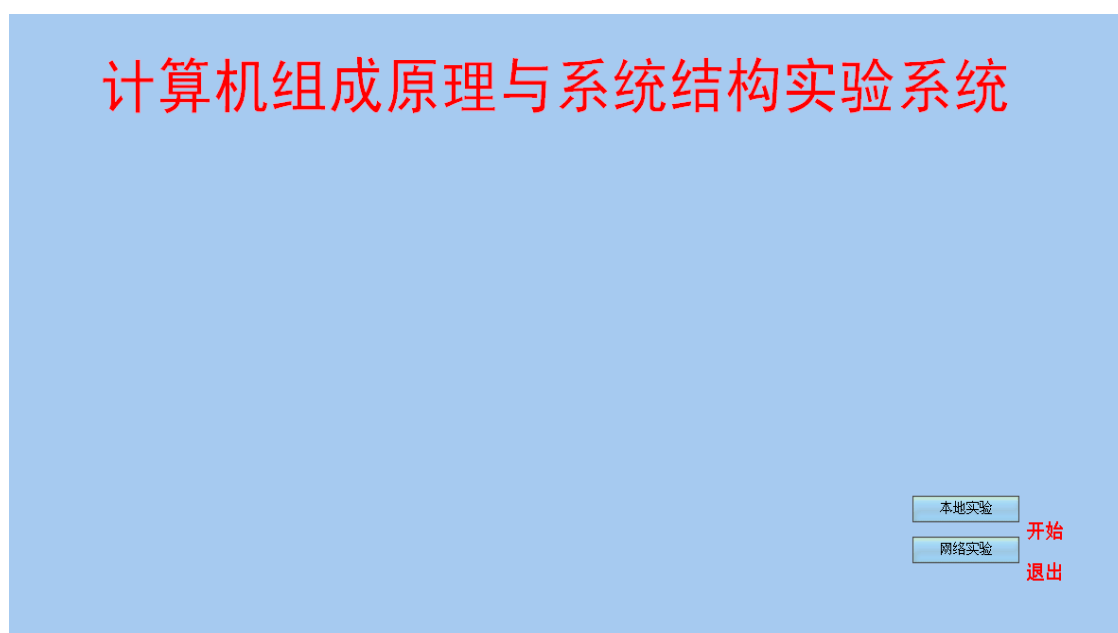


图 2.4 实验程序开始界面

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。

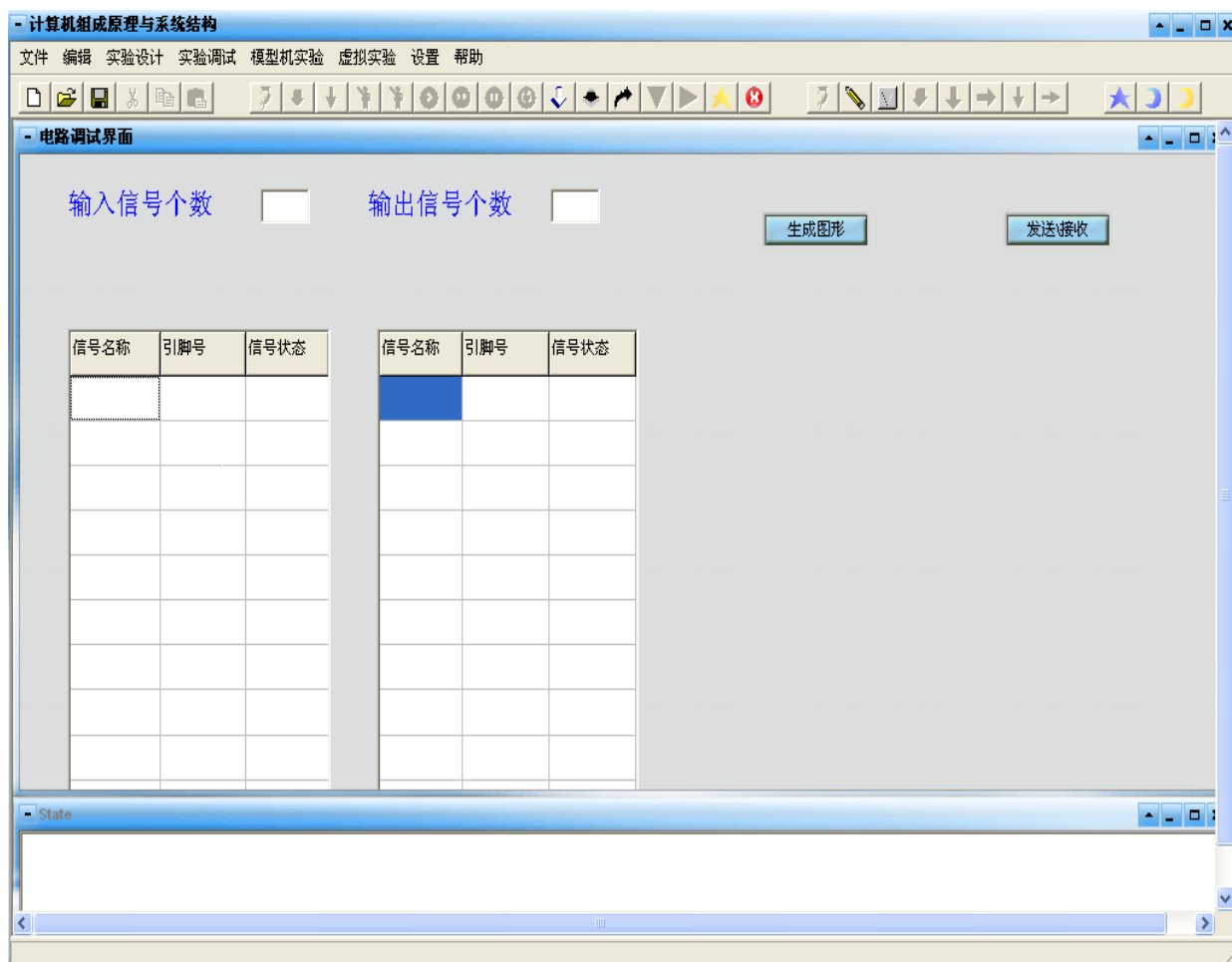


图 2.5 实验程序主界面

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。

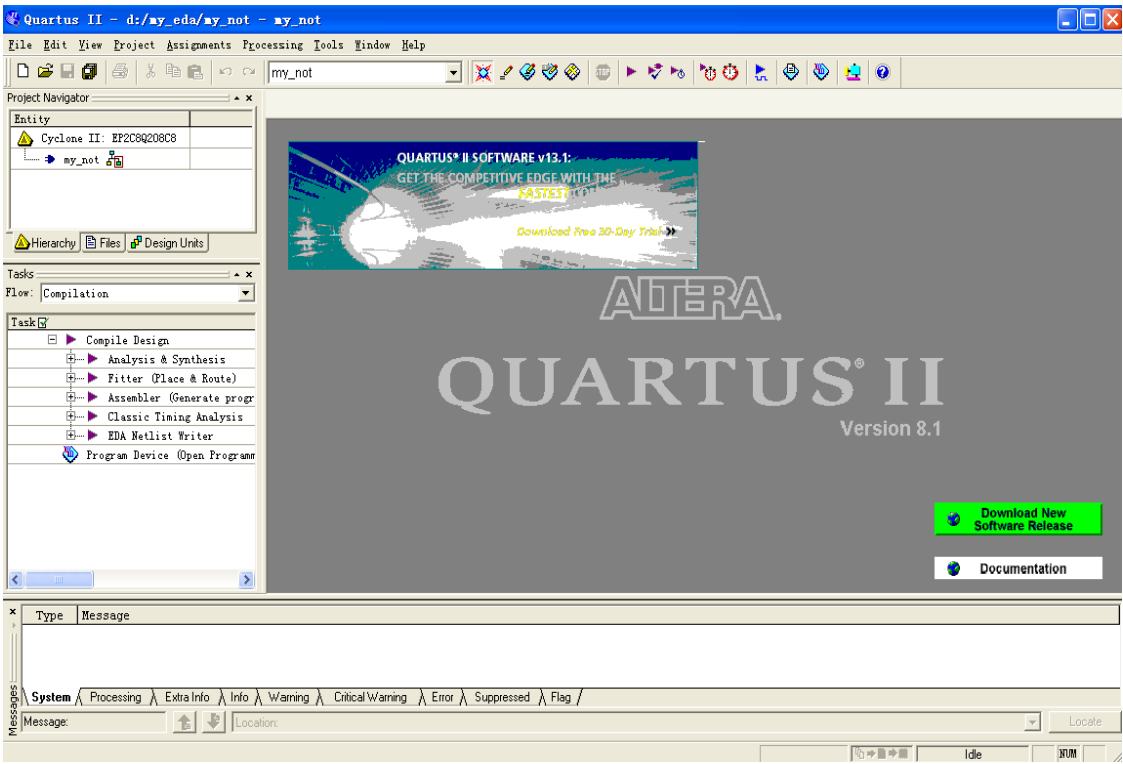


图 2.6 Quartus II 实验设计程序界面

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。

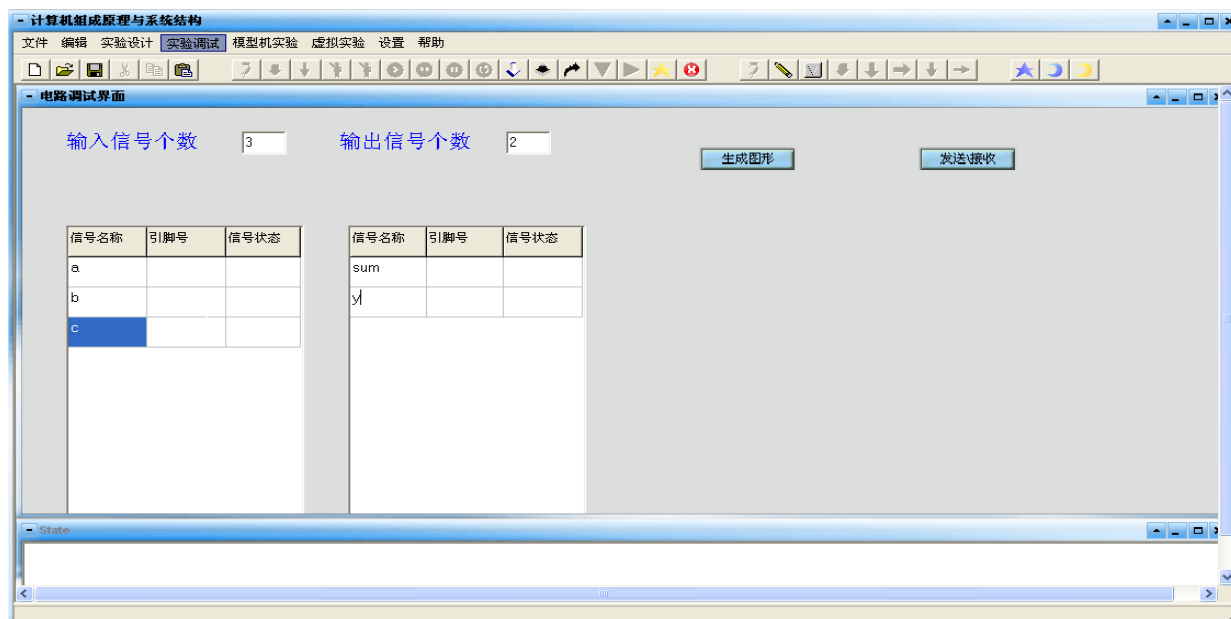


图 2.8 确定信号个数和信号名称

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**”按钮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。

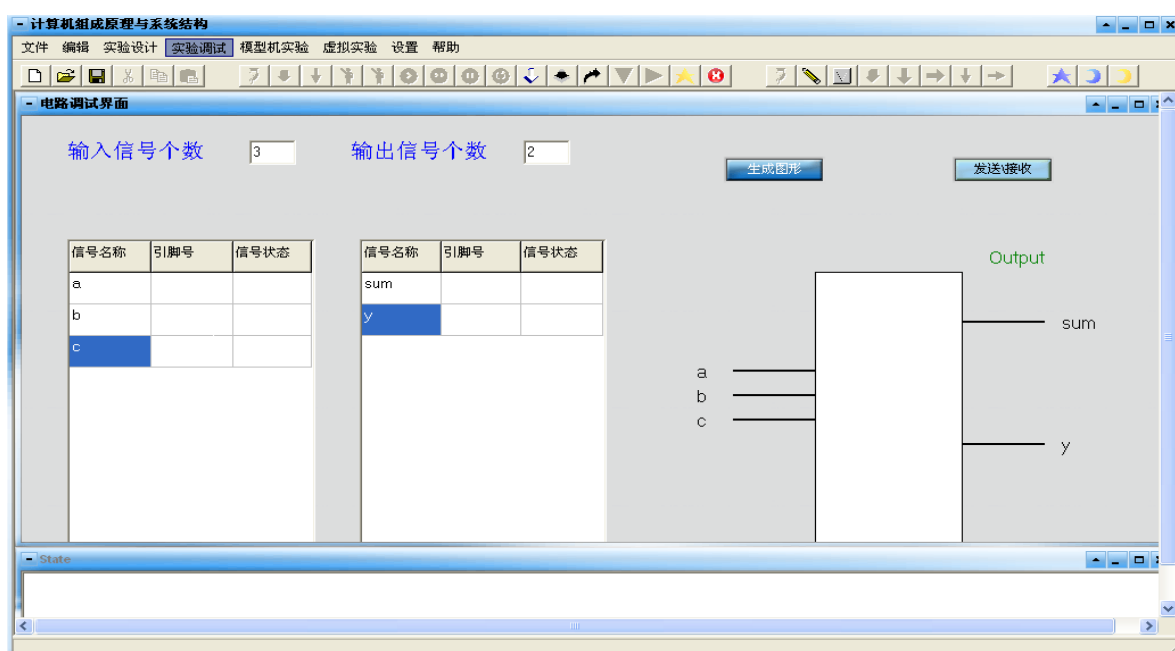


图 2.9 确定信号个数和信号名称



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。

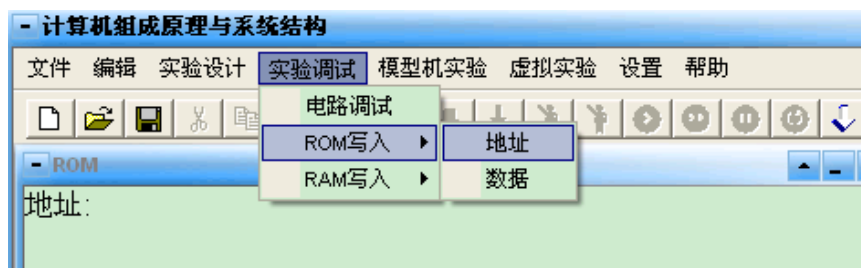


图 2.11 ROM 地址写入界面

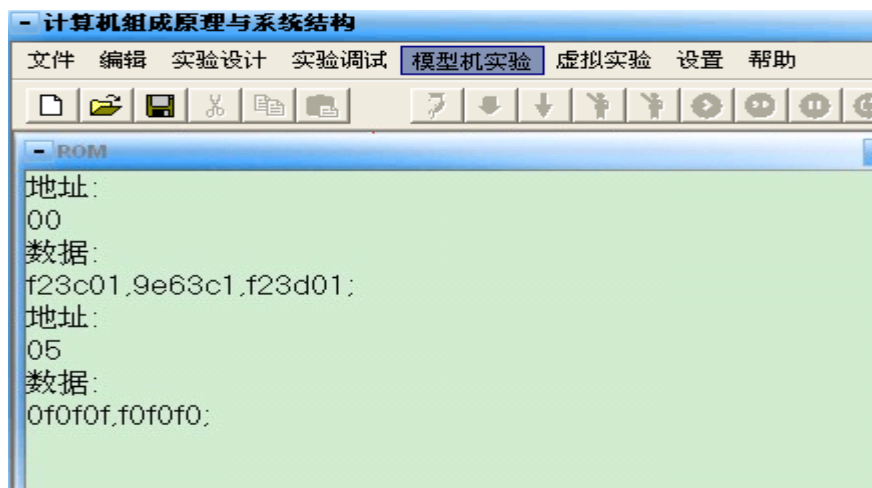


图 2.12 ROM 数据写入界面

◆ 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。



图 2.13 快捷键示意图

2.3.2 远程实验

见远程实验说明书。

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。

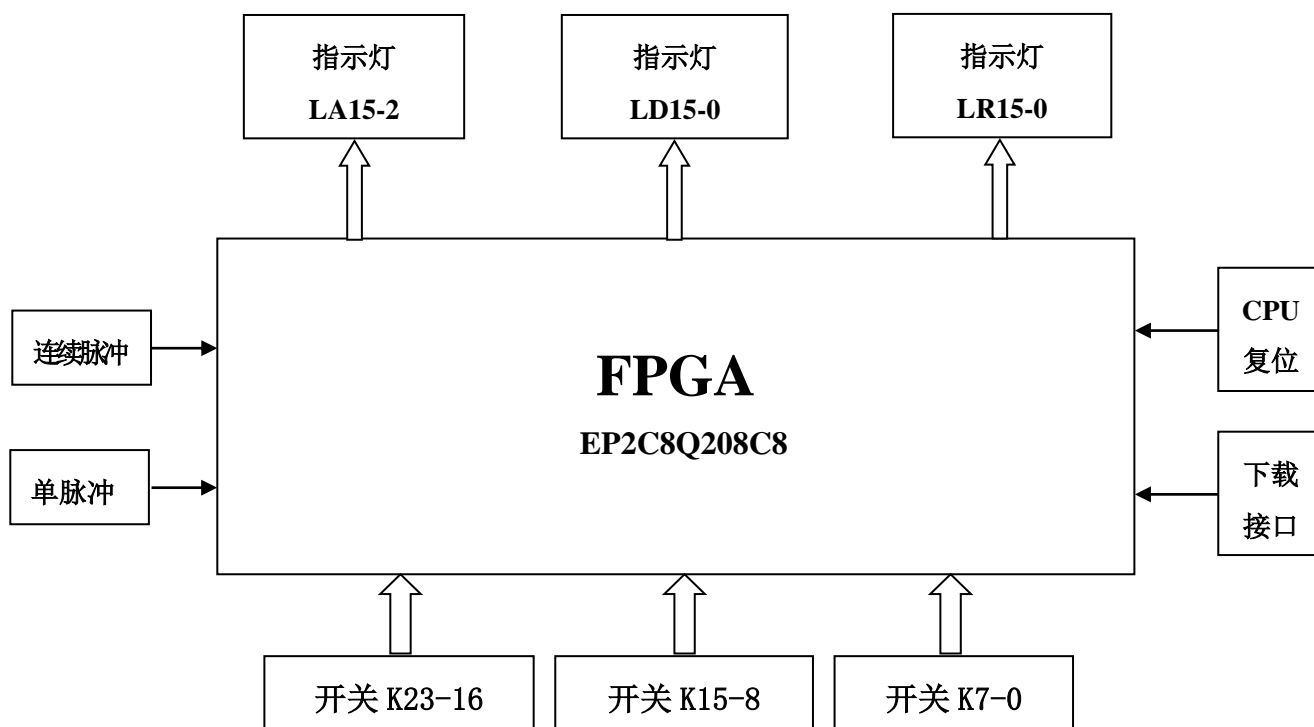


图 3.1 **FPGA** 与输入、输出资源的连接示意图

说明：1. 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；

2. JYS--X 数字实验扩展板（若已选购）的资源及使用方法见附录 3。

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： $A=a_3a_2a_1a_0$ ， $B=b_3b_2b_1b_0$ ， $C=c_3c_2c_1c_0$ ，为三路四位数据输入端， $Y=y_3y_2y_1y_0$ 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。

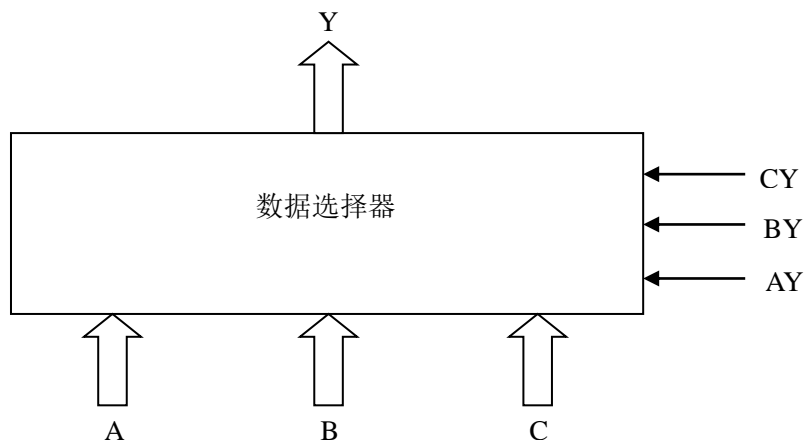


图 3.2 数据选择器框图

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 Quartus II 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。

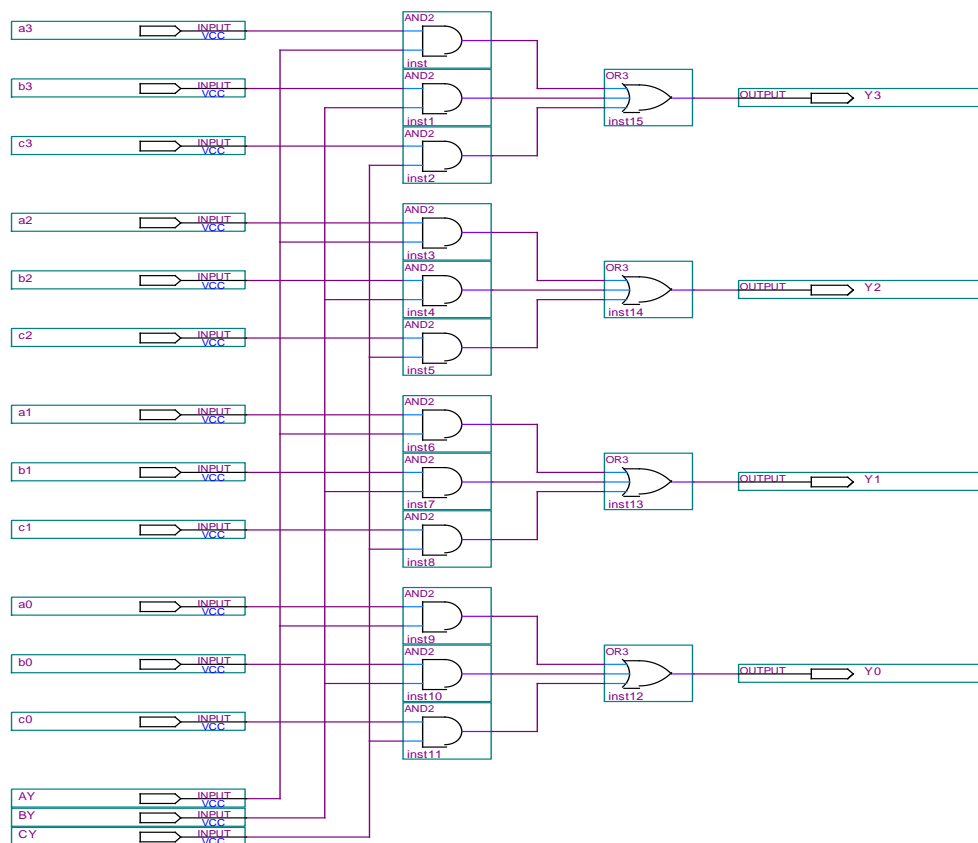


图 3.3 数据选择器电路原理图

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。

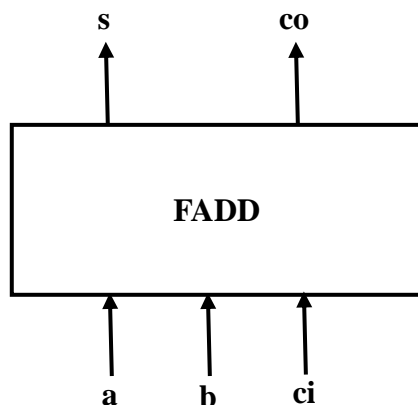


图 3.4 1 位二进制全加器框图

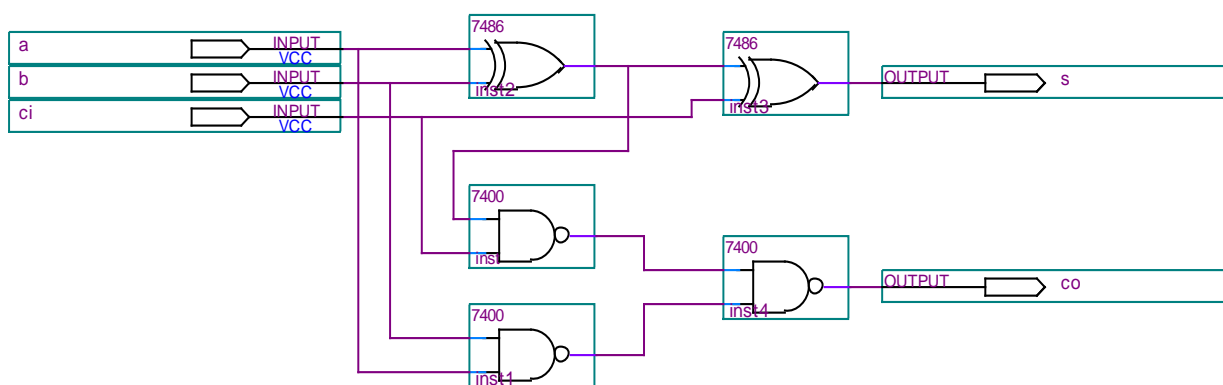


图 3.5 1 位二进制全加器电路原理图

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 Quartus II 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。

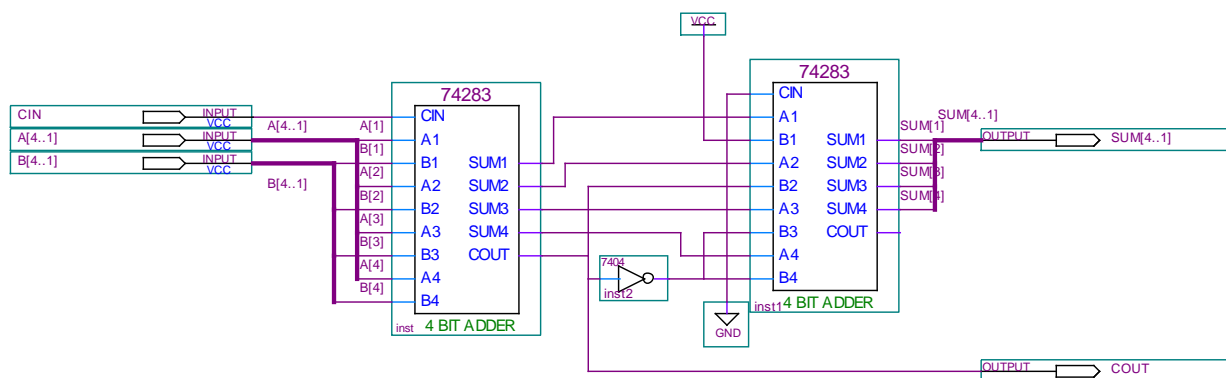


图 3.6 余三码编码的 1 位十进制数加法器原理图

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 Quartus II 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 i_2-i_0 为译码器输入端， y_7-y_0 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。

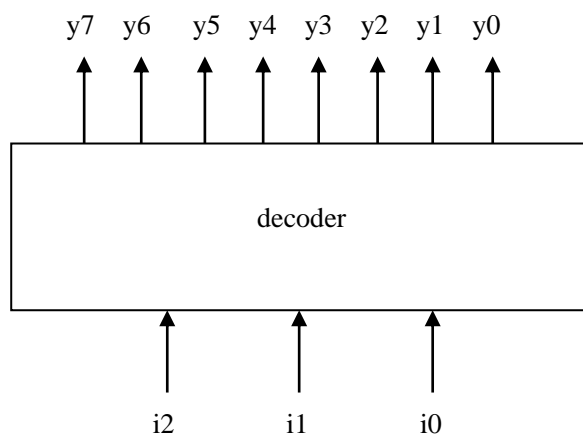


图 3.7 3 线—8 线译码器框图

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 Quartus II 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。

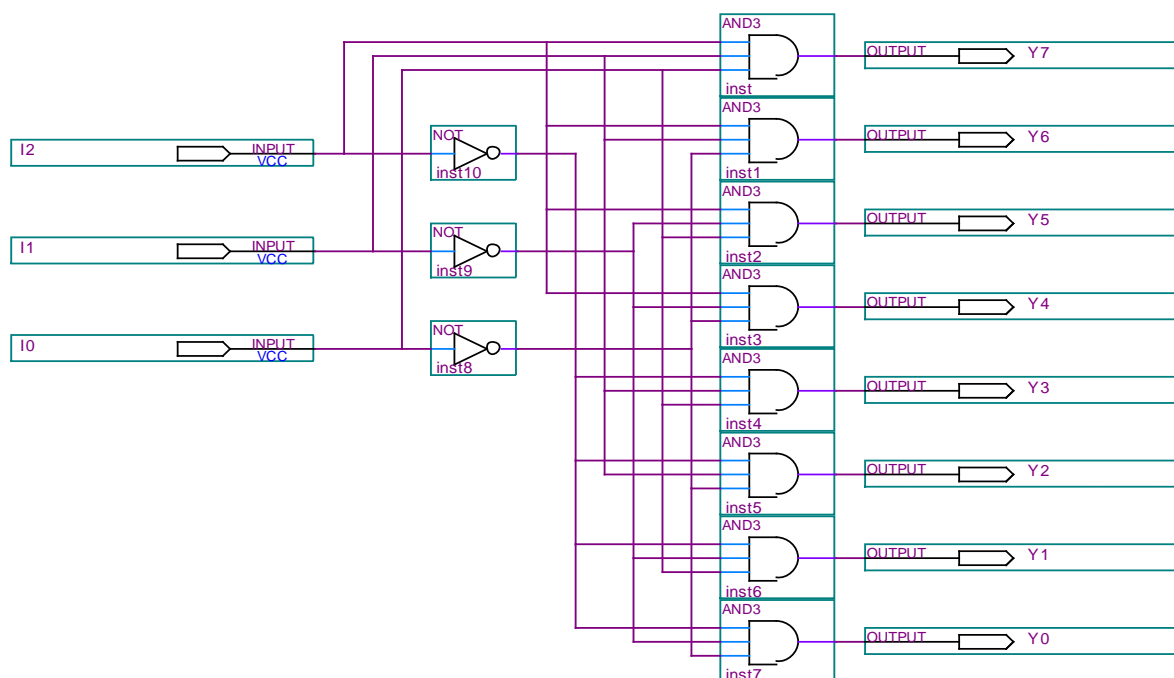


图 3.8 3 线—8 线译码器

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。

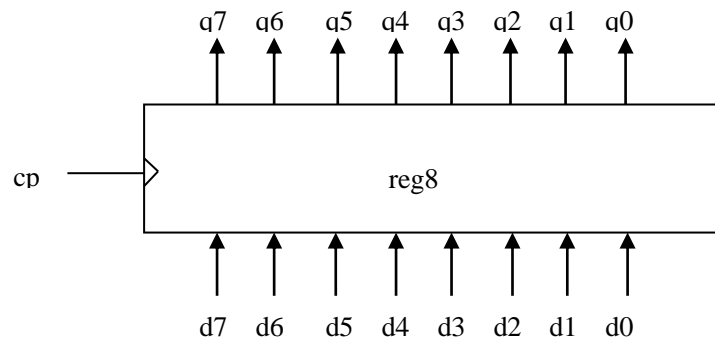


图 3.9 8 位寄存器的框图

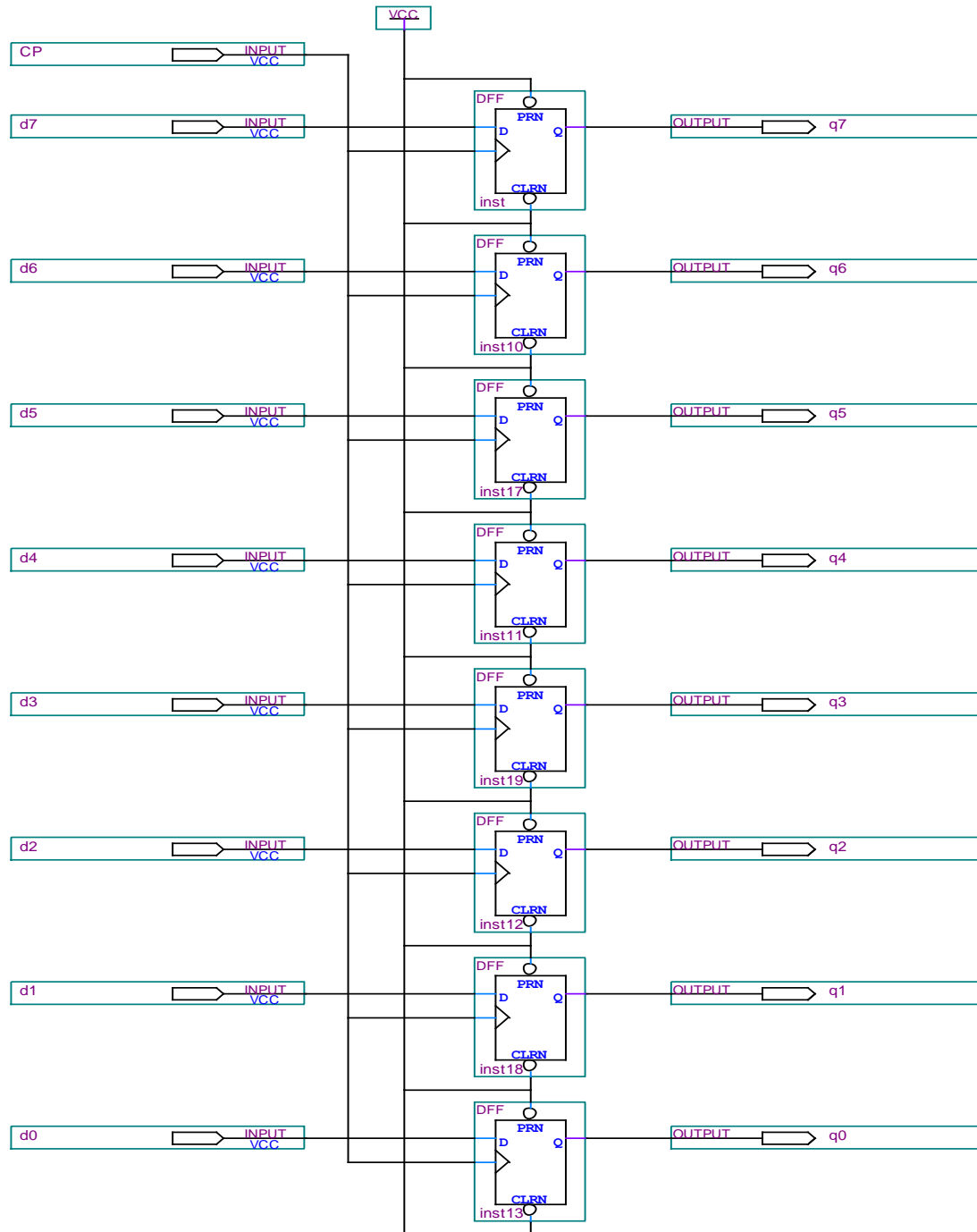


图 3.10 8 位寄存器电路原理图

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 Quartus II 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。



图 3.11 同步模 4 可逆计数器框图

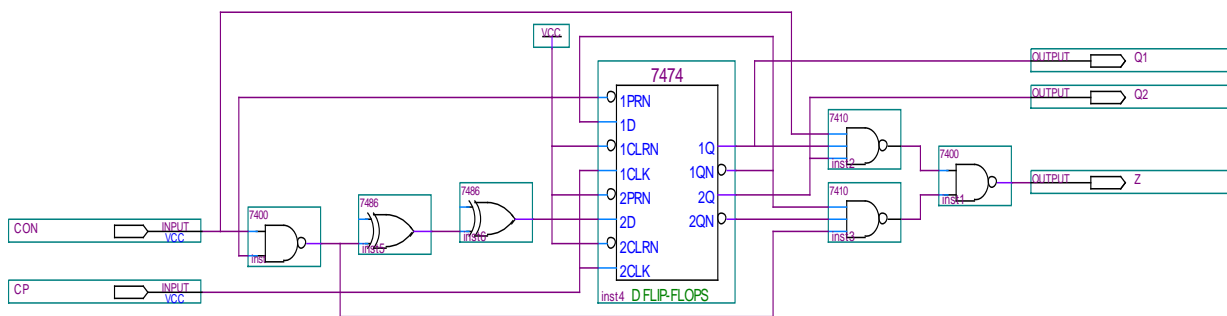


图 3.12 同步模 4 可逆计数器原理图

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 Quartus II 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。

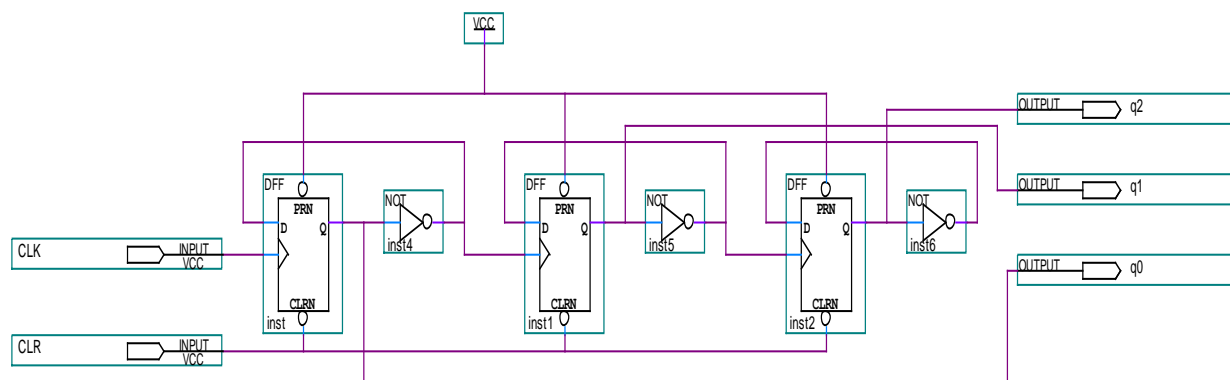


图 3.13 异步模 8 加 1 计数器原理图

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 Quartus II 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。

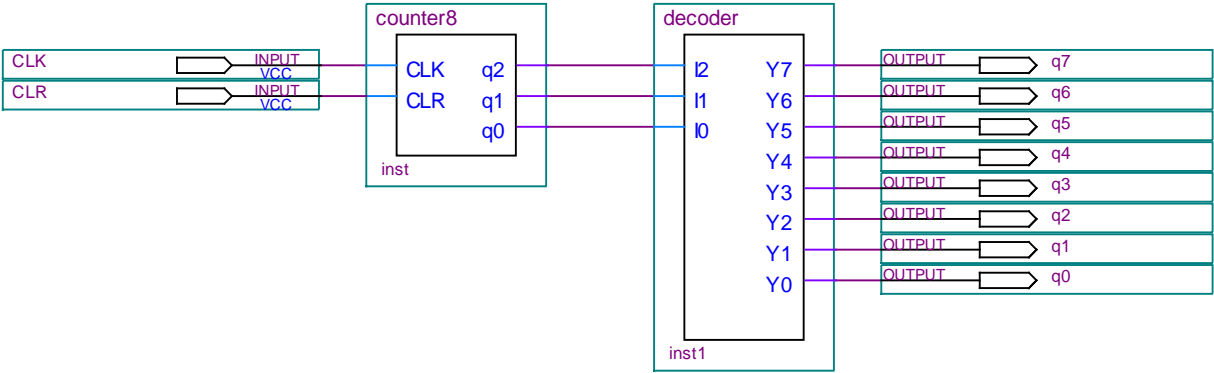


图 3.14 节拍发生器原理图

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 Quartus II 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。

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 |

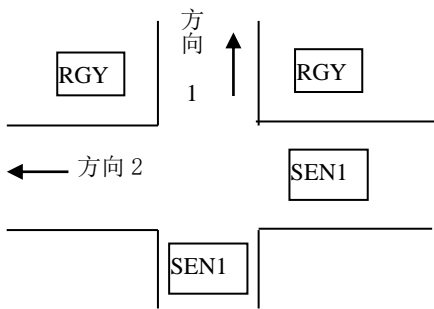


图 3-15 交通情况简图

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 1 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(R_1 , G_1 , Y_1 , R_2 , G_2 , Y_2)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 SEN1 和 SEN2 之一为 1 时，相应方向的绿灯应点亮。
- 2. 当 SEN1 和 SEN2 同时为 1 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 3.1 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	1	0	0	0	1	0
2	1	0	0	0	0	1
3	0	1	0	1	0	0
4	0	0	1	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拨号正确，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。

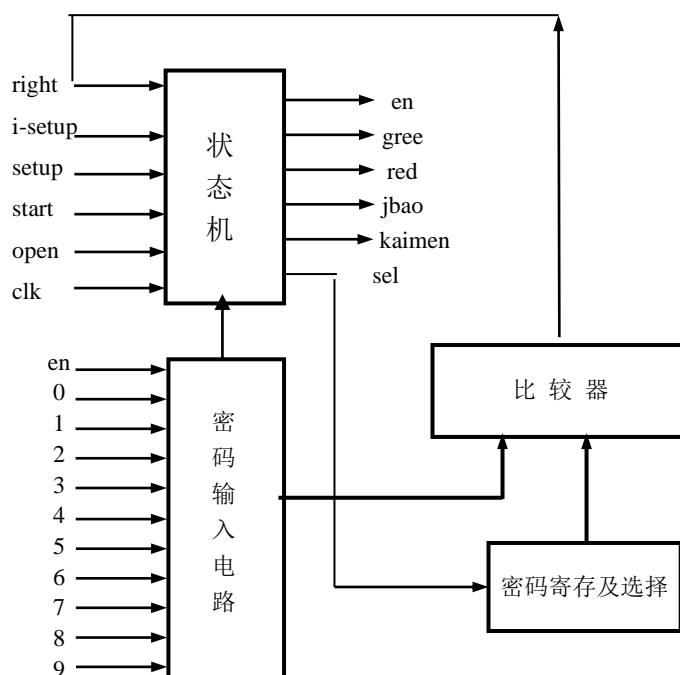


图 3.16 数字密码锁框图

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，按关门按钮电梯门关闭；

(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($a\oplus b$)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 Quartus II 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 $A(a_3a_2a_1a_0)$ 和 $B(b_3b_2b_1b_0)$ 的逻辑运算并生成元件符号。

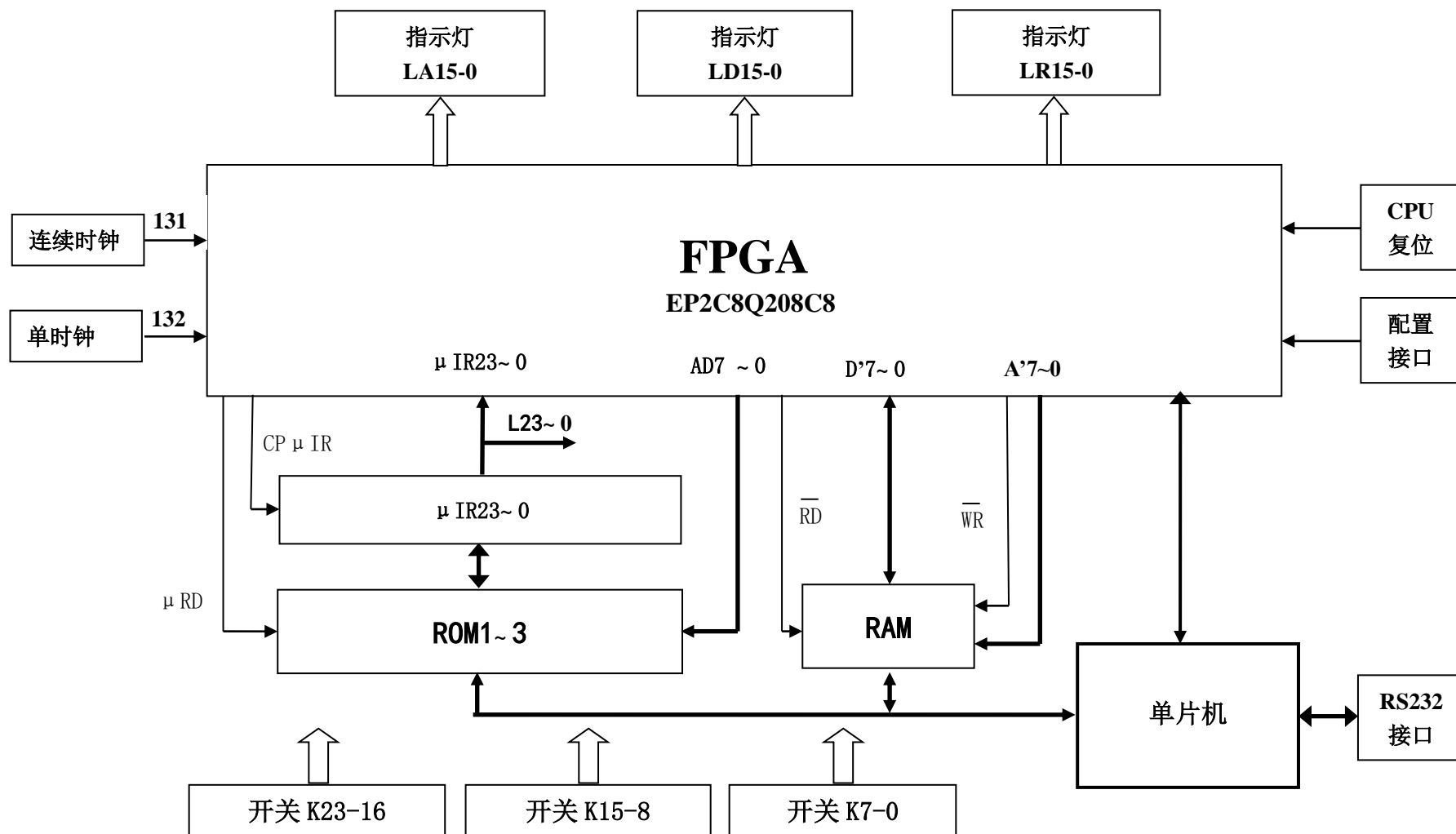


图 4.1 计算机组成原理实验资源示意图

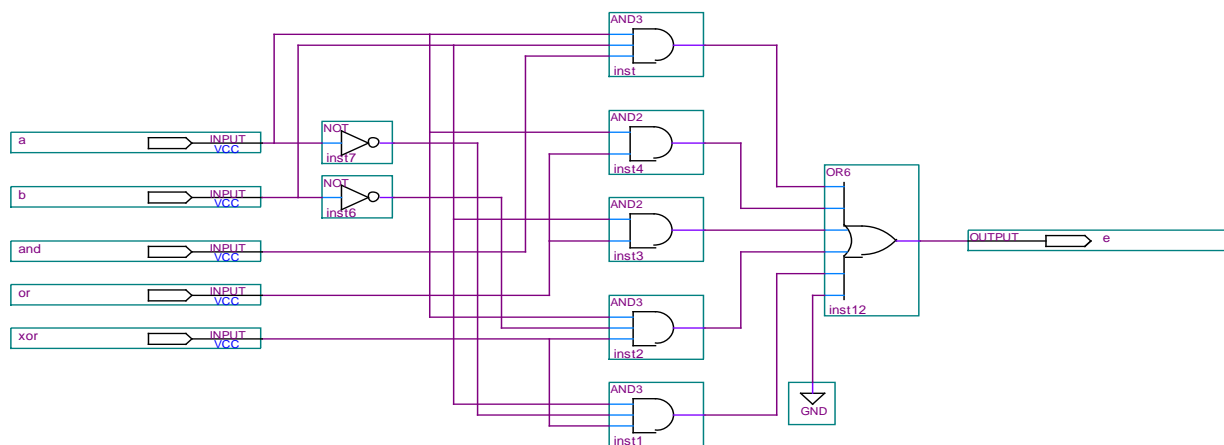


图 4.2 逻辑运算电路原理图

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。

在补码运标中：

$$(X)_{\text{补}} + (Y)_{\text{补}} = (X + Y)_{\text{补}}$$

$$(X)_{\text{补}} - (Y)_{\text{补}} = (X)_{\text{补}} + ((Y)_{\text{补}})_{\text{补}}$$

其中 $((Y)_{\text{补}})_{\text{补}}$ 是将 $(Y)_{\text{补}}$ 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 Quartus II 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。

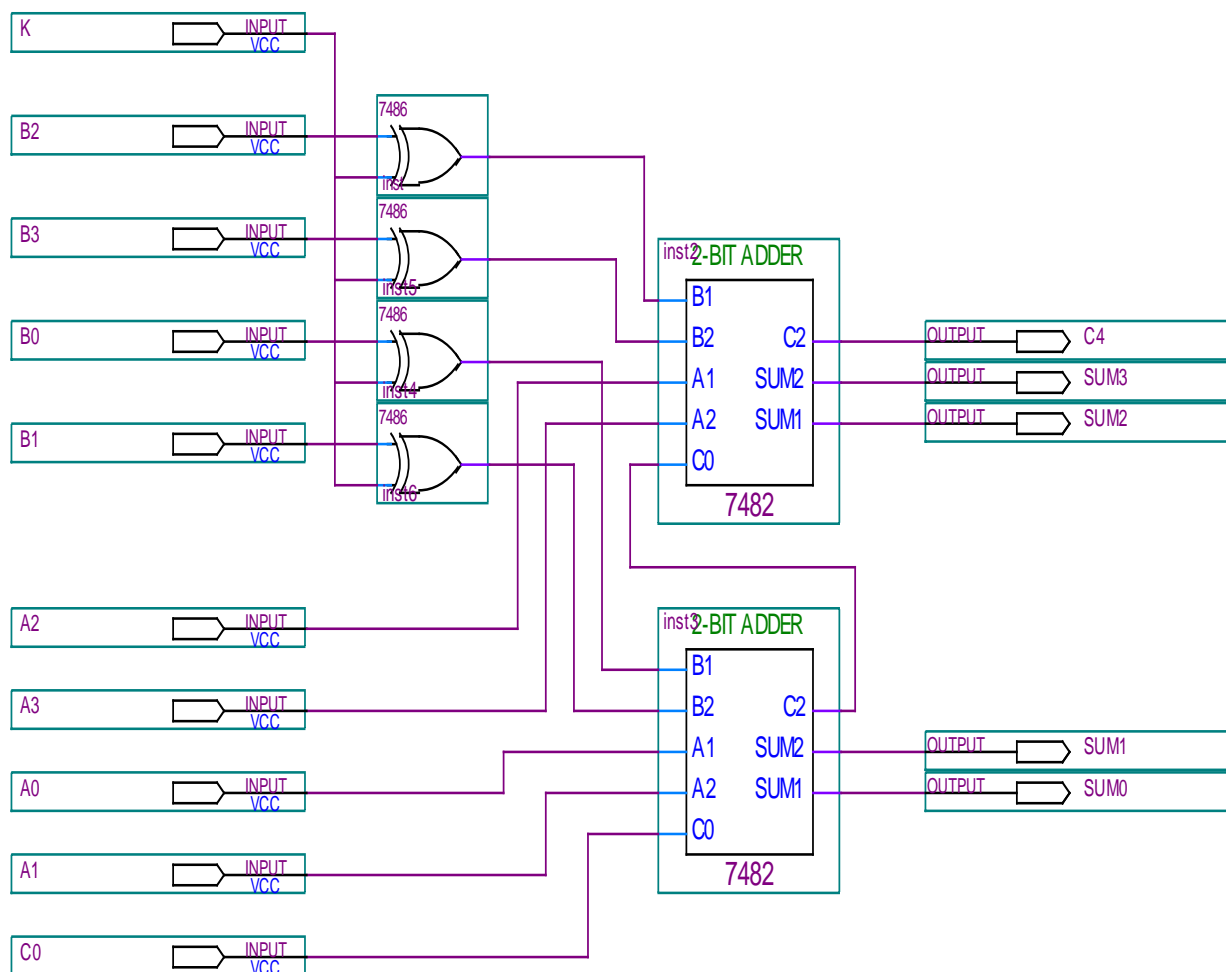


图 4.3 四位补码加法器原理图

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 ($\times 2$), 右移 1 位 ($\div 2$) 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。

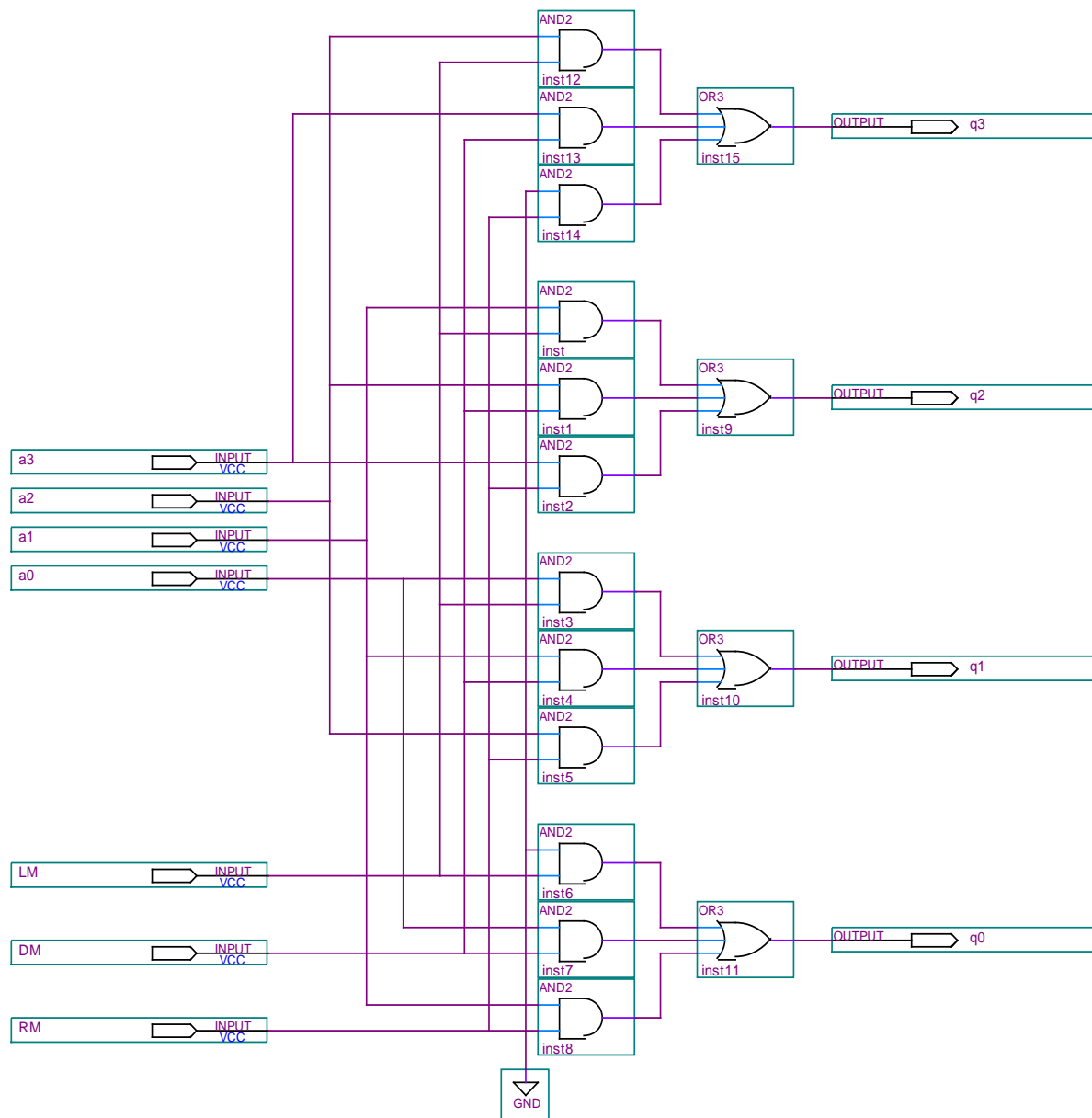


图 4.4 移位器电路原理图

实验 4 八位串/并进位运算器

1、实验内容及说明

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。

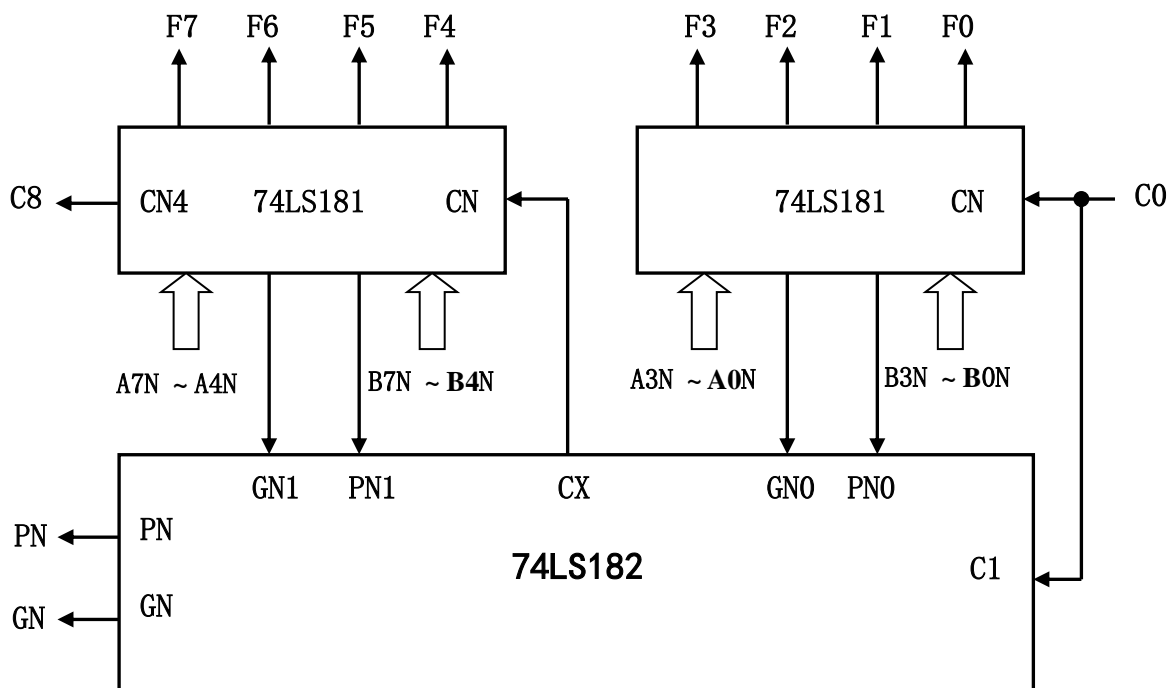


图 4.5 八位组间并行进位运算器原理图

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 Quartus II 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），移位器可调本章实验 3 中设计的移位器元件，四位加法器可调本章实验 2 中设计的补码加法器元件。

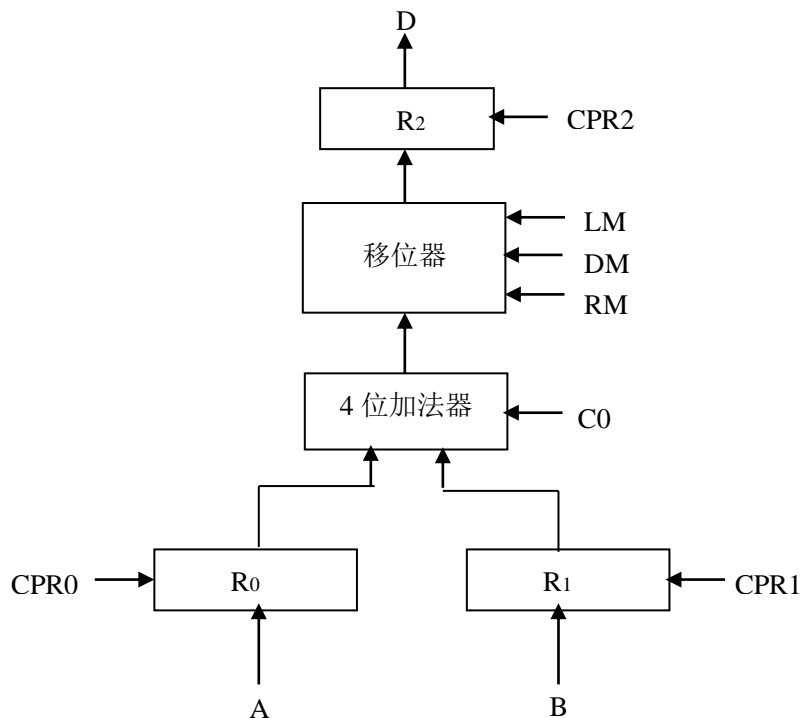


图 4.6 四位补码运算器电路框图

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 由低电平拨向高电平再拨向低电平，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 Quartus II 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 μPC 计数器和微指令寄存器 μIR 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 μPC 向控制存储器提供 8 位微地址，在控存读信号 $\overline{\mu RD}$ 的作用下，读出一条长 24 位的微指令代码，并在打入命令 $CP \mu IR$ 的作用下，送入 μIR_{23-16} 、 μIR_{15-8} 、 μIR_{7-0} 。

L_{23-16} 、 L_{15-8} 、 L_{7-0} 用于显示微指令寄存器 μIR_{23-16} 、 μIR_{15-8} 、 μIR_{7-0} 的内容。

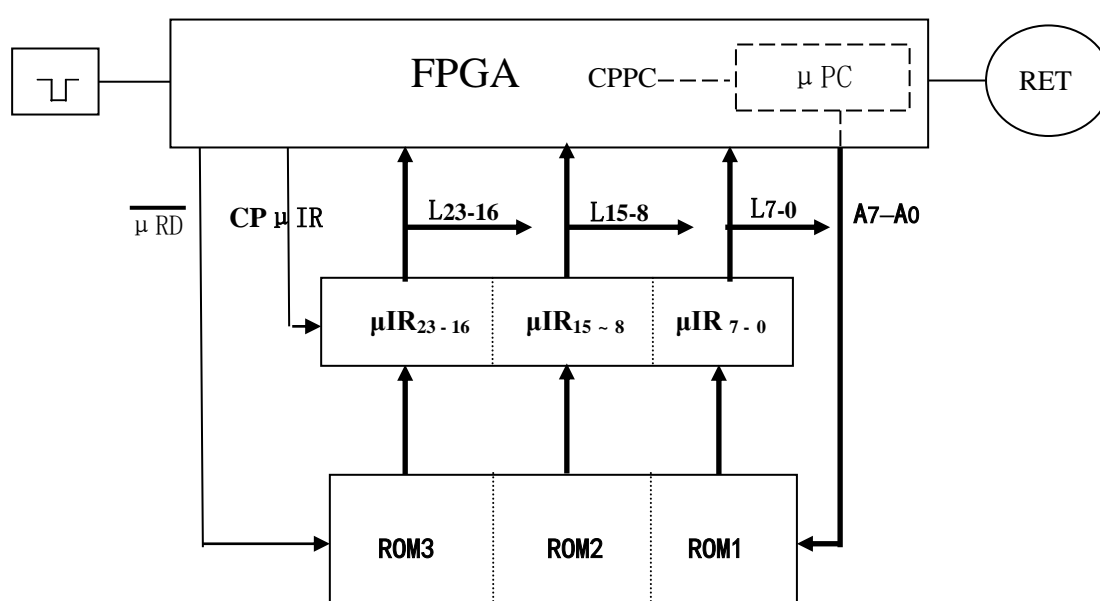


图 4.7 微程序控制器框图

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 $\overline{\mu RD}$ 。
- 负脉冲当作 $CP \mu IR$ 将读出的微指令打入微指令寄存器 μIR 。
- 负脉冲的上升沿使 $\mu PC + 1$ 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。

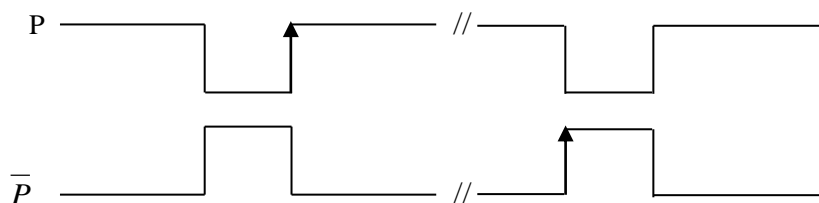


图 4.8 微程序时序

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 、CP μIR 。
- * 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 $\mu IR23-\mu IR0$ 同时显示在 L23-L0 上；

将微指令计数器 $\mu PC + 1$ 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 $\mu IR23-\mu IR20$ 上，CPR0、CPR1、CPR2 依次锁定在 $\mu IR7-\mu IR5$ 上，LM、DM、RM、C0 依次锁定在 $\mu IR4-\mu IR1$ 上，P 锁定在单脉冲按键上。Q3-Q0 依次锁定在 LD3-LD0 上。

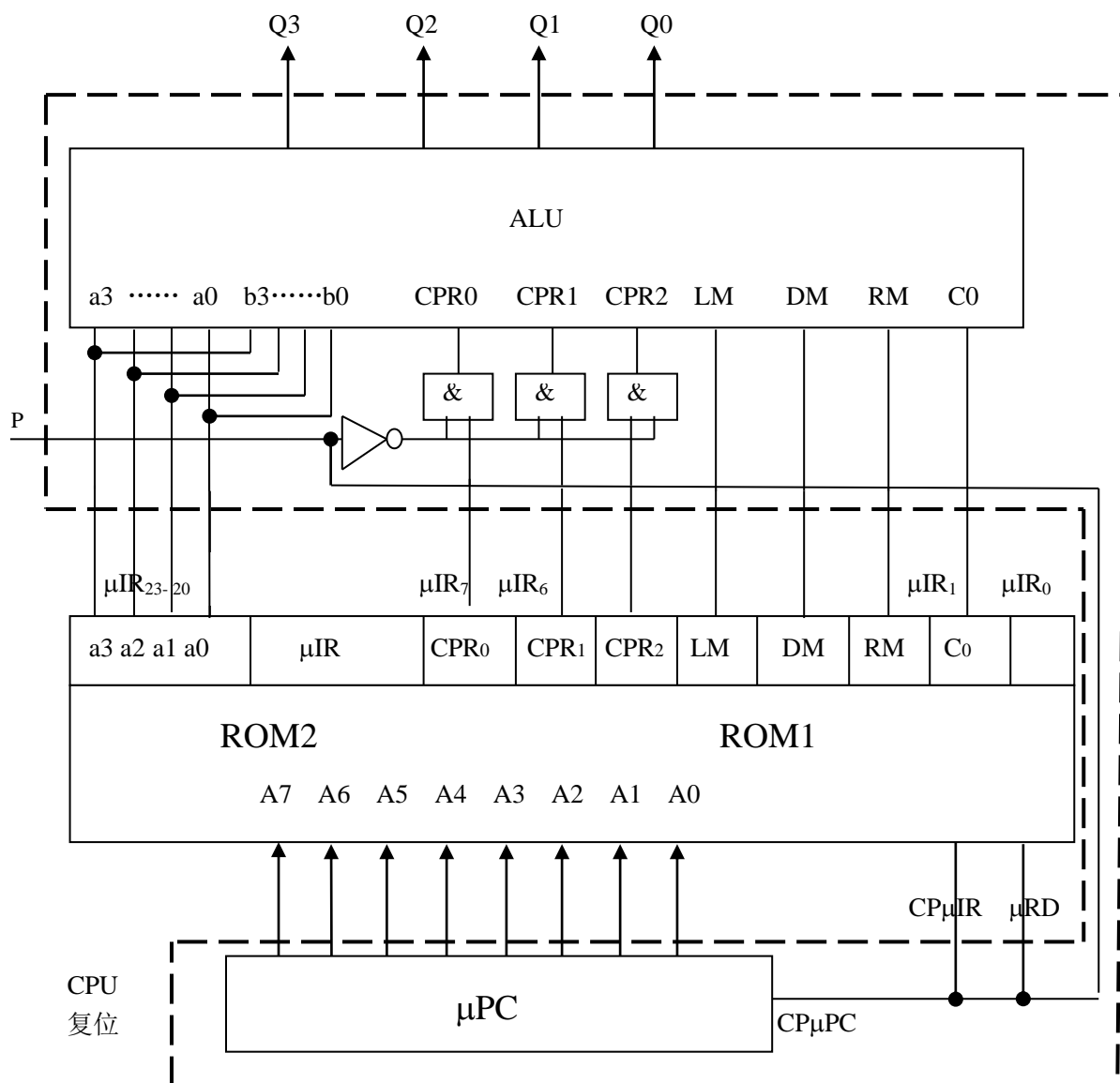
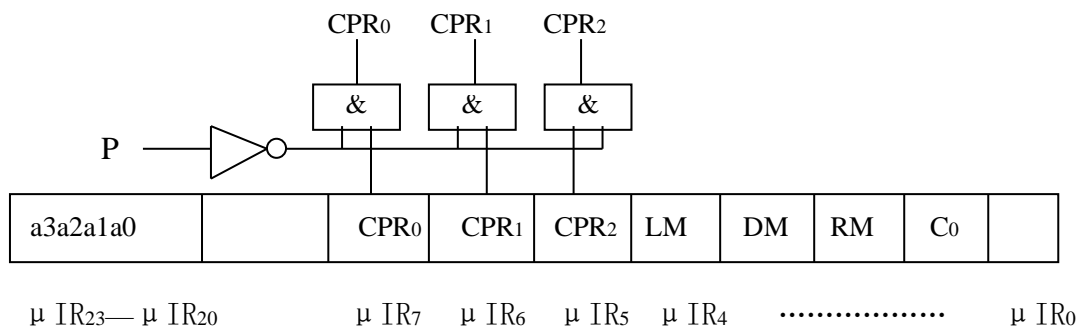


图 4.9 CPU 综合实验结构框图

(3) 适配、下载

(4) 编制微程序

微指令可确定如下格式：



将微指令格式分为两部分：前面部分 $\mu\text{IR}_{23} \sim \mu\text{IR}_{20}$ 可设置数据，后面部分 $\mu\text{IR}_7 \sim \mu\text{IR}_0$ 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 μ PC，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 $2 \times (R0 + R1) \Rightarrow R2;$

$1/2 (R0 + R1) \Rightarrow R2;$

.....

实验 8 模型机演示

1、演示内容及说明

本实验可以完成模型机数据通路的数据传送过程演示，通过演示使学生清晰地看到控制命令和信息流向的关系，不同的数据通路需要不同的控制命令。

2、演示步骤：

(1) 点击桌面的“计算机组成原理与系统结构”图标进入模型机实验。

(2) 编写汇编程序：

◆ 在 Code 窗口中输入汇编指令，如图 4.10 所示。完成所有指令输入后，点击按键“编译”快捷键完成汇编指令的编译，之后在 Micro OP 窗口中即可得到相应的指令流程，如图 4.11 所示。

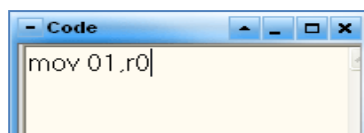
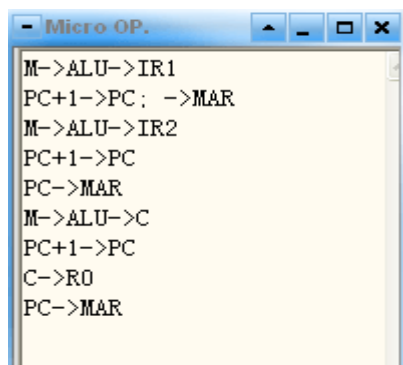


图 4.10 汇编指令输入窗口



4.11 编译后的微指令流程窗口

◆ 程序执行：点击“自动演示（单步）”快捷键即可执行一条微指令，执行过程中右侧的 Diagram 窗口会显示相应的微指令执行路径及控制命令提示，如图 4.12 所示。也可以点击“自动演示（连续）”快捷键完成自动演示功能。

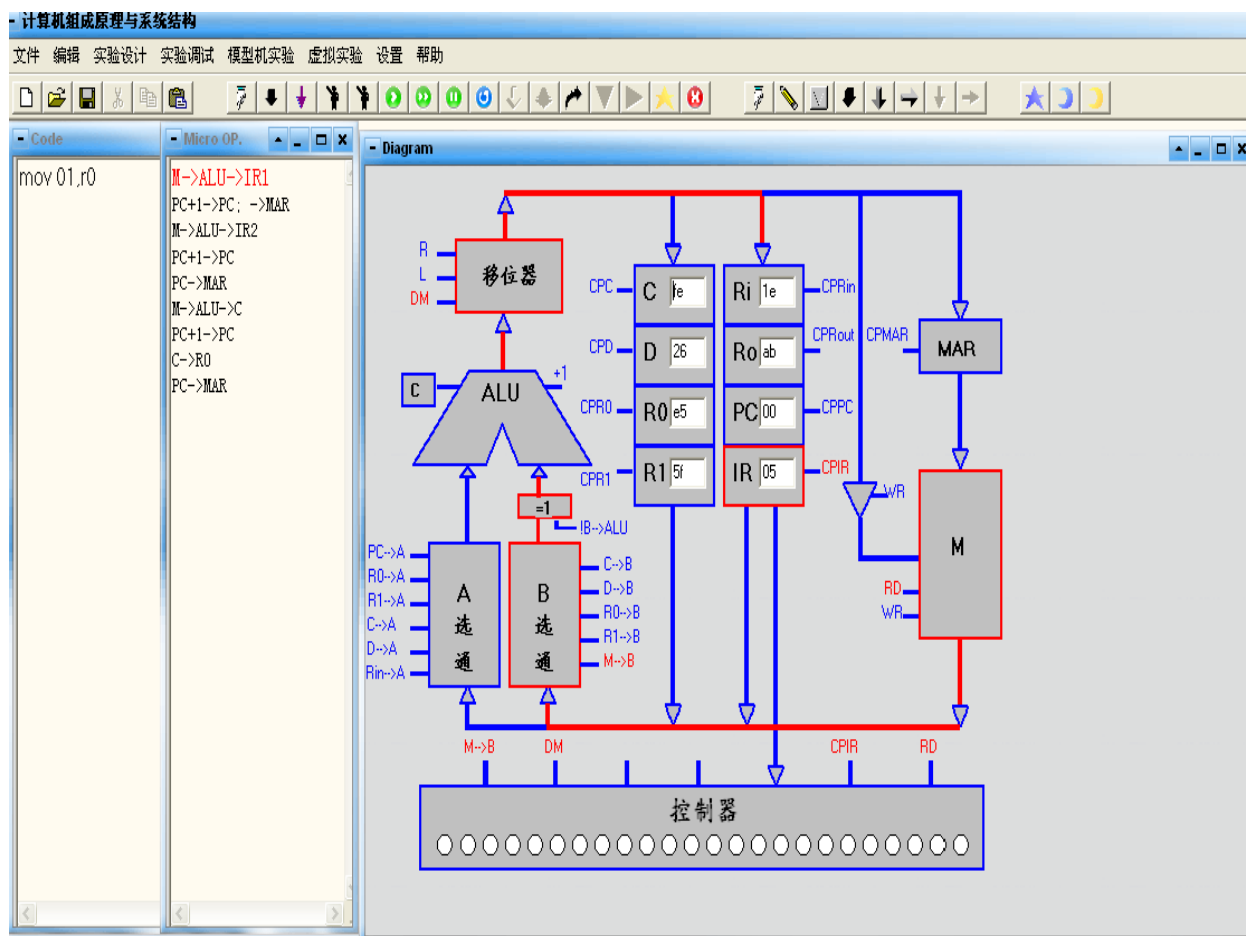


图 4.12 微指令执行路径及控制命令窗口

第 5 章 模型机课程设计

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。下面简要介绍模型实验系统。

5.1 模型机系统结构

5.1.1 模型机指令系统

1.模型机指令系统

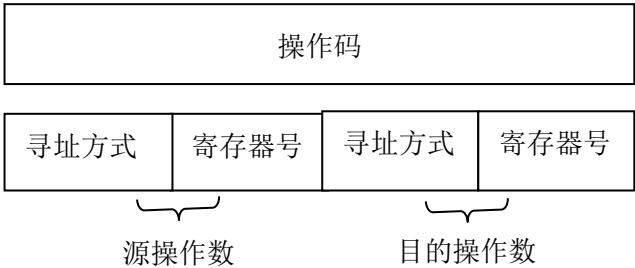
模型机指令系统比较简单，设置双操作数指令和单操作数指令，如表 5.1 所示。

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

2、指令格式

采用双字节指令，字长 16 位



寄存器设置：

R_0 }
 R_1 } 通用寄存器

C：存放从内存取出的源操数

D：存放从内存取出的目的数

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。

5.1.2 模型机总体结构

模型机总体结构如图 5.1 所示。

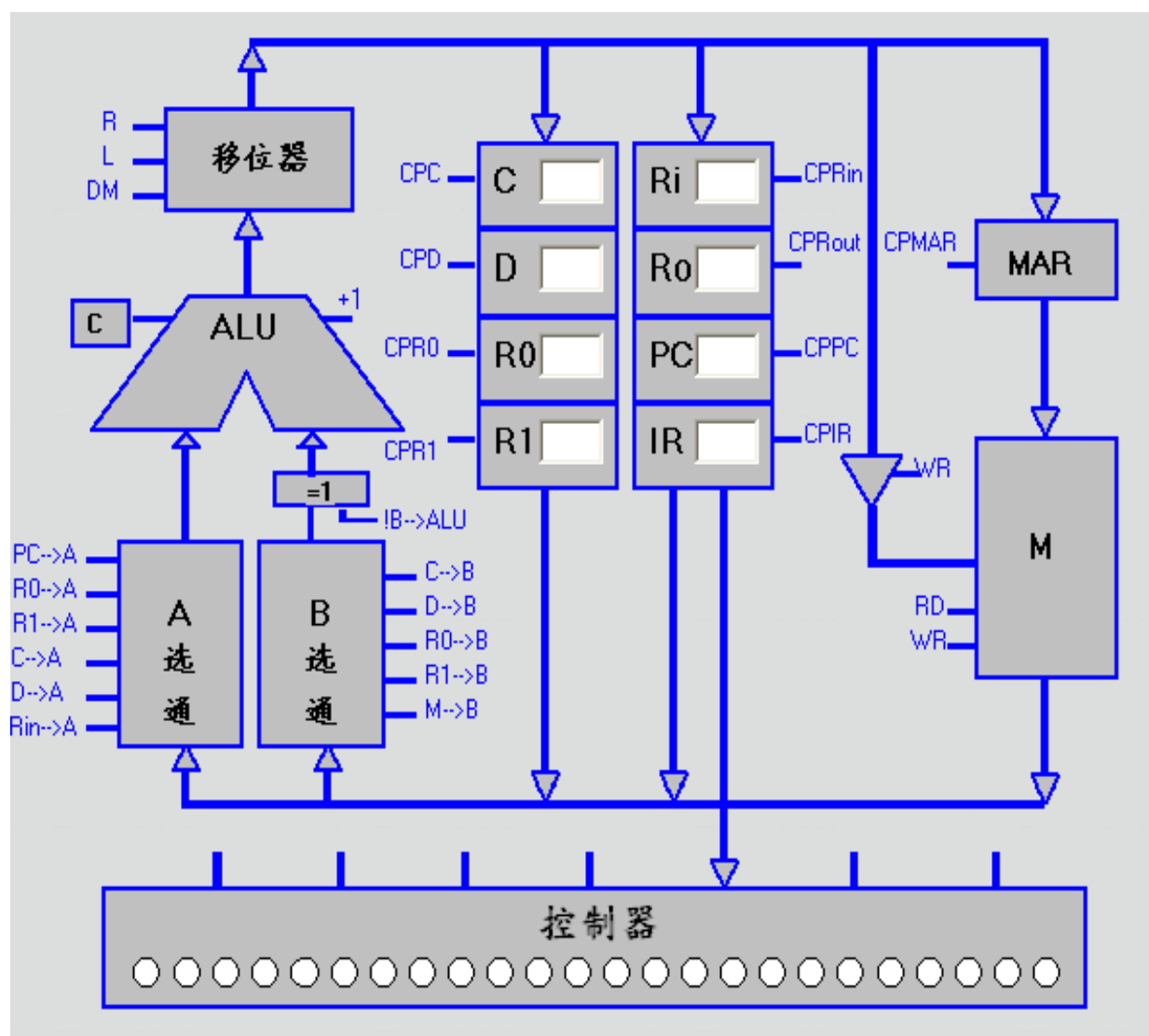


图 5.1 组成原理模型机结构图

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。

3、采用总线结构

如图 5.1 所示：

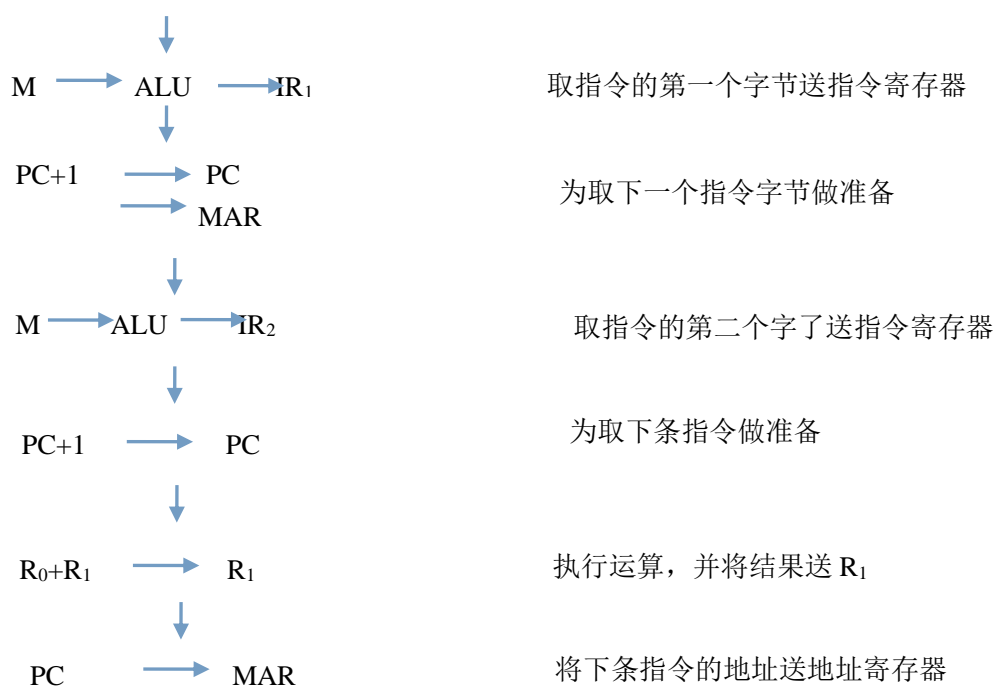
A 各寄存器及存储器接收移位器经总线送来的数据。

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。

5.1.3 指令流程

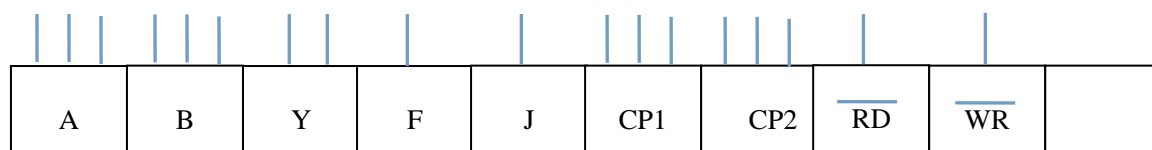
指令执行流程亦称指令执行过程，大都由四个阶段构成，即取指令，取源操作数，取目的操作数，执行运算。但不同的指令不同的寻址方式，指令流程也不尽相同，需要的数据通路也不一样，因此需要的数据通路也不一样，当然控制命令也不一样，下面以 $ADDR_0, R_1$ 为例的执行流程。



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。

5.1.4 微指令格式

1、模型机微指令字长 24 位，其格式如下：



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 \overline{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)	\overline{RD} (1 位)	\overline{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 模型机数据通路设计

1、设计目的：

通过该课程设计可使学生清楚地掌握计算机的基本结构、各部件的基本功能、部件间的连接方式及信息如何传送。

2、设计说明

(1) 以图 5.1 所示模型机数据通路结构为依据进行设计，其中数据通路宽度为 8 位，因此所有寄存器均为 8 位，移位器、ALU、选择器等均为 8 位，数据通路中，数据传送方向由箭头指示，所需的控制命令如 PC → A、R0 → B 等，寄存器打入脉冲如 CPR0 等均由微指令提供，存储器借用平台中的 RAM；

(2) 数据通路设计时所需元件可从元件库中调用。

3、实验要求

- (1) 画出数据通路设计图
- (2) 写出调试步骤
- (3) 制定测试方案
- (4) 分析该设计的优缺点
- (5) 提出改进方案

5.3 简单计算机的设计

1、设计目的

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。

2、设计说明

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。

3、设计要求

- (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) 评价该设计的优缺点；
- (5) 提出改进建议。

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	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	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	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

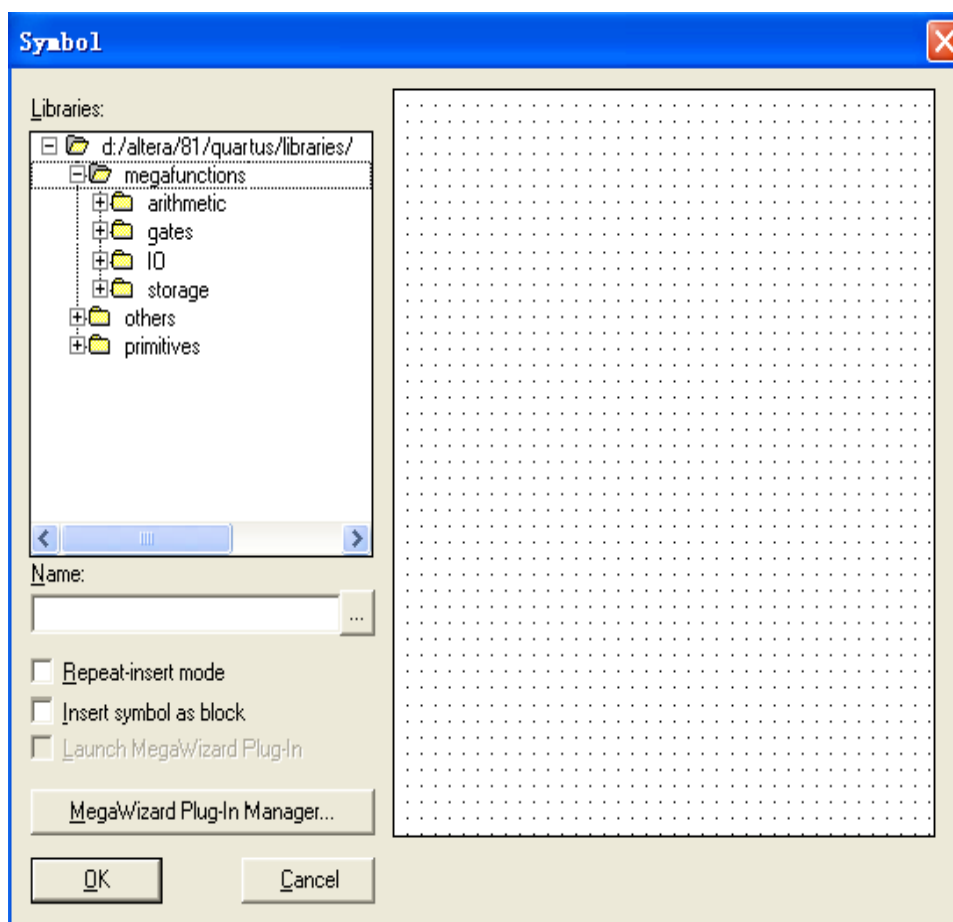
信号名	$\overline{\mu RD}$	CP μ IR	\overline{rd}	\overline{wr}	连续脉冲	单脉冲	CPU 复位
引脚号	60	61	63	64	131	132	206

表中： $\overline{\mu RD}$ 为 ROM 读信号、CP μ IR 为微指令寄存器时钟信号、 \overline{rd} 为 RAM 读信号、 \overline{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 Quartus II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。

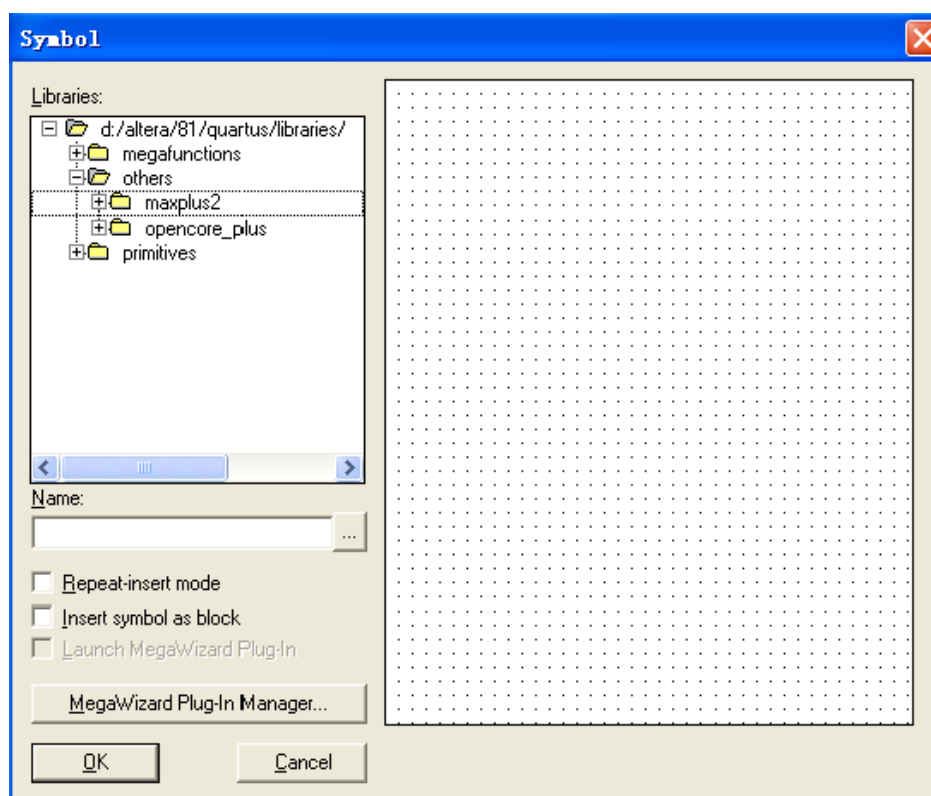
宏功能模块 (megafunctions)库如图附2.1所示，宏功能模块式参数化的模块，模块的各个参数由设计者为满足设计要求自行定制，只要修改模块参数，就可以得到满足需要的特定模块。宏功能模块包含算术运算类模块、门单元类模块、输入/输出类模块和存储器类模块。



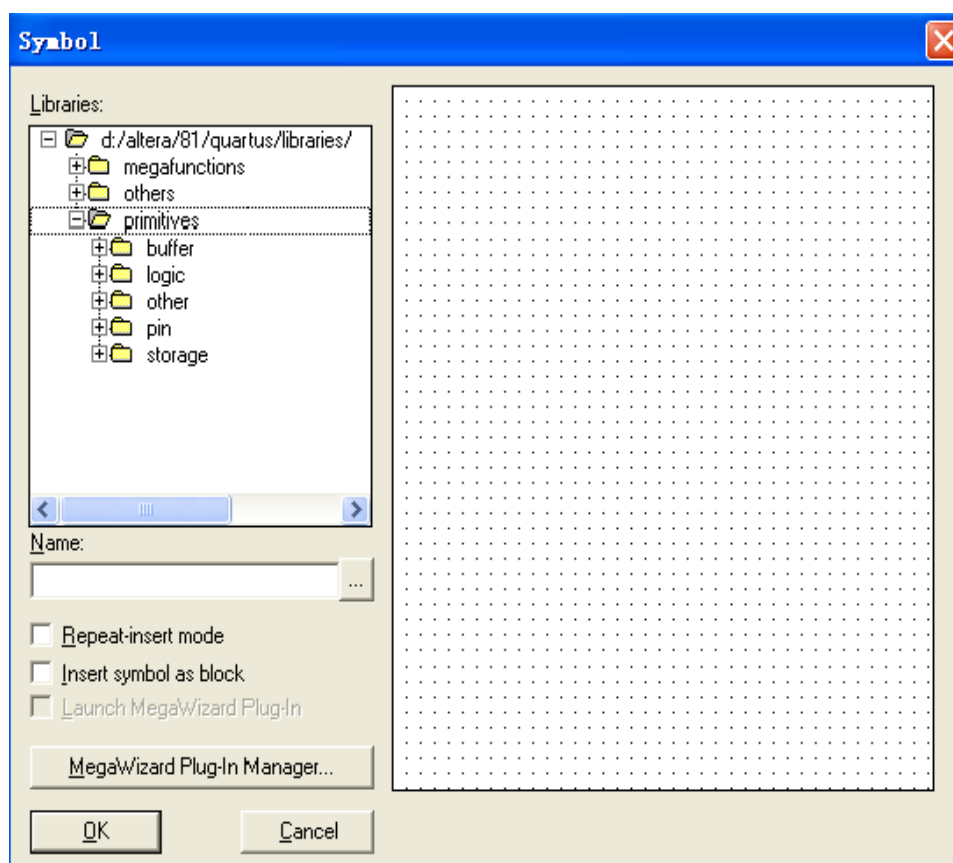
图附2.1 宏功能函数(megafunctions)库

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。

附录3 JYS-X 数字实验扩展板使用说明

JYS--X 数字实验扩展板共分六个区域（I 区 ~ VI 区），如图 2 所示。

I 区：高低电平输出。当拨动开关（K1 ~ K20）时可由 J7 ~ J9 接线插座输出高低电平；开关拨向上方时为高电平输出（发光管亮），拨向下方时为低电平输出（发光管灭）。

II 区：信号输出区。包括 GND、P+ P-、单脉冲、连续脉冲、VCC 和连续可调模拟电压等，并分别有 J1~J6 接线插座输出。调整 R_w 可从 J6 接线插座输出 0V ~ 4V 的连续模拟电压，P+ P-是由 KA 按键产生的正、负脉冲，单脉冲、连续脉冲来自主机。

III 区：工作区域。该区包括大、小面包板各四块，供设计电路和电路接线使用。其中四块大面包板板竖向的 5 个孔相连，四块小面包板横向共有二排插孔，每排从中间字符处分为二组，共分四组，每组的插孔相连，如图 1 所示。

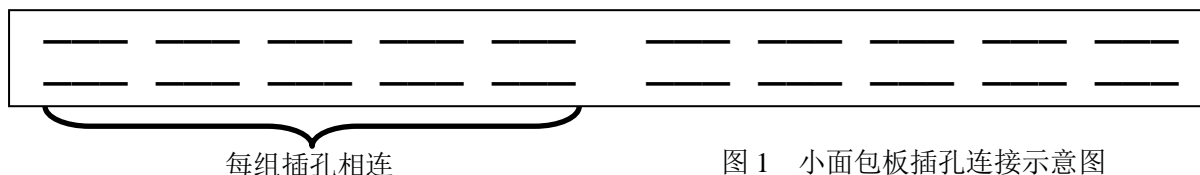


图 1 小面包板插孔连接示意图

IV 区：发光二极管显示。该区共有 16 个发光二极管，欲显示信号由 J13 和 J14 接线插座输入。

V 区：数码管输出显示。该区共有四个带有 4 位二进制码译码的显示数码管，显示数字由 J11 和 J12 二个接线插座输入，4 位二进制数的输入方式为 DCBA（高位在前），可显示 0 ~ 9 十个数字（10 ~ 15 以相应字形显示）。

VI 区：电源和主机连接插座。该区包括外接电源插座、电源开关、J10 接线插座和 J15 主机连接插座等。其中 J15 插座通过 16 线扁平电缆和组成原理实验主机相连，J15 主机连接插座的引脚功能以及和 J10 接线插座的连接关系如表 1 所示。

注意：扩展板只有在未与主机连接时，才可使用外接电源（5V/1A 稳压电源）。

表1 J15插座引脚功能及和J10插座引脚连接关系

J10 引脚号	1	3	5	7	—	—	—	—
J15 引脚号	1	3	5	7	9	11	13	15
功能	161	160	152	151	+5V	连续脉冲	单脉冲	GND
	EP2C8Q208 管脚号				连接到 II 区相应接线座			
J10 引脚号	2	4	6	8	—	—	—	—
J15 引脚号	2	4	6	8	10	12	14	16
功能	145	144	143	141	+5V	连续脉冲	单脉冲	GND
	EP2C8Q208 管脚号				连接到 II 区相应接线座			

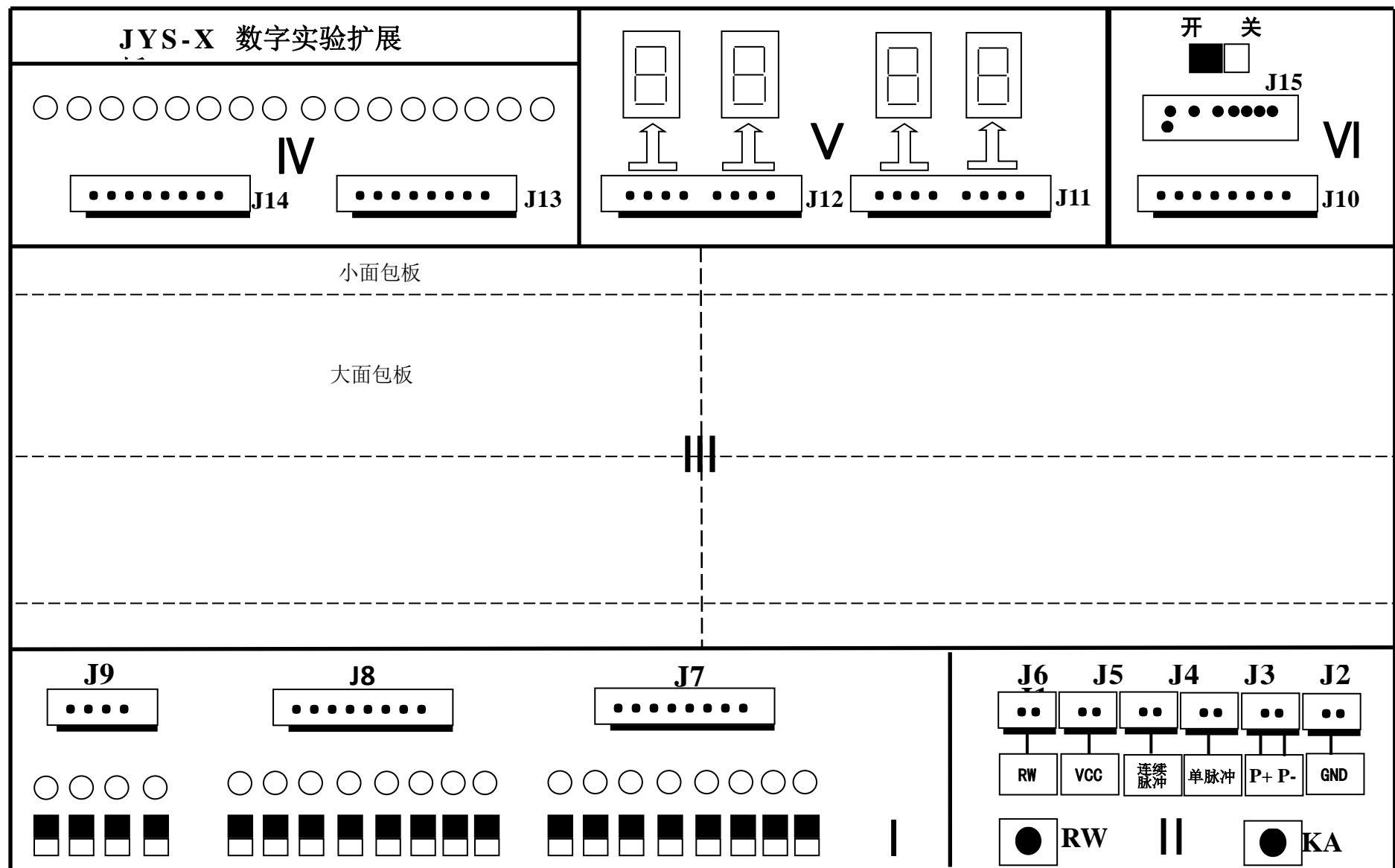


图2 JYS-X数字实验扩展板结构图