



普通高等教育“十一五”国家级规划教材

教育部“高等学校教学质量与教学改革工程”立项项目

张亮 曹庆华 傅翠娇 编著

计算机组成原理 实验教程

计算机科学与技术专业实践系列教材

清华大学出版社

013023635

TP303-33
12

计算机科学与技术专业实践系列教材



计算机组成原理 实验教程

张亮 曹庆华 傅翠娇 编著

TP303-33
12

清华大学出版社



北航

C1630547

013008010

内 容 简 介

本教材紧密结合“计算机组成原理”课程理论教学内容,基于 EDA 技术构建了一个多层次的计算机组成原理实验教学体系,包括基础实验、综合实验和高级设计实验 3 个层次。全书分为 3 章。第 1 章主要完成计算机基本组成部件的设计,第 2 章主要完成 MIPS 处理器的设计,第 3 章主要训练学生综合运用第 2 章学习的设计原理和方法,在原有指令系统的基础上进行扩展,并独立设计一个支持扩展指令集的 MIPS CPU 系统。

本书内容丰富,通俗易懂,由浅入深,实用性强。本书的一大特色是实验体系与先行及后续实验课程具有延续性。另外,每个实验有针对性地提出预习要求;典型电路给出了详细的实验原理;提供设计思路和必要的提示启发学生设计实验;提供主要模块的仿真波形。这些特色都便于有效地指导学生完成实验和深化学生对相关知识的理解,从而掌握利用硬件描述语言和 EDA 工具进行 MIPS 系统设计的方法。

本书可作为高等学校计算机专业学生“计算机组成原理”课程的配套实验教材,也可作为电子类、自动控制类各专业学生和相关领域工程技术人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

计算机组成原理实验教程 / 张亮,曹庆华,傅翠娇编著. —北京:清华大学出版社,2013.3

计算机科学与技术专业实践系列教材

ISBN 978-7-302-31421-9

I. ①计… II. ①张… ②曹… ③傅… III. ①计算机组成原理—实验—高等学校—教材 IV. ①TP301-33

中国版本图书馆 CIP 数据核字(2013)第 018504 号

责任编辑:张瑞庆

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:宋 林

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:13.75 字 数:332千字

版 次:2013年3月第1版 印 次:2013年3月第1次印刷

印 数:1~3000

定 价:24.00元

产品编号:051449-01

前 言

计算机组成原理实验是“计算机组成原理”课程教学中的一个重要实践环节。随着计算机技术的飞速发展,EDA 设计已经成为硬件设计的主要方式,原来基于小型模拟计算机系统的实验已经不能适应现代实验教学的需求。传统实验教学模式,大多采用中小规模集成电路芯片在面包板上搭建各种常见的数字电路,工作量大且易出错。而新的实验教学模式是利用硬件描述语言和 EDA 工具,首先在计算机上进行设计、修改和仿真,通过仿真波形可以直观地观察到电路内部的各个信号的变化情况,然后将设计下载到实验台上的 PLD 芯片中,进一步验证设计的正确性。通过一系列由浅入深、从局部到整体的实验,不仅加深学生对计算机组成原理基本知识的理解,而且有助于学生融会贯通这些知识点。通过在实验台上的验证与调试,锻炼学生的实际动手能力、分析问题和解决问题的能力。实验台提供了各种常用的输入输出设备,如按键、拨码开关、矩阵键盘、LED、数码管、数字 TFT 彩色液晶屏、字符型液晶屏、扬声器等,以及各种常见的输入输出接口,如 USB 接口、PS/2 键盘和 PS/2 鼠标接口、VGA 接口、以太网口、RS232 串口、SD 卡接口、语音采样口、立体声输出口、MIC 模拟输入口等。丰富的外设和接口有利于开设与计算机有关的各种实验,便于进行调试和直观观察实验现象。

因此,与理论教学相配套,我们将当前先进的设计理念和设计技术也引入到计算机组成原理实验教学中,基于 EDA 技术设计和开发了先进的、层次化的实验内容,并且编写了这本实验教材。全书实验内容分为基础实验、综合实验和高级设计实验三个层次。

第 1 章 MIPS 部件基础实验:主要完成 MIPS 系统中基础的组成部件设计,为后续处理器高级实验做好准备,包括存储单元(如 MIPS RAM 存储器、32 位 MIPS IR 指令寄存器、32 位 MIPS RegFile 寄存器堆、先进先出存储电路 FIFO 等)、运算单元(如 32 位 MIPS 乘法运算器、带进位算术逻辑运算器、32 位 MIPS 移位运算器等)、协处理器以及其他部件(如 32 位 MIPS 程序计数器、32 位 MIPS 多路选择器、32 位 MIPS 符号扩展单元、程序计数器电路等)。内容涵盖了计算机组成原理中独立部件设计的主要知识点。通过本章的实验,学生将加深对 MIPS 系统的组成部件及其电路结构、原理和功能理解,掌握利用硬件描述语言和 EDA 工具进行 MIPS 部件设计的一般方法。

第 2 章 MIPS 处理器综合实验:主要完成一个支持 57 条 MIPS 基本指令的处理器设计,包括指令系统、数据通路、多周期控制器以及多周期处理器的设计等。通过这些实验激发学生的学习兴趣,培养学生的综合设计能力,使理论课的知识与工程实践融会贯通。

第 3 章 MIPS 高级设计实验:本章实验是在第 1、2 章的基础上,进一步将 57 条 MIPS 指令集进行扩展,最多可扩展至 110 条指令。为了支持新增加指令的执行,原有的设计必须要做适当的调整,而对于这部分工作,我们只给出了目标或者方向,具体如何实现,需要学生灵活运用前面实验中所学习的知识和技术,甚至需要自行查阅一些相关文献才能顺利完成。本章主要是训练学生的独立思考和解决问题的能力。

在本教材中,每个实验开始前都有“预习要求”,针对实验原理和实验内容提出了若干思

考题和预习要求,引导学生在弄通实验原理的基础上做好充分的准备工作,以保证课堂实验效果。在每个实验完成后都有“实验报告要求”,通过撰写实验报告和回答与实验相关的思考题,引导学生对实验原理和设计思路进行总结,对实验现象进行分析,对典型问题进行探究和思考,达到巩固和深化知识的目的。

在教材的内容编排上,我们注重知识的贯通性和学生综合能力的培养,实验内容层层递进,难度逐步加大,有利于不同学校的教师根据实际情况,结合学生的水平和兴趣,开展分层次的教学,不同层次的学生可以有选择地完成不同的实验内容,从而达到因材施教、个性化发展的培养目标。

以 EDA 技术为基础的计算机组成原理实验,与先行的数字逻辑、后续的计算机接口与通信实验课程形成了完整的实验课程体系,通过这一系列硬件类课程的学习和实践,使计算机专业的学生在整个大学阶段熟悉从逻辑器件设计到计算机系统设计的全过程。

由于作者水平有限,肯定有不少考虑不周和不足之处,敬请使用本教材的教师和学生批评指正。

作者

2012年12月

目 录

第 1 章 MIPS 部件基础实验	1
实验 1-1 存储单元	1
实验 1-2 运算单元	30
实验 1-3 协处理器	48
实验 1-4 其他部件	53
第 2 章 MIPS 处理器综合实验	65
MIPS 处理器简介	65
实验 2-1 指令系统	68
实验 2-2 数据通路	84
实验 2-3 多周期控制器设计	97
实验 2-4 多周期处理器设计	111
第 3 章 MIPS 高级设计实验	125
实验 3-1 MIPS 指令扩展处理器设计	125
实验 3-2 基于 MIPS CPU 和 8255A 设计计算器	149
实验 3-3 基于 MIPS CPU、8255A、8253 设计循环程序	158
附录 1 32 位 MIPS 指令集	169
附录 1-1 R 型指令	169
附录 1-2 I 型指令	177
附录 1-3 陷阱指令	179
附录 1-4 存取指令	182
附录 1-5 分支指令	186
附录 1-6 跳转指令	189
附录 1-7 CP0 型指令	190
附录 1-8 异常指令	190
附录 2 状态机信号使能表	192
附录 3 输出信号表达式	197
附录 4 常量定义	202
附录 5 GW48-SOPC 电路模式	205
附录 6 GW48-SOPC 实验台引脚对照表	210
参考文献	211

第 1 章 MIPS 部件基础实验

基础实验包括 3 个实验,实验内容涉及 32 位 MIPS 系统中的基本组成部件(如存储器、指令寄存器、寄存器堆、算术逻辑运算器、乘除法器、移位器、程序计数器、多路选择器、符号扩展单元等)的设计、仿真以及硬件下载的方法和过程。通过本章的实验,学生将加深对 32 位 MIPS 系统中的基本组成部件及其电路结构、原理和功能的理解,掌握利用硬件描述语言和 EDA 工具进行 32 位 MIPS 系统基本部件设计的一般方法。

实验 1-1 存储单元

一、实验目的

- (1) 了解各种存储器的工作原理并掌握几种常用存储器的设计方法。
- (2) 了解指令寄存器、寄存器堆等的工作原理并掌握常用的设计方法。
- (3) 熟练掌握本实验单元中涉及的 Verilog HDL 语言的相关语法知识,并能独立运用 Verilog HDL 语言设计类似功能的器件。
- (4) 熟练运用 Verilog HDL 语言在 Quartus II 下,根据 32 位 mips 系统设计要求,编制存储器、指令寄存器、寄存器堆等,并能熟练地进行仿真测试和硬件下载。
- (5) 熟练运用原理图输入法在 Quartus II 下,根据先进先出存储电路图及功能要求,设计 FIFO 电路,并能熟练地进行仿真测试和硬件下载。

二、预习要求

- (1) 仔细阅读实验原理部分,熟悉实验中用到的 Verilog HDL 语法知识和相关理论知识。
- (2) 复习“计算机组成原理”课程关于存储单元的内容,熟悉 32 位 MIPS 存储器、指令寄存器、寄存器堆等的逻辑功能和工作原理。
- (3) 仔细阅读实验指导,用 Verilog HDL 语言和图形输入方式实现实验内容所设定的各项设计,并进行仿真。
- (4) 仔细阅读先进先出存储电路部分,熟悉 FIFO 的功能,并对照原理图,分析其工作原理。
- (5) 若仿真通过,则对各顶层设计文件做好引脚锁定。

三、实验内容

- (1) 设计 32 位 MIPS RAM 存储器。
- (2) 设计 32 位 MIPS IR 指令寄存器。
- (3) 设计 32 位 MIPS RegFile 寄存器堆。

(4) 设计先进先出存储电路 FIFO。

四、实验原理

1. 存储器

存储器(memory)是计算机系统记忆设备,用来存放程序和数据。计算机中的全部信息,包括输入的原始数据、计算机程序、中间运行结果和最终运行结果都保存在存储器中。它根据控制器指定的位置存入和取出信息。

存储器是用来存储程序和数据部件,有了存储器,计算机才有记忆功能,才能保证正常工作。按照用途分类,存储器可分为主存储器(内存)和辅助存储器(外存)。外存通常是磁性介质或光盘等,能长期保存信息。内存是指主板上的存储部件,用来存放当前正在执行的数据和程序,但是仅用于暂时存放程序和数据,关闭电源或断电时,数据就会丢失。

只读存储器(read-only memory,ROM)是一种只能读取数据的存储器。在制造过程中,先将数据转成电路,并制成光罩(mask),于集成电路制造过程中一并制造完成,其数据内容在制造后就不能更改,只能读不能写,因此得名。为了便于使用和大批量生产,进一步生产了可编程只读存储器(PROM)、可擦可编程只读存储器(EPROM)和电可擦可编程只读存储器(EEPROM)。ROM适用于内容固定不变、需大量生产的产品,例如电子计算机或嵌入式设备中的开机启动、字形表、电子游戏机程序与卡带等。

随机存取存储器(random access memory,RAM)又称为“随机存储器”,是与CPU直接交换数据的内部存储器,也叫主存。它可以随时读写(刷新时除外),而且速度很快,通常作为操作系统或者其他正在运行中的程序的临时数据存储媒介。主存(main memory)即电子计算机内部最主要的内存,用来加载各式各样的程序与数据,以供CPU直接运行与运用。由于DRAM的性价比很高,且扩展性也不错,是现今一般计算机主存的最主要部分。近年来生产最新的计算机所用的主存主要是DDR3 SDRAM。RAM内存可以进一步分为静态随机存取存储器(SRAM)和动态随机存取存储器(DRAM)两大类。SRAM具有快速访问的优点,但生产成本较为昂贵,一个典型的应用是高速缓存。而DRAM由于具有较低的单位容量价格,所以被大量地采用作为系统的主存。

下面通过两个实例,分别采用原理图输入法和文本输入法,介绍ROM和RAM的基本设计过程。

例 1.1 原理图输入法设计 ROM。

(1) 定制宏单元 LPM_ROM

① 首先编辑 mif 文件。选择 File|New 菜单命令,在 New 窗口中选择 Other Files 选项卡,进而选择 Memory Initialization File 项,单击 OK 按钮后,产生 ROM 数据文件大小选择窗口。由于采用的是 64 点 8 位正弦数据,选择 ROM 的参数,数据个数(Number)为 64,数据宽(Word Size)为 8。单击 OK 按钮,将出现如图 1.1 的 mif 文件数据表格。编辑数据,最后保存文件,在这里不妨取名为 singt. mif。

② 接着是 ROM 元件的选定,选择 Tools|Mega Wizard Plug-In Manager 菜单命令,产生如图 1.2 所示的界面,选中第一项,即 Create a new custom megafunction variation。

③ 单击 Next 按钮,产生如图 1.3 所示对话框,在左侧选择 storage 项下的 LPM_ROM,再选择 Cyclone 器件和 Verilog HDL 语言方式,最后输入 ROM 文件存放的路径和

Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	255	254	252	249	245	239	233	225
8	217	207	197	186	174	162	150	137
16	124	112	99	87	75	64	53	43
24	34	26	19	13	8	4	1	0
32	0	1	4	8	13	19	26	34
40	43	53	64	75	87	99	112	124
48	137	150	162	174	186	197	207	217
56	225	233	239	245	249	252	254	255

图 1.1 编辑 mif 文件

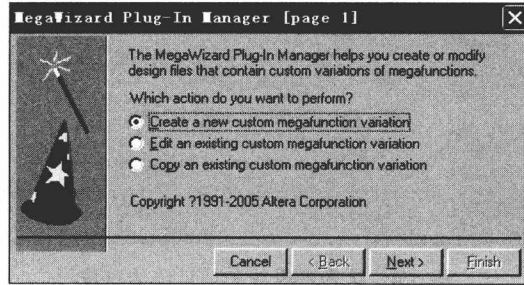


图 1.2 选择定制新的 LPM 功能单元

文件名。路径选择前述①中所建的工程路径,文件名为 sin_data.v。

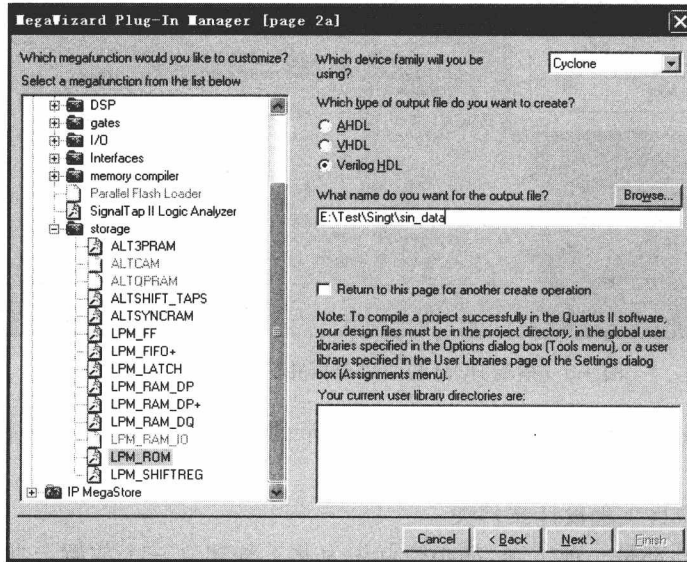


图 1.3 指定 LPM 宏功能

④ 单击 Next 按钮,出现如图 1.4 所示的对话框,选择 ROM 控制线和地址、数据线;在 What should the RAM block type be 栏选择默认 Auto,在适配中,Quartus II 将根据选中的目标器件系列,自动确定嵌入的 RAM 模块的类型;在 What clocking method would you like to use 栏中选择 Single clock 项。

⑤ 单击 Next 按钮后,出现如图 1.5 所示的对话框,在 Do you want to specify the

initial content of the memory 栏中,选择“**Yes, use this file for the memory content data**”项,并选择 Singt. mif;同时,选中 **Allow In-System Memory Content Editor to capture and update content independently of the system clock** 项。

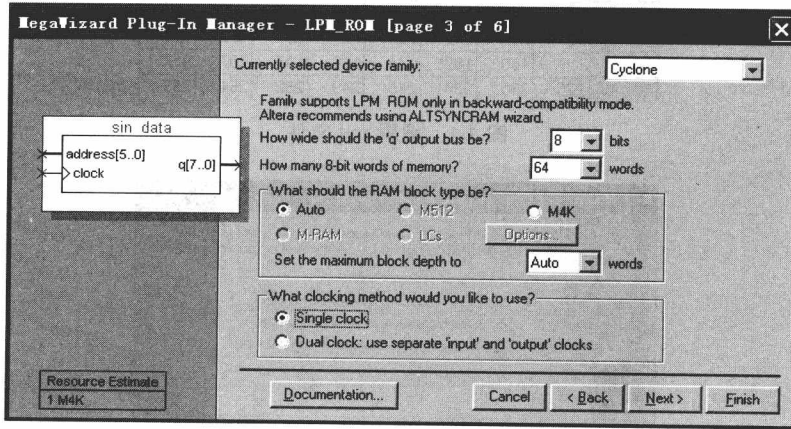


图 1.4 选择数据线和地址线的宽度

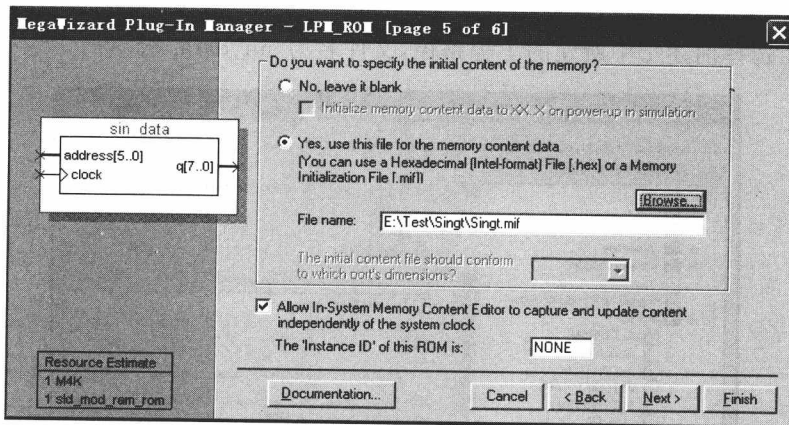


图 1.5 指定 ROM 初始化数据并选择在系统读写功能

⑥ 最后,单击 Next 按钮,出现如图 1.6 所示的界面,再单击 Finish 按钮后完成 ROM 的定制。

(2) 建立本项目工程设计文件夹

任何一项设计都是一项工程,都必须首先为此工程建立一个放置与此工程相关的所有设计文件的文件夹。不同的设计项目最好放在不同的文件夹中,而同一工程的所有文件都必须放在同一个文件夹中。假设文件夹取名为 ROM,路径为 D:\ROM。

注意: 尽量用英文字母来命名文件夹名,若为中文名可能会导致编译过程中报错。

(3) MIF 格式文件的建立

Quartus II 中 LPM_ROM 模块中的初始化数据文件的格式有两种: Memory Initialization File(. mif)格式和 Hexadecimal File(. hex)格式。在本例中,我们建立 mif 文件。

在 Quartus II 中,选择 ROM 数据文件编辑窗口: 在 File 菜单中选择 New 选项,并在

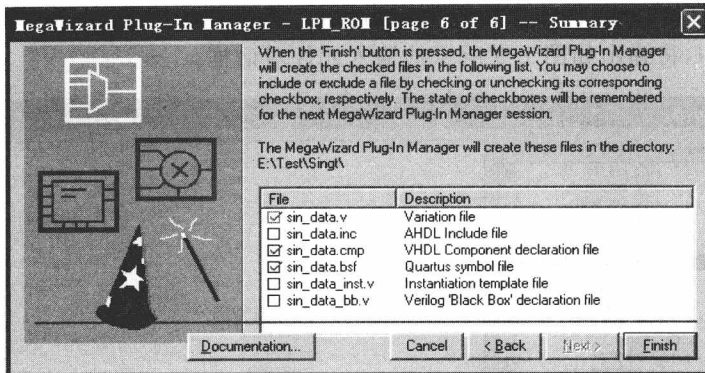


图 1.6 完成 LPM 宏单元的定制

New 窗口中选择 Other files 页,再选择 Memory Initialization File 项,单击 OK 按钮后,产生 ROM 数据文件大小选择窗口,这里选择 ROM 的数据数(Number)为 64,数据宽(Word size)取 24,单击 OK 按钮后,将出现如图 1.7 所示空的 MIF 数据表格。表格中的数据格式,可以通过鼠标右键单击窗口边缘的地址数据弹出的窗口选择。将数据逐个填入此表中。完成后,在 File 菜单中单击 Save as 按钮,保存此数据文件,取名为 rom_a.mif。

Addr	+0	+1	+2	+3	+4	+5	+6	+7
00	018108	00ED82	00C050	00E004	00B005	01A206	959A01	00E00F
08	00ED8A	00ED8C	00A008	008001	062009	062009	070A08	038201
10	001001	00ED83	00ED87	00ED99	00ED9C	31821D	31821F	318221
18	318223	00E01A	00A01B	070A01	00D181	21881E	019801	298820
20	019801	118822	019801	198824	019801	018110	000002	000003
28	000004	000005	000006	000007	000008	000009	00000A	00000B
30	00000C	00000D	00000E	00000F	000010	000011	000012	000013
38	000014	000015	000016	000017	000018	000019	00001A	00001C

图 1.7 rom_a.mif 中的数据

(4) 原理图输入项目设计

① 打开 Quartus II,选菜单 File|New,在弹出的 New 对话框中选择 Device Design Files 页的原理图文件编辑输入项 Block Diagram/Schematic File(如图 1.8 所示),按 OK 按钮后,将打开原理图编辑窗口。

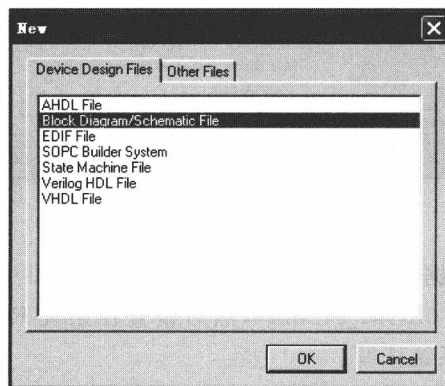


图 1.8 创建原理图文件

② 在编辑窗口中的任何一个位置上右击鼠标,将出现快捷菜单,选择其中的输入元件项 Insert|Symbol,将弹出如图 1.9 所示的输入元件对话框。

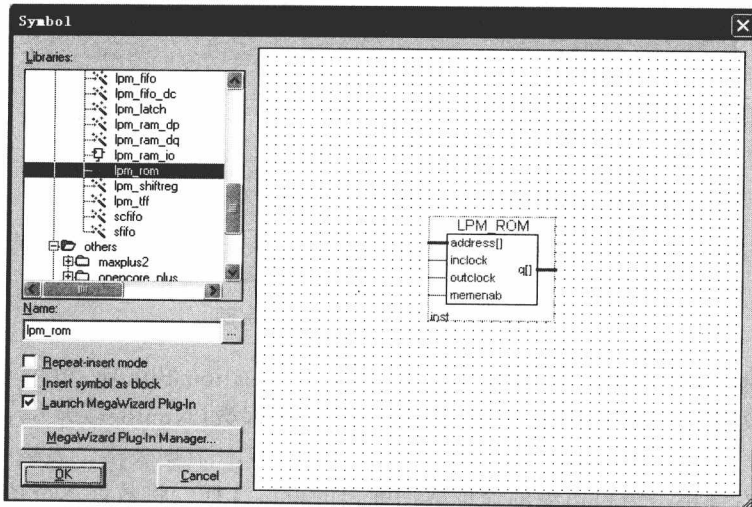


图 1.9 输入元件对话框

③ 单击“...”按钮,选中 lpm_rom 元件,此元件即显示在窗口中,然后单击 Symbol 窗口的 OK 按钮,即可进入 lpm_rom 元件的定制对话框,如图 1.10 所示。

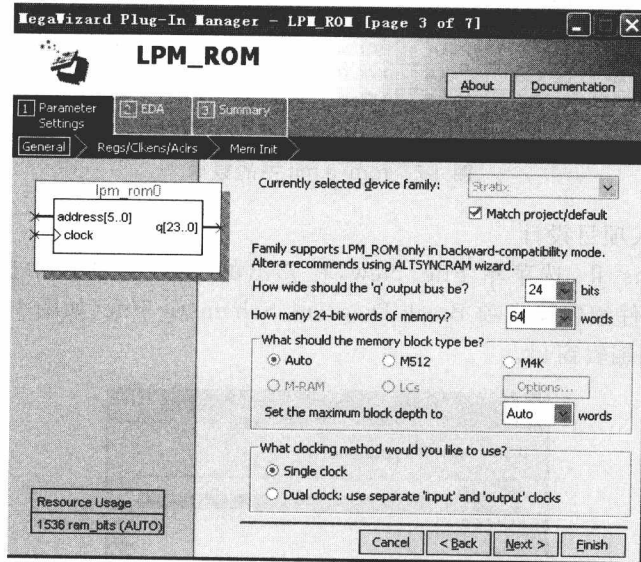


图 1.10 元件定制对话框

④ 在设置 lpm_rom 数据参数选择项 lpm_file 的对应窗口中(如图 1.11),用键盘输入 lpm_ROM 配置文件的路径(ROM_A.mif),然后设置在系统 ROM/RAM 读写允许,以便能对 FPGA 中的 ROM 在系统读写。

通过如上设置,在原理图编辑窗口中生成了 rom0 器件,其中地址总线宽度 address[]

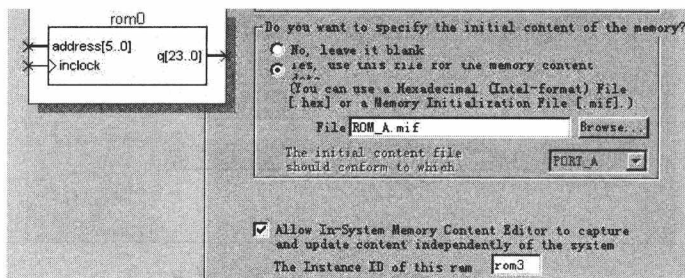


图 1.11 设置在系统 ROM/RAM 读写允许

和数据总线宽度 $q[]$ ，分别为 6 位和 24 位。用同样的方法，为 rom0 添加输入输出引脚，如图 1.12 所示进行设置和连接。

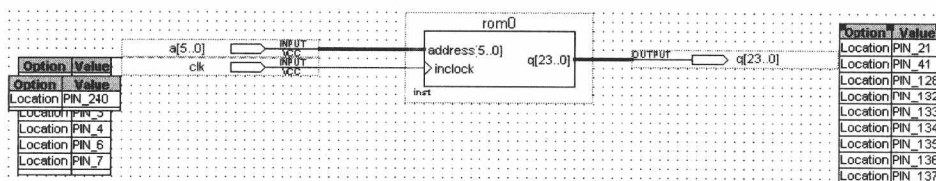


图 1.12 LPM_ROM 的结构图

⑤ 文件存盘。选择 File|Save As 命令，找到已设立的文件夹 D:\ROM，保存为 lpm_rom.bdf。当出现 Do you want to create 时，若单击“是”按钮，则直接进入创建工程流程。若单击“否”按钮，则可按照以下的方法进入创建工程流程。

(5) 创建工程

① 打开建立新工程管理窗口。选择菜单 File|New Project Wizard 命令，即弹出“工程设置”对话框，如图 1.13 所示。单击此对话框最上一栏右侧的“...”按钮，找到文件夹 D:\ROM，选中已存盘的 lpm_rom.bdf 文件，再单击“打开”按钮，即出现如图 1.13 所示的设置情况。其中第一行的 D:\ROM 表示工程所在的工作库文件夹；第二行为工程名；第三行是当前工程顶层文件的实体名。

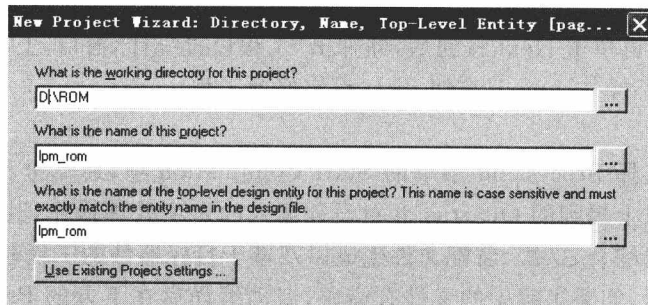


图 1.13 利用 New Project Wizard 创建工程

② 将设计文件加入工程中。单击下方的 Next 按钮，在弹出的对话框中单击 File 栏的按钮，将与工程相关的文件加入到此工程中，如图 1.14 所示。

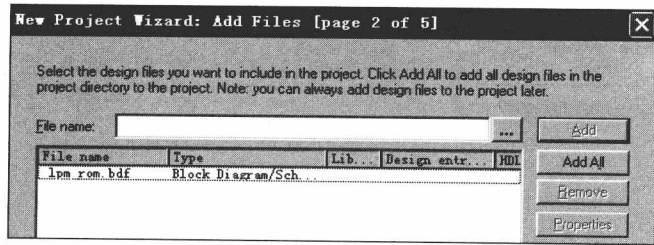


图 1.14 将设计文件加入到工程中

③ 单击 Next 按钮,选择仿真器和综合器类型。如果都选择默认的 None 项,表示都选 Quartus II 中自带的仿真器和综合器。在此都选择默认项 None。

④ 单击 Next 按钮,选择目标芯片,如图 1.15 所示。首先在 Family 栏选择芯片系列,在此选择 Cyclone 系列。然后,选择窗口右边的三个“列表”过滤选择:分别选择 Package 为 PQFP、Pin count 为 240 和 Speed grade 为 8。在 Available devices 栏选择适合的具体芯片,这里选择 EP1C12Q240C8。

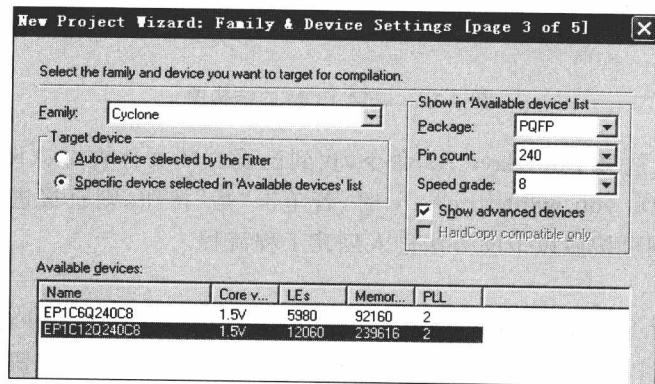


图 1.15 选择目标芯片

⑤ 单击 Next 按钮,进行工具设置。这里不做选择,表示仅选择 Quartus II 自含的所有设计工具。

⑥ 结束设置。再单击 Next 按钮后,即弹出“工程设置统计”窗口,上面列出了此项工程相关设置情况。然后单击 Finish 按钮,则已设定好此工程。

(6) 全程编译

编译前,首先选择 Processing 菜单的 Start Compilation 项,启动全程编译。这里所谓的全程编译,包括以上提到的 Quartus II 对设计输入的多项处理操作,其中包括排错、数据网表文件提取、逻辑综合、适配、装配文件生成以及基于目标器件的工程时序分析等。

如图 1.16 所示,如果工程中的文件有错误,启动编译后在下方的 Processing 处理栏中会显示出来。如果编译成功,可以见到,工程管理窗的左上角显示了工程的层次结构和其中结构模块耗用的逻辑宏单元数;在此栏下是编译处理流程,包括数据网表建立、逻辑综合、适配、配置文件装配和时序分析等;最下栏是编译处理信息;中栏是编译报告项目选择菜单,单击其中各项可以详细地了解编译与分析结果。

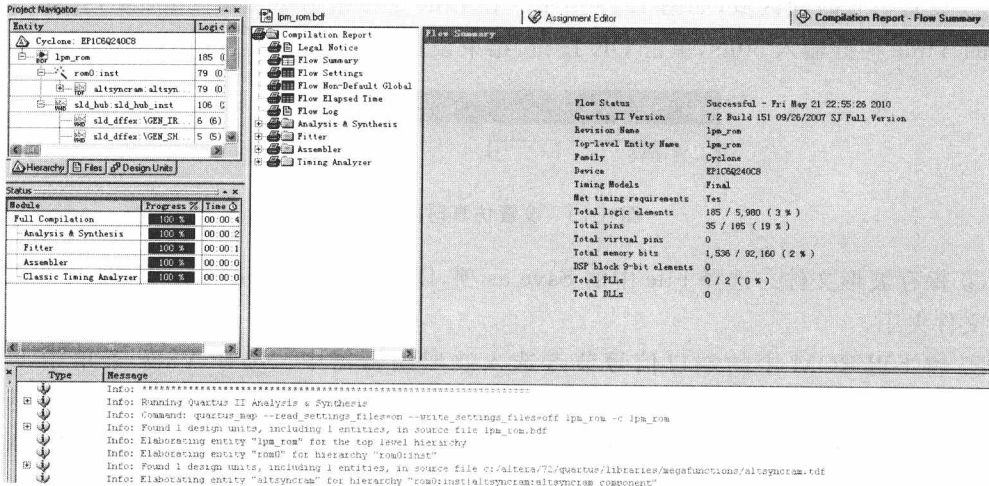


图 1.16 全程编译后出现的编译结果信息

(7) 时序仿真

① 创建波形文件。选择菜单 File 中的 New 项,在 New 窗口中,选择 Other Files 中的 Vector Waveform File(如图 1.17 所示),单击 OK 按钮,即出现空白的波形编辑器,如图 1.18 所示。

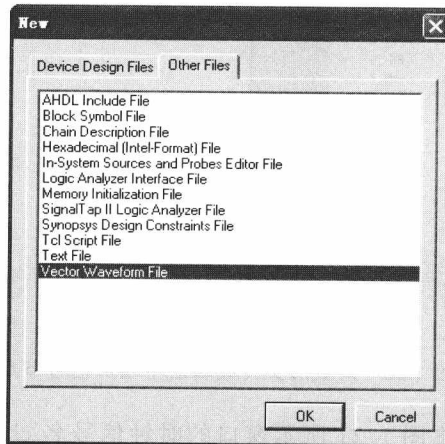


图 1.17 选择编辑矢量波形文件

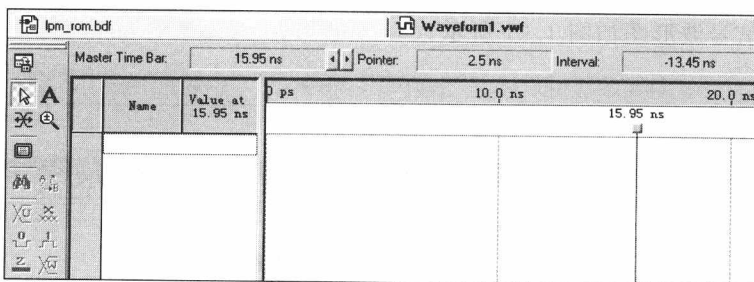


图 1.18 波形文件编辑器

② 设置仿真时间区域,如图 1.19 所示。在 Edit 菜单中选择 End Time 项,在弹出的窗口中的 Time 栏处输入 $50\mu\text{s}$,单击 OK 按钮,结束设置。

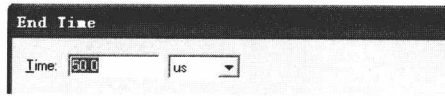


图 1.19 设置仿真时间长度

③ 保存波形文件。选择 File 中的 Save as 项,以名为 lpm_rom.vwf 的波形文件保存到项目文件夹中。

④ 将工程 ROM 中的端口信号节点选入波形编辑器中。选择 View 菜单中 Utility Windows 项的 Node Finder 选项。在 Filter 框中选 Pins:all,然后单击 List 按钮,于是在下方的 Nodes Found 窗口中出现设计中的 ROM 工程的所有端口引脚名。用鼠标将 clk、a、q 等信号分别拖到右侧,结束后单击 OK 按钮,如图 1.20 所示。

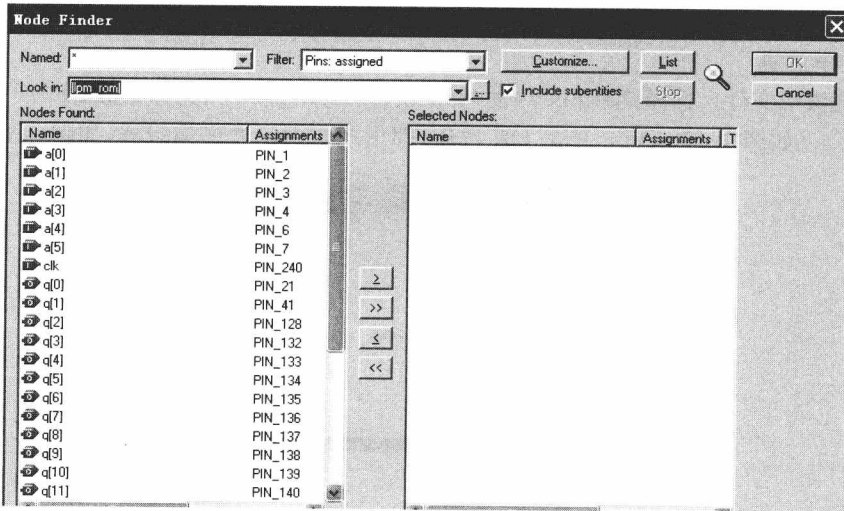


图 1.20 向波形文件编辑器中增加信号节点

⑤ 编辑输入波形。单击图 1.21 所示窗口的时钟信号名 clk,使之变成蓝色条,再单击左列的时钟设置键,在 Clock 窗口中设置 clk 的时钟周期为 100ns, Clock 窗口中的 Duty cycle 是占空比,默认为 50,即 50% 占空比,如图 1.22 所示。然后,分别设置 a、q 的电平。设置好的激励信号波形图如图 1.21 所示。

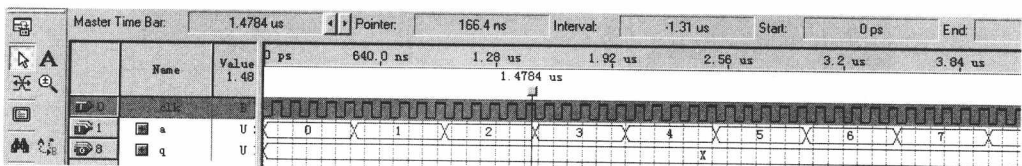


图 1.21 设置好的波形图

⑥ 输出信号数据格式设置。单击输出信号 q 左边的“+”号,则能展开 q 中的所有信号;如果双击此“+”号左边的信号标记,将弹出对该信号数据格式设置的对话框。在该对话框的 Radix 栏有 4 种选择,这里可选择十六进制数 Hexadecimal 表达方式,如图 1.23 所示。

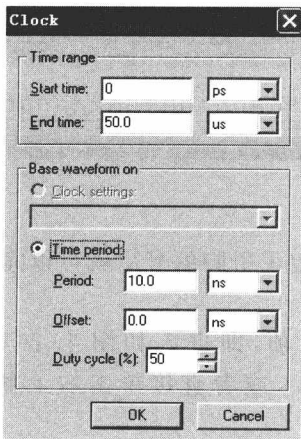


图 1.22 设置时钟 clk 的周期

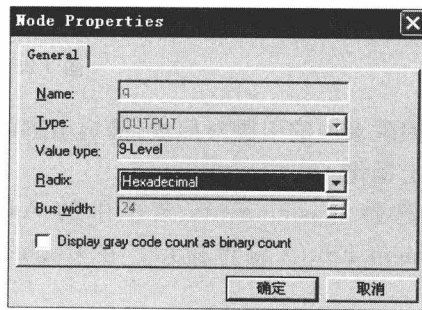


图 1.23 选择数据格式

⑦ 仿真器参数设置。如图 1.24 所示,选择菜单 Assignment 中的 Settings 选项,在 Settings 窗口下选择 Category | Simulator Settings,在右侧的 Simulation mode 项下选择 Timing 选项,即选择时序仿真,并选择仿真激励文件名 lpm_rom.vwf。毛刺检测 Glitch detection 为 1ns 宽度。选中 Run simulation until all vector stimuli are used 全程仿真等。

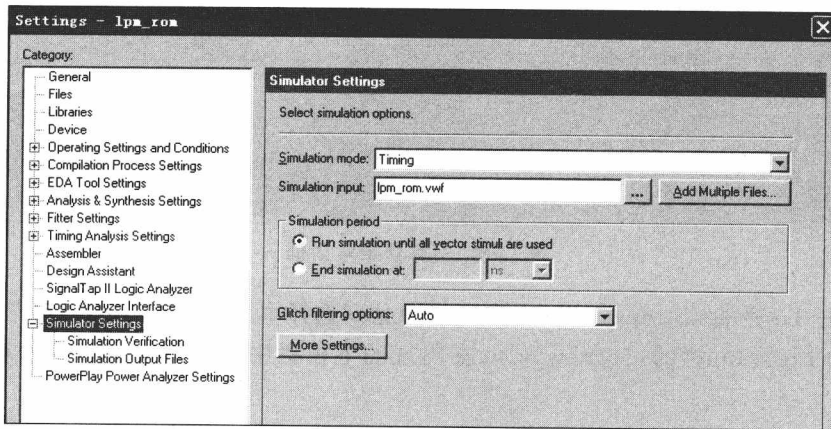


图 1.24 仿真器参数设置

⑧ 启动仿真。在菜单 Processing 项下选择 Start Simulation 项,直到出现 Simulation was successful,仿真结束。

⑨ 观察和分析仿真结果,验证硬件设计。如图 1.25 所示为本例的仿真波形输出图。

(8) 锁定引脚

为了能对计数器进行硬件测试,应该将输入输出信号锁定在芯片确定的引脚上,编译后下载。当硬件测试完成后,还必须对配置芯片进行编程,完成 FPGA 的最终开发。这里选