



普通高等教育“十一五”国家级规划教材 

现代计算机

潘松 潘明 编著

组成原理



 科学出版社
www.sciencep.com

普通高等教育“十一五”国家级规划教材

现代计算机组成原理

潘松 潘明 编著

科学出版社

北京

内 容 简 介

本书提供了基于 VHDL 和 EDA 技术的关于 CPU 和计算机系统的设计理论和设计技术方面较完整和丰富的内容。其中有基于微程序控制模式的 8 位 CISC 模型 CPU 设计技术; 基于状态机控制模式的 16 位 CISC CPU 设计技术; MCS51 系列单片机兼容型单片机 IP 软核系统设计方法; 基于流水线技术的 16 位 RISC CPU 设计技术, 以及基于 SOPC 技术的 32 位 NiosII 软核嵌入式系统软硬件设计技术。从授课内容到实验形式都能与目前国外计算机组成原理与计算机体系结构等同类课程的教学和实验有较好的接轨。

全书内容新颖实用, 吸收了欧美许多高校的计算机组成原理同类课程教学和实验方面的基本内容。书中每一章中都安排了对应的习题和实验, 首次为国内高校就这一课程的教学改革和相关实验内容的延拓提供了实用的教材。

本书可作为计算机专业的本科生、研究生的教科书, 或是作为传统的计算机组成原理课的教学与实验的补充资料, 也可作为电子类各专业高年级本科生、研究生系统设计参考教材, 或相关领域工程技术人员基于 EDA 技术的片上系统开发应用的参考书。

图书在版编目 (CIP) 数据

现代计算机组成原理/潘松, 潘明编著. —北京: 科学出版社, 2007

(普通高等教育“十一五”国家级规划教材)

ISBN 978-7-03-018449-8

I. 现… II. ①潘… ②潘… III. 计算机体系结构—高等学校—教材 IV. TP303

中国版本图书馆 CIP 数据核字 (2007) 第 005626 号

责任编辑: 赵卫江 / 责任校对: 赵燕

责任印制: 吕春珉 / 封面设计: 耕者设计工作室

科学出版社出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

铭浩彩色印装有限公司印刷

科学出版社发行 各地新华书店经销

*

2007 年 2 月第 一 版 开本: 787×1092 1/16

2007 年 2 月第 一 次印刷 印张: 26 1/2

印数: 1—3 000 字数: 614 000

定价: 33.00 元

(如有印装质量问题, 我社负责调换〈环伟〉)

销售部电话 010-62136131 编辑部电话 010-62138017 (HI01)

前 言

“现代计算机组成原理”可以作为普通计算机专业的一门独立的专业基础课，也可以作为现在的“计算机组成原理”和“计算机体系结构”同类课的后续课程，或作为这两门课程在教学内容和实验内容上的补充，即包括 EDA 技术、硬件描述语言、CPU 及计算机系统设计技术等内容的补充。

有关本书任务和目标的一些思考安排在第 1 章 1.1 节中。

作为预备知识，本书的第 2 章和第 3 章主要介绍针对 CPU 设计及其实现的 VHDL 语法知识、EDA 工具 QuartusII 6.0 的使用方法、FPGA 开发技术与测试方法。为了尽可能缩短授课时数，作者从典型电路设计实例的介绍中引出 VHDL 相关语句语法，同时，在给出完整并被验证过的 VHDL 描述外，还给出综合后的 RTL 电路图，以及表现该电路系统功能的时序波形图。对于容易出现设计错误或理解歧义的示例，还给出正误示例的比较和详细说明。目的是通过一些简单、直观、典型的实例，将 VHDL 中围绕 CPU 设计的最核心、最基本的内容解释清楚，使读者能在短时间内有效地掌握 VHDL 的主干内容，而不必花费过多的时间去单独学习语法。至于一些枝节的语法现象，如函数、过程等，在实际遇到时，可由学生自己去查阅有关资料。如果已在前期课程中讲授过 EDA 技术和 HDL 方面的知识，及 QuartusII 6.0 的应用，可以跳过这两章。

第 4 章主要介绍普通 CPU 组成部件的功能描述和基于 EDA 技术的设计方法，还包括一些基于 Cyclone FPGA 的开发技术和基于 QuartusII 6.0 的 LPM 宏功能模块的使用方法。其基本内容与普通“计算机组成原理”对应内容类似，只是给出了全新的设计方法。因此实验量比较大，且强化了 QuartusII 6.0 熟练使用方面的训练。

第 5 章的基本内容与普通“计算机组成原理”对应内容也有类似，即介绍了基于微程序控制的 8 位 CISC 模型 CPU 的设计方法。但在表述工具（VHDL）、设计工具（QuartusII）、实现平台（单一 FPGA）以及测试工具和方法上则与传统迥异。这主要表现在以下几个方面：① CPU 整机，包括 ALU、全部数据与控制信息通路、微程序 ROM 及其控制器、数据与程序 RAM、显示接口等部件全部实现于单一 FPGA 中；② CPU 整机由单一网表文件实现；③ 静态时序仿真可测试 CPU 内任何一点的信号波形与数据流向；④ 基于嵌入式逻辑分析仪，可以动态实时了解 CPU 内任何一点的信号波形与数据流向；⑤ 可在系统（In-System）了解、编辑、修改 CPU 中的微程序和软件指令。

第 5 章最后还介绍了一个基于有限状态机的 8 位 CISC CPU 的功能及指令系统，与传统 MCS51 单片机兼容的 IP 软核单片机及其系统构建方法、实现方法和调试方法。同样，此单片机 CPU、程序 ROM、数据 RAM 和接口等部件都可在单一 FPGA 中实现。若利用该 FPGA 中的嵌入式锁相环后，此单片机的运行速度可以比普通 8051 单片机快数倍。

第 6 章给出了一个基于有限状态机控制的 16 位 CISC CPU 的详细设计方法。全机由 VHDL 表述，借助于 QuartusII，其软硬件功能在单片 FPGA 中得到验证。通过此章

的学习和实验可使学生完整地掌握一个实用 16 位 CPU 的设计与实现技术。

第 7 章和第 8 章主要讨论基于 EDA 技术和 FPGA 平台的流水线构架 RISC CPU 的设计理论、设计技术和实现方法。第 8 章最后给出了相关的完整实验。这两章可作为《计算机体系结构》课的补充，主要指硬件设计和实现方面。

第 9 章和第 10 章介绍了比较新的基于 SOPC (System On Programmable Chip) 技术的 NiosII 嵌入式系统设计技术。NiosII 是一个 32 位 IP 软核嵌入式处理器，是一个面向用户的硬件结构可编辑系统，其整个应用系统，甚至多个并行处理器系统都可以同时在一单片 FPGA 中实现。它代表了现代计算机技术发展的一个新的趋势。

本书中各章大都安排了许多习题。绝大部分章节都安排了针对性较强的实验，使学生对每一章的课堂教学效果能及时通过实验得以强化。各章设置的大部分实验，除给出详细的实验目的、实验原理、实验思考题和实验报告要求外，还包含多个实验项目（层次），即：第一实验层次是与该章内容相关的验证性实验，课本提供了详细的并被验证的设计程序和实验方法，学生只需将书中提供的设计程序输入计算机，并按要求进行编译仿真，在实验系统上实现即可；第二实验层次是在上一实验基础上做进一步的发挥；此后的实验层次属于自主设计或创新性质的实验。授课教师可以根据本课程的实验学时数和教学实验的要求，以及学生的兴趣，以不同的方式或形式布置给学生完成。

现代计算机技术发展的速度异常迅猛，高等院校计算机科学与技术的教学将面临越来越大的挑战，这主要表现在两个方面：更多更新的知识有待传授；学生在该领域的自主创新能力有待更有效地提高。为了迎接这个挑战，本书力图在这两个方面都有所作为，但作者限于知识面，难免力不从心。作为抛砖引玉之作，望业内专家同行不吝斧正。

本书的撰写得到了 Altera 公司大学计划部 Bob Xu 先生的大力支持，在此表示衷心的感谢！

作者 E-mail: pan_m@163.com; 本书配套课件可浏览网址: www.kx-soc.com。

作者

于桂林电子科技大学

目 录

| | |
|--|----|
| 第 1 章 概述 | 1 |
| 1.1 现代计算机组成原理课的任务..... | 1 |
| 1.1.1 问题的提出..... | 1 |
| 1.1.2 探索解决问题的方法..... | 3 |
| 1.2 EDA 技术..... | 5 |
| 1.3 FPGA 器件..... | 6 |
| 1.3.1 FPGA 的发展历程..... | 6 |
| 1.3.2 Cyclone 系列 FPGA..... | 7 |
| 1.4 硬件描述语言 VHDL..... | 8 |
| 1.5 EDA 设计流程..... | 9 |
| 1.6 QuartusII 简介..... | 11 |
| 1.7 CISC 和 RISC 处理器..... | 12 |
| 1.8 FPGA 在现代计算机领域中的应用..... | 14 |
| 第 2 章 VHDL 与 QuartusII 应用 | 16 |
| 2.1 VHDL 基本语句语法..... | 16 |
| 2.1.1 组合电路的 VHDL 描述..... | 16 |
| 2.1.2 VHDL 语句结构..... | 18 |
| 2.2 时序电路描述..... | 22 |
| 2.2.1 D 触发器描述..... | 22 |
| 2.2.2 时序描述相关语法规则..... | 23 |
| 2.2.3 实现时序电路的 VHDL 不同表述..... | 27 |
| 2.3 全加器描述及相关语法..... | 29 |
| 2.3.1 半加器描述..... | 30 |
| 2.3.2 CASE 语句..... | 32 |
| 2.3.3 例化语句..... | 33 |
| 2.4 计数器设计及相关语法..... | 35 |
| 2.5 一般计数器设计..... | 38 |
| 2.5.1 相关语法说明..... | 39 |
| 2.5.2 程序功能和语法分析..... | 39 |
| 2.5.3 含并行置位的移位寄存器设计..... | 40 |
| 2.6 QuartusII 6.0 使用向导..... | 42 |
| 2.6.1 建立工作库文件夹和编辑设计文件..... | 42 |
| 2.6.2 创建工程..... | 43 |
| 2.6.3 编译前设置..... | 45 |

| | | |
|--------------|------------------------------|-----------|
| 2.6.4 | 全程编译 | 45 |
| 2.6.5 | 时序仿真 | 47 |
| 2.6.6 | 应用 RTL 电路图观察器 | 50 |
| 2.6.7 | 引脚锁定设置和下载 | 50 |
| 2.6.8 | 配置文件下载 | 52 |
| 2.6.9 | AS 模式和 JTAG 间接模式编程配置器件 | 53 |
| 2.7 | 嵌入式逻辑分析仪使用方法 | 55 |
| 2.8 | 原理图输入设计方法 | 58 |
| | 习题 | 61 |
| | 实验与设计 | 62 |
| 2-1 | 组合电路的设计 | 62 |
| 2-2 | 时序电路的设计 | 63 |
| 2-3 | 含异步清零和同步时钟使能的加法计数器的设计 | 63 |
| 2-4 | 用原理图输入法设计 8 位全加器 | 63 |
| 第 3 章 | VHDL 深入 | 65 |
| 3.1 | 数据对象 | 65 |
| 3.1.1 | 常数 | 65 |
| 3.1.2 | 变量 | 66 |
| 3.1.3 | 信号 | 66 |
| 3.1.4 | 进程中的信号赋值与变量赋值 | 67 |
| 3.2 | IF 语句概述 | 75 |
| 3.3 | 进程语句归纳 | 78 |
| 3.3.1 | 进程语句格式 | 78 |
| 3.3.2 | 进程结构组成 | 79 |
| 3.3.3 | 进程要点 | 79 |
| 3.4 | 并行语句例解 | 82 |
| 3.5 | 仿真延时 | 83 |
| 3.5.1 | 固有延时 | 83 |
| 3.5.2 | 传输延时 | 84 |
| 3.5.3 | 仿真 δ | 84 |
| 3.6 | 有限状态机 | 85 |
| 3.6.1 | 数据类型定义语句 | 85 |
| 3.6.2 | 一般有限状态机的设计 | 87 |
| 3.6.3 | Moore 型状态机 | 91 |
| 3.6.4 | Mealy 型状态机 | 96 |
| 3.7 | 双向和三态电路信号赋值 | 98 |
| 3.7.1 | 三态门设计 | 98 |
| 3.7.2 | 双向端口设计 | 99 |
| 3.7.3 | 三态总线电路设计 | 101 |

| | |
|-------------------------------------|------------|
| 习题 | 103 |
| 实验与设计 | 105 |
| 3-1 7段数码显示译码器设计 | 105 |
| 3-2 数控分频器的设计 | 107 |
| 3-3 8位十六进制频率计设计 | 109 |
| 3-4 ADC0809 采样控制电路实现 | 110 |
| 3-5 序列检测器设计 | 110 |
| 第4章 CPU 功能模块设计 | 112 |
| 4.1 8位 CPU 功能与结构 | 112 |
| 4.2 CPU 中的基本部件 | 113 |
| 4.2.1 算术逻辑单元 | 113 |
| 4.2.2 数据缓冲寄存器 | 116 |
| 4.2.3 移位运算器 | 116 |
| 4.2.4 程序存储器与数据存储器 | 118 |
| 4.2.5 先进先出存储器 FIFO | 127 |
| 4.2.6 流水线乘法累加器 | 128 |
| 4.2.7 程序计数器与地址寄存器 | 130 |
| 4.2.8 指令寄存器 | 132 |
| 4.2.9 指令译码器与控制器 | 132 |
| 4.2.10 时序产生器 | 136 |
| 4.3 数据通路设计 | 138 |
| 4.3.1 模型机的数据通路 | 138 |
| 4.3.2 模型机的电路结构 | 139 |
| 4.4 在系统存储器数据读写编辑器应用 | 140 |
| 4.5 嵌入式锁相环调用 | 142 |
| 4.5.1 建立嵌入式锁相环元件 | 142 |
| 4.5.2 测试锁相环 | 143 |
| 习题 | 144 |
| 实验与设计 | 145 |
| 4-1 算术逻辑运算单元 ALU 设计实验 | 145 |
| 4-2 带进位算术逻辑运算单元 ALU 设计实验 | 148 |
| 4-3 移位运算器设计实验 | 150 |
| 4-4 LPM_ROM 实验 | 151 |
| 4-5 LPM_RAM 实验 | 152 |
| 4-6 LPM_FIFO 实验 | 154 |
| 4-7 FPGA 与外部 16 位 RAM 接口实验 | 155 |
| 4-8 微控制器实验 1: 节拍脉冲发生器时序电路实验 | 156 |
| 4-9 微控制器实验 2: 程序计数器 PC 与地址寄存器 AR 实验 | 157 |
| 4-10 微控制器实验 3: 微控制器组成实验 | 159 |

| | |
|--|------------|
| 4-11 正弦信号发生器设计 | 162 |
| 第 5 章 8 位 CISC 计算机设计 | 163 |
| 5.1 8 位 CPU 结构 | 163 |
| 5.2 指令系统的结构及功能的确定 | 166 |
| 5.2.1 模型机指令系统 | 166 |
| 5.2.2 拟定指令流程和微命令序列 | 167 |
| 5.2.3 微程序设计 | 171 |
| 5.3 8 位 CPU 的硬件系统设计 | 173 |
| 5.3.1 CPU 顶层设计 | 173 |
| 5.3.2 取指令和指令译码 | 175 |
| 5.3.3 设计微代码表 | 176 |
| 5.3.4 建立数据通路 | 177 |
| 5.3.5 运算器 ALU 的设计 | 178 |
| 5.3.6 控制执行单元 | 178 |
| 5.3.7 在模型 CPU 中的软件执行 | 179 |
| 5.3.8 模型 CPU 的硬件仿真 | 182 |
| 5.4 具有移位功能的 CPU 设计 | 185 |
| 5.4.1 移位运算器的 VHDL 设计 | 185 |
| 5.4.2 移位运算器与 ALU 的结合设计 | 186 |
| 5.5 含更多指令的 CPU 模型机设计 | 187 |
| 5.5.1 指令系统的格式 | 188 |
| 5.5.2 寻址方式确定 | 189 |
| 5.5.3 微程序控制流程图设计 | 190 |
| 5.5.4 微程序代码在 LPM 模块中的加载 | 191 |
| 5.6 8051 单片机 IP 软核应用系统构建 | 193 |
| 5.6.1 K8051 单片机软核基本功能和结构 | 193 |
| 5.6.2 K8051 单片机软核实用系统构建和软件测试 | 196 |
| 习题 | 198 |
| 实验与设计 | 199 |
| 5-1 基本模型计算机设计与实现 | 199 |
| 5-2 带移位运算的模型机设计与实现 | 205 |
| 5-3 含 16 条指令的 CPU 设计与实现 | 210 |
| 5-4 较复杂 CPU 应用程序设计实验 | 213 |
| 5-5 K8051 单片机核基本系统构建和测试实验 | 219 |
| 5-6 基于 K8051 核的液晶显示与等精度频率测试系统设计 | 220 |
| 5-7 基于 K8051 核的数码管显示与等精度频率测试系统设计 | 222 |
| 第 6 章 16 位 CISC CPU 设计 | 223 |
| 6.1 顶层系统设计 | 223 |
| 6.1.1 16 位 CPU 的组成结构 | 223 |

| | | |
|--------------|--------------------------------|------------|
| 6.1.2 | 指令系统设计 | 224 |
| 6.1.3 | 顶层结构的 VHDL 设计 | 227 |
| 6.1.4 | 软件设计实例 | 230 |
| 6.2 | CPU 基本部件设计 | 232 |
| 6.2.1 | 运算器 | 232 |
| 6.2.2 | 比较器 | 234 |
| 6.2.3 | 控制器 | 236 |
| 6.2.4 | 寄存器与寄存器阵列 | 240 |
| 6.2.5 | 移位寄存器 | 242 |
| 6.2.6 | 三态寄存器 | 244 |
| 6.3 | CPU 的时序仿真与实现 | 245 |
| 6.3.1 | 编辑仿真波形文件 | 245 |
| 6.3.2 | 16 位 CPU 的调试运行 | 248 |
| 6.3.3 | 应用嵌入式逻辑分析仪调试 CPU | 248 |
| 6.3.4 | 对配置器件编程 | 250 |
| 6.4 | 应用程序设计实例 | 251 |
| 6.4.1 | 乘法算法及其硬件实现 | 251 |
| 6.4.2 | 除法算法及其硬件实现 | 251 |
| | 习题 | 252 |
| | 实验与设计 | 253 |
| 6-1 | 16 位计算机基本部件实验 | 253 |
| 6-2 | 16 位 CPU 设计综合实验 | 253 |
| 第 7 章 | 流水线结构 RISC CPU 设计 | 259 |
| 7.1 | 流水线的一般概念 | 259 |
| 7.1.1 | DLX 指令流水线结构 | 260 |
| 7.1.2 | 流水线 CPU 的时空图 | 263 |
| 7.1.3 | 流水线分类 | 264 |
| 7.2 | 流水线中的主要问题及处理 | 265 |
| 7.2.1 | 资源相关 | 265 |
| 7.2.2 | 数据相关及其分类 | 266 |
| 7.2.3 | 数据竞争的处理技术 | 267 |
| 7.2.4 | 控制相关 | 273 |
| 7.2.5 | 流水实现的关键技术 | 273 |
| 7.3 | 流水线的性能评价 | 275 |
| 7.3.1 | 流水线的性能指标 | 275 |
| 7.3.2 | 应用举例 | 277 |
| 7.3.3 | Amdahl 定律 | 283 |
| | 习题 | 284 |
| | 实验与设计 | 286 |

| | |
|-------------------------------------|------------|
| 7-1 乘法器实验 | 286 |
| 7-2 除法器实验 | 290 |
| 第 8 章 16 位流水线 CPU 设计 | 293 |
| 8.1 流水线 CPU 的结构 | 293 |
| 8.2 指令系统设计 | 294 |
| 8.2.1 寄存器型 (R 型) | 296 |
| 8.2.2 寄存器立即数型 (RI 型) | 296 |
| 8.2.3 立即数型 (I 型) | 296 |
| 8.2.4 立即移位型 (SI 型) | 296 |
| 8.3 数据通路设计 | 297 |
| 8.3.1 R 型数据通路 | 297 |
| 8.3.2 RI 型数据通路 | 297 |
| 8.3.3 装入字数据通路 | 297 |
| 8.3.4 存储字数据通路 | 298 |
| 8.3.5 寄存器转移数据通路 | 298 |
| 8.4 流水线各段设计和功能描述 | 299 |
| 8.4.1 Stage 1 取指令段 | 299 |
| 8.4.2 Stage 2 译码段 ID | 306 |
| 8.4.3 Stage 3 执行有效地址计算段 (EXE) | 313 |
| 8.4.4 Stage 4 访存段 (MEM) | 318 |
| 8.4.5 Stage 5 回写段 (WB) | 321 |
| 8.4.6 一些关键功能部件的设计 | 321 |
| 8.4.7 控制单元 | 328 |
| 8.4.8 中断与异常 | 328 |
| 8.4.9 流水线 CPU 系统电路结构 | 329 |
| 8.4.10 CPU 与 LCD 显示模块的接口 | 330 |
| 习题 | 333 |
| 实验与设计 | 333 |
| 8-1 Stage 1 取指令段实验 | 333 |
| 8-2 Stage 2 指令译码段实验 | 336 |
| 8-3 Stage 3 指令执行段实验 | 340 |
| 8-4 Stage 4/5 存储与写回段实验 | 343 |
| 8-5 数据相关性控制实验 | 344 |
| 8-6 数据通路实验 | 345 |
| 8-7 流水线 CPU 综合实验 | 346 |
| 第 9 章 32 位 IP 软核嵌入式系统 | 351 |
| 9.1 Nios/NiosII 软核处理器 | 351 |
| 9.2 Nios 软核处理器及其设计流程 | 353 |
| 9.2.1 Nios 处理器 | 353 |

| | |
|---|------------|
| 9.2.2 Nios 处理器内部结构 | 354 |
| 9.2.3 Nios 系统硬件开发流程 | 355 |
| 9.2.4 Nios 系统软件开发流程 | 356 |
| 9.3 32 位 NiosII 系统设计 | 357 |
| 9.3.1 NiosII 系统的优势 | 358 |
| 9.3.2 NiosII 系统开发流程 | 359 |
| 9.4 定制 Avalon 总线外设 | 362 |
| 9.5 DMA | 364 |
| 9.6 定制硬件指令 | 364 |
| 第 10 章 NiosII 嵌入式系统软硬件设计 | 367 |
| 10.1 NiosII 基本硬件系统构建 | 367 |
| 10.1.1 设计模型准备 | 367 |
| 10.1.2 NiosII 系统加入组件 | 370 |
| 10.1.3 NiosII 系统生成前设置与系统生成 | 376 |
| 10.1.4 NiosII 硬件系统生成 | 377 |
| 10.2 NiosII 软件设计与运行流程 | 379 |
| 10.3 加入用户自定义组件设计 | 390 |
| 10.4 加入用户自定义指令设计 | 398 |
| 10.5 Flash 编程下载 | 401 |
| 习题 | 403 |
| 实验与设计 | 403 |
| 10-1 设计一个简单的 NiosII 系统 | 403 |
| 10-2 秒表程序设计 | 403 |
| 10-3 为 NiosII 系统定制复数乘法器硬件加速指令 | 404 |
| 10-4 利用 NiosII 系统完成简单计算器程序设计与功能实现 | 406 |
| 10-5 Avalon 从外设 (数码管动态扫描显示模块) 设计 | 406 |
| 附录 GW48-CP+实验系统相关资料 | 407 |
| 一、GW48-CP+系统实验信号名与芯片引脚对照表 | 407 |
| 二、实验电路结构图 | 408 |
| 参考文献 | 412 |

第1章 概 述

以下首先给出对本书的一些相关思考, 然后对后面各章可能涉及到的概念和硬件设计平台做一简述。

1.1 现代计算机组成原理课的任务

图 1-1 是目前我国大多数高校计算机科学与技术专业的基本课程体系简图。图中给出的课程设置结构显然是合理的。例如, 可以发现计算机组成原理处于核心地位, 成为计算机专业的一门核心课程, 这是因为它在计算机类专业的课程地位上起着承上启下、软硬件兼容的重要作用, 这一事实也得到了专业人士的普遍认可。

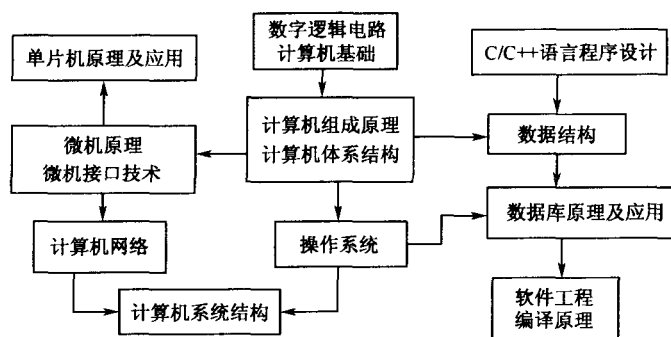


图 1-1 计算机专业基本课程体系简图

1.1.1 问题的提出

如果仔细分析图中所列的课程内容, 并考虑到一些高校在相关课程上的实际教学操作, 不难发现其中隐含着如下两点缺憾。

1. 缺失 CPU 硬件设计这一重要内容

首先必须明确, CPU、嵌入式处理器、DSP 处理器乃至计算机系统的设计技术及相关课程理应纳入计算机科学与技术专业中, 计算机专业对于软硬件综合设计人才的培养是责无旁贷的。而电子信息、通信工程、工业自动化等专业从整个课程体系来说, 都不可能插入这一课程。

其次是必要性。显然如果只会使用计算机而不会设计 CPU 和计算机, 只能永远依赖于别人的硬件平台, 这绝非我国的办学宗旨。

随着科学技术的发展, 核心技术已经愈来愈集中在集成电路芯片和软件这两项技术之中, 其中 CPU 和 OS 设计技术是最核心的两项技术。特别是高性能计算机技术一直是衡量国家实力的一个重要标志。美国、日本和西欧的一些国家不断加大这方面的

资助力度。美国的许多高校本科计算机专业中都安排了 CPU 设计方面的课程和实验内容。

例如麻省理工学院的一门相关课程是计算机系统设计。学生在实验课中,需自主完成(即自行设计)ALU、单指令周期 CPU(Single-cycle CPU)、多指令周期 CPU(Multi-cycle CPU),乃至实现流水线 32 位 MIPS CPU 和 Cache 的设计。斯坦福大学计算机系的本科生也有相似的课程和实验,即计算机组成与设计课。实验要求学生以各自独立的形式,用 VHDL 语言自主实现 CPU、VGA 显示控制模块等接口,最后实现于 FPGA 中,并完成软硬件调试。此外,如加州大学伯克利分校和杨百翰大学等学校在基于 FPGA 的超级计算机研制方面也有大量成果。

然而我国在 CPU 和计算机系统设计方面还十分落后,具有成熟的自主知识产权的 8 位/16 位 CPU 产品基本没有,高端的 32 位到 64 位高性能处理器更是空白。国产装备中的微处理器几乎全部采用进口的国际流行的通用或专用微处理器。这种受制于人的状况对于我国的 IT 产业、国家重要的经济军事战略乃至国家安全都十分不利!

从我国高校教学方面来看也同样不容乐观。虽然由图 1-1 可见,计算机组成原理和计算机体系结构是计算机专业的核心课程,且从两门课的内容来说理应肩负这一重任。但实际情况并非如此,调研表明,国内除诸如中国科技大学、北京大学、复旦大学、哈尔滨工业大学、成都电子科技大学等少数高校十分重视计算机系统课程建设,深入广泛地采用 EDA 技术与硬件描述语言完成实验和设计训练,并将计算机组成原理和计算机体系结构课的实验明确为 CPU 设计外,其他学校则大多仅将计算机组成原理定位为“计算机科学导论”和计算机模型认知的层面上;而在实验与实践方面,作为探讨 CPU 硬件原理和设计的计算机体系结构课基本没有对应的硬件实验;计算机组成原理的实验则主要是在一些由分离元件构成的实验平台上,完成简单模型 CPU 的验证性实验,基本谈不上设计,更没有国外高校类同的自主创新型 CPU 设计实验任务。这显然难以满足教育部在《关于加强高等学校本科教学工作,提高教学质量的若干意见》中关于“高等学校要重视本科教学的实验环节,保证实验课的开出率达到本科教学合格评估标准,并开出一批新的综合性、设计性实验”的要求。

2. 自主创新能力培养与训练方面的课程内容偏少

计算机学科领域中自主创新能力的培养,目标应该是拥有自主知识产权计算机部件或系统设计技术及创新设计能力的人才的培养。这就要求包含“自主”这一重要因素。

“创新”未必包含“自主”。例如,某项计算机软件的设计完成,某 DSP 算法的实现,某嵌入式系统控制软件的开发成功等,都可能包含一些前人未曾有过的创新,甚至可以有自己的知识产权。但我们却从来没有听说过,它们会是拥有完全自主知识产权的项目。这是因为,这些尽管属于创新型项目,但却都基于某类 CPU 平台上的软件,离开了这些 CPU,软件中的所有创新价值都将归于零,因为 CPU 是别人的。这就是说,创新能力的培养决不能脱离自主创造设计能力的培养,没有了自主的创新便不是真正的创新。

胡锦涛在 2006 年全国科学大会上的讲话中提出了到 2020 年将我国建设成为创新型国家的宏伟目标,并在讲话中多次强调:建设创新型国家,核心就是坚持自主创新,将

增强自主创新能力作为发展科学技术的战略基点。江泽民也曾指出：原始创新孕育着科学技术质的变化和发展，是一个民族对人类文明进步做出贡献的重要体现，也是当今世界科技竞争的制高点。

根据以上的讨论，不难发现，在计算机技术的教学中，唯有处于核心地位的计算机组成原理和计算机体系结构课程中加入相关的教学内容和合理的实验设置才能够承担起除基本知识传播外，对于学生自主创新能力，即原始性创新能力培养的重任。

然而正如前面指出的，就目前的情况看，这两门课尚未能承担其应有的角色。

1.1.2 探索解决问题的方法

基于以上问题的考虑，本书拟给出探索性的对策，以资业内专家参考。

本书提供了基于 VHDL 和最新 EDA 技术的关于 CPU 和计算机系统的设计理论和设计技术方面较丰富与完整的内容。其中有基于微程序控制模式的 8 位 CISC 模型 CPU 设计，基于状态机控制模式的 16 位 CISC CPU 设计，MCS51 单片机系列兼容型单片机软核系统设计，基于流水线技术的 16 位 RISC CPU 设计，以及基于 SOPC 技术的 32 位 NiosII 软核嵌入式系统软硬件设计。

由于所有示例和设计都是基于 Altera 最新推出的 Cyclone/II 系列 FPGA 硬件平台和新版 QuartusII 6.0 工具软件平台的，从而使得整个设计，从每一个逻辑门至锁存器、移位寄存器，从 RAM、ROM、FIFO、锁相环 PLL、Cache、ALU 至 DMA、硬件通信接口，从 8 位 CPU 至 32 位嵌入式系统，即从最基本的部件至整个宏观系统，几乎全部能用 VHDL 或嵌入式核表达，并实现于一个单片 FPGA 中。同时利用平台提供的高效的软硬件调试和测试工具优化设计，如逻辑锁定技术、增量编译技术、基于功能和时序仿真的硬件静态测试方法、基于能完整配置进 FPGA 内部的嵌入式逻辑分析仪 SignalTapII 的动态实时测试方法，以及内部存储器在系统编辑器 In-System Memory Content Editor 测试工具等；而在实现技术方面，又可利用 HardCopy 技术来完善设计。显然这一切在传统的教学内容、实验模式和实验手段上是没有的。根据以上提出的教学与实验的内容和模式，本书的目标如下。

1. 与国际接轨

无疑，教程中的多数内容是能与发达国家许多高校计算机本科相关教学内容和实验模式相吻合的。以致我们的学生也有可能像国外学生那样，带上自己的设计芯片、系统设计 HDL 程序、整机网表文件和时序仿真结果去应聘，其效果或许比普通的成绩单更能说明问题。当然，仅就目前国内多数传统实验模式来说是难以实现的，因为它含有许多分离的专用器件，无法将学生所有的创新思想、创新成果和自主能动性仅用一个网表文件来表达，并实现于单一芯片 FPGA 中。

2. 含有符合现代工程设计技术的 CPU 设计内容

现代计算机组成原理中明确包含了符合现代工程设计技术的 CPU 设计理论、设计方法和实现技术，其意义是多方面的。

首先，在原有的计算机组成原理和计算机体系结构课中增加了理论向工程实际转化

的符合现代计算机系统设计规范的硬件设计内容。而在传统的实验模式中往往不是这样，因为在传统的实验模式中虽然也有“设计”内容，但主要是根据不同实验系统各自规定的方法，用既定的分离元件（也有的包括部分可编程器件）和接口器件进行拼搭搭接而成。学生显然无法从这样的“设计”过程中了解真实的现代实用 CPU 基本设计技术。

其次，这使得学生在了解了计算机组成原理和软件设计技术的同时，学会了计算机硬件设计技术。计算机的软硬件设计技能是一个合格的计算机专业学生本应具备的基本知识。而且，离开了硬件设计，自主创新能力的培养便无从谈起。

然而目前的状况却令人堪忧。不少计算机专业学生存在“重软轻硬”、“欺软怕硬”，甚至“只软不硬”的现象，学生们只将注意力和兴趣集中在各种编程环境、开发工具、数据库、计算机网络的集成技术上面，对于硬件技术的学习和应用研究不感兴趣或望而生畏。甚至有的学者也认为，计算机专业的学生可以“只用键盘、鼠标就能演奏出各种美妙的音乐”。这种现象和认识对于我国培养自主创新型人才是极为不利的。

3. 创造能力的培养

从创造心理学的角度看，单纯的逻辑性思维和收敛性思维绝不可能产生创造，尤其是原始创造。创造型思维向来植根于发散性思维。美国心理学家吉尔福特认为，发散思维能力是创造力的核心。一切创造和创新都是发散性思维与收敛性思维、非逻辑思维与逻辑思维、分析思维与直觉思维共同作用的结果。司马光砸缸、爱因斯坦相对论、薛定谔方程、达尔文进化论、德布罗意波粒二象性学说、麦克斯韦方程、魏格纳大陆板块漂移学说，无一不是这些思维能力共同作用的经典范例。

在此不妨以程序设计为例。对于普通的软件描述语言，如 C 语言或汇编语言，无论程序多么复杂，CPU 在一个指令周期中只能执行一条指令，完成一个单项操作功能，且按照顺序执行规则实现整体功能。因而在这类软件开发和排错中，可以进行程序指令的“单步执行”调试。而设计人员也只需逐条顺序堆积指令，便能使计算机完成任何大型的任务。换言之，软件设计者可以通过编写这些顺序执行的指令，把任何复杂的任务转变为一个单向一维的工作流程。

从创造能力培养的角度看，软件设计人员只需拥有逻辑上的单向一维思维能力就能保证软件设计的成功，即过量单纯的软件设计，不断强化了设计者纯逻辑性思维和收敛性思维。显然，这与培养植根于多维多向的发散性思维方式和非逻辑思维方式的创造能力是相违背的。因此，多数纯软件设计训练只能归类为技能性和知识性训练。

爱因斯坦说过：“想象力比知识更重要，因为知识是有限的，而想象力概括着世界上的一切，推动着进步，并且是知识进化的源泉。”因此，计算机专业课程除了传授技能和知识外，激发学生的想象力、拓展学生对多学科的适应力、培养学生的创造力将更为重要。

4. 增加了有关现代计算机组成的新知识点

仅就知识点而言，本书也给出了许多与现代计算机组成相关的内容。如最新版的 QuartusII 应用、大规模 FPGA 应用、基于 JTAG 的多种测试手段、嵌入式模块的应用、

SOPC 技术、软核处理器系统设计、硬件描述语言、硬件仿真、软硬件联合设计与调试方法等。

1.2 EDA 技术

EDA (Electronic Design Automation) 技术作为现代电子设计技术的核心, 它依赖功能强大的计算机, 在 EDA 工具软件平台上, 对以硬件描述语言为系统逻辑描述手段完成的设计文件, 自动地完成逻辑编译、逻辑化简、逻辑分割、逻辑综合、结构综合 (布局布线), 以及逻辑优化和仿真测试, 直至实现既定的电子线路系统功能。EDA 技术使得设计者的工作仅限于利用软件的方式, 就能完成对系统硬件功能的实现。

在硬件实现方面, EDA 技术融合了大规模集成电路制造技术, IC 版图设计技术、FPGA/CPLD 编程下载技术、自动测试技术等; 在计算机辅助工程方面融合了计算机辅助设计 (CAD)、计算机辅助制造 (CAM)、计算机辅助测试 (CAT)、计算机辅助工程 (CAE) 技术以及多种计算机语言的设计概念; 而在现代电子学方面则容纳了更多的内容, 如计算机设计技术、电子线路设计理论、数字信号处理技术、数字系统建模和优化技术及基于微波技术的长线技术理论等。因此 EDA 技术为现代电子理论和设计的表达与实现提供了可能性。

一般地, 利用 EDA 技术进行电子系统设计的最后目标, 是完成专用集成电路 ASIC 的实现, ASIC 作为最终的物理平台, 集中容纳了用户通过 EDA 技术将电子应用系统的既定功能和技术指标具体实现的硬件实体。一般而言, 专用集成电路就是具有专门用途和特定功能的独立集成电路器件, 根据这个定义, 作为 EDA 技术最终实现目标的 ASIC, 可以通过以下两种途径来完成。

1. 超大规模可编程逻辑器件

FPGA (Field Programmable Gate Array) 和 CPLD (Complex Programmable Logic Device) 是实现这一途径的主流器件。它们的特点是直接面向用户, 具有极大的灵活性和通用性, 使用方便, 硬件测试和实现快捷, 开发效率高, 成本低, 上市时间短, 技术维护简单, 工作可靠性好等。FPGA 和 CPLD 的应用是 EDA 技术有机融合软硬件电子设计技术、SoC 和 ASIC 设计, 以及对自动设计与自动实现最典型的诠释。由于 FPGA 和 CPLD 的开发工具、开发流程和使用方法与 ASIC 有类似之处, 因此这类器件通常也被称为可编程 ASIC。

2. 半定制或全定制 ASIC

基于 EDA 设计技术的半定制或全定制 ASIC, 根据它们的实现工艺, 可统称为掩模 (Mask) ASIC, 或直接称 ASIC。可编程 ASIC 与掩模 ASIC 相比, 不同之处在于前者具有面向用户的灵活多样的可编程性。掩模 ASIC 大致分为门阵列 ASIC、标准单元 ASIC 和全定制 ASIC。