

Verilog HDL 程序设计教程

王金明 编著

徐志军 主审

人民邮电出版社

图书在版编目 (CIP) 数据

Verilog HDL 程序设计教程 / 王金明编著. —北京: 人民邮电出版社, 2004.1
ISBN 7-115-11939-2

I. V... II. 王... III. 硬件描述语言, VHDL—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 102959 号

Verilog HDL 程序设计教程

- ◆ 编 著 王金明
主 审 徐志军
责任编辑 杨 凌

- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67129258
北京汉魂图文设计有限公司制作
北京隆昌伟业印刷有限公司印刷
新华书店总店北京发行所经销

- ◆ 开本: 787×1092 1/16
印张: 20.25
字数: 484 千字 2004 年 1 月第 1 版
印数: 4 501 - 5 500 册 2005 年 1 月北京第 2 次印刷

ISBN 7-115-11939-2/TN · 2224

定价: 34.00 元 (附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

内 容 提 要

本书对 Verilog HDL 程序设计作了系统全面的介绍，以可综合的设计为重点，同时对仿真和模拟也作了深入的阐述。

本书以 Verilog-1995 标准为基础，全面介绍了 Verilog HDL 的词法、语法、语句，可综合程序的编写，仿真程序的编写，一般数字逻辑的实现，复杂逻辑和算法的实现等，讨论了设计中的方法与技巧，并以大量经过验证的设计实例为依据，深入浅出地阐述了 Verilog 程序开发所涉及的各个方面。对 Verilog-2001 的新增语法结构也作了介绍。

着眼于实用是本书的出发点，由于 HDL 语言的学习与使用必须依托一定的 EDA 环境，因此对典型 EDA 软件的使用与接口也作了介绍。

本书可作为电子工程、通信工程及相关专业高年级本科生和研究生教学用书，也可供从事电路设计和系统开发的工程设计人员阅读参考。

前 言

在现代电子设计领域，EDA 技术已成为每个工程师和设计师必须掌握的工具，起着越来越重要的作用。从 PCB 设计到芯片版图制作，从原理图输入到 HDL 语言描述，从 FPGA 设计到 ASIC 实现，从综合到仿真等等，这些功能的实现，无不依靠 EDA 技术提供强有力的支持。

EDA 技术的发展使得现代电子系统设计从设计思想、设计工具一直到实现方式都产生了深刻的变化，因此，怎样在当前的 EDA 教学中体现出这些变化，抓住 EDA 技术的实质，则是每一个从事 EDA 教学的老师所面临和思考的问题，其根本目的是在有限的学时内将 EDA 技术最本质的内容作为讲授的重点传达给学生，使其能够掌握 EDA 技术的精髓。

HDL 语言是 EDA 技术中重要的一个方面，作者认为，要学习和掌握 EDA 技术，必须至少熟练掌握一种 HDL 语言，其中 Verilog HDL 和 VHDL 显然是首选，这两种语言是已成为 IEEE 标准的硬件描述语言，并各有特色。Verilog 语言是在 C 语言的基础上发展而来的硬件描述语言，具有简洁、高效、易用的特点，是使用最广泛的硬件设计语言之一。在 ASIC 设计领域，Verilog 语言已成为了事实上的标准。IEEE 先后推出了两个 Verilog 标准：IEEE Std. 1364-1995 (Verilog-1995) 和 IEEE Std. 1364-2001 (Verilog-2001)，后者在前者的基础上对 Verilog 语言做了若干改进和扩充，使其功能更强、使用更方便。本书以讲授 Verilog-1995 为主，系统介绍 Verilog 语言的词法、语法、语句，以可综合的设计为重点，同时兼顾仿真和模拟，通过大量的例程由浅入深地展示了 Verilog HDL 程序设计的方法与技巧。对于 Verilog-2001，由于目前的综合和仿真工具还没有全面的支持，因此只作了比较简单的介绍。

EDA 软件工具也是设计开发中重要的一环，优秀的 EDA 工具为 HDL 语言的学习和开发提供了理想的平台，它将设计者的思想高效、高质量地转化为物理电路，并以直观、便捷的形式提供了仿真模拟的手段。脱离了 EDA 工具的支持，HDL 语言就失去了它的意义，基于此考虑，本书用一定的篇幅介绍了典型 EDA 工具的使用方法，将 Verilog 程序设计和典型 EDA 设计工具的使用相结合，并尽量给出综合或仿真的结果，以便于对照。

本书所附的光盘包括了书中所有实例的源代码，这些实例全部通过了综合或仿真，所用软件为仿真软件 ModelSim、综合软件 Synplify Pro/Synplify 以及集成工具 MAX+PLUS II 等。如将这些程序移植到其他综合器和仿真器平台上，由于 Verilog 语言的标准性，同样会得出相同或相似的综合和仿真结果。

本书共分 13 章。

第 1 章对 EDA 技术的发展与特点作了综述，解释了有关的概念。第 2 章介绍流行的 EDA 软件和典型的基于 FPGA 的 EDA 设计流程。

第 3 章以完整的 Verilog 设计过程为例展示了 Verilog 程序设计的全过程，力图使读者快速建立 HDL 设计的整体印象。在第 4、5、6 章中详细介绍了 Verilog 的词法、语法、语句、模块等内容，同时对 Verilog-2001 中新增的语法结构作了简介。第 7 章介绍 Verilog 程序设计

的风格，包括结构描述、行为描述和数据流描述三种方式。

第 8 章是有关仿真的内容，包括系统任务和函数、UDP 元件、延时模型等，然后是电路仿真的具体实例。第 9 章列举了常用数字电路的 Verilog 描述方法。第 10 章对 Verilog 程序设计的方法与技巧进行了探讨和总结。第 11 章和第 12 章是典型数字电路的设计举例，以及一些较复杂数字逻辑和算法的实现方法。第 13 章详细介绍了典型 EDA 软件的使用方法和相互间的接口。

写作本书的目的在于将作者学习使用 Verilog 语言的心得体会以及一些积累与读者共享。假如本书对您学习使用 Verilog 语言能有所帮助的话，则是作者最大的荣幸。

徐志军教授对全书进行了审阅，并提出了修改意见，张雄伟教授也给予了具体建议，杭州电子工业学院的潘松老师和黄继业老师提供了诸多帮助，作者在此对他们一并表示诚挚的谢意。

此外，还要感谢 Altera 公司中国区项目经理徐平波先生和骏龙公司曾雪峰工程师对作者的大力支持。

由于水平所限，加上时间仓促，书中错误与疏漏之处在所难免，真诚希望同行和广大读者给予批评指正。

电子邮件地址：wjm_ice@sina.com。

作者

2003 年 10 月于南京

目 录

第 1 章 EDA 技术综述	1
本章内容简介	1
1.1 引言	1
1.2 EDA 技术及其发展	2
1.3 设计方法与设计技术	3
1.3.1 Top-down 设计	3
1.3.2 Bottom-up 设计	5
1.3.3 IP 复用技术与 SOC	5
1.4 EDA 设计的实现	6
1.5 硬件描述语言	7
思考与练习	9
第 2 章 EDA 设计软件与设计流程	10
本章内容简介	10
2.1 EDA 软件工具概述	10
2.1.1 集成的 CPLD/FPGA 开发工具	10
2.1.2 输入工具 (Design Input Tools)	11
2.1.3 逻辑综合工具 (Synthesis Tools)	12
2.1.4 仿真工具 (Simulation Tools)	12
2.1.5 IC 版图工具	13
2.1.6 其他 EDA 工具	14
2.2 EDA 设计的流程	14
2.2.1 输入 (Design Input)	14
2.2.2 综合 (Synthesis)	15
2.2.3 适配 (Fitter)	16
2.2.4 仿真 (Simulation)	16
2.2.5 编程 (Program)	17
思考与练习	17
第 3 章 Verilog HDL 设计初步	18

本章内容简介	18
3.1 Verilog 语言的历史及与 C 语言的比较	18
3.2 完整的 Verilog HDL 设计	20
3.2.1 4 位全加器和 4 位计数器	20
3.2.2 综合	21
3.2.3 仿真	22
3.3 Verilog 模块基本结构剖析	27
3.3.1 Verilog 模块的结构	27
3.3.2 逻辑功能定义	29
思考与练习	30
第 4 章 Verilog HDL 语言要素	32
本章内容简介	32
4.1 词法	32
4.1.1 空白符和注释	32
4.1.2 数字与字符串 (Numbers & Strings)	33
4.1.3 标识符 (Identifiers)	35
4.1.4 运算符 (Operators)	36
4.1.5 关键字 (Keywords)	36
4.2 数据类型	36
4.2.1 连线型 (Net Type)	36
4.2.2 寄存器型 (Register Type)	38
4.2.3 parameter	38
4.3 寄存器和存储器	39
4.3.1 寄存器	39
4.3.2 存储器	39
4.4 运算符	40
4.4.1 运算符	40
4.4.2 运算符的优先级	44
思考与练习	44
第 5 章 Verilog HDL 行为语句	45
本章内容简介	45
5.1 概述	45
5.2 过程语句	46
5.2.1 always 过程语句	46
5.2.2 initial 语句	50
5.3 块语句	51
5.3.1 串行块 begin-end	51

5.3.2 并行块 fork-join	52
5.4 赋值语句	53
5.4.1 持续赋值与过程赋值	53
5.4.2 阻塞赋值与非阻塞赋值	55
5.5 条件语句	56
5.5.1 if-else 语句	56
5.5.2 case 语句	58
5.5.3 条件语句使用要点	60
5.6 循环语句	61
5.6.1 for 语句	61
5.6.2 repeat 语句	62
5.6.3 while 和 forever 语句	63
5.7 编译向导	65
5.7.1 宏替换 `define	66
5.7.2 文件包含 `include	66
5.7.3 条件编译 `ifdef、`else、`endif	67
思考与练习	68
第 6 章 进程、任务与函数	69
本章内容简介	69
6.1 进程 (process)	69
6.1.1 进程	69
6.1.2 进程间的通信	70
6.2 任务 (task)	71
6.3 函数 (function)	74
6.3.1 函数	74
6.3.2 任务与函数的区别	79
6.4 顺序执行与并发执行	79
6.5 Verilog-2001 新增语法结构简介	81
思考与练习	84
第 7 章 Verilog HDL 的描述风格	85
本章内容简介	85
7.1 概述	85
7.2 结构描述	86
7.2.1 Verilog HDL 内置门元件	86
7.2.2 门级结构描述	89
7.3 行为描述方式	90
7.4 数据流描述方式	91

7.5 不同描述风格的例子	93
7.5.1 半加器	93
7.5.2 1 位全加器	95
7.5.3 4 位全加器	98
思考与练习	100
第 8 章 仿真	101
本章内容简介	101
8.1 概述	101
8.2 系统任务与系统函数	102
8.2.1 \$display 与 \$write	102
8.2.2 \$monitor 与 \$strobe	103
8.2.3 \$time 与 \$realtime	104
8.2.4 \$finish 与 \$stop	105
8.2.5 \$readmemh 与 \$readmemb	105
8.2.6 \$random	106
8.2.7 文件输出	106
8.3 用户自定义元件 (UDP)	107
8.3.1 组合电路 UDP 元件	107
8.3.2 时序逻辑 UDP 元件	110
8.3.3 UDP 元件缩记符	111
8.4 延时模型的表示	112
8.4.1 时间标尺定义 `timescale	112
8.4.2 延时的表示方法	113
8.4.3 延时说明块 (specify 块)	114
8.5 测试平台 (Test Bench)	114
8.5.1 Test Bench	114
8.5.2 测试程序的编写	116
8.6 仿真示例	119
8.6.1 组合电路的仿真	119
8.6.2 时序电路的仿真	124
思考与练习	125
第 9 章 Verilog HDL 设计进阶	126
本章内容简介	126
9.1 基本组合电路的设计	126
9.1.1 简单门电路	126
9.1.2 编译码器	129
9.1.3 数据选择器	132

9.1.4 用组合电路实现的 ROM	133
9.2 基本时序电路的设计	134
9.2.1 D 触发器与 JK 触发器	134
9.2.2 锁存器与寄存器	136
9.2.3 计数器	139
9.2.4 ROM/RAM 模块	140
9.2.5 串/并转换	141
9.3 简单的微处理器	142
9.3.1 设计实现	142
9.3.2 仿真	143
9.4 乘累加器 (MAC) 的设计	144
思考与练习	147
第 10 章 设计方法与设计技巧的探讨	148
本章内容简介	148
10.1 可综合的设计技术的讨论	148
10.1.1 可综合的 Verilog HDL 结构	148
10.1.2 可综合设计的要点	150
10.2 流水线设计技术 (Pipeline Design)	151
10.3 资源共享 (Resource Sharing)	155
10.4 有限状态机 (FSM) 设计	157
10.4.1 基于状态机的设计	157
10.4.2 频率计控制器设计举例	160
10.4.3 几点讨论	162
10.5 多层次结构电路的设计	164
10.5.1 图形与文本混合设计	164
10.5.2 文本设计	165
10.6 阻塞与非阻塞赋值使用要点	167
10.6.1 进程的设计	167
10.6.2 阻塞赋值与非阻塞赋值	168
10.6.3 应用要点	171
10.7 片内存储器的使用	172
10.7.1 片内存储器	172
10.7.2 设计举例	173
10.8 如何消除毛刺	175
10.8.1 毛刺的产生	176
10.8.2 毛刺的消除	177
思考与练习	179

第 11 章 Verilog HDL 综合设计实践	180
本章内容简介	180
11.1 数字跑表	180
11.1.1 设计输入与编译	180
11.1.2 仿真	183
11.2 4 位数字频率计	183
11.2.1 功能与原理	183
11.2.2 设计实现	184
11.3 交通灯控制器	187
11.3.1 功能要求	187
11.3.2 设计实现	188
11.4 乐曲演奏电路	191
11.4.1 音调的控制	192
11.4.2 音长的控制	193
11.4.3 乐曲演奏电路源程序	193
11.5 自动售饮料机的设计	197
11.6 实用多功能数字钟	199
11.6.1 功能	200
11.6.2 源程序	200
11.7 计费器设计	205
思考与练习	209
第 12 章 算法与复杂逻辑的实现	210
本章内容简介	210
12.1 加法器设计	210
12.1.1 级连加法器	211
12.1.2 并行加法器	212
12.1.3 超前进位加法器	212
12.1.4 流水线加法器	219
12.2 乘法器设计	219
12.2.1 并行乘法器	219
12.2.2 移位相加乘法器	220
12.2.3 查找表乘法器	220
12.2.4 加法树乘法器	223
12.3 FIR 滤波器的设计	225
12.3.1 FIR 滤波器的结构	225
12.3.2 抽头系数编码	226
12.3.3 源代码及仿真	227

12.4 数字相关器	231
12.5 信道编译码器	233
12.5.1 线性分组码编译码器	233
12.5.2 循环码编译码器	237
12.6 CRC 校验码	241
思考与练习	244
第 13 章 EDA 软件使用指南	245
本章内容简介	245
13.1 概述	245
13.2 Synplify Pro 使用指南	246
13.2.1 Synplify Pro 的性能特点	247
13.2.2 Synplify Pro 使用指南	248
13.2.3 Synplify Pro 与 MAX+PLUS II 的接口	251
13.3 Synplify 使用指南	253
13.3.1 Synplify 的功能特点	253
13.3.2 Synplify 使用举例	254
13.4 ModelSim 使用指南	256
13.4.1 ModelSim 功能仿真	257
13.4.2 ModelSim 时序仿真	262
13.5 ispLEVER 使用指南	264
13.5.1 ispLEVER 的特点	264
13.5.2 基于 ispLEVER 的 Verilog 设计	264
13.6 MAX+PLUS II 使用指南	268
13.7 Quartus II 使用指南	274
13.7.1 创建工程文件	274
13.7.2 编译	278
13.7.3 仿真	281
13.7.4 Synplify Pro 与 Quartus II 的接口	284
13.8 结束语	285
附录 A Verilog HDL (IEEE Std 1364-1995) 关键字	286
附录 B Synplify Pro/Synplify 可综合的 Verilog 结构	287
附录 C MAX+PLUS II 软件支持的 Verilog 结构	288
附录 D 形式化句法定义	289
附录 E 附带光盘内容及说明	304
参考文献	308

第 1 章 EDA 技术综述

本章内容简介

- EDA 技术的发展
- EDA 技术的特点
- Top-down 设计
- Bottom-up 设计
- IP 复用 (IP reuse) 技术
- 系统芯片 (SOC)
- 可编程逻辑器件 (PLD)
- 专用集成电路 (ASIC)
- 硬件描述语言 (HDL)

本章对 EDA 技术进行综述,介绍 EDA 技术的发展及有关的概念。

EDA 技术仍在不断地向前发展,其自动化和智能化程度不断提高。超大规模集成电路集成度和工艺水平的提高,嵌入式系统的开发与应用,对 EDA 技术提出了更高的要求,也促使其不断进步。现有的 HDL 语言还只能提供行为级或功能级的描述,尚无法完成复杂的系统级的抽象描述,人们正在这方面努力寻求突破。

1.1 引 言

我们已经进入了数字化和信息化的时代,其特点是各种数字产品的广泛应用。现代数字产品在性能提高、复杂度增大的同时,其更新换代的步伐也越来越快,实现这种进步的因素在于生产制造技术和电子设计技术的进步。

生产制造技术以微细加工技术为代表,目前已进展到深亚微米阶段,可以在几平方厘米的芯片上集成数千万个晶体管。摩尔曾经对半导体集成技术的发展做出预言:大约每 18 个月芯片的集成度提高 1 倍,功耗下降一半,称为摩尔定律 (Moore's law)。几十年来,集成电路的发展与这个预言基本一致。数字芯片经历了从 SSI、MSI、LSI 到 VLSI,直到现在的系统芯片 (SOC, System On Chip),我们已经能够把一个完整的电子系统集成在一个芯片上。此外,还有一种器件的发明与使用极大地改变了我们设计制作电子系统的方式与方法,这就是可编程逻辑器件 (PLD, Programmable Logic Device)。PLD 器件是 20 世纪 70 年代后发展

起来的一种器件，它经历了从可编程逻辑阵列（PLA，Programmable Logic Array）通用阵列逻辑（GAL，Generic Array Logic）等简单形式到现场可编程门阵列（FPGA，Field Programmable Gate Array）和复杂可编程逻辑器件（CPLD，Complex Programmable Logic Device）等高级形式的发展，它的广泛使用不仅简化了电路设计，降低了成本，提高了系统的可靠性，而且给数字系统的设计方式也带来了革命性的变化。PLD 器件目前仍在朝密度更高、速度更快、功耗更低、功能更强的方向发展。

电子设计技术的发展也是日新月异的，电子系统的设计理念和设计方法在过去的几十年时间里也发生了深刻的变化。从电子 CAD（Computer Aided Design）电子 CAE（Computer Aided Engineering）到电子设计自动化（EDA，Electronic Design Automation），设计的自动化程度越来越高，设计的复杂性也越来越强。

目前，EDA 技术已成为现代电子设计技术的核心，没有 EDA 技术的支持，想要完成超大规模集成电路的设计制造是不可想象的。反过来，生产制造技术的进步又不断对 EDA 技术提出新的要求，促使其不断向前发展。

1.2 EDA 技术及其发展

在现代的电子设计中，EDA 技术已经成为一种普遍的工具。对设计者而言，熟练地掌握 EDA 技术，可以大大提高工作效率，起到事半功倍的效果。

EDA 即电子设计自动化。EDA 技术的发展是以计算机科学、微电子技术的发展为基础并融合了应用电子技术、智能技术以及计算机图形学、拓扑学、计算数学等众多学科的最新成果发展起来的。简单地说，EDA 就是立足于计算机工作平台而开发出来的一整套先进的设计电子系统的软件工具。

一般认为，EDA 技术经历了下面 3 个发展阶段。

1. CAD 阶段

电子 CAD 阶段是 EDA 技术发展的早期阶段。在这个阶段，一方面，计算机的功能还比较有限，个人计算机还没有普及；另一方面，电子设计软件的功能也较弱。人们主要是借助于计算机对所设计电路的性能进行一些模拟和预测；另外，就是完成 PCB 板的布局布线、简单版图的绘制等工作。

2. CAE 阶段

随着集成电路规模的扩大，电子系统设计的逐步复杂，电子 CAD 的工具逐步完善和发展，尤其是人们在设计方法学、设计工具集成化方面取得了长足的进步，就进入了电子 CAE 阶段。在这个阶段，各种单点设计工具、各种设计单元库逐渐完备，并且开始将许多单点工具集成在一起使用，大大提高了工作效率。

3. EDA 阶段

20 世纪 90 年代以来，微电子工艺有了惊人的发展，工艺水平已经达到了深亚微米级，在一个芯片上已经可以集成上百万、上千万乃至上亿个晶体管，芯片的工作速度达到了吉比特级，这样就对电子设计的工具提出了更高的要求，也促进了设计工具的发展。

在今天，EDA 技术已经成为电子设计的普及工具。无论是设计芯片还是设计系统，没有

EDA 工具的支持，都是难以完成的。EDA 工具已经成为设计师必不可少的武器，起着越来越重要的作用。

从发展的过程看，EDA 技术一直是滞后于制造工艺的发展的，它在制造技术的驱动下，不断地向前进步；从长远看，EDA 技术将随着微电子技术、计算机技术的不断发展而发展。

总的来说，EDA 技术将向着智能性更高、功能更强、高层综合的方向发展。另外一个发展方向就是支持软、硬件协同设计。其次，随着芯片规模的扩大，一些新的问题也需要 EDA 工具加以解决。比如，在大型的系统设计中，设计验证工作将变得比设计本身还要艰巨，因此，更好、更快的仿真验证工具也是 EDA 需要加以完善的地方。

EDA 的发展是没有止境的，它将在诸多因素的推动下不断前进。

1.3 设计方法与设计技术

近 20 年来，电子系统的设计方法和设计技术都发生了深刻的变化。在以前，数字系统大多是采用搭积木式的方式设计的，即由一些固定功能的器件加上一定的外围电路构成模块，由这些模块再进一步形成各种功能电路。构成系统的“积木块”是各种标准芯片，如 74/54 系列（TTL）、4000/4500 系列（CMOS）芯片等等，这些芯片的功能是固定的，用户只能根据需要从这些标准器件中选出最适合的，并按照推荐的电路搭成系统。在设计时，几乎没有灵活性可言，设计一个系统所需的芯片种类多且数量大。

PLD 器件和 EDA 技术的出现改变了传统的设计思路，使人们可以通过设计芯片来实现各种不同的功能。新的设计方法能够由设计者自己定义器件的内部逻辑和管脚，将原来由电路板设计完成的工作大部分放在芯片的设计中进行。这样不仅可以通过芯片设计实现各种数字逻辑功能，而且由于管脚定义的灵活性，大大减轻了原理图和印制板设计的工作量和难度，增加了设计的自由度，提高了效率。同时，这种设计减少了所用芯片的种类和数量，缩小了体积，降低了功耗，提高了系统的可靠性。

在基于 EDA 技术的设计中，有两种基本的设计思路，一种是自顶向下的设计思路，一种是自底向上的设计思路。下面分别进行介绍。

1.3.1 Top-down 设计

Top-down 设计，即自顶向下的设计。这种设计方法首先从系统设计入手，在顶层进行功能方框图的划分和结构设计。在功能级进行仿真、纠错，并用硬件描述语言对高层次的系统行为进行描述，然后用综合工具将设计转化为具体门电路网表，其对应的物理实现可以是 PLD 器件或专用集成电路（ASIC）。由于设计的主要仿真和调试过程是在高层次上完成的，这不仅有利于早期发现结构设计上的错误，避免设计工作的浪费，而且也减少了逻辑功能仿真的工作量，提高了设计的一次成功率。

在 Top-down 设计中，将设计分成系统级、功能级、门级、开关级等几个不同的层次，按照自上而下的顺序，在不同的层次上，对系统进行设计与仿真。

图 1.1 是这种设计方式的示意图。如图所示，在 Top-down 的设计过程中，需要有 EDA 工具的支持，有些步骤 EDA 工具可以自动完成，比如综合等，有些步骤 EDA 工具为用户提

供了操作平台。

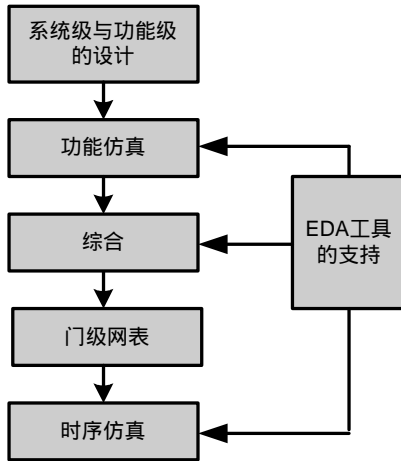


图 1.1 Top-down 设计方式示意图

Top-down 设计须经过“设计—验证—修改设计—再验证”的过程，不断反复，直到得到的结果能够完全实现所要求的逻辑功能，并且在速度、功耗、价格和可靠性方面实现较为合理的平衡为止。不过，这种设计也并非是绝对的，在设计过程中，有时也需要用到自下而上的方法，就是在系统划分和分解的基础上，先进行底层单元设计，然后再逐步向上进行功能块、子系统的设计，直至构成整个的系统。

如图 1.2 所示是用 Top-down 的设计方式设计一个 CPU 的示意图。首先在系统级进行划分，将整个 CPU 划分为几个模块，如 ALU、PC、RAM 模块等，对每个模块再分别进行设计与描述，然后通过 EDA 工具将整个设计综合为门级网表实现之。在设计过程中，需要进行多次仿真和验证，不断修改设计。

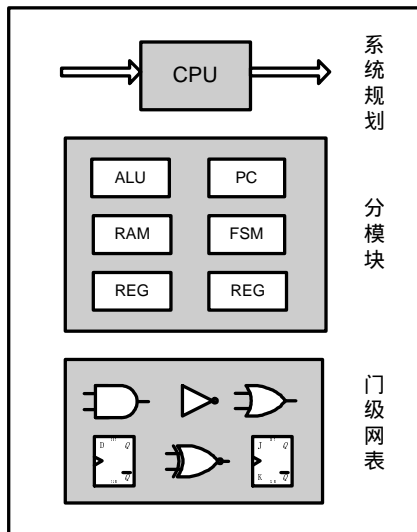


图 1.2 CPU 的 Top-down 设计方式示意图

1.3.2 Bottom-up 设计

Bottom-up 设计，即自底向上的设计，这是一种传统的设计思路。这种设计方式一般是设计者选择标准的集成电路，或者将各种基本单元，如各种门电路以及像加法器、计数器等模块做成基本单元库，调用这些基本单元，逐级向上组合，直到设计出满足自己需要的系统为止。这样的设计方法就如同一砖一瓦建造金字塔，不仅效率低、成本高，而且容易出错。

目前，Top-down 设计更为人们所接受，也为更多的 EDA 工具所支持，这是由于这种设计思想更符合人们逻辑思维的习惯，也容易使设计者对复杂的系统进行合理的划分与不断的优化。而 Bottom-up 设计往往使设计者关注了细节，却对整个的系统缺乏规划，当设计出现问题时，如果要修改的话，就会很麻烦，甚至前功尽弃，不得不从头再来。因此，作为设计者，应该培养 Top-down 的设计习惯。

1.3.3 IP 复用技术与 SOC

当电子系统的设计越来越向高层发展的时候，基于 IP 复用 (IP reuse) 的设计技术越来越显示出其优越性。IP (Intellectual Property)，其原来的含义是指知识产权、著作权等，在 IC 设计领域可将其理解为实现某种功能的设计，IP 核 (IP 模块) 则是指完成某种功能的设计模块。

一般将 IP 核分为硬核、固核和软核 3 种类型。软核指的是在寄存器级或门级对电路功能用 HDL 进行描述，表现为 Verilog HDL 或 VHDL 代码，用户在使用软核的时候可以修改，以满足自己所需要的功能。软核主要用于接口、算法、编译码和加密等模块的设计。

硬核指的是以版图形式描述的设计模块，它基于一定的设计工艺，而且用户不能改动，用户得到的硬核仅是产品的功能，而不是产品的设计。常用的硬核有存储器、模拟器件和一些接口等。固核介于硬核和软核之间，它允许用户重新定义关键的性能参数，内部连线也可以重新优化。

一般软核的开发成本较低，使用灵活，但其可预测性差，延时不一定能达到要求；而硬核的开发成本相对较高，但可预测性强，可靠性高，能够很快地投入使用；固核的性能介于硬核和软核之间。

如图 1.3 所示，由微处理器核 (MPU core)、数字信号处理器核 (DSP core)、存储器 (RAM/ROM) 核、A/D 核、D/A 核以及 USB 接口核等构成了一个系统芯片 (SOC)。

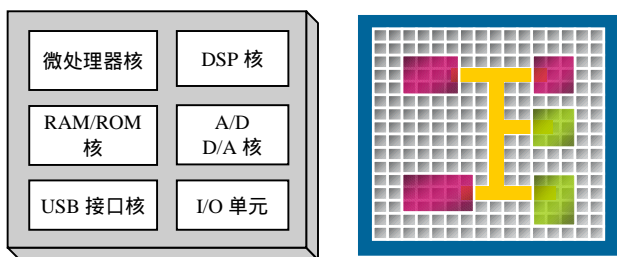


图 1.3 系统芯片 (SOC) 示意图

用户在设计一个系统时，可以自行设计各个功能模块，也可以用 IP 模块来构建。IP 核