

**ELECTRONIC
ENGINEER**

XIDIAN UNIVERSITY PRESS

Basis of IC Design

IC 设计基础

任艳颖 王 彬 编著

*Specially Designed
for Engineers and Technicians of Electronics*



西安电子科技大学出版社

<http://www.xduph.com>

IC 设计基础

任艳颖 王 彬 编著

西安电子科技大学出版社

2003

内 容 简 介

作为 IC 的一名设计者,应当精通电路基础结构、硬件设计语言、EDA 工具、应用协议等知识。本书从工程开发角度出发,结合实际,系统介绍了这些内容,可帮助读者了解标准化设计流程,提高设计技能,跨超芯片设计的门槛。

全书共分 7 章。第 1 章介绍了 IC 设计流程、常用工具的使用、Verilog 设计语言;第 2 章介绍了时序电路的设计;第 3 章对综合工具 DC 进行了说明,并分析了基本语言结构的硬件实现;第 4 章给出了一些常用模块的构造;第 5 章对存储器的结构及设计进行了说明;第 6 章介绍了图像视频芯片的设计;第 7 章介绍了 CISC 处理器和 RISC 处理器的设计。本书每章之后附有练习,便于读者检测掌握的程度。

突出应用,强调由电路结构学习设计语言、强调思维方式“硬件化”、强调标准化的设计风格,是本书最大的特色。书中的代码与脚本都经过精心挑选,具有典型性,读者可在实践中借鉴。

本书可作为 IC 设计培训教材,也可作为有关技术人员的参考书。

图书在版编目 (CIP) 数据

IC 设计基础 / 任艳颖, 王彬编著. —西安: 西安电子科技大学出版社, 2003.5

ISBN 7-5606-1217-2

I. I… II. ①任… ②王… III. 集成电路—电路设计 IV. TN402

中国版本图书馆 CIP 数据核字 (2003) 第 016303 号

策 划 戚文艳 陈宇光

责任编辑 戚文艳

出版发行 西安电子科技大学出版社 (西安市太白南路 2 号)

电 话 (029)8242885 8201467 邮 编 710071

<http://www.xduph.com> E-mail: xdupfb@pub.xaonline.com

经 销 新华书店

印刷单位 西安文化彩印厂

版 次 2003 年 5 月第 1 版 2003 年 5 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 23

字 数 541 千字

印 数 1~4 000 册

定 价 32.00 元

ISBN 7-5606-1217-2 / TN·0217

XDUP 1488001-1

*** 如有印装问题可调换 ***

序 言

从沙子中提出高纯度的硅，制成芯片，单位价值可超过黄金，从这个意义上讲，集成电路设计是真正的点石成金术。

作为通信、计算机、多媒体等众多电子信息产品的核心，集成电路的发展水平对一个国家的科技实力有重大影响。集成电路水平最高的是美国，其次是日本。与它们相比，我们的差距不小，要迎头赶上。近年来，国内集成电路水平得到长足发展，特别是随着一批晶圆厂的开工，直接带动了上游设计公司的发展。这个行业的产值、设计能力、从业人员的规模，都较以往有了质的飞跃。就生产能力而言，我们与国外的差距已大大缩小，因而低下的设计能力已成为急需克服的瓶颈。

由于历史原因，我国集成电路的发展曾一度停滞。对于 40 年来一直按照摩尔定理突飞猛进的集成电路产业来说，这段停滞期使我们远远落后了。这一点，设计与高等学校相关院系的感受尤为明显：由于设计水平低，技术积累匮乏，我们的设计公司只能进行一些低层次的重复开发，大都仅限于几万门的设计；由于人才出现断层，作为人才高地的高校无法提供足够的培训和教育资源。提高设计者的水平非一日之功，提供满足各层次读者需求的教材却是当务之急。现在虽然已经有一些论述集成电路设计的书籍问世，但大都侧重于 VHDL/Verilog 语言的介绍，或者 FPGA 工具的解释，或者半导体集成电路理论的阐述，从工程角度出发，展示集成电路设计的实际过程及经验的书籍较少。这种状况，显然不能满足实际需求。

本书结合工程实践，向读者展示了集成电路，特别是专用集成电路设计的全过程，内容涵盖电路基础、EDA 工具、设计语言直至特定应用的规划。值得一提的是，作者将自己的经验和体会融入本书中，相信会让读者耳目一新。

郑世宝
于上海交通大学
2003 年 4 月

前 言

能够亲眼见证自己的成果为千家万户所用，从这个意义上讲，IC 设计是一个很有意思的行业。不过，IC 设计也是一个充满挑战的领域。从数万个到数百万个乃至上亿个元器件设计规模、千差万别的应用、错综复杂的构造、日新月异的进步，要求设计者十八般武艺样样皆通。IC 工程师是一种典型的“杂食动物”，既要了解 CPU 的构造、SDRAM/EEPROM/闪存的控制、时钟、总线、射频、版图这样的电路知识，又需要熟悉 Verilog/VHDL 等硬件设计语言，还要掌握主流的综合工具、仿真工具、静态时序分析工具，而更重要的是还要对所做的应用有透彻的了解。比如，要做蓝牙基带芯片，就必须熟悉蓝牙的协议；要做非接触式 IC 卡，就必须熟悉 14443 标准；要做数字电视芯片，就必须熟悉 MPEG/DVB 标准。掌握这些知识，不是一日之功（而且，别忘了，这些东西天天在发展，你也得天天补充新知识），修得正果，非得经过八十一难不可。

另一方面，这个领域的初学者往往得不到有效的帮助。国内产业化的 IC 设计刚刚起步，可供学习的资源匮乏，加之许多有经验的工程师不愿意将自己的经验拿出来共享，所以，在成为合格的 IC 工程师之前，许多人都会经历一个痛苦期：不知道从何着手，不知道该学些什么，得不到指导，无法增长实际经验。

电路结构、设计语言、开发工具、应用这几个方面的知识，是初学者必须掌握的。这些要靠多实践，才能真正掌握。最好是有高手带着，参与实际项目的开发。几个项目做下来，就小成了。但良师难求，所以我们期望本书能够扮演这种指导者的角色，帮助初学者顺利跨过最初的几步。因此，我们无意于将它写成一本 Verilog 的语法书，或者某种工具的说明，或者半导体集成电路教材，而是按照工程上开发芯片的要求，介绍必须掌握的最重要的知识。

本书的第 1 章是 IC 设计基础。这一章给读者提供电路基础与工具软件方面的知识，介绍了实际芯片设计的流程，包括规格定义、架构选择、时序规划等重要内容。在 IC 设计中，常用 FPGA 进行原型机验证，所以本章也对 FPGA 的结构进行了说明。本章对一些常用工具软件的使用进行了说明（包括 Altera 的 quartus 与 xilinx 的 ISE，仿真工具 ModelSim 与 NC simulator，综合工具 FPGA Express 等）。此外，还对应用最广的硬件设计语言 Verilog 进行了介绍。由于本书并不是一本 Verilog 语法书，所以我们只写了那些在设计与仿真时最常用的东西。

第 2 章是时序电路设计，对时钟策略、时钟偏移、总线、同步等重要概念进行了阐述。

综合是 IC 设计很重要的一步。对电路性能有重要影响。在第 3 章中，我们着重阐述了最流行的综合工具 DC Compiler 中一些重要概念、使用方法、脚本文件的书写等。本章中还列出了一些基本的程序描述与相应的电路结构。希望大家能够好好研究这个列表，弄清楚所写的程序到底会综合出什么结果。

第 4 章介绍了一些重要基本模块的实现，包括数学运算单元、编码器解码器、存储控制器、异步传输收发器、CRC 等内容。掌握了这些基本单元，就可以用搭积木的方法来完成较大的设计。

存储器的知识是 IC 设计者必须掌握的。在第 5 章中，我们对各种存储器的结构进行了阐述，并对模拟电路的设计进行了说明和演示。

接下来，我们讲述了实际芯片的开发。第 6 章介绍了视频芯片的设计，第 7 章介绍了微处理器的设计。读者可以通过这两章了解开发应用芯片的步骤和方法。视频与微处理器的设计都非常有意思，相信许多人都会感兴趣。

由于篇幅限制，我们只是将那些最重要的东西，结合自己的体会进行了论述，希望读者在本书帮助下，能够顺利地开始自己的 IC 设计历程。

本书错误与不足之处，欢迎大家批评指正。

作者

2003 年 4 月

目 录

第 1 章 IC 设计基础.....	1	2.2.1 全局时钟	86
1.1 系统设计流程	1	2.2.2 门控时钟	86
1.2 ASIC 设计流程	3	2.2.3 行波时钟	87
1.2.1 规格定义	4	2.2.4 时钟偏移	87
1.2.2 工艺选择	6	2.2.5 系统级的同步: 锁相环	90
1.2.3 架构选择	6	2.3 时序	92
1.2.4 电路设计	10	2.3.1 时序图	92
1.2.5 设计验证	41	2.3.2 建立时间/保持时间	93
1.2.6 测试	42	2.3.3 静态时序分析中的概念	94
1.3 FPGA 的设计	42	2.4 总线设计	95
1.3.1 FPGA 中逻辑实现原理	43	2.4.1 总线宽度	95
1.3.2 Altera 的 FPGA	43	2.4.2 总线时钟	95
1.3.3 Xilinx 的 FPGA	49	2.4.3 总线仲裁	96
1.4 常用软件的使用	51	2.4.4 总线操作	97
1.4.1 常用软件的分​​类	51	2.5 练习	97
1.4.2 Chronology Timing Designer 的使用	52	第 3 章 综合	98
1.4.3 ModelSim 的使用	53	3.1 综合 (Synthesis) 的概念	98
1.4.4 NC Simulator 的使用	58	3.2 Design Compiler 简介	99
1.4.5 FPGA Express 的使用	59	3.3 综合条件的设置	102
1.4.6 Silicon Ensemble 的使用	59	3.3.1 操作环境	102
1.5 Verilog	60	3.3.2 导线负载模型	103
1.5.1 Verilog 语言基础	60	3.3.3 设计约束	104
1.5.2 基本概念	61	3.3.4 设计规则约束	107
1.5.3 设计仿真	65	3.3.5 其它	108
1.5.4 系统任务及函数	73	3.4 综合过程示例	109
1.5.5 其它重要的内容	80	3.5 综合的 SDF 文件	111
1.6 练习	83	3.6 关于测试	113
第 2 章 时序电路的设计	84	3.7 面向综合的设计	115
2.1 时序逻辑电路	84	3.7.1 速度与面积的优化: 16 位桶形移位 寄存器	121
2.1.1 双稳态电路	85	3.7.2 Net 类型与 Register 类型	124
2.1.2 单稳态电路	85	3.7.3 if 语句和 Case 语句的综合	124
2.1.3 无稳态电路	85	3.7.4 阻塞赋值与非阻塞赋值	127
2.1.4 施密特触发器	85	3.7.5 状态机的编码	128
2.2 时钟策略	85		

3.7.6 使用流水线	131	4.14 FIR 滤波器	244
3.7.7 设计中不期望的锁存器	131	4.15 练习	245
3.7.8 对可综合设计的一些建议	135	第 5 章 存储器的结构和设计	246
3.8 基本设计单元的综合	137	5.1 基础知识	246
3.9 静态时序分析	151	5.1.1 存储机制及存储器类型	246
3.10 练习	151	5.1.2 SRAM	247
第 4 章 基本模块的设计	152	5.1.3 DRAM	250
4.1 差错控制编码	152	5.1.4 FIFO	257
4.1.1 奇偶校验模块	152	5.1.5 移位寄存器	257
4.1.2 汉明码编解码器	154	5.1.6 CAM	257
4.1.3 CRC 码	162	5.1.7 ROM	257
4.2 基本数学逻辑	166	5.1.8 PROM	258
4.2.1 加法器	166	5.1.9 NVRWM	258
4.2.2 乘法器	174	5.2 HSPICE 介绍	260
4.2.3 除法器	177	5.2.1 电路设计中常见的分析类型	260
4.2.4 算术逻辑单元 ALU	177	5.2.2 HSPICE 基础知识	262
4.3 线性反馈移位寄存器	179	5.3 存储器设计	264
4.3.1 串/并转换模块的功能	179	5.4 练习	268
4.3.2 生成伪随机数	182	第 6 章 图像与视频芯片的设计	269
4.3.3 产生定时标志信号	184	6.1 色度空间转换器	270
4.4 桶形移位寄存器	187	6.1.1 亮度信号和色差信号	270
4.5 串/并转换模块	192	6.1.2 RGB-YCbCr 的模块设计	271
4.6 加解密模块	194	6.1.3 YCbCr-RGB 的模块设计	271
4.6.1 简单加密模块	195	6.2 DCT(离散余弦变换)	272
4.6.2 DES 加密	197	6.2.1 DCT 原理	272
4.6.3 其它加密	206	6.2.2 DCT 模块设计	273
4.7 信源编码	207	6.3 zigzag 扫描	275
4.8 RAM 存储器	209	6.3.1 zigzag 概念	275
4.8.1 RAM 的设计	209	6.3.2 zigzag 模块设计	275
4.8.2 双端口 RAM	211	6.4 量化	277
4.9 DRAM 控制器	213	6.4.1 量化的概念	277
4.10 SRAM 控制器	221	6.4.2 量化模块设计	277
4.11 异步 FIFO	224	6.5 霍夫曼编码/解码	277
4.12 数字锁相环	229	6.5.1 霍夫曼码原理	277
4.12.1 简单的数字锁相环	230	6.5.2 霍夫曼码编码/解码模块设计	279
4.12.2 较复杂的锁相环	234	6.6 JPEG	280
4.13 UART (通用异步收发器)	235	6.7 MPEG	281
4.13.1 简单的 UART	236	6.8 VGA 控制器	282
4.13.2 复杂的 UART	238	6.8.1 视频基础知识	282

6.8.2	VGA 控制器的设计	284	7.7.1	8051 的存储结构	328
6.9	练习	291	7.7.2	8051 的指令集	329
第 7 章	CPU 的设计	292	7.8	8051 的设计	332
7.1	基础知识	292	7.8.1	设计要求	332
7.2	8 位 RISC 的设计	294	7.8.2	架构规划	333
7.3	8 位 CPU 的扩展	298	7.8.3	时序规划	334
7.4	16 位 RISC 的设计	303	7.8.4	低功耗方式和时钟生成模块	335
7.4.1	架构	303	7.8.5	控制模块的设计	337
7.4.2	数据通路的实现	304	7.8.6	数据通路部分的设计	340
7.4.3	控制器的实现	309	7.8.7	定时器/计数器的设计	344
7.5	商业 RISC 介绍	318	7.8.8	串口的设计	346
7.5.1	ARM 体系结构及实现	318	7.8.9	中断控制系统	354
7.5.2	MIPS 体系结构	325	7.9	练习	355
7.6	RISC 与 CISC	327	参考文献	356	
7.7	8051 基础	328			

第 1 章 IC 设计基础

1.1 系统设计流程

在设计系统时，我们常采用自顶向下的思路。首先，是设计系统级的功能行为，接下来我们要将系统分解为不同的模块，用 HDL 代码实现。然后将代码综合为具体的门级实现。IC(Integrated Circuit, 集成电路)设计中的基本概念如图 1.1 所示。

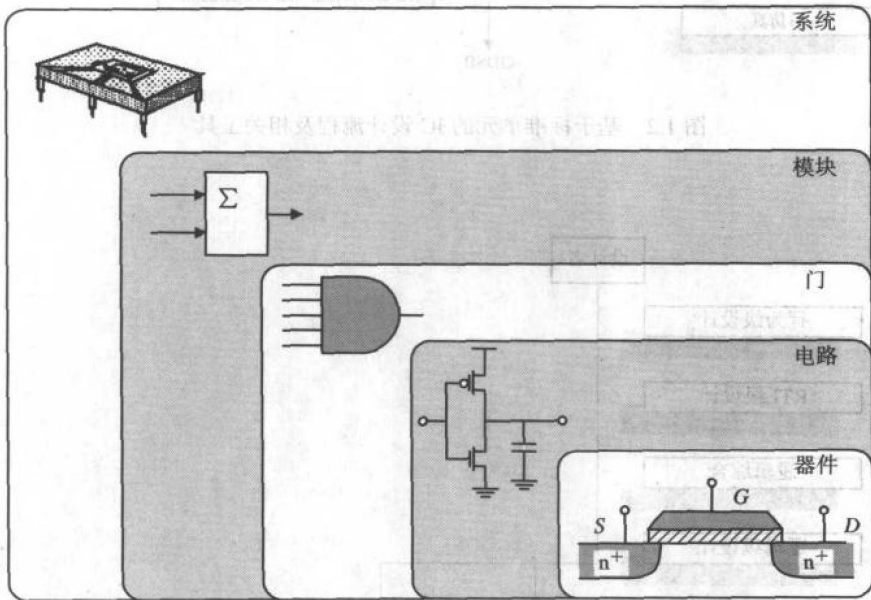


图 1.1 IC 设计中的基本概念

IC 设计分为两类：基于标准单元的设计和基于全定制的设计。图 1.2 是基于标准单元的 IC 设计流程及相关工具。图 1.3 是基于全定制的 IC 设计流程及相关工具。

图 1.4 是一个系统开发流程的例子。

IC 产品的设计过程是：设计者根据设计要求，提出设计构思，并将这个构思逐步细化，直到具体代码实现；再由代码综合出门级网表，生成版图，最终制成产品。在 IC 产品的设计中，好的设计思想价值千金，当然，有了好的设计思想之后，也需要高水平的设计技能来实现。

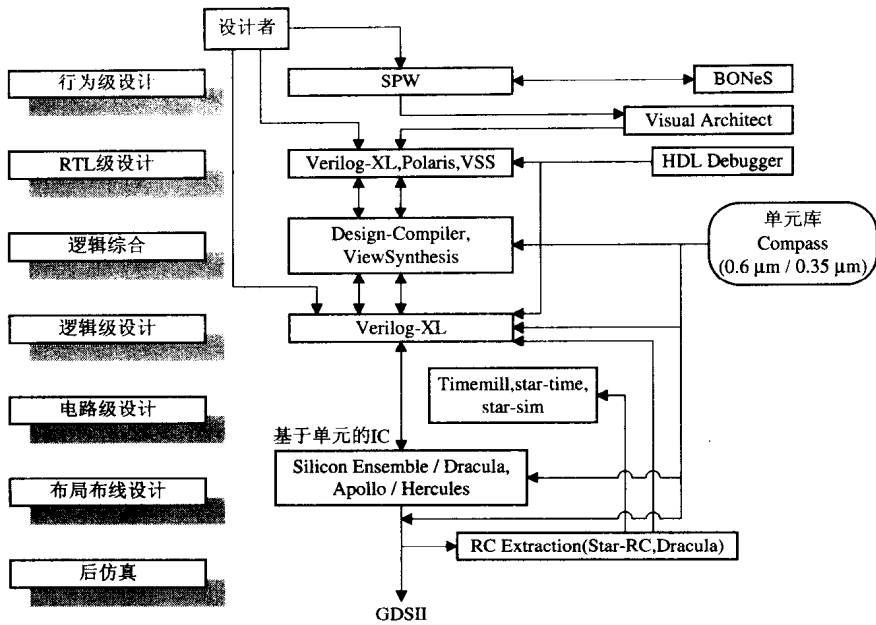


图 1.2 基于标准单元的 IC 设计流程及相关工具

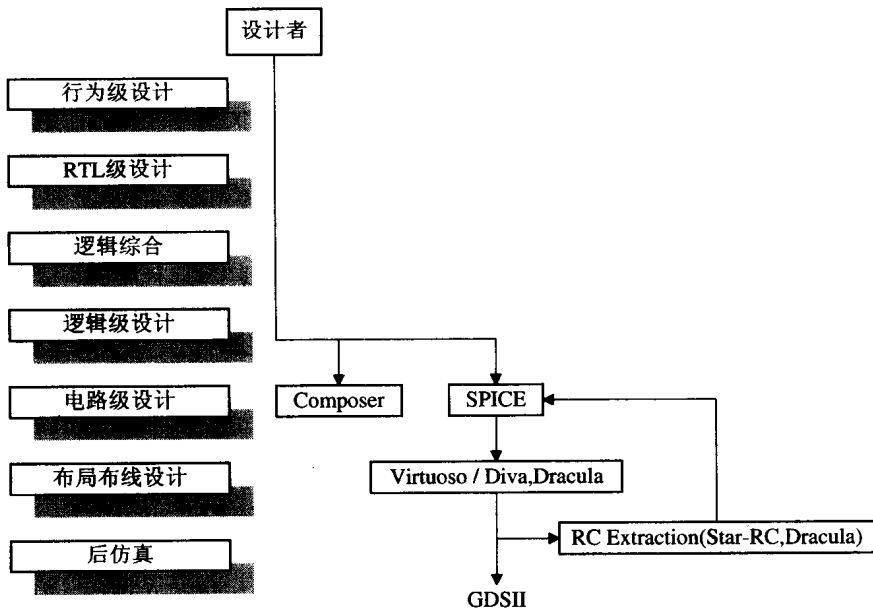


图 1.3 基于全定制 IC 设计流程及相关工具

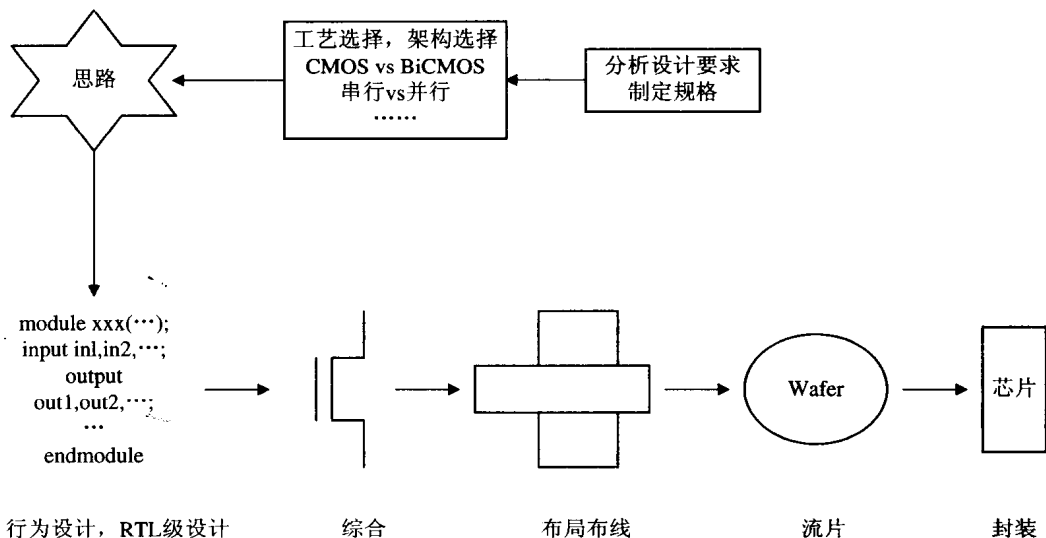


图 1.4 系统开发流程

根据笔者的看法，要成为一个好的 IC 设计师，应具备的条件有：

- 对电路有较好的掌握，并至少熟练掌握一门设计语言(Verilog 或 VHDL)。
- 熟悉工具的使用(仿真工具，综合工具等)。
- 对所做的应用有透彻理解。例如，做蓝牙基带芯片，就要了解蓝牙标准；做非接触 IC 卡，就要了解 14443 标准；做数字电视时，就要了解 MPEG/DVB 标准。

本章对上述三个方面进行了阐述，以后的章节也将围绕这三个方面展开。

笔者推荐使用以下 IC 设计开发工具：

编辑工具：ultraedit32。activeHDL 也不错。

仿真工具：modelsim。在工作站上，则是 Ncsim 最受欢迎。

综合工具：面向 ASIC(专用集成电路)的有 Synopsys 公司的 DC Compiler(已有 NT 版本)；面向 FPGA(现场可编程门阵列)的综合工具有 FPGA Express(Synopsys 已专注于 FPGA Compiler)和 Synplicity 公司的 Synplify。

FPGA 开发工具：Xilinx 公司的产品选用 ISE，Altera 公司的产品选用 Quartus 或 MaxPlus II。

1.2 ASIC 设计流程

ASIC 设计流程如下：

规格定义：规定所需的时钟频率、输入输出时序、功能(输入是什么，要求的输出是什么等等)、面积、功率、信噪比等等。

IC 设计的第一步，是给出设计规格，包括电气规格、功能规格等。电气规格是环境所能容忍的电压范围、直流特性、交流特性等；功能规格是指电路要实现的功能，可以用真值表、状态图、时序图、流程图等表示。此外，还要考虑静电防护(ESD)、I/O 电容、可测

性、封装形式等。

工艺选择：选择哪种工艺(例如，是 CMOS 还是 GaAs，是 0.6 微米还是 0.35 微米)。

架构选择：是并行的还是串行的架构，是否需要流水化结构。

架构要基于设计规格来选择。根据功能要求，设计者要考虑：采用并行设计还是串行设计？设计中各模块都用什么时钟？各时钟的相位有什么关系？是否用流水线？传输信号时是采用差分信号还是单一端口？用什么方式进行补偿？是采用动态逻辑还是静态逻辑？等等。

电路设计：划分模块，定义各模块的功能并定义模块间的连接关系。电路设计一般要从行为级开始，在 RTL 级完成设计。经过综合后，得到门级的网表。

电路仿真：包括功能仿真及时序仿真。在 RTL 级可以进行功能仿真，而时序仿真要在门级才能进行。

布局布线：一般是自动完成的。必要时要进行手工布线。

布局的验证：包括设计规则检查(DRC)等。

布局后的仿真：通常称为后仿真。这个时候可以得到布线的延时，因而时序仿真更接近真实的情况。

可靠性分析：考虑电子迁移/ESD 等。

为了清楚起见，在下面的介绍中，我们将电路仿真、布局布线及布局后的仿真都归到电路设计的范畴。

1.2.1 规格定义

规格定义给出了我们的设计目标，是 IC 设计的第一步。它不仅是设计的依据，也是验证工作的依据。规格定义必须完整、清晰。

1. 规格描述

在具体工程中，我们要写的规格很复杂。下面是工程中要定义的系统规格的框架：

- 系统整体描述
- 管脚封装图
- 架构
- 寄存器说明
- 系统功能描述(模块级)
- 直流特性(DC)
- 交流特性(AC)

我们将在后面的章节给出具体的规格描述示范。这里对 DC 特性及 AC 特性进行一些说明。

2. DC 特性(直流特性)

表 1.1 是 DC 特性描述的一个示例。其中， $V_{CC}=5\text{ V}$ ，温度范围为 -10°C 到 70°C 。

表 1.1 DC 特性

符号	含义	最小值	典型值	最大值	单位	说明
V_{CC}	电源电压	4.5		5.5	V	
V_{IL1}	低电平输入电压			1	V	clk, nwr, nrd
V_{IH1}	高电平输入电压	3.7			V	clk, nwr, nrd
V_{IL2}	Input Low Voltage			1		其它的输入引脚
V_{IH2}	Input High Voltage	3.5			V	其它的输入引脚
V_{OL1}	低电平输出电压			0.45	V	out1, out2 $I_o = -10\text{ mA}$
V_{OH1}	高电平输出电压	4			V	out1, out2 $I_o = -10\text{ mA}$
V_{OL2}	低电平输出电压			0.45	V	out3, out4 $I_o = -1.5\text{ mA}$
V_{OH2}	高电平输入电压	4			V	out3, out4 $I_o = -1.5\text{ mA}$
R_{pull}	上拉电阻	20k		80k	Ω	cs, nrst
I_{IL}	低电平输入泄漏电流			1	μA	$V_{IN} = 0.4\text{ V}$
I_{IH}	高电平输入泄漏电流			1	μA	$V_{IN} = V_{CC} - 0.4\text{ V}$

3. AC 特性(交流特性)

表 1.2 是 AC 特性描述的一个示例，各符号的含义如图 1.5 所示。

表 1.2 AC 特性

符号	最小值	最大值	单位
t_{LHLL}	20		ns
t_{CHWL}	30		ns
t_{AVLL}	8		ns
t_{LLAX}	8		ns
t_{LLWL}	15		ns
t_{RLDV}		100	ns
t_{RHDX}	0		ns
t_{RHDZ}		10	ns
t_{DVWH}	10		ns
t_{WHDX}	5		ns
t_{WLWH}	200		ns
t_{WHWL}	1000		ns
t_{AVWL}	30		ns
t_{WHAX}	5		ns
t_{RHAX}	5		ns

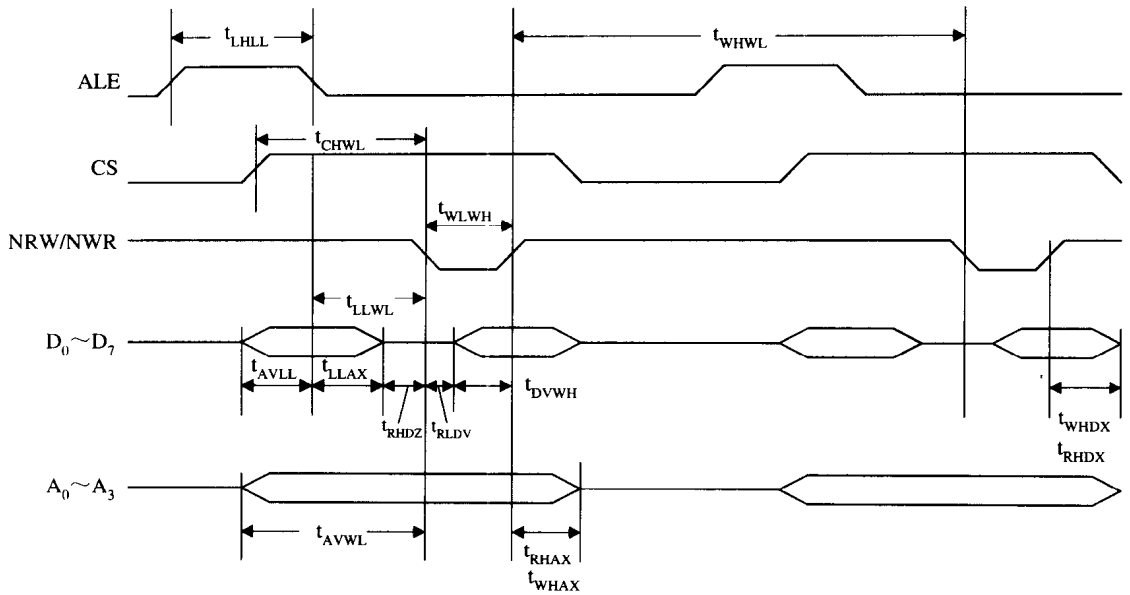


图 1.5 AC 特性

1.2.2 工艺选择

图 1.6 示出了硅工艺类别。

表 1.3 给出了不同工艺间的差别。

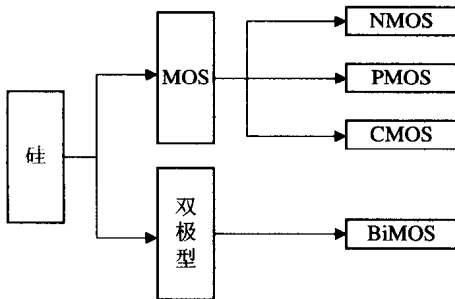


图 1.6 硅工艺类别

表 1.3 不同工艺的选择

	MOS	双极型	GaAs
工作频率	200 MHz	900 MHz	1.8 GHz
驱动能力	低	高	
工作电压	低电压/低阈值 SOI		高电压
类型	数字	模拟	混合
	高精度电容		
设计风格	有可用的单元库		

1.2.3 架构选择

依据系统规范，需要决定设计采用并行还是串行结构，是否需要加流水线等。串/并行结构的特点很好理解：串行结构速度较慢，但面积小；并行结构与之相反。流水线就是在

系统的关键路径上插入寄存器，这样可以使最大时钟频率提高。

1. 并行/串行结构

例如，对一个随机数发生器，我们采用串行设计时，可以采用如图 1.7 所示的结构。

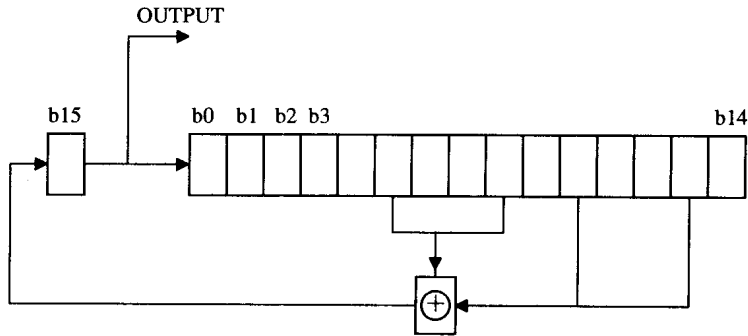


图 1.7 串行的随机数发生器

也可以采用并行的结构来实现随机数发生器。例如，构造一次生成一字节的随机数发生器时，用 8 个移位寄存器，按照与图 1.7 相似的结构并列起来就可以了。并行的随机数发生器的结构图如图 1.8 所示。

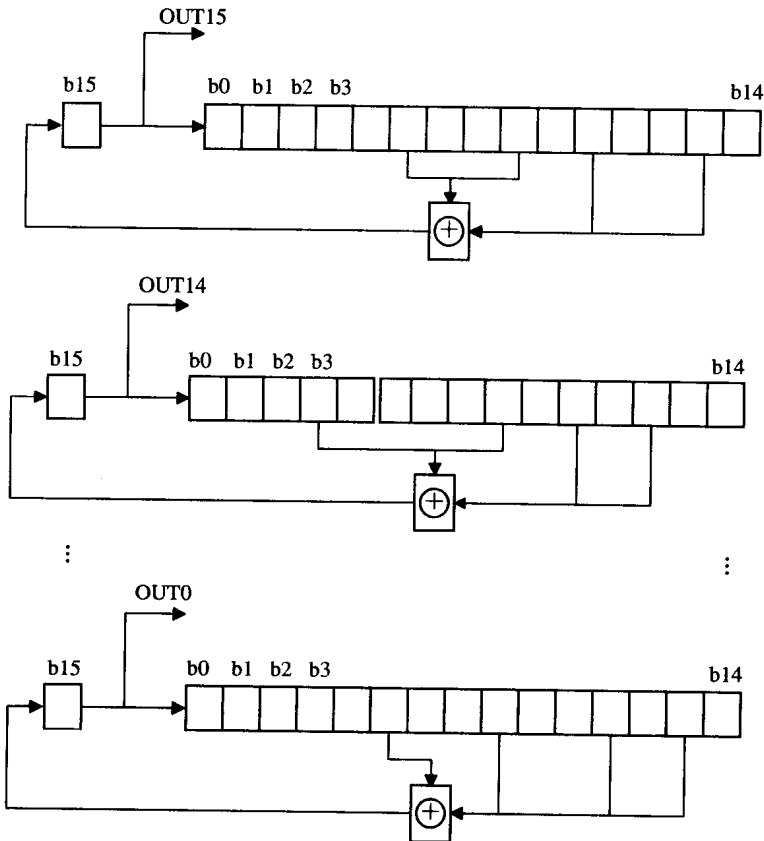


图 1.8 并行随机数发生器

2. 流水线结构(pipeline)

流水线是一种通过增加面积来提高速度的通用的设计技术，在电路的关键路径中插入寄存器，可以提高系统运行速度。图 1.9 的例子说明了此方法的用法。

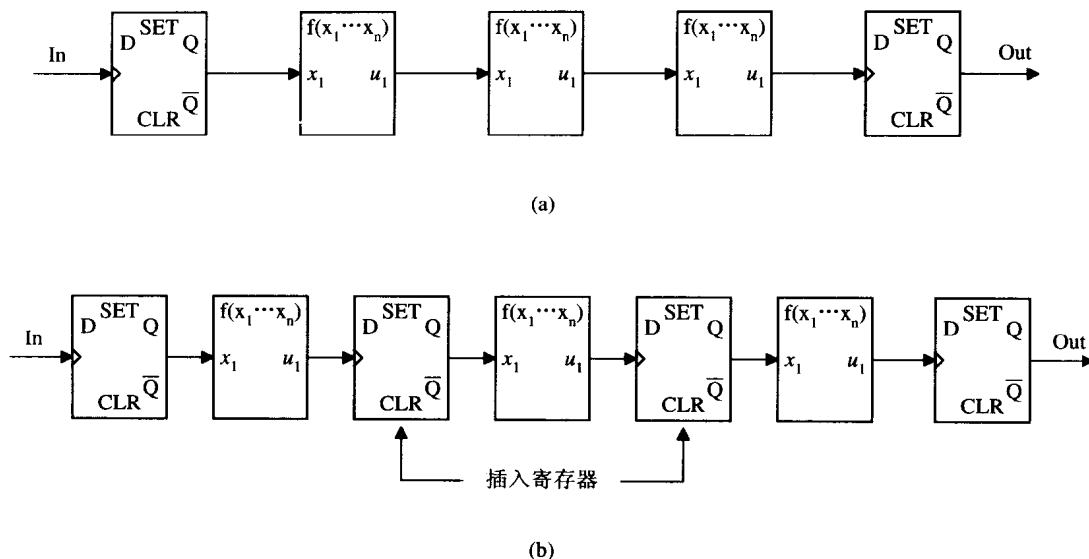


图 1.9 流水线结构
(a) 无流水线; (b) 流水线

对于图 1.9(a)中的结构，有

$$T_{\min} = t_{p,\text{reg}} + t_{p1,\text{logic}} + t_{p2,\text{logic}} + t_{p3,\text{logic}} + t_{\text{setup,reg}}$$

$t_{p,\text{reg}}$ 和 $t_{\text{setup,reg}}$ 是寄存器的传输延时和建立时间。 $t_{p,\text{logic}}$ 代表通过组合网络的最坏情况延时路径的延时。通常这个延时要比寄存器的延时大许多，对电路的性能起主要影响。在图 1.9(a)中，三个组合逻辑的延时之和决定了系统时钟的速度。

流水线是一种打破性能瓶颈的方法。假定我们在三个组合逻辑间引入寄存器，如图 1.9(b)所示，则有：

$$T_{\text{minpipe}} = t_{p,\text{reg}} + \max(t_{p1,\text{logic}}, t_{p2,\text{logic}}, t_{p3,\text{logic}}) + t_{\text{setup,reg}}$$

假定所有的逻辑模块的传输延时大致相同，并且锁存器的延时与组合逻辑延时相比可以忽略，则 $T_{\text{minpipe}} = T_{\min}/3$ 。流水线网络的性能是原始电路性能的 3 倍。

付出的代价是增加两个额外的寄存器，并使输出延迟了两个时钟。这种代价是很小的。所以流水线在高性能数据通路中应用很多。但要注意，只有在关键路径上增加寄存器才有意义。如果寄存器的延时与组合逻辑的延时相当，增加额外的寄存器只会增加硬件过载，并不会获得性能的改善。

在流水线结构中，可以按流水方式工作，即将一个计算任务细分成若干个子任务，每个子任务由专门的部件处理，多个计算任务依次进行并行处理。该方法可以大大提高指令的执行速度。