

嵌入式硬件系统设计与开发系列

SoC 原理、实现与应用

王道宪摇主编

刘摇丽摇编著

内 容 简 介

本书对系统集成芯片的发展进行了详细全面的讲述,主要介绍了在 CAD 工具的平台上进行以系统级设计为核心的系统芯片的设计方法。

本书从基本单元电路设计出发,以 VHDL 语言为基本设计手段,讨论了各种典型的数字集成系统的设计,以及系统芯片实现的两种基本途径:即半定制高密度可编程逻辑器件(CPLD)的实现和全定制的专用集成电路(ASIC)的实现。

全书共分 8 章,主要内容包括集成电路工艺及 CMOS 基本原理、基本数字电路设计、硬件描述语言 VHDL 基本数字逻辑单元的设计、系统集成芯片的层次结构设计、可编程逻辑器件、专用集成电路设计及可测试结构设计,全书语言顺畅,以循序渐进的叙述方法讲述 CMOS 的基本原理和设计实现。

本书对系统芯片原理与设计实现这一十分丰富的数字电路内容进行了详细的叙述,是电路设计和芯片设计专业人员和爱好者值得一看的书,也可作为高校本科生和专科生的教材。

图书在版编目(CIP)数据

CMOS 原理、实现与应用 刘颖编著—北京:国防工业出版社,1999

(嵌入式硬件系统设计与开发系列 刘道宪主编)

国防工业出版社—北京—1999—16 开

I. ①刘… II. ①刘… III. ①集成电路—芯片—设计
IV. ①TN79

中国版本图书馆 CIP 数据核字(1999)第 11111 号

(北京市海淀区紫竹院南路 10 号)

(邮政编码 100088)

北京奥鑫印刷厂印刷

新华书店经售

*

开本 16 开 印张 16 印张 16 千字

1999 年 1 月第 1 版 1999 年 1 月北京第 1 次印刷

印数 1—1000 册 定价 16.00 元

(本书如有印装错误,我社负责调换)

国防书店:(010)61611111 发行邮购:(010)61611111

发行传真:(010)61611111 发行业务:(010)61611111

前 言

随着 VLSI 工艺技术的发展,器件特征尺寸越来越小,芯片规模越来越大,数百万门级的电路可以集成在一个芯片上。多种兼容工艺技术的开发,可以将差别很大的不同种器件在同一个芯片上集成,为系统集成开辟了广阔的工艺技术途径。

真正称得上系统级芯片 (SoC) 集成,不只是把功能复杂的若干个数字逻辑电路放在同一个芯片上,做成一个完整的单片数字系统,而且在芯片上还应包括其他类型的电子功能器件,如模拟器件和专用存储器。在某些应用中,可能还会扩大一些,包括射频器件甚至 MEMS 等。通常,SoC 起码应在单片上包括数字系统和模拟电子器件。

由于单片 SoC 设计在速度、功耗、成本上和多芯片系统相比占有较大的优势,而且电子系统的专用性对不同的应用,要求有专用的系统,因此 SoC 设计在未来的集成电路 (IC) 设计业中将有举足轻重的地位。本文在分析 SoC 特点的基础上讨论单片系统所必需的设计技术以及工艺加工方法。

20 世纪下半叶,微电子技术得到了迅速发展。由于集成电路设计和工艺水平的提高,目前已经可以在一个单片上集成 109 个晶体管,从而有可能将原先许多 IC 组成的电子系统集成在一个单片上,构成所谓的 SoC。与 IC 相比,SoC 不再是一种功能单一的单元电路,而是将信号采集、处理和输出等完整的系统集成在一起,成为一个有专用目的的电子系统单片。

SoC 的出现是电子系统设计领域的一场革命,它对电子信息产业的影响将不亚于 20 世纪 60 年代 IC 的出现所产生的影响。正因为如此,当今电子系统的设计已不再是利用各种通用 IC,进行 PCB 板级的设计和调试,而是转向以 HDPLD 或 ASIC 为物理载体的 SoC 的设计。另一方面,由于 IC 工艺的成熟和 EDA 工具的迅速发展,使得电子系统的设计者并不需要过多地关注半导体集成工艺,完全可以利用现有的成熟工艺,在 EDA 工具的帮助下完成整个系统从行为层到物理层的全部设计,并最终在 HDPLD 上实现,或委托 IC 制造商进行 ASIC 的生产。

综上所述,SoC 的出现是电子设计领域的一场革命,同时也对电子工程技术人员提出了新的挑战。SoC 的设计以 IP 核为基础,以硬件描述语言为主要设计手段,借助于以计算机为平台的 EDA 工具进行,其设计方法和手段都与传统的设计不同,要求电子系统设计者必须掌握新的设计方法和技术。目前,在国外技术发达国家,SoC 是一个发展迅速而又前景良好的新技术。在国内,这方面的研究和开发上处于起步阶段,今后若干年内将会得到迅速发展。

本书是 SoC 原理与设计的中高级读本,实用性强是本书的一个鲜明特色。书中通过大量实例介绍了 SoC 的基本原理、实现和设计,以及对数字电路系统的建模方法。书中使用普通的术语介绍 SoC 芯片系统原理的语义,而没有使用正式定义的专业术语。本书

试图对 SoC 进行完整地讨论，但只限于讨论 SoC 设计中最常用的特性。

SoC 设计复用的关键是模块化。很明显，片上互连（将不同模块连接在一起）是一个越来越重要的问题。单一的片上总线已经不能适应很多应用，因而已使用层次化的总线来提供必要的带宽。目前的片上总线是时钟型的，总线上所有的客户模块必须使用共同的全局时钟来实现接口。设计复杂的 SoC 是当前世界最具有挑战性工程任务之一，这是一项快速发展的学科，其新思想和新方法不断翻新。成功的工程师需要的不仅是现代化知识，工程是一个创造性的学科，在如半导体工业这样高速发展的产业中，最好的工程师确实是非常具有创造性的。每个新的设计都比以前的更复杂，且要求更高的性能、更低的功耗和更多的功能。在如此快速发展的工艺基础上，没有一种常规的设计方法能给出最佳的设计能力。

书中通过大量的实例介绍了 SoC 的基本原理和设计的基本方法，这些实例不仅对读者掌握 SoC 本身和建模方法有很大的帮助，而且对实际数字系统设计也有帮助。

希望此书的出版对推动我国 IC 和 SoC 设计水平的提高有所促进，对高等学校的教学和课程改革有所帮助。由于作者水平有限，加之时间仓促，书中难免存在错误和不足，敬请广大读者予以批评指正。

编 者

目 录

第 1 章 SoC 简介	1
1.1 半导体核心技术	1
1.1.1 SoC 特点	1
1.1.2 SoC 设计技术	1
1.1.3 SoC 的产品效益	3
1.1.4 技术挑战	3
1.1.5 商业挑战	4
1.2 SoC 对产业产生巨大冲击	5
1.2.1 从产品迈向解决方案	5
1.2.2 SoC 设计的平台化	5
1.2.3 系统业者/IC 产品业者分工模式的改变	5
1.2.4 供应链各部门间联盟合作之风盛行	6
1.2.5 晶圆制造的生态变动	6
1.3 SoC 软/硬件协同设计	6
1.3.1 可用性问题	6
1.3.2 SoC 设计过程的质量保证	7
1.3.3 IC 业的虚拟再集成	8
1.4 SoC 功能验证	8
1.4.1 模块为基础 SoC 的验证	9
1.4.2 平台为基础的 SoC 验证	10
1.4.3 传统的验证方式	10
1.4.4 可验证性设计	10
1.5 SoC 与单片机应用技术的发展	11
1.5.1 SoC 技术与应用	11
1.5.2 SoC 的应用	13
1.5.3 SoC 技术中的单片机	13
第 2 章 处理器设计与 ARM SoC 体系结构	16
2.1 处理器设计	16
2.1.1 指令设计	16
2.1.2 处理器设计中的权衡	20
2.2 ARM 指令格式和时序	24
2.2.1 处理器模式	24

2.2.2	寄存器	24
2.2.3	流水线	25
2.2.4	时序	25
2.2.5	指令	26
2.3	精简指令集计算机	35
2.3.1	数据处理指令	37
2.3.2	数据传送指令	41
2.3.3	控制流指令	47
2.3.4	编写简单的汇编语言程序	52
2.3.5	程序设计	53
2.4	低功耗设计	54
2.5	ARM 体系结构	57
2.5.1	Acorn RISC 机器	57
2.5.2	体系结构的继承	58
2.5.3	ARM 编程模型	59
2.5.4	ARM 开发工具	63
第 3 章	ARM SoC 的组织与实现	66
3.1	流水线 ARM 的组织	66
3.1.1	3 级流水线 ARM 的组织	66
3.1.2	5 级流水线 ARM 的组织	69
3.2	ARM 指令执行	72
3.3	ARM 的实现	75
3.4	ARM 协处理器接口	87
3.5	VHDL 语言	89
3.5.1	VHDL 语言简介	89
3.5.2	VHDL 语言基础知识	92
3.5.3	VHDL 基本单元	94
3.5.4	VHDL 语句	98
3.5.5	VHDL 程序举例	99
第 4 章	存储器层次	101
4.1	存储器容量及速度	101
4.2	片上存储器	102
4.3	Cache	102
4.4	Cache 设计示例	108
4.5	存储器管理	108
4.6	存储元件与阵列	112
4.6.1	一般属性	112
4.6.2	锁存器	113
4.6.3	时钟与同步	115

4.6.4	主-从触发器和边沿触发器	116
4.6.5	寄存器	117
4.6.6	随机存取存储器	121
4.6.7	只读存储器	123
第 5 章	体系结构对操作系统的支持	131
5.1	操作系统简介	131
5.1.1	多用户系统	131
5.1.2	存储器管理	132
5.1.3	保护	132
5.1.4	资源分配	132
5.1.5	单用户系统	132
5.1.6	嵌入式系统	133
5.2	ARM 系统控制协处理器	133
5.2.1	CP15 指令	133
5.2.2	保护单元	134
5.2.3	MMU	134
5.3	保护单元寄存器 CP15	134
5.4	ARM 保护单元	136
5.4.1	保护单元的结构	136
5.4.2	区域优先级	136
5.4.3	Harvard 核	137
5.5	CP15 MMU 寄存器	137
5.6	ARM MMU 结构	139
5.6.1	存储器粒度	139
5.6.2	页域	139
5.6.3	转换过程	140
5.6.4	段转换	141
5.6.5	页转换	141
5.6.6	访问权限	143
5.6.7	Cache 和写缓冲控制	144
5.6.8	外部故障	144
5.7	同步	144
5.7.1	互斥	145
5.7.2	SWAP	145
5.8	上下文切换	145
5.8.1	同时切换	146
5.8.2	存储器状态	146
5.8.3	浮点状态	146
5.8.4	转换状态	146

VIII

5.9 输入/输出.....	147
5.9.1 存储器映射的外围设备	147
5.9.2 存储器映射问题	147
5.9.3 直接存储器访问	147
5.9.4 快速中断请求	148
5.9.5 中断延迟	148
5.9.6 Cache 和 I/O 交互作用	148
5.9.7 减小延迟	149
5.9.8 其他 Cache 问题	149
5.9.9 操作系统问题	149
第 6 章 基本数字逻辑单元设计	151
6.1 逻辑电路.....	151
6.1.1 数字逻辑分类	151
6.1.2 组合逻辑电路	151
6.2 组合逻辑电路的设计	153
6.2.1 组合逻辑电路的分析	153
6.2.2 组合逻辑电路的设计	157
6.3 集成逻辑门.....	171
6.3.1 概述	171
6.3.2 发射极耦合逻辑门	175
6.3.3 MOS 逻辑门.....	178
6.3.4 NMOS 逻辑门电路.....	178
6.3.5 PMOS 逻辑门电路	181
6.3.6 CMOS 逻辑电路	183
6.4 不同逻辑系列的配合问题	190
6.4.1 逻辑电平的配合	191
6.4.2 驱动能力的配合	191
6.4.3 各种集成逻辑门性能比较	191
6.5 时序逻辑电路设计	192
6.5.1 基本 RS 触发器	193
6.5.2 锁存器	195
6.5.3 主从 RS 触发器	197
6.5.4 钟控 RS 触发器	198
6.5.5 主从 JK 触发器	200
6.5.6 沿触发 JK F/F	205
6.5.7 D 触发器	207
6.5.8 T 触发器.....	208
6.5.9 触发器逻辑功能的转换	208
第 7 章 SoC 的层次结构设计	212

7.1	SoC 的结构	212
7.1.1	引言	212
7.1.2	系统集成芯片的硬件结构	212
7.1.3	嵌入式软件	214
7.2	数字结构的层次结构设计	215
7.2.1	芯片的划分	215
7.2.2	系统间互连的表示	222
7.3	系统的仿真和测试	231
7.3.1	概述	231
7.3.2	仿真程序的设计方法	231
7.3.3	TEXIO 建立测试程序	235
7.4	SoC 中的嵌埋式精简指令集处理器 RISC	237
7.4.1	概述	237
7.4.2	RISC 的定义与特点	238
7.4.3	RISC 的指令特点	240
7.4.4	RISC 的并行处理技术	242
7.4.5	RISC/DSP 结构	245
7.4.6	RISC 核的设计	249
7.5	SoC 的软/硬件协同设计	250
7.5.1	软/硬件协同设计的概念	250
7.5.2	性能分析	251
7.6	性能评估	251
7.6.1	时间性能估计	252
7.6.2	代价估计	252
7.7	嵌入式实时操作系统 RTOS	253
7.7.1	实时操作系统	253
7.7.2	嵌入式实时操作系统	254
7.7.3	实时多任务高度	255
7.7.4	信号与信号量(semaphore)	256
第 8 章	可编程逻辑器件	258
8.1	概述	258
8.1.1	可编程逻辑器件的发展	258
8.1.2	用户再构造电路和可编程 ASIC 电路	258
8.1.3	可编程逻辑器件的分类	259
8.2	可编程逻辑器件的编程元件	261
8.2.1	熔丝型开关	261
8.2.2	反熔丝开关	261
8.2.3	浮栅编程技术	262
8.3	可编程阵列逻辑 (PAL) 器件	265

8.3.1	现场可编程逻辑阵列(FPLA)器件	265
8.3.2	PAL 器件的基本结构	267
8.3.3	PAL 器件的输出和反馈结构	267
8.3.4	PAL 器件编号与典型 PAL 器件介绍	274
8.3.5	PAL 器件的应用	275
8.4	PAL 与 GAL 器件的电路结构	276
8.4.1	PLD 的电路表示方法	276
8.4.2	PLD 的基本电路结构	280
8.4.3	PAL 器件的电路结构	282
8.4.4	通用阵列逻辑 GAL(Generic Array Logic)	283
8.5	ispLSI 系列 CPLD	288
8.5.1	概述	288
8.5.2	ispLSI1000 系列 CPLD 的结构特点	289
8.5.3	ispLSI CPLD 的测试和编程特性	293
8.5.4	ispLSI 2000 系列的结构	294
8.5.5	ispLSI3000 系列 CPLD	294
8.5.6	ispLSI5000 系列 CPLD	296
8.5.7	ispLSI8000 系列 CPLD 的结构和工作原理	297
8.6	现场可编程门阵列	299
8.6.1	概述	299
8.6.2	XC4000 系列 FPGA 的结构和工作原理	300
8.6.3	Spartan 系列 FPGA	306
8.7	Virtex-II 系列 FPGA 的结构和性能	306
8.7.1	概述	306
8.7.2	Virtex-IIFPGA 的总体结构	308
8.7.3	Virtex-IIFPGA 的可构造逻辑模块	308
8.7.4	18K 位可选择 RAM 模块	310
8.7.5	嵌入式乘法器	310
8.7.6	全局时钟多路缓冲器	310
8.7.7	数字时钟管理器 DCM	311
8.7.8	输入输出模块	313
8.7.9	有源互连技术	314
8.8	基于 HDPLD 的系统设计实现	315
8.8.1	设计实现概述	315
8.8.2	器件的选择	315
8.8.3	HDPLD 的设计流程	316
第 9 章	可测试结构设计	318
9.1	可测试设计的意义	318
9.2	可测性基础	319

9.2.1 故障模型	319
9.2.2 可测性分析	322
9.2.3 测试向量生成	325
9.3 可测性结构设计	325
9.3.1 专门测试设计	326
9.3.2 扫描测试技术	327
9.3.3 内建自测试技术	329
9.3.4 系统测试技术	330
9.4 智能型电子系统设计方法与过程	334
9.4.1 系统设计方法	334
9.4.2 微机应用系统硬件设计与调试原则	337
9.4.3 微机应用系统软件开发	340
9.5 微型计算机应用系统设计	350
9.5.1 微型计算机的系统板组成	351
9.5.2 PC/AT 总线与时序	353
9.5.3 系统存储器空间和 I/O 地址分配	354
9.5.4 PC/AT 系统的 I/O 通道	356
9.5.5 系统接口部件	358
9.5.6 外部设备接口	361
参考文献	370

第 1 章 SoC 简介

1.1 半导体核心技术

系统级芯片 (System on Chip, SoC) 可定义为具备完整系统构架与功能的晶片, 其构架包含可执行控制/运算或信号处理功能的处理器、记忆体、周边电路及系统 IP 特定逻辑电路。SoC 应至少负责终端系统主要功能的一部分, 但仍可与其他微处理器搭配使用, 或外加 DRAM 或类比/混合信号元件等其他元件。

过去, 加工技术是集成电路 (IC) 发展的瓶颈所在, 加工微缩进展所能提供的晶体管门的数目无法满足线路设计的需求。若将许多复杂功能电路强加在单芯片上, 则因硅片面积太大将导致成品率降低。因此将系统中所需的个别功能分别以离散 IC 方式产出, 再在电路板层次进行组合设计。

20 世纪 90 年代后期, 随着半导体加工技术跨入深次微米时代, 加工微缩的结果已可提供晶体管门电路在百万以上的设计和加工能力, 使 SoC 的概念有了实现的可能。

作为 ASIC (Application Specific IC) 设计方法学中的新技术, SoC 始于 20 世纪 90 年代中期。1994 年 Motorola 公司发布的 Flex Core™ 系统 (用来制作基于 68000™ 和 Power PCTM 的定制微处理器) 和 1995 年 LSI Logic 公司为 SONY 公司设计的 SoC, 是基于 IP (Intellectual Property) 核完成 SoC 设计的最早报道。由于 SoC 可以充分利用已有的设计积累, 显著地提高 ASIC 的设计能力, 因此发展非常迅速。

1.1.1 SoC 特点

SoC 是在单片上实现全电子系统的集成, 与 IC 相比, 它不再是一种应用目的的电子系统单片, 而是将信号采集、处理和输出等完整的系统集成在一起, 成为一个有某种应用目的的电子系统单片。电子系统传统设计的方法是在 PCB 级完成的。系统设计人员利用各 IC 制造商生产的通用集成电路, 在 PCB 上构成系统, 系统的调试也在 PCB 上进行。这种开发设计方法要求设计者具有丰富的硬件知识和调试能力, 产品开发周期长, 投资较大, 设计修改困难。此外, 由于 PCB 连线的延时、空间尺度、重量和可靠性的制约, 整机性能受到很大的限制。如果能将整个系统最终集成在一个单片上, 无疑对于提高产品性能、缩小产品体积具有极大的帮助。因而, SoC 是电子系统设计开发的合理选择。

1.1.2 SoC 设计技术

1. 设计再利用

数百万门规模的 SoC 设计, 不能一切从头开始, 要将设计建立在较高的层次上, 更多地采用 IP 复用技术, 只有这样, 才能较快地完成设计, 保证设计成功, 得到低价格的

SoC，满足市场需求。

设计再利用是建立在芯核（core）基础上的，它是将已经验证的各种超级宏单元模块电路制成芯核，以便以后的设计利用。芯核通常分为 3 种，一种称为硬核，具有和特定工艺相联系的物理版图，已被投片测试验证。可被新设计作为特定的功能模块直接调用。第二种是软核，是用硬件描述语言或 C 语言写成，用于功能仿真。第三种是固核（firmcore），是在软核的基础上开发的，是一种可综合的并带有布局规划的软核。

随着工艺技术的发展，深亚微米（DSM）使 SoC 更大更复杂。这种综合方法将遇到新的问题，因为随着工艺向 $0.18\mu\text{m}$ 或更小尺寸发展，需要精确处理的不是门延迟而是互连线延迟。再加之数百兆的时钟频率，信号间时序关系十分严格，因此很难用软的 RTL 综合方法达到设计再利用的目的。

建立在芯核基础上的 SoC 设计，使设计方法从电路设计转向系统设计，设计重心将从今天的逻辑综合、门级布局布线、后模拟转向系统级模拟，软硬件联合仿真，以及若干个芯核组合在一起的物理设计。

2. 低功耗设计

SoC 因为百万门以上的集成度和数百兆时钟频率下工作，将有数十瓦乃至上百瓦的功耗。巨大的功耗给使用封装以及可靠性方面都带来问题，因此降低功耗的设计是 SoC 设计的必然要求。

3. 可测性设计技术

SoC 是将芯核和用户自己定义的逻辑（UDL）一起集成。芯核深埋在芯片中，芯核不能事先测试，只能在 SoC 被制造出来后作为 SoC 的一部分和芯片同时测试。因此对 SoC 测试存在许多困难，首先，芯核是别人的，选用芯核的设计者不一定对芯核十分了解，不具备对芯核的测试知识和能力。再加之芯核深埋在芯片之中，不能用测试单个独立芯核的方法去处理集成后的芯核测试。

4. 深亚微米 SoC 的物理综合

由于深亚微米时互连线延迟是主要延迟因素，而延迟又取决于物理版图。因此，传统的自上而下的设计方法只有在完成物理版图后才知道延迟大小。如果这时才发现时序错误，必须返回前端，修改前端设计或重新布局，这种从布局布线到重新综合的重复设计可能要进行多次，才能达到时序目标。随着特征尺寸的减少，互连线影响越来越大。传统的逻辑综合和布局布线分开的设计方法已经无法满足设计要求。必须将逻辑综合和布局布线更紧密地联系起来，用物理综合方法，使设计人员同时兼顾考虑高层次的功能问题、结构问题和低层次上的布局布线问题。

5. 设计难技术

设计验证是设计工作中十分重要的一环，电路规模越大系统越复杂占用验证时间越长。目前市场上已经有了适合不同设计领域和设计对象的 CAD 工具，但如果用这些工具来验证 SoC 设计需将它们按需要组合，并集成在同一环境中。

模拟电路模拟需要晶体管级模型，大部分模拟工具都是从 SPICE 衍生出来，由于要求解电路方程，电路越复杂模拟时间越长。利用并行结构分别进行数值解算和利用模型进行模拟，可大大提高模拟速度，能对数万元器件电路乃至芯核进行模拟。但要对整个数百万门规模的 SoC 进行模拟还是有困难的。

在 SoC 上,几乎都要用到微处理器以及专门的软件和硬件。硬件和软件之间是密切相关的,但在系统被做出之前,软硬件之间的相互作用通常是很难精确测出的,一些设计错误也不会明显表现出来。为了解决这一问题,必须采用硬件/软件协同验证技术。

1.1.3 SoC 的产品效益

SoC 具有以下几方面的优势,因而创造其产品价值与市场需求。

1. 降低耗电量

随各类电子产品小型化、可携性的需求趋势,对各类用电器的省电需求将大幅提升,由于大部分的电能皆消耗在 IC 和外部信号的传输上,SoC 产品则将其转为 IC 内部信号的传输,可大幅降低功耗。

2. 减少体积

数颗 IC 整合为一颗 SoC 后,可有效缩小在电路板上所占面积,尤其对可携式产品而言,除省电外,易有助于达到重量轻、体积小的特色。

3. 增加系统功能

随着 SoC 体积的缩小,在系统产品相同的内部空间内,可整合入更多的功能元件、组件或次系统,而得以提升系统功能的丰富度。

4. 提高速度

由于芯片内部间信号传递距离的缩短,可提升信号的传输速度,而使终端产品性能有所提升。

5. 节省成本

在理想状况下,SoC 的出现使得在相同的系统功能要求下,不需再生产、封装、测试与组装多颗离散 IC,亦缩减了印刷电路板的面积,因此可适度节省成本。不过,现实情况下,由于晶片变大与功能及制程的复杂化,使生产成品率大幅下降,测试困难度与测试成本则会大幅提升,需要技术面与商业面的同步突破,方能真正达到降低成本的效果。

1.1.4 技术挑战

虽然 SoC 将带来许多市场新契机,不过,随着晶片集成度的提高与系统架构落实于晶片层次,不论在设计方法学、晶圆制程、封装、测试等方面,都面临着极大的挑战。

1. SoC 设计

目前,半导体业界的现象是半导体制造技术越走越快,但 IC 设计与验证能力却追赶不上,制造与设计间出现明显落差,成为 SoC 发展的最大瓶颈。SoC 设计所遇到的主要技术问题在于需要一套 IP 重复使用与以平台为基础的设计方法学。而这套方法学需要以公司内部或市面上具备众多质佳而易整合的 IP 作为基础,且必须包括逻辑、混沌电路、RF 电路与各类记忆体电路的设计方法的整合。

此外,还要追求低廉的 NRE 开发成本与快速的产品设计周期,开发出来的产品必须易于大量生产,且与个别独立的晶片组解决方案相比较,具备足够的市场竞争力。

2. SoC 制造

SoC 制造设计各类制程的整合,必须克服不同电路区块不同制程相容性的问题,其

中较简单的是逻辑电路间的整合，难度较高的是类比电路与逻辑电路间的整合，最难的是逻辑电路与记忆体间的整合，特别是嵌入 DRAM 的情况。此类制程整合叠加的状况，会使 SoC 制程过于繁杂，影响技术可行性或经济效益。

而由于各种特殊制程之微缩进展不一，使得在打造 SoC 制程时，微缩进展最落后的功能区块部分将成为 SoC 之经济瓶颈所在，整体的 SoC 制程均需迁就于其中。以 SoC 的各功能区块为例，微处理器及 DSP 需要先进制程技术，但类比 IC 却需要低阶制程技术，使用类比技术把各个元件整合在一起后，有可能使得成本不一定是最佳情况。

3. SoC 封装

在封装技术方面，改善晶片与接脚的连接方式，提高晶片与封装基板的热导传输，进而提高散热率，已是势在必行。此外，由于晶片功能提高，工作频率过高，将导致连接线上的电感效应，造成信号互相干扰所引发的杂信，因而限制晶片达到更高的性能，这是性能导向的 SoC 需面对的问题。

在 IC 朝小型化、高速化、高集成度发展的趋势下，以打线为主的传统封装技术，已无法满足未来技术需要，晶片级封装及 I/O 高脚位锡球封装、CSP(Chip Scale Packaging)、BGA 与 TAB 将是未来 SoC 封装技术的主流。

由于 SoC 受技术瓶颈高、生产效率低及研发时间与成本高昂等因素影响，使其市场应用仍不普及，并可能造成效益近似 SoC 的多晶片模组 (Multi Chip Module, MCM) 封装结构或是系统封装 (System in a Package, SiP) 先行卡位市场。

4. SoC 测试

以往测试设备商大都是针对单一功能设计机器，因此不仅是记忆体与逻辑 IC 的测试机台泾渭分明，就算是逻辑 IC 机台也会因应不同功能需求设计专属机种。SoC 趋势下，测试机台走向多功能单一机型，以测试各种逻辑、类比与记忆线路，缩短测试时间，加快测试速度，并满足客户“一机多用”的需求。

此外，随着 SoC 复杂度提高，要在短时间内完成测试程式设计也有困难，在设计阶段即加入测试概念，如内建自动测试技术 BIST，以及测试性设计 DFT，可有效缩短测试时程与降低测试成本。

1.1.5 商业挑战

过去，半导体业者仅需掌握生产终端产品的系统厂商的需求，SoC 的趋势则使 IC 产品业者必须开始倾听终端产品使用者的心声，方能在产品规格与开发速度上符合市场需求。但 IC 产品业者并没有真正接触终端客户的管道，这是产品开发上的一大挑战。

在产品规划时，系统公司规划的是下一周期的市场，从确认需求和规格，到开发产品，乃至销售至市场，这样一周期通常至少需要一年多的时间，而 IC 公司则必须提早看 2 个周期后的市场，方能及时完成设计。因此，从选好系统客户，拿下设计采用，直到等客户完成产品开发，系统产品销售至市场，IC 产品出货才能提升，所以 IC 公司必须早三四年预测市场，而对 SoC 来说，要掌握足够的系统 Know-How，并看到未来的市场，会是难度极高的挑战。

SoC 是市场导向、应用导向的 IC 产品，在许多区隔中产品生命周期较短，但 SoC 的开发整合工作往往多而复杂，这使得由设计至真正大量产品出货的时间会相对拉长，

成本增加，未必能有理想获益。

当 SoC 的目的是在价格导向的市场时，例如 PC 或消费性电子产品市场，在采用 SoC 的晶片时，所能支付的价格较低，但 SoC 从设计到制造的总成本会比传统的方法更高，尤其是当采用先进制程时，如此一来，产生附加价值虽高，但获利却相对有限。

此外，SoC 往往需要先进的制程，但 EDA 工具的建置、光罩成本与量产时的投片费用将十分高昂，对许多小资本的 IC 设计公司来说，更是进入市场的先天障碍。

1.2 SoC 对产业产生巨大冲击

半导体产业持续有新兴的技术/产品推出，如 SOI、MRAM、FRAM 等，其所影响的层面可能是某个产品市场的取代，或是导致性能的提升。但 SoC 的影响则远大于此，由于其改变了系统与零组件间的分工形态，冲击层面将遍及整个电子产业供应链的各个环节。

1.2.1 从产品迈向解决方案

以前，IC 产业者可以单凭系统中特定功能的离散 IC，如微处理器、周边 IC 或界面 IC，在市场上创造不错的业绩。一旦跨入 SoC 时代，单一 SoC 便可包括某一特定应用的完整系统功能，此产品趋势将导致市场区隔的细化与业者间的跨界竞争。

即便 IC 产品整合度尚未至此地步，IC 产品业者也已纷纷就应用面进行思考。由单一特定功能在 IC 往完整 IC 解决方案布局，至少是逻辑电路部分的完整 IC 布局，以提供客户完整的系统解决方案，方可满足客户需求，并抢占市场先机。

1.2.2 SoC 设计的平台化

由于跨入 SoC 时代，由应用面导致市场区隔的细化、过多的产品市场，从而将导致 IC 业者研发资源需求大增，并造成产品开发时程的严重负荷，这都将促使 SoC 产品走向平台化的设计模式，即提高设计生产力，迅速针对不同市场区隔推出 SoC 产品。

此 SoC 平台将包括微处理器/DSP、作业系统、芯片总线、关键的特定功能 IP，以及完整的软/硬件设计开发环境，并具备弹性而可扩充的特性，使 IC 产品业者得以凭此平台，快速打造符合市场需求的产品。

1.2.3 系统业者/IC 产品业者分工模式的改变

SoC 会促使系统产品硬件规划的附加价值，由系统业者端向 IC 产品业者端移动，对系统业者而言，以硬件设计与组装来降低生产成本或增加性能与功能的能力将为之减弱，所以必须靠其他要素来维持自身附加价值。除了可强化品牌/通路外，产品本身的优势则将转进至 IC 功效的发挥或应用软件的支援。

对 IC 产品业者而言，在供应链中的附加价值等可进一步提升，更有机会在供应链间利润重分配的过程中取得主动权。而供应链中 IC 产品业者与终端消费者间的距离亦将得以压缩。

此外，在系统业者与 IC 产品业者分工模式改变的情况下，系统业者内部研发资源亦

将逐步向 IC 产品业者流动。

1.2.4 供应链各部门间联盟合作之风盛行

由于打造 SoC 需要软件、硬件、IC 设计、IC 制造、封装、测试、半导体设备、IP、IC 设计服务与 EDA 业者间价值活动的整合方得以实现，其牵连到的各产业部门非常广泛，且单一部门/业者往往无力于公司内部建置所有资源，而必须向外取得，因此跨各产业部门间的联盟合作将颇为盛行。

如 IC 产品业者由于无法接触到“终端客户”，对系统需求的掌握度往往不够精确，因此促使 IC 产品业者与系统业者结盟，共同进行产品定义，以使 SoC 产品能在市场上获得成功。

1.2.5 晶圆制造的生态变动

SoC 对晶圆制造的生态分布将造成莫大打击。目前晶圆制造业者可分为晶圆代工、逻辑/非挥发性记忆体 IDM 厂、DRAM 厂等族群，各族群有着不同的核心能力。

SoC 则适于晶片内整合不同的功能线路区块，此产品的改变打破了传统制程上、产品上的分别，乃至晶圆制造族群间的界限，各族群将由自身核心竞争力出发，寻求最有利于自己的 SoC 产品定义方式，规划一条自目前产业位置至 SoC 时代的演化之路，并期待能在跨族群的竞争下抢占上风。

1.3 SoC 软/硬件协同设计

SoC 软/硬件协同设计的内容是软/硬件协同设计有能力以很高的水平来指定一个系统，将那个系统划分为硬件和软件，并分析主要的设计决策，以确保它们对系统具有所希望的影响。

协同设计是一种变动方法学，会影响设计师（Architect）及硬件和软件这两支设计队伍。协同设计是很强劲的方法学，可能导致人事组织上发生变化。如果硬件和软件设计队伍以前彼此没有交谈，那么协同设计可以令他们沟通。但是，假使其中一支队伍不需要采用这种方法学，则由此产生的抵抗力也能破坏总的努力。

不过，正如 Alcatel 公司为其 SoC（芯片上的系统）从事协同设计的公司的工程经理 Tony Denayer 指出的那样：“这种方法学要在设计周期的初始阶段有较大的投资，但于周期末尾却有相当的回报。”

1.3.1 可用性问题

SoC 是一种设计工具。每一种工具都需要用户花点时间去认识和使用。如果协同设计工具采用了非标准的语言，那么，学习时就比采用 C 语言的工具困难得多。C 语言自面世以来已成为系统级设计方面约定俗成的标准。采用 HDL（Hardware Description Language）语言来设计系统规格则需要大量的电脑运算时间，去模拟运行在系统上仅仅几秒钟的软件，而且，这对于软件设计队伍来说，是完全陌生的。

采用 C 语言的挑战是，它并没有包含硬件设计必需的全部结构。因此，需要增添一