

内摇容摇筒摇介

摇摇本书首先概述了数字集成电路发展的历史与未来 指出了硬件描述语言(匀匀匀)在设计数字电路中所起的作用 并系统讲解了灾灾匀匀匀的语法要点。在此基础上 本书以灾灾匀匀匀为工具 介绍了几种描述电路的方法与技巧 列举了几个典型电路的描述实例 然后用愿说系单片机、硬盘控制器和孕烷总线控制器接口等子系统的设计实例分别讲解了自顶向下的层次化设计方法、同步与异步数据流的控制以及 酝葬匀匀匀状态机在总线控制等方面的设计技巧。文中还对灾灾匀匀匀建模与调试、月嘴载电路的原理与灾灾匀匀匀实现作了详细论述 并提供了具体例子 最后以一个真实 粤嘴脱例子的简单介绍作为全书的结尾。

本书是灾灾匀匀匀用于数字电路设计的中高级读本 可作为大专院校计算机、微电子学和半导体专业高年级本科生和研究生的教材 也可作为数字集成电路芯片设计人员的参考书。

摇图书在版编目(悦孕)数据

摇灾灾匀匀匀硬件描述语言 辇建国编著 援—北京 :国防工业出版社 圆田源年

摇(电路设计自动化丛书)

摇陨旱苑 陨旱苑 陨旱苑

摇 I 圆灾灾匀匀匀 II 辇建国 III 圆硬件描述语言 灾灾匀匀匀程序设计 IV 圆裁园

摇中国版本图书馆 悦孕数据核字(圆田源)第 圆源号

(北京市海淀区紫竹院南路 圆号)

(邮政编码 圆田源)

北京奥隆印刷厂印刷

新华书店经售

*

开本 苑缘伊 员缘 员缘 员缘 印张 圆源 摇远千字

圆田源年 员月第 员版 摇圆田源年 员月北京第 员次印刷

印数 员— 员册 摇定价 猿元

(本书如有印装错误 我社负责调换)

目 录

第一章 绪论	1
1.1 初步了解 灾变与灾变语言	2
1.2 灾变语言的历史	3
1.3 灾变语言的主要能力	4
1.4 系统集成电路设计技术	5
1.5 系统级集成电路设计方法	6
1.6 系统级集成电路设计中的 问题	7
1.7 系统级集成电路测试技术	8
1.8 系统级集成电路芯片加工技术	9
1.9 系统级集成电路的发展未来	10
1.10 灾变语言与 灾变语言的区别	11
第二章 灾变语言初探	12
2.1 概述	13
2.2 术语	14
2.3 相关概念	15
2.4 灾变语言行为描述语言	16
2.5 灾变语言数据流描述语言	17
2.6 灾变语言网表描述语言	18
2.7 灾变语言硬件仿真	19
2.8 灾变语言背景	20
2.9 灾变语言历史与特点	21
2.10 灾变语言已存在的语言	22
2.11 灾变语言灾变语言要求	23
2.12 灾变语言	24
2.13 灾变语言中的基本概念	25
2.14 灾变语言基本概念	26
2.15 灾变语言并发性和时序	27
2.16 灾变语言对象与数据类型	28
2.17 灾变语言灾变语言的主要构件	29
2.18 灾变语言建模	30
2.19 灾变语言行为建模引论	31

摇摇圆源猿摇传输延时和固有延时的对比	猿猿
摇摇圆源猿摇仿真 圆源猿	猿圆
摇摇圆源猿摇驱动	猿猿
摇摇圆源猿摇类属	猿源
摇摇圆源猿摇块语句	猿猿
圆源猿摇顺序进程	猿苑
摇摇圆源猿摇进程语句	猿苑
摇摇圆源猿摇信号赋值和变量赋值	猿愿
摇摇圆源猿摇顺序语句	猿怨
圆源猿摇值类属性	源
摇摇圆源猿摇值类型属性	源缘
摇摇圆源猿摇数组属性	源远
摇摇圆源猿摇值块属性	源苑
摇摇圆源猿摇函数类属性	源愿
摇摇圆源猿摇信号类属性	源猿
第三章圆源猿早结构	源
猿源猿摇模块	源
猿源猿摇模块测试	源远
猿源猿摇时延及数据流	源猿
摇摇猿源猿摇时延	源猿
摇摇猿源猿摇数据流描述方式	源源
猿源猿摇行为描述方式	源缘
猿源猿摇结构化描述形式	源苑
猿源猿摇混合设计描述方式	源怨
猿源猿摇设计模拟	源怨
猿源猿摇描述	源圆
摇摇猿源猿摇灾圆猿猿语言的猿种描述方法	源圆
摇摇猿源猿摇词法习俗	源猿
猿源猿摇数据类型	源远
摇摇猿源猿摇按物理数据类型分	源愿
摇摇猿源猿摇按抽象数据类型分	源愿
猿源猿摇运算符和表达式	源怨
摇摇猿源猿摇算术运算符	源怨
摇摇猿源猿摇符号运算符	源圆
摇摇猿源猿摇关系运算符	源圆
摇摇猿源猿摇逻辑运算符	源猿
摇摇猿源猿摇位逻辑运算符	源猿
摇摇猿源猿摇一元约简运算符	源源

远源摇时间控制	员猿
摇摇远源摇延时控制	员猿
摇摇远源摇边沿触发事件控制	员苑
摇摇远源摇电平敏感事件控制(憎源语句)	员缘
远源摇赋值语句	员苑
摇摇远源摇过程赋值语句的基本格式	员苑
摇摇远源摇过程赋值的两种延时方式	员愿
摇摇远源摇阻塞型过程赋值	员猿
摇摇远源摇非阻塞型过程赋值	员源
摇摇远源摇连续赋值语句	员苑
摇摇远源摇过程连续赋值语句	员愿
远源摇分支语句	员愿
摇摇远源摇蚤源藻藻条件分支语句	员愿
摇摇远源摇糟藻藻分支控制语句	员员
远源摇循环控制语句	员苑
摇摇远源摇藻藻藻循环语句	员苑
摇摇远源摇藻藻藻循环语句	员愿
摇摇远源摇憎藻藻循环语句	圆园
摇摇远源摇藻藻循环语句	圆员
远源摇任务(嗒)与函数(枣)	圆源
摇摇远源摇任务(嗒)	圆源
摇摇远源摇函数(枣)	圆缘

第七章 灾灾早匀蕴简单设计	圆源
苑源摇加法器源程序	圆源
苑源摇计数器	圆缘
苑源摇锁存器	圆员
苑源摇元件例化	圆员
苑源摇圆位寄存器	圆员
苑源摇带藻藻藻功能的寄存器	圆员
苑源摇一个简单的状态机	圆圆
苑源摇加法器源程序	圆猿
苑源摇用状态机设计的交通灯控制器	圆源
苑源摇一个简单的灾灾	圆苑
苑源摇状态机举例	圆猿
苑源摇可综合风格的计数器设计	圆苑
第八章 硬盘控制器子系统模块化设计	圆圆
愿源摇功能描述	圆圆

悦猿不支持的操作符	猿怨
悦猿不支持的门级结构	猿怨
悦猿不支持的其他结构	猿园
附录 阅猿灾猿猿灾猿匀灾猿蕴设计练习	猿员
练习一、简单的组合逻辑设计	猿员
练习二、简单时序逻辑电路的设计	猿园
练习三、利用条件语句实现较复杂的时序逻辑电路	猿源
练习四、设计时序逻辑时采用阻塞赋值与非阻塞赋值的区别	猿缘
练习五、用 灾猿猿灾猿匀灾猿蕴解决实现较复杂的组合逻辑电路	猿愿
练习六、在 灾猿猿灾猿匀灾猿蕴中使用函数	猿园
练习七、在 灾猿猿灾猿匀灾猿蕴中使用任务(灾猿猿灾猿匀灾猿蕴)	猿园
练习八、利用有限状态机进行复杂时序逻辑的设计	猿源
练习九、利用状态机的嵌套实现层次结构化设计	猿苑
练习十、通过模块之间的调用实现自顶向下的设计	猿园

第一章 绪 论

近 30 年来,数字电路设计技术获得了飞速发展。最早的数字电路是用电子管和晶体管搭的,直到逻辑门可以做单芯片上才出现了集成电路。最早的集成电路是只有很少门数的小规模电路,随着技术的不断发展,出现了几百门的中规模集成电路和几千门的大规模集成电路。从这时起,设计过程开始变得非常复杂,设计者已经感觉到自动化设计的必要性。最早出现的计算机辅助设计(CAD),使设计人员能够开始用电路和逻辑仿真技术来验证大约 100 个晶体管的功能模块。当然,这种电路也可以在面包板上进行测试。随着超大规模集成电路技术的出现,设计者可以在单芯片上集成上万个晶体管。由于电路的复杂程度越来越高,在面包板上验证电路已经不太可能,计算机辅助技术成为设计和验证超大规模集成电路的关键。

随着集成电路的深亚微米制造技术、设计技术的迅速发展,集成电路已进入片上系统时代。所谓的片上系统,又称为系统级芯片,就是系统级集成电路,其英文缩写为 SOC (System On a Chip) 或者 SLI (System Level IC)。系统级集成电路(SOC)在单一硅芯片上实现信号采集、转换、存储、处理和 I/O 等功能,或者说在单一硅芯片上集成了数字电路、模拟电路、信号采集和转换电路、存储器、MPU、MCU、DSP、MPEG 等,实现了一个系统的功能。SOC 是在 ASIC 的基础上发展起来的电路,它与 ASIC 完全不同,具有很多独特的优点。

(1) SOC 增加了功能:从单一功能增加到多个功能,实现了一个系统的功能,达到了高速、高集成度和低功耗。

(2) SOC 大大降低了整机的成本:过去构成一个系统需用多块 IC 芯片,现在只要一块 SOC 就够了。

(3) SOC 大大降低了整机的体积:这是系统制造商进一步发展的方向,尤其对于便携式的电脑、通信及多媒体产品的生产厂家更具有吸引力。

(4) SOC 促进了整机系统更新换代的速度:它缩短了供需双方之间的差距,使整机更受用户的欢迎,便于占领市场。

SOC 的这些优点正好顺应了通信、电脑、消费类电子产品轻、薄、短和耗电少的发展方向,因此市场对 SOC 产品有强烈的需求。目前 SOC 是微电子产业界最热门的话题之一,如果说 VLSI 促进了 PC 的广泛应用而带来了信息产业的第一次革命,那么 SOC 的发展正在带来信息产业的第二次革命。美国、日本和欧洲许多国家的各大半导体公司纷纷加大对 SOC 生产线的投资力度,建立 $0.15\ \mu\text{m} \sim 0.20\ \mu\text{m}$ 加工线,用以生产 SOC。例如 NEC 公司为生产 SOC 建立了两条 IC 芯片加工线,一条为低功耗 SOC 工艺线(UC3),最细线宽为 $0.18\ \mu\text{m}$,功耗达到 $19\text{nW}/\text{MHz}/\text{门}$;另一条是高性能 SOC 工艺线,最细线宽为 $0.15\ \mu\text{m}$,速度达到 500MHz 。这两条工艺线加工 SOC 的集成度可达 3400 万门/

20mm, BGA 封装引脚可达 3000pin, NEC 已经在 1999 年第二季度投产。SGS-Thomson 公司正在建设一条开发生产 SOC 芯片的加工线, 采用 $0.12\ \mu\text{m}$ 、 300mm 技术。

SOC 逐渐成为市场的热点, 迫使可编程器件 CPLD 和 FPGA 转向 SOC。Xilinx 公司将其 FPGA 产品转向为 SOC, 该公司首先推出了百万门 Virtex 系列 FPGA, 为解决系统级设计问题提供了新的 FPGA 平台。Quick logic 公司宣布将利用该公司标准产品 FPGA 和硬核模块向客户提供可编程 SOC 产品。Altera 公司的新型可编程逻辑器件 APEX 20K 系列已于 1999 年第一季度正式面世, 它是第一个成功地将乘积项、查询表功能及内嵌式集成的体系结构。APEX 20K 也是第一个器件密度达到百万门, 系统性能可支持 64 位 / 68MHz PCI 标准的产品系列, 将整个复杂的系统集成进一片可编程芯片内, 实现 SOC; 其第一个产品 EP20K400 已供应市场。

系统级集成电路实现的必要条件之一是其线宽需达到深亚微米级。当代 SOC 芯片多数为 $0.25\ \mu\text{m} \sim 0.18\ \mu\text{m}$ 设计规则, 这与传统的 IC 设计技术完全不同, 因此给 SOC 芯片设计带来了新的困难。集成电路设计进入深亚微米阶段后, 特征尺寸缩小, 其横向和纵向尺寸也都大大缩小, 芯片内的互连线长度却急剧增大。互连线与连线间的电阻和电容对信号传输的影响非常显著, 这一变化引入了许多新问题, 给 SOC 芯片 EDA 设计提出了更多的挑战。

器件的特征尺寸降到深亚微米级, 器件的物理特性和电学特性将会发生很大的变化。原来的模型已不再适合, 必须考虑深亚微米尺寸所引起的物理效应, 单元本身的固有延迟相对来说大大减小, 而互连线所引起的延迟在整个单元延迟中所占的比例越来越大: 深亚微米连线变细, 连线间距变小, 连线变长, 这就增加了连线的分布电容。由于信号频率很高, 所以会引入串扰影响和噪声影响; 由于互连线变细, 易于引起电迁徙和热载流子效应, 因此在集成电路的设计策略上需做较大的调整, 即要从原来面向电路单元的设计策略——先安排电路模块, 然后考虑互连引线, 更改为面向电路互连引线的设计策略——先安排电路互连线网, 然后再挂电路模块, 这样可以使从总体设计上保证芯片高速工作。特别是深亚微米级芯片的速度较快, 这时对时序的要求更为严格, 因此前端的逻辑设计与后端的物理设计间很难保持一致。对于在逻辑设计中仿真分析后功能和时序都正确的网表, 在布线设计后却由于芯片空间和连线的限制, 造成互连引线的延迟与逻辑设计中使用的模型不一致, 使得时序不再满足约束的要求。这时必须回到逻辑设计中进行修改, 然后再进行仿真分析。由于逻辑设计和布局布线之间的控制因素不统一, 将会导致逻辑设计和物理设计的循环不收敛, 因而会使设计周期大大加长。

为了能在设计的初期就获得有关互连线的信息, 目前常用的一种设计方法是在设计流程中加入布局规划, 通过布局规划对电路进行预布局, 并得出电路互连延迟估计, 然后这些估计被用来指导后续的设计过程。这种方法的最大困难在于在布线规划中很难保证电路之间最大互连延迟估计的准确度。另外, 由于在设计初期即对电路的物理位置进行了约束, 势必会影响到电路的优化程度。目前采用高层次设计方法, 试图在设计初期获得一些有关深亚微米集成电路连线的物理信息, 并以此来指导后续的设计过程, 但目前仍不能十分有效地避免设计过程的迭代。高层次设计方法现在仍不能适应深亚微米高性能系统设计的要求, 应从设计流程、电路结构以及算法等多方面进行综合考虑, 力图把几方面的方法有效地结合起来, 以探索得到全面的解决方案。

最近 10 年来, 硬件描述语言 (HDL) 在逻辑电路设计中得到了广泛应用。工程管理者不再面临在设计中是否使用硬件描述语言的困境, 相反, 他们关心的是选用哪种语言能更好地与他们的设计环境相结合。使用硬件描述语言, 设计者能够更好地从功能和行为上表述自己的设计, 而且还可以加上详细的注解, 以便在以后的设计中重复使用。在具体设计之前, 通过抽象的功能描述, 可以找到灵活的系统结构, 并发现设计的瓶颈之所在。

1.1 初步了解 Verilog HDL

硬件描述语言 (Hardware Description Language) 是硬件设计人员和电子设计自动化 (EDA) 工具之间的界面, 其主要目的是用来编写设计文件, 建立电子系统行为级的仿真模型即利用计算机的巨大能力对用 Verilog HDL 或 VHDL 建模的复杂数字逻辑进行仿真, 然后再自动综合以生成符合要求且在电路结构上可以实现的数字逻辑网表 (Netlist), 根据网表和某种工艺的器件自动生成具体电路, 然后生成该工艺条件下这种具体电路的延时模型, 仿真验证无误后, 用于制造 ASIC 芯片或写入 EPLD 和 FPGA 器件中。

在 EDA 技术领域中把用 HDL 语言建立的数字模型称为软核 (Soft Core), 把用 HDL 建模和综合后生成的网表称为固核 (Hard Core), 对这些模块的重复利用缩短了开发时间, 提高了产品开发率和设计效率。

随着 PC 平台上的 EDA 工具的发展, PC 平台上的 Verilog HDL 和 VHDL 仿真综合性能已相当优越, 这就为大规模普及这种新技术铺平了道路。目前国内只有少数重点设计单位和高校有一些工作站平台上的 EDA 工具, 而且大多数只是做一些线路图和版图级的仿真与设计, 只有个别单位展开了利用 Verilog HDL 和 VHDL 模型 (包括可综合和不可综合的) 进行复杂的数字逻辑系统的设计。随着电子系统向集成化、大规模、高速度的方向发展, HDL 语言将成为电子系统硬件设计人员必须掌握的语言。

为何使用硬件描述语言?

传统的用原理图设计电路的方法已逐渐消失, 取而代之, HDL 语言正被人们广泛接受, 出现这种情况有以下几点原因:

(1) 电路设计将继续保持向大规模和高复杂度发展的趋势。20 世纪 90 年代设计的规模已达到百万门的数量级。作为科学技术大幅度提高的产物, 芯片的集成度和设计的复杂度都大大增加, 芯片的集成密度已达到 100 万个晶体管以上, 为使如此复杂的芯片变得易于人脑的理解, 用一种高级语言来表达其功能性而隐藏具体实现的细节是很必要的。这也就是在大系统程序编写中高级程序设计语言代替汇编语言的原因。工程人员将不得使用 HDL 进行设计, 而把具体实现留给逻辑综合工具去完成。

(2) 电子领域的竞争越来越激烈。刚刚涉入电子市场的成员要面对巨大的压力, 提高逻辑设计的效率, 降低设计成本, 更重要的是缩短设计周期。多方位的仿真可以在设计完成之前检测到其错误, 这样能够减少设计重复的次数。因此, 有效的 HDL 语言和主计算机仿真系统在将设计错误的数目减少到最低限方面起到不可估量的作用, 并使第一次投片便能成功地实现芯片的功能成为可能。

(3) 探测各种设计方案将变成一件很容易,很便利的事情,因为只需要对描述语言进行修改,这比更改电路原理图原型要容易实现得多。

比起传统的原理图设计方法来说,HDL有许多优点,主要有:

(1) 用 HDL 设计电路能够获得非常抽象级的描述。设计者不用选择特定的制造工艺就能写出电路的寄存器传输级(RTL: Register Transfer Level)描述。逻辑综合工具能自动将 RTL 描述转换成任何一种制造工艺。如果出现了新工艺,设计者不用再重新设计电路。当使用新工艺时,他们需要做的只是简单地把电路的 RTL 描述输入到逻辑综合工具中,就能产生出一个新的门级网表。逻辑综合工具将针对新工艺自动地进行电路面积和时序的优化。

(2) 用 HDL 描述电路设计,在设计的前期就可以完成电路功能级的验证。由于设计者工作在 RTL 级,他们可以不断地优化和修改 RTL 描述,直到满足所需要的功能为止,这时能够发现并改进设计中的绝大部分错误。在设计晚期的门级网表或物理版图中出现功能性错误的概率已经非常小,这样可以非常显著地缩短设计周期。

(3) 用 HDL 设计电路类似于计算机编程。带有注解的文字性描述更有利于电路的开发与调试。比起门级原理图来说,HDL 还提供了非常简明的设计表达。在非常复杂的设计中,门级原理图几乎是不可理解的。

逻辑综合把 HDL 推到了数字电路设计的最前沿,设计者不再需要用手工放置门电路的办法来设计数字电路。HDL 也被用作进行系统级设计,完成诸如系统板、互连总线、FPGA(现场可编程门阵列)和 PAL(可编程阵列逻辑)等电路的仿真。用 HDL 设计每一种电路的方法都是通用的。

Verilog HDL 是一种硬件描述语言,用于从算法级、门级到开关级的多种抽象设计层次的数字系统建模。被建模的数字系统对象的复杂性可以介于简单的门和完整的电子数字系统之间。数字系统能够按层次描述,并可在相同描述中显式地进行时序建模。

Verilog HDL 语言具有下述描述能力:设计的行为特性、设计的数据流特性、设计的结构组成以及包含响应监控和设计验证方面的时延和波形产生机制。所有这些都使用同一种建模语言。此外,Verilog HDL 语言提供了编程语言接口,通过该接口可以在模拟、验证期间从设计外部访问设计,包括模拟的具体控制和运行。

Verilog HDL 语言不仅定义了语法,而且对每个语法结构都定义了清晰的模拟、仿真语义。因此,用这种语言编写的模型能够使用 Verilog 仿真器进行验证。Verilog 语言从 C 编程语言中继承了多种操作符和结构,提供了扩展的建模能力 HDL。Verilog HDL 语言的核心子集非常易于学习和使用,这对大多数建模应用来说已经足够。当然,完整的硬件描述语言足以对从最复杂的芯片到完整的电子系统进行描述。

1.2 Verilog HDL 的历史

早期的集成电路设计实际上就是掩模设计,电路的规模是非常小的,电路的复杂度也很低,工作方式则主要依靠手工作业和个体劳动。40 年后的今天超大规模集成电路(VLSI)的电路规模都在百万门量级,由于集成电路大规模、高密度、高速度的需求,

使电子设计愈来愈复杂，为了完成 10 万门以上的设计，需要制定一套新的方法，就是采用硬件描述语言设计数字电路。HDL(Hardware Description Language)于 1992 年由 Iverson 提出，随后许多高等学校、科研单位、大型计算机厂商都相继推出了各自的 HDL，但最终成为 IEEE 技术标准的仅有两个，即 VHDL 和 Verilog HDL。这两种语言提供非常简洁，可读性很强的句法，使用 Verilog 语言已经成功地设计了许多大规模的硬件。

Verilog HDL 是在 1983 年，由 GDA(Gate Way Design Automation)公司的 Phil Moorby 首创的。Phil Moorby 后来成为 Verilog-XL 的主要设计者和 Cadence 公司(Cadence Design System) 的第一个合伙人。在 1984—1985 年，Moorby 设计出第一个关于 Verilog-XL 的仿真器，1986 年他对 Verilog HDL 的发展又做出另一个巨大贡献，提出了用于快速门级仿真的 XL 算法。

随着 Verilog-XL 算法的成功，Verilog HDL 语言得到迅速发展。1989 年，Cadence 公司收购了 GDA 公司，Verilog HDL 语言成为 Cadence 公司的私有财产。1990 年，Cadence 公司公开了 Verilog HDL 语言，成立了 OVI(Open Verilog International)组织来负责 Verilog HDL 的发展。IEEE 于 1995 年制定了 Verilog HDL 的 IEEE 标准即 Verilog HDL 1364-1995。

1987 年，IEEE 接受 VHDL (VHSIC Hardware Description Language) 为标准 HDL，即 IEEE1076-87 标准，1993 年进一步修订定为 ANSI/IEEE1076-93 标准。现在很多 EDA 供应商都把 VHDL 作为其 EDA 软件输入/输出的标准。例如，Cadence、Synopsys、Viewlogic、Mentor Graphic 等厂商都提供了对 VHDL 的支持。

Verilog HDL 语言最初是于 1983 年由 Gateway Design Automation 公司为其模拟器产品开发的硬件建模语言。那时它只是一种专用语言。由于他们的模拟、仿真器产品的广泛使用，所以 Verilog HDL 作为一种便于使用且实用的语言逐渐为众多设计者所接受。在一次努力增加语言普及性的活动中，Verilog HDL 语言于 1990 年被推向公众领域。Open Verilog International (OVI) 是促进 Verilog 发展的国际性组织。1992 年，OVI 决定致力于推广 Verilog OVI 标准成为 IEEE 标准。这一努力最后获得成功，Verilog 语言于 1995 年成为 IEEE 标准，称为 IEEE Std1364 - 1995。完整的标准在 Verilog 硬件描述语言参考手册中有详细描述。

1.3 Verilog HDL 的主要能力

下面列出的是 Verilog 硬件描述语言的主要能力：

- (1) 基本逻辑门，例如 and、or 和 nand 等都内置在语言中。
 - (2) 用户定义原语 (UDP) 创建的灵活性。用户定义的原语既可以是组合逻辑原语，也可以是时序逻辑原语。
 - (3) 开关级基本结构模型，例如 pmos 和 nmos 等也被内置在语言中。
- Gateway Design Automation 公司后来被 Cadence Design Systems 公司收购。
- (4) 提供显式语言结构指定设计中的端口到端口的时延及路径时延和设计的时序检查。
 - (5) 可采用 3 种不同方式或混合方式对设计建模。这些方式包括：行为描述方式——

使用过程化结构建模；数据流方式——使用连续赋值语句方式建模；结构化方式——使用门和模块实例语句描述建模。

(6) Verilog HDL 中有两类数据类型：线网数据类型和寄存器数据类型。线网类型表示构件间的物理连线，而寄存器类型表示抽象的数据存储元件。

(7) 能够描述层次设计，可使用模块实例结构描述任何层次。

(8) 设计的规模可以是任意的，语言不对设计的规模（大小）施加任何限制。

(9) Verilog HDL 不再是某些公司的专有语言而是 IEEE 标准。

(10) 人和机器都可阅读 Verilog 语言，因此它可作为 EDA 的工具和设计者之间的交互语言。

(11) Verilog HDL 语言的描述能力能够通过使用编程语言接口（PLI）机制进一步扩展。PLI 是允许外部函数访问 Verilog 模块内信息、允许设计者与模拟器交互的例程集合。

(12) 设计能够在多个层次上加以描述，从开关级、门级、寄存器传送级（RTL）到算法级，包括进程和队列级。

(13) 能够使用内置开关级原语在开关级对设计完整建模。

(14) 同一语言可用于生成模拟激励和指定测试的验证约束条件，例如输入值的指定。

(15) Verilog HDL 能够监控模拟验证的执行，即模拟验证执行过程中设计的值能够被监控和显示。这些值也能够用于与期望值比较，在不匹配的情况下，打印报告消息。

(16) 在行为级描述中，Verilog HDL 不仅能够在 RTL 级上进行设计描述，而且能够在体系结构级描述及其算法级行为上进行设计描述。

(17) 能够使用门和模块实例化语句在结构级进行结构描述。

(18) 图 1-1 显示了 Verilog HDL 的混合方式建模能力，即在一个设计中每个模块均可以在不同设计层次上建模。

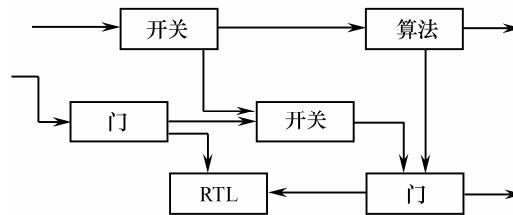


图 1-1 混合设计层次建模

(19) Verilog HDL 还具有内置逻辑函数，例如 &（按位与）和 |（按位或）。

(20) 对高级编程语言结构，例如条件语句、情况语句和循环语句，语言中都可以使用。

(21) 可以显式地对并发和定时进行建模。

(22) 提供强有力的文件读写能力。

(23) 语言在特定情况下是非确定性的，即在不同的模拟器上模型可以产生不同的结果；例如，事件队列上的事件顺序在标准中没有定义。

HDL 语言的主要特征：

(1) HDL 语言既包含一些高层程序设计语言的结构形式，同时也兼顾描述硬件线路

连接的具体构件。

(2) 通过使用结构级或行为级描述，可以在不同的抽象层次描述设计。HDL 语言采用自顶向下的数字电路设计方法，主要包括 3 个领域 5 个抽象层次。如表 1-1 所列。

表 1-1 HDL 抽象层次描述

	行为领域	结构领域	物理领域
系统级	性能描述	部件及它们之间的逻辑连接方式	芯片、模块、电路板和物理划分的子系统
算法级	I/O 应答算法级	硬件模块数据结构	部件之间的物理连接, 电路板、底盘等等
寄存器传输级	并行操作寄存器传输、状态表	算术运算部件、多路选择器、寄存器总线、微定序器、微存储器之间的物理连接方式	芯片、宏单元
逻辑级	用布尔方程述	门电路、触发器、锁存器	标准单元布图
电路级	微分方程表达	晶体管、电阻、电容、电感元件	晶体管布图

(3) HDL 语言是并发的，即具有在同一时刻执行多任务的能力。一般来讲，编程语言是非并行的、但在实际硬件中许多操作都是在同一时刻发生的，所以 HDL 语言具有并发的特征。

(4) HDL 语言有时序的概念。一般来讲，编程语言是没有时序概念的，但在硬件电路中从输入到输出总是有延迟存在的，为描述这些特征 HDL 语言需要建立时序的概念。因此，使用 HDL 除了可以描述硬件电路的功能外，还可以描述其时序要求。

1.4 系统集成电路设计技术

1.4.1 系统级集成电路设计方法

SOC 芯片不仅要集成一个复杂的系统，芯片的集成密度大、速度快，还要求解决各种干扰问题。毫无问题地完成系统级设计，是一项十分艰难的任务。按照国际上对集成电路设计人员的设计能力评估，在 RTL 层次，当今职业设计人员的设计能力平均每天完成 200 门 IC 设计工作量，而片上系统的集成度大致在 1000 万个等效门，因此如果从头开始完成上述 SOC 芯片设计不光需要相当多的人力和时日，还要花费大量的 NRE (非重复性工程费)。现在电子产品的生命周期正在不断缩短，创新产品、新结构产品的周期一般不超过 1 年，而要求完成芯片设计的时间就更要短些，因此必须找出新的解决途径。我们平常做整机系统设计或电路板设计，需要一些现成的元器件和 IC 芯片，然后把它们组合在一起，调试、查错，最终完成产品。现在把这种方式应用到 SOC 芯片设计上，把已经经过验证的 IC 电路以模块的形式去参加 SOC 芯片的设计，这样设计就变得容易了，也可大大缩短设计时间，从而解决了 SOC 芯片上市时间和设计成本高的问题。因此人们在进行 SOC 芯片设计时，必须重复使用已验证的 IC 模块。这些已验证的、可重复利用的 IC 模块具有知识产权问题，通常被称为 IP (Intelligent Property)，也有人称它为系统宏单元 (System-Level Macro) 或虚拟部件 (Virtual Component) 芯核 (Core)。

SOC 芯片设计有 3 种不同方法,第一种方法称为专用系统设计方法。系统商确定出系统指标,SOC 芯片中应采用的 IP 模块,如通用 IP、专用的 IP 全由芯片厂商(可能是半导体厂商,也可能是专门设计芯片的公司)设计完成。这种设计方法的特点是片上系统成本可能最低,但需要较长的设计时间,灵活性较小,适于 SOC 芯片产量特别大的情况。第二种方法称为部分集成法。系统厂商设计 SOC 芯片中的专用电路部分,SOC 芯片中的 DSP IP、MPU IP 以及存储器 IP 等由 IP 公司提供,由半导体厂商或专门设计芯片的公司完成整体设计。这种方法具有一定的灵活性,新产品开发时间也较短。第三种方法称为桌面集成法。各种 IP 模块的供给公司将它们提供给电子系统商,由电子系统商设计其专用电路部分,并与外购的各种 IP 相结合完成 SOC 芯片设计。这种设计方法成本最低,设计灵活性最大。

1.4.2 系统级集成电路设计中的 IP 问题

IP 模块是 SOC 芯片中的一个非常复杂的问题,SOC 芯片设计业如今面临着很多问题。IP 模块知识产权保护是在 SOC 芯片设计中再利用的关键,连美国的知识产权专家都认为,目前没有现成的法律可以保护这种知识产权,估计也很难批准通过这样的法律,因为这样的法律执行起来也很难。除法律手段外,依靠技术手段来保护知识产权的工作正在进行当中。当前尚存在以下两个问题:

(1) 从 IP 模块的提供者来看,问题是如何设计商用 IP,如何进行恰当的描述使得既能方便使用者进行再利用又不暴露知识产权的秘密以及如何对 IP 模块进行维护,使它适应技术的发展。

(2) 从 IP 模块的使用方面来看,问题是通过什么渠道可以找到所需要的 IP 模块,如何对它进行评估、验证,如何能够购买到,如何正确使用以及许多标准化的问题。

为了解决上述 IP 模块在 SOC 芯片设计中存在的问题,1996 年 9 月世界 35 个著名公司成立了一个国际性企业联合组织:虚拟插座接口联盟(The Virtual Socket Interface Alliance,简称 VISA)。参加该组织的公司包括半导体制造公司、设计公司、EDA 公司、IP 提供者公司及整体系统公司,目前已发展到 170 多家厂商。VSIA 的目标是开发 IP 模块的接口,制定一系列开放标准、IP 功能评价和验证方案。此外还有其他一些有关 IP 组织,如可再利用的专用 IP 开发者协会 RAPID(Reusable Application Specific Intellectual Property Developers),该协会也在进行这方面的工作。为了更好地交易 IP 模块,东芝、摩托罗拉、西门子等 9 家有关半导体公司宣布:他们和英国苏格兰发展局联合在英国成立了虚拟部件交易所(VCX),会员公司可获得 VCX 提供在交易所登记的 IP 模块信息,协助买卖双方订立合同,并征收权利使用费、监督盗用、仲裁争议等。

IP 模块的再利用,除能缩短 SOC 芯片设计时间外,还能降低设计和制造的成本,提高可靠性,因而将会给 IC 产业和电子工业带来巨大的商业利益,也会引起 IC 产业结构的变革。现在已经出现了一些与 IP 模块有关的业种,如 IP 模块提供者、IP 价值评价机构等。SOC 芯片设计不再局限于单个设计部门,不再局限于一个公司内部,而是将整个 IC 产业的各种资源实现优化配置,促使 IC 产业更快地发展。

通常 IP 模块分为 3 类:即硬 IP、软 IP 和固 IP,也称硬核(Hardcore)、软核(Softcore)和固核(Firmcore)。

硬核的电路布局和工艺是固定的，不能更改。硬核已完成了全部的前端和后端设计，制造也已确定。它的特点是灵活性最小，知识产权的保护比较简单。IP 模块提供者供给用户的是封装好的行为模型，用户只能从外部测试硬核的性能，却无法得到厂家真正的电路设计。

软核是包括逻辑描述（RTL 和门级 Verilog HDL 或 VHDL 代码）、网表和不能物理实现的用于测试的文档（Test Bench Tile）。与硬核相比较软核有最大的灵活性，用户能把 RTL 和门级 HDL 表达的软核修改为自己所需要的设计，综合到选定的厂商工艺上，并通过布局布线实现具体电路。

固核是一种介于软核和硬核之间的 IP，通常以 RTL 代码和对应具体工艺网表的混合形式提供。固核既不是独立的，也不是固定的，它可根据用户要求进行修改，使它适用于某种可实现的工艺过程。固核允许用户重新确定关键的性能参数。从完成 IP 模块设计所花费的代价来看，硬核代价最高；从 IP 模块的使用灵活性来讲，软核的可重复使用性最高；从期望 IP 模块的价值最高的角度出发，人们期望 IP 完成物理设计，但这会使 IP 模块的可重复性降低。固核可根据系统设计的需求进行修改，但知识产权不易保护，虽然用户乐于接受，但缺乏固核的提供者。

1.4.3 系统级集成电路测试技术

SOC 芯片的测试技术难度较大，因为在一块硅片上涉及到几种技术的集成。过去的测试设备主要是针对专用优化的自动测试仪，一台 VLSI 测试仪可能会有高功效的数字测试能力，但对检测混合信号 IC 未必行得通；另一方面混合信号测试仪可能在模拟功能检测方面十分优秀，但对高性能的数字 IC 却无能为力。SOC 芯片测试设备则必须能够精确地检测模拟和数字两种电路，并支持扫描检测和嵌入式存储器检测。检测工程师还必须面对缺少检测触点的现状。总之，SOC 芯片的集成度和功能已太庞大，在输入引脚加测试向量，再从输出引脚观察结果的传统的检测方法已不再适用。采用传统方法，测试向量集会过分庞大，执行时间也会长得惊人。

目前开发 SOC 芯片测试仪的供应商只有几家，如 Schlumberger、Advantest、Teradyne 和 LTX。它们的 SOC 芯片测试仪器频率最高达 1GHz，引脚数量最高为 2048pin，通道数最高达 1024。大部分产品是在原逻辑和存储器测试仪器上添加硬件和软件，特别在测试方法上进行了改进。目前 SOC 芯片的测试现状还不能令人满意。SOC 芯片测试并无统一标准，美国 IEEE 学会专业组正在注意该标准的制订，SOC 芯片商需要拥有成套的 SOC 芯片测试方法，需要满足各个生产阶段的测试技术要求和降低测试成本。扫描检测法、内建自测试电路技术及 IDDQ 检测等新技术将在 SOC 芯片测试中发挥更大的作用。

1.4.4 系统级集成电路芯片加工技术

半导体加工技术是实现 SOC 芯片的基础，深亚微米加工技术是实现 SOC 芯片的关键条件，目前的 SOC 芯片特征尺寸为 $0.18\ \mu\text{m} \sim 0.25\ \mu\text{m}$ ，集成度达数百万门/芯片。为了继续提高 SOC 芯片的性能、增加芯片功能、扩大芯片品种和降低芯片成本，必须继续缩小 IC 的特征尺寸，增大集成度。按照摩尔定律，半导体技术每隔 3 年，IC 特征尺寸缩小 30%，集成度提高 4 倍。人们正在积极开发新的加工技术，进行专项工艺研究，其中