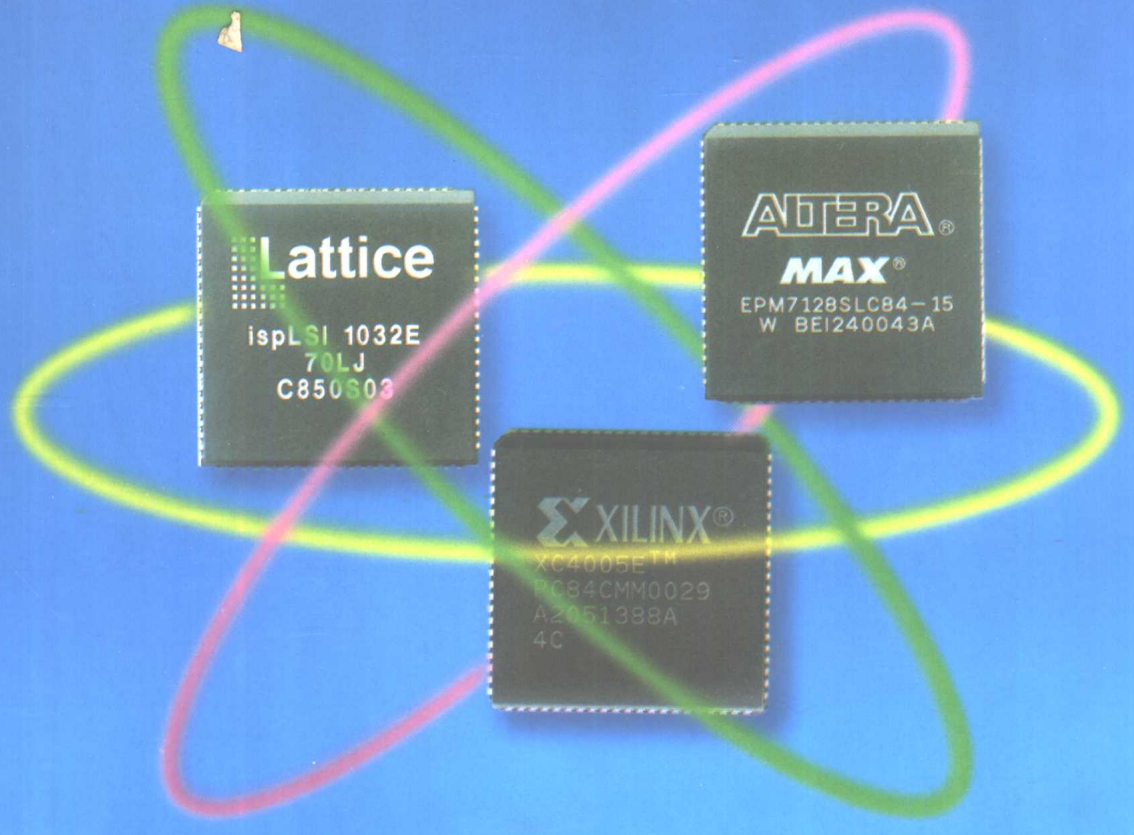
可编程逻辑器件与EDA技术

李景华　杜玉远　主编



NEUPRESS
东北大学出版社

可编程逻辑器件及EDA技术

李景华　杜玉远　主编

东北大学出版社

图书在版编目（CIP）数据

可编程逻辑器件及 EDA 技术/李景华，杜玉远主编. —沈阳：东北大学出版社，2000.12（2001.3 重印）

ISBN 7-81054-495-0

Ⅰ.可… Ⅱ.①李… ②杜… Ⅲ.可编程序控制器-高等学校-教材 Ⅳ.TP332.3

中国版本图书馆 CIP 数据核字（2000）第 87351 号

内 容 简 介

本书系统地介绍了常用的可编程逻辑器件的基本工作原理，ABEL-HDL 和 VHDL 语言、典型 EDA 开发系统的使用、典型数字系统设计举例。全书内容新颖，举例充实。

本书以数字电路和系统设计为主线，突出了采用 EDA 技术和使用大规模和超大规模集成电路来实现数字系统设计的特点。本书在数字系统层次化设计方法、VHDL 语言、可编程逻辑器件的基础、典型 EDA 系统的使用说明等几部分的内容体现了作者多年从事可编程逻辑器件开发和研究的成果。全书易读易懂。该书可作为相关专业大学本科高年级学生和研究生的教科书或参考书，也是从事电子电路和系统设计工程师的一本很好的自学参考书。

（沈阳市和平区文化路 3 号巷 11 号　　邮政编码 110004）

电话：(024)23890881　　传真：(024)23892538

网址：http://www.neupress.com　E-mail：neuph@neupress.com

东北大学印刷厂印刷　　东北大学出版社发行

开本：787mm×1092mm　1/16　　字数：524 千字　　印张：21

印数：1001～3000 册

2000 年 12 月第 1 版　　2001 年 3 月第 2 次印刷

责任编辑：李毓兴　　责任校对：米　戎

封面设计：唐敏智　　责任出版：杨华宁

定价：32.00 元

前　言

20世纪90年代是可编程逻辑器件和EDA技术发展最快的时期。随着深亚微米半导体制造技术的进步，可编程逻辑器件向着高密度、高速度、低价格迅速发展，计算机技术的普及也加速了电子设计自动化技术的进程。各半导体制造商以及第三方软件商近些年来推出了各种版本的EDA开发系统。这些EDA开发系统的自动化和智能化程度也日臻完善，特别是以VHDL为代表的硬件描述语言改革了传统的数字系统的描述方法和设计方法。软件实现硬件化、硬件设计软件化、用户自制大规模和超大规模集成电路等过去被认为是梦想的事情而今都成为了现实。

可编程逻辑器件及EDA技术的学习和应用在国外已经相当普及。与国外相比，我国尚有较大差距。除了经济和认识上的问题，教材和资料的缺乏也是重要的原因。出于这一考虑编写此书，旨在介绍常用可编程逻辑器件的基本原理、ABEL和VHDL硬件描述语言、常用的EDA系统的使用、典型数字系统的设计方法和实例。阅读此书，能够帮助相关专业的大学生、研究生更新知识点，学习掌握当今社会急需的EDA设计技术和设计方法。对从事电子系统设计的现场工程师来说，通过此书可以方便快捷地掌握最新设计方法和开发手段，加速产品的更新换代。

全书共分12章，绪论、第1章～第4章由李景华和宋勤编写，第5章～第11章由杜玉远和李汇明编写，全书由李景华教授统稿。在编写过程中李丽娟、薛蕴全等同学参与了程序的举例、校对和制图工作。

在本书的编写过程中引用了不少专家和学者的著作，也引用了国外公司的一些产品手册的数据资料，在此深表谢意。

由于可编程逻辑器件和EDA技术发展很快，编者水平有限，在书中可能存在不足或错误，恳请读者批评指正。

目　录

绪　论

(1) 可编程 ASIC 现状与发展

专用集成电路 ASIC 是面向用户特定用途或特定功能的大规模、超大规模集成电路。专用集成电路的英文是 Application Specific Integrated Circuit, ASIC 是其英文缩写。ASIC 有数字的、模拟的、数字和模拟混和的。按制造方式区分,有全定制 ASIC、有半定制 ASIC、可编程 ASIC 三种。其中可编程 ASIC 可做到用户在现场对其编程来实现各种特定逻辑功能。正是可编程 ASIC 独特的器件性能和应用方式使用户可"自制"大规模数字集成电路的理想成为现实。现在,使用可编程 ASIC 和相应的 EDA 开发系统,用户可以借助计算机实现各种实际的数字电路或电子系统的设计、功能模拟、时间模拟以及系统调试。因此,可编程 ASIC 的问世及广泛应用促进了电子系统设计方法的重大变革这一说法毫不过分。

从 20 世纪 70 年代 Intel 公司第一个推出 4004MPU 起到 80 年代初,是 MPU 技术飞速发展的时期。MPU 技术的快速渗透刺激了 MPU 外围 LSI 器件的发展。当时由 MPU、MPU 的外围 LSI 器件、通用 IC 这三大积木块搭起来可以标准地实现一个复杂的电子系统。到了 20 世纪 80 年代中期 MPU 由 8 位、16 位发展到 32 位,速度和集成度越来越高,再加上电子产品的少批量多品种化趋势,高速低功耗及小型化的要求,原来的电子系统中 MPU 的外围 LSI 和通用 IC 适应不了这一技术上的变化。20 世纪 80 年代中期以来可编程 ASIC 以其现场可编程、高速、高集成度的优势充当了电子系统中新的积木块。由 MPU、存储器和可编程 ASIC 这三个可编程的积木块组成现代电子系统已形成了趋势或潮流。是否采用可编程 ASIC 来实现电子产品的设计已成为衡量电子产品是否先进的标准之一。可以说可编程 ASIC 技术是现代电子系统设计的新潮流,对于一个现代电子系统设计工程师来说,学习应用可编程 ASIC 技术势在必行。

(2) 关于可编程 ASIC 器件分类以及选择问题的讨论

使用可编程 ASIC 进行逻辑设计,面临的第一个问题就是依据设计要求如何选择器件问题。可编程 ASIC 包括多个分支,每个分支又包括多种可编程逻辑器件,它们的集成度、结构、制造工艺、编程方式各不相同。因此,选用可编程 ASIC 器件与选用通用 IC(TTL74 系列和 CMOS4000 系列)有很大区别。例如,选用通用 IC 时特别关心芯片具有的逻辑功能。然而,选用可编程 ASIC 就不同了。从市场买来的可编程 ASIC 芯片没有任何逻辑功能,通常称为白片或空片,待用户开发编程后才具有逻辑功能。下面就可编程 ASIC 器件分类问题进行具体讨论。

①根据芯片的集成度和结构的复杂度对器件进行分类

常用可编程 ASIC 器件根据其集成度和结构复杂度的不同大致分为三类:

a. 简单可编程逻辑器件 SPLD ;

b. 复杂可编程逻辑器件 CPLD;

c. 现场可编程门阵列 FPGA 。

简单可编程逻辑器件 SPLD 属于小规模可编程 ASIC。集成度小于 PALCE22V10 或 GAL22V10 的 PLD 可视为简单可编程逻辑器件。它们的特点是都具有可编程的与阵列、不可

编程的或阵列、输出逻辑宏单元 OLMC 和输入输出逻辑单元 IOC。Lattice 公司的 PAL 和 PALCE 系列、GAL 系列等都属于 SPLD,它们适合规模较小的逻辑设计。例如选用一片或两片简单的 SPLD 就可代替印刷电路板中多片中、小规模的通用 IC,实现浓缩原来的组合电路和时序电路,借以达到联线少、电路体积小、可靠性高之目的。

复杂可编程逻辑器件 CPLD 属于中规模可编程 ASIC。集成度大于 PAL22V10 或 GAL22V10 的 PLD 都可视为 CPLD。Lattice 公司的 ispLSI/pLSI 1000 系列和 MACH5 系列,Xilinx 公司的 XC9500 系列,Altera 公司的 MAX7000 系列和 MAX9000 系列都是 CPLD 的代表性产品。CPLD 在集成度和结构上呈现的特点是具有更大的与阵列和或阵列,增加了大量的宏单元和布线资源,触发器的数量明显增加。高速的译码器、多位计数器、寄存器、时序状态机、网络适配器、总线控制器等较大规模的逻辑设计可选用 CPLD 来实现。需要说明,近年来各芯片生产厂家又纷纷推出规模更大的 CPLD。Lattice 公司的 ispLSI/pLSI3256,其集成度达 14000 个等效 PLD 门,寄存器数量达 480 个。Lattice 公司的 ispLSI6000 系列,其集成度达到 25000 个等效 PLD 门且具有 320 个宏单元。Altera 公司的 MAX9000 最高集成度可达 24000 个等效 PLD 门,逻辑宏单元达 1024 个。因此,具有复杂算法的数字滤波器等数字信号处理单元的逻辑设计也可选用这些具有更高集成度的 CPLD 来实现。

现场可编程门阵列 FPGA 是集成度和结构复杂度最高的可编程 ASIC。Xilinx 公司的 FPGA XC4025 有 2.5 万个等效 PLD 门,XC4085 集成度可达 8.5 万个等效 PLD 门。有资料表明 Xilinx 公司最新推出低电压 FPGA XCV1000 芯片最大门数可达 100 万个等效 PLD 门。Actel 公司第二代 Anti-fuse FPGA 的集成度达 2 万个等效 PLD 门。Altera 的 FPGA FLEX10K250 的集成度为 25 万个等效 PLD 门。Xilinx 公司的 XC5000/4000 系列,Actel 公司的 ACT 系列,Altera 公司的 FLEX8000,FLEX10K 和 FLEX20K 系列等是 FPGA 的代表产品。运算器、乘法器、数字滤波器、二维卷积器等具有复杂算法的逻辑单元和信号处理单元的逻辑设计可选用 FPGA 实现。Xilinx 公司和 Altera 公司最新开发先进的 IP CORE(IP 核),为 FPGA 在数字系统设计和 DSP 技术领域的应用提供了范例。

② 根据可编程 ASIC 器件的制造技术和编程方式进行分类

早期的可编程 ASIC 多采用双极型制造技术,对器件编程采用摧毁熔丝的办法实现。虽然现在市场仍可买到采用双极熔丝制造技术的可编程 ASIC 器件,但是随着 CMOS 技术的进步,CMOS 制造技术已经成为近几年可编程 ASIC 器件制造的主导技术。概括起来有以下四种:

a. 双极熔丝制造技术的可编程 ASIC(Lattice 的 PAL 系列);

b. EECMOS 制造技术的可编程 ASIC(Lattice 的 GAL 和 ispLSI/pLSI 系列);

c. SRAM 制造技术的可编程 ASIC(Xilinx 的 FPGA,Altera 的 FPGA);

d. 反熔丝制造技术的可编程 ASIC(Actel 的 FPGA)。

由于采用的制造技术不同,可编程 ASIC 具有不同的应用特性。采用双极熔丝和反熔丝制造技术的可编程 ASIC 用户只能一次性编程,通常称为 OTP(one time programming)器件。而采用 EECMOS 和 SRAM 制造技术的可编程 ASIC 具有用户可重复编程的特性,可以实现电擦电写。采用 EECMOS 技术、双极熔丝技术、反熔丝制造技术的可编程 ASIC 编程后几乎可永久性保存编程数据。而采用 SRAM 技术制造的 FPGA 则具有数据挥发性,又称易失性。具有挥发性的 FPGA,当系统断电或掉电后,写入 FPGA 中的编程数据要丢失。因此,必须把要下载

到 FPGA 的数据借用编程器固化到与其联用的 EPROM 或 EEPROM 中，待重新上电时，芯片将编程数据再下载到 FPGA 中。FPGA 的数据挥发性，决定有些环境不宜选用，必须特殊注意。然而 FPGA 的挥发性也可被利用。当要求器件在运行中能动态地实现重新配置来实现新的逻辑功能的场合，FPGA 是理想的选择器件。对于大批量生产的电子产品，过去一般先选用 GAL 和 FPGA 进行开发、研制，待定型后用 PAL 或反熔丝 FPGA 替代，可以减少成本。然而，近年来随着可编程 ASIC 应用广泛，芯片价格降低，对小批量多品种的电子产品直接选用 SPLD，CPLD，FPGA 来实现设计日渐普遍。

由于可编程 ASIC 器件采用制造技术的不同，带来编程方法也有区别。选用器件时也必须考虑到手头具有的或应该具有的编程工具和开发工具。可编程 ASIC 的编程方式有两种，一种是采用专用编程器进行编程，一种是在系统编程。后者甩掉了专用编程器，而且也不用将芯片从电路系统取下，只利用计算机和一组下载电缆就可以在系统编程。Lattice 和 Xilinx 以及 Altera 等几家大公司现在都有在系统可编程 ASIC 产品。在系统编程方式已成为各可编程 ASIC 生产厂家推崇、用户极力追求的方向。

(3) 可编程 ASIC 的一般开发步骤

可编程 ASIC 虽然种类很多、编程方法不同，但开发步骤大体相同。其开发步骤如图 0-1 所示。

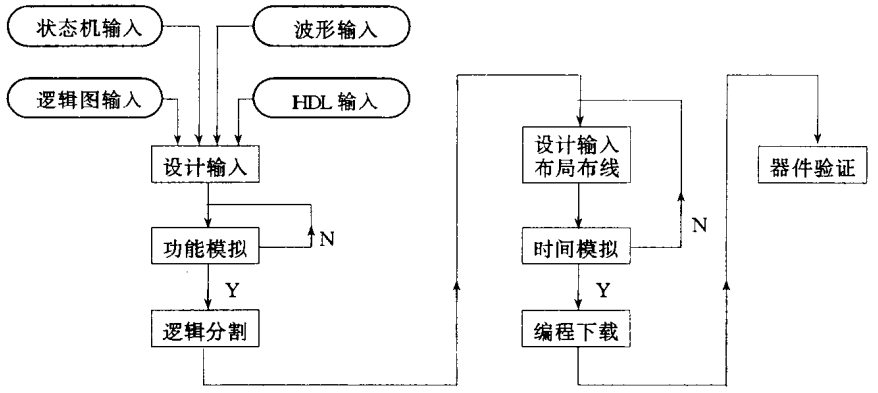


图 0-1　ASIC 开发步骤流程图

①设计输入(entry)

随着电子设计自动化 EDA 技术的进步和软件开发系统的日趋完善，使可编程 ASIC 的设计输入方式更加灵活方便。使用 SPLD 和 CPLD 以及 FPGA 实现电子系统设计时可采用逻辑原理图(schematic)输入方式、硬件描述语言 HDL 输入方式、状态机输入方式等。近几年在 ASIC设计领域十分流行一种电子系统的设计描述语言——硬件描述语言 HDL(Hardware Description Language)，它把电子系统设计、仿真综合和测试联系起来，不仅支持电路级别的设计描述，而且还支持对寄存器传输级系统结构级和系统行为级的描述。所要设计电路和系统的硬件构成及其功能均可借用 HDL 进行设计描述并进行设计输入。目前 VHDL 和 Verilog—VHDL 以及 ABEL—HDL 都是广泛使用的设计输入硬件描述语言。设计输入中出现错误，专用的设计软件会自动进行编译并发出警告。国内外近几年采用 VHDL 进行系统逻辑设计已成为流行方式。因此，跟踪新技术，学习 VHDL 语言及其用 VHDL 语言实现数字系统设计是

电子工程师一项紧迫的任务。

②功能模拟(function simulation)

在功能模拟阶段主要对所设计的电路及所输入的电路进行功能验证。此外,电路中各逻辑门或各单元模块的输入、输出是否有矛盾,是否有扇入、扇出不合理,违反扇入扇出条件;各单元模块有无未加处理的输入信号端、输出端是否悬空、是否允许使能等项内容均在功能模拟阶段进行检查验证。功能模拟也是由相应专门软件完成,如有上述问题功能模拟软件自动发出警告,指出错误的信息。

③逻辑分割(partitioning)

在逻辑分割阶段设计者输入的电路将变成器件内部门阵列、宏单元或微型子逻辑功能块能够实现的电路。例如,器件内部的各子逻辑功能块能够实现4输入变量的任意逻辑函数,然而设计输入的是要实现5变量逻辑函数的逻辑电路,这就必须采用逻辑分割的办法,将其用多个子逻辑功能块来实现。逻辑分割的过程就是将复杂电路分解成由若干子逻辑功能块实现的过程。逻辑分割也是借助专门软件实现的。

④布局和布线(place and routing)

在布局和布线阶段是用子逻辑功能块将要实现的逻辑电路布置在与所选用的实际芯片相同的虚拟芯片上。布局和布线完了,所实现的电路就定下来了,其电路性能也就确定了。布局和布线的方法不同,布线引起的延时就不同,所实现电路的速度指标受到影响。有时,布线不好会造成芯片资源浪费或电路不可实现。布局和布线是一项复杂的工作。电路密度过高,自动布线不易进行。施加一定量的手动布线,以期解决布线浪费和减少布线死区是常有的事。

⑤时间模拟(timing simulation)

时间模拟是在布局和布线之后进行。FPGA具有统计型的连线结构,布线软件对有相同逻辑功能的电路完全可能给出不同的布线模式。因此,其系统的时间特性也完全可能不同。有时布线延时还会给电路功能实现带来新的障碍,所以用FPGA设计实现的电路进行时间模拟是非常必要的。在使用简单SPLD和CPLD器件实现设计时,由于器件的连线结构属于确定型的,布线延时基本是一定的,时间模拟不进行有时也能满足设计要求,然而对一个复杂系统而言,系统内部时间特性复杂,竞争冒险可能存在,一般在设计一个实际的复杂系统时,时间模拟这一步是必不可少的。通过时间模拟可得到系统内部的延时特性,发现竞争冒险等信息。时间模拟对提高系统稳定性十分重要。一个EDA开发系统,是否具有timing simulation功能也是衡量这个EDA开发系统先进性的一项指标,是购买和选用EDA开发系统应该注意的问题。

⑥写入下载数据(download)

在写入下载数据阶段,所设计的电路或系统即将完成。所选用的器件若是CPLD和SPLD,一般选用在系统编程或使用合适的编程器将相应的JED下载数据写入到芯片中。所选用的器件若是FPGA,则需要用专门的EPROM编程器对与FPGA相配置的EPROM芯片进行编程,将FPGA的配置数据先写入EPROM中。

(4) TOP-DOWN和BOTTOM-UP设计思想

利用可编程ASIC器件进行数字系统设计时,芯片本身将成为包含有数以万计元件的电子系统。很难想象由一个设计者独立设计不产生错误。为了保证设计的快速性、正确性,提高芯片的利用率,减少仿真的工作量,通常采用称为top-down和bottom-up层次化的设计方法。

自顶向下(top-down)设计首先是从系统级开始入手。把系统分成若干基本单元模块,然后再把作为基本单元的这些模块分成下一层的子模块。再把子模块作为基本单元继续分解成下一层次的基本单元,该层次的基本单元又由再下一层次的基本单元互联而成。一直做到可以用 CAD 或 EDA 元件库中的元件能实现为止。自顶向下的层次结构化设计思想可用图 0-2 绘出的 top-down 设计树来描述。在这个设计树中顶层模块对应着系统级的行为描述,树枝对应着基本单元的结构分解,最底层模块对应基本单元的电路结构描述。

top-down 的设计过程可理解为从硬件的高层次抽象描述向最低层结构描述的一系列转换过程,直到最后得到可实现的硬件单元描述为止。

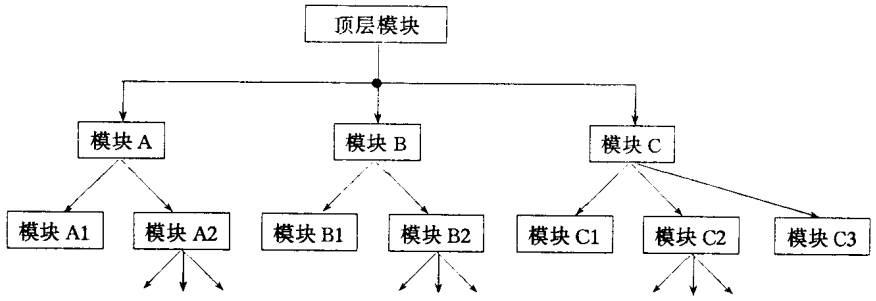


图 0-2　TOP-DOWN 设计树

硬件可以在结构域中描述,也可以在行为域中描述。在结构域中,硬件系统由基本单元的互连来描述。在行为域中,硬件系统由其输入、输出关系来描述。采用 top-down 层次结构化设计方法,设计者可在一个硬件系统的不同层次的模块下进行设计。总体设计师可以在上层模块级别上对其下层模块设计者所做的设计进行行为级模拟验证。在 top-down 的设计过程中,划分每一个层次模块时要对目标模块做优化,在实现模块时要进行模拟仿真。虽然 top-down 的设计过程是理想的,但它的缺点是得到的最小可实现的物理单元不标准,成本可能较高。

bottom-up 层次结构化设计是 bottom-up 设计的逆过程。它虽然也是从系统级开始的,即从图 0-2 中设计树的树根开始,但在层次模块划分时,首先考虑的是实现模块的基本物理单元是否存在,划分过程必须是从存在的基本单元出发。设计树最末枝上的单元要么是已经制造出的单元,要么是已经开发成功的单元,或者是可以买得到的单元。自底向上(bottom-up)的设计过程采用的全是标准单元,通常比较经济。但完全采用自底向上的设计有时不能完全达到指定的设计目标要求。用可编程 ASIC 实现一个好的电子系统设计通常采用 top-down 和 bottom-up 两种方法的结合,充分考虑设计过程中多个指标的平衡。

采用层次结构化设计自动综合工具是必须的,理想中的高级 EDA 软件开发系统应该具有强大的综合器,它能将高层次系统行为级描述自动翻译成门级逻辑电路的描述,也能将同层次的行为级描述转换为结构描述。但是,现在多数 EDA 开发软件还不能完全做到将系统行为级描述自动翻译成门级电路的结构描述。在实际设计实现时,如果系统不是特别复杂,往往是从 RTL 级即寄存器传输级开始进行设计描述。在描述之后,一般还要进行逻辑综合。经过逻辑综合,各种逻辑功能可直接用 EDA 系统元件库中的相应单元实现。逻辑综合器可以接受图 0-1 中绘出的各种输入方式。目前 EDA 技术依然处在发展阶段。

(5) 设计库及库元件

在层次化设计中所用的模块有两种，一是预先设计好的标准模块，二是由用户设计的具有特定应用功能的模块。前者一般要存放在 EDA 开发系统中各种类型的元件库之中，后者必须经过模拟仿真和调试证明无误后，建立一个图形符号后再存放在用户的设计库中准备在更上层的设计中使用。也同使用设计库中的逻辑门电路一样，设计库中的一个比较高级的模块一般都要由两个模型构成，一个是模块的图形符号，另一个是模块的功能模型。图形符号在建立原理图时使用，功能模型在逻辑模拟仿真时使用。模块的功能模型可以是逻辑图形式，也可以是 VHDL 描述的，还可以是真值表或逻辑方程式描述的。众所周知，一个 2 输入的与非门 NAND2 可以是 7400，74LS00，也可以是 74S00，这些具体的器件的功能相同，但是传输延时，功耗都各不相同。也像门电路一样，一个已知的图形符号可以用来代表一个或几个功能模型，这些模型的功能相同，参数可以不同。

(6)画层次原理图

画层次原理图类似用逻辑门符号画一个逻辑图，先将选用的模块图形符号和连接器符号放在画页上，然后用连线将它们连接起来，最后将选用的符号名放在相应的模块及其结点上。选用符号名要注意遵循以下规则：一般把在一个层次原理图中所使用的模块的每一个拷贝叫做这个模块的例化。为了模拟仿真和建立设计文件，每个例化都要起一个名字，如在图 0-3 中 4 位全加器模块 FA4 起名为 Adder；名为 Adder 的模块由 4 个一位全加器子模块 FA1 实现，这 4 个子模块分别起名为 add0，add1，add2，add3。同样，构成一位全加器的各个逻辑门及其信号线也要起一个名，它们的名字分别是 $X1, X2, A1, A2, A3, R1$。为了调试或模拟仿真，常常要研究模块中的一个指定信号。例如，假设要研究图 0-3 中 1 位全加器 FA1 的工作情况，需要观察完整系统模拟时的信号 x1 的值。因为有 4 个 FA1 的例化，例化名称要被合并成如下的信号名，顶层模块名/次层模块名：信号名。因此，要监视全加器 add2 这个例化中信号线 x1 的时候，这个信号名应该写成 adder /add2：x1，通过上述的书写规则，就可指定顶层模块中 adder 中的模块 add2 的信号是 x1，这个起名规则可以扩展到任何一个层次。

(7) 层次连接器符号和总线

为了建立层次原理图，一个抽象级别的模块输入和输出引脚的名称要与次层模块原理图相应信号的名称保持惟一性或者一致性，如图 0-4 所示。层次模块 FA1 的输入为 a, b, c_{in}，而输出是 s, c_{out}。信号的名称也要标在相同模块底层原理图的输入和输出引脚连接器符号上。  ——是输出连接器 ——  是输出连接器 一般输入连接器符号标在原理图左侧，输出连接器符号标在原理图右侧，其底层原理图输入输出引脚的层次连接器符号旁边也要标出相同的名称 a, b, c_{in}, s, c_{out}。当模块有多重的输入和输出信号时，层次的相互连接器就要画成如图 0-5 所示的总线形式。每条总线都要赋予一个名称以便于模拟仿真，由图 0-5 可以看出 4 位加法器模块 FA4 有 8 个输入和 4 个输出，为了方便，把传送二个 4 位输入和一个 4 位输出的数据信号线表示成三条总线形式，其定义如下：

$$A(3:0)=\{A(3)\quad A(2)\quad A(1)\quad A(0)\}$$
$$B(3:0)=\{B(3)\quad B(2)\quad B(1)\quad B(0)\}$$
$$S(3:0)=\{S(3)\quad S(2)\quad S(1)\quad S(0\}$$

考虑到在逻辑模拟仿真时对每个独立的信号都要进行模拟和监测，总线的每个分支信号线也要给出惟一的名称，在图 0-5 中 $A(0)$、$A(1)$、$A(2)$、$A(3)$、$B(0)$、$B(1)$、$B(2)$、$B(3)$、$S(0)$、$S(1)$、$S(2)$、$S(3)$就是独立信号线的名称。

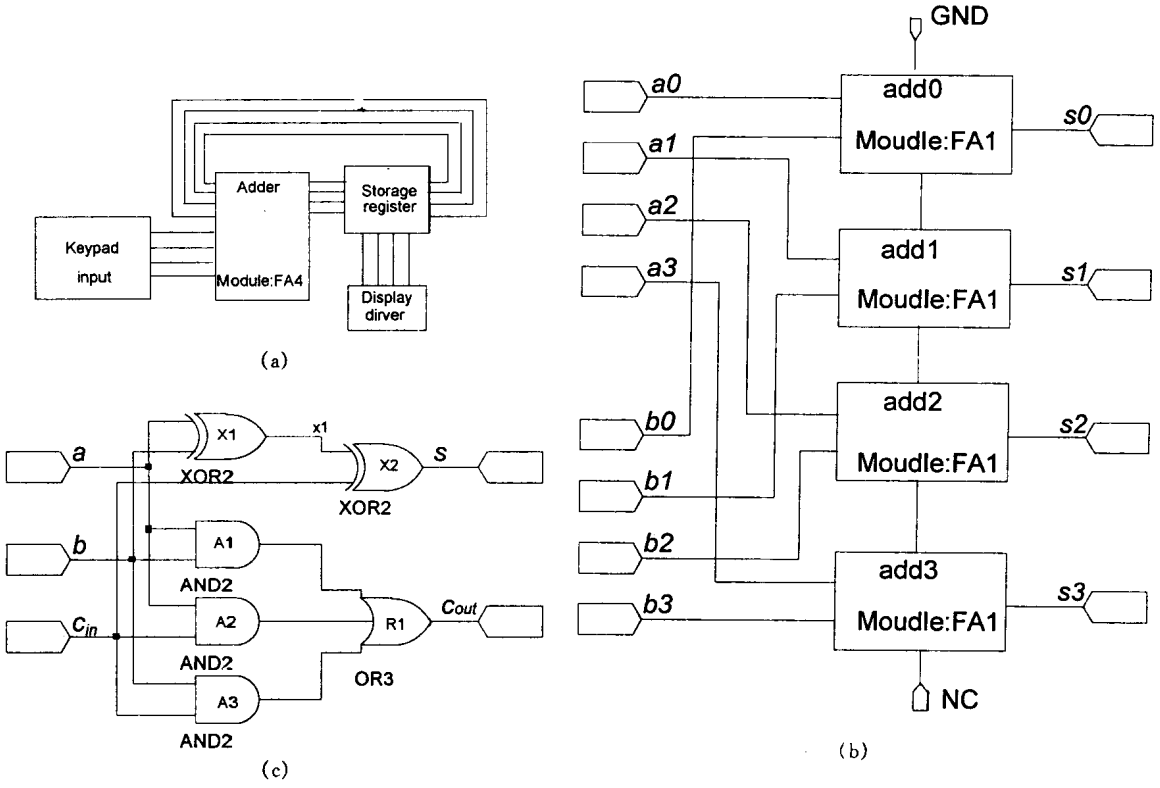


图 0-3　层次设计示意图

(a)顶层图;(b)模块 FA4 的次层图;(c)模块 FA4 的次层原理图

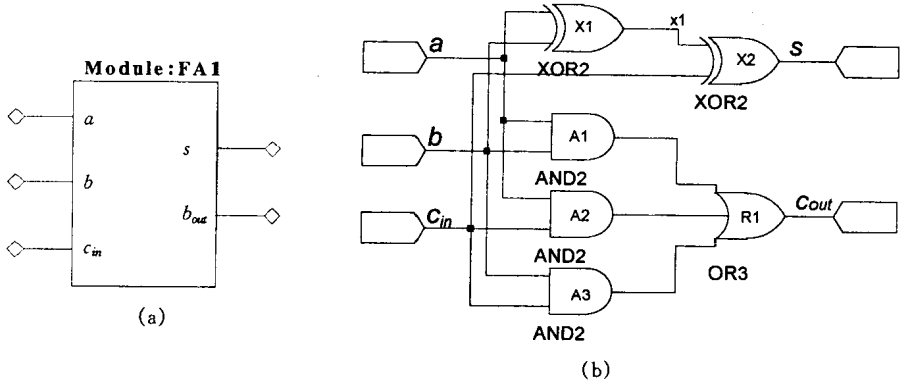


图 0-4　全加器符号与原理图之间的层次连接器

(a)全加器模块符号;(b)全加器原理图

为了进一步简化模块原理图画法,有时常常在模块上定义多重引脚,如图 0-5(b)所示。每个多重引脚代表一组相关信号的集合,允许把总线直接连接到模块的引脚上,但是必须清楚每条总线代表的是 4 个信号的连接。

(8) 层次化设计的模拟

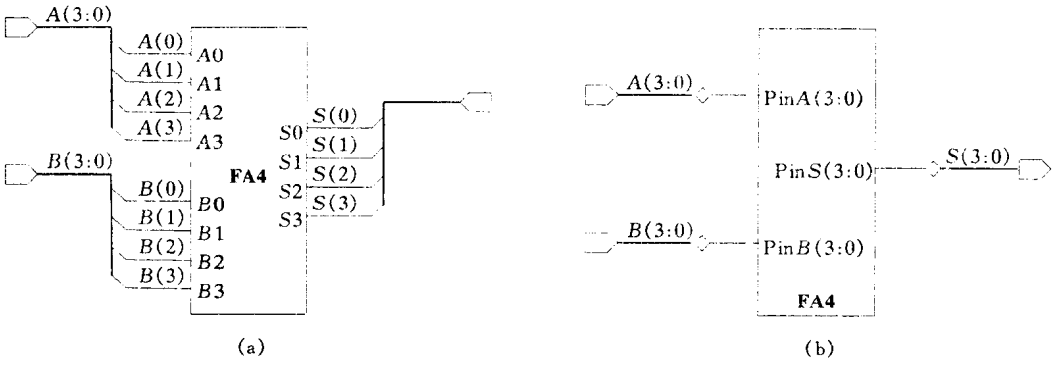


图 0-5 使用总线简化逻辑图

(a)具有分支的总线;(b)多重引脚的总线模型

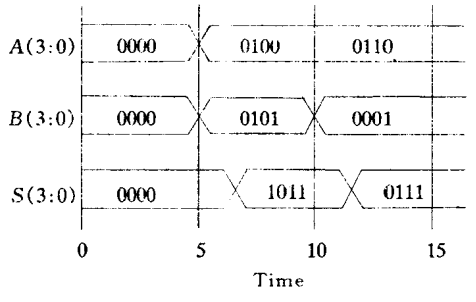
采用层次化设计实现的系统必须进行设计模拟和验证。一个层次化设计中最底层的元件或模块必须首先进行模拟仿真,当其工作正确之后,再进行高一抽象级别模块的模拟仿真。最后还要对最上层系统进行模拟仿真,最终完成系统设计。

在模拟仿真时,绝大多数的 EDA 系统首先要将模块用相应的电路来代替,这一过程一般称为展平.展平工作一直做到最底层模块都要用基本的逻辑门实现为止。在展平过程中所有元件及所有的信号线都必须有指定过的名称。

模拟仿真结果可以是给出正确的波形,也可以是给出一些时延参数。图 0-6 给出的是用总线表示的波形。

Time	A(3:0)	B(3:0)	S(3:0)
0	0000	0000	0000
5	0110	0101	0000
7	0110	0101	1011
10	0110	0001	1011
12	0110	0001	0111

(a)



(b)

图 0-6 用二进制数表示的总线模拟值

(a)表格形式的模拟值;(b)波形表示的模拟值