清华大学电子与信息技术系列教材

数字集成系统的结构化设计与高层次综合

王志华　邓仰东　编著

清华大学出版社
http://www.tup.tsinghua.edu.cn

清华大学电子与信息技术系列教材

数字集成系统的结构化设计与高层次综合

王志华　邓仰东　编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

采用结构化设计技术，可以有效地提高集成电路设计方案的可重复使用性；采用高层次综合技术，可以有效地提高数字系统的设计效率。本书讨论有效提高数字集成系统设计能力的结构化设计方法和高层次综合技术。重点讨论结构化设计思想在不同层次的应用，同时还包含了高层次综合的基本思想。前 8 章讨论数字系统的设计，第 9 章简单介绍了模拟与混合系统设计中的特殊问题。

本书可作为高等院校电子类高年级本科生与研究生的教材，也可以作为相关领域的工程技术人员的参考资料。

书　名：数字集成系统的结构化设计与高层次综合
作　者：王志华　邓仰东　编著
出版者：清华大学出版社(北京清华大学学研大厦，邮编 100084)
http：//www.tup.tsinghua.edu.cn
印刷者：北京市人民文学印刷厂
发行者：新华书店总店北京发行所
开　本：787×1092 1/16　**印张**：25.5　**字数**：582 千字
版　次：2000 年 7 月第 1 版　2000 年 7 月第 1 次印刷
书　号：ISBN 7-302-03837-6/TN・106
印　数：0001～3000
定　价：29.50 元

前言

当前，集成电路产业的发展日新月异，1999 年世界集成电路的销售额达 1570 亿美元。以集成电路为基础的电子信息产品的世界贸易总额于 1998 年就超过了 8000 亿美元，成为世界第一大产业。据国际权威机构预测，到 2012 年世界集成电路的年销售额将达到 1 万亿美元，并支持 6 ～ 8 万亿美元的电子设备和 30 万亿美元的电子信息服务，相当于今天全世界 GNP 的总和。集成电路的发展加速了人类社会的信息化进程，已经成为信息产业乃至 21 世纪实现知识经济的技术基础之一。我国微电子产业的发展，经历了自主研发创业、引进吸收提高、重点建设三个阶段。1999 年我国集成电路的总消耗量为 436 亿元，其中国产芯片的总量约为 83.8 亿元，占芯片消耗总量的 19%。预计到 2010 年我国集成电路市场的容量将达到 6000 亿元，国产集成电路的产值将达到 2000 亿元。

与集成电路产业迅速发展的同时，微电子技术也在不断改进，电路性能迅速提高。根据国际权威机构估计，2010 年动态存储器(DRAM)的存取时间将降低至 10ns 以下；电源电压有可能降至 0.6V，数字电路的时钟频率可以提高到 3GHz。与此同时，集成度和芯片规模也在提高，工业化生产的典型硅集成电路的管芯面积将达 2.5cm×2.5cm，特征尺寸将下降为 0.07μm。对于以 DRAM 为特征的硅存储器，集成度将达到 8.5Gbit/cm^2；全定制设计的数字系统，集成度将达 25Mgate/cm^2；电子系统集成中常用的标准单元，集成度也将达到 10Mgate/cm^2。

随着微电子技术的进步，人们对集成系统的需求也在提高。计算机、通信、消费类电子产品、工业及军事等各种领域都大量需要集成电路，比如在现代战争中，指挥、通信、计算机与情报获取都必须以集成电路为核心。在军舰、战车、飞机、导弹和航天器中集成电路的成本分别占到总成本的 22%、24%、33%、45%和 66%。在这些领域大量需求的是集成系统，即将完整的电子系统或子系统集成在单个芯片内。单个芯片内可能集成了微控制器核心(MCU core)、数字信号处理器核心(DSP core)、存储器核心(memory core)、射频处理器(RF processor)、数模和模数转换器等嵌入式硬件以及完成特定功能所必须的嵌入式软件。

对集成系统需求的提高也对设计者提出了挑战。集成系统的规模通常在 1000 万个等效门以上，完成集成系统设计，除了设计时间很长之外，还要花费高额的一次性研制(nonrecurring engineering，NRE)费用，因此，保证首次投片成功非常重要。对于高技术产业，新产品上市时间(time to market)的压力越来越大，如果由于产品的开发周期过长而错过了时机，那么将花费很大的代价才能挤入市场，甚至会彻底失去产品市场。

技术的发展使得集成电路的制造能力不断提高，工业的发展需要规模越来越大、性能

越来越好的集成电路，设计能力不足已成为充分利用集成电路的制造能力，满足社会需求的瓶颈。为了提高设计能力，可以采取两条途径：① 采用更为结构化的设计方法学，使得设计者可以大量重复使用他人过去设计的部件。② 采用高层次设计工具，使得设计者可以在高层次开始设计。

本书主要讨论数字集成系统的结构化设计方法。数字系统可结构化地划分为从顶向下的不同层次，在每个设计层次，设计者需要构造系统的可仿真模型，需要对系统进行结构性划分，还需要对划分得到的子系统构造可仿真模型。对于每个设计层次构造的系统模型，向上可以作为更大系统的子系统，使更大系统的设计进程独立于本层次的设计过程；向下可以作为子系统的设计规范，使各个子系统的设计过程相互独立。采用结构化的设计技术，数字集成系统的设计就可以重复使用不同层次的 IP(intellectual property)芯核，完整的数字集成系统的设计就不再局限在单一的设计部门，甚至不再局限于同一个企业内部。在完整的系统中，各个 IP 芯核全由相应的专家完成，通过技术的转移即可迅速地实现系统。采用结构化的设计技术，可以有效地提高设计方案的可重复使用性，从而提高人们的设计能力。本书还介绍了数字系统高层次综合的技术，包括算子调度、资源分配的基本方法和技术。为读者使用高层次设计工具进行数字系统的设计建立了理论基础。

全书共分 9 章。第 1 章是引言，首先分析了电子系统集成的瓶颈所在，讨论了系统设计的重要性。然后介绍了集成系统的设计层次，讨论了各设计层次的问题、描述方法及使用的设计工具，在讨论了数字系统设计的不同描述方法之后，简单介绍了系统的分解、设计和综合的概念。

第 2 章介绍使用硬件描述语言 VHDL 构造数字系统的模型的技术，对使用 VHDL 进行结构化集成电路设计和综合的过程进行了解释。VHDL 本身的语法及其特征已有大量图书资料进行介绍，本书只简单讨论用 VHDL 构造硬件模型时应特殊考虑的问题。

第 3 章主要介绍了基本逻辑单元的 VHDL 模型，在介绍了各种组合逻辑电路和时序逻辑电路的模型之后，讨论了逻辑门级模型的精度问题。为了能够精确构造数字系统的模型，使模型精确描述时序关系，提高仿真效率和模型的通用性，该章还介绍了工业标准的 VITAL、预定义库、建模指导方针和从标准延时格式获得反向标注的语法。最后讨论了多值逻辑的需求，并介绍了 IEEE 9 值逻辑。

第 4 章讨论系统级设计问题，即把硬件的自然语言描述转换为真值表、状态图或算法模型的过程。结合设计实例对进程模型图、子系统间的互连、时分复用等问题进行了讨论，并重点介绍了如何将硬件系统的自然语言描述转换为算法模型。最后以与 Intel 8031/51 兼容的 8 位微控制器设计为例说明了系统设计的概念。

第 5 章讨论集成电路的寄存器传输级设计，包括如何将数字系统划分为控制单元和数据路径两部分，以有限状态机方式和微码 ROM 两种方式描述控制单元，以及如何用 VHDL 数据流方式和寄存器级单元序列设计数据路径，再以互连的形式将两部分连接成完整的系统。作为设计实例，该章详细讨论了超级精简指令集计算机 URISC 的设计，并对第 4 章中的实例 Intel 8031/51，完成了从系统级到寄存器级的变换。由于数字系统设计过程中主导的设计方法是把寄存器传输级代码作为综合器的输入，因此，该章还从综合的角度讨论了寄存器级的 VHDL 代码与综合得出的电路的关系。

第 6 章讨论组合逻辑及时序逻辑电路的多层次混合设计问题，即如何混合使用硬件描述语言、卡诺图、状态表等多种方法设计组合逻辑及时序逻辑电路。涉及的电路设计覆盖了系统级的算法描述直至逻辑门级电路描述。此外，该章用较多的篇幅讨论了微代码控制单元的设计问题，介绍了一个基本微代码控制器 BMCU，并设计了该控制器的微代码，用以实现串并转换电路的控制单元。

采用高层次设计工具，可以有效提高数字设计能力。第 7 章讨论高层次综合问题，即从系统级的行为描述出发，由 EDA 工具经过一系列自动转换，生成寄存器传输级描述。其中，行为级描述是设计对象的功能或算法表示，而寄存器传输级描述则是实现这些功能或算法的某个硬件结构的表示。该章首先概要地介绍了行为综合的发展，然后论述行为级描述的内部表示（中间格式）和行为级综合的目标体系（目标结构），最后讨论行为综合的流程和行为级 VHDL 建模，并以一个实例说明了行为综合的建模过程。

高层次综合或者算法综合的意义是把硬件的抽象行为描述自动转换为寄存器级模型。对于给定的系统级的算法描述，由自动综合可以转换为多种不同寄存器级模型，而自动综合本身也是一个困难的、不成熟的而同时又很有吸引力的研究领域。第 8 章介绍综合过程中需要的调度和分配算法，对各种调度和分配算法的优点及局限性进行了讨论。

第 9 章讨论硬件描述语言在数字系统设计领域之外的应用，包括数模混合信号系统、超大规模集成电路测试、微机械/微结构设计、性能分析等方面。这部分的研究已吸引了世界各地的学者进行广泛的研究，并已经取得了一些成果。

阅读本书要求读者具备数字电路与逻辑设计、程序设计语言或 VHDL 语言等方面的基础知识，尽管 VHDL 语言贯穿在全书之中，但并不要求读者在阅读本书之前全面掌握 VHDL。随着本书对硬件造型内容的深入讨论，对 VHDL 的语言特征，特别是与硬件设计和综合密切相关的语言特征也逐步进行了介绍。

本书是在清华大学电子工程系研究生课程讲义的基础上改写而成，它是清华大学电子工程系有关教师长期工作的积累，许多教师对本书的内容以及课程本身提出过许多宝贵的建议，包括刘润生教授、董在望教授、刘序明教授和汪蕙教授等。此外，刘宝琴教授认真地审阅了原稿，在此一并感谢。

本书取材于国际上关于结构化集成电路设计的最新专著及文献，综合了分散于不同文献中的知识和设计经验，也包含了笔者在清华大学期间的研究工作的总结。笔者有幸通过本书的编写与读者交流数字集成系统设计方面的体会，并期望为我国集成系统设计人才的培养与信息产业的发展贡献微薄之力。书中出现的错误和不足，欢迎指正。

作　者

2000 年 2 月

目 录

第1章 引 言

超大规模集成电路(VLSI)可以划分为通用集成电路和专用集成电路两大类。存储器等大量生产且设计规则者称为通用集成电路,面向某一应用背景而专门设计者称为专用集成电路(ASIC)。目前,专用集成电路设计中面临着设计能力几乎没有改进而设计复杂度却不断提高的矛盾。统计表明,对于一个规模较大的专用集成电路设计,通常由10～15个人在18个月的时间内完成,每个设计者的平均设计能力为每天十几个器件。与此同时,专用集成电路的设计复杂度却在成指数增加,1984年专用集成电路的设计所要求的电路规模大致在几十万个晶体管,1996年的专用集成电路要求的设计规模达到了100万至200万个晶体管,估计到2000年专用集成电路要求的设计规模会达到700万至800万个晶体管,到2010年要求的设计规模将达4000万个晶体管。按目前的设计能力,使用最先进的逻辑和寄存器变换层次的综合工具,10～15个人在18个月中能完成的设计规模大致为100万～200万个晶体管。这意味着如果不改变设计方法,到2010年,设计能力与设计需求之间的差距会达20～40倍。显然,人们必须改进设计方法,在保证设计质量的前提下提高设计能力,即提高专用集成电路设计的"生产率"。实现这一目标有两条途径:

(1) 采用高层次设计工具,使得设计者可以在高层次开始设计。现在寄存器变换层次的综合工具已经被设计者广泛接受,今后要采用更高层次的行为综合工具(有时称为结构综合或者高层次综合)。

(2) 采用更为结构化的设计方法学,使得设计者可以大量重复使用前人过去设计的部件。统计表明,新设计中通常会有70%是过去用过的部件,现在由于设计方法不合适或者缺乏设计工具而不得不重新设计这些部件。

1.1 设计层次

标准硬件描述语言(HDL)的出现(比如VHDL和Verilog语言),促进了集成电子系统的高层次设计。硬件描述语言可以用来描述完整的电子系统,也可以描述电子系统的子系统,可以在一个层次化的设计描述中把多个子系统连接在一起。过去十几年间,开发出了许多电子设计自动化工具,包括综合和仿真工具,这些工具的采用,有助于在高层次上描述和设计集成电路。今天硬件描述语言已应用到从逻辑门到行为等各种抽象层次的设计描述中,并已为集成电路设计界的广大技术人员所接受。

在集成电路的设计过程中,时序是一个重要的概念。事实上,如果不特别强调抽象层次,则集成电路的设计过程可以看作是时序描述逐次精确的过程,即从高层次的概念描述

(运算、结构、语句、原始单元等)到可详细给出时序特征的基本单元描述的变换过程。通常将集成电路设计的层次划分为 6 个层次,如表 1.1 所示。最高层次为系统级,最低层次为版图级(又称物理级)。硬件设计方案可以在任何层次描述。

表 1.1 集成电路设计的层次

抽象层次	时序单位	基本单元	电路的功能(行为)描述
系统级	数据处理	进程及通信	自然语言描述或者相互通信的进程
算法级	运算步	运算的控制	行为有限状态机、数据流图、控制流图
寄存器变换级	时钟周期	寄存器、运算、变换	布尔方程、二元决策图、有限状态机
逻辑门级	延时	逻辑门、器件(晶体管)	原理图
电路级	物理时间	晶体管、R、L、C 等	电压、电流的微分方程
物理(版图)级		几何图形	

设计描述的最低层次为版图级。版图级用几何图形描述硬件,即用硅表面上的扩散区、多晶硅和金属等来表示硬件。在这一层次,硬件的描述只是结构,而硬件的功能则隐含在器件的物理方程中。

版图级层次之上的层次是电路级。电路级用有源器件和无源器件的互连关系描述硬件。电路级的基本单元可以为双极型晶体管、MOS 晶体管、电阻、电容等。器件的互连关系描述了电路的结构,电压电流之间所满足的微分方程描述了电路的功能。

电路级之上是逻辑门级,这是数字系统设计的主要层次。在逻辑门级设计时,电路的基本单元通常是与门(AND)、或门(OR)、异或门(XOR)、反向器等逻辑门,有时还带有少量晶体管构成的开关以及 D 触发器、J-K 触发器、锁存器等逻辑单元。在这一层次,基本的时序单元是延时。基本电路单元的互连构成逻辑门级的结构描述。这一层次最典型的描述是原理图。当然,也可以用 VHDL,Verilog 这样的硬件描述语言来描述,设计所需元件(逻辑门、线网、晶体管器件)的特征由延时描述。为了计算整个设计的性能,需要仿真工具,也需要时序分析工具。时钟周期通常由两个存储器件之间的最长路径确定,一条路径上可能包含多个非存储部件。布尔方程是这一层次上行为描述的基本形式。

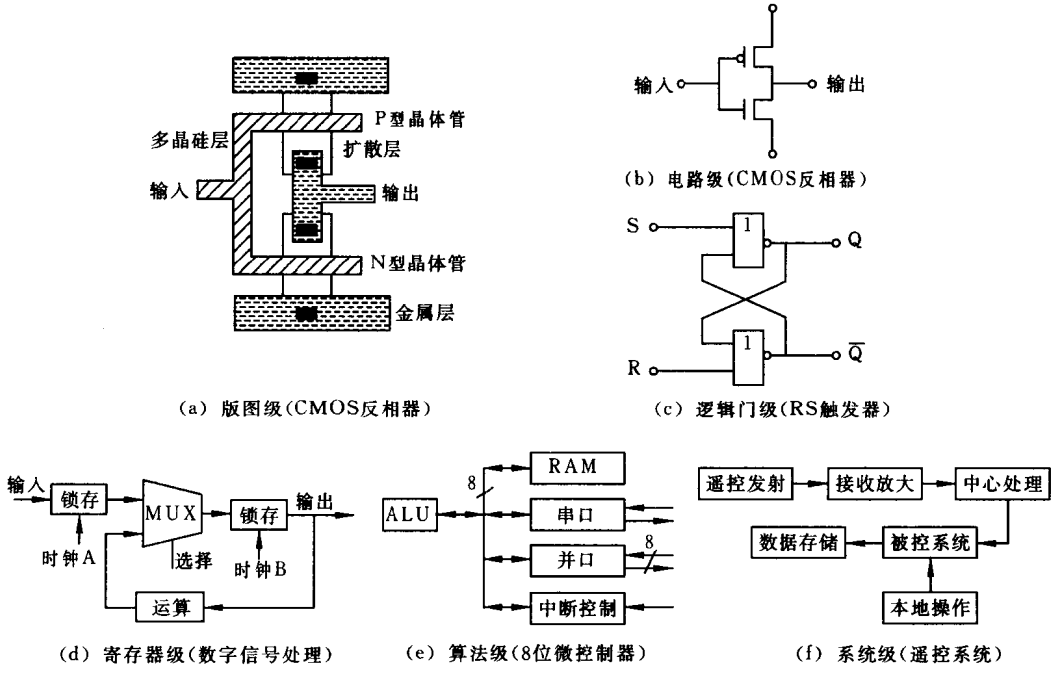
逻辑门级之上的层次是寄存器变换级(RTL 级)。寄存器级最基本的设计单元是寄存器、计数器、多路选择器、算术逻辑单元(ALU)等。这里的基本单元有时也称为功能块,相应于 VLSI 设计时的宏单元,因此 VLSI 设计的寄存器级也称为宏单元级。尽管寄存器级设计的基本单元也可以继续展开成基本逻辑门,但在寄存器级设计时通常不作这种展开,而是把它们作为整体处理。寄存器级设计的结构描述是其基本单元的互连。由于寄存器级设计时基本单元的行为通常由真值表或状态表描述,这一层次上完整硬件的行为通常也由真值表或状态表描述。有时将这一层次的行为描述成为数据流图,它反映了流经实际硬件的数据分布。在寄存器变换级,通常在时钟周期意义上定义系统。典型的描述方法为在哪个时钟周期中完成哪些运算。在这一层次上,可以用于综合工具的典型描述为布尔方程、有限状态机(FSM)和二元决策图(BDD)。这种描述可以从电路的硬件描述语言中自动提取出来。这一层次的主要设计工作是优化、综合、状态编码以及工艺映射。这里的主

要优化工作是减少完成运算所需要的时钟周期数和逻辑门数。优化过程通常要在这两个指标之间进行折衷。如果使用硬件描述语言 VHDL 写出一个系统的 RTL 级描述，总是假定两个 wait 语句之间的所有运算都可以用互连的逻辑门实现，且这些逻辑门可以在一个时钟周期内完成相应的运算。将时钟周期分解成延时单元的工作由 RTL 综合器自动完成。当然，集成电路设计者可以通过写出不同风格的 VHDL 源代码或者设定不同的约束条件控制综合结果。不同风格的描述包括同步描述和以时钟周期为基础的描述。

RTL 级之上的层次是行为级，也称为算法级。在这一层次，设计由运算步来描述。运算语句和控制语句(loop，wait)被用来组织输入、输出以及不同运算。这一层次的电路描述通常是事件驱动的描述，一般由外部世界和内部运算之间交换数据的协议构成。运算步指的是两次相邻的输入、输出或者同步点之间的数据处理。一个运算步通常会占去多个时钟周期。某些情况下运算步可能包含与输入数据有关的运算，这意味着运算步的计算时间不可预测。这一层次的典型描述方法是行为有限状态机(BFSM，behavioral finite state machine)、控制流图(CFG，control flow graph)、数据流图(DFG，data flow graph)和控制数据流图(CDFG，control data flow graph)。

层次化设计的最高层次是系统级。系统级的基本单元可以是计算机、磁盘驱动器、总线接口等大的数字单元。在系统级，行为描述的内容通常为一些性能指标。对于计算机系统，性能指标可能为每秒兆指令(MIPS)、总线传输速率等。对于通信系统，性能指标可能为系统带宽、传输速率等。如果性能指标是确定性物理量，通常在系统级要采用很高级的数据类型，比如复数、实数、时间、频率等等。

图 1.1 给出了数字系统设计时不同层次中所采用的基本单元实例。图(a)所示为最低



(a) 版图级(CMOS反相器)　(b) 电路级(CMOS反相器)　(c) 逻辑门级(RS触发器)

(d) 寄存器级(数字信号处理)　(e) 算法级(8位微控制器)　(f) 系统级(遥控系统)

图 1.1　集成电路设计的不同层次中所采用的基本单元举例

层次，即物理级（版图级）层次，它是一个 CMOS 反相器的物理版图，它由分布在硅片表面的不同层的几何图形描述，器件的行为隐含在器件几何尺寸、工艺参数和器件物理方程中。图(b)为电路级层次，CMOS 反相器由互连的两个 MOS 晶体管构成，电路行为隐含在电路方程中。图(c)为逻辑门级层次，反相器就是一个逻辑器件，其行为由布尔方程描述。在更高的层次，电子系统由更大的模块构成，其实例如图(d)，(e)，(f)所示。

对于大部分硬件描述语言，都允许在中间三个层次（逻辑门级、RTL 级、算法级）描述电路。在每个层次相应于使用硬件描述语言的一个子集写出系统的描述。当描述一个大型系统时通常将不同层次的描述方法混合使用，很少单独采用某一层次的描述。一般情况下，完整的系统由多个模块构成，每个模块分别在不同层次描述。例如，图 1.2 给出了一个电话答录机的方框图。答录机工作于电话线和电话机之间，它通过一个 D/A 和 A/D 与这些外部环境相连接，D/A 和 A/D 由逻辑门层次和电路层次描述。电话答录机还连接到磁带录音接口和定时电路，这两个模块在 RTL 层次描述。电话答录机的中央控制单元则在行为（算法）层次描述。

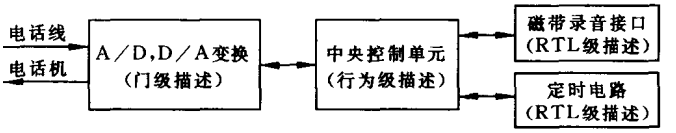


图 1.2　多层次描述示例（磁带式电话答录机）

由于在不同层次描述了一个系统，所以需要不同层次的设计工具。比如，在设计 A/D 和 D/A 转换器时，需要原理图输入工具、电路和逻辑仿真工具以及布图工具；设计两个接口部分时需要 RTL 层次的设计工具（比如逻辑综合工具）；中央控制单元的设计则需要高层次设计工具。

1.2　设计描述的文字表示及图形表示

按照集成电路的描述方式划分，设计方案可以由图形表示，也可以由文字表示，两种

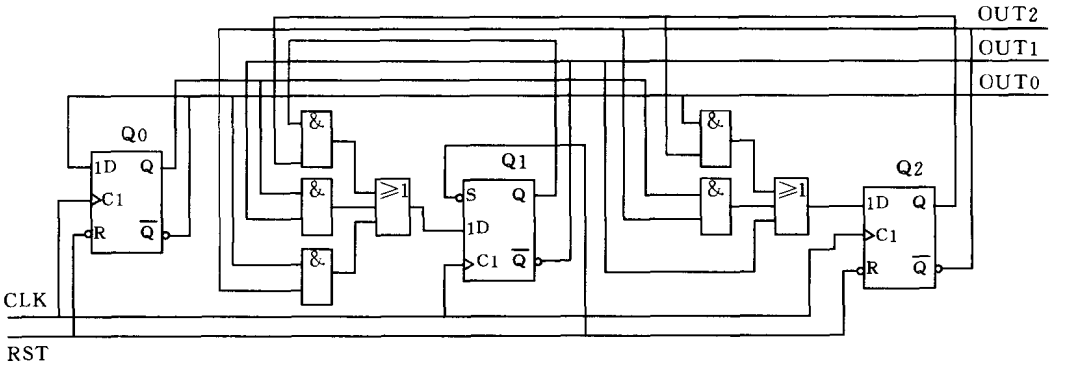
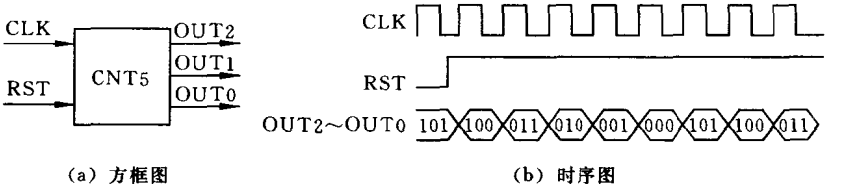
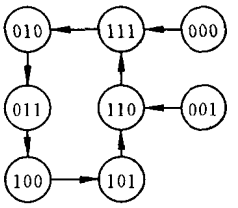


图 1.3　同步 5—0 减法计数器 CNT5

方式各有其优点。过去，对于数字电路设计，人们更喜欢图形方式，即用方块图、时序图、逻辑图（原理图）描述硬件。然而有了硬件描述语言之后，文字描述变得越来越重要。按照集成电路描述的内容划分，可以分为结构描述和行为描述。结构描述用基本单元表示设计方案，行为描述采用算法表示电路的功能。图 1.3 是一个二进制同步减法计数器，在复位状态时计数值为 5，在计数状态下，每输入一个时钟脉冲，计数值减 1。这是典型的结构描述。（在以后的讨论中，把此电路称为 CNT5。）同一电路可由图 1.4 所示的方框图、时序图、状态转换图或状态表来描述，这些都是硬件系统的图形表示。其中方框图是一种结构描述，时序图、状态转换图、状态表则是行为描述。



(a) 方框图　　(b) 时序图



(c) 状态转换图

计数值			当前状态			下一状态		
OUT2	OUT1	OUT0	Q2	Q1	Q0	Q2	Q1	Q0
1	0	1	0	1	0	0	1	1
1	0	0	0	1	1	1	0	0
0	1	1	1	0	0	1	0	1
0	1	0	1	0	1	1	1	0
0	0	1	1	1	0	1	1	1
0	0	0	1	1	1	0	1	0

(d) 状态表

图 1.4　电路 CNT5 的图形描述方式

除了图形表示方式之外，还可以用文字描述硬件。与图形描述一样，文字描述也可以分为行为描述和结构描述。硬件的不同表述方法如图 1.5 所示。图中硬件描述可以按描述方法分为文字描述和图形描述，文字和图形都可以用来描述电路结构，也可以用来描述电路行为。硬件描述可以按描述内容分为结构描述和行为描述，结构描述和行为描述都可以用文字方式表述，也都可以用图形方式表述。

文字描述可以是自然语言、方程（布尔方程或微分方程）以及硬件描述语言等。这里所谓硬件描述语言指的是用于硬件造型的具有特定结构的高级编程语言。对于上述电路 CNT5，可以分别写出其自然语言描述和 VHDL 描述。

用自然语言描述如下：

电路 CNT5 有两个输入端口和三个输出端口，其功能为二进制减法计数器。如果复位信号 RST 输入低电平，则计数值复位为 5（OUT2～OUT0 为 101）。在复位信号 RST 输入为 1 的前提下，电路对于时钟 CLK 的输入脉冲实现循环减法计数。

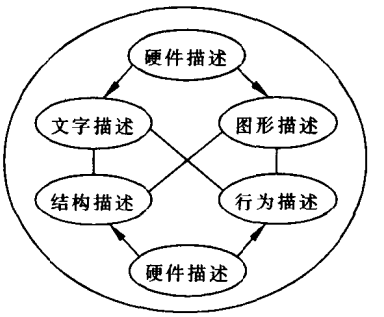


图 1.5　硬件的不同描述方法

用 VHDL 行为描述如下：

```
library IEEE;
use IEEE.Std_logic_1164.all;
entity CNT5 is
    port (CLK, RST: in Std_logic; OUT: out Std_logic_vector(2 downto 0);
end CNT5;

architecture DATAFLOW of CNT5 is
    signal Q2, Q1, Q0: Std_logic;
    variable D2, D1, D0: Std_logic;
begin
    D2 := (not Q1) or (not Q2 and Q0) or (Q2 and not Q0);
    D1 := (not Q2 and not Q0 ) or (not Q1 and Q0) or (Q2 and Q1);
    D0 := nor Q0;
    STATE: block(CLK = '1' and (not CLK'STABLE) or RST = '0')
    begin
        Q2 <= guarded '0' when RST = '0' else D2;
        Q1 <= guarded '1' when RST = '0' else D1;
        Q0 <= guarded '0' when RST = '0' else D0;
    end block STATE;
    OUT <= (not Q2) & (not Q1) & (not Q0);
end DATAFLOW;
```

上述 VHDL 源代码中使用了 IEEE 颁布的 1164 标准程序包中定义的数据类型 Std_logic 和 Std_logic_vector。本书的 VHDL 程序中，除了特别声明之外，都使用这个程序包。使用这个程序包时，需要在代码中书写下述的两行代码：

```
library IEEE;
use IEEE.Std_logic_1164.all;
```

为了书写简单，本书中经常略去这两行代码。

文字形式是描述硬件行为的重要方式，而文字形式的主要语言是硬件描述语言。用硬件描述语言描述硬件的行为，通常有两个主要类型：算法描述型和数据流描述型。它们的定义如下：

算法描述型：通过定义输入/输出响应的方式描述硬件行为，行为描述与硬件的物理实现无关。

数据流描述型：描述硬件行为时，数据的相关性与硬件的物理实现一致。

根据上述定义，硬件行为的算法描述是纯过程或者说是一段程序，用以检查硬件功能是否正确，并不关心如何制造出这个器件；而硬件行为的数据流描述则表明了数据如何在硬件内部流动。一般可以认为，数据流型硬件行为描述是硬件的寄存器级实现，算法型硬