电路设计自动化丛书

VHDL 硬件描述语言

辛春艳　编著

内 容 简 介

本书全面地介绍了 VHDL 硬件描述语言的基本知识和利用 VHDL 进行数字电路系统设计的方法。全书共分 13 章：第 1～6 章主要介绍 VHDL 语言的基本语法知识；第 7～9 章介绍利用 VHDL 设计组合逻辑电路和时序逻辑电路（包括状态机）的基本方法；第 10、11 章简单扼要地介绍了 VHDL 设计中的仿真和综合的内容；第 12 章介绍 ALTERA 公司的 MAX＋PLUS II 开发工具的使用；第 13 章给出了 3 个 VHDL 层次性设计的实例，以进一步提高读者学习和使用 VHDL 的能力。

本书注重基础知识的介绍，力求向读者系统地讲解 VHDL 硬件描述语言的使用。它可以作为具有一定的数字电路基础知识的大学本科和研究生用书，也可供从事电路设计的工程人员参考。

图书在版编目(CIP)数据

VHDL 硬件描述语言/辛春艳编著．—北京：国防工业出版社，2002.1
（电路设计自动化丛书）
ISBN 7-118-02741-3

Ⅰ.V… Ⅱ.辛… Ⅲ.硬件描述语言，VHDL—程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2001)第 097346 号

（北京市海淀区紫竹院南路 23 号）
（邮政编码 100044）
北京奥隆印刷厂印刷
新华书店经售
*
开本 787×1092 1/16 印张 18½ 425 千字
2002 年 1 月第 1 版 2002 年 1 月北京第 1 次印刷
印数：1—4000 册 定价：25.00 元

目　录

第 1 章　EDA 与硬件描述语言

1.1　电子设计自动化 (EDA) 技术

1.1.1　EDA 的发展

EDA（Electronic Design Automation，即电子设计自动化）技术可以看做是电子 CAD 技术的高级阶段。20 世纪 70 年代，随着中小规模集成电路的开发应用，传统的手工制图设计印刷电路板和集成电路的方法已无法满足设计的精度和效率的要求。因此工程师们开始进行二维平面图形的计算机辅助设计，以解脱复杂机械的版图设计工作，这就产生了第一代 EDA 工具。80 年代，电子产品的规模和复杂程度的增加促使第二代 EDA 工具的产生。第二代 EDA 主要以计算机仿真和自动布局布线技术为核心，与此同时，还引出了以半定制概念为特征的专用集成电路概念，集成电路制造业在 80 年代发生了一场革命性的变化。

第二代 EDA 应用软件主要有数字电路分析、模拟电路分析、印刷电路板、现场可编程门阵列的布局布线等等。它们的主要特点是以软件工具为核心，即针对产品开发分为设计分析生产测试等多个独立的软件包，每个软件只能完成其中的一项工作，通过顺序循环完成设计的全过程。但是由于软件开发商的不统一，一个工具的输出作为另几个工具的输入需要进行界面处理，影响了设计的速度。并且第二代 EDA 工具不能进行系统级的仿真与综合，在产品开发的后期才发现设计错误时，进行修改无疑十分困难且浪费大量的人力。

EDA 技术发展到 90 年代，出现了以高级语言描述、系统级仿真和综合为特征的第三代工具。新一代的 EDA 技术使设计者摆脱了大量的辅助性工作，把精力集中于“要设计什么”，而不是“如何设计”。借助第三代 EDA 工具，设计人员方便地把自己的新构思和新方案转化为新产品，产品的研制周期得以大大缩短。同时，第三代 EDA 工具进一步促进了专用集成电路的设计与发展。第三代 EDA 是由自顶向下的设计思想，PCB 布线和设计可制造性分析，数字与模拟混合信号电子系统的分析和设计仿真，DSP 设计等软件包与面向用户的统一的数据库及管理框架共同组合而成的。第三代 EDA 系统主要以并行设计工程的方式和系统级目标设计方法为支持。系统设计的核心是可编程器件的设计。由于可编程器件自身的可重复编写的特性，使电子设计的灵活性和工作效率大大提高。

著名的 EDA 软件厂商主要有 Cadence Design Systems, Mentor Graphics, Viewlogic Systems, Synopsys, Zuken,Compass Design Automation, Logic Modeling 等。

但是第三代 EDA 技术面临新的挑战。当前 EDA 技术主要用于数字设计，而对模拟

电路的仿真功能有限。与数字电路仿真技术的飞速发展相比，模拟电路的仿真仍然是spice，但模拟电路在电子产品的数字化进程中某些电路是不可替代的，因此 IEEE 已经努力改变模拟电路 EDA 工具落后的情况。

1.1.2 层次性设计技术

传统的电路设计方法是自底向上的，即利用已有的逻辑元器件进行逻辑设计，完成各模块后进行连接，最后形成电路系统。设计时根据基本的电路知识选择适合设计者习惯的、价格合理的、使用方便的逻辑元器件，根据电路要完成的功能，画出真值表，利用卡诺图得到各元件输入输出之间的关系，实现各单元器件之间的连接，最后完成整个硬件系统的设计。通常采用电路原理图输入进行设计。自底向上的设计方法进行仿真和调试的仪器一般为逻辑分析仪和示波器等，因此只有硬件系统实际构成之后才能进行仿真和测试。基于以上步骤可以看出，尽管众多的电子工程师对传统的硬件电路设计方法比较熟悉，并有一些 CAD 辅助工具，但这种自底向上的设计方法对设计人员的电路知识要求较高，设计周期长，采用原理图输入的方法阅读、修改困难，在大规模乃至超大规模系统的设计中这种设计方法远远不能满足市场的要求。

基于 EDA 技术的所谓的自顶向下的设计方法，设计过程的层次化、结构化使得设计能力有了很大的提高。层次化、结构化的设计方法中，完整的硬件设计划分为一些可操作的模块，允许多个设计者同时设计一个硬件系统中的不同模块，每个设计者只负责自己所承担的部分。数字系统的设计中，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，硬件系统由比它原始的基本单元的互连来描述；在行为域中，硬件系统由它们的输入输出关系描述。

通常将数字系统设计分为 6 个层次，见表 1.1。最高层为系统级，最低层为版图级或物理级。硬件设计方案可以在任何层次上进行描述。

表 1.1 集成电路设计的层次

设计层次	行为描述	结构描述	设计考虑
系统级	自然语言描述的性能指标	方框图	系统功能
芯片级	算法	微处理器、RAM、ROM 等组成的方框图	时序、同步、测试
寄存器级	数据流图、有限状态机、状态表、状态图	寄存器、ALU、计数器、MUX、ROM 等	时序、同步、测试
逻辑门级	布尔方程、卡诺图、Z 变换	逻辑门、触发器	选用适当的基本门实现硬件
电路级	电压、电流的微分方程	晶体管、R、L、C 等	电路性能、延时、噪声
版图级	几何图形与工艺规则		

版图级采用几何图形描述，它们用来表示硅表面上的扩散区、多晶硅和金属等。从硬件设计者的角度看，这是单纯的结构描述。

版图级之上是电路级。电路级表示了一般有源器件和无源器件之间的互连关系。电路级的基本单元可以是双极晶体管、MOS 晶体管、电阻和电容等。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电路的行为。

电路级之上是逻辑门级，这是数字系统设计的主要级。在门级设计时，基本单元为与、或、异或、反向器等逻辑门，以及 D 触发器、JK 触发器、锁存器等基本逻辑单元，这些基本单元的互连构成了门级逻辑的结构描述。布尔方程是这一级行为描述的基本形式。

逻辑门级之上是寄存器级。这时基本的设计单元是寄存器、计数器、多路选择器、算术逻辑单元等。这里的基本单元有时也称为功能块，相对于 VLSI 设计中的宏单元，因此寄存器级也称为宏单元级。尽管寄存器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做这种展开，而是把它们作为整体来处理。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，这一级上完整硬件的行为通常也由真值表或状态表表述。有时将这一级的行为用数据流图来表达，它反映了通过实际硬件的数据分布。

芯片级是寄存器级之上的一个层次。芯片级结构描述的基本单元是微处理器、存储器、串行接口、并行接口、中断控制器等大的电路单元。芯片级设计最主要的要求是无论用了多大的基本单元，必须能完备地描述其行为。芯片级的行为描述内容通常是器件的输入输出响应、芯片实现算法等。硬件描述语言通常用于这一级的行为描述。

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、A/D 采集卡、总线接口等大单元。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，性能指标可以是兆每秒指令、总线传输速率等；对于通信系统，性能指标可以是系统的带宽、传输速率等。如果性能指标是确定性物理量，通常在系统级设计要采用较高级的数据类型，如复数、实数、频率等。

在 ASIC 设计中,通常系统级就是芯片级设计。因此，通常对系统级和芯片级不加区分。由于可编程集成电路由内部的开关元素决定其功能，因此不需要掩膜级和电路级的设计，而是由门级网表直接映射为可编程集成电路的内部结构，得到控制开关元素的数据。因为省去了需要集成电路专门知识和经验的版图设计，简化的设计流程和快速的成片方式为整机工程师设计自己所需的专用集成电路打开了方便之门。

1.1.3　EDA 技术的基本特征和工具

1. EDA 技术的基本特征

现代 EDA 技术的基本特征是采用高级描述语言描述，具有系统级仿真和综合能力。

? 自顶向下设计方法

自顶向下的设计方法首先从系统设计入手，在顶层进行功能方框图的划分和结构设计。在方框图级进行仿真、纠错，并用硬件描述语言对高层次的系统行为进行描述，在系统一级进行验证，然后用逻辑综合优化工具生成具体的门级逻辑电路的网表，其对应的物理实现级可以是印刷电路板或专用集成电路。与传统的自下向上的设计方法相比，自顶向下设计方法将有利于在设计的早期发现结构设计中的错误，提高设计的一次成功率。

? 硬件描述语言

用硬件描述语言进行电路系统设计是当前发展的趋势。与传统的原理图输入设计方法相比，硬件描述语言更适用于规模日益增大的电路系统，它还是进行逻辑综合优化的重要工具。硬件描述语言使得设计者在比传统的门级设计方法更为抽象的层次上描述设计的结构和内部特性。它的突出的优点有：语言的公开可利用性，设计与工艺的无关性，宽范围的描述能力，便于组织大规模系统的设计，以及便于设计的复用和继承等等。目前最常用的硬件描述语言有 VHDL 与 Verilog—HDL，它们都已经成为 IEEE 标准。另外还有一些 EDA 厂商自行开发的语言，如 M，UDL/I 等。

? 逻辑综合优化

逻辑综合功能将高层次的系统行为设计自动翻译成门级逻辑的电路描述，做到了设计与工艺的独立。优化则是对于上述综合生成的网表，根据布尔方程功能等效的原则，用更小更快的综合结果替代一些复杂的单元，根据指定的目标可映射成新的网表。

? 开放性和标准化

框架是一种软件平台结构，它为 EDA 工具提供操作环境，框架的关键在于提供与硬件平台无关的图形用户界面以及工具之间的通信、设计数据和设计流程的管理等，此外还应包括各种与数据库相关的服务项目。任何一个 EDA 工具系统只要建立了一个符合标准的开放式框架结构，就可以接纳其他厂商的 EDA 工具一起进行设计工作。框架作为一套使用和配置 EDA 软件包的规范，就可能实现各种 EDA 工具间的优化组合，并集成在一个易于管理的统一的环境之下，实现资源共享。

近年来，硬件描述语言等设计数据格式的逐步标准化，不同设计风格和应用的要求导致各具特色的 EDA 工具被集成在同一个工作站上，从而使 EDA 框架标准化。新的 EDA 系统不仅能够实现高层次的自动逻辑综合、硅编译器和测试码的生成，而且可以各个仿真器对同一个设计进行协同仿真，进一步提高了 EDA 系统的工作效率和设计的正确性。

2. EDA 的基本工具

集成电路技术的不断进展对 EDA 技术提出了新的要求,促进了 EDA 系统的发展。但是总的来说,EDA 系统的设计能力一直难以赶上集成电路技术的要求。EDA 工具的发展经历了两个主要阶段：物理工具和逻辑工具。现在 EDA 和系统设计工具正在逐渐被理解成一个整体的概念：电子系统设计自动化。

物理工具用来完成设计中的实际物理问题，如芯片布局、印刷电路板布线等等。同时，它们也能够提供一些设计的电气性能的分析，如设计规则检查。这些工作现在主要由集成电路厂家来完成。

逻辑工具是基于网表、布尔逻辑、传输时序等概念的，首先由原理图编辑器或硬件描述语言进行设计输入，然后利用 EDA 工具完成综合、仿真、优化等过程，最后生成物理工具可以接受的网表或 VHDL 的结构化描述。

设计工具的分类如图 1.1 所示。下面介绍一下基本的设计工具。

(1) 编辑器：编辑器包括文字编辑器和图形编辑器。在系统设计中文字编辑器用来编辑硬件系统的自然描述语言，在其他层次用来编辑电路的硬件描述语言文本。在门级、

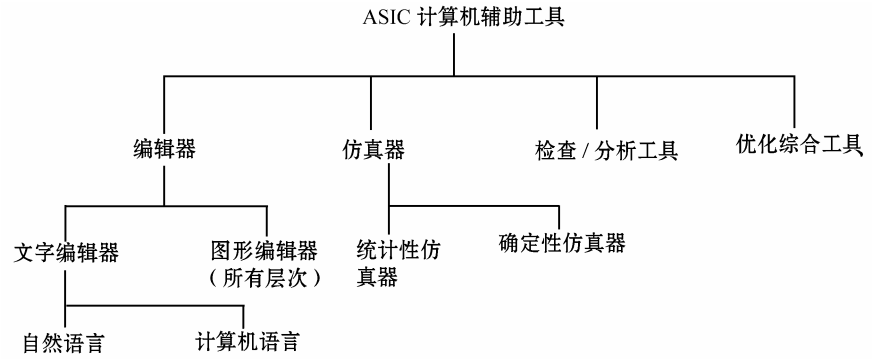

图 1.1　设计工具的分类图

寄存器级以及芯片级，所用描述语言通常为 VHDL 和 Verilog。

图形编辑器可用于硬件设计的各种层次。在版图级，图形编辑器用来编辑表示硅加工过程的几何图形，在高于版图层次的其他级，图形编辑器用来编辑硬件系统的方框图、原理图等。图形化的硬件原理图表示的是硬件的结构描述。

典型的原理图输入工具包括基本单元符号库、原理图形编辑器、网表产生工具。

(2) 仿真器：仿真器又称模拟器，帮助设计者验证设计的正确性。在硬件系统设计的各个层次都要用到仿真器。在数字系统设计时，硬件系统由数字逻辑器件以及它们之间的互连来表示，仿真器的用途是确定系统的输入输出关系，所采用的方法是把每一个数字逻辑器件映射为一个或几个进程，把整个系统映射为由进程互连组成的网络就是设计的仿真模型。

仿真器只是仿真系统的一部分，仿真的过程简单介绍如下。

描述硬件行为的 VHDL 源代码由文本编辑器输入计算机后，先由源代码分析器检查错误。源代码分析器可以检查三类错误：词法错误、句法错误和语义错误。模型经过源代码分析器检查之后，转换为适合于仿真器及其他设计工具使用的中间格式，并存储在设计库中。为了证实用仿真器进行仿真，还需要用户输入的元件模型与设计库中已有的设计单元连接在一起，构成完整的仿真模型。仿真模型建立之后，就可以用仿真器进行仿真了。图 1.1 还给出了设计库与原理图编辑器及与其他设计工具的接口。

(3) 检查/分析工具：在集成电路设计的各个层次都会用到检查/分析工具。在版图级，必须用设计规则检查/分析工具来保证版图所表示的电路可以可靠地制造出来；用于逻辑门级设计的检查/分析工具可以用来检查是否有违反扇出规则的连接关系；时序分析器可以用来检查最坏情形时电路中的最大最小延时。

(4) 优化综合工具：优化综合工具用来把一种硬件描述转化为另一种描述，这里的转化过程通常伴随着设计的某种改进。在逻辑门级，可以用逻辑最小化对布尔表达式进行简化；在寄存器级优化工具可以用来确定控制序列和数据路径的最优组合。各个层次的综合工具可以将硬件的高层次描述转化为低层次描述，也可以将硬件的行为描述转换为结构描述。

1.2 硬件描述语言

1.2.1 VHDL 硬件描述语言

在传统的硬件电路设计中，主要的设计文件是电路原理图，而采用 HDL 设计系统硬件电路时主要使用 HDL 编写源程序。所谓硬件描述语言（HDL），就是该语言可以描述硬件电路的功能，信号连接关系及定时关系。许多公司开发了自己专有的 HDL，包括 Zycad 公司的 ISP，Gateway Design Automation 公司的 Verilog 以及 Mentor Graphics 公司的 BLM。其中，Silicon Compiler 公司的 M 及 Gateway 公司的 Verilog 以 C 语言为基础。UDL/I 在日本以标准 HDL 的形式出现。多年来设计者一直使用这些专用的 HDL。

1982 年，各 ASIC 芯片厂商相继开发了用于各自目的的 HDL。1987 年底，IEEE 确认美国国防部开发的 VHDL 为标准硬件描述语言（IEEE-1076）。之后，各 EDA 公司研制的硬件电路设计工具逐渐向 VHDL 靠拢，VHDL 在电子设计领域得到广泛的接受，1993 年，IEEE 对 VHDL 进行了修订，公布了新版本的 VHDL（即 IEEE-1076-1993）。现在，VHDL 和 Verilog 作为 IEEE 的工业标准硬件描述语言，在电子工程领域，从各公司的设计人员到各大学的教授、学生，都极其重视对其的学习研究，VHDL 已成为事实上的通用硬件描述语言。有专家认为，在 21 世纪中，几乎全部的数字系统设计任务将由 VHDL 与 Verilog 语言承担，VHDL 将是电子工程设计人员的必备知识。

VHDL 和其他语言相比，最大的区别在于设计方法上的差别。VHDL 的主要优点有：

(1) VHDL 支持自顶至下的和基于库的设计方法，而且支持同步电路、异步电路、现场可编程门阵列器件（field programmable gate array,FPGA）以及其他随机电路的设计。VHDL 具有比其他硬件描述语言更强的行为描述能力，基于抽象的行为描述风格避开了具体的器件结构，使设计人员能从逻辑行为上描述和设计大规模电子系统。目前流行的 EDA 工具和 VHDL 综合器大都能实现行为描述到 RTL 描述的转换。

(2) VHDL 语句的行为描述能力和程序结构决定了它具有支持大规模设计的分解和已有设计再利用的功能，它支持系统的数学模型直到门级电路的描述，并且高层次的行为描述与低层次的门级电路描述、结构描述可以混合使用。这些特点符合 IC 设计的市场要求。VHDL 支持系统级描述，这是它优于其他 VHDL 的最重要的特点。例如，Verilog 语言是一种门级电路描述语言，其风格接近于电路原理图，设计者需要搞清楚具体的电路结构的细节，因此工作量通常较大。VHDL 语言却最适合于描述电路的行为，即描述电路的功能，然后由综合器来生成符合要求的电路网络。设计者在熟悉基本单元电路的描述风格，积累一定的设计经验后，就会为用 VHDL 设计同等性能电路的高效率所鼓舞。

(3) VHDL 的硬件描述与具体的工艺技术和硬件结构无关，当门级或门级以上的描述通过仿真检验后，再利用相应的工具将设计映射成不同的工艺，因此电路的设计与工艺的改变是相互独立的。彼此的改变不会产生不良影响，并且 VHDL 硬件描述语言的实现目标器件的选择范围广泛，可使用各系列的 CPLD、FPGA 及各种门阵列器件。

(4) VHDL 具有类属描述语句和子程序调用等功能，对于已完成的设计源程序，可以

通过修改类属参数表和函数的办法来改变设计的规模和结构。VHDL 具有丰富的仿真语句和库函数，使得门电路级的功能仿真、检查成为可能，使设计者对整个工程设计的结构和功能的可行性做出决策。

(5) VHDL 作为一种 IEEE 的工业标准，使 VHDL 的设计成果便于重复利用和交流。这就更进一步推动了 VHDL 语言的推广及完善。另外，由于其语法严格，给阅读和使用带来极大的便利。

尽管 VHDL 具有诸多优点，但是作为一门硬件描述语言，学习 VHDL 还需要注意一些问题：

(1) 了解数字逻辑方面的硬件电路知识，包括目标芯片基本结构方面的知识，编程人员应以一种并行思路去理解和应用 VHDL。电路系统的内部的子系统乃至部分元器件的工作状态和工作方式可以是相互独立、互不相关的，也可以是前后相继、互为因果的。在任意时刻的电路系统中可以有许多相关和不相关的事件同时并行发生。VHDL 对应电路的实际工作方式，以并行和顺序的多种语句方式来描述在某一时刻中所有可能会发生的事件，即它具有描述由相关和不相关的多维时空复合体电路的功能。因此，在学习 VHDL 语言时应该注意以多维并发思路进行电路系统的设计。

(2) VHDL 语言描述行为并不能代表硬件电路的实际行为方式。例如，MUX 等以并行工作为特征的组合电路在 VHDL 中却以顺序语句描述。顺序语句是按时钟节拍运行的，是典型的计算机指令式语句，而实际的电路并非如此运行。VHDL 语言综合要通过行为级－RTL 级－门电路级的转化，而某些 VHDL 语句如 WAIT 语句并不能被逻辑综合器映射成对应的电路。更重要的是，在 VHDL 程序设计中对硬件资源的耗费有明确的估计。一个成功的 VHDL 工程设计，除了达到一定的功能要求外，还必须占用尽可能少的硬件资源。任何规模的目标芯片的资源都是有限的，应该尽量选择恰当的算法和语句，去除不必要的操作，避免无谓的资源浪费。

(3) VHDL 语言作为 IEEE 制定的工业标准硬件描述语言具有很多优点，但它也具有文本的局限性及隐含信息内容，具体表现如下：

① 通用整数等匿名类型或依据不同系统应用环境而定的语言内容信息，在编译处理时，应对这些现象予以定量限制。

② 文件定义的隐式操作，如 read、write、endfile 等。

③ 位串与符串上下文隐式转换。

④ 预定义属性的不明确性。按照文本手册定义，VHDL 语言的预定义属性 low、high 等存在着不明确性问题。为此可以对照国际的测试码人为地修订。

⑤ 接口的匹配。接口的匹配包括参数个数、类型等匹配检查。对在静态编译中不能完全处理的问题，采取部分匹配的方法，而对于全部的匹配因为参数变化种类情况太多，缺少足够信息而采取信息下传，在动态模拟调试时匹配。

⑥ 符合实际的文法约束。除了上面列出的问题外，在 VHDL 语言文本的语义限制中对于一些不合实际电路意义的现实物理约束没能反映出来，对此也做了扩充检查，如死进程的检测等。

⑦ 为了方便动态处理，在静态语义检查时，要考虑动态匹配时所需的信息，使动态提取确立检查变得简单，这方面的典型问题有类型、范围等。

⑧ 编译优化及错误的尽早发现。在静态语义处理中，若存在可计算值的表达式应尽量求取其对应的静态值，这样便于更早地发现存在的问题，同时也为其他工具的使用提供更优的中间结果显示。

综合以上事项，对于 VHDL 的学习，应在实践中总结经验，尽可能地了解软件语句和硬件结构之间的联系，了解软件描述功能背后的硬件工作行为和硬件结构方式及硬件测试、硬件仿真，以实现高质量的 VHDL 设计。VHDL 的设计流程如图 1.2 所示。

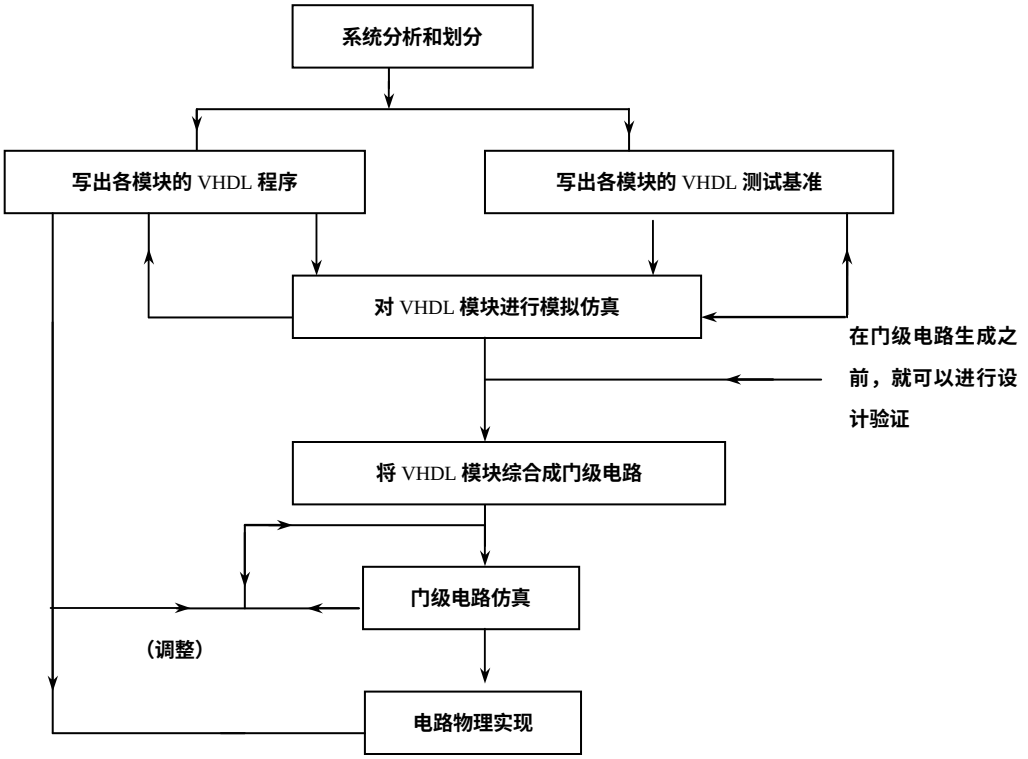

图 1.2　VHDL 设计流程

1.2.2　VHDL 的发展趋势

1. 面向对象的 VHDL 研究

VHDL 的系统级描述缺乏设计概念上的抽象性，面向对象的 VHDL 可提高设计者在较高的设计层次上描述模型的能力，帮助设计者实现更复杂设计、更大规模的元件的重用。VHDL 面向对象的发展是语言本身进步的方向之一。

面向对象的方法在软件开发中已被广泛地接受，它不仅仅是一种新的程序设计技术，而且是一种全新设计和构造软件的思维方法，它是计算机解决问题的方式更加类似于人类的思维方式和更强的管理能力。面向对象的语言必须包含抽象性、可封装性、模块化、层次化及信息机制。抽象性意味着一个对象的特性可以在类描述中文档化。可封装性是指代码和数据必须保存在同一单元中，封装性可有选择性的隐藏信息，使得某些信息对外界不可取。模块化定义了单元的重用。层次化使得对象的行为精炼，不必重复设立前

驱中已有的内容。

2. VITAL 的开发

一般的，VHDL 的设计均自 RTL 开始，模拟验证正确后综合到门级。最后移植到非 VHDL 模拟器上进行处理。这样做是因为 ASIC 开发缺少 VHDL 库。而且存在的 VHDL 库的速度太慢，实用性差。另一方面，在用 VHDL 设计电路时，缺乏统一的、有效的时序处理的描述方法。以前的唯一方法是通过组装实现，但此方法编译太慢，而且 VHDL 库过分依赖模拟器环境。

VITAL（VHDL Initiative Towards ASIC Libraries）的应运而生为面向 ASIC 设计的 VHDL 模型的标准化研究开辟了新思路。1995 年 9 月 VITAL 正式通过成为 IEEE std 1076.4-1995 标准。

VITAL 标准包括 4 个部分：时序程序包，基本元件包，延时机制，模型建立的规范文档。时序程序包和基本元件包与 std_logic_1164 一起放在 IEEE 库中。

VITAL 具有以下特点：

(1) 灵活的功能定义：VITAL 以过程和函数的形式提供元件功能的描述。函数计算元件的行为，过程实现电路的结构描述。

(2) 延时定义精确：延时可以以端到端的方式定义，可以依赖信号状态定义条件。

(3) 具有精确的时序检查功能：可进行建立时间和保持时间的检查，提供最小脉冲宽度、周期检查及事件冲突检查功能。

(4) 两级模型描述规范：VITAL Level 0 描述外部接口，VITAL Level 1 描述内部实现，定义了统一的、形式化的建模风格，利于在工具内部实现和优化，以提高模拟速度。

(5) 以工业标准为基础。

1.3　可编程集成电路与系统设计

现代 VLSI 技术的发展使以 FPGA 为代表的大容量可编程逻辑器件的等效门数迅速增加，要求设计人员保证在复杂程度大大提高的前提下在较短的时间内设计出性能完善的产品。但是，在许多情况下实现一个完整的电子系统的功能仍需要一片或多片专用集成电路与 CPU、存储器、A/D 转换器甚至模拟电路等不同电路协同工作来完成，因此设计可编程电路就要求：一方面要从传统的自底向上的门级设计方法转向更高层设计，来适应可编程集成电路复杂度的提高；另一方面专用集成电路设计要与系统设计紧密结合起来，专用集成电路是整机中一个有机组成部分，它的功能定义来源于系统功能的划分，它的设计完成之后还需要生成仿真模型，提取延时、电路和物理参数，与其他电路一起进行系统级仿真，然后再进入印刷电路板设计等后续流程。

一个典型的设计周期包括：概念设计，采用自然语言描述和方框图相结合的方法；功能分割，在计算机上用方框图描述，而每个功能块的定义采用 VHDL 等硬件描述语言，或直接用原理图输入；功能仿真，验证性能指标和逻辑的正确性；形成各个功能块的结构化原理图并再次验证，这使设计者必须选定目标器件，在原理图中采用厂家提供的单元库，综合过程也必须与厂家的工艺库为目标库，把文本语言的描述映射为各单元互连

关系的网表；把其中的部分或全部的功能块用一片或多片可编程集成电路实现；提取布局布线后集成电路的延迟参数，反标注至网表中，形成包含延时信息的仿真模型，利用时间仿真检查可编程集成电路的速度是否满足要求，时序关系是否正确；采用逻辑仿真、时序分析及数模混合仿真对整个系统设计进行验证；生成可编程集成电路的物理模型，在其中定义可编程 ASIC 器件的各个管脚，反映其与输入输出信号的映射关系；最后进行印刷电路板的设计和分析。