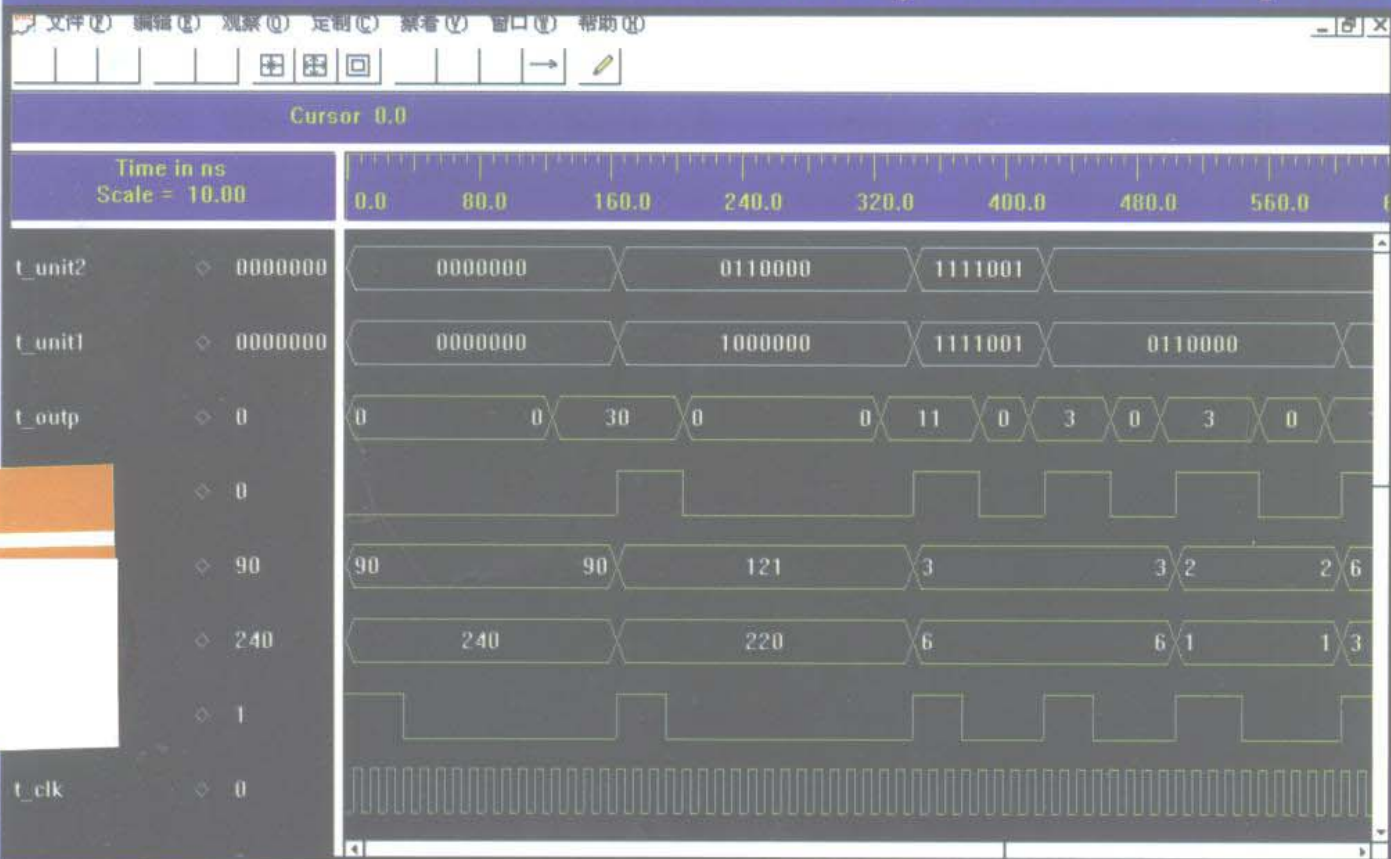


VHDL 语言

100 例详解

北京理工大学ASIC研究所

Detailed Solution of VHDL by 100 Examples



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



VHDL 语言 100 例详解

北京理工大学 ASIC 研究所

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书通过 100 个实例, 详细介绍硬件描述语言 VHDL 的各种语法现象及其在专用集成电路 (ASIC) 设计描述中的使用方法。书后附有光盘, 其中包括北京理工大学 ASIC 研究所自行研制的有自主知识产权的 Talent 高层次自动设计系统的多媒体演示软件和 VHDL 模拟器(学习版)及 100 例的描述与模拟测试向量文件, 读者可直接在微机上运行这些模拟题目, 借以更深入地掌握 VHDL 语言及其使用方法。本书的突出特点是实用性强, 理论联系实际, 是 ASIC 设计者难得的一本 VHDL 语言设计工具书。

本书适合于从事数字系统/ASIC 自动设计的研究、开发人员参考, 也适合于尚未掌握 VHDL 语言但已熟悉高级程序设计语言(如 C 语言或 ADA 语言)的读者学习 VHDL 语言, 也可以作为高等学校计算机、自动控制、信息处理、电子工程和通信等专业的研究生及高年级本科生的教学参考书。

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

书 名: VHDL 语言 100 例详解

作 者: 北京理工大学 ASIC 研究所

出版者: 清华大学出版社(北京清华大学学研楼, 邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 清华大学印刷厂

发行者: 新华书店总店北京发行所

开 本: 787×960 1/16 印张: 32 字数: 698 千字

版 次: 1999 年 12 月第 1 版 1999 年 12 月第 1 次印刷

书 号: ISBN 7-900625-02-X

印 数: 00001~5000

定 价: 52.00 元(含盘)

绪论——专用集成电路 (ASIC) 设计与 VHDL 语言

(代前言)

I. ASIC 设计面临严峻的挑战

人类即将迈进 21 世纪, 半导体器件制造技术高度微细化在迅猛发展。原计划 1998 年实现 0.25 微米技术的生产, 已于 1997 年提前实现; 而预计 2001 年能实现的 0.18 微米技术的生产, 有迹象表明今年(1999 年)即可实现。这使得芯片的设计产生重大变化, 设计制造集成度在 1000 万门以上的 ASIC 芯片成为可能。人们渴望已久的、在单块芯片上实现一个系统的全部功能, 即系统级芯片 (System Level IC, 简称 SLIC; 或 System-on-Chip, 简称 SoC) 的设计与制造正在或已经成为现实。

迈进 21 世纪, 网络与通信技术、多媒体技术以及新型体系结构的计算机设计, 随时都在向 SoC 的设计与制造提出新的严峻挑战。系统级芯片设计是电子信息整机和集成电路产品设计相结合的最完美体现。开展系统级芯片设计工作有利于摆脱电子信息产品设计与集成电路电路芯片设计相脱节的局面, 不但能取得重大经济效益, 而且能够加快我们民族电子产品, 特别是整机产品的发展步伐。

迈进 21 世纪, 制造技术与产品设计的需求都向集成电路设计业提出更严峻的挑战。要求设计行业能更快更好地设计出性能更优良、功能更完美、更复杂的 ASIC 产品。这迫使我们必须研究新的设计策略、设计方法和设计工具。在策略上最重要的是设计重用 (reuse)。集成电路产品的集成度, 目前仍然保持每 18 个月增长一倍的发展速度 (摩尔定律), 而产品的生命周期却日趋缩短, 因此迫切要求提高 ASIC 芯片的设计速度。其中最重要的是尽可能重复运用已有的设计成果, 采用具有知识产权的功能单元块 (称 IP)。因此, 必须重视 IP 的开发和重用。在设计方法方面是要研究在更高的层次上运用设计自动化 (EDA) 工具进行设计的方法。目前我国 ASIC 设计业的基本状况是芯片设计开发工作严重滞后于电子产品发展的需求, 滞后于芯片生产线的吞吐能力。并且设计和投产的 ASIC 产品门类单一, 品种太少, 性能较低。要改变这种状况, 急需提高设计能力。除了加强技术人才培养和设计队伍建设外, 最有效的方法之一是要大力发展高层次 VHDL/Verilog 自动设计技术。

II. 硬件描述语言 VHDL 的出现与发展状况

硬件描述语言 (HDL, Hardware Description Language) 至今约有 40 年的历史, 现已成功地应用于 ASIC 自动设计的模拟验证和综合优化等方面。其最大特点是借鉴高级程

序设计语言的功能特性对电路的行为与结构进行高度抽象化、规范化的形式描述，并对设计进行不同层次、不同领域的模拟验证与综合优化等处理，使设计过程达到高度自动化。至 80 年代末，硬件描述语言的发展趋势进入多领域、多层次并且迫切要求标准化和集成化。最终，只有 VHDL 和 Verilog 适应了这种发展趋势，先后成为 IEEE 制定的硬件描述语言的工业标准。

VHDL 语言的全称是“超高速集成电路硬件描述语言”（VHSIC Hardware Description Language）。VHDL 的结构和方法受到 ADA 语言的影响，并吸收了其他硬件描述语言的某些优点。1986 年 3 月，IEEE 开始致力于 VHDL 的标准化工作，为此，成立了审查和完善 VHDL 的标准化小组。美国空军全力支持这项工作，并与 Intermetrics 签订发展 VHDL（IEEE-1076）的支撑软件合同。1987 年 12 月 IEEE 推出 IEEE Std1076-1987。

VHDL 语言成为 IEEE 的标准后，很快在世界各地得到广泛应用，逐渐成为数字系统/ASIC 设计中的主要硬件描述语言。1995 年中国国家技术监督局组织编撰并出版《CAD 通用技术规范》，推荐 VHDL 语言作为我国电子设计自动化硬件描述语言的国家标准。

为了增强 VHDL 语言的描述能力，方便设计应用，IEEE 在广泛征求各方面意见的基础上，对 IEEE Std 1076-1987 标准进行了修改和扩充。修订版于 1993 年 4 月成为美国国家标准局（ANSI）标准，并于同年 9 月被 IEEE 认可为标准，即 IEEE Std 1076-1993。新版本提供了共享变量（shared Variable）、组（group）、层次化路径名（hierarchical path Name）、异族模块（foreign Model）、签名（signature）等描述机制，增加了一些逻辑和移位操作，修订了 87 版中语法的不一致性。

此外，IEEE 为了促进 VHDL 的应用还成立了专门机构 VASG（VHDL Analysis and Standards Group），下设多个专题组进行有关 VHDL 子标准的建立工作，如 VHDL 综合包标准工作组（1076.3），ASIC 建模标准工作组（1076.4）、综合互操作性工作组（1076.6）等，并已经制订了一系列 VHDL 的子标准，如 VHDL 模型的标准多值逻辑系统 IEEE 1164（Std-Logic 包），VITAL（VHDL Initiative Towards ASIC Libraries）等。这些标准的建立使得不同 EDA 工具间可以通过 VHDL 进行各种设计信息的数据交换，无疑将推动 VHDL 的更广泛的应用。

III. VHDL 语言的特点

VHDL 是一种独立于实现技术的语言，它不受某一特定工艺的束缚，允许设计者在其使用范围内选择工艺和方法。为了适应未来的数字硬件技术，VHDL 还提供了将新技术引入现有设计的潜力。VHDL 语言的最大特点是描述能力极强，覆盖了逻辑设计的诸多领域和层次，并支持众多的硬件模型。具体而言，VHDL 较其他的硬件描述语言有如下优越之处：

1. 支持从系统级到门级电路的描述，同时也支持多层次的混合描述；描述形式可以

是结构描述，也可以是行为描述，或者二者兼而有之。

2. 既支持自底向上 (bottom-up) 的设计，也支持自顶向下 (top-down) 的设计；既支持模块化设计，也支持层次化设计；支持大规模设计的分解和设计重用。

3. 既支持同步电路，也支持异步电路；既支持同步方式，也支持异步方式。

4. 支持传输延迟，也支持惯性延迟，可以更准确地建立复杂的电路硬件模型。

5. 数据类型丰富，既支持预定义的数据类型，又支持自定义的数据类型；VHDL 是强类型语言，设计电路安全性好。

6. 支持过程与函数的概念，有助于设计者组织描述，对行为功能进一步分类。

7. 提供了将独立的工艺集中于一个设计包的方法，便于作为标准的设计文档保存，也便于设计资源的重用。

8. VHDL 语言的类属提供了向设计实体传送环境信息的能力。

9. VHDL 语言的断言语句可用来描述设计本身的约束信息，支持设计直接在描述中书写错误条件和特殊约束，不仅便于模拟调试，而且为综合化简提供了重要信息。

IV. VHDL 语言高级综合

由于 VHDL 是标准的硬件描述语言，因此国际上越来越多的高级综合系统都以 VHDL 作为设计输入。但是 VHDL 语言的本质是基于模拟而非综合的，其丰富的语法成份和描述机制无法且没有必要都进行综合。要实现 VHDL 综合系统，首先需确立 VHDL 的可综合子集。国际上对 VHDL 可综合子集的确立进行了许多研究，取得了一些有意义的结果，但并没有形成统一的标准。为了满足开发综合系统的需要，IEEE 正积极着手 VHDL 可综合子集标准化工作，并推出了征求意见的草案，目前国内可查到的最新版本是 IEEE 1076.6/D2.0。

V. VHDL 语言混级模合拟

1. 硬件结构特性的体现——元件、信号与进程

VHDL 具有许多与数字硬件结构直接相关的概念，其中最主要的是元件，它是数字硬件结构“未知方框”的抽象。VHDL 中，元件由实体与结构体两个概念共同描述，其中实体描述元件与外部环境的接口，其功能及结构是完全隐蔽的。实体的功能定义在称为结构体的单元中，而结构体规定设计实体输入/输出之间的关系。一个实体可存在多个对应的结构体，即可分别以行为、结构、数据流及各种方式混合的描述手段实现。元件的存在使 VHDL 脱离普通程序语言的范畴，成为描述数字电路的专用硬件设计语言。

VHDL 中的信号概念是数字电路中连线的抽象，它是各元件、各进程之间通信的数据通路。VHDL 中的信号的状态可影响与信号相关的进程的运行，体现数字系统各单元的输入及输出的关系。

VHDL 中的进程完成电路行为的描述，由一系列顺序语句组成，是 VHDL 设计中进行功能描述的基本单元。由于进程的执行是并发的，因此在 VHDL 中引入 delta 延迟概念，用于表示时间上无穷小的模拟步，是 VHDL 中模拟进程同步机制的关键。一个模拟时刻包括若干 delta 延迟，所有进程均可能在特定条件下，在同一时刻的任一 delta 延迟点上激活。设计者的设计意图有时希望忽略在 delta 延迟点上的变化，着重于计算一个模拟时刻结束时的稳定状态，因此 VHDL'93 引进延迟进程的概念。此类进程只在某一时刻的最后一个 delta 延迟时激活，这样可降低处理频度，尤其是当用于时序检查方面时。例如对于信号赋值语句

$$S1 \leq A;$$

信号 S1 并不是立即得到所赋的信号 A 的值，而是必须经历 delta 延迟之后，S1 才更新为信号 A 的值。delta 延迟在模拟中由两阶段模拟算法实现。而对于包含以上信号赋值语句的进程，在一个模拟周期内可能频繁激活。有时设计者希望忽略这些延迟激活，因而引进延迟进程概念免除不必要的 delta 延迟处理。

因此，包含上述信号赋值语句的延迟进程将仅在满足激活条件时刻的最后一个 delta 延迟（即一时刻的稳定阶段）激活，激活频度将会大大降低。

延迟进程与非延迟进程的区别在于进程挂起等待之后的唤醒执行的时间不同。进程的激活要素包括三方面：一是敏感信号集，其次是激活条件，再次是等待时间。这些条件相互制约，当激活要素满足时，进程在指定时刻立即激活，所谓指定时刻可细化到某一 delta 延迟时刻。若是延迟进程，则激活推迟到当前模拟时刻的最后一个 delta 延迟时刻（即某一周期的稳定状态），且如果在最后一个 delta 时刻，有多个激活的延迟进程，则这些进程是执行顺序相关的。

2. 传输延迟、惯性延迟与阈值

VHDL'87 标准为信号传输的延迟提供两种延迟模式：传输延迟和惯性延迟。其中传输延迟相应于输入波形没有变化的传输，即任何宽度的脉冲均被传送，无滤除处理，类似电流通过电线上的延迟。而惯性延迟模式，宽度小于惯性延迟的脉冲均被滤除。这种延迟模式体现了开关电路的特性，如果脉冲的宽度小于开关电路的转换时间，或小于指定的脉宽，则不能传播。为了便于明确地定义最小脉宽限制，VHDL'93 引入阈值 (reject) 的概念。

3. 硬件的并发性模拟

VHDL 的并发性体现在两个方面，首先使用 VHDL 进行数字电路设计时存在并发性，即 VHDL 支持设计分解，可使被分解的各子部分的设计并行完成。其次，模型的设计主要由三部分组成：定义实体——确立模型与环境的接口；定义结构体——完成模型的功能描述；定义测试部分——为模型生成测试向量，并捕获模型输出信号状态以供分析。下

面，通过模型的实际设计过程加以说明。首先，在系统分析阶段，系统分析者可将设计对象分为若干独立的子元件，交给若干设计小组实现。系统分析者严格定义元件接口，并将元件之间的相互作用以文档形式提供给各设计小组。然后，各设计小组可独立并行地对子元件进行详细设计，并对子元件进行模拟验证，确保正确性。最终，系统设计者集成各子元件形成完整的设计，对整个设计进行模拟验证。设计的并行性可大大加快整体设计进程和提高设计质量。

其次，VHDL 之所以称为硬件描述语言，很重要的一点是因为它在模拟执行上具有并行性，这点很适于描述电路活动的并行性特点，是其他程序设计语言所不具备的。VHDL 中的进程类似于 UNIX 操作系统中的进程，它们的挂起、活动均是独立的。并行性使得 VHDL 的设计模拟可在并行机上进行，这样大大提高了模拟效率，是解决模拟时间瓶颈的方法之一。

4. 混合级描述及混合级模拟

VHDL 的描述范围覆盖系统级、算法级、寄存器传输级、逻辑电路级，具有连续性和完整性。VHDL 的结构描述方式和行为描述方式有机结合，各描述层次之间彼此衔接，协调一致。目前，较常用的大规模集成电路的设计方法包括基于标准单元库的自底向上的设计方法和自顶向下便于早期优化的 Top-Down 设计方法，以及自底向上和自顶向下相结合的设计方法。由于设计规模日益增大，设计复杂度急剧增加，传统的设计起点偏重低层的方法，会因设计规模的庞大增加很大的工作量。因此提高设计层次，注重早期优化，是现行较好的设计方式。目前，设计对象整体的设计过程经历多个层次。首先，在较高的抽象层次，进行前期的概念设计，优化设计模型；然后经由高级综合工具综合，产生寄存器传输级网表；最后经低层次综合工具，形成最终的设计结果。因此，由于存在多层次设计，就需要多个层次上的模拟，VHDL 模拟器可完成混合级模拟，可为各个层次的硬件设计提供有效模拟，反映设计意图，供设计者调试其设计。是适应当前电路设计的最佳选择之一。

VI. VHDL 语言高级综合系统 Talent

以硬件描述语言 (VHDL/Verilog) 高级综合为核心的高层次设计 (HLD) 方法正日益成为 EDA 的主流。但由于 HLD 跨越设计的多个层次与领域，完成整个流程涉及多种关键技术，如硬件描述语言可视化输入与编译、模拟与验证、综合 (行为级、RTL、逻辑级) 与工艺映射等，因此国际上只有少数几家 EDA 公司掌握了 HLD 的核心方法，所推出的相应的 EDA 工具也都价格昂贵。为了打破国外的技术垄断，推出具有独立知识产权的高层次 EDA 工具，为国内的集成电路设计业服务，北京理工大学 ASIC 研究所于“八五”期间进行了 VHDL 语言高级综合的研究，完成了原型的 VHDL 高级综合及混合级模拟系统 HLS/BIT，并在“九五”期间开展了相应的实用化工作，研制面向实用的专用集成电路高

层次自动设计系统（命名为 Talent）。

Talent 的系统目标是利用硬件描述语言 VHDL 进行数字系统设计的高层次行为功能描述，并通过综合将设计描述自动转换为低层次的设计实现，从而实现设计过程的高度自动化。其特点是基于硬件描述语言，以高级综合为核心，从高层次进行电路的自顶向下设计。其主要功能包括 VHDL 的编辑、编译、模拟验证，设计的自动综合与工艺映射，逻辑图自动生成等。其系统结构如图 0.1 所示，从图中可以看出，Talent 可分为设计输入、设计综合及设计验证三大部分。其设计过程如下。

1. 利用 VHDL 对设计进行功能和算法描述

通过 Talent 系统中 VHDL 智能编辑器可以方便地进行设计描述的录入和编辑，它针对 VHDL 语言的特点特别提供了标识符自动记忆、单词联想及 VHDL 固定语法结构联想式输入等功能。须注意进行设计描述时应根据 Talent 系统所确立的综合子集，使用综合所能接受的语法现象和描述方式。

2. 对 VHDL 设计描述进行编译

Talent 系统的 VHDL 编译器支持 VHDL87/93 全集，以语法分析器为核心，采取语法制导、分别编译（按次序编译）、一次扫描等技术，使系统具有很好的实用性。

3. 通过综合自动生成与工艺无关的 RTL 设计实现

综合又可分为数据流综合与控制流综合两部分，前者自动生成电路的数据通道部分并提取相应的控制信息，后者将所提取的控制信息通过时序逻辑综合及组合逻辑综合完成控制器的综合。其中数据流综合子系统完成高级综合的任务，是整个系统的核心。

4. 通过工艺映射与工艺无关的综合结果转换为与工艺相关的设计结构

综合结果的 RTL 网表与工艺无关，当 ASIC 投片制造时可根据特定的目标工艺，通过工艺映射将综合结果转换成工艺厂商所接收的设计格式。Talent 工艺映射子系统中提出了两级 RTL 映射策略，即 RTL 工艺无关的映射与工艺相关的映射，并采取了知识制导的工艺映射方法。目前通过工艺映射，Talent 系统已实现与几种现场可编程器件开发系统的联结，并生成实际的器件，借以完成器件实现电路的仿真。

5. 在设计各个阶段利用 VHDL 模拟进行设计的模拟验证

Talent 系统的 VHDL 混合级模拟器 (Vsim/Talent) 全面支持 VHDL87 和 VHDL93，并提供了强大的调试功能。其模拟核心采用事件驱动算法，对于同步电路设计采用基于周期的算法。模拟核心采用层次模拟，保留设计原型的元件之间的互连及嵌套关系，便于加载完善、灵活的调试系统，进行调试定位，信息查找和运行控制，符合设计者的思维习惯。

6. 利用逻辑图自动生成工具直观地观察设计结果

逻辑图自动生成工具将综合及工艺映射的结果分页自动生成逻辑图，并作为设计文档保存。其成图迅速，布局美观，走线均匀合理，合乎人的阅读习惯，并具有友好的用户界面及缩放、滚动等完善的编辑功能。

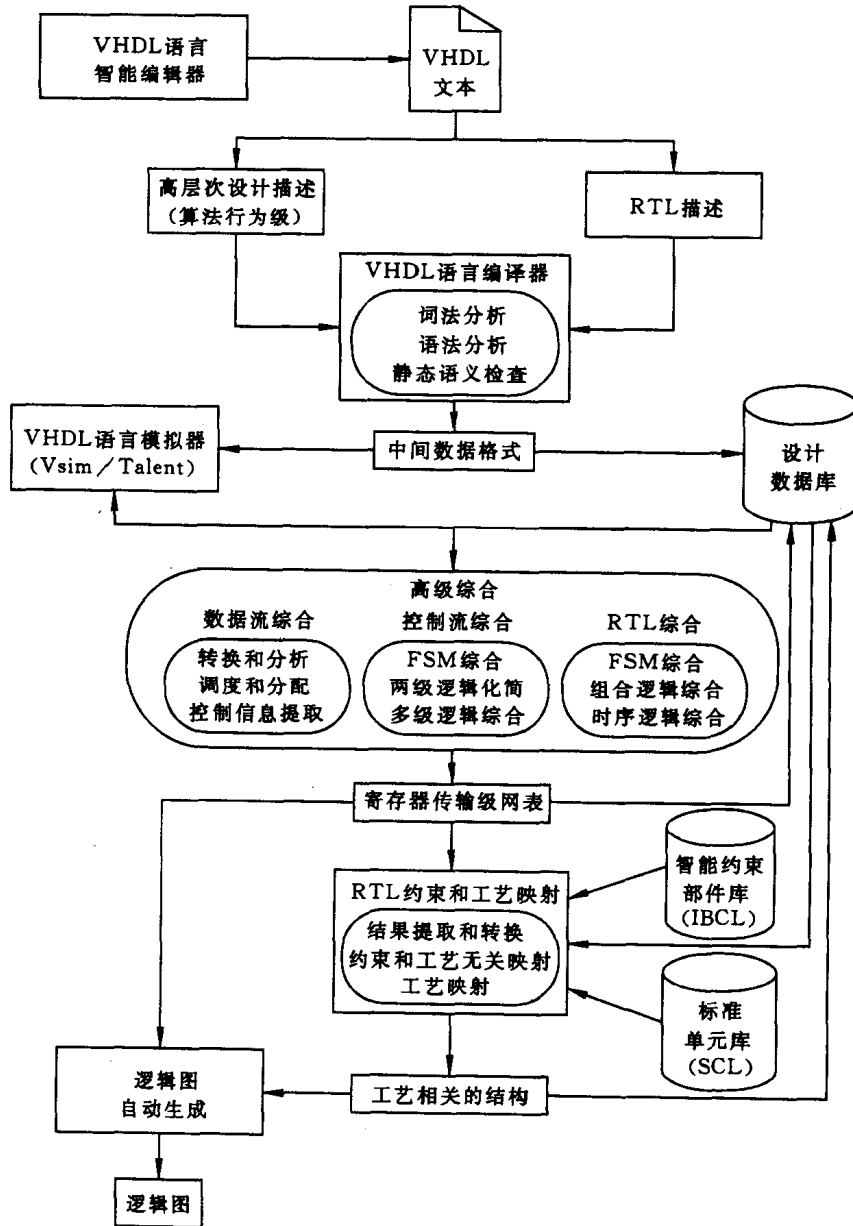


图 0.1 Talent 系统结构图

VII. VHDL 语言 100 例的选取与本书的撰写目标

与其他以 VHDL 为输入的 EDA 软件一样, Talent 自动设计系统与设计者的通信界面(工具)是 VHDL 语言。应用该系统完成 ASIC 设计的至关重要的前提是,设计者必须熟练掌握 VHDL 语言,必须能够运用 VHDL 语言对所设计的 ASIC 完成其行为功能或结构的描述。VHDL 语言规模很大,语法现象十分复杂。经验表明,一个初学者要想达到能得心应手地描述出一个有一定规模的 ASIC 芯片的程度,总需要一个学习和实践的过程。特别是对某些规模较大有实用意义的设计题目的描述,上机模拟和结果分析是一个相当枯燥艰难的历程。本书的撰写恰是为缩短这一进程,为读者提供一个 VHDL 语言学习和借鉴的捷径。

本书选取的 100 个例题全部在 Talent 系统上通过编译和模拟。其中许多题目曾用于调试和测试 Talent 系统。这些题目大致可分为 4 类。

首先选择组成数字系统/ASIC 的某些基本单元,如加(减)法器、乘(除)法器、比较器、选择器、寄存器等(第 1~8 例)。通过对这些单元电路的行为功能描述,阐明 VHDL 语言设计实体说明、结构体描述、并发进程语句与激活条件、顺序语句、变量与信号、位向量、延迟以及重载等 VHDL 语言的基本概念和描述方法。即使没学过硬件描述语言的人,只要有高级程序设计语言(如 C 语言或 ADA 语言)的基本知识,也可借助这些题目的详解掌握 VHDL 语言。

其次选择的一批例题是为了拓宽讨论 VHDL 语言的一些更为重要而复杂的语法现象。特别是对一些较难理解的语法问题和使用技巧,则通过反复举例进行充分解释。例如数据类型、函数及七值逻辑问题(第 9~18 例),死锁与振荡(第 20~23 例),分辨函数、分辨信号与属性(第 24~29 例),进程(第 30~33 例),类属(第 40, 41, 46 例)以及延迟分析(第 48~51 例)等都有重点地列举若干例题进行深入浅出的讨论。

随后开始触及某些单元电路的应用,例如各种功能的寄存/计数器(第 52~56 例),译码器(第 57~59 例),基本计算电路(第 60~63 例)及有限状态自动机(第 64 和 65 例)等等。通过这些举例读者可进一步深入而全面地理解和掌握前述语法现象的使用方法。

最后给出某些有一定实用意义的举例。如 DSP(第 66 和 67 例),整机性的设计举例(第 68~77、90~93 例),四位微处理器芯片 Am2901 和 Am2910(第 78~89 例)以及流水线结构的 RISC 机(第 94~100 例)。这些举例具有典型性和实用性。它们的 VHDL 描述较长,语法结构也相对复杂。读者需熟悉它们的组成原理和体系结构后,再来分析相应的 VHDL 源描述。弄清楚这些描述,非常有利于您用 VHDL 进行 ASIC 设计工作。

附录 I 中的 100 例内容一览表,摘要给出每一例涵盖的基本语法内容,可供读者迅速查寻所需要参考的题目。为了使读者准确掌握与理解 VHDL 语言的用语,附录 II 特地给出 VHDL 语言专用术语的中英文对照表。

书后所附光盘给出 Talent 系统的多媒体演示系统、100 例描述和模拟测试向量的全部有关文件，以及 Talent 系统 VHDL 模拟器（学习版本），读者可在熟读附录 III 的 Vsim/Talent 使用方法后，到微机上直接运行这些模拟文件，借以更深入地掌握有关的内容。

刘明业教授主持本书稿的撰写，并最后统稿全书。参加撰写工作的有石峰副教授/博士、韩曙副教授、张东晓博士、袁媛、陈东瑛、刘沁楠、吴清平、刁岚松、王作建、李春、李杰、谢巍、张俭锋等 13 位同志。叶梅龙教授精心审阅了本书各例的初稿，并进行了修改，使全书撰写格式和措辞用语等趋于统一和规范，同时归纳出书后附录 I 的一览表。袁媛为本书稿的校对、改错和部分录入工作付出了大量艰辛的劳动。

本书的素材是作者们多年从事科研开发工作的积累。他们的工作长期以来得到国家“八五”“九五”科技攻关项目、国防微电子技术预研项目、国家自然科学基金项目以及国家教委博士点建设基金项目的支持。对上述有关部委、国防科工委、电科院、北京华大集成电路设计中心、兵科院及北京理工大学的各级领导、师长和朋友的长期关怀、指导和帮助，作者在此谨致以诚挚的谢意。

限于我们的工作实践和认识水平，书中难免存在缺点、疏忽甚至错误。恳切希望广大读者批评指正。

刘明业

1999 年 9 月 9 日

于北京理工大学 ASIC 研究所

目 录

绪论——专用集成电路 (ASIC) 设计与 VHDL 语言 (代前言)	刘明业 (V)
I ASIC 设计面临严峻的挑战	(V)
II 硬件描述语言 VHDL 的出现与发展状况	(V)
III VHDL 语言的特点	(VI)
IV VHDL 语言高级综合	(VII)
V VHDL 语言混合级模拟	(VII)
VI VHDL 语言高级综合系统 Talent	(IX)
VII VHDL 语言 100 例的选取与本书的撰写目标	(XII)
第 1 例 带控制端口的加法器	袁 媛 (1)
第 2 例 无控制端口的加法器	袁 媛 (4)
第 3 例 乘法器	袁 媛 (6)
第 4 例 比较器	袁 媛 (8)
第 5 例 二路选择器	袁 媛 (11)
第 6 例 寄存器	袁 媛 (13)
第 7 例 移位寄存器	袁 媛 (16)
第 8 例 综合单元库	袁 媛 (22)
第 9 例 七值逻辑与基本数据类型	袁 媛 (29)
第 10 例 函数	袁 媛 (32)
第 11 例 七值逻辑线或分辨函数	袁 媛 (35)
第 12 例 转换函数	袁 媛 (38)
第 13 例 左移函数	袁 媛 (40)
第 14 例 七值逻辑程序包	袁 媛 (42)
第 15 例 四输入多路器	陈东瑛 (51)
第 16 例 目标选择器	吴清平 (57)
第 17 例 奇偶校验器	陈东瑛 (61)
第 18 例 映射单元库及其使用举例	陈东瑛 (69)
第 19 例 循环边界常数化测试	陈东瑛 (75)
第 20 例 保护保留字	袁 媛 (77)
第 21 例 进程死锁	刘沁楠 (79)

第 22 例	振荡与死锁.....	袁 媛 (81)
第 23 例	振荡电路.....	刁岚松 (83)
第 24 例	分辨信号与分辨函数.....	袁 媛 (87)
第 25 例	信号驱动源.....	刘沁楠 (92)
第 26 例	属性 TRANSACTION 和分辨信号.....	陈东瑛 (96)
第 27 例	块保护及属性 EVENT, STABLE.....	陈东瑛 (101)
第 28 例	形式参数属性的测试.....	刘沁楠 (104)
第 29 例	进程和并发语句.....	刁岚松 (107)
第 30 例	信号发送与接收.....	刁岚松 (111)
第 31 例	中断处理优先机制建模.....	吴清平 (113)
第 32 例	过程限定.....	刘沁楠 (116)
第 33 例	整数比较器及其测试.....	刘沁楠 (119)
第 34 例	数据总线的读写.....	刁岚松 (129)
第 35 例	基于总线的数据通道.....	李 春 (134)
第 36 例	基于多路器的数据通道.....	李 杰 (148)
第 37 例	四值逻辑函数.....	袁 媛 (152)
第 38 例	四值逻辑向量按位或运算.....	刁岚松 (156)
第 39 例	生成语句描述规则结构.....	袁 媛 (159)
第 40 例	带类属的译码器描述.....	袁 媛 (164)
第 41 例	带类属的测试平台.....	袁 媛 (169)
第 42 例	行为与结构的混合描述.....	袁 媛 (171)
第 43 例	四位移位寄存器.....	刘沁楠 (174)
第 44 例	寄存/计数器.....	袁 媛 (185)
第 45 例	顺序过程调用.....	陈东瑛 (189)
第 46 例	VHDL 中 generic 缺省值的使用.....	王作建 (191)
第 47 例	无输入元件的模拟.....	王作建 (196)
第 48 例	测试激励向量的编写.....	袁 媛 (201)
第 49 例	delta 延迟例释.....	吴清平 (206)
第 50 例	惯性延迟分析.....	吴清平 (210)
第 51 例	传输延迟驱动优先.....	陈东瑛 (213)
第 52 例	多倍(次)分频器.....	刁岚松 (216)
第 53 例	三位计数器与测试平台.....	刘沁楠 (220)
第 54 例	分秒计数显示器的行为描述.....	陈东瑛 (226)
第 55 例	地址计数器.....	陈东瑛 (234)
第 56 例	指令预读计数器.....	吴清平 (242)

第 57 例	加、减、乘指令的译码和操作	吴清平	(245)
第 58 例	2-4 译码器结构描述	刘沁楠	(248)
第 59 例	2-4 译码器行为描述	吴清平	(255)
第 60 例	转换函数在元件例示中的应用	王作建	(258)
第 61 例	基于同一基类型的两分辨类型的赋值相容问题	王作建	(261)
第 62 例	最大公约数的计算	刁岚松	(266)
第 63 例	最大公约数七段显示器编码	吴清平	(269)
第 64 例	交通灯控制器	吴清平	(272)
第 65 例	空调系统有限状态自动机	刁岚松	(276)
第 66 例	FIR 滤波器	谢 巍	(280)
第 67 例	五阶椭圆滤波器	刘沁楠	(290)
第 68 例	闹钟系统的控制器	张东晓	(302)
第 69 例	闹钟系统的译码器	陈东瑛	(311)
第 70 例	闹钟系统的移位寄存器	陈东瑛	(315)
第 71 例	闹钟系统的闹钟寄存器和时间计数器	陈东瑛	(317)
第 72 例	闹钟系统的显示驱动器	陈东瑛	(322)
第 73 例	闹钟系统的分频器	陈东瑛	(325)
第 74 例	闹钟系统的整体组装	张东晓	(327)
第 75 例	存储器	李 春	(333)
第 76 例	电机转速控制器	张俭锋	(337)
第 77 例	神经元计算机	袁 媛	(343)
第 78 例	Am2901 四位微处理器的 ALU 输入	韩 曙	(347)
第 79 例	Am2901 四位微处理器的 ALU	韩 曙	(353)
第 80 例	Am2901 四位微处理器的 RAM	韩 曙	(359)
第 81 例	Am2901 四位微处理器的寄存器	韩 曙	(363)
第 82 例	Am2901 四位微处理器的输出与移位	韩 曙	(365)
第 83 例	Am2910 四位微程序控制器中的多路选择器	韩 曙	(370)
第 84 例	Am2910 四位微程序控制器中的计数器/寄存器	韩 曙	(374)
第 85 例	Am2910 四位微程序控制器的指令计数器	韩 曙	(379)
第 86 例	Am2910 四位微程序控制器的堆栈	韩 曙	(382)
第 87 例	Am2910 四位微程序控制器的指令译码器	韩 曙	(390)
第 88 例	可控制计数器	韩 曙	(399)
第 89 例	四位超前进位加法器	韩 曙	(406)
第 90 例	实现窗口搜索算法的并行系统 (1) —— 协同处理器	李 杰	(410)
第 91 例	实现窗口搜索算法的并行系统 (2) —— 序列存储器	李 杰	(416)

第 92 例	实现窗口搜索算法的并行系统 (3) —— 字符串存储器	李 春	(419)
第 93 例	实现窗口搜索算法的并行系统 (4) —— 顶层控制器	李 春	(422)
第 94 例	MB86901 流水线行为描述组成框架	石 峰	(428)
第 95 例	MB86901 寄存器文件管理的描述	石 峰	(434)
第 96 例	MB86901 内 ALU 的行为描述	石 峰	(437)
第 97 例	移位指令的行为描述	石 峰	(440)
第 98 例	单周期指令的描述	石 峰	(442)
第 99 例	多周期指令的描述	石 峰	(445)
第 100 例	MB86901 流水线行为模型	石 峰	(458)
参考文献			(467)
附录 I	100 例内容摘要一览表	叶梅龙 袁 媛	(468)
附录 II	VHDL 专用术语中英文对照	刘沁楠	(475)
附录 III	Talent 系统 VHDL 模拟器使用说明	张俭锋	(486)

附光盘：内容包括 100 例有关的 VHDL 描述文件及 Talent 系统 VHDL 模拟器
(张东晓、吴清平、袁媛监制)

第 1 例 带控制端口的加法器

袁 媛

1. 电路系统工作原理

本例针对一个典型的加法器进行 VHDL 语言的描述,比较特殊的是该加法器带有一个控制端口。它用于完成两个位向量的相加,其电路系统示意图如图 1.1 所示。

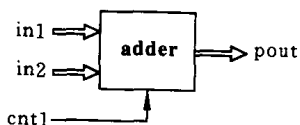


图 1.1 带控制端口的加法器

该加法器有两个输入端口 in1 和 in2,一个输出端口 pout,还有一个控制端口 cnt1,其中输入端口和输出端口的类型为位向量;控制端口的类型则是位,或者为“0”,或者为“1”。当控制端口 cnt1 为“1”时,对输入端口的两个位向量进行加法操作,否则不进行任何操作。

2. 电路的 VHDL 语言描述方法及语法分析

本例的描述与语法是一些基本的结构,十分简单,现详述如下。

(1) 带控制端口加法器的源描述

```
entity  adder  is
    port(   in1  :  bit_vector;
           in2  :  bit_vector;
           cnt1 :  bit;
           pout :  out bit_vector);
end  adder;
architecture  func  of  adder  is
begin
    process(cnt1)
        begin
            if  (cnt1=' 1' )  then  pout <= in1+in2;
            end  if;
        end  process;
    end  func;
```