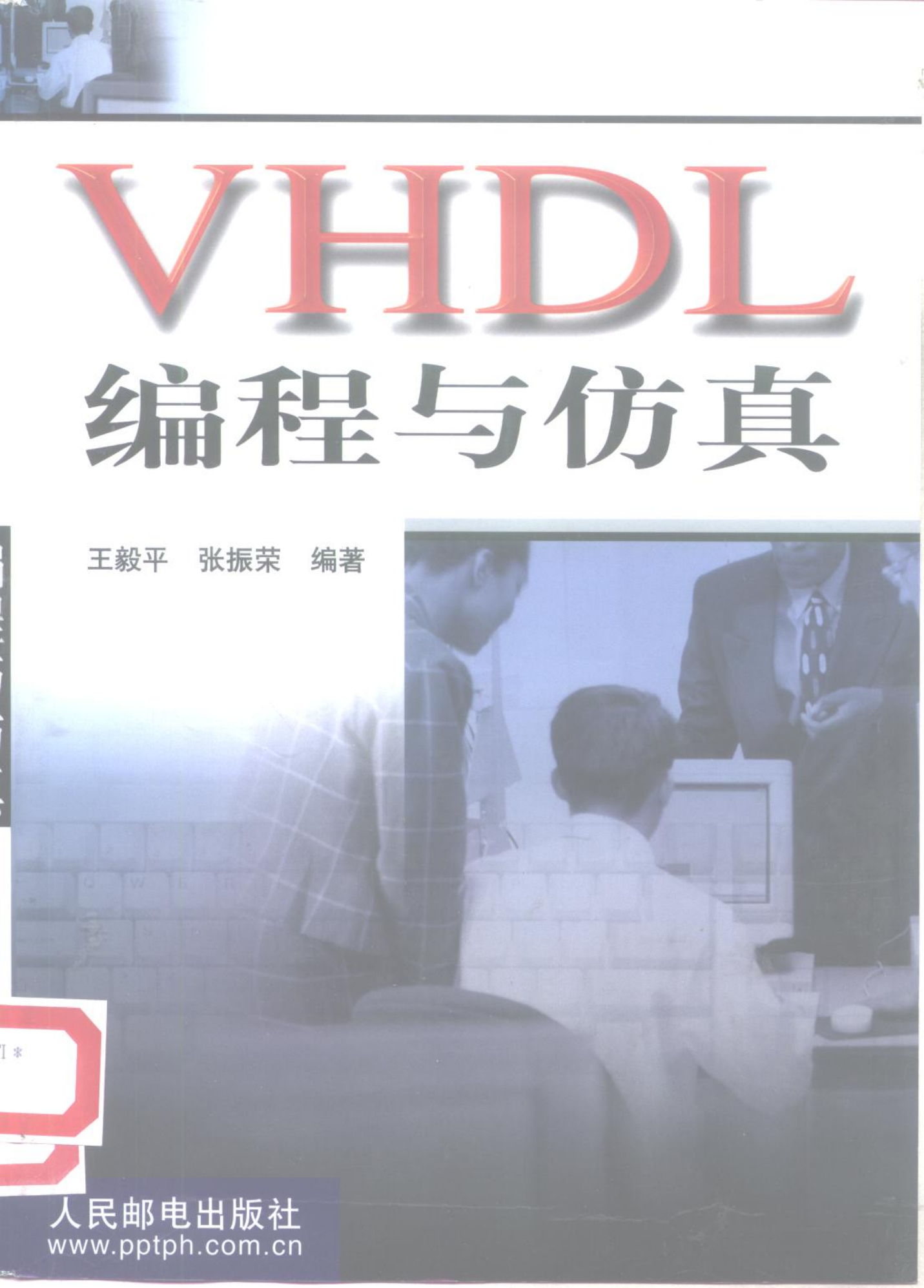
VHDL
编程与仿真

王毅平　张振荣　编著



人民邮电出版社
www.pptph.com.cn

VHDL 编程与仿真

王毅平　张振荣　编著

人 民 邮 电 出 版 社

内 容 提 要

本书主要介绍了 VHDL 语言基础及当前最为流行的基于 Windows 操作系统的 VHDL 仿真软件 Active-VHDL 的使用。主要内容包括：VHDL 的基本概念，Active-VHDL 的安装、启动，VHDL 的基本语法、结构及内部机制，Active-VHDL 的基本操作、仿真、分析及有限状态机的仿真等。本书最后一章还给出了大量的 VHDL 编程实例，其中许多实例可用作编程模板。

本书适用于专业硬件制作人员及电子系统硬件设计爱好者。

VHDL 编程与仿真

◆ 编　　著　王毅平　张振荣
责任编辑　梁　凝
◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@pptph.com.cn
网址　http://www.pptph.com.cn
北京汉魂图文设计有限公司制作
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销
◆ 开本：787 × 1092　1/16
印张：18.5
字数：453 千字　　2000 年 7 月第 1 版
印数：1 - 6 000 册　　2000 年 7 月北京第 1 次印刷
ISBN 7-115-08641-9/TP·1716

定价：30.00 元

前　言

当前，电子系统正在向大规模、集成化和高速度等方向发展，电子设计自动化（EDA）软件工具在市场上已大量出现，以硬件描述语言和逻辑综合为基础的自顶向下建模的电路设计方法在工业界已十分流行。

由于集成电路的设计规模日益增大，复杂程度日益增高，使得门级描述变得难以管理，不得不采用更抽象层次的描述方法，并接受高层次的、自顶向下的设计方法。逻辑图和布尔方程曾经是硬件描述的方法之一，但随着系统复杂程度的增加，这种描述变得过于复杂，不便于使用。在高于逻辑级的抽象层次上，这种方法很难以用简练的方式提供精确的描述，在自顶向下的设计方法中不能再把它当作通常的描述手段，而硬件描述语言（HDL ，Hardware Description Language）则逐渐成为满足以上要求的新方法。HDL 与高层次的软件程序设计语言类似，同时又提供了许多强大的功能，而 VHDL（Very High Speed Integrated Circuit Hardware Description Language）已被 IEEE 接受为标准 HDL，成为硬件设计者广泛接受的一种语言。

本书主要介绍了 VHDL 语言基础及当前最为流行的基于 Windows 操作系统的 VHDL 仿真软件 Active -VHDL 的使用。

本书在结构上主要分为四大部分：第一部分包括第一章和第二章，主要介绍了 VHDL 语言的发展和应用，并介绍了 Active - VHDL 的安装、启动及基本操作；第二部分包括第三章至第十二章，主要介绍 VHDL 基本语法、结构及内部机制；第三部分包括第十三章至第十八章，主要介绍 Active VHDL 中的基本操作、仿真、分析及有限状态机（FSM）的仿真；第四部分为第十九章，该部分给出大量的 VHDL 编程实例，可供编程人员参考，其中许多实例可用作编程模板。

本书适用于专业硬件制作人员及电子系统硬件设计爱好者。

由于时间仓促，书中难免存在不妥甚至错误之处，望广大读者不吝赐教。

编者

2000 年 4 月

目　录

第一章　VHDL 概述

本章主要内容：

- EDA 技术
- VHDL 语言
- Active-VHDL 简介

1.1　EDA 技术

电子设计自动化(EDA)技术从 70 年代开始经历了计算机辅助设计 (CAD)、计算机辅助工程(CAE)、电子系统设计自动化 (ESDA)3 个阶段。前两个阶段的 EDA 产品都只是个别或部分地解决了电子产品设计中的工程问题；第三代 EDA 工具根据工程设计中存在的瓶颈和矛盾对设计数据库实现了统一管理，并提出了并行设计环境的概念，提供了独立于工艺和厂家的系统级的设计工具。

1.1.1　自动控制系统设计领域里存在的普遍问题

目前以多台计算机为基础的集散控制系统已成为计算机控制发展的主流，但是传统的自动控制系统无论是硬件结构还是系统设计方法都存在一些问题。

1. 系统硬件结构存在的问题

在集散系统中，硬件的核心——基本监控单元几乎都是围绕各类单片机，再加上外围大规模集成电路(LSI)和一些通用集成电路(IC)这三大块来设计的。

这部分电路设计的主要问题是设计的通用性和可再利用性差。另外，由于微处理器(MPU)的速度、集成度愈来愈高，寻址能力越来越强，数据总线越来越宽，再加上电子产品高速、低功耗、高可靠性、小型化的发展趋势，导致原来电子系统中的 LSI 和通用 IC 适应不了这一技术发展的要求。

2. 传统的系统设计方法及局限性

传统的系统设计方法，在物理的原理样机生产出来之前，工程师只能在面包板或临时印制板上做部分电路的测试和验证，随着系统复杂程度的提高，在试验板上做整个系统的试验非常困难和费时，因此，传统方法一次设计需经多次修改和试验甚至是大的反复，设计周期长是不言而喻的。

1.1.2　自动控制系统设计的完整解决方案——采用 EDA 技术

采用 EDA 技术后，对自动控制系统设计的改进主要包括以下两个方面：

- 采用可编程逻辑器件后对系统硬件的改造；
- 对系统设计方法的改进。

下面分别介绍这两个方面。

1. 采用可编程逻辑器件后对系统硬件的改造

进入 90 年代以来,可编程逻辑器件以其高速、高集成度和现场可编程的优势，成为电子系统中新的积木块，与 MPU、存储器形成了三个可编程的新的结构体系，并成为电子系统新的设计潮流。

在单片机系统中，可采用一片 FPGA 来实现地址锁存、译码、总线驱动与扩展，甚至实现把 8 位存储器并入 16 位处理器，或把 16 位存储器并入 64 位处理器的控制与字节装配逻辑。

现场可编程逻辑器件在自控系统的大规模高速数据采集与处理系统中，可以代替 DSP 高速、准确地完成一些控制算法。在总线接口控制中，利用现场可编程逻辑器件的在线编程能力，可以使单块通用总线接口板的设计成为可能。在数据通信控制领域内，利用现场可编程逻辑器件，则可实现通信协议的动态转换和总线仲裁逻辑。

2. 对系统设计方法的改进

采用 EDA 技术后的系统设计方法有明显的改进。

EDA 环境下的系统设计一般都采用自顶向下(Top-Down)的系统设计方法，它是针对传统的自底向上(Bottom-Up)的系统设计方法而提出来的。

自底向上的设计方法是从已存在的单元出发进行系统设计，它限制了设计人员的创造力。自顶向下的设计过程可理解为从系统硬件的高层次抽象描述向最底层物理描述的一系列转换过程，直到最后得到可实现的硬件系统描述为止。

对于功率变换系统，由于其复杂性及电路形式的多样性，给直接在电路上进行电路性能的研究、理论分析及计算带来了诸多的不便，特别是在驱动电路、吸收回路、续流电路、滤波电路的参数选取上缺乏准确的指导，因此必须依靠计算机辅助分析的方法来研究功率变换电路的性能和特点。同时，自动控制系统一般工作在强电磁干扰环境下，由此带来了电磁兼容性设计等问题，使设计和调试反复多、周期长，影响了新产品的投入速度。

自动控制系统设计的 EDA 解决方案主要体现在以下 7 个方面：

- 自动控制系统设计的设计输入；
- 自动控制系统中模拟电路的设计；
- 自动控制系统中数字电路的设计；
- 自动控制系统中数模混合电路的设计；
- 自动控制系统中软、硬件的协同设计；
- 自动控制系统的印制板电路板(PCB)设计；
- 系统测试技术。

下面详细介绍这 7 个方面的解决方案。

1. 自动控制系统设计的设计输入

传统设计方法是工程师用纸和笔设计，或通过计算机画出原理图的设计输入，但这对于大系统的设计显然不适用。EDA 技术不仅支持原理图的设计输入，而且还可以支持 VHDL、VERILOG 的输入，甚至是图形化设计输入。如系统功能的流程图、状态转移图、真值表等。因此 EDA 的设计输入是一种概念输入，它突破了具体工艺的束缚，可以充分发挥设计者的创造力。

2. 自动控制系统中模拟电路的设计

工业自动控制系统中模拟电路的设计主要包括两方面内容：一是数据采集与处理部分

模拟信号的调节、整定与隔离；二是功率电子电路的设计。模拟电路设计可分为两方面：模拟IC的设计和模拟系统的设计。

对于模拟系统的设计，各家公司基本上都是以伯克莱分校的SPICE算法为核心，再加上一些分析工具，如容差分析、网络分析、电应力分析等。另外，还有一些功能较强的参数优化工具及丰富的模型库和建模工具。

3. 自动控制系统中数字电路的设计

数字电路设计是自动控制系统中的主要部分。

EDA技术对数字系统的设计是以可编程逻辑器件为基础，以硬件描述语言(HDL)为特色的高层次系统设计方法。近年来，使用硬件描述语言的行为级系统设计技术广受重视。在数字电路硬件描述语言领域内有两种IEEE的工业标准，即VHDL和VERILOG，它们都是技术独立的语言，不束缚于某一特定的工艺，也不把设计方法强加于设计者，大大提高了设计的效率和可再利用性。

在设计输入完成后，接下来做仿真。对于系统仿真，特别是板级仿真，最主要的问题是仿真库的问题，由于板级设计的千差万别，EDA厂商不可能提供一个全面的仿真库来满足所有板级设计者的要求。设计者碰到仿真库不全时，对于复杂的库可以向有关公司购买，对于简单的，可以自己建立一些仿真模型。对于仿真器除关注仿真库的丰富与否外，还必须注意仿真器的速度。据统计，使用EDA进行系统设计，其中有90%以上的时间花在电路仿真上，因此提高仿真效率对缩短产品上市时间是关键一环。

功能仿真正确后，要做逻辑综合。逻辑综合是把一个高层次的和工艺无关的描述转换为一个低层次的与特定工艺相关的逻辑电路的过程。在逻辑综合时要注意的是并非任意的VHDL代码综合出来的结果都能让人满意，不良的综合结果会产生芯片资源的浪费、功率的损耗。而改变这种状况的唯一途径就是改善VHDL代码的风格，使其便于综合，其中最重要的一点是：不要写纯软件性的代码。

在设计验证阶段最重要的是时序分析，并将结果返标回仿真器，然后再用实际延时值做仿真以得到真实结果。时序分析主要解决3方面的问题：

(1) 对芯片、板级、系统级的信号建立和保持时间的检查；

(2) 互连延时仿真和信号完整性分析；

(3) 对布局策略进行评估。

现行的时序分析基本都采用静态时序分析工具，它无需测试矢量，能很清楚地报告VHDL源代码中所存在时序的违反以及一些设计者没有料到的问题，而且它完全支持自顶向下的设计方法。

4. 自动控制系统中数模混合电路的设计

现在的工业自动控制系统中大部分是模数混合电路，甚至是模拟、数字和语言描述的系统组合。在一个系统中同时存在多种设计方法，以往的单点工具显然不能满足这种设计要求，因此，需要一个统一的仿真环境，在这个环境下各单点工具能动态地连接在一起。据此，各家公司先后推出了各种各样的数模混合分析环境，它允许各单点仿真工具和其它分析工具很好地集成在一起，同步协调地运行。

5. 自动控制系统中软、硬件的协同设计

几乎任何一个工业自动控制系统都包含有软件，这就要求在设计硬件的同时也必须兼顾软件调试。以前，软件设计和硬件设计是分开的，这种软、硬件的串行系统设计方式极大

地延长了设计周期。

目前，软、硬件协同设计的嵌入式系统设计是一种最有效的完整解决方法，这种方法采用软件和硬件共同仿真的技术(Co-Simulation)。(例如 Viewlogic 公司 Eagle 分部的 EAGLEI 和 EAGLEV 产品)，它提供了一个虚拟系统集成环境，在此环境下，由虚拟产品控制台 VPC 来同步软件仿真器、硬件仿真器与虚拟软件处理器 VSP(处理器的高性能软件模型)的运作，运用实际的应用程序去驱动硬件设计的软件模型，从而有效地解决了系统中软、硬件设计中的瓶颈，极大缩短了产品的上市时间。

6. 自动控制系统的印制电路板(PCB)设计

以往系统的功能设计和物理设计是两个独立的过程，随着设计复杂性和系统工作频率的提高，这种方式会产生一些严重的后果。因此，就需要一些工具来管理这两者之间的关系，并对一块板上的多种不同性能同时优化，以减少返工。实现这个目标的途径，就是把后端物理设计过程尽可能提到设计周期的前端，这样就能更好地了解本设计的物理因素，如热完整性、时间特性、电磁干扰和可靠性等的影响程度。

7. 系统测试技术

在半导体加工过程或 PCB 制造过程中，可能会引入各种失效，这些物理失效必然表现为电学故障，因此必须对特定用途集成电路(ASIC)或者 PCB 进行严密的测试(FPGA 因为出厂时经过严格测试，因此不须再测试)。

为了对系统进行测试，必须先用计算机辅助测试工具自动产生测试向量，然后进行故障仿真，直到使故障覆盖率达到规定的要求。由于电路规模日益复杂，测试码的生成变得越来越困难。因此，为了对 ASIC 或 PCB 进行测试，在设计时就必须充分考虑到其可测试性，即进行可测试性设计，随着系统设计的复杂性不断提高，为了达到高的故障覆盖率和快速测试就必须借助于良好的测试工具。

1.1.3 FPGA 的设计

FPGA 的设计是 VHDL 语言应用的一个主要方面，本节将介绍如何用高级综合器与 FPGA 开发工具相连接，从而实现从 VHDL 行为级描述到 FPGA 芯片的自动设计。

1. VHDL 高级综合器与 FPGA 工具相连所构成的系统

现场可编程门阵列 (Field Programmable Gate Array，FPGA)的出现使集成电路的设计与实现进入了一个崭新的阶段。用户在现场定制一块集成电路，再用高级综合器与 FPGA 开发工具相连接，就可以实现从超高速集成电路硬件描述语言 (Very High Speed Integrated Circuit Hardware Description Language，VHDL)行为级描述到 FPGA 芯片的自动设计，其所构成的系统如图 1.1 所示。

下面讨论两者的衔接问题：首先分析实现这种连接所面临的问题，然后介绍实现的方法，接着给出若干实例的运行结果 7。

2. 实现高级综合与底层物理设计衔接所面临的问题

将高级综合器与 FPGA 开发工具相连接，主要是将高级综合器综合出的寄存器传输级(Register Transfer Level，RTL)结构转化为 FPGA 工具所能接受的逻辑网表。但实际上，这种 RTL 结构与逻辑网表之间存在着相当的差距，主要有：

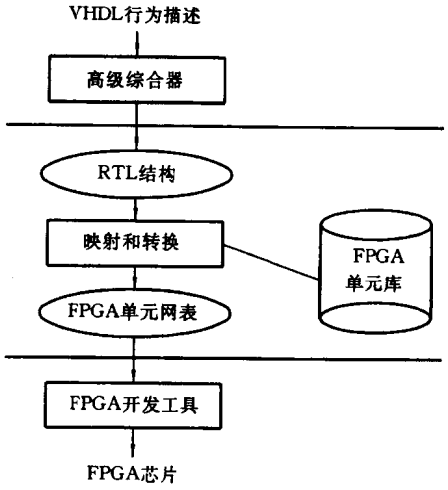


图 1.1 VHDL 高级综合器与 FPGA 工具相连所构成的系统

(1) 元件粒度的差异

通常 FPGA 宏单元的粒度较 RTL 元件小，而 FPGA 基本单元的粒度更小，有些 FPGA 工具只能接受基本单元组成的网表。

(2) 数据宽度差异

高级综合之后的 RTL 元件的数据宽度常常因不同的设计而异，因此需要动态地用 FPGA 库单元构造 RTL 元件，以满足不同的设计要求。

(3) 格式问题

高级综合器和所衔接的后端 CAD 工具在格式上必须一致。

此外，还有一个优化问题，主要是两个方面：一是速度；二是面积。

3. 高级综合与底层物理设计衔接的实现方法

高级综合与底层物理设计的衔接，是经过映射和转换两大步骤实现的。映射即工艺映射。转换的含义：一是将宏单元展开为基本单元；二是进行格式上的转换，使 FPGA 开发工具能够接受。

(1) 工艺映射

一般用一个有向流图 G=(V,A)表示高级综合后的 RTL 结构，其中，V={v1,v2,...,vm}是结点集合，每个结点表示一个抽象部件，Op(vi)为其操作(功能)，A=(a1,a2,...,an)是有向边集合，边 ai=〈vj,vk〉为结点 vj 和 vk 之间的连线(数据依赖关系)。用 L={u1,u2,...,um}表示特定的目标工艺库，用 functionality(ui)表示其中的库单元 ui 能够实现的所有功能集，cost(ui,bw(ui))为造价评估函数，delay(ui,bw(ui))为传输延迟评估函数。其中，bw(ui)表示库单元 ui 的位数。

定义 M 为所有映射 Mi:V→2L 的集合。那么，RTL 工艺映射的目标是寻找满足下列约束条件的映射 Mi：

① $vj \in V, Op(vj) \in functionality(Mi(vj))$

② $\alpha \times \sum cost(Mi(vj),bw(Mi(vj))) + \beta \times \sum delay(Mi(vj),bw(Mi(vj)))$最小

这里，$0 \leqslant \alpha, \beta \leqslant 1$,且 $\beta = 1.0 - \alpha$。

对于任一 $vj \in V$，做映射 Mi 时有两种情况：

① 如果 Mi(vj)={uk},抽象部件 vj 可以用工艺单元 uk 匹配。这时，又存在两种情况：

(a) 若 bw(vj)≤bw(uk),则 vj 可以直接映射为工艺单元 uk；

(b) 若(a-1)×bw(uk)＜bw(vj)≤a×bw(uk),a＞1,则 vj 可以映射为 a 个相同的工艺单元 uk，相应的操作称为单元扩展。

② 如果 Mi(vj)={uk1,uk2，…,ukm},抽象部件 vj 需要用 m 个不同的工艺单元生成，相应的操作称为单元重组。

至于 RTL 工艺映射结果的优化，主要是基于工艺单元库的结构优化。

(2) 转换

转换是借助于宏单元构造库实现的，具体地说，是要解决以下关键问题：

① 宏单元构造库的建立

宏单元构造库记述了宏单元的构造(即一个宏单元是由哪些基本单元构成)及单元间的构造。

② 数据通道和控制器综合结果的组装

通常，高级综合器对数据通道和控制器分别进行综合，形成两个综合结果。组装是将这两部分中对应的连线连接起来，使其成为一个整体。

③ 展开宏单元

利用宏单元构造库，将映射后网表中的宏单元逐个替换为 FPGA 基本单元构成的网表。在展开过程中，应采取相应措施以确保连线的正确，还要对某些元件进行特殊的处理，比如，总线类型的端口、ROM、硬宏单元等。

④ 格式转换

即形成 FPGA 工具能接受的网表格式。

(3) 将各级结果转换为 VHDL 描述

将各级结果转换成 VHDL 描述的好处是可以使系统成为开放的，也就是说可以与第三方厂商的电子设计自动化 (Electronic Design Automation ，EDA)工具进行信息交换，比如，使用 Cadence 等厂家的 VHDL 模拟器进行模拟验证。

(4) 将最终结果转换为 EDIF 描述

电子设计交换格式 (Electronic Design Interchange Format ，EDIF)是一种国际标准，将结果用 EDIF 描述，可使高级综合器与众多的 FPGA 开发工具或其它后端 CAD 工具进行衔接。

4. 若干应用实例

利用本系统可以运行若干 VHDL 描述的实例，具有代表性的有：

(1) 求解某算术表达式；

(2) 高速公路交通灯控制器；

(3) 条形码阅读预处理器；

(4) 五阶椭圆滤波器。

1.2 VHDL 语言

1.2.1 硬件描述语言(HDL)

HDL 与高层次的软件程序设计语言类似，同时它又提供了以下功能：

(1) 在希望的抽象层次上，可以对设计进行精确而简练的描述；