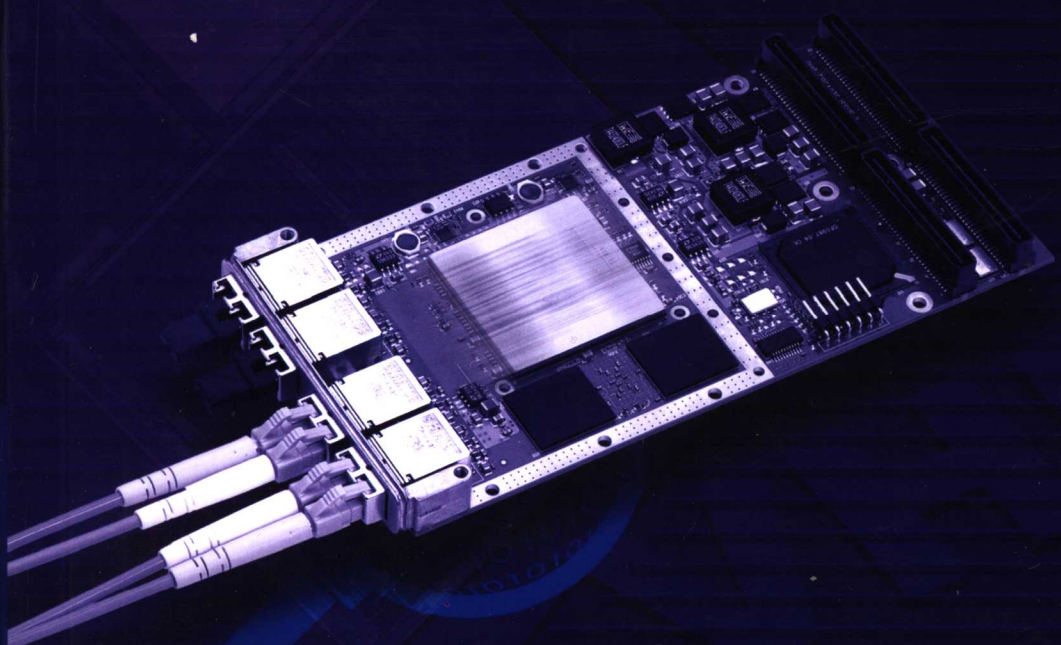
电气信息工程丛书

VHDL数字电路及系统设计

江思敏 编著



机械工业出版社
CHINA MACHINE PRESS

电气信息工程丛书

VHDL 数字电路及系统设计

江思敏　编著

机械工业出版社

本书主要讲述硬件描述语言 VHDL 以及其在数字电路设计中建模和仿真。内容包括 VHDL 基础语法、行为模型的描述、数字电路的 VHDL 建模、VHDL 硬件描述的仿真、综合和仿真工具软件等。全书以丰富的数字电路设计实例贯穿所有的知识点，相信读者可以快速掌握使用 VHDL 进行数字电路的描述以及数字 IC 的设计。

本书主要面向从事数字 IC 设计、FPGA/CPLD 以及 ASIC 设计的工程师和研究人员，非常适合使用 VHDL 进行数字电路设计的设计人员学习参考，也适合高校师生学习参考，是一本全面而实用的 VHDL 数字电路及系统设计的学习教程。

图书在版编目（CIP）数据

VHDL 数字电路及系统设计 / 江思敏编著. —北京：机械工业出版社，2006.8
（电气信息工程丛书）
ISBN 7-111-19693-7

Ⅰ. V… Ⅱ. 江… Ⅲ. 数学电路－电路设计 Ⅳ. TN79

中国版本图书馆 CIP 数据核字（2006）第 088901 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策　　划：胡毓坚
责任编辑：王　颖
责任印制：李　妍

唐山丰电印务有限公司印刷
2006 年 8 月第 1 版 • 第 1 次印刷
184mm×260mm • 20.5 印张 • 504 千字
0001－5000 册
定价：31.00 元

凡购本图书，如有缺页、倒页、脱页，由本社发行部调换
本社购书热线电话（010）68326294
编辑热线电话（010）88379739
封面无防伪标均为盗版

前　言

随着计算机和 EDA 技术的发展，IC 设计已经成为电路设计的一个重要分支。目前，电子产品的性能不断提高，结构越来越复杂，但是价格却在下降，而且产品更新换代的步伐也越来越快，这种技术的进步主要归功于生产制造技术和电子设计技术的发展。芯片的生产制造已经发展到深亚微米阶段，可以在几平方英寸的芯片上集成数千万个晶体管，而电子电路设计的发展则依赖于 EDA 技术的发展。EDA 是指以计算机为工作平台，融合了应用电子技术、计算机技术、智能化技术最新成果而研制成的电子 CAD 通用软件包，主要能辅助进行三方面的设计工作：IC 设计、电子电路设计以及 PCB 设计。基于 FPGA/CPLD 和 ASIC 的数字 IC 设计则是目前发展最为迅速的一个领域，而且广泛应用于计算机、通信和国防等各个领域。我国的 IC 技术的发展和应用也非常迅速，越来越多的技术人员开始从事 IC 和数字电路的设计。

通常，进行数字 IC 设计时，对整个系统进行了方案设计后，就可以使用 EDA 工具软件，采用硬件描述语言（HDL）来完成系统的行为设计，最后就可以进行综合并生成最终的目标元件。目前，最主要的硬件描述语言有 VHDL 和 Verilog HDL，本书主要讲述 VHDL 以及使用 VHDL 进行数字电路设计的方法。VHDL 是一种全方位的硬件描述语言，包括系统行为级、寄存器传输级和逻辑门级多个设计层次，支持结构、数据流、行为三种描述形式的混合描述，因此它几乎覆盖了以往各种硬件描述语言的功能，自顶向下或自底向上的电路设计过程都可以用 VHDL 来完成。使用 VHDL 还可以有效地实现代码的重用，可以有效地提高设计效率。本书详细地讲述了 VHDL 的基础语法、行为模型的描述、数字电路的 VHDL 建模、VHDL 硬件描述的仿真、综合和仿真工具软件等知识。丰富的数字电路设计实例贯穿本书所有的知识点，相信读者可以快速掌握使用 VHDL 进行数字电路的描述以及数字 IC 的设计。本书采用的逻辑符号是国外流行符号，附录列出了与国家标准对照的部分逻辑符号。

本书共分 9 章，第 1 章概括地讲述了 VHDL 的发展、设计方法以及 FPGA/CPLD 的简要介绍；第 2、3 章主要讲述了 VHDL 的程序结构、语法基础和硬件的描述方法；第 4、5 章讲述了如何使用 VHDL 子程序和元件例化进行建模，以及更有效的行为建模方式——有限状态机；第 6 章讲述了 VHDL 程序的仿真以及仿真平台文件的建立；第 7 章讲述了 VHDL 的综合和优化方法以及相应的编程技巧；第 8 章主要介绍了目前最新的综合软件 ISE 7.1 和 Quartus II 5.0，以及仿真平台 ModelSim SE 6.1；第 9 章讲述的是典型的 VHDL 数字电路设计综合实例。全书实例丰富，有助于读者的学习和参考。本书主要面向从事数字 IC 设计、FPGA/CPLD 以及 ASIC 设计的工程师和研究人员，非常适合使用 VHDL 进行数字电路设计的工程师学习参考，也适合高校师生参考使用。

本书在编写过程中参考了许多最新的 VHDL 标准和专著，在此编者深表感谢。由于水平有限，时间仓促，书中缺点和错误在所难免，敬请广大读者批评指正。

编者 Email: qingyuanbook@yahoo.com。

编　者

2006 年 3 月于芝加哥

目　录

第1章 VHDL程序设计概述

VHDL是一种硬件描述语言，主要用来设计数字电路。目前，由于数字IC技术的发展，越来越多的电子工程师选择VHDL作为IC设计工具，使VHDL和Verilog HDL一起成为当前最为流行的硬件描述语言。本章将主要介绍VHDL的基本概念以及典型的数字电路设计硬件平台CPLD/FPGA。

1.1 VHDL概况

1.1.1 VHDL的发展

VHDL是一种硬件描述语言，它是VHSIC Hardware Description Language的缩写，其中VHSIC是Very High Speed Integrated Circuit的缩写。20世纪70年代至80年代，美国国防部为方便管理各种电子电路技术文件，提出了VHDL这种语言规范，以便在各种抽象级描述硬件。使用这种硬件描述语言，可以使电子电路文件遵循统一的设计描述接口，以便实现芯片制造和CAD设计之间的信息交互、电子设计文件的信息共享。逐渐地，VHDL语言成为了一种硬件描述语言的标准。1987年，VHDL被IEEE确定为硬件描述语言标准，并发布为IEEE 1076/A标准。1993年，IEEE又对其进行了修正，并公布了新的VHDL版本，IEEE 1076-1993标准，这个新版本的标准包括了许多新的内容。后来IEEE分别在1996、1999、2000和2002年对VHDL进行了局部的修正。无论如何，目前VHDL语言已经是一种非常流行的硬件描述语言，几乎所有的硬件综合和仿真工具都包括了VHDL语言。特别地，VHDL语言原来完全是为数字系统设计而开发的硬件描述语言，但是目前IEEE也发展了可以描述模拟混合信号的标准IEEE Std 1076.1-1999，不过这不是本书所讲述的范围，本书将主要以IEEE Std 1076-2002修订版本为基础。

1.1.2 VHDL的特点

VHDL主要用于描述数字系统的结构、行为、功能和接口。除了含有许多具有硬件特征的语句外，VHDL的语言格式和描述方法以及语法和一般的计算机高级语言类似。在描述硬件的结构和行为中，VHDL具有如下几个特点：

（1）VHDL语言支持自顶向下（Top-Down）的设计方法，还支持同步电路、异步电路、FPGA以及其他随机电路的设计。

（2）VHDL丰富的仿真语句和库函数，使得在任何大系统的设计早期就能查验设计系统的功能可行性，随时可对设计进行仿真模拟。

（3）VHDL语言具有多层次描述系统硬件功能的能力，可以从系统的数学模型到门级电路，其高层次的行为描述可以与低层次的寄存器传输级（Register Transfer Level，RTL）描述和结构描述混合使用，还可以自定义数据类型，给编程人员带来较大的自由和方便。

（4）VHDL 具有电路仿真与验证功能，可以保证设计的正确性，用户甚至不必编写如何测试相量便可以进行源代码级的调试，而且设计者可以非常方便地比较各种方案的可行性及其优劣，不需做任何实际的电路实验。

（5）VHDL 语句的行为描述能力和程序结构决定了其具有支持大规模设计的分解和已有设计的再利用功能。符合市场需求的大规模系统高效、高速的完成必须有多人甚至多个开发组并行工作才能实现。

（6）对于用 VHDL 完成的一个确定的设计，可以利用 EDA 工具进行逻辑综合和优化，并自动把 VHDL 描述设计转变成门级网表。

1.1.3 VHDL 术语

在详细讲述 VHDL 语法之前，先介绍一下 VHDL 程序设计时常用的一些术语。在编写 VHDL 程序时，在每个程序描述中都有一些基本的 VHDL 程序构造块，如：

（1）实体（Entity）。实体是 VHDL 设计中最其本的组成部分之一，VHDL 表达的所有设计均与实体有关，是设计中最基本的构造块。实体类似于原理图中的一个部件符号，它并不描述设计的具体功能，只是定义所需的全部输入/输出信号。

（2）结构体（Architecture）。所有能被仿真的实体都具有结构体描述，结构体用于描述实体的结构或行为。一个实体可以有多个结构体，每个结构体分别代表该实体功能的不同实现方案。一个结构体可能是行为描述，而另一个结构体可能是设计的结构描述。

（3）配置（Configuration）。一个配置语句是用于绑定一个元件到实体的结构体。一个配置可以看作设计的一个元件表。一个实体可以用多个结构体描述，具体综合时，选择哪一个结构体来综合，由配置来确定，仿真时用配置语句进行配置能节省大量时间。

（4）程序包（Package）。在 VHDL 中，常量、数据类型与子程序可以在实体说明部分和结构体部分加以说明，且实体说明部分所定义的常量、数据类型与子程序在相应的结构体中是可见的（可以被使用的），但在一个实体的说明部分与结构体的部分对于其他实体的说明部分与结构部分是不可见的（实体相当于一个文件），程序包（Package）就是为了使一组常量说明、数据说明、子程序说明和元件说明等内容对于多个设计实体都成为可见的而提供的一种结构，可以这样理解一个实体里的 Package，即对常量等的定义在其实体中是可以被使用的。

（5）属性（Attribute）。属性是 VHDL 实体、结构体、类型及信号的一些特征，比如数据和预定义数据。属性对于综合非常有用。

（6）进程（Process）。进程是最具 VHDL 特色的语句，是 VHDL 中执行语句的基本单元。一个 VHDL 程序的仿真中执行的所有操作都分割为单个或多个进程。

（7）类属（Generic）。一个类属在 VHDL 中用于将信息传递给一个实体的参数。例如，具有上升和下降延迟的门级模型，上升和下降延迟的值就可以使用类属传递给实体。

（8）信号驱动器（Driver）。这是一个信号的驱动源。如果一个信号由两个源驱动，那么当两个源都有效时，则信号具有两个驱动器。

（9）总线（Bus）。总线通常指的是用于硬件设计中的特定通信方式或一组信号。在 VHDL 中，一条总线就是一种特定的信号，通过总线可以关闭其驱动器。

VHDL 作为硬件描述的标准语言，提供了灵活的硬件开发工具。目前，VHDL 的两个主

要应用领域，一个是可编程逻辑器件，包括复杂可编程逻辑器件（Complex Programmable Logic Device，CPLD）和场可编程门阵列（Field Programmable Gate Arrays，FPGA）；另一个应用领域就是专用集成电路（Application Specific Integrated Circuits，ASIC）。一旦写好了 VHDL 代码，它就可以在可编程元件中执行电路，或者实现 ASIC 芯片的制造。目前，许多复杂的商用芯片都是通过这种方式设计制造的。

与普通的计算机编程语言的顺序执行不同，VHDL 语句本质上是并行执行的，只有当 VHDL 在进程或函数中才是顺序执行的。

1.2 VHDL 设计流

VHDL 的主要应用是实现在可编程器件（CPLD 或 FPGA）或 ASIC 上电路或系统的综合。在设计 VHDL 项目时，通常遵循如图 1-1 所示的设计步骤。

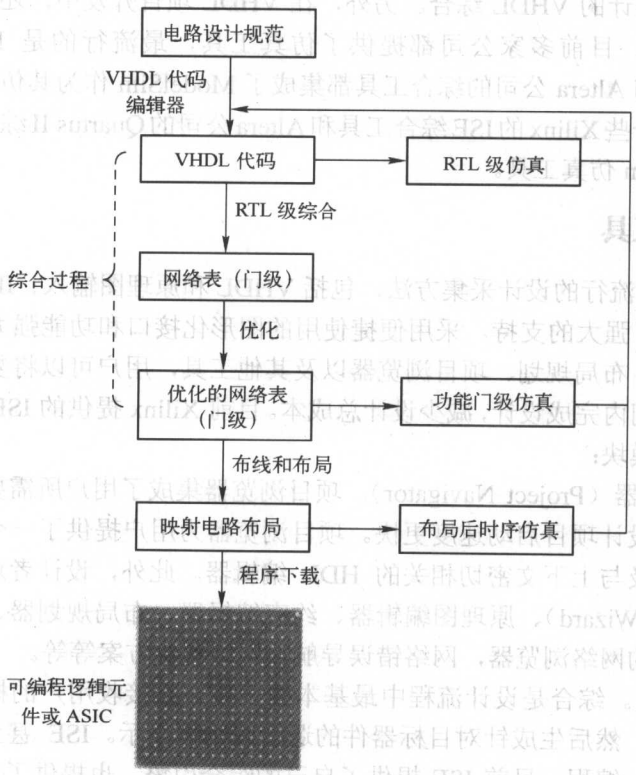


图 1-1　从 VHDL 到 CPLD/FPGA 的自顶向下的设计流

当开始写 VHDL 代码进行设计时，通常保存 VHDL 代码为.vhd 文件，文件名也通常以 ENTITY 的名称来命名。首先需要编写 VHDL 程序，然后可以进行仿真和综合。VHDL 的综合过程包括多个步骤，VHDL 综合过程的第一步就是编译，编译是将在 RTL（Register Transfer Level）级描述电路的高级 VHDL 语言转换为门级网络表。综合的第二步就是优化，用于优化门级网络表，以得到优化的速度。在优化阶段，可以对设计进行仿真。综合的最

后阶段，可以使用布线布局软件在 PLD 或 FPGA 上产生物理层的电路布局，或者在 ASIC 上形成掩膜。

1.3 VHDL 综合工具

目前，有很多电路设计自动化（Electronic Design Automation，EDA）工具提供了 VHDL 支持模块，可以使用 VHDL 进行数字电路的硬件描述和综合。比较流行的 VHDL 综合工具有两种类型，一种是由 FPGA 逻辑器件制造商提供的针对自己的产品而开发的 VHDL 综合工具，比如 Xilinx 的 ISE 主要针对 Xilinx 的 CPLD/FPGA 芯片实现 VHDL 的综合，Altera 的 Quartus II 就是面向 Altera 自己的 CPLD/FPGA 芯片而开发的综合软件。还有一种就是由专门的 EDA 软件开发商开发的通用综合工具，比如 Mentor Graphics 公司的 Leonardo Spectrum 综合工具、Synplicity 公司的 Synplify 综合器、Synopsis 公司的 Design Compiler® FPGA 等都可以用于逻辑器件设计的 VHDL 综合。另外，在 VHDL 项目开发中，还需要对 RTL 级和门级的逻辑进行仿真，目前多家公司都提供了仿真工具，最流行的是 Mentor Graphics 的 ModelSim，Xilinx 和 Altera 公司的综合工具都集成了 ModelSim 作为其仿真平台。

下面简单介绍一些Xilinx的ISE综合工具和Altera公司的Quartus II综合工具，以及Mentor Graphics 的 ModelSim 仿真工具。

1.3.1 ISE 综合工具

ISE 支持当今最流行的设计采集方法，包括 VHDL 和原理图输入、IP 核集成，并为重利用用户的 IP 提供了强大的支持。采用便捷使用的图形化接口和功能强大的问题解决技术，如架构向导、PACE 布局规划、项目浏览器以及其他工具，用户可以将复杂的设计过程进行简化并在更短的时间内完成设计，减少设计总成本。目前 Xilinx 提供的 ISE 版本为 7.1，ISE 7.1 具有如下几个主要模块：

（1）项目浏览器（Project Navigator）。项目浏览器集成了用户所需要的工具，可以在便捷的图形界面下使设计项目启动速度更快。项目浏览器为用户提供了一个窗口，显示项目层次和设计步骤，以及与上下文密切相关的 HDL 编辑器。此外，设计者还可使用直观的结构向导（Architecture Wizard）、原理图编辑器、约束编辑器、布局规划器、语言模板、可展开的设计过程、集成的网络浏览器，网络错误导航，网络解决方案等等。

（2）综合工具。综合是设计流程中最基本的一步。它接收用户的概念性硬件描述语言（HDL）设计定义，然后生成针对目标器件的逻辑或物理表示。ISE 甚至还允许设计采用混合 VHDL 和 Verilog 编程。目前 ISE 提供了自己的综合引擎，也提供了与其他综合工具的集成：

- Mentor Graphics® Leonardo Spectrum™。Xilinx ISE 提供了与 Mentor Graphics 公司的 Leonardo Spectrum 在 ISE 设计环境中无缝的集成。
- XST：Xilinx ISE 提供了它独有的综合引擎，作为第三方合作伙伴引擎的替代产品。

（3）仿真。ISE 提供了与 ModelSim 的无缝接口，可以实现设计过程中的 RTL 级的时序仿真以及门级的信号仿真。

（4）设计实现是一个为用户的设计进行转化、映射、布线、布局和生成 BIT 文件的过

程。利用 Xilinx Fmax 技术，ISE 可用最少的时间提供设计性能最优的解决方案。

（5）验证。ISE 为用户提供全部验证，具有在设计流程的任何点进行几乎所有种类的设计调试的能力，从静态时序分析到支持相同校验的形式验证。

1.3.2 Quartus II 综合工具

Altera® Quartus® II 设计软件提供完整的多平台设计环境，它可以轻易满足特定设计的需要。它是单芯片可编程系统（SOPC）设计的综合性环境。Quartus II 软件拥有 FPGA 和 CPLD 设计的所有阶段的解决方案。Quartus II 具有以下主要模块：

（1）设计输入。Quartus® II 软件中的工程由所有设计文件和与设计有关的设置组成。用户可以使用 Quartus II Block Editor、Text Editor、MegaWizard® Plug-In Manager（Tools 菜单）和 EDA 设计输入工具建立包括 Altera® 宏功能模块、参数化模块库（LPM）函数和知识产权（IP）函数在内的设计。

（2）综合。可以使用 Compiler 的 Quartus® II Analysis & Synthesis 模块分析设计文件和建立工程数据库。Analysis & Synthesis 使用 Quartus II Integrated Synthesis 综合 VHDL 设计文件（.vhd）或 Verilog 设计文件（.v）。用户喜欢的话，可以使用其他 EDA 综合工具综合 VHDL 或 Verilog HDL 设计文件，然后再生成可以与 Quartus II 软件配合使用的 EDIF 网表文件（.edf）或 VQM 文件（.vqm）。

（3）仿真。可以使用 Quartus II 仿真器在工程中仿真任何设计。视所需的信息类型而定，可以进行功能仿真以测试设计的逻辑运算，也可以进行时序仿真以在目标器件中测试设计的逻辑运算和最差时序。Quartus II 软件可以仿真整个设计，或仿真设计的任何部分。可以在工程中将任何设计实体指定为仿真焦点。在仿真设计时，仿真器仿真焦点实体及其所有附属设计实体。当然也可以使用第三方的仿真工具进行仿真。

（4）布局布线。Quartus® II Fitter 也称为 PowerFit™ Fitter，执行布局布线。Fitter 使用由 Analysis & Synthesis 建立的数据库，将工程的逻辑和时序要求与器件的可用资源相匹配。它将每个逻辑功能分配给最好的逻辑单元位置，进行布线和时序，并选择相应的互连路径和引脚分配。

（5）时序分析。Quartus® II Timing Analyzer 允许用户分析设计中所有逻辑的性能，并协助引导 Fitter 满足设计中的时序分析要求。默认情况下，Timing Analyzer 作为全编译的一部分自动运行，它观察和报告时序信息，例如，建立时间（t_{SU}）、保持时间（t_H）、时钟至输出延时（t_{CO}）、引脚至引脚延时（t_{PD}）、最大时钟频率（f_{MAX}）、延缓时间以及设计的其他时序特性。可以使用 Timing Analyzer 生成的信息分析、调试和验证设计的时序性能。还可以使用 Quartus II Timing Analyzer 进行最少的时序分析，它报告最佳情况时序结果，验证驱动芯片外信号的时钟至引脚的延时。

1.3.3 ModelSim 仿真工具

ModelSim 是业界最优秀的 HDL 语言仿真器，它提供最友好的调试环境，是惟一的单内核支持 VHDL 和 Verilog 混合仿真的仿真器。是作 FPGA/ASIC 设计的 RTL 级和门级电路仿真的首选，它采用直接优化的编译技术、Tcl/Tk 技术和单一内核仿真技术，编译仿真速度快，编译的代码与平台无关，便于保护 IP 核，个性化的图形界面和用户接口，为用户加快调错

提供强有力的手段。ModelSim 全面支持 VHDL 和 Verilog 语言的 IEEE 标准，并支持 C/C++ 功能调用和调试。关于 ModelSim 的详细应用请参考专门的手册，不过本书后面也会介绍与实例有关的仿真操作。

1.4 CPLD 和 FPGA 概述

可编程逻辑是大部分数字电路设计工程师实现其设计目标的重要手段，不管逻辑是简单的 I/O 还是复杂的状态机。大部分可编程逻辑是使用硬件描述语言来编写的，并且在可编程的逻辑元件上执行。可编程逻辑元件（PLD）是应用最广泛的用户定制的数字 IC，用户可以根据自己的需要来编写逻辑运算程序，并且将逻辑运算程序写入到可编程逻辑元件就可以实现自己的设计目标。可编程逻辑技术发展非常迅速，最初的逻辑元件有 GAL（Generic Array Logic）和 PAL（Programmable Array Logic）结构的 PLD 元件。在 20 世纪 90 年代，几个逻辑元件商开发出了更为复杂的 PLD，即 CPLD。CPLD 具有更为丰富的逻辑资源和门，因此得到非常广泛的应用。FPGA 作为一种可编程元件，早在 1985 年，Xilinx 公司就研究出了这种结构的逻辑元件，并且逐步得到广泛的应用，特别是在 1990 年代后期到新世纪初，FPGA 发展非常迅速，已经具有几百万甚至上千万的逻辑门，不但可以实现复杂的逻辑运算，而且还可以实现 CPU 和 DSP 的设计，并且在数字电路设计领域等到非常广泛的应用。下面分别介绍一下 GAL、CPLD 和 FPGA 这三种可编程逻辑元件。

1.4.1 GAL 元件

PLD 的最基本类型为通用阵列元件（GAL），GAL 是早期的 PAL 的增强结构。PAL 已经不再使用，但是 PAL 术语还依然广泛使用，常常指的就是 GAL 元件或其他新结构的 PLD 元件。目前 GAL 元件已经在数字系统设计中得到广泛应用。

通过布尔运算，任何逻辑表达式都可以表示为一个复杂的逻辑乘积。因此，通过一个可编程 AND/OR 门阵列，设计的逻辑运算可以实现各种特定的应用。GAL 元件提供了大量可编程的 AND 门（与门），如图 1-2 所示。每个 AND 门可以得到其输入的值或者其输入的反相值。一组 AND 门的输出可以送到 OR 门（或门），并且得到用户所定义的布尔运算表达式。

水平的 AND 门的线和垂直输入的相互连接就是可编程的连接。在 PLD 早期阶段，这些连接是通过熔丝来实现的，并且必须通过高电压才能配置元件。基于熔丝的元件是不可重复编程的。一旦细小的熔丝被熔化，它是不能恢复的。今天的器件通常都是基于 EEPROM 技术和 CMOS 开关来实现非易失的可重复编程特性。无论如何，因为历史的原因，基于熔丝的技术依然还有应用。一个熔丝的默认配置是已经连接的状态，从而将输入连接到 AND 门的输入。当连接熔丝被融化或编程后，AND 的输入和信号输入就被断开，从而将这个连接从布尔表达式中去除了，这种实现方式称为“反熔丝”技术。图 1-3 描述了一个 GAL 可编程与门的编程定制。

因为 AND 阵列是完全可编程的，所以 OR 连接就完全可以是直接硬连接的。每个 GAL 元件将不同数量的 AND 输入送到 OR 门，如果一个或多个 AND 在给定的布尔表达式中不需要的话，就可以将这些不需要的 AND 门强制为 0 而禁止，这可以通过不对 AND 的输入（或不熔化那些连接熔丝）进行编程来实现。

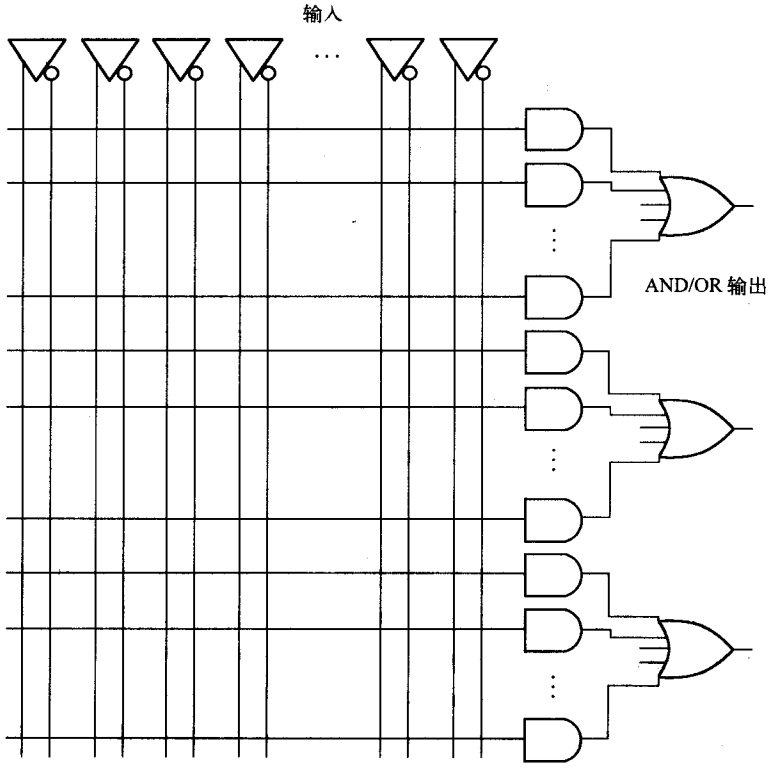


图 1-2　GAL/PAL 元件的 AND/OR（与/或）门结构

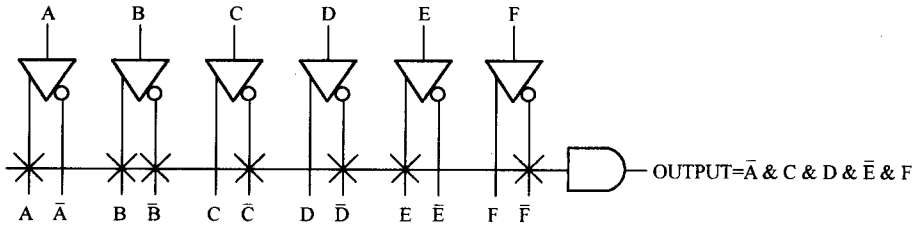


图 1-3　一个 GAL 可编程与门的编程定制

大部分 GAL 元件的逻辑定制是通过对 AND 阵列进行编程来实现的。无论如何，选择一个触发器、OR/NOR 极性和输入/输出配置是通过对一个可配置的 I/O 和反馈结构（称为 Microcell，宏单元）进行编程来实现的。一个宏单元的基本作用就是最终确定 AND/OR 布尔表达式如何进行处理，宏单元的相应 I/O 引脚如何操作。图 1-4 描述了一个 GAL 宏单元的电路表达。当然不同的 GAL 产品，其宏单元也存在一些差别。多路复用器决定了最后的 OR/NOR 终端的极性。配置宏单元的输出使能可以确定引脚如何动作。

按照目前的 PLD 元件的应用来说，GAL 是属于低密度的 PLD 元件，但是其低价格和高速度也是其优势，因此依然在很多场合有应用。同样，GAL 元件的开发也可以使用 VHDL 来实现，其设计流程遵循一般的 PLD 元件的开发过程。