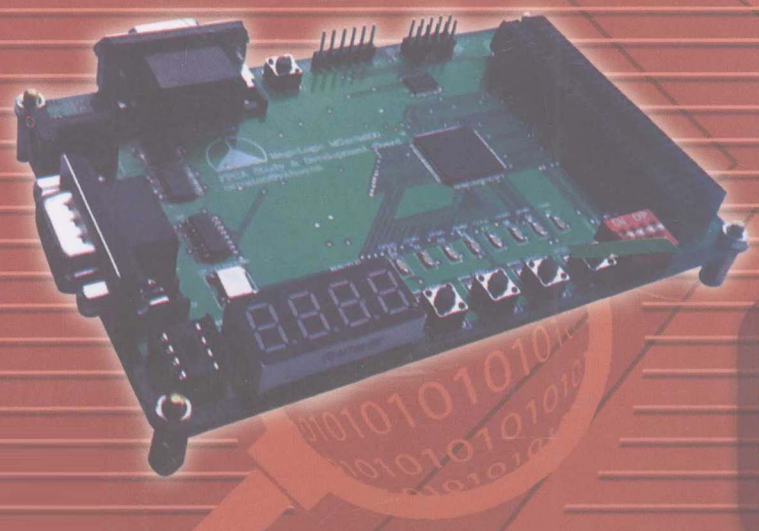
电气信息工程丛书

FPGA/VHDL
设计入门与进阶

杜 勇 编著



- 合理安排章节内容，轻松学习 FPGA 技术
- 详解硬件语言特点，迅速掌握编程技巧
- 深度剖析典型问题，顺利理清开发思路
- 讲解完整实例应用，体会 FPGA 学习乐趣

机械工业出版社
CHINA MACHINE PRESS

电气信息工程丛书

FPGA/VHDL 设计入门与进阶

杜 勇 编著

机械工业出版社

本书是 FPGA 设计的入门级教材，根据初学者的习惯安排章节内容。本书将开发工具与 VHDL 语言紧密结合起来介绍，便于读者尽快形成 VHDL 与 FPGA 设计的整体概念，从而迅速掌握 FPGA 设计技术。

本书主要介绍了 VHDL 语言、ISE 工具、ModelSim 工具、FPGA 设计技巧以及典型 FPGA 硬件电路板设计等相关内容，重点讲解 VHDL 语言与常规软件语言的区别，详细阐述 VHDL 语言设计的思路及方法，力求使读者能顺利弄懂硬件编程语言及 FPGA 设计的特点。

本书适合于 FPGA 设计初学者使用，可作为电子信息类本科高年级学生和研究生的参考教材，也可作为 FPGA 工程师的参考书。

图书在版编目（CIP）数据

FPGA/VHDL 设计入门与进阶/杜勇编著. —北京：机械工业出版社，2010.10
（电气信息工程丛书）
ISBN 978-7-111-32208-5

Ⅰ. ①F… Ⅱ. ①杜… Ⅲ. ①可编程序逻辑器件-基本知识 ②硬件描述语言，VHDL-程序设计 Ⅳ. ①TP332.1 ②TP312

中国版本图书馆 CIP 数据核字（2010）第 198747 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

责任编辑：时 静

责任印制：杨 曦

北京中兴印刷有限公司印刷

2011 年 1 月第 1 版·第 1 次印刷

184mm × 260mm · 16 印张 · 390 千字

0 001—3 500 册

标准书号：ISBN 978-7-111- 32208-5

定价：34.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务
社服务中心：（010）88361066
销 售 一 部：（010）68326294
销 售 二 部：（010）88379649
读者服务部：（010）68993821

网络服务
门户网：http://www. cmpbook. com
教材网：http://www. cmpedu. com
封面无防伪标均为盗版

前　言

为什么要学 FPGA

现代电子信息技术的迅猛发展正在快速地改变着人们的学习、生活及工作方式。随着新技术的不断涌现，新的设计手段正在逐渐取代传统的设计方法。跨入 21 世纪的第 2 个 10 年，如何才能称得上是一名合格的现代电子工程师呢，会不会 51 系列单片机，会不会设计印制电路板，这是绝大部分工科院校电子信息类的毕业生所能想到的具有竞争力的知识和能力储备。不幸的是，虽然以 8051 为代表的单片机仍然在向我们的生活渗透，但它早已无法主导当今电子技术的应用潮流，只写着能熟练进行单片机开发的简历难免淹没在求职信的汪洋大海中。

嵌入式系统已经成为当今电子信息技术设计的发展方向，它不仅涵盖了传统软硬件设计技术的方方面面，同时还融入了多任务实时操作系统等内容。ASIC（Application Specific Integrated Circuit，专用集成电路）、DSP（Digital Signal Processing，数字信号处理）、以 ARM（Advanced RISC Machines）技术为代表的 CPU（Central Processing Unit，中央处理器）以及本书将要讨论的 FPGA（Field Programmable Gate Array，现场可编程门阵列）已成为当今电子信息技术设计的四大基石。可以说，目前几乎所有中高档的电子产品均会用到其中的一项或几项技术。

ASIC 产品性能优良、成本低廉且使用简单，缺点是灵活性不够；DSP 技术擅长于复杂的信号处理及数学运算，缺点是并行运算的能力受限于芯片内部的处理器个数；ARM 技术在需要实时操作系统的场合独领风骚，缺点是门槛高、学习及使用均需要较多的前期知识储备（熟悉 C 语言、操作系统、一定的硬件知识等）；FPGA 技术的巨大优势在于其使用的灵活性及无与伦比的并行运算能力，且学习相对容易，即使只具备初步硬件设计概念的技术人员也能在较短的时间内学会简单的设计，当然，要想达到高手或专家级别，仍然需要付出长期艰苦的努力。

显然，现代电子信息领域的各种设计技术各有优缺点，也就有各自擅长的应用环境。电子信息类产品的功能如此复杂且丰富多彩，如果为了设计出高质量的产品而要掌握各种技术，那是不切实际的，因为人的生命毕竟是有限的，而知识是无限的。其实，各种技术及设计手段一直在不断互相融合、取长补短。比如，DSP 芯片集成了多个处理器以增强其并行处理能力，加入部分可编程逻辑器件的结构从而增加其使用的灵活性；FPGA 器件内部嵌入多个 DSP、ARM 或其他微处理器以弥补其在复杂数学运算或实时操作系统方面的不足。因此，你完全不必为众多的技术而感叹无从下手，精通一项、触类旁通是高手进阶的通常法则。

以个人从业经验来看，如果具有一定硬件基础知识，同时有较好的 C、C++、VC++等软件开发经验，学习 DSP 技术相对容易；如果同时具有良好的软硬件设计基础，那么可以去钻研 ARM 技术，以再次提升自身综合技术实力，向硬件设计专家的行列迈进；如果已具备基本的硬件知识，同时对硬件设计有着浓厚的兴趣，建议你加入 FPGA 技术的学习及应用

队伍中。

怎样学 FPGA

记得上大学时，某位老师习惯在讲完高深理论之后谆谆教诲：上大学最重要的是掌握学习的方法，具备自学的能力，所学的知识还在其次。真是金玉良言！

下面从几个方面向读者介绍怎样学 FPGA。

1．所需预备知识

曾有人在推荐学习 FPGA 设计时说，即使没有基本的硬件概念也可以很快学会使用 HDL（Hardware Description Language，硬件描述语言）进行 FPGA 设计。如果学习的终极目标只是设计简单的诸如点亮 LED 灯或数码管这样的水平，可以这样认为。但是，实际上不是这样简单，FPGA 设计的基本概念来源于数字电路，所以在学习 FPGA 设计之前十分有必要复习一下数字电路基础知识，要了解如门电路、触发器、3/8 译码器等相关内容，这也是必须在学习 FPGA 设计之前了解的知识。另一个常会问到的问题是，对诸如 C 语言等软件编程知识的掌握是否对学习 FPGA 设计有帮助？对此则要依情况而论。任何一个工科大学毕业生应该较为熟悉至少一种计算机编程语言，而 FPGA 设计也推荐用编程语言的方式实现，因此具有一些基本的编程知识是有帮助的。对于一个长期从事软件编程的技术人员来说，在初学 HDL 语言时，则还需要一个适应和转换过程。因为 HDL 虽说是一种语言，但其基本原理还是对硬件进行设计，与 VC++等软件的顺序执行过程、调试方法可以说是大相径庭。我的一位同事是 VC++高手，初学 HDL 一段时间后和我交流体会：开发环境怎么不设计得更好些，单步执行及断点调试等方法应用起来怎么这么麻烦！FPGA 技术经过几十年的发展，各厂商推出的开发环境的功能已十分强大。对于真实的电路板来说，上电后，器件内的所有信号均会开始动作，绝不会如软件一样按程序语句顺序执行，单步执行或断点调试在硬件设计里意义不大。原因十分简单，这种调试方法不能真实反应电路实际工作情况。另外，具备一定专业英语能力对提升设计能力也是十分有好处的。目前流行的 FPGA 开发环境几乎都是英文的，最重要的是，后期深入学习及使用时不可避免会使用 IP（Intellectual Property，知识产权）核进行设计，要使用这些 IP 核，则必须完全读懂其英文说明。

2．选择一本合适的参考教材

有人在学习一门新的编程语言或技术时，可以只看开发环境自带的英文帮助即可掌握，这样的人毕竟是极少数。通常情况下，选择一本合适的参考教材对初学者来说尤其重要，如果选得不好，不但学习效率不高，还可能动摇继续学习的决心。目前市场上各种教材琳琅满目，如何选择还真不是一件容易的事。建议大家：先询问周围从事类似工作的有相当经验的同事或朋友，再花点时间对可能的几本书进行比较，最后确定一本教材作为学习参考资料。有人可能习惯同时购买多本参考教材来进行学习，而实际上参考教材过多不一定是好事，因为需要同时在几本书之间来回调换，这样很容易破坏学习的连续性，从而降低学习效率。至于学习硬件是否必须购买开发试验板的问题，则可以这样理解在经济条件允许的条件下是需要的，毕竟有硬件实物进行实验可以给学习带来更直接的体验，也更易激发学习的热情。由于目前 FPGA 开发环境的功能已十分强大，仿真工具可以十分准确地对实现后的设计电路进

此为试读，需要完整PDF请访问：www.ertongbook.com

行仿真，所以即使在没有实验板的情况下，也完全可以有效地进行 FPGA 设计技术的学习。

3．在查找错误中学习

理论只有与实践结合才能产生巨大的能量。一些教材有配套光盘，建议在学习时尽量按书中的步骤手工输入逐条语句后进行调试，而不是简单地复制光盘内代码。不要担心输入代码过程中出现错误，影响学习进程，在查找问题的过程中学习恰恰是最具效率的。正所谓“错误越多，学习就越快”，当然前提是尽量避免犯相同的错误。

4．注重 HDL 语句与电路实现的相互转换

前面说过，HDL 是针对硬件设计的语言，几乎每条语句都会对应具体的电路结构。在设计时，养成时常用综合工具查看某段语句描述所对应的具体电路的习惯，从而逐渐将具体的 HDL 语句与电路对应起来，将大大提高编写程序的效率，同时可多做读语句画电路图，以及看电路图写语句的转换练习，从而将硬件设计思想深深印在脑海里。与 VC++等软件程序不同的是，HDL 程序的优劣不是看代码的长短，甚至也不是程序的易读性，而是看实现后电路所占的资源及最高运行速度。

本书的内容安排

为了更贴近初学者的学习习惯，本书在章节的内容安排上花费了不少工夫。根据作者的亲身学习体会，并结合作者开展的 FPGA 与 VHDL 技术培训情况，将本书分为 10 章。

第 1 章首先简要介绍了可编程逻辑器件及 HDL 语言的概念，接着用一定篇幅对 FPGA 与 CPLD 的结构及相互之间的异同进行了讲解，因为即使一些具有相当 HDL 编程经验的工程师仍然会把 FPGA 与 CPLD 混为一谈。九层之台起于垒土，合抱之木长于毫末。任何一种技术，专业概念是最基本的，也是成为专家所必须烂熟于心的基本常识。最后介绍了本书所使用的几种设计工具及其安装过程。FPGA 设计工具的安装相对于其他软件来讲显得略为复杂，也许这也可以体现其高度专业性技术的一面吧！

对于初学者来说，安装完工具软件后最急于想做的是什么？当然不是捧着书本逐页阅读工具使用说明或学习 HDL 语法，大家都会急于一试身手，在工具平台上编写或设计一个哪怕只有一行语句的程序。第 2 章简单介绍完 FPGA 设计流程后，即以一个相当经典并常见的功能电路为例，详细讲解了从设计输入到程序下载的一系列 FPGA 设计流程，读者按照书中的步骤即可逐步设计出一个较为复杂的七段数码管显示程序。相信通过这一章的学习，读者可以更深刻地体会到 FPGA 设计的优势，并因此对 FPGA 设计产生更为浓厚的兴趣。

经过第 2 章设计实例的真实体会后，我们还是需要静下心来了解、学习一些基本的相对枯燥而乏味的 VHDL 语法知识。第 3 章介绍的 VHDL 基本语法内容比较简单，学习起来应该困难不大，其中重点对 3 种常用的程序端口进行了说明。程序的端口是程序设计的对外接口，是最基本的内容也是最重要的。由于有第 2 章的基础，读者可以使用 ISE 软件实现本章给出的一些实例。

第 4 章讲解的 VHDL 程序设计的相关内容是学习的重点及难点。如果以前没有学习过 VHDL 语言，当说到 VHDL 最难的语句是赋值语句时，读者一定会以为在开玩笑。事实证明（从自己及同事的学习体会来看），只要深刻理解并掌握了不同情况下赋值语句的含义及所代表的电路结构，也就可以说基本掌握了 VHDL 语言。本章用了 6 种不同的实现方法对

一段程序进行反复讲解，希望能使读者尽快了解并掌握赋值语句。本章最后通过一个秒表功能电路实例对所介绍的 VHDL 语法进行总结，相对于第 2 章的数码管显示电路，这个电路要复杂得多。

第 5 章对 VHDL 的一些高级语法进行了介绍。高级语法就是在一般的工程设计中使用率较低的语法，但了解这些语法十分有助于对 VHDL 语法的深入理解和把握。在介绍这些语法的时候，也配备了相应的应用实例。本章最后还按作者自己的理解，对常用的两种程序建模方法进行了介绍，希望能给编写复杂 VHDL 程序的工程师带来些许启发。VHDL 语法十分完备，本书只针对初学者及工程设计中的语法进行了介绍，还有一些本书没有涉及的语法，有兴趣的读者在学完本书后可以参考专门的 VHDL 语法参考资料。

第 6、7 章全面介绍 ISE 开发工具的使用。因为第 2 章在介绍设计实例的时候，读者实际上已经对 ISE 有了初步的认识。第 6 章侧重于对各项工具全面的介绍，第 7 章则专门对时序约束、功耗分析器、在线逻辑分析仪工具进行了讲解，这部分内容只有较为复杂的设计中才会涉及，充分掌握并熟练应用这几项工具也是专业 FPGA 工程师必须具备的技能。

程序仿真是工程设计中必不可少的一个环节。一般说来，工程师花在设计仿真调试上的时间会占到整个项目设计时间的三分之一以上。ModelSim 仿真软件因其界面友好、功能强大等优势在业界得到了十分广泛的应用，ISE 软件为 ModelSim 预留了软件接口，可十分方便地集成在 ISE 环境中使用。第 8 章在介绍 ModelSimSE 5.8b 软件的使用时，重点讲解了常用的测试及查看方法。最后还专门对文件 IO 进行了讲解，当测试输入数据比较复杂，依靠编辑波形或代码无法形成所需测试数据时，可将测试激励数据存放在文本文件中，在测试激励文件中将数据从外部文本文件中读出，以此形成所需的激励数据。

第 9 章对作者从事 FPGA 设计工作中的一些经验进行了总结，主要介绍了芯片引脚状态设置、DCM（数字时钟管理）模块使用、全局时钟资源使用以及提高系统工作频率等内容，以浮点乘法器为例重点讲解了如何根据芯片结构最大化利用器件资源，提高设计性能的方法。

严格来讲，电路图及 PCB 的设计不属于本书讨论的范围，但据作者个人的从业经验，对于一名 FPGA 设计工程师来讲，不但需要精通 FPGA 的软件设计，了解一些简单的电路板及常用芯片知识，无疑会极大地丰富自己的知识面。第 10 章介绍了一个最简单的 FPGA 电路板设计实例，并对其中有关键电源芯片及 FPGA 芯片进行了介绍。如果读者只是初学的话，同时又喜欢做一些实际的动手实验，完全可以将本章所介绍的电路绘制成 PCB，并投入生产，而后将本书所涉及的设计实例在电路板上进行实验。

本书的目标

本书作为一本 FPGA 设计的入门级教材，内容力求精炼，不追求面面俱到，尤其对 VHDL 语言与常规软件语言的区别，以及硬件设计的思路及方法进行了详细阐述，力求使读者能顺利弄懂硬件编程语言的特点。通过学习本书，可以较快地掌握 FPGA 设计的基本方法和技巧，同时使读者在编程时掌握硬件设计的思路方法，而不是把 VHDL 简单地当做另外一种软件编程语言对待；可以熟练掌握使用开发工具进行程序编辑、综合、仿真、调试及下载实现，具备 FPGA 设计工程师的基本技能。书中有作者大量的设计技巧及经验总结，相信

对读者会有一定的帮助。

实际工作中，发现不少同事在学习 FPGA 设计时，总感到入门较难，难以把握及理解 HDL 语言的特点，学习很长一段时间后，还是会按 VC++的编程思路写出 VHDL 代码。在写作本书的过程中，作者结合自己的学习经验以及从事 FPGA 设计技术培训过程中的体会，不惜用大量笔墨重点对 VHDL 语言中不易理解的概念进行讲解，在工具介绍的过程中一改对菜单及工具简单罗列功能的介绍方式，而是尽量通过具体的实例进行演示，相信读者跟着书中的思路及实例进行学习，可以比较容易掌握 FPGA 与 VHDL 设计的特点及精髓，为进一步深入学习打下良好的基础。

FPGA 技术的发展可以说是日新月异，新技术新工具不断涌现，尤其近年来随着 FPGA 技术、DSP 技术、ARM 技术、ASIC 技术的不断进步，同时各种技术不断相互借鉴与融合，也催生出了一系列新的设计工具及方法。本书讨论的内容只涉及了 FPGA 设计技术中最基本的一些工具，关于系统设计软件System Generator for DSP、嵌入式开发平台软件 Platform Studio&EDK 等相关内容本书均未涉及，如果读者有志于从事 FPGA 设计相关领域的研究或工程应用工作，可以在掌握本书所介绍的基本 FPGA 设计知识及技能的基础上继续深入学习系统设计、嵌入式设计相关知识。

作者在编写该书的过程中查阅了大量的资料，在此对资料的作者及提供者表示感谢。最后还要感谢妻子刘帝英女士，她不仅忍受了作者因为编写本书而长期加班工作，同时还对全部书稿进行了详尽而细致的校对。由于作者水平有限，不足之处敬请读者批评指正。欢迎读者与作者交流 FPGA 设计体会，交流请发电子邮件至：duyongcn@yahoo.com.cn。

如读者需要 FPGA 学习电路板，也可直接与作者联系。

编　者

目　录

此为试读，需要完整PDF请访问：www.ertongbook.com

第 1 章　可编程逻辑器件基础

- PLD 概述
- CPLD 与 FPGA 的区别
- Xilinx 主要器件
- 设计工具及开发环境安装
- 小结

1.1 PLD 概述

1.1.1 基本概念及发展历史

1. 基本概念

随着数字集成电路的发展，越来越多的模拟电路逐渐由数字电路取代，同时数字集成电路本身也在不断地进行更新换代。它由早期的电子管、晶体管、中小规模集成电路发展到超大规模集成电路（Very Large-Scale Integrated Circuit，VLSIC）以及许多具有特定功能的专用集成电路。但是，随着微电子技术的发展，设计与制造集成电路的任务已不完全由半导体厂商来独立承担。电子工程设计师们更愿意自己设计专用集成电路芯片，而且希望 ASIC 的设计周期尽可能短，最好是在实验室里就能设计出合适的 ASIC 芯片，并且立即投入实际应用之中，因而出现了可编程逻辑器件（Programmable Logic Device，PLD），其中应用最广泛的即为现场可编程门阵列（Field Programmable Gate Array，FPGA）和复杂可编程逻辑器件（Complex Programmable Logic Device，CPLD）。PLD 的主要特点是芯片或器件的功能完全由用户通过特定软件编程控制，并完成相应功能，且可反复擦写。这样，用户在用 PLD 设计好 PCB（Print Circuit Board，印制电路板）后，只要预先安排好 PLD 引脚的硬件连接，即可只通过软件编程的方式灵活改变芯片功能，从而达到改变整块 PCB 功能的目的。这种方法不需对 PCB 进行任何更改，从而大大缩短产品的开发周期和成本。也就是说，由于使用了 PLD 进行设计，硬件设计已部分实现了软件化。随着生产工艺的不断革新，高密度、超大规模 FPGA/CPLD 器件越来越多的在电子信息类产品设计中得到应用，同时由于 DSP、ARM 与 FPGA 技术相互融合，在数字信号处理等领域，已出现了具有较强通用性的硬件平台，核心硬件设计工作正逐渐演变为软件设计。

2. 发展历史

早期的可编程逻辑器件在 20 世纪 70 年代初出现，这一时期只有可编程只读存储器（Programmable Read-only Memory，PROM）、可擦除可编程只读存储器（Erasable PROM，EPROM）和电可擦除只读存储器（Electrically EPROM，EEPROM）这 3 种。这类器件结构相对简单，只能完成简单的数字逻辑功能，但也足以给数字电路设计带来巨大变化。

20 世纪 70 年代中期出现了结构上稍复杂的可编程芯片，即可编程逻辑器件，它能够完成各种数字逻辑功能。典型的 PLD 由一个“与”门和一个“或”门阵列组成。由于任意一个组合逻辑都可以用“与 或”表达式来描述，所以 PLD 能以“乘积项”的形式完成大量的组合逻辑功能。这一阶段的产品主要有 PAL（Programmable Array Logic，可编程阵列逻辑）和 GAL（Generic Array Logic，通用阵列逻辑）。PAL 由一个可编程的“与”平面和一个固定的“或”平面构成。PAL 器件是现场可编程的，它的实现工艺有反熔丝技术、EPROM 技术和 EEPROM 技术。还有一类结构更为灵活的逻辑器件是 PLA（Programmable Logic Array，可编程逻辑阵列），它也由一个“与”平面和一个“或”平面构成，但是这两个平面的连接关系是可编程的。PLA 器件既有现场可编程的，也有掩膜可编程的。在 PAL 的基础上又发展了一种通用阵列逻辑，如 GAL16V8、GAL22V10 等。它采用了 EEPROM 工艺，实现了电可擦除、电可改写，其输出结构是可编程的逻辑宏单元，因而它的设计具有很强的灵活

性，至今仍有许多人使用。这些早期 PLD 器件的一个共同特点是，可以实现速度特性较好的逻辑功能，但其过于简单的结构也使它们只能实现规模较小的电路。

为了弥补这一缺陷，20 世纪 80 年代中期，Altera 和 Xilinx 两家公司分别推出了类似于 PAL 结构的扩展型 CPLD 和与标准门阵列类似的 FPGA，它们都具有体系结构和逻辑单元灵活、集成度高以及适用范围宽等特点。这两种器件兼容了 PLD 和 GAL 的优点，可实现较大规模的电路，编程也很灵活。与门阵列等其他 ASIC 相比，它们又具有设计开发周期短、设计制造成本低、开发工具先进、标准产品无需测试、质量稳定以及可实时在线检验等优点，因此被广泛应用于产品的原型设计和产品生产之中。几乎所有应用门阵列、PLD 和中小规模通用数字集成电路的场合均可应用FPGA和CPLD器件。

20 世纪 90 年代末以来，随着可编程逻辑器件工艺和开发工具日新月异的发展，尤其是 Xilinx 公司和 Altera 公司不断推出新一代超大规模可编程逻辑器件，FPGA 技术与 ASIC、DSP 及 CPU 技术不断融合，FPGA 器件中已成功以硬核的形式嵌入 ASIC、PowerPC 处理器、ARM 处理器，以 HDL 的形式嵌入越来越多的标准数字单元，如 PCI 控制器、以太网控制器、MicroBlaze 处理器、Nios 以及 Nios II 处理器等。新技术的发展不仅实现了软硬件设计的完美结合，也实现了灵活性与速度设计的完美结合，使得可编程逻辑器件超越了传统意义上的 FPGA 概念，其应用领域扩展到了系统级，并以此发展形成了现在流行的 SOC（System on Chip，系统级芯片）及 SOPC（System on a Programmable Chip，可编程片上系统）设计技术。

1.1.2　HDL 语言

PLD 器件出现后，需要有一种设计切入点（Design Entry）将设计者的意图表现出来，并最终在具体器件上实现。早期主要有两种设计方式：一种是采取画原理图的方式，就像 PLD 出现之前将分散的 TTL（Transistor-Transistor Logic）芯片组合成电路板一样进行设计，这种方式只是将电路板变成了一块芯片而已；还有一种设计方式是用逻辑方程式来表现设计者意图，将由多条方程式语句组成的文件经过编译器编译后产生相应文件，再由专用工具写到可编程逻辑器件中，从而实现各种逻辑功能。

随着 PLD 器件技术的发展，开发工具的功能已十分强大。目前设计输入方式在形式上虽然仍有原理图输入方式、状态机输入方式和 HDL 输入方式，但由于 HDL 输入方式具有其他方式无法实现的系列优点，其他输入方式已很少使用。HDL 设计输入方式，即采用编程语言进行设计输入的方式主要有以下优点：

1）HDL 没有固定的目标器件，即在做设计时基本上不需要考虑器件的具体结构。由于不同厂商生产的 PLD 器件虽然功能相似，但器件在内部结构上毕竟有不同之处，如采用原理图输入方式，则需要对具体器件的结构、功能部件有一定的了解，从而增加了设计的难度。

2）HDL 设计通用性、兼容性好，十分便于移植。用 HDL 进行设计时，在大多数情况下几乎不需要做任何修改就可以在各种设计环境、PLD 器件之间编译实现，这给项目的升级开发、程序复用、程序交流、程序维护带来很大的便利。

3）由于 HDL 设计之时不需考虑硬件结构，不需考虑布局布线等问题，只需结合仿真软件对设计结果进行仿真即可得到满意的结果，因此可以大大降低设计的复杂度和难度，同时也有利于初学者学习使用。

目前的 HDL 语言较多，主要有 VHDL（VHSIC Hardware Description Language，其中的 VHSIC 是 Very High Speed Integrated Circuit 的缩写，超高速集成电路硬件描述语言）、Verilog HDL、AHDL、SystemC、HandelC、System Verilog、System VHDL 等。其中主流工具语言为 VHDL 和 Verilog HDL，其他 HDL 仍在发展阶段、本身不够成熟，或者是公司专为自己产品开发的工具的应用面不够广泛。

VHDL 和 Verilog HDL 各具优势，在国内的应用平分秋色。VHDL 语法严谨，而正因为其严谨，使得描述具体设计时感觉较为繁琐；Verilog HDL 语法宽松，而因其宽松导致描述具体设计时更容易产生问题，且对于同一个设计，在应用不同 EDA（Electronic Design Automation，电子设计自动化）工具实现时，可能出现不同的实现结果，给程序交流、复用带来麻烦。Verilog HDL 的语言风格与 C 语言很相似，有 C 语言基础的读者在学习时会感到比较亲切。因 VHDL 语言的严谨风格，初学者学习时可以避免不必要的理解上的苦恼。至于初学者在两种语言的选择问题，花过多精力讨论本身意义不大。作者的建议是：依据周围的环境而定，周围同事用什么语言设计，就选择什么语言，这样主要便于学习与交流。

1.2 CPLD 与 FPGA 的区别

目前所说的 PLD 器件，通常情况下指的是 FPGA 与 CPLD 器件。作者发现，即使一些对 HDL 设计已经相当熟练的工程师有时也对 FPGA 与 CPLD 的概念十分模糊。实际上，FPGA 与 CPLD 器件因其内部结构不同，导致其集成度、运算速度、功耗及应用方面均有一定差别。因此，弄清两种 PLD 器件之间的差异，对具体设计时的芯片选型有重要意义。

1.2.1 CPLD 的结构

通常将以乘积项结构方式构成逻辑行为的器件称为 CPLD，如 Xilinx 的 XC9500 系列、Altera 的 MAX7000S 系列和 Lattice 的 Mach 系列等，这类器件的逻辑门密度在几千到几万个逻辑单元之间。

CPLD 是属于粗粒结构的可编程逻辑器件。它具有丰富的逻辑资源（即逻辑门与寄存器的比例高）和高度灵活的路由资源。CPLD 的路由是连接在一起的，而 FPGA 的路由是分割开的。FPGA 可能更灵活，但包括很多跳线，因此速度较 CPLD 慢。CPLD 以群阵列（Array of Clusters）的形式排列，由水平和垂直路由通道连接起来。这些路由通道把信号送到器件的引脚上或者传进来，并且把 CPLD 内部的逻辑群连接在一起。CPLD 之所以称作粗粒，是因为与路由数量相比，逻辑群要大得多。

CPLD 最基本的单元是宏单元。一个宏单元包含一个寄存器（使用多达 16 个乘积项作为其输入）及多个乘积项单元。因为每个宏单元（Macrocell）用了多个乘积项，因此设计人员可部署大量的组合逻辑而不用增加额外的路径，这就是为何 CPLD 被认为是“逻辑丰富”型器件的原因。宏单元以逻辑模块（Logic Block，LB）的形式排列，每个逻辑模块由多个宏单元组成。

图 1-1 为 Xilinx 公司 XC9500 系列 CPLD 器件结构图。器件主要由 I/O 块（Input/Output Block，IOB）、功能块（Function Block，FB）以及将它们连接起来的快速切换矩阵（Fast Connect Switch Matrix）组成。JTAG（Joint Test Action Group，联合测试行动小组，也指一种与 IEEE 1149.1 兼容的国际标准测试协议）控制器及系统编程控制器共同完成 HDL 程序下载

控制功能。IOB 为芯片的输入输出缓冲器，每个功能块可提供 36 个输入端和 18 个输出端的可编程逻辑。每个功能块由 18 个独立的宏单元组成，宏单元内部为乘积项结构，可实现组合逻辑或寄存器功能。由“与门”阵列送出的信号可作为实现组合逻辑功能的输入信号，也可作为时钟、复位、置位、输出使能等控制信号。宏单元内的寄存器可配置成 D 型或 T 型触发器（Flip-flop），也可被旁路，从而实现组合逻辑功能。功能块结构及宏单元结构分别如图 1-2、1-3 所示。

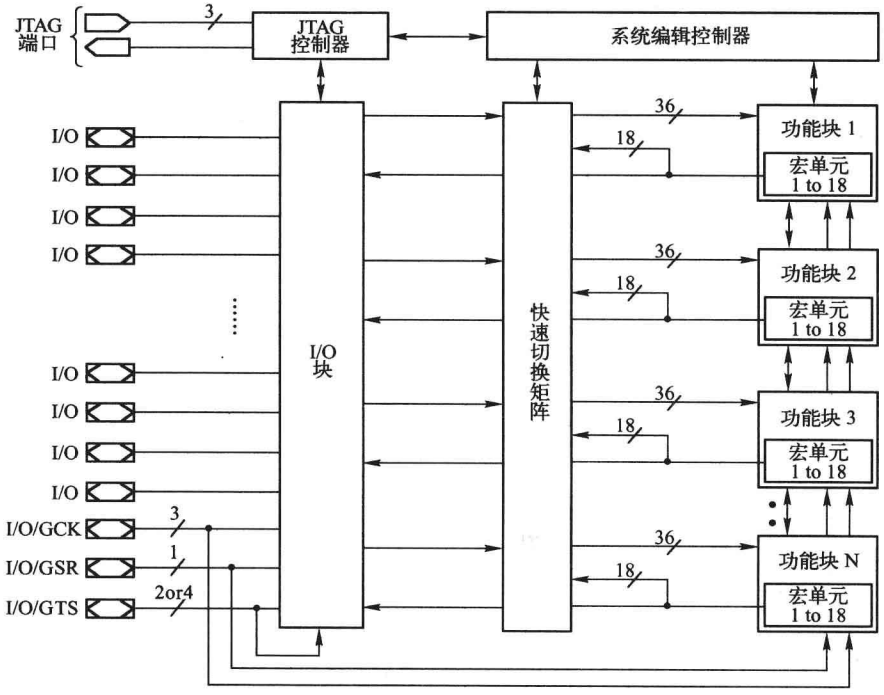


图 1-1　Xilinx 公司 XC9500 系列 CPLD 器件结构

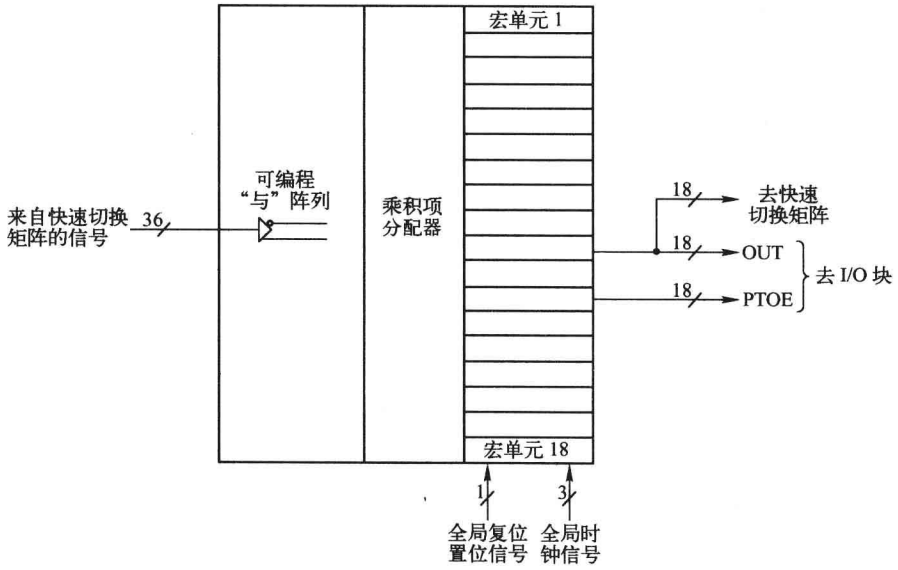


图 1-2　XC9500 系列 CPLD 器件功能模块(FB)结构

此为试读，需要完整PDF请访问：www.ertongbook.com