

# 零基础学

# FPGA

基于Altera FPGA 器件&Verilog HDL语言

葛亚明 彭永丰 薛冰◎等编著



|从学生到工程师的良师益友|



机械工业出版社  
China Machine Press

# 零基础学

# FPGA

基于Altera FPGA  
器件&Verilog HDL语言

葛亚明 彭永丰 薛冰◎等编著



机械工业出版社  
China Machine Press

本书以 Altera 公司的 FPGA 为样本,系统地介绍了 FPGA 的基本知识和相关软件的使用方法,重点讲述了 Verilog HDL 语言的编程特性、Quartus II 软件的使用技巧和 Altera 公司的高级软件工具,并给出了 SDRAM 综合设计实例。主要内容包括:FPGA 的开发流程和开发工具,Altera FPGA 的硬件结构,Verilog HDL 的基本知识,设计综合、优化和验证,基于 Altera FPGA 的开发流程,基于 Quartus II 的时序约束与分析,基于 ModelSim 的仿真,Verilog HDL 设计进阶,基于 Quartus II 的设计优化,Quartus II 的常用辅助设计工具,Altera 其他高级工具和 SRAM 控制器设计等。

全书重点突出,层次分明,注重知识的系统性、针对性和先进性;注重理论与实践联系,培养工程应用能力。另外,本书配套光盘给出了书中的实例文件、开发过程的操作录像文件、常用元器件及芯片等丰富的拓展资源,极大地方便了读者自学,动手实践。

本书既可作为高等院校电子科学与技术、微电子学、集成电路设计与集成系统、电气工程及其自动化、自动化等专业的教材,也可作为相关领域工程技术人员、IC 设计及嵌入式系统开发人员的参考书。

封底无防伪标均为盗版

版权所有,侵权必究

本书法律顾问 北京市展达律师事务所

## 图书在版编目 (CIP) 数据

零基础学 FPGA: 基于 Altera FPGA 器件 & Verilog HDL 语言/葛亚明等编著. —北京: 机械工业出版社, 2010. 7

ISBN 978-7-111-31120-1

I. 零… II. 葛… III. ①可程序逻辑器件—系统设计 ②硬件描述语言, Verilog HDL—程序设计 IV. ①TP332.1 ②TP312

中国版本图书馆 CIP 数据核字 (2010) 第 120373 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 张少波

北京诚信伟业印刷有限公司印刷

2010 年 9 月第 1 版第 1 次印刷

184mm × 260mm · 22.5 印张

标准书号: ISBN 978-7-111-31120-1

ISBN 978-7-89451-586-5 (光盘)

定价: 48.00 元 (附光盘)

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991; 88361066

购书热线: (010) 68326294; 88379649; 68995259

投稿热线: (010) 88379604

读者信箱: hzjsj@hzbook.com

FPGA 在近几年的发展中凸显出功能灵活、应用广泛的特点，不论是在简单的计数控制中，还是在复杂的算法设计中，都能满足各种需求。FPGA 设计倾向于实际应用，属于数字电路设计领域，在经过前期的模块代码设计后，通过特定的软件工具，将逻辑代码转换为映射到 FPGA 芯片的实际电路，完成代码到电路的转换。在目前越来越激烈的 IC 竞争市场中，FPGA 能迅速满足客户的定制需求，功能升级便利，是目前研究应用的热点。

为了使广大读者既能了解 FPGA 基础知识，又能将 FPGA 应用于实际，本书基于 Altera 公司的器件和 Verilog HDL 语言，结合实例对 FPGA 设计流程进行了深入浅出的论述。本书内容翔实、实践性强。在学习本书后，相信读者能够掌握 FPGA 开发的基本技术，并能够独立进行 FPGA 项目开发。

## 如何学习 FPGA 设计

学习一门技术往往从它的编程语言开始，FPGA 的编程语言有两种：VHDL 和 Verilog HDL。本书以 Verilog HDL 为例进行说明，是因为它易于学习，与 C 语言类似。如果读者具有 C 语言的基础，就不需要花很多时间便能迅速掌握 Verilog HDL。其次，精通了一家公司的 FPGA 产品，对其他几家公司的产品熟悉一下就可以了，所以本书选择应用广泛的 Altera FPGA 器件。再者，读者应熟悉 FPGA 设计的软件工具。最后，必须通过动手实践，将自己编写的程序在 FPGA 器件中运行起来。

## 本书特点

本书主要有以下特点。

### 1. 循序渐进，由浅入深

为了方便读者学习，本书在介绍 FPGA 的发展历史及其特点、开发流程和开发工具等基础上，针对 Altera FPGA 器件和 Verilog HDL 进行讲解，在后续章节中结合具体的实例，逐步介绍了 Quartus II 和 ModelSim 等常用 EDA（电子设计自动化）工具的使用、FPGA 设计的基本原则、技巧、IP 核，并通过实例对 FPGA 在接口设计领域的典型应用进行阐述。

### 2. 技术全面，内容充实

本书内容在保证实用的前提下，详细介绍了 FPGA 开发各个方面的知识。同时，针对有一定开发基础的读者，以 Altera 相关软件为例，讲解了一些 Altera 软件在时序分析及约束方面的知识。针对高级用户，讲解了 Altera 器件的一些高级特性及一些专用 IP 核的使用，并针对 Altera 软件中的高级工具作了一定的讲述。最后，介绍了目前主流的 FPGA 开发辅助工具，具有较强的实用性。读者可以根据书中的具体步骤实现所给实例，将理论与实践相结合，更利于学习。

### 3. 精选实例，注重实战

书中的每个知识点都有相应的实例代码，并对关键的代码部分进行了注释说明。本书配有大量实例，读者通过跟踪练习可加深对 FPGA 知识的掌握和理解，最终具备 FPGA 系统设计和开发的能力。

## 主要内容

本书共 13 章，各章的主要内容如下。

第 1 章：介绍 FPGA 的发展概况、开发流程和常用开发工具，通过示例讲解了 FPGA 的开发方法，以及 FPGA 常用开发工具的使用方法。读者如果是一个 FPGA 新手，本章将把读者带入 FPGA 世界。

第 2 章：介绍 Altera 公司 FPGA 的硬件结构。Altera 的主流 FPGA 分为两大类：一种侧重低成本应用，容量中等，性能可以满足一般的逻辑设计要求，如 Cyclone；还有一种侧重于高性能应用，容量大，性能可以满足各类高端应用，如 Stratix。通过本章的学习，读者可以充分理解 Altera 公司 FPGA 的结构特点和工作原理，最大限度地发挥它们在系统中的作用，使设计达到最优化。

第 3 章：介绍 Verilog HDL 语言的基本知识、Verilog HDL 语言的开发流程和使用 Verilog HDL 应具备的一些基础知识，并通过示例和实例讲解了它们的使用方法。

第 4 章：介绍 Verilog HDL 逻辑综合、语言优化和逻辑综合后的功能验证问题。逻辑综合就是从 Verilog HDL 的各种基本元素到硬件中所使用的各种基本元件的映射过程。为了减少系统的硬件资源开销，并获得最高的系统工作效率，则必须对 Verilog HDL 代码进行优化。为了验证设计的网表是否实现了规格说明书所规定的功能，则必须进行验证。

第 5 章：介绍 Quartus II 软件的功能、Altera Quartus II 设计软件为可编程片上系统（SOPC）提供的设计环境，以及如何帮助设计者进行 FPGA 设计，最后介绍了 IP 核的基本概念和分类，以及 Altera IP 核的使用流程和 OpenCore 安装。

第 6 章：介绍时序约束与时序分析的基本概念和分类、静态时序分析报告和其他场景时序分析方法，重点介绍了全局约束和个别时序约束的种类与设置方法、独立时钟和衍生时钟的定义与关系，以及最小化时序分析。

第 7 章：从入门的角度对 ModelSim 软件进行总体描述。介绍了 ModelSim 软件的安装过程和软件图形框架；通过分频电路详细描述了使用图形界面、使用交互式命令行方式和使用 Testbench 对设计进行仿真的过程以及在 ModelSim 软件中建立 Altera 仿真库的方法。

第 8 章：介绍了 Verilog 设计中的一些高级特性。叙述了在系统设计中常用的 FIFO、时钟、复位设计以及编码风格，通过本章的学习可以深化读者对 FPGA 编程思想与技巧的了解。

第 9 章：基于 Quartus 软件介绍了一些基本的设计优化方法，并介绍 Quartus 的常用工具以及基本的设计分析方法。按照设计人员在整个过程中遇到以及需要解决的问题依次介绍了时钟设计及分析、接口分析、综合报告最差时序分析及优化等。

第 10 章：介绍 Quartus II 提供的常用辅助工具。其中，引脚验证工具的主要功能是辅助设计人员进行引脚分配等检查；代码辅助工具主要是帮助设计人员进行比较直观的检查，使用代码相关工具检查软件综合后的模块关系是否与预想一致；SignalTap 是上板调试中常用的一种辅助调试工具，通过这一工具能便利地定位上板验证中出现的问题；通过 Chip Editor（芯片编辑器）可以进行一些最底层的修正，降低返工率。

第 11 章：介绍 Altera 器件的特性。主要包括时钟的管理、片内存储资源的使用及优化、片内提供的主要通信 IP 核的使用及相关内容、片内提供的常用存储控制器的实现结构框图，最后介绍了常用高速接口的核的生成以及相关配置。

第 12 章：介绍 Altera 软件提供的其他高级工具，主要包括 HardCopy、基于 Nios II 处理器的开发、DSP Builder 工具以及 SOPC Builder 相关介绍。这些工具涉及了 Altera 提供的高级特性，如结构化 ASIC 的开发、嵌入式开发等。

第 13 章：通过一个简单的 SDRAM 控制器设计，引导读者进行完整的、合理的工程开发，

通过实战，完全掌握 FPGA 的开发流程与技巧。

## 光盘内容

- 课后习题答案
- 书中全部实例文件
- 开发过程录像文件
- 常用芯片及元器件
- 常用学习交流网址

## 读者对象

- FPGA 初学者
- FPGA 开发人员
- 硬件电路工程师、IC 设计与验证工程师及嵌入式工程师
- 高等院校电子科学与技术、微电子学、集成电路设计与集成系统、电气工程及其自动化、自动化等相关专业师生

FPGA 技术发展迅速，应用领域日趋广泛。由于掌握的资料和水平有限，加之时间仓促，书中难免存在错误和不妥之处，恳请读者批评指正。

本书主要由葛亚明、彭永丰、薛冰编著，李军老师详细审阅了本书的全部底稿并提出了许多宝贵意见。此外，其他参与编著和资料整理的人员还有宋一兵、管殿柱、张兰勇、张玉廷、李高云、王五桂、常绪成、杨震、刘业明、程垠钟、范国良、顾铭明、贾云璐、李莉红、李秀鹏、刘杨、任冬、宋颖慧等，在此一并表示感谢！

感谢您选择了本书，希望我们的努力对您的工作和学习有所帮助，也希望您把对本书的意见和建议告诉我们。

作者联系方式：[gdz\\_zero@126.com](mailto:gdz_zero@126.com)

编辑联系方式：[sdl@hzbook.com](mailto:sdl@hzbook.com)

作者  
2010年6月

## 目 录 CONTENTS

### 前言

### 第1章 FPGA的开发流程和开发工具 ..... 1

1.1	FPGA 概述 .....	1
1.1.1	FPGA 的发展 .....	1
1.1.2	FPGA 的生产厂家及其产品 ...	2
1.2	FPGA 的开发流程 .....	4
1.2.1	FPGA 的设计方法 .....	4
1.2.2	典型 FPGA 的开发流程 .....	5
1.2.3	基于 FPGA 的 SOC 设计方法 .....	9
1.2.4	基于 IP 核的设计方法 .....	9
1.3	FPGA 的常用开发工具 .....	11
1.3.1	代码输入工具 .....	12
1.3.2	综合工具 .....	13
1.3.3	仿真工具 .....	13
1.3.4	实现与优化工具 .....	14
1.3.5	EDA 工具 .....	14
1.4	实践拓展 .....	15
1.5	思考与练习 .....	16

### 第2章 Altera FPGA 的硬件结构 ..... 17

2.1	主流高端 FPGA——Stratix 和 Stratix II .....	17
2.1.1	Stratix 器件 .....	17
2.1.2	Stratix II 器件 .....	27
2.2	Stratix IV FPGA 器件 .....	29

2.2.1	Stratix IV FPGA 的核心架构 .....	30
2.2.2	Stratix IV FPGA TriMatrix 存储器 .....	31
2.3	主流低端 FPGA——Cyclone 和 Cyclone II .....	32
2.3.1	Cyclone 器件 .....	32
2.3.2	Cyclone II 器件 .....	36
2.4	Cyclone IV 器件 .....	37
2.5	实例：FPGA 最小系统设计 .....	38
2.6	实践拓展 .....	46
2.7	思考与练习 .....	47

### 第3章 Verilog HDL 的基本知识 ... 48

3.1	Verilog HDL 简介 .....	48
3.2	Verilog HDL 的开发流程 .....	49
3.3	Verilog HDL 的基本概念 .....	50
3.3.1	程序格式 .....	50
3.3.2	注释、间隔符和标识符 ...	51
3.3.3	数值和字符串 .....	53
3.3.4	数据类型 .....	55
3.3.5	编译指令 .....	56
3.3.6	系统任务和函数结构 .....	59
3.3.7	运算符和表达式 .....	63
3.4	行为级描述 .....	66
3.4.1	过程结构 .....	66
3.4.2	语句块 .....	67

3.4.3	时序控制	68	<b>第5章 基于 Altera FPGA 的开发 流程</b>	103	
3.4.4	赋值语句	71			
3.4.5	分支语句	75			
3.4.6	循环控制语句	78			
3.4.7	任务与函数	80			
3.5	结构级描述	81			
3.5.1	模块级建模	81			
3.5.2	门级建模	82			
3.6	实例: 交通灯控制器设计	83			
3.7	实践拓展	86			
3.8	思考与练习	87			
<b>第4章 设计综合、优化和验证</b>					
4.1	Verilog HDL 语言综合	88	5.1	Quartus II 软件功能与特点	103
4.1.1	综合的基本概念	88	5.2	Quartus II 软件安装与授权	105
4.1.2	可综合的 Verilog HDL 结构	89	5.2.1	Quartus II 软件的安装 过程	105
4.1.3	可综合的 Verilog HDL 操作符	90	5.2.2	Quartus II 软件的授权 文件	108
4.1.4	通常忽略的 Verilog HDL 结构	90	5.2.3	在 Quartus II 软件中指定 授权文件	109
4.1.5	不可综合的 Verilog HDL 结构	90	5.3	Quartus II 软件的用户界面	110
4.2	Verilog HDL 语言设计优化	91	5.4	Quartus II 软件开发流程	113
4.2.1	公因子和公共子表达式	91	5.4.1	图形用户界面设计流程	113
4.2.2	循环语句优化	91	5.4.2	EDA 工具设计流程	115
4.2.3	触发器和锁存器的优化	92	5.4.3	命令行设计流程	117
4.2.4	算术表达式优化	92	5.5	设计输入	118
4.2.5	运算符优化	93	5.5.1	建立工程	119
4.2.6	其他优化方式	93	5.5.2	建立设计文件	120
4.3	实例: 阶乘模型	95	5.5.3	指定初始设计的约束 条件	121
4.4	功能验证	96	5.6	设计综合	122
4.4.1	验证方法学	96	5.6.1	使用 Quartus II 的集成 综合	123
4.4.2	搭建一个简单 TestBench	97	5.6.2	使用其他 EDA 综合工具	123
4.5	实例: 移位寄存器的测试 程序	99	5.7	布局布线	124
4.6	实践拓展	100	5.7.1	设置布局布线参数	125
4.7	思考与练习	102	5.7.2	物理综合优化参数设置	127
			5.7.3	分析适配结果	129
			5.7.4	优化适配	131
			5.8	仿真验证	135
			5.8.1	使用 EDA 工具进行仿真 设计	135
			5.8.2	使用 Quartus II 仿真器进行 仿真设计	138
			5.9	编程与配置	139
			5.10	实例: 3 线-8 线译码器电路 设计	142



5.11 IP 核 .....	149	7.3 ModelSim 图形用户界面 .....	180
5.11.1 IP 核的基本概念与分类 ..	149	7.4 ModelSim 的基本仿真方法 .....	180
5.11.2 Altera 公司的 IP 核及其 使用流程 .....	150	7.5 实例: 分频电路的图形界面 仿真 .....	180
5.11.3 OpenCore 的安装 .....	151	7.6 实例: 分频电路的命令行方式 仿真 .....	189
5.12 实践拓展 .....	151	7.7 实例: 分频电路的 TestBench 仿真 .....	191
5.13 思考与练习 .....	154	7.8 ModelSim 仿真工具高级操作 ..	195
<b>第 6 章 基于 Quartus II 的时序 约束与分析 .....</b>	<b>155</b>	7.8.1 force 命令 .....	195
6.1 时序约束与时序分析的基本 概念 .....	155	7.8.2 DO 文件 .....	195
6.1.1 周期和最高频率 .....	155	7.8.3 modelsim. ini 文件 .....	196
6.1.2 时钟建立时间和保持 时间 .....	156	7.8.4 SDF 文件 .....	196
6.1.3 时钟到输出延时和引脚 到引脚延时 .....	157	7.9 在 ModelSim SE 中创建 Altera 的仿真库 .....	196
6.1.4 时钟偏斜 .....	157	7.10 实践拓展 .....	198
6.2 时序约束的设置 .....	158	7.11 思考与练习 .....	201
6.2.1 设置全局时序约束 .....	158	<b>第 8 章 设计技巧及风格 .....</b>	<b>202</b>
6.2.2 设置个别时序约束 .....	162	8.1 Verilog HDL 设计进阶 .....	202
6.3 静态时序分析报告 .....	166	8.1.1 状态机设计 .....	202
6.4 实例: 定位到 Floorplan .....	168	8.1.2 速度与面积原则 .....	206
6.5 实例: 定位到工艺映射查 看器 .....	169	8.1.3 流水线设计 .....	206
6.6 其他场景时序分析方法 .....	171	8.1.4 异步时钟域设计 .....	208
6.6.1 多时钟域场景时序分析 ..	171	8.1.5 乒乓操作 .....	210
6.6.2 多周期约束 .....	171	8.2 实例: 串并转换 .....	211
6.6.3 异步时钟域时序分析 .....	173	8.3 层次化的设计 .....	214
6.7 最小化时序分析 .....	173	8.4 FIFO 设计 .....	215
6.8 实践拓展 .....	174	8.5 实例: 同步 FIFO 设计 .....	216
6.9 思考与练习 .....	175	8.6 实例: 异步 FIFO 设计 .....	219
<b>第 7 章 基于 ModelSim 的仿真 ..</b>	<b>176</b>	8.7 时钟设计 .....	223
7.1 ModelSim 简介 .....	176	8.7.1 数字锁相环介绍 .....	223
7.2 ModelSim 仿真软件的安装 .....	177	8.7.2 全局时钟网络应用设计 ..	224

8.8	复位设计 .....	225
8.8.1	同步复位 .....	225
8.8.2	异步复位 .....	226
8.9	编码风格 .....	226
8.9.1	coding style 的意义 .....	226
8.9.2	可重用设计 .....	227
8.9.3	组合逻辑设计 .....	228
8.9.4	同步逻辑设计 .....	229
8.9.5	信号敏感列表 .....	231
8.9.6	状态机设计的一般原则 .....	231
8.9.7	三态信号的设计 .....	232
8.10	实践拓展 .....	233
8.11	思考与练习 .....	233
<b>第9章 基于 Quartus II 的设计</b>		
	<b>优化 .....</b>	<b>234</b>
9.1	设计分析 .....	234
9.1.1	时钟资源分析 .....	234
9.1.2	I/O 接口分析 .....	235
9.1.3	最差路径分析 .....	235
9.2	设计优化基础 .....	236
9.2.1	设计优化基本流程 .....	237
9.2.2	首次编译的约束设置 .....	237
9.2.3	查看编译报告 .....	239
9.3	资源优化 .....	241
9.4	时钟频率优化 .....	244
9.4.1	设计优化 .....	244
9.4.2	布局布线工具设置 .....	245
9.4.3	网表优化和物理综合 .....	246
9.4.4	使用 LogicLock 优化 .....	247
9.5	增量编译 .....	249
9.6	实例计数器设计 .....	250
9.7	实践拓展 .....	254
9.8	思考与练习 .....	254

<b>第10章 Quartus II 的常用辅助</b>		
	<b>设计工具 .....</b>	<b>255</b>
10.1	引脚验证 .....	255
10.1.1	验证流程 .....	255
10.1.2	验证结果分析 .....	256
10.2	代码辅助工具 .....	258
10.2.1	RTL 用户界面 .....	258
10.2.2	原理图选择 .....	260
10.2.3	原理图关联 .....	262
10.2.4	使用 RTL Viewer 辅助 定位问题 .....	264
10.3	SignalProbe 及 SignalTap II 逻辑分析器 .....	264
10.3.1	SignalProbe .....	264
10.3.2	SignalTap II 逻辑分析器 .....	265
10.4	Chip Editor 底层编辑器 .....	268
10.4.1	Chip Editor 功能 .....	269
10.4.2	Chip Editor 视图 .....	269
10.4.3	Chip Editor 编辑使用 方法 .....	271
10.4.4	Chip Editor 应用 .....	273
10.5	实例：状态机设计 .....	273
10.6	实践拓展 .....	281
10.7	思考与练习 .....	282
<b>第11章 Altera 器件的其他特性</b>		<b>283</b>
11.1	时钟管理 .....	283
11.1.1	Altera 器件的时钟资源 .....	283
11.1.2	基于 Altera 器件的时钟 分配及管理 .....	284
11.2	片内存储资源 .....	287
11.2.1	Altera 器件的主要存储 资源 .....	287
11.2.2	Altera 主要存储资源的 使用 .....	288
11.2.3	Altera 器件存储资源的 优化 .....	293

11.3 常用通信 IP CORE .....	293	12.3.2 DSP Builder 设计流程 ...	318
11.3.1 主要通信用 IP CORE ...	293	12.4 SOPC Builder .....	321
11.3.2 FFT IP CORE .....	294	12.4.1 SOPC 技术.....	321
11.3.3 内部 DSP 使用 .....	296	12.4.2 SOPC Builder 使用 .....	321
11.4 常用存储控制器 .....	296	12.4.3 SOPC 设计流程.....	323
11.4.1 QDR SRAM 控制器 .....	296	12.5 实例：基于 Nios 系统开发 ...	324
11.4.2 DDR 控制器 .....	299	12.6 实践拓展 .....	332
11.5 高速接口 .....	300	12.7 思考与练习 .....	332
11.5.1 SPI-4 接口 .....	300	<b>第 13 章 SDRAM 控制器设计 ...</b>	<b>333</b>
11.5.2 Serdes 接口 .....	302	13.1 规格说明 .....	333
11.6 实例：异步接口转换设计 ...	302	13.1.1 SDRAM 芯片特性 .....	333
11.7 实践拓展 .....	311	13.1.2 设计要求 .....	334
11.8 思考与练习 .....	312	13.2 控制器分析及系统架构 .....	335
<b>第 12 章 Altera 其他高级工具 ...</b>	<b>313</b>	13.2.1 控制器分析 .....	335
12.1 HardCopy 流程 .....	313	13.2.2 控制器实现架构设计 ...	337
12.2 基于 Nios II 处理器的嵌入式 系统设计 .....	314	13.3 程序设计 .....	338
12.2.1 Nios CPU .....	314	13.4 验证环境搭建 .....	342
12.2.2 Avalon 总线 .....	315	13.5 系统验证 .....	343
12.2.3 软件开发流程与方法 ...	315	13.6 综合布线 .....	344
12.2.4 外设及中断 .....	316	13.7 实践拓展 .....	345
12.3 DSP Builder 工具 .....	317	13.8 思考与练习 .....	346
12.3.1 DSP Builder 软件安装与 设置 .....	317	<b>参考文献 .....</b>	<b>347</b>

# 第 1 章 FPGA 的开发流程和开发工具

## 本章要点

- FPGA 的发展历史、主要生产厂商及其产品
- FPGA 的开发流程、软硬件设计方法
- 基于 FPGA 的 SOC 设计及基于 IP 核的设计方法
- FPGA 的开发工具——代码输入工具、综合工具、仿真工具和实现与优化工具

## 1.1 FPGA 概述

FPGA 是英文 Field Programmable Gate Array 的缩写，即现场可编程门阵列。它是在可编程阵列逻辑（Programmable Array Logic, PAL）、门阵列逻辑（Gate Array Logic, GAL）、可编程逻辑器件（Programmable Logic Device, PLD）等可编程器件的基础上进一步发展的产物。它是作为专用集成电路（Application Specific Integrated Circuit, ASIC）领域中的一种半定制电路出现的，既解决了定制电路的不足，又克服了原有可编程器件门电路数有限的缺点。通过开发，FPGA 能完成任何数字器件的功能，上至高性能 CPU，下至简单的 74 系列电路，都可以用 FPGA 来实现。

### 1.1.1 FPGA 的发展

1985 年，Xilinx 公司推出了世界上第一款真正意义上的 FPGA。推出的这款 FPGA 包括两个器件和支持布局布线的设计工具。FPGA 发展异常迅速，在不到十年的时间里，时钟频率就从不到 10MHz 提高到 100MHz。设计工艺已经达到亚微米级别，FPGA 芯片的规模也从几千门增加到两万多等效门。大量功能强大且易于使用的软件工具也相继推出，使得 FPGA 很快占领了电子设计领域的一块高端市场。

20 世纪 80 年代推出的 FPGA 可以说是 Intel 公司于 1971 年推出第一款商用微处理器的延续。那个时期，典型的微处理器系统包含微处理器、存储器和一些特殊功能的中小规模（MSI/SSI）器件。为追求更小的尺寸、更低的成本、更快的错误恢复能力、高可靠性以及更快更易使用的原型，集成电路的设计者都意识到一定会有一种器件要取代当时的中小规模电路。这个概念的第一次尝试是 Signetics 公司于 1975 年推出的 83S100 FPLA（现场可编程逻辑阵列）。这款可编程器件实际上是一款 PLA 结构的器件。它由 16 个输入、48 个乘积项与阵列、8 个输出、48 个乘积项或阵列构成，通过 Ni-Cr（镍-铬）熔丝实现连续的断开或连接。这种方法在以降低速度和增加功耗为代价的前提下，给设计师以较大的电路设计空间。但是这款可编程器件需要人工来设置熔丝的断开或连接，因此实现起来比较复杂且容易出错。

鉴于当时各种可编程器件的速度、结构等都不能很好地满足市场的需求，Xilinx 于 1985 年推出了 2000 系列的 FPGA。该系列的 FPGA 是世界上第一款基于 SRAM 的可编程 FPGA，包括两个器件：第一个器件由  $8 \times 8$ （共 64 个）的可配置逻辑模块（Configurable Logic Block, CLB）构成，并在芯片的周边提供了 58 个输入/输出接口模块（I/O Block, IOB）；

第二个器件由  $10 \times 10$  的 CLB 构成, 并提供了总共 74 个 IOB 单元。自 Xilinx 推出第一款 FPGA 之后, 其他公司也相继推出各自的 FPGA 品。如 Actel 推出很有特色的反熔丝 (Anti-fused) FPGA。FPGA 市场内的竞争也愈演愈烈, IC 的制造商都意识到必须提供更加强大、更易使用的新产品才能占据市场。在这种形势下, Xilinx 在 1987 年就推出了它的第二款 3000 系列的 FPGA, 距第一款 FPGA 的推出只有两年的时间。也就是在那时, AT&T 成功地获得了这款 FPGA 器件的设计使用权。并开始提供自己的芯片和开发系统, 即 AT&T 3000 系列 FPGA。

自从第二代 FPGA 问世以来, 各种 FPGA 的应用层出不穷, 电路复杂度也相继上升。这时, Xilinx 就开始研制第三代 FPGA 产品, AT&T 也开始开发自己的下一代 FPGA。Xilinx 的第三代 FPGA 产品于 1991 年问世, 而 AT&T 的下一代产品直到 1992 年才研制成功。认识到 FPGA 市场潜在的广阔空间, 很多 IC 和软件厂商也开始向 FPGA 领域进军, 包括一些著名的公司, 如 Actel、AMD、Altera、Intel、Mento Graphics、Texas Instruments 以及 Toshiba。

目前, 比较典型的 FPGA 器件是 Xilinx 公司的 FPGA 器件系列和 Altera 公司的 FPGA 器件系列, 它们开发较早, 占据了较大的 PLD/FPGA 市场份额。截至目前, 在欧洲用 Xilinx 的人多, 在亚太地区用 Altera 的人多, 在美国则是平分秋色。全球 PLD/FPGA 产品 60% 以上是由 Altera 和 Xilinx 提供的。可以预见, Altera 和 Xilinx 将共同决定可编程逻辑器件技术的发展方向。当然, 其他众多 IC 厂商也在不断为技术的进步而努力, 如 Lattice、Vantis、Actel、Quicklogic、Lucent 等。

## 1.1.2 FPGA 的生产厂家及其产品

目前世界上有十几家生产 FPGA 的公司, 最大的三家是: Altera、Xilinx 和 Lattice, 其中 Altera 和 Xilinx 占据了主流的市场份额。

### 1. Altera 公司产品

Altera 公司九十年代以来发展迅猛, 是最大可编程逻辑器件供应商之一。Altera 的主流 FPGA 分为两大类, 一种侧重低成本应用, 容量中等, 性能可以满足一般的逻辑设计要求, 如 Cyclone, Cyclone II 等; 还有一种侧重于高性能应用, 容量大, 性能可以满足各类高端应用, 如 Stratix, Stratix II 等。

#### (1) Stratix 系列

Stratix 系列 FPGA 结合了高密度、高性能以及丰富的特性, 能够集成更多的功能, 提高系统带宽。表 1-1 给出了 Stratix 系列 FPGA 的推出时间、工艺技术及其建议在新设计中的使用情况。

表 1-1 Stratix 系列介绍

器件系列	Stratix	Stratix GX	Stratix II	Stratix II GX	Stratix III	Stratix IV
推出时间	2002	2003	2004	2005	2006	2008
工艺技术	130nm	130nm	90nm	90nm	65nm	40nm
是否建议在新设计中使用	否	否	是	是	是	是

- 所有 Stratix FPGA 系列都有等价的 HardCopy ASIC 器件。通过对设计进行 FPGA 原型开发, HardCopy ASIC 提供了低风险、低成本的量产途径。Stratix 系列 FPGA 也适用于标准单元 ASIC 的原型开发和验证。
- Stratix IV FPGA 在所有 40nm FPGA 中, 具有最大的密度、最好的性能和最低的功耗。

Stratix IV FPGA 系列提供增强型 (E) 和带有收发器 (GX 和 GT) 的增强型器件, 满足了无线和固网通信、军事、广播等众多市场和应用的需求。这一高性能 40nm FPGA 系列包含了同类最佳的 11.3Gbit/s 收发器。

- Stratix III FPGA 是业界功耗最低的高性能 65nm FPGA。可以借助逻辑型 (L)、存储器增强型 (E) 和数字信号处理型 (DSP) 来综合考虑设计资源要求, 而不会采用资源比实际需求大得多的器件进行设计, 从而节省了电路板, 缩短了编译时间, 降低了成本。Stratix III FPGA 面向大量应用的高端内核系统处理设计。
- Stratix II 和 Stratix II GX 型 FPGA 引入了自适应逻辑模块 (ALM) 体系结构, 采用了高性能 8 输入分段式查找表 (LUT) 来替代 4 输入 LUT, 这也是 Altera 目前最新的高端 FPGA 所采用的结构。这些 Stratix II GX FPGA 可以批量供货, 强烈建议新设计采用这一型号产品。
- Stratix 和 Stratix GX 是 Stratix FPGA 系列中最早的型号产品。这一高性能 FPGA 系列引入了 DSP 硬核 IP 模块以及 Altera 应用广泛的 TriMatrix 片内存储器和灵活的 I/O 结构。

## (2) Cyclone 系列

Cyclone 系列的 FPGA 是 Altera 中等规模 FPGA, 由 2003 年推出, 0.13 $\mu\text{m}$  工艺, 1.5V 内核供电, 与 Stratix 结构类似, 是一种低成本 FPGA 系列。

- Cyclone (飓风): Altera 最成功的器件之一, 性价比不错, 是一种适合中低端应用的通用 FPGA, 推荐使用。
- Cyclone II: Cyclone 的下一代产品, 2005 年推出, 90nm 工艺, 1.2V 内核供电, 属于低成本 FPGA, 性能和 Cyclone 相当, 提供了硬件乘法器单元。

## 2. Xilinx 公司产品

Xilinx 是 FPGA 的发明者, 老牌 PLD 公司, 是最大可编程逻辑器件供应商之一。产品种类齐全, 主要有 XC9500/4000、Cool Runner (XPLA3)、Spartan、Virtex 等。开发软件为 Foundation 和 ISE。通常, 在欧洲用 Xilinx 的人多, 在亚太地区用 Altera 的人多, 在美国则是平分秋色。全球 PLD/FPGA 产品 60% 以上是由 Altera 和 Xilinx 提供的, 可以说 Altera 和 Xilinx 共同决定了 PLD 技术的发展方向。也有一种说法认为 Xilinx 的高端 FPGA 领先其他厂商约半年左右的发展。

2004 年中期, Xilinx 在高集成度 FPGA 产品线上推出了最新的 Virtex-4 平台级 FPGA。新型 FPGA 通过实现大量基于 FPGA 的 RISC 处理器和嵌入处理器内核, 不仅能实现高度复杂逻辑控制, 还兼备强大的运算功能, 开始向传统的 DSP 应用领域发展。

Virtex 系列 FPGA 集成了许多满足系统级设计要求的新性能, 具有独特的结构特点。整个 Virtex 系列由 9 种器件组成, 系统门数从 5 万到 100 万门 (1728 到 27648 个逻辑单元), 提供给用户的 I/O 引脚数最多超过 500 个, 采用多种封装形式, 包括先进的 1.0mm Fine-Pitch TMBGA 和 0.8mm 芯片封装, 采用 5 层金属的 0.22 $\mu\text{m}$  CMOS 工艺, 实现 5V 容差的 I/O 接口, 借助于优选的时序驱动的布局和布线工具, 在 400MHz 的 P II CPU 上, 编译速度可达 20 万门/秒。

在推出 Virtex FPGA 之后不到一年, Xilinx 又推出了 Virtex-E 系列产品, 其性能和密度可与 ASIC 匹敌。Virtex-E 系列产品的特点是: 拥有 320 万个系统门, 832Kbit 的真双端口内部块状 RAM, 8 个 DLL 并支持超过 20 种不同的信号标准, 包括 LVDS、Bus LVDS 以及 LVPECL, 采用 0.18 $\mu\text{m}$  工艺制造, 在单个器件上实现了 2.1 亿个晶体管的密度。总之, Virtex 和 Virtex-E 系列不仅将 FPGA 性能推向一个新层面, 还解决了向系统集成的挑战。

Xilinx 产品的另一个发展方向是实现可编程逻辑器件在大批量生产中的应用, 所以对成本要求更高。Spartan 系列是以 XC4000 系列结构为基础, 并结合片上 RAM、强大的 IP 库支

持和大容量、低成本的特点,使其可在大批量生产中替代 ASIC。

Spartan 系列的主要特点是:系统门数可达 40000 门,灵活的片上存储器,分布式和块存储器,4 个数字延迟锁相环,有效的芯片级/板级时钟管理,Select I/O 技术保证同所有主要总线标准如 HSTL、GTL、SSTL 等的接口,具有功率管理(睡眠模式)。

### 3. Lattice

Lattice 是 ISP(在系统编程)技术的发明者,ISP 技术极大地促进了 PLD 产品的发展,与 Altera 和 Xilinx 相比,其开发工具比 Altera 和 Xilinx 略逊一筹。中小规模 PLD 比较有特色,不过大规模 PLD、FPGA 的竞争力还不够强。1999 年推出可编程模拟器件。1999 年收购 Vantis(原 AMD 子公司),成为第三大可编程逻辑器件供应商。2001 年 12 月收购 agere 公司(原 Lucent 微电子部)的 FPGA 部门。主要产品有 ispLSI2000/5000/8000, MACH4/5, ispM-ACH4000 等。

### 4. Actel

Actel 公司一直是世界反熔丝技术 FPGA 的领先供应商,主要有两大系列的反熔丝 FPGA 产品 SX-A 系列和 MX 高速系列。SX-A 系列 FPGA 的主要特点是功耗低,在接上了所有内部寄存器之后,200MHz 运行时的功耗不到 1W,而且价格也较为低廉,并拥有良好的性能。

由于反熔丝 PLD 具有抗辐射、耐高低温、功耗低和速度快的特性,所以在军品和宇航级上有较大优势。2004 年 3 月欧洲航天局发射的 Rosseta 太空船上就使用了约 400 片 Actel 的耐辐射 FPGA,在飞船的主计算机、电源系统、大规模存储控制器和天线点控制中发挥了令人满意的作用。Altera 和 Xilinx 则一般不涉足军品和宇航级市场。

SX-A (0.22/0.25 $\mu\text{m}$ ) 和 SX (0.35 $\mu\text{m}$ ) FPGA 系列可以提供 12000 ~ 108000 个可用门,64bit、66MHz 的 PCI,330MHz 的内部时钟频率,4ns 的时钟延迟,它的输入设置时间小于 0.6ns,不需要逐步锁定的循环指令,可提供 2.5V,3.3V 和 5V 的电压。这就使 FPGA 能够具有一些以前无法实现的功能,设计者能够把多个高性能的 CPLD 压缩到一片 FPGA 中,大大降低了功耗,节省了电路板空间,减少了费用。

## 1.2 FPGA 的开发流程

FPGA 的开发流程是利用 EDA 开发软件和编程工具对 FPGA 芯片进行开发的过程。本节首先介绍 FPGA 设计方法,其次给出典型 PFGA 开发流程,最后对基于 FPGA 的 SOC 设计方法和基于 IP 核的设计方法进行介绍。

### 1.2.1 FPGA 的设计方法

FPGA 是可编程芯片,因此 FPGA 的设计方法包括硬件设计和软件设计两部分。硬件包括 FPGA 芯片电路、存储器、输入/输出接口电路以及其他设备,软件即是相应的 HDL 程序以及嵌入式 C 程序。硬件设计是基础,但其方法比较固定,本书主要介绍软件的设计方法。

总的来说,在 FPGA 设计中,一般有两种方法:自上向下设计或自下向上设计。对较大规模的设计一般采用前者,首先进行模块分割,把模块分割成子模块,然后再把子模块分割成下一级子模块,一次逐级划分,利用模块分割可以简化设计,提高程序的可读性,提高逻辑综合效率,提高程序的可移植性。从而便于理解设计所完成的功能,便于读懂程序及调试。有利于提高设计的性能和可靠性。

目前微电子技术已经发展到 SOC 阶段,即集成系统阶段,相对于集成电路的设计思想有着革命性的变化。SOC 是一个复杂的系统,它将一个完整产品的功能集成在一个芯片上,包括核心处理器、存储单元、硬件加速单元以及众多的外部设备接口等,具有设计周期长、

实现成本高等特点，因此其设计方法必然是自顶向下的从系统级到功能模块的软、硬件协同设计，达到软、硬件的无缝结合。

这么庞大的工作量显然超出了单个工程师的能力，因此需要按照层次化、结构化的设计方法来实施。首先由总设计师将整个软件开发任务划分为若干个可操作的模块，并对其接口和资源进行评估，编制出相应的行为或结构模型，再将其分配给下一层的设计师。这就允许多个设计者同时设计一个硬件系统中的不同模块，并为自己所设计的模块负责，然后由上层设计师对下层模块进行功能验证。

自顶向下的设计流程从系统设计开始，划分为若干个二级单元，然后再把各个二级单元划分为下一层次的基本单元，一直下去，直到能够使用基本模块或者 IP 核直接实现为止。流行的 FPGA 开发工具都提供了层次化管理，可以有效地梳理错综复杂的层次，能够方便地查看某一层次模块的源代码以修改错误。

在工程实践中，还存在软件编译时长的问题。由于大型设计包含多个复杂的功能模块，其时序收敛与仿真验证复杂度很高，为了满足时序指标的要求，往往需要反复修改源文件，再对所修改的新版本进行重新编译，直到满足要求为止。这里面存在两个问题：首先，软件编译一次需要长达数小时甚至数周的时间，这是开发所不能容忍的；其次，重新编译和布局布线后结果差异很大，会将已满足时序的电路破坏。因此必须提出一种有效提高设计性能、继承已有结果、便于团队化设计的软件工具。FPGA 厂商意识到这类需求，由此开发出了相应的逻辑锁定和增量设计的软件工具。

### 1.2.2 典型 FPGA 的开发流程

FPGA 的开发流程一般包括设计定义、设计输入、功能仿真、综合优化、综合后仿真、实现、布线后仿真、板级仿真以及芯片编程与调试等主要步骤。图 1-1 为 FPGA 设计流程图。

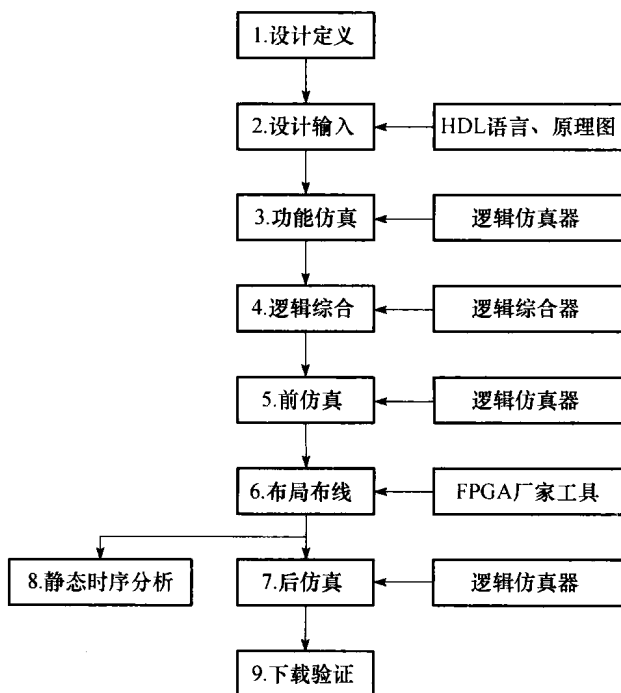


图 1-1 FPGA 设计流程



图 1-1 中各步骤涉及的工具说明如下：

- HDL 语言指 VHDL<sup>⊖</sup>和 Verilog HDL 等。
- 逻辑仿真器主要指 ModelSim、Verilog-XL 等。
- 逻辑综合器主要指 LeonardoSpectrum、Synplify、FPGA Express/FPGA Compiler 等。
- FPGA 厂家工具指的是如 Altera 的 Max Plus II、Quartus II，Xilinx 的 Foundation、Alliance、ISE 等。

### 1. 设计定义

这是由系统概要设计指导和详细设计具体规定下的 FPGA 模块必须完成的功能以及与外围器件的接口，包括接口信号规格、处理时钟频率、时序要求、管脚分配锁定等，是对 FPGA 进行编程设定的依据。对设计定义的要求是合理、清晰、准确。

### 2. 设计输入

设计输入是将所设计的系统或电路以开发软件要求的某种形式表示出来，并输入给 EDA 工具的过程。常用的方法有硬件描述语言（Hardware Description Language, HDL）与原理图输入两种方式。HDL 设计方式是现今设计大规模数字集成电路的常用形式，除 IEEE 标准中 VHDL 与 Verilog HDL 两种形式外，尚有各自 FPGA 厂家推出的专用语言，如 Quartus 下的 AHDL。HDL 语言描述在状态机（state machine）、控制逻辑、总线功能方面较强，使其描述的电路能在特定综合器（如 Synopsys 公司的 FPGA Compiler II 或 FPGA Express）作用下以具体硬件单元较好地实现。原理图输入方式是一种最直接的描述方式，在可编程芯片发展的早期应用比较广泛，它将所需的器件从元件库中调出来，画出原理图。原理图输入在顶层设计、数据通路逻辑、手工最优化电路等方面具有图形化直观、单元节俭、功能明确等特点，但效率很低，且不易维护，不利于构造和重用。更主要的缺点是可移植性差，当芯片升级后，所有的原理图都需要做一定的改动。

目前的设计趋势是以 HDL 语言为主，原理图为辅，进行混合设计以发挥二者各自的长处。HDL 语言中，VHDL 的数据类型丰富，对大型系统的描述能力强；Verilog 对 RTL 和门级等底层电路描述能力强，风格类似 C 语言。使用上，VHDL 在欧洲使用较多，Verilog 则在北美和亚洲使用较多。

### 3. 功能仿真

功能仿真也称前仿真，是在编译之前对用户所设计的电路进行逻辑功能验证，此时的仿真没有延迟信息，仅对初步的功能进行检测。仿真前，要先利用波形编辑器和 HDL 等建立波形文件和测试向量，仿真结果将会生成报告文件和输出信号波形，从中可以观察各个节点信号的变化。如果发现错误，则返回设计修改逻辑设计。图 1-2 为功能仿真示意图。

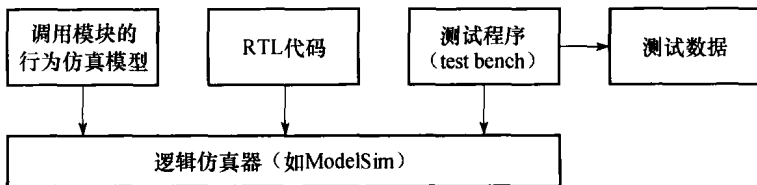


图 1-2 功能仿真示意图

从广义上讲，设计验证包括功能与时序仿真和电路验证。仿真是指使用设计软件包对已实现的设计进行完整测试，模拟实际物理环境下的工作情况。前仿真是指仅对逻辑功能进行

⊖ VHDL 系 Very-High-Speed Integrated Circuit Hardware Description Language 的简称，即超高速集成电路硬件描述语言。