

设计者的思想

Xilinx FPGA 高级设计及应用

汤琦 蒋军敏 编著



FPGA设计必备，让你轻松应对各种FPGA设计项目



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

设计者的思想

Xilinx FPGA 高级设计及应用

汤 琦 蒋军敏 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 提 要

本书用简洁的语言向读者展示了 FPGA 高质量和可靠性设计中必须掌握的概念、思想和设计方法,如 FPGA 设计思想、高速数据传输系统设计、高级配置管理、空间应用可靠性设计、分区设计和高效测试系统设计等。本书来源于工程实际,选取的专题都是实际工程开发中必须面对、难度很高的问题。作者结合多年的工作经验编写了本书,书中深入剖析了其实现原理和技术细节,并提供了使用方法和实例。

本书可作为从事 FPGA 设计的工程技术人员、硬件工程师和 IC 工程师的学习、参考用书,也可作为电子信息、通信工程及相关工科专业的教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

Xilinx FPGA 高级设计及应用/汤琦,蒋军敏编著. —北京:电子工业出版社,2012.4
(设计者的思想)

ISBN 978-7-121-16626-6

I. ①X… II. ①汤… ②蒋… III. ①可程序逻辑器件-系统设计 IV. ①TP332.1-62

中国版本图书馆 CIP 数据核字(2012)第 051976 号

策划编辑:王敬栋(wangjd@phei.com.cn)

责任编辑:李蕊

印 刷:北京东光印刷厂

装 订:三河市鹏成印业有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:18.5 字数:473.6 千字

印 次:2012 年 4 月第 1 次印刷

印 数:4 000 册 定价:49.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlls@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

FPGA (Field Programmable Gate Array) 是 1984 年由 Xilinx 公司提出的一类半定制通用器件。用户可以通过对 FPGA 器件的设计及配置来实现所需的逻辑功能。相比传统的 IC 设计来说,一方面基于 FPGA 的电路设计具有开发时间短、成本低廉、可靠性得到验证、开发风险小等优点;另一方面针对基于 SRAM 的 FPGA 具有可重复、在线或离线配置等特点,为设计的更改、升级和重构提供了基础,使得所设计的电路易于升级维护。

以前, FPGA 经常用做 ASIC 设计的原型或提供一个硬件平台来验证一个新算法的物理层实现。但是, FPGA 的低开发成本和短上市时间使越来越多的公司用它来制造最终产品,今天的 FPGA 器件中融合了微处理器、高速输入/输出 (I/O) 接口、时钟管理等丰富的硬 IP 模块,可以用来实现任何功能的电路,主要表现在 ASIC、数字信号处理应用 (DSP)、嵌入式微处理器应用、物理层通信芯片和可重配置计算技术方面。

- ASIC: 今天的 FPGA 正逐渐用来实现各种以前仅能由 ASIC 来完成的设计。
- 数字信号处理应用 (DSP): 高速 DSP 在传统上是由数字信号处理器来实现的。但是,今天的 FPGA 可以包含内嵌的乘法器、专用计算 IP 和大量的片上 RAM 等所有 DSP 操作所需的特性,这些特性再加上 FPGA 提供的并行性,其结果就是实现真正意义上的并行信号处理器,其速度是传统 DSP 芯片无法比拟的。
- 嵌入式微处理器: 小的控制功能在传统上是用微控制器的专用嵌入式微处理器处理的。这些低成本器件由围绕在处理器核外的片上程序和指令存储器、定时器和 I/O 组成。但是,随着 FPGA 价格的不断下降,甚至最小的器件都足以实现一个集成有可定制 I/O 功能的软处理器核,结果就是 FPGA 对嵌入式控制应用越来越具有吸引力。
- 物理层通信: FPGA 长期以来用于实现物理层通信芯片和网络协议层互连逻辑功能。今天的高端 FPGA 拥有多种高速收发器,这意味着通信和网络功能可以合并到同一个设备中。
- 可重配置计算技术: 这是指由 FPGA 提供的固有并行性和可重配置性来实现算法及功能模块的重构。

任何从事电子设计或电子设计自动化领域的人都知道,近年来这些领域所涉及的算法功能越来越复杂,导致其对 FPGA 实现的设计质量和可靠性要求越来越高。由于工作的原因,作者在过去四年中一直负责通信系统高质量、高可靠 FPGA 实现课题。四年来,查阅了大量的国内外文献、书籍和几乎全部的 Xilinx 设计文档,积累了丰富的高质量、可靠性设计经验和资料,这些经验都是文献学习、推理和大量实验的结果,并在实际工程中得到检验和推广应用。在研究过程中,有感于目前国内所出版的书籍大多侧重于语法及工具本身的介绍,对于可靠性设计应用及 FPGA 核心设计思想很少涉及,所以作者从实践出发并结合多年的开发

经验编写了本书。本书共分8章，各章的内容如下：

第1章介绍FPGA的基础知识，包括FPGA的发展、基本结构、FPGA的基本应用，还介绍了FPGA的设计开发流程和常用的开发工具。本书要求读者具有一定的FPGA设计基础，通过这一章的介绍，读者可以掌握FPGA的基础知识，并且能掌握FPGA的基本设计流程，为后续章节的学习做个铺垫。

第2章介绍FPGA设计思想，包括可综合设计思想、面积和速度互换思想、时钟设计思想、同步设计思想、延时电路设计思想、复位电路设计思想和抗干扰设计思想等，最后给出了高质量、可靠性设计的一般设计检查单。本章的内容并不针对特定的FPGA厂商或特殊FPGA器件，而是从硬件设计的原理及其对应的硬件描述、综合推断等层面进行分析说明。读者只有掌握这一章的内容，才能设计出性能高、运行稳定、资源利用率高的电路。

第3章介绍工程应用中常用的高速数据传输设计，主要包括高速通信现状及典型方案、高速数据传输中的同步技术、Xilinx FPGA对高速数据传输的支持、高速数据通信的加固设计等内容，并在此基础上给出了具体应用实例。

第4章针对Xilinx FPGA的配置管理，详细分析了其配置模式、常用的配置电路、配置过程，并针对高级应用，研究了配置命令、配置内存回读、配置内存重构（刷新）等高级配置管理。通过本章的学习，读者将对配置管理及应用有更深刻的理解。结合书中给出的实例，将会根据实际工程需要设计合适的配置管理方案。

第5章针对空间应用，介绍了空间环境对基于SRAM配置的FPGA电路的影响，并针对Xilinx FPGA的空间应用，给出了可靠性设计方法。

第6、7章分别针对大规模电路设计中的设计思想——分区设计和仿真验证进行了介绍。

第8章给出了一个综合应用实例——多输入、多输出数字波束形成系统的设计方案，并详细介绍了各具体模块的FPGA实现算法、仿真结果和实现代码。

本书中所涉及的源代码可以从华信教育资源网或<http://yydz.phei.com.cn>进行下载。

本书由汤琦博士主持编写。其中，第2、3、4、5章由汤琦编写，第1、6、7、8章及附录由蒋军敏编写，全书由汤琦统稿。

在准备本书的过程中，许多相关领域的专家、朋友提供了宝贵的帮助、建议和支持，他们是西安电子科技大学苏涛教授、武荣伟（华为工程师）、梁中英，中国空间技术研究院西安分院夏猛等，在此一并致以诚挚的谢意。

由于笔者水平有限，书中错误和疏漏之处在所难免，恳请广大读者批评指正。

编 者



目 录

第 1 章 概述	1
1.1 可编程逻辑器件与 PLD 开发工具	1
1.1.1 可编程逻辑器件	1
1.1.2 可编程逻辑器件的发展历史	1
1.1.3 PLD 开发工具	2
1.2 FPGA 工作原理与开发流程	3
1.3 可编程技术	5
1.3.1 SRAM 编程技术	5
1.3.2 Flash/E ² PROM 编程技术	6
1.3.3 反熔丝编程技术	6
1.3.4 编程技术比较	7
1.4 FPGA 芯片结构	7
1.4.1 可编程输入/输出单元 (IOB)	7
1.4.2 可配置逻辑块 (CLB)	8
1.4.3 数字时钟管理模块 (DCM)	9
1.4.4 嵌入块式 RAM (BRAM)	10
1.4.5 布线资源	10
1.4.6 内嵌功能单元	11
1.4.7 内嵌专用硬核	11
1.4.8 软核、硬核及固核的概念	11
1.5 Xilinx 公司 FPGA 介绍	12
1.5.1 Spartan 系列	12
1.5.2 Virtex 系列	17
第 2 章 FPGA 设计思想	22
2.1 可综合设计思想	22
2.1.1 VHDL 语言概述	22
2.1.2 设计层次	23
2.1.3 可综合描述规范	23
2.1.4 组合逻辑电路可综合设计	25
2.1.5 时序逻辑电路可综合设计	26
2.2 面积与速度互换思想	28
2.2.1 利用层次化设计控制设计结构	28
2.2.2 if 语句和 case 语句控制实现结构	30
2.2.3 减少关键路径的逻辑级数	31

2.2.4	流水线 Pipelining	32
2.2.5	串行转并行处理	34
2.2.6	组合逻辑和时序逻辑分离	35
2.3	时钟设计思想	38
2.3.1	工作时钟模型	38
2.3.2	全局时钟	39
2.3.3	门控时钟	39
2.3.4	多级逻辑时钟	42
2.3.5	行波时钟	44
2.3.6	多时钟系统	44
2.3.7	Xilinx FPGA 中的时钟资源	46
2.3.8	时序约束	54
2.4	同步设计思想	57
2.4.1	异步电路和同步电路	57
2.4.2	一般组合逻辑的同步设计	58
2.4.3	二次时钟的同步设计	59
2.4.4	多时钟系统的同步设计	59
2.4.5	非同源时钟同步化 (D 触发器使能信号的合理使用)	61
2.4.6	数据接口同步设计	62
2.5	延时电路设计思想	64
2.6	复位电路设计思想	66
2.6.1	同步复位	66
2.6.2	异步复位	68
2.6.3	触发器组模块的复位	70
2.6.4	复位电路的同步化方法	71
2.7	抗干扰设计思想	73
2.7.1	干扰产生的原因	73
2.7.2	干扰抑制设计方法	73
2.7.3	基于采样法的串口通信设计	75
2.8	可靠性设计检查单	79
第3章	高速数据传输设计	81
3.1	概述	81
3.1.1	高速数据通信的发展现状	81
3.1.2	几种高速数据通信方案简介	81
3.2	高速数据传输中的同步技术	83
3.2.1	同步方法及其特点	84
3.2.2	同步方式比较及对数据通信的影响	85
3.3	FPGA 对同步技术的支持	86
3.3.1	动态相位调整技术	86

3.3.2	基于 ChipSync 的动态相位调整方法	88
3.3.3	串行收发器 SERDES (ISERDES 和 OSERDES)	90
3.4	应用实例——基于 SERDES 的多路高速同步传输系统	96
3.4.1	系统方案	96
3.4.2	发送模块	97
3.4.3	接收模块	101
3.5	基于 RocketIO 的高速数据传输系统	113
3.5.1	自同步通信系统架构	113
3.5.2	RocketIO 简介	115
3.5.3	基于标准协议的可靠通信模型	116
3.5.4	应用实例——基于自定义协议的即时传输系统	118
3.6	高速数据通信的加固设计	121
3.6.1	数据通信加固的概念	121
3.6.2	交织汉明码原理及其性能分析	121
3.6.3	基于交织汉明码的高速通信加固设计	125
3.7	LVDS 应用设计	126
3.7.1	LVDS 简介	126
3.7.2	LVDS 系统设计	128
第 4 章	Xilinx FPGA 高级配置管理	130
4.1	配置模式	130
4.1.1	配置接口及配置模式	130
4.1.2	配置引脚	131
4.1.3	SelectMAP 接口	132
4.1.4	和配置相关的 BitGen 选项	133
4.2	配置电路	135
4.2.1	配置时的电源要求	135
4.2.2	常用配置存储器介绍	136
4.2.3	主串模式配置电路	136
4.2.4	主从模式配置电路	138
4.2.5	SelectMAP 模式配置电路	140
4.3	FPGA 配置过程	141
4.3.1	配置数据流加载过程	141
4.3.2	从串模式配置过程	143
4.3.3	SelectMAP 模式配置数据加载	144
4.3.4	延时加载	151
4.4	配置命令分析	151
4.4.1	配置帧	152
4.4.2	配置数据流类型	152
4.4.3	配置帧寻址方式	153

4.4.4	配置寄存器	155
4.4.5	配置命令执行过程分析	157
4.4.6	配置数据解析程序	160
4.5	配置内存回读	162
4.5.1	回读准备设计	163
4.5.2	回读指令序列	163
4.5.3	回读数据校验	166
4.6	配置内存重构 (刷新)	168
4.6.1	SelectMAP 模式下重构流程	168
4.6.2	刷新对系统功能的影响	169
4.7	SelectMAP 接口下重配置实现代码	177
4.8	配置数据文件格式分析	181
4.8.1	字节格式	181
4.8.2	文件格式	182
第 5 章	空间应用可靠性设计	188
5.1	集成电路单粒子效应的机理	188
5.2	单粒子翻转故障模式	189
5.3	SRAM 型 FPGA 单粒子问题的缓解措施	192
5.3.1	循环加电	192
5.3.2	配置管理	192
5.3.3	Xilinx 三模冗余	193
5.3.4	器件冗余	193
5.4	三模冗余设计方法介绍	193
5.4.1	三模冗余原理	194
5.4.2	TMR Tool 工具介绍	196
5.4.3	TMR Tool 设计流程	196
5.4.4	创建一个 ISE 工程完成三模冗余前的设计	197
5.4.5	创建一个 TMR Tool 工程产生三模冗余后的网表	197
5.4.6	创建第二个 ISE 工程完成三模冗余后的设计	201
5.4.7	三模冗余技术问题分析	201
5.5	Half-Latch 处理	207
5.6	异步 FIFO 处理	208
5.7	配置、刷新一体化方法	209
5.7.1	实现电路	209
5.7.2	工作流程及控制时序	210
5.7.3	SelectMAP 接口 Active 刷新实现	212
第 6 章	分区设计	219
6.1	为什么使用分区设计	219
6.2	分区与 SmartGuide	221



6.3	如何使用分区	222
6.4	用 Synplify 8.8.1 实现分区	223
6.5	分区保留级别	223
6.6	分区保留	224
6.7	对分区进行布局规划	225
6.8	删除分区	226
6.9	结论	226
第7章	高效验证 (TestBench) 设计	227
7.1	为什么要进行仿真验证	227
7.2	仿真验证程序设计	229
7.2.1	仿真的三个阶段	229
7.2.2	仿真的注意事项	229
7.2.3	仿真程序结构	229
7.3	使用 TestBench 对设计进行仿真	231
7.4	双向总线信号仿真	233
7.5	基于 TEXTIO 的交互式仿真	234
7.5.1	基于 TEXTIO 的测试程序	234
7.5.2	TEXTIO	235
7.5.3	基于 TEXTIO 的交互式仿真实例	236
7.6	几种常用的 FPGA 系数表文件产生方法	241
7.6.1	滤波器系数表	241
7.6.2	RAM 系数表	242
7.6.3	I/O 文件	243
第8章	综合实例——数字 DBF 系统	244
8.1	系统实现结构	244
8.2	数字下变频 (DDC)	244
8.3	数据传输模块	248
8.4	波束形成的实现	249
附录 A	类型转换	255
附录 B	文件操作	262
附录 C	常用元件的规范化设计示例	265
附录 D	FPGA 设计流程	279
参考文献	282

第 1 章 概 述

1.1 可编程逻辑器件与 PLD 开发工具

1.1.1 可编程逻辑器件

可编程逻辑器件 (Programmable Logic Device, PLD) 起源于 20 世纪 70 年代, 是在专用集成电路 (ASIC) 的基础上发展起来的一种新型逻辑器件, 是当今数字系统设计的主要硬件平台, 其主要特点就是完全由用户通过软件进行配置和编程, 从而完成某种特定的功能, 并且可以反复擦写。在修改和升级 PLD 时, 不需要额外地改变 PCB, 只在计算机上修改和更新程序, 使硬件设计工作成为软件开发工作, 缩短了系统设计的周期, 提高了实现的灵活性并降低了成本, 因此获得了广大硬件工程师的青睐, 形成了巨大的 PLD 产业规模。

目前常见的 PLD 产品有编程只读存储器 (Programmable Read Only Memory, PROM), 现场可编程逻辑阵列 (Field Programmable Logic Array, FPLA), 可编程阵列逻辑 (Programmable Array Logic, PAL), 通用阵列逻辑 (Generic Array Logic, GAL), 可擦除的可编程逻辑器件 (Erasable Programmable Logic Array, EPLA), 复杂可编程逻辑器件 (Complex Programmable Logic Device, CPLD) 和现场可编程门阵列 (Field Programmable Gate Array, FPGA) 等类型。PLD 器件从规模上又可以细分为简单 PLD (SPLD)、复杂 PLD (CPLD) 及 FPGA。它们内部结构的实现方法各不相同。

可编程逻辑器件按照颗粒度可以分为 3 类: 小颗粒度 (如“门海 (Sea of Gates)”架构)、中等颗粒度 (如 FPGA) 和大颗粒度 (如 CPLD)。按照编程工艺可以分为 4 类: 熔丝 (Fuse) 和反熔丝 (Antifuse) 编程器件、可擦除的可编程只读存储器 (UEPROM) 编程器件、电信号可擦除的可编程只读存储器 (E²PROM) 编程器件 (如 CPLD)、SRAM 编程器件 (如 FPGA)。在工艺分类中, 前 3 类为非易失性器件, 编程后配置数据保留在器件上; 第 4 类为易失性器件, 掉电后配置数据会丢失, 因此在每次上电后需要重新进行数据配置。

1.1.2 可编程逻辑器件的发展历史

可编程逻辑器件的发展可以划分为 4 个阶段, 即从 20 世纪 70 年代初到 70 年代中为第 1 阶段, 从 20 世纪 70 年代中到 80 年代中为第 2 阶段, 从 20 世纪 80 年代中到 90 年代末为第 3 阶段, 从 20 世纪 90 年代末到目前为第 4 阶段。

第 1 阶段的可编程逻辑器件只有简单的可编程只读存储器、紫外线可擦除只读存储器 (EPROM) 和电信号可擦除的可编程只读存储器 3 种。由于结构的限制, 它们只能完成简单的数字逻辑功能。

第2阶段出现了结构上稍微复杂的可编程阵列逻辑和通用阵列逻辑器件，正式被称为PLD，能够完成各种逻辑运算功能。典型的PLD由“与”、“非”阵列组成，用“与或”表达式来实现任意组合逻辑，所以PLD能以乘积和的形式完成大量的逻辑组合。

第3阶段Xilinx和Altera分别推出了与标准门阵列类似的FPGA和类似于PAL结构的扩展性CPLD，提高了逻辑运算的速度，具有体系结构和逻辑单元灵活、集成度高及适用范围宽等特点，兼容了PLD和通用门阵列的优点，能够实现超大规模的电路，编程方式也很灵活，成为产品原型设计和中、小规模（一般小于10000）产品生产的首选。这一阶段，CPLD、FPGA器件在制造工艺和产品性能方面都获得了长足的发展，达到了0.18工艺和门数超过百万门的规模。

第4阶段出现了SOPC和SOC技术，是PLD和ASIC技术融合的结果，涵盖了实时化数字信号处理技术、高速数据收发器、复杂计算及嵌入式系统设计技术的全部内容。Xilinx和Altera也推出了相应的SOCFPGA产品，制造工艺达到65nm，系统门数也超过百万门。并且，这一阶段的逻辑器件内嵌了硬核高速乘法器、Gbits差分串行接口、时钟频率高达500MHz的PowerPC微处理器、软核MicroBlaze、Picoblaze、Nios及NiosII，不仅实现了软件需求和硬件设计的完美结合，还实现了高速与灵活性的完美结合。这使其超越了ASIC器件的性能和规模，也超越了传统意义上FPGA的概念，使PLD的应用范围从单片扩展到系统级。目前，基于PLD片上可编程的概念仍在进一步向前发展。

1.1.3 PLD 开发工具

基于高复杂度PLD器件的开发，在很大程度上要依靠电子设计自动化（EDA）来完成。PLD的EDA工具以计算机软件为主，将典型的单元电路封装起来形成固定模块及标准的硬件开发语言（如HDL语言）供设计人员使用。设计人员考虑如何将可组装的软件库和软件包搭建出满足需求的功能模块甚至完整的系统。PLD开发软件需要自动地完成逻辑编译、化简、分割、综合及优化、布局布线、仿真，以及对于特定目标芯片的适配编译和编程下载等工作。典型的EDA工具中必须包含两个特殊的软件包，即综合器和适配器。综合器的功能就是将设计者在EDA平台上完成的针对某个系统项目的HDL、原理图或状态图形描述和给定的硬件系统组件进行编译、优化、转换和综合。

随着开发规模的级数性增长，必须减短PLD开发软件的编译时间并提高其编译性能，以及提供丰富的知识产权（IP）和资源供设计人员调用。此外，PLD开发界面的友好性及操作的复杂程度也是评价其性能的重要因素。目前在PLD产业领域中，各个芯片提供商的PLD开发工具已成为影响其成败的核心成分。Xilinx公司的ISE、Altera公司的Quartus II和Maxplus II是业界公认的优秀集成PLD开发软件。此外，综合软件Synplify和仿真软件ModelSim等诸多第三方EDK开发软件也满足上述要求。

目前，综合工具主要有DC（Design Compiler）、FC2（FPGA Compiler II）、Synplify Leonardo Galileo等。其中DC主要是用于ASIC的综合工具。FC2是DC在FPGA综合方面的增强版，FC2的开发队伍已解散。Leonardo是做FPGA综合工具的先驱Synplify Synplicity公司出品的，是目前成长最快的综合工具。但无论哪家综合工具，对FPGA的综合都必须紧密结合

各 FPGA 厂家的 FPGA 结构，否则无法更好利用 FPGA 提供的优点。从目前来看，对做 FPGA 优选 Synplify 或 Leonardo 综合工具，FPGA 综合性能差，但适宜 ASIC 综合。到目前为止，第三方的综合工具都未能很好地利用 FPGA 器件，如 Virtex 系列的进位链综合工具，不管代码风格如何变，都利用不上。因为综合工具一直在升级，因此无法得到更好的性能，但若用 Xilinx 自己的综合工具 XST，则可综合出来。从目前掌握的情况来看，Synplicity 公司的综合工具比较优秀。

1.2 FPGA 工作原理与开发流程

FPGA 是在 PAL、GAL、EPLD、CPLD 等可编程逻辑器件的基础上进一步发展起来的产物。它是作为 ASIC 领域中的一种半定制电路而出现的，即解决了定制电路的不足，又克服了原有可编程逻辑器件门电路有限的缺点。ASIC 和 FPGA 开发流程比较如图 1-1 所示。

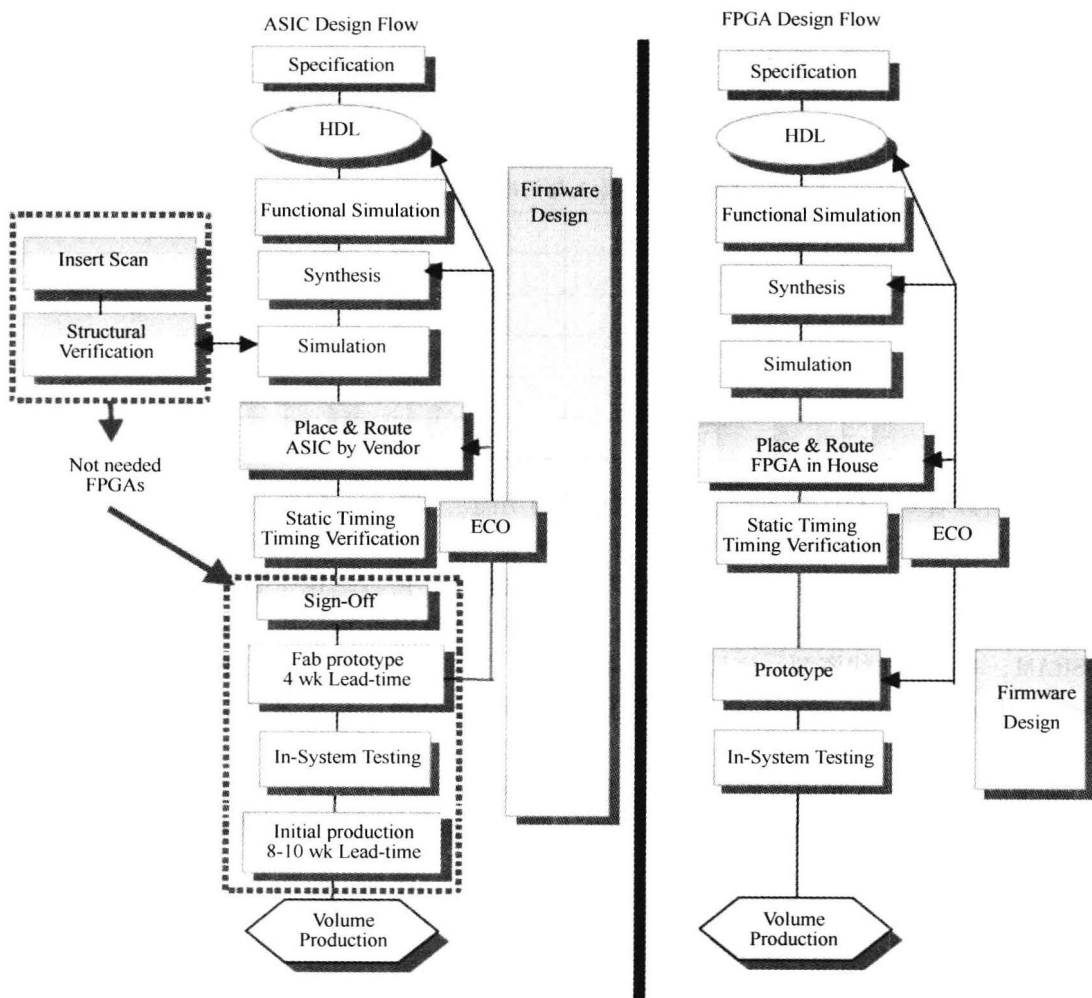


图 1-1 FPGA 和 ASIC 开发流程比较

由于 FPGA 可以被反复烧写，它实现组合逻辑的基本结构不可能像 ASIC 那样通过固定的与非门来完成，而只能采用一种易于反复配置的结构。查找表可以很好地满足这一要求，目前主流的 FPGA 都采用了基于 SRAM 工艺的查找表结构，也有一些军品和宇航级 FPGA 采用 Flash 或熔丝与反熔丝工艺的查找表结构，通过烧写文件改变查找表内容的方法来实现对 FPGA 的重复配置。

根据数字电路的基本知识可以知道，对于一个 n 输入的逻辑运算，不管是与或非运算还是异或运算等，最多只可能存在 $2n$ 种结果。所以，如果事先将相应的结果存放于一个存储单元，则相当于实现了与非门电路的功能。FPGA 的原理也是如此，它通过烧写文件去配置查找表的内容，从而在相同的电路情况下实现不同的逻辑功能。

查找表 (Look - Up - Table) 简称为 LUT，在本质上就是一个 RAM。目前 FPGA 中多使用 4 输入的 LUT，所以每一个 LUT 可以看成是一个有 4 位地址线的 RAM。当用户通过原理图或 HDL 语言描述一个逻辑电路以后，PLD/FPGA 开发软件会自动计算逻辑电路的所有可能结果，并把真值表 (即结果) 事先写入 RAM。这样，每输入一个信号进行逻辑运算就等于输入一个地址进行查表，找出地址对应的内容，然后输出即可。

下面给出一个 4 与门电路的例子来说明 LUT 实现逻辑功能的原理。

表 1-1 给出一个使用 LUT 实现 4 输入与门电路的真值表。

表 1-1 4 输入与门电路的真值表

实际逻辑电路		LUT 的实现方式	
a, b, c, d 输入	逻辑输出	RAM 地址	RAM 中存储的内容
0000	0	0000	0
0001	0	0001	0
⋮	⋮	⋮	⋮
1111	1	1111	1

从中可以看到，LUT 具有和逻辑电路相同的功能。实际上，LUT 具有更快的执行速度和更大的规模。由于基于 LUT 的 FPGA 具有很高的集成度，其器件密度从数万门到数千万门不等，可以完成极其复杂的时序与逻辑组合逻辑电路功能，所以适用于高速、高密度的高端数字逻辑电路设计领域。其组成部分主要有可编程输入/输出单元、基本可编程逻辑单元、内嵌 SRAM、丰富的布线资源、底层嵌入功能单元、内嵌专用单元等，主要设计和生产厂家有 Xilinx、Altera、Lattice、Actel、Atmel 和 QuickLogic 等公司，其中最大的是 Xilinx、Altera、Lattice 三家。

如前所述，FPGA 是由存放在片内的 RAM 来设置其工作状态的，因此工作时需要对片内 RAM 进行编程。用户可根据不同的配置模式，采用不同的编程方式。FPGA 有如下几种配置模式。

- (1) 并行模式：并行 PROM、Flash 配置 FPGA。
- (2) 主从模式：一片 PROM 配置多片 FPGA。
- (3) 串行模式：串行 PROM 配置 FPGA。
- (4) 外设模式：将 FPGA 作为微处理器的外设，由微处理器对其编程。

目前, FPGA 市场占有率最高的两大公司 Xilinx 和 Altera 生产的 FPGA 都是基于 SRAM 工艺的, 在使用时需要外接一个片外存储器以保存程序。上电时, FPGA 将外部存储器中的数据读入片内 RAM, 完成配置后进入工作状态; 掉电后, FPGA 恢复为白片, 内部逻辑消失。这样, FPGA 不仅能反复使用, 还无需专门的 FPGA 编程器, 只需通用的 EPROM、PROM 编程器即可。Actel、QuickLogic 等公司还提供反熔丝技术的 FPGA, 只能下载一次, 具有抗辐射、耐高低温、低功耗和速度快等优点, 在军品和航空航天领域中应用较多, 但这种 FPGA 不能重复擦写, 开发初期比较麻烦, 费用也比较昂贵。Lattice 是 ISP 技术的发明者, 在小规模 PLD 应用上有一定的特色。早期的 Xilinx 产品一般不涉及军品和宇航级市场, 但目前已经有 Q Pro - R 等多款产品进入该类领域。

1.3 可编程技术

可编程技术是 FPGA 的核心, 采用不同类型的存储器实现可编程功能对 FPGA 器件的结构和性能有着巨大的影响。SRAM 使用标准 CMOS 工艺设计加工, 在 FPGA 中应用最广泛; Flash/E²PROM 可编程技术和反熔丝结构具有独特的优点。本节将分别介绍这几种可编程技术, 分析其优劣势。

1.3.1 SRAM 编程技术

最基本的 5 管 SRAM 单元结构如图 1-2 所示, 通过传输管控制存储信息的读写。当传输管导通时, SRAM 单元内存储的信息可由数据端读取或改写; 当传输管截止时, 存储的信息被首尾相连的两个反相器锁定, 由 Q 和 Q' 端输出。理论上, SARM 单元可被配置无数次。

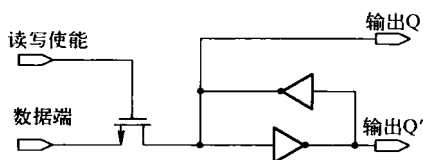


图 1-2 5 管 SRAM 单元结构

FPGA 中的 SRAM 单元主要实现以下功能: 作为多路开关、交叉开关、互连通道等可编程结构的控制端, 对信号的传输路径进行编程; 对片内相对独立的逻辑功能模块 (如可编程触发器、用户可编程 I/O) 进行配置; 作为查找表 (LookUp - Table, LUT) 的存储单元, 用来实现 FPGA 的逻辑功能; 使用 SRAM 存储阵列作为嵌入式存储器, 实现复杂的数字信号处理和存储功能。但是, 基于 SRAM 单元的可编程技术存在一系列需要解决的问题:

- 断电后, SRAM 存储的信息将全部丢失, 所以通常使用外部非易失性存储器来存储相关配置信息, 如 Flash 或 E²PROM 等。
- 上电复位后, 配置信息需从外部存储器写入器件, 配置信息有可能被窃取。
- SRAM 单元用来控制传输管的导通或截止, 对 FPGA 内的信号传输路径进行编程, 传输管并不是一种理想开关元件, 对信号传输会带来相当大的阻性和容性负载, 降低信

号完整性。

1.3.2 Flash/E²PROM 编程技术

Flash 和 E²PROM 存储结构都具有非易失性的特点，即使关闭电源，内部的存储信息也不会丢失。Flash 存储单元取消了 E²PROM 隧道型存储单元的选择管，结构更简单有效，可通过一个信号一次性擦除一个区域的存储信息，集成密度更高。基于 Flash 存储结构的可编程开关电路如图 1-3 所示，配置使晶体管和开关晶体管的浮栅（Floating Gate）连接在一起，通过控制配置晶体管的栅极和源/漏极之间的电压，向其浮栅注入电荷，就可改变开关晶体管导通或截止状态。基于浮栅结构的开关晶体管截面图如图 1-4 所示。

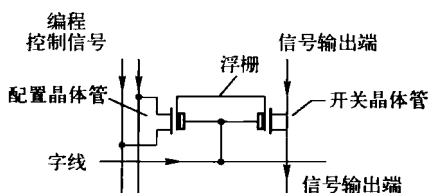


图 1-3 Flash 交叉开关原理示意图

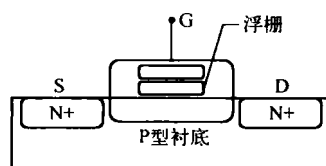


图 1-4 基于浮栅结构的开关晶体管截面图

与 SRAM 相比，Flash 结合了非易失性和可重复编程的特点，上电后无须配置。用 Flash 结构替代 FPGA 中的 5 管或 6 管 SRAM 存储单元，可大大减少晶体管数量，降低静态功耗，整个器件的静态电流可低至微安量级；而基于 SRAM 存储结构的主流商用 FPGA 产品，静态电流普遍在毫安量级。但是，基于 Flash 结构的 FPGA 具也有其自身局限性，表现在：

- ▶ Flash 存储单元擦写寿命是有限的，如 Actel 公司的 PorASIC3 系列产品，只能编程 500 次左右，这个次数对于大多数 FPGA 开发应用来说是远远不够的。
- ▶ Flash 结构需要特殊的半导体工艺，无法在第一时间应用最新工艺技术，器件规模和密度也远低于基于 SARM 可编程技术的 FPGA。目前基于 Flash 结构的最大规模的商用 FPGA 器件只有约 300 万系统门容量，而最新基于 SRAM 存储单元的产品容量可达数千万系统门。
- ▶ 同 SRAM 技术一样，Flash 技术也具有传输管带来的信号完整性问题。

1.3.3 反熔丝编程技术

反熔丝结构在编程之前通常是开路的，通过编程使反熔丝结构局部小区域内具有相当高的电流密度，瞬间产生巨大的热功耗，将薄绝缘层介质熔化形成永久性通路。反熔丝结构有两种，一种是多晶—扩散反熔丝，具有氧—氮—氧（Oxide—Nitride—Oxide, ONO）的介质夹层，简称 ONO 反熔丝；另一种是金属—金属（Metal—to—Metal, M2M）反熔丝，简称 M2M 反熔丝。二者相比，M2M 技术采用无源结构，具有更低编程电压和更小的电阻，是目前主流反熔丝工艺。采用反熔丝结构的 FPGA 具有非易失性，面积小，信号传输路径具有较小寄生电阻和电容，可上电后直接使用，信息安全性高等优点。由于不能重复编程，没有系统级相关配置电路，所以相对其他两种编程方式，开发成本更低。但是，反熔丝技术的局限

性非常明显，具体表现在以下几个方面：

- 无法重复编程，不适用新产品开发。
- 一次性编程不利于器件可靠性检测，编程后器件一次成功率低于另外两种技术。
- 在不同工艺下，反熔丝材料的电性能具有相当大的差异，在最新工艺节点下实现反熔丝结构非常困难。

采用反熔丝结构的 FPGA 在工艺上往往要落后于最新的工艺节点，规模和密度也低于采用 SRAM 可编程技术的 FPGA。目前最大规模的商用反熔丝 FPGA 产品只有约 400 万系统门容量。

1.3.4 编程技术比较

表 1-2 总结了 5 种 FPGA 编程技术的比较。从表中可以看出，使用反熔丝技术和 Flash 技术可以获得最佳的面积和速度性能；使用 EPROM 和 E²PROM 技术也可以获得较好的面积，但是性能较差；使用 SRAM 技术需要占用最大的芯片面积。

表 1-2 5 种编程技术比较

编程技术	重复编程	易失存储	单元面积	工 艺
反熔丝	否	否	1x	改进 CMOS
SRAM	是	是	5x	标准 CMOS
EPROM	是	否	1x	标准 EPROM
E ² PROM	是	否	2x	标准 E ² PROM
Flash	是	否	0.8x	标准 Flash

1.4 FPGA 芯片结构

目前主流的 FPGA 仍是基于查找表技术的，已经远远超出了先前版本的基本性能，并且整合了常用功能（如 RAM、时钟管理和 DSP）的硬核（ASIC 型）模块。如图 1-5 所示（注意，图 1-5 只是一个示意图，实际上每一个系列的 FPGA 都有其相应的内部结构），FPGA 芯片主要包括 7 部分，分别为可编程输入/输出单元、基本可编程逻辑单元、完整的时钟管理、嵌入块式 RAM、丰富的布线资源、底层内嵌功能单元和内嵌专用硬核。

1.4.1 可编程输入/输出单元（IOB）

可编程输入/输出单元简称 I/O 单元，是芯片与外界电路的接口部分，完成不同电气特性下对输入/输出信号的驱动与匹配要求，其示意结构如图 1-6 所示。FPGA 内的 I/O 按组分类，每组都能够独立地支持不同的 I/O 标准。通过软件的灵活配置，可适配不同的电气标准与 I/O 物理特性，可以调整驱动电流的大小，可以改变上、下拉电阻。目前，I/O 接口的频率越来越高，一些高端的 FPGA 通过 DDR 寄存器技术可以支持高达 2Gbps 的数据速率。