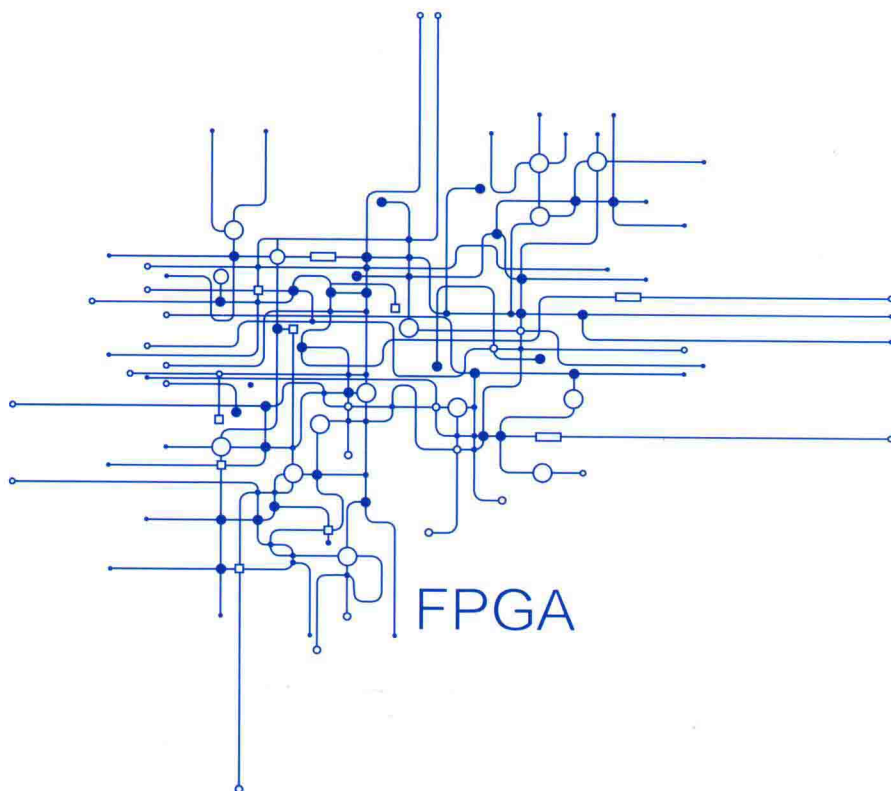


本书由浅入深，18个工程案例，全面阐释了FPGA的设计思路和开发技巧、解析Verilog和NIOS II处理器的C源码、图文并茂展示了必要板级装配和具体调试。帮助读者快速搭建起自己的项目原型。



例说FPGA

可直接用于工程项目的 第一手经验

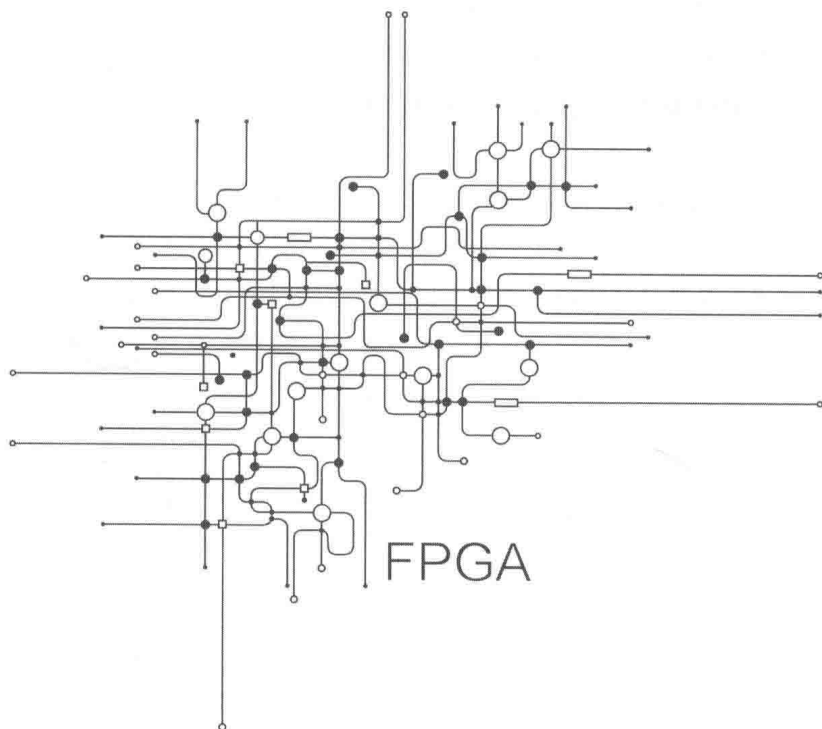
吴厚航 著



机械工业出版社
China Machine Press



电子与嵌入式系统
设计丛书



例说FPGA

可直接用于工程项目的
第一手经验

吴厚航 著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

例说 FPGA：可直接用于工程项目的 firsthand 经验 / 吴厚航著. —北京：机械工业出版社，2016.5

(电子与嵌入式系统设计丛书)

ISBN 978-7-111-53822-6

I. 例… II. 吴… III. 可程序逻辑器件—系统设计 IV. TP332.1

中国版本图书馆 CIP 数据核字 (2016) 第 104471 号

例说 FPGA：可直接用于工程项目的 firsthand 经验

出版发行：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：陈佳媛

责任校对：殷虹

印 刷：三河市宏图印务有限公司

版 次：2016 年 6 月第 1 版第 1 次印刷

开 本：186mm×240mm 1/16

印 张：23.25

书 号：ISBN 978-7-111-53822-6

定 价：69.00 元

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) 88379426 88361066

投稿热线：(010) 88379604

购书热线：(010) 68326294 88379649 68995259

读者信箱：hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问：北京大成律师事务所 韩光 / 邹晓东

前言

从亚微米到深亚微米，再到如今成熟的纳米级工艺，一不留神，双核、多核延续了摩尔定律的神话。技术的发展日新月异，身处技术研发一线的工程师们，稍不留神，或许就要错过许多新奇好玩的新技术。今年的硬蛋智能硬件展上，FPGA 厂商赛灵思的展区前人头攒动，老鼠迷宫、魔方机器人、爬墙机器人、智能书法器……各种新奇玩意层出不穷，简直让人过目不忘。再不动手玩起来，咱们可真要落伍啦！

话说 FPGA 可真是个好东西——速度快、资源多、灵活，在传统工控、机器视觉、数据中心等热门领域都能够独当一面。因此，一定要有一个属于自己的 FPGA 硬件平台，DIY 不再是个梦想。

为了协助大家更好地玩转 FPGA，我已经使出浑身解数，将各种自认为新奇好玩且十分“时髦前卫”的 FPGA 项目“跃然纸上”。没错，这本书可以让大家熟悉这些基于 FPGA 的工程项目的实现细节。更重要的是，我为这些项目量身定制了一整套“摸得着且看得见”的实实在在的 FPGA 硬件平台。不需要我说太多，书籍（文字）、板子、工程（代码）……真正的玩家已经怦然心动了。

至此不要再犹豫了，继续往下翻，开始你的 FPGA DIY 之旅吧。从基本的知识，到硬件平台的搭建，再到真正的项目实现，你可以尽情“玩”，也可以好好“学”，甚至可以动真格地把工程“产品化”。无论如何，这本书要带给你的，一定是你真正需要的。

吴厚航（网名：特权同学）

2015 年 12 月于上海

目 录

前言

第一部分 基本知识

第 1 章 FPGA 开发概述 2

- 1.1 FPGA 基础入门 2
- 1.2 FPGA 的优势与局限性 6
- 1.3 FPGA 开发所需的技能 7
- 1.4 FPGA 应用领域 8
- 1.5 FPGA 开发流程 9

第 2 章 FPGA 板级电路设计 11

- 2.1 FPGA 板级电路设计五要素 11
- 2.2 FPGA 核心电路板设计 17
- 2.3 扩展外设子板设计 34

第二部分 工程实例

第 3 章 工程实例 1——PLL 例化 配置与 LED 闪烁 50

- 3.1 功能概述 50
- 3.2 新建 Quartus II 工程 50
- 3.3 IP 核配置——PLL 54
- 3.4 引脚分配 60
- 3.5 闲置引脚设置 64

3.6 Verilog 代码解析 65

3.7 板级调试 72

第 4 章 工程实例 2——DDR2 控制 器集成与读写测试 73

- 4.1 功能概述 73
- 4.2 IP 核配置——片内 RAM 74
- 4.3 IP 核配置——DDR2 控制器 79
- 4.4 DDR2 引脚电平设置 87
- 4.5 Verilog 代码解析 90
- 4.6 板级调试 100

第 5 章 工程实例 3——Qsys 系统 搭建与软件开发 102

- 5.1 功能概述 102
- 5.2 Qsys 系统搭建 103
- 5.3 Verilog 代码解析 117
- 5.4 软件工程——创建与编译 119
- 5.5 软件工程——导入已有工程 125
- 5.6 软件工程——运行调试 126
- 5.7 板级调试 128

第 6 章 工程实例 4——NAND Flash 读写测试 130

- 6.1 功能概述 130
- 6.2 IP 核配置——自定义 Qsys
组件 131

6.3 Verilog 代码解析	146	10.6 板级调试	214
6.4 软件程序解析	160		
6.5 板级调试	162		
第 7 章 工程实例 5——多分辨率 VGA 显示驱动	164	第 11 章 工程实例 9——200 万 像素 CMOS 摄像头采 集显示	221
7.1 功能概述	164	11.1 功能概述	221
7.2 装配说明	166	11.2 装配说明	223
7.3 复用引脚设置	167	11.3 IP 核配置——ROM 配置与 初始化	223
7.4 Verilog 代码解析	169	11.4 IP 核配置——FIFO 配置	228
7.5 板级调试	174	11.5 Verilog 代码解析	231
		11.6 板级调试	237
第 8 章 工程实例 6——7 寸 800 × 480 LCD 显示驱动	175	第 12 章 工程实例 10——AV (PAL/NTSC) 视频 采集显示	238
8.1 功能概述	175	12.1 功能概述	238
8.2 装配说明	176	12.2 装配说明	239
8.3 Verilog 代码解析	176	12.3 IP 核配置——In-System Sources and Probes Editor 配置	239
8.4 板级调试	177	12.4 IP 核配置——FIFO 配置	242
第 9 章 工程实例 7——LCD 触摸屏驱动设计	179	12.5 IP 核配置——移位寄存器 配置	244
9.1 功能概述	179	12.6 IP 核配置——乘法器配置	248
9.2 装配说明	181	12.7 Verilog 代码解析	254
9.3 Verilog 代码解析	181	12.8 板级调试	272
9.4 软件程序解析	183		
9.5 板级调试	185	第 13 章 工程实例 11——FX2 硬件和驱动安装以及 Bulkloop 实验	275
第 10 章 工程实例 8——电子 点菜单设计	187	13.1 功能概述	275
10.1 功能概述	187	13.2 装配说明	276
10.2 装配说明	189	13.3 FX2 驱动安装说明	276
10.3 IP 核配置——FIFO 配置	189	13.4 板级调试	278
10.4 Verilog 代码解析	201		
10.5 软件程序解析	206		

第 14 章 工程实例 12——基于 FX2 和 FPGA 的 Loopback/StreamIN/StreamOUT 实例 281

- 14.1 功能概述 281
- 14.2 装配说明 282
- 14.3 IP 核配置——FIFO 配置 284
- 14.4 IP 核配置——RAM 配置 286
- 14.5 IP 核配置——SignalTap II 配置 288
- 14.6 Verilog 代码解析——fx2_loopback 工程 292
- 14.7 Verilog 代码解析——fx2_StreamIN 工程 295
- 14.8 Verilog 代码解析——fx2_StreamOUT 工程 297
- 14.9 板级调试 298

第 15 章 工程实例 13——基于 VGA 显示器的 720p 的广告机设计 304

- 15.1 功能概述 304
- 15.2 装配说明 305
- 15.3 Verilog 代码解析 305
- 15.4 软件程序解析 306
- 15.5 板级调试 310

第 16 章 工程实例 14——TXT 文本阅读器设计 313

- 16.1 功能概述 313
- 16.2 装配说明 314
- 16.3 GB2312 中文字符集简介 314
- 16.4 Verilog 代码解析 317
- 16.5 软件程序解析 318
- 16.6 板级调试 325

第 17 章 工程实例 15——工业现场实时监控界面设计 329

- 17.1 功能概述 329
- 17.2 装配说明 330
- 17.3 IP 核配置——双口 RAM 330
- 17.4 Verilog 代码解析 335
- 17.5 软件程序解析 339
- 17.6 板级调试 341

第 18 章 工程实例 16——多分辨率 HDMI 显示驱动设计 344

- 18.1 功能概述 344
- 18.2 装配说明 346
- 18.3 Verilog 代码解析 347
- 18.4 软件程序解析 350
- 18.5 板级调试 352

第 19 章 工程实例 17——基于 HDMI 的 AV 采集显示 353

- 19.1 功能概述 353
- 19.2 装配说明 354
- 19.3 Verilog 代码解析 354
- 19.4 软件程序解析 358
- 19.5 板级调试 358

第 20 章 工程实例 18——双摄像头图像采集 360

- 20.1 功能概述 360
- 20.2 装配说明 361
- 20.3 Verilog 代码解析 362
- 20.4 板级调试 365

第一部分

基本知识

第 1 章 FPGA 开发概述

第 2 章 FPGA 板级电路设计

第 1 章

FPGA 开发概述

本章导读

本章从 FPGA 的一些基本概念入手，将 ASIC、ASSP、ARM、DSP 与 FPGA 比对，同时也论及 FPGA 开发语言及主要厂商；接着对 FPGA 技术在嵌入式应用中的优势和局限性进行讨论；FPGA 是一项综合性很强的技术门类，因此我们在本章也会关注它所涉及的一些基本技能；最后我们将论述 FPGA 的应用领域和开发流程。当然了，如果你自认为已经入门 FPGA，对于本章可能会提及的一些基础概念耳熟能详，那么不妨跳过本章。

1.1 FPGA 基础入门

2015 年伊始，Intel 欲出资百亿美金收购 Altera 的新闻在半导体业内引发一场不小的轰动。而时隔不到半年，这则新闻居然成为现实，并且最终的收购价格 167 亿美金也比传闻要高出许多（如图 1-1 所示）。在早些年里，其实 Intel 与 Altera 已经或多或少有些合作了。Intel 此举的动机非常明显，不仅是希望在 PC 领域继续拓展服务器市场与数据中心的应用——Altera 的可编程器件在这方面能够提供强劲的性能竞争力；而且也是期望借助 Altera 来扩宽自身的多元化发展战略——Altera 是目前唯一一家毛利率好于 Intel 的半导体公司，并且其三分之二的收入来自于电信、无线、军事和太空等领域。无论如何，Altera 能够以其单一的 FPGA 产品线“傍上”Intel，或多或少都证明了 FPGA 存在的价值。

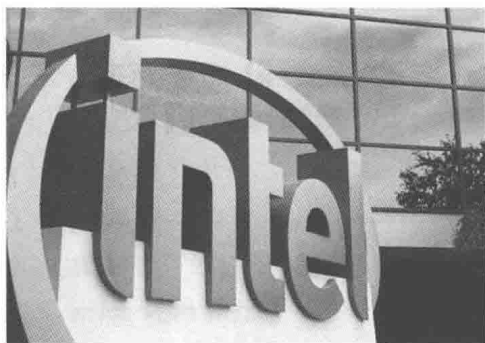


图 1-1 Intel 与 Altera 的联姻

要说 Intel，谁人不晓，基本上家家户户都或多或少有一些带着 Intel 烙印的家当；而反观 Altera，别人可以不知道，但是你要是不知道就要遭受各种鄙视了。身处电子行业，没有玩过 FPGA 不稀奇，但是没有听说过 FPGA 就有些难为情了。FPGA？Altera？什么情况？好吧，就当你真的什么都不知道吧，下面先开始我们的“扫盲行动”。入过门的别嫌烦，直接略过本节就好。

1. FPGA 是什么

简单来说，FPGA 就是“可反复编程的逻辑器件”。如图 1-2 所示，这是一个 Altera 公司最新的 Cyclone V Soc FPGA 器件，从外观上看，似乎和一般的 CPU 芯片没有太大差别。

FPGA 取自 Field Programmable Gate Array 这四个英文单词的首个字母，译为“现场 (Field) 可编程 (Programmable) 逻辑阵列 (Gate Array)”。至于 FPGA 到底是什么，能够干什么，又有什么过人之处？下面我们就把它和它的“师兄师弟”摆在一起，一一呈现这些问题的答案。

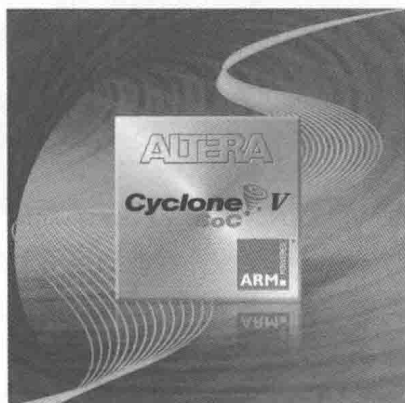


图 1-2 Altera 公司最新的 Cyclone V Soc FPGA 器件

2. FPGA、ASIC 和 ASSP

抛开 FPGA 不提，大家一定都很熟悉 ASIC 与 ASSP。所谓 ASIC，即专用集成电路 (Application Specific Integrated Circuit) 的简称；而 ASSP，即专用标准产品 (Application Specific Standard Parts) 的简称。电子产品中，它们无所不在，还真是比 FPGA 普及得多得多。但是 ASIC 以及 ASSP 的功能相对固定，它是为了专一功能或专一应用领域而生，希望对它进行任何的功能和性能的改善往往是无济于事的。打个浅显的比喻，如图 1-3 所示，如果说 ASIC 或 ASSP 是布满铅字的印刷品，那么 FPGA 就是可以自由发挥的白纸一张。



图 1-3 ASIC/ASSP 和 FPGA 就如同印刷品和白纸

使用了 FPGA 器件的电子产品，在产品发布后仍然可以对产品设计做出修改，大大方便了产品的更新以及针对新的协议标准做出的相应改进，从而可以加速产品的上市时间，并降低产品的失败风险和维护成本。相对于无法对售后产品设计进行修改的 ASIC 和 ASSP 来说，这是 FPGA 特有的一个优势。由于 FPGA 可编程的灵活性以及近年来电子技术领域的快速发展，FPGA 也正在向高集成、高性能、低功耗、低价格的方向发展，并且逐渐具备了与 ASIC 和 ASSP 相当的性能，使其被广泛地应用在各行各业的电子及通信设备中。

3. FPGA、ARM 和 DSP

与 ASIC 相比，FPGA、ARM 和 DSP 都具备与生俱来的可编程特性。或许身处开发第一线的底层工程师要说 No 了，很多 ASIC 不是也开放了一些可配置选项，实现“可编程”特性吗？是的，但与 FPGA、ARM、DSP 能够“为所欲为”的任意操控一整个系统而言，ASIC 的那点“可编程”性的确摆不上台面。当然，换个角度来看，FPGA、ARM 和 DSP 都或多或少集成了一些 ASIC 功能，正是这些 ASIC 功能，加上“可编程”特性，使得它们相互区别开了，并且各自独霸一方。

ARM (Advanced RISC Machines) 是微处理器行业的一家知名企业，设计了大量高性能、廉价、耗能低的 RISC 处理器、相关技术及软件。由 ARM 公司设计的处理器风靡全球，大有嵌入式系统无处不 ARM 的趋势。因此，我们通常所说的 ARM，更多的是指 ARM 公司的处理器，即 ARM 处理器，如图 1-4 所示。ARM 通常包含一颗强大的处理器内核，并且为这颗处理器量身配套了很多成熟的软件工具以及高级编程语言，这也是它备受青睐的原因之一。当然了，ARM 不只是一颗处理器而已，因为在 ARM 内核处理器周边，各种各样精于控制的外设比比皆是，例如 GPIO、PWM、AD/DA、UART、SPI、IIC……ARM 的长处是控制和管理，在很多工业自动化中大有用武之地。

DSP (digital signal processor)，如图 1-5 所示，即数字信号处理器，是一种独特的微处理器，有自己的完整指令系统，能够进行高速、高吞吐量的数字信号处理。它不像 ARM 那么“胡里花俏”，它更“专”。它只专注一件事，就是对各种语音、数据和视频做运算处理；或者也可以这么说，DSP 是为各种数学运算量身打造的。



图 1-4 ARM 处理器

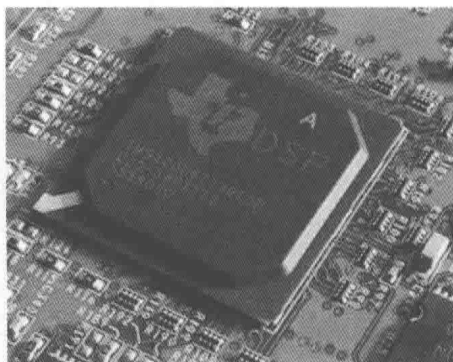


图 1-5 DSP

相比之下，套用近些年一句业内很火的广告词“All Programmable”来形容 FPGA 再合适不过了。ARM 虽然有很多外设，DSP 虽然具备强大的信号运算能力，但在 FPGA 眼里，这些都只不过是“小菜一碟”。或许说得有些过了，但是，毫不夸张地讲，ARM 能做的，DSP 能做的，FPGA 一定也都能做；而 FPGA 可以做的，ARM 不一定行，DSP 也不一定行。在很多原型产品设计过程中，时不时有人会提出基于 FPGA 的方案。在一些灵活性要求高、定制化程度高、性能要求也特别高的场合，FPGA（如图 1-6 所示）再合适不过了，甚至有时会是设计者别无选择的选择；当然了，客观来看，FPGA 固然强大，它高高在上的成本、功耗和开发复杂性还是会让很多潜在的目标产品望而却步，而在这些方面，ARM 和 DSP 正好弥补了 FPGA 所带来的缺憾。

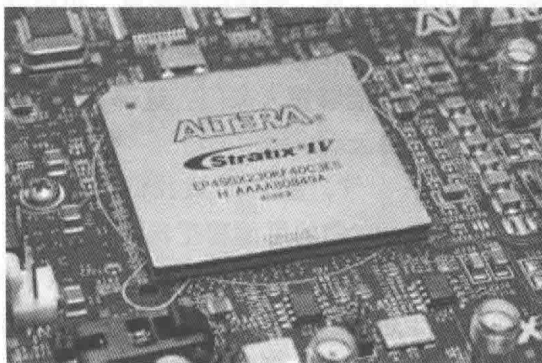


图 1-6 FPGA 器件

总而言之，在嵌入式系统设计领域，FPGA、ARM 和 DSP 互有优劣，各有所长。很多时候它们所实现的功能无法简单的相互替代，否则我们就不会见到如 TI 的达芬奇系列 ARM 中有 DSP、Xilinx 的 Zynq 或 Altera 的 Soc FPGA 中有 ARM 的共生现象了。FPGA、ARM 和 DSP，它们将在未来很长的一段时间内呈现三足鼎立的局面。

4. Verilog 与 VHDL

说到 FPGA，我们一定关心它的开发方式。FPGA 开发本质上就是一些逻辑电路的实现而已，因此早期的 FPGA 开发通过绘制原理图（和现在的硬件工程师绘制原理图的方式大体相仿）完成。而随着 FPGA 规模和复杂性的不断攀升，这种落后的设计方式几乎已经被大家遗忘了，取而代之的是能够实现更好的编辑性和可移植性的代码输入方式。

说到 FPGA 的设计代码方式，经过近三十年的发展，只有 Verilog 和 VHDL 二者最终脱颖而出，成为了公认的行业标准。对于这两种不同的语法，它们的历史渊源、孰优孰劣这里就不提了。美国和中国台湾地区的逻辑设计公司大都以 Verilog 语言为主，国内目前学习和使用 Verilog 的人数也在逐渐超过 VHDL。从学习的角度来讲，Verilog 相对 VHDL 有着快速上手、易于使用的特点，博得了更多工程师的青睐。即便是从来没有接触过 Verilog 的初学者，只要凭着一点 C 语言的底子加上一些硬件基础，三两个月很快就可以熟悉 Verilog 语法。当然了，仅仅是入门还是远远不够的，真正掌握 Verilog 必须花费很多时间和精力，并通过一些项目的实践，才会慢慢对可编程逻辑器件的设计有更深入的理解和认识。

5. Altera、Xilinx 和 Lattice

相比于互联网的那些“暴发户”，半导体行业则更讲究历史底蕴，“今天丑小鸭，明天白天鹅”的故事要少得多，因此两家历史最为久远的 FPGA 供应商 Altera 和 Xilinx 凭着一直以来的专注，确保了它们在这个行业的统治地位。当然了，很大程度上也是由于 FPGA 技术相

对于一般的半导体产品有着更高的门槛，从器件本身到一系列配套的工具链，再到终端客户的技术支持——这一箩筐的麻烦事，想想就头疼。

不过，提醒大家注意的是，开篇我们提到的 Intel 大手笔新闻，无论是不是空穴来风，最终 Altera 是否能够嫁入豪门，这都在预示着一个新的趋势——FPGA 技术将迎来它的又一个繁茂复活的春天。

唠叨一大气，赶紧回到我们的主题上来。目前 FPGA 器件的主流厂商 Altera 公司和 Xilinx 公司，它们的可编程逻辑器件占到了全球市场的 60% 以上。从明面上的“竞争对手”到今天暗地里还客气地互称“友商”，不难看出两家公司走过历史长河，虽然有“明争暗斗”，但确也不经意间的彼此促进，互相激励。的确，翻看 FPGA 的发展史，往往充斥着这两家公司不断上演的“你方唱罢我登场”的情节，并且偶尔也会有第三者如 Lattice 小弟的“插足”戏份。不过这好在新千年过后，各方重新定位，Altera 和 Xilinx 便牢牢把持住象征统治地位的中高端市场，而 Lattice 也只能在低端市场找找“山中无老虎，猴子称大王”的感觉了。

不论是 Altera、Xilinx 还是 Lattice，甚至一些后来者，如笔者接触过的国内 FPGA 厂商京微雅格，它们的 FPGA 器件内部结构虽然略有差异，但在开发流程、开发工具乃至原厂提供的各种支持上，都是“换汤不换药”的。所以，这对于用户而言，绝对是一个福音，只要好好掌握一套方法论，任何厂商的器件都可以通吃。

1.2 FPGA 的优势与局限性

若要准确评估 FPGA 技术能否满足开发产品的功能、性能以及其他各方面的需求，深入理解 FPGA 技术是至关重要的。在产品的整个生命周期中，如果产品功能可能会有较大的升级或变更，那么使用 FPGA 技术来实现就会有很大的优势。

在考虑是否使用 FPGA 技术来实现目标产品时，我们需要重点从以下几个方面进行评估。

- ❑ 可升级性——产生在设计过程中，甚至将来产品发布后，是否有较大的功能升级需求？是否应该选择具有易于更换的同等级、不同规模的 FPGA 器件？
- ❑ 开发周期——产品开发周期是否非常紧迫？若使用 FPGA 开发，是否比其他方案具有更高的开发难度，能否面对必须在最短的时间内开发出产品的挑战？
- ❑ 产品性能——产品的数据速率、吞吐量或处理能力是否有特殊要求？是否应该选择性能更好或速度等级更快的 FPGA 器件？
- ❑ 实现成本——是否有基于其他 ASIC、ARM 或 DSP 的方案，能够以更低的成本实现设计？FPGA 开发所需的工具、技术支持、培训等额外的成本有哪些？通过开发可复用的设计，是否可以将开发成本分摊到多个项目中？是否有已经实现的参考设计或者 IP 核可供使用？
- ❑ 可用性——器件的性能和尺寸的实现，是否可以赶上量产？是否有固定功能的器件可以代替？在产品及其衍生品的开发过程中，是否实现了固定功能？

- 其他限制因素——产品是否要求低功耗设计？电路板面积是否大大受限？工程实现中是否还有其他的特殊限制？

基于以上的这些考虑因素，我们可以从如下三大方面总结出在产品的开发或产品的生命周期中，使用 FPGA 技术实现所能够带来的潜在优势。

□ 灵活性

可重编程，可定制。

易于维护，方便移植、升级或扩展。

降低 NRE 成本，加速产品上市时间。

支持丰富的外设接口，可根据需求配置。

□ 并行性

更快的速度、更高的带宽。

满足实时处理的要求。

□ 集成性

更多的接口和协议支持。

可将各种端接匹配元件整合到器件内部，有效降低 BOM 成本。

单片解决方案，可以替代很多数字芯片。

减少板级走线，有效降低布局布线难度。

当然了，在很多情况下，FPGA 不是万能的。FPGA 技术也存在着一些固有的局限性。从以下这些方面看，选择 FPGA 技术来实现产品的开发设计有时并不是明智的决定。

- 在某些性能上，FPGA 可能比不上专用芯片；或者至少在稳定性方面，FPGA 可能要逊色一些。
- 如果设计不需要太多的灵活性，FPGA 的灵活性反而是一种浪费，会潜在增加产品的成本。
- 相比特定功能、应用集中的 ASIC，使用 FPGA 实现相同功能可能产生更高的功耗。
- 在 FPGA 中除了实现专用标准器件（ASSP）所具有的复杂功能，还要添加一些额外的功能，实属一大挑战。FPGA 的设计复杂性和难度可能会给产品的开发带来一场噩梦。

1.3 FPGA 开发所需的技能

前面的文字已经做了很多铺垫，相信读者在初学 FPGA 时也一定早有耳闻。的确，FPGA 是一门相对复杂的技术，先不说设计本身，单是器件的内部结构、烦琐的开发流程、开发过程中需要接触的各种 EDA 工具，有时都会把工程师们折腾得晕头转向。

相比于单一处理器的开发，今天的 FPGA 开发却可能会涉及嵌入式系统设计的方方面面。如图 1-7 所示，很多时候，FPGA 工程师需要具备的技能可能要比其他任何设计团队多得多，如系统设计、功能分配、嵌入式处理器的实现、DSP 算法的实现、HDL 设计输入、仿

真、设计优化和高速电路板的布局布线以及特定信号接口的实现等。

至今在业内可能还有争论，到底 FPGA 是属于纯粹软件的范畴还是纯粹硬件的范畴。从 FPGA 设计的方式和手段上看，和软件有些相似；但从 FPGA 设计的思想和实现方面看，更多的趋向于硬件层面。因此，用“软硬兼施”这个词来形容 FPGA 工程师的工作就再贴切不过了。我们说的“软”，一方面的确是传统意义上的软件编程工作可能需要在 FPGA 中实现，另一方面的“软”则指的是 FPGA 逻辑实现所需要的代码输入方式。

单纯的软和硬，还只是很低级的技能。无论是在 FPGA 中内嵌了处理器，还是 FPGA 外部需要接口的处理器，FPGA 都不可避免地要和纯软件打交道，这种软硬衔接部分的设计非常有讲究，这也是 FPGA 工程师系统思维的一种要求。

虽然 FPGA 开发中需要涉及众多的技能门类，对于一些规模不大的工程项目，可能 FPGA 工程师确实要承担起“既当爹又当妈”的重任；但是对于中大规模的工程项目，一定会有一个分工明确的团队协作完成。这样的团队，绝对是一个不折不扣的嵌入式系统精英大荟萃：板级硬件工程师进行 FPGA 的板级原理图绘制、layout 等工作；FPGA 工程师做底层逻辑实现；嵌入式软件工程师做软件编程工作；也可能会有算法工程师的参与；当然还可能会有系统工程师统领全局（实际情况是，这个角色往往由项目经理兼任），对各个模块（包括软硬衔接部分）和接口进行划分和界定。

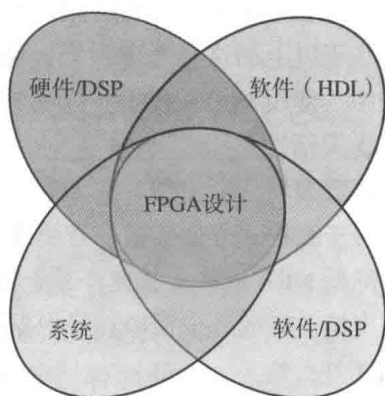


图 1-7 FPGA 技术是多种设计技能的交叉

1.4 FPGA 应用领域

目前 FPGA 虽然还受制于较高的开发门槛以及器件本身昂贵的价格，并从应用的普及率上来看和 ARM、DSP 还是有一定的差距，但是在非常多的应用场合，工程师们还是要别无选择地使用它。FPGA 所固有的灵活性和并行性其他芯片所不具备的，所以它的应用领域涵盖得很广。从技术角度来看，主要是有以下需求的应用场合（如图 1-8 所示）：

- ❑ 逻辑粘合，如一些嵌入式处理常常需要地址或外设扩展，CPLD 器件尤其适合。已经少有项目会选择一颗 FPGA 器件专门用于逻辑粘合的应用，但是在已经使用的 FPGA 器件中顺便做些逻辑粘合的工作倒是非常普遍。
- ❑ 实时控制，如液晶屏或电机等设备的驱动控制，此类应用也以 CPLD 或低端 FPGA 为主。
- ❑ 高速信号采集和处理，如高速 AD 前端或图像前端的采集和预处理，近年来持续升温的机器视觉应用也几乎无一例外地使用了 FPGA 器件。

- 协议实现，如更新较快的各种有线和无线通信标准、广播视频及其编解码算法、各种加密算法等，使用 FPGA 比 ASIC 更有竞争力。
- 各种原型验证系统，由于工艺的提升，流片成本也不断攀升，而在流片前使用 FPGA 做前期的验证已成为非常流行的做法。
- 片上系统，如 Altera 公司的 Soc FPGA 和 Xilinx 公司的 Zynq，这类 FPGA 器件，既有成熟的 ARM 硬核处理器，又有丰富的 FPGA 资源，大有单芯片一统天下的架势。

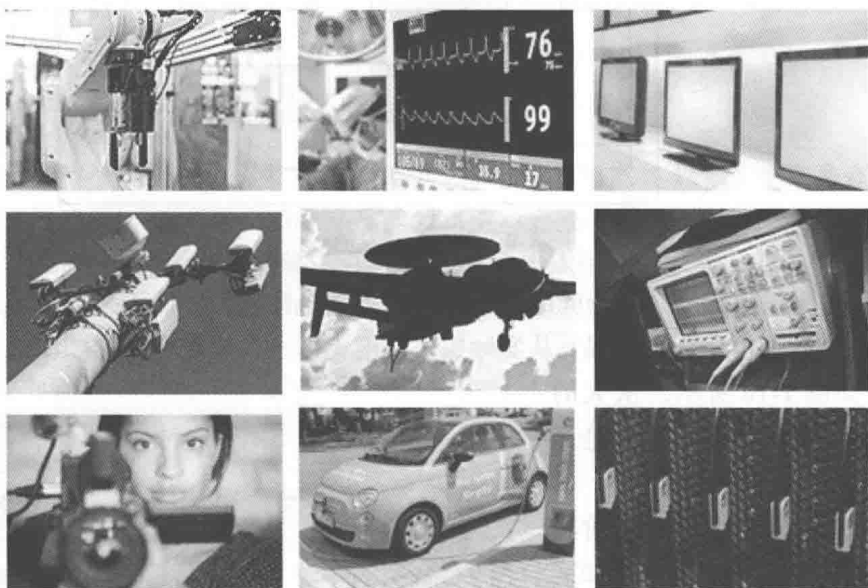


图 1-8 FPGA 应用精彩纷呈

当然了，若从具体的应用领域来看，FPGA 在电信、无线通信、有线通信、消费电子产品、视频和图像处理、车载、航空航天和国防、ASIC 原型开发、测试测量、存储、数据安全、医疗电子、高性能计算以及各种定制设计中都有涉猎。总而言之，FPGA 所诞生并发展的时代是一个好时代，与生俱来的一些特性也注定了它将会在这个时代的大舞台上大放光彩。

1.5 FPGA 开发流程

如图 1-9 所示。这个流程图是一个相对较高等级的 FPGA 开发流程，从项目提上议程开始，设计者需要进行 FPGA 功能的需求分析，然后进行模块的划分，比较复杂和庞大的设计则会通过模块划分把工作交给一个团队的多人协作完成。各个模块的具体任务和功能划分完毕（通常各个模块间的通信和接口方式也同时被确定），则可以着手进行详细设计，包括代码输入、综合优化、实现（映射和布局布线）。为了保证设计达到预期要求，各种设计的约束输入以及仿真验证也穿插其间。在 EDA 工具上验证无误后，则可以生成下载配置文件烧录到实际器件中进行板级的调试工作。从图 1-9 中的箭头示意不难看出，设计的迭代性是

FPGA 开发过程中的一个重要特点,这就要求设计者从一开始就要非常认真细致,否则后续的很多工作量可能就是不断返工。

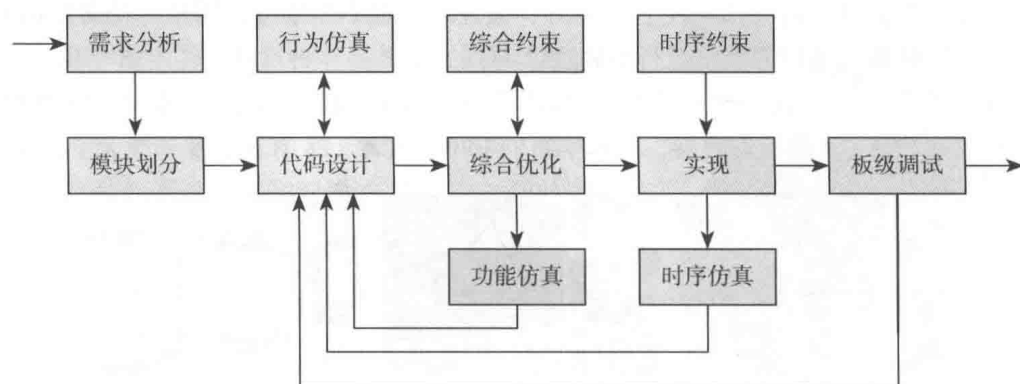


图 1-9 FPGA 开发流程

当然了,对于没有实际工程经验的初学者而言,这个流程图可能不是那么容易理解。不过没有关系,我们会简化这个过程,从实际操作角度,以一个比较简化的顺序的方式来理解这个流程。如图 1-10 所示,从大的方面来看, FPGA 开发流程不过是三个阶段,第一个阶段是概念阶段,或者也可以称之为架构阶段,这个阶段的任务是项目前期的立项准备,如需求的定义和分析、各个设计模块的划分;第二个阶段是设计实现阶段,这个阶段包括编写 RTL 代码、并对其进行初步的功能验证、逻辑综合和布局布线、时序验证,这一阶段是详细设计阶段;第三个阶段是 FPGA 器件实现,除了器件烧录和板级调试外,其实这个阶段也应该包括第二个阶段的布局布线和时序验证,因为这两个步骤都是和 FPGA 器件紧密相关的。我们这么粗略的三个阶段划分并没有把 FPGA 整个设计流程完全的孤立开来,恰恰相反,从我们的阶段划分中,我们也看到 FPGA 设计的各个环节是紧密衔接、相互影响的。

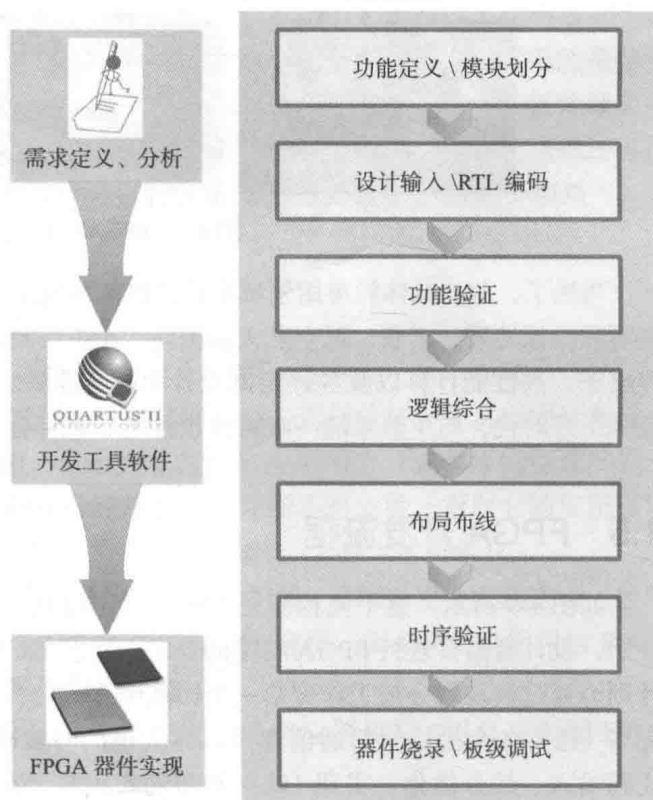


图 1-10 简化的 FPGA 开发流程