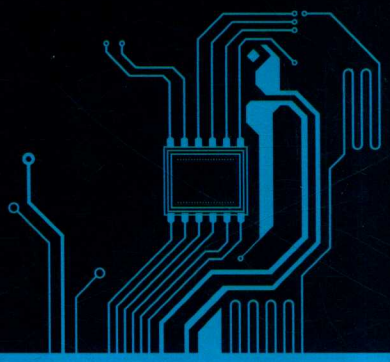
电子设计与嵌入式开发实践丛书

威视锐培训指定用书

Xilinx FPGA 伴你玩转USB 3.0与LVDS

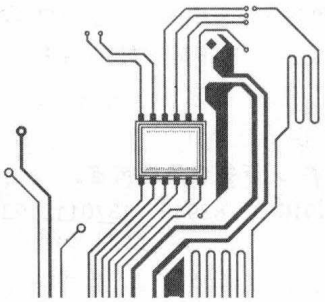
◎ 吴厚航 编著



- 基于Xilinx Artix-7 FPGA+USB 3.0+LVDS硬件开发平台
- 丰富的例程讲解：从基础的FPGA入门实例到基于FPGA的UART、DDR3、USB 3.0、LVDS传输实例
- 提供一站式开发学习方案：板级设计、软件工具与相关驱动安装、详细的项目工程解析与板级调试

清華大學出版社

电子设计与嵌入式开发
实践丛书



Xilinx FPGA
伴你玩转USB 3.0与LVDS

◎ 吴厚航　编著

清華大學出版社
北京

内 容 简 介

本书主要使用 Xilinx 公司的 Artix-7 FPGA 器件(引出自带的 LVDS 接口)和 Cypress 公司的 USB 3.0 控制器芯片 FX3,以及一些常见的 DDR3 存储器、UART 电路、扩展接口等,由浅入深地引领读者从板级设计、软件工具、相关驱动安装到基础的 FPGA 实例,从基于 FPGA 的 UART、DDR3、USB 3.0、LVDS 传输实例入手,掌握 FPGA 各种片内资源的应用以及接口时序的设计。

本书基于特定的 FPGA 开发平台,既有足够的理论知识深度进行支撑,也有丰富的例程进行实践讲解,并且穿插着笔者多年 FPGA 学习和开发过程中的各种经验和技巧。对于希望基于 FPGA 实现 USB 3.0 和 LVDS 开发的工程师,本书提供的很多实例都是很好的参考原型,可以帮助其实现快速系统原型的开发。

图书在版编目(CIP)数据

Xilinx FPGA 伴你玩转 USB 3.0 与 LVDS/吴厚航编著. —北京:清华大学出版社,2018
(电子设计与嵌入式开发实践丛书)
ISBN 978-7-302-49181-1

Ⅰ. ①X…　Ⅱ. ①吴…　Ⅲ. ①可编程序逻辑器件—系统设计　Ⅳ. ①TP332.1

中国版本图书馆 CIP 数据核字(2017)第 330845 号

责任编辑:刘　星
封面设计:刘　健
责任校对:徐俊伟
责任印制:杨　艳

出版发行:清华大学出版社
网　　址:http://www.tup.com.cn, http://www.wqbook.com
地　　址:北京清华大学学研大厦 A 座　　**邮　　编**:100084
社 总 机:010-62770175　　**邮　　购**:010-62786544
投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn
质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn
课件下载:http://www.tup.com.cn,010-62795954
印 装 者:北京嘉实印刷有限公司
经　　销:全国新华书店
开　　本:185mm×260mm　　**印　　张**:17.5　　**字　　数**:439 千字
版　　次:2018 年 5 月第 1 版　　**印　　次**:2018 年 5 月第1次印刷
印　　数:1～1500
定　　价:59.00 元

产品编号:077199-01

前言

FPGA 技术在当下各种电子设计应用中越来越火热，它的成本虽然还是高高在上，但是它给电子系统带来的不可限量的速度和带宽，以其在灵活性、小型性方面的优势，越来越被各种对性能要求高、偏重定制化需求的开发者所青睐。而使用 LVDS、USB 接口进行高速数据传输也是很多大数据采集应用的必然选择。

因此，笔者结合实际工程项目的需求，在本书中讲述基于 Xilinx Artix-7 FPGA 器件＋LVDS＋USB 3.0 的开发，通过实例讲解，相信可以帮助读者快速掌握这个原型系统，甚至将其移植到具体的项目或产品中。

本书共 8 章。

第 1 章阐释 FPGA、USB 和 LVDS 的一些基本概念和应用背景。

第 2 章从 FPGA 开发平台的电路板设计入手，介绍 FPGA 板级硬件电路设计要点，以及本书配套开发平台的外围电路的设计。

第 3 章讲述开发环境的搭建，包括 Xilinx FPGA 集成开发环境 Vivado、文本编辑器 Notepad＋＋、下载器驱动、UART 驱动、USB 3.0 控制器芯片 FX3 对应的 SDK 和驱动安装，帮助读者迅速解决这些最棘手的“软”问题。

第 4 章以一个最基本的 FPGA 实例引领读者掌握一个完整的 FPGA 开发流程，包括 FPGA 的下载配置和代码固化操作。

第 5 章为基础外设实例，包含基本的 LED 流水灯、拨码开关控制、PLL 配置、用户自定义 IP、UART 和搭建基于 MicroBlaze 处理器的嵌入式系统，通过这些基本的实例确保读者能够快速掌握基于 Xilinx FPGA 的开发。

第 6 章主要介绍 DDR3 SDRAM 的配置、仿真和板级调试。

第 7 章集中在 USB 3.0 控制器芯片 FX3 相关的实例上，既有单独 FX3 和 PC 的 USB 3.0 传输实例，也有 FPGA ＋ FX3 和 PC 之间的 USB 3.0 数据传输实例。

第 8 章讲解如何使用 FPGA 实现 LVDS 接口应用，包括 LVDS 基本的收发设计以及包含 CRC 校验的 LVDS 收发设计。

Foreword

本书配套例程请到清华大学出版社网站本书页面下载。

本书配套开发平台淘宝链接：

https://item.taobao.com/item.htm?spm=a1z10.5-c-s.w4002-15113370781.12.XApjMq&id=539571762506

吴厚航(网名:特权同学)

2017 年 12 月于上海

目录

第 1 章　FPGA、USB 与 LVDS 概述 …… 1

1.1　FPGA 发展概述 …… 1
1.2　FPGA 的优势 …… 3
1.3　FPGA 应用领域 …… 4
1.4　FPGA 开发流程 …… 5
1.5　USB 接口概述 …… 7
1.6　LVDS 接口概述 …… 9

第 2 章　实验平台板级电路详解 …… 11

2.1　板级电路整体架构 …… 11
2.2　电源电路 …… 12
2.3　FPGA 时钟与复位电路 …… 15
　2.3.1　FPGA 时钟晶振电路 …… 15
　2.3.2　FPGA 复位电路 …… 17
2.4　FPGA 配置电路 …… 18
2.5　FPGA 供电电路 …… 20
2.6　DDR3 芯片电路 …… 21
2.7　UART 芯片电路 …… 23
2.8　LVDS 接口电路 …… 24
2.9　USB 3.0 控制器 FX3 电路 …… 28
2.10　其他接口电路 …… 33
2.11　FPGA 引脚定义 …… 35

第 3 章　软件安装与配置 …… 47

3.1　Xilinx 账户注册与 Vivado 软件下载 …… 47
　3.1.1　Xilinx 账户注册 …… 47
　3.1.2　Vivado 下载 …… 50

Contents

3.2 Vivado 安装与免费 License 申请 …… 53
3.2.1 Vivado 安装 …… 53
3.2.2 免费 License 申请 …… 58
3.3 文本编辑器 Notepad++安装 …… 64
3.4 Vivado 中使用 Notepad++的关联设置 …… 67
3.5 串口芯片驱动安装 …… 69
3.5.1 驱动安装 …… 69
3.5.2 设备识别 …… 70
3.6 USB 3.0 控制器 FX3 的 SDK 安装 …… 70
3.7 USB 3.0 控制器 FX3 的驱动安装 …… 75
3.7.1 PC 与开发板的 USB 3.0 连接 …… 75
3.7.2 PC 与 USB 连接 …… 76
3.7.3 USB 3.0 控制器 FX3 驱动安装 …… 76

第 4 章 第一个例程与 FPGA 的下载配置 …… 78

4.1 流水灯实例 …… 78
4.1.1 功能概述 …… 78
4.1.2 新建 Vivado 工程 …… 78
4.1.3 创建工程源码、约束和仿真文件 …… 84
4.1.4 功能仿真 …… 94
4.1.5 编译 …… 96
4.2 Xilinx 7 系列 FPGA 配置概述 …… 98
4.2.1 不同配置模式的选择 …… 98
4.2.2 FPGA 配置比特流的大小 …… 98
4.2.3 FPGA 加载配置方式选择 …… 99
4.2.4 配置引脚功能定义 …… 100
4.3 XADC 温度监控界面 …… 101
4.4 bit 文件的 FPGA 在线烧录 …… 104
4.5 mcs 文件的 QSPI Flash 固化 …… 106
4.5.1 FPGA 配置设置选项 …… 106
4.5.2 生成 mcs 文件 …… 108
4.5.3 下载 mcs 件 …… 110

第 5 章 基础外设实例 …… 114

5.1 拨码开关的 LED 控制实例 …… 114
5.2 PLL 配置实例 …… 115
5.3 用户自定义 IP 核 …… 120
5.3.1 创建 IP 核 …… 120
5.3.2 移植 IP 核 …… 128

5.3.3 配置、例化 IP 核 …… 130
5.4 UART 的 loopback 实例 …… 134
5.4.1 功能概述 …… 134
5.4.2 代码解析 …… 135
5.4.3 板级调试 …… 144
5.5 MicroBlaze 的 Hello World 实验 …… 145
5.5.1 功能概述 …… 145
5.5.2 MicroBlaze 系统 IP 核配置 …… 146
5.5.3 MicroBlaze 处理器软件工程创建 …… 149
5.5.4 板级调试 …… 155

第 6 章 基于 FPGA 的 DDR3 存储器控制实例 …… 160

6.1 DDR3 IP 核配置与仿真 …… 160
6.1.1 DDR3 IP 核概述 …… 160
6.1.2 DDR3 IP 核配置 …… 160
6.1.3 DDR3 IP 核仿真 …… 171
6.2 基于在线逻辑分析仪监控的 DDR3 数据读/写 …… 178
6.2.1 功能概述 …… 178
6.2.2 DDR3 控制器 IP 接口时序解析 …… 178
6.2.3 代码解析 …… 182
6.2.4 在线逻辑分析仪配置 …… 187
6.2.5 在线逻辑分析仪调试 …… 190
6.3 基于 UART 命令的 DDR3 数据读/写 …… 192
6.3.1 功能概述 …… 192
6.3.2 代码解析 …… 192
6.3.3 板级调试 …… 196

第 7 章 USB 3.0 控制器 FX3 实例 …… 198

7.1 基于 FX3 内部 DMA 的 USB 传输 loopback 实例 …… 198
7.1.1 功能概述 …… 198
7.1.2 固件编译与固化文件生成 …… 198
7.1.3 硬件连接与设备识别 …… 202
7.1.4 FX3 的 USB Boot 加载 …… 205
7.1.5 板级调试 …… 207
7.2 FX3 的 SPI Flash 代码固化 …… 210
7.2.1 启动设置 …… 210
7.2.2 SPI Flash 固化 …… 211
7.3 基于 FX3 内部 DMA 的 USB 数据吞吐量测试 …… 212
7.3.1 硬件连接 …… 212

7.3.2 FX3 的 USB Boot 加载 …… 212
7.3.3 FX3 的基本 Streamer 功能演示 …… 215
7.4 基于 FX3 的 UVC(USB Video Class)传输协议实例 …… 217
7.4.1 硬件连接 …… 217
7.4.2 FX3 的 USB Boot 加载 …… 217
7.4.3 UVC 设备识别 …… 218
7.5 FX3 固件 SlaveFIFO 配置修改说明 …… 218
7.5.1 功能概述 …… 218
7.5.2 GPIF Ⅱ Designer 开启与工程加载 …… 219
7.5.3 GPIF Ⅱ接口配置与工程编译 …… 220
7.5.4 IDE 下 firmware 工程加载 …… 221
7.5.5 IDE 下 firmware 工程编译 …… 225
7.6 基于 FPGA-FX3 SlaveFIFO 接口的 loopback 实例 …… 226
7.6.1 功能概述 …… 226
7.6.2 Firmware 下载 …… 226
7.6.3 FPGA 代码解析 …… 227
7.6.4 ILA 在线逻辑分析仪查看接口时序 …… 229
7.7 基于 FPGA-FX3 SlaveFIFO 接口的 StreamOUT 实例 …… 232
7.7.1 功能概述 …… 232
7.7.2 Firmware 下载 …… 232
7.7.3 FPGA 代码解析 …… 232
7.7.4 ILA 在线逻辑分析仪查看接口时序 …… 234
7.7.5 Streamer 中查看数据吞吐量 …… 236
7.8 基于 FPGA-FX3 SlaveFIFO 接口的 StreamIN 实例 …… 238
7.8.1 功能概述 …… 238
7.8.2 Firmware 下载 …… 238
7.8.3 FPGA 代码解析 …… 238
7.8.4 ILA 在线逻辑分析仪查看接口时序 …… 239
7.8.5 Streamer 工具测试数据吞吐量 …… 241

第 8 章 LVDS 收发传输实例 …… 243

8.1 LVDS 数据收发实例 …… 243
8.1.1 功能概述 …… 243
8.1.2 bit align 处理 …… 244
8.1.3 代码解析 …… 245
8.1.4 装配说明 …… 257
8.1.5 板级调试 …… 257
8.2 带 CRC 校验的 LVDS 数据收发实例 …… 258
8.2.1 功能概述 …… 258

8.2.2 CRC 校验基本原理 …… 259
8.2.3 CRC8 检验代码生成 …… 260
8.2.4 代码解析 …… 264
8.2.5 装配说明 …… 268
8.2.6 板级调试 …… 268

参考文献 …… 269

第1章

FPGA、USB 与 LVDS 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 同台比对，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；另外，我们也将论述 FPGA 的应用领域和开发流程；最后，对于 USB 3.0 和 LVDS 接口也会做一些基础性的介绍。总而言之，本章不会讲述很高深的理论知识，而是力图以浅显易懂的语言，让读者搞清楚本书随后章节将要接触到的“高大上”的 FPGA、USB 3.0 和 LVDS 技术。

1.1 FPGA 发展概述

20 世纪 60 年代中期，TI 公司设计制造了各种各样的实现基本逻辑门电路功能的芯片，相信今天很多的工程师仍然很熟悉这些主要面向军工应用的 54XX 和商业应用的 74XX 芯片。据说早期的工程师甚至能够单凭着这些芯片架构出一个简单 CPU 的功能。还真别小瞧这些基本逻辑门电路，话说万丈高楼平地起，如果说今天在嵌入式领域“呼风唤雨”的各种功能强大的 ARM7、ARM9、DSP 是万丈高楼，那么称这些基本的逻辑门电路为一砖一瓦倒是一点也不为过。

从 1971 年 Intel 公司的第一个 4 位微处理器 Intel 4004 到 20 世纪 80 年代初被奉为经典的 8051 单片机，再到今天各大嵌入式处理器厂商竞相使用的由 ARM 公司推出的各种 Cortex 内核，嵌入式处理器的发展不可不说是翻天覆地。不过话又说回来，如果深入处理器的底层结构，你会发现它们最本质的东西并没有太大的改变。而处理器功能再强大，一个芯片尽可以将各种外设嵌入其中，但对于任何一个已经批量出货的芯片而言，它的功能是固定的，若想在既有外设功能的基础上有任何的扩展，或许不是遇到电气特性不支持就是遇到 I/O 太少的尴尬局面，而这些问题也就催生了可编程逻辑器件的诞生。今天的 CPU 周围已很难看见 54 或 74 字样的 ASIC 了，取而代之的可能是引脚密集的 CPLD 或 FPGA。的确，在系统的可扩展性和灵活性方面，FPGA/CPLD 有着得天独厚的优势。当然，今天动辄上百万门的 FPGA 器件可不是为干这点活而制造的，它更多地被应用到了通信、数据采集、网

络等对数据传输速度和吞吐量有更高要求的场合。

今天大家熟知的 FPGA/CPLD 也不是一开始就有的，第一款可编程逻辑器件(PLD)最初是在 1970 年以 PROM 的形式进入人们的视野，这种 PROM 结构的可编程逻辑器件可以实现简单的逻辑功能，很容易便可替代当时流行的 54 或 74 系列逻辑门电路。

受限于 PROM 的结构，第一款可编程逻辑器件输入相对较少。因此，可编程逻辑阵列(PAL)便孕育而生，PAL 由一个可编程的“与”平面和一个固定的“或”平面构成，或门的输出可以通过触发器有选择地被置为寄存状态。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。PAL 的问题在于其实现方式使得信号通过可编程连线的时间相对较长。在 PAL 的基础上，又发展了一种通用阵列逻辑(GAL)，它要比 PAL 速度快许多，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。

这些早期的 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。电子领域的发展趋势总是朝着速度更快、功能更强、体积更小、成本更廉价的方向迈进。复杂可编程逻辑器件(CPLD)的诞生也就顺理成章了。Altera 公司于 1984 年发明了基于 CMOS 和 EPROM 技术相结合的 CPLD。CPLD 可实现的逻辑功能相比 PAL 和 GAL 有了大幅度的提升，已经可以胜任设计中复杂性较高、速度也较快的逻辑功能，尤其在接口转换、总线控制和扩展方面有着较多的应用。经过几十年的发展，今天的 CPLD 功能和性能也得到了进一步的提升，其基本结构由可编程 I/O 单元、基本逻辑单元、布线池以及其他相关辅助功能块组成。Altera、Xilinx 和 Lattice 是主要的 CPLD 供应商。

其实无论是前面提到的 PAL、GAL 或是 CPLD，要实现大规模的复杂逻辑电路都显得无能为力。而 ASIC 的设计耗时又费钱，而且功能固定，在流片后很难随意更改。鉴于此，Xilinx 创始人之一 Ross Freeman 发明了现场可编程门阵列(FPGA)，Freeman 先生发明的 FPGA 是一块全部由“开放式门”组成的计算机芯片。采用该芯片，工程师可以根据需要进行灵活编程，添加各种新功能，以满足不断发展的协议标准或规范，工程师们甚至可以在设计的最后阶段对它进行修改和升级。Freeman 先生当时就推测低成本、高灵活性的 FPGA 将成为各种应用中定制芯片的替代品。也正是由于此项伟大的发明，让 Freeman 先生于 2009 年荣登美国发明家名人堂。

现今，伴随着制造工艺的不断进步，FPGA 在深亚微米甚至深亚纳米时代一直走在了创新第一线。如今的 FPGA 器件，其组成不仅限于基本的可编程 I/O 单元、可编程逻辑单元、丰富的布线资源，而且还拥有灵活的时钟管理单元、嵌入式块 RAM 以及各种通用的内嵌功能单元，很多器件还顺应市场需求内嵌专用的硬件模块。近些年来，可编程器件的龙头老大 Xilinx 和 Altera 更是相继推出了硬核 CPU+FPGA 的产品，此举大有单芯片“横扫千军”的架势。

电子行业在继续挑战摩尔定律的征程中，无论是可编程器件继续大放光彩，还是 ASIC 能够重获新生。可编程器件，尤其是 FPGA/CPLD 的发明和大量应用已经足够让我们肃然起敬。相信对于很多即将或者已经走上电子硬件设计的同仁们，对可编程器件的了解、熟悉甚至精通是提升自身技术能力的基本技能之一。

1.2 FPGA 的优势

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，如果产品功能必须进行较大的升级或变更，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，需要重点从以下几个方面进行评估。

- 可升级性——产品在设计过程中，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、吞吐量或处理能力上是否有特殊要求？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，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，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，会潜在地增加产品的成本。
- 相比固定功能、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，还得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 应用领域

FPGA 目前虽然还受制于较高的开发门槛以及器件本身昂贵的价格，应用的普及与 ARM、DSP 还是有一定的差距，但是在非常多的应用场合，工程师们还是会别无选择地使用它。FPGA 所固有的灵活性和并行性是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，主要是具有以下需求的应用场合。

- 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个 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 A/D 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是无一例外地都使用了 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。

当然，若从具体的应用领域来看，FPGA 在无线通信、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，如图 1.1 所示。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。

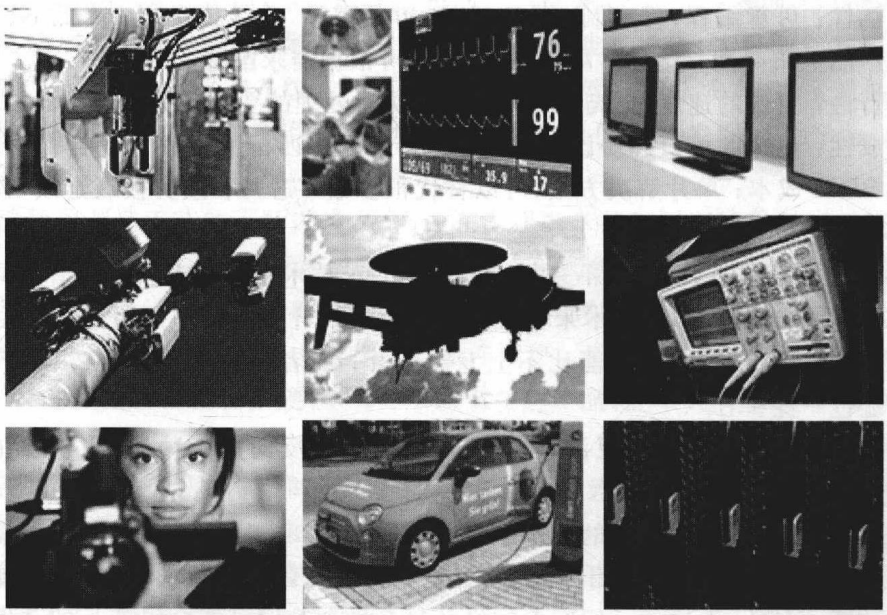


图 1.1 FPGA 应用精彩纷呈

1.4　FPGA 开发流程

如图 1.2 所示，是一个基于 FPGA 开发工具的开发流程图。当然，在此之前，从 FPGA 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，其各个步骤基本就如同图 1.2 所示的流程图，包括设计输入、设计综合、约束输入、设计实现、分析实现结果(查看工具给出的各种报告结果)。为了保证设计达到预期要求，设计仿真以及设计优化则穿插其间。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地返工。

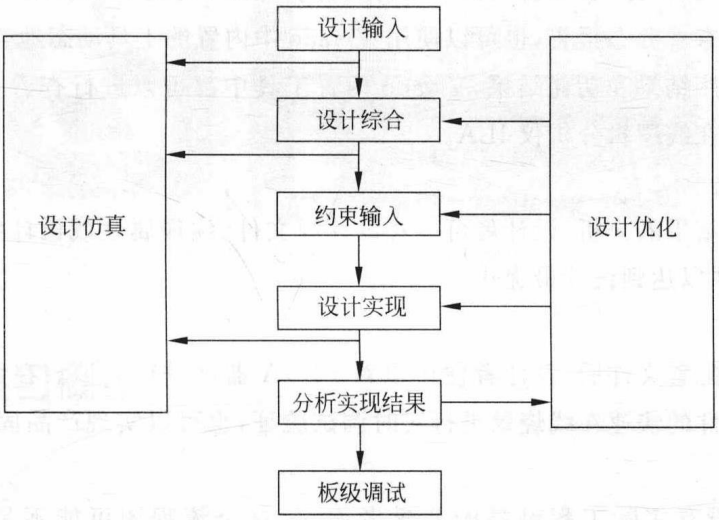


图 1.2　FPGA 开发流程

基于 Xilinx 的 Vivado 开发工具，对以上开发流程所涉及的各个步骤做简要的说明。

1) 设计输入

设计输入阶段，设计者需要创建 FPGA 工程，并且创建或添加设计源文件到工程中。FPGA 工程包含了各种不同类型的源文件和设计模块，例如 HDL 文件、EDIF 或 NGC 网表文件、原理图、IP 核模块、嵌入式处理器以及数字信号处理器模块等。

2) 设计综合

设计综合阶段，FPGA 开发工具的综合引擎将编译整个设计，并将 HDL 源文件转译为特定结构的设计网表。Vivado 设计工具内置 Synthesis 综合功能，也支持第三方综合工具，如 Synplify、Synplify Pro 和 Precision 等软件工具的使用。

3）设计仿真

在整个开发过程的任意时刻，设计者都可以使用仿真工具对 FPGA 工程进行功能验证，例如 Vivado 内置的仿真器或者第三方工具 ModelSim 仿真器。

4）约束输入

约束输入阶段，设计者可以指定时序、布局布线或者其他的设计要求。Vivado 工具支持专用的编辑器实现时序约束、I/O 引脚约束和布局布线约束。

5）设计实现

设计综合之后，接着就需要设计实现，将逻辑设计进一步转译为可以被下载烧录到目标 FPGA 器件中的特定物理文件格式。使用 Vivado 的工程导航窗口中支持的目标和策略设置属性，可以控制设计实现以及结果优化。为了更快地达到设计目标，可以使用 SmartXplorer 进行不同的处理策略实现，达到多次自动实现处理以完成设计目标。

6）分析实现结果

完成设计实现后，必须对设计约束、器件资源占用率、实现结果以及功耗等设计性能进行分析。既可以查看静态报告，也可以使用 Vivado 中内置的工具动态地查看设计综合实现的结果。对于时序结果和功耗结果，Vivado 内置工具中都可以进行查看。此外，在系统调试时也可以使用在线逻辑分析仪 ILA。

7）设计优化

基于对设计结果的分析，设计者可以对设计源文件、编译属性或设计约束进行修改，然后重新综合、实现以达到设计最优化。

8）板级调试

在生成下载配置文件后，设计者便可以对 FPGA 器件进行调试。在此过程中，既可以实现下载配置文件的快速在线烧录进行实时调试验证，也可以实现产品固化烧录使其可以离线运行。

当然，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 1.3 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。这么粗略的三个阶段划分并没有把 FPGA 整个设计流程完全孤立开来，恰恰相反，从这种阶段划分中，也可看到 FPGA 设计的各个环节是紧密衔接、相互影响的。

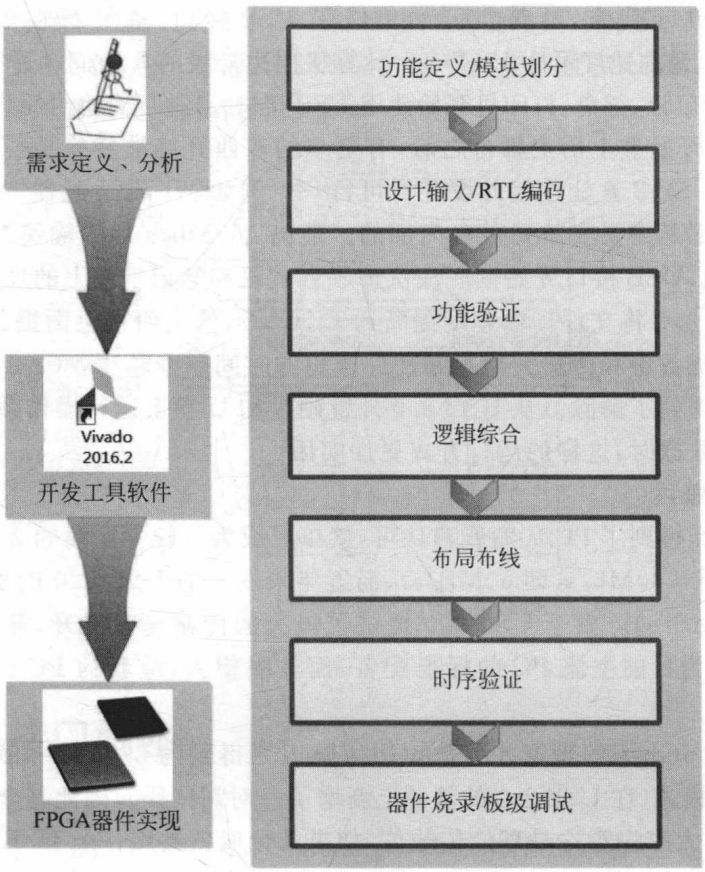


图 1.3　简化的 FPGA 开发流程

1.5　USB 接口概述

1. USB 发展史

可以这么说，如今 PC 所到之处，必有 USB 接口如影相随。和所有的技术一样，USB 也随着时间的推移而慢慢演变。在其问世的 20 多年里，USB 的速度不断提升，并且衍生出了许多不同的接口和线缆。

对于 20 世纪 90 年代就开始折腾计算机的玩家而言，对于如图 1.4 所示的大块头接口一定不陌生。而 USB 所取代的，正是这些“大家伙”当年所做的事——PC 与外部设备的数据传输。

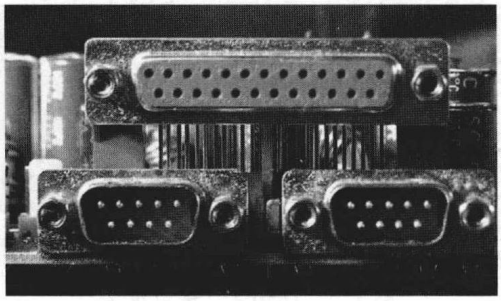


图 1.4　串口和并口