

FDA 工 | 程 | 技 | 术 | 丛 | 书 |

Xilinx大学计划推荐用书



资深工程师以实际经验讲述Xilinx ZYNQ-7000 AP SoC开发的设计方法与流程
讲究思想与方法，注重实践和调试！



THE DEVELOPMENT AND PRACTICAL GUIDE OF
XILINX ZYNQ-7000 AP SOC

Xilinx ZYNQ-7000 AP SoC 开发实战指南

符晓 张国斌 朱洪顺 编著
Fu Xiao Zhang Guobin Zhu Hongshun

清华大学出版社



EDA 工 | 程 | 技 | 术 | 丛 | 书 |



THE DEVELOPMENT AND PRACTICAL GUIDE OF
XILINX ZYNQ-7000AP SOC

Xilinx ZYNQ-7000 AP SoC 开发实战指南

符晓 张国斌 朱洪顺 编著
Fu Xiao Zhang Guobin Zhu Hongshun

清华大学出版社
北京

内 容 简 介

本书基于 Xilinx 公司的 ZYNQ-7000 AP SoC,介绍了其体系结构与开发思想,并使用多个实例讲述了其开发方法与流程。全书共 9 章。书中讲述了 ZYNQ-7000 AP SoC 家族的特点、体系与结构以及软件开发的独特之处;以 Vivado 开发套件为基础,讲述了 ZYNQ-7000 AP SoC 的软硬件开发流程;为了方便使用 ISE/PlanAhead 软件的读者入手,还简要描述了使用它们开发 ZYNQ-7000 AP SoC 嵌入式软件的方法,但本书仍以 Vivado 套件为主要工具进行开发讲解;给出了常用外设的使用示例,包括 MIO/EMIO 接口、通用 I/O、中断控制器、定时器系统等,还给出了 XADC 模块的使用示例;围绕 Vivado 以 IP 为中心的设计思想,用实例讲解了如何设计用户自定义 IP 核;使用 System Generator for DSP 在 Matlab/Simulink 环境下建模,介绍了基于模型的 DSP 算法设计,并通过多个实例讲解了其设计思想和设计流程;使用 Vivado HLS 软件,通过多个实例讲述了高层次综合的设计思想和设计流程。

本书可作为电子通信、软件工程、自动控制、智能仪器和物联网相关专业高年级本科生或研究生学习嵌入式操作系统及其应用技术的教材,也可作为嵌入式系统开发和研究人员的参考用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Xilinx ZYNQ-7000 AP SoC 开发实战指南/符晓,张国斌,朱洪顺编著. —北京:清华大学出版社,2016
EDA 工程技术丛书

ISBN 978-7-302-41491-9

I. ①X… II. ①符… ②张… ③朱… III. ①可编程序逻辑器件—系统设计—指南
IV. ①TP332.1-62

中国版本图书馆 CIP 数据核字(2015)第 212880 号

责任编辑:刘 星

封面设计:李召霞

责任校对:焦丽丽

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:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:19

字 数:463 千字

版 次:2016 年 1 月第 1 版

印 次:2016 年 1 月第 1 次印刷

印 数:1~2500

定 价:49.00 元

产品编号:060455-01

Xilinx ZYNQ[®]-7000 All Programmable(AP)SoC 系列器件将处理器的软件可编程能力与 FPGA 的硬件可编程能力完美结合,通过硬件、软件和 I/O 可编程性实现了扩展式系统级差异、集成和灵活性,并以其低功耗和低成本等系统优势实现无与伦比的系统性能,同时可以加速产品上市进程。与基于传统 SoC 的处理解决方案不同,ZYNQ-7000 器件的灵活可编程逻辑能实现优化与差异化功能,使设计人员可以根据大部分应用的要求添加外设和加速器。通过 ZYNQ-7000 AP SoC 平台,设计人员可以设计更智能的系统,控制和分析部分利用灵活的软件、紧密配合擅长实时处理的硬件,辅之以优化的系统接口,从而使得 BOM 成本可大幅削减、NRE 成本更低、设计风险减少、加快上市时间。

本书导读

本书基于 Xilinx 公司的 ZYNQ-7000 AP SoC,介绍了其体系结构与开发思想,并使用多个实例讲述了其开发方法与流程。

全书共 9 章。第 1 章讲述了 ZYNQ-7000 AP SoC 家族的特点,及其与传统 FPGA 和 SoC 的区别,给读者提供了一定的背景资料,使得读者对 ZYNQ-7000 AP SoC 的芯片和开发思想具有整体的概念。

第 2 章简要介绍了 ZYNQ-7000 AP SoC 的体系与结构,包括应用处理单元、接口与引脚、时钟、复位、JTAG 调试与测试、启动与配置、系统的互联结构和可编程逻辑,并着重描述了 ZYNQ-7000 AP SoC 软件开发的独特之处以及设计基于可编程逻辑的算法加速器时需要考虑的多个问题。初学者一开始可能不容易理解这些内容,但是对这些内容有基本的理解之后,能更好地在编程、开发时,针对 ZYNQ-7000 AP SoC 的体系设计更高效的软硬件架构。因为 ZYNQ-7000 AP SoC 的特性众多,本书并未对器件手册和用户指南进行简单的翻译、复制,而是根据作者的理解、认识进行了归类描述。

第 3 章以 Vivado 开发套件为基础,讲述了 ZYNQ-7000 AP SoC 的软硬件开发流程;为了方便使用 ISE/PlanAhead 软件的读者入手,还简要描述了使用它们开发 ZYNQ-7000 AP SoC 嵌入式软件的方法,但本书仍以 Vivado 套件为主要工具进行开发讲解。只有熟练掌握了 Vivado 套件的使用方法,才能做到高效的开发效率。

接下来的 3 章讲述了 ZYNQ-7000 AP SoC 的软硬件协同开发的思想与方法。其中,第 4 章为 ZYNQ-7000 AP SoC 中常用外设的使用示例,包括 MIO/EMIO 接口、通用 I/O、中断控制器、定时器系统等。第 5 章为 XADC 模块的使用示例。第 6 章围绕 Vivado 以 IP 为中心的设计思想,用实例讲解了如何设计用户自定义 IP 核。

第 7 章与第 8 章讲述了针对 ZYNQ-7000 AP SoC 的、不同于传统 HDL 和 C 手工编码的高级开发方法。其中,第 7 章使用 System Generator for DSP 在 Matlab/Simulink 环境下建模,介绍了基于模型的 DSP 算法设计,并通过多个实例讲解了其设计思想和设计流程。第 8 章讲述了高层次综合的设计思想,使用 Vivado HLS 软件,通过多个实例讲

前言

述了高层次综合的设计思想和设计流程。这些内容并不是初学者所必须掌握的,然而它们可以作为更高级的开发方式,在复杂的、面向产品的开发过程中起到非常重要的作用。

第9章详细介绍了本书中所使用的安富利 MicroZed 平台的特点、基本使用方法、常用外设的测试过程和运行开源 Linux 的方法。使用其他 ZYNQ-7000 AP SoC 硬件平台的读者,也可结合其对应的硬件接口,进行相关的测试与验证工作。

相关资源

本书中所有实例的相关源代码都可在清华大学出版社网站下载。

在开发过程中如果有疑问,欢迎到电子创新网的赛灵思社区交流: <http://xilinx.eetrend.com>; 作者将不定期在此网站发布勘误表、问题解答等。

有关 MicroZed 开发板的问题可访问以下网址:

<http://zedboard.com/product/microzed>

致谢

在本书的写作过程中,得到了赛灵思公司中国区公共关系经理张俊伟女士、工业市场营销经理林逸芳女士和亚太区业务拓展经理罗霖先生的诸多帮助和鼓励,并最终促成了本书的编写;感谢赛灵思为本书的编写所提供的软件授权。安富利公司的高级技术市场经理陈志勇博士和高级高级现场销售工程师黄志刚为本书的写作提供了技术支持和最新的 MicroZed 开发系统。感谢电子创新网为本书的实验部分提供了高速、稳定的下载地址。感谢清华大学出版社工作人员为本书的出版所做的大量工作。最后要感谢家人和朋友们的支持。

限于笔者的水平和经验,加之时间比较仓促,疏漏或者错误之处在所难免,敬请读者批评指正。有兴趣的朋友可发送邮件到 netsky985@163.com,与作者进行交流;也可发送邮件到 workemail6@163.com,与本书策划编辑进行交流。

免责声明

本书内容仅用于教学和科研目的,书中引用的部分例子、图形和图表等内容的知识产权归 Xilinx 公司与 Avent 公司所有,作者保留其余内容的所有权利。禁止任何单位或个人摘抄或扩充本书内容用于出版发行,严禁将本书内容应用与商业场合。

作者

2015年9月

第 1 章 不只是芯片,更是完整的平台产品	1
1.1 FPGA 的这三十年	1
1.2 FPGA 的芯片结构	1
1.3 传统的 FPGA 开发基本流程	3
1.4 Xilinx FPGA 家族介绍	6
1.5 Xilinx 开发工具与设计平台	7
1.5.1 ISE 与 Vivado、Vivado HLS 简介	7
1.5.2 System Generator 简介	11
1.6 为什么使用 ZYNQ	11
1.6.1 ZYNQ 家族的优势	12
1.6.2 ZYNQ 家族的主要应用	13
1.6.3 现有的 ZYNQ 家族器件	14
1.6.4 ZYNQ 家族的特性	15
1.7 UltraFast 设计方法	17
第 2 章 ZYNQ 的体系、结构与开发思想	19
2.1 应用处理器单元	20
2.1.1 APU 的基本功能	20
2.1.2 APU 的系统级视图	22
2.2 信号、接口与引脚	23
2.2.1 电源引脚	23
2.2.2 PS I/O 引脚	24
2.2.3 PS-PL 电平移位使能	25
2.2.4 PS-PL MIO-EMIO 信号与接口	25
2.3 时钟	28
2.3.1 时钟系统	28
2.3.2 CPU 时钟	30
2.4 复位	32
2.4.1 复位后的启动流程	34
2.4.2 复位资源	35
2.5 JTAG 调试与测试	36
2.6 启动与配置	38
2.6.1 PS 的启动过程	38
2.6.2 PL 的启动过程	41

目录

2.7	系统互联结构	42
2.8	可编程逻辑 PL	47
2.8.1	PL 的组件	48
2.8.2	输入/输出	51
2.8.3	PL 的配置	54
2.9	ZYNQ 开发思想	54
2.9.1	ZYNQ-7000 软件开发的特点	54
2.9.2	ZYNQ-7000 SoC 软件与应用的开发流程	56
2.9.3	设备的驱动架构	58
2.9.4	裸机程序开发流程	60
2.9.5	Linux 程序开发	63
2.10	设计基于 PL 的算法加速器	66
2.10.1	用 PL 为 PS 卸载	66
2.10.2	PL 与存储系统的性能	70
2.10.3	选择 PL 接口	72
第 3 章	ZYNQ-7000 AP SoC 设计与开发流程	77
3.1	ZYNQ-7000 AP SoC 开发流程简介	77
3.2	基于 Vivado+SDK 的设计与开发	79
3.2.1	使用 Vivado 构建硬件平台	79
3.2.2	使用 SDK 完成软件开发	91
3.2.3	启动镜像文件的生成与下载	97
3.3	基于 PlanAhead+SDK 的设计与开发	103
第 4 章	ARM Cortex-A9 外围设备应用实例	111
4.1	MIO/EMIO 接口	111
4.1.1	MIO/EMIO 接口功能概述	111
4.1.2	应用实例	112
4.2	通用 I/O 模块 GPIO	115
4.2.1	GPIO 简介	115
4.2.2	功能详述	116
4.2.3	编程指南	119
4.2.4	应用实例	120
4.3	中断控制器 GIC	121
4.3.1	GIC 简介	121

4.3.2	中断源分类	122
4.3.3	中断优先级仲裁	124
4.3.4	相关寄存器	125
4.3.5	应用实例	125
4.4	定时器系统	128
4.4.1	定时器系统简介	128
4.4.2	私有定时器、私有看门狗	129
4.4.3	全局定时器	129
4.4.4	系统看门狗	130
4.4.5	TTC 单元	131
4.4.6	编程指南	134
4.4.7	相关寄存器	134
4.4.8	应用实例	136
第 5 章	XADC 模块应用实例	139
5.1	简介	139
5.2	功能详述	141
5.2.1	XADC 模块相关引脚	141
5.2.2	模拟量输入类型及量化关系	142
5.2.3	电压、温度的记录与报警	146
5.2.4	自动校正功能	149
5.3	XADC 工作模式	150
5.3.1	单通道模式	150
5.3.2	自动序列模式	150
5.3.3	外部多路复用器模式	154
5.4	控制接口	155
5.4.1	DPR/JTAG-TAP 接口	155
5.4.2	常用接口单元	157
5.5	相关寄存器	158
5.5.1	状态寄存器	158
5.5.2	控制寄存器	159
5.6	应用实例	160
5.6.1	基于 Vivado 的 XADC 模块硬件配置	160
5.6.2	基于 SDK 的软件开发	164

目录

第 6 章 用户 IP 核的定制	167
6.1 基于 Vivado 的用户 IP 核封装与例化	167
6.1.1 用户 IP 核的建立	167
6.1.2 用户 IP 核逻辑功能的设计与封装	172
6.1.3 用户 IP 核的例化	177
6.2 基于 SDK 的编程指导	178
第 7 章 基于模型的 DSP 设计	181
7.1 System Generator 的安装、系统要求与配置	182
7.2 Simulink 的基本使用方法	183
7.3 创建基于 System Generator 的简单设计	187
7.4 定点数据类型的处理	195
7.5 系统控制与状态机	198
7.6 多速率与串并转换	201
7.7 使用存储单元	203
7.8 在 Vivado IDE 中使用 System Generator 模型	206
7.9 把 C/C++ 程序导入 System Generator 模型	209
7.10 把 System Generator 模型封装为自定义 IP	215
7.11 对 System Generator 中生成的 AXI4-Lite 接口的模型进行验证	224
第 8 章 Vivado 高层次综合	228
8.1 Vivado HLS 的基本开发方法	229
8.2 Vivado HLS 中的数据类型	239
8.2.1 任意精度整数类型	239
8.2.2 Vivado HLS 支持的数学函数类型	243
8.3 Vivado HLS 中的接口综合	244
8.3.1 模块级别的 I/O 协议	244
8.3.2 端口类型的处理	246
8.3.3 如何把数组实现为 RTL 接口	249
8.3.4 如何把数组实现为 AXI4 的相关接口	253
8.4 在 Vivado IPI 中使用 HLS 生成的 IP	257
8.5 把使用 HLS 生成的 IP 用作 PS 的外设	262

第 9 章	MicroZed 开发板的介绍	270
9.1	MicroZed 基本介绍	270
9.2	下载程序与测试	273
9.3	测试更多的 DDR 内存空间	277
9.4	在 MicroZed 上运行开源 Linux	279
9.4.1	在 Linux 中控制 GPIO	279
9.4.2	在 Linux 中进行以太网通信	282
9.4.3	测试 PS 与 USB 的通信	285
9.4.4	由 PS 向 PL 提供时钟信号	287
参考文献	293

1.1 FPGA 的这三十年

自从三十年前，赛灵思制造出了世界上首款 FPGA(1985 年推出相当于 1000 个 ASIC 门的 XC2064)以来，FPGA 已经取得了长足的发展。最早的 FPGA 定位于门阵列和 ASIC 的替代品，主要用作“粘性逻辑”，协助两个最初设计不相互通信的器件进行对话，此外也能在设计最后时刻为大型 ASIC 补充此前缺失的功能。限于当时的工艺水平，最早只有 1K 级别的可用门资源。

随着工艺水平与设计方法的日新月异，如今的 FPGA 已经从当初 XC2064 的 $2\mu\text{m}$ 工艺、64 个逻辑模块、85 000 个晶体管和不到 1K 的门，发展到新的 UltraScale 系列，其中包括目前最强大的使用 3D IC 堆叠技术的 20nm 工艺、高达 4 407 480 个逻辑单元的 XCVU440(截至 2014 年 6 月)。其应用领域也从早期的逻辑胶合发展成为集算法逻辑、数字信号处理 DSP、高速串行收发以及软核/硬核处理器为一体的片上系统 SoC。

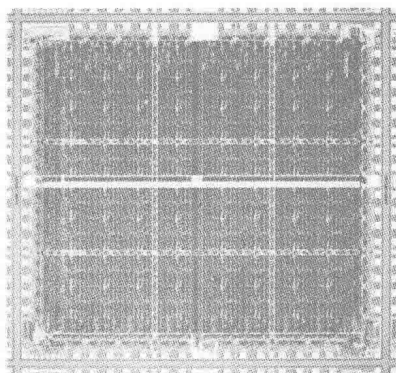


图 1-1 Xilinx XC2064 FPGA：
LUT 的诞生

1.2 FPGA 的芯片结构

FPGA 是由通过可编程相互连接的可配置逻辑块(CLB)矩阵构成的可编程半导体器件。相对于专为特定设计定制构建的专用集成电路(ASIC)而言，FPGA 能通过编程来满足应用和功能要求。虽然市面上也有一次性可编程(OTP) FPGA，但绝大多数是基于 SRAM 的类型，可随着设计的演化进行重新编程。FPGA 可支持工程师在设计

周期的后期进行修改,甚至能够在生产后给产品升级新的功能。此外,Xilinx FPGA 能够远程完成现场升级,消除了与重新设计或手动更新电子系统有关的成本。

目前的FPGA已经远远超出了先前版本的基本性能,并且整合了如RAM、时钟管理和DSP这些常用功能的硬(ASIC型)块。FPGA的基本单元结构如图1-2所示。

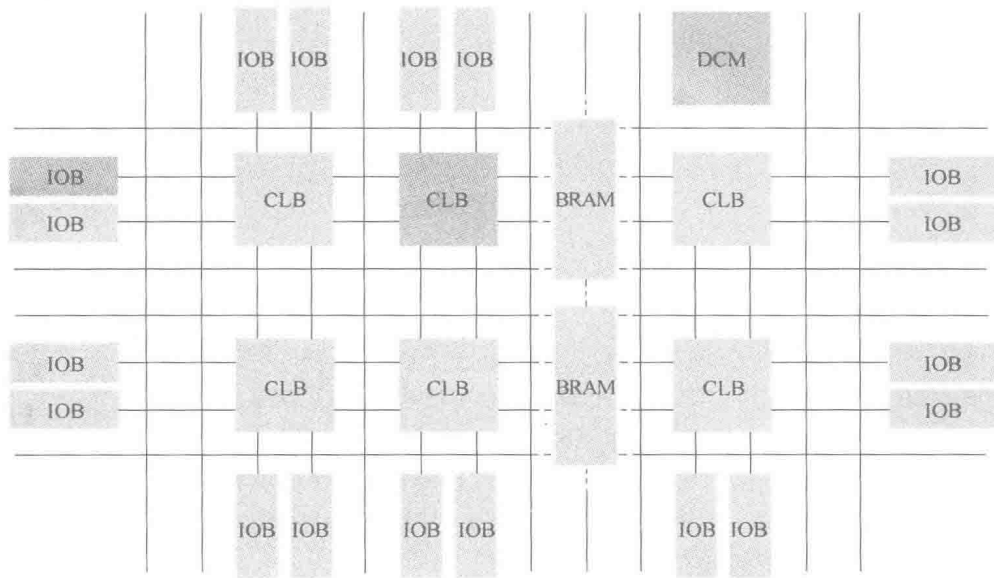


图 1-2 FPGA 的基本单元结构

下面简要介绍图 1-2 中各基本模块的组成与特点。

1. 可配置逻辑块(CLB)

CLB是FPGA的基本逻辑单元,其基本结构如图1-3所示。CLB实际数量和特性会依器件的不同而改变,但是每个CLB都包含一个由4或6个输入、一些选择电路(多路复用器等)和触发器组成的可配置开关矩阵。开关矩阵具有高度的灵活性,经配置可以处

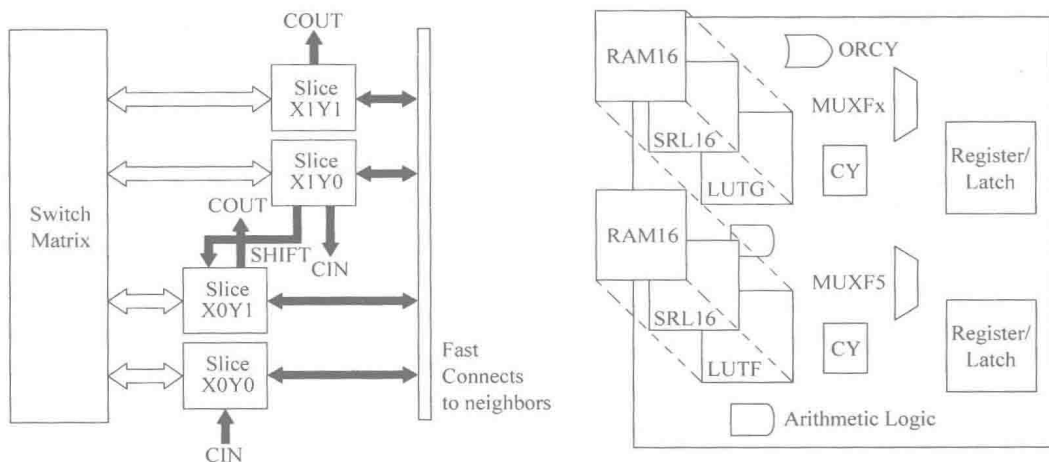


图 1-3 CLB 的基本结构

理组合型逻辑、移位寄存器或 RAM。

2. 互联

CLB 提供逻辑性能,灵活的互联布线则负责在 CLB 和 I/O 之间传递信号。布线有几种类型,包括设计用于专门实现 CLB 互联、器件内的高速水平和垂直长线以及时钟与其他全局信号的全局低偏移布线。

3. SelectIO(IOB)

目前的 FPGA 可支持多种 I/O 标准,例如多种多样的电平标准,因而为系统设计提供了理想的接口桥接。FPGA 内的 I/O 按 Bank 分组,如图 1-4 所示。每个 Bank 能独立支持不同的 I/O 标准。目前最先进的 FPGA 提供了十多个 I/O Bank,能够提供灵活的 I/O 支持。

4. 存储器

嵌入式模块 RAM 存储器在大部分 FPGA 中都可使用,这能为用户的设计实现片上存储。Xilinx FPGA 提供高达 10Mb 的片上存储(每个区块大小为 36Kb),能够支持真正的双端口运行。

5. 完整的时钟管理

目前大多数的 FPGA 都提供数字时钟管理(所有 Xilinx FPGA 都具有此特性)。Xilinx 推出的最高级 FPGA 提供了数字时钟管理和锁相环锁定功能,不仅提供了精确时钟综合功能,而且能够降低抖动和实现过虑。

6. 其他

除了上述的基本逻辑模块之外,目前的 Xilinx FPGA 还包括大量的硬件乘法器、数字信号处理单元,甚至是复杂的 PowerPC 硬核处理器和双核 ARM A9 处理器。

1.3 传统的 FPGA 开发基本流程

传统的 FPGA 设计流程就是利用 EDA 开发软件和编程工具对 FPGA 芯片进行开发的过程。典型 FPGA 的开发流程一般如图 1-5 所示,包括功能定义/器件选型、设计输入、功能仿真、综合优化、综合后仿真、布局与布线、时序仿真、板级仿真与验证以及芯片编程与调试等主要步骤。对于包含 ARM9 双核的 ZYNQ 器件的开发,图 1-5 对应它的可编程逻辑部分(PL)的开发流程,其完整的处理器系统(PS)的开发以及与 PL 的协同工作,则是本书后续章节要重点描述的过程。

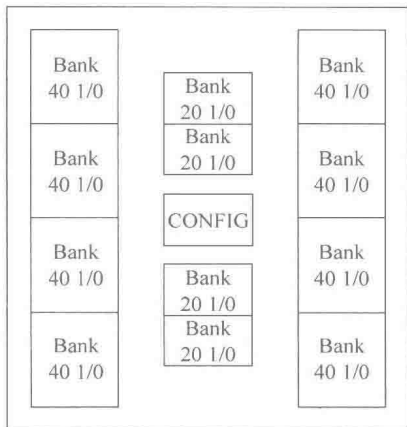


图 1-4 IOB 的图示

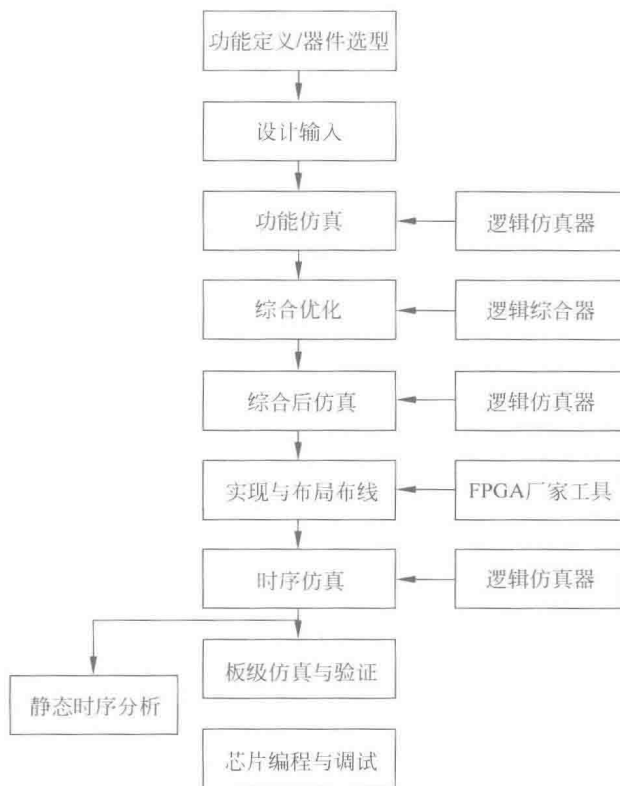


图 1-5 FPGA 开发的基本流程

1. 功能定义/器件选型

在FPGA设计项目开始之前,必须有系统功能的定义和模块的划分,另外就是要根据任务要求,如系统的功能和复杂度,对工作速度和器件本身的资源、成本以及连线的可布性等方面进行权衡,选择合适的设计方案和合适的器件类型。一般都采用自顶向下的设计方法,把系统分成若干个基本单元,然后再把每个基本单元划分为下一层次的基本单元,一直这样做下去,直到可以直接使用EDA元件库为止。

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(HDL)和原理图输入方式等。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。这种方法虽然直观并易于仿真,但效率很低,且不易维护,不利于模块构造和重用。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,在实际开发中应用最广的就是HDL语言输入法。利用文本描述设计,可以分为普通HDL和行为HDL。普通HDL有ABEL、CUR等,支持逻辑方程、真值表和状态机等表达方式,主要用于简单的小型设计。而在中大型工程中,主要使用行为HDL,其

主流语言是 Verilog HDL 和 VHDL。这两种语言都是美国电气与电子工程师协会(IEEE)的标准,其共同的突出特点是语言与芯片工艺无关,利于自顶向下设计,便于模块的划分与移植,可移植性好,具有很强的逻辑描述和仿真功能,而且输入效率很高。除了 IEEE 标准语言外,还有厂商自己的语言。也可以用 HDL 为主、原理图为辅的混合设计方式,以发挥两者的各自特色。

3. 功能仿真

功能仿真也称为前仿真,是在编译之前对用户所设计的电路进行逻辑功能验证,此时的仿真没有延时信息,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(即将所关心的输入信号组合成序列),仿真结果生成报告文件和输出信号波形,从中便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。常用的工具有 Model Tech 公司的 ModelSim、Synopsys 公司的 VCS 和 Cadence 公司的 NC-Verilog 以及 NC-VHDL 等软件。

4. 综合优化

所谓综合就是将较高级抽象层次的描述转化成较低层次的描述。综合优化根据目标与要求优化所生成的逻辑连接,使层次设计平面化,供 FPGA 布局布线软件进行实现。就目前的层次来看,综合(Synthesis)优化是指将设计输入编译成由与门、或门、非门、RAM、触发器等基本逻辑单元组成的逻辑连接网表,而并非真实的门级电路。真实具体的门级电路需要利用 FPGA 制造商的布局布线功能,根据综合后生成的标准门级结构网表来产生。

为了能转换成标准的门级结构网表,HDL 程序的编写必须符合特定综合器所要求的风格。由于门级结构、RTL 级的 HDL 程序的综合是很成熟的技术,所有的综合器都可以支持到这一级别的综合。常用的综合工具有 Synplicity 公司的 Synplify/Synplify Pro 软件以及各个 FPGA 厂家自己推出的综合开发工具。

5. 综合后仿真

综合后仿真检查综合结果是否和原设计一致。在仿真时,把综合生成的标准延时文件反标注到综合仿真模型中去,可估计门延时带来的影响。但这一步骤不能估计线延时,因此和布线后的实际情况还有一定的差距,并不十分准确。目前的综合工具较为成熟,对于一般的设计可以省略这一步,但如果在布局布线后发现电路结构和设计意图不符,则需要回溯到综合后仿真来确认问题之所在。在功能仿真中介绍的软件工具一般都支持综合后仿真。

6. 实现与布局布线

布局布线可理解为利用实现工具把逻辑映射到目标器件结构的资源中,决定逻辑的最佳布局,选择逻辑与输入/输出功能链接的布线通道进行连线,并产生相应文件(如配置文件与相关报告)。实现是将综合生成的逻辑网表配置到具体的 FPGA 芯片上,布局布线是其中最重要的过程。布局将逻辑网表中的硬件原语和底层单元合理地配置到芯

片内部的固有硬件结构上,并且往往需要在速度最优和面积最优之间做出选择。布线根据布局的拓扑结构,利用芯片内部的各种连线资源,合理正确地连接各个元件。

目前,FPGA 的结构非常复杂,特别是在有时序约束条件时,需要利用时序驱动的引擎进行布局布线。布线结束后,软件工具会自动生成报告,提供有关设计中各部分资源的使用情况。由于只有 FPGA 芯片生产商对芯片结构最了解,所以布局布线必须选择芯片开发商提供的工具。

7. 时序仿真

时序仿真,也称为后仿真,是指将布局布线的延时信息反标注到设计网表中来检测有无时序违规(即不满足时序约束条件或器件固有的时序规则,如建立时间、保持时间等)现象。时序仿真包含的延时信息最全,也最精确,能较好地反映芯片的实际工作情况。由于不同芯片的内部延时不一样,不同的布局布线方案也给延时带来不同的影响。因此在布局布线后,通过对系统和各个模块进行时序仿真,分析其时序关系,估计系统性能,以及检查和消除竞争冒险是非常必要的。在功能仿真中介绍的软件工具一般都支持综合后仿真。

8. 板级仿真与验证

板级仿真主要应用于高速电路设计中,对高速系统的信号完整性、电磁干扰等特征进行分析,一般都以第三方工具进行仿真和验证。

9. 芯片编程与调试

设计的最后一步就是芯片编程与调试。芯片编程是指产生使用的数据文件(Bitstream Generation,位数据流文件),然后将编程数据下载到FPGA芯片中。其中,芯片编程需要满足一定的条件,如编程电压、编程时序和编程算法等方面。逻辑分析仪(Logic Analyzer,LA)是FPGA设计的主要调试工具,但需要引出大量的测试引脚,且LA价格昂贵。

目前,主流的FPGA芯片生产商都提供了内嵌的在线逻辑分析仪(如Xilinx ISE中的ChipScope、Altera Quartus II中的SignalTap II以及SignalProb)来解决上述矛盾,它们只需要占用芯片少量的逻辑资源,具有很高的实用价值。

1.4 Xilinx FPGA 家族介绍

Xilinx 提供广泛的现场可编程门阵列(FPGA)芯片器件,为设计提供高级功能、低功耗、高性能及高价值。Xilinx 提供综合而全面的多节点产品系列,充分满足各种应用需求,其工艺结点如图 1-6 所示。例如采用业界一流 28nm HPL 工艺技术的 7 系列 All Programmable FPGA,可在优化性能价格与功耗比的同时,实现突破性性能、容量与系统集成度。Xilinx 20nm UltraScale 器件是该公司 Virtex 与 Kintex FPGA 以及 3D IC 系列的扩展部件,不但可提供前所未有的系统集成度,同时还支持 ASIC 类系统级性能。

目前市场上主流的 Xilinx FPGA 系列如表 1-1 所示。

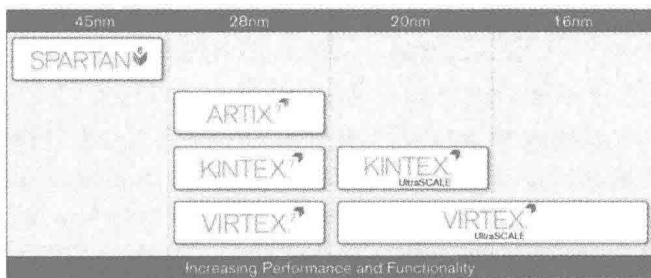


图 1-6 Xilinx FPGA 工艺结点

表 1-1 市场上主流的 Xilinx FPGA 系列

资源 \ 系列名称	Spartan-6	Artix-7	Kintex-7	Virtex-7	Kintex UltraScale	Virtex UltraScale
逻辑单元	147 443	215 360	477 760	1 954 560	1 160 880	4 407 480
BlockRAM/Mb	4.8	13	34	68	76	132.9
DSP Slice	180	740	1920	3600	5520	2880
DSP 性能(对称 FIR)/GMACs	140	930	2845	5335	8180	4268
收发器数量	5	16	32	96	64	120
收发器速度/(Gb/s)	3.2	6.6	12.5	28.05	16.3	32.75
总收发器带宽(全双工)/(Gb/s)	50	211	800	2784	2086	5886
存储器接口(DDR3)	800	1066	1866	1866	2400	2400
PCI Express 接口	x1 Gen1	x4 Gen2	x8 Gen2	x8 Gen3	x8 Gen3	x8 Gen3
模拟混合信号(AMS)/XADC	—	XADC	XADC	XADC	系统监视器	系统监视器
配置 AES	有	有	有	有	有	有
I/O 引脚	576	500	500	1200	832	1456
I/O 电压/V	1.2~3.3	1.2~3.3	1.2~3.3	1.2~3.3	1.0~3.3	1.0~3.3

因为经过三十年的发展，FPGA 的家族已十分庞大，故在此不一一列出。如果需要其他有关 Xilinx FPGA 系列的详细信息，例如 Spartan3、Virtex4/5/6 等，请到 Xilinx 网站查询 (http://china.xilinx.com/support/index.html/content/xilinx/zh/supportNav/silicon_devices.html)。有关表 1-1 中芯片的更详细信息，既可以在上述网址查询，也可以在安装了最新版本的 Xilinx FPGA 开发软件后，使用其中的 DocNav 软件进行智能查询与阅读。

1.5 Xilinx 开发工具与设计平台

1.5.1 ISE 与 Vivado、Vivado HLS 简介

早在 1997 年，Xilinx 就推出了 ISE 设计套件，界面如图 1-7 所示。ISE 套件采用了当时非常具有创新性的基于时序的布局布线引擎，并随着 FPGA 能够执行日趋复杂的功