



XILINX大学合作计划指定教材

Xilinx FPGA

设计与实践教程

■ 赵吉成 王智勇 编著



西安电子科技大学出版社
<http://www.xduph.com>

XILINX 大学合作计划指定教材

Xilinx FPGA 设计与实践教程

赵吉成 王智勇 编著

西安电子科技大学出版社

内 容 简 介

本书系统讲述了 FPGA 的软硬件开发知识,并以 Spartan-3 开发套件为硬件平台,配合经典的实例应用,使读者能够从硬件设计、软件开发和系统设计等方面系统掌握 FPGA 的使用方法。

本书共四篇 16 章。第一篇为 FPGA 设计基础与 ISE 开发基本流程,共 2 章,内容包括 PLD 技术基础, Xilinx FPGA 的开发、仿真以及实现整个流程。第二篇为数字电路设计基础与 VerilogHDL 描述,共 5 章,介绍了基于 VerilogHDL 的数字电路基础、同步电路设计思想和高级技巧。第三篇为基于 FPGA 的接口开发,共 5 章,结合 Xilinx FPGA 开发板,详细讲述了 UART 串口通信控制器、PS/2 键盘/鼠标接口控制器、VGA 图形图像显示控制器以及 RAM 接口控制器等案例的设计、开发以及验证。第四篇为基于 FPGA 的软核微控制器 PicoBlaze,共 4 章,以 PicoBlaze 为例,介绍了 PicoBlaze 微处理器的软硬件开发、中断设计等。

本书可作为电子类、计算机类、自动化类等相关专业研究生和高年级本科生教材或参考书,也可作为数字电路设计人员以及 FPGA 爱好者的参考书。

图书在版编目(CIP)数据

Xilinx FPGA 设计与实践教程 / 赵吉成,王智勇编著. —西安:西安电子科技大学出版社,2012.1

XILINX 大学合作计划指定教材

ISBN 978-7-5606-2629-1

I. ① X… II. ① 赵… ② 王… III. ① 可程序逻辑器件—系统设计—教材
IV. ① TP332.1

中国版本图书馆 CIP 数据核字(2011)第 138665 号

策 划 戚文艳

责任编辑 夏大平 戚文艳

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子信箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2012 年 1 月第 1 版 2012 年 1 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 26.5

字 数 624 千字

印 数 1~2000 册

定 价 52.00 元

ISBN 978-7-5606-2629-1 / TP · 1292

XDUP 2921001-1

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜,谨防盗版。

序

FPGA 技术自 20 世纪 80 年代诞生以来，逐渐从粘合逻辑不断深入到数字信号处理、高速计算、嵌入式等领域，并蚕食着 ASIC、DSP 以及嵌入式微处理器的市场，成为当前最为流行和应用最为广泛的数字系统设计平台之一。Xilinx 作为全球高端 FPGA 领导厂商，在 FPGA 的芯片设计与 EDA 工具开发方面锐意创新，不断为客户提供更为高效的体验，其产品应用广泛。作为学生或工程师，只有在熟悉 Xilinx FPGA 的软硬件开发流程的基础上，掌握基于 FPGA 的数字电路设计技巧，并通过大量的实践训练来提升数字设计的能力，才能更好适应不断革新的数字电子技术发展的需求。

随着 Xilinx 对 FPGA 技术的不断创新，广大工程师和研究人员迫切希望加入到研究 FPGA 技术的行列当中。Xilinx 公司通过大学计划、联合实验室、开源社区、创新大赛等方式，在广大研发人员和在校学生中推广 FPGA 技术。但是市面上一直以来缺少一本从实践出发，能够有效提升研发人员实际能力的教材，《Xilinx FPGA 设计与实践教程》恰好可以弥补这一点。

本书整体而言，有以下三个特色：

(1) 内容非常丰富。全书覆盖了 FPGA 开发的整个流程、基于 Verilog 语言的数字电路设计、经典 FPGA 案例设计与验证、基于源码开放的微处理器 PicoBlaze 软硬件开发四个部分，富含 120 多个例程和近 100 道思考与练习题，对于提升读者的设计与实践能力有着实质性的帮助。

(2) 描述精辟透彻。全书知识点讲述中，能够从深层次剖析知识的本质，由浅入深引领读者理解数字设计的关键知识点，如时序电路基本模型、带数据路径状态机和同步电路设计方法等，其理论剖析深入而透彻。

(3) 经典实用。书中结合 Xilinx 的开发板讲述了 UART 串口通信、PS/2 键盘/鼠标接口控制器、VGA 显示接口控制以及 SRAM 控制器等富有代表性的案例的设计、仿真以及验证，所有案例采用统一的设计思想和代码风格，使读者在牢固巩固数字设计基本方法的同时，从工程应用方面不知不觉上升一个高度。

本书写作上的亮点在于能够通过简单的实例引申出可编程逻辑设计的高级技巧和思想，通过巧妙的实践设计帮助读者理解和消化高深的理论，并养成良好的设计习惯。我郑重地向大家推荐这本书，使更多的读者能够早日掌握 FPGA 的开发技能，促进 FPGA 技术的进一步推广。

赛灵思(Xilinx, Inc)中国区大学计划经理
谢凯年博士
2011 年 4 月

前 言

FPGA 自 1984 年由 Xilinx 公司发明以来, 逐渐引起了全球集成电路格局的变化。从一开始仅仅实现粘合逻辑到后来的数字信号处理以及目前的系统级解决方案, FPGA 不断“蚕食”着微处理器、DSP 以及 ASIC 的市场, 成为目前发展最为迅猛的热门技术之一。在中国, FPGA 技术引入较晚, 21 世纪初才逐渐在通信、图像等个别领域开始使用, 再加上我国技术人员对数字设计的积累较少, 目前市场上关于 FPGA 设计的书籍大部分为片面的软件工具介绍或者 HDL 语言介绍, 基本上缺乏系统性和实践性。本书结合作者多年 FPGA 设计的实践经验以及在 FPGA 数字电路设计方面的培训经验, 本着“在实践中学习”的学习理念, 以“快速掌握数字电路设计技术”为目的, 结合大量实践例程, 由浅入深、循序渐进地讲述了基于 VerilogHDL 的数字设计和常用接口设计与嵌入式软核微处理器 PicoBlaze 的软硬件开发。

本书适合从事 FPGA 或者 ASIC 开发的工程师以及电子相关专业的研究生和高年级本科生使用。本书共四篇。前三篇要求读者具有一定的数字电路基本知识; 对于第四篇, 读者具有一定的汇编语言基础会更好。

本书的亮点

(1) 本书专注于数字电路的硬件设计方法和技巧, 而并非一部 VerilogHDL 语法手册或者 FPGA 开发工具手册。在内容上包含了大量的数字电路设计模块框架, 既可方便用户掌握典型应用电路的设计思想和方法, 又可使之顺利地套用在自己的设计当中。我们关注于让读者掌握一种编程框架, 然后灵活地应用于多个程序中, 而不是掌握多种方法来描述同一个电路。这也是本书实践性的体现。

(2) 在描述语言上, 首次采用 Verilog HDL-2001 标准进行讲解。该标准相对于 Verilog HDL-1995, 有很多增强的地方, 不管是对形成良好的代码风格, 还是对于综合工具的更好支持, 都有很多优越之处, 所以非常有必要让数字设计工程师们掌握。

(3) 所有实践例程都有很强的兼容性, 具体体现在以下四点:

① 方便应用到更大型的设计当中, 因为每个接口开发例程都包含了良好的接口开发功能, 避免读者学会很多小的例程, 却无法设计大的系统的尴尬;

② 支持与系统其它接口良好的匹配;

③ 可以应用在 ASIC 设计或者不同型号的 FPGA 设计当中;

④ 支持采用不同的综合工具进行综合。

(4) 总结了工程应用中 HDL 设计中的难点, 并结合多家著名公司的经典数字设计面试题内容, 分析了同步电路设计中关于时钟、接口、时序方面的处理方法以及优秀的 HDL 代码风格编码方法, 有利于读者理解数字设计的难点和重点。

(5) 本书基于“XILINX 大学合作计划”提供的 Spartan-3 系列开发套件设计, 支持的

开发板包括 Spartan-3 Starter、Nexys-2 以及 Basys 开发板等，也支持具有相关接口的其它 Xilinx 公司的 FPGA 开发板。

内容简述

本书共四篇 16 章，内容包括 FPGA 设计基础与 ISE 开发基本流程、数字电路设计基础与 VerilogHDL 描述、基于 FPGA 的接口开发和基于 FPGA 的软核微控制器 PicoBlaze。

第一篇为 FPGA 设计基础与 ISE 开发基本流程，介绍了 FPGA 的基本概念、可编程器件编程技术和现代 FPGA 通用结构，并以 ISE12.1 为平台介绍了 FPGA 开发基本流程。本篇共 2 章，即第一、二章。

第一章为 FPGA 设计基础，介绍了 FPGA 基本概念、可编程技术发展演变以及 FPGA 技术应运而生的历史过程，以 Spartan-3 系列 FPGA 为例介绍了现代 FPGA 基本结构。

第二章为 ISE12.1 开发环境与 S3 开发板，借助一个简单的例程详细介绍了 ISE12.1 的开发流程，包括设计输入、仿真、约束、综合、实现和编程的整个流程，并介绍了 Synplify9.2 和 ModelSim6.5 的使用，简单介绍了 S3 开发板的结构。

第二篇为数字电路设计基础与 VerilogHDL 描述，介绍了 VerilogHDL 基础语法以及基本数字电路的 HDL 描述和对应硬件结构、组合逻辑和时序逻辑的 VerilogHDL 描述、常用状态机描述和同步电路数字设计原则及 VerilogHDL 常见难点解析。本篇内容涉及 Verilog 语言以及同步电路设计的方方面面，为本书重点掌握的内容。本篇共 5 章，即第三~七章。

第三章为 VerilogHDL 语言基础，介绍了 VerilogHDL 编程框架和基本的语法，以简单的组合逻辑电路描述，帮助读者理解 VerilogHDL 硬件描述语言的特性。

第四章为组合逻辑设计，介绍了 VerilogHDL 语言相关操作符以及数据传输语句描述，以比较器、加法器、多路选择器等基本电路为例，帮助读者理解组合逻辑电路的各种数据描述方法。

第五章为时序逻辑设计，介绍了规则时序逻辑电路的基本模型、基本组成单元等，并结合大量实例阐述了时序电路的设计要则以及基本设计框架。

第六章为时序状态机设计，介绍了 Moore 和 Mealy 状态机的状态图描述和 HDL 描述框架，并且以大量实例介绍了带数据路径(FSMD)状态机的设计方法。

第七章为数字电路设计原则与 VerilogHDL 难点解析，重点为同步电路基本模型和设计原则、异步电路的同步设计原则，并针对 VerilogHDL 语言重点介绍了阻塞和非阻塞赋值的区别以及优秀 HDL 代码设计风格，最后为 TestBench 平台的基本创建。

第三篇为基于 FPGA 的接口开发，结合 S3 开发板重点介绍了基于 FPGA 的 UART 接口、PS/2 键盘接口、PS/2 鼠标接口、异步 RAM 控制器、VGA 图形显示控制器和接口开发，重点训练基于 VerilogHDL 的设计与验证等实践能力。本篇共 5 章，即第八~十二章。

第八章为 UART 串口通信控制器，介绍了 UART 传输和发送接口模块的 HDL 设计和验证方法。

第九章为 PS/2 键盘接口控制器，介绍了基于 S3 开发板 PS/2 接口的键盘输入接口控制。

第十章为 PS/2 鼠标接口控制器，介绍了基于 S3 开发板 PS/2 接口的鼠标输入、输出接口控制。

第十一章为 RAM 接口控制器，介绍了基于 IS61LV25616AL 的异步 SRAM 接口控制器

设计和 Xilinx Spartan-3 内部存储器的设计。

第十二章为 VGA 图形图像显示控制器，介绍了基于 S3 开发板的图形图像显示开发。

第四篇为基于 FPGA 的软核微控制器 PicoBlaze，介绍了 PicoBlaze 的结构、软硬件开发流程、软件开发指令以及中断和接口设计。本篇共 4 章，即第十三~十六章。

第十三章为基于 Xilinx FPGA 的微处理器，介绍了 PicoBlaze 的硬件和指令结构。

第十四章为 PicoBlaze 汇编语言开发，介绍了基本的汇编语言和 PicoBlaze 的开发流程。

第十五章为 PicoBlaze 接口开发，介绍了 PicoBlaze 接口特性以及硬件接口定制设计方法。

第十六章为 PicoBlaze 中断，介绍了 PicoBlaze 硬件中断接口和软件中断处理方法。

致谢

全书由赵吉成编写，王智勇参与了本书第四篇的编写。

本书在编写过程中，参考了大量的文献和著作，这里向其作者表示深切的谢意。Xilinx 公司大学计划总经理谢凯年先生为本书编著者提供了硬件平台和很多宝贵的建议，促成了本书的出版并欣然作序；西安电子科技大学出版社戚文艳和夏大平两位老师为本书的修改和编辑付出了辛苦劳动，并提出了很多宝贵意见。感谢他(她)们以及所有为本书出版作出贡献的人。

最后感谢我的妻子在怀孕期间还坚持帮我修改书稿；感谢我的家人，在他们的鼓励下我才能够完成本书。

编者
2011 年 4 月

目 录

第一篇 FPGA 设计基础与 ISE 开发基本流程

第一章 FPGA 设计基础 1	2.1.1 ISE12.1 套件分类..... 26
1.1 FPGA 的基本概念..... 1	2.1.2 ISE12.1 功能介绍..... 27
1.2 可编程逻辑技术发展简介..... 2	2.1.3 ISE12.1 用户界面和菜单操作..... 27
1.2.1 可编程技术发展演变过程..... 2	2.2 S3 开发板简介..... 28
1.2.2 FPGA 技术..... 5	2.3 ISE 开发流程..... 29
1.3 FPGA 器件编程技术..... 6	2.3.1 创建工程和设计输入..... 30
1.3.1 熔丝互连编程技术..... 6	2.3.2 创建 TestBench 并进行 RTL 仿真... 33
1.3.2 基于反熔丝的编程技术..... 8	2.3.3 添加约束..... 36
1.3.3 基于 SRAM 的可编程技术..... 10	2.3.4 综合与实现..... 37
1.3.4 基于 FLASH 或 E ² PROM 的 可编程技术..... 11	2.3.5 生成配置文件并对 FPGA 进行配置..... 38
1.4 通用 FPGA 的构成结构..... 11	2.4 第三方开发工具..... 40
1.4.1 现代 FPGA 的基本逻辑单元..... 12	2.4.1 ModelSim 介绍..... 40
1.4.2 Xilinx Spartan-3 FPGA 的 基本结构..... 14	2.4.2 在 ModelSim 中编译 Xilinx 的 器件库..... 41
1.5 Xilinx FPGA 的开发流程..... 20	2.4.3 ModelSim 功能仿真举例..... 42
1.6 FPGA 技术的未来发展..... 23	2.4.4 Synplify Pro 介绍..... 47
本章小结..... 25	2.4.5 关联 ISE 和 Synplify Pro..... 47
思考与练习..... 25	2.4.6 Synplify Pro 使用流程..... 48
第二章 ISE12.1 开发环境与 S3 开发板 26	本章小结..... 52
2.1 ISE12.1 软件综述..... 26	思考与练习..... 52

第二篇 数字电路设计基础与 VerilogHDL 描述

第三章 VerilogHDL 语言基础 53	3.2.1 模块的结构..... 54
3.1 基本知识介绍..... 53	3.2.2 编程框架..... 56
3.2 模块结构和编程框架..... 54	3.3 数据类型和基本元素..... 58

3.3.1	基本概念	58
3.3.2	基本数据类型值	59
3.3.3	数据类型	59
3.3.4	常量	61
3.4	结构化描述	62
3.5	TestBench 简介	64
	本章小结	66
	思考与练习	67
第四章	组合逻辑设计	68
4.1	基本操作符	68
4.1.1	算术操作符	69
4.1.2	移位操作符	70
4.1.3	关系运算符与相等运算符	70
4.1.4	位操作、复制和逻辑操作运算符	70
4.1.5	连接与复制运算符	72
4.1.6	条件运算符	72
4.1.7	位宽调整操作	73
4.1.8	关于 Z 和 X 的综合	74
4.2	组合逻辑描述	75
4.2.1	使用 always 模块描述组合逻辑	75
4.2.2	使用赋值语句描述组合逻辑	76
4.2.3	举例说明	76
4.3	条件控制语句	78
4.3.1	if-else 语句	78
4.3.2	case 语句	80
4.3.3	casez 和 casex 语句	82
4.3.4	“full case”和“parallel case” 语句	83
4.4	条件控制语句的布线结构	84
4.4.1	优先级布线网络	84
4.4.2	多路选择布线网络	85
4.5	always 语句的编程指导	86
4.6	工程实践	89
4.6.1	十六进制数到七段数码管译码器	89
4.6.2	带符号加法器设计	92
4.6.3	桶形移位器设计	95
	本章小结	96
	思考与练习	97

第五章	时序逻辑设计	98
5.1	时序电路基础	98
5.1.1	时序电路基本存储单元	98
5.1.2	同步时序电路	99
5.1.3	时序电路分类	100
5.2	时序电路基本单元的 HDL 描述	101
5.2.1	D 触发器	101
5.2.2	寄存器	104
5.2.3	寄存器文件	104
5.3	简单例程	105
5.3.1	移位寄存器	105
5.3.2	二进制计数器和变量	107
5.4	时序电路的 TestBench	111
5.5	工程实践	114
5.5.1	LED 时序动态选择电路	114
5.5.2	秒表设计	122
5.5.3	FIFO 缓冲器设计	126
	本章小结	131
	思考与练习	131
第六章	时序状态机设计	133
6.1	有限状态机	133
6.1.1	Moore 和 Mealy 状态机	133
6.1.2	有限状态机的描述方式	133
6.1.3	有限状态机的 HDL 开发	135
6.2	状态机设计实例	138
6.2.1	上升沿检测电路	138
6.2.2	按键防抖动电路	143
6.2.3	电路硬件验证	147
6.3	带数据路径的状态机(FSMD)	149
6.3.1	简单寄存器传输操作	149
6.3.2	FSMD 状态描述	150
6.3.3	FSMD 的模块框图	151
6.4	FSMD 的 HDL 代码开发	152
6.4.1	基于 FSMD 描述的按键 防抖动电路	152
6.4.2	显性描述数据路径	153
6.4.3	隐含描述数据路径	156
6.5	设计举例	158

6.5.1 斐波纳契序列(Fibonacci Number)		7.3.2 组合逻辑电路中的赋值描述.....	186
实现电路.....	158	7.3.3 时序电路赋值描述.....	187
6.5.2 频率检测器设计.....	162	7.3.4 时序电路中的混合赋值.....	188
6.5.3 除法电路设计.....	165	7.4 优秀 HDL 代码风格.....	191
本章小结.....	169	7.4.1 代码风格的含义.....	191
思考与练习.....	169	7.4.2 通用代码风格.....	191
第七章 数字电路设计原则与		7.4.3 Xilinx 芯片专用代码风格.....	195
VerilogHDL 难点解析	171	7.4.4 推荐时序电路描述代码风格.....	196
7.1 时序电路基础.....	171	7.5 TestBench 编写.....	199
7.1.1 同步电路的时序分析.....	173	7.5.1 基于 HDL 的 TestBench 编写.....	199
7.1.2 异步电路和同步电路的区别.....	174	7.5.2 always 和 initial 模块.....	199
7.1.3 同步时序设计规则.....	175	7.5.3 顺序执行语句.....	200
7.2 异步电路中的同步处理方法.....	175	7.5.4 时序控制语句.....	201
7.2.1 时钟的同步处理.....	175	7.5.5 系统函数和任务.....	203
7.2.2 接口电路处理.....	177	7.5.6 用户自定义函数和任务.....	208
7.2.3 全局信号处理.....	182	7.5.7 TestBench 举例.....	211
7.3 阻塞赋值与非阻塞赋值.....	184	本章小结.....	218
7.3.1 概述.....	185	思考与练习.....	218

第三篇 基于 FPGA 的接口开发

第八章 UART 串口通信控制器	219	第九章 PS/2 键盘接口控制器	237
8.1 UART 传输系统.....	219	9.1 PS/2 基础.....	237
8.2 UART 接收模块设计.....	220	9.1.1 PS/2 端口的物理接口.....	237
8.2.1 设计方案.....	220	9.1.2 PS/2 接口主从设备通信协议.....	237
8.2.2 波特率产生器.....	221	9.1.3 PS/2 接收模块设计.....	238
8.2.3 UART 接收器.....	221	9.2 PS/2 键盘扫描设计.....	242
8.2.4 接口电路.....	225	9.2.1 关于键盘扫描编码.....	242
8.3 UART 发送模块设计.....	228	9.2.2 按键扫描电路设计.....	243
8.4 UART 系统的总结.....	231	9.3 PS/2 键盘接口电路.....	246
8.4.1 完整 UART 系统.....	231	9.3.1 接口电路设计.....	246
8.4.2 UART 验证电路.....	233	9.3.2 接口电路验证.....	248
8.4.3 Windows 的超级终端.....	234	本章小结.....	251
8.4.4 定制 UART.....	235	思考与练习.....	251
本章小结.....	236	第十章 PS/2 鼠标接口控制器	253
思考与练习.....	236	10.1 PS/2 鼠标接口电路.....	253

10.1.1 关于鼠标.....	253	11.3.3 选择设计 II	291
10.1.2 鼠标 PS/2 通信协议	253	11.4 Xilinx Spartan-3 内部存储器	292
10.1.3 初始化过程	254	11.4.1 概述	292
10.2 PS/2 传输子系统设计	254	11.4.2 利用 CoreGenerator 定制嵌入式 RAM 模块	292
10.2.1 主系统对 PS/2 设备的 通信协议	254	11.5 Xilinx 嵌入式存储器例化举例	293
10.2.2 设计与编码	255	11.5.1 单端口 RAM	293
10.3 PS/2 鼠标数据传输系统	260	11.5.2 双端口 RAM	295
10.3.1 双向传输 PS/2 接口电路设计	260	11.5.3 ROM	297
10.3.2 双向传输 PS/2 验证电路	262	本章小结	299
10.4 PS/2 鼠标数据接口电路	265	思考与练习	299
10.4.1 传输 PS/2 接口电路设计	265		
10.4.2 传输 PS/2 接口电路测试	268		
本章小结	270		
思考与练习	270		
第十一章 RAM 接口控制器	271		
11.1 关于 IS61LV25616AL SRAM	271	第十二章 VGA 图形图像显示 控制器	301
11.1.1 芯片介绍以及 I/O 接口	271	12.1 CRT 显示器原理	301
11.1.2 时序参数	272	12.1.1 CRT 显示的基本原理	301
11.2 基本存储控制器	274	12.1.2 视频显示基本术语	302
11.2.1 设计框图	274	12.1.3 S3 开发板上的 VGA 端口	303
11.2.2 时序要求	276	12.1.4 VGA 视频控制器	303
11.2.3 存储器文件与 SRAM 的对比	276	12.2 VGA 同步电路	304
11.2.4 设计安全性	276	12.2.1 水平同步	304
11.2.5 ASMD 状态机图	276	12.2.2 垂直同步	305
11.2.6 时序分析	277	12.2.3 HDL 实现	306
11.2.7 HDL 代码设计	278	12.2.4 测试电路	308
11.2.8 基本测试电路	281	12.3 像素产生电路	309
11.2.9 完整的 SRAM 测试电路	283	12.3.1 矩形图形显示	311
11.3 更加完善的设计	289	12.3.2 非矩形目标显示	316
11.3.1 异步 SRAM 的时序信息	289	12.3.3 动态目标显示	317
11.3.2 选择设计 I	290	12.4 位图显示方案	324
		12.4.1 采用双端口 RAM 实现	324
		12.4.2 采用单端口 RAM 实现	328
		本章小结	328
		思考与练习	329

第四篇 基于 FPGA 的软核微控制器 PicoBlaze

第十三章 基于 Xilinx FPGA 的 微处理器	331	13.1 PicoBlaze 架构介绍	331
		13.1.1 微处理器的应用	333

13.1.2 PicoBlaze 处理器的特点	333	思考与练习	371
13.1.3 顶层 HDL 模型	335		
13.1.4 设计流程	335	第十五章 PicoBlaze 接口开发	372
13.2 指令设置	336	15.1 输出端口	372
13.2.1 编程模型	337	15.1.1 输出指令和时序	372
13.2.2 指令格式	337	15.1.2 输出接口	373
13.2.3 逻辑指令	338	15.2 输入端口	375
13.2.4 算术指令	339	15.2.1 输入指令和时序	375
13.2.5 比较和测试指令	339	15.2.2 输入接口	375
13.2.6 移位和循环指令	340	15.3 求平方和电路接口开发	377
13.2.7 数据传输指令	341	15.3.1 输出接口	377
13.2.8 程序流程控制指令	342	15.3.2 输入接口	379
13.2.9 中断相关指令	344	15.3.3 汇编程序设计	380
13.2.10 KCPSM3 汇编宏命令	345	15.3.4 HDL 程序开发	389
13.3 PicoBlaze 文件结构	346	本章小结	392
本章小结	349	思考与练习	392
思考与练习	349		
第十四章 PicoBlaze 汇编语言开发	350	第十六章 PicoBlaze 中断	393
14.1 PicoBlaze 汇编基础	350	16.1 PicoBlaze 中断处理机制	393
14.1.1 KCPSM3 语法规则	350	16.1.1 软件中断处理过程	393
14.1.2 位操作	350	16.1.2 中断时序描述	394
14.1.3 多字节操作	351	16.2 外部中断接口	395
14.1.4 常用控制语句结构的 汇编语言描述	352	16.2.1 单个中断请求	395
14.2 子程序开发	355	16.2.2 多个中断请求	395
14.3 PicoBlaze 汇编程序开发	356	16.3 软件开发	396
14.3.1 开发流程	356	16.3.1 中断处理主程序	396
14.3.2 程序举例	357	16.3.2 中断服务程序	397
14.3.3 说明文档与注释	363	16.4 设计举例	397
14.4 PicoBlaze 软件开发流程	365	16.4.1 中断接口	397
14.4.1 使用 KCPSM3 编译	365	16.4.2 中断服务子程序开发	397
14.4.2 使用 PBlazeIDE 仿真	366	16.4.3 汇编程序开发	398
14.4.3 使用 JTAG 接口下载代码	369	16.4.4 HDL 代码开发	405
14.4.4 代码综合	369	本章小结	409
本章小结	371	思考与练习	409
		参考文献	410

第一篇

FPGA 设计基础与 ISE 开发基本流程

第一章 FPGA 设计基础

FPGA(Field Programmable Gate Array)中文译名为现场可编程逻辑门阵列。它作为一种可编程逻辑器件,在 20 世纪 90 年代获得了突飞猛进的发展。在微电子技术的不断推动下,FPGA 在规模上进一步提高,在功耗方面进一步降低,其应用更加广泛,在通信、广播视讯、工业、医疗与自动化等领域成为发展最快的技术之一,也成为目前应用最为广泛的数字系统的主流平台之一。

本章主要从三个方面介绍 FPGA:首先,详细介绍 FPGA 的相关概念以及 FPGA 在数字系统中的主要应用;其次,介绍 FPGA 相关背景知识,使读者容易理解半导体工艺的发展与可编程设计技术发展的密切联系规律,进一步明确使用 FPGA 进行数字设计的优越性;最后,详细介绍通用 FPGA 的内部结构,使读者能够从深层次了解 FPGA。

1.1 FPGA 的基本概念

顾名思义,现场可编程逻辑门阵列(FPGA)是由可编程配置的逻辑块组成的数字集成电路,这些逻辑块之间有着丰富的可配置的互连资源,设计者可以通过对这些资源进行不同的配置和编程来达到自己所要实现的目标。我们要从如下几个方面来理解 FPGA 的特性。

首先,FPGA 名称中的“现场可编程”是指编程“在现场”进行(与那些内部功能已被制造商固化的器件正相反)。这意味着 FPGA 的编程具有更强的灵活性和创新性,我们可以在实验室进行配置,或者可以对已经应用于实际的电子系统中的某些功能进行改进,或者

可以根据用户需求, 实现新的协议或者标准来对当前应用作进一步的完善和改进。总而言之, FPGA 的现场可编程特性满足了用户实现任意数字逻辑的愿望, 成为用户灵活“武装”自己产品的最有效的武器。

其次, FPGA 名称中的“逻辑门阵列”不仅仅指的是传统意义上的逻辑门阵列。FPGA 是可编程逻辑器件(PLD)和专用集成电路(ASIC)技术发展 to 一定程度的产物。PLD 能够实现灵活的逻辑可编程功能, 但是其可编程规模小, 无法实现复杂的逻辑功能, 而 ASIC 虽然能够实现复杂的逻辑功能, 但是昂贵的工艺过程和巨额的流片费用, 在很多时候令大家望而却步。FPGA 的诞生恰好弥合了 PLD 和 ASIC 之间的这道鸿沟, 其逻辑规模可以达到 ASIC 的级别, 而且不必承担如 ASIC 开发带来的数额巨大的不可重现工程(NRE)成本。随着 FPGA 技术的不断发展, FPGA 器件逐渐变成一种数字化平台系统, 其“逻辑门”已经不再和 PLD 或者 ASIC 一样, 而是包含了现代 FPGA 中的各种资源, 如可编程逻辑块、RAM 资源、数字信号处理模块、微处理器等(1.4 节将详细介绍 FPGA 的结构)。

1.2 可编程逻辑技术发展简介

自 20 世纪 80 年代 Xilinx 公司首创 FPGA 技术以来, 可编程逻辑器件得到了飞速的发展。FPGA 已成为目前数字系统的主流平台之一。下面从可编程器件的基本概念、工艺演变过程和新技术开拓等方面阐述 FPGA 技术在现代数字系统中应用的必然, 以及起到的历史革命性作用。

1.2.1 可编程技术发展演变过程

世界上第一款可编程逻辑器件是 1970 年以 PROM 的形式进入人们视野的, 但当时还非常简单, 仅仅到 20 世纪 70 年代末, 复杂实用的 PLD 器件便应用在工程当中了。为了在复杂程度上进行区分, 后来出现了新的名词即简单可编程逻辑器件(SPLD)和复杂可编程逻辑器件(CPLD)。SPLD 至今依然有人沿用, 而 CPLD 成为当今 PLD 器件的代名词。PLD 器件经历了在结构上的不断改进, 从一开始的 PROM 器件, 到 PLA、PAL, 再到后来的 GAL, 逐渐演变到今天通用的 CPLD 结构。

1. PROM 器件

PROM(Programmable Read-Only Memory, 可编程只读存储器)基本结构其实就是由与(AND)阵列函数驱动可编程的或(OR)阵列函数。一个 3 输入 3 输出的基于 PROM 结构的可编程逻辑器件结构如图 1-1 所示。图中, “&”代表逻辑“与”; “!”代表逻辑“非”。

在 OR 门阵列中的可编程连线可以用熔丝、EPROM 晶体管或者 E²PROM 器件中的 E²PROM 晶体管等来实现。PROM 器件可以用来实现任何组合逻辑块, 但是它无法实现太多的输入和输出。PROM 器件最初主要作为存储器来存放计算机程序和常数值, 工程师也发现它可以用来实现简单的逻辑功能, 比如状态机查找表等。随着 PROM 的大量应用, 其他在其基础上改进的可编程器件也纷纷面世。

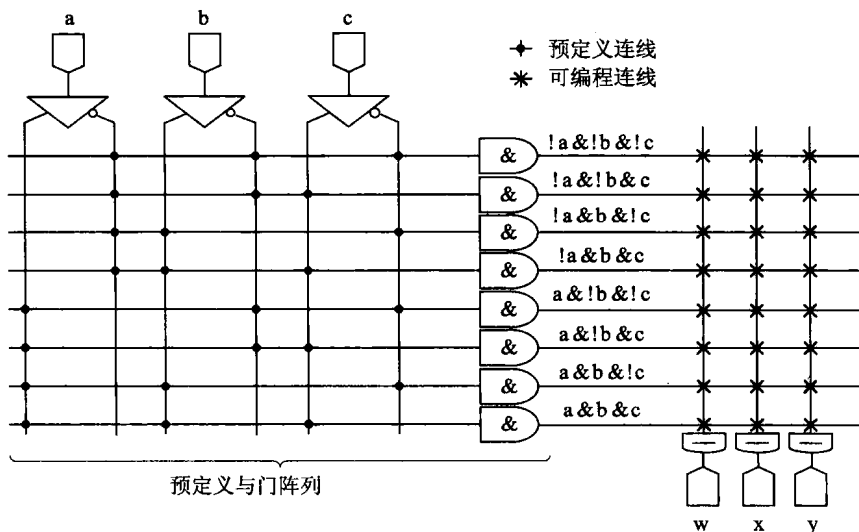


图 1-1 3 输入 3 输出的基于 PROM 结构的可编程逻辑器件结构图

2. PLA 器件

由于 PROM 器件对地址有限制, 可编程器件的下一步演化就是 PLA(Programmable Logic Array, 可编程逻辑阵列)器件。PLA 器件是 可编程逻辑器件中用户可配置性最好的, 因为它的 AND 和 OR 阵列都是可配置的。正是由于 AND 阵列也可编程, 因而 AND 阵列中的 AND 函数的数目便可以与器件的输入数目独立, 只要引入更多的行, 便可在阵列中形成额外的 AND 函数。类似地, OR 阵列也是与 AND 阵列独立的, 引入更多的列就可以形成更多的列函数。如果我们要用 PLA 器件完成下面三个公式, 则可以按图 1-2 所示的连线方式进行编程:

$$w = (a \& c) | (!b \& !c) \tag{1-1}$$

$$x = (!a \& !b \& !c) | (!b \& !c) \tag{1-2}$$

$$y = !a \& !b \& !c \tag{1-3}$$

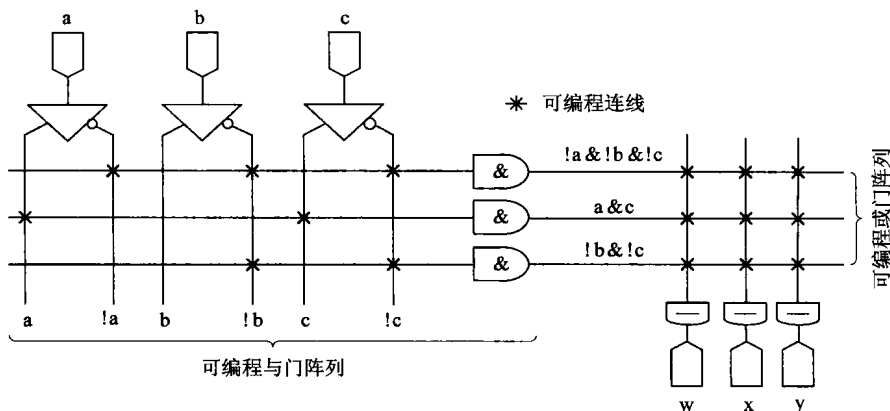


图 1-2 PLA 编程示意图

PLA 的优点是它对于大型设计非常有用，因为它可以实现大量公共乘积项，可用于多个输出。而 PLA 的缺点是信号通过可编程连线所花费的时间相对更长，所以整个器件的速度受到很大的影响。

3. PAL 器件

PAL(Programmable Array Logic, 可编程阵列逻辑)器件便是为了解决 PLA 的速度问题而产生的，其结构示意图如图 1-3 所示。

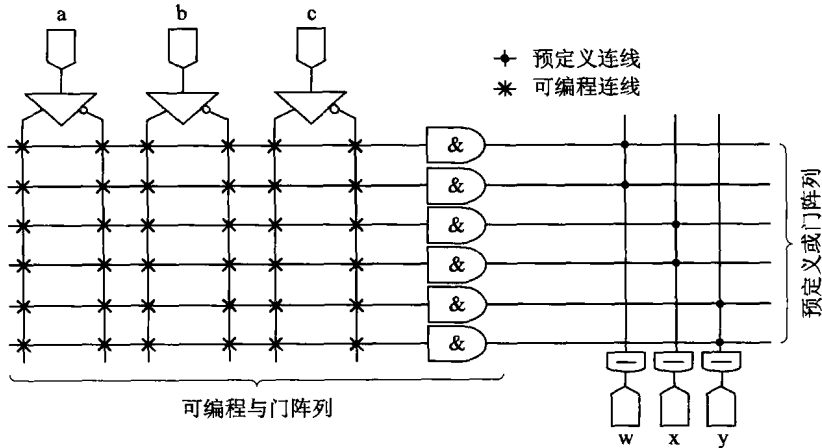


图 1-3 PAL 结构示意图

从图 1-3 中可以看到，PAL 的结构与 PROM 的正好相反，体现在 PAL 是由一个可编程 AND 阵列和一个预定义的 OR 阵列组成的，相对于 PLA 器件速度要快得多，但是它只允许有限数量的乘积项相或，对器件的应用灵活性又一次进行制约。要解决这些问题，需要跳出 PROM 器件的阴影，不再针对 PROM 器件做结构上简单的改进，而是采用新的方法，也就是下面我们讨论的 GAL 器件。

4. GAL 以及 CPLD 器件

GAL(Generic Array Logic, 通用逻辑阵列)器件是 Lattice 公司于 1985 年推出的新型的可编程逻辑器件，GAL 器件的输出端不再是简单采用或阵列实现，而是采用了逻辑宏单元 (OLMC)，通过编程可以将 OLMC 设置成不同的输出方式。这样，采用同一型号的 GAL 器件就可以实现 PAL 器件所有的输出电路工作模式，使 GAL 器件成为通用可编程逻辑器件。

GAL 系列器件诞生之后很长时间受到工程师的青睐，其在数字系统中的粘合逻辑功能方面，对原来传统意义上的 74 系列器件提出了挑战。GAL 器件不仅在性能上有很大的提高，而且还附加了很多独有的功能。比如，电子标签，方便了用户的文档管理；加密单元，防止他人抄袭电路；采用高性能的 E²COMS 工艺，保证了 GAL 器件的高速度和低功耗等。但是 GAL 器件依然属于低密度器件，其规模还是比较小，仅相当于几十个门电路。

真正的可编程时代的到来应该是伴随着 CPLD(Complex Programmable Logic Device, 复杂可编程逻辑器件)的诞生。在 20 世纪 90 年代前后，目前世界著名的可编程逻辑器件公司如 Xilinx、Altera、Lattice 等都争相研究新型的复杂可编程逻辑器件。CPLD 一般都是基于乘积项结构的，如 Xilinx 公司的 XC9500、CoolRunner II 系列器件，Altera 的 MAX7000、

MAX3000 以及 MAX-II 系列器件, Lattice 的 ispMACH4000、ispMACH5000 系列器件等, 都是基于乘积项的 CPLD。CPLD 采用的理念是一个普通的器件中包括一定数量的基本逻辑块, 分享一个公共的可编程互连矩阵。

总体来说, CPLD 的结构由四个部分组成: 可编程 I/O 单元、可编程基本逻辑单元(CLB)、可编程布线资源(布线池、布线矩阵)和其他辅助模块(时钟资源), 如图 1-4 所示。

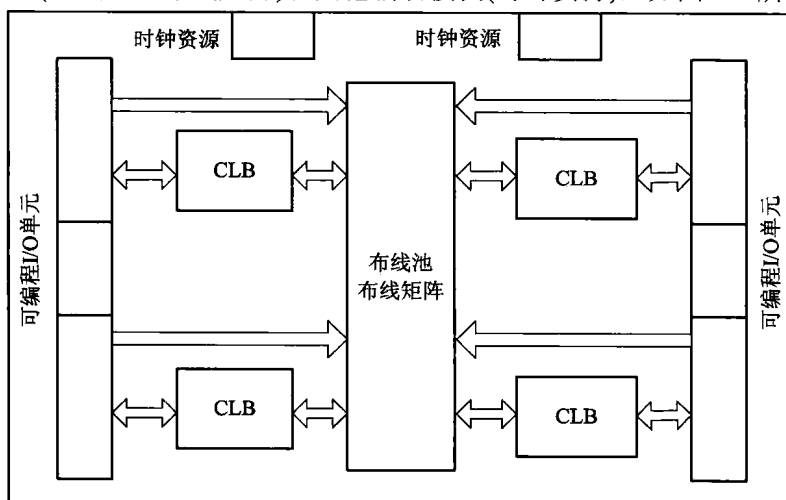


图 1-4 CPLD 的结构示意图

在工艺上, CPLD 都采用 E²COMS 结构, 有些 CPLD 还集成了 RAM、FIFO 等存储器。目前, 各家公司最新推出的 CPLD 不仅在工艺上有很大的突破, 功耗更低, 速度更快, 而且在规模上也有所突破; CPLD 在粘合逻辑处理方面虽已达到炉火纯青的地步, 但是其在逻辑规模上的限制使得设计人员面对大型的逻辑设计却一脸茫然, 望而却步。

1.2.2 FPGA 技术

约在 20 世纪 80 年代早期, ASIC(Application Specific Integrated Circuit, 专用集成电路)技术已经在飞速发展。但是在很多应用场合, 昂贵的 ASIC 费用不是广大客户所想要的, 而且 ASIC 流片的风险太大, 周期太长, 在不确定芯片需求量很大的情况下, 人们是非常谨慎的。而 CPLD 技术虽然有飞跃, 但是依然不能实现复杂的功能, 尤其是无法实现复杂逻辑运算的功能。ASIC 与 CPLD 之间的鸿沟越来越明显。幸运的是, 1984 年世界上首款 FPGA 在 Xilinx 诞生。首款 FPGA 基于 CMOS 工艺, 并且采用 SRAM 单元, 最小单元由一个 3 输入查找表(LUT)与寄存器组成。首款 FPGA 的诞生, 已经给人们发出了一个信息, 除了 ASIC 和 CPLD 之外, 另外一种新型结构的可编程逻辑器件会给逻辑设计带来新的活力。

FPGA 刚开始大部分用来作粘合逻辑、中等复杂程度的状态机和相对有限的数据处理任务。在 20 世纪 90 年代, FPGA 的规模和复杂度开始增加, 市场扩展到通讯和网络领域, 而且都涉及到大量数据的处理。21 世纪初, FPGA 在消费类产品, 如汽车和工业领域的应用也经历了爆炸式的增长, 发展到现在, FPGA 的黄金时代已经到来。ASIC 验证、微处理器核的嵌入、系统级的解决方案, 这些 FPGA 不断创新的理念逐渐适应了目前市场的需求。

对于 ASIC 公司来说, FPGA 经常用于提供一个硬件验证平台来验证新算法新协议的物