



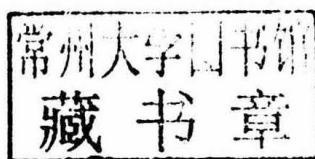


普通高校



# CPLD/FPGA 设计与应用基础教程 ——从 Verilog HDL 到 SystemVerilog

郭利文 邓月明 编著



北京航空航天大学出版社

## 内 容 简 介

本书涵盖了 Verilog HDL 和 SystemVerilog 设计、仿真及验证所需的理论知识点,同时涵盖了时序约束等与 CPLD/FPGA 设计相关的重要知识点。从 Verilog HDL 基础语法出发,逐渐过渡到 SystemVerilog。本书包含了 Verilog HDL 和 SystemVerilog 基础语法及最新进展,所涉及的实例均在实际中应用过,所涉及各类 CPLD/FPGA 平台均为目前全球主流 CPLD/FPGA 开发平台。

本书既可作为高年级本科生或研究生的 CPLD/FPGA 教材,又可作为从事 CPLD/FPGA 项目开发实践的工程技术人员的参考书。

### 图书在版编目(CIP)数据

CPLD / FPGA 设计与应用基础教程:从 Verilog HDL 到 SystemVerilog / 郭利文,邓月明编著. -- 北京:北京航空航天大学出版社,2019.4

ISBN 978-7-5124-2991-8

I. ①C… II. ①郭… ②邓… III. ①可编程序逻辑器件—系统设计—教材 IV. ①TP332.1

中国版本图书馆 CIP 数据核字(2019)第 070598 号

版权所有,侵权必究。

## CPLD/FPGA 设计与应用基础教程 ——从 Verilog HDL 到 SystemVerilog

郭利文 邓月明 编著

责任编辑 张军香

\*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@buaacm.com.cn 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

\*

开本:710×1 000 1/16 印张:26.25 字数:559 千字

2019 年 8 月第 1 版 2019 年 8 月第 1 次印刷 印数:3 000 册

ISBN 978-7-5124-2991-8 定价:79.00 元

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

# 前 言

---

2006年,自Google首席执行官埃里克·施密特在搜索引擎大会上首次提出“云计算”的概念以来,过去十年来围绕云计算、大数据等方面的科技与研究迅速发展,日新月异。特别是2016年AlphaGo第一次战胜了人类顶尖围棋棋手后,基于云计算、大数据而衍生出来的人工智能又达到了一个新的起点。过去三年,全球的科技界以云计算、大数据、人工智能以及互联网科技为核心,重新对过去进行总结,对未来进行布局,尤其是半导体行业,更是发生了翻天覆地的变化——2015年,ADI收购Linear;西部数据收购Sandisk;收购了LSI和PLX公司的安华高收购Broadcom后,华丽变身为新Broadcom公司。2016年,高通收购NXP,软银收购ARM,Intel收购Altera;2017年,Intel收购Mobileye,东芝芯片业务出售,Broadcom收购Brocade,Marvell收购Carvium。2018年收购了Actel公司和PMC公司后的Microsemi被Microchip收购,而高通并购NXP失败……。各种大型跨国、跨行业、跨领域的半导体并购案层出不穷,不仅涉及数字器件领域,也涉及模拟器件领域;不仅涉及科技公司本身,还涉及各类大型金融财团;不仅有大鱼吃小鱼的并购,还有蛇吞象的重组;不仅有对本行业的兼并,还有跨领域的转型……。随着人工智能、大数据和云计算的迅速发展,需要大量的计算能力和资源,因此,具有超强计算天赋的CPLD/FPGA迅猛发展,它们出现在各类数据中心和HPC的基础架构中,并承担着核心的计算任务。

本书基于此时代背景,结合当前主流的CPLD/FPGA设计理念,根据作者多年的实践经验,系统比较了目前最为流行的Verilog HDL和SystemVerilog的语法特点,从基础的语法结构入门到简单程序设计,从有限状态机到接口,从设计到仿真,从断言到功能覆盖,从功能到时序,一一涵盖,系统地对Verilog HDL和SystemVerilog语法应用进行了详细探讨。全书实例丰富,图文并茂,由浅入深,详细地介绍了CPLD/FPGA的设计与应用。

全书分为四大部分,共11章。第一部分是第1章,重点介绍CPLD/FPGA的基本概念,包括发展历程、硬件架构及基本原理,并简单介绍CPLD/FPGA的设计理念、设计语言及验证流程等。第二部分涵盖第2~5章,主要介绍传统的Verilog



HDL 的语法逻辑,其中第 2 章重点介绍 Verilog HDL 语言的语法基础及相关应用,包括模块与端口的定义、注释、时延以及三种抽象层级不同的描述:数据流描述、行为级描述和结构化描述等。第 3 章主要介绍 Verilog HDL 语法的基本要素,包括标识符、数据类型、数值集合、关键词、参数、表达式及编译程序指令等。第 4 章主要介绍 Verilog HDL 语法中的语句块、高级程序设计语句、模块的参数描述、任务及函数等高阶描述。第 5 章重点介绍 Verilog HDL 语言中的任务及函数。第三部分包括第 6、7、9、10 章和第 11 章,重点介绍 SystemVerilog 的基础语法,以及如何进行设计、仿真、断言及功能覆盖等。其中,第 6 章重点介绍 SystemVerilog 之有别于 Verilog HDL 语言的各种语法概念。第 7 章重点讨论如何进行有限状态机的设计,包括有限状态机的基本概念、算法描述、基本语法要素、状态初始化与编码、Full Case 与 Parallel Case 及有限状态机的描述等。第 9 章主要讲述在基于时钟的硬件设计世界里,如何通过硬件线程以及线程与线程之间的接口进行 SystemVerilog 设计,同时重点介绍 SystemVerilog 的新类型 interface 及新结构体 modport。第 10 章主要就 SystemVerilog 特有的仿真特性进行具体详细的介绍,并重点介绍 SystemVerilog 的类、随机化及并行线程的使用。第 11 章主要讲述 SystemVerilog 语言最为重要的两个验证性能:断言与功能覆盖,并分别详细介绍断言和功能覆盖,包括断言的种类、断言的构成、序列与属性的特点等,同时全面讲述功能覆盖的组合、特点以及如何进行覆盖率分析等。第四部分是第 8 章,主要就同步数字电路时序分析与优化方面进行重点讨论,包括同步数字电路的基本概念、D 触发器的工作原理、亚稳态的产生原理,以及同步寄存器、同步数字系统的时序约束、时钟的概念、IO 时序分析、时序例外、PLL 及如何进行时序优化。

与其他教材相比,本书的主要特点体现在如下几方面:

①内容新颖。本书融会贯通了 Verilog HDL 和 SystemVerilog 基础语法及其最新进展,所涉及的实例及各类 CPLD/FPGA 平台均为目前全球的主流 CPLD/FPGA 开发平台。

②技术实用。全书以夯实基础为出发点,以实例讲解为突破口,加强学习和教学,其中的实例都是从工程实践中提炼出来的。

③知识点丰富。全书涵盖了 Verilog HDL 和 SystemVerilog 设计、仿真及验证所需的理论知识点,同时涵盖了时序约束等与 CPLD/FPGA 设计相关的重要知识点。从 Verilog HDL 基础语法出发,逐渐过渡到 SystemVerilog,这也是本书的重点和特色之一。

④适应面广。本书所涉及的大部分实例不依赖于具体平台和厂商支持,因此可以直接移植到各家的 CPLD/FPGA 开发系统中。本书既涵盖了 Verilog HDL 的基础语法,也重点讲述了 SystemVerilog 的设计验证与仿真,对于想要学习硬件可编程逻辑语言的工程师或者学生来说,均可以找到适合各自入门的章节,并迅速提高。因此,本书不仅适合于工程技术人员阅读,也适用于高校师生作为学习 CPLD/FPGA

的教学用书。

⑤适合教学。本书从 Verilog HDL 和 SystemVerilog 基础语法入门,由浅入深,大量的短小实例使读者可及时理解消化所讲的理论知识点,保证知识点教与学的完整性;同时在每章通过一个综合实例来覆盖本章所涉及的各种主要知识点,使读者可系统地掌握知识点。

本书另附配套光盘 1 张,提供了本书的多媒体课件及书中所有实例的设计源代码,供师生教学参考。

全书由郭利文、邓月明编著。高芳莉、何睦等工程师和湖南师范大学 2016 级通信工程专业翦璋、2017 级物联网工程专业黄正宇等同学为本书的编写付出了诸多努力,提供了许多详细的建议和意见,从而促成了本书的迅速问世,在此一并表示感谢。感谢我的导师王玲教授对我的学术悉心指导。同时还要感谢教育部产学研协同育人项目(编号 201701060014、2018002216005)对本书编写工作的资助。

在本书的编写过程中,作者参考了大量的国内外著作和资料,吸取了最近数年 CPLD/FPGA 最新的发展成果;听取了多方面的宝贵意见和建议,同时也根据具体的建议对某些章节进行了调整。在此对这些文献的作者及给予本书作者帮助的同仁致以衷心的感谢。

在本书的编写过程中,家人的宽容和帮助一直是作者前行的动力,感谢家人在作者挑灯夜战时默默的奉献,感谢女儿每晚默默的陪伴。

由于作者水平有限,书中错误和不足之处在所难免,敬请各位读者批评指正。

郭利文

2019 年 3 月 9 日



<b>第 1 章 概 述</b> .....	1
1.1 CPLD/FPGA 发展演变 .....	1
1.2 乘积项结构的基本原理 .....	3
1.3 查找表结构的基本原理 .....	5
1.4 Virtex UltraScale+ 系列 FPGA 简介 .....	7
1.5 CPLD/FPGA 设计与验证流程 .....	8
1.5.1 系统级功能定义与模块划分 .....	8
1.5.2 寄存器传输级与门级描述 .....	9
1.5.3 系统综合编译 .....	10
1.5.4 布局规划与布线 .....	11
1.5.5 仿 真 .....	11
1.5.6 程序设计下载配置 .....	12
1.5.7 测试与验证 .....	15
1.6 CPLD/FPGA 开发平台简介 .....	16
1.7 硬件描述语言的介绍 .....	16
1.8 硬件语言与软件语言的区别 .....	20
本章小结 .....	20
思考与练习 .....	21
<b>第 2 章 Verilog HDL 入门指南</b> .....	22
2.1 模 块 .....	22
2.2 模块端口及声明 .....	24
2.3 注 释 .....	25
2.4 数据流描述 .....	26



2.4.1 连续赋值语句	27
2.4.2 时延	29
2.5 行为级描述	31
2.5.1 initial 语句	31
2.5.2 always 语句	32
2.5.3 时序控制	36
2.6 结构化描述	39
2.6.1 门级建模及描述	39
2.6.2 用户定义原语(UDP)	42
2.6.3 模块例化	46
2.7 混合描述	49
本章小结	51
思考与练习	52

**第3章 Verilog HDL 语法要素** ..... 53

3.1 标识符	53
3.2 数值集合	54
3.2.1 数字	54
3.2.2 字符串	56
3.2.3 参数	56
3.3 数据类型	56
3.3.1 线网类型	57
3.3.2 变量类型	60
3.4 数组	63
3.5 内建门级原语	64
3.6 操作数	66
3.6.1 常数、参数、线网与变量	66
3.6.2 位选择及部分位选	67
3.6.3 存储单元	68
3.6.4 功能调用	68
3.7 操作符	68
3.7.1 算术操作符	70
3.7.2 关系操作符	71
3.7.3 相等操作符	71
3.7.4 逻辑操作符	72
3.7.5 按位操作符	72
3.7.6 缩减操作符	73

3.7.7 移位操作符 .....	74
3.7.8 条件操作符 .....	75
3.7.9 拼接复制操作符 .....	76
3.8 编译指令 .....	76
3.9 实例:带可预置数据的8位自增/减计数器设计 .....	79
本章小结 .....	80
思考与练习 .....	80
<b>第4章 Verilog HDL 语法进阶描述 .....</b>	<b>82</b>
4.1 语句块 .....	82
4.1.1 顺序语句块 .....	82
4.1.2 并行语句块 .....	83
4.2 过程赋值语句 .....	85
4.2.1 阻塞赋值语句 .....	85
4.2.2 非阻塞赋值语句 .....	88
4.2.3 过程赋值语句的使用原则 .....	90
4.3 过程性连续赋值语句 .....	91
4.4 高级程序设计语句 .....	93
4.4.1 条件语句 .....	93
4.4.2 case 语句 .....	97
4.4.3 循环语句 .....	101
4.4.4 generate 语句 .....	105
4.5 参数化设计 .....	111
4.6 实例:基于SFF8485规格的SGPIO协议的Verilog HDL实现 .....	116
4.6.1 SGPIO 协议简介 .....	116
4.6.2 SGPIO 协议接收者的 Verilog HDL 代码设计 .....	118
本章小结 .....	129
思考与练习 .....	129
<b>第5章 任务及函数 .....</b>	<b>130</b>
5.1 任 务 .....	130
5.1.1 任务声明 .....	130
5.1.2 任务调用 .....	131
5.2 函 数 .....	132
5.2.1 函数声明 .....	132
5.2.2 函数调用 .....	133
5.3 系统任务和系统函数 .....	133



5.3.1	显示任务 .....	134
5.3.2	仿真控制任务 .....	135
5.3.3	文件输入输出任务 .....	135
5.3.4	变换函数 .....	138
5.3.5	概率分布函数 .....	138
5.3.6	仿真时间函数 .....	139
5.4	命名事件 .....	139
5.5	层次路径名 .....	140
5.6	共享任务和函数 .....	141
5.7	实例:带可预置数据的 8 位自增/减计数器设计 .....	143
	本章小结 .....	146
	思考与练习 .....	146
<b>第 6 章 SystemVerilog 基础语法 .....</b>		<b>147</b>
6.1	基本数据类型 .....	147
6.1.1	logic 类型 .....	148
6.1.2	2 值数据类型 .....	149
6.1.3	枚举类型 .....	150
6.1.4	typedef .....	155
6.1.5	结构体和共同体 .....	157
6.2	数 组 .....	161
6.2.1	多维数组 .....	161
6.2.2	动态数组 .....	162
6.2.3	关联数组 .....	164
6.2.4	队 列 .....	166
6.2.5	数组的基本操作方法 .....	168
6.2.6	字符串 .....	170
6.3	过程语句 .....	172
6.3.1	always_comb 语句和 assign 语句 .....	172
6.3.2	always_latch 语句 .....	174
6.3.3	always_ff 语句 .....	175
6.3.4	final 语句 .....	176
6.4	unique 和 priority .....	176
6.4.1	unique .....	176
6.4.2	priority .....	178
6.5	循环语句 .....	180
6.5.1	while 循环 .....	180

6.5.2 do...while 循环 .....	181
6.5.3 foreach 循环 .....	182
6.5.4 continue 和 break .....	183
6.6 模块例化 .....	184
6.7 实例:采用 SystemVerilog 实现汉明码的编码设计 .....	187
本章小结 .....	189
思考与练习 .....	189
<b>第 7 章 有限状态机设计</b> .....	<b>191</b>
7.1 有限状态机的基本概念 .....	191
7.1.1 Mearly 型状态机 .....	193
7.1.2 Moore 型状态机 .....	194
7.2 有限状态机的算法描述 .....	194
7.3 有限状态机描述的基本语法 .....	195
7.4 状态初始化 .....	198
7.5 状态编码 .....	202
7.5.1 二进制码(Binary 码) .....	203
7.5.2 格雷码(Gray 码) .....	203
7.5.3 独热码(one-hot 码)和独冷码(one-cold 码) .....	205
7.5.4 状态编码原则和编译指导 .....	205
7.6 Full Case 与 Parallel Case .....	206
7.7 状态机的描述 .....	210
7.7.1 一段式状态机 .....	211
7.7.2 两段式状态机 .....	213
7.7.3 三段式状态机 .....	219
7.7.4 小 结 .....	221
7.8 实例:交通信号灯控制系统的 SystemVerilog 程序设计 .....	222
本章小结 .....	229
思考与练习 .....	229
<b>第 8 章 同步数字电路与时序分析</b> .....	<b>231</b>
8.1 同步数字电路的基本概念 .....	231
8.1.1 同步数字电路 .....	231
8.1.2 时钟域 .....	233
8.2 D 触发器的工作原理 .....	234
8.3 亚稳态的产生原理及同步寄存器 .....	236
8.4 同步数字系统的时序约束 .....	244



8.5 时钟 .....	246
8.5.1 时钟偏斜与抖动 .....	246
8.5.2 $F_{\max}$ .....	249
8.6 IO 时序分析 .....	249
8.6.1 输入时序分析 .....	250
8.6.2 输出时序分析 .....	251
8.7 时序例外 .....	252
8.7.1 False Path .....	253
8.7.2 MultiCycle Path .....	253
8.8 PLL .....	256
8.9 时序优化 .....	257
8.10 实例:采用 SystemVerilog 实现对开关信号的消抖设计 .....	259
本章小结 .....	265
思考与练习 .....	265
<b>第 9 章 硬件线程与接口 .....</b>	<b>267</b>
9.1 硬件线程的基本概念 .....	267
9.1.1 数据路径 .....	268
9.1.2 硬件线程的算法描述 .....	271
9.2 硬件线程的连接 .....	272
9.3 硬件线程的同步 .....	273
9.4 实例:基于串并转换的硬件线程连接实现 .....	276
9.4.1 Master_Interface 硬件线程介绍 .....	278
9.4.2 Slave_Interface 硬件线程介绍 .....	280
9.4.3 代码实现 .....	281
9.5 异步硬件线程的连接 .....	289
9.6 接口 .....	292
9.6.1 接口声明和例化 .....	293
9.6.2 modport .....	294
9.7 实例:采用接口实现 SGPIO 的数据传送 .....	294
9.7.1 SGPIO 简介 .....	294
9.7.2 SGPIO 程序设计 .....	296
本章小结 .....	301
思考与练习 .....	302
<b>第 10 章 SystemVerilog 仿真基础 .....</b>	<b>303</b>
10.1 仿真简介 .....	303

10.1.1	仿真入门	303
10.1.2	仿真器原理	306
10.1.3	测试平台	308
10.2	program	310
10.3	面向对象编程与类	312
10.3.1	面向对象编程简介	312
10.3.2	类简介	313
10.3.3	静态变量与静态方法	316
10.3.4	this	317
10.3.5	类的内嵌	318
10.3.6	对象的基本操作	318
10.3.7	类的继承与多态	322
10.4	随机化	326
10.4.1	随机化基础	326
10.4.2	randcase	330
10.4.3	randsequence	330
10.4.4	随机约束基础	332
10.4.5	权重分布	334
10.4.6	约束操作符	335
10.5	并行线程	341
10.5.1	wait	345
10.5.2	Disable	346
10.5.3	mailbox	347
10.5.4	命名事件	350
10.5.5	semaphore	353
10.6	实例:简单的多口路由仿真程序设计	355
	本章小结	361
	思考与练习	361
<b>第 11 章</b>	<b>断言与功能覆盖</b>	<b>362</b>
11.1	断言	362
11.1.1	立即断言	363
11.1.2	时序操作符	365
11.1.3	序列	367
11.1.4	属性	368
11.1.5	并行断言	369
11.1.6	重复操作符	371



11.1.7	逻辑操作符	374
11.1.8	条件操作符	376
11.1.9	断言系统函数	377
11.2	覆盖率介绍	379
11.2.1	代码覆盖率	380
11.2.2	断言覆盖率	381
11.2.3	功能覆盖率	381
11.3	功能覆盖	382
11.3.1	覆盖点与覆盖组	382
11.3.2	交叉覆盖	384
11.3.3	仓	385
11.3.4	翻转覆盖	388
11.3.5	覆盖选项	390
11.3.6	采样函数	393
11.3.7	覆盖率数据分析	394
11.4	实例:对有限状态机进行功能覆盖设计	395
	本章小结	399
	思考与练习	399
	参考文献	401

# 第 1 章

## 概 述

本章重点介绍 CPLD/FPGA 的一些基本概念,包括其发展历程、硬件架构及基本原理,并简单介绍 CPLD/FPGA 的设计理念、设计语言及验证流程等。

本章的主要内容如下:

- CPLD/FPGA 发展演变及介绍;
- 乘积项结构的基本原理;
- 查找表结构的基本原理;
- Virtex UltraScale+ 系列 FPGA 简介;
- CPLD/FPGA 设计与验证流程;
- CPLD/FPGA 开发平台简介;
- 硬件描述语言的介绍;
- 硬件语言与软件语言的区别。

### 1.1 CPLD/FPGA 发展演变

20 世纪 70 年代,世界第一颗可编程逻辑器件(PLD, Programmable Logic Device)诞生。其输出结构是可编程的逻辑宏单元(MC, Macro Cell),与传统硬件数字电路设计不同,它采用软件设计来完成芯片的硬件结构设计,因而比传统的数字电路具有更强的灵活性。但是,由于其结构过于简单,只能用于实现规模较小的电路。随着芯片制造工艺和技术的发展,到了 20 世纪 80 年代中期,综合了 PAL(可编程阵列逻辑, Programmable Array Logic)和 GAL(通用阵列逻辑, General Array Logic)器件的优点而推出了复杂可编程逻辑器件(CPLD, Complex Programmable Logic Device)。相较于 PAL 和 GAL, CPLD 主要由可编程逻辑宏单元围绕中心的可编程互连矩阵单元组成。其中可编程逻辑宏单元结构复杂,并且具有复杂的 I/O 单元互连结构,规模大,属于大规模集成电路范畴。CPLD 目前被广泛应用于网络、仪器仪表、汽车电子、数控机床及航天测控设备等领域,成为电子产品中不可或缺的组成部分。

1985 年, Xilinx 公司推出了全球第一款现场可编程门阵列(FPGA, Field Pro-



grammable Gate Array)产品 XC2064,采用  $2\ \mu\text{m}$  制造工艺,包含 64 个逻辑模块和 85 000 个晶体管,不超过 1 000 个逻辑门。它是作为专用集成电路(ASIC, Application Specific Integrated Circuit)领域中的一种半定制电路而出现的,既弥补了定制电路的不足,又克服了原有可编程器件门电路数量有限的缺点。相较于 PAL、GAL 和 CPLD,FPGA 采用了逻辑单元数组(LCA, Logic Cell Array)这样一个概念,利用小型查找表结构实现组合逻辑,同时,每个查找表连接到一个 D 触发器,由此构成一个既可以实现组合逻辑功能又能实现时序逻辑功能的基本逻辑单元模块。这些模块之间利用金属联机互连或者直接连接到 IO 模块上。因此一个 LCA 内部包括可配置逻辑模块(CLB, Configurable Logic Block)、输入输出模块(IOB, Input Output Block)及内部联机(Interconnect)三部分。由于 ASIC SoC 设计周期平均值是 14 个月到 24 个月,用 FPGA 进行开发,开发周期平均可降低 55%,因此 FPGA 被广泛应用于芯片的原型设计。

随着制造工艺的进步,以及 CPLD/FPGA 可编程设计及可定制的优点,CPLD/FPGA 等设计公司在 20 世纪 90 年代如雨后春笋般出现,最后又通过市场兼并整合,形成了 Xilinx、Altera、Lattice 三大阵营公司。随着云计算和大数据时代的到来,特别是深度学习及人工智能的出现,相较于 CPU/GPU 等 ASIC,FPGA 的硬件架构具有决定性的优势,各云计算和大数据公司纷纷开始部署基于 FPGA 的异构系统。特别是 2015 年 Intel 以 167 亿美元收购了全球第二大 CPLD/FPGA 公司 Altera,并迅速整合进入其最新一代服务器 Purley 平台,意图打造 CPU+FPGA 的强势组合。FPGA 又迎来了新的发展机遇。

Xilinx 公司作为 FPGA 的发明者,也是世界上最大的可编程逻辑器件领导厂商,一直引领 CPLD/FPGA 领域的技术变革和市场方向。推出了全面的多节点产品组合,以满足广泛的应用需求。不仅拥有 XC9500XL、CoolrunnerII 等低功耗高性能的 CPLD 产品,而且同时拥有各种不同制程的 Spartan、Virtex、Artix、Zynq、Kintex 等系列的 FPGA 产品,不同系列的产品面向的市场也各不相同,如 Spartan 系列主要面向以 IO 性能优先的应用,Zynq 系列面向以系统性能优先的应用等。最近五年,Xilinx 公司着力于云计算、大数据、人工智能、深度学习及自动驾驶等市场,集中力量发展高端 FPGA 产品,针对各系列 FPGA,推出了最新的 28 nm 的 7 系列产品,同时针对 Virtex、Kintex 和 Zynq 系列,特别推出了 16 nm 工艺的 UltraSCALE+ 系列产品。

2015 年 Intel 收购 Altera 公司后,一跃而成为引领全球的可编程逻辑器件厂商之一。在收购之前,Altera 公司已经在 CPLD 和 FPGA 领域深度布局,拥有完善的产品线以应对各种不同的市场应用,包括 Max 系列的可编程逻辑器件及 Stratix、Arria 及 Cyclone 系列的 FPGA 产品。其中值得一提的是,MAX 10 作为一

款跨界产品,模糊了 CPLD/FPGA 之间的界限,填补了高端 CPLD 和低端 FPGA 之间的空白。被 Intel 收购之后,Intel 开始着力于新一代云计算、大数据、深度学习及人工智能方面的布局,通过尝试打通 CPU 和 FPGA 之间的连接,实现 CPU 和 FPGA 之间的 UPI 和 PCIE 互联,采用离散(Stratix 10 系列 FPGA)或者 MCP(Arria 10 系列 FPGA)封装等方式,且拥有 HSSI 等高吞吐量接口,针对 CPU 和 FPGA 不同的优势,实现任务分类,关键任务并行进行,有效提升了服务器的处理性能。

Lattice 公司是 ISP(In System Program,在系统可编程)技术的发明者,而这项技术极大地促进了 PLD 产品的发展。它通过相继收购 Vantis(原 AMD 子公司)、Agere(原 Lucent 微电子部)及 SiliconBlue 公司,成为了全球第三大可编程逻辑器件领导厂商。与 Xilinx 和 Intel 发展高端 FPGA 策略不同,Lattice 着力于发展 CPLD 产品及低端 FPGA 产品。主要产品包括 iCE 系列、MachXO 系列、ECP 系列 CPLD/FPGA 产品及可编程仿真器件等。其中 MachXO 系列和 MAX 10 产品定位相似,弥补了市场上高端 CPLD 和低端 FPGA 之间的空白。

最近几年,以 Intel、Broadcom 为代表的跨国、跨行业、跨领域的大型并购案层出不穷,IC 世界格局不断改变。2016 年,全球功率电子产品供货商 Microsemi 公司正式收购了 Actel 公司,从而成为了全球前四大 FPGA 玩家之一。Actel 为现今航天与军事市场提供应用最广泛的混合信号耐辐射 FPGA 产品。收购后,Microsemi 也积极进军民用和商用市场,并推出了 PolarFire、IGLOO2、RTG4 及 SmartFusion2 等面向不同领域及应用的中端 FPGA 和 SoC FPGA 等。2018 年,Microsemi 公司被 Microchip 公司收购。

作为关键核心技术,中国也在努力布局 FPGA。目前主要有京微雅格、紫光同创等 FPGA 公司在进行国产 FPGA 的研发,不过在短时间内,还无法与上述几家公司相匹敌。

## 1.2 乘积项结构的基本原理

图 1-1 所示为 Lattice 公司出品的 ispMACH 4000 系列 CPLD 的通用逻辑块结构,36 个来自通用布线池的输入进入“与”逻辑矩阵(也就是乘积项结构)后,通过逻辑分配器及宏单元运算后进入输出布线池。传统 CPLD 基本上都遵循乘积项的基本结构,例如 Altera 公司的 MAX 系列及 Xilinx 公司的 XC9500 系列等。

所谓乘积项结构实际上就是一个“与或”结构——其中,“与”逻辑可进行编程设计,“或”逻辑固定,可形成一个组合逻辑。图 1-2 所示为 ispMACH 4000 系列的乘积项结构示意图,由图可知,83 个“与”门输入端均可以根据用户的需要进行软件程序设计,从而实现具体的组合逻辑。