

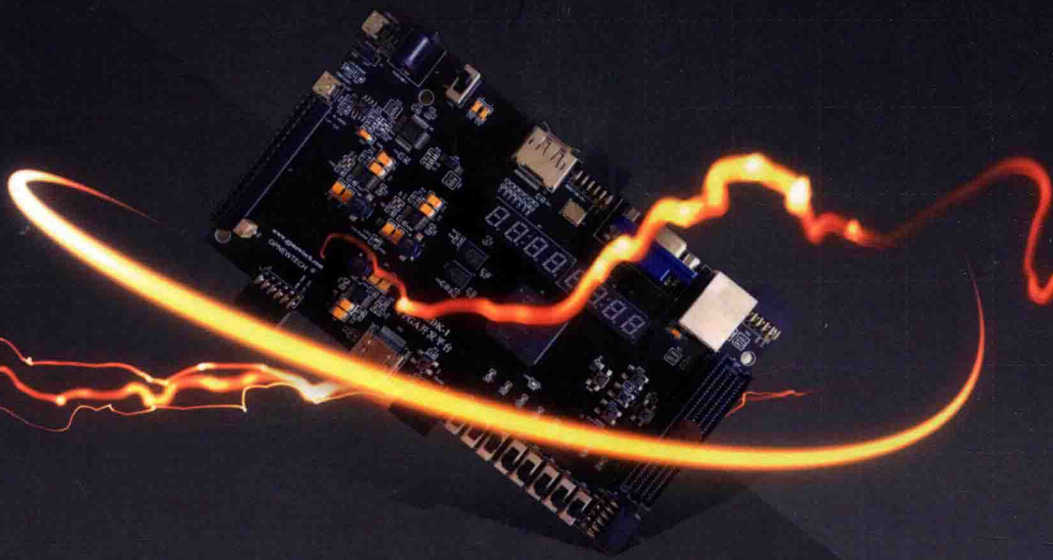
学习资源
见书中
学习说明

电子系统EDA新技术丛书

Xilinx FPGA 权威设计指南

基于Vivado 2018集成开发环境

◎ 何 宾 编著



- ★ 系统介绍Vivado 2018集成开发环境的新功能和特性
- ★ 涵盖高级综合工具、部分可重配置工具和调试工具的高级应用
- ★ 深度剖析UltraScale FPGA内部原语的原理和使用方法



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

电子系统 EDA 新技术丛书

介绍

Xilinx FPGA 权威设计指南

基于 Vivado 2018 集成开发环境

何 宾 编著



电子工业出版社

Publishing House of Electronics Industry

内 容 简 介

本书系统地介绍了 Xilinx 新一代集成开发环境 Vivado 2018 的设计方法、设计流程和具体实现。全书共 11 章, 内容包括 Xilinx 新一代 UltraScale 结构、Vivado 集成设计环境导论、Vivado 工程模式基本设计实现、Vivado 非工程模式基本设计实现、创建和封装用户 IP 核流程、Vivado 高级约束原理及实现、Vivado 调试工具原理及实现、Vivado 部分可重配置原理及实现、Vivado HLS 原理详解、Vivado HLS 实现过程详解、HDMI 显示屏驱动原理和实现。

本书参考了 Xilinx 新一代的 Vivado 2018 设计套件设计资料, 理论与应用并重, 将 Xilinx 新一代的设计理论贯穿在具体的设计实现中。

本书可作为使用 Xilinx 集成开发环境 Vivado 进行 FPGA 设计的工程技术人员的参考书, 也可作为电子信息类专业高年级本科生和研究生的教学与科研用书, 还可作为 Xilinx 公司 Vivado 相关培训的培训教材及工程技术人员的参考书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目 (CIP) 数据

Xilinx FPGA 权威设计指南: 基于 Vivado 2018 集成开发环境/何宾编著. —北京: 电子工业出版社, 2018.10
(电子系统 EDA 新技术丛书)

ISBN 978-7-121-34937-9

I. ①X… II. ①何… III. ①现场可编程门阵列—系统设计—指南 IV. ①TP331.2-62

中国版本图书馆 CIP 数据核字 (2018) 第 196842 号

策划编辑: 张 迪 (zhangdi@phei.com.cn)

责任编辑: 张 迪

印 刷: 三河市良远印务有限公司

装 订: 三河市良远印务有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 34.25 字数: 876 千字

版 次: 2018 年 10 月第 1 版

印 次: 2018 年 10 月第 1 次印刷

定 价: 129.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: (010) 88254469, zhangdi@phei.com.cn。

前 言

全球知名的可编程逻辑元器件生产厂商——美国 Xilinx 公司，于 2012 年发布了新一代 Vivado 集成开发环境，使得新一代 FPGA 的设计环境和设计方法发生了重大变化。2014 年年初，Xilinx 新一代 UltraScale 结构的 FPGA 也进入量产阶段。这些都标志着未来在高性能数据处理方面（如人工智能、云计算和物联网等），FPGA 将发挥越来越重要的作用。

Xilinx 新一代集成开发环境 Vivado 突出基于知识产权（Intellectual Property, IP）核的设计方法，更加体现系统级设计的思想，进一步增强了设计者对 FPGA 底层物理约束和时序约束能力，这样设计者可以尝试选择不同的设计策略，然后评估在不同策略下的最终实现结果，从中找到最佳的设计方案。

Vivado 设计套件所集成的高级综合工具 HLS，真正实现了从软件算法到 FPGA 物理硬件实现的自动转换，加速了算法模型的构建与实现过程，显著提高了设计效率，缩短了产品的上市时间，并将 FPGA 的设计方法从传统的基于 HDL 的 RTL 级设计提高到了基于 C/C++/SystemC 的高层次设计，为 FPGA 的设计方法带来了一场深刻的变革。

本书是在《Xilinx FPGA 设计权威指南——Vivado 2014 集成开发环境》（电子工业出版社，2015）的基础上，针对读者提出的意见和建议，对原书进行了大幅度修订。主要修订内容包括：

（1）采用 Xilinx 公司最新的 Vivado 2018 设计套件，书中所有的设计实例均在 Vivado 2018 设计套件中进行了验证。

（2）在介绍 UltraScale 结构时，增加了对内部单元原理和用法的进一步解释与说明，用于帮助读者能够正确地理解并掌握 Xilinx FPGA 底层原语的原理及使用方法，以实现 RTL 级上的最佳设计。

（3）在介绍第 2 章和第 3 章内容时，重新提供了设计实例，并增加了设计实例的难度，以帮助读者能够进一步学习和掌握 Vivado 设计套件在工程模式和非工程模式下的完整设计流程，并增加了对一些设计流程细节的说明。此外，对从用户 HDL 代码中生成 HDL 例化模板的方法进行了详细介绍。

（4）在第 5 章介绍 IP 封装和调用的内容时，增加了不包含源文件的 IP 封装和调用的方法。

（5）在第 7 章介绍 Vivado 调试工具时，增加了虚拟输入输出（Virtual Input Output, VIO）IP 核在调试设计中的应用。

（6）在第 8 章介绍部分可重配置原理及实现的内容时，增加了工程模式下的可重配置实现过程，以及部分重配置控制器 PRC 的原理及应用。

（7）考虑到 Vivado HLS 工具正逐渐大规模广泛应用于 Xilinx FPGA 设计中，因此本书大幅度增加了对 HLS 原理内容的介绍，使所涵盖的内容更全面。本次修订后，将 HLS 相关

内容扩展为两章，分别介绍 HLS 原理和实现过程。并且，对 HLS 实现过程的一些细节问题进行了进一步的详细说明，对设计实例中的一些代码和用户策略进行了修正，以满足读者对掌握 HLS 设计方法的要求。

参加本书编写的人员还有王中正、张艳辉和汤宗美。其中，王中正参与第 3 章和第 6 章设计实例的验证，张艳辉参与第 7 章设计实例的验证，汤宗美参与第 8 章设计实例的验证。全书由何宾统稿和定稿。

本书的修订得到 Xilinx 大中华区新任大学计划经理陆佳华先生和 Xilinx 亚太区传媒总监张俊伟女士的大力支持和帮助，他们为本书的编写提供了大量的资料和硬件设计平台。回想起来，2008 年作者与 Xilinx 大学计划建立正式的合作关系，到今年整整十年了。在这十年间，相继得到 Xilinx 公司不同技术专家的无私帮助和鼎力支持，才能使作者编写并出版一系列 Xilinx FPGA 方面的著作和教材，以促进该技术在中国的普及和推广，进一步提升国内相关技术的科技实力。也要感谢电子工业出版社的编辑和相关工作人员，他们的辛勤工作保证了本书的高质量出版。

由于 FPGA 技术发展迅速，作者水平有限，书中难免会有疏漏之处，欢迎读者批评指正。

作者

2018 年 9 月于北京



学习说明

Study Shows

1. 本书提供的教学视频、教学课件、设计文件、硬件原理图、使用说明下载地址
北京汇众新特科技有限公司技术支持网址：

<http://www.edawiki.com>

注意：所有教学课件及工程文件仅限购买本书读者学习使用，不得以任何方式传播！

2. 本书作者联络方式

电子邮件：hb@gpnewtech.com

3. 购买硬件事宜由北京汇众新特科技有限公司负责

公司官网：<http://www.gpnewtech.com>

市场及服务支持热线：010-83139176，010-83139076

4. 何宾老师的微信公众号



5. 赛灵思提供的设计资源

网址：<https://github.com/Xilinx>

目 录

第1章	Xilinx 新一代 UltraScale 结构	1
1.1	UltraScale 结构特点	1
1.2	可配置逻辑块	2
1.2.1	可配置逻辑块的特点	2
1.2.2	多路复用器	6
1.2.3	进位逻辑	9
1.2.4	存储元素	13
1.2.5	分布式 RAM	16
1.2.6	只读存储器 (ROM)	18
1.2.7	移位寄存器	21
1.3	时钟资源和时钟管理模块	23
1.3.1	时钟资源	26
1.3.2	时钟管理模块	28
1.4	块存储器资源	29
1.5	专用的 DSP 模块	33
1.6	SelectIO 资源	36
1.7	高速串行收发器	41
1.8	PCI-E 模块	43
1.9	Interlaken 集成块	43
1.10	Ethernet 模块	43
1.11	系统监控器模块	44
1.12	配置模块	44
1.13	互联资源	45
第2章	Vivado 集成设计环境导论	46
2.1	Vivado 系统级设计流程	46
2.2	Vivado 功能和特性	48
2.3	Vivado 中电路结构的网表描述	49
2.4	Vivado 中工程数据的目录结构	50

2.5	Vivado 中 Journal 文件和 Log 文件功能	50
2.5.1	Journal 文件 (Vivado.jou)	50
2.5.2	Log 文件 (Vivado.log)	51
2.6	Vivado 两种设计流程模式	52
2.6.1	工程模式和非工程模式不同点比较	53
2.6.2	工程模式和非工程模式命令的不同	53
2.7	Vivado 中的 XDC 文件	55
2.7.1	XDC 的特性	55
2.7.2	XDC 与 UCF 比较	55
2.7.3	约束文件的使用方法	56
2.7.4	约束顺序	56
2.7.5	XDC 约束命令	58
2.8	Vivado 集成设计环境的启动方法	59
2.9	Vivado 集成设计环境主界面	60
2.10	Vivado 设计主界面及功能	63
2.10.1	流程处理主界面及功能	63
2.10.2	工程管理器主界面及功能	64
2.10.3	工作区窗口	66
2.10.4	设计运行窗口	67
2.11	Vivado 支持的属性	68
3	第 3 章 Vivado 工程模式基本设计实现	76
3.1	创建新的设计工程	76
3.2	创建并添加一个新的设计文件	80
3.3	RTL 详细描述和分析	85
3.3.1	详细描述的实现	85
3.3.2	生成 HDL 例化模板	87
3.4	设计综合和分析	89
3.4.1	综合过程的关键问题	89
3.4.2	设计综合选项	89
3.4.3	执行设计综合	92
3.4.4	综合报告的查看	96
3.5	设计行为级仿真	97
3.6	创建实现约束	102
3.6.1	实现约束的原理	102
3.6.2	I/O 规划器功能	103
3.6.3	添加引脚约束	104

3.6.4	添加简单的时钟约束	109
3.7	设计实现和分析	111
3.7.1	设计实现原理	112
3.7.2	设计实现选项	112
3.7.3	设计实现及分析	117
3.7.4	静态时序分析	123
3.8	设计时序仿真	126
3.9	生成编程文件	127
3.9.1	配置器件属性	127
3.9.2	生成可编程文件	128
3.9.3	生成可编程文件选项	128
3.10	下载比特流文件到 FPGA	130
3.11	生成并烧写 PROM 文件	132
4	第 4 章 Vivado 非工程模式基本设计实现	136
4.1	非工程模式基本命令和功能	136
4.1.1	非工程模式基本命令列表	136
4.1.2	典型 Tcl 脚本的使用	137
4.2	Vivado 集成开发环境分析设计	138
4.2.1	启动 Vivado 集成开发环境	138
4.2.2	打开设计检查点的方法	139
4.3	修改设计路径	139
4.4	设置设计输出路径	140
4.5	读取设计文件	140
4.6	运行设计综合	141
4.7	运行设计布局	142
4.8	运行设计布线	144
4.9	生成比特流文件	145
4.10	下载比特流文件	145
5	第 5 章 创建和封装用户 IP 核流程	148
5.1	Vivado 定制 IP 流程	148
5.2	创建并封装包含源文件的 IP	149
5.2.1	创建新的用于创建 IP 的工程	149
5.2.2	设置定制 IP 的库名和目录	150
5.2.3	封装定制 IP 的实现	151

5.3	调用并验证包含源文件的 IP 设计	155
5.3.1	创建新的用于调用 IP 的工程	156
5.3.2	设置包含调用 IP 的路径	156
5.3.3	创建基于 IP 的系统	158
5.3.4	系统行为级仿真	162
5.3.5	系统设计综合	165
5.3.6	系统实现和验证	166
5.4	创建并封装不包含源文件的 IP	166
5.4.1	创建网表文件	166
5.4.2	创建新的设计工程	167
5.4.3	设置定制 IP 的库名和目录	168
5.4.4	封装定制 IP 的实现	168
5.5	调用并验证不包含源文件的 IP 设计	169
5.5.1	创建新的用于调用 IP 的工程	169
5.5.2	设置包含调用 IP 的路径	170
5.5.3	创建基于 IP 的系统	170
5.5.4	系统设计综合	171
第 6 章	Vivado 高级约束原理及实现	173
6.1	时序检查概念	173
6.1.1	基本术语	173
6.1.2	时序路径	173
6.1.3	建立和保持松弛	175
6.1.4	建立和保持检查	176
6.1.5	恢复和去除检查	179
6.2	时序约束概念	180
6.2.1	时钟定义	180
6.2.2	时钟组	186
6.2.3	I/O 延迟约束	189
6.2.4	时序例外	192
6.3	生成时序报告	205
6.4	添加时序约束	212
6.4.1	时序约束策略 1	212
6.4.2	时序约束策略 2	214
6.5	物理约束原理	219
6.5.1	网表约束	219
6.5.2	布局约束	220



6.5.3	布线约束	221
6.6	布局约束实现	223
6.6.1	修改综合属性	224
6.6.2	布局约束方法	224
6.7	布线约束实现	227
6.7.1	手工布线	227
6.7.2	进入分配布线模式	228
6.7.3	分配布线节点	230
6.7.4	取消分配布线节点	230
6.7.5	完成并退出分配布线模式	230
6.7.6	锁定 LUT 负载上的单元输入	231
6.7.7	分支布线	231
6.7.8	直接约束布线	233
6.8	修改逻辑实现	233
6.9	配置约束原理	235
6.10	增量编译	235
6.10.1	增量编译流程	235
6.10.2	运行增量布局和布线	236
6.10.3	使用增量编译	238
6.10.4	增量编译高级分析	240
7	第 7 章 Vivado 调试工具原理及实现	241
7.1	设计调试原理和方法	241
7.2	创建新的设计	242
7.2.1	创建新的 FIFO 调试工程	242
7.2.2	添加 FIFO IP 到设计中	243
7.2.3	添加顶层设计文件	246
7.2.4	使用 HDL 例化添加 FIFO 到设计中	247
7.2.5	添加约束文件	251
7.3	网表插入调试探测流程方法及实现	253
7.3.1	网表插入调试探测流程的方法	253
7.3.2	网表插入调试探测流程的实现	255
7.4	使用添加 HDL 属性调试探测流程	261
7.5	使用 HDL 例化调试核调试探测流程	262
7.6	VIO 原理及应用	267
7.6.1	设计原理	267
7.6.2	添加 VIO 核	268

7.6.3	生成比特流文件	271
7.6.4	下载并调试设计	272
第8章	Vivado 部分可重配置原理及实现	274
8.1	可重配置导论	274
8.1.1	可重配置的概念	274
8.1.2	可重配置的应用	275
8.1.3	可重配置的特点	278
8.1.4	可重配置术语解释	280
8.1.5	可重配置的要求	282
8.1.6	可重配置的标准	283
8.1.7	可重配置的流程	285
8.2	基于工程的部分可重配置实现	285
8.2.1	设计原理	285
8.2.2	建立可重配置工程	289
8.2.3	创建新的分区定义	291
8.2.4	添加新的可重配置模块	292
8.2.5	设置不同的配置选项	294
8.2.6	定义分区的布局	298
8.2.7	执行 DRC	301
8.2.8	实现第一个运行配置并生成比特流文件	302
8.2.9	实现第二个运行配置并生成比特流文件	304
8.2.10	实现第三个运行配置并生成比特流文件	304
8.2.11	实现第四个运行配置并生成比特流文件	305
8.2.12	下载不同运行配置的部分比特流	305
8.3	基于非工程的部分可重配置实现	307
8.3.1	查看脚本	307
8.3.2	综合设计	309
8.3.3	实现第一个配置	310
8.3.4	实现第二个配置	315
8.3.5	验证配置	317
8.3.6	生成比特流	317
8.3.7	部分重配置 FPGA	318
8.4	部分重配置控制器 PRC 的原理及应用	320
8.4.1	部分重配置控制器原理	320
8.4.2	实现原理	324
8.4.3	创建和配置新的设计	325

8.4.4	添加 ILA 核	327
8.4.5	添加和设置 PRC 核	328
8.4.6	设置不同的配置选项	331
8.4.7	定义分区的布局	334
8.4.8	实现第一个运行配置并生成比特流文件	335
8.4.9	实现第二个运行配置并生成比特流文件	337
8.4.10	实现第三个运行配置并生成比特流文件	337
8.4.11	创建板支持包工程	338
8.4.12	创建应用程序工程	339
8.4.13	创建 zynq_fsbl 应用	343
8.4.14	创建启动镜像	343
8.4.15	从 SD 卡启动引导系统	345
第 9 章	Vivado HLS 原理详解	347
9.1	高级综合工具概述	347
9.1.1	高级综合工具的功能和特点	347
9.1.2	Vivado HLS 工具的优势	348
9.1.3	HLS 中使用术语说明	349
9.1.4	从 C 中提取硬件结构	350
9.1.5	不同的命令对 HLS 综合结果的影响	352
9.2	高级综合工具调度和绑定	354
9.2.1	高级综合工具调度	355
9.2.2	高级综合工具绑定	355
9.3	C 代码的关键属性	356
9.3.1	函数	357
9.3.2	类型	357
9.3.3	循环	364
9.3.4	数组	366
9.3.5	I/O 端口	366
9.3.6	运算符	367
9.4	C 代码级和 RTL 级算法验证	368
9.5	添加命令	370
9.6	延迟和吞吐量的概念	373
9.6.1	设计延迟	373
9.6.2	设计吞吐量	374
9.7	改善延迟	374
9.7.1	延迟最小化	374

9.7.2	用户延迟的定义	375
9.7.3	循环的处理	377
9.8	改善吞吐量	380
9.8.1	数据流优化	381
9.8.2	流水线优化	384
9.9	性能瓶颈——数组	390
9.9.1	数组分割	391
9.9.2	数组重组	394
9.9.3	数据打包	395
9.10	改善面积和资源	396
9.10.1	绑定配置	396
9.10.2	分配命令	397
9.10.3	指定资源	397
9.10.4	函数内联	398
9.10.5	循环合并和平坦化	401
9.10.6	映射数组	401
9.10.7	任意精度整数	403
9.11	I/O 类型	404
9.11.1	组合和时序逻辑设计	404
9.11.2	Vivado HLS I/O 选项	405
9.11.3	模块级协议说明	408
9.11.4	端口级 I/O 协议	412
9.12	命令和编译指示	422
第10章	Vivado HLS 实现过程详解	425
10.1	FIR 滤波器不同语言的描述方式	425
10.1.1	C 语言描述风格	425
10.1.2	C++语言描述风格	427
10.1.3	SystemC 语言描述风格	428
10.2	Vivado HLS 不支持的操作	429
10.3	测试平台编码	431
10.4	指针的使用	432
10.5	流的使用	437
10.6	支持库	438
10.6.1	FFT	439
10.6.2	FIR	442
10.6.3	移位寄存器	446
10.6.4	线性算法库	447

75	10.6.5 OpenCV 库.....	448
77	10.7 Vivado HLS 数字系统实现.....	453
80	10.7.1 基于 HLS 实现组合逻辑.....	453
81	10.7.2 基于 HLS 实现时序逻辑.....	468
84	10.7.3 基于 HLS 实现矩阵相乘.....	476
90	第 11 章 HDMI 显示屏驱动原理和实现	495
94	11.1 HDMI 的发展历史.....	495
95	11.2 HDMI 视频显示接口定义.....	496
96	11.3 HDMI 链路结构.....	497
96	11.4 HDMI 链路时序要求.....	499
97	11.5 HDMI 编码算法.....	500
97	11.6 HDMI 并行编码数据转换原理及实现.....	504
98	11.6.1 数据并行-串行转换.....	505
01	11.6.2 三态并行-串行转换器.....	505
01	11.6.3 OSERDESE2 原语.....	505
03	11.6.4 TMDS 信号转换模块.....	508
04	11.6.5 HDMI 并行编码数据转换的实现.....	508
04	11.7 系统整体设计结构.....	510
05	附录 a7-edp-1 开发板原理图.....	512

第1章 Xilinx 新一代 UltraScale 结构

UltraScale 结构是业界首款采用最先进的 ASIC 架构优化的 All Programmable 结构。本章主要对 UltraScale 结构的 Kintex 和 Virtex 器件特性进行说明, 并对其内部所提供的设计资源进行详细的说明和必要的分析。通过这些分析, 帮助读者在 Vivado 集成开发环境中更加高效地开发基于 UltraScale 结构的 FPGA 应用。

1.1 UltraScale 结构特点

UltraScale 结构能从 20nm 平面的 FET 结构扩展至 16nm 鳍式的 FET 晶体管, 甚至更高的技术, 同时还能够从单芯片扩展到 3D IC。

通过 Xilinx Vivado 设计套件的分析型协同优化方法, UltraScale 结构可以提供海量数据的布线功能, 同时还能智能地解决先进工艺节点上的头号系统性能瓶颈。这种协同设计可以在不降低性能的前提下实现超过 90% 的利用率。

UltraScale 架构不仅能够解决系统总吞吐量扩展和时延方面的局限性, 而且还能够直接应对先进工艺节点上的头号系统性能瓶颈, 即互连问题。UltraScale 新一代互连架构的推出体现了可编程逻辑布线技术的真正突破。

Xilinx 致力于满足从多吉字节智能包处理到多太字节数据路径等新一代应用需求, 即必须支持海量数据流。在实现宽总线逻辑模块(将总线宽度扩展至 512 位、1024 位甚至更高)的过程中, 布线或互连堵塞问题一直是影响实现时序收敛和高质量结果的主要制约因素。过于拥堵的逻辑设计通常无法在早期器件架构中进行布线。即使工具能够对堵塞的设计进行布线, 最终设计也经常需要在低于预期的时钟速率下运行。而 UltraScale 布线架构则能够完全消除布线堵塞问题。结论很简单, 即只要设计合理, 就能够进行布线。

如表 1.1、表 1.2 所示, 给出了 UltraScale 结构的 Kintex 器件特性和 UltraScale 结构的 Virtex 器件特性。

表 1.1 UltraScale 结构的 Kintex 器件特性

	XCKU035	XCKU040	XCKU060	XCKU075	XCKU100	XCKU115
逻辑单元	355,474	424,200	580,440	756,000	985,440	1,160,880
块 RAM (MB)	19.0	21.1	38.0	41.8	59.1	75.9
DSP 切片	1,700	1,920	2,760	2,592	4,200	5,520
PCI-E 块	2	3	3	4	6	6
GTH 16 GB/s 收发器	16	20	32	52	64	64
I/O 引脚	520	520	624	728	832	832
I/O 电压	1.0~3.3V	1.0~3.3V	1.0~3.3V	1.0~3.3V	1.0~3.3V	1.0~3.3V

表 1.2 UltraScale 结构的 Virtex 器件特性

	XCVU065	XCVU080	XCVU095	XCVU125	XCVU160	XCVU190	XCVU440
逻辑单元	626,640	780,000	940,800	1,253,280	1,621,200	1,879,920	4,432,680
块 RAM (MB)	44.3	50.0	60.8	88.6	115.2	132.9	88.6
DSP Slices	600	672	768	1,200	1,560	1,800	2,880
PCI-E 块	2	4	4	4	4	6	6
Interlaken	3	6	6	6	9	9	0
100G 以太网	3	4	4	6	7	9	3
GTH 16 GB/s 收发器	20	32	32	40	52	60	48
GTY 32.75 GB/s 收发器	20	32	32	40	52	60	0
I/O 引脚	520	832	832	1,040	1,040	1,040	1,456
I/O 电压	1.0~3.3V	1.0~3.3V	1.0~3.3V	1.0~3.3V	1.0~3.3V	1.0~3.3V	1.0~3.3V

下面将对 UltraScale 器件内部设计资源进行进一步说明，以帮助读者充分了解 UltraScale 器件所提供的设计资源。这样，在使用 Vivado 集成开发环境进行 FPGA 的设计时，可以更加高效地利用这些资源，从而进一步提高设计效率。

1.2 可配置逻辑块

可配置的逻辑块（Configurable Logic Block, CLB）是主要的逻辑资源，用于实现时序和组合逻辑电路。

1.2.1 可配置逻辑块的特点

UltraScale 结构的 CLB 提供了高性能和低功耗的可编程逻辑，每个 CLB 连接一个开关矩阵，用于访问通用的布线资源。一个 CLB 包含一个切片（Slice），每个切片提供 8 个 6 输入的查找表和 16 个触发器，切片中的查找表（Look Up Table, LUT）按列排列。UltraScale 架构中包含两种类型的切片，即 SliceL 和 SliceM。如图 1.1 所示，给出了 SLICEL（L 表示逻辑）的内部结构，其中：

- (1) 左侧为 8 个 6 输入的 LUT，从下到上依次用 A、B、C、D、E、F、G、H 标记。
- (2) 右侧为 16 个锁存/触发器资源。
- (3) 包含 F7、F8 和 F9 类型的多路复用器。
- (4) LUT 右侧的部件为一个 8 位的进位链。
- (5) 进位链后的第一列为 F7 类型的多路复用开关，从下到上依次用 F7MUX_AB、F7MUX_CD、F7MUX_EF 和 F7MUX_GH 表示。
- (6) F7 类型的多路复用开关后为上下两个 F8 类型的多路复用开关，从下到上依次用 F8MUX_BOT 和 F8MUX_TOP 表示。
- (7) F8 类型的多路复用开关后为一个 F9 类型的多路复用开关，用 F9MUX 表示。

