

FPGA应用开发 实战技巧精粹

杨跃 编著

华清远见嵌入式培训中心 审校

内容全面：240个FPGA应用开发常用技巧，覆盖FPGA应用开发的方方面面

步骤清晰：详细列出每个操作步骤，按步骤操作即可得出相应结果

团队创作：FPGA开发团队多年开发经验的总结



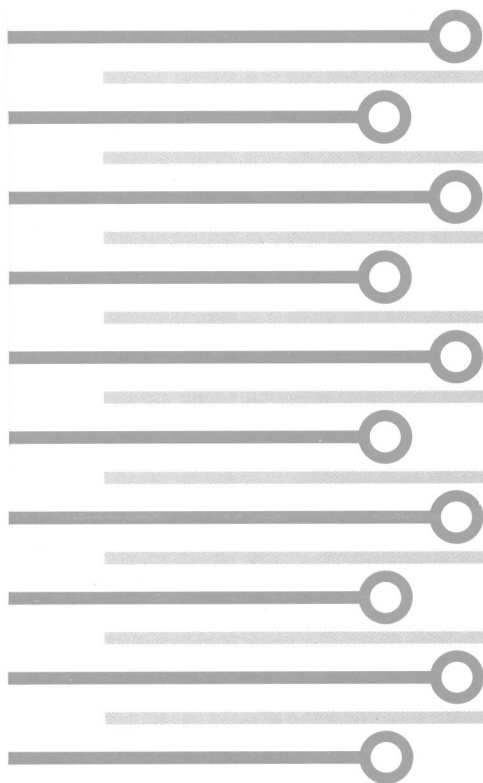
人民邮电出版社
POSTS & TELECOM PRESS

FPGA应用开发 实战技巧精粹

杨跃 编著

华清远见嵌入式培训中心 审校

TP332.1
82



人民邮电出版社
北京

图书在版编目 (CIP) 数据

FPGA应用开发实战技巧精粹 / 杨跃编著. —北京: 人民邮电出版社, 2009.4
ISBN 978-7-115-19661-3

I. F… II. 杨… III. 可程序逻辑器件—系统设计—高等学校—教材 IV. TP332.1

中国版本图书馆CIP数据核字 (2009) 第008137号

内 容 提 要

本书着眼于实践, 介绍了 FPGA 设计开发的大量技巧, 内容包括 Xilinx 开发软件 ISE 的使用技巧、Altera 开发软件 Quartus 的使用技巧、仿真工具 ModelSim 的使用技巧、综合工具 Synplify 的使用技巧以及设计数字电路的一般技巧。

本书立足于实际, 介绍的技巧都是在实际开发中经常使用到的, 作者结合多年的工作经验组织编写了本书, 书中总结了各种使用技巧并且提供了具体的使用方法和实例。读者可以参考书中的技巧, 通过实际动手操作掌握各个技巧的使用方法, 并且在实际开发工作中运用。

本书可以作为高等院校电子工程、通信工程、计算机、自动化等专业的教材, 也可以作为电子工程师的参考手册。

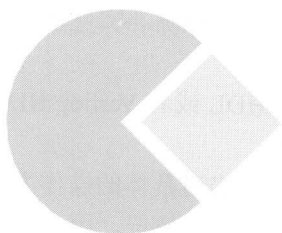
FPGA 应用开发实战技巧精粹

-
- ◆ 编 著 杨 跃
审 校 华清远见嵌入式培训中心
责任编辑 屈艳莲
执行编辑 黄 焱
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市潮河印业有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 23.5
字数: 573 千字 2009 年 4 月第 1 版
印数: 1—3 500 册 2009 年 4 月河北第 1 次印刷

ISBN 978-7-115-19661-3/TP

定价: 45.00 元

读者服务热线: (010)67132692 印装质量热线: (010)67129223
反盗版热线: (010)67171154



前言

写作背景

FPGA (Field Programmable Gate Array) 是 1984 年由 Xilinx 公司发明的一类半定制的通用器件。用户可以通过对 FPGA 器件的编程来实现所需的逻辑功能。FPGA 设计电路有以下几个优点: 电路执行速度快、上市时间短、成本低廉、可靠性高、易于维护升级。

近年来, 随着 FPGA 规模越来越大、速度越来越快, 并且成本也不断降低, 在许多应用中 FPGA 已经开始取代 ASIC, 使 FPGA 的应用领域不断扩大。目前 FPGA 广泛应用于通信、信号处理、嵌入式处理器、图像处理和工业控制等领域。

学习掌握 FPGA 的开发和使用是实践性很强的活动。针对这个特点, 作者从实践出发并结合多年的开发经验编写了本书。书中总结了许多使用 FPGA 设计电路的技巧, 这些技巧包括 Xilinx 开发工具 ISE 和 Altera 开发工具 Quartus 的使用技巧, 仿真工具 ModelSim 的使用技巧, 综合工具 Synplify 的使用技巧, 以及电路设计的基本技巧。

本书主要内容

本书共分 6 章, 各章的内容如下。

第 1 章介绍 FPGA 的基础知识, 包括 FPGA 的基本结构、FPGA 的基本应用, 介绍了 FPGA 的设计开发流程和常用的开发工具, 最后展望了其发展趋势。本书要求读者具有一定的 FPGA 设计基础, 通过这一章的介绍, 读者可以掌握 FPGA 的基础知识, 并且能掌握 FPGA 的基本设计流程, 为后面章节的学习做个铺垫。

第 2 章介绍了 Xilinx 开发工具 ISE 的使用技巧。通过这一章的学习, 读者可以使用 ISE 进行完整的 Xilinx FPGA 设计。如果读者已经会使用 ISE, 也可以参考本章的技巧以便更加熟练地使用 ISE。

第 3 章介绍了 Altera 开发工具 Quartus 的使用技巧。通过这一章的学习, 读者可以使用 Quartus 设计完整的 Altera FPGA。这一章中的技巧也可以作为设计者的参考手册, 供那些掌握了 Quartus 的读者查阅相关技巧。

第 4 章介绍了逻辑电路的设计技巧。这些技巧是进行数字电路设计的通用技巧, 适用于各个系列的 FPGA。读者只有掌握这一章的技巧, 才能设计出性能高、运行稳定、资源利用

率高的电路。

第 5 章介绍了 ModelSim 仿真工具的使用技巧。设计者完成初期开发工作后, 首先需要进行仿真, 确认功能正确, 才能在实际的硬件环境中调试, 否则会浪费过多的时间查找错误。ModelSim 还提供了后仿真的功能, 使设计者能够尽量在设计初期就发现问题。

第 6 章介绍了目前使用最为广泛的综合工具 Synplify 的使用技巧, 包括基本的综合技巧, 综合属性的使用技巧。只有掌握了这一章的技巧, 才能使用综合工具更加准确地实现设计者设计的电路, 充分地体现设计者的设计意图。

本书要求读者有一定的 FPGA 设计基础, 需要读者适当了解 VHDL 或者 Verilog HDL 语言, 这部分知识读者可以参阅相关书籍。

书中的所有技巧都是结合实际开发的, 以实际操作为主线, 通过描述具体的操作方法并结合例子, 力图使读者能够以最直观的方法掌握实际开发中会遇到的各种情况。因为 FPGA 开发有实践性强的特点, 读者一定要多动手、多思考, 只有这样才能更快地掌握 FPGA 开发的技巧, 并培养出自己特定的开发习惯。

本书可以作为高等院校电子工程、通信工程、计算机、自动化等专业的教材, 也可以作为电子工程师的参考手册。

参与本书编写的人员

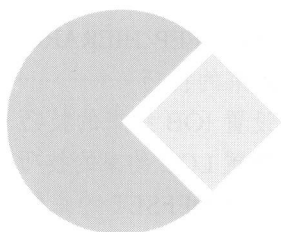
本书第 1 章由杨跃、王磊编写, 第 2 章、第 6 章由杨跃编写, 第 3 章、第 4 章由王磊编写, 第 5 章由王玮磊编写。全书由清华大学吴俊博士、电子科技大学丘志杰博士和北京邮电大学申滨博士统筹规划, 并对全书进行了审校。

在准备本书的过程中, 许多相关领域的专家、工程师和朋友们提供了宝贵的帮助、建议和支持。中国矿业大学刘卫东博士、电子科技集团公司第三十研究所石元兵工程师、AMD 公司王路工程师、阿尔卡特朗讯公司陈国龙工程师对全书的内容进行了审阅, 并提出许多非常有价值的修改建议。周江、向君才、杨钧、陈鹏负责验证了全书的例子, 田雷、赵帅负责对全书的文字和图形进行了仔细的校对, 同时, 参与编写工作的还有刘燕祎、周晶、周丰、梅乐夫、房明浩、王亮、门店宏、吴洋、石峰、张圣亮、邱文勋、刘鲲、矫津毅、林远长、董前程、岂兴明、汤嘉立、刘变红、周建兴、张高煜、赵红波、邓志宝、刘坤、刘明辉、李鹏、白学明、步士建, 在此一并致以诚挚的谢意。

本书编写过程中得到了华清远见嵌入式培训中心的大力支持, 他们为本书的写作提出了许多建议性的意见, 在此表示衷心的感谢。嵌入式系统相关资料、公开视频, 请参见 <http://www.farsight.com.cn/download>。

由于笔者水平有限, 书中错误和疏漏之处在所难免, 恳请广大读者批评指正。本书责任编辑的联系方式是 huang_yan@ptpress.com.cn, 欢迎大家来信交流。

编 者
2009.1



目 录

第 1 章 基础知识	1		
1.1 FPGA 简介	1		
1.2 FPGA 基本结构	2		
1.2.1 FPGA 的基本结构	2		
1.2.2 Altera FPGA 基本结构	2		
1.2.3 Xilinx FPGA 基本结构	5		
1.3 FPGA 的应用	7		
1.4 FPGA 设计流程	7		
1.5 FPGA 的常用开发工具	10		
1.6 FPGA 设计技术展望	11		
1.6.1 未来可编程器件的 发展趋势	11		
1.6.2 未来 EDA 设计方法的 发展趋势	12		
第 2 章 Xilinx FPGA 设计技巧	14		
2.1 ISE 基本使用技巧	14		
2.1.1 新建项目的技巧	14		
2.1.2 新建 HDL 文件的技巧	15		
2.1.3 添加 HDL 文件的技巧	16		
2.1.4 新建原理图设计的技巧	16		
2.1.5 在原理图中调用模块的 技巧	17		
2.1.6 编辑原理图的技巧	18		
2.1.7 用 Constraints Editor 设置 约束的技巧	20		
2.1.8 用 PACE 设置引脚与 区域约束的技巧	23		
2.1.9 使用 XST 进行综合的 技巧	26		
2.1.10 设计实现的技巧	29		
2.1.11 生成下载文件的技巧	30		
2.1.12 下载 FPGA 的技巧	32		
2.2 仿真验证技巧	34		
2.2.1 新建测试平台的技巧	34		
2.2.2 图形化编辑激励信号的 技巧	36		
2.2.3 在 ISE 中仿真的技巧	37		
2.2.4 在 ISE 中调用 Model Sim 的技巧	38		
2.2.5 使用 ModelSim 行为 仿真的技巧	39		
2.3 命令行方式使用 ISE 的技巧	43		
2.3.1 调用命令行的技巧	43		
2.3.2 命令行的语法技巧	44		
2.3.3 NGDBUILD 命令使用 技巧	45		
2.3.4 MAP 命令使用技巧	47		
2.3.5 PAR 命令使用技巧	49		
2.3.6 BITGEN 命令使用技巧	50		
2.3.7 TRACE 命令使用技巧	51		

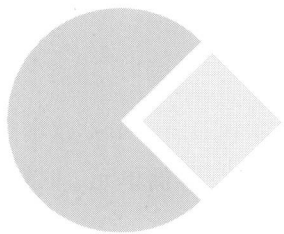
2.3.8	DATA2MEM 命令使用技巧	52	2.6.10	设置 IOBDELAY 约束的技巧	88
2.3.9	自动执行命令行的技巧	52	2.6.11	设置 KEEP 约束的技巧	89
2.4	CORE Generator 使用技巧	55	2.6.12	设置 IOSTANDARD 约束的技巧	89
2.4.1	新建 CORE Generator 项目的技巧	55	2.6.13	设置 KEEP_HIERARCHY 约束的技巧	90
2.4.2	新建 IP 的技巧	58	2.6.14	设置 IOB 约束的技巧	91
2.4.3	修改已有 IP 的参数的技巧	59	2.6.15	设置 LOC 约束的技巧	92
2.4.4	Architecture Wizard 使用技巧	60	2.6.16	设置 OFFSET 约束的技巧	94
2.4.5	在设计中例化 IP 的技巧	62	2.6.17	设置 PERIOD 约束的技巧	97
2.4.6	选择不同版本 IP 的技巧	63	2.6.18	设置 PIN 约束的技巧	99
2.5	使用 Xilinx 硬件资源的技巧	64	2.6.19	设置 TIMEGRP 约束的技巧	99
2.5.1	DCM 使用技巧	64	2.7	Chipscope 调试技巧	100
2.5.2	PMCD 使用技巧	66	2.7.1	使用 Chipscope Inserter 的技巧	101
2.5.3	BLOCK RAM 使用技巧	69	2.7.2	生成 ChipScope 下载文件的技巧	105
2.5.4	分布式 RAM 使用技巧	72	2.7.3	使用 ChipScope 下载 FPGA 的技巧	106
2.5.5	FIFO 使用技巧	72	2.7.4	使用 ChipScope Analyzer 的技巧	107
2.5.6	IDDR 使用技巧	74	2.7.5	直接从 ISE 调用 ChipScope 的技巧	109
2.5.7	ODDR 使用技巧	79	2.7.6	使用 ICON Core 的技巧	110
2.6	设置约束的技巧	82	2.7.7	使用 ILA Core 的技巧	112
2.6.1	使用 UCF 文件的技巧	82	第 3 章	Altera FPGA 设计技巧	116
2.6.2	编写 UCF 文件的语法技巧	83	3.1	使用 Altera 器件的技巧	116
2.6.3	设置 TNM_NET 分组约束的技巧	84	3.1.1	PLL 的使用技巧	116
2.6.4	设置 TNM 分组约束的技巧	84	3.1.2	LVDS 的使用技巧	118
2.6.5	设置 TIMESPEC 时序约束的技巧	85	3.1.3	FIFO 的使用技巧	121
2.6.6	设置 AREA_GROUP 约束的技巧	85	3.1.4	Embedded RAM 的使用技巧	122
2.6.7	设置 DRIVE 约束的技巧	87	3.1.5	Shift Register 的使用技巧	124
2.6.8	设置 IBUF_DELAY_VALUE 约束的技巧	87			
2.6.9	设置 IFD_DELAY_VALUE 约束的技巧	88			

3.1.6	动态相位对准 (DPA) 的使用技巧	125	3.5.1	配置编译器的技巧	151
3.1.7	Virtual JTAG 的使用技巧	126	3.5.2	创建引脚分配的技巧	153
3.1.8	DSP 块的使用技巧	127	3.5.3	查看编译结果的技巧	154
3.2	使用 Quartus 开发的基本技巧	128	3.5.4	设置布局布线选线的技巧	155
3.2.1	运行新项目向导的技巧	128	3.5.5	使用资源优化向导的技巧	157
3.2.2	设置项目参数的技巧	129	3.6	时序分析的技巧	159
3.2.3	添加 HDL 设计的技巧	131	3.6.1	查看时序分析结果的技巧	159
3.2.4	新建 HDL 设计的技巧	132	3.6.2	执行多时钟时序分析的技巧	160
3.2.5	运行时序向导设置时序约束的技巧	132	3.6.3	设定多周期路径的技巧	161
3.2.6	编译设计的技巧	134	3.6.4	查看特殊路径时序结果的技巧	161
3.2.7	配置 FPGA 的技巧	135	3.6.5	使用时序优化器的技巧	163
3.3	使用 Quartus 输入设计的技巧	136	3.6.6	使用 LogicLock 改善时序要求的技巧	164
3.3.1	使用版本兼容的设计的技巧	136	3.6.7	使用渐进式编译改善时序的技巧	165
3.3.2	使用 Block Editor 的技巧	137	3.7	使用 Quartus 仿真的技巧	166
3.3.3	使用 Text Editor 的技巧	138	3.7.1	设置仿真工具的技巧	166
3.3.4	使用 Symbol Editor 的技巧	138	3.7.2	创建波形文件的技巧	167
3.3.5	使用 Megafunction 的技巧	140	3.7.3	生成仿真输出文件的技巧	167
3.3.6	使用语言模板的技巧	140	3.7.4	生成仿真库的技巧	168
3.4	使用 Quartus 综合的技巧	142	3.7.5	分析仿真结果的技巧	170
3.4.1	设置综合属性的技巧	142	3.7.6	使用仿真器工具的技巧	170
3.4.2	综合 VHDL 和 Verilog 设计的技巧	142	3.8	使用 Quartus 下载的技巧	172
3.4.3	使用编译指令和属性的技巧	143	3.8.1	创建下载文件的技巧	172
3.4.4	使用逻辑选项的技巧	143	3.8.2	加载文件格式转换的技巧	173
3.4.5	优化综合网表的技巧	146	3.8.3	使用 Programmer 配置器件的技巧	174
3.4.6	检查设计可靠性的技巧	147	3.8.4	建立辅助下载文件的技巧	175
3.4.7	查看状态机综合结果的技巧	148	3.9	SignalTap II 调试技巧	176
3.4.8	查看综合报告的技巧	149	3.9.1	设置和运行 Logic Analyzer 的技巧	176
3.4.9	使用增量综合的技巧	151			
3.5	使用 Quartus 布局布线的技巧	151			

3.9.2	增量式编译使用 Signal TapII 的技巧	178	4.1.2	Bottom-Up 方式的设计技巧	198
3.9.3	使用外部逻辑分析仪的技巧	178	4.1.3	VHDL 设计 FPGA 的技巧	198
3.9.4	使用 SignalProbe 的技巧	179	4.1.4	Verilog HDL 设计 FPGA 的技巧	200
3.9.5	使用 RTL 查看器的技巧	180	4.1.5	状态机设计的技巧	206
3.9.6	使用技术映射查看器的技巧	181	4.2	数字系统设计技巧	207
3.9.7	使用芯片编辑器的技巧	182	4.2.1	同步电路设计技巧	207
3.10	使用 Altera IP Core 的技巧	183	4.2.2	异步时钟域数据同步的技巧	209
3.10.1	启动 MegaWizard Plug-In 的技巧	183	4.2.3	亚稳态	211
3.10.2	充分利用 IP 宏功能的技巧	184	4.2.4	系统原则的技巧	211
3.10.3	生成所需 IP 的技巧	185	4.2.5	硬件设计原则的技巧	213
3.10.4	例化 IP 的技巧	187	4.2.6	选择 if 语句与 case 语句的技巧	214
3.10.5	推断 IP 宏功能的技巧	187	4.2.7	分离组合电路与时序电路的技巧	216
3.10.6	使用黑盒子方式例化宏功能的技巧	188	4.2.8	乒乓操作的技巧	219
3.10.7	按推断进行例化的技巧	188	4.2.9	串并转换的技巧	221
3.10.8	使用 Clear-Box 的技巧	188	4.2.10	流水线操作设计的技巧	221
3.11	设置约束的技巧	189	4.3	代码风格技巧	222
3.11.1	设置引脚约束的技巧	189	4.3.1	VHDL 的编码风格技巧	222
3.11.2	使用 Assignment Editor 的技巧	189	4.3.2	Verilog HDL 的编码风格技巧	233
3.11.3	设置时间约束的技巧	190	4.3.3	命名的技巧	240
3.11.4	设置位置约束的技巧	192	4.3.4	添加注释的技巧	241
3.11.5	设置区域的技巧	192	4.3.5	模块划分的技巧	242
3.11.6	导出和导入约束文件的技巧	193	4.3.6	模块重用的技巧	243
3.11.7	验证引脚分配的技巧	194	4.3.7	编写可综合代码的技巧	244
第 4 章	逻辑电路设计技巧	196	4.4	提高速度的技巧	246
4.1	FPGA 设计的基本技巧	196	4.4.1	设置速度约束的技巧	246
4.1.1	Top-Down 方式的设计技巧	196	4.4.2	专用资源提高速度的技巧	247
			4.4.3	分配关键路径的技巧	247
			4.4.4	进行特殊约束的技巧	249
			4.4.5	减少逻辑级数的技巧	249
			4.4.6	分割组合逻辑的技巧	250

4.4.7	转移组合逻辑的技巧	251	5.5.3	VCOM 命令使用技巧	278
4.5	降低面积的技巧	252	5.5.4	VLOG 命令使用技巧	280
4.5.1	模块划分的技巧	252	5.5.5	VSIM 命令使用技巧	281
4.5.2	复用模块的技巧	253	5.5.6	FORCE 命令使用技巧	282
4.5.3	利用代码风格降低 面积的技巧	254	5.5.7	ADD WAVE 命令使用 技巧	284
4.5.4	使用分布式 RAM 的 技巧	255	5.5.8	RUN 命令使用技巧	286
4.5.5	布局布线的技巧	256	5.5.9	DO 文件使用技巧	287
4.5.6	面积和速度的平衡与 互换技巧	257	5.6	ModelSim 脚本仿真技巧	287
第 5 章	ModelSim 仿真技巧	259	5.6.1	Testbench 的创建技巧	287
5.1	ModelSim 介绍	259	5.6.2	ModelSim DO 脚本文件 编写技巧	298
5.2	图形界面使用技巧	260	5.6.3	运行 ModelSim DO 脚本的技巧	301
5.2.1	使用 Wizard 创建工程的 技巧	260	第 6 章	Synplify 综合工具使用技巧	302
5.2.2	编译文件的技巧	262	6.1	Synplify Pro 基本使用技巧	302
5.2.3	启动仿真的技巧	262	6.1.1	新建 HDL 源文件的 技巧	302
5.2.4	产生信号激励的技巧	263	6.1.2	使用文本编辑器的技巧	304
5.2.5	查看波形的技巧	265	6.1.3	检查 HDL 源文件的 技巧	304
5.3	编译 ModelSim 仿真库的技巧	265	6.1.4	使用外部文本编辑器的 技巧	306
5.3.1	在 ModelSim 中编译 Xilinx 仿真库的技巧	266	6.1.5	设置工程参数的技巧	307
5.3.2	在 ISE 中编译 Xilinx 仿 真库的技巧	267	6.1.6	综合及查看综合结果的 技巧	308
5.3.3	使用脚本编译 Xilinx 仿 真库的技巧	269	6.1.7	混合语言综合的技巧	311
5.3.4	在 ModelSim 中编译 Altera 仿真库的技巧	270	6.1.8	新建设计实现的技巧	313
5.4	使用 FPGA 开发工具调用 ModelSim 的技巧	271	6.2	分析综合结果的技巧	313
5.4.1	在 ISE 中调用 Model Sim 的 技巧	271	6.2.1	使用日志观察窗口的 技巧	313
5.4.2	在 Quartus 中调用 ModelSim 的技巧	273	6.2.2	使用信息查看器的技巧	314
5.5	命令行方式仿真技巧	277	6.2.3	分析关键路径时序的 技巧	316
5.5.1	VLIB 命令使用技巧	277	6.2.4	使用 Stand-alone 时序 分析的技巧	317
5.5.2	VMAP 命令使用技巧	278	6.3	利用脚本运行 Synplify 的 技巧	319

6.3.1	使用 add_file 命令的技巧	319	6.4.8	使用 syn_encoding 属性的技巧	335
6.3.2	使用 assign_to_region 命令的技巧	320	6.4.9	使用 syn_hier 属性的技巧	337
6.3.3	指定 constraint_file 命令的技巧	321	6.4.10	使用 syn_netlist_hierarchy 属性的技巧	338
6.3.4	使用 create_region 命令的技巧	322	6.4.11	使用 syn_ramstyle 属性的技巧	339
6.3.5	使用 gen_env 命令的技巧	322	6.4.12	使用 syn_loc 属性的技巧	340
6.3.6	使用 impl 命令的技巧	322	6.4.13	使用 syn_noclockbuf 属性的技巧	341
6.3.7	使用 project 命令的技巧	324	6.4.14	使用 syn_keep 属性的技巧	343
6.3.8	使用 project_file 命令的技巧	326	6.4.15	使用 syn_noprune 属性的技巧	346
6.4	综合属性 Attribute	327	6.4.16	使用 syn_preserve 属性的技巧	350
6.4.1	添加综合属性的技巧	327	6.4.17	使用 translate_off/translate_on 属性的技巧	352
6.4.2	使用 SCOPE 窗口的技巧	328	6.5	Synplify 的高级综合技巧	354
6.4.3	使用 altera_chip_pin_lc 属性的技巧	329	6.5.1	共享资源的技巧	354
6.4.4	使用 black_box_pad_pin 属性的技巧	330	6.5.2	使用流水线的技巧	355
6.4.5	使用 black_box_tri_pins 属性的技巧	332	6.5.3	使用 Retiming 的技巧	357
6.4.6	使用 syn_allow_retiming 属性的技巧	333	6.5.4	插入探针 Probes 的技巧	358
6.4.7	使用 syn_black_box 属性的技巧	334	6.5.5	使用推论 RAM 的技巧	359
			6.5.6	使用推论移位寄存器的技巧	363



第1章 基础知识

本章介绍 FPGA 的基础知识, 包括 FPGA 的结构、特点以及 FPGA 的发展趋势。通过本章的介绍, 读者可以对 FPGA 有初步的了解。

1.1 FPGA 简介

FPGA (Field Programmable Gate Array, 现场可编程门阵列) 是 1984 年由 Xilinx 公司发明的, 它是一类半定制的通用器件, 用户可以通过对 FPGA 器件编程实现所需的逻辑功能。简而言之, FPGA 就是可以再编程的芯片。通过改变芯片内部的逻辑电路和布线资源, 不需要修改电路板即可更改硬件电路功能。

相对于 CPLD (Complex Programmable Logic Device, 复杂可编程逻辑器件) 而言, FPGA 中的寄存器资源比较丰富, 更适合同步时序电路较多的数字系统。在这两类可编程逻辑器件中, CPLD 提供的逻辑资源较少, 而 FPGA 提供了最高的逻辑密度、最丰富的特性和极高的性能。FPGA 已经在通信、消费电子、医疗、工业和军事等各应用领域中占据了重要地位。

相对于 ASIC (Application Specific Integrated Circuits, 专用集成电路) 而言, FPGA 是半定制的通用器件。如果需要改变电路功能, 不需要花费重新设计 ASIC 的时间。

使用 FPGA 设计电路有以下几个优点。

- 电路执行速度快。FPGA 内部是通过对电路编程生成逻辑电路来实现功能的, 这点与处理器编程是不相同的。处理器是串行执行的, 但是 FPGA 通过电路实现功能, 是并行执行, 因此 FPGA 的运行速度大大高于通用处理器或者 DSP。

- 上市时间短。由于 FPGA 的在线可编程特性, 设计者不用进行 ASIC 设计的冗长构建过程; 而且由于设计软件性能不断提高, 设计者可以在更高的抽象层级进行设计; 不同的 FPGA 生产厂商还提供了大量的 IP Core, 这些都使 FPGA 设计更快、更方便。

- 成本低廉。随着电子技术的发展, 基于电子技术的各种应用也在改变。使用 FPGA 可以在不修改电路板的前提下修改电路实现, 而且 FPGA 相对 ASIC 的重新开发的费用非常低廉, 所以使用 FPGA 进行设计的成本相对较低。

- 可靠性高。FPGA 都是经过专门验证的半定制通用器件, 因而具有较高的可靠性。

• 易于维护升级。FPGA 芯片具有即时升级 (Field-Upgradable) 特性, 而且可以通过在 FPGA 配置芯片中存储多个配置文件实现多种电路功能, 还可以通过网络进行远程配置。这些特性使得 FPGA 易于维护和升级。

1.2 FPGA 基本结构

1.2.1 FPGA 的基本结构

通常 FPGA 包含几种主要资源: 逻辑块、输入/输出块、布线资源、存储单元、时钟管理模块以及专用硬件资源。

• 逻辑块中一般包含查找表 LUT 和寄存器。LUT 本质上就是一个 RAM, 目前 FPGA 中多使用四输入的 LUT, 所以每一个 LUT 可以看成是一个有 4 位地址线的 16×1 的 RAM。当用户设计一个逻辑电路以后, PLD/FPGA 开发软件会自动计算逻辑电路的所有可能的结果, 并把结果事先写入 RAM。每输入一组信号进行逻辑运算就等于输入一个地址进行查表, 找出地址对应的内容, 然后输出。

- 输入/输出块可以兼容多种电平, 并且包含专用 I/O 寄存器资源和专用的延迟单元。
- 布线资源连接 FPGA 内的逻辑块、输入/输出块、存储单元等硬件资源。
- 存储单元是专用资源, 使用这些专用存储资源不会占用逻辑资源。
- 时钟管理模块提供高性能时钟控制, 可用于锁定时钟、频率合成。

在接下来的两小节中会分别介绍 Altera 和 Xilinx 系列 FPGA 的基本结构。

1.2.2 Altera FPGA 基本结构

与通常的 FPGA 一样, Altera 的 FPGA 包括逻辑块、输入/输出块、布线资源、存储单元、时钟管理模块以及专用硬件资源。图 1.1 所示是这些资源的位置分布情况。

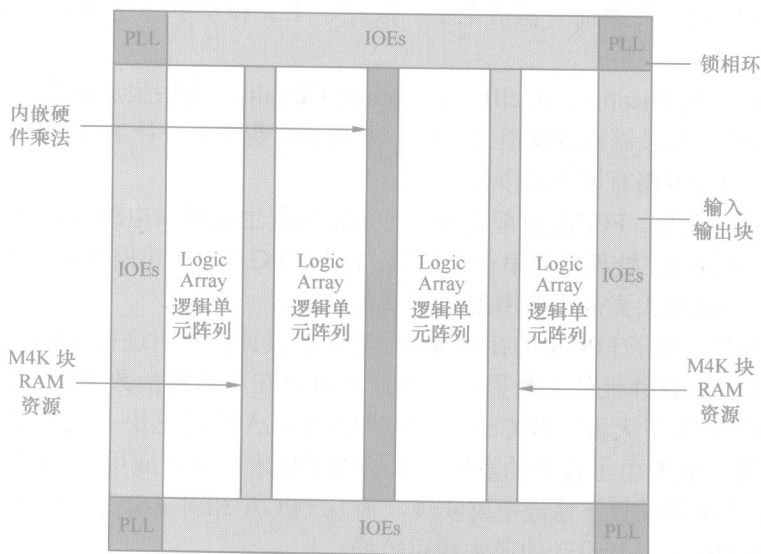


图 1.1 Cyclone 系列 FPGA 平面布置图

可以看到逻辑块以阵列的形式分布在 FPGA 中，锁相环位于芯片的四个角，存储块和嵌入式乘法器位于专门的列，输入/输出 I/O 单元位于芯片的四周。

(1) 在 Altera 的 FPGA 中，最小的逻辑块是 LE (Logic Element)，每个 LE 包括如下的特性。

- 包含一个四输入的查找表 LUT，用于实现一个四变量的逻辑函数。
- 包含一个可编程的寄存器。
- 有一条进位链。
- 有一条寄存器链接。
- 可以驱动各种连接类型，包括本地连接、行连接、列连接、寄存器连接和直接互连。
- 支持寄存器包装。
- 支持反馈寄存器。

Altera 的 LE 的结构如图 1.2 所示。

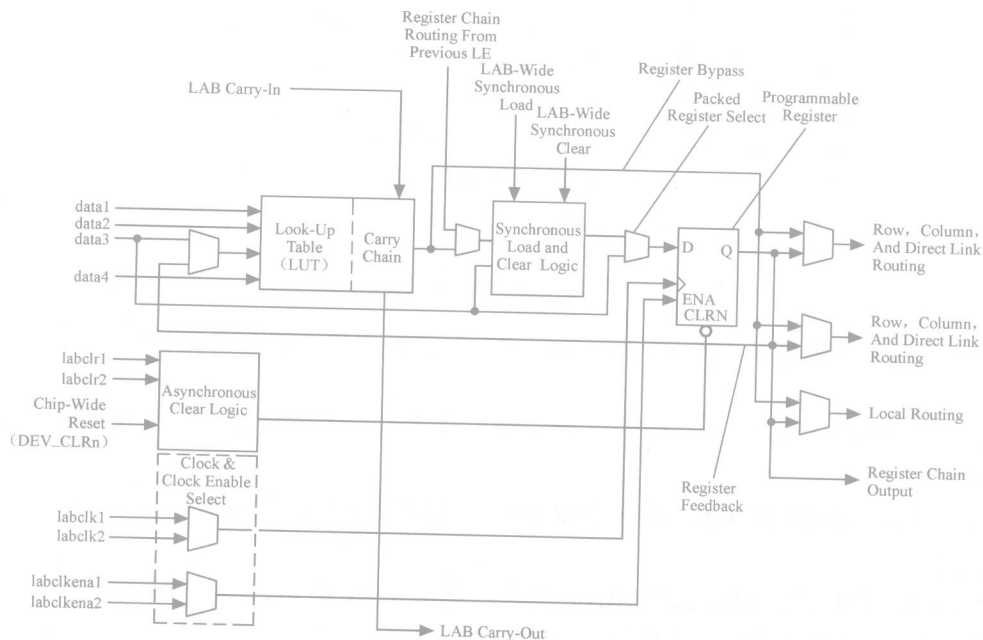


图 1.2 Cyclone 器件 LE 结构

(2) 输入/输出单元 IOE 是芯片与外界电路的接口部分，包括如下特性。

- 支持差分 and 单端 I/O 标准。
- 支持 LVTTTL、LVCOMS、SSTL、HSTL、LVDS、LVPECL 和 PCI 等电气标准。
- 可以调整上拉、下拉电阻。
- 有三态缓冲器。
- 可编程输入、输出延迟。
- 支持高速率接口。

图 1.3 所示为 Altera 的 IOE 结构。

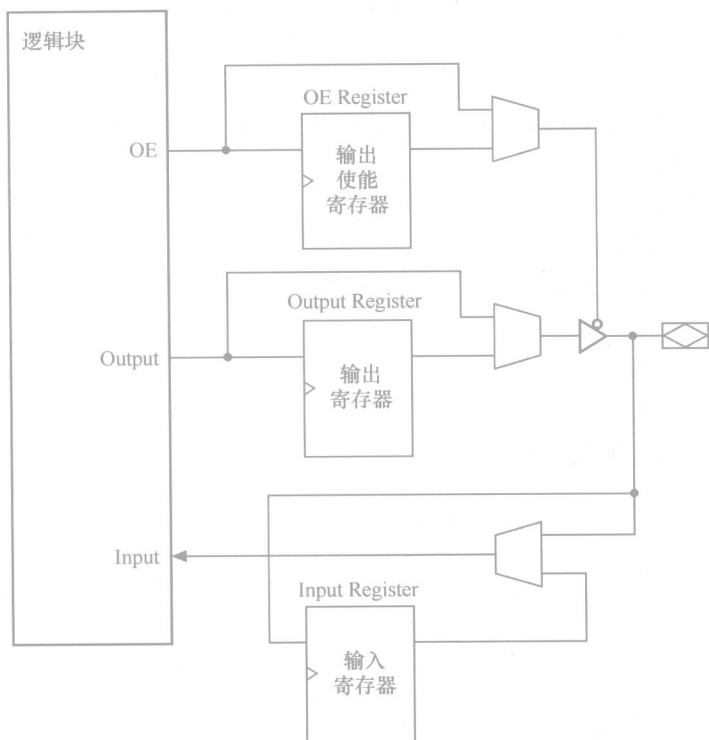


图 1.3 Altera IOE 结构

(3) 布线资源连接芯片内部所有的硬件单元，连线的长度和工艺决定着信号的驱动能力和传输延时。

在设计过程中，设计者一般不需要干预布线资源，设计软件会自动选择布线资源进行布线。

(4) Altera 的 FPGA 中有内嵌的块 RAM，可以配置为单口 RAM、双口 RAM、FIFO 等存储单元。

以 Cyclone II 器件的嵌入式块 RAM 为例，这些 M4K RAM 在芯片中以列的方式存在，形成 RAM 列。M4K 存储块包括输入寄存器和输出寄存器以提高读写性能，输出寄存器可以被旁路，但是输入寄存器必须使用。

(5) Altera 器件提供全局时钟网络和一定数量的 PLL。

以 Cyclone II 器件为例，Cyclone II 器件的时钟网络包括如下特性。

- 多达 16 个全局时钟网络。
- 有 4 个 PLL (Phase-Locked Loops)。
- 全局时钟网络动态时钟资源选择。
- 全局时钟动态使能。

图 1.4 所示为 Cyclone II 的全局时钟网络示意图，时钟信号通过全局时钟网络分配到各个硬件单元，保证了各个硬件单元的时钟抖动最小。

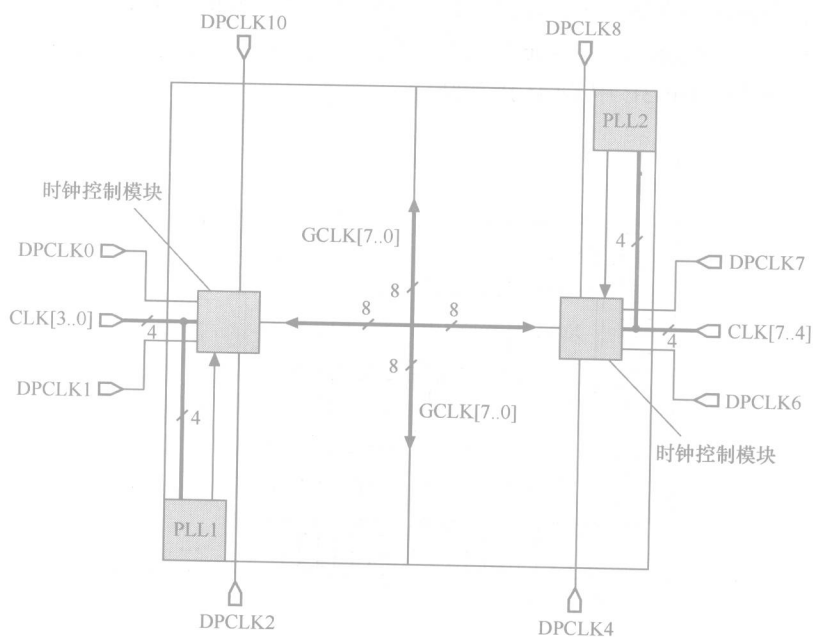


图 1.4 Cyclone II 全局时钟网络

1.2.3 Xilinx FPGA 基本结构

Xilinx 的 FPGA 包括逻辑块、输入/输出块、布线资源、存储单元、时钟管理模块以及专用硬件资源。

如图 1.5 所示，以 Spartan-3E 为例说明 Xilinx FPGA 结构。从图中可以看到逻辑块分布于整个芯片，在芯片的顶部和底部有时钟管理模块 DCM，块 RAM 和乘法器位于专门的列，输入/输出块位于芯片的四周。

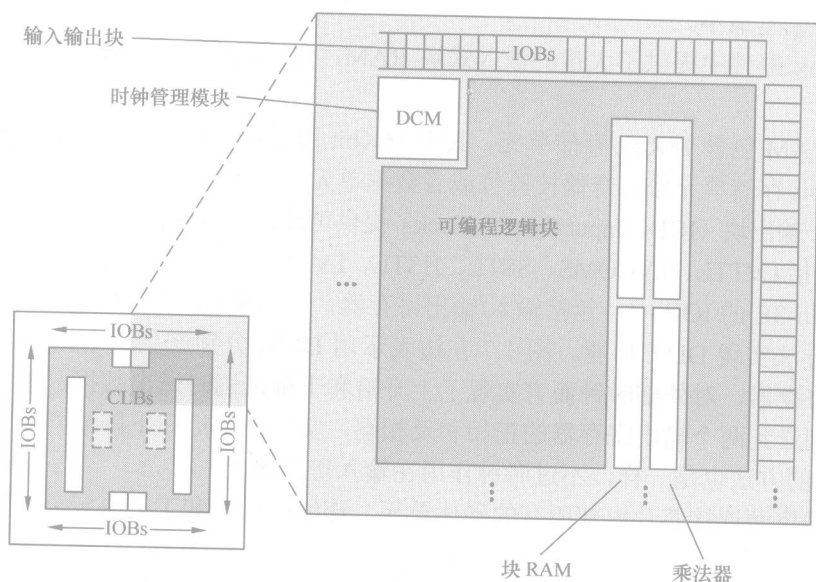


图 1.5 Spartan-3E 的结构

(1) 可配置逻辑块 (CLB, Configurable Logic Blocks) 是 Xilinx Spartan-3E 系列 FPGA 的基本逻辑单元。它包括 4 个 Slice, 每个 Slice 包括两个查找表用于执行组合逻辑, 另外还有两个触发器。

图 1.6 所示为 Spartan-3E 中 CLB 排列结构。

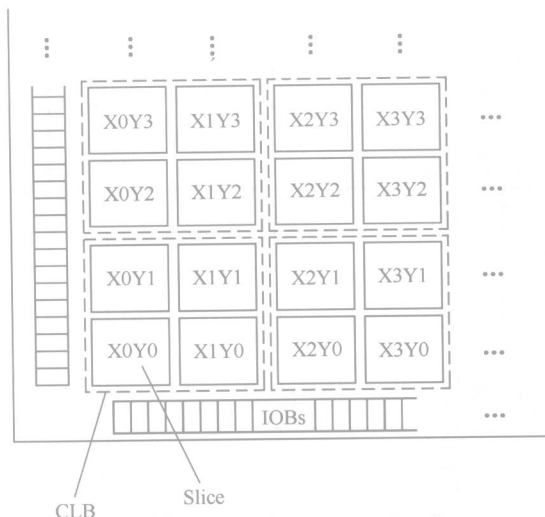


图 1.6 Spartan-3E 中 CLB 排列结构

在 Slice 中 LUT 可以作为 16 位的存储器 RAM16, 也可以作为 16 位的移位寄存器 SRL16。

(2) 在 Xilinx 器件中, 提供全局时钟布线资源和数字时钟管理 (DCM, Digital Clock Managers) 模块。

DCM 可以实现时钟锁定功能、频率合成功能、分频和倍频功能。

Xilinx 器件通过全局时钟缓冲器 BUFG 将时钟分配到专用的全局时钟网络, 保证各个模块之间的时钟偏差最小。

(3) Xilinx 器件中提供了大量的嵌入式块 RAM, 以实现单口 RAM、双口 RAM、FIFO 等存储应用。

每个块 RAM 包含 18Kbit 存储单元, 其中 16Kbit 用于数据存储, 在特定的应用中, 可以将额外的 2Kbit 存储单元作为奇偶校验位或者数据单元。

(4) 输入/输出块 (IOB, Input/Output Block) 提供可编程单向或者双向端口, 支持一系列的电气标准, 包括 LVTTTL、LVCOMS、SSTL、HSTL、LVDS、LVPECL 和 PCI 等。

在 Spartan-3E 的 IOB 中, 包括输入/输出寄存器, 可以实现 DDR 功能。如图 1.7 所示, 可以有两种方法实现 DDR 功能。图 1.7 左边表示用 DCM 分别产生两个相差 180° 的时钟送到两个输出寄存器, 对外部时钟而言实现了上升沿和下降沿同时送出数据; 图 1.7 右边是将一个时钟分别送到两个输出寄存器的正相和反相端, 实现上升沿和下降沿同时送出数据。实现 DDR 输入的方法也是一样, 不过时钟作用在输入寄存器上。

(5) 布线资源连接芯片内部所有的硬件单元, 连线的长度和工艺决定着信号的驱动能力和传输延时。

设计过程中, 设计者一般不需要干预布线资源, 设计软件会自动选择布线资源进行布线。