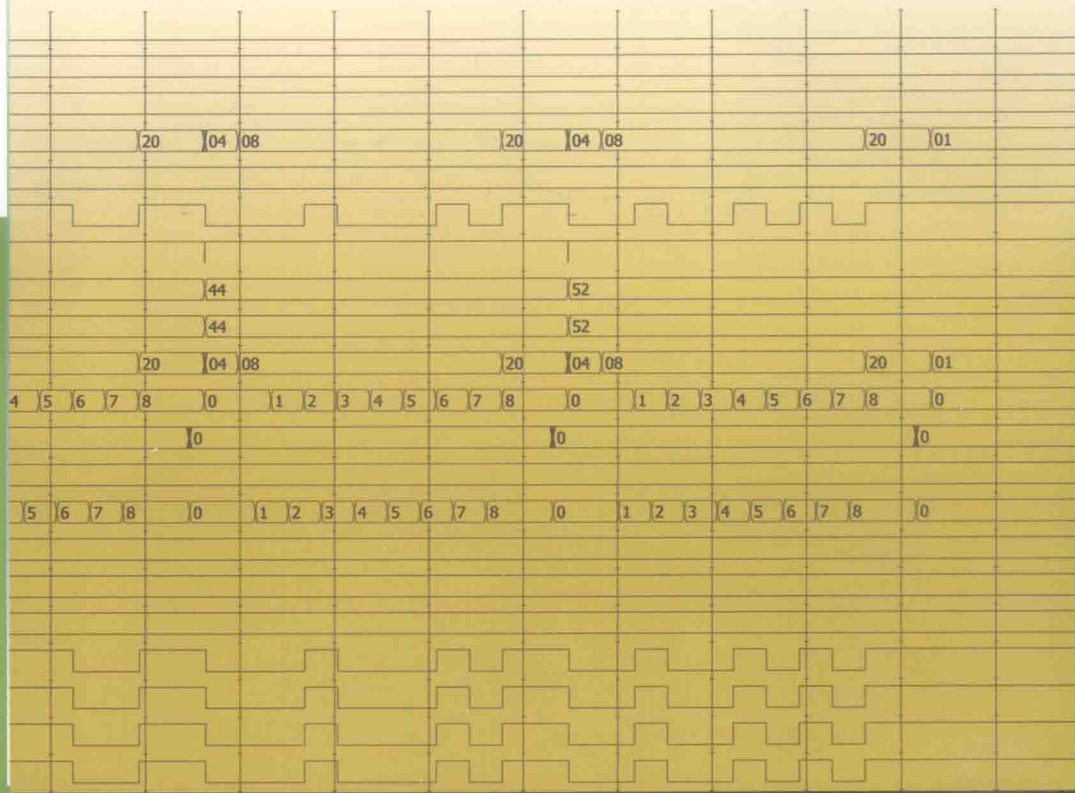


F P G A 应用 技术 丛书

FPGA项目开发 实战讲解

李宪强 编著



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>

FPGA 应用技术丛书

FPGA 项目开发实战讲解

李宪强 编著

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书立足于工程实践经验,首先介绍FPGA的原理、语法、设计技巧,然后详细介绍7项常用技术的实现。每项技术从以下几方面介绍:相关技术介绍,FPGA方案框图设计,子模块设计,仿真和测试,项目开发过程中遇到的问题、定位和解决措施,其中插入了大量的状态机转换图和关键代码,方便读者学习。

本书适合FPGA初学者,或者对FPGA设计有一定经验,但对复杂的项目经常感到束手无策的开发人员阅读,也可作为相关领域从业人员的工程参考书。

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

FPGA项目开发实战讲解/李宪强编著. —北京:电子工业出版社,2015.4

(FPGA应用技术丛书)

ISBN 978-7-121-25640-0

I. ①F… II. ①李… III. ①可程序逻辑器件—系统开发 IV. ①TP332.1

中国版本图书馆CIP数据核字(2015)第044224号

策划编辑:曲 昕

责任编辑:康 霞

印 刷:北京京科印刷有限公司

装 订:北京京科印刷有限公司

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:16.25 字数:416千字

版 次:2015年4月第1版

印 次:2015年4月第1次印刷

定 价:48.00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。

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

服务热线:(010)88258888。

前言

笔者发现市场上关于 FPGA 的书籍有以下几种：介绍语法、介绍软件安装和使用、介绍 FPGA 芯片本身、介绍 FPGA 小项目（如开发 100 例，这样的小项目不能使读者学会真正的 FPGA 开发）、介绍 FPGA 大项目（如以太网开发、PCI_E 系统开发，但介绍太笼统，没有介绍读者关心的具体操作步骤，最终读者还是不能掌握实际的开发技巧），所以笔者立足于让读者真正能掌握 FPGA 的开发技巧，以中等难度的项目为例，详细介绍开发中的每个环节，读者读完后，完全可以亲自动手完成一个中等难度的项目，建立自信，真正学会 FPGA 开发。

本书涉及的所有技术点都是笔者在项目设计中亲自完成的。其中用 7 章从易到难介绍了 7 个开发项目，每个项目从以下几方面介绍：相关技术、FPGA 方案设计、子模块设计、仿真和测试、项目开发过程中遇到的问题、定位和解决措施。在编写过程中，插入了大量的状态机转换图、仿真结果和关键代码，方便读者学习，同时还加上了笔者在开发过程中遇到的实际问题及解决问题的手段、机理分析。

本书的读者群是 FPGA 初学者，或者对 FPGA 设计有一定经验，但对复杂的项目经常感到束手无策的开发人员，以及想进一步提高的朋友。

在 7 个开发项目中，异步 FIFO 控制器是每个 FPGA 开发人员都需要面对的问题，一些从事 FPGA 开发多年的朋友仍然不能掌握其中的细节，虽然看似简单，但真正弄清楚后能提高很多。

串口通信控制器是比较常用的模块，80% 以上的项目都需要用到串口进行通信、控制、参数配置等，所以串口的开发比较有代表性。

A/D 采集控制器也是非常常用的，但不同 A/D 芯片的工作模式不一样，笔者在开发过程中遇到了一些障碍，所以拿出来和读者分享，希望对读者有一定的启发作用，遇到调试不顺的时候，需要慢慢分析，不能轻易打退堂鼓，也不能对技术指标打折扣，要 100% 完成才可以。

HDLC 也是非常常用的技术，在商用网络通信及军工领域中应用非常广泛。

SDRAM 应用也非常广泛，学会了这个模块，理解 DDR SDRAM 就容易很多了。

NAND FLASH 作为大容量存储器，在开发中有一定的难度，第 9 章将告诉读者面对一个相对复杂的项目，如何一步步分解，把复杂的工程分解成多个相对简单的模块。

1553B 控制器在航空航天领域中用得非常多，目前，在这个领域实现 1553B 控制器主要用进口 ASIC 芯片，但价格昂贵，国产化和 FPGA 化的趋势正在一步步加深，军工类院校和研究所对这项技术会非常感兴趣。

本书的编写重点不是介绍语法，而是讲解实际项目，每个项目的编写都有大量的状态转移图、时序仿真图和测试结果，异步 FIFO、串口控制器、A/D 和 D/A 控制器、HDLC 控制器四部分给出了完整的源代码，后面几章限于篇幅只列出了实现的关键代码，但是读者根据比较详细的设计描述和状态机，可以轻易地将代码编写出来。对照代码，用心领悟，沉下心，集中精力搞定其中一两个项目，就会有质的提高。

我的联系方式：lxqiang_2008@aliyun.com，期待和读者进一步沟通交流。

编著者

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396; (010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

目录

CONTENTS

第 1 章	FPGA 介绍	1
1.1	FPGA 的优势	1
1.2	FPGA 典型结构图	2
1.3	LUT 原理	6
1.4	FPGA 上电配置过程时序图	7
1.5	FPGA 基本开发流程	10
第 2 章	Verilog 语法介绍	13
2.1	Verilog HDL 语言简介	13
2.1.1	什么是 Verilog HDL	13
2.1.2	Verilog HLC 的历史	13
2.2	Verilog HDL 和 VHDL 语言比较	14
2.3	Verilog 语法介绍	15
2.3.1	模块	15
2.3.2	常量	16
2.3.3	parameter 与 define 的区别	18
2.3.4	reg 和 wire	19
2.3.5	操作符	21
2.3.6	assign 和 always 语句	21
2.3.7	if 和 case 语句	23
2.3.8	循环语句	25
2.3.9	模块实例化	25
2.3.10	预处理指令	26
2.3.11	系统任务	26
2.3.12	task 和 function 语句的区别	29
2.3.13	文件操作	30
第 3 章	FPGA 开发必要的技术积累	33
3.1	锁存器、触发器	33

3.2	时序的基本概念	34
3.2.1	建立和保持时间	34
3.2.2	同步系统中时序分析	34
3.3	异步时钟域和亚稳态	37
3.3.1	什么是亚稳态	37
3.3.2	亚稳态发生场合	37
3.3.3	亚稳态危害	38
3.3.4	怎么减少亚稳态的发生	38
3.3.5	案例：亚稳态导致通信异常	39
3.4	状态机的设计	39
3.5	复位信号	40
3.5.1	同步复位和异步复位比较	40
3.5.2	案例：和复位信号相关的项目	41
3.6	串并转换和并串转换	42
3.7	流水技术	42
3.7.1	流水技术原理	42
3.7.2	案例：NAND FLASH 流水实现高速存储	43
3.8	乒乓操作	44
3.9	双向端口使用	45
3.9.1	数据总线的设计	45
3.9.2	程序设计	45
3.9.3	案例：MCU 接口程序	46
3.10	Xilinx 原语介绍	49
3.10.1	时钟组件	49
3.10.2	吉比特收发器组件	50
3.10.3	I/O 端口组件	50
3.10.4	IBUFDS	51
3.10.5	IDELAY	52
3.10.6	OBUFDS	53
3.11	DCM 原理	54
3.12	RAM 核的使用	56
3.13	FPGA 时序约束	60
3.13.1	FPGA 约束介绍	60
3.13.2	案例：500MSPS 的 A/D 转换器	63
3.14	生成 IP 核	70
3.14.1	生成 IP 核的方法	70
3.14.2	案例：生成 IP 核的具体操作过程	71

3.15	保持层次关系	73
3.16	Chipscope ICON 和 ILA	75
3.16.1	Chipscope 介绍	75
3.16.2	案例: 以 DSP 接口程序为例介绍使用流程	77
3.17	Verilog 编码风格注意事项总结	79
3.18	FPGA 学习步骤	80
3.19	FPGA 工作方式	81
第 4 章	FIFO 控制器设计	83
4.1	FIFO 相关的概念	83
4.2	设计指标	84
4.3	同步 FIFO 设计	84
4.3.1	双端口 RAM 介绍	84
4.3.2	同步 FIFO 设计	85
4.4	异步 FIFO 设计	88
4.4.1	空/满标志的产生	88
4.4.2	亚稳态问题	88
4.4.3	格雷码	89
4.4.4	异步 FIFO 设计及结果	90
第 5 章	串口通信模块设计	95
5.1	串口通信概述	95
5.1.1	串口介绍	95
5.1.2	串口帧格式	96
5.2	设计指标	97
5.3	串口通信模块方案	97
5.3.1	原理框图	97
5.3.2	FPGA 方案	98
5.4	串口通信各个模块设计	99
5.4.1	MCU 配置模块设计	99
5.4.2	发送模块设计	106
5.4.3	接收模块设计	114
5.4.4	顶层模块	123
5.5	串口通信模块调试注意事项	126
第 6 章	数/模及模/数转换控制模块设计	127
6.1	相关技术介绍	127
6.1.1	数/模转换器 (DAC) 介绍	127

6.1.2	模/数转换器 (ADC) 介绍	129
6.1.3	数/模转换芯片 AD5641 介绍	131
6.1.4	模/数转换芯片 ADS7230 介绍	132
6.2	技术指标要求	135
6.3	D/A 模块设计	136
6.4	A/D 模块设计	141
6.5	遇到的问题、排查及定位	148
第 7 章	HDLC 总线控制器设计	149
7.1	HDLC 协议介绍	149
7.2	CRC 校验	150
7.3	设计技术指标	150
7.4	HDLC 电路图	150
7.5	HDLC 总线控制器模块划分	151
7.6	HDLC 总线控制器各个模块设计	152
7.6.1	CRC 模块设计	152
7.6.2	发送编码设计及结果	152
7.6.3	接收解码设计及结果	163
7.7	遇到的问题及解决办法	174
第 8 章	SDRAM 控制器设计	175
8.1	SDRAM 简介	175
8.2	SDRAM 芯片介绍	176
8.3	SDRAM 设计方案	178
8.3.1	刷新周期问题	180
8.3.2	初始化模块	183
8.3.3	定时刷新请求信号产生模块	184
8.3.4	读/写请求产生模块	185
8.3.5	读/写状态机	187
第 9 章	NAND FLASH 控制器设计	195
9.1	NAND FLASH 介绍	195
9.1.1	NAND FLASH 信号功能介绍	195
9.1.2	芯片存储阵列结构和地址寻址	196
9.1.3	NAND FLASH 时序图介绍	197
9.2	设计技术指标	201
9.3	原理图介绍	202
9.4	NAND FLASH 控制器方案	202

9.5	NAND FLASH 各个模块设计	203
9.5.1	MCU 接口模块	203
9.5.2	FLASH 管理模块	205
9.5.3	NAND FLASH 坏块表查找模块设计	209
9.5.4	NAND FLASH 页查找模块设计	210
9.5.5	NAND FLASH 读模块设计	211
9.5.6	NAND FLASH 写模块设计	212
9.5.7	NAND FLASH 块擦除模块设计	214
9.6	遇到的问题及解决办法	217
9.6.1	对 FLASH 的读/写不能完成	217
9.6.2	读/写不稳定	217
第 10 章	1553B 总线控制器设计	218
10.1	1553B 总线介绍	218
10.1.1	总线拓扑结构	219
10.1.2	编码方式	220
10.1.3	1553B 数据格式	220
10.1.4	总线数据传输过程说明	223
10.2	设计技术指标	224
10.3	原理图介绍	224
10.4	收发器芯片介绍	224
10.5	1553B 总线控制器方案	225
10.6	1553B 总线控制器各个模块设计	226
10.6.1	曼彻斯特编码模块设计	226
10.6.2	曼彻斯特解码模块设计	230
10.6.3	协议处理模块设计	231
10.6.4	RT 接收模块设计	238
10.6.5	RT 发送模块设计	242
10.6.6	RT-RT 模块设计	243
10.6.7	Broadcast 模块设计	243
10.7	测试结果	245
10.8	遇到的问题及解决办法	246
参考文献		247
后记		248

第1章

FPGA 介绍

➔ 1.1 FPGA 的优势

- 运行速度快

FPGA 内部集成锁相环，可以实现外部时钟倍频，核心频率可达几百兆，而单片机运行速度则低得多。在高速场合，单片机无法代替 FPGA。FPGA 运算速度快，编程简易，而且有些高端的 FPGA 芯片内部集成了很多有用的模块，如串行收发模块，如果不用 FPGA 而是另搭电路，结构将会比较复杂，用 FPGA 可以简化设计。

- FPGA 引脚多，容易实现大规模系统

单片机 IO 口有限，而 FPGA 动辄数百个 IO 引脚，甚至上千个引脚，可以方便地连接外设，比如一个系统有多路 AD、DA，单片机要仔细地进行资源分配与总线隔离，而 FPGA 由于有着丰富的 IO 资源，可以用不同 IO 引脚很容易地连接各外设。

- FPGA 内部程序并行执行性

单片机程序是串行执行的，执行完一条才能执行下一条，在处理突发事件时只能调用有限的中断资源；而 FPGA 的不同逻辑可以并行执行，可以同时处理不同任务，这使得 FPGA 的工作效率更高。

利用硬件并行执行的优势，FPGA 打破了顺序执行的模式，在每个时钟周期内完成更多的处理任务，超越了数字信号处理器（DSP）的运算能力。

- FPGA 包含大量软核，可以方便地进行二次开发

FPGA 甚至包含单片机和 DSP 软核，并且其 IO 数仅受 FPGA 自身 IO 限制，所以 FPGA 又是单片机和 DSP 的超集，也就是说，单片机和 DSP 能实现的功能，FPGA 一般都能实现。

- FPGA 设计灵活

FPGA 最大的特点就是灵活，它能够实现你想实现的任何数字电路，可以定制各种电路。减少受制于专用芯片的束缚，真正为自己的产品量身定做，在设计的过程中可以灵活地更改设计。而且 FPGA 强大的逻辑资源和寄存器资源可以让用户轻松地发挥设计理念，其并行执行、硬件实现的方式可以满足设计中大量的高速电子线路设计需求。FPGA 拥有比 DSP 更快的速度，可以实现非常复杂的高速逻辑，有着比 ASIC（专用芯片）更短的设计周期和灵活性，免去昂贵的开版费用，而且可以随时裁减，增加用户想

要的功能,达到规避设计风险、回避芯片厂商的限制的目的。另外,知识产权的概念不断涌现,仿制别人抄袭,FPGA能够防止别人抄袭,让设计的智慧得到充分保护,并让公司的利益在较长时间内得到保证。随着FPGA芯片供应商和第三方公司的重视,现在有现成的IP核,这进一步缩短了设计周期,减小了开发成本。现在很多免费的软IP核和硬IP核的出现更是压缩了成本。

➔ 1.2 FPGA典型结构图

基于LUT的FPGA具有很高的集成度,其器件密度从数万门到数千万门不等,可以完成极其复杂的时序与逻辑组合的逻辑电路功能,因此其适用于高速、高密度的高端数字逻辑电路设计领域。其组成部分主要有可编程输入/输出单元、基本可编程逻辑单元、内嵌SRAM、丰富的布线资源、底层嵌入功能单元、内嵌专用单元等。FPGA的主要设计和生产厂家有赛灵思、Altera、Lattice、Actel、Atmel和QuickLogic等,其中最大的是美国赛灵思公司,在可编程市场中占有50%以上的份额,比其他所有竞争对手市场份额的总和还多。

FPGA是由存放在片内RAM中的程序来设置其工作状态的,因此,工作时需要对片内的RAM进行编程。用户可以根据不同的配置模式,采用不同的编程方式。

目前,FPGA市场占有率最高的两大公司——赛灵思公司和Altera生产的FPGA都是基于SRAM工艺的,需要在使用时外接一个片外存储器以保存程序。上电时,FPGA将外部存储器中的数据读入片内RAM,完成配置后,进入工作状态;掉电后SRAM存储的数据丢失,内部逻辑消失。这样,FPGA能反复使用,且无须专门的FPGA编程器,只需配备一片存储代码的FLASH存储器即可。

Actel、QuickLogic等公司还提供反熔丝技术的FPGA,它具有抗辐射、耐高低温、低功耗和速度快等优点,在军品和航空航天领域中应用较多,但这种FPGA不能重复擦写,开发初期比较麻烦,费用也比较昂贵。Lattice是ISP技术的发明者,在小规模PLD应用上有一定的特色。赛灵思公司的早期产品一般不涉及军品和宇航级市场,但目前已经有多款产品进入该领域。

目前主流的FPGA仍是基于查找表技术的,已经远远超出了先前版本的基本性能,并且整合了常用功能(如RAM、时钟管理和DSP)的硬核(ASIC型)模块。

此处,我们以XC2V3000为例,介绍典型的FPGA结构图。XC2V3000拥有300万门的超大系统容量,采用0.15 μm /0.12 μm CMOS 8层金属混合工艺设计,内核电压为1.5 V,根据输入/输出参考电压的不同设计可支持多种接口标准,内部时钟频率可达420 MHz。XC2V3000的内部结构如图1-1所示,它主要由可配置逻辑(CLB)、用户可编程I/O(IOB)、Block SelectRAM、数字时钟管理模块(DCM)、数字阻抗匹配模块(DCI)和硬件乘法器等组成。其中CLB用于实现FPGA的绝大部分逻辑功能;IOB用于提供封装引脚与内部逻辑之间的接口;BlockRAM用于实现FPGA内部的随机存取,它可配

置 RAM、双口 RAM、FIFO 等随机存储器；DCM 用于提供灵活的时钟管理功能；硬件乘法器用于提高 FPGA 的数字信号处理能力。

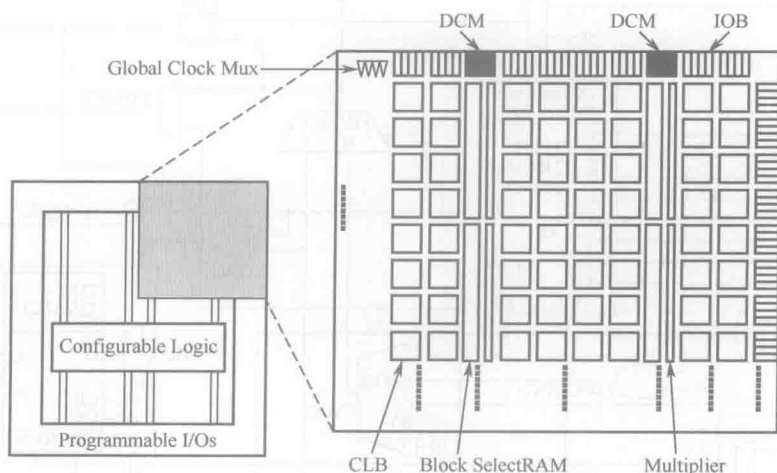


图 1-1 Virtex 系列产品 XC2V3000 的内部结构

1. 可配置逻辑块 (CLB)

如图 1-2 所示, XC2V3000 的 CLB 模块由 4 个相同的 Slice 和附加逻辑构成, 用于实现组合逻辑和时序逻辑。

每个 Slice 由 2 个 4 输入函数发生器、进位逻辑、算术逻辑、存储逻辑和函数复用器组成。算术逻辑包括 1 个异或门 (XORG) 与 1 个专用与门 (MULTAND), 1 个异或门可以使 1 个 Slice 实现 2 位全加操作, 专用与门用于提高乘法器的效率。进位逻辑由专用进位信号和函数复用器 (MUXC) 组成, 共同实现快速的算术加减法操作。具体结构详见图 1-3 和图 1-4。

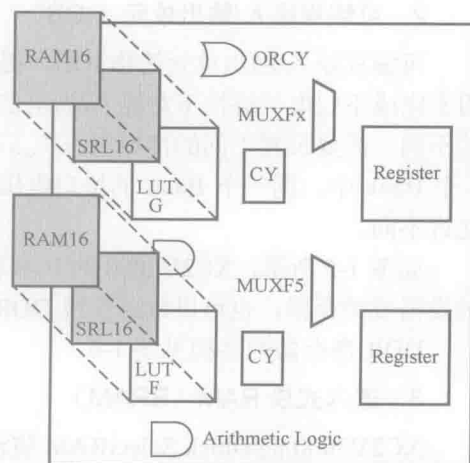
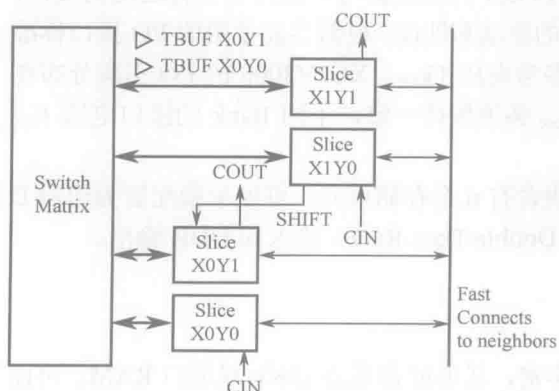


图 1-2 Virtex 系列产品 XC2V3000 的 CLB 结构 图 1-3 Virtex 系列产品 XC2V3000 的 SLICE 结构

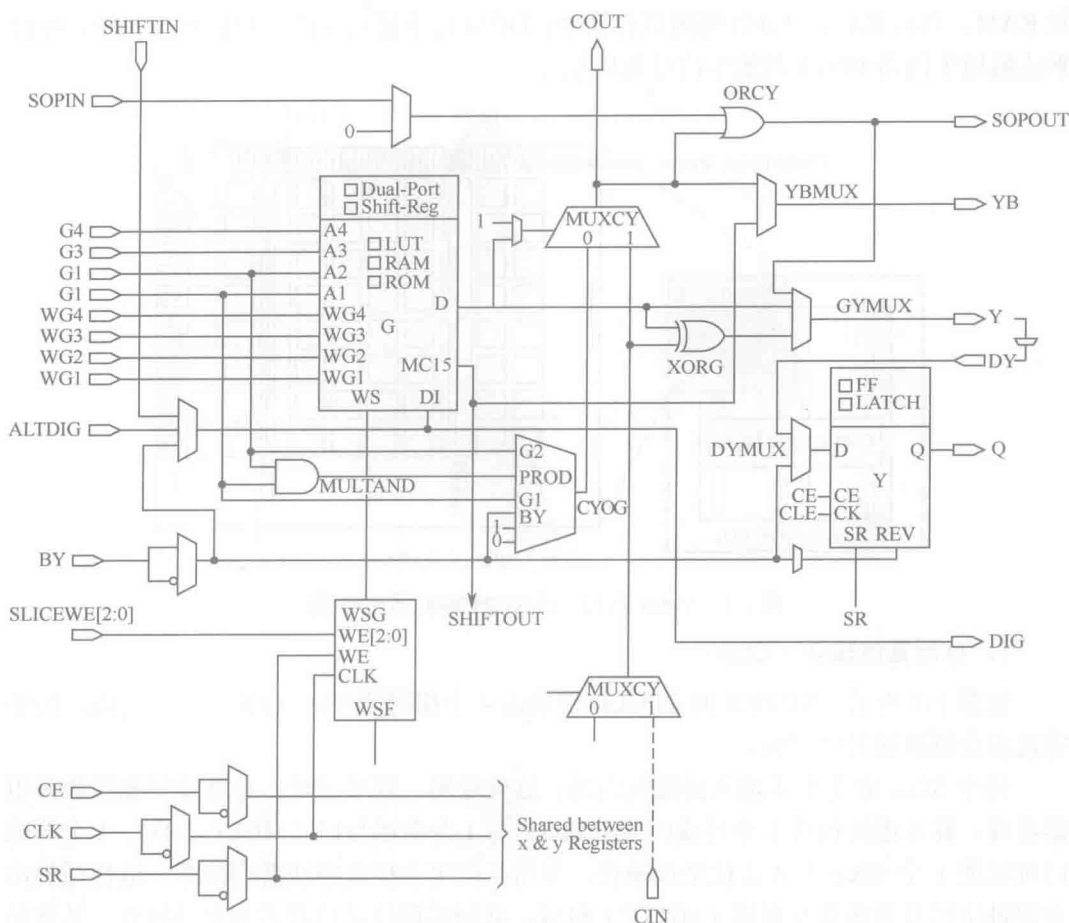


图 1-4 Virtex 系列产品 Slice 结构

2. 可编程输入/输出单元 (IOB)

可编程输入/输出单元简称 IOB，提供 FPGA 内部逻辑与外部封装引脚之间的接口，用于完成不同电气特性下对输入/输出信号的驱动和匹配。根据当前使用的 I/O 接口标准的不同，需要设置不同的接口电压 V_{cco} 和参考电压 V_{REF} 。XC2V3000 的 I/O 引脚分布在 8 个 Bank 中，同一个 Bank 的接口电压 V_{cco} 必须保持一致，不同 Bank 的接口电压 V_{cco} 允许不同。

如图 1-5 所示，XC2V3000 的 IOB 模块含有 6 个存储单元，可以单独配置为边沿 D 触发器或锁存器，也可以成对实现 DDR (Double-Data-Rate) 输入和 DDR 输出。

DDR 寄存器的结构见图 1-6。

3. 嵌入式块 RAM (BRAM)

XC2V3000 的 Block SelectRAM 资源丰富，其单位容量为 18Kb 双端口 RAM，可以自由设定数据宽度和深度纵横比，并支持三种并发读/写 (Read-During-Write) 模式。

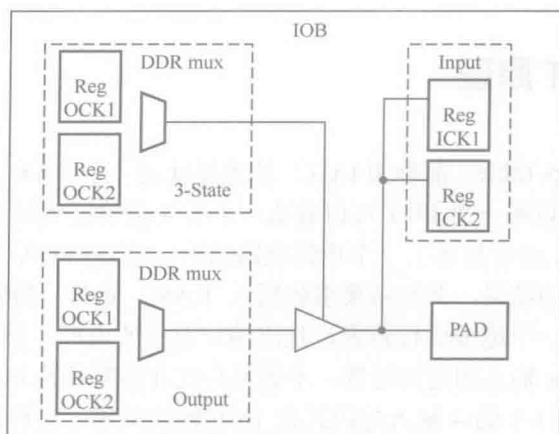


图 1-5 Virtex 系列产品 XC2V3000 的 IOB 结构

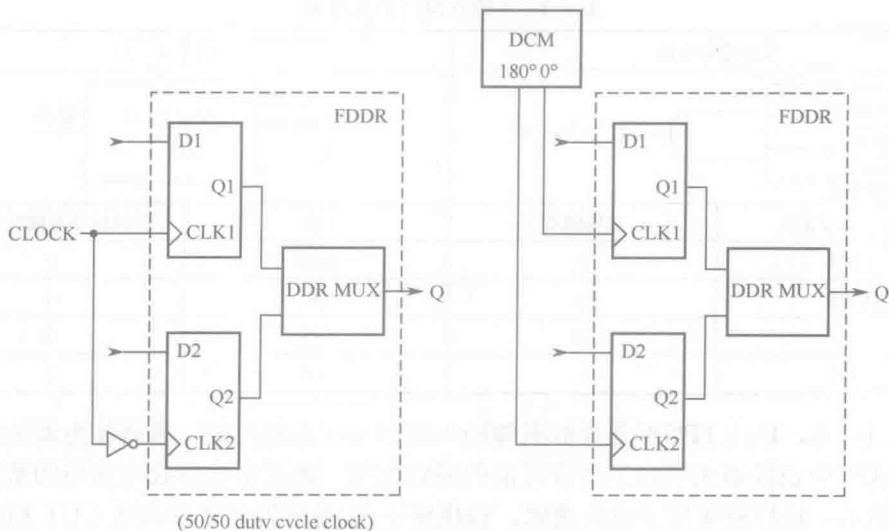


图 1-6 DDR 寄存器结构

4. 丰富的布线资源

布线资源连通 FPGA 内部所有的单元, IOB、CLB、BRAM、DCM 等都使用相同的内连阵列。XC2V3000 内部共有 16 条全局时钟线, 用于芯片内部全局时钟和全局复位/置位的布线。24 条纵横交错的长线资源实现了芯片 Bank 间的高速通信和第二全局时钟信号的布线。

5. 底层内嵌功能单元

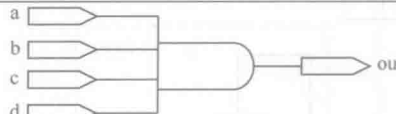
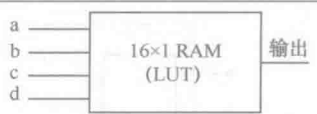
XC2V3000 集成了丰富的内嵌功能模块, 如 DCM (数字时钟管理器)、DCI (数字控制阻抗)、MicroBlaze 软处理器核等, 使其具有软、硬件联合设计的能力, 可以用有限的资源完成系统级的设计任务。

1.3 LUT 原理

查找表 (Look-Up-Table) 简称为 LUT，其本质就是一个 RAM。目前 FPGA 中多使用 4 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。当用户通过原理图或 HDL 语言描述了一个逻辑电路以后，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。这样，每输入一个信号进行逻辑运算就相当于输入一个地址进行查表，找出地址对应的内容，然后将其输出即可。

我们知道，一个 n 输入的逻辑运算，不管是与或非运算还是异或运算，最多只可能存在 2^n 种结果，如表 1-1 的 4 输入与门共有 16 种输出结果。这样就将实际逻辑电路转换成了 LUT 结构。

表 1-1 4 输入与门的真值表

实际逻辑电路		LUT 的实现方式	
			
a、b、c、d 输入	逻辑输出	地址	RAM 中存储的内容
0000	0	0000	0
0001	0	0001	0
.....	0		0
1111	1	1111	1

A, B, C, D 由 FPGA 芯片的引脚输入后进入可编程连线，然后作为地址线连到 LUT。LUT 中已经事先写入了所有可能的逻辑结果，通过地址查找到相应的数据，然后将其输出，这样就实现了组合逻辑。该电路中 D 触发器是直接利用 LUT 后面的 D 触发器来实现的。时钟信号 CLK 由 I/O 脚输入后进入芯片内部的时钟专用通道，直接连接到触发器的时钟端。触发器的输出端与 I/O 脚相连，把结果输出到芯片引脚，这样 PLD 就完成了图 1-7 所示电路的功能（以上步骤都是由软件自动完成的，不需要人为干预）。

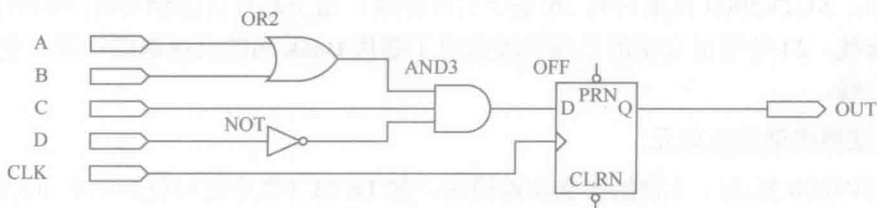


图 1-7 4 输入与门的逻辑实现

这个电路是一个很简单的例子，只需要一个 LUT 加一个触发器就可以完成。对于