



XILINX大学合作计划指定教材

基于XILINX FPGA 片上嵌入式系统的用户IP开发

■ 叶肇晋 张稀楠 马 磊 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

内 容 简 介

基于 XILINX FPGA 片上 嵌入式系统的用户 IP 开发

叶肇晋 张稀楠 马 磊 编著

西安电子科技大学出版社

2008

ISBN 978-7-5506-2132-7/TP · 1090

XDP 2427001-1

第 1 版 (2008 年 6 月)

第 1 次印刷

201 千字

787 毫米 × 1092 毫米 1/16 开 8.625

2008 年 6 月第 1 次印刷

印刷单位: 西安电子科技大学印刷厂

本书经国家版权局登记, 著作权登记号: 01-2008-01234567

http://www.xjtu.edu.cn E-mail: xjtu@xjtu.edu.cn

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

西安电子科技大学图书馆

内 容 简 介

本书基于 XILINX 的嵌入式开发平台,讲解了嵌入式系统的基本概念、FPGA 和 MicroBlaze 处理器以及最新的多端口内存控制器(MPMC)的原理。通过不同的总线和接口实验,详细讲述了怎样开发用户自定义 IP。

本书内容翔实,实践性强,结合具体的实验详细讲解了开发工程的过程,以帮助用户较快熟悉用户 IP 的开发。

本书适用于具有一定 XILINX FPGA 开发应用经验,熟悉 DHL 语言,致力于基于 FPGA 片上系统开发的工程师。

图书在版编目(CIP)数据

基于 XILINX FPGA 片上嵌入式系统的用户 IP 开发 / 叶肇晋, 张稀楠, 马磊编著.

—西安: 西安电子科技大学出版社, 2008.12

XILINX 大学合作计划指定教材

ISBN 978-7-5606-2135-7

I. 基… II. ①叶… ②张… ③马… III. 微型计算机—系统设计—高等学校—教材 IV. TP360.21

中国版本图书馆 CIP 数据核字(2008)第 147165 号

策 划 戚文艳

责任编辑 张 玮 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

http://www.xduph.com E-mail: 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 2008 年 12 月第 1 版 2008 年 12 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 8.625

字 数 201 千字

印 数 1~4000 册

定 价 19.00 元(含光盘)

ISBN 978-7-5606-2135-7/TP·1090

XDUP 2427001-1

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。

目 录

前 言

随着 FPGA 技术的发展, 基于 FPGA 片上系统的开发已成为目前 FPGA 应用的一个热点。

本书针对基于 FPGA 片上系统开发的核心、用户 IP 的开发, 结合 XILINX 的嵌入式开发工具 EDK, 详细讲解了如何开发和调试客户自己的用户硬件外设(用户 IP), 使得开发者可以很快地熟练使用 EDK 以及进行片上系统开发。

书中内容主要针对嵌入式用户硬件外设的开发流程和调试方法, 不涉及开发语言的细节。在使用本书前必须熟练掌握硬件描述语言。

本书共 7 章。前三章以基本概念介绍为主, 分别介绍了基于 XILINX FPGA 嵌入式系统片上系统开发的基本概念, MicroBlaze 的架构、接口及使用, MPMC 的架构、接口及使用。后四章以实验为主, 分别介绍了在 XILINX 嵌入式开发平台常用接口上用户 IP 开发的实现: 第 4 章是 EDK 工具的使用流程; 第 5 章是基于 PLB 总线接口的用户 IP 的开发; 第 6 章是基于 FSL 总线接口的用户 IP 的开发; 第 7 章是基于 MPMC/NPI 接口的用户 IP 的开发。

基于 FPGA 的片上嵌入式开发实践性较强, 本书结合具体的实验详细讲解了开发过程。附带光盘中提供了所有例程的源代码和工程文件, 供读者参考。

在本书的编纂过程中, 复旦大学电气工程专业大四学生张稀楠完成了第 4~7 章的实验部分, 并完成了上述各章主要内容的初稿; 大四学生马磊完成了第 2、3 章内容和第 4~6 章中基本概念的介绍和相关结构分析的初稿, 并对第 1 章进行了修改。他们都是本书的作者。

另外, 上海交通大学微电子学院研二学生许昀、祝翔宇和大四学生阙志强共同协助完成了本书的审核与整理工作, 在此表示由衷的感谢。

鉴于笔者经验有限, 在编写和校核过程中难免存在纰漏, 望读者多多指正, 及时提出宝贵意见。

最后, 衷心地希望本书能给从事嵌入式系统开发或学习的读者以帮助。

叶肇晋

2008 年 6 月

第 4 章 嵌入式开发套件(EDK)概述及使用流程

4.1 用 BSB 建立一个 EDK 设计

4.1.1 在 XPS 开发环境下创建工程

4.1.2 使用 Platform Studio 调整设计

4.2 使用 XMD 对 EDK 设计进行调试

4.3 加入一个用户 IP

目 录

第 1 章 基于 XILINX FPGA 嵌入式系统片上系统开发概述.....	1
1.1 XILINX 简介.....	1
1.2 基于 XILINX FPGA 片上嵌入式系统简介.....	2
1.3 XILINX FPGA 片内资源简介.....	3
1.3.1 CLB.....	4
1.3.2 BRAM.....	5
思考题.....	7
第 2 章 MicroBlaze 的构架及接口.....	8
2.1 MicroBlaze 处理器.....	8
2.2 MicroBlaze 的结构、机制和特性.....	9
2.2.1 MicroBlaze 系统及内部结构.....	9
2.2.2 MicroBlaze 的数据存储结构和指令.....	10
2.2.3 MicroBlaze 的流水线结构.....	13
2.2.4 MicroBlaze 的中断机制.....	14
2.2.5 MicroBlaze 的缓存机制和 MMU.....	16
2.2.6 MicroBlaze 的高级特性介绍.....	17
2.3 MicroBlaze 的总线接口.....	17
思考题.....	19
第 3 章 MPMC 的构架、接口及使用.....	20
3.1 MPMC 简介.....	20
3.1.1 软件直接内存存储支持(SDMA).....	20
3.1.2 MPMC 的内部结构.....	20
3.1.3 MPMC 接口.....	21
3.2 MPMC 的使用.....	22
思考题.....	30
第 4 章 嵌入式开发套件(EDK)概述及使用流程.....	31
4.1 用 BSB 建立一个 EDK 设计.....	31
4.1.1 在 XPS 开发环境下创建工程.....	31
4.1.2 使用 Platform Studio 调整设计.....	39
4.2 使用 XMD 对 EDK 设计进行调试.....	44
4.3 加入一个用户 IP.....	49

4.3.1	生成一个用户 IP	49
4.3.2	用户 IP 样本目录	51
4.3.3	修改用户 IP 实现功能	52
4.4	用 Chipscope 对用户 IP 进行调试	55
	思考题	58
第 5 章	基于 PLB 总线接口的用户 IP 的开发	59
5.1	PLB 总线和 IPIF 简介	59
5.1.1	PLB 简介	59
5.1.2	IPIF 简介	59
5.2	用向导建立一个用户 IP	61
5.3	在用户 IP 中添加功能代码实现 PWM 功能	64
5.4	硬件上的验证	74
	思考题	80
第 6 章	基于 FSL 总线接口的用户 IP 的开发	81
6.1	FSL 总线简介	81
6.2	用向导建立一个 FSL 设计的模版	82
6.3	在用户 IP 中添加功能代码实现 UART 功能	86
6.4	硬件验证	100
6.4.1	通过超级终端验证	100
6.4.2	通过 Chipscope 验证	101
	思考题	103
第 7 章	基于 MPMC NPI 接口的用户 IP 的开发	104
7.1	NPI 简介	104
7.2	NPI 接口开发例程一	112
7.2.1	用户 IP 设计	112
7.2.2	硬件验证	122
7.3	NPI 接口开发例程二	125
7.3.1	用户 IP 设计	125
7.3.2	在 EDK 中打开并编辑用户 IP	129
	思考题	132

第1章 基于 XILINX FPGA 嵌入式系统 片上系统开发概述

1.1 XILINX 简介

XILINX 是全球领先的可编程逻辑完整解决方案的供应商。XILINX 公司成立于 1984 年,首创了现场可编程逻辑阵列(FPGA)这一创新性的技术,并于 1985 年首次推出了商业化产品。目前 XILINX 满足了全世界对 FPGA 产品一半以上的需求。XILINX 可编程逻辑解决方案缩短了电子设备制造商开发产品的周期并加快了产品面市的速度,从而减小了制造商的风险。XILINX 产品已经被广泛应用于从无线电话基站到 DVD 播放机的数字多媒体电子技术中。传统的半导体公司只有几百个客户,而 XILINX 在全世界有 7500 多家客户及 50 000 多个设计。

XILINX 的主流 FPGA 分为两大类,一类是侧重于低成本应用,容量中等,性能可以满足一般的逻辑设计要求的 Spartan 系列;还有一类是侧重于高性能应用、容量大、性能能满足各类高端应用的 Virtex 系列。用户可以根据自己的实际需求进行选择。在性能可以满足的情况下,优先选择低成本器件。目前高端的 Virtex 系列已经从开始的 Virtex 发展到最近的 Virtex 5,如图 1.1.1 所示。

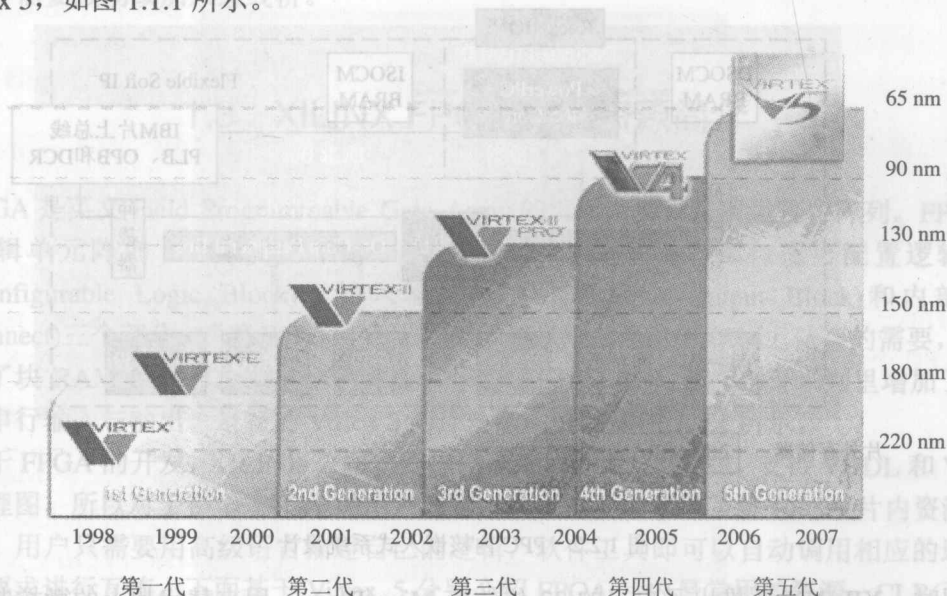


图 1.1.1 Virtex 的发展史

1.2 基于 XILINX FPGA 片上嵌入式系统简介

随着 FPGA 技术的发展, FPGA 的逻辑容量越来越大, 成本越来越低。在这种趋势下, FPGA 可以代替系统中越来越多的器件, 最后发展到 FPGA 片上系统。图 1.2.1 是 FPGA 由分立的器件在功能上渐渐合并成片上系统。

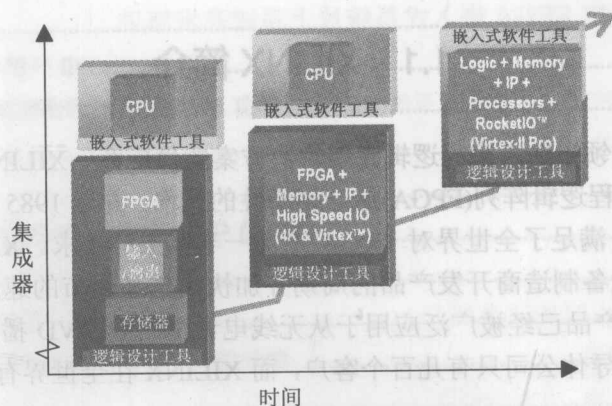


图 1.2.1 系统设计的集成

为了迎合这种趋势, 从 Virtex-II Pro 开始, XILINX 在其高端的部分器件中增加了多个硬核 PowerPC(PPC)内核, 以推动这种趋势。图 1.2.2 是 PPC 内核嵌入式设计的结构组成。

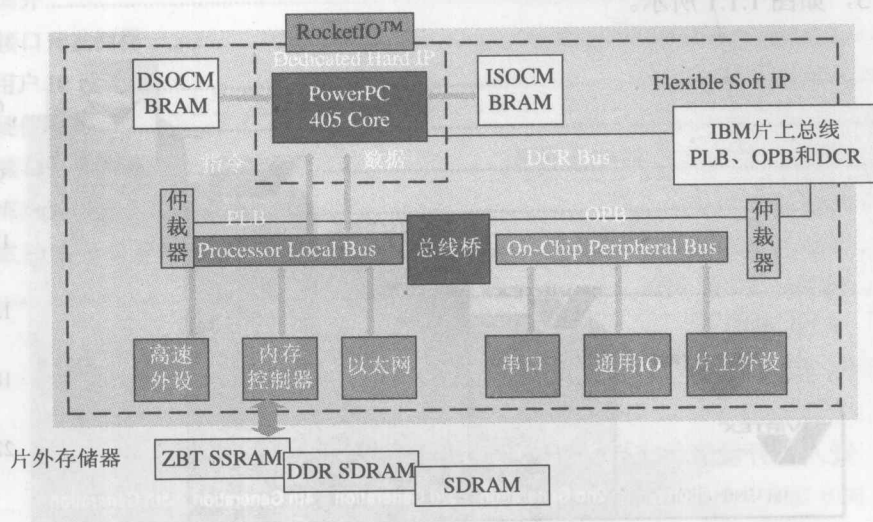


图 1.2.2 PPC 内核嵌入式系统设计

同时, XILINX 还推出了自己的 32 位软核 MicroBlaze, 用来替代片上的微控制器, 帮助用户缩小 PCB 板的面积和器件数量, 并降低整个系统的成本。最初 MicroBlaze 是基于

OPB(On-chip Peripheral Bus)总线(如图 1.2.3 所示)的,在 MicroBlaze 发展到 7.0(对应的 EDK 版本为 9.2)的时候也采用了和 PPC 一样的 PLB 总线。同时 XILINX 还在 EDK 上整合了 MPMC(多端口内存控制)。用户的数据可以直接从外设写入内存,也可以直接从内存读出,大大提高了系统的效率。这样,大量的数据搬运可以不通过 PLB 总线而直接完成。PLB 的带宽可以完全由 MicroBlaze 使用。MPMC 的构架和使用将在第 3 章中详细讲述。

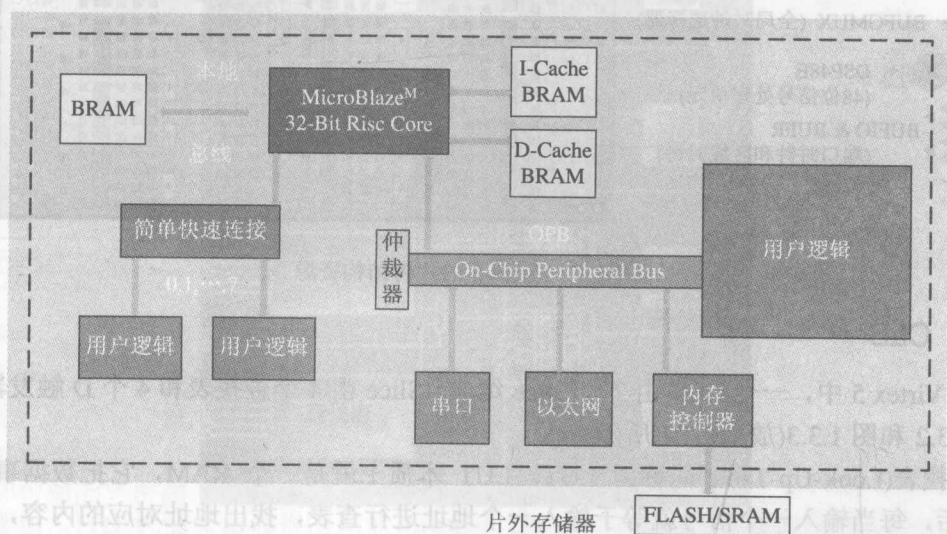


图 1.2.3 基于 OPB 的 MicroBlaze

考虑到带有 PPC 的芯片和开发板价格较高,所以本书将以 MicroBlaze 7.0 和 EDK 9.2 为基础来讲述怎样开发用户 IP。由于 MicroBlaze 和 PPC 目前都使用同样的总线,因此并不需要改动或加入总线桥。

1.3 XILINX FPGA 片内资源简介

FPGA 是英文 Field Programmable Gate Array 的缩写,即现场可编程门阵列。FPGA 采用了逻辑单元阵列 LCA(Logic Cell Array)这样一个概念,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)和内部连线(Interconnect)三个部分。随着 FPGA 的发展,为了满足对信号处理和片上储存的需要,FPGA 又增加了块 RAM(BRAM)、时钟管理单元和硬件乘法器,之后又在部分系列里增加了 PPC 和高速串行输入与输出。最新的 Virtex 5 的片内结构资源如图 1.3.1 所示。

对于 FPGA 的开发,XILINX 公司已经提供了集成开发环境 ISE,支持 VHDL 和 Verilog 及电原理图。所以对于初学 FPGA 的用户而言,并不需要细致地了解 FPGA 片内资源的所有细节。用户只需要用高级语言描述自己的逻辑,软件工具即可以自动调用相应的逻辑资源并按要求进行互连。下面基于 Virtex 5 分别介绍 FPGA 片内最常用的资源:CLB(可编程逻辑块)和 BRAM(块 RAM)。

- CLB (可编程逻辑块)
- BRAM (块RAM)
- I/O (输入/输出)
- CMT (时钟管理单元)
- ▶ BUFGMUX (全局时钟选择器)
- ⊗ ⊕ DSP48E (48位信号处理单元)
- ▶ BUFG & BUFR (端口时钟和区域时钟)

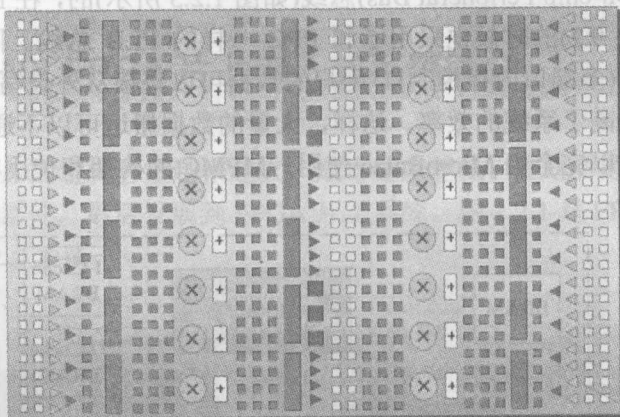


图 1.3.1 Virtex 5 的片内结构资源

1.3.1 CLB

在 Virtex 5 中, 一个 CLB 由 2 个 Slice 组成, Slice 由 4 个查找表和 4 个 D 触发器组成, 如图 1.3.2 和图 1.3.3(放大后的)所示。

查找表(Look-Up-Table)简称为 LUT。LUT 本质上就是一个 RAM, 它把数据事先写入 RAM 后, 每当输入一个信号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。对于任意逻辑, 只要将其真值表写入查找表, 就能实现其逻辑功能。Virtex 5 的查找表是 6 输入的, 所以对于任意的 6 输入、1 输出逻辑, 只要一个查找表就可以实现。这是 FPGA 开发和 ASIC 开发的一个重要区别。在 FPGA 的代码中增加逻辑的复杂度并不会增加资源和降低速度, 而主要取决于逻辑输入和输出的个数。

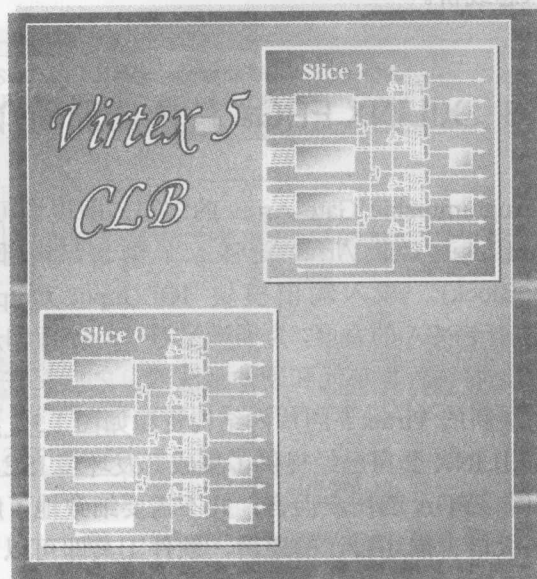


图 1.3.2 Virtex 5 的常用资源 CLB(可编程逻辑块)中的 Slice 结构

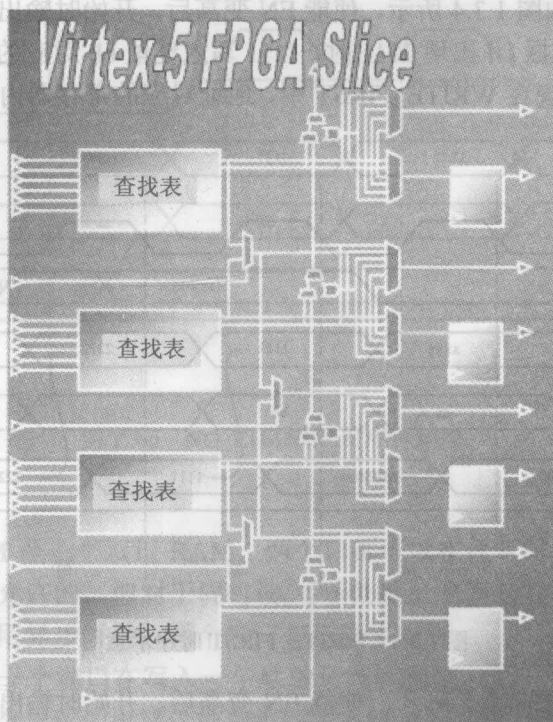


图 1.3.3 放大后的 Slice 结构

在 Virtex 5 部分 Slice 中的查找表不仅可以作为逻辑，还可以用作分布式 RAM 和 32 位移位寄存器，我们把这样的 Slice 叫做 SliceM；把仅可以作为逻辑的 Slice 叫做 SliceL。

在查找表和触发器的中间位置是 2 选 1 的选择器和快速进位链，它主要用于宽输入的逻辑和在加法中构成进位链。

1.3.2 BRAM

在 Virtex 5 中，BRAM 的大小是 36 Kb。每个 BRAM 都是双端口的。两个端口可以同时读/写 BRAM 里面的数据进行读/写，而且两个端口可以工作在不同的时钟频率下。当两个端口同时读/写同一地址时，可能会发生冲突，但是不会损坏器件，在本节中会具体说明这种情况。此外，在 Virtex 5 中，BRAM 还可以用做 FIFO。

Virtex 4/Spartan 3 中每个 BRAM 也是双口的，不过和 Virtex 5 相比，大小是其一半(18 Kb)。Virtex 5 的每个 BRAM 可以在绝大多数情况下等价于 Virtex 4/Spartan 3 中的两个 BRAM。

1. 写入模式

BRAM 有三种不同的写入模式，写入模式决定了一个新数据写入后在输出端口上数据的情况。这三种模式分别是 WRITE_FIRST、READ_FIRST 和 NO_CHANGE，默认的模式是 WRITE_FIRST。双口 BRAM 的两个口可以独立地设置写入模式。

当新数据写入时，WRITE_FIRST 输出端口上出现新的数据，READ_FIRST 输出端口上维持旧的数据，NO_CHANGE 输出端口上一直维持读操作时读取的数据。

WRITE_FIRST: 如图 1.3.4 所示, 使能 EN 变高后, 开始时输出数据 DO 输出的是地址 aa 对应的值, 在输入数据 DI 出现 1111 时, 1111 写入地址 bb 对应的寄存器, 同时输出数据 DO 也输出 1111。就是说在 WRITE_FIRST 中, 地址对应的寄存器内容和输出都会随着输入的改变而改变。

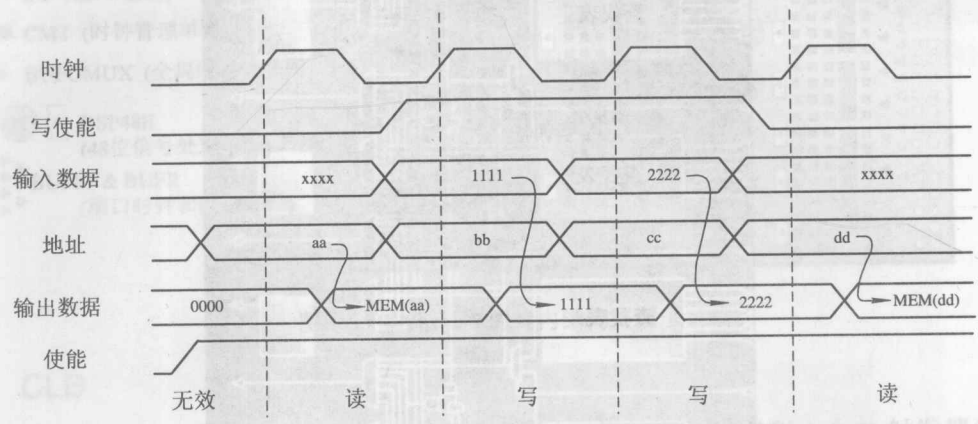


图 1.3.4 WRITE_FIRST 时序示意图

READ_FIRST: 如图 1.3.5 所示, 使能 EN 变高后, 开始时输出数据 DO 输出的是地址 aa 对应的值, 在输入数据 DI 出现 1111 时, 地址 bb 对应的寄存器值变为 1111, 但输出数据 DO 输出的仍然是地址 bb 的旧值。就是说在 READ_FIRST 中, 地址对应的寄存器内容会随着输入的改变而改变, 但输出不会改变, 直到下次输出该地址值时才输出存入的值。

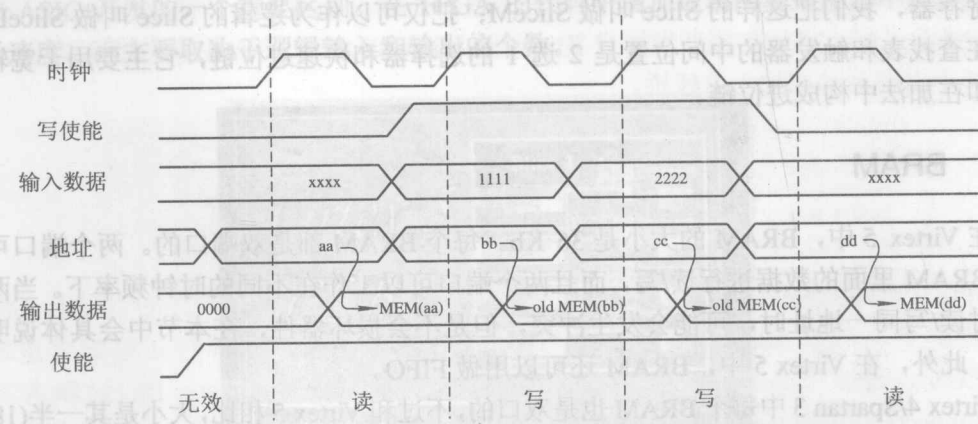


图 1.3.5 READ_FIRST 时序示意图

NO_CHANGE: 如图 1.3.6 所示, 使能 EN 变高后, 开始时输出数据 DO 输出的是地址 aa 对应的值, 在输入数据 DI 出现 1111 时, 地址 bb 对应的寄存器值变为 1111, 但输出数据 DO 输出值保持不变。就是说在 NO_CHANGE 中, 当写使能 WE 为高时, 地址对应的寄存器内容会随着输入的改变而改变, 但输出一直保持初始输出的值。

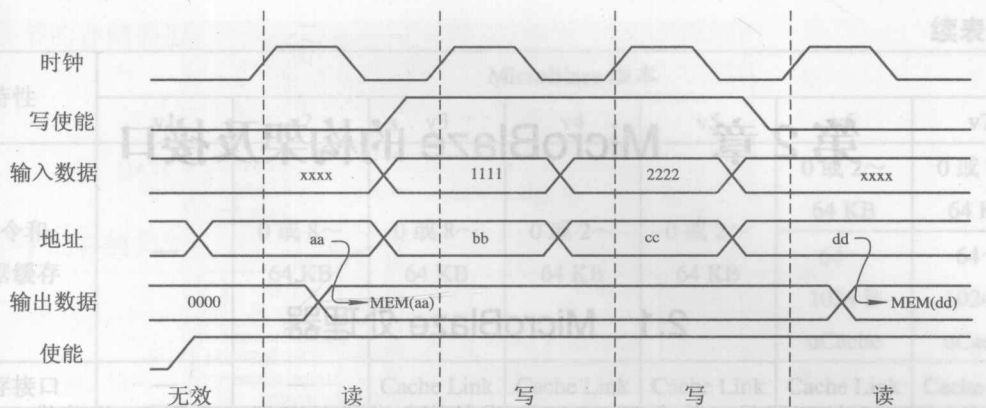


图 1.3.6 NO_CHANGE 时序示意图

2. 避免冲突

Virtex 5 的块 RAM 是一个双口 RAM，两个口都可以在任意时间访问任何地址。两个端口同时读一处空间是允许的，但对于同时读/写一处地址空间是有限制的。下面分两种情况来讨论：两个端口使用异步时钟；两个端口使用同步时钟。

对于异步时钟，当一个端口在写入一个地址空间的瞬间，另一个端口是不可以读取或写入同一空间的，否则会导致读出错误的数据或写入错误的数据。

对于同步时钟，当一个端口在写入一个地址空间的瞬间，另一个端口是不可以写入同一空间的，否则会导致读出错误的数据或写入错误的数据。如果这个端口配置成 READ_FIRST 或 WRITE_FIRST 模式，就可以同时读出有效的数据；配置成 NO_CHANGE 模式会得到无效的数据。

3. 硬件 FIFO 支持

许多 FPGA 设计需要使用 FIFO，在 Virtex 5 中的 BRAM 可用作 FIFO。Virtex 5 的 BRAM 中内建了 FIFO 的逻辑，所以当用 Virtex 5 的 BRAM 用作 FIFO 时不需要使用额外的 CLB 资源。Virtex 5 中的 BRAM 可以用作同步的 FIFO，也可以用作异步的 FIFO。同时，Virtex 5 的 BRAM 还可以配置成标准的 FIFO 或 FWFT(First-Word Fall-Through)的 FIFO。

思考题 怎样在 FPGA 中实现一个 16 位加法器？

第2章 MicroBlaze 的构架及接口

2.1 MicroBlaze 处理器

MicroBlaze 处理器是一款由 XILINX 开发的嵌入式软处理器,采用 32 位哈佛 RISC 构架。基本的 MicroBlaze 构架包含 32 个通用寄存器、1 个算术逻辑单元(ALU)、1 个移位器和 2 级中断。MicroBlaze 还具有可配置的特性,如桶形移位器、内存管理/内存保护单元、浮点单元(FPU)、高速缓存、异常处理和调试逻辑。用户可以有条件地增加高级特性来满足实际需要。

MicroBlaze 是一个高度灵活可以配置的软核。可以根据设计的需要对 MicroBlaze 进行裁减,用最少的资源完成设计。

MicroBlaze 的基本特性如下:

- (1) 32 个 32 位的通用寄存器。
- (2) 32 位指令, 3 个操作数和 2 种地址模式。
- (3) 32 位地址。
- (4) 单流水线。

除了这些固定的特性以外, MicroBlaze 还提供了一系列的可选特性以适应用户不同的设计需要。这些特性在不同版本的 MicroBlaze 中是不同的。目前最新版本的 MicroBlaze (v7.0) 可以支持所有这些可选的特性。表 2.1.1 就列出了 MicroBlaze 各版本的特性。

表 2.1.1 不同版本 MicroBlaze 的性能

特 性	MicroBlaze 版本						
	v1	v2	v3	v4	v5	v6	v7
流水线级数	3	3	3	3	5	3 & 5	3 & 5
最大 Integer Perf /DMIPS	82	125	125	166	240	240	240
本地内存/KB	0 或 8~64	0 或 8~64	0 或 8~64	0 或 2~128	0 或 2~256	0 或 2~256	0 或 2~256
桶形移位 寄存器	可选	可选	可选	可选	可选	可选	可选
硬件乘法器	—	可选	可选	可选	可选	可选	可选
协处理接口	—	FSL	FSL	FSL	FSL	FSL	FSL

续表

特性	MicroBlaze 版本						
	v1	v2	v3	v4	v5	v6	v7
指令和数据缓存	—	0 或 8~ 64 KB	0 或 8~ 64 KB	0 或 2~ 64 KB	0 或 2~ 64 KB	0 或 2~ 64 KB	0 或 2~ 64 KB
						64~ 1024 B uCache	64~ 1024 B uCache
缓存接口	—	—	Cache Link	Cache Link	Cache Link	Cache Link	Cache Link
浮点单元	—	—	—	单精度	单精度	单精度	单精度
				33 MFLOPS	50 MFLOPS	50 MFLOPS	50 MFLOPS
内存管理单元	—	—	—	—	—	—	可选(MPU 或 MMU)
							完整 Linux 支持
调试接口	ROM monitor	JTAG HW Debug	JTAG HW Debug	Debug + Trace	Debug + Trace	Debug + Trace	Debug + Trace
目标器件	Virtex II	Virtex II Pro	Virtex II Pro	Virtex 4	Virtex 5	Virtex 5 family	Virtex 5 family
	Spartan 2	Spartan 2E	Spartan 3	Spartan 3E		Spartan 3 family	Spartan 3 family

2.2 MicroBlaze 的结构、机制和特性

2.2.1 MicroBlaze 系统及内部结构

MicroBlaze 采用 32 位哈佛 RISC 架构，32 位独立的地址总线 and 数据总线。在采用缓存的模式下，独立的指令和数据可以缓存并通过 XCL(XILINX CacheLink)接口连接到内存。在不采用缓存的机制下，地址和数据总线直接通过 PLBv64 访问内存。MicroBlaze 没有区分数据接口为 I/O 和存储，所以处理器存储的接口是 PLB 和 LMB，且 LMB 存储地址空间不和 PLB 重合。MicroBlaze 还提供最大 256 K 的片上本地内存接口和片上 BRAM 连接，对于简单的设计，可以将代码直接放入片上 BRAM，也可以将复杂设计的 BOOTLOADER 放在片上的 BRAM 中。用户 IP 可以通过 PLBv64 总线或快速点对点连接(FSL)接入系统。一般而言，PLBv64 比较适合基于总线形式的用户 IP 开发；FSL 比较适合基于简单点对点输入/输出或协处理器的用户 IP 开发。图 2.2.1 是一个 MicroBlaze 片上系统的结构图，图 2.2.2 则是 MicroBlaze 的内部结构图，用户可以根据图来加深对 MicroBlaze 的了解。

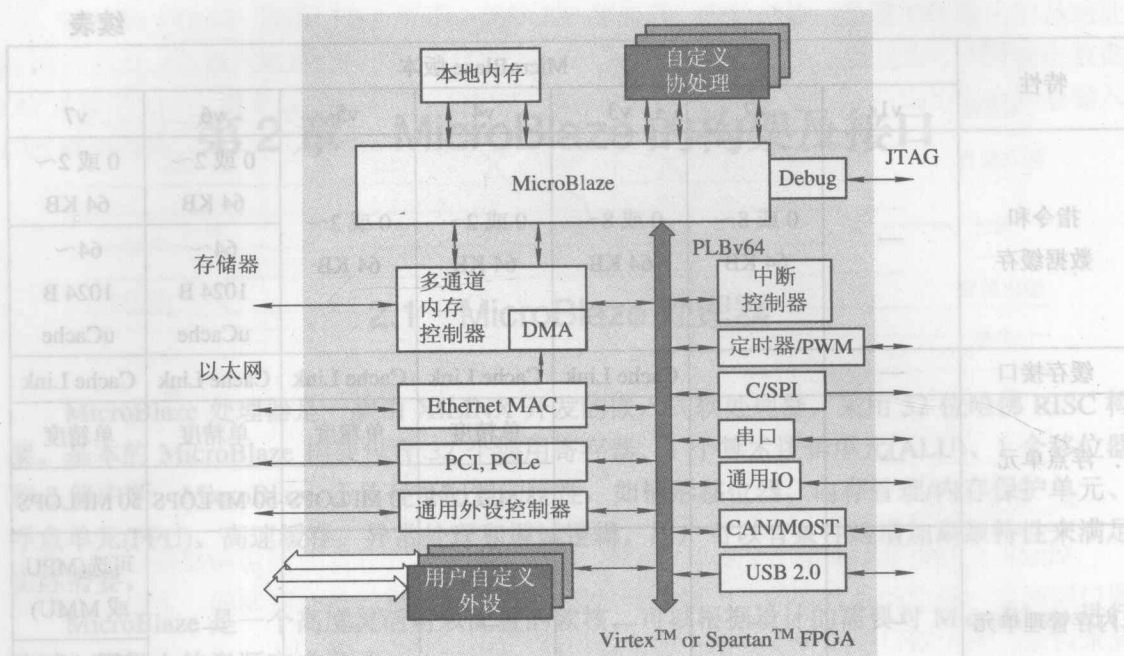


图 2.2.1 MicroBlaze 系统结构图

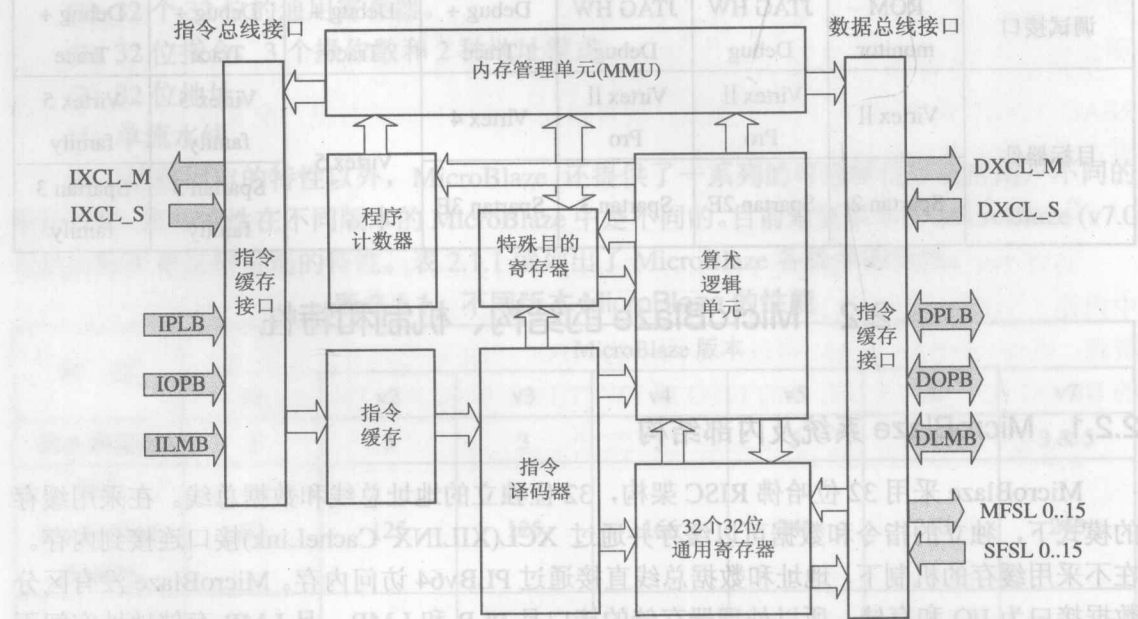


图 2.2.2 MicroBlaze 的内部结构图

2.2.2 MicroBlaze 的数据存储结构和指令

MicroBlaze 支持 8 位(字节)、16 位(半字)和 32 位(字)的数据形式，并采用大字序方式，其具体的形式如下所示：

字节的存储类型:

字节地址	n	
位编号	0	7
字序	高位	低位

半字的存储类型:

字节地址	n	n+1
字节编号	0	1
字节次序	高字节	低字节
位编号	0	15
字序	高位	低位

字的存储类型:

字节地址	n	n+1	n+2	n+3
字节编号	0	1	2	3
字节次序	高字节			低字节
位编号	0			31
字序	高位			低位

MicroBlaze 的所有指令都是 32 位的, 且有两种定义模式。A 型指令有最多 2 个寄存器源操作数和 1 个寄存器目的操作数; 而 B 型指令有 1 个寄存器源操作数、1 个 16 位立即数以及 1 个寄存器目的操作数。该立即数可以通过使用 Imm 指令将之扩展为 32 位。所有指令被分成 5 种指令类型: 算术指令、逻辑指令、程序流控制指令、读取/存储指令和特殊指令类型。表 2.2.1 和表 2.2.2 各自列出了 MicroBlaze 指令集的语法定义和参考实例。

表 2.2.1 MicroBlaze 指令集的语法定义

符号	解 释	符号	解 释
Ra	R0~R31, 通用寄存器, 源操作数 a	X[y]	寄存器 x 的第 y 位
Rb	R0~R31, 通用寄存器, 源操作数 b	X[y:z]	寄存器 x 的第 y~z 位
Rd	R0~R31, 通用寄存器, 目的操作数 d	X	寄存器 x 的状态取返
SPR[x]	特殊目的寄存器 x	Imm	16 位立即数
MSR	机械状态寄存器=SPR[1]	Immx	x 位立即数
ESR	特殊状态寄存器=SPR[5]	FSLx	3 位快速单向链路(FSL)端口指向端 x
EAR	特殊状态寄存器=SPR[3]	C	进位标记, MSR[29]
FSR	浮点单元状态寄存器=SPR[7]	Sa	特殊目的寄存器的源操作数
PVRx	处理器版本寄存器 x=SPR[8192+x]	Sd	特殊目的寄存器的目的操作数
BTR	分支目标寄存器=SPR[11]	S(x)	符号扩展自变量 x~32 位值
PC	执行下一阶段程序计数器=SPR[0]	*Addr	在指定地址 Addr 上的存储内容(根据数据大小调整)