

基于 FPGA 和多 DSP 的并行信号处理系统的实现

潘昉晟^{1,2}, 赵峰¹, 奚军², 骆意²

(1. 上海交通大学微电子学院, 上海 200030; 2. 华东计算技术研究所, 上海 200233)

摘要: 为了克服在高速实时信号处理领域中传统单 DSP 系统处理能力的瓶颈, 多 DSP 并行处理技术应运而生, 成为当前该领域研究的热点。该文提出了一种基于 FPGA 内部的软 FIFO 互连结构和 4 片 TI 公司的高性能浮点型 DSP——TMS320C6701 构成的多个 DSP 之间通信的并行信号处理系统。

关键词: 并行处理; FPGA; 多 DSP

Implementation of Parallel Signal Processing System Based on FPGA and Multi-DSP

PAN Fangsheng^{1,2}, ZHAO Feng¹, XI Jun², LUO Yi²

(1. School of Micro-electronic, Shanghai Jiaotong University, Shanghai 200030; 2. East China Institute of Computing Technology, Shanghai 200233)

【Abstract】 In the field of high-speed real-time signal processing, the conventional single-chip system can hardly achieve goals due to poor processing ability. Therefore, multiple-processor parallel signal processing system turns to be the focus. The paper presents a parallel processing system based on 4 chips of TMS320C6701—high performance floating-point DSP of TI. The system accomplishes the communication between DSPs through the software-FIFO architecture in FPGA.

【Key words】 Parallel processing; FPGA; Multi-DSP

1 概述

计算机技术和数字信号处理技术的飞速发展, 使得单个 DSP 在包括医学、图像处理、军事等很多领域中很难满足特定任务对系统实时性的要求, 从而使利用多 DSP 并行处理技术实现高速数据传输和处理成为研究的重点。本文介绍了一种基于 FPGA 的多 DSP 通用并行处理系统的实现, 并行处理模块选用 TI 公司的 TMS320C6000 系列处理器 TMS320C6701。TMS320C6701 (以下简称 C6701) 是 TI 公司的第 1 代采用 VelociTI™ 高性能超长指令字结构 (VLIW) 的高性能 32 位浮点型数字信号处理芯片, 其运行时钟频率最高可达 167MHz, 峰值处理速度可以达到 1GFLOPS 和 334MMACS 的运算能力。除了高速的内核处理能力外, C6701 的另一个最大的优势在于它强大的通信接口, 能够很好地支持数字信号处理器同几乎所有类型的外围设备进行数据交互, 十分适合组成高性能浮点的多 DSP 系统。

针对 C6701 的强大片外扩展能力, 利用 EMIF 接口与外部控制器或存储器进行数据交换是很普遍的思路, 应用相当广泛。通常在两片 C6701 互连的系统中采用双口 RAM 进行数据交换, 可以实现很高的性价比。随着处理器数目的增多, 双口 RAM 或多端口 RAM 在控制、布线以及价格等多方面的劣势逐渐显现出来。在这种情况下, FIFO 互连无疑是更好的选择。发送方 DSP 先将数据送往 FIFO, 一旦 FIFO 写满, FIFO 会向接收方 DSP 请求中断, 当交换数据较多时, 接收方 DSP 可以 DMA 方式读取数据, 以避免对系统整体运行速度的干扰。

用 FIFO 互连的 4 片 C6701 的系统结构如图 1 所示。这种方法的缺点也很明显, 就是需要较多的 FIFO 器件。例如

当 4 片 DSP 进行全互连通信时, 需要 6 个双向 FIFO 或 12 个单向 FIFO。

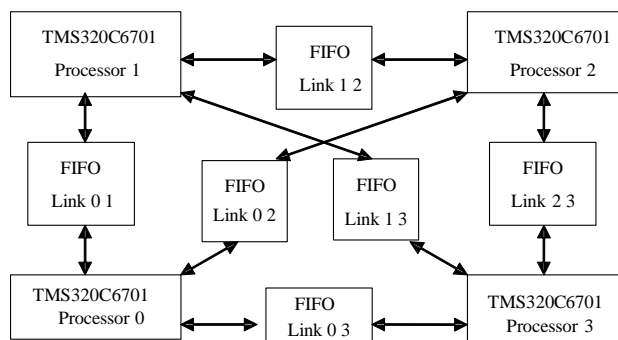


图 1 FIFO Link 实现 4DSP 互连

FPGA (Field Programmable Gate Array), 即现场可编程门阵列, 它是作为专用集成电路 (ASIC) 领域中的一种半定制电路而出现的, 既解决了定制电路的不足, 又克服了原有可编程器件门电路数有限的缺点。FPGA 芯片是小批量系统提高系统集成度、可靠性的最佳选择之一。在本系统中, 选用了 Altera 公司的 Stratix 系列的 FPGA, 利用其内部丰富的 RAM 资源, 建立多 FIFO 的互连结构。

2 多 DSP 并行处理系统的硬件设计

图 2 是通用信号处理模块的系统结构。系统的信号处理单元由 4 片 TMS320C6701 组成的 DSP 阵列构成 (P0~P3),

作者简介: 潘昉晟(1981-), 男, 硕士生, 主研方向: 嵌入式系统和无线通信; 赵峰, 博士; 奚军, 高工; 骆意, 助理

收稿日期: 2006-01-07 **E-mail:** panfangsheng@hhnec.com

每个 DSP 都有各自的片外存储器扩展,包括 SDRAM 和 FLASH 存储器(其中 P0 增加了 SBSRAM 存储器),4 个 DSP 之间通过 EMIF 与 FPGA 的连接实现数据的通信,整个系统通过 P0 的 HPI 总线连接到 PCI 总线上,从而实现与数据处理机(主机)的通信。

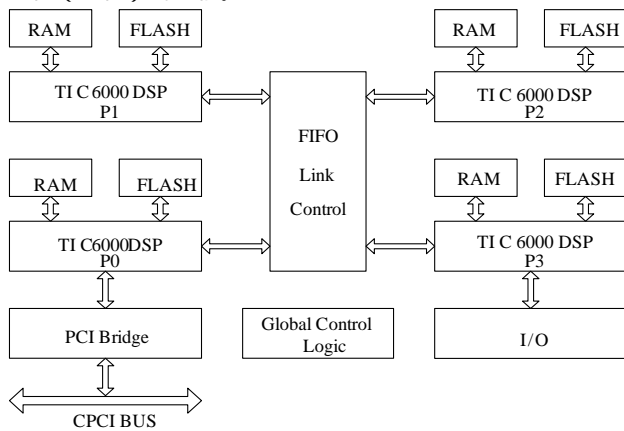


图 2 通用信号处理模块结构

3 FPGA 内部 FIFO 互连结构的实现

在本系统中,基于 FPGA 的 FIFO 互连结构是实现多 DSP 间通信的交换枢纽,FIFO 工作特性的好坏直接影响到整个数据处理的结果。如何利用最少的系统资源,以最灵活的方式为系统提供高速高效的数据交换是该部分设计需要考虑的首要问题,因此 FIFO 控制器的设计便成为设计的关键。

3.1 FIFO 概述

FIFO (FirstInFirstOut) 是一种先进先出存储器,在数据结构上属于队列型存储结构。FIFO 器件按照读写时钟的要求可分为异步 FIFO 和同步 FIFO,按传输方向可分为单向 FIFO 和双向 FIFO。目前的 FIFO 多是基于双端口 RAM 的结构,内部具有输入、输出两套数据线;独立的读写地址指针分别在读写时钟的控制下顺序地从存储单元存取数据,类似一个环形指针(从第 1 个至最后 1 个再转回第 1 个存储单元);控制比较部分给出 RAM 空或满或半满的状态指示。其结构如图 3。这种 FIFO 芯片的深度和宽度都可以做得很大,速度也与 RAM 的性能相差不多。

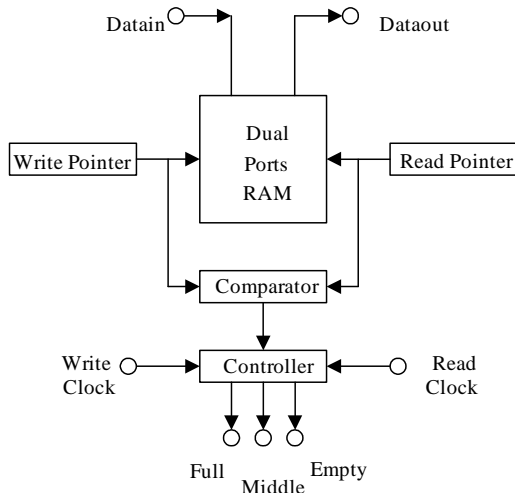


图 3 基于双端口 RAM 结构的 FIFO

在含有多时钟的系统中存在着如何设计基于异步时钟的系统之间的接口的问题。异步 FIFO 是解决这个问题一种简便、快捷的解决方案。使用异步 FIFO 可以在两个不同时钟

系统之间快速而方便地传输实时数据。在网络接口、图像处理等方面,异步 FIFO 得到了广泛的应用。典型的异步 FIFO 读写时序如图 4 所示。

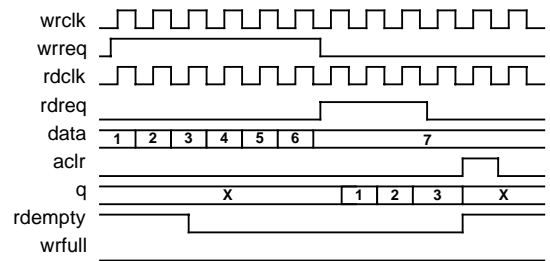


图 4 异步 FIFO 读写时序

当写请求 wrreq 有效期间,数据在 wrclk 的上升沿写入 FIFO 中的存储区;在读请求 rdreq 有效期间,FIFO 内部数据在读时钟 rdclk 的上升沿之后在输出数据线上出现。rdempty 和 wrfull 分别为读空和写满的标志信号。在使用时有两种实现方案:(1)读写方分别提供持续的读写时钟,并且需要控制读请求或写请求的脉冲宽度以实现数据的准确写入和读出;(2)将 FIFO 的读写请求均设为有效,直接将读写方各自的读写请求接入 wrclk 和 rdclk,这样就不需要考虑脉宽的问题。很明显第(2)种方案在异步通信的情况下更有优势。

3.2 FIFO 互连结构在 Stratix 系列 FPGA 中的实现

针对以上实际 FIFO 的特点,高性能的 FPGA 是确保系统实时性要求是否能够满足的关键。Stratix 系列 FPGA 基于 1.5V 0.13μm 全铜 SRAM 工艺,具有最多达 114 140 个 LE、10Mbit 嵌入存储器、28 个 DSP 块(专为 DSP 应用优化的 224 个乘法器)和高性能 I/O 能力。Altera 公司的 Stratix 器件是复杂高性能系统的理想方案,具有丰富的先进特性,包括:

- (1)高性能架构加快模块化设计,获得最大的系统性能;
- (2)丰富的 TriMatrix™存储资源进行片内存储;
- (3)高带宽 DSP 块用于大量信号处理的应用;
- (4)经验证的基于 True-LVDS™电路的差分 I/O 技术,具有 840Mbps 性能;
- (5)健全的时钟管理和频率合成,能管理片内和片外时序,用全功能的增强锁相环(PLL)获得最大的系统性能;
- (6)差分 and 串行片内匹配确保最佳的信号质量和可靠的数据传送。

当一个设计中包含多个同构 FIFO 结构时,常见的设计思路是设计一个通用的基本模块,再用这个模块生成若干个实例,由 FPGA 开发系统自动映射每个 FIFO 到 FPGA 的 RAM 块中。通常在 FPGA 中,当 FIFO 规模超过 RAM 模块的容量时,综合系统会将两块 RAM 区拼接起来提供所需容量,而第 2 块 RAM 区中多余的存储资源就无法再使用,这样就会产生资源的浪费,当 FIFO 数目很多时,所浪费的资源也是很可观的。在 Stratix 系列 FPGA 中,Altera 公司为我们提供了另外一种方案,使用 FIFO Partitioner。

FIFO Partitioner 是一种映射多个 FIFO 至同一物理存储块中的工具,它可以自动地产生必要的逻辑,将 M-RAM 区分割成多个用户自定义的 FIFO。当 FIFO 功能无法有效地在 M512-RAM 或 M4K-RAM 区中实现时,使用 FIFO Partitioner 是很高效的手段。

图 5 描述了利用 FIFO Partitioner 生成的 FIFO 模块。该模块由 12 个同构的单向异步 FIFO 组成,满足 4 个 DSP 全互连通信的需要,每个 FIFO 存储深度为 128 个字,字长为 32

位,写入和读出数据总线均为 32 位,带有空标志、满标志和写入字数计数标志。同时还需要异步清零信号输入以及供给模块内部用于实现 FIFO 控制的高速 TDM_CLK 的输入。TDM_CLK 的频率要求是 N 个 FIFO 模块中最高读写时钟的 N 倍,因此为了提高 FIFO 模块的带宽,需要将其设定在系统所能实现的最高频率,在本系统中通过对外部输入的 25MHz 时钟倍频,用 FPGA 内部的 PLL 模块来实现 250MHz 时钟,这样单个 FIFO 的读写时钟最高频率也被限定在 20MHz。

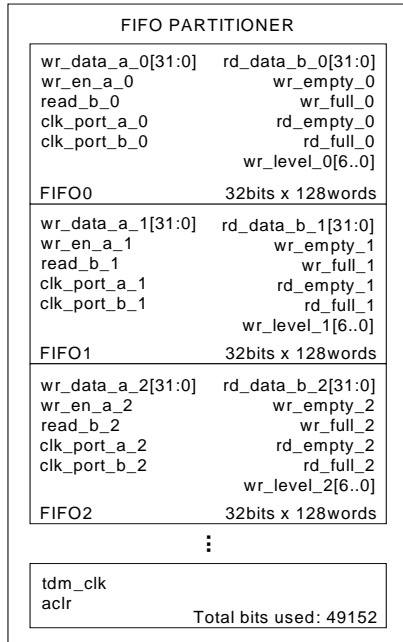


图 5 FIFO Partitioner

FIFO 模块构建好以后,接下来需要考虑的是 DSP 同 FIFO 的连接以及不同 DSP 之间的通信控制。由于 4DSP 的系统具有对称性,以 DSP1 和 DSP2 为例进行描述。DSP1 的 32 位数据总线作为 FIFO1TO2 的数据写入总线, DSP1 的 AWE 反相后作为写入时钟,写使能由 DSP1 地址译码产生,仅当 DSP1 向 FIFO1TO2 中写数据时有效; DSP2 的数据总线作为数据输出端, DSP2 的 ARE 反相后作为读出时钟,读使能由 DSP2 地址译码产生,仅当 DSP2 从 FIFO1TO2 中读数据时有效。如果每次通信量都与 FIFO 容量相同,可以将 FIFO1TO2 的写满标志作为 DSP2 的外部中断信号,通知 DSP2 读取 FIFO 数据;如果每次通信量不定,当 DSP1 向 FIFO1TO2 中以 DMA 方式写入数据结束后,可以由 DSP1 向 DSP2 发出一个脉冲信号作为 DSP2 的读 FIFO 中断。每个 DSP 向相应地发往 DSP2 的 FIFO 写完数据后都会中断 DSP2,这样 DSP2 读 FIFO 的中断源就有 3 个,为了鉴别中断源,需要在 FPGA 设置表示中断源和中断目的的中断标志寄存器,当 DSP2 响应读 FIFO 中断后,就能通过读取中断标志寄存器来判断从哪个 FIFO 中读取数据。相应的 VHDL 代码如下:

```

CONTROL1    <=  DSP1_ADDRESS & DSP1_EN_L;
CONTROL3    <=  DSP3_ADDRESS & DSP3_EN_L;
CONTROL4    <=  DSP4_ADDRESS & DSP4_EN_L;
FIFOWR12_END <=  '1' WHEN CONTROL1="11101000"
ELSE '0';
FIFOWR32_END <=  '1' WHEN CONTROL3="11101000"
ELSE '0';

```

```

FIFOWR42_END <=  '1' WHEN CONTROL4="11101000"
ELSE '0';
FIFOWR12_INT <=  FIFOWR12_END ;
FIFOWR32_INT <=  FIFOWR32_END ;
FIFOWR42_INT <=  FIFOWR42_END ;
DSP2_INT     <=  FIFOWR12_INT OR FIFOWR32_INT
OR FIFOWR42_INT ;
INT_FLAG     <=  "00000000000000000000" &
FIFOWR12_END & FIFOWR13_END & FIFOWR14_END
& FIFOWR21_END & FIFOWR23_END &
FIFOWR24_END
& FIFOWR31_END & FIFOWR32_END &
FIFOWR34_END & FIFOWR41_END
& FIFOWR42_END & FIFOWR43_END ;

```

当 FIFO 的读写时钟均工作在 20MHz 时,整个 FIFO 互连结构可以达到的带宽为 $20M \times 4 \times 32 / 8 = 320MB/s$ 。把读 FIFO 中断及相应的 DSP 的响应时间考虑在内(约为 100ns,相当于 2 个 FIFO 的写周期),则实际带宽比 320MB/s 稍小。

4 硬件系统的并行效率评价

在单处理器系统中,处理一个简单脉冲压缩算法的处理时间大致包括等待数据更新、传输和脉冲压缩两部分;而多处理器系统则可以以流水线方式对脉冲压缩部分进行完全的分布式并行处理。

测试结果如表 1 所示。处理数据长度为 3 584 点。DSP 主频 133MHz。

表 1 测试结果

处理阶段	处理周期数		处理时间 (ms)	
	多处理器系统	单处理器系统	多处理器系统	单处理器系统
脉冲压缩	28 236	125 903	0.353	1.573
数据传输	14 336	14 336	0.179	0.179
总计	42 572	140 239	0.532	1.752
加速比	3.29			
并行效率	0.84			

5 结束语

根据以上分析可以得知:本系统作为一个通用并行处理平台,能够对大数据量实时信号进行快速处理,对于不同任务设计可以采用不同的并行算法,并可获得不同的并行效率。而系统运用了高效率的硬件并行结构,为软件并行算法的实现创造了条件,使其达到了较好的性能,已达到工程实际应用水平。

参考文献

- 1 SX-8[Z]. <http://www.hpce.nec.com/>.
- 2 卡 勒. 并行计算机体系结构: 硬件/软件结合的设计与分析[M]. 北京: 机械工业出版社, 2002: 20-27.
- 3 Texas Instruments. TMS320C6000 Peripherals Reference Guide[Z]. 2001.
- 4 Texas Instruments. TMS320C6000 CPU and Instruction Set Reference Guide[Z]. 2000.
- 5 奥本海姆, 谢弗. 数字信号处理[M]. 西安: 西安交通大学出版社, 1996.
- 6 倪 明, 柴小丽, 蒋其明等. 计算机结构与系统设计[Z]. 华东计算技术研究所, 2002.
- 7 任丽香, 马淑芬. TMS320C6000 系列 DSPs 的原理与应用[M]. 北京: 电子工业出版社, 2000.