

# 数据并行计算机体系结构研究

王鹏飞, 张发存, 段敬红

(西安理工大学计算机科学与工程学院, 西安 710048)

**摘要:** 通过对国内外并行计算机体系结构的分析与研究, 提出一种面向多媒体应用的嵌入式数据并行计算机体系结构模型, 将可重构总线与 PIM 技术相结合, 弥补传统 PIM 体系结构下处理元之间通信复杂度高、结构可适应性弱等不足。描述其主要部件的功能和结构, 定义该体系结构计算机的指令集, 通过一个典型的算法样例介绍其汇编语言程序组成和并行计算过程。

**关键词:** PIM 技术; 数据并行体系结构; 可重构总线; 处理元阵列; 指令集体系结构

## Research on Data Parallel Computer Architecture

WANG Peng-fei, ZHANG Fa-cun, DUAN Jing-hong

(School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China)

**【Abstract】** By the analysis of the architectures of parallel computer, an embedded data parallel computer architecture model is proposed for multimedia processing applications. The limitations of high communication complexity and weak adaptability of the conventional PIM architectures is bridged by this model combined the PIM technology with the reconfigurable bus. The main components and the instruction set are described in detail. A typical algorithm example is given to show the composition of assembly language program and the process of parallel computation.

**【Key words】** PIM technology; data parallel architecture; reconfigurable bus; Processing Element(PE) array; instruction set architecture

DOI: 10.3969/j.issn.1000-3428.2011.15.081

### 1 概述

随着各个领域对高性能计算的要求越来越高, 尤其是多媒体领域大数据量高实时性的需求, 使得传统的单处理器体系结构已经很难适应大规模并行计算的需求, 于是多处理器并行体系结构逐渐成为研究的热点。PIM<sup>[1]</sup>(Processor in Memory)技术由于在 DRAM 芯片上集成了处理器, 从而降低了存储延迟, 并增加了处理器与存储器之间的数据带宽, 使得它成为先进微体系结构设计的一个重要方法。目前针对不同的应用领域已经提出了多个基于 PIM 的体系结构, 如 IRAM、HTMT、FlexRAM、DIVA 等。为了更好地发挥 PIM 技术的优势, 新技术也在不断地被引入, 如使用动态可配置的 SoC(System on Chip)来提高 PIM 平台适应性、伸缩性的 DCIM 体系结构以及采用二维结构的 Ssim 模型<sup>[2]</sup>。基于 PIM 的体系结构主要被用于数据密集型处理, 如多媒体图像和视频压缩、运动估计等方面<sup>[3]</sup>。

然而以上结构中的一个不足是处理元(Processing Element, PE)之间数据的交换要通过专用的路由部件与通信总线来完成, 实现复杂度高, 并且 PE 之间的互连网络都是固定的, 很难根据具体的应用灵活配置。本文将可重构总线与 PIM 技术结合起来, 提出一种新型的数据并行计算机体系结构, 它是基于 PIM 技术的, PE 之间的通信由可重构总线来完成, 总线可根据具体应用来配置, 既满足多处理器并行运算的需求, 同时也降低通信开销和复杂度。

### 2 基于 PIM 技术的数据并行计算机体系结构模型

如图 1 所示, 基于 PIM 技术的并行计算机由宿主机(Main Processor, MP)、PIM 协处理器(PIM Coprocessor)以及指令、数据总线组成, 其中, PIM 协处理器包含 PE/DRAM 阵列、控制器和连接处理元的可重构总线。

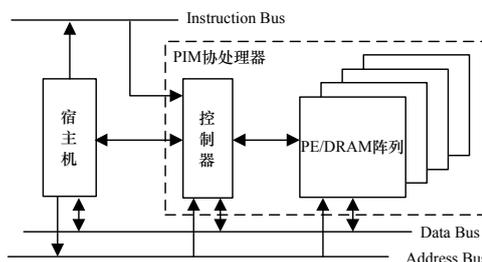


图 1 基于 PIM 的并行计算机体系结构

处理元阵列由 128 个 PE 组成, PE 之间通过一条可重构总线相互连接, 如图 2 所示。在该体系结构模型中, 根据多媒体处理的特点, PE 节点的个数可以取成 2 的幂次方, 比如 64, 128 等。

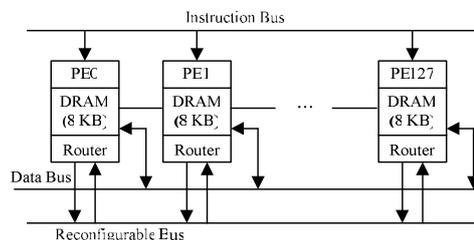


图 2 一维处理元阵列结构

#### 2.1 PE 节点结构

单个 PE 节点由算术逻辑部件 ALU、状态寄存器 PSR、

基金项目: 陕西省教育厅基金资助项目(07JK357); 西安理工大学创新基金资助项目(116-210711)

作者简介: 王鹏飞(1983—), 男, 硕士研究生, 主研方向: 计算机体系结构; 张发存、段敬红, 副教授

收稿日期: 2011-03-16 E-mail: wangpf2008@126.com

移位寄存器 SR, 4 个通用寄存器、路由器和片上存储器 DRAM 组成。PE 内各功能部件之间的通信是通过 3 条总线来实现的, 其中, 总线 A 和 B 是源操作数总线, 总线 C 是目的操作数总线。结构如图 3 所示。

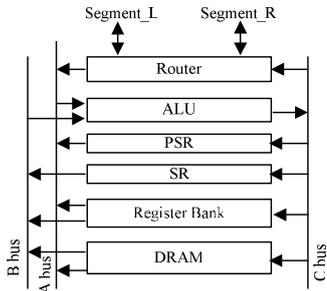


图 3 PE 节点结构

PE 的 ALU 具有 16 bit 字宽, 可以执行 16 bit 运算。ALU 有 4 个数据来源, 分别是通用寄存器、DRAM、PSR/SR 以及立即数, 通过数据源总线 A、B 传入 ALU 数据输入端, ALU 计算完后将数据传到总线 C, 然后 C 可将数据送往通用寄存器、DRAM、PSR/SR 或路由器中任一个。

路由器在节点之间传输数据时起缓冲作用, 数据必须先由寄存器传递给路由器, 然后才能通过总线传输。状态寄存器 PSR、移位寄存器 SR 和 4 个通用寄存器用来协助 ALU 计算, 保存计算过程中的状态、源和目的操作数以及中间结果等数据。

2.2 可重构总线

PE 之间的通信是通过一条可重构<sup>[4]</sup>总线来实现的。如图 4 所示, 128 个 PE 连接到一条可重构总线上, 每个 PE 都可以在总线上进行读写操作。总线上有 128 个开关将总线分成不连贯的子总线或者段总线。对于  $0 \leq i < 128$ , 处理元  $i$  与  $i+1$  之间的总线上设置一个开关  $i$ 。总线上的任何一个 PE 所写的数据在同一段中可以被其他 PE 所读取。

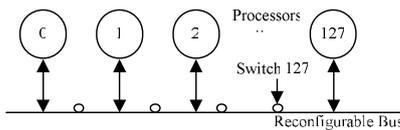


图 4 可重构总线结构

PE 间的通信可分为总线通信和邻近通信两种方式。总线通信是指总线上所有开关都打开, 其中一个 PE 发送数据其他 PE 接收数据。邻近通信是指相邻 PE 之间的数据传输, 通过邻居本地化来实现。邻居本地化是一种总线分离技术, 也是动态可重构的基础。它给阵列中每个 PE 一个标志位, 用于表示 PE 的状态, 用  $f(i)$  表示, 值为 1 表示激活为 0 表示被屏蔽。那么对于  $0 \leq i < j < N$ , 处理元  $j$  是  $i$  的邻居当且仅当  $f(i)=f(j)=1$ , 并且对于任意的  $k(i < k < j) f(k)=0$ , 则认为处理元  $j$  是  $i$  最邻近的被激活 PE, 当然 PE 也可能没有邻居, 在确定邻居节点后就可以进行数据的传输了。

2.3 控制器

控制器包含两大功能模块: 对指令的译码、分发和执行逻辑; 三级指令流水线及三组指令并发控制逻辑<sup>[5]</sup>。

第一部分为 PE 阵列运算提供必要的准备, 如指令、操作数据、控制信号等。第二部分指令流水线完成取指、译码、和执行的三级流水控制, 检测指令之间的数据相关性和逻辑相关性; 三组指令并发是指存在三组不同的资源部件(ALU

逻辑部件、路由器逻辑部件和 DRAM 存储器逻辑部件) 可以接收来自控制器的指令, 同时执行。比如在 ALU 进行运算的同时, 可以进行节点间的通信和数据的存取。

2.4 存储空间分配

基于 PIM 的并行计算机体系结构采用哈佛结构, 即代码空间与数据空间与分开编址。程序存储器在控制器上, 数据存储器和处理元阵列集成在一个芯片上。

程序存储器和数据存储器独立编址。程序存储器按照 8 位字节寻址, 数据存储器按照 16 位字寻址, 阵列包含 128 个 PE, 单个 PE 的片上存储容量是 8 KB, 它是根据多媒体处理的特点, 选取 2 的幂次方作为存储容量大小, 一个 PE 占存储空间的 4096H 字。

3 指令集体系结构 ISA

基于 PIM 技术的数据并行计算机所有的指令长度固定为 32 位。数据类型包含 16 位定点有符号和无符号数。寻址方式与通用计算机的寻址方式基本一致, 包括寄存器方式、立即数方式、位移量方式、相对寻址方式和存储器寻址方式。与传统存储器寻址方式不同的是, 该体系结构下的存储器寻址方式用于 2 种方式, 一是 DRAM 内部访存, 即 PE 访问自身节点的 DRAM 存储区, 二是 DRAM 外部访存, 是宿主主机读写所有 DRAM 数据。

指令按功能分为计算类指令、传送类指令、存取指令、处理元与总线连接指令、控制类指令等, 如表 1 所示。其中算术、逻辑、比较操作指令分为整数操作和立即数操作, 两者的助记符区别在于立即数操作指令后缀会加字母 i, 如 add 与 addi, 表中只列出了整数操作。

表 1 指令集

类型	助记符	功能
算术操作	padd, psub, pmul	加、减、乘、除、
	pdiv, pincls, pmod	条件计数和求余操作
逻辑操作	pand, por	与、或、非、异或运算
	pxor, pnot	
比较操作	pcmpie, pcmpilpcmpile	等于、小于、小于等于比较
	psll, psrl	整数左移、右移
移位操作	psra, pshlr	逻辑右移, 循环左移
	pmove, pmovge	寄存器之间根据源寄存器的值是否等于、大于等于、大于、小于等于、小于、不等于 0 进行数据传送
传送操作	pmovg, pmovle	
	pmovl, pmovne	
总线连接	prcvw, prcve	从 PE 东西端口接收总线上的数据或从东西端口发送数据到总线
	psndw, psnde	路由器与寄存器之间的数据传输
数据存取	pdcone, pdconw, pdconew	处理元的东、西、东西端口与总线断开, 内部端口与总线连接、断开, 及总线复位
	pconiew, pdconiew, prst	
空操作	ppload, ppstore	处理元从内部 DRAM 加载、保存数据、读 PE 索引号
	prindex	
控制操作	pnop, pbnop, pbnnop, penop	空操作、条件空操作开始指令、空操作结束指令
	bt, blink, bsub, return	跳转指令
控制操作	ldn, ldary, ldcell	控制器寄存器、DRAM 数据加载、保存指令
	ldreg, stary	
	loop	循环控制指令

指令根据执行主体的不同又可分为控制器指令和阵列指令(PE 执行)。控制器指令对程序的执行过程进行控制, 即顺序、条件转移或循环。表 1 中的控制操作即为控制器指令, 其余为阵列指令。

在汇编语言级除了基本的机器指令外, 还需要定义一些伪指令, 完成数据定义及存储器分配(DW、DD)、段定义(CREGION、DREGION)、过程定义(PROC)等功能, 其定义方法与传统汇编语言相似。

### 4 应用举例

图像运算根据其计算特点可分为点运算、局部运算和全局运算。点运算比较简单, 每一个 PE 处理一个像素, 局部和全局运算计算复杂度往往较高, 限于篇幅下边通过一个常用于直方图运算中的前缀和算法对该数据并行计算机的并行处理过程加以介绍。

所谓前缀和是指给定数列  $a[n]$ , 令  $S[k]=a[0]+a[1]+\dots+a[k]$ , ( $k=0,1,\dots,n-1$ ), 数列  $S[k]$ 即为数列  $a[n]$ 的前缀和。前缀运算也可以定义在除加法之外的其他组合运算上, 如减法、乘法、求最大值、求最小值、逻辑运算等。图 5 给出了数列前缀和的部分求解过程。

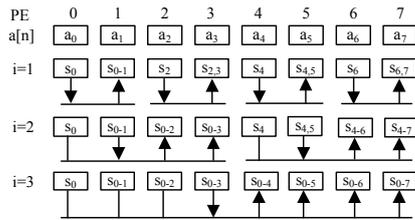


图 5 部分前缀和求解过程

图中 8 个数值经过 3 次计算完成, 每次通过动态配置可重构总线来完成 PE 间的数据传输, 最终计算出结果。汇编语言程序如后续代码所示, 程序由数据段和代码段两部分组成, 0~4 行为数据段, 为输入数据、结果、循环变量分配空间; 其余为代码段。从以上过程可以抽象出包含  $N$  个数的数列前缀和计算过程, 步骤如下:

(1)将源数据  $a[k]$ 加载到  $PE_x$  内部 DRAM 中的  $x$  地址处,  $0 \leq k < N$ ,  $N$  为数列中元素个数, 其值应小于等于 PE 节点个数,  $0 \leq x \leq 0xFFF$ ; 令  $i=1$ ; 该部分对应代码的 6~9 行, 第 8 行省略了部分初始化过程, 第 9 行初始化循环计数器。

(2)总线复位、PE 读自身索引号  $index$ , 对应第 10 行~11 行。

(3)配置总线: 如果  $index \% 2^i = 0$ , 则 PE 断开其内部端口与西侧端口, 对应 12 行~18 行。

(4)发送、接收数据: 如果  $index \% 2^i = 2^{i-1} - 1$ , 则该 PE 发送数据; 如果  $index \% 2^i > 2^{i-1} - 1$ , 则 PE 从总线上接收数据并与本地数据做加法, 并保存加法结果到地址  $x$  处, 对应 19 行~31 行。

(5) $i=i+1$ , 如果  $i \leq \lg N$ , 则回到第 2 步, 否则继续执行, 对应 32-33 行, 第 32 行省略了 DRAM 地址 1 处  $2^i$  与地址 2 处  $2^{i-1}$  值的更新逻辑。

(6)计算结束, 每一个 PE 内部 DRAM 中地址  $x$  处为计算结果, 对应 34 行~35 行。

整个计算过程需要迭代  $\lg N$  次, 相比串行算法速度提升了  $N/\lg N$  倍。汇编语言程序如下:

```

0 DREGION
1   src DW 0,1,2,3,4,5,6,7
2   rslt DW 8 DUP(0)
3   LCnt DW 3 ;log28
4 ENDD
5 CREGION
6 start: LDN AR1, AR0, src
7   LDARY AR1, 0
    
```

```

8   ...
9   LDN AR4, AR0, LCnt
10  begin: prst
11    prindex r3
12    ppload r1, 1 ;load 2^i
13    pmod r3, r3, r1
14    pcmpie psr, r3, 0
15    pnbnp
16    pdconiew ;adjust bus
17    pdconw
18    penop
19    ppload r0, 0 ;load src data
20    ppload r1, 2 ;load 2^{i-1}
21    psubi r2, r1, 1
22    pcmpie psr, r3, r2
23    pnbnp
24    psnde r0 ;send
25    penop
26    pcmil psr, r2, r3
27    pnbnp
28    prcvw r1 ;receive
29    padd r2, r0, r1
30    ppstore r2, 0
31    penop
32    ...
33    LOOP AR4, begin
34    LDN AR1, AR0, rslt
35    STARY AR1, 0 ;save result
36  ENDC
37  END start
    
```

### 5 结束语

PIM 技术缓解了存储器与处理器之间的性能差距, 有效提高了计算机系统整体性能。本文将 PIM 技术的优势以及可重构的设计方法有效地与并行计算结合起来, 提出了该数据并行计算机体系结构模型, 并通过对各种图像算法的汇编语言级描述, 表明该并行计算机体系结构模型具有逻辑可行性。

### 参考文献

- [1] Gokhale M, Holmes B, Iobst K. Processing in Memory: The Terasys Massively Parallel PIM Array[J]. IEEE Computer, 1995, 28(4): 23-31.
- [2] 师小丽, 张发存. 数据并行计算机体系结构建模[J]. 计算机工程, 2009, 35(12): 233-234.
- [3] Jasionowski B J, Lay M K, Margala M. A Processor-In-Memory Architecture for Multimedia Compression[J]. IEEE Trans. on Very Large Scale Integr. Syst., 2007, 15(4): 478-483.
- [4] Todman T J, Constantinides G A, Wilton S J E. Reconfigurable Computing: Architectures and Design Methods[J]. Computers and Digital Techniques, 2005, 152(2): 193-207.
- [5] 陈朝阳, 王忠, 沈绪榜, 等. LS MPP 并行图像处理机[J]. 计算机学报, 2002, 25(3): 292-296.

编辑 陈文