

基于 BAP128 的可重构阵列及图像标记算法

吴 松, 姚庆栋

(浙江大学信息与电子工程学系, 杭州 310027)

摘 要: 基于 BAP128 芯片, 提出了一种可重构阵列, 它改善了 SIMD 阵列的通信性能. 在此基础上, 提出了一个快速图像标记算法. 对于 $N = n \times n$ 的可重构阵列和二值图像, 可以在 $O(\log N)$ 的时间里完成图像标记.

关键词: BAP128; 可重构阵列; 图像标记算法

中图分类号: TN911.73 **文献标识码:** A **文章编号:** 0372-2112 (2000) 05 0108-03

Reconfigurable Mesh Based on BAP128 and Image Component Labeling Algorithm

WU Song, YAO Qing-dong

(Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China)

Abstract: Based on BAP128 chip, we propose a reconfigurable mesh, which improves the communication performance of SIMD array. Using the reconfigurable mesh, we give a fast image component labeling algorithm. On a binary image of size $N = n \times n$ stored one pixel per PE, the labeling algorithm can run $O(\log N)$ time.

Key words: BAP128; reconfigurable mesh; image component labeling algorithm

1 引言

二维 SIMD 阵列以其简单的拓扑连接及与图像间的自然映射关系, 成为最适合图像理解中低层处理的结构. 为改善 SIMD 阵列的通信性能, 出现了一种可重构阵列^[1~4], 在指令控制下, 可重构阵列的广播总线连接可以动态改变. 可重构阵列的优点有: 广播总线的采用降低了通信直径; 总线结构的动态配置可适合不同的算法.

BAP128^[5]是浙江大学信息与通信工程研究所设计开发的 SIMD 阵列芯片, 使用 BAP128 已经构成规模为 16×64 的嵌入式 SIMD 结构实时图像处理系统^[6]. 本文在 BAP128 基础上提出了一种可重构阵列作为改进 BAP128 结构的一种方案, 以此为基础提出了快速图像标记算法, 并将其与 BAP128 阵列上的图像标记算法的性能进行了比较.

2 基于 BAP128 的可重构阵列

2.1 BAP128 SIMD 芯片

BAP128 是浙江大学信息与通信工程研究所为实时图像处理而设计开发的大规模 SIMD 阵列芯片. 每片 BAP128 上集成了 128 个处理器单元 (PE), 排成 8×16 的阵列. 每个 PE 单元为一个位串行处理器, 内部有一个一位 ALU, 四个位寄存器, 一个可变长度移位寄存器, 一个激活控制标志寄存器和 256 位 SRAM.

2.2 基于 BAP128 的可重构阵列

在可重构阵列中, 每个 PE 通过 1bit 的总线与东、西、南、

北四个方向上的 PE 连接. 每个 PE 具有四个端口: N、S、E、W, 端口间存在可控制的开关, 通过开关的通断可以控制 PE 中端口连接的动态改变. 在 BAP128 的基础上, 我们在每个 PE 内增加四个开关, 开关由相应增加的四个寄存器控制, 同时将 PE 内部的 NS、EW 寄存器分别复用为广播发送、广播接收寄存器, 另外再增加广播允许寄存器用以控制总线上的广播, 加上必要的多路选择器, 同时在控制指令集中增加相应的控制指令, 这样不需要增加太多的片内资源, 就可以将 BAP128 改进为可重构阵列. 一个 4×4 的可重构阵列可参见图 1.

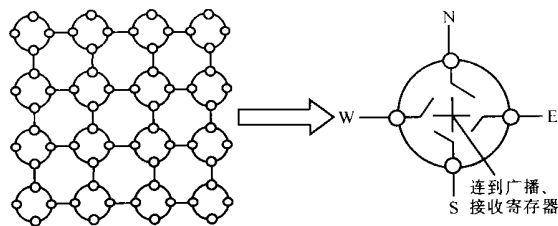


图 1 4×4 可重构阵列结构

由图 1 可见, 每个 PE 的四个端口分别通过开关与 PE 中的广播和接收寄存器相连, 这样每个 PE 内部的端口连接共有 16 种方式, 如图 2 所示.

每个 PE 内部的广播允许寄存器决定本 PE 在接收到广播指令时是否将广播寄存器中的数值广播到总线上, 在接收到广播指令的同时, 广播允许寄存器为 1 的 PE 将广播寄存器中的数值广播到总线上, 同时总线上的各 PE 将总线上的数

值存入接收寄存器。可重构阵列支持多广播模式,也就是说位于同一条总线上的多个 PE 可以同时向总线广播,但前提为这些 PE 广播相同的值(0 或 1)。

在已有的可重构阵列的模型中, RMESH^[1]不存在图 2 中 E、W、N、S 四种连接方式, PARBS^[2]、MRN^[3]允许每个 PE 内部存在两条互不相连的总线。这些模型对总线的广播和接收都是通过总线上的 PE 端口进行的,也就是说 PE 内的广播、接收寄存器需要通过多路选择器分别连接到四个端口上,进行广播、接收操作时需要指定相应的端口;本文的模型通过将四个开关连接成十字型,广播、接收寄存器连到开关的中心点,不需要建立额外的广播、接收寄存器到四个端口的连接就可以进行广播和接收操作。与[1~3]相比,本文提出的模型需要的资源要少,而且由于在进行广播、接收操作时不需指定端口,也简化了相应的指令集。由于结构相对简单,且基于已实现的结构,本文的模型便于实现和操作。

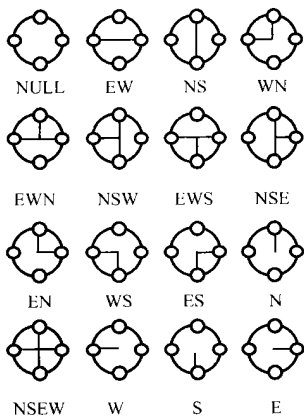


图 2 可重构阵列 PE 端口连接

3 快速图像标记算法

图像标记是图像理解中层任务中的基本操作。在 $N = n \times n$ 的 SIMD 阵列上, $n \times n$ 的二值图像可在 $O(n)$ 时间内完成标记^[7];文[3]在规模为 $2n \times 2n$ 的可重构阵列上对 $n \times n$ 的二值图像提出了 $O(\log N)$ 时间的图像标记算法;文[1]在规模为 $n \times n$ 的可重构阵列上提出了 $O(\log N)$ 时间的图像标记算法,但它可能产生多重分支。

图像标记就是将输入二值图像中各自独立的连接区域标以不同的标号。每个区域内部像素的标号相同。图像标记的最简单方法是,将每个区域内各像素坐标的最大值传遍区域,这样每个区域中的像素具有相同的标号:像素坐标最大值。

可重构阵列规模为 $N = n \times n$, 输入二值图像大小为 $n \times n$, 每个 PE 存储一个像素值。快速图像标记算法的基本思想是将各区域用广播总线相连,求出每个区域中的一个代表点(坐标最大点),同时将这个代表点的坐标广播给区域内的所有点,从而完成标记。

引理:在 $N = n \times n$ 的可重构阵列上,每个 PE 存储一个各不相同 k bit 数值,则总线上 PE 存储的最大值可在 $O(k)$ 时间内求出。

在文[1]中提出了类似的算法,但用到了分支语句,这里的算法则不产生分支,执行时间为常数。首先定义一个内容为 1、位于总线上各 PE 内部的模板,它与各 PE 的某一位的与作为这一位的广播允许,这样如果总线上有 PE 广播,广播的数值必然为 1。广播允许与模板的计算采用下面的公式:

广播允许 = 当前模板 & S

下一模板 = 当前模板 & ($\sim R \parallel S$)

上式中 S 为发送寄存器中的数值, R 为接收寄存器在广播指令后收到的数值, &、 \parallel 、 \sim 分别为与、或和非操作。从 k bit 的最高位开始,由当前模板计算出广播允许赋给广播允许寄存器,在总线上广播,再根据接收值计算出下一模板;然后进行下一位的广播...,经过 k 步,总线上所有 PE 都接收到了总线上各 PE 中存储的最大值。最后生成的模板中,存在唯一的 1 表示此 PE 存储最大值。

下面给出快速图像标记算法:

第一步:建立区域广播总线连接。每个存储非 0 像素的 PE 检查其邻域,建立到存储非 0 像素的邻域 PE 的连接。

第二步:求出每个连接区域的代表点。 $\log N$ bit 阵列坐标事先已经存储在阵列中。对 $\log N$ bit 的阵列坐标使用本节的引理,经过 $O(\log N)$ 时间,可在每个区域中求出一个代表点。

第一步用 $O(1)$ 时间,第二步需要 $O(\log N)$ 时间。这样得到下面的结论。

结论:在规模为 $N = n \times n$ 的可重构阵列上,输入二值图像为 $n \times n$,每个 PE 存储一个像素,图像标记算法可在 $O(\log N)$ 时间内完成。

4 BAP128 阵列与可重构阵列上图像标记算法性能比较

首先给出一些基本操作的时间^[5],每个时钟周期为一拍: BAP 上一位数据移位需要 3 拍,一位二操作数逻辑运算需要 3 拍,阵列屏蔽需要 1 拍,清除屏蔽需要 1 拍,一位数据在不同的内存地址间移动需要 2 拍;在可重构阵列上除了与 BAP 同样的操作外,每个 PE 内部开关的状态设定需要 2 拍,每一位广播操作需要 2 拍。

同样采用区域内最大坐标广播的算法在 BAP128 阵列和可重构阵列上进行图像标记,可将两者的性能对比如下:

可重构阵列上的快速图像标记算法,第一步建立总线连接需要在四个端口上分别进行,每个端口上需要做一位的移位、逻辑与、屏蔽、置开关状态和清除屏蔽,这样四个端口的连接建立需要 36 拍。开关建立后,根据上节的算法,要做 $\log N$ 次循环,每次循环需要做一位广播允许计算、广播和模板计算,共需要 13 拍,这样 $\log N$ 次循环需 $13 \log N$ 拍。整个快速图像标记算法需要 $13 \log N + 36$ 拍。

在 BAP128 阵列上进行图像标记,每一步区域内具有最大坐标的 PE 将坐标值传递给邻域的 PE,直至阵列中的坐标值不再发生变化。这种算法需要的步数由处理的图像中具有最大直径的图像区域决定,对于 $N = n \times n$ 的图像,直径取值从 1 到 $O(N)$ 。假定最大的区域直径为 r ,标记在四个方向上分别进行,每个方向需要做 $1 + \log N$ 位的移位、 $\log N$ 位的比较、一位的与、屏蔽、 $\log N$ 位的数据移动和清除屏蔽,需 $7 + 8 \log N$ 拍,四个方向共需 $28 + 32 \log N$ 拍,标记算法的时间为 $r(28 + 32 \log N)$ 拍。

由以上分析可看出,对于可重构阵列而言,快速图像标记算法的执行时间与具体的图像的复杂度无关,是一个常数,而

对于 BAP 阵列来说, 图像的区域直径与标记算法的执行时间成正比.

5 结论

作为对 BAP128 结构改进的一种方案, 在 BAP128 的基础上, 我们提出了一种可重构阵列, 它不仅增强了阵列的通信性能, 还可根据算法的要求动态重构. 基于可重构阵列的快速图像标记算法, 对于 $N = n \times n$ 的二值图像和可重构阵列, 能够在 $O(\log N)$ 的恒定时间完成标记任务.

参考文献

- [1] R. Miller, et al. Parallel computations on reconfigurable meshes. IEEE Trans. Comput., 1993, 42: 678~ 692
- [2] J. Jang, et al. A fast algorithm for computing a histogram on reconfigurable mesh. IEEE P. A. M. I. 1995, 17(2): 97~ 106
- [3] S. Olariu, et al. Fast component labelling and convex hull computation on reconfigurable mesh. Image and Vision Computing, 1993, 11(7): 447~ 455
- [4] H. Li, M. Maresca. Polymorphic tours network. IEEE Trans. Comput., 1989, 38(9): 1345~ 1351
- [5] 张明. 阵列处理器芯片 BAP 64 的设计与并行算法研究. 博士学位论文. 杭州: 浙江大学, 1995

- [6] X. Chen, et al. SIMD massively parallel processing system for real-time image processing. Int'l Symp. on Multispectral Image Processing (ISMIP' 98), Wuhan: SPIE3545, 1998: 596~ 599
- [7] Hongchi Shi, et al. A fast algorithm for image component labeling with local operators on mesh connected computers. J. Parallel Distrib. Computing, 1994, 23: 455~ 461



吴松 1973 年出生, 1997 年在浙江大学获硕士学位, 同年继续攻读博士学位. 主要研究方向是: 图像理解, 并行结构.



姚庆栋 1932 年出生, 现为浙江大学信息与电子工程系教授、博士生导师. 主要研究领域: 图像编码和通信、高清晰度电视和数字电视、数字通信系统、实时图像处理、图像制导和导航、并行处理、专用集成电路设计等.