一种适于片上路由器的自适应缓冲调整策略*

石 伟,郭御风,窦 强,张 明,任 巨
(国防科技大学 计算机学院,湖南 长沙 410073)

摘 要:在典型的片上网络路由节点中,来自不同方向的报文被存储在相互独立的缓冲资源中。在网络负载不均衡的情况下,某些方向的报文将很快填满该方向的缓冲,而其他方向仍可能有较多的缓冲资源处于空闲状态,这样就导致了网络中的缓冲资源利用率不高,进而影响片上网络的整体性能。提出了一种自适应的片上缓冲调整策略,能够根据网络负载情况动态调节缓冲结构,有效地提高了缓冲资源的利用率。在90nm CMOS工艺下设计实现了多端口共享缓冲资源的片上网络路由器,实验结果表明,在负载不均衡的网络中,提出的路由器能够带来性能改进及功耗降低;在达到相同性能的情况下,新路由器的面积较典型路由器减少了20.3%,而其缓冲功耗节约了41%左右。

关键词:片上网络;低功耗;虚通道;动态调整;层次位线缓冲
中图分类号:TP 302 **文献标志码**:A **文章编号**:1001-2486(2013)03-0048-07

An adaptive buffer regulating scheme for on-chip routers

SHI Wei, *GUO Yufeng*, *DOU Qiang*, *ZHANG Ming*, *REN Ju*
(College of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: In the traditional Network-on-Chip routers, packets from different directions are temporarily stored in different buffer regions, and these buffering resources are independent from each other. Under non-uniform traffic patterns, buffers in some input channel will be crammed by the coming packets quickly, while the others are still in idle state. As a result, the buffers are utilized inefficiently, and it has a negative influence on the overall network performance. In the method proposed, an adaptive buffer regulating scheme that can be used to achieve similar performance by using less buffering resources was introduced. The VLSI implementation of a router with the buffer regulating scheme was completed under 90nm CMOS process. The experimental results show that the proposed router can bring significant performance improvement and power reduction under non-uniform traffic patterns, 20.3% area saving of the proposed router and 41% power reduction of the buffers can be achieved compared to the traditional one.

Key words: network-on-chip; low power; virtual channel; dynamic regulation; hierarchical bit-line buffer

计算与通信之间逐渐增大的代价差距使得设计由计算中心型向通信中心型转变,片上网络(Network-on-Chip,NoC)[1]作为一种新型的通信方式,逐渐取代传统的片上总线通信方式。

缓冲结构在片上路由器中起着重大的作用,直接影响着片上网络的整体性能。另外,研究表明,片上路由器中超过50%的功耗是由缓冲消耗的[2],且缓冲的漏流功耗占路由器总漏流功耗的64%左右[3]。大的缓冲容量能够提高系统性能,而片上资源受限及缓冲的大功耗特性限制了缓冲的大容量设计,提高缓冲资源的利用率已成为当前研究的共同目标。为了提高缓冲的利用率和获得更好的性能,虫孔交换[4]与虚通道(Virtual Channel,VC)[5]两种技术首先被引入路由器设计中。在采用虫孔交换与虚通道的典型路由器中,路由器每个端口中的VC数目和VC深度是相同的,这种静态的对称缓冲结构具有资源利用率低问题。为此,人们提出了动态调整缓冲结构,研究表明动态调整缓冲结构能够有效提高缓冲利用率。

DAMQ缓冲[6]是最早提出的一种基于动态调整思想的缓冲结构,能够提高网络中缓冲的利用率,但仍然存在队头阻塞问题。文献[2]提出了一种虚通道动态分配结构ViChar,根据实际网络负载情况动态调整输入端口VC的数目及深度。在采用相同缓冲资源的情况下,ViChar的性能比典型路由器提高了25%。文献[7]在ViChar基础上提出了一种具有拥塞缓解能力的动态缓冲调整路由器结构,进一步提高了片上网络性能。上述几种动态调整的缓冲结构均旨在研究单端口

* **收稿日期**:2012-10-19
基金项目:国家"核高基"重大专项项目(2009ZX01028-002-002);国家自然科学基金资助项目(61202481,61202123,61202122)
作者简介:石伟(1982—),男,江苏涟水人,助理研究员,博士,E-mail: shiwei@nudt.edu.cn

缓冲资源总量保持不变的情况下怎样提高缓冲资源的利用效率。而不同的应用程序具有不同的通信特征与通信需求,网络中各节点的通信负载必然各不相同。在这种负载不均衡的网络中,如果路由节点各端口具有相同的缓冲资源,必然使得有的端口中资源较为紧张,而其他端口中资源处于相对空闲状态,使得缓冲资源的利用率变低。

为进一步解决资源均衡分布与负载不均衡的对立问题,文献[8]提出一种根据应用定制缓冲的方法,灵活性较差。文献[9 - 11]提出了多个端口共享缓冲资源的片上路由器结构,但这些共享缓冲采用统一的多端口寄存器文件来实现,致使面积或功耗急速增大。我们提出了一种基于层次位线结构的共享缓冲结构[12],片上路由器的多个端口能够有效动态共享缓冲资源,在较低功耗与面积开销的情况下有效提高了路由器的性能。基于提出的层次位线共享缓冲结构,本文对多端口动态缓冲共享技术进行深入研究,提出了一种自适应缓冲调整策略,该策略能够根据实际网络负载将缓冲资源有效地分配给各个端口使用。最后,在 90nm 工艺下设计实现了多端口共享缓冲资源的片上路由器 MPBSR(Multi-port Buffer Shared Router)。

1　NoC 路由器典型结构

文献[13]提出了一种采用了虫孔交换与虚通道技术的片上网络路由器典型结构。该路由器由 P 个输入端口、P 个输出端口、路由计算(Routing Computation, RC)模块、虚通道分配(Virtual-Channel Allocator, VA)模块、交叉开关分配(Switch Allocator, SA)模块及交叉开关组成。

输入端口负责接收并缓存相邻节点传来的报文,当报文的头微片(Head Flit)到达输入端口后,RC 根据报文头中的目的地址信息与路由算法计算出相应的输出端口。因为每个报文都要传输到下一级路由节点的某一特定 VC 中,所以还要为该报文分配一个 VC。虚通道状态(VC Status)寄存器记录了相邻节点中 VC 的使用状态,凡是空闲的 VC 都能够分配给本地报文使用。获得下一级 VC 使用权的报文通过交叉开关传输到输出端口。由于同一输入端口中多个报文可能同时需要传输,或者不同输入端口中的报文可能需要传输到同一输出端口,SA 将交叉开关的使用权分配给仲裁获胜的报文。在典型路由器结构中,本文假定每个输入端口包含 v 个 VC,每个 VC 可以存储 k 个 flit。另外,本文采用二维 mesh 片上网络结构,所以 $P=5$,分别用 E、S、W、N 与 C 表示来自四个相邻路由节点及本地计算节点的连接。

2　多端口缓冲资源动态共享技术

本节提出了一种多端口缓冲资源动态共享技术,路由器中的缓冲资源能够根据网络负载动态地分配给不同的输入端口使用。

2.1　多端口共享 VC 缓冲结构

多端口共享 VC 缓冲结构能够自动调整各输入端口中 VC 的数量来适应通信负载的变化,其结构如图 1 所示。图中每个输入端口包含 m 个私有 VC,这些私有 VC 只能被本端口使用;另外,所有输入端口共享使用 n 个共享 VC。因此单个路由器中包含 $Pm+n$ 个 VC。每个共享 VC 通过多路选择器连接到多个输入端口,选择信号由 VC 动态调整逻辑产生。选择信号在某一时刻最多能够选通一路连接,即任何时刻每个 VC 只能被单个输入端口使用。当某个输入端口负载较重时,调整逻辑会将多个共享 VC 分配给该端口使用;相反地,调整逻辑会为轻负载端口分配较少的 VC。

虚通道的缓冲存储单元可以采用 SRAM 或 D 触发器实现,但其共享访问特性极大地增加了存储体的读写访问端口,进而增大了缓冲的功耗、访问延迟及控制复杂度。设计一款好的缓冲结构直接关系到多端口共享虚通道技术能否有效得到实现。本文基于层次位线(Hierarchical Bit-line)[14]技术设计缓冲结构,使其能够高效支持多端口共享 VC 的策略,并有效解决功耗、延迟及逻辑复杂等问题。在层次位线缓冲结构中,一组存储单元首先连接到子位线上,然后子位线再通过选择器连接到多组高层位线上,这里选择器由单个传输管实现。全局字线(Global Word-line, GWL)用于控制选择器的闭合,它由 VC 调整逻辑控制产生。

层次位线缓冲结构如图 2 所示。为简单起见,图中只给出一组读写端口。图中还标出了 flit 的存储位置及一个虚通道的结构组成。若全局字线 GWL_{ij} 为高,读写子位线则分别连接到高层读写位线上,表明虚通道 i 被分配给端口 j;否则该 VC 处于空闲状态。因此在某一时刻,全局字线 GWL_{i1},…,GWL_{ip} 最多只有一个为 1。这种采用层次位线的缓冲降低了位线的读写负载,从而有效降低了缓冲功耗。为了进一步降低位线的读写负载,*VC* 调整信号可以与读写字线进行逻辑与操作后再连接全局位线上。尽管采用层次位线结构的缓冲减少了每个存储单元的读写端口,但每个存

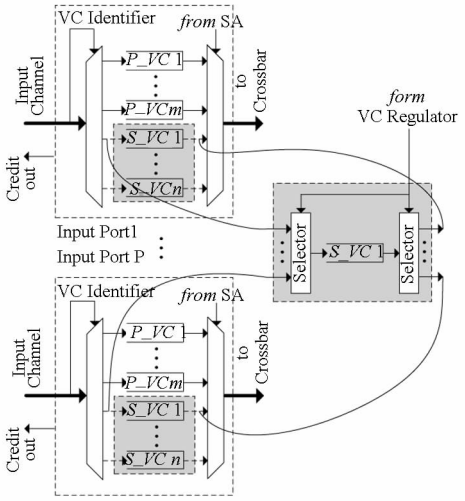

图 1　多端口共享 VC 的缓冲结构

Fig. 1　Buffer structure with multiple-port shared VC

储单元仍然需要 P 组高层读位线与 P 组高层写位线来达到多端口共享的目的。随着工艺的进步,可以采用丰富的高层金属资源来进行位线设计。

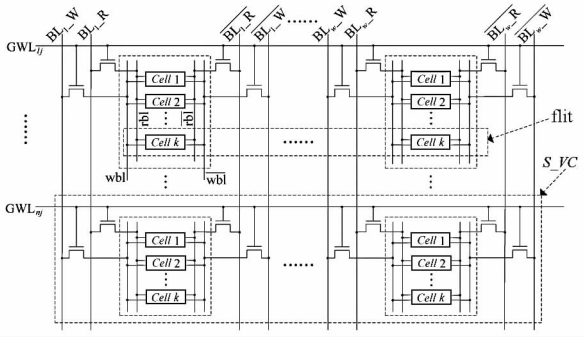

图 2　采用层次位线的缓冲结构

Fig. 2　Structure of hierarchical bit-line buffer

2.2　VC 动态调整策略

VC 的动态调整是 MPBSR 实现的特点。图 3 是 VC 动态调整的结构框图。虚通道调整器(VC Regulator)根据各个输入端口中 VC 的实际使用情况,将共享的 VC 动态分配给不同的输入端口使用。VC 调整器主要包含三个子模块:虚通道状态表(VC Status Table)、可用虚通道表(VC Availability Table)以及动态调整逻辑(Regulation Logic)。

虚通道状态表记录每个共享虚通道的当前状态,空闲(free)或占有(occupied)。当某个虚通道处于 free 状态时,动态调整逻辑可以将该虚通道分配给某个端口使用,然后将此虚通道状态置为 occupied 且记录该端口的编号。虚通道状态表结构如图 4(a)所示。可用虚通道表记录本节点中

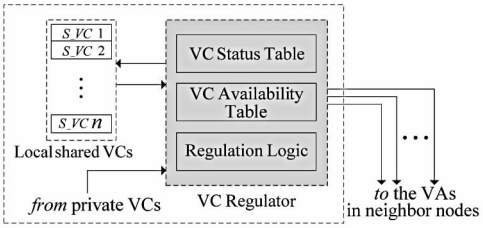

图 3　VC 动态调整框图

Fig. 3　Block diagram of VC regulation

当前已经分配给各个输入端口但还没有使用的 VC 标号,其结构如图 4(b)所示。图中每一行对应一个输入端口,且该行中状态位为 1 的虚通道可以被该方向端口使用。可用虚通道表与 VC 状态寄存器的作用相同,在可用虚通道发生变化后,虚通道调整器及时更新相邻节点中 VC 状态寄存器。动态调整逻辑根据虚通道状态表和可用虚通道表提供的信息动态决定是否将空闲的公共虚通道分配给各端口使用。动态调整逻辑进行 VC 调整的条件包括:1)某个端口中的可用虚通道数少于临界最小值 A 且该端口中已分配虚通道数未达到所允许最大虚通道数 B;2)存在处于空闲状态的公共 VC。临界值 A、B 的设定关系到 VC 动态调整策略能否有效工作,4.1 节将对其进行具体讨论。

VC id	State	Owner
0	occu.	E
1	occu.	W
2	free	Null
⋮	⋮	⋮
⋮	⋮	⋮
n	free	Null

(a) VC Status Table

VC id / Owner	private VCs				shared VCs			
	1	2	…	m	1	2	…	n
E	0	1	…	0	0	1	…	0
S	0	1	…	1	0	0	…	0
W	1	1	…	0	0	0	…	1
N	1	0	…	1	0	0	…	0
C	1	0	…	1	0	0	…	0

(b) VC Availability Table

图 4　虚通道动态调整数据结构

Fig. 4　Table structure for VC regulation

虚通道调整器工作过程主要包括:1)根据 VC 状态表与可用虚通道表的当前信息对 VC 进行动态调整分配,并更新表中信息;2)根据调整分配结果控制本地公共缓冲结构作相应的改变;3)根据可用虚通道表中的信息更新相邻节点中的 VC 状态表,以供相邻节点的 VA 使用;4)当报文的头微片到达 VC 后,将该 VC 在可用虚通道表中的状态置为不可用;5)当报文的尾微片离开私有 VC 后,本地路由节点将发出一个 credit out 信号通知更新 VC 状态表中虚通道的状态;而当报文的尾微片离开共享 VC 后,只是通知更新虚通道状态表与可用虚通道表中的相应信息,并调整共享缓冲结构。

3　MPBSR 路由器 VLSI 结构

基于典型 NoC 路由器结构及上述多端口缓冲共享结构,本节对 MPBSR 路由器进行详细设计。

3.1　虚通道动态调整电路

虚通道调整逻辑关系到缓冲动态调整策略能否正确高效运行。图 5 给出了四端口(E、S、W、N)VC 请求及分配电路结构。图中 m_i 表示端口 i 当前可用 VC 数目,n_i 表示端口 i 当前已经分配的 VC 数目,当端口 i 符合 VC 申请条件,即发出 VC 申请请求信号 R_i。当多个端口同时对少数几个可分配 VC 进行请求时,需要对多个端口的请求进行仲裁。端口优先权寄存器(Port Priority)中记录了各个端口进行 VC 请求的优先级,其中端口 i_1 的优先级最高,端口 i_4 的优先级最低。简单的优先级设定采用 round-robin 方式。图中用 VC 标号寄存器文件记录当前可分配的 VC 标号(从 index 为 1 的寄存器开始记录)。请求 R_i 经过选择开关与加法器后得到不同优先级端口访问 VC 标号寄存器文件的标号,如优先级最高的请求端口将访问 index 为 1 的 VC 标号寄存器。当从可用 VC 标号寄存器文件中读出可分配 VC 标号后需要再次进行置换,以还原到原来的端口请求顺序。其中 j_i 是 Port Priority 中值为 i 的寄存器标号,即 $v[j_i] == i$。置换后的 N_i 是分配给端口 P_i 的可用 VC 标号。如果 R_i 与 N_i 均有效,则说明端口 P_i 请求 VC 成功,需要更新虚通道状态表与可用虚通道表,并对缓冲结构做出相应修改。在 90nm 工艺下,对虚通道动态调整模块进行综合,得到其逻辑延迟约为 0.52ns。

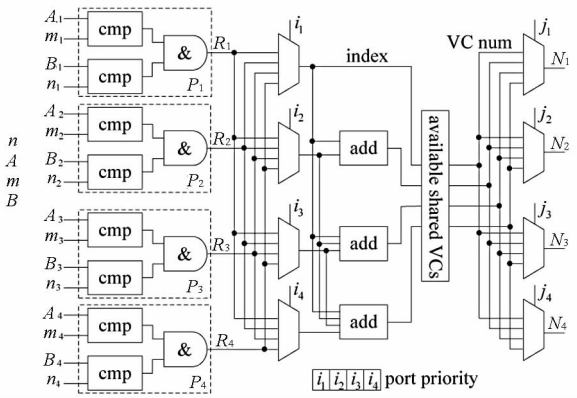

图 5　多端口 VC 请求及分配电路结构

Fig. 5　VC regulation circuit for 4 input channels

3.2　虚通道分配模块

虚通道分配(VA)模块是路由器中另一个重要部分,其性能的好坏直接关系到路由器的性能。如果 VA 延时太大,它将成为流水线的瓶颈。VA 采用两级仲裁结构,如图 6(a)所示。路由器结构中,每个端口的虚通道数最大可以增加到 $m + n$(记作 v')。VA 第一级仲裁中,每个本地 VC 申请一个输出端口。因此,需要 P 个 v':1 仲裁器,每个输入端口中申请某个特定输出端口的请求数减少到 1。也就是说,每个输入通道中最多只有一个报文能够请求某一输出端口。第二级仲裁器对申请同一输出端口的多个输入端口请求进行仲裁。两级仲裁之后,所有输入端口中只有一个虚通道能够申请到某个特定的输出端口。随后,空闲虚通道分配模块(Free VC Allocation)将输出端口对应的下一级空闲 VC 分配给获得该输出端口的本地报文[2]。

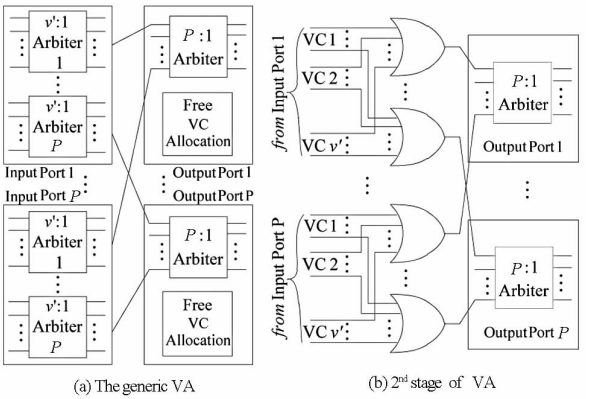

图 6　虚通道分配

Fig. 6　Structure of VC allocation

由于第一级仲裁与第二级仲裁之间不存在偏序关系,第二级仲裁能够提前到与第一级仲裁并行执行。图 6(b)给出了改进后的第二级仲裁结构。每个 VC 有 P 个输入请求,每个请求对应一个输出端口。路由逻辑返回的端口对应的请求为 1,其余为 0。v'输入或门统计出该输入端口是否请求相应的输出端口。P:1 仲裁器决定哪一个输入端口将获得相应的输出端口使用权。

3.3　MPBSR 路由器流水结构

与典型路由器一样,MPBSR 路由器仍采用路由计算(RC)、虚通道分配(VA)、交叉开关分配(SA)及报文传输(Crossbar)四段流水方式实现,其流水线结构如图 7 所示。在 RC 段,当报文的头微片存入 VC 时,该 VC 将被标识为不可用状态,因此需要更新可用虚通道表内容。RC 段的操作可能导致输入端口的可用 VC 数少于临界值,从而触发虚通道调整。VC 调整安排在 VA 段执行,VC 调整结束需要更新可用虚通道表、虚通

道状态表及缓冲结构。在报文传输阶段,当报文的尾微片离开 VC 后,该 VC 可以分配给新的报文使用,因此需要更新可用虚通道表与虚通道状态表。从图中可以看出,RC、VA 及 Crossbar 段都需要对可用虚通道表与虚通道状态表进行修改。

在 MPBSR 中,当 VC 分配给相邻节点中报文使用后,只有当该报文真正到达本地节点后,VC 动态调整逻辑才能感知到可用 VC 减少的信息,然后再进行 VC 调整。但在 VC 分配至 VC 调整这段时间内,可能导致输入端口没有空闲 VC 可用,进而影响系统的性能。这种缓冲信息的延迟感知是本文路由器调整策略面临的一个重要问题。在传统路由器中,VC 分配不涉及该报文在 SA 段能否获得交叉开关使用权。本文提出一种 VC 延迟分配策略,只有报文在申请 VC 和交叉开关使用权都成功以后,才将 VC 真正分配给该报文使用。即只有在报文能够真实传输到下一节点时才修改虚通道状态表中的状态,将分配的 VC 置为不可用状态。这种 VC 分配方式能够提高 VC 的利用率,避免 VC 被分配给未传输的报文。当报文获得交叉开关使用权后,该报文将一直占有交叉开关的使用权,直至缓冲中的报文微片传输结束或遇到流控[7]。VC 延迟分配及报文连续传输有效解决了本文路由器缓冲信息延迟感知的问题,并降低了报文的平均传输延迟。

4 实验结果

为了分析自适应缓冲共享策略对片上网络路由器性能的影响,我们设计实现了一个时钟精确的共享缓冲片上网络模拟器 NoCsim。该模拟器支持不包括随机通信(Random)、热点通信(Hotspot)、矩阵置换通信(Matrix Transpose)及各种定制的通信模式。实验中,模拟器采用 8×8 二维 mesh 结构及 XY 维序路由方式,每个虚通道的长度为 8 个微片。为获取稳定的网络性能,每次模拟的时间为 200 000 个时钟周期,数据统计在 100 000 个时钟周期后进行。

4.1 临界值分析

临界最小值 A 关系到 VC 动态调整的粒度。临界值 A 越小,VC 越能被更细粒度地共享,越能体现本文动态调整策略的优势。通过对采用 $A=1$ 的 MPBSR 进行分析,3.3 节介绍的 VC 延迟分配及报文连续传输策略可以解决本文路由器的缓冲信息延迟感知问题,因此,临界值 A 取 1。增加 VC 数目能够有效提高链路的利用率,当 VC 数目达到一定数量后,网络的性能将受限于链路的带宽。一旦链路带宽成为网络性能瓶颈,再增加 VC 数量将很难进一步获得性能提高,反而浪费了缓冲资源。因此,临界值 B 关系到缓冲共享技术能够带来多大的性能提高及其资源利用效率如何。为每个端口设定不同的虚通道数目,进行性能仿真,仿真结果表明,虚通道数目小于等于 4 时,虚通道的增加能够有效提高系统吞吐量;虚通道数目大于 6 以后,性能的提高越来越有限。因此,B 确定在 4 ~ 6。

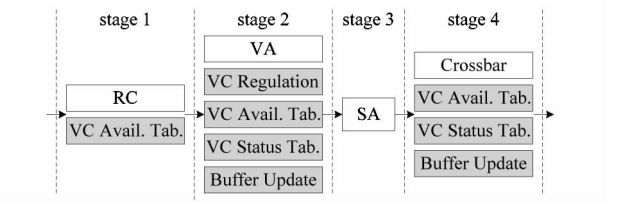

图 7 MPBSR 路由器流水结构

Fig. 7 MPBSR pipeline

4.2 网络性能分析

在矩阵置换及热点两种模式下,对 MPBSR 及典型路由器分别进行了性能仿真。典型路由器每个输入端口的虚通道数是相同的,分别为 1(No VC)、2(Uniform 2 VC)和 3(Uniform 3 VC)。MPBSR 每个输入端口都有一个私有的虚通道,其余虚通道为多个端口共享使用。临界值 A 设为 1,而 B 设为 4。图 8 和图 9 分别是两种路由器在矩阵置换与热点通信模式下的性能比较。从图中可以看出,当网络的报文注入率到达一个临界点后,报文的平均延迟将急剧上升。临界点对应的注入率与平均延迟分别称为系统的饱和注入率及饱和平均延迟。饱和注入率正比于整个网络的吞吐量。

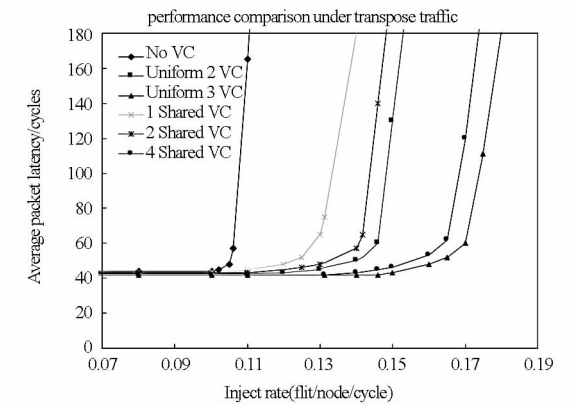

图 8 矩阵置换模式下性能比较

Fig. 8 Performance comparison under matrix transpose

图 8 中,在没有采用虚通道的网络中,系统的饱和注入率为 0.106 flit/node/cycle。随着共享

VC 数目的增加,路由器的饱和注入率将迅速提高。当共享 VC 数目达到 4 时,系统的饱和注入率为 0.167 flit/node/cycle;而每个端口采用 3 个 VC 的典型路由器的饱和注入率仅为 0.17 flit/node/cycle。图 9 表明热点模式下的系统饱和注入率具有与矩阵置换模式下系统饱和注入率相似的趋势。2 个共享虚通道的 MPBSR 的性能接近于两虚通道典型路由器性能,而 4 个共享虚通道的 MPBSR 的性能接近于三虚通道典型路由器性能。MPBSR 的性能要优于具有相同数量缓冲的典型路由器。

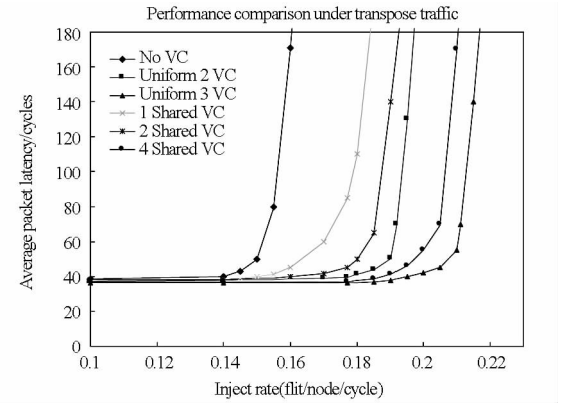

图 9　热点模式下性能比较

Fig. 9　Performance comparison under Hotspot

4.3　面积与功耗

我们在 90nm 工艺下对传统结构的缓冲及具有自适应调整能力的缓冲进行了实现,其中控制逻辑采用基于标准单元的半定制方式实现,存储阵列采用全定制方式实现。图 10 给出了多种不同配置情况下两种缓冲结构的面积比较,其中传统结构缓冲记为 C,共享结构缓冲记为 U。在相同容量的情况下,具有调整能力的统一缓冲面积稍大于采用分布实现的传统缓冲面积。增加的面积主要由连接两级位线的晶体管引起的。由于 90nm 工艺具有丰富的布线资源,多个端口对应的多条读写位线没有引起较大的面积增加。

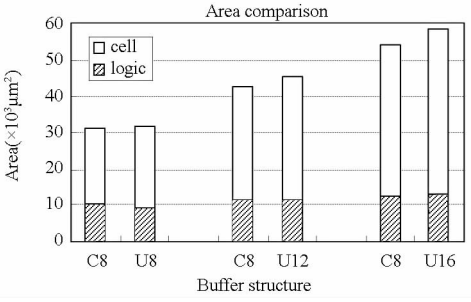

图 10　不同结构及配置的缓冲面积比较图

Fig. 10　Area comparison of different buffers

在 500MHz 的工作频率下对两种缓冲结构进行功耗测试,两种缓冲的单端口读写功耗如图 11 所示。在 VC 数目相同的情况下,本文的缓冲结构具有较低功耗,主要是层次位线结构降低了位线负载电容。由 4.2 节的性能分析结果知,具有 4 个共享虚通道(U8)的 MPBSR 的性能接近于三虚通道(C12)典型路由器性能。在这种配置下,U8 的功耗与面积较 C12 分别降低了 41% 与 25%。

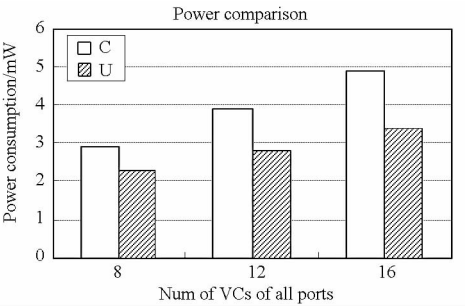

图 11　不同结构及配置的缓冲功耗比较

Fig. 11　Power comparison of different buffers

在 90nm 工艺下,对 MPBSR 和典型路由器进行了实现,其中典型路由器缓冲的配置为 C3×4,而 MPBSR 缓冲的配置为 U8。典型路由器与 MPBSR 的面积比较如表 1 所示。在 MPBSR 中,每个端口的潜在 VC 数目要多于典型路由器,并且 MPBSR 需要进行动态 VC 调整,因此控制逻辑面积要大些。MPBSR 端口多个 VC 使得 SA 面积增加了 $1.9\times10^3\mu m^2$,而 VA 面积的减小主要由于其第二级仲裁面积的减小。两种路由器的其他部分变化较小,面积基本相当。MPBSR 总面积较典型路由器面积减小了约 20.3%。

表 1　典型路由器与 MPBSR 的面积比较($\times10^3\mu m^2$)

Tab. 1　Area comparison of generic router and MPBSR

	典型路由器	MPBSR
路由逻辑	1.3	1.3
缓冲	42.27	31.72
VC 控制逻辑	9.7	12.2
VA	38.1	23.5
SA	2.2	4.1
交叉开关	6.4	6.9
总计	99.97	79.72

5　结论

提出了一种根据片上网络中实际负载情况,为各个端口动态分配缓冲资源的自适应缓冲调整策略,并设计实现了一种片上路由器 MPBSR。实验结果表明,在网络负载不均衡的情况下,

MPBSR 路由器具有一定的优势。在达到相当性能的情况下,MPBSR 路由器较典型路由器节约了25%的缓冲资源与41%的缓冲功耗,路由器整体面积则减小了20.3%。

参考文献(References)

[1] Dally W, Towles B. Route packets, not wires: On-chip interconnection networks [C]//Proceedings of the Design Automation Conference (DAC), 2001: 683 – 689.

[2] Nicopoulos C, Vijaykrishnan N, et al. ViChaR: a dynamic virtual channel regulator for network-on-chip router [C]// Proceedings of the IEEE/ACM International Symposium on Microarchitecture (MICRO), 2006: 333 – 344.

[3] Xuning C, Peh L. Leakage power modeling and optimization in interconnection networks [C] // Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED), 2003: 90 – 95.

[4] Felperin S, Raghavan P, et al. A theory of wormhole routing in parallel computers [J]. IEEE Transactions on Computer, 1996, 45(6): 704 – 713.

[5] Dally W. Virtual-channel flow control[J]. IEEE Transactions on Parallel and Distributed Systems, 1992, 3(2): 194 – 205.

[6] Tamir Y, Frazier G. High-performance multi-queue buffers for VLSI communication switches [C]//Proceedings of the International Symposium on Computer Architecture (ISCA), 1988: 343 – 354.

[7] Lai M, Wang Z, Gao L, et al. A dynamically-allocated virtual channel architecture with congestion awareness for on-chip routers [C]//Proceedings of Design Automation Conference (DAC), 2008: 630 – 633.

[8] 尹亚明,陈书明,孙书为,等. 一种面向应用的 NOC 缓冲区分配算法[J]. 国防科技大学学报,2009,31(5):44 – 49.
YIN Yaming, CHEN Shuming, SUN Shuwei, et al. An application-specific buffer allocation algorithm for network-on-chip [J]. Journal of National University of Defense Technology, 2009, 31(5): 44 – 49. (in Chinese)

[9] Leung L F, et al. Optimal link scheduling on improving best-effort and guaranteed services performance in network-on-chip system [C]//Proceedings of Design Automation Conference (DAC), 2006: 833 – 838.

[10] Neishaburi M, et al. Reliability aware NoC router architecture using input channel buffer sharing [C] //Proceedings of GLSVLSI, 2009: 511 – 516.

[11] Liu J, et al. A shared self-compacting buffer for network-on-chip systems[C]//Proceedings of MWSCAS, 2006: 26 – 30.

[12] Shi W, Xu W, Ren H, et al. A novel shared-buffer router for network-on-chip based on hierarchical bit-line buffer[C]// Proceedings of International Conference on Computer Design (ICCD), 2011: 267 – 272.

[13] Mullins R, West A, Moore S, et al. Low-latency virtual channel routers for on-chip networks [C]//Proceedings of International Symposium on Computer Architecture (ISCA), 2004: 188 – 197.

[14] Karandikar A, Parhi K. Low power SRAM design using hierarchical divided bit-line approach [C]//Proceedings of International Conference on Computer Design (ICCD), 1998: 82 – 88.