

一种基于层次位线缓冲的异步片上路由器*

石伟,郭御风,王蕾,龚锐,窦强
(国防科技大学 计算机学院,湖南 长沙 410073)

摘要:片上缓存资源是片上路由器的重要组成部分,其结构好坏直接关系到片上互联网络的实现复杂度、整体性能及功耗开销。鉴于异步电路的握手工作方式,异步路由器一般采用基于移位寄存器的异步 FIFO (First In First Out)实现片上缓冲,这种结构导致了报文传输延迟及数据翻转次数增加。提出一种基于层次位线缓冲的异步 FIFO 结构,设计实现了一种新的异步路由器结构。相对于传统异步路由器,新的异步路由器能够有效降低路由器设计的硬件复杂度,减少数据的冗余翻转,降低功耗。实验结果表明在相同配置的情况下,新异步路由器面积降低了 39.3%;当异步 FIFO 深度为 8 的时候,新异步路由器能够获得 41.1% 的功耗降低。

关键词:异步电路;片上网络;低功耗;层次位线缓冲
中图分类号:TP302 **文献标志码:**A **文章编号:**1001-2486(2012)06-0007-07

A novel asynchronous on-chip router based on hierarchical bit-line buffer

SHI Wei, GUO Yufeng, WANG Lei, GONG Rui, DOU Qiang

(College of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: Buffer resources are key components of the on-chip router, and their structures exert significant influence on the performance and power consumption of the interconnection network. In general, asynchronous FIFO based on shift registers is adopted to implement on-chip buffer resources. Packets transmitted traverse the FIFO queue step by step, leading to higher propagation delay of packets and larger transition counts in the circuit. In this research, an asynchronous FIFO based on hierarchical bit-line buffer is proposed, and then a new asynchronous on-chip router is presented in detail. Compared with the traditional asynchronous router, the newly presented one has lower hardware complexity and power consumption. Experiments show that the new router can achieve 39.3% area saving and 41.1% power reduction when the depth of asynchronous FIFO is configured with 8.

Key words: asynchronous circuit; network-on-chip; low power; hierarchical bit-line buffer

随着片上网络(Network-on-Chip, NoC)^[1]的兴起,传统互联网络面临的带宽、可扩展性等问题得到了一定的缓解,但是其面临时钟网络分布及功耗等问题仍然存在^[2]。而异步 NoC 能够有效解决同步 NoC 的功耗及时钟分布问题,并且能够获得传统 GALS (Globally Asynchronous Locally Synchronous)技术的优点^[3],因此异步 NoC 逐渐得到了广泛关注。异步 NoC 可以看作由异步路由器与异步互连线组成,其中异步互连线与传统的点对点异步互连线结构相似。

在同步路由器中,虚通道(Virtual Channel, VC)的缓冲存储单元可以采用 SRAM^[4]或 D 触发器实现^[2],而异步路由器均是采用基于移位寄存器的异步 FIFO 实现片上缓冲^[5],其主要原因是

移位寄存器逐级传递的工作方式与异步电路的握手工作方式能够达到很好的统一。基于移位寄存器的异步 FIFO 在简化异步路由器设计的同时,也带来了一定的问题。在基于移位寄存器的异步 FIFO 中,数据进入缓冲以后,需要逐级向前传递,直至到达输出端口。数据每向前传递一级,报文的传输延迟都将相应增加;同时,电路中出现了一些冗余信号翻转,增加了电路的功耗。在基于 SRAM 的 FIFO 结构中,数据不需要在电路中逐级移动,但是其结构特点与异步电路的握手工作方式不符。

为了解决传统异步路由器中异步 FIFO 的长传输延迟与高功耗冗余的问题,本文基于层次位线缓冲结构^[6]提出了一种新的异步路由器结构。

* 收稿日期:2012-09-06
基金项目:国家“核高基”重大专项资助项目(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

在新的异步路由器中,报文控制信息存储在移位寄存器 FIFO 中,而报文其他内容存储在层次位线缓冲中。采用这种混合存储方式的好处包括:1)只有少量的数据需要在移位寄存器 FIFO 中逐级传递,减少了数据翻转量,降低了功耗;2)异步握手电路的局部时钟控制器只需要驱动少数几个寄存器,降低了驱动负载,提高了传输速度;3)由于基于移位寄存器的异步 FIFO 的数据通路宽度得到有效的降低,路由器中交叉开关的数据宽度得到有效降低;4)采用层次位线结构的缓冲具有较大存储容量,且功耗更低。

1 相关工作

1.1 传统异步路由器结构

文献[7]提出的异步交叉开关结构 CHAIN 可以看成一种简单的片上异步路由器,其传输链路采用 1-of-4 的编码方式。文献[8]提出的异步交叉开关 Nexus 同样可以看作一种简单的异步片上路由器,采用 QDI(Quasi Delay Insensitive)实现。随后,虚通道等技术被逐渐引入到片上路由器设计中,出现了 QoS 路由器^[9]、FAUST 异步路由器^[10]、MANGO 路由器^[3]及 QNoC 路由器^[5,11]等。

QoS 路由器是曼彻斯特大学提出的一种支持 QoS 的异步路由器结构,该路由器采用 5 个输入端口与 5 个输出端口,支持 GS(Guaranteed Service)与 BE(Best Effort)两个服务级。每个服务级对应一个虚通道。FAUST 异步路由器同时支持 RT(Real Time)与 BE 两个服务级。FAUST 采用 1-of-4 编码的 QDI 异步逻辑实现,能够获得一定的功耗降低及性能提高。

MANGO 路由器所支持的服务级数不再局限于两级,它可以支持任意多的服务。MANGO 路由器由一个 GS 路由器模块与一个 BE 路由器模块组成,GS 路由器模块与 BE 路由器模块共享输入输出端口及输出链路。GS 路由器模块为 GS 报文服务,而 BE 路由器模块为 BE 报文服务。MANGO 中的虚通道为 GS 报文提供面向连接服务,而为 BE 报文提供无连接服务。由于 MANGO 路由器提供面向连接服务,降低了传输链路的使用率,影响了片上网络的整体性能。

针对 MANGO 中的资源利用率低问题,文献[11]提出了一款异步路由器 QNoC,达到了相应同步实现的性能。在前面的异步路由器中,异步路由器基本上都引入了服务级的概念。引入“服务级”概念一方面能够达到为某些特殊报文提供性能保证的目的,另一方面也简化了硬件的设计

难度。在上述异步路由器中,每个服务级中只有一个 VC,每个 VC 只为一个服务级使用,因此降低了 VC 分配和交叉开关分配的难度;但是其 VC 独占的特征也降低了 VC 的使用效率。随后, QNoC 的设计者在文献[11]的基础上提出了一种新的 QNoC 路由器^[5]。在新的 QNoC 路由器中,每个 VC 可以被同一服务级中多个报文共享使用,提高了资源利用率及网络整体性能。其他异步路由器还包括 ASPIN^[12]及 WS^[13]等。总之,这些异步路由中异步 FIFO 都是基于移位寄存器,存在报文传输延迟大与数据翻转次数多的问题。

1.2 层次位线缓冲

我们提出了一种基于层次位线的片上统一缓冲结构(Unified Buffer Structure, UBS)^[6],并基于该缓冲结构设计实现了一款同步片上路由器。该缓冲结构能够有效解决传统的多端口存储体存在的问题。基于层次位线的片上缓冲的电路级实现如图 1 所示。在层次位线缓冲结构中,一组存储

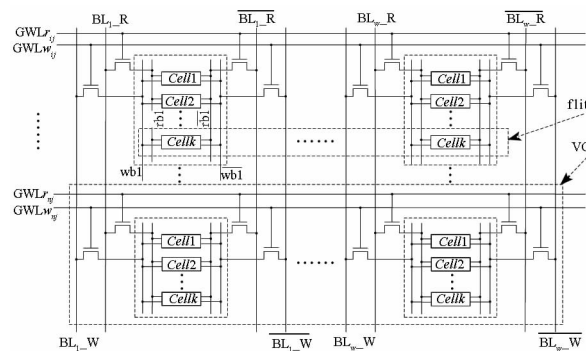


图 1 采用层次位线的片上路由器缓冲结构

Fig. 1 Structure of hierarchical bit-line buffer

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。为了简单起见,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 VC 的结构组成。若全局字线 $GWLw_{ij}$ 为高,写子位线则连接到高层写位线上,表明虚通道 i 被分配给输入端口 j ; 否则该 VC 处于空闲状态。因此,在某一时刻,全局字线 $GWLw_{i1}, \dots, GWLw_{ip}$ 最多只有一个为 1。若全局字线 $GWLr_{ij}$ 为高,读子位线则连接到高层读位线上,表明虚通道 i 连接给输出端口 j 。因此,在某一时刻,全局字线 $GWLr_{i1}, \dots, GWLr_{ip}$ 也最多只有一个为 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,可以将 VC 调整信号与读写字线进行逻辑“与”操作后再连接到全局位线上。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访问的目的。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;另外,可以通过单端口访问技术^[14]来减少缓冲的读写位线数目。

2 基于层次位线缓冲的异步路由器

由于传统的异步路由器结构存在缓冲传输延迟大、交叉开关复杂度大等问题,本节在文献[5]提出的异步路由器基础上,将层次位线缓冲结构引入到异步路由器设计中,提出了一种新的异步路由器结构(Asynchronous Hierarchical Bit-line Buffer Router, AHIBB)。AHIBB 路由器的整体结构如图 2 所示。AHIBB 由 5 个输入通道(Input Channel)与 5 个输出通道(Output Channel)组成,输入通道包含 v 个输入 VC(IP-VC),输出通道主

要包括 VA(Virtual Channel Allocator)模块、输出 VC(OP-VC)及 VC 仲裁器。当报文微片到达输入虚通道端口时,报文微片中的路由信息、报文类型信息进入基于移位寄存器的异步 FIFO,该部分信息用于后续 VA 等步骤中;报文微片的剩余部分进入 UBS。当 VA 模块在输入 VC 与 OP-VC 之间建立一条固定通信链路以后,存储在基于移位寄存器的异步 FIFO 中数据通过建立的通信链路申请输出虚通道端口。VC 仲裁模块对申请的报文进行仲裁,获胜的报文获得输出通道端口使用权,同时 VC 仲裁器还要从 UBS 中读出获胜报文微片的剩余部分,重新组合成原来的报文微片。因此,路由器只对路由信息、报文类型等部分控制信息进行逐级传输,以达到数据转发的功能;而对其他信息只需存取一次。AHIBB 路由器的上述工作方式有效解决了传统异步路由器面临的功耗、延迟等问题。

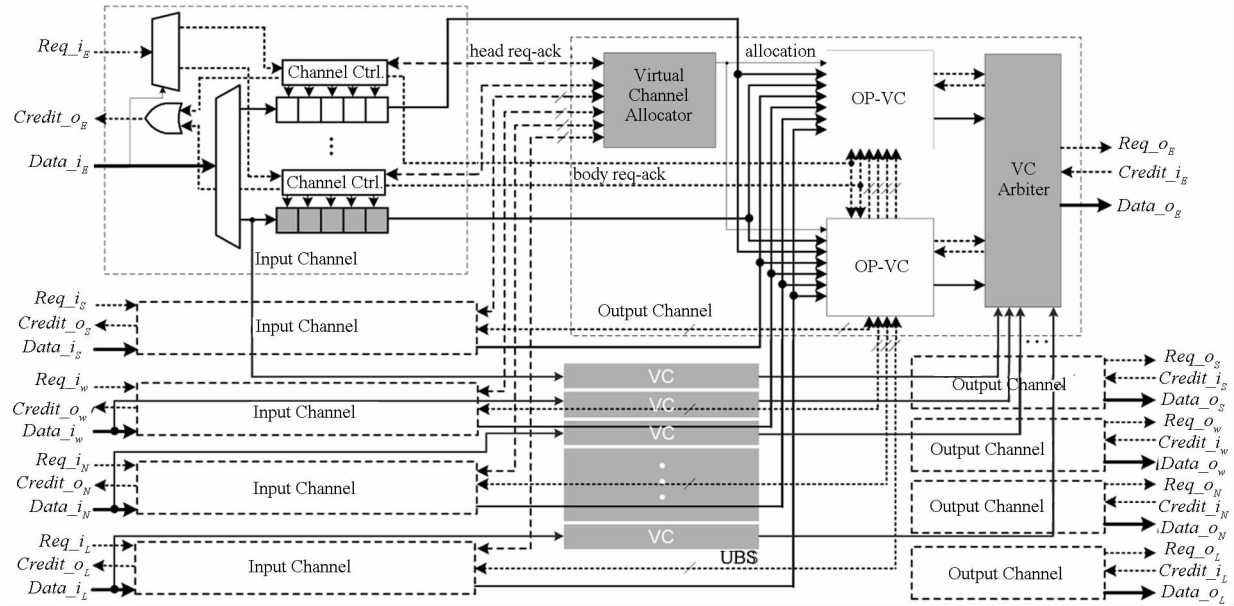


图 2 AHIBB 路由器基本结构

Fig. 2 Structure of asynchronous router with hierarchical bit-line buffer

3 异步路由器微体系结构

3.1 输入通道结构

本文的输入通道结构与文献[5]中输入通道结构相似,但是去除了服务级的概念,传输报文可以使用输入通道中任何 VC,而非一个特定的具有相同服务级的 VC。图 3 给出了一个 $v = 4$ 时的输入通道结构,当一个新的微片到达输入端口后,输入请求信号 REQ_IN 根据微片中的 VC 号被发送到对应的虚通道 IP-VC。被选中的 IP-VC 与输入通道的请求应答信号(REQ_IN 与 ACK_IN)进行

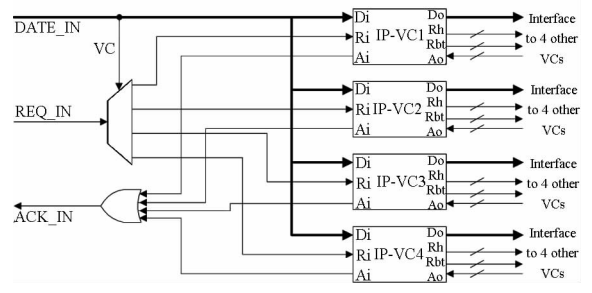


图 3 输入通道结构

Fig. 3 Structure of input channel

握手,并锁存数据。当 IP-VC 将新来的报文微片锁存后,一方面向前一级发出应答信号以通知其

进行下一次数据传输;另一方面根据该报文的路由地址信息向相应的输出通道发出传输请求。

在传统的异步虚通道结构中,虚通道 IP-VC 主要由异步 FIFO 模块与请求模块两部分组成。其中,异步 FIFO 模块采用微流水线方式实现。在以微流水线方式实现的异步 FIFO 中,数据需要从输入端向后逐级传输才能到达 FIFO 的输出端。深度较大的异步 FIFO 能够暂存较多的报文微片,但却增加了单个报文微片的传输延迟。另一方面,数据从输入端口逐级向输出端口传递的特性势必会增加信号翻转的次数。本文的异步虚通道结构如图 4 所示。

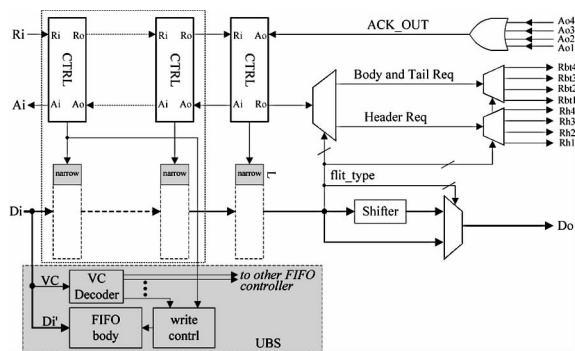


图 4 AHIBB 异步虚通道结构

Fig. 4 Asynchronous virtual channel of AHIBB

我们通过以下几种方式来降低数据传输延迟与数据传输带来的功耗开销。首先,保留原有基于移位寄存器的异步 FIFO 的基本结构,但是基于移位寄存器的异步 FIFO 中只存储少数报文控制信息,如报文微片类型,报文路由信息等,而其他报文数据存储到 UBS 中去。其次,为了尽量减小移位寄存器对传输延迟的影响,我们采用高性能的细粒度异步流水线对移位寄存器 FIFO 进行重新设计。采用这种存储方式的好处包括:1) 只有少量的数据需要在移位寄存器中逐级传递,减少了数据翻转量;2) 局部控制器只需要驱动少数几个寄存器,降低了驱动负载;3) 保留的原有异步 FIFO 基本结构使得 AHIBB 继承了传统异步路由器的优点;4) 基于移位寄存器的异步 FIFO 的数据通路宽度得到有效的降低,AHIBB 中交叉开关的数据宽度得到控制;5) 采用层次位线结构的 UBS 具有较大存储容量,同时具有相对移位寄存器更低的功耗。

当报文的头微片到达 FIFO 输出端口 L 后,微片中的路由信息用于控制 MUX 模块,进而将传输请求 Rb_i 发送到正确的输出通道。同时,还需要对头微片中的路由信息进行移位操作。当报文的体微片和尾微片到达 FIFO 输出端口后,它

们不需要进行移位操作,并将传输请求信号 Rbt_i 广播给所有的输出通道。

3.2 输出通道结构

AHIBB 路由器输出通道结构如图 5 所示,主要由 VA、输出虚通道 OP-VC 与虚通道仲裁模块组成。VA 模块接收来自所有输入通道的报文头微片传输请求,然后对这些请求进行仲裁,将空闲的 OP-VC 分配给获胜的请求报文。3.3 节将对 VA 模块进行详细说明。当 VA 将 OP-VC 分配给某一输入端口虚通道中的报文后,VA 采用通过设置 VC_INDEX 信号建立一条输入虚通道到输出虚通道之间的固定链路。此后,该输入虚通道中报文微片的传输请求信号只能发送到对应的输出 OP-VC。输出虚通道 OP-VC 的数目与相邻输入端口中的虚通道数目相同,且成一一一对应关系。因此,VA 对 OP-VC 的控制实际上就是对下一级输入端口虚通道的控制。OP-VC 接收输入端口中的报文微片,并将其传输到下一级路由节点中去。3.4 节将对 OP-VC 的微结构进行详细介绍。由于多个 OP-VC 可能同时对传输链路发出使用请求,因此需要使用 VC 仲裁器对这些请求进行仲裁,获胜的 OP-VC 才能够使用传输链路。3.5 节将对 VC 仲裁器结构进行进一步描述。

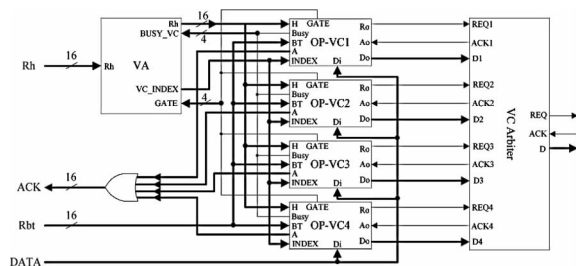


图 5 AHIBB 路由器输出通道结构

Fig. 5 Output channel of AHIBB router

3.3 VA 模块结构

VA 负责对多个异步 VC 请求信号进行仲裁,然后在获胜的 IP-VC 与新分配的 OP-VC 之间建立一条固定的传输链路。AHIBB 的异步 VA 模块的微结构如图 6 所示。图 6 中的 MUTEX-NET 仲裁模块^[5]接收来自所有输入通道中的报文头微片请求,然后进行仲裁。如果某一请求获胜,则图中静态优先级仲裁器 SPA 的使能信号 EN 有效,否则 EN 无效。SPA 对当前可用的 VC 进行仲裁,获胜的 VC 将被分配给获胜的输入端口 VC_INX_VCi 信号用于在上述两个获胜者之间建立传输路径。由于 SPA 负责对 VC 进行仲裁,SPA 的路数 M 等于下一节点中的 VC 数目。图中 BUSY_VC

信号表示对应的 VC 是否空闲。图 6 中的 GATE_VCi 信号表示 VCi 已经成功分配给某一请求报文,因此 SPA 能够接收新的请求并进行 VC 分配。

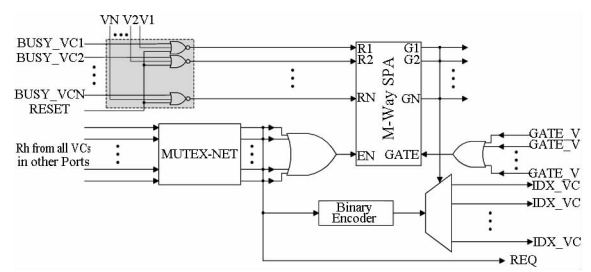


图 6 AHiBB 的虚通道分配模块
Fig. 6 VC allocator of AHiBB router

由于 AHiBB 路由器采用的 UBS 具有多端口共享 VC 的特点,因此 VA 模块中的候选 VC 数目为 N 个,其中 N 表示一个输入通道内所能包含的 VC 最大数目。对于一个 VC 来说,它能够被多个端口共享,则多个端口中都将存在这一 VC 的信息,如 VC 分配信息。如果不处理好上述问题,VC 可能被同时分配给多个报文使用,从而引起错误。为了避免 VC 被同时分配给相邻节点中的多个输入报文使用,我们在 VA 模块中增加了 VC 有效信号 V_i 。当 V_i 为 0 时,说明该 VCi 被分配给本输出通道使用;否则,VCi 不可用。 V_i 信号的具体值可以根据文献[6]介绍的 VC 调整策略来确定。

3.4 输出虚通道结构

输出虚通道 VC-OP 用于连接输入通道中的 IP-VC 与相邻路由节点中的 VC,其结构如图 7 所示[5]。当仲裁逻辑将输出 VC 分配给某一输入 VC, VA 将获胜的输入 VC 编号 INX 发送给相应

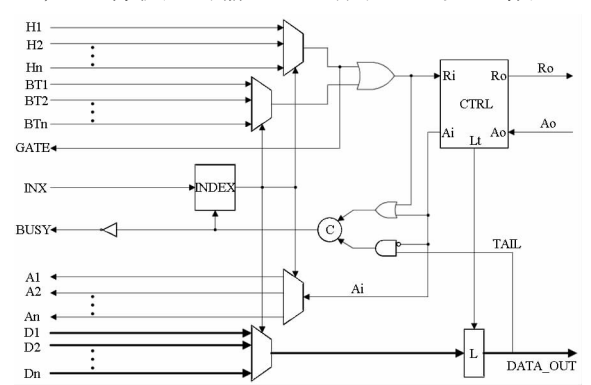


图 7 输出虚通道结构
Fig. 7 Structure of output VC

的输出 VC。INX 信号控制 MUX 模块,使输入 VC 的传输请求能够传输到局部 Latch 锁存控制器 CTRL。当报文头微片请求通过 MUX 后,VC-OP 立即向 VA 模块发出 GATE 信号,GATE 信号通知

VA 可以接收新的 VC 请求并进行 VC 分配。另外,VC-OP 将 BUSY 信号置为高,至此,VC 被分配给某一特定的输入 VC,该 VC 将不会再被分配给其他 VC。由于报文头微片请求建立了一条固定的传输链路,其后的体微片与尾微片能够通过锁存器 L 直接传输到输出端口 DATA_OUT 上。当报文尾微片通过 VC-OP 后,Latch 锁存器 INDEX 打开,INDEX 的输出信号变为 0,关闭所有 MUX 模块。同时,BUSY 信号被置低,表示 VA 模块可以将该输出 VC 重新分配给输入 VC 使用。

3.5 VC 仲裁器结构

当数据被锁存到 VC-OP 中的锁存器 L 后,报文微片将请求使用传输链路。由于多个 VC-OP 可能同时请求使用传输链路,虚通道仲裁器负责对多个 VC-OP 进行仲裁。虚通道仲裁器结构如图 8 所示。首先,请求输出端口的多个请求信号 R_i 通过 MUX-Net 进行仲裁,获胜的请求控制 MUX 模块进行数据传输。当报文微片传输到输出链路上以后,本地锁存器 L 可以供下一次数据传输使用。由于获胜报文微片的一部分内容存储在 UBS 中,因此 VC 仲裁器需要向 UBS 发出读请求,然后将读出来的数据与数据 D_i 合并形成最终的输出报文微片。

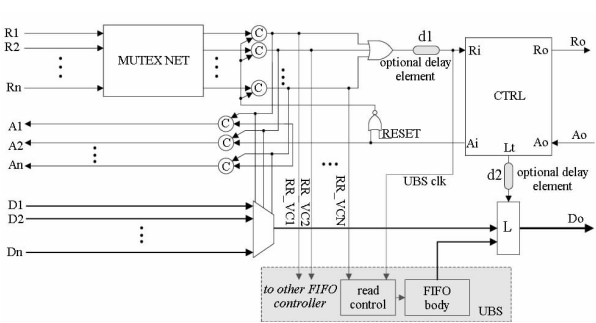


图 8 AHiBB 的 VC 仲裁模块结构
Fig. 8 Arbiter structure of AHiBB

图中 MUX-Net 模块首先对请求输出链路的 N 个请求进行仲裁,获胜的请求在向局部控制器 CTRL 发出请求的同时,立即向对应 VC 的虚拟局部控制器发出读请求。虚拟控制器接收到读请求以后,立即向对应 VC 的读控制器发出请求,读控制器负责将 FIFO 头部的数据读出。在 CTRL 向锁存器 L 发出锁存信号时,由移位寄存器方向来的数据 D_i 与由 UBS 方向来的数据都必须都已经有效,因此 VC 仲裁器中的信号必须满足两个条件:1)在 UBS 时钟信号有效之前,UBS 的读地址等信息必须已经准备就绪;2)在 L_t 锁存报文微片数据时,UBS 已经读出有效数据。为了满

足上述两个条件,我们可以在 VC 仲裁器中增加两个可选的延迟单元,延迟单元的大小根据相关电路实际延迟确定。延迟单元 d1 延迟了 UBS 局部时钟信号的产生,其大小根据虚拟控制器及物理控制器的延迟确定。延迟单元 d2 用于等待 UBS 输出结果有效,其大小根据 UBS 实际读出延迟与 CTRL 的延迟确定。

4 异步路由器实现及评测

本节在 90nm 工艺下对 AHiBB 路由器进行 VLSI 实现并评测。为了更好地比较,我们还同时设计实现了一款异步路由器 AGR (Asynchronous Generic Router),AGR 与 AHiBB 的整体结构基本相同,不同之处在于 AGR 采用传统的移位寄存器 FIFO 实现输入端口虚通道。

4.1 AHiBB 实现及评测

本节对 AGR 路由器与 AHiBB 路由器进行了 VLSI 实现。为了简化电路设计实现的复杂度,AGR 路由器与 AHiBB 路由器的每个输入端口均包含 4 个 VC。

AGR 路由器与 AHiBB 路由器面积比较如表 1 所示。在 AGR 路由器中,缓冲采用移位寄存器 FIFO 实现,导致了其面积大幅上升。但是,由于移位寄存器异步 FIFO 的异步信号能够表征缓冲的满/空情况,因此其控制逻辑得到了一定的简化。在输出 VC 与 VC 仲裁器中,存在一级移位寄存器,其面积均达到了 $20 \times 10^3 \mu\text{m}^2$ 左右。另外,由于每个输入 VC 需要连接到每个输出 VC 端口,交叉开关的复杂度大幅增加。

表 1 AGR 路由器与 AHiBB 路由器面积比较 ($\times 10^3 \mu\text{m}^2$)

Tab. 1 Area comparison of AGR and AHiBB		
模块	AGR 路由器	AHiBB 路由器
缓冲	129.6	74.1
VC 控制模块	3.5	11.3
VC 分配模块	6.7	6.7
输出 VC	16.5	4.7
交叉开关	22.7	3.1
VC 仲裁器	18.5	19.9
总量	197.5	119.8

在 AHiBB 路由器中,我们采用 UBS 替代一部分缓冲,减少了缓冲的面积。由于移位寄存器 FIFO 数据宽度得到了大幅降低,输出 VC 与交叉开关的面积也得到了较大的降低。但是,由于

UBS 的引入,VC 控制器的面积增加了 $7.8 \times 10^3 \mu\text{m}^2$ 。另外,VC 仲裁器需要根据获胜报文的信息读出 UBS 中相应的报文微片,读控制器使得其面积稍微增加。总之,AHiBB 路由器较 AGR 路由器面积降低了 39.3% 左右。

4.2 性能及功耗分析

为了对 AGR 与 AHiBB 路由器进行定量比较,我们对在 VC 不同深度情况下报文微片经过两种路由器的延迟及功耗进行测量。图 9 为一个报文微片经过两种路由器所需要的延迟。由于 AHiBB 的移位寄存器数据通路宽度较窄,控制通路驱动数据通路的局部时钟的负载也较小,因此 AHiBB 中异步 FIFO 的速度要稍快于 AGR 中的异步 FIFO。然而,AHiBB 中引入了基于 SRAM 的 FIFO 结构,读写这种 FIFO 的时间要稍大于普通的移位寄存器 FIFO。在 FIFO 深度较小时,AHiBB 中读写 SRAM 的时间占主导地位,其报文微片的传输延迟也较 AGR 大些。随着 FIFO 深度的加大,数据通路宽度降低导致的延迟减小优势逐渐明显,AHiBB 中报文微片传输延迟越来越接近 AGR 路由器。对 AGR 与 AHiBB 路由器的异步流水线进行分析,其延迟最大的流水段为 VC 仲裁级。AGR 的 VC 仲裁级延迟约为 2.2ns;而 AHiBB 的 VC 仲裁级延迟约为 2.5ns,其主要原因是该级增加了 SRAM 读。基于上述分析,AGR 输出端口的吞吐量可以达到 450Mflit/s,而 AHiBB 路由器输出端口的吞吐量可以达到 400Mflit/s。

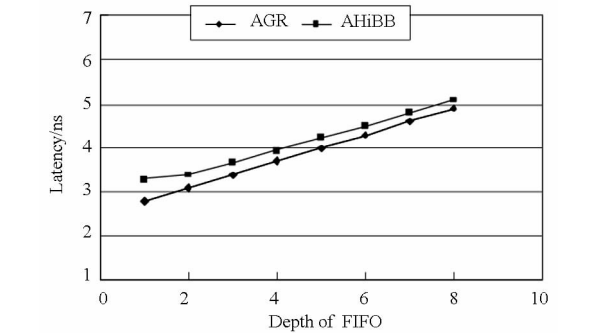


图 9 AGR 与 AHiBB 路由器的性能比较
Fig. 9 Performance comparison between AGR and AHiBB

图 10 为 AHiBB 与 AGR 路由器在只有一个端口活动情况下所消耗功耗的比较情况。在 FIFO 深度较小的情况下,AGR 的功耗较小,其主要原因是 AHiBB 读写 SRAM 的功耗较大。随着 FIFO 深度的增加,AGR 功耗增长迅速,很快就达到并超过 AHiBB 路由器。在 AHiBB 中,报文微片被存储在 SRAM 中直至被输出端口读出;而

AGR 中的报文微片需要在移位寄存器异步 FIFO 中逐级移动,增加了路由器的功耗。

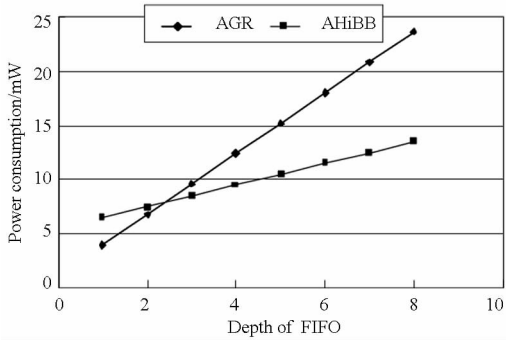


图 10 AGR 与 AHiBB 路由器的功耗比较
Fig. 10 Power comparison between AGR and AHiBB

5 结论

基于层次位线缓冲设计实现了一款异步片上路由器 AHiBB。层次位线缓冲的结构与传统的异步路由器缓冲结构能够达到很好的统一,从而可以采用层次位线缓冲与移位寄存器缓冲相结合的方式来实现异步片上路由器的缓冲。采用这种缓冲结构能够有效降低路由器设计的硬件复杂度,AHiBB 的面积较传统异步路由器降低了 39.3%。当 FIFO 深度达到 8 时,AHiBB 能够获得 41.1% 左右的功耗降低。

参考文献 (References)

[1] Dally W, Towles B. Route packets, not wires: on-chip interconnection networks [C]// Proceedings of Design Automation Conference (DAC), 2001: 684 – 689.
[2] Nicopoulos C, et al. ViChar: a dynamic virtual channel regulator for network-on-chip router [C] //Proceedings of International Symposium on Microarchitecture (MICRO), 2006: 333 – 344.

[3] Bjerregaard T, et al. An OCP compliant network adapter for GALS-based SoC design using the MANGO network-on-chip [C]//Proceedings of Symposium on System-on-Chip (SoC), 2005: 171 – 174.
[4] Xu Y, et al. Simple virtual channel allocation for high throughput and high frequency on-chip routers [C]// Proceedings of International Symposium on High Performance Computer Architecture (HPCA), 2010: 1 – 11.
[5] Dobkin R, Ginosar R, Kolodny A. QNoC asynchronous router [J]. Integration, 2009, 42 (2) : 103 – 115.
[6] 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.
[7] Bainbridge W J, Furber S B. Chain: a delay-insensitive chip area interconnect [J]. IEEE Micro, 2002, 22 (5) : 16 – 23.
[8] Lines A. Nexus: an asynchronous crossbar interconnect for synchronous system-on-chip designs [C]//Proceedings of Symposium on High Performance Interconnects, 2003: 2 – 9.
[9] Felicijan T, Furber S B. An asynchronous on-chip network router with Quality-of-Service (QoS) Support [C] // Proceedings of International SoC Conference, 2004: 274 – 277.
[10] Pascal V. Design of on-chip and off-chip interface for a GLAS NoC architecture [C]// Proceedings of IEEE Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC), 2006: 172 – 181.
[11] Dobkin R, et al. An asynchronous router for multiple service levels networks on chip [C]// Proceedings of IEEE Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC), 2005: 44 – 53.
[12] Sheibanyrad A. Asynchronous implementation of a distributed network-on-chip [D]. Paris: University Pierre et Marie Curie, 2008.
[13] Song W, Edwards D. A low latency wormhole router for asynchronous on-chip networks [C]// Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC), 2010: 437-443.
[14] Singh J, Pradhan D K, Hollis S, et al. Single ended 6T sram with isolated read-port for low-power embedded systems [C]//Proceedings of Conference on Design, Automation and Test in Europe (DATE), 2009: 917 – 922.