

一种二维片上网络路由器的设计实现

蔡升

(上海大学 微电子研究与开发中心, 上海 200444)

摘要: 片上网络 (Network on Chip, NoC) 作为解决众核互连的主流方案, 其性能很大程度上取决于网络的拓扑结构; 而网络拓扑结构的效能受到网络路由器的直接影响; 因此, 基于特定拓扑结构的路由器设计实现具有非常重要的研究意义; 对此将 XY 路由算法应用于路由器节点中, 设计了基于 2D Mesh 拓扑结构、轮询仲裁机制与虫孔交换流控的片上网络路由器, 并使用 Modelsim 对路由器进行了功能验证; 实验结果表明, 设计的路由器能满足微片数据的处理, 能够正确的收发数据包。

关键词: 片上网络; 路由算法; 路由器

Design and Implementation of a Two-Dimensional Network-on-Chip (NoC) Router

Cai Sheng

(Microelectronics R&D Center, Shanghai University, Shanghai 200444, China)

Abstract: Network on Chip (NoC) is the mainstream solution for many core interconnection. Its performance depends largely on the topology of the network. And network router directly affects the efficiency of topology. Therefore, based on specific topology, router design and implementation has great research significance. Applied XY routing algorithm to the routing node, and designed the router based on 2D Mesh topology, Round-Robin arbitration mechanism and wormhole exchange flow control. The experimental results carried on Modelsim show that the router can transfer the micro-chip data rightly and can correctly send the data packet.

Keywords: network on chip; routing algorithm; router

0 引言

随着半导体工艺的发展与集成度的提高, 单芯片上的晶体管数量越来越多^[1-2], 芯片的规模与日俱增。为了满足多媒体、网络通信与大数据计算的需求, 目前多核片上系统 (MPSoC) 普遍采用总线互连的方式集成数个处理器核与功能 IP 核, 并向着众核片上系统发展。片上网络技术采用包交换与分层的方式代替传统的总线, 实现了 IP 核与互连层之间的分离, 相对总线互连具有拓展性高、可靠性高与功耗低的特点, 已经成为众核互连的主流解决方案^[3-6]。片上网络互连层主要由通信链路, 路由器节点与网络接口 3 个部分组成。拓扑结构与路由器效能直接决定网络的整体性能。因此, 针对片上网络互连层的研究, 特别是基于特定拓扑结构的路由器硬件设计, 是一种提升 NoC 性能的有效途径^[7-8]。

1 NoC 体系结构

NoC 的互连层与分布式计算机系统的互连网络有相似之处, 如支持包交换、拥有路由协议与高可拓展性等。由于 NoC 内部的 IP 核通过数字电路工艺互连, 与分布式计算机采用的物理线路互连差别较大, 这让 NoC 发展出了独有的拓扑结构, 路由算法、数据格式与流控策略。

收稿日期: 2019-03-08; 修回日期: 2019-04-09。

作者简介: 蔡升 (1994-), 男, 浙江台州人, 硕士研究生, 主要数字集成电路技术方向的研究。

1.1 NoC 拓扑结构

拓扑结构是片上网络互连层研究的一个重要领域, 拓扑结构的类型会直接对网络整体性能与硬件实现造成显著的影响。例如, 网络拓扑决定片上网络路由器节点与通信链路的空间分布和互连布局, 会影响到数据微片的路由路径距离和每跳路由的线路长度。拓扑结构也决定源节点与目的节点之间的路由路径跳数与链路资源数量, 会对数据流量的分布与路由器的负载分布造成直接影响, 与网络吞吐率等性能指标关系紧密。

NoC 拓扑结构主要分为规则拓扑结构与不规则拓扑结构。常见的拓扑结构有 2D Mesh 结构、2D Tours 结构与 Octagon 结构等。由于 2D Mesh 拓扑结构拥有良好的拓展性、结构规整、容易实现高效的路由计算与仲裁, 因此本文基于 2D Mesh 拓扑进行路由器的设计研究。基于 2D Mesh 的 NoC 如图 1 所示, 其路由器节点拥有 5 组输入输出端口, 4 组端口用于与相邻路由器的连接, 本地端口则通过网络接口与 IP 核相连。

1.2 NoC 路由算法

路由算法确定数据包下一跳路由所需要的输出端口, 能够将数据从源节点准确的传输到目的节点, 是影响片上网络性能的重要因素。NoC 路由算法的电路实现不能太复杂, 因为复杂的电路会增加关键路径的延时, 占用更多的硬件面积, 并增加功耗。路由算法按照是否具有适应性可分为自适应路由算法与确定性路由算法。

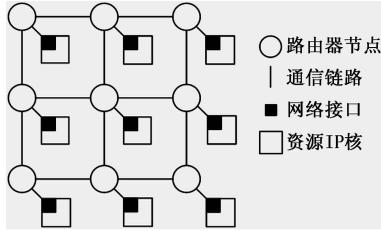


图 1 Mesh 拓扑结构图

自适应路由算法是研究的热点，其规划的路径不仅与源节点和目的节点的相对位置有关，还受到网络的状态，如温度^[9-10]、功耗^[11]、缓存空间与路由器负载的影响。在已知源节点与目的节点位置的情况下，自适应路由算法会实时感知网络状态，规划最优路径传输数据。此类算法的优点是能够较好的避免数据拥塞，获得更高的片内通信吞吐率。其缺点是路由逻辑复杂，需要特殊的硬件结构，硬件实现的难度巨大，而且存在死锁与感知滞后问题。

确定性路由算法实现简单，在网络低负载条件下能够获得较小的延时，且不会产生死锁，能够满足大部分数据传输任务的需求，所以本文采用 XY 确定性路由算法。该算法首先在 X 维度上路由，当数据包与达目的节点处于同一列时，再进行 Y 维度上的路由，从而达到目的节点。

1.3 NoC 数据格式

数据包的最小组成单位是微片，其分为头微片、体微片与尾微片。对于本文设计的路由器，规定微片位宽为 32bit，具体格式如表 1 所示。位于 [33: 32] 的数值用于区分微片种类 (00 为头微片, 01 为体微片, 11 为尾微片)。头微片携带着源节点地址 (S_y, S_x)、目的节点地址 (D_y, D_x) 和有效数据长度等附加信息。地址每个维度位宽为 4，最多支持 256 个节点。体微片和尾微片不携带路由信息，它们沿着头微片建立的路径传输到目的节点。

1.4 NoC 流控机制

片上网络主要的流控机制有电路交换技术，储存转发交换技术，虚直通交换技术与虫孔交换技术。电路交换技术、储存转发交换技术与虚直通交换技术均存在资源利用率不高、串行化延迟较大与硬件资源消耗较多等问题^[12]。因此，综合虑资源与功耗等方面的因素，本文选用虫孔交换技术作为路由器的流控机制。

虫孔交换技术以微片作为数据传输的最小粒度，并将数据包划分成多个微片 (flits)。头微片 (head flit) 存储路由信息，包括数据包的源节点、目的节点等控制信息。头微片后面的微片按照流水的顺序跟随头微片在传输路径中行进。某个链路上的微片传输完成后，该链路就会被解除占用。每一个路由器节点的仅需若干个微片大小的缓存空间，

当下一级路由器缓存尚有空间是就立刻发送微片。如果传输过程中网络发生了拥塞，数据包的后续微片会暂存在后面的各个路由器节点。当网络出现拥塞时，会有较多的路由器节点与链路被占用，采用虫孔交换方式在网络产生比较严重的拥塞时候还会产生死锁。后续路由节点的缓存单元只有一个微片的空间时，虫孔交换仍然能传送数据包。如图 2 所示，当路由器节点 0 与路由器节点 1 在网络不存在拥塞时，应用虫孔交换与虚直通的表现相同。但是，当网络出现拥塞时，两种流控机制会有不同的表现。如图 2 所示，尽管在时钟周期 2 路由器节点 2 的缓存空闲空间 (2 个微片的空间) 无法容纳整个数据包，路由器节点 2 仍然可以传输头微片与第一个数据微片。当路由器节点 1 和路由器节点 2 之间的链路在第 4 与第 5 个时钟周期内均处于空闲状态时，这条链路也无法被其它传输任务使用，直到第 7 个时钟周期时尾微片完成传输。因此，在虫孔交换机制下，数据包会以微片的组成形式保存在一系列的路由器节点中，但其传输路径中的链路通道会一直被该数据包传输任务使用。组成数据包的微片均不会丢失或受到其他数据包传输任务的干扰，数据包能够保持自身完整性。

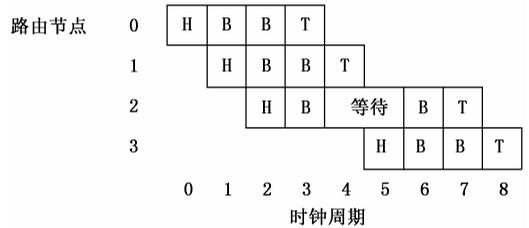


图 2 虫孔交换流控机制

2 路由器设计

确定片上网络互连层的主要技术特征，包括拓扑结构、路由算法后，就要对路由器硬件进行研究设计。路由器是片上网络互连层的主要部件，也是片上网络的关键部件，主要负责数据的暂存与转发。其主要模块有输入缓存模块、输出端口、路由计算模块、交叉开关模块与路由仲裁模块等，具体如图 3 所示。路由器输入输出端口用于连接相邻路由器与本地 IP 核，对于本文的 2D Mesh 拓扑结构，对应路由器拥有五个方向的输入输出端口，分别是本地 (Local, L)、东 (East, E)、西 (West, W)、南 (South, S) 与北 (North, N)。当数据微片进入路由器的缓存后，路由计算模块根据微片输入端口位置、当前路由器的位置信息、源节点与目的节点信息以及路由器状态，按照硬件固化的路由算法进行路径规划，建立路由器输入端口与输出端口的连接。

表 1 数据包格式

	[31:30]	[29:26]	[25:21]	[20:17]	[16:13]	[12:9]	[8:5]	[4:0]
头微片	Type:00	Reserved	Length	D_y	D_x	S_y	S_x	Dir_in
体微片	Type:01				Data			
尾微片	Type:11				Data			

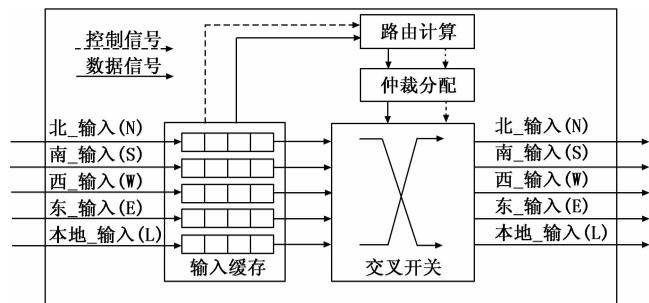


图 3 路由器结构图

2.1 输入缓存模块

输入缓存模块的作用是缓存相邻路由器以及本地计算 IP 核发送的数据。对于文中的 2D Mesh 片上网络，需要有 5 组缓存单元来满足北 (N)、南 (S)、东 (E)、西 (W) 与本地 (L) 方向输入微片的缓存。缓存单元的位宽等于微片的位宽，缓存单元的深度可以根据网络面向的应用与路由算法进行调整，其作用是将依次接受到的微片按需顺序输出。由于片上网络支持异步全局，局部同步的设置，也需要用缓存模块来保证片内互连层，即路由网络与资源 IP 核的跨时钟域数据传输。结合以上几点，每个数据通道的缓存单元均用先进先出 (first in first out) 的异步 FIFO 实现。图 4 是经典的异步 FIFO 硬件结构，它主要由一个双端口的 RAM、写控制模块、读控制模块与同步逻辑组成。双端口 RAM 主要用于数据微片的缓存，写控制模块和读控制模块主要作用是响应外部设备发出的读写请求、读写指针的产生和格雷码转换，并根据读写指针位置输出表示 FIFO 满或空的信号。由读控制模块和写控制模块输出的格雷码指针信号会进入同步逻辑，同步逻辑会将格雷码格式的读指针或者写指针同步到写模块和读模块用于产生空或者满信号。

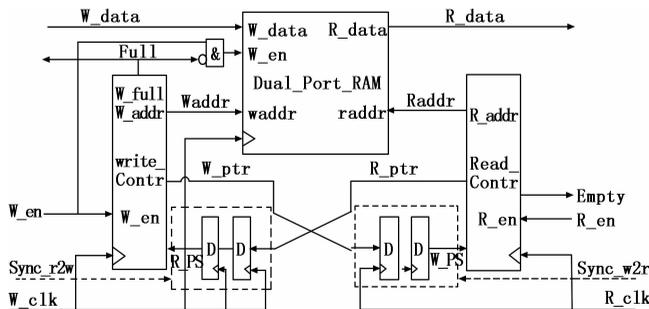


图 4 FIFO 结构图

2.2 路由计算模块

路由算法的实现载体是片上网络路由器的路由计算模块，算法会以数字电路的形式固化在路由计算模块中。路由计算模块会综合数据包头微片包含的信息与路由算法来计算微片的输出端口。本文的路由计算模块是 XY 路由算法的硬件实现。它的运算过程是：路由计算模块首先依据微片里的 Type 位信息判断该微片是否为头微片。如果是头微片，路由计算单元就会依据头微片包含的源节点、目的节

点与固化的路由算法计算出微片的输出端口，并向控制交叉开关的仲裁模块提交相应的申请信号。如果微片类型为体微片或者尾微片，则不进行路由计算，直接提交输出口使用申请。

对于本文应用的 XY 路由算法，首先需要解析头微片中包含的目的节点信息，随后与当前节点的位置信息进行比较。接着判断当前节点与目的节点的 X 坐标大小：当前节点 X 坐标小于目的节点时，向 X 正方向路由；当前节点 X 坐标大于目的节点时，向 X 负方向路由；当前节点 X 坐标等于目的节点时，接着判断当前节点与目的节点 Y 坐标的大小。当前节点 Y 坐标小于目的节点时，向 Y 正方向路由；当前节点 Y 坐标大于目的节点时，向 Y 负方向路由；当前节点 Y 坐标等于目的节点时，表明数据微片到达目的节点，停止路由。

2.3 路由仲裁模块

在片上网络运行的过程中，同时存在多对 IP 核进行数据的发送与接收，部分路由器节点会因此同时接收到多个头微片。路由计算单元读取头微片的附加信息，计算出结果后向路由仲裁单元申请交叉开关的使用权。路由仲裁单元批准某一输入端口的请求后，会向交叉开关发送对应的控制信号，建立固定的输入输出端口连接，并发送缓存读取信号读取微片数据，输入交叉开关对应的输入端，使微片从指定端口输出。

路由仲裁单元会在同一时刻收到某输出端口的多个使用请求，需要设计仲裁策略进行处理。仲裁策略一般分为固定优先级仲裁策略与轮询仲裁策略 (Round-Robin, RR)。固定优先级策略赋予每个输入通道固定的优先级，如输入通道的优先级从高至低依次为北 (N)、南 (S)、西 (W)、东 (E)、本地 (L)。一旦北 (N) 输入通道需要使用某输出端口时，不管剩余输入通道是否请求使用该端口，总是响应北 (N) 通道的请求。固定优先级策略的硬件实现较为简单，但不会均衡资源的分配。轮询仲裁策略 (Round-Robin, RR) 会对输入通道设置默认的优先级，在首轮仲裁中会对提出请求的通道

进行比较，批准最高优先级通道的请求。随后，该通道的优先级降至最低，优先级比该通道低的端口依次提升优先级，原理如图 5 所示。在图 5 中，路由器输入通道的默认优先级为北 (N)、南 (S)、西 (w)、东 (E) 与本地 (L)。五个通道会用相应的 Req_n、Req_s、Req_w、Req_e 与 Req_l 信号表示对输出端口的申请。在图 5 中，对于第一次仲裁，由于北 (N) 通道的优先级是最高的，即使北 (N)、南 (S)、西 (W)、东 (E) 与本地 (L) 通道均发起申请，仲裁的结果仍为批准北 (N) 方通道的请求。仲裁结束后，北 (N) 方通道的优先级降到最低，其余通道的优先级依次提升。对于第二次仲裁，只有南 (S)、西 (W)、东 (E) 与本地 (L) 方向输入通道的请求信号数据。根据第一次仲裁过后的优先级，仲裁分配单元批准南 (S) 方向通道的请求。仲裁器响应全部申请流程的波形如图 6

所示，可以看到仲裁器会按照优先级依次批准申请。

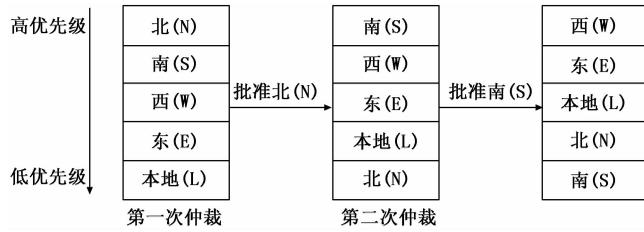


图 5 轮询仲裁示意图

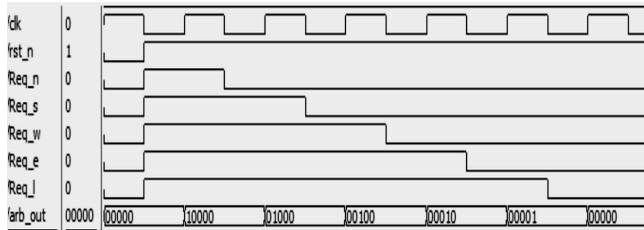


图 6 轮询仲裁波形图

2.4 交叉开关模块

交叉开关模块是数据路由器的核心结构，其作用是建立各输入通道缓存单元与输出端口的连接，使某端口的输入数据从指定端口输出。由于二维片上网络路由器有 5 组输入输出通道，交叉开关会有 5 组输入端口与 5 组输出端口的控制信号接口。二维片上网络路由器的交叉开关可以看做 5 个 5 输入多路选择器的组合。每个多路选择器的输入端对应 5 个通道的输入，输出端对应特定方向上的输出端口，其会按照控制信号输出指定输入通道的内容。交叉开关模块的控制信号则来源于路由仲裁模块，具体结构如图 7 所示。

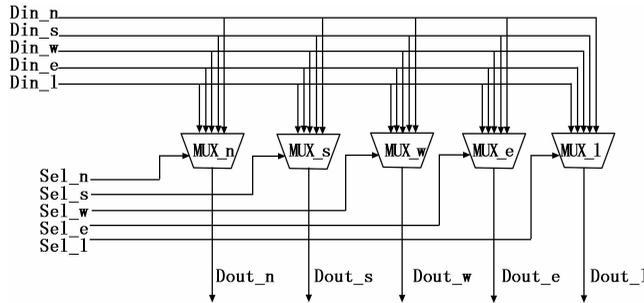


图 7 交叉开关结构图

交叉开关的工作波形如图 8 所示，5 个输入通道分别输入不同的微片。测试时对于每个输出通道对应的多路选择器，输入了特定的激励信号使得同方向输入端口数据被原路输出，以便观察。例如本地（L）通道输入 32' h40044000，在向本地（L）输出端的多路选择器控制信号端（Sel_l）输入相应的控制信号后，本地（L）输出端就会输出 32' h40044000。

3 仿真与结果分析

路由器仿真验证是片上网络仿真的一个重要环节，为

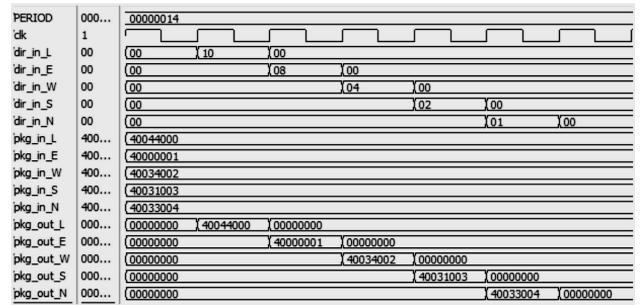


图 8 交叉开关波形图

了确保 NoC 路由器的设计符合目标，本文采用 Verilog HDL 实现了路由器的各个模块，并在 Modelsim 10.1 中对模块进行了测试，最后进行了路由器整体测试。

图 9 是路由器整体测试波形图，为了显示清晰，数据包采用 3 个微片的长度。假设路由器所在节点为（3，3），对于东（E）方向通道输入的数据包，解析其头微片信息可知其目的节点为（2，2）。根据 XY 路由算法可知路由器节点应该向 X 轴负方向路由数据包，即将数据包从西（W）方向端口输出至（2，3）节点。图 8 中西（W）方向端口以流水的方式输出东（E）方向输入的数据包，证明路由器工作正常。为了测试路由器中的路由仲裁模块与交叉开关模块功能，在五个方向上的输入端口均输入了数据包。实验结果表明，路由器能够建立各端口的输入输出连接，交叉开关模块与仲裁模块的功能符合设计目标。

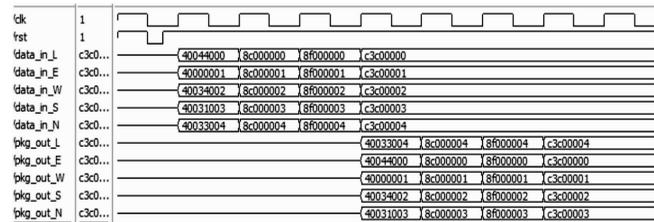


图 9 路由器波形图

4 结束语

NoC 是未来众核芯片的主流架构，而路由器又是 NoC 片内互连的关键部件，因此研究路由器的设计实现有着重要的意义。本文提出了一种可应用于 2D Mesh 拓扑结构，采用轮询仲裁机制与虫孔交换流控的片上网络路由器设计。随后对路由器的输入缓存模块、交叉开关模块、路由计算模块与路由仲裁模块进行了硬件设计与仿真，最后对路由器整体功能进行测试。实验结果表明，该路由器设计工作正常，能够正确收发数据包。

参考文献：

[1] Choquette J H, Giroux O, Foley D, et al. Volta: Performance and Programmability [J]. IEEE Micro, 2018, 38 (2): 42-52.
 [2] Foley D, Danskin J M. Ultra-Performance Pascal GPU and NVLink Interconnect [J]. IEEE Micro, 2017, 37 (2): 7-17.

(下转第 217 页)