

一种完全自适应片上路由器的设计

吕耀刚¹, 蒋 林²

(1 西安邮电学院 计算机学院, 陕西 西安 710061; 2 西安邮电学院 电子工程学院, 陕西 西安 710061)

摘 要: 针对片上网络的死锁问题, 文中根据所提出的完全自适应路由思想—基于虚拟网络的自适应路由算法, 设计了一个完全自适应片上路由器。重点介绍了路由算法及路由器的系统结构, 最终设计实现了一个高效、低代价的完全自适应路由器, 并在 2D Mesh 拓扑结构下对其性能进行了模拟验证。实验得出该路由器实现了无死锁的自适应路由, 并降低了平均网络延迟, 提高了网络吞吐量。

关键词: 片上网络; 路由算法; 路由器; 无死锁

中图分类号: TP302; TP393

文献标识码: A

文章编号: 1000-7180(2011)07-0159-05

Design and Implementation of One Fully Adaptive Router on Network on Chip

LV Yao-gang¹, JIANG Lin²

(1 School of Computer Science, Xi'an University of Posts and Telecommunications, Xi'an 710061, China;

2 School of Electronic Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710061, China)

Abstract: Due to the deadlock problem of network on chip, the paper designs one fully adaptive router, which based on the proposed thought—adaptive routing algorithms of virtual network. It mainly introduces the structure of the router's system and algorithm, in the end, a high effective and good performance router is completed, and simulates in the 2D Mesh topology structure. It makes the router deadlock-free adaptive routing come true, reducing the average network latency, and improving the network throughput.

Key words: network on chip; routing algorithm; router; deadlock-free

1 引言

NoC 是指一个芯片内集成大量的计算资源、存储资源以及连接这些资源的通信网络, 片内的通信网络用来实现这些资源之间的高速通信。它采用基于包交换和分层方法来替代原先的传统总线, 实现了处理单元(IP 核)与通信结构(网络)的分离^[1]。通讯网络主要由路由器、网络接口和链路三个基础部件组成。其中路由器的数据输出效率直接决定网络的整体性能, 设计开销直接影响整个网络架构的开销。作为 NoC 的核心组件, 片上路由器的设计得到了广泛的关注。随着 NoC 在异构多核处理器等领域的广泛应用, 其网络流量特性多表现为非均匀分布,

为了追求系统性能的进一步提升, 简单的确定性路由算法已不能再满足现在的需求。路由器采用的路由算法决定着路由器的结构和性能^[2], 一个好的路由算法, 能在网络重负载下提高网络的吞吐率, 在网络轻负载下降低网络平均时延, 对网络的性能有着举足轻重的影响。

2 基于 VN 的自适应路由算法^[3]

2.1 VN 网络概述

定义 1 虚拟网络(Virtual Network, VN): 设 $N_i = (V_{m,n}, E_k)$ 为虚拟网络中的子网络, 其中 $V_{m,n}$ 为网络中的坐标点, E_k 为子网络中通道的集合, m, n 为节点坐标, $1 \leq i \leq N$, $1 \leq k \leq 4$ 。

收稿日期: 2010-10-26; 修回日期: 2010-11-09

基金项目: 国家“八六三”计划项目(2007AA01Z111)

$E_1 = (V_{m_1, n_1} \rightarrow V_{m_2, n_2})$ 其中 $m_2 \geq m_1, n_2 \geq n_1$

$E_2 = (V_{m_1, n_1} \rightarrow V_{m_2, n_2})$ 其中 $m_2 > m_1, n_2 < n_1$

$E_3 = (V_{m_1, n_1} \rightarrow V_{m_2, n_2})$ 其中 $m_2 < m_1, n_2 > n_1$

$E_4 = (V_{m_1, n_1} \rightarrow V_{m_2, n_2})$ 其中 $m_2 < m_1, n_2 < n_1$

$VN = N_1 \cup N_2 \cup N_3 \cdots \cup N_N$

虚拟网络是用于将向一组特定目的地路由的通道集合. 对应于不同虚拟网络的通道集是不相交的, 每个报文都会根据目的地注入不同的虚拟网络并在该网络中路由, 直到到达目的地. 虚拟网络在一定程度上减轻了定义无死锁路由的任务. 在某些方法中, 在给定虚拟网络中传输的报文具有一定的、转移到其他虚拟网络的自由度. 虚拟网络可以使用每个虚拟网络中不相交的虚通道集, 并将这些通道映射到同一组物理通道加以实现.

2.2 VN 路由算法

根据 2D-Mesh 网络结构的特点, 利用坐标来标识网络中的节点.

定义 2 方向偏移: 源节点坐标 S 为 (s_x, s_y) , 目的节点坐标 D 为 (d_x, d_y) . ΔX 表示目的节点同源节点在 X 维的方向偏移, 即 $\Delta X = d_x - s_x$. ΔY 表示目的节点同源节点在 Y 维的方向偏移, 即 $\Delta Y = d_y - s_y$.

定义 3 节点路由方向集合 $D = \{X+, X-, Y+, Y-\}$. $X+$ 表示 X 维的正方向, $X-$ 表示 X 维的负方向. $Y+, Y-$ 依此类推. 具体方向如图 1 所示.

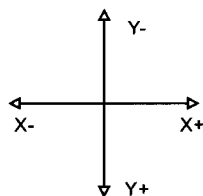


图 1 节点路由方向

在下面的算法描述中, 链路状态信号 $full_x$, $full_y$ 分别表示 X 维方向的通道已满和 Y 维方向的通道已满.

文中提出的路由算法具体思想: 先根据目的节点和源节点的位置来选择虚拟网络, 再根据各虚拟网络当前状态来确定下一跳路由器地址.

当 $\Delta X \leq 0$ 且 $\Delta Y \leq 0$, 即目的节点在源节点的左上方时, 选择 $X-Y-$ 虚拟网络, 若 $full_x$ 值为 1 则选择 Y 维方向, 若 $full_y$ 值为 1 则选择 X 维方向;

当 $\Delta X < 0$ 且 $\Delta Y > 0$, 即目的节点在源节点的左下方时, 选择 $X-Y+$ 虚拟网络, 若 $full_x$ 值为 1 则选择 Y 维方向, 若 $full_y$ 值为 1 则选择 X 维方向;

当 $\Delta X > 0$ 且 $\Delta Y > 0$, 即目的节点在源节点的

右下方时, 选择 $X+Y+$ 虚拟网络, 若 $full_x$ 值为 1 则选择 Y 维方向, 若 $full_y$ 值为 1 则选择 X 维方向;

当 $\Delta X > 0$ 且 $\Delta Y < 0$, 即目的节点在源节点的右上方时, 选择 $X+Y-$ 虚拟网络, 若 $full_x$ 值为 1 则选择 Y 维方向, 若 $full_y$ 值为 1 则选择 X 维方向.

选择 $X+Y+$ 虚拟网络算法 C 语言伪代码如下:

```

If ( $\Delta X > 0$  且  $\Delta Y > 0$ ) then
Channel := Select ( $X+, Y+$ );
Endif
if ( $full\_x == 1$ ) then
Router :=  $Y+$ ;
Endif
If ( $full\_y == 1$ ) then
Router :=  $X+$ ;
Endif

```

显然, 注入 $X+Y+$ 虚拟网络的报文只能沿 X 维和 Y 维的正方向移动, 报文根据其目的地注入某一个虚拟网络. 一旦报文在某个给定的虚拟网络中路由, 所有属于最小路径的通道都可用于路由. 但是, 报文不能被传送到另一个虚拟网络. 显然, 采用该路由算法通道间没有环相关, 从而避免了死锁.

3 基于 VN 的路由器设计与实现

3.1 路由器总体结构

目前在片上网络中路由器的结构主要采用虫孔交换, 比较典型的如虚通道虫孔交换路由器, 文中根据虚通道虫孔路由器基本结构, 采用提出的 VN 路由算法技术, 设计了一种自适应片上路由器, 该路由器内部结构包括虚通道 (VC), 路由单元, 交换分配模块 (SA), 交叉开关与输出模块. 每个物理通道是由 4 个虚通道组成, 每个虚通道是由先进先出 (FIFO) 缓冲器构成, 用于存储传输中的数据信息; 路由单元主要实现 VN 路由算法, 为输入的数据信息选择输出链路并相应地进行开关置位; 交换分配器用于产生数据信息通过数据路径时所需的控制信息; 交叉开关与输出模块主要负责将路由器的输入缓冲器连接到路由器的输出缓冲器并最终将数据输出. 总体结构如图 2 所示.

根据 NoC 的应用情况, 路由器数据格式定义如下: 每个数据片由 32 位数据和 8 位控制信息组成, 即每个时钟周期内每个端口最多接收到一个数据片. 8 位的控制信息中 6 位是虚拟网络标识符 (高 3 位表示端口号, 低 3 位表示同一端口内的虚拟网络

号),2 位指示当前数据片是否为包头或包尾(01:表示头数据片,10:表示尾数据片,00:表示中间数据片,11:保留用于传送路由器的配置信息)。头数据片

的格式如表 1 所示. 非头片(中间数据片和尾数据片)的格式如表 2 所示. 为了提高通道利用率每个数据包由 4 个以上数据片组成.

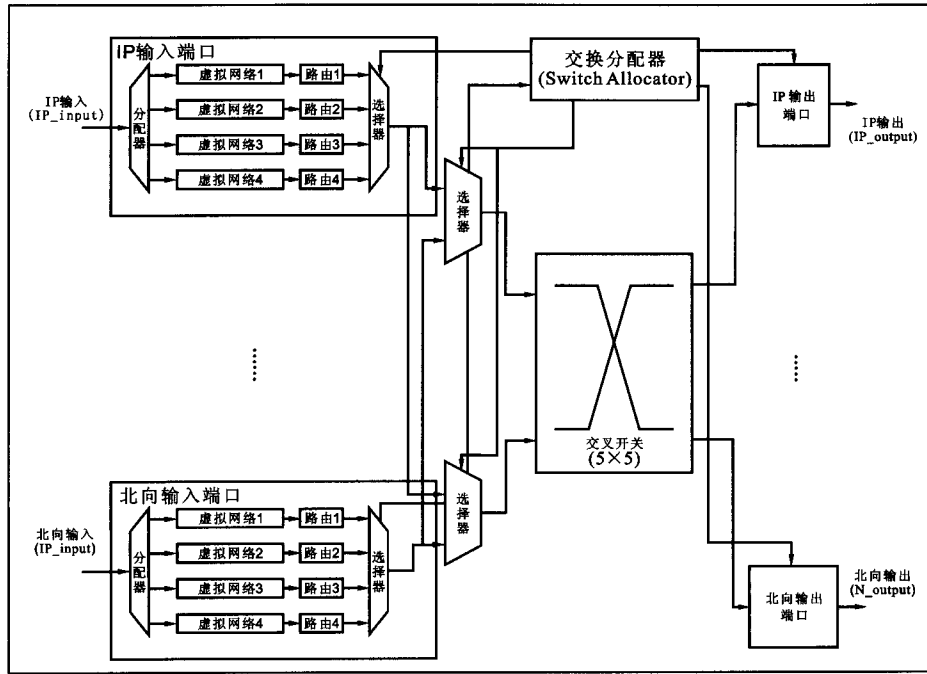


图 2 虚拟网络路由器内部结构

表 1 头片格式定义

端口号(3 位)	虚网络号(3 位)	Flit 类型(2 位)	源地址(6 位)	目标地址(6 位)	配置信息(20 位)
Ip:000	虚网络 1:000	头片:01	高 3 位为 X 地址:	高 3 位为 X 地址:	待定
E:001	虚网络 2:001		000—111;	000—111;	
W:010	虚网络 3:010		低 3 位表示 Y 地址:	低 3 位表示 Y 地址:	
S:011	虚网络 4:011		000—111	000—111	
N:100	Default: 111				
111 为无效					

表 2 中间片和尾片格式

端口号(3 位)	虚网络号(3 位)	Flit 类型(2 位)	数据净荷(32 位)
Ip:000; E:001; W:010; S:011; N:100;		尾片:10	
111 为无效	同头数据片	中间片:00	

3.2 路由策略的实现方式

路由策略是用来确定包交换每一跳的路由,文中的路由策略为:IP 核中的数据在进入路由器之前,根据源地址和目的地址选择进入相应子网络,子网络一旦确定不再改变,直到到达目的节点. 在每个子网络中,数据片可根据节点链路的当前状态来确定下一跳路由器地址. 对路由节点采用 XY 坐标二进制编码方式,在判断下一跳路由器地址时,只需比较当前地址与目的地址的大小,从而修改数据包头中的端口号.

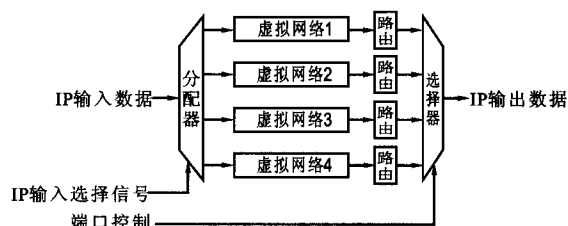
3.3 功能部件设计

输入模块:该模块由五部分组成,如图 3 所示. 分别为 IP、东、西、南、北五个端口方向. 其中 IP 方向与其它四个端口不同,首先要根据头片中源地址和目的地址的位置关系来判断数据进入哪个虚拟网络. 东、西、南、北四个端口的处理方式相同,根据输入的数据头片中的端口号及网络号,来路由由下一级要走的端口号. 各个端口均采用 4 个虚通道,每个虚通道 4 个数据片深度的缓存,用来接收来自 IP 核产生的数据. 同时,进行路由算法的计算(即路由进入哪个子网);4 个虚通道,两个方向中的两个通道

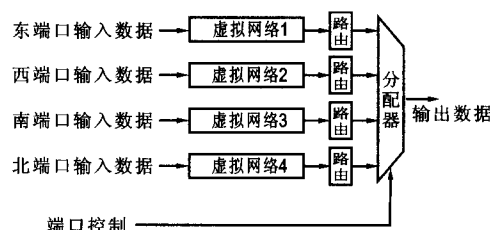
可以组成一个网络. 每个通道深度为 4 个数据片, 用来接收同一个子网络中的数据. 同时, 进行路由计算下一跳. 路由算法采用的是自适应算法, 当 X 方向有阻塞时, 选择 Y 方向, 若 Y 方向阻塞时等待.

虚通道缓存阵列模块用于缓存 flit 数据, 主要是由多个 FIFO 组成, 为了和虚通道虫孔交换路由器进行比较, 缓存的大小(FIFO 的数量)和端口数的多少有关, FIFO 的宽度是 flit 的宽度, 每个输入端口可以独立地将输入 flit 写入分配的 FIFO, 每个

输入端口可以同时写不同的 FIFO. FIFO 中 flit 根据使能信号和已经计算好的输出端口请求相应的输出, 对多个 FIFO 中的 flit 同时请求输出, 采用轮询的策略, 将获得发送权 FIFO 中的 flit 送到输出端口. 缓存阵列可以同时向不同的输出通道发送 flit 数据. 缓存阵列中的基本结构如图 4 所示, 主要根据交换控制模块产生的控制信号进行正确的输入和输出.



(a) IP 端口输入模块结构图



(b) 东西南北端口输入模块结构图

图 3 输入模块结构图

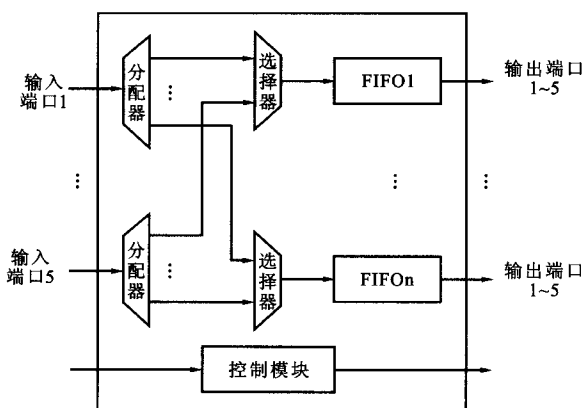


图 4 虚通道缓存阵列结构图

路由单元: 由于虚拟网络中的每个子网络路由算法不同, 所以每个子网络中都有一个路由单元, $X+Y$ + 虚拟网络中的路由单元只路由 $X+$ 和 $Y+$ 两个方向, $X+Y-$ 虚拟网络只路由 $X+$ 和 $Y-$ 两个方向, $X-Y+$ 虚拟网络只路由 $X-$ 和 $Y+$ 两个方向, $X-Y-$ 虚拟网络只路由 $X-$ 和 $Y-$ 两个方向. 在每个子网络中, 路由器根据当前网络的状态来判断是走 X 方向还是走 Y 方向, 具体的状态转移过程如图 5 所示.

交换分配模块: 交换分配模块在 4 个虚拟网络中, 需要根据端口的请求信号来产生相应的控制信号. 当不同的 5 个端口数据同时请求一个输出端口时, 为优先级高的端口中的子网优先进行分配.

交叉开关与输出模块: 由一个 40 位宽度的 5×5 全连接交叉开关和输出端口构成, 如图 6 所示. 主要是接收输入端口的输入数据并发送到相应的输出端

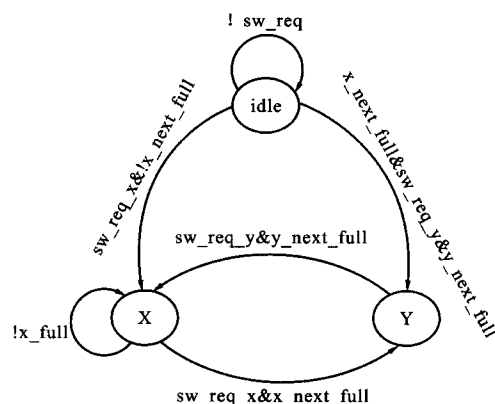


图 5 路由状态图

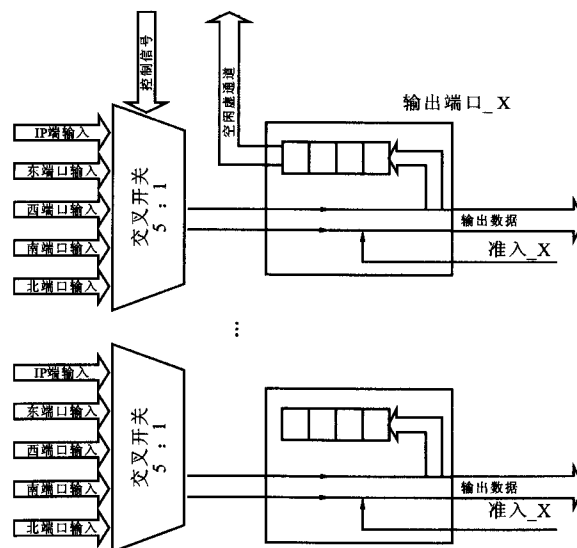


图 6 交叉开关及输出模块结构图

口. 在输出端口输出数据时,若当此数据片为尾数据片,则释放其所占用的虚通道资源,若该数据片为头数据片或中间数据片,则无需释放占用的虚通道资源.

4 实验结果

对基于 VN 的自适应片上路由器采用 Verilog 语言描述电路,使用 Cadence 公司在 Ncverilog 进行了功能仿真,结果与设计一致,选用 Altera 公司 Stratix III 系列的芯片进行了电路验证. 验证结果证明设计方案和设计电路完全符合技术规范要求.

在 4×4 2D Mesh 拓扑结构下,采用自相似分布,泊松分布,均匀分布及随机分布四种流量模型. 每个数据包长度为 4 个 Flit,每个 Flit 为 40 位宽的数据信息,FIFO 深度为 4. 实验结果如图 7 所示. 图 7 (a)给出了在不同流量模型下的网络平均延迟随着负载量变化的曲线图,可以看到不同的流量模型呈现出

不同的变化趋势,由于均匀分布流量平均,所以随着负载的增加 15% 后,延迟趋于平稳,而其他三种流量在网络负载量轻时起伏比较大,随着负载超过 15% 也慢慢趋于平稳. 总趋势是在负载量从 0~15% 变化区间内网络平均延迟随着负载的增大快速增大. 当负载量超过 15% 之后,随着负载量的增加网络延迟开始下降,并慢慢趋近于一个比较稳定的值. 图 7 (b)给出了在不同时间流量模型下的网络平均吞吐量随着负载量的增加而变化的曲线图. On_off 模型吞吐量起伏变化较大,是因为其在 On 阶段数据量比较大,其他流量变化比较平稳. 总趋势是当负载达到 15% 左右的时候吞吐量达到最大值,在负载量从 0~15% 变化区间内吞吐量随着负载的增大快速增大. 当负载量超过 15% 之后,由于负载过大,网络出现拥塞情况,所以吞吐量开始降下来,并慢慢接近一个平稳值. 可以看出,该自适应片上路由器有效地降低了网络延迟和提高了网络的吞吐量.

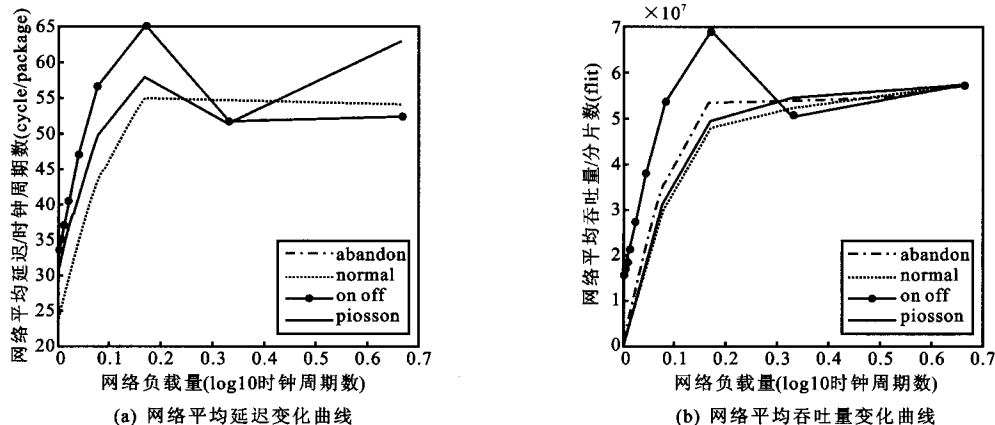


图7 不同流量模型下的网络平均延迟和网络平均吞吐量变化曲线

5 结束语

文中设计并实现了一个基于虚拟网络的自适应片上路由器,该路由器采用基于虚拟网络的路由算法实现,并采用 Verilog 语言描述电路,在 FPGA 开发板上对电路进行了功能验证,并在 2D Mesh 拓扑结构下对其性能进行了模拟验证. 实验结果表明,该路由器可有效避免网络的拥塞和死锁,极大的提高了网络吞吐量和降低了网络延迟.

参考文献:

[1] Hu J, Marculescu R. Energy and performance-aware

mapping for regular NoC architectures[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2005;551-562.

[2] 刘宝起,王长山,刘瑞. 基于保证 QoS 的片上网络路由算法[J]. 微电子学与计算机,2009,26(7):53-56.

[3] 李翠锦,吕耀刚,蒋林,等. 一种片上网络自适应路由算法仿真与验证[J]. 电子设计工程,2010,18(7):1-4.

作者简介:

吕耀刚 男,(1985-),硕士研究生. 研究方向为集成电路设计.

蒋林 男,(1970-),博士,教授,硕士生导师. 研究方向为专用集成电路设计.