案卷号	
日期	

RDMP Router 设计方案

作　　者：万健
完成日期：2010/07/28
签 收 人：
签收日期：

修改情况记录：

版本号	修改批准人	修改人	修改日期	签收人
V1.0		万健	2010/7/28	
V1.1		秦云海	2010/7/29	
V1.2		秦云海	2010/8/11	

目 录

插图目录

表格目录

RDMP Router 设计方案

1 路由器概述

本设计所完成的路由器是面向NoC平台的，它构成了平台的通信节点，承担NoC系统资源节点间所有的通信任务。路由器将根据NoC网络协议完成数据包的接收、优先级编码、地址解码、仲裁、发送等功能。

2 功能概述

通讯节点（Router）在数据传输的不同阶段具有不同的功能。在链路建立阶段，通讯节点首先存储请求包，然后进行地址解码、仲裁（路由），再将请求包转发到下一通讯节点；在数据传输阶段，通讯节点不对数据包进行任何处理，直接将数据包经过一级流水线发送到下一通讯节点。通讯节点的功能特点决定了它应该有两条通路：路由通路和数据通路。一个通讯节点只有一条路由通路，路由通路根据请求包建立网络链路，路由延迟较大（包括：6个网络时钟的请求握手延迟和1个网络时钟的响应延迟）；数据通路由路由通路进行控制，负责数据传输，时延较小（仅为1个网络时钟延迟），一个通讯节点有多个数据通路(参数配置)。

3 路由器结构设计

根据功能描述，Router需要两条通路：路由通路和数据通路，如图3-1所示。

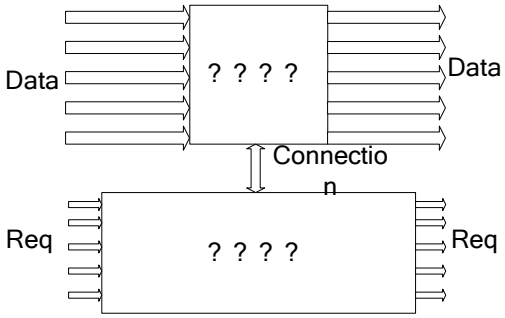


图 3-1 PCCs 交换结构图

路由器设计为五个双向端口：东(East)、南(South)、西(West) 、北(North)和本地(Local)。每个端口具有输入和输出通道。本地端口负责路由器和本地子系统的通信，其它端口负责路由器与近邻路由器的通信。Router结构如图3-2所示（图中箭头只表示数据和控制信号的方向，不区分数据信号和控制信号）。

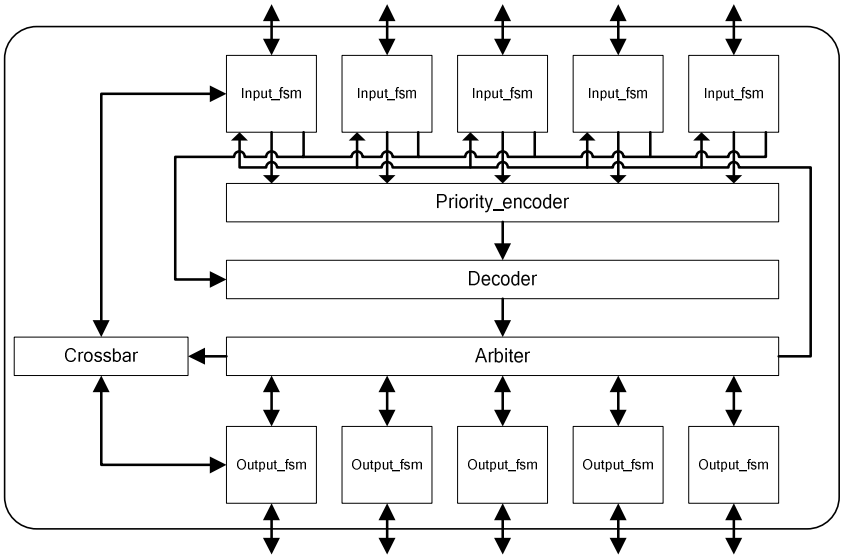


图 3-2 Router 结构图

3.1 路由通路

图3-3以5输入5输出 PCCs交换节点为例，画出了通讯节点的路由通路架构图。该通路包括输入状态机(Input_fsm)、优先级控制器(Priority_encoder)、地址解码器(Decoder)、仲裁器(Arbiter)和输出状态机(Output_fsm)。

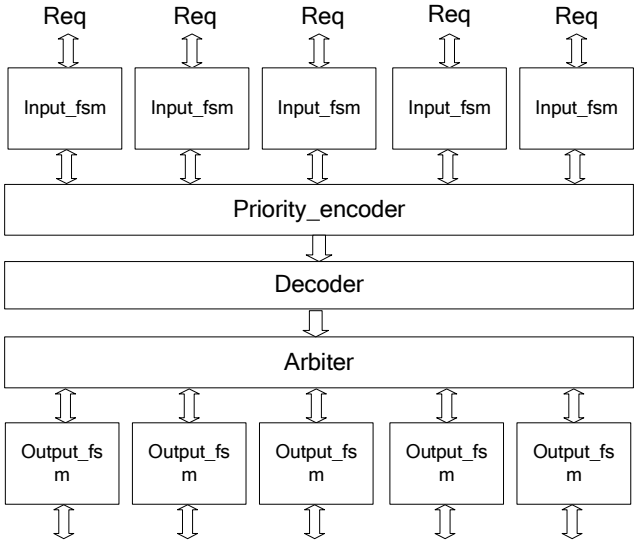


图 3-3 路由通路结构图(5×5)

3.2 数据通路

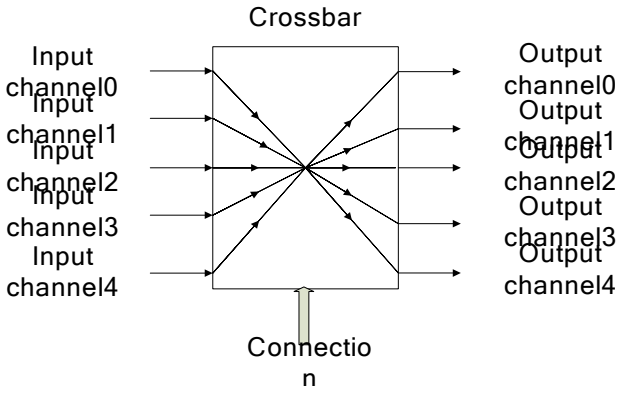


图 3-4 数据通道示意图

数据通路是一个参数化的交叉开关(Crossbar)，如图3-4所示。链路建立成功之后，仲裁器(Arbiter)模块通过Connection信号控制数据通路的连接，输入端口通过数据通路直接与相应的输出端口连接，进入数据传输阶段。在数据传输阶段，输入端口的数据信号通过数据通路直接传向下一节点，而不经过路由通路。数据通路只包含一级流水线，因而数据信号跨过一个通讯节点的时间延迟仅为一个周期。当目标节点释放链路时，数据通路内输入端口与输出端口的对应关系被撤消。

4 具体模块设计

4.1 Input_fsm（输入状态机）

4.1.1 功能概述

Input_fsm模块的功能是控制输入通道的工作状态。Input_fsm有五个工作状态下：F_IDLE（空闲）、F_REQ（请求）、F_PRELOCK（半锁定）、F_LOCK（锁定）和F_FAIL（失败）。复位时，输入通道工作在空闲态(F_IDLE)，并使地址寄存器（address_r）复位。当上游节点通过该通道发送路由请求包(stb信号为高，data信号为建立链路请求包，同时fwd信号为高)时，如果该通道工作在空闲态，则将该请求包的地址信息进行存储，并进入请求状态(F_REQ)，同时向优先级控制器发出路由请求。如果根据优先级该通道的请求得到授权(grant)，它将进入半锁定状态(F_PRELOCK)；如果请求被拒绝(deny)，该通道将进入失败状态(F_FAIL)。处于半锁定状态时，请求包的目标地址被提取出来并送入地址解码器、仲裁器(Arbiter)等，如果路由成功(pack)，通道将进入锁定状态(F_LOCK)，并使pack信号为高电平，告知请求建立链路的一方链路建立成功；如果路由失败(deny)，将进入失败状态。在失败状态下，该节点向上游节点发送路由失败信号（fail），则请求建立链路的一方应将stb信号置为低电平，取消链路建立请求。节点被锁定以后，便可高效传输数据包，此时fwd信号应该与数据信号data对应，当fwd信号为高电平时表示数据链路上的数据有效，否则数据链路上的数据无效。如果节点收到源节点发出的链路释放信号（cancel），则请求建立链路的一方应将stb信号置为低电平，节点跳出锁定态，回到空闲态。如图4-1所示。

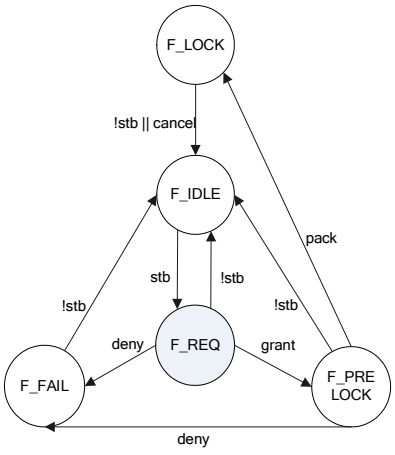


图 4-1 Input_fsm 状态转移图

4.1.2 端口说明

Input_fsm模块的信号由图4-2所示：

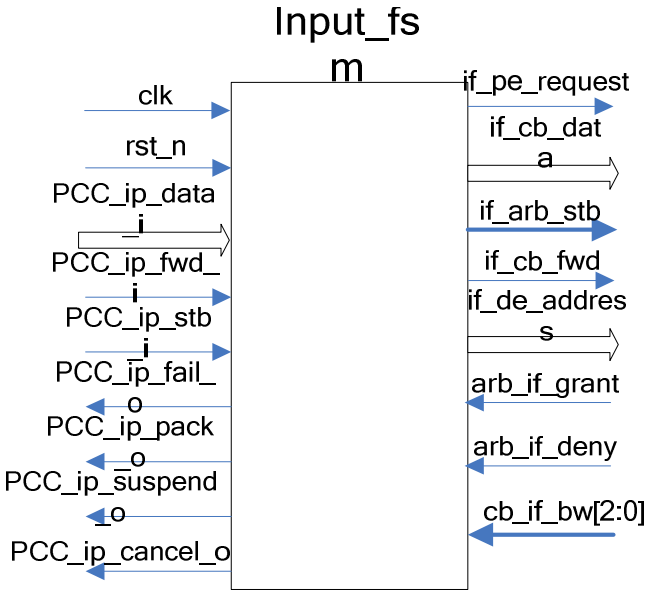


图 4-2 Input_fsm 模块的管脚结构图

Input_fsm模块端口说明见表4-1。

表 4-1 Input_fsm 模块端口说明表格

端口名	方向	描述
clk	左侧：输入	网络时钟信号
rst_n	左侧：输入	复位信号，低电平有效
PCC_ip_data_i	左侧：输入	输入数据信号，可传输路由请求包、数据或地址、Burst 类型包、结束包。
PCC_ip_fwd_i	左侧：输入	传输控制信号。该信号为高电平时，input_fsm 模块对 ip_data_i 和 ip_fwd_i 上的信号进行寄存。
PCC_ip_stb_i	左侧：输入	链路建立信号。该信号从低电平变为高电平时，表示有链路建立请求，input_fsm 的状态由 F_IDLE 变为 F_REQ。
PCC_ip_fail_o	左侧：输出	路由阻塞信号。当 input_fsm 的状态为 F_FAIL 时，此信号为高电平；其余时候都为低电平。
PCC_ip_pack_o	左侧：输出	应答信号，转发 pack 信号。在 F_PRELOCK 状态，若此信号为有效（高电平），则状态变为 F_LOCK。
PCC_ip_suspend_o	左侧：输出	挂起信号，表示链路被挂起，不能传输数据，此处转发收到的 suspend 信号。
PCC_ip_cancel_o	左侧：输出	链路释放信号，转发 cancel 信号。在 F_LOCK 状态，若此信号为有效（高电平），则状态变为 F_IDLE。

arb_if_grant	右侧：输入	arbiter 模块传入的确认信号，表示 arbiter 模块已经对请求做出了仲裁，并且有可用的路由路径。
arb_if_deny	右侧：输入	arbiter 模块传入的否认信号，表示没有可用的路由路径。
cb_if_bw	右侧：输入	crossbar 模块传回的 pack、suspend 和 cancel 信号。
if_pe_request	右侧：输出	在 F_REQ 状态，此信号有效（高电平），表示优先级请求。
if_cb_data	右侧：输出	将数据放到 crossbar 的数据输入上。
if_arb_stb	右侧：输出	将链路建立信号放到 arbiter 的输入上。
if_cb_fwd	右侧：输出	将传输控制信号放到 crossbar 的传输控制输入上。
if_de_address	右侧：输出	将地址信号传递给 decoder 模块。

4.2 Priority_encoder（优先级编码器）

4.2.1 功能概述

通讯节点具有多个输入和输出通道，而只有一个路由通路。当多个输入通道同时发出路由请求时，由priority_encoder模块判定各通道获得路由权的顺序。在本设计中，优先级顺序是：L>E>S>W>N。图4-4是优先级控制器管脚结构图。

4.2.2 端口说明

Priority_encoder模块的信号由图4-3所示：

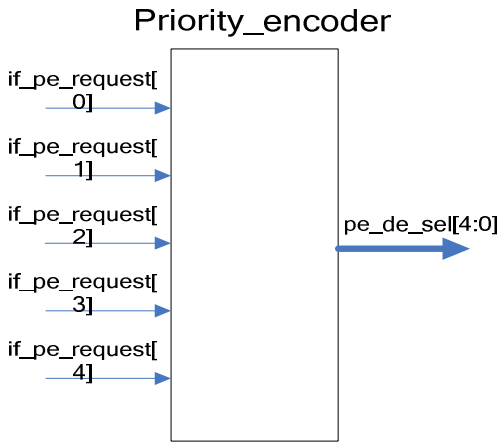


图 4-3 Priority_encoder 模块的管脚结构图

Priority_encoder模块端口说明见表4-2。

表 4-2 Priority_encoder 模块端口说明表格

端口名	方向	描述
if_pe_request[0]	左侧：输入	L 方向的优先级编码请求信号。
if_pe_request[1]	左侧：输入	E 方向的优先级编码请求信号
if_pe_request[2]	左侧：输入	S 方向的优先级编码请求信号
if_pe_request[3]	左侧：输入	W 方向的优先级编码请求信号
if_pe_request[4]	左侧：输入	N 方向的优先级编码请求信号
pe_de_sel	右侧：输出	选择信号，表示某一方向的请求信号被选择。

4.3 Decoder（地址解码器）

4.3.1 功能概述

地址解码器是通讯节点的核心功能模块。其输入是目标节点地址，输出是面向各个方向的路由请求信号。本设计中，通讯节点采用的路由算法是回退转向路由（参见第5部分路由算法）。地址解码器通过比较目标地址和本地地址，对L、E、S、W和N五个方向做出选择。

4.3.2 端口说明

Decoder模块的信号由图4-4所示：

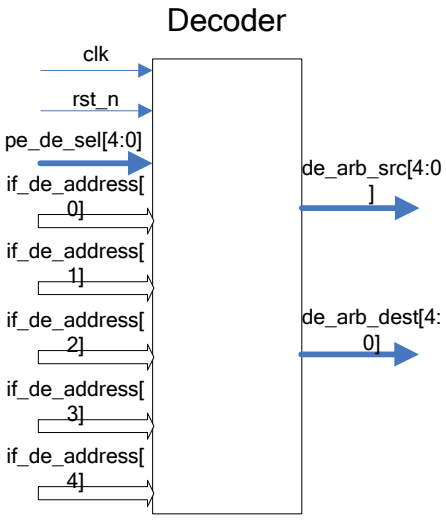


图 4-4 Decoder 模块的管脚结构图

Decoder模块端口说明见表4-3。

表 4-3 Decoder 模块端口说明表格

端口名	方向	描述
clk	左侧：输入	网络时钟信号
rst_n	左侧：输入	复位信号，低电平有效
pe_de_sel	左侧：输入	priority_encoder 模块中选中的请求信号。
if_de_address[0]	左侧：输入	input_fsm0 模块传入的地址信号。
if_de_address[1]	左侧：输入	input_fsm1 模块传入的地址信号。
if_de_address[2]	左侧：输入	input_fsm2 模块传入的地址信号。
if_de_address[3]	左侧：输入	input_fsm3 模块传入的地址信号。
if_de_address[4]	左侧：输入	input_fsm4 模块传入的地址信号。
de_arb_src 1	右侧：输出	表示被选择的输入端口号。
de_arb_dest	右侧：输出	表示选择的路由方向。

批注 [z1]: 由优先级编码器决定选择的那一个输入。

4.4 Arbiter（仲裁器）

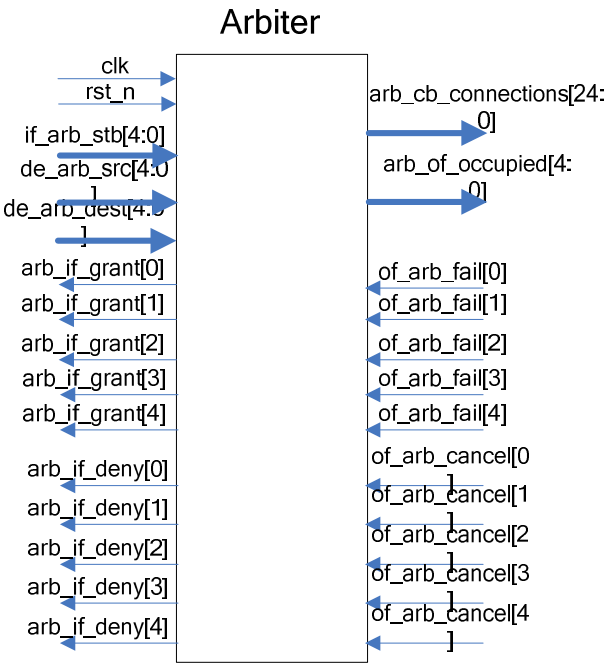
4.4.1 功能概述

Arbiter 模块接受Decoder 模块的地址解码结果，并根据固定的优先级顺序（L>E>S>W>N），选择合适的输出端口进行传输。如果选择的第一个端口已经被占用，则转而检查另一端口的状态；如果没有合适的输出端口可以选择，则向上游节点返回路由阻塞信号（deny）。上游节点收到该信号后，根据回退转向路由算法可以重新选择另一方向进行路由。

批注 [z2]: 由此实现回退转向路由。

4.4.2 端口说明

Arbiter模块的信号由图4-5所示：



图表 4-5 Arbiter 模块的管脚结构图

Arbiter模块端口说明见表4-4。

表 4-4 Arbiter 模块端口说明表格

端口名	方向	描述
clk	左侧：输入	网络时钟信号
rst_n	左侧：输入	复位信号，低电平有效
if_arb_stb	左侧：输入	建立链路信号，高电平有效
de_arb_src	左侧：输入	decoder 模块传入的被选择的输入端口号。
de_arb_dest	左侧：输入	decoder 模块传入的被选择的路由方向。
arb_if_grant[0]	左侧：输出	arbiter 模块传给 input_fsm0 模块的确认信号，表示是否有可用的路由路径。
arb_if_grant[1]	左侧：输出	arbiter 模块传给 input_fsm1 模块的确认信号，表示是否有可用的路由路径。
arb_if_grant[2]	左侧：输出	arbiter 模块传给 input_fsm2 模块的确认信号，表示是否有可用的路由路径。
arb_if_grant[3]	左侧：输出	arbiter 模块传给 input_fsm3 模块的确认信号，表示是否有可用的路由路径。
arb_if_grant[4]	左侧：输出	arbiter 模块传给 input_fsm4 模块的确认信号，表示是否有可用的路由路径。
arb_if_deny[0]	左侧：输出	是 arbiter 模块传给 input_fsm0 模块的否认信号，表示没有可用的路由路径。
arb_if_deny[1]	左侧：输出	是 arbiter 模块传给 input_fsm1 模块的否认信号，表示没有可用的路由路径。

arb_if_deny[2]	左侧：输出	是 arbiter 模块传给 input_fsm2 模块的否认信号，表示没有可用的路由路径。
arb_if_deny[3]	左侧：输出	是 arbiter 模块传给 input_fsm3 模块的否认信号，表示没有可用的路由路径。
arb_if_deny[4]	左侧：输出	是 arbiter 模块传给 input_fsm4 模块的否认信号，表示没有可用的路由路径。
of_arb_fail[0]	右侧：输入	output_fsm0 模块传人的 fail 信号，有效时（高电平）表示此输出端口向下游 Router 路由不成功。
of_arb_fail[1]	右侧：输入	output_fsm1 模块传人的 fail 信号，有效时（高电平）表示此输出端口向下游 Router 路由不成功。
of_arb_fail[2]	右侧：输入	output_fsm2 模块传人的 fail 信号，有效时（高电平）表示此输出端口向下游 Router 路由不成功。
of_arb_fail[3]	右侧：输入	output_fsm3 模块传人的 fail 信号，有效时（高电平）表示此输出端口向下游 Router 路由不成功。
of_arb_fail[4]	右侧：输入	output_fsm4 模块传人的 fail 信号，有效时（高电平）表示此输出端口向下游 Router 路由不成功。
of_arb_cancel[0]	右侧：输入	output_fsm0 模块传入的 cancel 信号，有效时（高电平）表示此输出端口的链路释放。
of_arb_cancel[1]	右侧：输入	output_fsm1 模块传入的 cancel 信号，有效时（高电平）表示此输出端口的链路释放。
of_arb_cancel[2]	右侧：输入	output_fsm2 模块传入的 cancel 信号，有效时（高电平）表示此输出端口的链路释放。
of_arb_cancel[3]	右侧：输入	output_fsm3 模块传入的 cancel 信号，有效时（高电平）表示此输出端口的链路释放。
of_arb_cancel[4]	右侧：输入	output_fsm4 模块传入的 cancel 信号，有效时（高电平）表示此输出端口的链路释放。
arb_cb_connections	右侧：输出	此信号是传给 crossbar 模块的仲裁结果，表示了输出端口与输入端口的对应关系。
arb_of_occupied	右侧：输出	此信号是传给 output_fsm 模块的仲裁结果，表示某个输出端口是否被选择。

4.5 Output_fsm（输出状态机）

4.5.1 功能概述

输出状态机工作在两个状态：空闲态和锁定态(如图4-6所示)，前者表示输出通道当前可用，后者表示输出通道已被其它通道占用。在空闲态时，如果仲裁器授权该通道传输数据(occupied为高电平)，则相应输出状态机进入锁定态，直到上一通讯节点发出链路释放信号，使arbiter模块输入释放端口信号（occupied为低电平），输出状态机随即返回空闲态。

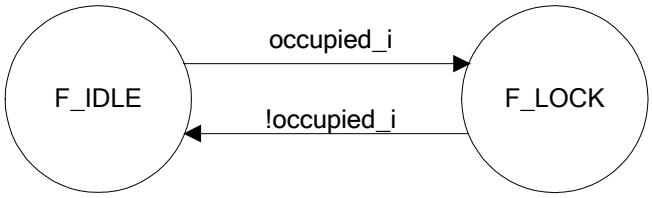


图 4-6 Output_fsm 状态转移图

4.5.2 端口说明

Output_fsm模块的信号由图4-7所示：

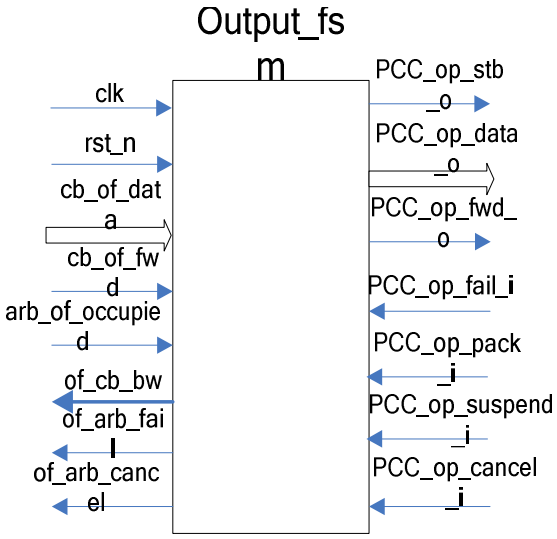


图 4-7 Output_fsm 模块的管脚结构图

Output_fsm模块端口说明见表4-5。

表 4-5 Output_fsm 模块端口说明表格

端口名	方向	描述
clk	左侧：输入	网络时钟信号
rst_n	左侧：输入	复位信号，低电平有效
cb_of_data	左侧：输入	由 crossbar 模块传入的数据信号
cb_of_fwd	左侧：输入	由 crossbar 模块传入的传输控制信号。
arb_of_occupied	左侧：输入	由 arbiter 模块传入的状态控制信号，表示此输出端口是否被选择占用。
of_cb_bw	左侧：输出	传出给 crossbar 模块的 pack、suspend 和 cancel 信号。
of_arb_fail	左侧：输出	传出给 arbiter 模块的 fail 信号。

of_arb_cancel	左侧：输出	传出给 arbiter 模块的 cancel 信号。
PCC_op_fail_i	右侧：输入	下游路由器返回的路由阻塞信号。此信号有效时，表示下游路由器不能提供合适的路由路径。
PCC_op_pack_i	右侧：输入	此信号有效时，表示链路建立成功。
PCC_op_suspend_i	右侧：输入	此信号有效时，表示此链路的建立成功，但是被挂起，不能传输数据。
PCC_op_cancel_i	右侧：输入	此信号有效时，表示此链路释放。
PCC_op_stb_o	右侧：输出	对下游路由器的链路建立信号。
PCC_op_data_o	右侧：输出	传给下游路由器的数据信号。
PCC_op_fwd_o	右侧：输出	传给下游路由器的传输控制信号。

4.6 Crossbar（交叉开关）

4.6.1 功能概述

Crossbar模块是一个可参数化的交叉开关。链路建立成功之后，仲裁器模块通过Connection信号控制数据通路的连接，输入端口通过数据通路直接与相应的输出端口连接。在数据传输阶段,输入端口的数据信号通过数据通路直接传向下一节点,而不经过路由通路。数据通路只包含一级流水线，因而数据信号跨过一个通讯节点的时间延迟仅为一个网络时钟。当目标节点释放链路时，数据通路内输入端口与输出端口的对应关系被撤消。

4.6.2 端口说明

Crossbar模块的信号由图4-8所示：

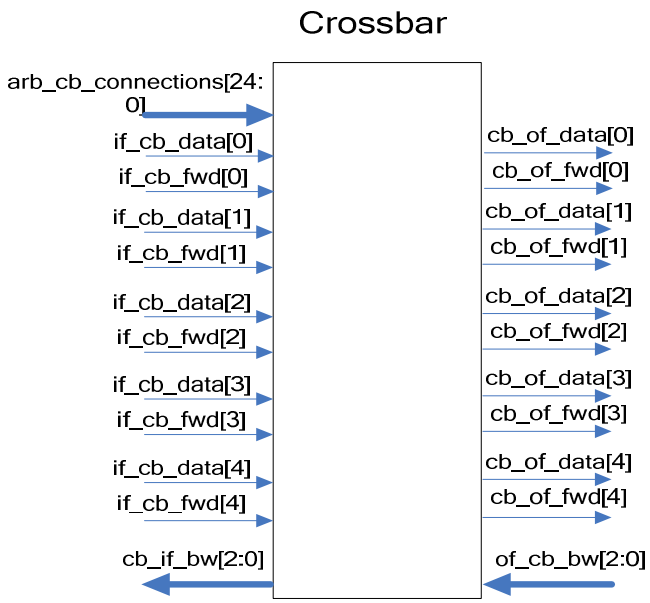


图 4-8 Crossbar 模块的管脚结构图

Crossbar模块端口说明见表4-6。

表 4-6 Crossbar 模块端口说明表格

端口名	方向	描述
arb_cb_connections	左侧：输入	Arbiter 模块传入的控制信号，包含了输入端口与输出端口之间的对应关系。
if_cb_data[0]	左侧：输入	
if_cb_data[1]	左侧：输入	input_fsm1 模块传入的数据信号。
if_cb_data[2]	左侧：输入	input_fsm2 模块传入的数据信号。
if_cb_data[3]	左侧：输入	input_fsm3 模块传入的数据信号。
if_cb_data[4]	左侧：输入	input_fsm4 模块传入的数据信号。
if_cb_fwd[0]	左侧：输入	input_fsm0 模块传入的传输控制信号。
if_cb_fwd[1]	左侧：输入	input_fsm1 模块传入的传输控制信号。
if_cb_fwd[2]	左侧：输入	input_fsm2 模块传入的传输控制信号。
if_cb_fwd[3]	左侧：输入	input_fsm3 模块传入的传输控制信号。
if_cb_fwd[4]	左侧：输入	input_fsm4 模块传入的传输控制信号。
cb_if_bw	左侧：输出	传给 Input_fsm 模块的 pack、suspend 和 cancel 信号。
of_cb_bw	右侧：输入	Output_fsm 模块传入的 pack、suspend 和 cancel 信号。
cb_of_data[0]	右侧：输出	由 crossbar 模块传给 output_fsm0 模块的数据信号。
cb_of_data[1]	右侧：输出	由 crossbar 模块传给 output_fsm1 模块的数据信

		号。
cb_of_data[2]	右侧：输出	由 crossbar 模块传给 output_fsm2 模块的数据信号。
cb_of_data[3]	右侧：输出	由 crossbar 模块传给 output_fsm3 模块的数据信号。
cb_of_data[4]	右侧：输出	由 crossbar 模块传给 output_fsm4 模块的数据信号。
cb_of_fwd[0]	右侧：输出	由 crossbar 模块传给 output_fsm0 模块的传输控制信号。
cb_of_fwd[1]	右侧：输出	由 crossbar 模块传给 output_fsm1 模块的传输控制信号。
cb_of_fwd[2]	右侧：输出	由 crossbar 模块传给 output_fsm2 模块的传输控制信号。
cb_of_fwd[3]	右侧：输出	由 crossbar 模块传给 output_fsm3 模块的传输控制信号。
cb_of_fwd[4]	右侧：输出	由 crossbar 模块传给 output_fsm4 模块的传输控制信号。

5 路由方法

回退转向路由方法是一种动态路由方法，下面结合PCCs交换节点的路由过程介绍回退转向路由方法的原理。通讯节点在处理输入通道的请求时，首先由地址解码模块(Address decoder)从输入端口读取目标节点地址，并与本地地址比较，同时根据地址解码算法选择两个可能的输出方向(如图5-1所示)。这些可能的方向都交给仲裁器模块(Arbiter)进行仲裁，其使用destinations状态寄存器记录各个输入端口的可选输出端口。destinations信号的位宽为25，每五位分为一组，最低一组表示与0号输出端口对应的五个输入端口请求状况。例如，信号destinations为00000-00000-00000-01100- 00001时，表示输出端口0是输入端口0的可选输出端口，而输出端口1是输入端口2和3的可选输出端口。

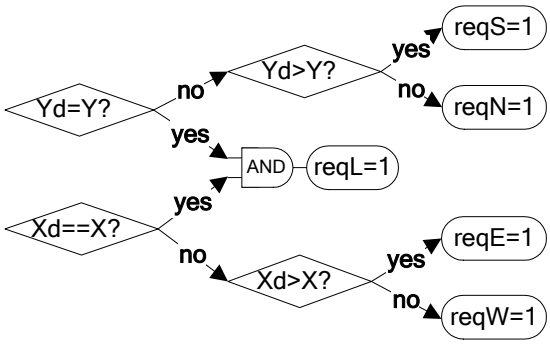


图 5-1 路由方法示意图

仲裁器模块根据优先级顺序(本地、东、南、西、北，即本地优先级最高，其次是水平方向，最后是竖直方向)和端口的占用情况，将输入端口连接到适当的输出端口。在某一时刻，一个输出端口只能与一个输入端口相连，当数据传送结束输出端口被释放后，其他的输入端口才可以选择此输出端口。如果某输入端口的所有可能选择的输出端口都被占用，那么该通讯节点将向上游节点或网络接口发送路由失败信号(Fail)。

6 数据格式

1、引导包格式（对于非芯片间传输，其中的 25:16 位均保留使用）

65	64	63:26	25：22	21：18	17：16	15：12	11：8	7：4	3：0
1	0	保留使用	目的片目的节点 y 坐标	目的片目的节点 x 坐标	源芯片标识	源节点 y 坐标	源节点 x 坐标	本片目的节点 y 坐标	本片目的节点 x 坐标

2、配置包格式（各 Wrapper 可根据需要选择使用一组或两组配置包，建议 DDR 用两组，以保持原状态机不变）

65	64	63	62：0
0	1	0	配置信息

3、数据包格式

65	64	63：0
0	0	数据

4、结束包格式

65	64	63	62：0
0	1	1	x

7 链路建立、撤销流程

链路建立过程：

1) 链路建立方（以下简称源）请求建立链路：将 stb 信号置高电平，同时给出正确包格式的请求包（数据格式见第 6 节《数据格式》）。注：请求建立链路和链路建立成功后发送数据过程中，stb 信号必须保持高电平。
2) 请求建立链路时，链路的目的方（以下简称目的方）收到高电平的 stb 信号和请求包。目的方确认可以建立链路时，将 pack 信号置高电平（一个时钟周期）。
3) 源收到 pack 信号有效（高电平）时，表示链路建立成功并锁定。
4) 如果目的方收到建立链路请求（即，收到高电平的 stb 信号和请求包），但是不能建立链路，此时需要将 fail 信号置为高电平（一个周期），表示目的方暂时无法建立链路，接收数据。
5) 如果源收到 fail 信号有效（高电平），表示链路建立不成功，此时源需要将 stb 信号置为低电平，取消建立链路请求。注：此时收到的有效 fail 信号，有两个可能来源：第一，网络上没有可以建立的通路；第二，目的方不能建立链路，接收数据。
6) 数据传输过程：
7) 链路建立成功后，可以开始数据传输过程，每个时钟发送一个数据。数据不需要连续传输，但需要注意的是，data 信号要与 fwd 信号对应，fwd 信号高电平时表示数据信号有效。

撤销链路过程：

1) 目的方接收到源发送的结束包后，将 cancel 信号置为高电平时（数据传输结束），PCCs 将沿锁定的链路依次撤销链路，并将 cancel 信号传递到源。
2) 源接收到 cancel 信号时，表示链路已经撤销，将 stb 信号置为低电平。
3) 数据传输过程中，目的方可以发送 suspend 信号（1bit）给源。suspend 信号具体作用由双方规定，PCCs 只负责传递此信号。

8 接口协议

（1）由 Wrapper 或 NI 传至 PCC（其中，routing 为上述引导包，setting 为配置包，可为一个或两个，payload 为数据包，end 为结束包）时，PCCs 入口协议如图 8-1 所示。

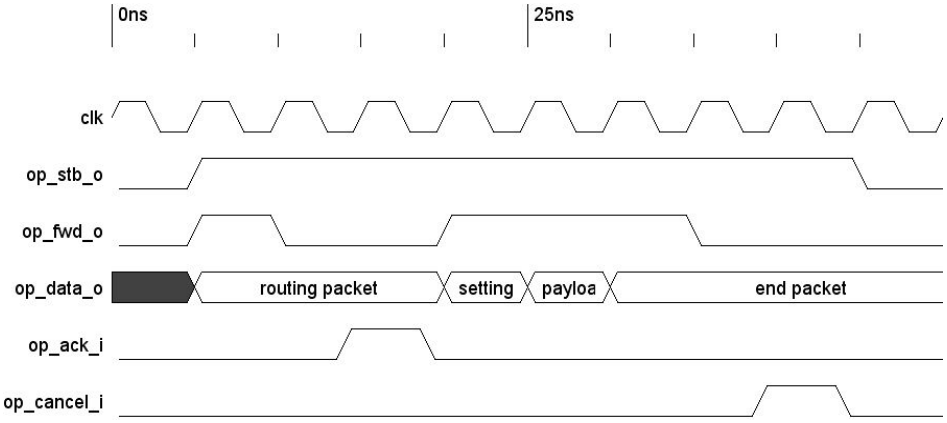


图 8-1 PCCs 入口协议

（2）由 PCC 传至 Wrapper 或 NI 时，PCCs 出口协议如图 8-2 所示。

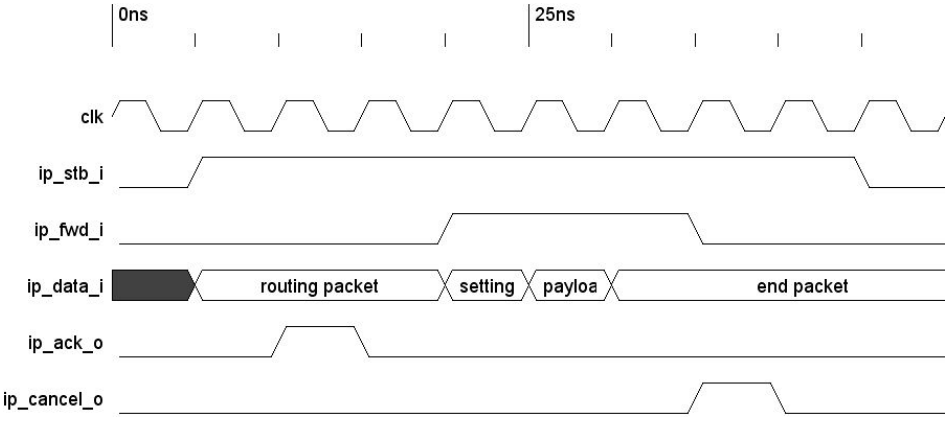


图 8-2 PCCs 出口协议

9 网络地址编址方式