



[12] 发明专利申请公开说明书

[21] 申请号 96196300.X

[43]公开日 1998年9月16日

[11] 公开号 CN 1193393A

[22]申请日 96.6.17

[30]优先权

[32]95.6.15 [33]US[31]08 / 490,654

[86]国际申请 PCT / US96 / 10451 96.6.17

[87]国际公布 WO97 / 00480 英 97.1.3

[85]进入国家阶段日期 98.2.16

[71]申请人 英特尔公司

地址 美国加利福尼亚州

[72]发明人 E·加尔布斯 P·桑哈高埃特

M·戈尔德施米特

N·伊斯坎德尔埃

[74]专利代理机构 中国专利代理(香港)有限公司

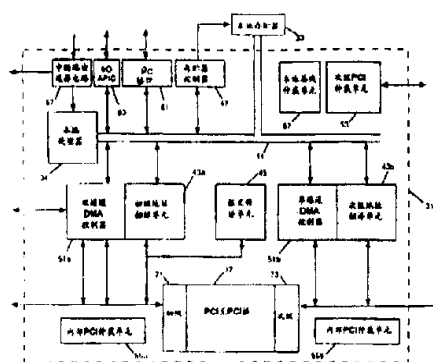
代理人 王 勇 陈景峻

权利要求书 5 页 说明书 78 页 附图页数 10 页

[54]发明名称 集成PCI至PCI桥的I/O处理器的体系结构

[57]摘要

一种集成了高性能处理器到 PCI 至 PCI 总线桥 (32) 的多功能设备 (31)。本发明用高性能处理器、PCI 至 PCI 总线桥 (32)、PCI 总线—处理器之间的地址翻译单元 (43a、43b)、直接内存存取 (DMA) 控制器 (51a、51b)、存储器控制器 (47)、次级 PCI 总线仲裁单元 (53)、互连集成 (I²C 电路总线接口单元 (61)、高级可编程中断 (APIC) 总线接口单元 (63)、报文传递单元 (45) 组成一个使用本地存储器 (33) 的单一系统。PCI 总线是一种符合工业标准的高性能、低等待系统总线。PCI 至 PCI 总线桥 (32) 为两个独立的 32 位 PCI 总线提供连接通路, 并使它们具有克服 PCI 电气负载限制的能力。本地处理器 (34) 的加入, 为 PCI 总线桥 (32) 增添了智能。



权 利 要 求 书

1. 一种连接第一外部总线与第二外部总线的集成电路, 包括:

a) 连接到所述第一外部总线的第一内部总线;

b) 连接到所述第二外部总线的第二内部总线;

5 c) 具有连接到所述第一内部总线的第一总线接口和连接到所述第二内部总线的第二总线接口的总线桥装置, 所述总线桥用于允许事务处理在所述第一与第二外部总线之间被传递;

d) 连接到所述总线桥装置、用于处理从外部源接收的事务处理和输入到所述第一总线接口与所述第二总线接口的事务处理的本地处理器装
10 置;

e) 在所述处理器装置与外部存储器之间传递数据的本地总线。

2. 权利要求 1 定义的集成电路, 进一步包括:

a) 连接到所述本地总线和所述第一内部总线的第一地址翻译电路, 该翻译电路适用于将所述第一内部总线上的向内事务处理地址转换成所述
15 本地处理器装置可用的地址, 及将所述本地总线上的向外事务处理地址转换成连接到所述第一外部总线的设备可用的地址;

b) 连接到所述本地总线和所述第二内部总线的第二地址翻译电路, 该翻译电路适用于将所述第二内部总线上的向内事务处理地址转换成所述
20 本地处理器装置可用的地址, 及将所述本地总线上的向外事务处理地址转换成连接到所述第二外部总线的设备可用的地址。

3. 权利要求 2 定义的集成电路, 进一步包括:

连接到所述第一地址翻译电路的报文传递单元, 该报文传递单元适用于
25 在新数据被放置到所述第一内部总线上时产生一个供所述本地处理器装置使用的中断, 及在所述本地处理器装置在所述本地总线上放置数据时在所述
30 所述第一内部总线的至少一条中断线上产生一个供连接到所述第一外部总线的设备使用的中断。

4. 权利要求 1 定义的集成电路, 进一步包括:

连接到所述本地总线和所述第一内部总线、用于在所述第一内部总线
与
所述本地存储器之间传输数据块的第一 DMA 控制器装置;

30 连接到所述本地总线和所述第二内部总线、用于在所述第二内部总线
与
所述本地存储器之间传输数据块的第二 DMA 控制器装置。

5. 权利要求 2 定义的集成电路, 进一步包括:



a] 控制对所述第一总线接口与所述第一地址翻译电路之间的所述第一内部总线进行访问的第一总线仲裁装置;

b] 控制对所述第二总线接口与所述第二地址翻译电路之间的所述第二内部总线进行访问的第二总线仲裁装置。

5 6. 权利要求 2 定义的集成电路, 进一步包括:

控制对所述本地处理器、所述第一地址翻译电路、所述第二地址翻译电路之间的所述本地总线进行访问的本地总线仲裁装置。

7. 权利要求 1 定义的集成电路, 进一步包括:

10 控制用所述本地总线上的数据和地址向所述外部存储器写入数据、从所述外部存储器读取数据的存储器控制器装置。

8. 权利要求 1 定义的集成电路, 进一步包括:

a] 为多个中断源选择路由到所述本地处理器装置的一个单一中断输入端的外围设备中断控制器;

15 b] 定义所述多个中断源中的哪个中断源引起所述单一中断的状态寄存器。

9. 权利要求 1 定义的集成电路, 进一步包括:

a] 连接到适合于管理采用所述集成电路的系统的所述外部传感器的 I²C 总线;

20 b] 允许所述本地处理器装置担当位于所述 I²C 总线上的主和从设备的 I²C 接口装置。

10. 权利要求 1 定义的集成电路, 进一步包括:

a] 连接外部处理器的 APIC 总线;

b] 使所述外部处理器与所述本地处理器之间能够进行通信的 APIC 接口装置。

25 11. 一个包括至少一个与一个初级 PCI 总线相连的主处理器、至少有一个与一个次级 PCI 总线相连的外围设备的系统, 该系统在一块单一集成电路上含有:

a] 连接到所述初级 PCI 总线的第一内部总线;

b] 连接到所述次级 PCI 总线的第二内部总线;

30 c] 具有连接到所述第一内部总线的第一总线接口和连接到所述第二内部总线的第二总线接口的总线桥装置, 所述总线桥用于允许在所述初级 PCI 总线与所述次级 PCI 总线之间传递事务处理;



d) 连接到所述总线桥装置、处理从外部源接收的事务处理和输入到所述第一总线接口和所述第二总线接口的事务处理的本地处理器装置;

e) 在所述处理器装置与外部存储器之间传递数据的本地总线。

12. 权利要求 11 定义的系统, 其中, 所述集成电路进一步包括:

5 a) 连接到所述本地总线和所述第一内部总线的第二地址翻译电路, 该翻译电路适用于将所述第一内部总线上的向内事务处理地址转换成可被所述本地处理器装置使用的地址, 及将所述本地总线上的向外事务处理地址转换成可被与所述初级 PCI 总线相连的设备使用的地址;

10 b) 连接到所述本地总线和所述第二内部总线的第二地址翻译电路, 该翻译电路适用于将所述第二内部总线上的向内事务处理地址转换成可被所述本地处理器装置使用的地址, 及将所述本地总线上的向外事务处理地址转换成可被与所述次级 PCI 总线相连的设备使用的地址。

13. 权利要求 12 定义的系统, 其中, 所述集成电路进一步包括:

15 连接到所述第一地址翻译电路的报文传递电路, 该报文传递单元适用于在新数据被放置到所述第一内部总线上时产生一个由所述本地处理器装置使用的中断, 及在所述本地处理器向所述本地总线上放置数据时在所述第一内部总线的至少一条中断线上产生一个由连接到所述初级 PCI 总线的设备使用的中断。

14. 权利要求 11 定义的系统, 其中, 所述集成电路进一步包括:

20 连接所述本地总线和所述第一内部总线、用于在所述第一内部总线和所述本地存储器之间传输数据块的第一 DMA 控制器装置;

连接所述本地总线和所述第二内部总线、用于在所述第二内部总线和所述本地存储器之间传输数据块的第二 DMA 控制器装置。

15. 权利要求 12 定义的系统电路, 其中, 所述集成电路进一步包括:

25 控制对所述第一总线接口和所述第一地址翻译电路之间的所述第一内部总线访问的第一总线仲裁装置;

控制对所述第二总线接口和所述第二地址翻译电路之间的所述第二内部总线访问的第二总线仲裁装置;

16. 权利要求 12 定义的系统, 其中, 所述集成电路进一步包括:

30 控制对所述本地处理器、所述第一地址翻译电路、所述第二地址翻译电路之间的所述本地总线访问的本地总线仲裁装置。

17. 权利要求 11 定义的系统, 其中, 所述集成电路进一步包括存储

器控制器装置，用于控制用所述本地总线上的数据和地址对所述外部存储器进行的读写。

18. 权利要求 11 定义的系统，其中，所述集成电路进一步包括：

5 a) 外围设备中断控制器装置，用于为多个中断源选择路由到所述本地处理器装置的单一的中断输入端；

b) 定义所述多个中断源中哪个中断源引起了所述单一中断的状态寄存器。

19. 权利要求 11 定义的系统，其中，所述集成电路进一步包括：

10 a) 用于连接适合于对使用所述集成电路的系统进行管理的外部传感器的 I²C 总线；

b) 使所述本地处理器装置能够担当位于所述 I²C 总线上的主和从设备作用的 I²C 接口装置。

20. 权利要求 11 定义的系统，其中，所述集成电路进一步包括：

a) 连接到外部处理器的 APIC 总线；

15 b) 使所述外部处理器与所述本地处理器之间能够进行通信的 APIC 接口装置。

21. 权利要求 1 定义的集成电路，其中，所述总线桥装置进一步包括：

a) 连接在所述第一总线接口与所述第二总线接口之间的下游投递缓冲器；

20 b) 连接在所述第一总线接口与所述第二总线接口之间的上游投递缓冲器；

c) 一组与所述第一总线接口连接的配置寄存器。

22. 权利要求 1 定义的集成电路，其中，所述本地处理器装置包括一个微处理器。

25 23. 权利要求 11 定义的系统，其中，所述总线桥装置进一步包括：

a) 在所述第一总线接口与所述第二总线接口之间连接的下游投递缓冲器；

b) 在所述第一总线接口与所述第二总线接口之间连接的上游投递缓冲器；

30 c) 一组与所述第一总线接口连接的配置寄存器。

24. 权利要求 11 定义的系统，其中，所述本地处理器装置包括一个微处理器。

25. 权利要求 4 定义的集成电路, 进一步包括:

a] 控制所述第一总线接口、所述第一地址翻译电路、所述第一 DMA 控制器装置之间对所述第一内部总线访问的第一总线仲裁装置;

b] 控制所述第二总线接口、所述第二地址翻译电路、所述第二 DMA 控制器装置之间对所述第二内部总线访问的第二总线仲裁装置。

26. 权利要求 14 定义的系统, 其中, 所述集成电路进一步包括:

a] 控制所述第一总线接口、所述第一地址翻译电路、所述第一 DMA 控制器装置之间访问所述第一内部总线的第一总线仲裁装置;

b] 控制所述第二总线接口、所述第二地址翻译电路、所述第二 DMA 控制器装置之间访问所述第二内部总线的第二总线仲裁装置。

说明书

集成 PCI 至 PCI 桥的 I/O 处理器的体系结构

5 本发明涉及计算机系统体系结构领域。更确切地说，本发明涉及一种用于实现计算机和服务器系统中的智能输入/输出子系统的智能总线桥。

高性能计算机系统普遍包括独立的输入/输出子系统。这种输入/输出子系统一般包括一个独立于主微处理器、执行输入/输出功能的微处理器。例如，这种输入/输出子系统可以为计算机系统执行复杂的通信网络接口功能或磁盘控制功能。

10 一般来说，输入/输出子系统中连接有用于在部件总线上通信的一组专用输入/输出设备。这种输入/输出子系统的处理器一般是通过总线来执行输入/输出功能的，它不干涉计算机系统中其它处理器的操作。总线上输入/输出事务处理的这种分离，使这种计算机系统的主处理器（组）能改进性能。这种体系结构普遍存在于主机计算机系统中，其中该处理器和输入/输出子系统被称为 I/O 通道。

随着采用微处理器的计算机系统、特别是服务器/客户机系统的出现，越来越要求有功能更强的微处理器来实现功能更强的服务器/客户机系统。如附图 1 所示的单系统 11 中多微处理器的组合部分地满足了这种需要。随着为实现功能更强的服务器/客户机系统所需 I/O 设备数量的增加而遇到的另一个问题是，连接输入/输出子系统与计算机系统中其它元素的标准部件总线具有电力负载限制。这种电力负载限制，限制了连接到标准部件总线的部件的数目。例如，现有技术水平的一个总线标准，要求系统部件互连总线上的连接器只有一种电力负载。这种电力负载限制保证满负荷总线上信号的质量足以用来进行可靠的操作。

25 这一方面，由于有些输入/输出子系统需要的通过本地部件总线通信的部件的数量大，有可能会突破标准部件总线上各连接器的规定电力负载要求，所以输入/输出子系统也可以包括一个将本地部件总线 17 连接到计算机系统中通过网络 I/O 卡 21 与诸如局域网的网络连接或通过 SCSI 控制器 23 与存储设备连接的其它部件总线 19 的总线桥电路 13。这种总线桥将微处理器（组）25、存储器 27 和输入/输出子系统的部件与其它部件总线电隔离开来。这种总线桥电路使输入/输出子系统能容纳实现输入/输出功能所需的大量部件，同时又满足对其它部件总线的电力负载要求。



此外，系统 11 中的微处理器（组）一般来说一定会与连接到部件总线的其它总线代理 (agents) 进行竞争。一般来说，这种对总线的争夺会降低微处理器（组）为输入/输出子系统执行输入/输出功能时的性能。

5 本发明是一种将高性能处理器集成到 PCI 至 PCI 总线桥 (P2P) 的多功能设备。参见附图 2，本发明将诸如英特尔公司的 80960 JF 处理器的高性能处理器（本地处理器）、PCI 至 PCI 总线桥 32、PCI 总线-处理器地址翻译单元、直接存储器访问 (DMA) 控制器、存储器控制器、次级 PCI 总线仲裁单元、互连集成电路 (I²C) 总线接口单元、高级可编程中断 (APIC) 总线接口单元、报文传递单元，组成一个使用本地存储器 33
10 的单一系统 31。该系统是一种集成处理器，它既照顾了智能 I/O 应用的各种需要，又有助于降低智能 I/O 系统的成本。

PCI 总线是一种符合工业标准 (PCI 本地总线标准 2.1 版) 的高性能、低迟滞系统总线。PCI 至 PCI 桥为两个独立的 32 位 PCI 总线之间提供了一条连接通路，提供了克服电力负载限制的能力。本地处理器的加入则为
15 PCI 桥增添了智能。附图 3 内虚框 31 所示的本地处理器和其它功能块，表示以下将称为 P2P 处理器的一个框图。

P2P 处理器是一种多功能 PCI 设备。功能元件 0 是 PCI 至 PCI 桥单元。功能元件 1 是地址翻译单元。P2P 处理器中含有可通过初级 PCI 总线访问的 PCI 配置空间。

20 较佳实施例中的本地处理器 34，是英特尔 i960 系列中的一个 80960 JF 处理器。在 P2P 处理器中实现的 80960 JF 处理器无需功能性的改动。本文提供了实施本发明所需的本地处理器有关部分的全部资料，英特尔公司的《i960 Jx 微处理器用户手册》中还有更详细的资料。

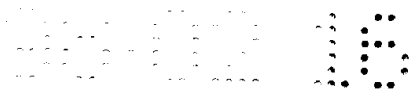
本地处理器是以其自身的 32 位地址空间而不是 PCI 地址空间进行操作的。本地处理器上的存储器可以：

- 对 PCI 地址空间可见的
- 保持对本地处理器专用的
- 这两者的结合

本地处理器总线 41

30 本地处理器总线连接到 P2P 处理器的 I/O 接插引线，提供对外部设备的总线访问。P2P 处理器为本地总线仲裁提供支持。

地址翻译单元 43a、43b 和报文传递单元 45



地址翻译单元允许 PCI 事务处理直接访问本地处理器的本地存储器 33。本地处理器 34 可直接访问两个 PCI 总线。地址翻译被提供用于 PCI 地址空间与本地处理器地址空间之间的事务处理。地址翻译是通过可编程寄存器控制的，可编程寄存器既可从 PCI 接口访问，又可从本地处理器访问，使得两种地址空间的映射灵活。报文传递单元 45 提供了一种在 PCI 系统与本地处理器之间传送数据的机制，它通过中断通知各系统有新数据到来。报文传递单元可用于发送和接收报文。

PCI 至 PCI 桥单元 32

PCI 至 PCI 桥单元连接两个独立的 PCI 总线。该桥允许某些总线事务处理被从一个 PCI 总线转移到另一个 PCI 总线上。它还允许完全独立的 PCI 总线操作，包括独立的时钟。专用数据队列支持 PCI 总线的高性能带宽。PCI 64 位双地址周期（DAC）寻址得到支持。

PCI 至 PCI 桥单元有可通过初级 PCI 总线访问的专门的 PCI 配置空间。

P2P 处理器中的 PCI 至 PCI 桥单元完全符合 PCI 专业组（Special Interest Group）出版的 PCI 桥体系结构标准（1.0 版）。

专用 PCI 设备

P2P 处理器设计上明确支持可使用次级 PCI 总线而又避开 PCI 配置软件检测的专用 PCI 设备。PCI 至 PCI 桥和次级地址翻译单元 43b 合作，在 PCI 配置周期隐藏专用设备，并允许这些设备使用专用的 PCI 地址空间。这些设备可由次级地址翻译单元通过 PCI 配置周期进行配置。

集成存储器控制器 47

集成存储器控制器为外部存储器系统提供直接控制。它支持 DRAM、SRAM、ROM 和 Flash 存储器。集成存储器控制器为通常不需要外部逻辑的存储器 33 提供一种直接连接接口。其特点是具有可编程的芯片选择、等待状态发生器、字节奇偶校验。

可将外部存储器配置为 PCI 可寻址存储器或作为专用本地处理器存储器。

DMA 控制器 51a 和 51b

DMA 控制器允许在 PCI 总线代理与本地存储器之间进行等待时间短、吞吐量高的数据传输。

有三个分离的 DMA 通道可用于数据传输。其中两个通道专门用于初



级 PCI 总线的数据传输、一个通道专门用于次级 PCI 总线的数据传输。DMA 控制器支持链接式数据传输和非对齐 (unaligned) 数据传输。只有通过本地处理器 34 才可被编程。

次级 PCI 仲裁单元 53

5 次级 PCI 仲裁单元为次级 PCI 总线提供 PCI 仲裁。它执行一个具有可编程优先权的公平算法。设有 6 个 PCI 请求和同意信号对。该仲裁单元可被锁闭，以允许进行外部仲裁。

内部 PCI 和本地总线仲裁单元 55a、55b、57

10 P2P 处理器中含有两个内部仲裁单元，控制对设备内的内部 PCI 总线的访问，它们是为初级桥接口、初级 ATU、DMA 通道 0、DMA 通道 1 进行仲裁的初级内部 PCI 仲裁单元 55a，为次级桥接口、次级 ATU、DMA 通道 2 进行仲裁的次级内部 PCI 仲裁单元 55 b。每个内部仲裁单元使用一种固定的轮式仲裁方案，总线上的每个设备具有相同的优先权。

15 P2P 处理器也要求有一种仲裁机制来控制本地总线的所有权。本地总线仲裁单元 (LBAU) 57 执行的公平算法，使每个总线主有获得本地总线控制权的机会。该算法综合采用了轮式仲裁方案和优先权机制。

I²C 总线接口单元 61

20 I²C 总线 (互连集成电路) 接口单元允许本地处理器担当驻留在 I²C 总线上的主控设备和从属设备的角色。I²C 总线是 Philips 公司开发的一种串行总线，由一个两针接口组成。该总线允许 P2P 处理器为了系统管理功能而与其它 I²C 外围设备和微控制器进行对接。它以最低的硬件要求为经济性的系统在 I/O 子系统上将状态信息和可靠性信息中转到外部设备。

APIC 总线接口单元 63

25 APIC 总线接口单元提供了对允许进行 I/O 高级可编程中断控制器 (APIC) 软件模拟的三线 APIC 总线的接口。可以在总线上发送中断报文，也可以接收 EOI 报文。

中断路由分配 67

提供了 4 个 PCI 中断输入，它们可被选择路由到本地处理器中断输入端，也可被选择路由到 PCI 中断输出接插引线。

30 术语与约定

表示数字

本文所有数字均为十进制，除非另加注明。文本中的十六进制数表示



为“nnnH”，其中的H表示十六进制。二进制数以下标2标注。

域

数据结构中的保留域，是处理器不使用的域。保留域可以被软件使用，处理器不改变这些域。

5 保留域是一种可以被一个实现所使用的域。如果保留域的初始值由软件提供，则该值必须为0。软件不得修改保留域或依赖保留域中的任何值。

只读域可被读取以返回当前值。对只读域的写操作将被作为空（no-op）操作对待，既不改变当前值，也不产生出错条件。

10 读/清除域也可被读取以返回当前值。以数值0对读/清除域进行写操作不会改变该域的内容；以数值1对读/清除域进行写操作会使该域的内容被清除（复位到0值）。例如，假设某读/清除域的值为F0H，用数据值55H写入，结果，域的内容变为A0H。

术语

为便于讨论P2P体系结构，使用了下列术语：

下游(Downstream) 位于或朝向编号较高（配置后的）的PCI总线

双字（DWORD） 32位数据字

主处理器 位于P2P处理器上游的处理器

本地总线 本地处理器总线

本地存储器 本地总线上的存储器子系统

上游 位于或朝向（配置后）编号较低的PCI总线

15 附图简要说明

附图1是表示使用现有技术PCI至PCI桥的一个系统的方框图。

附图2是表示使用按照本发明方法发明的带处理器的PCI至PCI桥的一个系统的方框图。

附图3是表示所发明的P2P处理器的方框图。

20 附图4表示PCI至PCI桥的初级地址空间与次级地址空间之间事务处理流的方向。

附图5是表示PCI至PCI桥的方框图。

附图6是表示初级地址翻译单元和次级地址翻译单元的方框图。

附图7是表示直接存储器访问控制器与各种总线的连接的方框图。

25 附图8是存储器控制器的方框图。

附图9是表示本地处理器与PPIC之间的连接的方框图。



附图 10 是表示内部 PCI 仲裁单元的连接方框图。

附图 11 是表示 I²C 总线接口单元及其与本地总线的接口的方框图。

发明的详细说明

现在就按附图 3 中的功能方框对本发明加以说明。

5 本地处理器

下面描述在 P2P 处理器中用作本地处理器的 80960 JF 微处理器。文中表述了 80960 JF 处理器的配置方法和与《i960 Jx 微处理器用户手册》中的描述不同的部分。

概述

10 80960 JF 微处理器在 P2P 处理器中的实现无需作功能性的改动，即不改变内部逻辑。有关 80960 JF 微处理器的更详细资料请参阅《i960 Jx 微处理器用户手册》。

性质

80960 JF 微处理器有如下基本性质：

- 15 · 高性能指令执行内核
- 4K 字节的两路组相联 (2-way set associative) 指令高速缓冲存储器
- 2K 字节的直接映射数据高速缓冲存储器
- 32 个 32 位整数寄存器
- 20 · 可编程总线控制器
- 1K 字节的内部数据随机存取存储器 (RAM)
- 本地寄存器高速缓冲存储器，它提供多达 8 个本地寄存器组的存储空间

空间

- 高级中断控制器
- 25 · 2 个 32 位定时器

差异点

以下描述系统设计中 P2P 处理器中使用的 80960 JF 微处理器有影响的决定。

存储器区域

30 因为 P2P 处理器的外围存储器映射寄存器宽度为 32 位，存储器区 0 和 1 必须被指定是一个 32 位的区域。因而，PMCON0_1 寄存器中必须设置总线宽度位为 102，指示 32 位宽的总线。



总线

为使 DMA 访问的性能最优，允许本地总线上本地处理器之外的总线主在本地处理器总线上具有无限制的脉冲串长度。然而，长度超过 4 个字的脉冲串的地址不作递增。这其中的含义是，本地总线上存储器控制器必须对脉冲串中的每个访问递增地址。

PCI 至 PCI 桥单元

概述

PCI 至 PCI 桥单元 32 是一个允许扩展 PCI 总线至其 10 个电气 PCI 负荷的物理限度以上的设备。该桥单元使用分层体系总线的概念，分层体系总线中的每个总线是一个电独立的单位，但分层体系内的所有总线逻辑上构成一个总线。PCI 至 PCI 桥单元不增加 PCI 总线的带宽，它只允许为需要超过电力标准允许数目的 I/O 部件的应用对总线进行扩展。

PCI 至 PCI 桥单元提供：

- 支持任一方向上的并发操作的独立的 32 位初级 PCI 总线和次级 PCI 总线；
- 位于桥的次级端的分离的存储器和 I/O 地址空间；
- 用于上游、下游事务处理的 2 个 64 字节投递缓冲器（posting buffers）；
- VGA 调色板探测（palette snooping）和次级总线上 VGA 兼容的寻址；
- 从次级 PCI 接口的 64 位寻址模式；
- 专用设备配置及次级 PCI 总线上专用 PCI 设备的地址空间；
- 允许在初级接口和次级接口上正解码（positive decoding）的特别操作模式。

操作原理

桥单元起着初级 PCI 总线和次级 PCI 总线之间的地址过滤器的作用。

PCI 支持 3 种分离的地址空间：

- 4G 字节的存储器地址空间
- 64K 字节的 I/O 地址空间（16 位寻址）
- 单独的配置空间

PCI 至 PCI 桥以存储器和 I/O 地址空间内的连续的地址范围编程，这些存储器和 I/O 地址空间然后变成次级 PCI 地址空间。出现在其范围在程



序设置的次级空间内的桥的初级端上任何地址均被从初级端传递到次级端，而范围在该次级空间之外的地址则被桥忽略。桥的次级端的工作与初级端相反，如附图 4 所示，它忽略程序设置的次级空间范围内的地址，而将该次级空间之外的所有地址传递到初级端。

5 PCI 桥的初级和次级接口各自实现符合 PCI 2.1 标准的主和目标设备。在桥的一端被启动的 PCI 事务处理将把启动事务处理的总线桥接口当作目标，该事务处理将由按主设备操作的该目标总线接口完成。桥对初级端和次级端上的 PCI 设备都是透明的。

P2P 处理器的 PCI 至 PCI 桥单元至少具有《PCI 至 PCI 桥体系结构规范》（1.0 版）和《PCI 本地总线规范》（2.1 版）中所要求的性质。以下说明桥的功能，适当时要引用《PCI 至 PCI 桥和 PCI 总线规范》的内容。

体系结构说明

PCI 至 PCI 桥单元逻辑上可划分以下 4 个主要部分：

- 初级 PCI 接口
- 15 · 次级 PCI 接口
- 投递缓冲器
- 配置寄存器

附图 5 中桥的框图显示了这些主要功能单元。

初级 PCI 接口

20 PCI 至 PCI 桥的初级 PCI 接口 71 既可以作为 PCI 总线事务处理的目标也可以作为 PCI 总线事务处理的启动器。在大多数系统中，初级接口与一般来说是系统分层体系中最低编号 PCI 总线的主机/PCI 桥的 PCI 侧相连。初级接口由 50 个在《PCI 至 PCI 桥体系结构规范》（1.0 版）中定义的必须信号接插引线和 4 个可选中断接插引线组成。

25 初级接口既实现启动器（主）又实现目标（从）PCI 设备。当次级总线上有事务处理被启动时，初级主状态机（the primary master state machine — 《PCI 本地总线规范》（2.1 版）对其有描述）就如同启动设备一样完成该事务处理（读或写）。初级 PCI 接口作为需要在次级总线上完成的事务处理的 PCI 目标设备，接收事务处理并将请求传递到次级端。

30 作为目标设备，初级 PCI 接口用正解码（positive decoding）要求在桥的下方提出的 PCI 事务处理，然后将该事务处理传递到次级主接口（secondary master interface）上。



初级 PCI 接口负责所有命令解释、地址解码及出错处理。

初级接口与次级接口的 PCI 配置、中断路由选择逻辑（下文描述）、次级 PCI 总线仲裁是通过初级接口完成的。配置空间寄存器支持这些功能。

5 次级 PCI 接口

PCI 至 PCI 桥单元的次级 PCI 接口 73 的作用方式几乎与初级接口的一样。它既包括一个 PCI 主设备也包括一个 PCI 从设备，并用系统使用的一组新的 PCI 电气负载实现“第二” PCI 总线。次级 PCI 接口由 49 个必须信号接插引线组成。S_RST#是次级端上的一个输出而不是输入。

10 次级 PCI 接口作为从（目标）设备，负责要求在桥的次级存储器或 I/O 地址空间中安置不下的 PCI 事务处理，将这些事务处理向上传递过桥，给初级端的主设备。次级 PCI 接口作为主设备（启动器），负责完成在桥的初级侧上启动的事务处理。次级 PCI 接口使用桥地址寄存器的逆解码（inverse decoding），只将范围在初级地址空间内的地址传递过桥。

15 次级 PCI 接口还为次级总线上的专用 PCI 设备实现一个单独的地址空间，这时次级 PCI 接口忽略并且不传递由本地处理器在配置时定义的初级地址范围。

作为一种特殊的操作模式，次 CPI 接口根据其自有的存储器和 I/O 地址寄存器执行正地址解码。这种操作模式是通过二级解码选通寄存器（SDER）得以实现的，具有锁闭次级接口上标准桥地址寄存器的逆解码的副作用。

投递缓冲器

25 为了对桥的对面隐盖在读和写事务处理期间仲裁 PCI 目标和获得 PCI 目标中产生的迟滞，PCI 至 PCI 桥单元实行两个 64 字节的投递缓冲器 77 和 79。桥既支持延迟事务处理（Delayed transactions）又支持投递事务处理（Posted transactions）。

延迟事务处理中，完成事务处理所需的信息被锁存，事务处理被重试信号中止。然后，桥就代表启动器执行事务处理。要求启动器重复被用重试信号中止的原始事务处理，以便完成该事务处理。

30 投递事务处理中，允许事务处理在目标总线上完成之前，在启动总线上先行完成。

下面详细讨论延迟事务处理和投递事务处理。



桥使用两个投递缓冲器:

- 用于寄存自初级接口流向次级接口的数据的下游投递缓冲器 77;
- 用于寄存自次级接口流向初级接口的数据的上游投递缓冲器 79.

各缓冲器均有关联的地址/控制寄存器, 用来保存有关事务处理的信息。

配置寄存器

每个 PCI 设备实现一个独立的配置地址空间及配置寄存器 81。桥配置首部格式的前 16 个字节实现所有 PCI 设备所要求的公用配置寄存器。只读首部类型寄存器中的值定义了首部内其余 48 个字节的格式, 对 PCI 至 PCI 桥返回 01H。

初级总线上的设备只能以 0 类型的配置命令访问 PCI 至 PCI 桥的配置空间。次级 PCI 总线上的设备不能以 PCI 配置周期访问桥的配置空间。配置寄存器中存放着桥的两边均需要的地址译码、出错条件和状态信息。

地址解码

P2P 处理器提供 3 个独立的地址区间, 它们被用于确定以任一方向通过 P2P 处理器的桥部分被发送的是哪些存储器和 I/O 地址。用于存储器事务处理的地址区间有 2 个, 用于 I/O 事务处理的地址区间有 1 个。桥用一个基地址寄存器和界限寄存器来实现一个地址区间。在初级接口上的地址区间被正解码, 将区间内的任何地址认为是次级地址, 因此能被向下游发送过桥。在次级接口上的地址区间被逆解码。这意味着程序设置的地址区间之外的任何地址均能被向上游发送过桥。

标准桥单元地址解码也能被次级解码选通寄存器 (SDER) 修改。这个寄存器内的数据位允许次级桥接口的正地址解码, 禁止 PCI 至 PCI 桥所使用的基本逆地址解码。

I/O 地址空间

PCI 至 PCI 桥单元实现一个用于 PCI I/O 事务处理的可编程地址区间。桥配置空间中的 I/O 基址寄存器 (IOBR) 和 I/O 界限寄存器 (IOLR) 定义一个连续的 I/O 地址空间。IOBR 的高 4 位对应 I/O 地址的 AD[15:12], 低 12 位总是 000H, 促使 I/O 地址空间的 4K 字节对齐。IOLR 的高 4 位也对应 I/O 地址的 AD[15:12], 低 12 位为 FFFH, 促使粒度为 4K 字节。

桥单元将从初级接口向次级接口发送一个具有在 IOBR 和 IOLR (含)



定义地址区间之内的地址的 I/O 事务处理，在这种情况下，对于该桥接 I/O 事务处理来说，初级接口起着 PCI 目标的作用，次级接口起着 PCI 启动器的作用。

5 次级总线上出现一个 I/O 读或写事务处理时，如果事务处理的地址在 IOBR 和 IOLR 定义地址区间之外，桥单元就将该事务处理发送至初级接口。在这种情况下，次级接口起着 PCI 目标的作用，初级接口起着 PCI 启动器的作用。

10 P2P 处理器只支持 16 位的地址用于 I/O 事务处理，因此，地址大于 64K 字节的 I/O 事务处理既不会被发送到初级接口也不会被发送到次级接口。即使数据位 AD[31:16]没有在 IOBR 和 IOLR 中被实现，桥也要假设 AD[31:16]=000H。桥单元仍然必须在 I/O 事务处理期间执行一个完整的 32 位译码操作，按《PCI 本地总线规范》检验 AD[31:16]=000H。

ISA 模式

15 P2P 设备的 PCI 至 PCI 桥单元在桥控制寄存器（BCR）中实现一个 ISA 模式位，用于为从属 PCI 总线上的 ISA I/O 卡提供 ISA 指示。ISA 模式只影响 IOBR 和 IOLR 寄存器定义地址区间之内的那些 I/O 地址。当通过对 ISA 模式位置位启动 ISA 模式时，对于地址在各个自然对齐的 1K 字节块的高 768 字节（300H）的 I/O 事务处理，桥将其滤除而不发送；与其相反，次级总线上的 I/O 事务处理则逆解码这些 ISA 地址，因而发送地址在 20 各个自然对齐的 1K 字节块的高 768 字节（300H）的 I/O 事务处理。

存储器地址空间

PCI 至 PCI 桥单元支持两个用于从初级接口向下游发送存储器访问至次级接口的独立地址区间。存储器基址寄存器（MBR）和存储器界限寄存器（MLR）定义一个地址区间，可预取存储器基址寄存器（PMBR）和可预取存储器界限寄存器（PMLR）定义另一个地址区间。可预取地址 25 区间用于确定哪些存储器空间可以预取而无副作用。这两对寄存器都确定桥何时将发送“存储器读”、“存储器读行”（Memory Read Line）、“存储器读多次”（Memory Read Multiple）、“存储器写”、“存储器写及无效”事务处理过桥。如果两对寄存器重叠，则产生一个地址区间， 30 该地址区间是与优先于桥读事务处理响应的可预取区间组合的两个寄存器的和。

寄存器 MBR、MLR、PMBR、PMLR 的高 12 位对应主存储器地



址或次级存储器地址的地址位 D[31:20]。为译码目的，桥假设两个存储器基址寄存器的 AD[19:0]均为 00000H，两个存储器界限寄存器的 AD[19:0]均为 FFFFFH。这促使由桥单元支持的各存储器地址区间被对齐在 1M 字节的边界上并具有 1M 字节的大小粒度（size granularity）。所有 4 个寄存器的低 4 位均为只读型，被读时返回零。

地址范围处于两对寄存器（MBR-MLR 与 PMBR-PMLR）所定义地址区间之内的任何 PCI 存储器事务处理（而不是 I/O 事务处理），均被从初级接口向下游发送过桥至次级接口。在曾被初级总线上初级从接口要求（claimed）的次级总线上，次级主接口（secondary master interface）总是使用相同的命令类型（在“存储器写及无效”事务处理期间的某些情况除外）。所有的双地址周期（dual address cycles）（64 位地址的事务处理）总是被次级接口要求。

地址范围处于两对寄存器（MBR-MLR 与 PMBR-PMLR）所定义地址区间之外的任何 PCI 存储器事务处理，均被从次级接口向上游发送过桥至初级接口。次级接口从次级总线将所有的双地址周期发送到初级总线。双地址周期被限制在 64 位地址空间的高 4G 字节。

通过桥配置空间中的下列寄存器位可修改桥对初级、次级接口中任一接口的响应：

- 初级命令寄存器（PCMD）中的主启动位
- 初级命令寄存器（PCMD）中的存储器启动位
- 桥控制寄存器（BCR）中的 VGA 启动位
- 次级解码启动寄存器（SDER）中的次级正存储器解码启动位

次级解码启动寄存器（SDER）中的次级正存储器解码启动位修改次级地址解码。它启动一对地址区间寄存器、次级存储器基址寄存器（SMBR）与次级存储器界限寄存器（SMLR），这些寄存器为次级总线上要求存储器事务处理及发送过桥而定义一个地址窗。该解码和事务处理要求的工作方式与 MBR/MLP 和 PMBR/PMLR 地址对在初级总线上的正解码方式一样。次级正存储器解码启动位也锁闭在要求地址范围超出 MBR/MLP 和 PMBR/PMLR 规定地址区间的存储器事务处理的次级接口上执行的逆解码。在代表 MBR/MLP 和 PMBR/PMLR 地址对的初级接口上不执行逆解码。

64 位地址解码—双地址周期

桥单元仅支持在桥单元的次级接口上进行 64 位寻址的双地址周期命令。双地址周期允许通过使用两个 PCI 地址阶段 (phases) 进行 64 位寻址—第一阶段为低 32 位, 第二阶段为高 32 位。

5 一般来说, 桥单元解码所有双地址周期并将其从初级接口发送至次级接口, 而不管在寄存器对 MBR/MLP 和 PMBR/PMLR 中定义的地址区间如何。如果 SDER 寄存器中的次级差解码启动位置位, 双地址周期将不被发送。

桥单元用减解码定时 (在 FRAME# 被确立的 5 个时钟之后确立 DELSEL#) 要求双地址周期。这就使得次级总线上的其它代理能在桥单元之前要求双地址周期。

初级接口不发送双地址周期。

保存和发送 64 位地址的高 32 位的机制是将与次级至初级数据路径关联的 32 位地址寄存器相加。这些寄存器存储双地址周期的第二地址阶段期间传输的 64 位地址的高 32 位。此外, 主和从状态机必须能够支持双地址周期和 DAC 命令。

通过桥配置空间中的下列寄存器位可修改对次级接口上 DAC 周期的响应:

- 初级命令寄存器 (PCMD) 中的主启动位
- 初级命令寄存器 (PCMD) 中的存储器启动位

20 要允许桥使桥能够响应无论 32 位还是 64 位的任何种类的存储器周期, PCMD 寄存器中的存储器启动位必须被置位。

要允许初级接口对主 PCI 事务处理, PCMD 寄存器中的主启动位必须被置位。

桥操作

25 P2P 处理器的桥单元能够从一个 PCI 接口向另一个 PCI 接口发送所有类型的存储器、IO 和配置命令。表 1 中定义了 PCI 至 PCI 桥单元及其两个 PCI 接口所支持和所不支持的 PCI 命令。PCI 命令是编码在任一接口的 C/BE [3 : 0] # 插接引线上的。为避免两个接口的不同所引起的死锁, 当两个接口上同时出现事务处理时, 桥将优先权赋予初级接口。

表 1. PCI 命令

C/BE#	PCI 命令	启动器: 初级总线 目标: 次级总线	启动器: 次级总线 目标: 初级总线
0000 ₂	中断确认	不问	不问
0001 ₂	特殊周期	不问	不问
0010 ₂	I/O 读	发送	发送
0011 ₂	I/O 写	发送	发送
0100 ₂	保留	不问	不问
0101 ₂	保留	不问	不问
0110 ₂	存储器读	发送	发送
0111 ₂	存储器写	发送	发送
1000 ₂	保留	不问	不问
1001 ₂	保留	不问	不问
1010 ₂	配置读	发送	发送
1011 ₂	配置写	发送	发送
1100 ₂	存储器读多次	发送	发送
1101 ₂	双地址周期	不问	不问
1110 ₂	存储器读行	发送	发送
1111 ₂	存储器写与无效	发送	发送

PCI 接口

P2P 的桥单元有一个初级 PCI 接口和一个次级 PCI 接口。当事务处理在初级总线上被启动并由桥要求时，初级接口就起着 PCI 目标设备的作用，次级接口则起着对次级总线上真正的 PCI 目标的启动设备的作用。初级总线为启动总线，次级总线为目标总线。该序列对于次级总线上启动的事务处理被颠倒。对各接口定义如下：

初级接口

桥单元的初级 PCI 接口 71 是连接到 P2P 设备桥接的两个 PCI 总线之间编号较低的 PCI 总线的接口。

初级 PCI 接口必须遵守《PCI 本地总线规范》和《PCI 至 PCI 桥体系结构规范》中规定的对 PCI 主和从设备的定义。

次级接口



桥单元的次级 PCI 接口 73 是连接到 P2P 设备桥接的两个 PCI 总线之间编号较高的 PCI 总线的接口。

次级 PCI 接口必须遵守《PCI 本地总线规范》和《PCI 至 PCI 桥体系结构规范》中规定的对 PCI 主和从设备的定义。

5 投递缓冲器

PCI 至 PCI 桥有两个投递缓冲器用于延迟事务处理和投递事务处理。下游投递缓冲器 77 位于从初级接口至次级接口的数据路径中；上游投递缓冲器 79 位于从次级接口至初级接口的数据路径中。附图 5 表示了位于初级接口和次级接口之间的两个投递缓冲器。

10 使用下游投递缓冲器的有：

- 初级总线的“投递写”
- 初级总线的“延迟写请求”
- 返回到次级总线的“延迟读完成”
- 返回到次级总线的“延迟写完成”

15 使用上游投递缓冲器的有：

- 次级总线的“投递写”
- 次级总线的“延迟写请求”
- 返回到初级总线的“延迟读完成”
- 返回到初级总线的“延迟写完成”

20 写投递允许桥达到其全部带宽潜力，与此同时又掩盖了穿越桥所关联的等待和获取目标总线所关联的等待。两组投递缓冲器可同时被使用。

投递缓冲器的组织

每个投递缓冲器能存放组织成 4 字节长度的 16 项数据（16 个双字）的 64 个字节。每个缓冲器能存放：

- 25
- 多达 64 字节的一个“投递写”事务处理，或者
 - 多达 64 字节的一个“延迟完成”事务处理，或者
 - 多达 4 字节的一个“延迟写”事务处理

每个投递缓冲器与一个地址寄存器和一组标记位（tag bits）及有效位（valid bits）相关联。

30 桥也可以在投递缓冲器之外存储一个“延迟读请求”。

投递缓冲器的内部寻址以一种循环方式进行，其方法是，当有事务处理进入一个空缓冲器时，它立即被发送到缓冲器的顶部。移动缓冲器中的



一项数据到下一项不需要 PCI 时钟。

投递缓冲器的操作

两个投递缓冲器都被用于帮助桥达到其全部的带宽潜力并掩盖为每个穿越桥的事务处理获取两个总线所产生的等待。要允许缓冲器投递事务处理，EBCR 寄存器中的“投递禁止”位必须被清零。

投递缓冲器的性质为从初级接口到次级 PCI 接口和从次级接口到初级 PCI 接口的同时操作创造了条件。这就是说，两个 PCI 接口上向对面接口的事务处理可以同时发生。从事务处理被启动到桥的时刻起，目标接口就试图获得对目标总线的主控权。这所用的机制就是在初级接口和次级接口上所用的标准 PCI 仲裁机制。

作为缺省的复位状态，投递缓冲器将被标记为无效。随后的任何 PCI 复位事件都会迫使所有缓冲器通过被标记无效而被清零。

事务处理定序规则

因为桥能处理多重事务处理，所以必须进行适当的定序以避免发生死锁条件，提高通量。表 2 中的内容是多重事务处理的定序规则。表中第 1 栏是刚刚被锁存的事务处理。该表指出了新的事务处理能超越（pass）上一个被接受的事务处理（用“是”表示）、新的事务处理不能超越上一个被接受的事务处理（用“非”表示）、新的事务处理不应被接受（用“不接受”表示）等情形。未被接受的事务处理，应当向其发送一个 Retry 信号。

表 2. 事务处理的超越

超越 ?	超越被 接受投递 存储器写?	超越被 接受延迟 读请求?	超越被 接受延迟 写请求?	超越被 接受延迟 读完成?	超越被 接受延迟 写完成?
新的 投递存储器写	非	是	是	是	是
新的 延迟读请求	非	不接受	不接受	非	是
新的 延迟写请求	非	不接受	不接受	非	是
新的 延迟读完成	非	是	是	不接受	不接受
新的 延迟写完成	是	是	是	非	不接受

寄存器定义

以下描述 PCI 至 PCI 桥配置寄存器。配置空间由按预定格式安排的 8 位、16 位、24 位、32 位寄存器组成。对配置寄存器的访问要通过桥的初级
5 端上的 0 类型的配置读与写和本地处理器的本地操作。

下文详细介绍了除《PCI 本地总线规范》和《PCI 至 PCI 桥体系结构规范》定义的寄存器之外的每个寄存器的功能、访问类型（读/写、读/清零、只读）以及复位缺省条件。前文已说过，要求初级端上带活跃的 IDSEL 或存储器映射的本地处理器访问的 0 类型配置命令对这些寄存器进行读或
10 写。对位移量达 3EH 的寄存器的格式，《PCI 至 PCI 桥体系结构规范》（1.0 版）中有所定义，因此本文不再赘述。位移量大于 3EH 的寄存器特定于 P2P 处理器的具体实现。

要允许本地处理器访问桥配置空间，还有另外一个要求。有些只被 0 类型配置读写命令读取的寄存器是由本地处理器可写的寄存器。这允许在
15 PCI 配置开始之前对某些配置寄存器进行初始化。

本地处理器按照存储器映射寄存器读写桥配置空间。表 3 显示了寄存器及其在 PCI 配置命令中使用的相关位移量以及其在本地处理器地址空间中的存储器映射地址。

在桥的初级端上确立 P_RST#信号影响桥配置空间中含有的大多数寄



寄存器的状态。除非另加注明，否则，当进行初级复位时，所有数据位及寄存器都要返回到其业已声明的缺省状态值。除非明确注明，否则次级 S_RST#输出的复位状态不影响寄存器的状态。

表 3. PCI 至 PCI 桥配置寄存器地址

寄存器名	大小 (字节数)	地址位移量
销售商标识符寄存器—VIDR	2	00H
设备标识符寄存器—DIDR	2	02H
初级命令寄存器—PCMDR	2	04H
初级状态寄存器—PSR	2	06H
修订标识符寄存器—RIDR	1	08H
类别代码寄存器—CCR	3	09H
高速缓冲存储器行大小寄存器—CLSR	1	0CH
初级等待定时器寄存器—PLTR	1	0DH
首部类型寄存器—HTR	1	0EH
初级总线号寄存器—PBNR	1	18H
次级总线号寄存器—SBNR	1	19H
下属总线号寄存器—SubBNR	1	1AH
次级等待定时器寄存器—SLTR	1	1BH
I/O 基址寄存器—IOBR	1	1CH
I/O 界限寄存器—IOLR	1	1DH
次级状态寄存器—SSR	2	1EH
存储器基址寄存器—MBR	2	20H
存储器界限寄存器—MLR	2	22H
可预取存储器基址寄存器—PMBR	2	24H
可预取存储器界限寄存器—PMLR	2	26H
桥控制寄存器—BCR	2	3EH
扩展的桥控制寄存器—EBCR	2	40H
次级 IDSEL 选择寄存器—SISR	2	42H
初级桥中断状态寄存器—PBISR	4	44H
次级桥中断状态寄存器—SBISR	4	48H
次级仲裁控制寄存器—SACR	4	4CH



PCI 中断路由选择寄存器— PIRSR	4	50H
次级 I/O 基址寄存器— SIOBR	1	54H
次级 I/O 界限寄存器— SIOLR	1	55H
次级存储器基址寄存器— SMBR	2	58H
次级存储器界限寄存器— SMLR	2	5AH
次级解码启动寄存器— SDER	2	5CH

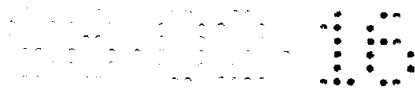
如前文所述，从销售商标识符寄存器到桥控制寄存器_BCR 中的各数据位遵守《PCI 本地总线规范》中的定义，因此本文不必叙述。下文描述的对象是按照本发明为实现 PCI 至 PCI 桥而添加到《PCI 本地总线规范》的寄存器。如表 3 所示，添加的寄存器的地址位移量从 40H 开始。

5 扩展的桥控制寄存器— EBCR

扩展的桥控制寄存器用于控制桥所执行的《PCI 至 PCI 桥体系结构规范》中规定的基本功能以外的扩展功能。寄存器中有启动位和锁闭位，用于控制桥的扩展功能。

表 4a 扩展的桥控制寄存器— EBCR

位	缺省值	读/写	描述
15:07	00000000 ₂	只读	保留
06	随初级 PCI 总线复位时 CONFIG_MODE 接插引线的外部状态而变化	读/写	配置周期禁止—当该位被置位时，P2P 处理器的初级 PCI 接口用“Retry”条件响应所有的配置周期。当该位被清零时，P2P 处理器响应恰当的配置周期。该位的缺省条件根据在 P_RST# 上升边沿的 CONFIG_MODE 接插引线的外部状态而定。如果接插引线的外部状态是高电平，该位被置位；如果接插引线的外部状态是低电平，该位被清零。
05	0 ₂	只读	保留
04	0 ₂	只读	保留
03	随初级 PCI 总线复位时 SYNC#接插引线的	只读	SYNC#模式—描述下面的三个时钟中哪些是同步的：初级 PCI 总线、次



	外部状态而变化		级 PCI 总线、本地处理器。如果被清零，所有三个时钟同步；如果被置位，初级 PCI 总线时钟与次级 PCI 总线时钟及本地处理器时钟异步。该位的缺省值根据在 P_RST# 上升边的 SYNC# 接插引线的外部状态而定。
02	0 ₂	读/写	复位桥—当该位被置位，整个 PCI 至 PCI 桥将被复位。桥的所有寄存器被设置到其缺省值（BCR 中的次级总线复位位除外），所有状态机被复位，所有缓冲器被清除。BCR 中的次级总线复位位在保证最短 PCI 复位时间内被置位。要清除该复位位以取消（deassert）次级总线复位的确立，需要使用软件。
01	随初级 PCI 总线复位时 RST_MODE 接插引线的外部状态而变化	读/写	处理器复位—该位只复位本地处理器而不复位桥的次级端。置位该位将置处理器于复位状态并保持该状态。要使该位清零以取消本地处理器复位，需要使用软件。该位的缺省条件根据在 P_RST# 上升边沿的 RST_MODE 接插引线的外部状态而定。如果接插引线的外部状态是高电平，该位被置位；如果接插引线的外部状态是低电平，该位被清零。
00	0 ₂	读/写	投递锁闭—如果该位置位，则不允许桥无论从哪个桥接口投递写事务处理。所有事务处理被作为延迟事务处理处理。如果该位清零，则允许桥投递写事务处理。



初级桥中断状态寄存器用于向本地处理器告知初级桥接口中断的中断源。此外，对该寄存器进行写操作可以清除向 P2P 处理器的中断单元的中断的中断源。该寄存器所有数据位对 PCI 是只读的，对本地总线是读/清零的。

5 位 4:0 (分别) 是初级状态寄存器的第 8 位和位 14:11 的直接反映 (这些位由硬件同时设置, 但必须单独清除)。通过向该寄存器中适当的某些位写入 1, 就可以清除产生初级桥中断的条件。

表 4b 初级桥中断状态寄存器— PBISR

位	缺省值	读/写	描述
31:05	0000000H	只读	保留
04	02	读/清零	P_SERR#被确立—如果初级 PCI 总线上确立 P_SERR#, 则该位被置位。
03	02	读/清零	PCI 主流产—每当由初级主接口启动的事务处理以主-流产方式结束时, 该位被置位。
02	02	读/清零	PCI 目标流产(主)—每当由初级主接口启动的事务处理以主-流产方式结束时, 该位被置位。
01	02	读/清零	PCI 目标流产(目标)—每作为目标的初级主接口, 以目标流产方式停止 PCI 总线上的事务处理时, 该位被置位。
00	02	读/清零	PCI 主奇偶校验出错—初级接口在符合以下三个条件时将该位置位: 1)总线代理自己确立 P_SERR#或探测到 P_SERR#已被确立 2)为该位置位的该代理在出错的操作中担当了总线主的角色 3)(命令寄存器中的)奇偶校验错误响应位被置位

次级桥中断状态寄存器— SBISR

10 次级桥中断状态寄存器用于向本地处理器告知次级桥接口中断的中断源。此外, 对该寄存器进行写操作可以清除对 P2P 处理器的中断单元的中断源。

断的中断源。该寄存器所有数据位对 PCI 是只读的，对本地总线是读/清零的。

5 位 4:0（分别）是次级状态寄存器的第 8 位和位 14:11 的直接反射（这些位由硬件同时设置，但必须单独清除）。通过向该寄存器中适当的某些位写入 1，就可以清除产生次级桥中断的条件。

表 4c 次级桥中断状态寄存器—PBISR

位	缺省值	读/写	描述
31:05	0000000H	只读	保留
04	02	读/清零	P_SERR#确立 (asserted) — 如果次级 PCI 总线上 P_SERR#确立，则该位被置位。
03	02	读/清零	PCI 主流产 — 每当由次级主接口启动的事务处理以主-流产方式结束时，该位被置位。
02	02	读/清零	PCI 目标流产 (主) — 每当由次级主接口启动的事务处理以主-流产方式结束时，该位被置位。
01	02	读/清零	PCI 目标流产 (目标) — 每当作为目标的次级主接口，以目标流产方式停止 PCI 总线上的事务处理时，该位被置位。
00	02	读/清零	PCI 主奇偶校验出错 — 次级接口在符合以下三个条件时将该位置位： 1) 总线代理自己确立 P_SERR#或探测到 P_SERR#已被确立 2) 为该位置位的该代理在出现错误的该操作中担当总线主的角色。 3) (命令寄存器中的) 奇偶校验错误响应位被置位

次级 IDSEL 选择寄存器—SISR

10 次级 IDSEL 选择寄存器控制从初级接口到次级接口的类型 1 至类型 0 转换中的 S_AD[20:16]的使用。在缺省条件下的操作中，类型 1 至类型 0 转换期间，对地址 P_AD[15:11]一种独特编码，会导致在次级地址总线 S_AD[31:16]上确立一个数据位。这被用于在由类型 0 配置命令设定为目标的设备上确立 IDSEL。这个寄存器允许用次级地址位 S_AD[20:16]来配置

专用 PCI 设备，而不管初级地址 P_AD[15:11]（类型 1 配置命令中的设备号）的状态如何，具体方法是在类型 1 至类型 0 转换期间，将次级地址位 S_AD[20:16]全部清零。

5 如果 S_AD[20:16]内有任何地址位要被用于专用次级 PCI 设备，本地处理器必须保证在主机试图配置分层结构的 PCI 总线之前，将 SISR 寄存器中的对应位置位。

表 4d 次级 IDSEL 选择寄存器— SISR

位	缺省值	读/写	描述
04	0 ₂	读/写	AD20 - IDSEL 禁止 - 如果该位置位，对任何可能的类型 1 至类型 0 转换，AD20 将被撤销。如果该位清零，AD20 在类型 1 至类型 0 转换期间当初级地址[15:11]=00100 ₂ 时被确立。
03	0 ₂	读/写	AD19 - IDSEL 锁闭 - 如果该位置位，对任何可能的类型 1 至类型 0 转换，AD19 将被撤销。如果该位清零，AD19 在类型 1 至类型 0 转换期间当初级地址[15:11]=00100 ₂ 时被确立。
02	0 ₂	读/写	AD18 - IDSEL 锁闭 - 如果该位置位，对任何可能的类型 1 至类型 0 转换，AD18 将被撤销。如果该位清零，AD18 在类型 1 至类型 0 转换期间当初级地址[15:11]=00100 ₂ 时被确立。
01	0 ₂	读/写	AD17 - IDSEL 锁闭 - 如果该位置位，对任何可能的类型 1 至类型 0 转换，AD17 将被撤销。如果该位清零，AD17 在类型 1 至类型 0 转换期间当初级地址[15:11]=00100 ₂ 时被确立。
00	0 ₂	读/写	AD16 - IDSEL 锁闭 - 如果该位置位，对任何可能的类型 1 至类型 0 转换，AD16 将被撤销。如果该位清零，AD16 在类型 1 至类型 0 转换期间当初级地址[15:11]=00100 ₂ 时被确立。

次级仲裁控制寄存器— SACR

10 次级仲裁控制寄存器用于设置使用次级 PCI 总线的各个设备的仲裁优先级。写一个值设置仲裁优先级，读寄存器则返回编程的值。每个设备被赋予一个 2 位的优先级值。优先级如表 4e 所示。

表 4e 编程的优先级控制

2 位编程值	优先级
00 ₂	高优先级
01 ₂	中优先级
10 ₂	低优先级
11 ₂	禁止

SACR 寄存器中也有用于次级总线仲裁单元的次级仲裁器启动位。该位清零时，次级总线仲裁器被禁止，桥将在 S_GNT0#上驱动 S_REQ0#，并在 S_REQ0#上抽样 S_GNT #。缺省状态是内部次级仲裁单元启动（次级仲裁单元启动位置位）。

PCI 中断路由选择寄存器— PIRSR

以下参照 PCI 和外部中断控制器（PPIC）说明 PCI 中断路由选择寄存器。

次级 I/O 基址寄存器— SIOBR

当次级 PCI 接口被启动用于正解码时，要使用次级 I/O 基址中的位。次级 I/O 基址寄存器为用于确定何时从桥的次级接口向初级接口发送 I/O 事务处理的正解码地址区间定义底部地址（含底部地址）。该寄存器必须在次级解码启动寄存器（SDER）被置位之前用一个有效值编程。桥仅支持由寄存器的最低 4 位有效位为 0H 所表示的 16 位寻址方式。高 4 位被编程为地址区间的底部的 S_AD[15:12] 的值，基址的 S_AD[11:0] 总为 000H，促使次级 I/O 地址区间是 4K 字节对齐的。

为了地址解码的目的，桥假设 I/O 地址的高 16 位地址位 S_AD[31:16] 为 0。桥仍然必须按照《PCI 本地总线规范》对全部的 32 位地址进行地址解码，并查验高 16 位为 0000H。

正次级 I/O 地址区间（由 SIOBR 与 SIOLR 定义）不受桥控制寄存器（BCR）中的 ISA 启动位状态的影响。

表 4f 次级 I/O 基址寄存器— SIOBR

位	缺省值	读/写	描述
07:04	0H	读/写	次级 I/O 基址 - 该域被编程为有待从桥的次级端传送到初级端的正解码次级 I/O 地址区间的底部地址的 S_AD[15:12]。



03:00	0H	只读	I/O 寻址能力 - 0H 值表示桥只支持 16 位的 I/O 寻址。
-------	----	----	-------------------------------------

次级 I/O 界限寄存器 - SIOLR

当次级 PCI 接口被启动用于正解码时，要使用次级 I/O 界限寄存器中的位。次级 I/O 界限寄存器为用于确定何时从桥的次级接口向初级接口发送 I/O 事务处理的正解码地址区间定义高地址（含高地址）。该寄存器必须在桥命令寄存器中的 I/O 空间启动位和次级解码启动寄存器（SDER）中次级 I/O 解码启动位的被置位之前用一个大于或等于 SIOBR 的有效值编程。如果 SIOBR 的值大于 SIOLR 的值，则从次级接口发送到次级接口（正解码）的 I/O 周期（cycles）无定义。桥仅支持由寄存器的最低 4 位有效位为 0H 所表示的 16 位寻址方式。高 4 位被编程为 S_AD[15:12] 的值，作为地址区间的顶部地址。基址的 S_AD[11:0] 总为 0000H，促使 I/O 地址区间具有 4K 字节的粒度（granularity）。

为了地址解码的目的，桥假设 I/O 地址的高 16 位地址位 S_AD[31:16] 为 0。桥仍然必须按照《PCI 本地总线规范》对全部的 32 位地址进行地址解码，并查验高 16 位为 0000H。

（由 SIOBR 与 SIOLR 定义的）次级 I/O 地址区间不被桥控制寄存器（BCR）中的 ISA 启动位修改。

表 4g 次级 I/O 界限寄存器— SIOLR

位	缺省值	读/写	描述
07:04	0H	读/写	次级 I/O 界限地址 - 该域被编程为有待从桥的次级端传送到初级端的正解码次级 I/O 地址区间的顶部的 S_AD[15:12].
03:00	0H	只读	次级 I/O 寻址能力 - 0H 值表示桥只支持 16 位的 I/O 寻址。

次级存储器基址寄存器— SMBR

当次级 PCI 接口被启动用于正地址解码时，要使用次级存储器基址寄存器中的位。如果 SDER 中的专用地址空间启动位置位，也要用次级存储器基址寄存器中的位在次级 PCI 总线上定义一个专用地址空间。次级存储器基址寄存器为用于确定何时从桥的次级接口向初级接口发送事务处理的内存映射地址区间定义底部地址（含底部地址）。该寄存器必须在寄存器 SDER 中的次级正存储器解码启动位被置位之前用一个有效值编程。高 12 位对应 32 位地址的 S_AD[31:20]。为了地址解码的目的，桥假设存储器基址的低 20 位地址位 S_AD[19:0] 为 0。这意味被定义的地址区间的底部地址将被对齐在 1M 字节的边界上。

表 4h 次级存储器基址寄存器— SMBR

位	缺省值	读/写	描述
15:04	0H	读/写	次级存储器基址 - 该域被编程为有待从桥的次级端传送到初级端接口的正解码次级存储器地址区间的底部地址 S_AD[30:20].
03:00	0H	只读	保留

次级存储器界限寄存器— SMLR

当桥单元的次级接口被启动用于正地址解码时，要使用次级存储器界限寄存器中的位。次级存储器界限寄存器为用于确定何时从桥的次级接口向初级接口发送事务处理的内存映射地址区间定义顶部地址（含底部地址）。次级存储器界限寄存器必须在存储器空间启动位和次级正存储器解码启动位被置位之前用一个大于或等于 SMBR 的有效值编程。一旦存储器空间启动位和次级正存储器解码启动位被置位时，如果 SMLR 的值不是大于或等于 SMBR 的值，则从次级接口发送到初级接口的正解码事务处理不

确定。高 12 位对应 32 位地址的 S_AD[31:20]。为了地址解码的目的，桥假设存储器基址的低 20 位地址位 S_AD[19:0] 为 FFFFFH。这就促使在存储器地址区间产生 1M 字节的粒度。

表 4i 次级存储器界限寄存器— SMLR

位	缺省值	读/写	描述
15:04	000H	读/写	次级存储器界限地址 - 该域被编程为有待从桥的次级端传送到初级端的正解码次级存储器地址区间的顶部地址的 S_AD[30:20]。
03:00	0H	只读	保留

5 次级解码启动寄存器— SDER

次级解码启动寄存器用于控制桥单元的次级 PCI 接口上的地址解码功能。次级正 I/O 解码启动位一旦置位，就致使桥对由 SIOBR/SIOLR 地址对定义的地址区间内的事务处理进行解码，要求取得控制权并将事务处理发送过桥单元。次级存储器解码启动位虽然与次级正 I/O 解码启动位的功能相同，但只在由 SMBR/SMLR 地址对定义的地址区间内起作用。对这些寄存器位中无论哪一位置位，会禁止次级接口上的所有逆解码。

10

次级差解码启动位允许在次级接口上进行差桥解码，以支持初级接口上的标准总线扩展。该位允许次级接口上差解码的条件是，或者次级正 I/O 解码启动位置位，或者次级正存储器解码启动位置位。

15

专用 (private) 存储器空间启动位允许在次级 PCI 总线上创建专用存储器空间。该位与 SMBR/SMLR 寄存器对联合使用。如果该位置位，则对于地址在 SMBR/SMLR 地址区间内的事务处理，桥就不予处理。



表 4j 次级解码启动寄存器— SMLR

位	缺省值	读/写	描述
15:04	000000000000 ₂	读/写	保留
03	0 ₂	读/写	专用存储器空间启动 - 该位置位时, 禁止桥对 SMBR/SMLR 的地址区间内的地址进行发送, 这就在次级 PCI 总线上创建了一个允许进行对等到对等 (peer to peer) 事务处理的专用存储器空间。
02	0 ₂	读/写	次级差解码启动 - 该位置位时, 允许次级接口 (在 S_FRAME# 确立后 5 个时钟时) 使用差解码要求次级总线上的事务处理。 S_FRAME# 后第 4 个时钟时未被要求的任何事务处理均被次级接口在第 5 个时钟时要求并被发送到初级 PCI 接口。
01	0 ₂	读/写	次级正存储器解码启动 - 该位置位时, 允许桥单元的次级接口对次级总线上的存储器地址进行正解码, SMBR/SMLR 地址范围内的地址将被发送过桥, 逆解码被禁止。
00	0 ₂	读/写	次级正 I/O 解码选通 - 该位置位时, 允许桥单元的次级接口对次级总线上的 I/O 地址进行正解码, SIOBR/SIOLR 地址范围内的地址将被发送过桥, 逆解码被禁止。

地址翻译单元

以下说明初级、次级 PCI 总线与本地总线之间连接的方法。描述了接口的操作方式、建立和实现。

5 概述

P2P 处理器提供 PCI 总线与本地总线之间的接口。该接口由 2 个地址翻译单元 (ATU) 43a/43b 和 1 个报文传递 (messaging) 单元 45 组成。地址翻译单元既支持向内 (inbound) 地址翻译, 也支持向外 (outbound)



地址翻译。第一个地址翻译单元称为初级 ATU43a，它提供初级 PCI 总线与本地总线之间的直接访问。第二个地址翻译单元称为次级 ATU43b，它提供次级 PCI 总线与本地总线之间的直接访问。用这种方式使用两个 ATU 比现有技术具有明显的优点。

5 在向内事务处理期间，ATU 将（PCI 总线主启动的）PCI 地址转换为本地处理器的地址并启动本地处理器上的数据传递。在向外事务处理期间，ATU 将本地处理器的地址转换为 PCI 地址并启动各 PCI 总线上的数据传递。

这 2 个地址翻译单元与报文传递单元表现为初级 PCI 总线上的单一 PCI 设备。这几个单元结合起来就是多功能 P2P 处理器中的第二个功能单元。附图 6 中表示了地址翻译单元与报文传递单元。

下面说明地址翻译单元与报文传递单元的功能。所有被表示的单元都有一个内存映射的寄存器接口，无论对于 PCI 接口 91 或本地总线接口还是这二者同时来说，该接口是可见的。

15 地址翻译单元数据流

初级地址翻译单元和次级地址翻译单元从两个方向经过 P2P 处理器的事务处理。初级 ATU 允许初级 PCI 总线上的 PCI 主向本地处理器启动事务处理，也允许本地处理器向初级 PCI 总线启动事务处理。次级 ATU 执行同样的功能，不过这是对次级 PCI 总线和次级 PCI 总线主而言。我们把
20 在 PCI 总线上启动、目标是本地总线的事务处理称为向内事务处理，而将在本地总线上启动、目标是 PCI 总线的事务处理称为向外事务处理。

ATU 地址翻译

地址翻译单元执行一个地址开窗口方案（address windowing scheme）来决定要求与翻译哪些地址到适当的总线。

25 初级 ATU 中有一条连接初级 PCI 总线和本地总线的数据通路。以这种方式连接初级 ATU 使得无需用次级 PCI 总线上的资源就能传递数据，次级 ATU 中有一条连接次级 PCI 总线和本地总线的数据通路。次级 ATU 允许次级 PCI 总线主直接访问本地总线和存储器。这些事务处理由一个次级总线主启动，不需要占用初级 PCI 总线上的任何带宽。

30 ATU 单元能识别和生成多种类型的 PCI 周期。表 5 显示了向内 ATU 与向外 ATU 均支持的 PCI 命令。向内 ATU 所见的操作类型由启动事务处理的（初级总线或者次级总线上的）PCI 主所决定。要求向内事务处理取



决于地址在编程的向内翻译窗口内。向外 ATU 所能使用的操作类型由本地地址和固定的向外开窗口方案所决定。

表 5. ATU 命令支持

PCI 命令类型	对向内事务处理要求	由向外事务处理生成
I/O 读	否	是
I/O 写	否	是
存储器读	是	是
存储器写	是	是
存储器写和无效	是	否
存储器读行	是	否
存储器多次读	是	否
配置读	是	是
配置写	是	是
双地址周期	否	是

两个 ATU 都支持《PCI 本地总线规范》所规定的 64 位寻址扩展。64 位寻址扩展只用于向外数据事务处理（即由本地处理器启动的数据传递）。

向内地址翻译

地址翻译单元提供了使 PCI 总线主能直接访问本地总线的机制。这些 PCI 总线主能对 P2P 处理器内存映射寄存器或本地存储器空间进行读或者写。PCI 总线主正在访问本地总线的事务处理称作向内事务处理。

向内地址翻译涉及两个步骤：

1. 地址查验

- 确定 32 位 PCI 地址是否在为向内（初级或次级）ATU 定义的地址窗口范围内。

- 用快速 DEVSEL# 定时要求 PCI 事务处理。

2. 地址翻译

- 将 32 位 PCI 地址翻译成 32 位本地地址

初级 ATU 在向内地址翻译时使用下述寄存器：

- 初级向内 ATU 基址寄存器

- 初级向内 ATU 界限寄存器

- 初级向内 ATU 翻译值寄存器



次级 ATU 在向内地址翻译时使用下述寄存器:

- 次级向内 ATU 基址寄存器
- 次级向内 ATU 界限寄存器
- 次级向内 ATU 翻译值寄存器

5 一般来说, 初级向内 ATU 地址就是初级 PCI 地址, 次级向内 ATU 地址就是次级 PCI 地址。如果某地址既能被 ATU 要求, 又能被桥要求, 则向内 ATU 的 PCI 接口享有优先权。

向内地址查验是通过 32 位 PCI 地址、基址寄存器和界限寄存器确定的, 查验算法是:

10 如果 (PCI 地址 & 界限寄存器 = 基址寄存器)
则向内 ATU 要求 PCI 地址

该算法将 32 位的输入 PCI 地址与关联的向内界限寄存器逐位进行逻辑“与”运算。如果运算结果与基址寄存器的值匹配, 查验结果就是该向内 PCI 地址在向内翻译窗口范围内, 因而由该 ATU 要求。

15 一旦事务处理被要求, 就必须将 IAQ 中的地址从 32 位 PCI 地址翻译成 32 位本地处理器地址。翻译的算法是:

本地地址 = (PCI 地址 & ~界限寄存器) | 值寄存器

20 该算法首先将 32 位的输入 PCI 地址与界限寄存器的“反”逐位进行逻辑“与”运算, 再将结果与值寄存器逐位进行逻辑“或”运算, 结果就是本地地址。这个翻译机制可用于除向内配置读命令和向内配置写命令之外的所有向内存储器读命令和向内存储器写命令。多个 PCI 地址对同一物理本地地址的地址混淆问题, 可将向内值寄存器编程在与所关联的界限寄存器值相匹配的界限上而加以避免, 但这只是通过应用程序设计而进行的。

向外地址翻译

25 地址翻译单元除了提供向内翻译机制, 还提供了将由本地处理器启动的周期翻译到 PCI 总线所需的硬件。这就是向外地址翻译。向外事务处理就是以某个 (初级或次级) PCI 总线作为目标的处理器读或处理器写。ATU 本地总线从接口将要求本地处理器总线周期并代表本地处理器完成 PCI 总线上的周期。初级和次级 ATU 支持两种不同的向外翻译方式:

- 30
- 地址翻译窗口
 - 直接寻址窗口

扩展 ROM 翻译单元



初级向内 ATU 支持一个用于放置扩展 ROM 的（由基址/界限寄存器定义的）地址区间。《PCI 本地总线规范》中对扩展 ROM 的格式和用法有详细说明。

在对关联设备进行初始化的通电序列期间，主处理器只执行一次来自扩展 ROM 的初始化码。该初始化码一经执行，就可弃之不用。

向内初级 ATU 支持一个按照向内翻译窗口的方式工作的向内扩展 ROM 窗口翻译。来自扩展 ROM 的读命令将被传递到本地总线和存储器控制器。其地址翻译算法与向内翻译的相同。支持 8 位和 32 位这两种不同宽度的扩展 ROM。要用软件在 ATUCR 中对扩展 ROM 宽度位进行编程，以反映扩展 ROM 的物理配置。该位确定了 ATU 访问扩展 ROM 的方式（见下文）。

向内 ATU 执行下述功能：

- 初级 ATU 查验对扩展 ROM 的一次“命中”（hit）。
- 初级 ATU 用寄存器 ERTVR 和 ERLR 来翻译（IAQ 中的）地址。
- 为适应 8 位设备，向内 ATU 在本地总线上要执行 4 次独立的读操作，以向初级 PCI 总线返回一个 32 位的值。每次读操作由一个以字节启动位 BE1:0# 用作字节地址的 8 位周期构成。每次读操作向初级 ATU 内部的地址拼装硬件（在 AD7:0 字节通路(byte lane)上）返回一个数据字节。

- P2P 存储器控制器对来自初级 ATU 的每一个本地总线请求作出响应，执行一次读操作。因为访问的是 8 位的设备，所以存储器控制器用一个 8 位周期执行一次 8 位读操作。存储器控制器（根据字节启动位）将数据返回给适当的字节通路。

- 在完成了四个周期时，ATU 内的拼装硬件将（来自延时读事务处理的）整个 32 位字返回给初级 PCI 总线。拼装硬件负责保证这些字节在正确的通路上。

如果初级 PCI 总线请求的小于 4 个字节，初级 ATU 就相应地调整进行字节读操作的次数。如果访问的地址与本地总线地址位置不对齐，就会导致读操作寻址到边界对齐的地址。

寄存器定义

每个 PCI 设备实现自己单独的配置地址空间和配置寄存器。《PCI 本地总线规范》（2.1 版）要求，配置空间应具有 256 个字节的长度，其中的前 64 个字节要具有预定的首部格式。



初级和次级 ATU 都是通过初级接口上的 0 类型配置命令而编程的。可以通过次级向内配置周期对次级 ATU 编程。ATU 和报文发送单元配置空间是 P2P 处理器多功能 PCI 设备的首要功能。

除了所要求的 64 字节首部格式外，ATU 和报文发送单元配置空间具有支持其单元功能的扩展寄存器空间。《PCI 本地总线规范》中对如何访问配置寄存器空间和编程配置寄存器空间有详细的说明。

配置空间由以预定格式安排的 8 位、16 位、24 位、32 位寄存器组成。表 6 是对 PCI 功能 1 配置地址空间内的所有寄存器的概括。

表中列出了每个寄存器的功能、访问类型（读/写、读/清零、只读）和复位缺省条件。表 6 中还为每个寄存器给出了 PCI 寄存器号。本文曾经指出，要对这些寄存器进行读或写，需要使用具有活跃的 IDSEL 或内存映射本地处理器访问的初级或次级总线上的 0 类型配置命令。



表 6. 地址翻译单元 PCI 配置寄存器总汇

寄存器名称	寄存器 位数	PCI 配置 周期 寄存器号	本地处理器 周期 地址位移
ATU 销售商(Vendor)标识符	16	0	00H
ATU 设备标识符	16	0	02H
初级 ATU 命令寄存器	16	1	04H
初级 ATU 状态寄存器	16	1	06H
ATU 修订标识符	8	2	08H
ATU 类别码	24	2	09H
ATU 高速缓存行(Cacheline)大小	8	3	0CH
ATU 等待定时器	8	3	0DH
ATU 首部类型	8	3	0EH
BIST 寄存器	8	3	0FH
初级向内 ATU 基址	32	4	10H
保留	32	5	14H
保留	32	6	18H
保留	32	7	1CH
保留	32	8	20H
保留	32	9	24H
保留	32	10	28H
保留	32	11	2CH
扩展 ROM 基址	32	12	30H
保留	32	13	34H
保留	32	14	38H
ATU 中断线 (line)	8	15	3CH
ATU 中断接插点 (pin)	8	15	3DH
ATU 最小许可(MinimumGrant)	8	15	3EH
ATU 最大等待(MaximumLatency)	8	15	3FH
初级向内 ATU 界限寄存器	32	16	40H
初级向内翻译值寄存器	32	17	44H



次级向内 ATU 基址寄存器	32	18	48H
次级向内 ATU 界限寄存器	32	19	4CH
次级向内 ATU 翻译值寄存器	32	20	50H
初级向外存储器窗口值寄存器	32	21	54H
保留	32	22	58H
初级向外 I/O 窗口寄存器	32	23	5CH
初级向外 DAC 窗口值寄存器	32	24	60H
初级向外高 64 位 DAC 寄存器	32	25	64H
次级向外存储器窗口值寄存器	32	26	68H
次级向外 I/O 窗口寄存器	32	27	6CH
保留	32	28	70H
扩展 ROM 界限寄存器	32	29	74H
扩展 ROM 翻译值寄存器	32	30	78H
保留	32	31	7CH
保留	32	32	80H
保留	32	33	84H
ATU 配置寄存器	32	34	88H
保留	32	35	8CH
初级 ATU 中断状态寄存器	32	36	90H
次级 ATU 中断状态寄存器	32	37	94H
初级 ATU 命令寄存器	32	38	98H
次级向外 DAC 窗口值寄存器	32	39	9CH
次级向外高 64 位 DAC 寄存器	32	40	A0H
初级向外配置周期地址寄存器	32	41	A4H
次级向外配置周期地址寄存器	32	42	A8H
初级向外配置周期数据寄存器	32	PCI 配置空间中不提供	ACH
次级向外配置周期数据寄存器	32	PCI 配置空间中不提供	B0H
保留	32	45	B4H
保留	32	46	B8H



保留	32		BCH
保留	32		C0H

由于 ATU 销售商(Vendor)标识符寄存器至 ATU 最大等待寄存器遵守《PCI 本地总线规范》中的定义, 因此本文不再赘述。

初级向内 ATU 界限寄存器—PIALR

5 初级向内地址翻译出现在从 PCI 总线(起源于初级 PCI 总线)到本地总线发生的数据传递上。地址翻译部件将 PCI 地址转换成本地处理器地址。所有数据传递被直接翻译, 这样, 启动数据传递的总线主就要将没有对齐的数据传递分解成多次数据传递。各字节启动位要对哪些数据路径有效作出规定。

10 初级向内翻译的基址在“初级向内 ATU 基址寄存器—PIABAR”中规定。确定块大小的要求时, 初级翻译界限寄存器提供初级基址寄存器的块大小要求。用于这些地址翻译的其它寄存器在上文的“向内地址翻译”中作了讨论。

15 本地处理器值寄存器中所含的编程值必须与基址寄存器中的编程值自然对齐。界限寄存器被用作屏蔽寄存器, 这使得本地处理器值寄存器中编程的低地址位无效。《PCI 本地总线规范》中有关于编程基址寄存器的其它资料。

表 7a. 初级向内 ATU 界限寄存器—PIALR

位	缺省值	读/写	说明
31:04	0000000H	只读	初级向内翻译界限—这是一个回读值, 它确定了初级 ATU 翻译单元所要求的内存块大小。
03:00	0000 ₂	只读	保留

初级向内 ATU 翻译值寄存器—PIATVR

20 初级向内 ATU 翻译值寄存器(PIATVR)中含有用于转换初级 PCI 总线地址的本地地址。作为初级向内 ATU 地址翻译的结果, 转换成的地址将被驱动到本地总线上。

表 7b. 初级向内 ATU 翻译值寄存器—PIATVR

位	缺省值	读/写	说明
31:02	0000.0800	只读	初级向内 ATU 翻译值—该寄存器中含有用



	H		于将初级 PCI 地址转换成本地地址的值。初级向内地址翻译值在本地总线上必须是字对齐的。其缺省地址允许翻译单元对内部的 P2P 内存映射寄存器进行访问。
01:00	00 ₂	只读	保留

次级向内 ATU 基址寄存器— SIABAR

次级向内 ATU 基址寄存器 (SIABAR) 定义次级向内翻译窗口起始处的内存地址块。向内 ATU 对总线请求进行解码, 并将其连同映射到本地存储器的翻译地址传递到本地总线。 SIABAR 定义基址并描述所需内存块的大小。对基址寄存器的影响是, 如果有值 FFFF.FFFFH 被写到 SIABAR, 则下一个对该寄存器的读操作返回的是初级向内 ATU 界限寄存器 (SIALR) 的数据, 而不是 SIABAR 的数据。

在基址寄存器中的编程值必须符合对 PCI 编程关于地址对齐的要求。《 PCI 本地总线规范》中有关于编程基址寄存器的其它资料。

表 7c. 次级向内 ATU 基址寄存器— SIBADR

位	缺省值	读/写	说明
31:04	XXXXXXXX H	读/写	次级翻译基址—这些寄存器位定义的是, 当被次级 PCI 总线寻址时, 次级翻译功能要对之作出回答的实际位置。缺省的块大小不确定。
03	1 ₂	只读	可预取指示器—该位指示, 存储器空间是可预取的。
02:01	00 ₂	只读	地址类型—这两位定义在何处定位内存块。基址必须位于地址空间中的前 4G 字节的任何位置 (即地址的低 32 位) 。
00	0 ₂	只读	存储器空间指示器—该位域指示的是, 寄存器内容描述了存储器或 I/O 空间的基址。 ATU 并不占用 I/O 空间, 因此该位必须为 0 。

次级向内 ATU 界限寄存器— SIALR

次级向内地址翻译出现在从次级 PCI 总线向本地总线发生数据传递时。地址翻译块将 PCI 地址翻译成本地处理器地址。所有数据传递被直接翻译, 这样, 启动数据传递的总线主就要将没有对齐的数据传递分解成多

次数据传递。各字节启动位规定哪些数据路径有效。

确定块大小的要求时，次级翻译界限寄存器提供次级基址寄存器的块大小要求。用于这些地址翻译的其它寄存器在上文的“向内地址翻译”中作了讨论。

- 5 本地处理器值寄存器中所含的编程值必须与基址寄存器中的编程值自然对齐。界限寄存器被用作屏蔽寄存器，这使得本地处理器值寄存器中编程的低地址位无效。

表 7d. 次级向内 ATU 界限寄存器— SIALR

位	缺省值	读/写方式	说明
31:04	0000000H	只读	次级向内 ATU 界限—这是一个回读值，它确定了次级 ATU 翻译单元所要求的内存块大小。
03:00	0000 ₂	只读	保留

次级向内 ATU 翻译值寄存器— SIATVR

- 10 次级向内 ATU 翻译值寄存器 (SIATVR) 中含有用于将次级 PCI 总线地址转换成本地地址的本地地址。作为次级向内 ATU 地址翻译的结果，转换成的地址将被驱动到本地总线上。

表 7e. 次级向内 ATU 翻译值寄存器— SIATVR

位	缺省值	读/写	说明
31:02	0000.0800 H	只读	次级向内 ATU 翻译值—该值用于将次级 PCI 地址转换成一个本地地址。次级向内地址翻译值在本地总线上必须是字对齐的。其缺省地址允许翻译单元访问内部的 P2P 寄存器。
01:00	00 ₂	只读	保留

初级向外存储器窗口值寄存器— POMWVR

- 15 初级向外存储器窗口值寄存器 (POMW0VR) 中含有用于为向外事务处理转换本地地址的初级 PCI 地址。该地址作为初级向外 ATU 地址翻译的结果，将被驱动到初级 PCI 总线上。

初级存储器窗口 0 在本地处理器地址 8000.000H ~ 807F.FFFFH，固定长度为 8M 字节。

表 7f. 初级向外存储器窗口值寄存器— POMW0VR

位	缺省值	读/写	说明
31:02	0000.0000H	读/写	初级向外存储器窗口值—该值用于将本地地址转换成 PCI 地址
01:00	00 ₂	只读	猝发 (burst) 次序—该位表示存储器猝发期间的地址顺序。所有目标都必须检查地址位 0、1 的状态以确定启动器意欲在猝发式数据传递期间使用的存储器寻址顺序。 (00 ₂) 猝发式数据传递期间使用线性的或顺序的寻址顺序。

初级向外 I/O 窗口值寄存器— POIOWVR

初级向外 I/O 窗口值寄存器 (POIOWVR) 中含有 ATU 将要把本地总线访问转换成的、初级 PCI I/O 读或写。

5 初级 I/O 窗口在本地处理器地址 8200.000H ~ 8200.FFFFH，固定长度为 64K 字节。

表 7g. 初级向外 I/O 窗口值寄存器— POIOWVR

位	缺省值	读/写	说明
31:02	0000.0000H	读/写	初级向外 I/O 窗口值—该值用于将本地地址转换成 PCI 地址
01:00	00 ₂	只读	猝发 (burst) 次序—该位表示存储器猝发期间的地址顺序。所有目标都必须检查地址位 0、1 的状态以确定启动器意欲在猝发式数据传递期间使用的存储器寻址顺序。 (00 ₂) 猝发式数据传递期间使用线性的或顺序的寻址顺序。

初级向外 DAC 窗口值寄存器— PODWVR

10 初级向外 DAC 窗口值寄存器 (PODWVR) 中含有用于转换本地地址的初级 PCI 的 DAC 地址。该地址作为初级向外 ATU 地址翻译的结果，将被驱动到初级 PCI 总线上。该寄存器与初级向外高 64 位 DAC 寄存器联合

使用。

初级 DAC 窗口在本地地址 8080.000H ~ 80FF.FFFFH，固定长度为 8M 字节。

表 7h. 初级向外 DAC 窗口值寄存器—PODWVR

位	缺省值	读/写	说明
31:02	00000000H	读/写	初级向外 DAC 窗口值—该值被初级 ATU 用于将本地地址转换成 PCI 地址
01:00	00 ₂	只读	猝发 (burst) 次序—该位表示存储器猝发期间的地址顺序。所有目标都必须检查地址位 0、1 的状态以确定启动器意欲在猝发式数据传递期间使用的存储器寻址顺序。 (00 ₂) 猝发式数据传递期间使用线性的或顺序的寻址顺序。

5 初级向外高 64 位 DAC 寄存器—POUDR

初级向外高 64 位 DAC 寄存器 (POUDR) 定义双地址周期期间所用地址的高 32 位。这使得初级向外 ATU 能够寻址到 64 位主机地址空间内的任何位置。

表 7i. 初级向外高 64 位 DAC 寄存器—POUDR

位	缺省值	读/写	说明
31:00	0000.0000H	读/写	这些寄存器位定义双地址周期 (DAC) 期间所驱动的地址的高 32 位。

10 次级向外存储器窗口值寄存器—SOMWVR

次级向外存储器窗口值寄存器 (SOMWVR) 中含有用于为向外事务处理转换本地地址的次级 PCI 地址。该地址作为次级向外 ATU 地址翻译的结果，将被驱动到次级 PCI 总线上。

次级存储器窗口来自本地地址 8100.000H ~ 817F.FFFFH，固定长度为 8M 字节。

表 7j. 次级向外存储器窗口值寄存器—SOMWVR

位	缺省值	读/写	说明
31:02	0000.0000H	读/写	次级向外存储器窗口值—该值用于将本



			地地址转换成 PCI 地址
01:00	00 ₂	只读	猝发 (burst) 次序—该位表示存储器猝发期间的地址顺序。所有目标都必须检查地址位 0、1 的状态以确定启动器意欲在猝发式数据传递期间使用的存储器寻址顺序。 (00 ₂) 猝发式数据传递期间使用线性的或顺序的寻址顺序。

次级向外 I/O 窗口值寄存器—SOIOWVR

次级向外 I/O 窗口值寄存器 (SOIOWVR) 中含有用于转换本地地址的次级 PCI I/O 地址。该地址作为次级向外 ATU 地址翻译的结果, 将被驱动到次级 PCI 总线上。

- 5 如果 ATUCR 中的次级 PCI 引导模式位被置位, 则该寄存器被用于翻译访问区间在 FE00.0000H ~ FFFF.FFFFH 范围的本地地址。如果该位被清零, 则该寄存器被用于翻译访问次级 I/O 窗口中 FE00.0000H ~ FFFF.FFFFH 范围的本地地址。

10 次级 I/O 窗口在本地地址 8201.0000H ~ 8201.FFFFH, 固定长度为 64K 字节。

表 7k. 初级向外 I/O 窗口值寄存器—POIOWVR

位	缺省值	读/写	说明
31:02	00000000H	读/写	次级向外 I/O 窗口值—该值用于将本地地址转换成 PCI 地址
01:00	00 ₂	只读	猝发 (burst) 次序—该位表示存储器猝发期间的地址顺序。所有目标都必须检查地址位 0、1 的状态以确定启动器意欲在猝发式数据传递期间使用的 I/O 寻址顺序。 (00 ₂) 猝发式数据传递期间使用线性的或顺序的寻址顺序。

扩展 ROM 界限寄存器—ERLR

扩展 ROM 界限寄存器 (ERLR) 定义初级 ATU 将要定义为扩展 ROM 地址空间的地址的块大小。块大小是通过从本地处理器将一个值写入



ERLR 而编程的。可能的编程值的范围是从 2K 字节 (FFFF.F800H) 到 16M 字节 (FF00.0000H)。

表 7l. 扩展 ROM 界限寄存器— ERLR

位	缺省值	读/写	说明
31:11	000000H	只读	扩展 ROM 界限—扩展 ROM 翻译单元要求的内存块大小。缺省值为 0，表示无扩展 ROM 地址空间。
10:00	000H	只读	保留

扩展 ROM 翻译值寄存器— ERTVR

- 5 扩展 ROM 翻译值寄存器 (ERTVR) 中含有初级 ATU 要转换的、初级 PCI 总线访问的本地地址。该地址作为初级扩展 ROM 地址翻译的结果，将被驱动到本地总线上。

表 7m. 扩展 ROM 翻译值寄存器— ERTVR

位	缺省值	读/写	说明
31:02	00000000H	只读	扩展 ROM 本地处理器翻译值—该值用于将 PCI 地址转换成用于扩展 ROM 访问的本地地址。扩展 ROM 地址翻译值必须是本地总线上字对齐的。
01:00	00 ₂	只读	保留

ATU 配置寄存器— ATUCR

- 10 ATU 配置寄存器 (ATUCR) 中含有的控制位，允许或禁止由门铃寄存器生成的中断。 ATU 配置寄存器还控制初级和次级向外翻译单元的向外地址翻译，它含有一个定义扩展 ROM 宽度的位。

表 7n. ATU 配置寄存器 - ATUCR

位	缺省值	读/写	说明
31:12	00000H	只读	保留
11	0 ₂	读/写	次级 PCI 引导模式 - 该位如果置位, 次级 ATU 将要求地址范围在 FE000000H ~ FFFFFFFFH 的本地总线访问。这就使本地处理器能够从次级 PCI 总线进行引导。在这种模式下, 翻译算法要用到次级向外 I/O 窗口值寄存器。
10	0 ₂	只读	保留
09	0 ₂	只读	保留
08	0 ₂	读/写	直接寻址启动 - 该位如果置位, 就允许进行通过地址翻译单元的直接寻址。地址范围在 0000.1000H ~ 07FF.FFFFH 的本地总线周期不作地址翻译就被自动地传递到 PCI 总线。要求直接寻址事务处理的 ATU 取决于次级直接寻址选择位的状态。
07	0 ₂	读/写	次级直接寻址选择 - 对该位置位的结果是, 直接寻址事务处理通过次级 ATU 被传递到次级 PCI 总线。该位如果清零, 直接寻址要使用初级 ATU 和初级 PCI 总线。要允许直接寻址, 必须将直接寻址启动位置位。
06	0 ₂	读/写	扩展 ROM 宽度 - 该位如果清零, 表示使用的是 8 位的扩展 ROM; 如果置位, 表示使用的是 32 位的扩展 ROM。该位与 ERBAR 的地址解码启动位 (位 0) 联合使用。
05	0 ₂	读/写	次级 ATU PCI 错误中断启动 - 该位用作次级 ATU 中断状态寄存器的 4:0 位的屏蔽位。如果置位, SATUISR 中的这些位中任何位置位时允许对本地处理器的中断; 如果

			清零，则不允许中断。
04	0 ₂	读/写	初级 PCI 错误中断启动 - 该位用作初级 ATU 中断状态寄存器的 4:0 位的屏蔽位。如果置位，SATUISR 中的这些位中任何位置位时允许对本地处理器的中断；如果清零，则不允许中断。
03	0 ₂	读/写	ATU BIST 中断启动 - 如果置位，则允许在 ATUBISTR 寄存器的启动 BIST 位置位时对本地处理器的中断；该位内容也反映为 ATUBISTR 寄存器的寄存器位 BIST 能位 (Capable bit)7 中。
02	0 ₂	读/写	次级向外 ATU 启动 - 该位如果置位，使能次级向外地址翻译单元；如果清零，则禁止次级向外地址翻译单元。
01	0 ₂	读/写	初级向外 ATU 启动 - 该位如果置位，使能初级向外地址翻译单元；如果清零，则禁止初级向外地址翻译单元。
00	0 ₂	读/写	门铃中断启动 - 该位若置位便使能报文传递单元为每个向内门铃写操作生成一个本地处理器中断。如果清零，则不生成中断。

初级 ATU 中断状态寄存器 - PATUISR

初级 ATU 中断状态寄存器 (PATUISR) 用于向本地处理器告知初级 ATU 或门铃中断的中断源。此外，可以对该寄存器进行写操作以清除对 P2P 处理器的中断单元的中断的中断源。该寄存器的所有位对 PCI 是只读的，对本地总线是读/清零的。

寄存器位 4:0 的内容是初级 ATU 状态寄存器的位 8 和位 14:11 的直接反映 (这些位的设置是由硬件同时作出的，但清除时则需分别进行)。寄存器位 6:5 由出现与存储器控制器关联的错误时置位。位 8 用于软件 BIST，位 10:9 用于报文传递单元。通过在该寄存器中的适当的位写入 1 可以清除引起初级 ATU 中断或门铃中断的中断条件。

表 70. 初级 ATU 中断状态寄存器 - PATUISR

位	缺省值	读/写	说明
31:11	000000H	只读	保留
10	0 ₂	只读	保留
09	0 ₂	只读	保留
08	0 ₂	读/清零	ATU BIST 中断 - 该位如果置位, 表明主处理器已经将启动 BIST 位 (即 ATUBISTR 寄存器的位 6) 置位, 并且 ATU BIST 中断启动位 (即 ATUCR 寄存器的位 12) 被启动。本地处理器可以启动软件 BIST 并将结果存储到 ATUBISTR 寄存器的位[3:0]。
07	0 ₂	只读	保留
06	0 ₂	读/清零	本地处理器存储器故障 - 当存储器控制器检测到一个存储器故障而初级 ATU 又是事务处理的主时, 该位被置位。
05	0 ₂	读/清零	本地处理器总线故障 - 当存储器控制器检测到一个总线故障而初级 ATU 又是事务处理的主时, 该位被置位。
04	0 ₂	读/清零	P_SERR#确立 - 如果 P_SERR#在 PCI 总线上被确立, 则该位被置位。
03	0 ₂	读/清零	PCI 主控流产 - 每当由 ATU 主接口启动的事务处理以主-流产方式结束时, 该位被置位。
02	0 ₂	读/清零	PCI 目标流产 (主) - 每当由 ATU 主接口启动的事务处理以主-流产方式结束时, 该位被置位。
01	0 ₂	读/清零	PCI 目标流产 (目标) - 每当作为目标的 ATU 接口以目标流产方式停止事务处理时, 该位被置位。
00	0 ₂	读/清零	PCI 主奇偶校验出错 - ATU 接口在符合以下三个条件时将该位置位:

			<p>1)总线代理自己确立 P_SERR#或检测到 P_SERR#已被确立</p> <p>2)为该位置位的代理在出错的操作中担当了总线主的角色</p> <p>3)(命令寄存器中的)奇偶校验错误响应位被置位</p> <p>该位可被主处理器对 PATUSR 的位 8 进行读/清零操作而清零。</p>
--	--	--	--

次级 ATU 中断状态寄存器— SATUISR

次级 ATU 中断状态寄存器 (SATUISR) 用于向本地处理器告知次级 ATU 中断的中断源。此外, 可以对该寄存器进行写操作以清除对 P2P 处理器的中断单元的中断的中断源。该的所有寄存器位对 PCI 是只读的, 对本地总线是读/清零的。

通过在该寄存器中的适当的位写入 1 可以清除引起次级 ATU 中断的中断条件。

表 7p. 次级 ATU 中断状态寄存器— SATUISR

位	缺省值	读/写	说明
31:07	0000000H	只读	保留
06	0 ₂	读/清零	本地处理器存储器故障—当存储器控制器检测到一个存储器故障而次级 ATU 作为事务处理的主时, 该位被置位。
05	0 ₂	读/清零	本地处理器总线故障—当存储器控制器检测到一个总线故障而次级 ATU 作为事务处理的主时, 该位被置位。
04	0 ₂	读/清零	P_SERR#确立—如果 P_SERR#在 PCI 总线上被确立, 则该位被置位。
03	0 ₂	读/清零	PCI 主流产—每当由 ATU 主接口启动的事务处理以主-流产方式结束时, 该位被置位。
02	0 ₂	读/清零	PCI 目标流产(主)—每当由 ATU 主接口启动的事务处理以主-流产方式结束时, 该



			位被置位。
01	0 ₂	读/清零	PCI 目标流产 (目标) — 每当作为目标的 ATU 主接口, 以目标流产方式停止事务处理时, 该位被置位。
00	0 ₂	读/清零	<p>PCI 主奇偶校验出错—次级 ATU 接口在符合以下三个条件时将该位置位:</p> <ol style="list-style-type: none"> 1) 总线代理自己确立 P_SERR# 或检测到 P_SERR# 已被确立 2) 为该位置位的代理在出错的操作中担当了总线主角色 3) (命令寄存器中的) 奇偶校验错误响应位被置位 <p>该位可被主处理器对 PATUSR 的位 8 进行读/清零操作而清零。</p>

次级 ATU 命令寄存器—SATUCMD

次级 ATU 命令寄存器 (SATUCMD) 中的位遵守《PCI 本地总线规范》的定义, 大多数情况下, 它们影响次级 PCI 总线上设备的状态

表 7q. 次级 ATU 命令寄存器—SATUCMD

位	缺省值	读/写	说明
31:10	000000H	只读	保留
09	0 ₂	读/写	快速背对背启动—如果该位被清零, 就不允许次级 ATU 接口在其总线上生成快速背对背周期。
08	0 ₂	读/写	S_SERR# 确立—如果该位被清零, 就不允许次级 ATU 接口在 PCI 接口上确立 S_SERR。
07	0 ₂	只读	等待周期控制—控制地址/数据的步进。未实现, 是个保留位域。
06	0 ₂	读/写	奇偶校验检测启动—如果位被置位, 在有奇偶校验错误被检测出时, 次级 ATU 接口必须采取正常行动。如果该位被清零, 则

			禁止奇偶校验的检测。
05	0 ₂	只读	VGA 调色板探测启动—次级 ATU 接口不支持 I/O 写操作，因此不进行 VGA 调色板探测。
04	0 ₂	只读	存储器写和无效启动—不适用。未实现，是个保留位。
03	0 ₂	只读	特殊周期启动— ATU 接口根本不响应特殊周期命令。不适用。未实现，是个保留位。
02	0 ₂	读/写	总线主启动—次级 ATU 接口能够作 PCI 总线上的主。该位的值为 0 时，禁止设备生成 PCI 访问；值为 1 时，允许设备担当主。该启动位也控制 DMA 通道 0 和 1 的主接口。在 PCI 总线上启动 DMA 传输之前，该位必须被置位。
01	0 ₂	读/写	存储器启动—控制次级 ATU 接口对 PCI 存储器地址的响应。如果该位清零，ATU 接口对 PCI 总线上的 PCI 存储器访问就不予响应。
00	0 ₂	读/写	I/O 空间启动—控制次级 ATU 接口对初级端的 I/O 事务处理的响应。未实现，是个保留位。

次级向外 DAC 窗口值寄存器— SODWVR

次级向外 DAC 窗口值寄存器 (SODWVR) 中含有用于转换本地地址的次级 PCI DAC 地址。该地址作为次级向外 ATU 地址翻译的结果，将被驱动到次级 PCI 总线上。该寄存器与次级向外高 64 位 DAC 寄存器联合使用。

次级 DAC 窗口在本地处理器地址 8180.0000H ~ 81FF.FFFFH，固定长度为 8M 字节。

表 7r. 初级向外 DAC 窗口值寄存器— PODWVR

位	缺省值	读/写	说明
31:02	00000000H	读/写	次级向外 DAC 窗口值—次级 ATU 用该值将本地地址转换成 PCI 地址
01:00	00 ₂	只读	猝发 (burst) 次序—该位表示存储器猝发期间的地址顺序。所有目标都必须检查地址位 0、1 的状态以确定启动器意欲在猝发式数据传递期间使用的存储器寻址顺序。 (00 ₂) 猝发式数据传递期间使用线性的或顺序的寻址顺序。

次级向外高 64 位 DAC 寄存器— SOUDR

次级向外高 64 位 DAC 寄存器 (SOUDR) 定义双地址周期期间所用地址的高 32 位。这使得次级向外 ATU 能够寻址到 64 位主机地址空间内的任何位置。

表 7s. 次级向外高 64 位 DAC 寄存器— SOUDR

位	缺省值	读/写	说明
31:00	0000.0000 H	读/写	次级向外高 64 位 DAC 地址—这些寄存器位定义双地址周期 (DAC) 期间所用地址的高 32 位。

初级向外配置周期地址寄存器— POCCAR

初级向外配置周期地址寄存器 (POCCAR) 用于存放 32 位 PCI 配置周期地址。本地处理器写入允许进行初级向外配置读或写操作的 PCI 配置周期地址。然后, 本地处理器对初级向外配置周期数据寄存器执行读或写操作, 在初级 PCI 总线上启动配置周期。

表 7t. 初级向外配置周期地址寄存器— POCCAR

位	缺省值	读/写	说明
31:00	0000.0000	只读	初级向外配置周期地址—这些



	H		位定义初级向外配置读或写周期期间所用的 32 位 PCI 地址。
--	---	--	----------------------------------

次级向外配置周期地址寄存器— SOCCAR

次级向外配置周期地址寄存器（SOCCAR）用于存放 32 位 PCI 配置周期地址。本地处理器写入允许进行次级向外配置读或写操作的 PCI 配置周期地址。然后，本地处理器对次级向外配置周期数据寄存器执行读或写操作，在次级 PCI 总线上启动配置周期。

表 7u. 次级向外配置周期地址寄存器— SOCCAR

位	缺省值	读/写	说明
31:00	0000.0000 H	只读	次级向外配置周期地址—这些位定义次级向外配置读或写周期期间所用的 32 位 PCI 地址。

初级向外配置周期数据寄存器— POCCDR

初级向外配置周期数据寄存器（POCCDR）用于启动初级 PCI 总线上的配置读或写操作。该寄存器只有逻辑意义而没有物理意义，就是说，它是一个地址而不是寄存器。本地处理器用 POCCAR 中找到的地址读出或写入该数据寄存器的内存映射地址，启动初级 PCI 总线上的配置周期。对于配置写操作，从本地总线上锁存该数据，而被直接传递到 ATU 的 ODQ。对于读操作，该数据被直接从 ATU 的 IDQ 返回给本地处理器，根本不存入数据寄存器（该寄存器物理上不存在）。

POCCDR 只对本地处理器地址空间有用并表现为 ATU 配置空间内的一个保留值。

次级向外配置周期数据寄存器— SOCCDR

次级向外配置周期数据寄存器（SOCCDR）用于启动次级 PCI 总线上的配置读或写操作。该寄存器只有逻辑意义而没有物理意义，就是说，它是一个地址而不是寄存器。本地处理器用 SOCCAR 中找到的地址读出或写入数据寄存器的内存映射地址，启动次级 PCI 总线上的配置循环。对于配置写操作，从本地总线锁存该数据，并被直接传递到 ATU 的 ODQ。对于读操作，该数据被直接从 ATU 的 IDQ 返回给本地处理器，根本不存入数据寄存器（该寄存器物理上不存在）。

SOCCDR 只对本地处理器地址空间有用并表现为 ATU 配置空间内的一个保留值。

报文传递单元

以下描述 P2P 处理器的报文传递单元，报文传递单元与上述的初级地址翻译单元（PATU）有密切关系。

概述

5 报文传递单元提供了一种在 PCI 系统与本地处理器之间传递数据并通过中断向各系统通报新数据到达的机制。报文传递单元可用于发送和接收报文。

10 报文传递单元有 5 个不同的报文传递机构。各机构都允许主处理器或外部 PCI 代理与 P2P 处理器通过报文传送和中断生成进行通信。这 5 个机构是：

- 报文寄存器
- 门铃寄存器
- 环形队列
- 索引寄存器
- 15 · APIC 寄存器

报文寄存器允许 P2P 处理器与外部 PCI 代理通过用 4 个 32 位报文寄存器的其中一个传送报文而进行通信。本文中所说的报文，是指 32 位长度的任意数据值。报文寄存器综合了邮箱寄存器和门铃寄存器的特点。向报文寄存器写数据，可以有选择地引起中断。

20 门铃寄存器允许 P2P 处理器确定 PCI 中断信号，允许外部 PCI 代理向本地处理器生成一个中断。

环型队列支持使用 4 个环形队列的报文传送方案。

索引寄存器支持用 P2P 处理器的部分本地存储器来实现大的寄存器组的报文传送方案。

25 APIC 寄存器用提供用于访问 APIC 寄存器的外部 PCI 接口的方法支持 APIC 总线接口单元。

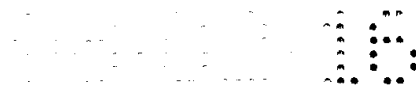
操作原理

报文传递单元有 5 个独特的报文传递机构。

报文寄存器类似于是邮箱寄存器与门铃寄存器的组合。

30 门铃寄存器支持硬件中断，也支持软件中断。门铃寄存器有两个用途：

- 被写时产生中断。
- 保留其它报文传递机构生成的中断的状态。



报文传递单元使用初级地址翻译单元 (PATU) 中初级向内翻译窗口的前 4K 字节。初级向内翻译窗口的地址存于初级向内 ATU 基址寄存器中。

表 8 概括报文传递单元中所用的 5 个报文传递机构。

5

表 8. 报文传递单元概述

机构	数量	是否确立 PCI 中断信 号	是否生成本 地处理器中 断
报文寄存器	2 个向内的 2 个向外的	是	可任选
命令寄存器	1 个向内的 1 个向外的	是	可任选
环形队列	4 个环形队 列	根据一定条 件	根据一定条 件
索引寄存器	1004 个 32 位存储器单 元	否	可任选
APIC 寄存器	1 个寄存器 选择 1 个窗口	否	是

报文寄存器

P2P 处理器可以通过报文寄存器传送和接收报文。向报文寄存器写入数据时，会导致向本地处理器或 PCI 中断信号生成中断。向内的报文由主处理器发送，P2P 处理器接收。向外的报文由 P2P 处理器发送，主处理器接收。

10

向外报文中断状态被记录在向外门铃寄存器中，向内报文中断状态被记录在向内门铃寄存器中。

向外报文

向外报文寄存器被本地处理器写入数据时，中断可能生成在中断线 P_INTA#、P_INTB#、P_INTC#、P_INTD# 上。究竟用的是哪个中断线，由 ATU 中断接插线寄存器的值决定。

15

PCI 中断被记录在向外门铃寄存器中。中断使向外门铃寄存器的向外



报文位置位。该位是读/清零的，由报文传递单元硬件置位，软件清零。

当外部 PCI 代理将数值 1 写到向外门铃寄存器的向外报文位来清除该位时，中断被清除。

向内报文

5 向内报文寄存器被外部 PCI 代理写入数据时，就可能向本地处理器生成一个中断。该中断可以被向内门铃屏蔽寄存器中的屏蔽位屏蔽。

本地处理器中断记录向内门铃寄存器中。中断使向内门铃寄存器的向内报文位置位。该位是读/清零的，由报文传递单元硬件置位，软件清零。

10 当外部本地处理器将数值 1 写到向内门铃寄存器的向内报文中断位时，中断被清除。

门铃寄存器

门铃寄存器有两个：向内门铃寄存器和向外门铃寄存器。向内门铃寄存器允许外部 PCI 代理向本地处理器生成中断。向外门铃寄存器允许本地处理器生成 PCI 中断。两个寄存器都保存硬件生成的中断与软件生成的中断的组合。它们包含来自其它报文传递单元机构的中断状态，也允许软件对寄存器位进行个别设置来产生中断。

向外门铃

向外门铃寄存器被本地处理器写入数据时，中断可能生成在中断插接线 P_INTA#、P_INTB#、P_INTC#、P_INTD#上。如果门铃寄存器中有寄存器位被写上数值 1，就有中断生成。向寄存器的任何一位写入数值 0，不会改变该位的值，也不会导致中断的生成。门铃寄存器中的寄存器位一旦被置位，就不能被本地处理器清零。

究竟用的是哪个中断插接线，由 ATU 中断接插线寄存器决定。

25 中断可以被向外门铃屏蔽寄存器中的屏蔽位屏蔽。如果某特定位对应的屏蔽位被置位，则该位就不会生成中断。向外门铃屏蔽寄存器只影响中断的生成，并不改变已经写到向外门铃寄存器中的值。

当外部 PCI 代理将数值 1 写到向外门铃寄存器中已经置位的寄存器位时，中断被清除。向寄存器位写 0 不会改变该位的值，也不清除中断。

30 总之，本地处理器通过对向外门铃寄存器中寄存器位的设置来生成中断，外部 PCI 代理也是通过对相同的寄存器中寄存器位的设置来清除中断。

向内门铃



向内门铃寄存器被外部 PCI 代理写入数据时，会对本地处理器生成中断。如果门铃寄存器中有寄存器位被写上数值 1，就有中断生成。向寄存器的任何一位写入数值 0，不会改变该位的值，也不会导致中断的生成。向内门铃寄存器中的寄存器位一旦被置位，就不能被任何外部 PCI 代理清

5

零。
中断可以被向内门铃屏蔽寄存器中的屏蔽位屏蔽。如果某特定位置位的屏蔽位被置位，则该位就不会生成中断。门铃屏蔽寄存器只影响中断的生成，并不改变已经写到向内门铃寄存器中的值。

向内门铃寄存器中为 NMI（非屏蔽中断）中断保留了一个寄存器位。该中断不能被向内门铃屏蔽寄存器屏蔽。

10

当本地处理器将数值 1 写到向内门铃寄存器中已经置位的寄存器位时，中断被清除。向寄存器位写 0 不会改变该位的值，也不清除中断。

环形队列

报文传递单元中实现 4 个环形队列：2 个向外环形队列，2 个向内环形队列。在此，向外或向内表示报文的流动方向。向内报文或者是其它处理器投递的、供本地处理器处理的新报文，或者是可被其它处理器再使用的空的或空闲报文。向外报文或者是本地处理器投递的、供其它处理器处理的报文，或者是可被本地处理器再使用的空闲报文。

15

这 4 个环形队列被用于按下述方式传送报文。2 个向内队列用于处理向内报文，向外队列则用于处理向外报文。向内队列中的一个被指定为空闲队列，它包含向内空闲报文；另一个向内队列被指定为投递队列，它包含向内投递报文。与此类似，向外队列中的有一个被指定为空闲队列，另一个向外队列被指定为投递队列。

20

次级 PCI 总线仲裁单元

下文描述次级 PCI 总线仲裁单元 53，内容是仲裁的操作模式、建立和实现。

25

概述

PCI 本地总线需要有一个为系统环境内每个 PCI 总线服务的中央仲裁资源。PCI 采用基于存取的仲裁概念而不是传统的时隙法。PCI 总线上的每个设备每当作为总线主要求总线进行存取时，都要为总线作出仲裁。

30

PCI 仲裁利用了一种简单的 REQ#和 GNT#握手协议。当设备要求总线时，它就确立它的 REQ#输出。仲裁单元 53 通过确立该代理的 GNT#输



入，允许发出请求的该代理访问总线。PCI仲裁是一种“隐藏式”的仲裁方案，其中的仲裁序列发生在后台，此时其它某个总线主正控制着总线。其优点是，总线仲裁开销不占用PCI的带宽。对仲裁器的唯一要求是必须实现一种公平的仲裁算法。所选择的仲裁算法必须保证，任何时刻单个PCI总线上不会有一个以上的GNT#活跃。

操作原理

次级总线仲裁器53支持多达6个次级总线主，加上次级总线接口自身。各个请求可以编程到三个优先级之一，也可以禁止。用应用软件编程的内存映射控制寄存器，确定每个总线主的优先级。对各优先级的处理是按轮式调度法进行的。轮式调度法是这样一种机制，即其中的每个设备轮流充当总线主，轮流顺序按环形排列。下一个可能的总线主直接排在当前总线主的前面，上一个总线主直接排在当前总线主的后面。

轮式仲裁方案支持3个级别的轮式仲裁。这三个级别是低、中、高优先级。用轮式机制保证了各优先级都有一个排列在前的设备。为了实现公平仲裁的思想，在次高级优先权中为各优先级（不含最高优先级）的在前设备保留一个位置。如果在某个仲裁序列中，某优先级的在前设备未获得总线，就将其提升到次高级优先权队列中。一旦该设备获得了总线，就恢复其编程的优先级。这种保留位置的方法保证了算法的公平性，原因在于它允许较低优先级的请求通过优先级机制被提升到最高优先级设备的位置上并在下一开启中被授予总线。

仲裁器通过REQ#-GNT#协议与总线上所有发出请求的代理联络。总线主确立REQ#输出，等待GNT#输入的确立。代理可以在前一个总线控制者仍然控制着总线的状态下就获得总线。仲裁器只负责确定下一次将总线分配给哪个PCI设备。各单个PCI设备则负责确定何时总线被释放，自己能开始访问总线。

通过编程接口可以将次级总线仲裁器53关闭，以实行例行仲裁算法。当次级仲裁器53被关闭时，一组REQ#-GNT#信号充当P2P的次级PCI接口的仲裁信号。

优先级机制

优先级机制可以用BIOS码或应用软件进行编程。单个总线主的优先级将决定该设备在轮式调度中被置于的位置的级别。该优先级确定的是设备的起始优先级或最低优先级，如果应用软件为设备编程了低优先级，该



设备可以被提升到中优先级，再被提升到高优先级，直到获得本地总线。一旦获得了总线，其优先级就被重新设置为编程的优先级，于是又可以再次重新启动仲裁。本发明领域的熟练人员应完全能够掌握实现适当的次级 PCI 总线仲裁单元所需的进一步细节。

5 DMA 控制器

以下描述本发明使用的集成的直接存储器存取 (DMA) 控制器，内容包含 DMA 控制器的操作模式、建立、外部接口和实现。

概述

DMA 控制器提供低等待、高通量的数据传输能力。DMA 控制器优化了 PCI 总线与本地处理器存储器之间数据的块传输。作为 PCI 总线上的启动器的 DMA 具有提供 33MHz 上最大为 132M 字节/秒通量的 PCI 猝发能力。

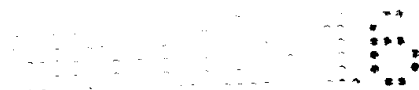
DMA 控制器的硬件负责执行数据传输及提供编程接口，其特点是：

- 3 个独立通道
- 使用 P2P 存储器控制接口
- 本地处理器接口上的寻址范围是 2^{32}
- 使用 PCI 双地址周期 (DAC)，在初级和次级 PCI 接口上的寻址范围是 2^{64}
- 对初级和次级 PCI 总线的独立 PCI 接口
- PCI 总线和本地处理器本地总线的不对齐数据传输的硬件支持
- 对 PCI 总线和 P2P 本地总线的完整的 132M 字节/秒猝发支持
- 对 PCI 总线的双向直接寻址
- 由本地处理器完全可编程
- 支持数据块集结和扩散的自动数据链接
- 需求模式支持 DMA 通道 0 上的外部设备

附图 7 表示了 DMA 通道到 PCI 总线的连接线路。

操作原理

DMA 控制器 51a 和 51b 提供 3 个高通量的 PCI - 存储器传输通道。通道 0 与通道 1 传输初级 PCI 总线和本地处理器本地存储器之间的数据块。通道 2 传输次级 PCI 总线和本地处理器本地存储器之间的数据块。除通道 0 外的其它通道完全相同。通道 0 多了支持需求模式传输。各个通道都有一个 PCI 总线接口和一个本地处理器本地总线接口。



各个 DMA 通道对 PCI 总线和本地处理器本地总线实行直接寻址。它们双向支持 PCI 总线的全部 64 位地址区间的数据传输, 这包括用 PCI DAC 命令的 64 位寻址。通道提供一个专用寄存器存放 64 位地址的高 32 位地址。DMA 通道不支持跨越 32 位地址边界的数据传输。

5 本发明领域的熟练人员应完全能够掌握实现供本发明中使用的适当的 DMA 机制所需的进一步细节。

存储器控制器

以下描述本发明使用的集成的存储器控制器 47, 内容有控制器的操作模式、建立、外部接口和实现。

10 概述

P2P 处理器中集成一个主存储器控制器 47, 为 P2P 处理器和存储器系统 33 之间提供一个直接接口。存储器控制器支持:

· 容量达 256M 字节的 32 位或 36 位 (32 位内存数据加 4 个奇偶校验位) DRAM

- 15 · 交错的或非交错的 DRAM
- 快速页 - 模式 (FPM) 的 DRAM
- 扩展数据外出 (EDO) 的 DRAM
- 猝发扩展数据外出 (BEDO) 的 DRAM
- 两个独立的 SRAM/DRAM 存储器体

20 · 容量达 16M 字节 (每存储器体) 的 8 位或 32 位 SRAM/ROM

存储器控制器为 DRAM 阵列生成行地址选通 (RAS#)、列地址选通 (CAS#)、写启动 (WE#) 和 12 位多路复用地址 (MA[11:0])。DRAM 地址锁存启动 (DALE#) 和 LEAF# 信号用于交错的 DRAM 中的地址和数据锁存。

25 存储器控制器支持两个 DRAM、ROM 体或快速存储器, 每一个体支持 64K 字节至 16M 字节的存储容量。每一个存储器体能被独立地配置成宽度 8 位或 32 位的存储器。存储器控制器提供芯片启动 (CE#) 信号、存储器写启动 (MWE#) 信号和递增猝发地址。

附图 8 是 P2P 处理器中集成的存储器控制器的概略图。

30 操作原理

存储器控制器 47 以最佳方式将本地总线主的猝发存取协议翻译成被寻址存储器所支持的存取协议。地址解码单元 101 对内部地址/数据总线上的



本地总线地址进行解码，生成适当的地址和控制信号到与存储器控制器相连的存储器阵列 33。本地总线主生成的猝发存取提供第一个地址。存储器控制器提供递增地址，送到 MA[11:0]插线上的存储器阵列。地址的递增要进行到本地总线主结束周期（表现为 BLAST#信号的确立），或者到
5 （ DRAM 读周期）发生一个本地总线奇偶校验错误。一个单一数据传输周期的最大猝发量是 2K 字节。本地总线主负责跟踪递增猝发的计数并在达到 2K 字节地址边界时结束数据传输。

在 MA[11:0]总线 103 上出现的地址取决于被寻址的存储器体的类型。如果是 DRAM， MA[11:0]就提供多路复用的行地址和列地址。列地址递
10 增到最接近的 2K 字节地址边界。如果是 SRAM 与 FLASH/ROM 存储器体， MA[11:0]总线上的地址就取决于地址相位期间出现在 AD[13:2]信号上的地址。对于猝发数据，猝发计数器将地址递增到最接近的 2K 字节地址边界。

存储器控制器生成编程到存储器控制器的寄存器 107 中的，用于控制
15 连接到存储器阵列的信号等待状态的数目。此外，在存储器存取期间，要求为（本地处理器之外的）本地总线主提供 WAIT#信号，以指示等待状态生成器 109 生成的其它等待状态。

可以为 DRAM 阵列启动字节宽数据奇偶校验生成和校验单元 111。奇
偶校验检测到奇偶校验错误时发出存储器故障错误。发生错误的字地址被
20 俘获入寄存器中。

存储器控制器提供总线监控器 113 来检测不返回外部 RDYRCV#信号的地址区间，其目的是对向未定义地址区间的访问进行检测。当检测发现
错误时，等待状态生成器就生成一个内部 LRDYRCV#去完成总线访问，并可选择地生成一个总线故障信号。

25 本发明领域的熟练人员应完全能够掌握实现供本发明使用的适当的存储器控制器所需的进一步细节。

PCI 和外围设备中断控制器

以下描述 P2P 处理器中断控制器支持，内容有中断的操作模式、建立、
外部存储器接口和实现。

30 概述

PCI 和外围设备中断控制器（ PPIC ） 67 具有向本地处理器和 PCI
总线两者生成中断的能力。 P2P 处理器中有许多可能向本地处理器产生中



断的外围设备，它们有：

- DMA 通道 0
- 初级 ATU
- DMA 通道 1
- 5 · 次级 ATU
- DMA 通道 2
- I²C 总线接口单元
- 桥初级接口
- APIC 总线接口单元
- 10 · 桥次级接口
- 报文传递单元

除内部设备之外，外部设备也会向本地处理器生成中断。外部设备可通过插接线 XINT:7:0#和 NMI#插接线生成中断。

15 PCI 和外围设备中断控制器提供直接 PCI 中断的能力。路由选择逻辑在软件控制下，启动截获外部次级 PCI 中断并把它们传递到初级 PCI 中断线的能力。

《I960 Jx 微处理器用户手册》进一步描述了本地处理器中断和中断优先权机制。该用户手册还彻底说明了本地处理器中断控制器的各种不同操作模式。

20 操作原理

PCI 和外围设备中断控制器有两个功能：

- 内部的外围设备中断控制
- PCI 中断路由选择

25 外围设备中断控制机构将给定外围设备的多个中断源综合成一个中断，送给本地处理器。为了向执行中的软件提供中断源信息，使用一个内存映射状态寄存器来描述中断源。所有外围设备中断都由各自的外围设备控制寄存器来单独启动。

30 PCI 中断路由选择机构允许主机软件（或本地处理器软件）为 PCI 中断选择路由，或者通向本地处理器，或者通向 P_INTA#、P_INTB#、P_INTC#和 P_INTD#输出插接线。路由选择机构是通过一个可从初级 PCI 桥配置空间或 P2P 处理器本地总线访问的内存映射寄存器被控制的。

本地处理器中断

本地处理器的中断控制器有 8 个外部中断插接线及一个非屏蔽中断插接线用于监测外部中断请求。这 8 根外部中断插接线可被配置为专用、扩



展或混合模式中的一种。在专用模式中，可单个地将插接线映射到中断向量上。扩展模式的中断线可解释为能代表一个中断向量的位域，用采用这种模式的中断线可以直接请求 240 个中断向量。混合模式中 5 个插接线在扩展模式中工作，并能请求 32 个不同中断，另外还有 3 插接线在专用模工
5 中工作。

本地处理器的 9 个中断线的定义和编程选择如下：

XINT7:0#外部中断（输入）—这些线产生中断请求。插接线可由软件配置为 3 种模式：专用、扩展或混合模式。各线可编程成边沿检测输入或级检测输入。此外，也可以在程序控制下为这些线选择去抖动
10 （debouncing）模式。

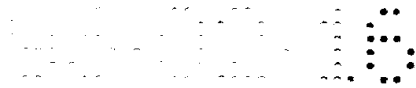
NMI#非屏蔽中断（输入）—它导致一个非屏蔽中断事件的发生。NMI 是识别出的优先级最高的中断。NMI#线是边沿激活的输入。可以在程序控制下为 NMI#选择去抖动模式。该线是内部同步的。

为使 P2P 处理器操作正确，本地处理器外部中断线必须编程成只用于
15 直接模式操作、电平敏感中断和快速取样模式。这是通过本地处理器的内存映射寄存器空间中的中断控制寄存器（ICON）完成的。《I960 Jx 微处理器用户手册》中有关于对本地处理器中断控制器编程的完备资料。

对 P2P 中断机构的利用依赖于对本地处理器中断控制器和 PCI 中断路由选择寄存器中 XINT 选择位的配置。表 9 描述了本地处理器中断控制器
20 启动的操作模式和功能，表 9 描述了 XINT 选择位的用法。

表 9. P2P 中断控制器编程概要

本地处理器中断控制器模式									
专用模式	扩展模式	混合模式	ICO N.im	电平 触发 中断	边沿 触发 中断	ICO N.sd m	取样 模式	ICO N.sd m	本地处理器中 断
启动	禁止	禁止	00 ₂	启动	禁止	0 ₂	快速	1 ₂	XINT0#/XINT A#
启动	禁止	禁止	00 ₂	启动	禁止	0 ₂	快速	1 ₂	XINT1#/XINT B#
启动	禁止	禁止	00 ₂	启动	禁止	0 ₂	快速	1 ₂	XINT2#/XINT C#



启动	禁止	禁止	00 ₂	启动	禁止	0 ₂	快速	1 ₂	XINT3#/XINTD#
启动	禁止	禁止	00 ₂	启动	禁止	0 ₂	快速	1 ₂	XINT4#
启动	禁止	禁止	00 ₂	启动	禁止	0 ₂	快速	1 ₂	XINT5#
启动	禁止	禁止	00 ₂	启动	禁止	0 ₂	快速	1 ₂	XINT6#
启动	禁止	禁止	00 ₂	启动	禁止	0 ₂	快速	1 ₂	XINT7#
N/A	N/A	N/A	N/A	N/A	缺省	N/A	快速	1 ₂	NMI#

操作框

PCI 和外围设备中断控制器与本地处理器连接，连接线路如附图 9 所示。

PCI 中断路由选择

- 5 4 个 PCI 中断输入可由多路转换器 121 选择路径由本地处理器的中断输入端或 PCI 的中断输出端。中断输入的路由选择由 PCI 中断路由选择寄存器中的 XINT 选择位控制。PCI 中断路由选择寄存器由表 10 所示。

表 10. PCI 中断路由选择寄存器概要

XINT 选择位	描述
0	INTA#/XINT0# 输入线被路由选择到本地处理器的 XINT0# 输入线 INTB#/XINT1# 输入线被路由选择到本地处理器的 XINT1# 输入线 INTC#/XINT2# 输入线被路由选择到本地处理器的 XINT2# 输入线 INTC#/XINT3# 输入线被路由选择到本地处理器的 XINT3# 输入线
1	INTA#/XINT0# 输入线被路由选择到 P_INTA# 输出线 INTB#/XINT1# 输入线被路由选择到 P_INTB# 输出线



	INTC#/XINT2#输入线被路由选择到 P_INTC#输出线
	INTC#/XINT3#输入线被路由选择到 P_INTD#输出线

前面说过，本地处理器的 XINT0# ~ XINT3# 必须编程为电平敏感的以供应 PCI 中断。此外当 XINT 选择位置位时，本地处理器输入的外部逻辑必须驱动一个不活跃电平（‘1’）。

内部外围设备中断路由选择

5 本地处理器的中断输入 XINT6#、XINT7#、NMI# 从多个内部中断源接受输入。在这三个输入端的各个之前有一个内部锁存器来对这些不同的中断源进行必要的多路转换。应用软件可以通过读相应的中断锁存器来确定中断是由哪个外围设备引起的。读取该外围设备的状态信息可以更详细地确定中断的确切原因。

10 本地处理器的中断 XINT6# 接受来自外部线和 3 个 DMA 通道的中断。各 DMA 通道中断要么是 DMA 传输结束的中断，要么是 DMA 链结束的中断。XINT6 中断锁存器 123 接收 DMA 通道的中断输入也接收外部线 XINT6#。这些中断源的任何有效中断将锁存器内的位置位，向本地处理器的 XINT6# 输入输出一个电平敏感中断。只要中断锁存器中有 1，它就应该不停地向该处理器的中断输入端驱动一个有效低输入。XINT6 中断锁存器
15 可以通过 XINT6 中断状态寄存器被读取。清除内部外围设备处的中断源就能清除 XINT6 中断锁存器。

向 XINT6 中断锁存器驱动输入的单元中断源的详细情况如表 11 所示。

20

表 11. XINT6 的中断源

单元	中断条件	寄存器	位位置
DMA 通道 0	链结束 传输结束	通道状态寄存器	08
		0 通道状态寄存器	09
DMA 通道 1	链结束 传输结束	通道状态寄存器	08
		1 通道状态寄存器	09
		1	



DMA 通道 2	链结束	通道状态寄存器	08
		2	09
	传输结束	通道状态寄存器	
		2	

本地处理器的 XINT7#中断接受来自外部中断线、APIC 总线接口单元、初级 ATU、I²C 总线接口单元和报文传递单元的中断。XINT7 中断锁存器 125 既接收上述 4 个单元的各个的一个中断输入，也接收外部线 XINT7#。这些中断源的任何有效中断将锁存器内的寄存器位置位并向本地处理器的 XINT7#输入输出一个电平敏感中断。只要中断锁存器中有 1，它就

5 就应该不停地向该处理器的中断输入端驱动一个有效低输入。XINT7 中断锁存器可以通过 XINT7 中断状态寄存器被读取。清除内部外围设备处的中断源就能清除 XINT7 中断锁存器。

10 向 XINT7 中断锁存器驱动输入的单元中断源的详细情况如表 12 所示。

表 12. XINT7 的中断源

单元	出错条件	寄存器	寄存器位位置
APIC 总线接口单元	EOI 报文已接收	APIC 控制/状态寄存器	14
	APCI 报文已发送	APIC 控制/状态寄存器	06
I ² C 总线接口单元	接受缓冲器满	I ² C 状态寄存器	07
	传输缓冲器空	I ² C 状态寄存器	06
	从属地址检测	I ² C 状态寄存器	05
	检测出“停止”	I ² C 状态寄存器	04
报文传递单元	检测出“总线错误”	I ² C 状态寄存器	03
	检测出仲裁丢失	I ² C 状态寄存器	02
初级 ATU	门铃中断	初级 ATU 中断状态寄存器	09
初级 ATU	ATU 的 BIST 开始	初级 ATU 中断状态	08



	寄存器	
--	-----	--

本地处理器的非屏蔽中断（NMI）接受来自外部线、初级和次级 ATU、初级和次级桥接口、本地处理器和三个 DMA 通道中各个通道的中断。这 8 个中断的各个表示外围设备上出现了错误条件。NMI 中断锁存器 127 既接收上述 8 个源中各个的中断输入，也接收外部线 NMI#。这些中断源的任何有效中断将锁存器内的位置位，向本地处理器的 NMI# 输入输出一个边沿触发中断。NMI 中断锁存器可以通过 NMI 中断状态寄存器被读取。清除内部外围设备处的中断源就能清除 NMI 中断锁存器。

向 NMI 中断锁存器驱动输入的单元中断源的详细情况如表 13 所示。

表 13. NMI 的中断源

单元	出错条件	寄存器	位位置
初级 PCI 桥接口	PCI 主奇偶校验错误	初级桥中断状态寄存器	00
	PCI 目标流产 (目标)	初级桥中断状态寄存器	01
	PCI 目标流产 (主)	初级桥中断状态寄存器	02
	PCI 主流产	初级桥中断状态寄存器	03
	P_SERR# 确立	初级桥中断状态寄存器	04
次级 PCI 桥接口	PCI 主奇偶校验错误	次级桥中断状态寄存器	00
	PCI 目标流产 (目标)	次级桥中断状态寄存器	01
	PCI 目标流产 (主)	次级桥中断状态寄存器	02
	PCI 主流产	次级桥中断状态寄存器	03
	S_SERR# 确立	次级桥中断状态寄存器	04



初级 ATU	PCI 主奇偶校验 错误	初级 ATU 中断状态寄 存器	00
	PCI 目标流产 (目 标)	初级 ATU 中断状态寄 存器	01
	PCI 目标流产 (主)	初级 ATU 中断状态寄 存器	02
	PCI 主流产	初级 ATU 中断状态寄 存器	03
	P_SERR#断言	初级 ATU 中断状态寄 存器	04
	本地处理器总线 故障	初级 ATU 中断状态寄 存器	05
	本地处理器内存 故障	初级 ATU 中断状态寄 存器	06
次级 ATU	PCI 主奇偶校验 错误	次级 ATU 中断状态寄 存器	00
	PCI 目标流产 (目 标)	次级 ATU 中断状态寄 存器	01
	PCI 目标流产 (主)	次级 ATU 中断状态寄 存器	02
	PCI 主流产	次级 ATU 中断状态寄 存器	03
	S_SERR#确立	次级 ATU 中断状态寄 存器	04
	本地处理器总线 故障	次级 ATU 中断状态寄 存器	05
	本地处理器内存 故障	次级 ATU 中断状态寄 存器	06
本地处理器	本地处理器总线 故障	本地处理器状态寄存 器	05
	本地处理器内存	本地处理器状态寄存 器	06



	故障	器	
DMA 通道 0	PCI 主奇偶校验 错误	通道状态寄存器 0	00
	PCI 目标流产 (主)	通道状态寄存器 0	02
	PCI 主流产	通道状态寄存器 0	03
	本地处理器总线 故障	通道状态寄存器 0	05
	本地处理器内存 故障	通道状态寄存器 0	06
DMA 通道 1	PCI 主奇偶校验 错误	通道状态寄存器 1	00
	PCI 目标流产 (主)	通道状态寄存器 1	02
	PCI 主流产	通道状态寄存器 1	03
	本地处理器总线 故障	通道状态寄存器 1	05
	本地处理器内存 故障	通道状态寄存器 1	06
DMA 通道 2	PCI 主奇偶校验 错误	通道状态寄存器 2	00
	PCI 目标流产 (主)	通道状态寄存器 2	02
	PCI 主流产	通道状态寄存器 2	03
	本地处理器总线 故障	通道状态寄存器 2	05
	本地处理器内存 故障	通道状态寄存器 2	06

下面描述路由选择寄存器、XINT6 中断状态寄存器、XINT7 中断状态寄存器、NMI 中断状态寄存器。

P2P 处理器外部中断接口



P2P 处理器的外部中断输入接口由以下引线组成:

表 14. 中断输入线描述

信号	描述
INTA#/XINT0#	该中断输入可被传递到 P_INTA# 输出或本地处理器的中断输入 XINT0
INTB#/XINT1#	该中断输入可被传递到 P_INTB# 输出或本地处理器的中断输入 XINT1
INTC#/XINT2#	该中断输入可被传递到 P_INTC# 输出或本地处理器的中断输入 XINT2
INTD#/XINT3#	该中断输入可被传递到 P_INTD# 输出或本地处理器的中断输入 XINT3
XINT4#	该中断输入总是连到本地处理器的中断输入 XINT4
XINT5#	该中断输入总是连到本地处理器的中断输入 XINT5
XINT6#	该中断输入由 3 个内部中断共享。这些中断是每个内部 DMA 通道的中断。所有这些中断都被传递到本地处理器的中断输入 XINT6#。要确定确切的中断源，软件必须读 XINT6 中断状态寄存器。
XINT7#	该中断输入由 4 个内部中断共享。这些中断是 APIC 总线接口单元、初级 ATU、I ² C 总线接口单元和报文传递单元发出的中断。所有这些中断都被传递到本地处理器的中断输入 XINT7#。要确定确切的中断源，软件必须读 XINT7 中断状态寄存器。
NMI#	该中断输入由 8 个内部中断共享。这些中断包括初级和次级 ATU、初级和次级 PCI 桥接口、本地处理器和三个 DMA 通道中各个通道发出的采集的出错中断。所有这些中断都被传递到本地处理器的输入 NMI#。要确定确切的中断源，软件



	必须读 NMI 中断状态寄存器。
--	------------------

PCI 向外门铃中断

P2P 处理器具有在任意一个初级 PCI 中断线上生成中断的能力。这是通过对初级 ATU 内部门铃端口寄存器中的一个位进行置位来完成的。位 0 至 3 分别对应 P_INTA# 至 P_INTD#。对某个寄存器位进行置位就生成对应的 PCI 中断。

寄存器定义

PCI 和外围设备中断控制器有 4 个控制和状态寄存器：

- PCI 中断路由选择寄存器
- XINT6 中断状态寄存器
- XINT7 中断状态寄存器
- NMI 中断状态寄存器

每个寄存器都是一个 32 位寄存器，并且是本地处理器存储器空间中的内存映射寄存器。

所有寄存器作为 P2P 的内存映射寄存器都是可见的，可以通过内部存储器总线进行存取。PCI 中断路由选择寄存器可以从内部存储器总线及通过 PCI 配置寄存器空间访问（功能 0#）。

PCI 中断路由选择寄存器—PIRSR

PCI 中断路由选择寄存器（PIRSR）决定外部输入线的路由选择。输入线由 4 个次级 PCI 中断输入组成，它们被选择路径到初级 PCI 中断或本地处理器中断。PCI 中断线被定义为“电平敏感”中断，确立电平是低电平。中断线的确立和撤销与 PCI 或处理器时钟是异步的。

如果次级 PCI 中断输入被选择路径到初级 PCI 中断线，本地处理器的中断输入线 XINT3:0# 必须被置于不活跃状态。

XINT6 中断状态寄存器—X6ISR

XINT6 中断状态寄存器（X6ISR）表示当前挂起的 XINT6 中断。XINT6 的中断源可以通过 XINT6 中断锁存器或外部 XINT6# 输入线连接的内部外围设备。上文中内部外围设备中断路由选择部分描述了可以在 XINT6# 输入上生成的那些中断。

X6ISR 被应用软件用于确定 XINT6# 输入上中断的中断源以及清除该中断。该寄存器中的所有位都是定义为只读的。寄存器中的所有位都在对应的中断源（表 11 中所示的状态寄存器源）被清除时清零。X6ISR 反映



着 XINT6 中断锁存器的输入的当前状态。

由于 P2P 外围设备单元的异步性，当应用软件读 X6ISR 寄存器时，可能有多个中断处于活跃状态。应用软件必须恰当地对待这些多重中断条件。此外，应用软件可以随后读 X6ISR 寄存器，以确定在上个中被处理期间是否又有中断发生。来自 X6ISR 寄存器的所有中断在本地处理器中都具有相同的优先级（《i960 JX 微处理器用户手册》中有对优先级机制设置的说明）。

表 15 详细定义了 X6ISR。

表 15. XINT6 中断状态寄存器— X6ISR

位	缺省值	读/写	说明
31:04	0000000H	只读	保留
03	0 ₂	只读	外部 XINT6# 中断挂起—如果该位置位，则外部 XINT6# 输入上有挂起的中断。如果清零不存在中断。
02	0 ₂	只读	DMA 通道 2 中断挂起—如果该位置位，则 DMA 通道 2 已经发出一个通道链结束的活跃条件信号。如果清零则不存在中断条件。
01	0 ₂	只读	DMA 通道 1 中断挂起—如果该位置位，则 DMA 通道 1 已经发出一个通道链结束的活跃条件信号。如果清零则不存在中断条件。
00	0 ₂	只读	DMA 通道 0 中断挂起—如果该位置位，则 DMA 通道 0 已经发出一个通道链结束的活跃条件信号。如果清零则不存在中断条件。

10 XINT7 中断状态寄存器— X7ISR

XINT7 中断状态寄存器（X7ISR）表示当前挂起的 XINT7 中断。



XINT7 的中断源可以通过 XINT7 中断锁存器或外部 XINT7# 输入线连接的内部外围设备。

X7ISR 被应用软件用于确定 XINT7# 输入上中断的中断源以及清除该中断。该寄存器中的所有位都定义为只读的。寄存器中的所有位都在对应的中断源（表 12 中所示的状态寄存器源）被清除时清零。X7ISR 反映着 XINT7 中断锁存器的输入的当前状态。

由于 P2P 外围设备单元的异步性，当应用软件读 X7ISR 寄存器时，可能有多个中断处于活跃状态。应用软件必须恰当地对待这些多重中断条件。此外，应用软件可以随后读 X7ISR 寄存器，以确定在上个中被处理期间是否又有中断发生。来自 X7ISR 寄存器的所有中断在本地处理器中都具有相同的优先级。

表 16 详细定义了 X7ISR。

表 16. XINT7 中断状态寄存器— X7ISR

位	缺省值	读/写	说明
31:05	0000000H	只读	保留
04	0 ₂	只读	外部 XINT7# 中断挂起—如果该位置位，则外部 XINT7# 输入上有中断挂起。如果清零不存在中断。
03	0 ₂	只读	初级 ATU/ 开始 BIST 中断挂起—如果该位置位，则主处理器已经在寄存器 ATUBISTR 中设置了开始 BIST 的请求。如果清零则没有开始 BIST 中断挂起。
02	0 ₂	只读	向内门铃中断挂起—如果该位置位，则向内报文传递单元发出的中断正在挂起。如果清零则没有中断挂起。
01	0 ₂	只读	I ² C 总线中断挂起—如果该位置位，则 I ² C 总线接口单元单元发出的中断正在挂起。如果清零则没有中断挂起。



00	0 ₂	只读	APIC 中断挂起—如果该位置位, 则 APIC 总线接口单元发出的中断正在挂起。如果清零则没有中断挂起。
----	----------------	----	---

NMI 中断状态寄存器—NISR

NMI 中断状态寄存器 (NISR) 表示当前挂起的 NMI 中断。NMI 的中断源可以通过 NMI 中断锁存器或外部 NMI# 输入线连接的内部外围设备。

5 NISR 被应用软件用于确定 NMI# 输入上中断的中断源以及清除该中断。该寄存器中的所有位都是只读的。寄存器中的所有位都在对应的中断源 (表 13 中所示的状态寄存器源) 被清除时清零。NISR 反映着 NMI 中断锁存器的输入的当前状态。

10 由于 P2P 外围设备单元的异步性, 当应用软件读 NISR 寄存器时, 可能有多个中断处于活跃状态。应用软件必须恰当地对待这些多重中断条件。此外, 应用软件可以随后读 NISR 寄存器, 以确定在上个中被处理期间是否又有中断发生。来自 NISR 寄存器的所有中断在本地处理器中都具有相同的优先级。

表 17 示出了 NMI 中断状态寄存器的各定义位。

15

表 17. NMI 中断状态寄存器—NISR

位	缺省值	读/写	说明
31:09	000000H	只读	保留
08	0 ₂	只读	外部 NMI# 中断挂起—如果该位置位, 则外部 NMI# 输入上有中断挂起。如果清零不存在中断。
07	0 ₂	只读	DMA 通道 2 错误—如果该位置位, 则 DMA 通道 2 内存在一个 PCI 或本地总线错误条件。若清零则无出错条件。
06	0 ₂	只读	DMA 通道 1 错误—如果该位置位, 则 DMA 通道 1 内存在一个 PCI 或本地总线错误条件。若清



			零则无出错。
05	0 ₂	只读	DMA 通道 0 错误—如果该位置位，则 DMA 通道 0 内存在一个 PCI 或本地总线错误。若清零则无出错。
04	0 ₂	只读	次级桥错误—如果该位置位，则桥的次级接口内存在一个 PCI 错误条件。若清零则无出错。
03	0 ₂	只读	初级桥接口错误—如果该位置位，则桥的初级接口内存在一个 PCI 错误条件。若清零则无出错。
02	0 ₂	只读	次级 ATU 错误—如果该位置位，则次级 ATU 内存在一个 PCI 或本地总线错误条件。若清零则无出错。
01	0 ₂	只读	初级 ATU 错误—如果该位置位，则初级 ATU 内存在一个 PCI 或本地总线错误条件。若清零则无出错。
00	0 ₂	只读	本地处理器错误—如果该位置位，则内部存储器控制器内存在一个由本地处理器引起的错误条件。若清零则无出错。

内部仲裁

以下描述 P2P 处理器的内部仲裁—包括对处理器中内部本地总线的仲裁以及对处理器内各个 PCI 接口的仲裁，描述的内容有仲裁的操作模式、建立、外部存储器接口和实现。

5 本地总线仲裁

P2P 处理器需要有一个仲裁机构来控制本地总线的所有权。连接到本地总线的总线主由 3 个 DMA 通道、初级 PCI 地址翻译单元、次级 PCI 地址翻译单元、本地处理器和外部总线主所组成

本地总线仲裁单元 (LBAU) 57 实现一种公平算法, 允许每一个总线



主有机会取得对本地总线的控制权。该算法在轮式调度方案中加入了一种优先化的机制。在最佳实施例中，实现应允许应用软件独立地向各个本地总线主分配优先权。

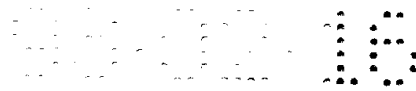
本地总线仲裁单元的责任是将本地总线授予总线主。所有总线主都有一个逻辑，一旦它们已经失去自己的内部 GNT#信号，就要从本地总线上卸任自己作为总线主。有一个可编程的 12 位计数器，用于限制总线主控制本地总线的时间量，它指示总线主在有其它总线主请求总线的情况下，何时必须交出占有权。

外部总线主可以用于本地总线上，方法是增加控制 HOLD/HOLDA 的外部逻辑。P2P 处理器允许有一个外部总线主加入公平算法。如果本地总线上使用的外部总线主不止一个，就需要用外部逻辑将所有外部设备当作一个设备来对待（检测 HOLD，驱动 HOLDA）。

本地总线仲裁单元控制本地处理器的补偿单元(backoff unit)。补偿单元允许本地处理器被“补偿”，以防止可能发生的死锁现象。处理器补偿时，保持在等待状态（L_RDYRCV#不活跃）。内部缓冲器使多路转换地址/数据总线处于三稳态，使其它本地总线主（ATU、DMA 等）控制总线，因而就避免出现发生向外事务处理要求使用正被向内事务处理使用的资源这种情况。此外，补偿单元使本地总线的性能在所有向外的处理器读操作期间达到最优。

除了本地总线仲裁单元之外，P2P 处理器还有两个本地 PCI 仲裁单元。本地初级仲裁单元 55a 控制对内部初级 PCI 总线的访问。仲裁在 PCI 至 PCI 桥单元的初级 ATU、DMA 通道 0 与 1、初级接口之间争夺初级 PCI 总线时发生。本地次级仲裁单元 55b 控制对内部次级 PCI 总线的访问。仲裁在 PCI 至 PCI 桥单元的次级 ATU、DMA 通道 2、次级接口之间争夺次级 PCI 总线时发生。这两个本地 PCI 仲裁单元都以相似的方式工作。理想的情况是，仲裁逻辑被设计成允许多个总线主控制本地总线。当有总线主对本地总线提出请求时，本地总线仲裁单元首先应通过确立一个 HOLD 请求信号，从本地处理器得到本地总线的控制权。本地处理器应将总线交给仲裁逻辑，方法是确立 HOLDA 信号并将处理器信号线置于三态模式。仲裁逻辑接着就应将本地总线交给其它总线主，方法是返回对应的内部 GNT#信号。

内部 PCI 总线仲裁



P2P 处理器中有两个内部仲裁单元。控制对设备内的内部 PCI 总线的访问。附图 10 表示这些内部仲裁单元以及它们所控制的资源。

初级内部 PCI 仲裁单元为下列内部单元进行仲裁：

- 初级桥接口
- 初级 ATU
- DMA 通道 0
- DMA 通道 1

次级内部 PCI 仲裁单元为下列内部单元进行仲裁：

- 次级桥接口
- 次级 ATU
- DMA 通道 2

各内部 PCI 仲裁单元采用固定的轮式仲裁方案，总线上的每个设备具有相同的优先权。

对固定的轮式仲裁的解释如下：

- 复位后，仲裁令牌属于各内部 PCI 仲裁单元内的设备#1。
- 在有设备向仲裁单元确立内部 REQ#的每个时钟上执行仲裁。
- 令牌的下一个所有者（例如总线）是设备号与当前令牌所有者（总线空闲时则为上一个令牌所有者）最接近的设备。例如，如果设备#3 是当前所有者，当设备#4、设备#1 都提出总线请求，则设备#4 赢出。

· 当仲裁单元向内部总线主控启动内部授权时，传递令牌。这是仲裁单元的授权信号。向该内部总线主的实际输出仍然用外部 GNT#输入屏蔽着。

操作原理

内部 PCI 总线上的各单元都要请求使用总线进行主控操作。每当附属
于内部总线的资源发出请求（REQ#）时，便出现仲裁。向下一个资源的
授权是按照轮式方案作出的。内部 PCI 总线的授予与外部 PCI 总线的状态
关系密切。外部中断请求线（P_REQ#或 S_REQ#）的状态是各内部 PCI
总线上请求线的逻辑“或”的直接反映。

内部 PCI 总线主可以在任何时候接受内部 PCI 仲裁单元的内部 GNT#
（幕后仲裁）。内部总线主仍然负责继续监视 FRAME#、IRDY#、和它们的
内部 GNT#输入以保证在开始访问总线之前保持对总线的所有权。在时
钟脉冲的上升沿，FRAME#、IRDY#必须是高电平，授权输入信号必须是



低电平，就是这个上升沿确定了总线主然后驱动 FRAME#低电平以开始一个周期的那个时钟周期。内部 PCI 仲裁单元对外部授权信号（ P_REQ#或 S_REQ#）进行监控，仅当在外部授权信号为真的基础上进行内部授权。

内部 PCI 仲裁单元在下列情形撤销内部总线主的 GNT#:

- 外部授权信号（ P_GNT#或 S_GNT#）转为不活跃状态。
- 内部仲裁单元将内部总线主的内部授权信号变为不活跃状态。
- 当前的总线所有者撤销其 REQ#输出。

各总线主的授权输入可被视为外部授权和来自内部 PCI 总线仲裁单元内部的内部授权的逻辑“或”。仲裁单元根据谁是轮式方案决定的赢者向内部总线主激活内部授权信号，但驱动到内部总线主的授权输入的实际 GNT#信号是从外部授权和内部授权输入的逻辑“或”得出的。

内部仲裁单元负责保证，任何时刻只有一个内部 GNT#是激活的。一旦某个内部总线主失去了内部 GNT#信号，它最终就必须释放总线的所有权。内部 GNT#信号遵守《 PCI 本地总线规范》关于 GNT#信号撤销的规则（仲裁信号协议）。本发明领域的熟练人员显然完全能够掌握实现内部 PCI 仲裁单元所需的其它细节。

I²C 总线接口单元

以下描述 P2P 处理器的 I²C（互连集成电路）总线接口单元，内容是 I²C 总线接口单元的操作模式、建立和实现。

20 概述

I²C 总线接口单元 61 允许本地处理器 34 担当 I²C 总线上的主和从设备。I²C 总线是 Philips 公司开发的一种串行总线，由一种双插线接口组成。SDA 是数据线，用于实现输入输出功能；SCL 是时钟线，用于对 I²C 总线的参照与控制。

25 I²C 总线允许 P2P 处理器与其它 I²C 外围设备和微控制器接口，实现系统的管理功能。该串行总线能以最少的硬件构造一个经济的系统，用于向外部设备传递 P2P 子系统上的状态和可靠性数据。

30 I²C 总线接口单元是一种位于内部 P2P 本地总线上外围设备。I²C 总线接受或发送出去的数据是经由一个缓冲接口传输的。控制和状态数据是通过本地处理器的一组内存映射寄存器传递的。《 I²C 总线规范》对 I²C 总线的操作有内容完备的说明。

操作原理

I²C 总线为在仅使用双线接口的总线上的代理之间交换信息定义了一个完整的串行协议。总线上的各个设备均由唯一的 7 位地址识别，既可以作为信息的发送方，也可以作为接收方。I²C 总线除了在发送方和接收方之间起作用外，还在主/从模式中发挥功能。

举一个 I²C 总线操作的例子，假设微处理器担当总线上主。作为主的微处理器可能要寻址作为从设备的接收写数据的 EEPROM。此时微处理器就是主—发送方，EEPROM 就是从—接收方。假若微处理器想要读数据，则微处理器就是主—接收方，EEPROM 就是从—发送方。在这两种情况下，都是由主来启动和结束事务处理。

I²C 总线允许系统有多个主，就是说，可以同时有多个设备试图启动数据传输。I²C 总线为处理这种情况定义仲裁过程。如果有两个或两个以上的主同时驱动总线，当其它主产生一个 0 时产生一个 1 的第一个主就会在仲裁中失败。这取决于 SDA 和 SCL I²C 总线线路的“线—与”（wired-AND）操作。

I²C 总线的串行操作使用一种线—与总线结构。这是用于多个设备驱动总线线路及互相之间发信号，告知诸如有关仲裁结果、等待状态、出错条件等事件的方法。例如，当一个主在数据传输期间驱动时钟（SCL）线路时，它在每个时钟脉冲的高电平时刻传输一个数据位。如果从设备不能按主要求的速率接收或驱动数据，它可以将时钟线的电平在高电平之间保持在低电平上以实质上插入等待状态。线—与操作在设备的输出阶段执行的。I²C 总线上的最大数据传输速度是每秒 400K 位。

I²C 总线事务处理由本地处理器作为主而启动或者由本地处理器作为从设备而接收。这两个条件都会导致处理器对 I²C 总线进行读操作、写操作或者读、写操作兼有。

操作框

P2P 处理器的 I²C 总线接口单元是一个与本地总线相连的从外围设备。该单元用 P2P 处理器的中断机制来向本地处理器通知 I²C 总线上的活动。附图 11 是 I²C 总线接口单元及其与本地总线接口的方框图。

组成 I²C 总线接口单元的是两个连接到 I²C 总线的线接口 61、用于对本地处理器来回传输数据的 8 位缓冲器 61a，一组控制和状态寄存器 61b，一组用于并行/串行转换的移位寄存器 61c。



I²C 中断的信号是通过处理器中断 XINT7#和 XINT7 中断状态寄存器 (X7ISR) 发出的。在缓冲器满、缓冲器空、检测到从地址、仲裁失败或总线出错等情况发生时, I²C 总线接口单元将寄存器 X7ISR 内的一位置位。所有中断条件都是由本地处理器明确地清除的。

I²C 数据缓冲寄存器 IDBR 是一个 8 位数据缓冲器, 它从一端接收来自 I²C 总线的移位寄存器接口的一个字节的数, 并从另一端接收来自 P2P 处理器本地总线的并行数据。用户不可以访问串行移位寄存器。

APIC 总线接口单元

以下描述 APIC 总线接口单元 63。该单元提供本地总线与 3 线的 APIC 总线之间的通信机构, 它有两个基本功能:

它使本地处理器能够将中断报文发送到 APIC 总线, 并可以有选择地在该报文发出后被中断。本地处理器然后就能读取报文传输的结果状态, 检查是否出现错误。

它也能接收 APIC 总线的 EOI 报文, 并可以选择中断本地处理器, 向本地处理器报告已经有了 EOI 向量。

下面叙述该接口的操作模式、建立和实现。

APIC 体系结构概述

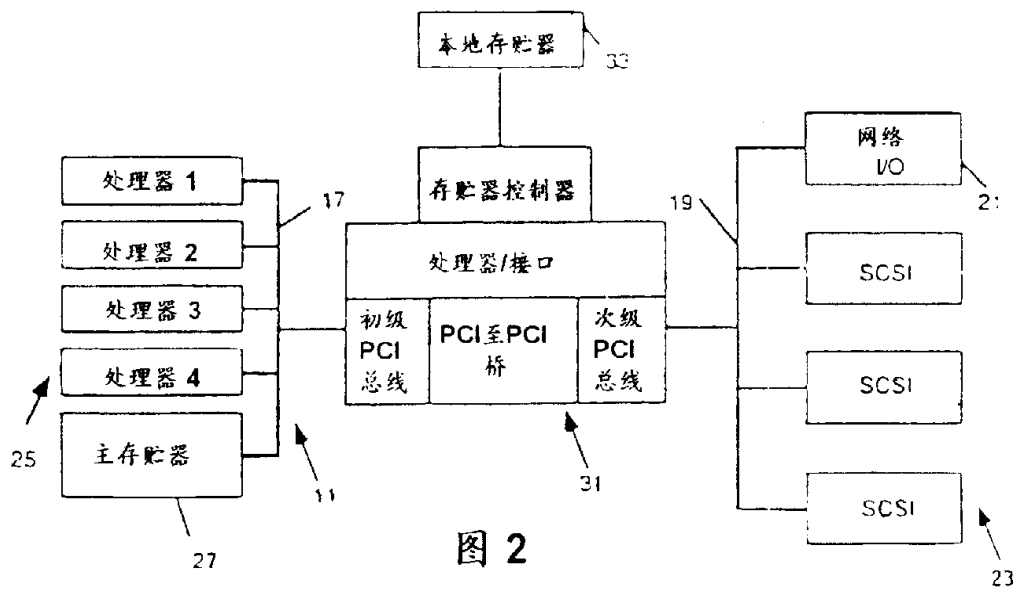
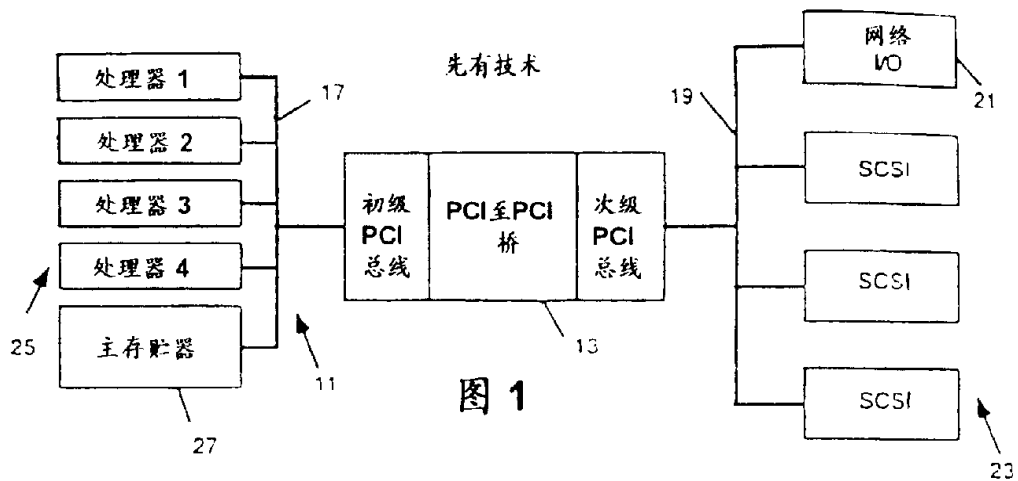
APIC 中断体系结构被规定为所有与《多处理器规范 (MPS)》兼容的系统的中断体系结构。Intel 公司出版了 MPS1.1 版, 订购号为 242016-003。APIC 体系结构的主要特点是:

1. APCI 为 Intel 体系结构的 CPU - 例如 90 和 100MHZ 的奔腾处理器 - 提供多处理器中断管理, 提供跨越所有处理器的静态和动态的对称中断分配。
2. 对称中断分配包括将中断向最低优先级处理器的路由选择。
3. APIC 在具有多个 I/O 子系统的系统中工作, 每个子系统都可以有自己的中断集。
4. APCI 提供处理器间中断, 允许处理器中断包括其自身的任何处理器或处理器组。
5. 各个 APIC 中断输入线都可以独立地由软件编程为边沿触发型或电平触发型。中断向量和中断导引 (steering) 信息可以按管脚引线作规定。
6. APIC 支持由软件来裁剪成适应不同的系统体系结构和不同使用模型的命名/寻址方案。

7. APIC 支持与 NMI、INIT 和系统管理中断 (SMI) 有关的全系统的处理器控制功能。
 8. APIC 与 8259A 型 PIC 共存, 保持了 PC 的兼容性。
 9. APIC 为每个中断输入线提供可编程的中断优先级(向量)。因为 APIC
- 5 编程接口由两个 32 位的内存单元组成, 可以用 P2P 处理器中的本地处理器模拟 I/O APIC 的功能。

本发明领域的熟练人员显然完全能够掌握适合与本发明一起使用的 I/O APIC 的具体实现细节。

说明书附图



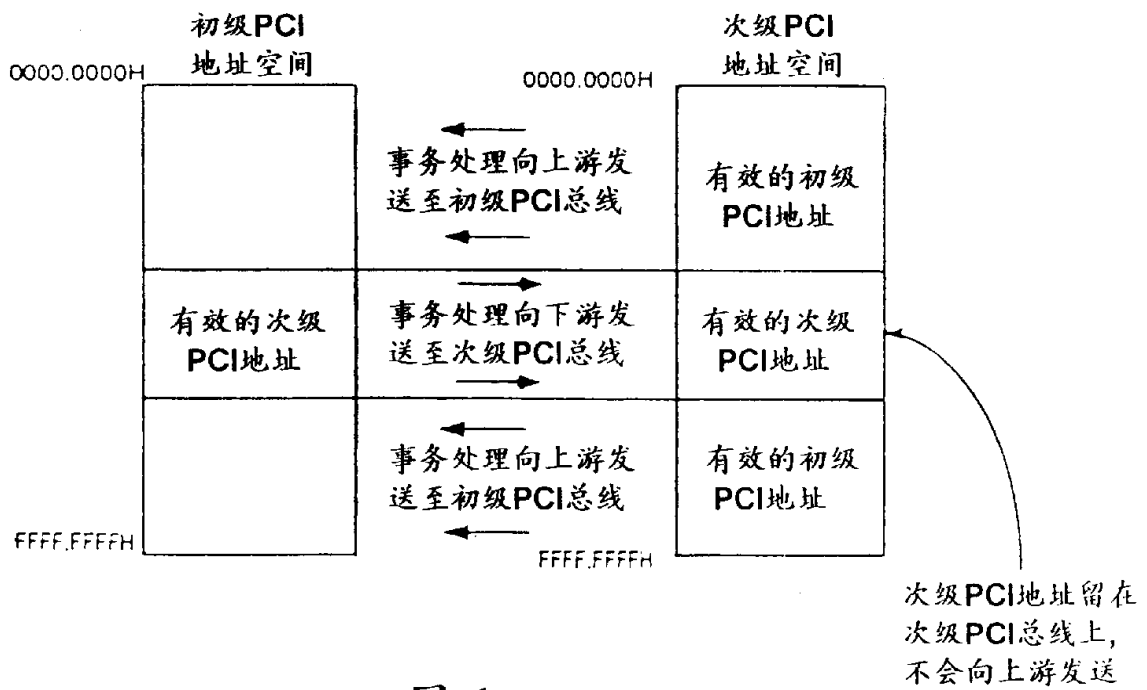


图 4

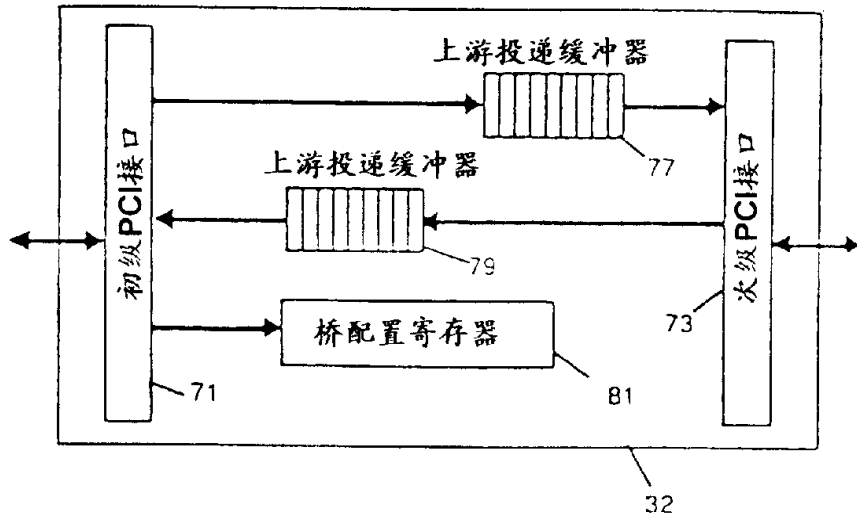


图 5

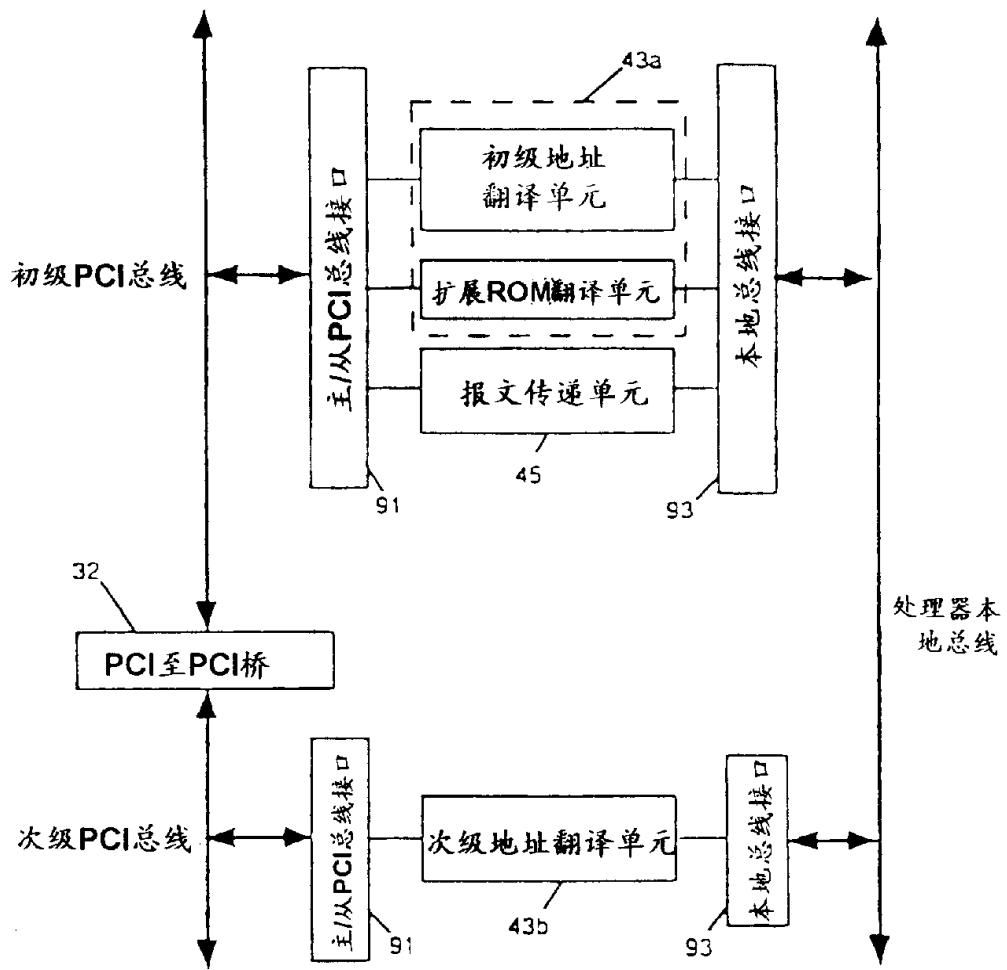


图 6

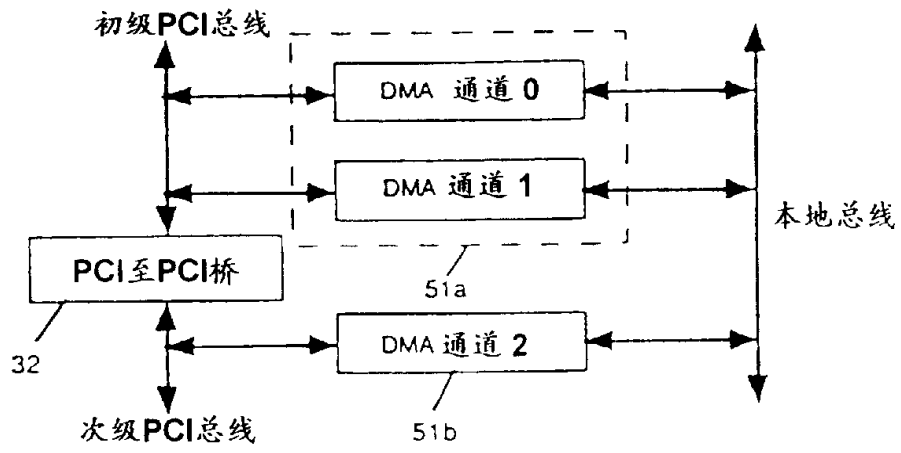


图 7

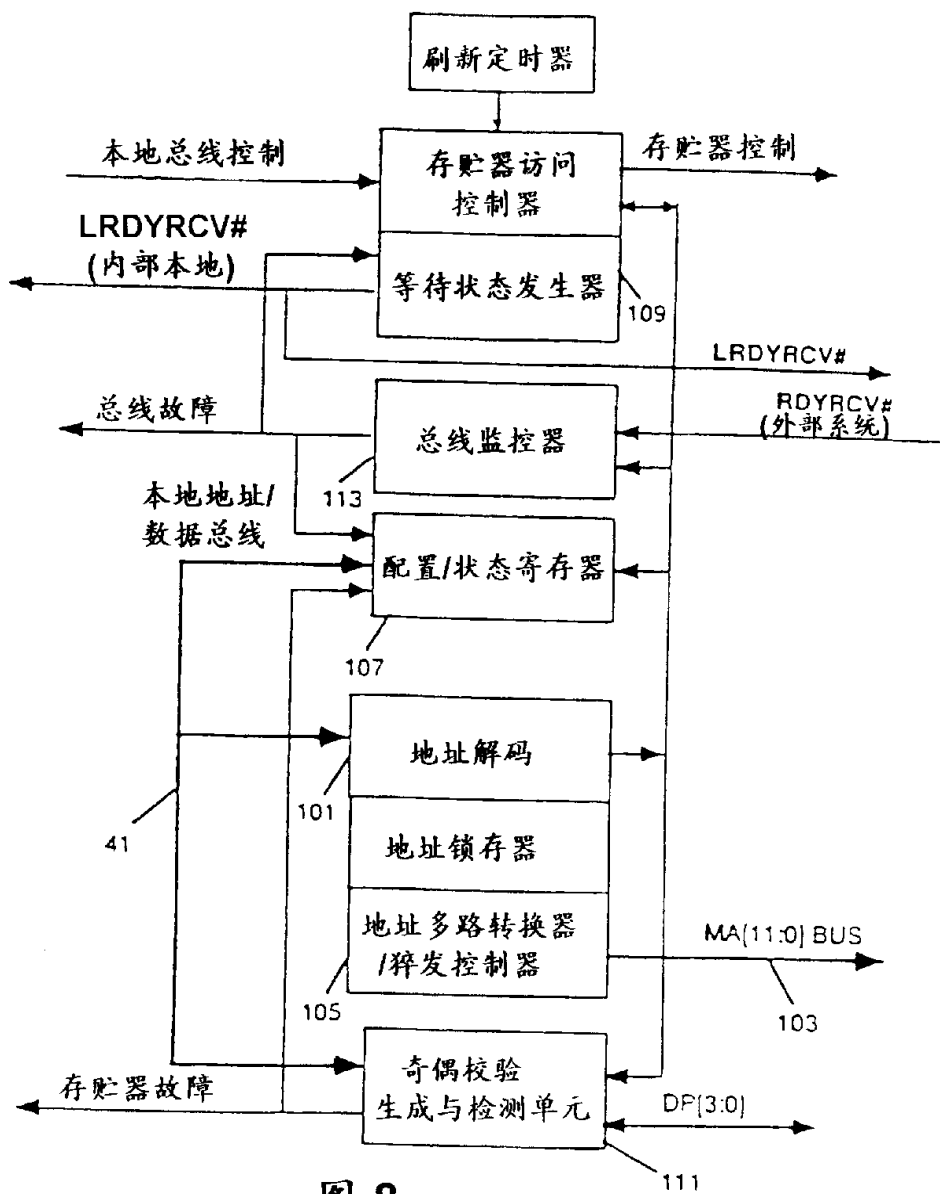


图 8

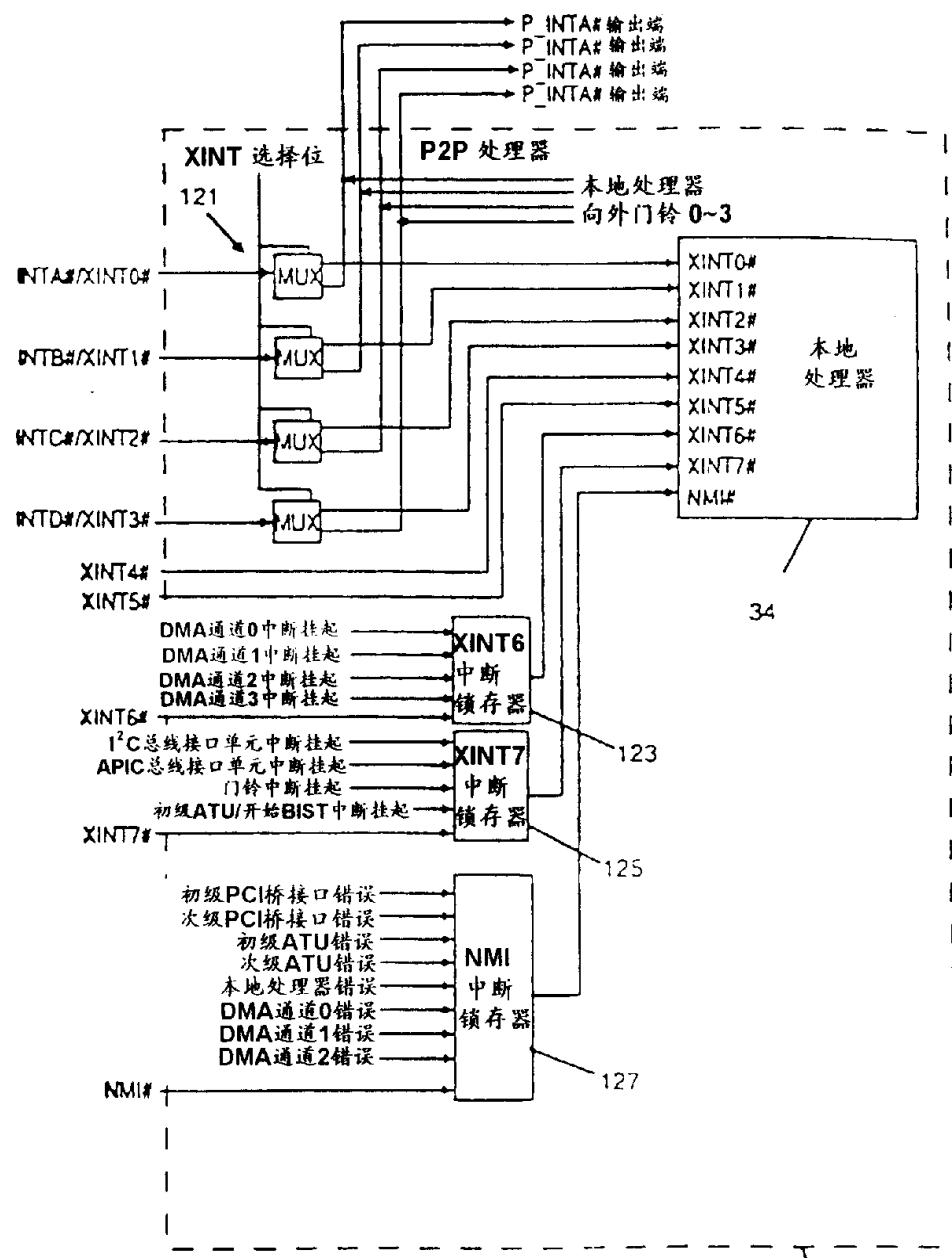


图 9

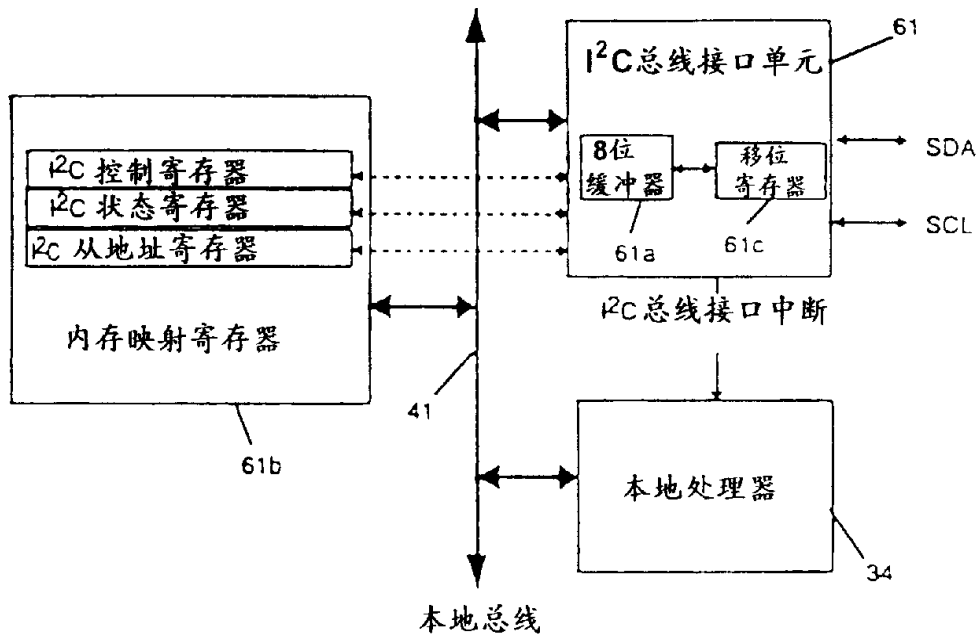


图 11