

[19]中华人民共和国国家知识产权局

[51]Int. Cl<sup>6</sup>

G06F 13/00

G06F 9/00

# [12] 发明专利申请公开说明书

[21] 申请号 97194540.3

[43]公开日 1999年6月2日

[11]公开号 CN 1218565A

[22]申请日 97.3.6 [21]申请号 97194540.3

[30]优先权

[32]96.3.13 [33]US [31]08/614,659

[86]国际申请 PCT/US97/03667 97.3.6

[87]国际公布 WO97/34235 英 97.9.18

[85]进入国家阶段日期 98.11.10

[71]申请人 钻石多媒体系统公司

地址 美国加利福尼亚州

[72]发明人 斯里德哈·贝格尔 詹姆斯·K·吉福德

阿德里安·刘易斯 唐纳德·J·斯潘塞

托马斯·E·基尔伯恩

丹尼尔·B·戈克诺尔

[74]专利代理机构 柳沈知识产权律师事务所

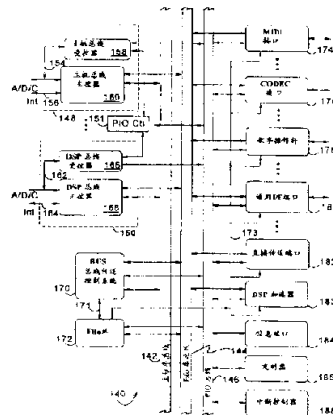
代理人 孙履平

权利要求书 3 页 说明书 53 页 附图页数 8 页

[54]发明名称 多路径 FIFO 库缓冲器以及总线传送控制系统

[57]摘要

一种总线传送控制系统通过一个缓冲池管理着多个异步数据流的传送,这一总线传送控制系统包括一个具有许多存储块的缓冲池,其中的每个存储块可以存储多个数据字节,并且把多个数据传送设备连接到缓冲池上,通过缓冲池来传送需要在多个数据传送设备之间传送的一或多个数据流段,用一个传送控制器来维持关于存储块中的数据状态的状态信息,它包括用来反复计算这种状态信息的控制逻辑,并且可以优先选择一个第一数据传送设备和一个预定的存储块。



ISSN 1008-4274



## 权 利 要 求 书

1. 通过一个缓冲器对多个异步数据流的传送进行管理的一种总线传送控制系统，上述总线传送控制系统包括：

5 a) 含有多个存储块的一个缓冲池，其中的每个上述存储块可提供储存多个数据字节；

b) 连接到上述缓冲池上的多个数据传送设备，以便允许通过上述缓冲池来传送需要在上述多个数据传送设备之间传送的一个或多个数据流段；

10 c) 一个传送控制器，用来维持关于上述存储块中的数据状态的状态信息，上述传送控制器包括用来反复计算上述状态信息的控制逻辑，并且可提供优先的选择一个第一数据传送设备和预定的一个上述存储块。

2. 按照权利要求 1 的总线传送控制系统，其特征是上述第一数据传送设备通过请求来自上述传送控制器的程序，以便获得上述状态信息中的预定部分，并且在上述缓冲池和上述第一数据传送设备之间完成了预定数据流的  
15 预定段的传送之后更新上述状态信息的上述预定部分。

3. 按照权利要求 2 的总线传送控制系统，其特征是上述多个数据传送设备包括第二数据传送设备，其中的上述第二数据传送设备通过请求来自上述传送控制器的程序，以便获得上述状态信息中的另一个预定部分，并且在上述缓冲池和上述第二数据传送设备之间完成了另一个预定数据流的另  
20 一个预定段的传送之后更新上述状态信息的上述另一个预定部分。

4. 按照权利要求 3 的总线传送控制系统，其特征是上述第一和第二预定数据流是相同的数据流。

5. 按照权利要求 3 或 4 的总线传送控制系统，其特征是把上述状态信息作为多个状态单元存储在一个状态表中，其中的上述传送控制器包括连接到上述状态表的一个扫描器，用于在分别为上述第一和第二数据传送设备编  
25 程时迭代地选择最高优先权的第一和第二状态单元。

6. 按照权利要求 5 的总线传送控制系统，其特征是上述传送控制器维持上述第一和第二状态单元各自的标识符，用于更新上述第一和第二状态单元，与分别用于对上述第一和第二数据传送设备编程时用来选择上述最高优  
30 先权的第一和第二状态单元的下一代无关。

7. 一种总线主控数据传送控制系统，通过一个判优的存取数据传送总



线在上述总线主控数据传送控制系统可以存取的一个存储器与一个缓冲池之间传送数据段，上述总线主控数据传送控制系统包括：

a) 可提供储存多个数据包的缓冲池；

5 b) 第一总线主控单元，可以连接到第一判优存取数据传送总线上，上述第一总线主控单元可提供在上述第一判优存取数据传送总线和第一总线主控数据包缓冲器之间以及在上述第一总线主控数据包缓冲器和上述缓冲池之间传送第一预定数据包；

10 c) 第二总线主控单元，可以连接到第二判优存取数据传送总线上，上述第二总线主控单元可提供在上述第二判优存取数据传送总线和第二总线主控数据包缓冲器之间以及在上述第二总线主控数据包缓冲器和上述缓冲池之间传送第二预定数据包；以及

d) 一个总线传送控制器，它包括了用于存储第一和第二预定总线传送控制单元的表，在表中分别关于上述缓冲池的上述第一和第二总线主控单元来规定总线主控数据的传送。

15 8. 按照权利要求 7 的总线主控数据传送控制系统，其特征是，上述第一和第二预定的总线传送控制单元有计划地与上述的第一和第二总线主控单元和上述缓冲池中一个预定的存储块相关联。

20 9. 按照权利要求 8 的总线主控数据传送控制系统，其特征是进一步包括一个判优系统，用于对上述第一和第二总线传送控制单元进行选择，以便分别用来对传送上述第一和第二预定数据包的上述第一和第二总线主控单元之一进行编程。

25 10. 按照权利要求 9 的总线主控数据传送控制系统，其特征是，上述总线传送控制器通过上述缓冲池来管理上述第一和第二数据包的传送，其中的上述判优系统被连接到上述总线传送控制器，用来决定对上述第一和第二总线传送控制单元的选择。

11. 按照权利要求 10 的总线主控数据传送控制系统，其特征是上述第一和第二预定数据包代表着数据包的第一数据流中的单一的数据包，并且用存储在上述表中的第三总线传送控制单元来规定数据包的第三数据流中的第三预定数据包的第三总线主控数据传送。

30 12. 一种 I/O 通道控制器，在一个存储器和多个 I/O 接口单元之间用来支持多个数据流的异步传送上述 I/O 通道控制器包括：



a) 一个 FIFO 库缓冲器, 包括多个动态分配的 FIFO 缓冲器用来对各个数据流执行异步的起动和传送;

b) 多个 I/O 接口单元; 以及

c) 一个传送控制单元, 可提供储存一个或多个传送控制数据单元上述

5 传送控制单元包括用于相关于预定的门限量来监视被任何一个上述动态分配的 FIFO 缓冲器所缓冲的数据量的控制逻辑, 并且用来在上述存储器和上述 FIFO 库缓冲器之间有选择地开始传送由单独的一个上述传送控制数据单元所限定的一部分上述数据流。

# 说明书

## 多路径FIFO库缓冲器以及 总线传送控制系统

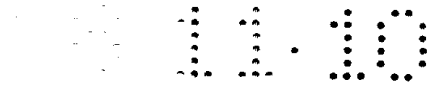
5 本发明主要涉及到高性能的外围设备数据接口，特别是涉及一种多路径的，由总线主控的输入/输出(I/O)通道控制器结构，可以用于通用的个人计算机，计算机工作站，嵌入式通信和网络数据路径选择以及转换等功能。

与个人计算机和计算机工作站的应用有关的典型的多媒体，网络，以及通信技术的持续发展主要是取决于对高数据通过量的数字数据流执行集中的计算操作。通常需要的集中计算操作包括三维(3D)音频产生及其控制，基于音频合成的波形表，产生和检测V.34协议的串行数据信号，以及模拟语音的滤波和压缩。如果用软件来完成这些操作，个人计算机或是工作站系统的主处理机需要执行的每秒百万条指令循环(MIPS)通常是在20到40MIPS之间。另外，为了支持传送给普通的外部设备编码器/解码器芯片(CODEC)的有关的高通过量数据流，还需要为每个数字数据传送流额外耗费一两个  
10  
15 MIPS。

然而，个人计算机和工作站系统中使用的普通的微处理器一般仅仅能够支持60 MIPS，典型的情况是20到30 MIPS。上下文的切换，存储器管理，以及外围设备的等待状态进一步限制了用于计算处理以及传送数字数据流的有效MIPS。在实际情况下，由于在执行多媒体功能的同时必须要保留用来执行典型的终端用户应用的有效的主处理机MIPS，为了处理数字数据流，必须要提供众所公知的外围设备硬件支持。  
20

除了用于多媒体，网络，以及通信应用的外围设备硬件支持之外，编址还需要考虑硬件和软件接口的要求。除了计算功能之外，所使用的外围设备硬件必须是高效的，而不会对主处理机带来硬件服务要求的负担。任何额外增加的支持都会直接危及到通过增加外围设备硬件而获得的有效MIPS的净增加。  
25

同样，外围设备的软件接口也需要与主处理机所执行的操作系统实现有效的接口，这样才能有效地支持外围设备硬件的使用。无效的软件接口在主处理机与外围设备硬件进行通信时会导致MIPS投入的增加。另外，由于与外围设备硬件的软件接口的管理工作的复杂性，所增加的任何处理的负担都会  
30

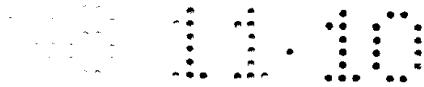


直接造成有效支持的MIPS的减少它是通过使用外围设备硬件而获得的。

5 为了用外围设备来支持多媒体，网络，以及通信应用，惯用的方案中包括提供各种专用的集成电路(芯片)组合，它基本上是用硬件实现的，或者是仅有局部构形的可编程计算功能和高级的软件可编程数字信号处理器(DSP)。廉价的密集封装式专用功能芯片通常只能执行有限的信号处理功能或是少量的有关功能。支持的功能通常限于专用功能的范围，而用于初始构形选择和节制的动态控制编程功能即使是有也是非常有限的。

10 作为硬件的外围设备，专用功能芯片通常很少或是不能直接支持提供对数字数据流的连续的实时信号处理的管理，而是仅仅支持多数据流的传送。这种芯片通常仅仅被作为数字数据流的用户或制造者(源或是接收器)，将数据流按照由主处理机确定的最高的或是某一指定的有效速率从芯片中取出，或是引入芯片。因此，采用专用功能芯片的外围设备硬件经常会出现数据流传送中断的问题，并且传送速度会受到限制，例如是由于过多的主处理机中断，上下文的切换，以及各种存储器管理的核心程序处理引起，还有主处理机CPU循环与主处理机同时执行的其他应用之间的主要的竞争。对于多逻辑并行的直接存储器存取(DMA)数据传送来说，系统广泛的竞争或是有限的系统硬件支持也会降低对专用功能芯片的数据传送速率的支持。另外，专用功能芯片中典型的单一数据流性质直接要求主处理机实际介入数据流的  
15 初始化工作，传送控制，以及任何必要的数据流混合或是多路复用工作。因此，尽管专用功能芯片可以明显地提高个人计算机或是工作站系统对多媒体和数字信号的处理能力，仍然给主处理机留下了一个明显并且基本上不受限制的处理负担。

25 在外围设备硬件系统中，特别是在需要执行复杂和高速信号处理计算的场合，可以使用通用的数字信号处理器。普通的DSP芯片能够在完全适合数据流处理的计算结构中提高高达50的MIPS。一般来说，这种DSP芯片比较昂贵，并且需要比较强的软件开发程序来完成为执行其所需功能而需要的软件算法。然而，DSP芯片结构通常不能对扩展的数据传送操作或是存储器管理功能提供优化的控制。但是，这种结构可以对内部专用存储器和扩展的本地  
30 连接存储器或是直接连接的专用功能外围设备芯片完成优化的数据读出，处理和写入。因此，实际上还是需要让主处理机介入对作为外围设备硬件执行



的DSP存储器空间的数据传送工作。不幸的是，这样做通常会使DSP在获得和支持主处理机性能的问题上象专用功能芯片的情况下一样受到同样限制。

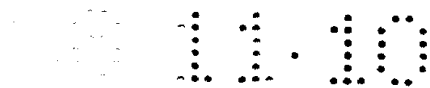
5 为了限制对多媒体外围设备硬件的数据中的传送中断，以便提高从主处理机获得的可以得到支持的数据传送速率，可以用诸如SPOX的一个小的优先实时核心程序来扩大由主处理机执行的普通操作系统。这种核心程序可以用一个能够支持多媒体外围设备硬件的实时中断和数据传送需求的低级设备驱动器来实现。尽管这样一种实时核心程序能够保证为DSP存储空间提供最大的数据传送支持的执行，主处理机同样要实际招致数据流传送工作的附加管理工作，并且要对实时核心程序本身执行额外的附加管理。

10 因此，目前个人计算机和工作站系统实现的多媒体，网络，以及通信外围设备硬件的子系统不能很好地满足通过附加的外围设备硬件来有效地提供额外处理能力这样要求的编址。

15 目前已经出现了诸如本机信号处理(NSP)和Direct-X等等各种基于主机的信号处理结构，并且广泛地规定了对几种目标，多媒体网络以及相关通信操作的操作系统所要求的良好结构软件接口的编址。基于主机的信号处理结构依赖于专门用主处理机本身来执行的至少是高级的信号处理功能。这种结构至少具有两个直接的优点。第一个优点是，如果在最终将数据流传送给外围设备硬件之前对数据流进行重要的处理，这种数据流是混合的多路复用数据流，或者是可以减少计算量，以便减轻在传送给外围设备硬件的数据中包含的附加处理。因此，个人计算机或是工作站系统的有效处理性能可以得到明显的改进。

20 第二个优点是操作系统提供了一种可能的综合性应用程序接口(API)，这样就趋于具体实现实际和功能性的外围设备硬件。因此，多媒体，网络，以及相关的通信应用就可以承担有效的扩展，或者是比任何特定的外围设备硬件的实际提供的必要功能完全不同的支持。对特定功能的直接支持并不是由特定例的外围设备硬件直接提供的，这种功能是由主处理机用软件来执行的，它是由基于主机的信号处理器来执行的，目前可以达到由具体例的外围设备硬件来支持的功能性等级。

25 尽管基于主机的信号处理可以提高个人计算机或是工作站系统在执行多媒体，网络，以及通信功能等方面的效率，由基于主机的信号处理器支持的许多功能仍然需要很大的计算量。因此，基于主机的信号处理为主处理机



带来了许多直接的负担。另外，尽管基于主机的信号处理可以明显地减少来自或到外围设备硬件传送的数据量，主处理机上承担的实际计算事务并没有受到明显的影响，总而言之还是保留了很多。

5 本发明的总的目的是提供一种外围设备I/O控制器，用于在建立在中央处理器和多媒体，网络，以及相关外围设备的通信之间的一条高效全带宽数据传送路径上支持多路并行的可变带宽数据流。

10 本发明是这样实现的，一种总线传送控制系统通过一个缓冲池管理着多个异步数据流的传送。这一总线传送控制系统包括一个具有许多存储块的缓冲池，其中的每个存储块可以提供存储多个数据字节，并且把多个数据传送设备连接到缓冲池上，通过缓冲池来传送需要在多个数据传送设备之间传送的一或多个数据流段。用一个传送控制器来维持关于存储块中的数据状态的状态信息，它包括用来反复计算这种状态信息的控制逻辑，并且可以优先选择一个第一数据传送设备和一个预定的存储块。

15 采用本发明的多路并行数字数据流通道控制器的计算机系统可以并行支持来自或到一个辅助数据处理装置的多个I/O数据流的实时传送。这种计算机系统可以包括第一处理单元，其中的第一存储器提供存储以各自的数据传送速率为特征的多个数据流，第二处理单元，用于控制多个数据流的内部数据段的数据，以及连接在第一和第二处理单元之间的一个通道控制器，用来在第一和第二处理器之间传送多个数据流。这种通道控制器可以根据各个  
20 数据流的数据传送速率来选择传送多个数据流的数据段。

因此，本发明的优点在于可以为主处理机系统动态地分配多通道控制接口，这样就允许对通道控制器的操作提供灵活的程序和控制。

25 本发明的另一个优点是可以尽量减少主处理机在执行基本的自主数据传送功能和动态流程管理时的负担。通道控制器可以支持内部中断管理，这样就减少了主机和DSP的中断支持负担，从而允许并且支持通过一个连接主机和辅助信号处理装置的由综合中断源管理的数据流通道的实时多路并行通道信号处理。

30 本发明的再一个优点是具有动态的缓冲器容量，对流过通道控制器的多数据流进行速率控制，优化地利用最大的有效外围设备I/O通道数据带宽。数据流的传送与数据流的类型无关。通道控制器的每个数据流通道都是逻辑



上独立的通道。可以对这些通道进行串行组合，通过附加或是本机的信号处理电路以及外部的外围设备信号处理装置来提供离散的数据路由路径。

5 本发明的又一个优点是具备自主的总线主控操作系统，不需要惯用的直接存储器存取操作并且不需要主处理机介入通过I/O通道控制器的数据传送。数据流段的交错传送可以有效地同时通过传送控制器并行地传送数字数据流。这样，在为多路并行的数据流提供实时传送支持的同时，就可以明显地减少主机应用的上下文切换和重要的主机用户中断核心程序模式切换的必要性。

10 本发明的其它优点是可以动态地支持可变速率的实时信号处理，在数据源和接收器之间按照流向的外围设备为每个数据传送通道建立的动态数据速率来保证有效的连续数据流。

本发明另外的优点是便于扩展对辅助信号处理装置的支持，这其中包括完全编程的装置以及其它仅有部分编程甚至完全用硬件实现功能的装置。

15 本发明再有的优点就是通道控制器作为一种构成的外围设备功能是可以编程的，其独立性可以大大地扩展，并且具有配合加强数据的外围设备子系统进行信号处理的功能，这样就能有效地降低计算机系统的成本。

20 根据以下结合附图对本发明作出的详细说明可以进一步认识到本发明的上述和其他优点和特征，在所有附图中用相同的符号来表示相同的部分，在附图中：

图1a是个人计算机系统的一个框图，实现本发明构成的I/O通道控制器子系统的第二实施例；

图1b是用本发明的第二实施例构成的一个I/O通道控制器子系统的框图；

25 图2是在本发明的另一个最佳实施例构成的一个系统的基础上的一个专用功能嵌入I/O通道的框图；

图3是一个软件结构的示意图，表示I/O通道控制器子系统和操作系统以及多个外围设备硬件接口之间的关系；

30 图4是一个示意图，表示在操作系统的基础上按照本发明构成的一个I/O通道控制器所支持的高级控制和实际的数据流动情况；

图5a是本发明一个最佳实施例的I/O通道控制器的详细框图；

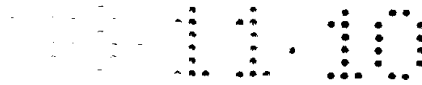


图5b是本发明一个最佳实施例的总线传送控制系统的详细框图;

图5c是本发明一个CODEC实施例的集成化外围设备接口的详细框图;

图5d是本发明一个最佳实施例的中断控制器集成化外围设备的详细框图;

5 图6是按照本发明的一个最佳实施例构成的I/O通道控制器中的一个FIFO库子系统的详细框图;

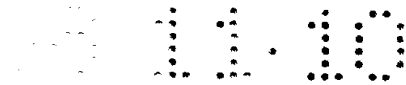
图7是本发明一个最佳实施例的I/O通道控制器中的总线传送单元控制器子系统的详细框图; 以及

10 图8a-c是一些控制流程图, 表示建立在本发明实施例构成的系统基础上的一个I/O通道控制器的最佳操作方式。

### I. I/O控制器系统结构

图1a提供了一个最好是用个人计算机结构实现的计算机系统10的框图, 它被连接到一个多功能I/O外围设备控制器硬件系统(I/O通道控制器)22。由典型的一个Intel<sup>®</sup> Pentium<sup>®</sup> CPU, Motorola PowerPC<sup>®</sup> CPU等等构成的主处理机12通过一条处理机总线16被连接到系统的主存储器14。用一个普通的外围部件互连(PCI)桥接口18提供高性能的数据, 并且控制处理机总线16和一个中层PCI总线20之间的连接。多功能外围设备系统22通过一个普通的PCI适配总线连接器24被连接到PCI总线20。I/O通道控制器核26集中  
15 响应管理的数据, 并且将相对于PCI总线20传送的控制信号连接到适配总线24。I/O通道控制器核26基本上作为一个独立的总线主外围设备控制器来工作, 用于支持在PCI总线20和系统22之间传送的多路并行的数据和控制信号。这些并行的传送最好是按照独立的数据流对来管理, 并且执行控制线索其中每个数据流是由通过I/O通道控制器26的离散的数据块传送构成的, 它包括  
20 了一或多个字节的数据。同时受到支持的数据流的数据块彼此间交错, 从而有效地实现并行的数据流数据传送。

按照本发明, 数据块的交错是由控制器子系统22来确定的, 并且一般与主处理机12的执行无关。由I/O通道控制器核26实现的交错数据块反映了各个数据流各自所要求的传送速率, 它是由被传送数据流数据的性质来决定  
30 的。因此, 其中数据流数据代表的是恒定数据频率音频信号流, 尽管在总体上维持一个数据音频数据流编码器/译码器(CODEC)独立要求的反映数据流



传送速率的固定吞吐量速率，音频数据流中的交错数据块也可能是变化的。其中要求的数据速率不是恒定的，例如在采用可变速率的音频或视频压缩时，尽管还按照要求的数据传送速率适合压缩的数据流，数据块的交错仍会进一步变化。

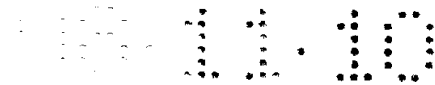
5           在本发明的一个最佳实施例中，I/O通道控制器核心26支持多个直接连接的串行总线外围设备28。这些串行总线外围设备28最好包括用来实现专用的高速，典型的串行定向数据传送功能的专用功能芯片。典型的串行总线外围设备28包括一个专用功能芯片，用来提供高速串行通用同步/异步速率传送(USART)功能。其他可能的串行总线外围设备28可以实现一组专用功能芯片，用来提供基于音频子系统的廉价硬件。

10           I/O通道控制器核26最好还可以提供一个高速的通用数据和控制总线30，用来支持宽范围的数据处理和外围设备传送。在总线30上可以连接一或多个数字信号处理器(DSP)32，36及其相关的DSP RAM 34，为间接连接的外围设备提供高性能的计算支持，例如用于串行总线外围设备28，并行总线外  
15 围设备38，以及主处理机12。尽管这取决于特定实现的DSP 32，36，DSP RAM 34最好是存在于DSP 32，36内部，并且由总线30变换为可以存取的公共地址空间。DSP 32，36最好还为直接连接的外围设备提供接口控制支持，例如那些串行总线外围设备40，42。

20           DSP 32，36通常可以独立并且彼此并行地执行各种不同的计算和控制功能。例如，数字信号处理器32可以执行一范围的信号处理功能，适当地支持各种电信专用串行总线外围设备40。这种数字信号处理器32可以执行信号处理算法，也就是执行必要的协议和信号处理，用来实现高速传真调制解调器，综合业务数字网(ISDN)连接，以及无线数据收发信机系统。直接连接到  
25 数字信号处理器32的串行总线外围设备40还可以适当包括普通老式电话系统(POTS)接口，ISDN接口，以及符合低功率FCC的RF或是IR数据收发信机。

          第二数字信号处理器36可以独立或是协同地提供音频和视频信号处理功能，包括3D音频合成，基于音频信号产生和混频的波形表，语音和音乐再现，以及语音识别。直接连接到数字信号处理器36的串行总线外围设备42则可以包括适当的CODEC和滤波器，用于执行高速双向的数-模信号转换。

30           并行总线外围设备38最好是直接连接到总线30，用来支持高速的并行功能。受到支持的并行功能可以包括普通的高速双向并行数据传送高速以太网



型网络接口以及用于普通光盘驱动器，CD-ROM驱动器等等的普通EIDE和SCSI接口。并行总线外围设备38还可以包括通过视频控制器对视频显示控制器的帧缓冲器直接或间接的接口。并行总线外围设备的一个附加功能是通过直接连接到CODEC的一或多个串行控制线41提供简单的并行/串行转换功能，而这种CODEC是被作为直接连接的串行总线外围设备42另外来支持的。在实际情况下，串行连接41可以用另一个DSP 32或更象是主处理机12经通过总线30的连接而比较直接地控制CODEC的存取。

5 采用这种I/O通道控制器26，可以尽量减少在各种串行和并行总线外围设备28，38，40，42和数字信号处理器32，36当中进行数据传送时的争论。由多功能外围设备系统22来支持的最大数据通过量主要会受到PCI总线20和系统存储器14之间的可用数据传输带宽的有效限制。

对外围设备系统22所支持的数据通过量的更明确地实际限制是数字信号处理器32，36的有用计算性能。然而，这种限制可以用多种不同方式来克服，这其中包括增加连接到总线30上的数字信号处理器的数量，并且在并行的处理器当中分配计算负荷。另外，利用高性能的数字信号处理器也可以提供更大的计算吞吐量。

15 以下参见图1b，在图中表示了另一种外围设备系统44。I/O通道控制器核46从流通在串行总线外围设备28和可能的并行总线外围设备30的数据流的可能的时间临界数据传送路径中进一步去掉了数字信号处理器32提供对数字信号处理器32增强的支持。因此，当数字信号处理器32可以处理通过串行总线外围设备40直接连接的数据流时，利用I/O通道控制器核46本身来控制串行和并行总线外围设备28，38与DSP RAM 34之间的所有数据传送，就可以省掉数字信号处理器32的许多即使不是大部分的存储器管理和重复的数据传送功能。这样，DSP 32就可以用更多的时间来执行计算工作，而不用执行那些并不是非常适合数字信号处理器32的扩展的数据传送工作。因此，数字信号处理器32几乎是用简单的存储器负荷以及存储工作对RAM 34和数字信号处理器32之间移动的数据进行专用操作。由数字信号处理器32处理后存储在RAM 34中的传送数据然后被传送给串行和并行总线外围设备28，38，不需要数字信号处理器32实际进行任何有效附加的处理。

25 30 在图2中表示了本发明的一种嵌入的应用。系统10'可以作为一个智能上独立的或是遥控管理的数据流处理器来工作。例如Intel 80196那样的一个

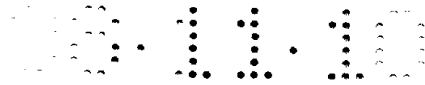
嵌入式处理器12' 和一个由DRAM和闪速EPROM构成的主存储器14' 通过处理器总线16' 来完成内部连接。用I/O通道控制器核26' 来支持处理器总线16' 与I/O总线30' 之间的一或多个可能会重叠的数据流。在I/O总线30' 上最好连接一或多个DSP 32' 和任何数量的网络外围设备, 例如网络控制器38', 38"。

5 DSP 32' 可以直接用于网络控制器38', 38"产生的中断, 这是在支持其各自的网络数据流传送功能时。DSP可以利用I/O通道控制器来实际控制数据在I/O总线30' 上的传送, 并且在必要时可以通过主存储器14'。

10 在一个最佳嵌入式的实施例中, 嵌入式系统10' 用作一个高性能的智能网络数据路由器。嵌入式的CPU 12' 执行主存储器14' 中的闪速EPROM所提供的控制程序, 用来支持普通的SNMP路由器管理功能, 以及数据采集和诊断功能。DSP 32' 执行内部DSP 存储器提供的控制程序, 这种程序可以用嵌入式CPU 12' 从主存储器14' 卸载, 也可以由DSP 32' 内部或是相关的闪速EPROM来提供。DSP控制程序可以用于产生和管理路由数据, 并且对DSP存储器内部的树进行滤波。当网络控制器38' 接收到数据包时, 源和目的数据包被传送给  
15 DSP32' 并且相对于该路由和滤波器树进行处理, 从中确定是否以及如何传送这一数据包。DSP 32' 和嵌入式CPU 12' 都可以被用来执行协议转换, 并且根据需要来实现保密协议。

## II. 软件系统结构

20 在图3中用示意图50表示了I/O通道控制器依据与计算机系统10的主要部件相关数据流的操作方式。输出的数据流是由主处理机12结合着一个操作系统来执行的应用程序而产生的。在应用层52内部执行的应用形成将数据56传送给存储器14内的主存储器存储空间54。在按照通过基于主机的信号处理器来操作的主处理机12所执行的应用的方向, 可以进一步控制这种数据。这样, 数据流60就能在存储器空间54和主机信号处理层58之间反复地传送。由  
25 主处理机12实际执行的具体的信号处理工作主要是由各种设备驱动器来确定的具体用来一起实现基于主机的信号处理器API (HBSP API) 的全部或是有效功能。HBSP API的调用最终会直接或是间接地解决调用一个I/O通道控制器设备驱动程序, 用来起动一个数据传送控制线索, 以从主存储器空间54向I/O通道控制器核62传送数据流64, 用于下一步的传送以及可能的处理。在  
30 I/O通道控制器设备驱动程序的执行过程中, 一旦由主处理机12在规定数据流传送64的主存储器空间54中产生了一个控制线索, 主处理机12的操作就简



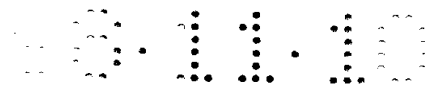
单允许按照I/O通道控制器62的要求来处理这种控制线索。然后由I/O通道控制器核62作出响应，实际执行数据流的传送。

5 主处理机由构造一或多个连接的总线传送单元(BTU)控制块在主存储器空间54内建立一个控制线索，使主存储器空间54中提供的流数据与控制线索相联系。在执行I/O通道控制器设备驱动程序时，主处理机12用初始BTU控制块来初始化I/O通道控制器62，并且编程一定构造的I/O通道控制器62的寄存器，以便为通过I/O通道控制器62的数据流构成选择的一个特定的路由。然后，主处理机12实际上并不参与向I/O通道控制器核62传送数据流64的工作。此后的数据流传送是在I/O通道控制器核62的直接控制下进行的，它是由主处理机12仲裁的一个可以对主存储器空间54进行存取的总线主外围设备。接着，根据各个控制线索的限定，有效地进行彼此独立地起动并且连续地向I/O通道控制器62传送多个并行数据流，和可以异步地起动，并且由I/O通道控制器核62来完成。

15 每个总线传送单元至少隐式地规定了一个有关数据流的传送目的。数据流可以传送给数字信号处理器68的RAM存储空间，或者是传送给各种直接连接的串行和并行总线外围设备28，38的一个特定的外部硬件接口74。随着数字信号处理器68对一个数据流70进行的处理，产生的数据72可以被传送给串行总线外围设备40，42的一个直接连接的外部硬件接口74。或者是可以把结果数据72返回存储在DSP RAM 空间内。数字信号处理器可以从DSP RAM中继续把数据流72再传送给任何一个直接连接的串行总线外围设备40，42。

20 用来对存储在DSP RAM中的数据流69，80的传送控制线索进行限定的总线传送单元控制块可以被设在系统主存储器14或是DSP存储器34中。最好是由主处理机12在系统主存储器14中建立DSP控制线索总线传送单元控制块。DSP RAM34存储空间的管理是由主处理机12来全部执行的，相关于使用DSP RAM 34由此也允许DSP 32，36彼此独立地操作。

25 DSP总线传送单元控制块仅限于控制一或多个数据流的传送，通过和在I/O通道控制器核62的控制下传送给直接连接到I/O通道控制器62上的任何一个串行和并行总线外围设备28，38的外部硬件接口74。或者是I/O通道控制器核62可以将处理的数据流84送回主存储器空间54，以便由基于主机的信号处理器58进行可能的进一步的处理，或是供应用52使用。



数字信号处理器68和I/O通道控制器核62从其直接连接的外部硬件接口74上接收输入的数据流76, 82。此处数据流76是被数字信号处理器68直接接收的, 最好这一数据流在经过处理后就被存入DSP RAM 34。接着, 用系统主存储器14内部提供的BTU控制块对这种存储的数据传送进行协调。用这种DSP

5 BTU控制线索来指导对I/O控制器62的数据流69的传送, 将其作为数据流84传送给主存储器空间54, 或是作为数据流80改方向传送给外部硬件接口74。

来自直接连接到I/O通道控制器核62的外部硬件接口74的输入数据流82作为数据流84被改方向传送到主存储器空间54。或者是可以将数据流82作为数据流70传送给DSP RAM 34。在后一种情况下, 数据流典型的是由数字信号

10 处理器68进行处理的, 并且作为数据流72直接提供给外部硬件接口74, 或者是作为数据流69提供给I/O通道控制器62, 然后再作为数据流80回到外部硬件接口74, 或者是作为数据流84提供给主存储器空间54。

总而言之, 输入数据流最终按规定送回主存储器空间54。一旦它出现在主存储器空间54中, 基于主机的信号处理器58就可以对数据流60进行进一步

15 的处理。最后, 操作系统可以将代表着完全经过处理后的输入数据流的一个数据流86从主存储器空间54返回到应用程序52。

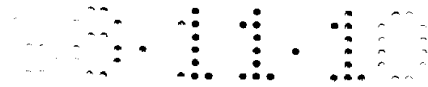
这样, 本发明的数据流管理系统50就可以提供多个数据路径, 它们是可以灵活限定的, 并且可以交互式地交叉, 从而通过基于主机的信号处理器

20 58, I/O通道控制器62, 数字信号处理器68, 以及通过外部硬件接口74连接的外部硬件之间的有序组合或是子组合(sub-combination)而有效地获得对多个有效的并行数据流的最大处理能力。

在图4中表示了另一种系统50, 它表示在本发明的计算机系统10内部建立的控制和数据流动路径。一个应用程序层90包括任意数量的共同执行的应用92, 94。这些应用92, 94使应用程序接口(API)调入操作系统层96, 该层

25 中包括基本操作系统98以及通常包括MIDI 100, TAPI 102和语音API 104操作系统扩展部分的操作扩展部分。在与如用于Ring 3操作系统扩展部分的Direct-X API说明书一致的操作系统层96内部可以产生任何数量的附加或是不同的API。在图中还有一个存储块106, 它在逻辑上与操作系统98一同驻留在惯用的操作系统层中。

30 设备驱动程序接口层108包括必要的各种Ring 0设备驱动程序, 用来支持操作系统层96和构成外围设备层114一部分的任何部件之间的通信和数据



5 传送。具体地说，设备驱动程序接口层108包括一个中断处理程序110和一个设备驱动程序112，由主处理机12来执行这些程序，用来提供支持建立在外围设备控制层114内部的I/O通道控制器116。

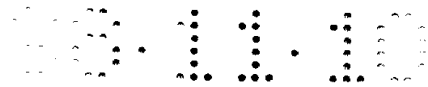
最后，按照本发明还提供了一个控制器外围设备层118。这一控制器外围设备层118通常包括一个数字信号处理器120，一个可以供DSP 120存取的存储器122，以及DSP 120和I/O通道控制器116都潜在地可以存取的专用功能外部硬件124。

10 在典型的控制流程中，由一个应用 A 92通过与操作系统层96进行惯用的相互作用的联系而在存储器106内部产生一个数据流。由于至少是数据流的起始部分是在存储器106中构成的，操作系统层96就会调用设备驱动程序112，以准备通过I/O通道控制器116向存储器122或是控制器外围设备层118的外部硬件124传送这一数据流。设备驱动程序112在存储器106内部产生一或多个BTU控制块，用来限定为执行数据流传送所必须的控制线索。设备驱动程序112对初始BTU的控制线索进行编程，进入I/O通道控制器116中并且提供  
15 一个允许根据编程的BTU来操作I/O通道控制器的信号。然后，I/O通道控制器116的操作基本上是自主地完成I/O通道控制器116内的BTU编程所限定的数据流的传送，和存储器106内的其他BTU限定数据流的传送，它也是控制线索的一部分。

其中数据流传送的目标是存储器122，设备驱动程序112就会首先在存储器106内产生另一系列的一个或多个BTU控制块，用来建立适合管理DSP一侧  
20 传送的数据流的控制线索。初始的BTU是由设备驱动程序112编程进入I/O通道控制器116内的。接着，设备驱动程序112可以向数字信号处理器120指派一个指令，规定要求的DSP的功能。最好是通过使用编程的I/O (PIO)把指令指派给I/O通道控制器核116保留的一或多个寄存器，并且可以由数字信号处理器120响应一个表示指令信息已经指派的DSP中断来读出。根据这一指派的指令，DSP 120可以根据需要来初始化以执行该指令的功能，然后向I/O通道控制器116提供一个信号136，按照预先编程的BTU来起动控制器的操作。

30 根据得到的两个至少是隐含地联系在一起允许的BTU，I/O通道控制器116起动自主的数据传送功能，直接通过126从存储器106检索由主机一侧的BTU限定的一部分数据流，将数据暂时性地存储在一个内部FIFO库(pool)中，并且自主地将数据传送给存储器122中由DSP一侧的BTU所限定的一系列



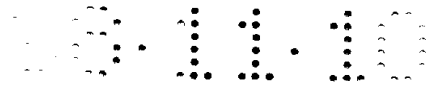


位置中。I/O通道控制器在访问存储器106时的自主操作最好是作为一种总线主控的设备，它直接参与由主处理机12和PCI桥接口18的硬件总线管理控制器来支持的存储器循环仲裁。因此，在实际执行数据传送126时基本上不需要使用主处理机12的执行循环。

5        为了在存储器106中有效地形成数据流，需要主处理机12的执行循环，并且典型地执行设备驱动程序112，以便初步建立和管理一个或多个主机一侧和DSP一侧的一系列BTU，这些BTU限定了数据流传送的控制线索。然而，主处理机12在产生和管理BTU时所需的执行循环比在应用92，94的控制下经过初步处理后再把数据流存储在存储器106中所需的执行循环相对较小。另  
10  外，I/O通道控制器116的操作可以使存储器106中形成的数据流松散地和I/O通道控制器116控制下的实际传送的数据126同步。因此，主处理机12在产生和管理主机一侧的BTU时对执行设备驱动程序112的限制并不需要很严格地与通过I/O通道控制器116的数据传送所要求的实时时间相联系。

      自主的按需要驱动I/O通道控制器核116本身的操作和FIFO库缓冲功能  
15  进一步助长了主处理机12在执行对I/O通道控制器116的支持时采用的这种松散的同步。I/O通道控制器核116可以自主地从存储器106和存储器122中取出连续的BTU，尽管在传送数据流时仅仅是从存储器106中取出。因此，在一个BTU控制的数据流的一部分完成时，I/O通道控制器核116可以按照控制线索从存储器中取下一个BTU，并且用新取的BTU代替已经完成的BTU。这样，  
20  只要是在控制线索中为提供数据流的持续传送限定了连续的源和目标控制BTU，就可以自主地向存储器122传送输出数据流。DSO一侧BTU的速率被完成，被再装入和被再启动可以有效地反映出数字信号处理器120在分别执行每项指令功能时所要求的数据吞吐量速率。

      如果数据流目标是外部的硬件124，就可以产生一种简化的操作方式。  
25  I/O通道控制器116为通过数据路径130传送给外部硬件124的数据提供有限但是适当的缓冲。当外部硬件124需要来自I/O通道控制器116的数据时，就采用当前的BTU来控制从存储器106传送给I/O通道控制器核116的数据流的传送。当外部硬件124引起了被I/O通道控制器116缓冲的有效数据时，控制器116就自主地操作，从当前BTU所限定的存储器106中的位置上再次装填缓冲器。  
30  在当前BTU涉及的数据被外部硬件124用掉之后，I/O通道控制器116就



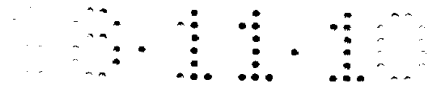
从存储器106中自主地装载这一控制线索中的下一个顺序的BTU，以便维持数据流126的传送。

I/O通道控制器核116按照大体上类似的方式来处理输入数据流。DSP120从直接连接的外部硬件接收到的数据流可以由DSP120进行处理，并且存储在存储器122中。主处理机12对接收的每个输入数据流进行预先处理，并且利用I/O通道控制器核116中编程的初始BTU在存储器106中形成一或多个DSP侧BTU的对应的控制线索，必须依次获得DSP 120的允许。在存储器106中再次用I/O通道控制器核116中编程的初始BTU来产生一或多个主机侧BTU的控制线索。尽管是空的，初始的主机侧BTU最好还是被置于允许状态。

然后将用来指定DSP 120所执行的功能的一个指令信息提供给I/O通道控制器核116内部的一个信息存储寄存器。DSP 120应答这种信息，初始化所要求的功能，接收从外部提供的数据流的一部分，并且经过DSP 120处理之后存储在存储器122内的一个已知的地址块内。然后向I/O通道控制器核116提供一个允许信号136，起动初始的DSP侧BTU，并且允许自主传送接收的数据流。在完成了DSP侧BTU指定的数据流传送之后，向I/O通道控制器核116装载下一个DSP侧BTU，并且保持禁止状态。同时接收从外部提供的数据流的下一部分，并且经过处理后存入存储器122内由下一个DSP侧BTU涉及的另一个已知的地址块内。同样，当这一地址块被处理后的数据填充之后，就响应允许信号136来起动当前的DSP侧BTU。存储器122内的两个地址块在连续交替地处理一个数据流的过程中被连续交替地使用。

在当前的主机侧BTU在存储器106中指定的地址块被经过处理的输入数据流填充之后，这一BTU就结束了，I/O通道控制器核116向中断处理器110发出一个中断132，至少为主机可以进行处理的地址块中的信号调用设备驱动程序112。最好在允许状态下还应该用I/O通道控制器核116自主地装载用来指定存储器106中的另一个地址块的下一个BTU，这样才能连续地接收经过处理的输入数据流。

按照一种合理的假设，在I/O通道控制器核116和存储器106之间的数据传送要比DSP能够接收、处理、并且将数据存入存储器122的速率实际上快得多，在I/O通道控制器116内进行适当的缓冲至少可以保证数据流128产生按需要的驱动传送，不会出现数据溢出。



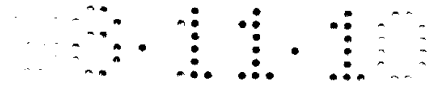
I/O通道控制器核116还可以为直接从外部硬件124接收的任何数据流提供数据缓冲。一般来说，外部硬件124可以响应主处理机执行的设备驱动程序来进行操作。I/O通道控制器核116最好为设备驱动程序112提供一个寄存器接口，允许编程的I/O通过外部硬件124的可编程寄存器。在外部硬件124的初始化过程中，在存储器106中产生一个BTU控制线索，在I/O通道控制器核116中 5 对一个初始BTU编程，并且用来控制接收的数据。I/O通道控制器核116按照外部要求的数据传送速率来直接缓冲一个输入数据流130。接着，I/O通道控制器116自主地操作，在当前BTU的控制下将接收的数据流传送给存储器106。按照另一个合理的假设，I/O通道控制器核116向存储器106传送数据的速率应该能够明显地大于数据流130的接收速率，这样也能有效地防止数据溢出。 10

### III. I/O通道控制器结构

在图5a中表示了按照本发明的最佳实施例构成的一个I/O通道控制器140的 15 总体结构。I/O通道控制器140最好采用三个主要的内部总线，包括总线主控程序总线142，FIFO库总线144，以及一个PIO总线146。总线主控程序总线142在任意数量的总线接口模块148，150和总线传送控制系统170之间提供一条通信通道。FIFO库总线144是一条专用的总线，主要用于总线接口模块148，150，FIFO库172，以及任何数量的集成化外围设备和外围设备接口 20 174-188之间的高速脉冲串数据传送。PIO总线146用于在总线接口模块148，150，总线传送控制系统170，以及集成化的外围设备和外围设备接口174-188之间提供一条通用的可编程I/O通信路径。PIO总线存取控制器151在各个总线接口模块148，150之间操作作为一个PIO总线存取判优器。

各个总线接口模块148，150用于将I/O通道控制器核140连接到各自的外部通信总线。主机接口模块148用来将I/O通道控制器140连接到外部PCI总线 25 20，而DSP总线接口模块150同样可以为连接到外部数字信号处理器32和DSP RAM 34的外围设备总线30提供一个接口。

主机接口模块148连接到总线20的地址，数据和控制(A/D/C)线154上，并且为I/O通道控制器核140内部的中断控制器所驱动的中断线156提供一个 30 外部接口。在主机接口模块148内部，地址，数据和控制线154连接到一个主机总线受控单元158和一个主机总线主控单元160。主机总线受控单元158为



PI0总线146提供一种基于惯用的寄存器式可编程I/O接口，以便使PI0总线146可以访问主处理机12。这种寄存器接口还允许通过PI0总线146对内部连接到PI0总线146的各种单元的可编程构造和数据寄存器进行程序存取。主机总线受控单元158同时提供了一种可以从外部存取的寄存器接口，允许对主机接口模块148进行程序控制。主机总线主控单元160本身最好是包括适合支持PCI总线20上的普通总线主控存取仲裁的控制逻辑。

DSP总线接口模块150同样连接到DSP总线30的地址，数据和控制线上。地址，数据和控制线162连接到DSP总线受控单元166和DSP总线主控单元168。用I/O通道控制器核140内部的中断控制器驱动的一条中断线164的外部接口也是由DSP总线接口模块150来支持的。DSP总线受控单元166提供一个可编程寄存器接口，它可以用于DSP总线主控单元168的程序控制，并且支持总线30和PI0总线146之间的通用程序I/O存取。DSP总线主控单元168还包括适用于允许DSP总线30上的普通总线主控存取仲裁的控制逻辑。

主机和DSP接口模块148，150对PI0总线146的配合存取是通过PI0控制器151来控制的。必须对唯一对PI0总线进行存取的每个总线受控单元158，166进行仲裁，并且通过PI0总线146通向任一个可以存取的接口寄存器。另外，主机总线受控单元158可以请求PI0控制器151产生一个PI0总线旁路。从主处理机12通过主机总线受控单元158，PI0总线146和DSP总线受控单元166产生一条旁路数据路径。这样就能为主处理机12提供一条数据路径，允许主处理机12通过DSP总线30来控制所有可以存取的外围设备。这样做的好处是可以简化某些并行总线外围设备38的操作控制，例如直接设定或是读出一个CODEC 42的控制模式。

总线传送控制系统170作为一个中央控制系统来工作，用于管理在主机和DSP总线主控单元160168，FIFO库172，以及连接到FIFO库总线144上的集成化外围设备及外围设备接口174-183之间传送的脉冲串数据的传送。在总线传送控制系统170中保存一个可编程的BTU表。在本发明的一个最佳实施例中，这种BTU表可以同时存储48个BTU，每个BTU占用16个连续存储的字节。表I为最佳的BTU结构提供了详细的说明。

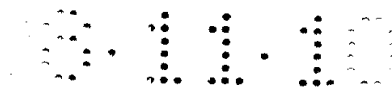


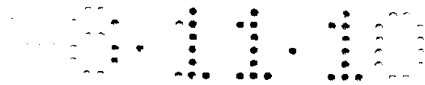
表 I

	字段	字节	位
5	地址	4	[31: 0]
	传送长度	2	[15: 0]
	连接描述符	2	[15: 0]
	控制字	2	[15: 0]
	状态字	2	[15: 0]
10	保留	4	[31: 0]

BTU结构的地址字段限定了主机和DSP总线主单元160, 168使用的存储器地址, 用于在系统主存储器14和I/O通道控制器26之间以及I/O通道控制器  
 15 46和DSP RAM 34之间传送数据。在数据传送的过程中更新BTU的地址字段值, 用来保持作为准确的地址指针的地址值。相反, 传送长度值随着传送的进展而减少, 保持对BTU表示的剩余传送长度的精确反映。在本发明的最佳实施例中, 主机地址空间被处理为一个平面的32位地址字段。用于DSP总线主传送的地址字段可以采用不同的限定方式。如表II中所示, 用限定的高数位或是地址  
 20 字段的位就可以容易实施DSP RAM 34的分区对称, 用来有选择地把多个数字信号处理器32, 36或是可以提供或者利用DSP RAM34内的离散存储器的其他外围设备38的DSP RAM变换成对应I/O通道控制器核140的公共地址空间。

表 II

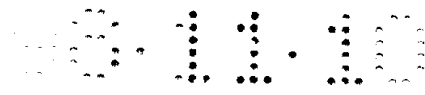
	地址(仅是 DSP)	说明	位
25	DSP	DSP#1 或 DSP#0	31
	DM	数据(1)或程序(0)存储器	30
	保留	留做以后使用	[29: 15]
30	地址	字节地址	[15: 0]



连接描述符字段允许在当前控制线索的下一个顺序BTU中存储一个阵列指针。最好是把没有驻留在I/O通道控制器26提供的BTU存储空间BTU中的一个控制线索的后续BTU存储在系统主存储器14内部的一个表中，尽管BTU也可以存储在DSP RAM 34中。此处把BTU存储在系统主存储器14中，在I/O通道控制器核140的构造寄存器中单独存储二字节的本机基本连接地址值。在系统主存储器14中构成BTU表基本地址这种本机基本连接地址与当前的BTU连接描述符值是衔接在一起的，可以为存储在系统主存储器14中的下一个连接的BTU提供存储器参考。与此相反，这种BTU连接描述符本身就足以为存储在DSP RAM 34中的连接的BTU提供存取时所需的全部地址说明。当I/O通道控制器核26内的当前BTU的传送长度被递减到零时，在自动装载下一个连接BTU时就采用这种连接描述符字段。是否自动采用连接描述符字段来装载下一个BTU是由存储在构造控制字寄存器内部的控制位的值确定的。在表III中限定了控制字寄存器的定义。

15 表 III

控制字	说明	位
总线主控制器	来自 DSP/主机存储器的总线主控制器	15
中断	在完成 BTU 时中断	[14: 13]
20	00 无中断	
	01 主机中断	
	10 DSPA 中断	
	11 DSPB 中断	
连接定义	连接信息有效/无效(即, 等待)	12
25 连接主控制器	连接到 BTU 连接表	[11: 10]
	00 无连接	
	01 从主机连接	
	10 从 DSPA 连接	
	11 从 DSPB 连接	
30 数据转换	在传送期间选择数据变换	[9: 8]
脉冲串长度	每次脉冲串 32 字节/不是 16 字节	7



方向	来自/不是到该 FIFO 的传送	6
保留	留做内部使用	5
FIFO 编号	从 FIFO 库中选择 FIFO	[4: 0]

5

在控制字字段内用总线主控器位来定义BTU是对应着主机侧还是DSP侧的控制线索。中断子字段定义了BTU传送长度递减到零时是否以及向谁发出唯一的中断。中断源矢量的编码和实际产生的外部中断信号是由I/O通道控制器核140内部的一个中断控制器188来控制的。连接定义位规定了连接描述符的字段是否被维持在有效值。用连接主控器子字段来存储BTU连接表在系统主存储器14中的基本地址位置。数据转换子字段对准备传送的数据进行数据变换编码，例如字节顺序颠倒或是字节/字转换。

10

脉冲串长度位在16和32字节之间选择脉冲串传送长度，而方向位规定了脉冲串传送的方向。FIFO编号子字段被用来识别在当前BTU的控制下被用作数据传送源或是目标的FIFO库172内部的一个具体FIFO的数字标识符。

15

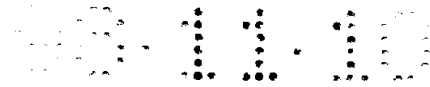
最后，在表IV中详述的一个BTU的状态字字段定义了一些单独的位，用来反映规定的BTU是否忙碌，例如在BTU更新操作期间是由I/O通道控制器核26自动执行的，以及BTU是否有效并且当前可使用，或是是否被允许参与总线主控器存取的仲裁。

20

表 IV

子字段	说明	位
忙碌位	BTU 正在忙碌传送/连接(只读)	1
BTU 允许	允许 BTU 对总线主控器判优	0

30



多级电路优先权判优器 (CPA) 最好是用总线传送控制系统170来实现的，用于连续地计算当前允许的BTU。判优器被连接到BTU表上，按一种方式可以直接识别出那些具有其BTU允许位已经被设定的BTU。CPA在识别的允许的BTU组内部进行操作，逐渐选择用于每个主机和DSP总线主控单元160，168的BTU，并且为其分级。

由CPA操作响应存取请求组而执行的核心算法存在于判优循环中进一步限定了识别出的允许的BTU组。所有可能的请求者都被从左到右的顺序排列成基本的相对优先权。本发明的CPA为被允许请求者的优先权提供了附加的定量。具体地说，直接优先权判优循环中获胜的请求者实际上在当前的循环中没有优先权。用优先权再一次向右递减时，对于优先获胜的请求者右侧的第一个允许的请求者(如果有)被推到当前判优循环中的最高优先权。如果在当前循环中的优先权获胜请求者右侧没有一个允许的请求者，就按照允许的请求者的优先权的从左到右的基本关系来选择一个当前的获胜请求者。

CPA的判定矩阵最好是用一种修改的反二叉树实施。用附加的控制逻辑来维持状态信息，用来识别直接优先仲裁循环中的获胜请求者。它还可以提供一种用来识别子树的状态信息，在子树中包括优先获胜的请求者，以及这一个优先获胜请求者在子树的最低级节点上是否处在相对于最低级二进制节点的左或右。接着在所有潜在请求者当中提供一种有效的机制，用来保证实质上可信的仲裁。

在图5b中详细地表示了总线传送控制系统170。总线传送控制系统170包括环形的优先权判优器190。通过控制线路194从直接连接到总线主控程序总线142和PIO总线146上的设备来接收对BTU表192要求存取的各个存取请求。总线主控程序总线142和PIO总线146的数据和BTU地址部分都被连接到BTU表192，以便读出和写入BTU。总线主控程序总线142同时被连接到一个BTU扫描器196，判优器190将其当作连续请求存取BTU表192来处理。BTU扫描器196在工作中连续地扫描存储在BTU表192中的允许的BTU，用于BTU标识符以及一个内部寄存器组198内部的主机和DSP分级寄存器的有关数据的分级。在表V中提供了最佳的BTU寄存器阵列定义。



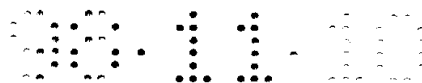


表 V

	子字段	说明	字节
5	BTU 控制	复位, 单步扫描, 允许, 并且有效去掉编程限制	2
	主机的基本连接值	主存储器中的主机 BTU 表偏移	2
	主机最佳 BTU	最佳 BTU 寄存器	2
	主机当前的 BTU	当前的 BTU 寄存器	2
10	主机扫描 BTU	扫描 BTU 寄存器	2
	DSP 最佳 BTU	最佳 BTU 寄存器	2
	DSP 当前的 BTU	当前的 BTU 寄存器	2
	DSP 扫描 BTU	扫描 BTU 寄存器	2

15

对 BTU 表 192 的每次存取是通过判优器 190 的操作而唯一地仲裁的。而总线主控单元 160, 168 另外在空闲的时候各自单独请求对 BTU 表 192 进行存取, 用来获得准备在总线主控单元 160, 168 中编程的 BTU 存储数据, 用于定义总线主控脉冲串数据传送。总线主控单元 160, 168 还在脉冲串数据传送之后请求对 BTU 表 192 进行存取, 以便更新当前有效的 BTU。

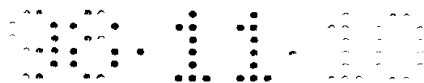
20

总线受控单元 158, 166 可以通过 PIO 总线 146 对 BTU 表 192 进行存取。对这种存取的支持可以典型的允许主处理机 12 和 DSP 32, 36 在 BTU 表 192 中对最初的 BTU 进行编程随后允许该 BTU。PIO 总线 142 还可以用于允许主处理机 12 根据需要来检查 BTU 的状态, 以便监视低级的数据传送进程, 并且分析可能出现的错误状态。

25

BTU 扫描器 196 基本可以使用所有其他的 BTU 表存取循环。由于仅仅考虑那些被允许的和绝对有效的 BTU, 扫描器 196 的扫描速度提高到最大, 同时可以把扫描器 196 的整体复杂性降低或减至最小。在对每个有效和允许的 BTU 进行扫描的同时, BTU 扫描器 196 还通过 FIFO 标记总线 171 获得支持的 FIFO 标记状态信息, 用于正在被扫描的 BTU 所参考的 FIFO。这样, BTU 扫描器 196 就能有效地和有能力和有能力用于对所有 FIFO 标记存储器进行访问。

30



最后，总线传送控制系统170还要在BTU表192中每次更新操作时按照程序的控制潜在产生的中断。一项特定的更新操作是否会导致发出一个中断是由更新操作来确定的，取决于这种更新操作是否代表着总线主控器传送错误，BTU传送完成，以及正在被更新的BTU的控制字内部的可编程中断指示位。如果需要产生一个中断，就通过控制线200向I/O通道控制器核26内部的中断控制器发出中断。

每个BTU存取判优循环可以被视为开始，此时主机或是DSP总线主控单元160，168完成了一次由BTU控制的总线主控脉冲串传送。相应的BTU被更新，并且选择下一个BTU作为产生的优先的判优循环，将其用作下一次总线主控脉冲串传送中为总线主控单元160，168编程的数据源。在完成之前不会对单个的BTU所规定的的数据传送进行处理，而是在脉冲串长度的边界上进行分段或是局部的处理，并且用其他BTU指定的数据传送进行交错。

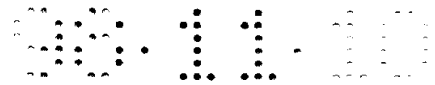
由于前分级的BTU仲裁，下一个BTU可以直接被用于对总线主控循环编程。因此，当下一个BTU被用于在主机或是DSP总线主控单元160，168降底分级编程时，环形优先权判优器就在当前的总线主控脉冲串传送完成之前操作来选择并且分级下一个新的BTU。接着，只要用I/O通道控制器26和系统主存储器14及RAM34之间可以支持的潜在的总线主控传送速度就能够限定I/O控制器可以支持的最大的潜在数据带宽。

在表VI中表示了主机和DSP总线主控单元160，168所采用的总线主控程序地址和控制接口。

表 VI

25	总线主控地址总线	说明	位
	地址	传送的起始地址	[31: 0]
	总线主控 Xfer 控制	说明	位
	连接	连接而不是传送	18
30	传送完成	传送之后向 FIFO 发送 TD	17
	BTU 完成	传送之后向 FIFO 发送 BTUD	16

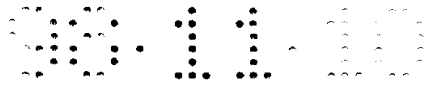




之后，通过环形优先权判优器使BTU循环，直到再次选择到这一BTU来控制总线主控脉冲串数据传送。零传送长度的BTU可以通过总线主控程序总线的低级部分把连接标识符适当地传送给主机或是DSP总线主控单元160，168。将BTU存储在系统主存储器14中，就使用总线主控程序总线的高级地址部分同时向主机总线主控单元160提供基于主机的连接标识符。RAM 34内的BTU单独被存储在连接标识符的地址范围之内。无论是在什么情况下，连接控制位都被设定成连接的BTU传送脉冲串正在编程的信号。主机和DSP总线主控单元160，168的内部FIFO至少应该具有与BTU规模相等的深度。这样，就能利用单一的总线主控脉冲串传送从系统主存储器14或是DSP RAM 34中恢复所需的BTU。在所需的BTU被存入一个总线主控的内部FIFO之后，主机或是DSP总线主控单元160，168就请求对总线主控程序总线142进行访问，以便向总线传送控制系统170传送这一BTU。访问请求是通过在总线主控程序总线142上维持一个请求控制信号来完成的。总线传送控制系统170产生的应答信号同意对一个主要用于连接到FIFO库总线144的字宽更新/连接数据总线173进行唯一的访问。

表 VII

FIFO 更新总线	说明	位
更新标志	表示一个更新循环	1
FIFO 编号	被更新的 FIFO 编号	5
读出大小	FIFO 内的数据字节数	6
写入大小	FIFO 内的打开字节数	6
25 读出字节允许	在 FIFO 内读出字节选择	4
写入字节允许	在 FIFO 内写入字节选择	4
BTU 完成	BTU 完成标志到标志存储	1
传送完成	信号传送完成	1
可用空间	FIFO 内的可用空间	1
30 可用数据	FIFO 内的可用数据	1



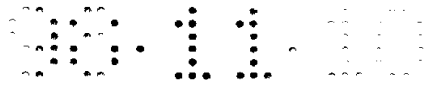
用总线传送控制系统170驱动一个二位总线主连接地址总线，用来最佳地配合为了向总线传送控制系统170内部的BTU表传送完整的BTU时所需的三个更新/连接数据总线存取循环；第四位目前是保留的，没有传送。有效的对外连接的BTU用对连接的BTU重写。由总线传送控制系统来保持当前BTU在BTU表中的位置，使得对BTU的状态更新操作接在每次总线主脉冲串数据传送之后。如果当前的BTU也是一个对外连接BTU，就直接提供更新操作，用来代替对内连接BTU对对外连接BTU的重写。然后使对内连接BTU循环进入环形优先权判优器的后续仲裁操作。

10 IV. 分布式状态和控制系统

总线传送控制系统170主要用于针对FIFO库172来管理通过总线主接口148，150的数据传送。然而，FIFO库172本身执行的存取控制逻辑用来集中控制对FIFO库总线144数据存取循环的特许。如表VIII中所示，FIFO库总线144应该包括32位宽度的数据总线，五位宽度的地址总线，以及唯一存取请求和特许控制线，用来允许主控接口148，150和集成的外围设备以及外围设备接口174-182单独地对FIFO库总线144的存取请求，并且用来接收各个存取应答信号。

表 VIII

20	FIFO 库总线	说明	位
	请求	对 FIFO 库存取的线路请求	10
	请求完成	本次传送中对 FIFO 库的最后请求	10
	应答	用于 FIFO 库存取的应答线路	10
	FIFO 编号	被存取的 FIFO 编号	5
25	允许字节	FIFO 内的字节选择	4
	数据输入	从设备到 FIFO 库的数据	32
	数据输出	从 FIFO 库到设备的数据	32
	方向	数据传送方向	1
	BTU 完成	BTU 完成标志到标志存储	1
30	数据有效	FIFO 中的数据有效	1

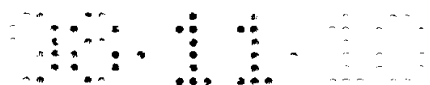


FIFO更新总线(表VI)与FIFO库总线相联系,并且通常是它的一部分。FIFO更新总线最好包括一个五位FIFO状态总线,至少被用来向连接到FIFO库总线144上的所有集成化外围设备和外围设备接口174-183发布FIFO库172内部的一个FIFO的FIFO标识符。在向FIFO库172中相应的FIFO增加数据或是从中提取数据时通过更新总线来发布FIFO标识符。通过有效地发布FIFO库172内的FIFO的状态变化提示,FIFO库可以潜在地简化在FIFO库172内部环形优先权判优器执行的必要操作,并且可以简化各个集成化外围设备和外围设备接口174-183内部所需要的相应的控制逻辑。

分布式控制逻辑被提供给连接到FIFO库总线144上每个集成化外围设备和外围设备接口174-183。这些集成化外围设备和外围设备接口174-183中每一个最好是可以一或多个FIFO标识符编程的,其对应着FIFO库172中用来支持其各自功能需要的FIFO的数量。每个FIFO支持一个无方向性的数据流,因此,全双工通信需要有如此的FIFO对。集成化外围设备和外围设备接口174-183中各自的分布式逻辑监视通过FIFO状态总线发布的FIFO标识符,用来与它们预先编程的标识符进行匹配。在发现一个匹配的标识符时,集成化外围设备或是外围设备接口174-183就可以在FIFO库总线144上发出一个请求数据传送循环的控制信号。FIFO库172按顺序发出唯一应答信号特许访问FIFO库总线。这样,FIFO库172的环形优先权判优器就仅仅需要考虑来自那些可以执行实时数据传送的集成化外围设备和外围设备接口174-183的访问请求。另外,由于可能出现任何数量和集成化外围设备和外围设备接口174-183类型的组合,FIFO库172的环形优先权判优器与需要考虑任一个特定的集成化外围设备和外围设备接口174-183所需要的任何特殊方式的数据传送循环有效分离。可能对FIFO库总线144提出适当的存储器存取循环请求影响的具体操作是由各个集成化外围设备和外围设备接口174-183在其内部根据具体的功能来管理的。接着通过I/O控制器140有效地分配存取确定和仲裁控制逻辑的基本部分。关于FIFO库172的环形优先权判优器的控制逻辑不需要对在不同规定的I/O通道控制器核140执行之间变化的任何集成化外围设备或外围设备接口174-183的类型,数量或是功能进行根本或甚至是明显的修改。

30

## V. 集成化外围设备及其接口



集成化外围设备和外围设备接口174-188各自具有基于寄存器的接口，通过PIO总线146接到剩余的I/O通道控制器核140。主处理机12和数字信号处理器32，36可以通过主机和DSP总线受控单元158，166访问每一个这种寄存器接口。由每一例集成化外围设备和外围设备接口174-188提供给PIO总线的寄存器接口可由简单变到明显地复杂。总而言之，需要采用寄存器接口对外围设备174-188的要求的操作进行编程，使外围设备174-188复位，在可能产生中断并且向外部处理器发出中断时用于指定条件，以及在外围设备174-183也连接到FIFO库总线144上时用于存储由特定的外围设备174-188支持的每一个并行数据流的FIFO标识符。

另外，每个集成化外围设备和外围设备接口174-183还包括一个数据分级的FIFO，用来临时存储发往或是来自FIFO库总线144的数据。这种内部FIFO最好应该具有足够的深度，防止在集成化外围设备或是外围设备接口174-183原先支持的数据通过量速率下发生数据的欠载和溢出，以及在最坏情况下出现在FIFO库172和特定的集成化外围设备或是外围设备接口174-183之间的总线数据传送等待时间。其中原先通过集成化外围设备或外围设备接口的数据传送速率足够低，内部FIFO可以仅仅用一单个缓冲寄存器来简单实现。然而，外围设备的内部FIFO通常具有二到四字节的深度。

一种优选的MIDI接口外围设备接口174构成了PIO总线146的一个比较简单的寄存器接口。用MIDI接口外围设备174构成的这种优选控制寄存器包括一个发送控制寄存器和一个接收控制寄存器。表IX表示这种寄存器接口的细节。

表 IX

MIDI XMit 控制寄存器	说明	位
保留	没用，或是测试值	[31: 7]
发送允许	允许MIDI 发送	6
复位	MIDI 发送端口复位	5
FIFO 编号	来自 FIFO 库的 FIFO	[4: 0]



	MIDI 接收控制寄存器	说明	位
	保留	没用, 或是测试值	[31:10]
	中断选择	00=>没有产生中断	[9:8]
5		01=>中断主机 CPU	
		10=>中断 DSP 1	
		11=>中断 DSP2	
	接收允许	允许MIDI接收	7
	复位	MIDI发送端口复位	6
10	FIFO 编号	来自FIFO库的FIFO	[5:0]

发送控制寄存器包括一个五位的FIFO ID字段, 一个复位位, 和一个发送允许位。接收控制寄存器应该包括一个五位的FIFO ID字段, 一个复位位, 一个接收允许位, 以及一个二位的中断字段, 它编码表示是否响应通过外部接口或是MIDI接口外围设备174接收到的数据字节或是字的匹配错误状态而产生一个中断, 如果是, 就结束中断。这种外围设备174应该是仅仅在错误状态下才产生中断。最好是采用逻辑连接的直接传送端口来产生发送/接收中断的数据, 用来控制通过MIDI接口外围设备174的数据流。

CODEC外围设备接口176的寄存器接口可能比MIDI接口174要复杂得多。其中把普通的外部CODEC连接到CODEC接口176上, 寄存器接口需要提供对外部CODEC本身的控制和数据寄存器的存取。或者是, CODEC外围设备接口176本身就可以完成普通CODEC的部分或是全部功能, 这样就会把需要连接到CODEC外围设备接口176上的外部部件限制在比较少的模拟部件。在后一种情况下, CODEC接口176是一种完全集成化的外围设备, 它可以向PIO总线146直接提供CODEC控制寄存器。在任何一种情况下, CODEC接口176都包括必要的逻辑, 允许CODEC接口176对FIFO库总线144的存取进行仲裁, 并且对应于CODEC接口176寄存器中编程的FIFO标识符来监视FIFO库172中的FIFO的状态变化。

图5c表示用来支持一个外部CODEC, 例如一个模拟器件 1843 CODEC的CODEC接口176的最佳实施方案。基本的CODEC接口数据路径和控制逻辑180为在FIFO库172内部潜在分配的各对FIFO之间和到CODEC接口176的内部的FIFO





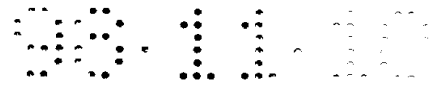
对通过FIFO库总线144的数据部分的多个基本的双向数据提供支持。最好是能够通过四个内部FIFO210-212来缓冲四个输出数据流。一个数据多路转换器216可以按照外部CODEC产生的控制模式分别从FIFO210-214中提供选择8或16位的数据。字节或是字数据流在寄存器接口和I/O控制单元220的控制下通过一个时钟锁存和串行转换器218。寄存器接口和I/O控制单元220通过时钟和控制线222提供时钟和控制信号。寄存器接口和I/O控制单元220通过在线224上交换时钟和控制信号来协调外部CODEC的数据传送。这样，通过寄存器接口和I/O控制单元220的操作，就可以在串行数据输出线226上向外部CODEC提供串行化的数据。

CODEC模式指令以及关于数据的指令也从串行输出线226上提供给外部CODEC。模式指令和数据最好是通过PIO总线146提供给寄存器接口和I/O控制单元220。然后，在通过线214发出一个指令模式交易时，模式指令和数据就会通过线214，多路转换器216，以及锁存器218发送。

一个时钟逆串行转换和锁存器230在线226上接收来自外部CODEC的串行数据。根据外部CODEC的当前指令模式，通过一个多路分配器232和通过线234向寄存器接口和I/O控制单元220提供状态信息和有关的数据。其他字节或是字宽数据在寄存器接口和I/O控制单元220的控制下有选择地从多路分配器232提供给四个外部FIFO 236，238中的任何一个。

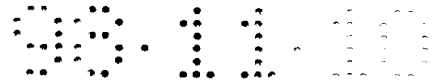
寄存器接口和I/O控制单元220还要提供请求和应答控制逻辑，用来支持FIFO库总线144上的数据传送。寄存器接口和I/O控制单元220应该提供多达8个FIFO标识符的寄存器容量。每个FIFO标识符允许一个特定的内部FIFO与FIFO库172中的任何一个FIFO取得唯一的联系。用寄存器接口和I/O控制单元220连续地监视PIO总线144的FIFO更新部分，用来识别一个标识的FIFO的状态在何时发生变化。寄存器接口和I/O控制单元220以其FIFO标识符符合更新总线上发布的FIFO标识符的该内部FIFO的名义发出一个FIFO库存取请求。

主处理机12可以通过PIO总线146对FIFO标识符寄存器和其他的CODEC接口控制寄存器进行访问。在表X中表示了提供给PIO总线146的最佳CODEC寄存器接口。



CODEC 寄存器	说明	位
5	主控	DAC/ADC 控制 [15]
	音频指令	音频 CODEC 的指令 [14: 9]
	音频数据	音频 CODEC 的模式设定数据 [8: 3]
	电话指令	电话 CODEC 的指令 2
	电话状态	电话 CODEC 的模式设定数据 1
	状态指令	只读 CODEC 状态位 8
	状态数据	只读模式设定数据
	FIFO 编号 0	FIFO 允许位和库号 8
10	...	... 8
	FIFO 编号 7	FIFO 允许位和库号 8
	时钟	时钟计数器控制 16
	主机屏蔽	主机中断屏蔽值 12
	DSP 屏蔽	DSP 中断屏蔽值 12
15	中断控制	传送完成, 欠/超负荷运行位 12

主控	说明	位
20	CODEC 复位	复位所有涉及 CODEC 的逻辑 [15]
	DAC 模式	设定 DAC0, 1 操作模式 [14: 9]
	ADC 模式	设定 ADC0, 1 操作模式 [8: 3]
	音频允许	音频 CODEC 复位 2
	电话允许	电话 CODEC 复位 1
	CODEC 标识符	外部 CODEC 类型 0
25	中断控制	说明 位
	传送完成 FIFO 0	传送完成中断位 1
	...	
30	传送完成 FIFO 3	传送完成中断位 1
	欠运行 FIFO 0	欠运行中断位 (DAC1) 1



...			
欠运行 FIFO 3	欠运行中断位 (DAC4)		1
超负荷 FIFO 4	超负荷中断位 (ADC5)		1
...			
5 超负荷 FIFO 7	超负荷中断位 (ADC8)		1

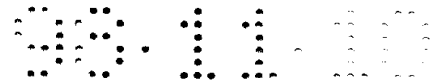
---

10 遥控外围设备和潜在的有关控制台可以通过数字遥控接口178得到支持。遥控外围设备通常提供的电阻是与遥控器的实际位置成比例的。数字遥控接口178最好构成一个定时器和一个电容器，它与遥控外围设备的电阻的组合可以反复地用数字值来表示由数字遥控接口自动产生的遥控器位置。这样，在确定遥控器位置的工作中就可以减轻主处理机12的定时器设定和功能的需求。数字遥控接口174应该是可以编程的，在确定了数字值发生变化时向主处理机12或是DSP32, 36发出中断信号。即使是在不产生中断的情况下，也能够通过由数字遥控接口178提供的寄存器接口从PIO总线146上连续地存  
15 取这种数字值。

20 根据任意相关的控制台的构造及其数据需求，例如是为了支持各种游戏和控制应用，一个数据流可以通过FIFO库172和数字遥控接口178。其中数据流的传送速率很低并且导致等待时间，在到或是从数字遥控接口178传送数据时就应该采用逻辑连接的直接传送端口为主处理机12或是一个DSP 32, 36产生中断。其中等待时间不是主要问题，或者是需要的数据速率比较高，就可以采用通过FIFO库172的全总线主控数据传送方式。

25 另一种类型的集成化外围设备或外围设备接口是由通用的接口端口180构成的。在一个最佳实施例中，接口端口180可以实现很宽的高速并行端口，可以适用于视频控制器的接口，或是直接用于视频帧缓冲器的接口。从FIFO库总线144上传送的顺序的双字宽数据可以缓冲成一个64位，128位，或是更大的字，并且并行地传送给视频控制器或是帧缓冲器。在本例中，在接口端口180内可以用寄存器缓冲一或多个视频字，这样就能使传送给控制器或帧缓冲器的字获得良好的定时匹配。

30 还可以利用接口端口180对视频控制器或是帧缓冲器进行高速的读取。在按顺序并行读出的操作中可以从视频控制器或是帧缓冲器中把一两个甚



至多个帧的字读入一个字缓冲寄存器，也可以针对FIFO库总线144构成一个内部FIFO。这样就能到和通过FIFO库172传送帧字中的双字宽部分。

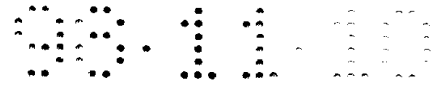
直接传送端口集成化外围设备182被用来通过FIFO库172为典型的窄带宽，对等待时间敏感的数据流产生一种高速，低等待时间的连接。直接传送端口集成化外围设备182提供了六个独立的传送端口。每个传送端口是由可以通过PIO总线146进行存取的一个控制寄存器和一个数据寄存器构成。在表XI中提供了控制寄存器的定义。

10

表 XI

直接传送端口 Ctl.	说明	位
空间/有效数据	设定空间/数据的有效性	15
允许	传送端口允许标记	14
15 复位	传送端口复位	13
数据宽度	传送 8 或 16 个数据	9
方向	传送给 FIFO	8
中断选择	00=>不产生中断	[7: 6]
	01=>中断主机 CPU	
20	10=>中断 DSP 1	
	11=>中断 DSP 2	
FIFO 编号	FIFO 库编号	[4: 0]
直接传送端口 Ctl.	说明	位
25 高位字节	字数据的高位字节	[15: 8]
低位字节	数据的低位或是仅有的字节	[7: 0]

一个直接传送端口的控制寄存器可以用一个FIFO标识符，一个中断目标，一个数据传送方向，数据字节或是字宽，一个复位位，一个允许位，以及一个空格或是数据有效位对端口编程。

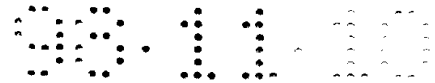


直接传送端口的典型操作是用于监视PI0总线146的更新总线部分中一个标识的FIFO的任何状态变化，并且根据编程的中断选择位直接向主处理机12或是DSP32, 36发出一个中断。在两个处理器之间利用两个直接传送端口和FIFO库172中的一单个FIFO构成一条数据流传送路径。如果指定FIFO的空间是有效的，就应用一个直接传送端口的程序发出主机中断。第二直接传送端口的程序可以在同一识别的FIFO中的数据有效时发出一个DSP中断。为了起  
5 启动数据流的传送，并且随之响应主机的中断，主处理机12可以通过主机总线受控单元158和PI0总线146向联系着主处理机12的直接传送端口的I/O数据寄存器传送数据。直接传送端口把接收的数据放在FIFO库172中识别的  
10 FIFO中。

根据FIFO库172在FIFO状态总线上播发的FIFO标识符，第二直接传送端口发出一个DSP中断。DSP32, 36作出响应，访问直接传送端口的I/O数据寄存器。从FIFO库172的FIFO中读出数据，并且传送给DSP 32, 36进行处理，通常写入RAM 34。

15 直接传送端口可以被用在与总线主控接口148, 150和其他访问FIFO库172的集成化外围设备和外围设备接口174-183的各种组合中。例如，通过MIDI接口外围设备174的输出数据流可以由主处理机12传送给FIFO库172的一个FIFO。在通过更新总线播发FIFO状态的变化时，MIDI接口外围设备174通过FIFO库判优器发出一个访问FIFO库172的请求。一旦得到访问的授权，  
20 就把FIFO库172的数据传送给MIDI接口外围设备174的内部FIFO，并且迅速地输出。

当数据从FIFO库172识别的FIFO中被移出时，另一个已经编程的直接传送端口就作出响应，更新同一个识别的FIFO，不产生主机中断。在本例中，由主处理机12来确定把MIDI数据写入识别FIFO的定时。当主处理机12决定要  
25 写入MIDI数据时，就通过与直接传送端口关联的主机寄存器用分级的数据将新的数据写入识别的FIFO。其中需要以最大MIDI数据速率来输出数据，主处理机12可以在FIFO库的FIFO中写入超过一个字节或字的数据。然后，主处理机12可以根据由主处理机确定的时间表来恢复输出的MIDI数据。接着还可以为从主存储器14到MIDI接口外围设备174的数据传送维持一种低并且基本上  
30 固定的等待时间循环。



通过MIDI接口外围设备174输入的数据需要按照时间标记数据存储在系统主存储器14或是RAM 34中。这种时间标记数据需要从MIDI接口外围设备174实际接收到的时间起按照最小和规则的等待时间提供给输入数据。因此，应该有一个直接传送端口与FIFO库172中的MIDI输入数据FIFO相关联，并且在识别的FIFO内的数据变成有效时应用程序向主机12或是DSP 32发出一个中断。

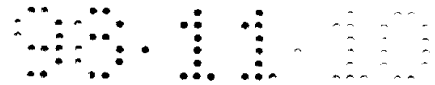
也可以把一个DSP 加速器集成化外围设备183作为I/O通道控制器140的一部分。一般形式的DSP加速器183应该经过PIO总线146构成一个可用的寄存器接口，用来支持与FIFO库172的一或多个输入FIFO的连接，并且支持一或多个输出FIFO通过FIFO库总线144把数据送回FIFO库172。从内部来看，DSP加速器183可以用作专用的硬件，微控制器，或是双重结构来实现，构成一个可编程的控制寄存器结构的，任意多个集中计算的数据处理工作，诸如数字信号滤波，混频和调制等等。因此，可以连接到DSP 加速器183上的FIFO库172的FIFO数量取决于需要由DSP加速器 183完成的具体可能的功能。

信息端口集成化外围设备184提供了一条简单的数据路径，用来在主机和连接到I/O通道控制器140的DSP 处理器之间传送少量的数据。在本发明的最佳实施例中，在信息端口集成化外围设备184内部提供了分别专用于由具体的主处理机12或是DSP 32, 36使用的一个源的三个信息端口。每个源的专用信息端口是用可以从PIO总线146上访问的两个16位寄存器提供的。表XII表示了这种信息端口组以及每个信息端口的控制和数据寄存器。

20

表 XII

信息端口	说明
25 端口 0	主机信息发送端口
端口 1	DSP 1 信息发送端口
端口 2	DSP 2 信息发送端口
端口#的信息端口控制寄存器	说明
30 复位	传送端口复位
	位
	15



中断选择            00=>不产生中断                            [14: 13]  
                       01=>中断主机 CPU  
                       10=>中断 DSP 1  
                       11=>中断 DSP 2

5    存取指针            信息中的字偏移指针                        [7: 0]

端口#的信息端

口数据寄存器	说明	位
数据	读/写数据寄存器	[15: 0]

10

存取指针字段操作作为一个自动增量指针，进入到维持在信息端口内部的一个八字(8×16位)深度的信息FIFO。为了发送信息，源处理器要清除存取指针字段，在信息端口的数据寄存器中写入达八个数据字，并且编程一个用来规定这一信息的目标处理器的中断。在中断选择字段被置为非零状态时有效地发出这种中断。

15

为了响应一个信息中断而读出一个信息，目标处理器可以读出适当的存取指针字段，以识别其信息长度，将存取指针字段置零，从起点上读出信息，然后从信息端口的数据寄存器中读出存取指针所规定数量的数据字。在信息数据被读出之后，可以把中断选择子字段简单复位到零。或者是可以使这一子字段复位到一个值，用来规定具有零长度信息的信息源处理器在接收到信息时发出一个应答。应答或是返回信息也可以在设定用来规定源处理器的中断选择子字段之前被写入信息FIFO。在所有情况下，目标处理器都可以通过识别用来传送信息的特定信息端口而唯一地识别出源处理器，因为每个信息端口与一个源处理器的关系是唯一的。

20

25

一个定时器集成化外围设备186被用来支持通用和专用功能的时基操作。在本发明的最佳实施例中，用PIO总线146上的定时器外围设备186构成了三个双字寄存器，其细节如表XIII中所示。

30

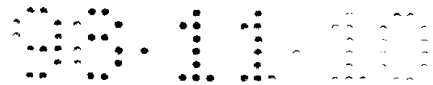


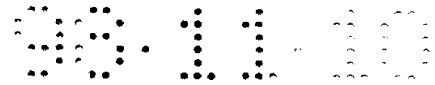
表 XIII

定时器源寄存器		
	说明	位
5	定时器复位	复位所有定时器
	计数值	设定时钟源的基本频率
	递减计数	寄存器递减计数
定时器 0 寄存器		
	说明	位
10	允许	定时器允许
	模式	多发射/单发射模式
	清除中断	清除中断定时器状态
	中断选择	00=>不产生中断
		01=>中断主机 CPU
15		10=>中断 DSP 1
		11=>中断 DSP 2
	定时器值	定时器递减计数值
主机等待定时器		
	说明	位
20	定时器值	定时器递减计数值

25 第一个这种控制寄存器被用作一个15位的自激励计数器，它被作为其他定时器功能的可编程时基。定时器源寄存器应该包括一个15位的递减计数字段，它采用提供给I/O控制器140的系统时钟的每个循环来计数。一个15位的计数值字段用来存储一个静态的基时钟计数值，它被用于在每当递减计数的子字段达到零值时对递减计数子字段进行重新初始化。用一个单独的位作为整个定时器外围设备186的复位控制。

30 一个通用定时器零寄存器被用来在16位递减计数定时器在定时器基本时钟的驱动下每次期满时产生一个中断。一个定时器零寄存器包括一个16位的定时器值，它在完成定时器基本时钟的每个循环时被递减。一个两位子字





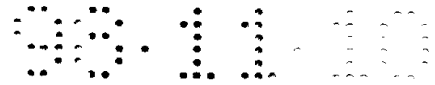
段被用来规定处理器在定时器终止到零值时被中断。同时还提供了中断清除和定时器允许位。最后，用一个模式控制位来规定是否要把定时器零当作一个多发射定时器或是一个单发射定时器。在多发射模式下，每当定时器值期满时就产生一个中断，并且重新装载定时器值。仅有在前一个中断被清除之后才会发出第二个中断。在单发射模式下，每当产生一个中断时就直接停止定时器。

还可以构成一个特殊的主机中断等待时间定时器。这种定时器是一个由定时器基本时钟驱动的增量计数定时器。一个16位定时器值随着对主处理机12确定的一个中断从零开始增加。这一主机中断等待时间定时器在主机中断等待时间定时寄存器的定时器值子字段被读出时被停止，并且清除定时器值。这种特殊的定时器允许I/O控制器核140通常通过主处理机12支持分析，根据I/O通道控制器140产生的中断而有效地执行主机中断服务程序。

最后，在典型的操作期间用一个中断控制器集成化外围设备188为I/O通道控制器140产生的许多不同的中断源的管理提供支持。为了在主机和DSP处理器上有效地使用可用的中断输入，I/O控制器140应该向每个处理器发出一个单一矢量的中断。如图5d中所示，在本发明的实施例中，用一个中断寄存器242在线156'上产生一个主机中断，在线164'上产生第一DSP中断，并且在线165'上产生第二DSP中断。中断寄存器246的输入244是从I/O控制器140内部的独立中断源或是成组中断寄存器246接收到的。可以通过PIO总线从中断寄存器242中读出成组中断。在表XIV中提供了编组中断的基本定义。

表 XIV

25	群中断寄存器	说明	位
	定时器屏蔽	允许中断源	12
	直接 XFer 端口屏蔽	允许中断源	11
	信息屏蔽	允许中断源	10
	外围设备屏蔽	允许中断源	9
30	BTU 屏蔽	允许中断源	8
	定时器中断	群中断有效位	4



直接 XFer 端口中断	群中断有效位	3
信息端口中断	群中断有效位	2
外围设备中断	群中断有效位	1
BTU 中断	群中断有效位	0

5

中断寄存器 242 应该能够屏蔽每一群中断。如果采用中断屏蔽允许一个中断群产生一个中断，就用具有群中断信号 244 的中断目标信息来选择具体的中断目标 156', 164', 165'。接着，尽管中断是通过中断寄存器 242 进行编组的，特定中断目标的单独中断源仍然是确定的。

10

如果要用 I/O 控制器 140 的单个部件例如总线传送控制系统 170 来产生大量的中断，或是许多相关的部件可以按照惯例编在一起，中断请求线 248, 252 就可以连接到第二级编组寄存器 246, 250。在本发明的一个实施例中，有许多集成化外围设备和外围设备接口 174-186 要经过外围设备编组中断寄存器 246，从而向中断寄存器 242 提供单一成组的中断。例如 MIDI 接口 174 等等内部外围设备，以及连接到 DSP 总线 30 上的那些外部外围设备各自都可以作为外围设备成组中断的源。表 XV 为可以通过 PIO 总线 146 进行存取的成组屏蔽和中断状态寄存器提供了基本的定义。

15

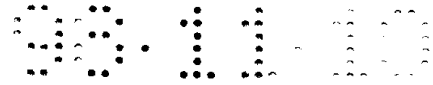
表 XV

20

群中断寄存器	说明	位
中断 7 屏蔽	允许中断源 7	15
...	...	
中断 1 屏蔽	允许中断源 1	9
25 MIDI 屏蔽	允许 MIDI 中断源	8
外部中断 7	外部中断 7 有效位	7
...	...	
外部中断 1	外部中断 1 有效位	1
MIDI 中断	外围设备中断有效位	0

30

群中断寄存器 250 应该提供一个用来反映每个直接传送端口中断的中断状态



的直接端口下标字段。用单个位的标记来表示当前可以读出的下标是否有效。可以通过 PIO 总线 146 来访问直接传送端口寄存器(表 XVI)。

5

表 XVI

ITP 中断寄存器	说明	位
ITP 下标无效	有效/无效标志; 写入无效数据	7
ITP 下标	ITP 阵列中的 ITP 下标	[2: 0]

10

最后用一个 BTU 群中断寄存器 254 提供一个控制寄存器(表 XVII), 它包括一个 BTU 索引字段, 用来识别作为中断信号 256 的源的一个具体的 BTU。

表 XVII

15

BTU 中断寄存器	说明	位
BTU 下标无效	有效/无效标志; 写入无效数据	7
传送完成	BTU 传送完成	6
BTU 下标	BTU 表中的 BTU 下标	[5: 0]

20

用一个独立的传送完成位指示出一个中断是否代表了数据流的传送已经完成。最后, 用一个信号控制位来规定读出时的 BTU 索引字段是否有效。

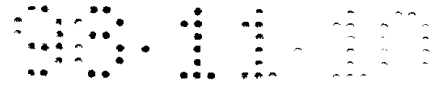
这样, 中断控制器集成化外围设备 188 就可以根据任何大量的独立内部中断为任何数量的不同处理器快速提供一种简单的寄存器系统 240。当一个处理器被中断时, 为了唯一地识别出这一中断源, 最多只需要有两个通过 PIO 总线 146 的寄存器读出操作。

25

## VI. 多路径弹性缓冲 FIFO 库

在图6中用260详细表示了FIFO库172以及许多FIFO存取请求设备262<sub>0</sub>-262<sub>n</sub>之间的控制关系。FIFO存取请求设备262<sub>0</sub>-262<sub>n</sub>最好包括总线主控接口148, 150以及集成化外围设备和外围设备接口174-183。每个设备262<sub>0-n</sub>连续

30



地监视FIFO更新总线173用于对指定到特定设备260<sub>0-n</sub>的任意FIFO的状态的改变进行监视具体地，每个设备260<sub>0-n</sub>监视更新总线，对被更新的FIFO数目进行监视，并且确定在更新的FIFO中是有足够的有效空间还是数据可用来支持设备262<sub>0-n</sub>和FIFO库存储器284之间通过FIFO库总线144的数据传送。

5 根据在分配给一个特定设备2620-n的一个FIFO中播发的限制的状态变化，设备2620-n通过FIFO库总线144向FIFO判优器264播发一个存取请求控制信号。在每一个判优循环中，FIFO判优器264实际上仅仅认为这样的设备2620-n是由一组悬而未决的请求控制信号唯一地代表那些请求对FIFO库284进行的存取。在FIFO判优器264的当前判优循环中的下一个设备被选中使有  
10 权对FIFO库284进行存取。由FIFO判优器264向选中的设备2620-n送回一个唯一的应答信号。该设备2620-n响应这一信号，通过FIFO库总线144把至少包括由设备2620-n维持的FIFO标识符在内的控制信号提供给选择器，一个FIFO标记存储器286内的数据存储线，以及包括一个FIFO读/写指针存储阵列272，274的一个FIFO控制存储器。

15 标记存储器和指针阵列286，272，274最好是由一个RAM寄存器阵列构成的，其数量上的深度等于由FIFO库284所支持的FIFO数。在本发明的实施例中，FIFO库可以存储24个16字节深度的FIFO。在表XVIII中列出了FIFO标记存储器286中每条线的标记位的定义。

20

表 XVIII

FIFO 标志	说明	位
传送完成	传送完成的标志	12
25 BTU 完成	当前 BTU 完成的标志	11
FIFO 长度	FIFO 长度大于 16 字节	10
读出长度	FIFO 中的数据字节数	[9: 4]
读出字节允许	在 FIFO 内选择读出字节	[3: 2]
写入字节允许	在 FIFO 内选择写入字节	[1: 0]

30



从FIFO库总线144提供到线276上的FIFO数被用作在FIFO标记存储器286中唯一地选择一条阵列线的地址，用来更新存储在选定阵列线中的控制数据。FIFO数进一步被用来在FIFO读、写指针阵列272，274中选择一条对应的阵列线，允许对识别的FIFO使用读出或是写入指针。最后把线276上的FIFO数提供给一个I/O多路复用器282，允许在FIFO库存储器284中识别的FIFO和FIFO库总线144之间传送数据。

在表XIX中给出了用来组合读出和写入指针阵列272，274的字段定义。

表 XIX

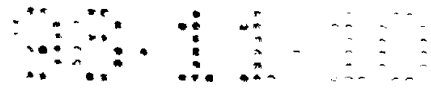
10

FIFO R/W 指针	说明	位/s
读出指针	当前 FIFO 读出位置	[15: 9]
写入指针	当前 FIFO 写入位置	[8: 2]
FIFO 长度	FIFO 的分配长度	[1: 0]

15

长度字段限定了FIFO的长度。在本发明的实施例中，一个FIFO长度值“00”代表一个16字节的FIFO深度。值“01”表示32字节的FIFO深度，而值“10”代表64字节的FIFO深度。读出和写入指针的作用是在当前选定的FIFO中提供六位的FIFO字节偏移。每个读出和写入指针中的高位规定了16字节FIFO的第5位，32字节FIFO的第6位，以及64字节FIFO的第7位，它被用来指示在对FIFO变址的过程中是否出现了FIFO绕回的情况。

为了在FIFO库284内简化FIFO的分配和管理，FIFO标记存储器286和读、写指针阵列272，274中的每个阵列线对应着FIFO库284内部指定的FIFO RAM块。每个FIFO RAM块应该是位于FIFO库284内部的一个16字节边界被按16字节FIFO RAM的顺序对齐。因此，FIFO标记存储器286和读、写指针阵列272，274的最大相互深度确定了在FIFO库284内部可以同时支持的FIFO的最大数量。在此由读、写指针阵列272，274的长度字段所限定的FIFO字节长度是32或64，就在FIFO库284内部采用二或四个相邻的FIFO RAM块。这样，每个32字节FIFO或64字节FIFO的规格就需要占用独立使用的一或三条额外的阵列线。



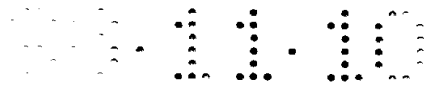
FIFO指针逻辑单元266和包括读、写指针阵列272, 274在内的FIFO控制存储器共同操作构成了一种状态机器, 它是在选择了一个设备2620-n参与对FIFO库284的存取时由FIFO判优器264通过线268上的一个允许信号来起动的。选定的设备2620-n将其FIFO编号提供给线276, 用来选择对应的FIFO读、写指针, 允许的有效字节, 以及例如数据传送方向的其他控制信息, 通过FIFO库总线144提供给FIFO指针逻辑单元266。数据传送方向控制信号确定了是否要把FIFO读、写指针阵列值提供给I/O多路复用器282。允许的字节限定了在FIFO库284和FIFO库总线144之间执行的数据传送的宽度和顺序, 这样就能在交替存取FIFO库284内部的一个共同的FIFO的任何设备2620-n的数据流传送宽度之间实现独立性。

在经由FIFO库总线144通过I/O多路复用器282完成了由选定的设备2620-n所确定的每一次读出或写入循环时, FIFO指针逻辑单元266就在FIFO控制存储器272, 274内适当地更新读、写指针阵列值。FIFO指针逻辑单元266还要在FIFO标记存储器286内更新对应的FIFO标记的线。具体地说, FIFO指针逻辑单元266要更新被存取的FIFO内剩余的数据字节数以及定性的标记, 例如传送完成标记和BTU完成标记。

在按照可以直接通过或是从选定设备2620-n的内部FIFO接收的字节数以及在FIFO库284中对应的FIFO中可供使用的字节数完成了一或多个字节的数据传送之后, 用选定的设备2620-n取出FIFO存取请求。然后开始一个新的判优循环, 并且通常用FIFO库判优器264来选择下一个设备2620-n对FIFO库284进行存取。

设备2620-n通过FIFO库总线144向FIFO指针逻辑单元266发送一个传送完成信号, 表示有一个设备2620-n的发送或是接收已经结束了。总线主控设备148, 150还会在当前的BTU完成时发出一个BTU完成信号。在两种情况下, FIFO指针逻辑单元266都要潜在地处理局部的FIFO数据传送工作, 它们可能同时保留在一个参与设备2620-n中相应的内部FIFO中, 或是出现在FIFO库284的相应的FIFO中。

基于主处理机12的初始化以及随之对FIFO标记存储器286和FIFO控制存储器272, 274进行的状态检查可以通过FIFO指针逻辑单元266和FIFO库判优器264的协同而有效地执行。最好是通过控制, FIFO索引和数据线290为PIO总线146提供一组可编程的寄存器, 以便从PIO总线146读出或是写入与一个



库FIFO相关联的当前的FIFO标记。在表XX中提供了用FIFO指针逻辑单元构成的FIFO标记阵列的寄存器说明。

表 XX

5

FIFO 标志	说明	位/s
FIFO 标志 0	FIFO 0 的 FIFO 标志	16
FIFO 标志 1	FIFO 1 的 FIFO 标志	16
...	...	...
FIFO 标志 23	FIFO 23 的 FIFO 标志	16

10

在表XXI中有效提供了用FIFO指针逻辑单元266构成的FIFO控制存储器272, 274的寄存器定义。

15

表 XXI

FIFO 指针	说明	位/s
FIFO 控制 Regs 0	FIFO 0 的 FIFO R/W Ptrs	16
FIFO 控制 Regs 1	FIFO 1 的 FIFO R/W Ptrs	16
.....	.....	...
FIFO 控制 Regs 23	FIFO 23 的 FIFO R/W Ptrs	16

20

25

虽然主处理机12和数字信号处理器32, 36都可以读、写FIFO标记和控制存储器272, 274, 包括产生和破坏FIFO库284内部定义的FIFO在内的初始化工作最好还是由主处理机12独自来管理。通过规定主处理机12作为对FIFO库284内部的FIFO的唯一管理者, 就不需要在对FIFO标记存储器286进行存取的主机和DSP之间以及FIFO到FIFO的读、写指针阵列272, 274之间实现互锁。

30

反之可以用信息端口184在主处理机12和DSP32, 36之间传递短信息, 和用来

根据主处理机12或是DSP32, 36发出的请求对FIFO库284内部新近产生的任何特定FIFO的该产生和FIFO标识符作出请求和应答。

由可编程的读、写指针阵列272, 274提供的一个附加特征在于FIFO库284内部的FIFO大小排列可以按照需要而变动, 从而避免数据的溢出和欠运行。通过对保留在FIFO控制存储器272, 274中的容量字段进行重新编程, 可以使FIFO库284内的64字节和32字节FIFO收缩递减降到最小的16字节长度。因此, 在FIFO存储块内的FIFO库284内部就可以分配新的16字节和潜在的32字节FIFO, 或者是在收缩的FIFO中不再使用这些存储块。反之, FIFO长度也可以动态地从16字节生长到64字节, 所需的限制是基本FIFO RAM块应该在FIFO库284内部准确地对齐, 相邻的FIFO RAM 块已经不再使用了, 为了便于对FIFO进行管理, 这种FIFO最好是空的。

### VII. 总线传送单元扫描控制逻辑

在图7中详细表示了总线传送控制系统170。总线传送控制系统170的工作是管理着三个基本上独立的高优先权过程, 这些过程每个都需要访问BTU表192。BTU判优器190作为一个主控制器来接收存取请求, 并且在三个过程当中选择一个对BTU表进行存取。这三个过程包括PCI接口事务处理程序, DSP接口事务处理过程, 以及总线受控事务处理过程。

PCI接口160产生对BTU判优器190的存取和更新请求。一个存取请求表示PCI接口160可以编程, 以便执行主机总线主控的数据传送操作。PCI接口160提出的更新存取请求是在完成了主机总线主控传送时更新对应于BTU表192内部的一个BTU。同样, DSP接口168也向BTU判优器190发出存取和更新请求, 请求来自BTU表192内的一个BTU的程序, 并且在完成了DSP总线主控数据传送时按顺序更新BTU。最后, 每个总线受控单元158, 166可以请求对BTU表192进行存取, 以便编程一个新的BTU, 或是检查当前BTU的内容。

BTU判优器190执行一种固定的优先权选择机制, 用PIO总线请求, 该请求具有最高优先权, 主机和DSP总线主控接口其次, 最后是BTU扫描器。而BTU扫描器则可以保证所有BTU在访问BTU表192时具有一般认为的公平。通过PIO控制器151的操作, 主机和DSP总线受控单元158, 166提出共同的一个存取请求, 并且通过PIO总线146中的控制线从BTU判优器190上接收一个共同的允许响应。由PIO控制器来确定是否允许总线接口模块148, 150对来自BTU判优器190的允许响应作出响应。当通过控制线292向主机或是总线受控单元158,



166提供一个允许PIO信号，一个BTU下标值就可以通过PIO总线146和地址线294提供给一个地址多路复用器296。同时，BTU判优器190通过控制线298提供一个选择信号，把PIO提供的下标选定为一个地址，用来选择进入BTU表192的入口。这样，主机或是DSP总线受控单元158，166就可以对BTU表192内部  
5 对应的BTU进行读出或是写入。

PCI总线主控接口160和DSP总线主控接口168都可以在接口另外处于空闲时向BTU判优器190产生存取请求。BTU判优器190对PCI或是DSP存取请求的授权被有条件地附加在由BTU扫描器196确定的限于对总线主控接口160或168传送的一个有效或是允许的BTU标识上。连接BTU扫描器196和FIFO标记阵列270的控制线171把一个阵列下标提供给标记阵列270，并且送回存储在相应阵列线中的至少一个子集合的FIFO标记。FIFO标记阵列270最好实施为一个双端口的存储器，以便在设备262<sub>on</sub>的存取过程中使BTU扫描器196能够连续地对FIFO标记阵列270独立地进行大量的扫描。用双端口存储器构成的FIFO  
10 标记阵列270提供了适当的存取互锁，当扫描器试图在标记管理器280正在更新的过程中读出一个标记阵列线时，可以暂时停止BTU扫描器196。一般来说，扫描器停止是很少并且短暂的。

BTU扫描器196的工作是对BTU表192执行按规定顺序的扫描，以识别出下一个潜在的BTU，用于PCI总线主接口160或是DSP总线主接口168的编程。BTU扫描器196的扫描顺序是由BTU表192内部的每个BTU入口中允许位的状态来规定的。BTU扫描器196可以通过控制线300直接读出这些允许位的状态。这样就能把BTU扫描器196的操作有效地限制为仅仅对允许的BTU进行顺序扫描。这样，当某些BTU由于某种原因被禁止并且被立即排除到扫描范围之外的同时，在BTU表192中新近产生和建立的BTU被直接纳入BTU扫描范围之内。当BTU扫描器196选定了下一个规定的BTU进行检查时，就通过BTU地址线302  
20 把BTU地址提供给多路复用器296。多路复用器296的缺省状态表示BTU扫描器提供的地址是提供给BTU表192的缺省的允许地址。BTU的至少一部分从BTU表192中被读出，并且存储在扫描寄存器304或是306中。通过控制线308，310把扫描寄存器的内容提供给BTU扫描器196。在表XXII中规定了由扫描寄存器304，306存储的BTU信息。

表 XXII

扫描寄存器	说明	位/s	
5	FIFO 死锁	死锁/不规则传送长度	12
	BTU 完成	在当前脉冲之后完成最佳 BTU	11
	中断允许	在当前 BTU 完成时强制中断	10
	BTU 忙碌	当前正在使用扫描 BTU	9
	超越	将扫描 BTU 移动到最佳 BTU 区域	8
10	编程有效	最佳 BTU 可以用于总线主程序; 当扫描器通过 BTU 下标值时设定	7
	有效	最佳 BTU 具有有效的 BTU 下标	6
	BTU 下标	扫描寄存器中的 BTU 下标	[5: 0]

在表 XXIII 中规定了用最佳的 BTU 寄存器 312, 314 存储的 BTU 信息。

表 XXIII

最佳寄存器	说明	位/s	
20	FIFO 字节允许	FIFO 内部的字节选择器	[15: 14]
	脉冲串长度	传送的字节数	[13: 8]
	传送完成	最佳 BTU 脉冲串后面的数据流结束	7
25	连接	BTU 连接的程序, 不传送	6
	BTU 下标	最佳寄存器中的 BTU 下标	[5: 0]

在表XXIV中规定了用当前的BTU寄存器320, 322存储的BTU信息。

表 XXIV

5	当前寄存器	说明	位/s
	中断选择	00=> 不产生中断 01=> 中断主机 CPU 10=> 中断 DSP 1 11=> 中断 DSP 2	[15:14]
10	脉冲串长度	传送的字节数	[13:8]
	传送完成	当前 BTU 脉冲串后面的数据流结束	7
	连接	BTU 连接的程序, 不传送	6
	BTU 下标	最佳寄存器中的 BTU 下标	[5:0]

15

在表XXV中规定了BTU控制寄存器。

表 XXV

20	最佳寄存器	说明	位/s
	复位	复位整个 BTU 子系统	9
	脉冲串允许	单步 BTU 扫描器	8
	保留		[7:2]
	一直是 V4P	BTU 总是对程序有效	1
25	允许	允许操作 BTU 扫描器	0

根据扫描寄存器304, 306提供的信息, BTU扫描器196确定具体的接口160, 168以及BTU所参考的FIFO。然后通过控制线171提供FIFO下标, 从FIFO  
30 标记阵列270中获得对应的FIFO标记数据。具体地说, BTU扫描器196认为的FIFO标记数据包括参考的FIFO是否能调节或是随即存储用来支持总线主脉



冲串传送的足够的数  
据。在本发明的一个  
实施例中，每次脉冲  
串传送的最大长度  
是16字节。然而，  
BTU扫描器196需  
要FIFO标记中有  
一个是传送完成  
标记，它可以使  
BTU扫描器196  
适当地考虑允许  
小于16字节的脉  
冲串传送。另外，  
设定死锁标记的  
用意是强制执行  
局部数据传送，  
而不需要等待  
全部有效的脉  
冲串。此处由于  
传送数据长度  
不规则或是存  
在某种错误而  
造成只有局部  
脉冲串的传送，  
就可能出现死  
锁。参与的设  
备的最佳数据  
传送长度在  
FIFO中无效，  
这种最佳数据  
长度不能传  
送给FIFO。在  
任何情况下，  
使FIFO保留死  
锁状态的设  
备都会设定  
一个死锁标  
记，强制下  
一个设备在  
访问FIFO时  
采用短数据  
传送。

10 此处BTU扫描器196确定了可以通过有效的总线主控接口160，168执行脉冲串数据传送，就需要在BTU进入就绪时确定标识的BTU相对于最佳BTU寄存器312，314是否具有足够的优先权等级。BTU的优先权等级主要是根据一个BTU在脉冲串数据传送的执行中最近使用的情况来确定的。此处扫描寄存器304识别的一个BTU比最佳BTU扫描寄存器312识别的BTU具有更高的优先权，或者若最佳BTU寄存器312是空的，就在BTU扫描器196的控制下通过控制线316提供适当的控制信号，把扫描寄存器34的内容复制到最佳BTU寄存器312。反之，如果由扫描寄存器306识别的一个BTU的优先权等级比最佳BTU寄存器314指定的BTU低，BTU扫描器196就不会把扫描寄存器306的内容复制到最佳BTU寄存器314。在这两种情况下，BTU扫描器196已经完成了当前的扫描循环，并且开始从BTU表192中选择下一个限定的BTU，以便在总线主脉冲串数据传送的控制中用于下一步的分级。

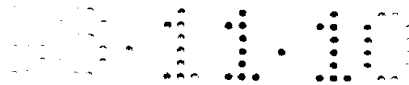
BTU扫描器196与BTU判优器190协同工作，指示出最佳BTU寄存器312，314之一或是双方是否识别了一个BTU用来执行脉冲串数据传送。根据这种信息，BTU判优器190就可以确定是否响应一个存取请求向一个接口160，168提供允许PCI或是允许DSP的信号。在BTU判优器190发出允许PCI或是允许DSP的信号时，BTU判优器190会通知BTU扫描器196。这时，BTU扫描器196就可以允许一位并行传送，把对应的最佳BTU寄存器312，314的内容传送给对应的当前BTU寄存器320，322。同时，BTU扫描器196更新一个内部BTU优先权位目录，在对扫描和最佳BTU寄存器304，306，312，314中识别的那些BTU的相对优先权进行判别时接着使用这种目录。



允许的最佳BTU寄存器312, 314的内容被提供给对应的PCI或是DSP接口160, 168。同时, 通过地址线324向多路复用器296提供一个BTU标识符, 用来为BTU表192中标识的BTU 选择最佳的BTU寄存器。按照BTU判优器发出的允许控制信号, 通过控制线298向多路复用器296提供当前的BTU地址选择信号。这样就能寻址完整的BTU, 并且用于对PCI或是DSP接口160, 168编程, 用来执行脉冲串模式的数据传送。编程的PCI和DSP接口160, 168象任何设备262<sub>0-n</sub>一样可以请求按照脉冲串数据传送的指定方向对FIFO库284进行存取。如果要把数据传送到FIFO库284内部的一个FIFO, 就在设备160, 168请求优先存取FIFO库284之前执行脉冲串数据传送。此处数据是从FIFO库284向外传送的, 为了在执行脉冲串数据传送之前从FIFO获得数据, 首先要对设备160, 168进行仲裁。在PCI和DSP接口内部保留的脉冲串数据FIFO都具有足够的存储器可用来每次缓冲一单个数据脉冲串。在本发明的一个实施例中, PCI内部的脉冲串数据FIFO的深度为32字节, 而DSP的内部FIFO是16字节。

一旦PCI或是DSP接口160, 168完成了相对于FIFO库284的总线主控数据传送, 接口160, 168就请求一次PCI或DSP更新。对BTU判优器190提出的更新请求并不是限定选择一个新的BTU, 而是直接采用由BTU寄存器320, 322当前的内容所标识的BTU。因此, 更新操作用于编程直接领先的脉冲串数据传送使用的BTU脉冲串数据传送。按照BTU判优器190发出的允许PCI或是允许DSP的信号, PCI或是DSP接口160, 168允许将更新的信息写回由对应的当前BTU寄存器320, 322内容识别的BTU。通过这种更新方式, 就能把传送完成标记的状态有效地传送给参考的BTU。FIFO标记存储器内相同的标记也被更新, 虽然这种更新最好可以结合着脉冲串传送本身来进行。在完成了BTU的更新之后, 对应的PCI或是DSP接口160, 168再次向BTU判优器190认定一个存取请求, 并且等待BTU程序。

PCI和DSP接口160, 168可以结合着脉冲串数据传送的执行过程并且潜在地响应数据传送错误被编程来产生中断。PCI和DSP接口160, 168向一个接口中断控制器326提供完成和错误控制信号。在最佳寄存器312, 314的内容被复制到当前寄存器320, 322时, 在BTU表192中被提供或进入的编程PCI和DSP接口160, 168的脉冲串完成响应中断操作被存储在当前的寄存器320, 322中。每当PCI或DSP接口被允许执行一次脉冲串数据传送时, 程序中断响应状态就从当前寄存器320或是322适当提供给接口中断控制器326。在完成了脉



冲串数据传送或是检测到错误状态时，如果产生任何中断，接口中断控制器 326就在控制线328上向主中断控制器188提供中断定义控制信号。

### VIII. 请求驱动的数据传送操作

5 在图 8a 到 8c 中表示了多种数据和控制路径的实例。使用两个 FIFO 和四个 BTU 可以产生一个 CODEC 接口 176 的逻辑双向接口。由于 FIFO 对于数据传送而言是任意地没有方向性的，用一个独立的 BTU 与每个 FIFO 相关联，用来管理数据传送路径。

CODEC 接口 176 作为 FIFO 1 的数据用户来操作。CODEC 接口 176 按照  
10 CODEC 接口 176 的内部操作所产生的数据传送速率请求存取 FIFO 库 284。同时，BTU 1 参与 BTU 扫描器 196 的扫描操作。在 CODEC 接口 176 从 FIFO 1 中提取数据时，FIFO 标记被更新，直到在 FIFO 1 内部有了足够的空间可用于脉冲串数据传送时为止。然后，在 BTU 判优器 190 对 PCI 接口 160 或是 DSP 接口 166 的程序进行选择时，BTU 1 可以入选，用于按照 FIFO 1 的程序所  
15 规定的从系统主存储器 14 或是 DSP RAM 34 到 FIFO 1 的脉冲串数据传送。

在 BTU 1 的传送长度被消耗时，总线主控数据传送可以通过 PCI 或是 DSP 接口 160，168 来执行，用于在 BTU 表 192 中装载下一后续的 BTU，以便维持 CODEC 接口 176 上的连续的数据流。由于流数据可以作为片段存储在系统主存储器 14 或是 DSP RAM 34 中，用来分别识别数据片段的链接顺序的 BTU  
20 可以形成一种有效的耗散/收集机制。由于用来传送流数据的 BTU 的过程以及链式或是连接到 BTU 的再装入过程是相对于主处理机 12 和 DSP 32，36 而自动执行的，特别是在不需要产生要求主处理机 12 或是 DSP 32，36 提供服务的中断的情况下，主处理机 12 和 DSP 32，36 所维持的使用效率基本上与 I/O 通道控制器 22 的输出操作无关。最好是在控制 FIFO 1 流数据传  
25 送的 BTU 链式顺序中的最后一个 BTU 编程时在完成了 BTU 所规定的数据传送基础上起动一个中断。

从 CODEC 接口 176 接收到的数据被请求驱动进入 FIFO 2 用于传送给系统主存储器 14。当 CODEC 接口 176 接收到数据时，接口就请求访问 FIFO 库，按照与 CODEC 接口 176 接收数据的速率协调一致的数据速率来传送数  
30 据。然后，当 FIFO 2 被充满到足够用于总线主控传送的数量时，就可以选用 BTU 2 对 PCI 接口 160 编程，用来执行从 FIFO 2 到系统主存储器 14 的脉



冲串数据传送。

在 CODEC 接口 176 和 FIFO1 或是 FIFO2 之间的数据传送结束时，对应的 FIFO 标记就会被更新，反映出传送完成的状态。这种状态将会影响到最后的链式后继块 BTU 中的 BTU1 或是 BTU2。最后，传送完成会造成一次清除 FIFO 的 FIFO 复位操作。接着，在传送数据给 FIFO1 或是来自 FIFO2 的传送数据结束时可以产生一个中断。

图 8b 表示一种更加复杂的双向数据流传送方式，在其中包含把数据从系统主存储器 14 通过 FIFO 3 传送给一个 DSP 32，再传送给直接连接到 DSP 32 的一个 CODEC 接口 40。由于从系统主存储器 14 传送的数据和传送给 DSP RAM 34 的数据是总线主控的传送，采用了两个独立的 BTU3 和 4 来控制通过 FIFO 3 的数据传送。在本例中，用 DSP32 在 BTU 表 192 中为 BTU 4 编程，同时用主处理机 12 为 BTU 3 编程。由于 DSP 32 可以被用来管理 RAM 34 中由 BTU 4 指定的数据目标部分，可以用 DSP 32 来产生 BTU 4，以及用来规定传送对应的指定长度数据块的链式顺序，并且在向 DSP RAM 34 传送每一个或是仅有最后一个数据块的工作完成时产生一个中断。同时，当 FIFO3 中有足够的空间可以用于数据脉冲串传送时，BTU 3 和链式后继块就可以在总线传送控制系统 170 的操作中生效。在执行对 FIFO3 的最后一次数据脉冲串传送时，BTU3 后面的链式顺序 BTU 应该指定向主处理机 12 产生一个中断。或者是主处理机 12 可以通过信息端口 184 向 DSP 32 发出一个简单的信息，在最后一排空的 FIFO3 上指定出数据流的传送已经完成。

DSP32 从 CODEC 接口 40 接收到的数据可以由数字信号处理器 32 进行处理，并且存储在 RAM 34 内部的缓冲位置中。如果在 DSP RAM 34 中已经形成了一个数据块，DSP 32 就可以把一个 BTU 5 编程入 BTU 表 192。在得到允许之后，就可以用 BTU 5 来编程从 DSP RAM 34 到 FIFO4 的脉冲串数据传送。当 FIFO 4 被填充到足以支持一次脉冲串数据传送时，由主处理机 12 预先产生的一个 BTU 6 将规定用于优先选择，以编程 PCI 总线主控接口 160 使执行脉冲串传送，按照 BTU 6 或是链式顺序的 BTU 指定的地址把数据传送给系统主存储器 14。如果要终止数据流，DSP 32 可以通过信息端口 184 向主处理机 12 传送一个信息，规定需要传送的数据的总数。然后由主处理机 12 产生一个结束 BTU，用来反映剩余数据的总期望值，并且在数据已经被传送到系统主存储器 14 时产生一个中断。然而，DSP 32 则代为指示 DSP 总线主控

单元 150 结合流数据的当前和结束部分的传送而发出一个传送完成信号。

在图 8c 中表示了一个更复杂的例子。通过 FIFO 5 连接一个 MIDI 接口 174，用于接收来自系统主存储器 14 的数据。主处理机 12 最好利用一个直接传送端口 182<sub>0</sub> 向 FIFO 5 提供数据。由于输出 MIDI 数据传送流的定时是由主处理机 12 本身来管理的，主处理机 12 用直接传送端口 182<sub>0</sub> 被动地向 FIFO 5 提供数据。之所以说是被动的，是因为在数据进入 FIFO 5 时不产生中断。反之，MIDI 接口 174 可以在数据存在于相应的 FIFO 5 时提出对 FIFO 库的存取请求。一般来说，完成从 FIFO 5 到 MIDI 接口 174 的数据传送所需的等待时间是很小的，并且可以适应 MIDI 设备的正常数据速率。接着，与本发明中通过 I/O 通道控制器核 26 来管理的其它数据传送的按请求驱动操作方式不同，通过 MIDI 接口 174 对外提供的数据的速率是源，而不是按请求驱动的。

从 MIDI 接口 174 发生或是输入的数据按照由 MIDI 接口 174 产生的接收数据速率被直接提供给一个 FIFO 6。在通过接口 174 接收到数据的每一个字节或是字时，立即发出一个请求，把数据传送给 FIFO 库 186 中的一个对应的 FIFO。再者，一般来说，完成从 MIDI 接口 174 到 FIFO 6 的数据传送所需的等待时间是很小的，并且可以适当与 MIDI 设备的正常数据速率一致。

在这种情况下，每当输入 MIDI 数据被置入 FIFO 6 时，最好建立第二直接传送端口 182<sub>1</sub> 主动地对一个 DSP 36 产生一个中断。DSP 36 响应这一中断，通过直接传送端口 182<sub>1</sub> 从 FIFO 6 中检索数据，记录一个对应的时间标记，并且将二者存入 DSP RAM 34。接着，DSP 36 就可以按照最小的基本固定的等待时间从 MIDI 接口 174 上接收时间标记，并且处理接收的数据。尽管直接传送端口 182<sub>1</sub> 可以产生许多 DSP 中断，MIDI 数据传送的读出速率是非常低的，而 DSP 36 完全可以为中断提供有效的服务。在任何情况下，在这些涉及 MIDI 的中断的整个处理的系统开销都是与 DSP 36 隔离的，并且与主处理机 12 和任何其它 DSP 32 之间都是屏蔽的。对 DSP 36 造成的数据处理负荷是可以选择的，因此，相关于 DSP RAM 34 从直接连接的 CODEC 接口 42 要被提供或是接收数据的任何并行处理都处在 DSP 36 的数据处理能力范围之内。

最后，由 DSP 36 接收和处理的数据可以通过 CODEC 接口 42 传送，用于进一步的处理，或是通过 FIFO 7 传送到系统主存储器 14，用于 BTU8 和 9



或是双方的相互控制。

按照本发明，参照图 a-c 所述的过程可以并且通常是相互独立的。每一种过程可以代表一个按照在主处理机 12 和数据源与联系着 I/O 通道控制器 22 的接收器之间的一条独立的控制线索传送的一个独立的数据流。然而，本  
5 发明的特点是允许这些和潜在的其它过程可以同时进行。另外，由于每个数据流是按请求驱动操作的，I/O 通道控制器 22 完全是在动态和连续的基础上分配的，这样就能确保每个数据流的基本数据传送速率能满足外围设备数据源或是接收器的要求。

#### IX. 总结

10 以上说明了一种多功能的 I/O 外围设备支持系统。这种外围设备系统基本上可以自动地操作多路径的数据流传送，并且特别适合连接到外围设备系统上的集成化和外部外围设备所要求的 I/O 带宽。另外，这种外围设备系统可以支持多个数字信号处理器实现存储器到存储器的数据传送，其能用来卸载处理，以及中断来自主处理机的集中型的作业，同时，可以尽量减少和直接支持主处理机在对数字信号处理器的管理方面的系统开销。  
15

根据在上文中公开的本发明的最佳实施例，本领域的技术人员很容易对所述的实施例进行各种修改和变更。因此需要指出，在附加的权利要求书的范围之内，本发明可以用上述具体说明以外的形式来体现。

# 说明书附图

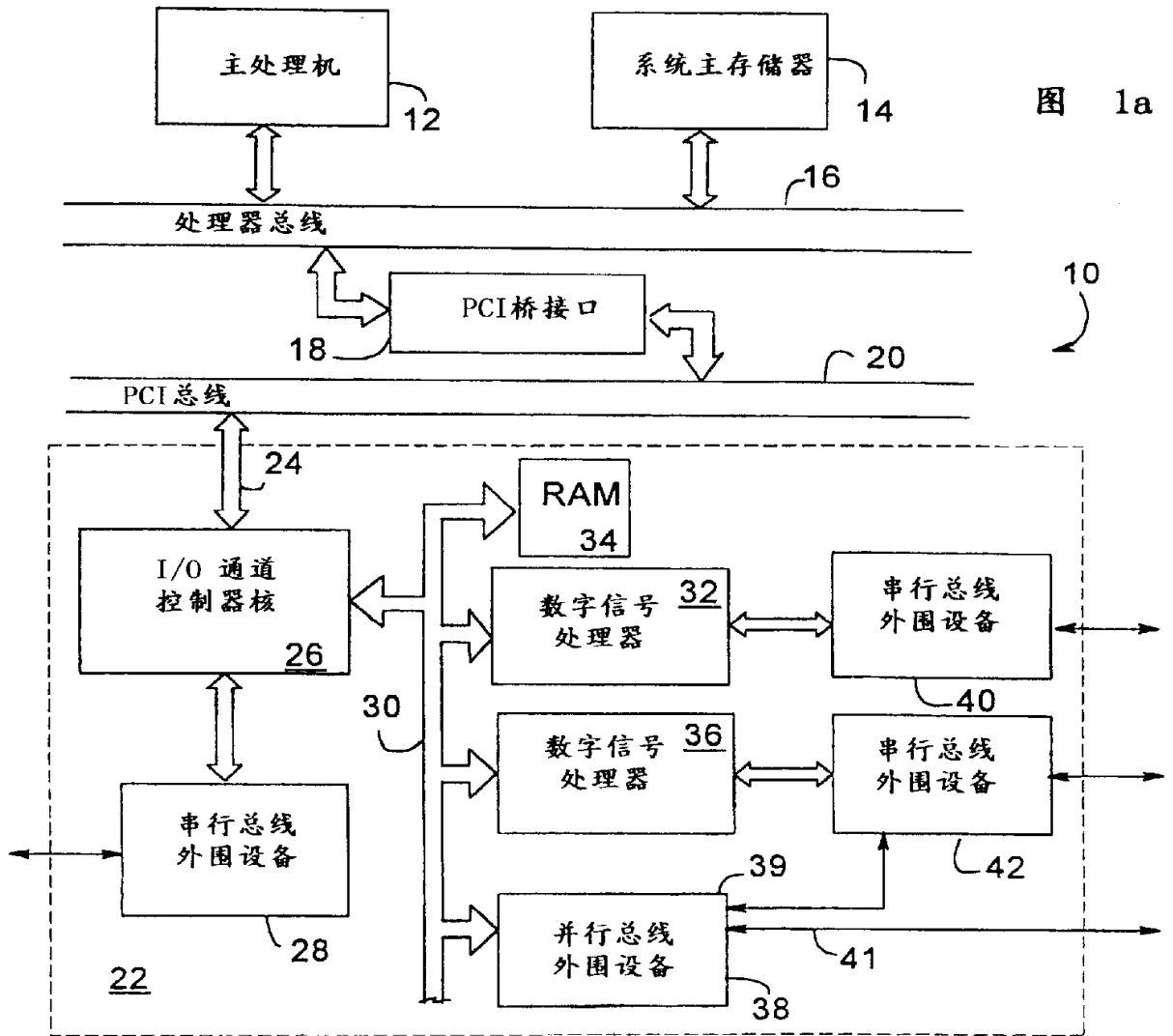


图 1a

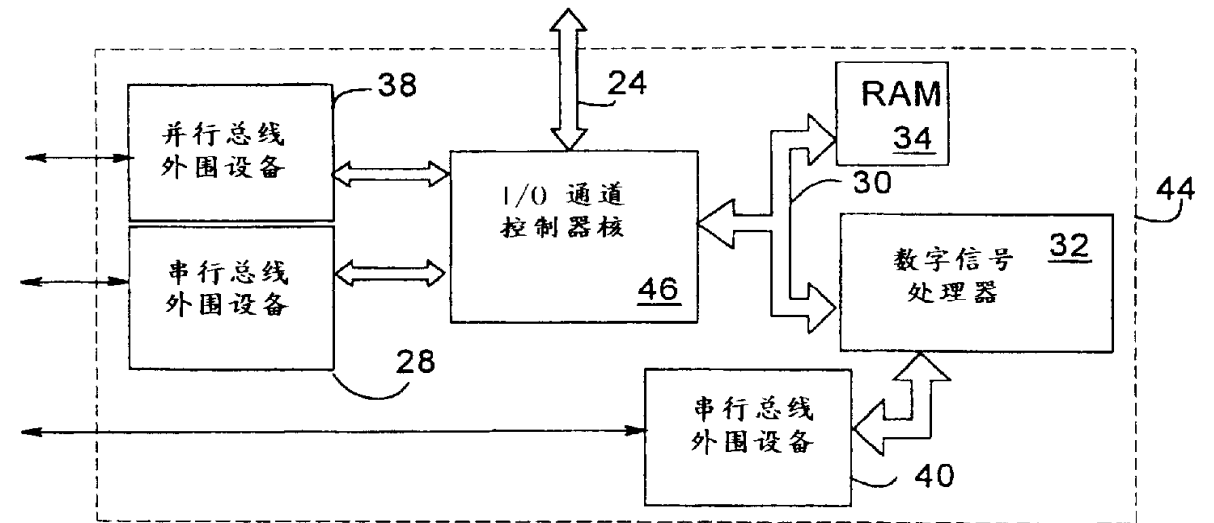


图 1b

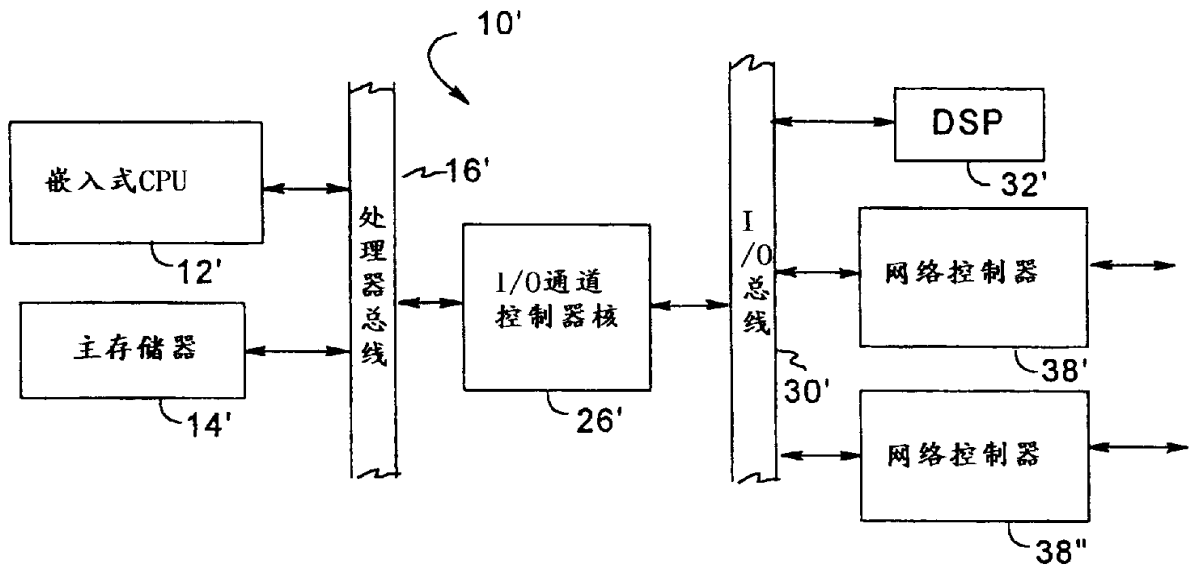


图 2

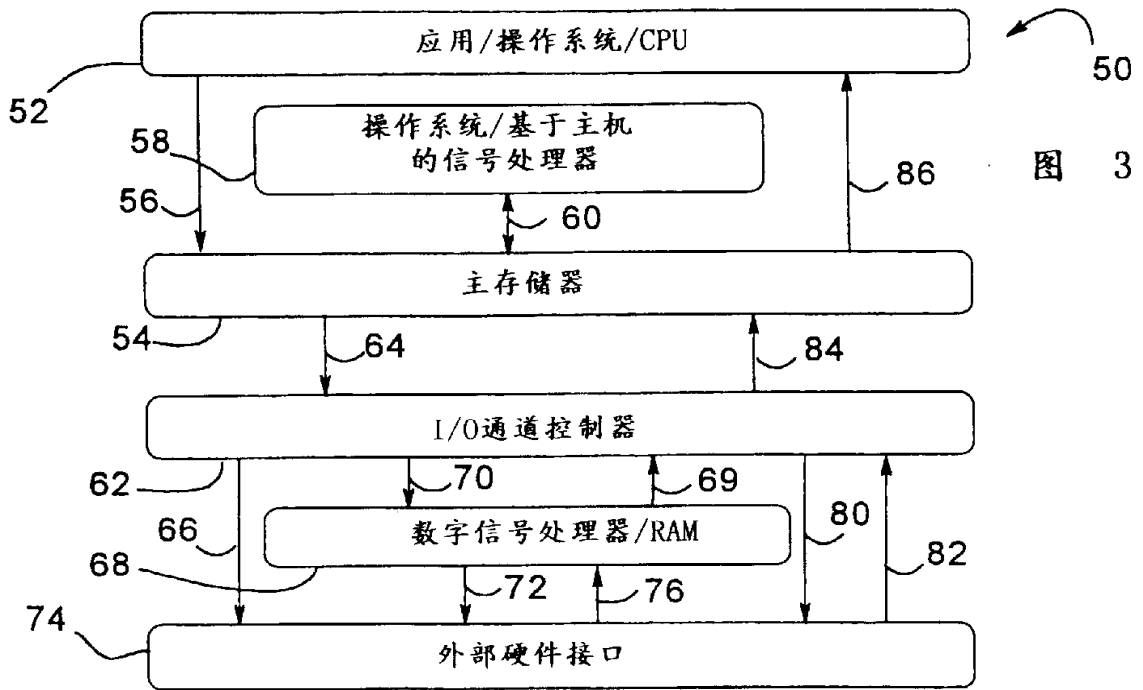


图 3

图 4

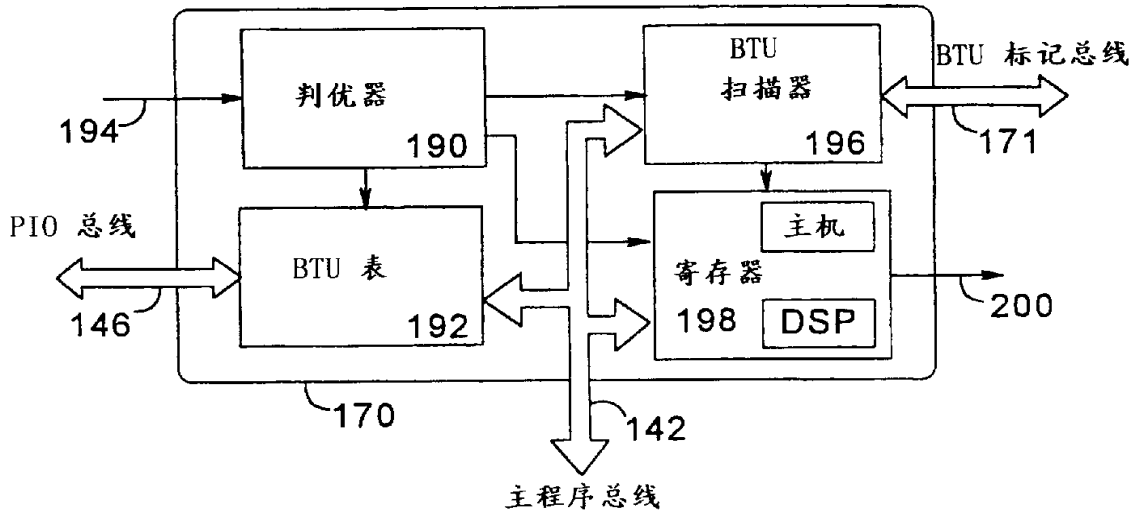
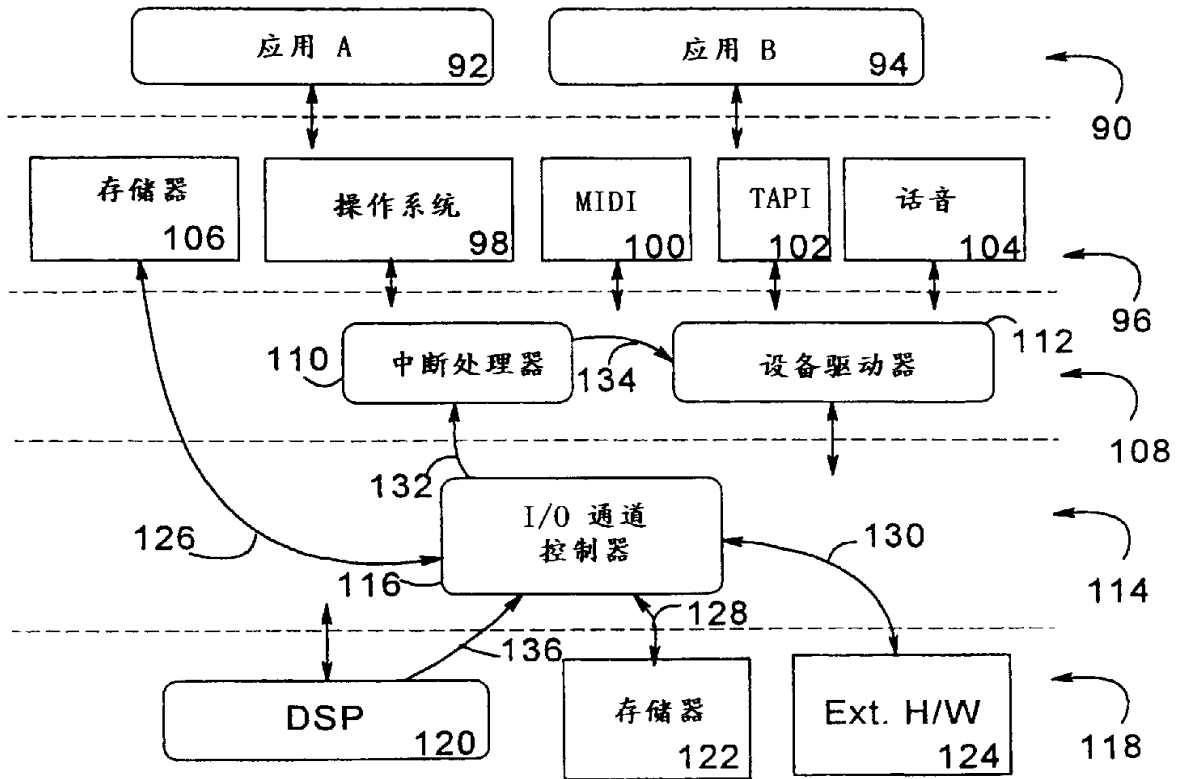


图 5b

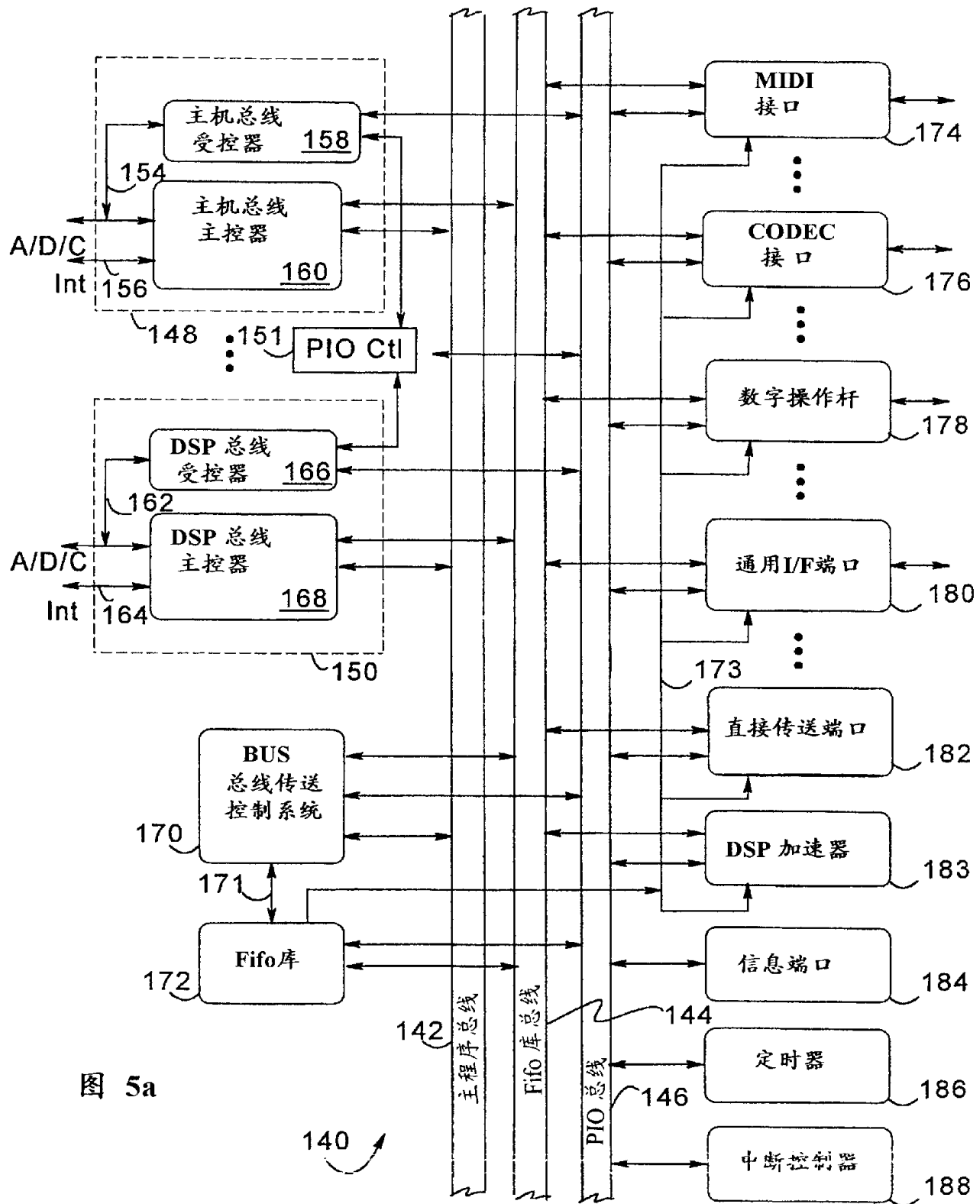


图 5a

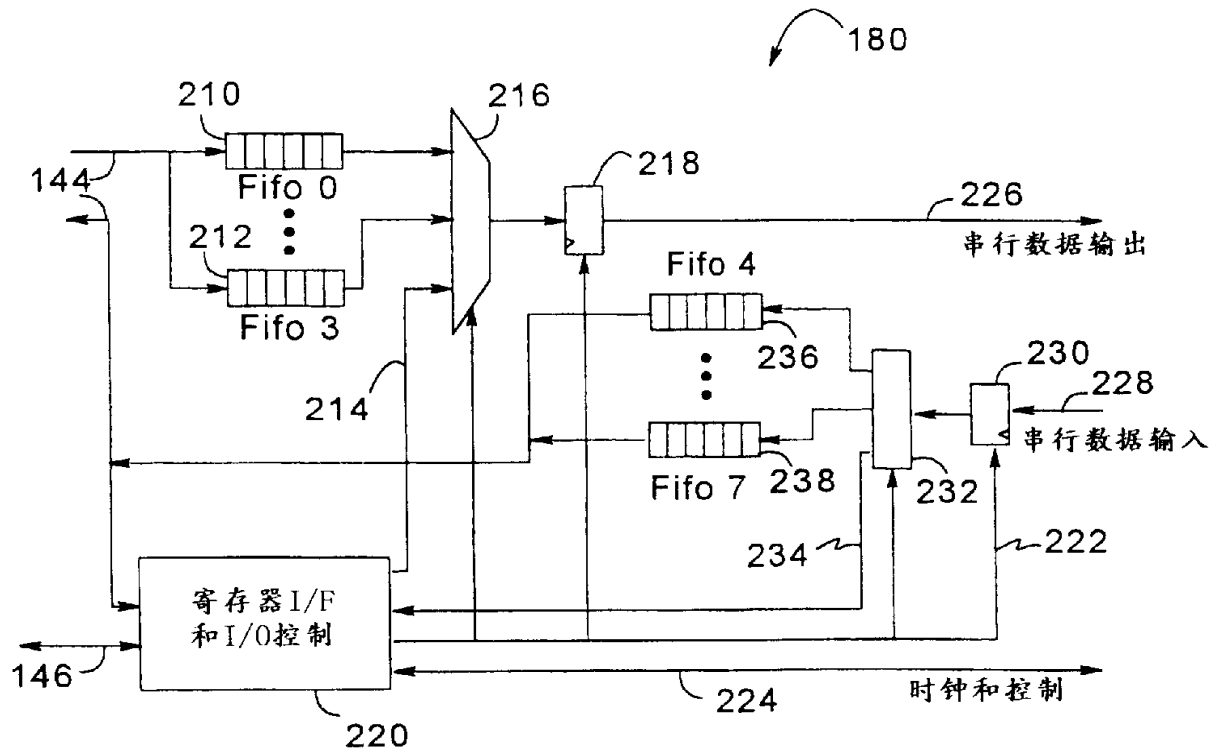


图 5c

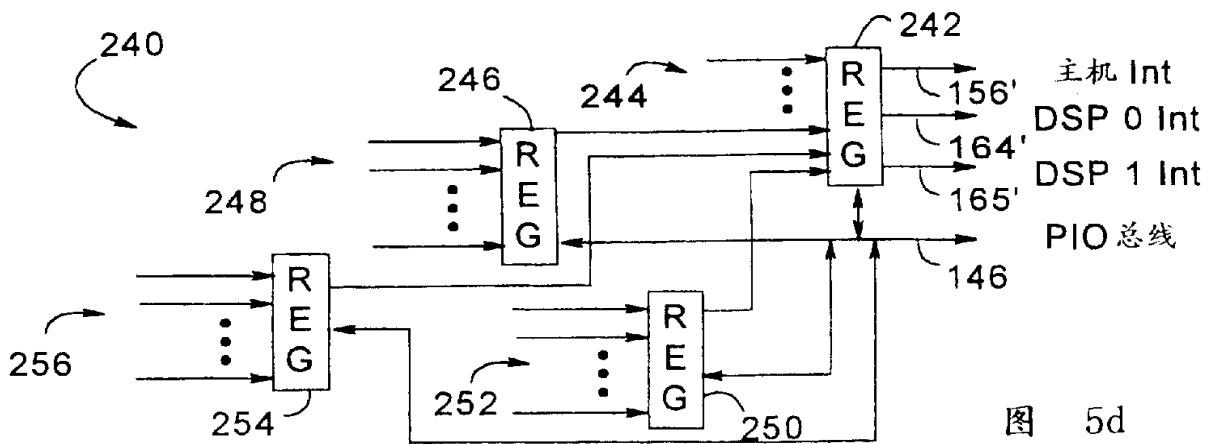


图 5d

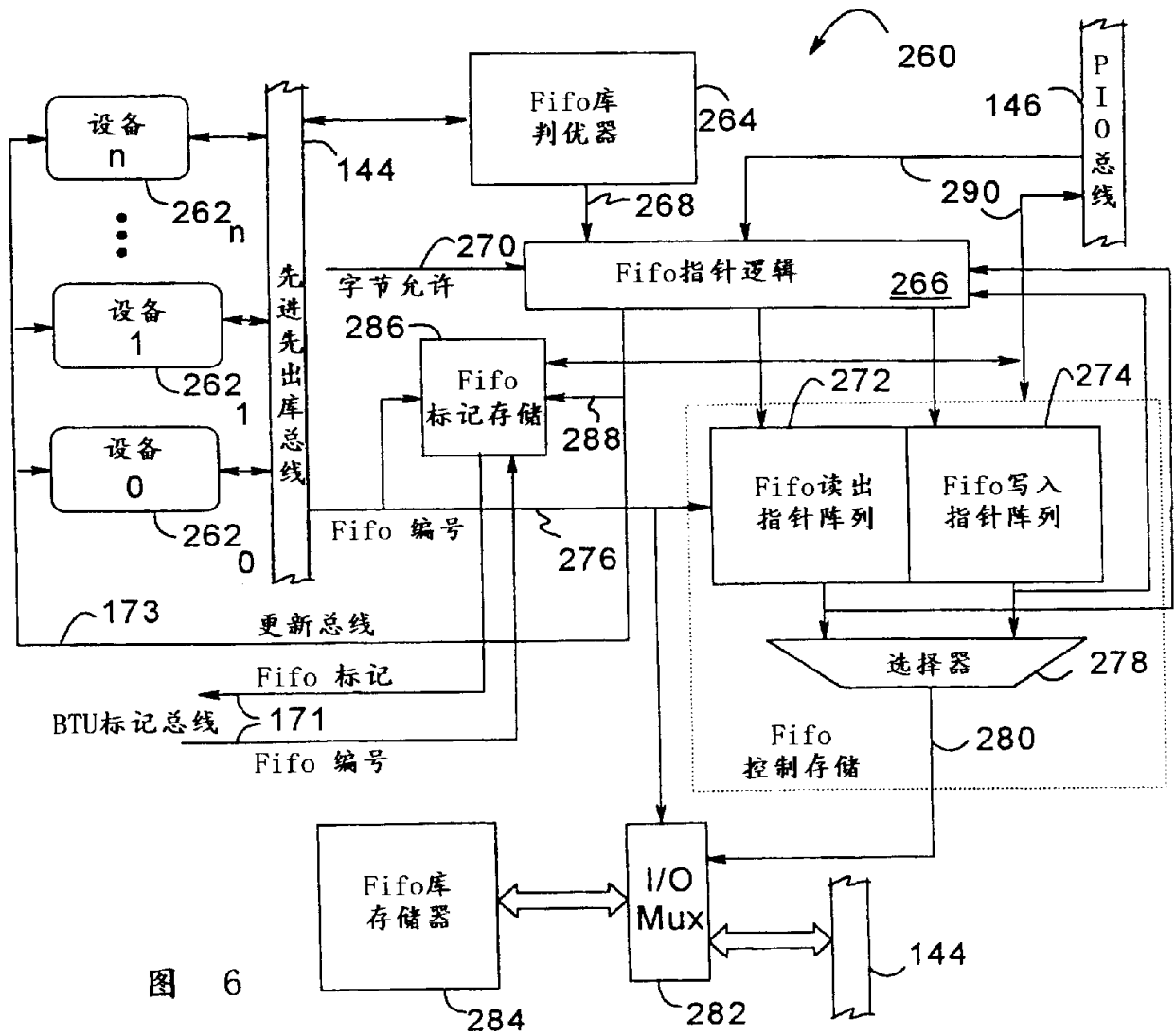


图 6

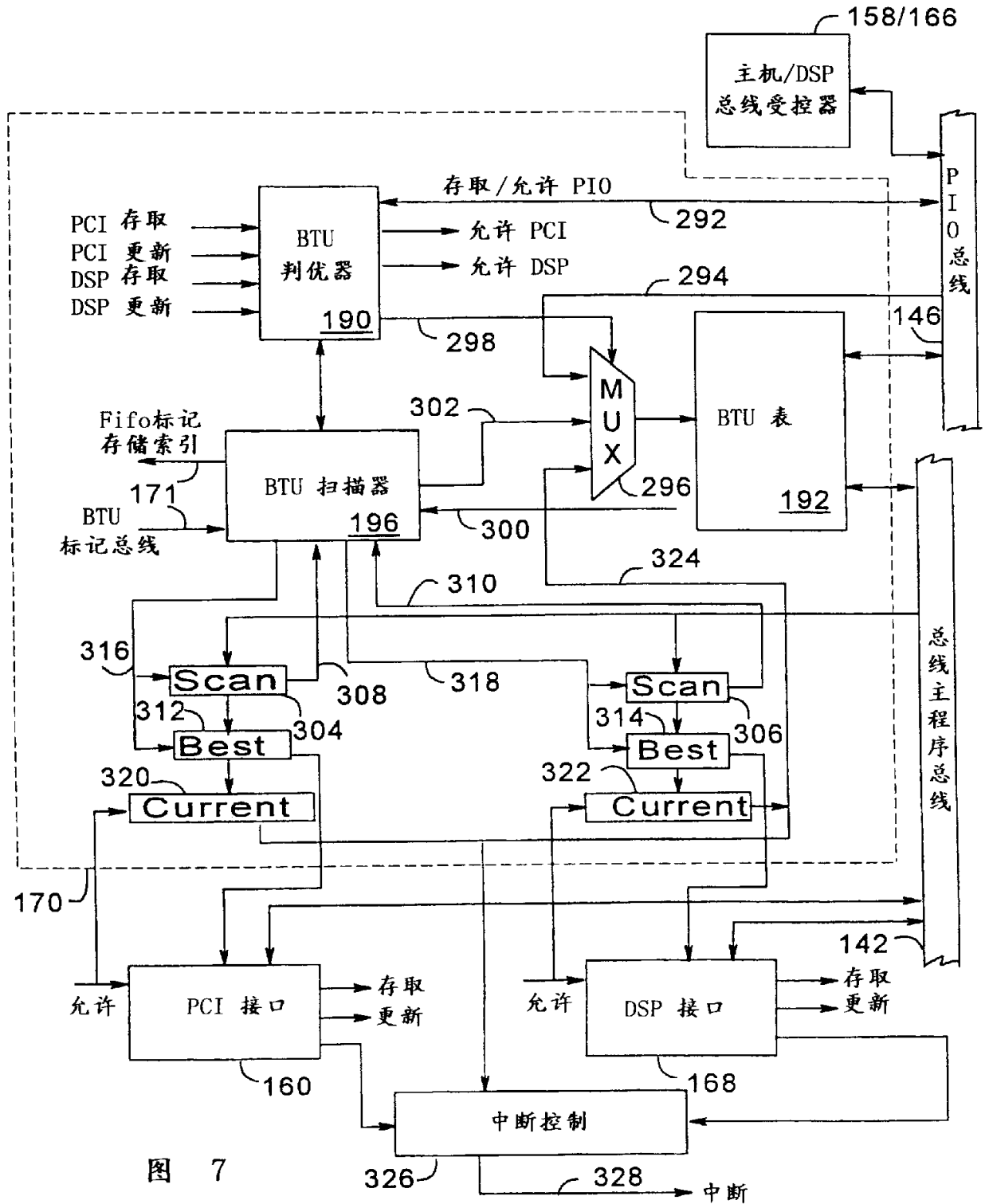


图 7



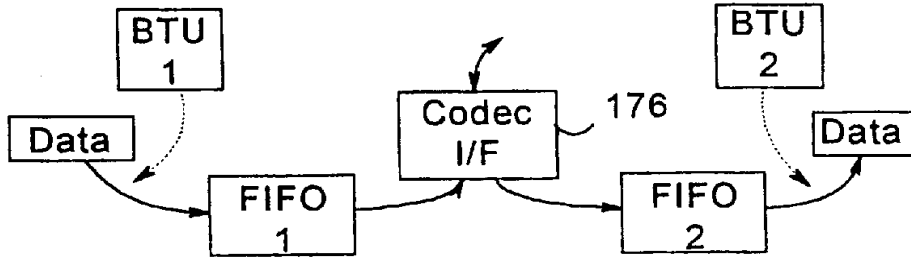


图 8a

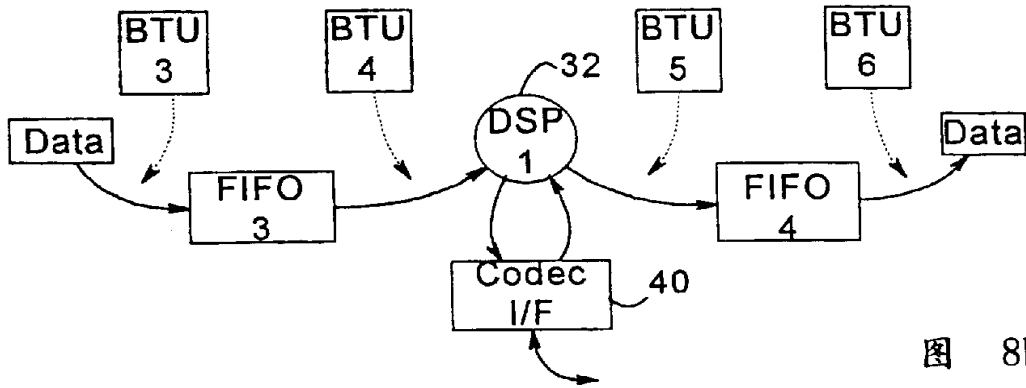


图 8b

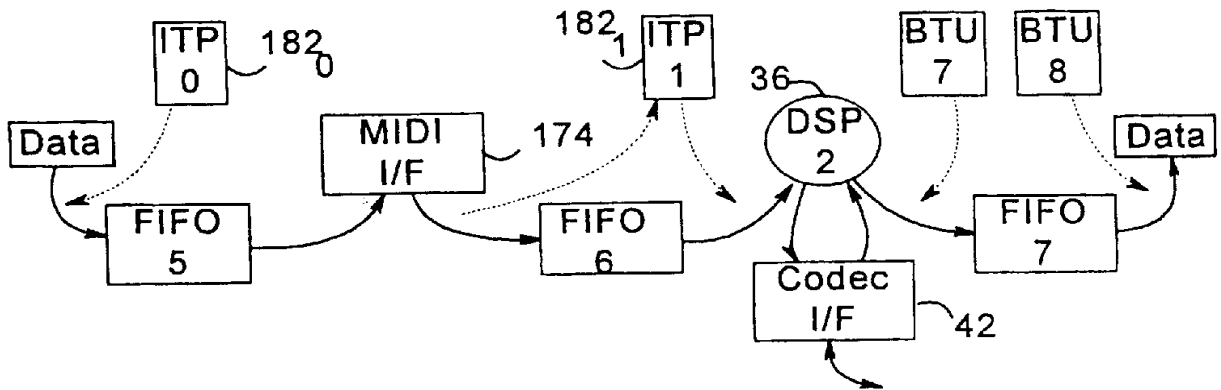


图 8c