



[12] 发明专利申请公布说明书

[21] 申请号 200680043319.6

[43] 公开日 2008年12月3日

[11] 公开号 CN 101317166A

[22] 申请日 2006.9.29

[21] 申请号 200680043319.6

[30] 优先权

[32] 2005.9.29 [33] US [31] 11/238,790

[86] 国际申请 PCT/US2006/038081 2006.9.29

[87] 国际公布 WO2007/041301 英 2007.4.12

[85] 进入国家阶段日期 2008.5.20

[71] 申请人 P. A. SEMI 公司

地址 美国加利福尼亚

[72] 发明人 D·戈 M·D·海特 陈宗建

R·沃德万 古为春

[74] 专利代理机构 中国国际贸易促进委员会专利
商标事务所
代理人 郭放

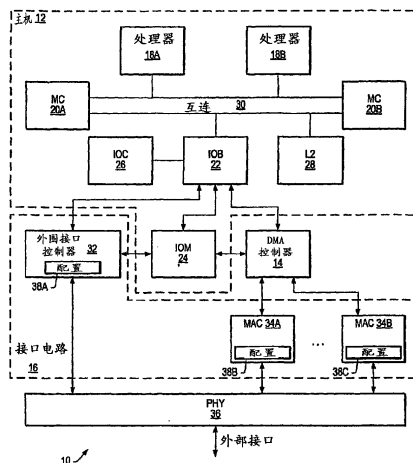
权利要求书4页 说明书38页 附图11页

[54] 发明名称

统一 DMA

[57] 摘要

在一种实施方案中，提供一种装置，包括：第一接口电路；被耦合到第一接口电路的直接存储器存取 (DMA) 控制器；以及被耦合到 DMA 控制器的主机。第一接口电路被配置为根据协议在接口上进行通信。主机包括被至少部分地映射到主机的存储器系统中的多个存储位置的至少一个地址空间。DMA 控制器被配置为在第一接口电路和地址空间之间执行 DMA 传输，并且 DMA 控制器还被配置为在多个存储位置的第一批和多个存储位置的第二批之间执行 DMA 传输。



1.一种装置，包括：

第一接口电路，被配置为根据协议在接口上进行通信；

直接存储器存取控制器、即 DMA 控制器，被耦合到第一接口电路；以及

主机，被耦合到 DMA 控制器，其中主机包括至少一个地址空间，该地址空间被至少部分地映射到主机的存储器系统中的多个存储位置，

其中，DMA 控制器被配置为在第一接口电路与地址空间之间执行 DMA 传输，并且 DMA 控制器还被配置为在所述多个存储位置中的第一批和所述多个存储位置中的第二批之间执行 DMA 传输。

2.如权利要求 1 中所述的装置，其中 DMA 控制器包括发送控制电路、接收控制电路、和被耦合到发送控制电路和接收控制电路的环回电路，其中发送控制电路被配置为从所述第一批中读取数据并将数据提供给环回电路；所述环回电路被配置为将数据提供给接收控制电路；所述接收控制电路被配置为将数据写入到第二批。

3.如权利要求 2 中所述的装置，其中接收控制电路还被配置为将数据从第一接口电路中写入到地址空间；DMA 控制器包括仲裁器，该仲裁器被配置为在环回电路和第一接口电路之间进行仲裁以将数据提供给接收控制电路。

4.如权利要求 1 到 3 中任一项所述的装置，其中第一接口电路包括介质访问控制器、即 MAC。

5.如权利要求 4 中所述的装置，还包括具有第一接口电路的多个接口电路，其中所述多个接口电路被耦合到 DMA 控制器。

6.如权利要求 1 到 3 中任一项所述的装置，其中第一接口电路包括外围接口控制器电路。

7.如权利要求 6 中所述的装置，还包括具有第一接口电路的多个接口电路，其中所述多个接口电路被耦合到 DMA 控制器。

8.权利要求 6 或 7 中所述装置，其中所述多个接口电路中的第二接口电路包括介质访问控制器、即 MAC。

9.如权利要求 1 到 8 中任一项所述的装置，其中直接存储器存取控制器、即 DMA 控制器被配置为执行从地址空间到目标的 DMA 传输，

其中，所述目标根据 DMA 传输被分配给哪条通道而是接口电路或主机；DMA 控制器被配置为对从地址空间中为 DMA 传输而读取的第一 DMA 数据至少执行第一操作，该第一操作产生结果。

10.如权利要求 9 中所述的装置，其中第一操作将第一 DMA 数据转换成第二 DMA 数据；DMA 控制器被配置为将第二 DMA 数据传输到目标。

11.如权利要求 9 或 10 中所述的装置，其中第一操作包括加密。

12.如权利要求 11 中所述的装置，其中 DMA 控制器还被配置为对第一 DMA 数据应用散列函数，并且其中所述结果包括散列函数的输出。

13.如权利要求 9 或 10 中所述的装置，其中所述结果被存储在与第一 DMA 数据相对应的 DMA 描述符数据结构中。

14.如权利要求 13 中所述的装置，其中第一操作是散列函数。

15.如权利要求 13 中所述的装置，其中第一操作是循环冗余校验码生成。

16.如权利要求 13 中所述的装置，其中第一操作是校验和生成。

17.如权利要求 9 中所述的装置，其中 DMA 控制器被配置为从多个通道中读取数据并对数据进行 XOR 运算以产生所述结果。

18.一种方法，包括以下步骤：

直接存储器存取控制器、即 DMA 控制器在第一接口电路和主机中的地址空间之间执行第一 DMA 传输，其中地址空间被至少部分地映射到主机中的多个存储位置，第一接口电路被配置为根据协议在接口上进行通信；以及

DMA 控制器在所述多个存储位置中的第一批和所述多个存储位

置中的第二批之间执行 DMA 传输。

19.如权利要求 18 中所述的方法，其中第一接口电路包括介质访问控制器、即 MAC。

20.如权利要求 18 或 19 中所述的方法，其中 DMA 控制器被耦合到包括第一接口电路的多个接口电路，所述多个接口电路被耦合到 DMA 控制器。

21.如权利要求 18 到 20 中任一项所述的方法，其中所述多个接口电路中的第二接口电路包括外围接口控制器电路。

22.一种直接存储器存取控制器、即 DMA 控制器，包括：

发送控制电路，被配置为从主机中的地址空间中读取第一 DMA 数据，并根据 DMA 传输被分配给哪条通道而将该第一 DMA 数据提供给 DMA 控制器中的卸载引擎或者提供给接口电路；以及

卸载引擎，被耦合以从发送控制电路中接收第一 DMA 数据，其中卸载引擎被配置为对第一 DMA 数据至少执行第一操作以产生结果，卸载引擎被配置为在将第一 DMA 数据提供给卸载引擎的 DMA 传输期间至少开始执行所述第一操作；以及

接收控制电路，被耦合到卸载引擎以接收所述结果，其中接收控制电路被配置为根据描述 DMA 传输的 DMA 描述符数据结构，将结果写入到主机中的地址空间；接收控制电路还被配置为将从接口电路接收的数据写入到所述地址空间。

23.如权利要求 22 中所述的 DMA 控制器，其中第一操作将第一 DMA 数据转换成第二 DMA 数据；卸载引擎被耦合到接收控制电路并被配置为将第二 DMA 数据提供给接收控制电路。

24.如权利要求 22 或 23 中所述的 DMA 控制器，其中第一操作包括加密。

25.如权利要求 22 到 24 中任一项所述的 DMA 控制器，其中卸载引擎还被配置为对第一 DMA 数据应用散列函数；所述结果包含散列函数的输出。

26.如权利要求 22 到 25 中任一项所述的 DMA 控制器，其中所

述结果被存储在与第一DMA数据相对应的DMA描述符数据结构中。

27.如权利要求26中所述的DMA控制器,其中第一操作是散列函数。

28.如权利要求26中所述的DMA控制器,其中第一操作是循环冗余校验码生成。

29.如权利要求26中所述的DMA控制器,其中第一操作是校验和生成。

30.如权利要求22到29中任一项所述的DMA控制器,还包括异或电路、即XOR电路,其中发送控制电路被配置为从多个通道中读取数据并将数据提供给XOR电路,所述XOR电路被配置为对数据进行XOR运算以产生结果。

31.一种直接存储器存取控制器、即DMA控制器,该DMA控制器被配置为从主机中的地址空间中读取第一DMA数据,并根据DMA传输被分配给哪条通道来将第一DMA数据提供给DMA控制器中的卸载引擎或者提供给接口电路;卸载引擎被配置为对第一DMA数据至少执行第一操作以产生结果,并且卸载引擎被配置为在将第一DMA数据提供给卸载引擎的DMA传输期间至少开始执行所述第一操作;DMA控制器被配置为根据描述DMA传输的DMA描述符,将结果写入到主机中的地址空间,并且DMA控制器还被配置为将从接口电路接收的数据写入到地址空间。

统一 DMA

技术领域

本发明涉及集成电路领域，更确切地说，涉及包括一个或多个集成电路的系统中的直接存储器存取（DMA）。

背景技术

在包括一个或多个处理器、存储器、和输入/输出（I/O）设备或接口的典型系统中，直接存储器存取（DMA）传输经常被用于在 I/O 和存储器之间传输数据。在一些系统中，每个使用 DMA 的 I/O 设备或接口中都包括单独的 DMA 电路。在其它系统中，一个或多个 I/O 设备可以共享 DMA 电路。

一些系统还包括用于将数据从存储器的一个区域拷贝到另一个区域的“数据移动器”。数据移动器可以卸载处理器的负担，否则处理器将不得不执行指令以实现数据移动（例如以处理器所用的位宽来读/写数据，一般一次 32 位或 64 位）。数据移动器的编程模型一般不同于 DMA 编程模型，后者适合在 I/O 设备和存储器之间通信。

发明内容

在一种实施方案中，提供一种装置，包括：第一接口电路；被耦合到第一接口电路的直接存储器存取（DMA）控制器；以及被耦合到 DMA 控制器的主机。第一接口电路被配置为根据协议在接口上进行通信。主机包含被至少部分地映射到主机的存储器系统中的多个存储位置上的至少一个地址空间。DMA 控制器被配置为在第一接口电路和地址空间之间执行 DMA 传输，并且 DMA 控制器还被配置为在多个存储位置的第一批和多个存储位置的第二批之间执行 DMA 传输。还提出了一种方法。

附图说明

下列详细描述参考了附图，这些附图的简要描述如下。

图 1 为系统的一种实施方案的框图。

图 2 为图 1 所示 DMA 控制器的一种实施方案的框图。

图 3 为图 2 所示卸载引擎 (offload engine) 的一种实施方案的框图。

图 4 为图 1 系统中的 DMA 的一种实施方案的框图。

图 5 为描述符环 (descriptor ring) 和缓冲器指针环 (buffer pointer ring) 的一种实施方案的框图。

图 6 为流程图，显示了图 2 所示的接收预取引擎 (receive prefetch engine) 的一种实施方案的操作。

图 7 为流程图，显示了图 2 所示的接收控制电路的一种实施方案的操作。

图 8 为流程图，显示了图 2 所示的发送控制电路的一种实施方案的预取操作。

图 9 为流程图，显示了图 2 所示的发送控制电路的一种实施方案的传输操作。

图 10 为框图，显示了描述符环，具有包含在传输描述符中的控制描述符。

图 11 为流程图，显示了控制描述符的处理的一种实施方案。

图 12 为框图，显示了接收 DMA 描述符的一种实施方案。

图 13 为框图，显示了发送 DMA 描述符的一种实施方案。

图 14 为框图，显示了拷贝 DMA 描述符的一种实施方案。

图 15 为卸载 DMA 描述符的一种实施方案的框图。

图 16 为控制描述符的一种实施方案的框图。

图 17 为图 3 所示的校验和生成器的一种实施方案。

图 18 为图 17 所示的全加器的一种实施方案。

具体实施方式

虽然本发明允许有各种修改和替代形式,但是具体的实施方案将通过附图中的举例来显示并且在本文中详细描述。然而,应理解的是,附图和详细描述并非意图将本发明限制于所示的具体形式,而恰恰相反,本发明涵盖由所附权利要求所限定的本发明精神和范围内的所有修改、等价物和替代物。

现在转到图 1,显示了系统 10 的一种实施方案的框图。在所示实施方案中,系统 10 包括主机 12、DMA 控制器 14、接口电路 16、和物理接口层 (PHY) 36。DMA 控制器 14 被耦合到主机 12 和接口电路 16。接口电路 16 还被耦合到物理接口层 36。在所示实施方案中,主机 12 包括:一个或多个处理器,诸如处理器 18A~18B;一个或多个存储器控制器,诸如存储器控制器 20A-20B; I/O 桥接器 (IOB) 22; I/O 存储器 (IOM) 24; I/O 高速缓冲存储器 (高速缓存) (IOC) 26; 二级 (L2) 高速缓存 28; 和互连 30。处理器 18A~18B、存储器控制器 20A-20B、IOB 22、和 L2 高速缓存 28 被耦合到互连 30 上。IOB 22 还被耦合到 IOC 26 和 IOM 24 上。DMA 控制器 14 也被耦合到 IOB 22 和 IOM 24 上。在所示实施方案中,接口电路 16 包括:外围接口控制器 32; 以及一个或多个介质访问控制电路 (MAC), 诸如 MAC 34A~34B。MAC 34A~34B 被耦合到 DMA 控制器 14 上并被耦合到物理接口层 36 上。外围接口控制器 32 还被耦合到 I/O 桥接器 22 和 I/O 存储器 34 上 (并因而被间接耦合到 DMA 控制器 14 上) 以及被耦合到物理接口层 36 上。外围接口控制器 32 和 MAC 34A-34C 的每个都包括配置寄存器 38A-38C。在一些实施方案中,系统 10 的部件可以被集成到单个集成电路上作为片上系统。在其它实施方案中,系统 10 可以被实现为两个或更多个集成电路。

主机 12 可以包含一个或更多的地址空间。主机 12 中的地址空间的至少一部分可以被映射到主机 12 中的存储位置。即,主机 12 可以包括被映射到主机地址空间中的地址上的存储器系统。例如,存储器控制器 20A-20B 的每个都可以被耦合到包含被映射到地址空间中

的存储位置的存储器（未显示）。在一些情况下，整个地址空间可以被映射到存储位置上。在其它情况下，地址空间中的一些可以是存储器映射 I/O（memory-mapped I/O）（例如，受外围接口控制器 32 控制的外围接口可以包括一些存储器映射 I/O）。

DMA 控制器 14 被配置为在接口电路 16 和主机地址空间之间执行 DMA 传输。特别地，DMA 传输可以在地址空间所被映射到的存储位置和接口电路 16 之间进行。另外，在一些实施方案中，DMA 控制器 14 可以被配置为在地址空间内的存储位置组之间执行 DMA 传输。即，这种 DMA 传输的来源和目的两者均可以为存储位置。在一些实施方案中，数据移动器的功能可以因此而被并入到 DMA 控制器 14 中，并且可以不需要单独的数据移动器。用于存储器到存储器 DMA 传输的编程模型可以类似于用于其它 DMA 传输的编程模型（例如，DMA 描述符，以下将详细描述）。存储器到存储器 DMA 传输也可以被称为拷贝 DMA 传输。

在一些实施方案中，DMA 控制器 14 可以被配置为在 DMA 数据正在被传输时对 DMA 数据执行一个或更多的操作（或“功能”）。在一些实施方案中，可以对在地址空间和接口电路之间的传输执行操作，以及可以对拷贝 DMA 传输执行操作。在一些实施方案中，由 DMA 控制器 14 所执行的操作可以减少处理器 18A ~ 18B 上的处理负荷，因为处理器不需要执行 DMA 控制器 14 所执行的操作。在一种实施方案中，DMA 控制器 14 所执行操作中的一些是对分组数据的操作（例如，加密/解密，循环冗余校验码（CRC）生成或校验，校验和生成或校验等）。操作还可以包括异或（XOR）操作，这种异或操作例如可以被用于廉价磁盘冗余阵列（RAID）处理。

总而言之，DMA 传输可以是从来源到目的的数据传输，其中目的中的至少一个是存储位置或主机地址空间中的其它地址。DMA 传输的完成无需让被传输数据经由系统中的处理器（例如处理器 18A ~ 18B）。DMA 控制器 14 可以通过对来源进行读取并对目的进行写入来完成 DMA 传输。例如，从存储器到接口电路 16 的 DMA 传输可以

这样完成: DMA 控制器 14 产生存储器读取请求(在所示实施方案中,对在互连 30 上执行连贯读取任务来读数据的 IOB 22)并将读取数据作为 DMA 数据发送到接口电路 16。在一种实施方案中, DMA 控制器 14 可以为经由外围接口控制器 32 的 DMA 传输产生读取请求以将数据读入 IOM 24, 并且外围接口控制器 32 可以从 IOM 24 读取数据并发送该数据。可以通过 DMA 控制器 14 从接口电路 16 接收数据并产生存储器写入请求(对 IOB 22, 在所示实施方案中)以将 DMA 数据传输到存储器中, 来完成从接口电路 16 到存储器的 DMA 传输。在一种实施方案中, 外围接口控制器 32 可以将数据写入到 IOM 24, 并且 DMA 控制器 14 可以使数据被写入到存储器中。这样, DMA 控制器 14 可以为外围接口控制器 32 提供 DMA 协助。可以通过产生到来源存储位置的存储器读取请求和到目的存储位置(包括来自存储器读取请求的 DMA 数据)的存储器写入请求, 来完成拷贝 DMA 传输。

主机 12 可以大体上包括一个或更多的处理器和被配置为与映射在主机 12 地址空间内的存储器接口的存储器控制器。主机 12 可以非必需地包括其它电路, 诸如 L2 高速缓存 28, 以增强主机 12 中处理器的性能。此外, 主机 12 可以包括作为与各种 I/O 电路和 DMA 控制器 14 接口的电路。虽然主机 12 的一种实现如图 1 中所示, 但是其它实施方案可以包括对 DMA 控制器 14 和接口电路 16 的任何配置和接口。

处理器 18A ~ 18B 包括执行在由处理器 18A ~ 18B 实现的指令集体系结构中所定义的指令的电路。可以在各种实施方案中实现任意的指令集体系结构。例如, 可以实现 PowerPC™ 指令集体系结构。其它示例性指令集体系结构可以包括 ARM™ 指令集、MIPS™ 指令集、SPARC™ 指令集、x86 指令集(也被称为 IA-32)、IA-64 指令集等。

存储器控制器 20A-20B 包含被配置为到存储器的接口的电路。例如, 存储器控制器 20A-20B 可以被配置为与诸如同步 DRAM (SDRAM)、双倍数据速率(DDR)SDRAM、DDR2 SDRAM、Rambus DRAM (RDRAM) 等的动态随机存取存储器 (DRAM) 接口。存储

器控制器 20A-20B 可以为它们从互连 30 所耦合到的存储器接收读取和写入任务，并且可以执行对存储器的读取/写入操作。读取和写入任务可以包括由 IOB 22 代表 DMA 控制器 14 和/或外围接口控制器 32 发起的读取和写入任务。另外，读取和写入任务可以包括由处理器 18A ~ 18B 和/或 L2 高速缓存 28 所产生的任务。

为了由处理器 18A ~ 18B 和/或互连 30 上的其它主体 (agent) 低延迟地访问，L2 高速缓存 28 可以包括被配置为高速缓存与存储器控制器 20A-20B 所耦合到的存储器中多个存储位置对应的副本的高速缓存存储器 (cache memory)。L2 高速缓存 28 可以包括任意容量和构造 (例如直接映射、组关联 (set associative) 等)。

IOB 22 包括被配置为在互连 30 上代表 DMA 控制器 14 和外围接口控制器 32 传达任务的电路。互连 30 可以支持高速缓存一致性 (cache coherency)，并且 IOB 22 可以参与一致性并确保由 IOB 22 所发起的任务的一致性。在所示实施方案中，IOB 22 采用 IOC 26 对新近由 IOB 22 所发起的任务进行高速缓存。在各种实施方案中，IOC 26 可以具有任意的容量和配置，并且可以是连贯的 (coherent)。IOC 26 可以被用于，例如，对由于由 DMA 控制器 14 和外围接口控制器 32 所产生的读取/写入而仅被部分更新的数据块进行高速缓存。在某些情况下，使用 IOC 26 可以避免在互连 30 上发生读取 - 修改 - 写入 (read-modify-write) 序列。另外，如果 IOC 26 对高速缓存块具有足够的所有权来完成读取/写入，那么对于由 DMA 控制器 14 或外围接口控制器 32 所产生的读取/写入，互连 30 上的任务可以因 IOC 26 中的高速缓存命中而被避免。其它实施方案可以不包括 IOC 26。

IOM 24 可以被用作在 IOB 22 与外围接口 32 或 DMA 控制器 14 之间传输的数据的分级缓冲器 (staging buffer)。这样，IOB 22 和 DMA 控制器 14/外围接口控制器 32 之间的数据路径可以经由 IOM 24。控制通道 (包括读取/写入请求、与请求相关的主机地址空间中的地址等) 可以直接位于 IOB 22 和 DMA 控制器 14/外围接口控制器 32 之间。其它实施方案可以不包括 IOM 24。

互连 30 可以包括任意用于在处理器 18A ~ 18B、存储器控制器 20A-20B、L2 高速缓存 28、以及 IOB 22 相互之间通信的通信介质。例如，互连 30 可以为支持一致性的总线。替代地，互连 30 可以为上述主体之间的点对点互连、基于分组的互连、或者任意其它互连。

接口电路 16 一般包括被配置为根据任意接口协议在对系统 10 的接口上通信、以及与系统 10 中的其它组件通信以接收在接口上被发送的通信或提供从接口接收的通信的电路。接口电路可以被配置为将源自系统 10 的通信转换为接口协议，以及为了系统 10 中的发送将从接口接收的通信进行转换。例如，接口电路 16 可以包括被配置为根据外围接口协议（例如，外围接口控制器 32）进行通信的电路。作为另一例子，接口电路 16 可以包括被配置为根据网络接口协议（例如 MAC 34A ~ 34B）进行通信的电路。

MAC 34A ~ 34B 可以包括实现为了网络接口而定义的介质访问控制电路功能的电路。例如，MAC 34A ~ 34B 中的一个或多个可以实现千兆位以太网标准（Gigabit Ethernet Standard）。MAC 34A ~ 34B 中的一个或多个可以实现万兆位以太网附加单元接口（10 Gigabit Ethernet Attachment Unit Interface, XAUI）标准。其它实施方案可以实现其它以太网标准，诸如 10 兆位以太网或者 100 兆位以太网标准，或者任何其它的网络标准。在一种实现中，具有 6 个 MAC，其中 4 个是千兆以太网 MAC，2 个是 XAUI MAC。其它实施方案可以具有更多或更少的 MAC，以及任意 MAC 类型的混合。

除其它动作之外，实现以太网标准的 MAC 34A ~ 34B 可以从所接收的分组中去掉帧间间隔（inter-frame gap, IFG）、前导码（preamble）、和帧起始分隔符（start of frame delimiter, SFD），并且可以对于到存储器的 DMA 将剩余分组数据提供给 DMA 控制器 14。MAC 34A-34D 可以被配置为接收自 DMA 控制器 14 的分组插入 IFG、前导码、和 SFD 作为发送 DMA 传输（transmit DMA transfer），并且可以将分组发送给 PHY 36 用于发送。

外围接口控制器 32 包括被配置为对外围接口进行控制的电路。

在一种实施方案中,外围接口控制器 32 可以控制外围部件互连(PCI) Express 接口。除了 PCI Express 接口之外或者代替 PCI Express 接口,其它实施方案可以实现其它外围接口(例如,PCI、PCI-X、通用串行总线(USB)等)。

PHY 36 可以大体上包括被配置为在到系统 10 的外部接口上受接口电路 16 的控制进行物理通信的电路。在一种具体实施方案中,PHY 36 可以包括可以被构造以用作 PCI Express 通路(lane)或作为以太网连接的一组串行器/解串器(serializer/deserializer; SERDES)电路。PHY 36 可以包括为经由 SERDES 和同步先进先出(FIFO)缓冲器的传送执行 8b/10b 编码/解码的电路,并且还可以包括逻辑地配置 SERDES 链路以用作 PCI Express 或以太网通信链路的电路。在一种实施方案中,PHY 可以包括 24 个能被配置为 PCI Express 通路或以太网连接的 SERDES。任意所需数目的 SERDES 可以被配置为 PCI Express,并且任意所需数目的 SERDES 可以被配置为以太网连接。

在所示实施方案中,在外围接口控制器 32 和 MAC 34A~34B 中显示了配置寄存器 38A-38C。在每个外围接口控制器 32 和 MAC 34A~34B 中可以有一个或更多的配置寄存器。其它配置寄存器也可以存在于系统 10 中,图 1 中未显示。配置寄存器可以被用于配置外围接口控制器 32 和 MAC 34A~34B 的各种可编程选择的功能、使能或禁止各种功能、为操作配置外围接口控制器 32 和 MAC 34A~34B 等。在下述的一种实施方案中,可以在控制描述符中指定配置寄存器以用于外围接口控制器 32 和 MAC 34A~34B 的在线及时重构(on-the-fly reconfiguration)。

要注意的是,在各种实施方案中,系统 10 可以包括一个或任意多个图 1 中所示的任意元件(例如,处理器、存储器控制器、高速缓存、I/O 桥接器、DMA 控制器、和/或接口电路等)。

现在参考图 2,显示了 DMA 控制器 14 的一种实施方案的框图。对于图 2 的实施方案,将讨论用于引起 DMA 传输的描述符软件模型。在一些实施方案中,除了描述符模型之外或者代替描述符模型,可以

支持基于寄存器的软件模型。在基于寄存器的模型中，每个 DMA 传输可以被编程到 DMA 控制器 14 中，DMA 控制器 14 可以执行 DMA 传输。在传输完成时，DMA 控制器 14 既可以中断处理器 18A ~ 18B 其中之一也可以提供可供软件查询判断 DMA 传输何时已结束的状态（例如，在 DMA 控制器 14 内的寄存器中）。

在描述符模型中，利用存储器中的描述符数据结构，软件可以建立多个要执行的 DMA 传输。一般来说，DMA 描述符可以包括存储器中的数据结构，该数据结构描述 DMA 传输。DMA 描述符中的信息例如可以指定 DMA 传输的来源和目标、传输的尺寸、以及传输的各种属性。在一些情况下，DMA 传输的来源和目标可以是隐式的（implicit）。在存储器中的数据结构中（例如，“描述符环”中）可以存储多个描述符，并且 DMA 控制器 14 可以用数据结构中第一个描述符的地址来编程。DMA 控制器 14 可以读取描述符并执行所指示的 DMA 传输。各种控制机制可以被用于在软件和硬件之间控制描述符的所有权。例如，描述符可以包括对 DMA 控制器 14 表明描述符中所描述的 DMA 传输预备被执行的有效位或使能位。描述符中的中断位可以被用于表明 DMA 控制器 14 要在给定 DMA 传输结束时中断处理器 18A ~ 18B，或者传输结束位可以被用于表明描述符描述了最后的 DMA 传输以及 DMA 控制器 14 应被中止。替代地，DMA 控制器 14 可以实现可以被软件递加计数以表明有多少描述符可供 DMA 控制器 14 使用的描述符计数寄存器。DMA 控制器 14 可以将描述符计数寄存器递减计数以表明已发生了描述符的预取。在其它实施方案中，DMA 控制器 14 可以将描述符计数寄存器递减计数以表明描述符的消耗（consumption）（即，具体的 DMA 传输的性能）。在另一些其它实施方案中，DMA 控制器 14 可以使用单独的描述符处理计数寄存器以表明有多少描述符已被处理或预取。

DMA 控制器 14 可以执行发送（Tx）DMA 传输和接收（Rx）DMA 传输。Tx DMA 传输以主机 12 中的地址空间作为来源（例如，耦合到存储器控制器 20A-20B 的存储器中的存储位置）。Rx DMA 传

输以主机 12 中的地址空间作为目标。Tx DMA 传输可以以接口电路 16 作为目标,或者可以以主机 12 地址空间中的其它地址作为目标(例如,对于拷贝 DMA 传输)。以主机地址空间为目标的 Tx DMA 传输可以使用 Rx DMA 数据路径来将从来源地址读取的数据写入到目标地址。环回电路 (loopback circuit) 40 可以提供 Tx DMA 数据路径和 Rx DMA 数据路径之间的链路。即,“环回电路”包括属于 DMA 控制器的电路,其被耦合以从发送数据路径接收 Tx DMA 数据并在接收 DMA 数据路径上提供 Rx DMA 数据。由环回电路 40 在接收 DMA 数据路径上所提供的数据可以是来自发送 DMA 数据路径所接收的数据(例如,用于拷贝 DMA 功能)。在一些实施方案中,由环回电路 40 所提供的数据可以由环回电路 40 从所接收数据转换的数据。在一些实施方案中,由环回电路 40 所提供的数据可以由环回电路 40 所接收的数据加上环回电路 40 对数据计算的结果(例如,校验和、CRC 数据等)。替代地,由环回电路 40 所提供的数据可以由环回电路 40 所接收的数据(或者可以并非所提供的数据),并且结果可以被存储在用于 DMA 传输的描述符中。被转换的数据、或者被计算并包含于数据中或写入 DMA 控制符中的结果一般可以在此处被称为“结果”。

这样,在一些实施方案中,环回电路 40 可以被配置为对 Tx DMA 数据执行一种或多种操作(或“功能”)以产生结果(例如,被转换的 DMA 数据、或者从数据产生的结果)。在图 2 的实施方案中,环回电路 40 可以包括被耦合到发送数据路径的拷贝 FIFO 42、卸载引擎 44、和异或 (XOR) 电路 46。拷贝 FIFO 42 可以存储来自 Tx DMA 数据路径的发送数据用于 Rx DMA 数据路径上的传送。因此,拷贝 FIFO 42 可以执行拷贝 DMA 操作。卸载引擎 44 可以被配置为对 DMA 数据执行各种操作,产生被转换的数据或者与数据分离的结果。在各种实施方案中,卸载引擎 44 可以被配置为提供任意所需的操作集合。在一种实施方案中,卸载引擎 44 可以被配置为执行对分组处理提供辅助的操作。例如,已经开发了各种网络安全协议来提供分组的加密

和/或认证。认证一般包括对一些或所有的分组生成散列 (hash)。这样, 卸载引擎 44 可被配置为在 DMA 传输中对分组数据执行加密/解密和/或散列函数。另外, 卸载引擎 44 可以被配置为执行校验码生成/校验、和/或 CRC 生成/校验。校验和和/或 CRC 保护被用于各种分组协议中。XOR 电路 46 可以对 DMA 数据(例如, 来自多个来源的 DMA 数据)进行按位 XOR。XOR 电路 46 可以被用于例如支持廉价磁盘冗余阵列 (RAID) 处理和其它使用 XOR 功能的类型或处理。

环回电路 40 (更确切地说, 环回部件 42、44、和 46) 可以在将 DMA 数据提供给环回电路 40 的 DMA 传输期间操作 DMA 数据。即, 环回电路 40 可以至少开始执行对 DMA 数据的操作而 Tx DMA 传输提供剩余的 DMA 数据。一般来说, 结果可以被写入存储器, 或者更通常地, 写入主机地址空间 (例如, 作为被转换的 DMA 数据、被附加到 DMA 数据、或者写入到单独的结果存储位置诸如用于 Tx DMA 传输的 DMA 描述符中的字段)。

环回电路 40 还可以包括用于卸载引擎 44 和 XOR 电路 46 的 FIFO (被耦合到卸载引擎 44 的卸载 FIFO 48 和被耦合到 XOR 电路 46 的 XOR FIFO 50)。FIFO 48 和 50 可以分别临时存储来自卸载引擎 44 和 XOR 电路 46 的数据, 直到可以在接收 DMA 数据路径上发送 DMA 数据为止。在所示实施方案中提供了仲裁器 52, 其被耦合到 FIFO 42、48 和 50, 以在 FIFOs 之间进行仲裁。仲裁器 52 还被耦合到可临时存储来自环回电路 40 的要被写入目标的数据的环回 FIFO 54。

在所示实施方案中, DMA 控制器 14 包括 Tx DMA 数据路径上的 Tx 控制电路 56, 以及 Rx DMA 数据路径上的 Rx 控制电路 58。Tx 控制电路 56 可以从主机 12 中预取数据以用于发送 DMA 传输。特别地, Tx 控制电路 56 可以预取 DMA 描述符, 并可以处理 DMA 描述符以判断 DMA 数据的来源地址。然后, Tx 控制电路 56 可以预取 DMA 数据。虽然术语预取被用于指 Tx 控制电路 56 的操作, 但是预取通常可以是所产生的从主机地址空间中读取描述符和 DMA 数据的读取操

作。

Tx 控制电路 56 将 DMA 数据发送给目标。在本实施方案中，目标可以是接口电路 16 或环回电路 40 之一（更确切地说，所示实施方案中的拷贝 FIFO 42、卸载引擎 44、和 XOR 电路 46 之一）。Tx 控制电路 56 可以识别发送数据的目标（例如，通过发送目标标识符）。替代地，可以在 Tx 控制电路 56 与接口电路 16 之间、以及在 Tx 控制电路 56 与环回部件 42、44 和 46 之间提供物理上分离的路径。Tx 控制电路 56 可以包括一组缓冲器 62 以临时存储要被发送的数据。Tx 控制电路 56 还可以与数据一起提供多个控制信息。控制信息可以包括来自 DMA 描述符的信息。控制信息可以包括，对于环回电路 40 而言，用于在目标地址空间中存储数据的缓冲器指针。控制信息也可以包括任何其它可以被包括于 DMA 描述符中的以及可以被接口电路 16 或环回电路 14 所使用的控制信息。关于 DMA 描述符的讨论，以下会提供更详细的示例。

Rx 控制电路 58 可以接收要被写入主机 12 地址空间的 DMA 数据，并且可以产生写入操作以将数据存储到存储器。在一种实施方案中，软件可以分配存储器中的缓冲器以存储所接收的 DMA 数据。可以向 Rx 控制电路 58 提供缓冲器指针（在主机地址空间中标识缓冲器的地址）。Rx 控制电路可以使用缓冲器指针以产生用于写入操作的地址以存储数据。可以提供 Rx 预取引擎 60 以为 Rx 控制电路 58 预取缓冲器指针。Rx 预取引擎 60 被耦合以向 Rx 控制电路 58 提供缓冲器指针。Rx 预取引擎 60 可以包括一组缓冲器 64 以临时存储为 Rx 预取引擎 60 所使用的预取缓冲器指针。类似地，Rx 控制电路 58 可以包括一组缓冲器 68 以临时存储所接收的要被写入存储器的 DMA 数据。

在一种实施方案中，Rx 控制电路 58 可以被配置为生成描述符用于所接收的 DMA 数据。即，软件可以分配缓冲器以存储 DMA 数据并可以提供缓冲器指针，而不是让软件为所接收的 DMA 数据创建 DMA 描述符。Rx 控制电路 58 可以将所接收的 DMA 数据存储于所

分配的缓冲器中，并可以为 DMA 传输创建描述符。由 Rx 控制电路 58 创建的描述符可以包括指向存储所接收的 DMA 数据以及其它描述 DMA 传输的信息的一个或多个缓冲器的一个或多个缓冲器指针。接收 DMA 描述符的示例实施方案显示于图 12 中并在以下被更详细地描述。因为 Rx 控制电路 58 为所接收的 DMA 数据创建描述符，所述描述符可以比那些由软件所创建的描述符更有效。例如，软件可能不得不创建可接收最大可能 DMA 传输（或者对于更大传输可能需要多个描述符）的接收 DMA 描述符，以及为了存储最大可能 DMA 传输可能不得不分配足够的缓冲器。另一方面，由 Rx 控制电路 58 创建的描述符对于实际所接收的传输而言可以足够大（并且可以耗用足够的缓冲器来存储所接收的数据），但是不需要更大。

在所示实施方案中，Rx 控制电路 58 可以从仲裁器 66 接收 DMA 数据，仲裁器 66 被耦合到环回 FIFO 54 并也从接口电路 16 接收 DMA 数据。仲裁器 66 可以在环回 FIFO 54 和接收自接口电路 16 的 DMA 数据之间进行仲裁以将数据传输给 Rx 控制电路 58。

仲裁器 52 和 66 可以实现任何所需的仲裁方案。例如，基于优先级的方案、轮询调度方案（round-robin）、加权轮询方案（weighted round-robin）、或者这些方案的组合都可以被使用。在一些实施方案中，仲裁方案可以是可编程的。由仲裁器 52 所实现的仲裁方案可以不同于由仲裁器 66 所实现的方案。

在所示实施方案中，Tx 控制电路 56、Rx 预取引擎 60、和 Rx 控制电路 58 被耦合到 IOM/IOB 接口单元 70。IOM/IOB 接口单元 70 可以代表 Tx 控制电路 56、Rx 预取引擎 60、和 Rx 控制电路 58 与 IOB 22 和 IOM 24 进行通信。IOM/IOB 接口单元 70 可以从 Tx 控制电路 56、Rx 预取引擎 60、和 Rx 控制电路 58 接收读取和写入请求，并且可以与 IOB 22 和 IOM 24 进行通信以满足这些请求。

特别地，IOM/IOB 接口单元 70 可以从 Tx 控制电路 56 接收对描述符和 DMA 数据的读取请求，以及从 Rx 预取引擎 60 对存储了缓冲器指针的存储器的读取请求，并且可以将请求传送给 IOB 22。IOB 22

可以表明哪个 IOM 24 入口 (entry) 存储了包括所请求数据的缓存行 (cache line) (例如在从主机地址空间或 IOC 26 中读取数据后, 或者数据可能由于前一个请求而已经在 IOM 24 中), 并且 IOM/IOB 接口 70 可以从 IOM 24 中读取数据并将它提供给 Tx 控制电路 56 或 Rx 预取引擎 60。IOM/IOB 接口单元 70 也可以从 Rx 控制电路 58 接收写入请求, 并可以将写入数据存储于 IOM 24 中 (在由 IOB 22 为写入数据所分配的入口处)。一旦缓存行积累在 IOM 24 中 (或者 DMA 传输完成, 无论哪个发生), IOM/IOB 接口单元 70 就可以通知 IOB 22 并且可以提供要被写入的缓存行的地址 (从指向要被写入的缓冲器的缓冲器指针导出)。

在一种实施方案中, DMA 控制器 14 可以支持各种通道 (channel) 用于发送 DMA 传输和接收 DMA 传输。在各种实施方案中, 可以支持任意数目的通道。例如, 在一种实现中, 可以提供 20 个发送 DMA 通道以及 64 个接收 DMA 通道。每个通道可以是独立的从来源到目的的逻辑数据路径。可以根据需要由软件来分配通道。

更确切地说, 每个发送通道可以被分配给接口电路 16 之一或者环回组件电路 42、44、或 46 之一。并非所有的发送通道都需被使用 (即, 一些传输通道可以被禁止)。Tx 控制电路 56 可以以每个通道为基础 (per-channel basis) 预取 DMA 描述符和 DMA 数据。即, Tx 控制电路 56 可以为每个具有可用的 DMA 描述符的通道独立地生成预取。Tx 控制电路 56 可以在所生成的预取之中进行选择以将读取请求发送给 IOM/IOB 接口单元 70。

每个接收通道都可以被分配给接口电路 16 之一。并非所有的接收通道都需要被使用 (即, 一些接收通道可以被禁止)。Rx 控制电路 58 可以将通道编号与所接收数据一起接收。环回电路 40 可以为 DMA 提供来自 DMA 描述符的缓冲器指针, Rx 控制电路 58 可以使用缓冲器指针以将 DMA 数据写入到主机地址空间。接口电路 16 可以利用所分配的通道来编程, 或者可以采用分组过滤以确定通道。接口电路 16 可以将通道编号与 DMA 数据一起提供, Rx 控制电路 58 可以

使用由 Rx 预取引擎 60 为通道提供的缓冲器指针以将 DMA 数据写入主机地址空间。

DMA 控制器 14 可以包括各种配置寄存器 38D-38H，如图 2 所示。配置寄存器 38D-38H 可以是可编程的以使能/禁止 DMA 控制器 14 的各种可编程特征和/或配置可编程特征，如上所述。例如，Tx 控制电路 56 中的配置寄存器 38D 可以包括用于每个通道的描述符环的地址，以及表示可用描述符的数量的描述符计数。配置寄存器 38D 还可以包括对接口电路 16 和部件件环回功能的发送通道的分配。各种其它的按通道的配置和非通道相关的配置可以被存储在配置寄存器 38D 中。类似地，配置寄存器 38E 可以存储用于每个接口电路 16 的缓冲器指针环地址、缓冲器环计数等，以及各种非通道相关的配置。配置寄存器 38F 可以存储各种接收 DMA 配置。配置寄存器 38G 可以将用于环回电路 40 的配置视为整体进行存储，以及所需要用于每个组件电路的配置。配置寄存器 38G 也可以存储用于仲裁器 52 的配置（例如，选择仲裁方案，为所选择仲裁方案的配置进行编程）。配置寄存器 38H 可以存储用于仲裁器 66 的仲裁配置。

要注意的是，虽然 Tx 控制电路 56 实现了预取以获取描述符和 DMA 数据，其它实施方案可以不实现预取。因而，总的来说，可以有被配置为执行发送 DMA 传输（和到环回电路 40 的 DMA 传输）的 Tx 引擎 56 或 Tx 控制电路 56。

要注意的是，本说明说明了用于 DMA 传输的缓冲器和缓冲器指针。由缓冲器指针所指向的缓冲器（与硬件存储缓冲器诸如 62、64、和 68 相对）可以包括连续的存储区域。软件可以分配存储区域以存储 DMA 数据（用于发送或者作为接收 DMA 数据的区域）。缓冲器指针可以包括主机地址空间中的存储区域的地址。例如，缓冲器指针可以指向存储区域的基地址(base)或者存储区域的边界地址(limit)。

现在转到图 3，显示了卸载引擎 44 的一种实施方案的框图。在所示实施方案中，卸载引擎 44 包括输入缓冲器 80、输出缓冲器 82、一组安全电路 84A 到 84D、CRC 生成器 86、以及校验和生成器 88。

输入缓冲器 80 被耦合到 Tx 控制电路 56 并被耦合到安全电路 84A ~ 84D、CRC 生成器 86、以及校验和生成器 88。输出缓冲器 82 被耦合到安全电路 84A ~ 84D、CRC 生成器 86、以及校验和生成器 88。输出缓冲器 82 还被耦合到卸载 FIFO 48。图 3 更详细地显示了用于一种实施方案的安全电路 84A，而安全电路 84B ~ 84D 可以类似。安全电路 84A 包括散列电路 90 和密码电路 92。散列电路 90 和密码电路 92 两者都被耦合到输入缓冲器 80 和输出缓冲器 82。另外，散列电路 90 的输出被耦合作为密码电路的输入而密码电路 92 的输出被耦合作为散列电路 90 的输入，呈“蝶形”构造。

安全电路 84A ~ 84D 可以被配置为执行各种操作以卸载分组处理的安全功能。特别地，安全电路 84A ~ 84D 可以被配置为执行包括在各种安全分组规范（例如，安全互联网协议（IPSec）或安全套接层（SSL））中的加密/解密（共同称为密码、或密码功能）和散列函数。

一般来说，使用安全分组协议的通信包括协商会话（negotiation session），其中终端交流它们所能使用的协议、所支持的安全方案、加密和散列的类型、密钥或证书交换等。然后就是使用所商定的协议、加密等的成批传输（bulk transfer）阶段。在成批传输期间，分组可以被接收到主机 12 中（例如，从接口电路 16 之一经过接收 DMA 路径）。软件可以查阅存储器中的数据结构以获取密钥、加密算法等等，并准备经由卸载引擎 44 的 DMA 传输来解密和/或认证分组。类似地，软件可以为安全发送准备分组并使用经由卸载引擎 44 的 DMA 传输来加密和/或认证分组。

散列电路 90 可以实现可被用于分组认证的各种散列函数。一般说来，至少对一部分分组计算散列，并将散列结果包括在分组中。当分组在其目的被接收时，散列可以被校验以检测分组中是否有任何字段已被改变过（从而检测分组在从其来源发送的过程中是否被修改过）。在一种实施方案中，散列电路 90 中可以支持下列散列函数：消息摘要 5（MD-5）/安全散列算法 1（SHA-1），和散列消息认证码

(HMAC)。其它实施方案可以实现 SHA-2。其它实施方案可以实现任何其它的散列函数，包括上述函数的子集或超集和其它函数。

密码电路 92 可以被配置为执行密码功能。依据安全分组规范，密码功能可以被应用于分组的至少一部分，可能包括散列数据。在各种实施方案中，可以支持任何密码功能集合。例如，在一种实施方案中，密码电路 92 中可以实现下列加密/解密算法：数据加密标准 (DES)、三重数据加密标准 (3DES)、高级加密标准 (AES)、Kasumi、所谓的 Ron 氏码 4 (ARC4) 和/或 Ron 氏码 4 (RC4)。

在一些情况下，如果认证和密码功能都被使用，那么当准备用于发送的分组时，先执行加密，然后对已加密数据执行认证散列（例如 IPsec）。在其它情况下，先执行认证散列，然后执行分组（包括散列数据）的加密（例如 SSL）。在这两种情况中，对所接收的数据以相反的顺序执行认证散列和解密。

通过电路 90 和 92 之间的蝶形连接，安全电路 84A 到 84D 可以支持在单个 DMA 传输中的数据加解密和散列的两个顺序中的任何一个。即，如果先执行加解密，那么提供给安全电路 84A 的数据可以被发送到密码电路 92，密码电路 92 的输出可以被发送到散列电路 90 的输入以对加密（或解密）数据计算散列函数。如果先执行散列，那么提供给安全电路 84A 的数据可以被发送到散列电路 90，散列电路 90 的输出可以被发送到密码电路 92 的输入。安全电路 84A ~ 84D 也支持在给定的 DMA 传输中只执行散列或者只执行密码功能。用于 DMA 传输的来自 DMA 描述符的控制信息可以控制数据经过安全电路 84A ~ 84D 的路线。

所示实施方案显示了 4 个安全电路 84A ~ 84D。其它实施方案可以包括任意数量的安全电路，包括一个安全电路。在一种实施方案中，安全电路 84A ~ 84D 可以用两倍于系统 10 中所用系统时钟的频率进行钟控，每个系统时钟周期可以接收两个操作（一个在系统时钟的上半个周期执行，另一个在系统时钟的下半个周期执行）。因而，可以有 8 个可供软件选择来执行安全功能的逻辑安全电路。

CRC 生成器 86 可以被配置为对在指定了 CRC 生成的 DMA 传输中所提供的数据生成 CRC 数据。CRC 生成也可以被用于校验来自所接收的分组的 CRC 数据。例如，CRC 生成器 86 中所生成的 CRC 数据可以与所接收的分组中的相应 CRC 数据进行比较。替代地，可以通过 CRC 生成器 86 将所接收的分组的 CRC 数据包括在 DMA 传输中，并且可以根据预定值来校验结果以检测所接收分组中的错误。在一些实施方案中，可以有不止一个 CRC 生成器 86。此外，CRC 生成器 86 可以被以两倍于系统时钟频率进行钟控，类似于安全电路 84A ~ 84D，以提供比卸载引擎 44 中物理上所提供的 CRC 生成器更多的逻辑 CRC 生成器。在一种特定实施方案中，可以有 4 个以系统时钟频率的两倍钟控的 CRC 生成器 86，以提供与安全电路 84A ~ 84D 相等数目的逻辑单元（8 个）。

校验和生成器 88 可以被配置为对在指定了校验和生成的 DMA 传输中所提供的数据生成校验和。校验和生成也可以被用于校验来自所接收分组的校验和数据。例如，校验和生成器 88 中所生成的校验和数据可以与所接收的分组中的相应校验和数据进行比较。替代地，可以通过校验和生成器 88 将所接收的分组的校验和数据包括在 DMA 传输中，并且可以根据预定值来校验结果以检测所接收分组中的错误。在一些实施方案中，可以有不止一个校验和生成器 88。

输入缓冲 80 可以临时存储由 Tx 控制电路 56 所提供的数据，直到目标电路 84A ~ 84D、86、或 88 可以对数据操作为止。电路 84A ~ 84D、86、和 88 可以将数据输出到输出缓冲器 82 以被写入卸载 FIFO 48。在其它实施方案中，可以不包括输入缓冲器 80 和/或输出缓冲器 82。

图 4 为显示了可以在系统 10 的一种实施方案中所实现的 DMA 模型的一种实施方案的框图。如上所述，DMA 传输可以从来源地址空间或接口（模块 100）到目的地址空间或接口（模块 102）发生。图 4 中的模块 104 表示 DMA 传输。可选地，DMA 传输可以包括对 DMA 数据执行一个或多个操作、或功能（模块 106）以产生结果。结

果被传回到 DMA 模块 104 并可以被提供给目标地址空间。在一些情况中,结果可以是可以被写入目标地址空间的被转换的 DMA 数据。在其它情况中,结果可以与 DMA 数据分开并可以附加到 DMA 数据(例如,附加于 DMA 数据的末尾)或者可以被存储于单独的位置(例如,在用于 DMA 传输的 DMA 描述符中)。

在一些情况下,图 4 可以表示单个 DMA 传输。在其它情况中,可以使用多个 DMA 传输以完成图 4 的模型。例如,功能 106 可以由环回电路 40(或者其部件,对于图 2 的实施方案)执行。因而,在所示实施方案中,如果要求在目标是接口电路 16 的 DMA 传输上执行功能,那么可以执行两个 DMA 传输。第一 DMA 传输可以是从来源地地址空间到目标地址空间,指定要被执行的所需功能。接着,第二 DMA 传输可以被执行,使用第一 DMA 传输的目标地址空间作为来源并将所需接口电路作为目标。

例如,分组可以由软件进行准备以用于经由 MAC 34A~34B 的发送,并且分组可以被存储于主机 12 中的存储器中。可能需要使用密码和/或散列的安全发送,因此软件可以建立从主机 12 中的第一存储区域到第二存储区域的第一 DMA 传输,并且第一 DMA 传输可以指定卸载引擎 44 中的加密和/或散列。软件也可以准备从第二存储区域到目标 MAC 34A~34B 的第二 DMA 传输。类似地,分组可以被接收作为到主机 12 中的存储器的 DMA,并且软件可以提供卸载引擎 44 建立 DMA 传输以校验散列和/或解密分组。

其它实施方案可以允许当数据被发送到接口电路 16 时应用功能。例如,在一种实施方案中,IOB22 中可以支持部分校验和生成以用于生成 TCP 分组的 TCP 标头中所包括的校验和。当为这种分组指定 DMA 传输时,IOB 22 可以累加 IOM 24 中的所述分组数据,并且可以为每个缓存行生成部分校验和。可以在 DMA 控制器 14 中,使用来自 IOB 22 的每个缓存行的部分校验和以及没有被计算校验和的数据,生成最终的校验和,并且 DMA 控制器 14 可以将所算的校验和插入到 TCP 标头中。其它实施方案可以允许环回电路 40 的输出被直接

发送到接口电路 16，作为其中执行所述功能和发送所述数据的单个 DMA 传输的一部分。此外，其它实施方案可以允许来自 Rx DMA 数据路径的数据被输入到环回电路 40 以在接收 DMA 传输期间对所接收的 DMA 数据执行功能。

接下来转到图 5，显示了存储描述符数据结构和缓冲器指针数据结构的存储区域 110 的框图。在图 5 的实施方案中，描述符数据结构包括一组描述符环 112A ~ 112N。对于由 DMA 控制器 14 支持的每个 DMA 通道（例如图 5 中的通道 0 到通道 N）都可以有一个描述符环。即，DMA 通道和描述符环之间可以是一一对应的，并且对于给定 DMA 通道的 DMA 传输可以在分配给该通道的描述符环 112A ~ 112N 中具有相对应的描述符。另外，在图 5 的实施方案中，缓冲器指针数据结构可以包括一组缓冲器指针环 114A ~ 114M。每个接口电路 16（例如，图 5 中的接口电路 0 到 M，其中接口电路 16 的数量可以是 M+1）都可以有一个缓冲器指针环。即，接口电路和描述符环之间可以是一一对应的，并且用于 DMA 在该接口上所接收的数据的缓冲器指针可以取自分配给该接口电路的缓冲器指针环 114A ~ 114M。

每个描述符环 112A ~ 112N 可以包括一组描述符用于相应的 DMA 通道。对于发送 DMA 通道，可以按环内所包括的顺序来处理描述符，从环中第一个描述符到最后一个，然后在最后的描述符被处理之后环绕到环中的第一个描述符。这样，在给定时刻，环中的任意描述符都可以被视为接下来要被处理的“当前描述符”。软件可以以各种方式控制可用于 DMA 通道上处理的描述符的数目，如上所述。因此，如果在给定的发送 DMA 通道上（在相应的描述符环中）有可用的描述符，那么 DMA 控制器 14 可以执行指定的 DMA 传输（与其它 DMA 通道一起进行资源仲裁）。对于本实施方案中的接收 DMA 通道，相应的描述符环中的描述符可以在 DMA 传输被在该通道上接收到时被消耗。DMA 控制器 14 可以使用用于存储所接收的 DMA 数据的缓冲器指针将当前描述符写入，以及将其它有关 DMA 传输的信息诸如传输状态信息写入。

其它实施方案可以使用其它数据结构（例如描述符的链表）。每个描述符环 112A ~ 112N 的基地址可以被提供给 DMA 控制器 14（例如，编程到配置寄存器 38D 或 38F 中，取决于通道是发送通道还是接收通道）。描述符环 112A ~ 112N 的其它属性也可以被编程（例如范围（extent））。在一些实施方案中，给定环中的描述符可以是固定尺寸的，使得给定描述符可以位于环的基地址的固定偏移处。在其它实施方案中，描述符可以是可变尺寸的。在另一些实施方案中，描述符可以是固定尺寸或可变尺寸的，取决于配置寄存器 38D 或 38F 中的可编程属性。所述属性可以按通道在通道上可编程的，或者可以是将通道视为整体进行编程。

每个缓冲器指针环 114A ~ 114M 包括指向存储器中由软件所分配的缓冲器的缓冲器指针，以用于从相应接口接收来自 Rx DMA 传输的 DMA 数据。与描述符环 112A ~ 112N 相似，软件可以以任意所需方式在缓冲器指针环 114A ~ 114M 中产生 DMA 控制器 14 可用的缓冲器指针。用于每个接口电路的缓冲器指针环的基地址可以被编程到 DMA 控制器 14 中（例如，图 2 的实施方案中，在 Rx 预取引擎 60 中的配置寄存器 38E 中），并且在任何给定时刻，缓冲器指针环中的缓冲器指针之一都可以是要被为相应接口而消耗的下一个缓冲器指针。

在一些实施方案中，通过提供与接口电路而不是与 DMA 通道相关联的缓冲器指针环 114A ~ 114M，软件可以将缓冲器分配给数目较少的接口电路而不是数目较多的 DMA 通道。在一些情况中，存储器的分配可以更有效率。处理较多业务量的接口电路可以被分配更多缓冲，而无需软件预先知道将接收业务量的是什么通道。当从给定接口接收 DMA 数据时，数据可以被存储在分配给该接口的缓冲器中，并且缓冲器指针可以被写入到接收到 DMA 数据的通道的描述符中。描述符可以是描述符环 112A ~ 112N 其中之一，取决于与 DMA 传输相关联的是哪个接收 DMA 通道。

对于每个缓冲器指针，缓冲器指针环 114A ~ 114M 也可以包括

尺寸字段(图 5 中的 Sz)。尺寸字段可以表明缓冲器指针所指向的缓冲器的尺寸。因此,软件可以基于例如可用存储量、给定接口上的 DMA 传输的期望尺寸等来分配不同尺寸的缓冲器。

现在转到图 6 的流程图,显示了对于给定接口电路,Rx 预取引擎 60 的一种实施方案的操作。Rx 预取引擎 60 可以包括对于并行并独立地操作的每个接口电路实现了图 6 中所示操作的电路。虽然图 6 中为了易于理解按特定顺序表示了模块,但是模块可以实现图 6 中所示操作的组合逻辑电路并行地实现。在一些实施方案中,可以将模块中的一个或多个或以流程图作为整体跨多个时钟周期进行流水作业。

Rx 预取引擎 60 可以判断(在接口电路相应的缓冲器指针环 114A~114M 中)是否有接口电路可用的缓冲器指针(判断模块 120)以及接口电路是否需要缓冲器指针(判断模块 122)。如果至少一个缓冲器指针是可用和需要的(判断模块 120 和 122,分支“是”),Rx 预取引擎 60 可以生成预取请求以从主机 12 的存储器中的缓冲器指针环 114A~114M 读取缓冲器指针(模块 124)。

如果在相应的缓冲器指针环 114A~114M 中有缓冲器指针还没有被 Rx 预取引擎 60 所预取,那么缓冲器指针可以基本上是“可用的”。缓冲器指针可以由软件插入到缓冲器指针环 114A~114M 中,并且软件可以以任意前述方式表明它们是可用的(例如,在缓冲器指针环入口中使用有效位、类似于上述的 DMA 描述符计数那样递增缓冲器指针环计数器等)。缓冲器指针也可以以各种方式被视为是“被需要的”。例如,如果接收 DMA 通道被使能并且没有为通道预取缓冲器指针,那么缓冲器指针可以是“被需要的”。在一些实施方案中,Rx 预取引擎 60 可以被编程以表明应被预取的缓冲器指针的数量,或者应被预取的缓冲器指针的最小和最大数量。Rx 预取引擎 60 可以为缓冲器指针生成预取请求以尝试预取所编程数量的缓冲器指针。

如上所述,对于每个被使能的接口电路,图 6 的操作可以并行执行。如果并发地生成了多于一个的预取请求,那么 Rx 预取引擎 60 也

可以包括用于在预取请求之中进行选择的电路。例如，在一种实施方案中，Rx 预取引擎 60 可以实现在预取请求之中的固定优先级方案。在其它实施方案中，Rx 预取引擎 60 可以选择与目前被预取和预备了最少指针的接口电路相对应的预取请求。作为另一个例子，Rx 预取引擎 60 可以根据哪个接口电路在当前预取缓冲器指针数目和用于该接口电路的缓冲器指针所需数目之间具有最大差别，来对请求进行加权。也可以使用轮询调度或基于优先级的选择机制，并且如果需要的话，这些机制可以包括接口电路之中的可编程的权重。饥饿预防 (starvation prevention) 机制诸如按接口超时 (per-interface timeout) 也可以被用于确保为每个接口电路预取指针。

所预取的缓冲器指针在要被读取的 IOM 24 中可用时，可以由 IOM/IOB 接口电路 70 通知 Rx 预取引擎 60。在一些实施方案中，Rx 预取引擎 60 可以将一些或全部的指针从 IOM 24 读取到缓冲器 64 中，并且可以视需要将所预取的缓冲器指针提供给 Rx 控制电路 58。

现在参考图 7 的流程图，表示了 Rx 控制电路 58 响应接收数据的一种实施方案的操作。可以从环回电路 40 或接口电路 16 接收数据。虽然图 7 中为了易于理解按特定顺序显示了模块，但是可以利用实现图 7 所示作的组合逻辑电路并行地实现模块。在一些实施方案中，可以将模块中的一个或多个或以流程图作为整体跨多个时钟周期进行流水作业。

Rx 控制电路 58 可以判断是否缓冲器已被用于接收 DMA 数据 (判断模块 130)。如果缓冲器在此前已被选择存储 DMA 数据并且还未满，那么缓冲器可能在被使用。Rx 控制电路 58 可以被配置为为每个 Rx DMA 通道保持活动缓冲 (active buffer) 并且为环回电路 40 保持一个或更多的缓冲器指针。替代地，环回电路 40 可以给每个数据传输提供缓冲器指针，而 Rx 控制电路 58 可以不为环回电路 40 保持指针。如果缓冲器没有在被使用 (判断模块 130，分支“不”)，Rx 控制电路 48 可以选择下一个缓冲器指针以识别要被使用的缓冲器 (模块 132)。对于来自环回电路 40 的 DMA 传输，下一个缓冲器指针可

以被提供自 Rx 预取引擎 60，并且 Rx 控制电路 58 可以表明对 Rx 预取引擎 60 的指针消耗。

在两种情况的任何一种中，Rx 控制电路 58 可以将所接收的数据写入到缓冲器（模块 134）。即，Rx 控制电路 58 可以向 IOM/IOB 接口电路 70 生成写入请求以将 DMA 数据写入到存储器中。至少对于从接口电路 16 所接收的数据，Rx 控制电路 58 可以监控写入到缓冲器中的字节数以判断何时缓冲已满。在一些实施方案中，缓冲器尺寸可以由环回电路 40 提供，并且 Rx 控制电路 58 也可以对于环回传输监视缓冲器满。如果缓冲已满，或者如果数据传输被来源（环回电路或接口电路 16）指示为 DMA 传输结束（判断模块 136，分支“是”），那么 Rx 控制电路 58 可以将缓冲器指针写入到用于 DMA 传输的描述符（模块 138）。对于环回传输，可以没有用于接收数据的描述符，这是因为缓冲器指针可以是来自来源 DMA 描述符的目的缓冲器指针，并且可以不用为这样的传输执行模块 138。如果数据传输被指示为 DMA 传输结束（判断模块 140，分支“是”），Rx 控制电路 58 可以执行各种传输处理结束（模块 142）。例如，对于来自接口电路 16 的 DMA 传输，Rx 控制电路 58 可以为 DMA 传输生成各种状态、为传输生成 DMA 描述符标头、并且将 DMA 描述符写入到与接收了数据的 Rx DMA 通道相对应的描述符环 112A ~ 112N 中。另外，Rx 控制电路 58 可以向 IOM/IOB 电路 70 发出传输结束信号。发出传输结束信号可以通知 IOM/IOB 电路 70 该通道上再不会提供数据。如果未完成的缓存行被 DMA 更新，那么 IOM/IOB 电路 70 可以使被更新的数据被提交给存储器（例如，在 IOC 26 中，在一些实施方案中，以避免互连 30 上的读取 - 修改 - 写入）。

接下来转到图 8 的流程图，表示了对于给定 Tx DMA 通道的 Tx 控制电路 56 的一种实施方案的操作。Tx 控制电路 56 可以包括为每个并行且独立操作的 Tx DMA 通道实现图 8 中所示操作的电路。虽然图 8 中为了易于理解按特定顺序显示了模块，但是可以以图 8 中所示操作的组合逻辑电路来并行地实现模块。在一些实施方案中，可

以将模块中的一个或多个或以流程图作为整体跨多个时钟周期地进行流水作业。

Tx 控制电路 56 可以判断（在通道相应的缓冲器指针环 112A ~ 112N 中的）描述符是否可供通道预取（判断模块 150），以及通道是否需要描述符（判断模块 152）。如果至少一个描述符是可用的和需要的（判断模块 150 和 152，分支“是”），Tx 控制电路 56 可以生成请求以从主机 12 的存储器中的缓冲器指针环 112A ~ 112N 读取描述符（模块 154）。

类似于上述有关图 6 的“可用的”缓冲器指针，如果在相应的描述符环 112A ~ 112N 中有描述符还没有被 Tx 控制电路 56 所预取，那么描述符通常可以是“可用的”。用于 Tx DMA 通道的描述符可以由软件插入到描述符环 112A ~ 112N 中，并且软件可以以任意前述方式表明它们是可用的（例如，在描述符环入口中使用有效位、递增描述符环计数器等）。描述符也可以以各种方式被视为是“被需要的”。例如，如果 Tx DMA 通道被使能并且没有为通道预取描述符，那么描述符可以是“被需要的”。在一些实施方案中，只要在存储描述符的 IOM 24 和/或缓冲 62 中有空间，Tx 控制电路 56 就可以预取描述符。在其它实施方案中，Tx 控制电路 56 可以是可编程的以指示应被预取的描述符数量、或者应被预取的描述符的最小和最大数量。Tx 控制电路 56 可以生成预取描述符的请求以尝试预取所编程数量的描述符。

当被预取的描述符在要被读取的 IOM 24 中可用时，可以由 IOM/IOB 接口电路 70 通知 Tx 控制电路 56。在一些实施方案中，Tx 控制电路 56 可以将一些或全部的描述符从 IOM 24 中读入到缓冲器 62。

Tx 控制电路 56 可以判断 DMA 数据是否可供通道预取（在通道上被发送）（判断模块 156），以及通道是否需要 DMA 数据（判断模块 158）。如果 DMA 数据是可用的和需要的（判断模块 156 和 158，分支“是”），那么 Tx 控制电路 56 可以生成请求以从主机 12 的地址空间中（例如，从主机 12 中的存储位置）读取 DMA 数据（模块 160）。

如果 Tx 控制电路 56 有描述符要被处理（例如，描述符是通道的描述符环中下一个要被处理的），那么 DMA 数据可被认为是可用于预取的，描述符数据在缓冲器 62 或 IOM 24 中，并且描述符数据描述了要被执行的有效 DMA 传输。如果通道上的上一个 DMA 数据已被发送（或即将被发送），那么 DMA 数据可以是需要的。在一些实施方案中，Tx 控制电路 56 可以编程为在给定时间要预取多少数据，如果已被预取但还尚未发送的 DMA 数据数量少于所需数量，那么 DMA 数据可以是需要的。在一些实施方案中，发送通道中的仲裁方案也可以影响是否需要 DMA 数据（例如，如果通道在较长的时间内不会赢得仲裁，那么可能还不需要 DMA 数据，因为直到赢得仲裁为止它不可被发送）。

要注意的是，模块 156、158 和 160 所示的操作可以与模块 150、152 和 154 的操作无关（而不是使用所预取的描述符来判断 DMA 数据是否可用）。因此，实现模块 156、158 和 160 的电路可以与实现模块 150、152 和 154 的电路无关，并且可以用这种电路并行地求值。

如上所述，可以为每个被使能的 Tx DMA 通道并行执行图 8 的操作。如果并发地生成多于一个的预取请求，那么 Tx 控制电路 56 也可以包括用于在预取请求之中进行选择。例如，Tx 控制电路 56 可以选择与目前被预取和预备了最少描述符或最少量 DMA 数据的 Tx DMA 通道相对应的预取请求。作为另一个例子，Tx 控制电路 56 可以基于哪个 Tx DMA 通道在当前预取描述符/DMA 数据数目和用于该通道的描述符/DMA 数据所要数目之间具有最大差别，来对请求进行加权。也可以使用轮询调度或基于优先级的选择机制，并且如果需要的话，这些机制可以包括在通道之中可编程的权重。饥饿预防机制诸如按接口超时也可以被用于确保为每个被使能的通道预取描述符和 DMA 数据。

当被预取的 DMA 数据在要被读取 IOM 24 中可用时，可以由 IOM/IOB 接口电路 70 通知 Tx 控制电路 56。在一些实施方案中，Tx 控制电路 56 可以将一些或全部的 DMA 数据从 IOM 24 中读入到缓冲

器 62。另外，Tx 控制电路 56 可以将所预取的 DMA 数据发送给目标，如图 9 中的实施方案中所示。虽然图 9 中为了易于理解按特定顺序显示了模块，但是可以实现图 9 中所示操作的组合逻辑电路来并行地实现模块。在一些实施方案中，可以将模块中的一个或多个或以流程图作为整体跨多个时钟周期进行流水作业。

如果 Tx DMA 通道被分配给环回电路 40（或更确切地说，分配给环回电路 40 中的环回部件——判断模块 162，分支“是”），Tx 控制电路 56 可以从用于传输的和要与 DMA 数据被一起发送的 DMA 描述符中提取一个或多个目的缓冲器指针（模块 164）。也可以提供每个目的缓冲器指针的尺寸信息，以使 Rx 控制电路 58 可被告知缓冲器的尺寸。环回电路 40 可以接收目的缓冲器指针并将它们与要被写入到存储器的数据（原始 DMA 数据，例如在拷贝 DMA 传输中；或者被转换的 DMA 数据，例如，如果进行了加密的话）一起提供。替代地，如果环回结果要被写入到 DMA 描述符中（例如，散列结果、CRC 结果、或校验和结果），那么可以提供指向 DMA 描述符的指针。Tx 控制电路 56 可以将数据连同通道数目、指针（如果有的话）、控制信息（如果有的话）一起发送给目标接口电路 16 或环回部件（模块 166）。控制信息也可以被从 DMA 描述符中提取（例如，选择所需环回操作的控制信息、用于接口电路的控制信息等）。

可以为每个 Tx DMA 通道并行地执行图 9 的操作。即，Tx 控制电路 56 可以包括为每个 Tx DMA 通道并行地实现图 9 的操作的单独的电路。仲裁机制可以被用于在有 DMA 数据要发送的通道之间进行仲裁，以选择通道以在到接口电路 16 和环回电路 40 的发送数据路径上进行发送。

描述符环 112A ~ 112N 在以上已被描述为存储各种描述符。在一种实施方案中，描述符环 112A ~ 112N 可以存储传输描述符（描述从来源到目的的 DMA 传输）和控制描述符两者。控制描述符可以不指定 DMA 传输，而是代之以将用于通道的控制信息进行编码。例如，控制描述符可以被用于指定配置（例如，一个或多个配置寄存器

38A-38G 的内容)。这样,在 DMA 传输之间,控制描述符可以被用于对 DMA 控制器 14 或其部件、或系统 10 的其它部件在线及时地重构。因此,软件可以,例如,建立一个或更多的传输描述符以在第一配置下执行 DMA 传输,建立一个或更多的控制描述符以改变配置,建立一个或更多的附加传输描述符以在第二配置下执行 DMA 传输,并在随后让描述符作为单位在 DMA 通道上可用。DMA 传输、重构、和附加的 DMA 传输可以都被执行而不用处理器干预。

也提出了控制 DMA 控制器 14 却不明确地(*explicitly*)引发 DMA 传输的其它类型的控制描述符。例如,提出了时间触发器描述符,当时间触发器描述符被处理时,在 DMA 通道上引起时间延迟。例如,系统 10 可以包括可以被用延迟值编程的计时器,并且描述符环中下一个描述符的处理可以被延迟直到计时器终止为止。计时器的期满可以对 DMA 控制器 14 生成触发事件以“唤醒”DMA 控制器 14。在各种实施方案中,用户可以用延迟值对计时器编程,或者延迟值可以被包括在时间触发器描述符中。在其它实施方案中, DMA 控制器 14 可以实现计时器。时间触发器描述符可以有各种用途。例如,时间触发器描述符可以与从来源地址空间拷贝到目的地址空间的拷贝 DMA 描述符交织。来源地址空间可以包括一个或更多的存储器映射 I/O 寄存器或其它系统状态寄存器。这样,重复的拷贝 DMA 传输可以被用于读取寄存器并将结果写入到目的,其被时间触发器描述符延迟。拷贝 DMA 传输可以取代处理器 18A ~ 18B 上的软件循环查询,解放处理器用于其它操作。

图 10 为与通道 0 对应的描述符环 112A 的一种实施方案的框图,说明了将控制描述符嵌入描述符环中的传输描述符内。可以用图 10 的方式将多于一个的控制描述符按要求(连贯地或者在环中的不同点处)嵌入。

在图 10 中,显示了两个传输描述符 170A ~ 170B,接着是控制描述符 172,接着又是两个传输描述符 170C ~ 170D。这样,例如,由描述符 170A ~ 170B 所指定的两个 DMA 传输将由 Tx 控制电路 56

执行，接着处理控制描述符 172。在处理控制描述符 172 之后，接着会执行由描述符 170C ~ 170D 所指定的两个 DMA 传输。

图 11 为流程图，表示了 Tx 控制电路 56 的一种实施方案可以采用的附加处理，用于 Tx DMA 通道所对应的描述符环中的嵌有传输描述符的控制描述符。Tx 控制电路 56 可以包括为每个并行且独立操作的 Tx DMA 通道实现图 11 中所示操作的电路。虽然图 11 中为了易于理解按特定顺序显示了模块，但是可以实现图 11 中所示操作的组合逻辑电路来并行地实现模块。在一些实施方案中，可以将模块中的一个或多个或以流程图作为整体跨多个时钟周期进行流水作业。

Tx 控制电路 56 可以从描述符环中预取描述符（模块 174），假设一个是通道可用的和需要的（例如，如图 8 所述）。Tx 控制电路 56 可以在描述符返回到 Tx 控制电路 56 的时候处理该描述符，并且可以判断该描述符是否为控制描述符（判断模块 176）。例如，描述符标头可以包括指明该描述符是传输描述符亦或是控制描述符的类型字段。如果描述符是传输描述符（判断模块 176，分支“否”），那么 Tx 控制电路 56 可以处理传输描述符（模块 177）。例如，处理传输描述符可以类似于上述图 8 和图 9 的相关描述。

如果描述符是控制描述符（判断模块 176，分支“是”），那么 Tx 控制电路可以判断控制描述符是否是时间触发器描述符（判断模块 178）。可以通过控制描述符标头中的编码来表明时间触发器描述符。替代地，时间触发器描述符可以指定要被载入到配置寄存器 38A-38G 中的值，这类似于其它配置描述符。所载入的值可以指定所需的延迟。Tx 控制电路 56 可以通过对载入配置寄存器的寄存器地址进行译码来检测时间触发器。如果控制描述符是时间触发器描述符（判断模块 178，分支“是”），那么 Tx 控制电路 56 可以延迟处理下一个来自描述符环的描述符直到时间触发器描述符所指定的延迟期满为止（模块 180）。如果控制描述符不是时间触发器描述符（判断模块 178，分支“否”），Tx 控制电路 56 可以用控制描述符值进行重构（模块 182）。例如，控制描述符可以包含标识配置寄存器 38A-38G

的寄存器地址，以及要写入配置寄存器的值。Tx 控制电路 56 可以发起写入操作。替代地，控制描述符可以包括指向存储一系列寄存器地址和要被写入那些寄存器的值的存储器缓冲器的缓冲器指针。这样，使用控制描述符可以进行较大量的配置。在一些实施方案中，可以通过使用控制描述符来取消在处理器 18A ~ 18B 上执行的对每个配置寄存器进行写入的指令。

图 12 ~ 16 举例说明了根据 DMA 控制器 14 的一种实施方案所述的各种类型的描述符。通常，描述符包含标头、非必需的存储结果（例如，由环回电路 40 生成的结果）的数据字段、以及指向存储着 DMA 数据的缓冲器（来源缓冲器指针）或指向可被用于存储 DMA 数据的缓冲器（目的地缓冲器指针）的一个或更多的缓冲器指针。

在本实施方案中，描述符根据它们是接收或是发送 DMA，亦或是视选择所定的由环回电路 40 所执行的功能而不同。接收描述符被用于 Rx DMA 传输，其它类型的描述符为 Tx DMA 传输和环回功能所使用。DMA 控制器 14（更确切地说，在一种实施方案中，Tx 控制电路 56）可以基于对接口电路 16 或环回电路 40 中功能分配给定的 Tx DMA 通道，来判断用于该通道的描述符环中的描述符格式。

在图 12 ~ 16 中，详细图示了各种字段（例如，特别是标头字段）。虽然在图 12 ~ 16 中显示了某些特定信息，但这不是意图在所示信息、或所示信息的子集、或所示信息的替代物之外排除其它信息的使用。在各种实现中，视需要可以包括各种附加信息。

图 12 为接收描述符 190 的一种实施方案的框图。接收描述符 190 可以是由 Rx 控制电路 58 所生成的并被写入到与 Rx DMA 通道相应的描述符环 112A ~ 112N 中的描述符的格式。在图 12 的实施方案中，接收描述符 190 包括标头字段 192（用于一种实施方案的图 12 的分解图中所示）、非必需的散列结果字段 194、以及一个或更多的缓冲器指针字段 196A ~ 196N。每个缓冲器指针字段 196A ~ 196N 包括可以编码了缓冲器尺寸的尺寸字段、以及编码了缓冲器指针的指针字段。Rx 控制电路 58 可以被配置为用来自缓冲器指针环 114A ~ 114M 的用

于存储所接收 DMA 数据的缓冲器指针来写入缓冲器指针字段 196A ~ 196N。散列结果字段 194 可以被用于存储散列结果。

接收标头字段 192 的分解图包括类型字段 192A、样式字段 (style field) 192B、散列保留字段 192C、缓冲器计数字段 192D、分组长度字段 192E、和状态字段 192F。类型字段 192A 可以编码描述符类型, 例如控制描述符或传输描述符。类型字段 192A (及下述其它类似的用于其它描述符的类型字段) 可以将描述符标识为传输描述符, 除了图 16 中所示的控制描述符之外。样式字段 192B 可以编码描述符的样式, 其是指缓冲器指针字段 196A ~ 196N 中的缓冲器指针是否包括来源和目的指针两者或是只有目的指针 (因为来源对于 Rx DMA 传输而言是接口电路)。散列保留字段 192C 可以被用于表明非必需的散列结果字段 194 是否被包含在描述符中。缓冲器计数字段 192D 可以被用于对用于存储接收 DMA 数据的缓冲器的数目也就是缓冲器指针字段 196A ~ 196N 的数目的计数进行编码。分组长度字段 192E 可以被用于缓冲器中存储的分组的长度 (例如, 以字节为单位) 进行编码。状态字段 192F 可以包括各种用于传输的状态, 可以包括由 Rx 控制电路 58 所生成的状态 (例如, 错误指示) 和/或由来源接口电路 16 所提供的状态。Rx 控制电路 58 可以在终止 DMA 传输时对接收标头 192 进行写入。

图 13 是发送描述符 200 的一种实施方案的框图。发送描述符 200 可以由 Tx 控制电路 56 为对接口电路 16、尤其是对 MAC 34A ~ 34B 的 Tx DMA 传输所用的描述符格式, 并且可以由软件写入到被分配给接口电路 16 的 Tx DMA 通道所对应的描述符环 112A ~ 112N 中。在图 13 的实施方案中, 发送描述符 200 包括标头字段 202 (用于一种实施方案的图 13 的分解图中所示)、以及一个或更多的缓冲器指针字段 204A ~ 204N。每个缓冲器指针字段 204A ~ 204N 包括可以编码了缓冲器尺寸的尺寸字段、以及编码了缓冲器指针的指针字段。Tx 控制电路 56 可以被配置为读取缓冲器指针字段 204A ~ 204N 以从用于发送的缓冲中预取 DMA 数据。

发送标头字段 202 的分解图包括类型字段 202A、样式字段 202B、MAC 配置字段 202C、分组长度字段 202D、和分组信息字段 202E。类型字段 202A 和样式字段 202B 可类似于上述的类型字段 192A 和样式字段 192B。MAC 配置字段 202C 可以用各种用于被 DMA 传输作为目标的 MAC 34A ~ 34B 的分组特定的 (packet-specific) MAC 配置信息进行编码。例如, MAC 配置字段 202C 可以包括虚拟局域网 (VLAN) 配置 (例如, 无、插入、移除、或修改)、CRC 配置 (例如, 无、插入 CRC、填充 (pad) CRC、修改 CRC)、以及是否修改 MAC 来源地址。分组长度字段 202D 可以用缓冲中所存储的分组的长度 (例如, 以字节为单位) 进行编码。分组信息字段 202E 可以用各种描述分组的信息 (例如, IP 标头长度、以太网标头长度、分组类型 (TCP/UDP)、校验和使能等) 进行编码。

图 14 为拷贝描述符 210 的一种实施方案的框图。拷贝描述符 210 可以是 Tx 控制电路 56 为使用拷贝 FIFO 42 的拷贝 DMA 传输 (从主机 12 中的一个存储区域到主机 12 中的另一个存储区域) 所用的描述符的格式。这样, 可以在与分配给拷贝 FIFO 42 的 Tx DMA 通道所对应的描述符环 112A ~ 112N 中使用拷贝描述符 210。在图 14 的实施方案中, 拷贝描述符 210 包括标头字段 212 (用于一种实施方案的图 14 分解图中所示)、和一个或更多的缓冲器指针字段 214A ~ 214N。每个缓冲器指针字段 214A ~ 214N 包括可以编码了缓冲器尺寸的尺寸字段、以及编码了缓冲器指针的指针字段。另外, 在该实施方案中, 每个缓冲器指针字段 214A ~ 214N 包括来源/目的 (S/D) 字段, 其中将指针标识为来源指针 (定位存储着源 DMA 数据的缓冲器) 或者目的指针 (定位要存储 DMA 数据的缓冲器)。Tx 控制电路 56 可以被配置为读取缓冲器指针字段 214A ~ 214N 以从用于发送的来源缓冲器中预取 DMA 数据, 并将指向用于发送的拷贝 FIFO 42 的目的指针提供给 Rx 控制电路 58。

在一种实施方案中, 对于拷贝描述符 210 中的给定目的指针, 可以有多个的来源指针。DMA 控制器 14 可以将数据从来源缓冲器

中按拷贝描述符 210 中所列出的顺序拷贝到目的缓冲器中。这样，DMA 控制器 14 可以支持在拷贝操作中将多个存储区域中的散乱数据收集到目的存储区域中。同样，在一种实施方案中，对于拷贝描述符 210 中的给定来源指针，可以有多于一个的目的指针。在这样的实施方案中，可以支持来自来源缓冲的数据的散乱 (scatter)。

发送标头字段 212 的分解图包括类型字段 212A、样式字段 212B、来源类型字段 212C、目的类型字段 212D、和逻辑块长度字段 212E。类型字段 212A 和样式字段 212B 可类似于上述的类型字段 192A 和样式字段 192B。来源类型字段 212C 和目的类型字段 212D 可以被编码以表明当 DMA 传输进行时来源缓冲器指针和目的缓冲器指针应如何被修改。例如，在一种实施方案中，每个缓冲器指针可以是下列类型之一：顺序递增；顺序递减；或固定（以各种固定宽度，例如，1、2、4、8 或 16 字节）。顺序递增表示地址在每个数据发送之后被增加所发送的数据量。递减也类似，只是地址被递减。在数据被写入连序的存储位置的情况下，顺序递增或顺序递减可以被用于存储区域。如果地址为映射到寄存器或设备端口的存储器，那么可以使用固定选项，并且宽度可以是每个对寄存器/器件的发送的宽度。来源类型字段 212C 也可以有用于零的编码，并且被用于将一组零写入到目的。目的类型字段 212D 也可以有只用于预取的编码，其中来源 DMA 数据被预取但不被写入到目的。在一些实施方案中，逻辑块长度字段可以被用于表明可跨越多个 DMA 描述符的逻辑 DMA 块的长度。即，可以实际上用多个描述符指定逻辑 DMA 操作，并且逻辑 DMA 块长度可以为逻辑 DMA 操作的长度（例如，关于多个描述符的数据传输的总和）。

XOR 电路 46 可以使用与发送 DMA 描述符 200 类似的描述符。多个通道可以被分配给 XOR 电路 46，并且每个通道中的描述符可以指定 XOR 来源中的一个。第一通道也可以为 XOR 结果指定目的（目的缓冲器）。

图 15 是卸载描述符 220 的一种实施方案的框图。卸载描述符 220

可以是 Tx 控制电路 56 为指定卸载引擎 44 的 DMA 传输所用的描述符的格式。这样，可以在与分配给卸载引擎 44 的 Tx DMA 通道所对应的描述符环 112A ~ 112N 中使用卸载描述符 220。图 15 的实施方案中，卸载描述符 220 包括标头字段 222（用于一种实施方案的图 15 中分解图中所示）、非必需的结果保留字段 224、以及一个或更多的缓冲器指针字段 226A ~ 226N。每个缓冲器指针字段 226A ~ 226N 包括可以编码了缓冲器尺寸的尺寸字段、以及编码了缓冲器指针的指针字段。另外，在该实施方案中，每个缓冲器指针字段 226A ~ 226N 包括来源/目的(S/D)字段，其将指针标识为来源指针（定位存储着源 DMA 数据的缓冲器）或者目的指针（定位要被存储 DMA 数据的缓冲器）。Tx 控制电路 56 可以被配置为读取缓冲器指针字段 226A ~ 226N 以从来源缓冲器中预取 DMA 数据，并标识目的指针，如果有的话。如果被转换的 DMA 数据是卸载引擎 44 的结果，那么可以有用于被转换的 DMA 数据的目的指针。如果生成了与 DMA 数据分开的结果（例如，存储于结果保留字段 224 中），在一些情况中可以没有目的指针，并且 DMA 数据可以不被写入到目的。

卸载标头字段 222 的分解图包括类型字段 222A、样式字段 222B、结果保留字段 222C、加密模式字段 222D、功能序号字段 222E、逻辑块长度字段 222F、和卸载控制字段 222G。类型字段 222A 和样式字段 222B 可类似于上述的类型字段 192A 和样式字段 192B，而逻辑块长度字段 222F 可以类似于上述的逻辑块长度字段 212E。结果保留字段 222C 可以被编码以表明结果保留字段 224 是否被包括在卸载标头描述符 220 中，并且还可以表明结果保留字段 224 的尺寸（例如，在一种实施方案中，为 64 位、192 位、或 320 位）。如果结果不是被转换的 DMA 数据，或者是在被转换的 DMA 数据之外被生成的，那么结果保留字段 224 可以被用于存储由卸载引擎 44 所生成的结果。加密模式字段 222D 可以用卸载引擎 44 为处理数据需要的模式进行编码，如果有的话。加密模式可以包括无操作、只签名（例如生成器 86 或 88 中的 CRC 或校验和）、只加密、只解密、加密并且散列（有先

散列或先加密的选项)、或者解密并且散列(有先散列或先解密的选项)。功能序号字段 222E 可以用那些使用安全电路 84A~84D 的模式(例如上述由安全电路 84A~84D 所组成的 8 种逻辑功能其中之一)的功能序号进行编码。卸载控制字段 222G 可以包括用于 DMA 传输的附加控制信息。例如,每个签名标头、密码标头、密码尾部的长度,并且散列尺寸可以被包括在卸载控制字段 222G 中。同样,所选的加密/解密(密码)算法、散列算法、和分组密码模式可以被编码于卸载控制字段 222G 中。卸载控制字段 222G 中还可以包括各种其它的控制位。

图 16 是控制描述符 230 的一种实施方案的框图。控制描述符 230 可以是 Tx 控制电路 56 在任意 DMA 通道中所用的控制描述符的格式。在图 16 的实施方案中,控制描述符 230 包括标头字段 232(用于一种实施方案的图 16 分解图中所示)、和数据或尺寸/指针字段 234。数据或尺寸/指针字段 234 可以存储要被写入配置寄存器 38A-38G 的内嵌数据(inline data)、或者可以指向存储一系列寄存器地址和要被写入寄存器的配置数据的缓冲器的存储缓冲器指针(和缓冲器尺寸)。

控制标头字段 232 的分解图包括类型字段 232A、内嵌/指针字段 232B、接口/功能 ID 字段 232C、和寄存器地址字段 232D。类型字段 232A 可以类似于上述的类型字段 192A,将描述符 230 标识为控制描述符。内嵌/指针字段 232B 可以被编码以表明数据或尺寸/指针字段 234 是否存储着要被写入配置寄存器的内嵌数据、指向寄存器地址和配置数据缓冲器的缓冲器指针、或事件。接口/功能 ID 字段 232C 被编码以标识配置数据的目标(例如接口电路、卸载引擎 44 中的功能等)。在本实施方案中,寄存器地址可以属于(local to)接口/功能 ID。如果控制描述符 230 有内嵌数据,那么寄存器地址字段 232D 可以用寄存器地址编码。

现在转到图 17,显示了校验和生成器 88 的一种实施方案的框图。图 17 的一种实施方案中,校验和生成器 88 包括多个的 16 位 3:2 进位保留加法器(CSA)240A~240G、全加器 242、和 16 位累加器(ACC)

244。校验和生成器 88 被耦合以接收 128 位 (16 字节) 输入 (例如从 Tx 控制电路 56)。128 位输入被划分成 16 位部分以作为输入送到 CSA 240A ~ 240C。另外,累加器 244 的输出被作为输入耦合到 CSA 240C。每个 CSA 240A ~ 240C 输出和以及进位项,在图 17 中标为“S”和“C”。CSA 240A ~ 240C 的和以及进位输出被输入到 CSA 240D 和 240E。CSA 240D 的和以及进位输出与 CSA 240E 的进位输出被输入到 CSA 240F。CSA 240E 的和输出与 CSA 240F 的和以及进位输出被输入到 CSA 240G。CSA 240G 的和以及进位输出被输入到全加器 242,以产生要被存储在累加器 244 中的和。

CSA 接收 N 位输入并产生和以及进位输出项而不会将进位项逐位地行波传递 (ripple)。即,和输出位是该位位置上的输入位的和,而与其它位位置上的位无关。每个和位可以是该位位置上输入位的 XOR。进位输出位是来自给定位位置求和的进位,与其它位位置上的进位无关。进位项在算术上被视为下一个最高有效位位置的进位输入。因此,最低有效位位置上的进位位为零,并且 N 位 CSA 在逻辑上有 N + 1 位进位位。

如箭头 246 所示,在每个点处,来自 CSA 的进位项输出被作为输入提供给另一 CSA 或全加器 242,最高有效进位位被“绕回 (wrap around)”到进位项的最低有效位。即,进位项的最高有效位被从最高有效位位置上删除并被插入在最低有效位位置,替换总是零的最低有效位。按本方式,每个 CSA 和全加器 242 的输入总是 16 位,并且所有的进位位都出现在输入上。对于图 17 中所示的 16 位 CSA,进位项输出在逻辑上可以是具有最低有效位 0 的 17 位值。进位项输出是下一个 CSA (或全加器 242) 的输入,如下所示:

$$\text{In}[15:0] = \text{Carry}[15:1] \parallel \text{Carry}[16] \quad (\text{并且 } \text{Carry}[0] = 0 \text{ 被丢弃})$$

最高有效进位位到最低有效进位位的绕回可以通过一个 CSA 的进位项输出与下一个 CSA 的输入之间的布线路径来完成。

在一些实施方案中,如图 17 中所示的校验和生成器 88 可以一遍 (one pass) 生成 16 位校验和,将校验和存储在累加器 244 中。在本实施方案中,因为在每个 128 位输入累加在校验和中时,进位已被绕回并被 CSA 240A ~ 240G 和全加器 242 相加,所以累加器 244 也可以是 16 位。

对于校验和生成,要生成 1 的反码和 (one's complement sum)。将 CSA 输出的进位位从最高有效位绕回到最低有效位能够生成正确的反码和。

CSA 240A ~ 240G 可以被视为 CSA 级的集合。第一级 CSA (CSA 240A ~ -240C) 具有被耦合到校验和生成器输入的输入和被耦合到累加器输出的输入。在其它级上,输入被耦合到前一级 (或几级,例如在 CSA 240G 的情况中) CSA 的输出。在每一级上,来自前级的进位输出的最高有效位绕回到要输入到该级的最低有效位。

本实施方案可以将 16 位 CSA 用于产生 TCP、IP 和 ICMP 中所用的 16 位校验和。其它实施方案可以使用更大或更小的校验和,并且在这样的实施方案中可以使用更大或更小的 CSA。这样,总的来说,可以使用 N 位 3:2 CSA。同样,虽然所示实施方案中使用 128 位输入,但是其它实施方案可以使用任意位宽的输入。CSA 的级数可以据此而改变。

虽然图 17 的实施方案显示为在校验和生成器 88 中被实现,但是类似电路可以在 IOB 22 中被用于部分校验和生成 (如前所述) 以及在接口电路 16 中以完成部分校验和。

现在转到图 18,显示了全加器 242 的一种实施方案的框图。在所示实施方案中,全加器 242 可以包括 XOR 电路 250 和一对进位生成电路 252A ~ 252B。XOR 电路 250 和进位生成电路 252A ~ 252B 被耦合以接收全加器的输入“a”和“b” (为图 17 中 CSA 240G 的和以及进位输出,其中进位输出最高有效位被移除并插入在最低有效位处)。进位生成器 252A 被耦合以接收为 0 的进位入 (C_{in}) 输入,而进位生成器 252B 被耦合以接收为 1 的进位入输入。进位生成器 252A 的进位

出 (C_{out}) 输出被作为多路选择器 (mux) 254 的选择控制, 该 mux 254 以进位生成器 252A ~ 252B 的进位输出 (carry output) 作为输入。mux 254 的输出被耦合作为第二 XOR 电路 256 的输入, 该第二 XOR 电路 256 也将 XOR 电路 250 的输出作为输入。XOR 电路 256 的输出是全加器 242 的输出。

XOR 电路 250 将输入 a 和 b 按位异或, 有效地执行另一个与 CSA 加法器 240 类似的和项生成。进位生成器 252A ~ 252B 对输入 a 和 b 进行真进位生成 (true carry generation), 并且如果进位生成器 252A 的进位出 (carry out) 是零, 那么进位生成器 252A 的输出被 mux 254 选择并与来自 XOR 电路 250 的和项进行 XOR, 由此产生最终的反码和。另一方面, 如果进位生成器 252A 的进位出 (carry out) 是 1, 那么进位生成器 252B 的输出被 mux254 选择。由于生成器 252B 的进位入 (carry in) 是 1, 进位生成器 252B 已经有效地将来自最高有效位的进位输出再次绕回。以另一方式来看, 进位生成器 252A ~ 252B 和 mux 254 可以基于进位是否产生而有选择地将又一个进位位绕回。

一旦上述公开被完全理解, 对于本领域技术人员而言, 很多变形和修改将变得明显。要注意的是所附的权利要求应被解释为涵盖所有这样的变形和修改。

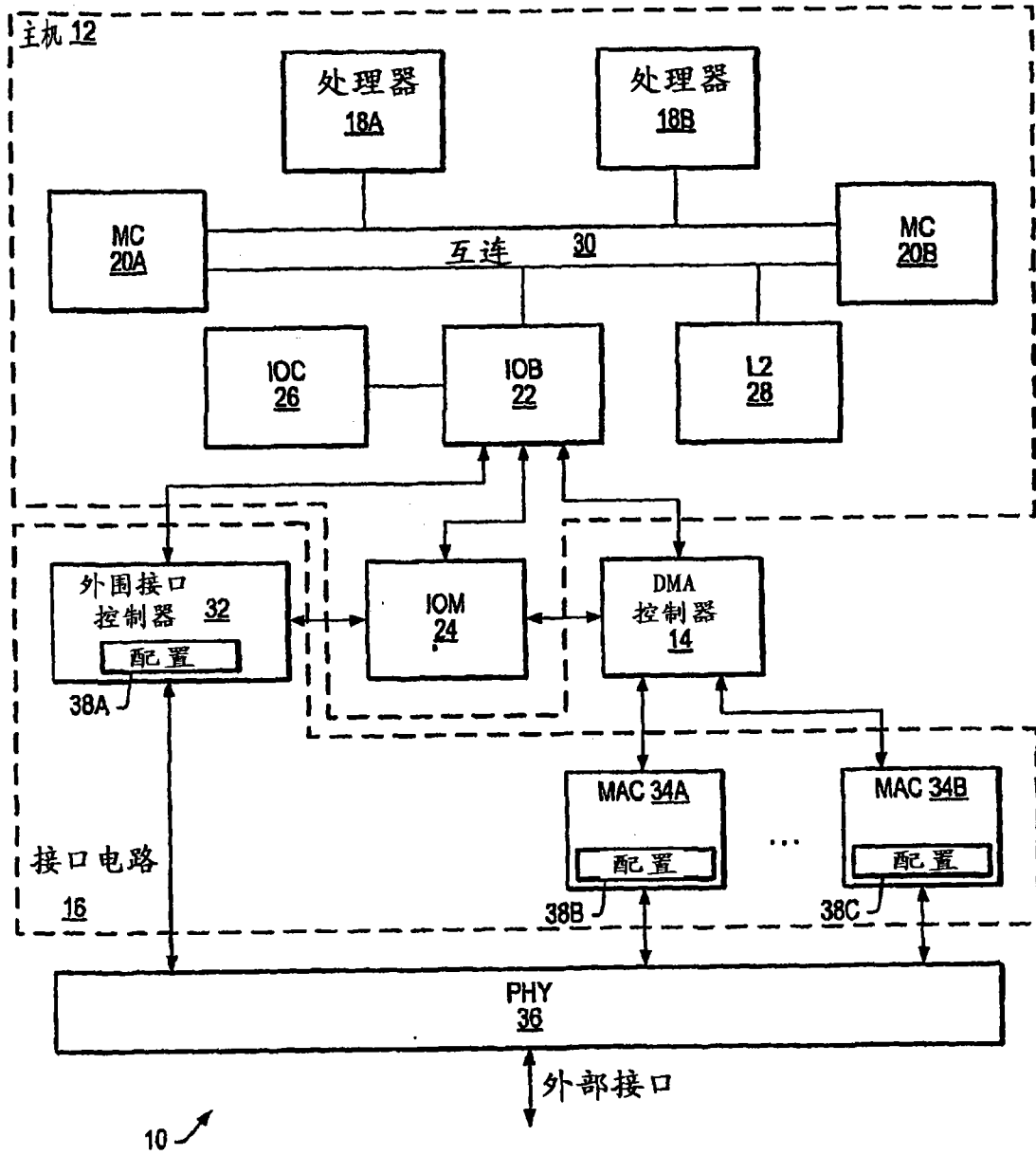


图 1

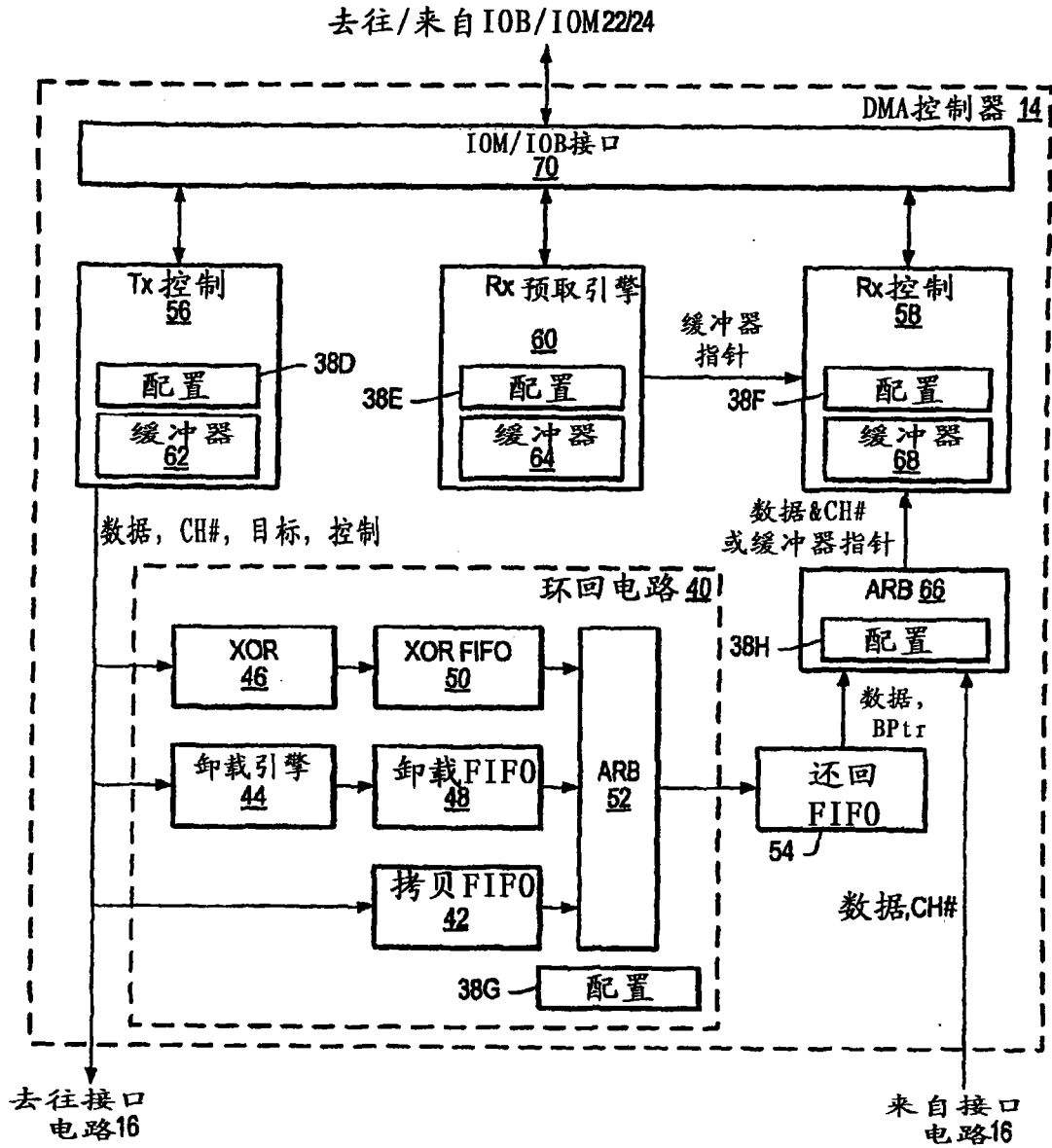


图2

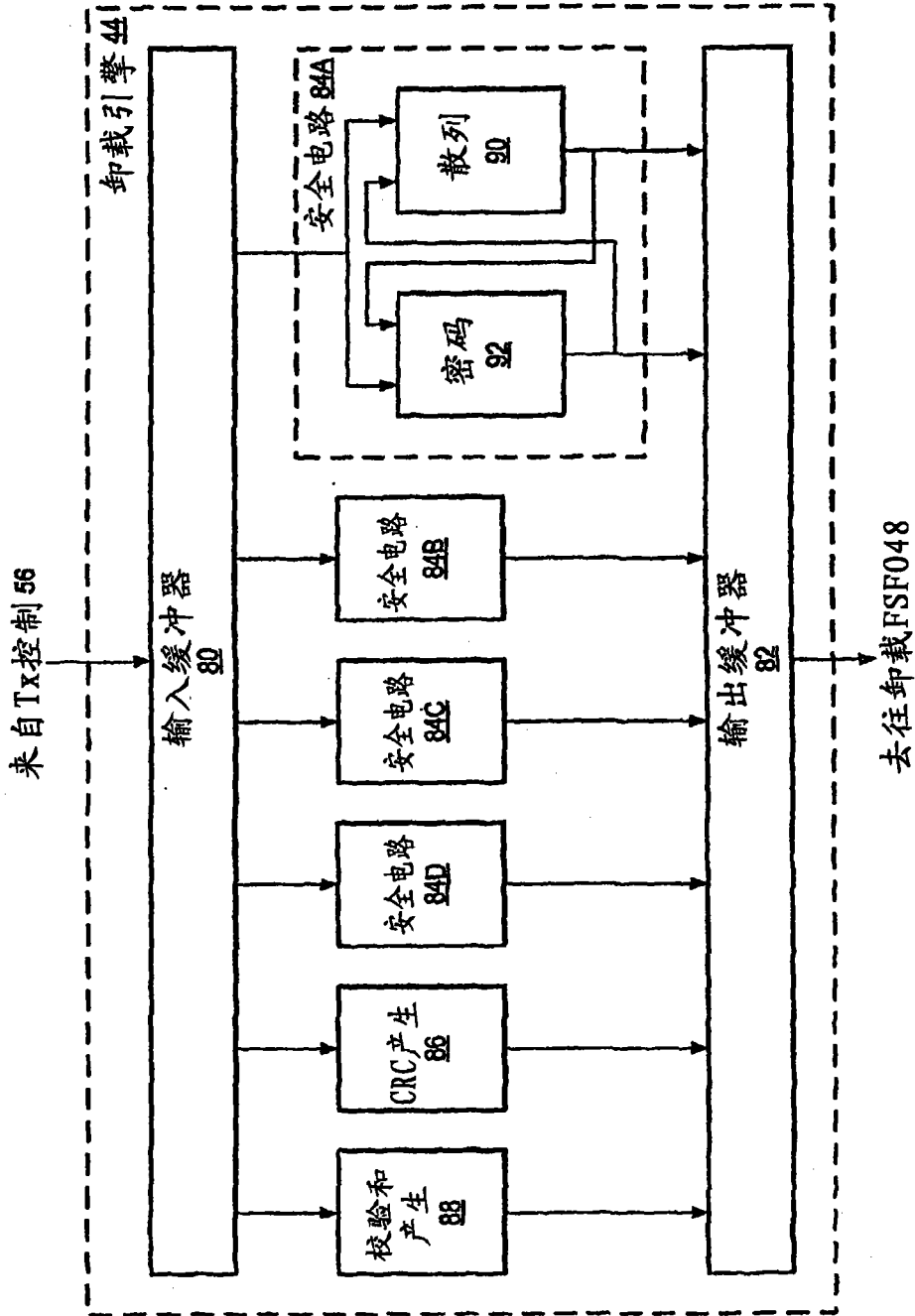


图 3

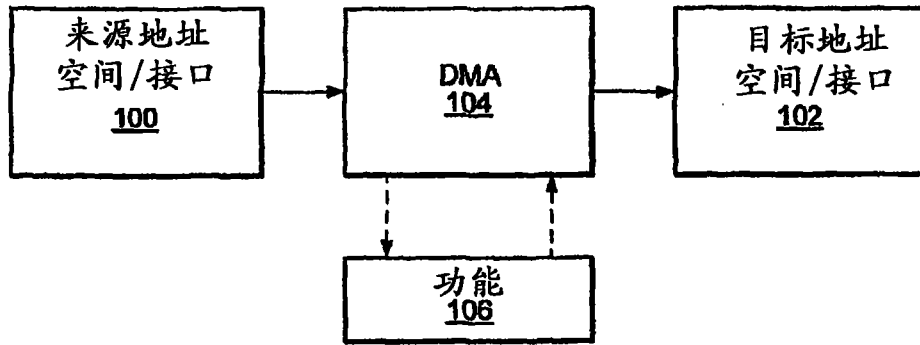


图 4

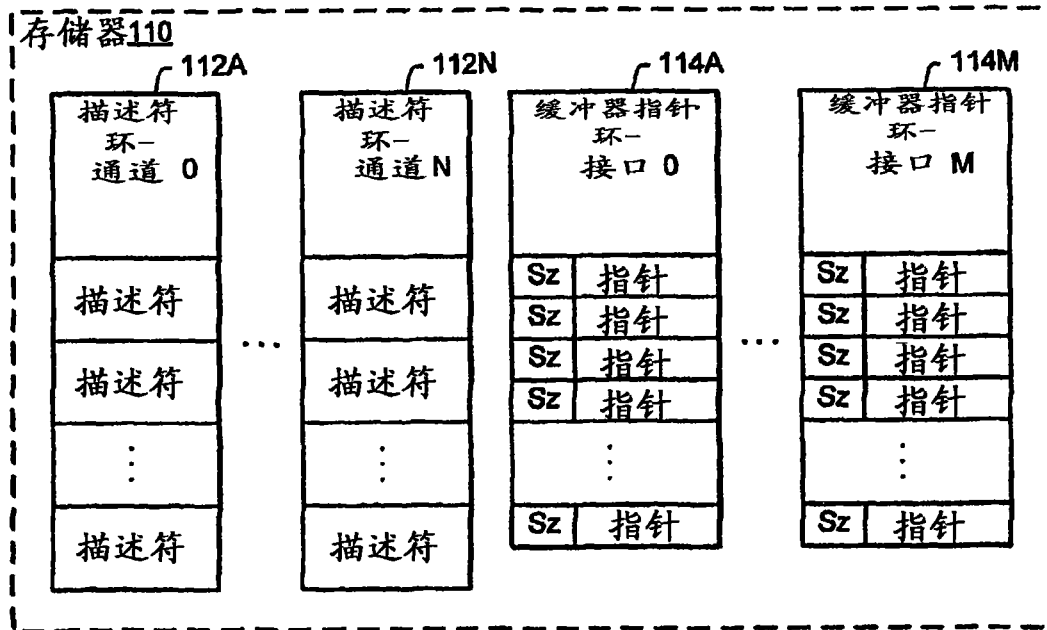


图 5

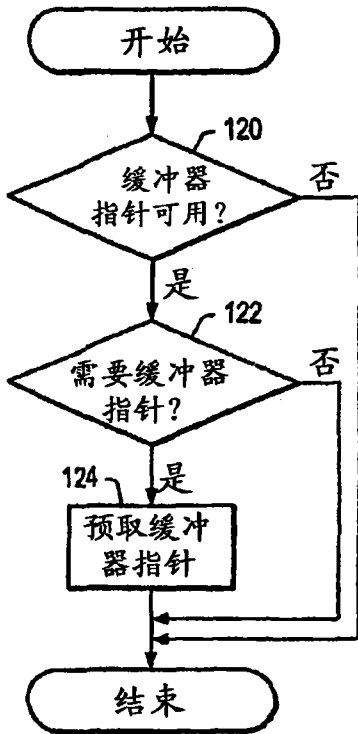


图6

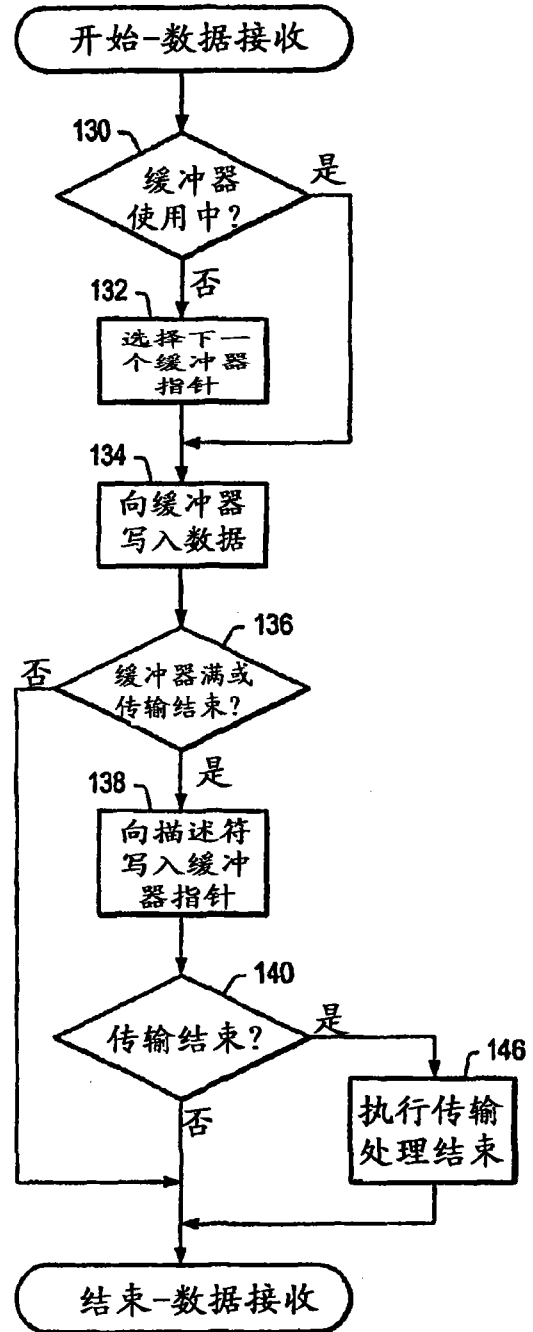


图7

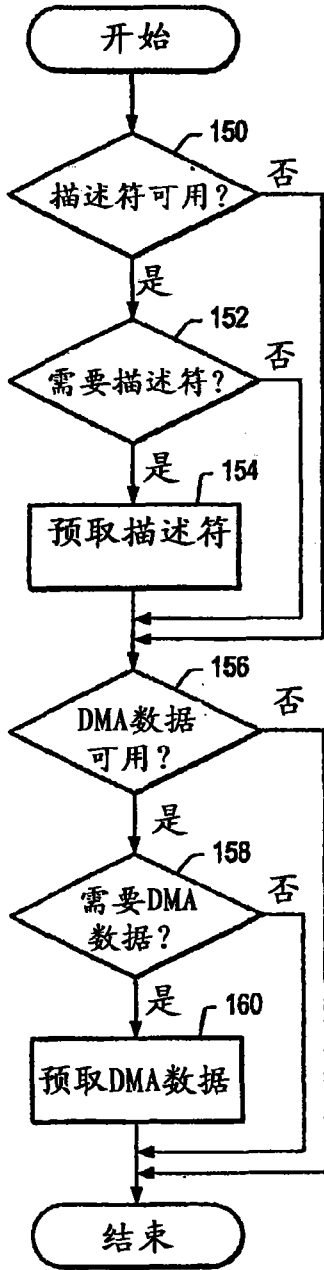


图8

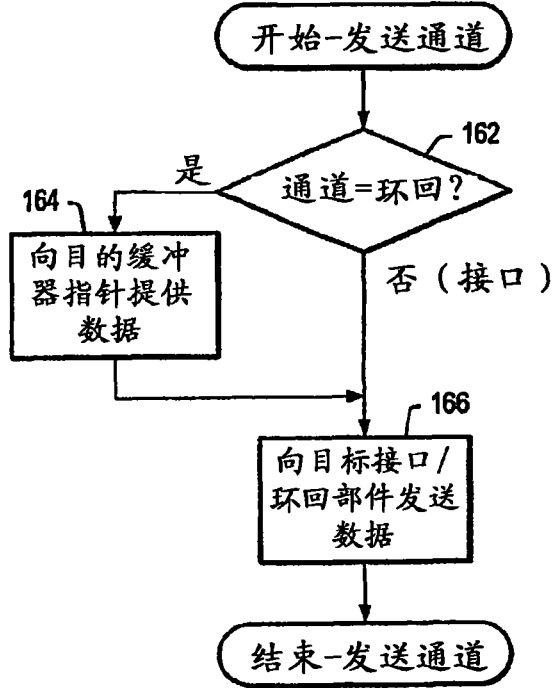


图9

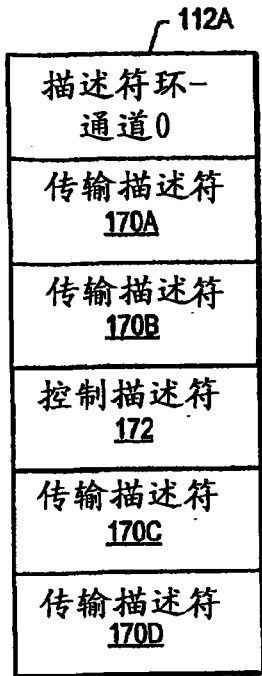


图 10

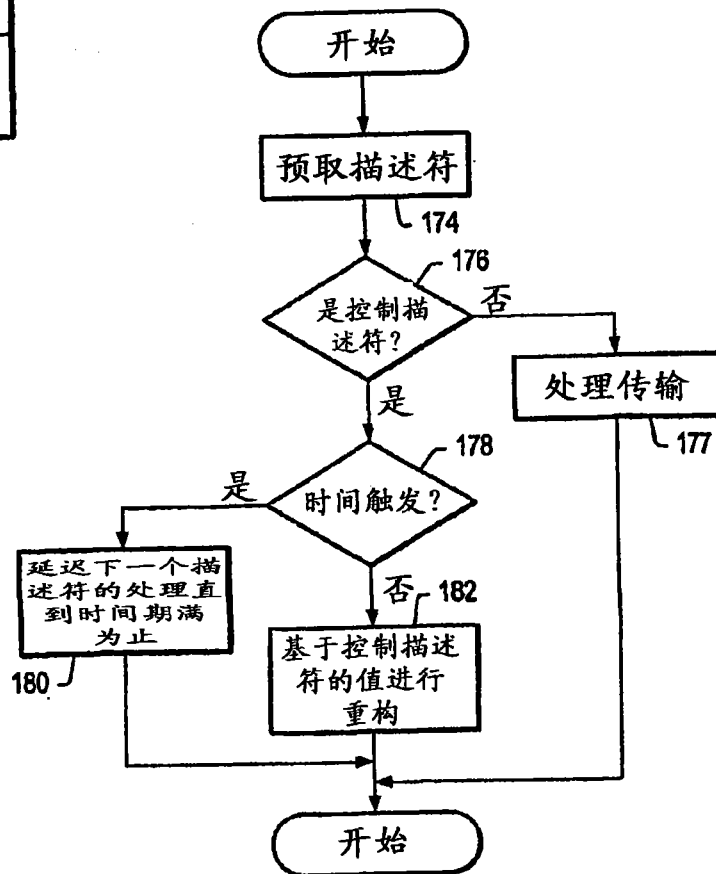


图 11

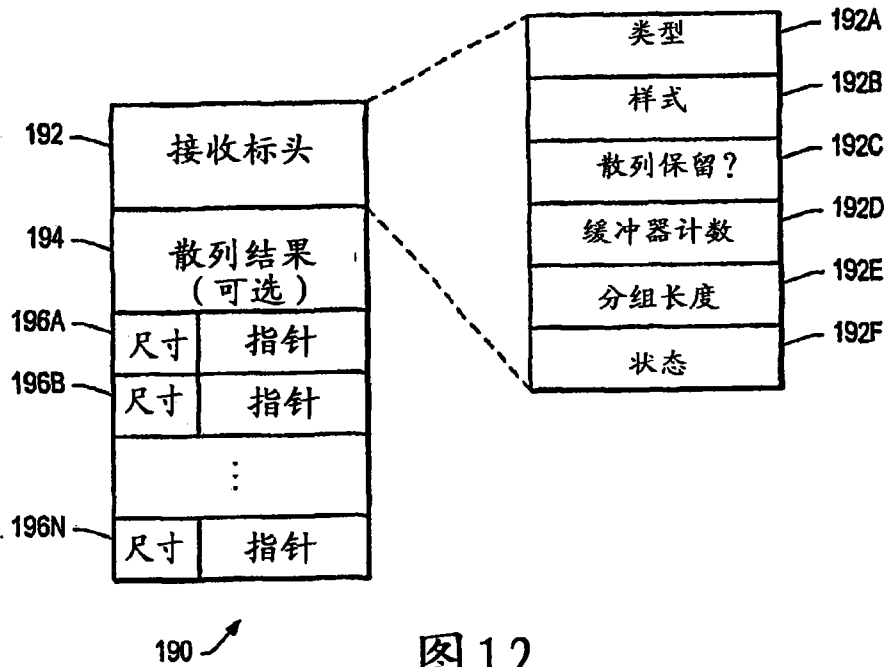


图 12

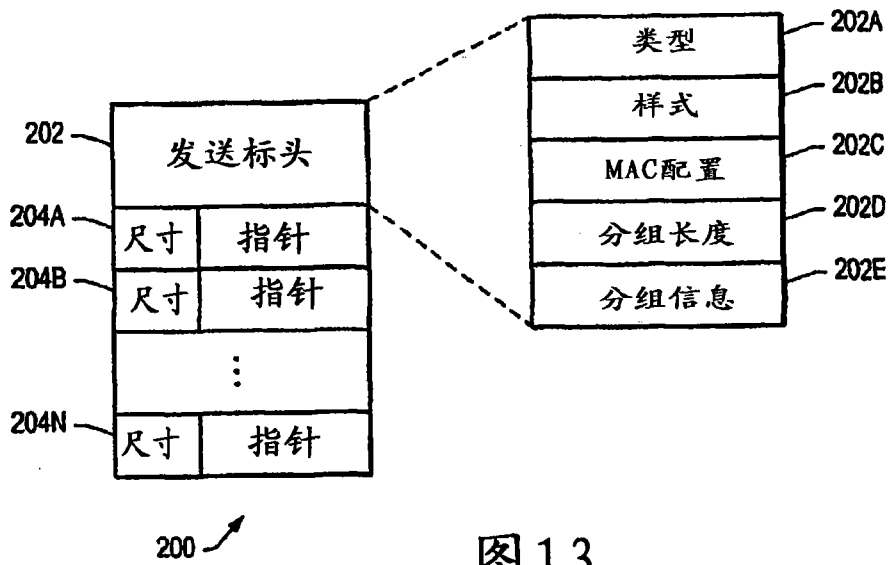


图 13

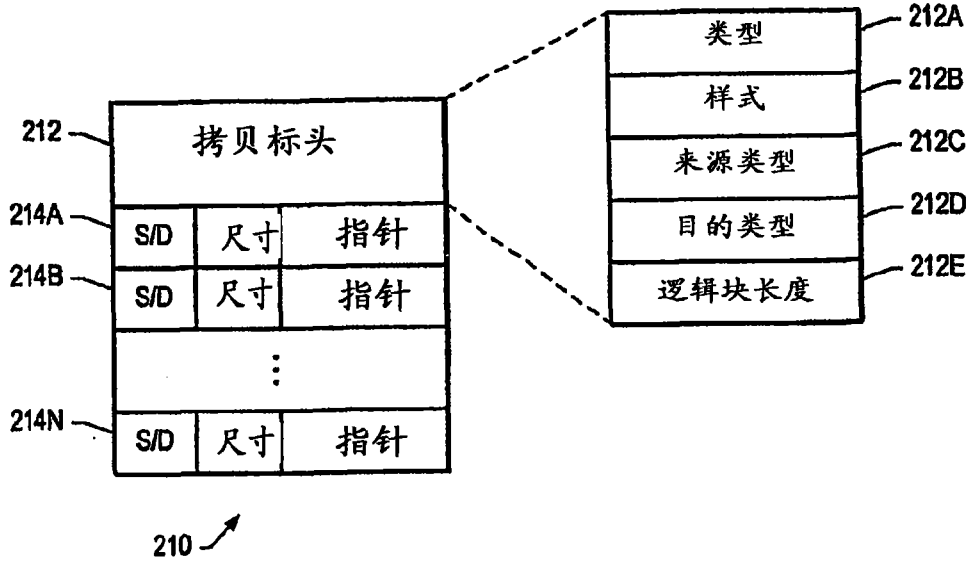


图 14

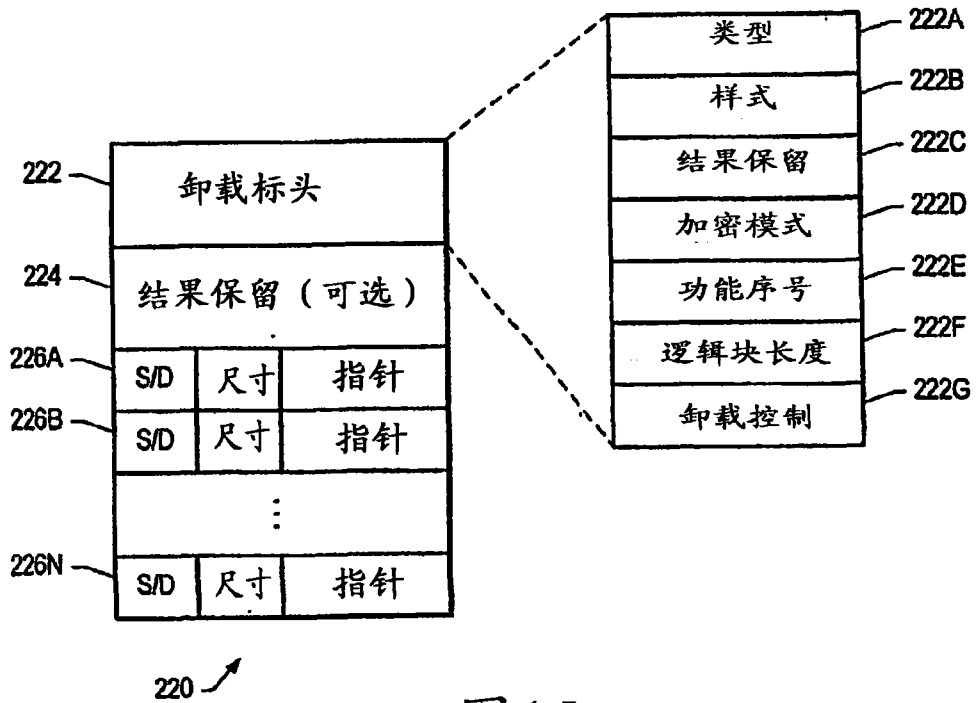
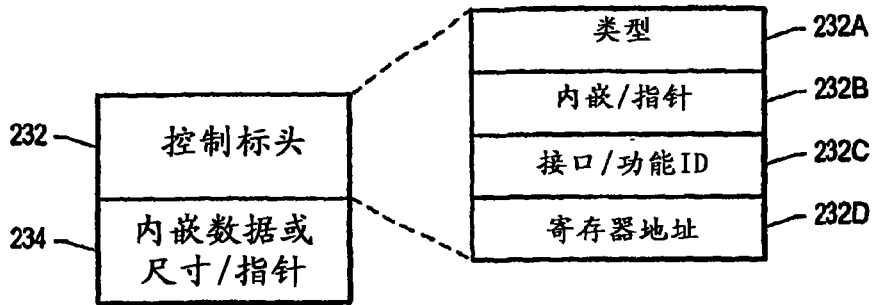


图 15



230 ↗

图 16

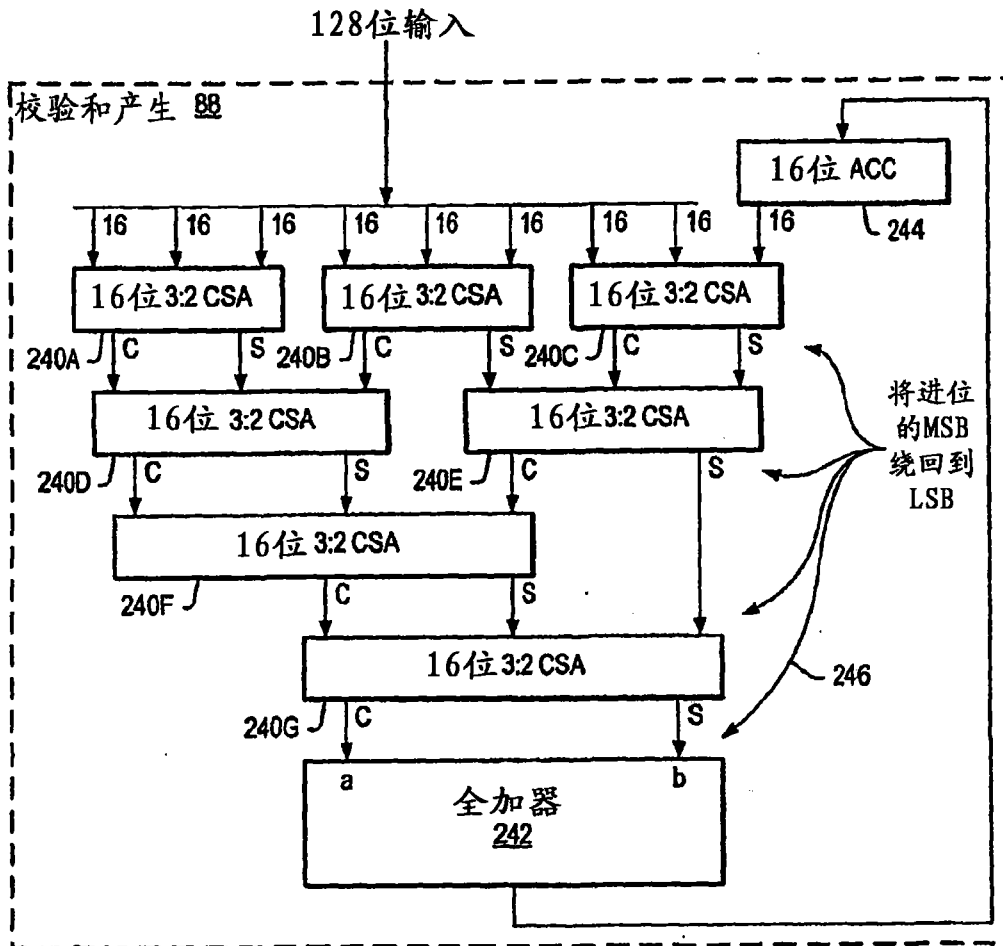


图 17

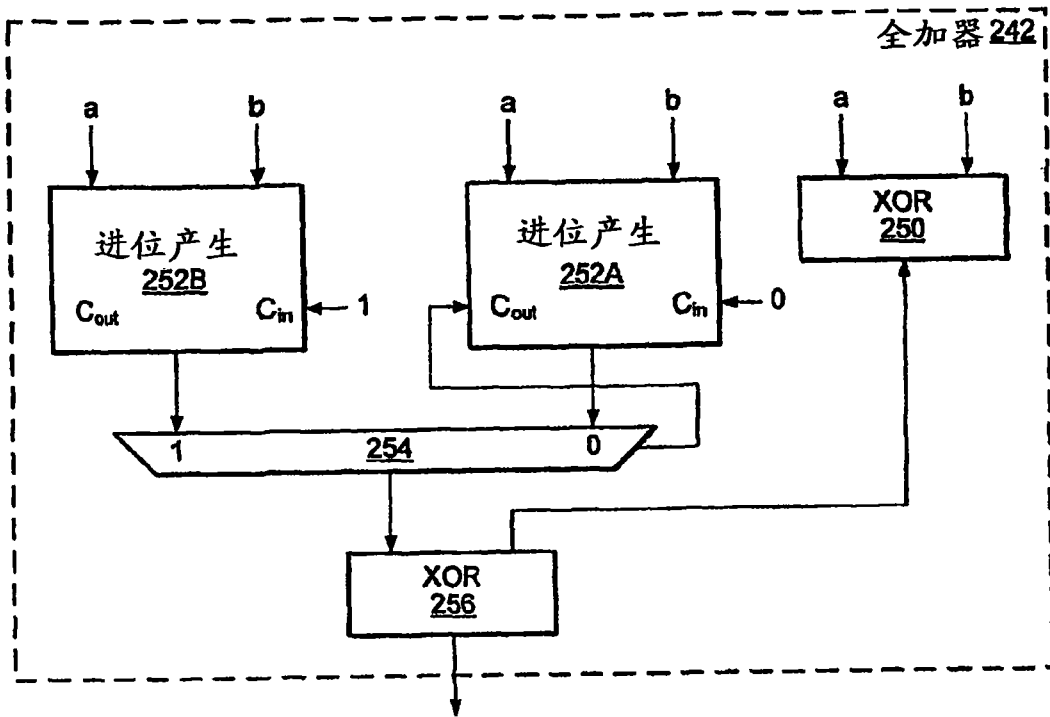


图 18