第33卷 第20期 计 算 机 工 程 2007年10月
Vol.33 No.20 Computer Engineering October 2007
·网络与通信· 文章编号：1000—3428(2007)20—0099—02 文献标识码：A 中图分类号：TP393

高速网络接口卡DMA引擎的设计与实现

戴 斌，胡晓峰，孙志刚，卢泽新
（国防科技大学计算机学院，长沙 410073）

摘 要：随着DMA技术在高速网络接口卡设计中的广泛应用，DMA引擎已成为高速网络接口卡的一个重要组成部分。该文提出了一种面向高速网络接口卡的DMA引擎的实现方案，并将之成功地应用于千兆位以太网网络接口卡的具体实现中。

关键词：DMA引擎；高速网络接口卡；中断机制；FPGA

Design and Implementation of DMA Engine in High Speed Network Interface Card

DAI Bin, HU Xiao-feng, SUN Zhi-gang, LU Ze-xin
(School of Computer, National University of Defense Technology, Changsha 410073)

【Abstract】 As direct access to memory(DMA) is widely used in the high speed network interface card (NIC) design, the DMA engine has become one important component of the high speed NIC. This paper presents the implementation method of the DMA engine in the high speed NIC. This method is used successfully in the 1 000Mb/s Ethernet NIC.

【Key words】 direct access to memory (DMA) engine; high speed network interface card(NIC); interrupt mechanism; FPGA

随着Internet链路速率的迅速提高，高速网络接口卡(简称网卡)在高性能服务器中得到了广泛应用[1~2]。为提高I/O效率，减轻服务器CPU负担，高速网卡常采用DMA(direct access to memory)方式与主机进行通信[3]。DMA引擎是实现DMA工作方式的功能部件，它负责在主机与网卡之间交互数据，并产生数据接收和发送中断。DMA引擎是提高网卡和主机间通信效率的重要组成部分。

本文针对高速网卡的通信需求，设计了基于描述符机制的 DMA 引擎。它的关键性能参数可由用户根据网络应用环境进行配置。实际测试结果表明， DMA 引擎的千兆位以太网网卡性能达到预想的要求。

1 DMA引擎的设计

DMA 引擎采用 DMA 方式直接向主机内存读写数据，有效地降低了主机 CPU 的工作负载。DMA 引擎的性能参数可由用户配置。这些性能参数包括：发送描述符个数，接收描述符个数，接收中断阈值，发送中断阈值，接收中断超时阈值，发送中断超时阈值(定义见 1.3 节)。用户通过软件修改相应性能参数就可以方便改变 DMA 引擎中各个模块的行为。

1.1 DMA引擎的结构

DMA 引擎的结构如图 1 所示，包括报文发送模块(packet transmit module)、报文接收模块(packet receive module)、寄存器管理模块(register management module)、描述符管理模块(descriptor management module)和中断管理模块(interrupt management module)。

报文发送模块的主要任务是根据到达 DMA 引擎的发送描述符通过 DMA 方式读取发送报文；报文接收模块的主要任务是根据接收描述符将到达 DMA 引擎的接收报文通过 DMA 方式写到主机内存；寄存器管理模块的主要任务是支持软件对 DMA 引擎的配置；描述符管理模块的主要任务是接收描述符和发送描述符的队列管理；中断管理模块的主要任务是产生接收和发送中断。其中，描述符管理模块和中断管理模块是 DMA 引擎的关键模块。

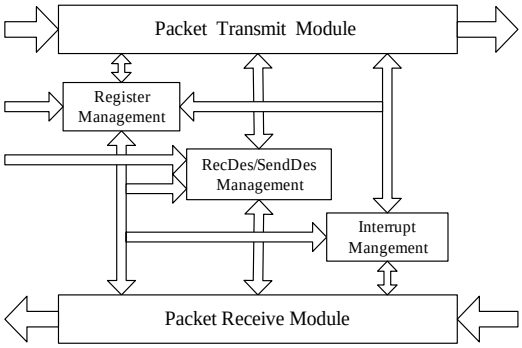

图 1 DMA 引擎结构

1.2 描述符机制

描述符机制是 DMA 引擎的重要支撑机制，描述符指定用于 DMA 操作的主机内存区域的地址。DMA 引擎包含接收描述符队列和发送描述符队列。接收报文时，网卡从接收描述符队列选择接收描述符，确定保存报文的内存地址。发送数据时，网卡根据发送描述符确定发送数据的内存地址。

通常采用数组或链表方式实现描述符队列。选择数组结构设计描述符队列，各数组元素组织成环形结构，以方便描述符的回收以及驱动程序和 DMA 引擎之间的协同。

接收描述符结构包含下列域：报文长度域，回写标志域

作者简介：戴 斌(1982－)，男，硕士研究生，主研方向：计算机网络，路由器体系结构；胡晓峰，博士；孙志刚，副研究员；卢泽新，研究员

收稿日期：2006-11-02 **E-mail：**liuleinaner@163.com

和地址域。发送描述符结构包含下列域：报文长度域和地址域。报文长度域用来指示接收或者发送报文的长度；地址域用来指示接收或者发送报文在主机内存中的地址；回写标志域用来指示描述符回写是否完成。可以看到，发送描述符不包含回写标志域，其原因将在后面进行阐述。

1.3 中断机制

中断管理模块负责向 CPU 通知网卡中已发生的事件，例如报文发送中断和报文接收中断。中断管理模块的工作机制对网卡性能有很大影响，为此，采用中断阈值与超时相结合的高效的中断机制。首先，DMA 引擎在接收或发送一定数量的报文后，再置接收或发送中断，从而减少报文接收中断或发送中断的次数。这个固定的报文数量被定义成接收/发送中断阈值。为了避免因为网络流量过低导致较长的网络延迟，DMA 引擎还提供了超时机制。以报文接收为例，假设接收中断阈值为 *s*，如果在接收到 *s* 个报文前，接收定时器超时，DMA 引擎立即产生接收中断。中断阈值与超时相结合的中断机制能够极大地减少网卡产生的中断次数，同时将报文延时控制在较小的范围内。但是，中断阈值和超时值的设置对网卡性能影响很大，如何选择合适的值是一个需要进一步深入研究的问题。

1.4 DMA 引擎数据报文接收过程

数据报文的接收过程是一个软硬件交互过程，如图 2 所示。

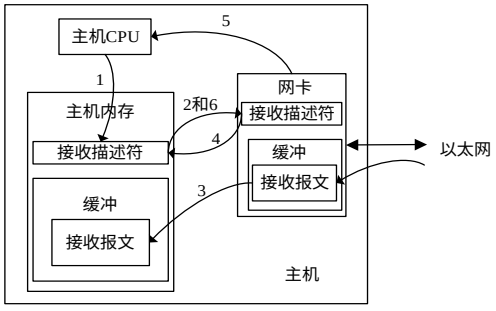

图 2 数据接收过程

步骤 1 驱动程序初始化所有可用的接收描述符。然后，驱动程序通告网卡接收描述符在内存的起始地址以及可用接收描述符的个数；

步骤 2 DMA 引擎根据接收描述符的起始地址信息和其他相关信息通过 DMA 方式将所有可用的接收描述符下载到 DMA 引擎；

步骤 3 当有数据报文到达网卡时，DMA 引擎就根据接收描述符指示的地址发动一次 DMA 操作，将到达的数据报文写入主机内存；

步骤 4 DMA 操作完成后，DMA 引擎又立即发动另一次 DMA 操作，回写刚才已使用的接收描述符中的若干域(报文长度域，回写标志域)；

步骤 5 当接收报文的数量达到接收中断的阈值时，DMA 引擎通告主机接收中断，从而主机处理接收数据并进行接收描述符的回收(当接收报文的数量未到接收中断的阈值但接收定时器超时的情况下，DMA 引擎亦通告主机接收中断)；

步骤 6 主机在完成接收描述符的回收后又将此次处理完成接收描述符个数通告给 DMA 引擎，DMA 引擎根据相关信息读取可用接收描述符。

1.5 DMA 引擎数据报文发送过程

数据报文的发送过程也是一个软硬件交互过程，如图 3。

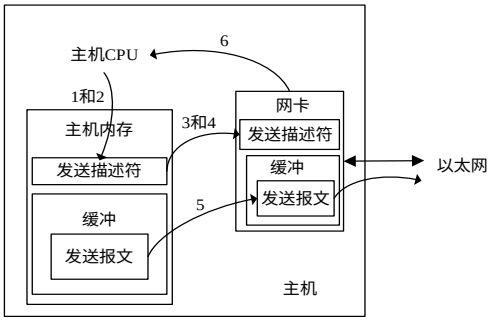

图 3 数据发送过程

步骤 1 驱动程序初始化所有可用的发送描述符，同时通告网卡发送描述符的在内存中的起始地址；

步骤 2 当有数据报文需要发送时，驱动程序将数据报文在内存中的起始地址和报文长度填写到发送描述符的相应域中；

步骤 3 发送描述符填写完成后，网卡驱动程序就通告网卡此次需要处理的发送描述符个数；

步骤 4 DMA 引擎发动一次 DMA 操作将需要处理的发送描述符下载到网卡上；

步骤 5 发送描述符的下载完成后，DMA 引擎就根据发送描述符的内容发动 DMA 操作，将发送描述符指向的数据报文发送到网络上；

步骤 6 当发送报文的数量达到发送中断阈值时，DMA 引擎通告主机发送中断，从而驱动程序开始进行发送中断的处理，并进行发送描述符的回收(当发送报文的数量未到发送中断的阈值但发送超时的情况下，DMA 引擎亦通告主机发送中断)。

随着网络链路速率的提高以及 PCI Express(PCIe)接口技术的出现，高速网卡开始采用 PCIe 作为其与主机的接口技术。后面一节中设计并实现的千兆位以太网网卡采用了 PCIe 作为其与主机的接口技术。在处理发送描述符的过程中没有涉及发送描述符的回写，这是由于考虑到 PCIe 传输可靠性的提高，发送描述符只要下载到 DMA 引擎，发送数据就几乎可以 100%正确地被发送，因此没有必要进行发送描述符的回写。这样处理带来的好处是简化了 DMA 引擎的逻辑设计并且提高了数据传输的效率。发送描述符结构定义中不包含回写标志域。

2 在千兆位网卡设计中的应用及测试结果

设计并实现了一块千兆位以太网网卡。千兆位以太网网卡有 3 个模块组成，分别为 PCIe 接口模块、DMA 引擎模块和以太网接口模块。PCIe 接口模块主要负责跟主机进行数据交互；DMA 引擎模块主要实现 DMA 引擎的功能；以太网接口模块主要负责跟物理网络进行数据交互。在 Altera Stratix II 的 FPGA 中实现了 DMA 引擎模块和 PCIe 接口模块的全部逻辑，其中 PCIe 模块的逻辑采用的是 PLDA PCIe 核。

在实际的网络环境中测试了网卡的性能。测试环境包括一台Dell PowerEdge SC1420 服务和一台浪潮英信服务器 NF130。Dell服务器配置笔者设计的千兆位以太网网卡，浪潮服务器配置基于Intel82554 设计的千兆位网卡，二者均采用 Linux 2.6 内核，测试工具为Iperf[4]。

网卡的性能参数配置如下：接收和发送描述符个数为 32，接收中断阈值设为 7，发送中断阈值设为 11，接收和发

(下转第 103 页)