高技术通讯 2013 年 第 23 卷 第 7 期:685 ~ 693

doi:10.3772/j.issn.1002-0470.2013.07.004

多微通道内存系统设计方法①

张广飞②[*,**,***] 王焕东[*] 陈新科[*,**,***] 黄 帅[*] 陈李维[*,**,***]

([*]中国科学院计算机系统结构重点实验室 北京 100190)

([**]中国科学院计算技术研究所 北京 100190)

([***]中国科学院研究生院 北京 100049)

摘 要 通过建立内存系统排队模型,分析了影响内存系统性能的原因——内存控制器的内存命令处理速度受访存请求页命中率、Bank 级并行度和读写命令切换率的影响,进而提出了一种多微通道内存系统设计方法。用此方法多微通道内存控制器通过对内存颗粒进行细粒度控制,可以提高访存请求页命中率和 Bank 级并行度,隐藏数据总线读写切换延迟。该结构在提高内存系统带宽利用率的同时,缩短访存请求延迟,并提高内存功耗有效性。将多微通道内存控制器设计应用于多核处理器平台,充分分析各种宽度访存通道对应用程序性能的影响。实验结果表明,相比传统内存控制器设计方法,多微通道内存控制器将内存系统带宽提高了 21.8%,访存延迟和功耗分别降低 14.4% 和 26.2%。

关键词 DRAM 系统,内存控制器,片上多核,多通道,访存特性

0 引 言

片上多核处理器设计已经成为现代微处理器技术发展的趋势,日益增长的处理器性能对内存系统性能提出了更高的要求。随着内存性能的提高和容量的增加,内存系统功耗问题日益严重,内存系统设计成为当今体系结构领域研究的热点问题[1]。近年来,内存芯片峰值带宽增长迅速,但内存系统带宽利用率却相对较低。制约内存系统性能的瓶颈是内存控制器的内存命令处理速度,而影响内存命令处理速度的关键因子是访存请求页命中率、Bank 级并行度和读写命令切换率。传统内存控制器以提高内存系统频率和数据总线宽度为首要设计目标,忽略了对上述三个关键因子的优化。为了分析这三个因子对内存系统性能的影响,本研究对内存系统进行了建模分析,并根据分析结论提出了一种多微通道内存系统设计方法。该方法提高了内存系统页命中率和 Bank 级并行度,隐藏了数据总线读写切换延迟,而且在提高内存系统性能的同时,降低了内存系统功耗开销。

1 内存系统特性分析

内存系统主要由内存控制器和内存芯片组成。内存芯片对不同内存命令之间的时间间隔有着严格的规定,但是总的来讲,内存芯片有三个主要特性决定了内存系统的整体性能。

1.1 页命中率

内存芯片主要由多个内存颗粒组成,而每个内存颗粒通常包含多个 Bank,每个 Bank 由多个行(也叫做页)组成。例如 DDR3 SDRAM 颗粒由 8 个 Bank 组成。如图 1 所示,访存请求对内存颗粒的读写操作必须在相应 Bank 的行缓存中进行[1]。

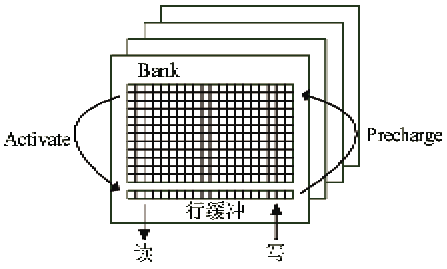

图 1 内存颗粒

① 国家"核高基"科技重大专项课题(2009ZX01028-002-003, 2009ZX01029-001-003)和国家自然科学基金(60921002, 61003064)资助项目。

② 男,1984 年生,博士生;研究方向:计算机体系结构;联系人,E-mail: guangfeizhang@hotmail.com

(收稿日期:2012-06-04)

一次完整的访存请求主要需要三类访存命令：(1)Precharge，对内存颗粒内部存储单元充放电，为后续访存命令做准备；(2)Activate，将要读写的页读入行缓存；(3)Read/Write，在行缓存中对数据进行读写。如果访存请求要访问的页已经在行缓存中，则该访存请求称为页命中，否则为页冲突。页命中的访存请求可以直接在页缓存中进行读写操作，否则需要进行上述三个步骤。页命中率是指发生页命中的访存请求数量占访存请求总数的比值。不同应用的访存请求页命中率不同。页命中率越高，内存系统带宽利用率越高。

1.2 Bank 级并行

为提高系统带宽利用率，内存芯片允许内存控制器向内存颗粒不同 Bank 先后发射不同的访存命令。内存芯片可流水控制处理这些访存命令，隐藏访存命令延迟。对访存命令的并行处理称为 Bank 级并行。Bank 级并行有利于提高内存系统性能。

1.3 数据总线切换率

读写访存命令共用一组数据总线。当数据总线发生读写反向时，数据总线需要经历数据总线恢复延迟。数据总线反向率是指数据总线反向次数与访存请求数量的比值。数据总线反向率越低，系统带宽利用率越高[2]。

2 内存系统性能分析

访存延迟的大小是评价内存系统性能的主要标准[3]。为了简化叙述，本文在多核处理器系统范畴内展开论述，并假设多核处理器系统中所有的访存请求都来自最后一级缓存缺失，且每个访存请求传输的数据量相同。

如图 2 所示，访存延迟是指从最后一级缓存发出访存请求到收到访存请求响应所经历的时钟周期数。访存延迟 t_m 主要由 4 部分组成：

(1) t_{flow}，访存请求在内存控制器流水线中移动需要的时钟周期数。在本文中，t_{flow} 为 15。t_{flow} 由内存控制器的具体结构决定。

(2) t_{queue}，访存请求在访存队列的排队时钟周期数。t_{queue} 的值由访存请求到达速率和内存系统访存请求处理能力共同决定。

(3) t_{process}，内存控制器处理内存命令所需时钟周期数。其值主要由访存请求的页命中率、读写内存命令切换率和内存命令 Bank 级并行度共同决定。

(4) t_{transfer}，在内存控制器和内存芯片之间传输数据需要的时钟周期数。t_{transfer} 的值主要由内存颗粒类型决定。本文实验分析采用 DDR3 SDRAM 内存颗粒，t_{transfer} 值为 4。则访存延迟公式为

$$t_m = t_{\text{flow}} + t_{\text{queue}} + t_{\text{process}} + t_{\text{transfer}} \tag{1}$$

如图 2 所示，多核处理器内存系统的命令队列和命令处理单元为一个单服务员排队系统[4]。顾客(访存请求)到达率 λ 由多核处理器最后一级缓存缺失速率决定。该排队系统对访存请求的服务速率 μ 取决于访存命令处理时间 t_{process}，即

$$\mu = 1/t_{\text{process}} \tag{2}$$

内存系统的访存强度 ρ 主要由访存请求到达速率和访存请求服务速率共同决定，即

$$\rho = \lambda/\mu \tag{3}$$

根据排队论，若 $\rho < 1$，内存系统处于稳定状态。由 Little 定理可知，若内存系统处于稳定状态，命令队列中访存请求的平均排队时间取决于访存请求达到速率和访存请求服务速率，即

$$t_{\text{queue}} = \lambda^{-1} \times \frac{\rho}{1-\rho} = \frac{1}{\mu \times (1-\rho)} \tag{4}$$

经上述分析可知，若内存控制器流水级数和内存芯片参数不变，访存延迟 t_m 取决于访存请求到达速率和访存请求服务速率。访存请求到达速率是外部因素，访存请求处理速率 t_{process} 是制约内存系统性能的关键因素。

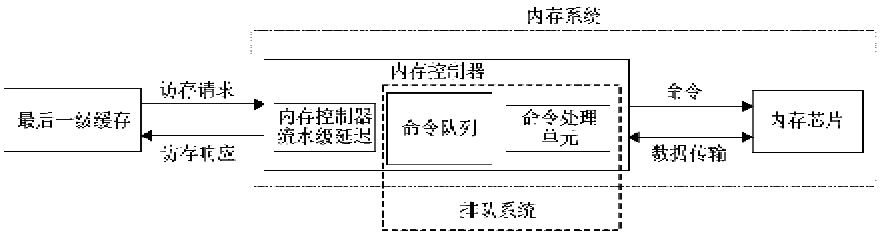

图 2 内存系统模型

Bank 级并行度是指内存系统中内存命令并行处理的程度[5]，其具体定义如下：

在时钟周期 m，有 n 条发射到 Bank_i($0 < i < N_{\text{Bank}}$，N_{Bank} 为内存颗粒 Bank 数目)的 DRAM 命令仍

在进行,则称内存系统在时钟周期 m 的 Bank 级并行度为 n。内存系统在 k 个时钟周期内的 Bank 级并行度为

$$R_{\mathrm{BLP},k}=\frac{\sum_{0}^{k-1}n_m}{\sum_{0}^{k-1}p_m}\left(p_m=\begin{cases}1, & \text{若 } n_m>0\\ 0, & \text{若 } n_m=0\end{cases}\right) \quad (5)$$

其中,n_m 是在时钟周期 m 正进行的内存命令数,P_m 是在时钟周期 m 是否有内存命令尚在进行。

由表 1 可知,根据是否页命中和引起数据总线反向,访存请求可分为 4 种类型。每种类型访存请求的延迟为该类型延迟与该类访存请求发生概率的乘积,则访存请求处理时间为

$$t_{\mathrm{process}}=(T1+T2+T3+T4)/R_{\mathrm{BLP}} \quad (6)$$

从式(6)可以看出,内存系统对访存请求的处理速率 t_{process} 主要取决于内存系统页命中率,Bank 级并行度和读写切换率。

表 1　访存类型分类

访存类型	页命中与否	读写反向	访存延迟	发生概率
$T1$	命中	不反向	t_{rw}	$R_{rhr}\cdot(1-R_{rwt})$
$T2$	命中	反向	$t_{rw}+t_{reverse}$	$R_{rhr}\cdot R_{rwt}$
$T3$	冲突	不反向	$t_p+t_a+t_{rw}$	$(1-R_{rhr})\cdot(1-R_{rwt})$
$T4$	冲突	反向	$t_p+t_a+t_{rw}+t_{reverse}$	$(1-R_{rhr})\cdot R_{rwt}$

图 3 描述了内存系统处于不同状态时访存延迟变化情况。实验采用的访存序列页命中率为 50%,数据总线切换率为 20%。该实验通过加快访存请求到达速率,分析访存强度对访存延迟的影响。

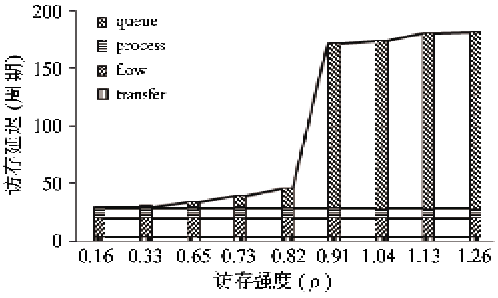

图 3　稳定和不稳定状态

如图 3 所示,当访存强度较低时,内存系统处于稳定状态,访存延迟较小。访存延迟主要由访存请求处理延迟 t_{process} 和访存请求流水延迟 t_{flow} 组成,访存请求排队延迟 t_{queue} 和访存数据传输延迟 t_{transfer} 占访存延迟比重较低。随着访存强度增加,访存延迟增长缓慢。当访存强度接近 1 时,内存系统接近不稳定状态,访存延迟快速增加。随着访存强度增加,访存请求排队延迟 t_{queue} 快速增加,并占据访存延迟主要部分,而访存数据传输延迟 t_{transfer} 可忽略不计。

3　多微通道内存控制器设计

3.1　设计思想

传统的内存 Rank 通过多个内存颗粒并联以获得更宽的数据总线,芯片状态机通过统一的地址和控制总线同时控制多个颗粒,这些颗粒同时传输相同方向的数据。DDR3 SDRAM 有 X4,X8,X16 三种宽度内存颗粒。如图 4 所示,该内存系统使用 4 个 X16 内存颗粒组成一个 64 位 Rank。

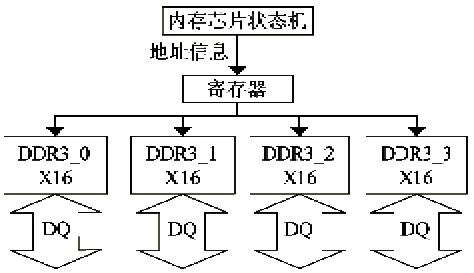

图 4　传统 Rank 组织形式

传统的 Rank 组织形式可以增加数据总线宽度,提高内存系统理论峰值带宽,缩短数据传输延迟,但也不可避免地在以下方面带来弊端:

(1) 页命中率。页命中率对内存性能起着至关重要的作用。传统 Rank 组织形式将所有颗粒的 Bank 并联起来统一管理。当多个功能单元并行访问内存系统时,有限的 Bank 数量容易造成严重的 Bank 冲突[6]。每个内存命令访问 Rank 中的所有颗粒,Bank 冲突使芯片状态机对所有内存颗粒进行页打开和关闭操作。若访存冲突严重,会造成大量功耗浪费。

(2) Bank 级并行性。当内存系统 Bank 冲突严重时,内存系统可以依靠 Bank 级并行降低页冲突带来的性能损失。每个 Rank 能够达到的 Bank 级并

行性受限于内存芯片参数 tRRD（两个连续 Activatee 内存命令之间的最小时间间隔）和 tFAW（在该时间段内，最多允许 4 个 Activate 命令发送到一个内存 Rank）。内存颗粒并联的 Rank 组织形式会限制内存系统 Bank 级并行性[7]。

（3）数据流并行性。各个内存颗粒的数据总线被并联成一条较宽的数据总线，每个时刻只能为一个数据流服务。若发生数据总线读写反向，所有颗粒的数据总线都要等数据总线稳定后才能传输数据，从而导致内存系统性能的下降[8]。

（4）命令总线利用率。若内存系统 Bank 冲突严重，由于数据总线利用率和 Bank 级并行度受到限制，内存颗粒命令总线利用率会下降[9]。

通过分析内存系统性能发现，影响内存系统性能的关键因素是内存命令处理速率 $t_{process}$，而内存命令处理速率主要取决于访存请求页命中率，Bank 级并行度和读写切换率。根据传统 Rank 组织形式的不足，针对影响内存系统性能的关键因素，本文提出细粒度的 MRank（微 Rank）组织形式。如图 5 所示，每个内存颗粒为一个 MRank，且每个 MRank 有独立的 CS（Chip Select）信号。芯片状态机通过独立的 CS 信号单独控制每个 MRank。通过细粒度颗粒控制，每个内存命令只访问一个 MRank。

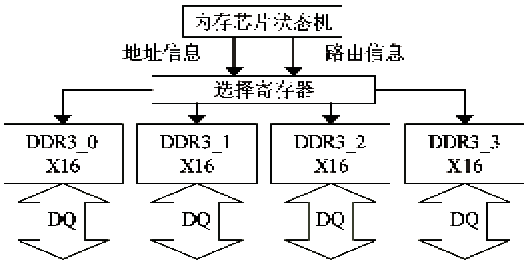

图 5 细粒度 Rank 组织形式

内存系统 MRank 数目和访存请求可同时操作的 Bank 数目都成倍增加，有利于提高页命中率和改善内存命令 Bank 级并行性。MRank 组织形式使每个 MRank 的数据独立传输，多个数据流可同时接受服务，从而提高了数据流并行性。当发生读写反向时，只有发生读写反向的 MRank 的数据总线停止传输数据，其它 MRank 的数据传输不会受到影响。MRank 组织方式起到了隐藏数据总线读写反向延迟的作用。内存命令处理延迟的下降必然会造成访存请求排队延迟的下降，从而在访存强度较高时，缩短内存系统访存延迟。

相对于传统的 Rank 组织形式，MRank 数据总线变窄，数据传输时间变长。第 2 节分析访存延迟发现，数据传输延迟占访存延迟比例很低，数据传输时间变长造成的负面影响对访存延迟影响不大。

所有内存命令都针对某个 MRank，每次 Bank 冲突只需要对该 MRank 进行页打开和关闭操作。在访存冲突严重时，不会造成严重的功耗浪费[10]。此外，多个 MRank 复用一套命令总线。当内存系统 Bank 冲突严重时，可取得较高的命令总线利用率。

3.2 结构实现

DDR3 SDRAM 有 X4，X8，X16 三种数据宽度的内存颗粒。如果假设每次访存请求访问的数据量都是 64 字节，内存系统数据总线宽度为 64 位。考虑芯片管脚数目和数据传输延迟对内存系统开销和性能的影响，多微通道内存系统使用 X16 颗粒组成多微通道内存系统。

为了对 MRank 进行细粒度控制，需要在内存控制器中为每个 MRank 增加独立的控制通道。如图 6 所示，原来 64 位的 Rank 可以分成 4 个微通道。多微通道内存控制器通过独立的命令队列对 MRank 进行控制。采用独立的访存队列分别管理每个通道可以简化逻辑设计，避免死锁和数据流停顿现象的发生。此外，独立的队列设计可以缩短队列长度，减少队列读写端口，优化物理时序设计。

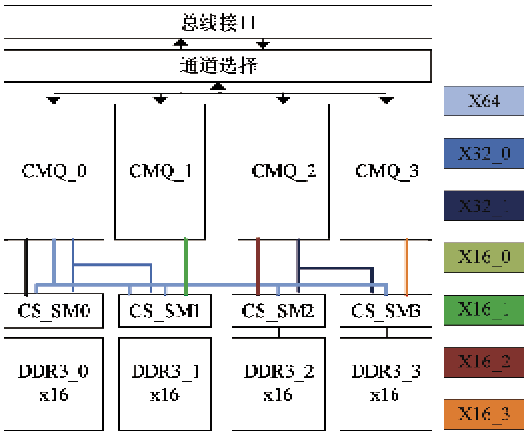

图 6 多微通道内存控制器结构图

多微通道内存控制器共有三种模式：

（1）X64 模式。4 个 X16 内存颗粒并联成 1 个 Rank，命令队列 0 同时管理这 4 个内存颗粒。

（2）X32 模式。内存颗粒 0 和 1 并联成 MRank0，内存颗粒 2 和 3 并联成 MRank1。命令队列 0 和 2 分别管理 MRank0 和 MRank1。

（3）X16 模式。内存颗粒 0，1，2 和 3 分别为 MRank0，MRank1，MRank2 和 MRank3；命令队列 0，

1,2 和 3 分别管理这 4 个 MRank。

多微通道内存控制器提供软件配置接口,应用程序可以将内存系统配置成上述任何一种模式。

多微通道内存系统完成一个访存请求需要五个步骤:

(1) 地址映射。将访存地址转换成 Rank 地址 CS,行地址 RA(Row Address),Bank 地址 BA(Bank Address)和列地址 CA(Column Address)。

(2) 通道选择。根据软件配置和第 3.3 节提出的负载均衡策略,按照访存地址将访存请求映射到对应的通道。

(3) 队列选择。多微通道内存控制器实现了多个 CMQ(Command Queue),每个 CMQ 负责不同内存颗粒的管理。CMQ _0 可以控制内存颗粒 DDR3 _0,DDR3 _1,DDR3 _2 和 DDR3 _3。CMQ _1 可以控制内存颗粒 DDR3 _1。CMQ _2 可以控制内存颗粒 DDR3 _2 和 DDR3 _3。CMQ _3 可以控制内存颗粒 DDR3 _3。

i. 若配置成 X64 模式,内存系统为一个 64 位通道。所有的访存请求进入 CMQ _0,并同时访问 4 个内存颗粒。

ii. 若配置成 X32 模式,内存系统被分成两个 32 位通道。访存请求根据地址空间配置分别进入 CMQ _0 或者 CMQ _2,并同时访问内存颗粒 DDR3 _0 和 DDR3 _1 或者 DDR3 _2 和 DDR3 _3。

iii. 若配置成 X16 模式,内存系统被分成 4 个 16 位通道。访存请求根据地址空间配置分别进入 CMQ _0,CMQ _1,CMQ _2 或者 CMQ _3,并同时分别访问内存颗粒 DDR3 _0,DDR3 _1,DDR3 _2 和 DDR3 _3。

(4) 访存调度。在每个 CMQ 内部,访存请求按照 FR-FCFS(First Ready, First Come First Serve)访存调度算法进行调度。

(5) 访问 DRAM 颗粒并返回访存结果。当内存系统处于多通道模式时,多个数据流并行访问 DRAM 颗粒。内存控制器对外只有一套接口总线。当多个数据流同时返回访存结果时,按照轮转调度方法每拍返回一个访存结果。

3.3 管理策略

负载均衡问题是内存系统设计在地址空间管理方面的难点,其具体体现在三个问题上:

(1) 通道负载均衡问题。多通道内存系统设计提供了较高的数据流并行性,但是只有多个处理器核分别同时访问多个内存通道时,多通道内存系统设计的优势才能体现出来。

(2) Rank 负载均衡问题。负载均衡问题并不仅仅存在于多通道内存系统设计中。在单通道内存系统中,一个内存控制器可以连接多个 Rank。这种组织形式除了可以获得较高的内存容量外,还可以增加内存系统可以访问的 Bank 数目,增加页命中率和 Bank 级并行度。若多个处理器核对内存系统的访问集中在某个 Rank 上,则会引起严重的 Bank 冲突,降低内存系统性能。

(3) Bank 负载均衡问题。Bank 负载均衡问题是指多个数据流访问主要集中在某个 Bank 上,从而引起严重 Bank 冲突的情况。

解决负载均衡问题最常见的办法是地址交错技术[11]。根据交错地址层次,地址交错可分为:

(1) 通道级交错,指应用程序连续的地址访问落在不同的通道中。

(2) Rank 级交错,指应用程序连续的地址访问落在相同通道的不同 Rank 中。

(3) Bank 级交错,指应用程序连续的地址访问落在相同 Rank 的不同 Bank 中。

图 7 举例介绍了传统地址翻译格式。在通常情况下,访存地址会被翻译成通道地址 CH,Rank 地址 CS,行地址 ROW,Bank 地址 BA 和列地址 COL。按照这种地址翻译格式,应用程序连续的地址访问会落在同一个通道或者 Rank 中,从而引起严重的通道,Rank 甚至 Bank 负载不均衡问题。

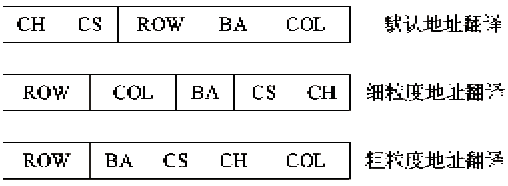

图 7 地址翻译格式举例

根据交错地址粒度,地址交错可分为:

(1) 细粒度地址交错。如图 7,细粒度地址交错是指用低位访存地址作为通道,Rank 或者 Bank 的选择信息。按照这种地址翻译格式,应用程序对连续地址的访问会落在不同通道的不同 Rank 中。

(2) 粗粒度地址交错。如图 7,粗粒度地址交错是指用较高位访存地址作为通道,Rank 或者 Bank 的选择信息。按照这种地址翻译格式,应用程序的连续地址访问会在通道,Rank 或者 Bank 内部保持一定程度的连续性。

细粒度地址交错可以将所有进程的访存请求均

匀地分配到各个通道,Rank 和 Bank 上,有利于负载均衡。但是细粒度地址交错容易破坏访存连续性,使在 Bank 内的连续访存请求变成对不同 Bank 的访存请求,而粗粒度地址交错维护了这种连续性。

为保证多微通道内存系统各通道,MRank 和 Bank 负载均衡,同时不破坏数据流内存地址访问的连续性,多微通道内存系统采用图 8 描述的三级地址交错模式。如图 8 所示,当多微通道内存系统配置为 X16 模式时,内存系统由 4 个 16 位通道组成。每个通道并联 4 个 MRank,这 4 个 MRank 时分复用对应 16 通道的数据总线。4 个 16 位通道时分复用一套命令地址总线。

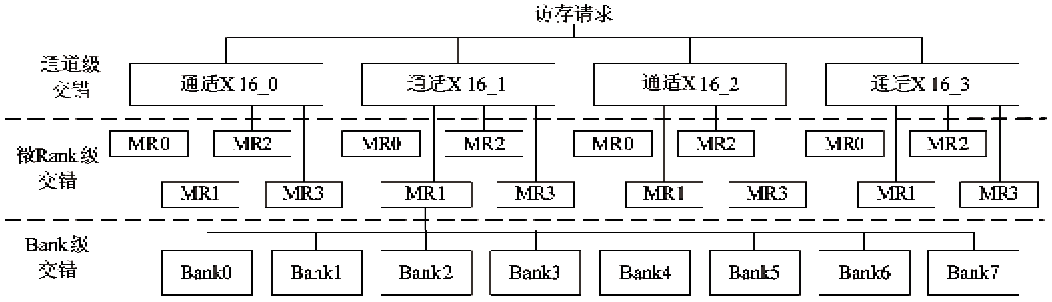

图 8 三级地址交错模式

多微通道内存系统采用如图 9 所示的地址翻译格式。访存请求进入内存系统后根据该地址翻译格式分别得到通道地址 CH,Rank 地址 CS,行地址 ROW,Bank 地址 BA 和列地址 COL。其中 CH,CS 和 BA 用于将访存请求路由到对应的 Bank。COL 由 COL_{high} 和 COL_{low} 得到。COL_{low} 的位数决定了进程打开内存页后可以连续访问的数据量。在本文的实验部分,COL_{low} 的值设为 8。

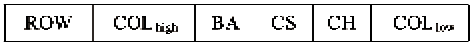

图 9 多微通道地址翻译

4 实验环境

为评估上面提出的优化方法,本文选用龙芯 3B1500 多核处理器作为评估平台[12],其主要参数如表 2 所示。

表 2 龙芯 3B1500 多核处理器参数

参数	描述
工艺	32nm
功耗	40W
主频	1.5GHz
指令集	MIPS64,LISA64,LISA64v
微结构	8 核,4 发射 64 位超标量,乱序执行
私有一级缓存	指令缓存 64kB,数据缓存 64kB
私有二级缓存	128kB
共享三级缓存	8MB
内存控制器	双通道 DDR3-1600
HyperTransport	双通道,最高频率 800MHz

本文实验采用 DDR3 1600 内存颗粒,表 3 介绍了实验使用的内存颗粒参数。本文的实验在 EVE ZeBu 硬件加速仿真平台上进行,测试使用 LMbench 测试程序[14]。

表 3 内存芯片颗粒参数

参数	描述	参数	描述
数据宽度	16 Bit	运行速率	-125
最高工作电压	1.575 V	最低工作电压	1.425 V
容量	2Gb	tREFI	6240 DRAM 周期
tRP	9 DRAM 周期	tWR	9 DRAM 周期
tRCD	9 DRAM 周期	tRC	39 DRAM 周期
tCCD	9 DRAM 周期	tRFC	128 DRAM 周期
tRRD	4 DRAM 周期	tFAW	32 DRAM 周期

5 实验结果与分析

5.1 带宽测试结果及分析

Bw _ mem 是一个存储带宽测试程序,通过对不同大小数据块进行读写以测试不同存储层次的带宽。

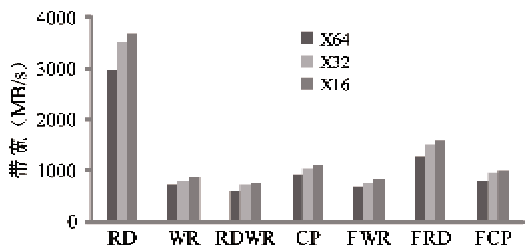

图 10 带宽测试结果

图10介绍了Bw_mem测试程序不同选项的带宽测试结果。相对于传统的64位单通道内存系统,32位和16位多微通道内存系统的带宽分别增长了17.3%和21.8%。

内存系统带宽的增长主要是由于下述五个原因:

(1)页命中率。多微通道内存系统独立控制每个内存颗粒,多个Bank可以同时打开,有效降低了Bank冲突,从而大大提供了内存带宽利用率。

(2)Bank级并行度。多微通道内存系统通过独立控制各个内存颗粒,每个内存Rank被切割成了多个可以被并行访问的MRank,从而减少了内存颗粒参数对Bank级并行度的限制,提高了内存系统并行处理访存请求的能力。

(3)数据流并行性。在多微通道内存系统中,多个数据流并行处理,有效提高了内存系统并行性和资源利用率。

(4)隐藏读写切换延迟。传统内存系统的64位数据总线被切分成多套窄数据总线,这些数据总线可并行地传输不同方向的数据(读数据和写数据)。在传统64位内存系统中,当数据总线发生读写数据传输方向转换时,整个数据总线都需要停滞一段时间,从而引入了读写切换延迟。而在多微通道内存系统中,当一套窄数据总线由于发生读写数据传输方向转换而停滞时,其它的数据总线可继续传输数据,从而达到了隐藏读写切换延迟的目的。

(5)命令总线利用率提高。在传统64位内存系统中,每个数据通道独占一套命令总线,受内存芯片各种时序参数的限制,内存系统命令总线利用率不高。在多微通道内存系统中,多个通道共享一套命令总线,这些通道的访存请求被串行处理,有效提高了内存系统命令总线的利用率。

5.2 延迟测试结果及分析

图11介绍了这些测试程序的访存延迟测试结果。相对于传统的64位单通道内存系统,32位和16位多微通道内存系统的访存延迟分别下降了3.9%和14.4%。在64位,32位和16位内存系统中,在内存控制器和内存芯片之间传输64字节数据分别需要4,8和16个内存时钟周期。虽然X16内存系统需要较长的数据传输延迟,但是数据传输时间仅仅占了访存延迟的一小部分。在内存系统带宽压力较大时,访存请求在访存队列中排队,排队延迟占了访存延迟的绝大部分。多微通道内存系统通过独立的控制内存颗粒,不同的数据流在内存系统中并行传输,内存页命中率和Bank级并行度的提高,读写切换延迟的隐藏,使得内存命令平均处理延迟减少,从而大大降低了内存系统排队延迟。

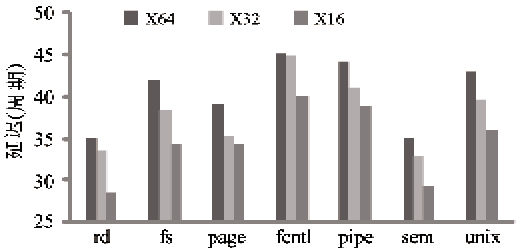

图11 延迟测试结果

5.3 功耗测试结果及分析

多微通道内存系统设计不需要对内存芯片做出任何修改,但是需要增加额外的CS信号用作细粒度的内存颗粒控制。多微通道内存系统中有多个内存通道,且每个内存通道都有独立的访存队列和CS状态机。这些功能单元引入了额外的片上面积和功耗开销。

多微通道内存控制器时钟频率为800MHz,使用32nm工艺。表4介绍了多微通道内存控制器的面积。从表中可以看出,多微通道内存控制器的面积主要由命令队列,CS状态机和物理接口组成。在具体实现时,由于多个访存队列缓解了内存控制器的排队压力,访存队列不需要采用多项队列设计。本文实现的多微通道内存控制器中,需要使用4个访存队列,每个访存队列采用8项队列设计。每个访存队列单元包括访存队列项,队列控制逻辑和读写数据缓存。新增加的三个访存队列和CS状态机占了内存控制器面积的21.3%。

表4 多微通道内存控制器片上面积开销

模块	面积(μm^2)
内存控制器	907050
8项命令队列	63507
CS_SM	862
物理接口	424918

本文对多微通道内存系统的功耗开销进行评估[15]。图12介绍了LMbench测试程序的访存功耗测试结果,该结果包括内存控制器和内存芯片的功耗。相对于传统的64位单通道内存系统,32位和16位多微通道内存系统的访存功耗分别下降了12.3%和26.2%。多微通道内存控制器独立的控制每个内存颗粒,在功耗消耗方面有下面三个优势:

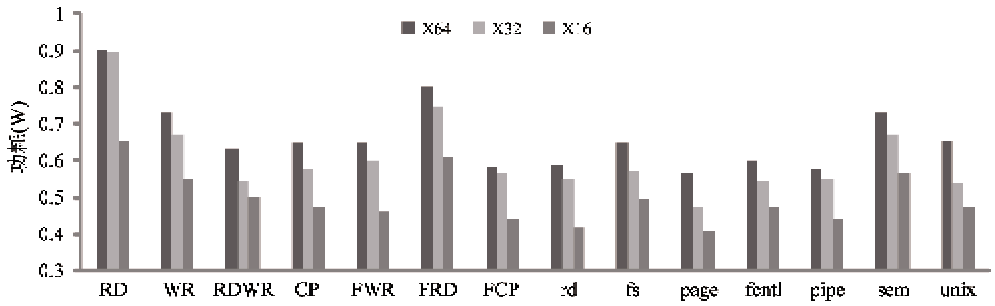

图 12 功耗测试结果

(1) 内存颗粒操作功耗降低。由于每个内存颗粒单独控制,每个内存命令需要控制的内存颗粒数目减少,从而降低了内存颗粒操作功耗。

(2) 内存颗粒冲突功耗降低。多微通道内存系统 Bank 冲突的减少除了提高内存系统性能以外,还减少了内存命令的数量(Precharge 和 Activate 命令),从而起到了降低内存系统功耗的作用。

(3) 细粒度低功耗状态的使用。多微通道内存系统设计使内存系统 MRank 数量大大增加。当 MRank 没有被访问时,可以通过细粒度的低功耗状态控制,使其进入低功耗状态[16],从而进一步降低内存系统功耗开销。

6 结 论

内存系统对访存请求的处理速率是制约内存系统性能的关键因素,而访存请求的处理速率主要取决于内存系统页命中率,Bank 级并行度和读写切换率。通过并行访问组成内存芯片的多个颗粒,多微通道内存系统设计针对这三个关键访存特性进行优化。相对于传统的 64 位单通道内存系统,32 位和 16 位多微通道内存系统的带宽分别增长了17.3%和21.8%,访存延迟分别下降了3.9%和14.4%,访存功耗分别下降了12.3%和26.2%。本文提出的多微通道设计方法已经应用于龙芯 3B1500 多核处理器中,并取得了预期的优化效果。

参考文献

[1] Rixner S, Dally W J, Kapasi U J, et al. Memory access scheduling. In: Proceedings of the 27th Annual International Symposium on Computer Architecture, 2000. 128-138

[2] 曾洪博,胡明昌,李文等. 一种高性能北桥芯片的设计及性能分析. 计算机研究与发展, 2007, 44(9): 1501-1509

[3] Eyerman S, Eeckhout L. System-level performance metrics for multiprogram workloads. In: Proceedings of the 41th Microarchitecture, New York, USA, 2008. 42-53

[4] Nesbit K J, Aggarwal N, Laudon J, et al. Fair queuing memory systems. In: Proceedings of the 39th Microarchitecture, New York, USA, 2006. 208-222

[5] Mutlu O, Moscibroda T. Parallelism-aware batch scheduling: Enhancing both performance and fairness of shared DRAM systems. In: Proceedings of the 35th Annual International Symposium on Computer Architecture, New York, USA, 2008. 63-74

[6] Frederick A W, Craig H. Improving power and data efficiency with threaded memory modules. In: Proceedings of the 24th International Conference on Computer Design. NJ: IEEE, 2006. 417-424

[7] Brewer T M. Instruction set innovations for the Convey HC-1 computer. In: Proceedings of the 43th Microarchitecture, New York, USA, 2010. 70-79

[8] Zheng H, Lin J, Zhang Z, et al. Mini-rank: Adaptive DRAM architecture for improving memory power efficiency. In: Proceedings of the 41th Microarchitecture, New York, USA, 2008. 210-221

[9] Yoon D H, Jeong M K, Erez M. Adaptive granularity memory systems: A tradeoff between storage efficiency and throughput. In: Proceedings of the 38th Annual International Symposium on Computer Architecture, New York, USA, 2011. 295-306

[10] Vogelsang T. Understanding the energy consumption of dynamic random access memories. In: Proceedings of the 43th Microarchitecture. New York, USA, 2010. 363-374

[11] 李文. 存储控制系统性能优化技术研究:[博士学位论文]. 北京:中国科学院计算技术研究所, 2005. 11-15

[12] 王焕东, 高翔, 陈云霁等. 龙芯 3 号互联架构的设计与实现. 计算机研究与发展, 2008, 45(12): 2001-2010

[13] 胡伟武, 张福新, 李祖松. 龙芯 2 号处理器设计和性能分析. 计算机研究与发展, 2006, 43(6): 959-966

[14] Micro Corporation. Calculating memory system power for DDR3. http://download. micron. com/pdf/technotes/TN4603. pdf, 2007

[15] Larry M V, Carl S. Lmbench: portable tools for performance analysis. In: Proceedings of the annual conference on USENIX, New York, USA, 1996. 23-38

[16] Delaluz V, Kandemir M, Vijaykrishnan N, et al. DRAM energy management using software and hardware directed power mode control. In: Proceedings of the 7th International Symposium on High Performance Computer Architecture. NJ: IEEE, 2001. 159-169

Design of multiple micro-channel memory systems

Zhang Guangfei[*,**,***], Wang Huandong[*], Chen Xinke[*,**,***], Huang Shuai[*], Chen Liwei[*,**,***]

([*]Key Laboratory of Computer System and Architecture, Chinese Academy of Sciences, Beijing 100190)

([**]Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

([***]Graduate University of Chinese Academy of Sciences, Beijing 100049)

Abstract

According to the queuing theory, a memory system performance model was established, and the key factors affecting the memory controller and then the performance of the memory system were analyzed. Further, a method for design of multiple micro-channel memory systems (MMCM) was proposed. The design can bring the advantages below. By controlling DRAM devices concurrently, multiple banks can be opened simultaneously, which decreases bank conflicts and promotes DRAM data bus utilization. All the channels share the same DRAM command bus in sequence. The DRAM operation power is reduced to a large extent since fewer DRAM devices are involved in every DRAM command. Different data streams corresponding to different DRAM commands can flow from or to DRAM devices concurrently. The memory queuing latency can be reduced. MMCM systems can achieve the best performance/power efficiency. The experimental results show that, compared with conventional desgins MMCM can improve the memory system bandwidth by 21.8%, and decrease the memory access latency by 14.4% with 26.2% reduction in DRAM power consumption on average.

Key words: DRAM system, memory controller, chip multiprocessor, multi-channel, memory access characteristic