

用于 DDR3 访存优化的数据缓冲机制*

陈胜刚^{1,2}, 付兴飞¹, 曾思¹, 刘胜¹

(1. 国防科技大学 计算机学院, 湖南 长沙 410073; 2. 并行与分布处理国家重点实验室, 湖南 长沙 410073)

摘要:为提高 DDR3 控制器访存效率,设计了基于 DDR3 存储器预取访问数据长度的数据缓冲机制,将访存请求分为三种基本类型并分别排队处理,降低数据丢弃和实际动态随机访问存储器访问发生次数。针对图像和视频类应用程序的实验结果表明,相对于传统先到先服务的 DDR3 访存控制器,该机制取得了平均 21.3%、最好 51.3% 的性能提升,硬件开销在可接受范围内。

关键词:DDR3 控制器;访存优化;数据缓冲

中图分类号:TN95 **文献标志码:**A **文章编号:**1001-2486(2017)06-039-06

DDR3 data buffering for memory access optimization

CHEN Shenggang^{1,2}, FU Xingfei¹, ZENG Si¹, LIU Sheng¹

(1. College of Computer, National University of Defense Technology, Changsha 410073, China;

2. National Laboratory for Parallel and Distributed Processing, Changsha 410073, China)

Abstract: In order to improve the memory access efficiency of the DDR3 memory controller, a data buffering mechanism based on DDR3 memory access burst length was proposed. The application requests were guided into three different queues. The data buffering mechanism can make use of the additional data obtained from DRAM (dynamic random access memory) in one of the former request, thus reducing the actual external DRAM access needed. Experiments on several image and video application show that the proposed mechanism can improve the memory controller by an average 21.3% and a peak by 51.3% at an acceptable hardware cost when compared with the FCFS (first-come-first-serve) baseline DDR3 memory controller.

Key words: DDR3 memory controller; memory access optimization; data buffering

DDR3 存储器是一种采用时钟双沿工作的高速存储器,是处理器常用的片外存储器。与静态随机访问存储器(Static Random Access Memory, SRAM)所采用的 CMOS 工艺不同,DDR3 存储器采用动态随机访问存储器(Dynamic Random Access Memory, DRAM)动态电路工艺,多采用电容储值,读写之前必须先对数据线进行预充电;读是破坏性的,读后必须写回;漏电流的存在使得 DDR3 必须保持定期刷新(读出放大后再写回)。

由于这些复杂的读写过程以及高速电路信号完整性的要求,JESD79-3^[1]国际标准对 DDR3 的接口信号、交流/直流特性和逻辑组织方式等方面进行了规定;为了获得更好的封装性和便携性,电子器件工程联合委员会(Joint Electron Device Engineering Council, JEDEC)定义了多钟双列直插式存储模块(Dual In-line Memory Module, DIMM)的国际规范^[2],将多颗 DDR3 DRAM 芯片通过特

定的印制电路板封装成现成的存储模块,具备符合规范的电气特征和功能特征,容量更大,方便拔插使用。基于 DDR3 的这些特性,JESD79-3 标准兼容的 DDR3 一般采用 $8n$ 的预取模式(其中 n 是 DRAM 芯片的数据位宽, n 通常为 32 或 64),这意味着每一次 DRAM 访问都会涉及 256 位或者 512 位的数据。

但是,以 Jpeg、视频编解码等为代表的多媒体应用处理对象是块状数据,在 DDR3 存储器中的地址不连续,应用程序一次访问的数据需求无法连续消费一个 DDR3 $8n$ 预取的数据(本文称此类 DDR3 存储器访问为非完全 DRAM 请求,与系统层的小粒度访存请求作区别),因此未经优化的访存控制器往往会丢掉多余的数据,而在需要的时候又重新读取,造成典型的带宽浪费;另外,多媒体应用的两次访存之间间隔一般比较大,而连续的访存间往往缺乏物理连续性,再加上系统层

* 收稿日期:2016-07-20

基金项目:国家自然科学基金资助项目(61402499,61602493,61402500,61672526)

作者简介:陈胜刚(1981—),男,四川邻水人,助理研究员,博士,E-mail:shgchen@nudt.edu.cn

的地址映射和乱序执行等机制,DDR3 存储控制器难以对接收到的请求捕捉特定的规律,地址跳跃的 DRAM 请求访问极易造成系统层 Cache 频繁失效替换,从而失去预期效果。

因此,在 DDR3 存储控制器中对 DDR3 访存请求进行优化,是提高 DDR3 访问效率的必要措施。文献[3-4]针对 AXI(advanced extensible interface)协议的 DDR3 访存控制器进行优化,采用请求合并处理或专用的读写队列提高控制器访存效率。文献[5-7]采用命令调度、乱序执行的方式提高已打开页面的命中率。这种方法要求程序员对程序访存的序列控制能力要强,要求较高,且难以实现。文献[8-9]利用 DDR3 多 Bank 特性,将连续的访问映射到不同的 Bank 中,从而实现同时对多个页面的并发流水访问。这种方法受限于 DDR3 的 Bank 数目,且方法过于不灵活,易失效。文献[10]对存储系统的能耗问题进行了研究。

1 DDR3 存储器的页缓冲机制

图 1 给出了一种多个存储空间(Rank)的 DDR3 DIMM Rank0 的地址逻辑组织方式示意图,其中每个 Rank 实际上对应一个片选信号 CS#。整个存储体一般由 8 个体(Bank)组成,每个 Bank 按照行(Row)和列(Column)的二维方式组织,其中 Row 又称为页面(Page)。DDR3 DIMM 通过 Rank 地址(CS#)、行地址(Row Address, RA)、列地址(Column Address, CA)以及体地址(Bank Address, BA)来寻址存储单元,每个单元存储的数据比特数就是当前 DDR3 存储体的位宽;DDR3 Chip 的位宽有 4、8、16 位三种,而 DIMM 的位宽则更宽,一般常见的有 32、64 位等。

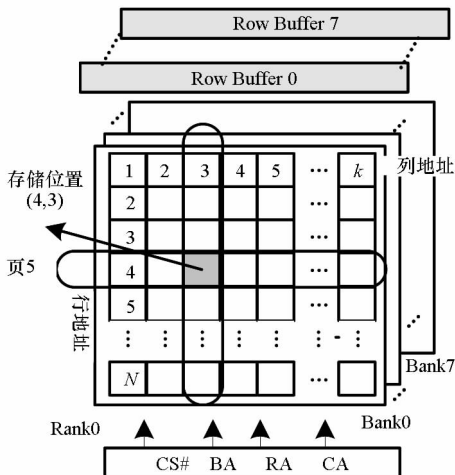


图 1 DDR3 存储器一般逻辑结构

Fig. 1 Logic structure of common DDR3 memory

根据 JESD79-3 规范,DDR3 存储器的每个 Bank 包含一个行缓冲(Row Buffer),又称为页缓冲(Page Buffer),是一个页面读写数据的缓冲。根据规范,对一个处于 IDLE 状态的 DDR3 Bank 进行读写之前,必须先激活要访问的行(页)(将数据读出到页缓冲),然后才能对该行进行访问;每个 Bank 同时只能允许激活一行,若要访问其他的行,必须先关闭当前行(写回页缓冲的数据),再激活其他的行进行访问。当该 Bank 的连续访问均命中已打开页面,访问速度较快;一旦访问换页,页缓冲中的数据将会失效写回并读出新的数据,引起大量的时间开销。

由此可知,较多的打开页将获得更高的访问性能,但 DDR3 每个 Bank 同时只能打开一个页面,故其总共最多同时打开 8 个页面,这对利用页缓冲机制提高 DDR3 访问带来了极大的限制。因此,研究者们提出了一种试图提高页缓冲数量的机制,虚拟打开页机制,是在控制器中设置多个虚拟打开页面,命中的访问直接发生在虚拟打开页中,当虚拟打开页发生写回的时候才真正访问 DDR3 存储器。虚拟打开页缓冲提高了命中访问的效率,同时在写回 DDR3 时由于访问地址连续且数据量大,DDR3 的页缓冲利用效率高,存储访问效率自然较高。

然而,虚拟打开页机制的致命缺点是硬件开销非常大,一般的 DDR3 DIMM 的页面大小达到了 8 KB ~ 16 KB,多个虚拟打开页的开销将迅速赶超 L1 甚至 L2 Cache 的硬件开销。

2 基于访问长度的数据缓冲机制

根据对多媒体算法的特征分析发现:

1)位宽为 64 位的 DDR3 DIMM 的依次访问数据量为 512 位(8 × 64 位),小于一般系统通用 Cache 的设计容量,大于多媒体应用程序一次连续地址访问的数据量,而相差的这部分数据在较短的时间内将会被访问到;

2)多媒体应用程序的访问大多数是基于块状数据的访问,块的形状决定了一定时间内不会访问完 DDR3 DIMM 的一整行数据(一般 DDR3 DIMM 的一整行数据大小为 16 KB)。

因此,提升 DDR3 访问的效率主要有两个方向,一是充分利用每一次 DRAM 访问的带宽,二是充分访问已经打开的页面。根据传统虚拟打开页缓冲机制和以上观察,本文设计了一种基于 DDR3 DRAM 访问长度的数据缓冲机制,用于缓存 DDR3 存储控制器的 DRAM 请求的数据,解决

虚拟打开页缓冲机制的开销问题,并获得访存效率提升。

本文设计的基于访问长度的数据缓冲机制一共设置了 m 个数据缓冲行,每一个数据缓冲行的大小为 DDR3 DIMM 一次 DRAM 访问的数据长度。这既能降低缓冲开销,又能解决非完全 DRAM 访存请求的数据浪费问题,同时最多允许同一个 Bank 中虚拟了 m 个打开页面的片段。

图2是基于访问长度的数据缓冲机制示意图,其中 TAG(标识)体用于保存数据缓冲行的状态和标识;数据缓冲按照传统 Cache 划分方法,采用全相联方式组织;访问被分为了三种基本类型并进入相应的队列:未命中请求(Not-Hit Request, NHR)队列、半命中请求(Half-Hit Request, HHR)队列、全命中请求(Full-Hit Request, FHR)队列;重定序模块将返回给应用层的数据按照访问请求的顺序排队,便于系统集成,从而降低上层处理难度。

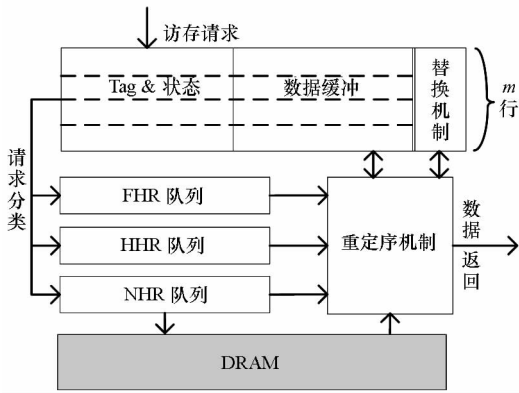


图2 数据缓冲机制示意图

Fig. 2 Diagram of the data buffering mechanism

2.1 TAG 体及数据缓冲

与通用 Cache 一样,本文设计了 Tag 体,不仅保存了访存地址,还包括数据缓冲行的状态等标志。请求分类机制根据请求的地址,对所有 m 行的 TAG 进行比较匹配,以确定请求的命中情况。

根据命中情况,分类后的读请求分别进入三个队列:

1) NHR 队列:请求未命中队列中所有的数据缓冲行,该请求会直接向 DRAM 发起正常的访问。替换策略机制将会为该请求分配一行,被替换的行并不执行写回操作。替换策略采用的是最简单的最近最久未使用(Least Recently Used, LRU)方法。

2) HHR 队列:缓存实际上是“缺失后缺失”

的请求,即当前请求命中了一个之前的 NHR 请求新分配的行,NHR 新分配的行数据还未就绪。该请求等待数据就位,并直接从数据缓冲中读取所需数据。HHR 请求会在对应的缓冲行的标签(Tag)体留下标识信息。

3) FHR 队列:请求直接命中数据缓冲中的一个有效行。

对于写请求,本文采取的是类写直达、不按写分配法,执行写操作时其基本操作过程如下:

1) 类写直达:所有的写请求直接写入下一级 DRAM 中;与写直达法不同的是,若发生命中,并不会将写数据写入对应的缓冲行,而是采取直接将对应的缓冲行“脏”标志位拉高,阻止后续读请求命中该行。那么发生在之前的 NHR 能够正确地将旧数据返回该行,HHR 读请求就能获得正确的旧数据;当写之后再具有相同地址的请求发生,则需要作为 NHR 请求重新分配。这种做法可以迅速结束写请求,防止请求在控制器中排队,充分利用 DDR3 存储总线;同时,直接写入下层 DRAM,以防止后续 NHR 的读请求与之发生写后读的相关性冒险。

2) 不按写分配:写请求并不会分配新行。不按写分配保证了缓冲行分配仅发生在 NHR 请求时,从而简化数据流的设计,也简化了读写强序的保证。

因此,实际的 DRAM 访存请求只有 NHR 请求和写请求。

2.2 基于顺序标签的数据返回再定序仲裁

再定序机制用于对读返回数据进行重新排序,将按照上层读请求发出的严格顺序和格式返回给上层系统,这在大多数的实际系统中是非常关键的。

本文引入顺序标签机制,每个到达控制器的访存请求都将赋予唯一顺序标签,再定序机制根据顺序标签确定读请求的返回顺序。NHR 队列、HHR 队列和 FHR 队列内的请求都是按照访存的先后顺序排列的。因此再定序仲裁只需要在这三个队列中进行顺序标签的比较和仲裁就能够实现读返回的顺序。

再定序过程中,HHR 读请求需要从数据缓冲中读取对应的数据返回;因为 HHR 之前必定有一个 NHR 读请求,因此 HHR 数据返回时,数据缓冲中的数据必定已经被 NHR 读请求更新为正确数据。数据返回再定序仲裁的基本算法过程如算法1所示。

算法 1 再定序过程

Alg. 1 Reordering procedure

1. 从请求中获取一个完成顺序标签;
2. 将完成顺序标签与每个队列的头项进行比较;
3. if NHR 队列与之匹配 then;
4. 直接将数据送给上层应用;
5. 并将数据写入一个新的缓冲行;
6. else if HHR 队列头项与之匹配,那么:
7. 从缓冲中搜索请求的数据;
8. 并将数据送给上层应用;
9. else if FHR 队列与之匹配,那么:
10. 直接将数据送给上层应用;
11. end if
12. 完成顺序标签自增 1。

3 实验与分析

本文对所设计的直接面向 DDR 的基于访问长度的数据缓冲机制进行 Verilog 描述和物理实现,并在一款自主设计的 SoC 芯片上进行了应用程序性能评估。系统和实验的基本参数如表 1 所示。

表 1 实验设置

Tab. 1 Experimental setting

项目	描述
外部存储器	DDR3 64 位 UDIMM
系统总线	自研 256 位类 AXI 带突发内部总线
工具	Verilog、NCVerilog ©、Encounter ©等
目标设计	无缓冲的 FCFS DDR3 基线存储控制器
行大小	512 位(即 64 位宽 DIMM 长度)
数据缓冲行	三种配置:1,4,8

本文首先选取了典型多媒体应用程序 Jpeg_dec 和 H264_enc,并选取了包含大量非完全 DRAM 访存请求的 Mem_test 程序;同时为了进行对比,选取了此类访存请求较少甚至几乎没有的应用程序快速傅里叶变换(Fast Fourier Transform, FFT)和 Matrix_cal 作为实验对象。

定义非完全 DRAM 访存请求的特征为访存中对物理地址连续的数据实际需求量小于 DDR3 DIMM 的一次物理访存的数据量。对几种应用程序的非完全 DRAM 访存请求特性分析如表 2 所示。

表 2 应用程序

Tab. 2 Application description

应用程序	描述	非完全 DRAM 请求所占比例
Jpeg_dec	Jpeg 解码程序	100%
H264_enc	H264 基线编码程序	100%
Mem_test	存储性能测试程序	92.1%
FFT	快速傅里叶变换算法	9.0%
Matrix_cal	矩阵乘积	0

从表 2 中可以看出,Jpeg_dec 和 H264_enc 由于采用 8 位数据存储图像,并且数据处理对象为 8×8 或者 16×16 的块数据,因此几乎所有的访存均为非完全 DRAM 访存请求。Mem_test 是一个用于测试处理器存储器性能的测试程序,与设置有关,多为非完全 DRAM 访存请求,但访存局部性较差。而 FFT 和 Matrix_cal 中,数据对象很大,因此非完全 DRAM 访存请求较少,甚至没有。

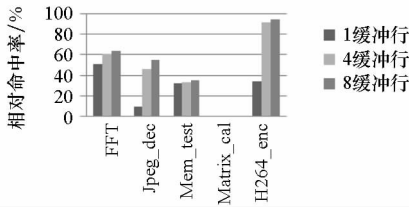
3.1 数据缓冲命中分析

命中数据缓冲将极大地降低访存的时间,因此本文针对三种配置,分析访存请求的数据缓冲行命中率。本文分析了两种命中率,相对命中率和总体访存绝对命中率,结果如图 3 所示,其中缓冲行绝对命中率是相对其所有的 DDR3 访存而言的,相对命中率是指非完全 DRAM 访存请求中的命中率。

图 3(a)给出了应用程序数据缓冲的相对命中率。从图中可以看出,FFT、Jpeg_dec 和 Mem_test 的非完全 DRAM 访存请求中,约 39% ~ 64% 的访存能命中数据缓冲,而 H264_enc 有 90% 以上的非完全 DRAM 访存请求能有效命中数据缓冲。Matrix_cal 由于非完全 DRAM 访存请求量稀少,且访存地址变化较大,所以几乎无命中数据缓冲。相对命中率在一定程度上反映了数据缓冲对应用程序中非完全 DRAM 访存的优化程度。

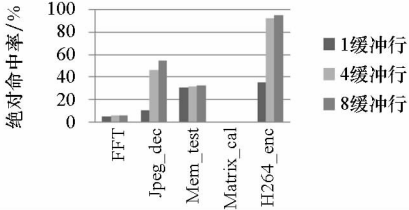
图 3(b)给出了应用程序的绝对命中率结果。H264_enc 数据缓冲命中率达到 94.4%,这是因为其访存几乎全为非完全 DRAM 访存请求,且由于访问块重用性好,故多发生首次失效。Jpeg_dec 数据缓冲命中率仅 32.3%,因为其访存的块相对较小,算法对数据的重复使用相对较差。

从图中还可以观察到,FFT、Mem_test 程序的命中率对于数据缓冲行的数目敏感度不高。这是由于 FFT 和 Mem_test 由于数据访问跳跃性更大,



(a) 缓冲行相对命中率

(a) Relative buffer hit rate



(b) 缓冲行绝对命中率

(b) Buffer hit rate

图 3 缓冲行命中率

Fig. 3 Buffer hit rate

总是只能保持连续一两次的缓冲行命中,因此命中率对缓冲行的数目不敏感。而 Jpeg_dec 和 H264_enc 访存呈不同大小的块状,缓冲行越多效果就越好。但也正是由于访存呈现确定性的块状大小,因此当缓冲行的数目增大到 4 以上的时候,由于首次访存失效的固定性存在,增加更多的缓冲行已经无法更明显地提升命中率了。

3.2 应用程序总体性能提升

命中率只能说明程序的访存行为,对系统整体运行性能(运行时间)的影响才是最终的目的。测试程序都是数据密集型应用,访存性能的高低直接影响运算单元运算能力的发挥:更高的 DRAM 供数带宽可有效降低运算单元等待时间,从而降低测试程序的执行时间。

本文针对数据缓冲行的三种配置分别作了应用程序性能测试,结果如图 4 所示,其中百分比是相对于目标设计的 FCFS DDR3 基线控制器的性能得到的。

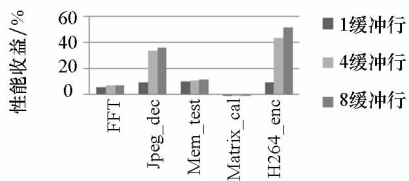


图 4 应用程序总体性能收益

Fig. 4 Applications total performance gains

从图 4 可以看出,除了 Matrix_cal 之外,其他应用程序均取得了性能提升。Matrix_cal 的数据需求特征是大量数据传输,本身已经能够获得较

高的 DDR3 访存效率,本文设计的数据缓冲机制无法对其起到加速效果。对于非完全 DRAM 访存请求量较少的 FFT 也是同样的道理。Jpeg_dec 和 H264_enc 的访存性能提升较大,最高分别达到 36.0% 和 51.3%,这与其具备较高的非完全 DRAM 访存请求的现象是一致的。

与之前的分析一样,FFT 和 Mem_test 的总体性能提升对缓冲行的数目不敏感,而 Jpeg_dec 和 H264_enc 在缓冲行达到 4 之后,性能提升速率明显变小。因此,针对本次这些应用程序而言,缓冲行的设置取 4 即可达到较好的优化效果。

3.3 硬件存储开销分析

本文设计的数据缓冲机制主要硬件开销集中在数据缓冲和命令队列,表 3 给出了数据缓冲行为 4 的数据缓冲机制的存储开销细节。其中,地址宽度假设为 29 位,数据位宽为 256 位,NHR、HHR 和 FHR 的队列长度为 8。从表中可以看出,本位设计的基于 DDR3 访问长度的数据缓冲机制总体存储开销为 7304 位(不到 1KB),开销相对较小。

表 3 数据缓冲的存储开销统计(缓冲行=4)

Tab. 3 Hardware cost when 4 lines in data buffer

开销项	开销内容简述	开销/bit
Tag 体	顺序标签	$20 \times 4 = 80$
	地址	$29 \times 4 = 116$
	状态项	$3 \times 4 = 12$
数据	数据缓冲	$512 \times 4 = 2048$
请求队列	NHR 队列	$(29 + 20 + 1) \times 8 = 400$
	HHR 队列	$(29 + 20) \times 8 = 392$
	FHR 队列	$(20 + 512) \times 8 = 4256$
总体		$80 + 116 + 12 + 2048 + 400 + 392 + 4256 = 7304$

4 结论

本文提出一种专注于具有非完全 DRAM 访存请求的应用程序的基于 DDR3 存储器访问长度的数据缓冲机制,核心思想是充分利用现代 DRAM 的预取架构,来提高系统程序的 DRAM 访存效率。本文通过改进传统虚拟打开页请求机制,建立了多个较小的数据缓冲行缓存将来可能有效的数据,访存请求被识别并划分为三个请求队列,分别为 NHR, HHR 和 FHR。其中只有 NHR 向 DRAM 发起访存行为,其他两种请求直接从缓

冲行获取所需数据。另外,为了保证最终的数据返回顺序和其请求到达顺序相同,本文设计了一种基于顺序标签的重排序机制,保证数据返回顺序与系统层请求顺序完全一致,这在许多系统应用中是非常关键的。

针对部分应用程序的实验和结果显示,相比于传统 FCFS DDR3 控制器,本文设计并取得了平均 21.3%、最高 51.3% 的大幅度性能提升,硬件开销也在可接受范围内。

值得指出的是,当系统层的 Cache 与 DDR3 控制器之间没有其他存储层次的时候,本文设计的数据缓冲机制的角色实际上已经被代替。但由于该机制开销小、复杂度低,因此在访存通路中不易产生负面影响。

参考文献 (References)

- [1] JEDEC Solid State Technology Association. SDRAM standard: JESD79 - 3F [S]. Arlington, USA: JEDEC Solid State Technology Association, 2012.
- [2] JEDEC Solid State Technology Association. DDR3 SDRAM unbuffered DIMM design specification: 4_20_19R22A [S]. USA: JEDEC Solid State Technology Association, 2013.
- [3] 陈胜刚, 付兴飞, 曾思. 基于协议控制器的 DDR3 访存控制器的设计与优化[J]. 微电子学与计算机, 2016, 33(6): 6 - 10.
CHEN Shenggang, FU Xingfei, ZENG Si. Design and optimization of a DDR3 memory controller with protocol controller[J]. *Microelectronics & Computer*, 2016, 33(6): 6 - 10. (in Chinese)
- [4] Rixner S, Dally W J, Kapasi U J, et al. Memory access scheduling [C]//Proceedings of the 27th Annual International Symposium on Computer Architecture, 2000: 128 - 138.
- [5] Lee R B, Smith M D. Media processing: a new design target [J]. *IEEE Micro*, 1996, 16(4): 6 - 9.
- [6] Shao J, Davis B T. A burst scheduling access reordering mechanism [C]//Proceedings of International Symposium on High-performance Computer Architecture, 2007: 285 - 294.
- [7] Zhang Z, Zhu Z C, Zhang X D. A permutation-based page interleaving scheme to reduce row-buffer conflicts and exploit data locality [C]//Proceedings of the 33rd Annual International Symposium on Micro-Architecture, 2000: 32 - 41.
- [8] Hur I, Lin C. Adaptive history-based memory schedulers [C]//Proceedings of the Annual International Symposium on Micro-Architecture, 2004: 343 - 354.
- [9] Shao J, Davis B T. The bit-reversal SDRAM address mapping [C]//Proceedings of the 9th International Workshop on Software and Compilers for Embedded Systems, 2005: 62 - 71.
- [10] Malladi K T, Nothaft F A, Periyathambi K, et al. Towards energy-proportional datacenter memory with mobile DRAM [C]//Proceedings of 39th Annual International Symposium on Computer Architecture, 2012: 37 - 48.
- [11] Gupta M, Nagawat A K. Design and implementation of high performance advanced extensible interface (AXI) based DDR3 memory controller [C]//Proceedings of International Conference on Communication and Signal Processing, 2016: 1175 - 1179.