


Y1523575

分类号 TP332

学号 GS07061037

U D C

密级 公 开

工程硕士学位论文

高性能 DDR3 存储控制器的研究与实现

硕士生姓名 万 轶

工 程 领 域 计算机技术

研 究 方 向 计算机硬件

指 导 教 师 窦 强 副研究员

李永进 副研究员

国防科学技术大学研究生院

二〇〇八年十月

摘 要

存储控制器是计算机系统中的重要组成部分，它是CPU和存储器之间交换数据的桥梁和纽带。存储控制器决定了计算机系统所能支持的最大存储器容量、存储器Bank数、存储器类型、速度以及存储器颗粒的数据宽度等重要参数，如何高效发挥存储器芯片有效数据带宽是决定计算机系统的存储器性能的关键，也是影响计算机系统整体性能的重要因素。

本文首先分析了DDR2存储器面临的挑战，对DDR3存储控制器技术发展的必要性进行了论述。随后本文介绍了DDR3的主要技术特点和DDR3存储器的控制状态机和控制命令。

基于当前国际上存储控制器的技术发展趋势，本文提出了一种DDR3存储控制器的总体设计方案，该方案将存储控制器的功能进一步划分为传输层和物理层，随后对其中主要模块的功能和实现细节进行了详细描述。该控制器可高效地完成访存请求调度，提高存储总线利用率，从而提高访存带宽，降低访存延迟，对今后其它支持DDR3存储器的数字系统设计提供了一定的参考。

关键字：DDR2，DDR3，存储控制器，传输层，物理层

ABSTRACT

Memory controller is an important part of the computer, which controls data exchange between the memory and CPU. The memory controller determines maximum memory capacity, memory bank, memory type, speed, memory data width and other important parameters. In other words, memory controller determines the memory performance of a computer system, which become an important factor in overall computer systems performance.

This paper first describes challenges DDR2 technology confronted, put forward the necessity of technology development of DDR3. Then introduces the key technology aspects of DDR3. It also describes the state transition diagram and control commands of DDR3.

According to the technology trends of current memory controllers, this paper proposed a design of high performance DDR3 memory controller. In this design, DDR3 memory controller is divided into two layers: transport layer and physical layer. Then, functions and implementation details of each major modules of these two layers is discussed thoroughly. The memory controller can efficiently process memory access requests, increase the utilization of memory bus, thus it could improve memory bandwidth and memory latency. And it could also be a good reference for future digital system designs which support DDR3 memory.

Keyword: DDR2, DDR3, memory controller, transport layer, physical layer

图目录

图 1.1	DDR、DDR2 与 DDR3 的结构对比以及相应的预取设计	7
图 2.1	DDR、DDR2 和 DDR3 的预取比较	8
图 2.2	DDR3 总线拓扑变化：从“星”型变为“Fly-by”(点对点)型	12
图 3.1	DDR3 状态简图	14
图 3.2	模式寄存器	17
图 3.3	读命令示例	19
图 3.4	存储器写操作	19
图 3.5	理论配置时间	21
图 4.1	存储控制器总体结构	22
图 4.2	UIB 逻辑结构图	24
图 4.3	DRAM 时钟域到 cpu 时钟域的转换电路	25
图 4.4	DRAM 时钟域到 cpu 时钟域的转换时序	25
图 4.5	cpu 时钟域到 DRAM 时钟域的转换电路	26
图 4.6	cpu 时钟域到 DRAM 时钟域的转换时序	26
图 4.7	UIB 读请求处理	28
图 4.8	UIB 读请求处理时序图	28
图 4.9	写请求处理处理模块	30
图 4.10	UIB 模块写请求处理时序图	30
图 4.11	dram_dp 模块中 256 位数据的选择	31
图 4.12	新 ecc 码产生	32
图 4.13	写数据被发送到物理层	32
图 4.14	仲裁级别	37
图 4.15	物理地址映射单元	39
图 4.16	数据通路	40
图 4.17	写数据通路结构图	41
图 4.18	物理层读写控制流程图	44
图 4.19	写数据传输通路结构图	45
图 4.20	写结构时序图	46
图 4.21	dram_mem 结构	46
图 4.22	读数据通路结构图	50
图 4.23	读数据通路时序图	51
图 5.1	写数据发送结构	53

图 5.2	写数据发送时序	53
图 5.3	时钟相位检测逻辑	55
图 5.4	DDR3 控制器数据采样电路	57
图 5.5	输入数据组装	58

表目录

表 2.1	DDR3 和 DDR2 基本规格对比	5
表 2.2	DDR2 和 DDR3 提供的带宽比较	9
表 3.1	DDR3 状态图部分命令	15
表 3.2	Bank 地址二进制数位配置	17
表 3.3	扩展模式寄存器 (EMR)	17
表 3.4	扩展模式寄存器 2 (EMR2)	17
表 3.5	DDR3 命令的种类和操作真值表	20
表 4.1	L2 数据校验位生成	33
表 4.2	L2 数据 ecc 校验综合代码表	33
表 4.3	dram_mem 模块的输入输出信号	48

独创性声明

本人声明所呈交的学位论文是我本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表和撰写过的研究成果，也不包含为获得国防科学技术大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文题目：高性能 DDR3 存储控制器的研究与实现

学位论文作者签名：王林 日期：2008年12月20日

学位论文版权使用授权书

本人完全了解国防科学技术大学有关保留、使用学位论文的规定。本人授权国防科学技术大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档，允许论文被查阅和借阅；可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密学位论文在解密后适用本授权书。)

学位论文题目：高性能 DDR3 存储控制器的研究与实现

学位论文作者签名：王林 日期：2008年12月20日

作者指导教师签名：黄勇 日期：2008年12月20日

第一章 绪论

在整个计算机系统中，存储器可谓是决定整机性能的关键因素之一。随着微电子技术的飞速发展，处理器的性能在成倍的提高，工作主频和总线带宽都很高，这就需要存储器提供很高的数据传输率来配合。由于CPU在运算时所需要的数据都要从存储器中获取，如果存储器系统无法及时给CPU供应数据，CPU不得不长时间处在一种等待状态，造成硬件资源闲置，性能自然无法完全发挥。因此，在计算机系统中，存储器带宽要和CPU的前端总线带宽相匹配，这样才不至于影响处理器性能的发挥^[1]。特别对于多路服务器来说，其对存储器带宽和存储器容量是极度渴求的，传统的存储访问技术已经无法满足其需求。而且处理器的速度提升还在继续进行，需要存储器提供更多的数据来满足处理器的要求，但是目前存储器的速度提升已经相当困难，这样就导致存储器成为计算机系统速度大幅度提高的最大瓶颈^[2]。为了适应更高速度的数据处理需求，联合电子器件工程委员会JEDEC (Joint Electronic Devices Engineering Council)在2007年6月颁布了JESD79-3规范，即DDR3存储器规范。DDR3 SDRAM是一种新型高速、大容量的双速率同步存储器。如何能够快速准确地给DDR3 SDRAM提供来自处理器的读写命令并在处理器和DDR3存储器间快速、可靠地传输数据，这就是DDR3存储控制器要解决的问题。

1.1 DDR 存储器芯片的发展

随着集成电路和高端处理器的发展，人们对计算机的运算速度和工作频率提出了越来越高的期望和要求，在处理器主频和带宽已经很高的情况下，存储器已经成为计算机系统整体性能的瓶颈，要解决这个瓶颈，就需要存储器提供足够高的数据传输率。为此，业界对计算机的存储器提出了全新的要求^[3]：

- 功耗和发热量较小

采用新的工艺，使DDR存储器拥有更低的工作电压，性能更好，更省电。使得存储器在控制成本的基础上也减少能耗和发热量。

- 工作频率更高

通过降低能耗，实现DDR更高的工作频率，从而在一定程度上弥补延迟时间较长的缺点。

- 容量更大

采用多Bank的设计，使DDR能够应对未来大容量芯片的需求。

- 更先进的地址/命令与控制总线的拓朴架构

通过采用新型的拓扑结构来减轻地址/命令/控制总线的负载，提高信号传输的质量。

按照 JEDEC 制定的 DDR 标准，800MHz 是 DDR2 家族中最高的频率。但在 2008 年的一次 IDF 上，英特尔指出了 DDR2-800 面临的几点挑战^[4]：

●DDR2-800 之后，DDR2 芯片的核心速度将超过 200MHz，由于速度提高，必将造成高功耗、低产出。由于核心频率超过 200MHz 的 DDR2 产品（比如 DDR2-1066 其核心频率为 266MHz）生产难度加大很多，良品率低，再加上速度的提高，这必将造成高功耗。

●为支持 DDR2-800 以上的速度，地址/命令/控制网络需要采用新的拓扑结构来提高信号的完整性。现在的 DDR2 DIMM 中，地址、命令总线所采用星型拓扑结构限制了存储器工作频率的进一步提高，也不符合信号的完整性需要。

●需要降低低核心电压，来抵消功耗的提高。由于 DRAM 芯片功耗与工作频率成正比等问题存在，要控制功耗，就需要降低核心电压来抵消功耗的提高，但这样又势必影响存储器的稳定性。

很显然，DDR2 标准已经跟不上处理器发展的趋势。正是由于 DDR2 本身设计上的限制，使其不能完全满足更高带宽平台的应用，新的存储器标准 DDR3 呼之欲出。

1.2 DDR 存储控制器的发展

众所周知，在计算机系统中，存储器性能会直接影响到处理器的性能发挥，因为处理器和存储器间的数据交换必须经过存储控制器，所以存储控制器技术就成为计算机整体性能发挥的一个关键环节。

AMD 公司刚推出的 AM2 处理器采用了 CPU 集成存储控制器技术。早期的存储控制器全部是集成在主板的北桥芯片中，AMD 改变了这一传统，那就是把存储控制器直接集成到 CPU 核心中。从理论上讲，CPU 集成存储控制器，CPU 和存储器之间的数据传输不再需要经过北桥芯片，因此可以缩短 CPU 与存储器之间的数据交换周期。但是由于 AMD 的存储控制器集成在 CPU 内部，因此存储器的工作频率与 CPU 相同，并且不能进行频率异步设置，在超频的时候会导致存储器的频率同 CPU 的频率一起升高，一旦超过存储器的承受能力，就会导致存储器无法工作，这会大大限制处理器的超频能力。

Intel 公司的 Conroe 处理器则使用北桥集成双通道存储控制器技术，其中存储控制器仍然设计在北桥芯片当中，它的双通道存储控制器的原理就是在北桥芯片中集成两个存储控制器，而且可以独立工作，CPU 能够分别在任一存储器通道中进行寻址、读取，这样就在理论上可以使存储器的带宽、传输速度增加一倍。但是

由于数据交换需要通过北桥，这无疑为处理器访问存储器带来更高的延时。

在目前Nehalem处理器中，英特尔又开发了名为QuickPath Interconnect（简称QPI）的总线技术^[5]。QPI是在处理器中集成了存储控制器的QuickPath体系架构，主要用于处理器之间和系统组件之间的互联通信（诸如I/O）。Nehalem处理器集成了全新的DDR3存储控制器，采用三通道控制器，因此存储器数据位宽从128位提升到了192位。存储控制器本身的带宽最大限度为每秒64GB。但在处理器采用了集成存储控制器后，它就能直接与物理存储器阵列相连接，从而极大程度上减少了存储器延迟。集成存储控制器对寄存器和寄存器的双列直插存储器模块都能提供支持，集成存储控制器可以支持DDR3-800，DDR3-1066，DDR3-1333 JEDEC标准，也为将来的升级预留了空间。

1.3 课题的研究目标、内容和意义

1.3.1 课题的研究目标、内容

本文从计算机硬件技术分析的角度，结合与DDR2的比较，对DDR3存储器的关键技术进行了初步系统地分析和研究。在此基础上，具体研究了DDR3存储控制器的总体结构、逻辑设计（包括传输层和物理层的模块分析）。研究的内容包括：

- 1、DDR3关键技术的分析；
- 2、DDR3存储控制器的总体功能和结构；
- 3、DDR3存储控制器传输层的设计实现（主要包括用户请求模块UIB、纠错回写控制单元SCRIB以及仲裁器的设计）；
- 4、DDR3存储控制器物理层的设计实现（主要包括DQS的相位控制以及数据的接受和发送）；
- 5、DDR3存储控制器的改进方法和发展方向。

1.3.2 课题意义

存储器是影响计算机系统整体性能的重要因素，它的数据传输率的提升对提高处理器性能有积极作用。如何能快速准确的给存储器发送命令并传输数据，是存储控制器的研究目的。

在存储控制器的设计中，访存请求的仲裁和调度对提高存储器的利用率和存储系统的性能有着至关重要的影响，是存储控制器设计的重点；在高频率工作条件下，访存数据的正确写入和准确可靠地接收是存储控制器设计的难点。

本文在深入研究分析DDR3关键技术的基础上，对DDR3存储控制器传输层和物理层实现进行了详细描述，并对存储控制器设计的重点和难点进行了重点阐述，提出了自己的创新观点，对同类设计的工程实现具有很好的借鉴意义和应用价值。

1.4 论文结构

整个论文的结构安排如下：

第一章为绪论，综合介绍了课题研究的背景、意义、主要研究的内容。

第二章结合DDR2存储器技术，对DDR3存储器关键技术进行了深入、全面地分析。

第三章对DDR3 SDRAM进行了综述，介绍了DDR3存储器的基本功能以及相关的定义和命令。

第四章详细论述了DDR3存储控制器传输层的设计实现方法，包括存储控制器的总体结构、各种逻辑模块和仲裁器的设计实现。

第五章详细论述了DDR3存储控制器物理层的设计实现，包括地址命令的发送、写数据发送和读数据的接收。

第六章 对未来技术进行了展望。

以上六章构成了论文的主体。

第二章 DDR3 技术分析

2.1 DDR3 存储器技术优势

与 DDR2 技术相比, DDR3 存储器技术具有明显的优势, 表 2.1 是 DDR3 和 DDR2 技术的比较。

表 2.1 DDR3 和 DDR2 基本规格对比

	DDR3	DDR2
电压 VDD/VDDQ	1.5V/1.5V (+/-0.075)	1.8V/1.8V (+/-0.1)
I/O 接口	SSTL_15	SSTL_18
数据传输率 (Mbps)	800/1066/1333/1600	400/533/667/800
容量标准	512MB-8GB	256MB-4GB
CL(CAS 延时)	5/6/7/8/9/10/11	3/4/5/6
AL(附加延时)	0/CL-1/CL-2	0/1/2/3/4
RL(读取延时)	AL+CL	AL+CL
WL(写入延时)	AL+CWL CWL=5/6/7/8	RL-1
预取设计(bit)	8	4
逻辑 Bank 数量	8 (512MB/1GB/2GB/4GB/8GB) 16	4 (256MB/512MB) 8 (1GB/2GB/4GB)
突发长度	BL4/BL8	BL4/BL8
封装	FBGA 78-ball:x4/x8 96-ball:x16	FBGA 60-ball:x4/x8 84-ball:x16
引脚标准	240Pin DIMM	240Pin DIMM

在 2007 年 Intel 开发论坛 (IDF) 上, 英特尔对 DDR3 存储器的优势以简明的文字作了概括:

性能与可扩充性

- DDR3 存储器目前可达到 1600MHz (与 DDR2 的 800MHz 相比)

- DDR3 技术具有可扩展性
- 最大超频性能: DDR2 1.2GT/s, DDR3 2GT/s+

更低功耗的架构

- 比 DDR2 减少 25%的功耗 (DDR2-800 与 DDR3-800 相比)
- DDR3 1066 的功耗比 DDR2 800 更低
- DDR3 休眠模式可以降低 15%的功耗

更高容量发展蓝图

- 采用更高密度设计
- 4Gb DRAM 将有可能实现

DDR3 存储控制器的具体优势主要表现在 8-bit 预取技术、低功耗设计技术、和相对于 DDR2 改进的其它一些重要变化。

2.2 8-bit 预取技术

DDR3 存储器颗粒有三种不同的频率指标, 它们分别是 DRAM 核心频率、时钟频率和(等效)数据传输频率^[6], 核心频率即为存储器 Cell 阵列(Memory Cell Array)的工作频率, 时钟频率则为存储器总线的工作频率、数据传输的 I/O Buffer 也受其控制, 而等效数据传输频率则是指数据传送的频率。早期的 SDRAM 数据传输频率与时钟周期同步, 并且也与 DRAM 核心频率相同。以 PC-133 SDRAM 为例, 它的核心频率、时钟频率与数据传输频率都是 133MHz。

由于 SDRAM 内部存储单元的核心频率提高比较困难且成本较高, 且 DDR2-800 的核心频率已经达到了 200MHz, 为了解决外部数据传输率和核心速度之间的矛盾, DDR3 引进了新的 8-bit Prefetch (数据预取架构) 技术来保证数据传输率持续的增长。

DDR3 800 内部 Cell (存储单元) 的核心频率仅为 100MHz, 采用 8-bit 的预取技术后, 却能提供和核心频率为 200MHz 的 DDR2-800 同样的带宽。也就是说引入 DDR3, 通过这种并行运行的方式来增加内部带宽, 可以让存储器频率提升回到一个新的起跑线。

图 1.1 介绍了 DDR、DDR2 与 DDR3 的结构对比以及相应的预取设计。

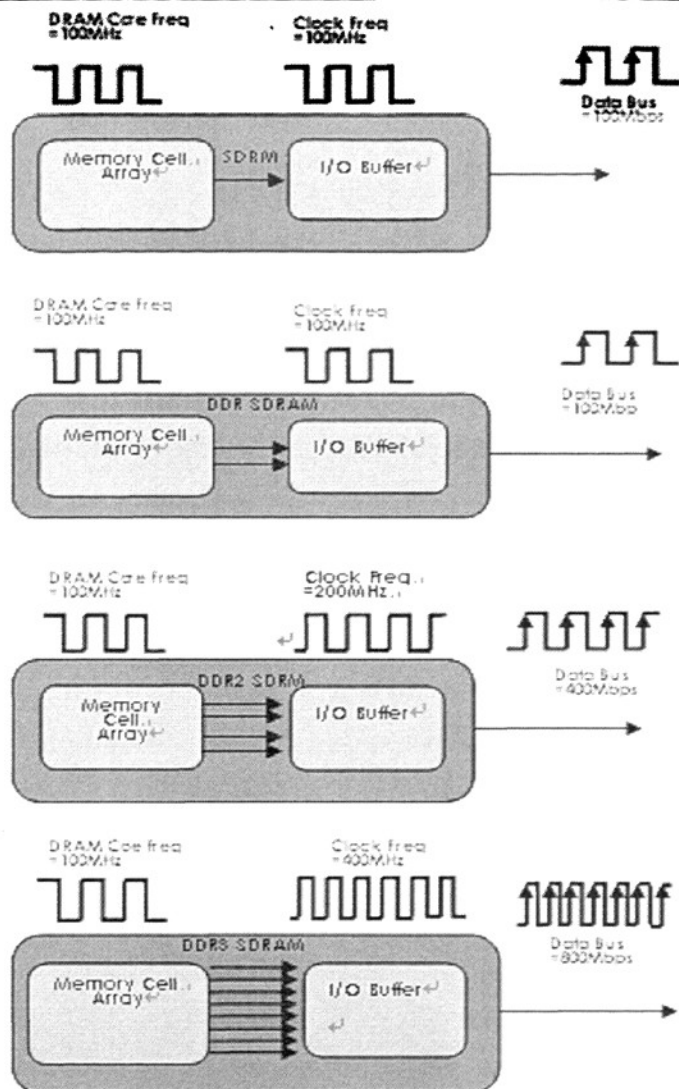


图 1.1 DDR、DDR2 与 DDR3 的结构对比以及相应的预取设计

DDR 存储器实现了双倍数据率传输 (Double Data Rating)，它可以在一个时钟周期传输两次数据，即在时钟的上升沿和下降沿分别激发一次，这样它的数据传输频率就达到时钟频率的两倍，而它的核心频率仍然与时钟频率相同，例如 DDR400 存储器，它的核心频率与时钟频率都只有 200MHz。双倍数据率意味着存储器核心要在一个时钟周期内供给两倍的数据，为了做到这一点，DDR 存储器引入了 2bit 预取设计，在预取机制中，存储器 Cell 阵列的数据先被输入到 I/O Buffer 缓冲中，然后再从 I/O Buffer 向存储器总线输出。而 2bit 预取则意味着每个时钟周期存储器 Cell 阵列都会发送 2bit 数据到 I/O Buffer 内暂存，以满足下个时钟周期的两次数据传送。

DDR2 在 DDR 基础上将性能提升了一倍，它的最高数据传输频率达到 800MHz，但 DDR2 800 的核心频率仍然只有 200MHz、也就是和 DDR400 相当，主要的原因在于 DDR2 引入了 4bit 预取和频率不对等的设计，控制 I/O Buffer 的时钟频率提升到核心频率的 2 倍，而数据传输频率仍为时钟频率的 2 倍，也就是说 DDR2 的数据传输频率达到核心频率的 4 倍。接下来的 DDR3 同样是沿着这条道路前进：它将预取位数提高到 8bit，并将时钟频率提升为核心频率的 4 倍——假设其核心频率为 100MHz，那么时钟频率就达到 400MHz，数据传输频率则高达 800MHz，这也就是 DDR3 800 的设计标准。

此外，由于 DRAM 内部存储单元的核心频率提高比较困难且成本较高，且 DDR2-800 的核心频率已经达到了 200MHz，为了解决外部数据传输率和核心速度之间的矛盾，DDR3 引进了新的 8-bit Prefetch（数据预取架构）技术来保证数据传输率持续的增长。

DDR3 800 内部 Cell（存储单元）的核心频率仅为 100MHz，采用 8-bit 的预取技术后，却能提供和核心频率为 200MHz 的 DDR2-800 同样的带宽。也就是说引入 DDR3，通过这种并行运行的方式来增加内部带宽，可以让存储器频率提升回到一个新的起跑线。图 2.1 是 DDR、DDR2 和 DDR3 预取技术的比较^[7]。

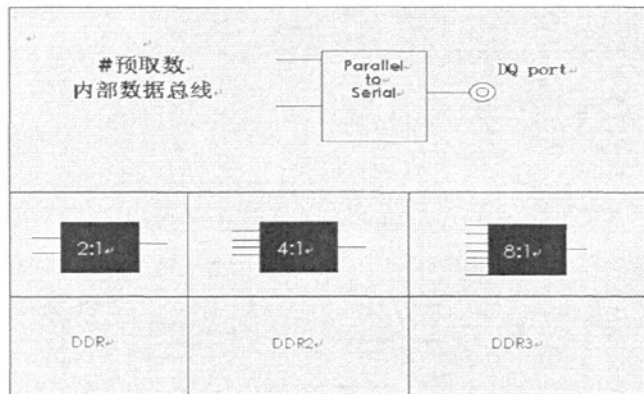


图 2.1 DDR、DDR2 和 DDR3 的预取比较

在 DDR2 时代，使用了 4-bit 预取技术，一次从存储单元预取 4-bit 的数据，然后在 I/O 时钟上升沿和下降沿传输出去，由于 4-bit 需要 2 个时钟周期才能完成传输，这样 DDR2 的 I/O 时钟频率为存储单元频率两倍。

由于 DDR3 采用了 8-bit 预取技术，一次从存储单元预取 8-bit 的数据，在 I/O 端口处上下沿触发传输，8-bit 需要 4 个时钟周期完成，所以 DDR3 的 I/O 时钟频率是存储单元核心频率的 4 倍，由于是上下沿都在传输数据，所以实际有效的数据传输频率达到了核心频率的 8 倍。

显然，通过使用 Prefetch 架构可以解决存储单元内部数据总线频率（核心频

率) 较低的瓶颈。8-bit 预取, 正是 DDR3 提升带宽的关键技术。同样的核心频率, DDR3 能提供两倍于 DDR2 的带宽。这样 DRAM 内核的频率只有接口频率的 1/8, DDR3 800 的核心工作频率只有 100MHz, 当 DRAM 内核工作频率为 200MHz 时, 接口频率已经达到了 1600MHz。表 2.2 是 DDR2 和 DDR3 带宽的比较。

表 2.2 DDR2 和 DDR3 提供的带宽比较

存储器	标准	核心频率 (MHZ)	I/O 频率 (MHZ)	有效传输频率 (MHZ)	带宽 (GB/s)
DDR2-667	PC2-5300	166	333	667	5.3
DDR2-800	PC2-6400	200	400	800	6.4
DDR3-800	PC3-6400	100	400	800	6.4
DDR3-1066	PC3-8500	133	533	1066	8.5
DDR3-1333	PC3-10600	166	667	1333	10.6
DDR3-1600	PC3-12800	200	800	1600	12.8

在同样核心频率下, DDR3 能提供两倍于 DDR2 的带宽。在这种情况下, 如果 DDR3 的核心频率为 200MHz, 那么数据传输频率可以达到 1600MHz, 这样带宽就能达到惊人的 12.8GB/s。

2.3 DDR3 的低功耗设计技术

存储器频率越来越高, 带来的负面影响就是功耗的增加, 新一代 DDR3 存储器为降低功耗作了一定的改进, 以抵消频率提高带来的负面影响^[6]。减少功耗最通常的方式就是降低存储器的核心电压。DDR SDRAM 的核心电压为 2.5V, DDR2 的核心

电压为 1.8V, DDR3 的核心电压进一步降低为 1.5V^[9]。此外, DDR3 还采用复位控制、根据温度自刷新、局部自刷新等其它技术进一步降低功耗^[10]。这样, DDR3 存储器模块拥有比 DDR2 更好的带宽功耗比, 对比现有的 DDR2-800 产品, DDR3-800、1066 的带宽功耗比分别为 0.72、0.83, 不单存储器带宽大幅提升, 功耗表现也比 DDR2 更好, 体现出比 DDR2 更大的优越性。

2.3.1 复位 (Reset)

复位是 DDR3 新增的一项重要功能, 复位使 DDR3 的初始化处理变得简单。当 Reset 命令有效时, DDR3 存储器将停止所有的操作, 并切换至最少量活动的状态, 以节省电力。在复位期间, DDR3 存储器将关闭内在的大部分功能, 所有数据接收与发送器都将关闭。所有内部的程序装置将复位, DLL (延迟锁定回路) 与时钟电路将停止工作, 而且不理睬数据总线上的任何动静, 以使 DDR3 达到最节省电力的目的。

2.3.2 根据温度自刷新 (ASR)

刷新操作分为两种: 自动刷新 (Auto Refresh, 简称 AR) 与自刷新 (Self Refresh, 简称 SR)。为了最大的节省电力, DDR3 采用了一种新型的自动刷新设计 (ASR, Automatic Self-Refresh)。当开始 ASR 之后, 将通过一个内置于 DRAM 芯片的温度传感器来控制刷新的频率, 温度较低时, 刷新频率比较慢, 随着温度的升高, 刷新的频率越来越快。这样在保证数据不丢失的情况下, 尽量减少刷新频率, 降低功耗。

此外, 由于 DDR3 的 ASR 是可选设计, 并非市场上所有的 DDR3 存储器都支持这一功能, 因此还有一个附加的功能就是自刷新温度范围 (SRT, Self-Refresh Temperature)。通过模式寄存器, 可以选择两个温度范围, 一个是普通的温度范围 (例如 0℃至 85℃), 另一个是扩展温度范围, 比如最高到 95℃。对于 DRAM 内部设定的这两种温度范围, DRAM 将以恒定的频率和电流进行刷新操作。

2.3.3 局部自刷新 (PASR, Partial Array Self-Refresh)

这是 DDR3 的一个可选功能, 通过这一功能, DDR3 存储器芯片可以只刷新部分逻辑 Bank, 而不是全部刷新, 从而最大限度的减少因自刷新产生的电力消耗。这个功能对于移动计算的意义更为重要, 使得笔记本电脑可以在系统未处于最高性能模式下时, 通过更少的刷新节省更多的电能。

DDR3 在降低功耗方面作了许多努力，一方面能保证它可以达到更高的频率，另一方面，低功耗也可以让 DDR3 在小型移动设备中得到广泛应用。

2.4 DDR3 的其它重要特点

2.4.1 高密度大容量

支持高密度的存储器颗粒也是 DDR3 的重要特点之一。和 DDR2 的 256MB~4GB 的容量相比，DDR3 最高能达到 8GB 的容量。8GB 仅仅是在 8 个逻辑 Bank 情况下达到的容量，实际上 DDR3 为了应对未来大容量芯片的需求，还为未来的 16 个逻辑 Bank 做好了准备。

随着高密度 DDR3 存储器颗粒的普及应用，可大大降低相同容量存储器的制造和封装成本，并进一步推动 DDR3 存储器的普及速度。

从性能的角度来考虑，拥有更多的逻辑 Bank 数则意味着具有较少的 Bank 冲突，寻址流更加短暂、随意，从而提高寻址命中率和降低访问延时。

2.4.2 点对点连接 (Point-to-Point)

这是为了提高系统性能而进行的重要改动，也是与 DDR2 系统的一个关键区别。在 DDR3 系统中，一个存储控制器将只连接一个存储器。因此存储控制器与 DDR3 存储器模组之间是点对点 (P2P, Point-to-Point) 的关系 (单物理 Rank 的模组)，或者是点对双点 (P22P, Point-to-two-Point) 的关系 (双物理 Rank 的模组)，从而大大减轻了地址/命令/控制与数据总线的负载，提高了信号的完整性。

在减少存储控制器所支持 DIMM 数量的同时，DDR3 DIMM 地址/命令/控制总线的布线方式也发生了变化，以适应高速信号传输。在 DDR2 DIMM 中，为了减小地址/命令/控制总线到达各个 DRAM 存储器芯片的时间偏差，DIMM 上的布线采用了星型连接，使得信号线到达各个 DRAM 芯片的长度相等，如图 2.2 (a) 所示。这样又带来了分支线的问题，影响了信号传输质量。在 DDR3 DIMM 中，采用了“fly by”的布线方式，信号从 DIMM 中部进入后到达一端，按顺序依次连接各个 DRAM 芯片的信号引脚，取消了信号的分支，如图 2.2 (b) 所示。这样有利于提高信号完整性，而带来的时间偏差则通过读写调校 (leveling) 技术得到弥补。

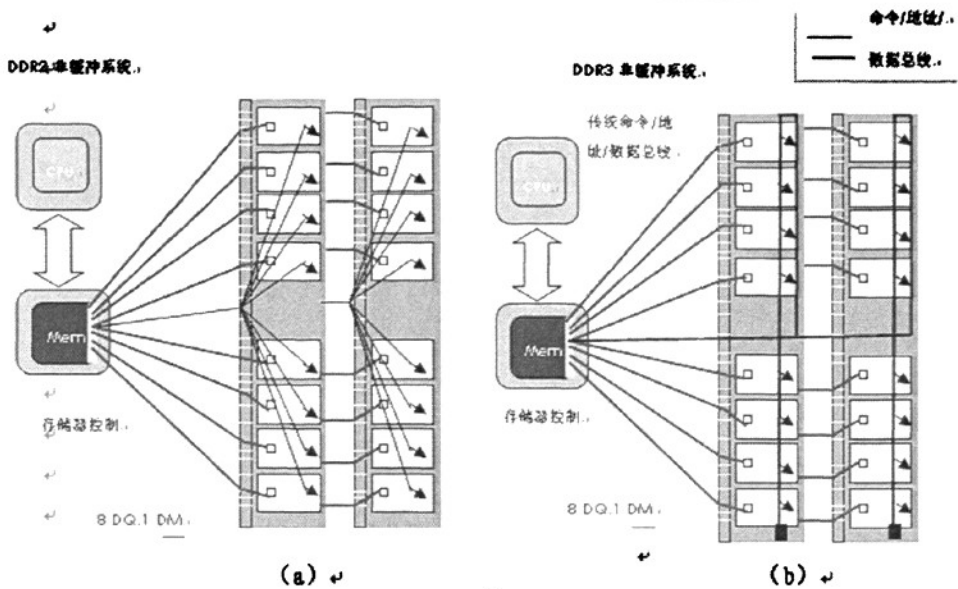


图 2.2 DDR3 总线拓扑变化：从“星”型变为“Fly-by”（点对点）型

2.4.3 突发长度 (BL, Burst Length)

由于 DDR3 的预取为 8bit，所以突发传输长度 (BL, Burst Length) 也固定为 8，而对于 DDR2 和早期的 DDR 架构的系统，BL=4 也是常用的，DDR3 为此增加了一个 4-bit Burst Chop (突发突变) 模式，即由一个 BL=4 的读取操作加上一个 BL=4 的写入操作来合成一个 BL=8 的数据突发传输，届时可通过 A12 地址线来控制这一突发模式。而且需要指出的是，任何突发中断操作都将在 DDR3 存储器中予以禁止，且不予支持，取而代之的是更灵活的突发传输控制。

对于连续传输的数据，如果有不需要的部分，以往的处理是需要利用突发中断等操作来对数据进行屏蔽。为了克服以往突发传输控制不灵活的缺陷，DDR3 不再支持任何突发中断操作，而改用顺序突发等更灵活的突发传输来进行控制。

2.4.4 封装 (Package)

DDR3 由于新增了一些功能，所以在引脚方面会有所增加，4/8bit 芯片采用 78 球 FBGA 封装，16bit 芯片采用 96 球 FBGA 封装，并且 DDR3 必须是绿色封装，不含任何有害物质。

2.4.5 ZQ 校准

ZQ 也是一个新增的引脚，在这个引脚上接有一个 240 欧姆的低公差参考电阻。通过一个 DRAM 命令，片上校准引擎（ODCE, On-Die Calibration Engine）来自自动校验数据输出驱动器导通电阻与 ODT 的终端匹配电阻值。当系统发出这一命令之后，将用相应的时钟周期（在加电与初始化之后用 512 个时钟周期，在退出自刷新操作后用 256 时钟周期、在其他情况下用 64 个时钟周期）对导通电阻和 ODT（On-Die Termination）电阻进行重新校准。

2.4.6 参考电压分成两个

对于存储器系统工作非常重要的参考电压信号 VREF，在 DDR3 系统中将分为两个信号：一个是为命令与地址信号服务的 VREFCA，另一个是为数据总线服务的 VREFDQ，它将有效的提高系统数据总线的信噪等级。

第三章 DDR3 SDRAM 存储器概述

DDR3 SDRAM存储器是从DDR、DDR2发展而来的一种高速同步动态随机访问存储器。DDR3 SDRAM存储器内部由8个存储体(bank)组成,每个存储体又由若干行(row)和列(col)构成一个存储单元阵列,存储器中信息就存放在这些存储单元中。要访问存储器的内容,就要通过存储器访问命令来寻址到存储单元。基本的存储器访问命令分为激活、读和写命令。激活命令和行地址一起发送到DRAM存储器,以打开一个存储体行,读或写命令和列地址一起发送到DRAM,以确定具体的存储单元,这样便可以对DRAM存储器进行读写访问了。

DDR3 SDRAM存储器的访问必须按照特定的规则进行,必须符合存储器工作的状态要求,图3.1是DDR3控制器中可能存在的状态转换和控制这些状态转化的命令之间的关系。

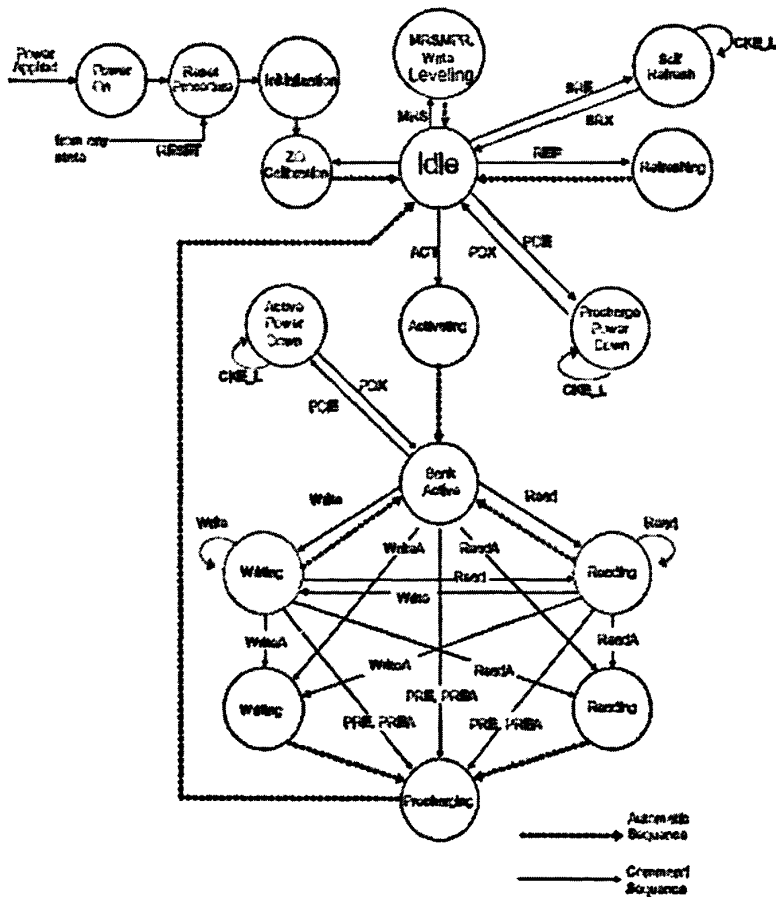


图 3.1 DDR3 状态简图

表3.1给出了上面的状态图中包含的命令及其功能描述。

表 3.1 DDR3 状态图部分命令

命令缩写	功 能	命令缩写	功 能	命令缩写	功 能
ACT	Active	Read	RD, RDS4, RDS8	PDE	Enter Power-down
PRE	Precharge	Read A	RDA, RDAS4, RDAS8	PDX	Exit Power-down
PREA	Precharge All	Write	WR, WRS4, WRS8	SRE	Self-Refresh entry
MRS	Mode, Register Set	Write A	WRA, WRAS4, WRAS8	SRX	Self-Refresh exit
REF	Refresh	RESET#	Start, RESET Procedure	MPR	Multi-Purpose Register

DDR3芯片内部控制器中同时可有八个Bank处于开放状态，Bank开放的顺序来源于控制器发出的命令。如果八个Bank已经处于开放状态，而且还有数据要进入Bank，那么这八个Bank中最后使用的Bank将会被关闭，从而再打开一个新的Bank。在自刷新的时候，所有的Bank都会被关闭，一旦有命令出现，Bank就会开放^[11]。

在DDR3控制器发送命令到存储器之前要做好相关的准备工作：

1、控制器要对放在FIFO中的地址进行解码。

2、如果Bank和行没有打开，控制器就必须打开Bank中的一个行。如果使用已经打开Bank中不同的行，控制器就会关闭这个已经打开的行而重新打开一个新行。如果Bank已经是打开状态或者是控制器在打开Bank之后，控制器就会转入读/写状态。

3、在进入写状态之后，如果DDR3控制器接受到一个读命令，那么它在发送这个读命令之前需要等待一段时间，等待时间的最小长度一般称为write_to_read time。同样地，在读状态的时候，如果DDR3控制器从命令逻辑块中得到一个写命令，它也会在发送这个写命令之前等待一段时间，等待时间的最小长度一般称为read_to_write time。在读或者写状态，DDR3还担负着裁定地址FIFO的读使能（信号）任务，从而获得下一个地址。

4、命令在被发送到DDR3存储器之前，会被流水发送，从而实现与地址信号同步。

3.1 加电和初始化

众所周知，DDR2 SDRAM的读写访问是猝发式的^[12]，数据长度为猝发长度BL，

BL可配置为4或者8。读写访问开始于激活命令，读写命令紧随其后^[13]。DDR3 SDRAM也是一样，激活命令给出的地址位用来选择要访问的存储体和行（BA0-BA2选择存储体，A0-A15选择行），读写命令给出的地址位用来选择猝发访问的起始列地址和是否自动预充电。

DDR3 SDRAM必须以预先确定的顺序加电和初始化，配置一些基本的操作参数，否则会导致不确定的操作^[14]。系统初始化的过程如下：

- 1、电源电压和时钟稳定后，维持复位信号有效至少 200 微秒，赋能 CKE（时钟使能）为高电平；
- 2、CKE 变为低电平，10 纳秒后复位信号失效，再等待 500 微秒，直到 CKE 变为高电平；
- 3、至少发送一个 NOP（空操作）命令或 DESEL（非选中）命令；
- 4、执行扩展模式寄存器 EMR2 设置命令；
- 5、执行扩展模式寄存器 EMR3 设置命令；
- 6、运行扩展模式寄存器 EMR1 设置命令，使能存储器芯片中的 DLL；
- 7、执行扩展模式寄存器 EMR 设置命令，复位存储器芯片中的 DLL；
- 8、执行 ZQCL 命令，进行 ZQ 校准；
- 9、等待锁相环相位锁定和 ZQ 校准完成。

完成上述初始化步骤之后，存储器便进入就绪状态，等待控制器的访问命令。

3.2 配置模式寄存器

在 DDR3 SDRAM 芯片中，设有模式寄存器和扩展模式寄存器，用来预先定义 SDRAM 的操作模式，这些寄存器是没有初始值的，必须在芯片加电之后，通过模式寄存器设置命令（MRS）对这些值进行设置。

模式寄存器

模式寄存器通常用来定义 DDR3 SDRAM 特殊的操作模式。包括突发长度，突发类型，CAS 延迟和操作模式。图 3.2 显示了模式寄存器的有关特征。Bank 地址 BA2、BA1 和 BA0 决定了模式寄存器的选择。

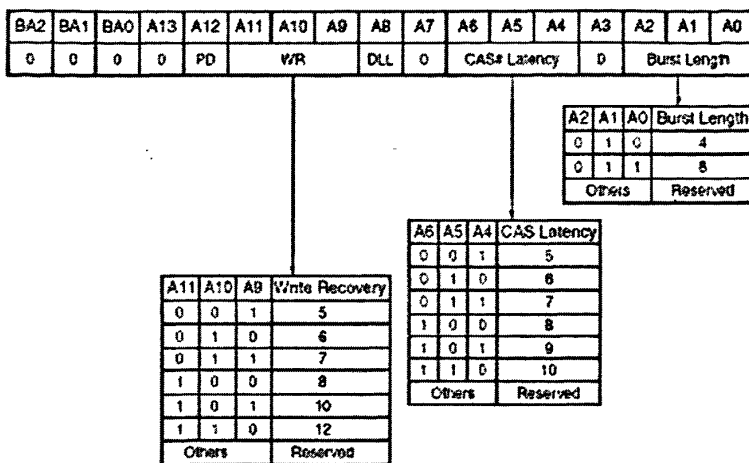


图 3.2 模式寄存器

与模式寄存器相对应的 Bank 地址二进制数位配置如表 3.2 所示。

表 3.2 Bank 地址二进制数位配置

BA1	BA0	Mode Register
0	0	Mode Register (MR)
0	1	EMR1
1	0	EMR2
1	1	EMR3

扩展模式寄存器

除了模式寄存器控制的功能外，扩展模式寄存器控制还具备以下功能：数字锁相环 DLL 有效或无效；输出驱动能力；片上端接电阻 (ODT) 控制；伴随 CAS 的附加延迟；片外驱动器校准；DQS 有效或无效；RDQS/RDQS 有效或无效以及 OUTPUT 有效或无效。具体见表 3.3、表 3.4 所示。

表 3.3 扩展模式寄存器 (EMR)

BA2	BA1	BA0	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	1	0	Qoff	TDQS	0	RTT	OCD Program		RTT	ODS	AL		RTT	ODS	DLL

表 3.4 扩展模式寄存器 2 (EMR2)

BA2	BA1	BA0	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	0	0	RTTWR		0	SRT	ASR	CWL			0	0	0

Bank地址位被设置到11 (BA1和BA0被设置为高位)。地址位全部设置成低位。

3.3 DDR3 命令

DDR3 存储器的工作和状态转换是通过命令来控制的。DDR3 存储器能够支持的命令主要有：

预充命令

预充命令 (precharge) 用来使在特定的 Bank 中打开的行无效。在预充命令发送之后的一个特定的时间后 (t_{RP}) 该存储体才可用。输入端 A10 决定对一个还是所有的存储体进行预充。一个存储体被预充之后, 进入空闲状态, 等待接收激活命令。

自动刷新命令

DDR3 是动态存储器, 必须要定期进行刷新才能维持其存储的内容。刷新间隔和 DDR3 存储器芯片的温度有关, 当温度在 $0-85^{\circ}\text{C}$ 时, 刷新间隔为 $7.8\mu\text{s}$, 当温度在 $85-95^{\circ}\text{C}$ 时, 刷新间隔为 $3.9\mu\text{s}$ 。

激活命令

在任何读写命令被发送到 DDR3 SDRAM 存储器的行上之前, Bank 中的行必须用激活 (Active) 命令进行激活。在列被激活后, 读写命令可根据 t_{RCD} 说明发送到列对象上。DDR3 SDRAM 设备同样支持附加在 (posted) CAS 的附加延迟。这些设备允许在 t_{RCD} 说明之前发送一个读或者写命令, 通过使用附加的延迟时钟周期延迟内部设备实际记录的读写命令。

读命令

读命令通常用来启动一个突发的存储器读操作, 以访问一个激活的行。BA0-BA2用来选择存储体Bank地址。A0 - Ai提供的输入地址用来选择开始列位置。在读操作完成之后, 这个行在随后的访问中仍然是活跃的, 直到该行被预充命令关闭。

图3.3展示的是一个潜在的附加延迟值为0的读命令的操作, 因此, 在这个例子中, 数据读出的延迟为5, 和CAS的读延迟相同。

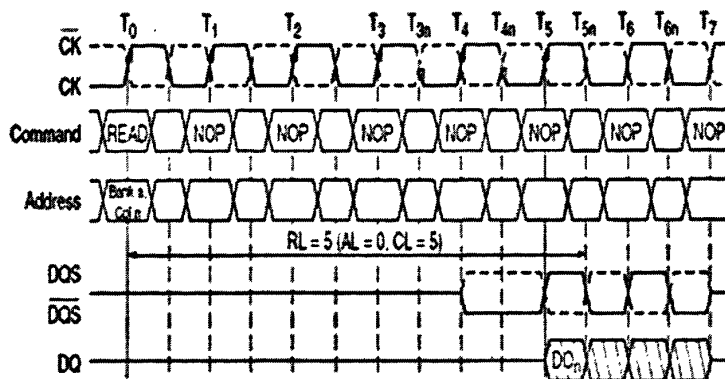


图 3.3 读命令示例

写命令

写命令用来启动一个突发的存储器写操作，BA0-BA2的值用来选择存储体，地址输入A0到Ai用来选择活跃行的起始列地址。DDR3 SDRAM写操作中的写延迟和读延迟相等。

写延迟 (WL) = 读延迟 = 附加延迟 + CAS 延迟

图 3.4 展示的是写延迟为 5 的示例，写命令到第一次在 DQS 信号上升边沿的时间长度取决于 WL。

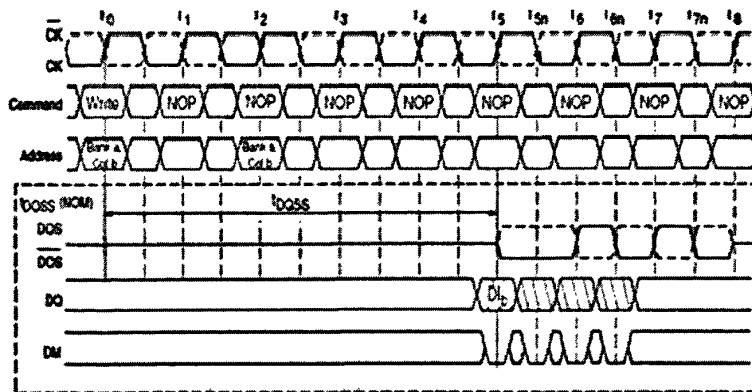


图 3.4 存储器写操作

DDR3 SDRAM命令的种类和操作见表3.5命令真值表：其中[BA=Bank地址，RA=R行地址，CA=列地址，BC=突发Chop，X=任意值，V=有效]

表 3.5 DDR3 命令的种类和操作真值表

Function	Abbreviation	CKE		\overline{CS}	\overline{RAS}	\overline{CAS}	\overline{WE}	BA0	A13	A12	A10	A0
		Previous Cycle	Current Cycle					-	-	/	/	-
								BA3	A15	BC	AP	A9,A11
寄存器组模式	MRS	H	H	L	L	L	L	BA	OP Code			
刷新	REF	H	H	L	L	L	H	V	V	V	V	V
自刷新入口	SRE	H	L	L	L	L	H	V	V	V	V	V
自刷新出口	SRX	L	H	H	V	V	V	V	V	V	V	V
				L	H	H	H					
单一 Bank 预冲	PRE	H	H	L	L	H	L	BA	V	V	L	V
所有 Bank 预冲	PREA	H	H	L	L	H	L	V	V	V	H	V
激活 Bank	ACT	H	H	L	L	H	H	BA	Row Address (RA)			
写命令 (BL8 or BL4)	WR	H	H	L	H	L	L	BA	RFU	V	L	CA
写命令 (BL4, on the Fly)	WRS4	H	H	L	H	L	L	BA	RFU	L	L	CA
写命令 (BL8, on the Fly)	WRS8	H	H	L	H	L	L	BA	RFU	H	L	CA
写命令和预冲(BL8 or BL4)	WRA	H	H	L	H	L	L	BA	RFU	V	H	CA
写和自预冲(BL4, on the Fly)	WRAS4	H	H	L	H	L	L	BA	RFU	L	H	CA
写和自预冲(BL8, on the Fly)	WRS8	H	H	L	H	L	L	BA	RFU	H	L	CA
写和自预冲(BL8 or BL4)	WRA	H	H	L	H	L	L	BA	RFU	V	H	CA
写和自预冲(BL4, on the Fly)	WRAS4	H	H	L	H	L	L	BA	RFU	L	H	CA
读命令 (BL8, on the Fly)	RDS8	H	H	L	H	L	H	BA	RFU	H	L	CA
读命令和自预冲(BL8 or BL4)	RDA	H	H	L	H	L	H	BA	RFU	V	H	CA
读和自预冲(BL4, on the Fly)	RDAS4	H	H	L	H	L	H	BA	RFU	L	H	CA
读和自预冲(BL8, on the Fly)	RDAS8	H	H	L	H	L	H	BA	RFU	H	H	CA
空操作	NOP	H	H	L	H	H	H	V	V	V	V	V
取消选定	DES	H	H	H	X	X	X	X	X	X	X	X
ZQ 校准 Long	ZQCL	H	H	L	H	H	L	V	V	V	H	V
ZQ 校准 Short	ZQCS	H	H	L	H	H	L	V	V	V	L	V
进入省电模式	PDE	H	L	L	H	H	H	V	V	V	V	V
				H	V	V	V					
退出省电模式	PDX	L	H	L	H	H	H	V	V	V	V	V
				H	V	V	V					

3.4 Write leveling

为了获得更好的信号完整性，DDR3存储器模块的命令、地址、控制信号和时钟都采用了新颖的Fly-by拓扑架构^[16]。Fly-by拓扑架构的好处在于它减少了stub的数量和长度，但同时，它也造成了DIMM上每个DRAM时钟和闸门之间的时间离散。这无疑加大了控制器对tDQSS、tDSS、tDSH规范的控制难度。为此，DDR3控制器需要采用“写调校”技术来弥补Fly-by拓扑架构带来时间离散这个负面影响。

控制器能通过用写调校技术来侦测DDR3 SDRAM的反应，从而以此来校正 $DQS - \overline{DQS}$ 到 $CK - \overline{CK}$ 之间的关系。存储控制器在 $DQS - \overline{DQS}$ 时钟上必须可调延迟，从而实现 $DQS - \overline{DQS}$ 时钟的上升边沿和DRAM pin中的时钟上升边沿保持对齐。和 $DQS - \overline{DQS}$ 上升沿一样，DRAM也是通过DQ总线来异步反馈 $CK - \overline{CK}$ 信号。控制器在状态从0转变成1之前会反复延迟 $DQS - \overline{DQS}$ 时钟。

$DQS - \overline{DQS}$ 延迟虽然已经确定，但是它也必须按照tDQSS规范来实现。除tDQSS之外，tDSS和tDSH规范也需要实现。达到这样目的的一种方法就是通过结合当前运用的tDQSS与适当的占空因数和 $DQS - \overline{DQS}$ 信号的跳动来实现。

根据实际运用中的tDQSS，为了更好地满足tDSS和tDSH规范，tDQSL and tDQSH的实际值毫无疑问地要比“AC Timing Parameters”的绝对限制要好，理论上的配置时间如图3.5所示。

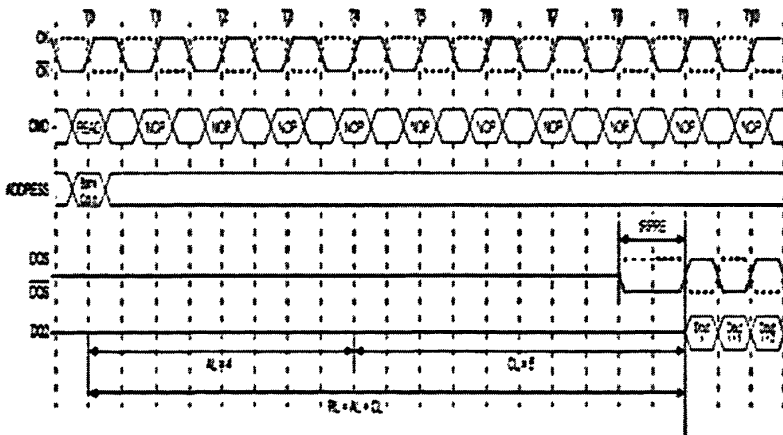


图 3.5 理论配置时间

第四章 DDR3 存储控制器传输层的设计实现

存储控制器连接上层用户逻辑和系统存储器，主要功能是接收上层用户逻辑发来的访存请求，并根据请求的优先级进行访存调度，按照存储器的时序要求，转换成存储器可以识别的各种命令，发送到存储器，最终完成对存储器数据的读写操作。本章我们提出了一种高效 DDR3 存储控制器设计方案，并给出了其传输层几个关键模块的设计实现细节。

4.1 存储控制器总体结构

存储控制器从层次上可分为传输层和物理层^[16]，传输层的主要功能是接收上层用户逻辑的访存请求，并将请求按照 DDR3 存储器所需要的节拍时序发送到物理层，并接收物理层返回的读数据，再按照上层逻辑所需要的时序发送给上层用户逻辑。物理层的主要功能是将传输层送来的访存命令、地址、数据等信息，按照 DDR3 的所规定的格式和电气特性发送到 DDR3 芯片，并接收 DDR3 的读出数据。图 4.1 介绍了 DDR3 存储器的总体结构。

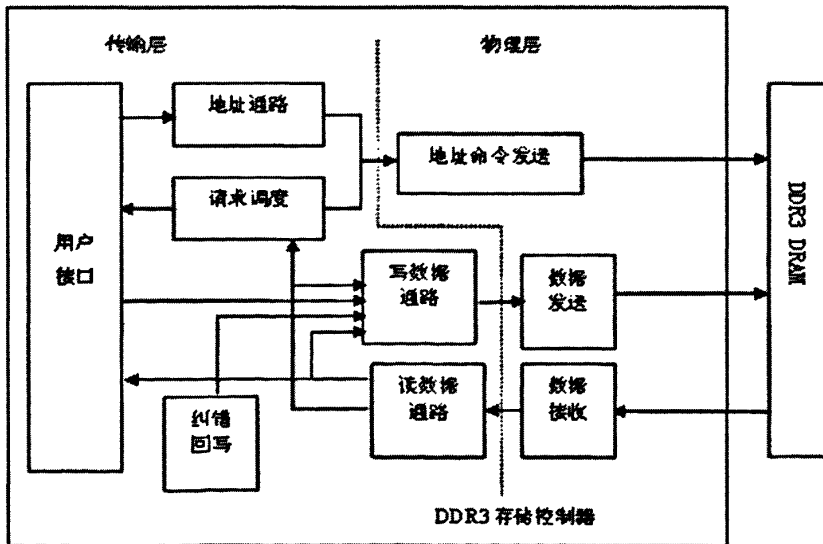


图 4.1 存储控制器总体结构

存储控制器在结构上主要由用户接口、请求调度、纠错回写、地址通路、写数据通路、读数据通路、地址命令发送、数据发送、数据接收等模块组成，如图

4.1 所示。其中，地址命令发送、数据发送、数据接收模块属于物理层，其他模块则属于传输层。各模块的主要功能描述如下：

●用户接口：接收来自上层用户逻辑部件的存储器请求，完成请求命令和数据的校验，将访存请求的地址转换成存储器的行地址、列地址和体地址，实现信号的跨时钟转换，并将请求转换成存储器的格式。

●请求调度：接收用户接口送来的访存请求，对请求进行分类管理，根据存储器当前状态，按照存储总线最大利用率原则，对访存请求进行排队和调度。该模块还完成存储器的初始化和刷新控制等操作。

●纠错回写：为了提高存储器的可靠性，控制器以一定的时间间隔发出存储器读请求，并对可以纠正的错误进行纠正后，将数据写回原处。

●地址通路：完成访存地址的寄存和判越界。

●写数据通路：保存用户接口送来的写入数据，并形成纠错码（ECC）。

●读数据通路：接收存储器的读出数据，进行数据的纠错和检错，并将数据返还给用户接口模块。

●地址命令发送：将 DRAM 命令、行列地址等控制信息发送到存储器。

●数据发送：将写存储器数据转换成 DDR 数据，并和选通信号一起发送到存储器。

●数据接收：接收存储器读出的数据，将 DRAM 的 DDR 数据转换成控制器内部时钟同步的数据。

4.2 用户接口模块 UIB

用户接口接收来自用户层（这里为上层用户逻辑）的读写请求和写入数据，然后对请求地址按照用户设定的模式进行地址变换，并将请求的命令和变化后的地址送到请求调度模块，将写入数据送到写地址通路模块。请求是顺序接收和顺序返回的^[17]。上层用户逻辑送来的写数据带有错误校验码（ECC），UIB根据数据重新形成ECC，并和上层用户逻辑来的ECC进行比较，以此完成上层用户逻辑数据的校验。UIB还接收读数据通路来的存储器读数据，并形成上层用户逻辑所需的ECC，一同送给上层用户逻辑。

由于上层用户逻辑的工作频率比存储控制器的工作频率高，这两部分的信号进出必须要进行时钟域的转换，包括请求信号、读数据、写数据。

图4.2为UIB的总体结构示意图。

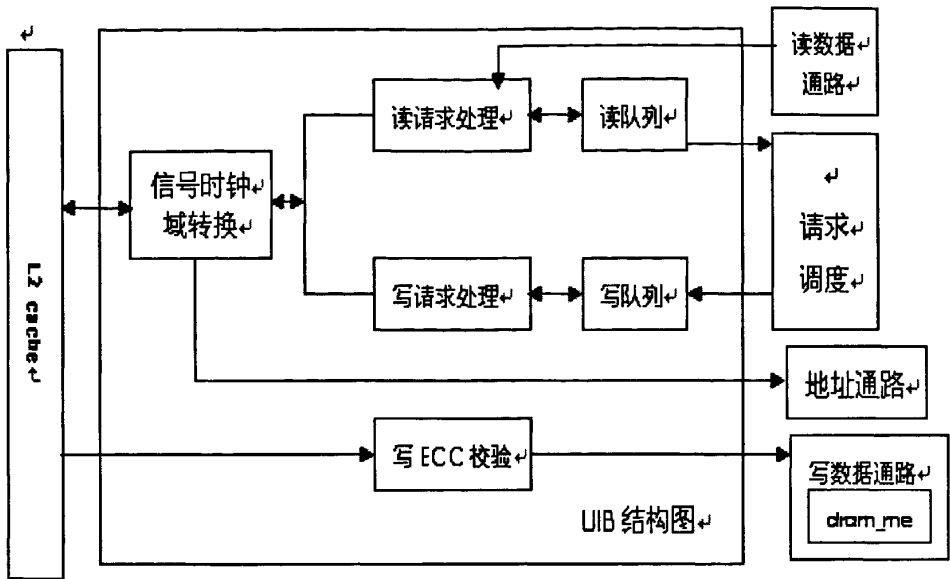


图 4.2 UIB 逻辑结构图

4.2.1 信号时钟域转换模块

该模块除了对读写请求进行处理外，还需要对信号和数据的时钟域进行切换。上层用户逻辑采用 `cmp_clk` 时钟频率对发来的请求信号、地址和数据转换到 DDR3 DRAM 控制器工作的 DRAM 时钟频率。同理，DDR3 DRAM 控制器返回的数据和信息也必须转换到 `cmp_clk` 时钟频率才能被上层用户逻辑正确接收。

信号的时钟域转换通过用同步信号作为采样寄存器使能端的方式完成，该模块中对应 `cpu` 时钟域和 DRAM 时钟域的切换有两个同步信号：

- `l2if_DRAM_rx_sync`: DRAM 时钟域到 `cpu` 时钟域的同步信号；
- `l2if_DRAM_tx_sync`: `cpu` 时钟到 DRAM 时钟的同步信号；

图 4.3 是 DRAM 时钟域到 `cpu` 时钟域的转换电路，同步信号 `l2if_DRAM_rx_sync` 控制寄存器的采样使能完成信号的跨时钟域采样。

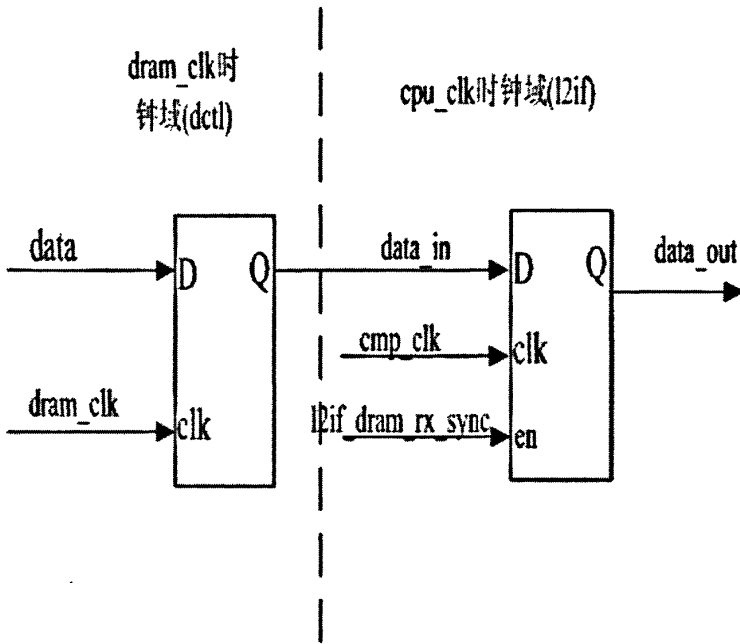


图 4.3 DRAM 时钟域到 cpu 时钟域的转换电路

信号转换时序图如 4.4 所示。同步信号 `l2if_DRAM_rx_sync` 是以 `DRAM_clk` 为周期的脉冲信号，脉冲宽度为 1 个 `cpu` 时钟周期。当 `l2if_DRAM_rx_sync` 脉冲将寄存器使能后，寄存器在 `cpu` 时钟的上升沿将 `DRAM` 时钟域的输入信号 `DRAM_in` 采样。

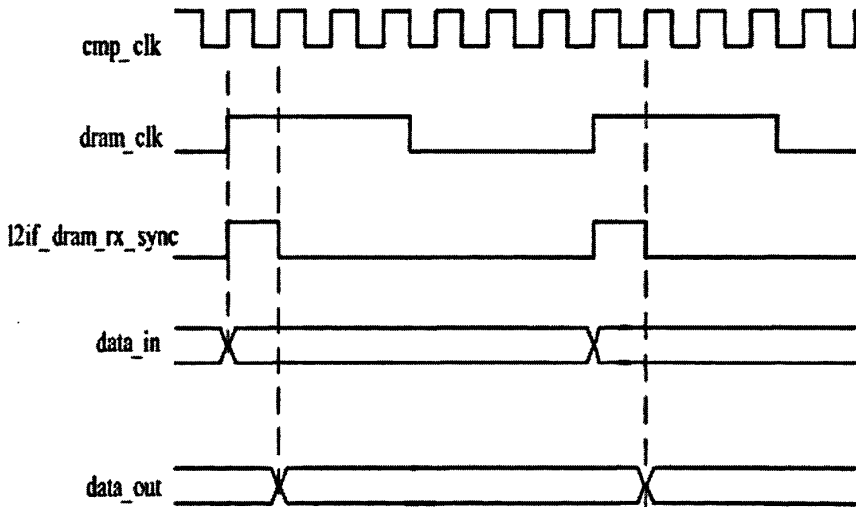


图 4.4 DRAM 时钟域到 cpu 时钟域的转换时序

图 4.5 是 cpu 时钟域到 DRAM 时钟域的转换电路,同步信号 l2if_DRAM_tx_sync 控制寄存器的采样使能完成信号的跨时钟域采样。

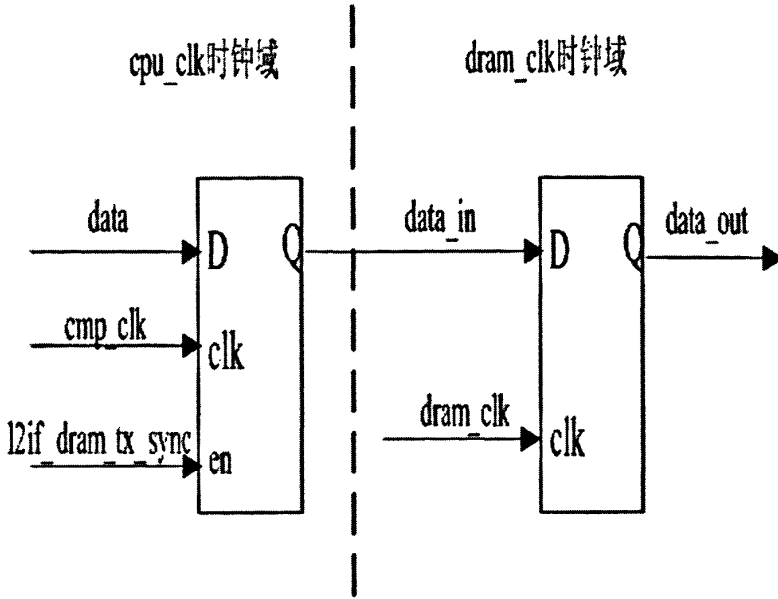


图 4.5 cpu 时钟域到 DRAM 时钟域的转换电路

信号转换时序图如 4.6 所示。同步信号 l2if_DRAM_tx_sync 是以 DRAM_clk 为周期的脉冲信号,脉冲宽度为 1 个 cpu 时钟周期。当 l2if_DRAM_tx_sync 脉冲将寄存器使能后,寄存器在 cpu 时钟的上升沿将数据数据采样。然后同步信号关闭,寄存器保持其采样值,直到后级寄存器在 DRAM 时钟的上升沿把数据采样输出到 DRAM 时钟域。

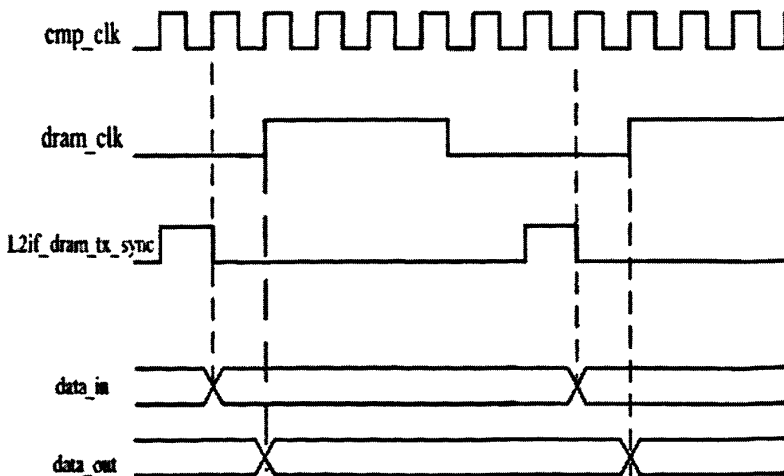


图 4.6 cpu 时钟域到 DRAM 时钟域的转换时序

4.2.2 读请求处理模块

该模块进行读请求处理时所执行的操作包括：读请求接收、读请求应答、读出数据 ECC 形成和数据返回。

· UIB 模块是 DRAM 控制器与上层用户逻辑之间的接口。在正常工作时，UIB 模块主要实现以下功能：

- 接收上层用户逻辑发送的读、写请求信息，并返回应答信号；
- 完成请求物理地址到 DRAM 内存实际地址的映射；
- 请求信号及读、写数据的传输和时钟域转换。

图 4.7 表示了 UIB 模块进行读请求处理时所执行的操作，包括：读请求处理、物理地址映射、读入数据的 ecc 校验和数据偏移地址返回。

1、读请求处理：对于正常读请求，UIB 模块考察读等待队列的填充情况。如果读等待队列已经填满则 UIB 模块不向上层用户逻辑发送应答信号，如果读等待队列不满则向上层用户逻辑发送应答信号。在发送应答信号的同时，UIB 模块向 dram_que 模块发送读请求信号和完成映射后的地址。

2、物理地址映射：UIB 模块接收到上层用户逻辑发送的请求地址后，根据从 dram_que 模块的配置寄存器读入的内存配置信息，将请求地址映射成 DRAM 内存中的实际地址，并在发送请求应答信号的同时将映射后的地址发送到 dram_que 模块中。

3、读入数据的 ecc 校验：从 dram_dp 模块中返回的读入数据是 256 位宽，在 UIB 模块中将数据拆分成两个 128 位的数据并分别由 ecc 校验单元进行校验。校验输出结果包括校验后数据、发生多个错误的标识信号、发生单个错误的标识信号和错误位置信号。

4、数据偏移地址返回：由于内存采用猝发方式读取数据，一次读请求将会返回 4 个 128 位数据。因此，UIB 模块在向上层用户模块返回 4 个读取数据的同时也返回数据的偏移地址（CAS 地址的第[1:0]位）。

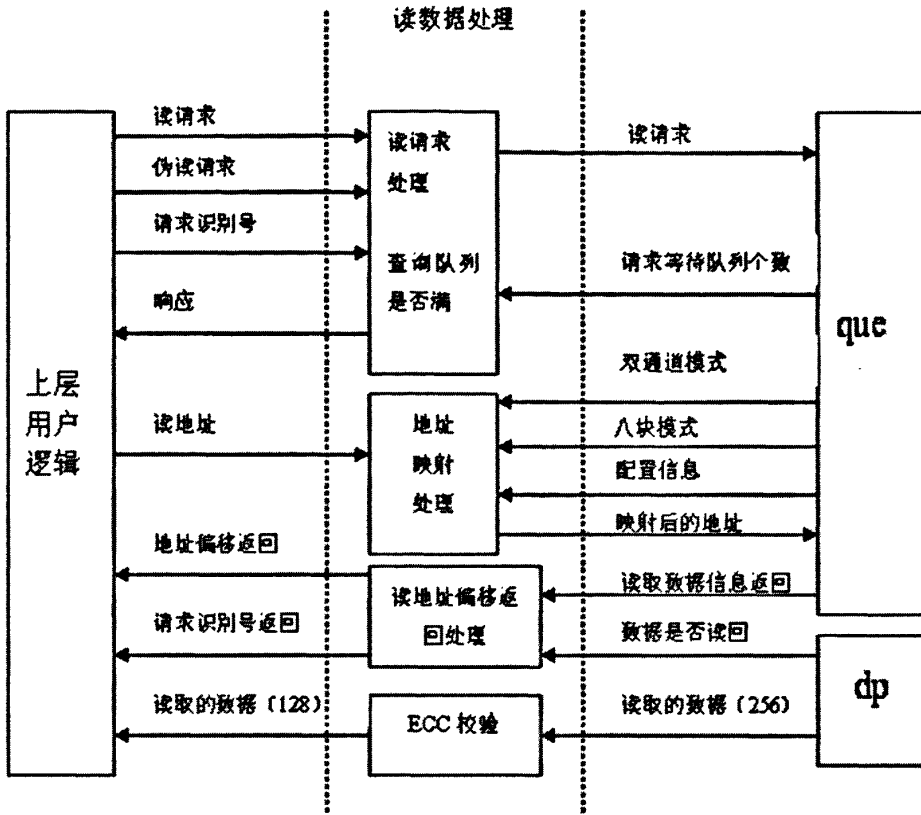


图 4.7 UIB 读请求处理

UIB 模块处理读请求时，关键信号的时序如 4.8 所示。

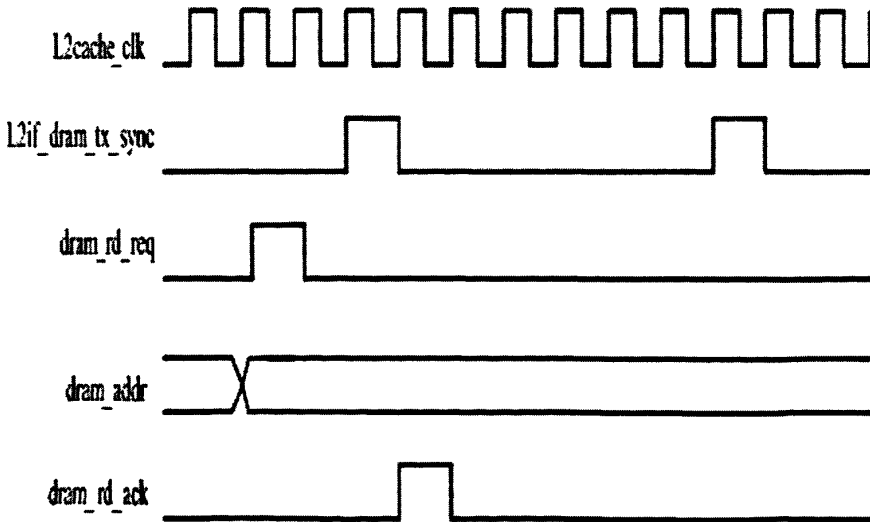


图 4.8 UIB 读请求处理时序图

在上图中各信号的含义如下：

- L2 Cache_clk: 二级 Cache 的工作时钟;
- l2if_DRAM_tx_sync: 时钟域转换的同步信号;
- DRAM_rd_req: 来自上层用户逻辑的读请求;
- DRAM_addr: 来自上层用户逻辑的请求地址;
- DRAM_rd_ack: 返回上层用户逻辑的读请求应答信号。

4.2.3 写请求处理模块

该模块进行写请求处理时所执行的操作包括：写请求接收、写请求应答、写数据寄存并向写数据通路模块发送。如图 4.9 所示。

1、写请求接收：用户接口模块 UIB 接收到上层用户逻辑发送的写请求后，考察请求调度模块中的写等待队列的填充情况。如果写等待队列已经填满则 UIB 模块不向上层用户逻辑发送应答信号，如果写等待队列不满则向上层用户逻辑发送应答信号。在发送应答信号的同时，UIB 模块向请求调度模块发送写请求信号和写请求地址。

2、写请求应答：对上层用户逻辑的请求经过查询判断后作出回应。

3、数据接收：用户接口模块和上层用户逻辑之间的写数据宽度为 64 位。UIB 接收上层用户逻辑来的写数据，经过寄存器寄存后，检查写入数据的 ECC 码是否正确，如果 ECC 正确，则向写数据通路模块发送写数据；如果 ECC 错误，则取消该请求。写数据通路模块包含写数据缓冲器，由双端口存储器构成。UIB 模块按照上层用户逻辑时钟频率向写数据缓冲器中写入数据，后级数据通路在写请求被响应后按照 DRAM 时钟频率从写数据缓冲器中读出写数据。由于写操作猝发 4 个 128 位写数据，因此 UIB 模块响应的写请求时连续向数据缓冲器中写入 8 个 64 位数据。

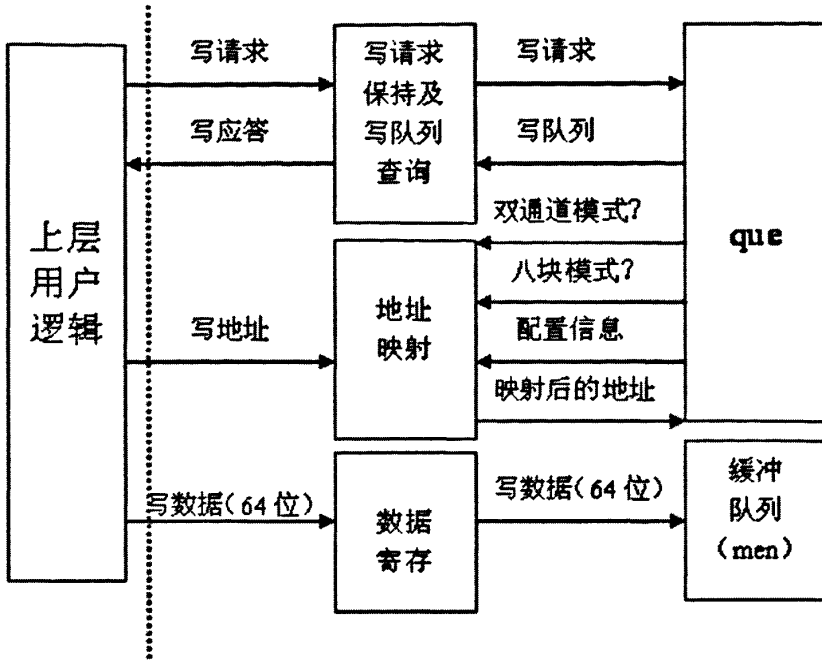


图 4.9 写请求处理处理模块

UIB 模块处理写请求时，关键信号的时序如图 4.10 所示。

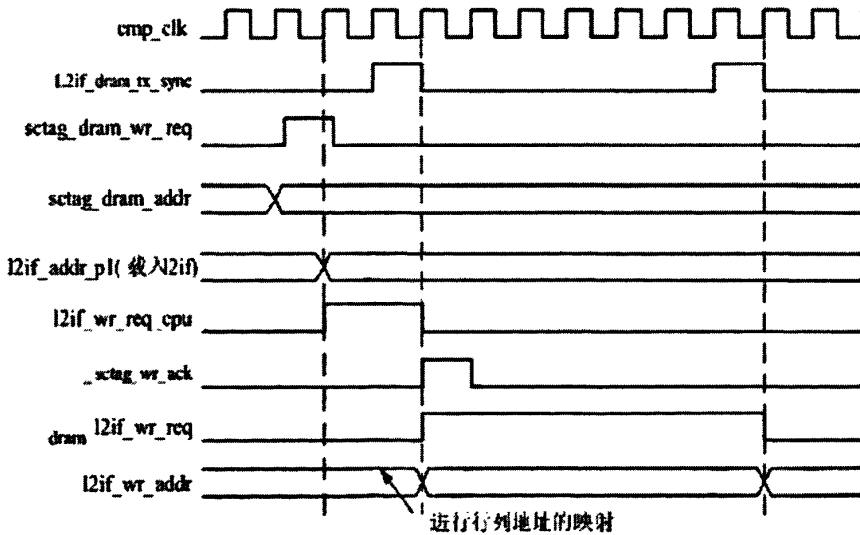


图 4.10 UIB 模块写请求处理时序图

在上图中各信号的含义如下：

DRAM_wr_req: 从上层用户逻辑发出的写请求信号；

DRAM_addr: 从上层用户逻辑发过来的请求地址；

DRAM_wr_ack: 发往上层用户逻辑的写应答信号。

在写数据通路中，上层用户逻辑每次发送写请求并被响应后，将 8 个 64 位的数据写入写缓冲模块 dram_mem 中。当写请求被响应且发送了写操作的 CAS 命令后，dram_dp 模块在 dram 时钟频率下从 dram_mem 模块中连续两拍读出两个 256 位的写数据。

从 dram_mem 读出的 256 位写数据 que_mem_data[255:0] 进入 dram_dp 模块后首先要和读数据通路输入的 scrb 数据 dp_scrb_data0[255:0]、dp_scrb_data1[255:0] 以及从另一通道输入的写数据 ch1_que_mem_data[255:0]（两通道模式下）一起被仲裁选择得到数据 dp_mux_data[255:0]，选择信号来自于 dram_que 模块。如图 4.11 所示。

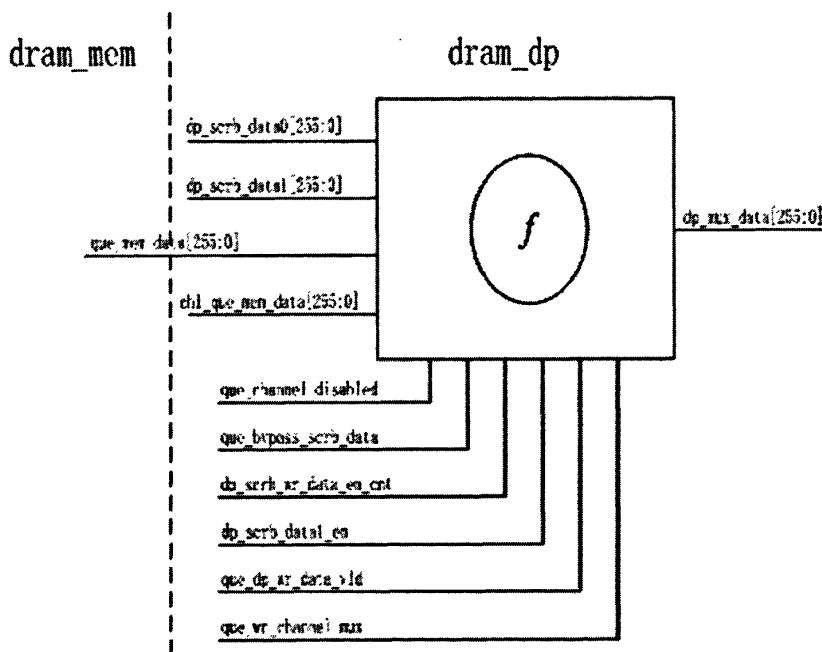


图 4.11 dram_dp 模块中 256 位数据的选择

经过选择得到的 dp_mux_data[255:0] 分成高 128 位和低 128 位分别由 ecc 生成单元产生相应的 ecc 码。新产生的 ecc 码将和读数据通路输入的 scrb 数据 ecc 码一起被仲裁选择得到 ecch[15:0] 和 ecc1[15:0]，选择信号同样来自于 dram_que 模块。如图 4.12 所示。

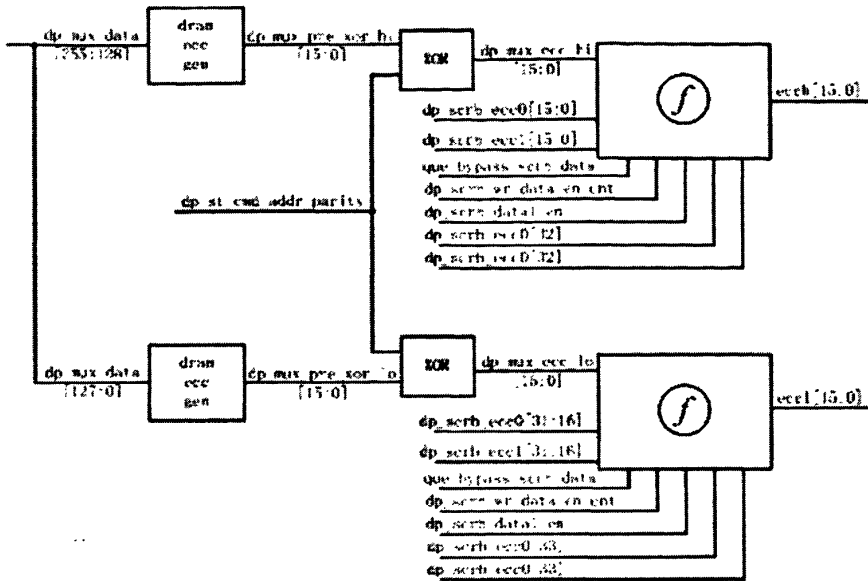


图 4.12 新 ecc 码产生

最后，256 位的写数据 $dp_mux_data[255:0]$ 和新产生的 ecc 码 $eccb[15:0]$ 、 $eccl[15:0]$ 经过错误注入操作后被组装成 288 位的写数据 $dram_io_data_out[287:0]$ 发送到物理层的 pad_ddr 模块。如图 4.13 所示。

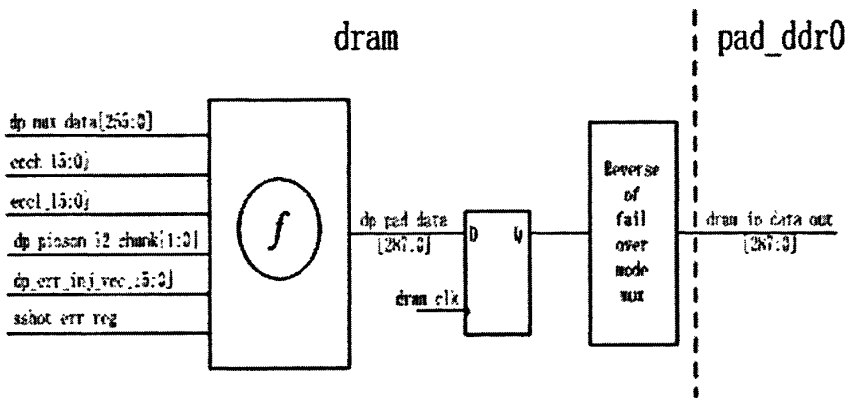


图 4.13 写数据被发送到物理层

4.2.4 ECC 写数据校验模块

ECC(Error Checking and Correcting, 错误检查和纠正)模块的功能是对从 L2cache 中读取数据的校验, 然后发送到写数据通路^[18]。表 4.1 和表 4.2 说明了写数据校验 ECC 的产生方式和 L2 数据 ECC 校验代码。

表 4.1 L2 数据校验位生成

Check [6:0]	Data[31:0]																															
	3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0
C0	0	1	0	1	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1
C1	1	0	0	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	1	1	0	0	1	1	0	1	1	0	1
C2	1	1	1	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	0	1	1	1	0
C3	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0
C4	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
C5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C6	0	1	0	1	1	0	1	1	0	1	0	1	1	0	0	1	0	1	1	1	0	0	0	0	1	1	0	1	1	1	1	

其中，1、0 代表的意思分别是：

1——该数据位参加校验位的异或运算；

0——该数据位不是 ECC 校验位的组成部分。

对于 synd{5:0}、synd{6}来说，这个 ECC 校验是校验位的逆位运算，但是对于数据{31:0}、{6:0}来说，它就是简单的异或运算。

表 4.2 L2 数据 ecc 校验综合代码表

Synd [6:4]	Synd {3:0}															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	ne	u	u	p/u	u	u	u	u	u	u	u	u	u	u	u	u
1	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u
2	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u
3	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u
4	C6	C0	C1	0	C2	1	2	3	C3	4	5	6	7	8	9	10
5	C4	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
6	C5	26	27	28	29	30	31	M	M	M	M	M	M	M	M	M
7	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M

其中，上表中的代码的意思分别是：

Ne——没有错误；

C0-C6——该数据的校验位上有一个位错；

0-31——该数据位上有一个位错；

P/U——该数据丢失、损坏和有两个或两个以上偶数个不可纠正的位错；

U——该数据有两个或两个以上偶数个不可纠正的位错；

M——该数据有三个或三个以上奇数个位错。

4.2.5 读写等待队列模块

在 `dram_que` 模块中, 对 `dram_l2if` 模块输入的读、写请求各有一套 8 入口的等待队列, 用于存放已从 `dram_l2if` 模块接收但还没有被响应的读、写请求。

读、写请求等待队列都采用从低到高的填充顺序, 即序号小的入口比序号大的入口具有更高的填充优先级。当接收到新的读、写请求后, 首先考察相应等待队列各入口的占用状态, 将未被占用的序号最小的等待队列入口使能信号置为高电平, 从而将读、写请求相关信息填充到该入口对应的寄存器中。当入口完成请求填充后, 其对应的占用状态位被置为高电平表示该入口被占用, 直到当其保存的请求被完整执行后, 入口占用状态位才被复位。

每个队列入口对应一组寄存器用于寄存读、写请求相关的信息, 以及入口占用状态位。读、写请求相关的信息包括请求 `id` 和请求地址信息。对于写等待队列入口, 除了入口占用状态位外还有一个由仲裁器访问的占用状态位, 该位用于仲裁器对入口状态的考察, 该位的置位条件与入口占用状态位相同, 但当请求的 `RAS` 命令被发送后该位即可被复位, 而不需等到请求的 `CAS` 命令被发送。

读、写请求等待队列各有一个请求排序模块, 用于对等待队列中的请求信息进行排序。该模块以等待队列入口序号作为各个请求的标识, 为每个请求赋以一个索引号。最早进入等待队列的请求具有最大的索引号, 最晚进入等待队列的请求索引号最小, 在请求仲裁阶段, 对于访问同一存储体地址的多个请求, 索引号最大的请求具有最高优先级。

新进入的读请求可以与等待队列中的请求一起参与仲裁, 但新进入的写请求必须先进入写等待队列, 不能直接参与仲裁器的仲裁。

4.3 请求调度模块 ARB

请求调度模块完成各种存储器请求的仲裁和调度。存储控制器中需要进行存储器访问的请求有: 上层用户逻辑发来的存储器读写请求、纠错回写模块发出的存储器读写请求、存储器的定时刷新请求等。请求调度模块的核心就是各种请求的仲裁, 仲裁器按照一定的优先级仲裁算法将这些访存请求发送给物理层。如何选择仲裁算法是本小节研究的主要内容。

为了合理高效地控制和管理系统中命令的传输, 必须使用一个特别的优先级仲裁算法, 以便在多个访存请求同时提出仲裁请求时, 能够依据某仲裁算法判决出哪个请求可优先访问存储器。仲裁算法必须确保任何时刻只有一条命令发送给物理层。DDR3 存储控制器的仲裁器是基于访问的, 也就是说, 一个请求模块必须

为它的每次访问提出仲裁请求。仲裁器采用并行仲裁(也叫独立请求仲裁)方案,每个请求模块都有自己独立的信号线,相互间没有任何控制关系。仲裁器直接识别所有模块的请求,并根据优先级仲裁选中一个模块,向它发出应答信号。仲裁算法必须保证所有的模块都能得到授权的机会,否则将会出现某个优先级低的模块永远不能进行访存操作的情况。

4.3.1 仲裁算法

为了合理的控制和管理系统中命令的传输,必须使用一个特别的优先级仲裁算法,以便在多个设备同时提出仲裁请求时,能够依据该仲裁算法判决出哪个设备获得使用权^[19]。通常使用的仲裁算法有两种:一种是固定优先级算法,一种是循环优先级算法。所谓固定优先级算法,就是指仲裁器的各个请求设备的优先级是事先确定好的,仲裁过程中固定不变;而循环优先级算法则不同,各个请求模块的优先级在仲裁过程中不是一成不变,而是根据一定规律发生变化的。

采用固定优先级算法,可以对那些有重要数据传输、或有大量实时数据传输以及经常有仲裁请求的模块赋予较高的优先权,以提高DDR3 存储器的数据传输率,从而有效提高处理器带宽。固定优先级算法的缺点是,它将规定模块的优先级,这样可能会出现“撑死”和“饿死”的现象。相反,采用循环优先级算法则可以克服这种“饥饿”不均的弊端。在循环优先级算法中,由于其优先级周期动态地改变,各个模块获得优先权的机会均等。所以,在一定意义上来说,循环优先级算法是最公平的算法。循环优先级的缺点是当处理某些模块传输大批量实时数据时会造成效率的降低。正因如此,在实际中常常将这两种算法结合起来使用,以便构成更为灵活的仲裁机制。

DDR3 SDRAM的初始化过程必须在正常读写操作之前完成,否则可能出现不可预测的错误,所以控制管理部件的初始化命令必须得到最高的优先级。由于各个请求模块处于平等的关系,为了将访问的机会平均分配,按照循环优先级算法判决使用权的归属。因此,DDR3存储控制器的仲裁器优先选择控制管理部件的初始化请求,其余按照循环优先级算法选择。

4.3.2 仲裁器实现

仲裁器接收外部模块的请求后,按照规定的仲裁算法找出优先级最高的请求,将其发送给物理层。仲裁器内部设置了一个命令缓冲寄存器,命令发出后同时将其寄存在寄存器中,等待物理层的应答信号,然后将寄存器置为无效。仅当寄存器无效时仲裁器才可以接收下一个请求,因此仲裁器应答外部请求的最快速度是

两个时钟周期一个请求。由于物理层服务一个请求的时间不会少于两个时钟周期，因此并不会影响工作性能。

请求处理流程：

1、请求排队

在请求调度模块中，设有读请求队列和写请求队列，访问请求首先根据其访问类型（读或写）在相应的队列中进行排队。在每一个队列中，仲裁器根据访存请求的存储体地址按数据带宽最大化的原则进行调度。

2、请求分类

仲裁器根据访存请求的不同阶段分成两种类型：RAS请求和CAS请求。

RAS请求：从没有被调度过的请求，包括从请求队列中来的请求、从纠错回写模块来的请求和刚刚进入请求调度模块还未进入请求队列的请求。

CAS请求：已经被仲裁器调度向存储器发送行地址，但尚未发送列地址的请求。

3、请求仲裁

对于各种访存请求，仲裁器采用了三个仲裁级别进行仲裁响应：

- 首先，是请求来源通道的仲裁，区分请求是来自本通道内部还是从另一通道输入；

- 其次，是访存请求类型仲裁，根据不同访存请求的类型进行优先仲裁；

- 最后，是先进先出仲裁，对于同样类型的访问请求响应等待时间最长那个。

1、请求来源仲裁

在二通道工作模式下，仲裁器除了仲裁本通道的读、写请求外还要考虑从另一通道传来的请求，不同通道之间的请求响应优先级原则如下：

- 如果申请访问该 Bank 的所有请求都来自本通道，则仲裁器按照常规仲裁原则仲裁，并将信号线 `que_channel_picked` 置“0”表示此请求来自本通道；

- 如果申请访问该 Bank 的所有请求都来自另一通道，则仲裁器仍按照常规仲裁原则仲裁，并将信号线 `que_channel_picked` 置“1”表示此请求来自另一通道；

- 如果两通道都有请求申请访问该 Bank，则仲裁器按照两通道轮流响应的优先级进行仲裁。同样，信号线 `que_channel_picked` 会被相应置位，表示响应的请求是来自本通道还是另一通道。

2、请求类型仲裁

在 DDR3 控制器中，对于不同类型的访存请求，仲裁的常规仲裁优先级如下：

- CAS 请求：最高优先级，保证请求尽快完成；

- 纠错回写 RAS 请求；

- 与读队列请求有地址相关性的写操作 RAS 请求；

- 读队列中的读 RAS 请求，或者当饥饿计数器（starvation counter）达到

极限时写队列中的写 RAS 请求；

●写队列中的写 RAS 请求，或者当饥饿计数器 (starvation counter) 达到极限时读队列中读 RAS 请求；

●新进入的读 RAS 请求。

3、先进先出仲裁

对于多个相同类型的访存请求，仲裁器采用先进先出的仲裁优先级响应策略。仲裁器判断请求先后次序的依据是读、写等待队列的排序模块对各个请求赋值的索引号，索引号越大表示该请求等待时间越长。如图 4.14 说明了仲裁器的仲裁级别。

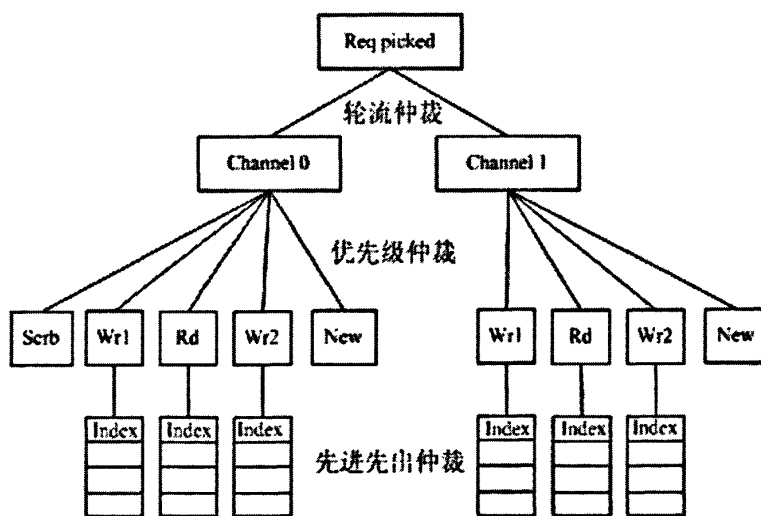


图 4.14 仲裁级别

除以上仲裁策略外，仲裁器中还包含多个计数器，用于保证 DDR3 控制器的有效工作，所使用的计数器主要有：

●que_wr_starve_cnt[5:0]写请求饥饿计数器：由于读请求比写请求具有更高的优先级，所以为了避免写请求被“饿死”，在模块中设置了 6bit 的写请求饥饿计数器。当写请求队列填满时，饥饿计数器开始从 0 逐拍加 1，计数溢出时则将信号线 que_pick_wr_first 设置为 1，此时仲裁器将优先响应写请求。同时写请求饥饿计数器开始每拍自减 1，当饥饿计数器计数值被减到 0 后信号 que_pick_wr_first 被复位为 0，仲裁器重新优先响应读请求。

●rrd_cnt[3:0]不同 Bank 间 RAS 命令到 RAS 的延迟计数器：当发送一个 RAS 命令后开始将控制寄存器 rrd_reg[3:0]中的存储值逐拍减 1，当计数值减到 0 后保持为 0 且将信号线 rrd_cnt_is_zero 设置为 1。

●rtw_cnt[3:0]读到写延迟计数器：满足读到写延迟协议，当发送一个读请求的 CAS 命令后，计数器开始将控制寄存器 rtw_reg[3:0]中的存储值逐拍减 1，当计数值减到 0 后保持为 0 且将信号线 rtw_cnt_is_zero 设置为 1。

●wtr_cnt[3:0]写到读延迟计数器：满足写到读延迟协议，当发送一个写请求的 CAS 命令后，计数器开始将控制寄存器 wtr_reg[3:0]中的存储值逐拍减 1，当计数值减到 0 后保持为 0 且将信号线 wtr_cnt_is_zero 设置为 1。

●rtr_cnt 读到读延迟计数器：满足读到读延迟协议，当发送一个读请求的 CAS 命令后，计数器开始从 1 逐拍减 1，当计数值减到 0 后保持为 0 且将信号线 rtr_cnt_is_zero 设置为 1。

●wtw_cnt 写到写延迟计数器：满足写到写延迟协议，当发送一个写请求的 CAS 命令后，计数器开始从 1 逐拍减 1，当计数值减到 0 后保持为 0 且将信号线 wtw_cnt_is_zero 设置为 1。

其中信号线 que_pick_wr_first 和 rrd_cnt_is_zero 用于控制 RAS 请求的响应仲裁，而信号线 rtw_cnt_is_zero、wtr_cnt_is_zero、rtr_cnt_is_zero 和 wtw_cnt_is_zero 用于控制 CAS 请求的响应仲裁。

仲裁器根据仲裁策略选择了响应的请求后，从相应的寄存器中取出请求地址等相关信息，并根据 DDR3 读、写命令生成正确的 RAS 命令和地址并输出到后级的物理层单元。

4.4 地址通路模块 (APB)

地址通路模块完成从用户地址到存储器行列地址的转换，并根据不同的映射模式对地址进行不同的变换。为了提高不同应用条件下的访存效率，DDR3 控制器采用 2 种映射模式，分别是 dram_addr_gen_hi 和 dram_addr_gen_lo^[20]。如图 4.15 所示。两个模块分别按高位地址和低位地址对物理地址进行映射，映射后的两种结果由信号 l2if_addr_bank_low_sel 进行选择使用。信号 l2if_addr_bank_low_sel 由 ucb 接口赋值，表示一种存储器组织形式。

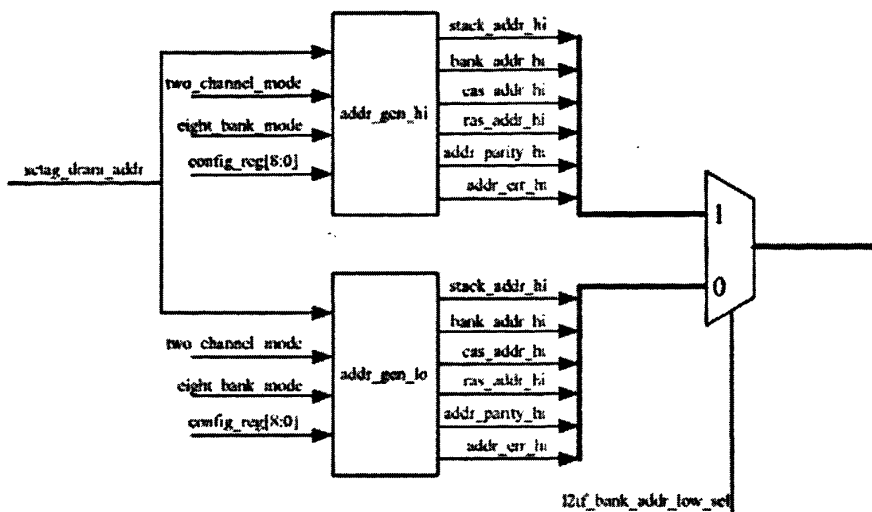


图 4.15 物理地址映射单元

dram_addr_gen_hi 和 dram_addr_gen_lo 分别根据表示存储器组织形式的配置信号对请求的物理地址进行映射。所使用的配置信号有：

- two_channel_mode: 表示 DRAM 控制器是处于双通道模式还是单通道模式；
- eight_bank_mode: 表示存储器中是否具有 8 个 bank；
- config_reg[8:0]: 表示存储器组织形式；
- rank1-present: 表示 DIMM 条是否是双面。

4.5 纠错回写控制单元 SCRB

DDR3 DRAM 控制器支持存储器数据纠错回写 (scrb) 操作, scrb 读、写请求及请求地址等信息都由 scrb 控制单元产生。

Scrb 使能信号 que_scrb_enable 由上层用户逻辑通过 DRAM 控制器的寄存器接口进行置位。当使能信号被置位后, 控制单元内部的 scrb 频率计数器 que_scrb_cnt[11:0] 开始计数。其计数溢出值可由 CPU 通过寄存器接口对配置寄存器 chip_config_reg 的第 [20:9] 位进行设置。当计数器 que_scrb_cnt[11:0] 的计数值大于 chip_config_reg[20:9] 中的设置值后, 信号 que_scrb_time 被置为高电平, 并开始发送 scrb 读请求。因此 chip_config_reg[20:9] 中的设置值表示 scrb 操作间隔的时钟周期数。

当 que_scrb_time 被置为高电平, 且此时没有 scrb 写操作时, 控制器将发送一个 scrb 读请求信号, 该信号将与读、写等待队列中的读、写请求以及新进入的读请求一起参与仲裁器仲裁。而 scrb 请求具有第二高优先级, 因此, scrb 读请求

信号将被先响应。Scrb 读请求信号发出的同时，控制器将计数器 `que_scrb_cnt[11:0]` 的内容清 0，若 `scrub` 使能信号仍然为高则计数器继续从 0 开始计数直到下一次 `scrub` 操作。

当 `scrub` 读请求被响应后，`scrub` 写使能被置位，准备发送写请求信号，并在下一拍将 `que_scrb_write_valid` 信号置 1 表示当前进入 `scrub` 写操作。当写请求被响应后，`que_scrb_write_valid` 被复位。

Scrb 写 pending 信号 `que_prev_scrb_wr_pending`：当进入 `scrub` 写操作后信号被置 1，表示写操作正执行。当 `scrub` 写请求的 CAS 命令被发送后，该信号被复位。该信号用于 `scrub` 读请求时考察当前是否有正在执行的写操作。

Scrb 地址生成：`scrub` 的请求地址由控制器自动生成。

4.6 写数据传输通路

如图 4.16 是一个 DDR3 DRAM 通道的数据通路结构。DDR3 DRAM 通道的数据通路可分为读数据通路和写数据通路。对于写数据通路，64 位的上层用户逻辑写数据首先在用户逻辑时钟频率下被逐拍写入数据缓冲 `dram_mem` 模块中。`dram_dp` 模块在 `dram` 时钟频率下从 `dram_mem` 模块中读出 256 位的写数据，并生成 32 位的 `ecc` 校验码。288 位的写数据与 `ecc` 校验码一起被发送到 DDR3 DRAM 控制器的物理层 `pad_ddr` 模块，并由 `pad_ddr` 模块拆分成两个 144 位的写数据发送到 DDR3 DRAM 存储器中。

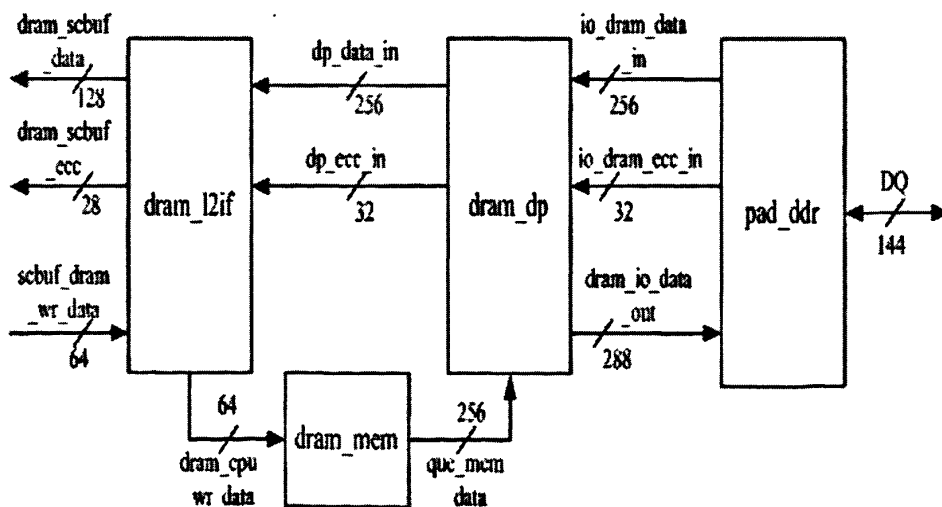


图 4.16 数据通路

物理层控制单元发送的命令如果是写命令，就要通知写数据通路传输写数据。

写数据通路主要完成写数据的发送和写通路的控制，写数据的发送由写数据传输通路完成，写通路的控制由写计数器完成。

物理层控制单元发送的命令如果是写命令，就要通知写数据通路传输写数据。写数据通路主要完成写数据的发送和写通路的控制，写数据的发送由写数据传输通路完成，写通路的控制由写计数器完成。写数据通路的结构如图4.17所示。

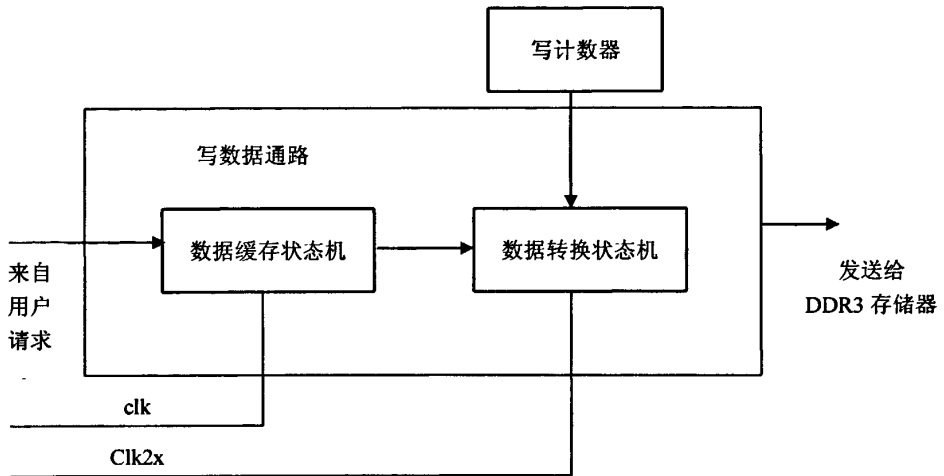


图4.17 写数据通路结构图

DDR3存储器的写数据来自用户层，读数据发送给用户层。用户层传输的128位数据为单沿触发，DDR3存储器传输的64位数据为双沿传输。写数据通路要将单沿触发的数据转换为双沿触发，并发送给DDR3存储器。由于写数据和数据选通信号DQS是中心对齐的，所以如何能产生正确的DQS信号也是写数据通路设计的问题之一。

写数据通路中的ECC是对被发送到DDR3 DRAM控制器的物理层的数据进行校验，其具体算法如下：

Check nibble0

$$(4 \text{ bits}) = (N_0 + 2*N_1 + 3*N_2 + 4*N_3 + 5*N_4 + 6*N_5 + 7*N_6 + 8*N_7 + 9*N_8 + A*N_9 + B*N_{10} + C*N_{11} + D*N_{12} + E*N_{13} + F*N_{14} + N_{15} + 2*N_{16} + 3*N_{17} + 4*N_{18} + 5*N_{19} + 6*N_{20} + 7*N_{21} + 8*N_{22} + 9*N_{23} + A*N_{24} + B*N_{25} + C*N_{26} + D*N_{27} + E*N_{28} + F*N_{29} + N_{31}) \wedge \{4\{\text{addr_parity}\}\}$$

Check nibble1

$$(4 \text{ bits}) = (N_0 + N_1 + N_2 + N_3 + N_4 + N_5 + N_6 + N_7 + N_8 + N_9 + N_{10} + N_{11} + N_{12} + N_{13} + N_{14} + N_{30} + N_{31}) \wedge \{4\{\text{addr_parity}\}\}$$

Check nibble2

$$(4 \text{ bits}) = (N_{15}+N_{16}+N_{17}+N_{18}+N_{19}+N_{20}+N_{21}+N_{22}+N_{23}+N_{24}+N_{25}+N_{26}+N_{27}+N_{28}+N_{29}+N_{30}+N_{31}) \wedge \{4\{\text{addr_parity}\}\}$$

Check nibble3

$$(4\text{bits}) = (N_0+9*N_1+E*N_2+D*N_3+B*N_4+7*N_5+6*N_6+F*N_7+2*N_8+C*N_9+5*N_{10}+A*N_{11}+4*N_{12}+3*N_{13}+8*N_{14}+N_{15}+9*N_{16}+E*N_{17}+D*N_{18}+B*N_{19}+7*N_{20}+6*N_{21}+F*N_{22}+2*N_{23}+C*N_{24}+5*N_{25}+A*N_{26}+4*N_{27}+3*N_{28}+8*N_{29}+N_{30}) \wedge \{4\{\text{addr_parity}\}\}$$

Syndrome nibble0

$$(4 \text{ bits}) = (C_0+N_0+2*N_1+3*N_2+4*N_3+5*N_4+6*N_5+7*N_6+8*N_7+9*N_8+A*N_9+B*N_{10}+C*N_{11}+D*N_{12}+E*N_{13}+F*N_{14}+N_{15}+2*N_{16}+3*N_{17}+4*N_{18}+5*N_{19}+6*N_{20}+7*N_{21}+8*N_{22}+9*N_{23}+A*N_{24}+B*N_{25}+C*N_{26}+D*N_{27}+E*N_{28}+F*N_{29}+N_{31}) \wedge \{4\{\text{addr_parity}\}\}$$

Syndrome nibble1

$$(4 \text{ bits}) = (C_1+N_0+N_1+N_2+N_3+N_4+N_5+N_6+N_7+N_8+N_9+N_{10}+N_{11}+N_{12}+N_{13}+N_{14}+N_{30}+N_{31}) \wedge \{4\{\text{addr_parity}\}\}$$

Syndrome nibble2

$$(4 \text{ bits}) = (C_2+N_{15}+N_{16}+N_{17}+N_{18}+N_{19}+N_{20}+N_{21}+N_{22}+N_{23}+N_{24}+N_{25}+N_{26}+N_{27}+N_{28}+N_{29}+N_{30}+N_{31}) \wedge \{4\{\text{addr_parity}\}\}$$

Syndrome nibble3

$$(4 \text{ bits}) = (C_3+N_0+9*N_1+E*N_2+D*N_3+B*N_4+7*N_5+6*N_6+F*N_7+2*N_8+C*N_9+5*N_{10}+A*N_{11}+4*N_{12}+3*N_{13}+8*N_{14}+N_{15}+9*N_{16}+E*N_{17}+D*N_{18}+B*N_{19}+7*N_{20}+6*N_{21}+F*N_{22}+2*N_{23}+C*N_{24}+5*N_{25}+A*N_{26}+4*N_{27}+3*N_{28}+8*N_{29}+N_{30}) \wedge \{4\{\text{addr_parity}\}\}$$

ECC 纠错算法过如下: If s0, s1, s2 and s3 are the 4 syndrome nibble,

Position 0→14(nibble position)

```
If (s2==0 && s1 != 0 && s0 !=0), then{
Nibble_to_correct = (s0/s1 - 1);
Corrected_data = s1 + erred_nibble;
}

Position 15->29(nibble position)
If (s1==0 && s0 !=0 && s2 !=0), then {
Nibble_to_correct = (so/s2+14) ;
Corrected_data = s2 + erred_nibble;
}
Position 30(nibble position)
If (s0 == 0 && s1 !=0 && s2 != 0 && s1 == s2), then {
Nibble_to_correct + N30
Corrected_data = s1+ erred_nibble;
}
Position 31(nibble position)
```

4.6.1 读写控制

读写数据通路中对读写控制的实现基本相同，在此进行统一介绍。

我们将数据传输的过程分为前奏、传输和收尾^[21]，前奏用来通知读写数据通路准备数据传输，收尾表示数据已经传输完毕。读写控制主要控制数据传输的开始(前奏阶段)，用计数器实现，计数器的值表示还需多少拍发送或接收数据。读写数据通路接收物理层控制单元给出的读写开始信号，将读计数器或写计数器置位，此后每拍减1。写计数器为1时，启动DQS信号形成写前奏；当读写计数器为0时，开始传输读写数据，同时分别控制DQS信号与DQ信号中心对拟写数据或根据DQS信号获取读数据(读数据时)。读写数据结束后就可以根据读写开始信号启动下一轮的数据传输。读写流程如图4.18所示。

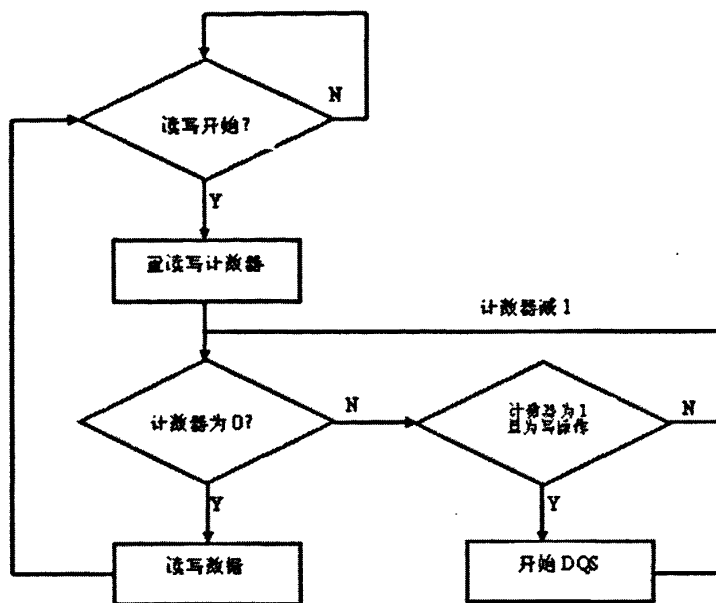


图 4.18 物理层读写控制流程图

读写计数器的初值分别来自传输层发送的读操作到读数据的间隔RdCounter和写操作到写数据的间隔WrCounter，这两个值是可配置的，保存在传输层的配置寄存器中。由于此时间间隔的存在，DDR3存储器控制器发出读写命令和传输数据之间需要等待。为了减少因等待引起的资源浪费，数据通路中允许两个命令同时存在。如果用一个计数器，那么每两个读写操作必然串行，所以对读写分别用了2个计数器，它们可以同时计数，一旦某个计数器为0，就开始传输数据，由于在物理层控制单元中已经控制了读写命令的间隔，因此不可能会有冲突的情况出现。在流程图中，可以有不同的命令存在于图中不同的流程中。

写数据通路中，提供一个双倍系统时钟clk2x控制DQS。计数器为1时，在系统时钟clk2x的下一上升沿将DQS降为0，形成1/2(clk2x)的写前奏阶段；随后的clk2x下降沿传输数据，在clk2x上升沿改变DQS，直到数据传输结束，这样就完成了数据的双沿触发和DQS与DQ的中心对齐；随后的clk2x上升沿变化DQS信号，之前1/2(clk2x)的DQS为低电平的阶段称为写收尾。在实际工程中，利用DQS使能信号dqs_en指示DQS信号的开始和结束，数据传输前的1/2(clk2x)和数据传输后的1/2(clk2x)分别使能和关闭dqs_en，以此形成写前奏和写收尾。

读数据通路中，计数器为0时，开始接收读数据。

与数据同时传输的还有ECC验证码和数据屏蔽信号。ECC验证码作为数据的一部分进行传输，数据屏蔽信号用于指示数据中哪些位有效。数据屏蔽信号是从物理层控制单元传来，在命令发给DDR3存储器时，需要将其暂存在数据通路中。写

数据时，与写数据同时发出；读数据时，在读数据到达后，根据数据屏蔽信号向用户层发出数据有效信号^[22]。

4.6.2 写数据传输通路

写数据传输通路主要控制写数据的传输和写收尾^[23]。接收物理层控制单元发送的写开始信号时，写数据通路准备传输写数据，等待一定的时间间隔WrCounter后，开始传输数据，与数据同时传输的还有数据选通信号DQS、数据屏蔽信号DM和ECC验证码，这些信号都是双沿触发信号，并且DQS信号与DQ信号是边沿对齐的。由此可见，写数据通路模块中需要解决的问题主要有以下两个：第一，用户层单沿触发的数据如何变换为DDR3存储器需要的双沿触发数据；其次，怎样产生与数据信号中心对齐的数据选通信号。

写数据的单沿触发转换为双沿触发的过程用两个状态机实现：数据缓存状态机和数据转换状态机。数据缓存状态机主要实现用户层单沿数据的缓存，在系统时钟吸的上升沿接收并缓存数据；然后使用数据转换状态机将数据分别在clk2x的下降沿分两次发出，完成数据从单沿触发到双沿触发的转换。数据缓存状态机工作于系统时钟clk域，数据转换状态机工作于系统时钟clk2x域，门级电路如图4.19所示。

在数据有效窗口中，DQS必须对齐DQ的中心位置以便DDR3存储器可以正确获取数据。写计数器值为0时，写数据传输通路接受用户层的写数据，利用系统时钟clk和clk2x将用户层单沿触发的数据转换为双沿触发后发送给DDR3存储器，同时在clk2x的上升沿产生数据选通信号，实现了DQS和DQ的中心对齐。写数据的时序如图4.20所示。

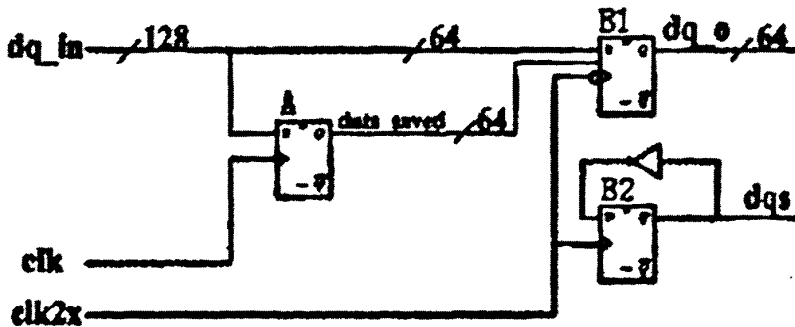


图 4.19 写数据传输通路结构图

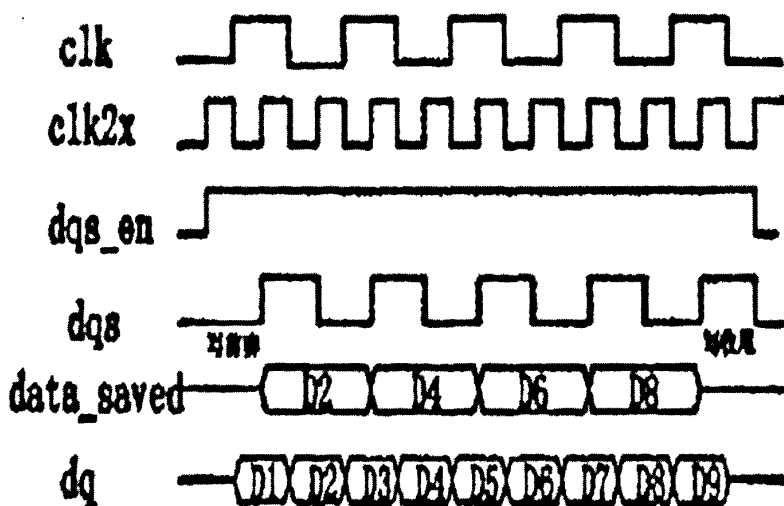


图 4.20 写结构时序图

dram_mem 模块是 L2-cache 写数据的数据缓冲 buffer，在代码中由四个 16x65bit 的寄存器宏单元 bw_rf_16x65 构成，其中每个寄存器都只使用低 64 位 [63:0]，最高位不被使用。dram_mem 模块的结构如图 4.21 所示，

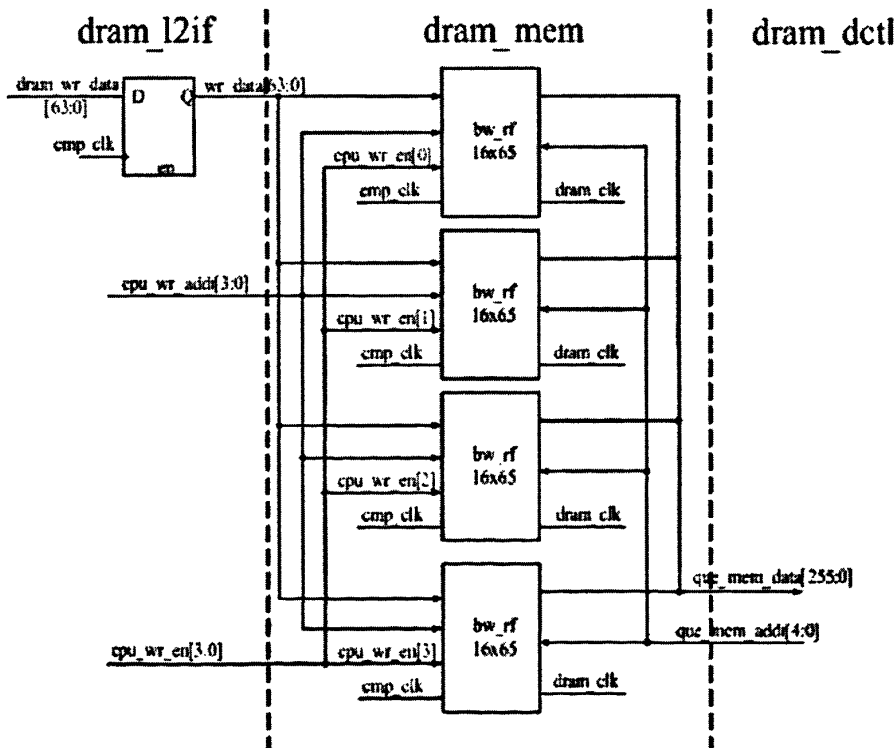


图 4.21 dram_mem 结构

宏单元 `bw_rf_16x65` 是双端口寄存器文件，读、写端口具有各自独立的地址、使能和时钟输入端。读、写端口可以工作在不同时钟频率下。

在 `dram_mem` 模块中，寄存器的写端口输入信号来自 `dram_l2if` 模块。其中四个寄存器的写数据端口都连接在数据输入端 `wr_data[63:0]`；四个寄存器的写地址端口都连接在地址输入端 `cpu_wr_addr[3:0]`；四个寄存器的写使能端分别连接写使能输入 `cpu_wr_en[3:0]` 的四个使能信号；写时钟端口连接时钟 `cmp_clk`。当 DRAM 控制器接收到上层用户逻辑的写请求后，`dram_l2if` 模块产生写地址 `cpu_wr_addr[3:0]`，并在 `cpu` 时钟频率下轮流将写使能信号 `cpu_wr_en[3:0]` 的四个使能端置位为“0”（使能端低电平有效），这样经过 4 个 `cpu` 时钟，`dram_l2if` 模块分别将 4 个 64bit 的数据写入每个寄存器的相同地址。此后，`dram_l2if` 模块将写地址 `cpu_wr_addr[3:0]` 加 1，并再次轮流将写使能信号 `cpu_wr_en[3:0]` 的四个使能端置位为“0”，这样，再经过 4 个 `cpu` 时钟，`dram_l2if` 模块分别将后 4 个 64bit 的数据写入每个寄存器的下一地址。每次写请求都连续写 8 个 64bit 的数据是因为 DRAM 控制器对内存的写操作数据位宽为 128bit，而突发长度为 4。

寄存器的读端口与 `dram_que` 模块和 `dram_dp` 模块进行交互，四个寄存器的读数据端口连接到数据读出端 `que_mem_data[255:0]` 并进入数据通路模块 `dram_dp`；四个寄存器的读地址端口连接到读地址输入端 `que_mem_addr[4:0]`，这里，地址输入端的低四位 `que_mem_addr[3:0]` 连接四个寄存器的读地址端，而最高位 `que_mem_addr[4]` 连接到四个寄存器的读使能端；读时钟端口连接 `dram` 时钟 `dram_clk`。当 DRAM 控制器发送了写请求的 CAS 命令后等待若干时钟周期（满足 DDR2 内存时序要求），由 `dram_que` 模块产生读数据地址 `que_mem_addr[4:0]` 其中最高位 `que_mem_addr[4]` 为“0”，使寄存器读端口有效。由于四个寄存器的读使能端连接相同的使能信号，因此四个寄存器同时读出 `que_mem_addr[3:0]` 指定地址的数据，第二拍时 `dram_que` 模块将地址 `que_mem_addr[3:0]` 加 1，同时读出四个寄存器中下一个地址的数据。读数操作在 `dram` 时钟 `dram_clk` 下进行。

表 4.3 dram_mem 模块的输入输出信号

信号名称	I/O 状态	信号描述
que_mem_data [255:0]	O	输入到 dram_dp 模块的写缓冲队列读出数据
listen_out [64:0]	O	悬空信号
que_mem_addr [4:0]	I	来自 dram_que 模块的写缓冲队列读地址和使能信号
dram_cpu_wr_addr [3:0]	I	来自 dram_l2if 模块的写缓冲队列写地址
dram_cpu_wr_data [63:0]	I	来自 dram_l2if 模块的写缓冲队列写数据
dram_clk	I	dram 时钟
clk	I	cpu 时钟
dram_cpu_wr_en[3:0]	I	来自 dram_l2if 模块的写缓冲队列写使能信号
margin	I	保留
sehold	I	保留
mem_bypass	I	保留

4.7 读数据通路

对于读数据通路，从DDR3 DRAM存储器返回的144位读取数据首先物理层的pad_ddr模块采样并组装成288位的读取数据（包括256位的读取数据和32位校验码）传回DDR3 DRAM控制器的传输层。在传输层中读取数据首先被转换到cpu时钟频率并经过ECC校验操作，然后，在dram_l2if模块中256位数据被拆分成两个128位的数据，与新生成的28位ECC码一起被发送到上层用户逻辑。

读数据通路用来捕获来自存储器的读数据并且将其传输到内部的时钟领域。

同步的步骤包括：

第一步，DQ被每个DQ I/O的DDR flop所捕获。每个DQ IDDR的输入都是DQ IDDR的延迟版本，通过内建的IDELAY元件来实现。DQ IDELAY 被调节到能够在向IDRR输入的延迟DQ和DQS之间提供足够的时钟时钟。每个DQ所需要的IDELAY设置由一个时钟标度程序所决定，该程序每次系统重启后执行一次。

第二步, IDDR的输出(上升和下降数据)被发送到内部寄存器中, 和每个DQ I/O 都很近。该结构由内核时钟驱动。通过使用用来调整与内核时钟有关的IDDR输出的DQ和DQS IDELAY部件来进行同步。IDELAY的设置同样在初始时钟标准程序执行期间所确定。现在, 寄存器的输出和余下的DDR3接口逻辑所使用的时钟是同步的。

读数据通路接收物理层控制单元发出的读开始信号, 开始计时, 满足读操作到读数据的间隔Rdcounter后, 接收DDR3存储器传来的读数据DQ和数据选通信号DQS。DDR3存储器的读数据为双沿传输, DQS与DQ是边沿对齐的。由此可见, 读数据通路主要完成读通路的控制和读数据的获取。读通路的控制由读计数器完成, 读数据的获取由读数据传输通路完成。

DDR3 DRAM 控制器在向存储器的某一 bank 发送 CAS 命令后, 模块 dram_que 中对应该 bank 的数据返回计数器 b*_data_rtn_cnt[3:0]开始对 DRAM 时钟计数, 当计数值达到模式寄存器 mode_reg[6:4]中保存的 CAS Latency (CL) 后, 将会产生一个时钟周期宽度的脉冲信号 dram_io_pad_enable 该信号被发送到物理层的 pad 模块中, 启动 pad 模块中的读数据采样电路, 开始采样从存储器返回的读取数据。

为了实现读入数据的正确采样, DDR3 DRAM 存储器在向芯片发送数据 DQ 的同时也发送数据同步采样信号 DQS。DDR3 DRAM 存储器为每 4 位 DQ 数据分配一个 DQS 采样信号, 因此 144 位的读取数据一共有 36 位 DQS。每一个 DQS 信号都是与 DDR3 DRAM 时钟同频的脉冲, 在读取数据输入芯片时, pad 模块首先以 DQS 信号为寄存器的触发信号将读取数据采样到输入寄存器。由于从 DDR3 DRAM 存储器输入的 DQ 信号和 DQS 信号具有相同的相位, 因此在 pad 模块中首先需要将输入的 DQS 信号向后移相才能实现 DQ 信号的采样功能。为了保证 DQ 数据能够准确采样, 通常需要将 DQS 信号移相 90°, 保证 DQS 信号的边沿(上升沿和下降沿)对准 DQ 数据窗口的中心。

读数据通路接收物理层控制单元发出的读开始信号, 开始计时, 满足读操作到读数据的间隔RDcounter后, 接收DDR3存储器传来的读数据DQ和数据选通信号DQS。DDR3存储器的读数据为双沿传输, DQS与DQ是边沿对齐的。由此可见, 读数据通路主要完成读通路的控制和读数据的获取。读通路的控制由读计数器完成, 读数据的获取由读数据传输通路完成。读数据通路的结构如图4.22所示。

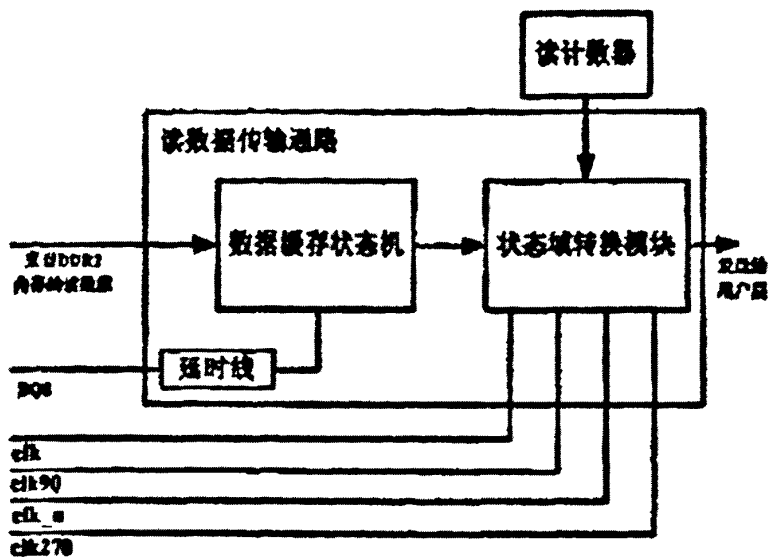


图 4.22 读数据通路结构图

将读数据传输的过程分为读前奏、读传输和读收尾，读前奏用来通知读数据传输通路准备获取数据，读收尾表示数据已经传输完毕。来自DDR3存储器的数据选通信号DQS和数据DQ的时序关系如图4.23所示。

读数据传输通路首先要正确获取读数据，然后根据数据屏蔽信号有选择地对获取的读数据进行处理，发送给用户层。从读时序图中可以看出，DQS和DQ是边沿对齐的，而这两个信号又与系统时钟异步，这就给数据的获取增加了难度。为了能够正确的获取读数据，需要将DQS移相 90° ，使其边沿对齐数据有效窗口的中心位置，并获得正确的建立时间和保持时间，然后在DQS的上升沿和下降沿缓存读数据，最后将数据从DQS域转换到系统时钟域，完成数据的获取。所以读数据传输通路中需要解决的几个问题如下：

- DQS延时：在数据有效窗口中，DQS必须重新对齐DQ中心位置(移相 90°)，以便正确获取数据。
- 数据采样：在读数据期间，必须对DDR3控制器的输入数据进行采样。
- 时钟域转换：数据采样后，它必须和一个异步时钟(控制器内部工作时钟)同步。在从DQS域到内部时钟域的转换过程中，必须考虑DQS选通信号和控制器系统时钟间的相对相位，以保证正确的建立和保持时间。同时，必须使组内数据的时钟倾斜和全部数据的时钟倾斜满足要求。

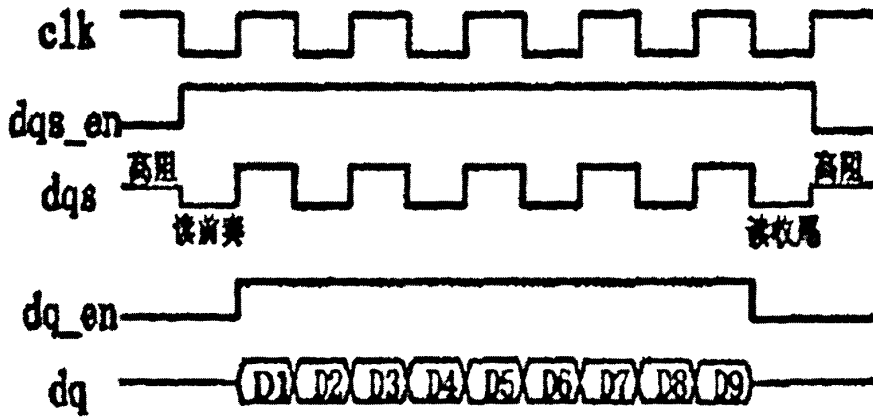


图 4.23 读数据通路时序图

第五章 DDR3 存储控制器物理层的设计实现

对于传输层送来的命令，物理层将对应信号按照DDR3标准中定义的时序和电气要求输出到存储总线上，同时根据传输层发送的时序要求控制命令之间的时间间隔^[24]。其主要功能是完成命令和地址的发送、数据选通信号的形成、写数据的发送以及读数据的接收。在DDR3控制器中，用户层传输的数据是256位的单沿触发数据，而控制器和外部的存储器之间的数据是128位的双沿触发数据，物理层写数据时要将单沿触发的数据转换为双沿触发，读数据时要将双沿触发的数据转换为单沿触发。

控制器的物理层由地址和命令发送、读数据接收和写数据发送三个模块组成。地址和命令模块发送和存储器地址及命令有关的片选信号、存储体地址、行地址、列地址以及行列选通等信号；数据发送模块传输数据信号、数据选通信号、数据屏蔽信号和使能信号^[25]；数据接收模块捕获从存储器读出的数据，并转换成控制器时钟域的信号送给DDR3控制器的传输层。

5.1 地址命令发送模块

地址命令发送模块接收控制器传输层送来的存储器命令、行地址和列地址，根据存储器对命令和地址信号的时序要求，将这些信号发送到存储器。存储器工作所需要的时钟信号也在该模块产生，并且时钟的上升沿位于地址和命令信号的中央位置。由于DDR3存储器中，地址和命令都是单数据率（SDR）传输信号，所以地址和命令发送模块在DDR3控制器中也是比较简单的一个模块，关键是要正确处理时钟和地址/命令的时序关系。从控制器传输层来的地址和命令信号由控制器核心工作时钟产生，而地址和命令的发送所需的时钟也是和CLK0信号同相位的，故地址和命令信号可经寄存器采用CLK0直接输出到存储器。为了使发送到存储器的地址和命令有最大的保持和建立时间，存储器的时钟则和控制器的核心工作频率反相^[26]。

5.2 数据发送模块

数据发送模块有两个功能：一是将写数据发送到存储器总线上，二是产生写数据选通信号DQS，并使DQS和写数据的中心对齐^[27]。

在 pad_ddr 模块中, 288 位的写数据被分割成高 144 位和低 144 位, 分别被正相时钟 clk 和反相时钟 tclk 采样到两组寄存器中, 形成正沿传输数据 data_pos 和负沿数据 data_neg。后级的二路选择器在 dqs_read 信号控制下分别将 data_pos 和 data_neg 选通到 dq[143:0], 形成双沿触发的 DQ 数据被发送到 DRAM 存储器中。在发送数据的同时 pad_ddr 模块产生与 dram 时钟同频同相的 DQS 信号发送到 DRAM 存储器中。写数据发送结构和时序分别如图 5.1 和图 5.2 所示。

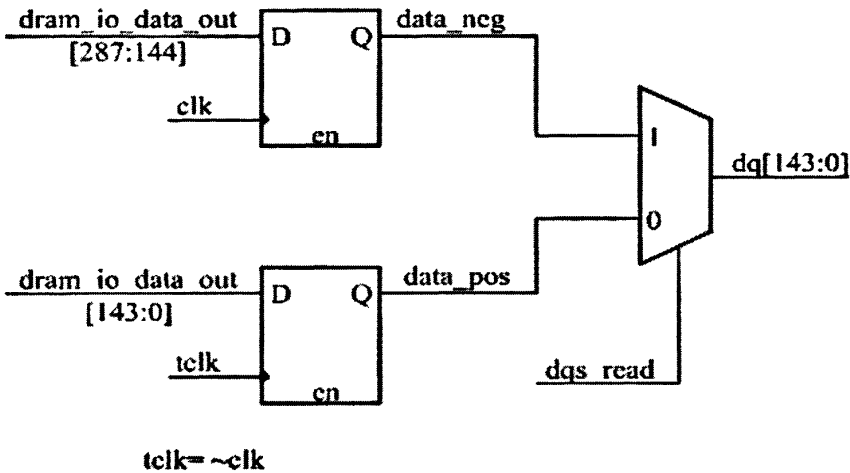


图 5.1 写数据发送结构

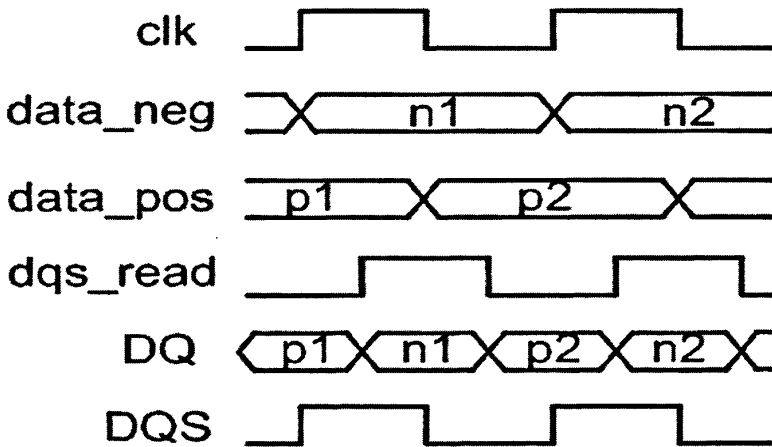


图 5.2 写数据发送时序

5.3 读数据接收模块

读数据接收模块用来捕获来自存储器的读出数据并且将其传输到DDR3控制器内部的核心时钟域。为了提高DDR3存储器的数据传输率，数据的接收使用同组的数据选通信号DQS作为接收时钟。考虑到信号的传输延迟以及工艺、电压、温度的影响，DQS延时前，需要对时序进行严格的分析^[28]。由于从存储器来的DQS和数据是边沿对齐的，所以要使用DQS作为接收时钟，就要将DQS向后推迟90°^[29]，这也是读数据接收模块的重要功能。

5.3.1 接收时钟

在读数据接收模块中，根据不同的存储器类型，每4位数据或8位数据使用一个接收时钟对读出数据进行采用接收，这样可以尽可能地减小数据到达接收寄存器的时间偏差，有利于接收数据的正确捕获，提高存储器的工作频率，从而提高存储系统的性能和可靠性。

根据DDR存储器规范，从存储器读出的数据和数据选通信号是边沿对齐的^[30]，因此要准确接收数据，作为接收时钟的数据选通信号的相位就必须后移四分之一时钟周期。为了达到此目的，首先要确定一个时钟周期需要的延时级数，然后在确定四分之一时钟周期所需的延时级数。

5.3.2 时钟相位检测

时钟信号的延时是通过延时单元级联实现的，为了确定延时一个时钟周期所需的延时单元级数，控制器采用了如图5.3所示的时钟相位检测逻辑。该逻辑由一个延时线dll_delay_line、两个延时单元dll_delay_element和两个相位检测寄存器组成，其中延时线dll_delay_line又由多级延时单元组成，其最大延时单元数取决于控制器的工作频率和每一级延时单元的延时，原则是延时线的最大延时稍大于一个时钟周期。

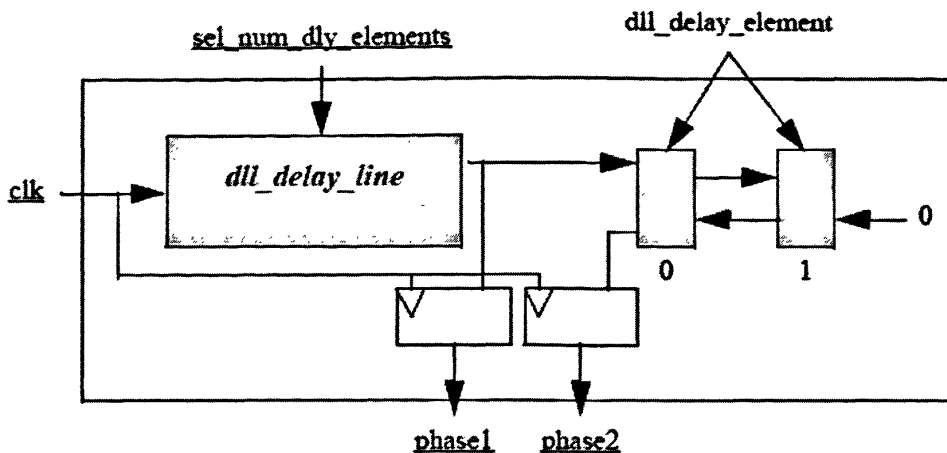


图 5.3 时钟相位检测逻辑

时钟相位检测连接的工作流程是：控制器通过寄存器给延时级数选择 `sel_num_dly_element` 赋一个较小的初始值，然后检查相位检测连接的输出端 `phase1` 和 `phase2`，如果两个输出都为低电平，则减小 `sel_num_dly_element` 值，如果 `phase2` 为高电平，则增加 `sel_num_dly_element` 值，如果 `phase1` 为高电平且 `phase2` 为低电平，则当前 `sel_num_dly_element` 的值便是一个时钟周期延时所需的级数。

5.3.1.2 接收时钟产生

通过时钟相位检测逻辑产生延时一个时钟周期所需的延时级数 `sel_num_dly_element` 后，该值右移2位（除以4）便得到产生 90° 时钟所需的延时单元数 `sel_num_dly_element_clk90`。CLK90的产生需要另外的延时线，其延时选择的输入就是 `sel_num_dly_element_clk90`，输入的时钟是 DQS，输出时钟便是接收数据所需的 90° 时钟。

5.3.1.3 物理实现的考虑

从上面数据接收时钟产生过程可知，计算延时级数和对时钟进行延时的逻辑是两套延时线，为了保证延时计算的精确，在物理实现上需要注意两点：一是两套延时线在布局时要尽量靠近，这样可以使两套延时线具有相近的温度、工艺和电压参数，从而具有相近的延时值。二是两套延时线在布线时要尽可能的相同，这也是物理保证二者具有相近的延时参数。

5.3.3 数据接收

为了准确接收从 DRAM 存储器返回的数据，在 DDR3 DRAM 控制器的请求调度模

块中，对应每一个存储体（Bank）设置了一个数据返回计数器 $b*_{data_rtn_cnt}[3:0]$ 。在向存储器的某一 Bank 发送 CAS 命令后，对应该 Bank 的数据返回计数便开始对 DRAM 时钟计数，当计数值达到模式寄存器 $mode_reg[6:4]$ 中保存的 CAS 延时值 CL 后，将会产生一个时钟周期宽度的脉冲信号 $DRAM_io_pad_enable$ ，该信号到达读数据接收模块后，启动模块中的读数据采样电路，开始采样从存储器返回的读取数据。

在 DDR3 控制器中，为了采样数据更稳定，采用了一种新型的采样电路结构。这种新型电路的思想是：利用多个数据采样寄存器，并使它们在读数据采用时钟 dqs_read 信号触发下轮流采样。这样可以将串行输入的多个数据采样到多个并行的通路中。再在后级利用多路选择器在 DRAM 时钟频率下从多个数据通路中轮流选择采样数据。在采样电路中，一共使用了 8 个采样寄存器，其中 4 个采样寄存器是正沿采样寄存器，4 个采样寄存器是负沿采样寄存器。在接收猝发为 8 的数据时，每个采样寄存器采样并保存一个输入数据，再由后端的两个 4 路选择器分别将正沿采样数据和负沿采样数据传输到后级。

采样电路的数据采样过程如下：

●读数据接收模块接收到从请求调度模块输入的 $DRAM_io_pad_enable$ 脉冲后分别启动计数器： $pad_ptr_pos[1:0]$ 、 $pad_pos_cnt[1:0]$ 和 $pad_neg_cnt[1:0]$ 。

●计数器 $pad_ptr_pos[1:0]$ ：对 dqs_read 信号的下降沿进行计数，每个 dqs_read 信号下降沿后 $pad_ptr_pos[1:0]$ 的值加 1。根据 $pad_ptr_pos[1:0]$ 的不同值，相应采样使能信号（ pad_ptr0_pos 到 pad_ptr3_pos ）被设置为 1。从 DRAM 输入的数据 DQ 由 dqs_read 的上升沿和下降沿分别采样到相应通道中（上升沿数据被采样到 $DRAM_pad_data0 \sim DRAM_pad_data3$ 中；下降沿数据被采样到 $DRAM_neg_pad_data0 \sim DRAM_neg_pad_data3$ 中）。

●计数器 $pad_pos_cnt[1:0]$ ：对 $DRAM_clk$ 的上升沿进行计数，每个 $DRAM_clk$ 上升沿后 $pad_pos_cnt[1:0]$ 的值加 1。四路选择器根据 $pad_pos_cnt[1:0]$ 的值从四个输入信号 $DRAM_pad_data0$ 、 $DRAM_pad_data1$ 、 $DRAM_pad_data2$ 和 $DRAM_pad_data3$ 中选择相应信号输出到 pad_pos_data 。

●计数器 $pad_neg_cnt[1:0]$ ：对 $DRAM_clk$ 的下降沿进行计数，每个 $DRAM_clk$ 下降沿后 $pad_neg_cnt[1:0]$ 的值加 1。四路选择器根据 $pad_neg_cnt[1:0]$ 的值从四个输入信号 $DRAM_neg_pad_data0$ 、 $DRAM_neg_pad_data1$ 、 $DRAM_neg_pad_data2$ 和 $DRAM_neg_pad_data3$ 中选择相应信号输出到 pad_neg_data 。

●两个 4 路选择器的输出数据 pad_pos_data 和 pad_neg_data 分别被后级的正沿采样寄存器和负沿采样寄存器在 $DRAM_clk$ 时钟的上升沿和下降沿采样得到 $io_DRAM_data_in_hi$ 和 $io_DRAM_data_in_lo$ 。

图 5.4 介绍了 DDR3 控制器数据采样电路的思想和过程。

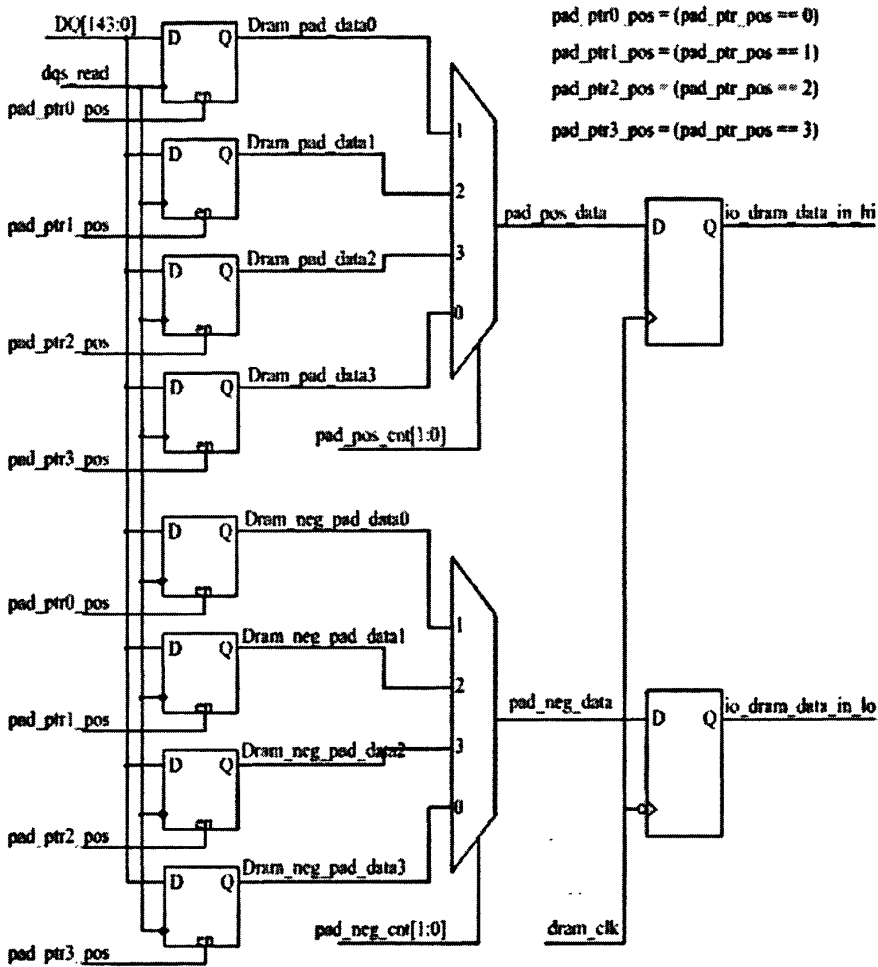


图 5.4 DDR3 控制器数据采样电路

DRAM_clk 时钟下上升沿采样的数据 io_DRAM_data_in_hi 和下降沿采样的数据 io_DRAM_data_in_lo 经过组装就形成了 256 位的单沿读数据 io_DRAM_data_in 和 32 位的单沿数据 ecc 校验码 io_DRAM_ecc_in, 并被输入到 DRAM 控制器传输层的读数据通路模块中。输入数据的组装如图 5.5 所示。随着物理层向传输层传输的第一个读入数据, 在读数据接收模块中会产生一个单周期宽度的脉冲信号 DRAM0_data_valid—读数据有效, 该信号一起被发送到 DRAM 控制器的传输层, 表示读出的数据是有效的。

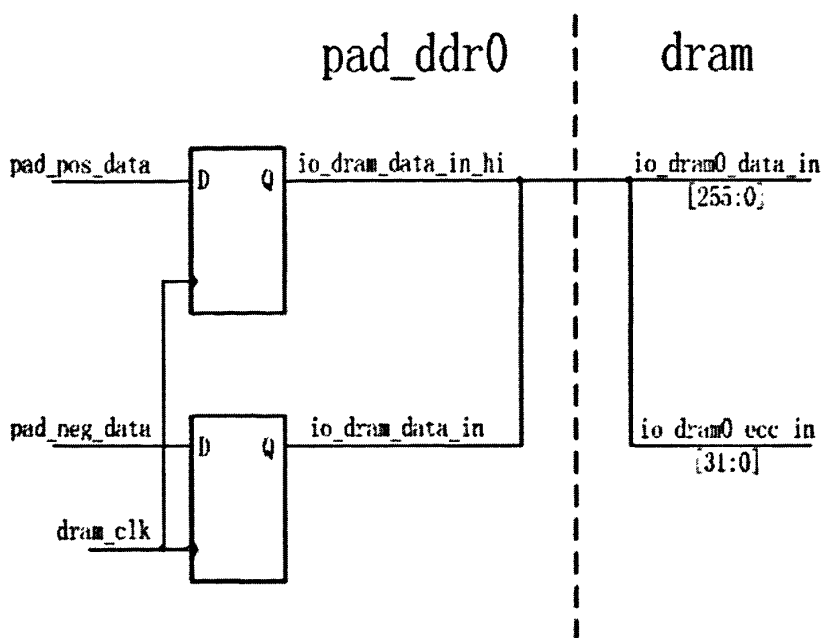


图 5.5 输入数据组装

第六章 结束语

6.1 研究工作总结

存储控制器是存储器的控制部件，同时也是处理器与存储器之间数据传输的桥梁，它的性能直接影响整个处理器的性能。处理器的数据经过存储控制器发送给存储器，来自存储器的数据经过存储控制器提供给处理器，如何能够高效地进行数据传输是存储控制器的研究重点。

本文结合DDR3和DDR2在关键技术上的改进，重点对DDR3存储控制器传输层和物理层的设计实现进行了描述，工作的重点如下：

1、对目前DDR SDRAM的发展状况进行了简述，分析了DDR3产生的原因和现状。重点分析了基于8-bit预取技术的DDR3存储控制器提速关键因素。

2、按照JESD79-3 规定的DDR3 SDRAM Standard，结合DDR2 SDRAM存储器的比较，对DDR3 SDRAM存储器优越性进行了深入的分析研究。

3、研究了DDR3 SDRAM的基本操作和各个信号的时序关系，这对存储控制器的设计有重要意义。主要涉及的问题包括DDR3 SDRAM的初始化操作的顺序进行，命令实现的功能，各个命令之间的时序要求，读写访问中的数据选通信号、数据屏蔽信号与数据信号的时序关系，DDR3 SDRAM的状态转移等。

4、提出DDR3存储控制器总体设计方案，并详细介绍了DDR3存储控制器传输层的实现，主要包括用户请求接口、控制器部件、错误检测和纠正ECC、存储体控制逻辑和仲裁器。

5、详细介绍了DDR3存储控制器物理层的实现。主要描述了读写数据通路的设计。其中写数据时DQS与DQ信号中心对齐方式的控制以及读数据的获取方法是实现中的难点。

6.2 后继工作

本文根据JEDEC绘制的使DDR的传输速率提升到3.2Gbps^[31]的蓝图，提出了一种DDR3存储控制器设计实现方案，该方案可以有效地降低存储器的访问延时，提高访存带宽。但受到时间和设计复杂性的限制，目前实现的控制器没有跟踪芯片温度的变化，而在实际运行过程中，温度的变化可以为控制器提供更多的性能优化机会，如采用自刷新技术等。

下一步的研究工作是将芯片工作温度作为控制器进行控制操作时的一个参考因素，进一步提高访存带宽，降低访存延迟。

致 谢

研究生阶段的学习即将结束，在二年多的学习生涯中，曾经得到过许多老师和同学的热情关怀和无私帮助，在此谨向他们表示最衷心的感谢和最诚挚的谢意！

衷心感谢我的导师窦强、李永进、李小芳老师！感谢您们在生活学习中对我的亲切关怀和悉心指导，是您们的远见卓识和对计算机技术发展的准确把握，促使我选择了本课题的研究；是您们的谆谆教诲与严格要求，促使我顺利完成论文。您们高深的学术造诣、纵观全局的洞察能力、幽默乐观的生活态度、对事业的不懈追求以及平易近人的处世态度拓宽了学生的人生视野，指引我前进的方向！衷心祝愿您们身体健康、一生平安！

师恩难忘！

非常感谢在论文前期给我帮助和关心的戴华东、庞征斌、李晋文老师！您们缜密的思维、敏锐的观察力和废寝忘食的敬业精神，给我留下了深刻印象。是您们悉心的指导和无私的帮助，使我能顺利确定课题研究的方向！

感谢实验室的师兄弟，他们在平时的生活中给了我很多的帮助，让我感受到同窗友情的温馨！特别是王勇，在论文修改和版面设计过程中给了我很大的帮助！

感谢继续教育学院二队的领导和同志们，能与你们一起学习、生活、娱乐、相伴走过两年的科大美好时光，是我一生中值得回忆的事情！祝愿大家学习顺利、工作愉快！

感谢武警北京总队的领导和同志们，谢谢你们给了我这次学习的机会并在此期间给我的关心、支持和帮助！

感谢所有关心和爱护我的人！

深深的感谢我的父母和亲人们！他们的关心和支持是我最大的学习动力，在我成长的每一个足印里，都倾注了他们的心血和汗水，再次感谢我远方的亲人！

最后，诚挚地感谢在百忙之中为评阅此论文而付出辛勤劳动的各位专家和学者！

参考文献

- [1] Ashok K.Sharma,Advanced Semiconductor Memories – Architectures, Designs and Applications,2003
- [2] 邓丽.高带宽低延迟的DDR2存储器控制器的研究与实现.国防科技大学硕士学位论文,2006.11
- [3] <http://space.danawa.com.cn/article/4235.html>
- [4] http://topic.expreview.com/2007-07-13/1184319604d2969_1.html
- [5] <http://km.funddj.com/KMDJ/Wiki/WikiViewer.aspx?keyid=253a46bc-2b8e-4ec0-b32f-d7336e363d70>
- [6] 朱卫东,蒋本善.计算机硬件技术基础.高等教育出版社,2001:142-148
- [7] <http://topic.express.com/2007-07-13/.htm>
- [8] (美)布达科.PCI Express系统体系结构标准教材.电子工业出版社,2005.11
- [9] http://www.oakpc.com/mimg/eart/news_16508_6.html
- [10] 万轶,窦强等.第十二届计算机工程与工艺全国学术年会.中国计算机学会计算机工程与工艺专业委员会,内蒙古,呼和浩特,2008:122-125
- [11] Matt Dipaolo. High-Performance DDR3 SDRAM Interface inVirtex-5 Devices, XAPP867 (v1.0), 2007.9
- [12] Jeff Janzen, DDR2 Offers New Features and Functionality, Micron Technology, 2003
- [13] JEDEC STANDARD DDR2 SDRRM SPECIFICAT10N.2005.1
- [14] DDR3 SDRAM Specification Device Operation & Timing Diagram.May 2008 ,revision 0.5
- [15] <http://bbs.3671041.com>
- [16] JenniferTran,Ratima,Synthesizable,266Mbits/SddrsdramController.XILINX,2001 .1
- [17] Arthi Palanisamy.DDR2 Controller (267 MHz and Above) Using Virtex-4 Device,XILINX,2006.2.
- [18] UltraSPARC T1™ Supplement to the UltraSPARC Architecture ,2005:315-317
- [19] 史美萍,窦文华.基于EPLD 的PCI总线仲裁器的设计与实现.电子技术应用
- [20] 乐大珩.DDR2控制器设计方案.国防科技大学计算机学院微电子与微处理器研究所610室,2005.7.1
- [21] 刘瑰,朱鸿宇.通用DDR SDRAM控制器的设计.微型机与应用,2004.11:23-25

- [22] 江先阳,刘新春,张佩街,孙凝晖,徐志伟.计算密集型体系集成DDR SDRAM控制器设计.计算机工程与科学, 2006.11:96-101
- [23] 杨少波,王勤民,张帆,曲晶.DDR内存接口的设计与实现.微计算机信息(嵌入式与SOC), 2005.11
- [24] Lakshmi Gopalakrishnan, DDR Controller Using Virtex-4 Devices, XILINX, 2006.2
- [25] 邵贝贝,刘慧根.微控制器原理与开发技术.清华大学出版社,1991.12
- [26] Olivier Despaux,DDR SDRAM Controller Using Virtex-4 FPGA Device,XILINX,2006.3
- [27] Feng Lin,A Register-Controller Symmetrical DLL for Double-Data-Rate DRAM,IEEE Journal of Solid-State Circuits,1999.4
- [28] Digital Delay Line_v3.0,IBM,2004.7
- [29] Digital Delay Line for Cu-11,IBM,2004.9
- [30] DDR SDRAM Functionality and Controller Read Data Capture,Micron,1999
- [31] <http://www.educhina.com/Tech Class/CE>

作者在学期间发表的论文

- [1] 万轶, 窦强, 李永进, 庞永超. DDR3关键技术分析和研究. 第十二届计算机工程与工艺学术年会, 2008:122-124
- [2] 庞永超, 万轶等. 基于GIS的某行业指挥定位系统研究与设计. 第十二届计算机工程与工艺学术年会, 2008:202-205

作者: [万轶](#)
学位授予单位: [国防科学技术大学](#)

本文读者也读过(8条)

1. [刘梅](#) [DDR3 SDRAM控制器物理层DLL的设计与实现](#)[学位论文]2009
2. [万轶](#). [窦强](#). [李永进](#). [庞永超](#) [DDR3关键技术分析和研究](#)[会议论文]-2008
3. [王继斌](#) [DDR3存储器前沿技术分析](#)[期刊论文]-[科技信息](#)2009(34)
4. [史林森](#) [基于自主CPU的DDR3系统协同仿真与设计](#)[学位论文]2011
5. [徐洋](#) [高速网络设备中DDR3存储器的性能分析](#)[期刊论文]-[硅谷](#)2010(17)
6. [李博](#) [高速并行总线-DDR接口噪声与时序分析](#)[学位论文]2009
7. [李晋文](#). [胡军](#). [曹跃胜](#). [史林森](#). [肖立权](#) [DDR3时序分析与设计](#)[会议论文]-2011
8. [陈昊](#) [DDR存储控制器的设计与应用](#)[学位论文]2006

本文链接: http://d.g.wanfangdata.com.cn/Thesis_Y1523575.aspx