**7.3 EMIF**

**7.3.1 Введение**

**7.3.1.1 Особенности**

Общие характеристики модуля EMIF следующие:

- 16-разрядный канал передачи данных во внешнюю память SDRAM

- Один 128-битный интерфейс OCPIP 2.2

- Поддержка следующих типов памяти:

- mDDR (LPDDR1)

- DDR2

- DDR3

Поддерживаются конфигурации внешней памяти:

- Емкость устройства памяти

- Адресация до 1 Гбайт

- Гибкие схемы мультиплексирования адресов банк/ряд/столбец/чип-селект

- Поддерживается функция усиления драйверов устройств для мобильной DDR

- Поддержка следующих задержек CAS:

- DDR2 => 2, 3, 4, 5, 6 и 7

- DDR3 => 5, 6, 7, 8, 9, 10 и 11

- mDDR => 2, 3 и 4

- Поддерживает следующее количество внутренних банков:

- DDR2 => 1, 2, 4 и 8

- DDR3 => 1, 2, 4 и 8

- mDDR => 1, 2 и 4

- Поддерживаются страницы размером 256, 512, 1024 и 2048 слов

- Поддерживает длину серии 8 (последовательная серия)

- Выравнивание/калибровка записи/чтения и обучение глазкам данных в сочетании с DID

- Режимы Self Refresh и Power-Down для низкого энергопотребления:

- Гибкое сопоставление адресов OCP и DDR для поддержки частичного самообновления массива в LPDDR1, DDR2 и DDR3.

- Самообновление с контролем температуры для LPDDR1 и DDR3 с датчиком температуры на кристалле.

- Периодическая калибровка ZQ для DDR3

- ODT на DDR2 и DDR3

- Приоритетное планирование обновлений

- Программируемая частота обновления SDRAM и счетчик отставания

- Программируемые временные параметры SDRAM

- Режимы Big и Little Endian

**7.3.1.2 Неподдерживаемые функции EMIF**

Следующие функции модуля EMIF4DC не поддерживаются в данном устройстве.

**Таблица 7-204. Неподдерживаемые функции EMIF**

|  |  |
| --- | --- |
| **Feature** | **Reason** |
| 32-bit data | Только 16 бит, выведенных наружу |
| Multiple DDR banks | Только 1 CS/ODT подключен |
| DDR2 CAS Latency 2 | Не поддерживается DID |
| Hardware leveling | Кремниевая ошибка. Необходимо использовать процедуру программного выравнивания. См. AM335x  Микропроцессоры ARM Cortex-A8 (MPU) Silicon Errata  (литературный номер SPRZ360). |

**7.3.2 Интеграция**

**7.3.2.1 Атрибуты подключения EMIF**

Общие атрибуты подключения для EMIF приведены в таблице 7-205.

**Таблица 7-205. Атрибуты подключаемости EMIF**

|  |  |
| --- | --- |
| **Attributes** | **Type** |
| Power Domain | Peripheral Domain |
| Clock Domain | PD\_PER\_L3\_GCLK (OCP)  PD\_PER\_EMIF\_GCLK (Func) |
| Reset Signals | POR\_N |
| Idle/Wakeup Signals | Smart Idle |
| Interrupt Requests | 1 interrupt to MPU Subsystem (DDRERR0) |
| DMA Requests | None |
| Physical Address | L3 Fast Slave Port |

**7.3.2.2 Управление тактовым генератором EMIF**

Интерфейс EMIF4 OCP (ocp\_clk) тактируется тактовым генератором L3 Fast, поступающим от Core PLL. Макросы DDR команд и данных DDR получают тактовую частоту от DDR PLL. PRCM делит этот тактовый сигнал на два, чтобы создатьфункциональный тактовый генератор EMIF (m\_clk).

OCP и функциональное тактирование могут быть асинхронными, поскольку синхронизация управляется во внутреннем FIFO EMIF4 внутреннего FIFO (EMIF4 установлен в асинхронный режим).

**Таблица 7-206. Тактовые сигналы EMIF**

|  |  |  |  |
| --- | --- | --- | --- |
| **Clock Signal** | **Maximum Frequency** | **Reference Source** | **Comments** |
| ocp\_clk  (Interface clock) | 200 MHz | CORE\_CLKOUTM4 | pd\_per\_l3\_gclk  From PRCM |
| ocp\_clk  (Interface clock) | 200 MHz(1) | DDR PLL CLKOUT / 2 | pd\_per\_emif\_gclk  From PRCM |
| cmd0\_dfi\_clk  cmd1\_dfi\_clk  cmd2\_dfi\_clk  data0\_dfi\_clk  data1\_dfi\_clk  (Macro clocks) | 400 MHz(1) | DDR PLL CLKOUT | clkout\_po  From DDR PLL |

(1) Максимальная частота зависит от типа DDR. Максимальную частоту DDR для вашего устройства можно узнать в процессоров AM335x Sitara (литературный номер SPRS717).

**7.3.2.3 Список выводов EMIF**

Сигналы внешнего интерфейса EMIF/DDR приведены в таблице 7-207.

**Таблица 7-207. Список выводов EMIF**

|  |  |  |
| --- | --- | --- |
| **Pin** | **Type** | **Description** |
| DDR\_CK  DDR\_NCK | O | Differential clock pair |
| DDR\_CKE | O | Clock enable |
| DDR\_CSn0 | O | Chip select |
| DDR\_RASn | O | Row address strobe |
| DDR\_CASn | O | Column address strobe |
| DDR\_WEn | O | Write enable |
| DDR\_BA[2:0] | O | Bank address |
| DDR\_A[15:0] | O | Row/column address |
| DDR\_DQS[1:0] | I/O | Data strobes |
| DDR\_DQSn[1:0] | I/O | Complimentary data strobes |
| DDR\_DQM[1:0] | O | Data masks |
| DDR\_D[15:0] | I/O | Data |
| DDR\_ODT | O | On-die termination |
| DDR\_RESETn | O | DDR device reset |
| DDR\_VREF | I | I/O Voltage reference |
| DDR\_VTP | I | VTP compensation pin |

**7.3.3 Функциональное описание**

**7.3.3.1 Описания сигналов**

Сигналы контроллера памяти DDR2/3/mDDR показаны на рисунке 7-201 и описаны в таблице 7-208.

Они включают следующие характеристики:

- Максимальная ширина шины данных (DDR\_D[15:0]) составляет 16 бит.

- Шина адреса (DDR\_A[15:0]) имеет ширину 16 бит с дополнительными 3 контактами адреса банка (DDR\_BA[2:0]).

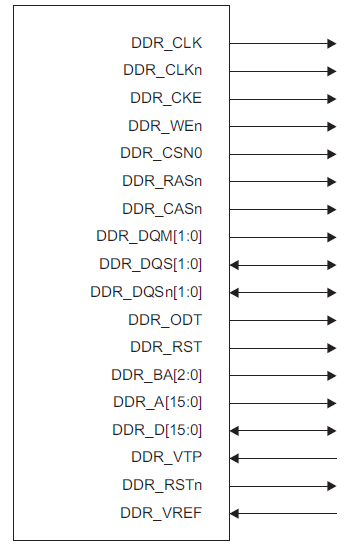
- Два дифференциальных выходных тактовых генератора (DDR\_CK и DDR\_nCK), управляемых внутренними источниками тактовой частоты

- Командные сигналы: стробирование адреса строки и столбца (DDR\_RASn и DDR\_CASn), строб разрешения записи (DDR\_WEn), строб данных (DDR\_DQS[1:0] и DDR\_DQSn[1:0]) и маска данных (DDR\_DQM[1:0]).

- Один сигнал выбора микросхемы (DDR\_CSN0) и один сигнал разрешения тактового генератора (DDR\_CKE).

- Один выходной сигнал завершения работы устройства (DDR\_ODT).

**Рисунок 7-201. Сигналы контроллера памяти DDR2/3/mDDR**



**Таблица 7-208. Описания сигналов контроллера памяти DDR2/3/mDDR**

|  |  |
| --- | --- |
| **Pin** | **Description** |
| DDR\_D[15:0] | Двунаправленная шина данных. Вход для чтения данных и выход для записи данных. |
| DDR\_A[15:0] | Выход внешнего адреса. |
| DDR\_CSN0 | Выход сhip select. |
| DDR\_DQM[1:0] | Активно-низкий выход маски данных. |
| DDR\_CLK/DDR\_CLKn | Дифференциальные выходы тактового генератора. Все сигналы интерфейсов DDR2/3/mDDR синхронизированы с этими выходами. |
| DDR\_CKE | Разрешение тактового генератора. Используется для выбора операций Power-Down и Self-Refresh. |
| DDR\_CASn | Активно-низкий стробирующий сигнал адреса столбца. |
| DDR\_RASn | Активно-низкий стробирующий сигнал адреса строки. |
| DDR\_WEn | Разрешение записи с активным нижним уровнем. |
| DDR\_DQS[1:0]/DDR\_DQSn[1:  0] | Дифференциальные двунаправленные сигналы стробирования данных. Входы выровнены по краям при чтении и выровнены по центру выходы при записи. |
| DDR\_ODT | Сигнал завершения работы с внешней памятью DDR2/3 SDRAM. ODT не поддерживается для mDDR. |
| DDR\_BA[2:0] | Выходы управления адресами банка-памяти. |
| DDR\_VREF | Опорное напряжение контроллера памяти. Это напряжение должно подаваться извне. Подробнее см. для получения более подробной информации. |
| DDR\_VTP | Подключение компенсационного резистора VTP DDR2/3/mDDR. |
| DDR\_RESETn | Выход сброса. Асинхронный сброс для устройств DDR3. |

**7.3.3.2 Управление тактовым генератором**

Тактовая частота DDR2/3/mDDR берется непосредственно с выхода VCO DDR PLL. Частоту DDR\_CLK можно определить по следующей формуле:

Частота DDR\_CLK = (входная тактовая частота DDRPLL x mulitplier)/((pre-divider+1)\*post-divider)

Второй выходной тактовый генератор контроллера памяти DDR2/3/mDDR DDR\_CLKn является обратным по отношению к DDR\_CLK. Вы можете изменить множитель, предварительный делитель и постделитель, чтобы получить желаемую частоту DDR\_CLK.

Подробную информацию о DDR PLL см. в разделе 8.1, Модуль управления питанием и тактовым генератором (PRCM).

**7.3.3.3 Обзор подсистемы контроллера памяти DDR2/3/mDDR**

Контроллер памяти DDR2/3/mDDR может беспроблемно подключаться к большинству стандартных устройств DDR2/3/mDDR SDRAM и поддерживает такие функции, как режим самообновления и приоритетное обновление. Кроме того, он обеспечивает гибкость за счет программируемых параметров, таких как частота обновления, задержка CAS и многие временные параметры SDRAM. Подсистема DDR2/3/mDDR состоит из следующих элементов:

- Контроллер памяти DDR2/3/mDDR

- Макрос команд

- Макрос данных

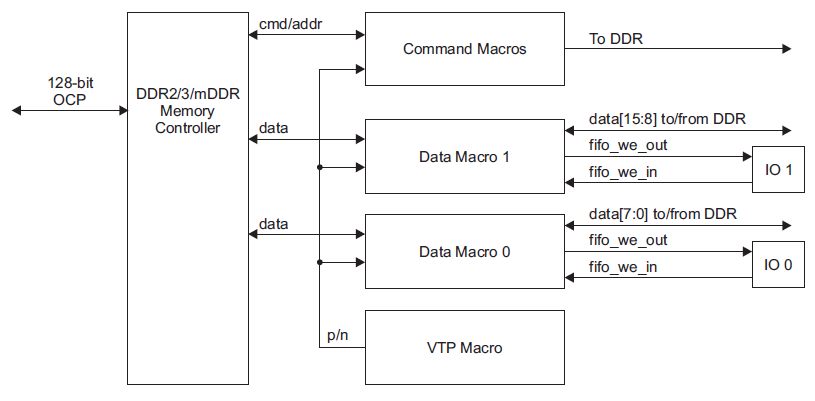
- Макрос контроллера VTP

- IOs для затвора DQS

Подсистема поддерживает устройства DDR2/DDR3 и mDDR(LPDRR1), совместимые со стандартом JEDEC. Подсистема не поддерживает CAS-задержку 2 для DDR2 из-за ограничений макросов данных и команд. Поддерживается 128-битный интерфейс OCP шириной 128 бит на стороне ядра для программирования. Подсистема может использоваться для подключения к 16-разрядным устройствам памяти.

На рисунке 7-202 показана блок-схема подсистемы DDR2/3/mDDR.

**Рисунок 7-202. Блок-схема подсистемы DDR2/3/mDDR**



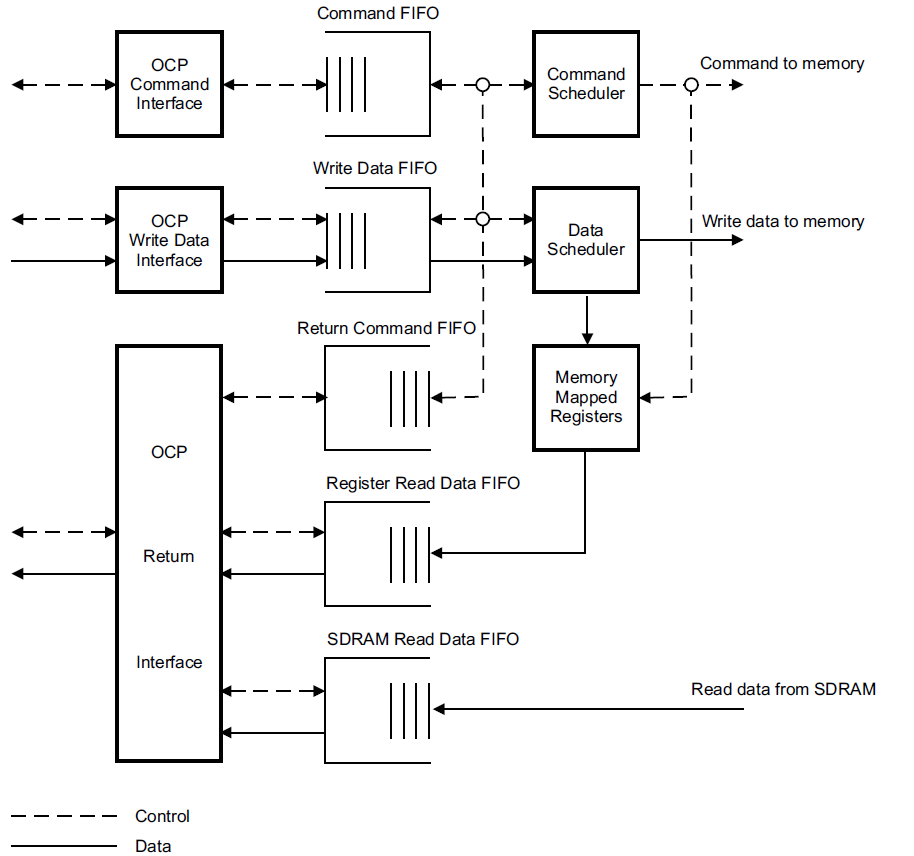
Где fifo\_we\_out = разрешающий выход DQS для согласования по времени между DQS и системным (память) тактовым генератором. fifo\_we\_in = разрешающий вход DQS для согласования по времени между DQS и системным тактовым генератором (память).

**7.3.3.3.1 Интерфейс контроллера памяти DDR2/3/mDDR**

Для эффективного перемещения данных из ресурсов чипа во внешнее устройство DDR2/3/mDDR SDRAM контроллер памяти DDR2/3/mDDR использует буфер FIFO команд, FIFO данных записи, FIFO команд возврата и два FIFO данных чтения. Назначение каждого FIFO описано ниже.

На рисунке 7-203 показана блок-схема FIFO контроллера памяти DDR2/3/mDDR. Команды, данные записи и данные чтения поступают в контроллер памяти DDR2 параллельно друг другу. Одна и та же периферийная шина используется для записи и чтения данных из внешней памяти, а также из внутренних регистров с привязкой к памяти.

**Рисунок 7-203. Блок-схема FIFO контроллера памяти DDR2/3/mDDR**



В командном FIFO хранятся все команды, поступающие на командный интерфейс OCP.

В FIFO данных записи хранятся данные записи для всех транзакций записи, поступающих на интерфейс данных записи OCP.

В FIFO команд возврата хранятся все транзакции возврата, которые должны быть выданы на интерфейс возврата OCP. К ним относятся команды возврата статуса записи и возврата данных чтения.

В двух FIFO данных чтения хранятся данные чтения, которые должны быть отправлены на интерфейс возврата OCP. Один FIFO данных для чтения хранит данные из регистров, отображаемых в памяти, а другой FIFO данных для чтения хранит данные для чтения из внешней памяти.

**7.3.3.3.2 Макрос данных**

Макрос данных состоит из 8 каналов данных, одной пары комплементарных стробов (одна пара на 8 бит данных) и одного канала маски данных (один на 8 бит данных).

Макрос данных состоит из макроса данных PHY, DLL и IO, интегрированных в макрос.

Макрос данных - это двунаправленный интерфейс. Он используется для передачи данных от контроллера памяти к внешней микросхемы памяти во время операции записи, а также для приема данных из памяти и передачи их в контроллер памяти во время операции чтения.

Во время операции записи макрос данных переводит 32/16-битные слова из контроллера памяти в 8-битные слова и передает их с удвоенной скоростью в память вместе со стробом. Строб выравнивается по центру по отношению к данным. Запись данных в память может быть запрещена с помощью сигнала маски данных.

Во время операции чтения макрос данных получает 8-битные данные DDR вместе со стробом, преобразует их в 32/16-битные слова и передает их контроллеру памяти вместе с сигналами read-valid.

**7.3.3.3.3 Командный макрос**

Он состоит из командного макроса PHY, DLL и интегрированных IO. Командный макрос действует как однонаправленный макрос, передающий адрес и управляющие биты от контроллера памяти к микросхеме памяти.

Тактовые импульсы DDR\_CLK и DDR\_CLKn используются памятью для регистрации команды и адреса, передаваемых по каналам передачи. Все сигналы адреса и управления передаются по тактовому центру относительно DDR\_CLK и DDR\_CLKn. Память, по положительному фронту DDR\_CLK и отрицательному фронту DDR\_CLKn, производит выборку всех адресных и управляющих сигналов.

**7.3.3.3.4 Макрос контроллера VTP**

Макрос контроллера VTP оценивает производительность кремния при текущем напряжении, температуре и технологическом процессе (VTP), чтобы позволить драйверам ввода-вывода установить постоянный заданный импеданс выходного драйвера. Контроллер работает сравнивая импеданс драйвера с внешним опорным резистором и регулируя импеданс драйвера для достижения соответствия импедансу. Контроллер VTP поддерживает следующие функции:

- Контроллер VTP генерирует информацию о напряжении, температуре и процессе (VTP) для передачи драйверам ввода-вывода устройства.

- Требуется тактовый вход от ядра, работающего на частоте 20 МГц или ниже.

- 56 тактов необходимо для обеспечения первоначальной установки выходов VTP после сброса.

- Может использоваться в режиме статического или динамического обновления

- Контроллер VTP имеет внутреннюю фильтрацию шумов, что позволяет ему контролировать ложные запросы на обновление из-за шума

Импеданс драйверов и выводов необходимо часто обновлять даже во время работы. В таких сценариях, где напряжение и температура являются переменными, макрос VTP может быть сконфигурирован в режиме динамического обновления. В отличие от этого, статический режим работы не позволяет динамически калибровать импеданс ввода-вывода, и следовательно, потребляет меньше энергии по сравнению с режимом динамического обновления.

Возможно, что в условиях повышенного шума контроллер динамического обновления может генерировать слишком частые запросы на обновление. Шум может заставить контроллер запросить изменение импеданса, которое может быть отменено на последующих тактовых циклах. Чтобы предотвратить чрезмерное количество запросов контроллера изменения импеданса, включен цифровой фильтр, который может быть настроен на регулирование частоты обновления. Например, если пользователь настроит значение фильтра как F2=0, F1=1 и F0=1, то обновление импеданса будет выполнено только в том случае, если от контроллера VTP поступит четыре последовательных запроса на обновление. Рекомендуется использовать значение фильтра 011'b.

В таблице 7-209 приведены сведения о конфигурации цифрового фильтра.

**Таблица 7-209. Конфигурация цифрового фильтра**

|  |  |  |  |
| --- | --- | --- | --- |
| **F2** | **F1** | **F0** | **Description** |
| 0 | 0 | 0 | Выкл. |
| 0 | 0 | 1 | Обновление по 2 последовательным запросам на обновление |
| 0 | 1 | 0 | Обновление по 3 последовательным запросам на обновление |
| 0 | 1 | 1 | Обновление по 4 последовательным запросам на обновление |
| 1 | 0 | 0 | Обновление по 5 последовательным запросам на обновление |
| 1 | 0 | 1 | Обновление по 6 последовательным запросам на обновление |
| 1 | 1 | 0 | Обновление по 7 последовательным запросам на обновление |
| 1 | 1 | 1 | Обновление по 8 последовательным запросам на обновление |

**7.3.3.3.5 Входы/выходы DQS-Gate**

Для эффективного моделирования задержки ввода/вывода на сигнале стробирования DQS во время запроса на чтение (приемник DQS и входы/выходы драйвера CLK), предполагается, что сигнал будет зациклен на одном входе/выходе, соединяющем fifo\_we\_in и fifo\_we\_out. Задержка на плате и в памяти, будучи достаточно постоянной при вариациях PTV, калибруется в IDID с помощью компенсированной линии задержки. Обратная связь выполняется на уровне матрицы без вывода сигналов на уровень корпуса. Каждый макрос данных поддерживает компенсацию задержки независимо друг от друга. Макросы данных и команд отвечают за компенсацию времени полета на уровне системы. Ниже перечислены управляющие воздействия, поддерживаемые подсистемой контроллера DDR2/3/mDDR.

- Выравнивание DDR\_DQS по отношению к DDR\_CLK во время цикла записи: Для работы с DDR3 инициируйте механизм выравнивания записи на каждом ранге по очереди, чтобы зафиксировать правильные настройки задержки для выравнивания DDR\_DQS с тактовым генератором DDR\_CLK для каждой памяти. Если вы хотите сделать это вручную, вы можете записать в управляющий регистр, который контролирует задержку DDR\_DQS по отношению к позиции тактового генератора DDR\_CLK. Для получения заданного перекоса для центрирования DDR\_DQS относительно тактового генератора на SDRAM можно запрограммировать следующий регистр.

Data Macro 0/1 Write DQS Slave Ratio Register=256 x ([задержка команды] - [задержка DDR\_DQS]) /

Период тактового генератора DDR\_CLK.

- Выравнивание ADDR/CMD по времени DDR\_CLK

- Выравнивание DDR\_DQ[15:0] по отношению к DDR\_DQS во время операции записи

- Смещение DDR\_D[15:0] по отношению к DDR\_DQS во время операции чтения

- Выравнивание окна FIFO WE

**7.3.3.4 Сопоставление адресов**

Контроллер памяти DDR2/3/mDDR рассматривает внешнюю память DDR2/3/mDDR SDRAM как один непрерывный блок памяти. Это утверждение справедливо независимо от количества устройств памяти, расположенных в пространстве выбора микросхем. Контроллер памяти DDR2/3/mDDR получает запросы на доступ к памяти DDR2/3/mDDR вместе с 32-битным логическим адресом от остальной части системы. В свою очередь, контроллер памяти DDR2/3/mDDR использует логический адрес для генерации адреса строки/страницы, столбца и банка для DDR2/3/mDDR SDRAM.

Количество используемых битов адреса столбца, строки и банка определяется полями IBANK, RSIZE и PAGESIZE (см. таблицу 7-210). Контроллер памяти DDR2/3/mDDR использует до 16 бит для адреса строки/страницы.

**Таблица 7-210. Информация о полях IBANK, RSIZE и PAGESIZE**

|  |  |  |
| --- | --- | --- |
| **Bit Field** | **Bit Value** | **Bit Description** |
| RSIZE |  | Определяет количество адресных линий, подключаемых к устройству памяти DDR2/3/mDDR |
| 0 | 9 рядов битов |
| 1h | 10 рядов битов |
| 2h | 11 рядов битов |
| 3h | 12 рядов битов |
| 4h | 13 рядов битов |
| 5h | 14 рядов битов |
| 6h | 15 рядов битов |
| 7h | 16 рядов битов |
| PAGESIZE |  | Определяет размер каждой страницы внешнего устройства памяти DDR2/3/mDDR |
| 0 | 256 слов (требуется 8 битов адреса столбца) |
| 1h | 512 слов (требуется 9 битов адреса столбца) |
| 2h | 1024 слова (требуется 10 битов адреса столбца) |
| 3h | 2048 слов (требуется 11 битов адреса столбца) |
| IBANK |  | Определяет количество внутренних банков на внешнем устройстве памяти DDR2/3/mDDR |
| 0 | 1 банк |
| 1h | 2 банка |
| 2h | 4 банка |
| 3h | 8 банков |
| EBANK |  | Определяет количество chip-select микросхем контроллера памяти DDR2/3/mDDR |
| 0 | Только CS0 |
| 1h | Зарезервировано |

При обращении к SDRAM, если поле REG\_IBANK\_POS в регистре SDRAM Config установлено в 0, а поле поле REG\_EBANK\_POS в регистре SDRAM Config 2 также установлено в 0, контроллер памяти DDR2/3/mDDR использует три поля, IBANK, EBANK и PAGESIZE в регистре SDRAM Config для определения соответствия между адресом источника и строкой, столбцом, банком и выбором микросхемы SDRAM. Если поле REG\_IBANK\_POS в регистре SDRAM Config установлено в 1, 2 или 3, или поле REG\_EBANK\_POS в регистре SDRAM Config 2 установлено в 1, контроллер памяти DDR2/3/mDDR использует 4 поля -IBANK, EBANK, PAGESIZE и ROWSIZE в регистре SDRAM Config для определения соответствия между адресом источника и строкой, столбцом, банком и chip-select SDRAM. Во всех случаях контроллер памяти DDR2/3/mDDR рассматривает свое адресное пространство SDRAM как единый логический блок, независимо от количества физических устройств или от того, сопоставлены ли они между собой, а также независимо от того, отображены ли эти устройства на 1 или 2 чип-селектах контроллера памяти DDR2/3/mDDR.

**7.3.3.4.1 Сопоставление адресов при REG\_IBANK\_POS=0 и REG\_EBANK\_POS=0**

При REG\_IBANK\_POS=0 и REG\_EBANK\_POS=0 схема отображения адресов действует следующим образом адрес источника увеличивается через границы страниц устройств памяти DDR2/3/mDDR, контроллер DDR2/3/mDDR переходит на ту же страницу в следующем банке текущего устройства DDR\_CSn[0]. Это перемещение по банкам текущего переходит к той же странице в следующем устройстве (если EBANK=1, DDR\_CSn[1]) и проходит через ту же страницу во всех ее банках перед переходом к следующей странице в первомустройстве (DDR\_CSn[0]). Контроллер DDR2/3/mDDR использует это перемещение по внутренним банкам и чип-селектам, выбирая при этом одну и ту же страницу, чтобы максимизировать количество открытых банков памяти DDR2/3/mDDR в общем пространстве устройств памяти DDR2/3/mDDR.

Таким образом, контроллер DDR2/3/mDDR может держать открытыми максимум 16 банков (8 внутренних банков и 2 чип-селекта) одновременно и может чередовать их между собой.

**Таблица 7-211. Сопоставление адресов OCP и адресов DDR2/3/mDDR для REG\_IBANK\_POS=0 и REG\_EBANK\_POS=0**

|  |  |  |  |
| --- | --- | --- | --- |
| **Logical Address** | | | |
| **Row Address** | **Chip Select** | **Bank Address** | **Column Address** |
|  | # битов, определяемых EBANK из  SDRCR | # Количество битов, определяемое IBANK из  SDRCR | # Количество битов, определяемое PAGESIZE  в SDRCR |
| 16 bits | EBANK=0 => 0 bits | IBANK=0 => 0 bits | PAGESIZE=0 => 8 bits |
|  | EBANK=1 => 1 bit | IBANK=1 => 1 bit | PAGESIZE=1 => 9 bits |
|  |  | IBANK=2 => 2 bits | PAGESIZE=2 => 10 bits |
|  |  | IBANK=3 => 3 bits | PAGESIZE=3 => 11 bits |

**7.3.3.4.2 Сопоставление адресов при REG\_IBANK\_POS = 1 и REG\_EBANK\_POS = 0**

При REG\_IBANK\_POS = 1 и REG\_EBANK\_POS = 0 чередование банков внутри устройства (на один чип-селект) ограничено 4 банками. Однако между двумя селекторами микросхем можно чередовать банки. Таким образом, контроллер контроллер DDR2/3/mDDR может держать открытыми максимум 16 банков (8 внутренних банков на 2 селектора микросхем) одновременно, но может чередовать только восемь из них.

**Таблица 7-212. Сопоставление адресов OCP и адресов DDR2/3/mDDR для REG\_IBANK\_POS=1 иREG\_EBANK\_POS=0**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Address** | | | | |
| **Bank Address[2]** | **Row Address** | **Chip Select** | **Bank Address[1:0]** | **Column Address** |
| # Количество битов, определяемое  IBANK из SDRCR | # Количество битов, определяемое  RSIZE из SDRCR | # битов, определяемых  EBANK из SDRCR | # Количество битов, определяемое  IBANK из SDRCR | # Количество битов, определяемое  PAGESIZE из SDRCR |
| IBANK=0 => 0 bits | RSIZE=0 => 9 bits | EBANK=0 => 0 bits | IBANK=0 => 0 bits | PAGESIZE=0 => 8 bits |
| IBANK=1 => 0 bits | RSIZE=1 => 10 bits | EBANK=1 => 1 bit | IBANK=1 => 1 bit | PAGESIZE=1 => 9 bits |
| IBANK=2 => 0 bits | RSIZE=2 => 11 bits |  | IBANK=2 => 2 bits | PAGESIZE=2 => 10 bits |
| IBANK=3 => 1 bit | RSIZE=3 => 12 bits |  | IBANK=3 => 3 bits | PAGESIZE=3 => 11 bits |
|  | RSIZE=4 => 13 bits |  |  |  |
|  | RSIZE=5 => 14 bits |  |  |  |
|  | RSIZE=6 => 15 bits |  |  |  |
|  | RSIZE=7 => 16 bits |  |  |  |

**7.3.3.4.3 Сопоставление адресов при REG\_IBANK\_POS=2 и REG\_EBANK\_POS = 0**

При REG\_IBANK\_POS=2 и REG\_EBANK\_POS = 0 чередование банков в устройстве (на один чип-селект) ограничено 2 банками. Однако между двумя chip-select можно чередовать банки. Таким образом, контроллер DDR2/3/mDDR может держать открытыми максимум 16 банков (восемь внутренних банков между двумя chip-select) одновременно, но может чередовать только четыре из них.

**Таблица 7-213. Сопоставление адресов OCP и адресов DDR2/3/mDDR для REG\_IBANK\_POS=2 и REG\_EBANK\_POS=0**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Address** | | | | |
| **Bank Address[2:1]** | **Row Address** | **Chip Select** | **Bank Address[1:0]** | **Column Address** |
| # Количество битов, определяемое  IBANK из SDRCR | # Количество битов, определяемое  RSIZE из SDRCR | # битов, определяемых  EBANK из SDRCR | # Количество битов, определяемое  IBANK из SDRCR | # Количество битов, определяемое  PAGESIZE из SDRCR |
| IBANK=0 => 0 bits | RSIZE=0 => 9 bits | EBANK=0 => 0 bits | IBANK=0 => 0 bits | PAGESIZE=0 => 8 bits |
| IBANK=1 => 0 bits | RSIZE=1 => 10 bits | EBANK=1 => 1 bit | IBANK=1 => 1 bit | PAGESIZE=1 => 9 bits |
| IBANK=2 => 1 bit | RSIZE=2 => 11 bits |  | IBANK=2 => 1 bit | PAGESIZE=2 => 10 bits |
| IBANK=3 => 2 bits | RSIZE=3 => 12 bits |  | IBANK=3 => 1 bit | PAGESIZE=3 => 11 bits |
|  | RSIZE=4 => 13 bits |  |  |  |
|  | RSIZE=5 => 14 bits |  |  |  |
|  | RSIZE=6 => 15 bits |  |  |  |
|  | RSIZE=7 => 16 bits |  |  |  |

**7.3.3.4.4 Сопоставление адресов при REG\_IBANK\_POS = 3 и REG\_EBANK\_POS = 0**

Если REG\_IBANK\_POS = 3 и REG\_EBANK\_POS = 0, контроллер DDR2/3/mDDR не может чередовать банки в пределах одного устройства (по chip-select). Однако он может чередовать банки между двумя chip-select. Таким образом, контроллер DDR2/3/mDDR может держать открытыми максимум 16 банков (8 внутренних банков между двумя селекторами микросхем) открытыми одновременно, но может перемежаться только между двумя из них.

**Таблица 7-214. Сопоставление адресов OCP и адресов DDR2/3/mDDR для REG\_IBANK\_POS=3 иREG\_EBANK\_POS=0**

|  |  |  |  |
| --- | --- | --- | --- |
| **Logical Address** | | | |
| **Bank Address** | **Row Address** | **Chip Select** | **Column Address** |
| # Количество битов, определяемое  IBANK из SDRCR | # Количество битов, определяемое  RSIZE из SDRCR | # битов, определяемых  EBANK из SDRCR | # Количество битов, определяемое  PAGESIZE из SDRCR |
| IBANK=0 => 0 bits | RSIZE=0 => 9 bits | EBANK=0 => 0 bits | PAGESIZE=0 => 8 bits |
| IBANK=1 => 1 bit | RSIZE=1 => 10 bits | EBANK=1 => 1 bit | PAGESIZE=1 => 9 bits |
| IBANK=2 => 2 bits | RSIZE=2 => 11 bits |  | PAGESIZE=2 => 10 bits |
| IBANK=3 => 3 bits | RSIZE=3 => 12 bits |  | PAGESIZE=3 => 11 bits |
|  | RSIZE=4 => 13 bits |  |  |
|  | RSIZE=5 => 14 bits |  |  |
|  | RSIZE=6 => 15 bits |  |  |
|  | RSIZE=7 => 16 bits |  |  |

**7.3.3.4.5 Сопоставление адресов при REG\_IBANK\_POS = 0 и REG\_EBANK\_POS = 1**

При REG\_IBANK\_POS = 0 и REG\_EBANK\_POS = 1 контроллер памяти DDR2/3/mDDR чередует между всеми банками внутри устройства (по выбору чипа). Однако контроллер памяти DDR2/3/mDDR не может чередовать банки между двумя селекторами микросхем. Таким образом, контроллер памяти DDR2/3/mDDR может одновременно держать открытыми максимум 16 банков (8 внутренних банков на 2 селектора микросхем), но может только перемежаться между 8 из них.

**Таблица 7-215. Сопоставление адресов OCP и адресов DDR2/3/mDDR для REG\_IBANK\_POS=0 и REG\_EBANK\_POS=1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Logical Address** | | | |
| **Chip Select** | **Row Address** | **Bank Address** | **Column Address** |
| # битов, определяемых  EBANK из SDRCR | # Количество битов, определяемое  RSIZE из SDRCR | # Количество битов, определяемое  IBANK из SDRCR | # Количество битов, определяемое  PAGESIZE из SDRCR |
| EBANK=0 => 0 bits | RSIZE=0 => 9 bits | IBANK=0 => 0 bits | PAGESIZE=0 => 8 bits |
| EBANK=1 => 1 bit | RSIZE=1 => 10 bits | IBANK=1 => 1 bit | PAGESIZE=1 => 9 bits |
|  | RSIZE=2 => 11 bits | IBANK=2 => 2 bits | PAGESIZE=2 => 10 bits |
|  | RSIZE=3 => 12 bits | IBANK=3 => 3 bits | PAGESIZE=3 => 11 bits |
|  | RSIZE=4 => 13 bits |  |  |
|  | RSIZE=5 => 14 bits |  |  |
|  | RSIZE=6 => 15 bits |  |  |
|  | RSIZE=7 => 16 bits |  |  |

**7.3.3.4.6 Сопоставление адресов при REG\_IBANK\_POS = 1 и REG\_EBANK\_POS = 1**

При REG\_IBANK\_POS = 1 и REG\_EBANK\_POS = 1 чередование банков внутри устройства (на один чип-селект) ограничено 4 банками. Кроме того, контроллер памяти DDR2/3/mDDR не может чередовать банки между двумя селекторами микросхем. Таким образом, контроллер памяти DDR2/3/mDDR может хранить максимум 16 банков (8 внутренних банков на 2 выбора микросхем) открытыми одновременно, но может чередовать только четыре из них.

**Таблица 7-216. Карта адресов OCP и DDR2/3/mDDR**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Address** | | | | |
| **Chip Select** | **Bank Address[2]** | **Row Address** | **Bank Address[1:0]** | **Column Address** |
| # битов, определяемых  EBANK из SDRCR | # Количество битов, определяемое  IBANK из SDRCR | # Количество битов, определяемое  RSIZE из SDRCR | # Количество битов, определяемое  IBANK из SDRCR | # Количество битов, определяемое  PAGESIZE из SDRCR |
| EBANK=0 => 0 bits | IBANK=0 => 0 bits | RSIZE=0 => 9 bits | IBANK=0 => 0 bits | PAGESIZE=0 => 8 bits |
| EBANK=1 => 1 bit | IBANK=1 => 0 bits | RSIZE=1 => 10 bits | IBANK=1 => 1 bit | PAGESIZE=1 => 9 bits |
|  | IBANK=2 => 0 bits | RSIZE=2 => 11 bits | IBANK=2 => 2 bits | PAGESIZE=2 => 10 bits |
|  | IBANK=3 => 1 bit | RSIZE=3 => 12 bits | IBANK=3 => 2 bits | PAGESIZE=3 => 11 bits |
|  |  | RSIZE=4 => 13 bits |  |  |
|  |  | RSIZE=5 => 14 bits |  |  |
|  |  | RSIZE=6 => 15 bits |  |  |
|  |  | RSIZE=7 => 16 bits |  |  |

**7.3.3.4.7 Сопоставление адресов при REG\_IBANK\_POS = 2 и REG\_EBANK\_POS = 1**

При REG\_IBANK\_POS = 2 и REG\_EBANK\_POS = 1 чередование банков внутри устройства (на один чип-селект) ограничено 2 банками. Кроме того, контроллер памяти DDR2/3/mDDR не может чередовать банки между двумя селекторами микросхем. Таким образом, контроллер памяти DDR2/3/mDDR может хранить максимум 16 банков (8 внутренних банков на 2 выбора микросхем) открытыми одновременно, но может чередовать только два из них.

**Таблица 7-217. Сопоставление адресов OCP и адресов DDR2/3/mDDR для REG\_IBANK\_POS = 2 и REG\_EBANK\_POS = 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Address** | | | | |
| **Chip Select** | **Bank Address[2:1]** | **Row Address** | **Bank Address[0]** | **Column Address** |
| # битов, определяемых  EBANK из SDRCR | # Количество битов, определяемое  IBANK из SDRCR | # Количество битов, определяемое  RSIZE из SDRCR | # Количество битов, определяемое  IBANK из SDRCR | # Количество битов, определяемое  PAGESIZE из SDRCR |
| EBANK=0 => 0 bits | IBANK=0 => 0 bits | RSIZE=0 => 9 bits | IBANK=0 => 0 bits | PAGESIZE=0 => 8 bits |
| EBANK=1 => 1 bit | IBANK=1 => 0 bits | RSIZE=1 => 10 bits | IBANK=1 => 1 bit | PAGESIZE=1 => 9 bits |
|  | IBANK=2 => 1 bits | RSIZE=2 => 11 bits | IBANK=2 => 1 bits | PAGESIZE=2 => 10 bits |
|  | IBANK=3 => 2 bit | RSIZE=3 => 12 bits | IBANK=3 => 1 bits | PAGESIZE=3 => 11 bits |
|  |  | RSIZE=4 => 13 bits |  |  |
|  |  | RSIZE=5 => 14 bits |  |  |
|  |  | RSIZE=6 => 15 bits |  |  |
|  |  | RSIZE=7 => 16 bits |  |  |

**7.3.3.4.8 Сопоставление адресов при REG\_IBANK\_POS = 3 и REG\_EBANK\_POS = 1**

При REG\_IBANK\_POS = 3 и REG\_EBANK\_POS = 1 контроллер памяти DDR2/3/mDDR не может перемежать банки внутри устройства (на один выбор микросхемы) или между двумя выборами микросхем. Таким образом, контроллер памяти DDR2/3/mDDR может держать открытыми максимум 16 банков (8 внутренних банков на два селектора микросхем) открытыми одновременно, но не может чередовать их между собой.

**Таблица 7-218. Сопоставление адресов OCP и адресов DDR2/3/mDDR для REG\_IBANK\_POS=3 и REG\_EBANK\_POS=1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Logical Address** | | | |
| **Chip Select** | **Bank Address** | **Row Address** | **Column Address** |
| # битов, определяемых  EBANK из SDRCR | # Количество битов, определяемое  IBANK из SDRCR | # Количество битов, определяемое  RSIZE из SDRCR | # Количество битов, определяемое  PAGESIZE из SDRCR |
| EBANK=0 => 0 bits | IBANK=0 => 0 bits | RSIZE=0 => 9 bits | PAGESIZE=0 => 8 bits |
| EBANK=1 => 1 bit | IBANK=1 => 1 bits | RSIZE=1 => 10 bits | PAGESIZE=1 => 9 bits |
|  | IBANK=2 => 2 bits | RSIZE=2 => 11 bits | PAGESIZE=2 => 10 bits |
|  | IBANK=3 => 3 bit | RSIZE=3 => 12 bits | PAGESIZE=3 => 11 bits |
|  |  | RSIZE=4 => 13 bits |  |
|  |  | RSIZE=5 => 14 bits |  |
|  |  | RSIZE=6 => 15 bits |  |
|  |  | RSIZE=7 => 16 bits |  |

Поскольку контроллер памяти DDR2/3/mDDR чередует меньшее количество банков, когда

IBANK\_POS!= 0 или EBANK\_POS= 1, производительность в этих случаях ниже, чем в случае IBANK\_POS= 0. Таким образом, эти случаи рекомендуется использовать только вместе с частичным самообновлением массива, когда производительность может быть компенсирована экономией энергии.

**7.3.3.5 Управление производительностью**

**7.3.3.5.1 Упорядочивание и планирование команд**

Контроллер памяти DDR2/3/mDDR выполняет переупорядочивание и планирование команд в попытке добиться эффективной передачи данных с максимальной пропускной способностью. Цель состоит в том, чтобы максимально использовать шину данных, адресную и командную шину, скрывая при этом накладные расходы на открытие и закрытие DDR2/3/mDDR SDRAM строк. Переупорядочивание команд происходит внутри командного FIFO.

Контроллер памяти DDR2/3/mDDR просматривает все команды, хранящиеся в командном FIFO, чтобы планирования команд во внешнюю память. Все команды от одного и того же мастера будут выполняться по порядку, независимо от приоритета мастера. Контроллер памяти не гарантирует упорядочивания между командами от разных мастеров. Однако контроллер памяти будет поддерживать согласованность данных. Поэтому контроллер памяти будет блокировать команду, независимо от приоритета мастера, если эта команда относится к тому же адресу блока (2048 байт), что и более старая команда. Таким образом, контроллер памяти может иметь одну или несколько ожидающих команд чтения или записи для каждого мастера. Среди всех ожидающих чтений контроллер памяти выбирает все чтения у которых уже открыты соответствующие банки SDRAM. Аналогично, среди всех ожидающих записи контроллер памяти выбирает все записи, соответствующие банки SDRAM которых уже открыты.

В результате вышеупомянутого упорядочивания в любой момент времени контроллер памяти может иметь несколько ожидающих операций чтений и записей, у которых открыты соответствующие банки. Затем контроллер памяти выбирает чтение с наивысшим приоритетом из ожидающих чтения, а запись с наивысшим приоритетом из ожидающих записи. Если две или более команд имеют наивысший приоритет, контроллер памяти выбирает самую старую команду. В результате у контроллера памяти может быть последняя команда чтения и последняя команда записи. Контроллер памяти выберет либо чтение, либо запись в зависимости от значения, запрограммированного в регистре Read Write Execution Threshold register. Контроллер памяти будет выполнять чтение до тех пор, пока не будет достигнут порог чтения, а затем затем переключится на выполнение записи. Затем контроллер памяти будет выполнять записи до тех пор, пока не будет достигнут порог записи, а затем снова переключится на выполнение чтения. Контроллер памяти будет удовлетворять требованиям пороговых значений только в том случае, если данный тип команды доступен для выполнения, в противном случае он переключится на другой тип. Аналогично, контроллер памяти будет удовлетворять пороговым значениям только в том случае, если в FIFO данного типа есть место (Read Data FIFO для чтения и Write Status FIFO для записи), иначе он переключится на другой тип.

Контроллер памяти завершает выполнение команды OCP, прежде чем переключиться на другую команду.

Все доступы к SDRAM объединяются в трубопровод для максимального использования внешней шины. Другими словами доступы к SDRAM выдаются один за другим таким образом, чтобы между двумя доступами было минимальное количество циклов простоя. Это включает в себя планирование, перечисленное выше, чтобы минимизировать накладные расходы на открытие и закрытие банков SDRAM. Все это выполняется при соблюдении временных требований доступа к SDRAM.

Помимо команд, получаемых от внутрикристальных ресурсов, контроллер памяти DDR2/3/mDDR также выдает команды обновления. Контроллер памяти DDR2/3/mDDR пытается задержать команды обновления как можно дольше, чтобы обеспечить максимальную производительность при соблюдении требований к обновлению SDRAM. По мере того как контроллер памяти DDR2/3/mDDR подает команды чтения, записи и обновления на устройство DDR2/3/mDDR SDRAM, он следует следующей схеме приоритетов:

1. (Наивысший приоритет) Запрос на обновление SDRAM из-за достижения уровня срочности обновления Refresh Must (см.Раздел 7.3.3.5.5).

2. Запрос на чтение или запись.

3. Команды активации SDRAM.

4. Команды деактивации SDRAM.

5. Запрос на глубокое отключение питания SDRAM.

6. Остановка тактового генератора SDRAM или запрос Power-Down.

7. Запрос на обновление SDRAM в связи с достижением уровня срочности обновления Refresh May или Release (см.раздел Планирование обновлений).

8. (Самый низкий приоритет) Запрос на самообновление SDRAM.

**7.3.3.5.2 Остановка команд**

Перечисленные выше правила переупорядочивания и планирования могут привести к «голоданию» команд, т. е. предотвращение обработки определенных команд контроллером памяти DDR2/3/mDDR.

Голодание команд возникает при следующих условиях:

- непрерывный поток высокоприоритетных команд чтения может блокировать низкоприоритетную команду записи

- Непрерывный поток команд DDR2/3/mDDR SDRAM на строку в открытом банке может блокировать команды на закрытый ряд в том же банке.

Чтобы избежать таких ситуаций, контроллер памяти DDR2/3/mDDR может на мгновение повысить приоритет самой старой команды в FIFO команд после выполнения определенного количества пересылок. Адрес REG\_COS\_COUNT\_1,REG\_COS\_COUNT\_2 в регистре конфигурации интерфейса (OCP\_CONFIG) задает количество пересылок, которые должны быть выполнены, прежде чем контроллер памяти DDR2/3/mDDR повысит приоритет самой старой команды. Дополнительные сведения см. в разделе Класс обслуживания (COS).

*ПРИМЕЧАНИЕ: Если оставить биты REG\_COS в значении по умолчанию (FFh) в регистре конфигурации интерфейса (OCP\_CONFIG) это отключает эту функцию контроллера памяти DDR2/3/mDDR. Это означает, что команды могут оставаться в командном FIFO неограниченно долго. Поэтому эти биты должны быть установлены в значение FEh сразу после сброса, чтобы включить эту функцию с максимальным уровнем допустимых передач данных в память. Рекомендуется установить приоритет на уровне системы, чтобы избежать размещения высокоскоростных мастеров на самые высокие уровни приоритета. Эти биты можно оставить как FEh, если только не требуется расширенное управление пропускной способностью/приоритетами.*

**7.3.3.5.3 Возможные условия гонки**

Когда некоторые мастера записывают данные в контроллер памяти DDR2/3/mDDR, может возникнуть ситуация гонки. Например, если мастер A передает программное сообщение через буфер в памяти DDR2/3/mDDR и не ждет сигнала о завершении записи, то при попытке мастера B прочитать программное сообщение он может прочитать устаревшие данные и, следовательно, получить неверное сообщение. Для подтверждения того, что запись от мастера A завершилась до того, как будет выполнено чтение от мастера B, мастер A должен дождаться статуса завершения записи от контроллера памяти DDR2/3/mDDR, прежде чем сообщить ведущему B, что данные готовы к считыванию.

Если мастер A не дожидается сообщения о завершении записи, он должен выполнить следующее обходное решение:

1. Выполните требуемую запись.

2. Выполните фиктивную запись в регистр идентификатора и ревизии модуля контроллера памяти DDR2.

3. Выполните фиктивное чтение регистра идентификатора и ревизии модуля контроллера памяти DDR2.

4. После завершения чтения на шаге 3 сообщите ведущему B, что данные готовы к считыванию.

Завершение чтения на шаге 3 гарантирует, что предыдущая запись была выполнена.

Список периферийных устройств ведущего устройства, для которых требуется данное обходное решение, см. в спецификации конкретного устройства.

**7.3.3.5.4 Класс обслуживания (COS)**

Команды в командном FIFO могут быть сопоставлены с двумя классами обслуживания, а именно 1 и 2. Отнесение команд к определенному классу обслуживания может быть выполнено на основе приоритета или идентификатора соединения ID.

Сопоставление на основе приоритета может быть выполнено путем установки соответствующих значений в регистре приоритета и класса обслуживания (PRI\_COS\_MAP). Сопоставление на основе ID соединения может быть выполнено путем установки соответствующих значений ID соединения и масок в регистрах Connection ID to Class of Service Mapping (CONNID\_COS\_1\_MAP и CONNID\_COS\_2\_MAP).

Для каждого класса обслуживания можно установить 3 значения идентификатора соединения и маски. В сочетании с масками, каждый класс обслуживания может иметь максимум 144 идентификатора соединения, сопоставленных с ним. Например, значение идентификатора соединения 0xFF вместе со значением маски 0x3 сопоставит все идентификаторы соединений от 0xF8 до 0xFF к данному классу обслуживания.

Каждый класс обслуживания имеет ассоциированный счетчик задержки (REG\_COS\_COUNT). Значение этого счетчика может быть установлено в регистре конфигурации интерфейса (OCP\_CONFIG). Когда счетчик задержки для команды истекает, т.е. достигает значения, запрограммированного для класса обслуживания, к которому принадлежит команда ,эта команда будет выполняться следующей. Если существует несколько команд, с истекшим сроком ожидания, команда с наивысшим приоритетом будет выполнена первой. Одно исключение если самая старая команда в очереди имеет истекший счетчик reg\_pr\_old\_count, то эта команда будет выполнена первой независимо от приоритета или класса обслуживания. Это сделано для предотвращения эффекта непрерывного блокирования.

Сопоставление идентификаторов соединений позволяет поместить один и тот же идентификатор соединения как в класс обслуживания 1, так и в класс 2. Кроме того, транзакция может принадлежать к одному классу обслуживания, если рассматривать ее по идентификатору соединения, и может принадлежать к другому классу обслуживания, если рассматривать ее по приоритету. В этих случаях команда будет принадлежать к обоим классам обслуживания. Контроллер памяти DDR2/3/mDDR попытается выполнить команду как можно быстрее, когда истечет меньший из двух счетчиков ( REG\_COS\_COUNT\_1 ИЛИ REG\_COS\_COUNT\_2).

**7.3.3.5.5 Планирование обновлений**

Контроллер памяти DDR2/3/mDDR выдает команды автообновления (REFR) на устройства DDR2/3/mDDR SDRAM с частотой, определяемой битовым полем частоты обновления (REFRESH\_RATE) в регистре управления обновлением SDRAM (SDRFC). Счетчик интервалов обновления загружается значением битового поля REFRESH\_RATE и уменьшается на 1 каждый цикл, пока не достигнет нуля. Как только счетчик интервалов достигает нуля, он перезагружается со значением значением бита REFRESH\_RATE. Каждый раз, когда счетчик интервалов истекает, счетчик отставания обновления увеличивается на 1. И наоборот, каждый раз, когда контроллер памяти DDR2/3/mDDR выполняет команду REFR счетчик бэклога уменьшается на 1. Это означает, что счетчик бэклога обновления фиксирует количество команд REFR, которые контроллер памяти DDR2/3/mDDR в данный момент не выполнил.

Контроллер памяти DDR2/3/mDDR выдает команды REFR в зависимости от уровня срочности. Уровень срочности определен ниже. При достижении уровня срочности обновления контроллер памяти DDR2/3/mDDR выдает команду REFR перед обслуживанием любых новых запросов доступа к памяти.

После команды REFR контроллер памяти DDR2/3/mDDR ожидает T\_RFC циклов, определенных в регистре SDRAM timing 1 register (SDRTIM1), перед повторной проверкой уровня срочности обновления. Счетчики обновления не работают, когда память SDRAM находится в режиме самообновления.

**Таблица 7-219. Режимы обновления**

|  |  |
| --- | --- |
| **Urgency Level** | **Description** |
| Refresh May | Счетчик отставания больше 0. Указывает на отставание команд REFR, когда контроллер памяти DDR2/3/mDDR не занят, он выдаст команду REFR. |
| Refresh Release | Счетчик отставания больше 4. Указывает на то, что отставание в обновлении команд REFR становится большим и когда контроллер памяти DDR2/3/mDDR не занят, он должен выдать команду REFR. |
| Refresh Must | Счетчик отставания больше 7. Указывает на то, что отставание в обновлении команд REFR становится чрезмерным и контроллеру памяти DDR2/3/mDDR следует выполнить цикл автоматического обновления перед обслуживанием новых  запросов доступа к памяти. |

Контроллер памяти DDR2/3/mDDR начинает обслуживать новые обращения к памяти после сброса уровня Refresh Release. Если какая-либо из команд в Command FIFO имеет счетчики задержки класса обслуживания, которые истекли, контроллер памяти DDR2/3/mDDR не будет дожидаться сброса уровня Refresh Release, а выполнит только одну команду обновления и выйдет из состояния обновления.

**7.3.3.5.6 Конфигурация счетчиков производительности**

В таблице 7-220 показаны возможные конфигурации фильтров для двух счетчиков производительности (REG\_CNTR1\_CFG и REG\_CNTR2\_CFG). Эти конфигурации фильтров можно использовать в сочетании с подключением OCP ID и/или внешним селектором микросхемы для получения статистики производительности для конкретного мастера OCP и/или внешнего селектора микросхемы.

**Таблица 7-220. Конфигурации фильтров для счетчиков производительности**

|  |  |  |  |
| --- | --- | --- | --- |
| **cntrN\_cfg(1)** | **cntrN\_region\_en** | **cntrN\_mconnid\_en** | **Description** |
| 0x0 | 0x0 | 0x0 or 0x1 | Подсчет общего количества обращений к SDRAM. |
| 0x1 | 0x0 | 0x0 or 0x1 | Подсчет общего количества активаций SDRAM. |
| 0x2 | 0x0 or 0x1 | 0x0 or 0x1 | Подсчет общего количества прочтений. |
| 0x3 | 0x0 or 0x1 | 0x0 or 0x1 | Подсчет общего количества записей. |
| 0x4 | 0x0 | 0x0 | Подсчет количества тактовых циклов DDR OCP. Командный FIFO заполнен. |
| 0x5 | 0x0 | 0x0 | Подсчет количества тактовых циклов DDR OCP.FIFO данных для записи заполнен. |
| 0x6 | 0x0 | 0x0 | Подсчет количества тактовых циклов DDR OCP.FIFO данных для чтения заполнен. |
| 0x7 | 0x0 | 0x0 | Подсчет количества тактовых циклов DDR OCP.FIFO команд возврата заполнен. |
| 0x8 | 0x0 or 0x1 | 0x0 or 0x1 | Подсчет количества приоритетных подъемов. |
| 0x9 | 0x0 | 0x0 | Подсчитывает количество тактов DDR, в течение которых команда была в ожидании. |
| 0xA | 0x0 | 0x0 | Подсчет количества тактов DDR, в течение которых шина данных памяти  передавала данные. |
| 0xB – 0xF | 0x0 | 0x0 | Зарезервировано для будущего использования. |

***(1)*** *Если MReqDebug установлен на 1 для определенной команды OCP, счетчики производительности не будут увеличиваться если значения cntrN\_cfg равны 0x0, 0x1, 0x2, 0x3 или 0xA.*

**7.3.3.6 Инициализация SDRAM**

**7.3.3.6.1 Инициализация DDR2**

При выходе из сброса, если поле reg\_sdram\_type в регистре SDRAM Config равно 2 и reg\_initref\_dis в регистре SDRAM Config равно 2 и бит SDRAM Refresh Control установлен в 0, EMIF выполняет последовательность инициализации DDR2 SDRAM следующим образом:

1. Подает низкий уровень на pad\_cke\_o.

2. Через 16 интервалов частоты обновления SDRAM выдает команду NOP, удерживая pad\_cke\_o на высоком уровне. Частота обновления SDRAM определяется в описании поля reg\_refresh\_rate (см. описание регистра SDRAM Refresh Control register).

3. Выдает команду PRECHARGE с высоким уровнем pad\_a\_o[10] для указания всех банков.

4. Выдает команду LOAD MODE REGISTER в регистр расширенного режима 2 (pad\_ba\_o[2:0] = 0x2) при этом pad\_a\_o[15:0] устанавливается следующим образом: