

9 ПРЯМОЙ ДОСТУП К ПАМЯТИ

Процессор использует прямой доступ к памяти (Direct Memory Access, DMA) для передачи данных между пространствами памяти или между пространством памяти и периферийным устройством. Процессор может задать параметры операций передачи данных и вернуться к выполнению основной программы, при этом передача данных выполняется независимо от работы процессора полностью интегрированным контроллером DMA.

Контроллер DMA может выполнять несколько типов передач данных:

- передачи между пространствами памяти (MDMA) (см. раздел “DMA типа “память-память””);
- передачи между памятью и последовательным периферийным интерфейсом (SPI) (см. главу 10, “Контроллеры SPI-совместимого порта”);
- передачи между памятью и последовательным портом (SPORT) (см. главу 12, “Контроллеры последовательных портов”);
- передачи между памятью и портом UART (см. главу 13, “Контроллер порта UART”);
- передачи между памятью и параллельным периферийным интерфейсом (PPI) (см. главу 11, “Параллельный периферийный интерфейс”);

Система содержит шесть периферийных устройств, поддерживающих DMA, включая контроллер DMA типа “память-память” (MDMA). Эти устройства поддерживаются следующими двенадцатью каналами DMA:

- контроллер DMA приёма/передачи PPI;
- контроллер DMA приёма SPORT0;
- контроллер DMA передачи SPORT0;
- контроллер DMA приёма SPORT1;
- контроллер DMA передачи SPORT1;
- контроллер DMA приёма/передачи SPI;
- контроллер DMA приёма UART;
- контроллер DMA передачи UART;
- канал передачи (приёмник) потока 1 MDMA;
- канал приёма (источник) потока 1 MDMA;
- канал передачи (приёмник) потока 0 MDMA;
- канал приёма (источник) потока 0 MDMA;

В данной главе описываются свойства, являющиеся общими для всех каналов DMA, например, настройка параметров DMA. Дополнительную информацию о свойствах, характерных для определённого периферийного устройства, см. в главах, описывающих данные устройства. Описание производительности и арбитража шин при выполнении операций DMA можно найти в разделе “Производительность DAB, DCB и DEB” главы 7.

Прямой доступ к памяти

Передачи данных в режиме DMA могут быть основаны на использовании дескрипторов или на использовании регистров. Для инициации последовательности передач в режиме DMA с использованием дескрипторов требуется наличие набора параметров, хранящегося в памяти. Этот тип передач позволяет выполнять связанные цепочки передач DMA. В операциях DMA с использованием дескрипторов канал DMA может быть запрограммирован на автоматическую настройку и начало другой передачи в режиме DMA по завершении текущей передачи. При выполнении операций DMA с использованием регистров процессор может напрямую программировать регистры управления DMA для инициации передач. При необходимости, по завершении передачи значение регистров управления может автоматически обновляться значениями начальной инициализации для обеспечения непрерывности передач.

Регистры DMA и DMA типа “память-память”


Для удобства в этой главе используются общие (не зависящие от конкретного периферийного устройства) названия регистров DMA и DMA типа “память-память”.

- Список общих имён регистров DMA приведён в таблице 9-1.
- Список общих имён регистров DMA типа “память-память” приведён в таблице 9-3.

Регистры DMA можно разделить на три категории:

- Регистры параметров; например, `DMAx_CONFIG` и `DMAx_COUNT`

Значения элементов дескрипторов могут загружаться напрямую только в регистры параметров; элементы дескрипторов перечислены в таблице 9-2, “Правила обозначения: регистры DMA, отображённые в карте памяти, и элементы дескрипторов”.

 Символ “x” в обозначении `DMAx` соответствует определённому периферийному устройству, поддерживающему DMA. Например, при назначении каналов DMA по умолчанию `DMA6_CONFIG` соответствует регистру `DMA_CONFIG` каналу приёма UART. Назначение каналов DMA по умолчанию см. в таблице 9-16.

- Регистры текущего состояния; например, `DMAx_CURR_ADDR` и `DMAx_CURR_X_COUNT`.
- Регистры управления/состояния; например, `DMAx_IRQ_STATUS` и `DMAx_PERIPHERAL_MAP`.

В таблице 9-1 перечислены общие названия регистров DMA. Для каждого регистра в таблице также показано смещение адреса регистра в карте памяти, относительного базового, краткое описание регистра, его категория и значение после сброса.

Прямой доступ к памяти

Таблица 9-1. Общие названия регистров DMA, отображённых в карте памяти

Смещение регистра	Общее название регистра	Описание регистра	Категория регистра
0x00	NEXT_DESC_PTR	Указатель следующего дескриптора	Регистр параметров
0x04	START_ADDR	Начальный адрес текущего буфера	Регистр параметров
0x08	DMA_CONFIG	Регистр конфигурации DMA, включает бит разрешения	Регистр параметров
0x0C	Зарезервировано	Зарезервировано	
0x10	X_COUNT	Счётчик внутреннего цикла	Регистр параметров
0x14	X_MODIFY	Инкремент адреса внутреннего цикла, в байтах	Регистр параметров
0x18	Y_COUNT	Счётчик внешнего цикла (только для двумерного DMA)	Регистр параметров
0x1C	Y_MODIFY	Инкремент адреса внешнего цикла, в байтах	Регистр параметров
0x20	CURR_DESC_PTR	Указатель текущего дескриптора	Регистр текущего состояния
0x24	CURR_ADDR	Текущий адрес DMA	Регистр текущего состояния
0x28	IRQ_STATUS	Регистр состояния прерываний: Содержит состояние прерываний завершения и ошибки DMA и состояние канала (работа/выборка/пауза)	Регистр управления/состояния
0x2C	PERIPHERAL_MAP	Определяет соответствие каналов DMA периферийным устройствам: Содержит 4-разрядное значение, определяющее периферийное устройство, связанное с каналом DMA (регистры каналов MDMA доступны только для чтения)	Регистр управления/состояния
0x30	CURR_X_COUNT	Текущее состояние счётчика (при одномерном DMA) или состояние внутрисканового счётчика X (при двумерном DMA); выполняется обратный счёт от значения X_COUNT	Регистр текущего состояния
0x34	Зарезервировано	Зарезервировано	
0x38	CURR_Y_COUNT	Текущее состояние счётчика строк (только для двумерного DMA); выполняется обратный счёт от значения Y_COUNT	Регистр текущего состояния
0x3C	Зарезервировано	Зарезервировано	

Обращение ко всем регистрам DMA может выполняться как к 16-разрядным, за исключением следующих регистров, обращение к которым также может выполняться как к 32-разрядным:

- NEXT_DESC_PTR
- START_ADDR
- CURR_DESC_PTR
- CURR_ADDR




Когда к этим регистрам осуществляется обращение как к 16-разрядным, доступны только младшие 16 битов.

Прямой доступ к памяти

Правила обозначения регистров DMA, отображённых в карте памяти

Чтобы избежать путаницы в названиях элементов дескрипторов и общих названиях регистров DMA, в данной главе используются правила обозначения, приведённые в таблице 9-2, где:

- В левом столбце перечислены общие названия регистров, отображённых в карте памяти, которые используются при обсуждении общих принципов DMA.

 Следует отметить, что общие названия, приведённые в левом столбце, не совпадают с действительными названиями, соответствующими конкретным ресурсам процессора.

- В среднем столбце перечислены специфические названия регистров, отображённых в карте памяти. Эти названия соответствуют конкретным ресурсам процессора.

В записи DMA_x символ “x” соответствует номеру канала DMA. Например, название DMA₃_IRQ_STATUS соответствует регистру IRQ_STATUS канала DMA #3.

Номер канала может быть назначен по умолчанию или задан программно. Номера каналов DMA и соответствующие им по умолчанию периферийные устройства перечислены в таблице 9-16.

- В последнем столбце перечислены макроопределения, соответствующие каждому из элементов дескрипторов в памяти.

Макроопределения, приведённые в последнем столбце, используются только для простоты описания работы системы DMA.

Таблица 9-2. Правила обозначения: регистры DMA, отображённые в карте памяти, и элементы дескрипторов

Общее название регистра, отображённого в карте памяти	Специфическое название регистра, отображённого в карте памяти (x = номер канала DMA)	Название соответствующего элемента дескриптора в памяти
DMA_CONFIG	DMA _x _CONFIG	DMACFG
NEXT_DESC_PTR	DMA _x _NEXT_DESC_PTR	NDPH (старшие 16 битов) NDPL (младшие 16 битов)
START_ADDR	DMA _x _START_ADDR	SAH (старшие 16 битов) SAL (младшие 16 битов)
X_COUNT	DMA _x _X_COUNT	XCNT
Y_COUNT	DMA _x _Y_COUNT	YCNT
X_MODIFY	DMA _x _X_MODIFY	XMOD
Y_MODIFY	DMA _x _Y_MODIFY	YMOD
CURR_DESC_PTR	DMA _x _CURR_DESC_PTR	Отсутствует
CURR_ADDR	DMA _x _CURR_ADDR	Отсутствует
CURR_X_COUNT	DMA _x _CURR_X_COUNT	Отсутствует
CURR_Y_COUNT	DMA _x _CURR_Y_COUNT	Отсутствует
IRQ_STATUS	DMA _x _IRQ_STATUS	Отсутствует
PERIPHERAL_MAP	DMA _x _PERIPHERAL_MAP	Отсутствует

Правила обозначения регистров DMA типа “память-память”

Названия регистров DMA типа “память-память” несколько отличаются от названий других регистров DMA. Потокам DMA типа “память-память” не может быть назначен другой канал, в то время как периферийному устройству, поддерживающему DMA, может быть поставлен в соответствие любой канал DMA от 0 до 7.

В таблице 9-3 приведены правила обозначения регистров DMA типа “память-память”. В каждом названии символы “yy” могут принимать четыре различных значения:

- S0, источник потока 0 DMA типа “память-память”;
- D0, приёмник потока 0 DMA типа “память-память”;
- S1, источник потока 1 DMA типа “память-память”;
- D1, приёмник потока 1 DMA типа “память-память”.

Таблица 9-3. Правила обозначения регистров DMA типа “память-память”

Общее название регистра, отображённого в карте памяти	Специфическое название регистра DMA типа “память-память”, отображённого в карте памяти (yy = S0, S1, D0 или D1)	Название соответствующего элемента дескриптора в памяти
DMA CONFIG	MDMA yy CONFIG	DMACFG
NEXT_DESC_PTR	MDMA yy NEXT_DESC_PTR	NDPH (старшие 16 битов) NDPL (младшие 16 битов)
START_ADDR	MDMA yy START_ADDR	SAH (старшие 16 битов) SAL (младшие 16 битов)
X COUNT	MDMA yy X COUNT	XCNT
Y COUNT	MDMA yy Y COUNT	YCNT
X MODIFY	MDMA yy X MODIFY	XMOD
Y MODIFY	MDMA yy Y MODIFY	YMOD
CURR_DESC_PTR	MDMA yy CURR_DESC_PTR	Отсутствует
CURR_ADDR	MDMA yy CURR_ADDR	Отсутствует
CURR_X COUNT	MDMA yy CURR_X COUNT	Отсутствует
CURR_Y COUNT	MDMA yy CURR_Y COUNT	Отсутствует
IRQ STATUS	MDMA yy IRQ STATUS	Отсутствует
PERIPHERAL_MAP	MDMA yy PERIPHERAL_MAP	Отсутствует

Регистр указателя следующего дескриптора (DMAx_NEXT_DESC_PTR/ MDMA_yy_NEXT_DESC_PTR)

Регистр NEXT_DESC_PTR определяет ячейку памяти, в которой будет осуществлён поиск начала следующего блока дескрипторов по окончании операции DMA, определяемой текущим блоком дескрипторов. Этот регистр используется в режимах больших и малых списков дескрипторов. В любом из этих режимов в начале выборки дескриптора содержимое 32-разрядного регистра NEXT_DESC_PTR копируется в регистр CURR_DESC_PTR. Затем, во время выборки дескриптора, после чтения каждого элемента содержимое регистра CURR_DESC_PTR инкрементируется.

Прямой доступ к памяти

i Перед началом операции DMA в режимах больших и малых списков дескрипторов должна выполняться прямая запись по адресу регистра NEXT_DESC_PTR, а не регистра CURR_DESC_PTR.

В режиме массива дескрипторов регистр указателя следующего дескриптора игнорируется, и выборка управляется только регистром CURR_DESC_PTR.

Регистр указателя следующего дескриптора (DMAx_NEXT_DESC_PTR/ MDMA_yy_NEXT_DESC_PTR)

Перед разрешением канала доступен для чтения и записи; после разрешения канала доступен только для чтения.

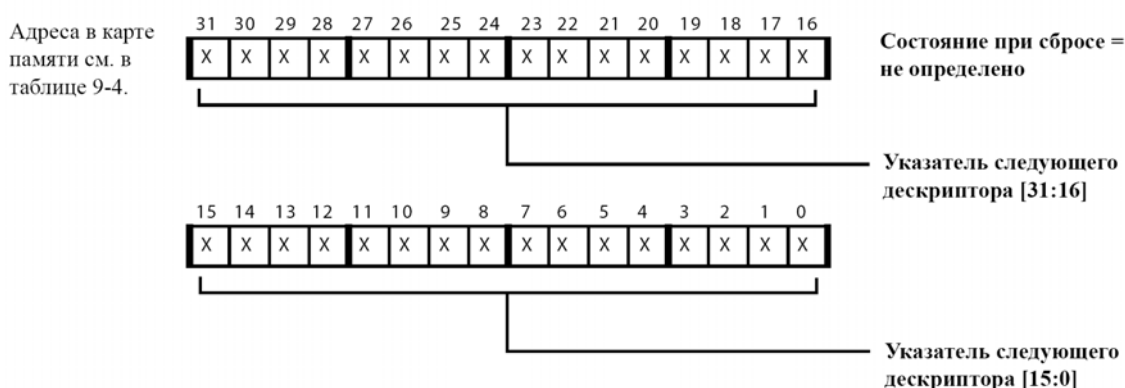


Рис. 9-1. Регистр указателя следующего дескриптора

Таблица 9-4. Адреса регистров указателя следующего дескриптора в карте памяти

Название регистра	Адрес в карте памяти
DMA0 NEXT_DESC_PTR	0xFFC0 0C00
DMA1 NEXT_DESC_PTR	0xFFC0 0C40
DMA2 NEXT_DESC_PTR	0xFFC0 0C80
DMA3 NEXT_DESC_PTR	0xFFC0 0CC0
DMA4 NEXT_DESC_PTR	0xFFC0 0D00
DMA5 NEXT_DESC_PTR	0xFFC0 0D40
DMA6 NEXT_DESC_PTR	0xFFC0 0D80
DMA7 NEXT_DESC_PTR	0xFFC0 0DC0
MDMA_D0 NEXT_DESC_PTR	0xFFC0 0E00
MDMA_S0 NEXT_DESC_PTR	0xFFC0 0E40
MDMA_D1 NEXT_DESC_PTR	0xFFC0 0E80
MDMA_S1 NEXT_DESC_PTR	0xFFC0 0EC0

Регистр начального адреса (DMAx_START_ADDR/ MDMA_yy_START_ADDR)

Регистр START_ADDR, показанный на рис. 9-2, содержит начальный адрес буфера данных, используемого в текущей операции DMA.

Прямой доступ к памяти

Регистр начального адреса (DMAx_START_ADDR/MDMA_yy_START_ADDR)

Перед разрешением канала доступен для чтения и записи; после разрешения канала доступен только для чтения.

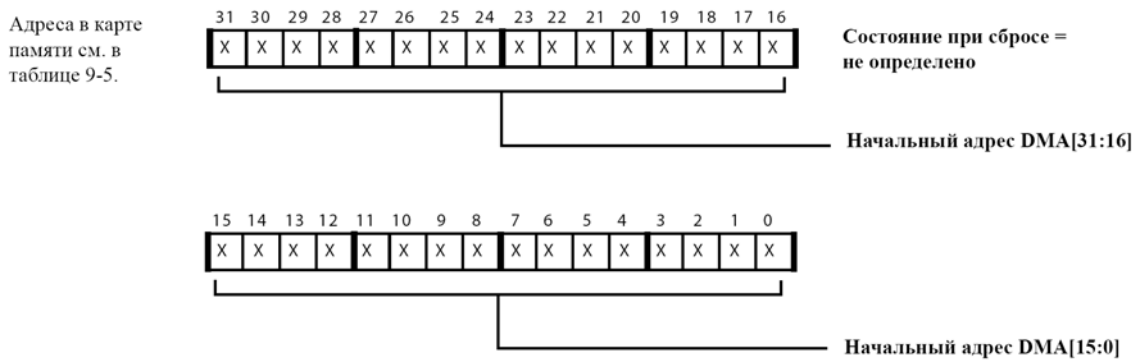


Рис. 9-2. Регистр начального адреса

Таблица 9-5. Адреса регистров начального адреса в карте памяти

Название регистра	Адрес в карте памяти
DMA0_START_ADDR	0xFFC 0C04
DMA1_START_ADDR	0xFFC 0C44
DMA2_START_ADDR	0xFFC 0C84
DMA3_START_ADDR	0xFFC 0CC4
DMA4_START_ADDR	0xFFC 0D04
DMA5_START_ADDR	0xFFC 0D44
DMA6_START_ADDR	0xFFC 0D84
DMA7_START_ADDR	0xFFC 0DC4
MDMA_D0_START_ADDR	0xFFC 0E04
MDMA_S0_START_ADDR	0xFFC 0E44
MDMA_D1_START_ADDR	0xFFC 0E84
MDMA_S1_START_ADDR	0xFFC 0EC4

Регистр конфигурации DMA (DMAx_CONFIG/MDMA_yy_CONFIG)

Регистр DMA_CONFIG, показанный на рис. 9-3, используется для настройки параметров и рабочих режимов DMA. Необходимо отметить, что запись в регистр DMA_CONFIG во время выполнения операции DMA вызывает ошибку DMA, за исключением случая, когда при записи бит DMA_EN устанавливается в 0.

Прямой доступ к памяти

Регистр конфигурации (DMAx_CONFIG/MDMA_yy_CONFIG)

Перед разрешением канала доступен для чтения и записи; после разрешения канала доступен только для чтения.

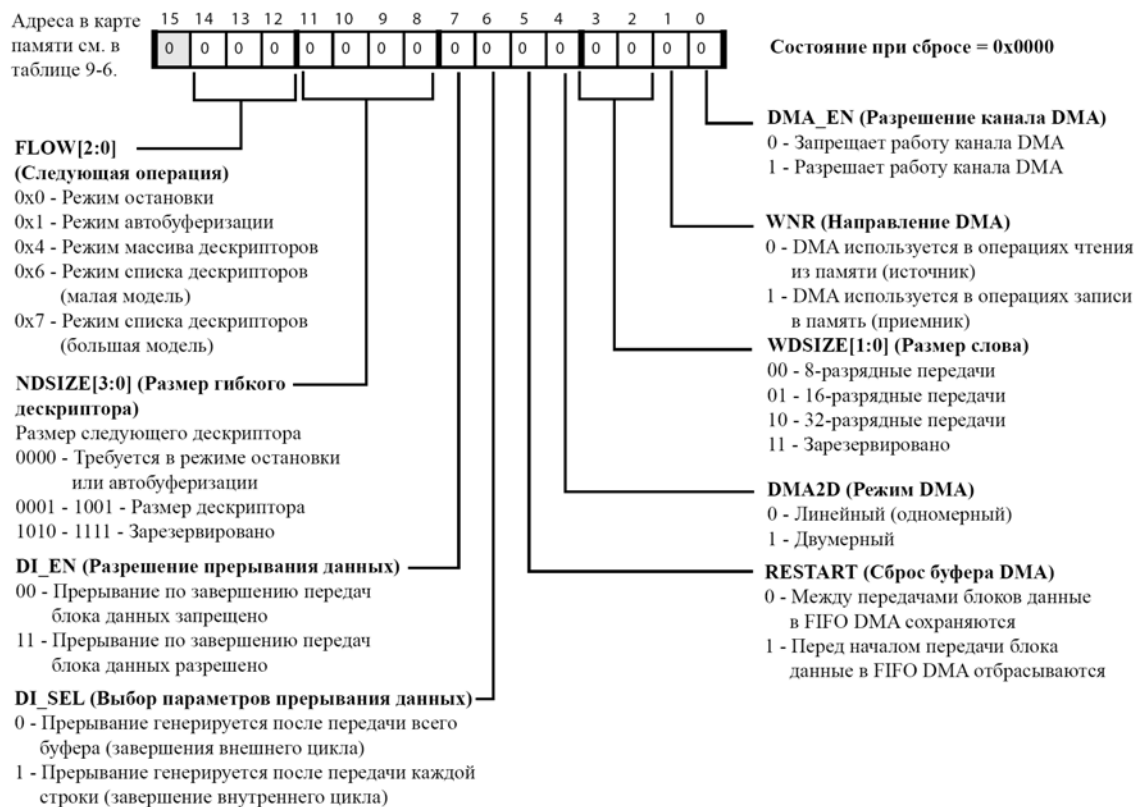


Рис. 9-3. Регистр конфигурации

Таблица 9-6. Адреса регистров конфигурации в карте памяти

Название регистра	Адрес в карте памяти
DMA0_CONFIG	0xFFC0 0C08
DMA1_CONFIG	0xFFC0 0C48
DMA2_CONFIG	0xFFC0 0C88
DMA3_CONFIG	0xFFC0 0CC8
DMA4_CONFIG	0xFFC0 0D08
DMA5_CONFIG	0xFFC0 0D48
DMA6_CONFIG	0xFFC0 0D88
DMA7_CONFIG	0xFFC0 0DC8
MDMA_D0_CONFIG	0xFFC0 0E08
MDMA_S0_CONFIG	0xFFC0 0E48
MDMA_D1_CONFIG	0xFFC0 0E88
MDMA_S1_CONFIG	0xFFC0 0EC8

Поля регистра DMAx_CONFIG используются для настройки параметров и рабочих режимов DMA.

- FLOW[2:0] (Следующая операция). Это поле определяет тип передачи DMA, следующей за выполняемой в настоящий момент. Возможны следующие опции:

Прямой доступ к памяти

0x0 – Остановка. При завершении передачи текущего блока данных канал DMA автоматически завершает работу после формирования прерывания (если данное прерывание разрешено). Бит состояния DMA_RUN в регистре DMAx_IRQ_STATUS обнуляется, а бит DMA_EN в регистре DMAx_CONFIG остаётся неизменным. В этом состоянии работа канала приостанавливается. Периферийные прерывания отфильтровываются устройством DMA. Работа канала может быть возобновлена записью в регистр DMAx_CONFIG, определяющей следующий передаваемый блок, с установкой бита DMA_EN в 1.

0x1 – Режим автобуферизации. В данном режиме дескрипторы не используются. Вместо этого, DMA выполняется непрерывно аналогично работе циклического буфера. При завершении передачи блока данных значения регистров параметров перезагружаются в регистры текущего состояния, что приводит к незамедлительному возобновлению DMA без непроизводительных затрат. Работа в режиме автобуферизации прекращается при записи пользователем значения 0 в бит DMA_EN регистра DMAx_CONFIG.

0x4 – Режим массива дескрипторов. В данном режиме выполняется выборка дескриптора из памяти, в котором отсутствуют элементы NDPH или NDPL. Так как дескриптор не содержит элемента указателя следующего дескриптора, система DMA по умолчанию использует для выборки дескрипторов регистр CURR_DESC_PTR. При этом возможно использование групп дескрипторов, размещённых в памяти непрерывно в виде массива.

0x6 – Режим списка дескрипторов (малая модель). В данном режиме выполняется выборка дескриптора из памяти, включая элемент NDPL. Выборка элемента NDPH не производится. Таким образом, старшие 16 битов поля указателя следующего дескриптора извлекаются из старших 16 битов регистра NEXT_DESC_PTR. При этом размещение дескрипторов в памяти ограничивается определённой страницей памяти размером 64 Кбайт.

0x7 – Режим списка дескрипторов (большая модель). В данном режиме выполняется выборка дескриптора из памяти, включая элементы NDPL и NDPH. За счет этого достигается максимальная гибкость размещения дескрипторов в памяти.

- NDSIZE[3:0] (Размер гибкого дескриптора). Данное поле определяет количество элементов дескриптора, загружаемых из памяти. В режимах остановки и автобуферизации значения данного поля должно равняться нулю. Если размер дескриптора, задаваемого полями NDSIZE и FLOW, выходит за пределы значения, определяемого элементом YMOD, возникает ошибка DMA.
- DI_EN (Разрешение прерывания по завершению передачи данных). Этот бит разрешает или запрещает генерацию прерывания после завершения передачи блока данных.
- DI_SEL (Выбор параметров прерывания по завершению передачи данных). Данный бит определяет частоту генерации прерывания по завершению передачи данных. В зависимости от его значения прерывание может

Прямой доступ к памяти

генерироваться по завершению передачи всего буфера или по завершению передачи каждой строки внутреннего цикла. Данный бит используется только при двумерном DMA

- **RESTART (Очистка буфера DMA).** В зависимости от значения данного бита перед началом операции со следующим блоком данных, принятые данные, содержащиеся в FIFO канала, сохраняются (`RESTART = 0`) или отбрасываются (`RESTART = 1`). Принятые данные автоматически удаляются при изменении значения бита `DMA_EN` из 0 в 1 (обычно это происходит при первом разрешении канала). Обычно необходимость сохранения принимаемых данных в FIFO между выполнением передач блока данных возникает, когда они образуют непрерывный поток передач. Однако, если блок данных является первым в новом потоке, бит `RESTART` должен быть установлен в 1 для удаления всех ранее принятых данных.



Бит `RESTART` влияет только на каналы записи в память в режиме DMA. Для каналов чтения из памяти в режиме DMA и каналов MDMA данный бит является зарезервированным и должен быть равен нулю.



Для каналов записи в память в режиме DMA бит `RESTART` влияет только на передачу первого блока данных, иницированную записью в регистр `DMAx_CONFIG`. Если бит `RESTART` устанавливается в элементе дескриптора, загружаемом в `DMACFG`, он игнорируется.

- **DMA2D (Режим DMA).** Данный бит определяет, учитываются ли при операциях DMA только регистры `X_COUNT` и `X_MODIFY` (режим одномерного DMA) или также регистры `Y_COUNT` и `Y_MODIFY` (режим двумерного DMA).
- **WDSIZE[1:0] (Размер передаваемого слова).** Механизм DMA поддерживает передачи 8-, 16- или 32-разрядных элементов. Каждая процедура запроса/предоставления доступа приводит к одному обращению к памяти (несмотря на то, что для передач 32-разрядных данных через 16-разрядный порт памяти или 16-разрядную шину доступа DMA требуется два такта). Величина инкремента (приращения) регистров указателя адреса DMA должна делиться на размер передаваемого слова без остатка (на 1 при передачах 8-разрядных слов, на 2 при передачах 16-разрядных слов и на 4 при передачах 32-разрядных слов).
- **WNR (Направление DMA).** Данный бит определяет направление DMA – чтение из памяти (0) или запись в память (1).
- **DMA_EN (Разрешение канала DMA).** Данный бит разрешает работу канала DMA.



Когда канал DMA периферийного устройства разрешён, поступление прерываний от периферийного устройства означает возникновение запросов DMA. Когда канал запрещён, устройство DMA игнорирует прерывание периферийного устройства и передаёт запрос на контроллер прерываний. Для того чтобы избежать непредсказуемых результатов, следует разрешать

Прямой доступ к памяти

канал DMA до разрешения работы соответствующего периферийного устройства и запрещать работу периферийного устройства до запрещения канала DMA.

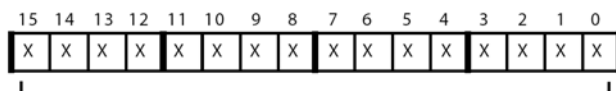
Регистр счётчика внутреннего цикла (DMA_x_X_COUNT/MDMA_{yy}_X_COUNT)

При двумерном DMA регистр X_COUNT, показанный на рис. 9-4, содержит значение счётчика внутреннего цикла. При одномерном DMA этот регистр задаёт количество читаемых элементов. Более подробное описание см. в разделе “Двумерный DMA”. Значение 0 регистра X_COUNT соответствует 65536 элементам.

Регистр счетчика внутреннего цикла (DMA_x_X_COUNT/MDMA_{yy}_X_COUNT)

До разрешения канала доступен для чтения и записи; после разрешения канала доступен только для чтения.

Адреса в карте памяти см. в таблице 9-7.



Состояние при сбросе = не определено

X_COUNT[15:0] (Счетчик внутреннего цикла)
Число читаемых элементов (при одномерном DMA);
число строк во внутреннем цикле (при двумерном DMA).

Рис. 9-4. Регистр счётчика внутреннего цикла

Таблица 9-7. Адреса регистров счётчика внутреннего цикла в карте памяти

Название регистра	Адрес в карте памяти
DMA0_X_COUNT	0xFFC0_0C10
DMA1_X_COUNT	0xFFC0_0C50
DMA2_X_COUNT	0xFFC0_0C90
DMA3_X_COUNT	0xFFC0_0CD0
DMA4_X_COUNT	0xFFC0_0D10
DMA5_X_COUNT	0xFFC0_0D50
DMA6_X_COUNT	0xFFC0_0D90
DMA7_X_COUNT	0xFFC0_0DD0
MDMA_D0_X_COUNT	0xFFC0_0E10
MDMA_S0_X_COUNT	0xFFC0_0E50
MDMA_D1_X_COUNT	0xFFC0_0E90
MDMA_S1_X_COUNT	0xFFC0_0ED0

Регистр инкремента адреса внутреннего цикла (DMA_x_X_MODIFY/MDMA_{yy}_X_MODIFY)

Регистр инкремента адреса внутреннего цикла (X_MODIFY) содержит знаковое в дополнительном коде значение инкремента адреса в байтах. В режиме

Прямой доступ к памяти

одномерного DMA значение инкремента определяет смещение в памяти, выполняемое после передачи каждого элемента.

i Следует отметить, что значение регистра X_MODIFY задаётся в байтах, независимо от размера слова при передачах DMA.

При двумерном DMA адрес инкрементируется в соответствии со значением этого регистра после передачи каждого элемента во внутреннем цикле вплоть до последнего, не включая его. После передачи последнего элемента каждого внутреннего цикла, за исключением передачи самого последнего элемента блока данных, используется значение инкремента, задаваемое регистром Y_MODIFY. При передаче последнего элемента блока данных всегда используется значение регистра X_MODIFY.

Поле X_MODIFY может обнуляться. При этом циклически выполняется чтение или запись в режиме DMA по одному и тому же адресу. Это свойство полезно, в частности, при выполнении передач между регистром данных и внешним периферийным устройством, отображённым в карте памяти.

Регистр инкремента адреса внутреннего цикла (DMAx_X_MODIFY/MDMA_yy_X_MODIFY)

До разрешения канала доступен для чтения и записи. После разрешения канала доступен только для чтения.

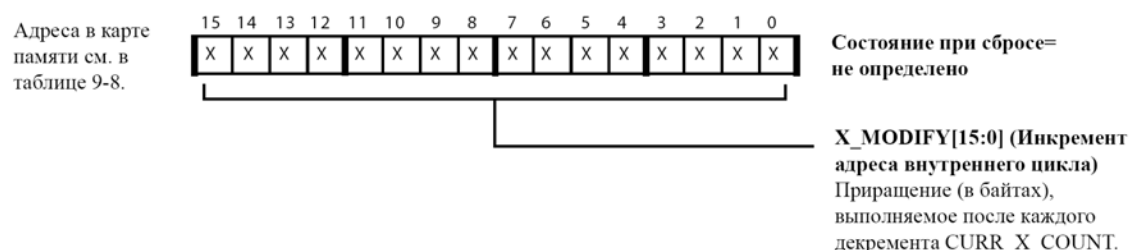


Рис. 9-5. Регистр инкремента адреса внутреннего цикла

Таблица 9-8. Адреса регистров инкремента адреса внутреннего цикла в карте памяти

Название регистра	Адрес в карте памяти
DMA0_X_MODIFY	0xFFC0_0C14
DMA0_X_MODIFY	0xFFC0_0C54
DMA0_X_MODIFY	0xFFC0_0C94
DMA0_X_MODIFY	0xFFC0_0CD4
DMA0_X_MODIFY	0xFFC0_0D14
DMA0_X_MODIFY	0xFFC0_0D54
DMA0_X_MODIFY	0xFFC0_0D94
DMA0_X_MODIFY	0xFFC0_0DD4
MDMA_D0_X_MODIFY	0xFFC0_0E14
MDMA_S0_X_MODIFY	0xFFC0_0E54
MDMA_D1_X_MODIFY	0xFFC0_0E94
MDMA_S1_X_MODIFY	0xFFC0_0ED4

Прямой доступ к памяти

Регистр счётчика внешнего цикла (DMAx_Y_COUNT/MDMA_yy_Y_COUNT)

В режиме двумерного DMA регистр счётчика внешнего цикла (Y_COUNT) содержит значение счётчика внешнего цикла. Он не используется в режиме одномерного DMA. Данный регистр содержит значение числа строк во внешнем цикле последовательности двумерного DMA. Более подробное описание см. в разделе “Двумерный DMA”.

Регистр счетчика внешнего цикла (DMAx_Y_COUNT/MDMA_yy_Y_COUNT)

До разрешения канала доступен для чтения и для записи. После разрешения канала доступен только для чтения.

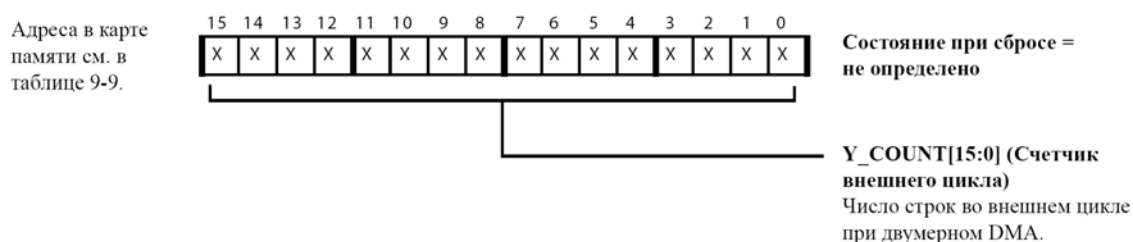


Рис. 9-6. Регистр счётчика внешнего цикла

Таблица 9-9. Адреса регистров счётчика внешнего цикла в карте памяти

Название регистра	Адрес в карте памяти
DMA0 Y COUNT	0xFFC0 0C18
DMA1 Y COUNT	0xFFC0 0C58
DMA2 Y COUNT	0xFFC0 0C98
DMA3 Y COUNT	0xFFC0 0CD8
DMA4 Y COUNT	0xFFC0 0D18
DMA5 Y COUNT	0xFFC0 0D58
DMA6 Y COUNT	0xFFC0 0D98
DMA7 Y COUNT	0xFFC0 0DD8
MDMA D0 Y COUNT	0xFFC0 0E18
MDMA S0 Y COUNT	0xFFC0 0E58
MDMA D1 Y COUNT	0xFFC0 0E98
MDMA S1 Y COUNT	0xFFC0 0ED8

Регистр инкремента адреса внешнего цикла (DMAx_Y_MODIFY/MDMA_yy_Y_MODIFY)

Регистр инкремента адреса внешнего цикла (Y_MODIFY) содержит знаковую величину в дополнительном коде. Инкремент адреса в байтах, задаваемый этим значением, выполняется после каждого декремента регистра CURR_Y_COUNT, за исключением случая передачи последнего элемента двумерного массива, когда значение CURR_Y_COUNT достигает нуля. Значение регистра Y_MODIFY задаёт смещение первого слова “строки” относительно последнего слова предыдущего “строки”. Более подробное описание см. в разделе “Двумерный DMA”.



Следует отметить, что значение регистра Y_MODIFY задаётся в байтах, независимо от размера слова при передачах DMA.

Прямой доступ к памяти

Регистр инкремента адреса внешнего цикла (DMAx_Y_MODIFY/MDMA_yy_YMODIFY)

До разрешения канала доступен для чтения и записи. После разрешения канала доступен только для чтения.

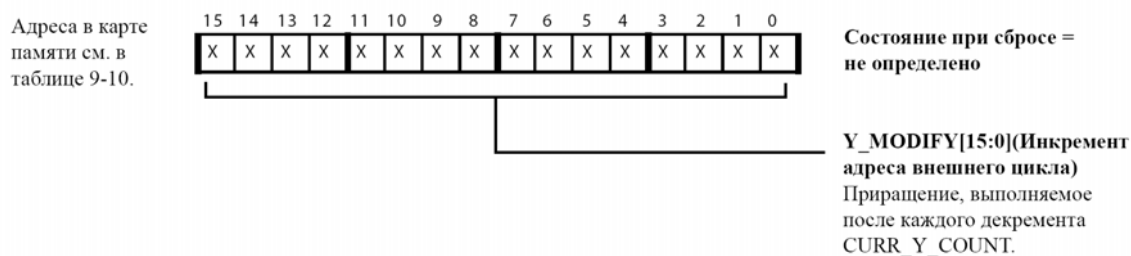


Рис. 9-7. Регистр инкремента адреса внешнего цикла

Таблица 9-10. Адреса регистров инкремента адреса внешнего цикла в карте памяти

Название регистра	Адрес в карте памяти
DMA0 Y MODIFY	0xFFC0 0C1C
DMA1 Y MODIFY	0xFFC0 0C5C
DMA2 Y MODIFY	0xFFC0 0C9C
DMA3 Y MODIFY	0xFFC0 0CDC
DMA4 Y MODIFY	0xFFC0 0D1C
DMA5 Y MODIFY	0xFFC0 0D5C
DMA6 Y MODIFY	0xFFC0 0D9C
DMA7 Y MODIFY	0xFFC0 0DDC
MDMA D0 Y MODIFY	0xFFC0 0E1C
MDMA S0 Y MODIFY	0xFFC0 0E5C
MDMA D1 Y MODIFY	0xFFC0 0E9C
MDMA S1 Y MODIFY	0xFFC0 0EDC

Регистр указателя текущего дескриптора (DMAx_CURR_DESC_PTR/MDMA_yy_CURR_DESC_PTR)

Регистр указателя текущего дескриптора (CURR_DESC_PTR) содержит значение адреса следующего загружаемого элемента дескриптора. В режимах, задаваемых полем FLOW, в которых используются дескрипторы (FLOW = 4, 6 или 7), значение этого регистра применяется для чтения и помещения в соответствующие регистры, отображённые в карте памяти, элементов дескрипторов перед началом передачи блока данных DMA. В режимах списка дескрипторов (FLOW = 6 или 7) перед загрузкой каждого дескриптора в данный регистр помещается значение, содержащееся в регистре NEXT_DESC_PTR. Затем, по мере чтения каждого элемента дескриптора, выполняется инкремент адреса, содержащегося в регистре CURR_DESC_PTR.

Когда чтение дескриптора завершено, регистр CURR_DESC_PTR содержит следующее значение:

Начальный адрес дескриптора + размер дескриптора (число элементов)

Прямой доступ к памяти



В режиме массива дескрипторов (FLOW = 4) перед началом работы в режиме DMA должна выполняться прямая запись по адресу в карте памяти в данный регистр, а не в регистр NEXT_DESC_PTR.

Регистр указателя текущего дескриптора (DMAx_CURR_DESC_PTR/ MDMA_yy_CURR_DESC_PTR)

До разрешения канала доступен для чтения и записи; после разрешения канала доступен только для чтения

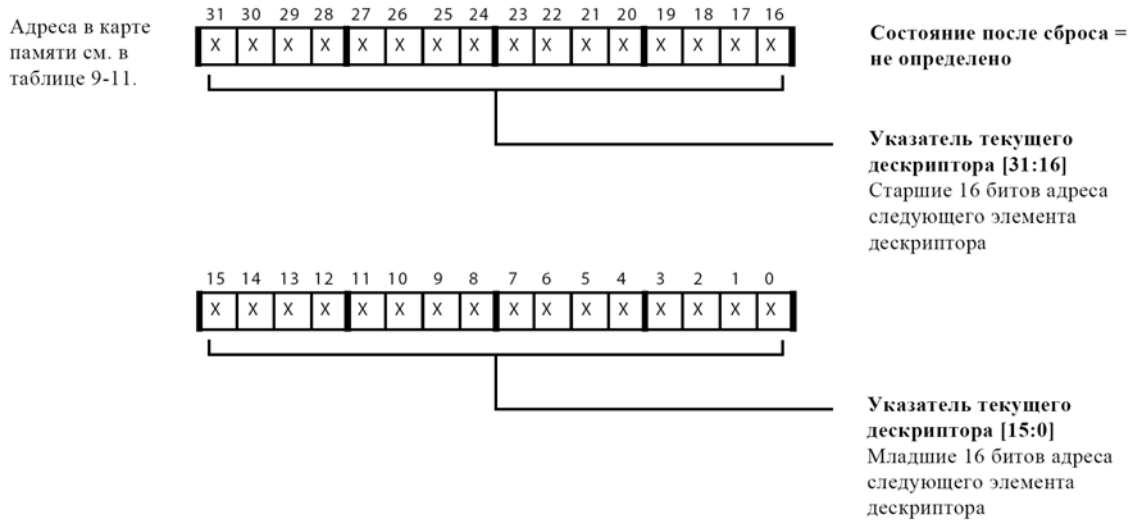


Рис. 9-8. Регистр указателя текущего дескриптора

Таблица 9-11. Адреса регистров указателя текущего дескриптора в карте памяти

Название регистра	Адрес в карте памяти
DMA0_CURR_DESC_PTR	0xFFC0_0C20
DMA1_CURR_DESC_PTR	0xFFC0_0C60
DMA2_CURR_DESC_PTR	0xFFC0_0CA0
DMA3_CURR_DESC_PTR	0xFFC0_0CE0
DMA4_CURR_DESC_PTR	0xFFC0_0D20
DMA5_CURR_DESC_PTR	0xFFC0_0D60
DMA6_CURR_DESC_PTR	0xFFC0_0DA0
DMA7_CURR_DESC_PTR	0xFFC0_0DE0
MDMA_D0_CURR_DESC_PTR	0xFFC0_0E20
MDMA_S0_CURR_DESC_PTR	0xFFC0_0E60
MDMA_D1_CURR_DESC_PTR	0xFFC0_0EA0
MDMA_S1_CURR_DESC_PTR	0xFFC0_0EE0

Регистр текущего адреса (DMAx_CURR_ADDR/ MDMA_yy_CURR_ADDR)

Регистр текущего адреса (CURR_ADDR), показанный на рис. 9-9, содержит адрес текущей передачи в режиме DMA для заданной сессии DMA. В начале сессии DMA в регистр CURR_ADDR загружается значение регистра START_ADDR; затем при каждой передаче это значение инкрементируется. Регистр текущего адреса является 32-разрядным.

Прямой доступ к памяти

Регистр текущего адреса (DMAx_CURR_ADDR/MDMA_yy_CURR_ADDR)

До разрешения канала доступен для чтения и записи; после разрешения канала доступен только для чтения

Адреса в карте памяти см. в таблице 9-12.

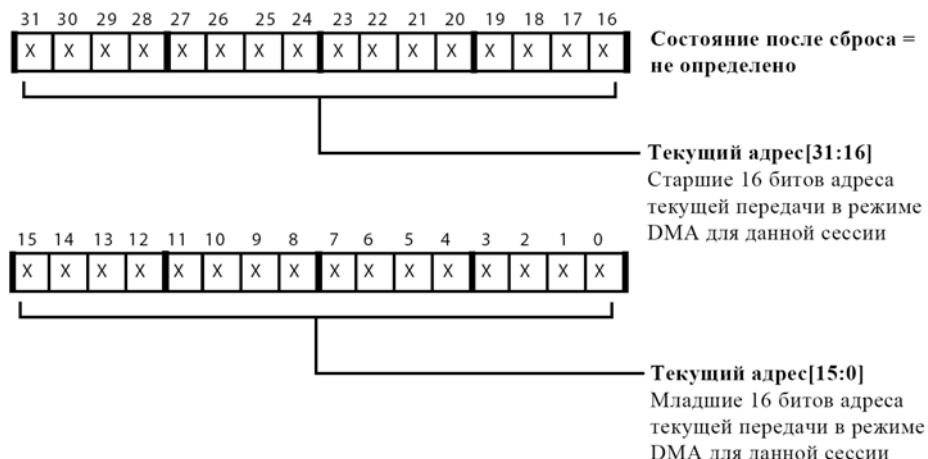


Рис. 9-9. Регистр текущего адреса

Таблица 9-12. Адреса регистра текущего адреса в карте памяти

Название регистра	Адрес в карте памяти
DMA0_CURR_ADDR	0xFFC0 0C24
DMA1_CURR_ADDR	0xFFC0 0C64
DMA2_CURR_ADDR	0xFFC0 0CA4
DMA3_CURR_ADDR	0xFFC0 0CE4
DMA4_CURR_ADDR	0xFFC0 0D24
DMA5_CURR_ADDR	0xFFC0 0D64
DMA6_CURR_ADDR	0xFFC0 0DA4
DMA7_CURR_ADDR	0xFFC0 0DE4
MDMA_D0_CURR_ADDR	0xFFC0 0E24
MDMA_S0_CURR_ADDR	0xFFC0 0E64
MDMA_D1_CURR_ADDR	0xFFC0 0EA4
MDMA_S1_CURR_ADDR	0xFFC0 0EE4

Регистр счётчика текущего внутреннего цикла (DMAx_CURR_X_COUNT/MDMA_yy_CURR_X_COUNT)

В начале каждой сессии DMA (в режиме одномерного DMA) и по завершению передачи в режиме DMA каждой строки (в режиме двумерного DMA) в регистр счётчика текущего внутреннего цикла (CURR_X_COUNT) загружается содержимое регистра X_COUNT. В остальное время значение этого регистра декрементируется после передачи каждого элемента. Достижение счётчиком нуля означает завершение DMA. При двумерном DMA значение регистра CURR_X_COUNT равняется нулю только при полном завершении передачи блока данных. В интервалах между передачами строк значение этого регистра равняется значению регистра X_COUNT.

Прямой доступ к памяти

Регистр счётчика текущего внутреннего цикла (DMAx_CURR_X_COUNT/MDMA_yy_CURR_X_COUNT)

До разрешения канала доступен для чтения и записи; после разрешения канала доступен только для чтения

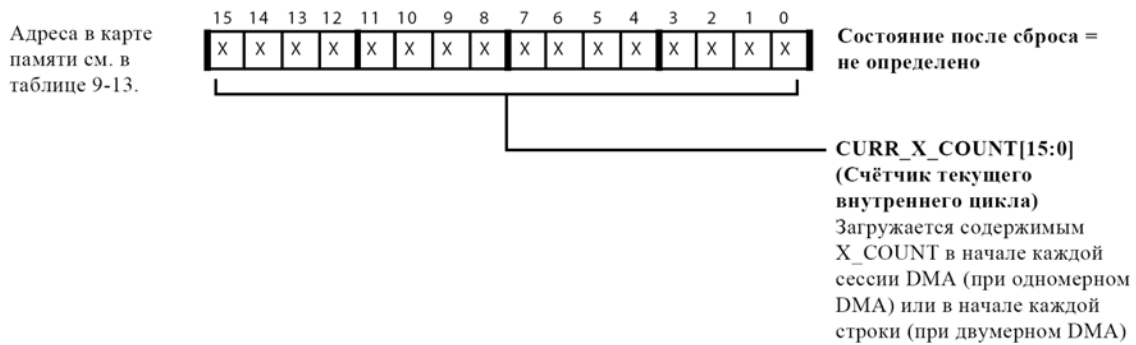


Рис. 9-10. Регистр счётчика текущего внутреннего цикла

Таблица 9-13. Адреса регистров счётчика текущего внутреннего цикла в карте памяти

Название регистра	Адрес в карте памяти
DMA0_CURR_X_COUNT	0xFFC0_0C30
DMA1_CURR_X_COUNT	0xFFC0_0C70
DMA2_CURR_X_COUNT	0xFFC0_0CB0
DMA3_CURR_X_COUNT	0xFFC0_0CF0
DMA4_CURR_X_COUNT	0xFFC0_0D30
DMA5_CURR_X_COUNT	0xFFC0_0D70
DMA6_CURR_X_COUNT	0xFFC0_0DB0
DMA7_CURR_X_COUNT	0xFFC0_0DF0
MDMA_D0_CURR_X_COUNT	0xFFC0_0E30
MDMA_S0_CURR_X_COUNT	0xFFC0_0E70
MDMA_D1_CURR_X_COUNT	0xFFC0_0EB0
MDMA_S1_CURR_X_COUNT	0xFFC0_0EF0

Регистр счётчика текущего внешнего цикла (DMAx_CURR_Y_COUNT/MDMA_yy_CURR_Y_COUNT)

В начале каждой сессии двумерного DMA в регистр счётчика текущего внешнего цикла (CURR_Y_COUNT) загружается содержимое регистра YCOUNT. В режиме одномерного DMA регистр CURR_Y_COUNT не используется. Значение данного регистра декрементируется при каждом истечении регистра CURR_X_COUNT (изменении его значения из 1 в X_COUNT или из 1 в 0) в режиме двумерного DMA, указывая на завершение передачи строки. При завершении сессии двумерного DMA CURR_Y_COUNT = 1 и CURR_X_COUNT = 0.

Прямой доступ к памяти

Регистр счётчика текущего внешнего цикла (DMAx_CURR_Y_COUNT/ MDMA_yy_CURR_Y_COUNT)

До разрешения канала доступен для чтения и записи; после разрешения канала доступен только для чтения

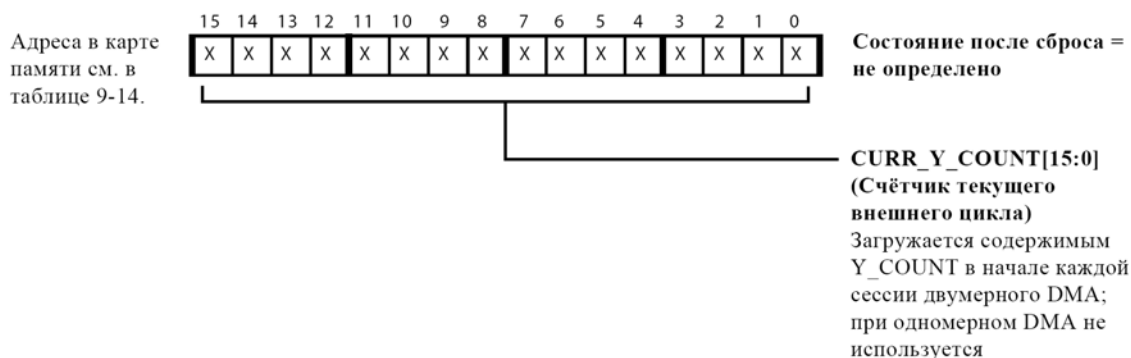


Рис. 9-11. Регистр счётчика текущего внешнего цикла

Таблица 9-14. Адреса регистров счётчика текущего внешнего цикла в карте памяти

Название регистра	Адрес в карте памяти
DMA0_CURR_Y_COUNT	0xFFC0_0C38
DMA1_CURR_Y_COUNT	0xFFC0_0C78
DMA2_CURR_Y_COUNT	0xFFC0_0CB8
DMA3_CURR_Y_COUNT	0xFFC0_0CF8
DMA4_CURR_Y_COUNT	0xFFC0_0D38
DMA5_CURR_Y_COUNT	0xFFC0_0D78
DMA6_CURR_Y_COUNT	0xFFC0_0DB8
DMA7_CURR_Y_COUNT	0xFFC0_0DF8
MDMA_D0_CURR_Y_COUNT	0xFFC0_0E38
MDMA_S0_CURR_Y_COUNT	0xFFC0_0E78
MDMA_D1_CURR_Y_COUNT	0xFFC0_0EB8
MDMA_S1_CURR_Y_COUNT	0xFFC0_0EF8

Регистр карты периферийных устройств (DMAx_PERIPHERAL_MAP/ MDMA_yy_PERIPHERAL_MAP)

Регистр PERIPHERAL_MAP каждого канала DMA содержит биты, которые:

- определяют соответствие между каналом и конкретным периферийным устройством;
- определяют, является ли канал каналом DMA периферийного устройства или каналом DMA типа “память-память”.

i Следует отметить, что между каналами DMA и периферийными устройствами должно существовать однозначное соответствие. Пользователь должен гарантировать отсутствие ситуаций, когда несколько каналов DMA назначаются одному периферийному устройству или один порт DMA назначается нескольким периферийным устройствам. Если одному периферийному устройству назначается несколько каналов, только один из них (канал с наиболее низким приоритетом) будет использоваться периферийным устройством. Если канал назначается

Прямой доступ к памяти

несуществующему периферийному устройству (например, когда поле RMAP содержит 0xF), его работа запрещается – запросы DMA игнорируются, и доступ в режиме DMA не выполняется. Запросы DMA при этом также не перенаправляются от периферийного устройства в контроллер прерываний.

Для того чтобы поменять приоритеты двух каналов DMA воспользуйтесь следующей процедурой. Предположим, что необходимо поменять приоритеты каналов 6 и 7.

1. Убедитесь, что запрещён DMA в каналах 6 и 7.
2. Запишите 0x7000 в регистр DMA6_PERIPHERAL_MAP и 0x6000 в регистр DMA7_PERIPHERAL_MAP.
3. Разрешите DMA в канале 6 и/или 7.

Регистр карты периферийных устройств (DMAx_PERIPHERAL_MAP/ MDMA_yy_PERIPHERAL_MAP)

До разрешения канала доступен для чтения и записи; после разрешения канала доступен только для чтения

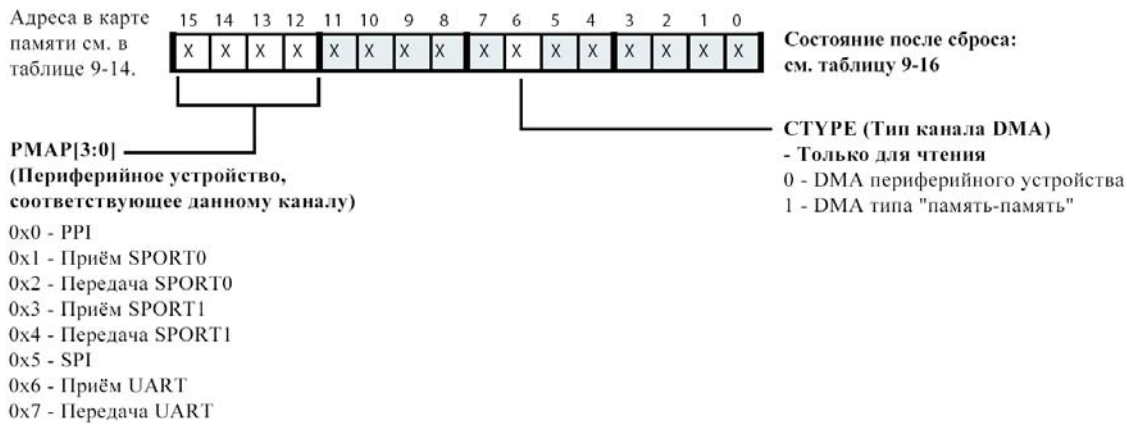


Рис. 9-12. Регистр карты периферийных устройств

Таблица 9-15. Адреса регистров карты периферийных устройств в карте памяти

Название регистра	Адрес в карте памяти
DMA0_PERIPHERAL_MAP	0xFFC0_0C2C
DMA1_PERIPHERAL_MAP	0xFFC0_0C6C
DMA2_PERIPHERAL_MAP	0xFFC0_0CAC
DMA3_PERIPHERAL_MAP	0xFFC0_0CEC
DMA4_PERIPHERAL_MAP	0xFFC0_0D2C
DMA5_PERIPHERAL_MAP	0xFFC0_0D6C
DMA6_PERIPHERAL_MAP	0xFFC0_0DAC
DMA7_PERIPHERAL_MAP	0xFFC0_0DEC
MDMA_D0_PERIPHERAL_MAP	0xFFC0_0E2C
MDMA_S0_PERIPHERAL_MAP	0xFFC0_0E6C
MDMA_D1_PERIPHERAL_MAP	0xFFC0_0EAC
MDMA_S1_PERIPHERAL_MAP	0xFFC0_0EEC

В таблице 9-16 перечислены значения регистров карты периферийных устройств в двоичной записи для каждого периферийного устройства, поддерживающего DMA.

Прямой доступ к памяти

Таблица 9-16. Отображение периферийных устройств в каналы DMA


Канал DMA	Периферийное устройство, по умолчанию соответствующее каналу	Значение регистра PERIPHERAL_MAP по умолчанию (в двоичной записи)	Комментарии
0 (наивысший приоритет)	PPI	b#0000 0000 0000 0000	
1	Приём SPORT0	b#0001 0000 0000 0000	
2	Передача SPORT0	b#0010 0000 0000 0000	
3	Приём SPORT1	b#0011 0000 0000 0000	
4	Передача SPORT1	b#0100 0000 0000 0000	
5	SPI	b#0101 0000 0000 0000	
6	Приём UART	b#0110 0000 0000 0000	
7	Передача UART	b#0111 0000 0000 0000	
8	Приёмник потока 0 DMA типа "память-память"	b#0000 0000 0100 0000	Не переназначается
9	Источник потока 0 DMA типа "память-память"	b#0000 0000 0100 0000	Не переназначается
10	Приёмник потока 1 DMA типа "память-память"	b#0000 0000 0100 0000	Не переназначается
11 (низший приоритет)	Источник потока 1 DMA типа "память-память"	b#0000 0000 0100 0000	Не переназначается


Регистр состояния прерываний (DMAx_IRQ_STATUS/ MDMA_yy_IRQ_STATUS)

Регистр состояния прерываний (IRQ_STATUS), показанный на рис. 9-13, содержит биты, указывающие информацию о том, что канал DMA:

- разрешён и работает, разрешён, но передачи остановлены, или запрещён;
- выполняет выборку данных или дескриптора DMA;
- обнаружил установление прерывания канала или глобального прерывания DMA;
- зарегистрировал возникновение ошибки DMA.

Необходимо отметить, что прерывание DMA_DONE устанавливается только после завершения последнего обращения к памяти (чтения или записи).

 При передаче из памяти в периферийное устройство в момент возникновения прерывания в FIFO канала DMA может находиться до четырёх слов данных. При этом, допускается немедленное начало передачи следующего блока данных. Однако если в приложении требуется знать момент действительной передачи последнего элемента данных в периферийное устройство, можно выполнить проверку состояния бита DMA_RUN. Бит DMA_RUN равен 1, пока в FIFO содержатся непереданные данные.

 Для каналов записи в память в режиме DMA после установки бита DMA_DONE состояние бита DMA_RUN не несёт информации. Оно не отображает состояние FIFO DMA.

Прямой доступ к памяти

i Если при выполнении передач DMA типа “память-память” нежелательно использование прерываний для информирования о завершении операции DMA, для определения момента завершения транзакции следует выполнять программный опрос бита DMA_DONE, а не бита DMA_RUN.

Регистр состояния прерываний (DMAx_IRQ_STATUS/MDMA_yy_IRQ_STATUS)

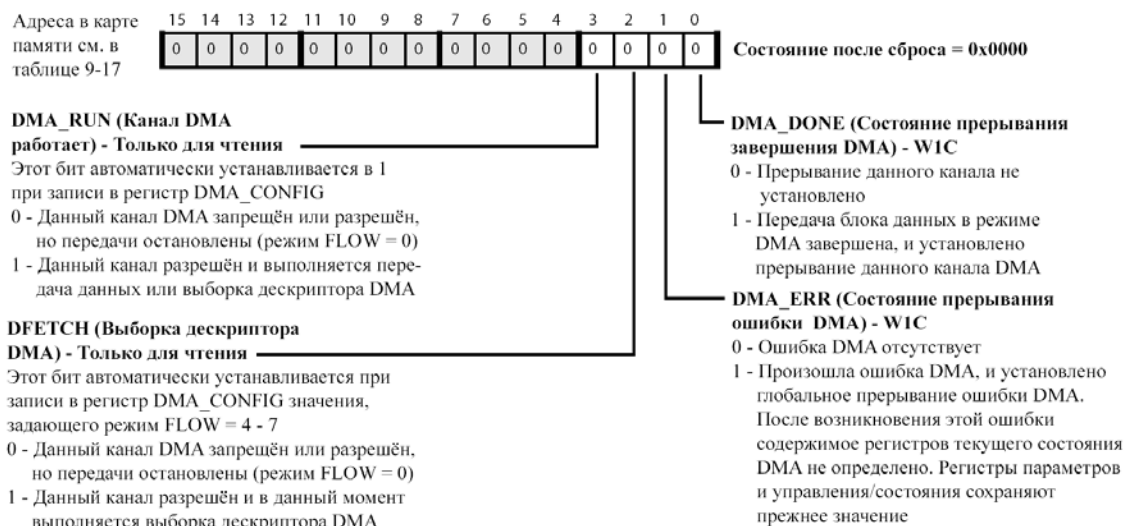


Рис. 9-13. Регистр состояния прерываний

Таблица 9-17. Адреса регистров состояния прерываний в карте памяти

Название регистра	Адрес в карте памяти
DMA0_IRQ_STATUS	0xFFC0 0C28
DMA1_IRQ_STATUS	0xFFC0 0C68
DMA2_IRQ_STATUS	0xFFC0 0CA8
DMA3_IRQ_STATUS	0xFFC0 0CE8
DMA4_IRQ_STATUS	0xFFC0 0D28
DMA5_IRQ_STATUS	0xFFC0 0D68
DMA6_IRQ_STATUS	0xFFC0 0DA8
DMA7_IRQ_STATUS	0xFFC0 0DE8
MDMA_D0_IRQ_STATUS	0xFFC0 0E28
MDMA_S0_IRQ_STATUS	0xFFC0 0E68
MDMA_D1_IRQ_STATUS	0xFFC0 0EA8
MDMA_S1_IRQ_STATUS	0xFFC0 0EE8

Процессор поддерживает гибкую структуру управления прерываниями с тремя источниками прерываний:

- прерывания, вызываемые передачей данных (прерывания данных);
- прерывания ошибок периферийных устройств;
- прерывания ошибок DMA(например, ошибки плохого дескриптора или ошибки шины).



Прерываниям ошибок периферийных устройств и прерываниям данных выделен уровень запроса прерывания (IRQ), отличный от уровня запроса прерывания, выделенного прерываниям ошибок DMA.

Прямой доступ к памяти

Таблица 9-18. Прерывания данных

Название прерывания	Описание
Нет прерывания	Для определённого блока данных прерывания могут быть запрещены.
Прерывания периферийных устройств	Прерывания периферийных устройств (не в режиме DMA).
Прерывание завершения передачи строки	Прерывание DMA может формироваться по завершению передачи строки (истечению CURR_X_COUNT).
Прерывание завершения передачи буфера	Прерывание DMA может формироваться по завершению передачи всего буфера (по истечению CURR_X_COUNT и CURR_Y_COUNT).

Прерывания всех каналов DMA обрабатываются по схеме логического ИЛИ, формируя одно прерывание ошибки DMA системного уровня. Для определения канала, вызвавшего прерывание ошибки DMA, следует выполнить чтение регистров IRQ_STATUS отдельных каналов.

-  Следует отметить, что индикаторы прерываний DMA_DONE и DMA_ERR имеют тип “запись-единицы-для-сброса” (W1C, write-one-to-clear).
-  При изменении режима работы периферийного устройства из режима DMA в режим без DMA следует запретить прерывания периферийного устройства (при помощи соответствующих регистров периферийного устройства или регистра IMASK), во избежание непреднамеренной генерации прерывания на разделяемой линии запроса DMA/прерывания.

Структура гибкого дескриптора

Гибкие дескрипторы DMA представляют собой структуры данных переменного размера, содержимое которых загружается в регистры параметров DMA. Последовательность регистров в дескрипторе значительным образом фиксирована (возможен выбор из трёх вариантов), а длина дескриптора полностью программируема. Регистры канала DMA упорядочены таким образом, что те регистры, которые чаще других приходится перезагружать перед началом передачи каждого нового блока данных, занимают младшие адреса в области регистров, отображённых в карте памяти.

Пользователь может выбирать, необходимо ли использование дескрипторов. Если дескрипторы не используются, для начала DMA пользователь может выполнить прямую запись в регистры DMA, отображённые в карте памяти, и использовать либо режим автобуферизации для непрерывной работы или режим остановки для выполнения однократной операции с буфером.

Для того чтобы использовать дескрипторы, пользователь выполняет запись в поле NDSIZE регистра DMAx_CONFIG значения количества регистров DMA, которые необходимо загрузить из дескриптора. Загрузка выполняется, начиная с регистра, имеющего младший адрес в карте памяти. Пользователь может выбирать размер

Прямой доступ к памяти

дескриптора от одного (младшие 16 битов регистра `START_ADDR`) до девяти (все параметры DMA) элементов.

Существует три варианта последовательностей элементов в дескрипторе, которые зависят от того, включается ли в дескриптор указатель следующего дескриптора, и если да, то какого типа:

- указатель следующего дескриптора не включается (режим массива дескрипторов);
- включаются младшие 16 битов указателя следующего дескриптора (режим списка дескрипторов, малая модель);
- включаются все 32 бита указателя следующего дескриптора (режим списка дескрипторов, большая модель).

Все остальные регистры, которые не загружаются из дескриптора, сохраняют предыдущие значения. Однако, между процедурами выборки дескриптора и началом операции DMA выполняется перезагрузка регистров `CURR_ADDR`, `CURR_X_COUNT` и `CURR_Y_COUNT`.

Существуют определённые настройки DMA, которые нельзя изменять для разных дескрипторов одной цепочки (в режимах массива дескрипторов и большой или малой модели списка дескрипторов). К ним относятся направление DMA, размер слова и пространство памяти (то есть невозможно переключение между внутренней и внешней памятью).

Передача последовательности буферов данных, располагающихся в различных пространствах памяти, не может управляться одной цепочкой дескрипторов. Вместо этого следует сгруппировать буферы в цепочки, каждая из которых располагается в одном пространстве памяти. При этом, цепочки нельзя связывать: следует передать первую цепочку, дождаться прерывания окончания её передачи, и затем начать передачу следующей цепочки записью в регистр `DMA_CONFIG`, отображённый в карте памяти.

Следует отметить, что в то время как пользователь может помещать буферы данных каждой цепочки в одно пространство памяти, сами структуры дескрипторов могут располагаться в любом пространстве памяти. Также можно связывать дескрипторы, располагающиеся в одном пространстве памяти, с дескрипторами, располагающимися в другом пространстве памяти, без всяких ограничений.

В таблице 9-19 показаны смещения элементов дескрипторов в трёх перечисленных режимах. Следует отметить, что названия в таблице обозначают элементы дескрипторов в памяти, а не регистры, отображённые в карте памяти, в которые загружаются дескрипторы.

Прямой доступ к памяти

Таблица 9-19. Регистры параметров и смещения дескрипторов

Смещение дескриптора	Режим массива дескрипторов	Режим малой модели списка дескрипторов	Режим большой модели списка дескрипторов
0x0	SAL	NDPL	NDPL
0x2	SAH	SAL	NDPH
0x4	DMACFG	SAH	SAL
0x6	XCNT	DMACFG	SAH
0x8	XMOD	XCNT	DMACFG
0xA	YCNT	XMOD	XCNT
0xC	YMOD	YCNT	XMOD
0xE		YMOD	YCNT
0x10			YMOD

Процесс выполнения операций DMA

На рис. 9-14 и 9-15 изображён процесс выполнения операций DMA.

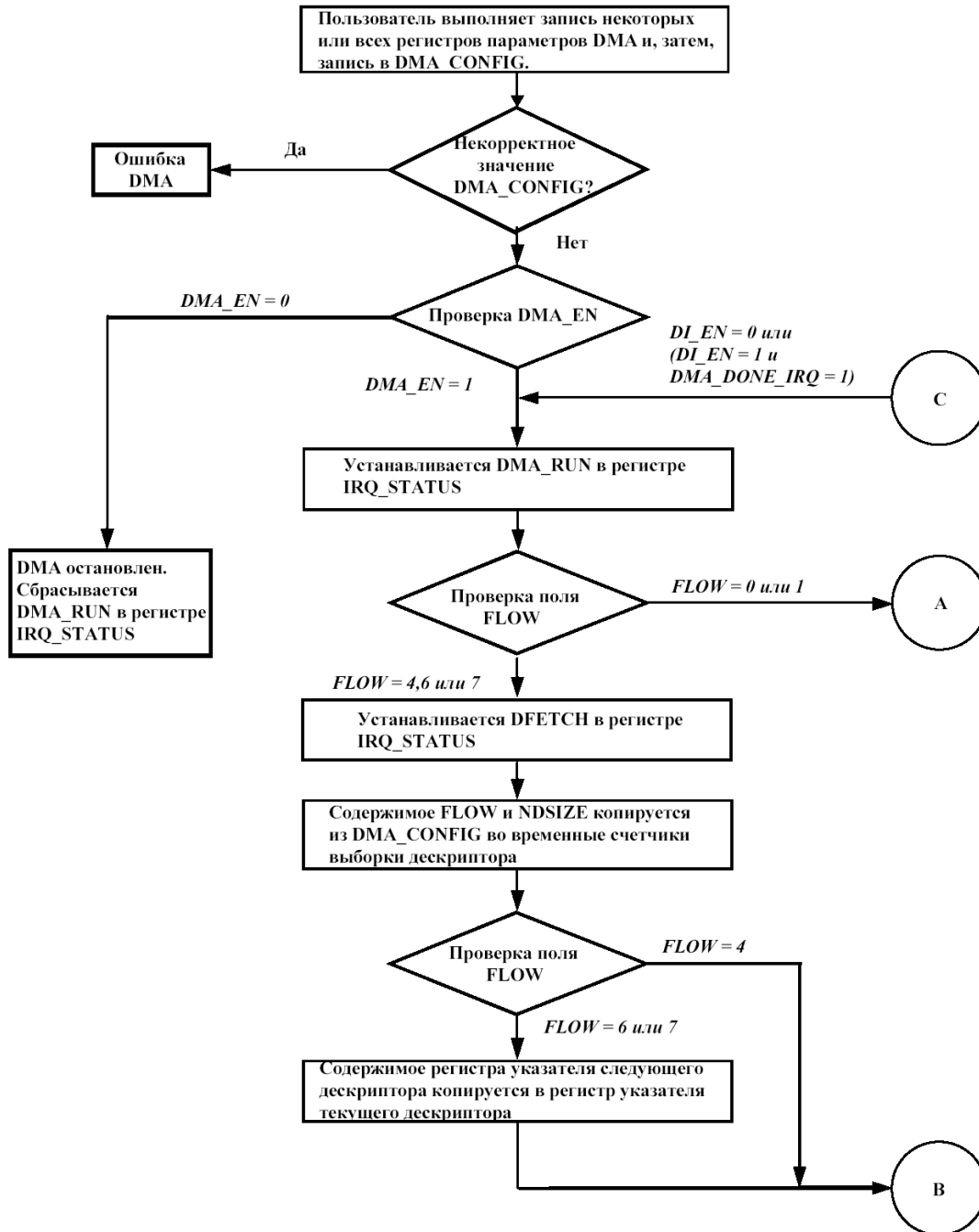


Рис. 9-14. Процесс выполнения DMA с точки зрения контроллера DMA (часть 1 из 2)

Прямой доступ к памяти

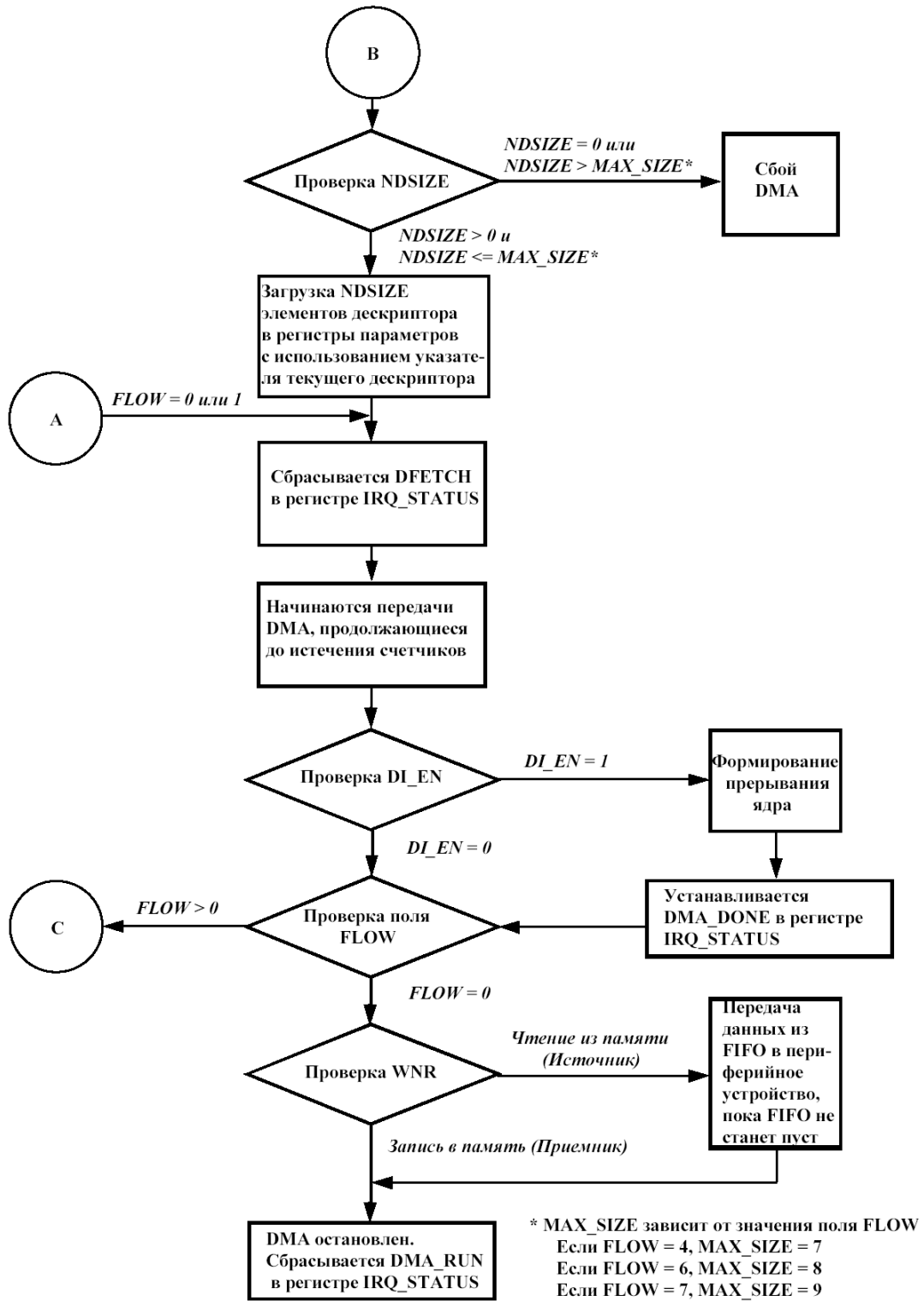


Рис. 9-14. Процесс выполнения DMA с точки зрения контроллера DMA (часть 2 из 2)

Запуск DMA

В этом разделе обсуждается запуск DMA “с нуля”. Аналогичным образом выполняется запуск DMA после остановки в режиме `FLOW = 0`.

- ⊘ Перед первой инициацией DMA в определённом канале, необходимо инициализировать все регистры параметров. Следует соблюдать особую осторожность при инициализации старших 16 битов регистров `NEXT_DESC_PTR` и `START_ADDR`, так как они могут стать недоступными в некоторых режимах работы, определяемых полем `FLOW`.

Для запуска операции DMA в определённом канале следует сначала выполнить прямую запись некоторых или всех регистров параметров DMA. На этом этапе необходимо, как минимум, выполнить запись в регистр `NEXT_DESC_PTR` (или в регистр `CURR_DESC_PTR` в режиме `FLOW = 4`). Однако, пользователь может также выполнить запись других регистров DMA, которые являются статическими при выполнении DMA (например, `X_MODIFY` или `Y_MODIFY`). Содержимое полей `NDSIZE` и `FLOW` в регистре `DMA_CONFIG` указывает, какие из регистров инициализируются, если это необходимо, выборкой из памяти элементов дескриптора. По завершении выборки дескриптора, если она выполняется, начинается операция DMA, инициируемая записью значения 1 в бит `DMA_EN` регистра `DMA_CONFIG`.

Контроллер DMA воспринимает прямую запись в регистр `DMA_CONFIG` как специальное условие первого запуска DMA для заданного канала или запуска механизма DMA после его остановки (в режиме `FLOW = 0`).

По завершении выборки дескриптора при `DMA_EN = 1` управление выполняется в соответствии со значением элемента `DMACFG` дескриптора, который был загружен в регистр `DMA_CONFIG`. До этого момента управление выполняется в соответствии со значением, которое было записано в регистр `DMA_CONFIG` напрямую. Другими словами, будут учитываться значения полей `WDSIZE`, `DI_EN`, `DI_SEL`, `RESTART` и `DMA2D`, взятые из элемента `DMACFG` дескриптора, а значения этих полей, записанные ранее напрямую в регистр `DMA_CONFIG`, будут игнорироваться.

Как показано на рис. 9-14 и 9-15, ход процесса настройки DMA определяется при запуске значениями битов `FLOW` и `NDSIZE`. Значение `FLOW` определяет необходимость загрузки дополнительных элементов дескриптора в регистры текущего состояния, а биты `NDSIZE` определяют количество элементов дескриптора, выбираемых из памяти перед запуском DMA. Регистры DMA, не включённые в дескриптор, не изменяют своего значения при загрузке дескриптора из памяти.

Если значение `FLOW` задаёт работу в режиме малой или большой модели списка дескрипторов, в регистр `CURR_DESC_PTR` копируется содержимое регистра `NEXT_DESC_PTR`. Затем выполняется выборка из памяти новых элементов дескриптора с индексацией при помощи регистра `CURR_DESC_PTR`,

Прямой доступ к памяти

инкрементируемого после каждой выборки. Если дескриптор содержит элементы NDPL и/или NDPH, их значение загружается в регистр NEXT_DESC_PTR, но выборка текущего дескриптора продолжается с использованием регистра CURR_DESC_PTR. По завершении выборки дескриптора регистр CURR_DESC_PTR указывает на следующее за последним элементом дескриптора 16-разрядное слово.

Если дескриптор не содержит ни NDPL, ни NDPH (то есть, в режиме массива дескрипторов, FLOW = 4), передача значения NDPL/NDPH в регистр CURR_DESC_PTR не выполняется. Вместо этого начинается выборка дескриптора с индексацией при помощи значения, содержащегося в регистре CURR_DESC_PTR в текущий момент.

Если дескриптор не содержит элемент DMACFG, управление операцией с блоком данных выполняется в соответствии с предыдущими настройками регистра DMA_CONFIG (значением, запрограммированным при запуске DMA в регистр DMA_CONFIG путём записи по его адресу в карте памяти). Если дескриптор содержит элемент DMACFG, значение, записанное в регистр DMA_CONFIG при помощи обращения по его адресу в карте памяти, влияет только на загрузку из памяти первого дескриптора. Управление последующими операциями DMA осуществляется в соответствии со значениями младшего байта элемента DMACFG дескриптора и регистров параметров, которые загружаются значениями элементов дескриптора. Запрограммированные путем прямой записи в регистр DMA_CONFIG значения битов DI_EN, DI_SEL, DMA2D, WDSIZE и WNR игнорируются.

Биты состояния DMA_RUN и DFETCH в регистре IRQ_STATUS отображают состояние канала DMA. После выполнения записи в регистр DMA_CONFIG биты DMA_RUN и DFETCH автоматически устанавливаются в 1. В результате загрузки из памяти первого дескриптора прерывания данных не генерируются.

После выполнения описанных шагов в регистры текущего состояния автоматически загружается содержимое соответствующих элементов дескриптора, перезаписывая предыдущее содержимое регистров:

- в регистр CURR_ADDR копируется содержимое элемента START_ADDR;
- в регистр CURR_X_COUNT копируется содержимое элемента X_COUNT;
- в регистр CURR_Y_COUNT копируется содержимое элемента Y_COUNT;

Затем начинается операция передачи данных в режиме DMA, как показано на рис. 9-15.

Обновление параметров DMA

По завершении передачи блока данных контроллер DMA выполняет следующие действия:

- Завершает передачу всех данных между памятью и устройством DMA.

Прямой доступ к памяти

- Устанавливает прерывание ядра, если оно разрешено битом `DI_EN`, и бит `DMA_DONE` в регистре `IRQ_STATUS` канала.
- Если `FLOW = 0` (режим остановки):

После передачи всех данных из FIFO DMA канала в периферийное устройство работа в режиме DMA останавливается сбросом бита `DMA_RUN` в регистре `IRQ_STATUS`.

- Во время выполнения выборки в режимах `FLOW = 4, 6 и 7` контроллер DMA устанавливает бит `DFETCH` в регистре `IRQ_STATUS` в значение 1. На данном этапе, выполнение операции DMA зависит от конкретного значения поля `FLOW`:

Если `FLOW = 4` (режим массива дескрипторов):

В регистры DMA из памяти загружается новый дескриптор; при этом используется значение регистра `CURR_DESC_PTR`, инкрементируемое после каждой выборки.

Размер дескриптора определяется перед началом выборки по значению поля `NDSIZE` регистра `DMA_CONFIG`.

Если `FLOW = 6` (режим малой модели списка дескрипторов):

В регистр `CURR_DESC_PTR` копируется 32 бита регистра `NEXT_DESC_PTR`. Затем в регистры DMA из памяти загружается новый дескриптор; при этом используется новое значение регистра `CURR_DESC_PTR`, инкрементируемое после каждой выборки. Первым загружаемым элементом дескриптора является новое 16-разрядное значение младших 16 битов регистра `NEXT_DESC_PTR`. За ним следуют остальные элементы дескриптора. Значение старших 16 битов регистра `NEXT_DESC_PTR` не изменяется. Таким образом, обеспечивается поддержка более эффективной модели дескриптора, чем при использовании режима большой модели списка дескрипторов. Данный режим подходит для случая, когда дескрипторы канала можно разместить в одной области памяти размером 64 Кбайта.

Если `FLOW = 7` (режим большой модели списка дескрипторов):

В регистр `CURR_DESC_PTR` копируется 32 бита регистра `NEXT_DESC_PTR`. Затем в регистры DMA из памяти загружается новый дескриптор; при этом используется новое значение регистра `CURR_DESC_PTR`, инкрементируемое после каждой выборки. Первым загружаемым элементом дескриптора является новое 32-разрядное значение регистра `NEXT_DESC_PTR`. За ним следуют остальные элементы дескриптора. Значение старших 16 битов регистра `NEXT_DESC_PTR` может отличаться от предыдущего. Таким образом, обеспечивается поддержка максимально гибкого списка дескрипторов,

Прямой доступ к памяти

который может располагаться в любой области внутренней, внешней памяти или ПЗУ.

Следует отметить, что при необходимости объединения цепочки дескрипторов, располагающихся в одной области памяти размером 64 Кбайта, с другой цепочкой дескрипторов, располагающихся вне этой области памяти, только в одном дескрипторе необходимо установить значение $FLOW = 7$. Этим дескриптором является дескриптор, используемый для объединения цепочек и располагающийся в конце области памяти размером 64 Кбайта. Во всех других дескрипторах, принадлежащих одной области памяти размером 64 Кбайта, может использоваться значение $FLOW = 6$.

- Если $FLOW = 1, 4, 6$ или 7 (режим автобуферизации, массива дескрипторов, малой модели списка дескрипторов и большой модели списка дескрипторов, соответственно):

Загружаются (или перезагружаются) регистры текущего состояния:

- в регистр `CURR_ADDR` загружается содержимое элемента `START_ADDR`;
- в регистр `CURR_X_COUNT` загружается содержимое элемента `X_COUNT`;
- в регистр `CURR_Y_COUNT` загружается содержимое элемента `Y_COUNT`;

Затем сбрасывается бит `DFETCH` в регистре `IRQ_STATUS`, после чего возобновляется передача в режиме DMA, как показано на рис. 9-15.

Действия, необходимые для остановки передач DMA

В режиме $FLOW = 0$ DMA автоматически останавливается по завершению передачи блока данных.

Если для управления DMA используется список или массив дескрипторов, и если каждый дескриптор содержит элемент `DMACFG`, для изящного завершения DMA в последнем дескрипторе элемент `DMACFG` должен содержать значение $FLOW = 0$.

В режиме автобуферизации ($FLOW = 1$), или при использовании списка дескрипторов, не содержащих элементы `DMACFG`, процесс передач в режиме DMA должен быть прекращён путём записи по адресу регистра `DMAx_CONFIG` в карте памяти значения, в котором бит `DMA_EN` равен 0. Обнуление всего регистра всегда приводит к изящному завершению DMA (без сбоя DMA).

Перед повторным разрешением канала DMA необходимо убедиться в завершении любых начатых операций чтения из медленной памяти (например, чтения из медленной внешней памяти). Разрешение канала до завершения любой такой операции чтения не рекомендуется.

Двумерный DMA

Механизм двумерного DMA поддерживает произвольные размеры строк и столбцов до 64К×64К элементов, а также произвольные значения X_MODIFY и Y_MODIFY до ±32 Кбайта. Кроме того, Y_MODIFY может иметь отрицательное значение, что позволяет реализовать перемежающиеся потоки данных. Значения X_COUNT и Y_COUNT определяют размеры строк и столбцов; значение X_COUNT должно быть не менее двух.

Начальный адрес и величины модификации задаются в байтах. Они должны быть кратны размеру слова при передачах DMA (задаваемому полем WDSIZE[1:0] в регистре DMA_CONFIG). Неправильное выравнивание этих величин вызывает ошибку DMA.

Значение X_MODIFY задаёт величину инкремента адреса в байтах, прибавляемую после каждой передачи, при которой декрементируется содержимое регистра CURR_X_COUNT. Значение X_MODIFY не используется, когда содержимое регистра CURR_X_COUNT изменяется из 1 в 0 при окончании внешнего цикла, за исключением случая последней передачи, когда CURR_Y_COUNT = 1 и содержимое регистра CURR_X_COUNT изменяется из 1 в 0.

Значение Y_MODIFY задаёт величину инкремента адреса в байтах, прибавляемую после каждой передачи, при которой декрементируется содержимое регистра CURR_Y_COUNT. Однако, значение Y_MODIFY не применяется при передаче последнего элемента массива, когда содержимое счётчика внешнего цикла (CURR_Y_COUNT) декрементируется из 1 в 0.

После завершения последней передачи CURR_Y_COUNT = 1, CURR_X_COUNT = 0 и содержимое CURR_ADDR равно адресу последнего элемента плюс X_MODIFY. Следует отметить, что если канал DMA запрограммирован на автоматическое обновление параметров (режим автобуферизации), то при первой передаче данных в эти регистры будут загружаться значения элементов X_COUNT, Y_COUNT и START_ADDR.

Примеры

Пример 1: Получение блока 16×8 байтов из буфера видеокadra размером (N×M) пикселей.

```
X_MODIFY = 1
X_COUNT = 16
Y_MODIFY = N-15 (смещение от конца одной строки до начала
другой)
Y_COUNT = 8
```

При этих значениях получаются следующие смещения адресов, относительного начального адреса:

Прямой доступ к памяти

0, 1, 2, ..., 15,
N, N + 1, ..., N + 15,
2N, 2N + 1, ..., 2N + 15, ...
7N, 7N + 1, ..., 7N + 15,

Пример 2: получение потока байтов видеоданных (R,G,B пикселей) ×(N×M размер изображения).

X_MODIFY = (N * M)
X_COUNT = 3
Y_MODIFY = 1 - 2(N * M) (отрицательное)
Y_COUNT = (N * M)

При этих значениях получаются следующие смещения адресов, относительно начального адреса:

0, (N * M), 2(N * M),
1, (N * M) + 1, 2(N * M) + 1,
2, (N * M) + 2, 2(N * M) + 2,
...
(N * M) - 1, 2(N * M) - 1, 3(N * M) - 1,

Дополнительные примеры двумерного DMA

При помощи гибких дескрипторов поддерживаются следующие варианты DMA:

- одиночный линейный буфер, по заполнению которого выполняется остановка DMA (FLOW = режим остановки);
- линейный буфер со смещением, превышающим 1 (X_MODIFY > 1);
- циклический, автоматически обновляемый буфер, по каждому заполнению которого генерируется прерывание;
- аналогичный буфер, по частичному заполнению которого (например, по заполнению 1/2 или 1/4 буфера) генерируется прерывание (используется при двумерном DMA);
- одномерный DMA с использованием набора идентичных буферов с попеременным заполнением, определяемых связанным кольцом трёхсловных дескрипторов, каждый из которых содержит указатель следующего дескриптора и 32-разрядный адрес;
- одномерный DMA с использованием с использованием связанного списка пятисловных дескрипторов, каждый из которых содержит указатель следующего дескриптора, 32-разрядный адрес, длину буфера и слово конфигурации (в стиле ADSP-2191);
- двумерный DMA с использованием массива однословных дескрипторов, задающих только базовый адрес DMA в общей странице данных;
- двумерный DMA с использованием связанного списка девятисловных дескрипторов, задающих все параметры.

DMA типа “память-память”

В этом разделе описывается контроллер DMA типа “память-память” (MDMA), обеспечивающий выполнение передач типа “память-память” между различными пространствами памяти, к которым относятся память L1 и области внешней синхронной/асинхронной памяти.

Каждый контроллер MDMA включает FIFO DMA, представляющий собой блок FIFO глубиной восемь 16-разрядных слов, используемый для обмена данными с L1 или шиной внешнего доступа (EAB). Обычно он используется для передач данных между внутренней и внешней памятью. Он также поддерживает передачи в режиме DMA из загрузочного ПЗУ по шине EAB. FIFO может использоваться для хранения данных, передаваемых в режиме DMA между двумя ячейками памяти L1 или между двумя ячейками внешней памяти.

В процессоре имеется четыре канала MDMA:

- два канала источника (для чтения из памяти);
- два канала приёмника (для записи в память).

Каждый канал приёмника/источника формирует “поток”. Этим двум потокам жёстко назначены приоритеты DMA с 8 по 11.

- Приоритет 8: поток приёмника DMA типа “память-память” D0
- Приоритет 9: поток источника DMA типа “память-память” S0
- Приоритет 10: поток приёмника DMA типа “память-память” D1
- Приоритет 11: поток источника DMA типа “память-память” S1

Поток 0 DMA типа “память-память” обслуживается раньше потока 1 DMA типа “память-память”, если не используется круговое планирование. Следует отметить, что запрещено программирование потока источника на запись в память или потока приёмника на чтение из памяти.

Каналы поддерживают 8-, 16- и 32- разрядные передачи DMA типа “память-память”, но оба конца (приёмный и передающий) должны быть запрограммированы на одинаковый размер слов. Другими словами, при выполнении передач DMA типа “память-память” не осуществляется упаковывание или распаковывание данных; каждая передача реализует одну операцию записи. Приоритет обоим концам FIFO MDMA определённого потока выделяется в одно время. Каждая пара разделяет 16-разрядный FIFO глубиной 8 слов. Механизм DMA источника заполняет FIFO, механизм DMA приёмника опустошает его. Глубина FIFO позволяет выполнять пакетные передачи по шине внешнего доступа (EAB) и шине доступа DMA (DAB) с перекрытием, что значительно увеличивает пропускную способность блочных передач между внутренней и внешней памятью. Для задания рабочих параметров каждой пары MDMA требуется два отдельных блока дескрипторов: один – для канала источника, другой – для канала приёмника.

Так как контроллеры DMA источника и приёмника используют один буфер FIFO, соответствующие блоки дескрипторов должны иметь одинаковый размер.

Прямой доступ к памяти

Допускаются различные комбинации дескрипторов на обоих концах, пока их общее количество остаётся одинаковым.

Для начала передач в режиме MDMA необходимо выполнить запись регистров потоков приёмника и источника, отображённых в карте памяти аналогично запуску операций в режиме DMA с периферийными устройствами.



Следует отметить, что запись в регистр `DMA_CONFIG` источника потока должна выполняться до записи в регистр `DMA_CONFIG` приёмника потока.

Работа в режиме MDMA начинается после записи регистр `DMA_CONFIG` приёмника с задержкой в 3 такта `SCLK`.

На первом этапе, если для любого из потоков MDMA была задана необходимость использования дескрипторов, выполняется выборка дескрипторов из памяти. Сначала выполняется выборка дескрипторов потока приёмника. Затем, с задержкой в 4 такта `SCLK` относительно получения последнего слова дескриптора из памяти (или, обычно, 8 тактов `SCLK` относительно выборки последнего слова дескриптора, вследствие конвейерного характера обращений к памяти), поток MDMA источника начинает выборку данных из буфера источника. Получаемые данные размещаются в 8-элементном FIFO потока MDMA, после чего, с задержкой в два такта `SCLK`, поток MDMA приёмника начинает запись данных в буфер памяти приёмника.

Пропускная способность MDMA

Если источник и приёмник находятся в различных пространствах памяти (один – во внутренней, другой – во внешней памяти), передачи между внутренней и внешней памятью обычно выполняются одновременно и непрерывно, поддерживая 100% использование интерфейсов внутренней и внешней памяти. На производительность в данном случае влияет отношение тактовых частот ядра и системы. При отношении ниже 2.5:1 задержки конвейера и синхронизации приводят к меньшему использованию ресурса шин в домене тактового сигнала системы. При отношении тактовых частот 2:1, например, DMA выполняется с частотой, равной 2/3 частоты тактового сигнала системы. При больших отношениях частот тактовых сигналов достигается полная пропускная способность.

Если источник и приёмник находятся в одном пространстве памяти (оба во внешней или оба во внутренней памяти), поток MDMA обычно выполняет предвыборку пакета данных источника в FIFO, и затем автоматически все данные передаются из FIFO в буфер приёмника. Длина пакета зависит от трафика и равна 3 плюс задержка обращения к памяти в режиме DMA, выраженная в тактах `SCLK` (обычно равная 7 при внутренних передачах и 6 при внешних передачах).

Оптимизация производительности DMA

Система DMA разработана таким образом, чтобы обеспечивать максимальную пропускную способность на один канал и максимальное использование внутренних шин, в то же время, учитывая неизбежные задержки обращения к памяти.

Ключевой особенностью архитектуры DMA является разделение операций с использованием шины DMA периферийных устройств (шины DAB) и операций с использованием шин между DMA и памятью (шины DMA ядра (DCB) и внешней шины DMA (DEB)). Каждый канал DMA периферийных устройств имеет собственный FIFO данных между шиной DAB и шинами памяти. Эти FIFO выполняют автоматическую предвыборку данных из памяти для передачи и буферизацию принимаемых данных для последующих операций записи в память. Это позволяет предоставление доступа периферийному устройству с очень малой задержкой по сравнению с полной задержкой конвейерного обращения к памяти. При этом достигается максимально возможная скорость повторения операций в режиме DMA (пропускная способность) для каждого канала.

Максимальная скорость передач каналов DMA периферийных устройств составляет одно 16-разрядное слово на два такта тактового сигнала системы для каждого канала в любом направлении.

Максимальная скорость передач каналов MDMA составляет одно 16-разрядное слово на один такт тактового сигнала системы (SCLK) для каждого канала

Если рассматривать трафик всех каналов DMA в совокупности:

- Максимальная скорость передач между периферийными устройствами и устройствами DMA – одна 16-разрядная передача на один такт тактового сигнала системы.
- Максимальная скорость передач между устройством DMA и внутренней памятью (L1) – одна 16-разрядная передача на один такт тактового сигнала системы.
- Максимальная скорость передач между устройством DMA и внешней памятью – одна 16-разрядная передача на один такт тактового сигнала системы.

Следующие условия ограничивают действительную производительность DMA:

- Доступы к внутренней или внешней памяти, конфликтующие с доступами ядра к той же области памяти. Они могут вызвать задержки, например, при доступе к одному банку L1, при открытии/закрытии страницы SDRAM или при заполнении строк кэша.
- Каждое изменение направления с передачи на приём на шине DAB вносит задержку в один такт SCLK.
- Изменение направления доступа по шине DCB (например, запись с последующим чтением) к одному банку внутренней памяти может внести задержки.

Прямой доступ к памяти

- Каждое изменение направления доступа по шине DEB (например, чтение с последующей записью) к внешней памяти может внести задержку в несколько тактов.
- Доступ по адресу в карте памяти к регистрам DMA, отличным от DMAx_CONFIG, DMAx_IRQSTAT или DMAx_PERIPHERAL_MAP, вносит останов в процесс DMA в один такт на одно передаваемое 16-разрядное слово. Доступы по адресу в карте памяти к регистрам управления/состояния не добавляют остановки или состояния ожидания.
- При операциях чтения регистров DMA, отличных от регистров управления/состояния, добавляется одно состояние ожидания шины PAB, что вносит задержку в несколько тактов ядра.
- Выборка дескриптора занимает один цикл DMA типа “память-память” на чтение одного 16-разрядного слова из памяти, но не задерживает передачи по шине DAB.
- Инициализация канала DMA вносит останов в процесс DMA в один такт. Это происходит при изменении бита DMA_EN из 0 в 1 или при установлении бита RESTART в регистре DMAx_CONFIG в 1.

Влияние некоторых из этих факторов может быть минимизировано путём надлежащей разработки программы. Зачастую возможно структурирование программы для избежания конфликтов между внешней и внутренней памятью, за счёт планирования низкой активности кэша при выполнении критических операций DMA и тщательного распределения буферов данных в пределах банков и страниц. Более того, необходимо минимизировать необязательные обращения к регистрам, отображённым в карте памяти, особенно, при использовании дескрипторов или в режиме автобуферизации.

Свойства управления трафиком процессора, описываемые в следующем разделе, позволяют минимизировать потерю эффективности, вызываемую чрезмерными изменениями направления передач (thrashing).

Назначение приоритетов и управление трафиком

Обычно каналы DMA обслуживаются строго в соответствии с их приоритетом. Приоритет канала равен его номеру, каналы с меньшим номером обслуживаются в первую очередь. Таким образом, периферийным устройствам, требующим высокой скорости передачи данных и низких задержек, следует назначать в регистрах DMAx_PERIPHERAL_MAP каналы с меньшими номерами (более высоким приоритетом). Потоки DMA типа “память-память” всегда имеют приоритет ниже, чем периферийные устройства, но так как они непрерывно запрашивают обслуживание, гарантируется выделение передач MDMA любых временных слотов, не задействованных операциями DMA периферийных устройств. По умолчанию, когда разрешены и готовы к обслуживанию несколько потоков MDMA, доступ предоставляется только одному из них. Если необходимо, чтобы доступная пропускная способность совместно использовалась потоками MDMA, можно задать режим поочерёдного выполнения каждым потоком фиксированного числа передач при помощи регистра MDMA_ROUND_ROBIN_PERIOD.

Прямой доступ к памяти

В системе DMA процессора имеется два полностью отдельных, но одновременно выполняемых, процесса назначения приоритетов – назначение приоритета шины DAB и назначение приоритетов шин памяти (DCB и DEB). Периферийные устройства, запрашивающие DMA по шине DAB, и чьи FIFO готовы к передачам, конкурируют друг с другом за доступ к этой шине. Схожим образом, но независимо, каналы, чьи FIFO требуют обслуживания памяти (выполнения операций предвыборки или постзаписи) конкурируют за доступ к шинам памяти. Потоки MDMA конкурируют за доступ к памяти как отдельное устройство; возможно одновременное предоставление доступа приёмнику и источнику, если при этом не возникают конфликты. В данном случае передачи из внутренней во внешнюю и из внешней во внутреннюю память могут выполняться с частотой тактового сигнала системы (SCLK). Примерами конфликтов памяти являются одновременное обращение к одному пространству памяти и одновременные попытки выборки дескрипторов. В случае если DMA запрашивается периферийным устройством, FIFO которого не готов (FIFO передачи пуст или заполнен FIFO приёма), может потребоваться дополнительная обработка. Дополнительную информацию см. в разделе “Срочные передачи DMA”.

Управление трафиком является важным фактором при оптимизации использования ресурсов DMA. Оно позволяет влиять на частоту изменения направления передач на шинах данных, путём автоматического группирования передач с совпадающим направлением. В блоке DMA процессора реализован механизм управления трафиком, регулируемый при помощи регистров DMA_TC_PER и DMA_TC_CNT. Этот механизм выполняет оптимизацию в режиме реального времени без вмешательства процессора и без необходимости настраивать потоки DMA на выполнение пакетных передач. Для каждой из трёх шин (DAB, DCB и DEB) трафик может независимо управляться при помощи простых счётчиков. Кроме того, при помощи поля MDMA_ROUND_ROBIN_CNT регистра DMA_TC_CNT возможно управление чередованием передач потоков MDMA. См. раздел “Приоритет и планирование MDMA”.

При использовании свойств управления трафиком система DMA, выделяя доступ, оказывает предпочтение передачам данных по шине DAB или шинам памяти, направление которых (чтение/запись) совпадает с направлением предыдущей передачи, до тех пор, пока не истечёт счётчик управления трафиком, или пока трафик не прекратится или не изменит направление самостоятельно. После того как значение счётчика трафика достигает нуля, предпочтение отдаётся противоположному направлению передач. Оказание предпочтения определённому направлению выражается в понижении приоритета каналов, работающих в противоположном направлении, на 16.

Например, если доступ к шине DAB запрашивается каналами 3 и 5, но канал 5 (имеющий более низкий приоритет) выполняет передачи, направление которых “совпадает с трафиком”, а канал 3 (имеющий более высокий приоритет) выполняет передачи в направлении, “противоположном трафику”, то эффективный приоритет канала 3 становится равен 19, и доступ предоставляется каналу 5. Если на следующем такте шины передачи DMA запрашиваются только каналами 3 и 6, которые выполняют передачи в направлении, “противоположном трафику”, то их эффективные приоритеты становятся равны 19 и 22,

Прямой доступ к памяти

соответственно. Одному из каналов (каналу 3) предоставляется доступ, несмотря на то, что его направление не совпадает с текущим приоритетным направлением. При этом к задержке, требуемой для изменения направления передач шины, не добавляются дополнительные такты.

При использовании подобного управления трафиком может достигаться компромисс между задержкой и увеличением эффективности (коэффициента использования) шины. Большие значения таймаутов могут привести к увеличению времени, требуемого для каждого предоставления доступа к шине, но зачастую при этом в перегруженных системах значительно возрастает (до 90%) достигаемая пропускная способность.

Для запрещения назначения приоритетов DMA с оказанием предпочтений, следует записать в регистр DMA_TC_PER значение 0x0000.

Регистр периода счётчика управления трафиком DMA (DMA_TC_PER) и регистр счётчика управления трафиком DMA (DMA_TC_CNT)

Поле MDMA_ROUND_ROBIN_COUNT отображает текущее значение счётчика передач, оставшихся до окончания периода кругового планирования MDMA. Оно инициализируется значением MDMA_ROUND_ROBIN_PERIOD при каждой записи в регистр DMA_TC_PER, при каждом предоставлении доступа другому потоку MDMA или каждый раз, когда все потоки MDMA находятся в состоянии ожидания. Затем при каждой передаче в режиме MDMA его значение декрементируется. Когда этот счётчик изменяет значение из 1 в 0, выбирается следующий требующий обслуживания поток MDMA.

Поле DAB_TRAFFIC_COUNT отображает текущее значение счётчика тактов, оставшихся до окончания периода трафика DAB. Оно инициализируется значением DAB_TRAFFIC_PERIOD при каждой записи в регистр DMA_TC_PER, или каждый раз, когда изменяется направление передач шины DAB или шина переходит в состояние ожидания. Затем на каждом такте тактового сигнала системы (за исключением интервалов останова DMA) счётчик декрементируется. Когда значение этого счётчика не равно нулю, доступы к DAB с совпадающими направлениями рассматриваются как предпочтительные. После того как счётчик изменяет значение из 1 в 0, предпочтение будет оказываться доступам к DAB, имеющим противоположное направление, что может привести к изменению направления передач шины. Когда значение этого счётчика равно 0 и происходит доступ к шине DAB, счётчик снова загружается значением DAB_TRAFFIC_PERIOD, и начинается новый цикл передач.

Поле DEB_TRAFFIC_COUNT отображает текущее значение счётчика тактов, оставшихся до окончания периода трафика DAB. Оно инициализируется значением DEB_TRAFFIC_PERIOD при каждой записи в регистр DMA_TC_PER, или каждый раз, когда изменяется направление передач шины DEB или шина переходит в состоянии ожидания. Затем на каждом такте тактового сигнала системы (за исключением интервалов останова DMA) счётчик декрементируется.

Прямой доступ к памяти

Когда значение этого счётчика не равно нулю, доступы к DEB с совпадающими направлениями рассматриваются как предпочтительные. После того как счётчик изменяет значение из 1 в 0, предпочтение будет оказываться доступам к DEB, имеющим противоположное направление, что может привести к изменению направления передач шины. Когда значение этого счётчика равно 0 и происходит доступ к шине DEB, счётчик снова загружается значением DEB_TRAFFIC_PERIOD, и начинается новый цикл передач.

Поле DCB_TRAFFIC_COUNT отображает текущее значение счётчика тактов, оставшихся до окончания периода трафика DCB. Оно инициализируется значением DCB_TRAFFIC_PERIOD при каждой записи в регистр DMA_TC_PER, или каждый раз, когда изменяется направление передач шины DCB или шина переходит в состоянии ожидания. Затем на каждом такте тактового сигнала системы (за исключением интервалов останова DMA) счётчик декрементируется. Когда значение этого счётчика не равно нулю, доступы к DCB с совпадающими направлениями рассматриваются как предпочтительные. После того как счётчик изменяет значение из 1 в 0, предпочтение будет оказываться доступам к DCB, имеющим противоположное направление, что может привести к изменению направления передач шины. Когда значение этого счётчика равно 0 и происходит доступ к шине DCB, счётчик снова загружается значением DCB_TRAFFIC_PERIOD, и начинается новый цикл передач.

Регистр периода счётчика управления трафиком DMA (DMA_TC_PER)

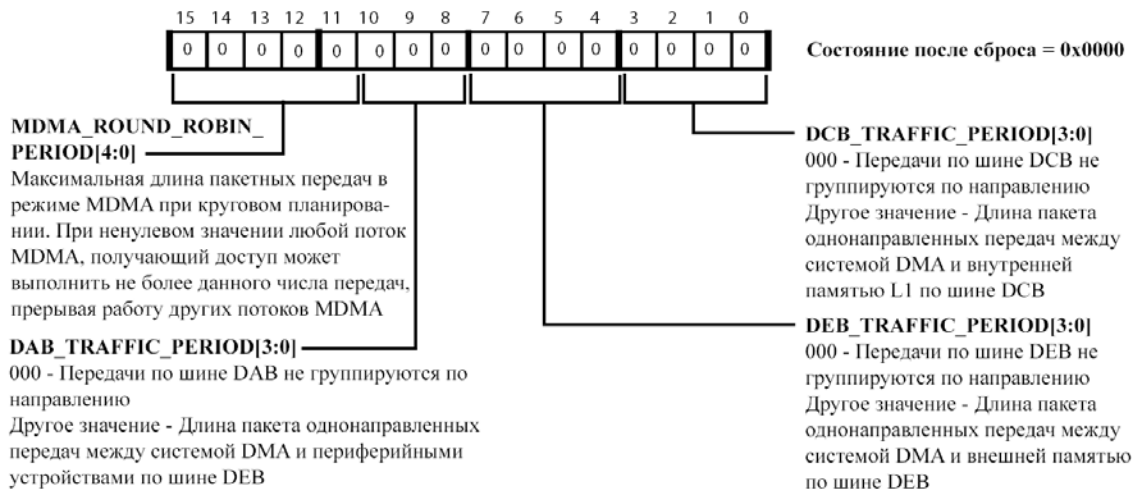


Рис. 9-16. Регистр периода счётчика управления трафиком DMA

Прямой доступ к памяти

Регистр счётчика управления трафиком DMA (DMA_TC_CNT)

Только для чтения

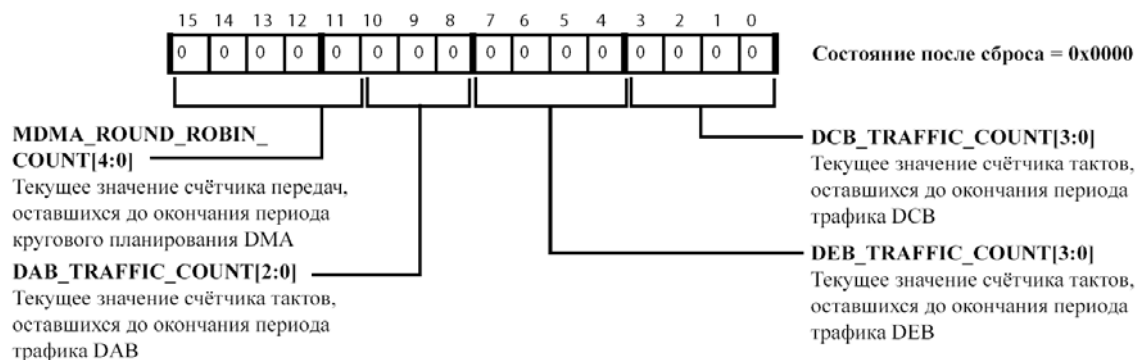


Рис. 9-17. Регистр счётчика управления трафиком DMA

Приоритет и планирование MDMA

Все операции MDMA имеют меньший приоритет, чем любая операция периферийного устройства. Таким образом, MDMA использует пропускную способность памяти, не задействованную трафиком DMA периферийных устройств.

Если используются два потока MDMA (S0–D0 и S1–D1), пользователь может выбирать между выделением пропускной способности с фиксированным приоритетом канала или с применением схемы кругового планирования. Этот выбор осуществляется программированием поля MDMA_ROUND_ROBIN_PERIOD в регистре DMA_TC_PER (см. раздел “Назначение приоритетов и управление трафиком”).

Если значение данного поля обнуляется, выполняются операции MDMA с фиксированным приоритетом. Когда поток 0 MDMA готов к выполнению передачи, он имеет преимущество перед потоком 1 MDMA. Так как обычно поток 0 MDMA может выполнять передачи данных на каждом такте, трафик потока 1 MDMA может быть задержан на неопределённое время до завершения любой или всех операций потока 0 MDMA. Этот метод может использоваться в системах, где необходимо немедленное выполнение кратковременных операций пересылок буферов данных, чувствительных к задержке, прерывающих длительные фоновые передачи с низким приоритетом.

Если в поле MDMA_ROUND_ROBIN_PERIOD записывается ненулевое значение в диапазоне $1 \leq P \leq 31$, используется метод кругового планирования. Доступ к шине предоставляется потокам MDMA поочерёдно для выполнения пакетов до P передач данных. Этот метод может применяться в системах, где необходимо совместное выполнение двух передач, каждой из которых должно быть гарантировано выделение части доступной пропускной способности. Например, один из потоков может быть запрограммирован на выполнение передач из внутренней во внешнюю память, а другой – на выполнение передач из внешней

Прямой доступ к памяти

памяти во внутреннюю. При этом каждому потоку будет выделена примерно равная часть пропускной способности.

При работе в режиме кругового планирования, поток MDMA в любой момент времени может иметь состояние “свободен” или “занят”. Изначально потоки MDMA имеют состояние “свободен”. На любом такте, в котором могут выполняться передачи в режиме MDMA (когда не выполняются приоритетные передачи периферийных устройств в режиме DMA), если оба потока MDMA запрашивают доступ, он будет предоставлен каналу с большим приоритетом (каналу 0). При этом данный канал принимает состояние “занят”. В поле счётчика MDMA_ROUND_ROBIN_COUNT регистра DMA_TC_CNT загружается значение периода P , содержащееся в поле MDMA_ROUND_ROBIN_PERIOD, и начинается выполнение передач в режиме MDMA. После выполнения передачи, соответствующей значению счётчика, равному 1, доступ автоматически без непроизводительных затрат передаётся другому потоку MDMA, и в счётчик MDMA_ROUND_ROBIN_COUNT снова загружается значение периода P , содержащееся в поле MDMA_ROUND_ROBIN_PERIOD. На данном такте, если другой поток MDMA готов к выполнению передач, он переходит в состояние “занят”. Если поток MDMA не готов к выполнению передач, передачи не выполняются, и на следующем такте он снова принимает состояние “свободен”.

Когда при работе в режиме кругового планирования активен только один поток MDMA, на каждые P циклов передач данных в режиме MDMA добавляется один цикл ожидания, что незначительно снижает пропускную способность на $1/(P+1)$ от максимального значения. Если используются оба потока MDMA, операции DMA типа “память-память” могут выполняться непрерывно с нулевыми непроизводительными затратами на чередование потоков (возникают только дополнительные такты, необходимые при изменении направления при работе с памятью, например, с чтения на запись). Путём выбора величин периода P , определяющих частоту чередования потоков MDMA, может достигаться максимальная эффективность передач.

Срочные передачи DMA

Обычно, передачи определённого периферийного устройства в режиме DMA происходят с регулярными интервалами. В общем случае, чем короче интервал, тем выше приоритет, который следует назначить периферийному устройству. Если средняя пропускная способность, задействованная всеми периферийными устройствами, составляет небольшую часть от общей пропускной способности, все требуемые запросы периферийных устройств будут удовлетворены.

Иногда мгновенный трафик DMA может превышать доступную пропускную способность, что приводит к перегрузке. Это может произойти, если выполняется временный останов памяти L1 или внешней памяти, например, для выполнения перекачки страницы SDRAM или заполнения строки кэша. Перегрузка также может возникать, если один или более каналов DMA выполняют большое количество запросов, например, для выборки дескрипторов или заполнения FIFO DMA или периферийного устройства.

Прямой доступ к памяти

Если перегрузка сохраняется, может возникнуть нехватка данных в периферийных устройствах с низким приоритетом DMA. Даже, несмотря на то, что периферийное устройство имеет низкий приоритет, если до конца регулярного интервала периферийного устройства не происходит передача данных, может произойти сбой системы. Для минимизации вероятности возникновения сбоя устройство DMA определяет периферийные устройства, которым необходимо срочно выполнить операции с данными, и предоставляет им обслуживание с наивысшим приоритетом.

Запрос канала DMA на обслуживание памяти определяется как “срочный”, если выполняются два условия:

- буфер FIFO не готов к передачам по шине DAB (то есть, пуст FIFO передачи или заполнен FIFO приёма), и
- периферийное устройство устанавливает активный сигнал на соответствующей линии запроса DMA.

Выборки дескриптора могут являться срочными, если их выполнение необходимо для инициации или продолжения цепочки передач DMA периферийного устройства, которому необходимо выполнить операцию с данными. Запросы DMA, поступающие от каналов MDMA никогда не бывают срочными.

Когда один или более каналов DMA формируют срочный запрос памяти, происходит два события:

- Приоритет всех несрочных запросов памяти понижается на 32, что гарантирует обслуживание только срочного запроса. Если имеется несколько срочных запросов, они конкурируют друг с другом; при этом учитывается направление запроса.
- Выполняемая передача, помечается как требующая срочной обработки системой памяти (L1 или внешней); кроме того все предыдущие незавершённые передачи в этой системе памяти также помечаются как требующие срочной обработки. Это может вызвать задержку ряда обращений ядра к внешней памяти на несколько тактов до удовлетворения срочного запроса периферийного устройства.

Обработка срочных передач DMA выполняется автоматически. Для выполнения этой функции не требуется управления со стороны пользователя.

Программное управление DMA

Существует несколько методов управления и синхронизации, которые могут использоваться при разработке программных модулей управления DMA и MDMA (также см. раздел “DMA типа “память-память”). Такие программные модули должны воспринимать запросы новых передач в режиме DMA от других модулей, интегрировать эти передачи в существующие очереди передач и уведомлять другие модули о завершении передач.

Прямой доступ к памяти

В процессоре имеется возможность управления каждым потоком DMA периферийного устройства или потоком MDMA при помощи отдельной задачи или совместно с любым другим потоком. Каждый канал DMA имеет независимый ортогональный набор регистров управления, ресурсов и прерываний. Таким образом, выбор схемы управления одним каналом не влияет на выбор схем управления другими каналами. Например, одно периферийное устройство может использовать работающую по прерываниям схему со связанным списком дескрипторов, а другое периферийное устройство – работающую по запросу схему с передачей одиночных буферов, синхронизируемую при помощи опроса регистра `IRQ_STATUS`.

Синхронизация программы и DMA

Критическим аспектом программного управления DMA является требование синхронизация завершения передачи буфера в режиме DMA с программой. Наилучшим образом синхронизация достигается при использовании прерываний, опроса регистра `IRQ_STATUS` или комбинации этих двух методов. Синхронизация с использованием опроса регистров адреса или счётчика даёт значительную погрешность, сравнимую с длинами конвейера.

В методах синхронизации, основанных на использовании прерываний, необходимо избегать возникновения чрезмерного количества прерываний или сбоев вызова обработчиков прерываний каналов DMA, вызванных значительными задержками при обработке прерываний. В целом, при проектировании системы необходимо поставить каждому каналу в соответствие только одно прерывание (например, по завершению списка дескрипторов), или гарантировать разнос во времени между возникновением прерываний, достаточный для завершения обслуживания каждого из них. Следует отметить, что так как каждому каналу прерывания соответствует отдельное прерывание, взаимодействием между прерываниями различных периферийных устройств значительно легче управлять.

Из-за влияния FIFO DMA и конвейера DMA/память в качестве метода точной синхронизации DMA с программой обработки данных не рекомендуется использование опроса регистров `CURR_ADDR`, `CURR_DESC_PTR` или `CURR_X/Y_COUNT`. Значения регистров текущего адреса, указателя и счётчика изменяются за несколько тактов до завершения соответствующей операции с памятью (момента, когда результаты операции станут доступны ядру процессора при помощи команд записи в память/чтения из памяти). Например, предположим, что при записи во внешнюю память в режиме DMA каналом А инициируется операция, вызывающая открытие страницы SDRAM, занимающее много тактов тактового сигнала системы. Система DMA может затем перейти к выполнению другой операции DMA каналом В, которая сама по себе не вызывает задержку, но будет остановлена медленной операцией, выполняемой каналом А. При помощи программного мониторинга на основании содержимого регистра `CURR_ADDR` канала В невозможно достоверно определить, произошла ли запись в ячейку памяти, на которую он указывает.

Прямой доступ к памяти

Принимая во внимание размер конвейера DMA/память, при помощи опроса регистров текущего адреса, указателя и счётчика можно добиться грубой синхронизации DMA и программы. Размер FIFO DMA канала DMA составляет четыре ячейки (либо четыре 8- или 16-разрядных элемента данных, либо два 32-разрядных элемента данных), размер FIFO MDMA – восемь ячеек (четыре 32-разрядных элемента данных). Контроллер DMA не обновляет регистры текущего адреса, указателя и счётчика, если указанные FIFO содержат данные, обработка которых не завершена (включая данные начатых, но незаконченных операций чтения).

Кроме того, совместная длина конвейеров интерфейсов L1 и системы DMA с внешней памятью составляет приблизительно шесть 8- или 16-разрядных элементов данных. Длина конвейеров DMA и устройства интерфейса внешней шины (EBIU) составляет приблизительно три элемента данных (при измерении от уровня, на котором результат обновления регистра DMA становится доступен операции чтения по адресу в карте памяти, до уровня, на котором выполняется строгое упорядочение доступов ядра и доступов в режиме DMA к памяти). На основании суммы размера FIFO DMA и длины конвейера DMA/память можно произвести оценку максимального числа незавершённых операций с памятью, производимых в один момент времени. (Следует отметить, что эта оценка даёт максимальное количество операций, так как конвейер DMA/память может включать трафик от других каналов DMA).

Например, предположим, что канал DMA периферийного устройства передаёт во внутреннюю память блок из 100 элементов данных, и при чтении регистра `CURR_X_COUNT` данного канала получено значение 60 (количество элементов, которые осталось передать). Таким образом, начата обработка, по меньшей мере, первых 40 элементов. Общая длина конвейера составляет не более, чем сумма четырёх (FIFO PDMA) и шести (конвейер DMA/память), то есть десять элементов данных. Таким образом, можно достоверно предположить, что передача в режиме DMA первых $40 - 10 = 30$ элементов завершена.

Для обеспечения точной синхронизации программа, вместо опроса регистров текущего адреса, указателя и счётчика, должна либо дожидаться прерывания, либо проверять значение регистра `IRQ_STATUS` канала. Когда система DMA генерирует прерывание или изменяет состояние бита `IRQ_STATUS`, гарантируется завершение последней операции с памятью для текущего блока данных и доступность её результата программе ЦСП. При чтении из памяти в режиме DMA, гарантируется приём последних данных, читаемых из памяти, в FIFO DMA. При записи в память в режиме DMA, гарантируется, получение устройством DMA подтверждения записи от L1 или EBIU.

Следующие примеры демонстрируют методы синхронизации программы с различными вариантами DMA.

Передачи отдельных буферов в режиме DMA

Синхронизация легко выполняется, если операции DMA периферийного устройства представляют собой изолированные передачи отдельных буферов.

Прямой доступ к памяти

Операции DMA инициируются программной записью в регистры управления канала. Пользователь может выбрать вариант работы с одним дескриптором в памяти, при этом программа должна выполнять запись только регистров DMA_CONFIG и NEXT_DESC_PTR. Пользователь также может выбрать вариант прямой программной записи всех регистров, отображённых в карте памяти, заканчивающейся записью в регистр DMA_CONFIG.

Простейшим способом оповещения о завершении DMA является использование прерывания. Прерывание разрешается битом DI_EN в регистре DMA_CONFIG и необходимой настройкой контроллера прерываний системы. Если использование прерываний нежелательно, программа может проверять завершение передач путём чтения регистра IRQ_STATUS и проверки бита DMA_RUN. Когда значение этого бита равно нулю, передача буфера завершена.

Непрерывные передачи в режиме автобуферизации

Если данные, передаваемые/получаемые периферийным устройством в режиме DMA, представляют собой периодический неменяющийся поток данных сигнала, эффективным может быть использование опции автобуферизации DMA (FLOW = 1). При этом выполняются передачи данных в буфер в памяти или из буфера в памяти при помощи циклической адресации с использованием одно- или двумерной индексации с нулевыми непроизводительными затратами процессора и системы DMA на организацию циклов. Возможные варианты синхронизации включают:

- Одномерный DMA с синхронизацией по прерываниям – программа прерывается по завершению передачи каждого буфера. Критичным аспектом при разработке программы является необходимость обработки первых данных буфера программой до выполнения следующей передачи в режиме DMA, способной выполнить запись или новое чтение первого элемента. Этот метод надёжен, если при проектировании системы гарантируется, что период передач данных при любых обстоятельствах превышает задержку прерывания.
- Двумерный DMA с синхронизацией по прерываниям (двойная буферизация) – буфер DMA разбивается на два или более буферов меньшего размера, и выбирается генерация прерывания (устанавливается DI_SEL = 1 в регистре DMA_CONFIG) по завершению каждого внутреннего цикла DMA. Таким способом может реализовываться традиционная схема с двойным буфером.

Например, следующими настройками задаются два буфера на 512 слов внутри буфера на 1К слов для приёма 16-разрядных данных от периферийного устройства:

```
START_ADDR = базовый адрес буфера
DMA_CONFIG = 0x10D7 (FLOW = 1, DI_EN = 1, DI_SEL = 1,
DMA2D = 1, WDSIZE = 01, WNR = 1, DMA_EN = 1)
X_COUNT = 512
```

Прямой доступ к памяти

X_MODIFY = 2 – 16 разрядные данные

Y_COUNT = 2 – два буфера

Y_MODIFY = 2, – совпадает с X_MODIFY. для непрерывности буферов

- Двумерный DMA с синхронизацией по опросу – если непроизводительные затраты на обслуживание прерывания неприемлемо высоки, и допустима грубая синхронизация при помощи опроса регистров счётчика/адреса, может использоваться схема двумерной многобуферной синхронизации. Например, предположим, что необходимо обрабатывать принимаемые данные, сгруппированные в пакеты по шестнадцать 32-разрядных элементов. Следующими настройками задаётся двумерный буфер DMA, состоящий из четырёх частей, каждая из которых может хранить один пакет данных:

START_ADDR = базовый адрес буфера

DMA_CONFIG = 0x101B (FLOW = 1, DI_EN = 0, DMA2D = 1,

WDSIZE = 10, WNR = 1, DMA_EN = 1)

X_COUNT = 16

X_MODIFY = 4 – 32-разрядные данные

Y_COUNT = 4 – четыре буфера

Y_MODIFY = 4, совпадает с X_MODIFY для непрерывности буферов

Часть программы, отвечающая за синхронизацию, может выполнять чтение регистра Y_COUNT для определения номера буфера, передача которого выполняется в текущий момент времени. При этом предполагается, что передача предыдущего буфера ещё не завершена из-за эффектов конвейера. Например, если при чтении Y_COUNT возвращается значение 3, программа предполагает, что в текущий момент времени выполняется передача буфера 3, однако часть буфера 2 может еще не быть принята. Таким образом, программа может выполнять обработку буфера 0 или 1.

- Одномерный DMA с использованием FIFO без синхронизации – если при проектировании системы гарантируется, что обработка данных периферийного устройства и скорость передач данных в режиме DMA сохраняют определённую степень корреляции, и можно пренебречь кратковременными изменениями задержки, возможно применение простого FIFO. При этом канал DMA может быть запрограммирован на работу с одномерной адресацией в режиме автобуферизации без использования прерываний или опроса регистров.

Структуры дескрипторов

Дескрипторы DMA могут использоваться для передач из памяти и в память структур данных, отличных от простых одномерных или двумерных массивов. Например, если необходимо передавать пакет данных из разных областей памяти (заголовка из одной области, полезной информации из списка блоков памяти, управляемого алгоритмов выделения областей памяти, и небольшой заключительной части, содержащей контрольную сумму), для каждой из них может быть задан отдельный дескриптор DMA. Дескрипторы могут затем быть

Прямой доступ к памяти

сгруппированы в список или массив, в соответствии со значением поля FLOW в DMA_CONFIG.

Программа может синхронизироваться с ходом передачи структуры данных при помощи настройки прерываний в одном или более дескрипторах. Например, программа может задавать генерацию прерываний в дескрипторе заголовка и дескрипторе заключительной части, но не в дескрипторах блоков полезной информации.

Важно запомнить значение различных полей элементов DMA_CONFIG дескрипторов при создании списка или массива дескрипторов. В частности:

- Младший байт DMA_CONFIG определяет тип передачи в режиме DMA, выполняемой с использованием *текущего* дескриптора (например, режим двумерного DMA с разрешёнными прерываниями).
- Старший байт DMA_CONFIG определяет формат *следующего* дескриптора цепочки. Поля NDSIZE и FLOW текущего дескриптора не соответствуют его формату; они определяют связь со следующим дескриптором, если она имеется.

С другой стороны, при перезапуске системы DMA, оба байта значения, записываемого в регистр DMA_CONFIG канала DMA, должны соответствовать текущему дескриптору. Как минимум, поля FLOW, NDSIZE, WNR и DMA_EN должны согласовываться с форматом текущего дескриптора; поля WDSIZE, DI_EN, DI_SEL, RESTART и DMA2D будут взяты из значения соответствующего элемента дескриптора при его чтении из памяти (значения полей, полученные при прямой записи в регистр, игнорируются)

Управление очередью дескрипторов

Разработчик системы может создать вспомогательную подпрограмму менеджера DMA, которая будет принимать запросы DMA от других частей программы. Программный менеджер DMA заранее не знает, какие новые запросы поступят, или что эти запросы могут содержать. Программа может управлять этими передачами при помощи циклического связанного списка дескрипторов DMA, в котором элементы NDPH и NDPL каждого дескриптора указывают на следующий дескриптор, и элементы последнего дескриптора указывают на первый.

Во фрагменте программы, выполняющем запись в этот список дескрипторов, может использоваться режим циклической адресации процессора (регистры I, L, M и B); таким образом, в нём не требуется применения операций сравнения и условных команд для управления циклической структурой. В данном случае запись элементов NDPL и NDPH может быть выполнена лишь один раз при запуске DMA; при каждой последующей записи нового содержимого дескриптора запись этих элементов может не выполняться.

Рекомендуется осуществлять синхронизацию очереди дескрипторов при помощи прерываний. Очередь дескрипторов структурируется таким образом, чтобы, по

Прямой доступ к памяти

меньшей мере, последний дескриптор всегда программировался на генерацию прерывания.

Существует два общих метода управления очередью дескрипторов при помощи прерываний:

- с прерыванием по каждому дескриптору;
- с минимальным числом прерываний – только по последнему дескриптору.

Очередь с прерыванием по каждому дескриптору

В данной системе синхронизация программного менеджера DMA с каналом DMA достигается разрешением прерывания по каждому дескриптору. Этот метод следует применять, только если при проектировании системы гарантируется раздельное обслуживание каждого прерывания (отсутствие наложения прерываний).

Для поддержки синхронизации очереди дескрипторов, часть программы, не связанная с обработкой прерывания, должна содержать счётчик числа дескрипторов, помещённых в очередь, а обработчик прерывания должен содержать счётчик числа дескрипторов, удалённых из очереди. Значения этих счётчиков совпадают только в момент паузы в работе канала DMA после завершения работы со всеми дескрипторами.

При получении каждого нового запроса, программный менеджер DMA инициализирует новый дескриптор, обеспечивая запись значения 0 в поле `FLOW` элемента `DMA_CONFIG`. Затем программа сравнивает счётчики дескрипторов для определения того, выполняются ли операции в канале DMA или нет. Если присутствует пауза в работе канала DMA (значения счётчиков совпадают), программа инкрементирует счётчик и запускает работу устройства DMA записью значения `DMA_CONFIG` нового дескриптора в регистр `DMA_CONFIG` канала DMA.

Если значения счётчиков не совпадают, программа модифицирует значение элемента `DMA_CONFIG` предпоследнего дескриптора, таким образом, чтобы его старшая половина (поля `FLOW` и `NDSIZE`) описывала новый дескриптор, помещаемый в очередь. Эта операция не нарушает работу канала DMA, при условии, что остальная часть структуры данных дескриптора была инициализирована ранее. В целях синхронизации программы с DMA, необходимо точно определить, какое из значений `DMA_CONFIG` (новое или старое) было прочитано каналом DMA.

Данную операцию синхронизации следует выполнять в обработчике прерывания. Сначала, при возникновении прерывания, обработчик должен выполнить чтение регистра `IRQ_STATUS` канала. Если бит состояния `DMA_RUN` установлен, канал переключился на обработку другого дескриптора, и в обработчике прерывания может быть выполнен инкремент счётчика дескрипторов, удалённых из очереди, и выход из обработчика. Если бит состояния `DMA_RUN` не установлен, присутствует пауза в работе канала, вызванная либо отсутствием дескрипторов, которые должны быть обработаны, либо тем, что последний дескриптор был помещён в

Прямой доступ к памяти

очередь слишком поздно (то есть, модификация элемента DMA_CONFIG предпоследнего дескриптора произошла после того, как этот элемент был прочитан контроллером DMA). В этом случае, в обработчике прерывания должны выполняться запись в регистр DMA_CONFIG канала DMA значения элемента DMA_CONFIG, соответствующего последнему дескриптору, инкремент счётчика дескрипторов, удалённых из очереди, и выход из обработчика.

Если задержки прерываний системы имеют значения, достаточные для пропуска любого из прерываний каналов DMA, возможен сбой в работе данного метода управления очередью дескрипторов. Обработчик прерываний, способный синхронизировать прерывания нескольких дескрипторов, представляет собой сложную программу, выполняющую несколько обращений к регистрам, отображенным в карте памяти для поддержания устойчивой работы. В системах, имеющих большие задержки прерываний, предпочтительным является метод синхронизации с минимальным количеством прерываний.

Очередь дескрипторов с минимальным количеством прерываний

В данной системе возможна генерация только одного прерывания DMA в любой момент времени. Обработчик прерывания DMA в такой системе может представлять собой программу очень малого размера. В данном режиме очередь дескрипторов разделяется на “активную” и “ожидающую” части; при этом прерывания разрешены только по последнему дескриптору в каждой из частей.

При обработке каждого нового запроса DMA, часть программы, не связанная с обработкой прерывания, заполняет новый дескриптор содержимым и добавляет его в ожидающую часть очереди. Значение поля FLOW элемента DMA_CONFIG дескриптора должно равняться нулю. Если до генерации прерывания DMA по завершению обработки очереди выполняется более одного запроса, часть программы, не связанная с обработкой прерывания, помещает в очередь последующие дескрипторы. Таким образом, формируется ожидающая часть очереди, отделенная от активной части, обрабатываемой устройством DMA. Другими словами, значение поля FLOW всех дескрипторов в ожидающей части, за исключением последнего, больше либо равно четырём, и их биты разрешения прерывания (DI_EN) сброшены. Последний дескриптор в ожидающей части содержит в поле FLOW значение 0 и его бит разрешения прерывания установлен. При этом гарантируется, что устройство DMA может автоматически обработать всю активную часть очереди и вызвать затем одно прерывание. Также при этом достигается возможность начала обработки ожидающей части очереди без использования обработчика прерывания, однократной записью в регистр DMA_CONFIG.

После помещения в очередь нового ожидающего дескриптора часть программы, не связанная с обработкой прерывания, должна оставить сообщение в ячейке памяти обмена сообщений для обработчика прерывания, содержащее значение DMA_CONFIG, используемое для начала обработки первого дескриптора в ожидающей части очереди (или 0, что означает отсутствие ожидающих дескрипторов).

Прямой доступ к памяти

Критическим моментом является то, что, если не выполняются тщательные меры синхронизации, программа не модифицирует содержимое очереди активных дескрипторов напрямую после начала ее обработки устройством DMA. В наиболее простой реализации этого метода программный менеджер DMA никогда не модифицирует дескрипторы в активной очереди; вместо этого он ожидает появления прерывания DMA, указывающего на завершение обработки всей активной очереди.

По прерыванию завершения обработки очереди обработчик прерывания выполняет чтение сообщения от части программы, не связанной с обработкой прерывания и записывает значение, содержащееся в нём, в регистр DMA_CONFIG канала DMA. При этой однократной записи в регистр выполняется новый запуск обработки очереди, и ожидающая очередь преобразовывается в активную. Затем обработчик прерывания должен передать сообщение для части программы, не связанной с обработкой прерывания, в котором указывается адрес последнего дескриптора, помещенного в активную очередь. С другой стороны, если при чтении сообщения обработчиком прерывания возвращается нулевое значение величины DMA_CONFIG, указывающее на отсутствие операций, которые необходимо выполнять, он должен передать обратно в часть программы, не связанной с обработкой прерывания, соответствующее сообщение (например, нулевое значение), указывающее на остановку обработки очереди. Этот простой обработчик может состоять из очень малого числа команд.

Часть программы, не связанная с обработкой прерывания, которая воспринимает новые запросы работы в режиме DMA, должна синхронизировать активацию новой операции DMA с обработчиком прерывания. При остановке очереди (т.е., когда в ячейке сообщений от обработчика содержится нулевое значение), за запуск обработки очереди (запись значения DMA_CONFIG первого дескриптора в регистр DMA_CONFIG канала) ответственна часть программы, не связанная с обработкой прерывания. Если остановка очереди не выполняется, часть программы, не связанная с обработкой прерывания, не должна выполнять запись в регистр DMA_CONFIG (так как это приведет к ошибке DMA); вместо этого она должна поместить дескриптор в ожидающую очередь и обновить содержимое ячейки сообщений для обработчика прерывания.

Ошибки (сбои) DMA

Контроллер DMA устанавливает флаги условий, вызывающих аномальное окончание процесса DMA (т.е. сбой). Это свойство используется при проектировании и отладке системы как способ обнаружения ошибок, вызванных неправильным программированием DMA. Ошибки (сбои) DMA обнаруживаются модулем канала DMA в случаях, перечисленных ниже. При обнаружении ошибки DMA канал незамедлительно останавливает выполнение операций (бит DMA_RUN устанавливается в 0), и удаляются любые данные, полученные в процессе предвыборки. Кроме того, устанавливается прерывание DMA_ERROR.

Прямой доступ к памяти

В контроллере DMA имеется единственное прерывание DMA_ERROR, устанавливаемое всякий раз, когда в любом из каналов обнаруживается условие ошибки.

Обработчик прерывания DMA_ERROR должен выполнять следующие операции для каждого канала:

- чтение регистра IRQ_STATUS каждого канала, для определения канала, в котором установлен бит DMA_ERR (равен 1).
- Исправление ошибки в этом канале (например, изменение значений регистров).
- Сброс бита DMA_ERR (запись в бит 1 регистра IRQ_STATUS значение 1).

Аппаратные средства DMA обнаруживают следующие условия ошибок, приводящие к возникновению прерывания ошибки DMA:

- Недопустимое значение в регистре конфигурации:
 - Некорректное значение поля WDSIZE (WDSIZE = b#11);
 - Бит 15 не установлен в 0;
 - Некорректное значение поля FLOW (FLOW = 2,3 или 5);
 - Значение поля NDSIZE не согласуется со значением поля FLOW. См. таблицу 9-20.
- Недопустимая запись в регистр при работе канала. Когда DMA_RUN = 1, возможна запись только в регистры DMA_CONFIG и IRQ_STATUS.
- Ошибка выравнивания адреса при любом обращении к памяти. Например, поле WDSIZE регистра DMA_CONFIG равно 1 (операции с 16-разрядными словами), а младший бит адреса не равен нулю, или WDSIZE = 2 (операции с 32-разрядными словами), а два младших бита адреса не равны нулю.
- Попытка перехода границы пространств памяти (из внутренней во внешнюю, или наоборот). Например, пересечение значением регистра текущего адреса DMA (CURR_ADDR) границы 0xF000 0000 или пересечение значением регистра указателя текущего дескриптора (CURR_DESC_PTR) границы 0xF000 0000.
- Ошибка обращения к памяти. Возникает либо при попытке обращения к адресу во внутренней памяти, соответствующей зарезервированной области или области, определенной как кэш, либо при ошибке, вызванной обращением к внешней памяти (сигнал ошибки поступает от интерфейса внешней памяти).

Некоторые запрещенные ситуации не обнаруживаются аппаратными средствами DMA. Их возникновение этих ситуаций не вызывает прерываний ошибки DMA:

- Значение бита направления (WNR) в регистре DMA_CONFIG не согласуется с направлением передач соответствующего периферийного устройства.
- Значение бита направления в регистре DMA_CONFIG не согласуется с направлением канала MDMA.
- Значение размера слова, задаваемое полем WDSIZE регистра DMA_CONFIG, не поддерживается соответствующим периферийным устройством.
- Значения полей WDSIZE регистров DMA_CONFIG источника и приемника потока MDMA не совпадают.
- Цепочка дескрипторов указывает на буферы данных, расположенные в разных пространствах памяти (внутренней и внешней).
- X_COUNT = 1 при двумерном DMA.

Прямой доступ к памяти

Таблица 9-20. Допустимые значения NDSIZE

FLOW	NDSIZE	Примечания
0	0	
1	0	
4	$0 < \text{NDSIZE} \leq 7$	Режим массива дескрипторов, выборка указателя дескриптора не выполняется
6	$0 < \text{NDSIZE} \leq 8$	Режим списка дескрипторов, выполняется выборка 16-разрядного указателя дескриптора
7	$0 < \text{NDSIZE} \leq 9$	Режим списка дескрипторов, выполняется выборка 32-разрядного указателя дескриптора