STM32F429: контроллер DMA

Добавил(а) microsin

Прямой доступ к памяти (Direct Memory Access, **DMA**) используется для быстрой передачи данных между периферийными устройствами микроконтроллера и памятью (периферия-память) и между областями памяти (память-память). Данные можно быстро перемещать под управлением DMA, не нагружая этой операцией CPU, что освобождает его для других полезных действий (выполнение кода). Непонятные термины и сокращения см. в Словарике [3].

В контроллере DMA скомбинирована мощная архитектура двойной master-шины AHB с независимым стеком FIFO, чтобы оптимизировать полосу пропускания системы, на основе комплексной матричной архитектуры шины.

Два контроллера DMA всего поддерживают 16 потоков (8 для каждого контроллера), каждый предназначен для обработки доступа к памяти от одного или большего количества периферийных устройств. Каждый поток может иметь всего до 8 каналов (запросов). И у каждого есть арбитр для обработки приоритета между запросами DMA.

[**Функции DMA**]

• Архитектура с двойной master-шиной AHB, одна для доступа к памяти, другая для доступа к периферийным устройства микроконтроллера.

• Интерфейс программирования AHB slave, поддерживающий только 32-битный доступ.

• 8 потоков (stream) для каждого контроллера DMA, обрабатывающих до 8 каналов (запросов) на поток.

• Буферы памяти FIFO глубиной 32 ячейки на поток, которые можно использовать в режиме FIFO или direct-режиме:

  – FIFO-режим: с программно выбираемым порогом: 1/4, 1/2 или 3/4 от размера FIFO.
  – Direct-режим. Каждый запрос DMA немедленно инициирует передачу из памяти или в память. Когда сконфигурирован direct-режим (FIFO запрещен) для передачи данных память-периферия, DMA предварительно загружает только одну порцию информации из памяти во внутренний FIFO, чтобы обеспечить немедленную передачу данных, как только запрос DMA инициируется периферийным устройством.

• Каждый поток может быть сконфигурирован аппаратурой, чтобы представлять собой:

  – regular-канал, который поддерживает передачи периферия-память, память-периферия и память-память.
  – канал с двойным буферизированием, который также поддерживает двойную буферизацию на стороне памяти.

• Каждый из 8 потоков соединен с выделенными аппаратными каналами DMA (запросами).

• Приоритеты между запросами потока DMA выбираются программно (по 4 уровням very high, high, medium, low), или аппаратно в случае эквивалентного уровня приоритета (запрос 0 имеет приоритет перед запросом 1, и так далее).

• Каждый поток также поддерживает программный триггер (инициатор) для передач память-память (это доступно только для контроллера DMA2).

• Может быть выбран каждый запрос на потоке, всего может быть до 8 запросов канала. Этот выбор конфигурируется программно, и позволяет нескольким периферийным устройствам инициировать запросы DMA.

• Количеством передаваемых элементов данных можно управлять либо от контроллера DMA, либо от периферийного устройства:

  – Управление потоком DMA: количество передаваемых элементов данных выбирается программно в диапазоне 1 .. 65535.
  – Управление потоком от периферийного устройства: количество передаваемых элементов данных неизвестно, и управляется от источника или от периферийного устройства, который аппаратно сигнализирует об окончании передачи.

• Независимая ширина передачи источника и места назначения (байт, полуслово, слово): когда ширина данных источника и места назначения не одинаковые, DMA автоматически упаковывает/распаковывает необходимые передачи, чтобы оптимизировать полосу пропускания. Эта функция доступна только в режиме FIFO.

• Адресация с инкрементом или без инкремента для источника и места назначения данных.

• Поддерживаются инкрементальные burst-передачи 4, 8 или 16 битов. Размер burst конфигурируется программно, обычно для половины размера FIFO периферийного устройства.

• Каждый поток поддерживает управление кольцевым буфером.

• 5 флагов событий (DMA Half Transfer, DMA Transfer complete, DMA Transfer Error, DMA FIFO Error, Direct Mode Error) обычно соединяемых операцией ИЛИ для генерации одного запроса прерывания для каждого потока.

[**Функциональное описание DMA**]

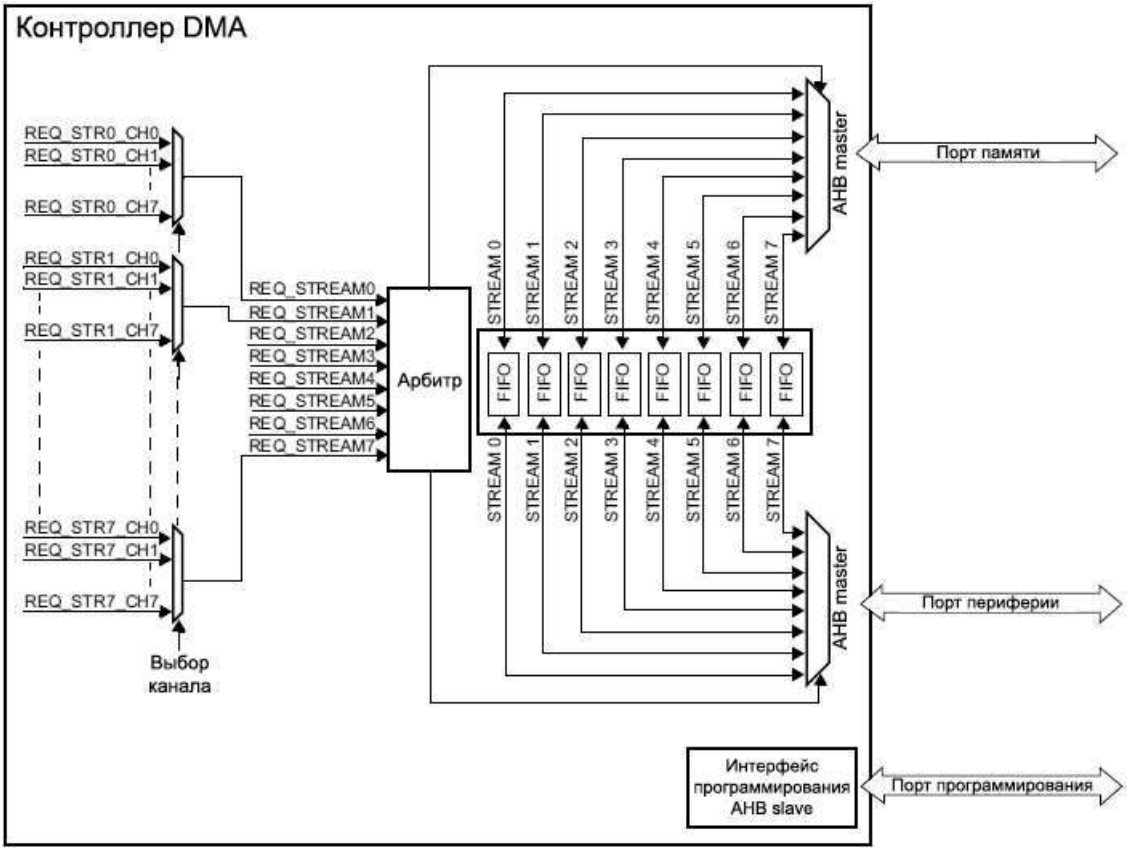


Рис. 32. Блок-схема DMA.

Контроллер DMA выполняет прямые передачи по памяти: как AHB master он берет управление матрицей шины AHB для инициализации транзакций AHB. Он может выполнять следующие транзакции:

• периферия-память
• память-периферия
• память-память

Контроллер DMA предоставляет два master-порта AHB: порт памяти AHB, предназначенный для подключения к областям памяти и порт периферии AHB, предназначенный для подключения к периферийным устройствам микроконтроллера. Однако, чтобы реализовать передачи память-память, порт периферии AHB также должен иметь доступ к областям памяти.

AHB slave-порт используется для программирования контроллера DMA (он поддерживает только 32-битный доступ).

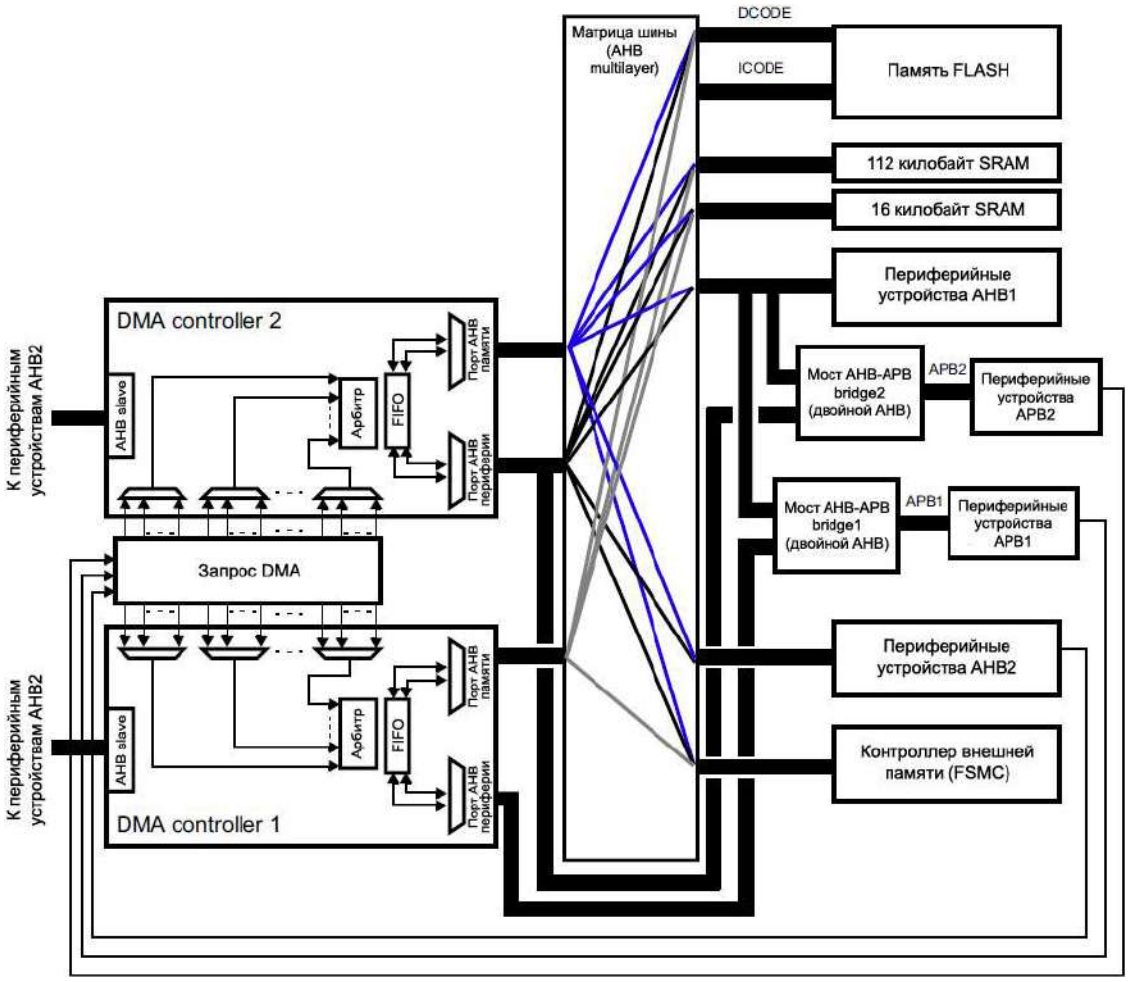


Рис. 33. Реализация системы двух контроллеров DMA (STM32F405xx/07xx и STM32F415xx/17xx).

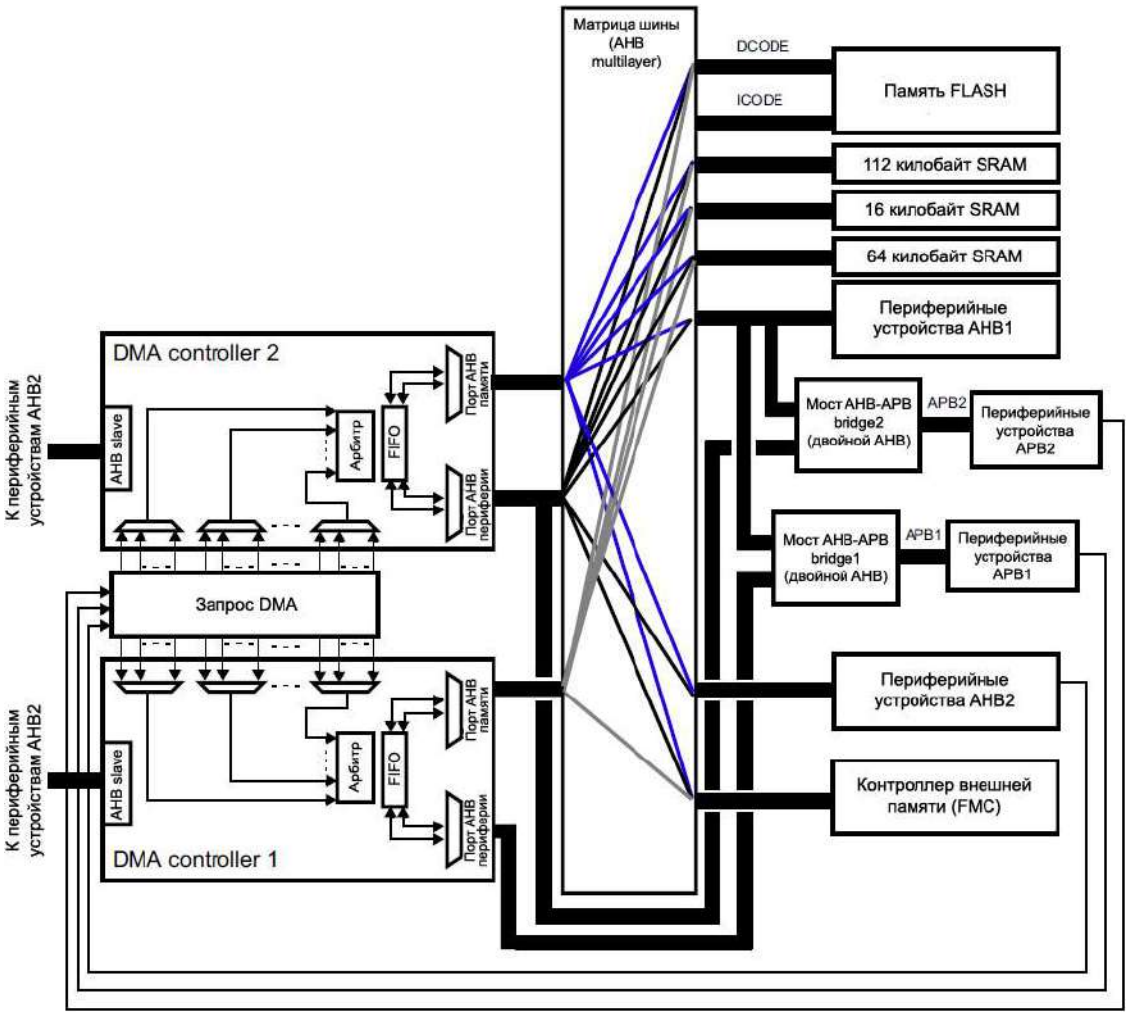


Рис. 34. Реализация системы двух контроллеров DMA (STM32F42xxx и STM32F43xxx).

Примечание: AHB порт периферии контроллера DMA1 не подключен к матрице шины, как контроллер DMA2. В результате только потоки DMA2 могут выполнять передачи память-память.

Транзакции DMA. Транзакция состоит из последовательности указанного количества передач данных. Количество передаваемых элементов данных и их ширина (8, 16 или 32 бит) выбирается программно.

Каждая передача DMA состоит из трех операций:

• Загрузка из регистра данных периферии или ячейки памяти, адресованных регистрами DMA_SxPAR или DMA_SxM0AR.
• Сохранение данных, загруженных из регистра данных периферии или ячейки памяти, адресованных регистрами DMA_SxPAR или DMA_SxM0AR.
• Пост-декремент регистра DMA_SxNDTR, который содержит количество транзакций, которые еще должны быть выполнены.

После события периферия отправляет сигнал запроса контроллеру DMA. Контроллер DMA обрабатывает запрос в зависимости от приоритетов канала. Как только контроллер DMA осуществил доступ к периферийному устройству, контроллером DMA отправляется сигнал подтверждения (Acknowledge). Периферийное устройство снимает свой запрос, как только получит сигнал Acknowledge от контроллера DMA. Как только запрос снят периферией, контроллер DMA освобождает сигнал Acknowledge. Если больше нет запросов, периферия может инициировать следующую транзакцию.

Выбор канала. Каждый поток может быть ассоциирован с запросом DMA, с выбором до 8 возможных запросов канала. Выбор управляется битами CHSEL[2:0] регистра DMA_SxCR.

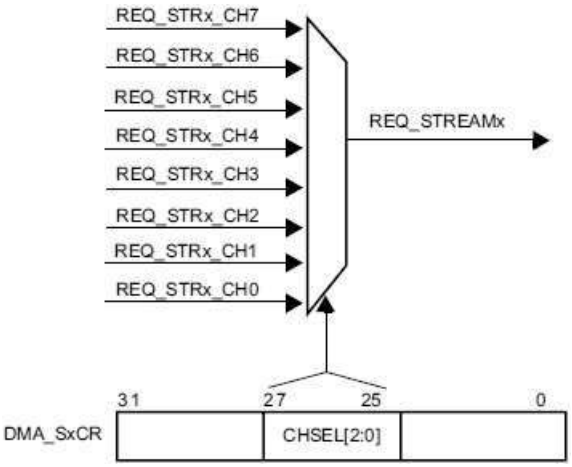


Рис. 35. Выбор канала.

8 запросов от периферийных устройств (TIM, ADC, SPI, I2C и т. д.) независимо подключаются к каждому каналу, и их соединение зависит от реализации продукта. См. следующие таблицы для примеров отображений запросов DMA.

Таблица 42. Отображение запроса DMA1.

Запросы периферии	Поток 0	Поток 1	Поток 2	Поток 3	Поток 4	Поток 5	Поток 6	Поток 7
Канал 0	SPI3_RX	-	SPI3_RX	SPI2_RX	SPI2_TX	SPI3_TX	-	SPI3_TX
Канал 1	I2C1_RX	-	TIM7_UP	-	TIM7_UP	I2C1_RX	I2C1_TX	I2C1_TX
Канал 2	TIM4_CH1	-	I2S3_EXT_RX	TIM4_CH2	I2S2_EXT_TX	I2S3_EXT_TX	TIM4_UP	TIM4_CH3
Канал 3	I2S3_EXT_RX	TIM2_UP TIM2_CH3	I2C3_RX	I2S2_EXT_RX	I2C3_TX	TIM2_CH1	TIM2_CH2 TIM2_CH4	TIM2_UP TIM2_CH4
Канал 4	UART5_RX	USART3_RX	UART4_RX	USART3_TX	UART4_TX	USART2_RX	USART2_TX	UART5_TX
Канал 5	UART8_TX(1)	UART7_TX(1)	TIM3_CH4 TIM3_UP	UART7_RX(1)	TIM3_CH1 TIM3_TRIG	TIM3_CH2	UART8_RX(1)	TIM3_CH3
Канал 6	TIM5_CH3 TIM5_UP	TIM5_CH4 TIM5_TRIG	TIM5_CH1	TIM5_CH4 TIM5_TRIG	TIM5_CH2	-	TIM5_UP	-
Канал 7	-	TIM6_UP	I2C2_RX	I2C2_RX	USART3_TX	DAC1	DAC2	I2C2_TX

Примечание (1): эти запросы доступны только на STM32F42xxx и STM32F43xxx.

Таблица 43. Отображение запроса DMA2.

Запросы периферии	Поток 0	Поток 1	Поток 2	Поток 3	Поток 4	Поток 5	Поток 6	Поток 7
Канал 0	ADC1	SAI1_A(1)	TIM8_CH1 TIM8_CH2 TIM8_CH3	SAI1_A(1)	ADC1	SAI1_B(1)	TIM1_CH1 TIM1_CH2 TIM1_CH3	-

Канал 1	-	DCMI	ADC2	ADC2	SAI1_B[1]	SPI6_TX[1]	SPI6_RX[1]	DCMI
Канал 2	ADC3	ADC3	-	SPI5_RX[1]	SPI5_TX[1]	CRYP_OUT	CRYP_IN	HASH_IN
Канал 3	SPI1_RX	-	SPI1_RX	SPI1_TX	-	SPI1_TX	-	-
Канал 4	SPI4_RX[1]	SPI4_TX[1]	USART1_RX	SDIO	-	USART1_RX	SDIO	USART1_TX
Канал 5	-	USART6_RX	USART6_RX	SPI4_RX[1]	SPI4_TX[1]	-	USART6_TX	USART6_TX
Канал 6	TIM1_TRIG	TIM1_CH1	TIM1_CH2	TIM1_CH1	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	-
Канал 7	-	TIM8_UP	TIM8_CH1	TIM8_CH2	TIM8_CH3	SPI5_RX[1]	SPI5_TX[1]	TIM8_CH4 TIM8_TRIG TIM8_COM

Примечание (1): эти запросы доступны только на STM32F42xxx и STM32F43xxx.

Арбитр. Арбитр управляет 8-ю запросами потока DMA, основываясь на их приоритете для каждого из двух master-портов AHB (порта памяти и порта периферии), и запускает последовательности доступа периферии/памяти.

Приоритеты обслуживаются в двух стадиях:

• Программно: приоритет каждого потока может быть сконфигурирован регистром DMA_SxCR. Имеется 4 уровня приоритета:

– Very high
– High
– Medium
– Low

• Аппаратно: если у двух запросов одинаковый уровень программного приоритета, то поток с меньшим номером получает приоритет над потоком с большим номером. Например, у Stream 2 приоритет выше, чем у Stream 4.

Потоки DMA. Каждый из 8 потоков контроллера DMA предоставляет однонаправленную передачу между источником (source) и местом назначения (destination) данных.

Каждый поток можно сконфигурировать для выполнения:

• Обычных транзакций: память-периферия, периферия-память и память-память.
• Транзакций с двойной буферизацией: передачи используют два указателя на память для памяти (в то время как DMA читает/записывает из одного буфера или в него, приложение может записывать/читать в другой буфер или из него).

Количество передаваемых данных (до 65535) программируется, и связано с шириной источника периферии, которая запрашивает передачу DMA, подключенную к AHB порту периферии. Регистр, который содержит количество передаваемых элементов данных, декрементируется после каждой транзакции.

Источник, место назначения и режимы передачи. И источник (source), и место назначения (destination) передачи может адресовать периферийные устройства и области памяти во всем адресном пространстве 4 гигабайта, по адресам от 0x00000000 до 0xFFFFFFFF.

Направление передач конфигурируется битами DIR[1:0] регистра DMA_SxCR: память-периферия, периферия-память или память-память. Таблица 44 показывает регистры, соответствующие адресу источника (source) и адресу места назначения (destination) данных.

Таблица 44. Адреса источника и места назначения данных и destination.

DIR[1:0][1]	**Направление**	**Адрес источника**	**Адрес назначения**
00	Периферия - память	DMA_SxPAR	DMA_SxM0AR
01	Память - периферия	DMA_SxM0AR	DMA_SxPAR
10	Память - память	DMA_SxPAR	DMA_SxM0AR
11	зарезервировано	-	-

Примечание (1): DIR[1:0] это поле бит направления данных в регистре DMA_SxCR.

Когда ширина данных (программируемая полем бит PSIZE или MSIZE регистра DMA_SxCR) составляет половину слова (16 бит) или слово (32 бита) соответственно, то адрес периферии или адрес памяти, записанный в регистры DMA_SxPAR или DMA_SxM0AR/DMA_SxM1AR, должен быть выровнен на адрес полуслова или слова соответственно.

Режим периферия-память

Режим память-периферия

Режим память-память

Инкремент указателя. Указатели на периферию и память могут опционально автоматически инкрементироваться после транзакции (post-increment) или оставаться постоянными после каждой передачи, в зависимости от бит PINC и MINC регистра DMA_SxCR.

Запрет режима инкремента полезен, когда данные источника периферии или данные назначения находятся в одном регистре.

Если режим инкремента разрешен, то следующая передача будет использовать предыдущий адрес, инкрементированный на 1 (для байт), 2 (для полуслов) или 4 (для слов) в зависимости от данных, запрограммированных в битах PSIZE или MSIZE регистра DMA_SxCR.

Чтобы оптимизировать операцию упаковки, можно фиксировать размер смещения приращения для периферийного адреса независимо от размера данных, передаваемых по периферийному порту AHB. Бит PINCOS в регистре DMA_SxCR используется для выравнивания размера смещения приращения на размер данных периферийного порта AHB, или нв 32-битный адрес (тогда адрес инкрементируется на 4). Бит PINCOS влияет только на периферийный порт AHB.

Если бит PINCOS установлен, адрес следующей передачи это адрес предыдущей передачи, инкрементированный на 4 (автоматически выровненный на 32-битный адрес), независимо от размера PSIZE. Однако на порт памяти AHB эта операция не влияет.

Режим кольцевого буфера (circular mode). Этот режим режим доступен для обработки кольцевых (circular) буферов и непрерывных потоков данных (например, режим сканирования ADC). Эта функция может быть разрешена с помощью бита CIRC в регистре DMA_SxCR.

Когда активен circular-режим, количество элементов данных для передачи перезагружается начальным значением, программируемым во время фазы конфигурации потока, и запросы DMA продолжают обрабатываться.

Примечание: в circular-режиме важно следовать следующему правилу в случае burst-режима, сконфигурированного для памяти:

DMA_SxNDTR = значение, нацело делящееся на ((Mburst beat) x (Msize)/(Psize)), где:

– (Mburst beat) = 4, 8 или 16 (в зависимости от бит MBURST в регистре DMA_SxCR)
– ((Msize)/(Psize)) = 1, 2, 4, 1/2 или 1/4 (Msize и Psize представлены битами MSIZE и PSIZE в регистре DMA_SxCR. Они зависят от байтов)
– DMA_SxNDTR = количество элементов данных для передаче на порту периферии AHB.

Например: Mburst beat = 8 (INCR8), MSIZE = '00' (байт) и PSIZE = 01 (полуслово), в этом случае DMA_SxNDTR должно нацело делиться на (8 x 1/2 = 4).

Если не следовать этой формуле, то корректное поведение DMA и целостность данных не гарантируется.

NDTR также должно нацело делиться на размер пакета периферии (peripheral burst size), умноженный на размер данных периферии, иначе это приведет к неправильной работе DMA.

Режим двойного буфера. Этот режим доступен для всех потоков DMA1 и DMA2. Режим двойного буфера разрешается установкой бита DBM в регистре DMA_SxCR.

Поток двойного буфера работает как обычный поток (с одним буфером) с тем отличием, что здесь есть два указателя на память. Когда разрешен режим двойного буфера, автоматически разрешается кольцевой режим (бит CIRC в регистре DMA_SxCR не имеет значения) и на каждом завершении транзакции указатели на память переключаются (меняются местами).

В этом режиме контроллер DMA переключается от одной цели в памяти к другой по окончании транзакции DMA. Это дает возможность обрабатывать программе один буфер, пока другой буфер заполняется (или считывается, в зависимости от направления транзакции) передачей DMA. Поток двойного буфера может работать в двух направлениях (память может быть либо источником, либо местом назначения данных), как это описано в таблице 45 ниже.

Примечание: когда поток разрешен, в режиме двойного буфера можно обновлять базовый адрес для порта памяти AHB на лету (DMA_SxM0AR или DMA_SxM1AR), если выполнять следующие условия:

• Когда бит CT равен 0 в регистре DMA_SxCR, можно записывать регистр DMA_SxM1AR. Попытка записать в этот регистр, когда CT = 1, установит флаг ошибки (TEIF), и поток будет автоматически запрещен.
• Когда бит CT равен 1 в регистре DMA_SxCR, можно записывать регистр DMA_SxM0AR. Попытка записать в этот регистр, когда CT = 0, установит флаг ошибки (TEIF), и поток будет автоматически запрещен.

Чтобы избежать ситуации ошибки, рекомендуется менять базовый адрес, как только был выставлен флаг TCIF, потому что в этот момент целевой адрес должен поменяться от области памяти 0 на область памяти 1, или от 1 на 0, в зависимости от значения бита CT регистра DMA_SxCR. Само собой, изменение базового адреса должно выполняться с соблюдением двух вышеупомянутых условий.

Для всех других режимов (кроме режима двойного буфера), регистры адреса памяти защищены от записи, пока разрешен поток DMA.

Таблица 45. Регистры адреса источника (source) и места назначения (destination) данных в режиме двойного буфера (DBM=1).

DIR[1:0][1]	Направление	Адрес источника	Адрес назначения
00	Периферия - память	DMA_SxPAR	DMA_SxM0AR / DMA_SxM1AR
01	Память - периферия	DMA_SxM0AR / DMA_SxM1AR	DMA_SxPAR
10	Не допускается[2]		

11	зарезервировано	-	-

Примечания:

(1) DIR[1:0] это поле бит направления данных в регистре DMA_SxCR.
(2) Когда разрешен режим двойного буфера, автоматически разрешается circular режим. Поскольку режим память-память не совместим с режимом circular, то когда разрешен режим двойного буфера, нельзя конфигурировать режим память-память.

Программируемая ширина данных, упаковка/распаковка, порядок следования байт (endianess). Количество элементов данных для передачи должно быть запрограммировано в DMA_SxNDTR (количество элементов данных передачи, number of data items to transfer, NDT) перед тем, как разрешать поток (кроме ситуации, когда потоком управляет периферийное устройство, установлен бит PFCTRL в регистре DMA_SxCR).

Когда используется внутренний FIFO, ширина данных источника и места назначения программируется битовыми полями PSIZE и MSIZE регистра DMA_SxCR (ширина может быть 8, 16 или 32 бита).

Когда PSIZE и MSIZE не равны:

• Ширина данных количества передаваемых элементов, сконфигурированных в регистре DMA_SxNDTR, равна ширине периферийной шины (сконфигурированной битами PSIZE в регистре DMA_SxCR). Например, в случае передач периферия-память, память-периферия или память-память, и если биты PSIZE[1:0] сконфигурированы для полуслова, количество передаваемых байт равно 2 x NDT.
• Контроллер DMA копирует только с little-endian адресацией для источника и места назначения данных. Это описано в таблице 46 ниже.

Таблица 46. Поведение упаковки/распаковки и endian (бит PINC = MINC = 1).

Ширина порта AHB памяти	**Ширина порта AHB периферии**	**Количество элементов данных для передачи (NDT)**	**Номер передачи памяти**	**Адрес порта памяти / прохождение байт**	**Номер передачи периферии**	**Адрес порта периферии / прохождение байт**	
						PINCOS = 1	**PINCOS = 0**
8	8	4	1 2 3 4	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]	1 2 3 4	0x0 / B0[7:0] 0x4 / B1[7:0] 0x8 / B2[7:0] 0xC / B3[7:0]	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]
8	16	2	1 2 3 4	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]	1 2	0x0 / B1|B0[15:0] 0x4 / B3|B2[15:0]	0x0 / B1|B0[15:0] 0x2 / B3|B2[15:0]
8	31	1	1 2 3 4	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]	1	0x0 / B3|B2|B1|B0[31:0]	0x0 / B3|B2|B1|B0[31:0]
16	8	4	1 2	0x0 / B1|B0[15:0] 0x2 / B3|B2[15:0]	1 2 3 4	0x0 / B0[7:0] 0x4 / B1[7:0] 0x8 / B2[7:0] 0xC / B3[7:0]	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]
16	16	2	1 2	0x0 / B1|B0[15:0] 0x2 / B1|B0[15:0]	1 2	0x0 / B1|B0[15:0] 0x4 / B3|B2[15:0]	0x0 / B1|B0[15:0] 0x2 / B3|B2[15:0]
16	32	1	1 2	0x0 / B1|B0[15:0] 0x2 / B3|B2[15:0]	1	0x0 / B3|B2|B1|B0[31:0]	0x0 / B3|B2|B1|B0[31:0]
32	8	4	1	0x0 / B3|B2|B1|B0[31:0]	1 2 3 4	0x0 / B0[7:0] 0x4 / B1[7:0] 0x8 / B2[7:0] 0xC / B3[7:0]	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]
32	16	2	1	0x0 /B3|B2|B1|B0[31:0]	1 2	0x0 / B1|B0[15:0] 0x4 / B3|B2[15:0]	0x0 / B1|B0[15:0] 0x2 / B3|B2[15:0]
32	32	1	1	0x0 /B3|B2|B1|B0 [31:0]	1	0x0 /B3|B2|B1|B0[31:0]	0x0 / B3|B2|B1|B0[31:0]

Примечание: порт периферии может быть источником или местом назначения данных (он также может быть источником в памяти в случае передачи память-память).

Эта процедура упаковки/распаковки может вводить риск повреждения данных, когда операция была прервана до полной упаковки/распаковки. Таким образом, чтобы гарантировать когерентность данных, поток может быть сконфигурирован для burst-передач: в этом случае, каждая группа передач, принадлежащая burst, является неделимой (см. далее секцию "Передачи single и burst").

В direct-режиме (DMDIS = 0 в регистре DMA_SxFCR), упаковка/распаковка данных невозможна. В этом случае не дозволяется иметь разную ширину данных для источника и места назначения: обе ширины одинаковые, и ширина определяется битами PSIZE регистра DMA_SxCR (биты MSIZE не имеют значения).

Поля PSIZE, MSIZE и NDT[15:0] должны конфигурироваться так, чтобы гарантировать, что последняя передача не была незавершенной. Это может произойти, когда ширина данных порта периферии (биты PSIZE) меньше, чем ширина данных порта памяти (биты MSIZE). Это ограничение обобщено в таблице 47.

Таблица 47. Ограничение на NDT по отношению к PSIZE и MSIZE.

PSIZE[1:0] регистра DMA_SxCR	MSIZE[1:0] регистра DMA_SxCR	NDT[15:0] регистра DMA_SxNDTR
00 (8 бит)	01 (16 бит)	Должно нацело делиться на 2
00 (8 бит)	10 (32 бита)	Должно нацело делиться на 4
01 (16 бит)	10 (32 бита)	Должно нацело делиться на 2

Передачи single и burst. Контроллер DMA может генерировать одиночные передачи (single transfer) или инкрементальные burst-передачи 4, 8 или 16 beat-пакетов.

Размер burst конфигурируется программой независимо для двух портов AHB путем использования бит MBURST[1:0] и PBURST[1:0] регистра DMA_SxCR.

Размер burst показывает количество beat-ов в burst, а не количество передаваемых байт.

Чтобы гарантировать когерентность данных, каждая группа передач, которая формирует burst, является неделимой: передачи AHB блокируются и арбитр матрицы шины AHB не дает оказывать влияния на DMA master во время последовательности burst-передачи.

В зависимости от конфигурации single или burst, каждый запрос DMA инициирует разное количество передач на порту периферии AHB:

• Когда порт периферии AHB сконфигурирован для single передач, каждый запрос DMA генерирует передачу данных байта, полуслова или слова, в зависимости от бит PSIZE[1:0] в регистре DMA_SxCR.
• Когда порт периферии AHB сконфигурирован для burst, каждый запрос DMA генерирует 4, 8 или 16 передач beat-пакетов байт, полуслов или слов, в зависимости от бит PBURST[1:0] и PSIZE[1:0] в регистре DMA_SxCR.

То же самое, что было упомянуто выше, должно учитываться и для порта памяти AHB с битами MBURST и MSIZE.

В direct-режиме поток может генерировать только single-передачи и биты MBURST[1:0] и PBURST[1:0] принудительно устанавливаются аппаратно.

Указатели адреса (регистры DMA_SxPAR или DMA_SxM0AR) должны быть выбраны так, чтобы гарантировать, что все передачи в burst-блоке выравнены на границу, равную размеру передачи.

Конфигурация burst должна быть выбрана, чтобы соответствовать протоколу AHB, где пакеты burst не должны пересекать границы адреса 1 килобайт, потому что минимальное адресное пространство, которое можно выделить для single slave, составляет 1 килобайт. Это означает, что граница адреса 1 килобайт не должна пересекаться burst-передачей блока, иначе будет сгенерирована ошибка AHB, о которой не сообщается регистрами DMA.

FIFO

Завершение передачи DMA. Разные события могут генерировать завершение передачи путем установки бита TCIFx в регистре статуса DMA_LISR или DMA_HISR:

• В режиме управления потоком от DMA:

– Счетчик DMA_SxNDTR достиг нуля в режиме память-периферия.
– Поток сконфигурирован перед завершением передачи (очисткой бита EN в регистре DMA_SxCR) и (при передачах периферия-память или память-память) все оставшиеся данные сбрасываются (flush) из FIFO в память.

• Режим управления потоком от периферии:

– Последний внешний запрос burst или одиночный запрос был сгенерирован от периферии и (когда DMA работает в режиме периферия-память) оставшиеся данные переданы из FIFO в память.
– Поток запрещен программно, и (когда DMA работает в режиме периферия-память) оставшиеся данные переданы из FIFO в память.

Примечание: завершение передачи зависит от оставшихся данных в FIFO, подлежащих передач в память только в случае режима периферия-память. Это условие не применимо в режиме память-периферия.

Если поток сконфигурирован в не кольцевом режиме (noncircular mode), то после окончания передачи (когда количество передаваемых данных достигло нуля) DMA останавливается (бит EN в регистре DMA_SxCR очищается программно), и никакой запрос DMA не обслуживается если программа заново не запрограммирует поток и заново не разрешит его (установкой бита EN в регистре DMA_SxCR).

Приостановка передачи DMA. В любой момент времени передача DMA может быть приостановлена, чтобы быть возобновленной позже, или полностью быть запрещенной до завершения передачи.

Есть два случая:

• Поток запрещает передачу, чтобы позже его не перезапускать из точки, где была остановка. В этой ситуации ничего делать не нужно, кроме как очистить бит EN в регистре DMA_SxCR, чтобы запретить поток. Потоку может понадобиться время, чтобы перейти в запрещенное состояния (сначала должна

завершится текущая передач). Флаг прерывания завершения передачи (бит TCIF а регистре DMA_LISR или DMA_HISR) установится, чтобы показать конец передачи. Значение бита EN в регистре DMA_SxCR теперь равно 0, чтобы подтвердить обрыв работы потока. Регистр DMA_SxNDTR содержит количество оставшихся элементов данных в момент, когда поток был установлен, так что программа может определить, сколько элементов данных было передано перед моментом, когда работа потока была прервана.

• Поток приостанавливает передачу до того, как достигнет 0 количество оставшихся элементов данных для передачи в регистре DMA_SxNDTR. Целью является перезапуск передачи после повторного разрешения потока. Чтобы перезапуск был с точки, где передача была остановлена, программа должна прочитать регистр DMA_SxNDTR после запрета потока записью бита EN в регистре DMA_SxCR (и затем проверить, что он стал равен 0), чтобы узнать количество данных, которое уже было собрано. Затем:

– Адрес периферии и/или памяти должен быть обновлен, чтобы подстроить указатели адреса.
– Регистр SxNDTR должен быть обновлен оставшимся количеством элементов данных, которое нужно передать (прочитанное значение, когда поток был запрещен).
– Затем поток может быть заново разрешен, чтобы передача перезапустилась с того места, где была остановлена.

Примечание: обратите внимание, что флаг установится прерывания завершения передачи (TCIF в регистре DMA_LISR или DMA_HISR), чтобы показать окончание передачи из-за обрыва работы потока.

Контроллер потока. Объект, который управляет количеством передаваемых данных, известен как контроллер потока (flow controller). Этот контроллер конфигурируется независимо для каждого потока с использованием бита PFCTRL в регистре DMA_SxCR.

Контроллером потока может быть:

• Контроллер DMA. В этом случае количество элементов данных для передачи устанавливает программа в регистре DMA_SxNDTR перед тем, как разрешается поток DMA.
• Периферия, которая является источником или местом назначения данных: это тот случай, когда количество передаваемых элементов данных неизвестно. Периферия аппаратно показывает контроллеру DMA момент переноса последних данных. Эта функция поддерживается только для периферийных устройств, которые могут сигнализировать об окончании передачи, т. е. только для SDIO.

Когда для определенного потока контроллером является периферия, то значение, записанное в регистр DMA_SxNDTR, не оказывает влияние на передачу DMA. В действительности, независимо от записанного значения, это значение принудительно установлено аппаратурой в 0xFFFF, как только поток был разрешен, чтобы соответствовать следующим схемам:

• Ожидаемое прекращение работы потока: бит EN в регистре DMA_SxCR сбрасывается в 0 программой, чтобы остановить поток до того, как аппаратный сигнал последних данных (single или burst) отправлен периферией. В этом случае поток выключается, и срабатывает сброс FIFO (flush) в случае передачи DMA периферия-память. Флаг TCIFx в регистре статуса соответствующего потока установится, чтобы показать завершение работы DMA. Чтобы узнать количество элементов данных, переданных во время передачи DMA, считывается регистр DMA_SxNDTR, и применяется следующая формула:

Количество переданных данных = 0xFFFF – DMA_SxNDTR

• Нормальное завершение работы потока из-за получения аппаратного сигнала последних данных: поток автоматически будет прерывает рабоу, когда периферия запрашивает последнюю передачу DMA (single или burst), и когда эта передача завершится. Установится флаг TCIFx в регистре статуса соответствующего потока, чтобы показать завершение передачи DMA. Чтобы узнать, сколько элементов данных передано, прочитайте регистр DMA_SxNDTR, и примените ту же самую формулу, показанную выше.

• Регистр DMA_SxNDTR достиг 0: установится флаг TCIFx в регистре статуса соответствующего потока, чтобы показать принудительное завершение передачи DMA. Поток автоматически отключается, даже если аппаратный сигнал последних данных (single или burst) еще не был выставлен. Уже переданные данные не потеряются. Это означает, что в одной транзакции блок DMA может обработать максимум 65535 элементов данных, даже когда потоком управляет периферия (peripheral flow control mode).

Примечание: когда сконфигурирован режим память-память, блок DMA всегда является контроллером потока, и аппаратура принудительно сбрасывает бит PFCTRL в 0.

Кольцевой режим (circular mode) запрещен, когда периферия управляет потоком (peripheral flow controller mode).

Обзор возможных конфигураций DMA. В таблице 49 суммарно показаны различные возможные конфигурации DMA.

Таблица 49. Возможные конфигурации DMA.

Режим передачи DMA	**Источник данных**	**Место назначения данных**	**Контроллер потока**	**Режим кольцевого буфера**	**Тип передачи**	**Режим direct**	**Режим двойного буфера**
Периферия-память	Порт AHB периферии	Порт AHB памяти	DMA	Возможен	Single	Возможен	Возможен
					Burst	Запрещено	
			Периферия	Запрещено	Single	Возможен	Запрещено
					Burst	Запрещено	
Память-периферия	Порт AHB памяти	Порт AHB периферии	DMA	Возможен	Single	Возможен	Возможен
					Burst	Запрещено	
			Периферия	Запрещено	Single	Возможен	Запрещено

Память-память	Порт AHB периферии	Порт AHB памяти	Только DMA	Запрещено	Burst Single Burst	Запрещено Запрещено	Запрещено

Процедура конфигурации потока. Для конфигурирования DMA-потока x (здесь x номер потока) нужно выполнить следующую последовательность действий:

1. Если поток разрешен, запретите его сбросом бита EN в регистре DMA_SxCR, затем прочитайте этот бит, чтобы убедиться, что не происходит текущая операция на потоке. Запись этого бита в 0 не сразу дает эффект остановки потока (бит сбросится не сразу), поскольку он физически обнуляется как только завершатся все текущие передачи. Когда бит EN читается как 0, это означает, что поток остановлен и готов к конфигурированию. Поэтому нужно подождать, когда сбросится бит EN перед тем, как начать конфигурировать поток. Все установленные биты в регистре статуса (DMA_LISR и DMA_HISR), отражающие предыдущую передачу блока данных DMA, должны быть очищены до того, как поток можно заново разрешить.

2. Установите адрес регистра порта периферии в регистре DMA_SxPAR. Данные будут перемещаться из этого регистра или в этот регистр (в зависимости от направления передачи), чтобы попасть в место назначения (из него) после события периферийного устройства (peripheral event).

3. Установите адрес памяти в регистре DMA_SxMA0R (и в регистре DMA_SxMA1R в случае режима двойной буферизации). Данные будут записаны в эту память или прочитаны из неё после события периферийного устройства.

4. Сконфигурируйте в регистре DMA_SxNDTR общее количество элементов данных для передачи. После события периферии или каждой порции (beat) передачи burst это значение декрементируется.

5. Выберите канал DMA (запрос), используя поле бит CHSEL[2:0] в регистре DMA_SxCR.

6. Если контроллером потока должна быть периферия, и если она поддерживает эту функцию, то установите бит PFCTRL в регистре DMA_SxCR.

7. Сконфигурируйте приоритет потока полем бит PL[1:0] регистра DMA_SxCR.

8. Сконфигурируйте использование FIFO (разрешен он или запрещен, а также порог заполнения при передаче и приеме).

9. Сконфигурируйте направление передачи данных периферии и памяти, режим incremented/fixed, транзакции single или burst, ширину данных периферии и памяти, кольцевой режим, режим двойного буфера и прерывания после половины и/или полной передачи, и/или прерывания при возникновении ошибок в регистре DMA_SxCR.

10. Активируйте поток установкой бита EN в регистре DMA_SxCR.

Как только поток разрешен, он может обслужить любой запрос DMA от периферии, подключенной к потоку.

Как только половина данных была передана на порту AHB места назначения, установится флаг половины передачи (half-transfer flag, HTIF), и сгенерируется прерывание, если установлен бит разрешения этого прерывания (HTIE). По окончании передачи установится флаг прерывания завершения передачи (transfer complete interrupt flag, TCIF), и будет сгенерировано прерывание, если оно разрешено (если установлен бит transfer complete interrupt enable, TCIE).

> Предупреждение: для выключения периферии, подключенной к запросу потока DMA, важно сначала выключить поток DMA, к которому привязана периферия, затем подождать сброса в 0 бита EN. И только потом можно безопасно запретить периферийное устройство.

Управление обработкой ошибок. Контроллер DMA может детектировать следующие ошибки:

• Transfer error: установится флаг прерывания при ошибке передачи (transfer error interrupt flag, TEIFx), когда:

– Во время доступа DMA на чтение или запись произошла ошибка шины.
– Программа запросила доступ на запись через регистр адреса памяти в режиме двойной буферизации, в то время как поток разрешен, и текущая целевая память является той, на которую влияет запись в регистр адреса памяти (см. выше секцию "Режим двойного буфера").

• Ошибка FIFO: установится флаг прерывания ошибки FIFO (FIFO error interrupt flag, FEIFx), если:

– Детектировано состояние недогрузки FIFO (FIFO underrun).
– Детектировано состояние переполнения FIFO (FIFO overrun). Это детектирование недоступно в режиме DMA память-память, потому что эти передачи обслуживаются внутри блока DMA.
– Поток разрешен, когда порог уровня заполнения FIFO (FIFO threshold level) несовместим с размером memory burst (см. таблицу 48 "Конфигурации порога FIFO").

• Ошибка direct mode: флаг прерывания по ошибке direct-режима (direct mode error interrupt flag, DMEIFx) может только установиться в режиме периферия-память, когда работает direct-режим, и когда сброшен бит MINC в регистре DMA_SxCR. Этот флаг установится, когда произошел запрос DMA, в то время как предыдущие данные не были полностью переданы в память (из-за того, что не была предоставлена шина памяти). В этом случае флаг показывает, что 2 элемента данных были последовательно переданы по одному и тому же адресу назначения, что может вызвать проблему, если получатель не может управлять этой ситуацией.

В direct-режиме флаг FIFO error также может установиться в следующих ситуациях:

• В режиме периферия-память FIFO может быть насыщен (overrun, переполнение), если шина памяти не была предоставлена для некоторых запросов периферии.

• В режиме память-периферия, может произойти недогрузка (underrun, опустошения) если шина памяти не была предоставлена перед возникновением запроса периферии.

Если флаг TEIFx или FEIFx установлен из-за несовместимости размера burst и уровня порога заполнения FIFO, отказавший поток автоматически запрещается путем очистки аппаратурой бита EN в регистре конфигурации (DMA_SxCR) соответствующего потока.

Если флаг DMEIFx или FEIFx установится из-за ситуации overrun или underrun, отказавший поток не будет автоматически запрещен, и в области ответственности программы будет запретить или не запрещать поток сбросом бита EN в регистре DMA_SxCR. Это потому, что не было потери данных, когда произошла ошибка такого рода.

Когда установился флаг прерывания по ошибке потока (TEIF, FEIF, DMEIF) в регистре DMA_LISR или DMA_HISR, генерируется прерывание, если установлен соответствующий бит разрешения прерывания (TEIE, FEIE, DMIE) в регистре DMA_SxCR или DMA_SxFCR.

Примечание: когда произойдет переполнение или недогрузка FIFO (overrun или underrun), данные не теряются, потому что запрос периферии не подтверждается потоком, пока не будет очищено состояние переполнения или недогрузки (overrun или underrun). Если подтверждение занимает слишком много времени, периферия сама может детектировать переполнение или недогрузку своего внутреннего буфера, и данные могут быть потеряны.

[**Прерывания DMA**]

Для каждого потока DMA может быть сгенерировано прерывание при следующих событиях:

• Половина передачи (half-transfer).
• Завершение передачи (transfer complete).
• Ошибка передачи (transfer error).
• Ошибка FIFO (overrun, underrun или ошибка уровня заполнения FIFO).
• Ошибка direct-режима.

Есть отдельные управляющие биты разрешения этих прерываний, как показано в таблице 50.

Таблица 50. Запросы прерываний DMA.

Событие прерывания	**Флаг события**	**Бит разрешения прерывания**
Половина передачи (half-transfer)	HTIF	HTIE
Завершение передачи	TCIF	TCIE
Ошибка передачи	TEIF	TEIE
Переполнение/недогрузка FIFO (FIFO overrun/underrun)	FEIF	FEIE
Ошибка режима direct	DMEIF	DMEIE

Примечание: перед установкой в 1 бита управления разрешением прерывания должен быть очищен флаг соответствующего события прерывания, иначе может быть немедленно сгенерировано прерывание.

[**Регистры DMA**]

Регистрам DMA осуществляется 32-битный доступ.

DMA low interrupt status register (DMA_LISR)

DMA high interrupt status register (DMA_HISR)

DMA low interrupt flag clear register (DMA_LIFCR)

DMA high interrupt flag clear register (DMA_HIFCR)

DMA stream x configuration register (DMA_SxCR) (x = 0..7)

DMA stream x number of data register (DMA_SxNDTR) (x = 0..7)

DMA stream x peripheral address register (DMA_SxPAR) (x = 0..7)

DMA stream x memory 0 address register (DMA_SxM0AR) (x = 0..7)

DMA stream x memory 1 address register (DMA_SxM1AR) (x = 0..7)

DMA stream x FIFO control register (DMA_SxFCR) (x = 0..7)

Таблица 51. Карта памяти регистров DMA. См. секцию 2.3 "Memory map" даташита [2] для информации по граничным адресам областей памяти.

Смещ.	Регистр	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	DMA_LISR	Зарезерв.				TCIF3	HTIF3	TEIF3	DMEIF3	Зарезерв.	FEIF3	TCIF2	HTIF2	TEIF2	DMEIF2	Зарезерв.	FEIF2	Зарезерв.				TCIF1	HTIF1	TEIF1	DMEIF1	Зарезерв.	FEIF1	TCIF0	HTIF0	TEIF0	DMEIF0	Зарезерв.	FEIF0
	Значение сброса					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0		0
0x0004	DMA_HISR	Зарезерв.				TCIF7	HTIF7	TEIF7	DMEIF7	Зарезерв.	FEIF7	TCIF6	HTIF6	TEIF6	DMEIF6	Зарезерв.	FEIF6	Зарезерв.				TCIF5	HTIF5	TEIF5	DMEIF5	Зарезерв.	FEIF5	TCIF4	HTIF4	TEIF4	DMEIF4	Зарезерв.	FEIF4
	Значение сброса					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0		0
0x0008	DMA_LIFCR	Зарезерв.				CTCIF3	CHTIF3	TEIF3	CDMEIF3	Зарезерв.	CFEIF3	CTCIF2	CHTIF2	CTEIF2	CDMEIF2	Зарезерв.	CFEIF2	Зарезерв.				CTCIF1	CHTIF1	CTEIF1	CDMEIF1	Зарезерв.	CFEIF1	CTCIF0	CHTIF0	CTEIF0	CDMEIF0	Зарезерв.	CFEIF0
	Значение сброса					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0		0
0x000C	DMA_HIFCR	Зарезерв.				CTCIF7	CHTIF7	CTEIF7	CDMEIF7	Зарезерв.	CFEIF7	CTCIF6	CHTIF6	CTEIF6	CDMEIF6	Зарезерв.	CFEIF6	Зарезерв.				CTCIF5	CHTIF5	CTEIF5	CDMEIF5	Зарезерв.	CFEIF5	CTCIF4	CHTIF4	CTEIF4	CDMEIF4	Зарезерв.	CFEIF4
	Значение сброса					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0		0
0x0010	DMA_S0CR	Зарезерв.				CHSEL[2:0]			MBURST[1:0]		PBURST[1:0]			CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Значение сброса					0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0014	DMA_S0NDTR	Зарезервировано																NDT[15:0]															
	Значение сброса																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0018	DMA_S0PAR	PA[31:0]																															
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x001C	DMA_S0M0AR	M0A[31:0]																															
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0020	DMA_S0M1AR	M1A[31:0]																															
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0024	DMA_S0FCR	Зарезервировано																								FEIE	Зарезерв.	FS[2:0]			DMDIS	FTH [1:0]	
	Значение сброса																									0		1	0	0	0	0	1
0x0028	DMA_S1CR	Зарезерв.				CHSEL [2:0]			MBURST[1:]		PBURST[1:0]		Зарезерв.	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Значение сброса					0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x002C	DMA_S1NDTR	Зарезервировано																NDT[15:0]															
	Значение сброса																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Смещ.	Регистр	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0030	DMA_S1PAR	PA[31:0]																															
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0034	DMA_S1M0AR	M0A[31:0]																															
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0038	DMA_S1M1AR	M1A[31:0]																															
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x003C	DMA_S1FCR	Зарезервировано																								FEIE	Зарезерв.	FS[2:0]			DMDIS	FTH [1:0]	
	Значение сброса																									0		1	0	0	0	0	1
0x0040	DMA_S2CR	Зарезерв.				CHSEL [2:0]			MBURST[1:0]		PBURST[1:0]		Зарезерв.	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR [1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Значение сброса					0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0044	DMA_S2NDTR	Зарезервировано																NDT[15:0]															
	Значение сброса																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0048	DMA_S2PAR	PA[31:0]																															
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004C	DMA_S2M0AR	M0A[31:0]																															
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0050	DMA_S2M1AR	M1A[31:0]																															
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0054	DMA_S2FCR	Зарезервировано																								FEIE	Зарезерв.	FS[2:0]			DMDIS	FTH [1:0]	
	Значение сброса																									0		1	0	0	0	0	1
0x0058	DMA_S3CR	Зарезерв.				CHSEL[2:0]			MBURST[1:0]		PBURST[1:0]		Зарезерв.	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Значение сброса					0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	DMA_S3NDTR																	NDT[15:0]															

0x005C		Зарезервировано																													
	Значение сброса															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0060	DMA_S3PAR	PA[31:0]																													
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0064	DMA_S3M0AR	M0A[31:0]																													
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Смещ.	Регистр	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0068	DMA_S3M1AR	M1A[31:0]																													
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x006C	DMA_S3FCR	Зарезервировано																								FEIE	Зарезерв.	FS[2:0]			DMDIS	FTH [1:0]	
	Значение сброса																									0		1	0	0	0	0	1
0x0070	DMA_S4CR	Зарезерв.			CHSEL[2:0]			MBURST[1:0]		PBURST[1:0]		Зарезерв.	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR [1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Значение сброса				0	0	0	0	0	0	0		0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0074	DMA_S4NDTR	NDT[15:0]																													
	Значение сброса																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0078	DMA_S4PAR	PA[31:0]																													
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x007C	DMA_S4M0AR	M0A[31:0]																													
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0080	DMA_S4M1AR	M1A[31:0]																													
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0084	DMA_S4FCR	Зарезервировано																								FEIE	Зарезерв.	FS[2:0]			DMDIS	FTH [1:0]	
	Значение сброса																									0		1	0	0	0	0	1
0x0088	DMA_S5CR	Зарезерв.			CHSEL[2:0]			MBURST[1:0]		PBURST[1:0]		Зарезерв.	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Значение сброса				0	0	0	0	0	0	0		0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x008C	DMA_S5NDTR	Зарезервировано																NDT[15:0]														
	Значение сброса																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0090	DMA_S5PAR	PA[31:0]																													
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0094	DMA_S5M0AR	M0A[31:0]																													
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0098	DMA_S5M1AR	M1A[31:0]																													
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x009C	DMA_S5FCR	Зарезервировано																								FEIE	Зарезерв.	FS[2:0]			DMDIS	FTH [1:0]	
	Значение сброса																									0		1	0	0	0	0	1
0x00A0	DMA_S6CR	Зарезерв.			CHSEL[2:0]			MBURST[1:0]		PBURST[1:0]		Зарезерв.	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Значение сброса				0	0	0	0	0	0	0		0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00A4	DMA_S6NDTR	Зарезервировано																NDT[15:.]														
	Значение сброса																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00A8	DMA_S6PAR	PA[31:0]																													
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Смещ.	Регистр	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00AC	DMA_S6M0AR	M0A[31:0]																													
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00B0	DMA_S6M1AR	M1A[31:0]																													
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00B4	DMA_S6FCR	Зарезервировано																								FEIE	Зарезерв.	FS[2:0]			DMDIS	FTH [1:0]	
	Значение сброса																									0		1	0	0	0	0	1
0x00B8	DMA_S7CR	Зарезерв.			CHSEL[2:0]			MBURST[1:0]		PBURST[1:0]			CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Значение сброса				0	0	0	0	0	0	0		0	0	0	0		-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00BC	DMA_S7NDTR	Зарезервировано																NDT[15:0]														
	Значение сброса																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C0	DMA_S7PAR	PA[31:0]																													
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C4	DMA_S7M0AR	M0A[31:0]																													
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C8	DMA_S7M1AR	M1A[31:0]																													
	Значение сброса	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0x00CC	DMA_S7FCR	Зарезервировано				FEIE	Зарезерв.	FS[2:0]			DMDIS	FTH [1:0]		
	Значение сброса					0		1	0	0	0	0	1

[**Ссылки**]

1. STM32F427xx STM32F429xx Datasheet site:st.com.

2. RM0090 Reference manual STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm®-based 32-bit MCUs site:st.com.

3. STM32: аббревиатуры и термины.

4. STM32: использование контроллера DMA.