

Вступление

В данном документе описывается использование прямого доступа к памяти (DMA) контроллер доступный в STM32F2, STM32F4 и STM32F7 серии. Функции контроллера DMA, системная архитектура, матричная шина многослойной и системная память способствуют, чтобы обеспечить высокую пропускную способность данных и развивать очень низкую задержку время ответа программного обеспечения. Это приложение к сведению также описывает некоторые советы и приемы, чтобы позволить разработчикам в полной мере использовать эти функции и обеспечить правильное время отклика для различных периферийных устройств и подсистем.

STM32F2, STM32F4 и STM32F7 упоминаются как «STM32F2 / F4 / F7 устройств» и контроллер DMA, как «DMA» в течение всего документа.

В этом документе STM32F4 серия выбрана в качестве иллюстративного примера. Поведение DMA одинакова по STM32F2, STM32F4 и STM32F7 серии, если не указано иное.

Справочные документы

Это приложение записку следует рассматривать в сочетании с STM32F2 / F4 / F7 справочниками:

- STM32F205 / 215 и STM32F207 / 217 Справочное руководство (RM0033)
- STM32F405 / 415, STM32F407 / 417, STM32F427 / 437 и STM32F429 / 439 Справочное руководство (RM0090)
- STM32F401xB / C и STM32F401xD / справочное пособие E (RM0368)
- STM32F410 справочное руководство (RM0401)
- STM32F411xC / справочное пособие E (RM0383)
- STM32F412 справочное руководство (RM0402)
- STM32F446xx справочное руководство (RM0390)
- STM32F469xx и STM32F479xx справочное руководство (RM0386)
- STM32F75xxx и STM32F74xxx справочное руководство (RM0385)
- STM32F76xxx и STM32F77xxx справочное руководство (RM0410)

содержание

1	Описание контроллера DMA.	6
	1.1 Свойства передачи DMA.	6
	1.1.1 DMA потоков / каналов.	7
	1.1.2 Поток приоритет.	9
	1.1.3 Адреса источника и назначения.	9
	1.1.4 Режим передачи.	10
	1.1.5 Размер трансфера.	10
	1.1.6 Приращение адрес источника / назначения.	10
	1.1.7 Источник и назначение данных ширина.	10
	1.1.8 Типы передачи.	10
	1.1.9 Режим DMA FIFO.	11
	1.1.10 Источник и назначение лопнуть размер.	12
	1.1.11 Режим двойного буфера.	13
	1.1.12 Управление потоком.	14
	1.2 Настройка передачи DMA.	14
2	соображения производительности системы.	16
	2.1 матрица Шина Многослойная.	17
	2.1.1 Определения.	17
	2.1.2 Циклическая схема приоритетов.	18
	2.1.3 BusMatrix арбитраж и передача DMA задерживает наихудший случай.	19
	2.2 DMA путей передачи.	20
	2.2.1 Dual DMA порт.	20
	2.2.2 Передача DMA состояний.	22
	2.2.3 DMA запрос арбитража.	23
	2,3 АНВ-к-АПБ моста.	24
	2.3.1 Двойной АНВ к АРВ порт.	24
	2.3.2 АНВ-к-АПБ моста арбитража.	24
3	Как предсказать DMA латентности.	26
	3.1 DMA время передачи.	26
	3.1.1 По умолчанию DMA синхронизации передачи данных.	26
	3.1.2 DMA время передачи в зависимости от одновременного доступа.	27
	3,2 Примеры	28

	3.2.1	ADC к SRAM передачи DMA	28
	3.2.2	SPI полнодуплексный передачи DMA	29
4		Советы и предупреждения при программировании контроллера DMA	31
	4,1	Последовательность Программного обеспечения для отключения DMA	31
	4,2	DMA управления флагом перед включением новой передачи.	31
	4,3	Последовательность программного обеспечения для включения DMA	31
	4,4	Память-память, а передача NDTR = 0.	31
	4,5	DMA периферийного разрыва с Pinc / MINC = 0.	31
	4,6	Дважды отображенных DMA запросы.	32
	4,7	Лучший DMA пропускной способности конфигурации.	32
	4,8	DMA передача подвески.	32
	4,9	Возьмите преимущество контроллера DMA2 и гибкость архитектуры системы.	33
	4.9.1	Инверсия переводы более DMA2 АНВ рассмотрения портов.	33
	4.9.2	Пример для инвертирования переводов Quad-SPI над DMA2 АНВ рассмотрения портов.	34
	4,10	STM32F7 DMA передачи и кэш обслуживание для неогерентности 35 избежать данных	
5		Вывод	36
6		Лист регистраций изменений	37

Список таблиц

Таблица 1.	STM32F427 / 437 и STM32F429 / 439 отображение запроса DMA1.	8
Таблица 2.	STM32F427 / 437 и STM32F429 / 439 отображение запроса DMA2.	9
Таблица 3.	Возможные конфигурации лопнуть.	13
Таблица 4.	Периферийные порт время доступа / передач по сравнению с путем DMA используется.	27
Таблица 5.	порт памяти время доступа / передачи.	27
Таблица 6.	DMA периферийных задержки передачи (АЦП) порт.	28
Таблица 7.	DMA памяти (SRAM) порт передачи задержки.	28
Таблица 8.	DMA АОБ направление конфигурация порта режим передачи данных по сравнению с.	33
Таблица 9.	Фрагмент кода.	35
Таблица 10.	Документ пересмотр истории.	37

Список рисунков

Фигура 1.	DMA блок-схему.	7
Фигура 2.	Выбор канала.	8
Рисунок 3.	адрес источника DMA и адрес получателя инкрементация.	10
Рисунок 4.	FIFO-структура.	11
На рисунке 5.	DMA лопнуть передачи.	12
Рисунок 6.	Режим двойного буфера.	13
На рисунке 7.	STM32F405 / 415 и STM32F407 / 417 архитектура системы.	17
Рисунок 8.	CPU и DMA1 запросить доступ к SRAM1.	18
На рисунке 9.	Пять мастеров запросить доступ SRAM.	19
Рисунок 10.	DMA, задержка передачи за счет передачи центрального процессора, выданного прерывания.	20
Рисунок 11.	DMA, два порта.	21
Рисунок 12.	Периферийное состояние передачи-к-памяти.	22
Рисунок 13.	Память-к-периферийная состояния передачи.	23
Рисунок 14.	Запрос DMA арбитража.	23
Рисунок 15.	AHB-к-APB1 моста одновременно процессор и запрос доступа DMA1.	25
Рисунок 16.	SPI полнодуплексного времени передачи DMA.	29
Рисунок 17.	DMA в режиме передачи памяти-к-Peripheral.	34

1 Описание контроллера DMA

DMA является AMBA расширенного модуля высокопроизводительной шины (AHB), который имеет три порта: AHB ведомый порт для программирования DMA и два основных портов (периферийные и памяти портов), которые позволяют DMA, чтобы инициировать передачу данных между различными ведомыми модулями. DMA позволяет передачу данных проходить в фоновом режиме, без вмешательства процессора Cortex-Mx. Во время этой операции, главный процессор может выполнять другие задачи, и прерывается только тогда, когда весь блок данных доступен для обработки. Большие объемы данных могут быть переданы без серьезных последствий для производительности системы. DMA в основном используется для реализации централизованного хранения буфера данных (как правило, в системе SRAM) для различных периферийных модулей.

Контроллер STM32F2 / F4 / F7, DMA, в полной мере использовать систему шин многослойного для того, чтобы обеспечить очень низкую задержку как для передачи DMA и для выполнения CPU / прерывания обнаружения событий / услуги.

1,1 свойства передачи DMA

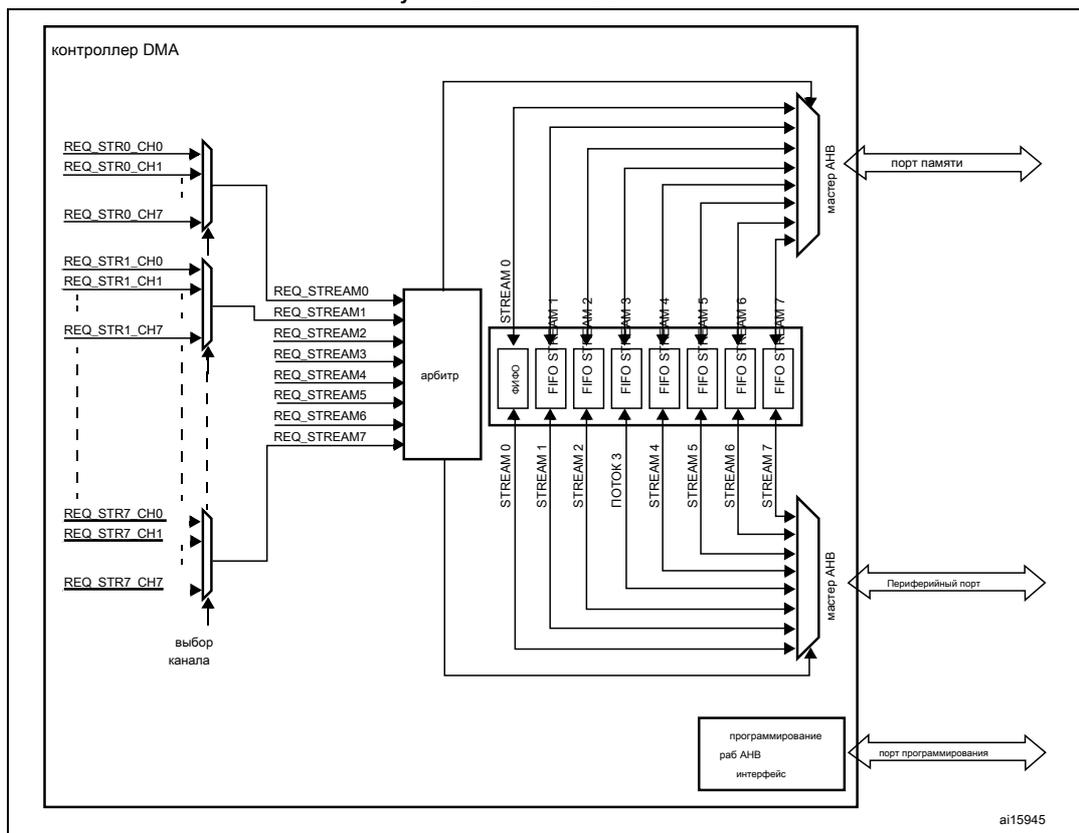
Передача DMA характеризуется следующими свойствами:

- DMA поток / канал
- приоритет потока
- Адреса источника и назначения
- режим передачи
- Размер передачи (только тогда, когда это DMA контроллер потока)
- Источник / адрес назначения приращения или не приращение
- Источник и ширина данных назначения
- Тип перевода
- режим FIFO
- Источник / назначение лопнуть размер
- Режим двойного буфера
- Управление потоком

STM32F2 / F4 / F7 устройство встроить два контроллера DMA, и каждый из DMA имеет два порта, один периферийный порт и один порт памяти, который может работать одновременно.

фигура 1 показана блок-схема DMA.

Рисунок 1. Блок-схема DMA



Следующие подразделы подробного описания каждого свойства передачи DMA.

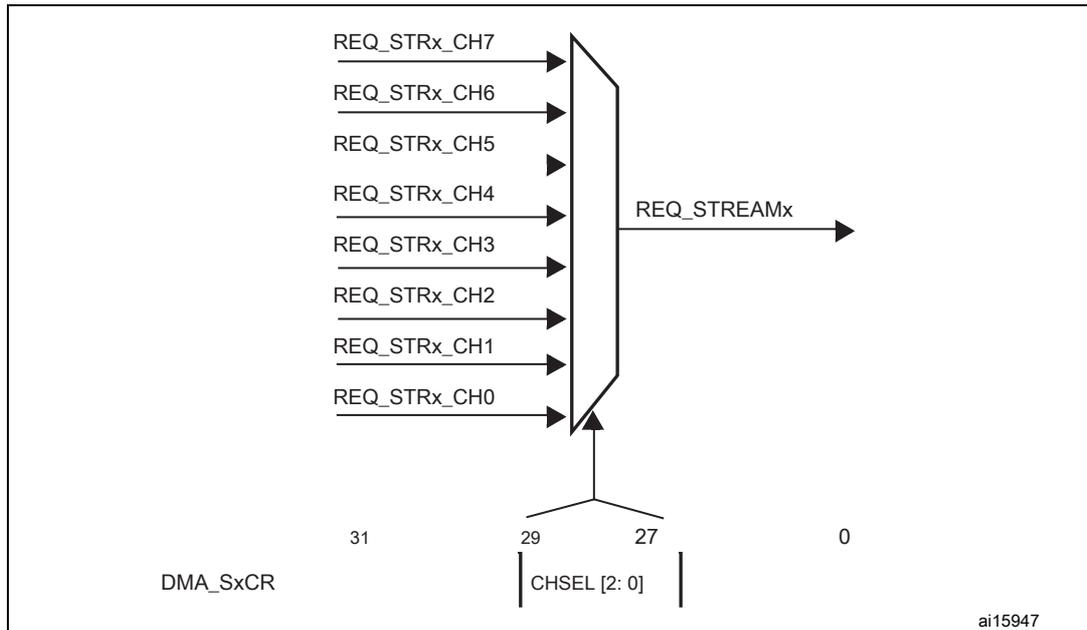
1.1.1 DMA потоков / каналов

STM32F2 / F4 / F7 устройство встроить два контроллера DMA, предлагая до 16 потоков в общей сложности восемь (на контроллер), каждый из которых предназначен для управления запросов доступа памяти из одного или нескольких периферийных устройств.

Каждый поток имеет до восьми выбираемых каналов (запросы) в общей сложности. Этот выбор программно-конфигурируемые и позволяет несколько периферийных устройств для иницирования DMA запросов.

[фигура 2](#) описывает выбор канала для выделенного потока.

Выбор Рисунок 2. Канал



Замечания:

Олько один канал / запрос может быть активным в то же время в потоке. Более один включен DMA поток не должен выполнять тот же периферийный запрос.

Таблица 1 и Таблица 2 показать STM32F427 / 437 и STM32F429 / 439 / DMA1 отображение запросов DMA2. В таблицах приведены доступные конфигурации DMA потоков / каналов в сравнении периферийных запросов.

Таблица 1. STM32F427 / 437 и STM32F429 / 439 отображение запроса DMA1

периферический Поток запрашивает	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
Канал 2 TIM4_CH1	-	-	I2S3_EXT_RX	TIM4_CH2	I2S2_EXT_Texac	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
Channel 4 UART5_RX	USART3_RX	UART4_RX	USART3_TX	UART4_TX	USART2_RX	USART2_TX	UART5_TX	UART5_TX
Канал 5 UART8_TX	UART7_TX	TIM8_TRIG	TIM8_TRIG	TIM3_UP_RX	TIM3_CH1	TIM3_TRIG	TIM3_CH2	UART8_RX
Канал 6 TIM5_CH3	TIM5_CH4	TIM5_TRIG	TIM5_CH1	TIM5_CH4	TIM5_TRIG	TIM5_CH2	-	TIM5_UP
канал 7	-	TIM6_UP	I2C2_RX	I2C2_RX	USART3_Texac	DAC1	DAC2	I2C2_TX

Таблица 2. STM32F427 / 437 и STM32F429 / 439 отображение запроса DMA2

периферический Поток запрашивает	Поток 0	Поток 1	Поток 2	Поток 3	Поток 4	Поток 5	Поток 6	Поток 7
канал 0	ADC1	SAI1_A	TIM8_CH1 TIM8_CH2 TIM8_CH3	SAI1_A	ADC1	SAI1_B	TIM1_CH1 TIM1_CH2 TIM1_CH3	-
канал 1	-	DCMI	ADC2	ADC2	SAI1_B SPI6_TX SPI6_RX			DCMI
канал 2	ADC3	ADC3	-	SPI5_RX SPI5_TX CRYP_	BHE	CRYP_IN HASH_IN		
Канал 3 SPI1_RX	-	SPI1_RX SPI1_TX	-	SPI1_TX	-			
Channel 4 SPI4_RX SPI4_TX USART1_		RX	SDIO	-	USART1_RX	SDIO	USART1_Texas	
канал 5	-	USART6_RX	USART6_RX	SPI4_RX SPI4_TX	- USART6_Texas		USART6_Texas	
Канал 6 TIM1_TRIG	грамм	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 SPI5_TX					TIM8_CH4 TIM8_TRIG TIM8_COM

отображение запроса STM32F2 / F4 / F7, DMA, спроектировано таким образом, что прикладное программное обеспечение имеет большую гибкость для отображения каждого запроса DMA для соответствующего периферийного запроса, и что большинство вариантов использования приложений покрыты мультиплексированием соответствующих DMA потоков и каналы, Обратитесь к DMA1 / DMA2 таблицы запроса картографирования в справочном руководстве, соответствующий микроконтроллер, который вы используете (см [Раздел: Справочные документы](#)).

1.1.2 приоритет потока

Каждый DMA порт имеет арбитр для обработки приоритета между другими потоками DMA. Поток приоритетом является программно-конфигурируемые (существует четыре уровня программного обеспечения). Если два или более DMA потоков имеют одинаковый уровень приоритета программного обеспечения, аппаратные средства приоритет используется (поток 0 имеет приоритет над потоком 1, и т.д.).

1.1.3 Адреса источника и назначения

Передача DMA определяется адресом источника и адрес назначения. Оба источника и назначения должны быть в диапазонах памяти АНВ или АРВ и должны быть приведены в соответствие с размером передачи.

1.1.4 режим передачи

DMA способен выполнять три различных режима передачи:

- Периферийное памяти,
- Память к периферийным,
- Память в память (только DMA2 способно сделать такую передачу, в этом режиме, круговые и прямые режимы не допускаются.)

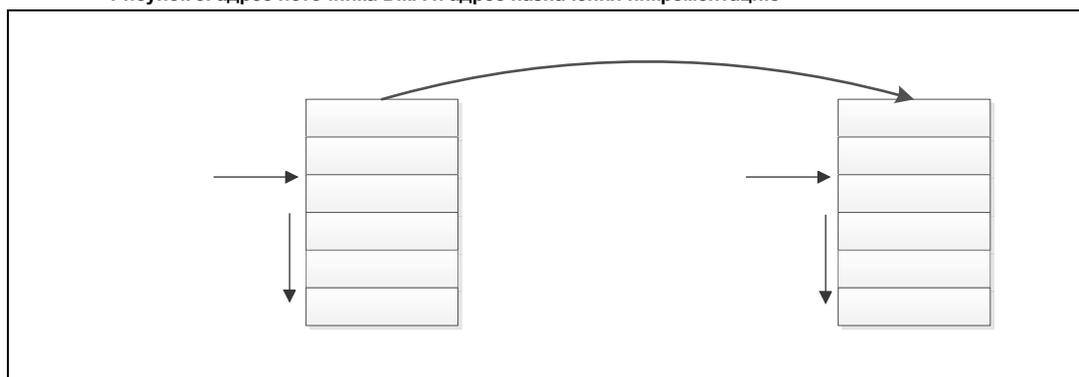
1.1.5 размер трансфера

Значение размера передачи должна быть определена только тогда, когда это DMA, контроллер потока. На самом деле, это значение определяет объем данных, которые должны быть переданы от источника к месту назначения. Размер передаваемого пакета определяется значением регистра DMA_SxNDTR и периферийной шириной на сторону данных. В зависимости от принятого запроса (взрыв или одного), значение размера передачи уменьшается на сумму переданных данных.

1.1.6 Приращение адрес источника / назначения

Можно настроить DMA для автоматического увеличения источника и / или адреса назначения после каждой передачи данных.

Рисунок 3. адрес источника DMA и адрес назначения инкрементацию



1.1.7 Источник и ширина данных назначения

Ширина данных для источника и назначения могут быть определены как:

- Байт (8 бит)
- Полуслво (16 бит)
- Слово (32 бит)

1.1.8 типы передачи

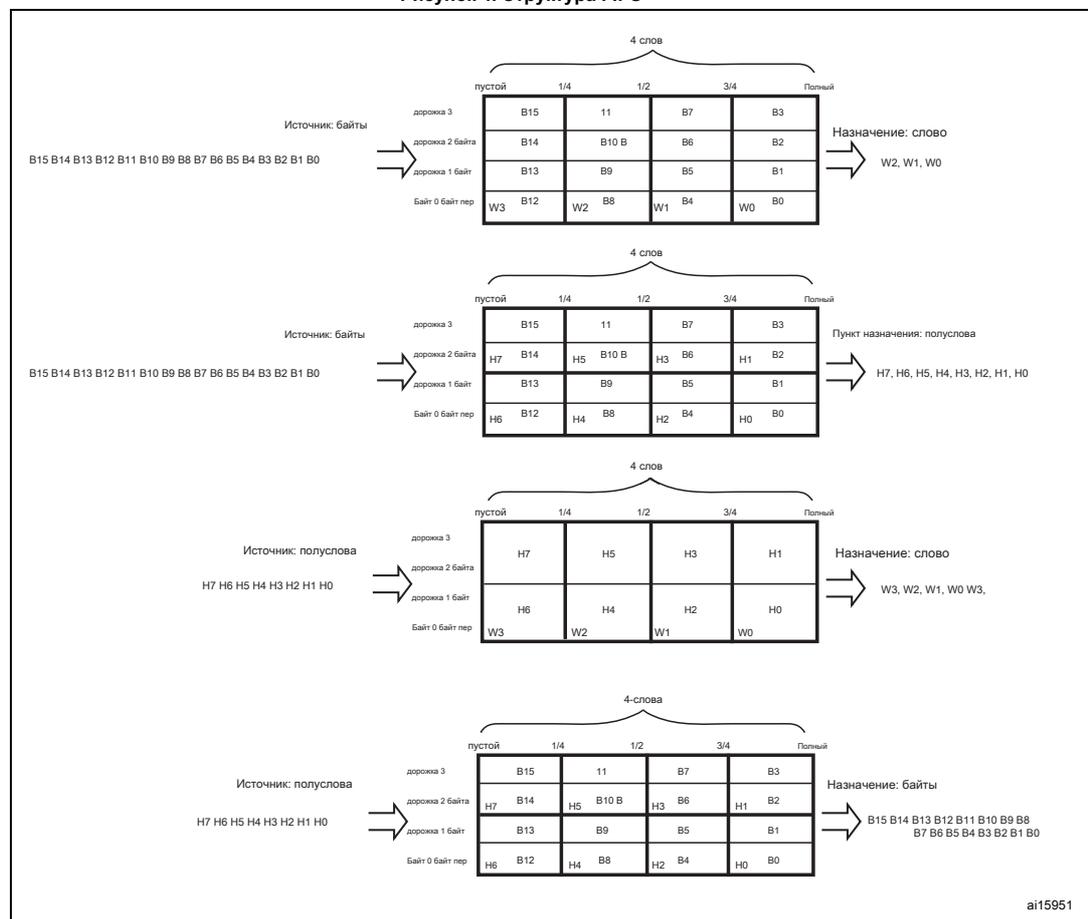
- Круговой режим: круговой режим доступен для обработки циклических буферов и потоков непрерывных данных (The DMA_SxNDTR регистр затем автоматически перезагружается с предварительно запрограммированным значением).
- Нормальный режим: после того, как DMA_SxNDTR регистре достигает нуля, поток отключен (Стандарт EN бит в регистре DMA_SxCR затем равен 0).

1.1.9 Режим DMA FIFO

Каждый поток имеет независимый 4-слово (4 * 32 бит) FIFO и пороговый уровень программно-конфигулируемых между 1/4, 1/2, 3/4 или полный. FIFO используется для временного хранения данных, поступающих от источника до передачи их к месту назначения. DMA FIFO может быть включена или отключена с помощью программного обеспечения; когда отключено, используется прямой режим. Если DMA FIFO включена, упаковка данных / распаковки и / или Пакетный режим может быть использован. Настроен порог DMA FIFO определяет DMA памяти время запроса на порт. В DMA FIFOs реализованы на STM32F2 / F4 / F7 устройств помощи:

- сокращение доступа SRAM и так дать больше времени для других мастеров, чтобы получить доступ к шине матрицы без дополнительного параллелизма,
- позволяет программное обеспечение, чтобы сделать взрыв транзакции, которые оптимизируют пропускную способность передачи данных,
- позволяют упаковки / распаковки данных для адаптации ширины данных источника и назначения без дополнительного доступа DMA.

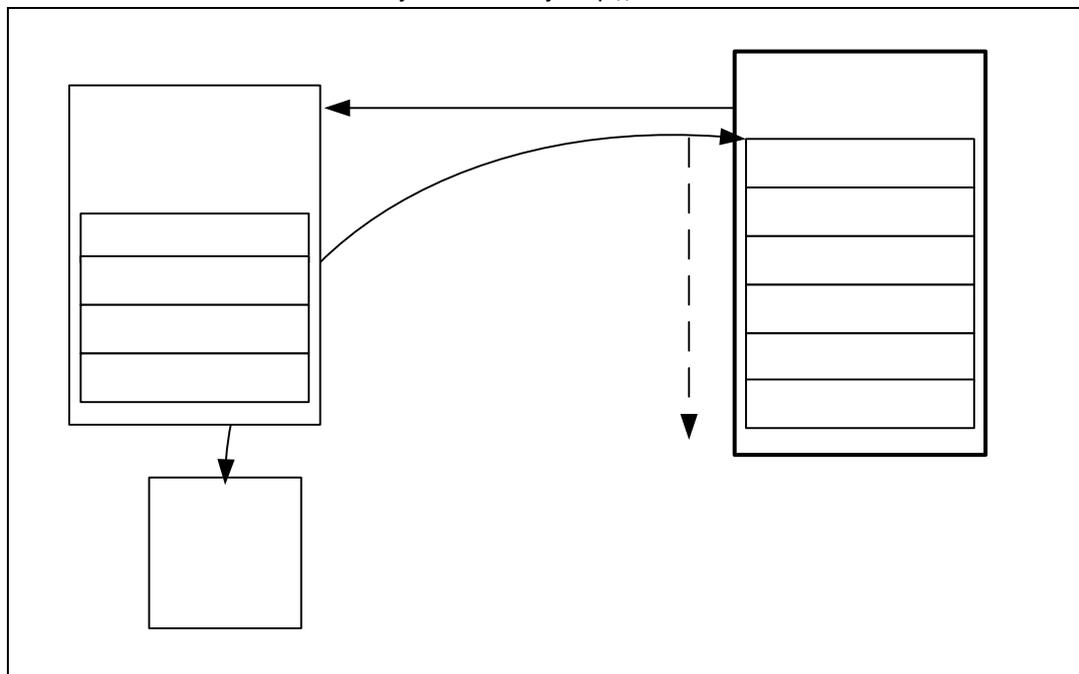
Рисунок 4. Структура FIFO



1.1.10 Источник и лопаются назначения размера

Выброс переводы гарантируется внедренных FIFOs DMA.

Рисунок 5. DMA лопнуть передачи



В ответ на запрос пакетного от периферийного DMA считывает / записывает число блоков данных (блок данных может быть слово, полуслова или байт), запрограммированный на размер разрыва (4x, 8x или 16x блока данных). Размер пакета на периферийный порт DMA должен быть установлен в соответствии с периферийными потребностями / возможностей.

DMA лопнуть размер на порт памяти и порог конфигурации FIFO должны совпадать. Это позволяет потоку DMA иметь достаточное количество данных в FIFO при запуске пакетной передачи на порт памяти. [Таблица 3](#) показывает возможные комбинации размера разрывной памяти, пороговое значение конфигурации FIFO и размера данных.

Для обеспечения согласованности данных, каждая группа передачи, которые образуют взрыв неделима: переводы АНВ заблокированы и вершитель матрицы АНВ шины не удаляют права доступа магистерского DMA во время последовательности пакетной передачи.

Таблица 3. Возможные конфигурации взрыв

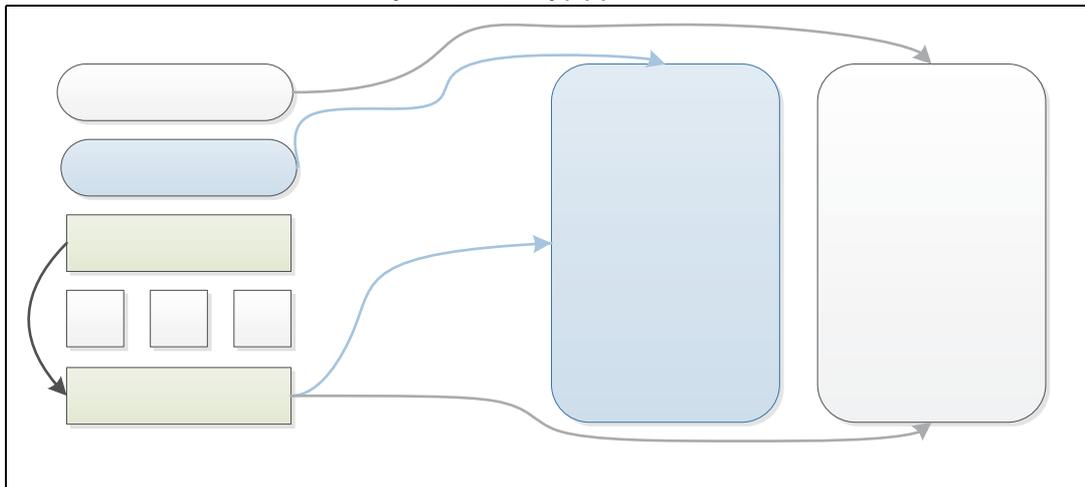
MSIZE	уровень FIFO	MBURST = INCR4	MBURST = INCR8	MBURST = INCR16
Байт	1/4	1 взрыв 4 байта	запрещено	запрещено
	1/2	2 всплески 4 байта	1 взрыв 8 байт	
	3/4	3 всплески 4 байта	запрещено	
	Полный	4 очередей из 4 байтов	2 всплески 8 байт	1 взрыв 16 байт
Полуслово	1/4	запрещено	запрещено	запрещено
	1/2	1 взрыв 4 недомолвок		
	3/4	запрещено		
	Полное	2 вспышки 4 недомолвок 1 лопаются 8 полуслова		
слово	1/4	запрещено	запрещено	запрещено
	1/2			
	3/4			
	Полный	1 взрыв 4 слов		

1.1.11 Режим двойного буфера

Поток двойного буфера работает как обычный (одного буфера) поток, с той разницей, что у него есть два указателя памяти. Когда режим двойной буфер включен, Круговой режим автоматически включается, и на каждом конце транзакции (DMA_SxNDR регистр до 0), указатели памяти меняются местами.

Это позволяет программное обеспечение, чтобы обработать одну область памяти в то время как вторая область памяти заполняется / используется передача DMA.

Рисунок 6. Дважды буфер режим



В режиме двойного буфера, можно обновить базовый адрес порта памяти АНВ на лету (DMA_SxM0AR или DMA_SxM1AR), когда поток включен:

- Когда бита СТ (Текущий Target) в регистре DMA_SxCR равно 0, то текущая целевой DMA памяти расположение 0 памяти и поэтому расположение в памяти базового адреса 1 (DMA_SxM1AR) может быть обновлено.
- Когда бит КТ в регистре DMA_SxCR равен 1, то текущая цель DMA памяти ячейки памяти 1 и так ячейку памяти базового адреса 0 (DMA_SxM0AR) могут быть обновлены.

1.1.12 Управление потоком

Регулятор потока представляет собой блок, который контролирует длину передачи данных и который несет ответственность за прекращение передачи DMA.

Регулятор потока может быть либо DMA или периферийного устройства.

- С DMA в качестве регулятора потока:

В этом случае, необходимо, чтобы определить значение размера передачи в регистре DMA_SxNDTR перед включением связанного DMA потока. Когда запрос DMA, подается, значение размера передачи уменьшается на величину переданных данных (в зависимости от типа запроса: взрыв или одиночный).

Когда значение переноса достигает размера 0, передача DMA завершена и поток DMA отключен.

- С периферии в качестве регулятора потока:

Это тот случай, когда количество элементов данных, подлежащая передаче неизвестно. Периферийное указывает, посредством аппаратных средств для контроллера DMA, когда последние данные передаются. Только SD / MMC и периферийные устройства JPEG поддерживают этот режим.

1.2 Настройка передачи DMA

Чтобы настроить DMA поток x (где x представляет собой номер потока), следует применять следующую процедуру:

1. Если поток включен, отключите его путем сброса EN бит в регистре DMA_SxCR, то прочитайте этот бит для того, чтобы подтвердить, что нет постоянной работы потока. Запись этого бита в 0 не вступают в силу немедленно, так как она на самом деле написано в 0, когда все текущие трансферты закончили. Когда бит EN читается как 0, это означает, что поток готов к настройке. Поэтому необходимо, чтобы ждать, пока EN бит должен быть очищен перед началом любой конфигурации потока. Все потоковые выделенные биты установлены в

- регистр состояния (DMA_LISR и DMA_HISR) из предыдущего блока данных передачи DMA должны быть очищены до того, как поток может быть повторно включен.
2. Установите адрес регистра периферийного порта в регистре DMA_SxPAR. Данные будут перемещены с / на этот адрес в / из периферийного порта после периферического события.
 3. Установите адрес памяти в регистре DMA_SxMA0R (и в регистре DMA_SxMA1R в случае режима двойного буфера). Данные будут записаны или считаны из этой памяти после периферического события.
 4. Настройка общего количества элементов данных, подлежащая передаче в регистре DMA_SxNDTR. После каждого периферийного события или каждый такт пакета, это значение уменьшается.
 5. Выберите канал DMA (запрос), используя CHSEL [2: 0] в регистре DMA_SxCR.
 6. Если периферийное устройство предназначено, чтобы быть контроллером потока, и если он поддерживает эту функцию, установите PFCTRL бит в регистре DMA_SxCR.
 7. Настройка приоритета потока с использованием PL [1: 0] биты в регистре DMA_SxCR.
 8. Использование Настройка FIFO (включить или выключить, порог в передаче и приеме).
 9. Настройка направления передачи данных, периферийные и память увеличивается / фиксированный режим, одиночные или лопаются операции, периферийная и памятей данных ширины, Круглопильный режим, режим двойного буфера и после прерывания половины и / или полной передачи, и / или ошибок в DMA_SxCR регистр.
 10. Активировать поток путем установки EN бит в регистре DMA_SxCR. Как только поток включен, он может служить любой запрос DMA из периферийных устройств, подключенных к потоку.

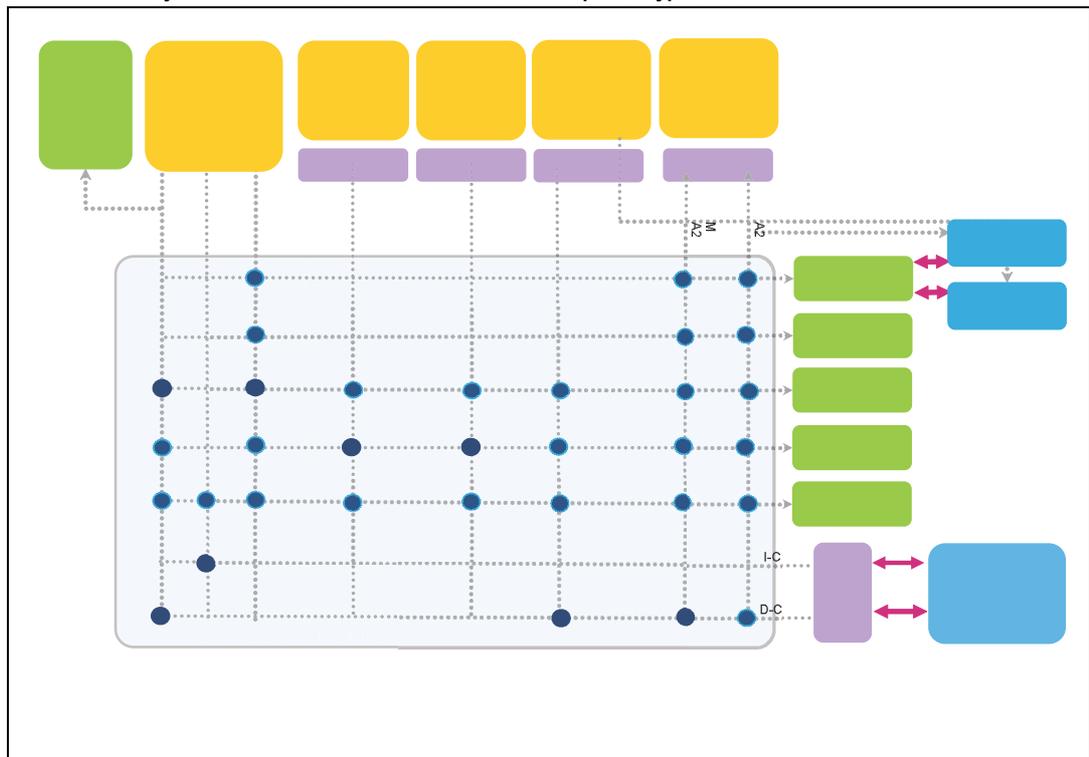
2 соображения производительности системы

STM32F2 / F4 / F7 устройства встроить мульти-мастера / архитектура мульти-рабов:

- Несколько мастеров:
 - кора головного мозга e- Mx основные шины AHB
 - DMA1 шина памяти
 - DMA2 шина памяти
 - DMA2 периферийная шина
 - шина Ethernet DMA
 - USB высокоскоростной DMA шины
 - Шина Chrom-APT Ускоритель
 - LCD-TFT шины
- Несколько рабов:
 - Встроенная флэш-интерфейсы, которые подключены к шине матрицы многослойного
 - Основные внутренние SRAM1 и вспомогательный внутренний SRAMs (SRAM2, SRAM3 при наличии на устройстве)
 - AHB1 периферийных устройства, включая AHB-к-APB APB мосты и периферийные устройства
 - AHB2 периферия
 - AHB3 периферической (например, FMC, Quad-SPI периферийных устройств, когда доступных на линии продукта)

Мастера и рабы соединены через шину матрицу многослойной, обеспечивающую одновременный доступ и эффективную работу, даже когда несколько высокоскоростных периферийные устройства работают одновременно. Эта архитектура показана на следующем рисунке для случая STM32F405 / 415 и STM32F407 / 417 линий.

Рисунок 7. STM32F405 / 415 и STM32F407 / 417 архитектура системы



2,1 Матрица Многослойная автобус

Матрица Шина многослойная позволяет мастерам выполнять передачу данных одновременно до тех пор, как они решают различные подчиненные модули. В верхней части архитектуры и двойной порт AHB DMA's Cortex-Mx, эта структура передачи данных совершенствует параллельность, способствуя тем самым сократить время выполнения и оптимизации эффективности DMA и потребление энергии.

2.1.1 Определения

- Мастер AHB: а мастер шины может инициировать операции читать и писать. Только один мастер может выиграть право собственности на автобусе в определенный период времени.
- AHB ведомое: ответ шина ведомого на операцию мастер чтения или записи. В автобусе ведомые сигналы обратно в основной успех, провал или состояния ожидания.
- AHB арбитром: а арбитр шины гарантирует, что только один мастер может инициировать операцию чтения или записи в одно время.
- матрица шины AHB: многослойная AHB матрица автобуса, который межсоединение AHB мастеров AHB рабов с выделенным AHB арбитром для каждого слоя. Арбитраж использует алгоритм циклического перебора.

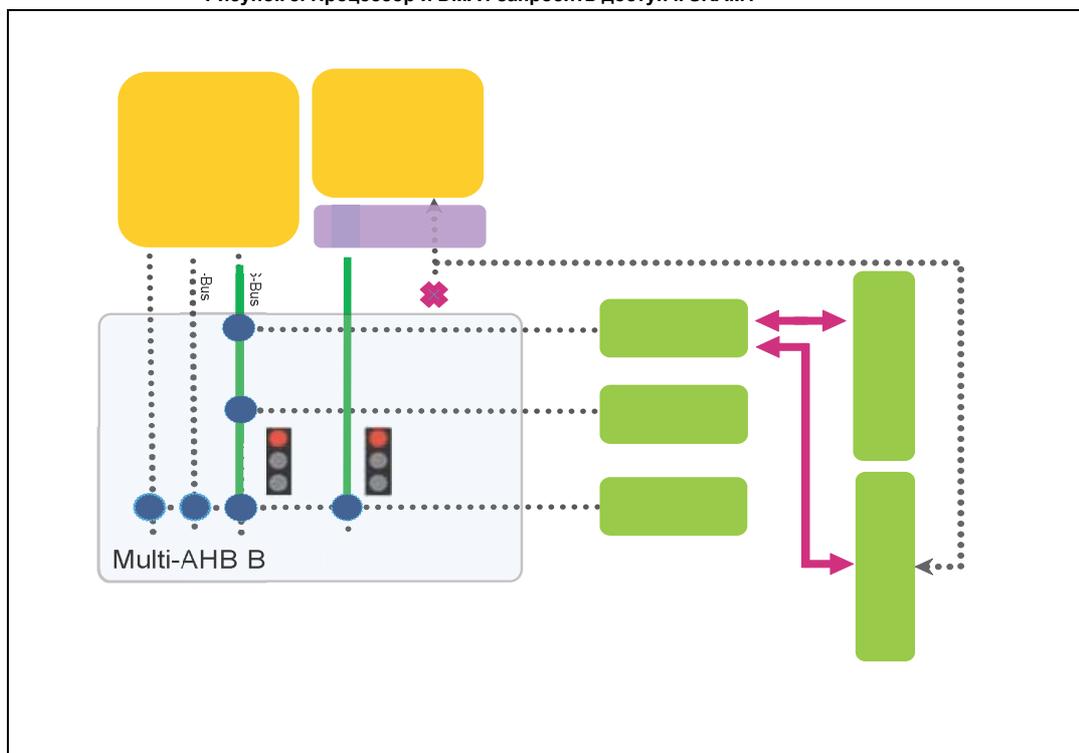
2.1.2 Схема приоритетов Круговой

Схема Приоритетной Круговой реализуется на уровне матрицы шины для того, чтобы гарантировать, что каждый мастер может получить доступ к любому рабу с очень низкой задержкой:

- Круговой арбитраж политика позволяет справедливое распределение полосы пропускания шины.
- Максимальное время ожидания ограничено.
- Круговой квант 1x передача.

Автобусная матрица арбитры вмешиваются для разрешения конфликтов доступа, когда несколько мастеров АНВ пытаются получить доступ к одной АНВ раба одновременно. В следующем примере (*Рисунок 8*), как процессор и DMA1 попытка доступа SRAM1 для чтения данных.

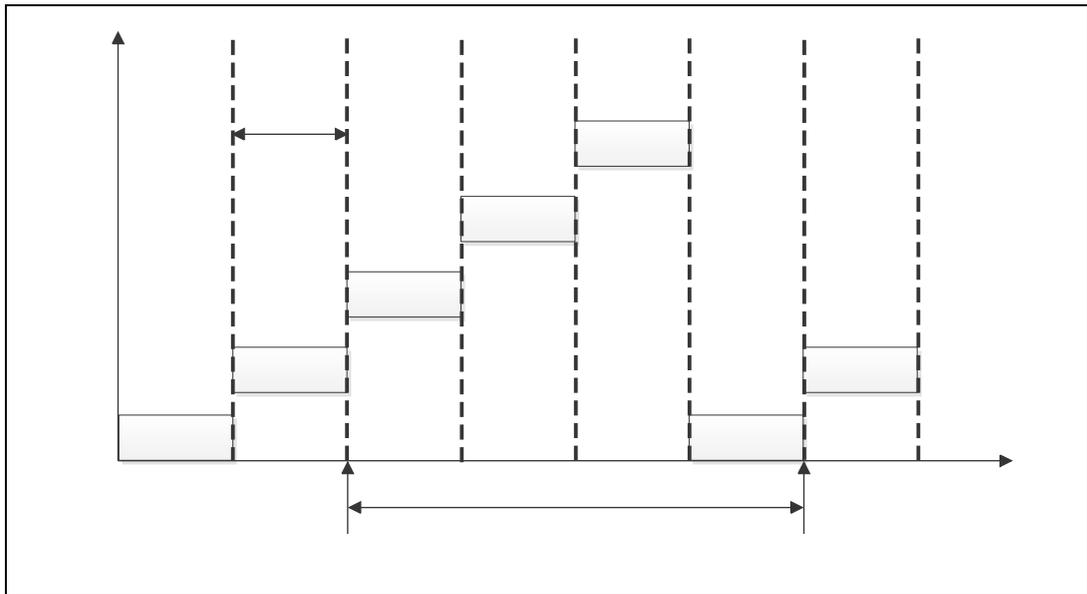
Рисунок 8. Процессор и DMA1 запросить доступ к SRAM1



В случае шины параллелизма доступа, как в приведенном выше примере, требуется шина матрицы арбитража. Политика Круговой затем применяется для того, чтобы решить эту проблему: если последний мастер, который тусклый автобус был ЦП, в течение следующего DMA1 доступа выигрывает автобус и доступ SRAM1 первым. Процессор имеет то права на SRAM1 доступа.

Это доказывает, что передача задержки, связанная с одним мастера зависит от количества других ожидания мастера-запросов на доступ к тем же АНВ раба. В следующем примере (*На рисунке 9*), пять мастеров пытаются получить доступ одновременно SRAM1.

Рисунок 9. Пять мастеров запроса доступа SRAM



Задержка связана с DMA1, чтобы выиграть матрицу шины снова и SRAM1 доступ (к примеру), равно времени выполнения всех ожидающих запросов, поступающих от других мастеров.

2.1.3 BusMatrix арбитраж и передача DMA задерживает наихудший случай

Задержки видели основной порт DMA на одной транзакции зависит от типов передачи других мастеров и длиной.

Например, если мы рассмотрим предыдущий DMA1 & пример CPU (*Рисунок 8*) с параллельностью к SRAM доступа, задержки при передаче DMA варьируется в зависимости от длины транзакций центрального процессора.

Если доступ к шине сначала предоставляется процессор и процессор не выполняет ни одной загрузки данных / магазина, время DMA ждать, чтобы получить доступ к SRAM может расширяться от одного АНВ цикла для одной загрузки данных / хранилища для N АНВ циклов, где N есть число слов данных в транзакции CPU.

Процессор блокирует АНВ шины, чтобы сохранить право собственности и уменьшает время ожидания во время нескольких операций загрузки / сохранения и прерывания записи. Это повышает прошивку отзывчивости, но это может привести к задержкам по сделке DMA.

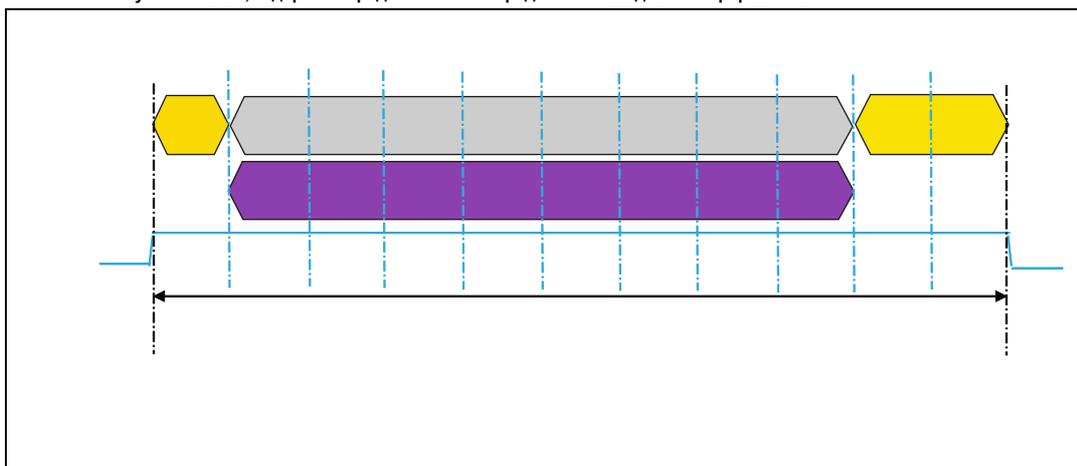
Задержка доступа DMA1 SRAM, когда в параллелизм с процессором зависит от типа передачи:

- передача CPU выданная прерывания (контекст сохранения): 8 циклов АНВ
- Передача ЦПА, выданная контроллером кэша (256 битого заполнение строки кэша / выселение): 8 циклов АНВ (а)
- Передача ЦПА выдана инструкция LDM / STM: 14 циклов АНВ (б)
 - Передача до 14 регистров из / в память

а. Только для устройств STM32F7xx

б. Задержка в связи с передачей выданной инструкцией LDM / STM может быть уменьшено путем конфигурирования компилятора расцепленной загрузки / сохранения нескольких инструкций в единые инструкции загрузки / сохранения.

Рисунок 10. DMA, задержка передачи за счет передачи CPU выданного прерывания



Выше рисунке подробно случай передачи DMA задержанного на передачу нескольких циклов процессора из-за прерывания записи. порт памяти DMA запускается для выполнения доступа к памяти. После арбитража шины АHB не предоставляется порт DMA1 памяти, но и CPU. Дополнительная задержка наблюдается для обслуживания запроса DMA. Это 8 АHB циклы для передачи CPU выданного прерывания.

Такое же поведение можно наблюдать с другими мастерами (например, DMA2, USB_HS, Ethernet, ...) при адресации одновременно один и тот же ведомый с длиной транзакции, отличной от одного блока данных.

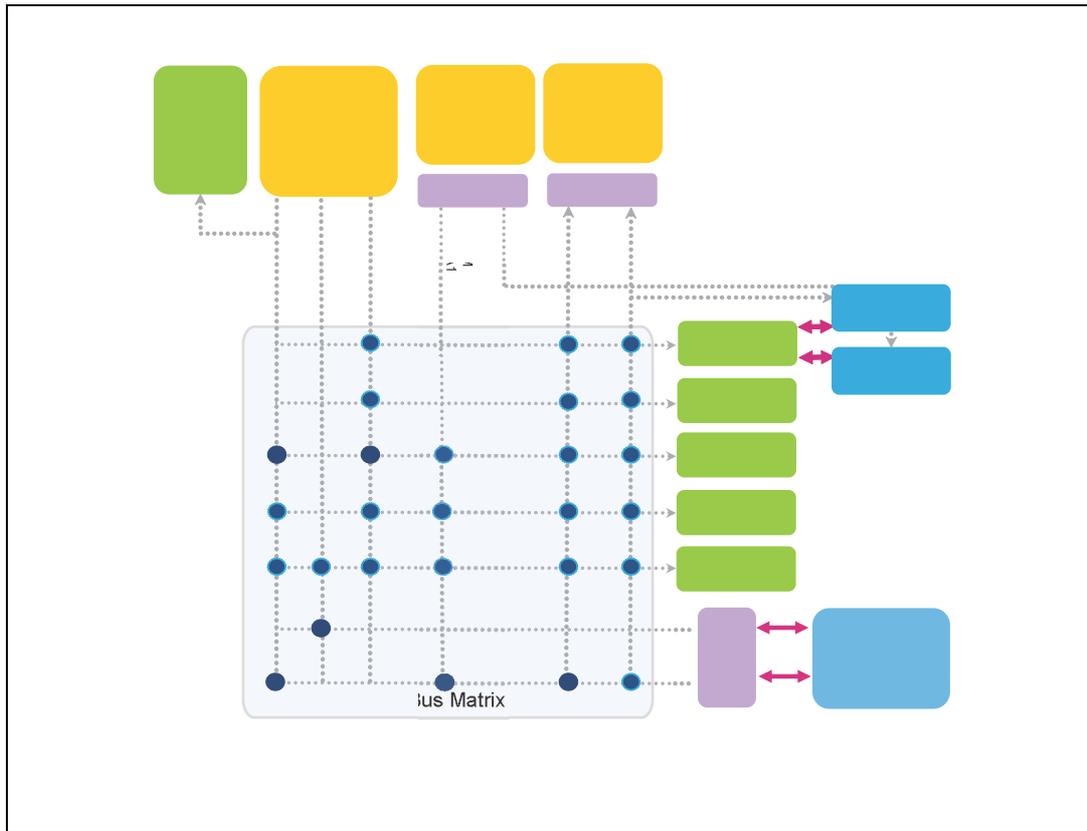
Для того, чтобы улучшить производительность доступа DMA через BusMatrix, рекомендуется, чтобы избежать конфликтов на шине.

2,2 Пути передачи DMA

2.2.1 порт Dual DMA

STM32F2 / F4 / F7 устройства вставлять две зоны управления дефектами. Каждый DMA имеет два порта, порт памяти и периферийный порта, которые могут работать одновременно не только на уровне DMA, но и с другими мастерами системы, используя внешнюю матрицу шины и выделенные пути DMA. Одновременная работа позволяет оптимизировать эффективность DMA и сократить время отклика (время ожидания между запросом и передачами данных).

Рисунок 11. DMA, два порта



Для DMA2:

- MEM (порт памяти) может получить доступ к АНВ1, АНВ2, АНВ3 (внешний контроллер памяти, ФКЦБ), SRAMs и флэш-памяти через матрицу шины.
- Периф (периферийный порт) может получить доступ к:
 - АНВ1, АНВ2, АНВ3 (внешний контроллер памяти, ФКЦБ), SRAMs и флэш-память через матрицу шины,
 - мост АНВ-к-АРВ2 посредством прямого пути (не пересекая матрицу шины). Для DMA1:
- MEM (порт памяти) может получить доступ к АНВ3 (внешний контроллер памяти, ФКЦБ), SRAMs и флэш-памяти через матрицу шины.
- Периф (периферийный порт) может получить доступ только к мосту АНВ-к-АРВ1 посредством прямого пути (не пересекая матрицу шины).

2.2.2 состояния передачи DMA

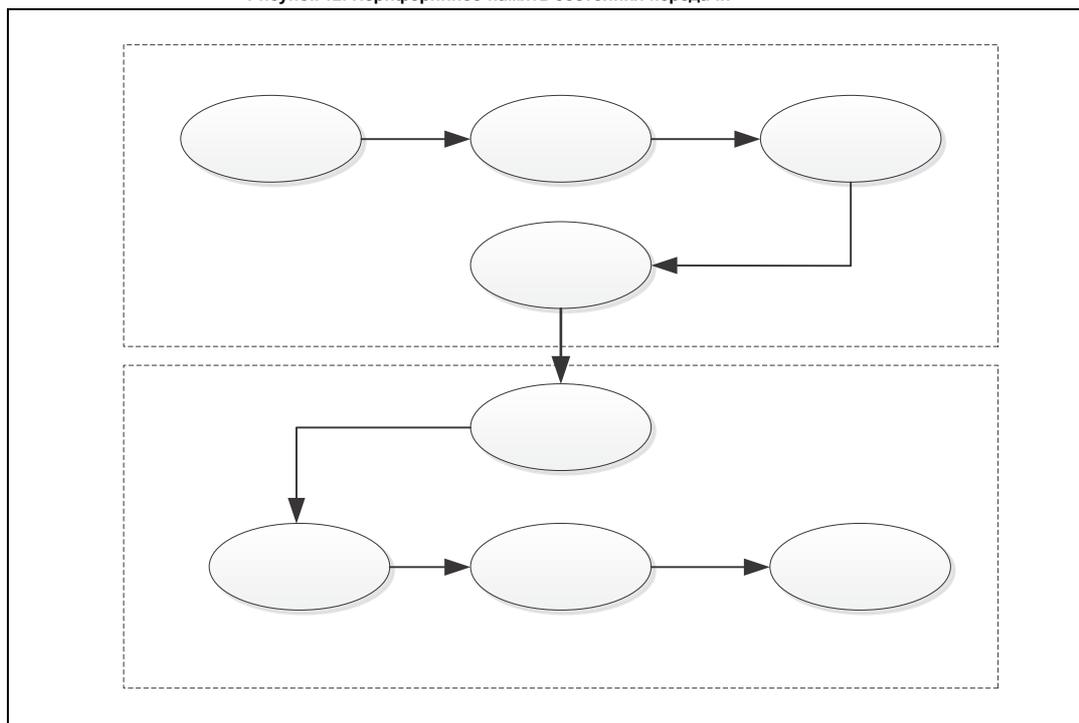
В этом разделе описываются этапы передачи DMA на уровне периферийного порта, а также на уровне порта памяти:

- Для передачи периферийного-памяти:

В этом режиме передачи, DMA, требует два доступов шины для выполнения передачи:

- Один доступа через периферийный порт запускается по запросу периферического, в
- Один доступ через порт памяти, который может быть вызван либо порога FIFO (когда используется режим FIFO) или сразу же после периферического чтения (если используется прямой режим).

Рисунок 12. Периферийное-память состояния передачи

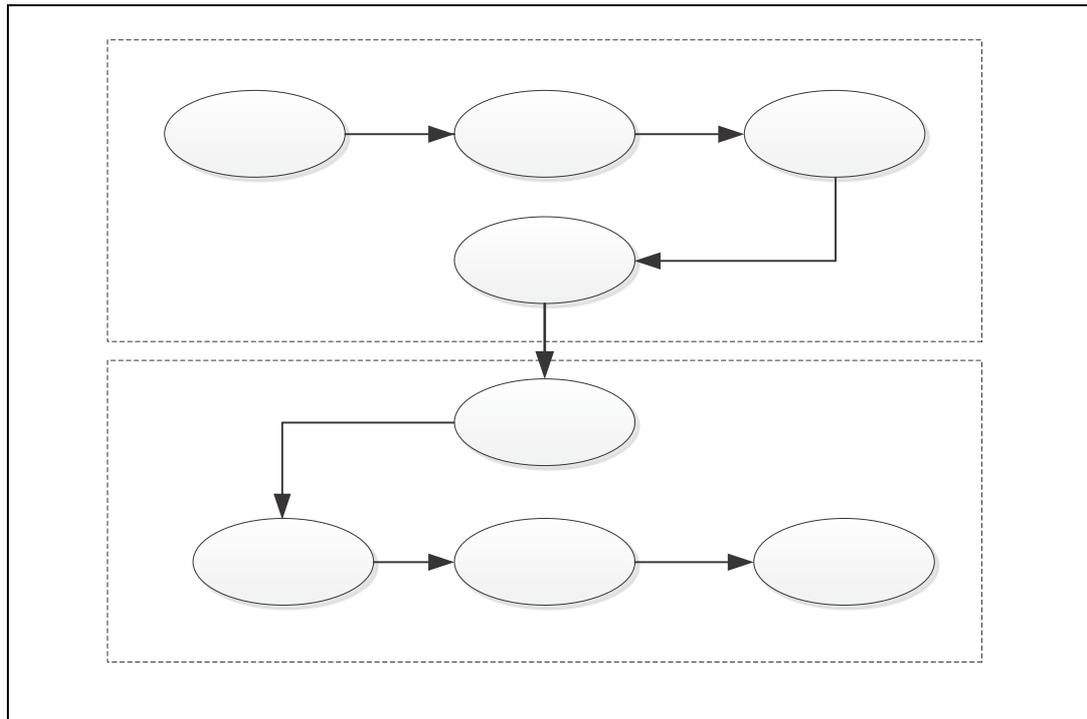


- Для памяти-к-периферийной передачи:

В этом режиме передачи, DMA, требует два доступов шины для выполнения передачи:

- DMA предполагает доступ Периферический и считывает данные из памяти и сохраняет его в FIFO, чтобы обеспечить немедленную передачу данных, как только периферийное запрос DMA срабатывает.
- Когда периферийный запрос инициируется, передача генерируется на периферийный порт DMA.

Рисунок 13. Память к периферийным состояний с переносом



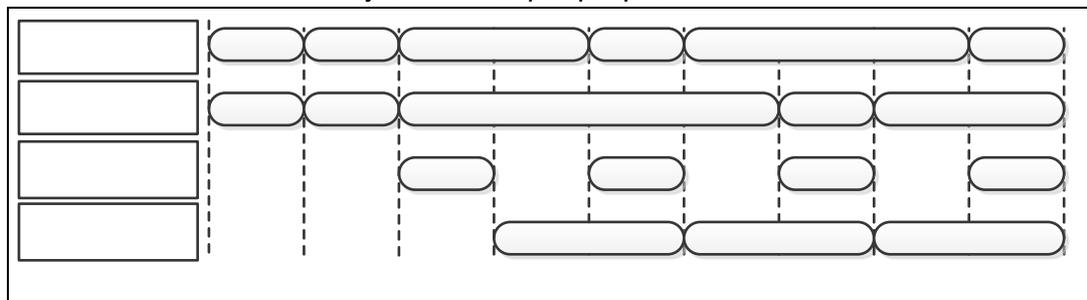
2.2.3 DMA запрос арбитража

Как описано в [Раздел 1.1.2: Стрим приоритет](#), STM32F2 / F4 / F7 DMA встраивает арбитр, который управляет восемь запросов DMA потоков на основе их приоритетов для каждого из двух основных портов (AHB памяти и периферийные порты) и запускают периферийные последовательности доступа / памяти.

Когда более чем один запрос DMA активен, DMA должен арбитром внутренне между активными запросами и решить, какой запрос должен быть подан первым.

На следующем рисунке показан две круговые запросы DMA срабатывает в то же время с помощью DMA потока «запрос 1» и DMA потока «запрос 2» (запросы 1 и 2 могут быть любым DMA периферийного запроса). На следующем AHB тактового цикла DMA арбитра проверки на активном ожидание запросов и предоставляет доступ к потоку «запрос 1», который имеет самый высокий приоритет. Следующий арбитражный цикл происходит в течение последнего цикла передачи данных потока «запрос 1». В то время, «запрос 1» маскируются и арбитр видит только «запрос 2» как активные, так что доступ зарезервирован к «запросу 2» на этот раз, и так далее.

Рисунок 14. DMA запрос арбитража



Общие рекомендации:

- Периферийные устройства высокоскоростной / с высокой пропускной способностью, должны иметь самые высокие приоритеты DMA. Это гарантирует, что латентность максимум данных уважаемая для этих периферийных устройств и чрезмерных / противоположный избегают условий.
- В случае одинаковых требований к пропускной способности, рекомендуется назначить более высокий приоритет периферийных устройств, работающих в режиме ведомого (которые не имеют никакого контроля на скорости передачи данных) по сравнению с теми, работающих в режиме Master (который может управлять потоком данных).
- Поскольку эти два Медиаинки могут работать параллельно на основе многослойной структуры шину матрицы, запросы высокоскоростные периферийные устройства могут быть сбалансированы между двумя DMA, когда это возможно.

2.3 АНВ-к-АПБ моста

STM32F2 / F4 / F7 устройства вставлять две АНВ-к-АПБ мосты, APB1 и APB2, к которым подключены периферийные устройства.

2.3.1 Двойной АНВ к APB порту

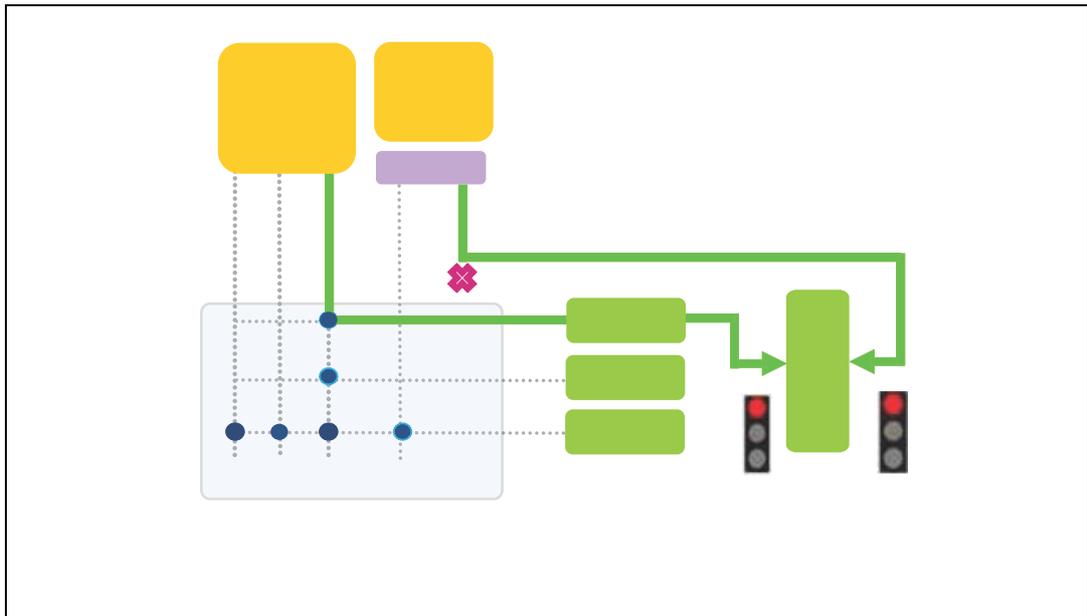
Мост АНВ-к-APB представляет собой архитектуру двойного порта, который позволяет получить доступ через два различных пути:

- Прямой путь (не пересекая матрицу шины), которые могут быть получены из DMA1 в APB1 или из DMA2 в APB2; в этом случае доступ не штрафует шины матрицы арбитра.
- Общий путь (через матрицу шины), которые могут быть получены либо из CPU или из DMA2, который нуждается в автобусе матрицы арбитража, чтобы выиграть автобус.

2.3.2 АНВ-к-АПБ моста арбитраж

Благодаря реализации прямых путей DMA на этих продуктах, арбитраж реализуются на уровне моста АНВ-к-APB для решения параллельных запросов доступа. На приведенном ниже рисунке показан запрос параллельного доступа на АНВ-APB1 моста, генерируемого процессором (доступ через матрицу шины) и DMA1 (доступ посредством прямого пути).

Рисунок 15. АНВ-к-АРВ1 мост одновременно процессор и запрос доступа DMA1



Чтобы предоставить доступ к шине, мост АНВ-АРВ применяет политику циклического:

- Круговой квант передачи 1x АРВ.
- Максимальная задержка на DMA периферийного порта ограничена (1 передачи АРВ). Только CPU и DMA может

генерировать параллельный доступ к АРВ1 и АРВ2 автобусов:

- Для АРВ1, одновременное доступа может быть сгенерировано, если процессор, DMA1 и / или DMA2 запросить одновременный доступ.
- Для АРВ2, одновременное доступа может быть создано, если процессор и DMA2 запросить одновременный доступ.

3 Как предсказать DMA латентности

При разработке приложения встроенного программного обеспечения, основанного на микроконтроллер, пользователь должен убедиться, что нет опустошения / перерасход может произойти, и поэтому зная точную DMA задержки для каждой передачи является обязательным, чтобы проверить, если внутренняя система может поддерживать общие данные пропускной способности, необходимой для заявления.

3,1 время передачи DMA

3.1.1 По умолчанию DMA времени передачи

Как описано в [Раздел 2.2.2](#), для выполнения передачи DMA из периферийного устройства в память, необходимы два доступа шины:

- Один доступ через периферийный порт инициируется периферийным запрос, который должен:
 - DMA запрос периферийного порта арбитраж
 - Периферийное адрес вычисления
 - Считывание данных из периферийного устройства к DMA FIFO (источник DMA)
- Один доступа через порт памяти, которая может быть вызвана порогом FIFO (когда используется режим FIFO) или сразу же после периферического чтения (если используются прямой режим), который должен:
 - DMA память запрос порта арбитраж
 - адрес вычисления памяти
 - Запись данных, загруженных в ОЗУ (DMA, место назначения)

При передаче данных из памяти в периферийное, два доступа также необходимы, как описано в [Раздел 2.2.2](#) :

- Первый доступ: DMA ожидает периферический доступ и считывает данные из памяти и сохраняет его в FIFO, чтобы обеспечить немедленную передачу данных, как только DMA периферийное запрос срабатывают. Эта операция потребности:
 - DMA память запрос порта арбитраж
 - адрес вычисления памяти
 - Считывание данных из памяти DMA FIFO (источник DMA)
- Второй доступ: когда периферийное запрос инициируется, передача генерируется на DMA периферийного порта. Эта операция потребности:
 - DMA запрос периферийного порта арбитраж
 - Периферийное адрес вычисления
 - Запись загруженных данных на периферийном адрес (DMA назначения) По

общему правилу, общее время передачи по DMA поток T_s равно:

$$T_s = T_{SP} (\text{периферийный доступ / время передачи}) + T_{SM} (\text{доступ к памяти время / передача}) C:$$

T_{SP} это общее время для DMA периферийного порта доступа и передачи, который равен: $T_{SP} = T_{PA} + T_{PAC} + T_{VMA} + T_{Po}$ восточному времени + T_{BS}

Где:

Таблица 4. Периферийного порта время доступа / передач по сравнению с путем DMA используется

Описание	Через шину матрицы		прямые пути DMA в
	Чтобы АНВ периферия	Для APB периферия	
Т РА: DMA периферийного арбитража порт	1 АОБ цикл	1 АОБ цикл	1 АОБ цикл
Т РАС: вычисление периферийного адреса	1 АОБ цикл	1 АОБ цикл	1 АОБ цикл
Т вма: шины матрицы арбитража (при отсутствии параллельности) (1)	1 АОБ цикл	1 АОБ цикл	N / A
Т по восточному времени: эффективная передача данных,	1 АОБ цикл (2) (3)	2 цикла APB	2 АПБ цикла
Т вS: синхронизация автобуса	N / A	1 АОБ цикл	1 АОБ цикл

1. В случае STM32F401 / STM32F410 / STM32F411 / STM32F412 линии, т БМА равна нулю.

2. Для КФМ, дополнительный цикл может быть добавлен в зависимости от внешней памяти, используемой. Дополнительные циклы АНВ добавляются в зависимости от внешних таймингов памяти.

3. В случае взрыва, зависит от длины пакета (т EDT МКП-4 = 4 циклов АНВ) эффективное время передачи данных.

• **Т SM это общее время для доступа к портам памяти DMA и передач, который равен: $T_{SM} = T_{MA} + T_{MAC} + T_{VMA} + T_{SRAM}$**

Где:

Таблица 5. Память Время порт доступа / передачи Описание

	Задержка
Т МА: порт памяти DMA арбитраж	1 АОБ цикл
Т МАС: адрес вычисления памяти	1 АОБ цикл
Т вма: шины матрицы арбитража (при отсутствии параллельности) (1)	1 АОБ цикл (2)
Т SRAM: SRAM чтения или записи	1 АОБ цикл

1. В случае STM32F401 / STM32F410 / STM32F411 / STM32F412 линии, т БМА равна нулю.

2. Для последовательных доступов SRAM (в то время как ни один другой мастер не получает доступ к той же SRAM в промежутке между), т = 0 ВМА цикла.

3.1.2 DMA время передачи против одновременного доступа

Дополнительная задержка может быть добавлена к времени DMA службы, описанному в [Раздел 3.1.1](#) когда несколько мастеров пытаются получить доступ одновременно к тому же рабе.

Для периферического и памяти в худшем случае времени доступа / передачи, следующие факторы влияют на общее время задержки для потока службы DMA:

- Когда несколько мастеров доступ к тому же АНВ назначения одновременно, DMA, латентность воздействие; передача DMA не может начаться, пока автобус матрица Арбитр не предоставляет доступ к DMA, как описано в [Раздел 2.1.2](#),
- Когда несколько мастеров (DMA, и центральный процессор) имеет доступ к тому же мосту АНВ-к-APB, время передачи DMA задерживается из-за мост арбитража АНВ-к-APB, как описано в [Раздел 2.3.2](#),

3,2 Примеры

3.2.1 Передача АЦП к SRAM DMA

Этот пример применим к продуктам STM32F2, STM32F405 / 415, STM32F407 / 417, STM32F427 / 437 и STM32F429 / 439 строк.

АЦП сконфигурирован в непрерывном режиме тройного Interleaved. В этом режиме, он преобразует входной канал непрерывно один аналоговый со скоростью максимального АЦП (36 МГц). АЦП предделитель установлен в 2, время выборки устанавливаются на 1,5 циклов, а задержка между двумя последовательными выборками АЦП режима Interleaved установлена на 5 циклов. Переводы stream0 DMA2 АЦП преобразуется значение в буфер SRAM. доступ DMA2 к АЦП осуществляется через прямой путь; однако, DMA доступ к SRAM осуществляется через матрицу шины.

Таблица 6. Передача DMA периферийного (АЦП) порт задержки АНВ /

частота APB2	F АНВ = 72 МГц / F, APB2 = 72 Соотношение МГц АНВ / АПБ = 1	F АНВ = 144 МГц / FAPB2 = 72 МГц Отношение АНВ / АПБ = 2
время передачи		
T РА: DMA периферийного арбитража порт	1 АОБ цикл	1 АОБ цикл
T РАС: вычисление периферийного адреса	1 АОБ цикл	1 АОБ цикл
T вма: автобус матрица арбитража	N / A (1)	N / A (1)
T по восточному времени: эффективная передача данных,	2 цикла АНВ	4 циклов АНВ
T вs: синхронизация автобуса	1 АОБ цикл	1 АОБ цикл
T сР: Общее время передачи DMA для периферийного порта	5 циклов АНВ	7 циклов АНВ

1. DMA2 доступ к АЦП через прямой путь: нет шины матрицы арбитража.

Таблица 7. DMA памяти (SRAM) проходное отверстие задержки процессора /

частота APB2	F АНВ = 72МГц / FAPB2 = 72МГц АНВ / отношение АПБ = 1	F АНВ = 144МГц / FAPB2 = 72МГц АНВ / отношение АПБ = 2
время передачи		
T МА: порт памяти DMA арбитраж	1 АОБ цикл	1 АОБ цикл
T МАс: адрес вычисления памяти	1 АОБ цикл	1 АОБ цикл
T вма: автобус матрица арбитража	1 АОБ цикл (1)	1 АОБ цикл (1)
T SRAM: Доступ к записи SRAM	1 АОБ цикл	1 АОБ цикл
T см: Общее время передачи DMA для порта памяти	4 циклов АНВ	4 циклов АНВ

1. В случае множественного доступа с DMA к SRAM, автобус матрица арбитража равно 0 цикла, если никакой другой мастер не обращались к SRAM между ними.

В этом примере, общая задержка DMA от АЦП DMA триггера (АЦП ЭОП), чтобы записать значение АЦП на SRAM равна 9 циклов АНВ для АНВ / APB предделителем равен 1 и 11 циклов АНВ для АНВ / APB предделителем равен 2.

Замечания:

При использовании буфера FIFO, доступ в порт памяти DMA запускается при достижении уровня FIFO настроенный пользователем.

3.2.2 SPI полнодуплексный передачи DMA

Этот пример применим к продуктам STM32F2, STM32F405 / 415, STM32F407 / 417, STM32F427 / 437 и STM32F429 / 439 линий, и на основе SPI1 периферии. Два DMA запросы настроены:

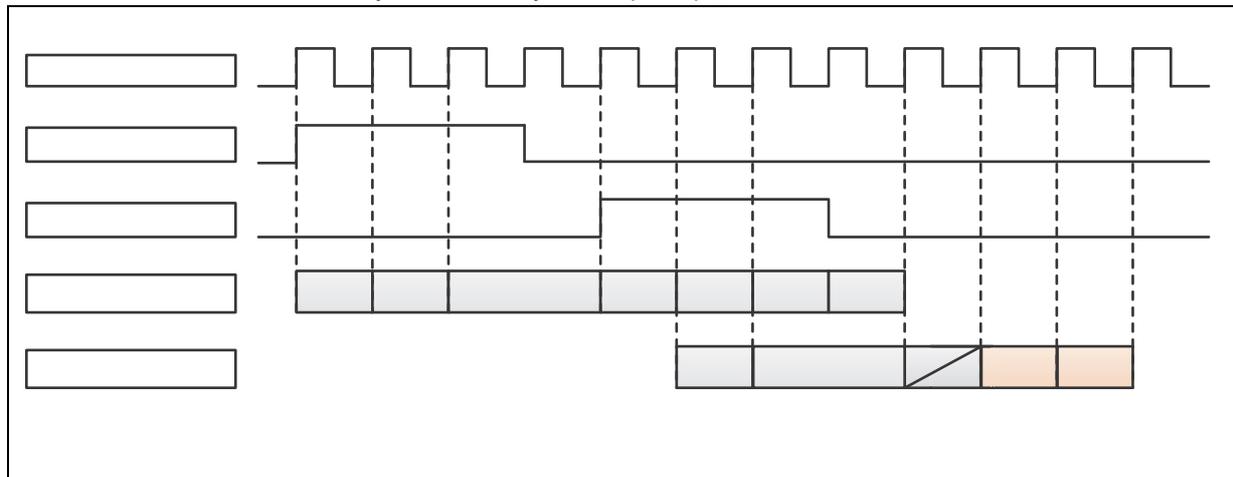
- DMA2_Stream2 для SPI1_RX: этот поток сконфигурирован, чтобы быть наивысшим приоритетом для того, чтобы служить во времени SPI1 полученные данные и передавать его из регистра SPI1_DR в буфер SRAM.
- DMA2_Stream3 для SPI1_TX: этот поток передает данные из буфера SRAM в регистр SPI1_DR.

Частота АОБ равна частоте APB2 (84 МГц) и SPI1 выполнен с возможностью работы на максимальной скорости (42 МГц). DMA2_Stream2 (SPI1_RX) срабатывает перед тем DMA2_Stream3 (SPI1_TX), который запускается два цикла АНВ позже. При такой конфигурации, процессор опрашивает бесконечно в регистре I2C1_DR. Зная, что I2C1 периферийное отображается на APB1 и что SPI1 периферийное отображается на APB2, системные пути, являются следующие:

- Прямой путь для DMA2 к APB2 доступа (не через шину матрицы),
- Процессор получает доступ к APB1 через шину матрицы.

Цель состоит в том, чтобы продемонстрировать, что тайминги DMA не влияет на опрос ЦП на APB1. На приведенном ниже рисунке приведен временные DMA для режимов передачи и приема, а также планирование времени для каждой операции:

Рисунок 16. SPI полнодуплексное время передачи DMA



Этот рисунок иллюстрирует следующие выводы:

- CPU опрос на APB1 не влияет на задержку передачи DMA на APB2.
- Для сделки DMA2_Stream2 (SPI1_RX), на восьмом цикле АНВ часов, нет шины матрицы арбитража, так как предполагается, что последний мастер, который получил доступ к SRAM является DMA2 (так не требуется повторного арбитража).
- Для операции DMA2_Stream3 (SPI1_TX), этот поток ожидает чтения из SRAM и записывает его на FIFO, а затем, как только срабатывает, то периферийный порт DMA (назначения SPI1) начинается операция.
- Для DMA2_Stream3, периферийная арбитража фаза DMA (1 АОБ цикл) выполняется во время цикла синхронизации автобуса DMA2_Stream2.

Такая оптимизация всегда выполняется, как это, когда запрос DMA срабатывает до конца текущей транзакции запроса DMA.

4 Советы и предупреждения при программировании контроллера DMA

4,1 Последовательность Программного обеспечения для отключения DMA

Чтобы отключить периферийное устройство, соединенное с запросом DMA потока, он является обязательным для:

1. выключить поток DMA, к которому подключен периферийная,
2. ждать, пока бит EN в регистре DMA_SxCR не будет сброшен («0»).

Только тогда можно Периферическая безопасно отключить. Запрос DMA битого разрешение в регистре периферийного управления должно быть сброшены («0»), чтобы гарантировать, что любой ожидающий запрос от периферийной стороны очищается.

Замечания:

В обоих случаях Transfer Complete Флаг прерывания (TCIF в DMA_LISR или DMA_HISR) устанавливается для указания конца передачи из-за потока DISABLE.

4,2 Управление флаг DMA перед включением новой передачи

Перед включением новой передачи, пользователь должен убедиться, что Transfer Complete Флаг прерывания (TCIF) в DMA_LISR или DMA_HISR очищается.

В качестве общей рекомендации, рекомендуется очистить все флаги в регистрах DMA_LIFCR и DMA_HIFCR перед началом новой передачи.

4,3 Последовательность программного обеспечения для включения DMA

Следующая последовательность программного обеспечения применяется при включении DMA:

1. Конфигурация подходящий поток DMA.
2. Включить поток DMA, используемый (установить EN бит в регистре DMA_SxCR).
3. Включить периферийные б. Запрос DMA включить бит в регистре периферийного управления должны быть установлены («1»).

Замечания:

Если пользователь дает использованную периферийный до соответствующего DMA потока, а (FIFO, флаг ошибка прерывания) «FEIF» может быть установлена в связи с тем, ДМО не готово предоставить первый требуемые данные периферийного устройству (в случае Методу- к периферийной передаче).

4,4 Память-память, а передача NDTR = 0

При настройке потока DMA для выполнения передачи памяти в память в нормальном режиме, как только NDTR достигает 0, Transfer Complete устанавливается. В это время, если пользователь устанавливает бит разрешения (EN бит в DMA_SxCR) этого потока, передача память-память автоматически снова повторно запускается с последним значением NDTR.

4.5 DMA периферийного разрыва с Pinc / MINC = 0

DMA Выброс функция с периферийным адресом приращением (Pinc) или адрес памяти приращением (MINC) отключить позволяет решать внутренние или внешние (периферийные устройства) ФКЦБ поддерживающих Серийн.съемк

(Вложение FIFOs). Этот режим гарантирует, что этот поток DMA не может быть прерван другим DMA потоком во время своих операций.

4,6 Дважды отображенных DMA запросы

Когда пользователь конфигурирует два (или больше) DMA потоков служат тот же периферийный запрос, программное обеспечение должно гарантировать, что текущий поток DMA полностью отключен (по опросу EN бит в регистре DMA_SxCR) перед включением нового потока DMA.

4,7 Лучший DMA пропускной способности конфигурации

При использовании STM32F4xx с пониженной частотой АНВ в то время как DMA обслуживает высокоскоростного периферийного устройства, рекомендуется, чтобы поместить в стек и кучу в СКК (которые могут быть направлены непосредственно в ЦПУ через D-Bus) вместо того, чтобы их на SRAM, что создаст дополнительный параллелизм между CPU и DMA доступ к памяти SRAM.

4,8 Передача подвески DMA

В любой момент времени, передача DMA, может быть приостановлено, чтобы быть перезапущен позже или быть окончательно отключен до конца передачи DMA. Есть два случая:

- Поток отключает передачу с не позднее, при перезапуске с точки, где оно было остановлено: нет никаких особых действий, чтобы сделать, за исключением того, чтобы очистить EN бит в DMA_SxCR регистре, чтобы отключить поток и подождать, пока бит EN не сбрасывается, Как следствие:
 - Регистр DMA_SxNDTR содержит количество оставшихся элементов данных в тот момент, когда поток был остановлен, так что программное обеспечение может определить, сколько элементов данных были переданы до поток был прерван.
- Поток приостанавливает передачу для того, чтобы возобновить ее позже повторное включение потока: перезапустить с того момента, когда была остановлена передача, программа должна прочитать регистр DMA_SxNDTR после отключения потока (EN укусил «0») в знать количество элементов данных, уже собранные. Потом:
 - Периферийный и / или адрес памяти должен быть обновлен, чтобы настроить адресные указатели.
 - Регистр SxNDTR должен быть обновлен с оставшимся количеством элементов данных, подлежащих передаче (считанное значение, когда поток был отключен).
 - Поток может затем быть повторно включен, чтобы перезапустить передачу с того момента, когда оно было остановлено.

Замечания:

В обоих случаях Transfer Complete Флаг прерывания (TCIF в DMA_LISR или DMA_HISR) устанавливается для указания конца передачи из-за прерывания потока.

4,9 Возьмите преимущества DMA2 контроллера и архитектуры системы гибкости

Идея этого раздела, чтобы показать, как воспользоваться преимуществами от гибкости предлагаемых STM32 архитектуры и контроллер DMA. В качестве иллюстрации этой гибкости, мы увидим, как инвертировать DMA2 АНВ периферийных и порты памяти и сохранить правильное управление передач данных периферийных устройств. Для достижения этой цели и взять на себя контроль над регулярным поведением DMA2 нам необходимо пересмотреть действующую модель DMA2.

Поскольку оба порта DMA2 подключены к АНВ BusMatrix, и имеющих симметричное соединение с АНВ рабов, эта архитектура позволяет транспортный поток в одном или другом направлении через порты периферийных и памяти в зависимости от конфигурации программного обеспечения.

4.9.1 Инверсия переводы через DMA2 АНВ рассмотрение портов

Программное обеспечение обладает гибкостью в режиме передачи настроить DMA2 потока в соответствии с его потребностями. В зависимости от этой конфигурации один DMA2 АНВ порт будет запрограммирован в направлении чтения, а другой в направлении Write.

Таблица 8 показывает направление DMA АНВ порта конфигурации режим передачи по сравнению с.

Передача направление порта в сравнении режима конфигурации Режим передачи DMA Таблица 8. АНВ

	порт памяти DMA2 АНВ	DMA2 АОБ периферийный порт
Память для периферийных	Читать направление	Написать направление
Периферийные к памяти	Написать направление	Читать направление
Память для памяти	Написать направление	Читать направление

Теперь внимание на транспортном потоке, как показано в [Раздел 2.2.2: состояния передачи DMA](#), Переводы по порту Peripheral вызваны периферийными запросами, передает в порте памяти вызывается либо порог FIFO (когда используется режим FIFO) или сразу же после периферического чтения (если используются прямой режим).

При управлении периферийными устройствами с портом памяти DMA2 мы должны позаботиться о заранее запускаемых переводах, переводов, буферизации и последнего управления данными:

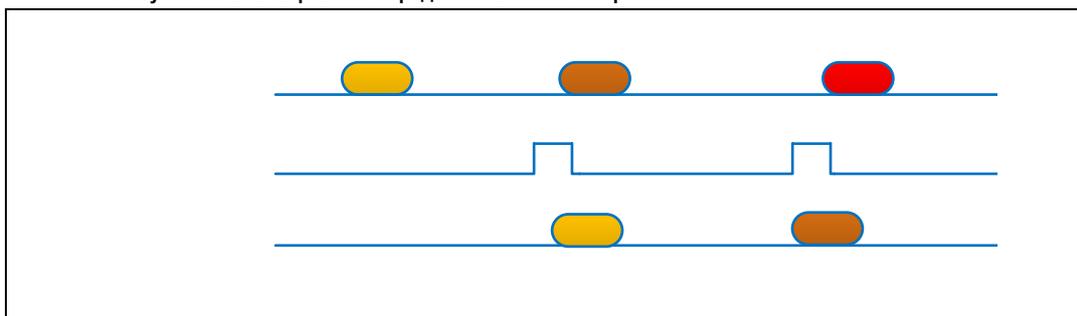
Pre-триггерным перевод:

Как описано в [Раздел 2.2.2: состояния передачи DMA](#) когда память в режим периферийной передачи сконфигурирована (считывание данных через порт памяти) DMA ожидает доступ периферического и считывает данные, как только DMA поток включен. Один данные буферизуются в прямом режиме и до 4 x 32-битовых слов, когда DMA FIFO включена.

При управлении периферийным считывает через порт памяти DMA2, программное обеспечение должно гарантировать, что периферический включено до включения DMA для того, чтобы гарантировать действия первого доступа DMA.

Рисунок 17 DMA иллюстрирует доступ по памяти и периферийных портов vs. периферийных триггеров.

Рисунок 17. DMA в режиме передачи памяти-к-Peripheral

**Управление последнего чтения данных**

Контроллер DMA оснащен 4 x 32-битные слова FIFO для каждого потока используются для буферизации данных между портами AHB. При управлении периферийным считывает через порт памяти DMA, программное обеспечение должно гарантировать, что 4x дополнительных слова считываются из периферийного устройства. Это является гарантией того, что последние действительные данные передаются из-DMA FIFO

Буферизация переводов при прямом режиме DMA отключен:

При записи данных через порт памяти на периферию в косвенном режиме (режим FIFO включен), потребности программного обеспечения позаботиться о том, доступ через этот порт запускаются запрограммированным порогом FIFO. При достижении порогового значения данные передаются-из FIFO до пункта назначения через порт памяти.

При записи в регистр (например GPIOs не имеющий FIFO), данные из DMA FIFO будут записаны последовательно до пункта назначения.

И последнее, но не в последнюю очередь, при замене периферийного управления от периферийного порта в порт памяти, потребности программного обеспечения для повторного рассмотрения размера передачи и адреса конфигурации приращения. Как описано в [Раздел 1.1.5: Размер Transfer](#), Размер передачи определяется шириной периферийной стороны передачи (байт, полуслово, слово) и число элементов данных, подлежащая передаче (значение программируется в регистре DMA_SxNTDR). В соответствии с новой конфигурацией DMA, когда инвертирующие порты запрограммированного значение в регистре DMA_SxNTDR может должно быть скорректировано.

4.9.2 Пример для инвертирования переводов Quad-SPI над DMA2 AHB рассмотрения портов

В этом примере DMA_S7M0AR программируется с Quad-SPI регистра данных адреса, DMA_S7PAR запрограммирован с данными буфера адреса (например, буфер в ОЗУ). В регистре DMA_S7CR направление потока DMA2 должны быть сконфигурировано в периферийном устройстве для режима передачи памяти при записи на Quad-SPI. DMA2 направление потока должно быть сконфигурировано в памяти для режима передачи периферийного при чтении из Quad-SPI. 4x Дополнительные слова (32 бита) необходимы для чтения операции, для того, чтобы гарантировать, что последний данные передаются из-за DMA FIFO в память RAM.

Таблица 9. Фрагмент кода

Запись операции	операция чтения
<pre> /* Программа M0AR с QUADSPI регистра данных адреса */ DMA2_Stream7->M0AR = (uint32_t) & QUADSPI->DR; /* Программа PAR с адресом буфера */ DMA2_Stream7->PAR = (uint32_t) & u32Buffer [0]; /* Запись количество элементов данных передачи */ DMA2_Stream7->NDTR = 0x100; /* Настройка DMA: MSIZE = PSIZE = 0x02 (Слово), CHSEL = 0x03 (QUADSPI), Pinc = 1, DIR = 0x00 */ DMA2_Stream7->CR = DMA_SxCR_PSIZE_1 DMA_SxCR_MSIZ_1 3ul << 25 DMA_SxCR_PINC; /* Включить генерацию запроса DMA */ QUADSPI->CR = QUADSPI_CR_DMAEN; /* Написать DLR Регистрация */ QUADSPI->DLR = (0x100 * 4) -1; /* Запись в QUADSPI DCR */ QUADSPI->CCR = QUADSPI_CCR_IMODE_0 QUADSPI_CCR_ADMODE_0 QUADSPI_CCR_DMODE QUADSPI_CCR_ADSIZE QUAD_IN_FAST_PROG_CMD; /* Написать AR Регистрация */ QUADSPI->AP = 0x00ul; /* Включить выбранный DMA2_Stream7 путем установки бита * EN / DMA2_Stream7->CR = (uint32_t) DMA_SxCR_EN; /* Подождите окончания передачи */ в то время как ((QUADSPI->SR & QUADSPI_SR_TCF) = QUADSPI_SR_TCF!); </pre>	<pre> /* Программа M0AR с QUADSPI регистра данных адреса */ DMA2_Stream7->M0AR = (uint32_t) & QUADSPI->DR; /* Программа PAR с адресом буфера */ DMA2_Stream7->PAR = (uint32_t) & u32Buffer [0]; /* Запись количество элементов данных передачи */ DMA2_Stream7->NDTR = 0x100; /* Настройка DMA: MSIZE = PSIZE = 0x02 (Слово), CHSEL = 0x03 (QUADSPI), Pinc = 1, DIR = 0x01 */ DMA2_Stream7->CR = DMA_SxCR_PSIZE_1 DMA_SxCR_MSIZ_1 3ul << 25 DMA_SxCR_PINC DMA_SxCR_DIR_0; /* Включить генерацию запроса DMA */ QUADSPI->CR = QUADSPI_CR_DMAEN; /* Написать DLR Регистрация */ QUADSPI->DLR = ((0x100 * 4) * 4) -1; /* Запись в QUADSPI DCR */ QUADSPI->CCR = QUADSPI_CCR_IMODE_0 QUADSPI_CCR_ADMODE_0 QUADSPI_CCR_DMODE QUADSPI_CCR_ADSIZE QUADSPI_CCR_FMODE_0 QUAD_OUT_FAST_READ_CMD; /* Написать AR Регистрация */ QUADSPI->AP = 0x00ul; /* Включить выбранный DMA2_Stream7 путем установки бита * EN / DMA2_Stream7->CR = (uint32_t) DMA_SxCR_EN; /* Подождите окончания передачи */ в то время как ((DMA2_Stream7->CR & DMA_SxCR_EN) == DMA_SxCR_EN); </pre>

Замечания:

Ограничение повреждения данных, когда DMA2 является управляющим в параллельных AHB и APB2 передач (см опечаток продукта листов, чтобы определить влияние STM32F2 / F4 микроконтроллеров) может быть преодолено путем замена DMA2 периферийных и порты памяти, как описано в этом разделе.

4,10

Передача STM32F7 DMA и обслуживание кэша, чтобы избежать несовместимости данных

Когда программное обеспечение использует Cacheable области памяти для исходных DMA / или буферы назначения, он должен вызвать чистый кэш перед началом операции DMA, чтобы гарантировать, что все данные фиксируются в памяти подсистемы. При чтении данных из периферийного устройства после завершения передачи DMA, программное обеспечение должно выполнить Invalidate кэша перед чтением обновленной области памяти.

Предпочтительно использовать не Cacheable областей для DMA буферов. Программное обеспечение может использовать MPU, чтобы создать не-кашируемый блок памяти для использования в качестве совместно используемой памяти между процессором и DMA.

5 Вывод

Контроллер DMA предназначен для покрытия большинства встроенных прецедентов заявок:

- Предоставление гибкости для прошивки, чтобы выбрать подходящую комбинацию между 16 потоками X 16 каналов (восемь для каждого DMA),
- Сокращение общего времени задержки для передачи DMA, благодаря двойной АНВ архитектуры порта и прямой путь к APB мостов избежать стойло процессора на доступ АНВ1, когда DMA обслуживает низкоскоростные APB периферийных устройств,
- Реализация FIFOs на DMA обеспечивает большую гибкость в прошивке для настройки различных размеров данных между источником и пунктом назначения, а также скоростей передачи-апе при использовании режима пакетной передачи инкрементный.

6 Лист регистраций изменений

история изменений Таблица 10. Документ

Дата	пересмотр	изменения
04-февраля-2014	1	Первый выпуск.
06-августа-2015	2	<p>добавленной <i>Раздел: Справочные документы</i> , Сфера Документ распространяется на STM32F2 и STM32F4 серии и перечень справочников обновляется. Удалены примечание 1 в <i>Таблица 1: STM32F427 / 437 и STM32F429 / 439 отображение запроса DMA1</i> и <i>Таблица 2: STM32F427 / 437 и STM32F429 / 439 отображение запроса DMA2</i> , Удалены <i>Таблица DMA1 отображение запроса на STM32F401 линии</i> и <i>таблица DMA2 отображение запроса на STM32F401 линии</i>.</p> <p>обновленный <i>Раздел 4.1: последовательность Программного обеспечения для отключения DMA</i> и <i>Раздел 4.3: последовательность программного обеспечения для включения DMA</i> ,</p>
23-Jun-2016	3	<p>Добавлено:</p> <ul style="list-style-type: none"> - <i>Раздел 4.9: Возьмите преимущества DMA2 контроллера и архитектуры системы гибкости</i> - <i>Раздел 4.10: STM32F7 DMA передачи и кэш обслуживание, чтобы избежать несовместимости данных</i> <p>Обновлен:</p> <ul style="list-style-type: none"> - <i>Вступление</i> - <i>Раздел 2: соображения производительности системы</i> - <i>Раздел 2.1.3: BusMatrix арбитраж и DMA передача задерживает наихудший случай</i> - <i>Рисунок 14: DMA запрос арбитража</i> <p>Удалены:</p> <ul style="list-style-type: none"> - <i>Таблица 1: Применимые продукты</i>

ВНИМАНИЕ - ПРОЧИТЕ ВНИМАТЕЛЬНО

STMicroelectronics NV и ее дочерние компании («СТ») оставляют за собой право вносить изменения, исправление, усовершенствование, модификацию и усовершенствование в продукты ST и / или в данный документ в любое время без предварительного уведомления. Покупатели должны получить последнюю информацию о продуктах ST до размещения заказов. продукты ST продаются в соответствии с условиями и условиями продажи компании ST на месте в момент подтверждения заказа.

Покупателям несут полную ответственность за выбор, отбор и использование продукции ST и ST не несет никакой ответственности за помощь приложения или при проектировании изделий приобретателей.

Никакая лицензия, не явно выраженных или подразумеваемых, в какой-либо права интеллектуальной собственности, предоставляется ST в настоящем документе.

Перепродажа продукции ST с положениями отличаются от информации, изложенный в данном документе, аннулирует никакой гарантии, выданную ST для такого продукта.

ST и логотип ST являются товарными знаками ST. Все другие названия продуктов или услуг являются собственностью их соответствующих владельцев.

Информация, содержащаяся в этом заменяет документ и заменяет ранее информации, поставляется в любых предыдущих версиях этого документа.

© 2016 STMicroelectronics - Все права защищены