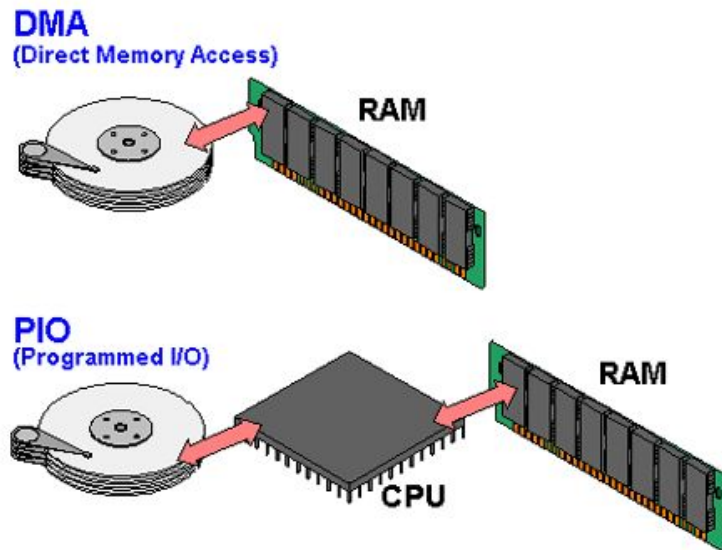


Embedded and OS design

Зачем нужен модуль прямого доступа к памяти? (DMA - direct memory access)

- Разгрузить основной процессор
 - Ядро инициирует работу DMA
 - Выполняет свои задачи
 - Получает прерывание по окончании работы DMA

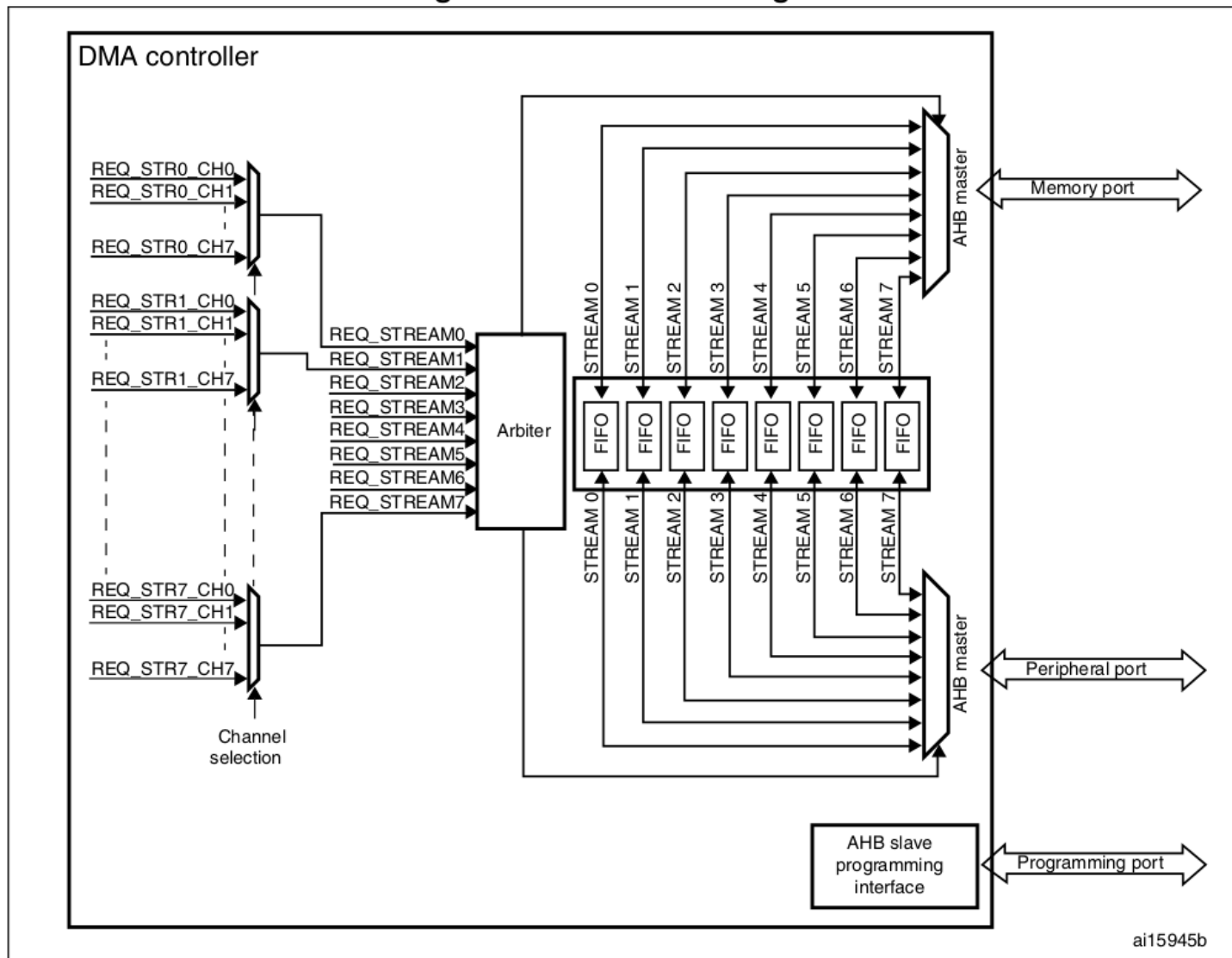


credits: pcmag.com

DMA в STM32F7

- 8 потоков по 8 каналов (запросов) в каждом
- FIFO буфферы на четыре 32-бит слова
- 4 программируемых уровня приоритета между каналами
- Передача периферия-память, память-периферия, память-память
- Поддержка двухбуферного режима
- Поддержка передачи данных различной ширины (слово, полуслово, байт)
 - Автоматическая конвертация форматов при использовании FIFO режима
- Автоинкремент указателей и поддержка кольцевого буфера

Figure 23. DMA block diagram



DMA Streams and Channels

- Каждый поток (stream) поддерживает до 8 источников запросов (channels)
- Возможные каналы для каждого потока заданы аппаратно и описаны в документации
- Выбор желаемых каналов происходит с помощью регистра DMA_SxCR

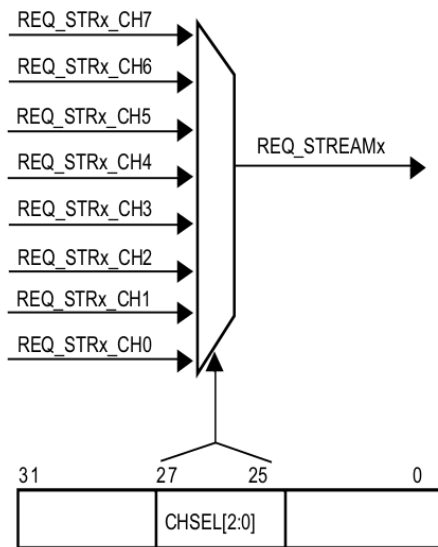


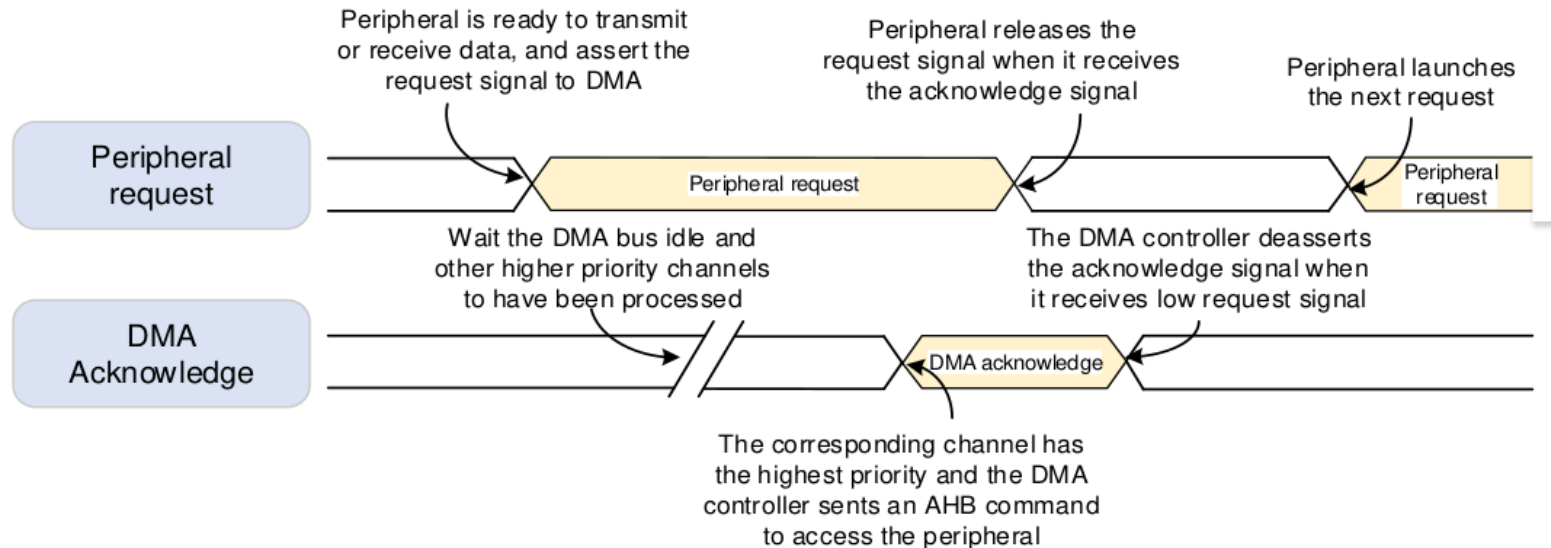
Table 25. DMA1 request mapping

Peripheral requests	Stream 0	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7
Channel 0	SPI3_RX	SPDIFRX_DT	SPI3_RX	SPI2_RX	SPI2_TX	SPI3_TX	SPDIFRX_CS	SPI3_TX
Channel 1	I2C1_RX	I2C3_RX	TIM7_UP	-	TIM7_UP	I2C1_RX	I2C1_TX	I2C1_TX
Channel 2	TIM4_CH1	-	I2C4_RX	TIM4_CH2	-	I2C4_RX	TIM4_UP	TIM4_CH3
Channel 3	-	TIM2_UP TIM2_CH3	I2C3_RX	-	I2C3_TX	TIM2_CH1	TIM2_CH2 TIM2_CH4	TIM2_UP TIM2_CH4
Channel 4	UART5_RX	USART3_RX	UART4_RX	USART3_TX	UART4_TX	USART2_RX	USART2_TX	UART5_TX
Channel 5	UART8_TX	UART7_TX	TIM3_CH4 TIM3_UP	UART7_RX	TIM3_CH1 TIM3_TRIG	TIM3_CH2	UART8_RX	TIM3_CH3
Channel 6	TIM5_CH3 TIM5_UP	TIM5_CH4 TIM5_TRIG	TIM5_CH1	TIM5_CH4 TIM5_TRIG	TIM5_CH2	-	TIM5_UP	-
Channel 7		TIM6_UP	I2C2_RX	I2C2_RX	USART3_TX	DAC1	DAC2	I2C2_TX

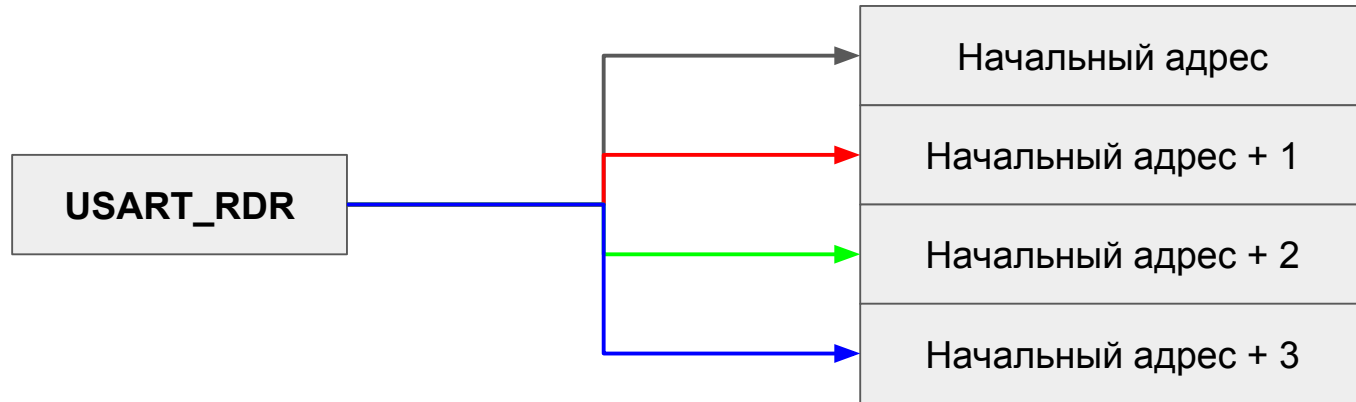
DMA Flow controller

- Flow controller — сущность, определяющая количество данных, которые должны быть переданы DMA
- Flow controller может быть:
 - Сам DMA, тогда количество передаваемых данных определяется регистром DMA_SxNDTR
 - Периферия, тогда данные передаются пока периферия не сигнализирует окончание передачи и количество переданных данных можно узнать с использованием регистра DMA_SxNDTR

Figure 9-2. Handshake mechanism



DMA. Параметры передачи



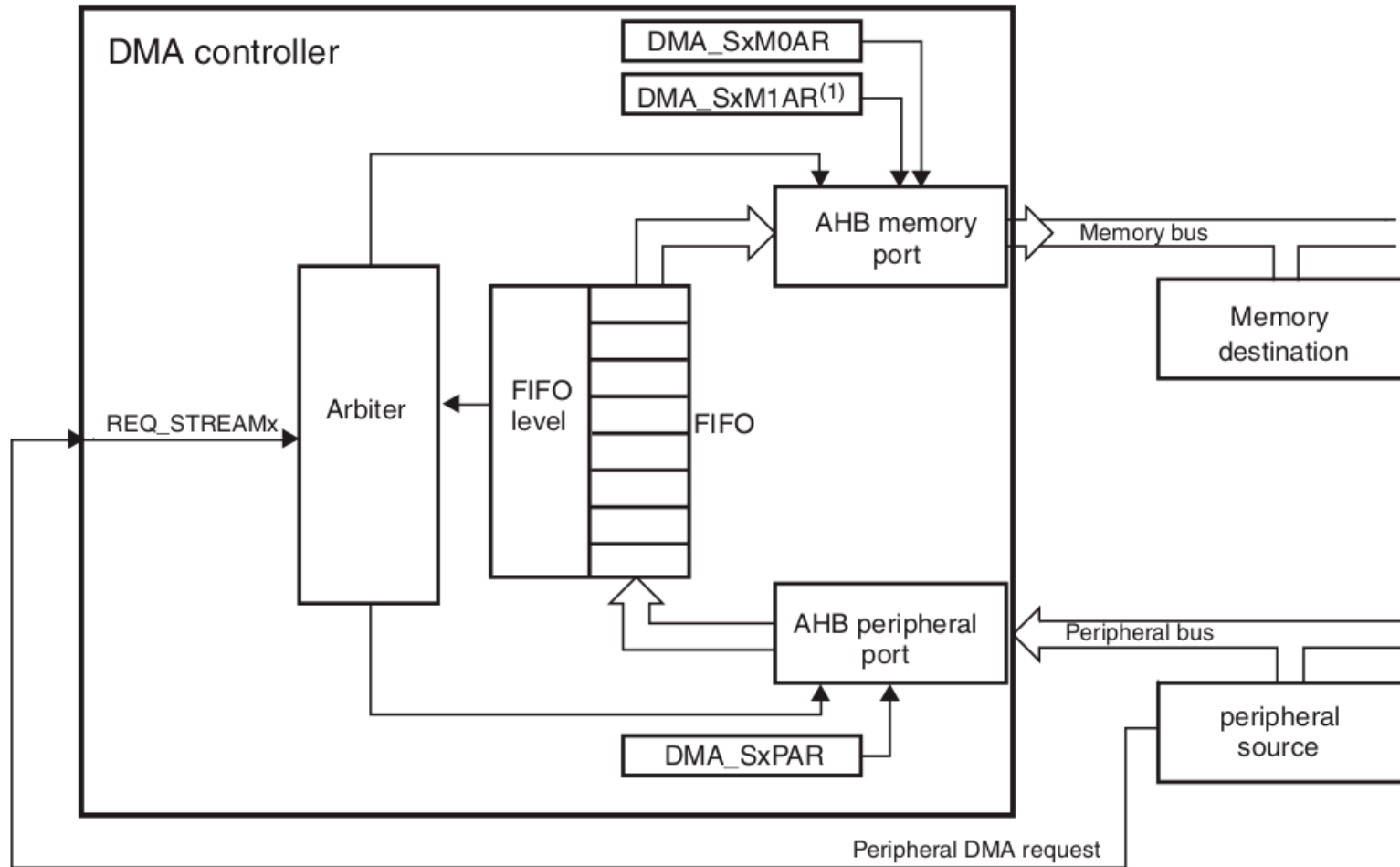
Размер сообщения конфигурируется как для источника, так и для получателя:
В данном случае: размер сообщения для источника - 1 байт (USART_RDR), размер
сообщения для получателя - 1 байт (SRAM)

DMA. Параметры передачи

Table 32. Possible DMA configurations

DMA transfer mode	Source	Destination	Flow controller	Circular mode	Transfer type	Direct mode	Double-buffer mode
Peripheral-to-memory	AHB peripheral port	AHB memory port	DMA	Possible	single	Possible	Possible
					burst	Forbidden	
			Peripheral	Forbidden	single	Possible	Forbidden
					burst	Forbidden	
Memory-to-peripheral	AHB memory port	AHB peripheral port	DMA	Possible	single	Possible	Possible
					burst	Forbidden	
			Peripheral	Forbidden	single	Possible	Forbidden
					burst	Forbidden	
Memory-to-memory	AHB peripheral port	AHB memory port	DMA only	Forbidden	single	Forbidden	Forbidden
					burst		

Figure 25. Peripheral-to-memory mode



DMA Stream initialization

1. If the stream is enabled, disable it by resetting the EN bit in the DMA_SxCR register, then read this bit in order to confirm that there is no ongoing stream operation. Writing this bit to 0 is not immediately effective since it is actually written to 0 once all the current transfers are finished. When the EN bit is read as 0, this means that the stream is ready to be configured. It is therefore necessary to wait for the EN bit to be cleared before starting any stream configuration. All the stream dedicated bits set in the status register (DMA_LISR and DMA_HISR) from the previous data block DMA transfer must be cleared before the stream can be re-enabled.
2. Set the peripheral port register address in the DMA_SxPAR register. The data is moved from/ to this address to/ from the peripheral port after the peripheral event.
3. Set the memory address in the DMA_SxMA0R register (and in the DMA_SxMA1R register in the case of a double-buffer mode). The data is written to or read from this memory after the peripheral event.
- 4) Configure the total number of data items to be transferred in the DMA_SxNDTR register. After each peripheral event or each beat of the burst, this value is decremented.
- 5) Select the DMA channel (request) using CHSEL[2:0] in the DMA_SxCR register.
- 6) If the peripheral is intended to be the flow controller and if it supports this feature, set the PFCTRL bit in the DMA_SxCR register.
- 7) Configure the stream priority using the PL[1:0] bits in the DMA_SxCR register.
- 8) Configure the FIFO usage (enable or disable, threshold in transmission and reception)
- 9) Configure the data transfer direction, peripheral and memory incremented/fixed mode, single or burst transactions, peripheral and memory data widths, circular mode, double-buffer mode and interrupts after half and/or full transfer, and/or errors in the DMA_SxCR register.
- 10) Activate the stream by setting the EN bit in the DMA_SxCR register.

DMA. Ошибки и Прерывания

- Детектируемые DMA ошибки:
 - Transfer error. Ошибки при чтении или записи данных
 - FIFO error. Underflow или overflow FIFO буфера
 - Direct mode error. Получение новых данных до окончания записи предыдущих
- Помимо ситуаций ошибок DMA поддерживает вызов прерываний при завершении половины транзакции и при полном завершении транзакции.

Table 33. DMA interrupt requests

Interrupt event	Event flag	Enable control bit
Half-transfer	HTIF	HTIE
Transfer complete	TCIF	TCIE
Transfer error	TEIF	TEIE
FIFO overrun/underrun	FEIF	FEIE
Direct mode error	DMEIF	DMEIE

Задание

- Добавить в задание по UART работу с DMA
 - Организовать получение данных с помощью DMA (можно использовать двухбуферный режим)
 - Организовать отправку строк ответов с помощью DMA