

Карта памяти MSP430F5529

4 x 32 Кб = 128 Кб Основная Flash-память (Bank A – Bank D)	Таблица векторов прерываний	243FFh 0FFFFh 0FF80h 04400h
4 x 2 Кб = 8 Кб RAM (Sector 0 – Sector 3)	Альтернативная таблица векторов	043FFh 02400h
2 Кб USB RAM (Sector 7)		023FFh 01C00h
Таблица дескрипторов устройств		01A00h–01A93h
4 x 128б = 512б Информационная Flash-память (Info A – Info D)		019FFh 01800h
4 x 512б = 2Кб Flash-память начальной загрузки (BSL0-BSL3 – BootStrap Loader)		017FFh 01000h
4К Порты в/в		00FFFh
		00000h 1

ОЗУ (RAM)

- **4 сектора по 2 Кб**
- **Каждый из секторов может быть полностью отключен битом *RCRSyOFF* регистра *RCCTL0*, данные теряются**
- **Каждый из секторов автоматически переходит в режим пониженного энергопотребления, когда это возможно**
- **USB RAM может использоваться как обычная RAM, если USB не используется**

ОЗУ (RAM)

- Чтение из отключенного сектора всегда дает 0
- Стек располагается в ОЗУ, поэтому нельзя отключать сектор, содержащий стек, если используются прерывания или LPM режим
- Адрес регистра RCSTL0 – 6900h
- Доступ к RCSTL0 защищен ключом. Перед изменением RCSTL0_L (младшего байта, содержащего флаги отключения секторов, номер бита соответствует номеру сектора), в регистр RCSTL0_H необходимо записать код 5Ah. При чтении старший байт содержит 69h

ОЗУ (RAM)

- **Задание.** Доступ к RCCTL0 защищен ключом. Перед изменением RCCTL0_L, в регистр RCCTL0_H необходимо записать код 5Ah. При чтении старший байт содержит 69h
- **Что за странная защита, если известен код? Почему и для чего так?**
- **Почему при чтении читается совершенно другое значение?**

Flash-память

- Основная flash память делится на сегменты по 512 б. Информационная flash память – на 4 сегмента по 128 б.
- Каждый из сегментов может быть стерт за один шаг, также как и все сегменты одновременно. Информационные сегменты стираются индивидуально
- Сегмент А может независимо блокироваться
- Процессор может выполнять пересылки в 1 байт, 1 слово или несколько слов

Flash-память

- По умолчанию flash-память в режиме чтения, в котором нельзя писать в память или стирать == ПЗУ
- Flash-память является *in-system programmable (ISP)* == не требует дополнительного внешнего источника питания
- В очищенном состоянии значение каждого бита flash-памяти равно 1
- Состояние каждого бита может быть изменено с 1 на 0 индивидуально
- Изменение из 0 в 1 требует цикла стирания

Flash-память

- *Наименьшей стираемой единицей данных flash-памяти является сегмент*
- *Имеется три режима стирания, выбираемых битами ERASE и MERAS*

Flash-память

- *Конфигурационные регистры FCTLx – 16-битные защищенные паролем регистры*
- *Считывание или запись состояния регистров осуществляется словами (16 бит)*
- *Запись в регистры должна сопровождаться записью пароля 0xA5 в старший байт (макроопределение FWKEY или FWPW)*
- *Запись в регистр FCTLx значения, отличного от 0xA5 приводит к программному сбросу PUC (Power Up Clear)*

Flash-память

- Чтение из регистра *FCTLx* читает *0x96* в старшем байте (макроопределение *FRPW* или *FRKEY*)
- Перед записью или стиранием необходимо снять бит *LOCK*
- После записи или стирания снять соответствующий бит и установить *LOCK*

Flash-память. Регистр FCTL1

Поле	Биты	Назначение
FRPW/FWPW	8-15	FCTL пароль
BLKWRT	7	Режим записи блока. BLKWRT и WRT используются совместно. 01 = запись байта или слова, 10 = запись длинного слова, 11 = запись блока длинными словами
WRT	6	
SWRT	5	Режим Smart write
MERAS	2	Режим полного стирания. MERAS и ERASE используются совместно. MERAS и ERASE автоматически сбрасываются при установке EMEX или завершении операции стирания. 00 = нет стирания, 01 = стирание сегмента, 10 = стирание банка, 11 = полное стирание (очистка всех банков flash-памяти)
ERASE	1	

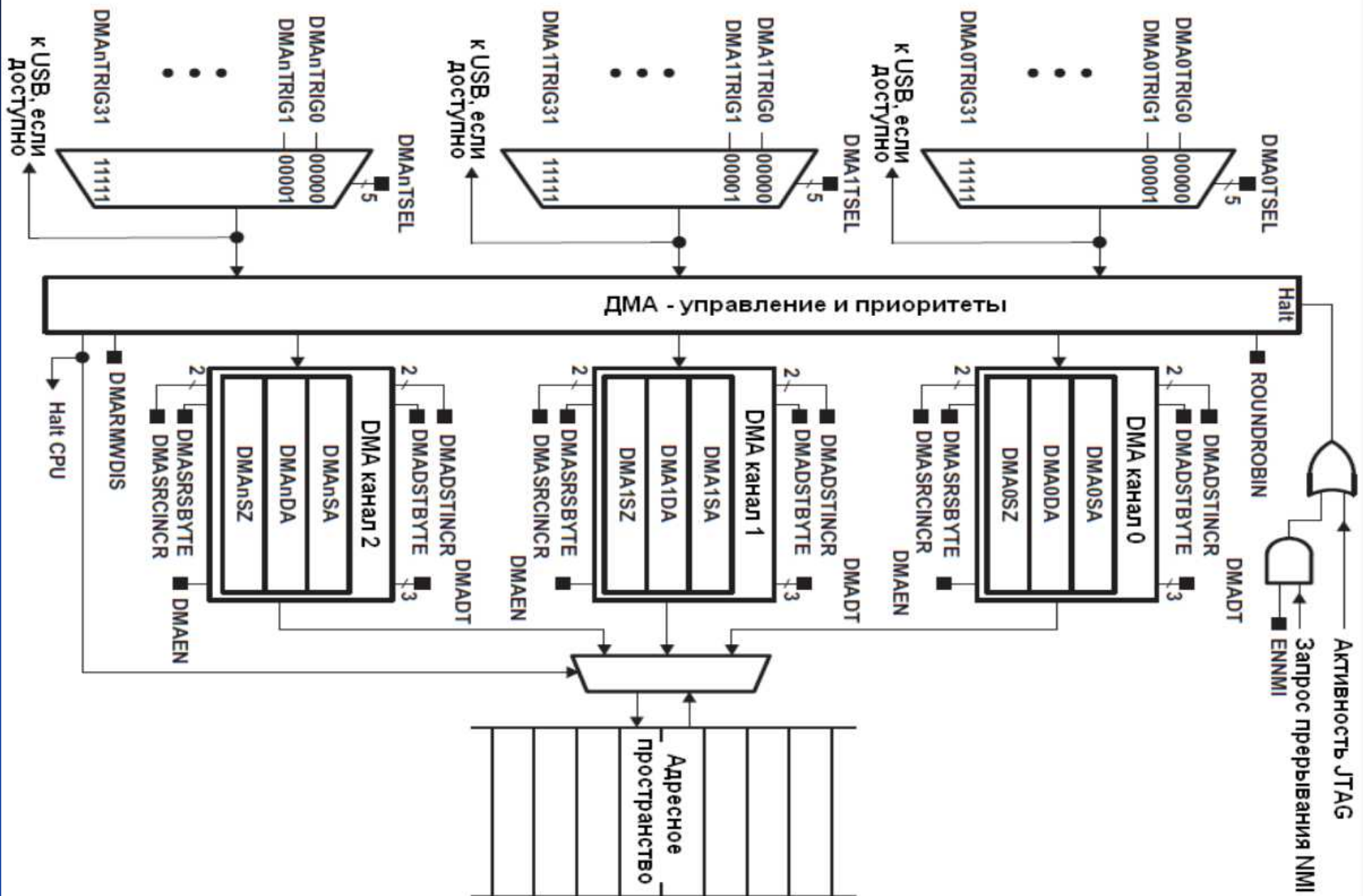
Flash-память. Регистр FCTL3

Поле	Биты	Назначение
FRPW/FWPW	8-15	FCTL пароль
LOCKA	6	Блокировка сегмента A (бит = 1). Запись 1 в этот бит меняет его состояние
EMEX	5	Аварийное завершение. Установка бита прекращает стирание или запись
LOCK	4	Блокировка Flash-памяти от записи и стирания
WAIT	3	Ожидание. Флаг показывает, что flash-память находится в процессе записи
ACCVIFG	2	Флаг прерывания нарушения доступа
KEYV	1	Нарушение пароля. Флаг должен быть сброшен программно
BUSY	0	Занято. Flash-память в процессе стирания или записи

Контроллер DMA

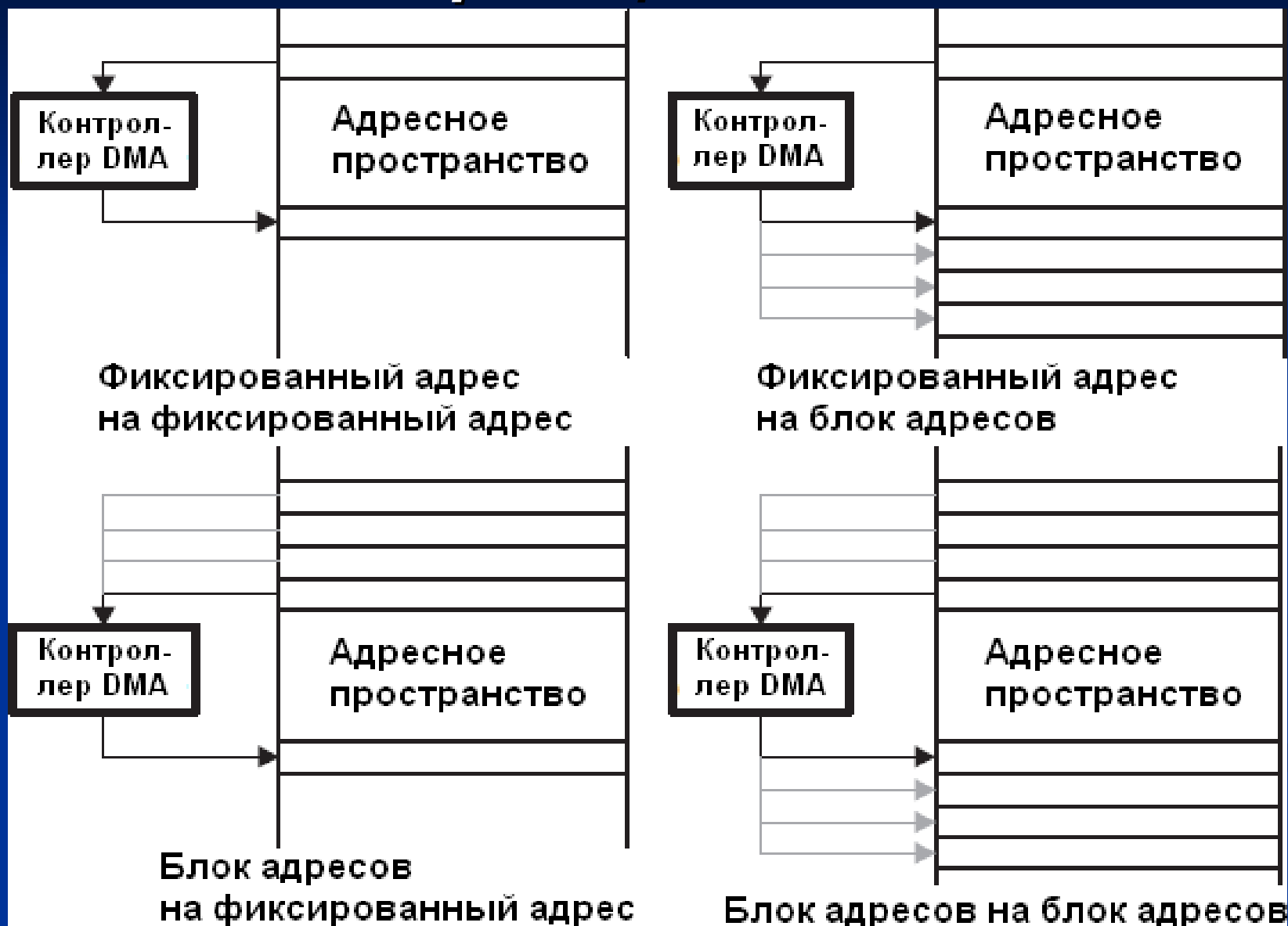
- *три независимых канала*
- *программируемые приоритеты каналов*
- *требуется всего 2 MCLK такта на пересылку*
- *возможность пересылки байт, слов или смешанные*
- *размер блока данных до 65 К байт или слов*
- *программируемый выбор триггеров передачи*
- *пересылки по перепаду сигнала триггера или по уровню*
- *4 режима адресации*
- *3 режима пересылки: одиночные, блочные и многоблочные*

Контроллер DMA



Контроллер DMA

■ Режимы адресации



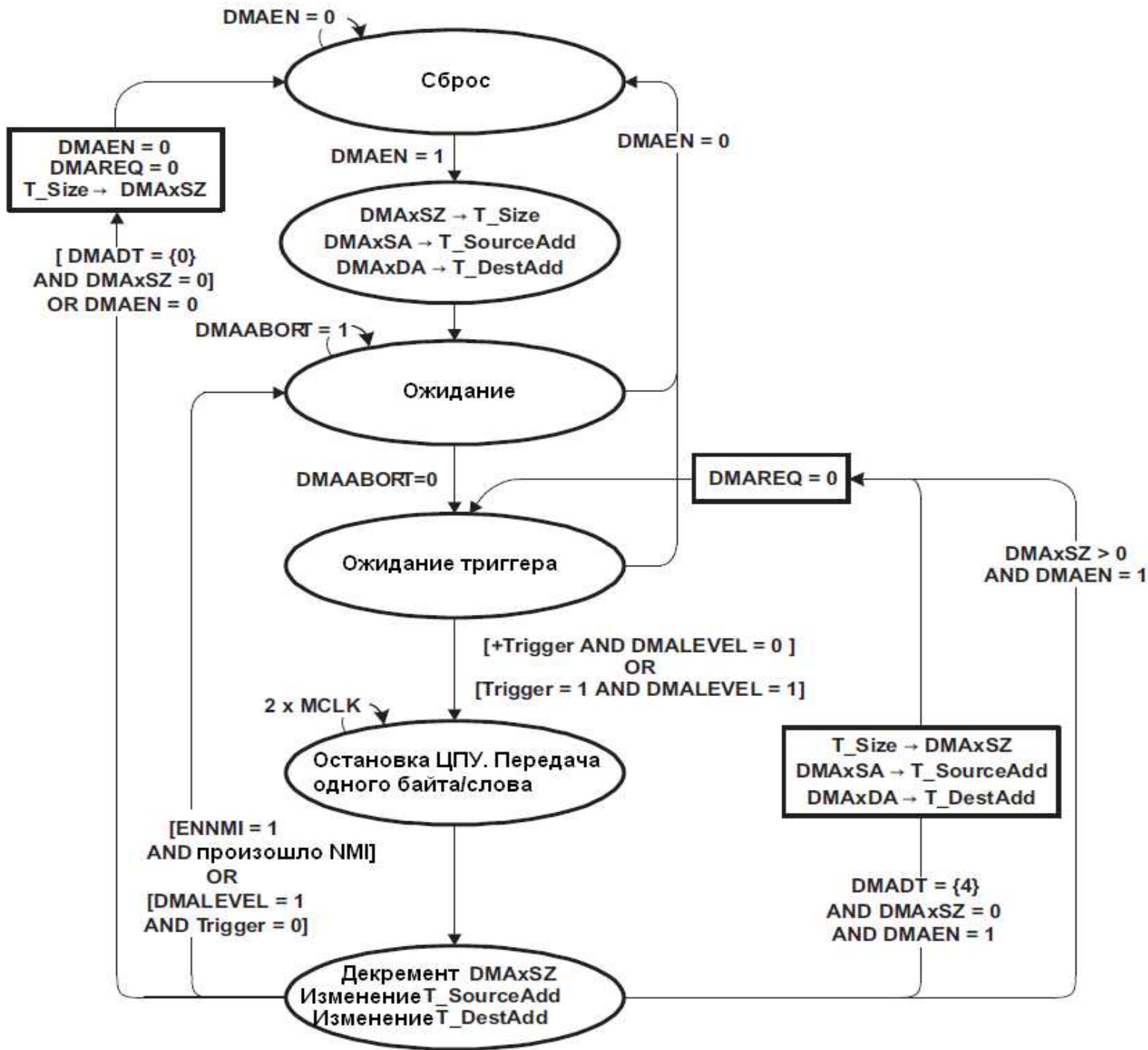
Контроллер DMA

- **Биты *DMASRCINCR* и *DMADSTINCR* выбирают, будут ли адреса источника и приемника, соответственно, инкрементироваться, декрементироваться или оставаться без изменений**
- **Пересылки возможны байт в байт, байт в слово, слово в байт и слово в слово**
- **При пересылках байта в слово старший байт результата обнуляется**
- **При пересылках слова в байт пересылается младший байт источника**

Контроллер DMA

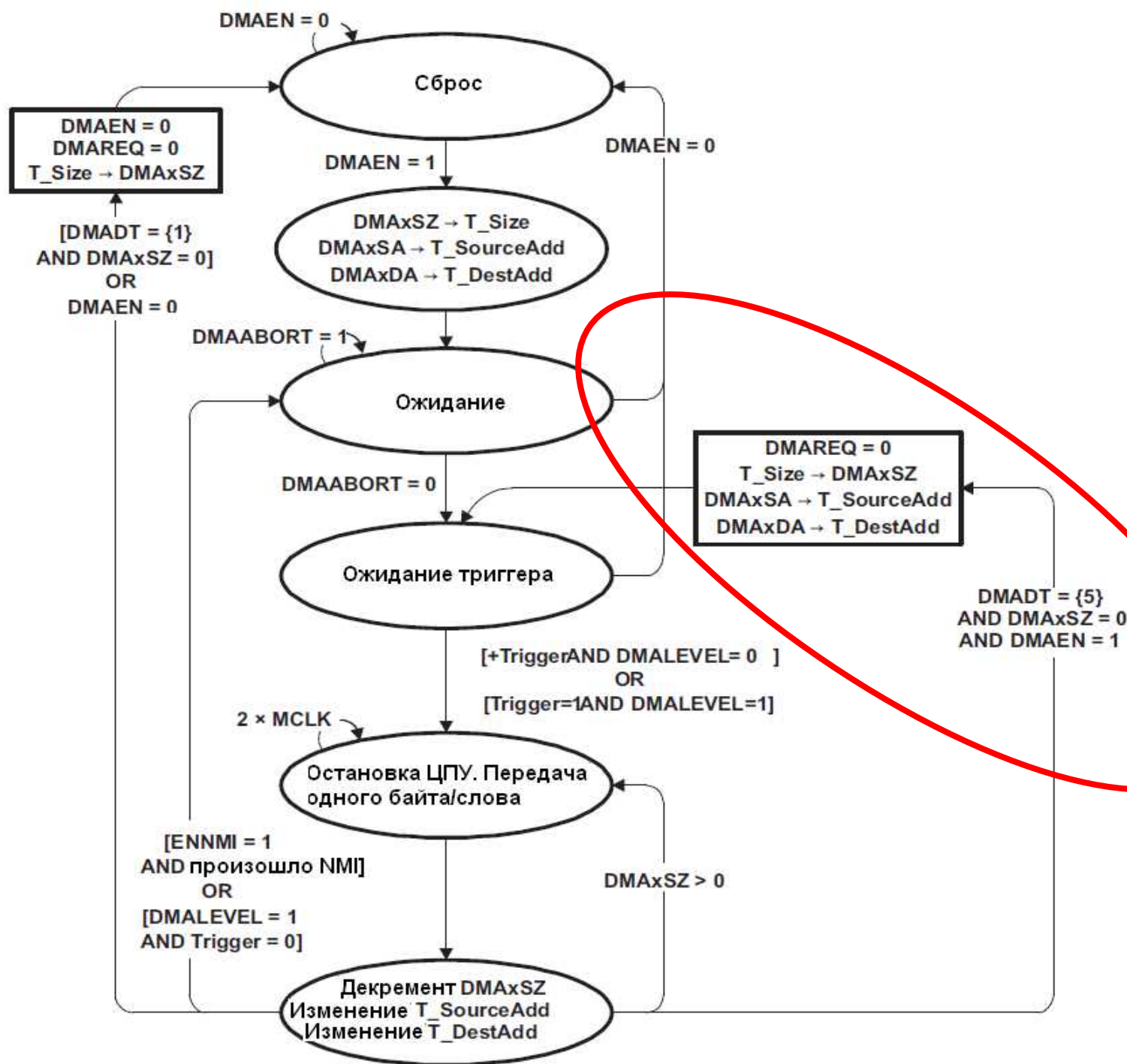
- Биты DMADT задают 6 режимов пересылки, программируемые отдельно для каждого из каналов:
- 000 — одиночная пересылка
- 001 — блочная пересылка
- 010, 011 — импульсная блочная пересылка
- 100 — повторяющаяся одиночная пересылка
- 101 — повторяющаяся блочная пересылка
- 110, 111 — повторяющаяся импульсная блочная пересылка

Контроллер DMA. Одиночные пересылки



Контроллер DMA. Одиночные пересылки

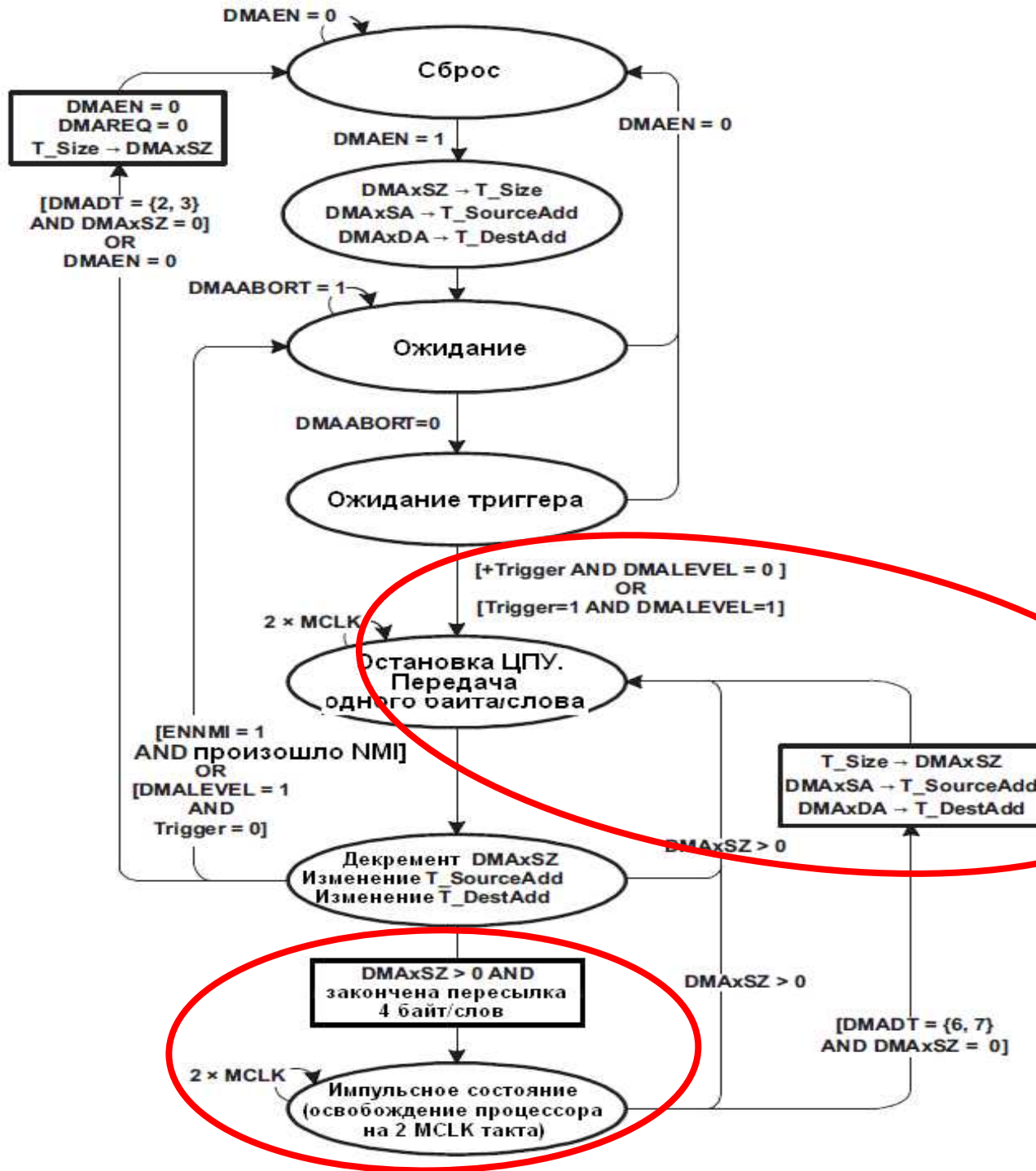
- **Перед пересылкой текущие значения адреса источника, адреса назначения и количества пересылок копируются во временные регистры, которые изменяются после каждой пересылки**
- **Каждая пересылка требует срабатывания триггера**
- **DMAEN автоматически сбрасывается, когда сделано DMAxSZ пересылок**
- **Когда DMAxSZ становится равен 0, он повторно копируется, и устанавливается соответствующий флаг DMAIFG**
- **В режиме повторяющихся одиночных пересылок, DMAEN остается активным, и при каждом новом срабатывании триггера происходит пересылка**



Контроллер DMA. Блочные
пересылки

Контроллер DMA. Блочные пересылки

- ***По срабатыванию одного триггера пересылается блок данных***
- ***В процессе пересылки игнорируются любые другие сработавшие триггеры***
- ***DMAxSZ определяет размер блока***
- ***На протяжении всего обмена процессор остановлен***
- ***DMAEN автоматически сбрасывается после передачи блока***
- ***В случае повторяющихся блочных пересылок DMAEN остается активным, и по окончании пересылки новое срабатывание триггера вызывает новую пересылку блока***
- ***Пересылка занимает $2 \times MCLK \times DMAxSZ$ тактов***



Контроллер DMA. Импульсно-блочные пересылки

Контроллер DMA. Импульсно-блочные пересылки

- **После каждых 4 пересылок байт либо слов, на 2 такта MCLK включается процессор (==20%)**
- **В остальном режим подобен блочному**
- **Существенное отличие в повторяющемся импульсно-блочном режиме:**
- **так как DMAEN остается активным, то после пересылки блока новый блок начинает пересылку следующего, и для этого не требуется срабатывание нового триггера**
- **Такая пересылка может быть остановлена сбросом DMAEN бита либо прерыванием NMI при установленном ENNMI**

Контроллер DMA

- При $DMALEVEL = 1$ триггер определен по уровню сигнала, а не по фронту
- Для правильной работы этого режима источником сигнала триггера должен быть выбран внешний источник $DMAE0$
- Пересылка активна все время, пока сигнал триггера остается высоким и $DMAEN = 1$
- В этом режиме сигнал триггера должен оставаться высоким на все время пересылки
- Если он станет низким для блочной или импульсно-блочной пересылки, контроллер DMA будет остановлен в текущем состоянии и продолжит работу при возврате сигнала в высокий уровень
- В это время простоя могут быть изменены регистры DMA
- Рекомендуется использовать для режима пересылок, когда $DMAEN$ автоматически сбрасывается ($DMA\overline{BT} = 0 \dots 3$).

Контроллер DMA

- Если триггеры разных каналов срабатывают одновременно, на выполнение ставится пересылка канала с максимальным приоритетом
- По умолчанию приоритет каналов DMA0 – DMA1 – DMA2
- Бит ROUNDROBIN устанавливает циклический приоритет: после завершения пересылки каналу назначается минимальный приоритет
- Если срабатывает триггер канала с большим приоритетом, текущая пересылка не прерывается

Контроллер DMA

- **Контроллер DMA требует 1-2 тактов MCLK для синхронизации перед каждым обменом + 2 такта MCLK на пересылку байта либо слова + 1 такт ожидания после пересылки**
- **Пересылка занимает 4-5 тактов**
- **В случае, если источник MCLK выключен, контроллер DMA временно включает MCLK, генерируемую DCOCLK, для выполнения пересылки**
- **В этом случае дополнительно требуется еще 5 мкс для запуска DCOCLK**

Контроллер DMA

- *Пересылки DMA не прерываются системными прерываниями, прерывания ожидают завершения пересылки*
- *Только прерывание NMI может прервать пересылку, если установлен бит ENNMI*
- *Выполнение обработчиков прерываний приостанавливается для DMA пересылки*
- *Чтобы этого не происходило, на время выполнения обработчика прерываний следует отключать DMA контроллер*
- *Каждый канал DMA имеет собственный флаг DMAIFG. Флаг устанавливается, когда соответствующий DMAxSZ становится равным нулю*
- *Если при этом установлены флаги DMAIE и GIE, возникает запрос на прерывание*

Контроллер DMA. Регистры

Регистр	Адрес	Назначение
DMACTL0	0500h	Регистр управления 0
DMACTL1	0502h	Регистр управления 1
DMACTL4	0508h	Регистр управления 4
DMAIV	050Eh	Вектор прерываний
DMA0CTL	0510h	Регистр управления каналом 0 DMA
DMA0SA	0512h	Адрес источника канала 0 DMA
DMA0DA	0516h	Адрес назначения канала 0 DMA
DMA0SZ	051Ah	Размер пересылки канала 0 DMA
DMA1CTL - DMA1SZ	0520h - 052Ah	Регистры канала 1 DMA
DMA2CTL- DMA2SZ	0530h - 053Ah	Регистры канала 2 DMA

Контроллер DMA. Поля регистров

Регистр	Биты	Поле	Назначение
DMACTL0	8-12	DMA1TSEL	Выбор источника триггера канала 1 DMA
	0-4	DMA0TSEL	Выбор источника триггера канала 0 DMA
DMACTL1	0-4	DMA2TSEL	Выбор источника триггера канала 2 DMA
DMACTL4	2	DMARMWDIS	Запрет прерывания цикла операции процессора на шине (чтение/ изменение/ запись)
	1	ROUNDROBIN	Циклический приоритет каналов
	0	ENNMI	Разрешение прерывания DMA пересылки посредством NMI

Контроллер DMA. Поля регистров

Регистр	Биты	Поле	Назначение
DMAxCTL	12-14	DMADT	Режим пересылки
	10-11	DMADSTINCR	Инкремент адреса назначения после пересылки (при пересылке слов адрес изменяется на 2): 00,01 — без изменений, 10 — декремент, 11 - инкремент
	8-9	DMASRCINCR	Инкремент адреса источника после пересылки
	7	DMADSTBYTE	Размер данных приемника: 0 - слово, 1 - байт
	6	DMASRCBYTE	Размер данных источника

Контроллер DMA. Поля регистров

Регистр	Биты	Поле	Назначение
DMAxCTL	5	DMALEVEL	Режим срабатывания триггера: 0 — по переднему фронту, 1 — по высокому уровню
	4	DMAEN	Разрешение DMA (=1)
	3	DMAIFG	Флаг прерывания DMA
	2	DMAIE	Разрешение прерывания DMA
	1	DMAABORT	Флаг, устанавливается, если NMI прервало пересылку DMA
	0	DMAREQ	Запрос DMA. Программно-управляемый запуск пересылки. Сбрасывается автоматически

Контроллер DMA. Поля регистров

Регистр	Биты	Поле	Назначение
DMAxSA	0-19	DMAxSA	Адрес источника. Обращение к регистру требует расширенных операций. Использование операций для слов очищает регистр
DMAxDA	0-19	DMAxDA	Адрес назначения. Аналогично
DMAxSZ	0-15	DMAxSZ	Количество передаваемых данных (байт или слов)
DMAIV	0-15	DMAIV	Вектор прерываний

Контроллер DMA. Источники триггеров

- ***Для каждого канала выбор сигнала источника триггера выполняется битами DMAxTSEL***
- ***При этом обязательно DMAEN должен быть = 0***
- ***Перечень источников одинаков для каждого из каналов***
- ***0 — DMAREQ. Запрос DMA (программный запуск). Триггер срабатывает при установке бита. Сигнал DMAREQ автоматически сбрасывается после начала пересылки***

Контроллер DMA. Источники триггеров

- ***1 — TA0CCR0 CCIFG; 2 — TA0CCR2 CCIFG;***
- ***3 — TA1CCR0 CCIFG; 4 — TA1CCR2 CCIFG;***
- ***5 — TA2CCR0 CCIFG; 6 — TA2CCR2 CCIFG;***
- ***7 — TB0CCR0 CCIFG; 8 — TB0CCR2 CCIFG***
- ***Запуск по каналам таймеров. Триггер срабатывает при установке бита. Соответствующий сигнал CCIFG автоматически сбрасывается после начала пересылки. Если установлен соответствующий бит CCIE, выбранный флаг CCIFG не запускает пересылку DMA***

Контроллер DMA. Источники триггеров

- ***16 — UCA0RXIFG; 17 — UCA0TXIFG;***
- ***18 — UCB0RXIFG; 19 — UCB0TXIFG;***
- ***20 — UCA1RXIFG; 21 — UCA1TXIFG;***
- ***22 — UCB1RXIFG; 23 — UCB1TXIFG***
- ***Запуск по каналам USCI. Триггер срабатывает при приеме (RX)/готовности к передаче (TX) данных по соответствующему каналу USCI. Сигнал RXIFG/TXIFG автоматически сбрасывается после начала пересылки. Если установлен соответствующий бит RXIE/TXIE, выбранный флаг RXIFG/TXIFG не запускает пересылку DMA***

Контроллер DMA. Источники триггеров

- ***24 — ADC12IFGx. Запуск по АЦП. Триггер срабатывает при установке бита (завершении одноканального преобразования АЦП или завершении последнего преобразования в последовательности). Программная установка бита не запускает триггер. Все ADC12IFG флаги автоматически сбрасываются, когда к соответствующему ADC12MEMx обратился контроллер DMA***
- ***27 — USB FNRXD. USB триггер***
- ***28 — USB ready. USB триггер***
- ***29 — MPY ready. Запуск по умножителю. Триггер срабатывает, когда умножитель готов для нового операнда***

Контроллер DMA. Источники триггеров

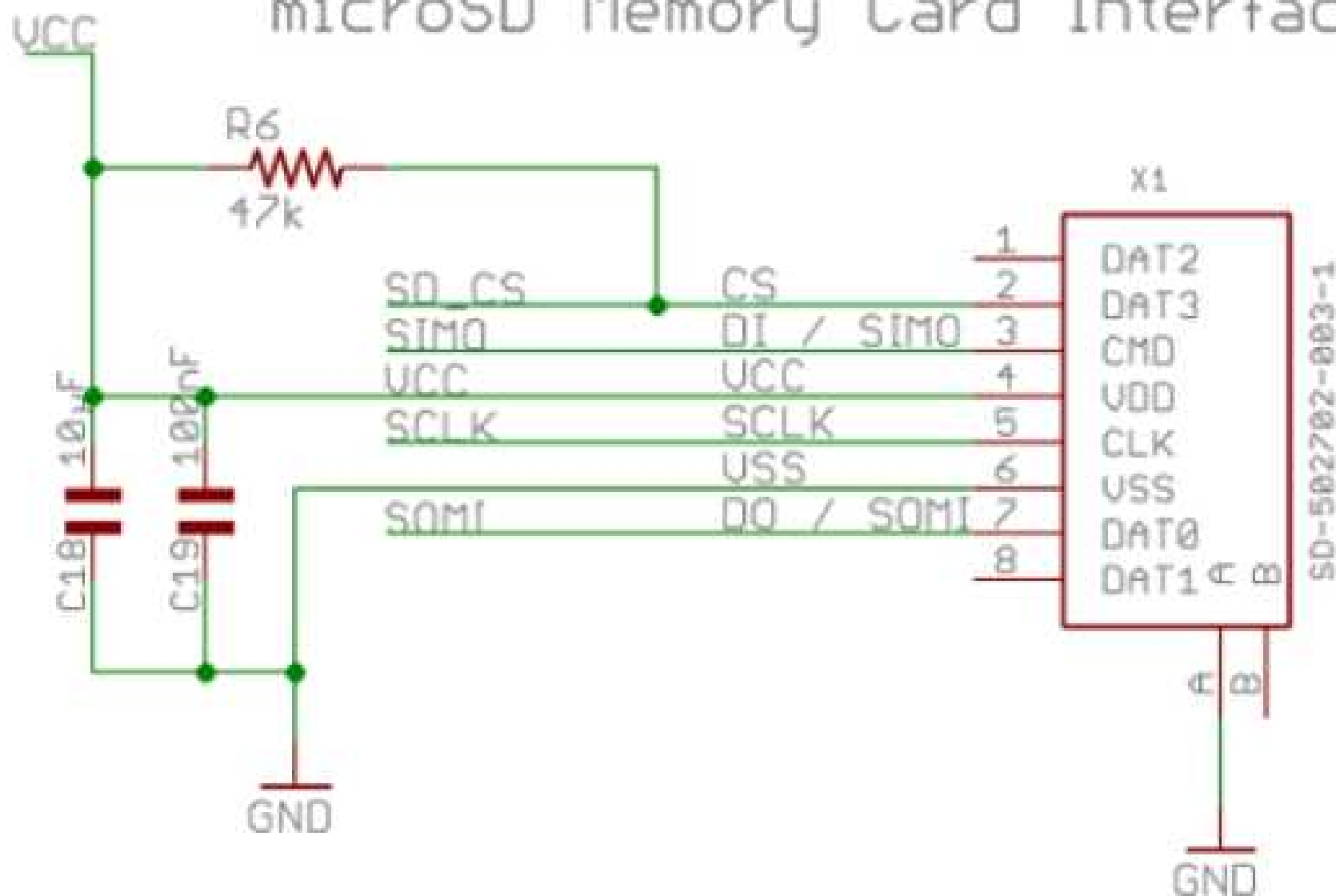
- ***30 — DMAxIFG. DMA2IFG – для канала 0, DMA0IFG – для канала 1, DMA1IFG – для канала 2. Триггер срабатывает при установке бита. Сигнал DMAxIFG не сбрасывается автоматически***
- ***31 — DMAE0. Пересылка по внешнему сигналу триггера***

SD- карта

- ***Multi Media Card — MMC***
- ***Secure Digital Memory Card, Secure Digital — SD***
- ***Обмен данными по двум протоколам: MMC и SPI***
- ***Протокол MMC обеспечивает большую скорость и возможность параллельного включения нескольких карт и является основным***
- ***SPI поддерживается микроконтроллером на аппаратном уровне и для многих платформ его удобнее использовать***

SD- карта

microSD Memory Card Interface



SD- карта

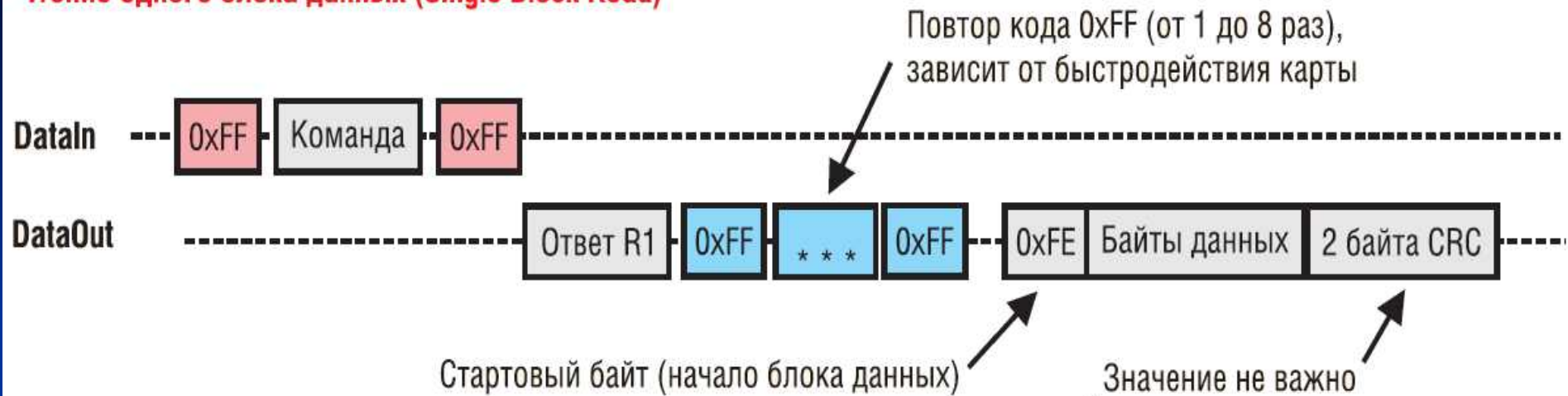
- ***В MSP-EXP430F5529 SD-карта подключена по SPI***
- ***На этом же канале подключен ЖКИ***
- ***Не должно быть двух активных устройств на SPI одновременно***
- ***SD_CS (DAT) P3.7/ TB0OUTH — разрешение устройства***
- ***SIMO (CMD) P4.1/ PM_UCB1SIMO/ PM_UCB1SDA — SIMO данные (запись в память)***
- ***SCLK (CLK) P4.3/ PM_UCB1CLK/ PM_UCA1STE — синхросигнал***
- ***SOMI (DAT0) P4.2/ PM_UCB1SOMI/ PM_UCB1SCL — SOMI данные (чтение из памяти)***

SD- карта

- ***Это всего лишь подключение SPI интерфейса***
- ***Этого недостаточно для работы с MMC/SD картой***
- ***Необходимо поддерживать протокол обмена, установленный для MMC/SD***

SD-карта

Чтение одного блока данных (Single Block Read)



Запись одного блока данных (Single Block Write)



0xFF – Байты, которые необходимо передавать для формирования требуемого количества дополнительных тактовых импульсов

SD- карта

- **MMC/SD карта принимает от микроконтроллера ряд команд, на которые она выдаёт либо ответы определённого типа, либо блоки данных. Ответ – R1, R2 или R3 – может состоять из 1, 2 или 5 байтов**
- **Собственно данные могут иметь длину 1...N байт, где N определяется размером физического сектора, в большинстве случаев – 512 байт**
- **Контрольная сумма является в SPI опциональной**
- **Значения CRC можно игнорировать, но сами эти байты должны обязательно передаваться/приниматься для соблюдения протокола обмена**

SD- карта

- **Перед передачей команды или после этого микроконтроллер должен выдавать не менее 8 тактовых импульсов по линии CLK, т.е. просто передавать «лишний» байт 0xFF**
- **При чтении блока данных после передачи соответствующей команды микроконтроллер принимает байты 0xFF до тех пор, пока не встретится байт 0xFE (стартовый байт блока данных)**
- **Любой иной байт (отличный от 0xFF), полученный в этот момент, будет означать ошибку**

SD- карта

- ***Все команды, воспринимаемые MMC/SD картой, имеют длину 6 байт***
- ***Первый байт команды:***
 - ***биты 7 и 6 = 01***
 - ***Биты 0..5 — это номер (код) команды***
 - ***Следующие 4 байта содержат аргумент команды, например, 32-битный адрес первого байта данных***
- ***Последний байт команды:***
 - ***биты 1..7 — CRC***
 - ***бит 0 = 1***

SD-карта

- Ответ содержит 1, 2 или 5 байт
- Первым передаётся старший байт
- Ответ формата R1 содержит один байт:
- бит 7 – всегда =0
- бит 6 — ошибка параметра команды
- бит 5 — ошибка адреса
- бит 4 — ошибка стирания
- бит 3 — ошибка контрольной суммы CRC
- бит 2 — неверная команда
- бит 1 — прервана команда стирания
- бит 0 — режим простоя, выполняется инициализация

SD-карта

- Ответ R2 содержит 2 байта
- Первый байт = R1
- 2 байт:
- бит 7 — выход за пределы / ошибка перезаписи
- бит 6 — ошибка параметра при стирании
- бит 5 — попытка записи в защищенную от записи область
- бит 4 — ошибка коррекции
- бит 3 — внутренняя ошибка
- бит 2 — общая / неизвестная ошибка
- бит 1 — попытка стирания защищенного от записи сектора / ошибка блокирования/разблокирования
- бит 0 — карта заблокирована

SD-карта

- Ответ R3 содержит 5 байт
- Первый байт = R1
- Остальные 4 байта - содержимое регистра OCR
- Если операция чтения данных завершилась неудачно и карта не может предоставить данные, она будет посылать байт ошибки данных
- При записи данных в MMC/SD карту после получения блока данных карта отвечает байтом подтверждения данных:
- Бит 4 = 0
- Биты 1..3 указывают статус операции, успешной записи соответствует значение 010
- Бит 0 = 1

SD-карта

- После подачи напряжения питания MMC/SD карта находится в режиме MMC, а не SPI
- Для перевода карты в режим SPI и инициализации:
- не выбирая устройство (сигнал CS = 1) послать 80 импульсов по линии CLK (передать 10 байт 0xFF)
- выбрать MMC/SDC карту (CS = 0)
- послать команду CMD0 (сброс): 0x40, 0, 0, 0, 0, 0x95 (контрольная сумма должна иметь реальное значение (0x95), т.к. команда посылается в режиме MMC)
- дождаться правильного ответа 0x01

SD-карта

- ***далее в цикле посылать команду CMD1 (инициализация) и ждать, когда будет получен ответ 0x00 (этот ответ означает, что карта инициализирована в режиме SPI и готова принимать команды)***
- ***Для SD-карт в случае отклонения команды CMD1 рекомендуется использовать команду ACMD41***

SD-карта

Команда	Код	Аргумент	От ве т	Да н н ые	Описание
CMD0	40h	Нет (0)	R1	-	GO_IDLE_STATE. Программный сброс
CMD1	41h	Нет (0)	R1	-	SEND_OP_COND. Запуск процесса инициализации
ACMD41	69h	*	R1	-	APP_SEND_OP_COND. Только для карт SD. Запуск процесса инициализации
ACMD означает последовательность двух команд CMD55 + CMD					
* Бит 30 - HCS, остальные в 0					
CMD8	48h	**	R7	-	SEND_IF_COND. Только для карт SDC v2. Проверка диапазона напряжения питания
** Биты 31..12 = 0, биты 11..8 — напряжение питания, биты 7..0 — 0xAA					
CMD9	49h	Нет (0)	R1	+	SEND_CSD. Чтение регистра CSD
CMD10	4Ah	Нет (0)	R1	+	SEND_CID. Чтение регистра CID

SD-карта

Команда	Код	Аргумент	Ответ	Данные	Описание
CMD12	4Ch	Нет (0)	R1b	-	STOP_TRANSMISSION. Остановка чтения данных
CMD16	50h	Длина блока [31..0]	R1	-	SET_BLOCKLEN. Установка размера блока чтения записи
CMD17	51h	Адрес [31:0]	R1	+	READ_SINGLE_BLOCK. Чтение блока
CMD18	52h	Адрес [31:0]	R1	+	READ_MULTIPLE_BLOCK. Чтение нескольких блоков
CMD23	57h	Число блоков [15:0]	R1	-	SET_BLOCK_COUNT. Только для MMC. Количество блоков для передачи со следующей командой многоблочного чтения/записи

SD-карта

Команда	Код	Аргумент	От ве т	Да нн ые	Описание
ACMD23	57h	Число блоков [22:0]	R1	-	SET_WR_BLOCK_ERASE_COUNT. Только для SD. Количество блоков для предвари-тельного стирания для последующей коман-ды многоблочной записи
CMD24	58h	Адрес [31:0]	R1	+	WRITE_BLOCK. Запись блока
CMD25	59h	Адрес [31:0]	R1	+	WRITE_MULTIPLE_BLOCK. Запись нескольких блоков
CMD55	77h	Нет (0)	R1	-	APP_CMD. Начало команды ACMD
CMD58	7Ah	Нет(0)	R1	-	READ_OCR. Чтение OCR

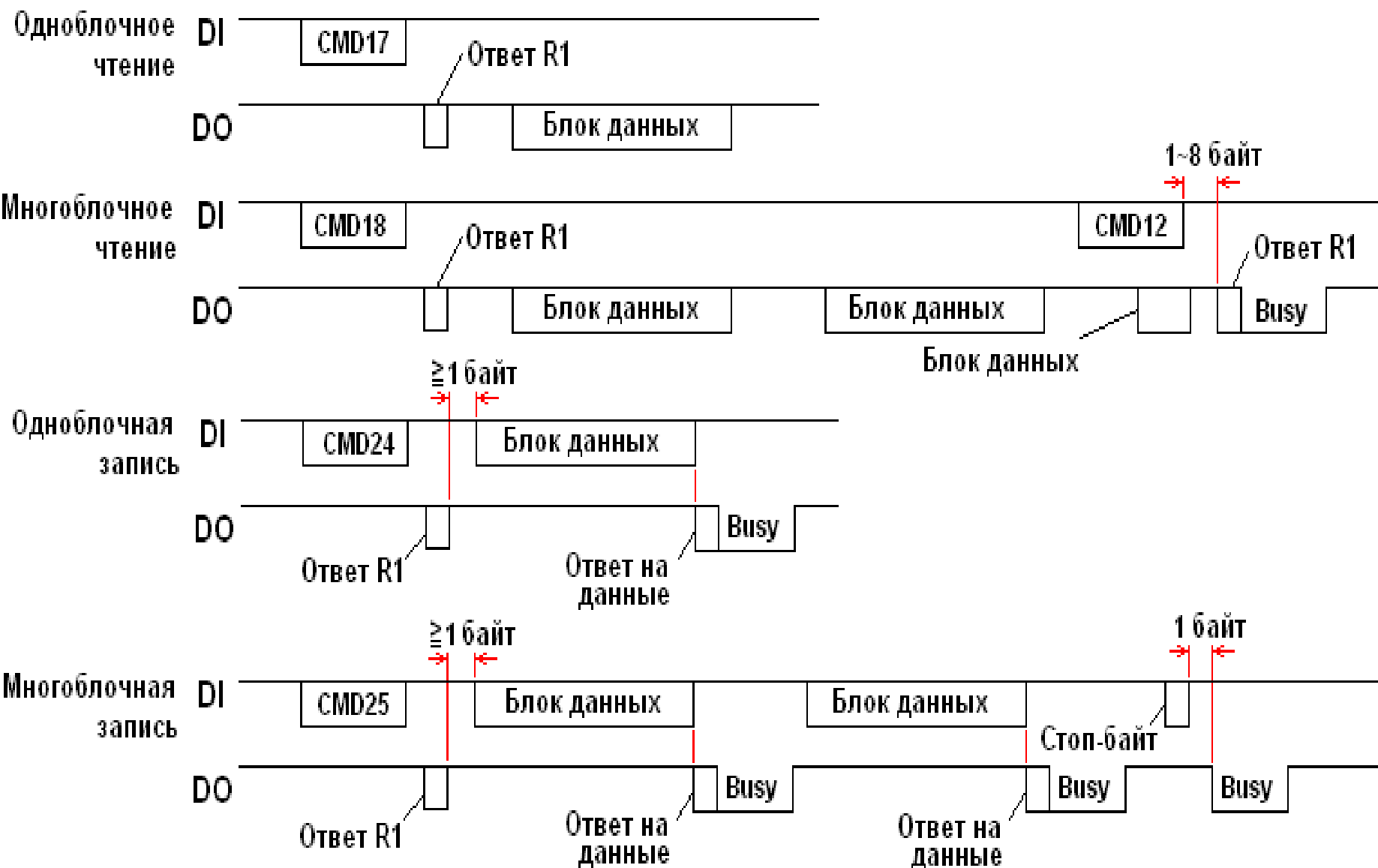
SD-карта

- ***После простоя более 5 мс карта памяти переходит в энергосберегающий режим, и способна принимать только команды CMD0, CMD1 и CMD58***
- ***Поэтому процесс инициализации (CMD1) необходимо практически каждый раз повторять при чтении/записи блока данных или делать проверку состояния карты***

SD-карта

- По SPI протоколу доступны регистры контроллера карты:
- CID (Card identification data): данные, по которым можно идентифицировать карту памяти (серийный номер, ID производителя, дату изготовления и т. д.)
- CSD (Card-specific data): информация о карте памяти (от размера сектора карты памяти до потребления в режиме чтения/записи)
- OCR (Operation Conditions Register): напряжения питания карты памяти, тип питания, статус процесса инициализации

SD-карта



SD-карта

- **Одноблочное чтение CMD17:**
- **Аргумент задает адрес начала чтения**
- **Чтение осуществляется побайтно**
- **В ответ на команду карта выдает контроллеру пакет данных**
- **После обнаружения маркера данных контроллер принимает следующий за ним блок данных и два байта CRC**
- **По умолчанию размер блока 512 байтов, его можно изменить CMD16**
- **Если во время чтения произошла ошибка, вместо пакета данных будет возвращен маркер ошибки**

SD-карта

- **Многоблочное чтение CMD18:**
- **Если перед командой с помощью CMD23 (только для MMC) не было задано число передаваемых блоков, будет инициировано неограниченное многоблочное чтение**
- **контроллер может прервать его CMD12**
- **Байт, получаемый сразу же после передачи CMD12, является наполняющим, его не нужно учитывать**
- **После этого байта следует ответ на команду**

SD-карта

- **Запись CMD24:**
- **контроллер после байтового промежутка (один или более байтов) передает пакет данных**
- **Формат пакета такой же, как и у команды чтения**
- **После передачи пакета карта сразу же выдает ответ на данные, за которым следует флаг занятости**
- **Большинство карт не могут менять размер записываемого блока, он является фиксированным и составляет 512 байтов**

SD-карта

- **Запись CMD24:**
- **По правилам режима SPI сигнал CS должен находится в активном уровне в течение всей транзакции, но есть исключение**
- **Когда карта занята, контроллер может снять сигнал CS, чтобы освободить шину SPI для другого устройства**
- **Если снова выбрать карту в то время, когда она занята выполнением внутреннего процесса, карта снова установит сигнал DO в низкий уровень**
- **Для сокращения времени ожидания лучше выполнять проверку на занятость непосредственно перед выдачей команды и пакета данных**

SD-карта

- **Запись CMD24:**
- **Кроме того, внутренний процесс иницируется спустя байт после ответа данных, т.е. необходимо выдать 8 тактовых импульсов, чтобы иницировать внутреннюю операцию записи**
- **Состояние сигнала CS во время этих восьми тактовых импульсов не учитывается, поэтому можно совместить эту инициацию с процессом освобождения шины**

SD-карта

- **Многоблочная запись CMD25:**
- **Если перед командой число передаваемых блоков не было задано CMD23 (для MMC) или ACMD23 (для SD), транзакция будет инициирована как неограниченная**
- **Запись будет продолжаться, пока контроллер не прервет ее передачей маркера остановки передачи (стоп-байт, Fdh)**
- **Флаг занятости появится байт спустя после стоп-байта**
- **Что же касается SD, то транзакция многоблочной записи должна прерываться стоп-байтом независимо от того, является ли она predetermined или неограниченной**

SD-карта

- **Этого протокола уже достаточно, чтобы обмениваться неформатированными данными**
- **Физически память MMC/SD карты разбита на сектора по 512 байт, карта имеет, как правило, файловую систему FAT16**
- **Поэтому для полноценной поддержки обмена файлами, которые потом будут видимы при использовании SD-карты на других устройствах, необходимо еще и поддержать файловую систему FAT**
- **Структуру MBR, FAT и формат каталога в этой файловой системе рассматривать не будем (но знать на экзамене надо)**⁶²