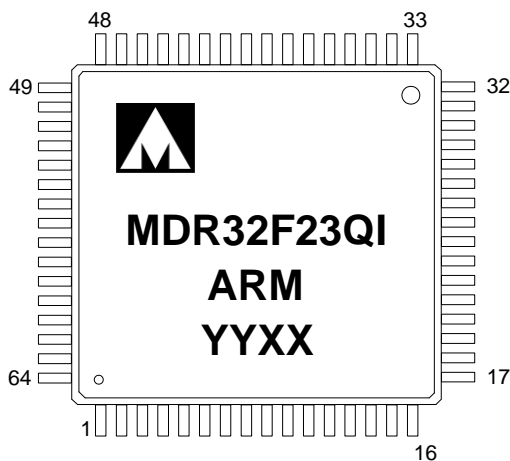




Микросхема 32-разрядного микроконтроллера  
с 12-разрядным АЦП  
К1986ВК234, К1986ВК234К



XX – год выпуска  
YY – неделя выпуска

Основные параметры  
микросхемы

- Напряжение источника питания 2,2 – 3,6 В;
- 32 разрядная RISC архитектура ARM Cortex-M0
- Встроенная память программ 64 Кбайт;
- Встроенная память данных 16 Кбайт;
  
- Температурный диапазон:

Обозначение	Диапазон
К1986ВК234	минус 40 – 85°С
К1986ВК234К	0 – 70°С

Тип корпуса:

– 64-х выводной пластиковый корпус LQFP10X10-64-OP-01.

## **Основные характеристики**

### *Ядро:*

- ARM 32-битное RISC ядро Cortex™-M0, тактовая частота до 36 МГц;
- Умножение за один цикл.

### *Память:*

- Встроенная энергонезависимая память программ FLASH типа размером 64 Кбайт;
- Встроенное ОЗУ размером 16 Кбайт.

### *Питание и тактовая частота:*

- Внешнее питание 2,2...3,6 В;
- Встроенный регулятор напряжения на 1,8 В для питания ядра;
- Встроенные схемы контроля питания;
- Встроенный домен с батарейным питанием;
- Встроенный подстраиваемый RC генератор HSI, 8 МГц;
- Встроенный подстраиваемый RC генератор LSI, 40 кГц;
- Внешний осциллятор HSE, 2...16 МГц;
- Внешний осциллятор LSE, 32 кГц;
- Встроенный умножитель тактовой частоты PLL для ядра.

### *Режим пониженного энергопотребления:*

- Режим SLEEP, DEEPSLEEP и Standby;
- Батарейный домен с часами реального времени и регистрами аварийного сохранения.

### *Аналоговые модули:*

- 24-х разрядный  $\Sigma\Delta$  АЦП (до 3 каналов);
- 12-ти разрядный АЦП (до 8 каналов) измеряемый диапазон от 0 до 3,6 В;
- Температурный датчик.

### *Периферия:*

- Контроллеры интерфейсов UART, SPI, I2C;
- Контроллер LCD дисплея;
- До 47 пользовательских линий ввода-вывода;
- Два блока 16-ти разрядных таймеров;
- Два сторожевых таймера.

### *Режим отладки:*

- Последовательный отладочный интерфейс SWD.

### *Корпуса:*

- 64-х выводной пластиковый LQFP10X10-64-OP-01.

### *Температурные диапазоны:*

- -40...+85°C;
- 0 – 70°C.

## **Общее описание и области применения микросхемы**

Микросхемы интегральные K1986BK234 (далее – микросхемы) предназначены для использования в приборах учета электроэнергии в однофазных (~220 В/50 Гц) сетях.

Микроконтроллер K1986BK234 является микроконтроллером со встроенной FLASH памятью программ и построен на базе низкопотребляемого процессорного RISC ядра ARM Cortex-M0. Микроконтроллер работает на тактовой частоте до 36 МГц и содержит 64 Кбайт FLASH памяти и 16 Кбайт ОЗУ. Микроконтроллер включает в себя развитую периферию для построения счетчиков электроэнергии 1-фазных сетей. Периферия включает в себя 3 канала для 1-фазной сети 24-х битных независимых  $\Sigma\Delta$  АЦП. Каждый канал  $\Sigma\Delta$  АЦП имеет предусилитель, фазовую подстройку (для коррекции фазы не хуже 0.1), а так же аппаратный блок для вычисления среднеквадратического значения сигнала. Каждый канал  $\Sigma\Delta$  АЦП может быть включен или отключен независимо от других каналов, и имеет отдельный канал прямого доступа в память. Еще один дополнительный 12-битный АЦП последовательного приближения может быть использован для мониторинга напряжения питания основного или батарейного доменов, а так же для измерения температуры или захвата внешнего сигнала. В состав микроконтроллера входят два UART, I<sup>2</sup>C и SPI интерфейсы. Микроконтроллер содержит два 16-ти разрядных таймера. Так же микроконтроллер содержит системный 24-х разрядный таймер и два сторожевых таймера.

Встроенные RC генераторы HSI (8 МГц) и LSI (40 кГц) и внешние генераторы HSE (2...16 МГц) и LSE (32 кГц) и схема умножения тактовой частоты PLL для ядра позволяют гибко настраивать скорость работы микроконтроллеров.

Архитектура системой шины за счет регулировки частоты периферийных блоков позволяет уменьшить потребление всей системы.

Встроенный регулятор для формирования питания внутренней цифровой части формирует напряжение 1,8 В и не требует дополнительных внешних элементов. Таким образом, для работы микроконтроллера достаточно одного внешнего напряжения питания в диапазоне от 2,2 до 3,6 В. Так же в микроконтроллере реализован батарейный домен, работающий от внешней батареи при отсутствии основного питания. В батарейном домене могут быть сохранены специальные флаги, а так же работают часы реального времени. Встроенные детекторы напряжения питания могут отслеживать уровень внешнего основного питания и уровень напряжения питания на батарее. Аппаратные схемы сброса по просадке питания позволяют исключить сбойную работу микросхемы при выходе уровня напряжения питания за допустимые пределы.

**Содержание**

Тип корпуса: .....	1
Основные характеристики .....	2
Общее описание и области применения микросхемы.....	3
Описание выводов .....	8
Диаграмма расположения выводов в корпусах.....	10
Структурная блок-схема .....	11
Описание функционирования микросхемы .....	12
Система питания.....	12
Структурная схема подачи питания.....	13
Схема сброса при включении и выключении основного питания.....	15
Организация памяти.....	16
Структурная схема .....	16
Базовые адреса процессора .....	17
Загрузочное ПЗУ и режимы работы микроконтроллера .....	19
UART загрузчик.....	20
Контроллер FLASH памяти программ .....	26
Работа Flash памяти программ в обычном режиме.....	26
Работа Flash памяти программ в режиме программирования.....	26
Описание регистров управления контроллера Flash памяти программ.....	30
Система команд .....	34
Встроенные функции.....	36
Описание инструкций.....	38
Операции сдвига .....	40
Выравнивание адресов.....	42
Адресация относительно счетчика команд PC .....	43
Условное исполнение .....	43
Команды доступа к памяти .....	45
Инструкции обработки данных .....	53
Инструкция умножения .....	61
Команды работы с битовыми полями.....	62
Инструкции передачи управления .....	63
Прочие инструкции .....	65
Блок АЦП для измерения напряжений и токов в электрической сети .....	73
Описание регистров управления блока 3 канального $\Sigma\Delta$ АЦП.....	74
Алгоритмы вычисления окончательных результатов и их соответствия внешним сигналам. ....	90
Сигналы тактовой частоты .....	98
Структурная схема .....	98
Встроенный RC Генератор HSI .....	98
Встроенный RC генератор LSI .....	98
Внешний осциллятор HSE .....	99
Внешний осциллятор LSE.....	99
Встроенный блок умножения системной тактовой частоты .....	99
Описание регистров блока контроллера тактовой частоты .....	100
Батарейный домен и часы реального времени .....	109
Часы реального времени .....	109
Регистры аварийного сохранения.....	110
Описание регистров блока батарейного домена .....	110
Порты ввода-вывода .....	118
Описание регистров портов ввода-вывода .....	120

Детектор напряжения питания.....	125
Описание регистров блока PVD .....	125
Таймеры общего назначения.....	129
Функционирование .....	129
Структурная схема .....	129
Инициализация таймера.....	130
Режим таймера.....	130
Режимы счета.....	131
Источник событий для счета .....	133
Режим захвата .....	139
Режим ШИМ .....	141
Описание регистров блока таймера .....	146
Контроллер АЦП .....	158
Преобразование внешнего канала.....	159
Последовательное преобразование нескольких каналов .....	159
Преобразование с контролем границ .....	160
Датчик опорного напряжения .....	160
Датчик температуры.....	160
Время заряда внутренней емкости .....	161
Описание регистров блока контроллера АЦП.....	162
Контроллер I2C интерфейса .....	168
Конфигурация системы.....	168
Протокол I2C.....	168
Сигнал START .....	168
Передача адреса.....	169
Передача данных .....	169
Сигнал STOP .....	169
Описание регистров контроллера I2C .....	169
Контроллер SSP.....	174
Основные характеристики модуля SSP .....	174
Программируемые параметры .....	175
Характеристики интерфейса SPI.....	176
Характеристики интерфейса Microwire .....	176
Характеристики интерфейса SSI.....	176
Общий обзор модуля SSP .....	176
Примеры конфигурации модуля в ведущем и ведомом режимах .....	189
Интерфейс прямого доступа к памяти.....	193
Программное управление модулем.....	195
Описание регистров контроллера SSP.....	195
SSPx_CR0. Регистр управления 0 .....	196
SSPx_CR1. Регистр управления 1 .....	196
SSPx_DR. Регистр данных.....	197
SSPx_SR. Регистр состояния .....	198
SSPx_CPSR. Регистр делителя тактовой частоты .....	199
SSPx_IMSC. Регистр установки и сброса маски прерывания .....	199
SSPx_RIS. Регистр состояния прерываний.....	200
SSPx_MIS. Регистр маскированного состояния прерываний.....	200
SSPx_ICR. Регистр сброса прерываний .....	200
SSPx_DMACR. Регистр управления прямым доступом к памяти .....	201
Прерывания.....	202
Сигналы запроса на прерывание:.....	202

SSPRXINTR.....	202
SSPTXINTR.....	202
SSPRORINTR.....	203
SSPRTINTR.....	203
SSPINTR.....	203
Контроллер UART.....	204
Основные характеристики модуля UART.....	204
Программное управление скоростью обмена.....	204
Отличительные особенности контроллера UART.....	205
Функциональные возможности.....	205
Модуль приемопередатчика:.....	206
Описание функционирования блока UART.....	207
Описание функционирования ИК кодека IrDA SIR.....	209
Описание работы UART.....	211
Линии управления модемом.....	216
Интерфейс прямого доступа к памяти.....	218
Прерывания.....	220
Программное управление модулем.....	223
Обобщенные данные о регистрах устройства.....	223
Контроллер LCD.....	239
Организация памяти данных LCD.....	240
Описание регистров контроллера LCD.....	243
Прерывания и исключения.....	246
Состояние исключений.....	246
Типы исключений.....	246
Обработчики исключений.....	247
Приоритеты исключений.....	248
Вход в обработчик и выход из обработчика.....	249
Управление электропитанием.....	253
Переход в режим пониженного энергопотребления.....	253
Ожидание прерывания.....	253
Переход в режим ожидания по выходу из обработчика исключения (режим sleep-on-exit).....	253
Выход из состояния ожидания.....	253
Выход из ожидания по команде WFE.....	254
Рекомендации по программированию режима энергопотребления.....	254
Контроллер прерываний NVIC.....	256
Регистр разрешения прерываний.....	256
Регистр запрета прерываний.....	257
Регистр установки состояния ожидания для прерывания.....	257
Регистр сброса состояния ожидания для прерывания.....	258
Регистры приоритета прерываний.....	258
Прерывания, срабатывающие по уровню сигнала.....	259
Аппаратное и программное управление прерываниями.....	260
Рекомендации по работе с контроллером прерываний.....	260
Блок управления системой ядра.....	262
Упрощенный доступ к регистрам блока управления системой из среды разработки программного обеспечения.....	262
Сторожевые таймеры.....	268
Описание регистров блока сторожевых таймеров.....	268
Предельно-допустимые характеристики микросхемы.....	273

Электрические параметры микросхемы .....	274
Типовые зависимости основных электрических параметров микросхемы от режимов и условий эксплуатации .....	278
Габаритный чертеж микросхемы.....	279
Информация для заказа .....	280
Лист регистрации изменений.....	281

**Описание выводов**

Таблица 1

Вывод	Число выводов корпуса 64-LQFP	Дополнительные функции вывода			
		Аналог.	Основ.	Альтер.	Переопр.
<b>Порт А</b>					
PA0	12	S0	TMR0_CH1	-	-
PA1	13	S1	TMR0_CH1N	-	-
PA2	14	S2	-	-	-
PA3	15	S3	-	-	-
PA4	16	S4	-	-	-
PA5	17	S5	-	-	-
PA6	18	S6	-	-	-
PA7	19	S7	-	-	-
PA8	20	S8	TMR0_ETR	-	-
PA9	21	S9	TMR0_BLK	-	-
PA10	22	S10	-	-	-
PA11	23	S11	-	-	-
PA12	24	S12	-	-	-
PA13	25	S13	-	-	-
PA14	26	S14	-	-	-
PA15	27	S15	-	-	-
<b>Порт В</b>					
PB0/MODE0	8	-	UART0_TXD	-	-
PB1	9	-	UART0_RXD	-	-
PB2/SWCLKTCK	62	-	nSIROUT0	-	-
PB3/SWDIO	63	-	nSIRIN0	-	-
PB4	59	OSC_IN32	nUART0DTR	-	-
PB5	60	OSC_OUT32	nUART0RTS	-	-
PB6	40	ADC7/BP0	nUART0RI	-	-
PB7	41	ADC6/BP1	nUART0DCD	-	-
PB8	42	ADC5/BP2	nUART0DSR	-	-
PB9	43	ADC4/BP3	nUART0CTS	-	-
PB10	28	S16	-	-	-
PB11	29	S17	-	-	-
PB12	30	S18	-	-	-
PB13	31	S19	-	-	-
PB14	32	S20	-	-	-
<b>Порт С</b>					
PC0/MODE1	7	-	UART1_TXD	-	-
PC1	48	ADC3	UART1_RXD	-	-
PC2	47	ADC2, VLCD	TMR1_CH1	-	-
PC3	46	ADC1_REF+	TMR1_CH1N	-	-
PC4	45	ADC0_REF-	-	-	-
PC5	33	S21	TMR1_ETR	-	-
PC6	34	S22	TMR1_BLK	-	-
PC7	35	S23	EXT_INT0	-	-
PC8	11	S24	EXT_INT1	-	-



## Спецификация K1986BK234, K1986BK234K

PC9	10	S25	EXT_INT2	-	-
PC10	36	S26	SSP_FSS	-	-
PC11	37	S27	SSP_CLK	-	-
PC12	38	S28	SSP_RXD	-	-
PC13	39	S29	SSP_TXD	-	-
PC14	3		SCL	-	-
PC15	4		SDA	-	-
<b>Порт АЦП</b>					
VR_1V	50				
I0P	51				
I0N	52				
V0P	53				
V0N	54				
I3P	55				
I3N	56				
<b>Системное управление</b>					
RESET	44	Сигнал внешнего сброса			
WAKEUP	58	Сигнал внешнего выхода из режима Standby			
COV_DET	58				
OSC_IN	2	Вход генератора HSE			
OSC_OUT	1	Выход генератора HSE			
<b>Питание</b>					
Ucc	5	Питание 2,2...3,6 В			
AGND	49				
BUcc	61	Батарейное питание 1.8...3,6 В			
GND	6				
AUcc	57	Аналоговое питание $\Sigma\Delta$ АЦП 2,4...3,6 В			
<b>Выводы для тестирования и исследования</b>					
JTAG_EN	64				

Диаграмма расположения выводов в корпусах

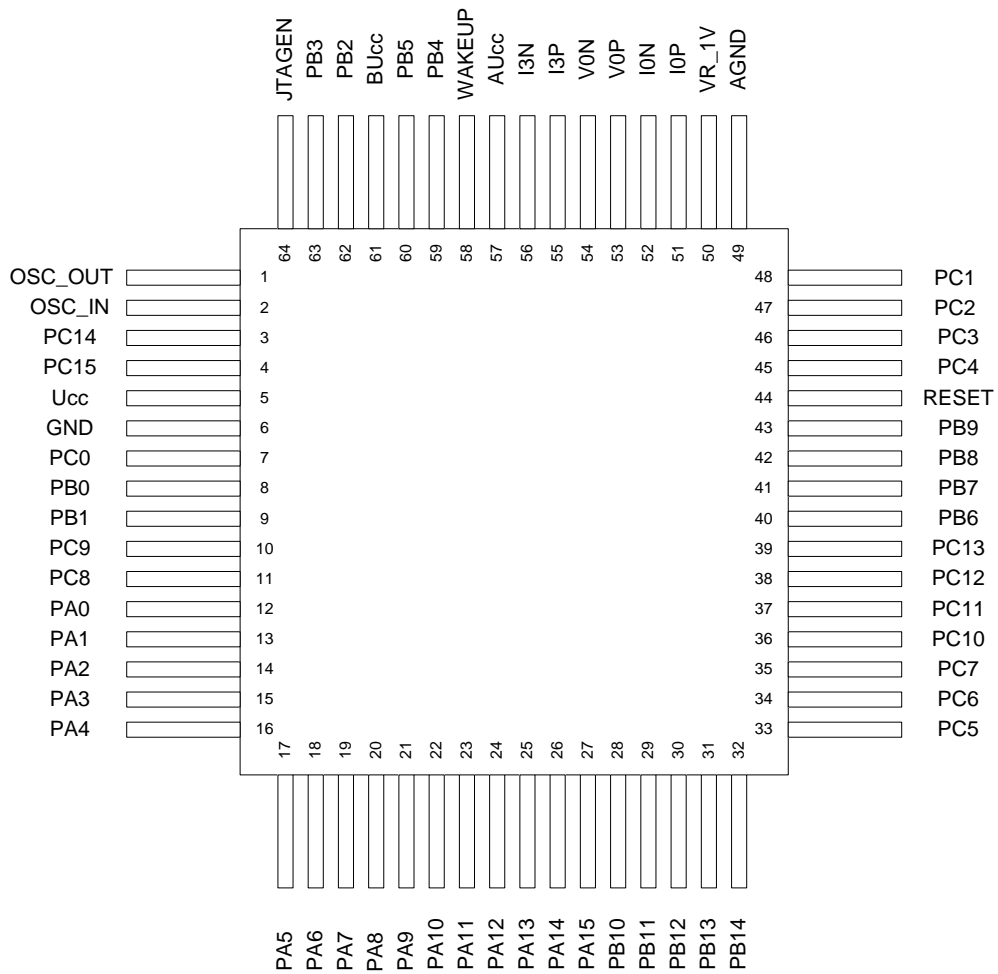


Рис. 1 – 64-х выводной пластиковый корпус LQFP

Структурная блок-схема

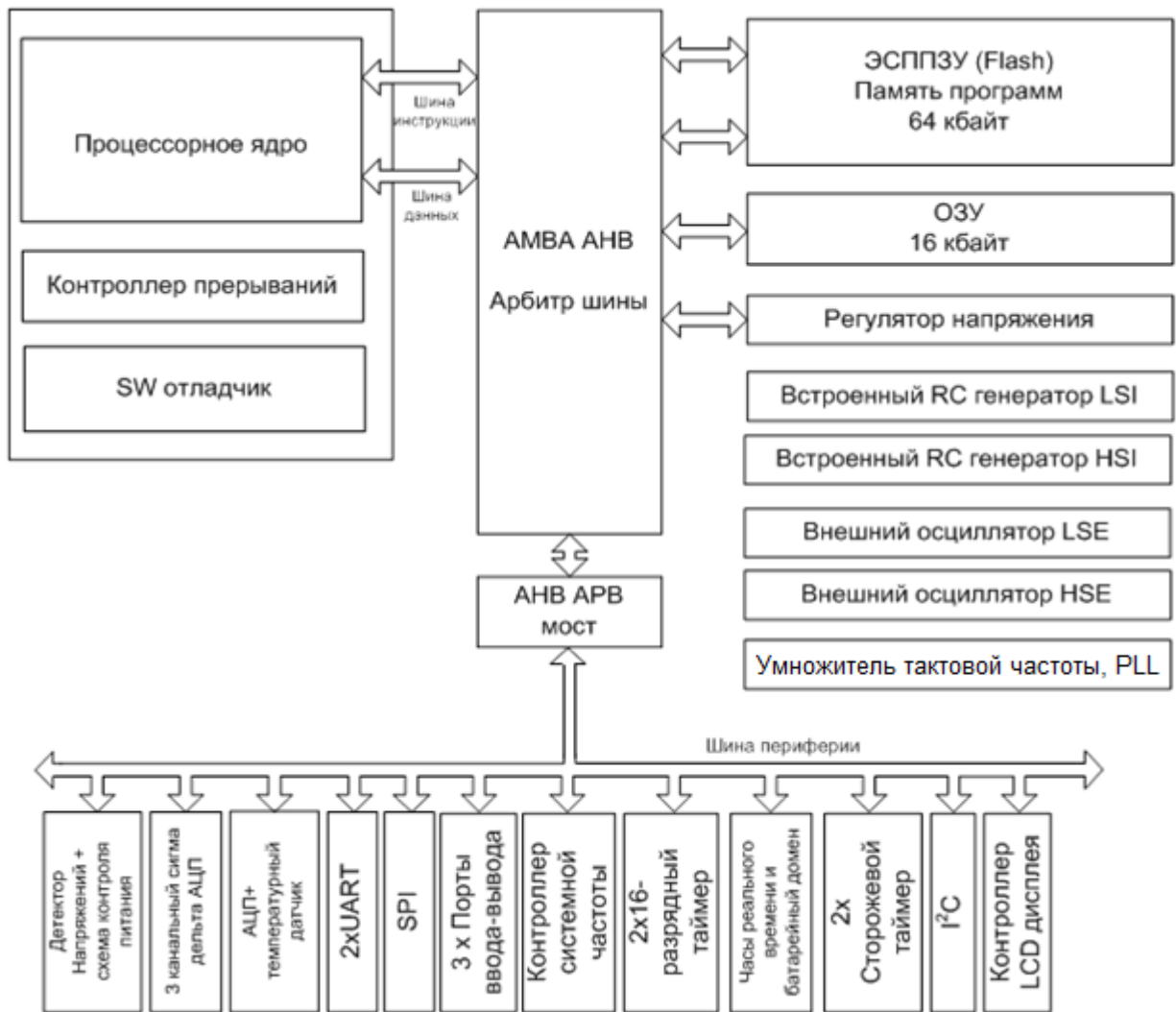


Рис. 2 Структурная блок-схема

## Описание функционирования микросхемы

### Система питания

Микроконтроллер этой серии имеет несколько типов выводов питания:

- *U<sub>сс</sub> вывод*: Основное питание микросхемы, включает питание пользовательских выводов, встроенного регулятора напряжения, умножителя тактовой частоты PLL, генераторов и АЦП последовательного приближения. Входное напряжение должно быть в пределах от 2,2 до 3,6 В.
- *VU<sub>сс</sub> вывод*: Питание батарейного домена используется при отсутствии основного питания U<sub>сс</sub> для питания батарейного домена и LSE генератора. Переключение с основного питания на батарейное происходит автоматически при снижении уровня U<sub>сс</sub> ниже 2,0 В. Переключение с батарейного питания на основное происходит автоматически спустя примерно 4 мс после превышения уровнем U<sub>сс</sub> 2,0 В. Входное напряжение должно быть в пределах от 1,8 до 3,6 В. Если в системе не требуется батарейного питания вывод VU<sub>сс</sub> должен быть объединен с U<sub>сс</sub>.
- *AU<sub>сс</sub> вывод*: Питание аналоговых блоков сигма дельта АЦП и формирователя опоры выведено на отдельные выводы для уменьшения помех создаваемых работой других блоков. На данные выводы должно подаваться напряжения с того же источника что и U<sub>сс</sub>, но при этом на печатной плате должны быть применены меры по снижению наводки помех. Для корректной работы АЦП входное напряжение должно быть в пределах от 3,0 до 3,6 В. Если входное напряжение будет в пределах от 2,2 до 3,0 В то корректная работа АЦП не гарантируется.
- *GND вывод*: Основная земля питания.
- *AGND вывод*: Земля аналогового питания AU<sub>сс</sub>. Данные выводы должны соединяться с GND, но при этом на печатной плате должны быть применены меры по снижению наводки помех.

Структурная схема подачи питания

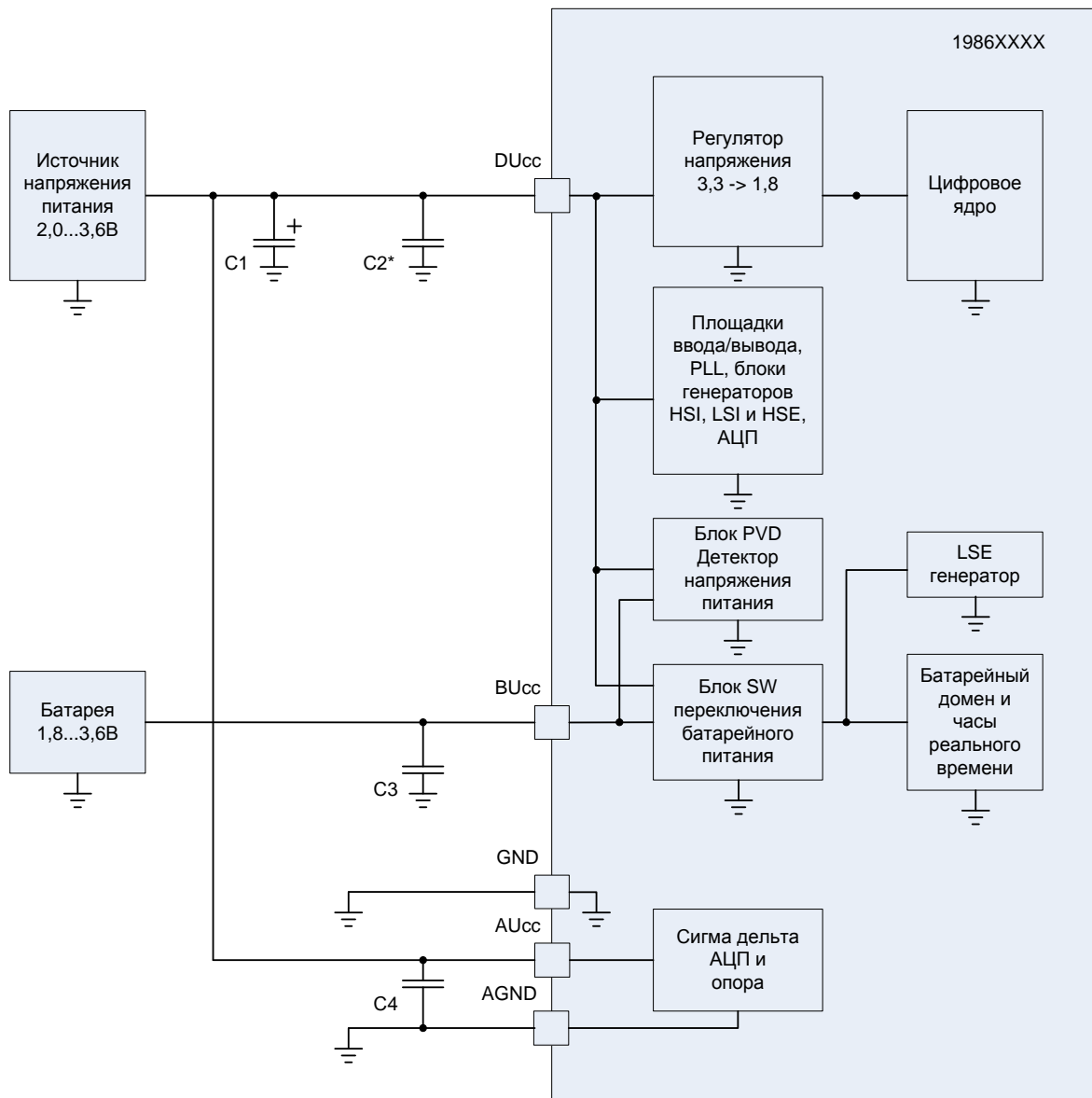


Рис. 3 – Схема подачи питания микросхемы

Примечание:

- \* - конденсаторы должны быть установлены у каждого вывода питания
  1. Конденсатор C1 = 22 мкФ, C2 = C3 = C4 = C5 = 0,1 мкФ.
  2. Если не используется батарейное питание, то вывод BUcc должен быть объединен с Ucc
  3. Если используется АЦП или ЦАП, то напряжение питания Ucc (AUcc и AUcc1) должно быть в пределах от 2,4 до 3,6В

Микроконтроллер имеет несколько режимов энергопотребления, подробнее смотри раздел «

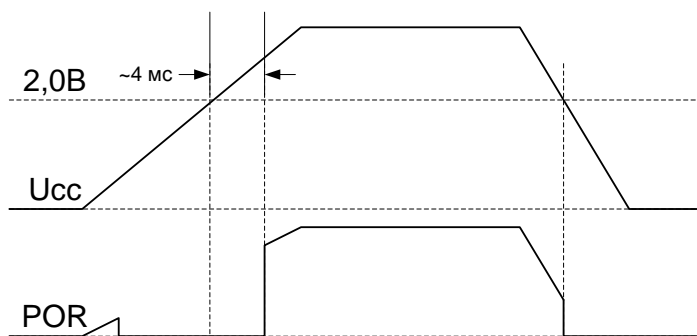
Управление электропитанием».

Микроконтроллер имеет встроенный детектор напряжения питания, подробнее смотри раздел «

Детектор напряжения питания».

**Схема сброса при включении и выключении основного питания**

При включении питания вырабатывается внутренний сигнал сброса POR для цифровой части, питание U<sub>CC</sub> нарастает и, пока оно не превысило уровень 2,0 В, сигнал сброса POR удерживается; после превышения данного уровня сигнал POR выдается еще на протяжении ~ 4 мс для того, чтобы гарантировано установилось напряжение питания, после чего сигнал POR снимается, и схема может начать работать.



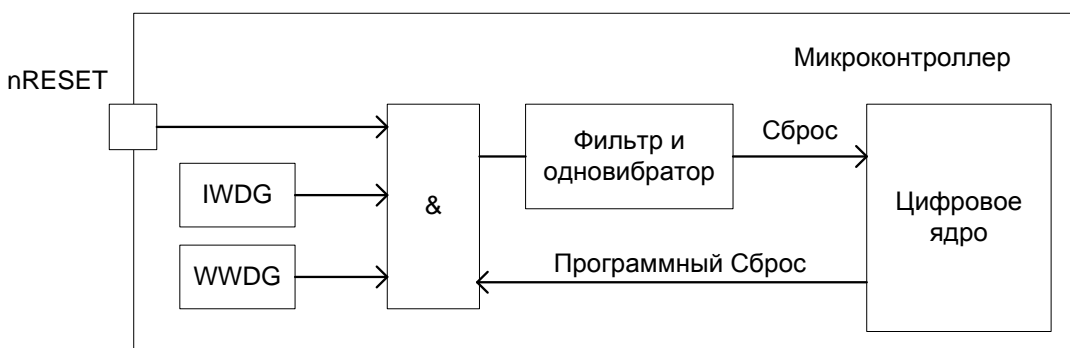
**Рис. 4** Схема включения основного питания U<sub>CC</sub>

При снижении напряжения питания U<sub>CC</sub> ниже уровня 2,0 В сигнал POR вырабатывается без задержки.

Сигнал POR также служит для переключения питания батарейного домена между BU<sub>CC</sub> и U<sub>CC</sub>.

При включении основного напряжения питания U<sub>CC</sub> автоматически включается встроенный регулятор напряжения для формирования напряжения DU<sub>CC</sub> (см Рис. 3) питания цифрового ядра. В ходе работы микроконтроллера встроенный регулятор может быть отключен.

Установка микроконтроллера в исходное состояние может быть произведена внешним сигналом сброса nRESET, или внутренними сигналами сброса сторожевых таймеров (IWDG, WWDG), или программным сбросом. При этом сигнал nRESET формируется специальной схемой сброса, содержащий фильтр импульсных помех и одновибратор для увеличения длительности этого сигнала.



**Рис. 5** Схема формирования сигнала сброса

При подаче на вход nRESET импульсов сброса длительностью менее 10 нс они отфильтровываются и не приводят к сбросу процессора. Если длительность импульса больше 200 нс, вырабатывается сигнал сброса. При этом длительность сформированного сигнала сброса будет не менее 20 мкс.

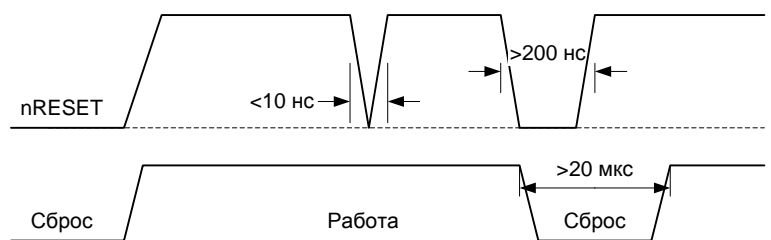


Рис. 6 Временная диаграмма формирования сигнала сброса

## Организация памяти

### Структурная схема

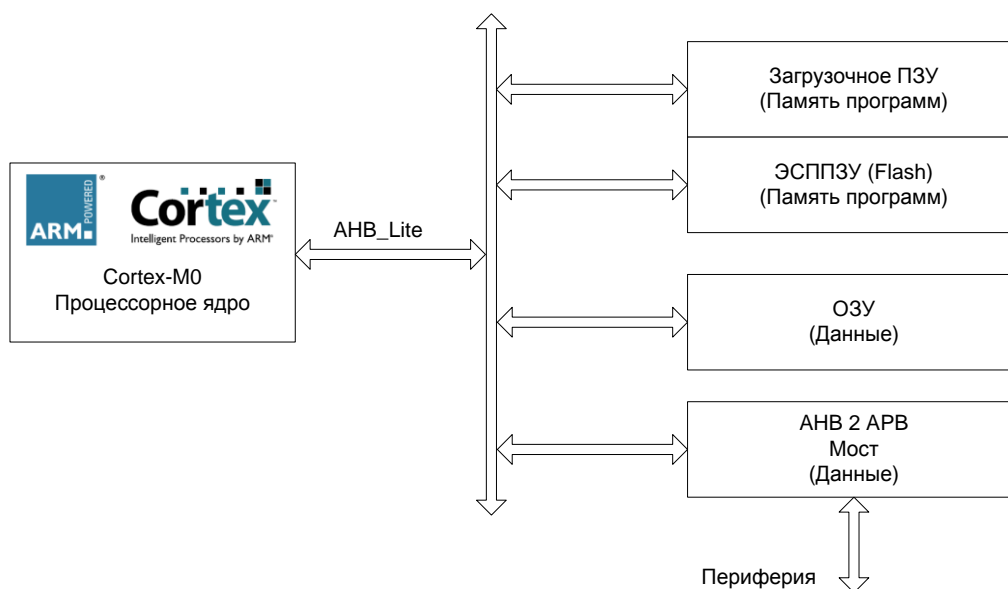
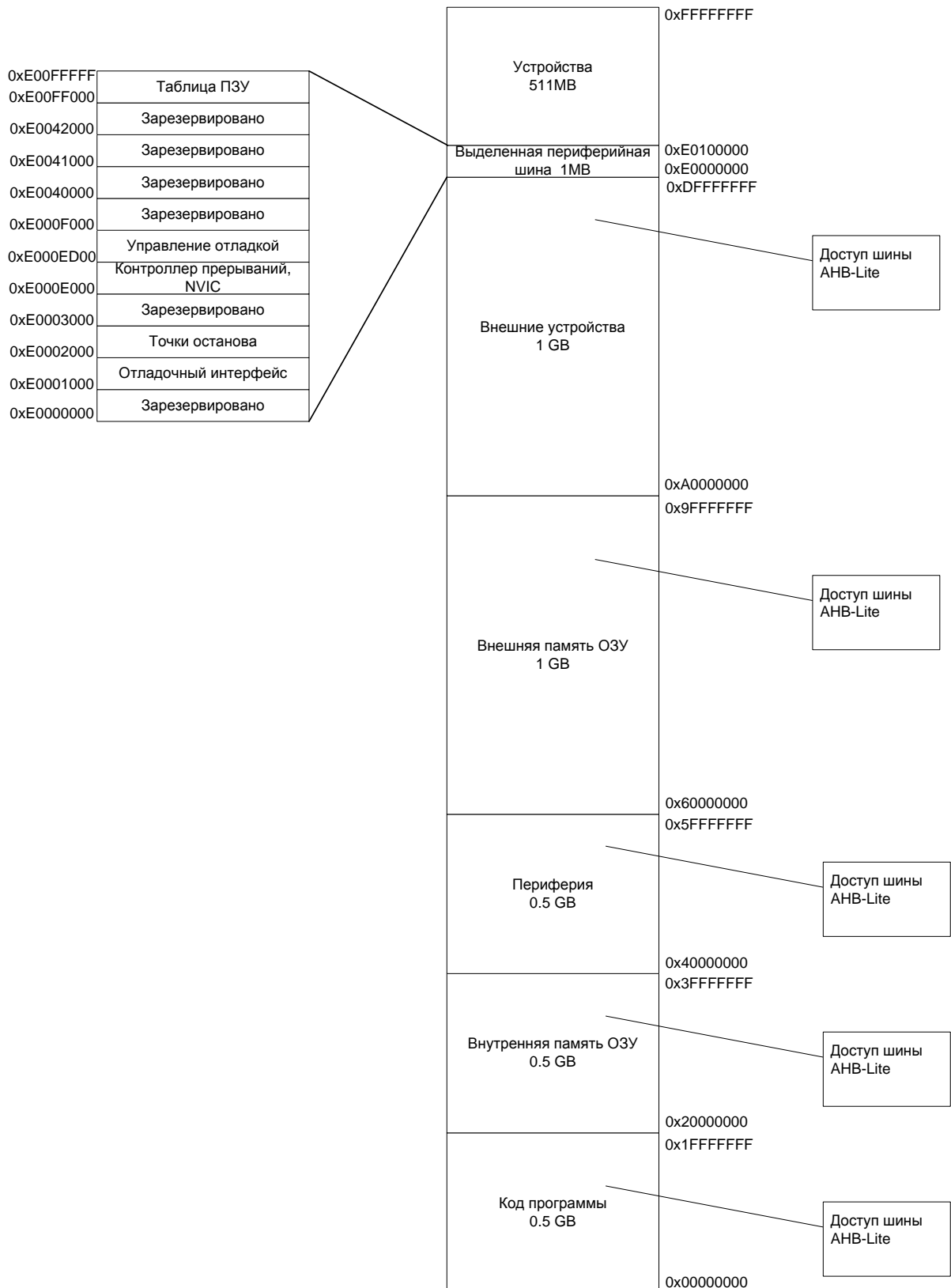


Рис. 7 Схема организации памяти

Процессорное ядро имеет одну системную шину AHB Lite (шина выборки инструкций и данных).

Все адресное пространство микроконтроллера единое и имеет максимальный объем 4 Гбайта. В данное адресное пространство отображаются различные модули памяти и периферии.





**Рис. 8** Адресное пространство микроконтроллера

## Спецификация K1986BK234, K1986BK234K

Адрес	Размер	Блок	Примечание
<b>Память программ</b>			
0x0000_0000		BOOT ROM	Загрузочная программа FPOR=0
0x0000_0000		EEPROM	Область Flash памяти программ с пользовательской программой FPOR=1
<b>Память данных</b>			
0x2000_0000		SYSTEM RAM	Область внутреннего ОЗУ
<b>Периферия</b>			
0x4000_0000		SPI	Регистры контроллера интерфейса SSP (см. раздел «Контроллер SSP»)
0x4000_8000		UART1	Регистры контроллера интерфейса UART1 (см. раздел «UART загрузчик»)
0x4001_0000		UART2	Регистры контроллера интерфейса UART2 (см. раздел «UART загрузчик»)
0x4001_8000		EEPROM_CNTRL	Регистры контроллера Flash памяти программ
0x4002_0000		RST_CLK	Регистры контроллера сигналов тактовой частоты
0x4003_0000		I <sup>2</sup> C	Регистры контроллера интерфейса I <sup>2</sup> C
0x4003_8000		LCD	Регистры контроллера LCD дисплея
0x4004_0000		ADC	Регистры управления АЦП
0x4004_8000		WWDG	Регистры контроллера сторожевого таймера WWDG
0x4005_0000		IWDG	Регистры контроллера сторожевого таймера IWDG
0x4005_8000		POWER	Регистры детектора напряжения питания
0x4006_0000		BKP	Регистры доступа и управления батарейным доменом
0x4006_8000		ADCIU	Регистры управления $\Sigma\Delta$ АЦП
0x4007_0000		TIMER1	Регистры управления 16-ти разрядного таймера 1
0x4007_8000		TIMER2	Регистры управления 16-ти разрядного таймера 2
0x4008_0000		PORTA	Регистры управления порта ввода-вывода А
0x4008_8000		PORTB	Регистры управления порта ввода-вывода В
0x4009_0000		PORTC	Регистры управления порта ввода-вывода С
<b>SYSTEM REGION</b>			
0xE000_0000			Системные регистры процессор ARM Cortex-M0

**Загрузочное ПЗУ и режимы работы микроконтроллера**

После включения питания и снятия сигналов сброса: внутренних (POR) и внешних (RESET), - микроконтроллер начинает выполнять программу из информационной загрузочной области FLASH. В загрузочной программе микроконтроллер определяет, в каком из режимов он будет функционировать и переходит в этот режим. Режим функционирования определяется внешними выводами MODE[1:0] (PB[0], PC[0]). Так же устанавливается бит FPOR в регистре BKP\_REG\_0E (см. Таблица 77), который может быть сброшен только при отключении основного питания Uсс. После перезапуска микроконтроллера уровни на выводах MODE[1:0] не влияют на режим функционирования микроконтроллера, если установлен бит FPOR. В пользовательской программе выводы PB[0], PC[0] могут использоваться пользователем.

**Таблица 3**

MODE[1:0]	Режим	Стартовый адрес/ таблица векторов прерываний	Описание
00	Микроконтроллер в режиме отладки	0x0000_0000	Процессор начинает выполнять программу из внутренней FLASH памяти программ. При этом разрешается работа отладочного интерфейса SW (Serial Wire).
01	UART загрузчик	Определяется пользователем	Микроконтроллер через интерфейс UART1 на выводах PC [0], PB[0] получает код программы в ОЗУ для исполнения.
10	UART загрузчик	Определяется пользователем	Микроконтроллер через интерфейс UART1 на выводах PC [0], PB[0] получает код программы в ОЗУ для исполнения.
11	Запрещенная ситуация	-	Режим для проверки микросхемы после производства. Микросхема перестает работать как микроконтроллер.

При работе в режиме отладки разрешается работа отладочного интерфейса Serial Wire. При этом к микроконтроллеру может быть подключен Serial Wire адаптер (не входит в комплект поставки) с помощью которого программные средства разработки позволяют работать с микроконтроллером в отладочном режиме.

В отладочном режиме можно:

- стирать, записывать, считывать внутреннюю FLASH память программ
- считывать и записывать содержимое ОЗУ, периферии
- выполнять программу в пошаговом режиме
- запускать программу в нормальном режиме
- останавливать программу по точкам остановки
- просматривать переменные выполняемой программы
- проводить трассировку хода выполнения программного обеспечения

**UART загрузчик**

В режиме UART загрузчика используют один и тот же периферийный модуль UART1, один и тот же протокол обмена (см. раздел «Протокол обмена по UART»).

**Таблица 4**

Режим	UART0_RXD	UART0_TXD
100b	PB1	PB0
101b	PB1	PB0
110b	PB1	PB0

Данные режимы предоставляют достаточный набор операций, необходимых для записи в ОЗУ какой-либо программы (в частности программатора FLASH-памяти), верификации ее и запуска на выполнение. Кроме того, существует возможность задания внешним устройством скорости обмена. Помимо доступа к ОЗУ может быть осуществлен доступ и к другим адресным диапазонам (ЭСППЗУ, ПЗУ, Периферия).

В качестве источника тактовой частоты UART1 используется внутренний RC-генератор HSI с частотой 8 МГц. Так как имеется разброс значений частоты HSI, то требуется этап подбора значения делителя частоты UART1 для синхронизации с внешним устройством.

**Параметры связи по UART**

Для связи по UART выбраны следующие параметры канала связи:

- Начальная скорость – 9600 бод
- Количество бит данных – 8
- Четность – нет
- Количество стоп бит – 1
- Загрузчик не использует FIFO UART1.
- Загрузчик всегда выступает в качестве Slave, а внешнее устройство, подающее команды – в качестве Master.
- Данные передаются младшим битом вперед.

**Протокол обмена по UART**

После синхронизации с Master загрузчик переходит в диспетчер команд. Таким образом, Master-у доступны следующие команды:

**Таблица 5 Команды UART загрузчика**

Команда	Код	ASCII Символ	Описание
CMD_SYNC	0x00		Пустая команда. Загрузчик ее принимает, но ничего по ней не делает.
CMD_CR	0x0D		Выдача приглашения Master-у.
CMD_BAUD	0x42	'B'	Установка скорости обмена.
CMD_LOAD	0x4C	'L'	Загрузка массива байт.
CMD_VFY	0x59	'Y'	Выдача массива байт.
CMD_RUN	0x52	'R'	Запуск программы на выполнение.

## **Синхронизация с внешним устройством**

Начальные условия

На этапе синхронизации с внешним устройством (Master) вывод PB1 используется как вход. Master постоянно посылает в канал синхросимвол – 0. Загрузчик подстраивает свою скорость таким образом, чтобы минимизировать ошибки обмена. Как только Загрузчик настроил скорость он переходит в диспетчер команд и выдает приглашение (3 байта 0x0D (перевод строки), 0x0A (возврат каретки), 0x3E ('>'),) Master-у.

Master завершает выдачу синхросимволов и, теперь, может подавать команды согласно протоколу обмена.

### **Команда CMD\_SYNC**

Пустая команда.

Загрузчик (Slave) ее принимает, но ничего по ней не делает. Код команды соответствует символу синхронизации.

**Таблица 6**

Код команды	CMD_SYNC = 0x00
ASCII символ, соответствующий коду команды	нет
Количество параметров команды	0
Формат команды:	
Master выдает код команды CMD_SYNC.	Slave принимает команду. Если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD (см. раздел «Сообщения об ошибках») и завершает обработку текущей команды.

### **Команда CMD\_CR**

Выдача приглашения Master-у.

**Таблица 7**

Код команды	CMD_CR = 0x0D
ASCII символ, соответствующий коду команды	нет
Количество параметров команды	0
Формат команды:	
Master выдает код команды CMD_CR.	Slave принимает команду. Если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды. Выдает код команды CMD_CR. Выдает код 0x0A Выдает код 0x3E (ASCII символ '>')

### **Команда CMD\_BAUD**

Установка скорости обмена.

**Таблица 8**

Код команды	CMD_BAUD = 0x42
ASCII символ, соответствующий коду команды	'B'
Количество параметров команды	1
Параметр	Новое значение скорости обмена [бод].
Формат команды:	
Master выдает код команды CMD_BAUD	Slave принимает команду. Если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды.
Master выдает параметр.	Slave принимает параметр. Если параметр принят с ошибками, то выдает код ошибки ERR_CHN или ERR_BAUD (см. раздел «Сообщения об ошибках») и завершает обработку текущей команды. Выдает код команды CMD_BAUD. Устанавливает новое значение скорости обмена.

### **Команда CMD\_LOAD**

Загрузка массива байт в память микроконтроллера.

**Таблица 9**

Код команды	CMD_LOAD = 0x4C
ASCII символ, соответствующий коду команды	'L'
Количество параметров команды	2
Параметр 1.	Адрес памяти приемника данных.
Параметр 2.	Размер массива в байтах
Формат команды:	
Master выдает код команды CMD_LOAD	Slave принимает команду. Если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды.
Master выдает параметр 1.	Slave принимает параметр 1. Если параметр принят с ошибками, то выдает код ошибки ERR_CHN и завершает обработку текущей команды.
Master выдает параметр 2.	Slave принимает параметр 2. Если параметр принят с ошибками, то выдает код ошибки ERR_CHN и завершает обработку текущей команды. Выдает код команды CMD_LOAD.

Master выдает массив байт младшим байтом вперед.	Slave принимает массив байт. Если хотя бы один байт принят с ошибками, то выдает код ошибки ERR_CHN и завершает обработку текущей команды, не дожидаясь окончания принятия всего массива. По окончании принятия массива выдает код ответа REPLY_OK = 0x4B ('K').
--	--

### Команда CMD\_VFY

Выдача массива байт из памяти микроконтроллера.

**Таблица 10**

Код команды	CMD_VFY = 0x59
ASCII символ, соответствующий коду команды	'Y'
Количество параметров команды	2
Параметр 1	Адрес памяти источника данных
Параметр 2	Размер массива в байтах
Формат команды:	
Master выдает код команды CMD_VFY	Slave принимает команду. Если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды.
Master выдает параметр 1	Slave принимает параметр 1. Если параметр принят с ошибками, то выдает код ошибки ERR_CHN и завершает обработку текущей команды.
Master выдает параметр 2	Slave принимает параметр 2. Если параметр принят с ошибками, то выдает код ошибки ERR_CHN и завершает обработку текущей команды. Выдает код команды CMD_VFY. Выдает массив байт младшим байтом вперед. По окончании передачи массива выдает код ответа REPLY_OK = 0x4B ('K')

### Команда CMD\_RUN

Запуск программы на выполнение.

**Таблица 11**

Код команды	CMD_RUN = 0x52
ASCII символ, соответствующий коду команды	'R'
Количество параметров команды	1
Параметр.	Адрес таблицы векторов загруженной программы
Формат команды:	

Master выдает код команды CMD_RUN.	Slave принимает команду. Если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды.
Master выдает параметр.	Slave принимает параметр. Если параметр принят с ошибками, то выдает код ошибки ERR_CHN и завершает обработку текущей команды. Выдает код команды CMD_RUN. Устанавливает значение MSP и PC согласно таблице векторов (NVIC не перепрограммируется) и, таким образом, Slave завершает выполнение команды.

### **Прием параметров команды**

Параметры команд – это 4-х байтные числа. Параметры передаются младшим байтом вперед.

В качестве значения параметра запрещено использовать число 0xFFFFFFFF.

Если при приеме параметра обнаружена аппаратная ошибка (UART установил в '1' какой-либо из флагов ошибки), то прием параметров не прекращается.

Анализ всех видов ошибок, связанных с передачей параметров, загрузчик производит только после принятия всех параметров команды.

### **Сообщения об ошибках**

Сообщения об ошибках – это 2-х байтные последовательности символов. Первый символ всегда 0x45 ('E'). Второй символ обозначает тип ошибки.

После выдачи сообщения об ошибке загрузчик переходит в режим ожидания следующей команды, поэтому Master после получения такого сообщения прекращает передачу байт, относящихся к текущей команде.

После принятия сообщения об ошибке Master должен подавать команду CMD\_CR до тех пор, пока не получит корректный ответ, соответствующий этой команде.

Возможны следующие сообщения об ошибках: ERR\_CHN, ERR\_CMD, ERR\_BAUD.

#### **Ошибка ERR\_CHN**

Аппаратная ошибка UART.

Код ошибки 0x69 ('i').

Выдается, если UART установил в '1' один из аппаратных флагов ошибки при приеме очередного байта.

#### **Ошибка ERR\_CMD**

Принята неизвестная команда.

Код ошибки 0x63 ('c').

Выдается диспетчером команд, если принят неизвестный код команды.



**Ошибка ERR\_BAUD**

Принята неизвестная команда.

Код ошибки 0x62 ('b').

Выдается диспетчером команд, если по принятому от Master-а значению скорости обмена невозможно вычислить корректное значение делителя частоты UART.

**Контроллер FLASH памяти программ**

Микроконтроллер содержит встроенную Flash память программ с объемом 64Кбайт основной памяти программ и 4Кбайта информационной памяти. В микроконтроллере реализованы два режима работы:

- обычный режим;
- режим программирования.

В обычном режиме (бит CON = 0, регистр EEPROM\_CMD (см. Таблица 13)) доступна основная память программ через системную шину для выборки инструкций и данных кода программы.

В режиме программирования (бит CON=1, регистр EEPROM\_CMD) основная и информационная память доступны как периферийное устройство и могут быть использованы для нужд разработчика приложения. В режиме программирования программный код должен выполняться из области системной шины или ОЗУ. Выполнение программного кода из Flash памяти программ в режиме программирования невозможно.

**Работа Flash памяти программ в обычном режиме**

Скорость доступа во Flash память ограничена и составляет порядка 55 нс, в результате выдача новых значений из Flash памяти может происходить с частотой не более 18 МГц. Для того, что бы процессорное ядро могло получать новые инструкции на больших частотах, в микроконтроллере реализуется Flash память с физической организацией 64К на 32 разряда. Таким образом, за 55 нс из Flash памяти извлекается 8 байт, в которых может быть закодировано 2 инструкции процессора. И пока ядро выполняет эти инструкции, из памяти извлекается следующая порция данных. Таким образом, тактовая частота может превышать частоты извлечения данных из памяти в несколько раз при линейном выполнении программы. При возникновении переходов в выполнении программы, когда из памяти программ не выбраны нужные инструкции, возникает пауза в несколько тактов процессора для того, чтобы данные успели считаться из Flash. Число тактов паузы зависит от тактовой частоты процессора. Так, при работе с частотой ниже 18 МГц пауза не требуется, так как Flash память успевает выдать новые данные за один такт, при частоте от 18 до 36 МГц требуется один такт паузы, и так далее. Число тактов паузы задается в регистре EEPROM\_CMD битами Delay[1:0]. В Таблица 12 приведены характеристики необходимой паузы для работы Flash памяти программ.

**Таблица 12**

Delay[1:0]	Тактов паузы	Тактовая частота	Примечание
0x00	0	До 18 МГц	
0x01	1	До 36 МГц	

Число тактов паузы устанавливается до момента повышения тактовой частоты или после снижения тактовой частоты.

**Работа Flash памяти программ в режиме программирования**

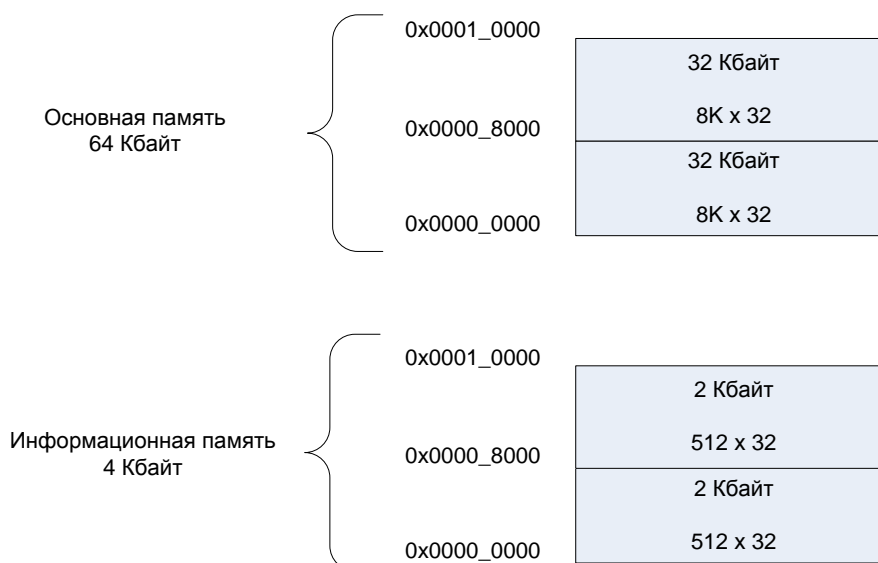
В режиме программирования Flash память программ не может выдавать инструкции и данные процессору, поэтому перевод памяти в режим

программирования (установка бита CON = 1) возможен только программой исполняемой из ОЗУ.

Информационная память может быть назначена на адрес 0x0000\_0000 (вместо загрузочной программы) с помощью управляющего бита FPOR (расположен в батарейном домене).

В режиме программирования возможны следующие операции как с основной (бит IFREN = 0, регистр EEPROM\_CON (см. Таблица 13)), так и с информационной (бит IFREN = 1) памятью:

- стирание блока памяти размером 2Кбайт или 32Кбайт;
- стирание страницы памяти размером 512 байт;
- запись 32-х битного слова в память;
- чтение 32-х битного слова из памяти.



**Рис. 9** Структура памяти

### **Стирание блока памяти размером 2Кбайт или 32Кбайт.**

Стирание памяти возможно только в режиме программирования. Для стирания всей памяти надо установить необходимое значение в бит IFREN в регистре EEPROM\_CMD (1 – для информационной памяти и 0 – для основной памяти), затем установить биты XE, MAS1 и ERASE (см. Таблица 14) в единицу, и спустя время  $T_{nvs} = 10$  мкс установить бит NVSTR в единицу. Полное стирание памяти длится время  $T_{me} = 40$  мс. Спустя это время необходимо очистить бит ERASE, и спустя время  $T_{nh1} = 100$  мкс очистить биты XE, MAS1 и NVSTR. Последующие операции с памятью можно выполнять спустя время  $T_{rcv} = 1$  мкс. Временная диаграмма стирания памяти представлена на Рис. 10.

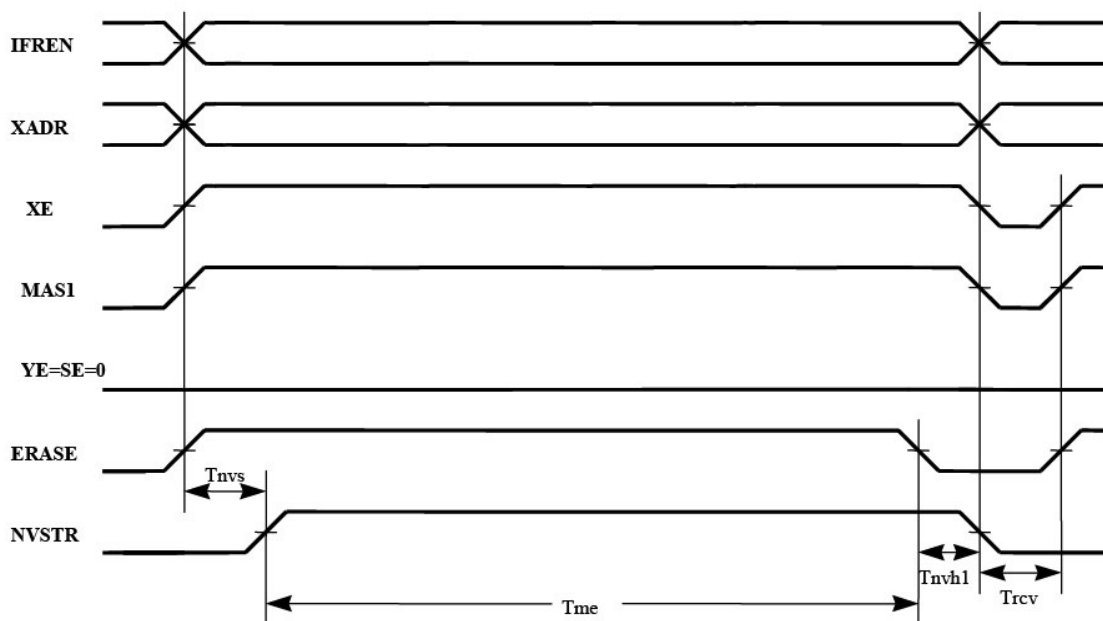


Рис. 10 Временная диаграмма стирания памяти

### Стирание страницы памяти размером 512 байт

Стирание страницы памяти возможно только в режиме программирования. Для стирания страницы памяти надо установить необходимое значение в бит IFREN (1 – для информационной памяти и 0 – для основной памяти), затем установить адрес стираемой страницы в регистре EEPROM\_ADR и установить биты XE и ERASE в единицу, и спустя время  $T_{nvs} = 10$  мкс установить бит NVSTR в единицу. Стирание страницы памяти длится время  $T_{erase} = 40$  мс. Спустя это время необходимо очистить бит ERASE, и спустя время  $T_{nvhl} = 5$  мкс очистить биты XE и NVSTR. Последующие операции с памятью можно выполнять спустя время  $T_{rcv} = 1$  мкс. Временная диаграмма стирания страницы памяти представлена на Рис. 11.

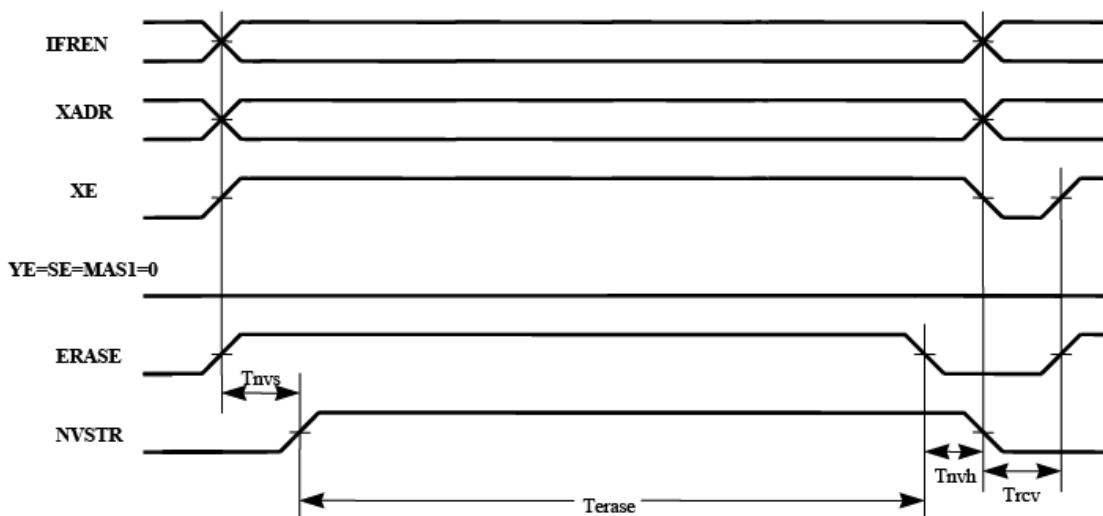


Рис. 11 Временная диаграмма стирания страницы

### Запись 32-х битного слова в память

Запись в память возможно только в режиме программирования. Для записи в память надо установить необходимое значение в бит IFREN (1 – для информационной памяти и 0 – для основной памяти), затем установить адрес по которому производится запись в регистре EEPROM\_ADR, в регистр EEPROM\_DI записать записываемое в память слово и установить биты XE и PROG в единицу, и спустя время  $T_{nvs} = 5$  мкс установить бит NVSTR в единицу. Спустя время  $T_{pgs} = 10$  мкс установить бит YE в единицу. Запись в память длится время  $T_{prog} = 40$  мкс. Спустя это время необходимо очистить бит YE, и спустя время  $T_{adh} = 20$  нс установить новый адрес и значение для записи в другую ячейку памяти. И спустя  $T_{adh} = 20$  нс установить YE в единицу и записать следующее слово. Если запись больше не требуется, то спустя время  $T_{rgh} = 20$  нс после очистки бита YE необходимо очистить бит PROG и спустя время  $T_{nvh} = 5$  мкс очистить биты XE и NVSTR. Последующие операции с памятью можно выполнять спустя время  $T_{rcv} = 1$  мкс. Временная диаграмма записи памяти представлена на Рис. 12.

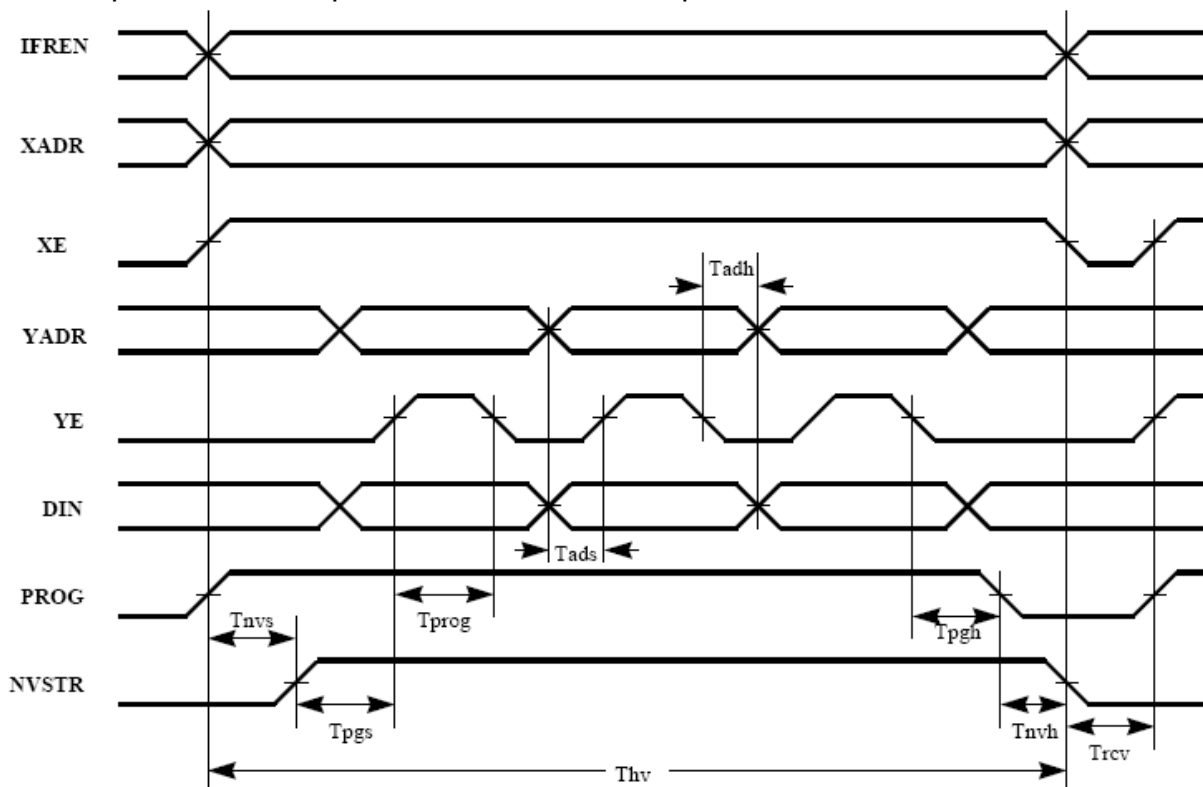
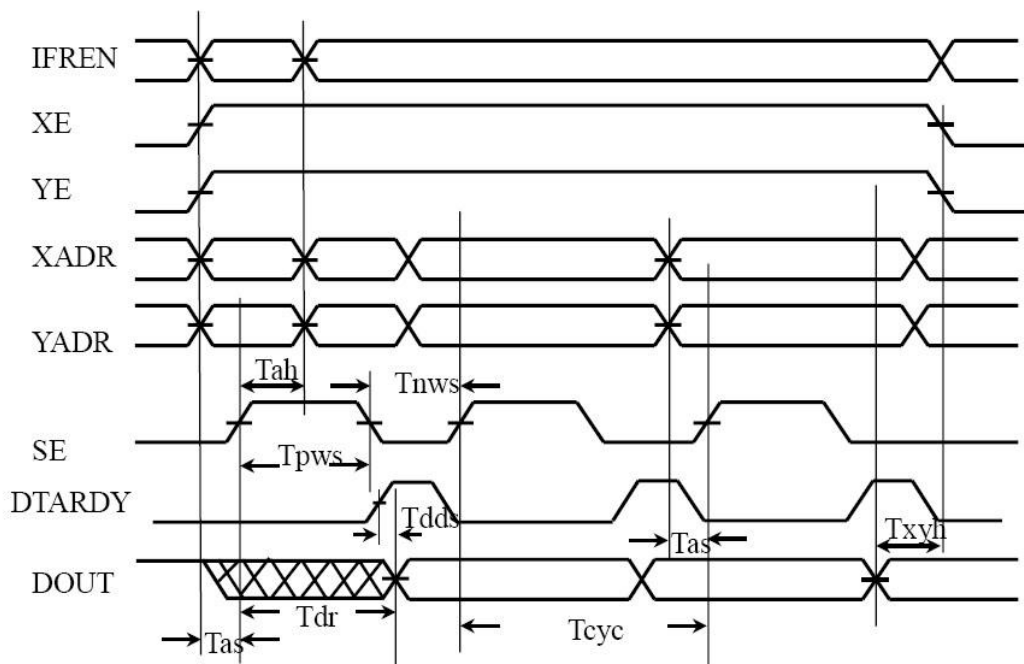


Рис. 12 Временная диаграмма записи памяти

### Чтение 32-х битного слова из памяти.

В обычном режиме работы для чтения доступна только основная память. Для этого необходимо просто считать требуемый адрес памяти. В режиме программирования для чтения доступна и основная и информационная память. Для чтения из памяти надо установить необходимое значение в бит IFREN (1 – для информационной памяти и 0 – для основной памяти), затем установить адрес из которого необходимо считать данные в регистре EEPROM\_ADR и установить биты XE, YE и SE в единицу, и спустя время  $T_{xa} = 55$  нс из регистра EEPROM\_DO можно считать данные. Если необходимо считать следующее слово, то в регистр EEPROM\_ADR необходимо записать новый адрес и спустя время  $T_{xa} = 55$  нс из регистра EEPROM\_DO можно считать следующие данные. Если чтение больше

не требуется, то можно очистить все биты управления. Временная диаграмма чтения памяти представлена на Рис. 13.



**Рис. 13** Временная диаграмма чтения памяти

Flash память программ поддерживает до 20 000 тысяч циклов перезаписи. Нельзя повторять циклы стирания – записи и стирания – стирания одной ячейки памяти с периодом менее 4 мс.

**Описание регистров управления контроллера Flash памяти программ**

**Таблица 13**

Базовый Адрес	Название	Описание
0x4001_8000	EEPROM_CNTRL	Регистры контроллера Flash памяти программ
Смещение		
0x00	EEPROM_CMD	Регистр управления EEPROM памятью
0x04	EEPROM_ADR	Регистр адреса
0x08	EEPROM_DI	Регистр данных на запись
0x0C	EEPROM_DO	Регистр данных считанных
0x10	EEPROM_KEY	Регистр ключа

**EEPROM\_CMD**

Номер	9	8	7	6	5...3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	100	0	0	0
	IFREN	SE	YE	XE	Delay[2:0]	RD	WR	CON

Номер	31		14	13	12	11	10
Доступ	U		U	R/W	R/W	R/W	R/W
Сброс	0		0	0	0	0	0
				NVSTR	PROG	MAS1	ERASE

R/W - бит доступен на чтение и запись;  
 RO - бит доступен только на чтение;  
 U - бит физически не реализован или зарезервирован.

**Таблица 14**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..14	-	Зарезервировано
13	NVSTR	Операции записи или стирания 0 – при чтении 1- при записи или стирании
12	PROG	Записать данные по ADR[16:2] из регистра EEPROM_DI 0 – нет записи 1 – есть запись
11	MAS1	Стереть весь блок, при ERASE =1 0 – нет стирания 1 – стирание
10	ERASE	Стереть строку с адресом ADR[16:9], ADR[8:0] значения не имеет 0 – нет стирания 1 – стирание
9	IFREN	Работа с блоком информации 0 – основная память 1 – информационный блок
8	SE	Усилитель считывания 0 – не включен 1 – включен
7	YE	Выдача адреса ADR[8:2] 0 – не разрешено 1 – разрешено
6	XE	Выдача адреса ADR[16:9] 0 – не разрешено 1 - разрешено
5...3	Delay[2:0]	Задержка памяти программ при чтении в циклах (в рабочем режиме) 000 – 0 цикл 001 – 1 цикл

2	RD	Чтение из памяти EEPROM (в режиме программирования) 0 – нет чтения 1 – есть чтение
1	WR	Запись в память EEPROM (в режиме программирования) 0 – нет записи 1 – есть запись
0	CON	Переключение контроллера памяти EEPROM на регистровое управление, не может производиться при исполнении программы из области EEPROM 0 – управление EEPROM от ядра, рабочий режим 1 – управление от регистров, режим программирования

**EEPROM\_ADR**

Номер	31	0
Доступ	R/W	R/W
Сброс	0	0
		ADR [31:0]

**Таблица 15**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	ADR[31:0]	Адрес обращения в память ADR[1:0] – не имеет значения, минимально адресуемая ячейка 32 бита

**EEPROM\_DI**

Номер	31	0
Доступ	R/W	R/W
Сброс	0	0
		DATA [31:0]

**Таблица 16**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	DATA[31:0]	Данные для записи в EEPROM

**EEPROM\_DO**

Номер	31	0
Доступ	R/W	R/W
Сброс	0	0
		DATA [31:0]



**Таблица 17**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	DATA[31:0]	Данные считанные из EEPROM

**EEPROM\_KEY**

Номер	31	0
Доступ	R/W	R/W
Сброс	0	0
		KEY [31:0]

**Таблица 18**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	DATA[31:0]	Ключ для разрешения доступа к Flash памяти через регистровый доступ. Перед переводом памяти в режим программирования необходимо в регистр EEPROM_KEY записать комбинацию 0x8AAA5551.

**Система команд**

В процессоре реализована версия системы команд Thumb. Поддерживаемые команды представлены в Таблица 19.

В таблице используются следующие обозначения:

- в угловых скобках <> записываются альтернативные формы представления операндов;
- в фигурных скобках {} указываются необязательные операнды;
- информация в столбце "операнды" может быть неполной;
- второй операнд Op2 может быть либо регистром, либо константой;
- большинство команд могут содержать суффикс кода условного выполнения.

Более подробная информация представлена в детальном описании команд.

**Таблица 19.** Система команд процессора Cortex-M0

Мнемокод команды	Операнды	Краткое описание	Флаги	Страница
ADC, ADCS	{Rd,} Rn, Op2	Сложение с переносом	N,Z,C,V	
ADD, ADDS	{Rd,} Rn, Op2	Сложение	N,Z,C,V	
ADR	Rd, label	Загрузка адреса, заданного относительно счетчика команд	-	
AND, ANDS	{Rd,} Rn, Op2	Логическое И	N,Z,C	
ASR, ASRS	Rd, Rn, Op2	Арифметический сдвиг вправо	N,Z,C	
B	label	Переход	-	
BIC, BICS	{Rd,} Rn, Op 2	Сброс битов по маске	N,Z,C	
BKPT	#imm8	Точка останова	-	
BL	label	Переход со связью	-	
BLX	Rm	Косвенный переход со связью	-	
BX	Rm	Косвенный переход	-	
CMN, CMNS	Rn, Op2	Сравнить с противоположным знаком	N,Z,C,V	
CMP, CMPS	Rn, Op2	Сравнить	N,Z,C,V	
CPSID	iflags	Изменить состояние процессора, запретить прерывания	-	
CPSIE	iflags	Изменить состояние процессора, разрешить прерывания	-	
CPY	Rd, Op2	Загрузка	N,Z,C	
DMB	-	Барьер синхронизации доступа к памяти данных	-	
DSB	-	Барьер синхронизации доступа к памяти данных	-	
EOR, EORS	{Rd,} Rn, Op2	Исключающее ИЛИ	N,Z,C	
ISB	-	Барьер синхронизации доступа к инструкциям	-	
LDM	Rn!, reglist	Загрузка множества	-	

Мнемокод команды	Операнды	Краткое описание	Флаги	Страница
		регистров, инкремент после доступа		
LDMFD, LDMIA	Rn!, reglist	Загрузка множества регистров, инкремент после доступа	-	
LDR	Rt, [Rn, #offset]	Загрузка слова в регистр	-	
LDRB	Rt, [Rn, #offset]	Загрузка байта в регистр	-	
LDRH	Rt, [Rn, #offset]	Загрузка полуслова в регистр	-	
LDRSB	Rt, [Rn, #offset]	Загрузка в регистр байта со знаком	-	
LDRSH	Rt, [Rn, #offset]	Загрузка в регистр полуслова со знаком	-	
LSL, LSLS	Rd, Rm, <Rs #n>	Логический сдвиг влево	N,Z,C	
LSR, LSRS	Rd, Rm, <Rs #n>	Логический сдвиг вправо	N,Z,C	
MOV, MOVS	Rd, Op2	Загрузка	N,Z,C	
MRS	Rd, spec_reg	Считать специальный регистр в регистр общего назначения	-	
MSR	spec_reg, Rm	Записать регистр общего назначения в специальный регистр	N,Z,C,V	
MUL, MULS	{Rd,} Rn, Rm	Умножение, 32-разрядный результат	N,Z	
MVN, MVNS	Rd, Op2	Загрузка инверсного значения	N,Z,C	
NEG	{Rd,} Rm	Инвертирование	N,Z,C,V	
NOP	-	Нет операции	-	
ORR, ORRS	{Rd,} Rn, Op2	Логическое ИЛИ	N,Z,C	
POP	reglist	Извлечь регистры из стека	-	
PUSH	reglist	Занести регистры в стек	-	
REV	Rd, Rn	Изменить на обратный порядок байтов в слове	-	
REV16	Rd, Rn	Изменить на обратный порядок байтов в полусловах	-	
REVSH	Rd, Rn	Изменить на обратный порядок байт в младшем полуслове, произвести распространение знакового бита в старшее полуслово	-	
ROR, RORS	Rd, Rm, <Rs #n>	Циклический сдвиг вправо	N,Z,C	
RSB, RSBS	{Rd,} Rn, Op2	Вычитание с противоположным порядком аргументов	N,Z,C,V	
SBC, SBCS	{Rd,} Rn, Op2	Вычитание с учетом переноса	N,Z,C,V	

Мнемокод команды	Операнды	Краткое описание	Флаги	Страница
SEV	-	Установить признак события	-	
STM	Rn!, reglist	Сохранение множества регистров, инкремент после доступа	-	
STMEA	Rn!, reglist	Сохранение множества регистров, инкремент перед доступом	-	
STMIA	Rn!, reglist	Сохранение множества регистров, инкремент после доступа	-	
STR	Rt, [Rn, #offset]	Сохранение регистра	-	
STRB	Rt, [Rn, #offset]	Сохранение регистра, байт	-	
STRH	Rt, [Rn, #offset]	Сохранение регистра, полуслово	-	
SUB, SUBS	{Rd,} Rn, Op2	Вычитание	N,Z,C,V	
SVC	#imm	Вызов супервизора	-	
SXTB	{Rd,}Rm{,ROR#n}	Преобразовать байт со знаком в слово	-	
SXTH	{Rd,}Rm{,ROR#n}	Преобразовать полуслово со знаком в слово	-	
TST	Rn, Op2	Проверка значения битов по маске	N,Z,C	
UXTB	{Rd,}Rm{,ROR#n}	Преобразовать байт без знака в слово	-	
UXTH	{Rd,}Rm{,ROR#n}	Преобразовать полуслово без знака в слово	-	
WFE	-	Ожидать событие	-	
WFI	-	Ожидать прерывание	-	
YIELD	-	Инструкция hint для аппаратного обеспечения при многопоточных задачах	-	

### **Встроенные функции**

Стандарт ANSI языка C не обеспечивает непосредственного доступа к некоторым инструкциям процессора Cortex-M0. В разделе описаны встроенные функции, которые указывают компилятору на необходимость генерации соответствующих инструкций. В случае, если используемый компилятор не поддерживает ту или иную встроенную функцию, рекомендуется включить в текст программы ассемблерную вставку с необходимой инструкцией.

В CMSIS предусмотрены следующие встроенные функции, расширяющие возможности стандарта ANSI C.

**Таблица 20** Встроенные функции CMSIS, позволяющие генерировать некоторые инструкции процессора Cortex-M0

Мнемокод команды процессора	Описание встроенной функции
CPSIE I	void __enable_irq(void)
CPSID I	void __disable_irq(void)
CPSIE F	void __enable_fault_irq(void)
CPSID F	void __disable_fault_irq(void)
ISB	void __ISB(void)
DSB	void __DSB(void)
DMB	void __DMB(void)
REV	uint32_t __REV(uint32_t int value)
REV16	uint32_t __REV16(uint32_t int value)
REVSH	uint32_t __REVSH(uint32_t int value)
SEV	void __SEV(void)
WFE	void __WFE(void)
WFI	void __WFI(void)

Кроме того, CMSIS также обеспечивает возможность чтения и записи специальных регистров процессора, доступных с помощью команд MRS и MSR (см. Таблица 19).

**Таблица 21.** Встроенные функции CMSIS для доступа к специальным регистрам процессора

Наименование специального регистра	Режим доступа	Описание встроенной функции
PRIMASK	Чтение	uint32_t __get_PRIMASK (void)
	Запись	void __set_PRIMASK (uint32_t value)
CONTROL	Чтение	uint32_t __get_CONTROL (void)
	Запись	void __set_CONTROL (uint32_t value)
MSP	Чтение	uint32_t __get_MSP (void)
	Запись	void __set_MSP (uint32_t TopOfMainStack)
PSP	Чтение	uint32_t __get_PSP (void)
	Запись	void __set_PSP (uint32_t TopOfProcStack)
APSR		

### Описание инструкций

В данном разделе представлена подробная информация об инструкциях процессора:

- операнды;
- ограничения на использование счетчика команд PC и указателя стека SP;
- формат второго операнда;
- операции сдвига;
- выравнивание адресов;
- выражения с участием счетчика команд;
- условное исполнение.

### Операнды

В качестве операнда инструкции может выступать регистр, константа, либо другой параметр, специфичный для конкретной команды. Процессор применяет инструкцию к операндам и, как правило, сохраняет результат в регистре-получателе. В случае, если формат команды предусматривает спецификацию регистра-получателя, он, как правило, указывается непосредственно перед операндами.

Операнды в некоторых инструкциях допускают гибкий формат представления, то есть могут быть как регистром, так и константой. Подробнее см. раздел «Формат второго операнда»

### Ограничения на использование PC и SP

Многие инструкции не позволяют использовать регистры счетчика команд (PC) и указателя стека (SP) в качестве регистра-получателя. Подробная информация содержится в описании конкретных инструкций.

Бит [0] адреса, загружаемого в PC с помощью одной из команд процессора BX, BLX, LDM, LDR или POP (см. Таблица 19) должен быть равен 1, так как этот бит указывает на требуемый набор команд, а процессор Cortex-M0 поддерживает только инструкции из набора Thumb.

### Формат второго операнда

Большинство команд обработки данных поддерживает гибкий формат задания второго операнда. Далее в описании синтаксиса инструкций процессора такой операнд будет обозначаться как *Operand2*. При этом в качестве операнда может выступать:

- константа;
- регистр с необязательным параметром сдвига.

### Константа

Данный тип второго операнда задается в формате:

`#constant`

где `constant` может быть:

- любой константой, которая может быть получена путем сдвига восьмиразрядного числа влево на любое количество разрядов в пределах 32-разрядного слова;
- любая константа в виде 0x00XY00XY;
- любая константа в виде 0xXY00XY00;
- любая константа в виде 0xXYXYXYXY.

во всех вышеописанных случаях X и Y представляют шестнадцатеричные цифры.

Кроме того, в небольшом количестве инструкций constant может принимать более широкий диапазон значений. Подробности изложены в описании соответствующих инструкций.

При использовании константного операнда Operand2 в командах MOVS, MVNS, ANDS, ORRS, EORS и TST (см. Таблица 19) в случае, если константа больше 255 и может быть получена путем сдвига восьмиразрядного числа, значение бита [31] константы влияет на значение флага переноса. Для всех остальных значений Operand2 изменения флага переноса не происходит.

### Замена инструкций

В случае, если пользователь указывает константу, не удовлетворяющую требованиям, ассемблер может сгенерировать код с использованием другой инструкции, обеспечивающей необходимую функциональность.

Например, команда CMP Rd, #0xFFFFFFFFE может быть преобразована в эквивалентную команду CMN Rd, #0x2.

### Регистр с необязательным параметром сдвига

В данном случае операнд Operand2 задается в форме:

Rm {, shift}

где:

Rm - регистр, содержащий данные для второго операнда инструкции;

shift - необязательный параметр, определяющий сдвиг данных регистра Rm. Он может принимать одно из следующих значений:

ASR #n - арифметический сдвиг вправо на n бит,  $1 \leq n \leq 32$ .

LSL #n - логический сдвиг влево на n бит,  $1 \leq n \leq 31$ .

LSR #n - логический сдвиг вправо на n бит,  $1 \leq n \leq 32$ .

ROR #n - циклический сдвиг вправо на n бит,  $1 \leq n \leq 31$ .

Случай, если сдвиг не указан, эквивалентен заданию сдвига LSL #0. При этом в качестве операнда используется непосредственно значение регистра Rm без каких-либо дополнительных преобразований.

При указании параметра сдвига в качестве операнда используется преобразованное соответствующим образом 32-разрядное значение регистра Rm, однако содержимое самого регистра Rm не меняется.

Использование операнда со сдвигом в некоторых инструкциях влияет на значение флага переноса. Более подробно действие операций сдвига и их влияние на флаг переноса рассмотрено в разделе «Операции сдвига».

### Операции сдвига

Операции сдвига переносят значение битов содержимого регистра влево или вправо на заданное количество позиций - длина сдвига. Сдвиг может выполняться:

- непосредственно с помощью инструкций ASR, LSR, LSL и ROR (см. Таблица 19), при этом результат сдвига заносится в регистр-получатель;
- во время вычисления значения второго операнда Operand2 команд, при этом результат сдвига используется как один из операндов инструкции.

Допустимая длина сдвига зависит от типа сдвига и инструкции, в которой он был применен. В случае, если этот параметр равен 0, фактически сдвиг не производится. Операции сдвига регистра влияют на значение флага переноса, за исключением случая, когда длина сдвига равна 0. Различные варианты сдвига и их влияние на флаг переноса описаны в следующем подразделе (Rm - сдвигаемый регистр, n - длина сдвига).

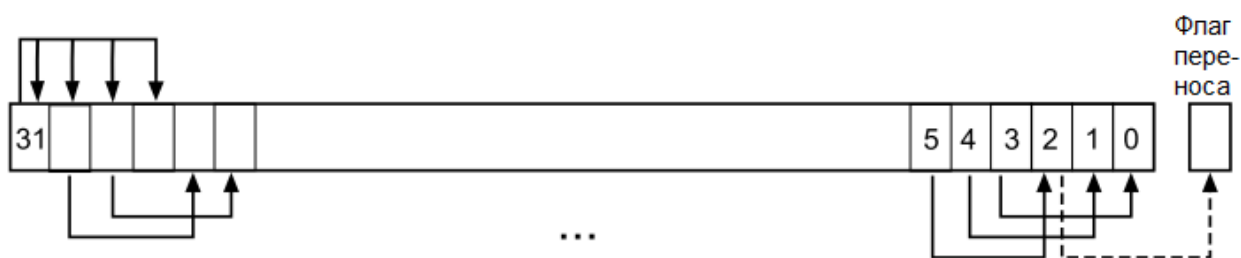
### ASR

Арифметический сдвиг вправо на n бит переносит крайние слева 32-n бит регистра Rm вправо на n позиций, то есть на место крайних справа 32-n. Бит [31] исходного значения регистра записывается в n крайних слева бит результата. См. иллюстрацию на Рис. 14.

Операцию ASR # n можно использовать для деления значения регистра Rm на  $2^n$ , с округлением результата в меньшую сторону (в направлении минус бесконечности).

При использовании инструкции ASRS, а также в случае, если сдвиг ASR #n используется при вычислении второго операнда команд MOVS, MVNS, ANDS, ORRS, EORS или TST (см. Таблица 19) флаг переноса принимает значение последнего бита, вытесненного в результате операции сдвига, то есть бита [n-1] регистра Rm.

В случае, если  $n \geq 32$ , все биты результата устанавливаются в значение бита [31] регистра Rm. Если при этом операция влияет на флаг переноса, то значение этого флага устанавливается равным значению бита [31] регистра Rm.



**Рис. 14** Инструкция ASR #3

### LSR

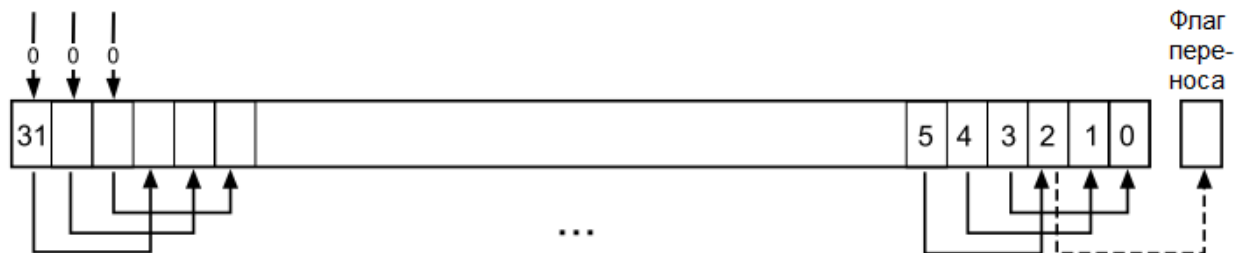
Логический сдвиг вправо на n бит переносит крайние слева 32-n бит регистра Rm вправо на n позиций, то есть на место крайних справа 32-n. При этом в n крайних слева бит результата записывается 0. См. иллюстрацию на Рис. 15.

Операцию LSR # n можно использовать для деления значения регистра Rm на  $2^n$ , в случае, если значение интерпретируется как целое число без знака.



При использовании инструкции LSRS, а также в случае, если сдвиг LSR #n используется при вычислении второго операнда команд MOVBS, MVNS, ANDS, ORRS, EORS или TST флаг переноса принимает значение последнего бита, вытесненного в результате операции сдвига, то есть бита [n-1] регистра Rm.

В случае, если  $n \geq 32$ , все биты результата устанавливаются в 0. Если  $n \geq 33$  и операция влияет на флаг переноса, то значение этого флага устанавливается равным 0.



**Рис. 15** Инструкция LSR # 3

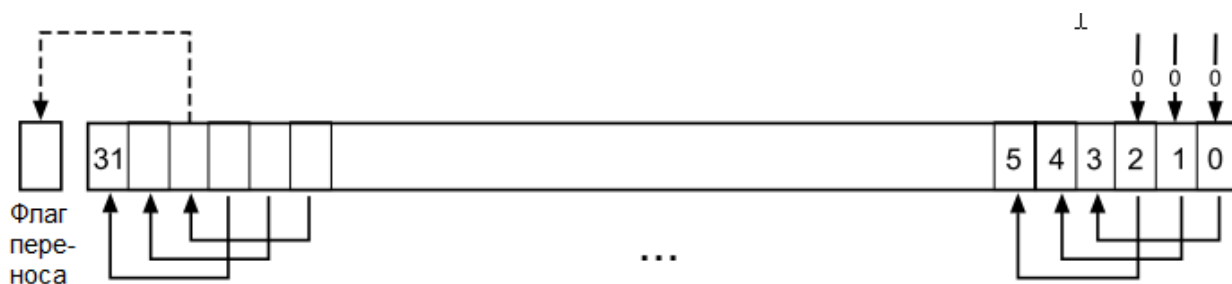
## LSL

Логический сдвиг влево на n бит переносит крайние справа 32-n бит регистра Rm влево на n позиций, то есть на место крайних слева 32-n. При этом в n крайних слева бит результата записывается 0. См. Рис. 16.

Операцию LSL # n можно использовать для умножения значения регистра Rm на  $2^n$ , в случае, если значение интерпретируется как целое число без знака, либо целое число со знаком, записанное в дополнительном коде. Переполнение при выполнении умножения не диагностируется.

При использовании инструкции LSLS, а также в случае, если сдвиг LSL #n используется при вычислении второго операнда команд MOVBS, MVNS, ANDS, ORRS, EORS или TST флаг переноса принимает значение последнего бита, вытесненного в результате операции сдвига, то есть бита [32-n] регистра Rm. Инструкция LSL #0 не влияет на значение флага переноса.

В случае, если  $n \geq 32$ , все биты результата устанавливаются в 0. Если  $n \geq 33$  и операция влияет на флаг переноса, то значение этого флага устанавливается равным 0.



**Рис. 16** Инструкция LSL # 3

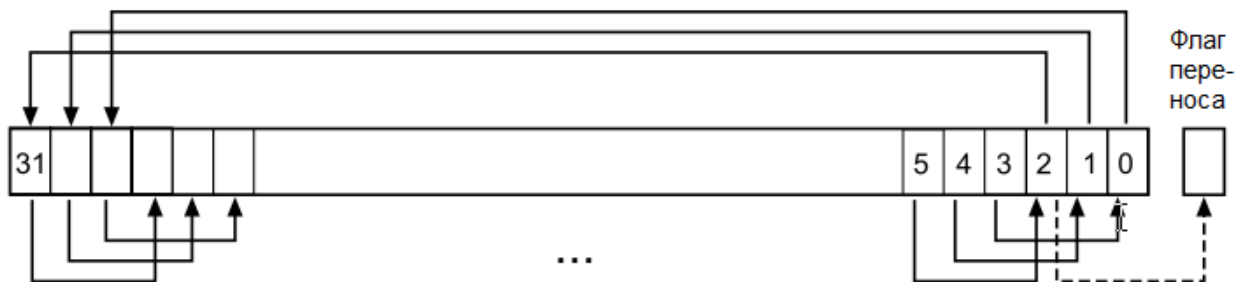
## ROR

Циклический сдвиг вправо на n бит переносит крайние слева 32-n бит регистра Rm вправо на n позиций, то есть на место крайних справа 32-n. При этом n крайних справа разрядов регистра переносятся в крайние n слева разрядов результата. См. Рис. 17.

При использовании инструкции RORS, а также в случае, если сдвиг ROR #n используется при вычислении второго операнда команд MOVVS, MVNS, ANDS, ORRS, EORS или TST флаг переноса принимает значение последнего сдвинутого бита, то есть бита [n-1] регистра Rm.

В случае, если n = 32, результат совпадает с исходным значением регистра. Если n = 32 и операция влияет на флаг переноса, то значение этого флага устанавливается равным биту [31] регистра Rm.

Операция циклического сдвига ROR с параметром, большим 32, эквивалентна циклическому сдвигу с параметром n-32.



**Рис. 17** Инструкция ROR # 3

### **Выравнивание адресов**

Под доступом по выровненным адресам понимаются операции, в которых чтение и запись слов, двойных слов, и более длинных последовательностей слов осуществляется по адресам, выровненным по границе слова, а доступ к полусловам осуществляется по адресам, выровненным по границе полуслова. Чтение и запись байтов гарантированно является выровненным.

Процессор Cortex-M0 поддерживает доступ по невыровненным адресам только для следующих инструкций:

- LDR;
- LDRH;
- LDRSH;
- STR;
- STRH.

Все остальные инструкции при попытке доступа по невыровненному адресу генерируют исключение (Hard fault). Более подробно данный вопрос рассмотрен в разделе «Обработчики исключений».

Невыровненный доступ к данным, как правило, осуществляется медленнее, чем выровненный. Кроме того, некоторые области адресного пространства могут не поддерживать доступ по невыровненному адресу. В связи с этим ARM рекомендует программистам обеспечивать необходимое выравнивание данных. Для того, чтобы избежать ситуаций, в которых невыровненный доступ осуществляется непреднамеренно, следует установить в «1» бит UNALIGN\_TRP регистра конфигурации и управления CCR, что приведет к формированию процессором исключительной ситуации в данной ситуации (см. раздел «Регистр конфигурации и управления»).

### **Адресация относительно счетчика команд PC**

В системе команд Cortex-M0 предусмотрена адресация команды или области данных в виде суммы значения счетчика команд PC плюс/минус численное смещение. Смещение вычисляется ассемблером автоматически исходя из адреса метки и текущего адреса. В случае если смещение слишком велико, диагностируется ошибка.

- для инструкций B, BL текущий адрес определяется как адрес этой инструкции плюс 4 байта;
- для всех остальных инструкций текущий адрес определяется как адрес инструкции плюс 4 байта, при этом бит [1] результата должен быть установлен в 0 для обеспечения выравнивания адреса по границе слова.
- ассемблер может поддерживать расширенные варианты синтаксиса для адресации относительно PC, например "метка плюс/минус число" или выражения типа [PC, #number].

### **Условное исполнение**

Большая часть команд обработки данных способна изменять значения флагов в регистре состояния APSR (специальный регистр процессора, см. Таблица 21) в зависимости от результата выполнения.

Некоторые команды влияют на все флаги, некоторые только на часть. В случае, если инструкция не меняет значение данного флага, сохраняется его старое значение. Более подробно влияние на флаги рассмотрено в описании конкретных инструкций.

Возможность исполнения или неисполнения инструкции, в зависимости от значения флагов, сформированных ранее, может быть достигнута либо за счет использования условных переходов, либо путем добавления суффикса условия исполнения к инструкции. В Таблица 22 представлен список суффиксов, которые можно добавить к инструкции для того, чтобы сделать ее условной.

При наличии одного из указанных суффиксов процессор проверяет значение флагов на соответствие заданному условию. Если условие не выполняется, то инструкция:

- не исполняется;
- не записывает значение операции в регистр-получатель;
- не влияет на флаги;
- не генерирует исключений.

Процессорное ядро поддерживает только одну инструкцию условного перехода B<с> (Branch), где <с> один из суффиксов условного исполнения.

Ниже в разделе рассматриваются:

- флаги условий;
- суффиксы условного исполнения.

### **Флаги условий**

Регистр состояния прикладной программы APSR содержит следующие флаги:

- N=1 в случае, если результат операции меньше нуля, 0 в противном случае.
- Z=1 в случае, если результат равен нулю, 0 в противном случае.
- C=1 в случае, если при выполнении операции возник перенос, 0 в противном случае.

- $V=1$  в случае, если при выполнении операции возникло переполнение, 0 в противном случае.

Перенос возникает в следующих случаях:

- результат сложения оказался больше или равен  $2^{32}$ ;
- результат вычитания больше или равен нулю;
- в результате работы внутренней логики процессора при операциях загрузки данных и логических операций.

Переполнение возникает в случае, если результат сложения, вычитания или сравнения больше или равен  $2^{31}$ , либо меньше  $-2^{31}$ .

Большая часть инструкций меняют значение флагов только в случае, если у них указан суффикс S. Подробную информацию смотрите в описании конкретных команд.

### Суффиксы условного исполнения

В мнемокодах команд, допускающих условное исполнение, предусмотрена возможность указания необязательного кода условия. В описании синтаксиса это обозначается как {cond}.

Если код условия указан, инструкция выполняется только при удовлетворении соответствующему условию флагов регистра APSR. Используемые коды представлены в Таблица 22. Там же указаны соответствующие логические выражения для значений флагов.

Условные команды рекомендуется использовать для снижения количества ветвлений в программе.

**Таблица 22** Суффиксы условного исполнения

Суффикс	Флаги	Значение
EQ	$Z = 1$	Равенство
NE	$Z = 0$	Неравенство
CS или HS	$C = 1$	Больше или равно, беззнаковое сравнение
CC или LO	$C = 0$	Меньше, беззнаковое сравнение
MI	$N = 1$	Меньше нуля
PL	$N = 0$	Больше или равно нулю
VS	$V = 1$	Переполнение
VC	$V = 0$	Нет переполнения
HI	$C = 1$ and $Z = 0$	Больше, беззнаковое сравнение
LS	$C = 0$ or $Z = 1$	Меньше или равно, беззнаковое сравнение
GE	$N = V$	Больше или равно, знаковое сравнение
LT	$N \neq V$	Меньше, знаковое сравнение
GT	$Z = 0$ and $N = V$	Больше, знаковое сравнение
LE	$Z = 1$ and $N \neq V$	Меньше или равно, знаковое сравнение
AL	1	Безусловное исполнение.

**Команды доступа к памяти**

Обобщенные данные о командах доступа к памяти представлены в Таблица 23:

**Таблица 23** Команды доступа к памяти

Мnemonic	Краткое описание	Страница
ADR	Загрузка адреса, заданного относительно счетчика команд	
LDM{mode}	Загрузка множества регистров	
LDR{type}	Загрузка регистра, непосредственно указанное смещение	
LDR{type}	Загрузка регистра, смещение в регистре	
LDR	Загрузка регистра по относительному адресу	
POP	Извлечение регистров из стека	
PUSH	Загрузка регистров в стек	
STM{mode}	Сохранение множества регистров	
STR{type}	Сохранение регистра, непосредственно указанное смещение	
STR{type}	Сохранение регистра, смещение в регистре	

**ADR**

Загрузка адреса, заданного относительно счетчика команд.

## Синтаксис

ADR Rd, label

где:

Rd - регистр-получатель.

label - относительный адрес, см. раздел «Адресация относительно счетчика команд PC».

## Описание

Инструкция ADR вычисляет адрес доступа к памяти путем сложения текущего значения счетчика команд PC и непосредственно заданного смещения, после чего записывает результат в регистр-получатель.

Благодаря использованию относительно адресации код команды не зависит от ее размещения в физической памяти.

При формировании с помощью команды ADR адреса перехода для команд BX или BLX программисту необходимо убедиться, что бит [0] формируемого адреса установлен в 1.

Значения смещения относительно PC должны находиться в пределах 0...1020.

## Ограничения

В качестве регистра Rd нельзя использовать указатель стека SP и счетчик команд PC.

## Флаги

Данная инструкция не влияет на состояние флагов.

## Примеры

ADR R1, TextMessage ; Загрузить адрес позиции, указанный  
; меткой TextMessage, в регистр R1

### **LDR и STR. Непосредственно заданное смещение**

Загрузка или сохранение регистра в режиме адресации со смещением, преиндексированием или постиндексированием.

Синтаксис

op{type} Rt, [Rn {, #offset}]	; адресация со смещением
op{type} Rt, [Rn, #offset]!	; преиндексирование
op{type} Rt, [Rn], #offset	; постиндексирование
opD Rt, Rt2, [Rn {, #offset}]	; адресация со смещением, двойное слово
opD Rt, Rt2, [Rn, #offset]!	; преиндексирование, двойное слово
opD Rt, Rt2, [Rn], #offset	; постиндексирование, двойное слово

где:

op - один из кодов операций:

- LDR загрузить регистр.
- STR сохранить регистр.

type - один из суффиксов размера данных:

- B - байт без знака, при загрузке старшие байты устанавливаются в нуль.
- SB - байт со знаком, при загрузке происходит распространение знакового бита в старшие байты (только LDR).
- H - беззнаковое полуслово, при загрузке старшие байты устанавливаются в нуль.
- SH - полуслово со знаком, при загрузке происходит распространение знакового бита в старшие байты (только LDR).
- без суффикса - 32-разрядное слово.

Rt - регистр, в который должна производиться загрузка или значение которого должно быть сохранено.

Rn - регистр, содержащий базовый адрес памяти.

offset - смещение относительно базового адреса Rn. В случае, если смещение не указано, оно подразумевается равным нулю.

Rt2 - дополнительный регистр, предназначенный для двухсловных операций чтения или записи.

Описание

LDR - загружает один или два регистра значением из памяти.

STR - сохраняет значение одного или двух регистров в память.

Инструкции с непосредственно заданным смещением могут функционировать в одном из следующих режимов адресации:

#### **Адресация со смещением**

Значение смещения добавляется к содержимому регистра Rn или вычитается из него. Результат используется в качестве адреса чтения или записи. Значение регистра Rn остается неизменным.

Синтаксис задания данного режима: [Rn, #offset].

**Адресация с преиндексированием**

Значение смещения добавляется к содержимому регистра Rn или вычитается из него. Результат используется в качестве адреса чтения или записи, а также записывается обратно в регистр Rn.

Синтаксис задания данного режима: [Rn, #offset]!

**Адресация с постиндексированием**

Содержимое регистра Rn используется в качестве адреса чтения или записи. Значение смещения добавляется к содержимому регистра Rn или вычитается из него, после чего записывается обратно в регистр Rn.

Синтаксис задания данного режима: [Rn], #offset .

Загружаемое или сохраняемое значение может быть байтом, полусловом, словом или двойным словом. Байты и полуслова могут интерпретироваться как числа со знаком или без знака. См. раздел «Выравнивание адресов».

В Таблица 24 показаны диапазоны значений смещения для различных форм адресации.

**Таблица 24 Диапазон значений смещения**

Тип инструкции	Смещение	Преиндексирование	постиндексирование
Слово, полуслово, байт	От 0 до 124	от 0 до 124	от 0 до 124
Двойное слово	Значения, кратные 4, в диапазоне от 0 до 1020		

**Ограничения**

Для команд загрузки регистров:

- использовать в качестве Rt регистры PC и SP можно только в командах загрузки слова;
- при загрузке двойных слов регистры Rt и Rt2 не должны совпадать
- в режимах адресации с пре- и постиндексированием регистр Rn не должен совпадать с регистрами Rt или Rt2.

В случае, если в команде загрузки слова в качестве регистра Rt используется счетчик команд PC:

- бит [0] загружаемого значения должен быть равен 1;
- передача управления происходит по адресу, соответствующему значению бита [0] в 0;

Для команд сохранения регистров:

- использовать в качестве Rt регистры SP можно только в командах записи слова;
- в качестве регистров Rt и Rn нельзя использовать счетчик команд PC;
- в режимах адресации с пре- и постиндексированием регистр Rn не должен совпадать с регистрами Rt или Rt2.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

LDR R8, [R10] ; Загрузка регистра R8 из ячейки по адресу,

STR R2, [R9,#const-struct] ; содержащемуся в R10.  
; const-struct - выражение с постоянным значением,  
STRH R3, [R4], #4 ; лежащим в диапазоне 0-124.  
; Записать содержимое R3, интерпретируемое как  
; полуслово, по адресу, содержащемуся в R4,  
после чего  
; увеличить R4 на 4

### LDR и STR. Смещение задано в регистре

Загрузка или сохранение регистра в режиме адресации со смещением, заданным в регистре.

Синтаксис

op{type} Rt, [Rn, Rm {, LSL #n}]

где:

op - один из кодов операций:

- LDR загрузить регистр.
- STR сохранить регистр.

type - один из суффиксов размера данных:

- B - байт без знака, при загрузке старшие байты устанавливаются в нуль.
- SB - байт со знаком, при загрузке происходит распространение знакового бита в старшие байты (только LDR).
- H - беззнаковое полуслово, при загрузке старшие байты устанавливаются в нуль.
- SH - полуслово со знаком, при загрузке происходит распространение знакового бита в старшие байты (только LDR).
- без суффикса - 32-разрядное слово.

Rt - регистр, в который должна производиться загрузка или значение которого должно быть сохранено.

Rn - регистр, содержащий базовый адрес памяти.

Rm - регистр, содержащий смещение относительно базового адреса.

LSL #n - необязательный параметр сдвига, в диапазоне от 0 до 3.

Описание

LDR - загружает регистра значением из памяти.

STR - сохраняет значение регистра в памяти.

Адрес области памяти, в которую будет производиться обращение, вычисляется на основании значения базового адреса в регистре Rn и смещения. Смещение определяется значением регистра Rm и параметром сдвига влево значения этого регистра.

Считываемое или записываемое значение может иметь размер байта, полуслова или слова. При загрузке данных из памяти байты и полуслова могут интерпретироваться либо как числа со знаком, либо как беззнаковые. См. Раздел «Выравнивание адресов».



## Ограничения

Для данных команд:

- Rn не может быть счетчиком команд PC;
- Rm не может быть SP или PC;
- использовать в качестве Rt регистр SP можно только в командах чтения и записи слова;
- использовать в качестве Rt регистр PC можно только в командах чтения слова;

В случае, если в команде загрузки слова в качестве регистра Rt используется счетчик команд PC:

- бит [0] загружаемого значения должен быть равен 1, передача управления при этом осуществляется по выровненному по границе полуслова адресу;

## Флаги

Данная инструкция не влияет на состояние флагов.

## Примеры

STR R0, [R5, R1] ; Записать значение R0 по адресу, равному сумме R5 и R1

LDRSB R0, [R5, R1, LSL #1]

; Читать байт по адресу, равному сумме R5 и R1,  
; умноженному на два, распространить знаковый бит на старшие значащие байты слова, загрузить результат  
; в регистр R0

STR R0, [R1, R2, LSL #2]

; Сохранить значение регистра R0 по адресу, равному  $R1+4*R2$

## **LDR, адресация относительно счетчика команд PC**

Загрузка регистра из памяти.

## Синтаксис

LDR{type}Rt, label

где:

type - один из суффиксов размера данных:

- B - байт без знака, при загрузке старшие байты устанавливаются в нуль.
- SB - байт со знаком, при загрузке происходит распространение знакового бита в старшие байты (только LDR).
- H - беззнаковое полуслово, при загрузке старшие байты устанавливаются в нуль.
- SH - полуслово со знаком, при загрузке происходит распространение знакового бита в старшие байты (только LDR).
- без суффикса - 32-разрядное слово.

Rt - регистр, в который должна производиться загрузка.

Rt2 - второй регистр, в который должна производиться загрузка.

label - относительный адрес, раздел «Адресация относительно счетчика команд PC».

### Описание

LDR - загружает регистра значением из памяти с адресом, заданным в виде метки, относительно счетчика команд PC.

Считываемое значение может иметь размер байта, полуслова или слова. При загрузке данных из памяти байты и полуслова могут интерпретироваться либо как числа со знаком, либо как беззнаковые. См. раздел «Выравнивание адресов».

Метка должна располагаться на ограниченном расстоянии от текущей инструкции. В Таблица 25 показаны возможные значения смещений между меткой данных и текущим значением счетчика команд.

Таблица 25 Диапазон значений смещения

Тип инструкции	Диапазон значений смещения
Слово, полуслово со знаком или без знака, байт со знаком или без знака	от 0 до 124
Двойное слово	от 0 до 1020

### Ограничения

В данной инструкции:

- использовать в качестве Rt регистры PC или SP можно только в командах чтения слова;
- нельзя использовать в качестве Rt2 регистры PC и SP;
- при загрузке двойных слов регистры Rt и Rt2 не должны совпадать.

В случае, если в команде загрузки слова в качестве регистра Rt используется счетчик команд PC:

- бит [0] загружаемого значения должен быть равен 1, передача управления при этом осуществляется по выровненному по границе полуслова адресу;

### Флаги

Данная инструкция не влияет на состояние флагов.

### Примеры

LDR R0, LookUpTable ; Загрузить R0 словом данных по адресу  
; с меткой LookUpTable  
LDRSB R7, localdata ; Загрузить байт данных по адресу с меткой  
localdata,  
; распространить значение знакового бита в  
старшие  
; байты слова данных, сохранить результат в R7

### LDM и STM

Загрузка или сохранение множества регистров.

### Синтаксис

op{addr\_mode} Rn!, reglist

где:

op - один из кодов операций:

- LDM загрузить множество регистров.
- STM сохранить множество регистров.

addr\_mode - один из режимов адресации:

- IA - с увеличением адреса после каждого доступа. Этот режим используется по умолчанию.
- EA - с увеличением адреса после каждого доступа. (только для STM)
- FD - с увеличением адреса после каждого доступа. (только для LDM)

Rn - регистр, содержащий базовый адрес памяти.

! - обязательный суффикс обратной записи значения базового регистра. В случае, если он присутствует в команде, последний адрес, по которому осуществлялся доступ, будет записан обратно в регистр Rn.

reglist - заключенный в фигурные скобки список из одного или нескольких регистров, которые должны быть записаны или считаны. В списке можно указывать диапазон номеров регистров. Начальный и конечный регистр диапазона разделены знаком "-". Элементы списка (отдельные регистры или диапазоны) разделяются запятыми. См. «Примеры» ниже.

Мнемокод LDMFD и LDMIA (см. Таблица 19) - это псевдокоманды LDM. Использование команды LDMFD обусловлено извлечением данных из полного нисходящего стека, с указателем на последний загруженный элемент (Full Descending stack).

Мнемокод STMEA и STMIA - это псевдокоманды STM. Использование команды STMEA обусловлено сохранением данных в пустой восходящий стек, с указателем на последнюю свободную ячейку (Empty Ascending stack).

### Описание

Инструкции LDM осуществляют загрузку регистров из списка reglist значениями слов данных из памяти с базовым адресом, содержащимся в регистре Rn.

Инструкции STM осуществляют сохранение слов данных, содержащихся в регистрах из списка reglist, в память с базовым адресом, содержащимся в регистре Rn.

Команды LDM, LDMIA, LDMFD, STM, STMIA и STMEA для доступа используют адреса памяти в интервале от Rn до Rn+4\*(n-1), где n - количество регистров в списке reglist. Доступ осуществляется в порядке увеличения номера регистра, при этом регистр с наименьшим номером соответствует наименьшему адресу памяти, а регистр с наибольшим номером - наибольшему адресу. Значение Rn+4\*(n-1) записывается обратно в регистр Rn.

### Ограничения

В описываемых в разделе командах:

- в качестве регистра Rn нельзя использовать счетчик команд PC;
- список регистров reglist не может содержать указатель стека SP;
- в любой инструкции STM в списке регистров reglist нельзя указывать PC;
- в любой инструкции LDM в reglist нельзя указывать одновременно PC и LR;

В случае, если инструкция LDM содержит в списке reglist счетчик команд PC:

- бит [0] загружаемого значения должен быть равен 1, передача управления при этом осуществляется по выровненному по границе полуслова адресу;

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

LDMFD R8!,{R0,R2,R9} ; LDMFD - синоним LDM

STMEA R1!,{R3-R6,R11,R12}; STMEA – синоним STM

Примеры неправильного использования

STM R5!,{R5,R4,R9} ; Сохраненное значение R5 является непредсказуемым

LDM R2!, {} ; Список должен содержать хотя бы один регистр

### **PUSH и POP**

Загружает или считывает регистры в стек или из стека, растущего вниз, с указателем на последний загруженный элемент (full-descending stack).

Синтаксис

PUSH reglist

POP reglist

где:

reglist - заключенный в фигурные скобки список из одного или нескольких регистров, которые должны быть записаны или считаны. В списке можно указывать диапазон номеров регистров. Начальный и конечный регистр диапазона разделены знаком "-". Элементы списка (отдельные регистры или диапазоны) разделяются запятыми.

Команды PUSH и POP являются синонимами команд STM и LDM в которых базовый адрес памяти содержится в регистре указателя стека SP, а режим записи обратной записи значения базового регистра включен.

Мнемокоды PUSH и POP являются предпочтительными вариантами записи.

Описание

PUSH - сохраняет регистры в стеке в порядке уменьшения номеров регистров, при этом регистр с большим номером сохраняется в память с большим значением адреса.

POP - восстанавливает значения регистров из стека в порядке увеличения номеров регистров, при этом регистр с меньшим номером считывается из памяти с меньшим значением адресом.

Ограничения

В данных инструкциях:

- список регистров reglist не должен содержать указатель стека SP;
- в инструкции PUSH список регистров не должен содержать счетчик команд PC;
- в инструкции POP список регистров не должен одновременно содержать регистры PC и LR.

В случае, если инструкция POP содержит в списке reglist счетчик команд PC:

- бит [0] загружаемого значения должен быть равен 1, передача управления при этом осуществляется по выровненному по границе полуслова адресу;

### Флаги

Данная инструкция не влияет на состояние флагов.

### Примеры

PUSH {R0,R4-R7}

PUSH {R2,LR}

POP {R0,R10,PC}

### Инструкции обработки данных

В Таблица 26 представлены инструкции обработки данных:

**Таблица 26** Команды обработки данных

Мнемокод	Краткое описание	Страница
ADC	Сложение с учетом переноса	
ADD	Сложение	
AND	Логическое И	
ASR	Арифметический сдвиг вправо	
BIC	Сброс битов по маске	
CMN	Сравнить с противоположным знаком	
CMP	Сравнить	
EOR	Исключающее ИЛИ	
LSL	Логический сдвиг влево	
LSR	Логический сдвиг вправо	
MOV	Загрузка	
MVN	Загрузка инверсного значения	
ORR	Логическое ИЛИ	
REV	Изменить на обратный порядок байтов в слове	
REV16	Изменить на обратный порядок байтов в полусловах	
REVSH	Изменить на обратный порядок байт в младшем полуслове, произвести распространение знакового бита в старшее полуслово	
ROR	Циклический сдвиг вправо	
RSB	Вычитание с противоположным порядком аргументов	
SBC	Вычитание с учетом переноса	
SUB	Вычитание	
TST	Проверка значения битов по маске	

### ADD, ADC, SUB, SBC и RSB

Сложение, сложение с переносом, вычитание, вычитание с переносом, вычитание с противоположным порядком аргументов.

### Синтаксис

op{S} {Rd,} Rn, Operand2

op {Rd,} Rn, #imm8 ; только для команд ADD, SUB и RSB

где:

op - один из кодов операции:

- ADD - сложение.
- ADC - сложение с учетом переноса.
- SUB - вычитание.
- SBC - вычитание с учетом переноса.
- RSB - вычитание с противоположным порядком аргументов.

S - необязательный суффикс. Если он указан, результат выполнения операции приводит к установке соответствующих флагов, см. раздел «Условное исполнение».

Rd - регистр-получатель результата. В случае, если регистр Rd не указан, результат записывается в Rn.

Rn - регистр, содержащий значение первого операнда.

Operand2 - второй операнд. См. раздел «Формат второго операнда».

imm8 - любое число в диапазоне от 0 до 1020 (Thumb) или 0-508 (Thumb-2) (для RSB только 0)

Описание

Команда ADD складывает значение Operand2 или imm8 со значением регистра Rn.

Команда ADC складывает вместе значения Rn и Operand2, а также флага переноса.

Команда SUB вычитает значение Operand2 или imm8 из значения регистра Rn.

Команда SBC вычитает значение Operand2 из значения регистра Rn. Если флаг переноса не установлен, результат дополнительно уменьшается на единицу.

Команда RSB вычитает значение регистра Rn из значения Operand2. Этот вариант команды полезен, так как существует широкий выбор вариантов построения Operand2.

Инструкции ADC и SBC полезны при реализации вычислений с повышенной разрядностью, см. раздел «Арифметика с повышенной разрядностью» ниже. См. также описание команды ADR.

Ограничения

Для рассматриваемых инструкций:

- в качестве Operand2 нельзя использовать SP или PC;
- использовать SP в качестве регистра Rd допустимо только в командах ADD и SUB, со следующими дополнительными ограничениями:
  - в качестве Rn также должен использоваться SP;
  - сдвиг в Operand2 должен быть не более 3 бит в режиме LSL;
- указатель стека SP может использоваться в качестве Rn только в командах ADD и SUB;
- счетчик команд PC может использоваться в качестве Rd только в команде: ADD PC, PC, Rn причем:
  - не допускается использование суффикса S;
  - в качестве Rn не допускается использовать PC и SP;
  - если инструкция условная, то она должна быть последней в IT-блоке.

- не считая команды ADD PC, PC, Rm в качестве регистра Rn можно использовать счетчик команд PC только в инструкциях ADD и SUB с дополнительными ограничениями:
  - не допускается использование суффикса S;
  - второй операнд должен находиться в интервале от 0 до 1020.
  - при использовании PC в операциях сложения или вычитания биты [1:0] счетчика команд округляются до 0b00 перед выполнением операции, обеспечивая выравнивание адреса по границе слова.
  - при необходимости сформировать адрес инструкции, необходимо скорректировать значение смещения относительно PC. ARM рекомендует использовать вместо инструкцию ADR, так как в этом ассемблер автоматически сгенерирует правильное смещение.
  - в случае, если PC используется в качестве Rd в команде ADD PC, PC, Rm бит[0] значения, записываемого в PC, будет проигнорирован, передача управления будет осуществляться по адресу, соответствующему нулевому значению этого бита.

### Флаги

В случае, если в команде указан суффикс S, процессор устанавливает флаги N, Z, C и V в соответствии с результатом выполнения операции.

### Примеры

ADD R2, R1, R3

SUBS R8, R6, #240 ; установить флаги по результату операции вычитания

RSB R4, R4, #0 ; вычесть содержимое регистра R4 из 0

### Арифметика с повышенной разрядностью

#### 64-разрядное сложение

Следующий пример показывает, как осуществить сложение 64-разрядного целого числа, записанного в паре регистров R2 и R3, с другим 64-разрядным числом, записанным в паре регистров R0 и R1, после чего записывает результат в пару регистров R4 и R5.

ADDS R4, R0, R2 ; сложить младшие значащие слова

ADC R5, R1, R3 ; сложить старшие значащие слова с учетом переноса

#### 96-разрядное вычитание

Данные с повышенной разрядностью не обязательно содержать в смежных регистрах. В примере, приведенном ниже, показан фрагмент кода, осуществляющий вычитание 96-разрядного целого числа, записанного в регистрах R9, R1 и R11, из другого числа, содержащегося в R6, R2 и R8. Результат записывается в регистрах R6, R9 и R2.

SUBS R6, R6, R9 ; вычитание младших значащих слов

SBCS R9, R2, R1 ; вычитание средних значащих слов с переносом

SBC R2, R8, R11 ; вычитание старших значащих слов с переносом

### AND, ORR, EOR, BIC

Логические операции И, ИЛИ, Исключающее ИЛИ, сброс битов по маске.

### Синтаксис

op{S}{Rd,} Rn, Operand2

где:

op - один из кодов операции:

- AND - логическое И.
- ORR - логическое ИЛИ.
- EOR - логическое Исключающее ИЛИ.
- BIC - сброс битов по маске.

S - необязательный суффикс. Если он указан, результат выполнения операции приводит к установке соответствующих флагов, см. раздел «Условное исполнение».

Rd - регистр-получатель результата.

Rn - регистр, содержащий значение первого операнда.

Operand2 - второй операнд. См. раздел «Формат второго операнда».

### Описание

Инструкции AND, ORR и EOR осуществляют, соответственно, логические операции И, ИЛИ и исключающего ИЛИ между аргументами, содержащимися в регистре Rn и вторым операндом Operand2.

Инструкция BIC выполняет операцию логического И между аргументами, содержащимися в регистре Rn и инверсным значением второго операнда Operand2.

### Ограничения

Не допускается использованием указателя стека SP и счетчика команд PC.

### Флаги

В случае, если в команде указан суффикс S, процессор:

- устанавливает флаги N и Z в соответствии с результатом выполнения операции;
- может изменить флаг C в ходе вычисления значения второго операнда, см. раздел «Формат второго операнда»;
- не влияет на значение флага V.

### Примеры

AND R9, R2, #0xFF00

ANDS R9, R8, #0x19

EORS R7, R11, #0x18181818

BIC R0, R1, #0xab

## ASR, LSL, LSR, ROR

Арифметический сдвиг вправо, логический сдвиг влево, логический сдвиг вправо, циклический сдвиг вправо и циклический сдвиг вправо с переносом.

### Синтаксис

op{S} Rd, Rm, Rs



op{S} Rd, Rm, #n

где:

op - один из кодов операции:

- ASR - арифметический сдвиг вправо.
- LSL - логический сдвиг влево.
- LSR - логический сдвиг вправо.
- ROR - циклический сдвиг вправо.

S - необязательный суффикс. Если он указан, результат выполнения операции приводит к установке соответствующих флагов, см. раздел «Условное исполнение».

Rd - регистр-получатель результата.

Rm - регистр, значение которого должно быть подвергнуто сдвигу.

Rs - регистр, содержащий параметр сдвига. Процессор анализирует только младший значащий байт регистра, таким образом, параметр сдвига может принимать значения от 0 до 255.

n - параметр сдвига. Диапазон допустимых значений параметра зависит от инструкции:

- ASR - от 1 до 32;
- LSL - от 0 до 31;
- LSR - от 1 до 32;
- ROR - от 1 до 31.

Команду

LSL{S} Rd, Rm, #0

рекомендуется записывать в формате

MOV{S} Rd, Rm.

Описание

Команда ASR, LSL, LSR и ROR сдвигает биты регистра Rm влево или вправо на заданное количество позиций, определяемое константой n или содержимым регистра Rs.

Во всех указанных инструкциях результат записывается в регистр Rd, при этом содержание регистра Rm остается неизменным. Детальное описание операций сдвига представлено в разделе «Операции сдвига».

Ограничения

Не допускается использованием указателя стека SP и счетчика команд PC.

Флаги

В случае, если в команде указан суффикс S, процессор:

- устанавливает флаги N и Z в соответствии с результатом выполнения операции;
- флаг C устанавливается в значение последнего сдвинутого бита, за исключением случая параметра сдвига, равного нулю. См. раздел «Операции сдвига».

Примеры

ASR R7, R8, #9 ; Арифметический сдвиг вправо на 9 бит

LSLS R1, R2, #3 ; Логический сдвиг влево на 3 бита с установкой флагов

LSR R4, R5, #6 ; Логический сдвиг вправо на 6 бит  
ROR R4, R5, R6 ; Циклический сдвиг вправо на количество бит, указанное  
; в младшем байте регистра R6

### **CMP и CMN**

Сравнение и сравнение с противоположным знаком.

#### Синтаксис

CMP Rn, Operand2  
CMN Rn, Operand2

где:

Rn - регистр, содержащий первый операнд.  
Operand2 - второй операнд. См. раздел «Формат второго операнда».

#### Описание

Данные инструкции осуществляют сравнение значений регистра и второго операнда. По результатам сравнения устанавливаются соответствующие флаги, однако сам результат в регистр не записывается.

Команда CMP вычитает из регистра Rn значение второго операнда Operand2. Она аналогична инструкции SUBS, за исключением того, что не сохраняет результат вычитания.

Команда CMN складывает значения регистра Rn и второго операнда Operand2. Она аналогична инструкции ADDS, за исключением того, что не сохраняет результат вычитания.

#### Ограничения

В данных инструкциях:

- не допускается использованием PC;
- в качестве второго операнда Operand2 нельзя использовать SP.

#### Флаги

Процессор устанавливает флаги N, Z, C и V в соответствии с результатом сравнения.

#### Примеры

CMP R2, R9  
CMN R0, #6400

### **MOV и MVN**

Загрузка в регистр прямого или инверсного значения второго операнда.

#### Синтаксис

MOV{S} Rd, Operand2  
MOV Rd, #imm8  
MVN{S} Rd, Operand2

где:

S - необязательный суффикс. Если он указан, результат выполнения операции приводит к установке соответствующих флагов, см. раздел «Условное исполнение».

Rd - регистр-получатель результата.

Operand2 - второй операнд. См. раздел «Формат второго операнда».

imm8 - любое значение в диапазоне от 0 до 255.

Инструкция MVN считывает значение второго операнда Operand2, производит его побитную инверсию, после чего помещает результат в регистр Rd.

### Ограничения

Регистры SP и PC допускается использовать в исключительно совместно с инструкцией MOV, при следующих ограничениях:

- второй операнд должен быть регистром без указания параметра сдвига;
- суффикс S не должен быть указан.

В случае, если в качестве Rd используется счетчик команд PC:

- бит [0] значения, загружаемого в PC, игнорируется;
- передача управления осуществляется по адресу, соответствующему загруженному значению с битом [0], принудительно установленным в 0.

### Флаги

В случае, если в команде указан суффикс S, процессор:

- устанавливает флаги N и Z в соответствии с результатом выполнения операции;
- может изменить флаг C в ходе вычисления значения второго операнда, см. раздел «Формат второго операнда»;
- не влияет на значение флага V.

### Примеры

MOVS R11, #0x000B; Записать значение 0x000B в R11, флаги устанавливаются

MOVS R10, R12 ; Записать регистр R12 в R10, флаги устанавливаются

MOVS R3, #23 ; Записать значение 23 в R3

MOVS R8, SP ; Записать значение указателя стека в регистр R8

MVNS R2, #0xF ; Записать значение 0xFFFFFFFF (инверсия значения 0x0F)

; в регистр R2, установить флаги

## REV, REV16, REVSH

Изменение порядка битов или байтов в слове.

### Синтаксис

ор Rd, Rn

где:

ор - один из кодов операции:

- REV - изменить на обратный порядок байтов в слове;
- REV16 - изменить на обратный порядок байтов в полусловах;

- REVSН - изменить на обратный порядок байт в младшем полуслове, произвести распространение знакового бита в старшее полуслово.

Rd - регистр-получатель результата.

Rn - регистр, содержащий операнд.

### Описание

Инструкции предназначены для изменения формата представления (endianness) данных:

- REV - преобразует 32-разрядное число в формате big-endian в число в формате little-endian и наоборот.
- REV16 - преобразует 32-разрядное число в формате big-endian в число в формате little-endian и наоборот.
- REVSН - выполняет одно из следующих преобразований:
  - 16-разрядное число со знаком в формате big-endian в 32-разрядное число со знаком в формате little-endian;
  - 16-разрядное число со знаком в формате little-endian в 32-bit 32-разрядное число со знаком в формате big-endian.

### Ограничения

Нельзя использовать указатель стека SP и счетчик команд PC.

### Флаги

Данная инструкция не влияет на состояние флагов.

### Примеры

REV R3, R7 ; Обратить порядок следования байтов в R7, записать в R3

REV16 R0, R0 ; Обратить порядок байтов в каждом 16-битном полуслове R0

REVSН R0, R5 ; Обратить полуслово со знаком

REVHS R3, R7 ; Обратить порядок при условии "больше или равно" (HS)

## TST

Проверить значение битов по маске, проверить равенство.

### Синтаксис

TST Rn, Operand2

где:

Rn - регистр, содержащий первый операнд.

Operand2 - второй операнд. См. раздел «Формат второго операнда».

### Описание

Данные инструкции позволяют проверить значение регистра с учетом значения второго операнда Operand2. По результату устанавливаются флаги, сам результат не сохраняется.

Команда TST выполняет побитовую операцию логического И между значениями Rn и Operand2. Она совпадает с инструкцией ANDS, за исключением того, что не сохраняет результат.

### Ограничения

Нельзя использовать указатель стека SP и счетчик команд PC.

### Флаги

В случае если в команде указан суффикс S, процессор:

- устанавливает флаги N и Z в соответствии с результатом выполнения операции;
- может изменить флаг C в ходе вычисления значения второго операнда, см. раздел «Формат второго операнда»;
- не влияет на значение флага V.

### Примеры

TST R0, R1 ; Побитовое И между R0 и R1  
; устанавливаются флаги, результат не сохраняется

### Инструкция умножения

В Таблица 27 представлена информация о команде умножения:

Таблица 27 Инструкции умножения и деления

Мнемокод	Краткое описание	Страница
MUL	Умножение, 32-разрядный результат	

## MUL

Умножение или умножение с накоплением (сложением, вычитанием) с использованием 32-разрядных операндов и выдающее 32-разрядный результат.

### Синтаксис

MUL{S} {Rd,} Rn, Rm ; Умножение

где:

S - необязательный суффикс. Если он указан, результат выполнения операции приводит к установке соответствующих флагов, см. раздел «Условное исполнение».

Rd - регистр-получатель результата. Если регистр Rd не указан, то в качестве получателя используется регистр Rn.

Rn, Rm - регистры, содержащий значения первого и второго сомножителей.

Ra - регистр, содержащий значение, к которому должно быть прибавлено или вычтено произведение.

### Описание

Команда MUL выполняет перемножение значений, содержащихся в регистрах Rn и Rm, после чего сохраняет 32 младших значащих бита произведения в Rd.

Результат выполнения операций не зависит от того, используются ли в качестве операндов числа со знаком или без знака.

### Ограничения

Нельзя использовать указатель стека SP и счетчик команд PC.

В случае, если инструкция MUL используется с суффиксом установки флагов S:

- регистры Rd, Rn и Rm должны находиться в диапазоне от R0 до R7;
- регистр Rd должен быть тем же, что и Rm;
- не допускается использование суффикса условного исполнения cond.

### Флаги

В случае, если в команде указан суффикс S, процессор:

- устанавливает флаги N и Z в соответствии с результатом выполнения операции;
- не влияет на значение флагов C и V.

### Примеры

MUL R10, R2, R5 ; R10 = R2 x R5  
MULS R0, R2, R2 ; R0 = R2 x R2, установить флаги

### Команды работы с битовыми полями

В Таблица 28 показаны инструкции, позволяющие манипулировать последовательностями смежных бит данных в регистрах или битовых полях:

Таблица 28 Инструкции упаковки и распаковки данных

Мнемокод команд	Краткое описание	Страница
SXTB	Преобразовать байт со знаком в слово	
SXTH	Преобразовать полуслово со знаком в слово	
UXTB	Преобразовать байт без знака в слово	
UXTH	Преобразовать полуслово без знака в слово	

### SXT и UXT

Преобразование байта или полуслова в слово с распространением знакового бита или нулей в старшие значащие разряды.

### Синтаксис

SXTextend Rd, Rm  
UXTextend Rd, Rm

где:

Суффикс *extend* может принимать одно из следующих значений:

- B - преобразование 8-битного числа в 32-битное.
- H - преобразование 16-битного числа в 32-битное.

Rd - регистр-получатель результата.

Rm - регистр-источник данных.

**Описание**

Команда SXTB младшие восемь бит [7:0] регистра Rm, преобразует в 32-разрядное число со знаком путем копирования знакового разряда [7] в биты [31:8], сохраняет результат в регистре Rd.

Команда UXTB младшие восемь бит [7:0] регистра Rm, преобразует в 32-разрядное число без знака путем копирования нуля в биты [31:8], сохраняет результат в регистре Rd.

Команда SXTH младшие шестнадцать бит [15:0] регистра Rm, преобразует в 32-разрядное число со знаком путем копирования знакового разряда [15] в биты [31:16], сохраняет результат в регистре Rd.

Команда UXTH младшие шестнадцать бит [15:0] регистра Rm, преобразует в 32-разрядное число без знака путем копирования нуля в биты [31:16], сохраняет результат в регистре Rd.

**Ограничения**

Нельзя использовать указатель стека SP и счетчик команд PC.

**Флаги**

Данная инструкция не влияет на состояние флагов.

**Примеры**

SXTH R4, R6 ; младшее полуслово из R6,  
 ; преобразовать в 32-разрядное  
 ; число с распространением знака, записать в R4  
 UXTB R3, R10 ; младший байт из R10, преобразовать в  
 ; 32-разрядное число, старшие байты заполнить  
 нулями ; записать результат в R3

**Инструкции передачи управления**

В Таблица 29 представлен список инструкций передачи управления.

**Таблица 29** Инструкции передачи управления

Мнемокод команды	Краткое описание	Страница
B	Переход	
BL	Переход со связью	
BLX	Косвенный переход со связью	
BX	Косвенный переход	

**B, BL, BX и BLX**

Команды ветвления.

**Синтаксис**

B {cond} label  
 BL label  
 BX Rm

BLX Rm

где:

- B - переход по непосредственно заданному адресу.
- BL - переход со связью по непосредственно заданному адресу.
- BX - косвенный переход по адресу, заданному значением регистра.
- BLX - косвенный переход со связью.

cond - необязательный код условия, см. раздел «Условное исполнение».

label - относительный адрес, см. раздел «Адресация относительно счетчика команд PC».

Rm - регистр, содержащий адрес, на который необходимо передать управления. Бит [0] этого регистра должен быть установлен в 1, однако передача управления будет выполнена по адресу, соответствующему значению бита [0], равному 0.

Описание

Все рассматриваемые в данном разделе инструкции осуществляют передачу управления на адрес, заданный меткой либо содержащийся в регистре Rm. кроме того:

- команды BL и BLX записывают адрес следующей инструкции в регистр связи LR (R14).
- команды BX и BLX формируют отказ (Hard fault) в случае, если bit[0] регистра Rm равен 0.

В Таблица 30 представлен диапазон адресуемых переходов для различных команд ветвления.

**Таблица 30** Диапазон адресуемых переходов для команд ветвления

Инструкция	Диапазон адресации
B {cond} label	от -1 МБайт до +1 МБайт относительно текущей позиции
BL label	от -16 МБайт до +16 МБайт относительно текущей позиции
BX Rm	любое значение, записанное в регистре
BLX Rm	любое значение, записанное в регистре

Ограничения

- в команде BLX не допускается использование регистра PC;
- в командах BX и BLX, бит [0] регистра Rm должен быть установлен в 1, при этом передача управления будет, тем не менее, осуществлена по адресу, соответствующему значению бита [0], равного 0;
- B {cond} - единственная условно исполняемая команда
- команды BLX и BX выполняется только в режиме Thumb. При попытке изменить режим при выполнении инструкции возникает исключение Hard Fault

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

- B loopA ; передача управления на метку loopA
- BLE ng ; условная передача управления на метку ng



BEQ target ; условный переход на метку target  
 BL funC ; переход со связью (вызов функции) funC, адрес возврата  
 будет ; записан в регистре LR  
 BX LR ; возврат из функции  
 BLX R0 ; переход со связью (вызов функции) по адресу, записанному в  
 R0

**Прочие инструкции**

В Таблица 31 представлен список не рассмотренных в предыдущих разделах инструкций процессора Cortex-M0:

**Таблица 31** Прочие инструкции

Мнемокод команды	Краткое описание	Страница
BKPT	Точка останова	
CPSID	Изменить состояние процессора, запретить прерывания	
CPSIE	Изменить состояние процессора, разрешить прерывания	
CPY	Аналогична MOV (см. Таблица 19)	
DMB	Барьер синхронизации доступа к памяти данных	
DSB	Барьер синхронизации доступа к памяти данных	
ISB	Барьер синхронизации доступа к инструкциям	
MRS	Загрузка из специального регистра в регистр общего назначения	
MSR	Записать регистр общего назначения в специальный регистр	
NOP	Нет операции	
SEV	Установить признак события	
SVC	Вызов супервизора	
WFE	Ожидать событие	
WFI	Ожидать прерывание	
YIELD	Аналогична инструкции hint (подсказка). Применяется в мультипоточковых приложениях	

**CPS**

Изменить состояние процессора.

Синтаксис

*CPS*effect iflags

где:

effect - один из возможных суффиксов:

- IE - сбрасывает специальный регистр в 0.

- ID - устанавливает специальный регистр в 1.

iflags - последовательность флагов:

- i - сбрасывает или устанавливает регистр PRIMASK (см. Таблица 21).

Описание

Команда CPS позволяет изменить значение специального регистра PRIMASK.

Ограничения

- команда CPS доступна только из привилегированного приложения, при вызове из непривилегированного приложения она игнорируется;

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

CPSID i ; Запретить прерывания и конфигурируемые обработчики отказов

CPSIE i ; Разрешить прерывания и конфигурируемые обработчики отказов

### **DMB**

Барьер синхронизации доступа к памяти данных.

Синтаксис

DMB

Описание

Команда DMB выполняет функцию барьерной синхронизации доступа к памяти данных. Она гарантирует, что все явные (explicit) операции доступа к памяти, которые были инициированы перед выполнением инструкции DMB, будут завершены до того, как начнется выполнение любой операции доступа к памяти после этой инструкции.

Команда DMB не влияет на очередность и порядок выполнения инструкций, не выполняющих доступа к памяти.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

DMB ; Барьер синхронизации доступа к памяти данных

### **DSB**

Барьер синхронизации доступа к памяти данных.

Синтаксис

DSB

Описание

Инструкция DSB выполняет функцию барьерной синхронизации доступа к памяти данных. Команды, которые будут следовать, в порядке выполнения, после DSB, не начнут исполняться до ее завершения. Инструкция DSB завершает свою работу после того, как будут выполнены все инициированные перед ней явные (explicit) операции доступа к памяти.

### Флаги

Данная инструкция не влияет на состояние флагов.

### Примеры

DSB ; Барьер синхронизации доступа к памяти данных

## ISB

Барьер синхронизации доступа к инструкциям.

### Синтаксис

ISB

### Описание

Команда ISB выполняет функцию барьерной синхронизации выполнения команд. Она осуществляет сброс конвейера инструкций процессора, гарантируя таким образом, что все команды, расположенные после инструкции ISB, по окончании ее исполнения будут загружены в конвейер повторно.

### Флаги

Данная инструкция не влияет на состояние флагов.

### Примеры

ISB ; Барьер синхронизации доступа к инструкциям

## MRS

Считать содержимое специального регистра в регистр общего назначения.

### Синтаксис

MRS Rd, spec\_reg

Rd - регистр-получатель результата.

spec\_reg - один из специальных регистров ядра: APSR, IPSR, EPSR, IEPSR, IAPSR, EAPSR, PSR, MSP, PSP, PRIMASK или CONTROL.

### Описание

Инструкции MRS совместно с MSR используются для чтения-модификации-записи элементов PSR, например, для сброса флага Q.

В коде, отвечающем за переключение процессов, необходимо обеспечить сохранение состояния приостановленного процесса, и восстановление состояния активизированного процесса. Необходимой составной частью сохраняемой (восстанавливаемой) информации является значение регистра PSR. При этом на этапе сохранения состояния используется команда MRS, а на этапе восстановления - команда MSR.

См. также описание инструкции MSR ниже.

**Ограничения**

В качестве регистра-получателя Rd нельзя использовать SP или PC.

**Флаги**

Данная инструкция не влияет на состояние флагов.

**Примеры**

MRS R0, PRIMASK; Читать значение PRIMASK и записать значение в R0

**MSR**

Записать регистр общего назначения в специальный регистр.

**Синтаксис**

MSR spec\_reg, Rn

Rn - регистр-источник данных.

spec\_reg - один из специальных регистров: APSR, IPSR, EPSR, IEPSR, IAPSR, EAPSR, PSR, MSP, PSP, PRIMASK или CONTROL.

**Описание**

Доступ к специальным регистрам в команде MSR различен для привилегированных и непривилегированных приложений. Непривилегированному приложению доступен только регистр APSR. При этом попытки записи в нераспределенные биты, а также в EPSR игнорируются.

Привилегированное приложение имеет доступ ко всем специальным регистрам.

См. также описание инструкции MRS выше.

**Ограничения**

В качестве регистра-источника данных Rn нельзя использовать SP или PC.

**Флаги**

Данная инструкция не влияет на состояние флагов.

**Примеры**

MSR CONTROL, R1 ; Записать значение регистра R1 в регистр CONTROL

**NOP**

Нет операции.

**Синтаксис**

NOP

**Описание**

Инструкция NOP ничего не делает. В частности, эта инструкция в некоторых случаях может быть автоматически исключена из конвейера команд, и таким образом, выполнена за ноль тактов. Команду NOP рекомендуется использовать

для заполнения, например, с целью разместить очередную инструкцию по адресу, выровненному по 64-битной границе.

**Флаги**

Данная инструкция не влияет на состояние флагов.

**Примеры**

NOP ; нет операции

**SEV**

Установить признак события.

**Синтаксис**

SEV

**Описание**

Инструкция SEV используется для передачи информации о событии всем процессорам в составе многопроцессорной системы. Кроме того, он устанавливает собственный регистр события в 1.

См. также раздел «

**Управление электропитанием».**

**Флаги**

Данная инструкция не влияет на состояние флагов.

**Примеры**

SEV ; Послать признак события

**SVC**

Вызов супервизора.

**Синтаксис**

SVC #imm8

**где:**

imm8 - константное выражение, целое число в диапазоне от 0 до 255 (8-битное число).

**Описание**

Инструкция SVC вызывает формирование исключения SVC. Параметр imm8 игнорируется процессором. При необходимости он может быть получен обработчиком исключения для определения запрошенного приложением варианта обслуживания.

**Флаги**

Данная инструкция не влияет на состояние флагов.

**Примеры**

SVC 0x32 ; Вызов супервизора  
; обработчик SVC может извлечь параметр по сохраненному в  
стеке,  
; адресу PC приложения.

**WFE**

Ожидать событие.

**Синтаксис**

WFE

**Описание**

В случае если регистр события равен 0, выполнение команды WFE приводит к приостановке исполнения команд до тех пор, пока не произойдет одно из следующих событий:

- исключение, не запрещенное путем установки маски или текущим уровнем приоритета;
- перевод исключения в состояние ожидания обслуживания при установленном в 1 бите SEVONPEND регистра управления системой SCR (см. Таблица 132);
- получение запроса на переход в режим отладки, в случае, если отладка разрешена;

- получение сигнала о событии от периферийного устройства или от другого процессора (по команде SEV) в многопроцессорной системе.  
В случае если регистр события равен 1, команда WFE сбрасывает его в 0, после чего завершает свое функционирование без приостановки процессора.  
Более подробная информация отражена в разделе «

**Управление электропитанием».**

**Флаги**

Данная инструкция не влияет на состояние флагов.

**Примеры**

WFE ; Ожидание события

**WFI**

Ожидание прерывание.

**Синтаксис**

WFI

**Описание**

Команда WFI приостанавливает процессор до тех пор, пока не произойдет одно из следующих событий:

- исключение;
- запрос на перевод в режим отладки, вне зависимости от того, разрешен или запрещен этот режим.

**Флаги**

Данная инструкция не влияет на состояние флагов.

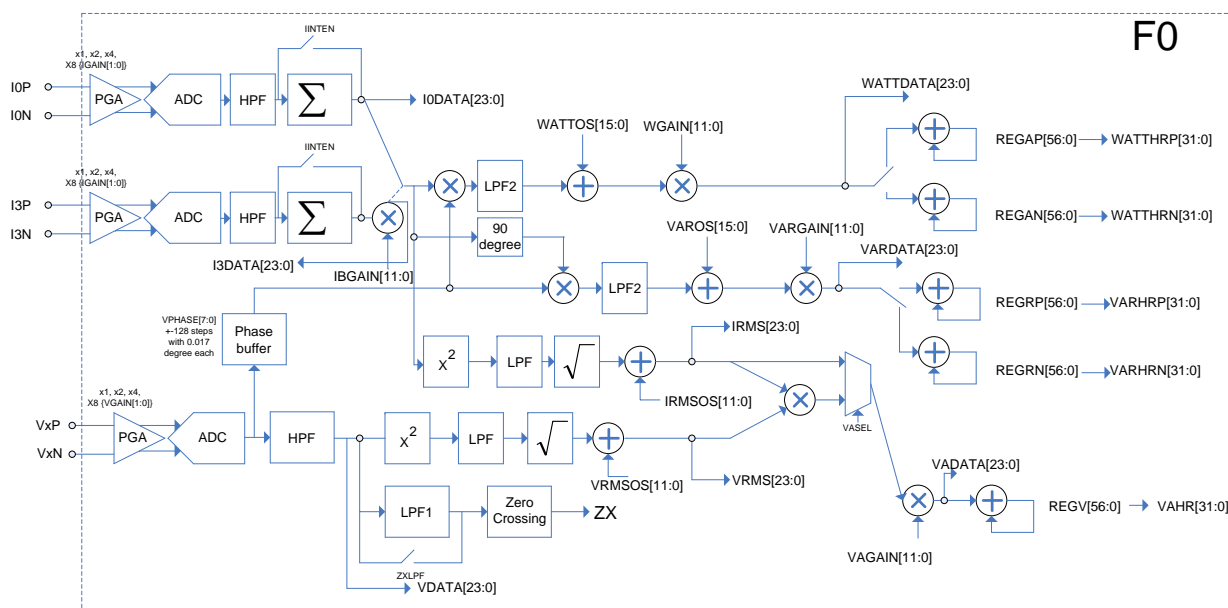
**Примеры**

WFI ; Ожидание прерывания



**Блок АЦП для измерения напряжений и токов в электрической сети**

Микроконтроллер имеет в своем составе блок из 3 каналов 24 битных  $\Sigma\Delta$  АЦП. Каждый из 3 каналов оцифровывает входной сигнал с выходной частотой отсчетов 4/8/16кГц. Кроме этого по вычисленным значениям тока и напряжения реализована возможность рассчитывать среднеквадратические значения тока/напряжения, вычислять активную и реактивную мощности, вычислять потребленную активную и реактивную энергию, частоту сигнала в каналах напряжения, превышение пикового значения, падение сигнала ниже установленного уровня. Эти дополнительные блоки позволяют снизить нагрузку на процессор, что в свою очередь снижает потребляемую мощность всего кристалла.



**Рис. 18** Структурная схема 3 каналов АЦП

Список вычисляемых параметров блоком АЦП:

3 независимых АЦП с выходной частотой отсчетов 4/8/16кГц (2 канала тока и 1 канал напряжения).

В блоке каналов реализован автоматический выбор канала тока (который имеет максимальное значение) для последующих расчетов мощностных характеристик. Если разница токов превышает 6%, то формируется прерывание. Все каналы АЦП имеют независимые калибровочные коэффициенты наклона характеристики.

Каждый канал тока имеет независимый интегратор.

Рассчитывается период сигнала по каналу напряжения. Количество периодов, в течении которого рассчитывается эта величина, можно задавать 1/2/4/8/16/32/64/128 периодам.

Есть проверка на пропажу периодического сигнала в канале напряжения. Проверяется просадка напряжения ниже заданного уровня, а так же превышения сигнала в каналах тока и напряжения установленного лимита.

Есть возможность скорректировать фазы сигналов в канале напряжения с точностью до 0.02%.

Вычисляются среднеквадратические, квадрат среднеквадратических значений токов и напряжений, а так же их независимая калибровка.

При вычислении активной и реактивной энергиях значение накопленной энергии в течении периода накапливаются в отдельных регистрах (для положительной и отрицательной энергии).

Вычисляется полная мощность и полная энергия.

Для предотвращения влияния высокочастотных помех на результаты вычисления необходимо поставить внешний антиэлайзинговый фильтр. Можно использовать простейший RC фильтр первого порядка с частотой среза 10кГц. Все цифровые фильтры настроены на указанные частоты среза при входной частоте АЦП равной 8.192МГц (выходной частоте отсчетов 4/8/16кГц)

**Описание регистров управления блока 3 канального  $\Sigma\Delta$  АЦП**

**Таблица 32**

Базовый Адрес	Название	Описание
0x4006_8000	ADCIU	Контроллер АЦП напряжения/тока
Смещение		
0x000	ADCIU_CTRL1	Общее управление для контроллера АЦП
0x004	ADCIU_CTRL2	
0x008	ADCIU_CTRL3	
0x00C	ADCUI_F0CTR	Управление в канале F0
0x010	ADCUI_F0WC	Управление расчета активной мощности в канале F0
0x014	ADCUI_F0WATTP	Значение положительной активной мощности в канале F0
0x018	ADCUI_F0WATTN	Значение отрицательной активной мощности в канале F0
0x01C	ADCUI_F0VC	Управление расчета реактивной мощности в канале F0
0x020	ADCUI_F0VARP	Значение положительной реактивной мощности в канале F0
0x024	ADCUI_F0VARN	Значение отрицательной реактивной мощности в канале F0
0x028	ADCUI_F0AC	Управление расчета полной мощности в канале F0
0x02C	ADCUI_F0VR	Значение полной мощности в канале F0
0x030	ADCUI_F0MD0	Параметры 0 канала F0
0x034	ADCUI_F0MD1	Параметры 1 канала F0
0x038	ADCUI_F0VPEAK	Пиковое значение в канале напряжения в канале F0
0x03C	ADCUI_F0IPEAK	Пиковое значение в канале тока в канале F0
0x040	ADCUI_F0VDAT	Отсчеты напряжения в канале F0
0x044	ADCUI_F0IODAT	Отсчеты тока I0 в канале F0
0x048	ADCUI_F0I3DAT	Отсчеты тока I3 в канале F0
0x04C	ADCUI_F0VRMS	Среднеквадратическое значение напряжение канала F0
0x050	ADCUI_F0VRMS2	Квадрат RMS в канале напряжения F0
0x054	ADCUI_F0IRMS	Среднеквадратическое значение тока канала

		F0
0x058	ADCUI_F0IRMS2	Квадрат RMS в канале тока F0
0x05C	ADCUI_F0STAT	Статус канала F0
0x060	ADCUI_F0MASK	Маска прерываний канала F0
0x064-0x110		Зарезервировано
0x114	ADCUI_CCAL1	Регистр 1 калибровки канала тока
0x118-0x11C		Зарезервировано
0x120	ADCUI_CCAL4	Регистр 4 калибровки канала тока

### ADCUI\_CTRL1

Номер	14	13:12	11:9	8	7	6	5	4	3	2	1	0
Доступ	-	R/W	R/W	R/W	R/W	R/W	-	-	-	-	-	R/W
Сброс	-	0	000	0	0	0	-	-	-	-	0	0
	-	APN OLO AD	PER_ LEN GTH	ZXLP F	RES OL	I3EN	-	-	-	-	VOEN	IOEN
Номер	31:30	29	28	27	26:25	24:23	22	21	20	19	18:17	16:15
Доступ	R/W	R/W	R/W	R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
Сброс	00	0	0	0	-	-	-	0	0	0	0	0
	OSR_ CO NF	IBO OST	RES ET_D IG	ZXR MS	-	-	-	BUF_ BYP	VREF_ SEL	FRE QSEL	VAN OLO AD	VAR NOL OAD

**Таблица 33** Описание битов регистра ADCUI\_CTRL

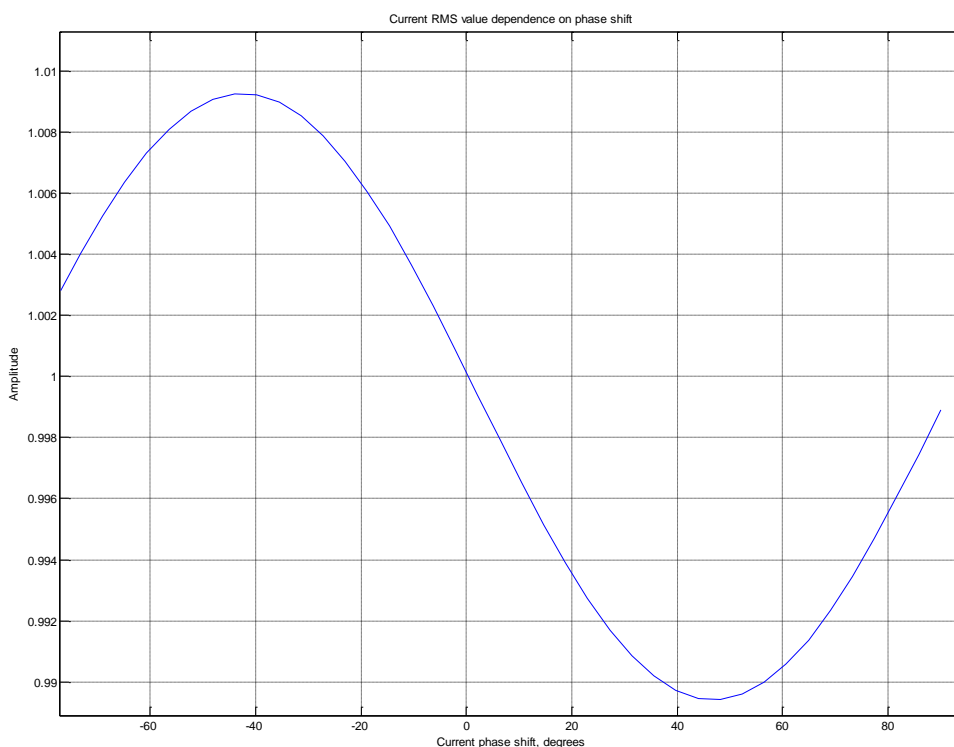
№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:30	OSR_CONF*	Выбор коэффициента передисретизации 00 – 256 (4кГц) 01 – 128 (8кГц) 10 – 64 (16кГц) 11 – Зарезервировано
29	IBOOST	Увеличение тока АЦП 0 – Нормальный режим 1 – Увеличение тока
28	RESET_DIG	Сброс цифровой части блоков АЦП 0 – нет сброса 1 – цифровая часть под общим сбросом
27	ZXRMS**	Управления обновления регистров со среднеквадратическими значениями 0 – непрерывное обновление 1 – обновление при пересечении напряжением "0"
26:22	-	-
21	BUF_BYP	Буферизация опорного напряжения 0 – опорное напряжение буферизировано 1 – опорное напряжение небуферизировано
20	VREF_SEL	Выбор опорного напряжения для АЦП 0 – внутреннее опорное напряжение 1 – внешнее опорное напряжение

19	FREQSEL	Разрешение вычисления длительности периода в каналах напряжения 1 – разрешено 0 – хранится последнее вычисленное значение
18:17	VANOLOAD	Режим “без нагрузки” при вычислении полной энергии 00 – вся вычисленная энергия накапливается 01 – не учитывается энергия ниже 0.012% от полной шкалы 10 – не учитывается энергия ниже 0.0061% от полной шкалы 11 – не учитывается энергия ниже 0.00305% от полной шкалы
16:15	VARNLOAD	Режим “без нагрузки” при вычислении реактивной энергии 00 – вся вычисленная энергия накапливается 01 – не учитывается энергия ниже 0.012% от полной шкалы 10 – не учитывается энергия ниже 0.0061% от полной шкалы 11 – не учитывается энергия ниже 0.00305% от полной шкалы
14	-	Зарезервировано
13:12	APNOLOAD	Режим “без нагрузки” при вычислении активной энергии 00 – вся вычисленная энергия накапливается 01 – не учитывается энергия ниже 0.012% от полной шкалы 10 – не учитывается энергия ниже 0.0061% от полной шкалы 11 – не учитывается энергия ниже 0.00305% от полной шкалы
11:9	PER_LENGTH	Диапазон вычисления периода и фазового сдвига 000 – в течение 1 периода 001 – в течение 2 периодов ... 111 – в течение 128 периодов
8	ZXLPF	Отключение низкочастотного фильтра перед детектором пересечения “0” в каналах напряжения 0 – фильтр включен 1 – фильтр отключен
7	RESOL	Разрешение выходных данных 0 – 16 бит 1 – 24 бита
6	I3EN	Разрешение работы канала I3 0 – канал отключен 1 – канал включен
5:2	-	-
1	V0EN	Разрешение работы канала V0 0 – канал отключен 1 – канал включен

0	IOEN	Разрешение работы канала IO 0 – канал отключен 1 – канал включен
---	------	--

\* - при увеличении частоты дискретизации все внутренние цифровые фильтры соответствующим образом корректируются, что сохраняет их частоты среза постоянными. Так же необходимо учитывать, что увеличение частоты дискретизации в 2 раза ведет к уменьшению суммарного коэффициента гармоник+шума (THD+noise) как минимум на 3дБ в полосе от 0Гц до половины частоты дискретизации (это следует из того, что шум интегрируется в частоте 2 раза большей).

\*\* - так как происходит одновременное обновление среднеквадратических значений, тока и напряжения, то значение тока будет зависеть от угла между напряжением и током. На графике ниже видна эта зависимость. Исходя из этих данных, можно скорректировать действительное значение тока.



**Рис. 19**

### ADCUI\_CTRL2

Номер	31:24	23:16	27:16
Доступ	-	R/W	R/W
Сброс	-	00h	0000h

SAGCYC	SAGLVL
--------	--------

**Таблица 34** Описание битов регистра ADCUI\_CTRL2

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..24	-	Зарезервировано
23:16	SAGCYC	Количество полутактов напряжения для вычисления просадки уровня напряжения
15:0	SAGLVL	Уровень разрешенной просадки напряжения

### ADCUI\_CTRL3

Номер 31:12 11:0  
 Доступ - R/W  
 Сброс - 000h

ZXTOUT
--------

**Таблица 35** Описание битов регистра ADCUI\_CTRL3

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..12	-	Зарезервировано
11:0	ZXTOUT	Значение time-out регистра, который устанавливает это значение при пересечении сигнала напряжения "0"

### ADCUI\_F0CTR

Номер 31:20 19:18 17:10 9:8 7:6 5 4 3 2 1 0  
 Доступ R/W R/W R/W R/W R/W WO WO WO R/W R/W R/W  
 Сброс 0 00 00 0 0 0 0 0 0 0 0

F0IRMSOS	F0I3GAIN	F0VPHASE	F0VGA	F0IGA	F0RVS	F0RRS	F0RAS	F0VAS	F0I3N	F0I0N
----------	----------	----------	-------	-------	-------	-------	-------	-------	-------	-------

**Таблица 36.** Описание битов регистра ADCUI\_F0CTR

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:20	F0IRMSOS	Калибровка вычислителя среднеквадратического значения тока
19:18	F0I3GAIN	Предусилитель в канале тока 3 00 – 0dB 01 – 6dB 10 – 12dB 11 – 18dB
17:10	F0VPHASE	Фазовый сдвиг канала напряжения относительно канала тока, записанный в дополнительном коде. От -126d (-123мкс) до +127d(+124мкс). "0" соответствует синфазному сигналу с током.

9:8	F0VGAIN	Предусилитель в канале напряжения 00 – 0dB 01 – 6dB 10 – 12dB 11 – 18dB
7:6	F0IOGAIN	Предусилитель в канале тока 0 00 – 0dB 01 – 6dB 10 – 12dB 11 – 18dB
5	F0RVRS	Запись в этот регистр сбрасывает счетчик переданной полной энергии
4	F0RRRS	Запись в этот регистр сбрасывает счетчик переданной реактивной энергии
3	F0RARS	Запись в этот регистр сбрасывает счетчик переданной активной энергии
2	F0VASEL	Выбор источника сигнала для сохранения в регистре полной энергии 0 – полная энергия 1 – среднеквадратическое значение тока
1	F0I3NTEN	Отключение интегратора в канале тока 3 0 – интегратор включен 1 – интегратор отключен
0	F0I0NTEN	Отключение интегратора в канале тока 0 0 – интегратор включен 1 – интегратор отключен

### ADCUI\_F0WC

Номер	31:28	27:16	15:0
Доступ	-	R/W	R/W
Сброс	-	7FFh	0000h

F0WGAIN	F0WATTOS
---------	----------

**Таблица 37.** Описание битов регистра ADCUI\_F0WC

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..28	-	Зарезервировано
27:16	F0WGAIN	Калибровка усиления канала
15:0	F0WATTOS	Калибровка смещения канала

### ADCUI\_F0WATTP

Номер	31:0
Доступ	RO
Сброс	

F0WATTHRP

**Таблица 38.** Описание битов регистра ADCUI\_F0WATTP

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:0	F0WATTHRP	Старшие 32 бита внутреннего 57 битного аккумулятора положительной активной энергии

### ADCUI\_F0WATTN

Номер 31:0  
 Доступ RO  
 Сброс

F0WATTHRN

**Таблица 39.** Описание битов регистра ADCUI\_F0WATTN

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:0	F0WATTHRN	Старшие 32 бита внутреннего 57 битного аккумулятора отрицательной активной энергии

### ADCUI\_F0VC

Номер	31:28	27:16	15:0
Доступ	-	R/W	R/W
Сброс	-	000h	0000h
		F0VARGAIN	F0VAROS

**Таблица 40.** Описание битов регистра ADCUI\_F0VC

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..28	-	Зарезервировано
27:16	F0VARGAIN	Калибровка усиления канала
15:0	F0VAROS	Калибровка смещения канала

### ADCUI\_F0VARP

Номер 31:0  
 Доступ RO  
 Сброс

F0VARHRP



**Таблица 41.** Описание битов регистра ADCUI\_F0VARP

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:0	F0VARHRP	Старшие 32 бита внутреннего 57 битного аккумулятора положительной реактивной энергии

### ADCUI\_F0VARN

Номер 31:0  
 Доступ RO  
 Сброс

F0VARHRN
----------

**Таблица 42.** Описание битов регистра ADCUI\_F0VARN

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:0	F0VARHRN	Старшие 32 бита внутреннего 57 битного аккумулятора отрицательной реактивной энергии

### ADCUI\_F0AC

Номер	31:28	27:16	15:12	11:0
Доступ	-	R/W	-	R/W
Сброс	-	000h	-	000h

F0VAGAIN	-	F0VRMSOS
----------	---	----------

**Таблица 43** Описание битов регистра ADCUI\_F0AC

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..28	-	Зарезервировано
27:16	F0VAGAIN	Калибровка усиления канала
15:12	-	Зарезервировано
11:0	F0VRMSOS	Калибровка вычислителя среднеквадратического значения напряжения

### ADCUI\_F0VR

Номер 31:0  
 Доступ RO  
 Сброс

F0VAHR
--------

**Таблица 44.** Описание битов регистра ADCUI\_F0VR

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:0	F0VAHR	Старшие 32 бита внутреннего 57 битного аккумулятора полной энергии

### ADCUI\_F0MDO

Номер	31:20	29	28:12	11:9	8	7	6	5	4	3:2	1:0
Доступ	R/W	R/W	RO	-	R/W	R/W	R/W	R	R	R/W	R/W
Сброс	00	0	-	0	0	0	0	0	0	00	00
	F0SEL_I_CH	F0I3SEL	F0PER_FREQ	-	I3GAIN	V0GAIN	I0GAIN	F0REACTS	F0ACTS	F0ISEL	F0VSEL

**Таблица 45** Описание битов регистра ADCUI\_F0MDO

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..30	F0SEL_I_CH	Выбор активного канала тока для вычисления мощностных хар-к 00,11 – автоматический выбор канала 01 – активный канал I0 10 – активный канал I3
29	F0I3SEL	Выбор источника сигнала для регистра ADCUI_F0I3DAT 0 – после фильтра высоких частот 1 – до фильтра высоких частот
28:12	F0PER_FREQ	Длительность такта в канале напряжения
11:9	-	Зарезервировано
8	I3GAIN	Усиление в канале I3 0 – нет усиления 1 – +6дБ усиление
7	V0GAIN	Усиление в канале V0 0 – нет усиления 1 – +6дБ усиление
6	I0GAIN	Усиление в канале I0 0 – нет усиления 1 – +6дБ усиление
5	F0REACTS	Знак реактивной энергии в последний период
4	F0ACTS	Знак активной энергии в последний период
3:2	F0ISEL	Выбор источника сигнала для регистра ADCUI_F0IDAT 00 – отсчеты тока 01 – отсчеты активной мощности 10 – отсчеты реактивной мощности 11 – отсчеты полной мощности
1:0	F0VSEL	Выбор источника сигнала для регистра ADCUI_F0VDAT 00 – отсчеты напряжения

		01 – отсчеты активной мощности 10 – отсчеты реактивной мощности 11 – отсчеты полной мощности
--	--	--

### ADCUI\_F0MD1

Номер 31:16 15:0  
 Доступ R/W R/W  
 Сброс 0000h 0000h

F0VPKLVL	F0IPKLVL
----------	----------

**Таблица 46** Описание битов регистра ADCUI\_F0MD1

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:16	F0VPKLVL	Предельный разрешенный уровень напряжения
23:16	F0IPKLVL	Предельный разрешенный уровень тока

### ADCUI\_F0VPEAK

Номер 31:24 23:0  
 Доступ R/W R/W  
 Сброс 000000h 000000h

-	F0VPEAK
---	---------

**Таблица 47** Описание битов в регистре ADCUI\_F0VPEAK

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:24	-	Зарезервировано
23:0	F0VPEAK	Пиковое значение напряжения. Обнуляется чтением из регистра Запись в этот регистр, сбрасывает пиковой значение

### ADCUI\_F0IPEAK

Номер 31:24 23:0  
 Доступ R/W R/W  
 Сброс 000000h 000000h

-	F0IPEAK
---	---------

**Таблица 48** Описание битов в регистре ADCUI\_F0IPEAK

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:24	-	Зарезервировано
23:0	F0IPEAK	Пиковое значение тока. Обнуляется чтением из регистра Запись в этот регистр, сбрасывает пиковой значение

### ADCUI\_F0V DAT

Номер 31:24 23:0  
 Доступ RO  
 Сброс 000000h

-	F0V DAT
---	---------

**Таблица 49** Описание битов в регистре ADCUI\_F0V DAT

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:24	-	Зарезервировано
23:0	F0V DAT	FIFO отсчетов напряжения (или одной из мощностей)

### ADCUI\_F0I0 DAT

Номер 31:24 23:0  
 Доступ RO  
 Сброс 000000h

-	F0I0 DAT
---	----------

**Таблица 50** Описание битов в регистре ADCUI\_F0I0 DAT

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:24	-	Зарезервировано
23:0	F0I0 DAT	FIFO отсчетов тока 0 (или одной из мощностей)

### ADCUI\_F0I3 DAT

Номер 31:24 23:0  
 Доступ RO  
 Сброс 000000h

-	F0I3 DAT
---	----------

**Таблица 51** Описание битов в регистре ADCUI\_F0I3 DAT

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:24	-	Зарезервировано
23:0	F0I3 DAT	FIFO отсчетов тока 3

### ADCUI\_F0VRMS

Номер 31:24 23:0  
 Доступ RO  
 Сброс 000000h

-	F0VRMS
---	--------

**Таблица 52** Описание битов в регистре ADCUI\_F0VRMS

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:24	-	Зарезервировано
23:0	F0VRMS	Среднеквадратическое значение напряжения

### ADCUI\_F0VRMS2

Номер 31:0  
 Доступ RO  
 Сброс

F0VRMS2
---------

**Таблица 53** Описание битов в регистре ADCUI\_F0VRMS2

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:0	F0VRMS2	Квадрат среднеквадратического значения напряжения

### ADCUI\_F0IRMS

Номер 31:24 23:0  
 Доступ RO  
 Сброс 000000h

-	F0IRMS
---	--------

**Таблица 54.** Описание битов в регистре ADCUI\_F0IRMS

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:24	-	Зарезервировано
23:0	F0IRMS	Среднеквадратическое значение тока

### ADCUI\_F0IRMS2

Номер 31:0  
 Доступ RO  
 Сброс

F0IRMS2
---------

**Таблица 55** Описание битов в регистре ADCUI\_F0VRMS2

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:0	F0IRMS2	Квадрат среднеквадратического значения тока

### ADCUI\_F0STAT

Номер	6	5	4	3	2	1	0					
Доступ	R/W	R/W	RO	RO	R/W	RO	RO					
Сброс	0	0	0	0	0	0	0					
	F0SAGF	F0IF_OVR	F0IF_FLL	F0IF_EMP	F0VF_OVR	F0VF_FLL	F0VF_EMP					
Номер	15	14	13	12	11	10	9	8	7			
Доступ	R/W	R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W			
Сброс	0	0	0	0	0	0	0	0	0			
	F0APSIGN	F0FAULTCON	F0ICHANNEL	F0ZXTOF	F0VAOV	F0VAROVP	F0WATTOVP	F0PEAKIF	F0PEAKVF			
Номер	31:27	26	25	24	23	22	21	20	19	18	17	16
Доступ	-	R/W	R/W	R/W	RO	RO	R/W	RO	-	RO	R/W	RO
Сброс	-	0	0	0	0	0	0	0	-	0	0	0
	F0VAROVN	F0WATTOVN	C3IF_OVR	C3IF_FLL	C3IF_EMP	F0ZEROCRS	F0VANLDFL	-	F0VARNLDL	F0VARNSIG	F0APNLDL	

**Таблица 56.** Описание битов в регистре ADCUI\_F0STAT

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..27	-	Зарезервировано
26	F0VAROVN	Флаг, что произошло переполнения регистра аккумулятора с отрицательной реактивной энергией Запись "1" сбрасывает этот флаг
25	F0WATTOVN	Флаг, что произошло переполнения регистра аккумулятора с отрицательной активной энергией Запись "1" сбрасывает этот флаг
24	C3IF_OVR	Флаг, что произошло переполнения FIFO C3IDAT Запись "1" сбрасывает этот флаг
23	C3IF_FLL	Флаг, что FIFO C3IDAT заполнено
22	C3IF_EMP	Флаг, что FIFO C3IDAT пусто
21	F0ZEROCRS	Флаг, что произошло пересечение "0" в канале напряжения. Запись "1" сбрасывает этот флаг
20	F0VANLDFL	Полная мощность ниже уровня сравнения
19	-	Зарезервировано

18	F0VARNLDFL	Реактивная мощность ниже уровня сравнения
17	F0VARSIGN	Смена знака реактивной мощности Запись "1" сбрасывает этот флаг
16	F0APNLDFL	Активная мощность ниже уровня сравнения
15	F0APSIGN	Смена знака активной мощности Запись "1" сбрасывает этот флаг
14	F0FAULTCON	Произошло автоматическое переключение активного канала тока Запись "1" сбрасывает этот флаг
13	F0ICHANNEL	Активный канал тока 0 – активный канал I0 1 – активный канал I3
12	F0ZXTOF	Флаг, что в течении TimeOut не было пересечение напряжением значения "0" Запись "1" сбрасывает этот флаг
11	F0VAOV	Флаг, что произошло переполнения регистра аккумулятора с полной энергией Запись "1" сбрасывает этот флаг
10	F0VAROVP	Флаг, что произошло переполнения регистра аккумулятора с положительной реактивной энергией Запись "1" сбрасывает этот флаг
9	F0WATTOVP	Флаг, что произошло переполнения регистра аккумулятора с положительной активной энергией Запись "1" сбрасывает этот флаг
8	F0PEAKIF	Флаг, что произошло превышение порогового значения тока Запись "1" сбрасывает этот флаг
7	F0PEAKVF	Флаг, что произошло превышение порогового значения напряжения Запись "1" сбрасывает этот флаг
6	F0SAGF	Флаг, что произошла просадка напряжения Запись "1" сбрасывает этот флаг
5	F0IF_OVR	Флаг, что произошло переполнения FIFO F0IDAT Запись "1" сбрасывает этот флаг
4	F0IF_FLL	Флаг, что FIFO F0IDAT заполнено
3	F0IF_EMP	Флаг, что FIFO F0IDAT пусто
2	F0VF_OVR	Флаг, что произошло переполнения FIFO F0VDAT Запись "1" сбрасывает этот флаг
1	F0VF_FLL	Флаг, что FIFO F0VDAT заполнено
0	F0VF_EMP	Флаг, что FIFO F0VDAT пусто

### ADCUI\_F0MASK

Номер	6	5	4	3	2	1	0
Доступ	R/W	R/W	RO	RO	R/W	RO	RO
Сброс	0	0	0	0	0	0	0
	F0SAGFM	F0IF_OVR M	F0IF_FLL M	F0IF_EMP M	F0VF_OVR RM	F0VF_FLL M	F0VF_EMP PM

Номер	15	14	13	12	11	10	9	8	7			
Доступ	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W			
Сброс	0	0	-	0	0	0	0	0	0			
	F0APSIGNM	F0FAULTCONM	-	F0ZXTOFM	F0VAOVM	F0VAROVPM	F0WATTOVPM	F0PEAKIFM	F0PEAKVFM			
Номер	31:27	26	25	24	23	22	21	20	19	18	17	16
Доступ	-	R/W	R/W	R/W	RO	RO	R/W	RO	-	RO	R/W	RO
Сброс	-	0	0	0	0	0	0	0	-	0	0	0
	F0VAROVNM	F0WATTOVNM	C3IF_OVRM	C3IF_FLLM	C3IF_EMPM	F0ZEROCRSM	F0VANLDFLM	-	F0VARNLDFLM	F0VARSIGNM	F0APNLDFLM	

**Таблица 57.** Описание битов в регистре ADCUI\_F0MASK

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..27	-	Зарезервировано
26	F0VAROVNM	Маска бита F0VAROVN
25	F0WATTOVNM	Маска бита F0WATTOVN
24	C3IF_OVRM	Маска бита C3IF_OVR
23	C3IF_FLLM	Маска бита C3IF_FLL
22	C3IF_EMPM	Маска бита C3IF_EMP
21	F0ZEROCRSM	Маска бита F0ZEROCRS
20	F0VANLDFLM	Маска бита F0VANLDFL
19	-	Зарезервировано
18	F0VARNLDFLM	Маска бита F0VARNLDFL
17	F0VARSIGNM	Маска бита F0VARSIGN
16	F0APNLDFLM	Маска бита F0APNLDFL
15	F0APSIGNM	Маска бита F0APSIGNM
14	F0FAULTCONM	Маска бита F0FAULTCON
13	-	Зарезервировано
12	F0ZXTOFM	Маска бита F0ZXTOF
11	F0VAOVM	Маска бита F0VAOV
10	F0VAROVPM	Маска бита F0VAROVP
9	F0WATTOVPM	Маска бита F0WATTOVP
8	F0PEAKIFM	Маска бита F0PEAKIF
7	F0PEAKVFM	Маска бита F0PEAKVF
6	F0SAGFM	Маска бита F0SAGF
5	F0IF_OVRM	Маска бита F0IF_OVR
4	F0IF_FLLM	Маска бита F0IF_FLL
3	F0IF_EMPM	Маска бита F0IF_EMP
2	F0VF_OVRM	Маска бита F0VF_OVR
1	F0VF_FLLM	Маска бита F0VF_FLL
0	F0VF_EMPM	Маска бита F0VF_EMP



**ADCUI\_CCAL1**

**Таблица 58**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..24	-	Зарезервировано
23:12	I0BGAIN	Калибровочный коэффициент канала I0
11:0	V0BGAIN	Калибровочный коэффициент канала V0

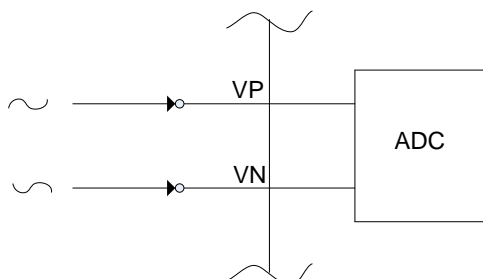
**ADCUI\_CCAL4**

**Таблица 59**

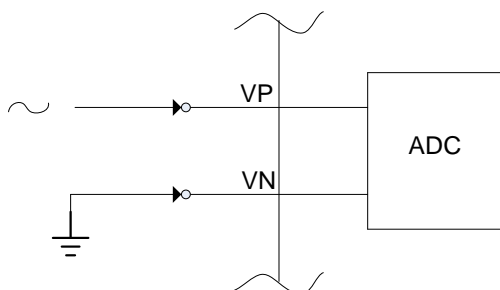
№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..12	-	Зарезервировано
11:0	I3BGAIN	Калибровочный коэффициент канала I3

**Алгоритмы вычисления окончательных результатов и их соответствия внешним сигналам.**

Все параметры вычисленных значений зависят от схемы включения микросхемы, а так же от формата выходных данных. На Рис. 20 приведены два вида включения АЦП: полностью дифференциальное и недифференциальное включение.



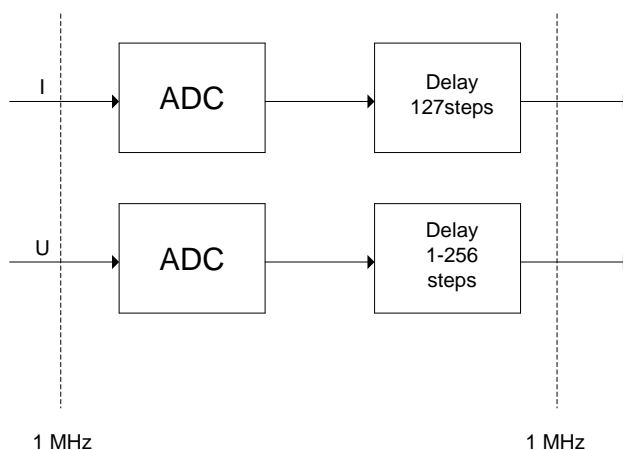
**Рис. 20** Дифференциальное включение



**Рис. 21** Недифференциальное включение

Необходимо иметь в виду, что значения напряжения и токов после АЦП в случае недифференциального включения в 2 раза меньше дифференциального. А мощностные характеристики в 4 раза меньше.

Для коррекции фазового сдвига в канале тока относительно канала напряжения в системе присутствует конфигурируемая линия задержки как показано ниже.



**Рис. 22** Контролируемый фазовый сдвиг в канале напряжения

Изменяя линию задержки в канале напряжения можно регулировать временной сдвиг одного канала относительно другого. Так как частота отсчетов после АЦП равна 1.024МГц, то один шаг равен 1/20480 периода 50Гц сигнала или 0.018 градуса. Необходимо иметь в виду, что в этой системе сдвиг

осуществляется во временной области, поэтому фазовый сдвиг в градусах зависит от частоты.

В качестве децемирующего фильтра используется фильтр со структурой  $\sin c^3$ , его характеристика приведена ниже:

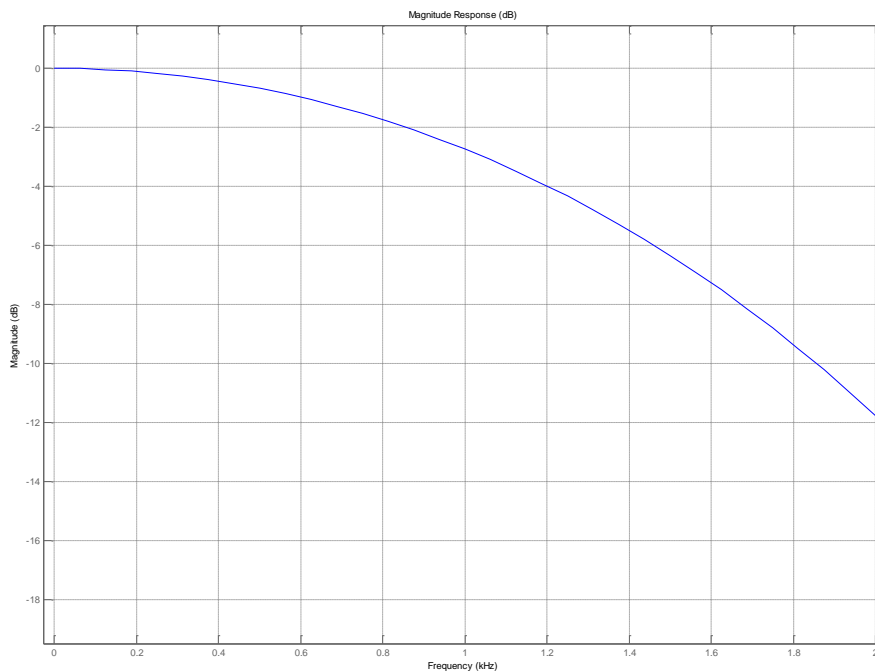


Рис. 23 Характеристика децемирующего фильтра в полосе 2кГц

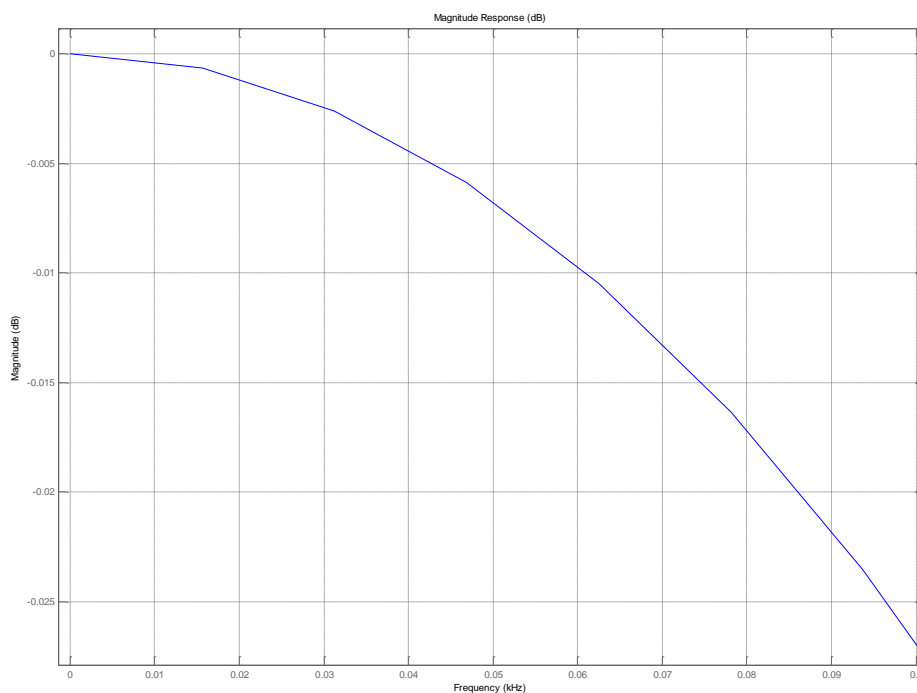


Рис. 24 Характеристика децемирующего фильтра в полосе 100Гц

Как видно из вышеприведенных характеристик, фильтр подавляет частоты близкие к 2кГц до величин 12дБ, что стоит учитывать при измерении гармоник основного тона. В полосе до 100Гц подавление незначительно (на частоте 50Гц подавление 0.008дБ).

Для коррекции влияния смещения в каналах тока и напряжения на вычисленную мощность после децимирующего фильтра в канале тока стоит высокочастотный фильтр. Он убирает постоянную составляющую сигнала. Частота среза фильтра равна 1Гц.

Если FIFO каналов сконфигурированы на прием отсчетов тока и напряжение (FxISEL=00, FxVSEL=00), то значения отсчетов FIFO можно перевести в напряжения на входе по следующим простейшим формулам. Значение приведены для усиления, равному 0дБ. Отсчеты, записанные в FIFO представлены в двоичном формате с дополнением до 2.

**Таблица 60**

<b>Режим</b>	<b>Входное напряжение АЦП, Вольт</b>
Дифференциальное включение, 24 бит. режим	$\frac{FxV\text{DAT}}{2^{23}}, \frac{FxI\text{DAT}}{2^{23}}$
Дифференциальное включение, 16 бит. режим	$\frac{FxV\text{DAT}}{2^{15}}, \frac{FxI\text{DAT}}{2^{15}}$
Недифференциальное включение, 24 бит. режим	$\frac{FxV\text{DAT}}{2^{22}}, \frac{FxI\text{DAT}}{2^{22}}$
Недифференциальное включение, 16 бит. режим	$\frac{FxV\text{DAT}}{2^{14}}, \frac{FxI\text{DAT}}{2^{14}}$

Каждый из каналов тока (кроме I0) может быть скорректирован с помощью коэффициентов IBGAIN в соответствии с нижеприведенной формулой. Значение IBGAIN записывается в двоичном формате с дополнением до 2.

$$I_{COR} = I_{ADC} * \left(1 + \frac{IBGAIN}{2^{11}}\right) \quad 1$$

В регистрах FxVRMS и FxIRMS (см. Таблица 32) хранится вычисленная величина среднеквадратического значения тока и напряжения в соответствующей фазе. В таблице ниже приведены значения среднеквадратических величин.

**Таблица 61**

<b>Режим</b>	<b>Напряжение, Вольт</b>
Дифференциальное включение	$\frac{FxV\text{RMS}}{2^{23}}, \frac{FxI\text{RMS}}{2^{23}}$
Недифференциальное включение	$\frac{FxV\text{RMS}}{2^{22}}, \frac{FxI\text{RMS}}{2^{22}}$

Для вычисления среднеквадратического значения используется следующий алгоритм (для примера выбран канал напряжения, но для канала тока алгоритм идентичный).

Входной сигнал представлен в виде:

$$V(t) = \sqrt{2} * V_{rms} * \sin(\omega t) \quad 2$$

Отсчеты напряжения поступают с частотой 4кГц. Далее каждый отсчет возводится в квадрат, что дает следующий результат:

$$V^2(t) = 2 * V_{rms} * \sin^2(\omega t) = V_{rms}^2 - V_{rms}^2 \cos(2\omega t) \quad 3$$

Таким образом, мы имеем сигнал с постоянной составляющей равной среднеквадратическому значению напряжения и пульсацией с удвоенной частотой по сравнению с входным сигналом. Для фильтрации пульсации полученный сигнал пропускается через фильтр с частотой среза 2Гц. Этот фильтр подавляет пульсации на частоте 100Гц (50Гц\*2) с коэффициентом 35дБ. Отфильтрованный сигнал поступает на блок извлечения квадратного корня. Результирующий сигнал имеет так же пульсации, но ослабленные фильтром. Поэтому рекомендуется использовать режим синхронизации записи среднеквадратического значения с моментом перехода напряжения через 0 (ZXRMS=1).

После извлечения квадратного корня величину смещения среднеквадратического значения можно скорректировать с помощью 12 битных значений FxVRMSOS и FxIRMSOS. Перед корректировкой значение сдвигается на 8 бит вправо, что дает шаг корректировки в 256 меньше. Эта корректировка нужна для того, что бы избавиться ошибки, вызванной шумами на входе АЦП, которые после возведения в квадрат и накопления будут давать отклонения среднего уровня величины  $V^2(t)$ .

Формула коррекции приведена ниже:

$$V_{cor}(t) = V_{rms}(t) + \frac{FxVRMSOS}{2^{20}}, \quad 4$$

Значения FxVRMSOS и FxIRMSOS представлены в виде знаковых величин в двоичном коде с дополнением до 2.

В регистрах FxVRMS2 и FxIRMS2 хранятся значения среднеквадратического значения напряжения и тока до извлечения квадратного корня. В таблице ниже приведены значения квадратов среднеквадратических величин.

**Таблица 62**

<b>Режим</b>	<b>Напряжение, Вольт<sup>2</sup></b>
Дифференциальное включение	$\frac{FxVRMS2}{2^{30}}, \frac{FxIRMS2}{2^{30}}$
Недифференциальное включение	$\frac{FxVRMS2}{2^{28}}, \frac{FxIRMS2}{2^{28}}$

Для вычисления реактивной мощности необходимо сдвинуть сигнал в канале тока на 90 градусов. Это осуществляется с помощью фильтров, которые в достаточно широком диапазоне сохраняют сдвиг равный 90 градусам для обоих каналов. Ниже приведена его фазовая характеристика.

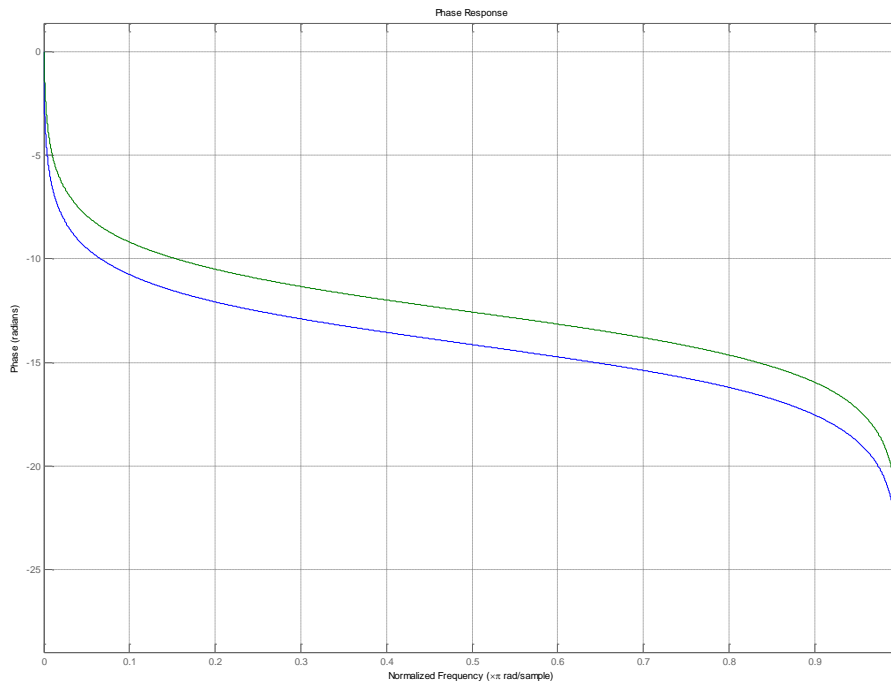


Рис. 25 Фильтр для сдвига сигнала на 90 градусов

Для вычисления активной и реактивной энергии используется подход, похожий на вычисление среднеквадратичного значения, только без извлечения квадратного корня:

$$V(t) = \sqrt{2} * V_{rms} * \sin(\omega t) \quad 5$$

$$I(t) = \sqrt{2} * I_{rms} * \sin(\omega t) \quad 6$$

Тогда мгновенное значение мощности равно произведению тока на напряжение

$$P(t) = V(t) * I(t) = V_{rms} * I_{rms} - V_{rms} * I_{rms} * \cos(2\omega t) \quad 7$$

Среднее значение мощности за целое количество тактов равно

$$P = \frac{1}{nT} \int_0^{nT} P(t) dt = V_{rms} * I_{rms} \quad 8$$

Таким образом, мгновенное значение мощности равно постоянно составляющей произведения тока на напряжение. Для выделения постоянной составляющей используется низкочастотный фильтр с частотой среза 7 Гц. Значение мгновенной мощности можно получить из FIFO *FxV DAT* и *FxI DAT*.

Если FIFO каналов сконфигурированы на прием отсчетов мощностей (*FxISEL=01/10/11*, *FxVSEL=01/10/11*), то значения отсчетов FIFO можно перевести в значения мощностей по следующим простейшим формулам (см. Таблица 63). Значение приведены для усиления равному 0дБ. Отсчеты, записанные в FIFO представлены в двоичном формате с дополнением до 2.

**Таблица 63**

<b>Режим</b>	<b>Мощность, Вольт*Ампер</b>
Дифференциальное включение, 24 бит. режим	$\frac{F_{xV DAT}}{2^{23}}$ , $\frac{F_{xI DAT}}{2^{23}}$
Дифференциальное включение, 16 бит. режим	$\frac{F_{xV DAT}}{2^{15}}$ , $\frac{F_{xI DAT}}{2^{15}}$
Недифференциальное включение, 24 бит. режим	$\frac{F_{xV DAT}}{2^{21}}$ , $\frac{F_{xI DAT}}{2^{21}}$
Недифференциальное включение, 16 бит. режим	$\frac{F_{xV DAT}}{2^{13}}$ , $\frac{F_{xI DAT}}{2^{13}}$

Каждый из каналов мощности имеет независимую калибровку смещения (16 бит), а так же усиления (12 бит). Перед корректировкой смещение сдвигается на 8 бит вправо, что уменьшает шаг корректировки в 256 раз. Корректировка осуществляется в соответствии со следующей формулой:

$$P_{cor} = (P + \frac{P_{os}}{2^{23}}) * (1 + \frac{P_{gain}}{2^{11}})$$

Вычисленная мощность после калибровки накапливается в регистре аккумулятора. Для каждой из 3 мощностей есть свой аккумулятор. Значение в них определяет потребленную энергию. В Таблица 60 приведена формула перевода значения в Ватт\*с.

**Таблица 64**

<b>Режим</b>	<b>Энергия, Ватт*с</b>
Дифференциальное включение	$\frac{F_{xWATTHR}}{512 * 4000}$ , $\frac{F_{xWATTHR}}{512 * 4000}$
Недифференциальное включение	$\frac{F_{xWATTHR}}{512 * 1000}$ , $\frac{F_{xWATTHR}}{512 * 1000}$

Типовая схема включения для учета электроэнергии по одной фазе

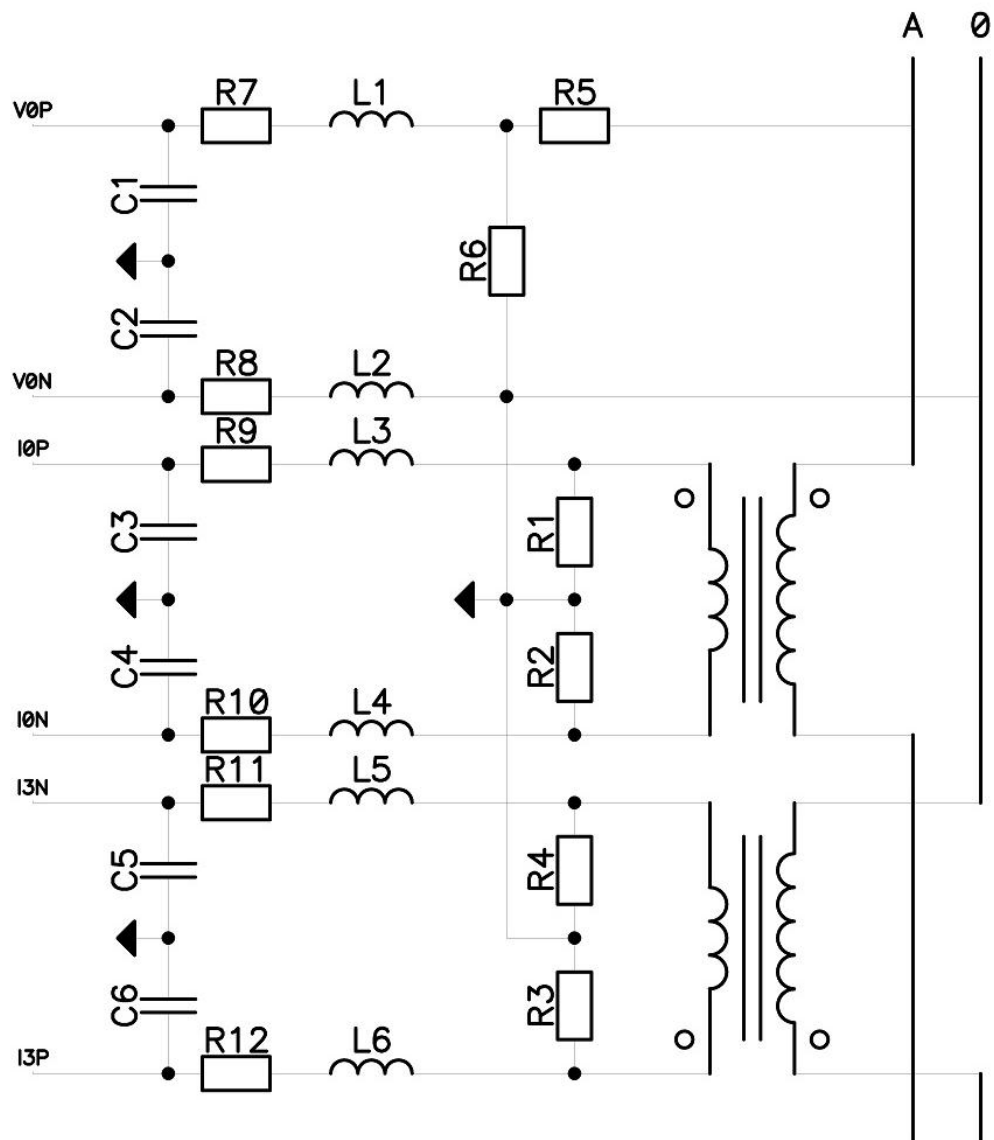


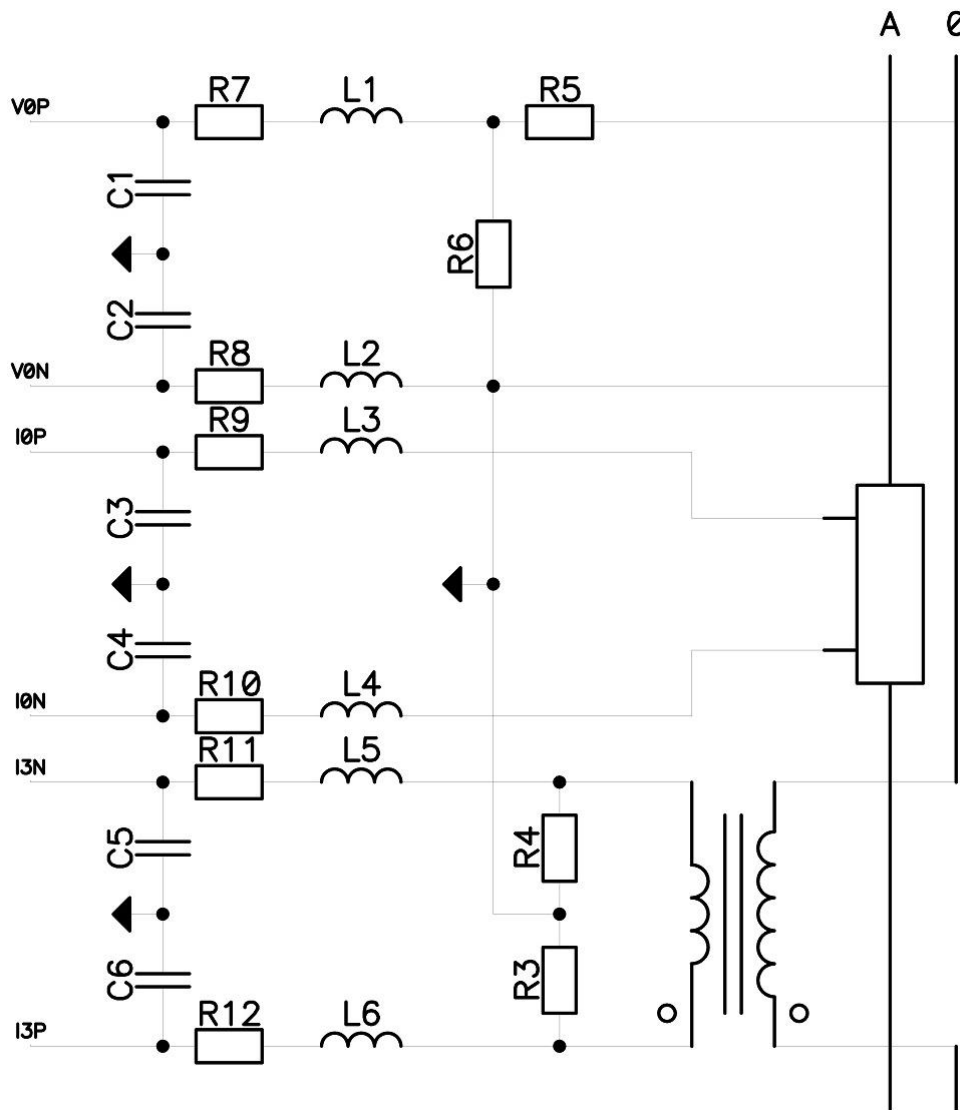
Рис. 26 Типовая схема включения для учета электроэнергии по одной фазе

В данной схеме предусмотрено измерение тока в обоих проводах однофазной сети. Микросхема позволяет в автоматическом режиме учитывать тот ток из каналов I0 и I3, который будет больше. Если учета тока по «нулю» не требуется, то часть схемы, относящуюся к каналу I3 можно убрать. Выход трансформаторов нагружен на резисторный делитель. Сумма сопротивлений этих резисторов, например  $R1+R2$ , должна соответствовать требуемой нагрузке выхода трансформатора. Оба резистора имеют одинаковое сопротивление и создают среднюю точку, относительно которой измеряется ток. В данной схеме включением средней точкой является аналоговая земля. С этой средней точкой соединен нулевой провод. Для подачи сигнала на канал напряжения используется резисторный делитель. При выборе трансформатора и расчете резисторного делителя, стоит помнить, что амплитуда сигнала на входе АЦП не должна выходить за рамки  $\pm 500$  мВ относительно аналоговой земли микросхемы. Непосредственно перед входом каналов АЦП должен быть установлен антиалиасинговый фильтр, рассчитанный на частоту срезу примерно в полтора



раза большую, чем частота дискретизации АЦП. В данном примере это простой RC фильтр низких частот первого порядка. Перед RC фильтром необходимо также установить индуктивности, фильтрующие радиочастотные помехи. Эти индуктивности не относятся функционально к антиалиасинговому фильтру и выбираются для наиболее широкополосного подавления радиочастот. Вы можете использовать свои варианты фильтров в зависимости от требований к конечному изделию.

**Типовая схема включения для учета электроэнергии по одной фазе с использованием шунта**



**Рис. 27** Типовая схема включения для учета электроэнергии по одной фазе с использованием шунта

Также для измерения тока может быть использован шунт. В этом случае следует использовать схему на рисунке выше. Аналогично предыдущей схеме на трансформаторах, здесь можно исключить часть схемы, относящуюся к каналу тока I3, если контроля тока в нулевом проводе не требуется.

## Сигналы тактовой частоты

Микроконтроллер имеет 2 встроенных генератора и 2 внешних осциллятора. А так же специализированный блок формирования тактовой синхронизации PLL микроконтроллера.

## Структурная схема

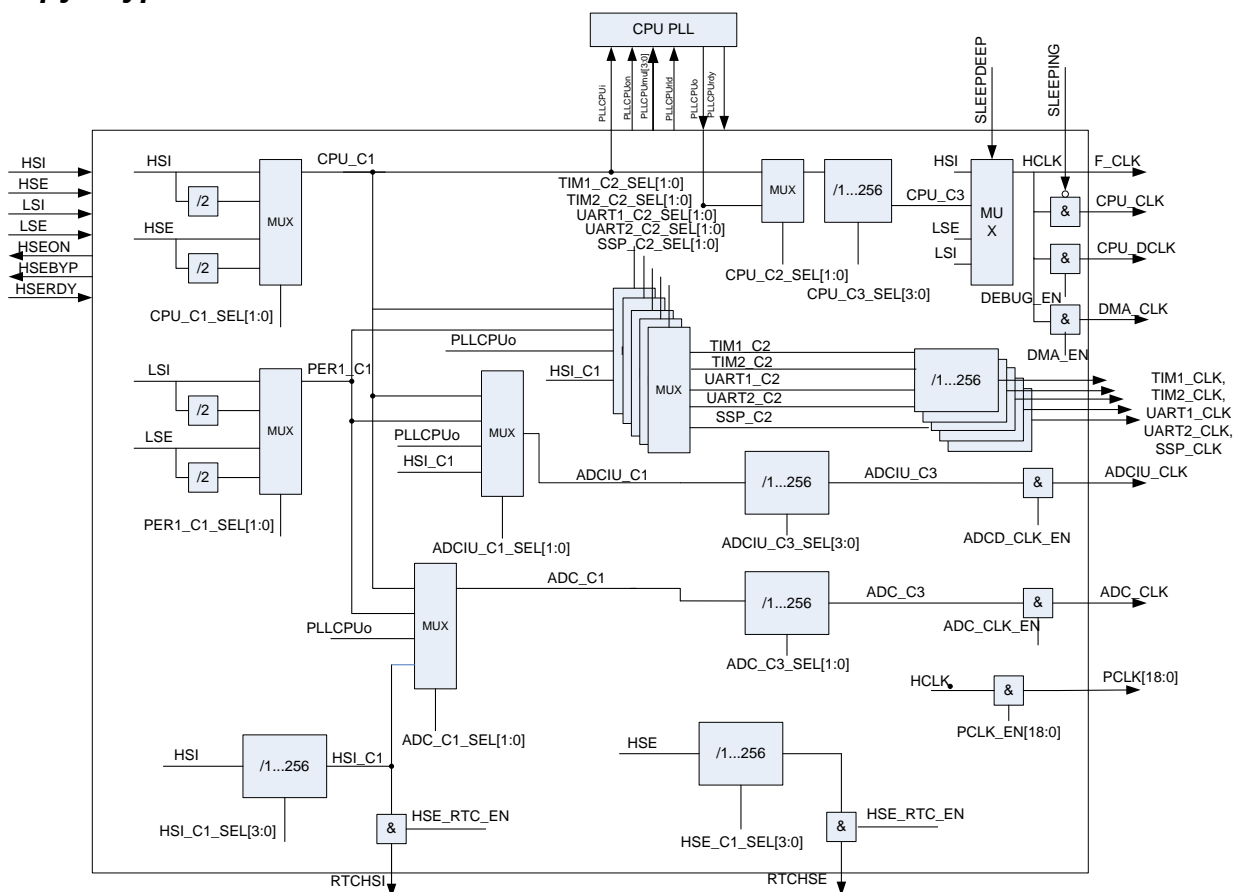


Рис. 28

## Встроенный RC Генератор HSI

Генератор HSI вырабатывает тактовую частоту 8 МГц. Генератор автоматически запускается при появлении питания Ucc и при выходе в нормальный режим работы вырабатывает сигнал HSIRDY в регистре батарейного домена BKP\_REG\_OF (см. Таблица 77). Первоначально процессорное ядро запускается на тактовой частоте HSI. При дальнейшей работе генератор HSI может быть отключен при помощи сигнала HSION в регистре BKP\_REG\_OF. Так же генератор может быть подстроен при помощи сигнала HSITRIM в регистре BKP\_REG\_OF.

## Встроенный RC генератор LSI

Генератор LSI вырабатывает тактовую частоту 40КГц. Генератор автоматически запускается при появлении питания Ucc и при выходе в нормальный режим работы вырабатывает сигнал LSIRDY в регистре BKP\_REG\_OF. Первоначально тактовая частота генератор LSI используется для формирования дополнительной задержки схемы POR. При дальнейшей работе

генератор LSI может быть отключен при помощи сигнала LSION в регистре BKP\_REG\_0F.

### ***Внешний осциллятор HSE***

Осциллятор HSE предназначен для выработки тактовой частоты 2...16 МГц с помощью внешнего резонатора. Осциллятор запускается при появлении питания Ucc и сигнала разрешения HSEON в регистре HS\_CONTROL (см. Таблица 65). При выходе в нормальный режим работы вырабатывает сигнал HSERDY в регистре CLOCK\_STATUS (см. Таблица 65). Так же осциллятор может работать в режиме HSEBYP, когда входная тактовая частота с входа OSC\_IN проходит напрямую на выход HSE. Выход OSC\_OUT находится в этом режиме третьем состоянии.

### ***Внешний осциллятор LSE***

Осциллятор LSE предназначен для выработки тактовой частоты 32 КГц с помощью внешнего резонатора. Осциллятор запускается при появлении питания BDUcc (Рис. 29) и сигнала разрешения LSEON в регистре BKP\_REG\_0F. При выходе в нормальный режим работы вырабатывает сигнал LSERDY в регистре BKP\_REG\_0F. Так же осциллятор может работать в режиме LSEBYP, когда входная тактовая частота с входа OSC\_IN32 проходит напрямую на выход LSE. Выход OSC\_OUT32 находится в этом режиме третьем состоянии. Так как генератор LSE питается от напряжения питания BDUcc и его регистр управления BKP\_REG\_0F расположен в батарейном домене, то генератор может продолжать работать при пропадании основного питания Ucc. Генератор LSE используется для работы часов реального времени.

### ***Встроенный блок умножения системной тактовой частоты***

Блок умножения позволяет провести умножение входной тактовой частоты на коэффициент от 2 до 16, задаваемый на входе PLLCPUMUL[3:0] в регистре PLL\_CONTROL (см. Таблица 65). Входная частота блока умножителя должна быть в диапазоне 2...16МГц выходная до 100 МГц. При выходе блока умножителя тактовой частоты в расчетный режим вырабатывается сигнал PLLCPURDY в регистре CLOCK\_STATUS. Блок включается с помощью сигнала PLLCPUON в регистре PLL\_CONTROL. Выходная частота может быть использована как основная частота процессора и периферии.

Управление тактовыми частотами ведется через периферийный блок RST\_CLK. При включении питания микроконтроллер запускается на частоте HSI генератора. Выдача тактовых сигналов синхронизации для всех периферийных блоков кроме RST\_CLK отключена. Для начала работы с нужным периферийным блоком необходимо включить его тактовую частоту в регистре PER\_CLOCK. Некоторые контроллеры интерфейсов (UART, Таймеры) могут работать на частотах отличных от частоты процессорного ядра, поэтому в соответствующих регистрах (UART\_CLOCK, TIM\_CLOCK) могут быть заданы их скорости работы. Для изменения тактовой частоты ядра можно перейти на другой генератор и/или воспользоваться блоком умножения тактовой частоты. Для корректной смены тактовой частоты сначала должны быть сформированы необходимые тактовые частоты и за тем осуществлено переключение на них на соответствующих мультиплексах управляемом регистре CPU\_CLOCK.

**Описание регистров блока контроллера тактовой частоты**

**Таблица 65**

Базовый Адрес	Название	Описание
0x4002_0000	RST_CLK	Контроллер тактовой частоты
Смещение		
0x00	CLOCK_STATUS	Регистр состояния блока управления тактовой частотой
0x04	PLL_CONTROL	Регистр управления блоками умножения частоты
0x08	HS_CONTROL	Регистр управления высокочастотным генератором и осциллятором
0x0C	CPU_CLOCK	Регистр управления тактовой частотой процессорного ядра
0x10	PER1_CLOCK	Регистр управления тактовой частотой периферийных блоков
0x14	ADC_CLOCK	Регистр управления тактовой частотой АЦП и $\Sigma\Delta$ АЦП
0x18	RTC_CLOCK	Регистр управления формированием высокочастотных тактовых сигналов блока RTC
0x1C	PER2_CLOCK	Регистр управления тактовой частотой периферийных блоков
0x24	TIM_CLOCK	Регистр управления тактовой частотой TIMER
0x28	UART_CLOCK	Регистр управления тактовой частотой UART
0x2C	SSP_CLOCK	Регистр управления тактовой частотой SSP

**CLOCK\_STATUS**

Номер	31	3	2	1	0
Доступ	U	U	RO	RO	RO
Сброс	0	0	0	0	0

-				-	HSE RDY	PLL CPU RDY	PLL USB RDY
---	--	--	--	---	---------	-------------	-------------

**Таблица 66**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..3	-	Зарезервировано
2	HSE RDY	Флаг выхода в рабочий режим осциллятора HSE 0 – осциллятор не запущен или не стабилен 1 – осциллятор запущен и стабилен
1	PLL CPU RDY	Флаг выхода в рабочий режим CPU PLL 0 – PLL не запущена или не стабильна 1 – PLL запущена и стабильна
0	PLL USB RDY	Флаг выхода в рабочий режим USB PLL 0 – PLL не запущена или не стабильна 1 – PLL запущена и стабильна

### PLL\_CONTROL

Номер	31	-	11..8	7..4	3	2	1	0
Доступ	U	U	R/W	U	R/W	R/W	U	U
Сброс	0	0	0000	0000	0	0	0	0
	-	-	PLL CPU MUL[3:0]	-	PLL CPU PLD	PLL CPU ON	-	-

**Таблица 67**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..12	-	Зарезервировано
11..8	PLL CPU MUL[3:0]	Коэффициент умножения для CPU PLL: $PLL_{CPU0} = PLL_{CPUi} \times (PLL_{CPUMUL} + 1)$
7..4	-	Зарезервировано
3	PLL CPU PLD	Бит перезапуска PLL При смене коэффициента умножения в рабочем режиме необходимо задать равным 1
2	PLL CPU ON	Бит включения PLL 0 – PLL выключена 1 – PLL включена
1	-	Зарезервировано
0	-	Зарезервировано

## HS\_CONTROL

Номер	31				2	1	0
Доступ	U				U	R/W	R/W
Сброс	0				0	0	0
	-	-			-	HSE BYP	HSE ON

**Таблица 68**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..2	-	Зарезервировано
1	HSE BYP	Бит управления HSE осциллятором 0 – режим осциллятора 1 – режим внешнего генератора
0	HSE ON	Бит управления HSE осциллятором 0 – выключен 1 – включен

## CPU\_CLOCK

Номер	31	10	9...8	7...4	3	2	1...0	
Доступ	U	U	R/W	R/W	U	R/W	R/W	
Сброс	0	0	00	0000	0	0	00	
	-	-	-	HCLK SEL[1:0]	CPU C3 SEL[3:0]	-	CPU C2 SEL	CPU C1 SEL[1:0]

**Таблица 69**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..10	-	Зарезервировано
9...8	HCLK SEL[1:0]	Биты выбора источника для HCLK 00 – HSI 01 – CPU_C3 10 – LSE 11 – LSI
7...4	CPU C3 SEL[3:0]	Биты выбора делителя для CPU_C3 0xxx – CPU_C3 = CPU_C2; 1000 - CPU_C3 = CPU_C2 / 2; 1001 - CPU_C3 = CPU_C2 / 4;

		1010 - CPU_C3 = CPU_C2 / 8; ... 1111 - CPU_C3 = CPU_C2 / 256;
3	-	Зарезервировано
2	CPU C2 SEL	Биты выбора источника для CPU_C2 0 – CPU_C1 1 – PLLCPUo
1...0	CPU C1 SEL[1:0]	Биты выбора источника для CPU_C1 00 – HSI 01 – HSI/2 10 – HSE 11 – HSE/2

### PER1\_CLOCK

Номер	13...12	11...10	9...8	7...6	5	4	2...0	1...0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	00	00	00	00	1	1	00	00
	SSP C2 SEL	UART2 C2 SEL	UART1 C2 SEL	TIM2 C2 SEL	DMA_ EN	DEBUG _EN	TIM1 C2 SEL	PER C1 SEL[1:0]

**Таблица 70**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..14	-	Зарезервировано
13..12	SSP C2 SEL	Биты выбора источника для SSP_C2 00 – CPU_C1 01 – PER1_C1 10 – PLLCPUo 11 – HSI_CLK
11..10	UART2 C2 SEL	Биты выбора источника для UART2_C2 00 – CPU_C1 01 – PER1_C1 10 – PLLCPUo 11 – HSI_CLK
9...8	UART1 C2 SEL	Биты выбора источника для UART1_C2 00 – CPU_C1 01 – PER1_C1 10 – PLLCPUo 11 – HSI_CLK
7...6	TIM2 C2 SEL	Биты выбора источника для TIM2_C2 00 – CPU_C1 01 – PER1_C1 10 – PLLCPUo

		11 – HSI_CLK
5	DMA_EN	Бит разрешения тактирования DMA контроллера
4	DEBUG_EN	Бит разрешения тактирования блока отладки ядра
3...2	TIM1 C2 SEL	Биты выбора источника для TIM1_C2 00 – CPU_C1 01 – PER1_C1 10 – PLLCPUo 11 – HSI_CLK
1...0	PER1 C1 SEL[1:0]	Биты выбора источника для PER1_C1 00 – LSI 01 – LSI/2 10 – LSE 11 – LSE/2

### ADC\_CLOCK

Номер	31...14	13	12	11...8	7...6	5...4	3...2	1...0
Доступ	U	R/W	R/W	R/W	U	R/W	U	R/W
Сброс	0	0	0	0000	00	00	00	00
	-	ADC CLK EN	ADCIU CLK EN	ADC C3 SEL[3:0]	-	ADC C2 SEL[1:0]	-	ADC C1 SEL[1:0]

**Таблица 71**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..14	-	Зарезервировано
13	ADC CLK EN	Бит разрешения выдачи тактовой частоты ADC CLK 0 – запрещен 1 – разрешен
12	ADCIU CLK EN	Бит разрешения выдачи тактовой частоты ADCIU CLK 0 – запрещен 1 – разрешен
11...8	ADC C3 SEL[3:0]	Биты выбора делителя для ADC_C3 0xxx – ADC_C3 = ADC_C1; 1000 - ADC_C3 = ADC_C1 / 2; 1001 - ADC_C3 = ADC_C1 / 4; 1010 - ADC_C3 = ADC_C1 / 8; ... 1111 - ADC_C3 = ADC_C1 / 256;
7...4	ADCIU C3 SEL[3:0]	Биты выбора делителя для ADCIU_C3 0xxx – ADCIU_C3 = ADCIU_C1; 1000 - ADCIU_C3 = ADCIU_C1 / 2; 1001 - ADCIU_C3 = ADCIU_C1 / 4; 1010 - ADCIU_C3 = ADCIU_C1 / 8;



		...
		1111 - ADCIU_C3 = ADCIU_C1 / 256;
3...2	ADCIU C1 SEL[1:0]	Биты выбора источника для ADCIU_C1 00 – CPU_C1 01 – PER1_C1 10 – PLLCPUo 11 – HSI_CLK
1...0	ADC C1 SEL[1:0]	Биты выбора источника для ADC_C1 00 – CPU_C1 01 – PER1_C1 10 – PLLCPUo 11 – HSI_CLK

### RTC\_CLOCK

Номер	31...14	10	9	8	7...4	3...0	
Доступ	U	U	R/W	R/W	R/W	R/W	
Сброс	0	0	0	0	0000	0000	
	-			HSI RTC EN	HSE RTC EN	HSI SEL[1:0]	HSE SEL[1:0]

**Таблица 72**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..10	-	Зарезервировано
9	HSI RTC EN	Бит разрешения HSI RTC 0 – запрещен 1 - разрешен
8	HSE RTC EN	Бит разрешения HSE RTC 0 – запрещен 1 - разрешен
7...4	HSI SEL[3:0]	Биты выбора делителя для HSI_C1 0xxx – HSI_C1 = HSI; 1000 - HSI_C1 = HSI / 2; 1001 - HSI_C1 = HSI / 4; 1010 - HSI_C1 = HSI / 8; ... 1111 - HSI_C1 = HSI / 256;
3...0	HSE SEL[3:0]	Биты выбора делителя для HSE_C1 0xxx – HSE_C1 = HSE; 1000 - HSE_C1 = HSE / 2; 1001 - HSE_C1 = HSE / 4; 1010 - HSE_C1 = HSE / 8; ... 1111 - HSE_C1 = HSE / 256;

### PER2\_CLOCK

Номер	18	0
Доступ	R/W	R/W
Сброс	0	0

PCLK\_EN[18:0]

**Таблица 73**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
18...0	PCLK_EN[18:0]	Биты разрешения тактирования периферийных блоков 0 – запрещено 1 – разрешено PCLK[0] – SPI PCLK[1] – UART1 PCLK[2] – UART2 PCLK[3] – EEPROM PCLK[4] – RST_CLK PCLK[6] - I2C PCLK[7] - LCD PCLK[8] - ADC PCLK[9] - WWDG PCLK[10] - IWDG PCLK[11] - POWER PCLK[12] - BKP PCLK[13] - ADCIU PCLK[14] - TIMER1 PCLK[15] - TIMER2 PCLK[16] - PORTA PCLK[17] - PORTB PCLK[18] - PORTC

### TIM\_CLOCK

Номер	31	26	25	24	23...16	15...0	7...0
Доступ	U	U	R/W	R/W	U	R/W	R/W
Сброс	0	0	0	0	0	00000000	00000000

-	-	-	TIM2 CLK EN	TIM1 CLK EN	-	TIM2 BRG [7:0]	TIM1 BRG [7:0]
---	---	---	-------------------	-------------------	---	----------------------	----------------------

**Таблица 74**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..26	-	Зарезервировано
25	TIM2 CLK EN	Разрешение тактовой частоты на TIM2 0 – нет частоты 1 – есть частота
24	TIM1 CLK EN	Разрешение тактовой частоты на TIM1 0 – нет частоты 1 – есть частота
23..16	-	Зарезервировано
15...8	TIM2 BRG [7:0]	Делитель тактовой частоты TIM2  xxxxx000 – TIM2_CLK == PER1_C2 xxxxx001 – TIM2_CLK == PER1_C2/2 xxxxx010 – TIM2_CLK == PER1_C2/4 ... xxxxx111 – TIM2_CLK == PER1_C2/128
7...0	TIM1 BRG [7:0]	Делитель тактовой частоты TIM1  xxxxx000 – TIM1_CLK == PER1_C2 xxxxx001 – TIM1_CLK == PER1_C2/2 xxxxx010 – TIM1_CLK == PER1_C2/4 ... xxxxx111 – TIM1_CLK == PER1_C2/128

## UART\_CLOCK

Номер	31	26	25	24	23...16	15...0	7...0
Доступ	U	U	R/W	R/W	U	R/W	R/W
Сброс	0	0	0	0	0	00000000	00000000
	-	-	UART2 CLK EN	UART 1 CLK EN	-	UART 2 BRG [7:0]	UART 1 BRG [7:0]

**Таблица 75**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..27	-	Зарезервировано
26	-	Зарезервировано
25	UART2 CLK EN	Разрешение тактовой частоты на UART2 0 – нет частоты 1 – есть частота
24	UART1 CLK EN	Разрешение тактовой частоты на UART 1 0 – нет частоты 1 – есть частота
23..16	-	Зарезервировано
15...8	UART2 BRG [7:0]	Делитель тактовой частоты UART 2  xxxxx000 – UART 2_CLK == PER1_C2 xxxxx001 – UART 2_CLK == PER1_C2/2

		xxxxx010 – UART 2_CLK == PER1_C2/4 ... xxxxx111 – UART 2_CLK == PER1_C2/128
7...0	UART1 BRG [7:0]	Делитель тактовой частоты UART1  xxxxx000 – UART 1_CLK == PER1_C2 xxxxx001 – UART 1_CLK == PER1_C2/2 xxxxx010 – UART 1_CLK == PER1_C2/4 ... xxxxx111 – UART 1_CLK == PER1_C2/128

### SSP\_CLOCK

Номер	31	26	25	24	23...16	15...0	7...0	
Доступ	U	U	U	R/W	U	U	R/W	
Сброс	0	0	0	0	0	0	00000000	
	-	-	-	-	SSP 1 CLK EN	-	-	SSP 1 BRG [7:0]

**Таблица 76**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..25	-	Зарезервировано
26	-	Зарезервировано
24	SSP1 CLK EN	Разрешение тактовой частоты на SSP 0 – нет частоты 1 – есть частота
23..8	-	Зарезервировано
7...0	SSP1 BRG [7:0]	Делитель тактовой частоты SSP  xxxxx000 – SSP 1_CLK == PER1_C2 xxxxx001 – SSP 1_CLK == PER1_C2/2 xxxxx010 – SSP 1_CLK == PER1_C2/4 ... xxxxx111 – SSP 1_CLK == PER1_C2/128

### Батарейный домен и часы реального времени

Блок батарейного домена предназначен для обеспечения функций часов реального времени и сохранения некоторого набора пользовательских данных при отключении основного источника питания. Так же в батарейном домене реализована функция контроля выхода COV\_DET (см. Таблица 1). Это позволяет, даже в отсутствии основного питания определять его состояние. При снижении питания U<sub>cc</sub> в блоке SW происходит автоматическое переключение питания V<sub>DUcc</sub> с U<sub>cc</sub> на V<sub>Ucc</sub>. Если на V<sub>Ucc</sub> имеется отдельный источник питания (батарейка), то батарейный домен остается включенным и может выполнять свои функции.

Структурная схема:

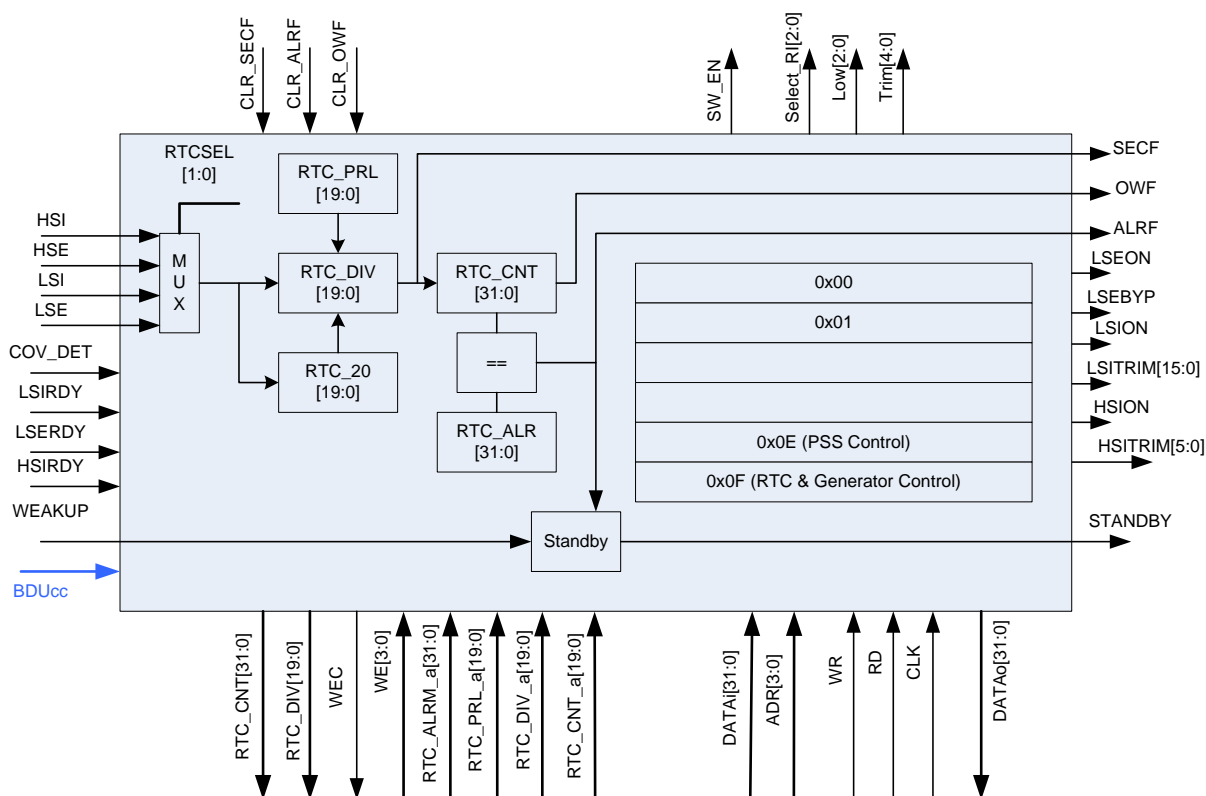


Рис. 29

### Часы реального времени

Часы реального времени позволяют организовать механизм отсчета времени в кристалле, в том числе при отключении основного источника питания. Включение часов реального времени осуществляется битом RTCEN (см. Таблица 80). В качестве источника тактовой частоты часов реального времени может выступать генератор LSI или осциллятор LSE или HSE, или HSI с дополнительным делителем до 256 (HSE и HSI формируются в блоке управления тактовыми частотами и могут быть выбраны только при наличии питания V<sub>DUcc</sub> (на внутреннем регуляторе), LSI может быть выбран при наличии питания U<sub>cc</sub>, LSE может быть выбран при наличии U<sub>cc</sub> или V<sub>Ucc</sub>). Выбор между источниками осуществляется битами RTCSEL. При возможном отключении основного источника питания U<sub>cc</sub> в качестве источника тактовой частоты должен использоваться осциллятор LSE, так как он также имеет питание V<sub>DUcc</sub>. Биты

управления осциллятором LSE расположены в батарейном домене и таким образом при отключении основного питания они не сбрасываются. При этом при первоначальном включении эти биты так же не определены и могут принять любое значение.

Для калибровки тактовой частоты используются биты CAL[7:0]. Значение CAL определяет какое число тактов из  $2^{20}$  будет замаскировано. Таким образом, с помощью битов CAL производится замедление хода часов. Изменение значения битов CAL может быть осуществлено в ходе работы часов реального времени. Регистр RTC\_DIV (см. Таблица 77) выступает в роли 20-ти битного предварительного делителя входной тактовой частоты, таким образом, чтобы на его выходе была тактовая частота в 1 Гц. Для задания коэффициента деления регистра RTC\_DIV используется регистр RTC\_PRL.

Регистр RTC\_CNT предназначен для отсчета времени в секундах. И работает на выходной частоте делителя RTC\_DIV. Регистр RTC\_ALR предназначен для задания времени, при совпадении с которым вырабатывается флаг прерывания и пробуждения процессора. Таким образом, бит STANBY, отключающий внутренний регулятор напряжения автоматически сбрасывается при совпадении RTC\_CNT и RTC\_ALR.

Бит STANDBY (см. Таблица 80) так же может быть сброшен с помощью вывода WAKEUP.

В батарейном домене реализована возможность мониторинга входного сигнала на COV\_DET. Во внутреннем регистре записывается контролируемый уровень ("0" или "1"), и если сигнал на входе станет отличным от записанного, то это событие регистрируется в управляющем бите.

### **Регистры аварийного сохранения**

Батарейный домен имеет 16 встроенных 32-х разрядных регистров аварийного сохранения. 16-тый регистр служит для хранения битов управления батарейным доменом, оставшиеся 15 регистров могут быть использованы разработчиком программы.

### **Описание регистров блока батарейного домена**

**Таблица 77**

Базовый Адрес	Название	Описание
0x4006_0000	ВКР	Контроллер батарейного домена и часов реального времени.
Смещение		
0x00	ВКР_REG_00	Регистр аварийного сохранения 0
...		
0x38	ВКР_REG_0E	Регистр аварийного сохранения 14
0x3C	ВКР_REG_0F	Регистр аварийного сохранения 15 и управления блоками RTC, LSE, LSI и HSI
0x40	RTC_CNT	Регистр основного счетчика часов реального времени
0x44	RTC_DIV	Регистр предварительного делителя основного счетчика
0x48	RTC_PRL	Регистр основания счета

		предварительного делителя
0x4C	RTC_ALRM	Регистр значения для сравнения основного счетчика и выработки сигнала ALRF
0x50	RTC_CS	Регистр управления и состояния флагов часов реального времени

BKP\_REG\_00  
 BKP\_REG\_01  
 BKP\_REG\_02  
 BKP\_REG\_03  
 BKP\_REG\_04  
 BKP\_REG\_05  
 BKP\_REG\_06  
 BKP\_REG\_07  
 BKP\_REG\_08  
 BKP\_REG\_09  
 BKP\_REG\_0A  
 BKP\_REG\_0B  
 BKP\_REG\_0C  
 BKP\_REG\_0D

Номер	31	0
Доступ	R/W	R/W
Сброс	0	0
BKP REG[31:0]		

**Таблица 78**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31...0	BKP REG[31:0]	Регистр аварийного сохранения

### BKP\_REG\_0E

Номер	15	14	13..12	11	10...8	7	6	5...3	2...0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	00	0	000	0	0	000	000
	ilimen	COV DET	Trim [4:3]	FPOR	Trim[2:0]	JTAG_B	JTAG_A	SelectRI [2:0]	LOW [2:0]

**Таблица 79**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
---	-------------------------	---

## Спецификация K1986BK234, K1986BK234K

31..16	-	Зарезервировано
15	ilimen	Бит разрешения защиты ограничения регулятора по току 150 мА
14	COVDET	Признак несанкционированного вскрытия устройства 1-вскрытия не было 0-осуществлялось вскрытие
13..12	Trim[4:3]	Коэффициент настройки опорного напряжения регулятора 00-1.8 В 01-1.6 В 10-1.4 В 11-1.2 В
11	FPOR	Флаг срабатывания POR Устанавливается в 1 загрузочным ПЗУ после сброса по питанию, при сбросе по питанию устанавливается в 0. Служит для анализа загрузочным ПЗУ, что сейчас идет выполнение программы после системного или программного сброса, либо после сброса по питанию.
10...8	Trim[2:0]	Коэффициент настройки опорного напряжения встроенного регулятора напряжения DUcc. С помощью Trim осуществляется подстройка напряжения DUcc 000 – DUcc + 0,10В – значение по умолчанию. 001 – DUcc + 0,06В 010 – DUcc + 0,04В 011 – DUcc + 0,01В 100 – DUcc – 0,01В 101 – DUcc – 0,04В 110 – DUcc – 0,06В 111 – DUcc – 0,10В
7	JTAG B	Разрешение работы порта JTAG B 0 – запрещен 1 – разрешен
6	JTAG A	Разрешение работы порта JTAG A 0 – запрещен 1 – разрешен
5...3	SelectRI[2:0]	Выбор дополнительной стабилизирующей нагрузки для встроенного регулятора напряжения DUcc. 000 – ~6 КОм (дополнительный ток потребления 300 мкА) 001 – ~270 КОм (дополнительный ток потребления 6,6 мкА) 010 – ~90 КОм (дополнительный ток потребления 20 мкА) 011 – ~24 КОм (дополнительный ток потребления 80 мкА) 100 – ~900 КОм (собственное потребление 2 мкА) 101 – ~2 КОм (дополнительный ток потребления 900 мкА)



		110 – ~400 Ом (дополнительный ток потребления 4,4 мА) 111 – ~100 Ом (дополнительный ток потребления 19 мА)
2...0	LOW[2:0]	Выбор режима работы встроенного регулятора напряжения DUcc. Значение LOW должно совпадать с значением SelectRI и выставляться в зависимости от тактовой частоты микроконтроллера 000 – Частота до 10 МГц 001 – Частота до 200 КГц 010 – Частота до 500 КГц 011 – Частота до 1 МГц 100 – При выключении всех генераторов 101 – Частота до 40 МГц 110 – Частота до 80 МГц 111 – Частота более 80 МГц

### BKP\_REG\_0F

Номер	15	14	13	12...5	4	3.2	1	0
Доступ	R/W	U	RO	R/W	R/W	R/W	R/W	R/W
Сброс	1	0	0	0000000	0	00	0	0
	LSI ON	-	LSE RDY	CAL[7:0]	RTC EN	RTC SEL[1:0]	LSE BYP	LSE ON
Номер	31	30	29...24	23	22	21	20...16	
Доступ	R/W	R/W	R/W	R/W	R/W	RO	R/W	
Сброс	0	0	0000	0	0	0	0000	
	RTC RESET	STANDBY	HSI TRIM [5:0]	HSI RDY	HSI ON	LSI RDY	LSI TRIM [4:0]	

**Таблица 80**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31	RTC RESET	Сброс часов реального времени 0 – часы не сбрасываются 1 – часы сбрасываются
30	STANDBY	Режим отключения регулятора DUcc на 1.8В 0 – регулятор включен и выдает напряжение Запись 1 – выключение регулятора Триггер сбрасывается по событию ALRF или по низкому уровню на выводе WAKEUP.
29..24	HSI TRIM[5:0]	Коэффициент подстройки частоты генератора HSI Смотри диаграмму зависимости (Рис. 31).
23	HSI RDY	Флаг выхода генератора HSI в рабочий режим 0 – генератор не запущен или не вышел в режим 1 – генератор работает в рабочем режиме
22	HSI ON	Бит управления генератором HSI 0 – генератор выключен

		1 – генератор включен
21	LSI RDY	Флаг выхода генератора LSI в рабочий режим 0 – генератор не запущен или не вышел в режим 1 – генератор работает в рабочем режиме
20..16	LSI TRIM[4:0]	Коэффициент подстройки частоты генератора LSI Смотри диаграмму зависимости (Рис. 30).
15	LSI ON	Бит управления генератором LSI 0 – генератор выключен 1 – генератор включен
14	-	Зарезервировано
13	LSE RDY	Флаг выхода генератора LSE в рабочий режим 0 – генератор не запущен или не вышел в режим 1 – генератор работает в рабочем режиме
12..5	CAL[7:0]	Коэффициент подстройки тактовой частоты часов реального времени, из каждых $2^{20}$ тактов будет замаскировано CAL тактов. 00000000 – 0 тактов 00000001 – 1 такт .... 11111111 – 256 тактов Таким образом, при частоте 32768.00000 Гц при CAL = 0 тактов, частота = 32768.00000 Гц при CAL = 1 такт, частота = 32767,96875 Гц; ... при CAL = 256 тактов, частота = 32760.00000 Гц;
4	RTC EN	Бит разрешения работы часов реального времени 0 – работа запрещена 1 – работа разрешена
3...2	RTC SEL[1:0]	Биты выбора источника тактовой синхронизации часов реального времени 00 – LSI 01 – LSE 10 – HSIRTC (формируется в блоке CLKRST) 11 – HSERTC (формируется в блоке CLKRST)
1	LSE BYP	Бит управления генератором LSE 0 – режим осциллятора 1 – режим работы на проход (внешний генератор)
0	LSE ON	Бит управления генератором LSE 0 – генератор выключен 1 – генератор включен

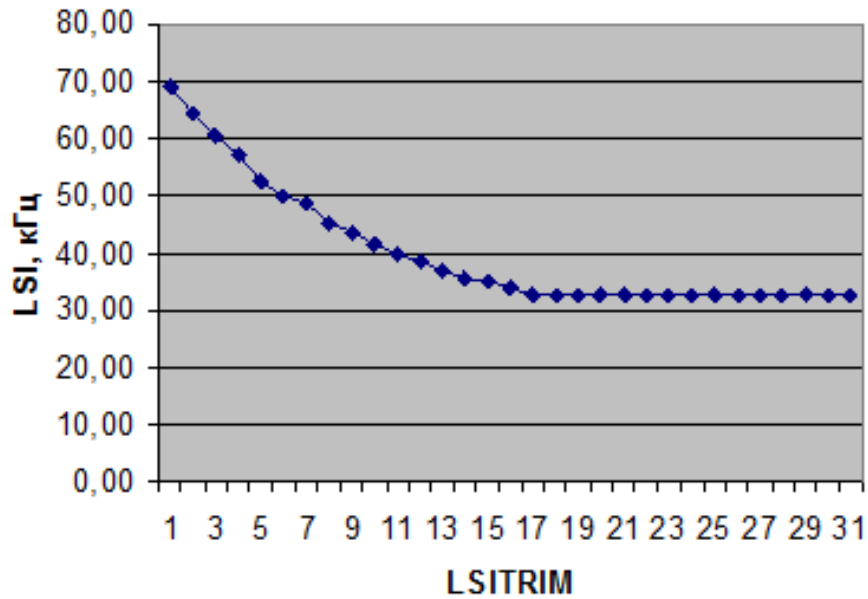


Рис. 30 Зависимость частоты LSI от значения LSITRIM

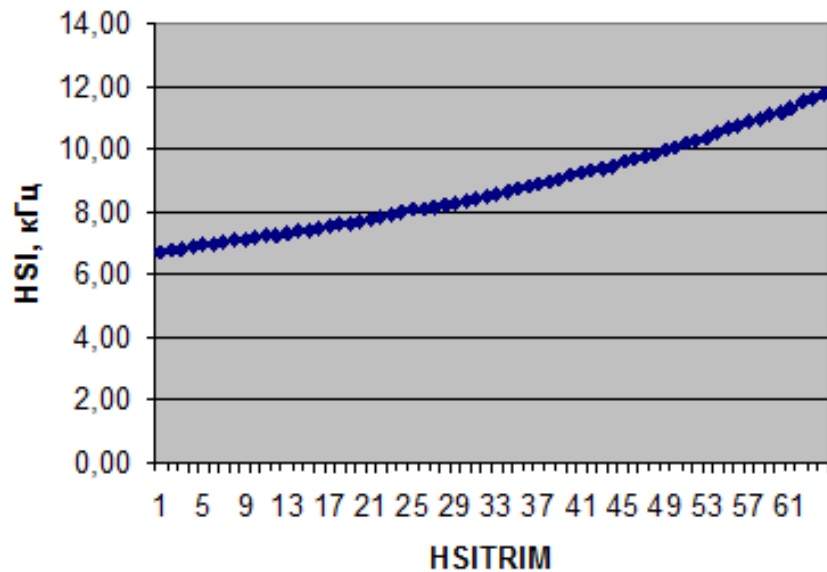


Рис. 31 Зависимость частоты HSI от значения HSITRIM

### RTC\_CNT

Номер	31	0
Доступ	R/W	R/W
Сброс	0	0

RTC CNT[31:0]
------------------

**Таблица 81**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	RTC CNT[31:0]	Значение основного счетчика часов реального времени

**RTC\_DIV**

Номер	31	20	19..0
Доступ	U	U	R/W
Сброс	0	0	0

-	-	-	-	-	-	-	RTC DIV [19:0]
---	---	---	---	---	---	---	----------------------

**Таблица 82**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	RTC DIV [19:0]	Значение счетчика предварительного делителя часов реального времени

**RTC\_PRL**

Номер	31	20	19..0
Доступ	U	U	R/W
Сброс	0	0	0

-	-	-	-	-	-	-	RTC PRL [19:0]
---	---	---	---	---	---	---	----------------------

**Таблица 83**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	RTC PRL [19:0]	Значение основания для счета счетчика предварительного делителя часов реального времени

**RTC\_ALRM**

Номер	31	0
Доступ	R/W	R/W
Сброс	0	0

RTC ALRM[31:0]
-------------------

**Таблица 84**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	RTC ALRM[31:0]	Значения для сравнения основного счетчика и выработки сигнала ALRF

**RTC\_CS**

Номер	31	6	5	4	2	1	0
Доступ	U	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0
	-	ALRF_IE	SECF_I E	OWF_IE	ALRF	SECF	OWF

**Таблица 85**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
30...7	-	Зарезервировано
6	WEC	Запись завершена 0 – можно записывать в регистры RTC 1 – идет запись в регистры RTC, запись в регистры запрещена.
5	ALRF_IE	Флаг разрешения прерывания по совпадению основного счетчика и регистра RTC_ALARM 0 – нет совпадения 1 – есть совпадение
4	SECF_IE	Флаг разрешения прерывания по разрешению счета основного счетчика от счетчика предварительного деления 0 – нет разрешения счета 1 – разрешение счета
3	OWF_IE	Флаг разрешения прерывания по переполнению основного счетчика RTC_CNT 0 – нет переполнения 1 – было переполнение
2	ALRF	Флаг совпадения основного счетчика и регистра RTC_ALARM 0 – нет совпадения 1 – есть совпадение
1	SECF	Флаг разрешения счета основного счетчика от счетчика предварительного деления 0 – нет разрешения счета 1 – разрешение счета
0	OWF	Флаг переполнения основного счетчика RTC_CNT 0 – нет переполнения 1 – было переполнение

**Порты ввода-вывода**

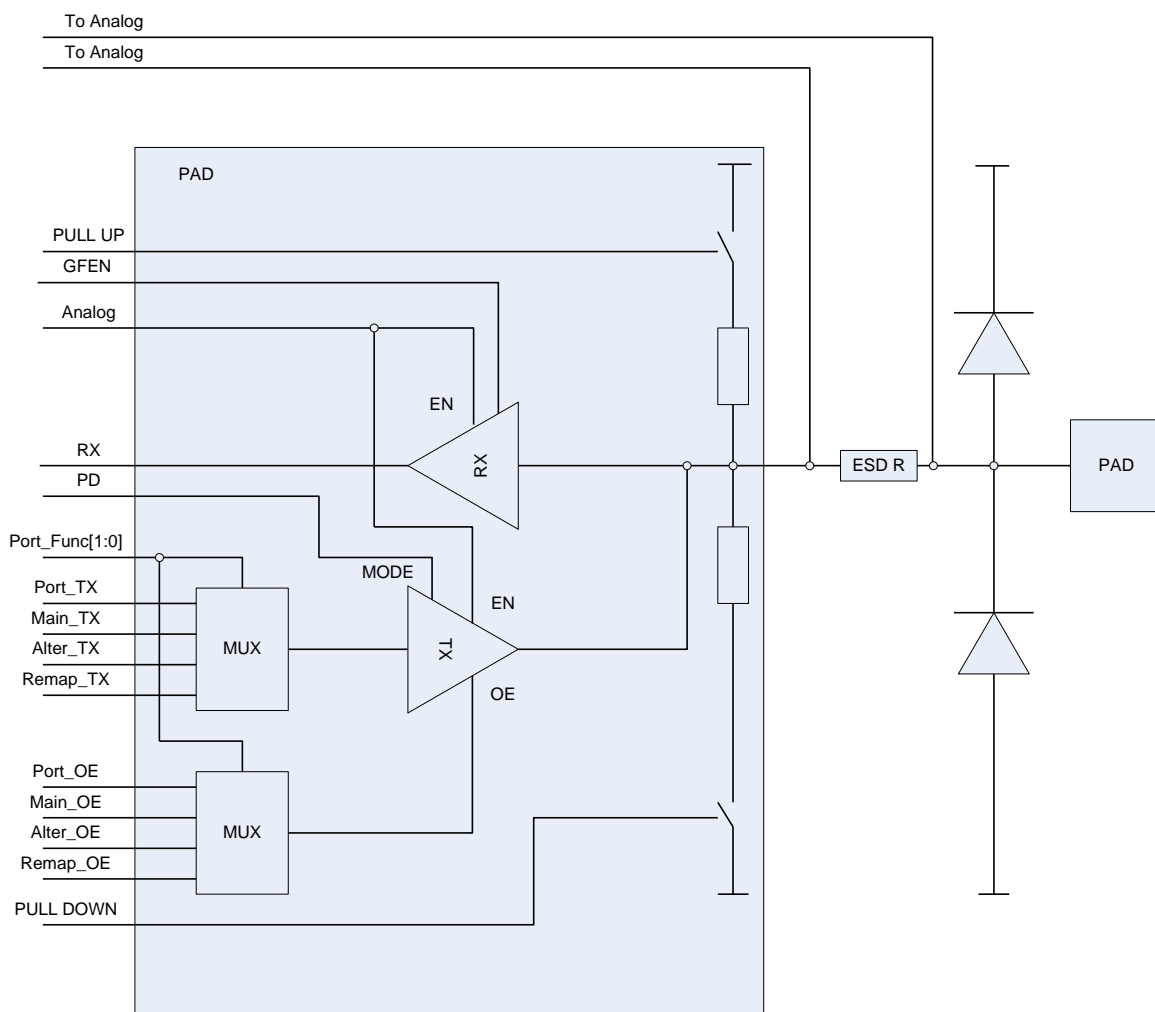
Микроконтроллер имеет 3 порта ввода/вывода – А, В, С. Порты 16-ти разрядные (кроме порта В) и их выходы мультиплексируются между различными функциональными блоками, управление для каждого вывода отдельное. Для того, что бы выходы порта перешли под управление того или иного периферийного блока необходимо задать для нужных выводов выполняемую функцию и настройки.

**Таблица 86**

Вывод	Аналоговая функция ANALOG_EN=0		Цифровая функция		
			Порт IO	Основная	
			MODE=0 ANALOG_EN=1	MODE=1 ANALOG_EN=1	
<b>Порт А</b>					
PA0	S0	10	PA0	TMR0_CH1	3
PA1	S1		PA1	TMR0_CH1N	
PA2	S2		PA2	-	
PA3	S3		PA3	-	
PA4	S4		PA4	-	
PA5	S5		PA5	-	
PA6	S6		PA6	-	
PA7	S7		PA7	-	
PA8	S8		PA8	TMR0_ETR	
PA9	S9		PA9	TMR0_BLK	
PA10	S10		PA10	-	
PA11	S11		PA11	-	
PA12	S12		PA12	-	
PA13	S13		PA13	-	
PA14	S14		PA14	-	
PA15	S15		PA15	-	
<b>Порт В</b>					
PB0	-		PB0 MODE0	UART0_TXD	4
PB1	-		PB1	UART0_RXD	
PB2	-		PB2 SWCLKTCK	nSIROUT0	
PB3	-		PB3 SWDIO	nSIRIN0	
PB4	OSC_IN32	1	PB4	nUART0DTR	5
PB5	OSC_OUT32		PB5	nUART0RTS	
PB6	ADC7S/BP0	2,10	PB6	nUART0RI	
PB7	ADC6S/BP1		PB7	nUART0DCD	
PB8	ADC5S/BP2		PB8	nUART0DSR	
PB9	ADC4S/BP3	PB9	nUART0CTS		
PB10	S16	10	PB10	-	
PB11	S17		PB11	-	
PB12	S18		PB12	-	
PB13	S19		PB13	-	
PB14	S20		PB14	-	
<b>Порт С</b>					

PC0	-		PC0	MODE1	UART1_TXD	6
PC1	ADC3S	2	PC1		UART1_RXD	5
PC2	ADC2S/VLCD		PC2		TMR1_CH1	
PC3	ADC1S/ADCS_REF+		PC3		TMR1_CH1N	
PC4	ADC0S/ADCS_REF-		PC4		-	
PC5	S21	10	PC5		TMR1_ETR	5
PC6	S22		PC6		TMR1_BLK	9
PC7	S23		PC7		EXT_INT0	
PC8	S24		PC8		EXT_INT1	
PC9	S25		PC9		EXT_INT2	7
PC10	S26		PC10		SSP_FSS	
PC11	S27		PC11		SSP_CLK	
PC12	S28	PC12		SSP_RXD		
PC13	S29		PC13		SSP_TXD	
PC14	-		PC14		SCL	8
PC15	-		PC15		SDA	

- 1 – Генератор LSE
- 2 – АЦП последовательного приближения
- 3 – Таймер 0
- 4 – UART0
- 5 – Таймер 1
- 6 – UART1
- 7 – Последовательный интерфейс SSP
- 8 – Последовательный интерфейс I2C
- 9 – Внешние прерывания
- 10 – Контроллер LCD



**Рис. 32**

**Описание регистров портов ввода-вывода**

**Таблица 87**

Базовый Адрес	Название	Описание
0x4008_0000	GPIO1	Порт А
0x4008_8000	GPIO2	Порт В
0x4009_0000	GPIO3	Порт С
<b>Смещение</b>		
0x00	PORT_RXTX[15:0]	Данные порта
0x04	PORT_OE[15:0]	Направление порта
0x08	PORT_FUNC[31:0]	Режим работы порта
0x0C	PORT_ANALOG[15:0]	Аналоговый режим работы порта
0x10	PORT_PULL[31:0]	Подтяжка порта
0x14	PORT_PD[31:0]	Режим работы выходного драйвера
0x18	PORT_PWR[31:0]	Режим мощности передатчика
0x1C	PORT_GFEN[15:0]	Режим работы входного



		фильтра
--	--	---------

**PORTx\_RXTX**

Номер	31	15	0
Доступ	U	R/W	R/W
Сброс	0	0	0
-			PORT RXTX[15:0]

**Таблица 88**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16	-	Зарезервировано
15...0	PORT RXTX[15:0]	Режим работы контроллера Данные для выдачи на выходы порта и для чтения.

**PORTx\_OE**

Номер	31	15	0
Доступ	U	R/W	R/W
Сброс	0	0	0
-			PORT OE[15:0]

**Таблица 89**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16	-	Зарезервировано
15...0	PORT OE[15:0]	Режим работы контроллера Направление передачи данных на выводах порта 1 – выход 0 - вход

**PORTx\_FUNC**

Номер	31	30	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	MODE15[1:0]			MODE1[1:0]		MODE0[1:0]

**Таблица 90**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..2	MODEx	Аналогично MODE0 для остальных битов порта
1...0	MODE0[1:0]	Режим работы вывода порта 00 – порт 01 – основная функция 10 – альтернативная функция 11 – переопределенная функция

### PORTx\_ANALOG

Номер	31	16	15	0
Доступ	U	R/W	R/W	R/W
Сброс	0	0	0	0
	-		ANALOG EN[15:0]	

**Таблица 91**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16		
15...0	ANALOG EN[15:0]	Режим работы контроллера 0 – аналоговый 1 – цифровой

### PORTx\_PULL

Номер	31	16	15	0
Доступ	U	R/W	R/W	R/W
Сброс	0	0	0	0
	PULL UP[15:0]		PULL DOWN[15:0]	

**Таблица 92**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16	PULL UP[15:0]	Режим работы контроллера Разрешение подтяжки вверх 0 – подтяжка в питание выключена 1 – подтяжка в питание включена (есть подтяжка)
15...0	PULL DOWN[15:0]	Режим работы контроллера Разрешение подтяжки вниз 1 – подтяжка в ноль включена (есть подтяжка)

		0 – подтяжка в ноль выключена
--	--	-------------------------------

**PORTx\_PD**

Номер	31	16	15	0
Доступ	U	R/W	R/W	R/W
Сброс	0	0	0	0

PORT SHM[15:0]	PORT PD[15:0]
----------------	---------------

**Таблица 93**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16	PORT SHM[15:0]	Режим работы контроллера Режим работы входа 0 – триггер Шмита выключен, гистерезис 200 мВ. 1 – триггер Шмита включен, гистерезис 400 мВ.
15...0	PORT PD[15:0]	Режим работы контроллера Режим работы выхода 0 – управляемый драйвер 1 – открытый сток

**PORTx\_PWR**

Номер	31	30	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0

PWR15[1:0]			PWR1[1:0]	PWR0[1:0]
------------	--	--	-----------	-----------

**Таблица 94**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..2	PWRx	Аналогично PWR0 для остальных битов порта
1...0	PWR0[1:0]	Режим работы вывода порта 00 – зарезервировано 01 – медленный фронт 10 – быстрый фронт 11 – максимально быстрый фронт

**PORTx\_GFEN**

Номер	31	16	15	0
-------	----	----	----	---

Доступ	U	R/W	R/W	R/W
Сброс	0	0	0	0
	-		GFEN[15:0]	

**Таблица 95**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16		
15...0	GFEN[15:0]	Режим работы входного фильтра 0 – фильтр выключен 1 – фильтр включен

**Детектор напряжения питания**

Блок детектора напряжения питания PVD предназначен для контроля питания Ucc и BUcc при работе микроконтроллера. Блок PVD позволяет сравнивать внешние уровни напряжения с внутренними опорными уровнями и в случае превышения или снижения ниже опорного уровня выработать сигнал или прерывание для программной обработки.

Уровень опорного напряжения для сравнения с Ucc задается битами PLS[2:0] в регистре PVDCS, для сравнения с BUcc задается битами PLBS[1:0] в регистре PVDCS. В соответствии с уровнями напряжения формируются флаги PVD и PBVD. Данные флаги выставляются при возникновении события и сбрасываются программно.

**Таблица 96**

Параметр	Не менее	Типовое	Не более
Входное напряжение, Ucc, В	2,0	-	3,6
Входное напряжение, BUcc, В	1,8	-	3,6
Уровень срабатывания PVD от Ucc, при PLS = "000", В		2,0	
Уровень срабатывания PVD от Ucc, при PLS = "001", В		2,2	
Уровень срабатывания PVD от Ucc, при PLS = "010", В		2,4	
Уровень срабатывания PVD от Ucc, при PLS = "011", В		2,6	
Уровень срабатывания PVD от Ucc, при PLS = "100", В		2,8	
Уровень срабатывания PVD от Ucc, при PLS = "101", В		3,0	
Уровень срабатывания PVD от Ucc, при PLS = "110", В		3,2	
Уровень срабатывания PVD от Ucc, при PLS = "111", В		3,4	
Уровень срабатывания PBVD от BUcc, при PBLBS = "00", В		1,8	
Уровень срабатывания PBVD от BUcc, при PBLBS = "01", В		2,2	
Уровень срабатывания PBVD от BUcc, при PBLBS = "10", В		2,6	
Уровень срабатывания PBVD от BUcc, при PBLBS = "11", В		3,0	

**Описание регистров блока PVD**

**Таблица 97**

Базовый Адрес	Название	Описание
0x4005_8000	POWER	Датчик подсистемы питания
Смещение		
0x00	PVDCS [12:0]	Регистр управления и состояния датчика питания

**PVDCS**

Номер	12	11	10	9	8...5	4...1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	000	00	0
	IEPVD	IEPVBD	PVD	PVBD	PLS [2:0]	PBLS [1:0]	PVD EN
Номер	31				15	14	13
Доступ	U				R/W	R/W	R/W
Сброс	0				0	0	0
	-	-	-	-	PVDB EN	INV	INVB

**Таблица 98**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16	-	Зарезервировано
15	PVDBEN	Бит разрешения работы блока датчика напряжения питания ВUсс 0 – датчик отключен 1 – датчик включен
14	INV	Флаг инверсии выхода от датчика PVD 0 – нет инверсии 1 – есть инверсия Если флаг не инвертируется, то флаг выставляется при превышении заданного уровня, если инвертируется, то при снижении ниже заданного уровня
13	INVB	Флаг инверсии выхода от датчика PVBD 0 – нет инверсии 1 – есть инверсия Если флаг не инвертируется, то флаг выставляется при превышении заданного уровня, если инвертируется, то при снижении ниже заданного уровня
12	IEPVD	Флаг разрешения прерывания от датчика PVD 0 – прерывание запрещено 1 – прерывание разрешено Очищается записью 0, если при очистке, датчик продолжает выдавать сигнал, то флаг не будет очищен.
11	IEPVBD	Флаг разрешения прерывания от датчика PVBD 0 – прерывание запрещено 1 – прерывание разрешено

## Спецификация K1986BK234, K1986BK234K

		Очищается записью 0, если при очистке, датчик продолжает выдавать сигнал, то флаг не будет очищен.
10	PVD	Результат сравнения напряжения основного питания 0 – напряжение питания меньше чем уровень задаваемый PLS 1 – напряжение питания больше чем уровень задаваемый PLS
9	PVBD	Результат сравнения напряжения батарейного питания 0 – напряжение питания меньше чем уровень задаваемый PBLIS 1 – напряжение питания больше чем уровень задаваемый PBLIS
8...5	PLS[3:0]	Уровень напряжения для сравнения с напряжением основного питания 0000 – 1,0В 0001 – 1,2В 0010 – 1,4В 0011 – 1,6В 0100 – 1,8В 0101 – 2,0В 0110 – 2,2В 0111 – 2,4В 1000 – 2,6В 1001 – 2,8В 1010 – 3,0В 1011 – 3,2В 1100 – 3,4В 1101 – 3,6В 1110 – 3,8В 1111 – 4,0В
4...1	PBLIS[3:0]	Уровень напряжения для сравнения с напряжением батарейного питания 0000 – 1,0В 0001 – 1,2В 0010 – 1,4В 0011 – 1,6В 0100 – 1,8В 0101 – 2,0В 0110 – 2,2В 0111 – 2,4В 1000 – 2,6В 1001 – 2,8В 1010 – 3,0В 1011 – 3,2В 1100 – 3,4В 1101 – 3,6В 1110 – 3,8В 1111 – 4,0В
0	PVDEN	Бит разрешения работы блока датчика напряжения

## Спецификация К1986ВК234, К1986ВК234К

---

		питания Uсс 0 – датчик отключен 1 – датчик включен
--	--	--



### Таймеры общего назначения

Все блоки таймеров выполнены на основе 16-битного перезагружаемого счетчика, который синхронизируется с выхода 16-битного предделителя. Перезагружаемое значение хранится в отдельном регистре. Счет может быть прямой, обратный или двунаправленный (сначала прямой до определенного значения, а затем обратный).

Каждый из двух таймеров микроконтроллера содержит 16-битный счетчик, 16-битный предделитель частоты и 1-канальный блок захвата/сравнения. Их можно синхронизировать системной синхронизацией, внешними сигналами или другими таймерами.

Помимо составляющего основу таймера счетчика, в каждый блок таймера также входит одноканальный блок захвата/сравнения. Данный блок выполняет, как стандартные функции захвата и сравнения, так и ряд специальных функций. Таймеры с одним каналом схем захвата и ШИМ с функциями формирования «мертвой зоны» и аппаратной блокировки. Каждый из таймеров может генерировать прерывания.

### Особенности

- 16-битный вверх, вниз, вверх/вниз счетчик.
- 16-разрядный программируемый предварительный делитель частоты.
- один 16-битный канал захвата на один таймер. Канало захвата может захватить (скопировать) текущее значение таймера при изменении некоторого входного сигнала.
- четыре 16-битных регистра сравнения (совпадения), которые позволяют:
  - осуществлять непрерывное сравнение, с дополнительной возможностью генерировать прерывание при совпадении;
- имеется один внешний вывод, соответствующий регистрам совпадения со следующими возможностями:
  - сброс в НИЗКИЙ уровень при совпадении;
  - установка в ВЫСОКИЙ уровень при совпадении;
  - переключение (инвертирование) при совпадении;
  - при совпадении состояние выхода не изменяется;
  - переключение при некотором условии;

### Функционирование

Таймер предназначен для того, чтобы подсчитывать циклы периферийной тактовой частоты  $F_{dts}$  или какие-либо внешние события и произвольно генерировать прерывания или выполнять другие действия. Значения таймера, при достижении которых будут выполнены те или иные действия, задаются двумя регистрами совпадения. Кроме того, в микроконтроллере имеется один вход захвата, чтобы захватить значение таймера при изменении некоторого входного сигнала, с возможностью генерировать прерывание.

### Структурная схема

Структурная схема блока таймера представлена на Рис. 33.

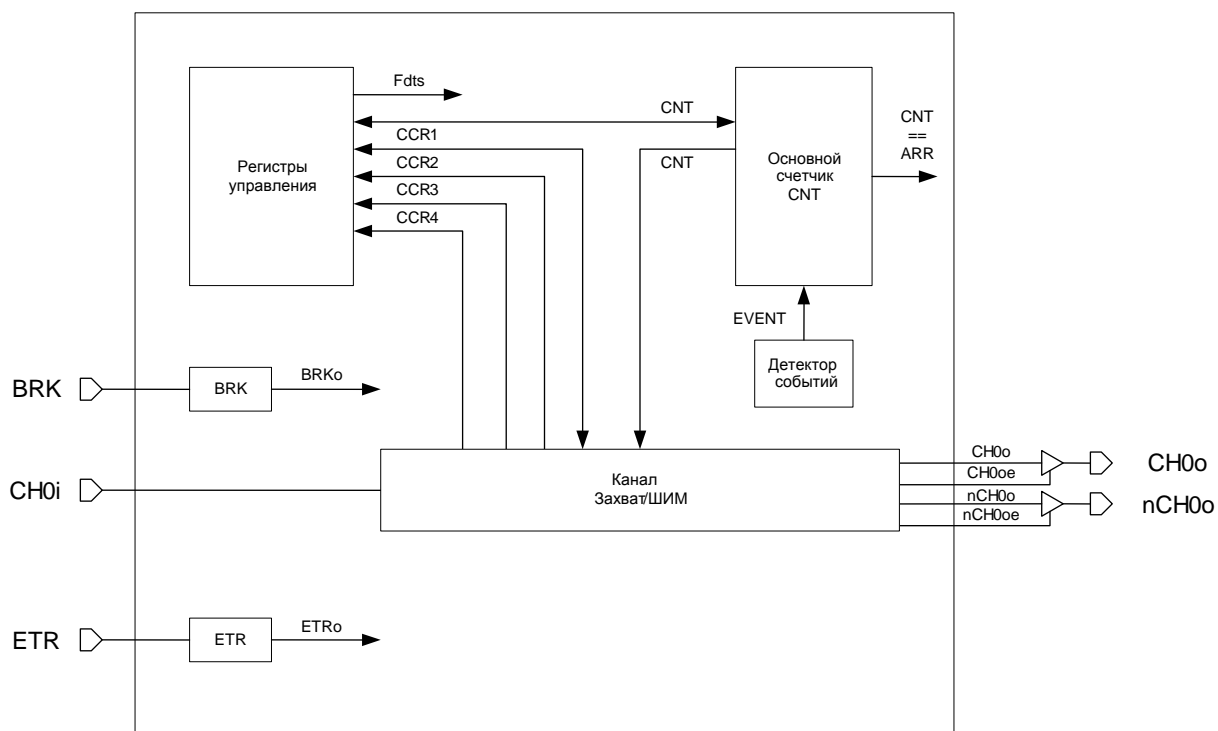


Рис. 33 Структурная схема таймера

Таймер содержит основной 16-ти битный счетчик CNT, блок регистров управления и один канал схемы Захвата/ШИМ.

Таймер позволяет работать в режимах:

- таймер
- расширенный таймер, с объединением нескольких таймеров
- Схема захвата
- Схема ШИМ

### Инициализация таймера

Перед началом работы с таймерами в первую очередь должны быть включены тактовые сигналы. Параметры задаются в блоке «Сигналы тактовой частоты».

Для задания тактовой частоты блока необходимо установить бит разрешения тактирования блока (бит 14 для таймера 1, бит 15 для таймера 2 регистра PER2\_CLOCK). В регистре TIM\_CLOCK (см. раздел «Внутренний тактовый сигнал (TIM\_CLK)») установить бит TIMyCLKEN чтобы разрешить тактовую частоту для определенного таймера, задать коэффициент деления тактовой частоты HCLK для каждого таймера.

После подачи тактового сигнала на блок таймера можно приступить к работе с ним.

### Режим таймера

Таймеры построены на базе 16-битный счетчика, объединенного с 16-битным предварительным делителем. Скорость счета таймера зависит от значения, находящегося в регистре делителя.

Счетчик может считать вверх, вниз или вверх и вниз.

Базовый блок таймера включает в себя:

- Основной счетчик таймера (TIMx\_CNT)
- Делитель частоты при счете основного счетчика (TIMx\_PSC):
- Основание счета основного счетчика (TIMx\_ARR)

Сигналом для изменения CNT может служить как внутренняя частота TIM\_CLK, так и события в других счетчиках, либо события на линиях TxCH0 данного счетчика.

Чтобы запустить работу основного счетчика необходимо задать:

- Начальное значение основного счетчика таймера – TIMx\_CNT;
- Значение предварительного делителя счетчика – TIMx\_PSG, при этом основной счетчик будет считать на частоте  $CLK = TIMx\_CLK / (PSG + 1)$ ;
- Значение основания счета для основного счетчика TIMx\_ARR.

Режим работы счетчика TIMx\_CNTRL:

- выбрать источник события переключения счетчика EVENT\_SEL;
- режим счета основного счетчика CNT\_MODE (значения 00 и 01 при тактировании внутренней частотой, значения 10 и 11 при тактировании внешними сигналами);
- направление счета основного счетчика DIR;
- разрешить работу счетчика CNT\_EN.

По событиям совпадения значения основного счетчика с значением нуля или значением основания счета генерируется прерывание, которое может быть замаскировано.

### **Режимы счета**

Счет вверх: CNT\_MODE = 00, DIR = 0

```
TIMx->TIMx_CNTRL = 0x00000000; //Режим инициализации таймера
```

```
//Настраиваем работу основного счетчика
```

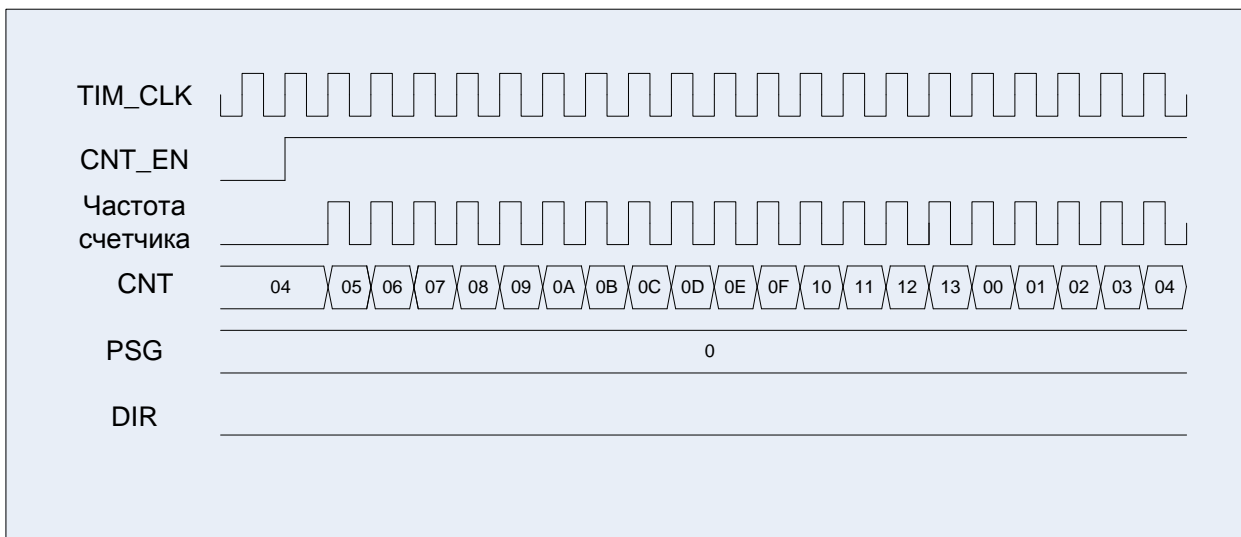
```
TIMx->TIMx_CNT = 0x00000004; //Начальное значение счетчика
```

```
TIMx->TIMx_PSG = 0x00000000; //Предделитель частоты
```

```
TIMx->TIMx_ARR = 0x00000013; //Основание счета
```

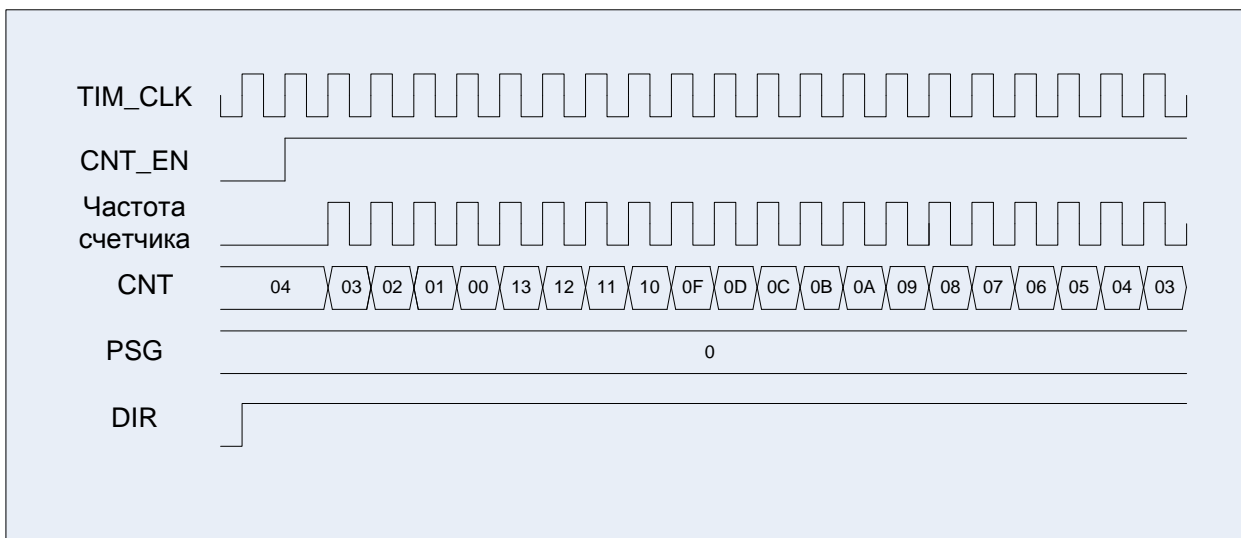
```
//Разрешение работы таймера.
```

```
TIMx->TIMx_CNTRL = 0x00000001; //Счет вверх по TIM_CLK.
```



**Рис. 34** Диаграммы работы таймера, счет вверх от 0 до 0x0013, стартовое значение 0x0004

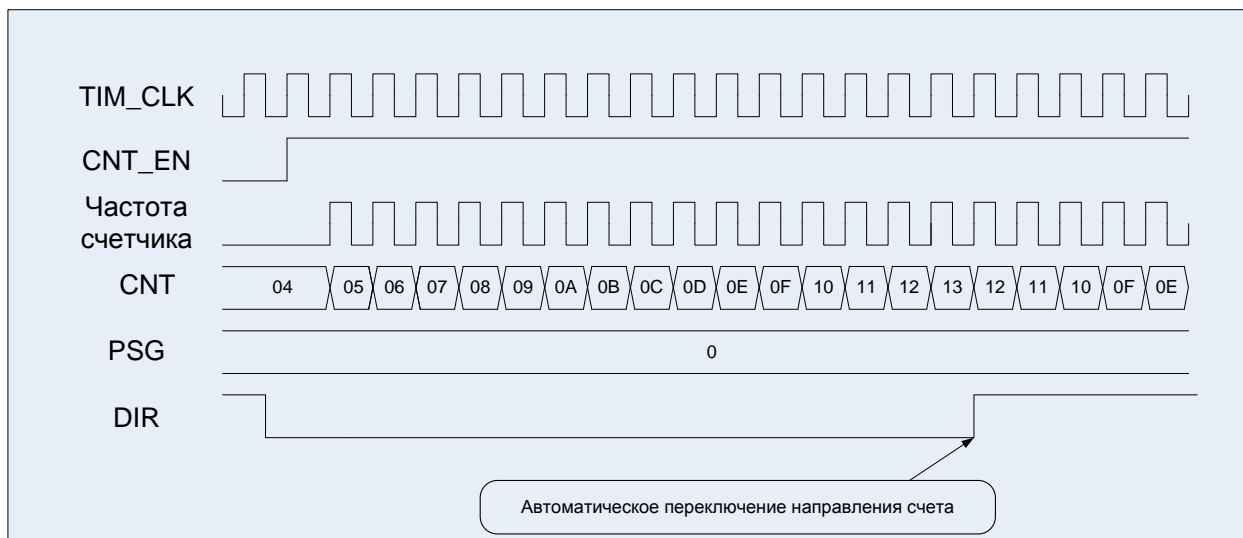
Счет вниз: CNT\_MODE = 00, DIR = 1  
 TIMx->TIMx\_CNTRL = 0x00000000; //Режим инициализации таймера  
 //Настраиваем работу основного счетчика  
 TIMx->TIMx\_CNT = 0x00000004; //Начальное значение счетчика  
 TIMx->TIMx\_PSG = 0x00000000; //Предделитель частоты  
 TIMx->TIMx\_ARR = 0x00000013; //Основание счета  
  
 //Разрешение работы таймера.  
 TIMx->TIMx\_CNTRL = 0x00000009; //Счет вниз по TIM\_CLK.



**Рис. 35** Диаграммы работы таймера, счет вниз от 0x0013 до 0, стартовое значение 0x0004

Счет вверх/вниз: CNT\_MODE = 01, DIR = 0  
 TIMx->TIMx\_CNTRL = 0x00000000; //Режим инициализации таймера  
 //Настраиваем работу основного счетчика  
 TIMx->TIMx\_CNT = 0x00000004; //Начальное значение счетчика  
 TIMx->TIMx\_PSG = 0x00000000; //Предделитель частоты  
 TIMx->TIMx\_ARR = 0x00000013; //Основание счета  
  
 //Разрешение работы таймера.

*TIMx->TIMx\_CNTRL = 0x00000041; //Счет вверх/вниз по TIM\_CLK.*



**Рис. 36** Диаграммы работы таймера, счет вверх/вниз, сначала вверх

Счет вверх/вниз: CNT\_MODE = 01, DIR = 1

*TIMx->TIMx\_CNTRL = 0x00000000; //Режим инициализации таймера*

*//Настраиваем работу основного счетчика*

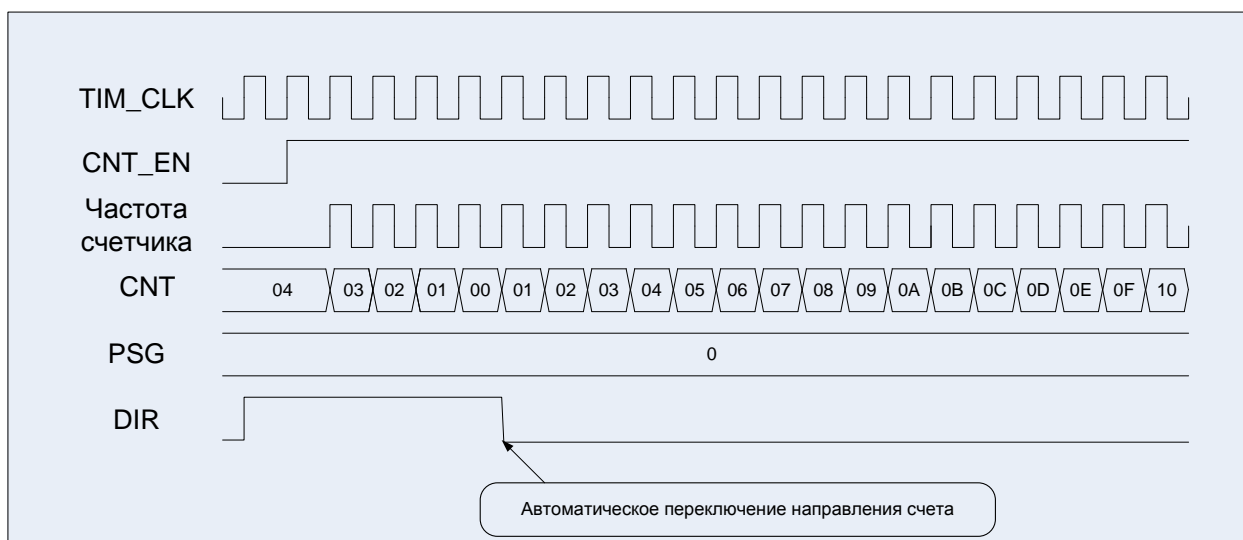
*TIMx->TIMx\_CNT = 0x00000004; //Начальное значение счетчика*

*TIMx->TIMx\_PSG = 0x00000000; //Предделитель частоты*

*TIMx->TIMx\_ARR = 0x00000013; //Основание счета*

*//Разрешение работы таймера.*

*TIMx->TIMx\_CNTRL = 0x00000049; //Счет вверх/вниз по TIM\_CLK.*



**Рис. 37** Диаграммы работы таймера, счет вверх/вниз, сначала вниз

### Источник событий для счета

Внутренний тактовый сигнал (TIM\_CLK)

События в других счетчиках (CNT==ARR в таймере X)

Внешний тактовый сигнал режим 1: События на линиях TxСНО данного счетчика.

Внешний тактовый сигнал режим 2: События на линиях TxСНО данного счетчика.

Внешний тактовый сигнал режим 3: События на входе ETR данного счетчика.

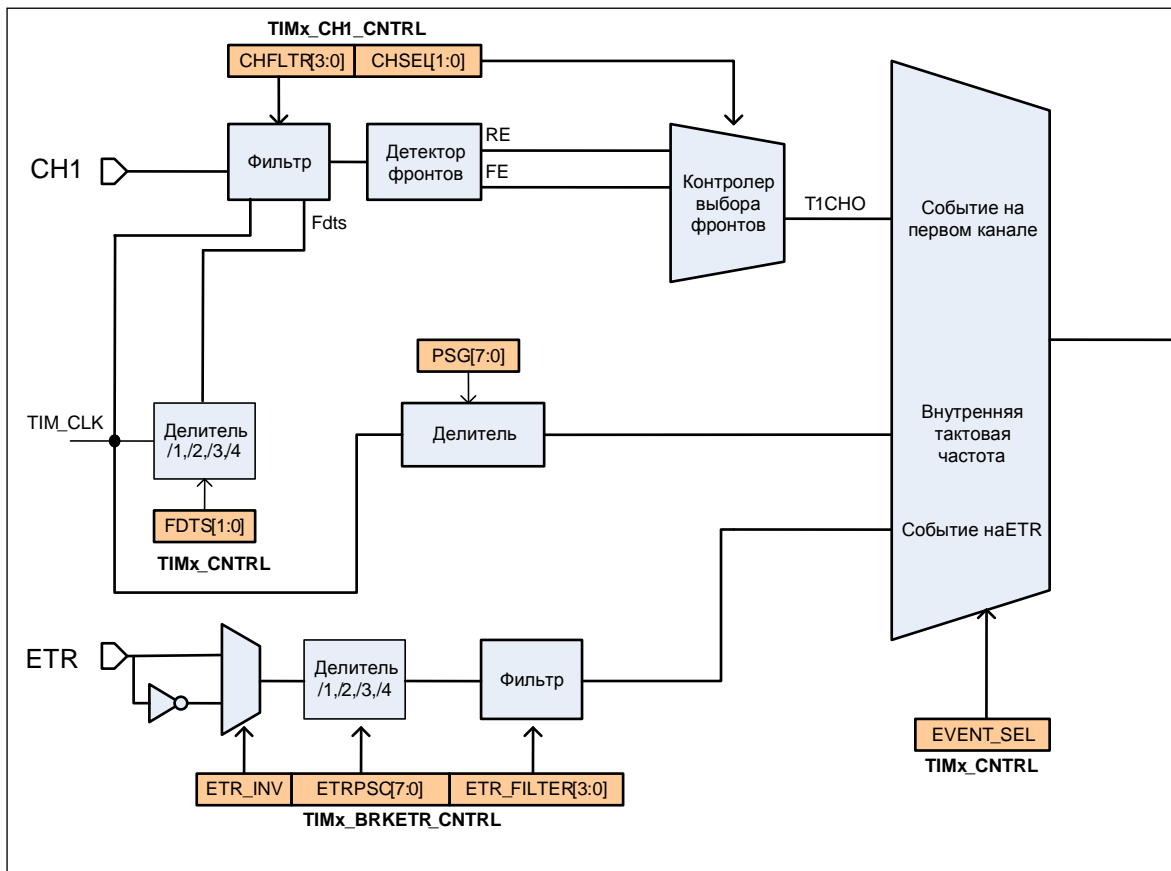


Рис. 38 Структурная схема формирования события для счета

### Внутренний тактовый сигнал (TIM\_CLK)

Этот режим выбирается, когда  $CNT\_MODE = 0x$ ,  $EVENT\_SEL = 0000$ . Для запуска этого режима необходимо задать начальное значение основного счетчика, значение предварительного делителя основного счетчика, основание счета для основного счетчика и задать режим работы в регистре **TIMx\_CNTRL**. Значения регистров **TIMx\_CNT**, **TIMx\_PSG** и **TIMx\_ARR** можно изменять даже во время работы счетчика, при этом их значения вступают в силу по  $CNT = ARR$  или  $CNT = 0$ , в зависимости от направления счета. Значение регистра основание счета (**TIMx\_ARR**) может вступить в силу мгновенно после записи его в регистр при условии установленного поля  $ARRB\_EN = 1$  (регистр **TIMx\_CNTRL**). Если значение предварительного делителя основного счетчика не равно нулю, то счетный регистр делителя будет инкрементироваться по каждому импульсу сигнала **TIM\_CLK** до тех пор, пока не достигнет значения, находящегося в регистре делителя. Далее счетный регистр делителя сбрасывается в ноль, содержимое основного счетчика таймера изменится на 1 и снова начинается счет. Поле **DIR** (регистра **TIMx\_CNTRL**) определяет, в какую сторону будет меняться значение счетчика:  $DIR = 0$  - счетчик считает вверх (см. Рис. 39),  $DIR = 1$  - счетчик считает вниз (см. Рис. 40). Если  $CNT\_MODE = 00$ , то направление счета определяется полем **DIR**, если  $CNT\_MODE = 01$ , счетчик считает вверх/вниз с автоматическим изменением **DIR** (см. Рис. 41).

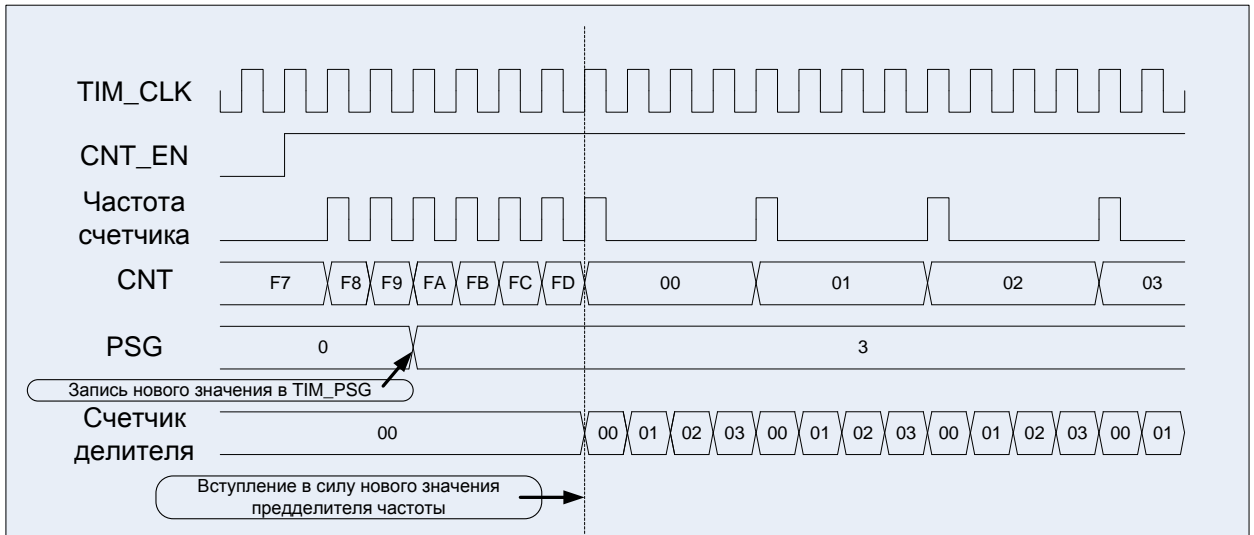


Рис. 39 Диаграммы работы счетчика: счет вверх

(CNT\_MODE = 00, EVENT\_SEL = 0000, DIR = 0)

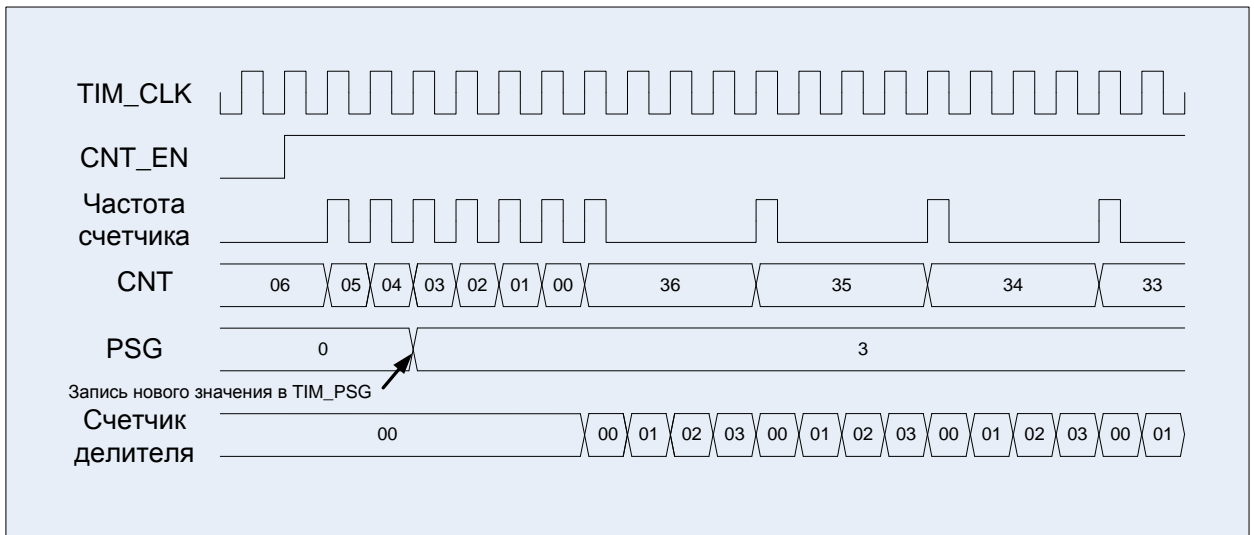


Рис. 40 Диаграммы работы счетчика: счет вниз

(CNT\_MODE = 00, EVENT\_SEL = 0000, DIR = 1).

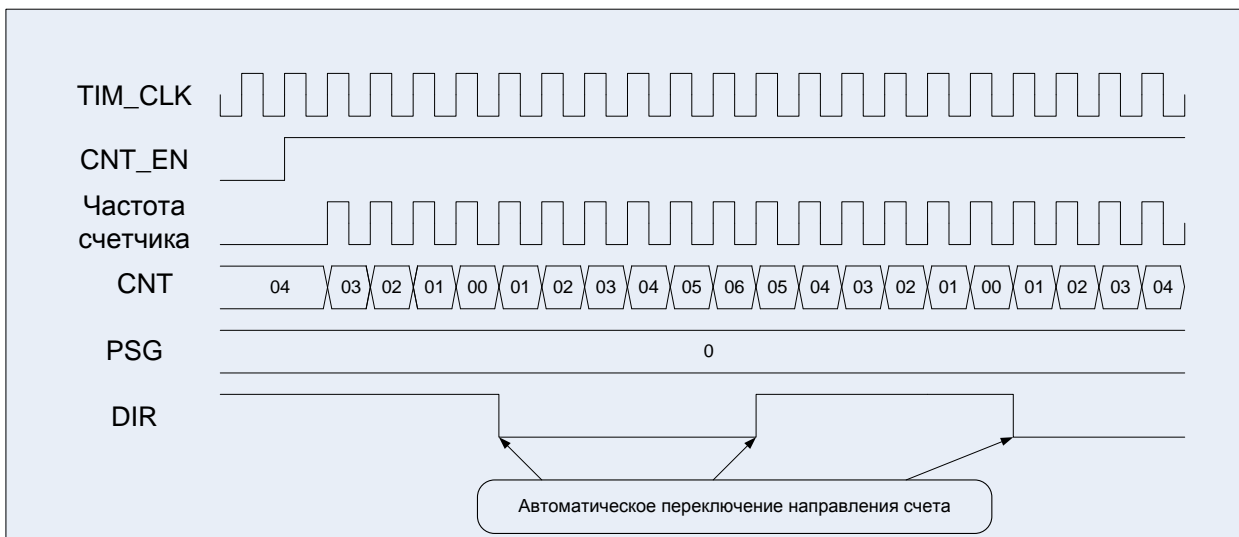


Рис. 41 Диаграммы работы счетчика: счет вниз/вверх

(CNT\_MODE = 01, EVENT\_SEL = 0000, DIR = 1).

### События в других счетчиках (CNT==ARR в таймере X )

Каждый из блоков таймеров полностью независим друг от друга, но у них предусмотрена возможность синхронизированной друг с другом работы. Это позволяет создавать более сложные массивы таймеров, которые работают полностью автономно и не требуют написания какого-либо кода программы для выполнения сложных временных функций.

У каждого таймера имеются входы запуска от других таймеров, а также внешние входы, связанные с выводами блоков захвата/сравнения.

У каждого из блоков таймеров имеется выход запуска, который соединен с входами других таймеров. Синхронизация таймеров возможна в нескольких различных режимах. Ниже показан пример каскадного соединения счетчиков.

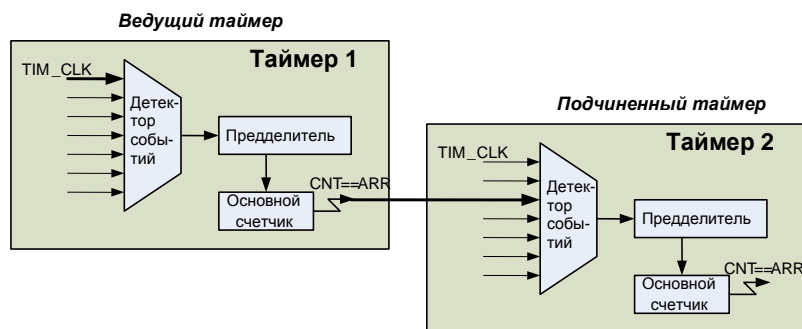


Рис. 42 Пример каскадного соединения таймеров



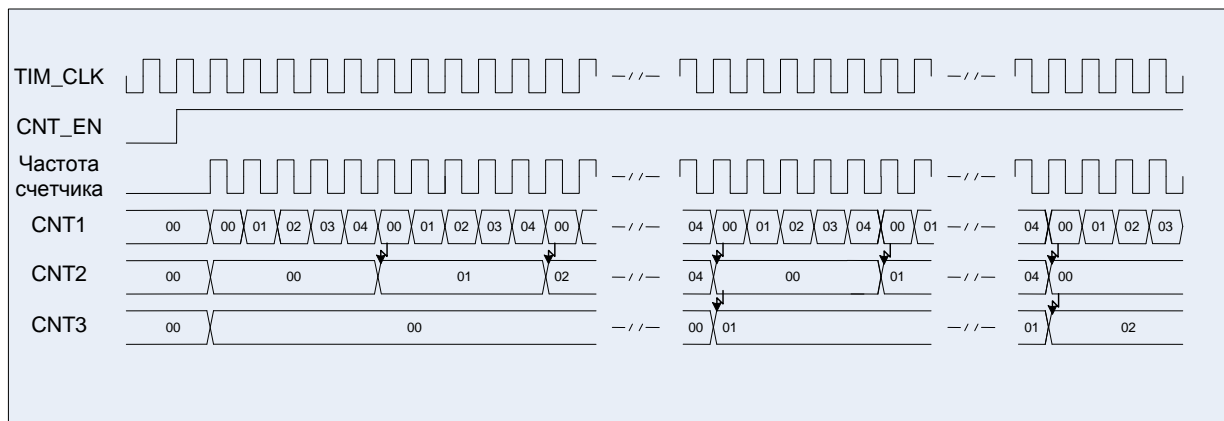


Рис. 43 Диаграммы работы двух таймеров в каскаде

DIR\_1, DIR\_2;  
 EVENT\_SEL\_1 = 0000, EVENT\_SEL\_2 = 0001;  
 CNT\_MODE\_1, CNT\_MODE\_2= 00;

### Внешний тактовый сигнал режим 1

События на линиях TxCHO данного счетчика

Этот режим выбирается, когда EVENT\_SEL = 01xx в регистре TIMx\_CNTRL (см. раздел «TIMx\_CNTRL»). Счетчик может считать по положительному фронту или по отрицательному фронту на выбранном входе. На входе сигнала стоит фильтр, с помощью которого можно контролировать длительность сигнала, для фильтрации можно использовать как сигнал TIM\_CLK, при этом может быть идентифицированная длительность 1, 2, 4, 8 TIM\_CLK, так же можно при фильтровании использовать производную от TIM\_CLK частоту семплирования данных (FDTS). Частота семплирования данных задается в регистре TIMx\_CNTRL в поле FDTS.

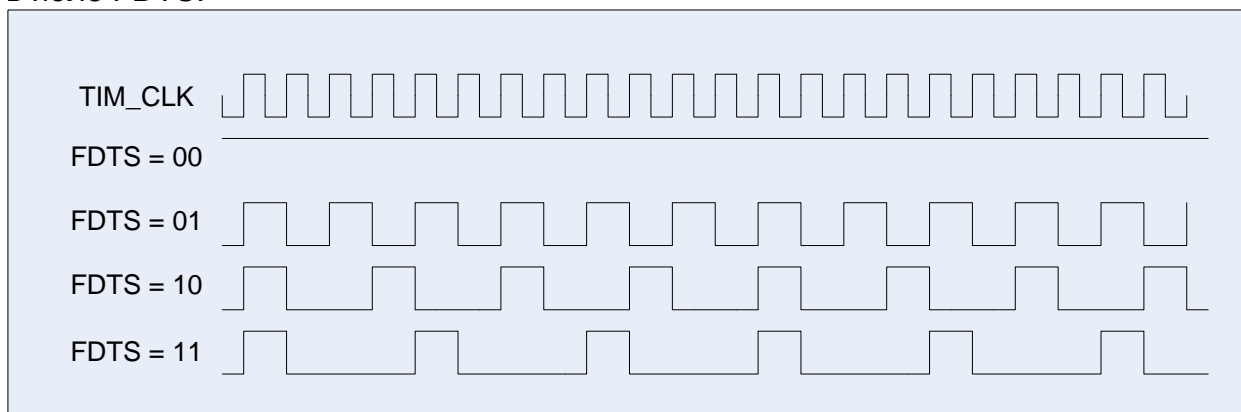


Рис. 44 Диаграммы возможных частот семплирования данных (FDTS)

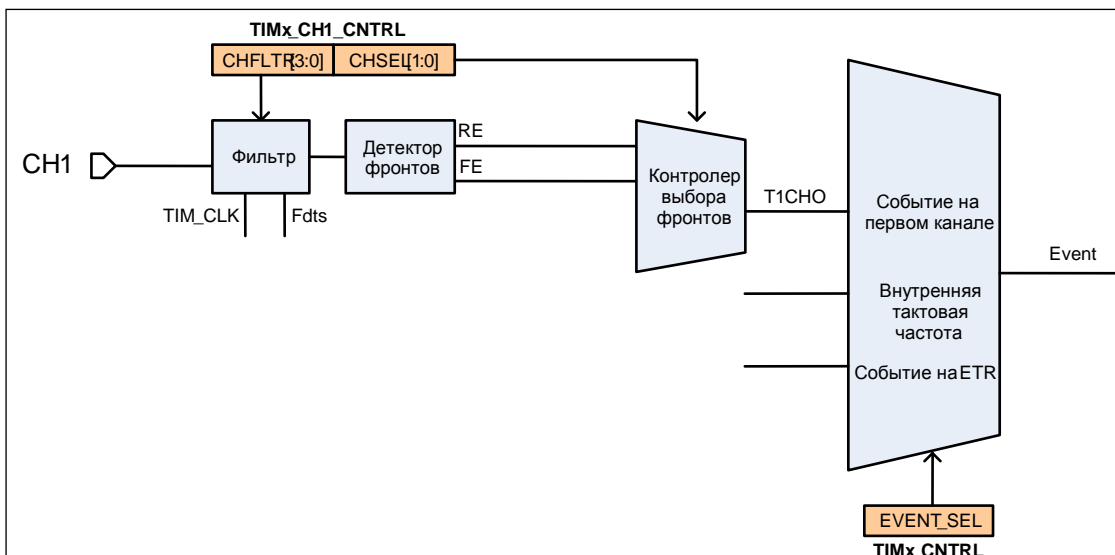


Рис. 45 Тактирование с входа первого канала

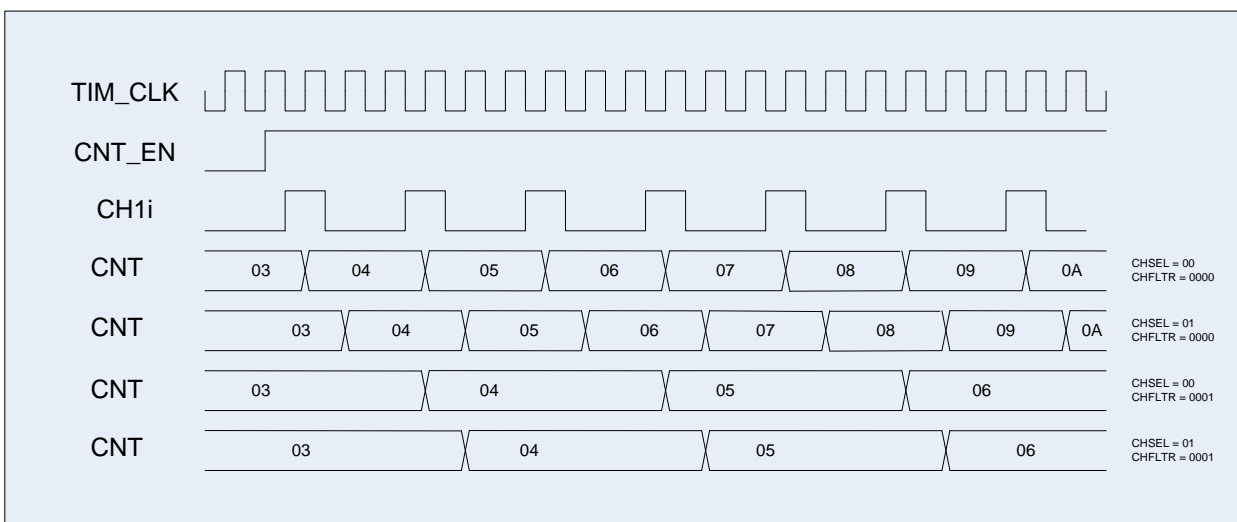


Рис. 46 Диаграмма внешнего тактирования с разными вариантами фильтра

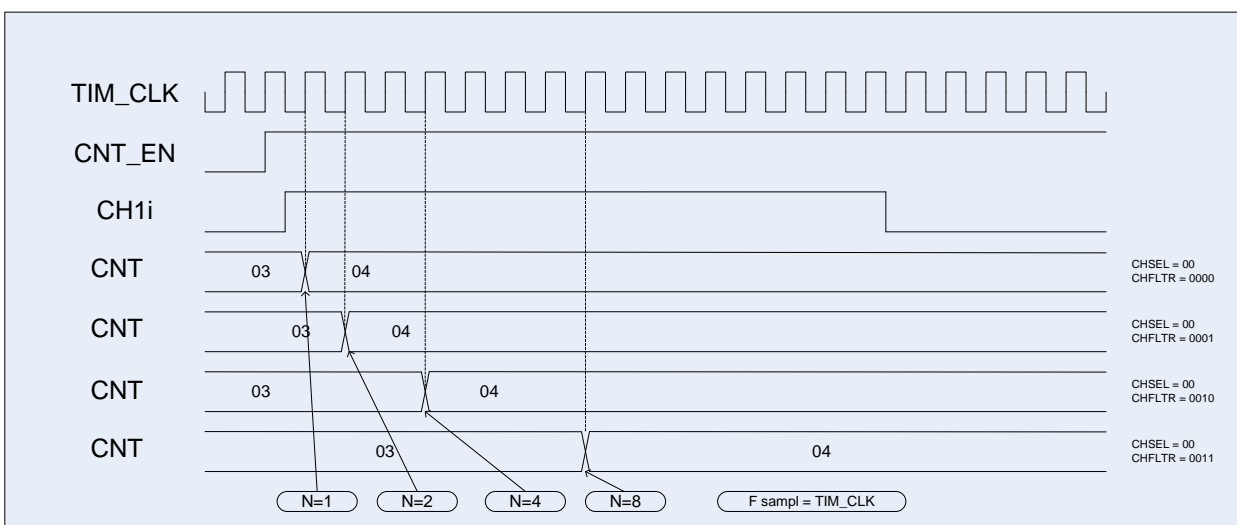


Рис. 47 Диаграмма внешнего тактирования с разными вариантами фильтра

## Внешний тактовый сигнал режим 2

События на линии ETR данного счетчика

Этот режим выбирается, когда EVENT\_SEL = 1000 в регистре TIMx\_CNTRL. В регистре TIMx\_BRKETR\_CNTRL можно настроить коэффициент деления 2, 4 или 8 (ETRPSC) данного входа тактовой частоты, а также использовать инверсию входа.

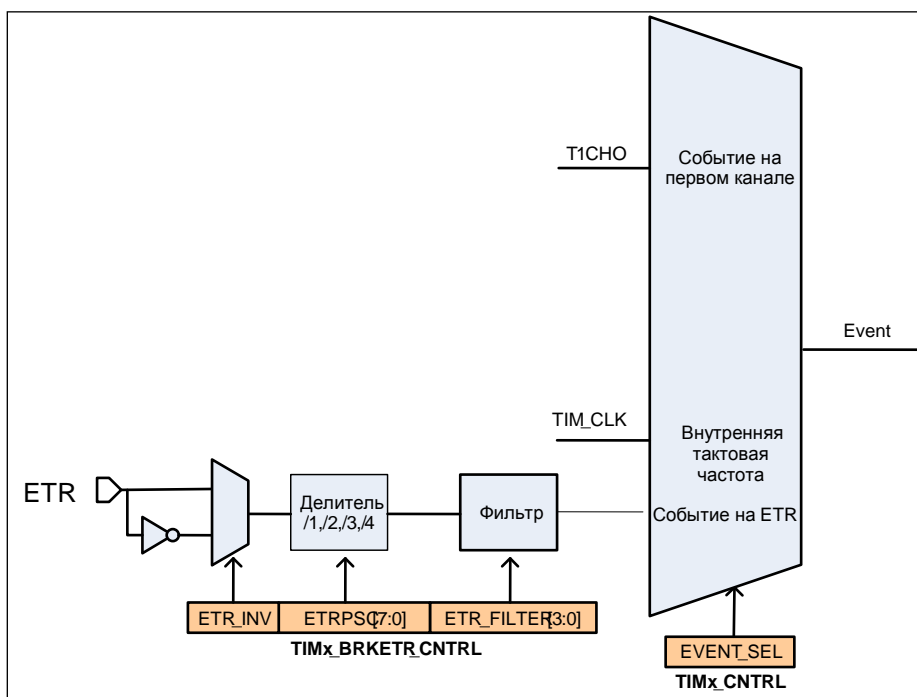


Рис. 48 Схема тактирования сигналом со входа ETR

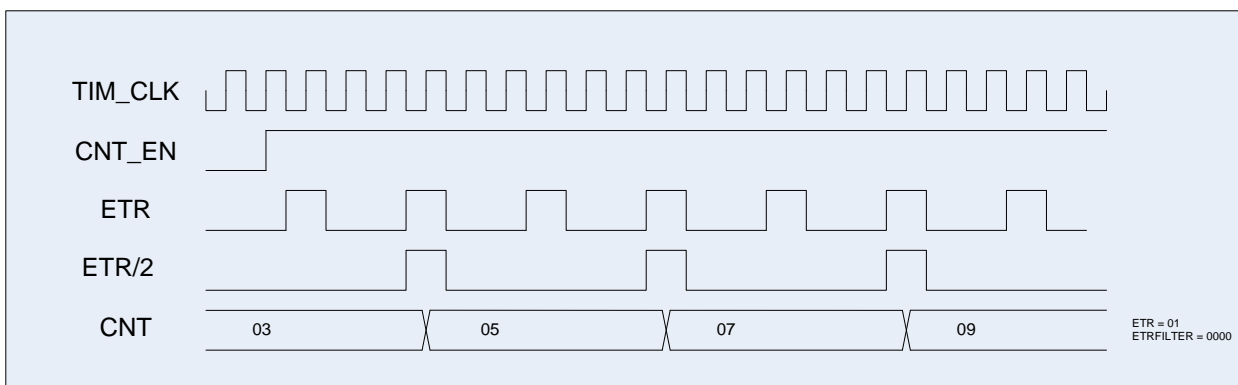


Рис. 49 Диаграмма тактирования сигналом со входа ETR

### Режим захвата

Структурная схема блока захвата представлена на Рис. 50.

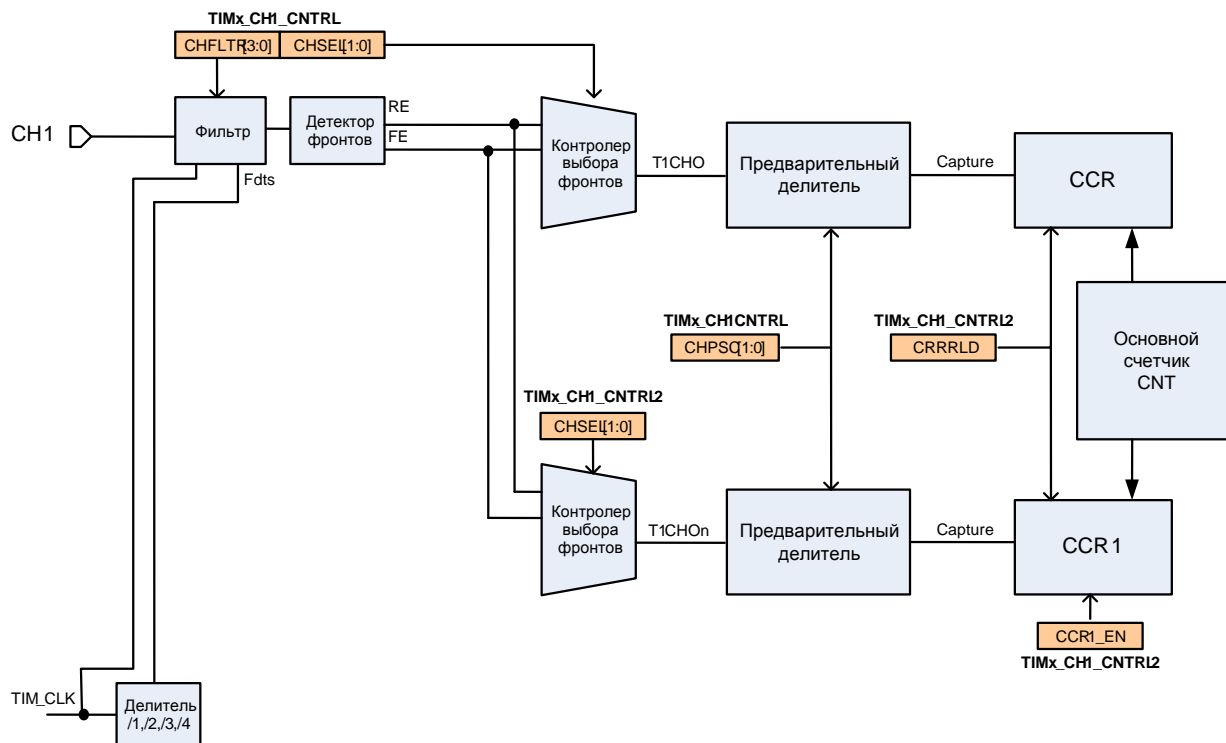


Рис. 50 Структурная схема блока захвата на примере канала 1

Для включения режима захвата для определенного канала необходимо в регистре управления каналом TIMx\_CHy\_CNTRL (см. раздел «TIMx\_CHy\_CNTRL») записать 1 в поле CAPnPWM. Для регистрации событий по линии CHx используется схема регистрации событий. Входной сигнал фиксируется в Таймере с частотой Fdts, или TIM\_CLK. Так же вход может быть настроен на прием импульсов заданной длины за счет конфигурирования блока FILTER. На выходе блока фильтр вырабатывается сигнал положительного перепада и отрицательного перепада. В блоке контроллера выбора фронтов производится выбор используемого для захвата сигнала, между положительным фронтом канала и отрицательным фронтом канала. После блока MUX предварительный делитель может быть использован для фиксации каждого события, каждого второго, каждого четвертого и каждого восьмого события. Выход предварительного делителя является сигналом Capture для регистра CCR, и Capture1 для регистра CCR1 при этом в регистры CCR и CCR1 записывается текущее значение основного счетчика CNT.

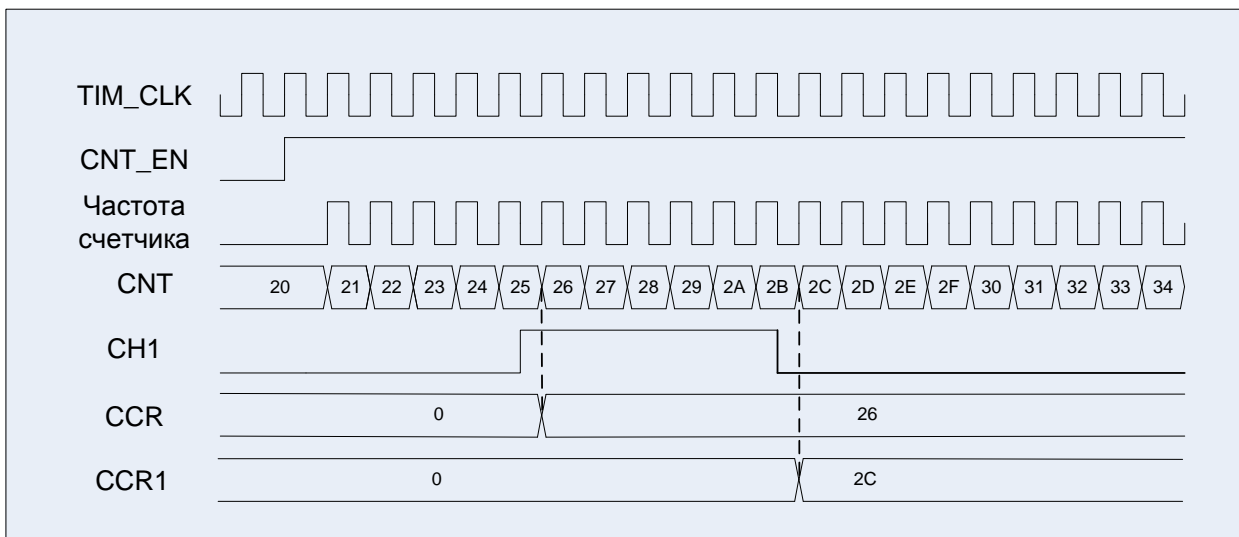


Рис. 51 Диаграмма захвата события со входа первого канала

На Рис. 51 показан пример захвата значения основного счетчика в регистр CCR по положительному фронту на входе канала, а в регистр CCR1 по отрицательному фронту на входе канала. В регистре TIMx\_IE можно разрешить выработку прерываний по событию захвата на определенном канале.

### Режим ШИМ

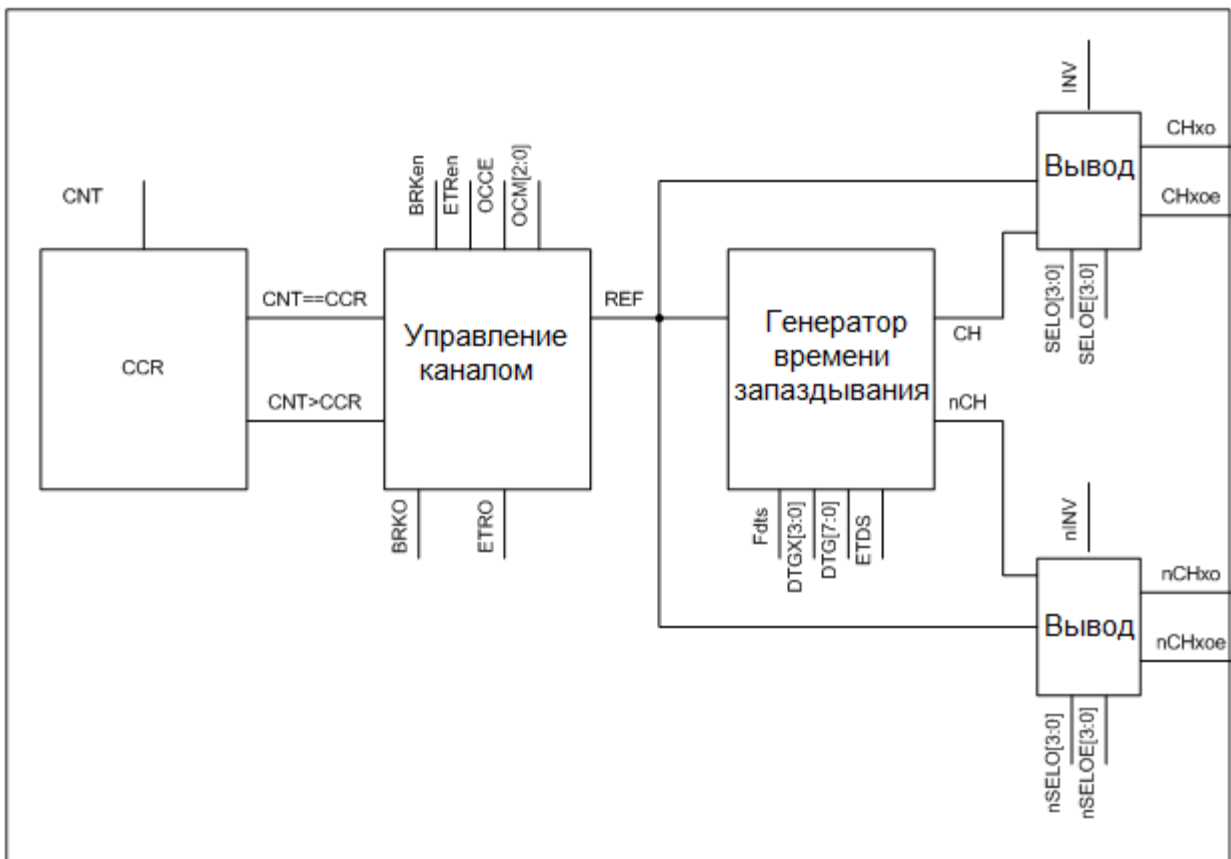
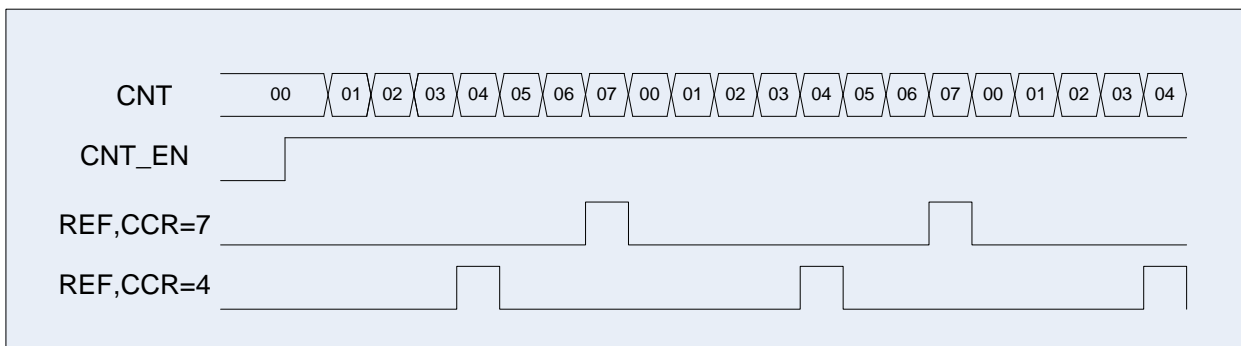


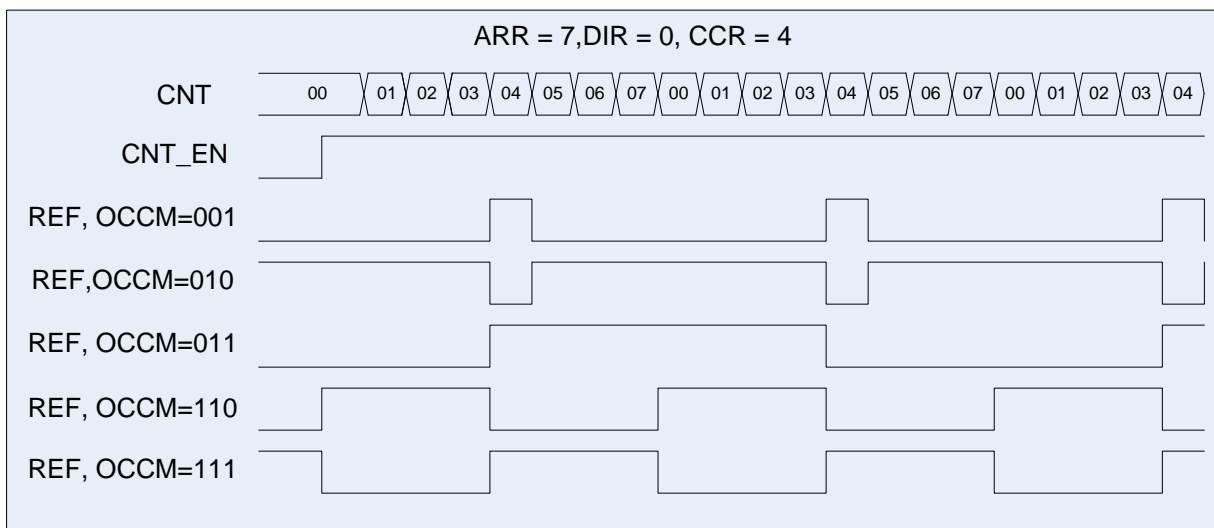
Рис. 52 Структурная схема блока сравнения

Для включения режима сравнения необходимо в регистре управления каналом TIMx\_CHy\_CNTRL (см. раздел «TIMx\_CHy\_CNTRL») записать 0 в поле

САРnPWM. При работе в режиме ШИМ выходной сигнал может формироваться на основании сравнения значения в регистре CCR (см. раздел «Регистр конфигурации и управления») и основного счетчика CNT или регистров CCR, CCR1 и значения основного счетчика CNT. Полученный сигнал может без изменения выдаваться на выходы CHxO и nCHxO. Либо с применением схемы запаздывания (Генератор времени запаздывания) формируются управляющие сигналы с мертвой зоной. У канала есть два выхода - прямой и инверсный. Для каждого выхода формируется как сигнал для выдачи, так и сигнал разрешения выдачи, т.е. если выход канала должен всегда выдавать тот или иной уровень, то на выводе разрешения выдачи CHxOE (для прямого) и на CHxNOE (для инверсного) должны формироваться "1". Если канал работает на вход (например, режим захвата), то там всегда должен быть "0" для прямого канала. Сигналы OE по тем же принципам, что и просто выходные уровни, но у них есть собственные сигналы разрешения вывода SELOE и nSELOE, в которых можно выбрать постоянный уровень, либо формируемый на основании REF.



**Рис. 53** Диаграмма работы схемы в режиме ШИМ, CCR1\_EN=0



**Рис. 54** Диаграмма работы схемы в режиме ШИМ, CCR1\_EN=0

Сигнал REF может быть очищен с использованием внешнего сигнала со входа ETR или внешнего триггерированного по PCLK сигнала со входа BRK.

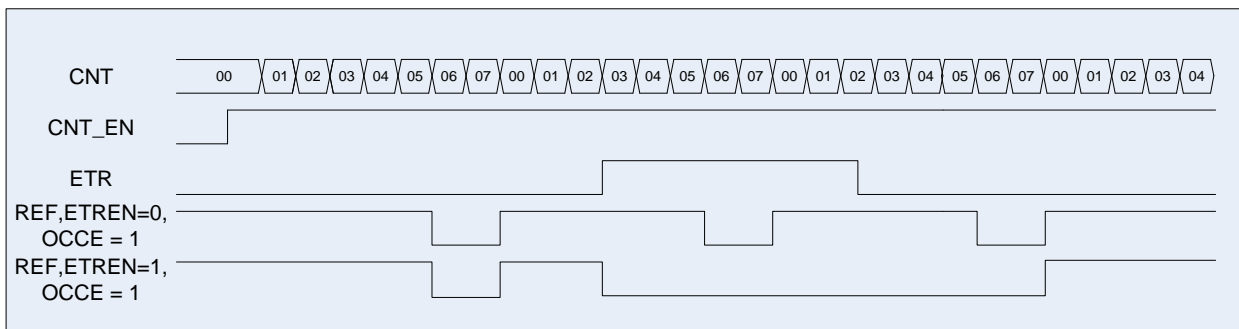


Рис. 55 Диаграмма работы схемы в режиме ШИМ, CCR1\_EN = 0

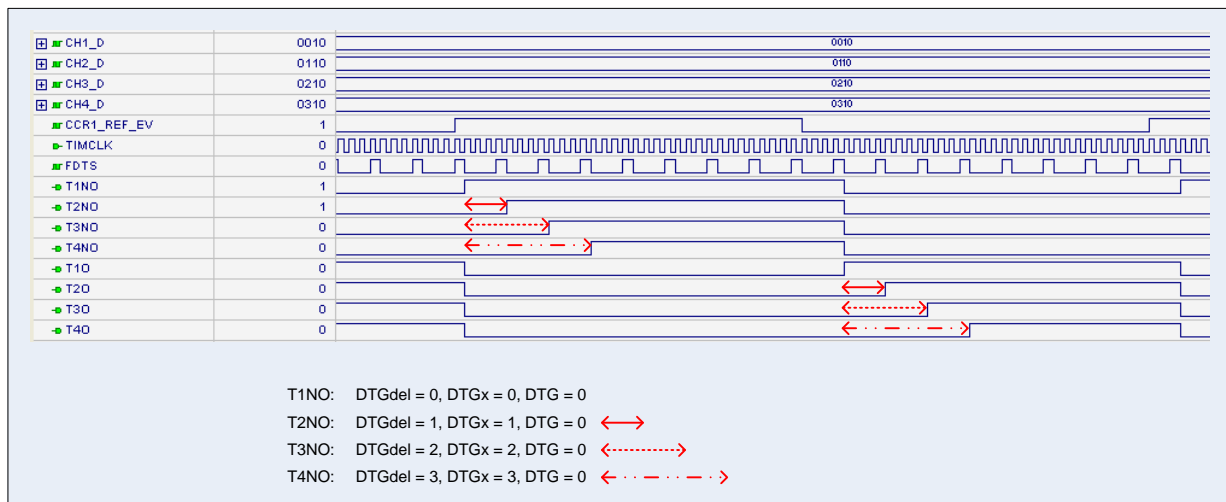


Рис. 56 Диаграмма работы схемы DTG

Если CCR1\_EN = 1, тогда значение основного счетчика CNT сравнивается со значениями регистров CCR и CCR1, и в зависимости от запрограммированного формата выработки сигнала REF (регистры управления каналами таймера TIMx\_CHy\_CNTRL поле OCCM) будет формироваться сигнал соответствующей формы.

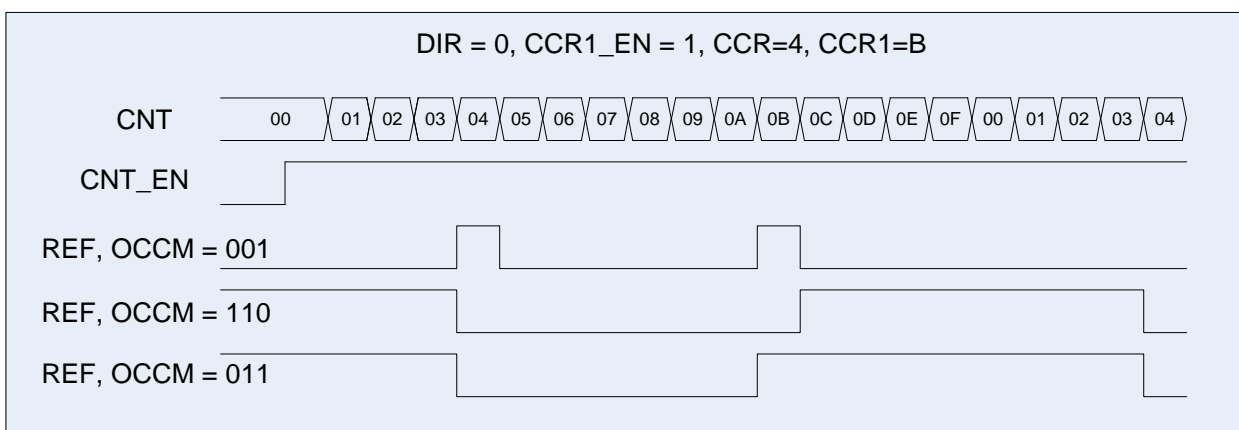


Рис. 57 Диаграмма работы схемы в режиме ШИМ, CCR1\_EN = 1

При записи новых значений CCR и CCR1, если установлен бит CRRRLD, то регистры CCR1 и CCR получают новые значения только при CNT = 0, иначе запись осуществляется немедленно. Факт окончания записи обозначается взведением флага WR\_CMPL.

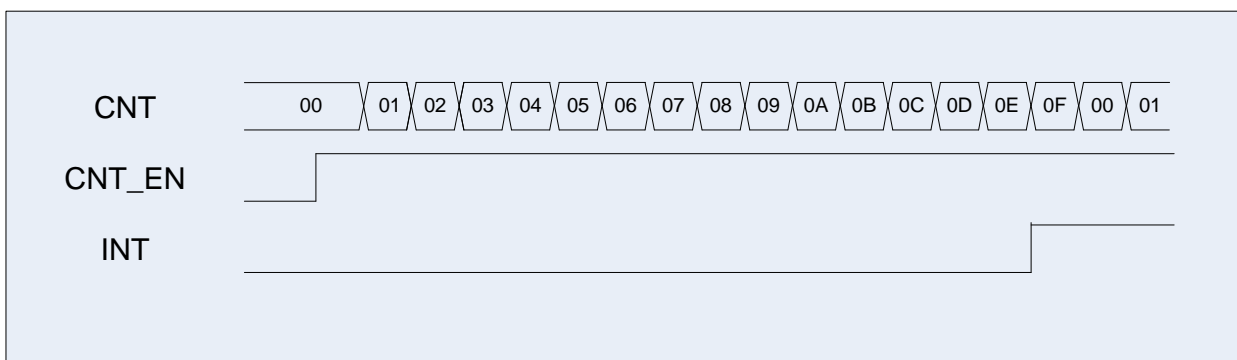
Примеры

**Обычный счетчик.**

```
RST_CLK->PER_CLOCK = 0xFFFFFFFF;
RST_CLK->TIM_CLOCK = 0x07000000;
TIMx->TIMx_CNTRL = 0x00000000;
//Настраиваем работу основного счетчика
TIMx->TIMx_CNT = 0x00000000; //Начальное значение счетчика
TIMx->TIMx_PSG = 0x00000000; //Предделитель частоты
TIMx->TIMx_ARR = 0x0000000F; //Основание счета
```

```
TIMx->TIMx_IE = 0x00000002; //Разрешение генерировать прерывание при CNT = ARR
```

```
TIMx->TIMx_CNTRL = 0x00000001; //Счет вверх по TIM_CLK. Разрешение работы таймера.
```



**Рис. 58**

**Режим захвата.**

```
RST_CLK->PER_CLOCK = 0xFFFFFFFF; //Разрешение тактовой частоты таймеров
RST_CLK->TIM_CLOCK = 0x07000000; //Включение тактовой частоты таймеров
TIMx->TIMx_CNTRL = 0x00000000; //Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMx->TIMx_CNT = 0x00000000; //Начальное значение счетчика
TIMx->TIMx_PSG = 0x00000000; //Предделитель частоты
TIMx->TIMx_ARR = 0x000000FF; //Основание счета
```

```
TIMx->TIMx_IE = 0x00001E00; //Разрешение генерировать прерывание
//по переднему фронту на выходе CAP
```

```
//Режим работы канала - захват
TIMx->TIMx_CHy_CNTRL[0] = 0x00008000;
```

```
//Режим работы выхода канала – канал на выход не работает
TIMx->TIMx_CHy_CNTRL[1] = 0x00000000;
```

```
TIMx->TIMx_CNTRL = 0x00000001; //Счет вверх по TIM_CLK. Разрешение работы таймера.
```



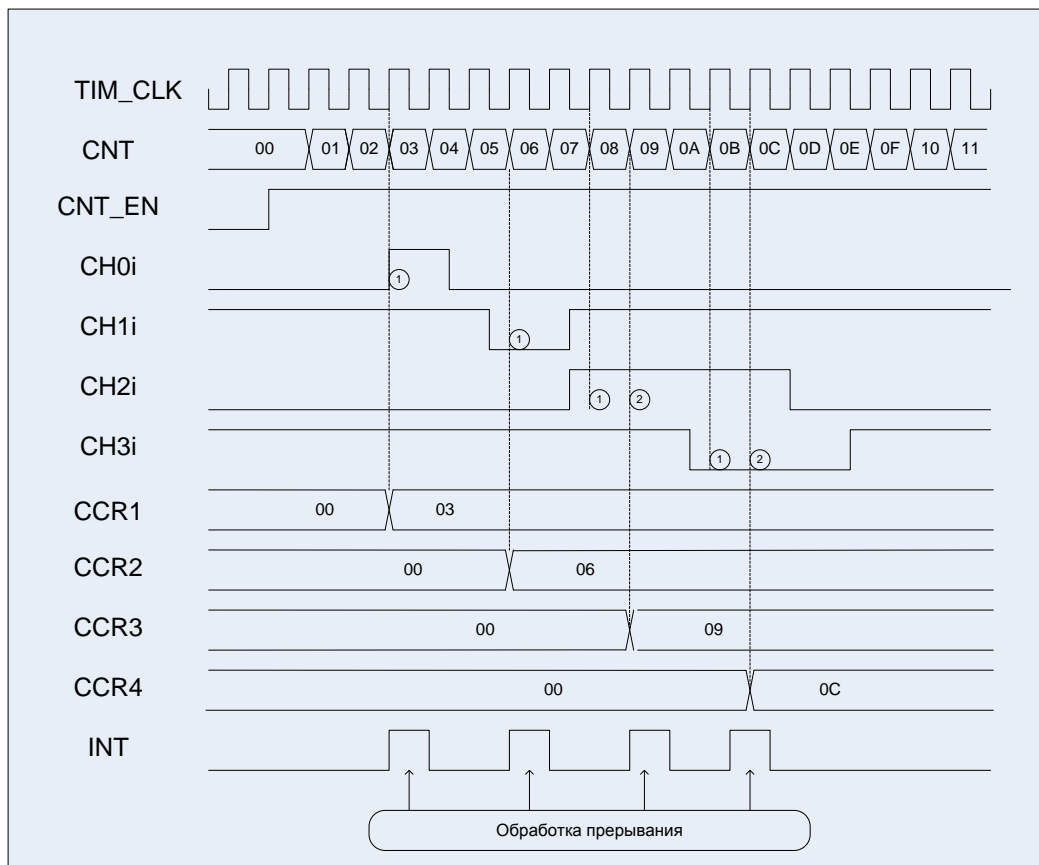


Рис. 59 Диаграммы примера работы в режиме захвата

### Режим ШИМ.

RST\_CLK->PER\_CLOCK = 0xFFFFFFFF; //Разрешение тактовой частоты таймеров

RST\_CLK->TIM\_CLOCK = 0x07000000; //Включение тактовой частоты таймеров

TIMx->TIMx\_CNTRL = 0x00000000; //Режим инициализации таймера

//Настраиваем работу основного счетчика

TIMx->TIMx\_CNT = 0x00000000; //Начальное значение счетчика

TIMx->TIMx\_PSG = 0x00000000; //Предделитель частоты

TIMx->TIMx\_ARR = 0x00000010; //Основание счета

TIMx->TIMx\_IE = 0x000001E0; //Разрешение генерировать прерывание

//по переднему фронту на выходе REF

//Режим работы канала - ШИМ

TIMx->TIMx\_CHy\_CNTRL[0] = 0x00000200;

//Режим работы выхода канала – канал на выход не работает

TIMx->TIMx\_CHy\_CNTRL 1[0] = 0x00000099;

//Разрешение работы таймера.

TIMx->TIMx\_CNTRL = 0x00000001; //Счет вверх по TIM\_CLK.

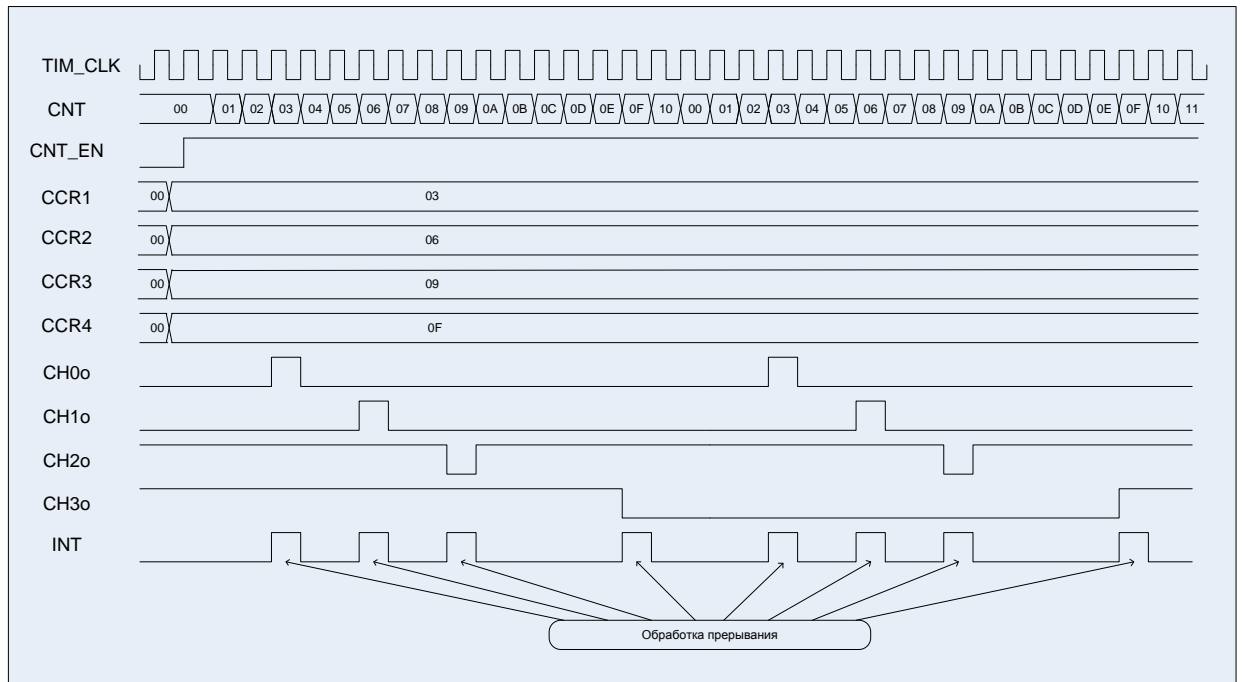


Рис. 60

**Описание регистров блока таймера**

Базовые адреса и смещения регистров управления.

Таблица 99

Адрес	Название	Описание
0x4007_0000	Timer1	Контроллер Timer1
0x4007_8000	Timer2	Контроллер Timer2
Смещение		
0x00	TIMx_CNT[15:0]	Основной счетчик таймера
0x04	TIMx_PSG[15:0]	Делитель частоты при счете основного счетчика
0x08	TIMx_ARR[15:0]	Основание счета основного счетчика
0x0C	TIMx_CNTRL[7:0]	Регистр управления основного счетчика
0x10	TIMx_CCR1[15:0]	Регистр сравнения, захвата для 1 канала таймера
0x14	-	-
0x18	-	-
0x1C	-	-
0x20	TIMx_CH1_CNTRL[15:0]	Регистр управления для 1 канала таймера
0x24	-	-
0x28	-	-
0x2C	-	-
0x30	TIMx_CH1_CNTRL1[15:0]	Регистр управления 1 для 1 канала таймера
0x34	-	-
0x38	-	-
0x3C	-	-
0x40	TIMx_CH1_DTG[15:0]	Регистр управления DTG для 1

		канала таймера
0x44	-	-
0x48	-	-
0x4C	-	-
0x50	TIMx_BRKETR_CNTRL[15:0]	Регистр управления входом BRK и ETR
0x54	TIMx_STATUS[15:0]	Регистр статуса таймера
0x58	TIMx_IE[15:0]	Регистр разрешения прерывания таймера
0x5C	-	-
0x60	TIMx_CH1_CNTRL2[15:0]	Регистр управления 2 для 1 канала таймера
0x64	-	-
0x68	-	-
0x6C	-	-
0x70	TIMx_CCR11[15:0]	Регистр сравнения1, захвата для 1 канала таймера
0x74	-	-
0x78	-	-
0x7C	-	-

### TIMx\_CNT

Основной счетчик таймера

Номер	31	15	0
Доступ	R/W	R/W	R/W
Сброс	0	0	0
	CNT[15:0]		

**Таблица 100**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16	-	Зарезервировано
15...0	CNT[15:0]	Значение основного счетчика таймера

### TIMx\_PSG

Делитель частоты при счете основного счетчика

Номер	31	15	0
Доступ	R/W	R/W	R/W
Сброс	0	0	0
	PSG[15:0]		

**Таблица 101**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16	-	Зарезервировано
15...0	PSG[15:0]	Значение предварительного делителя счетчика Основной счетчик считает на частоте $CLK = TIM\_CLK/(PSG+1)$

### TIMx\_ARR

Основание счета основного счетчика

Номер	31	15	0
Доступ	R/W	R/W	R/W
Сброс	0	0	0
			ARR[15:0]

**Таблица 102**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16	-	Зарезервировано
15...0	ARR[15:0]	Основание счета для основного счетчика $CNT = [0...ARR]$

### TIMx\_CNTRL

Регистр управления основного счетчика

Номер	31..10	11..8	7..6	5..4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0000	00	00	0	0	0	0
	-	EVENT SEL [3:0]	CNT MODE [1:0]	FDTS [1:0]	DIR	WR CMPL	ARRB EN	CNT EN

**Таблица 103**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..11	-	Зарезервировано
11..8	EVENT_SEL [3:0]	Биты выбора источника событий 0000 – всегда “0” 0001 – $CNT == ARR$ в таймере 1 0010 – $CNT == ARR$ в таймере 2 0100 – событие на первом канале 1000 – событие на ETR
7..6	CNT_MODE [1:0]	Режим счета основного счетчика 00 – счетчик вверх при $DIR=0$ счетчик вниз при $DIR=1$

		при PSG = 0 01 – счетчик вверх/вниз с автоматическим изменением DIR при PSG = 0 10 – счетчик вверх при DIR=0 счетчик вниз при DIR=1 при EVENT = 1 11 – счетчик вверх/вниз с автоматическим изменением DIR при EVENT = 1
5..4	FDTS[1:0]	Частота семплирования данных FDTS 00 – каждый TIM_CLK 01 – каждый второй TIM_CLK 10 – каждый третий TIM_CLK 11 – каждый четвертый TIM_CLK
3	DIR	Направление счета основного счетчика 0 – вверх, от 0 до ARR 1 – вниз, от ARR до 0
2	WR_CMPL	Окончание записи, при задании нового значения регистров CNT, PSG и ARR 1 – данные не записаны и идет запись 0 – новые данные можно записывать
1	ARRB_EN	Разрешение мгновенного обновления ARR 0 – ARR будет перезаписан в момент записи в ARR 1 – ARR будет перезаписан при завершении счета CNT
0	CNT_EN	Разрешение работы таймера 0 – таймер отключен 1 – таймер включен

### TIMx\_CCRy

Регистр сравнения, захвата для 'у' канала таймера

Номер	31	15	0
Доступ	R/W	R/W	R/W
Сброс	0	0	0
			CCR[15:0]

**Таблица 104**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16	-	Зарезервировано
15...0	CCR[15:0]	Значение CCR, с которым сравнивается CNT при работе в ШИМ режиме. Значение CNT при котором произошел факт захвата события, в режиме захвата

### TIMx\_CCRy1

Регистр сравнения, захвата для 'у' канала таймера

Номер	31	15	0
Доступ	R/W	R/W	R/W
Сброс	0	0	0
			CCR1[15:0]

**Таблица 105**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16	-	Зарезервировано
15...0	CCR1[15:0]	Значение CCR1, с которым сравнивается CNT при работе в ШИМ режиме. Значение CNT при котором произошел факт захвата события, в режиме захвата

### TIMx\_CNy\_CNTRL

Регистр управления для 'у' канала таймера

Номер	14	13	12	11...9	8	7...6	5...4	3...0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	000	0	00	00	0000
	WR CMPL	ETREN	BRKEN	OCCM [2:0]	OCCE	CHPSC [1:0]	CHSEL [1:0]	CHFLTR [3:0]

Номер	31	15
Доступ	U	R/W
Сброс	0	0
	-	CAP nPWM

**Таблица 106**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16	-	Зарезервировано
15	CAP nPWM	Режим работы канала Захват или ШИМ 1 – канал работает в режиме Захват 0 – канал работает в режиме ШИМ
14	WR CMPL	Флаг окончания записи, при задании нового значения регистра CCR 1 – данные не записаны и идет запись 0 – новые данные можно записывать
13	ETREN	Разрешения сброса по выводу ETR 0 – запрещен сброс 1 – разрешен
12	BRKEN	Разрешение сброса по выводу BRK

		0 – запрещен сброс 1 – разрешен
11...9	OSCM[2:0]	<p>Формат выработки сигнала REF в режиме ШИМ</p> <p>Если CCR1_EN = 0:</p> <p>000 – всегда 0</p> <p>001 – 1, если CNT = CCR</p> <p>010 – 0, если CNT = CCR</p> <p>011 – переключение REF, если CNT = CCR</p> <p>100 – всегда 0</p> <p>101 – всегда 1</p> <p>110 – 1, если DIR= 0 (счет вверх), CNT &lt; CCR, иначе 0</p> <p>          0, если DIR= 1 (счет вниз), CNT &lt; CCR, иначе 1</p> <p>111 – 0, если DIR= 0 (счет вверх), CNT &lt; CCR, иначе 1</p> <p>          1, если DIR= 1 (счет вниз), CNT &lt; CCR, иначе 0</p> <p>Если CCR1_EN = 1:</p> <p>000 – всегда 0</p> <p>001 – 1, если CNT = CCR или CNT = CCR1</p> <p>010 – 0, если CNT = CCR или CNT = CCR1</p> <p>011 – переключение REF, если CNT = CCR или CNT = CCR1</p> <p>100 – всегда 0</p> <p>101 – всегда 1</p> <p>110 – 1, если DIR = 0 (счет вверх), CCR1 &lt; CNT &lt; CCR, иначе 0</p> <p>          0, если DIR= 1 (счет вниз), CCR &lt; CNT &lt; CCR1, иначе 1</p> <p>111 – 0, если DIR = 0 (счет вверх), CCR1 &lt; CNT &lt; CCR, иначе 1</p> <p>          1, если DIR = 1 (счет вниз), CCR &lt; CNT &lt; CCR1, иначе 0</p>
8	OCCE	<p>Разрешение работы ETR</p> <p>0 – запрет ETR</p> <p>1 – разрешение ETR</p>
7...6	CHPSC[1:0]	<p>Предварительный делитель входного канала</p> <p>00 – нет деления</p> <p>01 – /2</p> <p>10 – /4</p> <p>11 – /8</p>
5...4	CHSEL[1:0]	<p>Выбор события по входному каналу</p> <p>00 – положительный фронт</p> <p>01 – отрицательный фронт</p>
3...0	CHFLTR[3:0]	<p>Сигнал зафиксирован:</p> <p>0000 – в 1 триггере на частоте TIM_CLK</p> <p>0001 – в 2 триггерах на частоте TIM_CLK</p> <p>0010 – в 4 триггерах на частоте TIM_CLK</p> <p>0011 – в 8 триггерах на частоте TIM_CLK</p> <p>0100 – в 6 триггерах на частоте FDTs/2</p>

		0101 – в 8 триггерах на частоте FDTS/2 0110 – в 6 триггерах на частоте FDTS/4 0111 – в 8 триггерах на частоте FDTS/4 1000 – в 6 триггерах на частоте FDTS/8 1001 – в 8 триггерах на частоте FDTS/8 1010 – в 5 триггерах на частоте FDTS/16 1011 – в 6 триггерах на частоте FDTS/16 1100 – в 8 триггерах на частоте FDTS/16 1101 – в 5 триггерах на частоте FDTS/32 1110 – в 6 триггерах на частоте FDTS/32 1111 – в 8 триггерах на частоте FDTS/32
--	--	--

### TIMx\_CHy\_CNTRL1

Регистр управления 1 для 'у' канала таймера

Номер	31...13	12	11...10	9...8	7...5	4	3...2	1...0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	00	00	0	0	00	00
	-	NINV	NSELO [1:0]	NSELOE [1:0]	-	INV	SELO [1:0]	SELOE [1:0]

**Таблица 107**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..13	-	Зарезервировано
12	NINV	Режим выходной инверсии 0 – выход не инвертируется 1 – выход инвертируется
11..10	NSELO[1:0]	Режим работы выхода канала 00 – всегда на выход выдается 0, канал на выход не работает 01 – всегда на выход выдается 1, канал всегда работает на выход 10 – на выход выдается сигнал REF. 11 - на выход выдается сигнал с DTG.
9...8	NSELOE[1:0]	Режим работы канала на выход 00 – всегда на ОЕ выдается 0, канал на выход не работает 01 – всегда на ОЕ выдается 1, канал всегда работает на выход 10 – на ОЕ выдается сигнал REF, при REF = 0 вход, при REF = 1 выход. 11 - на ОЕ выдается сигнал с DTG, при CHn = 0 вход, при CHn = 1 выход
7...5	-	
4	INV	Режим выходной инверсии 0 – выход не инвертируется 1 – выход инвертируется



3...2	SELO[1:0]	Режим работы выхода канала 00 – всегда на выход выдается 0, канал на выход не работает 01 – всегда на выход выдается 1, канал всегда работает на выход 10 – на выход выдается сигнал REF. 11 - на выход выдается сигнал с DTG.
1...0	SELOE[1:0]	Режим работы канала на выход 00 – всегда на OE выдается 0, канал на выход не работает 01 – всегда на OE выдается 1, канал всегда работает на выход 10 – на OE выдается сигнал REF, при REF = 0 вход, при REF = 1 выход. 11 - на OE выдается сигнал с DTG, при CH = 0 вход, при CH = 1 выход

### TIMx\_CHy\_CNTRL2

Регистр управления 2 для 'у' канала таймера

Номер	31				4	3	2	1...0
Доступ	U	U	U	U	U	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	00
	-	-	-	-	-	CRRRLD	CCR1_EN	CHSEL [1:0]

**Таблица 108**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..4	-	Зарезервировано
3	CRRRLD	Разрешение обновления регистров CCR и CCR1 0 – обновление возможно в любой момент времени 1 – обновление будет осуществлено только при CNT = 0
2	CCR1_EN	Разрешение работы регистра CCR1 0 – CCR1 не используется 1 – CCR1 используется
1...0	CHSEL1[1:0]	Выбор события по входному каналу для CAP1 00 – положительный фронт по CHi 01 – отрицательный фронт по CHi

### TIMx\_CHy\_DTG

Регистр управления DTG

Номер	31	16	15...8	7...5	4	3...0	
Доступ	U	U	R/W	U	R/W	R/W	
Сброс	0	0	0000000 0	000	0	0000	
	-		-	DTG[7:0]	-	EDTS	DTGx [3:0]

**Таблица 109**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16	-	Зарезервировано
15...8	DTGx[7:0]	Основной делитель частоты Задержка DTGdel = DTGx*(DTG+1).
7...5	-	Зарезервировано
4	EDTS	Частота работы DTG 0 – TIM_CLK 1 – FDTS
3...0	DTG [3:0]	Предварительный делитель частоты DTG

### TIMx\_BRKETR\_CNTRL

Регистр управления входом BRK и ETR

Номер	31	8	7...4	3...2	1	0	
Доступ	U	U	R/W	R/W	R/W	R/W	
Сброс			0000	00	0	0	
	-		-	ETR FILTER [3:0]	ETR PSC [1:0]	ETR INV	BRK INV

**Таблица 110**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..8	-	Зарезервировано
7...4	ETR FILTER[3:0]	Цифровой фильтр на входе ETR. Сигнал зафиксирован: 0000 – в 1 триггере на частоте TIM_CLK 0001 – в 2 триггерах на частоте TIM_CLK 0010 – в 4 триггерах на частоте TIM_CLK 0011 – в 8 триггерах на частоте TIM_CLK 0100 – в 6 триггерах на частоте FDTS/2 0101 – в 8 триггерах на частоте FDTS/2 0110 – в 6 триггерах на частоте FDTS/4 0111 – в 8 триггерах на частоте FDTS/4 1000 – в 6 триггерах на частоте FDTS/8 1001 – в 8 триггерах на частоте FDTS/8

		1010 – в 5 триггерах на частоте FDTS/16 1011 – в 6 триггерах на частоте FDTS/16 1100 – в 8 триггерах на частоте FDTS/16 1101 – в 5 триггерах на частоте FDTS/32 1110 – в 6 триггерах на частоте FDTS/32 1111 – в 8 триггерах на частоте FDTS/32
3...2	ETRPSC[1:0]	Асинхронный пред. делитель внешней частоты 00 – без деления 01 - /2 10 - /4 11 - /8
1	ETR INV	Инверсия входа ETR 0 – без инверсии 1 – инверсия
0	BRK INV	Инверсия входа BRK 0 – без инверсии 1 – инверсия

## TIMx\_STATUS

Регистр статуса таймера

Номер	13	9	5	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0
	CCR CAP1 EVENT	CCR REF EVENT	CCR CAP EVENT	BRK EVENT	ETR FE EVENT	ETR RE EVENT	CNT ARR EVENT	CNT ZERO EVENT

**Таблица 111**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..14	-	Зарезервировано
13	CCR CAP1 EVENT	Событие переднего фронта на входе CAP1 каналов таймера 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события.
12..10	-	Зарезервировано
9	CCR REF EVENT	Событие переднего фронта на выходе REF каналов таймера 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события.

8..6	-	Зарезервировано
5	CCR CAP EVENT	Событие переднего фронта на входе CAP каналов таймера 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события.
4	BRK EVENT	Триггерированное по PCLK состояние входа BRK, 0 – BRK == 0 1 – BRK == 1 Сбрасывается записью 0, при условии наличия 0 на входе BRK
3	ETR FE EVENT	Событие заднего фронта на входе ETR 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события.
2	ETR RE EVENT	Событие переднего фронта на входе ETR 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события.
1	CNT ARR EVENT	Событие совпадения CNT с ARR 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием совпадения, приоритет у нового события. Если с момента совпадения до момента программного сброса CNT и ARR не изменили состояния, то флаг повторно не взводится.
0	CNT ZERO EVENT	Событие совпадения CNT с нулем 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием совпадения, приоритет у нового события. Если с момента совпадения до момента программного сброса CNT не изменил состояния, то флаг повторно не взводится.

### TIMx\_IE

Регистр разрешения прерывания таймера

Номер	13	9	5	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0

CCR CAP1 EVENT IE	CCR REF EVENT IE	CCR CAP EVENT IE	BRK EVENT IE	ETR FE EVENT IE	ETR RE EVENT IE	CNT ARR EVENT IE	CNT ZERO EVENT IE
----------------------------	---------------------------	---------------------------	--------------------	--------------------------	--------------------------	---------------------------	----------------------------

**Таблица 112**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..14	-	Зарезервировано
13	CCR CAP1 EVENT IE	Флаг разрешения прерывания по событию переднего фронта на выходе CAP1 каналов таймера 0 – нет прерывания 1 – прерывание разрешено
12..10	-	Зарезервировано
9	CCR REF EVENT IE	Флаг разрешения прерывания по событию переднего фронта на выходе REF каналов таймера 0 – нет прерывания 1 – прерывание разрешено
8..6	-	Зарезервировано
5	CCR CAP EVENT IE	Флаг разрешения прерывания по событию переднего фронта на выходе CAP каналов таймера 0 – нет прерывания 1 – прерывание разрешено
4	BRK EVENT IE	Флаг разрешения по триггерированному по PCLK состоянию входа BRK, 0 – нет прерывания 1 – прерывание разрешено
3	ETR FE EVENT IE	Флаг разрешения прерывания по заднему фронту на входе ETR 0 – нет прерывания 1 – прерывание разрешено
2	ETR RE EVENT IE	Флаг разрешения прерывания по переднему фронту на входе ETR 0 – нет прерывания 1 – прерывание разрешено
1	CNT ARR EVENT IE	Флаг разрешения прерывания по событию совпадения CNT и ARR 0 – нет прерывания 1 – прерывание разрешено
0	CNT ZERO EVENT IE	Флаг разрешения прерывания по событию совпадения CNT и нуля 0 – нет прерывания 1 – прерывание разрешено

## Контроллер АЦП

В микроконтроллере реализован 12-ти разрядный АЦП. С помощью данного АЦП можно оцифровать сигнал с 8 внешних аналоговых выводов порта D и двух внутренних каналов, на которые выводится датчик температуры и источник опорного напряжения. Скорость выборки составляет до 500 тысяч преобразований в секунду.

Контроллер АЦП позволяет:

- оцифровать один из 8 внешних каналов;
- оцифровать значение встроенного датчика температуры;
- оцифровать значение встроенного источника опорного напряжения;
- осуществить автоматический опрос заданных каналов;
- выработать прерывание при выходе оцифрованного значения за заданные пределы.

Для осуществления преобразования требуется 28 тактов синхронизации CLK. В качестве синхросигнала может выступать частота процессора CPU\_CLK либо частота ADC\_CLK формируемая в блоке «Сигналы тактовой частоты». Выбор частоты осуществляется с помощью бита Cfg\_REG\_CLKS. В контроллере АЦП частота может быть поделена с помощью битов Cfg\_REG\_DIVCLK[3:0]. Максимальная частота CLK не может превышать 14 МГц.

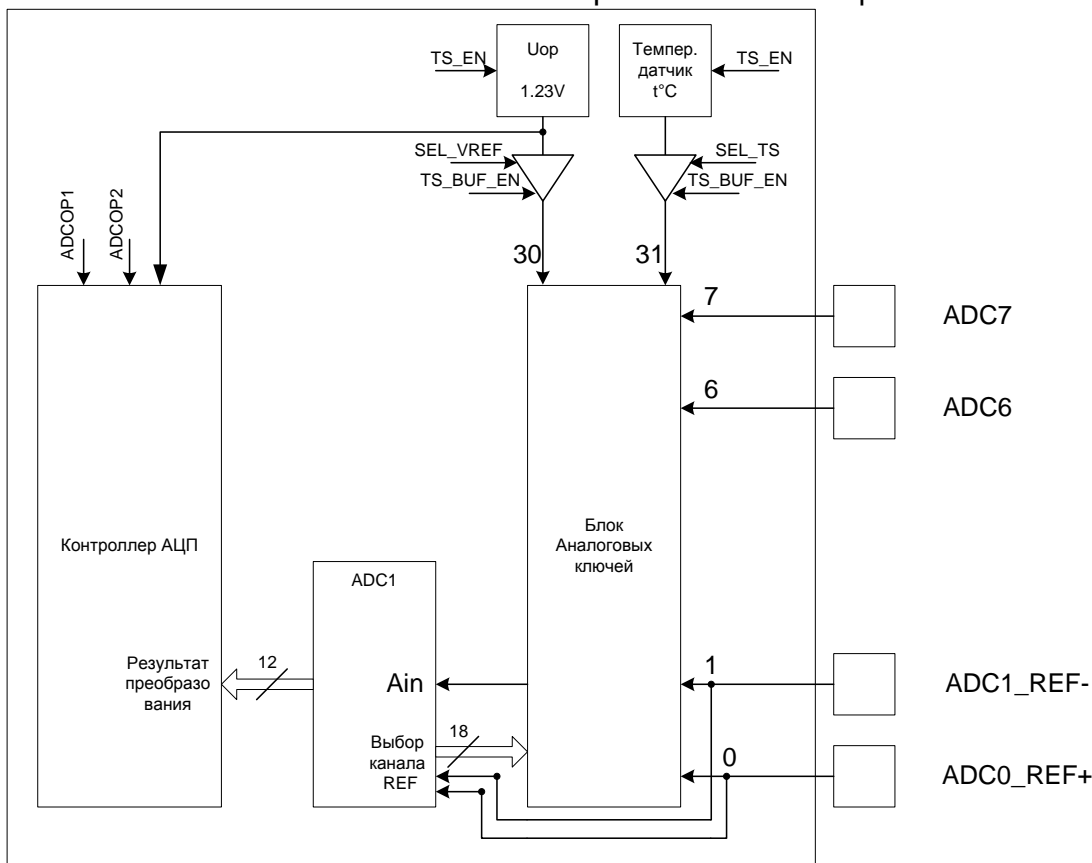


Рис. 61

Для включения АЦП необходимо установить бит Cfg\_REG\_ADON. Для снижения тока потребления вместо собственного источника опорного напряжения в АЦП может использоваться источник датчика температуры. Для этого необходимо включить блок датчика температуры и источник опорного напряжения, установив бит TS\_EN (см. Таблица 115) в 1. После включения можно

использовать источник опорного напряжения для АЦП вместо его собственного. Для этого необходимо установить биты ADC1\_OP в единицу. Для преобразования необходимо, чтобы выводы, используемые АЦП в порте D, были сконфигурированы как аналоговые и были отключены какие либо внутренние подтяжки.

### **Преобразование внешнего канала**

В регистре ADC1\_CFG в битах Cfg\_REG\_CHS[4:0] необходимо задать соответствующий выводу номер канала. Преобразование может осуществляться при внутренней опоре бит Cfg\_M\_REF = 0 и внешней Cfg\_M\_REF = 1, в этом случае опора берется с выводов ADC0\_REF+ и ADC1\_REF-. Биты Cfg\_REG\_CHCH, Cfg\_REG\_RNGC, Cfg\_REG\_SAMPLE, TS\_BUF\_EN, SEL\_VREF, SEL\_TS и Cfg\_Sync\_Conver должны быть сброшены.

Для начала преобразования необходимо записать 1 в бит Cfg\_REG\_GO.

После завершения преобразования будет взведен бит Flg\_REG\_EOCIF в регистре ADC1\_STATUS. А в регистре ADC1\_RESULT будет результат преобразования.

После считывания результата бит Flg\_REG\_EOCIF сбросится.

Если после первого преобразования результат не был считан и было выполнено второе преобразование, то в регистре результата ADC1\_RESULT будет значение от последнего преобразования, а помимо бита Flg\_REG\_EOCIF будет взведен бит Flg\_REG\_OVERWRITE. Флаг Flg\_REG\_OVERWRITE может быть сброшен только записью в регистр ADC1\_STATUS.

### **Последовательное преобразование нескольких каналов**

Для автоматического последовательного преобразования нескольких каналов или одного канала в регистре ADC1\_CHSEL необходимо установить единицы в битах соответствующих необходимым для преобразования каналам. Преобразование может осуществляться при внутренней опоре бит Cfg\_M\_REF = 0 и внешней Cfg\_M\_REF = 1, в этом случае опора берется с выводов ADC0\_REF+ и ADC1\_REF-. Биты, Cfg\_REG\_RNGC, TS\_BUF\_EN, SEL\_VREF, SEL\_TS и Cfg\_Sync\_Conver должны быть сброшены, а Cfg\_REG\_SAMPLE и Cfg\_REG\_CHCH должны быть установлены. С помощью битов Delay\_GO можно задать паузу между преобразованиями при переборе каналов. Для начала преобразования необходимо записать 1 в бит Cfg\_REG\_GO.

После завершения преобразования будет взведен бит Flg\_REG\_EOCIF в регистре ADC1\_STATUS. А в регистре ADC1\_RESULT будет результат преобразования.

После считывания результата бит Flg\_REG\_EOCIF сбросится.

Если после первого преобразования результат не был считан и было выполнено второе преобразование, то в регистре результата ADC1\_RESULT будет значение от последнего преобразования, а помимо бита Flg\_REG\_EOCIF будет взведен бит Flg\_REG\_OVERWRITE. Флаг Flg\_REG\_OVERWRITE может быть сброшен только записью в регистр ADC1\_STATUS.

Для последовательного преобразования одного и того же канала можно в регистре ADC1\_CHSEL выбрать только один канал и установить бит Cfg\_REG\_CHCH в 1, либо установить номер канала в битах Cfg\_REG\_CHS[4:0] и сбросить бит Cfg\_REG\_CHCH в 0. В этом случае процесс последовательного преобразования будет выполняться только для данного канала. Последовательное преобразование значения датчика температуры и источника

опорного напряжения могут выполняться только в режиме последовательного преобразования одного канала.

### **Преобразование с контролем границ**

При необходимости отслеживать нахождение оцифрованных значений в допустимых пределах можно задать нижнюю и верхнюю допустимые границы в регистрах ADC1\_L\_LEVEL и ADC1\_H\_LEVEL. При этом если установлен бит Cfg\_REG\_RNGC, то в случае если результат преобразования выходит за границы выставляется флаг Flg\_REG\_AWOIFEN. А в регистре результата будет полученное значение.

### **Датчик опорного напряжения**

С помощью АЦП можно осуществить преобразования источника опорного напряжения. Для этого необходимо включить блок датчика температуры и источник опорного напряжения установив бит TS\_EN в 1. После включения можно использовать источник опорного напряжения для АЦП вместо его собственного, что позволяет снизить ток потребления. Для этого необходимо установить биты ADC1\_OP в единицу. Для выбора источника опорного напряжения в качестве источника для преобразования необходимо в битах Cfg\_REG\_CHS установить значение 30 канала. Установить биты SEL\_VREF\_BUF и SEL\_VREF. После чего можно запустить процесс преобразования. Для начала преобразования необходимо записать 1 в бит Cfg\_REG\_GO.

После завершения преобразования будет взведен бит Flg\_REG\_EOCIF в регистре ADC1\_STATUS. А в регистре ADC1\_RESULT будет результат преобразования.

После считывания результата бит Flg\_REG\_EOCIF сбросится.

Если после первого преобразования результат не был считан и было выполнено второе преобразование, то в регистре результата ADC1\_RESULT будет значение от последнего преобразования, а помимо бита Flg\_REG\_EOCIF будет взведен бит Flg\_REG\_OVERWRITE. Флаг Flg\_REG\_OVERWRITE может быть сброшен только записью в регистр ADC1\_STATUS.

Для последовательного преобразования только источника опорного напряжения можно в регистре ADC1\_CHSEL выбрать только 30 канал и установить бит Cfg\_REG\_CHCH в 1, либо установить номер 30-го канала в битах Cfg\_REG\_CHS[4:0] и сбросить бит Cfg\_REG\_CHCH в 0. В этом случае процесс последовательного преобразования будет выполняться только для данного канала. При этом должны быть так же установлены биты SEL\_VREF\_BUF и SEL\_VREF.

### **Датчик температуры**

С помощью первого АЦП можно осуществить преобразования датчика опорного напряжения. Для этого необходимо включить блок датчика температуры и источник опорного напряжения установив бит TS\_EN в 1. После включения можно использовать источник опорного напряжения для АЦП вместо его собственного, что позволяет снизить ток потребления. Для этого необходимо установить биты ADC1\_OP в единицу. Для выбора датчика температуры в качестве источника для преобразования необходимо в битах Cfg\_REG\_CHS установить значение 31 канала. Установить биты TS\_BUF\_EN и SEL\_TS. После чего можно запустить процесс преобразования. Для начала преобразования необходимо записать 1 в бит Cfg\_REG\_GO.



После завершения преобразования будет взведен бит Flg\_REG\_EOCIF в регистре ADC1\_STATUS. А в регистре ADC1\_RESULT будет результат преобразования.

После считывания результата бит Flg\_REG\_EOCIF сбросится.

Если после первого преобразования результат не был считан и было выполнено второе преобразование, то в регистре результата ADC1\_RESULT будет значение от последнего преобразования, а помимо бита Flg\_REG\_EOCIF будет взведен бит Flg\_REG\_OVERWRITE. Флаг Flg\_REG\_OVERWRITE может быть сброшен только записью в регистр ADC1\_STATUS.

Для последовательного преобразования только датчика температуры можно в регистре ADC1\_CHSEL выбрать только 31 канал и установить бит Cfg\_REG\_CHCH в 1, либо установить номер 31-го канала в битах Cfg\_REG\_CHS[4:0] и сбросить бит Cfg\_REG\_CHCH в 0. В этом случае процесс последовательного преобразования будет выполняться только для данного канала. При этом должны быть так же установлены биты TS\_BUF\_EN и SEL\_TS.

**Время заряда внутренней емкости**

Процесс преобразования состоит из двух этапов: сначала происходит заряд внутренней емкости до уровня внешнего сигнала, и затем происходит преобразование уровня заряда внутренней емкости в цифровой вид. Таким образом, для точного преобразования внешнего сигнала в цифровой вид, за время первого этапа внутренняя емкость должна зарядиться до уровня внешнего сигнала. Это время определяется соотношением номинальной внутренней емкости, входным сопротивлением тракта АЦП и выходным сопротивлением источника сигнала. Приведенная ниже формула позволяет определить максимальное выходное сопротивление источника  $R_{AIN}$  для обеспечения качественного преобразования:

$$R_{AIN} < (T_S / (f_{C\_ADC} * C_{ADC} * \ln(2^N))) - R_{ADC} \quad 10$$

- где:  $T_S$  - время заряда внутренней емкости в тактах  
 $f_{C\_ADC}$  - рабочая частота АЦП  
 $C_{ADC}$  - внутренняя емкость АЦП (~15-20пФ)  
 $N$  - требуемая точность в разрядах  
 $R_{ADC}$  - входное сопротивление тракта АЦП (~500 Ом)

Если необходимо обеспечить преобразование с точностью 12 разрядов  $\pm 1/4$  LSB, то  $N = 14$ . Если необходимо обеспечить преобразование с точностью 10 разрядов  $\pm 1$  LSB, то  $N=10$ . Время заряда  $T_S$  = определяется битами DelayGo[2:0] и схемой самого АЦП и представлена в таблице. Время зарядки внутренней емкости задается битами DelayGo[2:0] определяется в тактах CPU\_CLK, независимо от того на какой частоте ADC\_CLK или CPU\_CLK идет само преобразование.

Время заряда внутренней емкости АЦП и время преобразования

**Таблица 113**

DelayGo[2:0]	Дополнительная задержка перед началом преобразования	Общее время TS заряда емкости АЦП перед началом преобразования	Общее время преобразования АЦП
--------------	--	--	--------------------------------

000	1 x CPU_CLK	4 x CLK + 1 x CPU_CLK	28 x CLK + 1 x CPU_CLK
001	2 x CPU_CLK	4 x CLK + 2 x CPU_CLK	28 x CLK + 2 x CPU_CLK
010	3 x CPU_CLK	4 x CLK + 3 x CPU_CLK	28 x CLK + 3 x CPU_CLK
011	4 x CPU_CLK	4 x CLK + 4 x CPU_CLK	28 x CLK + 4 x CPU_CLK
100	5 x CPU_CLK	4 x CLK + 5 x CPU_CLK	28 x CLK + 5 x CPU_CLK
101	6 x CPU_CLK	4 x CLK + 6 x CPU_CLK	28 x CLK + 6 x CPU_CLK
110	7 x CPU_CLK	4 x CLK + 7 x CPU_CLK	28 x CLK + 7 x CPU_CLK
111	8 x CPU_CLK	4 x CLK + 8 x CPU_CLK	28 x CLK + 8 x CPU_CLK

Помимо точности определяемой временем зарядки внутренней емкости АЦП точность преобразования имеет ошибки связанные с технологическими разбросами схемы и шумами и определяемые параметрами  $E_{DLADC}$ ,  $E_{ILADC}$  и  $E_{OFFADC}$ .

Для корректного задания режимов работы АЦП в регистре ADC1\_CFG необходимо сделать до задания бита Go, иначе новая конфигурация будет действовать со следующего преобразования.

**Описание регистров блока контроллера АЦП**

**Таблица 114**

Базовый Адрес	Название	Описание
0x4004_0000	ADC	Контроллер ADC
Смещение		
0x00	ADC1_CFG	Регистр управления ADC
0x04	ADC2_CFG	Регистр управления ADC
0x08	ADC1_H_LEVEL	Регистр верхней границы ADC
0x10	ADC1_L_LEVEL	Регистр нижней границы ADC
0x18	ADC1_RESULT	Регистр результата ADC
0x20	ADC1_STATUS	Регистр статуса ADC
0x28	ADC1_CHSEL	Регистр выбора каналов перебора ADC
0x30	ADC1_TRIM	Регистр настройки термодатчика

**ADCx\_CFG**

Номер	11	10	9	8...4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0

Cfg M_REF	Cfg REG RNGC	Cfg REG CHCH	Cfg REG CHS [4:0]	Cfg REG SAMPLE	Cfg REG CLKS	Cfg REG GO	Cfg REG ADON
-----------	--------------	--------------	-------------------	----------------	--------------	------------	--------------

Номер	27...25	24...21	20	19	18	17	16	15...12
Доступ	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0

Delay Go [2:0]	TR[3:0]	SEL VREF	SEL TS	TS_BUF EN	TS_EN / ADC1 OP	Cfg Sync Conver	Cfg REG DIVCLK [3:0]
----------------	---------	----------	--------	-----------	-----------------	-----------------	----------------------

Номер	31...28
Доступ	R/W
Сброс	0

Delay ADC [3:0]
-----------------

**Таблица 115**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..28	Delay ADC [3:0]	Задержка между началом преобразования ADC1 при последовательном переборе, либо работе на один канал. 0000 - 0 тактов CLK 0001 - 1 такт CLK ... 1111 - 15 тактов CLK
27..25	Delay Go [2:0]	Задержка перед началом следующего преобразования после завершения предыдущего при последовательном переборе каналов 000 - 0 тактов CLK 001 - 1 такт CLK ... 111 - 7 тактов CLK
24..21	-	Зарезервировано
20	SEL VREF	Выбор для оцифровки источника опорного напряжения на 1.23В 0 – не выбран 1 – выбран Должен использоваться совместно с выбором канала Cfg_REG_CHS = 30.
19	SEL TS	Выбор для оцифровки датчика температуры 0 – не выбран 1 – выбран Должен использоваться совместно с выбором канала Cfg_REG_CHS = 31.
18	TS_BUF EN	В регистре ADC1_CFG. Включение выходного усилителя для датчика температуры 0 – выключен 1 – включен Используется при TS_EN = 1. Для уменьшения тока

		потребления.
17	TS EN	В регистре ADC1_CFG. Включение датчика температуры и источника опорного напряжения 0 – выключен 1 – включен При включении датчика температуры и источника опорного напряжения выходной сигнал стабилизируется в течении времени Tstb.
17	ADC1 OP	В регистре ADC2_CFG. Выбор источника опорного напряжения 1.23В 0 – внутренний (не точный) 1 – от датчика температуры (точный)
16	Cfg Sync Conver	Записывать всегда ноль
15..12	Cfg REG DIVCLK [3:0]	Выбор коэффициента деления входной частоты 0000 – CLK 0001 – CLK/2 0010 – CLK/4 0011 – CLK/8 ... 1111 – CLK/32768
11	Cfg M_REF	Выбор источника опорных напряжений 0 – внутренне опорное напряжение (от AUdd и AUss) 1 – внешнее опорное напряжение (от Uref+ и Uref-)
10	Cfg REG RNGC	Разрешение автоматического контролирования уровней 1 – Разрешено, выработка прерывания при выходе за диапазон в регистрах границы обработки 0 – не разрешено
9	Cfg REG CHCH	Выбор переключения каналов 1 – переключение включено (перебираются каналы выбранные в регистре выбора канала) 0 – используется только выбранный канал
8...4	Cfg REG CHS [4:0]	Выбор аналогового канала, по которому поступает сигнал для преобразования. 00000 – 0 канал 00001 – 1 канал ... 11111 – 31 канал
3	Cfg REG SAMPLE	Выбор способа запуска АЦП. 1 – последовательное, автоматический запуск после завершения предыдущего преобразования 0 – одиночное.
2	Cfg REG CLKS	Выбор источника синхросигнала CLK работы ADC 1 – ACLK 0 – PCLK
1	Cfg REG GO	Начало преобразования Запись “1” начинает процесс преобразования, сбрасывается автоматически
0	Cfg	Включение АЦП

	REG ADON	1 – включено 0 – выключено
--	-------------	-------------------------------

**ADC1\_H\_LEVEL**

Номер	31	12	11...0
Доступ	U	U	R/W
Сброс	0	0	0

-	REG H LEVEL [11:0]
---	--------------------------

**Таблица 116**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..12	-	Зарезервировано
11...0	REG H LEVEL [11:0]	Верхняя граница зоны допуска.

**ADC1\_L\_LEVEL**

Номер	31	12	11...0
Доступ	U	U	R/W
Сброс	0	0	0

-	REG L LEVEL [11:0]
---	--------------------------

**Таблица 117**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..12	-	Зарезервировано
11...0	REG L LEVEL [11:0]	Нижняя граница зоны допуска.

**ADC1\_RESULT**

Номер	31	21	20...16	15...12	11...0
Доступ	U	U	RO	U	RO
Сброс	0	0	0	0	0

-	-	-	CHANNEL [11:0]	-	RESULT [11:0]
---	---	---	----------------	---	---------------

**Таблица 118**

№	Функциональное	Расшифровка функционального имени бита,
---	----------------	---

	имя бита	краткое описание назначения и принимаемых значений.
31..12	-	Зарезервировано
20..16	CHANNEL [11:0]	Канал результата преобразования
15..12	-	Зарезервировано
11...0	RESULT [11:0]	Значение результата преобразования

### ADC1\_STATUS

Номер	3	5	4	3	2	1	0
Доступ	U	U	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0
	-	-	ECOIF IE	AWOIFI E	Flg REG EOCIF	Flg REG AWOIFE N	Flg REG OVERW RITE

**Таблица 119**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31...5	-	Зарезервировано
4	ECOIF_IE	Флаг разрешения генерирования прерывания по событию Flg_REG_ECOIF 0 – прерывания не генерируется 1 – прерывание генерируется
3	AWOIF_IE	Флаг разрешения генерирования прерывания по событию Flg_REG_AWOIFEN 0 – прерывания не генерируется 1 – прерывание генерируется
2	Flg REG EOCIF	Флаг выставляется, когда закончено преобразования и данные еще не считаны. Очищается считыванием результата из регистра ADCx_RESULT. 1 – есть готовый результат преобразования 0 – нет результата
1	Flg REG AWOIFEN	Флаг выставляется, когда результат преобразования выше верхней или ниже нижней границы автоматического контролирования уровней. Очищается считывание результата из регистра ADCx_RESULT. 0 – результат в допустимой зоне 1 – вне допустимой зоны
0	Flg REG OVERWRITE	Данные в регистре результата были перезаписаны, данный флаг сбрасывается только при записи в регистр флагов. 0 – не было события перезаписи не считанного результата 1 – был результат преобразования, который не был

		считан
--	--	--------

**ADC1\_CHSEL**

Номер	31	0
Доступ	R/W	R/W
Сброс	0	0
SI_Ch_Ch_REF[31:0]		

**Таблица 120**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	SI_Ch_Ch_REF[31:0]	Выбор каналов автоматического перебора 0 в соответствующем бите канал не участвует в переборе 1 –канал участвует в переборе

**ADC1\_TRIM**

Номер	31..7	6	5	4	3	2	1	0
Доступ	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	1	0	0	0	0	0
-		SEL_VREF_BUF	TS_TRIM[4:0]					0

**Таблица 121**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..7	-	Зарезервировано
6	SEL_VREF_BUF	Включение выходного усилителя для источника опорного напряжения 0 – выключен 1 – включен Используется при TS_EN = 1. Для уменьшения тока потребления.
5..1	TS_TRIM[4:0]	Подстройка опорного напряжения
0	-	Зарезервировано

### Контроллер I2C интерфейса

I2C является двухпроводным, двунаправленным последовательным каналом связи с простым и эффективным методом обмена данными между устройствами. Интерфейс применяется, когда надо организовать обмен на коротком расстоянии между несколькими устройствами. Стандарт интерфейса I2C является многомастерным с обнаружением коллизий и арбитражем, исключающим потерю данных при обмене, когда два или более мастера пытаются осуществить передачу одновременно.

Интерфейс работает на 3 скоростях.

Нормальная: 100Kbps

Быстрая: 400Kbps

Очень быстрая: 3.5Mbps

### Конфигурация системы

I2C системы используют последовательную линию данных SDA и линию тактового сигнала SCL. Все устройства, подсоединенные к этим двум линиям, должны работать в режиме открытого стока, обеспечивая тем самым создание на линии «проводного И» за счет внешних резисторов подтяжки обеих линий к питанию.

Передача данных между мастером и ведомым осуществляется по линии SDA и синхронизируется по линии SCL. После завершения передачи информации осуществляется передача в обратную сторону одного бита подтверждения. Каждый принимаемый бит фиксируется принимающей стороной при высоком уровне SCL и может изменяться передатчиком при низком уровне. Изменение линии SDA при высоком уровне SCL является командным состоянием (см. раздел «Сигнал START» и «Сигнал STOP»)

### Протокол I2C

Нормальная передача по интерфейсу I2C содержит 4 фазы:

- Сигнал START
- Передача Адреса
- Передача Данных
- Сигнал STOP

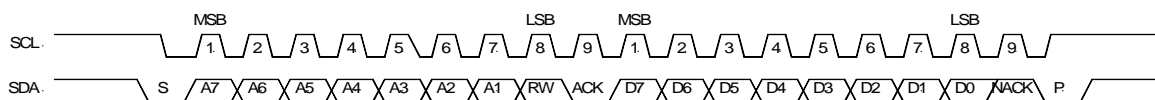


Рис. 62 Передача по I2C

### Сигнал START

Когда шина находится в свободном состоянии, т.е. не одно из устройств не осуществляет передачи (на линиях SCL и SDA высокий уровень), мастер может инициализировать процесс передачи через создание сигнала START на линии. Сигнал START или S бит задается, когда уровень на линии SDA переходит из высокого в низкий при высоком уровне на линии SCL. Появление сигнала START не означает начала передачи данных.

Повторный сигнал START является обычным сигналом START, но без предварительно сгенерированного до этого сигнала STOP. Мастер может



использовать метод для начала соединения с другим ведомым или с тем же ведомым, но с изменением режима работы (например, чтение после записи или наоборот) без перевода шины в свободное состояние.

Контроллер интерфейса генерирует сигнал START при записи единицы в бит START регистра I2C\_CMD (см. раздел «Описание регистров контроллера I2C») при установленных битах RD или WR. В зависимости от состояния линии SCL генерируется либо сигнал START, либо повторный сигнал START.

### **Передача адреса**

Первым байтом данных, передаваемым мастером сразу после сигнала START, является адрес ведомого. Это 7-ми битный адрес и следующий за ним бит RW. Бит RW определяет дальнейшее направление передачи данных. В системе не может быть несколько ведомых устройств с одним адресом. Ведомое устройство, у которого совпадает адрес с адресом в сообщении, подтверждает прием, выставляя ACK и опуская линию SDA в низкий уровень на 9-й SCL тактовый импульс. Контроллер также поддерживает 10-битный адрес путем генерации двух циклов передачи адреса.

Процесс выдачи адреса выполняется как цикл записи. Необходимо записать адрес ведомого в регистр I2C\_TXD (см. раздел «Описание регистров контроллера I2C») и установить бит WR в регистре I2C\_CMD. Контроллер осуществит передачу адреса в линию.

### **Передача данных**

После успешного подтверждения приема адреса одним ведомым устройством может быть начата передача данных в направлении, задаваемым битом RW в посылке мастера. Каждый передаваемый бит подтверждается ACK на 9-й SCL тактовый импульс. Если ведомое устройство выдало NACK (нет подтверждения), то мастер может сгенерировать либо сигнал STOP для прекращения передачи, либо повторный сигнал START для начала нового цикла передачи.

Если мастер является принимающим устройством и выдает NACK, то ведомое устройство отпускает линию SDA и мастер может сгенерировать сигнал STOP или повторный сигнал START.

Для записи данных в ведомое устройство запишите данные в регистр I2C\_TXD и установите бит WR. Для чтения данных из устройства установите бит RD. На время выполнения передачи контроллер интерфейса выставляет флаг TR\_PROG в регистре I2C\_STA. Когда передача завершена, этот флаг снимается и устанавливается флаг INT. Если при этом установлен бит разрешения INT\_EN, то генерируется прерывание контроллеру прерываний. Регистр I2C\_RXD содержит корректные принятые данные после установки флага INT. Пользователь может начать новый цикл чтения или записи только тогда, когда флаг TR\_PROG сброшен.

### **Сигнал STOP**

Мастер может завершить соединение путем создания сигнала STOP. Сигнал STOP или P бит определяется переходом линии SDA из низкого состояния в высокое, когда SCL находится в высоком состоянии.

### **Описание регистров контроллера I2C**

Таблица 122

Базовый Адрес	Название	Описание
0x4003_0000	I2C	Контроллер I2C
<b>Смещение</b>		
0x00	I2C->PRL	Младшая часть пред делителя частоты
0x04	I2C->PRH	Старшая часть пред делителя частоты
0x08	I2C->CTR	Управление контроллером I2C
0x0C	I2C->RXD	Принятые данные по I2C
0x10	I2C->STA	Статус I2C
0x14	I2C->TXD	Передаваемые данные по I2C
0x18	I2C->CMD	Управление I2C

### I2C->PRL

Регистр PRL

Номер	7	6	5	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0

PR[7:0]

**Таблица 123** Описание битов регистра PRL

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..8	-	Зарезервировано
7...0	PR[7:0]	Младшая часть пред. делителя

### I2C->PRH

Регистр PRH

Номер	7	6	5	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0

PR[15:8]

**Таблица 124** Описание битов регистра PRH

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..8	-	Зарезервировано
7...0	PR[15:8]	Старшая часть пред. делителя

### I2C->CTR

Регистр CTR

Номер	7	6	5	4	3	2	1	0
-------	---	---	---	---	---	---	---	---

Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0
	EN_I2C	EN_INT	S_I2C	-	-	-	-	-

**Таблица 125** Описание битов регистра CTR

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..8	-	Зарезервировано
7	EN_I2C	Разрешение работы контроллера I2C 0 – выключен 1 – включен
6	EN_INT	Разрешение прерывания от I2C 0 – запрещено 1 – разрешено
5	S_I2C	Скорость работы I2C 0 – до 400 КГц 1 – до 1 МГц
4...0	-	Зарезервировано

### I2C->RXD

Регистр RXD

Номер	7	6	5	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0
	RXD[7:0]							

**Таблица 126** Описание битов регистра RXD

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..8	-	Зарезервировано
7...0	RXD[7:0]	Последний полученный по I2C байт

### I2C->STA

Регистр STA

Номер	7	6	5	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0
	Rx ACK	BUSY	LOST ARB	-	-	-	TR PROG	INT

**Таблица 127** Описание битов регистра STA

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
7...0	STA[7:0]	Статусные биты регистра STA

		значений.
31..8	-	Зарезервировано
7	Rx ACK	Полученный от ведомого ACK 0 – ACK получен 1 – получен NACK
6	BUSY	Состояние шины I2C 1 – после получения состояния Start Bit 0 – после получения Stop bit
5	LOST ARB	Потеря арбитража 1 – потерян арбитраж 0 – нет потери арбитража Этот бит выставляется если: - Получен Stop bit, но он не был инициализирован этим контроллером - Если контроллер пытается выставить SDA в высокий уровень, но SDA остается в низком
4..2	-	Зарезервировано
1	TR PROG	Процесс передачи 1 – передаются данные 0 – передача завершена
0	INT	Флаг прерывания, выставляется всегда. Прерывание для процессора выдается если есть флаг EN_INT. 1 – есть прерывание 0 – нет прерывания Флаг выставляется если: - передача байта завершена - был потерян арбитраж

### I2C->TXD

Регистр TXD

Номер	7	6	5	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0
	TXD[7:0]							

**Таблица 128** Описание битов регистра TXD

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..8	-	Зарезервировано
7...0	TXD[7:0]	Байт для отправки по I2C При передаче адреса нулевой бит определяет режим передачи 0 – запись в ведомое устройство 1 – чтение из ведомого устройства

**I2C->CMD**

Регистр CMD

Номер	7	6	5	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0
	START	STOP	RD	WR	ACK	-	-	CLR INT

**Таблица 129** Описание битов регистра CMD

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..8	-	Зарезервировано
7	START	Отправить START bit Инициализируется записью 1 После завершения отправки автоматически не сбрасывается, очищается записью нуля
6	STOP	Отправить STOP bit Инициализируется записью 1 После завершения отправки автоматически не сбрасывается, очищается записью нуля
5	RD	Чтение из ведомого 0 – нет действия 1 – начать чтение
4	WR	Запись в ведомого 0 – нет действия 1 – начать запись
3	ACK	Отправить ACK при чтении 0 – отправить ACK 1 – отправить NACK
2...1	-	Зарезервировано
0	CLR INT	Очистить прерывание INT Запись 1 очищает прерывание

### **Контроллер SSP**

Модуль порта синхронной последовательной связи (SSP – Synchronous Serial Port) выполняет функции интерфейса последовательной синхронной связи в режиме ведущего и ведомого устройства и обеспечивает обмен данными с подключенным ведомым или ведущим периферийным устройством в соответствии с одним из протоколов:

- интерфейс SPI фирмы Motorola;
- интерфейс SSI фирмы Texas Instruments;
- интерфейс Microwire фирмы National Semiconductor.

Как в ведущем, так и в ведомом режиме работы модуль SSP обеспечивает:

- преобразование данных, размещенных во внутреннем буфере FIFO передатчика (восемь 16-разрядных ячеек данных) из параллельного в последовательный формат;
- преобразование данных из последовательного в параллельный формат и их запись в аналогичный буфер FIFO приемника (восемь 16-разрядных ячеек данных).

Модуль формирует сигналы прерываний по следующим событиям:

- необходимость обслуживания буферов FIFO приемника и передатчика;
- переполнение буфера FIFO приемника;
- наличие данных в буфере FIFO приемника по истечении времени таймаута.

Основные сведения о модуле представлены в следующих разделах:

- характеристики интерфейса SPI;
- характеристики интерфейса Microwire;
- характеристики интерфейса SSI.

### **Основные характеристики модуля SSP**

- может функционировать как в ведущем, так и в ведомом режиме;
- программное управление скоростью обмена;
- состоит из независимых буферов приема и передачи (8 ячеек по 16 бит) с организацией доступа типа FIFO (First In First Out – первый вошел, первый вышел);
- программный выбор одного из интерфейсов обмена: SPI, Microwire, SSI;
- программируемая длительность информационного кадра от 4 до 16 бит;
- независимое маскирование прерываний от буфера FIFO передатчика, буфера FIFO приемника, а также по переполнению буфера приемника;
- доступна возможность тестирования по шлейфу, соединяющему вход с выходом;

Структурная схема модуля представлена на Рис. 63.

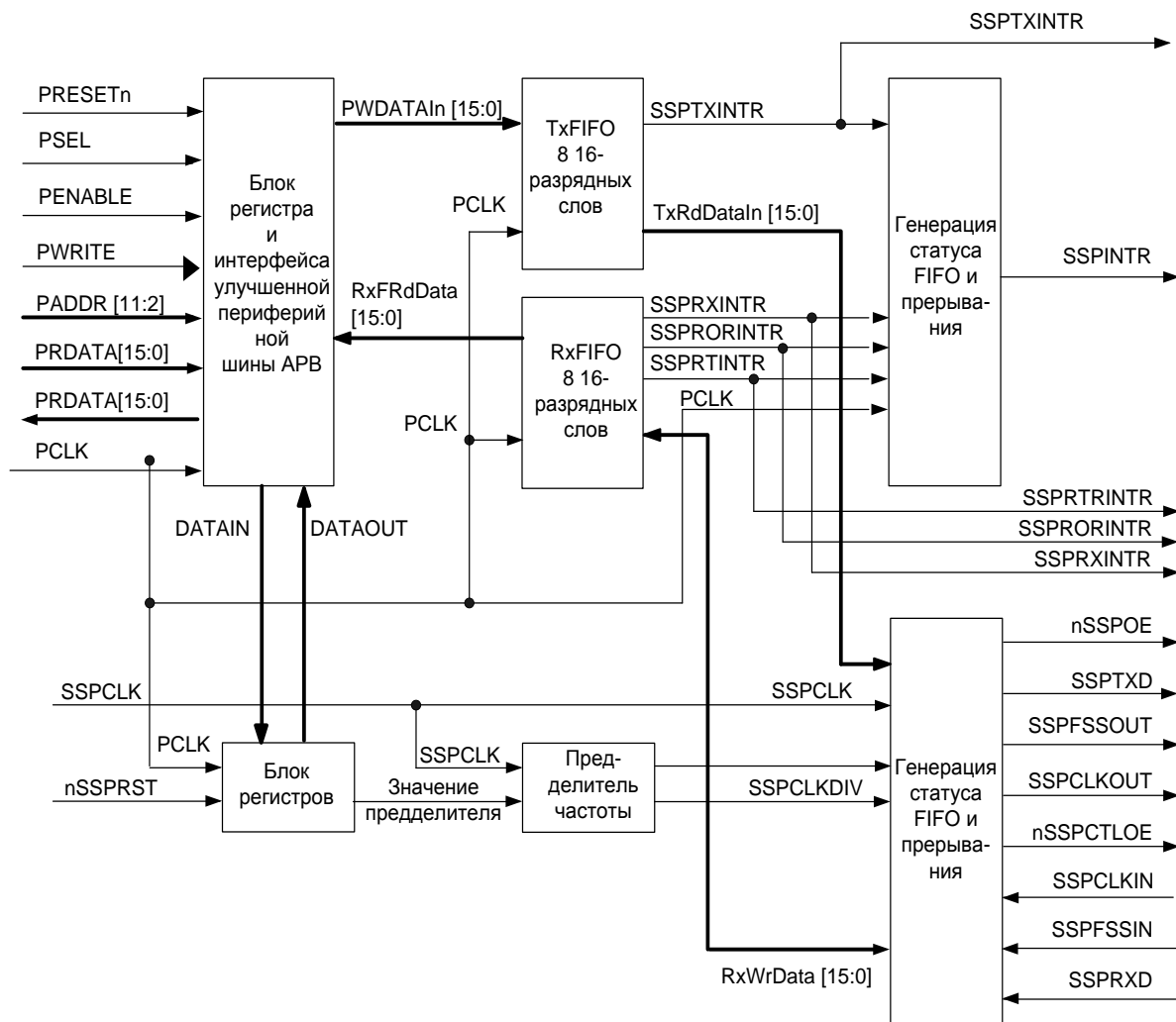


Рис. 63 Структурная схема модуля SSP

**Программируемые параметры**

Следующие ключевые параметры могут быть заданы программно:

- режим функционирования периферийного устройства – ведущее или ведомое;
- разрешение или запрещение функционирования;
- формат информационного кадра;
- скорость передачи данных;
- фаза и полярность тактового сигнала;
- размер блока данных – от 4 до 16 бит;
- маскирование прерываний.

### **Характеристики интерфейса SPI**

Последовательный синхронный интерфейс SPI фирмы Motorola обеспечивает:

- полнодуплексный обмен данными по четырехпроводной линии;
- программное задание фазы и полярности тактового сигнала.

### **Характеристики интерфейса Microwire**

Интерфейс Microwire фирмы National Semiconductor обеспечивает: полудуплексный обмен данными с использованием восьмибитных управляющих последовательностей.

### **Характеристики интерфейса SSI**

Интерфейс SSI фирмы Texas Instruments обеспечивает:

- полнодуплексный обмен данными по четырехпроводной линии;
- возможность перевода линии передачи данных в третье (высокоимпедансное) состояние.

### **Общий обзор модуля SSP**

Модуль SSP представляет собой интерфейс синхронного последовательного обмена данными, способный функционировать в качестве ведущего или ведомого устройства и поддерживающий протоколы передачи данных SPI фирмы Motorola, Microwire фирмы National Semiconductor, а также SSI фирмы Texas Instruments.

Модуль выполняет следующие функции:

- преобразование данных, полученных от периферийного устройства, из последовательной в параллельную форму;
- преобразование данных, передаваемых на периферийное устройство, из параллельной и последовательную форму;
- центральный процессор читает и записывает данные, а также управляющую информацию и информацию о состоянии;
- прием и передача данных буферизуются с помощью буферов FIFO, обеспечивающих хранение до восьми слов данных шириной 16 бит независимо для режимов приема и передачи.

Последовательные данные передаются по линии SSP\_TXD (см. раздел «Описание регистров контроллера SSP») и принимаются с линии SSP\_RXD.

Модуль SSP содержит программируемые делители частоты, формирующие тактовый сигнал обмена данными SSP\_CLK из сигнала, поступающего на линию SSPCLK. Скорость передачи данных может достигать более 2 МГц, в зависимости от частоты SSPCLK и характеристик подключенного периферийного устройства.

Режим обмена данными, формат информационного кадра и количество бит данных задаются программно с помощью регистров управления CR0 и CR1.

Модуль формирует четыре независимо маскируемых прерывания:

SSPTXINTR – запрос на обслуживание буфера передатчика;

SSPRXINTR – запрос на обслуживание буфера приемника;

SSPRORINTR – переполнение приемного буфера FIFO;



SSPRTINTR – таймаут ожидания чтения данных из приемного FIFO.

Кроме того, формируется общий сигнал прерывания SSPINTR, возникающий в случае активности одного из вышеуказанных независимых немаскированных прерываний, который идет на контроллер NVIC.

В зависимости от режима работы модуля сигнал SSPFSSOUT используется либо для кадровой синхронизации (интерфейс SSI, активное состояние – высокий уровень), либо для выбора ведомого режима (интерфейсы SPI и Microwire, активное состояние – низкий уровень).

### **Блок формирования тактового сигнала**

В режиме ведущего устройства модуль формирует тактовый сигнал обмена данными SSP\_CLK с помощью внутреннего делителя частоты, состоящего из двух последовательно соединенных счетчиков без цепи сброса.

Путем записи значения в регистр SSPCPSR можно задать коэффициент предварительного деления частоты в диапазоне от 2 до 254 с шагом 2. Так как младший значащий разряд коэффициента деления не используется, то исключается возможность деления частоты на нечетный коэффициент деления. Это, в свою очередь, гарантирует формирование тактового сигнала симметричной формы (с одинаковой длительностью полупериодов высокого и низкого уровней). Сформированный описанным образом сигнал далее поступает на второй делитель частоты, с выхода которого и снимается тактовый сигнал обмена данными SSP\_CLK.

Коэффициент деления второго делителя задается программно в диапазоне от 1 до 256, путем записи соответствующего значения в регистр управления SSPCR0.

### **Буфер FIFO передатчика**

Буфер передатчика имеет ширину 16 бит, глубину 8 слов, схему организации доступа типа FIFO - «первый вошел, первый вышел». Данные от центрального процессора сохраняются в буфере до тех пор, пока не будут считаны блоком передачи данных.

### **Буфер FIFO приемника**

Буфер приемника имеет ширину 16 бит, глубину 8 слов, схему организации доступа типа FIFO - «первый вошел, первый вышел». Принятые от периферийного устройства данные сохраняются в этом буфере блоком приема данных до тех пор, пока не будут считаны центральным процессором.

### **Блок приема и передачи данных**

В режиме ведущего устройства модуль формирует тактовый сигнал обмена данными SSP\_CLK для подключенных ведомых устройств. Как было описано ранее, данный сигнал формируется путем деления частоты сигнала SSPCLK.

Блок передатчика последовательно считывает данные из буфера FIFO передатчика и производит их преобразование из параллельной формы в последовательную. Далее поток последовательных данных и элементов кадровой

синхронизации, тактированный сигналом SSP\_CLK, передаётся по линии SSP\_TXD к подключенным ведомым устройствам.

Блок приемника выполняет преобразование данных, поступающих синхронно с линии SSP\_RXD, из последовательной в параллельную форму, после чего загружает их в буфер FIFO приемника, откуда они могут быть считаны процессором.

В режиме ведомого устройства тактовый сигнал обмена данными формируется одним из подключенных к модулю периферийных устройств и поступает по линии SSP\_CLK.

При этом блок передатчика, тактируемый этим внешним сигналом, считывает данные из буфера FIFO, преобразует их из параллельной формы в последовательную, после чего выдает поток последовательных данных и элементов кадровой синхронизации в линию SSP\_TXD.

Аналогично, блок приемника выполняет преобразование данных, поступающих с линии SSP\_RXD синхронно с сигналом SSP\_CLK, из последовательной в параллельную форму, после чего загружает их в буфер FIFO приемника, откуда они могут быть считаны процессором.

### **Блок формирования прерываний**

Модуль SSP генерирует независимые маскируемые прерывания с активным высоким уровнем. Кроме того, формируется комбинированное прерывание путем объединения указанных независимых прерываний по схеме ИЛИ.

Комбинированный сигнал прерывания подается на контроллер прерываний NVIC, при этом появляется дополнительная возможность маскирования устройства в целом, что облегчает построение модульных драйверов устройств.

### **Интерфейс прямого доступа к памяти**

Модуль обеспечивает интерфейс с контроллером DMA согласно схеме взаимодействия приемопередатчика и контроллера DMA.

### **Конфигурирование приемопередатчика**

После сброса работа блоков приемопередатчика запрещается до выполнения процедуры задания конфигурации.

Для этого необходимо выбрать ведущий или ведомый режим работы устройства, а также используемый протокол передачи данных (SPI фирмы Motorola, SSI фирмы Texas Instruments, либо Microwave фирмы National Semiconductor), после чего записать необходимую информацию в регистры управления CR0 и CR1.

Кроме того, для установки требуемой скорости передачи данных необходимо выбрать параметры блока формирования тактового сигнала с учетом значения частоты сигнала SSPCLK и записать соответствующую информацию в регистр PSR.

### **Разрешение работы приемопередатчика**

Разрешение осуществляется путем установки бита SSE регистра управления CR1. Буфер FIFO передатчика может быть либо проинициализирован путем записи в него до восьми 16-разрядных слов заблаговременно перед

установкой этого бита, либо может заполняться передаваемыми данными в процедуре обслуживания прерывания.

После разрешения работы модуля приемопередатчик начинает обмен данными по линиям SSP\_TXD и SSP\_RXD.

### Соотношения между тактовыми сигналами

В модуле имеется ограничение на соотношение между частотами тактовых сигналов CPU\_CLK и SSPCLK. Частота SSPCLK должна меньше или равна частоте CPU\_CLK. Выполнение этого требования гарантирует синхронизацию сигналов управления, передаваемых из зоны действия тактового сигнала SSPCLK в зону действия сигнала CPU\_CLK в течение времени, меньшего продолжительности передачи одного информационного кадра:  
 $F_{SSPCLK} \leq F_{PCLK}$ .

В режиме ведомого устройства сигнал SSP\_CLK от ведущего внешнего устройства поступает на схемы синхронизации, задержки и обнаружения фронта. Для того, чтобы обнаружить фронт сигнала SSP\_CLK, необходимо три такта сигнала SSP\_CLK. Сигнал SSP\_TXD имеет меньшее время установки по отношению к заднему фронту SSP\_CLK, по которому и происходит считывание данных из линии. Время установки и удержания сигнала SSP\_RXD по отношению к сигналу SSP\_CLK должно выбираться с запасом, гарантирующим правильное считывание данных. Для обеспечения корректной работы устройства необходимо, чтобы частота SSPCLK была как минимум в 12 раз больше, чем максимальная предполагаемая частота сигнала SSP\_CLK.

Выбор частоты тактового сигнала SSPCLK должен обеспечивать поддержку требуемого диапазона скоростей обмена данными. Отношение минимальной частоты сигнала SSPCLK к максимальной частоте сигнала SSP\_CLK в режиме ведомого устройства равно 12, в режиме ведущего – двум.

Так, в режиме ведущего устройства для обеспечения максимальной скорости обмена 1.8432 Мбит/с частота сигнала SSPCLK должна составлять не менее 3.6864 МГц. В этом случае в регистр CPSR должно быть записано значение 2, а поле SCR[7:0] регистра CR0 должно быть установлено в 0.

В режиме ведомого устройства для обеспечения той же информационной скорости необходимо использовать тактовый сигнал SSPCLK с частотой не менее 22.12 МГц. При этом в регистр CPSR должно быть записано значение 12, а поле SCR[7:0] регистра CR0 должно быть установлено в 0.

Соотношение между максимальной частотой сигнала SSPCLK и минимальной частотой SSPCLKOUT составляет  $254 * 256$ .

Минимальная допустимая частота сигнала SSPCLK определяется следующей системой соотношений, которые должны выполняться одновременно:  
 $F_{SSPCLK}(\min) \Rightarrow 2 \times F_{SSPCLKOUT}(\max)$  [в режиме Master]  
 $F_{SSPCLK}(\min) \Rightarrow 12 \times F_{SSPCLKIN}(\max)$  [в режиме Slave].

Аналогично, максимальная допустимая частота сигнала SSPCLK определяется следующей системой соотношений, которые должны выполняться одновременно:

$F_{SSPCLK}(\max) \leq 254 \times 256 \times F_{SSPCLKOUT}(\min)$  [в режиме Master]

$F_{SSPCLK}(\max) \leq 254 \times 256 \times F_{SSPCLKIN}(\min)$  [в режиме Slave].

### Программирование регистра управления SSPCR0

Регистр CR0 предназначен для:

- установки скорости информационного обмена;
- выбора одного из трех протоколов обмена данными;
- выбора размера слова данных.

Скорость информационного обмена зависит от частоты внешнего тактового сигнала SSPCLK и коэффициента деления блока формирования тактового сигнала. Последний задается совместно значением поля SCR (Serial Clock Rate – скорость информационного обмена) регистра SSPCR0 и значением поля CPSDVSR (clock prescale divisor value – коэффициент деления тактового сигнала) регистра SSPCPSR.

Формат информационного кадра задается путем установки значения поля FRF, а размер слова данных – путем установки значения поля DSS регистра SSPCR0. Для протокола SPI фирмы Motorola также задаются полярность и фаза сигнала (биты SPH и SPO).

### Программирование регистра управления SSPCR1

Регистр SSPCR1 предназначен для:

- выбора ведущего или ведомого режима функционирования приемопередатчика;
- включения режима проверки канала по шлейфу;
- разрешения или запрещения работы модуля.

Выбор ведущего режима осуществляется путем записи 0 в поле MS регистра SSPCR1 (это значение устанавливается после сброса автоматически).

Запись 1 в поле MS переводит приемопередатчик в режим ведомого устройства. В этом режиме разрешение или запрещение формирования сигнала передатчика SSP\_TXD осуществляется путем установки бита SOD (slave mode SSP\_TXD output disable – запрет линии SSP\_TXD для ведомого режима) регистра CR1. Указанная функция полезна при подключении к одной линии нескольких подчиненных устройств.

Для того, чтобы разрешить функционирование приемопередатчика, необходимо установить в 1 бит SSE (Synchronous Serial Port Enable – разрешение последовательного синхронного порта).

### Формирование тактового сигнала обмена данными

Тактовый сигнал обмена данными формируется путем деления частоты тактового сигнала SSPCLK. На первом этапе формирования частота этого сигнала делится на четный коэффициент CPSDVSR, лежащий в диапазоне от 2 до 254, доступный для программирования через регистр CPSR. Сформированный сигнал далее поступает на делитель частоты с коэффициентом  $(1 + SCR)$  от 1 до 256, где значение SCR доступно для программирования через CR0.

Частота выходного тактового сигнала обмена данными SSP\_CLK определяется следующим соотношением:

$$F_{SSPCLKOUT} = F_{SSPCLK} / (CPSDVR * (1+SCR))$$

Например, в случае, если частота сигнала SSPCLK составляет 3.6864 МГц, а значение CPSDVSR = 2, частота сигнала SSP\_CLK лежит в интервале от 7.2 кГц до 1.8432 МГц.

## **Формат информационного кадра**

Каждый информационный кадр содержит в зависимости от запрограммированного значения от 4 до 16 бит данных. Передача данных начинается со старшего значащего разряда. Возможно выбрать три базовых структуры построения кадра:

- SSI фирмы Texas Instruments;
- SPI фирмы Motorola;
- Microwire фирмы National Semiconductor.

Во всех трех режимах построения кадра тактовый сигнал SSP\_CLK формируется только тогда, когда приемопередатчик готов к обмену данными. Перевод сигнала SSP\_CLK в неактивное состояние используется как признак таймаута приемника, то есть наличия в буфере приемника необработанных данных по истечении заданного интервала времени.

В режимах SPI и Microwire выходной сигнал кадровой синхронизации передатчика SSP\_FSS имеет активный низкий уровень и поддерживается в низком уровне в течение всего периода передачи информационного кадра.

В режиме построения кадра SSI фирмы Texas Instruments перед началом каждого информационного кадра на выходе SSP\_FSS формируется импульс с длительностью, равной одному тактовому интервалу обмена данными. В этом режиме приемопередатчик SSP, равно как и ведомые периферийные устройства, передает данные в линию по переднему фронту сигнала SSP\_CLK, а считывает данные из линии по заднему фронту этого сигнала.

В отличие от полнодуплексных режимов передачи данных SSI и SPI, режим Microwire фирмы National Semiconductor использует специальный способ обмена данными между ведущим и ведомым устройством, функционирующий в режиме полудуплекса. В указанном режиме на внешнее ведомое устройство перед началом передачи информационного кадра посылается специальная восьмибитная управляющая последовательность. В течение всего времени передачи этой последовательности приемник не обрабатывает каких-либо входных данных. После того как сигнал передан и декодирован ведомым устройством, оно выдерживает паузу в один тактовый интервал после передачи последнего бита управляющей последовательности, после чего передает в адрес ведущего устройства запрошенные данные. Длительность блока данных от ведомого устройства может составлять от 4 до 16 бит, таким образом общая длительность информационного кадра составляет от 13 до 25 бит.

## Формат синхронного обмена SSI фирмы Texas Instruments

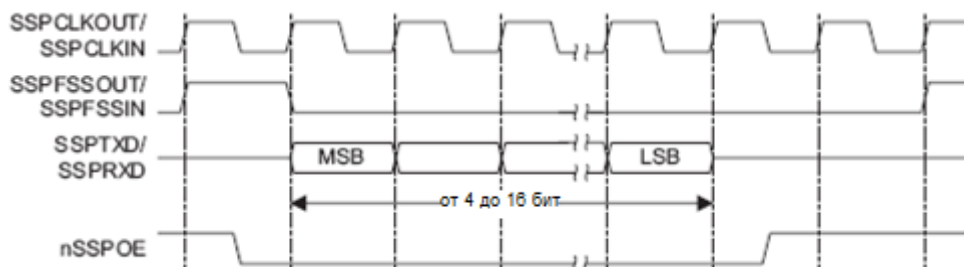


Рис. 64 Формат синхронного обмена протокола SSI фирмы Texas Instruments (единичный обмен)

В данном режиме при неактивном приемопередатчике SSP сигналы SSP\_CLK и SSP\_FSS переводятся в низкий логический уровень, а линия передачи данных SSP\_TXD поддерживается в третьем состоянии.

После появления хотя бы одного элемента в буфере FIFO передатчика сигнал SSPFSSOUT переводится в высокий логический уровень на время, соответствующее одному периоду сигнала SSP\_CLK. Значение из буфера FIFO при этом переносится в сдвиговый регистр блока передатчика. По следующему переднему фронту сигнала SSP\_CLK старший значащий разряд информационного кадра (4 – 16 бит данных) выдается на выход линии SSP\_TXD и т.д.

В режиме приема данных как модуль SSP, так и ведомое внешнее устройство последовательно загружают биты данных в сдвиговый регистр по заднему фронту сигнала SSP\_CLK. Принятые данные переносятся из сдвигового регистра в буфер FIFO после загрузки в него младшего значащего бита данных по очередному переднему фронту сигнала SSP\_CLK.

Временные диаграммы последовательного синхронного обмена по протоколу SSI фирмы Texas Instruments представлены на Рис. 64 (передача единичного информационно кадра) и Рис. 65 (передача последовательности кадров).

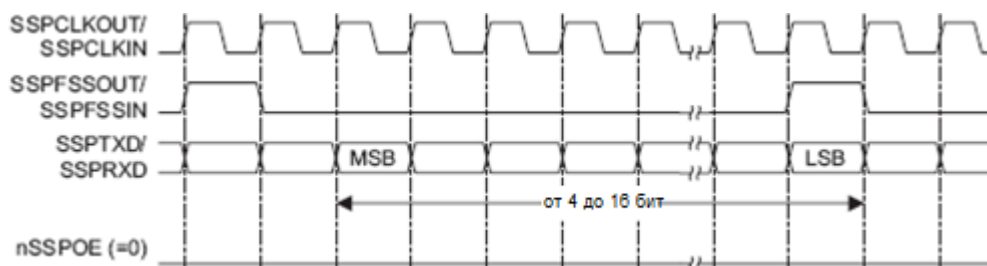


Рис. 65 Формат синхронного обмена протокола SSI фирмы Texas Instruments (непрерывный обмен)

## Формат синхронного обмена SPI фирмы Motorola

Интерфейс SPI фирмы Motorola осуществляется по четырем сигнальным линиям, при этом сигнал SSP\_FSS выполняет функцию выбора ведомого устройства. Главной особенностью протокола SPI является возможность выбора состояния и фазы сигнала SSP\_CLK в режиме ожидания (неактивном приемопередатчике) путем задания значений бит SPO и SPH регистра управления SSPSCR0.

### Выбор полярности тактового сигнала – бит SPO

Если бит SPO равен 0, то в режиме ожидания линия SSP\_CLK переводится в низкий логический уровень. В противном случае при отсутствии обмена данными линия SSP\_CLK переводится в высокий логический уровень.

### Выбор фазы тактового сигнала – бит SPH

Значение бита SPH определяет фронт тактового сигнала, по которому осуществляется выборка данных и изменение состояния на выходе линии.

В случае, если бит SPH установлен в 0, регистрация данных приемником осуществляется после первого обнаружения фронта тактового сигнала, в противном случае – после второго.

### Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=0

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=0, SPH=0 показаны на Рис. 66 (одиночный обмен) и Рис. 67 (непрерывный обмен).

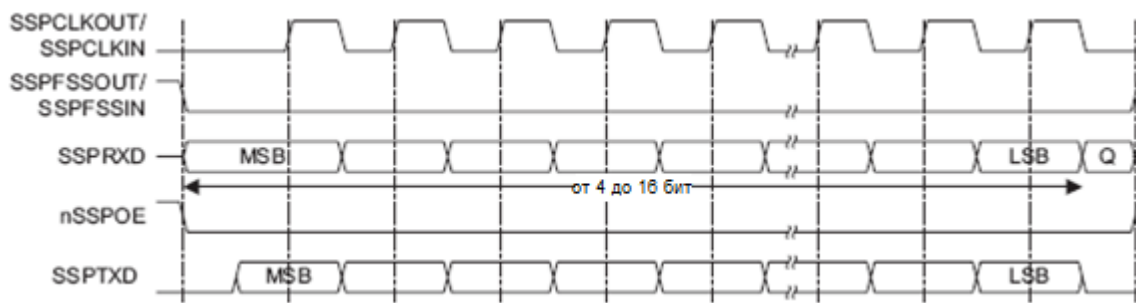


Рис. 66 Формат синхронного обмена протокола SPI фирмы Motorola, SPO=0,SPH=0 (одиночный обмен)

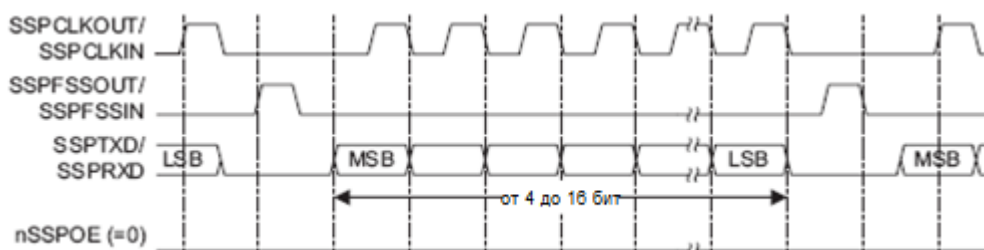


Рис. 67 Формат синхронного обмена протокола SPI фирмы Motorola, SPO=0, SPH=0 (непрерывный обмен)

#### Примечание

На Рис. 66 буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSP\_CLK имеет низкий логический уровень;
- сигнал SSP\_FSS имеет высокий логический уровень;
- сигнал SSP\_TXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSP\_FSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от

ведомого устройства на входную линию SSP\_RXD ведущего. Контакт передатчика SSPTXD переходит из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSP\_CLK на линии SSP\_TXD формируется значение первого бита передаваемых данных. К этому моменту должны быть сформированы данные на линиях обмена как ведущего, так и ведомого устройства. По истечении следующего полутакта сигнал SSP\_CLK переводится в высокий логический уровень.

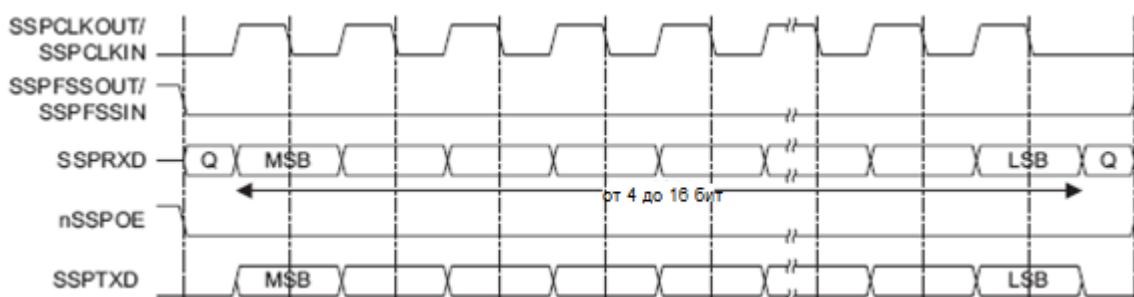
Далее данные регистрируются по переднему фронту и выдаются в линию по заднему фронту сигнала SSP\_CLK.

В случае передачи одного слова данных после приема его последнего бита линия SSP\_FSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSP\_CLK.

В режиме непрерывной передачи данных на линии SSP\_FSS должны формироваться импульсы высокого логического уровня между передачами каждого из слов данных. Это связано с тем, что в режиме SPH=0 линия выбора ведомого устройства в низком уровне блокирует запись в сдвиговый регистр. Поэтому ведущее устройство должно переводить линию SSP\_FSS в высокий уровень по окончании передачи каждого кадра, разрешая таким образом запись новых данных. По окончании приема последнего бита блока данных линия SSP\_FSS переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSP\_CLK.

### **Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=1**

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=0, SPH=1 показаны на Рис. 68 (одиночный и непрерывный обмен).



**Рис. 68** Формат синхронного обмена протокола SPI фирмы Motorola, SPO=0, SPH=1

#### *Примечание*

На Рис. 68 буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSP\_CLK имеет низкий логический уровень;
- сигнал SSP\_FSS имеет высокий логический уровень;
- сигнал SSP\_TXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSP\_FSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от



ведомого устройства на входную линию SSP\_RXD ведущего. Выходной контакт передатчика SSPTXD переходит из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSP\_CLK на линиях обмена как ведущего, так и ведомого устройств будут сформированы значения первых бит передаваемых данных. В это же время включается линия SSP\_CLK и на ней формируется передний фронт сигнала.

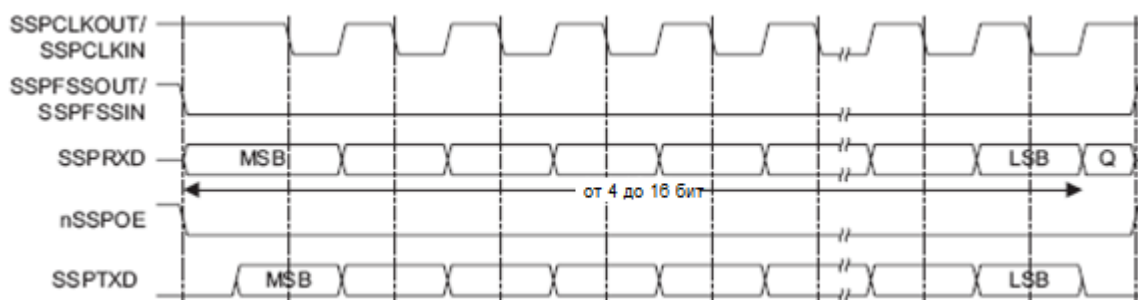
Далее данные регистрируются по заднему фронту и выдаются в линию по переднему фронту сигнала SSP\_CLK.

В случае передачи одного слова данных после приема его последнего бита линия SSP\_FSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSP\_CLK.

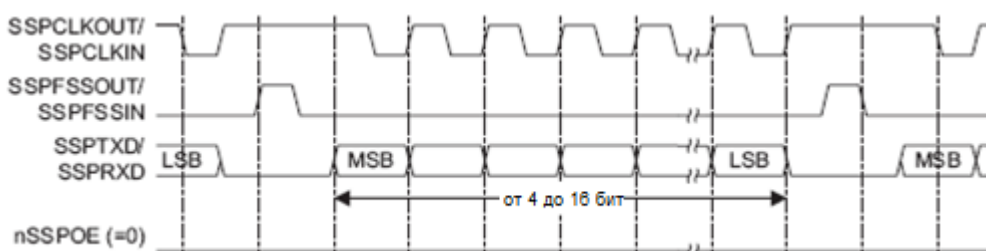
В режиме непрерывной передачи данных линия SSP\_FSS постоянно находится в низком логическом уровне, и переводится в высокий уровень по окончании приема последнего бита блока данных, как и в режиме передачи одного слова.

### **Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=0**

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=1, SPH=0 показаны на Рис. 69 (одиночный обмен) и Рис. 70 (непрерывный обмен).



**Рис. 69** Формат синхронного обмена протокола SPI фирмы Motorola, SPO=1, SPH=0 (одиночный обмен)



**Рис. 70** Формат синхронного обмена протокола SPI фирмы Motorola, SPO=1, SPH=0 (непрерывный обмен)

*Примечание*

На Рис. 69 буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSP\_CLK имеет высокий логический уровень;
- сигнал SSP\_FSS имеет высокий логический уровень;

- сигнал SSP\_TXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSP\_FSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSP\_RXD ведущего. Выходной контакт передатчика SSPTXD переходит из высокоимпедансного в активное состояние. По истечении полутакта сигнала SSP\_CLK, на линии SSP\_TXD формируется значение первого бита передаваемых данных. К этому моменту должны быть сформированы данные на линиях обмена как ведущего, так и ведомого устройства. По истечении следующего полутакта сигнал SSP\_CLK переводится в низкий логический уровень.

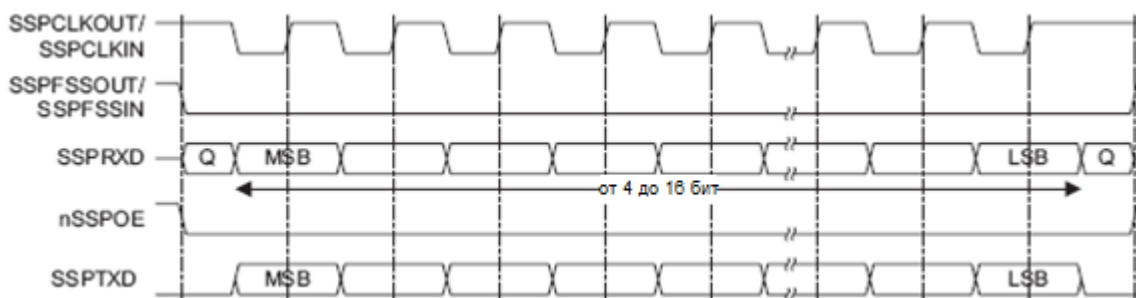
Далее данные регистрируются по заднему фронту и выдаются в линию по переднему фронту сигнала SSP\_CLK.

В случае передачи одного слова данных после приема его последнего бита линия SSP\_FSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSP\_CLK.

В режиме непрерывной передачи данных на линии SSP\_FSS должны формироваться импульсы высокого логического уровня между передачами каждого из слов данных. Это связано с тем, что в режиме SPH=0 линия выбора ведомого устройства в низком уровне блокирует запись в сдвиговый регистр. Поэтому ведущее устройство должно переводить линию SSP\_FSS в высокий уровень по окончании передачи каждого кадра, разрешая таким образом запись новых данных. По окончании приема последнего бита блока данных линия SSP\_FSS переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSP\_CLK.

### Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=1

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=1, SPH=1 показаны на Рис. 71 (одиночный и непрерывный обмен).



**Рис. 71** Формат синхронного обмена протокола SPI фирмы Motorola, SPO=1, SPH=1

*Примечание*

На Рис. 71 буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSP\_CLK имеет высокий логический уровень;
- сигнал SSP\_FSS имеет высокий логический уровень;
- сигнал SSP\_TXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSP\_FSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSP\_RXD ведущего. Выходной контакт передатчика SSP\_TXD переходит из высокоимпедансного в активное состояние. По истечении полутакта сигнала SSP\_CLK на линиях обмена как ведущего, так и ведомого устройств сформированы значения первых бит передаваемых данных. В это же время включается линия SSP\_CLK и на ней формируется передний фронт сигнала.

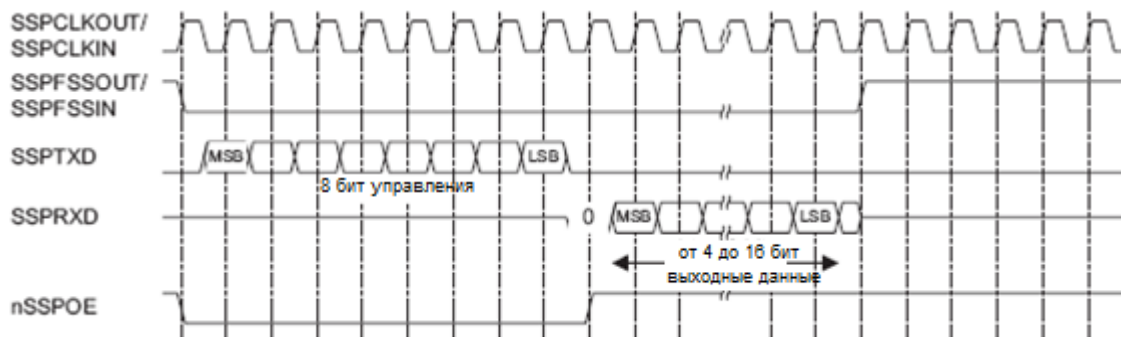
Далее данные регистрируются по переднему фронту и выдаются в линию по заднему фронту сигнала SSP\_CLK.

В случае передачи одного слова данных после приема его последнего бита линия SSP\_FSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSP\_CLK.

В режиме непрерывной передачи данных линия SSP\_FSS постоянно находится в низком логическом уровне и переводится в высокий уровень по окончании приема последнего бита блока данных, как и в режиме передачи одного слова.

## **Формат синхронного обмена Microwire фирмы National Semiconductor**

Временные диаграммы последовательного синхронного обмена в режиме Microwire показаны на Рис. 72 (одиночный обмен) и Рис. 73 (непрерывный обмен).



**Рис. 72** Формат синхронного обмена протокола Microwire фирмы National Semiconductor (одиночный обмен)

Протокол передачи данных Microwire во многом схож с протоколом SPI, за исключением того, что обмен в нем осуществляется в полудуплексном режиме, с использованием служебных последовательностей. Каждая информационный обмен начинается с передачи ведущим устройством специальной восьмьбитной управляющей последовательности. В течение всего времени ее передачи приемник не обрабатывает каких-либо входных данных. После того, как сигнал передан и декодирован ведомым устройством, оно выдерживает паузу в один тактовый интервал после передачи последнего бита управляющей последовательности, после чего передает в адрес ведущего устройства запрошенные данные. Длительность блока данных от ведомого устройства может составлять от 4 до 16 бит, таким образом, общая длительность информационного кадра составляет от 13 до 25 бит.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSP\_CLK имеет низкий логический уровень;
- сигнал SSP\_FSS имеет высокий логический уровень;
- сигнал SSP\_TXD переводится в высокоимпедансное состояние.

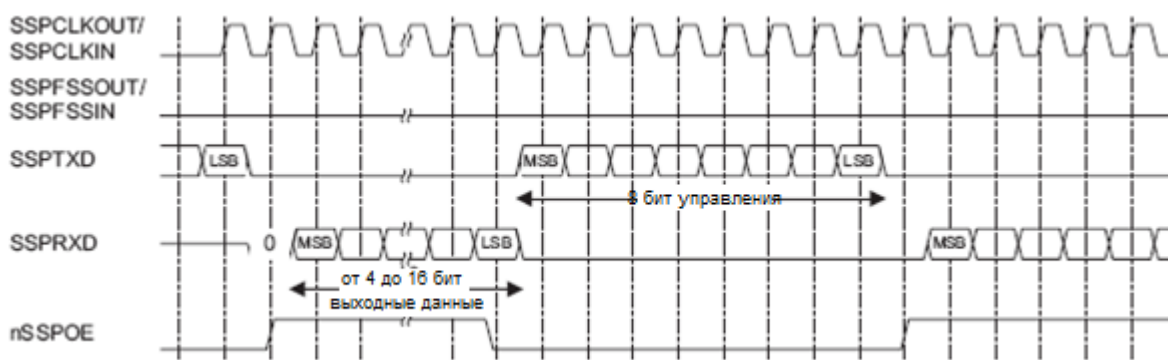
Переход в режим информационного обмена происходит после записи управляющего байта в буфер FIFO передатчика. По заднему фронту сигнала SSP\_FSS данные из буфера переносятся в регистр сдвига блока передатчика, откуда, начиная со старшего значащего разряда, последовательно выдаются в линию SSP\_TXD. Линия SSP\_FSS остается в низком логическом уровне в течение всей передачи кадра. Линия SSP\_RXD при этом находится в высокоимпедансном состоянии.

Внешнее ведомое устройство осуществляет прием бит данных по переднему фронту сигнала SSP\_CLK. По окончании приема последнего бита управляющей последовательности она декодируется в течение одного тактового интервала, после чего ведомое устройство передает запрошенные данные в адрес модуля SSP. Биты данных выдаются в линию SSP\_RXD по заднему фронту сигнала SSP\_CLK. Ведущее устройство, в свою очередь, регистрирует их по переднему фронту этого тактового сигнала. В случае одиночного информационного обмена по окончании приема последнего бита слова данных сигнал SSP\_FSS переводится в высокий уровень на время, соответствующее одному тактовому интервалу, что служит командой для переноса принятого слова данных из регистра сдвига в буфер FIFO приемника.

*Примечание:*

Внешнее устройство может перевести линию приемника в третье состояние по заднему фронту сигнала SSP\_CLK после приема последнего бита слова данных, либо после перевода линии SSP\_FSS в высокий логический уровень.

Непрерывный обмен данными начинается и заканчивается так же, как и одиночный обмен. Однако линия SSP\_FSS удерживается в низком логическом уровне в течение всего сеанса передачи данных. Управляющий байт следующего информационного кадра передается сразу же после приема младшего значащего разряда текущего кадра. Данные из сдвигового регистра передаются в буфер приемника после регистрации младшего разряда очередного слова по заднему фронту сигнала SSP\_CLK.

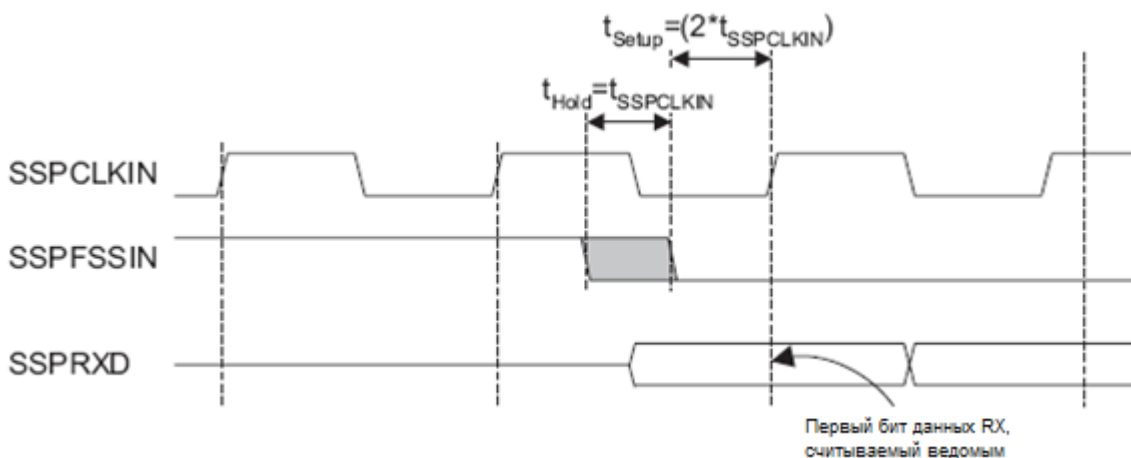


**Рис. 73** Формат синхронного обмена протокола Microwire фирмы National Semiconductor (непрерывный обмен)

Требования к временным параметрам сигнала SSP\_FSS относительно тактового сигнала SSP\_CLK в режиме Microwire.

Модуль SSP, работающий в режиме Microwire как ведомое устройство, регистрирует данные по переднему фронту сигнала SSP\_CLK после установки сигнала SSP\_FSS в низкий логический уровень. Ведущие устройства, формирующие сигнал SSP\_CLK, должны гарантировать достаточное время установки и удержания сигнала SSP\_FSS по отношению к переднему фронту сигнала SSP\_CLK.

Данные требования иллюстрирует Рис. 74. По отношению к переднему фронту сигнала SSP\_CLK, по которому осуществляется регистрация данных в приемнике ведомого модуля SSP, время установки сигнала SSP\_FSS должно быть как минимум в два раза больше периода SSP\_CLK, на котором работает модуль. По отношению к предыдущему переднему фронту сигнала SSP\_CLK должно обеспечиваться время удержания не менее одного периода этого тактового сигнала.



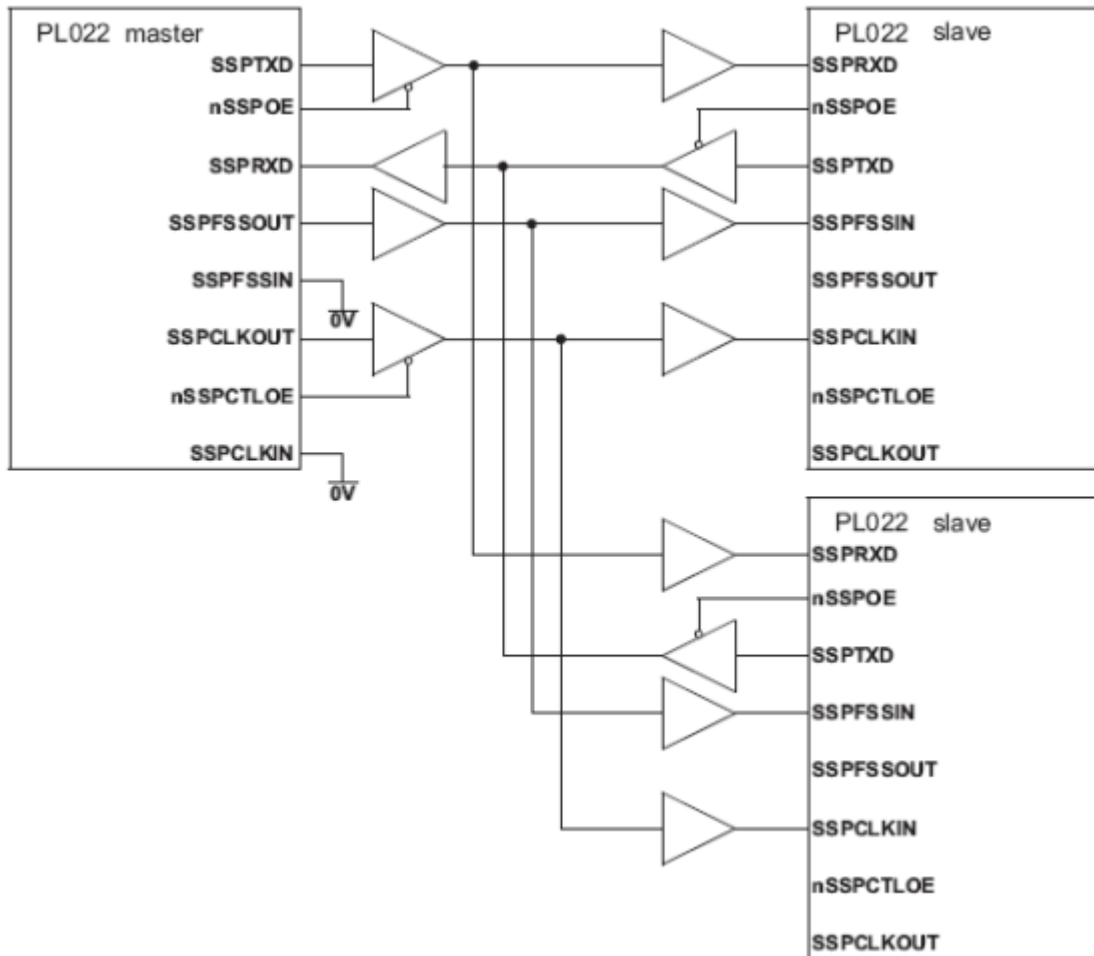
**Рис. 74** Формат кадра Microwire, требования к времени установки и удержания сигнала SSPFSSIN

**Примеры конфигурации модуля в ведущем и ведомом режимах**

На Рис. 75, Рис. 76 и Рис. 77 показаны варианты подключения модуля PrimeCell SSP (PL022) к периферийным устройствам, работающим в ведущем или ведомом режиме.

**Примечание**

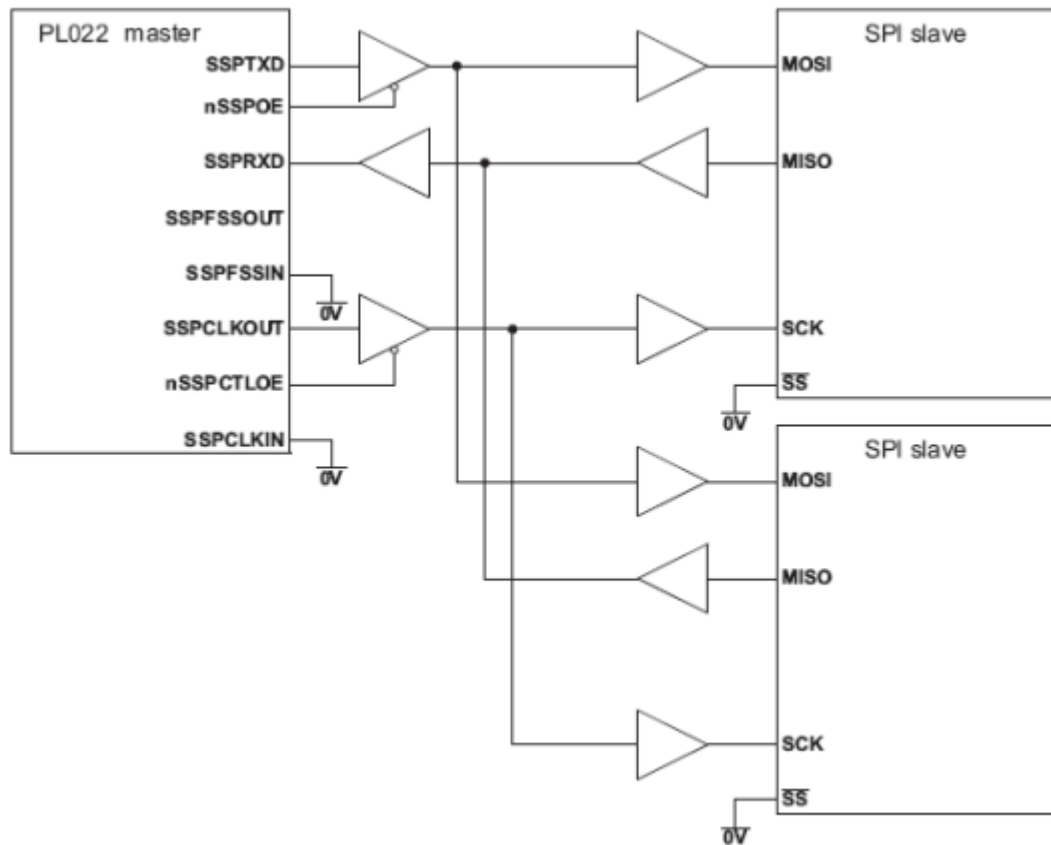
Модуль SSP не поддерживает динамическое изменение режима ведущий – ведомый. Каждый приемопередатчик должен быть изначально сконфигурирован в одном из этих режимов.



**Рис. 75** Ведущее устройство SSP подключено к двум ведомым

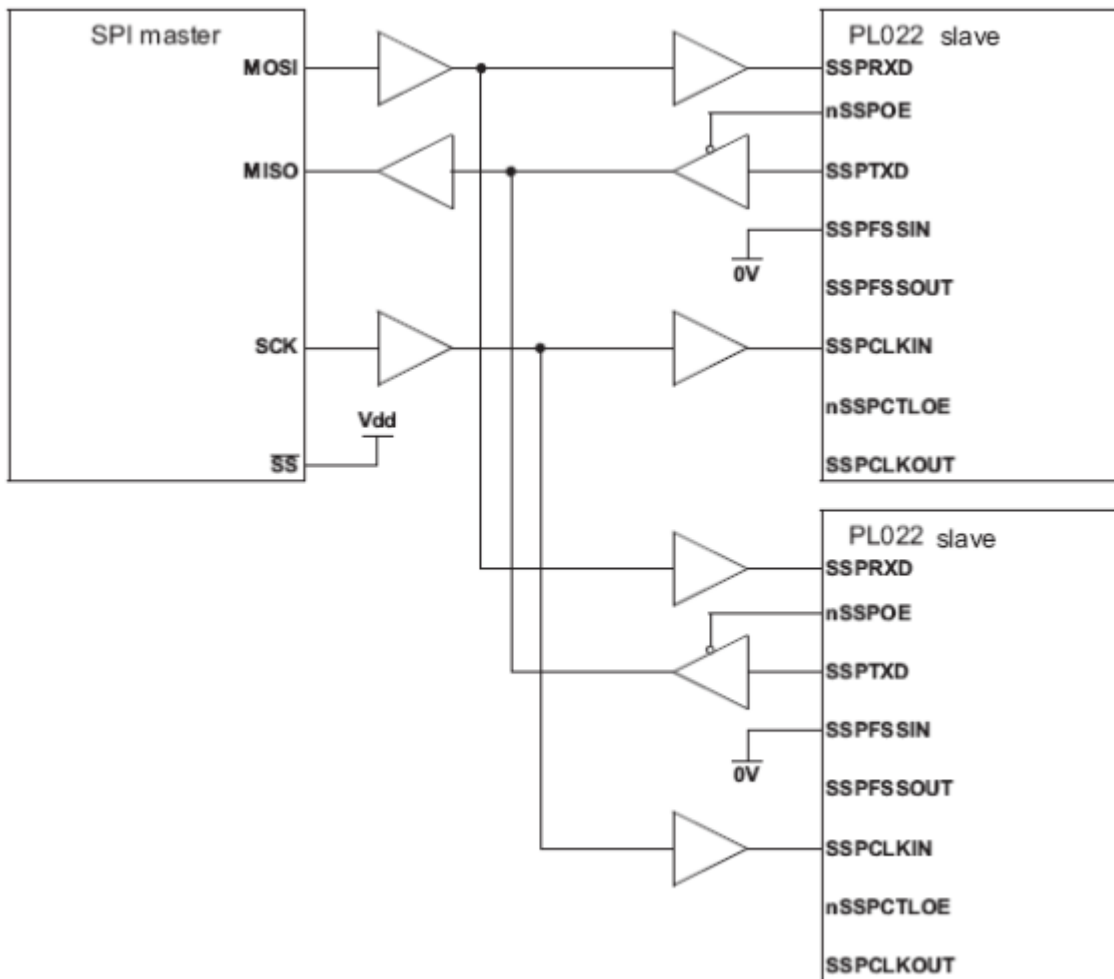
Рис. 75 показывает совместную работу трех модулей SSP, один из которых сконфигурирован в качестве ведущего, а два – в качестве ведомых устройств. Ведущее устройство способно передавать данные циркулярно в адрес двух ведомых по линии SSP\_TXD.

Для ответной передачи данных один из ведомых модулей разрешает прохождение сигнала от своей линии SSP\_TXD на вход SSP\_RXD ведущего.



**Рис. 76** Ведущее устройство SSP подключено к двум ведомым, поддерживающим протокол SPI

Рис. 76 показывает подключение модуля SSP, сконфигурированного как ведущее устройство, к двум ведомым устройствам, поддерживающим протокол SPI фирмы Motorola. Внешние устройства сконфигурированы как ведомые путем установки в низкий логический уровень сигнала выбора ведомого устройства Slave Select (SS). Как и в предыдущем примере, ведущее устройство способно передавать данные в адрес ведомых циркулярно по линии SSP\_TXD. Ответная передача данных на входную линию SSP\_RXD ведущего устройства одновременно осуществляется только одним из ведомых по соответствующей линии MISO.



**Рис. 77** Ведущее устройство, поддерживающее протокол SPI подключено к двум ведомым модулям SSP

Рис. 77 показывает ведущее устройство, поддерживающее протокол SPI фирмы Motorola, соединенное с двумя модулями SSP, сконфигурированными для работы в ведомом режиме. Линия Slave Select (SS) ведущего устройства в этом случае установлена в высокий логический уровень. Ведущее устройство осуществляет передачу данных по линии MOSI циркулярно в адрес двух ведомых модулей.

Для ответной передачи данных один из ведомых модулей переводит линию SSP\_TXD в активное состояние, разрешая таким образом прохождение сигнала от своей линии SSP\_TXD на вход SSP\_RXD ведущего.



### **Интерфейс прямого доступа к памяти**

Модуль SSP предоставляет интерфейс подключения к контроллеру прямого доступа к памяти. Работа в данном режиме контролируется регистром управления DMA SSPDMACR.

Интерфейс DMA включает в себя следующие сигналы:

Для приема:

- SSPRXDMASREQ – запрос передачи отдельного символа, инициируется приемопередатчиком. Сигнал переводится в активное состояние в случае, если буфер FIFO приемника содержит по меньшей мере один символ;
- SSPRXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переходит в активное состояние в случае, если буфер FIFO приемника содержит четыре или более символов;
- SSPRXDMACLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае, если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

Для передачи:

- SSPTXDMASREQ – запрос передачи отдельного символа, инициируется модулем приемопередатчика. Сигнал переводится в активное состояние в случае, если буфер FIFO передатчика содержит по меньшей мере одну свободную ячейку;
- SSPTXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переводится в активное состояние в случае, если буфер FIFO передатчика содержит четыре или менее символов;
- SSPTXDMACLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае, если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

Сигналы блочного и одноэлементного обмена данными не являются взаимоисключающими, они могут быть инициированы одновременно. Например, в случае, если заполнение данными буфера приемника превышает пороговое значение четыре, формируются как сигнал запроса одноэлементного обмена, так и сигнал запроса блочного обмена данными. В случае, если количество данных в буфере приема меньше порогового значения, формируется только запрос одноэлементного обмена. Это бывает полезно в ситуациях, при которых объем данных меньше размера блока. Пусть, например, нужно принять 19 символов. Тогда контроллер DMA осуществит четыре передачи блоков по четыре символа, а оставшиеся три символа передаст в ходе трех одноэлементных обменов.

#### *Примечание*

Для оставшихся трех символов контроллер SSP не инициирует процедуру блочного обмена.

Каждый инициированный приемопередатчиком сигнал запроса DMA остается активным до момента его сброса соответствующим сигналом DMACLR.

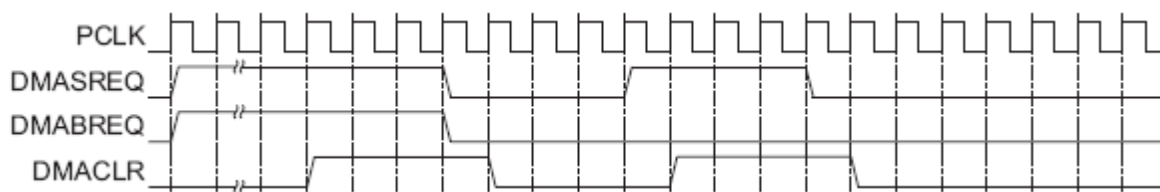
После снятия сигнала сброса модуль приемопередатчика вновь получает возможность сформировать запрос на DMA в случае выполнения описанных выше условий. Все запросы DMA снимаются после запрета работы приемопередатчика, а также в случае снятия сигнала разрешения DMA.

В Таблица 130 приведены значения порогов заполнения буферов приемника и передатчика, необходимых для срабатывания запросов блочного обмена DMABREQ.

**Таблица 130** Параметры срабатывания запросов блочного обмена данными в режиме ПДП

Пороговый уровень	Длина блока обмена данными	
	Буфер передатчика (количество незаполненных ячеек)	Буфер приемника (количество заполненных ячеек)
1/2	4	4

На Рис. 78 показаны временные диаграммы одноэлементного и блочного запросов ПДП, в том числе действие сигнала DMACLR. Все сигналы должны быть синхронизированы с PCLK.



**Рис. 78** Временные диаграммы обмена в режиме ПДП

**Программное управление модулем**

**Общая информация**

Следующие адреса являются резервными и не должны использоваться в нормальном режиме функционирования:

- адреса в со смещениями в диапазоне +0x028 ... +0x07C и +0xFD0 ... +0xFDC зарезервированы для перспективных расширений возможностей модуля;
- адреса в со смещениями в диапазоне +0x080 ... +0x088 зарезервированы для тестирования.

**Описание регистров контроллера SSP**

Данные о регистрах модуля PrimeCell SSP приведены в Таблица 131.

**Таблица 131** Обобщенные данные о регистрах модуля PrimeCell SSP

Сме- щение	Наимено- вание	Тип	Значен ие после сброса	Раз- мер, бит	Описание
0x000	SSPx_CR0	RW	0x0000	16	Регистр управления 0 (Таблица 132)
0x004	SSPx_CR1	RW	0x0	4	Регистр управления 1 (Таблица 133)
0x008	SSPx_DR	RW	0x----	16	Буфера FIFO приемника (чтение) Буфер FIFO передатчика (запись) (Таблица 134)
0x00C	SSx_PSR	RO	0x03	3	Регистр состояния (Таблица 135)
0x010	SSPx_CPSR	RW	0x00	8	Регистр делителя тактовой частоты (Таблица 136)
0x014	SSPx_IMSC	RW	0x0	4	Регистр маски прерывания (Таблица 137)
0x018	SSPx_RIS	RO	0x8	4	Регистр состояния прерываний без учета маскирования (Таблица 138)
0x01C	SSPx_MIS	RO	0x0	4	Регистр состояния прерываний с учетом маскирования (Таблица 139)
0x020	SSPx_ICR	WO	0x0	4	Регистр сброса прерывания (Таблица 140)
0x024	SSPx_DMAR	RW	0x0	2	Регистр управления прямым доступом к памяти (Таблица 141)

**Примечание**

В поле «тип» указан вид доступа к регистру: RW – чтение и запись, RO – только чтение, WO – только запись.

**SSPx\_CR0. Регистр управления 0**

Регистр SSPx\_CR0 содержит пять битовых полей, предназначенных для управления блоками модуля SSP. Назначение разрядов регистра представлено в Таблица 132.

**Таблица 132** Формат регистра SSPCR0

Бит	Наименование	Назначение
15:8	SCR	Скорость последовательного обмена. Значение поля SCR используется при формировании тактового сигнала обмена данными. Информационная скорость удовлетворяет соотношению: $F_{SSPCLK} / (CPSDVR * (1 + SCR))$ , где CPSDVR – четное число в диапазоне от 2 до 254 (см. регистр SSPCPSR), а SCR – число от 0 до 255.
7	SPH	Фаза сигнала SSP_CLK (используется только в режиме обмена SPI фирмы Motorola). См. раздел «Формат синхронного обмена SPI фирмы Motorola» .
6	SPO	Полярность сигнала SSP_CLK (используется только в режиме обмена SPI фирмы Motorola). См. раздел «Формат синхронного обмена SPI фирмы Motorola» .
5:4	FRF	Формат информационного кадра. 00 – протокол SPI фирмы Motorola; 01 – протокол SSI фирмы Texas Instruments; 10 – протокол Microwire фирмы National Semiconductor; 11 – резерв.
3:0	DSS	Размер слова данных. 0000 – резерв. 0001 – резерв. 0010 – резерв. 0011 – 4 бита. 0100 – 5 бит. 0101 – 6 бит. 0110 – 7 бит. 0111 – 8 бит. 1000 – 9 бит. 1001 – 10 бит. 1010 – 11 бит. 1011 – 12 бит. 1100 – 13 бит. 1110 – 14 бит. 1111 – 15 бит.

**SSPx\_CR1. Регистр управления 1**

Регистр SSPx\_CR1 содержит четыре битовых поля, предназначенных для управления блоками модуля SSP. Назначение разрядов регистра представлено в

Таблица 133.

**Таблица 133** Регистр SSPx\_CR1

Биты	Наименование	Назначение
15:4		Резерв, при чтении результат не определен. При записи следует устанавливать в 0.
3	SOD	Запрет выходных линий в режиме ведомого устройства. Бит используется только в режиме ведомого устройства (MS=1). Это позволяет организовать двусторонний обмен данными в системах, содержащих одно ведущее и несколько ведомых устройств. Бит SOD следует установить в случае, если данный ведомый модуль SSP не должен в настоящее время осуществлять передачу данных в линию SSP_TXD. При этом линии обмена данных ведомых устройств можно соединить параллельно. 0 – управление линией SSP_TXD в ведомом режиме разрешена. 1 – управление линией SSP_TXD в ведомом режиме запрещена.
2	MS	Выбор ведущего или ведомого режима работы: 0 – ведущий модуль (устанавливается по умолчанию); 1 – ведомый модуль.
1	SSE	Разрешение работы приемопередатчика: 0 – работа запрещена; 1 – работа разрешена.
0	LBM	Тестирование по шлейфу: 0 – нормальный режим работы приемопередатчика; 1 – выход регистра сдвига передатчика соединен со входом регистра сдвига приемника.

### **SSPx\_DR. Регистр данных**

Регистр SSP\_xDR имеет разрядность 16 бит и предназначен для чтения принятых и записи передаваемых данных.

Операция чтения обеспечивает доступ к последней несчитанной ячейке буфера FIFO приемника. Запись данных в этот буфер FIFO осуществляет блок приемника.

Операция записи позволяет занести очередное слово в буфер FIFO передатчика. Извлечение данных из этого буфера осуществляет блок передатчика. При этом извлеченные данные помещаются в регистр сдвига передатчика, откуда последовательно выдаются на линию SSP\_TXD с заданной скоростью информационного обмена.

В случае, если выбран размер информационного слова менее 16 бит, перед записью в регистр SSPx\_DR необходимо обеспечить выравнивание данных по правой границе. Блок передатчика игнорирует неиспользуемые биты. Принятые информационные слова автоматически выравниваются по правой границе в блоке приемника.

В режиме обмена данными Microwire фирмы National Semiconductor модуль SSP по умолчанию работает с восьмиразрядными информационными словами (старший значащий байт игнорируется). Размер принимаемых данных задается программно. Буфера FIFO приемника и передатчика автоматически не очищаются даже в случае, если бит SSE установлен в 0. Это позволяет заполнить буфер передатчика необходимой информацией заблаговременно, перед разрешением работы модуля.

Назначение разрядов регистра SSPx\_DR описано в Таблица 134.

**Таблица 134** Формат регистра SSPx\_DR

Бит	Наименование	Назначение
15:0	DATA	Принимаемые данные (чтение) Передаваемые данные (запись) В случае, если выбран размер информационного слова менее 16 бит, перед записью в регистр SSPx_DR необходимо обеспечить выравнивание данных по правой границе. Блок передатчика игнорирует неиспользуемые биты. Принятые информационные слова автоматически выравниваются по правой границе в блоке приемника.

**SSPx\_SR. Регистр состояния**

Регистр состояния доступен только для чтения и содержит информацию о состоянии буферов FIFO приемника и передатчика и занятости модуля SSP.

В Таблица 135 представлено назначение бит регистра SSPx\_SR.

**Таблица 135** Регистр SSPx\_SR

Биты	Наименование	Назначение
15:5		Резерв, при чтении результат не определен.
4	BSY	Флаг занятости модуля: 0 – модуль SSP неактивен; 1 – модуль SSP в настоящее время передает и/или принимает данные, либо буфер FIFO передатчика не пуст.
3	RFF	Буфер FIFO приемника заполнен: 0 – не заполнен; 1 – заполнен.
2	RNE	Буфер FIFO приемника не пуст: 0 – пуст; 1 – не пуст.
1	TNF	Буфер FIFO передатчика не заполнен: 0 – заполнен; 1 – не заполнен.
0	TFE	Буфер FIFO передатчика пуст: 0 – не пуст; 1 – пуст.

**SSPx\_CPSR. Регистр делителя тактовой частоты**

Регистр SSPx\_CPSR используется для установки параметров делителя тактовой частоты. Записываемое значение должно быть целым числом в диапазоне от 2 до 254. Младший значащий разряд регистра принудительно устанавливается в ноль. Если записать в регистр SSPx\_CPSR нечетное число, его последующее чтение даст результатом это число, но с установленным в ноль младшим битом.

Назначение бит регистра SSPx\_CPSR представлено в Таблица 136.

**Таблица 136** Регистр SSPx\_CPSR

Биты	Наименование	Назначение
15:8		Резерв, при чтении результат не определен. При записи следует заполнить нулями.
7:0	CPSDVSR	Коэффициент деления тактовой частоты. Записываемое значение должно быть целым числом в диапазоне от 2 до 254. Младший значащий разряд регистра принудительно устанавливается в ноль.

**SSPx\_IMSC. Регистр установки и сброса маски прерывания**

При чтении выдается текущее значение маски. При записи производится установка или сброс маски на соответствующее прерывание. При этом запись 1 в разряд разрешает соответствующее прерывание, запись 0 – запрещает.

После сброса все биты регистра маски устанавливаются в нулевое состояние.

Назначение битов регистра SSPx\_IMSC показано в Таблица 137.

**Таблица 137** Регистр SSP\_xIMSC

Биты	Наименование	Назначение
15:4		Резерв. Не модифицируйте. При чтении выдаются нули.
3	TXIM	Маска прерывания по заполнению на 50% и менее буфера FIFO передатчика. 1 – не маскирована, 0 – маскирована.
2	RXIM	Маска прерывания по заполнению на 50% и менее буфера FIFO приемника. 1 – не маскирована, 0 – маскирована.
1	RTIM	Маска прерывания по таймауту приемника (буфер FIFO приемника не пуст и не было попуток его чтения в течение времени таймаута). 1 – не маскирована, 0 – маскирована.
0	RORIM	Маска прерывания по переполнению буфера приемника. 1 – не маскирована, 0 – маскирована.

**SSPx\_RIS. Регистр состояния прерываний**

Этот регистр доступен только для чтения и содержит текущее состояние прерываний без учета маскирования. Данные, записываемые в регистр, игнорируются.

Назначение бит в регистре SSPx\_RIS представлено в Таблица 138.

**Таблица 138** Регистр SSP\_xRIS

Биты	Наименование	Назначение
15:4		Резерв. Не модифицируйте. При чтении выдаются нули.
3	TXRIS	Состояние до маскирования прерывания SSPTXINTR.
2	RXRIS	Состояние до маскирования прерывания SSPRXINTR.
1	RTRIS	Состояние до маскирования прерывания SSPRTINTR.
0	RORRIS	Состояние до маскирования прерывания SSPRORINTR.

**SSPx\_MIS. Регистр маскированного состояния прерываний**

Этот регистр доступен только для чтения и содержит текущее состояние прерываний с учетом маскирования. Данные, записываемые в регистр, игнорируются.

Назначение бит в регистре SSPx\_MIS представлено в Таблица 139.

**Таблица 139** Регистр SSPx\_MIS

Биты	Наименование	Назначение
15:4		Резерв. Не модифицируйте. При чтении выдаются нули.
3	TXMIS	Состояние маскированного прерывания SSPTXINTR.
2	RXMIS	Состояние маскированного прерывания SSPRXINTR.
1	RTMIS	Состояние маскированного прерывания SSPRTINTR.
0	RORMIS	Состояние маскированного прерывания SSPRORINTR.

**SSPx\_ICR. Регистр сброса прерываний**

Этот регистр доступен только для записи и предназначен для сброса признака прерывания по заданному событию путем записи 1 в соответствующий бит. Запись в любой из разрядов регистра 0 игнорируется.

Назначение бит в регистре SSPx\_ICR представлено в Таблица 140.

**Таблица 140** Регистр SSPx\_ICR

Биты	Наименование	Назначение
15:2		Резерв. Не модифицируйте. При чтении выдаются нули.
1	RTIC	Сброс прерывания SSPRTINTR.



0	RORIC	Сброс прерывания SSPRORINTR.
---	-------	------------------------------

**SSPx\_DMACR. Регистр управления прямым доступом к памяти**

Регистр доступен по чтению и записи. После сброса все биты регистра обнуляются.

Назначение бит регистра SSPx\_DMACR представлено в Таблица 141.

**Таблица 141** Регистр SSPx\_DMACR

Биты	Наименование	Назначение
15:2		Резерв. Не модифицируйте. При чтении выдаются нули.
1	TXDMAE	Использование ПДП при передаче. Если бит установлен в 1, разрешено формирование запросов ПДП для обслуживания буфера FIFO передатчика.
0	RXDMAE	Использование ПДП при приеме. Если бит установлен в 1, разрешено формирование запросов ПДП для обслуживания буфера FIFO приемника.

## **Прерывания**

В модуле предусмотрено пять маскируемых линий запроса на прерывание, в том числе, четыре независимые линии запроса с активным высоким логическим уровнем, а также один общий сигнал, представляющий собой комбинацию независимых по схеме ИЛИ.

### **Сигналы запроса на прерывание:**

SSPRXINTR – запрос на обслуживание буфера FIFO приемника.

SSPTXINTR – запрос на обслуживание буфера FIFO передатчика.

SSPRORINTR – переполнение буфера FIFO приемника.

SSPRTINTR – таймаут приемника.

SSPINTR – логическое ИЛИ сигналов SSPRXINTR, SSPTXINTR, SSPRTINTR и SSPRORINTR.

Каждый из независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре маски SSPx\_IMSC. Установка бита в 1 разрешает соответствующее прерывание, в 0 – запрещает.

Доступность как индивидуальных, так и общей линии запроса позволяет организовать обслуживание прерываний в системе как путем применения глобальной процедуры обработки, так и с помощью драйвера устройства, построенного по модульному принципу.

Прерывания от приемника и передатчика SSPRXINTR и SSPTXINTR выведены отдельно от прерываний по изменению состояния устройства. Это позволяет использовать данные сигналы запроса для обеспечения чтения и записи данных, согласованной с достижением заданного порога заполнения буферов FIFO приемника и передатчика.

Признаки возникновения каждого из условий прерывания можно считать либо из регистра прерываний SSPx\_RIS, либо из маскированного регистра прерываний SSPx\_MIS.

## **SSPRXINTR**

Прерывание по заполнению буфера FIFO приемника формируется в случае, если буфер приемника содержит четыре или более несчитанных слов данных.

## **SSPTXINTR**

Прерывание по заполнению буфера FIFO передатчика формируется в случае, если буфер передатчика содержит четыре или менее корректных слов данных.

Состояние прерывания не зависит от значения сигнала разрешения работы модуля PrimeCell SSP. Это позволяет организовать взаимодействие программного обеспечения с передатчиком одним из двух способов. Во-первых, можно записать данные в буфер заблаговременно, перед активизацией передатчика и разрешения прерываний. Во-вторых, можно предварительно разрешить работу модуля и формирование прерываний и заполнять буфер передатчика в ходе работы процедуры обслуживания прерываний.

**SSPRORINTR**

Прерывание по переполнению буфера FIFO приемника формируется в случае, если буфер уже заполнен и блоком приемника осуществлена попытка записать в него еще одно слово. При этом принятое слово данных регистрируется в регистре сдвига приемника, но в буфер приемника не заносится.

**SSPRTINTR**

Прерывание по таймауту приемника возникает в случае, если буфер FIFO приемника не пуст, и на вход приемника не поступало новых данных в течение периода времени, необходимого для передачи 32 бит. Данный механизм гарантирует, что пользователь будет знать о наличии в буфере приемника необработанных данных.

Прерывание по таймауту снимается либо после считывания данных из буфера приемника до его опустошения, либо после приема новых слов данных по входной линии SSPRXD. Кроме того, оно может быть снято путем записи 1 в бит RTIC регистра сброса прерывания SSPTICR.

**SSPINTR**

Все описанные сигналы запроса на прерывание скомбинированы в общую линию путем объединения по схеме ИЛИ сигналов SSPRXINTR, SSPTXINTR, SSPRTINTR и SSPRORINTR с учетом маскирования. Общий выход может быть подключен к системному контроллеру прерывания, что позволит ввести дополнительное маскирование запросов на уровне периферийных устройств.

## **Контроллер UART**

Модуль универсального асинхронного приемопередатчика (UART – Universal Synchronous Asynchronous Receiver Transmitter) представляет собой периферийное устройство микроконтроллера.

В состав контроллера включен кодек (ENDEC – ENcoder/DEcoder) последовательного интерфейса инфракрасной (ИК) передачи данных в соответствии с протоколом SIR (SIR – Serial Infra Red) ассоциации Infrared Data Association (IrDA).

### **Основные характеристики модуля UART**

Может быть запрограммирован для использования как в качестве универсального асинхронного приемопередатчика, так и для инфракрасного обмена данными (SIR).

Содержит независимые буферы приема (16x12) и передачи (16x8) типа FIFO (First In First Out – первый вошел, первый вышел), что позволяет снизить интенсивность прерываний центрального процессора.

Программное отключение FIFO позволяет ограничить размер буфера одним байтом.

### **Программное управление скоростью обмена.**

Обеспечивается возможность деления тактовой частоты опорного генератора в диапазоне (1x16 – 65535x16). Допускается использование нецелых коэффициентов деления частоты, что позволяет использовать любой опорный генератор с частотой более 3.6864 МГц.

Поддержка стандартных элементов асинхронного протокола связи – стартового и стопового бит, а та же бита контроля четности, которые добавляются перед передачей и удаляются после приема.

Независимое маскирование прерываний от буфера FIFO передатчика, буфера FIFO приемника, по таймауту приемника, по изменению линий состояния модема, а также в случае обнаружения ошибки.

Поддержка прямого доступа к памяти.

Обнаружение ложных стартовых бит.

Формирование и обнаружения сигнала разрыва линии.

Функция управления модемом (линии nUART0CTS, nUART0DCD, nUART0DSR, nUART0RTS, nUART0DTR и nUART0RI (см. Таблица 1)).

Возможность организации аппаратного управления потоком данных.

Полностью программируемый асинхронный последовательный интерфейс с характеристиками:

- данные длиной 5, 6, 7 или 8 бит;
- формирование и контроль четности (проверочный бит выставляется по четности, нечетности, имеет фиксированное значение, либо не передается);
- формирование 1 или 2 стоповых бит;
- скорость передачи данных – от 0 до UARTCLK/16 Бод.
- Кодек ИУ обмена данными IrDA SIR обеспечивает:
- программный выбор обмена данными по линиям асинхронного приемопередатчика либо кодека ИК связи IrDA SIR;
- поддержку функционирования с информационной скоростью до 115200 бит/с в режиме полудуплекса;

- поддержку длительности бит для нормального режима (3/16) и для режима пониженного энергопотребления (1.41 – 2.23 мкс);
- программируемое деление опорной частоты UARTCLK для получения заданной длительности бит в режиме пониженного энергопотребления.

Наличие идентификационного регистра, однозначно идентифицирующего модуль, что позволяет операционной системе выполнять автоматическую конфигурацию.

### **Программируемые параметры**

Следующие ключевые параметры могут быть заданы программно:

- скорость передачи данных – целая и дробная часть числа;
- количество бит данных;
- количество стоповых бит;
- режим контроля четности;
- разрешение или запрет использования буферов FIFO (глубина очереди данных – 16 элементов или один элемент, соответственно);
- порог срабатывания прерывания по заполнению буферов FIFO (1/8, 1/4, 1/2, 3/4 и 7/8);
- частота внутреннего тактового генератора (номинальное значение - 1.8432 МГц) может быть задана в диапазоне 1.42 – 2.12 МГц для обеспечения возможности формирования бит данных с укороченной длительностью в режиме пониженного энергопотребления;
- режим аппаратного управления потоком данных.

### **Отличительные особенности контроллера UART**

Контроллер отличается от промышленного стандарта асинхронного приемопередатчика следующими характеристиками:

- пороги срабатывания прерывания по заполнению буфера FIFO приемника – 1/8, 1/4, 1/2, 3/4 и 7/8;
- пороги срабатывания прерывания по заполнению буфера FIFO передатчика – 1/8, 1/4, 1/2, 3/4 и 7/8;
- отличается распределение адресов внутренних регистров и назначение бит в регистрах;
- недоступны изменения сигналов состоянии модема.

Следующие возможности контроллера не поддерживаются:

- полуторная длительность стопового бита (поддерживается только 1 или 2 стоповых бита);
- независимое задание тактовой частоты приемника и передатчика.

### **Функциональные возможности**

Устройство выполняет следующие функции:

- преобразование данных, полученных от периферийного устройства, из последовательной в параллельную форму;

- преобразование данных, передаваемых на периферийное устройство, из параллельной в последовательную форму.

Процессор читает и записывает данные, а также управляющую информацию и информацию о состоянии модуля. Прием и передача данных буферизуются с помощью внутренней памяти FIFO, позволяющей сохранить до 16 байтов независимо для режимов приема и передачи.

### **Модуль приемопередатчика:**

- содержит программируемый генератор, формирующий тактовый сигнал одновременно для передачи и для приема данных на основе внутреннего тактового сигнала UARTCLK;
- обеспечивает возможности, сходные с возможностями промышленного стандарта - контроллера UART;
- позволяет осуществлять обмен информацией с максимальной скоростью:
  - в режиме UART – до 921600 бит/с;
  - в режиме IrDA – до 460800 бит/с;
  - в режиме IrDA с пониженным энергопотреблением – до 115200 бит/с.

Режим работы приемопередатчика и скорость обмена данными контролируются регистром управления линией UARTLCR\_H и регистрами делителя скорости передачи данных – целой части (UARTIBRD) и дробной части (UARTFBRD).

Устройство может формировать следующие сигналы:

- независимые маскируемые прерывания от приемника (в том числе по таймауту), передатчика, а также по изменению состояния модема и в случае обнаружения ошибки;
- общее прерывание, возникающее в случае, если возникло одно из независимых немаскированных прерываний;
- сигналы запроса на прямой доступ к памяти (DMA) для совместной работы с контроллером DMA.

В случае возникновения ошибки в структуре сигнала, четности данных, а также разрыва линии соответствующий бит ошибки устанавливается и сохраняется в буфере FIFO. В случае переполнения буфера немедленно устанавливается соответствующий бит в регистре переполнения, а доступ к записи в буфер FIFO блокируется.

Существует возможность программно ограничить размер буфера FIFO одним байтом, что позволяет реализовать общепринятый интерфейс асинхронной последовательной связи с двойной буферизацией.

Поддерживаются входные линии состояния модема: «готовность к приему» (Clear To Send, CTS), «обнаружен информационный сигнал» (Data Carrier Detected, DCD), «источник данных готов» (Data Set Ready, DSR) и «индикатор вызова» (Ring Indicator, RI), а также выходные линии: «запрос на передачу» (Request to Send, RTS) и «приемник данных готов» (Data Terminal Ready, DTR). Доступна возможность аппаратного управления потоком данных по линиям nUARTCTS и nUARTRTS.

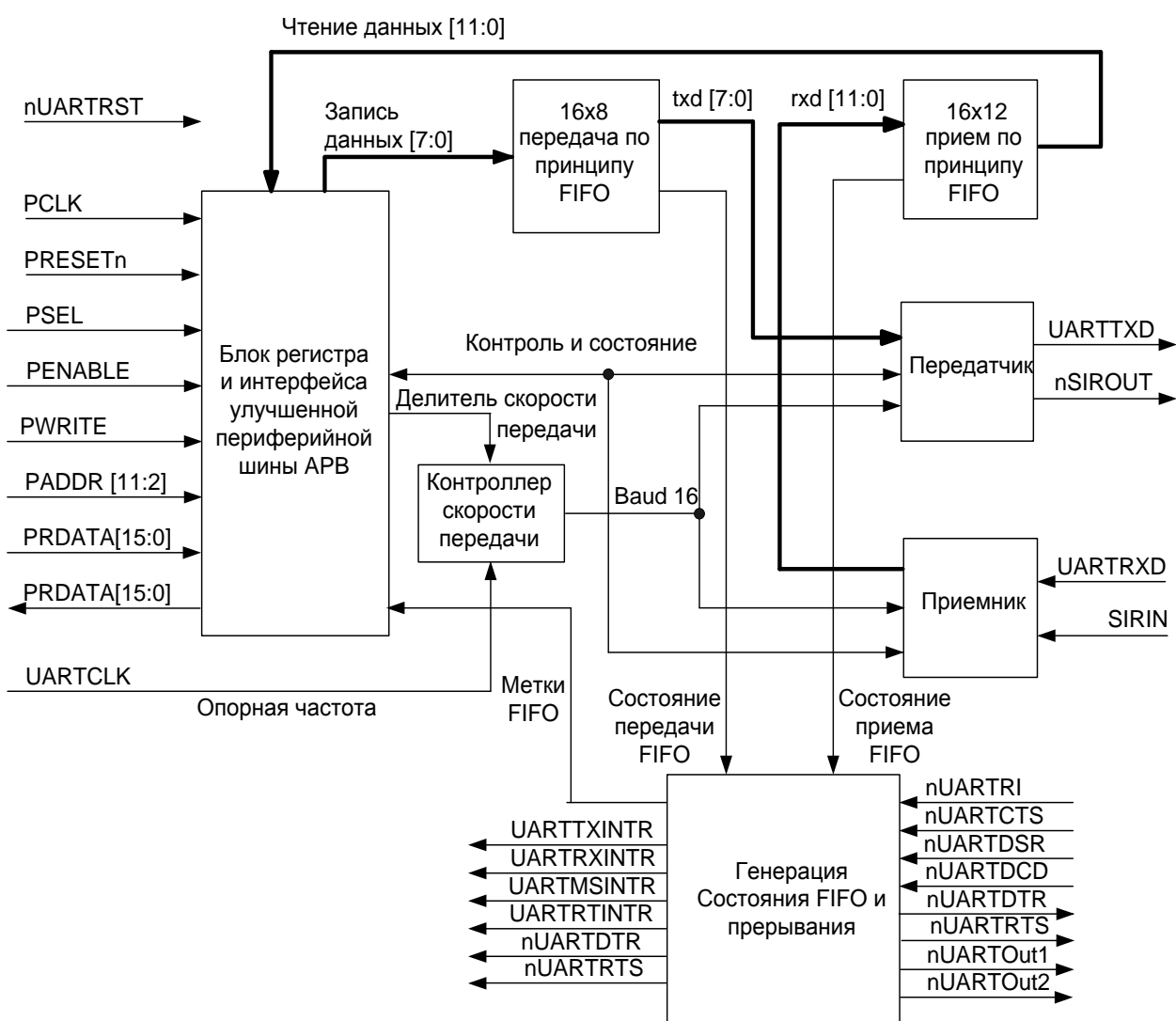
Блок последовательного интерфейса инфракрасной передачи данных в соответствии с протоколом IrDA SIR реализует протокол обмена

данными ENDEC. В случае его активизации обмен информацией осуществляется не с помощью сигналов UARTTXD и UARTRXD, а посредством сигналов nSIROUT и SIRIN.

В этом случае устройство переводит линию UARTTXD в пассивное состояние (высокий уровень), и перестает реагировать на изменение состояния модема, а также сигнала на линии UARTRXD. Протокол SIR ENDEC обеспечивает возможность обмена данными исключительно в режиме полудуплекса, то есть он не может передавать во время приема данных и принимать во время передачи данных.

В соответствии со спецификацией физического уровня протокола IrDA SIR, задержка между передачей и приемом должна составлять не менее 10 мс.

### Описание функционирования блока UART



**Рис. 79** Блок-схема универсального асинхронного приёмопередатчика (УАПП)

### Генератор тактового сигнала приёмопередатчика

Генератор содержит счетчики без цепи сброса, формирующие внутренние тактовые сигналы Baud16 и IrLPBaud16.

Сигнал Baud16 используется для синхронизации схем управления приемником и передатчиком последовательного обмена данными. Он представляет собой последовательность импульсов с шириной, равной одному периоду сигнала UARTCLK и частотой, в 16 раз выше скорости передачи данных. Сигнал IrLPBaud16 предназначен для синхронизации схемы формирования импульсов с длительностью, требуемой для ИК обмена данными в режиме с пониженным энергопотреблением.

### **Буфер FIFO передатчика**

Буфер передатчика имеет ширину 8 бит, глубину 16 слов, схему организации доступа типа «первый вошел, первый вышел». Данные от центрального процессора, записанные через шину APB, сохраняются в буфере до тех пор, пока не будут считаны логической схемой передачи данных. Существует возможность запретить буфер FIFO передатчика, в этом случае он будет функционировать как однобайтовый буферный регистр.

### **Буфер FIFO приемника**

Буфер приемника имеет ширину 12 бит, глубину 16 слов, схему организации доступа типа «первый вошел, первый вышел». Принятые от периферийного устройства данные и соответствующие коды ошибки сохраняются логикой приема данных в нем до тех пор, пока не будут считаны центральным процессором через шину APB. Буфер FIFO приемника может быть запрещен, в этом случае он будет действовать как однобайтовый буферный регистр.

### **Блок передатчика**

Логические схемы передатчика осуществляют преобразование данных, считанных из буфера передатчика, из параллельной в последовательную форму. Управляющая логика выдает последовательный поток бит в порядке: стартовый бит, биты данных, начиная с младшего значащего разряда, бит проверки на четность, и, наконец, стоповые биты, в соответствии с конфигурацией, записанной в регистре управления.

### **Блок приемника**

Логические схемы приемника преобразуют данные, полученные от периферийного устройства, из последовательной в параллельную форму после обнаружения корректного стартового импульса. Кроме того, производятся проверки переполнения буфера, проверки на ошибки контроля четности, на ошибки в структуре сигнала, а также на разрыв линии. Признаки обнаружения этих ошибок также сохраняются в выходном буфере.

### **Блок формирования прерываний**

Контроллер генерирует независимые маскируемые прерывания с активным высоким уровнем. Кроме того, формируется комбинированное прерывание путем объединения указанных независимых прерываний по схеме ИЛИ.

Комбинированный сигнал прерывания может быть подан на внешний контроллер прерываний системы, при этом появится дополнительная возможность маскирования устройства в целом, что облегчает построение модульных драйверов устройств.



Другой подход состоит в подаче на системный контроллер прерываний независимых линий запроса на прерывание от приемопередатчика. В этом случае процедура обработки сможет одновременно считать информацию обо всех источниках прерывания. Данный подход привлекателен в случае, если скорость доступа к регистрам периферийных устройств значительно превышает тактовую частоту центрального процессора в системе реального времени.

Для более подробной информации см. раздел «Прерывания».

### Интерфейс прямого доступа к памяти

Модуль обеспечивает интерфейс с контроллером DMA согласно схеме взаимодействия приемопередатчика и контроллера DMA.

### Блок и регистры синхронизации

Контроллер поддерживает как асинхронный, так и синхронный режимы работы тактовых генераторов CPU\_CLK и UARTCLK. Регистры синхронизации и логика квитирования постоянно находятся в активном состоянии. Это практически не отражается на характеристиках устройства и занимаемой площади. Синхронизация сигналов управления осуществляется в обоих направлениях потока данных, то есть как из области действия CPU\_CLK в область действия UARTCLK, так и наоборот.

### Описание функционирования ИК кодека IrDA SIR

Структурная схема кодека представлена на Рис. 80.

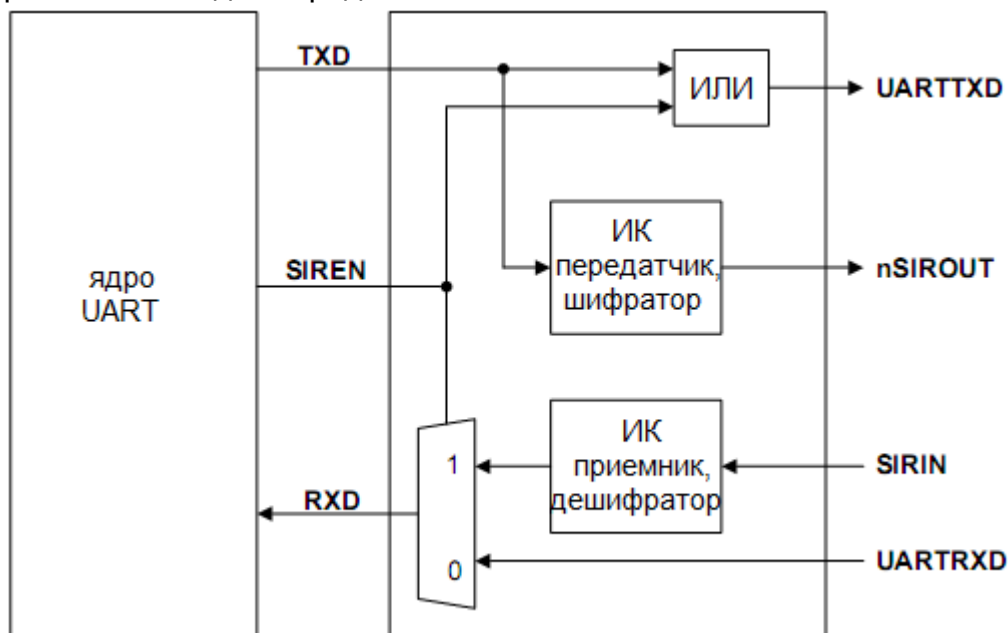


Рис. 80

### Кодер ИК передатчика

Кодер преобразует поток данных с выхода асинхронного передатчика, сформированный по закону модуляции без возврата к нулю (NRZ). Спецификация физического уровня протокола IrDA SIR подразумевает использование модуляции с возвратом к нулю и инверсией (RZI), в соответствии с которой передача

логического нуля соответствует излучению одного светового ИК импульса. Сформированный выходной поток импульсов подается на усилитель и, далее, на ИК светодиод.

Длительность импульса в режиме IrDA составляет, согласно спецификации, 3 периода внутреннего тактового генератора с частотой Baud16, то есть 3/16 периода времени, выделенного на передачу одного бита.

В режиме IrDA с пониженным энергопотреблением ширина импульса задана как 3/16 периода, выделенного на передачу бита, при скорости передачи данных 115200 бит/с. Данное требование реализуется за счет формирования трех периодов тактового сигнала IrLPBaud16 с номинальной частотой 1.8432 МГц, в свою очередь, формируемого путем деления частоты UARTCLK. Значение частоты IrLPBaud16 задается путем записи соответствующего коэффициента деления частоты в регистр UARTILPR.

Выход кодера имеет активное низкое состояние. При передаче логической единицы выход кодера остается в низком состоянии, при передаче логического нуля – формируется импульс, при этом выход кратковременно переводится в высокое состояние.

Как в нормальном режиме, так и в режиме пониженного энергопотребления использование нецелых значений коэффициента деления скорости передачи данных увеличивает джиттер («дребезжание») фронтов импульсов данных. Наличие джиттера в случае использования дробных коэффициентов деления связано с тем, что интервалы между тактовыми импульсами Baud16 будут нерегулярными – период сигнала Baud16 в разное время будет содержать различное количество периодов сигнала UARTCLK. Можно показать, что в наихудшем случае величина джиттера в потоке ИК импульсов может достигать трех периодов UARTCLK. В соответствии со спецификацией стандарта IrDA SIR, джиттер не должен превышать величины 13%. В случае, если частота сигнала UARTCLK составляет более 3.6834 МГц, а скорость передачи данных меньше или равна 115200 бит/с, величина джиттера не превышает 9%. Таким образом, требования стандарта выполняются.

### Декодер ИК приемника

Декодер преобразует поток данных, сформированных по закону возврата к нулю, полученного от приемника ИК сигнала, и выдает поток данных без возврата к нулю на вход приемника UART. В неактивном состоянии вход декодера находится нормально в высоком состоянии. Выходной сигнал кодера имеет полярность, противоположную полярности входа декодера.

Обнаружение стартового бита осуществляется при низком уровне сигнала на входе декодера.

#### *Примечание*

Для того, чтобы исключить ложные срабатывания UART от импульсных помех, на входе SIRIN игнорируются импульсы с длительностью менее, чем:

- 3/16 длительности Baud16 в режиме IrDA;
- 3/16 длительности IrLPBaud16 в режиме IrDA с пониженным энергопотреблением.

## **Описание работы UART**

### **Сброс модуля**

Приемопередатчик и кодек могут быть сброшены общим сигналом сброса процессора. Значения регистров после сброса описаны в разделе «Программное управление модулем».

### **Тактовые сигналы**

Частота тактового сигнала UARTCLK должна обеспечивать поддержку требуемого диапазона скоростей передачи данных:

$$F\_UARTCLK(\text{min}) \geq 16 * \text{baud\_rate\_max};$$
$$F\_UARTCLK(\text{max}) \leq 16 * 65535 * \text{baud\_rate\_min}.$$

Например, для поддержки скорости передачи данных в диапазоне от 110 до 460800 Бод частота UARTCLK должна находиться в интервале от 7.3728 МГц до 115.34 МГц.

Частота UARTCLK, кроме того, должна выбираться с учетом возможности установки скорости передачи данных в рамках заданных требований точности.

Также существует ограничение на соотношение между тактовыми частотами CPU\_CLK и UARTCLK. Частота UARTCLK должна быть не более, чем в 5/3 раз выше частоты CPU\_CLK.

$$F\_UARTCLK \leq 5/3 * F\_CPU\_CLK.$$

Например, при работе в режиме UART с максимальной скоростью передачи данных 921600 бод, при частоте UARTCLK 14.7456 МГц, частота CPU\_CLK должна быть не менее 8.85276 МГц. Это гарантирует, что контроллер UART будет иметь достаточно времени для записи принятых данных в буфер FIFO.

### **Работа универсального асинхронного приемопередатчика**

Управляющая информация хранится в регистре управления линией UARTLCR. Этот регистр имеет внутреннюю ширину 30 бит, однако внешний доступ по шине APB к нему осуществляется через следующие регистры:

- UARTLCR\_H – определяет:
  - параметры передачи данных;
  - длину слова;
  - режим буферизации;
  - количество передаваемых стоповых бит;
  - режим контроля четности;
  - формирование сигнала разрыва линии;
- UARTIBRD – определяет целую часть коэффициента деления для скорости передачи данных;
- UARTFBRD – определяет дробную часть коэффициента деления для скорости передачи данных.

## Коэффициент деления частоты

Коэффициент деления для формирования скорости передачи данных состоит из 22 бит, при этом 16 бит выделено для представления его целой части, а 6 бит – дробной части. Возможность задания нецелых коэффициентов деления позволяет осуществлять обмен данными со стандартными информационными скоростями, при этом используя в качестве UARTCLK тактовый сигнал с произвольной частотой более 3.6864 МГц.

Целая часть коэффициента деления записывается в 16-битный регистр UARTIBRD. Шестиразрядная дробная часть записывается в регистр UARTFBRD. Значение коэффициента деления связано с содержимым указанных регистров следующим образом:

Коэффициент деления =  $UARTCLK / (16 * \text{скорость передачи данных})$   
= BRD\_I + BRD\_F,  
где  
BRD\_I – целая часть, а BRD\_F – дробная часть коэффициента деления.

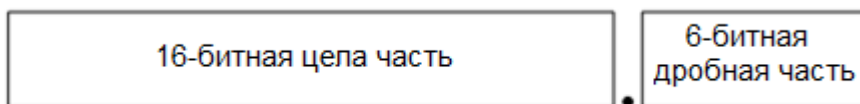


Рис. 81 Коэффициент деления

Шестибитное значение, записываемое в регистр UARTFBRD, вычисляется путем выделения дробной части требуемого коэффициента деления, умножения ее на 64 (то есть на  $2^n$ , где  $n$  – ширина регистра UARTFBRD) и округления до ближайшего целого числа:

$M = \text{integer}(BRD\_F * 2^n + 0.5)$ ,  
где

*integer* – операция отсечения дробной части числа,  $n = 6$ .

В модуле формируется внутренний сигнал Baud16, представляющий собой последовательность импульсов с длительностью, равной периоду сигнала UARTCLK и средней частотой, в 16 раз большей требуемой скорости обмена данными.

## Передача и прием данных

Принятые или передаваемые данные заносятся в 16-элементные буферы FIFO, при этом каждый элемент приемного буфера FIFO кроме байта данных хранит также четыре бита информации о состоянии модема.

Для передачи данные заносятся в буфер FIFO передатчика. Если работа приемопередатчика разрешена, начинается передача информационного кадра с параметрами, указанными в регистре управления линией UARTLCR\_H. Передача данных продолжается до опустошения буфера FIFO передатчика. После записи элемента в буфер FIFO передатчика сигнал BUSY переходит в высокое состояние. Это состояние сохраняется в течение всего времени передачи данных. В низкое состояние сигнал BUSY переходит только после того, как буфер FIFO передатчика станет пуст, а последний бит данных (включая стоповые биты) будет передан. Сигнал BUSY может находиться в высоком состоянии даже в случае, если приемопередатчик будет переведен из разрешенного состояний в запрещенное.

Для каждого бита данных (в приемной линии) производится три измерения уровня, решение принимается по мажоритарному принципу.

В случае, если приемник находился в неактивном состоянии (на линии входного сигнала UART\_RXD постоянно присутствовала единица) и произошел переход входного сигнала из высокого в низкий логический уровень (обнаружен стартовый бит), включается счетчик, тактируемый сигналом Baud16, после чего отсчеты сигнала на входе приемника регистрируются каждые восемь тактов (в режиме асинхронного приемопередатчика) или каждые четыре такта (в режиме ИК обмена данными) сигнала Baud16. Более частая выборка данных в режиме ИК обмена связана с необходимостью корректной обработки импульсов данных согласно протоколу SIR IrDA.

Стартовый бит считается достоверным в случае, если сигнал на линии UART\_RXD сохраняет низкий логический уровень в течение восьми отсчетов сигнала Baud16 с момента включения счетчика. В противном случае переход в ноль рассматривается как ложный старт и игнорируется.

В случае, если обнаружен достоверный стартовый бит, производится регистрация последовательности данных на входе приемника. Очередной бит данных фиксируются каждые 16 отсчетов тактового сигнала Baud16 (что соответствует длительности одного символа). Производится регистрация всех бит данных (согласно запрограммированным параметрам) и бита четности (если включен режим контроля четности).

Наконец, производится проверка присутствия корректного стопового бита (высокий логический уровень сигнала UART\_RXD). В случае, если последнее условие не выполняется, устанавливается признак ошибки формирования кадра. После того, как слово данных принято полностью, оно заносится в буфер FIFO приемника, наряду с четырьмя битами признаков ошибки, связанных с принятым словом (см. Таблица 142).

### **Биты ошибки**

Три бита признаков ошибки, ассоциированные с принятым символом данных, заносятся на позиции [10:8] слова данных в буфере FIFO приемника. Также предусмотрен признак ошибки переполнения буфера FIFO, расположенный на позиции 11 слова данных.

В Таблица 142 описано назначение всех битов слова данных в FIFO буфере приемника.

### **Бит переполнения буфера**

Бит переполнения непосредственно не связан с конкретным символом в буфере приемника. Признак переполнения фиксируется в случае, если буфер FIFO заполнен к моменту, когда очередной символ данных полностью принят (находится в регистре сдвига). При этом данные из регистра сдвига не попадают в буфер приемника и теряются с началом приема очередного символа. Как только в буфере приемника появляется свободное место, очередной принятый символ данных заносится в буфер FIFO вместе с текущим значением признака переполнения. После успешной записи данных в буфер признак переполнения сбрасывается.

Бит буфера FIFO	Назначение
11	Признак переполнения буфера
10	Ошибка – разрыв линии
09	Ошибка проверки на четность
08	Ошибка формирования кадра
07:00	Принятые данные

### **Запрет буфера FIFO**

Предусмотрена возможность отключения FIFO буферов приемника и передатчика. В этом случае приемная и передающая сторона контроллера UART располагают лишь однобайтными буферными регистрами. Бит переполнения буфера устанавливается при этом тогда, когда очередной символ данных уже принят, однако предыдущий еще не был считан.

В настоящей реализации модуля буферы FIFO физически не отключаются, необходимая функциональность достигается за счет логических манипуляций с флагами. При этом в случае, если буфер FIFO отключен, а сдвиговый регистр передатчика пуст (не используется), запись байта данных происходит непосредственно в регистр сдвига, минуя буферный регистр.

### **Проверка по шлейфу**

Проверка по шлейфу (замыкание выхода передатчика на вход приемника) выполняется путем установки в 1 бита LBE в регистре управления контроллером UARTCR.

### **Работа кодека ИК обмена данными IrDA SIR**

Кодек обеспечивает сопряжение асинхронного потока данных, сформированного приемопередатчиком, с полудуплексным последовательным интерфейсом IrDA SIR. Какая-либо аналоговая обработка сигнала при этом не выполняется. Назначение кодека – сформировать цифровой поток данных на вход приемника асинхронного сигнала и обработать цифровой поток данных с выхода передатчика.

Предусмотрено два режима работы:

В режиме IrDA уровень логического нуля передается на линию nSIROUT в виде импульса с высоким логическим уровнем и длительностью 3/16 от выбранного периода следования бит данных. Логическая единица при этом передается в виде постоянного низкого уровня сигнала. Сформированный выходной сигнал далее подается на передатчик ИК сигнала, обеспечивая излучение светового импульса всякий раз при передаче нулевого бита. На приемной стороне световые импульсы воздействуют на базу фототранзистора ИК приемника, который в результате формирует низкий логический уровень. Это, в свою очередь, обуславливает низкий уровень на входе SIRIN.

В режиме IrDA с пониженным энергопотреблением длительность передаваемых импульсов ИК излучения устанавливается в три раза выше длительности импульсов внутреннего опорного сигнала IrLPBaud16 (равной 1.63 мкс при номинальной частоте 1.8432 МГц). Данный режим активизируется путем установки бита SIRLP в регистре управления UARTCR.

Как в нормальном режиме, так и в режиме пониженного энергопотребления:

- кодирование осуществляется на основе бит данных, сформированных асинхронным передатчиком модуля;
- в ходе приема данных декодированные биты далее обрабатываются блоком асинхронного приема.

В соответствии со спецификацией физического уровня протокола IrDA SIR, обмен данными должен осуществляться в режиме полудуплекса, при этом задержка между передачей и приемом данных должна составлять не менее 10 мс. Эта задержка должна формироваться программно. Необходимость ее введения обусловлена тем, что воздействие передающего ИК светодиода на находящийся рядом ИК приемник может привести к искажению принимаемого сигнала или даже ввести приемный тракт в состояние насыщения. Задержка между окончанием передачи и началом приема данных именуется латентность, или время установки (готовности) приемника.

Сигнал IrLPBaud16 формируется путем деления частоты сигнала UARTCLK в соответствии с коэффициентом деления, записанным в регистре UARTILPR. Коэффициент деления вычисляется по формуле:

$$F\_UARTCLK / F\_IrLPBaud16,$$

где номинальное значение IrLPBaud16 составляет 1.8432 МГц. Коэффициент деления должен быть выбран так, чтобы выполнялось соотношение:

$$1.42 \text{ МГц} < F\_IrLPBaud16 < 2.12 \text{ МГц}.$$

### Проверка по шлейфу

Проверка по шлейфу выполняется после установки в 1 бита LBE регистра управления контроллером UARTCR с одновременной установкой в 1 бита SIRTEST регистра управления тестированием UARTTCCR.

В этом режиме данные, передаваемые на выход nSIROUT, должны подаваться на вход SIRIN.

### Примечание

Это единственный случай использования тестового регистра в нормальном режиме функционирования модуля.

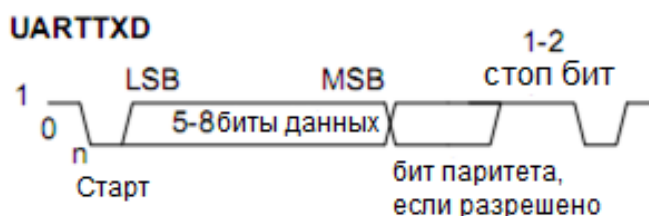


Рис. 82 Кадр передачи данных

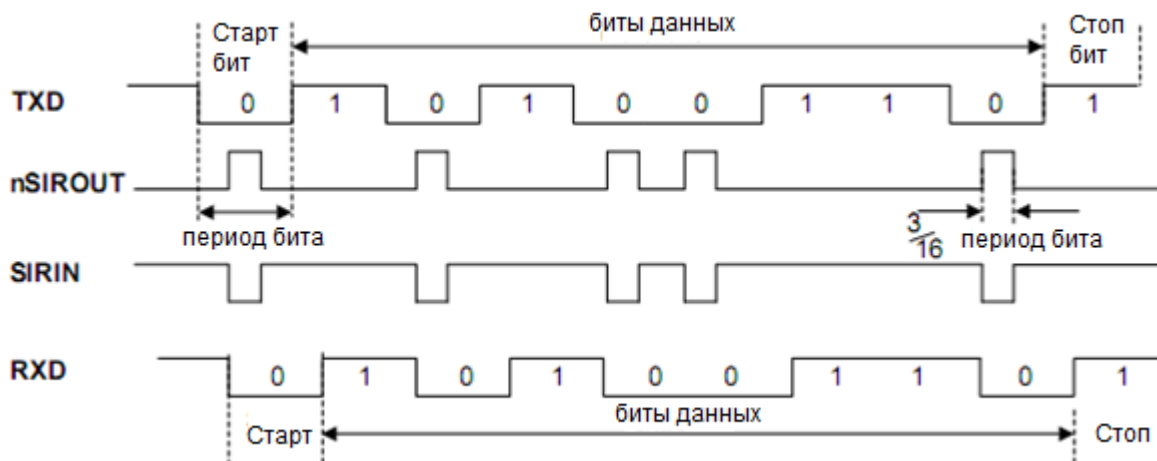


Рис. 83 Модуляция данных IrDA

### Линии управления модемом

Модуль универсального асинхронного приемопередатчика может использоваться как в режиме оконечного оборудования (DTE), так и в режиме оборудования передачи данных (DCE). На Рис.79 показаны сигналы модема в режиме DTE. Назначение сигналов в режиме DCE представлены в Таблица 143.

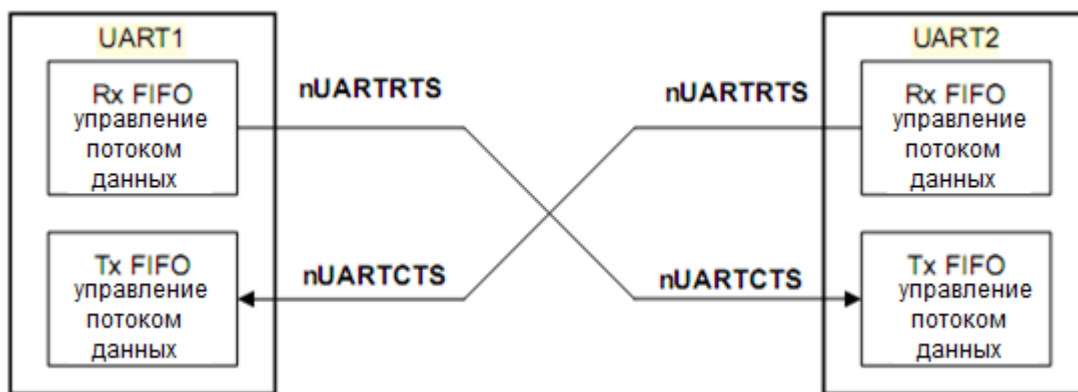
Таблица 143 Назначение управления модемом в режимах DTE и DCE

Сигнал	Назначение	
	Режим оконечного оборудования	Режим оборудования передачи данных
nUARTCTS	Готов к передаче данных	Запрос передачи данных
nUARTDSR	Источник данных готов	Приемник данных готов
nUARTDCD	Обнаружен информационный сигнал	-
nUARTRI	Индикатор вызова	-
nUARTCTS	Запрос передачи данных	Готов к передаче данных
nUARTDTR	Приемник данных готов	Источник данных готов
nUARTOUT1	-	Обнаружен информационный сигнал
nUARTOUT2	-	Индикатор вызова

### Аппаратное управление потоком данных

Программно активизируемый режим аппаратного управления потоком данных позволяет контролировать (приостанавливать и возобновлять) информационный обмен с помощью сигналов nUARTRTS и nUARTCTS. Иллюстрация взаимодействия двух устройств последовательной связи с аппаратным управлением потоком данных представлена на Рис. 84.





**Рис. 84** Взаимодействие двух устройств последовательной связи с аппаратным управлением потоком данных

Если разрешено управление потоком данных по сигналу RTS, линия nUARTRTS переводится в активное состояние только после того, как в FIFO буфере приема появляется заданное количество свободных элементов.

Если разрешено управление потоком данных по сигналу CTS, передача данных осуществляется только после перевода линии nUARTCTS в активное состояние.

Режим аппаратного управления потоком данных задается путем установки значений бит RTSEn и CTSEn в регистре управления UARTCR. В Таблица 144 показаны необходимые установки для различных режимов управления потоком данных.

**Таблица 144** Режимы управления потоком данных

CTSEn	RTSEn	Описание
1	1	Разрешено управление потоком данных по CTS и RTS
1	0	Управления потоком данных осуществляется по линии CTS
0	1	Управления потоком данных осуществляется по линии RTS
0	0	Управления потоком данных запрещено

Примечание. В случае, если выбран режим управления потоком данных по RTS, программное обеспечение не может использовать бит RTSEn регистра UARTCR для проверки состояния линии RTS.

### **Управление потоком данных по линии RTS**

Логика управления потоком данных по RTS использует данные о превышении пороговых уровней заполнения буфера FIFO приемника. В случае выбора режимов с управлением по RTS, сигнал на линии nUARTRTS переводится в активное состояние только после того, как в FIFO буфере приема появляется заданное количество свободных элементов. После достижения порогового уровня заполнения буфера приемника сигнал nUARTRTS снимается (переводится в пассивное состояние), указывая таким образом на отсутствие свободного места для сохранения принятых данных. При этом дальнейшая передача данных должна быть прекращена по завершении передачи текущего символа.

Обратно в активное состояние сигнал nUARTRTS переводится после считывания данных из приемного буфера FIFO в количестве, достаточном для того, чтобы заполнение буфера оказалось ниже порогового уровня.

В случае, если управление потоком данных по RTS запрещено, однако работа приемопередатчика UART разрешена, прием будет осуществляться до полного заполнения буфера FIFO, либо до завершения передачи данных.

### **Управление потоком данных по линии CTS**

В случае выбора одного из режимов с управлением потоком данных по CTS передатчик осуществляет проверку состояния линии nUARTCTS перед началом передачи очередного байта данных. Передача осуществляется только в случае, если данная линия активна, и продолжается до тех пор, пока активное состояние линии сохраняется и буфер передатчика не пуст.

При переходе линии nUARTCTS в неактивное состояние модуль завершает выдачу текущего передаваемого символа, после чего передача данных прекращается.

Если управление потоком данных по CTS запрещено, и при этом работа приемопередатчика UART разрешена - данные будут выдаваться до опустошения буфера FIFO передатчика.

### **Интерфейс прямого доступа к памяти**

Модуль универсального асинхронного приемопередатчика оснащен интерфейсом подключения к контроллеру прямого доступа к памяти. Работа в данном режиме контролируется регистром управления DMA UARTDMACR.

Интерфейс DMA включает в себя следующие сигналы:

#### **Для приема:**

UARTRXDMASREQ – запрос передачи отдельного символа, инициируется контроллером UART. Размер символа в режиме приема данных – до 12 бит. Сигнал переводится в активное состояние в случае, если буфер FIFO приемника содержит по меньшей мере один символ.

UARTRXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переходит в активное состояние в случае, если заполнение буфера FIFO приемника превысило заданный порог. Порог программируется индивидуально для каждого буфера FIFO путем записи значения в регистр UARTIFLS.

UARTRXDMACLR – сброс запроса на DMA, инициируется модулем приемопередатчика с целью сброса принятого запроса. В случае, если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

#### **Для передачи:**

UARTTXDMASREQ – запрос передачи отдельного символа, инициируется модулем приемопередатчика. Размер символа в режиме передачи данных – до восьми бит. Сигнал переводится в активное состояние в случае, если буфер FIFO передатчика содержит, по меньшей мере, одну свободную ячейку.

UARTTXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переводится в активное состояние в случае, если заполнение буфера FIFO передатчика ниже заданного порога. Порог программируется индивидуально для каждого буфера FIFO путем записи значения в регистр UARTIFLS.

UARTTXDMACLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае, если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

Сигналы блочного и одноэлементного обмена данными не являются взаимно исключаящими, они могут быть инициированы одновременно. Например, в случае, если заполнение данными буфера приемника превышает пороговое значение, формируется как сигнал запроса одноэлементного обмена, так и сигнал запроса блочного обмена данными. В случае, если количество данных в буфере приема меньше порогового значения формируется только запрос одноэлементного обмена. Это бывает полезно в ситуациях, при которых объем данных меньше размера блока. Пусть, например, нужно принять 19 символов, а порог заполнения буфера FIFO установлен равным четырем. Тогда контроллер DMA осуществит четыре передачи блоков по четыре символа, а оставшиеся три символа передаст в ходе трех одноэлементных обменов.

**Примечание**

Для оставшихся трех символов контроллер UART не может инициировать процедуру блочного обмена.

Каждый инициированный приемопередатчиком сигнал запроса DMA остается активным до момента его сброса соответствующим сигналом DMACLR. После снятия сигнала сброса модуль приемопередатчика вновь получает возможность сформировать запрос на DMA в случае выполнения описанных выше условий. Все запросы DMA снимаются после запрета работы приемопередатчика, а также в случае установки в ноль бита управления DMA TXDMAE или RXDMAE в регистре управления DMA UARTDMACR.

В случае запрета буферов FIFO устройство способно передавать и принимать только одиночные символы; как следствие, контроллер может инициировать DMA только в одноэлементном режиме. При этом модуль в состоянии формировать только сигналы управления DMA UARTRXDMASREQ и UARTTXDMASREQ. Для информации о запрете буферов FIFO см. описание регистра управления линией UARTLCR\_H.

Когда буферы FIFO включены, обмен данными может производиться в ходе как одноэлементных, так и блочных передач данных, в зависимости от установленной величины порога заполнения буферов и их фактического заполнения. Таблица 145 показывает значения параметров срабатывания запросов блочного обмена UARTRXDMABREQ и UARTTXDMABREQ в зависимости от порога заполнения буфера.

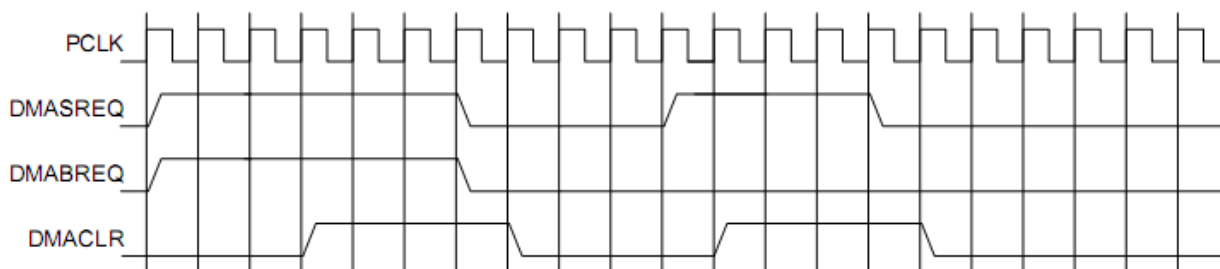
**Таблица 145** Параметры срабатывания запросов блочного обмена данными в режиме ПДП

Пороговый уровень	Длина блока обмена данными	
	Буфер передатчика (количество незаполненных ячеек)	Буфер приемника (количество заполненных ячеек)
1/8	14	2
1/4	12	4
1/2	8	8
3/4	4	12

7/8	2	14
-----	---	----

В регистре управления DMA UARTDMACR предусмотрен бит DMAONERR, который позволяет запретить DMA от приемника в случае активного состояния линии прерывания по обнаружению ошибки UARTEINTR. При этом соответствующие линии запроса DMA – UARTRXDMASREQ и UARTRXDMABREQ переводятся в неактивное состояние (маскируются) до сброса UARTEINTR. На линии запроса DMA, обслуживающие передатчик, состояние UARTEINTR не влияет.

На Рис. 85 показаны временные диаграммы одноэлементного и блочного запросов DMA, в том числе действие сигнала DMACLR. Все сигналы должны быть синхронизированы с CPU\_CLK. В интересах ясности изложения предполагается, что синхронизация сигналов запроса DMA в контроллере DMA не производится.



**Рис. 85** Временные диаграммы одноэлементного и блочного запросов DMA

### **Прерывания**

В модуле предусмотрено 11 маскируемых источников прерывания. В результате формируется один общий сигнал, представляющий собой комбинацию независимых сигналов, объединенных по схеме ИЛИ.

Сигналы запроса на прерывание:

UARTRXINTR – прерывание от приемника.

UARTTXINTR – прерывание от передатчика.

UARTRTINTR – прерывание по таймауту приемника.

UARTMSINTR – прерывание по состоянию модема:

UARTRIINTR, изменение состояния линии nUARTRI;

UARTCTSINTR, изменение состояния линии nUARTCTS;

UARTDCDINTR, изменение состояния линии nUARTDCD;

UARTDSRINTR, изменение состояния линии nUARTDSR.

UARTEINTR – ошибка:

UARTOEINTR, переполнение буфера;

UARTBEINTR, прерывание приема – разрыв линии;

UARTPEINTR, ошибка контроля четности;

UARTFEINTR, ошибка в структуре кадра.

UARTINTR – логическое ИЛИ сигналов UARTRXINTR, UARTTXINTR, UARTRTINTR, UARTMSINTR и UARTEINTR.

Каждый из независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре маски UARTIMSC. Установка бита в 1 разрешает соответствующее прерывание, в 0 – запрещает.

Доступность, как индивидуальных линий, так и общей линии запроса позволяет организовать обслуживание прерываний в системе, как путем применения глобальной процедуры обработки, так и с помощью драйвера устройства, построенного по модульному принципу.

Прерывания от приемника и передатчика UARTRXINTR и UARTTXINTR выведены отдельно от прерываний по изменению состояния устройства. Это позволяет использовать сигналы запроса UARTRXINTR и UARTTXINTR для обеспечения чтения и записи данных согласованно с достижением заданного порога заполнения буферов FIFO приемника и передатчика.

Прерывание по обнаружению ошибке UARTEINTR формируется в случае возникновения той или иной ошибки приема данных. Предусмотрен ряд условий формирования признака ошибки.

Прерывание по состоянию модема представляет собой комбинацию признаков изменения отдельных линий состояния модема.

Признаки возникновения каждого из условий прерывания можно считать либо из регистра прерываний UARTRIS, либо из маскированного регистра прерываний UARTMIS.

### UARTMSINTR

Прерывание по состоянию модема возникает в случае изменения любой из линий состояний модема (nUARTCTS, nUARTDCD, nUARTDSR, nUARTRI). Сброс прерывания осуществляется путем записи 1 в соответствующий (в зависимости от линии состояния модема, вызвавшей прерывание) разряд регистра сброса прерывания UARTICR.

### UARTRXINTR

Состояние прерывания от приемника может измениться в случае возникновения одного из следующих событий:

- буфер FIFO разрешен и его заполнение достигло заданного порогового значения. В этом случае линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после чтения данных из буфера приемника до тех пор, пока его заполнение не станет меньше порога, либо после сброса прерывания;
- буфер FIFO запрещен (имеет размер один символ), принят один символ данных. При этом линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после чтения одного байта данных, либо после сброса прерывания.

### UARTTXINTR

Состояние прерывания от передатчика может измениться в случае возникновения одного из следующих событий:

- буфер FIFO разрешен и его заполнение меньше или равно заданному пороговому значению. В этом случае линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после записи данных в буфера передатчика до тех пор, пока его заполнение не станет больше порога, либо после сброса прерывания;
- буфер FIFO запрещен (имеет размер один символ), данные в буферном регистре передатчика отсутствуют. При этом линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после записи одного байта данных, либо после сброса прерывания.

Для занесения данных в буфер FIFO передатчика необходимо записать данные в буфер либо перед разрешением работы приемопередатчика и прерываний, либо после разрешения работы приемопередатчика и прерываний.

### *Примечание*

Прерывание передатчика работает по фронту, а не по уровню сигнала. В случае, если модуль и прерывания от него разрешены до осуществления записи данных в буфер FIFO передатчика, прерывание не формируется. Прерывание возникает только при опустошении буфера FIFO.

## **UARTRTINTR**

Прерывание по таймауту приемника возникает в случае, если буфер FIFO приемника не пуст, и на вход приемника не поступало новых данных в течение периода времени, необходимого для передачи 32 бит. Прерывание по таймауту снимается либо после считывания данных из буфера приемника до его опустошения (или считывания одного байта в случае, если буфер FIFO запрещен), либо путем записи 1 в соответствующий бит регистра сброса прерывания UARTICR.

## **UARTEINTR**

Прерывание по обнаружению ошибки возникает в случае ошибки при приеме данных. Оно может быть вызвано рядом факторов:

- ошибка в структуре кадра;
- ошибка контроля четности;
- разрыв линии;
- переполнение буфера.

Причину возникновения прерывания можно определить, прочитав содержимое регистра прерываний UARTRIS, либо содержимое маскированного регистра прерываний UARTMIS.

Сброс прерывания осуществляется путем записи соответствующих бит в регистр сброса прерывания UARTICR. За прерываниями по обнаружению ошибки закреплены биты с 7 по 10.

## **UARTINTR**

Все описанные сигналы запроса на прерывание скомбинированы в общую линию путем объединения по схеме ИЛИ сигналов UARTRXINTR, UARTRXINTR, UARTRTINTR, UARTMSINTR и UARTEINTR с учетом маскирования. Общий выход может быть подключен к системному контроллеру прерывания, что позволит ввести дополнительное маскирование запросов на уровне периферийных устройств.

**Программное управление модулем**

**Общая информация**

Следующая информация применима ко всем регистрам контроллера:

- Базовый адрес контроллера фиксирован. Смещение каждого регистра относительно базового адреса постоянно.
- Не следует пытаться получить доступ к зарезервированным или неиспользуемым адресам. Это может привести к непредсказуемому поведению модуля.
- За исключением специально оговоренных в настоящем документе случаев:
  - не следует изменять значения не определенных в документе разрядов регистров;
  - не следует использовать значения не определенных в документе разрядов регистров;
  - все биты регистров (за исключением специально оговоренных случаев) устанавливаются в значение 0 после сброса по включению питания или системного сброса.
- Столбец «Тип» в Таблица 146 определяет режим доступа к регистру в соответствии с обозначениями:
  - RW – чтение и запись;
  - RO – только чтение;
  - WO – только запись.

**Обобщенные данные о регистрах устройства**

Данные о регистрах модуля универсального асинхронного приемопередатчика приведены в Таблица 146.

**Таблица 146** Обобщенные данные о регистрах устройства

Смещение	Наименование	Тип	Значение после сброса	Размер, бит	Описание
0x000	DR	RW	0x---	12/8	Регистр данных
0x004	RSR_ECR	RW	0x0	4/0	Регистра состояния приемника / Сброс ошибки приемника
0x008-0x014					Резерв
0x018	FR	RO	0b-10010-- -	9	Регистр флагов
0x01C					Резерв
0x020	ILPR	RW	0x00	8	Регистр управления ИК обменом в режиме пониженного энергопотребления
0x024	IBRD	RW	0x0000	16	Целая часть делителя скорости обмена данными
0x028	FBRD	RW	0x00	6	Дробная часть делителя

					скорости обмена данными
0x02C	LCR_H	RW	0x00	8	Регистр управления линией
0x030	CR	RW	0x0300	16	Регистр управления
0x034	IFLS	RW	0x12	6	Регистр порога прерывания по заполнению буфера FIFO
0x038	IMSC	RW	0x000	11	Регистр маски прерывания
0x03C	RIS	RO	0x00-	11	Регистр состояния прерываний
0x040	MIS	RO	0x00-	11	Регистр состояния прерываний с маскированием
0x044	ICR	WO	-	11	Регистр сброса прерывания
0x048	DMACR	RW	0x00	3	Регистр управления ПДП
0x080	UARTTCR	RW	0x00	3	Регистр управления тестированием

### DR. Регистр данных

В ходе передаче данных:

Если буфер FIFO передатчика разрешен, то слово данных, записанные в рассматриваемый регистр, направляется в буфер FIFO передатчика.

В противном случае, записанное слово фиксируется в буферный регистр передатчика (последний элемент буфера FIFO).

Операция записи в регистр инициирует передачу данных. Слово данных предваряется стартовым битом, дополняется битом контроля четности (если режим контроля четности включен) и стоповым битом. Сформированное слово отправляется в линию передачи данных.

В ходе приема данных:

Если буфер FIFO приемника разрешен, байт данных и четыре бита состояния (разрыв, ошибка формирования кадра, четность, переполнение) сохраняются в 12-битном буфере.

В противном случае байт данных и биты состояния записываются в буферный регистр (последний элемент буфера FIFO).

Полученные из линии связи байты данных считывается путем чтения из регистра UARTDR принятых данных совместно с соответствующими битами состояния. Информация о состоянии также может быть получена путем чтения регистра RSR\_ECR (см. Таблица 148).

**Таблица 147** Формат регистра DR

Бит	Наименование	Назначение
15:12		Резерв
11	OE	Переполнение буфера приемника. Бит устанавливается в 1 в случае, если на вход приемника поступают данные, в то время, как буфер заполнен. Сбрасывается в 0 после того, как в буфере появится свободное место.
10	BE	Разрыв линии. Устанавливается в 1 при обнаружении признака разрыва линии, то есть в случае наличия низкого логического уровня на входе приемника в течение времени, большего чем длительность



		передачи полного слова данных (включая стартовый, стоповый биты и бит проверки на четность). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер. В случае обнаружения разрыва линии в буфер загружается только один нулевой символ, прием данных возобновляется только после перехода линии в логическую 1 и последующего обнаружения корректного стартового бита.
9	PE	Ошибка контроля четности. Устанавливается в 1 в случае, если четность принятого символа данных не соответствует установкам битов EPS и SPS в регистре управления линией UARTLCR_H. При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер.
8	FE	Ошибка в структуре кадра. Устанавливается в 1 в случае, если в принятом символе не обнаружен корректный стоповый бит (корректный стоповый бит равен 1). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер.
7:0	DATA	Принимаемые данные (чтение) Передаваемые данные (запись)

Примечание. Необходимо запрещать работу приемопередатчика перед любым перепрограммированием его регистров управления. Если приемопередатчик переводится в отключенное состояние во время передачи или приема символа, то перед остановкой он завершает выполняемую операцию.

### **RSR\_ECR. Регистр состояния приемника / сброса ошибки**

Состояние приемника также может быть считано из регистра RSR. В этом случае информация о состоянии признаков разрыва линии, ошибки контроля четности и ошибки в структуре кадра относится к последнему символу, считанному из регистра данных DR. С другой стороны, признак переполнения буфера устанавливается немедленно после возникновения этого состояния (и не связан с последним считанным из регистра DR байтом данных).

Запись в регистр ECR приводит к сбросу признаков ошибок переполнения, четности, структуры кадра, разрыва линии. Кроме того, все эти признаки устанавливаются в 0 после сброса устройства.

В Таблица 148 представлено назначение бит регистра RSR\_ECR.

**Таблица 148** Регистр RSR\_ECR

Биты	Наименование	Назначение
7:0		Запись в регистр сбрасывает признаки ошибок формирования кадра, проверки на четность, разрыва линии и переполнения буфера.
7:4		Резерв, при чтении результат не определен
3	OE	Переполнение буфера приемника. Бит

		устанавливается в 1 в случае, если на вход приемника поступают данные, в то время, как буфер заполнен. Сбрасывается в 0 после записи в регистр UARTECR. Содержимое буфера остается верным, так как перезаписан был только регистр сдвига. Центральный процессор должен считать данные для того, чтобы освободить буфер FIFO.
2	BE	Разрыв линии. Устанавливается в 1 при обнаружении признака разрыва линии, то есть в случае наличия низкого логического уровня на входе приемника в течение времени, большего чем длительность передачи полного слова данных (включая стартовый, стоповый биты и бит проверки на четность). Бит сбрасывается в 0 после записи в регистр UARTECR. При включенном FIFO данная ошибка ассоциируется с символом, находящемся на вершине буфера. В случае обнаружения разрыва линии в буфер загружается только один нулевой символ, прием данных возобновляется только после перехода линии в логическую 1 и последующего обнаружения корректного стартового бита.
1	PE	Ошибка контроля четности. Устанавливается в 1 в случае, если четность принятого символа данных не соответствует установкам битов EPS и SPS в регистре управления линией LCR_H. Бит сбрасывается в 0 после записи в регистр ECR. При включенном FIFO данная ошибка ассоциируется с символом, находящимся на вершине буфера.
0	FE	Ошибка в структуре кадра. Устанавливается в 1 в случае, если в принятом символе не обнаружен корректный стоповый бит (корректный стоповый бит равен 1). Бит сбрасывается в 0 после записи в регистр ECR. При включенном FIFO данная ошибка ассоциируется с символом, находящимся на вершине буфера.

Примечание. Перед чтением регистра состояния RSR необходимо считать данные, принятые из линии, путем обращения к регистру данных DR. Противоположная последовательность действий не допускается, так как регистр RSR обновляет свое состояние только после чтения регистра DR. Вместе с тем, информация о состоянии приемника может быть получена непосредственно из регистра данных DR.

### **FR. Регистр флагов**

После сброса биты регистра флагов TXFF, RXFF и BUSY устанавливаются в 0, а биты TXFE и RXFE – в 1. В Таблица 149 представлена информация о назначении битов регистра.

**Таблица 149** Регистр FR

Биты	Наименование	Назначение
15:9		Резерв. Не модифицируйте. При чтении заполняются нулями.
8	RI	Инверсия линии nUARTRI
7	TXFE	Буфер FIFO передатчика пуст. Значение бита зависит от состояния бита FEN регистра управления линией LCR_H. Если буфер FIFO запрещен, бит устанавливается в 1 когда буферный регистр передатчика пуст. В противном случае он равен 1 если пуст буфер FIFO передатчика. Данный бит не дает никакой информации о наличии данных в регистре сдвига передатчика.
6	RXFF	Буфер FIFO приемника заполнен. Значение бита зависит от состояния бита FEN регистра управления линией LCR_H. Если буфер FIFO запрещен, бит устанавливается в 1 когда буферный регистр приемника занят. В противном случае он равен 1 если заполнен буфер FIFO приемника.
5	TXFF	Буфер FIFO передатчика заполнен. Значение бита зависит от состояния бита FEN регистра управления линией LCR_H. Если буфер FIFO запрещен, бит равен 1 когда буферный регистр передатчика занят. В противном случае он равен 1 если заполнен буфер FIFO передатчика.
4	RXFE	Буфер FIFO приемника пуст. Значение бита зависит от состояния бита FEN регистра управления линией LCR_H. Если буфер FIFO запрещен, бит устанавливается в 1 когда буферный регистр приемника пуст. В противном случае он равен 1 если пуст буфер FIFO приемника.
3	BUSY	UART занят. Бит равен 1 в случае, если контроллер передает в линию данные. Бит остается установленным до тех пор, пока данные, включая стоповые биты, не будут полностью переданы. Кроме того, бит занятости устанавливается в 1 при наличии данных в буфере FIFO передатчика, вне зависимости от состояния приемопередатчика (даже если он запрещен).
2	DCD	Инверсия линии nUARTDCD.
1	DSR	Инверсия линии nUARTDSR.
0	CTS	Инверсия линии nUARTCTS.

### **ILPR. Регистр управления ИК обменом в режиме пониженного энергопотребления**

Этот восьмиразрядный регистр, доступный для чтения и записи, содержит значение коэффициента деления частоты UARTCLK, для формирования тактового сигнала IrLPBaud16. Назначение разрядов регистра показано в Таблица 150.

Требуемое значение коэффициента деления для формирования сигнала IrLPBaud16 вычисляется по формуле:  $ILPDVSR = F\_UARTCLK / F\_IrLPBaud16$ , где номинальное значение частоты  $F\_IrLPBaud16$  составляет 1.8432 МГц.

Коэффициент деления должен быть установлен таким образом, чтобы выполнялось соотношение:  $1.42 \text{ МГц} < F\_IrLPBaud16 < 2.12 \text{ МГц}$ , что, в свою очередь, гарантирует формирование кодеком импульсов данных с длительностью 1.41-2.11мкс (в три раза длиннее периода сигнала IrLPBaud16).

**Таблица 150** Регистр ILPR

Биты	Наименование	Назначение
7:0	ILPDVSR	Коэффициент деления частоты UARTCLK, для формирования тактового сигнала IrLPBaud16. После сброса устанавливается в 0. Примечание. Коэффициент 0 – запрещенное значение. В случае его установки импульсы IrLPBaud16 формироваться не будут.

Примечание. В интересах подавления помех, при работе в режиме IrDA с пониженным энергопотреблением кодек игнорирует поступающие на вход SIRIN импульсы с длительностью, меньшей трех периодов сигнала IrLPBaud16.

### **IBRD. Регистр целой части делителя скорости передачи данных**

Назначение бит регистра представлено в Таблица 151.

**Таблица 151** Регистр IBRD

Биты	Наименование	Назначение
15:0	BAUDDIV_INT	Целая часть коэффициента деления частоты для формирования тактового сигнала передачи данных. После сброса устанавливается в 0.

### **FBRD. Регистр дробной части делителя скорости передачи данных**

Назначение бит регистра представлено в Таблица 152.

**Таблица 152** Регистр BFRD

Биты	Наименование	Назначение
5:0	BAUDDIV_FRAC	Дробная часть коэффициента деления частоты для формирования тактового сигнала передачи данных. После сброса устанавливается в 0.

Коэффициент деления вычисляется по формуле:

$$BAUDDIV = FUARTCLK / (16 * \text{Baud\_rate}),$$

где FUARTCLK – тактовая частота контроллера UART, Baud\_rate – требуемая скорость передачи данных.

Коэффициент BAUDDIV состоит из целой и дробной частей – BAUDDIV\_INT и BAUDDIV\_FRAC, соответственно.

Примечания.

Изменение содержимого регистров IBRD и FBRD вступают в силу только после завершения передачи и приема текущего символа данных.

Минимальный допустимый коэффициент деления – 1, максимальный 65535 ( $2^{16} - 1$ ). Таким образом, значение IBRD, равное 0 является недопустимым, при этом значение регистра FBRD игнорируется.

Аналогично, при IBRD равном 65535 (0xFFFF), значение FBRD не может быть больше нуля. Невыполнение этого условия приведет к прерыванию приема или передачи.

Далее приведен пример вычисления коэффициента деления.

Пример. Вычисление коэффициента деления.

Пусть требуемая скорость передачи данных составляет 230400 бит/с, частота тактового сигнала UARTCLK равна 4 МГц. Тогда:

Коэффициент деления =  $(4 \cdot 10^6) / (16 \cdot 230400) = 1.085$ .

Таким образом, BRDI = 1, BRDF = 0.085.

Следовательно, значение, записываемое в регистр BFRD, равно  $m = \text{integer}((0.085 \cdot 64) + 0.5) = 5$ .

Реальное значение коэффициента деления =  $1 + 5/64 = 1.078$ .

Реальная скорость передачи данных =  $(4 \cdot 10^6) / (16 \cdot 1.078) = 231911$  бит/с.

Ошибка установки скорости =  $(231911 - 230400) / 230400 \cdot 100\% = 0.656\%$ .

Максимальная ошибка установки скорости передачи данных с использованием шестиразрядного регистра BFRD =  $1/64 \cdot 100\% = 1.56\%$ . Такая ошибка возникает в случае  $m = 1$ , при этом разница накапливается в течение 64 тактовых интервалов.

В Таблица 153 представлены значения коэффициента деления для типичных скоростей передачи данных при частоте UARTCLK = 7.3728 МГц. При таких параметрах дробная часть коэффициента деления не используется, следовательно, в регистр FBRD должен быть записан ноль.

**Таблица 153** Коэффициенты деления для типичных скоростей передачи данных при частоте UARTCLK = 7.3728 МГц

Коэффициент деления	Скорость передачи данных
0x0001	460800
0x0002	230400
0x0004	115200
0x0006	76800
0x0008	57600
0x000C	38400
0x0018	19200
0x0020	14400
0x0030	9600
0x00C0	2400
0x0180	1200
0x105D	110

В Таблица 154 приведены значения коэффициента деления для типичных скоростей передачи данных при частоте UARTCLK = 4 МГц.

**Таблица 154** Коэффициенты деления для типичных скоростей передачи данных при частоте UARTCLK = 4 МГц

Целая часть	Дробная часть	Требуемая скорость	Реальная скорость	Ошибка, %
0x001	0x05	230400	231911	0.656
0x002	0x0B	115200	115101	0.086
0x003	0x10	76800	76923	0.160
0x006	0x21	38400	38369	0.081
0x011	0x17	14400	14401	0.007
0x068	0x0B	2400	2400	~0
0x8E0	0x2F	110	110	~0

### LCR\_H. Регистр управления линией

Данный регистр обеспечивает доступ к разрядам с 29 по 22 регистра LCR. При сбросе все биты регистра LCR\_H обнуляются.

Назначение разрядов регистра описано в Таблица 155.

**Таблица 155** Регистр LCR\_H

Биты	Наименование	Назначение
15:8		Резерв. Не модифицируйте. При чтении выдаются нули.
7	SPS	Передача бита четности с фиксированным значением. 0 – запрещена; 1 – на месте бита четности передается инверсное значение бита EPS, оно же проверяется при приеме данных. (При EPS=0 на месте бита четности передается 1, при EPS=1 – передается 0). Значение бита SPS не играет роли в случае, если битом PEN формирование и проверка бита четности запрещено (см.Таблица 141).
6:5	WLEN	Длина слова – количество передаваемых или принимаемых информационных бит в кадре: 0b11 – 8 бит, 0b10 – 7 бит, 0b01 – 6 бит, 0b00 – 5 бит.
4	FEN	Разрешение работы буфера FIFO приемника и передатчика. 0 – запрещено, 1 – разрешено.
3	STP2	Режим передачи двух стоповых бит. 0 – один стоповый бит, 1 – два стоповых бита. Приемник не проверяет наличие дополнительного стопового бита в кадре.
2	EPS	Четность/нечетность. 0 – бит четности дополняет количество единиц в информационной части кадра до

		нечетного, 1 – до четного числа. Значение бита EPS не играет роли в случае, если битом PEN формирование и проверка бита четности запрещено (см.Таблица 141).
1	PEN	Разрешение проверки четности. 0 – кадр не содержит бита четности, 1 – бит четности передается в кадре и проверяется при приеме данных (см.Таблица 141).
0	BRK	Разрыв линии. Если этот бит установлен в 1, то по завершении передачи текущего символа на выходе UARTTXD устанавливается низкий уровень сигнала. Для правильного выполнения этой операции программное обеспечение должно обеспечить передачу сигнала разрыва в течение, как минимум, времени передачи двух информационных кадров. В нормальном режиме функционирования бит должен быть установлен в 0.

Содержимое регистров LCR\_H, IBRD и FBRD совместно образует общий 30-разрядный регистр LCR, который обновляется по стробу, формируемому при записи в LCR\_H. Таким образом, для того, чтобы изменение параметров коэффициента деления частоты обмена данными вступило в силу, после их изменения значения регистров IBRD и/или FBRD необходимо осуществить запись данных в регистр LCR\_H.

Примечания.

Изменение значений трех регистров можно осуществить корректно двумя способами:

- Запись IBRD, запись FBRD, запись LCR\_H;
- Запись FBRD, запись IBRD, запись LCR\_H;

Для того, чтобы изменить значение лишь одного из регистров (IBRD или FBRD) необходимо выполнить следующие шаги:  
Запись IBRD (или FBRD), запись LCR\_H.

В Таблица 156 приведена таблица истинности для бит управления контролем четности SPS, EPS, PEN регистра управления линией LCR\_H.

**Таблица 156** Управление режимом контроля четности

PEN	EPS	SPS	Бит контроля четности
0	X	X	Не передается, не проверяется
1	1	0	Проверка четности слова данных
1	0	0	Проверка нечетности слова данных
1	0	1	Бит четности постоянно равен 1
1	1	1	Бит четности постоянно равен 0

Примечания:

Регистры LCR\_H, IBRD, and FBRD не должны изменяться:

- При разрешенной работе приемопередатчика;
- Во время завершения приема или передачи данных в процессе остановки (перевода в запрещенное состояние) приемопередатчика.

Целостность данных в буферах FIFO не гарантируется в следующих случаях:

- После установки бита разрыва линии BRK;
- Если программное обеспечение произвело остановку приемопередатчика при наличии данных в буферах FIFO, после его повторного перевода в разрешенное состояние.

## CR. Регистр управления

После сброса все биты регистра управления, за исключением битов 9 и 8 устанавливаются в нулевое состояние. Биты 9 и 8 устанавливаются в единичное состояние.

Назначение разрядов регистра управления показано в Таблица 157.

**Таблица 157** Регистр управления CR

Биты	Наименование	Назначение
15	CTSEn	Разрешение управления потоком данных по CTS. 1 – разрешено, данные передаются в линию только при активном значении сигнала nUARTCTS.
14	RTSEn	Разрешение управления потоком данных по RTS. 1 – разрешено, запрос данных от внешнего устройства осуществляется только при наличии свободного места в буфере FIFO приемника.
13	Out2	Инверсия сигнала на линии состояния модема nUARTOut2. В режиме оконечного оборудования (DTE) эта линия может использоваться в качестве линии «сигнал вызова» (RI).
12	Out1	Инверсия сигнала на линии состояния модема nUARTOut1. В режиме оконечного оборудования (DTE) эта линия может использоваться в качестве линии «обнаружен информационный сигнал» (DCD).
11	RTS	Инверсия сигнала на линии состояния модема nUARTRTS.
10	DTR	Инверсия сигнала на линии состояния модема nUARTDTR.
9	RXE	Прием разрешен. Установка бита в 1 разрешает работу приемника. Прием данных осуществляется либо по интерфейсу асинхронного последовательного обмена, либо по интерфейсу ИК обмена SIR, в зависимости от значения бита SIREN. В случае перевода приемопередатчика в запрещенное состояние в ходе приема данных, он завершает прием текущего символа перед остановкой.
8	TXE	Передача разрешена. Установка бита в 1 разрешает работу передатчика. Передача осуществляется либо по интерфейсу асинхронного последовательного обмена, либо по интерфейсу ИК обмена SIR, в



		зависимости от значения бита SIREN. В случае перевода приемопередатчик в запрещенное состояние в ходе передачи данных, он завершает передачу текущего символа перед остановкой.
7	LBE	1 – шлейф разрешен, 0 – запрещен. В режиме разрешенного шлейфа: Если установлены бит SIREN=1 и бит регистра управления тестированием TCR SIRTEST=1, то сигнал с выхода модема nSIROUT инвертируется и подается на вход модема SIRIN. Бит SIRTEST устанавливается в 1 для того, чтобы вывести устройство из полудуплексного режима, характерного для интерфейса SIR. После окончания тестирования по шлейфу бит SIRTEST должен быть установлен в 0. Если бит SIRTEST=0, то выходная линия передатчика UARTTXD коммутируется на вход приемника UARTRXD. Как в режиме SIR, так и в режиме UART, выходные линии состояния модема коммутируются на соответствующие входные линии. После сброса бит устанавливается в 0.
6:3		Резерв. Не модифицируйте. При чтении выдаются нули.
2	SIRLP	Выбор режима ИК обмена с пониженным энергопотреблением: 0 – длительность импульсов данных равна 3/16 длительности передачи бита. 1 – длительность импульсов данных равна трем тактам сигнала IrLPBaud16 вне зависимости от выбранной скорости передачи данных. Выбор этого режима снижает энергопотребление, однако может привести к уменьшению дальности связи.
1	SIREN	Разрешение работы модема ИК передачи данных IrDA SIR: 0 – запрещен. Сигнал nSIROUT находится в низком состоянии, данные на входе SIRIN не обрабатываются. 1 – разрешен. Данные передаются на выход nSIROUT и принимаются с входа SIRIN. Линия UARTTXD находится в высоком состоянии. Данные на входе UARTRXD и линиях состояния модема не обрабатываются. В случае, если UARTEN=0 значение бита не играет роли.
0	UARTEN	Разрешение работы приемопередатчика: 0 – работа запрещена. Перед остановкой завершается прием и/или передача обрабатываемого в текущий момент символа. 1 – работа разрешена. Производится обмен данными либо по линиям асинхронного обмена, либо по линиям ИК обмена SIR, в зависимости от состояния

		бита SIREN.
--	--	-------------

Примечание.

Для того, чтобы разрешить передачу данных, необходимо установить в 1 биты TXE и UARTEN. Аналогично, для разрешения приема данных необходимо установить в 1 биты RXE и UARTEN.

Рекомендуется следующая последовательность действий для программирования регистров управления:

- Остановите работу приемопередатчика.
- Дождитесь окончания приема и/или передачи текущего символа данных.
- Сбросьте буфер передатчика путем установки бита FEN регистра LCR\_H в 0.
- Изменить настройки регистра CR.
- Возобновите работу приемопередатчика.

### **IFLS. Регистр порога прерывания по заполнению буфера FIFO**

Данный регистр используется для установки порогового значения заполнения буферов передатчика и приемника, по достижению которых генерируется сигнал прерывания UARTTXINTR или UARTRXINTR, соответственно. Прерывание генерируется в момент перехода величины заполнения буфера через заданное значение.

После сброса в регистре устанавливается порог, соответствующий заполнению половины буфера. Формат регистра и значения его битов представлены в Таблица 158.

**Таблица 158** Регистр IFLS

Биты	Наименование	Назначение
15:6		Резерв. Не модифицируйте. При чтении выдаются нули.
5:3	RXIFLSEL	Порог прерывания по заполнению буфера приемника: b000 = Буфер заполнен на 1/8 b001 = Буфер заполнен на 1/4 b010 = Буфер заполнен на 1/2 b011 = Буфер заполнен на 3/4 b100 = Буфер заполнен на 7/8 b101-b111 = резерв.
2:0	TXIFLSEL	Порог прерывания по заполнению буфера передатчика: b000 = Буфер заполнен на 1/8 b001 = Буфер заполнен на 1/4 b010 = Буфер заполнен на 1/2 b011 = Буфер заполнен на 3/4 b100 = Буфер заполнен на 7/8 b101-b111 = резерв.

**IMSC. Регистр установки сброса маски прерывания**

При чтении выдается текущее значение маски. При записи производится установка или сброс маски на соответствующее прерывание.

После сброса все биты регистра маски устанавливаются в нулевое состояние.

Назначение битов регистра IMSC показано в Таблица 159.

**Таблица 159** Регистр IMSC

Биты	Наименование	Назначение
15:11		Резерв. Не модифицируйте. При чтении выдаются нули.
10	OEIM	Маска прерывания по переполнению буфера UARTOEINTR. 1 – установлена, 0 – сброшена.
9	BEIM	Маска прерывания по разрыву линии UARTBEINTR. 1 – установлена, 0 – сброшена.
8	PEIM	Маска прерывания по ошибке контроля четности UARTPEINTR. 1 – установлена, 0 – сброшена.
7	FEIM	Маска прерывания по ошибке в структуре кадра UARTFEINTR. 1 – установлена, 0 – сброшена.
6	RTIM	Маска прерывания по таймауту приема данных UARTRTINTR. 1 – установлена, 0 – сброшена.
5	TXIM	Маска прерывания от передатчика UARTTXINTR. 1 – установлена, 0 – сброшена.
4	RXIM	Маска прерывания от приемника UARTRXINTR. 1 – установлена, 0 – сброшена.
3	DSRMIM	Маска прерывания UARTDSRINTR по изменению состояния линии nUARTDSR. 1 – установлена, 0 – сброшена.
2	DCDMIM	Маска прерывания UARTDCDINTR по изменению состояния линии nUARTDCD. 1 – установлена, 0 – сброшена.
1	CTSMIM	Маска прерывания UARTCTSINTR по изменению состояния линии nUARTCTS. 1 – установлена, 0 – сброшена.
0	RIMIM	Маска прерывания UARTRIINTR по изменению состояния линии nUARTRI. 1 – установлена, 0 – сброшена.

**RIS. Регистр состояния прерываний**

Этот регистр доступен только для чтения и содержит текущее состояние прерываний без учета маскирования. Данные, записываемые в регистр, игнорируются.

Предупреждение. После сброса все биты регистра, за исключением битов прерывания по состоянию модема (биты с 3 по 0), устанавливаются в 0. Значение битов прерывания по состоянию модема после сброса не определено.

Назначение бит в регистре RIS представлено в Таблица 160.

**Таблица 160** Регистр RIS

Биты	Наименование	Назначение
15:11		Резерв. Не модифицируйте. При чтении выдаются нули.
10	OERIS	Состояние прерывания по переполнению буфера UARTOEINTR.
9	BERIS	Состояние прерывания по разрыву линии UARTBEINTR.
8	PERIS	Состояние прерывания по ошибке контроля четности UARTPEINTR.
7	FERIS	Состояние прерывания по ошибке в структуре кадра UARTFEINTR.
6	RTRIS	Состояние прерывания по таймауту приема данных UARTRTINTR (см. Примечание).
5	TXRIS	Состояние прерывания от передатчика UARTRXINTR.
4	RXRIS	Состояние прерывания от приемника UARTRXINTR.
3	DSRRMIS	Состояние прерывания UARTDSRINTR по изменению линии nUARTDSR.
2	DCDRMIS	Состояние прерывания UARTDCDINTR по изменению линии nUARTDCD.
1	CTSRMIS	Состояние прерывания UARTCTSINTR по изменению линии nUARTCTS.
0	RIRMIS	Состояние прерывания UARTRIINTR по изменению линии nUARTRI.

**Примечание**

Сигнал маски прерывания по таймауту используется в качестве разрешения перехода в режим пониженного энергопотребления. Поэтому чтение состояния прерывания по таймауту из регистров MIS and RIS даст одинаковый результат.

**MIS. Регистр маскированного состояния прерываний**

Этот регистр доступен только для чтения и содержит текущее состояние прерываний с учетом маскирования. Данные, записываемые в регистр, игнорируются.

После сброса все биты регистра, за исключением битов прерывания по состоянию модема (биты с 3 по 0), устанавливаются в 0. Значение битов прерывания по состоянию модема после сброса не определено. Назначение бит в регистре MIS представлено в Таблица 161.

**Таблица 161** Регистр MIS

Биты	Наименование	Назначение
15:11		Резерв. Не модифицируйте. При чтении выдаются нули.
10	OEMIS	Маскированное состояние прерывания по переполнению буфера UARTOEINTR.
9	BEMIS	Маскированное состояние прерывания по разрыву линии UARTBEINTR.
8	PEMIS	Маскированное состояние прерывания по ошибке

		контроля четности UARTPEINTR.
7	FEMIS	Маскированное состояние прерывания по ошибке в структуре кадра UARTFEINTR.
6	RTMIS	Маскированное состояние прерывания по таймауту приема данных UARTRTINTR.
5	TXMIS	Маскированное состояние прерывания от передатчика UARTTXINTR.
4	RXMIS	Маскированное состояние прерывания от приемника UARTRXINTR.
3	DSRMMIS	Маскированное состояние прерывания UARTDSRINTR по изменению линии nUARTDSR.
2	DCDMMIS	Маскированное состояние прерывания UARTDCDINTR по изменению линии nUARTDCD.
1	CTSMMIS	Маскированное состояние прерывания UARTCTSINTR по изменению линии nUARTCTS.
0	RIMMIS	Маскированное состояние прерывания UARTRIINTR по изменению линии nUARTRI.

### ICR. Регистр сброса прерываний

Этот регистр доступен только для записи и предназначен для сброса признака прерывания по заданному событию путем записи 1 в соответствующий бит. Запись в любой из разрядов регистра 0 игнорируется. Назначение бит в регистре ICR представлено в Таблица 162.

**Таблица 162** Регистр ICR

Биты	Наименование	Назначение
15:11		Резерв. Не модифицируйте. При чтении выдаются нули.
10	OEIC	Сброс прерывания по переполнению буфера UARTOEINTR.
9	BEIC	Сброс прерывания по разрыву линии UARTBEINTR.
8	PEIC	Сброс прерывания по ошибке контроля четности UARTPEINTR.
7	FEIC	Сброс прерывания по ошибке в структуре кадра UARTFEINTR.
6	RTIC	Сброс прерывания по таймауту приема данных UARTRTINTR.
5	TXIC	Сброс прерывания от передатчика UARTTXINTR.
4	RXIC	Сброс прерывания от приемника UARTRXINTR.
3	DSRMIC	Сброс прерывания UARTDSRINTR по изменению линии nUARTDSR.
2	DCDMIC	Сброс прерывания UARTDCDINTR по изменению линии nUARTDCD.
1	CTSMIC	Сброс прерывания UARTCTSINTR по изменению линии nUARTCTS.
0	RIMIC	Сброс прерывания UARTRIINTR по изменению линии nUARTRI.

### **DMACR Регистр управления прямым доступом к памяти**

Регистр доступен по чтению и записи. После сброса все биты регистра обнуляются.

Назначение бит регистра DMACR представлено в Таблица 163.

**Таблица 163** Регистр DMACR

Биты	Наименование	Назначение
15:13		Резерв. Не модифицируйте. При чтении выдаются нули.
2	DMAONERR	Если бит установлен в 1, в случае возникновения прерывания по обнаружению ошибки блокируются запросы DMA от приемника UARTRXDMASREQ и UARTRXDMABREQ.
1	TXDMAE	Использование DMA при передаче. Если бит установлен в 1, разрешено формирование запросов DMA для обслуживания буфера FIFO передатчика.
0	RXDMAE	Использование DMA при приеме. Если бит установлен в 1, разрешено формирование запросов DMA для обслуживания буфера FIFO приемника.

### **UARTTCR Регистр управления тестированием**

Регистр доступен по чтению и записи. После сброса все биты регистра обнуляются.

Назначение бит регистра UARTTCR представлено в Таблица 164.

**Таблица 164** Регистр UARTTCR

Биты	Наименование	Назначение
15:13		Резерв. При чтении значение непредсказуемо.
2	SIRTEST	Разрешение приёма данных в кольцевом режиме с выхода IrDA передатчика. 1- разрешено 0 – запрещено Используется совместно с установкой бита LBE в регистре UART_CR
1	TESTFIFO	Разрешение чтения данных из FIFO передатчика и запись в FIFO приёмника. 1- разрешено 0 - запрещено
0	ITEN	Перевод контроллера UART в тестовый режим 1- тестовый режим разрешён 0- тестовый режим запрещён

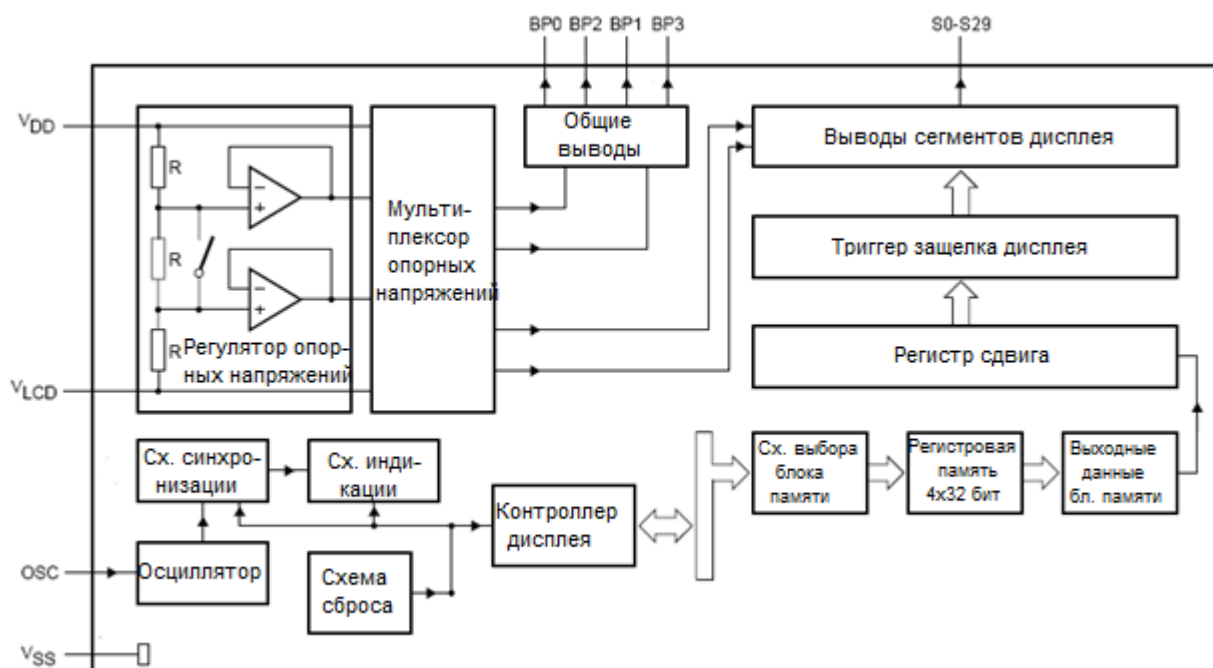
### Контроллер LCD

Контроллер LCD это периферийный блок, который сопрягается с любым жидкокристаллическим дисплеем имеющим низкую величину мультиплекса. Контроллер обеспечивает генерацию сигналов управления как для статических, так и для LCD с мультиплексом содержащих до четырёх общих выводов и до тридцати сегментов. Все возможные конфигурации дисплеев приведены в Таблица 165.

**Таблица 165**

Количество активных общих выводов	Число сегментов	Число 7-и сегментных цифр	Число 14-и сегментных букв	Число точек
4	120	15 цифр + 15 символьных обозначений	8 букв + 8 символьных обозначений	120 (4x30)
3	90	11 цифр + 11 символьных обозначений	6 букв + 6 символьных обозначений	90 (3x30)
2	60	7 цифр + 7 символьных обозначений	4 буквы + 4 символьных обозначения	60 (2x30)
1	30	3 цифры + 3 символьных обозначения	2 буквы + 2 символьных обозначения	30

Структурная схема контроллера LCD приведена на Рис. 86.

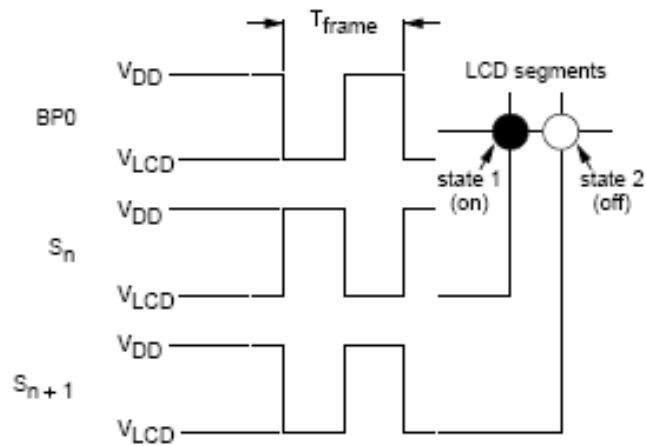


**Рис. 86**



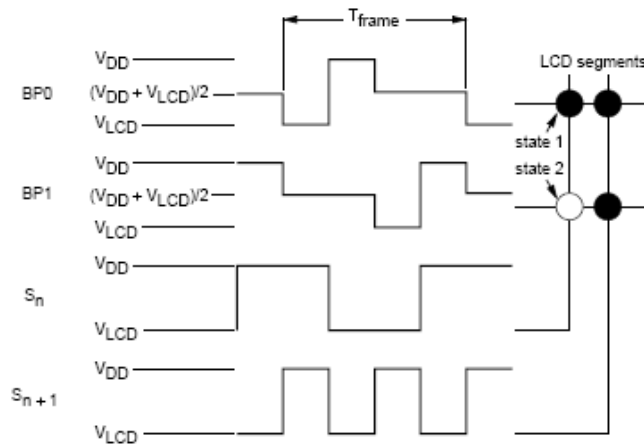


Ниже на рисунках приведены формы сигналов управления на площадках для различных конфигураций.



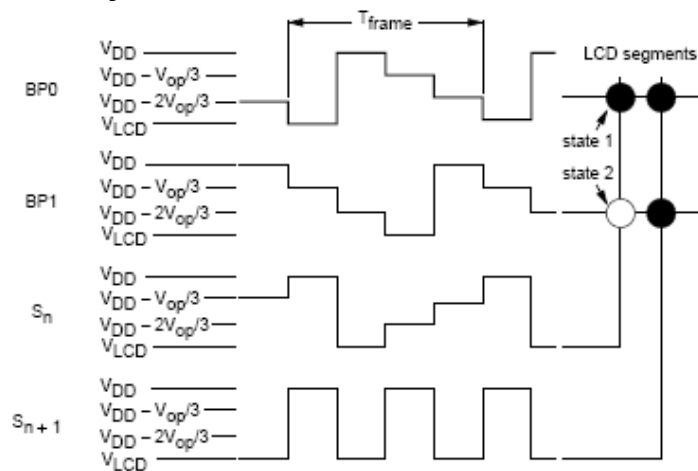
**Рис. 88**

**Статический режим управления**



**Рис. 89**

**Режим управления с мультиплексом 1:2 и 1/2 bias**



**Рис. 90**

**Режим управления с мультиплексом 1:2 и 1/3 bias**

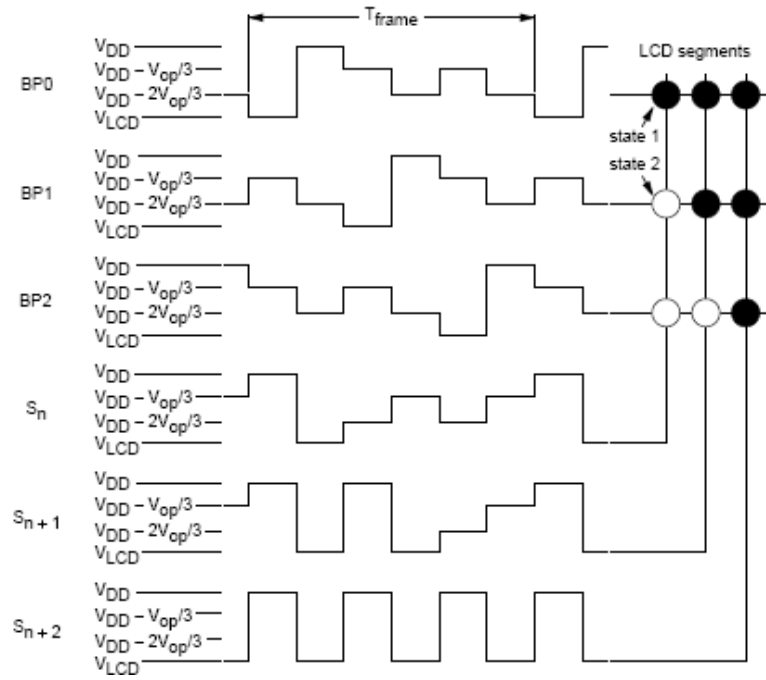


Рис. 91

Режим управления с мультиплексом 1:3

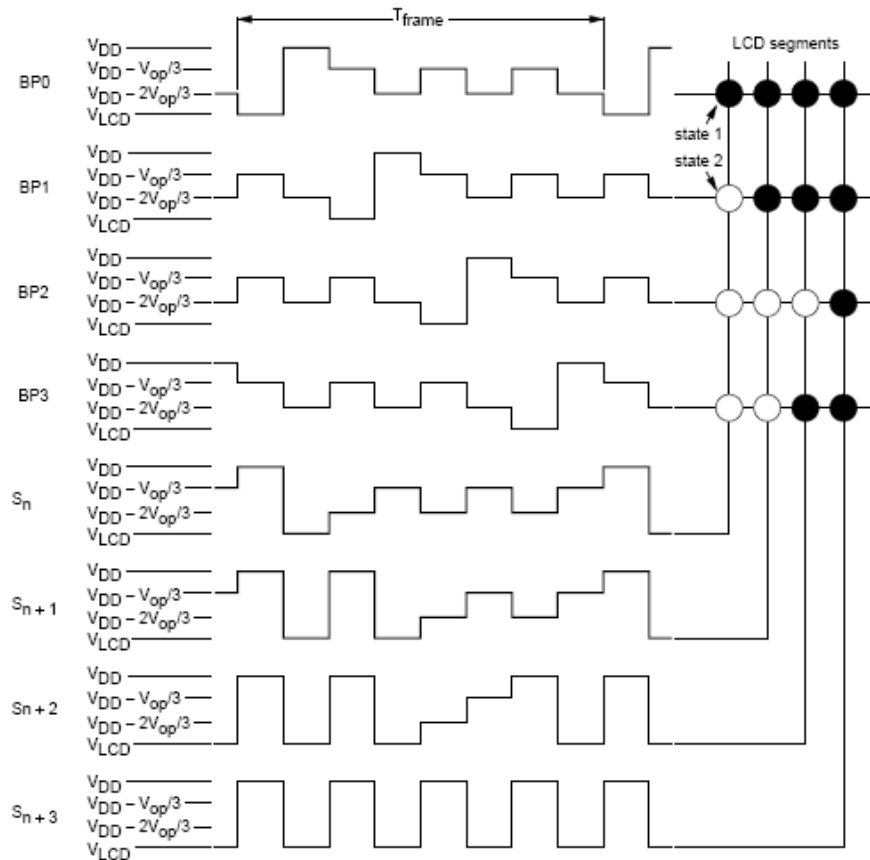


Рис. 92 Режим управления с мультиплексом 1:4

**Описание регистров контроллера LCD**

**Таблица 166**

Базовый Адрес	Название	Описание
0x4003_8000	LCD	Контроллер LCD дисплея
Смещение		
0x00	CR	Регистр управление контроллером
0x04	ROW1	Строка данных основной памяти для общего выхода BP0
0x08	ROW2	Строка данных основной памяти для общего выхода BP1
0x0C	ROW3	Строка данных основной памяти для общего выхода BP2
0x10	ROW4	Строка данных основной памяти для общего выхода BP3
0x14	AROW1	Строка данных альтернативной памяти для общего выхода BP0
0x18	AROW2	Строка данных альтернативной памяти для общего выхода BP1
0x1C	AROW3	Строка данных альтернативной памяти для общего выхода BP2
0x20	AROW4	Строка данных альтернативной памяти для общего выхода BP3

**CR. Регистр управления**

Номер	7	6	5	4	3	2	1	0
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса	0	0	0	0	0	0	1	1

VLCDON	A	BF1	BF0	E	B	M1	M0
--------	---	-----	-----	---	---	----	----

Номер	15	14	13	12	11	10	9	8
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса	0	0	0	0	0	0	0	0

DIV[15:8]
-----------

Номер	23	22	21	20	19	18	17	16
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса							0	0

DIV[23:16]
------------

- R/W - бит доступен на чтение и запись
- RO - бит доступен только на чтение
- U - бит физически не реализован или зарезервирован.

**Таблица 167**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..25		Зарезервировано
24	CLK_SEL	<b>Выбор частоты синхронизации блока</b> 1 – частота LSI 0 – частота LSE
23..8	DIV	<b>Делитель опорной частоты осциллятора 32,768 кГц</b> Значение делителя определяет длительность фрейма для соответствующей схемы управления. Выбирается с учётом следующих соотношений: 1:1 MUX $f_{frame}=32,768 /DIV$ ; 1:2 MUX $f_{frame} =32,768/2*DIV$ ; 1:3 MUX $f_{frame} =32,768/3*DIV$ ; 1:4 MUX $f_{frame} =32,768/4*DIV$ ;
7	VLCDON	<b>Выбор источника питания <math>V_{LCD}</math></b> 1- питания $V_{LCD}$ формируется на внешней площадке 0 – питание $V_{LCD}$ формируется внутренним источником опорного напряжения
6	A	<b>Выбор схемы мерцания LCD</b> 1 – альтернативная схема мерцания (из альтернативной памяти) 0 – нормальная схема мерцания (гашения с частотой заданной битами BF) Если BF=0, то мерцание отключено
5..4	BF	<b>Разрешение мерцания LCD</b> 0 – мерцание запрещено 1 – мерцание с частотой 2 Гц 2 - мерцание с частотой 1 Гц 3 - мерцание с частотой 0,5 Гц
3	E	<b>Разрешение работы контроллера</b> 1- контроллер включён 0- контроллер отключен
2	B	<b>Выбор режима bias</b> 0 – 1/3 bias 1 – 1/2 bias
1..0	M	<b>Выбор схемы управления</b> 0 – статическое управление 1- 1:2 MUX 2- 1:3 MUX 3- 1:4 MUX

**ROW4-ROW1. Регистры данных основной памяти**

Номер	31	30	29	0
Доступ*	U	U	R/W	R/W
Значени			0	0

е после сброса

-	-	S29-S0
---	---	--------

**AROW4-AROW1. Регистры данных альтернативной памяти**

Номер	31	30	29	0
Доступ*	U	U	R/W	R/W
Значение после сброса			0	0

-	-	S29-S0
---	---	--------

## **Прерывания и исключения**

### **Состояние исключений**

Inactive – исключение не находится в стадии Active или Pending

Pending – исключение находится в состоянии ожидания обработки процессором. Запрос прерывания от периферийных блоков или программы может изменить состояние соответствующего прерывания на состояние pending

Active – исключение начало обрабатываться процессором, но еще не закончено. Обработчик исключения может быть прерван другим обработчиком исключения. В этом случае оба исключения находятся в состоянии Active

Active и Pending – исключения начало обрабатываться процессором, но появилось новое исключение в состоянии pending от того же источника.

### **Типы исключений**

Исключения бывают следующих типов

## **RESET**

RESET вызывается при включении питания и горячем сбросе. Модель исключений трактует RESET как специальная форма исключения. Когда выставляется RESET, работа процессора останавливается потенциально в любой точке инструкций. Когда RESET убирается, выполнение перезапускается с адреса заданном в таблице векторов для сброса. Выполнение перезапускается в thread режиме.

## **NON MASKABLE INTERRUPT (NMI)**

Не маскируемое прерывание (NMI) может быть вызвано периферией или установлено программой. Это самое высокоприоритетное исключение после сброса. Всегда разрешено и имеет фиксированный приоритет -2.

NMI не может быть:

- замаскировано или предотвращено от активации из другого исключения
- прерывает любые исключения кроме RESET

## **HARD FAULT**

Hard Fault исключение происходит при ошибке при обработке исключений или потом что исключение не может быть обработано каким либо другим механизмом. Hard fault имеет фиксированный приоритет -1, означающий, что он имеет больший приоритет чем любое из исключений с конфигурируемым приоритетом.

## **SVCALL**

Исключение Supervisor Call (SVCALL) возникает при выполнении инструкции SVC. В приложениях с использованием Операционных Сред инструкция SVC может использоваться для доступа к функциям ОС и драйверам устройств.

## **PendSV**

PendSV является прерыванием запросом сервисов системного уровня. В приложениях с использованием ОС PendSV используется для переключения контекстов, когда нет других активных исключений.

## **SysTick**

Исключение SysTick является исключением генерируемым системным таймером, когда он обнуляется. Программное обеспечение так же может генерировать исключение SysTick. В приложениях с использованием ОС процессор может использовать это исключение для подсчета системных циклов

## **Прерывания (IRQ)**

Прерывания или IRQ это исключения вызываемые периферийными устройствами или программными запросами. Все прерывания асинхронны по отношению к выполняемым инструкциям. В системе прерывания используются для коммуникации периферии и процессора

**Таблица 168** Таблица различных типов исключений

Номер исключения	IRQ номер	Тип	Приоритет	Адрес вектора обработчика (смещение)	Активация
1	-	RESET	-3, наивысший	0x0000_0004	Асинхронный
2	-14	NMI	-2	0x0000_0008	Асинхронный
3	-13	Hard Fault	-1	0x0000_000C	-
4-10	-	Reserved	-	Зарезервировано	-
11	-5	SVCall	Конфигурируемый	0x0000_002C	Синхронный
12-13	-	-	-	Зарезервировано	-
14	-2	PendSV	Конфигурируемый	0x0000_0038	Асинхронный
15	-1	SysTick	Конфигурируемый	0x0000_003C	Асинхронный
16 и выше	0 и выше	IRQ	Конфигурируемый	0x0000_0040 и выше	Асинхронный

Для асинхронных исключений, кроме RESET, процессор может выполнить другие инструкции между возникновением сигнала исключения и входом в обработчик.

Программа в Privileged режиме может запретить прерывания, имеющие конфигурируемый приоритет.

### **Обработчики исключений**

Для обработки исключений используются:

Interrupt Service Routines (ISRs)

Прерывания с IRQ0 по IRQ31 обрабатываются ISRs  
Fault Handlers

Обрабатываются только fault handlers.

System handlers

NMI, PendSV, SVCall, systick и HardFault обрабатываются system handlers

Таблица векторов содержит указатель стека, вектор входа по RESET и стартовые адреса обработчиков, так же называемых векторами. На рисунке представлена последовательность векторов в таблице. Младший бит всех векторов должен быть равен 1, указывающий на то, что обработчик выполняется в Thumb режиме.

Номер исключения	номер IRQ	Вектор	Смещение
16+n	n	IRQn	0x40+4n
.		⋮	⋮
.		⋮	⋮
.		⋮	⋮
18	2	IRQ2	0x48
17	1	IRQ1	0x44
16	0	IRQ0	0x40
15	-1	если применяется SysTick	0x3C
14	-2	PendSV	0x38
13		Зарезервировано	
12			
11	-5	SVCall	0x2C
10			
9			
8			
7		Зарезервировано	
6			
5			
4			
3	-13	HardFault	0x10
2	-14	NMI	0x0C
			0x08
1		Reset	0x04
		Начальное значение указателя стека	0x00

**Рис. 93**

При системном сбросе, таблица векторов располагается по фиксированному адресу 0x00000000.

### Приоритеты исключений

- Более малое значение приоритета означает больший приоритет.



- конфигурируемы все приоритеты, кроме RESET и Hard Fault

Если программное обеспечение не задает приоритетов, то все они имеют приоритет 0.

Конфигурируемый приоритет может быть в диапазоне от 0 до 192 с шагом 64. Это означает что RESET, Hard Fault и NMI имеющие отрицательное значение приоритета всегда имеют больший приоритет.

Если имеется несколько исключений с одинаковым приоритетом, то больший приоритет имеет исключение с меньшим порядковым номером.

Если процессор выполняет обработчик исключения и происходит исключение с большим приоритетом, то происходит переход на обработчик исключения с большим приоритетом. Если при выполнении обработчика произошло исключение с таким же приоритетом, то это исключение будет выполнено по завершению текущего обработчика, несмотря на порядковый номер исключения.

### **Вход в обработчик и выход из обработчика**

При описании используются следующие термины:

### **Приоритетное прерывание**

Выполнение процессором процедуры обработки исключительной ситуации (далее по тексту – исключения), может быть прервано в случае возникновения исключения с приоритетом выше, чем у обрабатываемого. В случае, если внутри обработчика исключения возникает прерывание более высокого приоритета возникает ситуация, называемая вложенным исключением.

### **Возврат**

Возврат из обработчика осуществляется по завершении обработки исключительной ситуации, с одновременным выполнением следующих условий:

- в системе отсутствуют необработанные исключения с достаточным приоритетом;
- завершённый обработчик не обрабатывал запоздавшее исключение (late-arriving exception).

Процессор обращается к стеку и восстанавливает состояние, имевшее место до вызова обработчика.

### **Передача управления без восстановления контекста (tail-chaining)**

Данный механизм ускоряет процесс обработки исключений. По завершении выполнения обработчика осуществляется проверка наличия необработанных исключений и в случае, если исключения, требующие вызова обработчика присутствуют, восстановление состояния процессора из стека не производится, а управление передается непосредственно на новый обработчик.

### **Запоздавшее исключение (late-arriving exception)**

В случае, если во время сохранения состояния при входе в обработчик возникла исключительная ситуация с более высоким приоритетом, процессор передает управление непосредственно высокоприоритетному обработчику.

Подобный способ обработки высокоприоритетного исключения возможен до момента начала выполнения первой инструкции процедуры обработки исключительной ситуации. После возврата из обработчика запоздавшего исключения осуществляется передача управления на прерванный низкоприоритетный обработчик без восстановления контекста.

### **Вход в процедуру обработки исключения**

Вызов процедуры обработки исключения возникает в случае наличия необработанных исключительных ситуаций с достаточным приоритетом и выполнения одного из следующих условий:

- процессор находится в режиме приложения (thread mode);
- новая исключительная ситуация имеет приоритет выше, чем обрабатываемая в текущий момент времени, что приводит к приоритетному прерыванию выполнения текущего обработчика. В этом случае возникает вложение одного исключения в другое.

Для того, чтобы исключительная ситуация имела достаточный приоритет, необходимо, чтобы уровень ее приоритета был выше значений, заданных в регистрах маскирования. В противном случае исключение находится в состоянии ожидания, процедура его обработки не вызывается.

При необходимости вызова обработчика, за исключением случаев обработки запоздавшего исключения и передачи управления на обработчик без восстановления контекста, процессор заносит в текущий стек восемь слов данных, называемые далее стековым фреймом. Этот фрейм включает в себя следующие значения:

- Регистры R0-R3, R12;
- Адрес возврата;
- Регистр PSR;
- Регистр LR.

Указанная операция далее будет называться сохранением контекста. Непосредственно после ее выполнения указатель стека равен младшему адресу стекового фрейма.

Во время сохранения контекста производится выравнивание адреса стека по границе двойного слова.

Стековый фрейм содержит адрес возврата, указывающий на ближайшую невыполненную инструкцию прерванной программы. По завершении процедуры обработки исключения значений адреса возврата заносится в счетчик команд, после чего выполнение программы возобновляется с прерванной точки.

Одновременно с сохранением контекста процессор осуществляет выборку адреса точки входа в процедуру обработки исключения из таблицы векторов исключений. По завершении операции сохранения контекста процессор передает управление на полученный из таблицы адрес.

Одновременно в регистр LR записывается значение EXC\_RETURN, позволяющее определить, какой из двух указателей стека соответствует данному стековому фрейму и в каком режиме находился процессор перед входом в обработчик.

Если во время передачи управления не возникло исключения с более высоким приоритетом, процессор начинает выполнение вызванной процедуры обработки и автоматически изменяет состояние текущего прерывания с ожидающего обработки на активное.

В противном случае процессор передает управление обработчика высокоприоритетной исключительной ситуации без изменения состояния отложенного прерывания в соответствии с правилами, изложенными в разделе «Запоздавшее исключение (late-arriving exception)».

### **Возврат из обработчика исключения**

Возврат из обработчика исключения осуществляется в случае, если процессор находится в режиме обработчика (handler mode) и выполняет одну из следующих инструкций, позволяющих загрузить значение EXC\_RETURN в регистр PC:

- инструкцию POP с аргументом PC;
- инструкцию BX с любым регистром;

Значение EXC\_RETURN загружается в регистр LR по входу в обработчик исключения. Механизм обработки исключений использует это значение для того, чтобы определить, завершил ли процессор выполнение процедуры обработки исключительной ситуации. Младшие четыре бита EXC\_RETURN содержат информацию о состоянии стека и режиме работы процессора. Информация о назначении разрядов EXC\_RETURN[3:0] и особенности процесса возврата из обработчика исключения представлены в Таблица 169.

Процессор устанавливает биты EXC\_RETURN [31:4] в 0xFFFFFFFF. Загрузка данного значения в PC указывает на завершение процедуры обработки исключения и заставляет процессор выполнить необходимые действия для возврата из обработчика.

**Таблица 169** Возврат из обработчика исключения

EXC_RETURN [3:0]	Описание
bXXX0	Резерв.
b0001	Возврат в режим обработчика. Восстановление контекста осуществляется из стека MSP. Дальнейшая работа осуществляется со стеком MSP.
b0011	Резерв.
b01X1	Резерв.
b1001	Возврат в режим приложения. Восстановление контекста осуществляется из стека MSP. Дальнейшая работа осуществляется со стеком MSP.
b1101	Возврат в режим приложения. Восстановление контекста осуществляется из стека PSP. Дальнейшая работа осуществляется со стеком PSP.
b1X11	Резерв.



### **Управление электропитанием**

В процессоре Cortex-M0 предусмотрены следующие режимы ожидания (пониженного энергопотребления):

Sleep – останов синхросигнала для процессора

Deep sleep – останов синхросигнала для процессора, PLL и Flash

Выбор процессором конкретного режима ожидания определяется значением бита SLEEPDEEP регистра SCR (см. раздел «Регистр управления системой»)

Далее в разделе описаны механизмы перехода в режим пониженного энергопотребления и условия выхода из этого режима.

### **Переход в режим пониженного энергопотребления**

Система может формировать ложные сигналы событий, выводящие процессор из ожидания, например, они возникают при работе отладчика. Следовательно, программное обеспечение должно быть способно перевести процессор обратно в указанный ожидания. Для этого можно, например, организовать в программе пустой цикл.

### **Ожидание прерывания**

Инструкция ожидания прерывания WFI (wait for interrupt) после своего выполнения немедленно переводит процессор в режим пониженного энергопотребления.

Ожидание события

Инструкция ожидания сигнала события WFE (wait for event) переводит или не переводит процессор в режим пониженного энергопотребления в зависимости от результата проверки одноразрядного регистра события. При этом процессор проверяет значение регистра события, и в случае, если он равен 0, приостанавливает дальнейшее выполнение команд и переходит в состояние ожидания. В случае, если он равен 1, процессор записывает в регистр события 0 и продолжает нормальную работы без перехода в режим ожидания.

### **Переход в режим ожидания по выходу из обработчика исключения (режим sleep-on-exit)**

В случае, если бит SLEEPONEXIT регистра SCR установлен в 1, по завершении выполнения обработчика исключения процессор возвращается в режим приложения, после чего немедленно переходит в состояние пониженного энергопотребления.

Данный механизм рекомендуется использовать в задачах, в которых процессора используется только для обработки исключений.

### **Выход из состояния ожидания**

Условия выхода процессора из режима ожидания зависят от причины, по которой он был переведен в этот режим.

Выход из ожидания по команде WFI и в режиме sleep-on-exit

Как правило, процессор выходит из режима ожидания только в случае возникновения исключительной ситуации с приоритетом, достаточным для активизации соответствующего обработчика.

В некоторых приложениях может возникнуть необходимость выполнения процедур восстановления системы после выхода процессора из режима пониженного энергопотребления, однако до того, как он начнет выполнять обслуживание прерываний. Для того, чтобы добиться этого, достаточно установить бит PRIMASK в 1. В случае возникновения в системе разрешенного прерывания с приоритетом, выше текущего приоритета, процессор будет выведен из ожидания, однако не сможет передать управление обработчику прерывания до тех пор, пока бит PRIMASK не будет установлен в 0.

Более подробная информация о бите PRIMASK представлена в разделе, “Регистры маскирования исключений”.

### **Выход из ожидания по команде WFE**

Процессор выходит из режима ожидания в случае обнаружения исключительной ситуации с приоритетом, достаточным для активизации обработчика.

Кроме того, в случае установки бита SEVONPEND регистра SCR в 1, любое новое не обслуженное прерывание формирует сигнал события, и выводит процессор из ожидания, даже если оно запрещено или имеет приоритет, недостаточно высокий для запуска обработчика.

Более подробная информация о регистре SCR представлена в разделе “Регистр управления системой”.

### **Рекомендации по программированию режима энергопотребления**

В стандарте ANSI языка C отсутствует возможность непосредственной генерации инструкций WFI и WFE. В CMSIS предусмотрены встроенные функции, предназначенные для включения в код этих инструкций:

```
void __WFE(void) // Wait for Event
void __WFI(void) // Wait for Interrupt
```

Периферийные блоки формируют прерывания с IRQ0 до IRQ31

**Таблица 170**

Прерывания	Блок	Принцип формирования
IRQ1	UART1	Сигнал UARTINTR
IRQ2	UART2	Сигнал UARTINTR
IRQ3	SSP1	Сигнал SSPINTR
IRQ4	POWER	Сигнал прерывания от POWER Detecor
IRQ5	WWDG	Сигнал прерывания от WWDG
IRQ6	Timer1	Сигнал прерывания от Таймера TIM_STATUS и TIM_IE
IRQ7	Timer2	Аналогично
IRQ8	ADC	Сигналы прерываний от АЦП EOCIF_1 или AWOIF_1 или EOCIF_2 или AWOIF_2
IRQ9	I2C	Сигнал INT при EN_INT
IRQ10	BACKUP	Прерывание от ВКР и часов реального времени

IRQ11	Внешнее прерывание 1	Сигнал EXT_INT0 Выход PC[7] в основном режиме
IRQ12	Внешнее прерывание 2	Сигнал EXT_INT1 Выход PC[8] в основном режиме
IRQ13	Внешнее прерывание 3	Сигнал EXT_INT2 Выход PC[9] в основном режиме
IRQ14	ADC1U	Прерывание от АЦП для измерения напряжений и токов

### **Контроллер прерываний NVIC**

В разделе описан векторный контроллер прерываний с возможностью вложения (NVIC – Nested Vectored Interrupt Controller) и используемые им регистры.

Контроллер обеспечивает поддержку:

- программное задание уровня приоритета в диапазоне от 0 до 192 с шагом 64 независимо каждому прерыванию. Более высокое значение соответствует меньшему приоритету, таким образом, уровень 0 отвечает наивысшему приоритету прерывания;
- срабатывание сигнала прерывания по импульсу и по уровню;
- передача управления из одного обработчика исключения на другой без восстановления контекста.

Процессор автоматически сохраняет в стеке свое состояние (контекст) по входу в обработчик прерывания и восстанавливает его по завершению обработчика, без необходимости непосредственного программирования этих операций. Это обеспечивает обработку исключительных ситуаций с малой задержкой.

Назначение регистров контроллера прерываний представлено в Таблица 171.

**Таблица 171** Обобщенная информация о регистрах контроллера NVIC

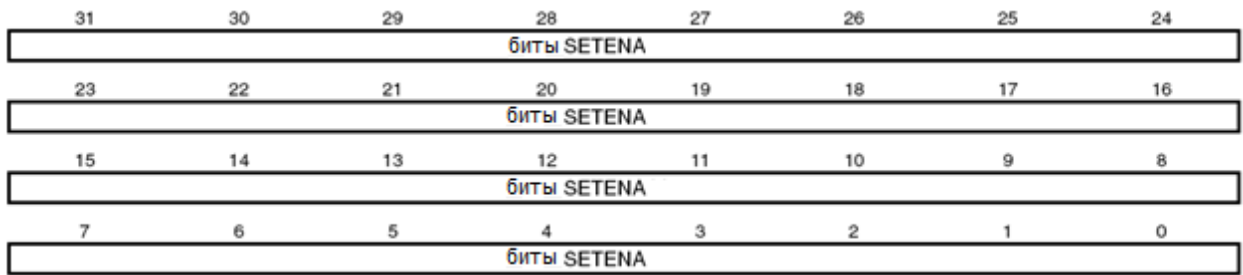
Адрес	Имя	Тип	Значение после сброса	Описание
0xE000E100	ISER	RW	0x00000000	Регистр разрешения прерываний
0xE000E180	ICER	RW	0x00000000	Регистр запрета прерывания
0xE000E200	ISPR	RW	0x00000000	Регистр перевода прерывания в состояние ожидания обслуживания
0xE000E280	ICPR	RW	0x00000000	Регистр сброса состояния ожидания обслуживания
0xE000E400 - 0xE000E41C	IPRO-7	RW	0x00000000	Регистр приоритета прерываний

#### **Регистр разрешения прерываний**

Регистр ISER предназначен для разрешения прерываний (запись) и определения, какие из прерываний разрешены (чтение). Более подробная информация представлена в Таблица 171.

Распределение битов регистра представлено на Рис. 94:





**Рис. 94**

Назначение бит SETENA:

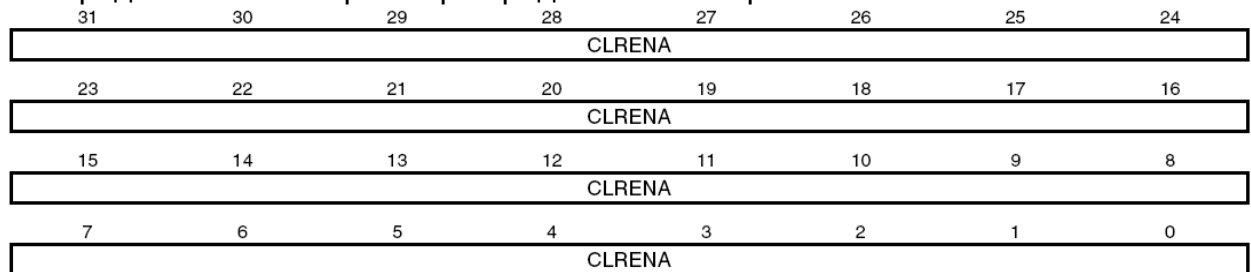
- запись: 0 – не влияет; 1 – разрешение прерывания.
- чтение: 0 – прерывание запрещено; 1 – прерывание разрешено.

При разрешении прерывания, находящегося в состоянии ожидания обработки, контроллер NVIC активизирует его в зависимости от приоритета. Запрос запрещенного прерывания, переводит его в состояние ожидания обработки, однако контроллер NVIC не активизирует его вне зависимости от приоритета.

**Регистр запрета прерываний**

Регистр ICER предназначен для запрета прерываний (запись) и определения, какие из прерываний разрешены (чтение). Более подробная информация представлена в Таблица 171.

Распределение битов регистра представлено на рис.Рис. 95:



**Рис. 95**

Назначение бит CLRENA:

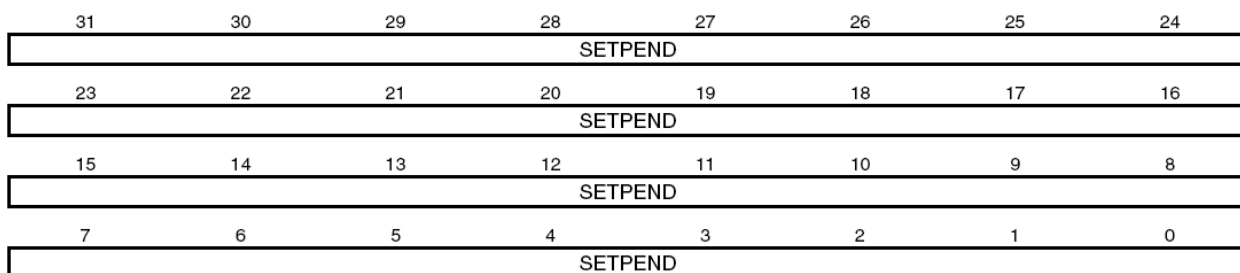
- запись: 0 – не влияет; 1 – запрет прерывания.
- чтение: 0 – прерывание запрещено; 1 – прерывание разрешено.

**Регистр установки состояния ожидания для прерывания**

Регистр ISPR предназначен для принудительного перевода прерываний в состояние ожидания обслуживания (запись) и определения, какие из прерываний находятся в этом состоянии (чтение).

Более подробная информация представлена в Таблица 171.

Распределение битов регистра представлено на рис.:



**Рис. 96**

Назначение бит SETPEND:

- запись: 0 – не влияет; 1 – перевод прерывания в состояние ожидания.
- чтение: 0 – прерывание не ожидает обслуживания; 1 – прерывание ожидает обслуживания.

Запись 1 в бит регистра ISPR, соответствующий:

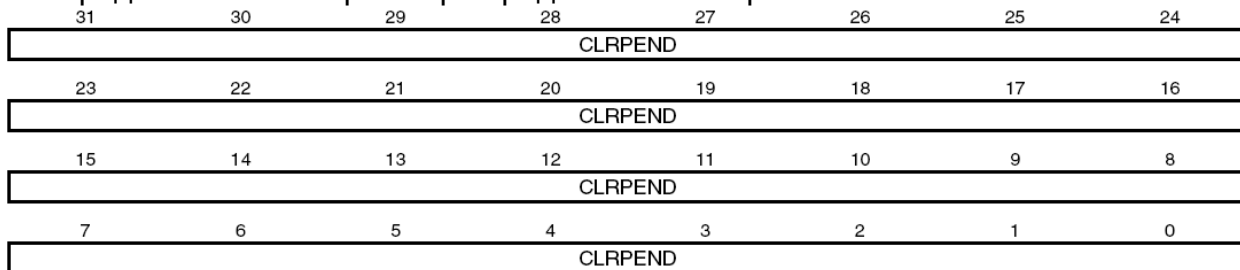
- прерыванию, уже ожидающему обслуживания – не влияет на работу системы;
- запрещенному прерыванию – переводит его в состояние ожидания.

**Регистр сброса состояния ожидания для прерывания**

Регистр ICPR предназначен для принудительного сброса состояния ожидания обслуживания прерывания (запись) и определения, какие из прерываний находятся в состоянии ожидания (чтение).

Более подробная информация представлена в Таблица 171.

Распределение битов регистра представлено на рис.:



**Рис. 97**

Назначение бит CLRPEND:

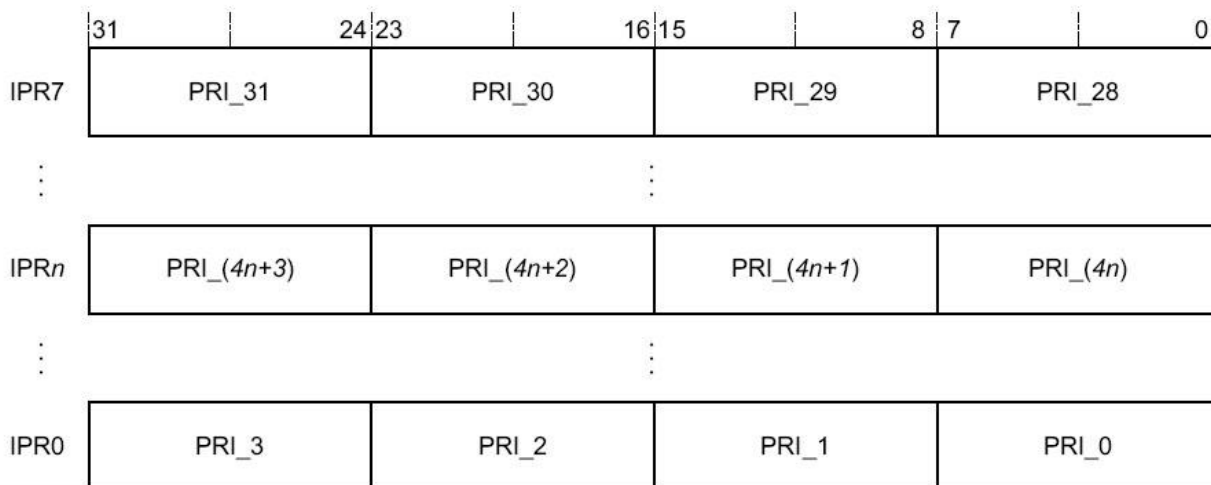
- запись: 0 – не влияет; 1 – сброс состояния ожидания.
- чтение: 0 – прерывание не ожидает обслуживания; 1 – прерывание ожидает обслуживания.

Запись 1 в разряд регистра ICPR, соответствующего прерыванию в активном состоянии, не влияет на работу системы.

**Регистры приоритета прерываний**

Регистры IPR0-IPR7 представляют собой набор 8-битовых полей, каждое из которых соответствует одному прерыванию. Регистры доступны по слову. Обобщенная информация об их характеристиках представлена в Таблица 171. Обобщенная информация о регистрах контроллера NVIC Таблица 171.

Каждый из регистров содержит четыре поля приоритета, которые отображаются на четыре элемента массива PRI[0] ... PRI[31] CMSIS, как показано ниже.



**Рис. 98**

Каждое поле содержит значение приоритета в диапазоне от 0 до 192, причем меньшие значения соответствуют более высокому приоритету соответствующего прерывания. Процессор обеспечивает доступ только к битам [7:6] приоритета, биты [5:0] при чтении всегда равны нулю, а при записи игнорируются. Поэтому, например, запись 255 в регистр запишется как 192.

Для того, чтобы определить номер регистра IPR и смещение данных в регистре необходимо выполнить следующие операции:

- для заданного номера прерывания N номер M соответствующего регистра приоритета равен  $M = N \text{ DIV } 4$ ;
- смещение данных в регистре в зависимости от значения  $N \text{ MOD } 4$  равно:

- 0 – биты регистра [7:0];
- 1 – биты регистра [15:8];
- 2 – биты регистра [23:16];
- 3 – биты регистра [31:24].

***Прерывания, срабатывающие по уровню сигнала***

Процессор способен обрабатывать прерывания, сформированные по уровню сигнала.

Прерывание такого типа считается активным до тех пор, пока периферийное устройство не снимет активный уровень сигнала запроса. Как правило, это происходит после соответствующего обращения процедуры обработки прерывания к периферийному устройству.

После того, как процессор передал управление на обработчик, он автоматически снимает признак ожидания обслуживания прерывания (см. раздел «

**Контроллер прерываний NVIC»).** Если прерывание формируется по уровню сигнала, а сигнал запроса не снят до возврата из обработчика, процессор вновь переведет прерывание в состояние ожидания обслуживания, что, в свою очередь, приведет к повторному вызову его обработчика. Таким образом, периферийное устройство может поддерживать сигнал запроса прерывания в активном состоянии до тех пор, пока не перестанет нуждаться в обслуживании.

### **Аппаратное и программное управление прерываниями**

Процессор Cortex-M0 регистрирует все поступающие прерывания. Перевод прерывания, сформированного периферийным устройством, в состояние ожидания обслуживания осуществляется в одном из следующих случаев:

- контроллер прерываний NVIC обнаруживает, что сигнал запроса имеет высокий логический уровень, а прерывание не активно;
- контроллер прерываний NVIC обнаруживает передний фронт сигнала запроса прерывания;
- программное обеспечение осуществляет запись в соответствующий разряд регистра ISPR0 или соответствующего значения в регистр STIR.

Прерывание находится в состоянии ожидания до тех пор, пока не произойдет одно из следующих событий:

- процессор передаст управление процедуре обработки прерывания. В этом случае прерывание переходит в активное состояние, после чего:
- по завершении обработки прерывания, срабатывающего по уровню, контроллер NVIC проверяет состояние сигнала запроса на прерывание. Если этот сигнал активен, прерывание вновь переводится в состояние ожидания обслуживания, что приводит к немедленной повторной передаче управления на обработчик. В противном случае прерывание переводится в неактивное состояние.
- если в период выполнения процедуры обработки прерывания, настроенного на срабатывание по фронту, не было зафиксировано импульсов на линии запроса, прерывание переводится в неактивное состояние.
- программное обеспечение осуществляет запись в соответствующий разряд регистра сброса состояния ожидания прерывания.

### **Рекомендации по работе с контроллером прерываний**

Доступ к регистрам контроллера из программного обеспечения должен осуществляться по корректно выровненным адресам. Процессор не поддерживает возможность доступа к контроллеру по невыровненным адресам. Требования по выравниванию приведены в описании регистров.

Прерывание может быть переведено в состояние ожидания обслуживания даже в случае, если оно запрещено.

Программное разрешение или запрещение прерываний может осуществляться с помощью инструкций CPSIE I и CPSID I. В CMSIS предусмотрены следующие встроенные функции, генерирующие эти инструкции:

```
void __disable_irq(void) // Disable Interrupts  
void __enable_irq(void) // Enable Interrupts
```

Кроме того, в CMSIS имеется ряд дополнительных функций, обеспечивающих управление контроллером прерываний NVIC:

**Таблица 172** Функции CMSIS для управления контроллером прерываний

Функция	Описание
void NVIC_EnableIRQ(IRQn_t IRQn)	Разрешить IRQn
void NVIC_DisableIRQ(IRQn_t IRQn)	Запретить IRQn
uint32_t NVIC_GetPendingIRQ (IRQn_t IRQn)	Вернуть истину, если прерывание IRQn ожидает обслуживания, ложь – в противном случае
void NVIC_SetPendingIRQ (IRQn_t IRQn)	Перевести IRQn в состояние ожидания обслуживания
void NVIC_ClearPendingIRQ (IRQn_t IRQn)	Сбросить состояние ожидания обслуживания для IRQn
void NVIC_SetPriority (IRQn_t IRQn, uint32_t priority)	Установить приоритет для IRQn
uint32_t NVIC_GetPriority (IRQn_t IRQn)	Считать приоритет IRQn
void NVIC_SystemReset (void)	Сбросить систему

Более подробная информация отражена в документации по CMSIS.

**Блок управления системой ядра**

Блок управления системой (SCB – System control block) обеспечивает доступ к информации о конфигурации и управление работой системы. Регистры блока управления системой представлены в Таблица 173.

**Таблица 173** Обобщенная информация о регистрах блока управления системой

Адрес	Имя	Тип	Значение после сброса	Описание
0xE000ED00	CPUID	RO	0x412FC230	Регистр идентификации процессора
0xE000ED04	ICSR	RW	0x00000000	Регистр управления прерываниями
0xE000ED0C	AIRCR	RW	0xFA050000	Регистр управления прерываниями и программного сброса
0xE000ED10	SCR	RW	0x00000000	Регистр управления системой
0xE000ED14	CCR	RW	0x00000200	Регистр конфигурации и управления
0xE000ED1C	SHPR2	RW	0x00000000	Регистр №2 приоритета системных обработчиков
0xE000ED20	SHPR3	RW	0x00000000	Регистр №3 приоритета системных обработчиков

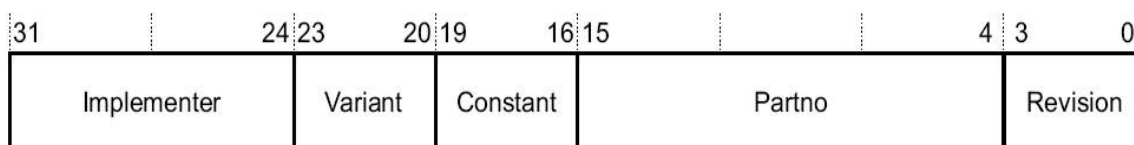
**Упрощенный доступ к регистрам блока управления системой из среды разработки программного обеспечения**

В целях повышения эффективности разработки программного обеспечения в CMSIS предусмотрен упрощенный доступ к регистрам SCB, а именно, регистры SHPR2-SHPR3 в CMSIS отображаются на массив SHP[1].

**Регистр идентификации процессора**

Регистр CPUID содержит информацию о модели процессора, версии и варианте его реализации. Подробная информация о регистре представлена в Таблица 173.

Назначение разрядов регистра представлено на Рис. 99:



**Рис. 99**

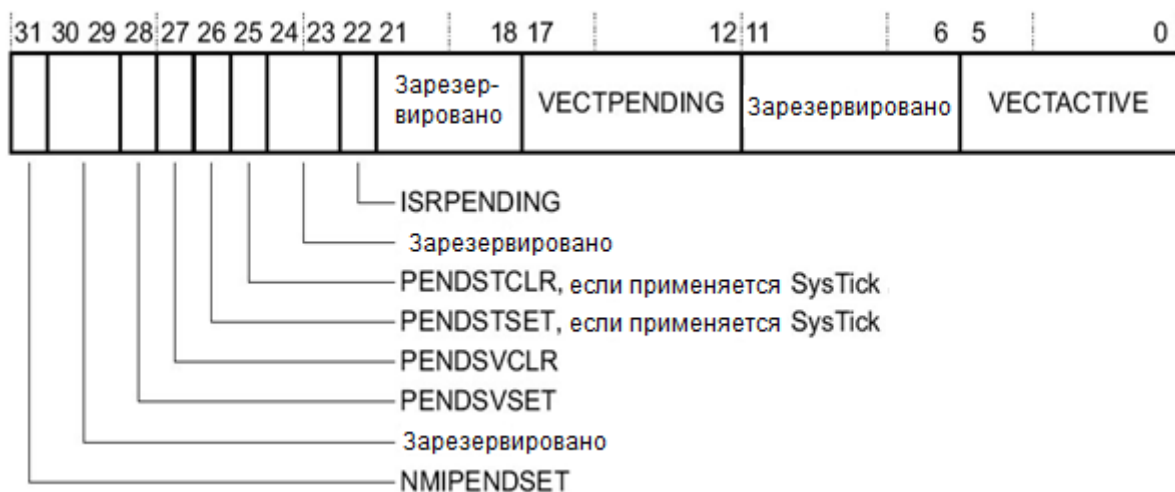
- Implementer – код разработчика 0x41 = ARM.
- Variant – значение r в номере версии rnrp изделия: 0x0 = r0p0;
- Constant – постоянное значение 0xC;
- PartNo – номер модели процессора: 0xC20 = Cortex-M0;
- Revision – значение p в номере версии rnrp изделия: 0x0 = r0p0.

## Регистр управления прерываниями

Регистр ICSR обеспечивает возможность установки и сброса состояния ожидания обслуживания для исключений PendSV и SysTick, а также доступ к следующей информации:

- номер текущего обрабатываемого исключения;
- наличие активных исключений, обработка которых была прервана;
- номер исключения с наивысшим приоритетом, ожидающего обслуживания;
- наличие прерываний, ожидающих обслуживания.

Подробная информация о регистре представлена в Таблица 173. Назначение разрядов регистра представлено на Рис. 100:



**Рис. 100**

NMIPENDSET (RW) – так как NMI имеет самый высокий приоритет, процессор переходит на обработчик прерывания, как только произошла запись “1” в этот бит. После перехода на обработчик прерывания, процессор очищает этот бит. Поэтому если обработчик прочитал “1”, это значит, что сигнал NMI опять перешел в активный уровень во время обработки прерывания.

PENDSVSET (RW) – бит установки состояния ожидания обслуживания для исключения PendSV. Запись: 0 – не влияет на работу системы, 1 – переводит исключение PendSV в состояние ожидания обслуживания. Чтение: 0 – исключение PendSV не ожидает обслуживания, 1 – ожидает.

Запись 1 в разряд PENDSVSET это единственно возможный способ перевода исключения PendSV в состояние ожидания обслуживания.

PENDSVCLR (WO) – бит сброса состояния ожидания обслуживания для исключения PendSV. Запись: 0 – не влияет на работу системы, 1 – сбрасывает состояние ожидания обслуживания для исключения PendSV.

PENDSTSET (RW) – бит установки состояния ожидания обслуживания для исключения SysTick. Запись: 0 – не влияет на работу системы, 1 – переводит исключение SysTick в состояние ожидания обслуживания. Чтение: 0 – исключение SysTick не ожидает обслуживания, 1 – ожидает.

PENDSTCLR (WO) – бит сброса состояния ожидания обслуживания для исключения SysTick. Запись: 0 – не влияет на работу системы, 1 – сбрасывает состояние ожидания обслуживания для исключения SysTick.

Данный бит доступен только для записи, при чтении результат не определен.

ISR\_PENDING (RO) – флаг наличия в системе прерываний (за исключением отказов), ожидающих обслуживания. 0 – ожидающие обслуживания прерывания отсутствуют, 1 – присутствуют.

VECT\_PENDING (RO) – содержит номер ожидающего обслуживания исключения с наивысшим приоритетом, обработка которого в системе разрешена. 0 – не обслуженных исключений нет, другое число – номер ожидающего обслуживания исключения.

Значение данного поля формируется с учетом полей BASEPRI и FAULTMASK, однако не учитывает влияние поля PRIMASK.

VECT\_ACTIVE (RO) – содержит номер активного исключения. 0 – режим приложения, другое число – номер текущего обслуживаемого исключения. Для получения номера запроса прерывания (IRQ) из значения VECT\_ACTIVE необходимо вычесть 16.

Запись в регистр ICSR может привести к непредсказуемым результатам в случае:

- одновременной установки в 1 битов PENDSVSET и PENDSVCLR;
- одновременной установки в 1 битов PENDSTSET и PENDSTCLR;

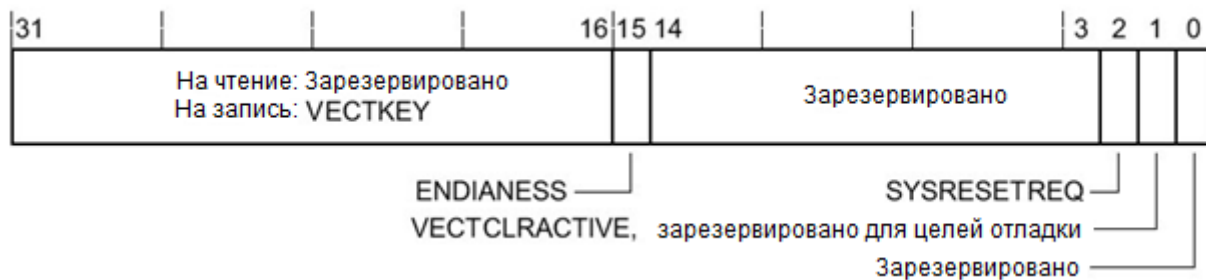
### Регистр управления прерываниями и программного сброса

Регистр AIRCR позволяет задавать группировку приоритетов исключений, порядок следования байт в слове (endian) при доступе к данным, а также управлять процессом сброса системы.

Подробная информация о регистре представлена в Таблица 173.

Для записи данных в регистр необходимо установить его поле VECTKEY в значение 0x05FA, в противном случае попытка записи будет проигнорирована процессором.

Назначение разрядов регистра представлено на Рис. 101:



**Рис. 101**

VECTKEY – ключ доступа к регистру. При записи должен быть равен 0x05FA, в противном случае попытка записи в регистр будет проигнорирована процессором.

ENDIANESS (RO) – порядок следования значащих разрядов при доступе к данным. 0 – младший байт идет первым (little-endian), 1 – старший байт идет первым (big-endian).

SYSRESETREQ (WO) – запрос сброса системы. 0 – не влияет на работу, 1 – инициирует сигнал сброса процессора. При чтении возвращает 0.

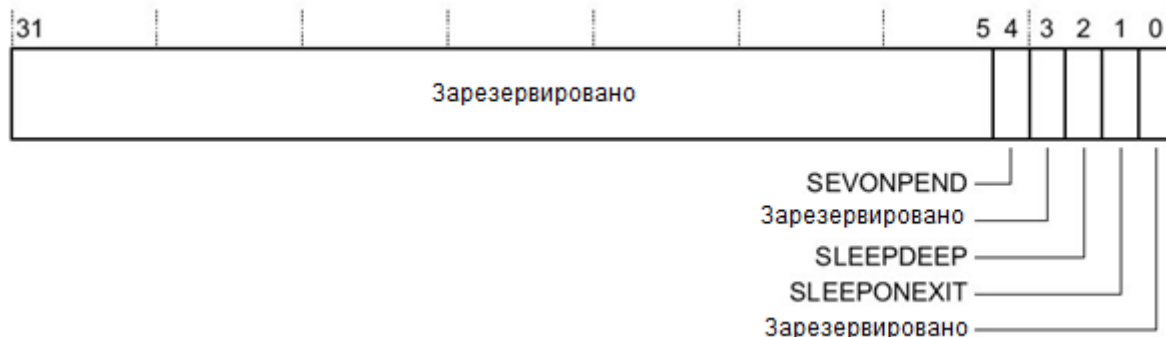
VECTCLRACTIVE (WO) – зарезервировано для целей отладки. При чтении возвращает 0. При записи данных в регистр значение поля должно быть равно 0, в противном случае результат непредсказуем.



### Регистр управления системой

Регистр SCR позволяет определить требования к переходу в и выходу из режима пониженного энергопотребления.

Подробная информация о регистре представлена в Таблица 173. Назначение разрядов регистра представлено на Рис. 102:



**Рис. 102**

SEVONPEND – разрешает или запрещает формирование сигнала события при переводе исключения в состояние ожидания обработки. 0 – выход из режима пониженного энергопотребления по прерыванию могут инициировать только разрешенные прерывания или события; 1 – выход может инициироваться разрешенными событиями и любыми, в том числе запрещенными, прерываниями.

Перевод прерывания в состояние ожидания обслуживания формирует событие, что в свою очередь приводит к выходу процессора из режима пониженного потребления, инициированного инструкцией WFE, либо к регистрации факта события, если эта инструкция еще не выполнялась.

Кроме того, процессор может быть выведен из режима пониженного энергопотребления при поступлении внешнего события, а также после выполнения инструкции SEV.

SLEEPDEEP – определяет режим пониженного энергопотребления процессора: 0 – спящий режим (sleep), 1 – режим глубокого сна (deep sleep).

SLEEPONEXIT – разрешает или запрещает перевод процессора в режим пониженного энергопотребления при выходе из обработчика события в режим выполнения прикладной программы: 0 – не переводить, 1 – переводить.

### Регистр конфигурации и управления

Регистр CCR управляет процессом перехода процессора в режим приложения, а также позволяет запретить или разрешить:

- игнорирование отказов доступа к шине в обработчиках тяжелых отказов и при эскалации отказа по FAULTMASK;
- генерацию исключений при делении на ноль и при доступе по невыровненному адресу;
- доступ к регистру STIR из непривилегированного приложения.

Подробная информация о регистре представлена в Таблица 173. Назначение разрядов регистра представлено на Рис. 103:



**Рис. 103**

STKALIGN определяет режим выравнивания адреса стека при обработке исключений: 0 = выравнивание по границе 4 байт; 1 = по границе 8 байт. При передаче управления на обработчик исключения процессор анализирует бит [9] сохраненного в стеке слова состояния PSR и определяет по нему режим выравнивания стека. При возврате из обработчика процессор использует сохраненный в стеке бит этого слова для восстановления требуемого режима выравнивания.

UNALIGN\_TRP всегда читается как “1”, информируя, что при любом невыровненном доступе к данным, происходит переход на HardFault.

### Регистры приоритета системных обработчиков

Регистры приоритета системных обработчиков SHPR2-SHPR3 позволяют установить уровень приоритета обработки исключений.

Доступ к регистрам осуществляется пословно. Подробная информация о регистре представлена в Таблица 173.

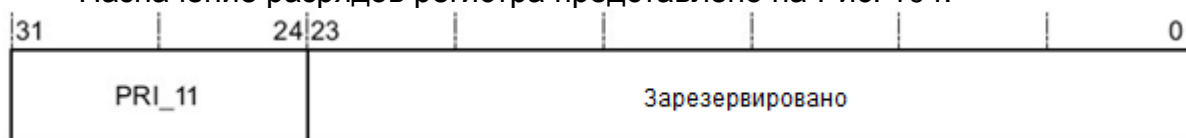
Поля PRI\_N регистров имеют ширину 8 бит, однако в процессоре реализована поддержка доступа только к старшем двум битам [7:6], при чтении данных из младшего бит [5:0] процессор возвращает нули.

**Таблица 174** Поля приоритета обработчиков системных отказов

Обработчик отказа	Поле	Описание регистра
Вызов SVCcall	PRI_11	Регистр №2 приоритета системных обработчиков
Вызов PendSV	PRI_14	Регистр №3 приоритета системных обработчиков
Вызов SysTick	PRI_15	

### Регистр №2 приоритета системных обработчиков

Назначение разрядов регистра представлено на Рис. 104:



**Рис. 104**

PRI\_11 Приоритет системного обработчика 11, вызов SVCcall

### Регистр №3 приоритета системных обработчиков

Назначение разрядов регистра представлено на Рис. 105:



Рис. 105

PRI\_15 Приоритет системного обработчика 15, вызов SysTick

PRI\_14 Приоритет системного обработчика 14, вызов PendSV

### **Рекомендации по программированию блока управления системой**

Необходимо убедиться, что программа использует для обращения к регистрам блока управления системой доступ по корректно выровненным адресам. Обращение ко всем регистрам должно быть выровнено по границе слова.

**Сторожевые таймеры**

**Описание регистров блока сторожевых таймеров**

**Таблица 175**

Базовый Адрес	Название	Описание
0x4005_0000	IWDG	Сторожевой таймер IWDG
Смещение		
0x00	IWDG_KR[15:0]	Регистр Ключа
0x04	IWDG_PR[2:0]	Делитель частоты сторожевого таймера
0x08	IWDG_PRL[11:0]	Регистр основания счета сторожевого таймера
0x0C	IWDG_SR[1:0]	Регистр статуса сторожевого таймера

**Таблица 176**

Базовый Адрес	Название	Описание
0x4004_8000	WWDG	Сторожевой таймер WWDG
Смещение		
0x00	WWDG_CR[7:0]	Регистр управления
0x04	WWDG_CFR[9:0]	Регистр конфигурации
0x08	WWDG_SR[0]	Регистр статуса

**IWDG\_KR**

Номер	15	0
Доступ	W	W
* Значение после сброса	0	0

KEY[15:0]
-----------

**Таблица 177**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16		Зарезервировано
15..0	KEY[15:0]	<b>Значение ключа (только запись, читается 0000h).</b> Эти биты должны перезаписываться программно через определённые интервалы ключевым значением AAAAh, в противном случае сторожевой таймер генерирует сброс, если таймер достиг значения нуля. Запись ключевого значения 5555h разрешает доступ по записи к регистрам IWDG_PR и IWDG_RLR. Запись ключевого значения CCCCCh разрешает работу сторожевого таймера (за исключением, если

		сторожевой таймер разрешается аппаратно битами конфигурации).
--	--	---

### IWDG\_PR

Номер	7	6	5	4	3	2	1	0
Доступ *	U	U	U	U	U	R/W	R/W	R/W
Значение после сброса						0	0	0
	-	-	-	-	-	PR2	PR1	PR0

**Таблица 178**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..3		Зарезервировано
2..0	PR[2:0]	<b>Делитель частоты сторожевого таймера.</b> 000 – делитель на 4 001 – делитель на 8 010 – делитель на 16 011 – делитель на 32 100 – делитель на 64 101 – делитель на 128 110 – делитель на 256 111 – делитель на 256 Чтение и запись этого регистра правомерна только, если бит PVU=0 в регистре IWDG_SR.

### IWDG\_RLR

Номер	11	0
Доступ *	R/W	R/W
Значение после сброса	1	1
	RLR[11:0]	

**Таблица 179**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..12		Зарезервировано
11..0	RLR[11:0]	<b>Значение перезагрузки сторожевого таймера.</b>

		Значение этих битов по доступу защищено с помощью регистра IWDG_KR. Эти биты записываются программно и определяют значение, загружаемое в сторожевой таймер в момент записи значение AAAAh в регистр IWDG_KR. Сторожевой таймер декрементируется начиная с этого значения. Период таймаута сторожевого таймера функция от этого значения и делителя частоты. Чтение и запись этого регистра правомерна только, если бит RVU=0 в регистре IWDG_SR.
--	--	---

### IWDG\_SR

Номер	7	6	5	4	3	2	1	0
Доступ *	U	U	U	U	U	U	R	R
Значение после сброса							0	0
	-	-	-	-	-	-	RVU	PVU

**Таблица 180**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..2		Зарезервировано
1	RVU	<b>Флаг обновления значения сторожевого таймера.</b> Этот бит устанавливается аппаратно и служит признаком того, что обновляется значение сторожевого таймера из регистра перезагрузки. Этот бит сбрасывается, если обновление завершено. Значение регистра перезагрузки может быть обновлено только, если этот бит равен нулю.
0	PVU	<b>Флаг обновления делителя частоты сторожевого таймера.</b> Этот бит устанавливается аппаратно и служит признаком того, что обновляется значение делителя частоты сторожевого таймера. Этот бит сбрасывается, если обновление завершено. Значение регистра делителя частоты может быть обновлено только, если этот бит равен нулю.

### WWDG\_CR

## Спецификация K1986BK234, K1986BK234K

Номер	7	6	5	4	3	2	1	0
Доступ*	R/S	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса	0	1	1	1	1	1	1	1
	WDGA	T6	T5	T4	T3	T2	T1	T0

**Таблица 181**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..18		Зарезервировано
7	WDGA	<b>Бит активации.</b> Этот бит устанавливается программно и очищается только аппаратно при сбросе. Когда WDGA=1, сторожевой таймер может генерировать сброс. 0 – сторожевой таймер отключен 1- сторожевой таймер включен
6..0	T[6:0]	<b>Значение семиразрядного счётчика (от старших разрядов к младшим).</b> Эти биты содержат значение сторожевого таймера, который декрементируется каждые $4096 \times 2^{WDGTB}$ циклов частоты PCLK периферийной шины APB

### WWDG\_CFR

Номер	7	6	5	4	3	2	1	0
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса	0	1	1	1	1	1	1	1
	WDGTB0	W6	W5	W4	W3	W2	W1	W0

Номер	15	14	13	12	11	10	9	8
Доступ*	U	U	U	U	U	U	R/S	R/W
Значение после сброса							0	0
	-	-	-	-	-	-	EWI	WDGTB1

**Таблица 182**

№	Функциональный	Расшифровка функционального имени бита, краткое
---	----------------	---

	ое имя бита	описание назначения и принимаемых значений.
31..10		Зарезервировано
9	EWI	<b>Раннее предупреждающее прерывание.</b> Если бит установлен, то разрешается генерация прерывания при достижении сторожевым таймером значения 40h. Прерывание запрещается только аппаратным сбросом.
8..7	WGTV[1:0]	<b>Делитель частоты сторожевого таймера.</b> 00 – частота таймера (PCLK / 4096) /1 01 – частота таймера (PCLK / 4096) /2 10 – частота таймера (PCLK / 4096) /4 11 – частота таймера (PCLK / 4096) /8
6..0	W[6:0]	<b>Значение окна.</b> Эти биты содержат значение окна, в пределах которого возможна инициализация битов T[6:0] значением в пределах 40h-7Fh. Если происходит инициализация битов в момент T>W, то формируется сброс на выходе RESET. Если таймер достигнет значения T=3Fh, то также формируется сброс.

### WWDG\_SR

Номер	7	6	5	4	3	2	1	0
Доступ	U	U	U	U	U	U	U	R/C
*								
Значение после сброса								0
	-	-	-	-	-	-	-	EWIF

**Таблица 183**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..1		Зарезервировано
0	EWIF	<b>Флаг раннего предупреждающего прерывания.</b> Этот бит устанавливается аппаратно, когда сторожевой таймер достигает значения 40h. Бит очищается программно записью нуля. Запись единицы не влияет. Этот бит также устанавливается, если прерывание запрещено EWI=0.



Предельно-допустимые характеристики микросхемы

Таблица 184 Предельно-допустимые и предельные режимы эксплуатации микросхем

Наименование параметра, единица измерения	Буквенное обозначение параметра	Норма параметра			
		Предельно-допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Напряжение источника питания, В	$U_{CC}$	2,2	3,6	–	4,0
Напряжение источника питания при использовании АЦП и ЦАП, В	$U_{CC}$	2,4	3,6	–	4,0
Напряжение источника питания батарейного домена, В	$U_{CCVB}$	1,8	$U_{CC}+0,3$	–	4,0
Входное напряжение низкого уровня, В, (при работе в цифровом режиме) на выводах: PA, PB, PC, RESET, WAKEUP, JTAG_EN	$U_{IL}$	0	0,8	минус 0,3	–
на выводе: OSC_IN HSE BYPASS=1	$U_{IL\_BHSE}$	0	0,8	минус 0,3	–
Входное напряжение высокого уровня, В, на выводах: PB(4-9), PC (1-4)	$U_{IH}$	2,0	$U_{CC}$	–	$U_{CC}+0,3$
на выводах: PA, PB(0-3,10-14), PC(0,7-15), RESET, WAKEUP, JTAG_EN		2,0	5,25	–	5,3
на выводах: OSC_IN при HSE BYPASS=1	$U_{IH\_BHSE}$	2,0	$U_{CC}$	–	$U_{CC}+0,3$
Выходной ток низкого уровня, мА, (при работе в цифровом режиме) на выводах: PA, PB, PC	$I_{OL}$	минус 6	–	минус 10	–
на выводах: PB (4, 5)		минус 6	–	минус 10	–
Выходной ток высокого уровня, мА, на выводах: PA, PB, PC	$I_{OH}$	–	6	–	10
на выводах: PB (4, 5)		–	3	–	10
Частота следования импульсов тактовых сигналов, МГц	$f_C$	–	36	–	–
Частота следования импульсов тактовых сигналов HSE, МГц при BYPASS=0	$f_{C\_HSE}$	2	16	–	–
при BYPASS=1		–	36	–	–
Частота следования импульсов тактовых сигналов LSE, кГц при BYPASS=0	$f_{C\_LSE}$	32	33	–	–
при BYPASS=1		–	1 000	–	–

Наименование параметра, единица измерения	Буквенное обозначение параметра	Норма параметра			
		Предельно- допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Частота следования импульсов тактовых сигналов PLL, МГц	$f_{C\_PLL}$	2	16	–	–
Длительность фронта/спада входного сигнала, нс, на выводе OSC_IN при HSE BYPASS=1	$\tau_r$ $\tau_f$	–	5	–	–
Емкость нагрузки, пФ, на выводах: PA, PB, PC	$C_L$	–	30	–	–
Время хранения информации, лет, при T=25 °C	$t_{GS}$	25	–	–	–
при T=85 °C		10	–	–	–
при T=125 °C		1	–	–	–
<b>Параметры площадок ЖК дисплея</b>					
Напряжение на входе VLCD, В	$V_{LCD}$	0	2,0	–	$U_{CC}+0,3$
<b>Параметры АЦП</b>					
Напряжение нижней границы внешней опоры АЦП, В	$U_{ADC1\_REF-}$	0	$U_{CCA}-2,4$	–	4,0
Напряжение верхней границы внешней опоры АЦП, В	$U_{ADC0\_REF+}$	2,4	$U_{CCA}$	–	4,0
Диапазон напряжения внешнего опорного источника АЦП, В $U_{REF(ADC)} = U_{ADC0\_REF+} - U_{ADC1\_REF-}$	$U_{REF(ADC)}$	2,4	$U_{CCA}$	–	–
Диапазон напряжения на входе** АЦП, В	$U_{AIN}$	$U_{ADC1\_REF-}$	$U_{ADC0\_REF+}$	минус 0,3	4,0
Частота следования импульсов тактовых сигналов АЦП, МГц	$f_{C\_ADC\_S}$	–	14	–	–
<b>Параметры <math>\Delta\Sigma</math> АЦП</b>					
Частота следования импульсов тактовых сигналов $\Delta\Sigma$ АЦП, МГц	$f_{C\_ADC\_D}$	–	8,196	–	–
Амплитуда входного дифференциального сигнала $\Delta\Sigma$ АЦП, В	$A_{NADC\_D}$	–	1	–	–
Напряжение на входе $\Delta\Sigma$ АЦП, В	$U_{IADC\_D}$	-0,5	+0,5	-0,8	$U_{CC}+0,3$
* - допускается использование отдельного источника для питания АЦП и ЦАП, но при этом его выходное напряжение не должно отличаться от $U_{CC}$ более чем на $\pm 0,3V$					
** - при использовании внутреннего опорного напряжения $U_{ADC1\_REF-} = AGND$ и $U_{ADC0\_REF+} = AU_{CC}$					
<b>П р и м е ч а н и я :</b>					
1 Не допускается одновременное задание двух предельных режимов.					
2 Значения параметров предельных режимов носят рекомендательный характер.					

**Электрические параметры микросхемы**

**Таблица 185** Электрические параметры микросхем при приёмке и поставке

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Выходное напряжение низкого уровня, В, на выводах: PA, PB, PC,	$U_{OL}$	–	0,4	25, 85, минус 80
Выходное напряжение высокого уровня, В, на выводах: PA, PB, PC при $2,2 \leq U_{CC} \leq 3,6$ В	$U_{OH}$	1,6	–	25, 85, минус 40
при $3,0 \leq U_{CC} \leq 3,6$ В		2,4	–	
Уровень напряжения питания для срабатывания схемы генерации сброса, В	$U_{BOR}$	1,8	2,1	25, 85, минус 40
Статический ток потребления в режиме покоя (регулятор напряжения включен), мкА	$I_{CCS1}$	–	100	25, 85, минус 40
Статический ток потребления в режиме покоя (регулятор напряжения выключен), мкА	$I_{CCS2}$	–	10	25, 85, минус 40
Динамический ток потребления, мА	$I_{OCC1}$	–	21	25, 85, минус 40
Динамический ток потребления, мА	$I_{OCC2}$	–	0,4	25, 85, минус 40
Входной ток утечки высокого уровня, мкА, (при работе в цифровом режиме) на выводах: PB(4-9), PC (1-4)	$I_{ILH}$	минус 1,0	1,0	25, 85, минус 40
Входной ток утечки высокого уровня, мкА, (при работе в цифровом режиме) на выводах: PA, PB(0-3,10-14), PC(0,5-15), RESET, WAKEUP	$I_{ILH}$	минус 1,0	1,0	25, 85, минус 40
Входной ток утечки высокого уровня, мкА, (при работе в цифровом режиме) на выводе: JTAG_EN	$I_{ILH1}$	минус 180	180	25, 85, минус 40
Входной ток утечки высокого уровня, мкА, на выводе: OSC_IN	$I_{ILH\_BHSE}$	минус 40	40,0	25, 85, минус 40
Входной ток утечки низкого уровня, мкА, (при работе в цифровом режиме) на выводах: PA, PB, PC, RESET, WAKEUP, JTAG_EN	$I_{ILL}$	минус 1,0	1,0	25, 85, минус 40
Входной ток утечки низкого уровня, мкА, на выводе OSC_IN	$I_{ILL\_BHSE}$	минус 40	40,0	25, 85, минус 40

## Спецификация K1986BK234, K1986BK234K

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °C
		не менее	не более	
Выходная частота LSI RC-генератора, кГц	$f_{O\_LSI}$	10	60	25, 85, минус 40
Выходная частота HSI RC-генератора, МГц	$f_{O\_HSI}$	6	10	25, 85, минус 40
Выходная частота PLL, МГц максимальная минимальная	$f_{O\_PLL}$	36 –	– 2	25, 85, минус 40
<b>Параметры АЦП последовательного приближения</b>				
Разрядность АЦП	$E_{NADC\_S}$	12	-	25, 85, минус 40
Дифференциальная нелинейность, единица младшего разряда	$E_{DLADC\_S}$	минус 1	2	25, 85, минус 40
Интегральная нелинейность, единица младшего разряда	$E_{ILADC\_S}$	минус 3	3	25, 85, минус 40
Ошибка смещения, единица младшего разряда	$E_{OFFADC\_S}$	минус 6	6	25, 85, минус 40
Ошибка усиления, %	$E_{GAINADC}$	минус 1	1	25, 85, минус 40
<b>Параметры площадок ЖК дисплея</b>				
Точность выходного напряжения на выводах PA, PB6-BP14, PC5-PC13 при формировании эквивалента $U_{CC}$ , $U_{CC} - VLCD$ , В	$U_{OH1}$	минус 0,1	0,1	25, 85, минус 40
Точность выходного напряжения на выводах PA, PB6-BP14, PC5-PC13 при формировании эквивалента $2 \cdot U_{CC}/3$ и $2 \cdot (U_{CC} - VLCD)/3$ , в процентах %	$U_{OH2}$	минус 4	0	25, 85, минус 40
Точность выходного напряжения на выводах PA, PB6-BP14, PC5-PC13 при формировании эквивалента $U_{CC}/3$ и $(U_{CC} - VLCD)/3$ , в процентах %	$U_{OH3}$	минус 3	2	25, 85, минус 40
Точность выходного напряжения на выводах PA, PB6-BP14, PC5-PC13 при формировании эквивалента $U_{CC}/2$ и $(U_{CC} - VLCD)/2$ , в процентах %	$U_{OH4}$	минус 3	2	25, 85, минус 40

## Спецификация K1986BK234, K1986BK234K

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Точность выходного напряжения на выводах PA, PB6-BP14, PC5-PC13 при формировании эквивалента "Общий" и VLCD, В	$U_{OH5}$	минус 0,1	0,1	25, 85, минус 40
Частота смены напряжений на выходах PA, PB6-BP14, PC5-PC13, Гц	$F_{VOUT}$	60	1000	25, 85, минус 40
<b>Параметры <math>\Delta\Sigma</math> АЦП</b>				
Соотношение сигнал шум, дБ усиление 0дБ, $A_{NADC\_D} = 1В$ , $f_{NADCO} = 4КГц$	$SNR_{D\_0}$	77	–	25, 85, минус 40
Соотношение сигнал шум, дБ усиление +6дБ, $A_{NADC\_D} = 0,5В$ , $f_{NADCO} = 4КГц$	$SNR_{D\_6}$	74	–	25, 85, минус 40
Соотношение сигнал шум, дБ усиление +12дБ, $A_{NADC\_D} = 0,25В$ , $f_{NADCO} = 4КГц$	$SNR_{D\_12}$	71	–	25, 85, минус 40
Соотношение сигнал шум, дБ усиление +18дБ, $A_{NADC\_D} = 0,125В$ , $f_{NADCO} = 4КГц$	$SNR_{D\_18}$	70	–	25, 85, минус 40
Ошибка усиления предусилителя, дБ	$GAIN_{ERR}$	–	0.25	25, 85, минус 40
Выходное напряжение VR_1V, В	$U_{OBIAS}$	0.96	1,01	25, 85, минус 40

Микросхемы должны быть устойчивы к воздействию статического электричества с потенциалом не менее 2 000 В.

**Типовые зависимости основных электрических параметров микросхемы от режимов и условий эксплуатации**

1. Зависимость динамического тока потребления от температуры при  $f_C = 36$  МГц.
2. Зависимость динамического тока потребления от напряжения питания при  $f_C = 36$  МГц
3. Зависимость статического тока потребления в режиме покоя (регулятор напряжения выключен) от температуры при разном напряжении питания
4. Зависимость тока потребления батарейного домена от температуры при разном напряжении питания
5. Зависимость выходного напряжения высокого уровня от выходного тока высокого уровня при напряжении питания 2,2 и 3,0 В
6. Зависимость выходного напряжения низкого уровня от выходного тока низкого уровня при напряжении питания 2,2 и 3,0 В

Габаритный чертеж микросхемы

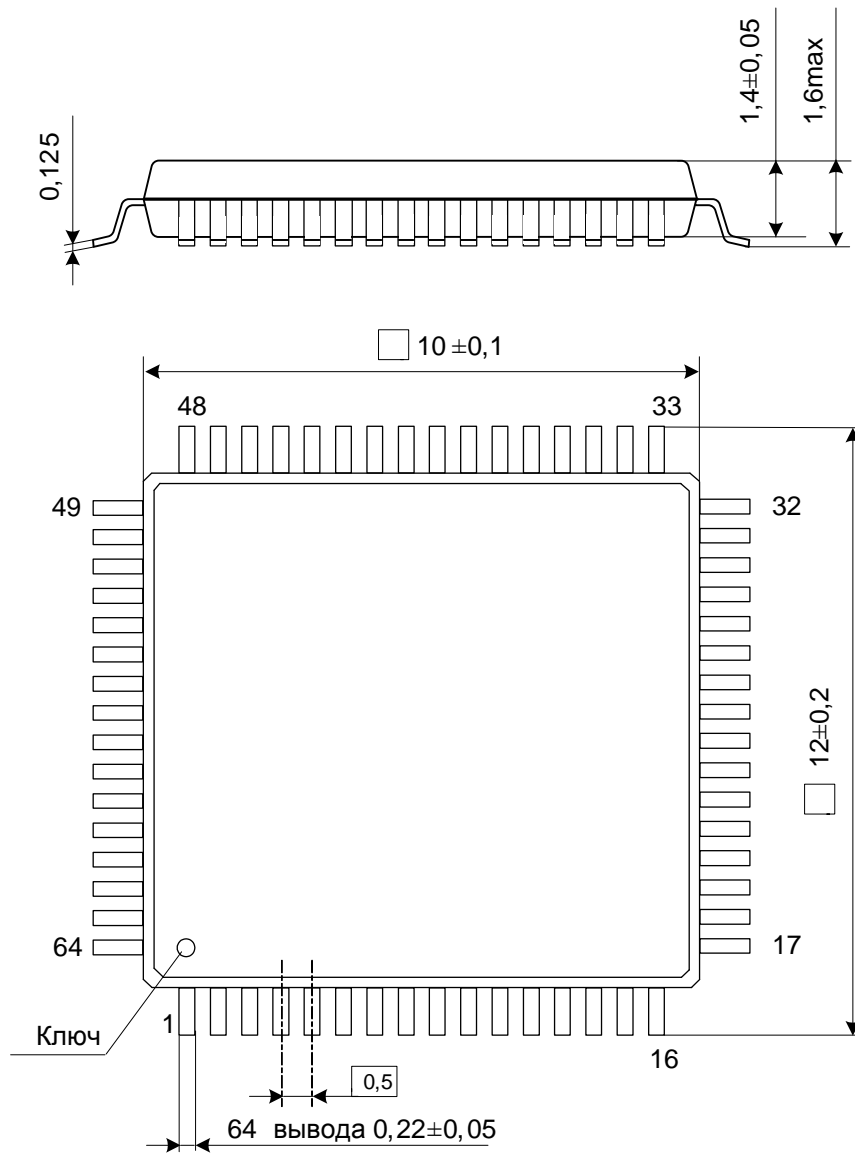


Рис. 106 Корпус LQFP64

**Информация для заказа**

<b>Обозначение</b>	<b>Маркировка</b>	<b>Тип корпуса</b>	<b>Температурный диапазон</b>
K1986BK234	MDR32F23QI	LQFP64	минус 40 – 85 °С
K1986BK234K	MDR32F23QC	LQFP64	0 – 70 °С



**Лист регистрации изменений**

<b>№ п/п</b>	<b>Дата</b>	<b>Версия</b>	<b>Краткое содержание изменения</b>	<b>№№ изменяемых листов</b>
1	30.07.13	1.0.0	Введена впервые	
2	28.11.2013	1.1.0	Перевод надписей рисунков на русский язык	По тексту
3	15.01.2014	1.2.0	Исправление названий разделов	26, 27
4	21.01.2014	1.3.0	Внесение исправлений в таблицу 3	18
5	31.01.2014	1.4.0	Исправлена разрядность CNT, PSG, ARR	145, 146
6	04.03.2014	1.4.1	В таблице 170 удалены 2 последние строки	255
7	28.05.2014	1.5.1	Добавлено примечание 2 к таблице 184	273
8	25.06.2014	1.6.0	Испралена маркировка. Исправлено обозначение спецификации	1, 280
9	08.07.2014	1.7.0	Добавлено значение статического электричества	277
10	09.07.2014	1.8.0	Исправлен габаритный чертеж	281
11	12.01.2015	1.9.0	Добвлены типовые схемы включения для учета электроэнергии	56, 57
12	26.06.2015	1.10.0	Изменено обозначение микросхем	По тексту
13	01.10.2015	1.11.0	Изменено обозначение микросхем	По тексту