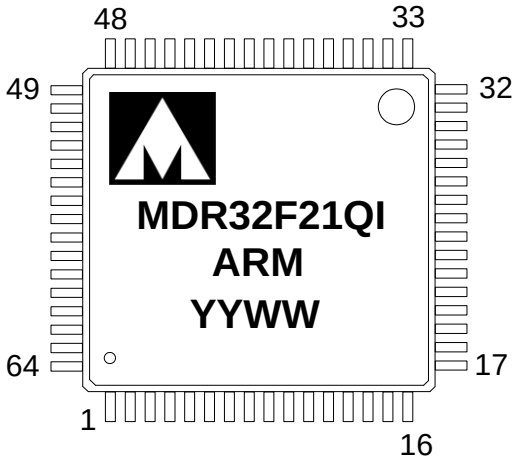
Микросхема контроллера для трехфазного электросчетчика К1986ВК234, К1986ВК234К



YY – год выпуска

WW – неделя выпуска

Основные параметры микросхемы

- Напряжение источника питания 3,0 – 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 LQFP64.

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 128 Кбайт;
- встроенное ОЗУ размером 16 Кбайт.

Питание и тактовая частота:

- внешнее питания 3,0…3,6 В;
- встроенный регулятор напряжения на 1,8 В для питания ядра;
- встроенные схемы контроля питания;
- встроенный домен с батарейный питанием;
- встроенный подстраиваемый RC генератор 8 МГц;
- встроенный подстраиваемый RC генератор 40 кГц;
- внешний осциллятор 2…16 МГц;
- внешний осциллятор 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-разрядный ∑Δ АЦП (до 7 каналов);
- 12-разрядный АЦП (до 8 каналов);
- измеряемый диапазон от 0 до 3,6 В;
- температурный сенсор.

Периферия:

- контроллер прямого доступа в память с функциями передачи Периферия-Память, Память-Память;
- контроллеры интерфейсов UART, SPI;
- до 39 пользовательских линий ввода/вывода;
- два блока 16-разрядных таймеров с 4-мя каналами захвата событий и ШИМ;
- сторожевой таймер;
- блок подсчета CRC с изменяемым полиномом.

Режим отладки:

- последовательный отладочный интерфейс SWD.

Содержание

Введение

Микроконтроллер этой серии является микроконтроллером со встроенной Flash памятью программ и построен на базе низкопотребляющего процессорного RISC ядра ARM Cortex-M0. Микроконтроллер работает на тактовой частоте до 36 МГц и содержит 128 Кбайт Flash памяти и 16 Кбайт ОЗУ. Микроконтроллер включают в себя развитую периферию для построения счетчиков электроэнергии 1- и 3-х фазных сетей. Периферия включает в себя 7 каналов для 3-фазной сети (или 3 канала для 1-фазной сети) 24-битных независимых ∑Δ АЦП. Каждый канал АЦП имеет предусилитель, фазовую подстройку (для коррекции фазы не хуже 0,1), а также аппаратный блок для вычисления среднеквадратического значения сигнала. Каждый канал 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 также для измерения температуры или захвата внешнего сигнала. В состав микроконтроллера входит 2 UART и 1 SPI интерфейс. Микроконтроллер содержит два 16-ти разрядных таймера с 4 каналами схем захвата и ШИМ с функциями формирования «мертвой зоны» и аппаратной блокировки. Так же микроконтроллер содержит системный 24-разрядный таймер и два сторожевых таймера.

Встроенные RC генераторы HSI (8 МГц) и LSI (40 кГц), 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. Контроллер DMA позволяет ускорить обмен информацией между ОЗУ и периферией без участия процессорного ядра.

Встроенный регулятор для формирования питания внутренний цифровой части формирует напряжения 1,8 В и не требует дополнительных внешних элементов. Таким образом, для работы микроконтроллера достаточно одного внешнего напряжения питания в диапазоне от 3,0 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 также работают часы реального времени. Встроенные детекторы напряжения питания могут отслеживать уровень внешнего основного питания, уровень напряжения питания на батарее. Аппаратные схемы сброса по просадке питания позволяют исключить сбойную работу микросхемы при выходе уровня напряжения питания за допустимые пределы.

Основные характеристики

Корпус	64 вывода
Ядро	ARM Cortex-M0
ПЗУ	128 Кбайт Flash
ОЗУ	16 Кбайт
Питание	3,0…3,6 В
Частота	36 МГц
Температура	-40…+85°С
User IO	39
UART	2
SPI	1
∑Δ АЦП	7
SAR ADC	1
Формирователь опоры	1

Структурная схема

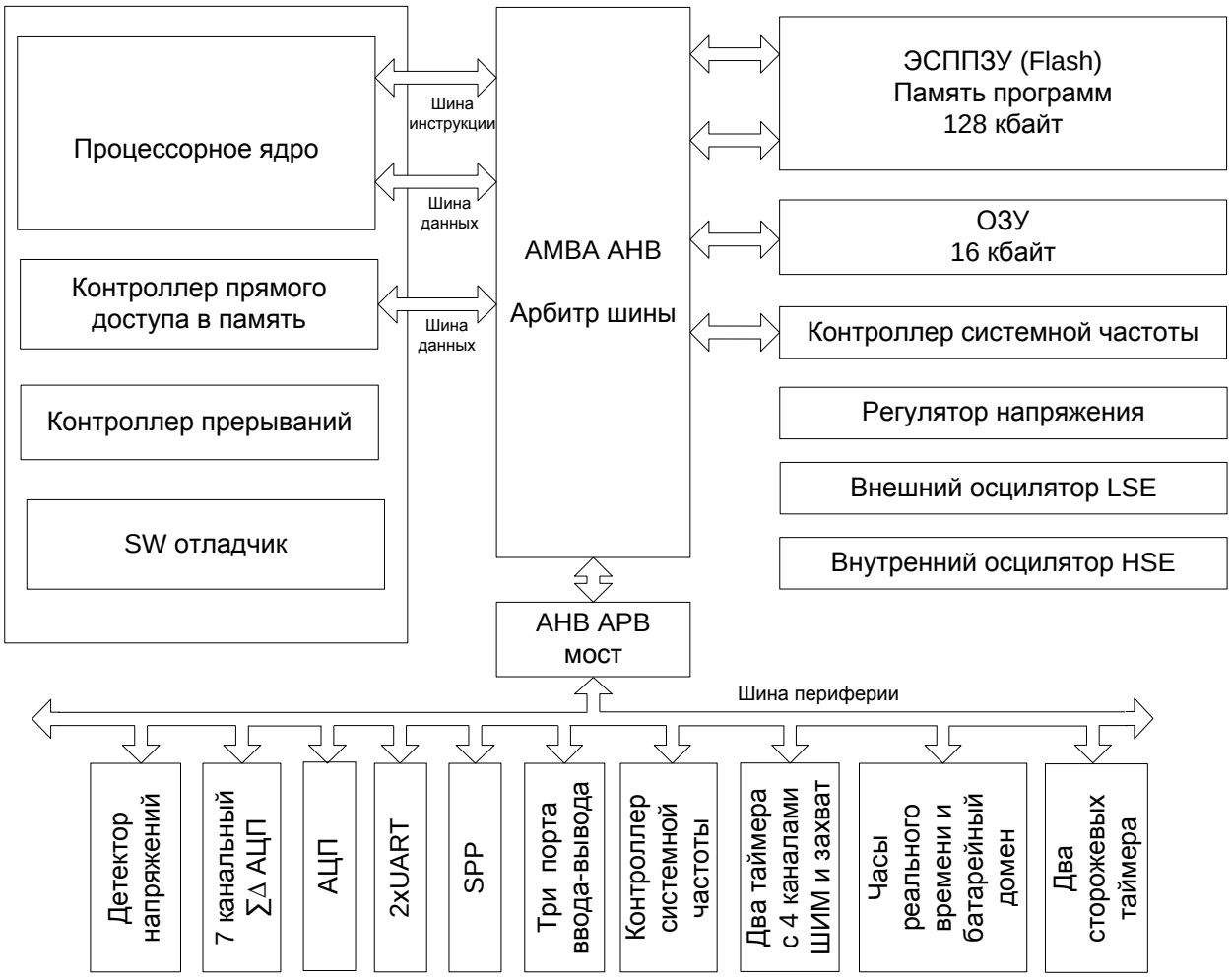


Рисунок 1. Структурная схема микросхемы

Описание выводов

Таблица 1. Описание выводов

Вывод	№ вывода корпуса	Дополнительные функции вывода			
		Аналог.	Основ.	Альтер.	Переопр.
Порт A					
PA0	8	–	TMR0_CH1	–	–
PA1	9	–	TMR0_CH1N	–	–
PA2	10	–	TMR0_CH2	–	–
PA3	11	–	TMR0_CH2N	–	–
PA4	12	–	TMR0_CH3	–	–
PA5	32	–	TMR0_CH3N	–	–
PA6/SWCLKTCK	38	–	TMR0_CH4	–	–
PA7/SWDIO	39	–	TMR0_CH4N	–	–
PA8	40	–	TMR0_ETR	–	–
PA9	41	–	TMR0_BLK	–	–
PA10	42	–	EXT_INT0	–	–
PA11	47	–	–	–	–
PA12	48	–	SSP_FSS	–	–
PA13	2	–	SSP_CLK	–	–
PA14	3	–	SSP_RXD	–	–
PA15	4	–	SSP_TXD	–	–
Порт B					
PB0/MODE0	34	–	UART0_TXD	–	–
PB1	35	–	UART0_RXD	–	–
PB2	36	–	nSIROUT0	–	–
PB3	37	–	nSIRIN0	–	–
PB4	15	OSC_IN32	nUART0DTR	–	–
PB5	16	OSC_OUT32	nUART0RTS	–	–
PB6	24	ADC7	nUART0RI	EXT_INT1	–
PB7	25	ADC6	nUART0DCD	EXT_INT2	–
PB8	26	ADC5	nUART0DSR	TMR1_ETR	–
PB9	27	ADC4	nUART0CTS	TMR1_BLK	–
PB10	45	–	TMR1_CH2	–	–
PB11	46	–	TMR1_CH2N	–	–
PB12	5	–	TMR1_CH3	–	–
PB13	6	–	TMR1_CH3N	–	–
PB14	7	–	TMR1_CH4	–	–
Порт C					
PC0/MODE1	43	–	UART1_TXD	–	–
PC1	20	ADC3	UART1_RXD	–	–
PC2	21	ADC2	TMR1_CH1	–	–
PC3	22	ADC1_REF+	TMR1_CH1N	–	–
PC4	23	ADC0_REF–	EXT_INT1	–	–
PC5	30	–	EXT_INT2	–	–
PC6	31	–	TMR1_ETR	–	–
PC7	44	–	TMR1_BLK	–	–
Порт АЦП					
VR_1V	50				
I0P	51				
I0N	52				
V0P	53				
V0N	54				

Вывод	№ вывода корпуса	Дополнительные функции вывода			
		Аналог.	**Основ.**	**Альтер.**	**Переопр.**
I3P	63				
I3N	64				
I1P	55				
I1N	56				
V1P	57				
V1N	58				
I2P	59				
I2N	60				
V2P	61				
V2N	62				
Системное управление					
RESET	33	Сигнал внешнего сброса			
WAKEUP	17	Сигнал внешнего выхода из режима Standby			
COV_DET	17				
OSC_IN	28	Вход генератора HSE			
OSC_OUT	29	Выход генератора HSE			
Питание					
UCC	13	Питание 3,0…3,6			
AGND	49				
BUCC	14	Батарейное питание 1.8…3,6 В			
GND	19				
AUCC	1	Аналоговое питание ∑ΔАЦП 2,4…3,6 В			
Выводы для тестирования и исследования					
JTAG_EN	18				

Расположение выводов в корпусе

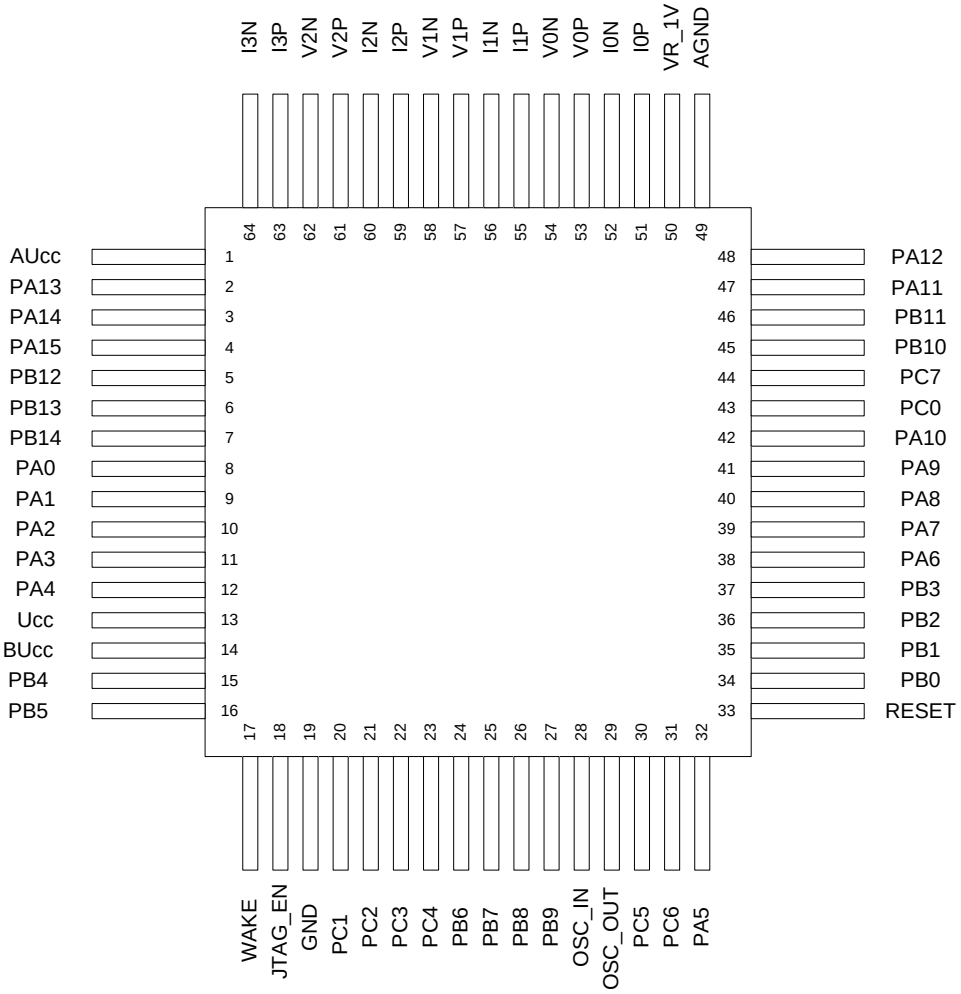


Рисунок 2. Микросхема в 64-выводном пластиковом корпусе LQFP64

Система питания

Микроконтроллер этой серии имеет несколько типов выводов питания.

Ucc выводы: Основное питание микросхемы, включает питание пользовательских выводов, встроенного регулятора напряжения, PHY, PLL, генераторов, компаратора и АЦП последовательного приближения.

BUcc вывод: Питание батарейного домена используется при отсутствии основного питания Ucc для питания батарейного домена и LSE генератора. Переключение с основного питания на батарейное происходит автоматически при снижении уровня Ucc ниже 2,0 В. Переключение с батарейного питания на основное происходит автоматически спустя примерно 4 мс после превышения уровнем Ucc значения 2,0 В. Если в системе не требуется батарейного питания, вывод BUcc должен быть объединен с Ucc.

AUcc выводы: 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 Ucc, но при этом на печатной плате должны быть применены меры по снижению наводки помех.

GND выводы: Основная земля питания.

AGND выводы: Земля аналогового питания AUcc. Данные выводы должны соединяться с GND, но при этом на печатной плате должны быть применены меры по снижению наводки помех.

Структурная схема подачи питания

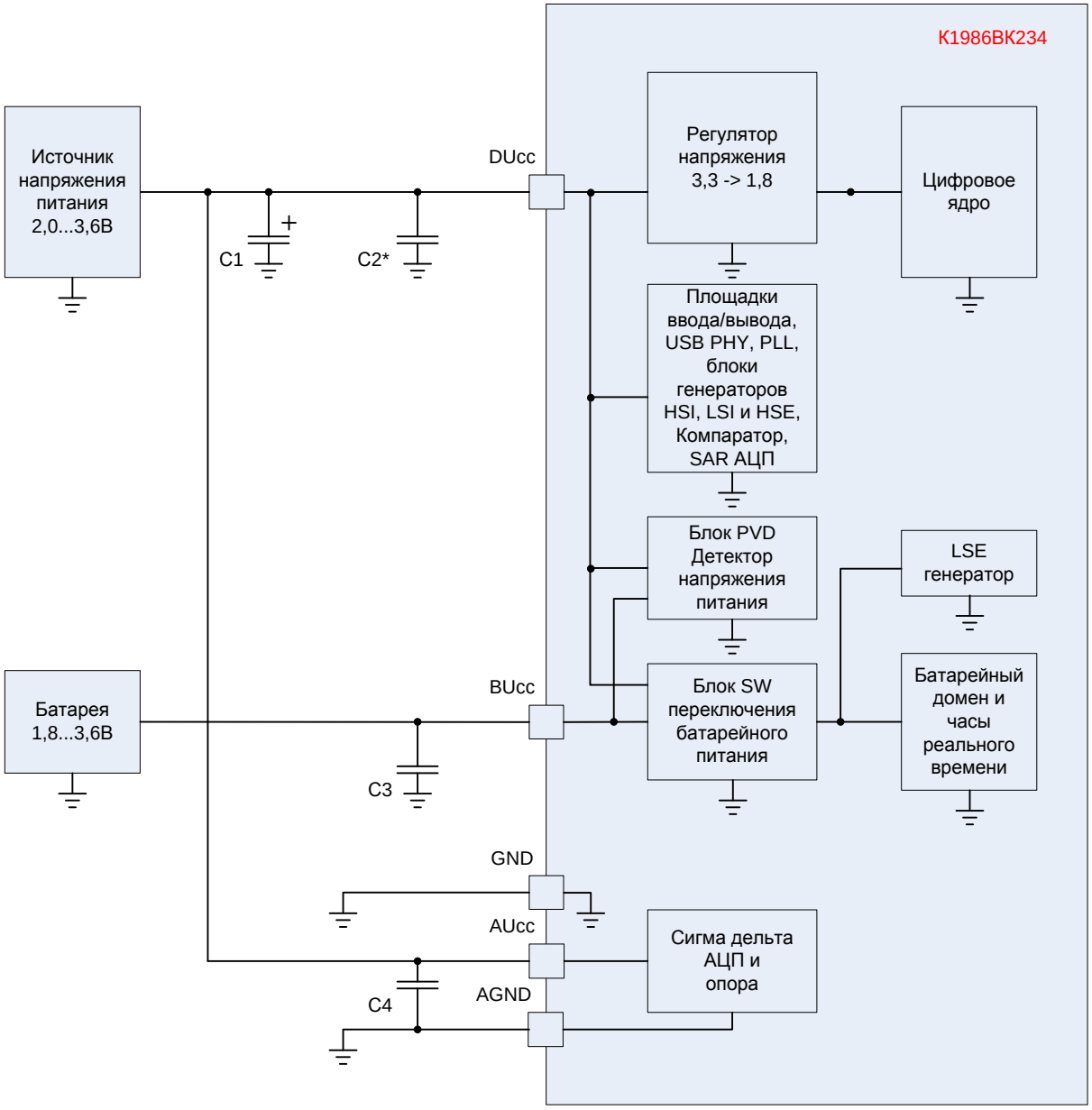


Рисунок 3. Структурная схема подачи питания

Примечание:

1. Конденсатор C1 = 22 мкФ, C2 = C3 = C4 = C5 = 0,1 мкФ.

   * - конденсаторы должны быть установлены у каждого вывода питания.

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 3,6 В;

Микроконтроллер имеет встроенный детектор напряжения питания, подробнее см. раздел «Детектор напряжения питания».

Схема сброса при включении и выключении основного питания

При включении питания, пока питание Ucc не превысило уровень Upor (2,0 В) вырабатывается внутренний сигнал сброса POR для цифровой части. После превышения уровня Upor, сигнал POR выдается еще на протяжении tpor (~4 мс), для того чтобы гарантировано установилось напряжение питания, после чего сигнал POR снимается и схема может начать работать.

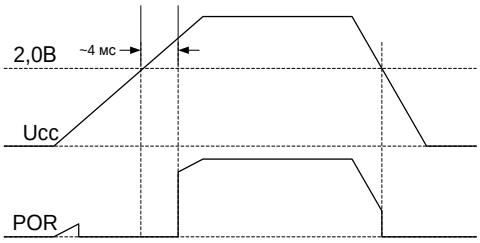


Рисунок 4. Диаграмма формирования сброса микроконтроллера

При снижении напряжения питания Ucc ниже уровня Upor сигнал POR вырабатывается без задержки.

Сигнал POR так же служит для переключения питания батарейного домена между BUcc и Ucc.

При включении основного напряжения питания Ucc автоматически включается встроенный регулятор напряжения для формирования напряжения питания цифрового ядра. В ходе работы микроконтроллера встроенный регулятор может быть отключен.

Микроконтроллер так же может быть сброшен внешним сигналом сброса RESET или внутренними сигналами сброса сторожевых таймеров или программным сбросом. При этом сигнал сброса формируется специальной схемой сброса, содержащий фильтр «иголок» по сигналу сброса и одновибратор для увеличения длительности сигнала сброса.

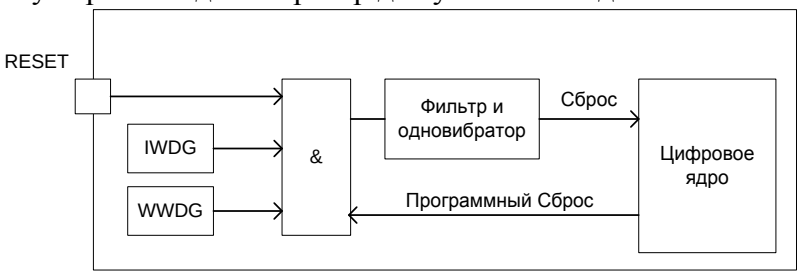


Рисунок 5. Блок-схема формирования сброса микроконтроллера

При приходе импульсов сброса длительностью менее tminreset, эти импульсы отфильтровываются и не приводят к сбросу процессора. Если длительность импульса больше **tmaxreset** вырабатывается сигнал сброса. При этом длительность сформированного сигнала сброса будет не менее **treset**.

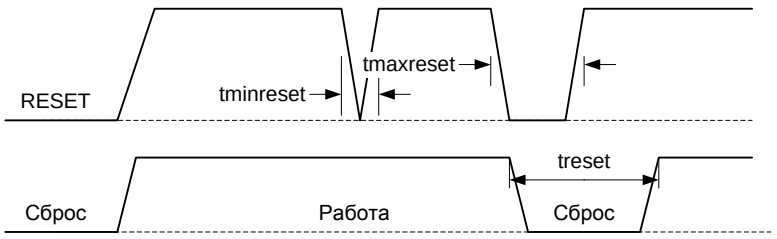


Рисунок 6. Диаграмма фильтрации помех при формировании сброса

Организация памяти

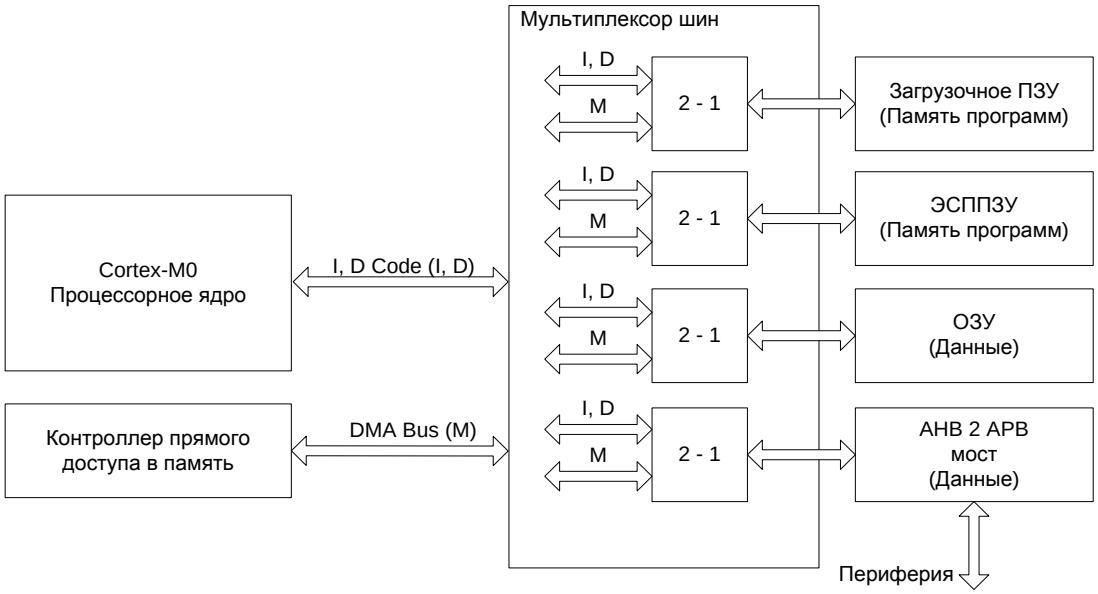


Рисунок 7. Структурная схема организации памяти

Процессорное ядро имеет три системные шины:

I, D Code – шина выборки инструкций и данных.

Также в микроконтроллере реализован контроллер прямого доступа в память (DMA), осуществляющий выборку через шину DMA Bus.

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

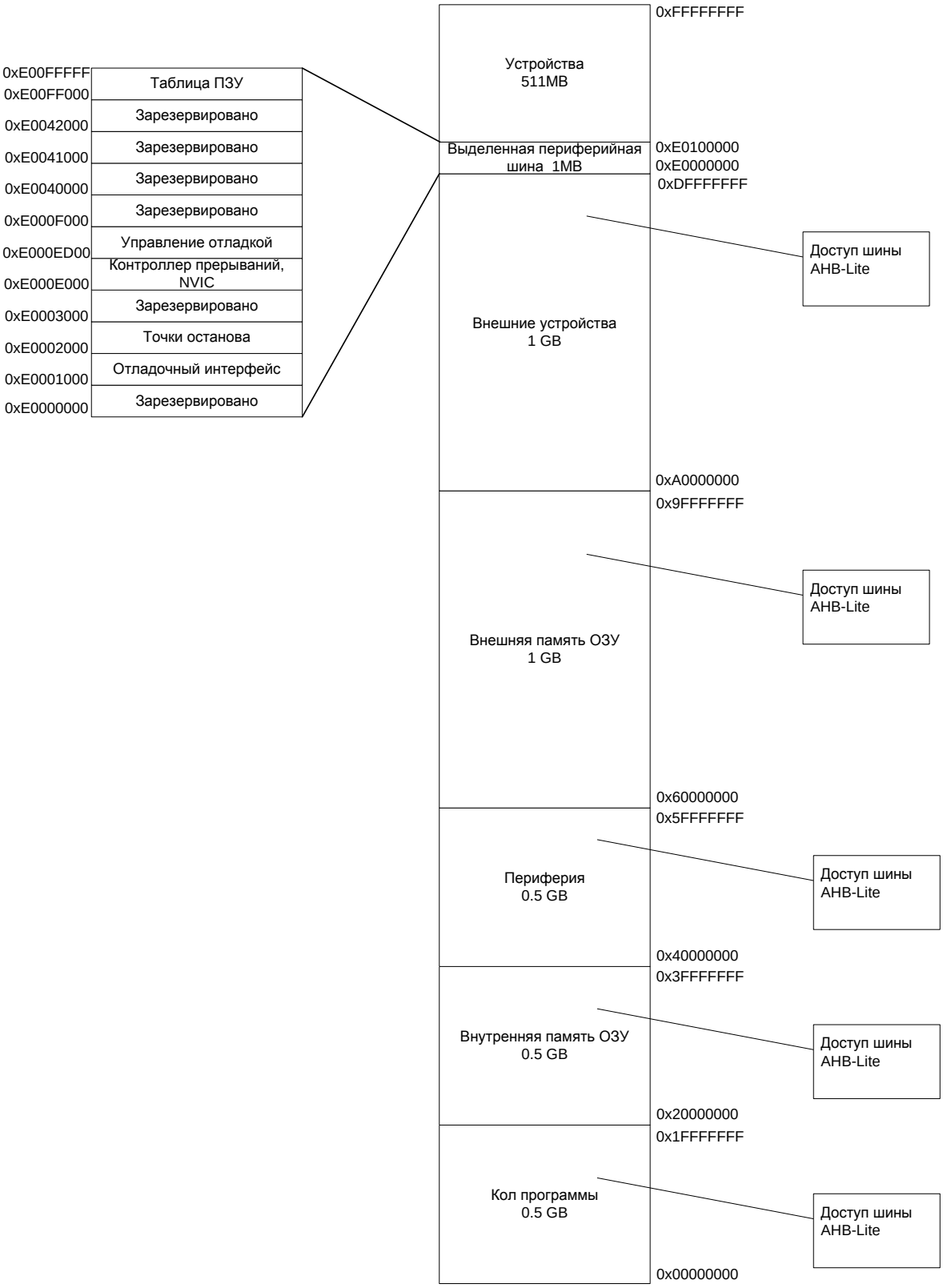


Рисунок 8. Структура адресного пространства микроконтроллера

Базовые адреса процессора

Таблица 2. Базовые адреса процессора

Адрес	Размер	Блок	Примечание
Память программ			
0x0000_0000		BOOT ROM	Загрузочная программа FPOR=0
0x0000_0000		EEPROM	Область Flash памяти программ с пользовательской программой FPOR=1
Память данных			
0x2000_0000		SYSTEM RAM	Область внутреннего ОЗУ
Периферия			
0x4000_0000		SPI	Регистры контроллера интерфейса SSP
0x4000_8000		UART1	Регистры контроллера интерфейса UART1
0x4001_0000		UART2	Регистры контроллера интерфейса UART2
0x4001_8000		EEPROM_CNTRL	Регистры контроллера Flash памяти программ
0x4002_0000		RST_CLK	Регистры контроллера сигналов тактовой частоты
0x4002_8000		DMA	Регистры контроллера прямого доступа в память
0x4004_0000		ADC	Регистры управления АЦП
0x4004_8000		WWDT	Регистры контроллера сторожевого таймера WWDT
0x4005_0000		IWDT	Регистры контроллера сторожевого таймера IWDT
0x4005_8000		POWER	Регистры детектора напряжения питания
0x4006_0000		BKP	Регистры доступа и управления батарейным доменом
0x4006_8000		ADCIU	Регистры управления ∑Δ АЦП
0x4007_0000		TIMER1	Регистры управления Таймер 1
0x4007_8000		TIMER2	Регистры управления Таймер 2
0x4008_0000		PORTA	Регистры управления порта A
0x4008_8000		PORTB	Регистры управления порта B
0x4009_0000		PORTC	Регистры управления порта C
0x4009_8000		CRC	Регистры управления аппаратного блока вычисления CRC
SYSTEM REGION			
0xE000_0000			Системные регистры процессор ARM Cortex-M0

Загрузочное ПЗУ и режимы работы микроконтроллера

После включения питания и снятия внутренних (POR) и внешних (RESET) сигналов сброса, микроконтроллер начинает выполнять программу из информационной загрузочной области FLASH BOOT ROM. 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, который может быть сброшен только при отключении основного питания Ucc. После перезапуска микроконтроллера уровни на выводах MODE[1:0] не влияют на режим функционирование микроконтроллера, если установлен бит FPOR

В пользовательской программе выводам PB[0], PC[0] пользователем могут присваиваться функции самостоятельно.

Таблица 3. Режимы работы микроконтроллера

MODE [1:0]	Режим	Стартовый адрес/ таблица векторов прерываний	Описание
00	Микроконтроллер без отладки	0x----_----	Процессор начинает выполнять программу из внутренней FLASH памяти программ. При этом отладочный интерфейс Serial Wire отключен
01	Микроконтроллер в режиме отладки	0x----_----	Процессор начинает выполнять программу из внутренней FLASH памяти программ. При этом разрешается работа отладочного интерфейса Serial Wire
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 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, но различные порты ввода/вывода (см. таблицу ниже).

Таблица 4

Режим	**Rx**	**Tx**
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 (EERPOM, ROM, 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.
Количетво бит данных – 8.
Четность – нет.
Количество Stop 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:

Команды UART загрузчика

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 Rx 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. Команда CMD_SYNC

Код команды	CMD_SYNC = 0x00
ASCII символ, соответствующий коду команды	нет
Количество параметров команды	0
Формат команды:	
Master: Выдает код команды CMD_SYNC.	Slave: Если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды

Команда CMD_CR

Выдача приглашения Master-у.

Таблица 7. Команда CMD_CR

Код команды	CMD_CR = 0x0D
ASCII символ, соответствующий коду команды	нет
Количество параметров команды	0
Формат команды:	
Master: Выдает код команды CMD_CR.	Slave: 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. Команда CMD_BAUD

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: Если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды
Master: Выдает параметр	Slave: Если параметр принят с ошибками, то выдает код ошибки ERR_CHN или ERR_BAUD 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. Команда CMD_LOAD

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: Если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды.
Master: Выдает параметр 1.	Slave: Если хотя бы один из параметров принят с ошибками, то выдает код ошибки ERR_CHN и завершает обработку текущей команды
Master: Выдает параметр 2.	Slave: Если хотя бы один из параметров 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. Команда CMD_VFY

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: Если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды
Master: Выдает параметр 1	Slave: Если хотя бы один из параметров принят с ошибками, то выдает код ошибки ERR_CHN и завершает обработку текущей команды
Master: Выдает параметр 2	Slave: Если хотя бы один из параметров 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. Команда CMD_RUN

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: Если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды
Master: Выдает параметр.	Slave: 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 свое выполнение.

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 определяет тип ошибки.

После выдачи сообщения об ошибке загрузчик переходит в режим ожидания следующей команды, поэтому Master после получения такого сообщения должен прекратить 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 128 Кбайт основной памяти программ и 8 Кбайта информационной памяти. В обычном режиме (бит CON = 0, регистр EEPROM_CMD) 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, чтобы процессорное ядро могло получать новые инструкции на больших частотах в микроконтроллере реализуется Flash память с физической организацией 128К 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 что бы 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 таблице приведены характеристики необходимой паузы для работы Flash памяти программ.

Таблица 12. Характеристики паузы для работы Flash-программ

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 ROM) с помощью управляющего бита INF_REMAP (расположен в батарейном домене).

В режиме программирования возможны следующие операции как с основной (бит IFREN = 0, регистр EEPROM_CON), так и с информационной (бит IFREN = 1) памятью:

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

Структура памяти представлена на рисунке.

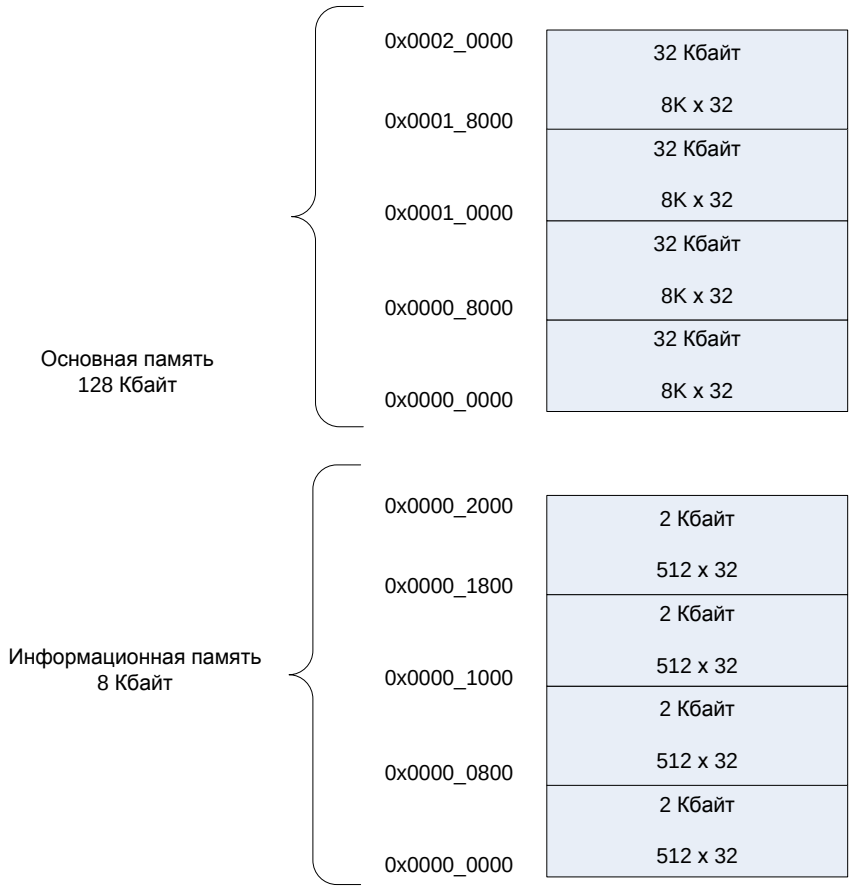


Рисунок 9. Структура памяти

Стирание блока памяти размером 2 Кбайт или 32 Кбайт.

Стирание памяти возможно только в режиме программирования. Для стирания всей памяти надо установить необходимое значение в бит IFREN (1 – для информационной памяти и 0 – для основной памяти), затем установить биты XE, MAS1 и ERASE в единицу, и спустя время tnvs = 10 мкс установить бит NVSTR в единицу. Полное стирание памяти длиться время tme = 40 мс. Спустя это время необходимо очистить бит ERASE, и спустя время tnvh1 = 100 мкс очистить биты XE, MAS1 и NVSTR. Последующие операции с память можно выполнять спустя время trcv = 1 мкс. Временная диаграмма стирания памяти представлена на рисунке.

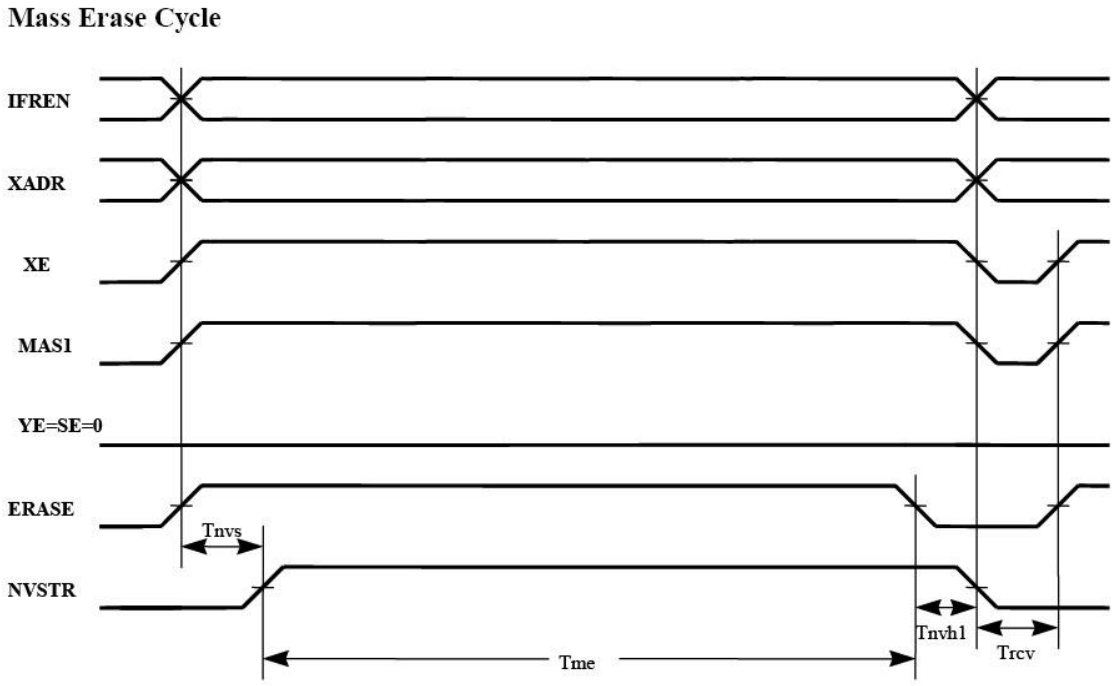


Рисунок 10. Временная диаграмма стирания памяти

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 tnvs = 10 мкс установить бит NVSTR в единицу. Стирание страницы памяти длится время terase = 40 мс. Спустя это время необходимо очистить бит ERASE, и спустя время tnvh = 5 мкс очистить биты XE и NVSTR. Последующие операции с памятью можно выполнять спустя время trcv = 1 мкс. Временная диаграмма стирания страницы памяти представлена на рисунке.

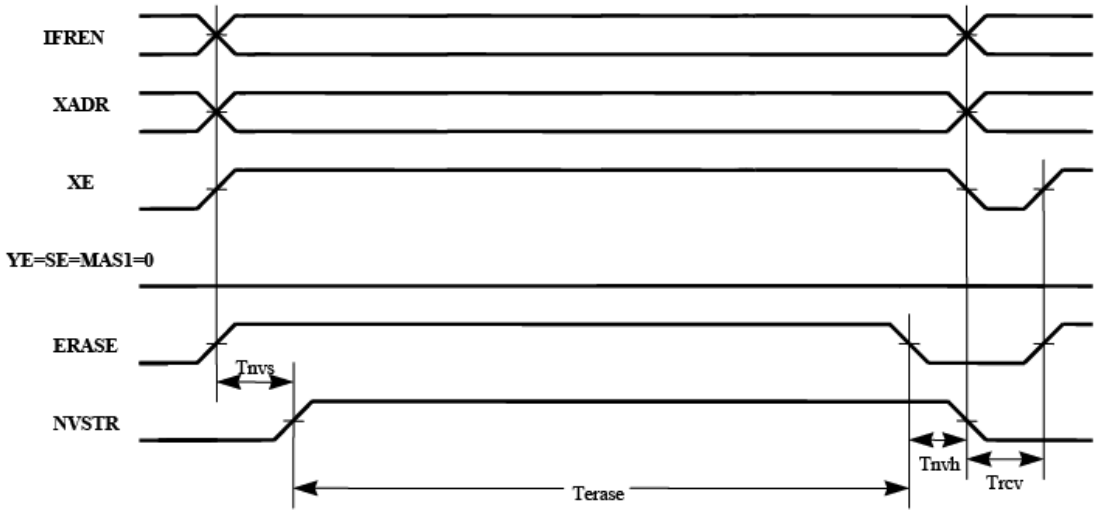


Рисунок 11. Временная диаграмма стирания страницы памяти

Запись 32-х битного слова в память.

Запись в память возможна 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 tnvs = 5 мкс установить бит NVSTR в единицу. Спустя время tpgs = 10 мкс установить бит YE в единицу. Запись в память длиться время tprog = 40 мкс. Спустя это время необходимо очистить бит YE, и спустя время tadh = 20 нс установить новый адрес и значение для записи в другую ячейку памяти. И спустя tadh = 20 нс установить YE в единицу и записать следующую слово. Если запись больше не требуется, то спустя время tpgh = 20 нс после очистки бита YE необходимо очистить бит PROG и спустя время tnvh = 5 мкс очистить биты XE и NVSTR. Последующие операции с памятью можно выполнять спустя время trcv = 1 мкс. Временная диаграмма записи памяти представлена на рисунке.

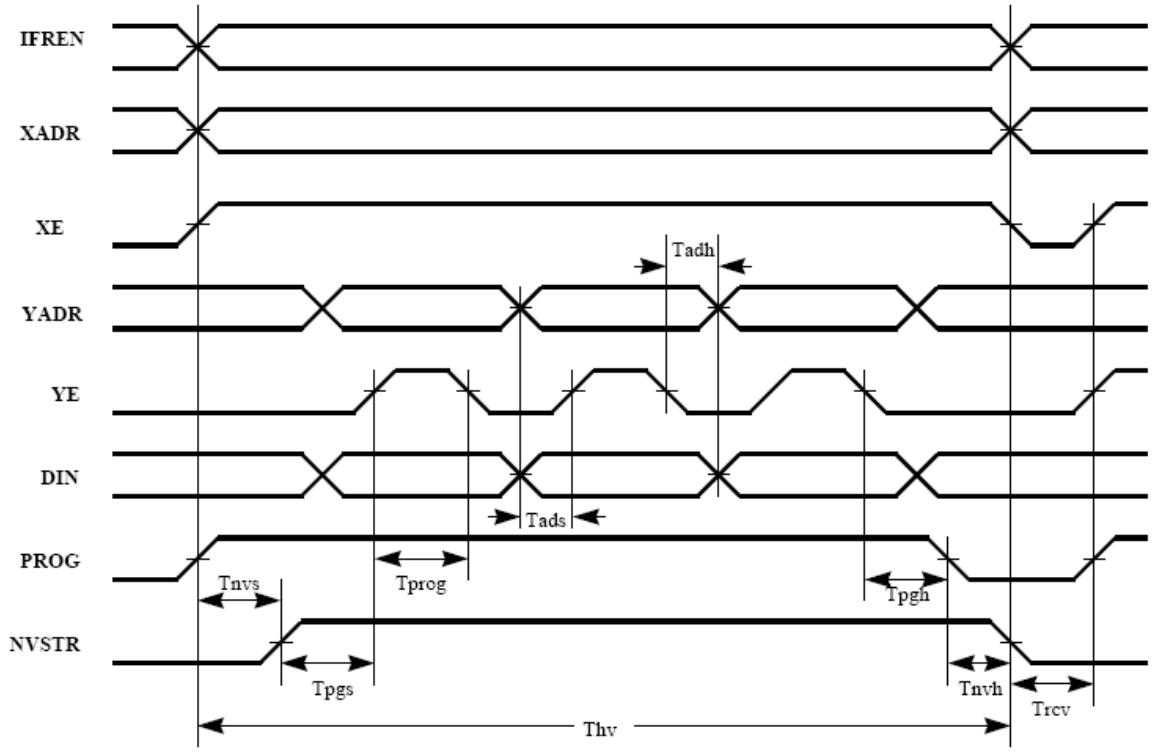


Рисунок 12. Временная диаграмма записи памяти

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 txa = 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 txa = 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 рисунке.

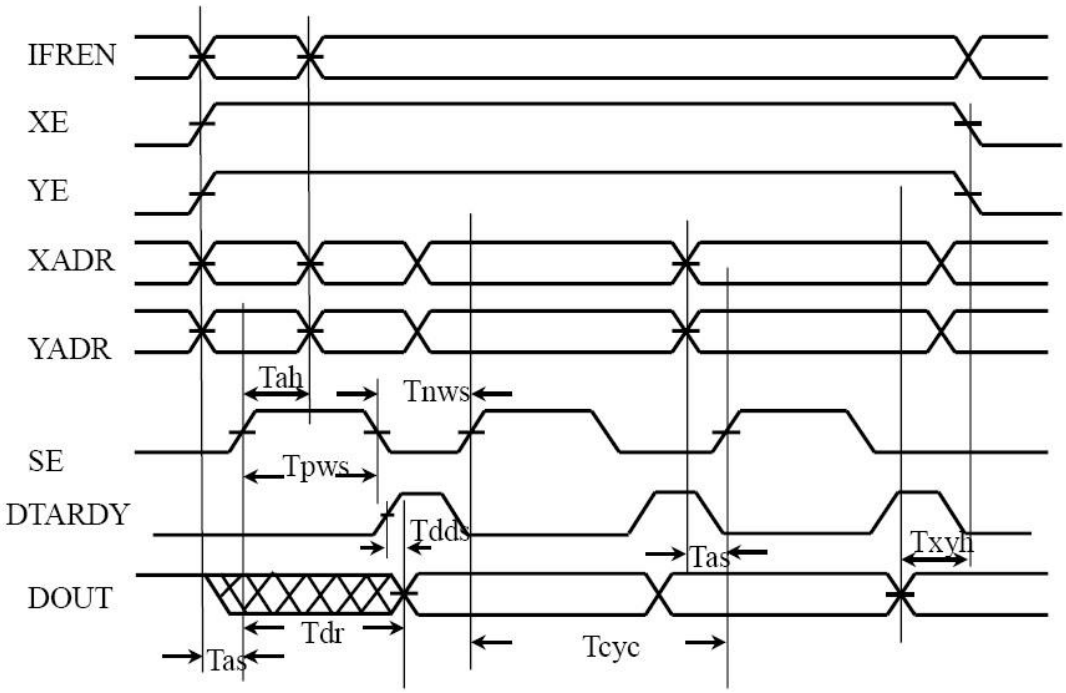


Рисунок 13. Временная диаграмма чтения памяти

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. Регистры управления контроллера Flash памяти программ

Базовый Адрес	**Название**	**Описание**
0x4001_8000	EEPROM_CNTRL	Регистры контроллера Flash памяти программ
Смещение		
0x00	EEPROM_CMD	Регистр управления EEPROM память
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

Таблица 14. Регистр EEPROM_CMD

Номер	31…14	13	12	11	10	9	8
Доступ	U	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0
		NVSTR	**PROG**	**MAS1**	**ERASE**	**IFREN**	**SE**

Номер	7	6	5…3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	100	0	0	0
	YE	**XE**	**Delay[2:0]**	**RD**	**WR**	**CON**

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

Таблица 15. Описание бит регистра EEPROM_CMD

Разряды	**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 EERPOM_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 EERPOM (в режиме программирования): 0 – нет чтения; 1 – есть чтение
1	WR	Запись в память EERPOM (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 EERPOM: 0 – управление EERPOM от ядра, рабочий режим; 1 – управление от регистров, режим программирования

EEPROM_ ADR

Таблица 16. Регистр EEPROM_ADR

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

Таблица 17 Описание бит регистра EEPROM_ADR

Разряды	**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

Таблица 18. Регистр EEPROM_DI

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

Таблица 19. Описание бит регистра EEPROM_DI

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...0	DATA[31:0]	Данные для записи в EERPOM

EEPROM_ DO

Таблица 20. Регистр EEPROM_DO

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

Таблица 21 Описание бит регистра EEPROM_DO

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...0	DATA[31:0]	Данные, считанные из EERPOM

EEPROM_ KEY

Таблица 22. Регистр EEPROM_KEY

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

Таблица 23. Описание бит регистра EEPROM_KEY

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...0	DATA[31:0]	Данные считанные из EERPOM

Система команд

В процессоре реализована версия системы команд Thumb. Поддерживаемые команды представлены в Таблица 24.

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 24. 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 бит 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 регистров, инкремент после доступа	-	
LDMFD, LDMIA	Rn!, reglist	Загрузка множества регистров, инкремент после	-	

Мнемокод команды	Операнды	Краткое описание	Флаги	Страница
		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 байт 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	
SEV	-	Установить признак события	-	
STM	Rn!, reglist	Сохранение множества регистров, инкремент после доступа	-	
STMEA	Rn!, reglist	Сохранение множества регистров, инкремент перед	-	

Мнемокод команды	Операнды	Краткое описание	Флаги	Страница
		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 многопотоковых задачах	-	

Встроенные функции

Стандарт ANSI языка С не обеспечивает непосредственного доступа к некоторым инструкциям процессора Cortex-M0. В разделе описаны встроенные (intrinsic) 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 25. 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.

Таблица 26. 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)

Описание инструкций

В 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.

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. «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 BX, BLX, LDM, LDR или POP 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 параметров 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 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, #0xFFFFFFFE 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 <= n <= 32;

LSL #n - логический сдвиг влево на n бит, 1 <= n <= 31;

LSR #n - логический сдвиг вправо на n бит, 1 <= n <= 32;

ROR #n - циклический сдвиг вправо на n бит, 1 <= n <= 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, 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. Рисунок 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 флаг переноса принимает значение последнего бита, вытесненного в результате операции сдвига, то есть бита [n-1] регистра Rm.

В случае если n >= 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.

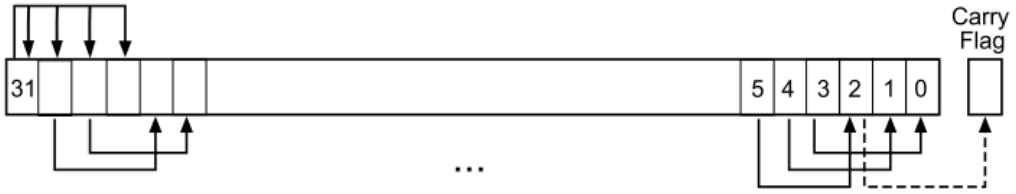


Рисунок 14. ASR #3

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. Рисунок 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 MOVS, 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 >= 32, все биты результата устанавливаются в 0. Если n >= 33 и операция влияет на флаг переноса, то значение этого флага устанавливается равным 0.

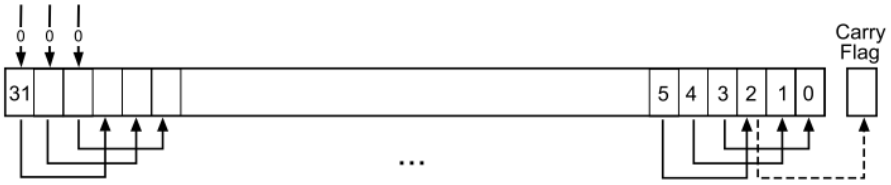


Рисунок 15. LSR #3

LSL

Запуск программы на выполнение.

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. Рисунок 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 MOVS, 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 >= 32, все биты результата устанавливаются в 0. Если n >= 33 и операция влияет на флаг переноса, то значение этого флага устанавливается равным 0.

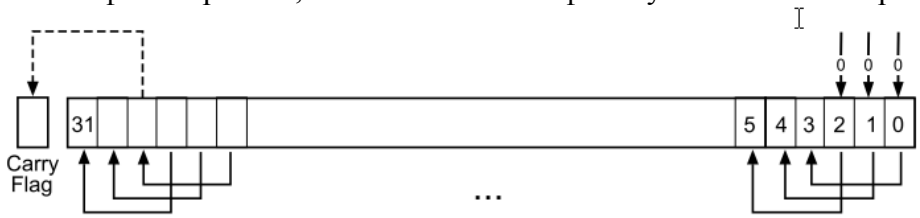


Рисунок 16. LSL #3

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. Рисунок 17.

При использовании инструкции RORS, а также в случае, если сдвиг ROR #n используется при вычислении второго операнда команд MOVS, 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.

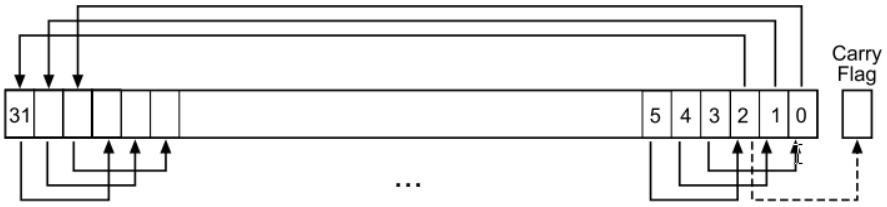


Рисунок 17. ROR #3

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 "Обработка отказов".

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. "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 прикладной программы (APSR) 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 27 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<c> (Branch), где <c> 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 см. 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 27. Там же указаны соответствующие логические выражения для значений флагов.

Условные команды рекомендуется использовать для снижения количества ветвлений в программе.

Таблица 27. 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

Суффикс	Флаги	Значение
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 != V	Меньше, знаковое сравнение
GT	Z = 0 and N = V	Больше, знаковое сравнение
LE	Z = 1 and N != V	Меньше или равно, знаковое сравнение
AL	1	Безусловное исполнение

Команды доступа к памяти

Обобщенные данные о командах доступа к памяти представлены в Таблица 28.

Таблица 28. Команды доступа к памяти

Мнемокод	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. "Адресация относительно счетчика команд".

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, пре-индексированием или пост-индексированием.

Синтаксис

```
op{type} Rt, [Rn {, #offset}]     ; адресация со смещением
op{type} Rt, [Rn, #offset]!       ; пре-индексирование
op{type} Rt, [Rn], #offset        ; пост-индексирование
opD Rt, Rt2, [Rn {, #offset}]     ; адресация со смещением, двойное слово
opD Rt, Rt2, [Rn, #offset]!       ; пре-индексирование, двойное слово
opD Rt, Rt2, [Rn], #offset        ; пост-индексирование, 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 или вычитается из содержимого регистра Rn. Результат используется в качества адреса чтения или записи. Значение регистра Rn остается неизменным.

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

Адресация с пре-индексированием

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

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

Адресация с пост-индексированием

Содержимое регистра Rn используется в качества адреса чтения или записи. Значение смещения добавляется к или вычитается из содержимого регистра Rn, 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. “Выравнивание адресов”.

В Таблица 29 показаны диапазоны значений смещения для различных форм адресации.

Таблица 29. Диапазон значений смещения

Тип инструкции	Смещение	Пре-индексирование	Пост-индексирование
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 пост-индексированием 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 пост-индексированием 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,
                           ; содержащемуся в R10.
STR R2, [R9,#const-struc]  ; const-struc - выражение с постоянным значением,
                           ; лежащим в диапазоне 0-124.
STRH R3, [R4], #4          ; 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 регистр 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. “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]
                            ; Считать байт по адресу, равному сумме R5 и R1,
                            ; умноженному на два, распространить значение
                            ;знакового бита 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, см. "Адресация относительно счетчика команд".

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. “Выравнивание адресов”.

Метка должна располагаться на ограниченном расстоянии от текущей инструкции. В Таблица 30 показаны возможные значения смещений между меткой данных и текущим значением счетчика команд.

Таблица 30 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".

Мнемокод LDMFD и LDMIA - 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 псевдокоманы 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 адреса.

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 таблице ниже представлены инструкции обработки данных.

Таблица 31. 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. «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. “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. «Арифметика с повышенной разрядностью».

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, Rm

причем:

  - не допускается использование суффикса S;
  - в качестве Rm 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. «“Условное исполнение».

Rd - регистр-получатель результата.

Rn - регистр, содержащий значение первого операнда.

Operand2 - второй операнд. См. «“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. «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. «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».

Описание регистров управления контроллера Flash-памяти программ

Таблица 32. Регистры управления контроллера Flash-памяти программ

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

EEPROM_CMD

Таблица 33 Регистр EEPROM_CMD

Номер	31…14	13	12	11	10	9	8
Доступ	U	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0
		NVSTR	**PROG**	**MAS1**	**ERASE**	**IFREN**	**SE**

Номер	7	6	5…3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	100	0	0	0
	YE	**XE**	**Delay[2:0]**	**RD**	**WR**	**CON**

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

Таблица 34 Описание бит регистра EEPROM_CMD

Разряды	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 EERPOM_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 … 111 – 7 циклов
2	RD	Чтение из памяти EERPOM (в режиме программирования): 0 – нет чтения; 1 – есть чтение
1	WR	Запись в память EERPOM (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 EERPOM 0 – управление EERPOM от ядра, рабочий режим; 1 – управление от регистров, режим программирования

EEPROM_ADR

Таблица 35 Регистр EEPROM_ADR

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

Таблица 36 Описание бит регистра EEPROM_ADR

Разряды	**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

Таблица 37 Регистр EEPROM_DI

Номер	31…0
Доступ	R/W
Сброс	0

	DATA [31:0]

Таблица 38 Описание бит регистра EEPROM_DI

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	DATA[31:0]	Данные для записи в EERPOM

EEPROM_DO

Таблица 39 Регистр EEPROM_DO

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

Таблица 40 Описание бит регистра EEPROM_DO

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	DATA[31:0]	Данные, считанные из EERPOM

EEPROM_KEY

Таблица 41 Регистр EEPROM_KEY

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

Таблица 42 Описание бит регистра EEPROM_KEY

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	KEY[31:0]	Ключ для разрешения доступа к Flash памяти через регистровый доступ. Перед переводом памяти в режим программирования необходимо в регистр EERPOM_KEY записать комбинацию 0x8AAA5551.

Процессорное ядро

Описание процессора и периферии ядра.

Процессорное ядро с минимизированным количеством вентилей, обладающее следующими характеристиками:

- процессорное ядро содержит в своём составе 3-х уровневый конвейер;
- набор инструкций архитектуры ARM v6-M, включающий 32 битные Thumb-2 инструкции, такие как BL, MRS, MSR, ISB, DSB и DMB;
- возможность запуска операционной системы и доступные для этого режима работы SVC инструкции, групповой регистр указателя стека и интегрированный системный таймер;
- системная модель исключительных ситуаций;
- режимы Handler и Thread;
- два указателя стека;
- возможность работы только в режиме Thumb;
- отсутствие аппаратной поддержки невыровненного доступа;
- содержит 13 32-разрядных регистров общего назначения, link регистр (LR), счётчик команд (PC), программный регистр статуса xPSR, и два групповых регистра указателя стека (SP).

Контроллер прерываний NVIC. Контроллер интегрирован в процессор для уменьшения задержек в процессе прерываний. Обладает следующими характеристиками:

- поддержка до 32 внешних прерываний;
- два бита приоритета, обеспечивающие четырёхуровневый приоритет прерываний;
- состояние процессора автоматически сохраняется при входе в прерывание и восстанавливается при выходе, что не вызывает потерь на выполнение инструкций.

Интерфейс памяти ITCM, DTCM, а также внешний интерфейс AHB-Lite.

TCM интерфейс не поддерживает тактов ожидания, поэтому при тактовой частоте ядра выше 25 МГц, акселератор Flash памяти выключает тактовую частоту ядра на необходимое количество тактов.

Полный набор отладочных модулей:

- полный доступ в режиме останова ко всей памяти и регистрам;
- отладочный порт DAP;
- модуль точек останова BPU;
- модуль наблюдения данных DW;
- 32-разрядный аппаратный умножитель.

Структурная схема процессора

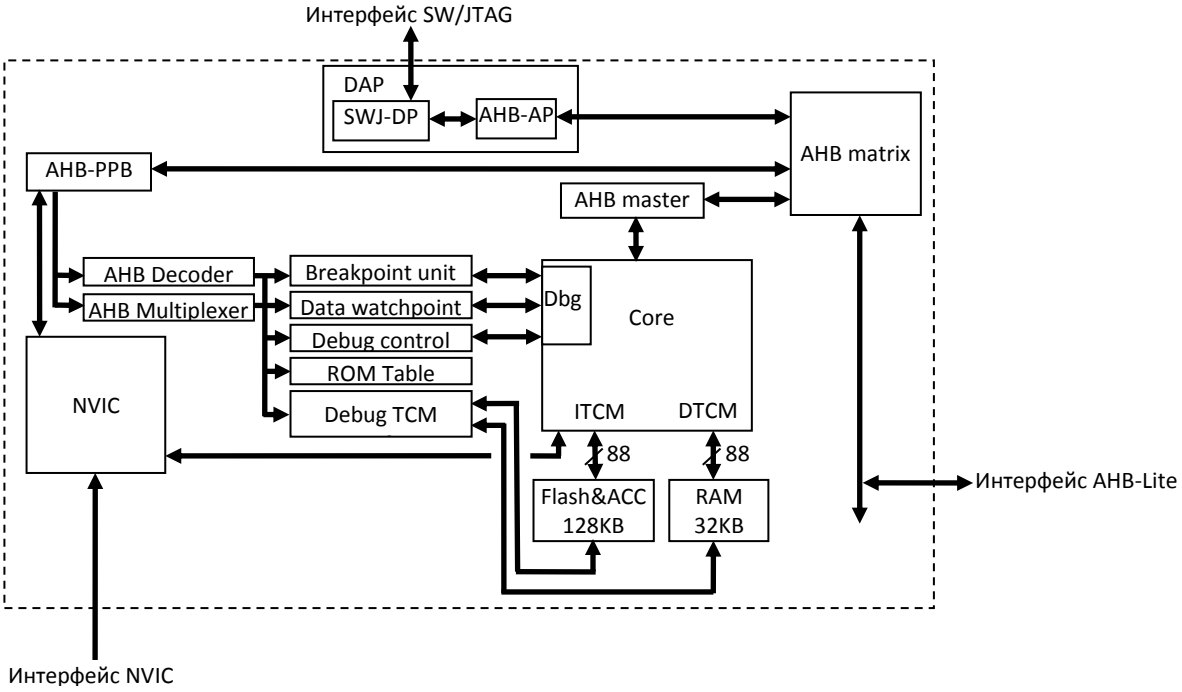


Рисунок 18. Структурная схема процессора

Периферийными блоками ядра являются:

- контроллер прерываний NVIC.

Реализует высокоскоростную обработку прерываний.

- Bus master.

Обеспечивает два интерфейса. Один связывает внутренние Private Peripheral Bus (PPB) сигналы с шиной AHB PPB. Второй интерфейс связывает сигналы внешней шины с AHB портом.

- AHB Private Peripheral Bus (AHB-PPB).

Обеспечивает доступ к контроллеру NVIC и компонентам модулей отладки.

- AHB decoder.

Дешифрирует адреса AHB шины для выработки сигналов выбора для периферии системы отладки.

- AHB multiplexer.

Объединяет все ответы ведомых для отладочных блоков.

- AHB matrix.

Выполняет функцию арбитража между процессором и отладочной системой при доступе к внутренней PPB и внешнему интерфейсу AHB-Lite.

- DAP.

Процессор содержит AHB-Access Port (AHB-AP). AHB-AP преобразует выходы от внешних DP компонентов в интерфейс AHB-Lite. AHB-AP master имеет наивысший приоритет в AHB matrix.

Serial-Wire JTAG Debug Port (SWJ-DP) это комбинация JTAG порта и Serial Wire порта, а также механизма, позволяющего переключаться между Serial Wire и JTAG.

- Debug TCM интерфейс.

Обеспечивает отладочный интерфейс для доступа к ITCM или DTCM. Только один TCM может быть доступен в любой момент времени.

- Breakpoint Unit.

Содержит в своём составе компаратор 4-х адресов инструкций. Можно сконфигурировать каждый компаратор адреса инструкции для выполнения останова программы с использованием аппаратной точки останова. Каждый компаратор может сравнивать адрес выбираемой инструкции с установленным адресом. Если адрес совпал, то BPU обеспечивает останов процессора в момент выполнения инструкции, вызвавшей совпадение. Точки останова поддерживаются только в области кода карты памяти.

- Data Watchpoint unit.

Содержит в своём составе два компаратора адреса. Можно сконфигурировать компараторы для сравнения адреса инструкции или адреса данных. Поддерживается также маскирование компараторов.

Watchpoint частично точно. Это означает, что останов ядра происходит после выполнения следующей инструкции, после той, адрес которой вызвал совпадение компаратора.

- Debug control.

Обеспечивает доступ к управляющим регистрам отладки через PPB для останова и пуска процессора. Помимо этого обеспечивается доступ к регистрам процессора, когда он остановлен.

- ROM table.

Разрешает стандартным отладочным средствам распознать процессор и доступную периферию отладки, а также определить адреса, необходимые для доступа к этой периферии.

Программная модель

Процессор обеспечивает облегчённую версию Thumb-2, это все инструкции определённые в архитектуре ARM v6-M. Процессор не поддерживает выполнение ARM инструкций.

Процессор не поддерживает различий между режимами User и Privileged. Процессор всегда в режиме Privileged.

Процессор может функционировать в режимах:

- Thread режим

Используется для исполнения приложений, процессор находится в этом режиме сразу после сброса

- Handler режим

Используется для обработки исключений. После обработки процессор переходит в Thread режим.

Процессор может функционировать в одном из состояний:

- Thumb state

Это нормальное исполнение Thumb и Thumb-2 инструкций с 16-битными и 32-битными выровненными по полуслову данными.

- Debug state

Это состояние, при котором ядро остановлено.

Стек

По окончанию сброса весь код использует main стек. Обработчик прерываний, такой как SVCall, может переключить стек, который отображался в Thread режиме, из main в process, модификацией значения EXC_RETURN при выходе. Все прерывания продолжают использовать main стек. Указатель стека, R13, совмещённый регистр переключается между main и process стеком. Только один стек, process или main, виден посредством регистра R13 в данный момент времени.

Также возможно переключение между стеками main и process в Thread режиме записью в Special-Purpose Control регистр инструкцией MSR.

Регистры ядра

Процессор содержит следующие 32-х разрядные регистры:

- 13 регистров общего назначения, R0-R12;
- указатель стека (SP, R13) и объединенные регистры, SP_process и SP_main;
- Link регистр (LR, R14);
- счётчик команд (PC, R15);
- программный регистр состояния, xPSR .

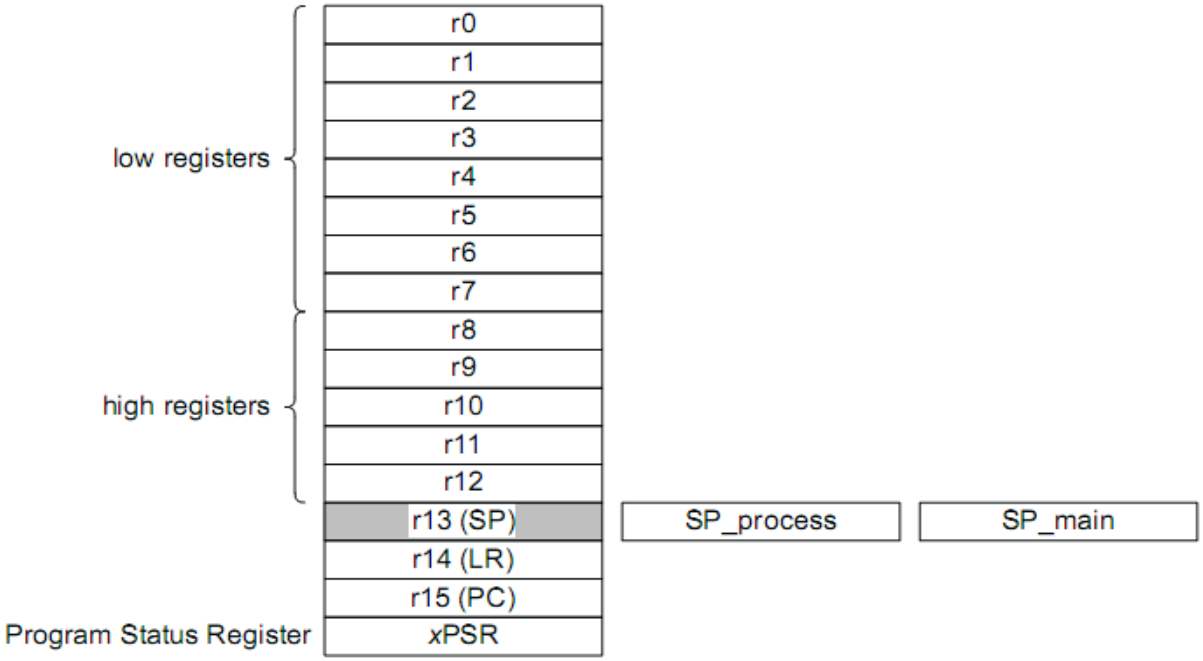


Рисунок 19. Регистры ядра

Регистры общего назначения R0-R12

Low registers или R0-R7 доступны для все инструкций, которые определены для работы с регистрами общего назначения.

High registers или R8-R12 не доступны для 16 разрядных инструкций.

Указать стека SP R13

Регистр R13 используется как указатель стека. Запись в биты [1:0] этого регистра игнорируется, так как он автоматически выровнен по границе слова (четырёх байт). Биты SP[1:0] могут быть очищены инструкцией SBZP. В режиме Handler всегда используется SP_main, а в режиме Thread может быть использован либо SP_main, либо SP_process.

Регистр связи LR R14

Регистр R14 это регистр связи для подпрограмм. LR содержит адрес возврата для PC после выполнения инструкций перехода. Регистр используется для сохранения информации об адресе возврата при уходе на обработку прерываний, вызовах функций и обработке исключений. Во всех остальных случаях регистр может быть использован как регистр общего назначения.

Счетчик команд PC R15

Program Counter это регистр R15. Он содержит адрес текущей инструкции. Бит 0 всегда 0, так как все инструкции выровнены по границе полуслов. При сбросе процессор считывает в этот регистр вектор сброса, который расположен по адресу 0x00000004.

Программный регистр состояния PSR

Регистр Program Status Register (PSR) объединяет:

- Application Program Status Register (APSR);
- Interrupt Program Status Register (IPSR);
- Execution Program Status Register (EPSR).

Эти регистры разделяют различные битовые поля в 32-разрядном PSR. Описание регистров приведено ниже. Доступ к этим регистрам может быть как индивидуальный, так и комбинированный к двум или всем трем разом, используя имена регистров как аргументы инструкций MSR или MRS. Например:

- читать все регистры, используя PSR с MRS инструкцией;
- записать только в APSR используя APSR с MSR инструкцией.

Таблица 43 Комбинация PSR и их атрибуты

Регистр	Тип	Комбинация
XPSR	RW (1),(2)	APSR, EPSR и IPSR
IEPSR	RO	EPSR и IPSR
IAPSR	RW(1)	APSR и IPSR
EAPSR	RW(2)	APSR и EPSR

1- игнорируется запись в IPSR биты.

2 - при чтении EPSR битов читаются нули, и запись в них игнорируется.

Подробнее в описании инструкции «MRS» и «MSR».

APSR

Регистр APSR содержит текущие флаги состояния выполнения предыдущей инструкции.

Таблица 44 Регистр APSR

Номер	31	30	29	28	27…0
Доступ					
Сброс					
	N	**Z**	**C**	**V**	**-**

Таблица 45 Описание бит регистра APSR

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.**
31	N	**Negative** 0 – результат операции положительный, нулевой, больше или равен 1 – результат операции отрицательный или меньше
30	Z	**Zero:** 0 – результат операции не нулевой 1 – результат операции нулевой
29	C	**Carry:** 0 – при суммировании не было переноса, при вычитании не было заема 1 – при суммировании был перенос, при вычитании был заем
28	V	**Overflow:** 0 – в результате операции не было переполнения 1 – в результате операции было переполнение
27…0	-	Зарезервировано

IPSR

Регистр IPSR содержит номер типа исключения для текущего обработчика прерывания.

Таблица 46 Регистр IPSR

Номер	31…6	5…0
Доступ		
Сброс		
	-	**ISR_NUMBER**

Таблица 47 Описание бит регистра IPSR

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…6	-	Зарезервировано
5…0	ISR_NUMBER	**Номер текущего исключения** 0 – Thread режим 2 – NMI 3 – Hard Fault 11 – SVCall 14 – PendSV 15 – SysTick 16 – IRQ0 … 47 – IRQ31

EPSR

Регистр EPSR содержит бит состояния Thumb инструкции.

Таблица 48 Регистр EPSR

Номер	31…25	24	23…0
Доступ			
Сброс			
	-	**T**	-

Таблица 49 Описание бит регистра EPSR

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31..25	-	Зарезервировано
24	T	Этот бит устанавливается в соответствии с вектором сброса, когда процессор выходит из состояния reset. Выполнение инструкции очистки T-бита регистра EPSR приводит к возникновению аппаратной ошибки Hard Fault. Это позволяет быть уверенным, что переключение в ARM состояние, не приведет к непредсказуемым последствиям.
23..0	-	Зарезервировано

Пока процессор не в режиме отладки, попытка читать EPSR, используя MSR инструкцию, всегда возвращает ноль, а попытка записать EPSR, используя MSR напрямую, игнорируется.

Сохранение xPSR бит

При входе в прерывание процессор сохраняет сгруппированные данные из трёх регистров в стек. Бит 9, помещённого в стек, xPSR содержит статус активного SP, когда начался процесс обработки прерывания.

PRIMASK

Priority Mask регистр.

Регистр PRIMASK используется для повышения приоритета.

Таблица 50 Регистр PRIMASK

Номер	31…1	0
Доступ		
Сброс		
	-	**PRIMASK**

Таблица 51 Описание бит регистра PRIMASK

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…1	-	Зарезервировано
0	PRIMASK	0 – не влияет 1 – увеличивает приоритет исполнения до 0

Для доступа к регистру применяются инструкции MSR и MRS, а также инструкция CPS для установки или очистки бита PRIMASK.

CONTROL

Контрольный регистр специального назначения.

Регистр определяет текущий указатель стека.

Таблица 52 Регистр CONTROL

Номер	31…2	1	0
Доступ			
Сброс			
	-	**Active Stack Pointer**	**-**

Таблица 53 Описание бит регистра CONTROL

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…8	-	Зарезервировано
1	Active Stack Pointer	0 – SP_main используется, как текущий указатель стека 1 – Для Thread режима, SP_process используется, как текущий указатель стека [a]
0	-	Зарезервировано

a. Попытка установить этот бит в режиме Handler игнорируется.

Типы данных

Процессор поддерживает следующие типы данных:

- 32-битное слово (words);
- 16-битное полуслово (half words);
- 8-битный байт (bytes).

Процессор может иметь доступ ко всем регионам памяти, включая code регион, со всеми типами данных. Для поддержки этого, система, включая память, должна поддерживать запись полуслов и байт без изменения соседних байт в слове. Процессор манипулирует всеми данными в little-endian режиме. Доступ в память инструкций и Private Peripheral Bus (PPB) всегда в little-endian режиме.

Система команд

В процессоре реализована версия системы команд Thumb. Поддерживаемые команды представлены в Таблица 54.

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 54 Система команд процессора

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 регистров, инкремент после	-	

Мнемокод команды	Операнды	Краткое описание	Флаги	Страница
		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	
SEV	-	Установить признак события	-	
STM	Rn!, reglist	Сохранение множества регистров, инкремент после доступа	-	

Мнемокод команды	Операнды	Краткое описание	Флаги	Страница
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	-	
YIELD	-	Инструкция hint для аппаратного обеспечения при многопотоковых задачах	-	

Встроенные функции

Стандарт ANSI языка С не обеспечивает непосредственного доступа к некоторым инструкциям процессора. В разделе описаны встроенные (intrinsic) 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 55 Встроенные функции CMSIS, позволяющие генерировать некоторые инструкции процессора

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)

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

Таблица 56 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)

Описание инструкций

В 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. “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 BX, BLX, LDM, LDR или POP должен быть равен 1, так как этот бит указывает на требуемый набор команд, а процессор поддерживает только инструкции из набора Thumb.

Формат второго операнда

Большинство команд обработки данных поддерживают 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 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, #0xFFFFFFFE 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 <= n <= 32;.
- LSL #n - логический сдвиг влево на n бит, 1 <= n <= 31;
- LSR #n - логический сдвиг вправо на n бит, 1 <= n <= 32;
- ROR #n - циклический сдвиг вправо на n бит, 1 <= n <= 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, 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. Рисунок 20.

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 флаг переноса принимает значение последнего бита, вытесненного в результате операции сдвига, то есть бита [n-1] регистра Rm.

В случае, если n >= 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.

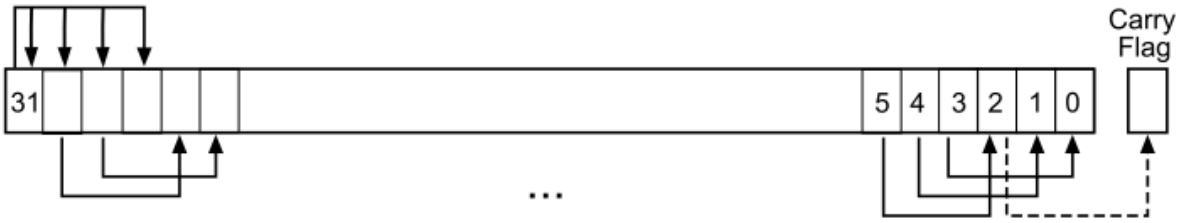


Рисунок 20. Инструкция ASR #3

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. Рисунок 21.

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 MOVS, 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 >= 32, все биты результата устанавливаются в 0. Если n >= 33 и операция влияет на флаг переноса, то значение этого флага устанавливается равным 0.

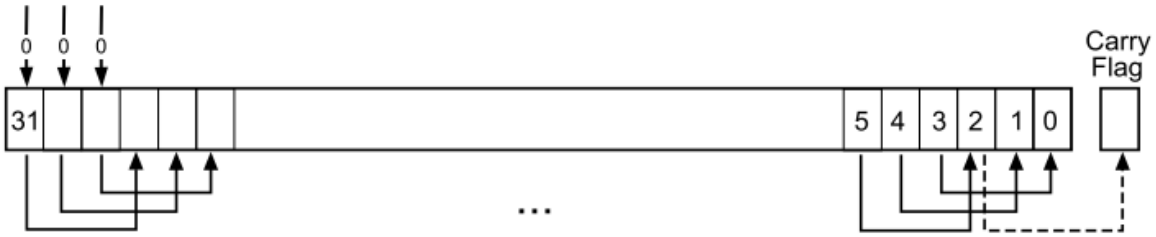


Рисунок 21. Инструкция LSR # 3

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. Рисунок 22.

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 MOVS, 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 >= 32, все биты результата устанавливаются в 0. Если n >= 33 и операция влияет на флаг переноса, то значение этого флага устанавливается равным 0.

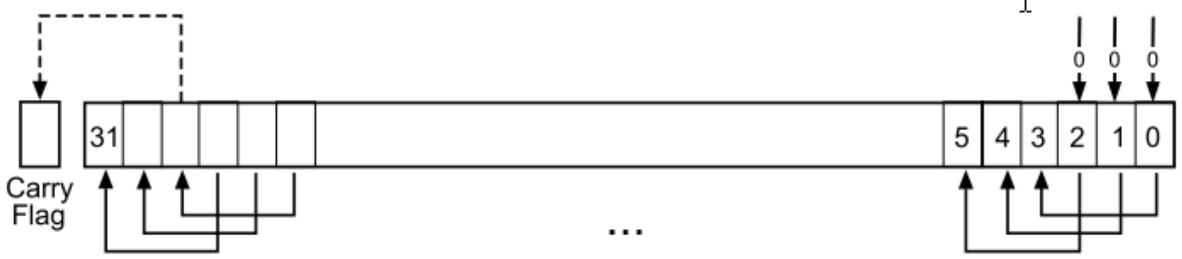


Рисунок 22. Инструкция LSL # 3

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. Рисунок 23.

При использовании инструкции RORS, а также в случае, если сдвиг ROR #n используется при вычислении второго операнда команд MOVS, 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.

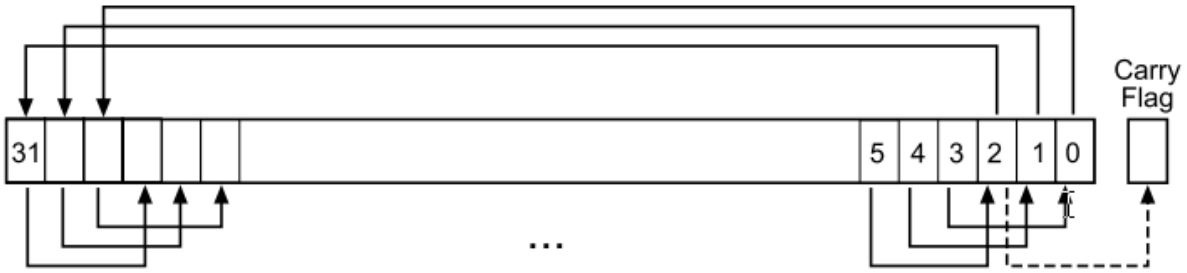


Рисунок 23. Инструкция ROR # 3

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 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 "Обработка отказов".

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. "Регистр конфигурации и управления").

Адресация относительно счетчика команд PC

В системе команд 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 прикладной программы (APSR) 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 57 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<c> (Branch), где <c> 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 см. 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 57. Там же указаны соответствующие логические выражения для значений флагов.

Условные команды рекомендуется использовать для снижения количества ветвлений в программе.

Таблица 57 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 != V	Меньше, знаковое сравнение
GT	Z = 0 and N = V	Больше, знаковое сравнение
LE	Z = 1 and N != V	Меньше или равно, знаковое сравнение
AL	1	Безусловное исполнение

Команды доступа к памяти

Обобщенные данные о командах доступа к памяти представлены в Таблица 58.

Таблица 58 Команды доступа к памяти

Мнемокод	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. "Адресация относительно счетчика команд".

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, пре-индексированием или пост-индексированием.

Синтаксис

```
op{type} Rt, [Rn {, #offset}]      ; адресация со смещением
op{type} Rt, [Rn, #offset]!        ; пре-индексирование
op{type} Rt, [Rn], #offset         ; пост-индексирование
opD Rt, Rt2, [Rn {, #offset}]      ; адресация со смещением, двойное слово
opD Rt, Rt2, [Rn, #offset]!        ; пре-индексирование, двойное слово
opD Rt, Rt2, [Rn], #offset         ; пост-индексирование, 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 или вычитается из содержимого регистра Rn. Результат используется в качества адреса чтения или записи. Значение регистра Rn остается неизменным.

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

Адресация с пре-индексированием

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

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

Адресация с пост-индексированием

Содержимое регистра Rn используется в качества адреса чтения или записи. Значение смещения добавляется к или вычитается из содержимого регистра Rn, 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. “Выравнивание адресов”.

В Таблица 59 показаны диапазоны значений смещения для различных форм адресации.

Таблица 59 Диапазон значений смещения

Тип инструкции	**Смещение**	**Пре-индексирование**	**Пост-индексирование**
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 пост-индексированием 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 пост-индексированием 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,
                           ; содержащемуся в R10.
STR R2, [R9,#const-struc]  ; const-struc - выражение с постоянным значением,
                           ; лежащим в диапазоне 0-124.
STRH R3, [R4], #4          ; 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. “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]
                      ; Считать байт по адресу, равному сумме R5 и R1,
                      ; умноженному на два, распространить значение знакового
                      ; бита 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, см. "Адресация относительно счетчика команд".

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. “Выравнивание адресов”.

Метка должна располагаться на ограниченном расстоянии от текущей инструкции. В Таблица 60 показаны возможные значения смещений между меткой данных и текущим значением счетчика команд.

Таблица 60 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".

Мнемокод LDMFD и LDMIA - 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 адреса.

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 61 представлены инструкции обработки данных.

Таблица 61 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. “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. “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. “Арифметика с повышенной разрядностью ”.

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, Rm

причем:

  - не допускается использование суффикса S;
  - в качестве Rm 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. “Условное исполнение”.

Rd - регистр-получатель результата.
Rn - регистр, содержащий значение первого операнда.
Operand2 - второй операнд. См. “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. “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. “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. “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:

Rm - регистр, содержащий первый операнд.
Operand2 - второй операнд. См. «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. “Условное исполнение”.

Rd - регистр-получатель результата.
Operand2 - второй операнд. См. “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 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. “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 0xFFFFFFF0 (инверсия значения 0x0F)
                  ; в регистр R2, установить флаги
```

REV, REV16, REVSH

Изменение порядка бит или байт в слове.

Синтаксис

```
op Rd, Rn
```

где:

op - один из кодов операции:

- REV - изменить на обратный порядок байт в слове;
- REV16 - изменить на обратный порядок байт в полусловах;
- REVSH - 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.
- REVSH - 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
REVSH R0, R5      ; Обратить полуслово со знаком
REVHS R3, R7      ; Обратить порядок при условии "больше или равно" (HS)
```

TST

Проверить значение бит по маске, проверить равенство.

Синтаксис

TST Rn, Operand2

где:

Rn - регистр, содержащий первый операнд.
Operand2 - второй операнд. См. “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. “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 таблице ниже представлена информация о команде умножения.

Таблица 62 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. “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 таблице ниже показаны инструкции, позволяющие манипулировать последовательностями смежных бит данных в регистрах или битовых полях:

Таблица 63 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 шестнадацать 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 таблице ниже представлен список инструкций передачи управления.

Таблица 64 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. “Условное исполнение”.
label - относительный адрес, см. "Адресация относительно счетчика команд".

Rm - регистр, содержащий адрес, на который необходимо передать управление. 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 таблице ниже представлен диапазон адресуемых переходов для различных команд ветвления.

Таблица 65 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 66 представлен список не рассмотренных в предыдущих разделах инструкций процессора Cortex-M0.

Таблица 66. 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	
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	Инструкция hint. Применяется в мультипотоковых 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.

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    ; Data Synchronisation Barrier
```

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: 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.

Ограничения

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

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

```
MRS R0, PRIMASK ; Считать 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.

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 “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;
- 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 7 каналов 24 битных ∑Δ АЦП. Все каналы разбиты на три пары F0-F2 (канал напряжения и канал тока) для 3х фазной сети и еще одного независимого канала тока. Каждый из 7 каналов оцифровывает входной сигнал с выходной частотой отсчетов до 4кГц. Кроме этого в каждой паре каналов F0-F2 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. Так же каждый АЦП имеет независимый канал DMA, обеспечивая возможность сохранения данных в ОЗУ без участия процессора.

Структурная схема 7 каналов АЦП приведена ниже.

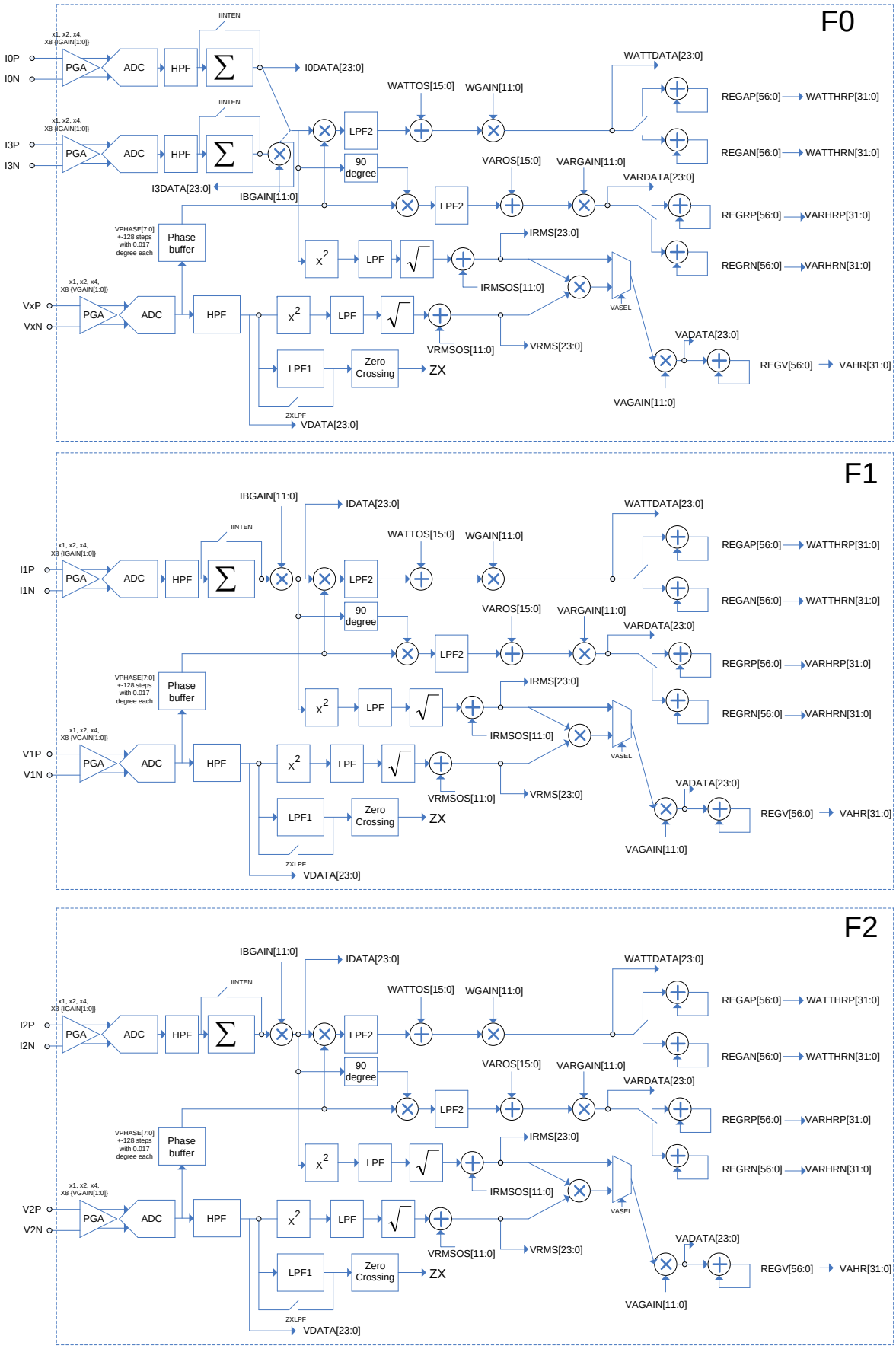


Рисунок 24. Структурная схема 7 каналов АЦП

Список вычисляемых параметров блоком АЦП:

- 7 независимых АЦП с выходной частотой отсчетов 4/8/16 кГц (4 канала тока и 3 канала напряжения). Эти каналы образуют 3 блока для измерения параметров каждой фазы F0-F2.
- В блоке каналов F0 реализуем автоматический выбор канала тока (который имеет максимальное значение) для последующих расчетов мощностных характеристик. Если разница токов превышает 6%, то формируется прерывание. Кроме этой функции в остальном блоки F0-F2 идентичны.
- Все каналы АЦП имеют независимые калибровочные коэффициенты наклона характеристики.
- Каждый канал тока имеет независимый интегратор.
- В каждом блоке АЦП (F0-F2) независимо рассчитывается период сигнала по каналу напряжения. Количество периодов, в течение которого рассчитывается эта величина, можно задавать равным 1/2/4/8/16/32/64/128 периодам.
- В каждом блоке есть проверка на пропажу периодического сигнала в канале напряжения.
- В каждом блоке проверяется просадка напряжения ниже заданного уровня, а так же превышения сигнала в каналах тока и напряжения установленного лимита.
- Есть возможность скорректировать фазы сигналов в каналах 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 течение периода накапливаются в отдельных регистрах (для положительной и отрицательной энергии).
- Вычисляется полная мощность и полная энергия.
- Вычисляется сдвиг фаз по отношению к фазе 0.

Для предотвращения влияния высокочастотных помех на результаты вычисления необходимо поставить внешний anti-aliasing фильтр. Можно использовать простейший RC фильтр первого порядка с частотой среза 10 кГц.

Все цифровые фильтры настроены на указанные частоты среза при входной частоте АЦП равной 1,024/2,048/4,096 МГц (выходной частоте отсчетов 4/8/16 кГц).

Описание регистров управления блока 7 каналов АЦП

Таблица 67 Регистры управления блока 7 каналов АЦП

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	ADCIU_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_F0I0DAT	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	ADCIU_F1CTR	Управление в канале F1
0x068	ADCUI_F1WC	Управление расчета активной мощности в канале F1
0x06C	ADCUI_F1WATTP	Значение положительной активной мощности в канале F1
0x070	ADCUI_F1WATTN	Значение отрицательной активной мощности в канале F1
0x074	ADCUI_F1VC	Управление расчета реактивной мощности в канале F1
0x078	ADCUI_F1VARP	Значение положительной реактивной мощности в канале F1
0x07C	ADCUI_F1VARN	Значение отрицательной реактивной мощности в канале F1
0x080	ADCUI_F1AC	Управление расчета полной мощности в канале F1
0x084	ADCUI_F1VR	Значение полной мощности в канале F1
0x088	ADCUI_F1MD0	Параметры 0 канала F1
0x08C	ADCUI_F1MD1	Параметры 1 канала F1
0x090	ADCUI_F1MD2	Параметры 2 канала F1
0x094	ADCUI_F1VPEAK	Пиковое значение в канале напряжения в канале F1
0x098	ADCUI_F1IPEAK	Пиковое значение в канале тока в канале F1
0x09C	ADCUI_F1VDAT	Отсчеты напряжения в канале F1
0x0A0	ADCUI_F1IDAT	Отсчеты тока в канале F1
0x0A4	ADCUI_F1VRMS	Среднеквадратическое значение напряжение канала F1
0x0A8	ADCUI_F1VRMS2	Квадрат RMS в канале напряжения F1
0x0AC	ADCUI_F1IRMS	Среднеквадратическое значение тока канала F1
0x0B0	ADCUI_F1IRMS2	Квадрат RMS в канале тока F1
0x0B4	ADCUI_F1STAT	Статус канала F1
0x0B8	ADCUI_F1MASK	Маска прерываний канала F1
0x0BC	ADCIU_F2CTR	Управление в канале F2
0x0C0	ADCUI_F2WC	Управление расчета активной мощности в канале F2
0x0C4	ADCUI_F2WATTP	Значение положительной активной мощности в канале F2
0x0C8	ADCUI_F2WATTN	Значение отрицательной активной мощности в канале F2
0x0CC	ADCUI_F2VC	Управление расчета реактивной мощности в канале F2
0x0D0	ADCUI_F2VARP	Значение положительной реактивной мощности в канале F2

0x0D4	ADCUI_F2VARN	Значение отрицательной реактивной мощности в канале F2
0x0D8	ADCUI_F2AC	Управление расчета полной мощности в канале F2
0x0DC	ADCUI_F2VR	Значение полной мощности в канале F2
0x0E0	ADCUI_F2MD0	Параметры 0 канала F2
0x0E4	ADCUI_F2MD1	Параметры 1 канала F2
0x0E8	ADCUI_F2MD2	Параметры 2 канала F2
0x0EC	ADCUI_F2VPEAK	Пиковое значение в канале напряжения в канале F2
0x0F0	ADCUI_F2IPEAK	Пиковое значение в канале тока в канале F2
0x0F4	ADCUI_F2VDAT	Отсчеты напряжения в канале F2
0x0F8	ADCUI_F2IDAT	Отсчеты тока в канале F2
0x0FC	ADCUI_F2VRMS	Среднеквадратическое значение напряжение канала F2
0x100	ADCUI_F2VRMS2	Квадрат RMS в канале напряжения F2
0x104	ADCUI_F2IRMS	Среднеквадратическое значение тока канала F2
0x108	ADCUI_F2IRMS2	Квадрат RMS в канале тока F2
0x10C	ADCUI_F2STAT	Статус канала F2
0x110	ADCUI_F2MASK	Маска прерываний канала F2
0x114	ADCUI_CCAL1	Регистр 1 калибровки канала тока
0x118	ADCUI_CCAL2	Регистр 2 калибровки канала тока
0x11C	ADCUI_CCAL3	Регистр 3 калибровки канала тока
0x120	ADCUI_CCAL4	Регистр 4 калибровки канала тока

ADCUI_CTRL1

Таблица 68 Регистр ADCUI_CTRL1

Номер	31…30	29	28	27	26…25	24:23
Доступ	R/W	R/W	R/W	R/W	-	R/W
Сброс	00	0	0	0	-	00
	OSR_CONF	**IBOOST**	**RESET_DIG**	**ZXRMS**	**-**	**CHOP_FREQ**

Номер	22	21	20	19	18…17	16…15
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	CHOP_EN	**BUF_BYP**	**VREF_SEL**	**FREQSEL**	**VANOLOAD**	**VARNOLOAD**

Номер	14	13…12	11…9	8	7	6
Доступ	-	R/W	R/W	R/W	R/W	R/W
Сброс	-	0	000	0	0	0
	-	**APNOLOAD**	**PER_LENGTH**	**ZXLPF**	**RESOL**	**I3EN**

Номер	5	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	V2EN	**I2EN**	**V1EN**	**I1EN**	**V0EN**	**I0EN**

Таблица 69 Описание бит регистра ADCUI_CTRL1

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...30	OSR_CONF*	Выбор коэффициента передискретизации: 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…25	-	Зарезегвировано
24...23	CHOP_FREQ	Частота chopper: 00 - частота АЦП / 2; 01 - частота АЦП / 4; 10 - частота АЦП / 8; 11 - частота АЦП / 16
22	CHOP_EN	Режим работы АЦП I3:

		0 – нормальный режим; 1 – режим chopper
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	VARNOLOAD	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	V2EN	Разрешение работы канала V2: 0 – канал отключен; 1 – канал включен
4	I2EN	Разрешение работы канала I2: 0 – канал отключен; 1 – канал включен

3	V1EN	Разрешение работы канала V1: 0 – канал отключен; 1 – канал включен
2	I1EN	Разрешение работы канала I1: 0 – канал отключен; 1 – канал включен
1	V0EN	Разрешение работы канала V0: 0 – канал отключен; 1 – канал включен
0	I0EN	Разрешение работы канала I0: 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 THD+noise как минимум на 3 дБ в полосе от 0 Гц до половины частоты дискретизации (это следует из того, что шум интегрируется в частоте 2 раза большей).

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

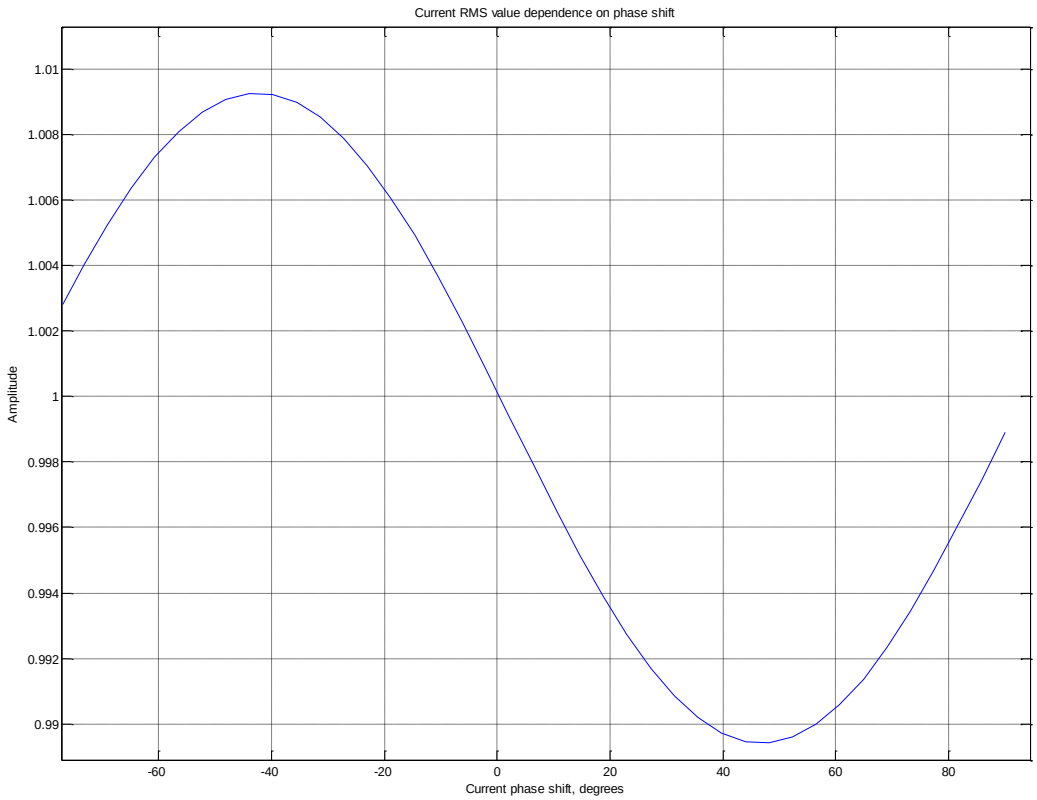


Рисунок 25.

ADCUI_CTRL2

Таблица 70 Регистр ADCUI_CTRL2

Номер	31…24	23…16	15…0
Доступ	-	R/W	R/W
Сброс	-	00h	0000h
	-	**SAGCYC**	**SAGLVL**

Таблица 71 Описание бит регистра ADCUI_CTRL2

Разряды	**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

Таблица 72 Регистр ADCUI_CTRL3

Номер	31…12	11…0
Доступ	-	R/W
Сброс	-	000h
	-	**ZXTOUT**

Таблица 73 Описание бит регистра ADCUI_CTRL3

Разряды	**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

Таблица 74 Регистр ADCUI_ F0CTR

Номер	31…20	19…18	17…10	9…8	7…6
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	0	00	00	0	0
	F0IRMSOS	**F0I3GAIN**	**F0VPHASE**	**F0VGAIN**	**F0IGAIN**

Номер	5	4	3	2	1	0
Доступ	WO	WO	WO	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	F0RVRS	**F0RRRS**	**F0RARS**	**F0VASEL**	**F0I3NTEN**	**F0I0NTEN**

Таблица 75 Описание бит регистра ADCUI_ F0CTR

Разряды	**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	F0I0GAIN	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

Таблица 76 Регистр ADCUI_ F0WC

Номер	31…28	27…16	15…0
Доступ	-	R/W	R/W
Сброс	-	7FFh	0000h
	-	**F0WGAIN**	**F0WATTOS**

Таблица 77 Описание бит регистра ADCUI_ F0WC

Разряды	**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

Таблица 78 Регистр ADCUI_ F0WATTP

Номер	31…0

Доступ	RO
Сброс	
	F0WATTHRP

Таблица 79 Описание бит регистра ADCUI_ F0WATTP

Разряды	**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

Таблица 80 Регистр ADCUI_ F0WATTN

Номер	31…0
Доступ	RO
Сброс	
	F0WATTHRN

Таблица 81 Описание бит регистра ADCUI_ F0WATTN

Разряды	**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

Таблица 82 Регистр ADCUI_ F0VC

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

Таблица 83 Описание бит регистра ADCUI_ F0VC

Разряды	**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

Таблица 84 Регистр ADCUI_ F0VARP

Номер	31..0
Доступ	RO
Сброс	
	F0VARHRP

Таблица 85 Описание бит регистра ADCUI_ F0VARP

Разряды	**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

Таблица 86 Регистр ADCUI_ F0VARN

Номер	31..0
Доступ	RO
Сброс	
	F0VARHRN

Таблица 87 Описание бит регистра ADCUI_ F0VARN

Разряды	**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

Таблица 88 Регистр ADCUI_ F0AC

Номер	31…28	27…16	15…12	11…0
Доступ	-	R/W	-	R/W
Сброс	-	000h	-	000h
	-	**F0VAGAIN**	-	**F0VRMSOS**

Таблица 89 Описание бит регистра ADCUI_ F0AC

Разряды	**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

Таблица 90 Регистр ADCUI_ F0VR

Номер	31…0
Доступ	RO
Сброс	
	F0VAHR

Таблица 91 Описание бит регистра ADCUI_ F0VR

Разряды	**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_F0MD0

Таблица 92 Регистр ADCUI_ F0MD0

Номер	31…20	29	28…12	11…9	8	7
Доступ	R/W	R/W	RO	-	R/W	R/W

Сброс	00	0		-	0	0
	F0SEL_I_CH	**F0I3SEL**	**F0PER_FREQ**	**-**	**I3GAIN**	**V0GAIN**

Номер	6	5	4	3:2	1:0
Доступ	R/W	R	R	R/W	R/W
Сброс	0	0	0	00	00
	I0GAIN	**F0REACTS**	**F0ACTS**	**F0ISEL**	**F0VSEL**

Таблица 93 Описание бит регистра ADCUI_ F0MD0

Разряды	**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 характеристик: 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 – +6 дБ усиление
7	V0GAIN	Усиление в канале V0: 0 – нет усиления; 1 – +6 дБ усиление
6	I0GAIN	Усиление в канале I0: 0 – нет усиления; 1 – +6 дБ 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

Таблица 94 Регистр ADCUI_ F0MD1

Номер	31…16	15…0
Доступ	R/W	R/W
Сброс	0000h	0000h
	F0VPKLVL	**F0IPKLVL**

Таблица 95 Описание бит регистра ADCUI_ F0MD1

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	F0VPKLVL	Предельный разрешенный уровень напряжения
15...0	F0IPKLVL	Предельный разрешенный уровень тока

ADCUI_F0VPEAK

Таблица 96 Регистр ADCUI_ F0VPEAK

Номер	31…24	23…0
Доступ		R/W
Сброс		000000h
	-	**F0VPEAK**

Таблица 97 Описание бит регистра ADCUI_ F0VPEAK

Разряды	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 пиковое значение

ADCUI_F0IPEAK

Таблица 98 Регистр ADCUI_ F0IPEAK

Номер	31:24	23:0
Доступ		R/W
Сброс		000000h
	-	**F0IPEAK**

Таблица 99 Описание бит регистра ADCUI_ F0IPEAK

Разряды	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 пиковое значение

ADCUI_F0VDAT

Таблица 100 Регистр ADCUI_ F0VDAT

Номер	31…24	23…0
Доступ		RO
Сброс		000000h
	-	**F0VDAT**

Таблица 101 Описание бит регистра ADCUI_ F0VDAT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F0VDAT	FIFO отсчетов напряжения (или одной из мощностей)

ADCUI_F0I0DAT

Таблица 102 Регистр ADCUI_ F0I0DAT

Номер	31…24	23…0
Доступ		RO
Сброс		000000h
	-	**F0I0DAT**

Таблица 103 Описание бит регистра ADCUI_ F0I0DAT

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…24	-	Зарезервировано
23...0	F0I0DAT	FIFO отсчетов тока 0 (или одной из мощностей)

ADCUI_F0I3DAT

Таблица 104 Регистр ADCUI_ F0I3DAT

Номер	31:24	23:0
Доступ		RO
Сброс		000000h
	-	**F0I3DAT**

Таблица 105 Описание бит регистра ADCUI_ F0I3DAT

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...24	-	Зарезервировано
23...0	F0I3DAT	FIFO отсчетов тока 3

ADCUI_F0VRMS

Таблица 106 Регистр ADCUI_ F0VRMS

Номер	31…24	23…0
Доступ		RO
Сброс		000000h
	-	**F0VRMS**

Таблица 107 Описание бит регистра ADCUI_ F0VRMS

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...24	-	Зарезервировано
23...0	F0VRMS	Среднеквадратическое значение напряжения

ADCUI_F0VRMS2

Таблица 108 Регистр ADCUI_ F0VRMS2

Номер	31…0
Доступ	RO
Сброс	
	F0VRMS2

Таблица 109 Описание бит регистра ADCUI_ F0VRMS2

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F0VRMS2	Квадрат среднеквадратического значения напряжения

ADCUI_F0IRMS

Таблица 110 Регистр ADCUI_ F0IRMS

Номер	31…24	23…0
Доступ		RO
Сброс		000000h
	-	**F0IRMS**

Таблица 111 Описание бит регистра ADCUI_ F0IRMS

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F0IRMS	Среднеквадратическое значение тока

ADCUI_F0IRMS2

Таблица 112 Регистр ADCUI_ F0IRMS2

Номер	31…0
Доступ	RO
Сброс	
	F0IRMS2

Таблица 113 Описание бит регистра ADCUI_ F0IRMS2

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F0IRMS2	Квадрат среднеквадратического значения тока

ADCUI_F0STAT

Таблица 114 Регистр ADCUI_ F0STAT

Номер	31…27	26	25	24	23	22
Доступ	-	R/W	R/W	R/W	RO	RO
Сброс	-		0	0	0	0
	-	**F0VAROVN**	**F0WATTOVN**	**C3IF_OVR**	**C3IF_FLL**	**C3IF_EMP**

Номер	21	20	19	18	17	16
Доступ	R/W	RO	-	RO	R/W	RO
Сброс	0	0	-	0	0	0
	F0ZEROCRS	**F0VANLDFL**	-	**F0VARNLDFL**	**F0VARSIGN**	**F0APNLDFL**

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	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**

Таблица 115 Описание бит регистра ADCUI_ F0STAT

Разряды	**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 переполнение регистра аккумулятора с полной энергией. Запись “1” сбрасывает этот флаг
10	F0VAROVP	Флаг, что произошло переполнение регистра аккумулятора с положительной реактивной энергией. Запись “1” сбрасывает этот флаг
9	F0WATTOVP	Флаг, что произошло переполнение 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 переполнение 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 переполнение 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

Таблица 116 Регистр ADCUI_ F0MASK

Номер	31…27	26	25	24	23	22
Доступ	-	R/W	R/W	R/W	RO	RO
Сброс	-		0	0	0	0
	-	**F0VAROVNM**	**F0WATTOVNM**	**C3IF_OVRM**	**C3IF_FLLM**	**C3IF_EMPM**

Номер	21	20	19	18	17	16
Доступ	R/W	RO	-	RO	R/W	RO
Сброс	0	0	-	0	0	0
	F0ZEROCRSM	**F0VANLDFLM**	-	**F0VARNLDFLM**	**F0VARSIGNM**	**F0APNLDFLM**

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	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_OVRM**	**F0IF_FLLM**	**F0IF_EMPM**	**F0VF_OVRM**	**F0VF_FLLM**	**F0VF_EMPM**

Таблица 117 Описание бит регистра ADCUI_ F0MASK

Разряды	**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_F1CTR

Таблица 118 Регистр ADCUI_ F1CTR

Номер	31:20	19:18	17:10	9:8	7:6
Доступ	R/W	-	R/W	R/W	R/W
Сброс	0	-	00	0	0
	F1IRMSOS	**-**	**F1VPHASE**	**F1VGAIN**	**F1IGAIN**

Номер	5	4	3	2	1	0
Доступ	WO	WO	WO	R/W	-	R/W
Сброс	0	0	0	0	-	0
	F1RVRS	**F1RRRS**	**F1RARS**	**F1VASEL**	**-**	**F1INTEN**

Таблица 119 Описание бит регистра ADCUI_ F1CTR

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...20	F1IRMSOS	Калибровка вычислителя среднеквадратического значения тока
19...18	-	Зарезервировано
17...10	F1VPHASE	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	F1VGAIN	Предусилитель в канале напряжения: 00 – 0dB; 01 – 6dB; 10 – 12dB; 11 – 18dB
7...6	F1IGAIN	Предусилитель в канале тока: 00 – 0dB; 01 – 6dB; 10 – 12dB; 11 – 18dB
5	F1RVRS	Запись в этот регистр сбрасывает счетчик переданной полной энергии
4	F1RRRS	Запись в этот регистр сбрасывает счетчик переданной реактивной энергии
3	F1RARS	Запись в этот регистр сбрасывает счетчик переданной активной энергии
2	F1VASEL	Выбор источника сигнала для сохранения в регистре полной энергии: 0 – полная энергия; 1 – среднеквадратическое значение тока
1	-	Зарезервировано
0	F1INTEN	Отключение интегратора в канале тока: 0 – интегратор включен; 1 – интегратор отключен

ADCUI_F1WC

Таблица 120 Регистр ADCUI_ F1WC

Номер	31…28	27…16	15…0
Доступ	-	R/W	R/W
Сброс	-	000h	0000h
	-	**F1WGAIN**	**F1WATTOS**

Таблица 121 Описание бит регистра ADCUI_ F1WC

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...28	-	Зарезервировано
27...16	F1WGAIN	Калибровка усиления канала
15...0	F1WATTOS	Калибровка смещения канала

ADCUI_F1WATTP

Таблица 122 Регистр ADCUI_ F1WATTP

Номер	31…0
Доступ	RO
Сброс	
	F1WATTHRP

Таблица 123 Описание бит регистра ADCUI_ F1WATTP

Разряды	**Функциональное**	**Расшифровка функционального имени бита, краткое**

	имя бита	описание назначения и принимаемых значений
31...0	F1WATTHRP	Старшие 32 бита внутреннего 57-битного аккумулятора положительной активной энергии

ADCUI_F1WATTN

Таблица 124 Регистр ADCUI_ F1WATTN

Номер	31…0
Доступ	RO
Сброс	
	F1WATTHRN

Таблица 125 Описание бит регистра ADCUI_ F1WATTN

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F1WATTHRN	Старшие 32 бита внутреннего 57-битного аккумулятора отрицательной активной энергии

ADCUI_F1VC

Таблица 126 Регистр ADCUI_ F1VC

Номер	31…28	27…16	15…0
Доступ	-	R/W	R/W
Сброс	-	000h	0000h
	-	**F1VARGAIN**	**F1VAROS**

Таблица 127 Описание бит регистра ADCUI_ F1VC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27....16	F1VARGAIN	Калибровка усиления канала
15…0	F1VAROS	Калибровка смещения канала

ADCUI_F1VARP

Таблица 128 Регистр ADCUI_ F1VARP

Номер	31…0
Доступ	RO
Сброс	
	F1VARHRP

Таблица 129 Описание бит регистра ADCUI_ F1VARP

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F1VARHRP	Старшие 32 бита внутреннего 57-битного аккумулятора положительной реактивной энергии

ADCUI_F1VARN

Таблица 130 Регистр ADCUI_ F1VARN

Номер	31…0
Доступ	RO

Сброс	
	F1VARHRN

Таблица 131 Описание бит регистра ADCUI_ F1VARN

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...0	F1VARHRN	Старшие 32 бита внутреннего 57-битного аккумулятора отрицательной реактивной энергии

ADCUI_F1AC

Таблица 132 Регистр ADCUI_ F1AC

Номер	31…28	27…16	15…12	11…0
Доступ	-	R/W	-	R/W
Сброс	-	000h	-	000h
	-	**F1VAGAIN**	-	**F1VRMSOS**

Таблица 133 Описание бит регистра ADCUI_ F1AC

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...28	-	Зарезервировано
27...16	F1VAGAIN	Калибровка усиления канала
15...12	-	Зарезервировано
11...0	F1VRMSOS	Калибровка вычислителя среднеквадратического значения напряжения

ADCUI_F1VR

Таблица 134 Регистр ADCUI_ F1VR

Номер	31…0
Доступ	RO
Сброс	
	F1VAHR

Таблица 135 Описание бит регистра ADCUI_ F1VR

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...0	F1VAHR	Старшие 32 бита внутреннего 57-битного аккумулятора полной энергии

ADCUI_F1MD0

Таблица 136 Регистр ADCUI_ F1MD0

Номер	31..29	28..12	11..8	7	6	5	4	3..2	1..0
Доступ	-	RO	-	R/W	R/W	R	R	R/W	R/W
Сброс	-		-	0	0	0	0	00	00
	-	**F1PER_ FREQ**	-	**V1GAIN**	**I1GAIN**	**F1REA CTS**	**F1ACTS**	**F1ISEL**	**F1VSEL**

Таблица 137 Описание бит регистра ADCUI_ F1MD0

Разряды	**Функциональное**	**Расшифровка функционального имени бита, краткое**

	имя бита	**описание назначения и принимаемых значений**
31...29	-	Зарезервировано
28...12	F1PER_FREQ	Длительность такта в канале напряжения
11...8	-	Зарезервировано
7	V1GAIN	Усиление в канале V1: 0 – нет усиления; 1 – +6 дБ усиление
6	I1GAIN	Усиление в канале I1: 0 – нет усиления; 1 – +6 дБ усиление
5	F1REACTS	Знак реактивной энергии в последний период
4	F1ACTS	Знак активной энергии в последний период
3...2	F1ISEL	Выбор источника сигнала для регистра ADCUI_F1IDAT: 00 – отсчеты тока; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты полной мощности
1...0	F1VSEL	Выбор источника сигнала для регистра ADCUI_F1VDAT: 00 – отсчеты напряжения; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты полной мощности

ADCUI_F1MD1

Таблица 138 Регистр ADCUI_ F1MD1

Номер	31…16	15…0
Доступ	R/W	R/W
Сброс	0000h	0000h
	F1VPKLVL	**F1IPKLVL**

Таблица 139 Описание бит регистра ADCUI_ F1MD1

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...16	F1VPKLVL	Предельный разрешенный уровень напряжения
15...0	F1IPKLVL	Предельный разрешенный уровень тока

ADCUI_F1MD2

Таблица 140 Регистр ADCUI_ F1MD2

Номер	31…17	16…0
Доступ	-	RO
Сброс	-	00000h
	-	**F1PHASE**

Таблица 141 Описание бит регистра ADCUI_ F1MD2

№	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...17	-	Зарезервировано
16...0	F1PHASE	Фазовый сдвиг канала напряжения V1 по отношению к V0

ADCUI_F1VPEAK

Таблица 142 Регистр ADCUI_ F1VPEAK

Номер	31…24	23…0
Доступ		R/W
Сброс		000000h
	-	**F1VPEAK**

Таблица 143 Описание бит регистра ADCUI_ F1VPEAK

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...24	-	Зарезервировано
23...0	F1VPEAK	Пиковое значение напряжения. Обнуляется чтением из регистра Запись в этот регистр, сбрасывает пиковое значение

ADCUI_F1IPEAK

Таблица 144 Регистр ADCUI_ F1IPEAK

Номер	31…24	23…0
Доступ		R/W
Сброс		000000h
	-	**F1IPEAK**

Таблица 145 Описание бит регистра ADCUI_ F1IPEAK

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...24	-	Зарезервировано
23...0	F1IPEAK	Пиковое значение тока. Обнуляется чтением из регистра Запись в этот регистр, сбрасывает пиковое значение

ADCUI_F1VDAT

Таблица 146 Регистр ADCUI_ F1VDAT

Номер	31…24	23…0
Доступ		RO
Сброс		000000h
	-	**F1VDAT**

Таблица 147 Описание бит регистра ADCUI_ F1VDAT

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...24	-	Зарезервировано
23...0	F1VDAT	FIFO отсчетов напряжения (или одной из мощностей)

ADCUI_F1IDAT

Таблица 148 Регистр ADCUI_ F1IDAT

Номер	31…24	23…0
Доступ		RO
Сброс		000000h
	-	**F1IDAT**

Таблица 149 Описание бит регистра ADCUI_ F1IDAT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F1IDAT	FIFO отсчетов тока (или одной из мощностей)

ADCUI_F1VRMS

Таблица 150 Регистр ADCUI_ F1VRMS

Номер	31…24	23…0
Доступ		RO
Сброс		000000h
	-	**F1VRMS**

Таблица 151 Описание бит регистра ADCUI_ F1VRMS

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23..0	F1VRMS	Среднеквадратическое значение напряжения

ADCUI_F1VRMS2

Таблица 152 Регистр ADCUI_ F1VRMS2

Номер	31…0
Доступ	RO
Сброс	
	F1VRMS2

Таблица 153 Описание бит регистра ADCUI_ F1VRMS2

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	F1VRMS2	Квадрат среднеквадратического значения напряжения

ADCUI_F1IRMS

Таблица 154 Регистр ADCUI_ F1IRMS

Номер	31…24	23…0
Доступ		R/W
Сброс		000000h
	-	**F1IRMS**

Таблица 155 Описание бит регистра ADCUI_ F1IRMS

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F1IRMS	Среднеквадратическое значение тока

ADCUI_F1IRMS2

Таблица 156 Регистр ADCUI_ F1IRMS2

Номер	31…0

Доступ	RO
Сброс	
	F1IRMS2

Таблица 157 Описание бит регистра ADCUI_ F1IRMS2

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...0	F1IRMS2	Квадрат среднеквадратического значения тока

ADCUI_F1STAT

Таблица 158 Регистр ADCUI_ F1STAT

Номер	31...27	26	25	24…22	21
Доступ	-	R/W	R/W	-	R/W
Сброс	-		0	-	0
	-	**F1VAROVN**	**F1WATTOVN**	-	**F1ZEROCRS**

Номер	20	19	18	17	16	15
Доступ	RO	-	RO	R/W	RO	R/W
Сброс	0	-	0	0	0	0
	F1VANLDFL	-	**F1VARNLDFL**	**F1VARSIGN**	**F1APNLDFL**	**F1APSIGN**

Номер	14…13	12	11	10	9	8	7
Доступ	-	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	-	0	0	0	0	0	0
	-	**F1ZXTOF**	**F1VAOV**	**F1VAROVP**	**F1WATTOVP**	**F1PEAKIF**	**F1PEAKVF**

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
	F1SAGF	**F1IF_OVR**	**F1IF_FLL**	**F1IF_EMP**	**F1VF_OVR**	**F1VF_FLL**	**F1VF_EMP**

Таблица 159 Описание бит регистра ADCUI_ F1STAT

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...27	-	Зарезервировано.
26	F1VAROVN	Флаг, что произошло переполнения регистра аккумулятора с отрицательной реактивной энергией. Запись “1” сбрасывает этот флаг
25	F1WATTOVN	Флаг, что произошло переполнения регистра аккумулятора с отрицательной активной энергией. Запись “1” сбрасывает этот флаг
24...22	-	Зарезервировано
21	F1ZEROCRS	Флаг, что произошло пересечение “0” в канале напряжения. Запись “1” сбрасывает этот флаг
20	F1VANLDFL	Полная мощность ниже уровня сравнения
19	-	Зарезервировано

18	F1VARNLDFL	Реактивная мощность ниже уровня сравнения
17	F1VARSIGN	Смена знака реактивной мощности. Запись “1” сбрасывает этот флаг
16	F1APNLDFL	Активная мощность ниже уровня сравнения.
15	F1APSIGN	Смена знака активной мощности. Запись “1” сбрасывает этот флаг
14...13	-	Зарезервировано
12	F1ZXTOF	Флаг, что в течении TimeOut не было пересечение напряжением значения “0”. Запись “1” сбрасывает этот флаг
11	F1VAOV	Флаг, что произошло переполнения регистра аккумулятора с полной энергией. Запись “1” сбрасывает этот флаг
10	F1VAROVP	Флаг, что произошло переполнения регистра аккумулятора с положительной реактивной энергией. Запись “1” сбрасывает этот флаг
9	F1WATTOVP	Флаг, что произошло переполнения регистра аккумулятора с положительной активной энергией. Запись “1” сбрасывает этот флаг.
8	F1PEAKIF	Флаг, что произошло превышение порогового значения тока. Запись “1” сбрасывает этот флаг
7	F1PEAKVF	Флаг, что произошло превышение порогового значения напряжения. Запись “1” сбрасывает этот флаг
6	F1SAGF	Флаг, что произошла просадка напряжения. Запись “1” сбрасывает этот флаг
5	F1IF_OVR	Флаг, что произошло переполнения FIFO F1IDAT. Запись “1” сбрасывает этот флаг
4	F1IF_FLL	Флаг, что FIFO F1IDAT заполнено
3	F1IF_EMP	Флаг, что FIFO F1IDAT пусто
2	F1VF_OVR	Флаг, что произошло переполнения FIFO F1VDAT. Запись “1” сбрасывает этот флаг
1	F1VF_FLL	Флаг, что FIFO F1VDAT заполнено
0	F1VF_EMP	Флаг, что FIFO F1VDAT пусто

ADCUI_F1MASK

Таблица 160 Регистр ADCUI_ F1MASK

Номер	31...27	26	25	24...22	21
Доступ	-	R/W	R/W	-	R/W
Сброс	-		0	-	0
	-	**F1VAROVNM**	**F1WATTOVNM**	-	**F1ZEROCRSM**

Номер	20	19	18	17	16
Доступ	RO	-	RO	R/W	RO
Сброс	0	-	0	0	0
	F1VANLDFLM	-	**F1VARNLDFLM**	**F1VARSIGNM**	**F1APNLDFLM**

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
Сброс	0	-	0	0	0	0	0	0
	F1SAGSIGNM	**F1IF_OVRM**	**F1ZXTOFM**	**F1VAOVM**	**F1VAROVPM**	**F1WATTOVPM**	**F1PEAKIFM**	**F1PEAKVFM**

Номер	6	5	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0
	F1SAGM	**F1IF_FLLM**	**F1IF_EMPM**	**F1VF_OVRM**	**F1VF_FLLM**	**F1VF_EMPM**	

Таблица 161 Описание бит регистра ADCUI_ F1MASK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...27	-	Зарезервировано
26	F1VAROVNM	Маска бита F1VAROVN
25	F1WATTOVNM	Маска бита F1WATTOVN
24...22	-	Зарезервировано
21	F1ZEROCRSM	Маска бита F1ZEROCRS
20	F1VANLDFLM	Маска бита F1VANLDFL
19	-	Зарезервировано
18	F1VARNLDFLM	Маска бита F1VARNLDFL
17	F1VARSIGNM	Маска бита F1VARSIGN
16	F1APNLDFLM	Маска бита F1APNLDFL
15	F1APSIGNM	Маска бита F1APSIGNM
14...13	-	Зарезервировано
12	F1ZXTOFM	Маска бита F1ZXTOF
11	F1VAOVM	Маска бита F1VAOV
10	F1VAROVPM	Маска бита F1VAROVP
9	F1WATTOVPM	Маска бита F1WATTOVP
8	F1PEAKIFM	Маска бита F1PEAKIF
7	F1PEAKVFM	Маска бита F1PEAKVF
6	F1SAGFM	Маска бита F1SAGF
5	F1IF_OVRM	Маска бита F1IF_OVR
4	F1IF_FLLM	Маска бита F1IF_FLL
3	F1IF_EMPM	Маска бита F1IF_EMP
2	F1VF_OVRM	Маска бита F1VF_OVR
1	F1VF_FLLM	Маска бита F1VF_FLL
0	F1VF_EMPM	Маска бита F1VF_EMP

ADCUI_F2CTR

Таблица 162 Регистр ADCUI_ F2CTR

Номер	31…20	19…18	17…10	9…8	7…6
Доступ	R/W	-	R/W	R/W	R/W
Сброс	0	-	00	0	0
	F2IRMSOS	**-**	**F2VPHASE**	**F2VGAIN**	**F2IGAIN**

Номер	5	4	3	2	1	0
Доступ	WO	WO	WO	R/W	-	R/W
Сброс	0	0	0	0	-	0
	F2RVRS	**F2RRRS**	**F2RARS**	**F2VASEL**	**-**	**F2INTEN**

Таблица 163 Описание бит регистра ADCUI_ F2CTR

Разряды	**Функциональное**	**Расшифровка функционального имени бита, краткое**

	имя бита	описание назначения и принимаемых значений
31...20	F2IRMSOS	Калибровка вычислителя среднеквадратического значения тока
19...18	-	Зарезервировано
17..10	F2VPHASE	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	F2VGAIN	Предусилитель в канале напряжения: 00 – 0dB; 01 – 6dB; 10 – 12dB; 11 – 18dB
7...6	F2IGAIN	Предусилитель в канале тока: 00 – 0dB; 01 – 6dB; 10 – 12dB; 11 – 18dB
5	F2RVRS	Запись в этот регистр сбрасывает счетчик переданной полной энергии
4	F2RRRS	Запись в этот регистр сбрасывает счетчик переданной реактивной энергии
3	F2RARS	Запись в этот регистр сбрасывает счетчик переданной активной энергии
2	F2VASEL	Выбор источника сигнала для сохранения в регистре полной энергии: 0 – полная энергия; 1 – среднеквадратическое значение тока
1	-	Зарезервировано
0	F2INTEN	Отключение интегратора в канале тока: 0 – интегратор включен; 1 – интегратор отключен

ADCUI_F2WC

Таблица 164 Регистр ADCUI_ F2WC

Номер	31…28	27…16	15…0
Доступ	-	R/W	R/W
Сброс	-	000h	0000h
	-	**F2WGAIN**	**F2WATTOS**

Таблица 165 Описание бит регистра ADCUI_ F2WC

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...28	-	Зарезервировано
27...16	F2WGAIN	Калибровка усиления канала
15...0	F2WATTOS	Калибровка смещения канала

ADCUI_F2WATTP

Таблица 166 Регистр ADCUI_ F2WATTP

Номер	31…0

Доступ	RO
Сброс	
	F2WATTHRP

Таблица 167 Описание бит регистра ADCUI_ F2WATTP

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...0	F2WATTHRP	Старшие 32 бита внутреннего 57-битного аккумулятора положительной активной энергии

ADCUI_F2WATTN

Таблица 168 Регистр ADCUI_ F2WATTN

Номер	31…0
Доступ	RO
Сброс	
	F2WATTHRN

Таблица 169 Описание бит регистра ADCUI_ F2WATTN

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...0	F2WATTHRN	Старшие 32 бита внутреннего 57-битного аккумулятора отрицательной активной энергии

ADCUI_F2VC

Таблица 170 Регистр ADCUI_ F2VC

Номер	31…28	27…16	15…0
Доступ	-	R/W	R/W
Сброс	-	000h	0000h
	-	**F2VARGAIN**	**F2VAROS**

Таблица 171 Описание бит регистра ADCUI_ F2VC

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...28	-	Зарезервировано
27...16	F2VARGAIN	Калибровка усиления канала
15...0	F2VAROS	Калибровка смещения канала

ADCUI_F2VARP

Таблица 172 Регистр ADCUI_ F2VARP

Номер	31…0
Доступ	RO
Сброс	
	F2VARHRP

Таблица 173 Описание бит регистра ADCUI_ F2VARP

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...0	F2VARHRP	Старшие 32 бита внутреннего 57-битного аккумулятора

		положительной реактивной энергии

ADCUI_F2VARN

Таблица 174 Регистр ADCUI_ F2VARN

Номер	31…0
Доступ	RO
Сброс	
	F2VARHRN

Таблица 175 Описание бит регистра ADCUI_ F2VARN

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...0	F2VARHRN	Старшие 32 бита внутреннего 57-битного аккумулятора отрицательной реактивной энергии

ADCUI_F2AC

Таблица 176 Регистр ADCUI_ F2AC

Номер	31…28	27…16	15…12	11…0
Доступ	-	R/W	-	R/W
Сброс	-	000h	-	000h
	-	**F2VAGAIN**	-	**F2VRMSOS**

Таблица 177 Описание бит регистра ADCUI_ F2AC

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...28	-	Зарезервировано
27...16	F2VAGAIN	Калибровка усиления канала
15...12	-	Зарезервировано
11...0	F2VRMSOS	Калибровка вычислителя среднеквадратического значения напряжения

ADCUI_F2VR

Таблица 178 Регистр ADCUI_ F2VR

Номер	31…0
Доступ	R
Сброс	
	F2VAHR

Таблица 179 Описание бит регистра ADCUI_ F2VR

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...0	F2VAHR	Старшие 32 бита внутреннего 57-битного аккумулятора полной энергии

ADCUI_F2MD0

Таблица 180 Регистр ADCUI_ F2MD0

Номер	31..29	28..12	11..8	7	6	5	4	3..2	1..0
Доступ	-	RO	-	R/W	R/W	R	R	R/W	R/W

Сброс	-		-	0	0	0	0	00	00
	-	**F2PER_FREQ**	-	**V2GAIN**	**I2GAIN**	**F2REACTS**	**F2ACTS**	**F2ISEL**	**F2VSEL**

Таблица 181 Описание бит регистра ADCUI_ F2MD0

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...29	-	Зарезервировано
28...12	F2PER_FREQ	Длительность такта в канале напряжения
11...8	-	Зарезервировано
7	V2GAIN	Усиление в канале V1: 0 – нет усиления; 1 – +6 дБ усиление
6	I2GAIN	Усиление в канале I1: 0 – нет усиления; 1 – +6 дБ усиление
5	F2REACTS	Знак реактивной энергии в последний период
4	F2ACTS	Знак активной энергии в последний период
3...2	F2ISEL	Выбор источника сигнала для регистра ADCUI_F1IDAT: 00 – отсчеты тока; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты полной мощности
1...0	F2VSEL	Выбор источника сигнала для регистра ADCUI_F1VDAT^ 00 – отсчеты напряжения; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты полной мощности

ADCUI_F2MD1

Таблица 182 Регистр ADCUI_ F2MD1

Номер	31…16	15…0
Доступ	R/W	R/W
Сброс	0000h	0000h
	F2VPKLVL	**F2IPKLVL**

Таблица 183 Описание бит регистра ADCUI_ F2MD1

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...16	F2VPKLVL	Предельный разрешенный уровень напряжения
15...0	F2IPKLVL	Предельный разрешенный уровень тока

ADCUI_F2MD2

Таблица 184 Регистр ADCUI_ F2MD2

Номер	31…17	16…0
Доступ	-	RO
Сброс	-	00000h
	-	**F2PHASE**

Таблица 185 Описание бит регистра ADCUI_ F2MD2

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...17	-	Зарезервировано
16...0	F2PHASE	Фазовый сдвиг канала напряжения V1 по отношению к V0

ADCUI_F2VPEAK

Таблица 186 Регистр ADCUI_ F2VPEAK

Номер	31…24	23…0
Доступ		R/W
Сброс		000000h
	-	**F2VPEAK**

Таблица 187 Описание бит регистра ADCUI_ F2VPEAK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F2VPEAK	Пиковое значение напряжения. Обнуляется чтением из регистра.

ADCUI_F2IPEAK

Таблица 188 Регистр ADCUI_ F2IPEAK

Номер	31…24	23…0
Доступ		R/W
Сброс		000000h
	-	**F2IPEAK**

Таблица 189 Описание бит регистра ADCUI_ F2IPEAK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F2IPEAK	Пиковое значение тока. Обнуляется чтением из регистра.

ADCUI_F2VDAT

Таблица 190 Регистр ADCUI_ F2VDAT

Номер	31…24	23…0
Доступ		R/W
Сброс		000000h
	-	**F2VDAT**

Таблица 191 Описание бит регистра ADCUI_ F2VDAT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F2VDAT	FIFO отсчетов напряжения (или одной из мощностей)

ADCUI_F2IDAT

Таблица 192 Регистр ADCUI_ F2IDAT

Номер	31…24	23…0
Доступ		R/W
Сброс		000000h
	-	**F2IDAT**

Таблица 193 Описание бит регистра ADCUI_ F2IDAT

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...24	-	Зарезервировано
23...0	F2IDAT	FIFO отсчетов тока (или одной из мощностей)

ADCUI_F2VRMS

Таблица 194 Регистр ADCUI_ F2VRMS

Номер	31…24	23…0
Доступ		R/W
Сброс		000000h
	-	**F2VRMS**

Таблица 195 Описание бит регистра ADCUI_ F2VRMS

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…24	-	Зарезервировано
23...0	F2VRMS	Среднеквадратическое значение напряжения

ADCUI_F2VRMS2

Таблица 196 Регистр ADCUI_ F2VRMS2

Номер	31…0
Доступ	RO
Сброс	
	F2VRMS2

Таблица 197 Описание бит регистра ADCUI_ F2VRMS2

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...0	F2VRMS2	Квадрат среднеквадратического значения напряжения

ADCUI_F2IRMS

Таблица 198 Регистр ADCUI_ F2IRMS

Номер	31…24	23…0
Доступ		R/W
Сброс		000000h
	-	**F2IRMS**

Таблица 199 Описание бит регистра ADCUI_ F2IRMS

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...24	-	Зарезервировано
23...0	F2IRMS	Среднеквадратическое значение тока

ADCUI_F2IRMS2

Таблица 200 Регистр ADCUI_ F2IRMS2

Номер	31…0
Доступ	RO
Сброс	
	F2IRMS2

Таблица 201 Описание бит регистра ADCUI_ F2IRMS2

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...0	F2IRMS2	Квадрат среднеквадратического значения тока

ADCUI_F2STAT

Таблица 202 Регистр ADCUI_ F2STAT

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
	F2SAGF	**F2IF_OVR**	**F2IF_FLL**	**F2IF_EMP**	**F2VF_OVR**	**F2VF_FLL**	**F2VF_EMP**

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
Сброс	0	-	0	0	0	0	0	0
	F2APSIGN	**-**	**F2ZXTOF**	**F2VAOV**	**F2VAROVP**	**F2WATTOVP**	**F2PEAKIF**	**F2PEAKVF**

Номер	31..27	26	25	24..22	21	20	19	18	17	16
Доступ	-	R/W	R/W	-	R/W	RO	-	RO	R/W	RO
Сброс	-		0	-	0	0	-	0	0	0
	-	**F2VAROVN**	**F2WATTOVN**	**-**	**F2ZEROCRS**	**F2VANLDFL**	**-**	**F2VARNLDFL**	**F2VARSIGN**	**F2APNLDFL**

Таблица 203 Описание бит регистра ADCUI_ F2STAT

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…27	-	Зарезервировано
26	F2VAROVN	Флаг, что произошло переполнение регистра аккумулятора с отрицательной реактивной энергией. Запись “1” сбрасывает этот флаг
25	F2WATTOVN	Флаг, что произошло переполнение регистра аккумулятора с отрицательной активной энергией. Запись “1” сбрасывает этот флаг
24…22	-	Зарезервировано
21	F2ZEROCRS	Флаг, что произошло пересечение “0” в канале напряжения. Запись “1” сбрасывает этот флаг
20	F2VANLDFL	Полная мощность ниже уровня сравнения

19	-	Зарезервировано
18	F2VARNLDFL	Реактивная мощность ниже уровня сравнения.
17	F2VARSIGN	Смена знака реактивной мощности. Запись “1” сбрасывает этот флаг
16	F2APNLDFL	Активная мощность ниже уровня сравнения.
15	F2APSIGN	Смена знака активной мощности. Запись “1” сбрасывает этот флаг
14…13	-	Зарезервировано.
12	F2ZXTOF	Флаг, что в течении TimeOut не было пересечения напряжением значения “0”. Запись “1” сбрасывает этот флаг
11	F2VAOV	Флаг, что произошло переполнение регистра аккумулятора с полной энергией. Запись “1” сбрасывает этот флаг
10	F2VAROVP	Флаг, что произошло переполнение регистра аккумулятора с положительной реактивной энергией. Запись “1” сбрасывает этот флаг
9	F2WATTOVP	Флаг, что произошло переполнение регистра аккумулятора с положительной активной энергией. Запись “1” сбрасывает этот флаг
8	F2PEAKIF	Флаг, что произошло превышение порогового значения тока. Запись “1” сбрасывает этот флаг.
7	F2PEAKVF	Флаг, что произошло превышение порогового значения напряжения. Запись “1” сбрасывает этот флаг
6	F2SAGF	Флаг, что произошла просадка напряжения. Запись “1” сбрасывает этот флаг
5	F2IF_OVR	Флаг, что произошло переполнение FIFO F2IDAT. Запись “1” сбрасывает этот флаг
4	F2IF_FLL	Флаг, что FIFO F2IDAT заполнено
3	F2IF_EMP	Флаг, что FIFO F2IDAT пусто
2	F2VF_OVR	Флаг, что произошло переполнение FIFO F2VDAT. Запись “1” сбрасывает этот флаг
1	F2VF_FLL	Флаг, что FIFO F2VDAT заполнено
0	F2VF_EMP	Флаг, что FIFO F2VDAT пусто

ADCUI_F2MASK

Таблица 204 Регистр ADCUI_ F2MASK

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
	F2SAGFM	**F2IF_OVRM**	**F2IF_FLLM**	**F2IF_EMPM**	**F2VF_OVRM**	**F2VF_FLLM**	**F2VF_EMPM**

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
Сброс	0	-	0	0	0	0	0	0
	F2APSIGNM	-	**F2ZXTOFM**	**F2VAOVM**	**F2VAROVPM**	**F2WATTOVP**	**F2PEAKIFM**	**F2PEAKVFM**

						M		

Номер	31..27	26	25	24..22	21	20	19	18	17	16
Доступ	-	R/W	R/W	-	R/W	RO	-	RO	R/W	RO
Сброс	-		0	-	0	0	-	0	0	0
	-	**F2VAROVNM**	**F2WATTOVNM**	-	**F2ZEROCRSM**	**F2VANLDFLM**	-	**F2VARNLDFLM**	**F2VARSIGNM**	**F2APNLDFLM**

Таблица 205 Описание бит регистра ADCUI_ F2MASK

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...27	-	Зарезервировано
26	F2VAROVNM	Маска бита F2VAROVN
25	F2WATTOVNM	Маска бита F2WATTOVN
24...22	-	Зарезервировано
21	F2ZEROCRSM	Маска бита F2ZEROCRS
20	F2VANLDFLM	Маска бита F2VANLDFL
19	-	Зарезервировано
18	F2VARNLDFLM	Маска бита F2VARNLDFL
17	F2VARSIGNM	Маска бита F2VARSIGN
16	F2APNLDFLM	Маска бита F2APNLDFL
15	F2APSIGNM	Маска бита F2APSIGNM
14...13	-	Зарезервировано
12	F2ZXTOFM	Маска бита F2ZXTOF
11	F2VAOVM	Маска бита F2VAOV
10	F2VAROVPM	Маска бита F2VAROVP
9	F2WATTOVPM	Маска бита F2WATTOVP
8	F2PEAKIFM	Маска бита F2PEAKIF
7	F2PEAKVFM	Маска бита F2PEAKVF
6	F2SAGFM	Маска бита F2SAGF
5	F2IF_OVRM	Маска бита F2IF_OVR
4	F2IF_FLLM	Маска бита F2IF_FLL
3	F2IF_EMPM	Маска бита F2IF_EMP
2	F2VF_OVRM	Маска бита F2VF_OVR
1	F2VF_FLLM	Маска бита F2VF_FLL
0	F2VF_EMPM	Маска бита F2VF_EMP

ADCUI_CCAL1

Таблица 206 Регистр ADCUI_ CCAL1

Номер	31…24	23…12	11…0
Доступ			
Сброс			
	-	**I0BGAIN**	**V0BGAIN**

Таблица 207 Описание бит регистра ADCUI_ CCAL1

Разряды	**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_CCAL2

Таблица 208 Регистр ADCUI_ CCAL2

Номер	31…24	23…12	11…0
Доступ			
Сброс			
	-	**I1BGAIN**	**V1BGAIN**

Таблица 209 Описание бит регистра ADCUI_ CCAL2

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...24	-	Зарезервировано
23...12	I1BGAIN	Калибровочный коэффициент канала I1
11...0	V1BGAIN	Калибровочный коэффициент канала V1

ADCUI_CCAL3

Таблица 210 Регистр ADCUI_ CCAL3

Номер	31…24	23…12	11…0
Доступ			
Сброс			
	-	**I2BGAIN**	**V2BGAIN**

Таблица 211 Описание бит регистра ADCUI_ CCAL3

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...24	-	Зарезервировано
23...12	I2BGAIN	Калибровочный коэффициент канала I2
11...0	V2BGAIN	Калибровочный коэффициент канала V2

ADCUI_CCAL4

Таблица 212 Регистр ADCUI_ CCAL4

Номер	31…12	11…0
Доступ		
Сброс		
	-	**I3BGAIN**

Таблица 213 Описание бит регистра ADCUI_ CCAL4

Разряды	**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 оконечных результатов и их соответствия внешним сигналам

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

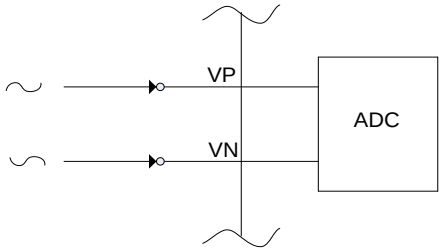


Рисунок 26. Дифференциальное включение

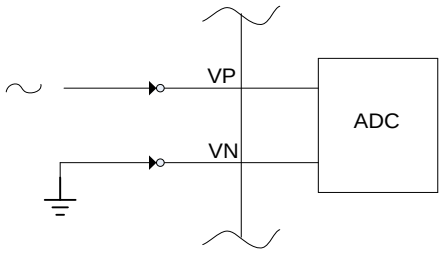


Рисунок 27. Недифференциальное включение

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

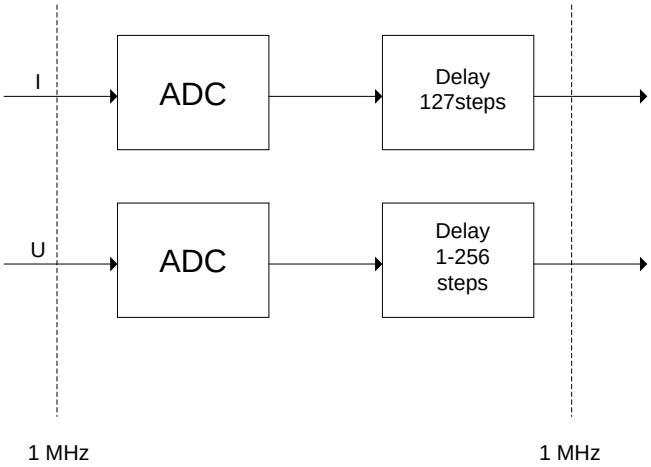


Рисунок 28. Контролируемый фазовый сдвиг в канале напряжения

Изменяя линию задержки в канале напряжения можно регулировать временной сдвиг одного канала относительно другого. Так как частота отсчетов после АЦП равна 1.024 МГц, то один шаг равен 1/20480 периода сигнала с частотой 50 Гц 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:

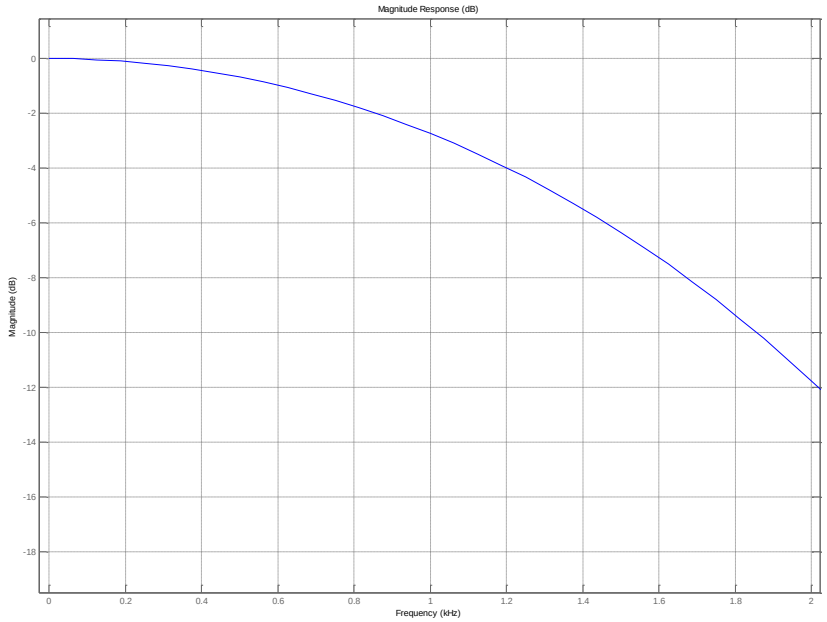


Рисунок 29. Характеристика децемирующего фильтра в полосе 2 кГц

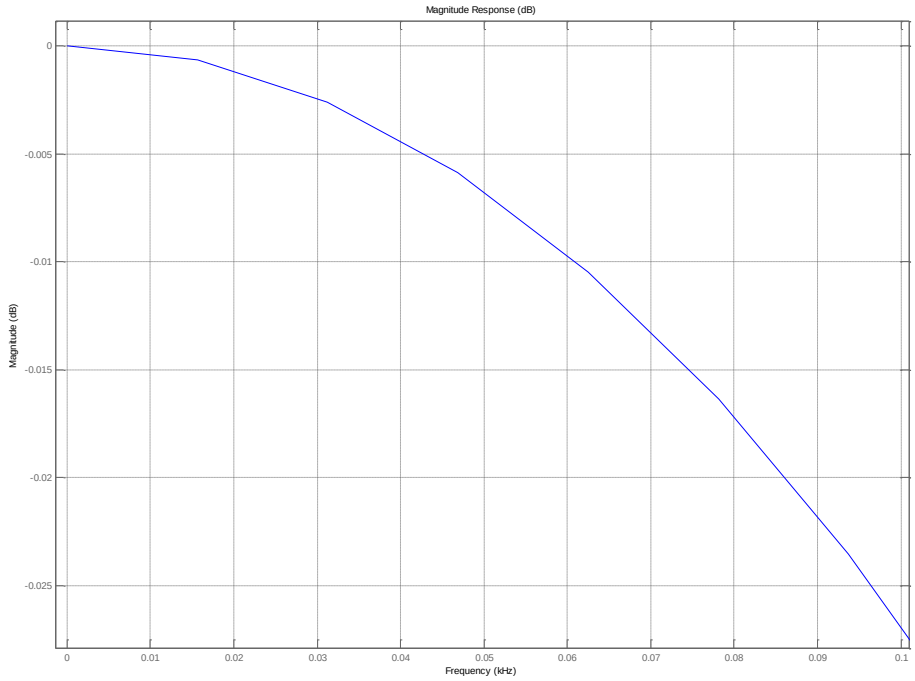


Рисунок 30. Характеристика децемирующего фильтра в полосе 100 Гц

Как видно из вышеприведенных характеристик фильтр подавляет частоты близкие к 2 кГц до величин 12 дБ, что стоит учитывать при измерении гармоник основного тона. В полосе до 100 Гц подавление незначительно (на частоте 50 Гц подавление 0,008 дБ).

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 1 Гц.

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. Значения приведены для усиления PGA равному 0 дБ. Отсчеты, записанные в FIFO, представлены в двоичном формате с дополнением до 2.

Таблица 214

Режим	***Входное напряжение АЦП, Вольт***
Дифференциальное включение, 24 бит. режим	$\frac{FxVDAT}{2^{23}}, \frac{FxIDAT}{2^{23}}$
Дифференциальное включение, 16 бит. режим	$\frac{FxVDAT}{2^{15}}, \frac{FxIDAT}{2^{15}}$
Недифференциальное включение, 24 бит. режим	$\frac{FxVDAT}{2^{22}}, \frac{FxIDAT}{2^{22}}$
Недифференциальное включение, 16 бит. режим	$\frac{FxVDAT}{2^{14}}, \frac{FxIDAT}{2^{14}}$

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.

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

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

Таблица 215

Режим	***Напряжение, Вольт***
Дифференциальное включение	$\frac{FxVRMS}{2^{23}}, \frac{FxIRMS}{2^{23}}$
Недифференциальное включение	$\frac{FxVRMS}{2^{22}}, \frac{FxIRMS}{2^{22}}$

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

Входной сигнал представлен в виде:

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

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

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

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 2 Гц. Этот фильтр подавляет пульсации на частоте 100 Гц (50 Гц *2) с коэффициентом 35 дБ. 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, чтобы избавиться от 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:

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

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 216

Режим	***Напряжение, Вольт^2***
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.

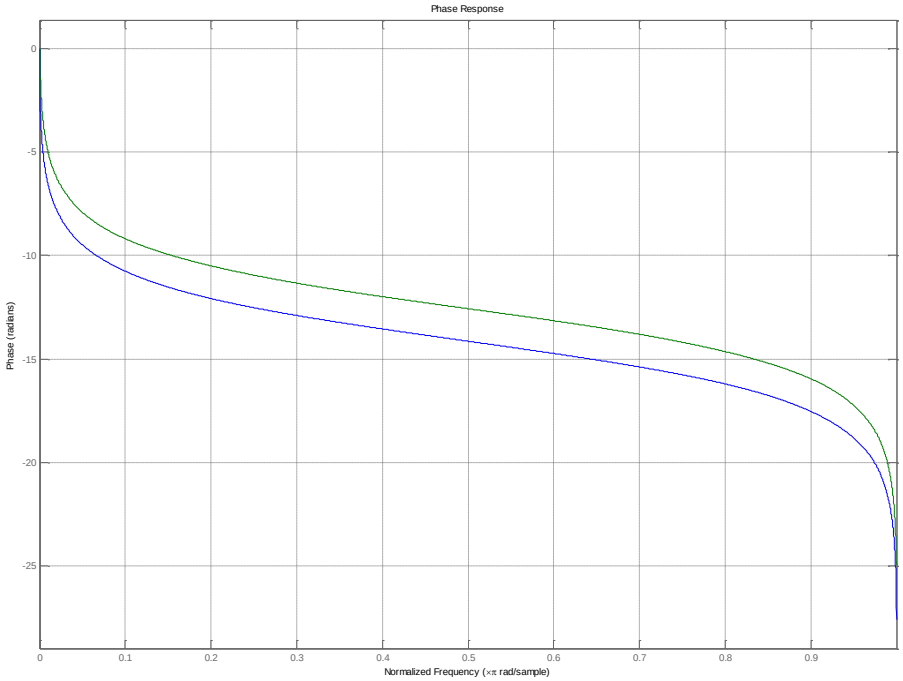


Рисунок 31. Фильтр для сдвига сигнала на 90 градусов

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

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

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

Тогда мгновенное значение мощности равно произведению тока на напряжение:

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

Среднее значение мощности за целое количество тактов равно:

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

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 *FxVDAT* и *FxIDAT* .

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. Значение приведены для усиления PGA, равного 0 дБ. Отсчеты, записанные в FIFO, представлены в двоичном формате с дополнением до 2.

Таблица 217

Режим	***Мощность, Вольт*Ампер***
Дифференциальное включение, 24 бит. режим	$\frac{FxVDAT}{2^{23}}$, $\frac{FxIDAT}{2^{23}}$

Дифференциальное включение, 16 бит. режим	$\frac{FxVDAT}{2^{15}}$, $\frac{FxIDAT}{2^{15}}$
Недифференциальное включение, 24 бит. режим	$\frac{FxVDAT}{2^{21}}$, $\frac{FxIDAT}{2^{21}}$
Недифференциальное включение, 16 бит. режим	$\frac{FxVDAT}{2^{13}}$, $\frac{FxIDAT}{2^{13}}$

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 аккумуляторе. Для каждой из 3 мощностей есть свой аккумулятор. Значение в них определяет потребленную энергию. В таблице ниже приведена формула перевода значения в Ватт*с.

Таблица 218 Формула перевода значения в Ватт*с

Режим	***Энергия, Ватт*с***
Дифференциальное включение	$\frac{FxWATTHR}{512*4000}$, $\frac{FxWATTHR}{512*4000}$
Недифференциальное включение	$\frac{FxWATTHR}{512*1000}$, $\frac{FxWATTHR}{512*1000}$

Типовая схема включения для учета электроэнергии по трем фазам

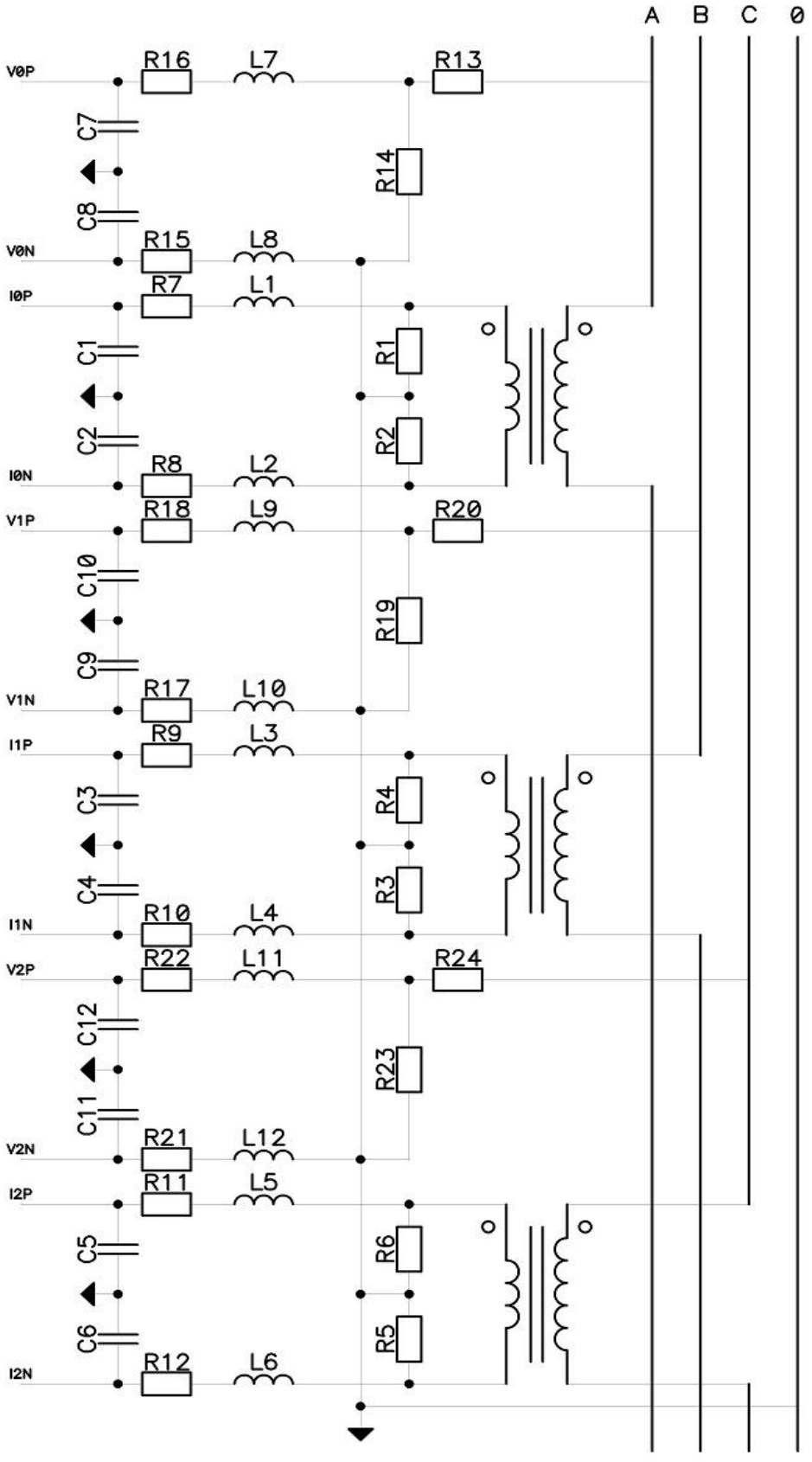


Рисунок 32. Типовая схема включения для учета электроэнергии по трем фазам

Данная схема включения может быть использована как пример для конструирования трехфазных счетчиков электроэнергии. По каждой из трех фаз установлен трансформатор тока. 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 включения средней точкой является аналоговая земля. С этой средней точкой соединена нейтраль трехфазной сети, если она используется. 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 ±500мВ 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

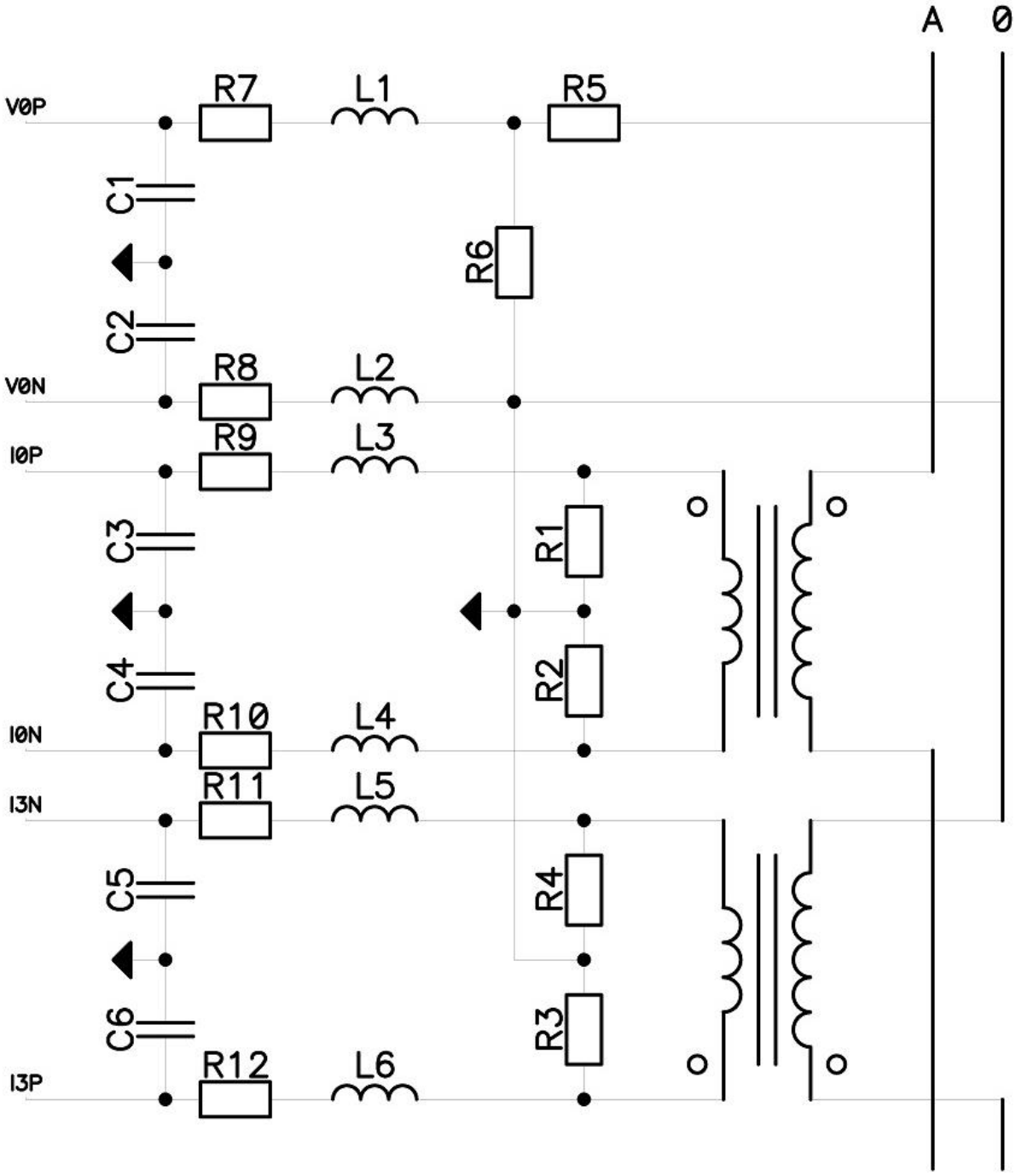


Рисунок 33. Типовая схема включения для учета электроэнергии по одной фазе

В случае, когда необходимо построить однофазный счетчик электроэнергии на базе данной микросхемы, можно взять за основу схему приведенную выше. В данной схеме предусмотрено измерение тока в обоих проводах однофазной сети. Микросхема позволяет в автоматическом режиме учитывать то значение тока из каналов I0 и I3, которое будет больше. Если учета тока по «нулю» не требуется, то часть схемы, относящуюся к каналу I3 можно убрать. В остальном, назначение элементов данной схемы аналогично схеме для учета электроэнергии по трем фазам.

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

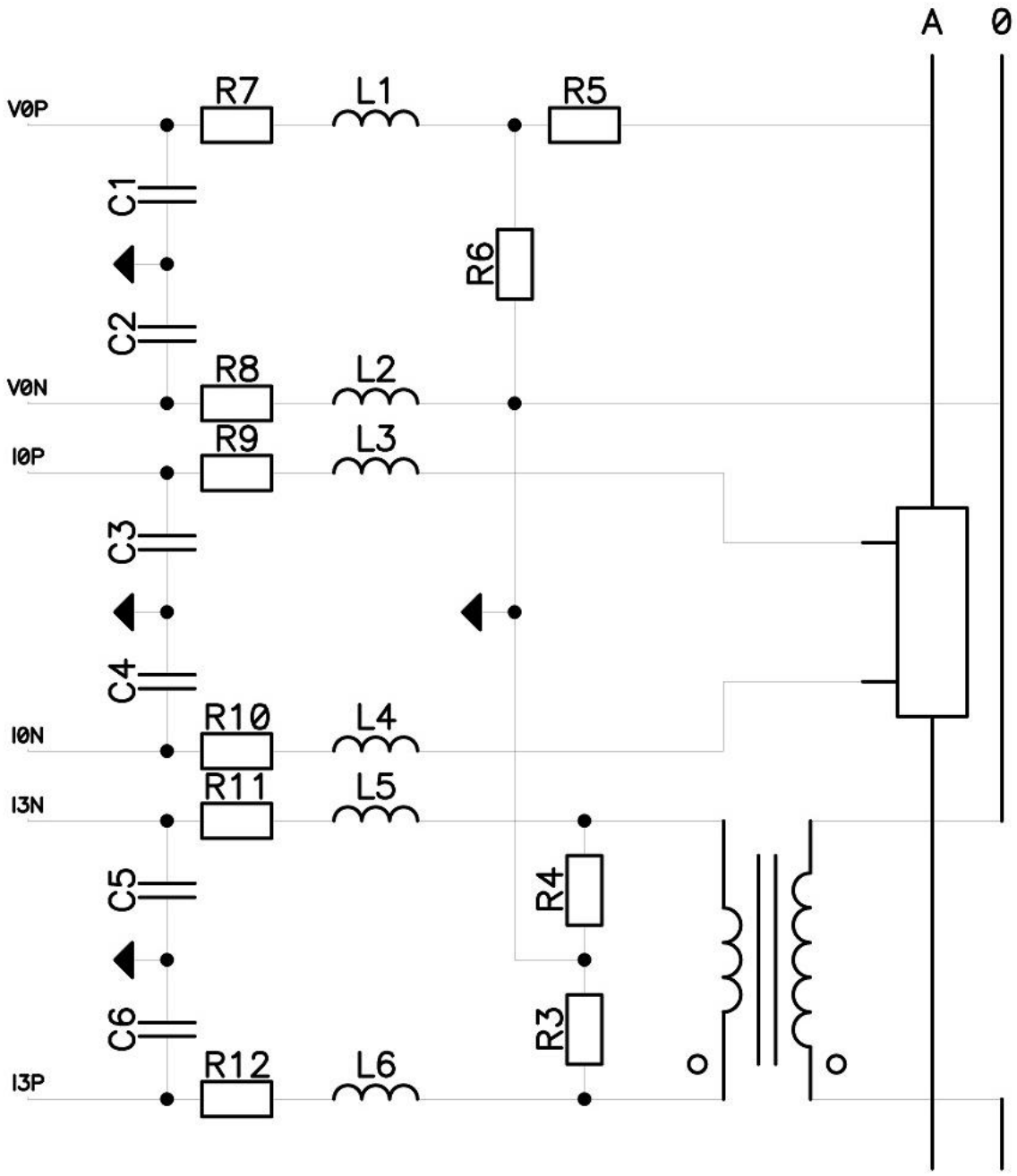


Рисунок 34. Типовая схема включения для учета электроэнергии по одной фазе с использованием шунта

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. Аппаратный блок вычисления CRC

Аппаратный блок вычисления CRC

Микроконтроллер имеет в своем составе блок для вычисления 16 битного CRC с произвольным полиномом. Контроллер принимает 32 битные слова и может их обрабатывать как в прямом порядке (начиная с младшего бита), так и в обратном

(начинаю со старшего бита). Скорость подсчета составляет 2 бита / PCLK (частота APB шины). Контроллер имеет FIFO на 4 отчетов, а так же DMA канал для загрузки новых слов. Запрос для DMA формируется, если в FIFO пусто. Контроллер начинает обрабатывать новые слова, как только они появляются в FIFO и обрабатывает до последнего слова. После обработки последнего слова выставляется флаг. Регистр CRC имеет доступ как на чтение (считать рассчитанное значение) так и на запись (установить начальное значение).

Описание регистров управления блока CRC

Таблица 219 Описание регистров управления блока CRC

Базовый Адрес	**Название**	**Описание**
0x4009_8000	CRC	Контроллер CRC
Смещение		
0x00	CRC_CTRL	Общее управление для контроллера CRC
0x04	CRC_STAT	Статус CRC блока
0x08	CRC_DATAI	Регистр FIFO входных данных
0x0C	CRC_VAL	Регистр подсчитанного CRC
0x10	CRC_POL	Полином для расчета CRC

CRC_CTRL

Таблица 220 Регистр CRC_CTRL

Номер	31…7	6…5	4…3	2	1	0
Доступ		R/W	R/W	R/W	R/W	R/W
Сброс		00	00	0	0	0
	-	**DCSize**	**DLSize**	**DMAEN**	**DATAINV**	**CRCEN**

Таблица 221 Описание бит регистра CRC_CTRL

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...7	-	Зарезервировано
6...5	DCSize	Размер данных при расчете CRC: 00 – вычисление для байт (8 бит), при этом DLSize может быть 00, 01, 10. 01 – вычисление для полуслов (16 бит), при этом DLSize может быть 01, 10. 10 – вычисление для слов (32 бит), при этом DLSize может быть только 10.
4…3	DLSize	Размер загружаемых данных: 00 – байт (8 бит), при этом загружаемый байт записывается в CRC_DATAI[7:0] 01 – полуслово (16 бит), при этом загружаемое полуслово записывается в CRC_DATAI[15:0] 10 – слово (32 бита), при этом загружаемое слово записывается в CRC_DATAI[31:0]
2	DMAEN	Разрешение формирования запроса для DMA: 0 – запрос не формируется; 1 – запрос формируется
1	DATAINV	Порядок вычисления CRC: 0 – начиная с младшего разряда;

		1 – начиная со старшего разряда
0	CRCEN	Разрешение работы блока: 0 – блок выключен; 1 – блок включен

CRC_STAT

Таблица 222 Регистр CRC_STAT

Номер	31:4	3	2	1	0
Доступ		R/W	R	R	R
Сброс		0	0	0	0
	-	**FIFOOVER**	**FIFOEMPTY**	**FIFOFULL**	**CONVCOMP**

Таблица 223 Описание бит регистра CRC_STAT

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...4	-	Зарезервировано
3	FIFOOVER*	Переполнение FIFO: 0 – корректная работа; 1 – была запись в полное FIFO, что привело к потери данных
2	FIFOEMPTY	FIFO пусто: 0 – FIFO имеет по крайне мере одну заполненную ячейку; 1 – FIFO пусто
1	FIFOFULL	FIFO заполнено: 0 – FIFO имеет по крайне мере одну свободную ячейку; 1 – FIFO не имеет свободных ячеек
0	CONVCOMP	Завершение расчета CRC: 0 – расчет идет; 1 – расчет завершен и FIFO пусто или блок отключен

* - сброс бита происходит записью “1” в разряд [3]

CRC_DATAI

Таблица 224 Регистр CRC_DATAI

Номер	31…0
Доступ	W
Сброс	
	DATA_IN

Таблица 225 Описание бит регистра CRC_DATAI

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...0	DATA_IN	Регистр для записи нового отчета в FIFO

CRC_VAL

Таблица 226 Регистр CRC_VAL

Номер	31…16	15…0
Доступ		R/W
Сброс		0000000000000000

	-	**CRCOUT**

Таблица 227 Описание бит регистра CRC_VAL

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...16	-	Зарезервировано
15...0	CRCOUT	Рассчитанное значение /начальное значение. Начальное значение нужно записывать, когда блок отключен или когда закончено преобразование

CRC_POL

Таблица 228 Регистр CRC_POL

Номер	31…17	16…0
Доступ		R/W
Сброс		1000000000000001
	-	**CRCPOL**

Таблица 229 Описание бит регистра CRC_POL

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...17	-	Зарезервировано
16...0	CRC_POL	Полином для расчета CRC. Так как это 16-битное CRC, то младший и старший биты всегда “1” и их нельзя изменить

Ниже приведен результирующий полином:

$$f(x) = x^{16} + x^{CRC\_POL[15]} + x^{CRC\_POL[14]} + \dots + x^{CRC\_POL[2]} + x^{CRC\_POL[1]} + 1$$

Сигналы тактовой частоты

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

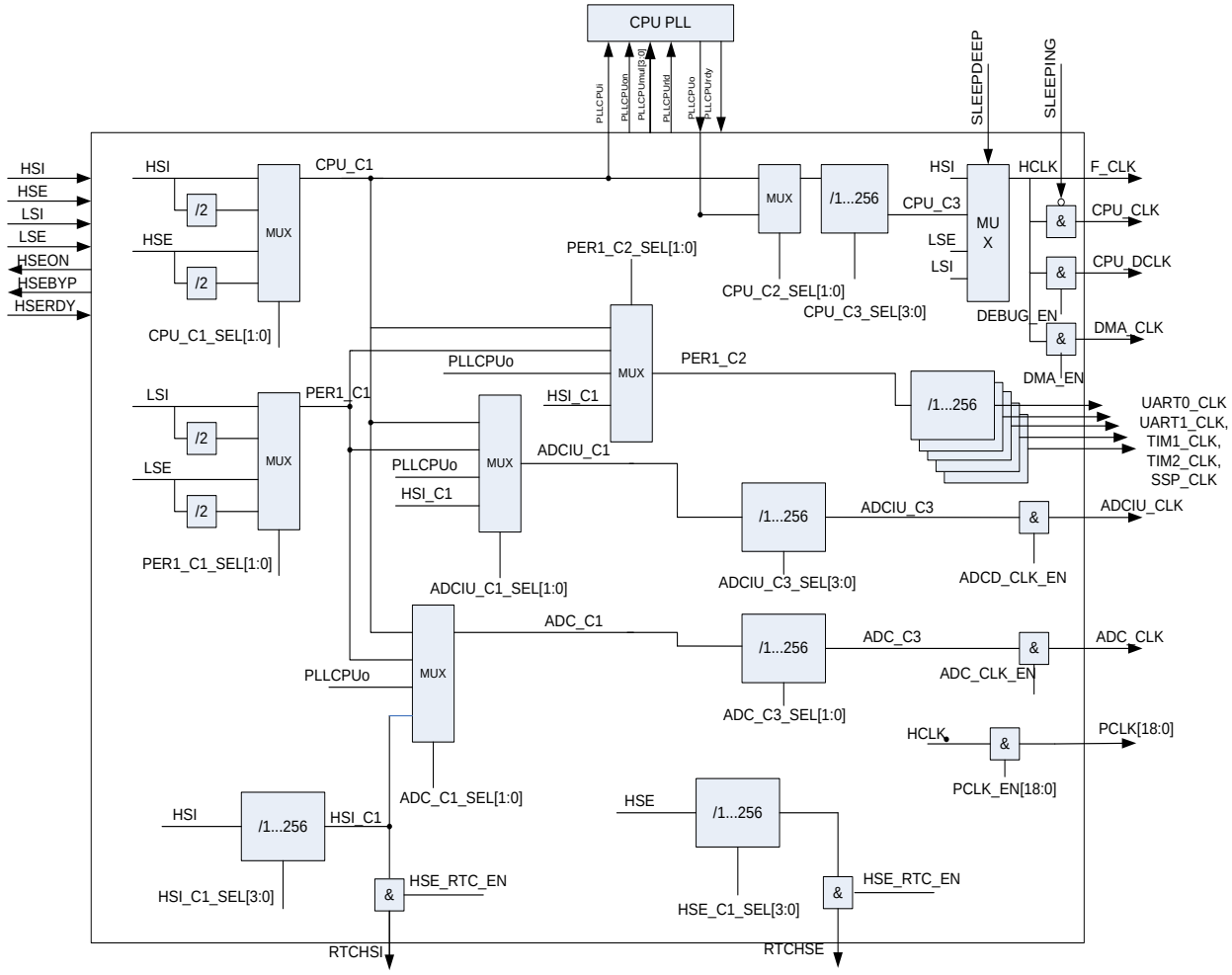


Рисунок 35. Структурная схема

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_0F. Первоначально процессорное ядро запускается на тактовой частоте HSI. При дальнейшей работе генератор HSI может быть отключен при помощи сигнала HSION в регистре BKP_REG_0F. Так же генератор может быть подстроен при помощи сигнала HSITRIM в регистре BKP_REG_0F.

Встроенный RC генератор LSI

Генератор LSI вырабатывает тактовую частоту 40 кГц. Генератор автоматически запускается при появлении питания Ucc и при выходе в нормальный режим работы вырабатывает сигнал LSIRDY в регистре BKP_REG_0F. Первоначально тактовая частота генератор LSI используется для формирования дополнительной задержки tpor. 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. При выходе в нормальный режим работы вырабатывает сигнал HSERDY в регистре CLOCK_STATUS. 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 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, задаваемых на входе PLLCPUMUL[3:0] в регистре PLL_CONTROL. Входная частота блока умножителя должна быть в диапазоне 2…16 МГц 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 мультиплексорах управляемом регистре CPU_CLOCK.

Описание регистров блока контроллера тактовой частоты

Таблица 230 Описание регистров блока контроллера тактовой частоты

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 ∑Δ 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

Таблица 231 Регистр CLOCK_STATUS

Номер	31…3	2	1	0
Доступ	U	RO	RO	RO
Сброс	0	0	0	0
	-	**HSE RDY**	**PLL CPU RDY**	**PLL USB RDY**

Таблица 232 Описание бит регистра CLOCK_STATUS

Разряды	**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

Таблица 233 Регистр PLL_CONTROL

Номер	31…12	11…8	7…4	3	2	1	0
Доступ	U	R/W	U	R/W	R/W	U	U
Сброс	0	0000	0000	0	0	0	0
	-	**PLL CPU MUL[3:0]**	-	**PLL CPU PLD**	**PLL CPU ON**	-	-

Таблица 234 Описание бит регистра PLL_CONTROL

Разряды	**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: PLLCPUo = PLLCPUi x (PLLCPUMUL+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

Таблица 235 Регистр HS_CONTROL

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

Таблица 236 Описание бит регистра HS_CONTROL

Разряды	**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

Таблица 237 Регистр CPU_CLOCK

Номер	31…10	9…8	7…4	3	2	1…0
Доступ	U	R/W	R/W	U	R/W	R/W

Сброс	0	00	0000	0	0	00
	-	**HCLK SEL[1:0]**	**CPU C3 SEL[3:0]**	**-**	**CPU C2 SEL**	**CPU C1 SEL[1:0]**

Таблица 238 Описание бит регистра CPU_CLOCK

Разряды	**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

Таблица 239 Регистр PER1_CLOCK

Номер	31…6	5	4	2…0	1…0
Доступ	U	R/W	R/W	R/W	R/W
Сброс	0	1	1	00	00
	-	**DMA_EN**	**DEBUG_EN**	**PER C2 SEL**	**PER C1 SEL[1:0]**

Таблица 240 Описание бит регистра PER1_CLOCK

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…6	-	Зарезервировано
5	DMA_EN	Бит разрешения тактирования DMA контроллера
4	DEBUG_EN	Бит разрешения тактирования блока отладки ядра
3…2	PER1 C2 SEL	Биты выбора источника для PER1_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

Таблица 241 Регистр 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**	**ADCI UCLK EN**	**ADC C3 SEL[3:0]**	**-**	**ADC C2 SEL[1:0]**	**-**	**ADC C1 SEL[1:0]**

Таблица 242 Описание бит регистра ADC_CLOCK

Разряды	**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 ADC_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

Таблица 243 Регистр RTC_CLOCK

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

Таблица 244 Описание бит регистра RTC_CLOCK

Разряды	**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

Таблица 245 Регистр PER2_CLOCK

Номер	19…0
Доступ	R/W
Сброс	0
	PCLK_EN[19:0]

Таблица 246 Описание бит регистра PER2_CLOCK

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
19…0	PCLK EN[19:0]	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[5] - DMA PCLK[8] - ADC PCLK[9] - WWDT PCLK[10] - IWDT PCLK[11] - POWER PCLK[12] - BKP PCLK[13] - ADCIU PCLK[14] - TIMER1 PCLK[15] - TIMER2 PCLK[16] - PORTA PCLK[17] - PORTB PCLK[18] - PORTC PCLK[19] - CRC

TIM_CLOCK

Таблица 247 Регистр TIM_CLOCK

Номер	31…26	25	24	23…16	15…0	7…0
Доступ	U	R/W	R/W	U	R/W	R/W
Сброс	0	0	0	0	00000000	00000000
	-	**TIM2 CLK EN**	**TIM1 CLK EN**	**-**	**TIM2 BRG [7:0]**	**TIM1 BRG [7:0]**

Таблица 248 Описание бит регистра TIM_CLOCK

Разряды	**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

Таблица 249 Регистр UART_CLOCK

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

Таблица 250 Описание бит регистра UART_CLOCK

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…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 UART: 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

Таблица 251 Регистр SSP_CLOCK

Номер	31…25	24	23…8	7…0
Доступ	U	R/W	U	R/W
Сброс	0	0	0	00000000
	-	**SSP 1 CLK EN**	-	**SSP 1 BRG [7:0]**

Таблица 252 Описание бит регистра SSP_CLOCK

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…25	-	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. Это позволяет, даже в отсутствии основного питания определять его состояние. При снижении питания Ucc в блоке SW происходит автоматическое переключение питания BDUcc с Ucc на BUcc. Если на BUcc имеется отдельный источник питания (батарейка), то батарейный домен остается включенным и может выполнять свои функции.

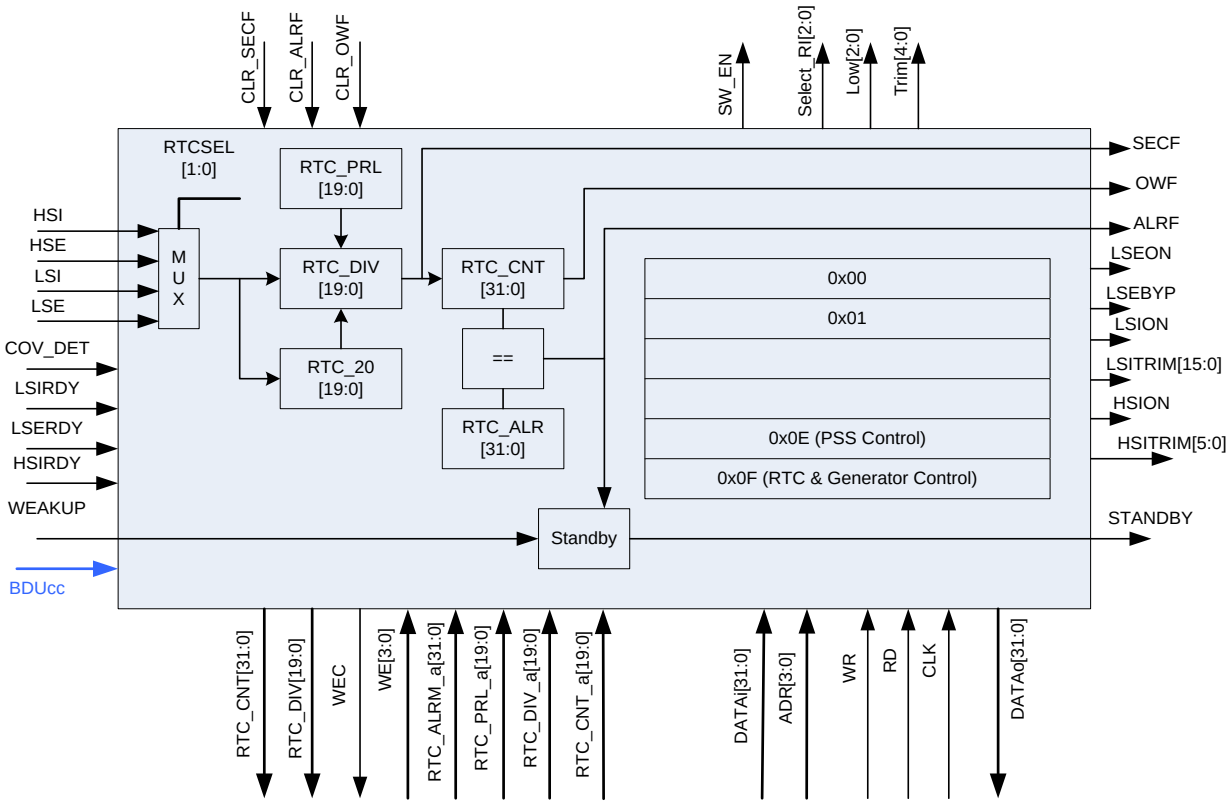


Рисунок 36. Структурная схема

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. 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 DUcc, LSI может быть выбран при наличии питания Ucc, LSE может быть выбран при наличии Ucc или BUcc). Выбор между источниками осуществляется битами RTCSEL. При возможном отключении основного источника питания Ucc в качестве источника тактовой частоты должен использоваться осциллятор LSE, так как он также имеет питание BDUcc. 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[6:0]. Значение CAL определяет, какое число тактов из 2^20 будет замаскировано. Таким образом, с помощью бит CAL производится замедление хода часов. Изменение значения бит CAL может быть осуществлено в ходе работы часов реального времени.

Регистр RTC_DIV 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 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-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 253 Описание регистров блока батарейного домена

Базовый Адрес	**Название**	**Описание**
0x4006_0000	BKP	Контроллер батарейного домена и часов реального времени.
Смещение		
0x00	BKP_REG_00	Регистр аварийного сохранения 0
…		
0x38	BKP_REG_0E	Регистр аварийного сохранения 14
0x3C	BKP_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

Таблица 254 Регистр BKP_REG

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

Таблица 255 Описание бит регистра BKP_REG

Разряды	**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

Таблица 256 Регистр BKP_REG_0E

Номер	31..16	15	14	13..12	11
Доступ		R/W	R/W	R/W	R/W
Сброс		0	0	00	0
	-	**ilimen**	**COVDET**	**Trim[4:3]**	**FPOR**

Номер	10…8	7	6	5…3	2…0
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	000	0	0	000	000
	Trim[2:0]	**JTAG_B**	**JTAG_A**	**SelectRI[2:0]**	**LOW[2:0]**

Таблица 257 Описание бит регистра BKP_REG_0E

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
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 + 0,10 В – значение по умолчанию. 001 – DUcc + 0,06 В 010 – DUcc + 0,04 В 011 – DUcc + 0,01 В 100 – DUcc – 0,01 В 101 – DUcc – 0,04 В 110 – DUcc – 0,06 В 111 – DUcc – 0,10 В
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 со 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

Таблица 258 Регистр BKP_REG_0F

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]**

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**

Таблица 259 Описание бит регистра BKP_REG_0F

Разряды	**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 1,8 В: 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
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
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

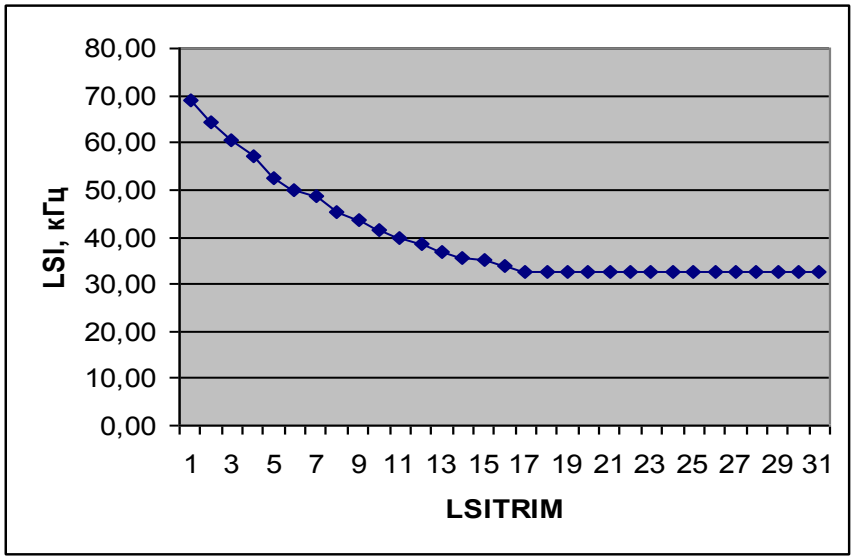


Рисунок 37. Зависимость частоты LSI от значения LSITRIM

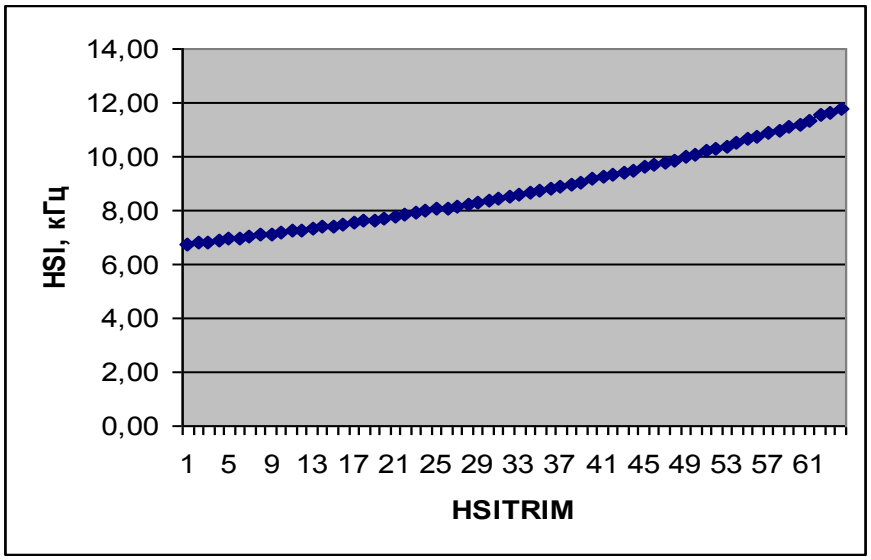


Рисунок 38. Зависимость частоты HSI от значения HSITRIM

RTC_CNT

Таблица 260 Регистр RTC_CNT

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

Таблица 261 Описание бит регистра RTC_CNT

Разряды	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

Таблица 262 Регистр RTC_DIV

Номер	31...20	19…0
Доступ	U	R/W
Сброс	0	0
	-	**RTC DIV [19:0]**

Таблица 263 Описание бит регистра RTC_DIV

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…20	-	-
19…0	RTC DIV [19:0]	Значение счетчика предварительного делителя часов реального времени

RTC_PRL

Таблица 264 Регистр RTC_PRL

Номер	31…20	19…0
Доступ	U	R/W
Сброс	0	0
	-	**RTC PRL [19:0]**

Таблица 265 Описание бит регистра RTC_PRL

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…20	-	-
19…0	RTC PRL [19:0]	Значение основания для счета счетчика предварительного делителя часов реального времени

RTC_ALRM

Таблица 266 Регистр RTC_ALRM

Номер	31…0
Доступ	R/W
Сброс	0
	RTC ALRM[31:0]

Таблица 267 Описание бит регистра RTC_ALRM

Разряды	**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

Таблица 268 Регистр RTC_CS

Номер	31…7	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_IE**	**OWF_IE**	**ALRF**	**SECF**	**OWF**

Таблица 269 Описание бит регистра RTC_CS

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…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 счетчики и регистра RTC_ALRM: 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 переполнения основного счетчика RTC_CNT: 0 – нет переполнения; 1 – было переполнение
2	ALRF	Флаг совпадения основного счетчики и регистра RTC_ALRM: 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. Порты 16-ти разрядные и их выводы мультиплексируются между различными функциональными блоками, управление для каждого вывода порта отдельное. Для того, чтобы выводы порта перешли под управление того или иного периферийного блока, необходимо задать для нужных выводов выполняемую функцию и настройки.

Таблица 270 Порты ввода-вывода

Вывод	**Аналоговая функция ANALOG_EN=0**		**Цифровая функция**		
			Порт IO **MODE=0 ANALOG_EN=1**	**Основная** **MODE=1 ANALOG_EN=1**	
Порт A					
PA0	-		PA0	TMR0_CH1	5
PA1	-		PA1	TMR0_CH1N	
PA2	-		PA2	TMR0_CH2	
PA3	-		PA3	TMR0_CH2N	
PA4	-		PA4	TMR0_CH3	
PA5	-		PA5	TMR0_CH3N	
PA6	-		PA6 SWCLKTCK	TMR0_CH4	
PA7	-		PA7 SWDIO	TMR0_CH4N	
PA8	-		PA8	TMR0_ETR	
PA9	-		PA9	TMR0_BLK	
PA10	-		PA10	EXT_INT0	
PA11	-		PA11	-	
PA12	-		PA12	SSP_FSS	6
PA13	-		PA13	SSP_CLK	
PA14	-		PA14	SSP_RXD	
PA15	-		PA15	SSP_TXD	
Порт B					
PB0	-		PB0 MODE0	UART0_TXD	7
PB1	-		PB1	UART0_RXD	
PB2	-		PB2	nSIROUT0	
PB3	-		PB3	nSIRIN0	
PB4	OSC_IN32	1	PB4	nUART0DTR	
PB5	OSC_OUT32		PB5	nUART0RTS	
PB6	ADC7S	2	PB6	nUART0RI	
PB7	ADC6S		PB7	nUART0DCD	
PB8	ADC5S		PB8	nUART0DSR	
PB9	ADC4S		PB9	nUART0CTS	
PB10	-		PB10	TMR1_CH2	8
PB11	-		PB11	TMR1_CH2N	
PB12	-		PB12	TMR1_CH3	
PB13	-		PB13	TMR1_CH3N	
PB14	-		PB14	TMR1_CH4	
Порт C					

Вывод	**Аналоговая функция ANALOG_EN=0**		**Цифровая функция**		
			Порт IO **MODE=0 ANALOG_EN=1**	**Основная** **MODE=1 ANALOG_EN=1**	
PC0	-		PC0 MODE1	UART1_TXD	9
PC1	ADC3S	3	PC1	UART1_RXD	
PC2	ADC2S/	4	PC2	TMR1_CH1	10
PC3	ADC1S/ADCS_REF+		PC3	TMR1_CH1N	
PC4	ADC0S/ADCS_REF-		PC4	EXT_INT1	
PC5	-		PC5	EXT_INT2	
PC6	-		PC6	TMR1_ETR	11
PC7	-		PC7	TMR1_BLK	

1 – Генератор LSE.
2,3 – АЦП последовательного приближения.
4 – АЦП последовательного приближения.
5 – Таймер 0.
6 – Последовательный интерфейс SSP.
7 – UART0.
8, 10, 12 – Таймер 1.
9 – UART1.

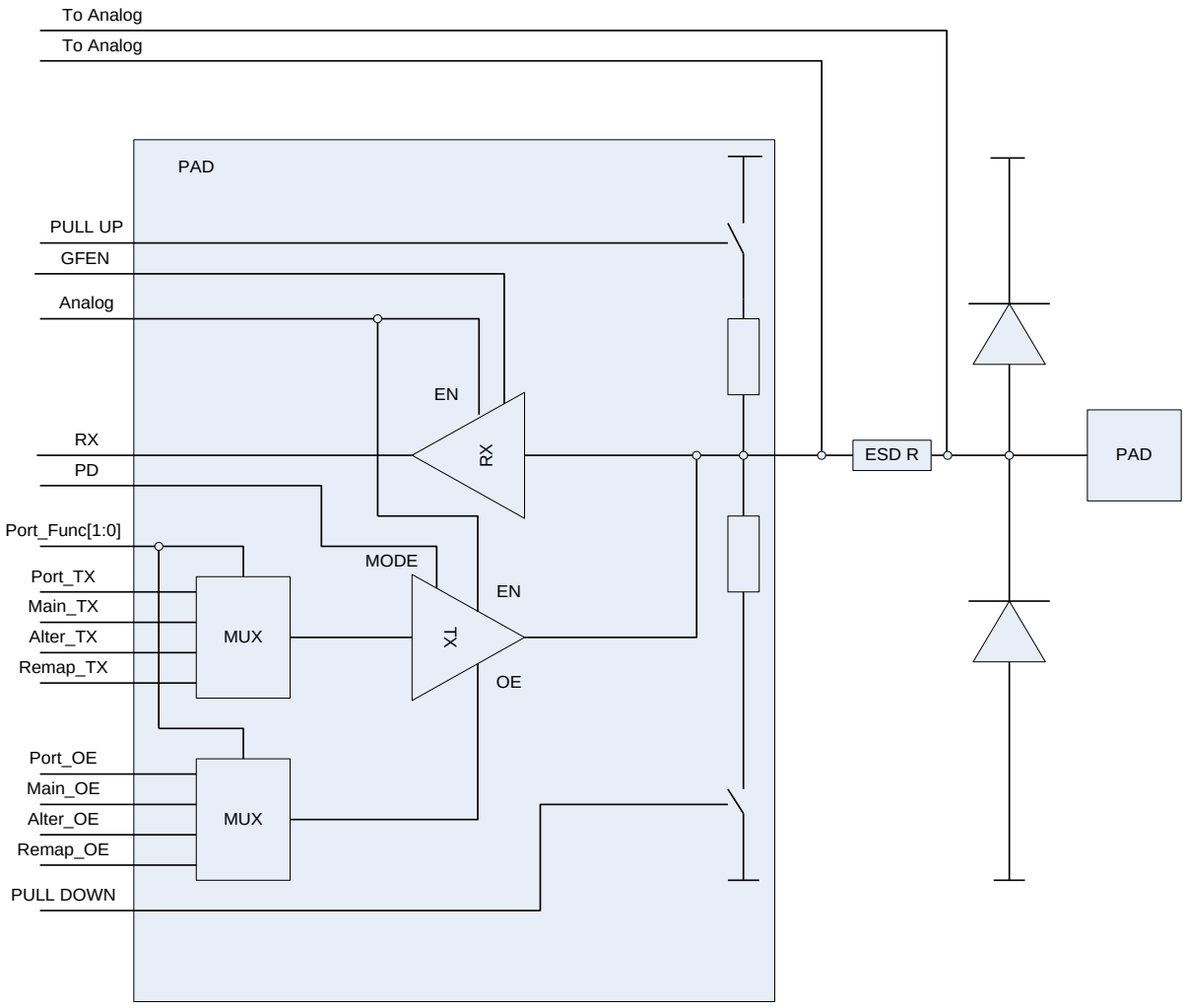


Рисунок 39. Блок схема разряда порта ввода-вывода

Описание регистров портов ввода-вывода

Таблица 271 Описание регистров портов ввода-вывода

Базовый Адрес	**Название**	**Описание**
0x4008_0000	GPIO1	Порт A
0x4008_8000	GPIO2	Порт B
0x4009_0000	GPIO3	Порт C
Смещение		
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

Таблица 272 Регистр PORTx_RXTX

Номер	31..16	15…0
Доступ	U	R/W
Сброс	0	0
	-	**PORT RXTX[15:0]**

Таблица 273 Описание бит регистра PORTx_RXTX

Разряды	**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 выводы порта и для чтения

PORTx_OE

Таблица 274 Регистр PORTx_OE

Номер	31…16	15…0
Доступ	U	R/W
Сброс	0	0
	-	**PORT OE[15:0]**

Таблица 275 Описание бит регистра PORTx_OE

Разряды	**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

Таблица 276 Регистр 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 277 Описание бит регистра PORTx_FUNC

Разряды	**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

Таблица 278 Регистр PORTx_ANALOG

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

Таблица 279 Описание бит регистра PORTx_ANALOG

Разряды	**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

Таблица 280 Регистр PORTx_PULL

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

Таблица 281 Описание бит регистра PORTx_PULL

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...16	PULL UP15:0]	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

Таблица 282 Регистр PORTx_PD

Номер	31…16	15…0
Доступ	U	R/W
Сброс	0	0
	PORT SHM[15:0]	**PORT PD[15:0]**

Таблица 283 Описание бит регистра PORTx_PD

Разряды	**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 Шмитта, выключен гистерезис 200 мВ; 1 – триггер Шмитта, включен гистерезис 400 мВ
15…0	PORT PD[15:0]	Режим работы контроллера. Режим работы выхода: 0 – управляемый драйвер; 1 – открытый сток

PORTx_PWR

Таблица 284 Регистр 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 285 Описание бит регистра PORTx_PWR

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...2	PWRx	Аналогично PWR0 для остальных бит порта
1…0	PWR0[1:0]	Режим работы вывода порта: 00 – зарезервировано; 01 – медленный фронт; 10 – быстрый фронт; 11 – максимально быстрый фронт

PORTx_GFEN

Таблица 286 Регистр PORTx_GFEN

Номер	31…16	15…0
Доступ	U	R/W
Сброс	0	0
	-	**GFEN[15:0]**

Таблица 287 Описание бит регистра PORTx_GFEN

Разряды	**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 последующей 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.

Параметр	**Не менее**	**Типовое**	**Не более**
Входное напряжение, Ucc, В	3,0	-	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 PBLS = “00”, В		1,8	
Уровень срабатывания PBVD от BUcc, при PBLS = “01”, В		2,2	
Уровень срабатывания PBVD от BUcc, при PBLS = “10”, В		2,6	
Уровень срабатывания PBVD от BUcc, при PBLS = “11”, В		3,0	

Описание регистров блока PVD

Таблица 288 Описание регистров блока PVD

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

PVDCS

Таблица 289 Регистр PVDCS

Номер	31…13	12	11	10
Доступ	U	R/W	R/W	R/W
Сброс	0	0	0	0
	-	**PVDBEN**	**INV**	**INVB**

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
Сброс	0	0	0	0	000	00	0
	IEPVD	**IEPVBD**	**PVD**	**PVBD**	**PLS[2:0]**	**PBLS[1:0]**	**PVDEN**

Таблица 290 Описание бит регистра PVDCS

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...13	-	Зарезервировано
12	PVDBEN	Бит разрешения работы блока датчика напряжения питания BUcc: 0 – датчик отключен; 1 – датчик включен
11	INV	Флаг инверсии выхода от датчика PVD: 0 – нет инверсии; 1 – есть инверсия. Если флаг не инвертируется, то он выставляется при превышении заданного уровня. Если инвертируется - то при снижении ниже заданного уровня
10	INVB	Флаг инверсии выхода от датчика PVBD: 0 – нет инверсии; 1 – есть инверсия. Если флаг не инвертируется, то он выставляется при превышении заданного уровня. Если инвертируется - то при снижении ниже заданного уровня
9	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.
8	IEPVBD	Флаг разрешения прерывания от датчика PVBD: 0 – прерывание запрещено; 1 – прерывание разрешено. Очищается записью 0. Если при очистке датчик продолжает выдавать сигнал, то флаг не будет очищен.

7	PVD	Результат сравнения напряжения основного питания: 0 – напряжение питания меньше, чем уровень задаваемый PLS; 1 – напряжение питания больше, чем уровень задаваемый PLS
6	PVBD	Результат сравнения напряжения батарейного питания: 0 – напряжение питания меньше, чем уровень задаваемый PBLS; 1 – напряжение питания больше, чем уровень задаваемый PBLS
5…3	PLS[2:0]	Уровень напряжения для сравнения с напряжением основного питания: 000 – 2,0 В 001 – 2,2 В 010 – 2,4 В 011 – 2,6 В 100 – 2,8 В 101 – 3,0 В 110 – 3,2 В 111 – 3,4 В
2…1	PBLS[1:0]	Уровень напряжения для сравнения с напряжением батарейного питания: 00 – 1,8 В 01 – 2,2 В 10 – 2,6 В 11 – 3,0 В
0	PVDEN	Бит разрешения работы блока датчика напряжения питания Ucc: 0 – датчик отключен; 1 – датчик включен

Таймеры общего назначения

Все блоки таймеров выполнены на основе 16-разрядного перезагружаемого счетчика, который синхронизируется с выхода 16-разрядного 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- разрядный счетчик, 16-разрядный предделитель частоты и 4-х 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 четырехканальный блок захвата/сравнения. Данный блок выполняет, как стандартные функции захвата и сравнения, так и ряд специальных функций. Таймеры имеют четыре канала схем захвата и ШИМ с функциями формирования «мертвой зоны» и аппаратной блокировки. Каждый из таймеров может генерировать прерывания и запросы ПДП.

Особенности:

- 16- разрядный вверх, вниз, вверх / вниз счетчик;
- 16-разрядный программируемый предварительный делитель частоты;
- до четырех независимых 16- разрядных каналов захвата на один таймер. Каждый из каналов захвата может захватить (скопировать) текущее значение таймера при изменении некоторого входного сигнала. В случае захвата имеется дополнительная возможность генерировать прерывание и/или запрос DMA.
- четыре 16- разрядных регистра сравнения (совпадения), которые позволяют осуществлять непрерывное сравнение, с дополнительной возможностью генерировать прерывание и/или запрос DMA при совпадении;
- имеется до четырех внешних выводов, соответствующих 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 Fdts или какие-либо внешние события и произвольно генерировать прерывания, запросы DMA или выполнять другие действия. Значения таймера, при достижении которых будут выполнены те или иные действия, задаются восьмью регистрами совпадения. Кроме того, в микроконтроллере имеются четыре входа захвата, чтобы захватить значение таймера при изменении некоторого входного сигнала, с возможностью генерировать прерывание или запрос DMA.

Структурная схема блока Таймер представлена на рисунке ниже.

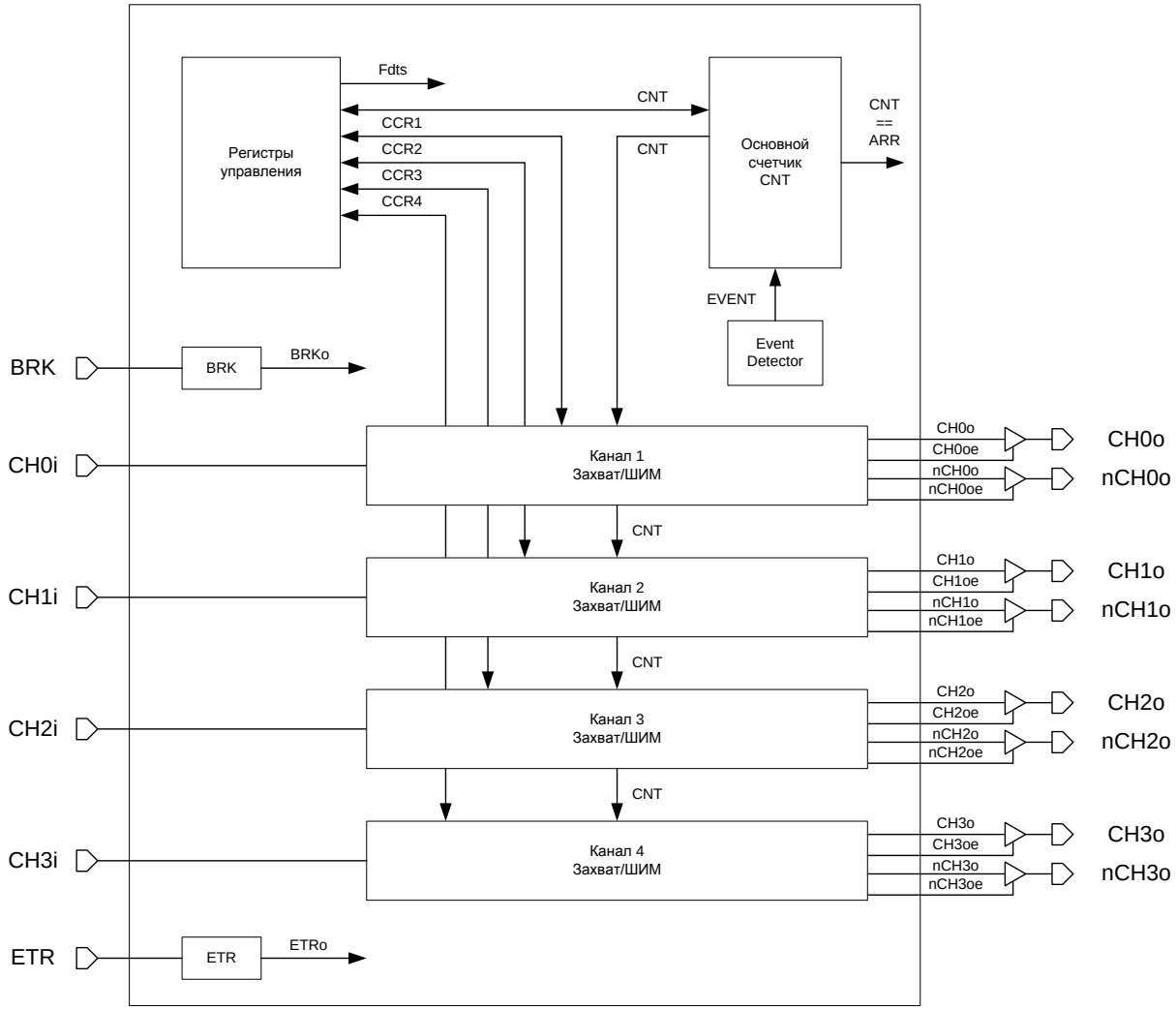


Рисунок 40. Структурная схема таймера

Таймер содержит основной 16-ти битный счетчик CNT, блок регистров управления и четыре канала схем 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 сброса и тактовой частоты».

Для задания тактовой частоты блока необходимо установить бит разрешения тактирования блока (бит 14 для таймера1, бит 15 для таймера2 регистра PER2_CLOCK). В регистре TIM_CLOCK установить бит TIMyCLKEN, чтобы разрешить подачу тактовой частоту для определенного таймера, задать коэффициент деления тактовой частоты HCLK для каждого таймера.

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

Режим таймера.

Таймеры построены на базе 16-разрядного счетчика, объединенного с 16- разрядным 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 состояния основного счетчика CNT может служить как внутренняя частота TIM_CLK, так и события в других счетчиках, либо события на линиях TxCHO основного счетчика.

Чтобы запустить работу основного счетчика таймера необходимо задать:

- Начальное значение основного счетчика– TIMx_CNT;
- Значение предварительного делителя счетчика – TIMx_PSG, при этом счетчик будет считать на частоте CLK= TIMx_CLK/(PSG+1);
- Значение основания счета для основного счетчика TIMx_ARR.

Режим работы счетчика TIMx_CNTRL. Необходимо:

- выбрать источник события переключения счетчика EVENT_SEL;
- выбрать режим счета основного счетчика CNT_MODE (значения 00 и 01 при тактировании внутренней частотой, значения 10 и 11 при тактировании внешними сигналами);
- выбрать направление счета основного счетчика DIR;
- разрешить работу счетчика CNT_EN.

По событиям совпадения значения основного счетчика со значением нуля или значением основания счета генерируется прерывание и запроса DMA, которые могут быть замаскированы.

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.
```

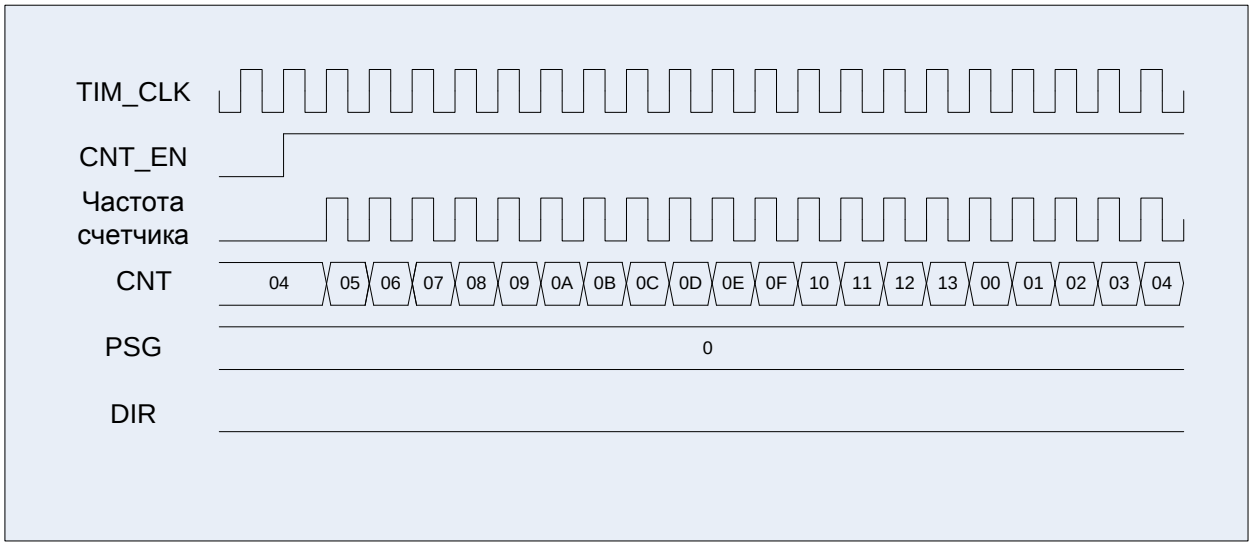


Рисунок 41. Диаграммы работы таймера, счет вверх от 0 до 0x0013, стартовое значение 0x0004

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.
```

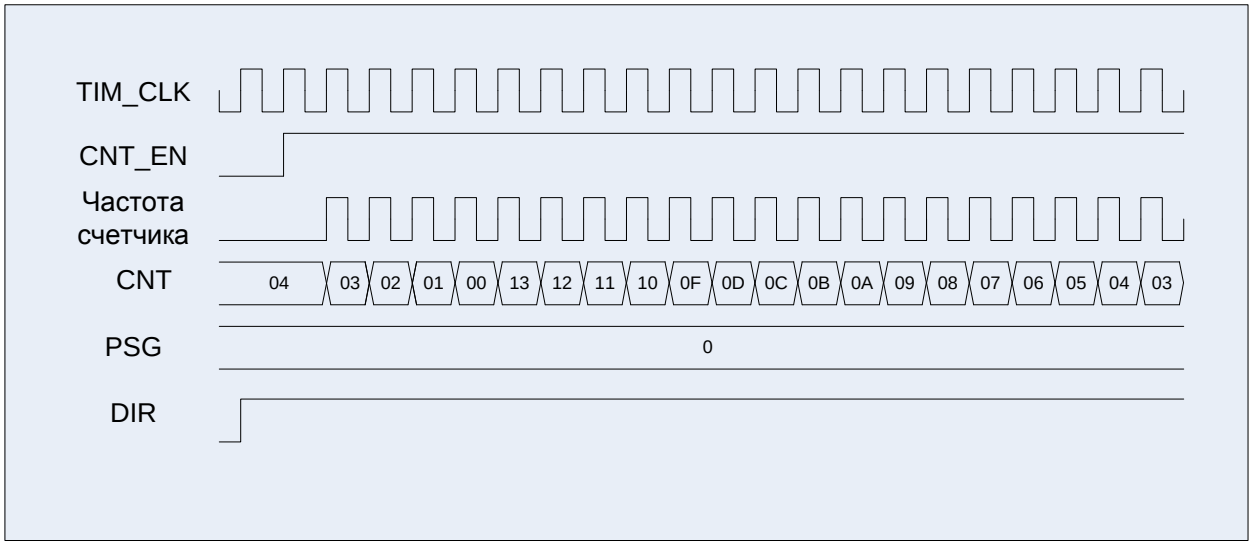


Рисунок 42. Диаграммы работы таймера, счет вниз от 0x0013 до 0, стартовое значение 0x0004

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.
```

Рисунок 43. 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.
```

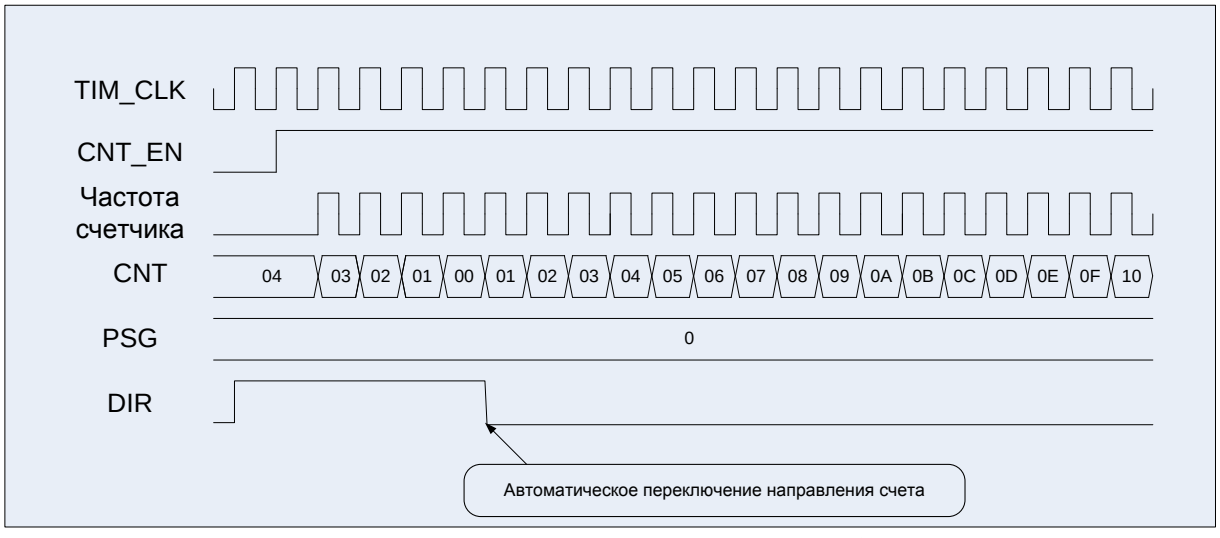


Рисунок 44. Диаграммы работы таймера, счет вверх/вниз, сначала вниз

Источники 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 TxCHO основного счетчика.
- Внешний тактовый сигнал, режим 2: События на линиях TxCHO основного счетчика.
- Внешний тактовый сигнал, режим 3: События на входе ETR основного счетчика.

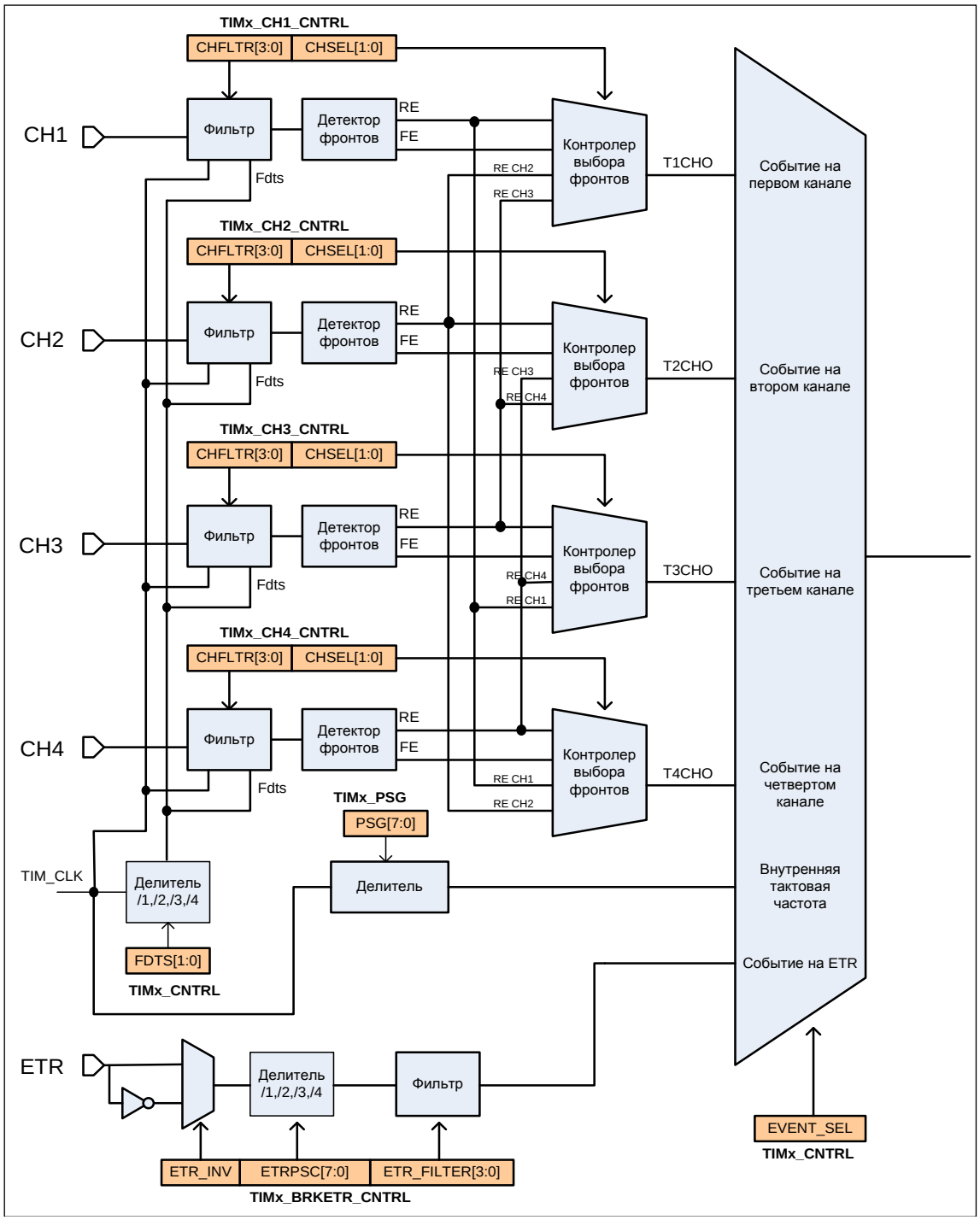


Рисунок 45. Структурная схема формирования события для счета

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 вступят 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 измениться на 1 и снова начинается счет. Поле DIR определяет, в какую сторону будет меняться значение счетчика: DIR = 0 - счетчик считает вверх (см. Рисунок 46), DIR = 1 – счетчик считает вниз (см. Рисунок 47). Если CNT_MODE = 00, то направление счета определяется полем DIR, если CNT_MODE = 01, счетчик считает вверх/вниз с автоматическим изменением DIR (см. Рисунок 48).

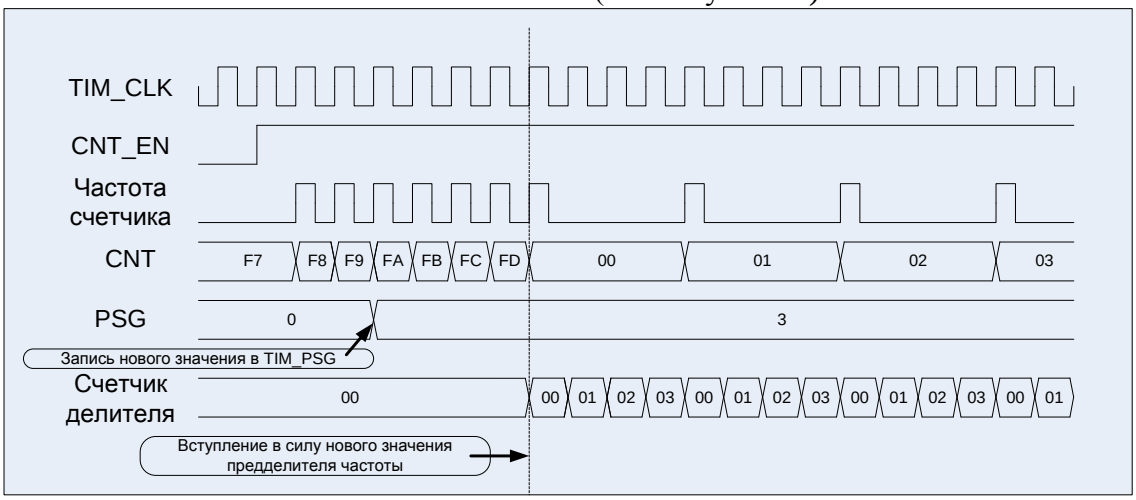


Рисунок 46. Диаграммы работы счетчика: счет вверх (CNT_MODE = 00, EVENT_SEL = 0000, DIR = 0)

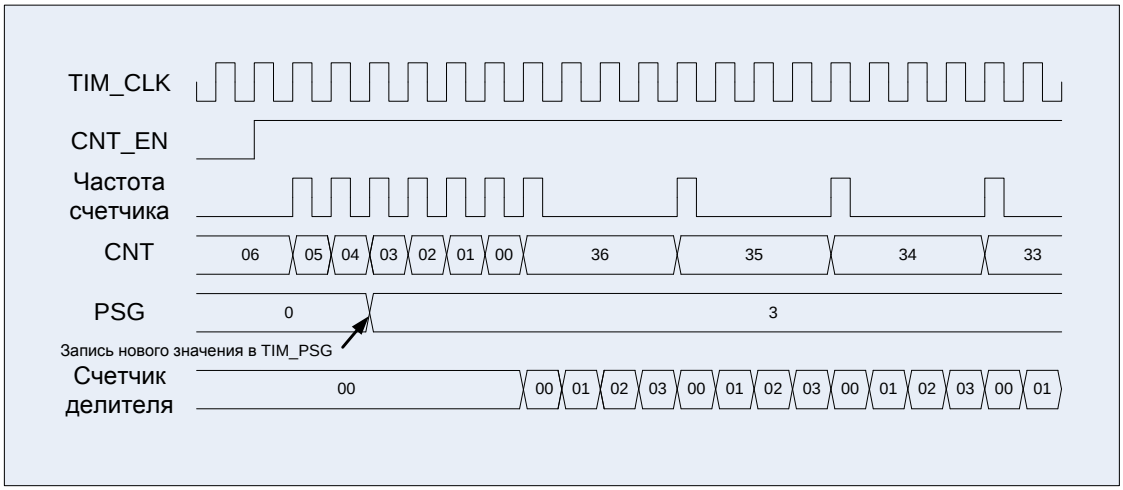


Рисунок 47. Диаграммы работы счетчика: счет вниз (CNT_MODE = 00, EVENT_SEL = 0000, DIR = 1)

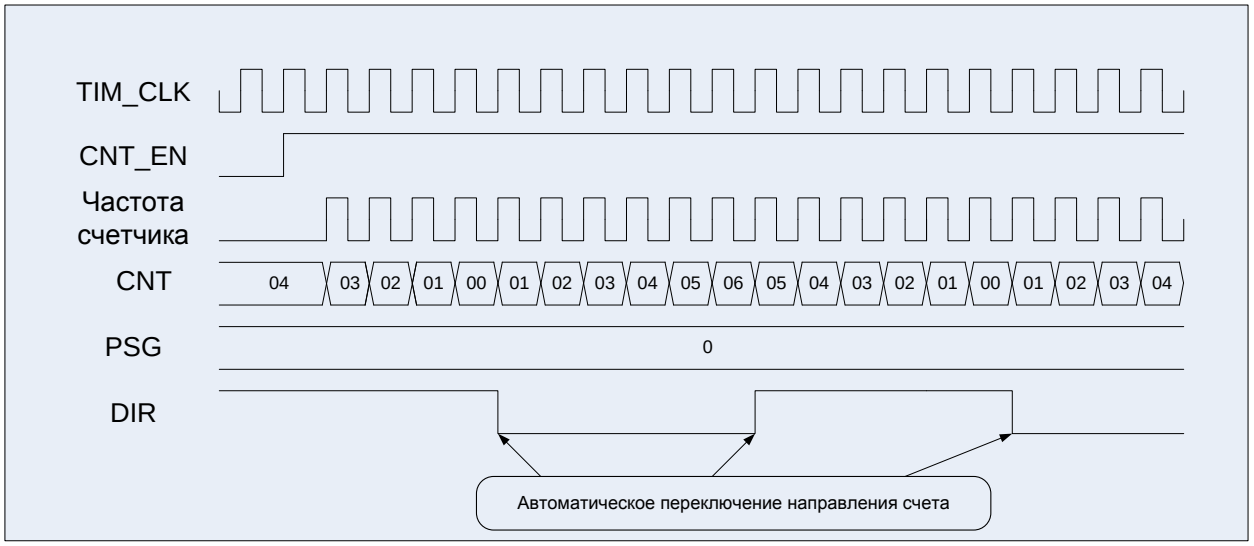


Рисунок 48. Диаграммы работы счетчика: счет вниз/вверх (CNT_MODE = 01, EVENT_SEL = 0000, DIR = 1)

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 трех таймеров, а также внешние входы, связанные с выводами блоков захвата/сравнения.

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

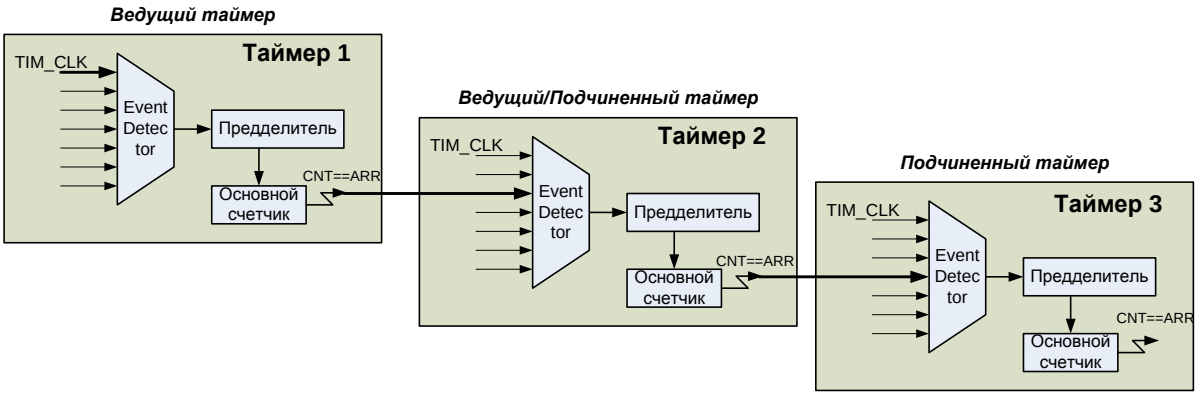


Рисунок 49. Пример каскадного соединения таймеров

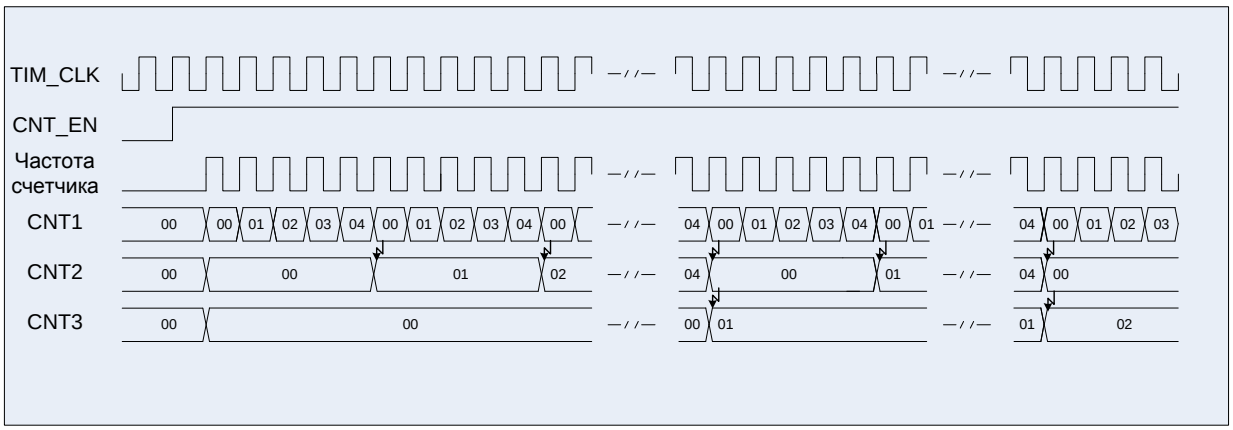


Рисунок 50. Диаграммы работы трех таймеров в каскаде

DIR_1, DIR_2, DIR_3 = 0;
EVENT_SEL_1 = 0000, EVENT_SEL_2 = 0001, EVENT_SEL_3 = 0010;
CNT_MODE_1, CNT_MODE_2, CNT_MODE_3 = 00;

Внешний тактовый сигнал, режим 1.
События на линиях TxCHO основного счетчика.

Этот режим выбирается, когда EVENT_SEL = 01xx в регистре TIMx_CNTRL. Счетчик может считать по положительному или отрицательному фронту на выбранном входе, или по положительному фронту на других каналах (см. рис.) 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 FDTS. Частота выборки данных задается в регистре TIMx_CNTRL в поле FDTS.

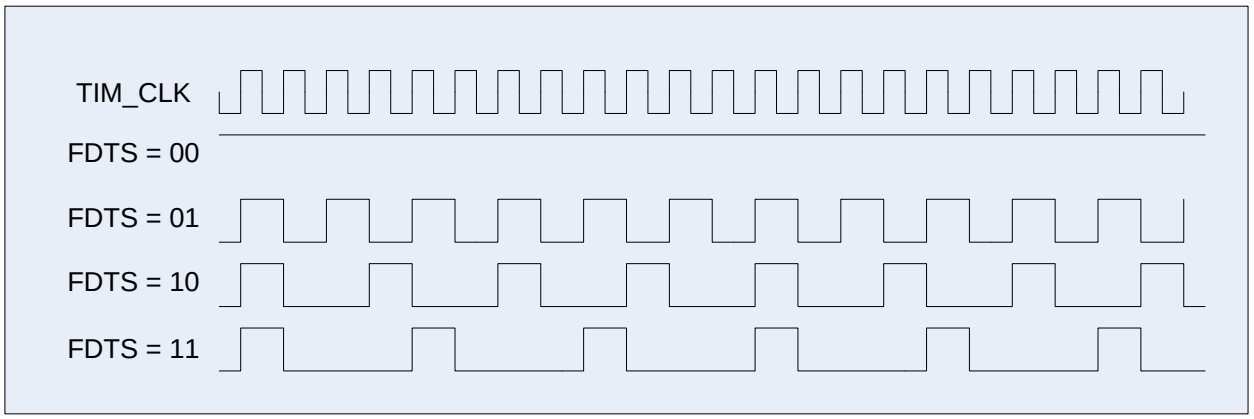


Рисунок 51. Диаграммы возможных частот выборки данных (FDTS)

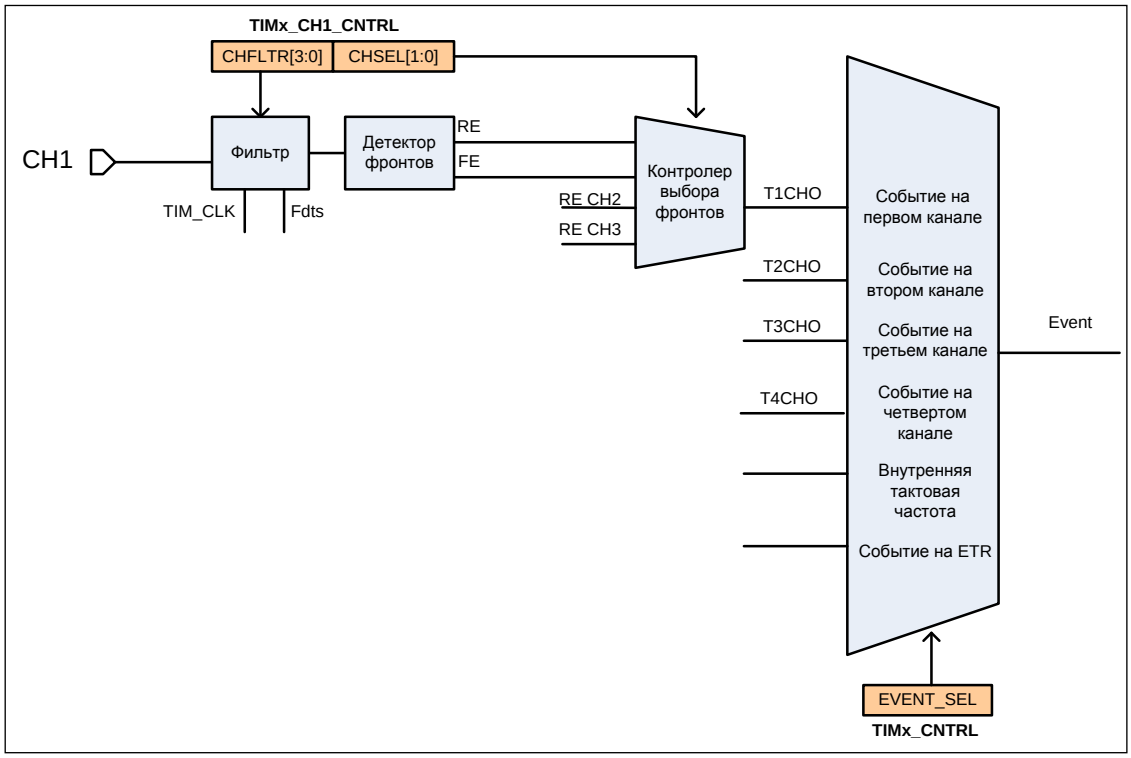


Рисунок 52. Тактирование с входа первого канала

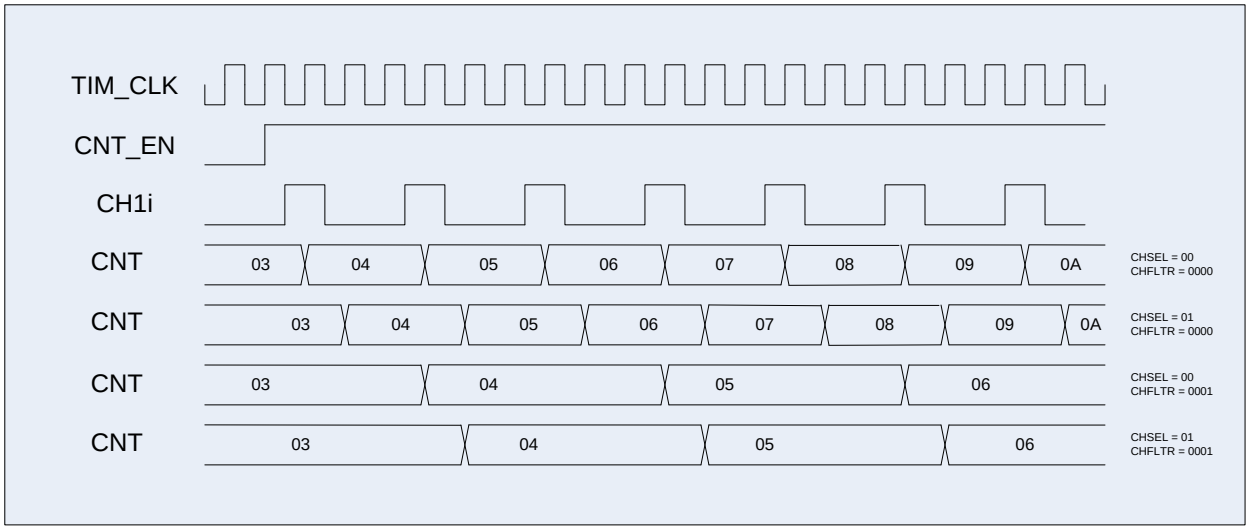


Рисунок 53. Диаграмма внешнего тактирования с разными вариантами фильтра

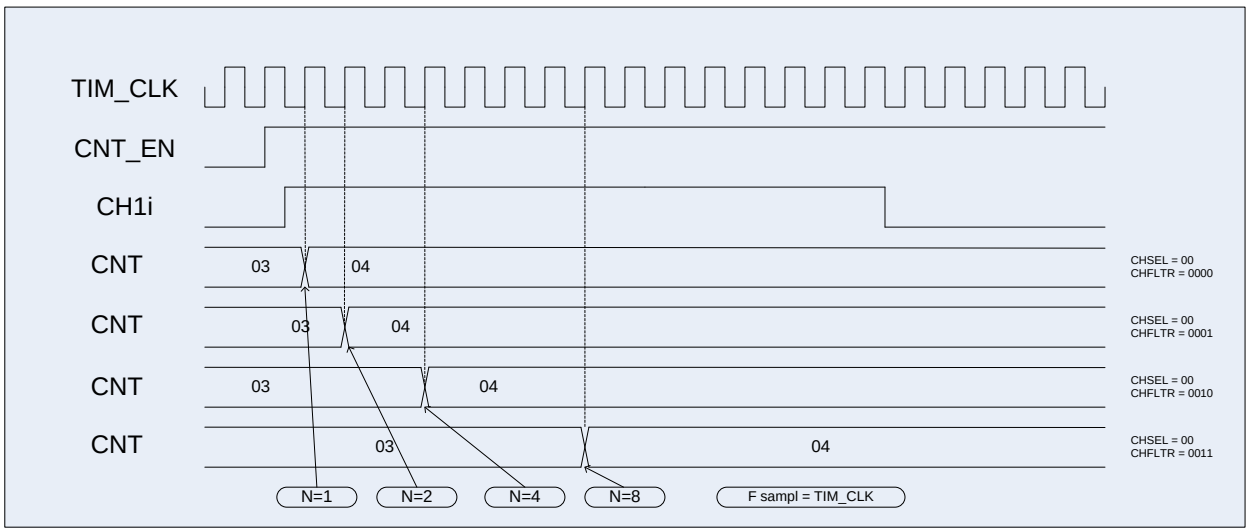


Рисунок 54. Диаграмма внешнего тактирования с разными вариантами фильтра

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) для данного входа тактовой частоты, а также использовать инверсию входа.

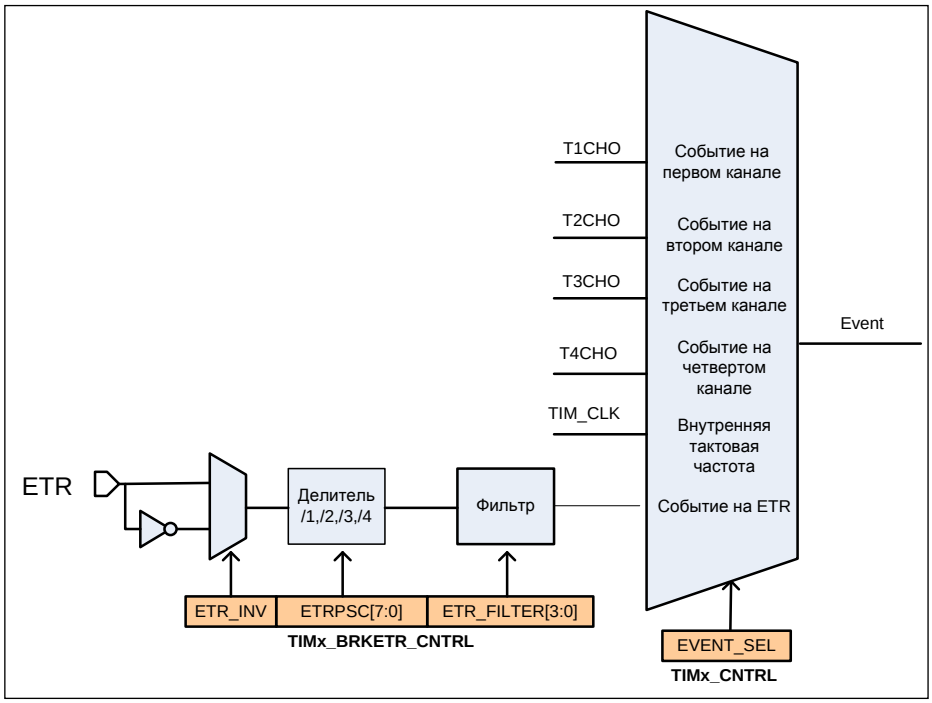


Рисунок 55. Схема тактирования сигналом со входа ETR

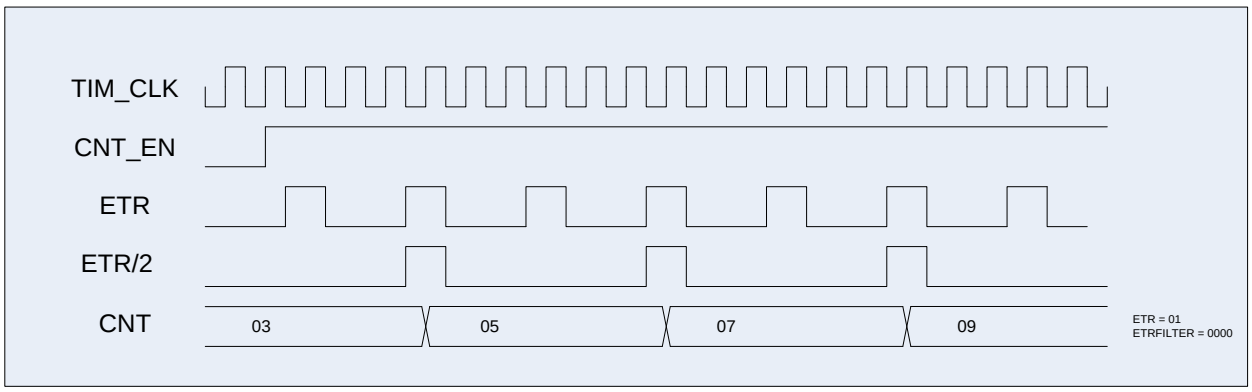


Рисунок 56. Диаграмма тактирования сигналом со входа ETR

Режим захвата

Структурная схема блока Захвата представлена на рисунке.

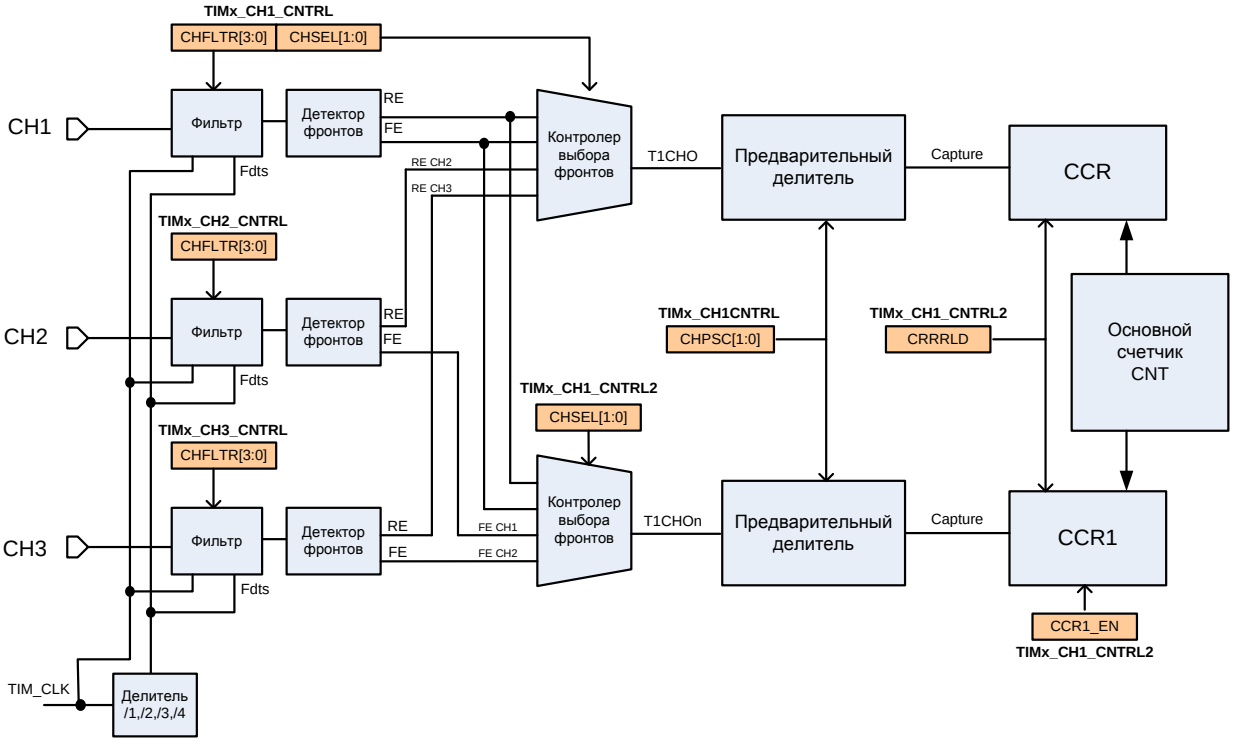


Рисунок 57. Структурная схема блока захвата на примере канала 1

Для включения режима захвата для определенного канала необходимо в регистре управления каналом TIMx_CHy_CNTRL записать 1 в поле CAPnPWM. Для регистрации событий по линии CHxi 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. На блоке MUX производится выбор используемого для Захвата сигнала между положительным фронтом канала, отрицательным фронтом канала и положительными и отрицательными фронтами сигналов от других каналов. После блока MUX может быть использован предварительный делитель 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.

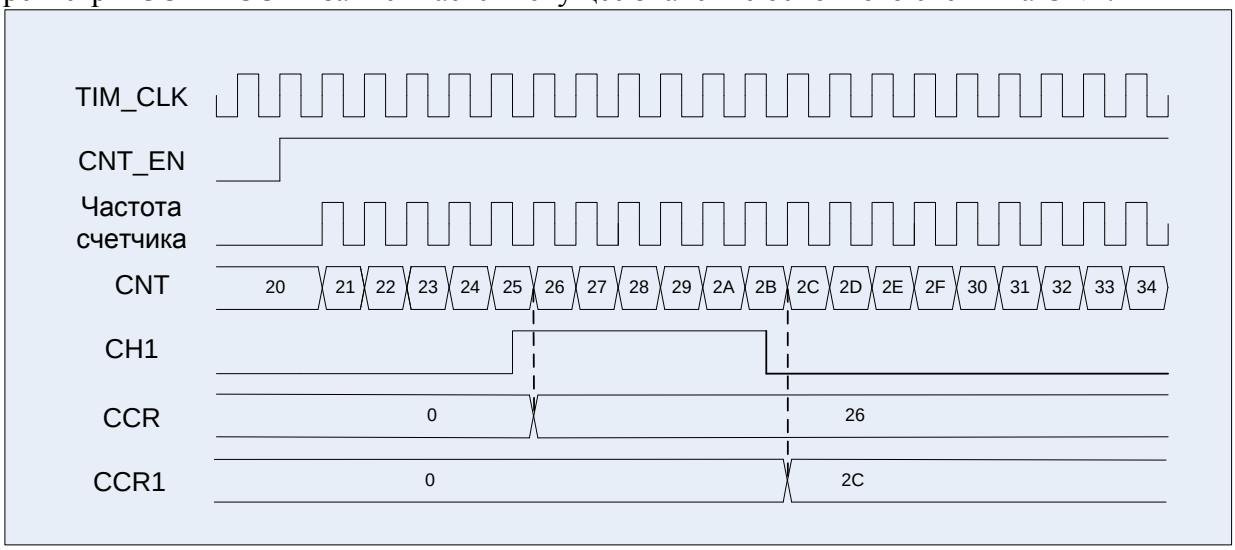


Рисунок 58. Диаграмма захвата события со входа первого канала

На рисунке показан пример захвата значения основного счетчика в регистр CCR по положительному фронту на входе канала и в регистр CCR1 по отрицательному фронту на входе канала. В регистре TIMx_IE можно разрешить выработку прерываний по событию захвата на определенном канале, а в регистре TIMx_DMA_RE можно разрешить формирование запросов DMA.

Режим ШИМ

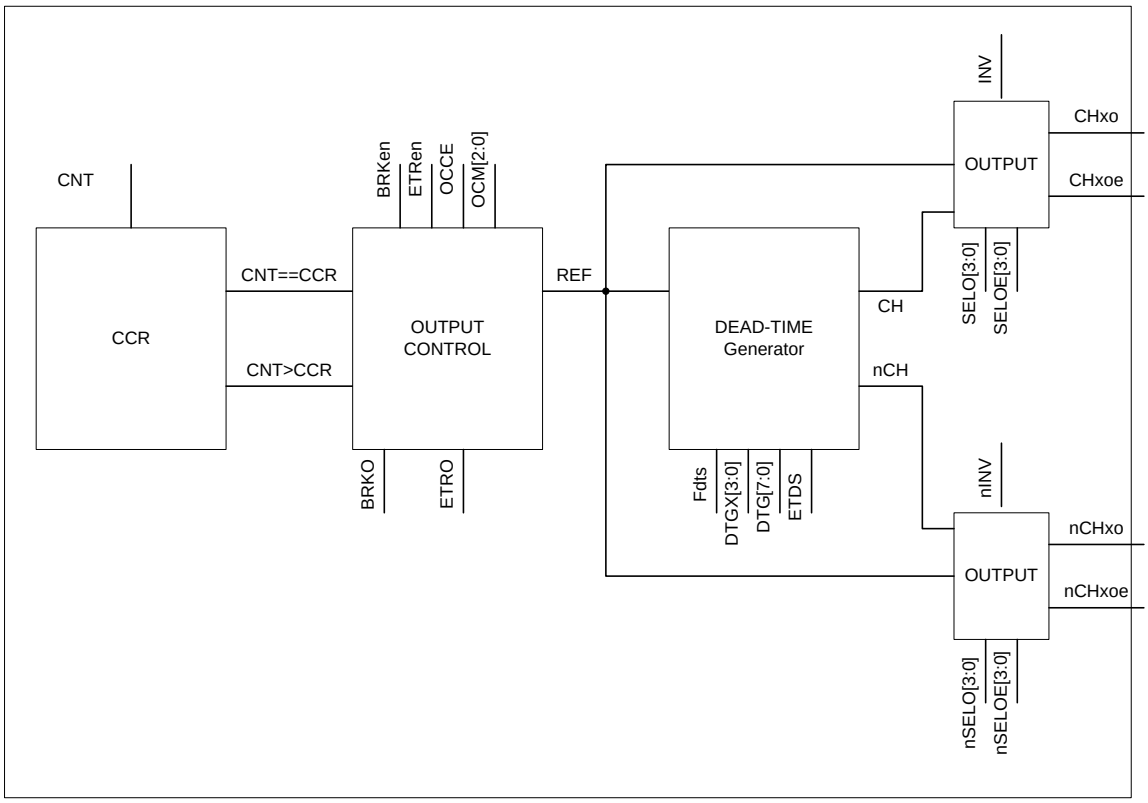


Рисунок 59. Структурная схема блока сравнения

Для включения режима сравнения для определенного канала необходимо в регистре управления каналом TIMx_CHy_CNTRL записать 0 в поле CAPnPWM. При работе в режиме ШИМ выходной сигнал может формироваться на основании сравнения значения в регистре CCR и основного счетчика CNT или регистров CCR, CCR1 и значения основного счетчика CNT. Полученный сигнал может без изменения выдаваться на выводы CHxO и nCHxO. Либо с применением схемы DEAD TIME Generator формируются управляющие сигналы с мертвой зоной. У каждого 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.

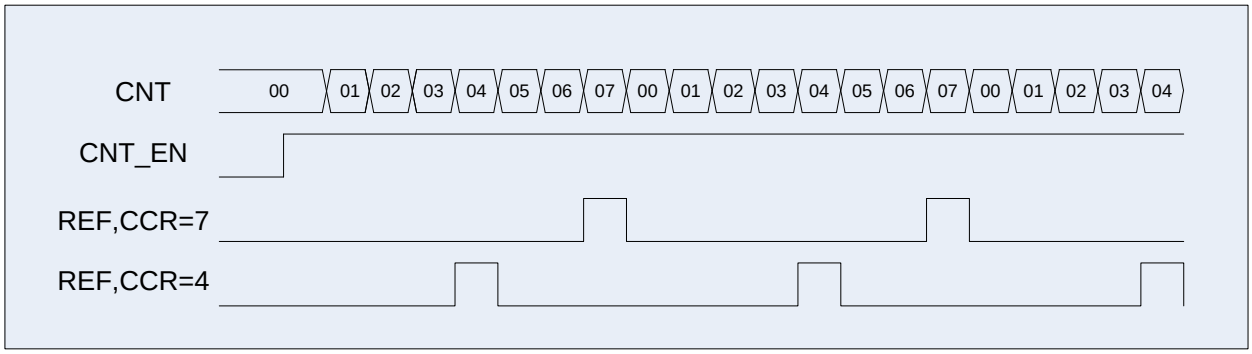


Рисунок 60. Диаграмма работы схемы в режиме ШИМ, CCR1_EN=0

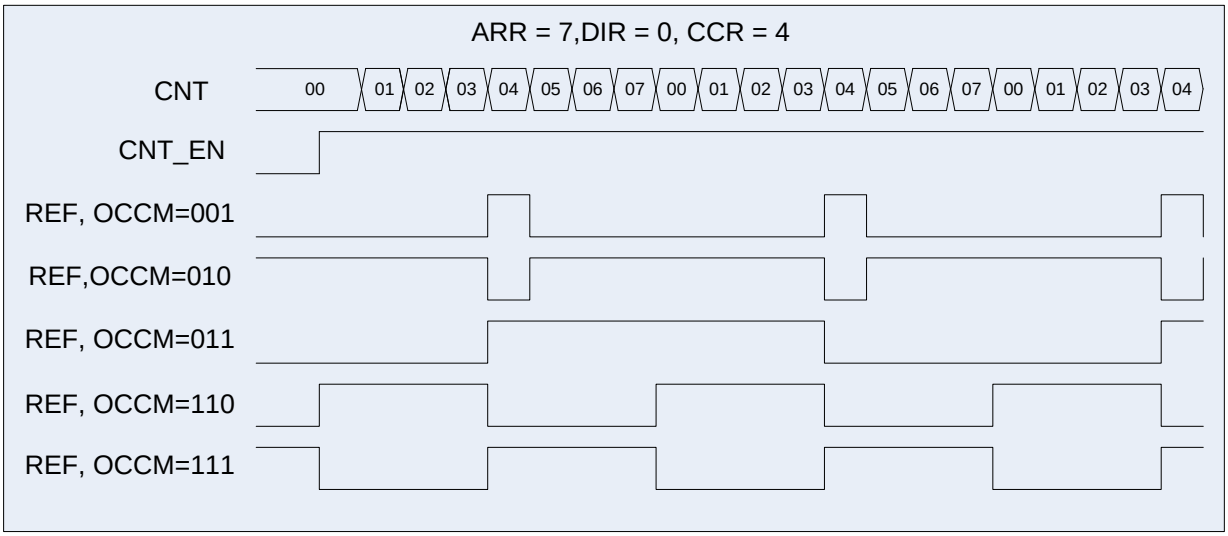


Рисунок 61. Диаграмма работы схемы в режиме ШИМ, CCR1_EN=0

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

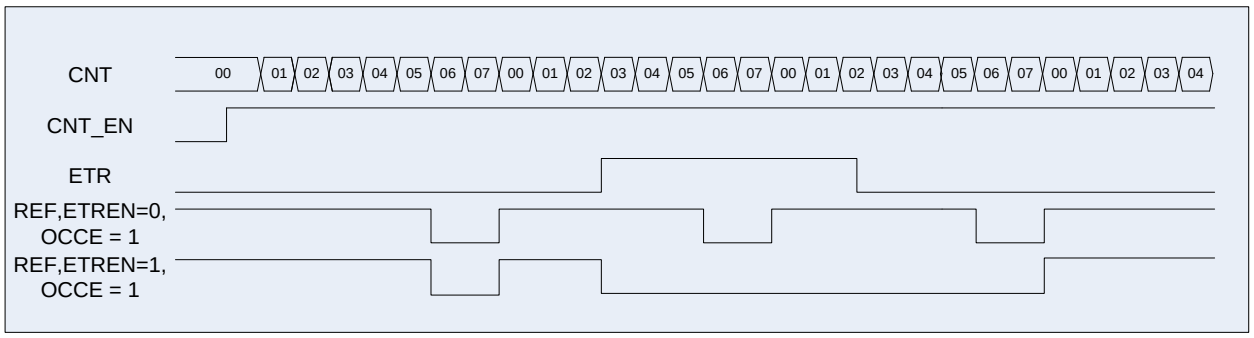


Рисунок 62. Диаграмма работы схемы в режиме ШИМ, CCR1_EN = 0

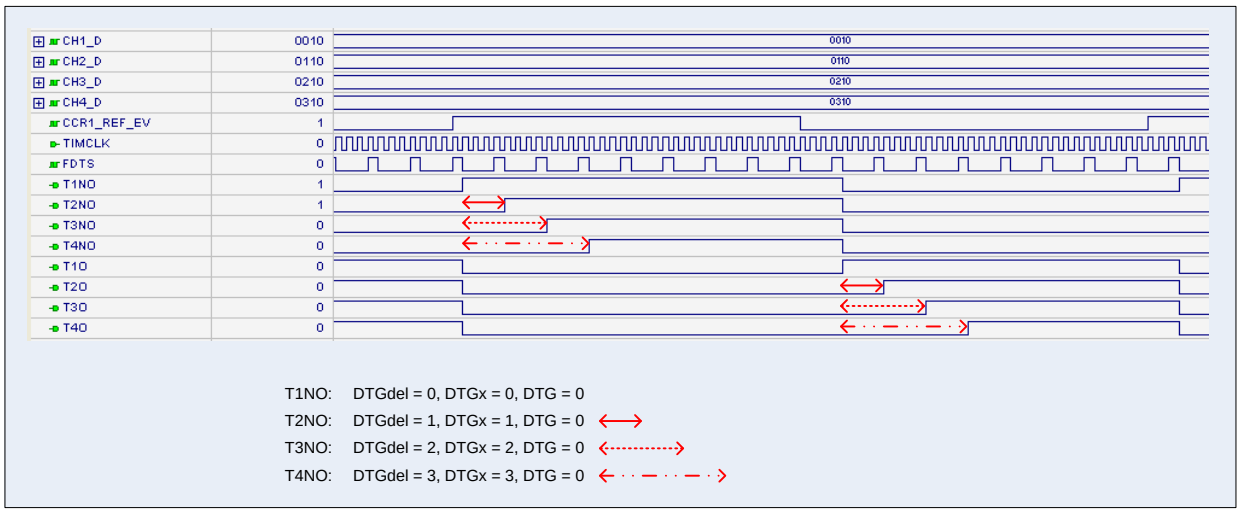


Рисунок 63. Диаграмма работы схемы DTG

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.

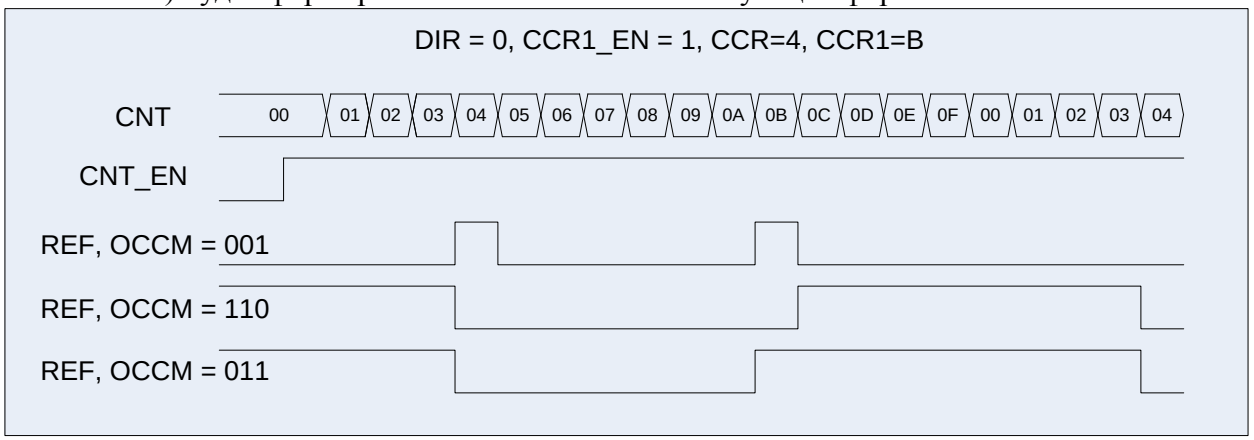


Рисунок 64. Диаграмма работы схемы в режиме ШИМ, CCR1_EN = 1

При записи новых значений CCR и CCR1, если установлен бит CRRRLD, то регистры CCR1 и CCR получат новые значения только при CNT = 0, иначе запись осуществляется немедленно. Факт окончания записи обозначается установкой 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.

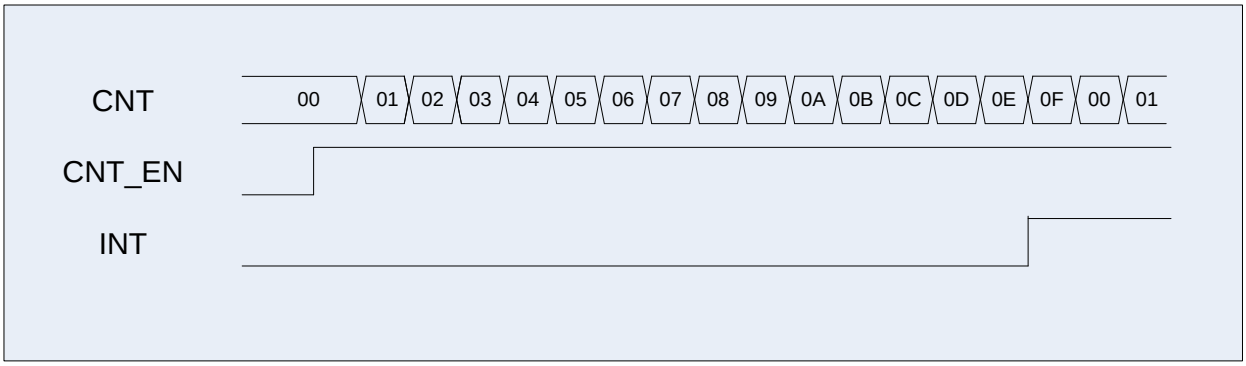


Рисунок 65. Диаграммы примера работы в режиме обычного счетчика

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 по всем каналам
//Режим работы каналов - захват
TIMx->TIMx_CHy_CNTRL[0] = 0x00008000;
TIMx->TIMx_CHy_CNTRL[1] = 0x00008002;
TIMx->TIMx_CHy_CNTRL[2] = 0x00008001;
TIMx->TIMx_CHy_CNTRL[3] = 0x00008003;

//Режим работы выхода канала – канал на выход не работает
TIMx->TIMx_CHy_CNTRL1[0]= 0x00000000;
TIMx->TIMx_CHy_CNTRL1[1]= 0x00000000;
TIMx->TIMx_CHy_CNTRL1[2]= 0x00000000;
TIMx->TIMx_CHy_CNTRL1[3]= 0x00000000;
```

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

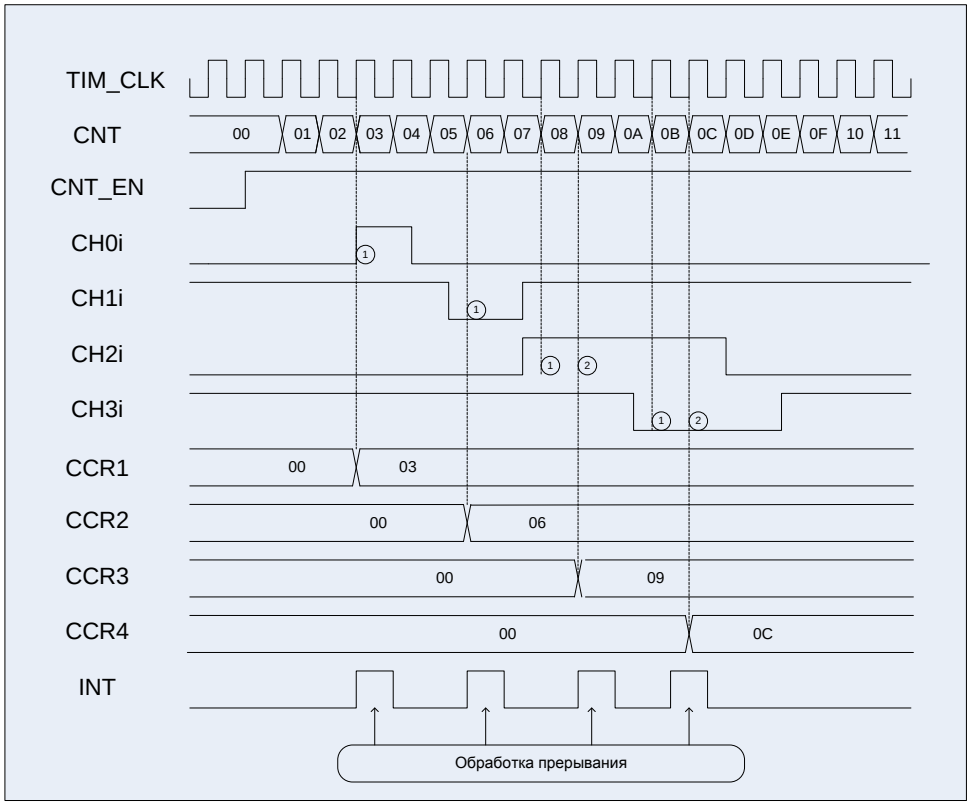


Рисунок 66. Диаграммы примера работы в режиме захвата

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 по всем каналам
//Режим работы каналов - ШИМ
TIMx->TIMx_CHy_CNTRL[0] = 0x00000200;
TIMx->TIMx_CHy_CNTRL[1] = 0x00000200;
TIMx->TIMx_CHy_CNTRL[2] = 0x00000400;
TIMx->TIMx_CHy_CNTRL[3] = 0x00000600;

//Режим работы выхода канала – канал на выход не работает
TIMx->TIMx_CHy_CNTRL1[0]= 0x00000099;
TIMx->TIMx_CHy_CNTRL1[1]= 0x00000099;
TIMx->TIMx_CHy_CNTRL1[2]= 0x00000099;
TIMx->TIMx_CHy_CNTRL1[3]= 0x00000099;

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

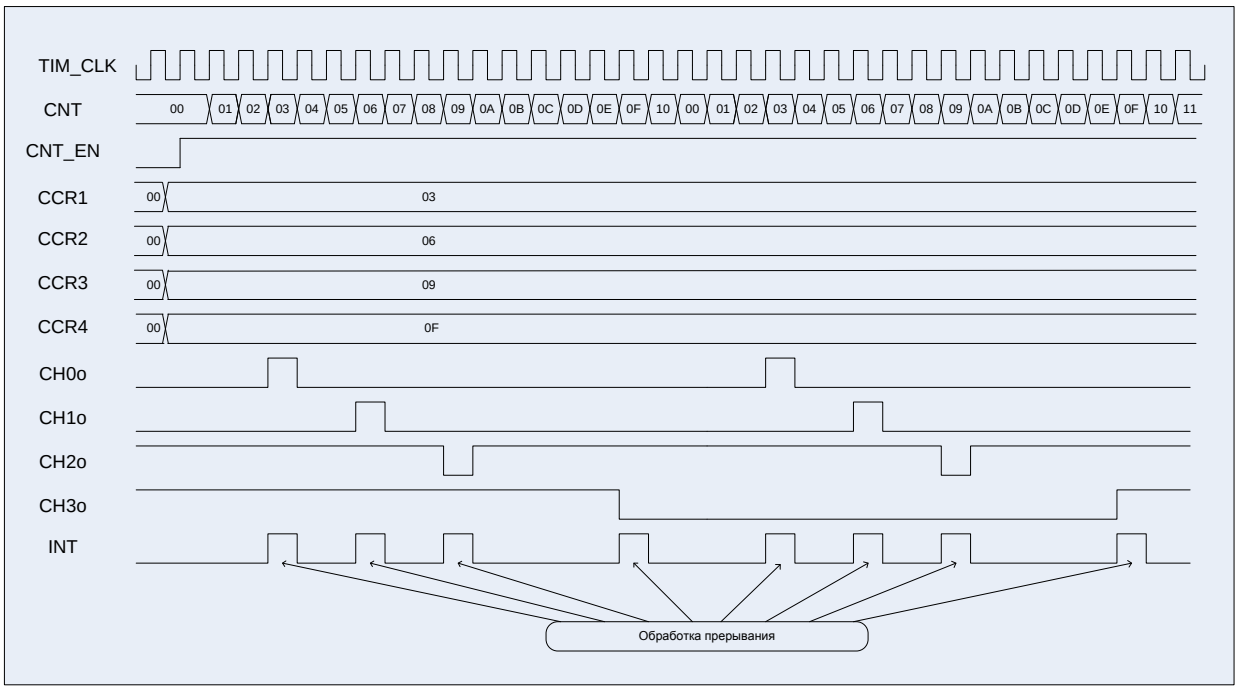


Рисунок 67. Диаграммы примера работы в режиме ШИМ

Описание регистров блока таймера

Таблица 291 Базовые адреса и смещения регистров управления

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	TIMx_CCR2[15:0]	Регистр сравнения, захвата для 2 канала таймера
0x18	TIMx_CCR3[15:0]	Регистр сравнения, захвата для 3 канала таймера
0x1C	TIMx_CCR4[15:0]	Регистр сравнения, захвата для 4 канала таймера
0x20	TIMx_CH1_CNTRL[15:0]	Регистр управления для 1 канала таймера
0x24	TIMx_CH2_CNTRL[15:0]	Регистр управления для 2 канала таймера
0x28	TIMx_CH3_CNTRL[15:0]	Регистр управления для 3 канала таймера
0x2C	TIMx_CH4_CNTRL[15:0]	Регистр управления для 4 канала таймера
0x30	TIMx_CH1_CNTRL1[15:0]	Регистр управления 1 для 1 канала таймера
0x34	TIMx_CH2_CNTRL1[15:0]	Регистр управления 1 для 2 канала таймера
0x38	TIMx_CH3_CNTRL1[15:0]	Регистр управления 1 для 3 канала таймера
0x3C	TIMx_CH4_CNTRL1[15:0]	Регистр управления 1 для 4 канала таймера
0x40	TIMx_CH1_DTG[15:0]	Регистр управления DTG для 1 канала

		таймера
0x44	TIMx_CH2_DTG[15:0]	Регистр управления DTG для 2 канала таймера
0x48	TIMx_CH3_DTG[15:0]	Регистр управления DTG для 3 канала таймера
0x4C	TIMx_CH4_DTG[15:0]	Регистр управления DTG для 4 канала таймера
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	TIMx_DMA_RE[15:0]	Регистр разрешения запросов DMA от прерываний таймера
0x60	TIMx_CH1_CNTRL2[15:0]	Регистр управления 2 для 1 канала таймера
0x64	TIMx_CH2_CNTRL2[15:0]	Регистр управления 2 для 2 канала таймера
0x68	TIMx_CH3_CNTRL2[15:0]	Регистр управления 2 для 3 канала таймера
0x6C	TIMx_CH4_CNTRL2[15:0]	Регистр управления 2 для 4 канала таймера
0x70	TIMx_CCR11[15:0]	Регистр сравнения1, захвата для 1 канала таймера
0x74	TIMx_CCR21[15:0]	Регистр сравнения1, захвата для 2 канала таймера
0x78	TIMx_CCR31[15:0]	Регистр сравнения1, захвата для 3 канала таймера
0x7C	TIMx_CCR41[15:0]	Регистр сравнения1, захвата для 4 канала таймера

TIMx_CNT

Основной счетчик таймера.

Таблица 292 Регистр TIMx_CNT

Номер	31…16	15…0
Доступ	R/W	R/W
Сброс	0	0
		CNT[15:0]

Таблица 293 Описание бит регистра TIMx_CNT

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31..16	-	Зарезервировано
15…0	CNT[7:0]	Значение основного счетчика таймера

TIMx_PSG

Делитель частоты при счете основного счетчика.

Таблица 294 Регистр TIMx_PSG

Номер	31…16	15…0
Доступ	R/W	R/W
Сброс	0	0
		PSG[15:0]

Таблица 295 Описание бит регистра TIMx_PSG

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	-	Зарезервировано
15…0	PSG[7:0]	Значение предварительного делителя счетчика Основной счетчик считает на частоте CLK = TIM_CLK/(PSG+1)

TIMx_ARR

Основание счета основного счетчика.

Таблица 296 Регистр TIMx_ARR

Номер	31…16	15…0
Доступ	R/W	R/W
Сброс	0	0
		ARR[15:0]

Таблица 297 Описание бит регистра TIMx_ARR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	-	Зарезервировано
15…0	ARR[7:0]	Основание счета для основного счетчика CNT = [0…ARR]

TIMx_CNTRL

Регистр управления основного счетчика.

Таблица 298 Регистр TIMx_CNTRL

Номер	31...12	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 299 Описание бит регистра TIMx_CNTRL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...12	-	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 0011 – CNT == ARR в таймере 3 0100 – событие на первом канале 0101 – событие на втором канале 0110 – событие на третьем канале 0111 – событие на четвертом 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 выборки 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 ‘y’ канала таймера.

Таблица 300 Регистр TIMx_CCRy

Номер	31…16	15…0
Доступ	R/W	R/W
Сброс	0	0
		CCR[15:0]

Таблица 301 Описание бит регистра TIMx_CCRy

Разряды	**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 ‘y’ канала таймера.

Таблица 302 Регистр TIMx_CCRy1

Номер	31…16	15…0
Доступ	R/W	R/W

Сброс	0	0
		CCR1[15:0]

Таблица 303 Описание бит регистра TIMx_CCRy1

Разряды	**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_CHy_CNTRL

Регистр управления для ‘y’ канала таймера.

Таблица 304 Регистр TIMx_CHy_CNTRL

Номер	31…16	15	14	13	12
Доступ	U	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0
	-	**CAP nPWM**	**WR CMPL**	**ETREN**	**BRKEN**

Номер	11…9	8	7…6	5…4	3…0
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	000	0	00	00	0000
	OCCM [2:0]	**OCCE**	**CHPSC [1:0]**	**CHSEL [1:0]**	**CHFLTR [3:0]**

Таблица 305 Описание бит регистра TIMx_CHy_CNTRL

Разряды	**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	OCCM[2:0]	Формат выработки сигнала REF в режиме ШИМ. Если CCR1_EN = 0: 000 – всегда 0 001 – 1, если CNT = CCR 010 – 0, если CNT = CCR 011 – переключение REF, если CNT =CCR

		100 – всегда 0 101 – всегда 1 110 – 1, если DIR= 0 (счет вверх), CNT<CCR, иначе 0 0, если DIR= 1 (счет вниз), CNT<CCR, иначе 1 111 – 0, если DIR= 0 (счет вверх), CNT<CCR, иначе 1 1, если DIR= 1 (счет вниз), CNT<CCR, иначе 0. Если CCR1_EN = 1: 000 – всегда 0 001 – 1, если CNT = CCR или CNT = CCR1 010 – 0, если CNT = CCR или CNT = CCR1 011 – переключение REF, если CNT =CCR или CNT =CCR1 100 – всегда 0 101 – всегда 1 110 – 1, если DIR = 0 (счет вверх), CCR1 < CNT < CCR, иначе 0 0, если DIR= 1 (счет вниз), CCR < CNT < CCR1, иначе 1 111 – 0, если DIR = 0 (счет вверх), CCR1 < CNT < CCR, иначе 1 1, если DIR = 1 (счет вниз), CCR< CNT< CCR1, иначе 0
8	OCCE	Разрешение работы ETR: 0 – запрет ETR; 1 – разрешение ETR
7…6	CHPSC[1:0]	Предварительный делитель входного канала: 00 – нет деления 01 – /2 10 – /4 11 – /8
5…4	CHSEL[1:0]	Выбор события по входному каналу: 00 – положительный фронт 01 – отрицательный фронт 10 – положительный фронт от других каналов Для 1 канала от 2 канала Для 2 канала от 3 канала Для 3 канала от 4 канала Для 4 канала от 1 канала 11 – положительный фронт от других каналов Для 1 канала от 3 канала Для 2 канала от 4 канала Для 3 канала от 1 канала Для 4 канала от 2 канала
3…0	CHFLTR[3:0]	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

TIMx_CHy_CNTRL1

Регистр управления 1 для ‘y’ канала таймера.

Таблица 306 Регистр TIMx_CHy_CNTRL1

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 307 Описание бит регистра TIMx_CHy_CNTRL1

Разряды	**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 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 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 ‘y’ канала таймера.

Таблица 308 Регистр TIMx_CHy_CNTRL2

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

Таблица 309 Описание бит регистра TIMx_CHy_CNTRL2

Разряды	**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 10 – отрицательный фронт от других каналов Для 1 канала от 2 канала Для 2 канала от 3 канала Для 3 канала от 4 канала Для 4 канала от 1 канала 11 – отрицательный фронт от других каналов Для 1 канала от 3 канала Для 2 канала от 4 канала Для 3 канала от 1 канала Для 4 канала от 2 канала

TIMx_CHy_DTG

Регистр управления DTG.

Таблица 310 Регистр TIMx_CHy_DTG

Номер	31…16	15…8	7…5	4	3…0
Доступ	U	R/W	U	R/W	R/W
Сброс	0	00000000	000	0	0000
	-	**DTG[7:0]**	-	**EDTS**	**DTGx[3:0]**

Таблица 311 Описание бит регистра TIMx_CHy_DTG

Разряды	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.

Таблица 312 Регистр TIMx_ BRKETR_CNTRL

Номер	31…8	7…4	3…2	1	0
Доступ	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 313 Описание бит регистра TIMx_ BRKETR_CNTRL

Разряды	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 предделитель 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.

Таблица 314 Регистр TIMx_ STATUS

Номер	31…17	16…13	12…9	8…5
Доступ	U	U	R/W	R/W
Сброс		0	0	0
	-	**CCR CAP1 EVENT [3:0]**	**CCR REF EVENT [3:0]**	**CCR CAP EVENT [3:0]**

Номер	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0
	BRK EVENT	**ETR FE EVENT**	**ETR RE EVENT**	**CNT ARR EVENT**	**CNT ZERO EVENT**

Таблица 315 Описание бит регистра TIMx_ STATUS

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...17	-	Зарезервировано
16…13	CCR CAP1 EVENT[3:0]	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. Бит 0 – первый канал Бит 3 – четвертый канал
12…9	CCR REF EVENT[3:0]	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. Бит 0 – первый канал Бит 3 – четвертый канал
8…5	CCR CAP	Событие переднего фронта на входе CAP каналов таймера:

	EVENT[3:0]	0 – нет события; 1 – есть событие. Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события. Бит 0 – первый канал Бит 3 – четвертый канал
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.

Таблица 316 Регистр TIMx_ IE

Номер	31…17	16…13	12…9	8…5
Доступ	U	R/W	R/W	R/W
Сброс		0	0	0
	-	**CCR CAP1 EVENT**	**CCR REF EVENT**	**CCR CAP EVENT**

		IE [3:0]	**IE [3:0]**	**IE [3:0]**

Номер	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0
	BRK EVENT IE	**ETR FE EVENT IE**	**ETR RE EVENT IE**	**CNT ARR EVENT IE**	**CNT ZERO EVENT IE**

Таблица 317 Описание бит регистра TIMx_ IE

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…17	-	Зарезервировано
16…13	CCR CAP1 EVENT IE [3:0]	Флаг разрешения прерывания по событию переднего фронта на выходе CAP1 каналов таймера: 0 – нет прерывания; 1 – прерывание разрешено. Бит 0 – первый канал Бит 3 – четвертый канал
12…9	CCR REF EVENT IE[3:0]	Флаг разрешения прерывания по событию переднего фронта на выходе REF каналов таймера: 0 – нет прерывания; 1 – прерывание разрешено. Бит 0 – первый канал Бит 3 – четвертый канал
8…5	CCR CAP EVENT IE [3:0]	Флаг разрешения прерывания по событию переднего фронта на выходе CAP каналов таймера: 0 – нет прерывания; 1 – прерывание разрешено. Бит 0 – первый канал Бит 3 – четвертый канал
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	Флаг разрешения прерывания по событию совпадения CNT и ARR:

	EVENT IE	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

TIMx_DMA_RE

Регистр разрешения запросов DMA от прерываний таймера.

Таблица 318 Регистр TIMx_ DMA_RE

Номер	31…17	16…13	12…9	8…5
Доступ	U	R/W	R/W	R/W
Сброс		0	0	0
	-	**CCR CAP1 EVENT RE [3:0]**	**CCR REF EVENT RE [3:0]**	**CCR CAP EVENT RE [3:0]**

Номер	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0
	BRK EVENT RE	**ETR FE EVENT RE**	**ETR RE EVENT RE**	**CNT ARR EVENT RE**	**CNT ZERO EVENT RE**

Таблица 319 Описание бит регистра TIMx_ DMA_RE

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...17	-	Зарезервировано
16...13	CCR CAP1 EVENT RE [3:0]	Флаг разрешения запроса DMA по событию переднего фронта на выходе CAP1 каналов таймера: 0 – нет запроса DMA; 1 – запрос DMA разрешен. Бит 0 – первый канал Бит 3 – четвертый канал
12…9	CCR REF EVENT RE[3:0]	Флаг разрешения запроса DMA по событию переднего фронта на выходе REF каналов таймера: 0 – нет запроса DMA; 1 – запрос DMA разрешен. Бит 0 – первый канал Бит 3 – четвертый канал
8…5	CCR CAP EVENT RE [3:0]	Флаг разрешения запроса DMA по событию переднего фронта на выходе CAP каналов таймера: 0 – нет запроса DMA; 1 – запрос DMA разрешен.

		Бит 0 – первый канал Бит 3 – четвертый канал
4	BRK EVENT RE	Флаг разрешения по триггерированному по PCLK состоянию входа BRK: 0 – нет запроса DMA; 1 – запрос DMA разрешен
3	ETR FE EVENT RE	Флаг разрешения запроса DMA по заднему фронту на входе ETR: 0 – нет запроса DMA; 1 – запрос DMA разрешен
2	ETR RE EVENT RE	Флаг разрешения запроса DMA по переднему фронту на входе ETR: 0 – нет запроса DMA; 1 – запрос DMA разрешен
1	CNT ARR EVENT RE	Флаг разрешения запроса DMA по событию совпадения CNT и ARR: 0 – нет запроса DMA; 1 – запрос DMA разрешен
0	CNT ZERO EVENT RE	Флаг разрешения запроса DMA по событию совпадения CNT и нуля: 0 – нет запроса DMA; 1 – запрос DMA разрешен

Контроллер АЦП

В микроконтроллере реализован 12-ти разрядный АЦП. С помощью 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 для каждого АЦП.
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 бит Cfg_REG_DIVCLK[3:0]. Максимальная частота CLK не может превышать 14 МГЦ.

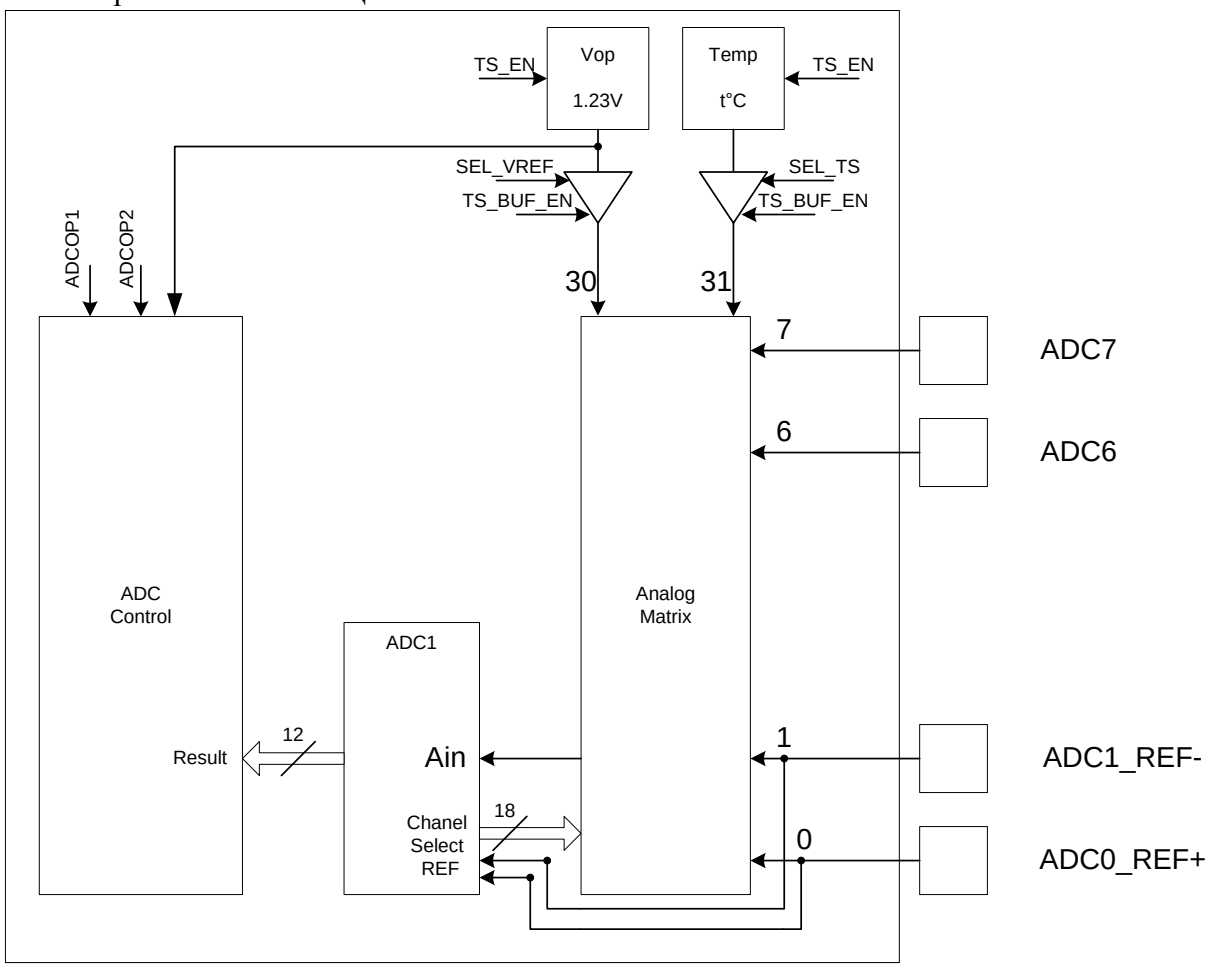


Рисунок 68.

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 в 1. После включения можно использовать источник опорного напряжения для АЦП вместо его собственного. Для этого необходимо установить биты ADCx_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 ADCx_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 случая 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 ADCx_STATUS. А в регистре ADCx_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 ADCx_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 ADCx_STATUS.

Последовательное преобразование нескольких каналов

Для автоматического последовательного преобразования нескольких каналов или одного канала в регистре ADCx_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 случая 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 бит 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 ADCx_STATUS. А в регистре ADCx_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 ADCx_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 ADCx_STATUS.

Для последовательного преобразования одного и того же канала можно в регистре ADCx_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 значение в допустимых пределах можно задать нижнюю и верхнюю допустимые границы в регистрах ADCx_L_LEVEL и ADCx_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 ADCx_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 TS_BUF_EN 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 TS_BUF_EN 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 ADCx_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:

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

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 ±1/4 LSB, то N = 14. Если необходимо обеспечить преобразование с точностью 10 разрядов ±1 LSB, то N=10. Время заряда T_S = определяется битами DelayGo[2:0] и схемой самого АЦП и представлена в Таблица 320.

Время заряда внутренней емкости АЦП и время преобразования

Таблица 320 Время заряда внутренней емкости АЦП и время преобразования

DelayGo[2:0]	**Дополнительная задержка перед началом преобразования**	**Общее время T_S заряда емкости АЦП перед началом преобразования**	**Общее время преобразования АЦП**
000	1	5	29
001	2	6	30
010	3	7	31
011	4	8	32
100	5	9	33
101	6	10	34
110	7	11	35
111	8	12	36

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 ADCx_CFG необходимо сделать до задания бита Go, иначе новая конфигурация будет действовать со следующего преобразования.

Описание регистров блока контроллера АЦП

Таблица 321 Описание регистров блока контроллера АЦП

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	ADC_TRIM	Регистр настройки термодатчика

ADCx_CFG

Таблица 322 Регистр ADCx_CFG

Номер	31…28	27…25	24…21	20	19
Доступ	R/W	U	R/W	R/W	R/W
Сброс	0	0	0	0	0
	Delay ADC [3:0]	**Delay Go [2:0]**	**TR[3:0]**	**SEL VREF**	**SEL TS**

Номер	18	17	16	15…12	11	10
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	TS_BUF EN / ADC2 OP	**TS_EN / ADC1 OP**	**Cfg Sync Conver**	**Cfg REG DIVCLK [3:0]**	**Cfg M_REF**	**Cfg REG RNGC**

Номер	9	8…4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	Cfg REG CHCH	**Cfg REG CHS [4:0]**	**Cfg REG SAMPLE**	**Cfg REG CLKS**	**Cfg REG GO**	**Cfg REG ADON**

Таблица 323 Описание бит регистра ADCx_CFG

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	Delay ADC [3:0]	Задержка между началом преобразования ADC1 и ADC2 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	TR[3:0]	Подстройка опорного напряжения Смотри диаграмму на **Ошибка! Источник ссылки не найден.**
20	SEL VREF	Выбор для оцифровки источника опорного напряжения на 1,23 В: 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.** Включения выходного усилителя для датчика температуры и источника опорного напряжения: 0 – выключен; 1 – включен. Используется при TS_EN = 1. Для уменьшения тока потребления
17	TS EN	**В регистре ADC1_CFG.** Включения датчика температуры и источника опорного напряжения: 0 – выключен; 1 – включен. При включении датчика температуры и источника опорного напряжения выходной сигнал стабилизируется в течение времени Tstb
17	ADC1 OP	**В регистре ADC2_CFG.** Выбор источника опорного напряжения 1,23 В: 0 – внутренний (не точный); 1 – от датчика температуры (точный)

16	Cfg Sync Conver	Всегда записывать ноль
15...12	Cfg REG DIVCLK [3:0]	Выбор коэффициента деления входной частоты 0000 – CLK 000 – 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 – последовательный: автоматический запуск после завершения предыдущего преобразования; 0 – одиночный
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 REG ADON	Включение АЦП: 1 – включено; 0 – выключено

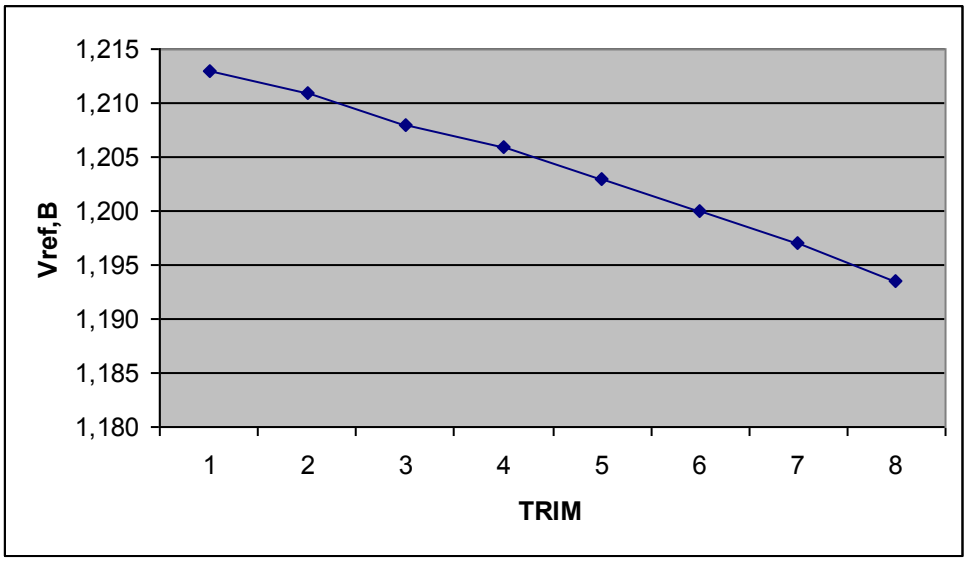


Рисунок 69. Зависимость источника опорного напряжения от подстройки

ADCx_H_LEVEL

Таблица 324 Регистр ADCx_ H_LEVEL

Номер	31…12	11…0
Доступ	U	R/W
Сброс	0	0
	-	**REG H LEVEL [11:0]**

Таблица 325 Описание бит регистра ADCx_ H_LEVEL

Разряды	**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.

ADCx_L_LEVEL

Таблица 326 Регистр ADCx_ L_LEVEL

Номер	31…12	11…0
Доступ	U	R/W
Сброс	0	0
		REG L LEVEL [11:0]

Таблица 327 Описание бит регистра ADCx_ L_LEVEL

Разряды	**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.

ADCx_RESULT

Таблица 328 Регистр ADCx_ RESULT

Номер	31…21	20…16	15…12	11…0
Доступ	U	RO	U	RO
Сброс	0	0	0	0
	-	**CHANNEL [11:0]**	-	**RESULT [11:0]**

Таблица 329 Описание бит регистра ADCx_ RESULT

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...21	-	Зарезервировано
20...16	CHANNEL [11:0]	Канал результата преобразования
15...12	-	Зарезервировано
11…0	RESULT [11:0]	Значение результата преобразования

ADCx_STATUS

Таблица 330 Регистр ADCx_ STATUS

Номер	31…5	4	3	2	1	0
Доступ	U	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	-	**ECOIF_IE**	**AWOIF_IE**	**Flg REG EOCIF**	**Flg REG AWOIFEN**	**Flg REG OVERWRITE**

Таблица 331 Описание бит регистра ADCx_ STATUS

Разряды	**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 – прерывание не генерируется; 1 – прерывание генерируется
3	AWOIF_IE	Флаг разрешения генерирования прерывания по событию Flg_REG_AWOIFEN: 0 – прерывание не генерируется; 1 – прерывание генерируется
2	Flg REG EOCIF	Флаг выставляется, когда закончено преобразование 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 верней 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

ADCx_CHSEL

Таблица 332 Регистр ADCx_ CHSEL

Номер	31…0
Доступ	R/W
Сброс	0
	Sl_Ch_Ch_REF[31:0]

Таблица 333 Описание бит регистра ADCx_ CHSEL

Разряды	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...0	Sl_Ch_Ch_REF[31:0]	Выбор каналов автоматического перебора: 0 – в соответствующем бите канал не участвует в переборе; 1 – канал участвует в переборе

ADCx_TRIM

Таблица 334 Регистр ADCx_ 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 335 Описание бит регистра ADCx_ TRIM

Разряды	**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 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 PrimeCell 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-разрядный 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.
- Содержит независимые буферы приема и передачи (8 ячеек, 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.
- Поддержка прямого доступа к памяти (ПДП).

Структурная схема модуля представлена ниже.

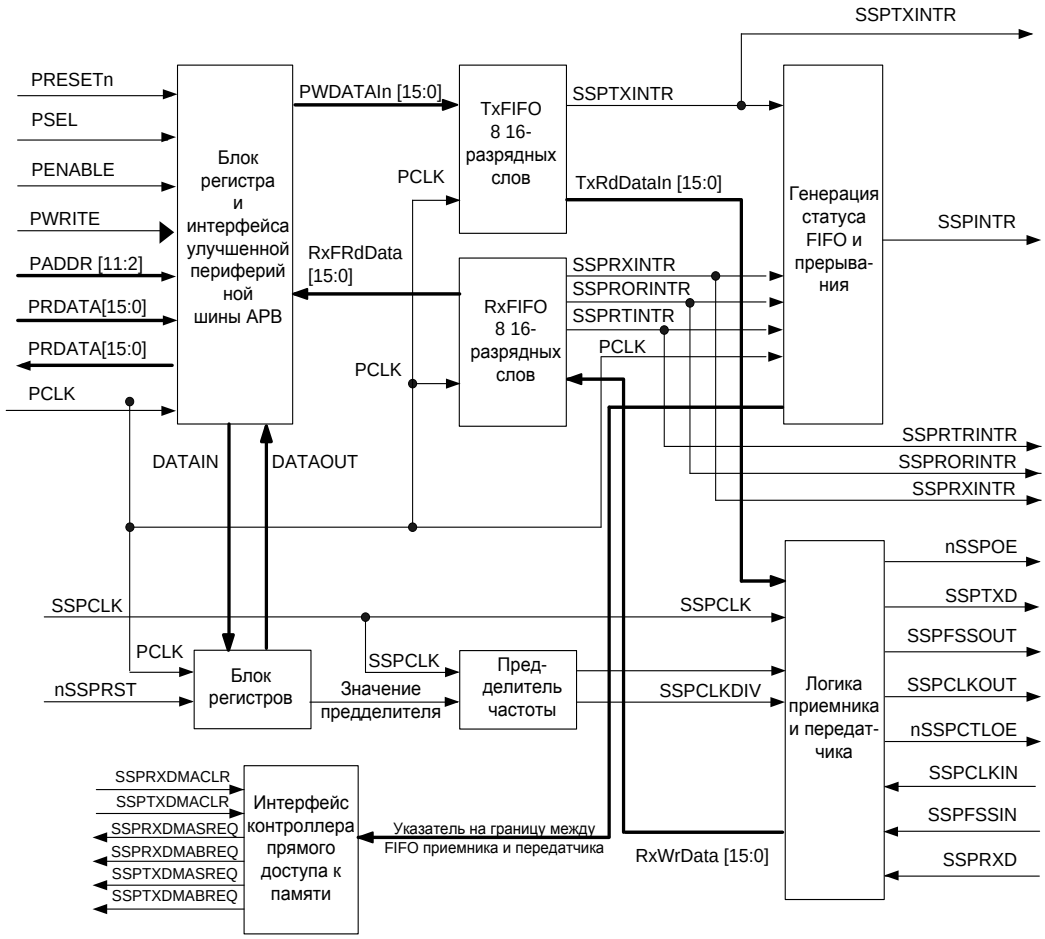


Рисунок 70. Структурная схема модуля SSP

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.

Описание функционирования

Глава содержит описание основных функциональных блоков синхронного последовательного интерфейса SSP и содержит следующие разделы:

- Общий обзор модуля.
- Функциональное описание модуля.
- Описание работы модуля.

Общий обзор модуля SSP

Модуль PrimeCell 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 в 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 и принимаются с линии SSP_RXD.

Модуль SSP содержит программируемые делители частоты, формирующие тактовый сигнал обмена данными SSPCLKOUT из сигнала, поступающего на линию SSP_CLK. Скорость передачи данных может достигать более 2 МГц, в зависимости от частоты SSP_CLK и характеристик подключенного периферийного устройства.

Режим обмена данными, формат информационного кадра и количество бит данных задаются программно с помощью регистров управления SSPCR0 и SSPCR1.

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.

Модуль также формирует сигналы запроса на прямой доступ к памяти (ПДП) для совместной работы с контроллером ПДП.

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 SSPCLKOUT 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, исключается возможность деления частоты на нечетный коэффициент, что, в свою очередь гарантирует формирование тактового сигнала симметричной формы (с одинаковой длительностью полупериодов высокого и низкого уровня).

Сформированный описанным образом сигнал далее поступает на второй делитель частоты, с выход которого и снимается тактовый сигнал обмена данными SSPCLKOUT.

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 «первый вошел, первый вышел». Данные от центрального процессора, записанные через шину AMBA APB, 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 «первый вошел, первый вышел». Принятые от периферийного устройства данные сохраняются блоком приема данных в нем до тех пор, пока не будут считаны центральным процессором через шину AMBA APB.

Блок приема и передачи данных

В режиме ведущего устройства модуль формирует тактовый сигнал обмена данными SSPCLKOUT для подключенных ведомых устройств. Как было описано ранее, данный сигнал формируется путем деления частоты сигнала SSPCLK.

Блок передатчика последовательно считывает значения из буфера FIFO передатчика и производит их преобразование из параллельной в последовательную форму. Далее поток последовательных данных и элементов кадровой синхронизации, тактированных сигналом SSPCLKOUT, передается 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 через интерфейс шины APB.

В режиме ведомого устройства тактовый сигнал обмена данными формируется одним из подключенных к модулю периферийных устройств и поступает по линии SSPCLKIN. 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 SSPCLKIN, из последовательной в параллельную форму, после чего загружает их в буфер FIFO приемника, откуда они могут быть считаны через интерфейс шины APB.

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 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.

Модуль SSP позволяет использовать оба описанных выше похода.

Предусмотрены независимые линии запроса прерывания по готовности приемника и передатчика SSPTXINTR и SSPRXINTR, что позволяет обслуживание устройства (чтение или запись данных) по достижению заданного уровня заполнения буферов FIFO приемника или передатчика.

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 SSPCR0 и SSPCR1.

Кроме того, для установки требуемой скорости передачи данных необходимо выбрать параметры блока формирования тактового сигнала с учетом значения частоты внешнего сигнала SSPCLK и записать соответствующую информацию в регистр SSPCPSR.

Разрешение работы приемопередатчика

Разрешение осуществляется путем установки бита SSE регистра управления SSPCR1. Буфер FIFO передатчика может быть либо проинициализирован путем записи в него до восьми 16-разрядных слов заблаговременно перед установкой этого бита, либо 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 PCLK и SSP_CLK. Частота SSP_CLK должна меньше или равна частоте PCLK. Выполнение этого требования гарантирует синхронизацию сигналов управления, передаваемых из зоны действия тактового сигнала SSP_CLK в зону действия сигнала PCLK в течение времени, меньшего продолжительности передачи одного информационного кадра:

FSSPCLK <= FPCLK.

В режиме ведомого устройства сигнал SSPCLKIN от ведущего внешнего устройства поступает на схемы синхронизации, задержки и обнаружения фронта. Для того чтобы обнаружить фронт сигнала SSPCLKIN необходимо три такта сигнала SSPCLK. Сигнал SSP_TXD имеет меньшее время установки по отношению к заднему фронту SSPCLKIN, по которому и происходит считывание данных из линии. Время установки и удержания сигнала SSP_RXD по отношению к сигналу SSPCLKIN должно выбираться с запасом, гарантирующим правильное считывание данных. Для обеспечения корректной работы устройства необходимо, чтобы частота SSP_CLK была как минимум в 12 раз больше, чем максимальная предполагаемая частота сигнала SSPCLKIN.

Выбор частоты тактового сигнала SSP_CLK должен обеспечивать поддержку требуемого диапазона скоростей обмена данными. Отношение минимальной частоты сигнала SSP_CLK к максимальной частоте сигнала SSPCLKOUT 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 SSP_CLK должна составлять не менее 3,6864 МГц. В этом случае в регистр SSPCPSR должно быть записано значение 2, а поле SCR[7:0] регистра SSPCR0 должно быть установлено в 0.

В режиме ведомого устройства для обеспечения той же информационной скорости необходимо использовать тактовый сигнал SSP_CLK с частотой не менее 22,12 МГц. При этом в регистр SSPCPSR должно быть записано значение 12, а поле SCR[7:0] регистра SSPCR0 должно быть установлено в 0.

Соотношение между максимальной частотой сигнала SSP_CLK и минимальной частотой SSPCLKOUT составляет 254 * 256.

Минимальная допустимая частота сигнала SSP_CLK определяется следующей системой соотношений, которые должны выполняться одновременно:

FSSPCLK(min) => 2 x FSSPCLKOUT(max) [for master mode]

FSSPCLK(min) => 12 x FSSPCLKIN(max) [for slave mode].

Аналогично, максимальная допустимая частота сигнала SSP_CLK определяется следующей системой соотношений, которые должны выполняться одновременно:

FSSPCLK(max) <= 254 x 256 x FSSPCLKOUT(min) [for master mode]

FSSPCLK(max) <= 254 x 256 x FSSPCLKIN(min) [for slave mode].

Программирование регистра управления SSPCR0

Регистр SSPCR0 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 SSP_CLK 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, кроме того, задается 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 SSPTXD output disable – запрет линии SSP_TXD для ведомого режима) регистра SSPCR1. 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 SSP_CLK. На первом этапе формирования частота этого сигнал делится на четный коэффициент CPSDVSR, лежащий в диапазоне от 2 до 254, доступный для программирования через регистр SSPCPSR. Сформированный сигнал далее поступает на делитель частоты с коэффициентом (1 + SCR) от 1 до 256, где значение SCR доступно для программирования через SSPCR0.

Частота выходного тактового сигнала обмена данными SSPCLKOUT определяется следующим соотношением:

FSSPCLKOUT = FSSPCLK / (CPSDVR x (1+SCR))

Например, в случае если частота сигнала SSP_CLK составляет 3,6864 МГц, а значение CPSDVSR = 2, частота сигнала SSPCLKOUT 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 SSPCLKOUT формируется только тогда, когда приемопередатчик готов к обмену данными. Перевод сигнала SSPCLKOUT 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 SSPFSSOUT 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 SSPFSSOUT формируется импульс с длительностью, равной одному тактовому интервалу обмена данными. В этом режиме приемопередатчик PrimeCell SSP, равно как ведомые периферийные устройства, передает данные в линию по переднему фронту сигнала SSPCLKOUT, 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

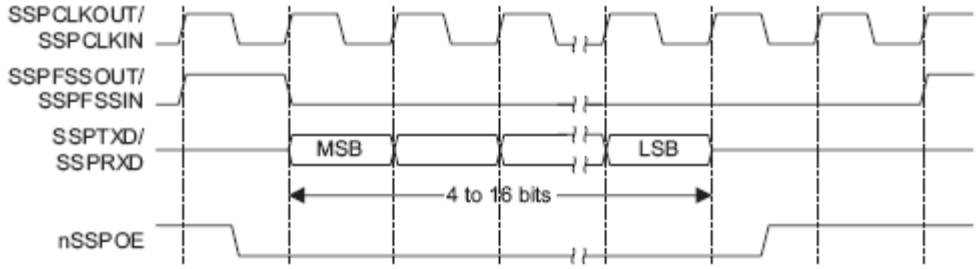


Рисунок 71. Формат синхронного обмена протокола SSI фирмы Texas Instruments (единичный обмен)

В данном режиме при неактивном приемопередатчике PrimeCell SSP сигналы SSPCLKOUT и SSPFSSOUT 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 SSPCLKOUT. Значение из буфера FIFO при этом переносится в сдвиговый регистр блока передатчика. По следующему переднему фронту сигнала SSPCLKOUT старший значащий разряд информационного кадра (4 – 16 бит данных) выдается на выход линии SSPTXD и т.д.

В режиме приема данных как модуль PrimeCell SSP, так и ведомое внешнее устройство последовательно загружают биты данных в сдвиговый регистр по заднему фронту сигнала SSPCLKOUT. Принятые данные переносятся из сдвигового регистра в буфер FIFO после загрузки в него младшего значащего бита данных по очередному переднему фронту сигнала SSPCLKOUT.

Временные диаграммы последовательного синхронного обмена по протоколу SSI фирмы Texas Instruments представлены на рисунке ниже.

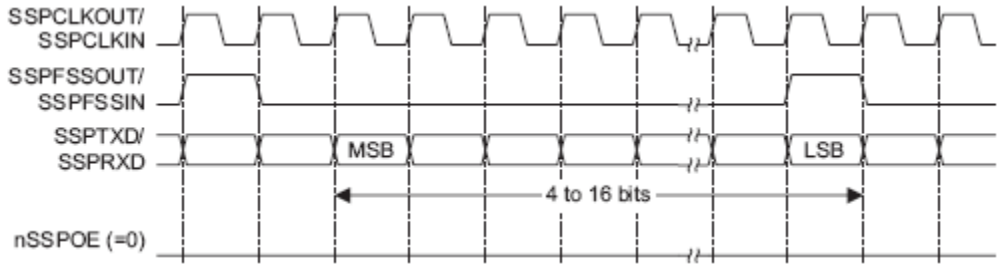


Рисунок 72. Формат синхронного обмена протокола SSI фирмы Texas Instruments (непрерывный обмен)

Формат синхронного обмена SPI фирмы Motorola

Интерфейс SPI фирмы Motorola осуществляется по четырем сигнальным линиям, при этом сигнал SSPFSSOUT выполняет функцию выбора ведомого устройства. Главной особенностью протокола SPI является возможность выбора состояния и фазы сигнала SSPCLKOUT 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 SSPCLKOUT переводится в низкий логический уровень. В противном случае при отсутствии обмена данными линия SSPCLKOUT 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 рисунках ниже.

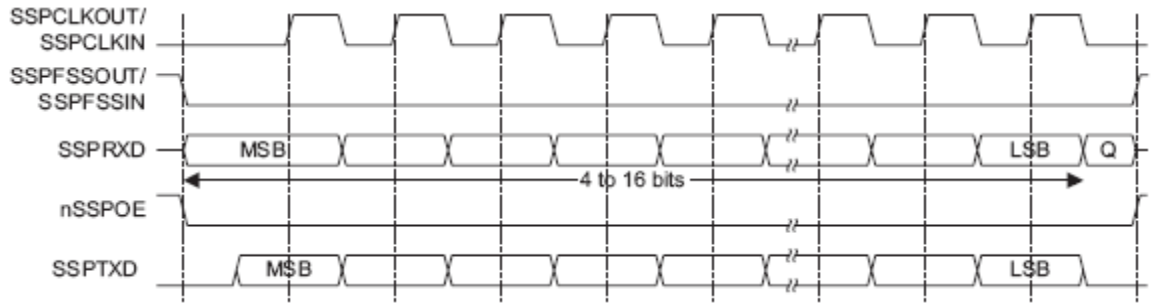


Рисунок 73. Формат синхронного обмена протокола SPI фирмы Motorola, SPO=0,SPH=0 (одиночный обмен)

Примечание: На рисунке буквой Q обозначен сигнал с неопределенным уровнем.

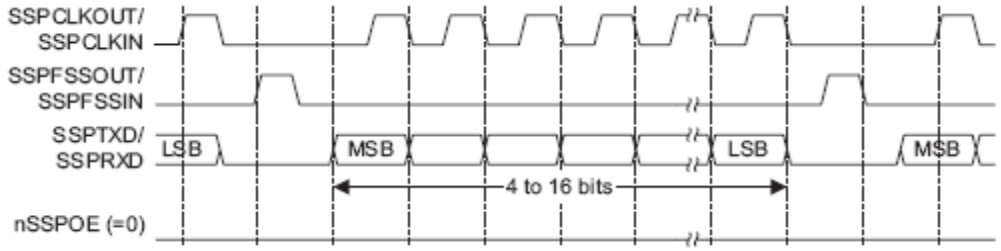


Рисунок 74. Формат синхронного обмена протокола SPI фирмы Motorola, SPO=0, SPH=0 (непрерывный обмен)

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLKOUT имеет низкий логический уровень;
- сигнал SSPFSSOUT имеет высокий логический уровень;
- сигнал SSPTXD переводится в низкий логический уровень;
- сигнал nSSPOE переводится в высокий уровень, переводя таким образом выходной контакт SSPTXD передатчика в высокоимпедансное состояние;
- если модуль сконфигурирован как ведущее устройство, линия nSSPCTLOE переводится в низкий уровень, разрешая передачу сигнала на выходной контакт SSPCLKOUT;
- если модуль сконфигурирован как ведомое устройство, линия nSSPCTLOE переводится в высокий уровень, отключая выходной контакт SSPCLKOUT.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSPFSSOUT переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSPRXD ведущего. Сигнал nSSPOE переводится в низкий уровень, переводя выходной контакт передатчика SSPTXD из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSPCLKOUT, на линии SSPTXD 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 SSPCLKOUT переводится в высокий логический уровень.

Далее данные регистрируются по переднему фронту и выдаются в линию по заднему фронту сигнала SSPCLKOUT.

В случае передачи одного слова данных, после приема его последнего бита линия SSPFSSOUT переводится в высокий логический уровень по истечении одного периода тактового сигнала SSPCLKOUT.

В режиме непрерывной передачи данных, на линии SSPFSSOUT 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 SSPFSSOUT в высокий уровень по окончании передачи каждого кадра, разрешая таким образом запись новых данных. По окончании приема последнего бита блока данных линия SSPFSSOUT переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSPCLKOUT.

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 рисунке ниже (одиночный и непрерывный обмен).

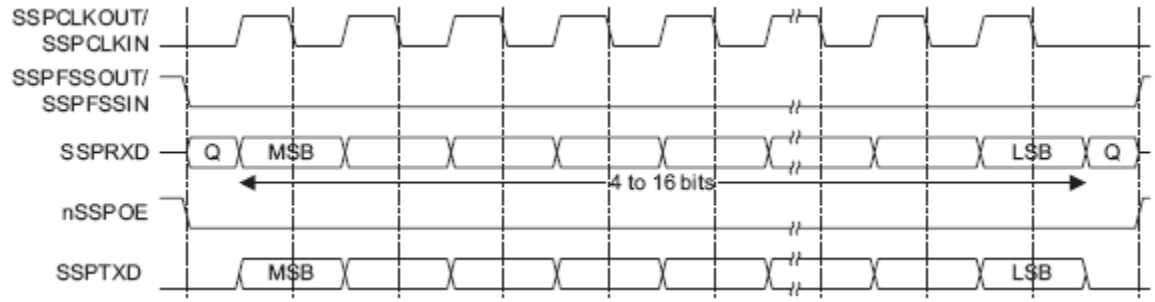


Рисунок 75. Формат синхронного обмена протокола SPI фирмы Motorola, SPO=0,SPH=1

Примечание. На рисунке буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLKOUT имеет низкий логический уровень;
- сигнал SSPFSSOUT имеет высокий логический уровень;
- сигнал SSPTXD переводится в низкий логический уровень;
- сигнал nSSPOE переводится в высокий уровень, переводя таким образом выходной контакт SSPTXD передатчика в высокоимпедансное состояние;
- если модуль сконфигурирован как ведущее устройство, линия nSSPCTLOE переводится в низкий уровень, разрешая передачу сигнала на выходной контакт SSPCLKOUT;
- если модуль сконфигурирован как ведомое устройство, линия nSSPCTLOE переводится в высокий уровень, отключая выходной контакт SSPCLKOUT.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSPFSSOUT переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSPRXD ведущего. Сигнал nSSPOE переводится в низкий уровень, переводя выходной контакт передатчика SSPTXD из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSPCLKOUT на линиях обмена как ведущего, так и ведомого устройств сформированы значения первых битов передаваемых данных. В это же время включается линия SSPCLKOUT и на ней формируется передний фронт сигнала.

Далее данные регистрируются по заднему фронту и выдаются в линию по переднему фронту сигнала SSPCLKOUT.

В случае передачи одного слова данных, после приема его последнего бита линия SSPFSSOUT переводится в высокий логический уровень по истечении одного периода тактового сигнала SSPCLKOUT.

В режиме непрерывной передачи данных, линия SSPFSSOUT 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 рисунках ниже.

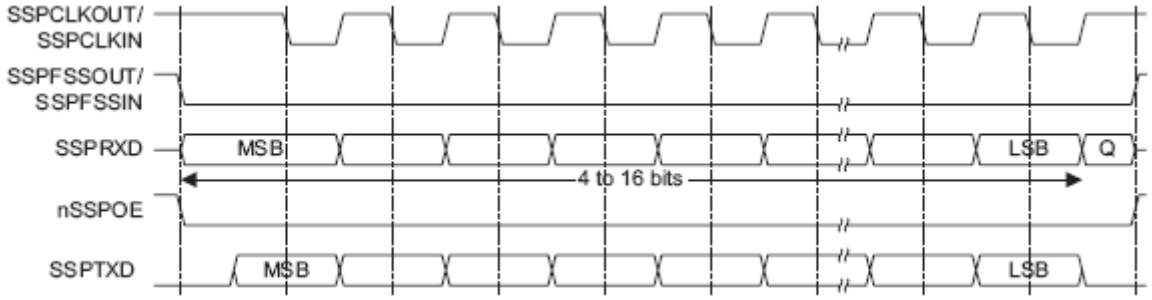


Рисунок 76. Формат синхронного обмена протокола SPI фирмы Motorola, SPO=1, SPH=0 (одиночный обмен)

Примечание. На рисунке буквой Q обозначен сигнал с неопределенным уровнем.

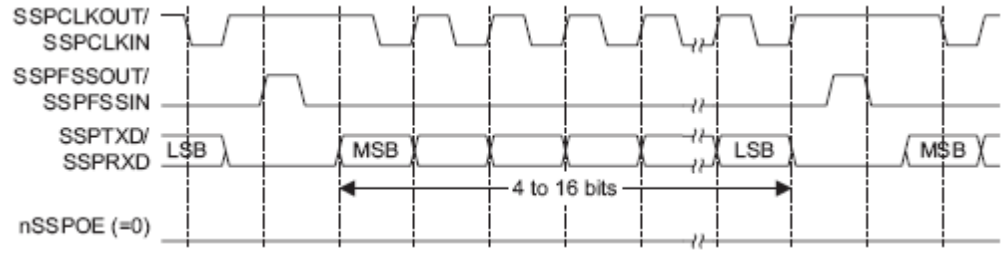


Рисунок 77. Формат синхронного обмена протокола SPI фирмы Motorola, SPO=1, SPH=0 (непрерывный обмен)

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLKOUT имеет высокий логический уровень;
- сигнал SSPFSSOUT имеет высокий логический уровень;
- сигнал SSPTXD переводится в низкий логический уровень;
- сигнал nSSPOE переводится в высокий уровень, переводя таким образом выходной контакт SSPTXD передатчика в высокоимпедансное состояние;
- если модуль сконфигурирован как ведущее устройство, линия nSSPCTLOE переводится в низкий уровень, разрешая передачу сигнала на выходной контакт SSPCLKOUT;

- если модуль сконфигурирован как ведомое устройство, линия nSSPCTLOE переводится в высокий уровень, отключая выходной контакт SSPCLKOUT.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSPFSSOUT переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSPRXD ведущего. Сигнал nSSPOE переводится в низкий уровень, переводя выходной контакт передатчика SSPTXD из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSPCLKOUT, на линии SSPTXD 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 SSPCLKOUT переводится в низкий логический уровень.

Далее данные регистрируются по заднему фронту и выдаются в линию по переднему фронту сигнала SSPCLKOUT.

В случае передачи одного слова данных, после приема его последнего бита линия SSPFSSOUT переводится в высокий логический уровень по истечении одного периода тактового сигнала SSPCLKOUT.

В режиме непрерывной передачи данных, на линии SSPFSSOUT 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 SSPFSSOUT в высокий уровень по окончании передачи каждого кадра, разрешая таким образом запись новых данных. По окончании приема последнего бита блока данных линия SSPFSSOUT переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSPCLKOUT.

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 рисунке ниже (одиночный и непрерывный обмен).

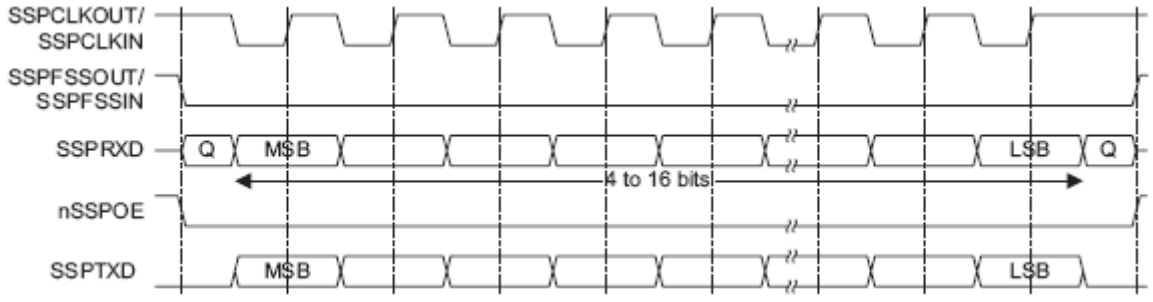


Рисунок 78. Формат синхронного обмена протокола SPI фирмы Motorola, SPO=1, SPH=1

Примечание. На рисунке буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLKOUT имеет высокий логический уровень;
- сигнал SSPFSSOUT имеет высокий логический уровень;
- сигнал SSPTXD переводится в низкий логический уровень;
- сигнал nSSPOE переводится в высокий уровень, переводя таким образом выходной контакт SSPTXD передатчика в высокоимпедансное состояние;

- если модуль сконфигурирован как ведущее устройство, линия nSSPCTLOE переводится в низкий уровень, разрешая передачу сигнала на выходной контакт SSPCLKOUT;
- если модуль сконфигурирован как ведомое устройство, линия nSSPCTLOE переводится в высокий уровень, отключая выходной контакт SSPCLKOUT.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSPFSSOUT переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSPRXD ведущего. Сигнал nSSPOE переводится в низкий уровень, переводя выходной контакт передатчика SSPTXD из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSPCLKOUT на линиях обмена как ведущего, так и ведомого устройств сформированы значения первых битов передаваемых данных. В это же время включается линия SSPCLKOUT и на ней формируется передний фронт сигнала.

Далее данные регистрируются по переднему фронту и выдаются в линию по заднему фронту сигнала SSPCLKOUT.

В случае передачи одного слова данных, после приема его последнего бита линия SSPFSSOUT переводится в высокий логический уровень по истечении одного периода тактового сигнала SSPCLKOUT.

В режиме непрерывной передачи данных, линия SSPFSSOUT 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 рисунках ниже.

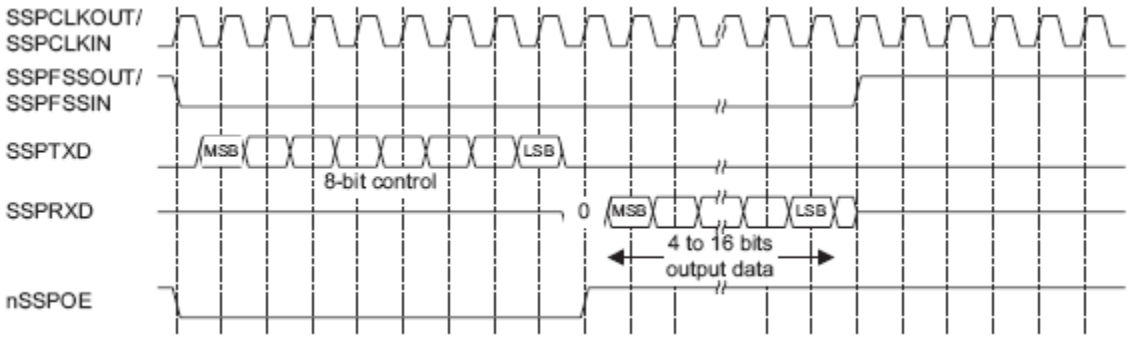


Рисунок 79. Формат синхронного обмена протокола Microwire фирмы National Semiconductor (одиночный обмен)

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 восьмибитной 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 SSPCLKOUT имеет низкий логический уровень;
- сигнал SSPFSSOUT имеет высокий логический уровень;
- сигнал SSPTXD переводится в низкий логический уровень;
- сигнал nSSPOE переводится в высокий уровень, переводя, таким образом, выходной контакт SSPTXD передатчика в высокоимпедансное состояние.

Переход в режим информационного обмена происходит после записи управляющего байта в буфер FIFO передатчика. По заднему фронту сигнала SSPFSSOUT данные из буфера переносятся в регистр сдвига блока передатчика, откуда, начиная со старшего значащего разряда, последовательно выдаются в линию SSPTXD. Линия SSPFSSOUT остается в низком логическом уровне в течение всей передачи кадра. Линия SSPRXD при этом находится в высокоимпедансном состоянии.

Внешнее ведомое устройство осуществляет прием бит данных по переднему фронту сигнала SSPCLKOUT. По окончании приема последнего бита управляющей последовательности она декодируется в течение одного тактового интервала, после чего ведомое устройство передает запрошенные данные в адрес модуля PrimeCell SSP. Биты данных выдаются в линию SSPRXD по заднему фронту сигнала SSPCLKOUT. 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 SSPFSSOUT переводится в высокий уровень на время, соответствующее одному тактовому интервалу, что служит командой для переноса принятого слова данных их регистра сдвига в буфер FIFO приемника.

Примечание. Внешнее устройство может перевести линию приемника в третье состояние по заднему фронту сигнала SSPCLKOUT после приема последнего бита слова данных, либо после перевода линии SSPFSSOUT в высокий логический уровень.

Непрерывный обмен данными начинается и заканчивается так же, как и в одиночный обмен. Однако линия SSPFSSOUT 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 SSPCLKOUT.

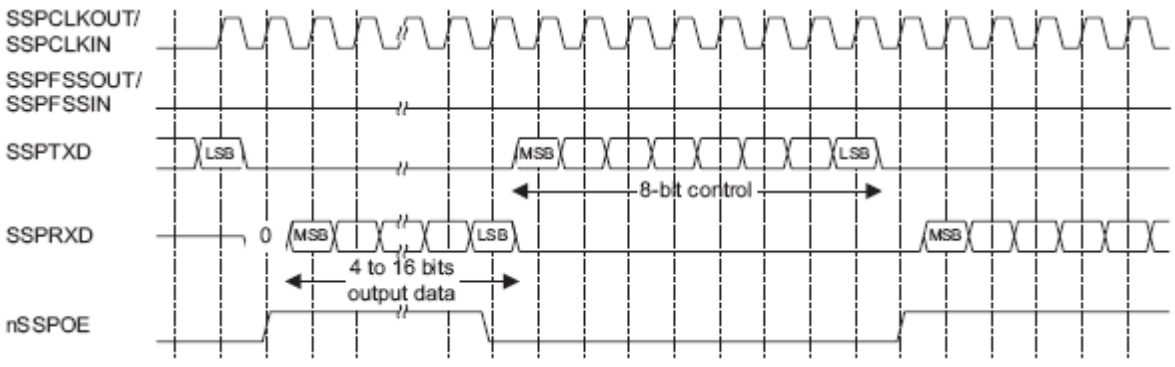


Рисунок 80. Формат синхронного обмена протокола Microwire фирмы National Semiconductor (непрерывный обмен)

Требования к временным параметрам сигнала SSPFSSIN относительно тактового сигнала SSPCLKIN в режиме Microwire

Модуль SSP, работающий в режиме Microwire как ведомое устройство, регистрирует данные по переднему фронту сигнала SSPCLKIN после установки сигнала SSPFSSIN в низкий логический уровень. Ведущие устройства, формирующие сигнал SSPCKLIN должны гарантировать достаточное время установки и удержания сигнала SSPFSSIN по отношению к переднему фронту сигнала SSPCLKIN.

Иллюстрация данных требований представлена на рисунке ниже. По отношению к переднему фронту сигнала SSPCLKIN, по которому осуществляется регистрация данных в приемнике ведомого модуля PrimeCell SSP, время установки сигнала SSPFSSIN должно быть, как минимум, в два раза больше периода SSPCLKIN, на котором работает модуль. По отношению к предыдущему переднему фронту сигнала SSPCLKIN должно обеспечиваться время удержания не менее одного периода этого тактового сигнала.

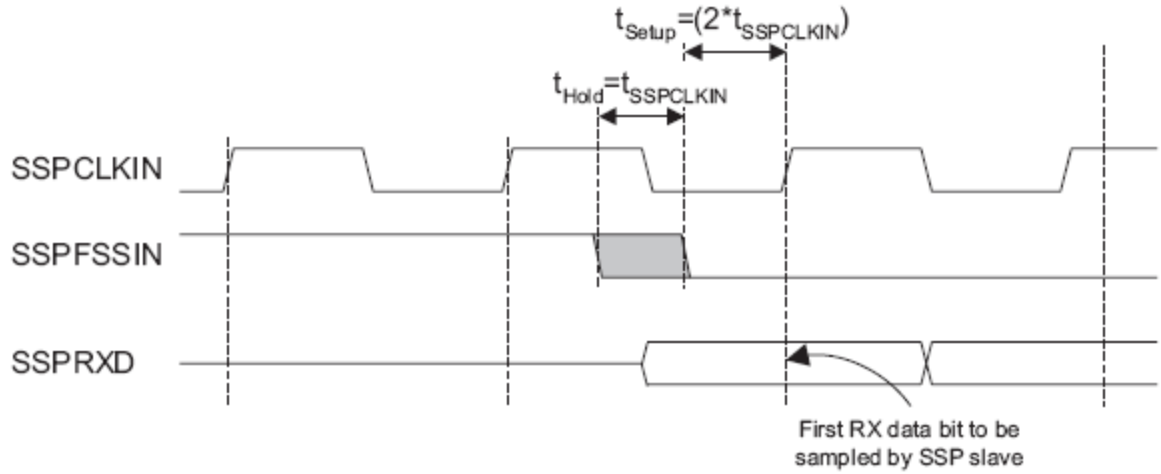


Рисунок 81. Формат кадра Microwire, требования к времени установки и удержания сигнала SSPFSSIN

Примеры конфигурации модуля в ведущем и ведомом режимах

На рисунках ниже показаны варианты подключения модуля PrimeCell SSP (PL022) к периферийным устройствам, работающим в ведущем или ведомом режиме.

Примечание. Модуль SSP (PL022) не поддерживает динамическое изменение режима ведущий – ведомый. Каждый приемопередатчик должен быть изначально сконфигурирован в одном из этих режимов.

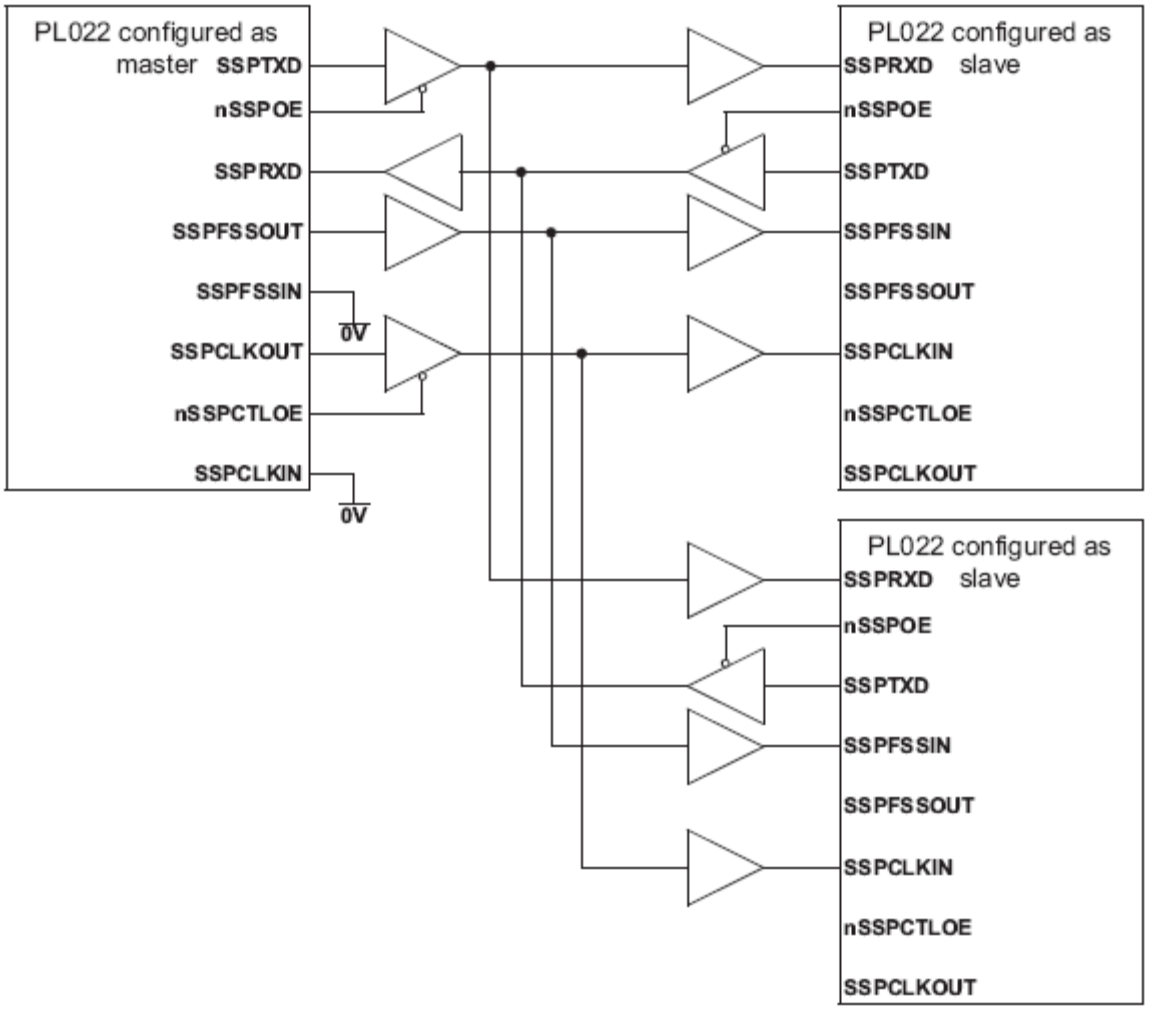


Рисунок 82. Ведущее устройство PrimeCell SSP подключено к двум ведомым

На рисунке показана совместная работа трех модулей PrimeCell SSP (PL022), 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 SSPTXD.

Для ответной передачи данных один из ведомых модулей переводит линию nSSPOE в активное состояние, разрешая таким образом прохождение сигнала от своей линии SSPTXD на вход SSPRXD ведущего.

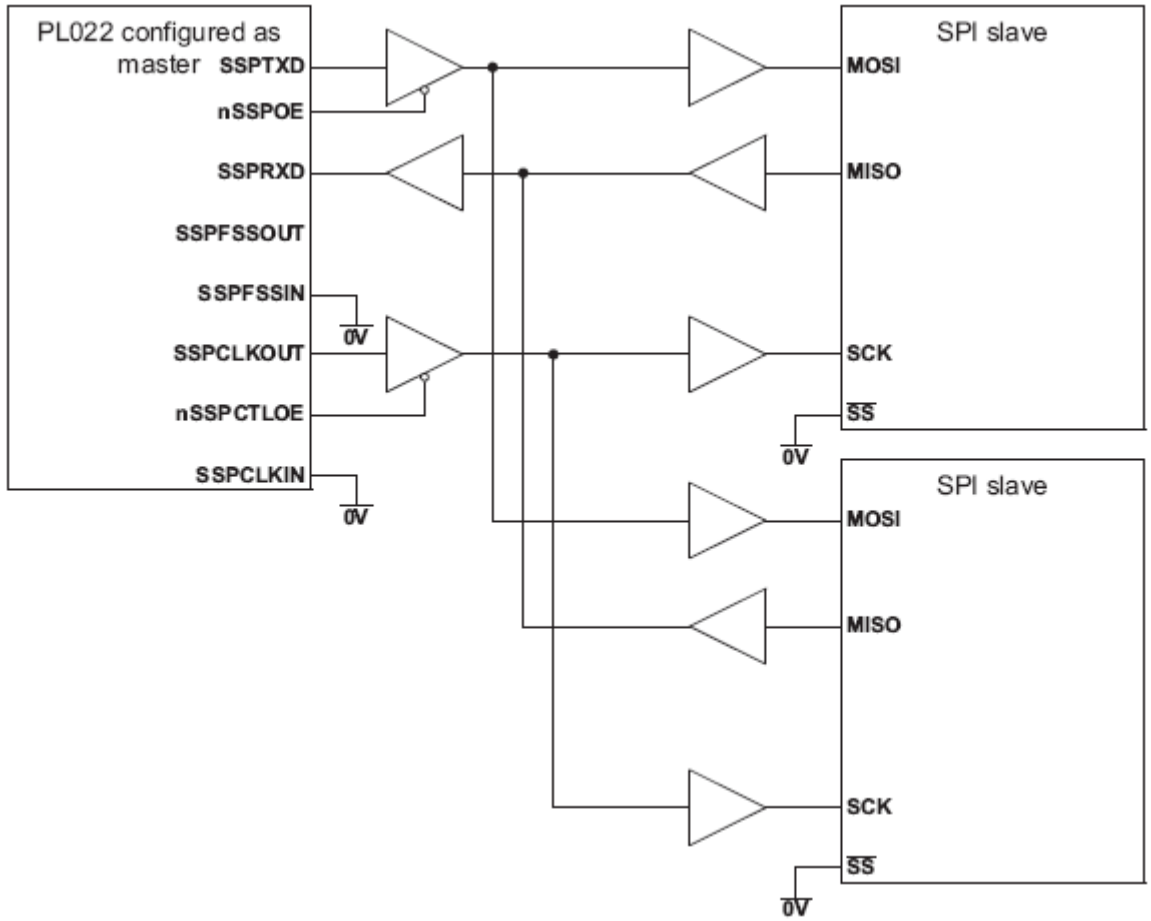


Рисунок 83. Ведущее устройство PrimeCell SSP подключено к двум ведомым, поддерживающим протокол SPI

На рисунке показано подключение модуля PrimeCell SSP (PL022), 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 SSPTXD. Ответная передача данных на входную линию SSPRXD ведущего устройства одновременно осуществляется только одним из ведомых по соответствующей линии MISO.

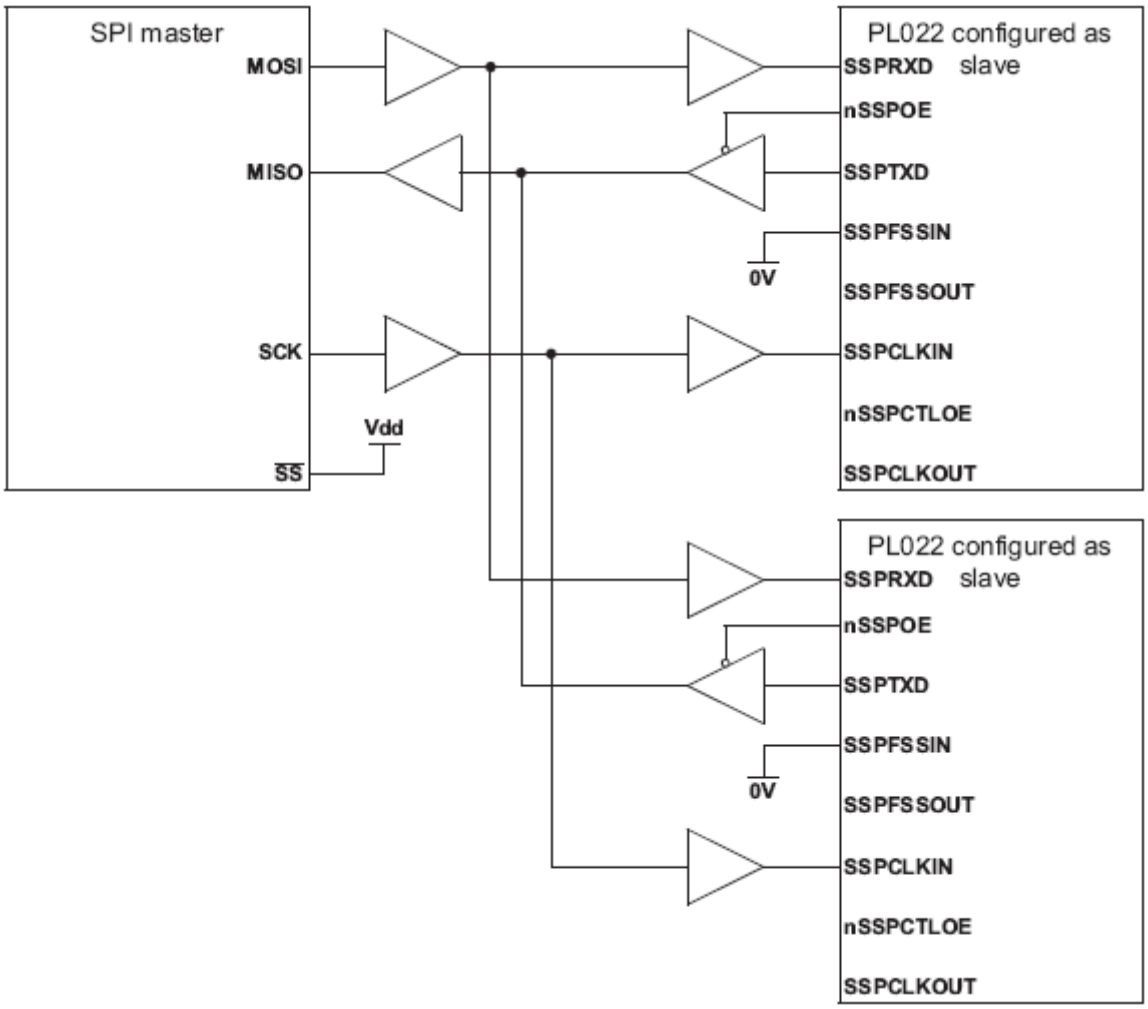


Рисунок 84. Ведущее устройство, поддерживающее протокол SPI подключено к двум ведомым модулям PrimeCell SSP

На рисунке показано ведущее устройство, поддерживающее протокол SPI фирмы Motorola, соединенное с двумя модулями PrimeCell SSP (PL022), 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 nSSPOE в активное состояние, разрешая, таким образом, прохождение сигнала от своей линии SSPTXD на вход SSPRXD ведущего.

Интерфейс прямого доступа к памяти

Модуль PrimeCell SSP предоставляет интерфейс подключения к контроллеру прямого доступа к памяти. Работа в данном режиме контролируется регистром управления ПДП SSPDMACR.

Интерфейс ПДП 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 ПДП, инициируется контроллером ПДП 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 ПДП, инициируется контроллером ПДП 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 исключающими, они могут быть инициированы одновременно. Например, в случае, если заполнение данными буфера приемника превышает пороговое значение четыре, 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. Тогда контроллер ПДП осуществит четыре передачи блоков по четыре символа, а оставшиеся три символа передаст в ходе трех одноэлементных обменов.

Примечание. Для оставшихся трех символов контроллер PrimeCell SSP не инициирует процедуру блочного обмена.

Каждый инициированный приемопередатчиком сигнал запроса ПДП остается активным до момента его сброса соответствующим сигналом DMACLR.

После снятия сигнала сброса модуль приемопередатчика вновь получает возможность сформировать запрос на ПДП в случае выполнения описанных выше условий. Все запросы ПДП снимаются после запрета работы приемопередатчика, а также в случае снятия сигнала разрешения ПДП.

В нижеследующей таблице приведены значения порогов заполнения буферов приемника и передатчика, необходимых для срабатывания запросов блочного обмена DMABREQ.

Таблица 336 Параметры срабатывания запросов блочного обмена данными в режиме ПДП

Пороговый	**Длина блока обмена данными**

уровень	Буфер передатчика (количество незаполненных ячеек)	Буфер приемника (количество заполненных ячеек)
½	4	4

На рисунке ниже показаны временные диаграммы одноэлементного и блочного запросов ПДП, в том числе действие сигнала DMACLR. Все сигналы должны быть синхронизированы с PCLK.

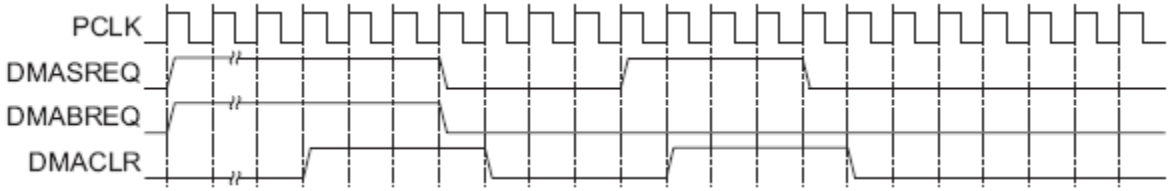


Рисунок 85. Временные диаграммы обмена в режиме ПДП

Программное управление модулем

Общая информация

Базовый адрес модуля не фиксирован и может быть различным в разных системах. Смещение каждого регистра относительно базового адреса постоянно.

Следующие адреса являются резервными и не должны использоваться в нормальном режиме функционирования:

- адреса со смещениями в диапазоне +0x028 ... +0x07C и +0xFD0 ... +0xFDC зарезервированы для перспективных расширений возможностей модуля;
- адреса со смещениями в диапазоне +0x080 ... +0x088 зарезервированы для тестирования.

Описание регистров контроллера SSP

Данные о регистрах модуля PrimeCell SSP приведены в нижеследующей таблице.

Таблица 337 Обобщенные данные о регистрах модуля PrimeCell SSP

Смещение	Наименование	Тип	Значение после сброса	Размер, бит	Описание
0x000	SSPCR0	RW	0x0000	16	Регистр управления 0
0x004	SSPCR1	RW	0x0	4	Регистр управления 1
0x008	SSPDR	RW	0x----	16	Буфера FIFO приемника (чтение) Буфер FIFO передатчика (запись)
0x00C	SSPSR	RO	0x03	3	Регистр состояния
0x010	SSPCPSR	RW	0x00	8	Регистр делителя тактовой частоты
0x014	SSPIMSC	RW	0x0	4	Регистр маски прерывания
0x018	SSPRIS	RO	0x8	4	Регистр состояния прерываний без учета маскирования
0x01C	SSPMIS	RO	0x0	4	Регистр состояния прерываний с учетом маскирования
0x020	SSPICR	WO	0x0	4	Регистр сброса прерывания
0x024	SSPDMACR	RW	0x0	2	Регистр управления прямым доступом к памяти
0x28-0x7C					Резерв
0x080-0x08C					Зарезервировано для тестирования
0x090-0xFCC					Резерв
0xFD0-0xFDC					Зарезервировано для расширенных кодов идентификации
0xFE0	SSPPeriphID0	RO	0x22	8	Регистр SSPPeriphID0
0xFE4	SSPPeriphID1	RO	0x10	8	Регистр SSPPeriphID1
0xFE8	SSPPeriphID2	RO	0x04	8	Регистр SSPPeriphID2
0xFEC	SSPPeriphID3	RO	0x00	8	Регистр SSPPeriphID3
0xFF0	SSPPCellID0	RO	0x0D	8	Регистр SSPPCellID0
0xFF4	SSPPCellID1	RO	0xF0	8	Регистр SSPPCellID1

0xFF8	SSPPCellID2	RO	0x05	8	Регистр SSPPCellID2
0xFFC	SSPPCellID3	RO	0xB1	8	Регистр SSPPCellID3

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 SSPCR0 содержит пять битовых полей, предназначенных для управления блоками модуля PrimeCell SSP. Назначение разрядов регистра представлено в таблице ниже.

Таблица 338 Формат регистра SSPCR0

Разряды	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 SSPCLKOUT (используется только в режиме обмена SPI фирмы Motorola). См. раздел «Формат SPI фирмы Motorola».
6	SPO	Полярность сигнала SSPCLKOUT (используется только в режиме обмена SPI фирмы Motorola). См. раздел «Формат 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 SSPCR1 содержит четыре битовых поля, предназначенных для управления блоками модуля PrimeCell SSP. Назначение разрядов регистра представлено в таблице ниже.

Таблица 339 Регистр SSPCR1

Разряды	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 PrimeCell SSP не должен в настоящее время осуществлять передачу данных в линию SSPTXD. При этом линии обмена данных ведомых устройств можно соединить параллельно. 0 – управление линией SSPTXD в ведомом режиме разрешена. 1 – управление линией SSPTXD 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 SSPDR 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 SSPTXD с заданной скоростью информационного обмена.

В случае, если выбран размер информационного слова менее 16 бит, перед записью в регистр SSPDR 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 PrimeCell 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 SSPDR описано в таблице ниже.

Таблица 340 Формат регистра SSPDR

Разряды	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 SSPDR 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 PrimeCell SSP.
В таблице ниже представлено назначение бит регистра SSPSR.

Таблица 341 Регистр SSPSR

Разряды	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 SSPCPSR 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 SSPCPSR нечетное число, его последующее чтение даст результатом это число, но с установленным в ноль младшим битом.

Назначение бит регистра SSPCPSR представлено в таблице ниже.

Таблица 342 Регистр SSPCPSR

Разряды	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 SSPIMSC показано в таблице ниже.

Таблица 343 Регистр SSPIMSC

Разряды	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 SSPRIS представлено в таблице ниже.

Таблица 344 Регистр SSPRIS

Разряды	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.

SSPMIS

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 SSPMIS представлено в таблице ниже.

Таблица 345 Регистр SSPMIS

Разряды	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 SSPICR представлено в таблице ниже.

Таблица 346 Регистр SSPICR

Разряды	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 UARTDMACR представлено в таблице ниже.

Таблица 347 Регистр SSPDMACR

Разряды	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.

Регистры идентификации оборудования SSPPeriphID0-3

Эти четыре восьмиразрядных регистра, расположенные по адресам 0xFE0–0xFEC и доступные только для чтения могут рассматриваться как один 32-разрядный регистр, содержащий следующую информацию:

- PartNumber[11:0] Идентифицирует тип периферийного устройства. Значение 0x022 соответствует синхронному последовательному приемопередатчику (SSP).
- Designer ID[19:12] Идентифицирует разработчика. Значение 0x41 (буква «A» в коде ASCII) соответствует ARM Ltd.
- Revision[23:20] Номер версии. Нумерация начинается с нуля и изменяется по мере разработки новых версий.
- Configuration[31:24] Вариант конфигурации периферийного устройства. Равен 0.

На рисунке ниже показано распределение бит в регистрах идентификации оборудования.

Рисунок 86. Распределение бит в регистре идентификации оборудования

Примечание. При распределении памяти в системе следует иметь в виду, что пространство адресов устройства занимает 4 Кбайт. Все обращения к регистрам идентификации должны быть 32-разрядными, с использованием инструкций LDR и STR.

Детальное описание восьмиразрядных регистров идентификации оборудования представлено в следующих подразделах.

Регистр SSPPeriphID0

В регистр записано фиксированное значение. См. таблицу ниже.

Таблица 348 Регистр SSPPeriphID0

Разряды	Наименование	Назначение
15…8		Резерв. При чтении не определено.
7…0	PartNumber0	0x22

Регистр SSPPeriphID1

В регистр записано фиксированное значение. См. таблицу ниже.

Таблица 349 Регистр SSPPeriphID1

Разряды	Наименование	Назначение
15…8		Резерв. При чтении не определено.
7…4	Designer0	0x1
3…0	PartNumber1	0x0

Регистр SSPPeriphID2

В регистр записано фиксированное значение. См. таблицу ниже.

Таблица 350 Регистр SSPPeriphID2

Разряды	Наименование	Назначение
15…8		Резерв. При чтении не определено.
7…4	Версия	Версия модуля.
3…0	Designer1	0x4

Регистр SSPPeriphID3

В регистр записано фиксированное значение. См. таблицу ниже.

Таблица 351 Регистр SSPPeriphID3

Разряды	Наименование	Назначение
15…8		Резерв. При чтении не определено.
7…0	Configuration	0x00

Регистры идентификации разработчика, SSPPCellID0-3

Эти четыре восьмиразрядных регистра, расположенные по адресам 0xFF0–0xFFC и доступные только для чтения могут рассматриваться как один 32-разрядный регистр, используются в качестве стандартного средства идентификации среди периферийных устройств.

Значение регистра SSPPCellID – 0xB105F00D.

На рисунке ниже показано распределение бит в регистрах SSPPCellID0-3.

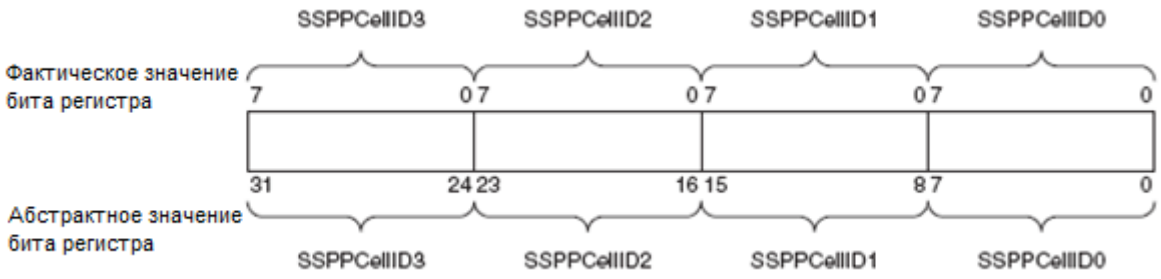


Рисунок 87. Распределение бит в регистрах идентификации разработчика.

Детальное описание восьмиразрядных регистров SSPPCellID0-3 представлено в следующих подразделах.

Регистр SSPPCellID0

В регистр записано фиксированное значение. См. таблицу ниже.

Таблица 352 Регистр SSPPCellID0

Разряды	Наименование	Назначение
15…8		Резерв. При чтении не определено.
7…0	SSPPCellID0	0x0D

Регистр SSPPCellID1

В регистр записано фиксированное значение. См. таблицу ниже.

Таблица 353 Регистр SSPPCellID1

Разряды	Наименование	Назначение
15…8		Резерв. При чтении не определено.
7…0	SSPPCellID0	0xF0

Регистр SSPPCellID2

В регистр записано фиксированное значение. См. таблицу ниже.

Таблица 354 Регистр SSPPCellID2

Разряды	Наименование	Назначение
15…8		Резерв. При чтении не определено.
7…0	SSPPCellID0	0x05

Регистр SSPPCellID3

В регистр записано фиксированное значение. См. таблицу ниже.

Таблица 355 Регистр SSPPCellID3

Разряды	Наименование	Назначение
15…8		Резерв. При чтении не определено.
7…0	SSPPCellID0	0xB1

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 SSPIMSC. 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 SSPRIS, либо из маскированного регистра прерываний SSPMIS.

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 типа «система на кристалле», совместимое с шиной AMBA (Advanced Microcontroller Bus Architecture), разработанное, испытанное и лицензированное компанией ARM.

Контроллер работает в качестве ведомого устройства, подключенного к шине APB (Advanced Peripheral Bus). В состав контроллера включен кодек (ENDEC – ENcoder/DEcoder) последовательного интерфейса инфракрасной передачи данных в соответствии с протоколом SIR (SIR – Serial Infra Red) ассоциации Infrared Data Association (IrDA).

Основные сведения о модуле представлены в следующих разделах:

- характеристики;
- программируемые параметры;
- отличия от приемопередатчика 16C650.

Примечание: вследствие изменений, внесенных в программную модель контроллера PL011, это изделие не обеспечивает обратной совместимости с предыдущей моделью PrimeCell UART PL010.

Основные характеристики модуля UART

Удовлетворяет спецификации AMBA Rev 2.0, что облегчает интеграцию модуля в систему на кристалле.

Может быть запрограммирован для использования как в качестве универсального асинхронного приемопередатчика, так и для инфракрасного обмена данными (SIR).

Содержит независимые буферы приема (32x12) и передачи (32x8) 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, стопового битов и 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 битов.

Формирование и обнаружения сигнала разрыва линии.

Поддержка функция управления модемом (линии CTS, DCD, DSR, RTS, DTR и RI).

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 – 32 элемента или один элемент, соответственно).
- Порог срабатывания прерывания по заполнению буферов FIFO (1/8, ¼, ½, ¾, 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.

Для проверки функционирования и соединений модуля предусмотрены дополнительные регистры тестирования.

Отличия от контроллера UART 16C650

Контроллер отличается от промышленного стандарта асинхронного приемопередатчика 16C650 следующими характеристиками:

- Пороги срабатывания прерывания по заполнению буфера FIFO приемника – 1/8, ¼, ½, ¾, и 7/8.
- Пороги срабатывания прерывания по заполнению буфера FIFO передатчика – 1/8, ¼, ½, ¾, и 7/8.
- Отличается распределение адресов внутренних регистров и назначение бит в регистрах.
- Недоступны изменения сигналов в состоянии модема.

Следующие возможности контроллера 16C650 не поддерживаются:

- Полуторная длительность стопового бита (поддерживается только 1 или 2 стоповых бита).
- Независимое задание тактовой частоты приемника и передатчика.

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.

Центральный процессор читает и записывает данные, а также управляющую информацию и информацию о состоянии через интерфейс шины AMBA APB. Прием и передача данных буферизуются с помощью внутренней памяти FIFO, позволяющей сохранить до 32 байт 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 индустриального стандарта - контроллера UART 16C650.

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 работа 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 (ПДП) для совместной работы с контроллером ПДП.

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 (ИК) 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

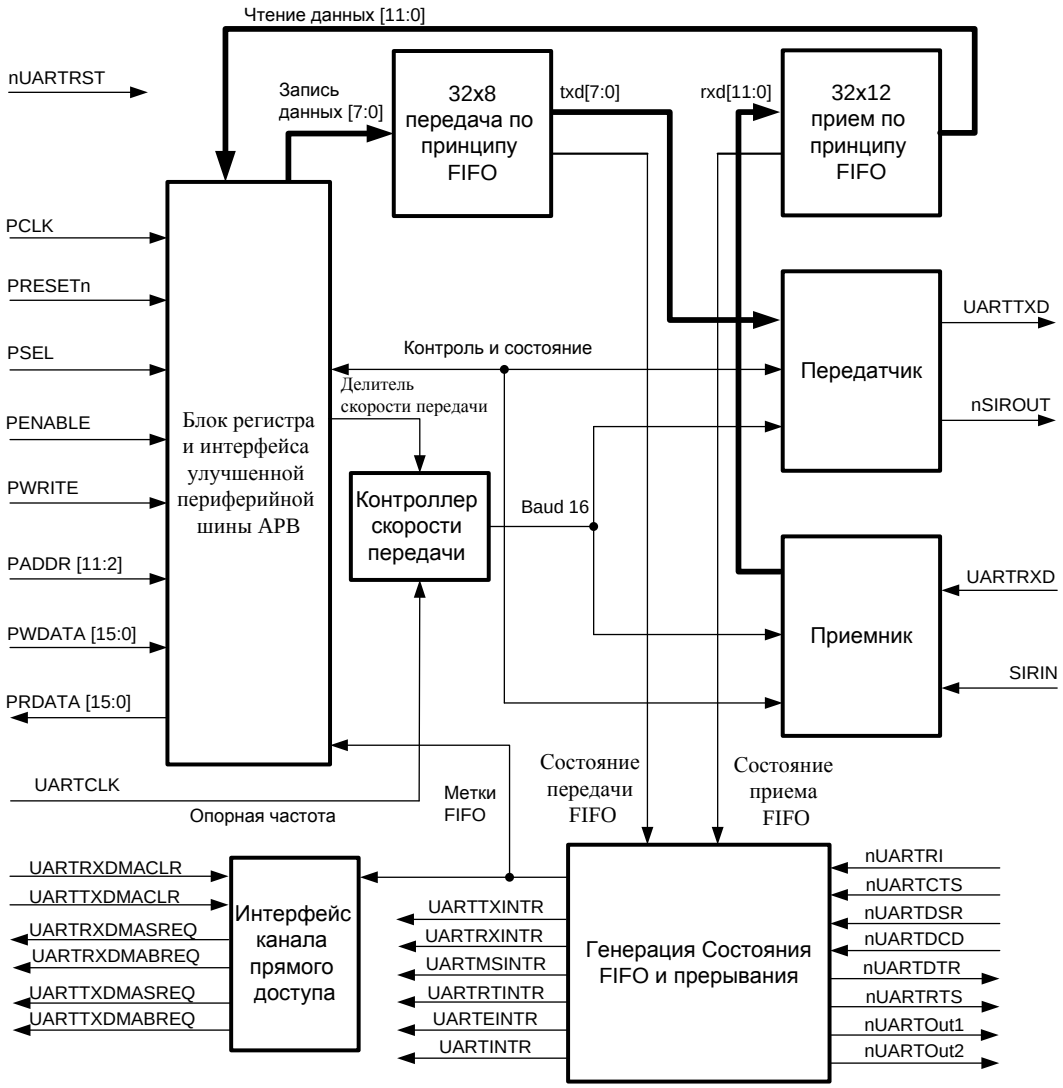


Рисунок 88. Блок-схема универсального асинхронного приёмопередатчика (УАПП)

Примечание. С целью обеспечения ясности на схеме не показаны схемы тестирования.

Генератор тактового сигнала приемопередатчика

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 32 слова, 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 32 слова, схему организации доступа типа «первый вошел, первый вышел». Принятые от периферийного устройства данные и соответствующие кодов 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, ошибки контроля четности, ошибки в структуре сигнала, а также разрыва 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.

Блок и регистры синхронизации

Контроллер поддерживает как асинхронный, так и синхронный режимы работы тактовых генераторов PCLK и UARTCLK. Регистры синхронизации и логика квитирования реализованы и 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 PCLK в область действия UARTCLK, так и наоборот.

Описание функционирования ИК кодека IrDA SIR

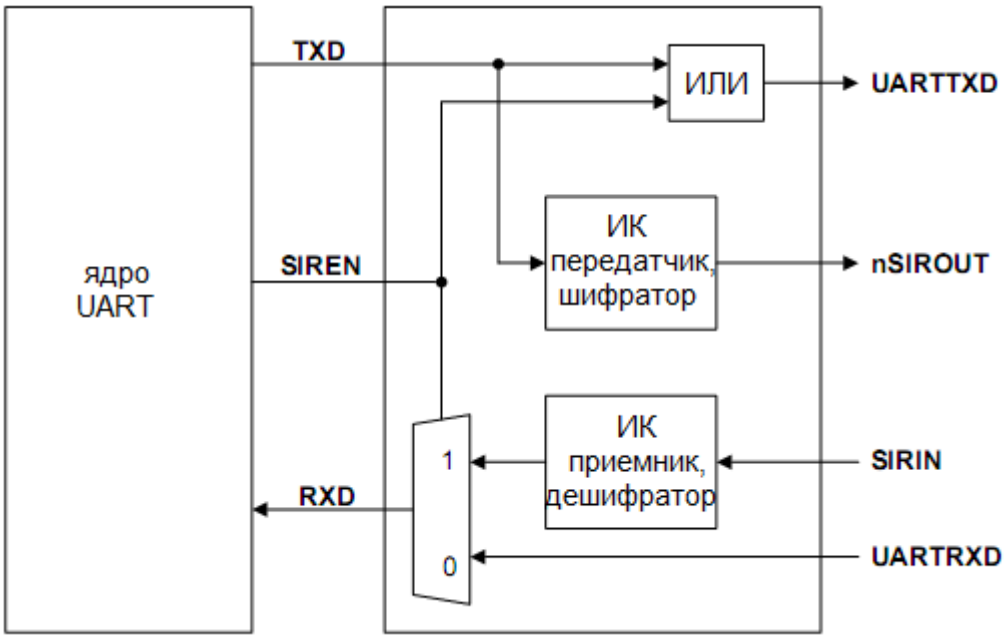


Рисунок 89. Структурная схема кодека

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, который, в свою очередь, формируется 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 приводит к увеличению джиттера 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

Сброс модуля

Приемопередатчик и кодек могут быть сброшены общим сигналом сброса PRESETn, а также специфическим для модуля сигналом nUARTRST. Схема сброса должна использовать сигнал PRESETn для активизации сигнала nUARTRST в асинхронном режиме и его снятия синхронно с UARTCLK. Сигнал PRESETn должен быть

установлен в низкий уровень в течение периода времени, достаточного для сброса самого медленного блока системы на кристалле, после чего переведен обратно в высокий уровень. В случае рассматриваемого модуля приемопередатчика необходимо, чтобы сигнал PRESETn находился в низком уровне в течение, как минимум, одного периода PCLK.

Тактовые сигналы

Частота, тактового сигнала UARTCLK должна обеспечивать поддержку требуемого диапазона скоростей передачи данных:

F_UARTCLK(min) >= 16 * baud_rate_max;

F_UARTCLK(max) <= 16 * 65535 * 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 PCLK и UARTCLK. Частота UARTCLK должна быть не более чем в 5/3 раз выше частоты PCLK.

F_UARTCLK <= 5/3 * F_PCLK.

Например, при работе в режиме UART с максимальной скоростью передачи данных 921600 бод, при частоте UARTCLK 14,7456 МГц, частота PCLK 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.

Дробный коэффициент деления

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 * скорость передачи данных) = BRD_I + BRD_F,
где BRD_I – целая часть, а BRD_F – дробная часть коэффициента деления.

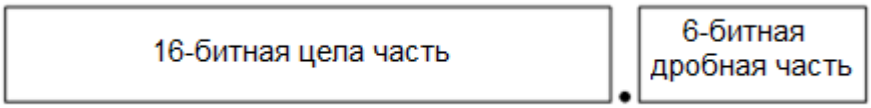


Рисунок 90.

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:

$$M = integer(BRD\_F * 2^n + 0.5),$$

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 больше требуемой скорости обмена данными.

Передача и прием данных

Принятые или передаваемые данные заносятся в 32-элементные буфера 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 состояния в запрещенное.

Для каждого отсчета данных (в приемной линии) производится три измерения уровня, решение принимается по принципу большинства голосов. В следующих разделах будет определено понятие среднего отсчета, при этом в качестве дополнительных двух отсчетов берутся предыдущий и последующий – во всем остальном документ про это нет ни слова – я бы убрал этот текст.

В случае если приемник находился в неактивном состоянии (на линии входного сигнала UARTRXD 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 UARTRXD 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 UARTRXD). 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.

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 таблице ниже 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, в то время как 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.

Таблица 356 Назначение бит буфера FIFO

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:

1. 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.
2. 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 МГц < F_IrLPBaud16 < 2,12 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 UARTTCR.

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.

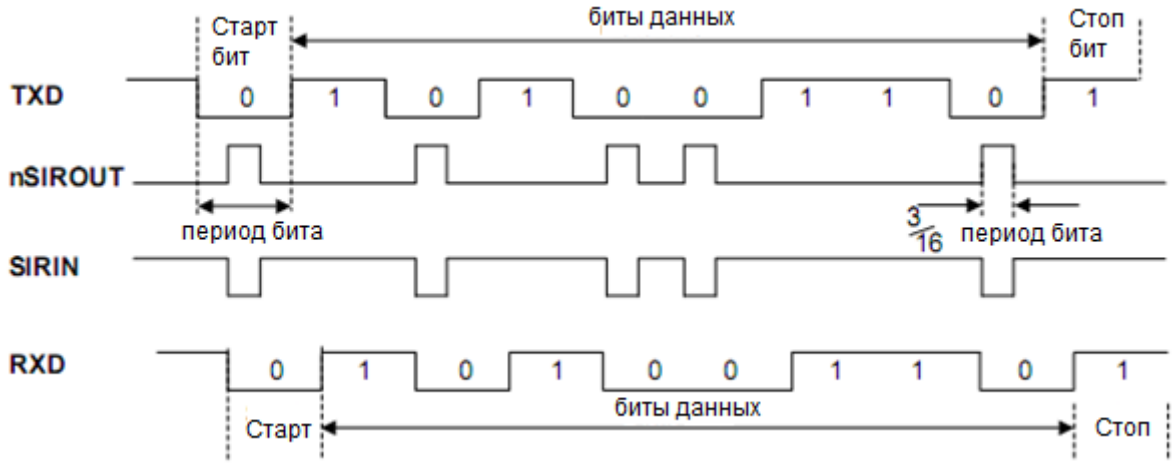


Рисунок 91. Модуляция данных IrDA

Линии управления модемом

Модуль универсального асинхронного приемопередатчика может использоваться как в режиме оконечного оборудования (DTE), так и в режиме оборудования передачи данных (DCE). На рисунке ниже показаны сигналы модема в режиме DTE. Назначение сигналов в режиме DCE представлено в таблице ниже.

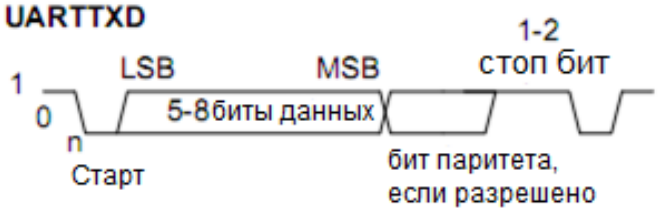


Рисунок 92. Кадр передачи данных

Таблица 357 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 рисунке ниже.

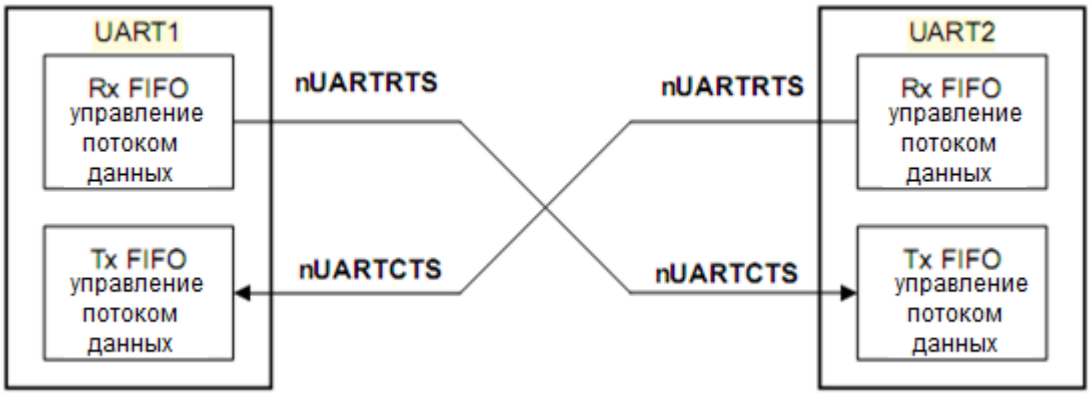


Рисунок 93. Взаимодействие двух устройств последовательной связи с аппаратным управлением потоком данных

Если разрешено управление потоком данных по сигналу RTS, линия nUARTRTS переводится в активное состоянии 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 таблице ниже показаны необходимые установки для различных режимов управления потоком данных.

Таблица 358 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 состоянии 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 его заполнение 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, однако 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 ПДП UARTDMACR.

Интерфейс ПДП 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 UARTIFLS17).
- UARTRXDMACLR – сброс запроса на ПДП, 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 ПДП, инициируется контроллером ПДП 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 исключающими, 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 ПДП 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 ПДП остается активным до момента его сброса соответствующим сигналом DMACLR.

После снятия сигнала сброса модуль приемопередатчика вновь получает возможность сформировать запрос на ПДП в случае выполнения описанных выше условий. Все запросы ПДП снимаются после запрета работы приемопередатчика, а также в случае установки в ноль бита управления ПДП TXDMAE или RXDMAE в регистре управления ПДП UARTDMACR.

В случае запрета буферов FIFO устройство способно передавать и принимать только одиночные символы, как следствие, контроллер может инициировать ПДП только в одноэлементном режиме. При этом модуль в состоянии формировать только сигналы управления ПДП 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. В таблице ниже приведены значения параметров срабатывания запросов блочного обмена UARTRXDMABREQ и UARTTXDMABREQ в зависимости от порога заполнения буфера.

Таблица 359 Параметры срабатывания запросов блочного обмена данными в режиме ПДП

Пороговый уровень	**Длина блока обмена данными**	
	Буфер передатчика	**Буфер приемника**

	(количество незаполненных ячеек)	**(количество заполненных ячеек)**
1/8	28	4
¼	24	8
½	16	16
¾	8	24
7/8	4	28

В регистре управления ПДП UARTDMACR предусмотрен бит DMAONERR, который позволяет запретить ПДП от приемника в случае активного состояния линии прерывания по обнаружению ошибка UARTEINTR. При этом соответствующие линии запроса ПДП – UARTRXDMASREQ и UARTRXDMABREQ переводятся в неактивное состояние (маскируются) до сброса UARTEINTR. На линии запроса ПДП, обслуживающие передатчик, состояние UARTEINTR не влияет.

На рисунке ниже показаны временные диаграммы одноэлементного и блочного запросов ПДП, в том числе действие сигнала DMACLR. Все сигналы должны быть синхронизированы с PCLK. В интересах ясности изложения предполагается, что синхронизация сигналов запроса ПДП в контроллере ПДП не производится.

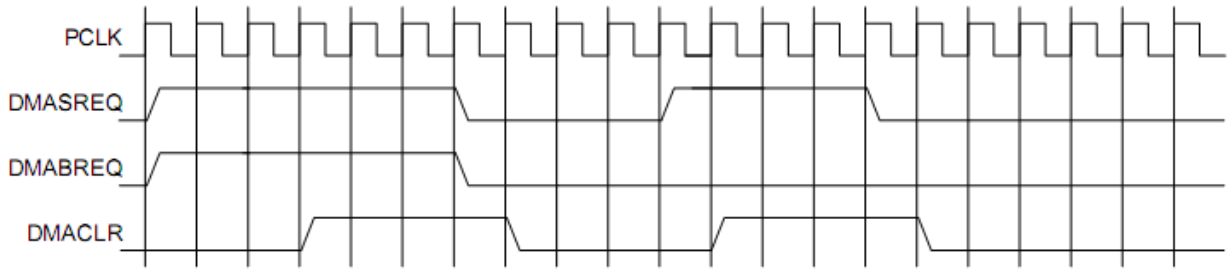


Рисунок 94. Одноэлементный и блочный запросы ПДП

Прерывания

В модуле предусмотрено 11 маскируемых источников прерывания. Их комбинации формируют пять независимых выходных сигналов запроса на прерывание, а также один общий сигнал, представляющий собой комбинацию независимых 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, 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, согласованной 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:

1. 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.
2. 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:

3. 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.
4. 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 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, UARTTXINTR, 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 не фиксирован и может быть различным в разных системах. 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, прим. перев.) устанавливаются в значение 0 после сброса по включению питания или системного сброса.

Столбец «тип» в нижеследующей таблице 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 таблице ниже.

Таблица 360 Обобщенные данные о регистрах устройства

Смещение	**Наименование**	**Тип**	**Значение после сброса**	**Размер, бит**	**Описание**
0x000	UARTDR	RW	0x---	12/8	Регистр данных
0x004	UARTRSR/ UARTECR	RW	0x0	4/0	Регистра состояния приемника/Сброс ошибки приемника
0x008-0x014					Резерв
0x018	UARTFR	RO	0b-10010---	9	Регистр флагов
0x01C					Резерв
0x020	UARTILPR	RW	0x00	8	Регистр управления ИК обменом в режиме пониженного энергопотребления
0x024	UARTIBRD	RW	0x0000	16	Целая часть делителя скорости обмена данными
0x028	UARTFBRD	RW	0x00	6	Дробная часть делителя скорости обмена данными
0x02C	UARTLCR_H	RW	0x00	8	Регистр управления линией
0x030	UARTCR	RW	0x0300	16	Регистр управления
0x034	UARTIFLS	RW	0x12	6	Регистр порога прерывания по заполнению буфера FIFO
0x038	UARTIMSC	RW	0x000	11	Регистр маски прерывания
0x03C	UARTRIS	RO	0x00-	11	Регистр состояния прерываний
0x040	UARTMIS	RO	0x00-	11	Регистр состояния прерываний с маскированием
0x044	UARTICR	WO	-	11	Регистр сброса прерывания
0x048	UARTDMACR	RW	0x00	3	Регистр управления ПДП
0x04C-0x07C					Резерв
0x080-0x08C					Регистры тестирования
0x090-0xFCC					Резерв
0xFD0-0xFDC					Зарезервировано для расширенных кодов идентификации

0xFE0	UARTPeriphID0	RO	0x11	8	Регистр UARTPeriphID0
0xFE4	UARTPeriphID1	RO	0x10	8	Регистр UARTPeriphID
0xFE8	UARTPeriphID2	RO	0x_41	8	Регистр UARTPeriphID2
0xFEC	UARTPeriphID3	RO	0x00	8	Регистр UARTPeriphID3
0xFF0	UARTPCellID0	RO	0x0D	8	Регистр UARTPCellID0
0xFF4	UARTPCellID1	RO	0xF0	8	Регистр UARTPCellID1
0xFF8	UARTPCellID2	RO	0x05	8	Регистр UARTPCellID2
0xFFC	UARTPCellID3	RO	0xB1	8	Регистр UARTPCellID3

UARTx_DR

Регистр данных.

В ходе передаче данных:

Если буфер FIFO передатчика разрешен, то слово данных, записанное 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 UARTRSR/UARTECR (см. Таблица 362).

Таблица 361 Формат регистра UARTDR

Разряды	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.

[1] Значение зависит от версии контроллера UART.

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.

UARTx_RSR_ECR

Регистр состояния приемника / сброса ошибки.

Состояние приемника также может быть считано из регистра UARTRSR. 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 UARTDR. С другой стороны, признак переполнения буфера устанавливается немедленно после возникновения этого состояния (и не связан с последним считанным из регистра UARTDR байтом данных).

Запись в регистр UARTECR приводит к сбросу признаков ошибок переполнения, четности, структуры кадра, разрыва линии. Кроме того, все эти признаки устанавливаются в 0 после сброса устройства.

В таблице ниже представлено назначение бит регистра UARTRSR/UARTECR.

Таблица 362 Регистр UARTRSR/UARTECR

Разряды	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 UARTLCR_H. Бит сбрасывается в 0 после записи в регистр UARTECR. 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 UARTECR. При включенном FIFO данная ошибка ассоциируется с символом, находящимся на вершине буфера.

Примечание. Перед чтением регистра состояния UARTRSR необходимо считать данные, принятые из линии, путем обращения к регистру данных UARTDR. Противоположная последовательность действий не допускается, так как регистр UARTRSR обновляет свое состояние только после чтения регистра UARTDR. Вместе с тем, информация о состоянии приемника может быть получена непосредственно из регистра данных UARTDR.

UARTx_FR

Регистр флагов.

После сброса биты регистра флагов TXFF, RXFF и BUSY устанавливаются в 0, а биты TXFE и RXFE – в 1. В таблице ниже представлена информация о назначении битов регистра.

Таблица 363 Регистр UARTFR

Разряды	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 UARTLCR_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 UARTLCR_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 UARTLCR_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 UARTLCR_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

UARTx_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 таблице ниже.

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 МГц < F_IrLPBaud16 < 2,12 МГц, что, в свою очередь, гарантирует формирование кодеком импульсов данных с длительностью 1,41-2,11мкс (в три раза длиннее периода сигнала IrLPBaud16).

Таблица 364 Регистр UARTILPR

Разряды	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.

UARTx_IBRD

Регистр целой части делителя скорости передачи данных.

Назначение бит регистра представлено в таблице ниже.

Таблица 365 Регистр UARTBIRD

Разряды	Наименование	Назначение
15…0	BAUDDIV_INT	Целая часть коэффициента деления частоты для формирования тактового сигнала передачи данных. После сброса устанавливается в 0.

UARTx_FBRD

Регистр дробной части делителя скорости передачи данных.

Назначение бит регистра представлено в таблице ниже.

Таблица 366 Регистр UARTBFRD

Разряды	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 * 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 UARTIBRD и UARTFBRD 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 UARTIBRD, равное 0 является недопустимым, при этом значение регистра UARTFBRD игнорируется.

Аналогично, при UARTIBRD, равном 65535 (0xFFFF), значение UARTFBRD 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*10^6)/(16*230400) = 1,085.
Таким образом, BRDI = 1, BRDF = 0,085.
Следовательно, значение, записываемое в регистр UARTBFRD, равно
m = integer((0,085*64)+0,5) = 5.

Реальное значение коэффициента деления = 1+5/64 = 1,078.
Реальная скорость передачи данных = (4*10^6)/(16*1,078) = 231911 бит/с.
Ошибка установки скорости = (231911-230400)/230400 * 100% = 0,656%.

Максимальная ошибка установки скорости передачи данных с использованием шестиразрядного регистра UARTBFRD = 1/64*100% = 1,56%. Такая ошибка возникает в случае m = 1, при этом разница накапливается в течение 64 тактовых интервалов.

В таблице ниже 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 UARTFBRD должен быть записан ноль.

Таблица 367 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 нижеследующей таблице приведены значения коэффициента деления для типичных скоростей передачи данных при частоте UARTCLK = 4 МГц.

Таблица 368 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

UARTx_LCR_H

Регистр управления линией.

Данный регистр обеспечивает доступ к разрядам с 29 по 22 регистра UARTLCR. При сбросе все биты регистра UARTLCR_H обнуляются.
Назначение разрядов регистра описано в таблице ниже.

Таблица 369 Регистр UARTLCR_H

Разряды	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 запрещены.
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
1	PEN	Разрешение проверки четности. 0 – кадр не содержит бита четности, 1 – бит четности передается в кадре и проверяется при приеме данных
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 UARTLCR_H, UARTIBRD и UARTFBRD совместно образует общий 30-разрядный регистр UARTLCR, который обновляется по стробу, формируемому при записи в UARTLCR_H. Таким образом, для того чтобы изменение параметров коэффициента деления частоты обмена данными вступило в силу, после их изменения значения регистров UARTIBRD и/или UARTFBRD необходимо осуществить запись данных в регистр UARTLCR_H.

Примечания:

Изменение значений трех регистров можно осуществить корректно двумя способами:

- Запись UARTIBRD, запись UARTFBRD, запись UARTLCR_H.
- Запись UARTFBRD, запись UARTIBRD, запись UARTLCR_H.

Для того чтобы изменить значение лишь одного из регистров (UARTIBRD или UARTFBRD) необходимо выполнить следующие шаги:

- Запись UARTIBRD (или UARTFBRD).
- Запись UARTLCR_H.

В таблице ниже приведена таблица истинности для бит управления контролем четности SPS, EPS, PEN регистра управления линией UARTLCR_H.

Таблица 370 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 UARTLCR_H, UARTIBRD, and UARTFBRD 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.

UARTx_CR

Регистр управления.

После сброса все биты регистра управления, за исключением бит 9 и 8, устанавливаются в нулевое состояние. Биты 9 и 8 устанавливаются в единичное состояние.

Назначение разрядов регистра управления показано в таблице ниже.

Таблица 371 Регистр управления UARTCR

Разряды	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 UARTTCR (стр. 4-5) SIRTEST=1, то сигнал с выхода кодека nSIROUT инвертируется и подается на вход кодека 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 кодека 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.

Примечания:

1. 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.

2. 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 UARTLCR_H в 0.
- Измените настройки регистра UARTCR.
- Возобновите работу приемопередатчика.

UARTx_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 таблице ниже.

Таблица 372 Регистр UARTIFLS

Разряды	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

UARTIMSC

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 UARTIMSC показано в таблице ниже.

Таблица 373 Регистр UARTIMSC

Разряды	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.

UARTx_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 бит прерывания по состоянию модема (биты с 3 по 0), устанавливаются в 0. Значение бит прерывания по состоянию модема после сброса не определено.

Назначение бит в регистре UARTRIS представлено в таблице ниже.

Таблица 374 Регистр UARTRIS

Разряды	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[2].
5	TXRIS	Состояние прерывания от передатчика UARTTXINTR.
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.

[2] Сигнал маски прерывания по таймауту используется в качестве разрешения перехода в режим пониженного энергопотребления. Поэтому чтение состояния прерывания по таймауту из регистров UARTMIS и UARTRIS даст одинаковый результат.

0	RIRMIS	Состояние прерывания UARTRIINTR по изменению линии nUARTRI.

UARTx_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 бит прерывания по состоянию модема (биты с 3 по 0), устанавливаются в 0. Значение бит прерывания по состоянию модема после сброса не определено.

Назначение бит в регистре UARTMIS представлено в таблице ниже.

Таблица 375 Регистр UARTMIS

Разряды	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.

UARTx_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 UARTICR представлено в таблице ниже.

Таблица 376 Регистр UARTICR

Разряды	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.

UARTx_DMACR

Регистр управления прямым доступом к памяти.

Регистр доступен по чтению и записи. После сброса все биты регистра обнуляются. Назначение бит регистра UARTDMACR представлено в таблице ниже.

Таблица 377 Регистр UARTDMACR

Разряды	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 ПДП от приемника UARTRXDMASREQ и UARTRXDMABREQ.
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.

Регистры идентификации оборудования UARTPeriphID0-3

Эти четыре восьмиразрядных регистра, расположенные по адресам 0xFE0–0xFEC и доступные только для чтения могут рассматриваться как один 32-разрядный регистр, содержащий следующую информацию:

- PartNumber[11:0] Идентифицирует тип периферийного устройства. Значение 0x011 соответствует асинхронному приемопередатчику (UART).

- Designer ID[19:12] Идентифицирует разработчика. Значение 0x41 соответствует ARM.
- Revision[23:20] Номер версии.
- Configuration[31:24] Вариант конфигурации периферийного устройства. Равен 0.

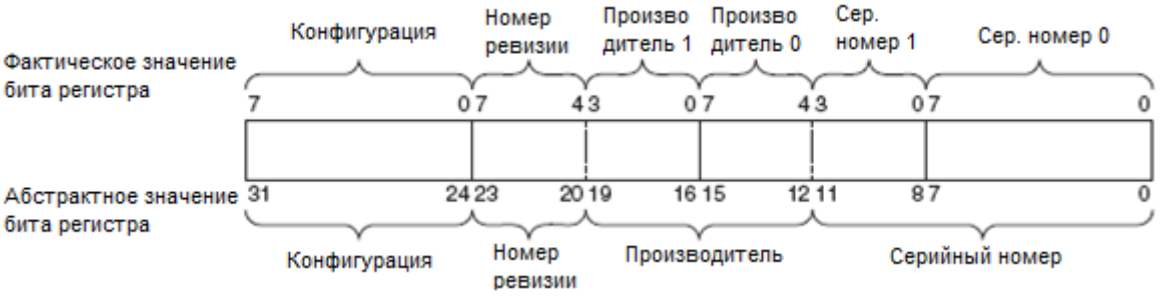


Рисунок 95. Распределение бит в регистрах идентификации оборудования

Примечание: При распределении памяти в системе следует иметь в виду, что пространство адресов устройства занимает 4 Кбайт. Все обращения к регистрам идентификации должны быть 32-разрядными, с использованием инструкций LDR и STR.

Детальное описание восьмиразрядных регистров идентификации оборудования представлено в следующих подразделах.

Регистр UARTPeriphID0

В регистр записано фиксированное значение. См. таблицу ниже.

Таблица 378 Регистр UARTPeriphID0.

Разряды	Наименование	Назначение
15…8		Резерв. При чтении не определено
7…0	PartNumber0	0x11

Регистр UARTPeriphID1

В регистр записано фиксированное значение. См. таблицу ниже.

Таблица 379 Регистр UARTPeriphID1.

Разряды	Наименование	Назначение
15…8		Резерв. При чтении не определено
7…4	Designer0	0x1
3…0	PartNumber1	0x0

Регистр UARTPeriphID2

В регистр записано фиксированное значение. См. таблицу ниже.

Таблица 380 Регистр UARTPeriphID2.

Разряды	Наименование	Назначение
15…8		Резерв. При чтении не определено
7…4	Версия	В зависимости от версии приемопередатчика: r1p0 0x0 r1p1 0x1 r1p3 0x2 r1p4 0x2 r1p5 0x3
3…0	Designer1	0x4

Регистр UARTPeriphID3

В регистр записано фиксированное значение. См. таблицу ниже.

Таблица 381 Регистр UARTPeriphID3

Разряды	Наименование	Назначение
15…8		Резерв. При чтении не определено
7…0	Configuration	0x00

Регистры идентификации оборудования PrimeCell, UARTPCellID0-3

Эти четыре восьмиразрядных регистра, расположенные по адресам 0xFF0–0xFFC и доступные только для чтения могут рассматриваться как один 32-разрядный регистр, используются в качестве стандартного средства идентификации среди периферийных устройств (used as a standard cross-peripheral identification system).

Значение регистра UARTPCellID – 0xB105F00D.

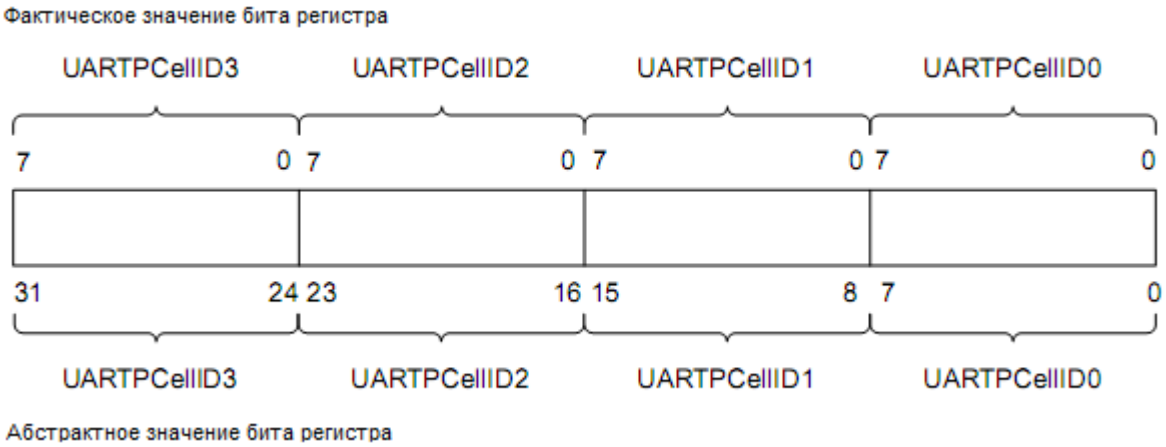


Рисунок 96. Распределение бит в регистрах UARTPCellID0-3

Детальное описание восьмиразрядных регистров UARTPCellID0-3 представлено в следующих подразделах.

Регистр UARTPCellID0

В регистр записано фиксированное значение. См. таблицу ниже.

Таблица 382 Регистр UARTPCellID0

Разряды	Наименование	Назначение
15…8		Резерв. При чтении не определено
7…0	UARTPCellID0	0x0D

Регистр UARTPCellID1

В регистр записано фиксированное значение. См. таблицу ниже.

Таблица 383 Регистр UARTPCellID1

Разряды	Наименование	Назначение
15…8		Резерв. При чтении не определено
7…0	UARTPCellID0	0xF0

Регистр UARTPCellID2

В регистр записано фиксированное значение. См. таблицу ниже.

Таблица 384 Регистр UARTPCellID2

Разряды	Наименование	Назначение
15…8		Резерв. При чтении не определено
7…0	UARTPCellID0	0x05

Регистр UARTPCellID3

В регистр записано фиксированное значение. См. таблицу ниже.

Таблица 385 Регистр UARTPCellID3

Разряды	Наименование	Назначение
15…8		Резерв. При чтении не определено
7…0	UARTPCellID0	0xB1

Контроллер прямого доступа в память DMA

Основные свойства контроллера DMA

Основные свойства и отличительные особенности:

- настраиваемое количество каналов ПДП;
- каждый канал ПДП имеет свои сигналы управления передачи данных;
- каждый канал ПДП имеет программируемый уровень приоритета;
- каждый уровень приоритета обрабатывается исходя из уровня приоритета, определяемого номером канала ПДП;
- поддержка различного типа передачи данных:
  - память – память;
  - память – периферия;
  - периферия – память;
- поддержка различных типов ПДП циклов;
- поддержка передачи данных различной разрядности;
- каждому каналу ПДП доступна первичная и альтернативная структура управляющих данных канала;
- все управляющие данные канала хранятся в системной памяти в формате «первый – младший значащий разряд»;
- все данные передаются с использованием одиночных AHB-Lite пакетов;
- разрядность данных приемника равна разрядности данных передатчика;
- количество передач в одном цикле ПДП может программироваться от 1 до 1024;
- инкремент адреса передачи может быть больше чем разрядность данных;
- наличие выходного сигнала состояния ошибки на шине AHB.

Термины и определения

При описании блока используются следующие термины:

Альтернативная	Альтернативная структура управляющих данных канала. Вы можете установить соответствующий регистр для изменения типа структуры данных (см. раздел «Структура управляющих данных канала»).
C	Идентификатор номера канала прямого доступа. Например: C=1 – канал DMA 1 C=23 – канал DMA 23
Канал	Возможны конфигурации контроллера с числом каналов до 32. Каждый канал содержит независимые сигналы управления передачей данных, которые могут инициировать передачу данных по каналу DMA.
Управляющие данные канала	Структура данных находится в системной памяти. Вы можете программировать эту структуру данных так, что контроллер может выполнять передачу данных по каналу DMA в желаемом режиме. Контроллер должен иметь доступ к области системной памяти, где находится эта информация.

	Примечание. Любое упоминание в спецификации структуры данных означает управляющие данные канала.
Цикл DMA	Все передачи DMA, которые контроллер должен выполнить для передачи N пакетов данных.
Передача DMA	Акция пересылки одного байта, полуслова или слова. Общее количество передач DMA, которые контроллер выполняет для канала.
Пинг – понг	Режим работы для выбранного канала, при котором контроллер получает начальный запрос и затем выполняет цикл DMA, используя первичную или альтернативную структуру данных. После завершения этого цикла DMA контроллер начинает выполнять новый цикл DMA, используя другую (первичную или альтернативную) структуру данных. Контроллер сигнализирует об окончании каждого цикла DMA, позволяя главному процессору перенастраивать неактивную структуру данных. Контроллер продолжает переключаться от первичной к альтернативной структуре данных и обратно до тех пор, пока он не прочитает «неправильную» структуру данных или пока он не завершит цикл без переключения к другой структуре.
Первичная	Первичная структура управляющих данных канала. Контроллер использует эту структуру данных, если соответствующий разряд в регистре chnl_pri_alt_set установлен в 0.
R	Степень числа 2, устанавливающая число передач DMA, которые могут произойти перед сменой арбитража. Количество передач DMA программируется в диапазоне от 1 до 1024 двоичными шагами от 2 в степени 0 до 2 в степени 100.
Исполнение с изменением конфигурации	Режим работы для выбранного канала, при котором контроллер получает запрос от периферии и выполняет 4 DMA передачи, используя первичную структуру управляющих данных, которые настраивают альтернативную структуру управляющих данных. После чего контроллер начинает цикл DMA, используя альтернативную структуру данных. После того, как цикл закончится и если периферия устанавливает новый запрос на обслуживание, контроллер выполняет снова 4 DMA передачи, используя первичную структуру управляющих данных, которые опять перенастраивают альтернативную структуру управляющих данных. После чего контроллер начинает цикл DMA, используя альтернативную структуру данных. Контроллер будет продолжать работать вышеописанным способом до тех пор, пока не прочитает неправильную структуру данных или процессор не установит альтернативную структуру данных для обычного цикла. Контроллер устанавливает флаг dma_done, если окончание подобного режима работы происходит после выполнения обычного цикла.

Функциональное описание

На рисунке ниже показана упрощенная структурная схема контроллера.

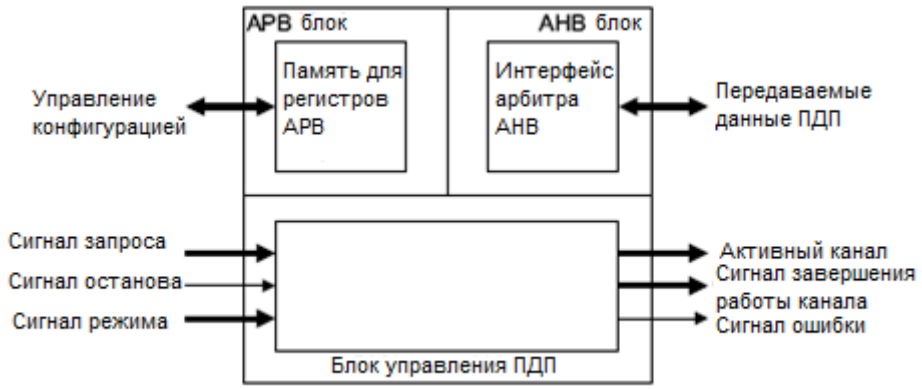


Рисунок 97. Структурная схема контроллера

Контроллер состоит из следующих основных функциональных блоков:

- блок, подключенный к шине APB;
- блок, подключенный к шине AHB;
- управляющий блок ПДП.

Блок, подключенный к шине APB

Блок содержит набор регистров, позволяющих настраивать контроллер, используя ведомый APB интерфейс. Регистры занимают адресное пространство емкостью 4 Кбайт.

Блок, подключенный к шине AHB

Контроллер содержит один блок типа «ведущий» шины AHB-Lite, который позволяет, используя 32-разрядную шину, передавать данные от источника к приемнику. Источник и приемник являются ведомыми шины AHB. Контроллер соответствует протоколу AMBA 3 AHB-Lite. Подробное описание протокола приведено в документе «Описание протокола AMBA 3 AHB-Lite v1.0».

Управляющий блок ПДП

Этот блок содержит схему управления, позволяющую реализовать следующие функции:

- осуществление арбитража поступающих запросов;
- индикацию активного канала;
- индикацию завершения обмена по каналу;
- индикацию состояния ошибки обмена по интерфейсу AHB-Lite;
- разрешение медленным устройствам приостанавливать исполнение цикла ПДП;
- ожидание запроса на очистку до завершения цикла ПДП;
- осуществление одиночных или множественных передач ПДП для каждого запроса;
- осуществление следующих типов ПДП передач:
  - память – память;
  - память – периферия;

- периферия – память.

Примечание:
Передачи типа периферия – периферия не поддерживаются, так как каждый канал имеет один интерфейс запроса ПДП.

На рисунке ниже показаны сигналы управления интерфейса ПДП.

Рисунок 98. Сигналы интерфейса управления ПДП

Пример использования блока ПДП

На рисунке ниже показана примерная структурная схема микроконтроллера с использованием контроллера ПДП.

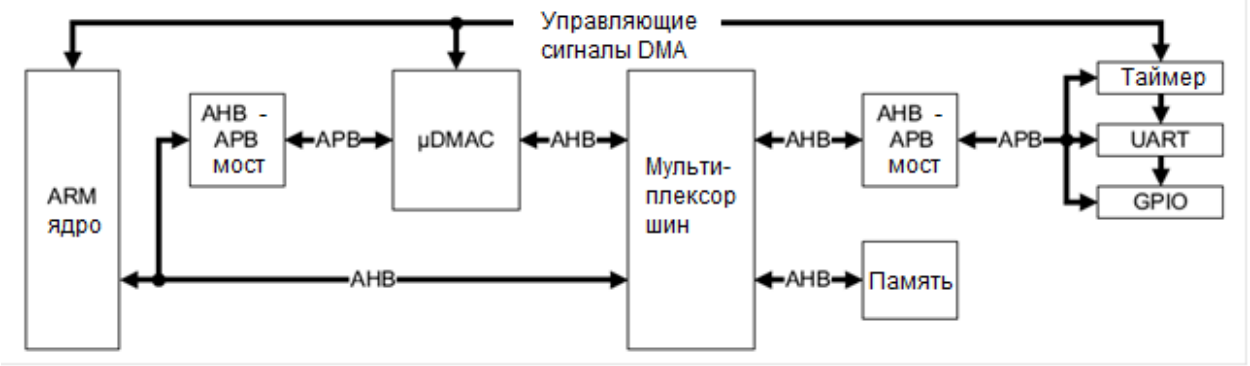


Рисунок 99. Пример структурной схемы микроконтроллера с использованием контроллера ПДП

Типы передач

Контроллер интерфейса не поддерживает пакетные передачи и поэтому сигнал HBURST удерживается всегда в состоянии логического нуля. Контроллер выполняет одиночные передач, в соответствии с протоколом AHB-Lite, используя сигналы комбинации шины HTRANS, перечисленные в таблице ниже.

Таблица 386 Комбинации шины HTRANS

HTRANS[1]	HTRANS[0]*	Значение
0	0	IDLE (нет передачи)
1	0	NONSEQ (одиночная передача)

* - сигнал постоянно удерживается в состоянии «логический ноль».

Отсутствие возможности осуществлять пакетные передачи имеет минимальное влияние на производительность системы, так как пакетные передачи более эффективны в

одноуровневых системах с шиной AHB, где блоки должны «захватывать» шину или обращаться к внешней памяти. В тоже время контроллер ПДП предназначен для использования в многоуровневых системах с шиной AHB-Lite, включающих встроенную память.

Разрядность передач данных

Контроллер интерфейса предоставляет возможность осуществлять передачу 8, 16 и 32 разрядных данных. Таблица ниже перечисляет значение комбинаций шины HSIZE.

Таблица 387 Комбинации шины HSIZE

HSIZE[2]*	HSIZE[1]	HSIZE[0]	Разрядность данных (бит)
0	0	0	8
0	0	1	16
	1	0	32
	1	1	**

* - сигнал постоянно удерживается в состоянии «логический ноль».

**- запрещенная комбинация.

Контроллер всегда использует передачи 32-разрядными данными при обращении к управляющим данным канала. Необходимо устанавливать разрядность данных источника соответствующую разрядности данных приемника.

Управление защитой данных

Контроллер позволяет устанавливать режимы защиты данных протокола AHB-Lite, определяемые шиной HPROT[3:1]. Возможен выбор следующих режимов защиты:

- кэширование;
- буферизация;
- привилегированный.

Таблица ниже перечисляет значения комбинаций шины HPROT.

Таблица 388 Режимы защиты данных

HPROT[3] Кэширование	HPROT[2] Буферизация	HPROT[1] Привилегиро-ванный	HPROT[0] Данные/команда	Описание
-	-	-	1*	Доступ к данным
-	-	0	-	Пользовательский доступ
-	-	1	-	Привилегированный доступ
-	0	-	-	Без буферизации
-	1	-	-	Буферизованный
0	-	-	-	Без кэширования
1	-	-	-	Кэшированный

Контроллер удерживает HPROT[0] в состоянии логической единицы, чтобы обозначить доступ к данным.

Для каждого цикла ПДП возможен выбор режимов защиты данных передач источника и приемника. Более подробно это описано в разделе Настройка управляющих данных.

Для каждого канала ПДП также возможен выбор режима защиты данных. Более подробно это описано в разделе Управление ПДП.

Инкремент адреса.

Контроллер позволяет управлять инкрементом адреса при чтении данных из источника и при записи данных в приемник. Инкремент адреса производится в зависимости от разрядности передаваемых данных. В таблице ниже перечислены возможные комбинации.

Таблица 389 Инкремент адреса

Разрядность данных	**Величина инкремента**
8	Байт, полуслово, слово
16	Полуслово, слово
32	Слово

Минимальная величина инкремента адреса всегда соответствует разрядности передаваемых данных. Максимальная величина инкремента адреса, осуществляемая контроллером - одно слово. Более подробно о настройке инкремента адреса написано в разделе Настройка управляющих данных. Этот раздел описывает разряды управления величиной инкремента адреса в управляющих данных канала.

Примечание.

Если необходимо оставлять адрес неизменным при чтении или записи данных, для примера, при работе с FIFO, можно соответствующим образом настроить контроллер на работу с фиксированным адресом (см. раздел Настройка управляющих данных).

Управление ПДП

Раздел описывает:

- правила обмена данными;
- диаграммы работы контроллера ПДП;
- правила арбитража ПДП;
- приоритет;
- типы циклов ПДП;
- индикация ошибок.

Правила обмена данными.

Контроллер использует правила обмена данными перечисленные в Таблица 390 при соблюдении следующих условий:

- канал ПДП включен, что выполняется установкой в состояние логической единицы разрядов управления chnl_enable_set[C] и master_enable;
- флаги запроса dma_req[C] и dma_sreq[C] не замаскированы, что выполняется установкой в состояние логического нуля разряда управления chnl_req_mask_set [C];
- контроллер находится не в тестовом режиме, что выполняется установкой в состояние логического нуля разряда управления int_test_en bit[C].

Таблица 390 Правила, при которых передача данных по каналам разрешена

и запросы не маскируются

Правило	Описание
1	Если dma_active[C] установлен в 0, то установка в 1 dma_req[C] или dma_sreq[C] на один или более тактов сигнала hclk, следующих или не следующих друг за другом, инициирует передачу по каналу номер C.
2	Контроллер осуществляет установку в 1 только одного разряда dma_active[C].
3	Контроллер устанавливает в 1 dma_active[C] в момент начала передачи по каналу C.
4	Для типов циклов ПДП отличных от периферийного «Исполнение с изменением конфигурации», dma_active[C] остается в состоянии 1 до тех пор, пока контроллер не окончит передачи с номерами меньше, чем значение 2R или число передач указанное в регистре n_minus_1. В периферийном режиме «Исполнение с изменением конфигурации», dma_active[C] остается в состоянии 1 в течение каждой пары ПДП передач, с использованием первичной и альтернативной структур управляющих данных. Так, что контроллер выполняет 2R передач, используя первичную структуру управляющих данных, затем без осуществления арбитража выполняет передачи с номерами меньше, чем значение 2R (или число передач указанное в регистре n_minus_1) передач, используя альтернативную структуру управляющих данных. По окончании последней передачи dma_active[C] сбрасывается в 0.
5	Контроллер устанавливает dma_active[C] в 0 на, как минимум, один такт сигнала hclk, перед тем как снова установит dma_active[C] или dma_active[ ] в 1.
6	Для каналов, по которым разрешена передача, контроллер осуществляет установку в 1 только одного dma_done[ ].
7	Если dma_req[C] устанавливается в состояние 1 в момент, когда dma_active[C] или dma_stall также в состоянии 1, то это означает, что контроллер обнаружил запрос.
8	Если разряды cycle_ctrl для канала установлены в состояние 3’b100, 3’b101, 3’b110, 3’b111, то dma_done[C] никогда не будет установлен в 1.
9	Если все передачи по каналу завершены, и разряды cycle_ctrl позволяют удержание dma_done[C], то по срезу сигнала dma_active[ ] произойдут события: - если dma_stall в состоянии 0, контроллер устанавливает dma_done[ ] в состояние 1 продолжительностью один такт hclk; - если dma_stall в состоянии 1 работа контроллера приостановлена. После того, как dma_stall будет установлен в 0, контроллер устанавливает dma_done[ ] в состояние 1 продолжительностью один такт hclk.
10	Состояние dma_waitonreq[C] можно изменять только при выключенном канале.
11	Если dma_waitonreq[C] в состоянии 1, то сигнал dma_active[C] не перейдет в состояние 0 до тех пор, пока: - контроллер завершит 2R передач (или число передач указанное в регистре n_minus_1); - dma_req[C] будет установлен в 0; - dma_sreq[C] будет установлен в 0.
12	Если за один такт сигнала hclk перед установкой dma_active[C] в 0, устанавливается в 1 dma_stall, то: - контроллер установит dma_active[C] в 0 на следующем такте сигнала hclk;

	- передача по каналу C не завершится пока не будет сброшен в 0 dma_stall.
13	Контроллер игнорирует dma_sreq[C] если dma_waitonreq[C] в состоянии 0.
14	Контроллер игнорирует dma_sreq[C] если chnl_useburst_set[C] в состоянии 1*.
15	Для типов циклов ПДП, отличных от периферийного «Исполнение с изменением конфигурации» по окончании 2R передач, контроллер устанавливает значение chnl_useburst_set[C] в состояние 0, если количество оставшихся передач меньше, чем 2R. В периферийном режиме «Исполнение с изменением конфигурации», контроллер устанавливает значение chnl_useburst_set[C] в состояние 0 только, если количество оставшихся передач с использованием альтернативной структуры управляющих данных меньше, чем 2R.
16	Для типов циклов ПДП, отличных от периферийного «Исполнение с изменением конфигурации», если за один такт hclk до установки dma_active[C] в 1 dma_sreq[C] и dma_waitonreq[C] установлены в 1 и dma_req[C] установлен в 0, то контроллер выполняет одну ПДП передачу. В периферийном режиме «Исполнение с изменением конфигурации», если за один такт hclk до установки dma_active[C] в 1 dma_sreq[C] и dma_waitonreq[C] установлены в 1 и dma_req[C] установлен в 0, то контроллер выполняет 2R передач с использованием первичной структуры управляющих данных, затем без осуществления арбитража выполняет одну передачу, используя альтернативную структуру управляющих данных.
17	Для типов циклов ПДП отличных от периферийного «Исполнение с изменением конфигурации», если за один такт hclk до установки dma_active[C] в 1, dma_sreq[C] и dma_req[C] установлены в 1, то приоритет предоставляется dma_req[c] и контроллер выполняет 2R (или число передач указанное в регистре n_minus_1) ПДП передач. В периферийном режиме «Исполнение с изменением конфигурации», если за один такт hclk до установки dma_active[C] в 1, dma_sreq[C] и dma_req[C] установлены в 1, то приоритет предоставляется dma_req[c] и контроллер выполняет 2R передач с использованием первичной структуры управляющих данных, затем без осуществления арбитража выполняет передачи с номерами меньше, чем значение 2R (или число передач указанное в регистре n_minus_1), используя альтернативную структуру управляющих данных.
18	Когда chnl_req_mask_set[C] установлен в 1, контроллер игнорирует запросы по dma_sreq[C] и dma_req[C].

* - Необходимо с осторожностью устанавливать эти разряды. Если значение, указанное в регистре n_minus_1, меньше, чем значение 2R, то контроллер не очистит разряды chnl_useburst_set и поэтому запросы по dma_sreq[C] будут маскированы. Если периферия не устанавливает dma_req[C] в состояние 1, то контроллер никогда не выполнит необходимых передач.

При отключении канала контролер осуществляет ПДП передачи, согласно правилам, представленным в таблице ниже.

Таблица 391 Правила осуществления ПДП передач при «запрещенных» каналах

Правило	**Описание**
19	Если dma_req[C] установлен в 1, то контроллер устанавливает dma_done[C] в 1. Это позволяет контроллеру показать центральному процессору запрос готовности, даже если канал выключен (запрещен).

20	Если dma_sreq[C] установлен в 1, то контроллер устанавливает dma_done[C] в 1 одновременно удерживая dma_waitonreq[C] в 1 и chnl_useburst_set[C] в состоянии 0. Это позволяет контроллеру показать центральному процессору запрос готовности, даже если канал выключен (запрещен).
21	dma_active[C] всегда удерживается в состоянии 0.

Диаграммы работы контроллера ПДП

Данный раздел описывает следующие примеры функционирования контроллера с использованием правил обмена данными представленными в Таблица 390:

- импульсный запрос на обработку;
- запрос по уровню на обработку;
- флаги завершения;
- флаги ожидания запроса на обработку.

Примечание:

Все диаграммы, показанные на Рисунок 100 -Рисунок 104 подразумевают следующее:

- hready находится в состоянии 1;
- AHB «ведомый» всегда дает ответ «OKAY».

Импульсный запрос на обработку

Рисунок 100 показывает временную диаграмму работы контроллера ПДП при получении импульсного запроса от периферии.

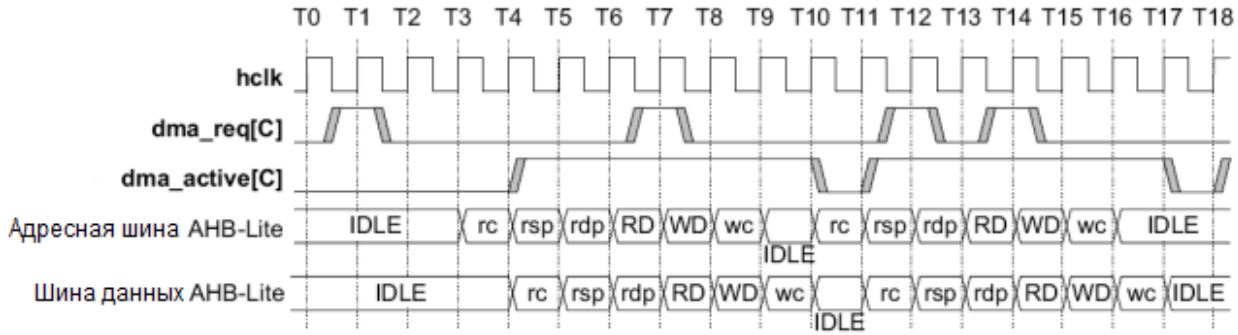


Рисунок 100. Диаграмма работы при получении импульсного запроса от периферийного блока

Запрос на обработку по уровню

Рисунок 101 показывает временную диаграмму работы контроллера ПДП при получении от периферии запроса на обработку по уровню.

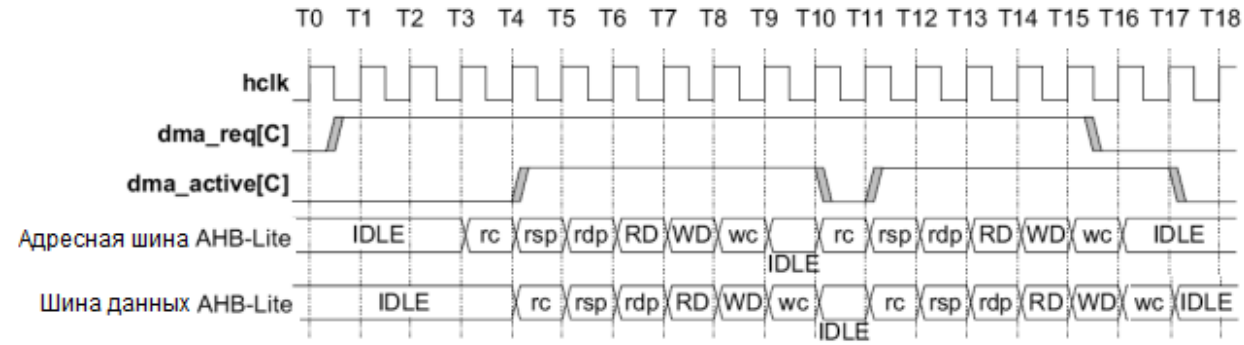


Рисунок 101. Диаграмма работы при получении от периферийного блока запроса на обработку по уровню.

Пояснения к Рисунок 101.

T1	Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18).
T4	Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает ПДП передачи по каналу C.
T4-T7	Контроллер считывает управляющую данные канала, где: rc – чтение настроек канала, channel_cfg; rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr; rdp - чтение указателя адреса окончания данных приемника, dst_data_end_ptr.
T7-T9	Контроллер выполняет передачу ПДП по каналу C, где: RD – чтение данных; WD – запись данных.
T9-T10	Контроллер осуществляет запись настроек канала, channel_cfg, где wc – запись настроек канала, channel_cfg.
T10	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи ПДП (см. правило 4). Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18).
T10-T11	Контроллер удерживает dma_active[C] на, как минимум, один такт hclk (см. правило 5).
T11	Если канал C имеет более высокий приоритет, то контроллер устанавливает dma_active[C] и начинает вторую ПДП передачу по каналу C.
T11-T14	Контроллер считывает управляющую данные канала.
T14-T16	Контроллер выполняет передачу ПДП по каналу C.
T15-T16	Периферийный блок обнаруживает, что передача ПДП началась и сбрасывает dma_req[C].
T16-T17	Контроллер осуществляет запись настроек канала, channel_cfg.
T17	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи ПДП (см. правило 4).

При использовании запроса на обработку по уровню, периферийный блок может не обладать достаточным быстродействием, чтобы во время снять сигнал запроса, в этом случае он должен установить сигнал dma_stall. Установка сигнала dma_stall предотвращает повторение выполненной передачи.

Флаги завершения

Рисунок 102 демонстрирует функционирование сигнала (флага) dma_done[] при следующих условиях:

- dma_stall и dma_waitonreq[] находятся в состоянии 0;
- dma_stall установлен в 1;
- dma_waitonreq[]установлен в 1.

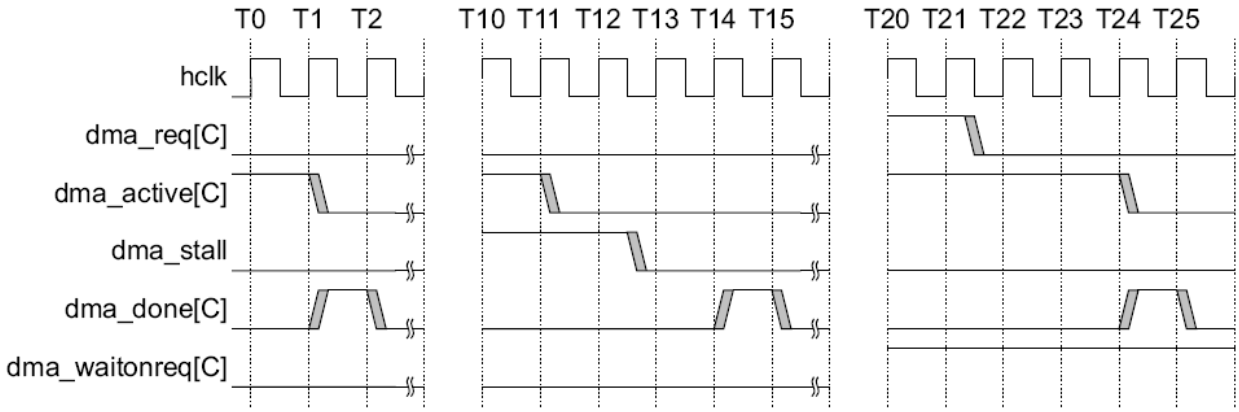


Рисунок 102. Диаграммы функционирования dma_done

Пояснения к Рисунок 102 от T0 до T2

T1 Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи ПДП (см. правило 4).

T1-T2 Контроллер завершает цикл ПДП и если cycle_ctrl[2] установлен в 0, то устанавливает в 1 dma_done[C] на один такт hclk (см. правила 8 и 9). Для других разрешенных каналов сигнал dma_done[C] останется в состоянии 0 (см. правило 6).

Пояснения к Рисунок 102 от T10 до T15

T11 Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи ПДП (см. правило 4).

Примечание:

Контроллер не устанавливает сигнал dma_done[C], так как сигнал dma_stall установлен в 1 в предшествующем такте hclk (см. правила 9 и 12).

T12-T13 Периферийный блок сбрасывает сигнал dma_stall.

T14-T15 Контроллер завершает цикл ПДП и если cycle_ctrl[2] установлен в 0, то устанавливает в 1 dma_done[C] на один такт hclk (см. правила 8 и 9). Для других разрешенных каналов сигнал dma_done[C] останется в состоянии 0 (см. правило 6).

Пояснения к Рисунок 102 от T20 до T25

T20 Контроллер выполнил передачу ПДП, но из-за установленного в 1 dma_waitonreq[C] он должен ожидать сброса в 0 сигнала dma_req[C], перед тем как сбросить dma_active[C] (см. правило 11) и установить dma_done[C] (см. правило 9).

T21-T25 Периферийный блок сбрасывает dma_req[C].

T24 Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи ПДП (см. правило 4).

T24-T25 Контроллер завершает цикл ПДП и если cycle_ctrl[2] установлен в 0, то устанавливает в 1 dma_done[C] на один такт hclk (см. правила 8 и 9). Для других разрешенных каналов сигнал dma_done[C] останется в состоянии 0 (см. правило 6).

Флаги ожидания запроса на обработку

Нижеприведенные рисунки демонстрируют примеры использования флагов ожидания запроса на обработку при выполнении 2R передач и одиночных передач:

- диаграмма работы контроллера ПДП при использовании периферией dma_waitonreq;
- диаграмма работы контроллера ПДП при использовании периферией dma_waitonreq совместно с dma_sreq.
-

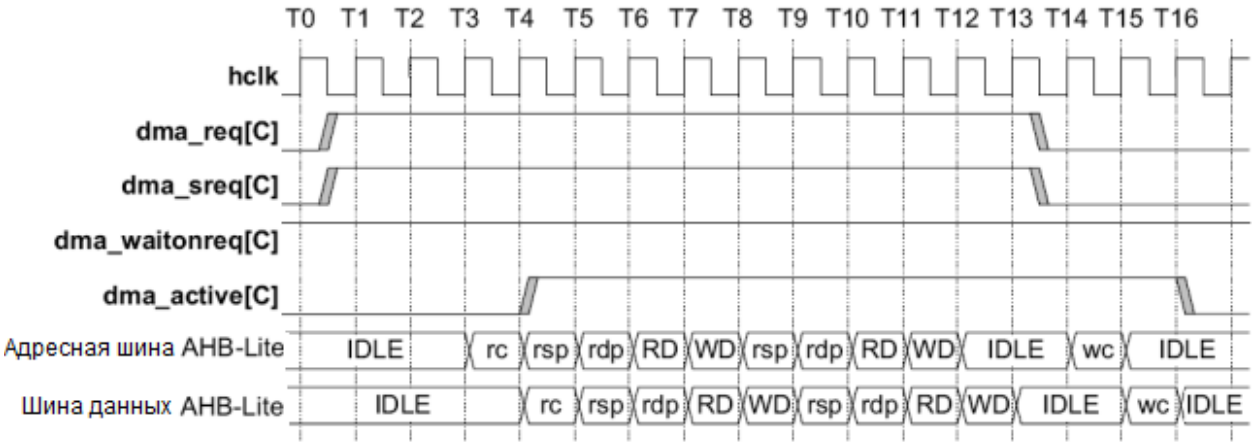


Рисунок 103. Диаграмма работы контроллера ПДП при использовании периферией dma_waitonreq

Пояснения к Рисунок 103.

T0-T16 Периферийный блок должен оставлять состояние dma_waitonreq[C] постоянно (см. правило 10).

T0-T1 Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18).

T3-T4 Периферийный блок удерживает dma_req[C] и dma_sreq[C] в 1. Контроллер игнорирует dma_sreq[C] запрос и отвечает на dma_req[C] запрос (см. правила 16 и 17).

T4 Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает ПДП передачи по каналу C.

T4-T7 Контроллер считывает управляющую данные канала, где:
rc – чтение настроек канала, channel_cfg;

rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr;
rdp - чтение указателя адреса окончания данных приемника, dst_data_end_ptr.

T7-T9 Контроллер выполняет передачу ПДП по каналу C, где:
RD – чтение данных;
WD – запись данных.

T9_T11 Контроллер считывает 2 указателя адреса окончания данных rsp и rdp.

T11-T13 Периферийный блок сбрасывает сигналы dma_req[C] и dma_sreq[C].

T15_T16 Контроллер осуществляет запись настроек канала, channel_cfg, где
wc – запись настроек канала, channel_cfg.

T16 Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи ПДП (см. правило 11).
Контроллер устанавливает значение по чтению регистра chnl_useburst_set[C] в 0, если количество оставшихся передач менее 2R (см. правило 15).

Рисунок 104 показывает работу контроллера ПДП при установке dma_waitonreq в 1 и выполнении одиночной ПДП передачи.

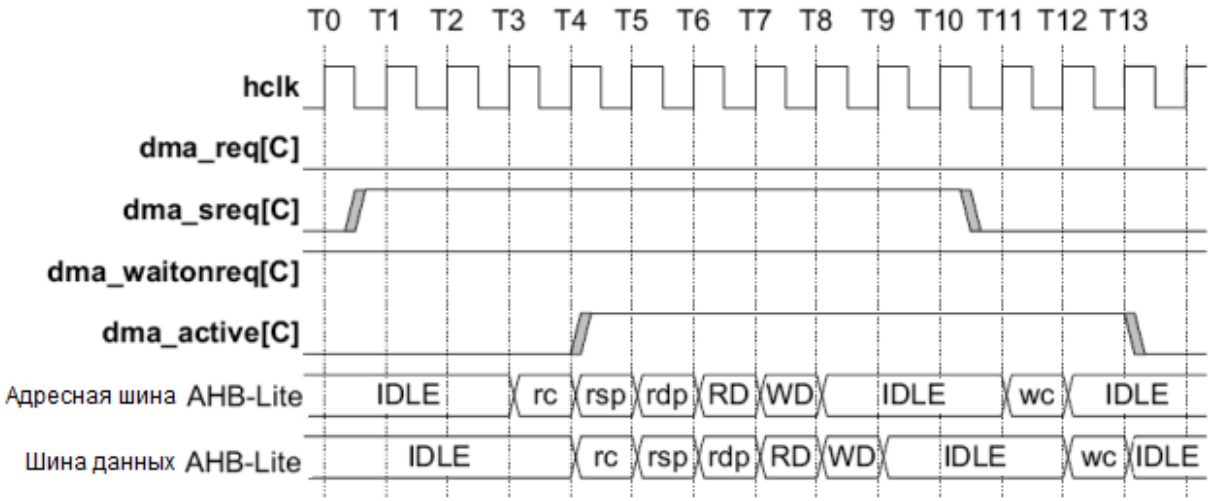


Рисунок 104. Диаграмма работы контроллера ПДП при использовании периферией dma_waitonreq совместно с dma_sreq

Пояснения к Рисунок 104.

T0-T13 Периферийный блок должен оставлять состояние dma_waitonreq[C] постоянно (см. правило 10).

T0-T1 Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_useburst_set[C] находится в состоянии 0 (см. правила 13 и 14).

T3-T4
Контроллер отвечает на dma_sreq[C] запрос (см. правила 16).

T4
Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает ПДП

передачи по каналу C.

T4-T7

Контроллер считывает управляющую данные канала, где:
rc – чтение настроек канала, channel_cfg;
rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr;
rdp - чтение указателя адреса окончания данных приемника, dst_data_end_ptr.

T7-T9

Контроллер выполняет передачу ПДП по каналу C, где:
RD – чтение данных;
WD – запись данных.
Это запрос в ответ на dma_sreq[], таким образом R=0 и следовательно контроллер исполнит 1 ПДП передачу.

T10-T11

Периферийный блок сбрасывает сигнал dma_sreq[C].

T12_T13

Контроллер осуществляет запись настроек канала, channel_cfg, где
wc – запись настроек канала, channel_cfg.

T13

Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи ПДП (см. правило 11).

Правила арбитража ПДП

Контроллер имеет возможность настройки момента арбитража при передачах ПДП. Эта возможность позволяет уменьшить время отклика при обслуживании каналов с высоким приоритетом.

Контроллер имеет 4 разряда, которые определяют количество транзакций по шине AHB до повторения арбитража. Эти разряды, так называемая степень R числа 2, изменение R напрямую устанавливает периодичность арбитража. Для примера, если R равно 4, то арбитраж будет проводиться через каждые 16 передач ПДП.
Таблица 392 показывает возможную периодичность арбитража.

Таблица 392 Периодичность арбитража в единицах передач по шине AHB

Значение R	**Периодичность арбитража каждые x передач ПДП**
b0000	1
b0001	2
b0010	4
b0011	8
b0100	16
b0101	32
b0110	64
b0111	128
b1000	256
b1001	512
b1010-b1111	1024

Примечание:

Необходимо с осторожностью устанавливать большие значения R для низкоприоритетных каналов, так как это может привести к невозможности обслуживать запросы по высокоприоритетным каналам.

При N > 2R (N- номер передачи) и если результат деления 2R на N не целое число, то контроллер всегда выполняет последовательность из 2R передач до тех пор, пока не станет верным N<2R. Контроллер выполняет оставшиеся N передач в конце цикла ПДП. Разряды степени R числа 2 находятся в структуре управляющих данных канала. Местонахождение этих разрядов описано в разделе «Управляющие данные канала».

Приоритет.

При проведении арбитража, определяется канал для обслуживания в следующем цикле ПДП. Определение следующего канала происходит по следующим признакам:

- номер канала;
- уровень приоритета, присвоенного каналу.

Каждому каналу может быть присвоен уровень приоритета по умолчанию (низкий) или высокий уровень приоритета. Присвоение уровня приоритета осуществляется установкой или сбросом разряда chnl_priority_set.

Канал номер 0 имеет высший уровень приоритета и уровень приоритета снижается с увеличением номера канала. Таблица 393 показывает уровень приоритета каналов ПДП в порядке его уменьшения.

Таблица 393 Уровень приоритета каналов ПДП

Номер канала	**Установка уровня приоритета**	**Уровень приоритета в порядке его уменьшения**
0	Высокий	Наивысший уровень приоритета
1	Высокий	-
2	Высокий	-
-	Высокий	-
-	Высокий	-
-	Высокий	-
30	Высокий	-
31	Высокий	-
0	По умолчанию (низкий)	-
1	По умолчанию (низкий)	-
2	По умолчанию (низкий)	-
-	По умолчанию (низкий)	-
-	По умолчанию (низкий)	-
-	По умолчанию (низкий)	-
30	По умолчанию (низкий)	-
31	По умолчанию (низкий)	Низший уровень приоритета

После окончания цикла ПДП, контроллер выбирает следующий для обслуживания канал из всех включенных каналов ПДП. Рисунок 105 показывает процесс выбора следующего канала для обслуживания.

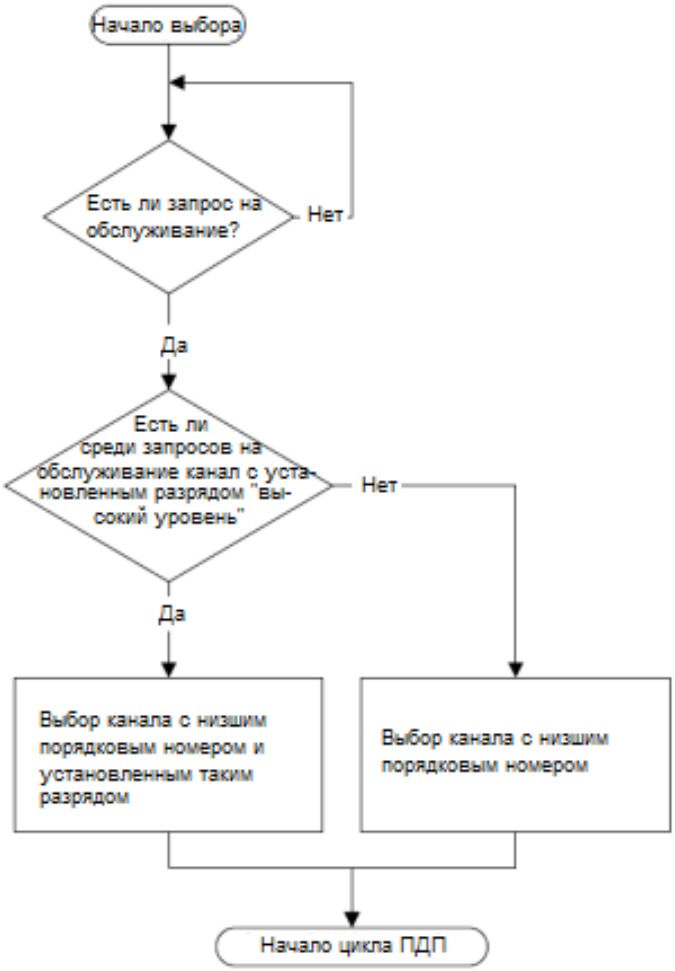


Рисунок 105. Алгоритм выбора следующего канала для обслуживания.

Начало выбора.

Есть ли запрос на обслуживание.

Есть ли среди запросов на обслуживание канал с установленным разрядом «высокий уровень».

Выбор канала с низшим порядковым номером и установленным таким разрядом.

Выбор канала с низшим порядковым номером.

Начало цикла ПДП.

Типы циклов ПДП.

Разряды cycle_ctrl определяют, как контроллер будет выполнять циклы ПДП. Описание значений этих разрядов приведено в Таблица 394.

Таблица 394 Типы циклов ПДП

cycle_ctrl	**Описание**
b000	Структура управляющих данных канала в запрещенном состоянии
b001	Обычный цикл ПДП
b010	Авто-запрос
b011	Режим пинг-понг
b100	Работа с памятью в режиме «Исполнение с изменением конфигурации» с использованием первичных управляющих данных канала

b101	Работа с памятью в режиме «Исполнение с изменением конфигурации» с использованием альтернативных управляющих данных канала
b110	Работа с периферией в режиме «Исполнение с изменением конфигурации» с использованием первичных управляющих данных канала
b111	Работа с периферией в режиме «Исполнение с изменением конфигурации» с использованием альтернативных управляющих данных канала

Примечание:

Разряды cycle_ctrl находятся в области памяти отведенной под channel_cfg, описанной в разделе Настройка управляющих данных канала.

Для всех типов циклов ПДП повторный арбитраж происходит после 2R передач ПДП. Если установить длинный период арбитража на низко приоритетном канале, то это заблокирует все запросы на обработку от других каналов до тех пор, пока не будут выполнены 2R передач ПДП по данному каналу. Поэтому устанавливая значение R необходимо учитывать, что это может привести к повышенному времени отклика на запрос на обработку от высокоприоритетных каналов.

Данный раздел описывает следующие типы циклов ПДП:

- недействительный;
- основной;
- авто-запрос;
- пинг-понг;
- работа с памятью «исполнение с изменением конфигурации»;
- работа с периферией «исполнение с изменением конфигурации».

Недействительный.

После окончания цикла ПДП контроллер устанавливает тип цикла в значение «недействительный», для предотвращения повтора выполненного цикла ПДП.

Основной.

В этом режиме контроллер работает только с основными или альтернативными управляющими данными канала. После того, как разрешена работа канала и контроллер получил запрос на обработку, цикл ПДП выглядит следующим образом:

1. Контроллер выполняет 2R передач. Если число оставшихся передач 0, контроллер переходит к шагу 3.

2. Осуществление арбитража:

- если высокоприоритетный канал выдает заброс на обработку, то контроллер начинает обслуживание этого канала;
- если периферийный блок или программное обеспечение выдает заброс на обработку (повторный запрос на обработку по каналу), то контроллер переходит к выполнению шагу 1.

3. Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала hclk. Это указывает центральному процессору о завершении цикла ПДП.

Авто-запрос.

Функционируя в данном режиме, контроллер ожидает получения одиночного запроса на обработку для разрешения работы и выполнения цикла ПДП. Такая работа позволяет выполнять передачу больших пакетов данных без существенного увеличения времени отклика на обслуживание высокоприоритетных запросов и не требует множественных запросов на обработку от процессора или периферийных блоков.

Контроллер позволяет настроить для использования первичную или альтернативную структуру управляющих данных канала. После того, как разрешена работа канала и контроллер получил запрос на обработку, цикл ПДП выглядит следующим образом:

1. Контроллер выполняет 2R передач для канала C. Если число оставшихся передач 0, контроллер переходит к шагу 3.

2. Осуществление арбитража:

- если высокоприоритетный канал выдает заброс на обработку, то контроллер начинает обслуживание этого канала;
- если периферийный блок или программное обеспечение выдает заброс на обработку (повторный запрос на обработку по каналу), то контроллер переходит к выполнению шагу 1.

3. Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала hclk. Это указывает центральному процессору о завершении цикла ПДП.

Пинг-понг.

В данном режиме работы контроллер выполняет цикл ПДП, используя одну из структур управляющих данных, а затем выполняет еще один цикл ПДП, используя другую структуру управляющих данных. Контроллер выполняет циклы ПДП с переключением структур до тех пор, пока не считает «неправильную» структуру данных и ли пока процессор не запретит работу канала.

Рисунок 106 демонстрирует пример функционирования контроллера в режиме Пинг-понг.

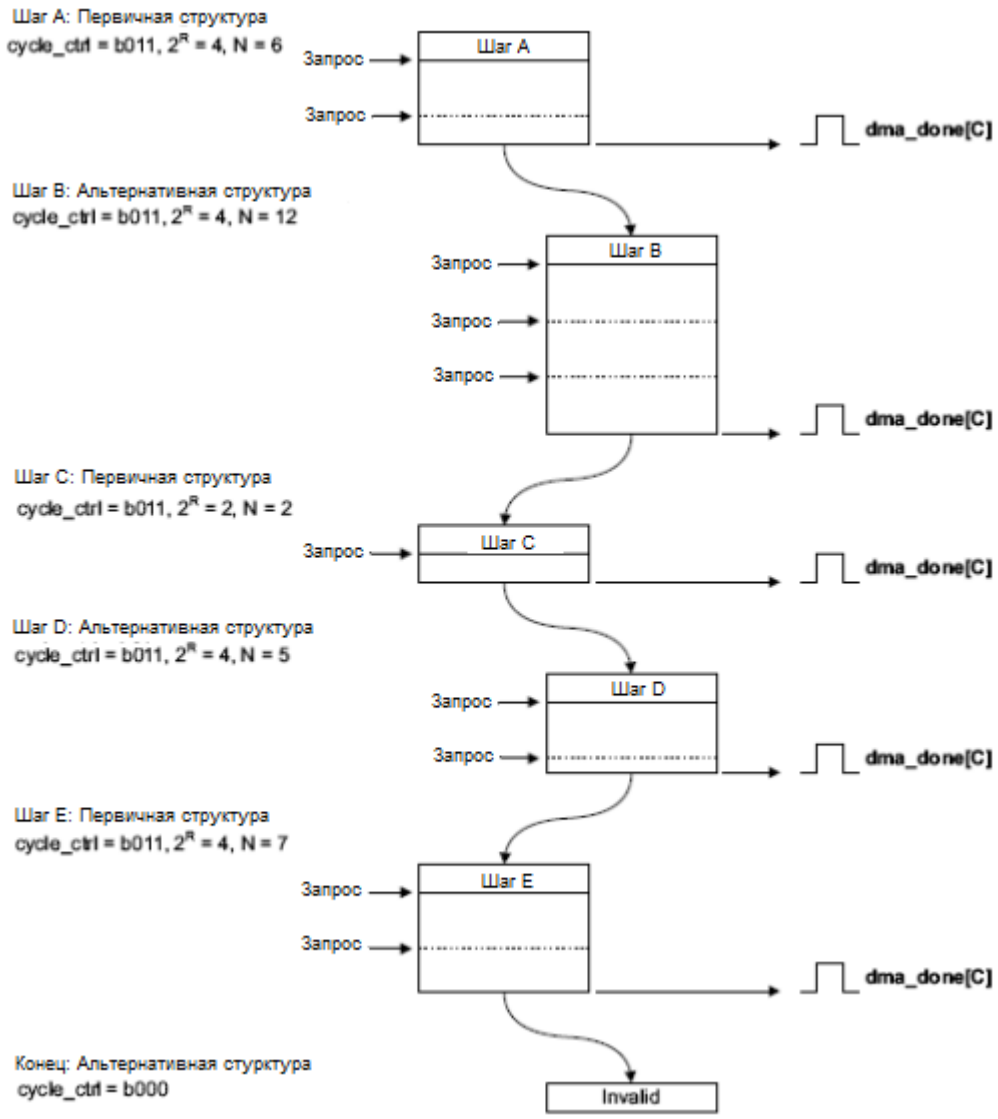


Рисунок 106. Пример работы в режиме Пинг-понг

Перевод рисунка
Шаг A. Первичная структура, cycle_ctrl=b011, 2R = 4, N=6
Шаг B. Альтернативная структура, cycle_ctrl=b011, 2R = 4, N=12
Шаг C. Первичная структура, cycle_ctrl=b011, 2R = 2, N=2
Шаг D. Альтернативная структура, cycle_ctrl=b011, 2R = 4, N=5
Шаг E. Первичная структура, cycle_ctrl=b011, 2R = 4, N=7
Конец. Альтернативная структура, cycle_ctrl=b000

Пояснения к Рисунок 106:

Шаг A	Процессор устанавливает первичную структуру управляющих данных для шага A. Процессор устанавливает альтернативную структуру управляющих данных для шага B. Это позволит контроллеру переключиться к шагу B незамедлительно после выполнения шага A, при условии что контроллер не получит запрос на обработку от высокоприоритетного канала.

Контроллер получает запрос и выполняет 4 передачи ПДП.
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала, контроллер продолжает цикл, в ситуации отсутствия высокоприоритетных запросов.
Контроллер выполняет оставшиеся 2 передачи ПДП.
Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража.

После выполнения шага A процессор может установить первичные управляющие данные канала для шага C. Это позволит контроллеру переключиться к шагу C незамедлительно после выполнения шага B, при условии что контроллер не получит запрос на обработку от высокоприоритетного канала.

После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг B:

Шаг B Контроллер выполняет 4 передачи ПДП.
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала, контроллер продолжает цикл, в ситуации отсутствия высокоприоритетных запросов.
Контроллер выполняет 4 передачи ПДП.
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала, контроллер продолжает цикл, в ситуации отсутствия высокоприоритетных запросов.
Контроллер выполняет оставшиеся 4 передачи ПДП.
Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража.

После выполнения шага B процессор может установить альтернативные управляющие данные канала для шага D.

После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг C:

Шаг C Контроллер выполняет 2 передачи ПДП.
Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража.

После выполнения шага C процессор может установить первичные управляющие данные канала для шага E.

После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг D:

Шаг D Контроллер выполняет 4 передачи ПДП.
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала, контроллер продолжает цикл, в ситуации отсутствия высокоприоритетных запросов.
Контроллер выполняет оставшуюся передачу ПДП.
Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража.

После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг E:

Шаг B Контроллер выполняет 4 передачи ПДП.
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала, контроллер продолжает цикл, в ситуации отсутствия высокоприоритетных запросов.
Контроллер выполняет оставшиеся 3 передачи ПДП.
Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража.

Если контроллер получит новый запрос на обработку от данного канала и он будет самым приоритетным, он предпримет попытку выполнения следующего шага. Однако, из-за того, что процессор не установил альтернативные управляющие данные и по окончанию шага D контроллер установил cycle_ctrl в состояние b000, передачи ПДП прекращаются.

Примечание:

Для прерывания цикла ПДП, исполняемого в режиме Пинг-понг, также возможен перевод режима работы контроллера на шаге E в Основной цикл ПДП, путем установки cycle_ctrl в 3'b001.

Режим работы с памятью «исполнение с изменением конфигурации».

В данном режиме контроллер, получая начальный запрос на обработку, выполняет 4 передачи ПДП, используя первичные управляющие данные. По окончании этих передач, контроллер начинает цикл ПДП, используя альтернативные управляющие данные. После чего контроллер выполняет еще 4 передачи ПДП, используя первичные управляющие данные. Контроллер продолжает выполнять циклы ПДА, меняя структуры управляющих данных, пока не произойдет одно из следующих условий:

- процессор переведет контроллер в режим Основной во время цикла с альтернативной структурой;
- контроллер считает «неправильную» структуру управляющих данных.

Примечание:

После исполнения контроллером N передач с использованием первичных управляющих данных, он делает эти управляющие данные «неправильными» путем установки cycle_ctrl в 3'b000.

Контроллер устанавливает флаг dma_done[C] в этом режиме работы только когда передача ПДП заканчивается с использованием основного цикла.

В данном режиме контроллер использует первичные управляющие данные для программирования альтернативных управляющих данных. Таблица 395 перечисляет области памяти channel_cfg, те которые должны быть определены константами, и те, значения которых определяются пользователем.

Таблица 395 Channel_cfg для первичной структуры управляющих данных в режиме работы с памятью «исполнение с изменением конфигурации»

Разряды	Обозначение	Значение	Описание
Области с константными значениями			
31…30	dst_inc	b'10	Контроллер производит инкремент адреса пословно
29…:28	dst_size	b'10	Контроллер осуществляет передачу пословно
27…26	src_inc	b'10	Контроллер производит инкремент адреса пословно
25…24	src_size	b'10	Контроллер осуществляет передачу пословно
17…14	R_power	b'0010	Контроллер выполняет 4 передачи ПДП
3	next_useburst	b'0	Для данного режима этот разряд должен быть равен 0
2…0	cycle_ctrl	b'100	Контролер работает в режиме работы с периферией «исполнение с изменением конфигурации»
Области со значениями определяемыми пользователем			
23…21	dst_prot_ctrl	-	Определяет состояние HPROT при записи данных в приемник
20…18	src_prot_ctrl	-	Определяет состояние HPROT при чтении данных

			из источника
13…4	n_minus_1	N*	Настраивает контроллер на выполнение N передач ПДП, где N кратно 4

* - Так как R_power установлены в состояние 4, необходимо задавать значение N кратное 4. Число равное N/4 это количество раз, которое нужно настраивать альтернативные управляющие данные.

Рисунок 107 демонстрирует пример функционирования в режиме работы с памятью «исполнение с изменением конфигурации».

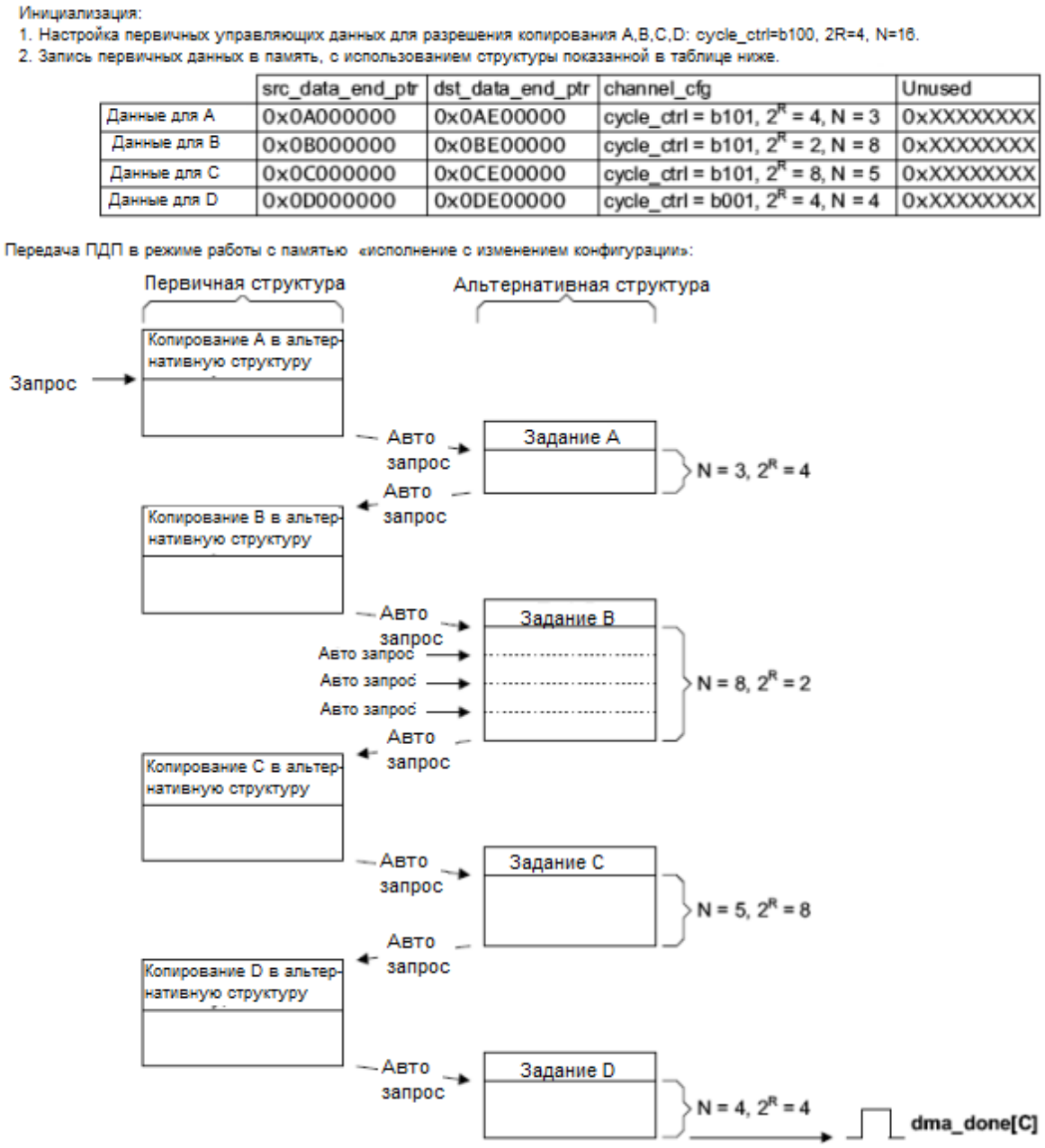


Рисунок 107. Пример функционирования контроллера в режиме работы с памятью «исполнение с изменением конфигурации»

Перевод к рисунку

Инициализация:

1. настройка первичных управляющих данных для разрешения копирования A, B, C, и D: cycle_ctrl=b100, 2R=4, N=16.

2. Запись первичных данных в память, с использованием структуры показанной в таблице ниже.

Передача ПДП в режиме работы с памятью «исполнение с изменением конфигурации».

Пояснения к Рисунок 107.

Инициализация:

1. Процессор настраивает первичную структуру управляющих данных для работы в режиме работы с памятью «исполнение с изменением конфигурации», путем установки cycle_ctrl в b100. Так как управляющие данные канала состоят из 4 слов необходимо установить 2R в 4. В этом примере количество задач равно 4 и, поэтому N установлен в 16.

2. Процессор записывает управляющие данные для задач A, B, C, D в область памяти с адресом указанным в src_data_end_ptr.

3. Процессор разрешает работу канала ПДП.

Передачи в данном режиме начинают исполняться при получении контроллером запроса на обслуживание по dma_req[] или запроса от процессора. Передачи выполняются следующим образом:

Первичная, копирование A.

По получению запроса на обслуживание, контроллер выполняет 4 передачи ПДП. Эти передачи записывают альтернативную структуру управляющих данных для задачи A.

Контроллер генерирует авто-запрос для канала, после чего проводит процедуру арбитража.

Задача A.

Контроллер выполняет задачу A. По окончании, контроллер генерирует авто-запрос для канала и проводит процедуру арбитража.

Первичная, копирование B.

Контроллер выполняет 4 передачи ПДП. Эти передачи записывают альтернативную структуру управляющих данных для задачи B.

Контроллер генерирует авто-запрос для канала, после чего проводит процедуру арбитража.

Задача B.

Контроллер выполняет задачу B. По окончании, контроллер генерирует авто-запрос для канала и проводит процедуру арбитража.

Первичная, копирование C.

Контроллер выполняет 4 передачи ПДП. Эти передачи записывают альтернативную структуру управляющих данных для задачи C.

Контроллер генерирует авто-запрос для канала, после чего проводит процедуру арбитража.

Задача C.

Контроллер выполняет задачу C. По окончании, контроллер генерирует авто-запрос для канала и проводит процедуру арбитража.

Первичная, копирование D.

Контроллер выполняет 4 передачи ПДП. Эти передачи записывают альтернативную структуру управляющих данных для задачи D.

Контроллер устанавливает cycle_ctrl первичных управляющих данных в b000 для индикации о том, что эта структура управляющих данных является «неправильной».

Контроллер генерирует авто-запрос для канала, после чего проводит процедуру арбитража.

Задача D.

Контроллер выполняет задачу D, используя основной цикл ПДП.

Контроллер устанавливает флаг dma_done[C] в состояние 1 на один такт сигнала hclk и входит в процедуру арбитража.

Режим работы с периферией «исполнение с изменением конфигурации».

В данном режиме контроллер, получая начальный запрос на обработку, выполняет 4 передачи ПДП, используя первичные управляющие данные. По окончании этих передач контроллер начинает цикл ПДП, используя альтернативные управляющие данные без осуществления арбитража и не устанавливая сигнал dma_active[C] в 0.

Примечание:

Это единственный случай, при котором контроллер не осуществляет процедуру арбитража после выполнения передачи ПДП, используя первичные управляющие данные.

После того как этот цикл завершился, контроллер выполняет арбитраж и по получении запроса на обслуживание от периферии, имеющего наивысший приоритет, он выполняет еще 4 передачи ПДП, используя первичные управляющие данные. По окончании этих передач контроллер начинает цикл ПДП, используя альтернативные управляющие данные без осуществления арбитража и не устанавливая сигнал dma_active[C] в 0.

Контроллер продолжает выполнять циклы ПДА, меняя структуры управляющих данных, пока не произойдет одно из следующих условий:

- процессор переведет контроллер в режим Основной во время цикла с альтернативной структурой;
- контроллер считает «неправильную» структуру управляющих данных.

Примечание:

После исполнения контроллером N передач с использованием первичных управляющих данных, он делает эти управляющие данные «неправильными» путем установки cycle_ctrl в 3’b000.

Контроллер устанавливает флаг dma_done[C] в этом режиме работы только когда передача ПДП заканчивается с использованием основного цикла.

В данном режиме контроллер использует первичные управляющие данные для программирования альтернативных управляющих данных. Таблица 396 перечисляет области памяти channel_cfg, те которые должны быть определены константами, и те, значения которых определяются пользователем.

Таблица 396 Channel_cfg для первичной структуры управляющих данных в режиме работы с периферией «Исполнение с изменением конфигурации»

Разряды	Обозначение	Значение	Описание
Области с константными значениями			
31…30	dst_inc	b’10	Контроллер производит инкремент адреса пословно
29…28	dst_size	b’10	Контроллер осуществляет передачу пословно
27…26	src_inc	b’10	Контроллер производит инкремент адреса пословно
25…24	src_size	b’10	Контроллер осуществляет передачу пословно
17…14	R_power	b’0010	Контроллер выполняет 4 передачи ПДП
2…0	cycle_ctrl	b’110	Контролер работает в режиме работы с периферией «исполнение с изменением конфигурации»
Области со значениями определяемыми пользователем			

23…21	dst_prot_ctrl	-	Определяет состояние HPROT при записи данных в приемник
20…18	src_prot_ctrl	-	Определяет состояние HPROT при чтении данных из источника
13…4	n_minus_1	N*	Настраивает контроллер на выполнение N передач ПДП, где N кратно 4.
3	next_useburst	-	При установке в 1, контроллер установит chnl_useburst_set[C] в 1 после выполнения передачи с альтернативной структурой.

* - Так как R_power установлены в состояние 4, необходимо задавать значение N, кратное 4. Число равное N/4 это количество раз, которое нужно настраивать альтернативные управляющие данные.

Рисунок 108 демонстрирует пример функционирования в режиме работы с периферией «исполнение с изменением конфигурации».

Инициализация:
1. Настройка первичных управляющих данных для разрешения копирования A, B, C, D cycle_ctrl = b110, 2^R = 4, N = 16.
2. Запись первичных данных в память, с использованием структуры показанной в таблице ниже.

	src_data_end_ptr	dst_data_end_ptr	channel_cfg	Unused
Данные для A	0x0A000000	0x0AE00000	cycle_ctrl = b111, 2^R = 4, N = 3	0xXXXXXXXX
Данные для B	0x0B000000	0x0BE00000	cycle_ctrl = b111, 2^R = 2, N = 8	0xXXXXXXXX
Данные для C	0x0C000000	0x0CE00000	cycle_ctrl = b111, 2^R = 8, N = 5	0xXXXXXXXX
Данные для D	0x0D000000	0x0DE00000	cycle_ctrl = b001, 2^R = 4, N = 4	0xXXXXXXXX

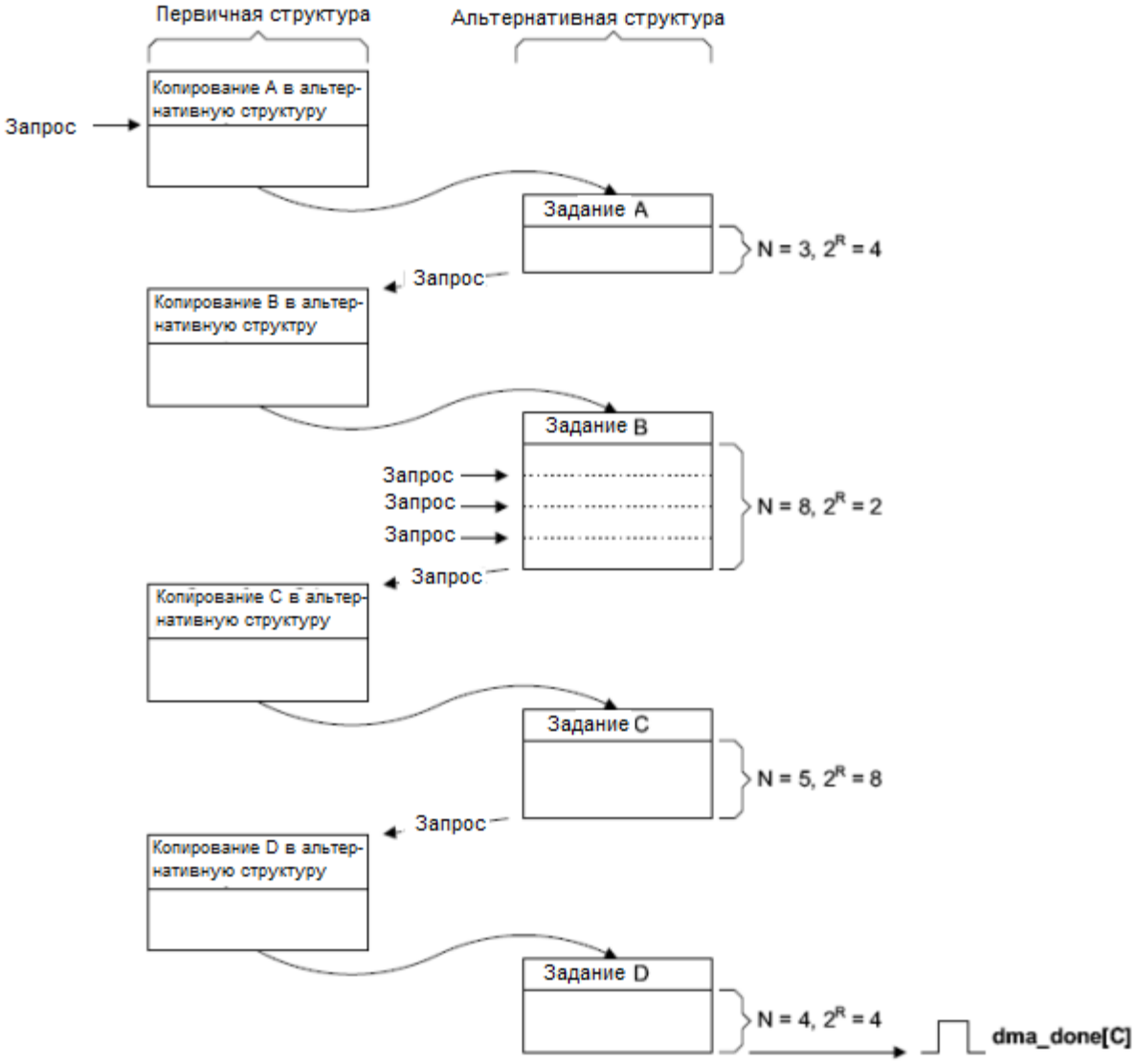


Рисунок 108. Пример функционирования контроллера в режиме работы с периферией «исполнение с изменением конфигурации»

Перевод к рисунку

Инициализация:

1. настройка первичных управляющих данных для разрешения копирования A, B, C, и D: cycle_ctrl=b110, 2R=4, N=16.

2. Запись первичных данных в память, с использованием структуры показанной в таблице ниже.

Передача ПДП в режиме работы с периферией «исполнение с изменением конфигурации».

Пояснения к Рисунок 108.

Инициализация:

1. Процессор настраивает первичную структуру управляющих данных для работы в режиме работы с периферией «исполнение с изменением конфигурации», путем установки cycle_ctrl в b110. Так как управляющие данные канала состоят из 4 слов необходимо установить 2R в 4. В этом примере количество задач равно 4 и, поэтому N установлен в 16.

2. Процессор записывает управляющие данные для задач A, B, C, D в область памяти с адресом указанным в src_data_end_ptr.

3. Процессор разрешает работу канала ПДП.

Передачи в данном режиме начинают исполняться при получении контроллером запроса на обслуживание по dma_req[]. Передачи выполняются следующим образом:

Первичная, копирование A.

По получению запроса на обслуживание, контроллер выполняет 4 передачи ПДП. Эти передачи записывают альтернативную структуру управляющих данных для задачи A.

Задача A.

Контроллер выполняет задачу A.

По окончании, контроллер проводит процедуру арбитража.

Первичная, копирование B.

Контроллер выполняет 4 передачи ПДП. Эти передачи записывают альтернативную структуру управляющих данных для задачи B.

Задача B.

Контроллер выполняет задачу B. Для завершения задачи периферия должна установить последовательно 3 запроса.

По окончании, контроллер проводит процедуру арбитража.

Первичная, копирование C.

Контроллер выполняет 4 передачи ПДП. Эти передачи записывают альтернативную структуру управляющих данных для задачи C.

Задача C.

Контроллер выполняет задачу C.

По окончании, контроллер проводит процедуру арбитража.

После выставления периферией нового запроса на обслуживание при условии, что этот запрос является наиболее приоритетным, процесс продолжается следующим образом:

Первичная, копирование D.

Контроллер выполняет 4 передачи ПДП. Эти передачи записывают альтернативную структуру управляющих данных для задачи D.

Контроллер устанавливает cycle_ctrl первичных управляющих данных в b000 для индикации о том, что эта структура управляющих данных является «неправильной».

Задача D.

Контроллер выполняет задачу D, используя основной цикл ПДП.

Контроллер устанавливает флаг dma_done[C] в состояние 1 на один такт сигнала hclk и входит в процедуру арбитража.

Индикация ошибок.

При получении контроллером по шине AHB-Lite ответа об ошибке, он выполняет следующие действия:

- отключает канал связанный с ошибкой;
- устанавливает флаг dma_err в состояние 1.

После обнаружения процессором флага dma_err, процессор определяет номер канала, который был активен в момент появления ошибки. Для этого он осуществляет следующее:

Чтение регистра chnl_enable_set с целью создания списка отключенных каналов.

Если канал установил флаг dma_done[], то контроллер отключает канал. Программа, выполняемая процессором, должна всегда хранить данные о каналах, которые недавно установили флаги dma_done[].

Процессор должен сравнить список выключенных каналов, полученный в шаге 1, с данными о каналах, которые недавно устанавливали флаги dma_done[]. Канал, по которому отсутствуют данные об установке флага dma_done[], это и есть канал, с которым связана ошибка.

Структура управляющих данных канала

В системной памяти должна быть отведена область для хранения управляющих данных каналов. Системная память должна:

- предоставлять смежную область системной памяти, к которой контроллер и процессор имеют доступ;
- иметь базовый адрес, который целочисленно кратен общему размеру структуры управляющих данных канала.

Рисунок 109 показывает область памяти необходимую контроллеру для структур управляющих данных канала, при использовании всех 32 каналов и опциональной альтернативной структуры управляющих данных.

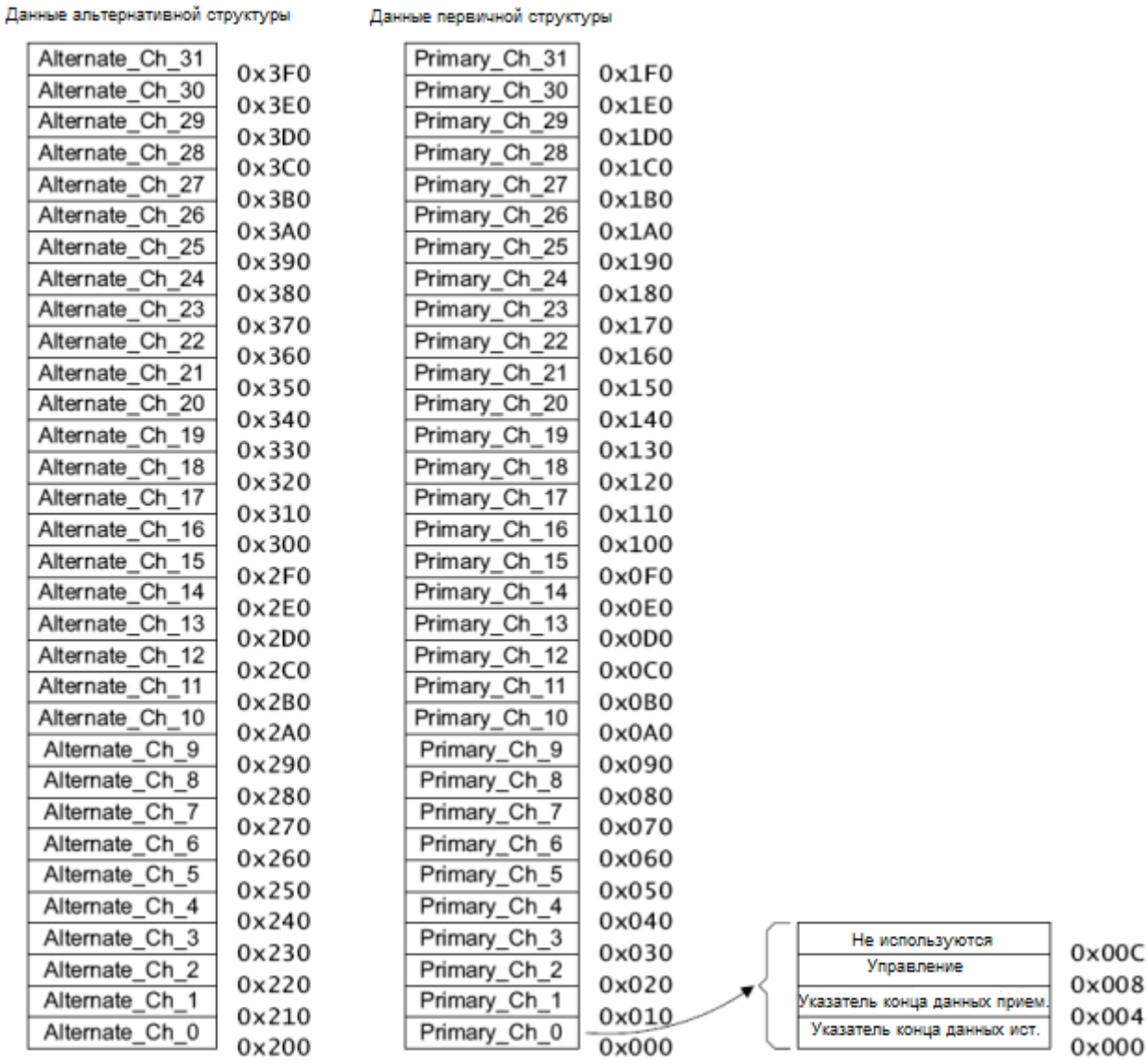


Рисунок 109. Карта памяти для 32-х каналов, включая альтернативную структуру управляющих данных

Пример, показанный на Рисунок 109, использует 1 Кбайт системной памяти. В этом примере контроллер использует младшие 10 разрядов адреса для доступа ко всем элементам структуры управляющих данных, и поэтому базовый адрес структуры должен быть 0xXXXXX000, далее 0xXXXXX400, далее 0xXXXXX800, далее 0xXXXXXC00.

Возможно установить базовый адрес для первичной структуры управляющих данных путем записи соответствующего значения в регистр ctrl_base_ptr.

Необходимый размер области системной памяти зависит от:

- количества каналов используемых в контроллере;
- от того, используется или нет альтернативная структура управляющих данных.

Таблица 397 перечисляет разряды адреса, которые используются контроллером при доступе к различным элементам структуры управляющих данных, в зависимости от количества каналов, используемых в контроллере.

Таблица 397 Разряды адреса соответствующие элементам структуры управляющих данных

Количество каналов, используемых в контроллере	**[9]**	**[8]**	**[7]**	**[6]**	**[5]**	**[4]**	**[3…0]**
1						A	0x0 0x4 0x8
2					A	C[0]	
3-4				A	C[1]	C[0]	
5-8			A	C[2]	C[1]	C[0]	
9-16		A	C[3]	C[2]	C[1]	C[0]	
17-32	A	C[4]	C[3]	C[2]	C[1]	C[0]	

, где

A — Выбирает одну из структур управляющих данных канала
A = 0 выбирает первичную структуру управляющих данных
A = 1 выбирает альтернативную структуру управляющих данных

C[x:0] — Выбирает канал ПДП.

Address[3:0] — Выбирает один из управляющих элементов:
0x0 выбирает указатель конца данных источника
0x4 выбирает указатель конца данных приемника
0x8 выбирает конфигурацию управляющих данных
0xC контроллер не имеет доступа к этому адресу. Если это необходимо, то возможно разрешить процессору использовать эти адреса в качестве системной памяти.

Примечание:

Совсем не обязательно вычислять базовый адрес альтернативной структуры управляющих данных, так как регистр alt_ctrl_base_ptr содержит эту информацию.

Рисунок 110 демонстрирует пример реализации контроллера с использованием 3 каналов ПДП и альтернативной структурой управляющих данных.

Рисунок 110. Карта памяти для трех каналов ПДП, включая альтернативную структуру управляющих данных

Перевод текста рисунка.
Destination end pointer - указатель конца данных приемника
Source end pointer - указатель конца данных источника
Control – управление.

Пример структуры управляющих данных на Рисунок 110 использует 128 байт системной памяти. В этом примере контроллер использует младшие 6 разрядов адреса для доступа ко всем элементам структуры управляющих данных, и поэтому базовый адрес структуры должен быть 0xXXXXXX00, далее 0xXXXXXX80.

Таблица 398 перечисляет все разрешенные значения базового адреса для первичной структуры управляющих данных, в зависимости от количества каналов ПДП, использованных в контроллере.

Таблица 398 Разрешенные базовые адреса

Кол-во каналов ПДП	**Разрешенные значения базового адреса для первичной структуры управляющих данных**
1	0xXXXXXX00, 0xXXXXXX20, 0xXXXXXX40, 0xXXXXXX60, 0xXXXXXX80, 0xXXXXXXA0, 0xXXXXXXC0, 0xXXXXXXE0
2	0xXXXXXX00, 0xXXXXXX40, 0xXXXXXX80, 0xXXXXXXC0
3-4	0xXXXXXX00, 0xXXXXXX80
5-8	0xXXXXX000, 0xXXXXX100, 0xXXXXX200, 0xXXXXX300, 0xXXXXX400, 0xXXXXX500, 0xXXXXX600, 0xXXXXX700, 0xXXXXX800, 0xXXXXX900, 0xXXXXXA00, 0xXXXXXB00, 0xXXXXXC00, 0xXXXXXD00, 0xXXXXXE00, 0xXXXXXF00,
9-16	0xXXXXX000, 0xXXXXX200, 0xXXXXX400, 0xXXXXX600, 0xXXXXX800, 0xXXXXXA00, 0xXXXXXC00, 0xXXXXXE00
17-32	0xXXXXX000, 0xXXXXX400, 0xXXXXX800, 0xXXXXXC00

Контроллер использует системную память для доступа к двум указателям адреса конца данных и разрядам управления каждого канала. Следующие подразделы описывают эти 32-разрядные области памяти и процедуру вычисления контроллером адреса передачи ПДП:

- указатель конца данных источника;
- указатель конца данных приемника;
- разряды управления;
- вычисление адреса.

Указатель конца данных источника.

Область памяти под названием src_data_end_ptr содержит указатель на последний адрес месторасположения данных источника. Таблица 399 перечисляет значения разрядов этой области.

Таблица 399 Значения разрядов src_data_end_ptr

Разряд	**Имя**	**Описание**
31…0	src_data_end_ptr	Указатель на последний адрес данных источника

Перед тем как контроллер выполнит передачу ПДП, необходимо определить эту область памяти. Контроллер считывает значение этой области перед началом 2R передачи ПДП.

Примечание:

Контроллер не имеет доступа по записи в эту область памяти.

Указатель конца данных приемника.

Область памяти под названием dst_data_end_ptr содержит указатель на последний адрес месторасположения данных приемника. Таблица 400 перечисляет значения разрядов этой области.

Таблица 400 Значения разрядов dst_data_end_ptr

Разряд	Имя	Описание
31…0]	dst_data_end_ptr	указатель на последний адрес данных приемника

Перед тем как контроллер выполнит передачу ПДП, необходимо определить эту область памяти. Контроллер считывает значение этой области перед началом 2R передачи ПДП.

Примечание:

Контроллер не имеет доступа по записи в эту область памяти.

Разряды управления.

Область памяти под названием channel_cfg обеспечивает управление каждой передачей ПДП. Рисунок 111 показывает название разрядов этой области.

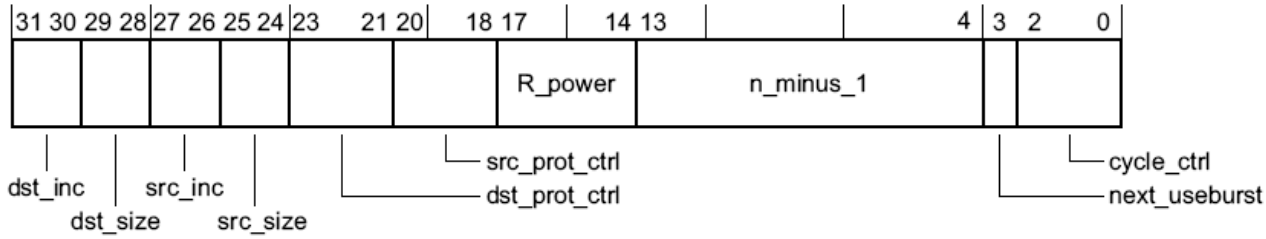


Рисунок 111. Название разрядов channel_cfg

Таблица 401 перечисляет назначение разрядов этой области памяти.

Таблица 401 Назначение разрядов channel_cfg

Разряд	Имя	Описание
31…30	dst_src	Шаг инкремента адреса приемника. Шаг инкремента адреса зависит от разрядности данных источника. Разрядность данных источника = байт b00 = байт b01 = полуслово (в русском обычно слово) b10 = слово (в русском обычно двойное слово) b11 = нет инкремента. Адрес остается равным значению области памяти dst_data_end_ptr Разрядность данных источника = полуслово b00 = зарезервировано b01 = полуслово b10 = слово b11 = нет инкремента. Адрес остается равным значению области памяти dst_data_end_ptr Разрядность данных источника = слово b00 = зарезервировано b01 = зарезервировано b10 = слово b11 = нет инкремента. Адрес остается равным значению области памяти dst_data_end_ptr
29…28	dst_size	Размерность данных приемника. Примечание: Значение этого поля должно быть равно значению поля

		src_size.
27…26	src_inc	Шаг инкремента адреса источника. Шаг инкремента адреса зависит от разрядности данных источника. Разрядность данных источника = байт b00 = байт b01 = полуслово (в русском обычно слово) b10 = слово (в русском обычно двойное слово) b11 = нет инкремента. Адрес остается равным значению области памяти src_data_end_ptr Разрядность данных источника = полуслово b00 = зарезервировано b01 = полуслово b10 = слово b11 = нет инкремента. Адрес остается равным значению области памяти src_data_end_ptr Разрядность данных источника = слово b00 = зарезервировано b01 = зарезервировано b10 = слово b11 = нет инкремента. Адрес остается равным значению области памяти src_data_end_ptr
25…24	src_size	Задает размерность данных источника b00 = байт b01 = полуслово (в русском обычно слово) b10 = слово (в русском обычно двойное слово) b11 = зарезервировано
23…21	dst_prot_ctrl	Задает состояние HPROT[3:1], когда контроллер записывает данные в приемник. Разряд [23] управляет разрядом HPROT[3] 0 = HPROT[3] в состоянии 0 и доступ не кэшируется 1 = HPROT[3] в состоянии 1 и доступ кэшируется Разряд [22] управляет разрядом HPROT[2] 0 = HPROT[2] в состоянии 0 и доступ не буферизуется 1 = HPROT[2] в состоянии 1 и доступ буферизуется Разряд [21] управляет разрядом HPROT[1] 0 = HPROT[1] в состоянии 0 и доступ не привилегированный 1 = HPROT[1] в состоянии 1 и доступ привилегированный
20…18	src_prot_ctrl	Задает состояние HPROT[3:1], когда контроллер считывает данные из источника. Разряд [20] управляет разрядом HPROT[3] 0 = HPROT[3] в состоянии 0 и доступ не кэшируется 1 = HPROT[3] в состоянии 1 и доступ кэшируется Разряд [19] управляет разрядом HPROT[2]

		0 = HPROT[2] в состоянии 0 и доступ не буферизуется 1 = HPROT[2] в состоянии 1 и доступ буферизуется Разряд [18] управляет разрядом HPROT[1] 0 = HPROT[1] в состоянии 0 и доступ не привилегированный 1 = HPROT[1] в состоянии 1 и доступ привилегированный
17…14	R_power	Задает количество передач ПДП до выполнения контроллером процедуры арбитража. Возможные значения: b0000 -- арбитраж производится после каждой передачи ПДП b0001 -- арбитраж производится после 2 передач ПДП b0010 -- арбитраж производится после 4 передач ПДП b0011 -- арбитраж производится после 8 передач ПДП b0100 -- арбитраж производится после 16 передач ПДП b0101 -- арбитраж производится после 32 передач ПДП b0110 -- арбитраж производится после 64 передач ПДП b0111 -- арбитраж производится после 128 передач ПДП b1000 -- арбитраж производится после 256 передач ПДП b1001 -- арбитраж производится после 512 передач ПДП b1010- b1111 -- арбитраж производится после 1024 передач ПДП. Это означает, что арбитраж не производится, так как максимальное количество передач ПДП равно 1024.
13…4	n_minus_1	Перед выполнением цикла ПДП, эти разряды указывают общее количество передач ПДП, из которых состоит цикл ПДП. Необходимо установит эти разряды в значение соответствующие размеру желаемого цикла ПДП. 10-разрядное число задает количество передач ПДП минус 1. Возможные значения: b0000000000 = 1 передача ПДП b0000000001 = 1 передача ПДП b0000000010 = 2 передачи ПДП b0000000011 = 3 передачи ПДП b0000000100 = 4 передачи ПДП b0000000101 = 5 передач ПДП . . . b1111111111 = 1024 передачи ПДП Контроллер обновит это поле перед тем, как произвести процесс арбитража. Это позволяет контроллеру хранить количество оставшихся передач ПДП до завершения цикла ПДП.
3	next_useburst	Контролирует, не установлен ли chnl_useburst_set[C] в состояние 1, если контроллер работает в режиме работы с периферией «исполнение с изменением конфигурации» и если контроллер завершает цикл ПДП, используя альтернативные управляющие данные. Примечание: Перед завершением цикла ПДП, использующего

		альтернативные управляющие данные, контроллер устанавливает chnl_useburst_set[C] в состояние 0, если количество оставшихся передач ПДП меньше, чем 2R. Установка next_useburst разряда определяет, будет ли контроллер дополнительно переопределять разряд chnl_useburst_set[C]. Если контроллер выполняет цикл ПДП в режиме работы с периферией «исполнение с изменением конфигурации», то после окончания цикла, использующего альтернативные управляющие данные, происходит следующее в зависимости от состояния next_useburst: 0 – контроллер не изменяет значение chnl_useburst_set[C]. Если chnl_useburst_set[C] установлен в 0, то для всех оставшихся циклов ПДП в режиме работы с периферией «исполнение с изменением конфигурации», контроллер отвечает на запросы по dma_req[] и dma_sreq[], при выполнении циклов ПДП он использует альтернативные управляющие данные. 1 -- контроллер изменяет значение chnl_useburst_set[C] в состояние 1. Поэтому для оставшихся циклов ПДП в режиме работы с периферией «исполнение с изменением конфигурации», контроллер реагирует только на запросы по dma_req[], при выполнении циклов ПДП он использует альтернативные управляющие данные.
2…0	cycle_ctrl	Режим работы при выполнении цикла ПДП. b000 Стоп. Означает, что структура управляющих данных является «неправильной». b001 Основной. Контроллер должен получить новый запрос для окончания цикла ПДП, перед этим он должен выполнить процедуру арбитража. b010 Авто-запрос. Контроллер автоматически осуществляет запрос на обработку по соответствующему каналу в течение процедуры арбитража. Это означает, что начального запроса на обработку достаточно для выполнения цикла ПДП. b011 Пинг-понг. Контроллер выполняет цикл ПДП используя одну из структур управляющих данных. По окончании выполнения цикла ПДП, контроллер выполняет следующий цикл ПДП, используя другую структуру. Контроллер сигнализирует об окончании каждого цикла ПДП, позволяя процессору перенастраивать неактивную структуру данных. Контроллер продолжает выполнять циклы ПДП, до тех пор пока он не прочитает

		«неправильную» структуру данных или пока процессор не изменит cycle_ctrl поле в состояние b001 или b 010. b100 Режим работы с памятью «исполнение с изменением конфигурации». Смотри соответствующий раздел. При работе контроллера в данном режиме значение этого поля в первичной структуре управляющих данных должно быть b100. b101 Режим работы с памятью «исполнение с изменением конфигурации». Смотри соответствующий раздел. При работе контроллера в данном режиме значение этого поля в альтернативной структуре управляющих данных должно быть b101. b110 Режим работы с периферией «исполнение с изменением конфигурации». Смотри соответствующий раздел. При работе контроллера в данном режиме значение этого поля в первичной структуре управляющих данных должно быть b110. b111 Режим работы с периферией «исполнение с изменением конфигурации». Смотри соответствующий раздел. При работе контроллера в данном режиме значение этого поля в альтернативной структуре управляющих данных должно быть b111.

В начале цикла ПДП или 2R передачи ПДП, контроллер считывает значение channel_cfg из системной памяти. После выполнения 2R или N передач, он сохраняет обновленное значение channel_cfg в системную память.

Контроллер не поддерживает значений dst_size отличных от значений src_size. Если контроллер обнаруживает неравные значения этих полей, он использует значение src_size для размерности данных приемника и источника и при ближайшем обновлении поля n_minus_1, он также устанавливает значение поля dst_size равное src_size.

После выполнения контроллером N передач, контроллер устанавливает значение поля cycle_ctrl в b000, делая тем самым channel_cfg данные «неправильными». Это позволяет избежать повторения выполненной передачи ПДП.

Вычисление адреса.

Для вычисления адреса источника передачи ПДП, контроллер выполняет сдвиг влево значения n_minus_1 на количество разрядов соответствующее полю src_inc и затем вычитает получившееся значение от значения указателя адреса конца данных источника. Подобным образом вычисляется адрес передатчика передачи ПДП, контроллер выполняет сдвиг влево значения n_minus_1 на количество разрядов соответствующее полю dst_inc и затем вычитает получившееся значение от значения указателя адреса конца данных приемника.

В зависимости от значения полей src_inc и dst_inc, вычисления адресов приемника и источника выполняются по следующим уравнениям:

src_inc=b00 and dst_inc=b00

- адрес источника = src_data_end_ptr - n_minus_1
- адрес приемника = dst_data_end_ptr - n_minus_1.

src_inc=b01 and dst_inc=b01

- адрес источника = src_data_end_ptr – (n_minus_1<<1)
- адрес приемника = dst_data_end_ptr – (n_minus_1<<1).

src_inc=b01 and dst_inc=b10

- адрес источника = src_data_end_ptr – (n_minus_1<<2)
- адрес приемника = dst_data_end_ptr – (n_minus_1<<2).

src_inc=b11 and dst_inc=b11

- адрес источника = src_data_end_ptr
- адрес приемника = dst_data_end_ptr.

Таблица 402 перечисляет адреса приемника цикла ПДП для 6 слов.

Таблица 402 Цикла ПДП для 6 слов с пословным инкрементом

Начальные значения channel_cfg перед циклом ПДП				
src_size=b10, dst_inc=b10, n_minus_1=b101, cycle_ctrl=1				
ПДП передачи	Указатель конца данных	Счетчик	Отличие*	Адрес
	0x2AC	5	0x14	0x298
	0x2AC	4	0x10	0x29C
	0x2AC	3	0xC	0x2A0
	0x2AC	2	0x8	0x2A4
	0x2AC	1	0x4	0x2A8
	0x2AC	0	0x0	0x2AC
Конечные значения channel_cfg после цикла ПДП				
src_size=b10, dst_inc=b10, n_minus_1=0, cycle_ctrl=0				

* это значение, полученное после сдвига влево значения счетчика на количество разрядов, соответствующее dst_inc.

Таблица 403 перечисляет адреса приемника для передач ПДП 12 байт с использованием «полусловного» инкремента.

Таблица 403 Цикла ПДП для 12 байт с «полусловным» инкрементом

Начальные значения channel_cfg перед циклом ПДП				
src_size=b00, dst_inc=b01, n_minus_1=b1011, cycle_ctrl=1, R_power=b11				
ПДП передачи	Указатель конца данных	Счетчик	Отличие*	Адрес
	0x5E7	11	0x16	0x5D1
	0x5E7	10	0x14	0x5D3
	0x5E7	9	0x12	0x5D5
	0x 5E7	8	0x10	0x5D7
	0x 5E7	7	0xE	0x5D9
	0x5E7	6	0xC	0x5DB
	0x5E7	5	0xA	0x5DD
	0x5E7	4	0x8	0x5DF

Значения channel_cfg после 2R передач ПДП				
src_size=b00, dst_inc=b01, n_minus_1=b011, cycle_ctrl=1, R_power=b11				
ПДП передачи	0x 5E7	3	0x6	0x5E1
	0x 5E7	2	0x4	0x5E3
	0x5E7	1	0x2	0x5E5
	0x5E7	0	0x0	0x5E7
Конечные значения channel_cfg после цикла ПДП				
src_size=b00, dst_inc=b01, n_minus_1=0, cycle_ctrl=0**, R_power=b11				

* это значение, полученное после сдвига влево значения счетчика на количество разрядов соответствующее dst_inc.

** после окончания цикла ПДП, контроллер делает channel_cfg «неправильным» сбрасывая в 0 поле cycle_ctrl.

Описание регистров контроллера DMA

Данная глава описывает регистры контроллера и управление контроллером через них.

Глава содержит следующие разделы:

- о регистровой модели контроллера.
- описание регистров.

Основные положения регистровой модели контроллера:

- нужно избегать адресации при доступе к зарезервированным или неиспользованным адресам, так как это может привести к непредсказуемым результатам;
- необходимо записывать неиспользуемые или зарезервированные разряды регистров нулями и игнорировать значения таких разрядов при считывании, кроме случаев специально описанных в разделе;
- системный сброс или сброс по установке питания сбрасывает все регистры в состояние 0, кроме случаев специально описанных в разделе;
- все регистры поддерживают доступ по чтению и записи, кроме случаев специально описанных в разделе. Доступ по записи обновляет содержание регистра, а доступ по чтению возвращает содержимое регистра.

Таблица 404 Перечень регистров контроллера

Наименование	**Смещение относительно базового адреса**	**Тип**	**Значение по сбросу**	**Описание**
dma_status	0x000	RO	0x-0nn0000*	Статусный регистр ПДП
dma_cfg	0x004	WO	-	Регистр конфигурации ПДП
ctrl_base_ptr	0x008	R/W	0x00000000	Регистр базового адреса управляющих данных каналов
alt_ctrl_base_ptr	0x00C	RO	0x000000nn**	Регистр базового адреса альтернативных управляющих данных каналов
dma_waitonreq_status	0x010	RO	0x00000000	Регистр статуса ожидания запроса на обработку каналов
chnl_sw_request	0x014	WO	-	Регистр программного запроса

				на обработку каналов
chnl_useburst_set	0x018	R/W	0x00000000	Регистр установки пакетного обмена каналов
chnl_useburst_clr	0x01C	WO	-	Регистр сброса пакетного обмена каналов
chnl_req_mask_set	0x020	R/W	0x00000000	Регистр маскирования запросов на обслуживание каналов
chnl_req_mask_clr	0x024	WO	-	Регистр очистки маскирования запросов на обслуживание каналов
chnl_enable_set	0x028	R/W	0x00000000	Регистр установки разрешения каналов
chnl_enable_clr	0x02C	WO	-	Регистр сброса разрешения каналов
chnl_pri_alt_set	0x030	R/W	0x00000000	Регистр установки первичной/альтернативной структуры управляющих данных каналов
chnl_pri_alt_clr	0x034	WO	-	Регистр сброса первичной/альтернативной структуры управляющих данных каналов
chnl_priority_set	0x038	R/W	0x00000000	Регистр установки приоритета каналов
chnl_priority_clr	0x03C	WO	-	Регистр сброса приоритета каналов
-	0x040-0x048		-	зарезервировано
err_clr	0x04C	R/W	0x00000000	Регистр сброса флага ошибки
-	0x050-0xDFC	-		зарезервировано
Регистры для тестирования				
-	0xE00-0xE48	-	-	Смотри раздел «Описание тестовых регистров контроллера»
-	0xE4C-0xFCC	-	-	зарезервировано
Регистры идентификации				
periph_id_4	0xFD0	RO	0x04	Регистр идентификации периферии 4
-	0xFD4-0xFDC	-	-	зарезервировано
periph_id_0	0xFE0	RO	0x30	Регистр идентификации периферии 0
periph_id_1	0xFE4	RO	0xB2	Регистр идентификации периферии 1
periph_id_2	0xFE8	RO	0x-B***	Регистр идентификации периферии 2
periph_id_3	0xFEC	RO	0x00	Регистр идентификации периферии 3
pcell_id_0	0xFF0	RO	0x0D	Регистр идентификации PrimeCell 0
pcell_id_1	0xFF4	RO	0xF0	Регистр идентификации PrimeCell 1

pcell_id_2	0xFF8	RO	0x05	Регистр идентификации PrimeCell 2
pcell_id_3	0xFFC	RO	0xB1	Регистр идентификации PrimeCell 3

* - значение по сбросу зависит от количества каналов ПДП, использованных в контроллере, а также от того, интегрирована ли схема тестирования.

** - значение по сбросу зависит от количества каналов ПДП, использованных в контроллере.

*** - значение зависит от номера версии контроллера.

DMA_STATUS

Статусный регистр ПДП

Данный регистр имеет доступ только на чтение. При чтении регистр возвращает состояние контроллера. Если контроллер находится в состоянии сброса, то чтение регистра запрещено. Рисунок 112 показывает наименование разрядов этого регистра. Таблица 405 перечисляет назначение разрядов регистра.

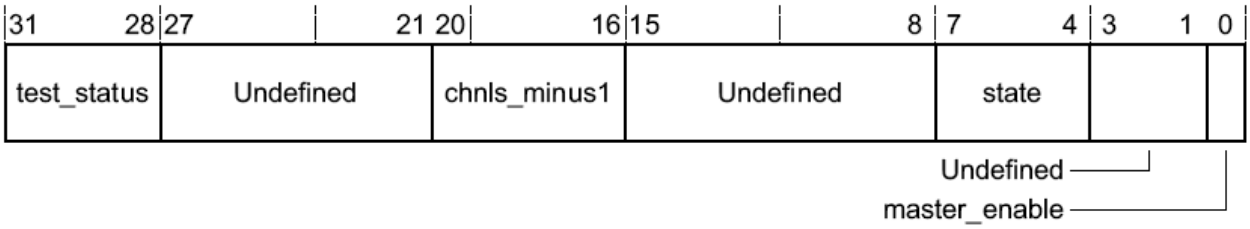


Рисунок 112. Наименование разрядов регистра dma_status

Таблица 405 Назначение разрядов регистра dma_status

Разряд	Наименование	Описание
31…28	test_status	Значение при чтении: 0x0 = контроллер не имеет интегрированной схемы тестирования 0x1 = контроллер имеет интегрированную схему тестирования 0x2 – 0xF = не определено
27…21	-	Не определено
20…16	chnls_minus1	Количество доступных каналов ПДП минус 1. Например: b00000 = контроллер имеет 1 канал ПДП b00001 = контроллер имеет 2 канала ПДП b00010 = контроллер имеет 3 канала ПДП … b11111 = контроллер имеет 32 канала ПДП
15…8	-	Не определено
7…4	state	Текущее состояние автомата управления контроллера. Состояние может быть одним из следующих: b0000 = в покое b0001 = чтение управляющих данных канала b0010 = чтение указателя конца данных источника b0011 = чтение указателя конца данных приемника b0100 = чтение данных источника b0101 = запись данных в приемник b0110 = ожидание запроса на выполнение ПДП

		b0111 = запись управляющих данных канала b1000 = приостановлен b1001 = выполнен b1010 = режим работы с периферией «исполнение с изменением конфигурации» b1011-b1111 = не определено
3…1	-	Не определено
0	master_enable	Состояние контроллера 0 = работа контроллера запрещена 1 = работа контроллера разрешена

DMA_CFG

Регистр конфигурации ПДП.

Данный регистр имеет доступ только на запись. Регистр определяет состояние контроллера. Рисунок 113 показывает наименование разрядов этого регистра. Таблица 406 перечисляет назначение разрядов регистра.

Рисунок 113. Наименование разрядов регистра dma_cfg

Таблица 406 Назначение разрядов регистра dma_cfg

Разряд	Наименование	Описание
31…8	-	Не определено, следует записывать 0.
7…5	chnl_prot_ctrl	Определяет уровни индикации сигналов HPROT[3:1] защиты шины AHB-Lite: Разряд [7] управляет сигналом HPROT[3], с целью индикации о появлении доступа с кэшированием. Разряд [6] управляет сигналом HPROT[2], с целью индикации о появлении доступа с буферизацией. Разряд [5] управляет сигналом HPROT[1], с целью индикации о появлении привилегированного доступа. Примечание: Если разряд[n] = 1, то соответствующий сигнал HPROT в состоянии 1. Если разряд[n] = 0, то соответствующий сигнал HPROT в состоянии 0.
4…1	-	Не определено, следует записывать 0.
0	master_enable	Определяет состояние контроллера 0 = запретить работу контроллера 1 = разрешить работу контроллера

CTRL_BASE_PTR

Регистр базового адреса управляющих данных каналов.

Данный регистр имеет доступ на запись и чтение. Регистр определяет базовый адрес системной памяти размещения управляющих данных каналов.

Примечание:

Контроллер не содержит внутреннюю память для хранения управляющих данных каналов.

Размер системной памяти предназначенной контроллеру зависит от количества каналов ПДП, использующихся контроллером, а также от возможности использования альтернативных управляющих данных каналов. Поэтому количество разрядов регистра необходимых для задания базового адреса варьируется, и зависит от варианта построения системы.

Если контроллер находится в состоянии сброса, то чтение регистра запрещено.

Рисунок 114 показывает наименование разрядов этого регистра, в зависимости от количества использующихся каналов ПДП. Таблица 407 перечисляет назначение разрядов регистра.

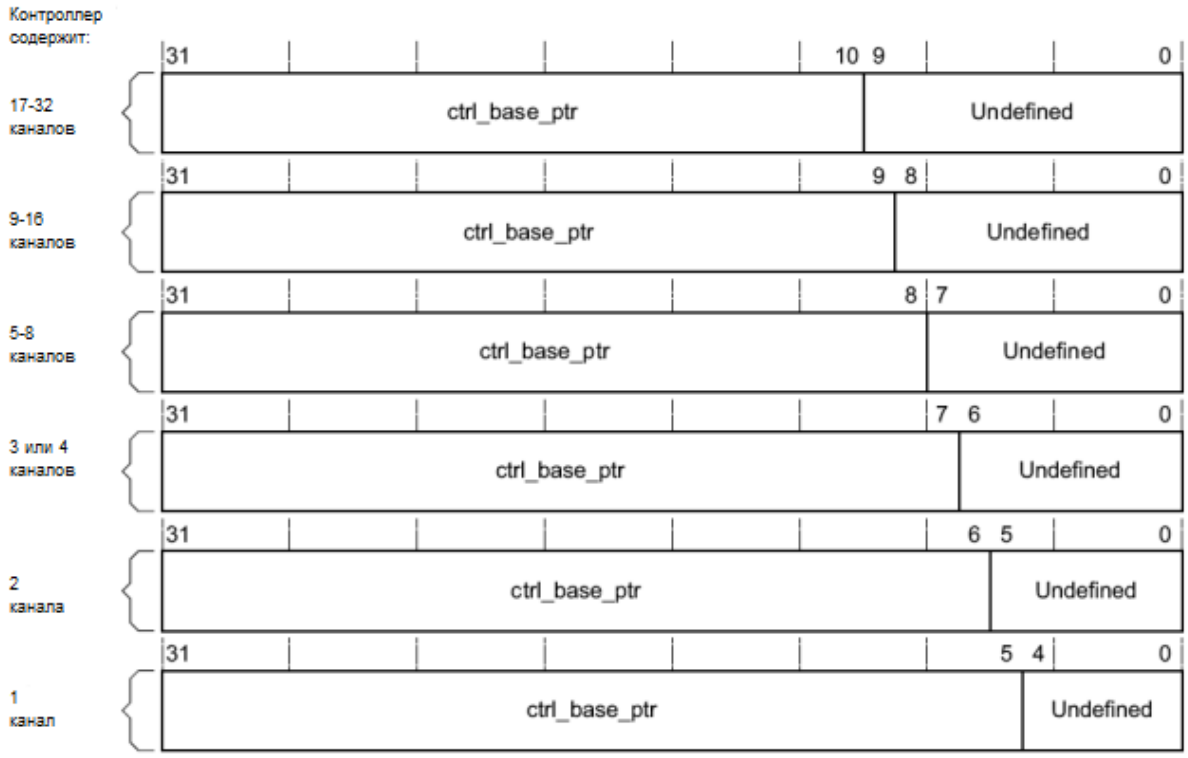


Рисунок 114. Наименование разрядов регистра ctrl_base_ptr

Таблица 407 Назначение разрядов регистра ctrl_base_ptr

Разряд	Наименование	Описание
[31:PL230_DMA_CHNL_BITS+5]	ctrl_base_ptr	Указатель на базовый адрес первичной структуры управляющих данных. См. соответствующий раздел.
[PL230_DMA_CHNL_BITS+4:0]	-	Не определено, следует записывать 0.

, где PL230_DMA_CHNL_BITS – минимальное число разрядов необходимых для представления количества использующихся каналов минус 1. Значения, которые могут присваиваться PL230_DMA_CHNL_BITS, следующие:

0 – контроллер содержит 1 канал ПДП
1 – контроллер содержит 2 канала ПДП
2 – контроллер содержит 3 или 4 канала ПДП
3 – контроллер содержит от 5 до 8 каналов ПДП
4 – контроллер содержит от 9 до 16 каналов ПДП
5 – контроллер содержит от 17 до 32 каналов ПДП

ALT_CTRL_BASE_PTR

Регистр базового адреса альтернативных управляющих данных каналов.

Данный регистр имеет доступ только на чтение. Регистр возвращает при чтении указатель базового адреса альтернативных управляющих данных каналов. Если контроллер находится в состоянии сброса, то чтение регистра запрещено. **Ошибка! Источник ссылки не найден.** показывает наименование разрядов этого регистра. Этот регистр позволяет не производить вычисления базового адреса альтернативных управляющих данных каналов. Таблица 408 перечисляет назначение разрядов регистра.

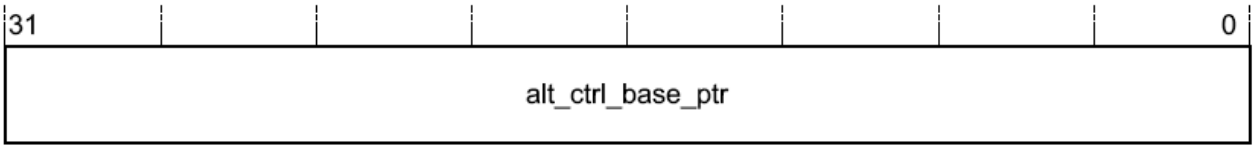


Рисунок 115. Наименование разрядов регистра alt_ctrl_base_ptr

Таблица 408 Назначение разрядов регистра alt_ctrl_base_ptr

Разряд	Наименование	Описание
31…0	alt_ctrl_base_ptr	Указатель на базовый адрес альтернативной структуры управляющих данных.

DMA_WAITONREQ_STATUS

Регистр статуса ожидания запроса на обработку каналов.

Данный регистр имеет доступ только на чтение. Регистр возвращает при чтении состояние сигналов dma_waitonreq[]. Если контроллер находится в состоянии сброса, то чтение регистра запрещено. Рисунок 116 показывает наименование разрядов этого регистра. Таблица 409 перечисляет назначение разрядов регистра.

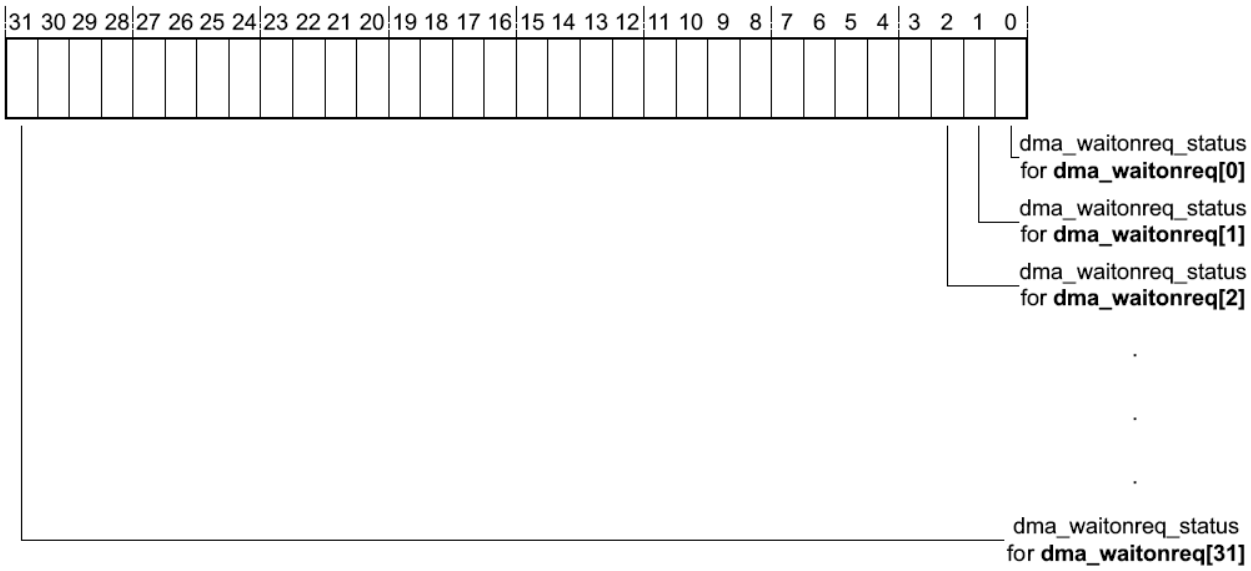


Рисунок 116. Наименование разрядов регистра dma_waitonreq_status

Таблица 409 Назначение разрядов регистра dma_waitonreq_status

Разряд	Наименование	Описание
31…0	dma_waitonreq_status	Состояние сигналов ожидания запроса на обработку каналов ПДП. Если при чтении Разряд [C] =0 означает, что dma_waitonreq[C] в состоянии 0 Разряд [C] =1 означает, что dma_waitonreq[C] в состоянии 1

CHNL_SW_REQUEST

Регистр программного запроса на обработку каналов.

Данный регистр имеет доступ только на запись. Регистр позволяет устанавливать программно запрос на выполнение цикла ПДП. Рисунок 117 показывает наименование разрядов этого регистра. Таблица 410 перечисляет назначение разрядов регистра.

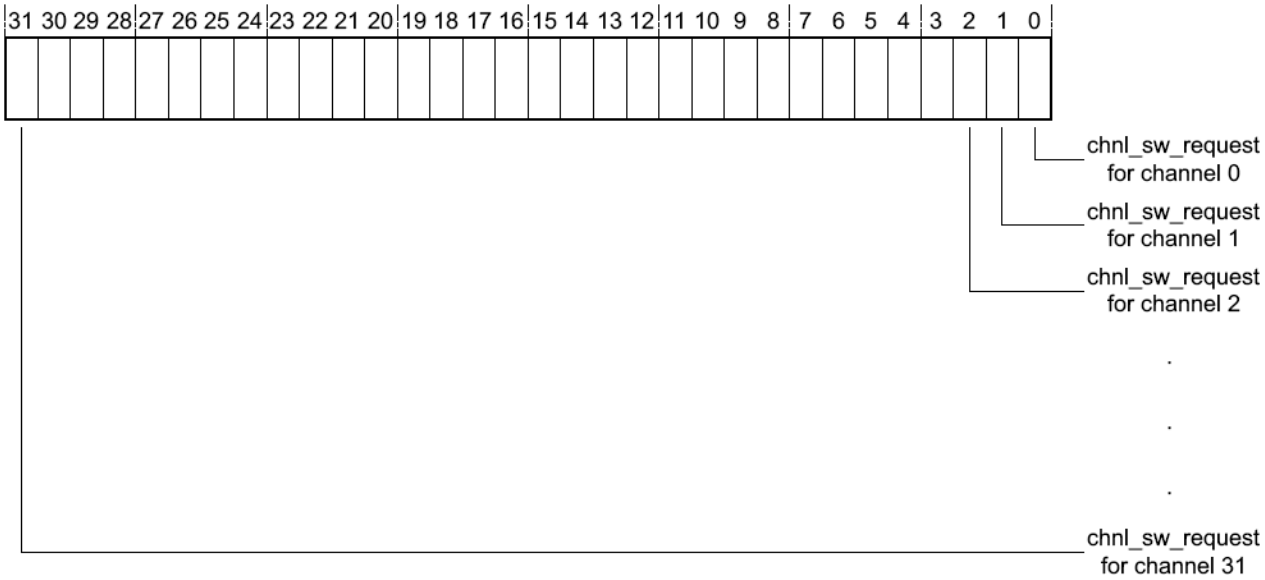


Рисунок 117. Наименование разрядов регистра chnl_sw_request

Таблица 410 Назначение разрядов регистра chnl_sw_request

Разряд	Наименование	Описание
31…0	chnl_sw_request	Устанавливает соответствующий разряд для генерации программного запроса на выполнение цикла ПДП по соответствующему каналу ПДП. Если при записи: разряд [C] =0 означает, что запрос на выполнение цикла ПДП по каналу C не будет установлен; разряд [C] =1 означает, что запрос на выполнение цикла ПДП по каналу C будет установлен; запись разряда соответствующего нереализованному каналу, означает, что запрос на выполнение цикла ПДП не будет установлен.

CHNL_USEBURST_SET

Регистр установки пакетного обмена каналов.

Данный регистр имеет доступ на чтение и запись. Регистр отключает выполнение одиночных запросов по установке dma_sreq[] и поэтому, будут обрабатываться и исполняться только запросы по dma_req[]. Регистр возвращает при чтении состояние установок пакетного обмена. Рисунок 118 показывает наименование разрядов этого регистра. Таблица 411 перечисляет назначение разрядов регистра.

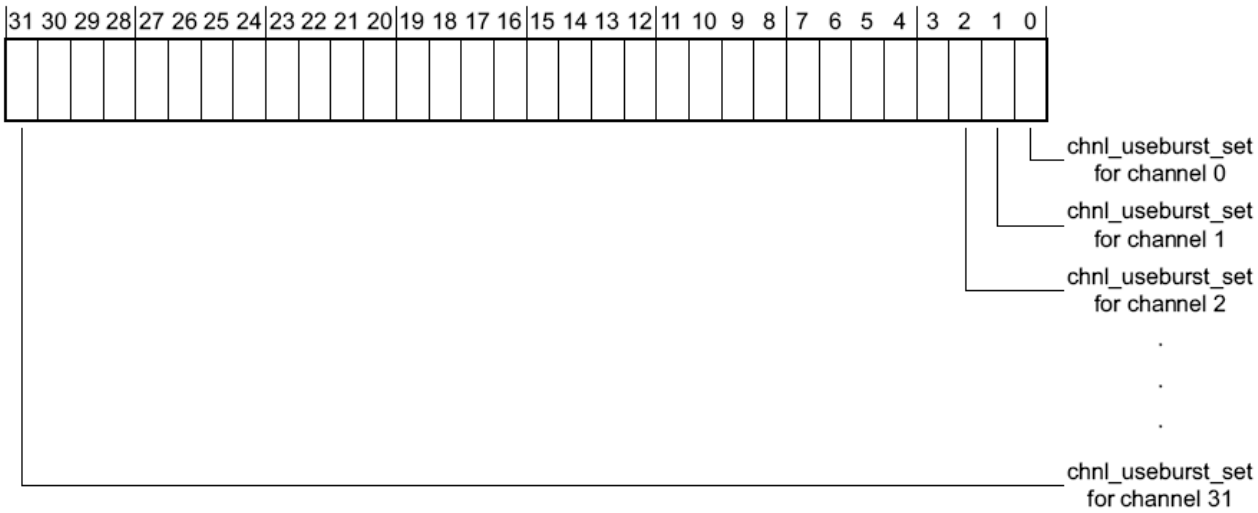


Рисунок 118. Наименование разрядов регистра chnl_useburst_set

Таблица 411 Назначение разрядов регистра chnl_useburst_set

Разряд	Наименование	Описание
31…0	chnl_useburst_set	Отключает обработку запросов на выполнение циклов ПДП от dma_sreq[] и возвращает при чтении состоянии этих настроек. Если при чтении: Разряд [C] =0 означает, что канал ПДП C выполняет циклы ПДП в ответ на запросы, полученные от dma_sreq[] и dma_req[]. Контроллер выполняет одиночные передачи или 2R передач. Разряд [C] =1 означает, что канал ПДП C выполняет циклы

		ПДП в ответ на запросы, полученные только от dma_req[]. Контроллер выполняет 2R передач. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать chnl_useburst_clr регистр и установить соответствующий разряд C в 0. Разряд [C] =1 отключает возможность обрабатывать запросы на выполнение циклов ПДП, полученные от dma_sreq[]. Контроллер выполняет 2R передач. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта.

После выполнения предпоследней передачи из 2R передач, в том случае если число оставшихся передач (N) меньше чем 2R, контроллер сбрасывает разряд chnl_useburst_set в 0. Это позволяет выполнять оставшиеся передачи, используя dma_sreq[] и dma_req[].

Примечание:

При программировании channel_cfg значением N меньшим, чем 2R, запрещена установка соответствующего разряда chnl_useburst_set, в случае если периферийный блок не поддерживает сигнал dma_req[].

В режиме работы с периферией «исполнение с изменением конфигурации», если разряд next_useburst установлен в channel_cfg, то контроллер устанавливает chnl_useburst_set [C] в 1, после окончания цикла ПДП, использующего альтернативные управляющие данные.

CHNL_USEBURST_CLR

Регистр сброса пакетного обмена каналов.

Данный регистр имеет доступ только на запись. Регистр разрешает выполнение одиночных запросов по установке dma_sreq[]. Рисунок 119 показывает наименование разрядов этого регистра. Таблица 412 перечисляет назначение разрядов регистра.

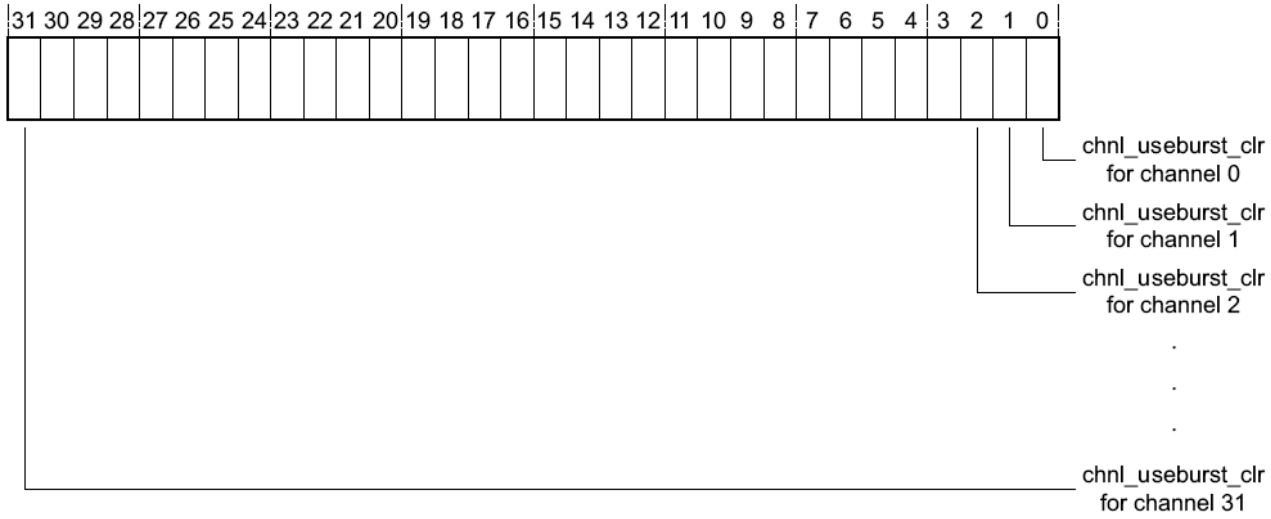


Рисунок 119. Наименование разрядов регистра chnl_useburst_clr

Таблица 412 Назначение разрядов регистра chnl_useburst_clr

Разряд	Наименование	Описание
31…0	chnl_useburst_clr	Установка соответствующего разряда разрешает обработку запросов на выполнение циклов ПДП от dma_sreq[]. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать chnl_useburst_set регистр для отключения обработки запросов от dma_sreq[]. Разряд [C] =1 разрешает обрабатывать запросы на выполнение циклов ПДП, полученные от dma_sreq[]. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта.

CHNL_REQ_MASK_SET

Регистр маскирования запросов на обслуживание каналов.

Данный регистр имеет доступ на чтение и запись. Регистр отключает установку запросов на выполнение циклов ПДП на dma_sreq[] и dma_req[]. Регистр возвращает при чтении состояние установок маскирования запросов от dma_sreq[] и dma_req[] на обслуживание каналов. Рисунок 120 показывает наименование разрядов этого регистра. Таблица 413 перечисляет назначение разрядов регистра.

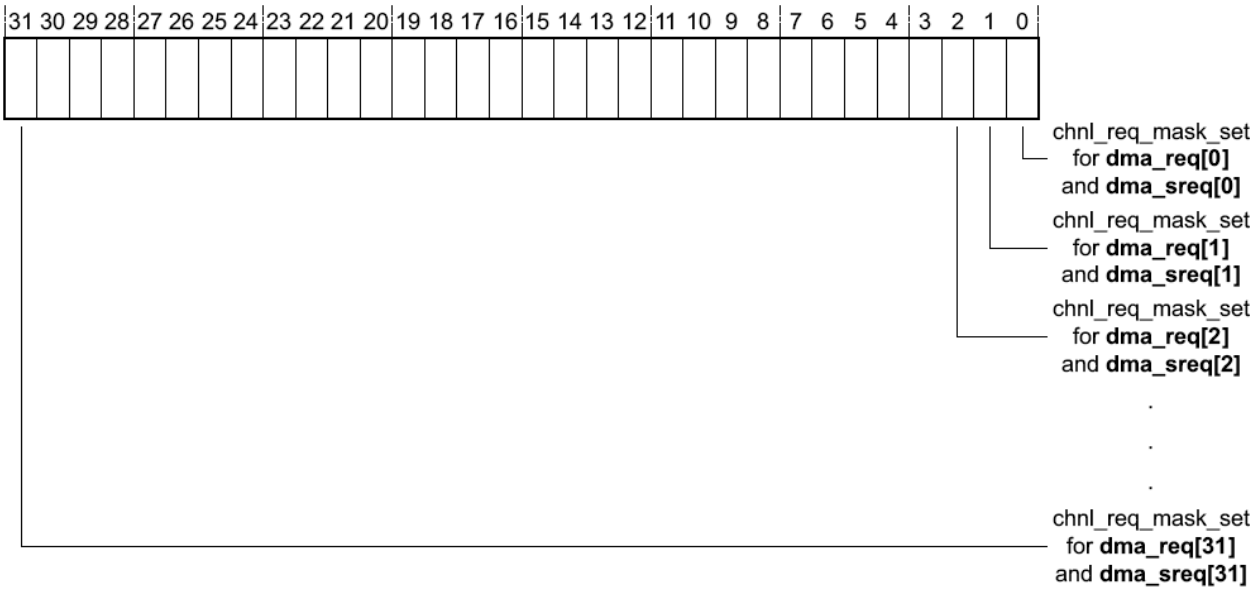


Рисунок 120. Наименование разрядов регистра chnl_req_mask_set

Таблица 413 Назначение разрядов регистра chnl_req_mask_set

Разряд	Наименование	Описание
31…0	chnl_req_mask_set	Отключает обработку запросов по dma_sreq[] и dma_req[] на выполнение циклов ПДП от каналов и возвращает при чтении состоянии этих настроек. Если при чтении: Разряд [C] =0 означает, что канал ПДП C выполняет циклы ПДП в ответ на запросы. Разряд [C] =1 означает, что канал ПДП C не выполняет циклы ПДП в ответ на запросы.

		При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать chnl_req_mask_clr регистр для разрешения установки запросов. Разряд [C] =1 отключает установку запросов на выполнение циклов ПДП, по dma_sreq[] и dma_req[]. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта.

CHNL_REQ_MASK_CLR

Регистр очистки маскирования запросов на обслуживание каналов.

Данный регистр имеет доступ только на запись. Регистр разрешает установку запросов на выполнение циклов ПДП на dma_sreq[] и dma_req[]. Рисунок 121 показывает наименование разрядов этого регистра. Таблица 414 перечисляет назначение разрядов регистра.

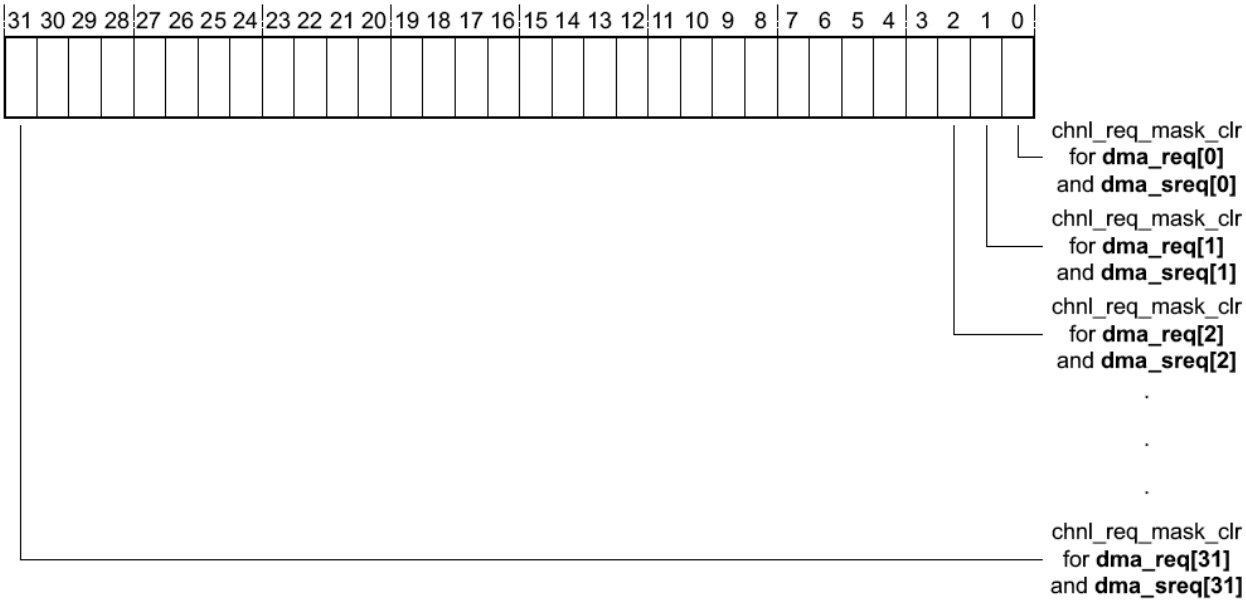


Рисунок 121. Наименование разрядов регистра chnl_req_mask_clr

Таблица 414 Назначение разрядов регистра chnl_req_mask_clr

Разряд	Наименование	Описание
31…0	chnl_req_mask_clr	Установка соответствующего разряда разрешает установку запросов по dma_sreq[] и dma_req[] на выполнение циклов ПДП от каналов. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать chnl_req_mask_set регистр для отключения установки запросов. Разряд [C] =1 разрешает установку запросов на выполнение циклов ПДП, по dma_sreq[] и dma_req[]. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта.

CHNL_ENABLE_SET

Регистр установки разрешения каналов.

Данный регистр имеет доступ на чтение и запись. Регистр разрешает работу каналов ПДП. Регистр возвращает при чтении состояние разрешений работы каналов ПДП. Рисунок 122 показывает наименование разрядов этого регистра. Таблица 415 перечисляет назначение разрядов регистра.

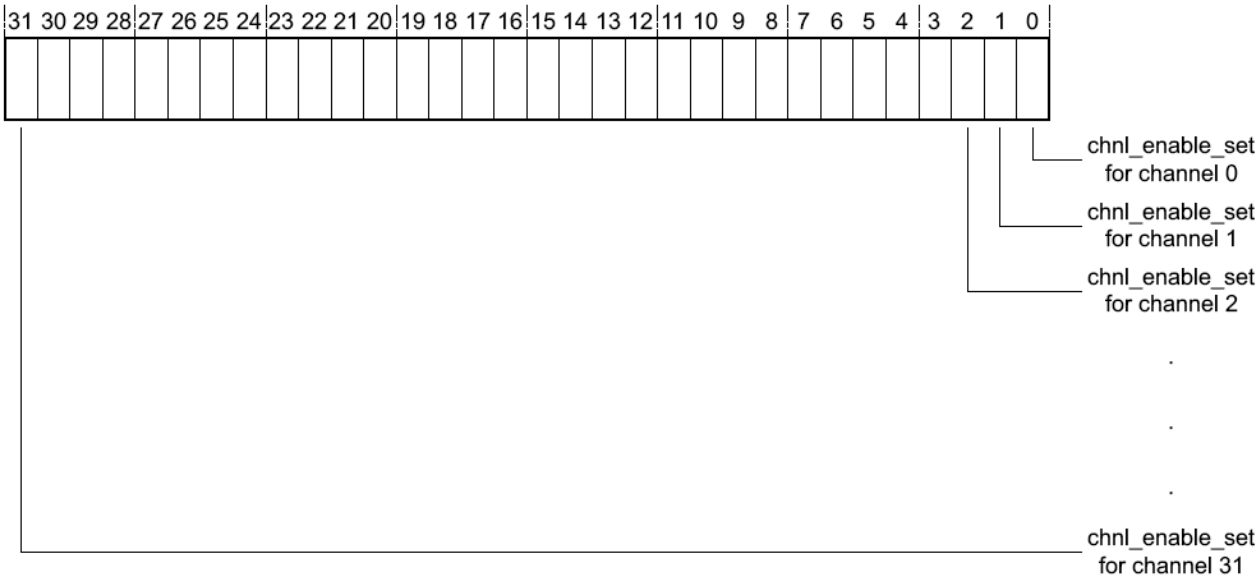


Рисунок 122. Наименование разрядов регистра chnl_enable_set

Таблица 415 Назначение разрядов регистра chnl_enable_set

Разряд	Наименование	Описание
31…0	chnl_enable_set	Разрешает работу каналов ПДП и возвращает при чтении состоянии этих настроек. Если при чтении: Разряд [C] =0 означает, что канал ПДП C отключен. Разряд [C] =1 означает, что работа канала ПДП C разрешена. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать chnl_enable_clr регистр для отключения канала. Разряд [C] =1 разрешает работу канала ПДП C. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта.

CHNL_ENABLE_CLR

Регистр сброса разрешения каналов.

Данный регистр имеет доступ только на запись. Регистр запрещает работу каналов ПДП. Рисунок 123 показывает наименование разрядов этого регистра. Таблица 416 перечисляет назначение разрядов регистра.

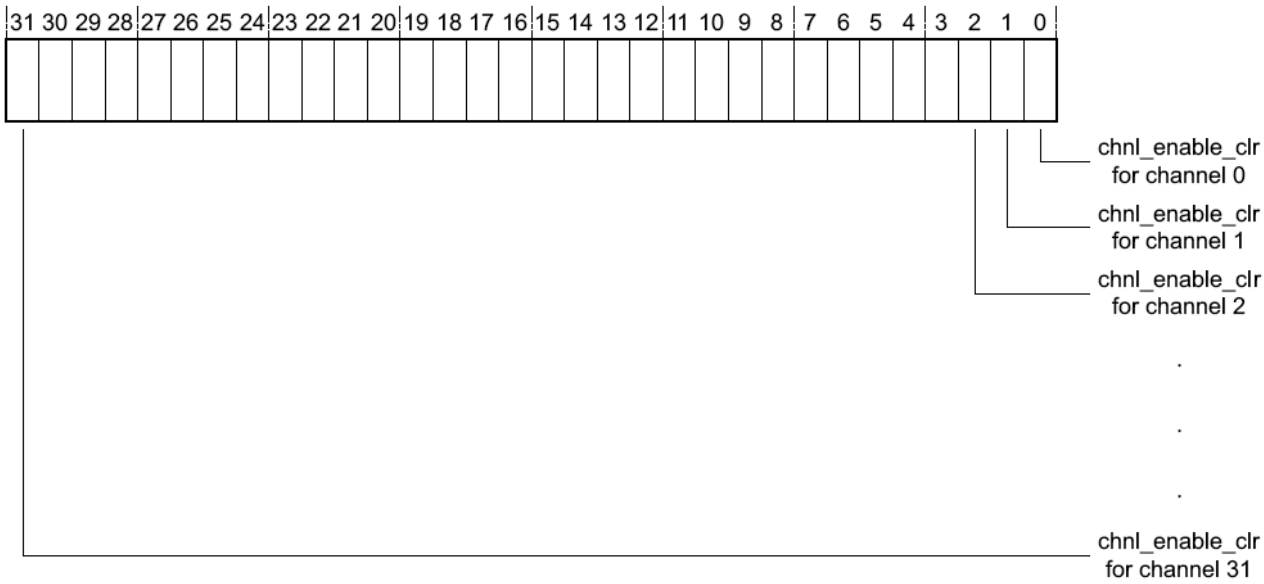


Рисунок 123. Наименование разрядов регистра chnl_enable_clr

Таблица 416 Назначение разрядов регистра chnl_enable_clr

Разряд	Наименование	Описание
31…0	chnl_enable_clr	Установка соответствующего разряда запрещает работу соответствующего канала ПДП. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать chnl_enable_set регистр для разрешения работы канала. Разряд [C] =1 запрещает работу канала ПДП C. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта. Примечание: Контроллер может отключить канал ПДП, установив соответствующий разряд в следующих случаях: - при завершении цикла ПДП; - при чтении из channel_cfg с полем cycle_ctrl установленным в b000; - при появлении ошибки на шине AHB-Lite.

CHNL_PRI_ALT_SET

Регистр установки первичной/альтернативной структуры управляющих данных каналов.

Данный регистр имеет доступ на запись и чтение. Регистр разрешает работу канала ПДП с использованием альтернативной структуры управляющих данных. Чтение регистра возвращает состояние каналов ПДП (какую структуру управляющих данных использует каждый канал ПДП). Рисунок 124 показывает наименование разрядов этого регистра. Таблица 417 перечисляет назначение разрядов регистра.

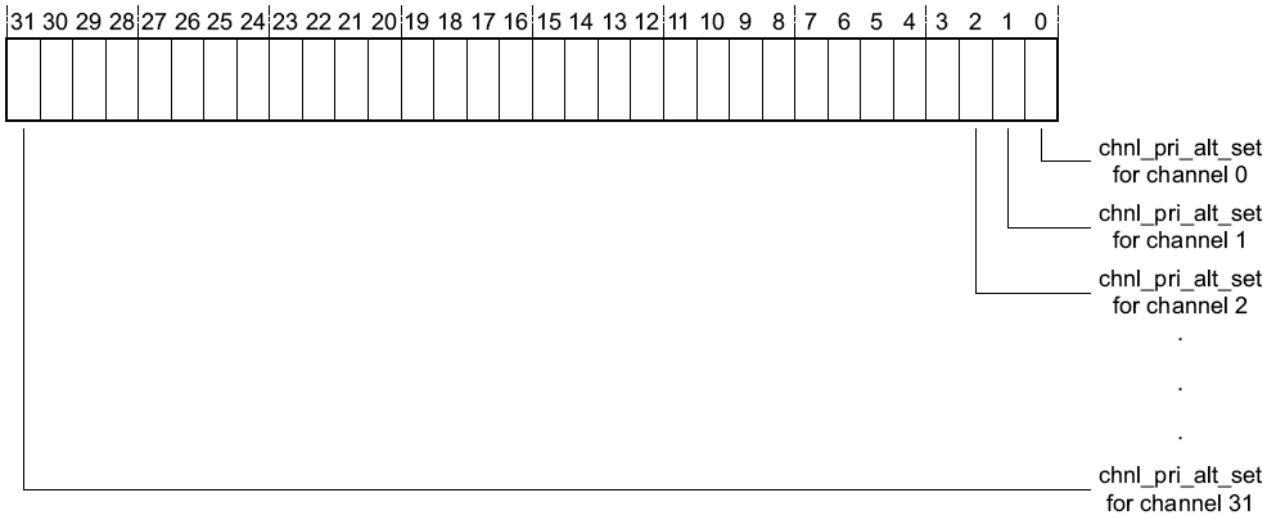


Рисунок 124. Наименование разрядов регистра chnl_pri_alt_set

Таблица 417 Назначение разрядов регистра chnl_pri_alt_set

Разряд	Наименование	Описание
31…0	chnl_pri_alt_set	Установка соответствующего разряда подключает использование альтернативных управляющих данных для соответствующего канала ПДП, чтение возвращает состояние этих настроек. Если при чтении: Разряд [C] =0 означает, что канал ПДП C использует первичную структуру управляющих данных. Разряд [C] =1 означает, что канал ПДП C использует альтернативную структуру управляющих данных. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать chnl_pri_alt_clr регистр для сброса разряда [C] в 0. Разряд [C] =1 подключает использование альтернативной структуры управляющих данных каналом ПДП C. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта. Примечание: Контроллер может переключить значение разряда chnl_pri_alt_set[C] в следующих случаях: - при завершении 4-х передач ПДП указанных в первичной структуре управляющих данных при выполнении цикла ПДП в режимах работы с памятью или периферией «исполнение с изменением конфигурации»; - при завершении всех передач ПДП указанных в первичной структуре управляющих данных при выполнении цикла ПДП в режиме Пинг-понг; - при завершении всех передач ПДП указанных в альтернативной структуре управляющих данных при

		выполнении цикла ПДП в режимах: • пинг-понг; • работа с памятью «исполнение с изменением конфигурации»; • работа с периферией «исполнение с изменением конфигурации».

CHNL_PRI_ALT_CLR

Регистр сброса первичной/альтернативной структуры управляющих данных каналов.

Данный регистр имеет доступ только на запись. Регистр разрешает работу канала ПДП с использованием первичной структуры управляющих данных. Рисунок 125 показывает наименование разрядов этого регистра. Таблица 418 перечисляет назначение разрядов регистра.

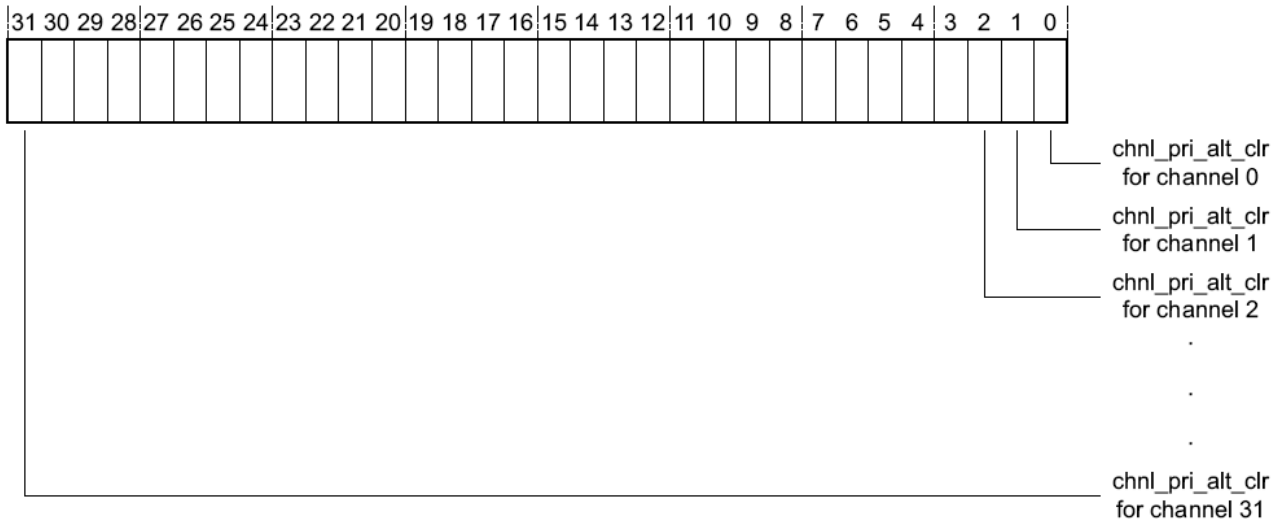


Рисунок 125. Наименование разрядов регистра chnl_pri_alt_clr

Таблица 418 Назначение разрядов регистра chnl_pri_alt_clr

Разряд	Наименование	Описание
31…0	chnl_pri_alt_clr	Установка соответствующего разряда подключает использование первичных управляющих данных для соответствующего канала ПДП. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать chnl_pri_alt_set регистр для выбора альтернативных управляющих данных. Разряд [C] =1 подключает использование первичной структуры управляющих данных каналом ПДП C. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта. Примечание: Контроллер может переключить значение разряда chnl_pri_alt_clr[C] в следующих случаях: - при завершении 4-х передач ПДП указанных в первичной структуре управляющих данных при выполнении цикла ПДП в режимах работы с памятью или периферией «исполнение с

		изменением конфигурации»; - при завершении всех передач ПДП указанных в первичной структуре управляющих данных при выполнении цикла ПДП в режиме Пинг-понг; - при завершении всех передач ПДП указанных в альтернативной структуре управляющих данных при выполнении цикла ПДП в режимах: • пинг-понг; • работа с памятью «исполнение с изменением конфигурации»; • работа с периферией «исполнение с изменением конфигурации».

CHNL_PRIORITY_SET

Регистр установки приоритета каналов.

Данный регистр имеет доступ на запись и чтение. Регистр позволяет присвоить высокий приоритет каналу ПДП. Чтение регистра возвращает состояние приоритета каналов ПДП. Рисунок 126 показывает наименование разрядов этого регистра. Таблица 1Таблица 419 перечисляет назначение разрядов регистра.

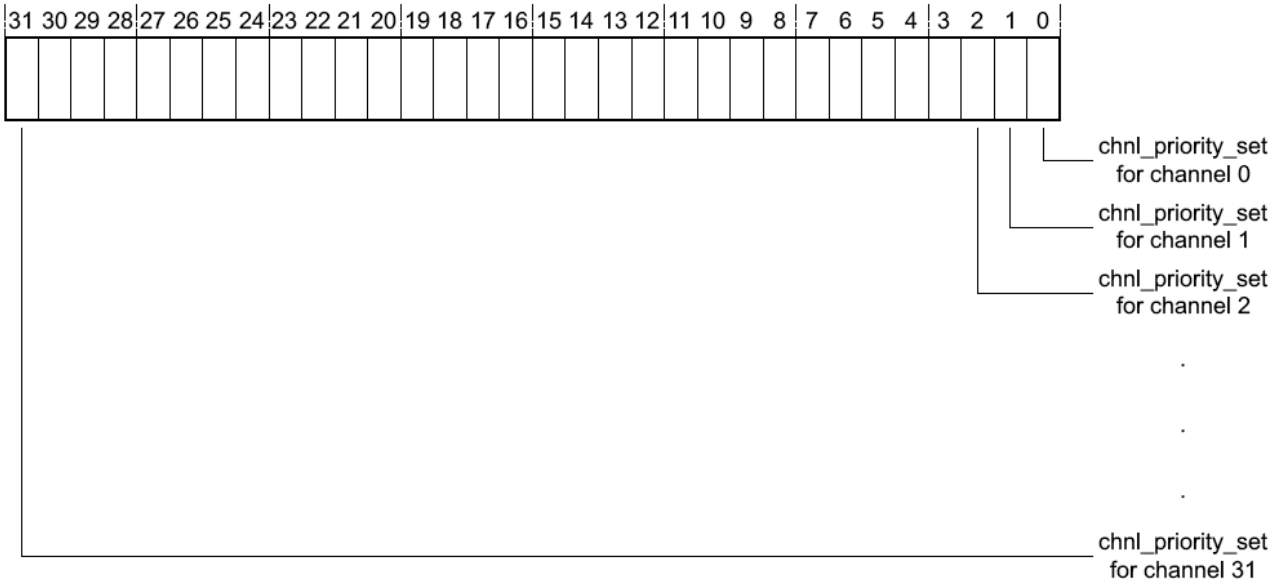


Рисунок 126. Наименование разрядов регистра chnl_priority_set

Таблица 419 Назначение разрядов регистра chnl_priority_set

Разряд	Наименование	Описание
31…0	chnl_priority_set	Установка высокого приоритета каналу ПДП, чтение возвращает состояние приоритета каналов ПДП. Если при чтении: Разряд [C] =0 означает, что каналу ПДП C присвоен уровень приоритета по умолчанию. Разряд [C] =1 означает, что каналу ПДП C присвоен высокий уровень приоритета. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать chnl_priority_clr регистр для установки каналу C уровня приоритета по умолчанию.

		Разряд [C] =1 устанавливает каналу ПДП C высокий уровень приоритета. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта.

CHNL_PRIORITY_CLR

Регистр сброса приоритета каналов.

Данный регистр имеет доступ только на запись. Регистр позволяет присвоить каналу ПДП уровень приоритета по умолчанию. Рисунок 127 показывает наименование разрядов этого регистра. Таблица 420 перечисляет назначение разрядов регистра.

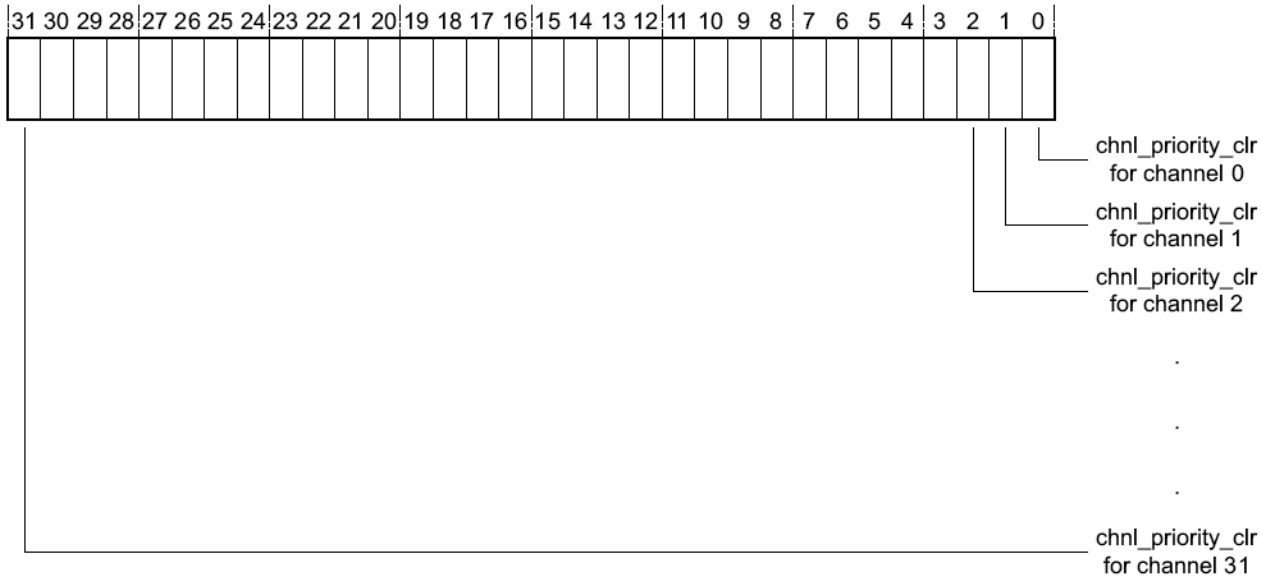


Рисунок 127. Наименование разрядов регистра chnl_priority_clr

Таблица 420 Назначение разрядов регистра chnl_priority_clr

Разряд	Наименование	Описание
31…0	chnl_priority_clr	Установка разряда присваивает соответствующему каналу ПДП уровень приоритета по умолчанию. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать chnl_priority_set регистр для установки каналу C высокого уровня приоритета. Разряд [C] =1 устанавливает каналу ПДП C уровень приоритета по умолчанию. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта.

ERR_CLR

Регистр сброса флага ошибки.

Данный регистр имеет доступ на запись и чтение. Регистр позволяет сбрасывать сигнал dma_err в 0. Чтение регистра возвращает состояние сигнала dma_err. Рисунок 128

показывает наименование разрядов этого регистра. Таблица 421 перечисляет назначение разрядов регистра.

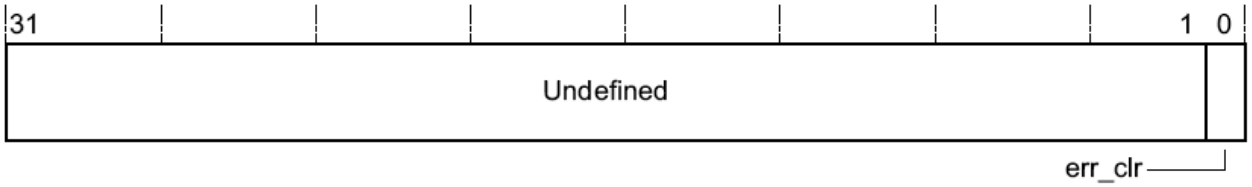


Рисунок 128. Наименование разрядов регистра err_clr

Таблица 421 Назначение разрядов регистра err_clr

Разряд	Наименование	Описание
31…1	-	Не определено. Следует записывать 0.
0	chnl_priority_set	Установка сигнала в состояние 0, чтение возвращает состояние сигнала (флага) dma_err. Если при чтении: Разряд [C] =0 означает, что dma_err находится в состоянии 0. Разряд [C] =1 означает, что dma_err находится в состоянии 1. При записи: Разряд [C] =0 не дает эффекта. Состояние dma_err останется неизменным. Разряд [C] =1 сбрасывает сигнал (флаг) dma_err в состояние 0. Для целей тестирования возможно использовать регистр err_set, чтобы установить сигнал dma_err в состояние 1. Примечание: При сбросе сигнала dma_err одновременно с появлением ошибки на шине AHB-Lite, то приоритет отдается ошибке и следовательно, значение регистра (и dma_err) останется неизменным (несброшенным).

Регистры идентификации.

Периферийные регистры идентификации размещаются по следующим адресам:

0xFE0 регистр идентификации периферии 0 periph_id_0.
0xFE4 регистр идентификации периферии 1 periph_id_1.
0xFE8 регистр идентификации периферии 2 periph_id_2.
0xFEC регистр идентификации периферии 3 periph_id_3.
0xFD0 регистр идентификации периферии 4 periph_id_4.

Каждый регистр имеет доступ только по чтению и возвращает 8-ми разрядное число. Можно рассматривать 8-ми разрядные регистры periph_id_[3:0] как один 32-разрядный регистр. Рисунок 129 показывает наименование разрядов этих регистров. Таблица 422 перечисляет назначение разрядов этих регистров.

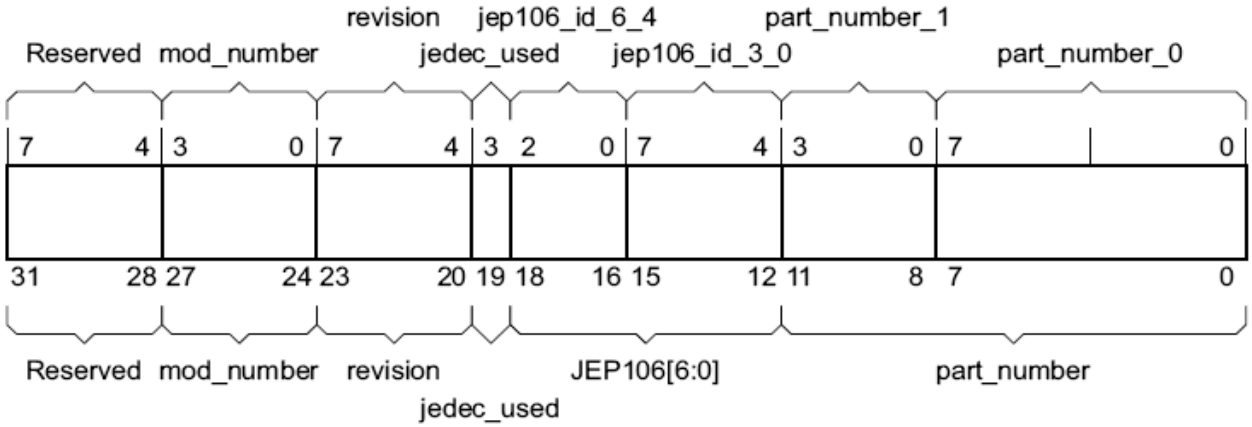


Рисунок 129. Наименование разрядов регистров periph_id_[3:0]

Таблица 422 Назначение разрядов регистров periph_id_[3:0]

Разряд	Наименование	Описание
31…28	зарезервировано	Зарезервировано для будущего использования. При чтении возвращает неопределенное значение.
27…24	mod_number	Идентифицирует данные, относящиеся к партнеру ARM.
23…20	revision	Идентифицирует номер версии периферийного блока. Номера версий начинаются с 0 и зависят от последовательности обновлений.
19	jedec_used	Идентифицирует, использует ли контроллер идентификационный код производителя в соответствии с JEP106.
18…12	JEP106[6:0]	Идентифицирует разработчика. Эти разряды должны иметь значение 0x41, указывая то, что блок был разработан компанией ARM.
11…0	part_number	Идентифицирует периферийный блок.

Следующие подразделы описывают регистры идентификации периферийного блока:

- регистр идентификации периферии 0;
- регистр идентификации периферии 1;
- регистр идентификации периферии 2;
- регистр идентификации периферии 3;
- регистр идентификации периферии 4.

Регистр идентификации периферии 0.

Регистр имеет доступ только по чтению. Значение регистра установлено аппаратно и остается неизменным после сброса. Рисунок 130 показывает наименование разрядов этого регистра. Таблица 423 перечисляет назначение разрядов этого регистра.

Рисунок 130. Наименование разрядов регистра periph_id_0

Таблица 423 Назначение разрядов регистра periph_id_0

Разряд	Наименование	Описание
31…8	-	Не определено.
7…0	part_number_0	При чтении возвращает значение 0x30.

Регистр идентификации периферии 1.

Регистр имеет доступ только по чтению. Значение регистра установлено аппаратно и остается неизменным после сброса. Рисунок 131 показывает наименование разрядов этого регистра. Таблица 424 перечисляет назначение разрядов этого регистра.

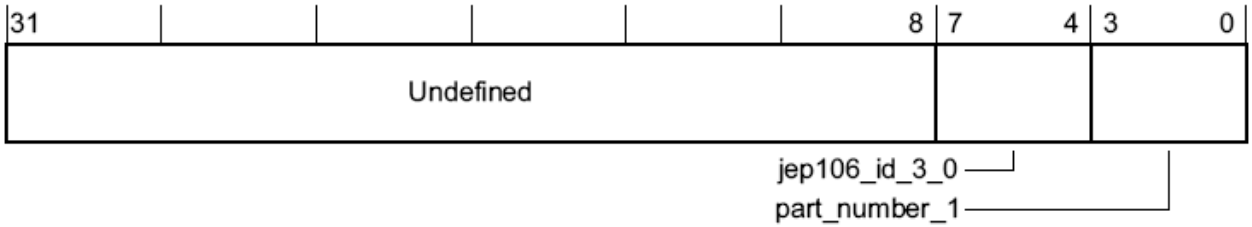


Рисунок 131. Наименование разрядов регистра periph_id_1

Таблица 424 Назначение разрядов регистра periph_id_1

Разряд	Наименование	Описание
31…8	-	Не определено.
7…4	jep106_id_3_0	Идентификационный код (разряды [3:0]) JEP106 (см. стандарт JEP106). При чтении возвращает значение 0xB, так как разработчиком периферийного блока является компания ARM.
3…0	part_number_0	При чтении возвращает значение 0x2.

Регистр идентификации периферии 2.

Регистр имеет доступ только по чтению. Значение регистра установлено аппаратно и остается неизменным после сброса. Рисунок 132 показывает наименование разрядов этого регистра. Таблица 425 перечисляет назначение разрядов этого регистра.

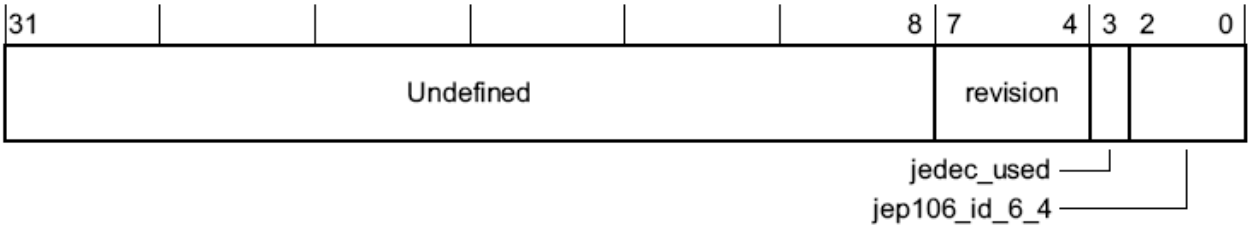


Рисунок 132. Наименование разрядов регистра periph_id_2

Таблица 425 Назначение разрядов регистра periph_id_2

Разряд	Наименование	Описание
31…8	-	Не определено.
7…4	revision	Номер версии контроллера. Если версия контроллера r0p0, то при чтении возвращает значение 0x0
3	jedec_used	Разряд показывает, использует ли контроллер идентификационный код производителя из списка JEDEC в соответствии с JEP106. При чтении всегда возвращает значение 0x1.

2…0	jep106_id_3_0	Идентификационный код (разряды [6:4]) JEP106 (см. стандарт JEP106). При чтении возвращает значение 0x3, так как разработчиком периферийного блока является компания ARM.

Регистр идентификации периферии 3.

Регистр имеет доступ только по чтению. Значение регистра установлено аппаратно и остается неизменным после сброса. Рисунок 133 показывает наименование разрядов этого регистра. Таблица 426 перечисляет назначение разрядов этого регистра.

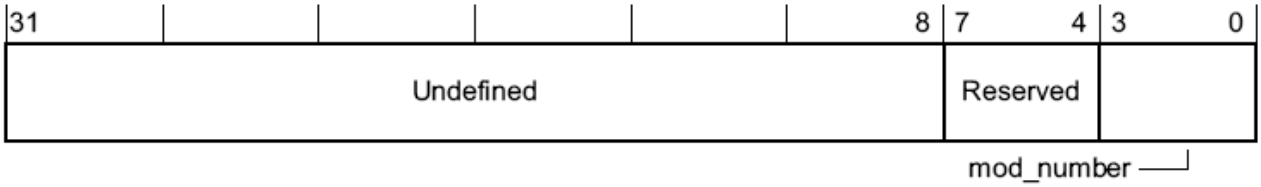


Рисунок 133 Наименование разрядов регистра periph_id_3

Таблица 426 Назначение разрядов регистра periph_id_3

Разряд	Наименование	Описание
31…8	-	Не определено.
7…4	зарезервировано	Зарезервировано для будущего использования. При чтении возвращает неопределенное значение.
3…0	mod_number	Потребитель должен изменить значение этих разрядов при изменении RTL описания контроллера. Компания ARM устанавливает значение этих разрядов в 0x0.

Регистр идентификации периферии 4.

Регистр имеет доступ только по чтению. Значение регистра установлено аппаратно и остается неизменным после сброса. Рисунок 134 показывает наименование разрядов этого регистра. Таблица 427 перечисляет назначение разрядов этого регистра.

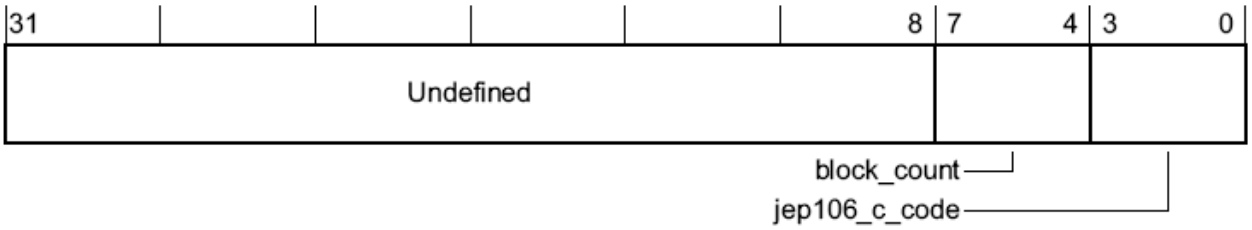


Рисунок 134. Наименование разрядов регистра periph_id_4

Таблица 427 Назначение разрядов регистра periph_id_4

Разряд	Наименование	Описание
31…8	-	Не определено.
7…4	block_count	Количество требуемых адресных блоков емкостью 4 Кбайт для доступа к регистрам, выражено как степень 2. При чтении возвращает значение 0x0.
3…0	jep106_c_code	Значение кода продолжения JEP106 показывает, как много продолжающихся знаков 0x7F будет представлено в идентификационном коде производителя (см. стандарт JEP106). При чтении возвращает значение 0x4

Регистры идентификации PrimeCell.

Значение идентификационного кода PrimeCell ID является 32-разрядным числом. Для того чтобы гарантировать доступ к данному коду в различных системах, оно реализовано как 8-ми разрядных регистра. В каждом регистре только младшие восемь разрядов содержат данные. Регистры идентификации PrimeCell размещаются по следующим адресам:

0xFF0 регистр идентификации PrimeCell 0 pcell_id_0.
0xFF4 регистр идентификации PrimeCell 1 pcell_id_1.
0xFF8 регистр идентификации PrimeCell 2 pcell_id_2.
0xFFC регистр идентификации PrimeCell 3 pcell_id_3.

Можно рассматривать все регистры как один 32-разрядный регистр, содержащий 32-разрядный идентификационный код PrimeCell ID. Можно использовать регистр для автоматической настройки BIOS. Значение регистра pcell_id_[3:0] установлено в 0xB105F00D. Рисунок 135 показывает наименование разрядов этих регистров. Таблица 428 перечисляет назначение разрядов концептуального 32-разрядного регистра.

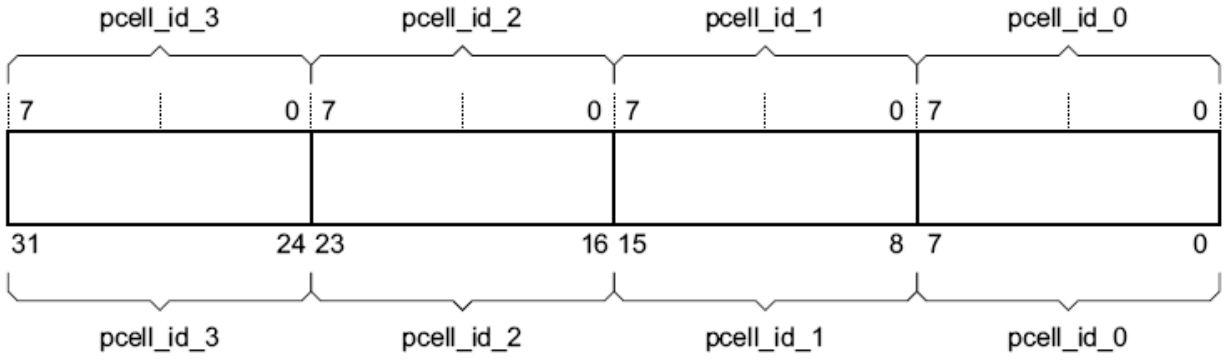


Рисунок 135. Наименование разрядов регистра pcell_id_[3:0]

Таблица 428 Назначение разрядов регистра pcell_id_[3:0]

Разряд	Наименование	Описание
31…24	pcell_id_3	При чтении возвращает значение 0xB1.
23…16	pcell_id_2	При чтении возвращает значение 0x05.
15…8	pcell_id_1	При чтении возвращает значение 0xF0.
7…0	pcell_id_0	При чтении возвращает значение 0x0D.

Следующие подразделы описывают регистры идентификации PrimeCell:

- регистр идентификации PrimeCell 0;
- регистр идентификации PrimeCell 1;
- регистр идентификации PrimeCell 2;
- регистр идентификации PrimeCell 3.

Регистр идентификации PrimeCell 0.

Регистр имеет доступ только по чтению. Значение регистра установлено аппаратно и остается неизменным после сброса. Рисунок 136 показывает наименование разрядов этого регистра. Таблица 429 перечисляет назначение разрядов этого регистра.

Рисунок 136. Наименование разрядов регистра pcell_id_0

Таблица 429 Назначение разрядов регистра pcell_id_0

Разряд	Наименование	Описание
31…8	-	Не определено.
7…0	pcell_id_0	При чтении возвращает значение 0x0D.

Регистр идентификации PrimeCell 1.

Регистр имеет доступ только по чтению. Значение регистра установлено аппаратно и остается неизменным после сброса. Рисунок 137 показывает наименование разрядов этого регистра. Таблица 430 перечисляет назначение разрядов этого регистра.

Рисунок 137. Наименование разрядов регистра pcell_id_1

Таблица 430 Назначение разрядов регистра pcell_id_1

Разряд	Наименование	Описание
31…8	-	Не определено.
7…0	pcell_id_1	При чтении возвращает значение 0xF0.

Регистр идентификации PrimeCell 2.

Регистр имеет доступ только по чтению. Значение регистра установлено аппаратно и остается неизменным после сброса. Рисунок 138 показывает наименование разрядов этого регистра. Таблица 431 перечисляет назначение разрядов этого регистра.

Рисунок 138. Наименование разрядов регистра pcell_id_2

Таблица 431 Назначение разрядов регистра pcell_id_2

Разряд	Наименование	Описание
31…8	-	Не определено.
7…0	pcell_id_2	При чтении возвращает значение 0x05.

Регистр идентификации PrimeCell 3.

Регистр имеет доступ только по чтению. Значение регистра установлено аппаратно и остается неизменным после сброса. Рисунок 139 показывает наименование разрядов этого регистра. Таблица 432 перечисляет назначение разрядов этого регистра.

Рисунок 139. Наименование разрядов регистра pcell_id_3

Таблица 432 Назначение разрядов регистра pcell_id_3

Разряд	Наименование	Описание
31…8	-	Не определено.
7…0	pcell_id_3	При чтении возвращает значение 0xB1.

Описание тестовых регистров контроллера

Данная глава описывает дополнительные регистры контроллера, предназначенные для отладки и функциональной верификации.

Глава содержит раздел «описание регистров».

Примечание:

Для доступа к данным ресурсам необходимо при реализации контроллера использовать интегрированные схемы тестирования.

Описание регистров

Этот раздел описывает все тестовые регистры контроллера. Смещение адреса каждого регистра относительно базового адреса фиксировано. Таблица 433 перечисляет тестовые регистры контроллера в порядке смещения адреса.

Таблица 433 Перечень тестовых регистров контроллера

Наименование	**Смещение относительно базового адреса**	**Тип**	**Значение по сбросу**	**Описание**
integration_cfg	0xE00	RW	0x0	Регистр конфигурации
-	0xE04	-	-	Зарезервировано
stall_status	0xE08	RO	0x0	Регистр состояния контроллера
-	0xE0C	-	-	Зарезервировано
dma_req_status	0xE10	RO	0x00000000	Регистр состояния запросов по dma_req каналов ПДП
-	0xE14	-	-	Зарезервировано
dma_sreq_status	0xE18	RO	0x00000000	Регистр состояния запросов по dma_sreq каналов ПДП
-	0xE1C	-	-	Зарезервировано
dma_done_set	0xE20	R/W	0x00000000	Регистр установки флагов окончания цикла ПДП
dma_done_clr	0xE24	WO	-	Регистр сброса флагов окончания цикла ПДП
dma_active_set	0xE28	R/W	0x00000000	Регистр установки флагов активности каналов ПДП
dma_active_clr	0xE2C	WO	-	Регистр сброса флагов активности каналов ПДП
-	0xE30 - 0xE44	-	-	Зарезервировано
err_set	0xE48	WO	-	Регистр установки флага ошибки

Регистр конфигурации

Данный регистр имеет доступ на запись и чтение. Регистр устанавливает возможность контроля флагов dma_active[], dma_done[] и dma_err со стороны интегрированных тестовых схем (тестовых регистров). Если контроллер находится в состоянии сброса, то чтение регистра запрещено. Рисунок 140 показывает наименование разрядов этого регистра. Таблица 434 перечисляет назначение разрядов регистра.

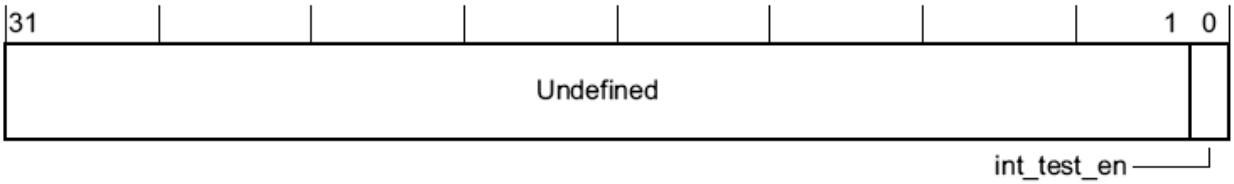


Рисунок 140. Наименование разрядов регистра integration_cfg

Таблица 434 Назначение разрядов регистра integration_cfg

Разряд	Наименование	Описание
31…1	-	Не определен. Следует записывать 0.
0	int_test_en	Разрешает работу интегрированной тестовой схемы (тестовых регистров). 0 – отключает интегрированную тестовую схему 1 – включает интегрированную тестовую схему Тестовые регистры будут определять состояние сигналов (флагов) dma_active[] dma_done[] dma_err.

Регистр состояния контроллера

Данный регистр имеет доступ только на чтение. Регистр возвращает состояния флага dma_stall не зависимо от состояния разряда int_test_en. Если контроллер находится в состоянии сброса, то чтение регистра запрещено. Рисунок 141 показывает наименование разрядов этого регистра. Таблица 435 перечисляет назначение разрядов регистра.

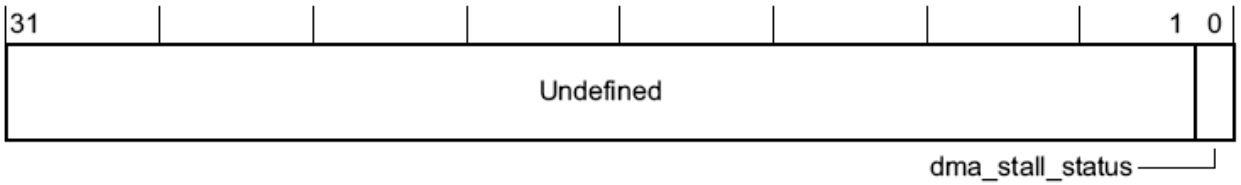


Рисунок 141. Наименование разрядов регистра dma_stall_status

Таблица 435 Назначение разрядов регистра dma_stall_status

Разряд	Наименование	Описание
31…1	-	Не определен
0	dma_stall_status	Возвращает состояния флага dma_stall При чтении: 0 = dma_stall в состоянии 0 1 = dma_stall в состоянии 1.

Регистр состояния запросов по dma_req каналов ПДП

Данный регистр имеет доступ только на чтение. Регистр возвращает состояния флагов dma_req[] не зависимо от состояния разряда int_test_en. Если контроллер находится в состоянии сброса, то чтение регистра запрещено. Рисунок 142 показывает наименование разрядов этого регистра. Таблица 436 перечисляет назначение разрядов регистра.

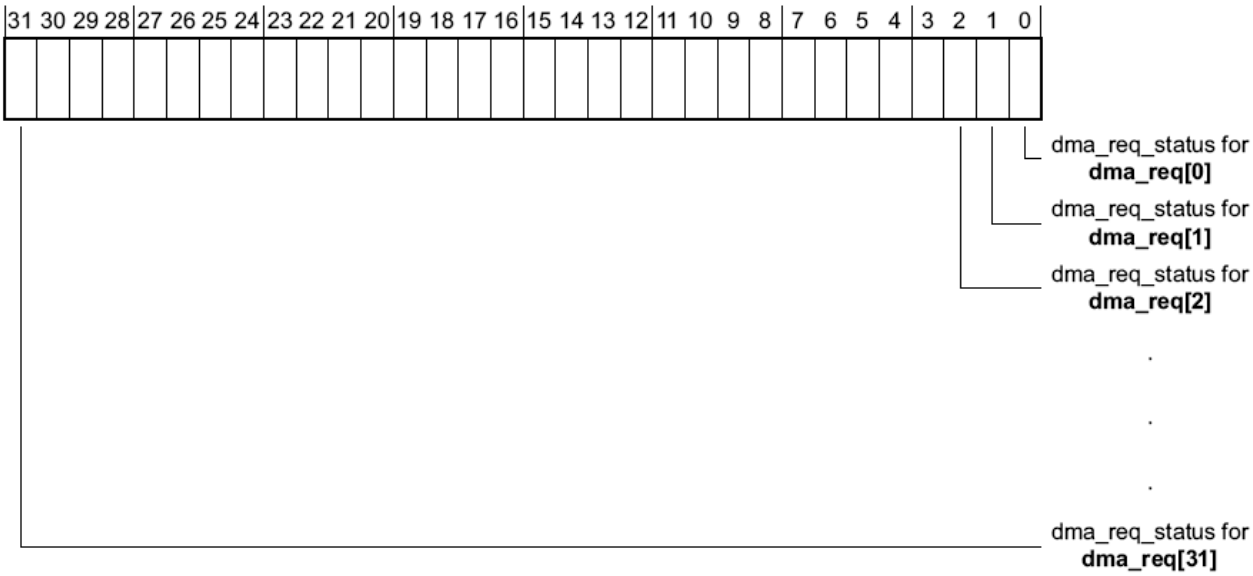


Рисунок 142. Наименование разрядов регистра dma_req_status

Таблица 436 Назначение разрядов регистра dma_req_status

Разряд	Наименование	Описание
31…0	dma_req_status	Возвращает состояния флагов запросов на выполнение цикла ПДП dma_req[] При чтении: Разряд [C] = 0 dma_req[C] в состоянии 0 Разряд [C] = 1 dma_req[C] в состоянии 1.

Регистр состояния запросов по dma_sreq каналов ПДП

Данный регистр имеет доступ только на чтение. Регистр возвращает состояния флагов dma_sreq[] не зависимо от состояния разряда int_test_en. Если контроллер находится в состоянии сброса, то чтение регистра запрещено. Рисунок 143 показывает наименование разрядов этого регистра. Таблица 437 перечисляет назначение разрядов регистра.

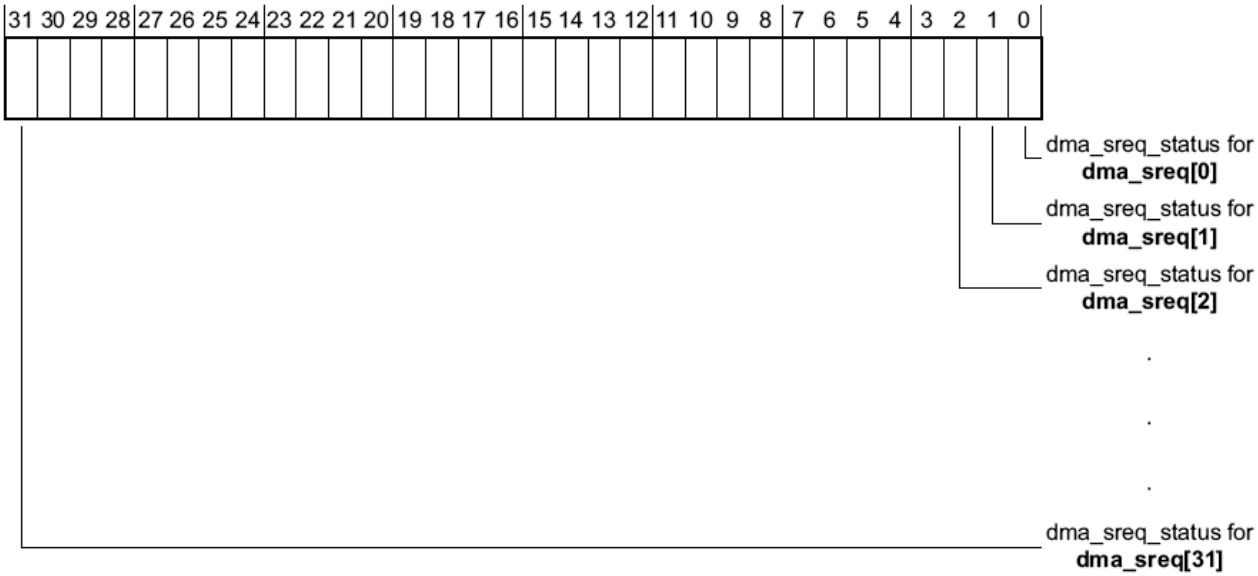


Рисунок 143. Наименование разрядов регистра dma_sreq_status

Таблица 437 Назначение разрядов регистра dma_sreq_status

Разряд	Наименование	Описание
31…0	dma_sreq_status	Возвращает состояния флагов запросов на выполнение цикла ПДП dma_sreq[] При чтении: Разряд [C] = 0 dma_sreq[C] в состоянии 0 Разряд [C] = 1 dma_sreq[C] в состоянии 1.

Регистр установки флагов окончания цикла ПДП

Данный регистр имеет доступ на запись и чтение. Регистр позволяет устанавливать сигнал (флаг) dma_done[]. Чтение регистра возвращает состояние сигналов dma_done[] . Рисунок 144 показывает наименование разрядов этого регистра. Таблица 438 перечисляет назначение разрядов регистра.

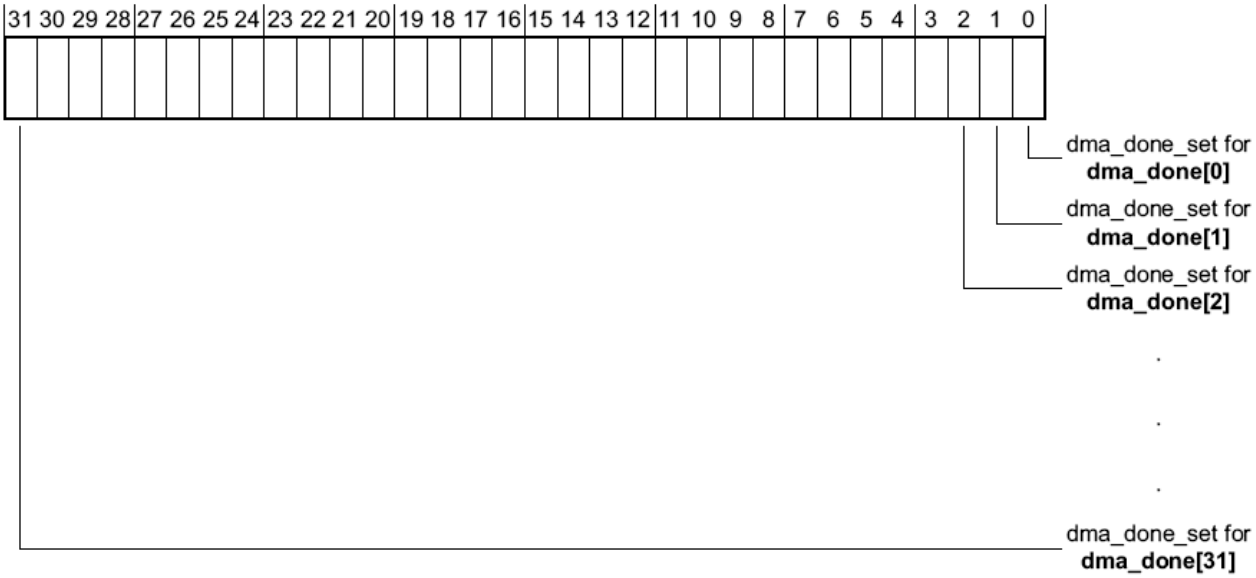


Рисунок 144. Наименование разрядов регистра dma_done_set

Таблица 438 Назначение разрядов регистра dma_done_set

Разряд	Наименование	Описание
31…0	dma_done_set	Установка сигнала (флага) dma_done[] в состояние 1, чтение возвращает состояние сигнала (флага) dma_done[] (только при int_test_en = 1). Если при чтении: Разряд [C] =0 означает, что dma_done[C] в состоянии 0. Разряд [C] =1 означает, что dma_done[C] в состоянии 1. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать dma_done _clr регистр для сброса dma_done[C] в состояние 0 Разряд [C] =1 устанавливает dma_done[] в состояние 1 (только при int_test_en = 1) Запись разряда соответствующего нереализованному каналу не дает никакого эффекта.

Регистр сброса флагов окончания цикла ПДП.

Данный регистр имеет доступ только на запись. Регистр позволяет сбросить сигнал (флаг) dma_done[] в состояние 0. Рисунок 145 показывает наименование разрядов этого регистра. Таблица 439 перечисляет назначение разрядов регистра.

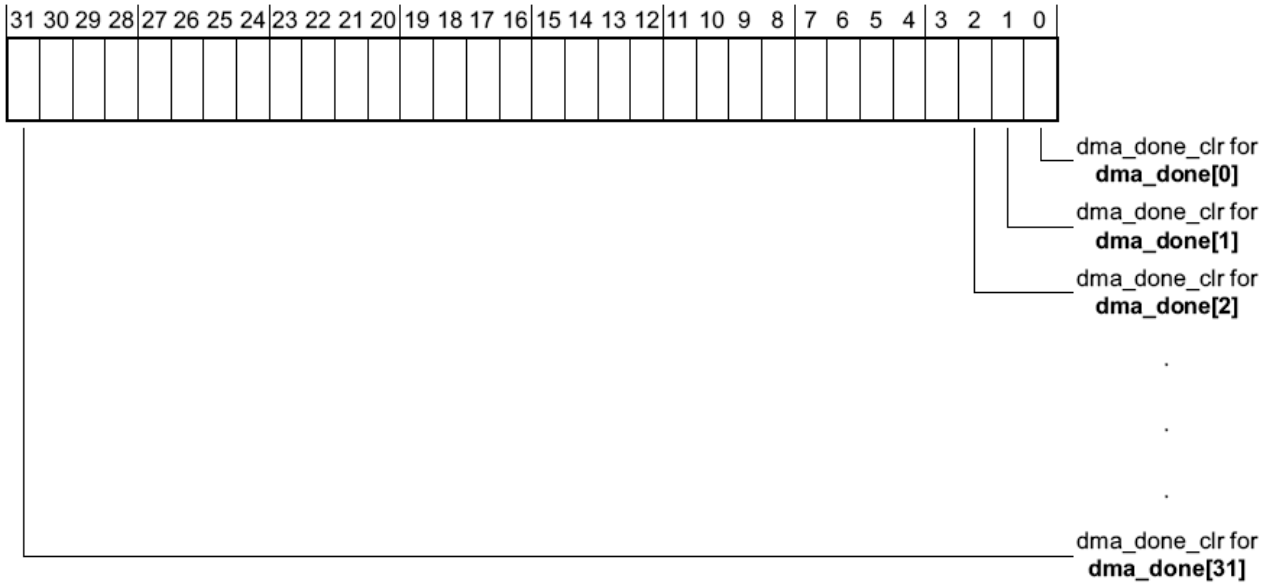


Рисунок 145. Наименование разрядов регистра dma_done_clr

Таблица 439 Назначение разрядов регистра dma_done_clr

Разряд	Наименование	Описание
31…0	dma_done_clr	Позволяет сбрасывать сигнала (флага) dma_done[] в состояние 0. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать dma_done _set регистр для установки dma_done[C] в состояние 1. Разряд [C] =1 устанавливает dma_done[] в состояние 0 (только при int_test_en = 1) Запись разряда соответствующего нереализованному каналу не дает никакого эффекта.

Регистр установки флагов активности каналов ПДП

Данный регистр имеет доступ на запись и чтение. Регистр позволяет устанавливать сигнал (флаг) dma_active[]. Чтение регистра возвращает состояние сигналов dma_active[] . Рисунок 146 показывает наименование разрядов этого регистра. Таблица 440 перечисляет назначение разрядов регистра.

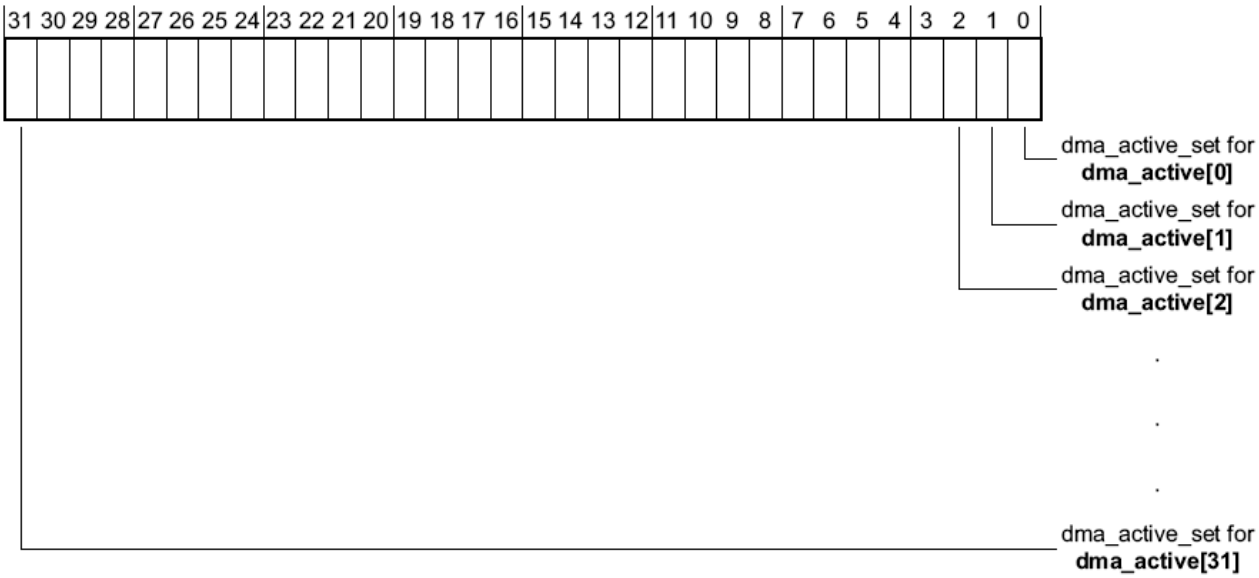


Рисунок 146. Наименование разрядов регистра dma_active_set

Таблица 440 Назначение разрядов регистра dma_active_set

Разряд	Наименование	Описание
31…0	dma_active_set	Установка сигнала (флага) dma_active [] в состояние 1, чтение возвращает состояние сигнала (флага) dma_active [] (только при int_test_en = 1). Если при чтении: Разряд [C] =0 означает, что dma_active [C] в состоянии 0. Разряд [C] =1 означает, что dma_active [C] в состоянии 1. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать dma_active _clr регистр для сброса dma_active [C] в состояние 0 Разряд [C] =1 устанавливает dma_active [] в состояние 1 (только при int_test_en = 1) Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта.

Регистр сброса флагов активности каналов ПДП

Данный регистр имеет доступ только на запись. Регистр позволяет сбросить сигнал (флаг) dma_active [] в состояние 0. Рисунок 147 показывает наименование разрядов этого регистра. Таблица 441 перечисляет назначение разрядов регистра.

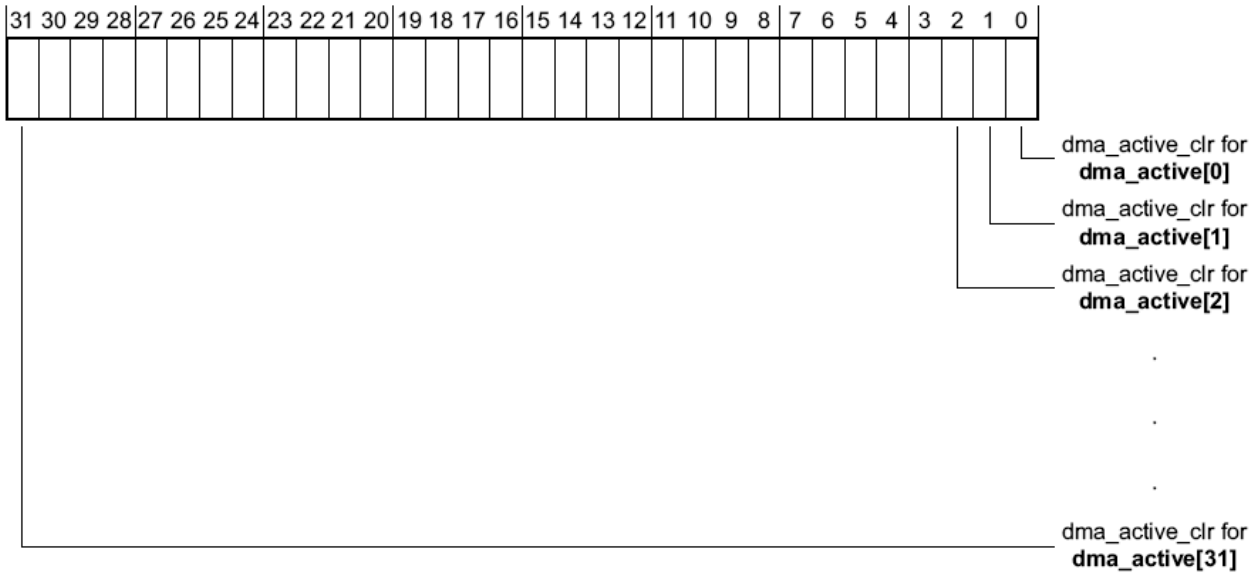


Рисунок 147. Наименование разрядов регистра dma_active _clr

Таблица 441 Назначение разрядов регистра dma_active _clr

Разряд	Наименование	Описание
31…0	dma_active_clr	Позволяет сбрасывать сигнала (флага) dma_active[] в состояние 0. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать dma_active_set регистр для установки dma_active[C] в состояние 1. Разряд [C] =1 устанавливает dma_active[] в состояние 0 (только при int_test_en = 1) Запись разряда соответствующего нереализованному каналу не дает никакого эффекта.

Регистр установки флага ошибки

Данный регистр имеет доступ только на запись. Регистр позволяет устанавливать сигнал dma_err в 1.

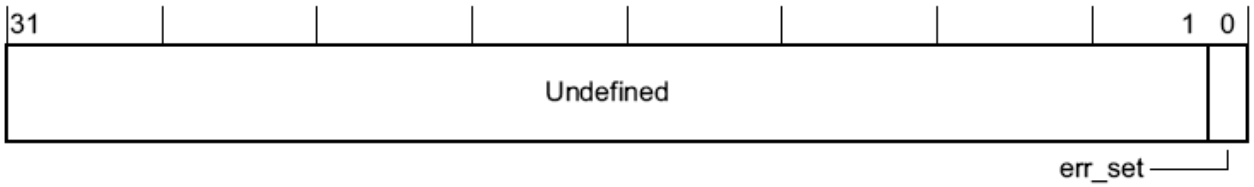


Рисунок 148 показывает наименование разрядов этого регистра. Таблица 442 перечисляет назначение разрядов регистра.

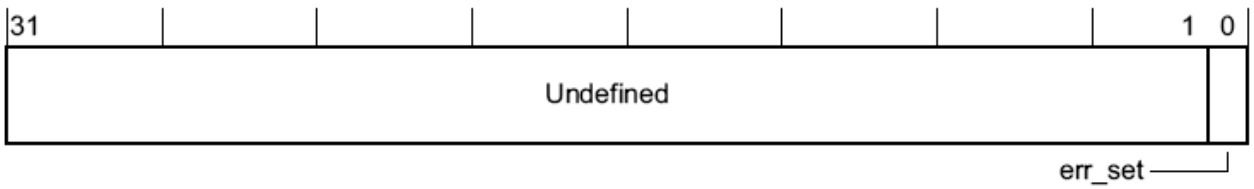


Рисунок 148. Наименование разрядов регистра err_set

Таблица 442 Назначение разрядов регистра err_set

Разряд	Наименование	Описание
31…1	-	Не определено. Следует записывать 0.
0	err_set	Установка сигнала dma_err в состояние 1. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать регистр err_clr для сброса флага dma_err в состояние 0. Разряд [C] =1 устанавливает сигнал (флаг) dma_err в состояние 1 (только при int_test_en = 1).

Таблица 443 Таблица запросов dma_req и dma_sreq от периферийных блоков

Номер запроса	dma_req	dma_sreq	Описание
0	UART1TXDMABREQ	UART1TXDMASREQ	Запрос от передатчика UART1
1	UART1RXDMABREQ	UART1RXDMASREQ	Запрос от приёмника UART1
2	UART2TXDMABREQ	UART2TXDMASREQ	Запрос от передатчика UART2
3	UART2RXDMABREQ	UART2RXDMASREQ	Запрос от приёмника UART2
4	SSP1TXDMABREQ	SSP1TXDMASREQ	Запрос от передатчика SPI
5	SSP1RXDMABREQ	SSP1RXDMASREQ	Запрос от приёмника SPI
6	CRCDMABREQ	CRCDMASREQ	Запрос от аппаратного блока вычисления CRC
7	-	-	-
8	-	-	-
9	-	-	-
10	TIM1DMAREQ	TIM1DMAREQ	Запрос от таймера общего назначения TIMER1
11	TIM2DMAREQ	TIM2DMAREQ	Запрос от таймера общего назначения TIMER2
12	ADCIUDMABREQ1	ADCIUDMASREQ1	Запросы от блока АЦП для измерения напряжений и токов в электрической сети
13	ADCIUDMABREQ2	ADCIUDMASREQ2	
14	ADCIUDMABREQ3	ADCIUDMASREQ3	
15	ADCIUDMABREQ4	ADCIUDMASREQ4	
16	ADCIUDMABREQ5	ADCIUDMASREQ5	
17	ADCIUDMABREQ6	ADCIUDMASREQ6	
18	ADCIUDMABREQ7	ADCIUDMASREQ7	
19	-	-	-
20	-	-	-
21	-	-	-
22	-	-	-
23	-	-	-
24	-	-	-
25	-	-	-
26	-	-	-
27	-	-	-
28	-	-	-
29	-	-	-
30	-	ADCDMASREQ	Запрос от АЦП последовательных приближений

31	-	-	-

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 – исключении начало обрабатываться процессором, но появилось новое исключении 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 специальную форму исключения. Когда выставляется RESET, работа процессора останавливается потенциально в любой точке инструкций. Когда RESET убирается, выполнение перезапускается с адреса заданного 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 444 Таблица различных типов исключений

Номер исключения	IRQ номер	Тип	Приоритет	Адрес вектора обработчика (смещение)	Активация
1	-	RESET	-3, наивысщий	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

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, что указывает на то, что обработчик выполняется в Thumb режиме.

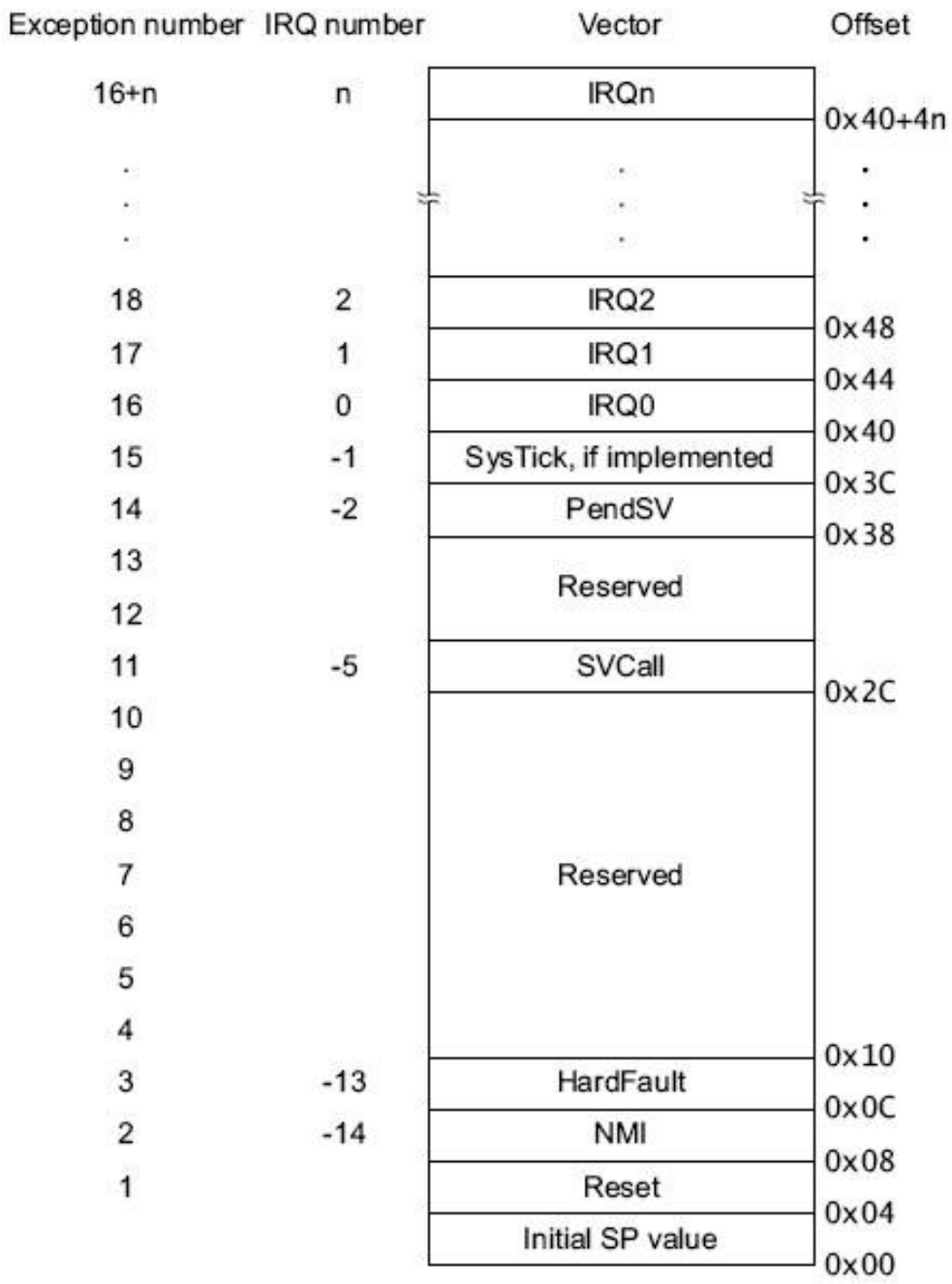


Рисунок 149.

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. Подробнее данный вопрос рассмотрен в разделе «Вход в процедуру обработки исключения».

Возврат

Возврат из обработчика осуществляется по завершении обработки исключительной ситуации, с одновременным выполнением следующих условий:

- в системе отсутствуют необработанные исключения с достаточным приоритетом;
- завершенный обработчик не обрабатывал запоздавшее исключение (late-arriving exception).

Процессор обращается к стеку и восстанавливает состояние, имевшее место до вызова обработчика. Более подробная информация дана в разделе «Возврат из обработчика исключения».

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 низко-приоритетный 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 (см. «Регистры маскирования исключений»). 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 значение 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».

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 содержать 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 445.

Процессор устанавливает биты EXC_RETURN [31:4] в 0xFFFFFFF. Загрузка данного значения в PC указывает на завершение процедуры обработки исключения и заставляет процессор выполнить необходимые действия для возврата из обработчика.

Таблица 445 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. “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. Более подробная информация представлена на стр. 165.

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.

Более подробная информация представлена на стр. 164.

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”, стр. 64.

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”, стр. 188.

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 446 Прерывания, формируемые периферийными блоками

Прерывания	**Блок**	**Принцип формирования**
IRQ0	DMA	Прерывания от DMA DMA_ERR или DMA_DONE. Обработка прерываний от DMA в соответствии с разделом Error signaling технического описания DMA
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	-	-
IRQ10	BACKUP	Прерывание от BKP и часов реального времени
IRQ11	Внешнее прерывание 1	Сигнал EXT_INT0 Вывод PA[10] в основном режиме
IRQ12	Внешнее прерывание 2	Сигнал EXT_INT1 Вывод PC[4], PB[6] в альтернативном

		режиме
IRQ13	Внешнее прерывание 3	Сигнал EXT_INT2 Вывод PC[5], PB[7] в альтернативном режиме
IRQ14	ADCIU	Прерывание от АЦП для измерения напряжений и токов
IRQ15	ADCIU	Прерывание от АЦП для измерения напряжений и токов
IRQ16	ADCIU	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 447.

Таблица 447 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	IPR0-7	RW	0x00000000	Регистр приоритета прерываний

Регистр разрешения прерываний

Регистр ISER предназначен для разрешения прерываний (запись) и определения, какие из прерываний разрешены (чтение).

Распределение бит регистра представлено на рис.:

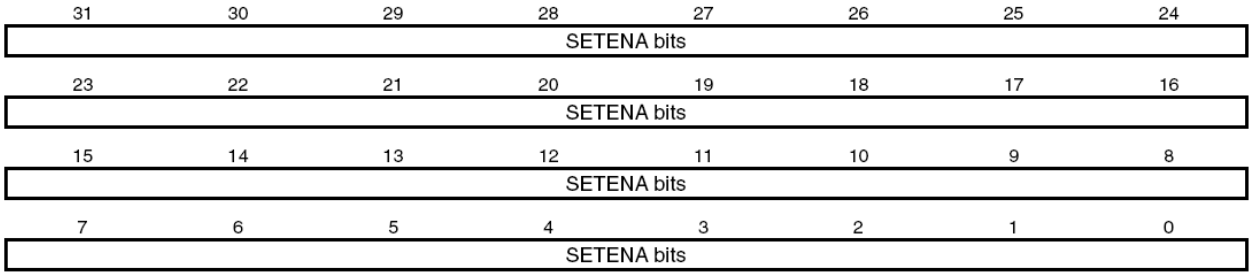


Рисунок 150. Распределение бит регистра

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).

Распределение бит регистра представлено на рис.:

Рисунок 151. Распределение бит регистра

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).

Распределение бит регистра представлено на рис.:

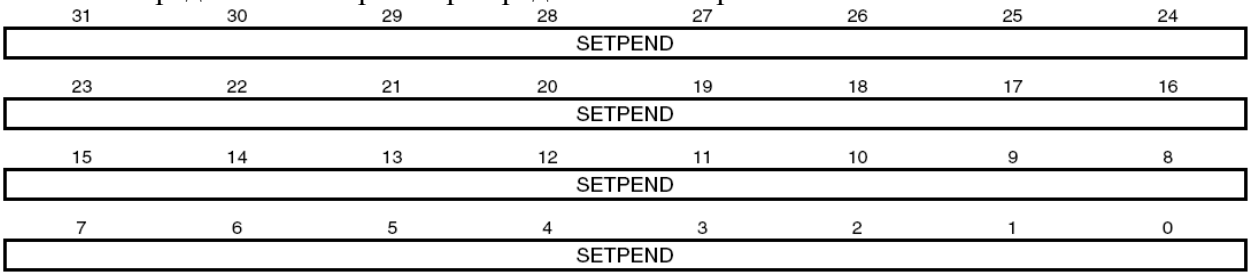


Рисунок 152. Распределение бит регистра

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).

Распределение бит регистра представлено на рис.:

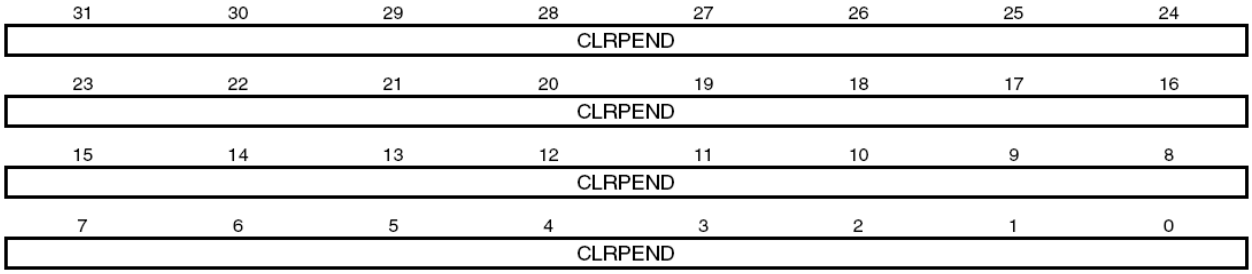


Рисунок 153. Распределение бит регистра

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 пословно.

Каждый из регистров содержит четыре поля приоритета, которые отображаются на четыре элемента массива PRI[0] .. PRI[31] CMSIS, как показано ниже.

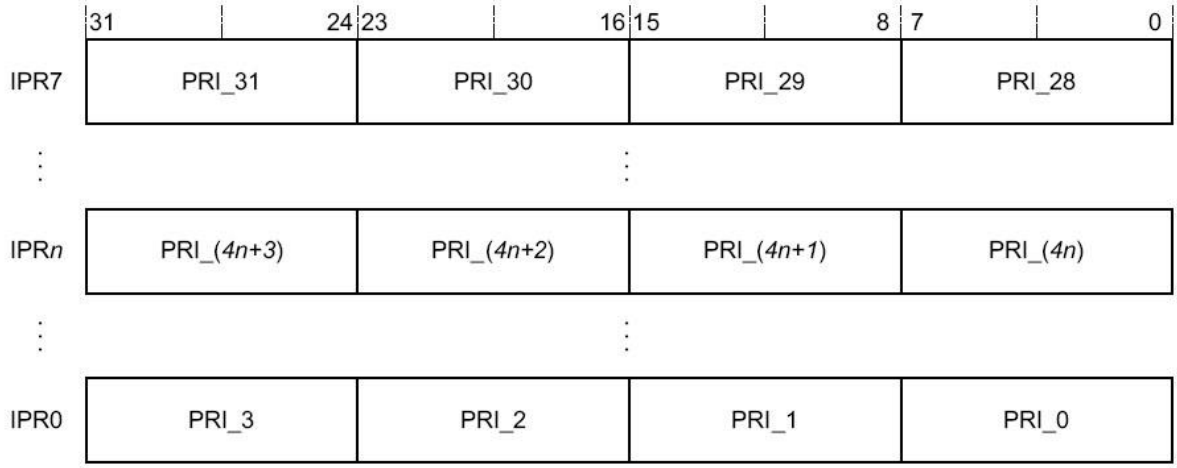


Рисунок 154. Поля приоритета регистров IPR0-IPR7

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 DIV 4;

смещение данных в регистре в зависимости от значения N 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 “Аппаратное и программное управление прерываниями”). 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 (см. “Регистр перевода прерывания в состояние ожидания обслуживания”) или соответствующего значения в регистр STIR (см. “Регистр программного формирования прерывания”).

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 448 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 449.

Таблица 449 Обобщенна информация о регистрах блока управления системой

Адрес	Имя	Тип	Значение после сброса	Описание
0xE000ED00	CPUID	RO	0x412FC230	Регистр идентификации процессора, стр. 181
0xE000ED04	ICSR	RW	0x00000000	Регистр управления прерываниями, стр. 182
0xE000ED0C	AIRCR	RW	0xFA050000	Регистр управления прерываниями и программного сброса, стр. 186
0xE000ED10	SCR	RW	0x00000000	Регистр управления системой, стр. 188
0xE000ED14	CCR	RW	0x00000200	Регистр конфигурации и управления, стр. 189
0xE000ED1C	SHPR2	RW	0x00000000	Регистр №2 приоритета системных обработчиков, стр. 193
0xE000ED20	SHPR3	RW	0x00000000	Регистр №3 приоритета системных обработчиков, стр. 194

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 Таблице 13-30, стр. 179.

Назначение разрядов регистра представлено на рис.:

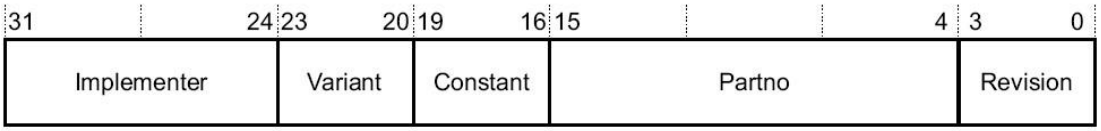


Рисунок 155. Назначение разрядов регистра

Implementer – код разработчика 0x41 = ARM.
Variant – значение r в номере версии rnpn изделия: 0x0 = r0p0;
Constant – постоянное значение 0xC;
PartNo – номер модели процессора: 0xC20 = Cortex-M0;
Revision – значение p в номере версии rnpn 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.

Назначение разрядов регистра представлено на рис.:

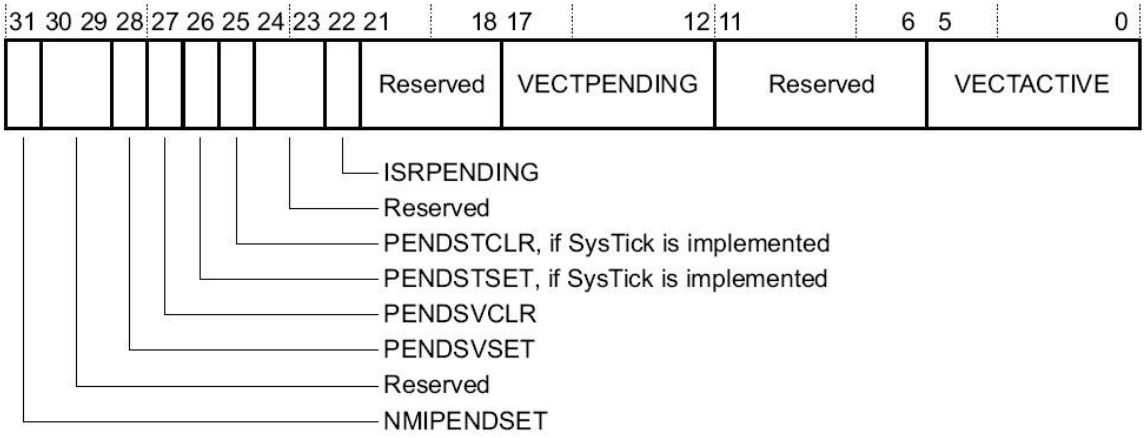


Рисунок 156. Назначение разрядов регистра

NMIPENDSET (RW) – так как NMI имеет самый высокий приоритет, процессор переход 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.

ISRPENDING (RO) – флаг наличия в системе прерываний (за исключением отказов), ожидающих обслуживания. 0 – ожидающие обслуживания прерывания отсутствуют, 1 – присутствуют.

VECTPENDING (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.

VECTACTIVE (RO) – содержит номер активного исключения. 0 – режим приложения, другое число – номер текущего обслуживаемого исключения. Для получения номера запроса прерывания (IRQ) из значения VECTACTIVE 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.

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.:

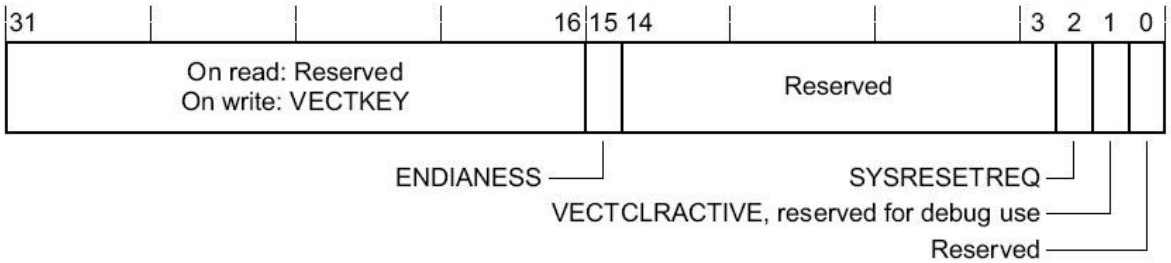


Рисунок 157. Назначение разрядов регистра

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.

Назначение разрядов регистра представлено на рис.:

Рисунок 158. Назначение разрядов регистра

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 (см. “Регистр программного формирования прерывания”).

Назначение разрядов регистра представлено на рис.:

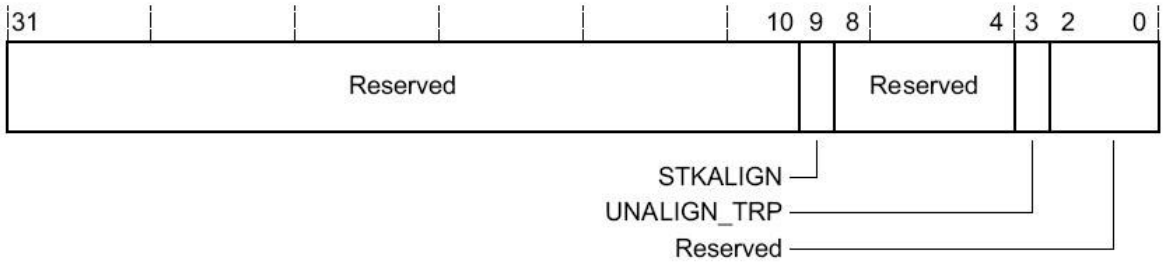


Рисунок 159. Назначение разрядов регистра

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.

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 450 Поля приоритета обработчиков системных отказов

Обработчик отказа	Поле	Описание регистра
Вызов SVCall	PRI_11	Регистр №2 приоритета системных обработчиков, стр. 193
Вызов PendSV	PRI_14	Регистр №3 приоритета системных обработчиков, стр. 194
Вызов SysTick	PRI_15	

Регистр №2 приоритета системных обработчиков

Назначение разрядов регистра представлено на рис.:

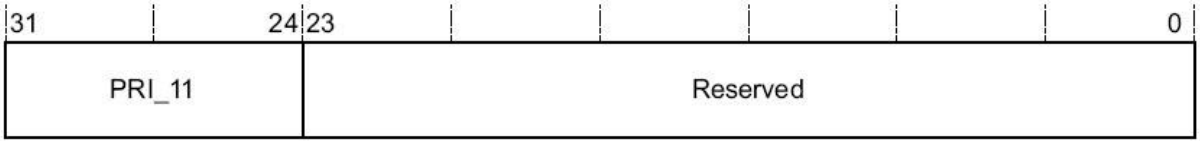


Рисунок 160. Назначение разрядов регистра

PRI_11 Приоритет системного обработчика 11, вызов SVCall.

Регистр №3 приоритета системных обработчиков

Назначение разрядов регистра представлено на рис.:

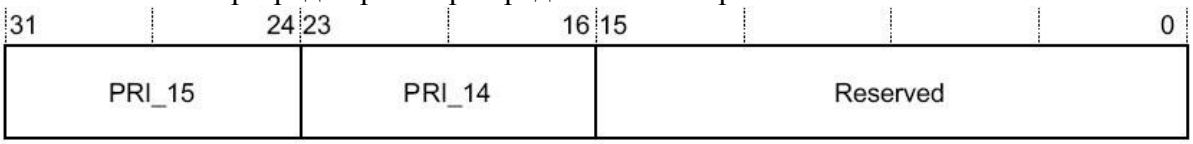


Рисунок 161. Назначение разрядов регистра

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 451 Описание регистров блока сторожевых таймеров

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 452 Описание регистров блока сторожевых таймеров

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

Таблица 453 Регистр IWDG_KR

Номер	31…16	15…0
Доступ		W
Сброс		0
	-	**KEY[15:0]**

Таблица 454 Описание бит регистра IWDG_KR

№	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…16		Зарезервировано
15...0	KEY[15:0]	**Значение ключа (только запись, читается 0000h).** 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 CCCCh разрешает работу сторожевого таймера (за исключением, если сторожевой таймер разрешается аппаратно битами конфигурации).

IWDG_PR

Таблица 455 Регистр IWDG_PR

Номер	31…3	2	1	0
Доступ	U	R/W	R/W	R/W
Сброс		0	0	0
	-	**PR2**	**PR1**	**PR0**

Таблица 456 Описание бит регистра IWDG_PR

№	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...3		Зарезервировано
2...0	PR[2:0]	**Делитель частоты сторожевого таймера.** 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

Таблица 457 Регистр IWDG_RLR

Номер	31…12	11…0
Доступ	U	R/W
Сброс		1
	-	RLR[11:0]

Таблица 458 Описание бит регистра IWDG_RLR

№	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...12		Зарезервировано
11...0	RLR[11:0]	**Значение перезагрузки сторожевого таймера.** Значение этих бит 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

Таблица 459 Регистр IWDG_SR

Номер	31…2	1	0
Доступ	U	R	R
Сброс		0	0
	-	**RVU**	**PVU**

Таблица 460 Описание бит регистра IWDG_SR

№	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…2		Зарезервировано
1	RVU	**Флаг обновления значения сторожевого таймера.** 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	**Флаг обновления делителя частоты сторожевого таймера.** 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

Таблица 461 Регистр WWDG_CR

Номер	31…8	7	6…0
Доступ	U	R/S	R/W
Сброс		0	1
	-	WDGA	T6…T0

Таблица 462 Описание бит регистра WWDG_CR

№	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...18		Зарезервировано
7	WDGA	**Бит активации.** 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]	**Значение семиразрядного счётчика (от старших разрядов к младшим).** Эти биты содержат значение сторожевого таймера, который декрементируется каждые $4096x2^{WDGTB}$ циклов частоты PCLK периферийной шины APB

WWDG_CFR

Таблица 463 Регистр WWDG_CFR

Номер	31…10	9	8	7	6…0
Доступ	U	R/S	R/W	R/W	R/W
Сброс		0	0	0	1
	-	**EWI**	**WDGTB1**	**WDGTB0**	**W6…W0**

Таблица 464 Описание бит регистра WWDG_CFR

№	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...10		Зарезервировано
9	EWI	**Раннее предупреждающее прерывание.** Если бит установлен, то разрешается генерация прерывания при достижении сторожевым таймером значении 40h. Прерывание запрещается только аппаратным сбросом.
8..7	WGTB[1:0]	**Делитель частоты сторожевого таймера.** 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]	**Значение окна.** 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

Таблица 465 Регистр WWDG_SR

Номер	31…1	0
Доступ	U	R/C
Сброс		0
	-	**EWIF**

Таблица 466 Описание бит регистра WWDG_SR

№	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...1		Зарезервировано
0	EWIF	**Флаг раннего предупреждающего прерывания.** 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 467 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}	3,0	3,6	–	4,0
Напряжение источника питания при использовании АЦП, В	U_{CCA}	3,0	3,6		4,0
Напряжение источника питания батарейного домена, В	U_{CCB}	1,8	3,6	–	4,0
Входное напряжение высокого уровня, В, на выводах: PA(5), PB(4-9), PC (1-6)	U_{IH}	2,0	U_{CC}	–	U_{CC}+0,3
на выводе OSC_IN при HSE BYPASS=1			U_{CC}	–	U_{CC}+0,3
на выводах: PA(0-4, 6-15), PB(0-3,10-14), PC(0, 7), RESET, WAKEUP, JTAG_EN			5,25	–	5,3
Входное напряжение низкого уровня, В, при работе в цифровом режиме на выводах: PA, PB, PC, RESET, WAKEUP, JTAG_EN	U_{IL}	0	0,8	–0,3	–
на выводе: OSC_IN при HSE BYPASS=1			0,8	–0,3	–
Выходной ток высокого уровня, мА, при работе в цифровом режиме на выводах: PA, PB, PC	I_{OH}	–6	–	–10	–
Выходной ток низкого уровня, мА на выводах: PA, PB, PC	I_{OL}	–	6	–	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		
Частота следования импульсов тактовых сигналов PLL, МГц	f_{C_PLL}	2	16	–	–
Емкость нагрузки, пФ, на выводах: PA, PB, PC	C_L	–	30	–	–

Наименование параметра, единица измерения	**Буквенное обозначение параметра**	**Норма параметра**			
		Предельно-допустимый режим		**Предельный режим**	
		не менее	**не более**	**не менее**	**не более**
Время хранения информации, лет, при T=25 °C	t_{GS}	25	–	–	–
при T=85 °C		10	–	–	–
Параметры АЦП					
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_{ADC1_REF-} = AGND, U_{ADC0_REF+} = AU_{CC}	U_{AIN}	U_{ADC1_REF-}	U_{ADC0_REF+}	–0,3	4,0
Частота следования импульсов тактовых сигналов АЦП, МГц	$f_{C\_ADC\_S}$	–	14	–	–
Параметры ∑Δ АЦП					
Частота следования импульсов тактовых сигналов ∑ΔАЦП, МГц	$f_{C\_ADC\_D}$	–	8,196	–	–
Амплитуда входного дифференциального сигнала Δ∑АЦП, В	A_{NADC_D}	–	1	–	–
Напряжение на входе ∑ΔАЦП, В	U_{IADC_D}	–0,5	0,5	–0,8	U_{CC}+0,3
П р и м е ч а н и е – Не допускается одновременное задание двух предельных режимов					

Электрические параметры микросхемы

Таблица 468 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 высокого уровня, В, на выводах PA, PB, PC	U_{OH}	2,4	–	25, 85, – 40
Выходное напряжение низкого уровня, В, на выводах PA, PB, PC	U_{OL}	–	0,4	
Напряжение срабатывания схемы генерации сброса, В	U_{BOR}	1,8	2,1	
Входной ток утечки высокого уровня, мкА, (при работе в цифровом режиме) на выводах: PA, PB, PC, RESET, WAKEUP	I_{ILH}	– 1,0	1,0	
на выводе: JTAG_EN		– 180	180	
на выводе OSC_IN		– 40	40	
Ток утечки низкого уровня цифровых входов, мкА, (при работе в цифровом режиме) на выводах: PA, PB, PC, RESET, WAKEUP, JTAG_EN	I_{ILL}	– 1	1	25, 85, – 40
на выводе OSC_IN		– 40	40	
Статический ток потребления, мкА при выключенном стабилизаторе напряжения	I_{CCS}	–	10	25, 85, – 40
при включенном стабилизаторе напряжения		–	50	
Динамический ток потребления, мА, при 33 кГц<fc≤36 МГц	I_{OCC}	–	20	25, 85, – 40
при 32 ≤fc ≤33 кГц		–	0,5	
Выходная частота HSI RC-генератора, МГц	f_{O_HSI}	6	10	25, 85, – 40
Выходная частота LSI RC-генератора, кГц	f_{O_LSI}	10	50	
Выходная частота PLL, МГц максимальная	f_{O_PLL}	36	–	25, 85, – 40
минимальная		–	2	
Параметры АЦП последовательного приближения				
Разрядность АЦП	E_{NADC}	12	–	25, 85, – 40
Дифференциальная нелинейность, единица младшего разряда	E_{DLADC}	–1	2	
Интегральная нелинейность, единица младшего разряда	E_{ILADC}	–3	3	
Ошибка смещения, единица младшего разряда	E_{OFFADC}	–6	6	
Ошибка усиления, %	$E_{GAINADC}$	–1	1	

Наименование параметра, единица измерения, режим измерения	**Буквенное обозначение параметра**	**Норма параметра**		**Температура среды, °С**
		не менее	**не более**	
Параметры ∑Δ АЦП				
Выходное напряжение VR_1V, В	U_{OBIAS}	0,96	1,01	25, 85, – 40
Соотношение сигнал / шум, дБ усиление 0дБ, A_{NADC_D} = 1В, f_{NADCO} = 4КГц	$SNR_{\_D\_0}$	77	–	
Соотношение сигнал / шум, дБ усиление +6дБ, A_{NADC_D} = 0,5В, f_{NADCO} = 4КГц	$SNR_{\_D\_6}$	74	–	
Соотношение сигнал / шум, дБ усиление +12дБ, A_{NADC_D} = 0,25В, f_{NADCO} = 4КГц	$SNR_{\_D\_12}$	71	–	
Соотношение сигнал / шум, дБ усиление +18дБ, A_{NADC_D} = 0,125В, f_{NADCO} = 4КГц	$SNR_{\_D\_18}$	70	–	
Ошибка усиления предусилителя, дБ	$GAIN_{ERR}$	–	0,25	

Справочные данные

Таблица 469 Справочные параметры микросхем

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды,°С
		не менее	не более	
Ток потребления батарейного домена, мкА, при: U_{CC}= 0 В Bucc = 3,6В	I_{CCB}	–	5	25, 85, – 40
Время установления сигналов PBD и PBVD, мкс	$t_{SU(PBD)}$ $t_{SU(PBVO)}$	–	2	
Гистерезис портов ввода/вывода, мВ, на выводах: PA-PC при: ModeRX = 0 ModeRX = 1	$\Delta U_{TH(PA-PF)}$	 100 200	 400 500	
на выводах: PA – PC при: U_{CC}=2,2 В, PowerTX=00, C_l= 50 пФ PowerTX=01, C_l= 50 пФ PowerTX=10, C_l= 50 пФ PowerTX=11, C_l= 50 пФ PowerTX=11, C_l= 30 пФ	$t_{W(PA-PF)}$	 – – – – –	 10 100 20 10 5	
Тактовые частоты и генераторы				
Время установления сигнала HSIRDY относительно HSION, мкс, при: U_{CC} = 2,2 В	$t_{SU(HSI)}$	–	1	25, 85, –40
Время установления сигнала LSIRDY относительно LSION, мс, при: U_{CC} = 2,2 В	$t_{SU(LSI)}$	–	80	
Время установления сигнала HSERDY относительно HSEON, мкс, при: U_{CC} = 2,2 В	$t_{SU(HSE)}$	–	$2048/f_{C\_HSE}$	
Время установления сигнала LSERDY относительно LSEON, мкс, при: U_{CC} = 2,2 В	$t_{SU(LSE)}$	–	$4096/f_{C\_LSE}$	
Время установления сигнала PLLRDY относительно PLLON, мкс, при: U_{CC} = 2,2 В	$t_{SU(PLL)}$	–	100	
Длительность сигнала сброса, мкс, при: U_{CC} = 2,2 В	$t_{W(RESET)}$	20	–	
Время запуска после сброса по POR, мс	t_{POR}	–	6	
АЦП				
Время выборки заряда АЦП, нс, при: Ucc = 3,6 В	t_{A_ADC}	–	$4\times f_{C\_ADC}$	25, 85, –40
Время преобразования АЦП, нс при: U_{CC} = 3,6 В	t_{AO_ADC}	–	$28\times f_{C\_ADC}$	

Наименование параметра, единица измерения, режим измерения	**Буквенное обозначение параметра**	**Норма параметра**		**Температура среды, °С**
		не менее	**не более**	
Ток потребления по входу внешней верхней границы опоры АЦП, мкА при: ADC1_Cfg_M_REF=1 или ADC2_Cfg_M_REF=1	I_{ADC0_VREF+}	–	50	25, 85, –40
Ток потребления по входу внешней нижней опоры опоры АЦП, мкА при: ADC1_Cfg_M_REF=1 или ADC2_Cfg_M_REF=1	I_{ADC0_VREF-}	-50	–	
Ток потребления по питанию АЦП, мА при: fadc=14 МГц, U_{CCA}=3,6В	I_{OCCADC}	–	3	
Минимальная частота преобразования АЦП, кГц	f_{C_ADCMIN}	10	–	
ΔΣ АЦП				
Выходная частота дискретизации, кГц	f_{NADCO}	–	4, 8, 16	25, 85, –40
Входное сопротивление, кОм	R_{NADC_D}	30	–	

Габаритный чертеж микросхемы

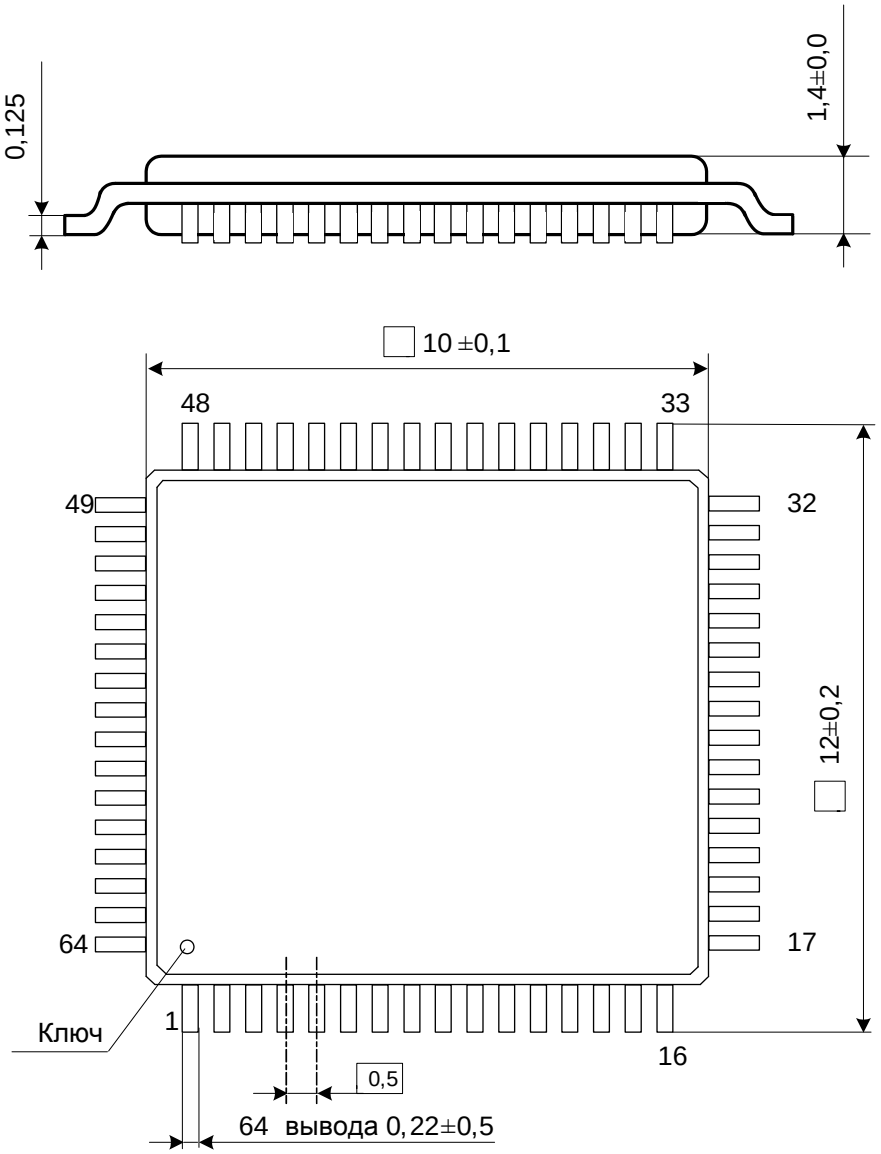


Рисунок 162. Корпус LQFP64

Информация для заказа

Обозначение	Маркировка	Тип корпуса	Температурный диапазон
К1986ВК234	MDR32F21QI	LQFP64	минус 40 – 85 °С
К1986ВК234К	MDR32F21QC	LQFP64	0 – 70 °С

Лист регистрации изменений

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
1	24.09.12	1.1.0	Введена впервые	
2	10.06.13	1.2.0	Добавлена структурная схема. Правки по тексту.	5
3	23.07.2013	1.2.1	Правки по тексту	
4	27.12.2013	1.3.1	Исправлена разрядность сигма-дельта АЦП. Перевод надписей на рисунках.	4, По тексту
5	15.01.2014	1.4.1	Исправление заголовков разделов.	24, 25
6	21.01.2014	1.5.1	Внесение исправлений в таблицу 3	16
7	04.03.2014	1.6.1	Исправление в табл.259 (бит 0)	170
8	28.05.2014	1.7.1	Добавлено примечание 2 к таблице 467	380
9	01.07.2014	1.8.0	На рисунке исправлена маркировка микросхемы в корпусе LQFP64, исправлена ориентация корпуса	1
10	12.01.2015	1.9.0	Добавлены типовые схемы включения для учета электроэнергии	152 – 155
11	26.06.2015	1.10.0	Изменено обозначение микросхем. Удален металлокерамический корпус	По тексту
12	01.10.2015	1.11.0	Изменено обозначение микросхем	По тексту
13	13.10.2015	1.12.0	Исправлено функциональное назначение Исправлено значение напряжения источника питания Исправлена таблица электрических параметров	1 По тексту 389, 390
14	02.03.2017	2.0.0	Изменено обозначение микросхем Таблицы параметров приведены в соответствие с ТУ	По тексту 387 – 392