

## LGT8F88P, LGT8F168P, LGT8F328P

### 1.Микроконтроллер RISC с Программируемой внутренней Флэш-памятью

**LGT8FX8P Series – Programming Manual 1.0.1 (От переводчика: Переводил для себя, поэтому некоторые моменты не редактированы.)**

#### 1.1. Обзор функций

- Высокопроизводительное маломощное 8-битное ядро LGT8XM
- Расширенная архитектура RISC
- 131 инструкция, более 80% для однотактной реализации.
- 32x8 рабочий регистр общего назначения.
- 32MHz работают до 32MIPS эффективности исполнения.
- Внутренний однотактный множитель (8x8)
- Энергонезависимая память программ и пространство для хранения данных.
  - Микросхема 32 Кбайт может быть запрограммирована в режиме онлайн.
  - Внутренняя память данных SRAM 2 Кбайт.
  - Программируемый аналоговый интерфейс E2PROM, поддержка доступа к байтам.
  - Новый алгоритм шифрования программ для обеспечения безопасности кода пользователя.
  - Два 8-битных таймера с независимой поддержкой предварительного делителя, выходной режим сравнения.
  - Два 16-разрядных таймера с независимой поддержкой предварительного блока для ввода и сравнения входных данных.
  - Встроенный 32KHz калибровочный RC-генератор для функции счетчика в реальном времени.
  - Может поддерживать до 9 выходов PWM, 4 программируемых управления мертвый зоной.
  - 12-канальный 12-битный высокоскоростной аналого-цифровой преобразователь (АЦП)
    - дополнительный внутренний, внешний источник опорного напряжения.
    - Каналы дифференциального усиления с программируемым усилением ( $x1/8/16/32$ ).
  - Режим автоматического контроля порогового напряжения.
  - Два аналоговых компаратора (AC), поддержка расширения от входного канала АЦП.
  - Внутренние напряжения калиброванных значений - 1.024V / 2.048V / 4.096V  $\pm 1\%$ .
  - 8-разрядный программируемый ЦАП, который генерирует источник опорного напряжения.
  - Программируемый сторожевой таймер (WDT).

- Программируемый синхронный / асинхронный последовательный интерфейс (USART / SPI).
  - Синхронный периферийный интерфейс (SPI), программируемый режим работы ведущий / ведомый.
  - Двухпроводный последовательный интерфейс (TWI), совместимый с ведущим-ведомым режимом I2C.
  - 16-разрядный цифровой арифметический ускоритель (DSC), который поддерживает прямой 16-разрядный доступ к данным.
  - Специальная функция процессора
- SWD Dual On-chip Debug / Production Interface.
- Поддержка внешнего прерывания и поддержка прерывания уровня ввода-вывода.
- Встроенная схема сброса питания (POR) и программируемая схема обнаружения низкого напряжения (LVD).
- Встроенный 1% может быть откалиброван 32-мегагерцовый генератор RC, поддержка многочастотного выхода.
- Встроенный 1% может быть откалиброван 32KHz RC генератор.
- Внешняя поддержка 32,768KHz и 400K ~ 32MHz вход кристалла.
- 6x на большой ток двухтактный привода ИО, поддержка высокоскоростных приложений ШИМ
- I / O и пакет: QFP48 / 32L
  - Наименьшее энергопотребление: 1uA @ 3,3V
  - Рабочая среда
- Рабочее напряжение: 1,8 В ~ 5,5 В
- Рабочая частота: 0 ~ 32 МГц
- Рабочая температура: -40C ~ + 85C
- HBM ESD:> 4KV

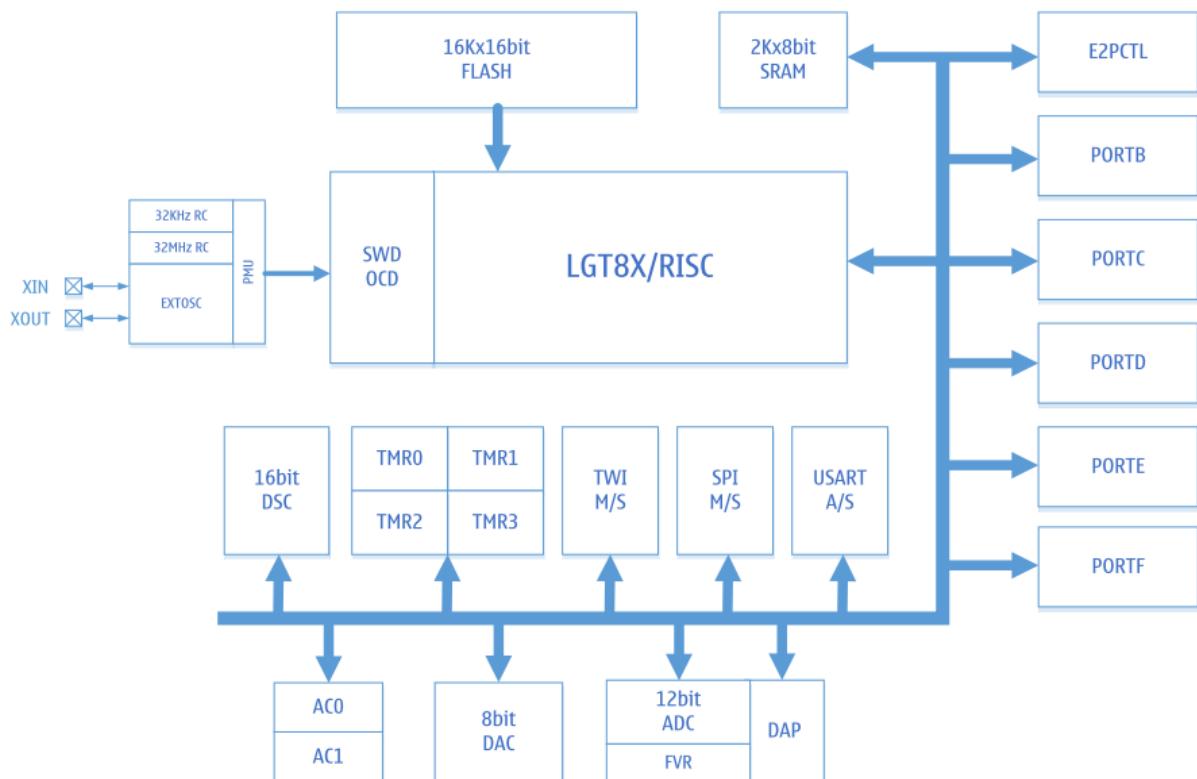


Рис1.1. Структура системы.

Имя модуля	Функция модуля
SWD	Отлаживать модуль, осуществлять онлайн-отладку и функцию ISP одновременно
LGT8X	8-разрядное высокопроизводительное ядро RISC
E2PCTL	Контроллер доступа FLASH данных
PMU	Модуль управления питанием, ответственный за управление переходом между состояниями системы
PORTB/C/D/E/F	Универсальные программируемые входные и выходные порты
DSC	16-разрядный цифровой арифметический ускоритель
ADC	8-канальный 12-разрядный аналого-цифровой преобразователь
DAP	Программируемый дифференциальный усилитель с усилением
IVREF	1.024V / 2.048V / 4.096V Внутренние опорные напряжения
ACO/1	Аналоговый компараторы
TMR0/1/2/B	8/16-разрядный таймер / счетчик, PWM-контроллер
WDT	Модуль сброса сторожевого таймера
SPI M/S	Ведущий - ведомый контроллер SPI
TWI M/S	Управляемый двухпроводным интерфейсом контроллер, совместимый с протоколом I2C
USART	Синхронный/асинхронный последовательный приемопередатчик
DAC	8-разрядный цифро-аналоговый преобразователь

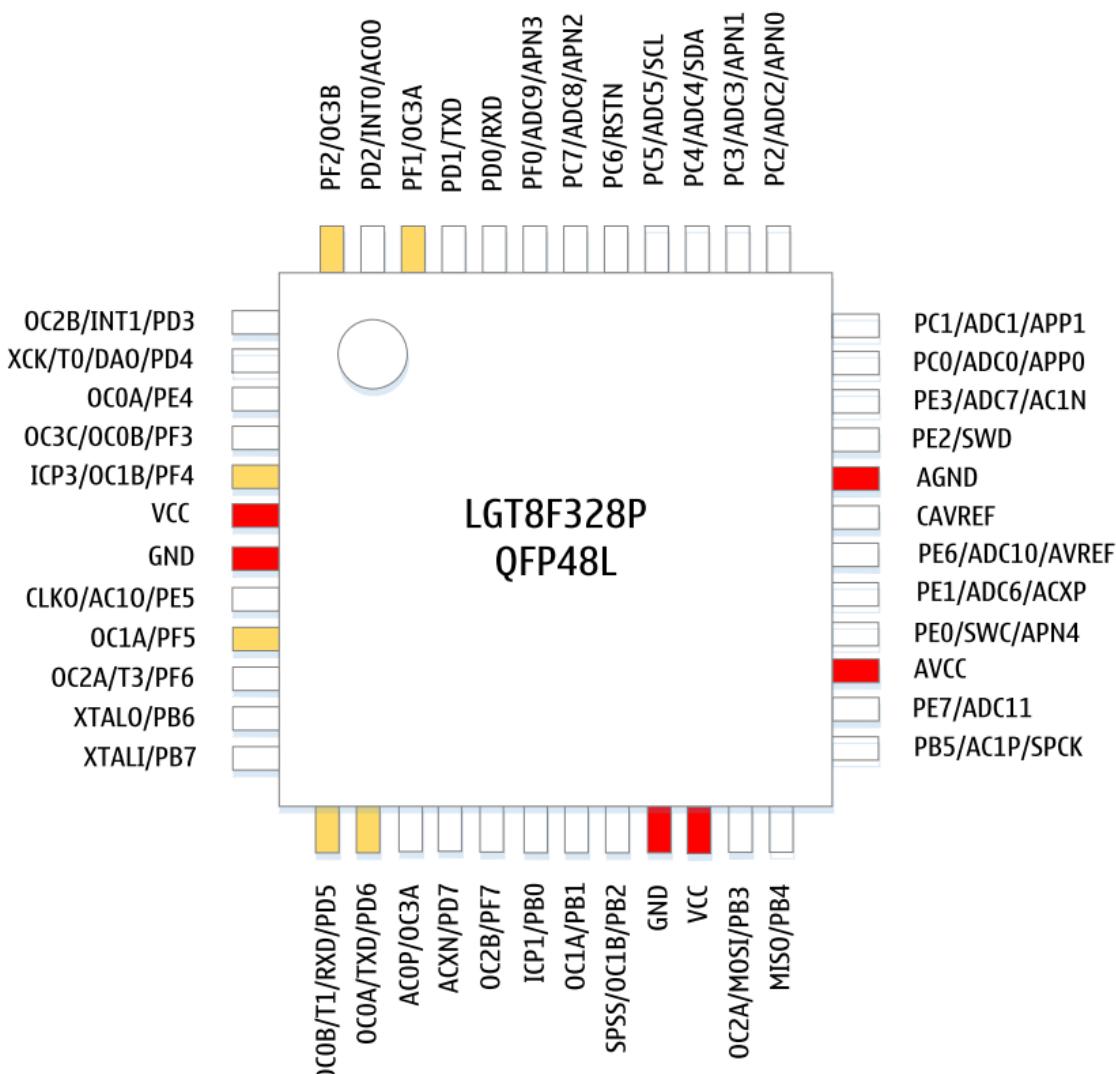


Рис.1.2. Корпус LGT8F328P - QFP48L

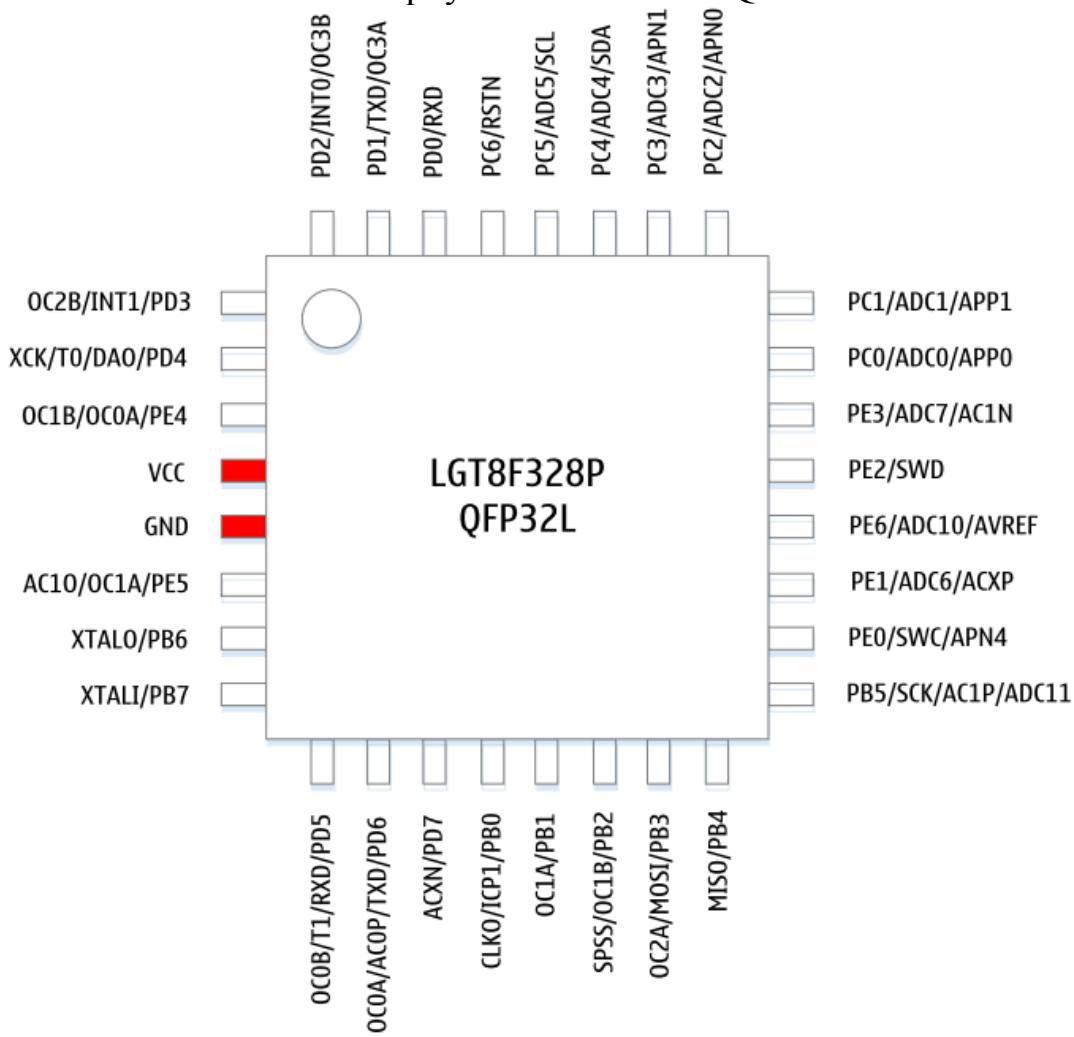


Рис.1.3. Корпус LGT8F328P - QFP32L

## 1.2. Описание выводов

В семействе корпусов LGT8FX8P корпус QFP48L приводит ко всем выводам. Другие корпуса, основанные на QFP48, будут содержать более одного внутреннего ввода-вывода, привязанного к одному Сгенерировано на выводе.

Особое внимание должно быть уделено при настройке ориентации контакта. В Табл.1.1 перечислены привязки различных контактов корпуса.

Таблица 1.1.

QFP48	QFP32	Описание функции
01	01	<b>PD3 / INT1 / OC2B *</b> PD3: Программируемый порт D3 INT1: Вход внешнего прерывания 1 OC2B: Таймер 2 совпадения выход B
02	02	<b>PD4 / DAO / T0 / XCK</b> PD4: программируемый порт D4 DAO: Внутренний выход ЦАП T0: Внешний таймер ввода Timer0 XCK: Синхронизирующие Такты USART
03	03	<b>PE4 / 0C0A *</b> PE4: Программируемый порт E4

		OC0A: Таймер 0 совпадения выход А
04	-	<b>PF3 / OC3C / OC0B *</b> PF3: Программируемый порт F3 OC3C: Таймер 3 совпадения выход C OC0B: Таймер 0 совпадения выход B
05	03	<b>PF4 / OC1B * / ICP3</b> PF4: Программируемый порт F4 OC1B: Таймер 1 совпадения выход B. ICP3: Вход для записи таймера 3
06	04	Vcc
07	05	GND
08	06	<b>PE5 / AC1O / CLKO *</b> PE5: Программируемый порт E5 C1O: Выход аналогового компаратора AC1 CLKO: Выходной сигнал системы
09		<b>PF5 / OC1A *</b> PF5: Программируемый порт F5 OC1A: Таймер 1 совпадения выход А
10	-	<b>PF6 / T3 / OC2A *</b> PF6: Программируемый порт F6 T3: Внешний входной сигнал таймера 3 OC2A: Таймер 2 совпадения выход А
11	07	<b>PB6 / XTAL0</b> PB6: Программируемый порт B6 XTAL0: Вывод резонатора IO
12	08	<b>PB7 / XTAL1</b> PB7: Программируемый порт B7 XTAL1: Вход резонатора IO
13	09	<b>PD5 / RXD * / T1 / OC0B</b> PD5: Программируемый порт D5 RXD: Прием данных USART (необязательно) T1: Внешний входной сигнал таймера 1 OC0B: Таймер 0 совпадения выход B
14	10	<b>PD6 / TXD * / OC0A</b> PD6: Программируемый порт D6 TXD: Передача данных USART (альтернативная) OC0A: Таймер 0 совпадения выход А
15		<b>AC0P / OC3A</b> AC0P: Аналоговый компаратор 0 Неинверсный вход OC3A: Таймер 3 совпадения выход А
16	11	<b>PD7 / ACXN</b> PD7: программируемый порт D7 ACXN: Аналоговый компаратор 0/1 Общий инверсный вход
17	-	<b>PF7 / OC2B</b> PF7: Программируемый порт F7 OC2B: таймер 2 совпадения выход B
18	12	<b>PB0 / ICP1</b> PB0: Программируемый порт B0 ICP1: Вход для записи таймера 1
19	13	<b>PB1 / OC1A</b> PB1: Программируемый порт B1 OC1A: Таймер 1 совпадения выход А
20	14	<b>PB2 / OC1B / SPSS</b>

		PB2: программируемый порт В2 OC1B: таймер 1 сравнивает выходные данные В. SPSS: выбор микросхемы режима ведомого SPI
21	-	GND
22	-	VCC
23	15	<b>PB3 / MOSI / OC2A</b> PB3: Программируемый порт В3 MOSI: Ведущий выход SPI / ведомый вход OC2A: Таймер 2 совпадения выход А
24	16	<b>PB4 / MISO</b> PB4: Программируемый порт В4 MISO: Ведущий вход SPI / ведомый
25	17	<b>PB5 / SPCK / AC1P</b> PB5: Программируемый порт В5 SPCK: Синхронизирующий сигнал SPI AC1P: Аналоговый компаратор 1 Неинверсный вход
26	-	<b>PE7 / ADC11</b> PE7: Программируемый порт Е7 ADC11: АЦП Аналоговый входной канал 11
27	-	AVCC: Питание внутренней аналоговой схемы
28	18	<b>PE0 / SWC / APN4</b> PE0: Программируемый порт Е0 SWC: Такты интерфейса отладки SWD APN4: Канал 4 обратного входа дифференциального усилителя
29	19	<b>PE1 / ADC6 / ACXP</b> PE1: Программируемый порт Е1 ADC6: Аналоговый входной канал АЦП 6 ACXP: Аналоговый компаратор 0/1 общий положительный вход
30	20	<b>PE6 / ADC10 / AVREF</b> PE6: Программируемый порт Е6 ADC10: АЦП Аналоговый входной канал 10 AVREF: Внешний опорное напряжение, вход АЦП
31	-	CVREF: АЦП выходного опорного напряжения Только для внешнего конденсатора фильтра 0.1uF
32	-	AGND: Внутренняя аналоговая схема заземления
33	21	<b>PE2 / SWD</b> PE2: Программируемый порт Е2 SWD: Линия передачи данных отладки SWD
34	22	<b>PE3 / ADC7 / AC1N</b> PE3: Программируемый порт Е3 ADC7: Аналоговый входной канал АЦП 7 AC1N: Инверсный вход аналогового компаратора
35	23	<b>PC0 / ADC0 / APP0</b> PC0: Программируемый порт С0 ADC0: Аналоговый входной канал АЦП 0 APP0: Входной канал прямого дифференциального усилителя 0
36	24	<b>PC1 / ADC1 / APP1</b> PC1: Программируемый порт С1 ADC1: Аналоговый входной канал АЦП 1 APP1: Канал прямого ввода дифференциального усилителя 1
37	25	<b>PC2 / ADC2 / APN0</b> PC2: Программируемый порт С2 ADC2: Аналоговый входной канал АЦП 2

		APN0: Канал инверсного входа дифференциального усилителя 0
38	26	<p style="text-align: center;"><b>PC3 / ADC3 / APN1</b></p> <p>PC3: Программируемый порт C3 ADC3: Аналоговый входной канал АЦП 3</p> <p>APN1: Канал инверсного входа дифференциального усилителя 1</p>
39	27	<p style="text-align: center;"><b>PC4 / ADC4 / SDA</b></p> <p>PC4: Программируемый порт C4 ADC4: АЦП Аналоговый входной канал 4 SDA: Линия передачи данных контроллера I2C</p>
40	28	<p style="text-align: center;"><b>PC5 / ADC5 / SCL</b></p> <p>PC5: Программируемый порт C5 ADC5: АЦП Аналоговый входной канал 5 SCL: Линия Тактов контроллера I2C</p>
41	29	<p style="text-align: center;"><b>PC6 / RESETN</b></p> <p>PC6: Программируемый порт C6 RESETN: Вход внешнего сброса</p>
42	-	<p style="text-align: center;"><b>PC7 / ADC8 / APN2</b></p> <p>PC7: Программируемый порт C7 ADC8: Канал аналогового ввода АЦП 8</p> <p>APN2: Канал инверсного входа дифференциального усилителя 2</p>
43	-	<p style="text-align: center;"><b>PF0 / ADC9 / APN3</b></p> <p>PF0: Программируемый порт F0 ADC9: Аналоговый входной канал АЦП 9</p> <p>APN3: Канал инверсного входа дифференциального усилителя 3</p>
44	30	<p style="text-align: center;"><b>PD0 / RXD</b></p> <p>PD0: Программируемый порт D0 RXD: Вход для приема данных USART</p>
45	31	<p style="text-align: center;"><b>PD1 / TXD</b></p> <p>PD1: Программируемый порт D1 TXD: Выход для приема данных USART</p>
46		<p style="text-align: center;"><b>PF1 / OC3A</b></p> <p>PF1: Программируемый порт F1 OC3A: Таймер 3 совпадения выход A</p>
47	32	<p style="text-align: center;"><b>PD2 / INT0 / AC0O</b></p> <p>PD2: Программируемый порт D2 INT0: Вход внешнего прерывания 0 AC0O: Выход аналогового сравнения 0</p>
48		<p style="text-align: center;"><b>PF2 / OC3B</b></p> <p>PF2: программируемый порт F2 OC3B: таймер 3 совпадения выход B</p>

## 2. Ядро LGT8XM

- Маломощное питание
- высокоэффективная архитектура RISC
- 16-разрядное расширение LD / ST (только для uDSU)
- 130 инструкций, из которых более 80% для одного цикла
- Поддержка встроенной отладки (OCD)

## 2.1. Обзор

В этом разделе описывается основная архитектура и функции LGT8XM (Рис.2.1.). Ядро является мозгом MCU и отвечает за то, чтобы программа была правильной.

По этому, ядро должно иметь возможность точно выполнять вычисления, управлять периферийными устройствами и обрабатывать различные прерывания.

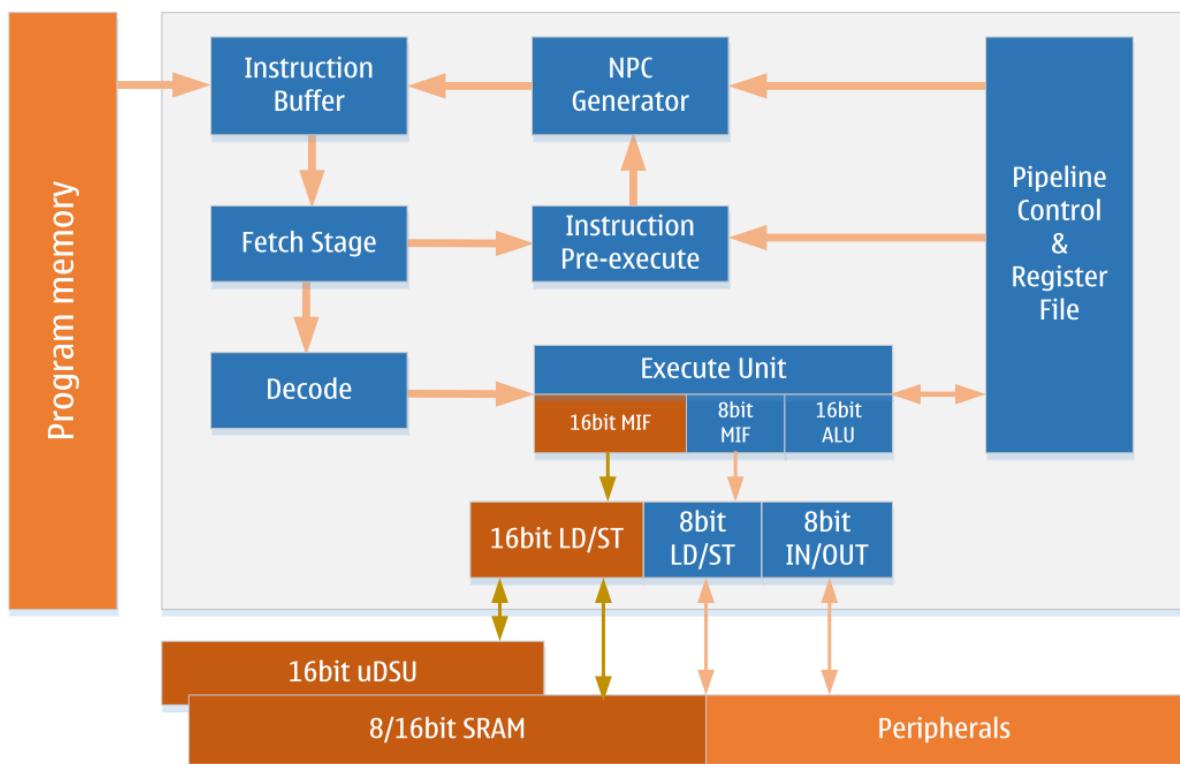


Рис.2.1. Структура ядра LGT8XM

Для большей эффективности и параллелизма ядро LGT8XM использует архитектуру Haval - отдельную шину данных и программ.

Инструкции выполняются с помощью оптимизированного двухступенчатого конвейера, который уменьшает количество недействительных инструкций в конвейере, уменьшает объем доступной памяти программ FLASH и, следовательно, снижает мощность, потребляемую ядром. В то же время основная инструкция LGT8XM извлекает в предварительном порядке для увеличения кеша (который может одновременно кэшировать две команды), с помощью модуля предварительного исполнения команды fetch.

Чтобы еще больше уменьшить частоту доступа к памяти программ FLASH, после обширного тестирования, LGT8XM может Уменьшите доступ к FLASH примерно на 50% по сравнению с другими ядрами той же архитектуры, что значительно снижает потребление энергии в системе.

Ядро LGT8XM имеет 32 файла регистра общего назначения для 8-битного высокоскоростного доступа, что позволяет выполнять однотактные арифметико-логические операции (ALU). В общем случае оба операнда операций ALU поступают из универсальных рабочих регистров, и результат операций ALU также записывается в файл регистра за один цикл.

32 из 32 через рабочий регистр для комбинации из двух, чтобы сформировать

три 16-разрядных регистра, могут использоваться для косвенного обращения к указателю адреса для доступа к внешнему пространству хранения и программному пространству FLASH.

LGT8XM поддерживает однотактные, 16-разрядные арифметические операции, что значительно повышает эффективность косвенной адресации. Три специальных 16-битных регистра в ядре LGT8XM называются регистрами X, Y, Z и будут подробно рассмотрены позже. ALU поддерживает арифметические и логические операции между регистрами, а также между константами и регистрами. В ALU также может быть выполнена одна операция с регистром. После завершения ALU-операции эффект результата операции в статусе ядра обновляется до регистра состояния (SREG).

Управление потоком программ посредством условного и безусловного перехода/вызова может быть адресовано всем программным областям. Большинство команд LGT8XM составляют 16 бит. Каждое адресное пространство программы соответствует 16-разрядной или 32-разрядной инструкции LGT8XM.

После того, как ядро ответит на вызов прерывания или подпрограммы, обратный адрес (ПК) сохраняется в стеке. Стек распределяется в SRAM общих данных системы, поэтому размер стека ограничивается только размером и использованием SRAM в системе.

Все приложения, которые поддерживают вызовы прерываний или подпрограмм, должны сначала инициализировать регистр указателя стека (SP), к которому можно получить доступ через пространство ввода-вывода.

Доступ к SRAM данных возможен через пять различных режимов адресации. Внутреннее пространство памяти LGT8XM линейно отображается в единое адресное пространство. Подробнее см. В главе хранения.

Ядро LGT8XM включает в себя гибкий контроллер прерываний, который может управляться с помощью бита разрешения глобального прерывания в регистре состояния. Все прерывания имеют отдельный вектор прерывания. Приоритет прерывания и адрес вектора прерывания имеют соответствующее соотношение, чем меньше адрес прерывания, тем выше приоритет прерывания.

Пространство ввода / вывода содержит 64 пространства регистров, которые могут быть непосредственно адресованы инструкциями IN / OUT. Эти регистры представляют собой реальный контроль над ядром и регистрами состояния, SPI и другими периферийными устройствами ввода-вывода. К этой части пространства можно получить доступ непосредственно инструкцией IN / OUT или по адресу (0x20 - 0x5F), который они сопоставляют с пространством памяти данных.

Кроме того, LGT8FX8P также включает расширенное пространство ввода / вывода, которое отображается в пространство памяти данных 0x60 - 0xFF, к которому можно получить доступ только с помощью инструкций ST / STS / STD и LD / LDS / LDD.

Для повышения вычислительной мощности ядра LGT8XM в популярную командную строку добавлено 16-разрядное расширение LD / ST. Это 16-разрядное расширение LD / ST работает с 16-разрядным цифровым арифметическим ускорителем (uDSU) для эффективных 16-разрядных операций с данными. В то же время ядро также увеличивает 16-разрядный доступ к оперативной памяти. Таким образом, 16-разрядное расширение LD / ST может передавать 16-разрядные данные между uDSU, RAM и рабочими регистрами.

Подробнее см. Раздел «Ускоритель цифровых вычислений».

## 2.2. Арифметический и логический блок (ALU)

LGT8XM содержит 16-разрядный арифметический и логический блок, который может быть выполнять в течение цикла из 16 арифметических операций. Эффективный ALU подключен к 32 рабочим станциям общего назначения. Арифметические и логические операции между двумя регистрами, или регистрами и константами могут быть выполнены за один цикл. Операции ALU делятся на три типа: арифметические, логические и битовые операции. В то же время часть ALU также содержит однотактный аппаратный множитель, который позволяет использовать два 8-битных регистра в течение одного цикла с прямой подписью или без знаковой арифметикой. Подробнее см. Раздел «Набор инструкций».

### 2.2.1 SREG - Регистр состояния

Регистр состояния в основном содержит информацию о результатах, сгенерированную при выполнении последней операции ALU. Эта информация используется для Контроль выполнения программы управления. Регистр состояния обновляется после завершения операции ALU, что устраняет необходимость использования отдельных Инструкции , что позволяет создание более компактного и эффективного кода.

Значение регистра состояния изменяется в ответ на прерывание и во время прерывания автоматически не сохраняется и восстанавливается, что требует программного обеспечения.

## *SREG - Регистр состояния системы*

## 2.2.2 Общий рабочий регистр

Общая архитектура рабочего регистра оптимизирована в соответствии с набором команд LGT8XM. Чтобы обеспечить эффективность и гибкость, необходимые для выполнения ядра, Внутренний рабочий регистр LGT8XM поддерживает несколько режимов доступа:

- 8-битное считывание в то же время 8-разрядной операции записи
- Два 8-битных одновременно считывают 8-разрядную операцию записи
- Два 8-битных одновременно считывают 16-разрядную операцию записи
- 16-разрядное считывание в то же время 16-разрядной операции записи

7	0	Addr.
R0		0x00
R1		0x01
R2		0x02
...		
R13		0x0D
R14		0x0E
R15		0x0F
R16		0x10
R17		0x11
...		
R26		0x1A X-регистр младший байт
R27		0x1B X-регистр старший байт
R28		0x1C Y-регистр младший байт
R29		0x1D Y-регистр старший байт
R30		0x1E Z-регистр младший байт
R31		0x1F Z-регистр старший байт

Рис.2.2. Общий рабочий регистры LGT8XM

Большинство инструкций имеют прямой доступ ко всем рабочим реестрам общего назначения, и большинство из них также являются инструкциями с одним циклом. Как показано на рисунке выше, каждый регистр соответствует адресу памяти данных, и эти рабочие регистры общего назначения отображаются в память данных. Как только они не существуют в SRAM, но это унифицированная организация хранения карт, чтобы получить к ним доступ с большой гибкостью.

Регистры X / Y / Z могут быть проиндексированы в любой регистр общего назначения.

## 2.2.3. Регистры X / Y / Z

Регистры R26 ... R31 могут быть объединены в комбинацию по два для формирования трех 16-разрядных регистров. Три 16-битных регистра в основном

используются как адресные указатели для косвенной адресации. Структура регистра X / Y / Z показана на Рис.2.3.

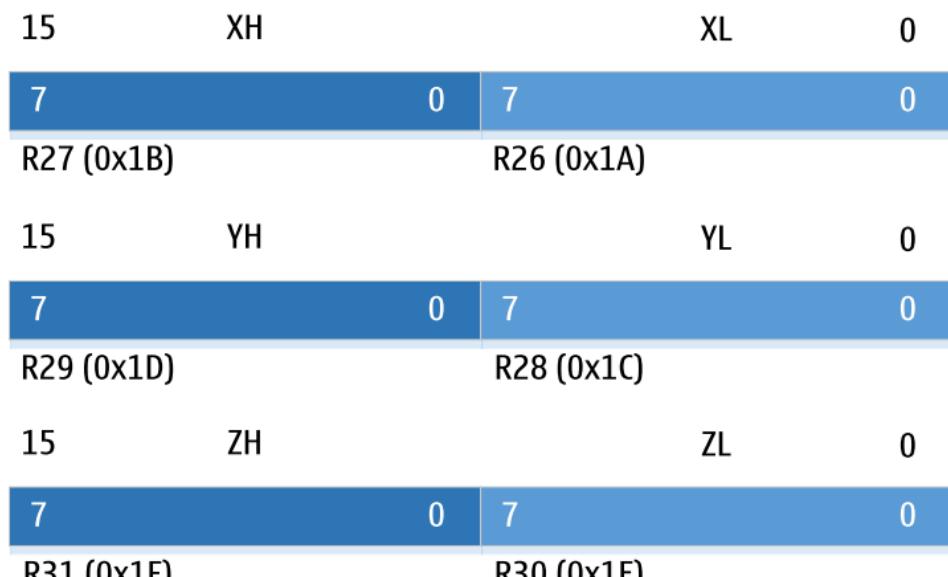


Рис.2.3. Регистры X, Y, Z.

В разных режимах адресации эти регистры используются как указатели с фиксированным смещением, автоинкрементным и автоматическим декрементом. Подробнее см. В разделе «Описание инструкций».

#### 2.2.4. Указатель стека

Стек используется для хранения временных данных, локальных переменных и обратных адресов для вызовов прерываний и подпрограмм.

**Важно отметить**, что стек не рассчитан на изменения от высокого адреса до низкого адреса. Регистр указателя стека (SP) всегда указывает на вершину стека.

Указатель стека указывает на физическое пространство, в котором находится SRAM данных, сохраняя пространство стека, необходимое для подпрограммы или вызова прерывания. Команда PUSH уменьшит указатель стека. Расположение стека в SRAM должно быть правильно установлено программным обеспечением до выполнения подпрограммы или прерывания. При нормальных обстоятельствах указатель инициализации указателя стека указывает на самый высокий адрес SRAM.

Указатель стека должен быть установлен на стартовый адрес SRAM. Адрес SRAM на карте памяти системных данных Обратитесь к разделу хранения системных данных.

#### *Инструкции по указанию указателя стека*

Указатель	Указатель стека	Описание
PUSH	Увеличение на 1	Данные, вводятся в стек
CALL ICALL RCALL	Увеличение на 2	Адрес прерывания или возврата для Вызова подпрограммы вводятся в стек
POP	Уменьшение на 1	Данные извлекаются из стека
RET RETI	Уменьшение на 2	Обратный адрес возврата вызова прерывания или подпрограммы извлекаются из стека

Указатель стека состоит из двух 8-разрядных регистров, выделенных в пространстве ввода-вывода. Фактическая длина указателя стека связана с реализацией системы. В некоторых реализациях чипов архитектуры LGT8XM пространство данных настолько мало, что только SPL может удовлетворить потребность в адресации, и в этом случае регистр SPH не появится.

#### *SPH / SPL - Определение регистра указателя стека*

Регистр указателей стека SPH / SPL	
SPH: 0x3E (0x5E)	По умолчанию: RAMEND
SPL: 0x3D (0x5D)	
SP	SP[15:0]
R/W	R/W
Определение бит	
[7:0] SPL	Указатель стека младшие 8 бит
[15:8] SPH	Указатель стека старшие 8 бит

#### 2.2.6. Сброс и обработка прерываний

Сброс и обработка прерываний LGT8XM поддерживает несколько источников прерываний. Эти прерывания и векторы возврата соответствуют отдельной записи вектора программы в программном пространстве.

В общем, все прерывания имеют отдельные управляющие элементы управления. Когда этот управляющий бит установлен, а бит включения глобального прерывания ядра включен, ядро может реагировать на это прерывание.

Самое низкое программное пространство зарезервировано по умолчанию для векторов сброса и вектора прерываний. Полный список прерываний, поддерживаемых LGT8FX8P, описан в разделе «Прерывание». Этот список также определяет приоритет различных прерываний. Чем ниже адрес вектора, тем выше приоритет соответствующего прерывания. СБРОС имеет наивысший приоритет, а затем INT0 - запрос внешнего прерывания 0.

Начальный адрес таблицы векторов прерываний, кроме вектора сброса, может быть переопределен до начала любого 256-байтового выравнивания и должен быть очищен с помощью управляющего регистра MCU (MCUCR) в битах IVSEL и реализации регистрового регистра векторной базы IVBASE. Когда ядро реагирует на прерывание, флаг включения глобального прерывания, I, автоматически очищается аппаратным обеспечением.

Пользователь может вложить прерывания, включив бит I. Любые последующие прерывания прерывают текущую процедуру обслуживания прерываний.

Бит I устанавливается автоматически после выполнения команды возврата прерывания (RETI), так что на последующее прерывание можно нормально ответить.

Существует основной тип прерывания. Первый тип запускается событием, которое устанавливает флаг прерывания после события прерывания. Для такого прерывания после того, как ядро отвечает на запрос прерывания, текущее значение ПК напрямую заменяется фактическим адресом вектора прерывания, выполняется соответствующая процедура обслуживания прерывания, и аппаратное обеспечение автоматически очищает бит флага прерывания.

Флаг прерывания также можно очистить, записав 1 в местоположение флага прерывания. Если бит прерывания очищается при прерывании, флаг прерывания по-прежнему будет установлен для записи события прерывания. Подождите, пока прерывание не будет включено, зарегистрированные события прерывания будут немедленно откликаться.

Аналогично, если бит разрешения глобального прерывания (SERG.I) очищается во время прерывания, соответствующий флаг прерывания также установлен для записи события прерывания. Эти биты разрешения глобального прерывания установлены. Будет выполнен в порядке приоритета.

Второй тип прерывания - это когда условие прерывания существует, прерывание отвечает. Это прерывание не требует флагов прерывания. Если условие прерывания исчезает до того, как прерывание включено, прерывание не будет отвечать. Когда ядро LGT8XM выходит из подпрограммы обслуживания прерываний, поток выполнения возвращается к основной процедуре. В основной программе после выполнения одной или нескольких инструкций в ответ на другой ожидающий запрос прерывания.

Следует отметить, что системный регистр состояния (SREG) автоматически не сохраняется, после входа в программу прерывания и не будет автоматически возобновляться после прерывания службы. Программное обеспечение должно обеспечить сохранение и восстановление SREG.

При использовании команды CLI для отключения прерываний - прерывания будут немедленно отключены. Все прерывания, которые происходят после команды CLI, не отвечают. Даже прерывания, которые происходят одновременно с выполнением команд CLI, не реагируют. В следующем примере показано, как использовать CLI, чтобы избежать прерывания записи EEPROM:

#### Примеры кода Ассемблера

IN R16, SREG	; хранить значение SREG
CLI	; отключить прерывания во время временной последовательности
SBI EECR, EEMPE	; начать запись EEPROM
SBI EECR, EEPE	
OUT SREG, R16	; восстановить значение SREG (включая бит I)

#### Примеры кода С

```
char cSREG;
cSREG = SREG;                                /* хранить значение SREG */
/* отключить прерывания во время временной последовательности */
_CLI();
EECR |= (1 << EEMPE);                      /* начать запись EEPROM */
EECR |= (1 << EEPE);
SREG = cSREG;                                  /* восстановить значение SREG (включая I-бит) */
```

Когда прерывание разрешено с использованием инструкции SEI, команда, следующая за инструкцией SEI, будет выполнена до того, как прерывание получит ответ, как в следующем примере кода:

#### Примеры кода Ассемблера

SEI	; установить глобальное прерывание
SLEEP	; войдите в режим сна, ожидая прерывания
; note:	войдет в сон перед любым ожидающим прерывания (ов)

### Примеры кода C

```
__enable_interrupt(); /* установить глобальное прерывание Включить */  
__sleep(); /* введите сон, ожидая прерывания */  
/* note: войдет в сон до любого ожидающего прерывания (ов) */
```

### 2.2.7. Время отклика прерывания

Ядро LGT8XM оптимизировано для ответа прерывания, поэтому любое прерывание должно обрабатываться в течение 4 тактовых циклов системы. После 4 циклов системных тактовых импульсов процедура обслуживания прерываний вводит цикл выполнения. В этих четырех тактах значение ПК перед прерыванием помещается в стек, а поток выполнения системы переходит к вектору прерывания, соответствующему подпрограмме обслуживания прерываний.

Если прерывание происходит во время выполнения команды с несколькими циклами, ядро гарантирует, что правильное выполнение текущей команды будет завершено.

Если прерывание происходит, когда система находится в режиме сна (SLEEP), ответ прерывания требует дополнительных 4 тактов. Этот увеличенный тактовый цикл используется для пробуждения цикла синхронизации операции из выбранного спящего режима. Подробное описание режима ожидания см. В соответствующем разделе управления питанием.

Для возврата из процедуры обслуживания прерываний требуется 2 такта. Во время этих двух тактовых циклов ПК восстанавливается из стека, указатель стека увеличивается, а бит управления глобальным прерыванием автоматически включается.

## 3. Блок Памяти

### 3.1. Обзор

В этом разделе в основном описываются различные внутренние ячейки памяти серии LGT8FX8P.

Архитектура LGT8XM поддерживает два основных типа внутренней памяти: хранение данных и хранение программ.

LGT8FX8P также содержит данные FLASH, внутренние, через внутренний контроллер могут быть реализованы возможности хранения данных интерфейса EEPROM.

Кроме того, система LGT8FX8P также содержит специальный блок памяти для хранения информации о конфигурации системы и глобального номера устройства чипа (GUID).

Микросхема LGT8FX8P содержит четыре различные модели LGT8F88P/168P/328P, четыре типа периферийных устройств и пакетов полностью совместимы, разница в программной памяти FLASH и внутренней SRAM данных, в Табл.3.1 ниже показано описание чипа серии LGT8FX8P.

Таблица 3.1. Разная конфигурация хранилища

Микросхема	FLASH	SRAM	E2PR0M	Вектор прерывания
------------	-------	------	--------	-------------------

LGT8F88P	8KB	1KB	2KB	1 слово инструкции
LGT8F168P	16KB	1KB	4KB	2 слово инструкции
LGT8F328P	32KB	2KB	Настраивается как 0K/1K/2K/4K/8K (совместно с FLASH)	2 слово инструкции

LGT8F328P не существует независимого внутреннего пространства FLASH для эмуляции интерфейса E2PROM, память, используемая для эмуляции E2PROM. Объем памяти и программный FLASH-обмен, пользователи могут выбирать в соответствии с потребностями приложения, с правильной конфигурацией.

Благодаря уникальной реализации эмулируемого интерфейса E2PROM системе требуется в два раза больше программ.

FLASH-эмуляция пространства E2PROM (Табл. 3.2.).

Место хранения, такое как LGT8F328P, если пользователь настроил 1 КБ пространства E2PROM, будет 2 Кбайт байт из зарезервированного программного пространства, оставляя 30 КБ пространства FLASH для хранения программ.

Таблица 3.2. Программая FLASH и конфигурации E2PROM LGT8F328P:

DEVICE	FLASH	E2PROM 1
LGT8F328P	32KB	0KB
	30KB	1KB
	28KB	2KB
	24KB	4KB
	16KB	8KB

### 3.2. Программируемый программный модуль памяти FLASH.

Микроконтроллеры серии LGT8FX8P включают в себя 8K / 16K / 32K байтов встроенных программируемых программных модулей памяти FLASH на кристалле. Программа FLASH может гарантировать как минимум 100 000 раз больше, чем цикл стирания.

LGT8FX8P интегрированный контроллер интерфейса FLASH для достижения внутрисистемного программирования (ISP) и функций самообновления программы.

Пожалуйста, обратитесь к описанию раздела контроллера интерфейса FLASH для получения подробной информации о реализации.

Программное пространство также можно получить непосредственно через инструкцию LPM (чтение), эта функция может быть получена с помощью постоянной таблицы поиска приложений. В то же время программное пространство FLASH также отображается на пространство хранения системных данных, пользователь также может использовать LD / LDD / LDS для достижения доступа к FLASH-пространству.

Программное пространство отображается в диапазон адресов, начиная с области хранения данных 0x4000, как показано на Рис.3.1.

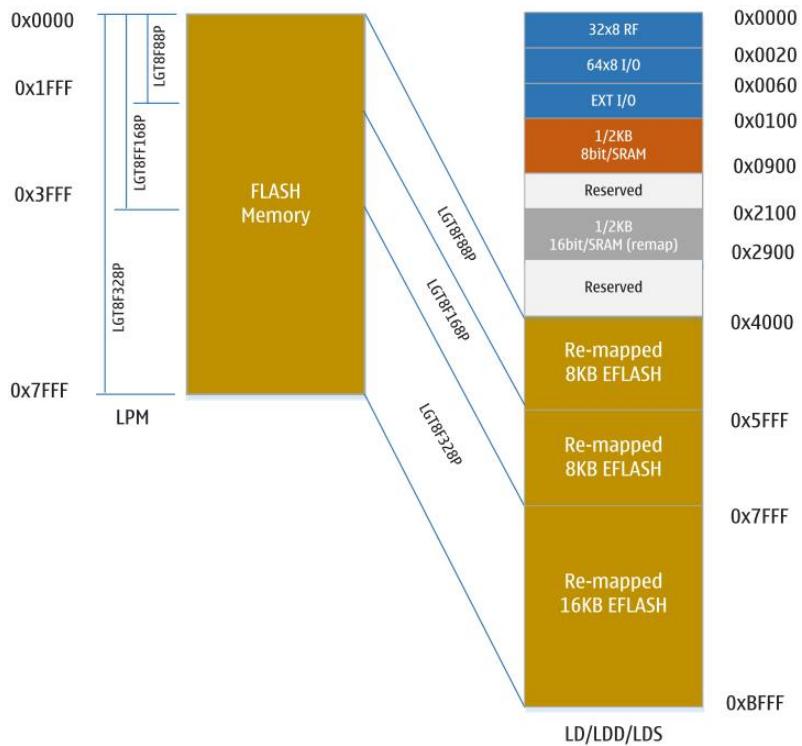


Рис.3.1. Программное пространство

### 3.3. Ячейка памяти данных SRAM

Семейство микроконтроллеров LGT8FX8P является относительно сложным микроконтроллером, который поддерживает множество различных типов периферийных устройств, а контроллеры для этих периферийных устройств выделяются в 64 регистрационных пространствах ввода/вывода. К нему можно получить доступ непосредственно инструкцией IN/OUT.

Другие регистры управления периферийными устройствами выделяются в области 0x60 ~ 0xFF, потому что эта часть пространства отображается на место хранения данных, только ST/STS/STD и LD/LDS/LDD и другие инструкции для посещения.

Системное пространство хранения данных LGT8FX8P начинается с адреса 0, отображает общий рабочий файл регистров, пространство ввода-вывода, расширенное пространство ввода-вывода и пространство SRAM для внутренних данных соответственно.

Начало 32-байтного адреса соответствует 32 общим рабочим зонам ядра LGT8XM.

Следующие 64 адреса - это стандартное пространство ввода-вывода, к которому можно напрямую обращаться по инструкциям IN/OUT.

Затем 160 адресов расширяют пространство ввода-вывода.

А затем до 2 Кбайт данных SRAM.

Часть пространства, начиная с 0x4000 и заканчивая 0xBFFF, отображает местоположение памяти программы FLASH.

Система 1К/2К байтов SRAM отображается в две области резерва. Это пространство, начиная с 0x0100 и заканчивая 0x0900, считывается и записывается ядром в 8-битных байтах.

От 0x2100 до 0x2900 область данных шириной 16 бит.

ОЗУ системы с адреса на 0x2100 в основном используется для работы с

модулем uDSU для достижения эффективного 16-разрядного хранения данных. Во время программирования обычный 8-разрядный адрес переменной адреса плюс смещение 0x2000, вы можете переключиться на 16-разрядный режим доступа.

Система поддерживает пять различных режимов адресации, которые охватывают все пространство данных: прямой доступ, косвенный доступ со смещением, косвенный доступ, косвенный доступ с декрементом перед доступом и косвенный доступ к адресам с инкрементом.

Рабочие регистры общего назначения R26-R31 Указатели адресов для косвенного доступа. Косвенный доступ может адресовать все пространство для хранения данных. Косвенный доступ со смещениями адресами может быть адресован 63 адресным пространствам вблизи адреса регистров Y/Z. Адресный регистр X/Y/Z автоматически уменьшается/увеличивается на аппаратное обеспечение до / после доступа, используя режим косвенного доступа регистра, который поддерживает адрес автоматического увеличения/уменьшения.

См. Описание набора инструкций.

16-разрядные регистры X / Y / Z и соответствующие режимы автоматической адресации (увеличение, уменьшение) также играют очень важную роль в 16-разрядном расширенном режиме. 16-разрядный расширенный режим может использовать режим увеличения / уменьшения LD / ST для автоматического увеличения, декремент-адрес с переменными. Этот режим очень эффективен при работе с массивом.

Конкретную реализацию см. В главе «Цифровой арифметический ускоритель (uDSU)».

### 3.4. Регистры ввода-вывода общего назначения

В пространстве ввода/вывода LGT8FX8P имеется три регистра ввода-вывода общего назначения, GPIO2/1/0, к которым можно получить доступ, используя инструкции IN/OUT для хранения данных, определяемых пользователем.

### 3.5. Периферийное пространство регистров

Подробное определение периферийного пространства ввода-вывода см. В главе «Обзор регистра» в листе данных LGT8FX8P. Таким образом, периферийные устройства LGT8FX8P назначаются пространству ввода-вывода.

Доступ ко всем адресам пространств ввода/вывода можно получить с помощью инструкций LD/LDS/LDD и ST/STS/STD. Доступ к данным осуществляется через 32 рабочих регистра общего назначения.

Регистры ввода-вывода от 0x00 до 0x1F могут быть доступны инструкциями бит-адреса SBI и CBI. В этих регистрах значение одного из битов может быть обнаружено с помощью инструкций SBIS и SBIC для управления исполнением программы. См. Описание набора инструкций.

При использовании инструкции IN/OUT для доступа к регистрам ввода-вывода необходимо адресовать адреса от 0x00 до 0x3F.

При использовании LD или ST-инструкций для доступа к пространству ввода-вывода он должен быть доступен (плюс смещение 0x20) через

сопоставленный адрес пространства ввода-вывода в едином пространстве карты памяти системных данных.

Другие периферийные регистры (0x60 ~ 0xFF), назначенные расширенному пространству ввода-вывода, могут быть доступны только с помощью инструкций ST/STS/STD и LD/LDS/LDD.

Чтобы быть совместимыми с будущими устройствами, зарезервированные биты должны записываться 0 при записи. Вы не можете выполнять запись на зарезервированном пространстве ввода-вывода. Некоторые регистры включают флаг состояния, который необходимо записать «1» для очистки.

Следует отметить, что инструкции CBI и SBI поддерживают только определенные биты, поэтому CBI/SBI может работать только с регистрами, содержащими эти флаги состояния. Кроме того, инструкция CBI/SBI может работать только в диапазоне регистров от 0x00 до 0x1F.

### 3.6. Контроллер FLASH (E2PCTL)

LGT8FX8P интегрирует гибкий и надежный контроллер чтения/записи EFLASH внутри, который может использовать существующее пространство хранения данных FLASH в системе для реализации пространства хранения для чтения и записи байтов и реализации приложений хранения, подобных E2PROM. Эмуляция интерфейса E2PROM с использованием алгоритма выравнивания стирания, вы можете увеличить использование цикла FLASH данных примерно 1 раз, чтобы обеспечить более 100 000 раз цикл стирания.

Контроллер E2PCTL также реализует операцию стирания в Интернете на программном пространстве FLASH и может автоматически обновлять прошивку через программное обеспечение. Через контроллер FLASH для доступа к программному пространству программы FLASH поддерживается только стирание страницы (1024 байта) и доступ к чтению и записи с 32-разрядной шириной.

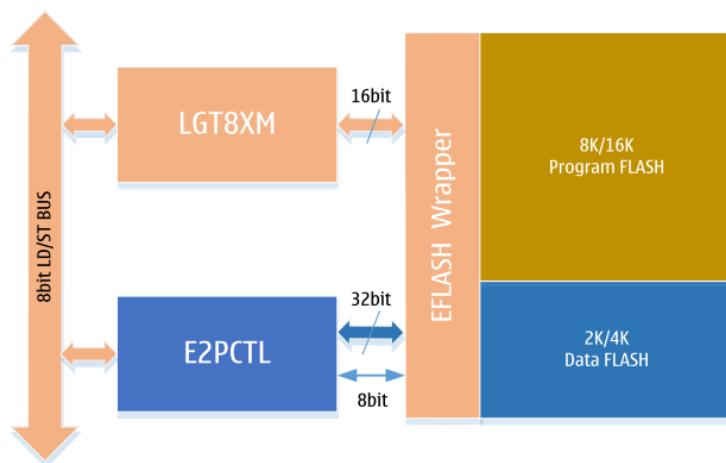


Рис.3.2. Структура контроллера LGT8F88D/168D E2PCTL

E2PCTL аналоговый E2PROM для доступа к пространству FLASH данных, вы можете поддерживать 8-битную 32-битную ширину чтения и записи. Посетите программу FLASH space, стирание страницы поддержки и чтение и запись 32-битных данных.

Поскольку минимальная внутренняя ячейка памяти FLASH для LGT8FX8P составляет 32 бита, рекомендуется использовать метод 32-разрядного доступа,

особенно для операций записи. 32-разрядный доступ к операциям чтения и записи не только с высокой эффективностью, но и для защиты срока службы ячейки памяти флэш-памяти.

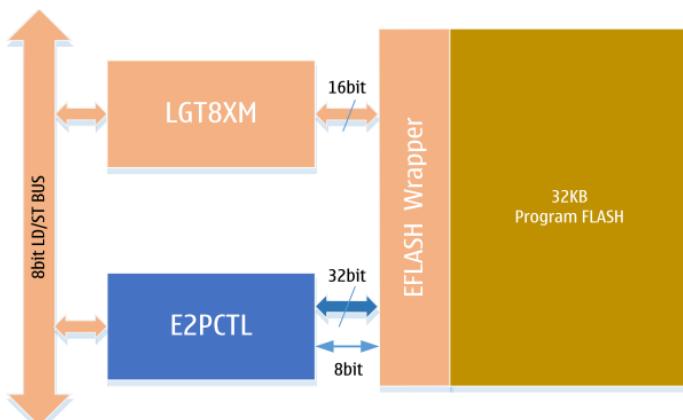


Рис.3.3. Структура контроллера LGT8F328P E2PCTL

LGT8F328P не содержит внутренних данных FLASH. Поэтому ядро LGT8XM и E2PCTL совместно используют внутренние 32 Кбайт памяти FLASH. Согласно потребностям пользователей, 32 Кбайт пространства FLASH можно разделить на пространство и пространство данных.

Конфигурируя контроллер E2PCTL, вы можете установить размер эмулируемого пространства E2PROM. E2PCTL использует режим обмена страницей для эмуляции логики E2PROM, алгоритма на странице (1 Кбайт) в качестве единицы. Поэтому, имитируя 1 Кбайт пространства E2PROM, вам необходимо занять 2 Кбайт пространства FLASH и т. д., Чтобы достичь 4 Кбайт E2PROM, необходимо взять 8 Кбайт пространства FLASH. Конкретная реализация, пожалуйста, обратитесь к алгоритму E2PCTL, чтобы получить описание.

### 3.6.1. Регистр данных E2PCTL

Контроллер E2PCTL имеет 4-байтовый буфер данных (E2PD0 ~ 3) внутри. Этот 4-байтовый буфер формирует 32-битный интерфейс данных, который, наконец, обращается к пространству FLASH.

Когда контроллер E2PCTL работает в режиме чтения и записи в байтах, EEDR действует как интерфейс для чтения и записи байтовых данных.

E2PCTL добавляет информацию адреса EEARL [1: 0] для загрузки данных в правильный буфер данных и в соответствии с текущим целевым адресом FLASH. Данные дополняют остальные три байта данных, в итоге объединяют полные 32-битные данные во FLASH.

Когда E2PCTL работает в 32-битном режиме чтения / записи, регистр EEDR все еще может использоваться как общий интерфейс данных для адресации и чтения внутреннего кэша данных через EEARL [1: 0] в качестве адреса для чтения и записи полных 32-разрядных данных. Кроме того, прямой доступ (E0 ~ 3) может выполняться непосредственно с использованием кэша данных, отображаемого в регистры в пространстве ввода-вывода.

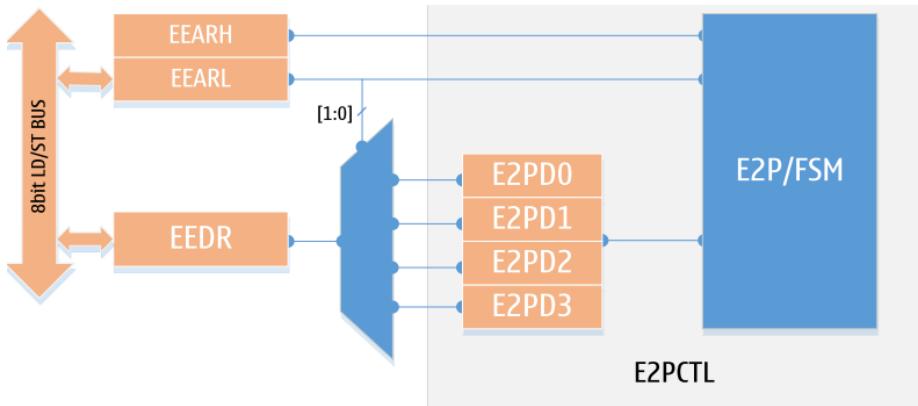


Рис.3.4. E2PCTL работают в диаграмме доступа к данным в режиме чтения и записи 8-битного байта.

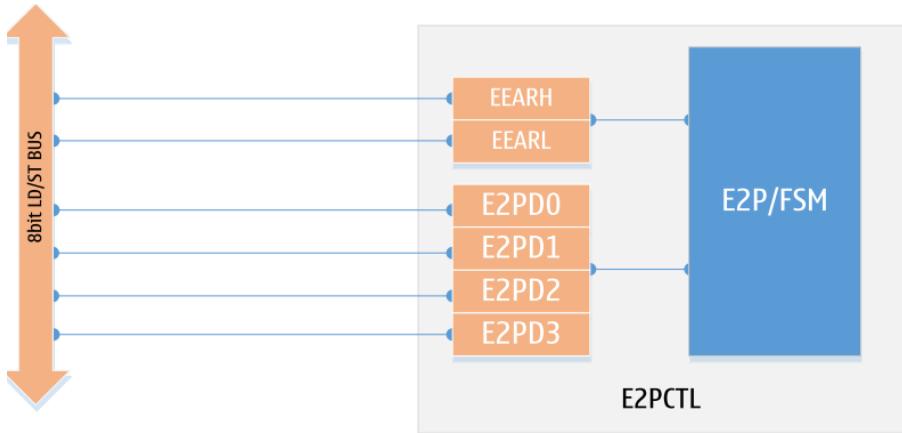


Рис.3.5. E2PCTL работает в 32-битной схеме доступа к данным для чтения и записи данных.

Режим байта используется для режима обратной записи / записи байта в обратном направлении для LGT8FX8D. Благодаря встроенной 32-битной ширине интерфейса FLASH LGT8FX8P использование 32-битного режима чтения и записи дает возможность чтения и записи, а срок службы стирания флэш-памяти принесет большие преимущества, рекомендуется использовать 32-битный режим чтения и записи.

### 3.6.2. E2PCTL аналоговый алгоритм интерфейса E2PROM.

Мы знаем, что FLASH-память должна быть удалена перед записью, а операция стирания основана на странице. Встроенная FLASH-память LGT8FX8P, размер страницы 1 Кбайт. Поэтому, чтобы обновить байт данных на странице, необходимо также сначала стереть всю страницу данных, а затем обновить данные целевого адреса и одновременно восстановить другие страницы данных, вся операция будет не только трудоемкой, но и принесенной. Потеря данных из-за случайных потерь мощности. E2PCTL внутренний алгоритм подкачки страниц для достижения аналогового E2PROM. Режим обмена свопами страниц, чтобы гарантировать, что при реализации операции стирания страницы не происходит из-за сбоя питания и других неожиданных причин потери исходных данных. В то же время алгоритм обмена использует поочередно два пространства для обмена, а также увеличивает срок службы имитируемого пространства E2PROM. С точки зрения эффективности контроллер E2PCTL реализует непрерывный режим

обновления данных, который уменьшает повторяющиеся циклы стирания, вызванные повторными обновлениями данных. С точки зрения реализации E2PCTL управляет каждой страницей отдельно и занимает последние 2 байта страницы как информацию о состоянии страницы. Таким образом, пользователи, использующие E2PROM более 1К пространства моделирования, вам нужно обратить внимание на адрес в специальном режиме на 1К. Поскольку последние 2 байта каждого 1 Кб зарезервированы для E2PCTL, пользователи не могут читать и записывать эти 2 байта пространства в обычном режиме.

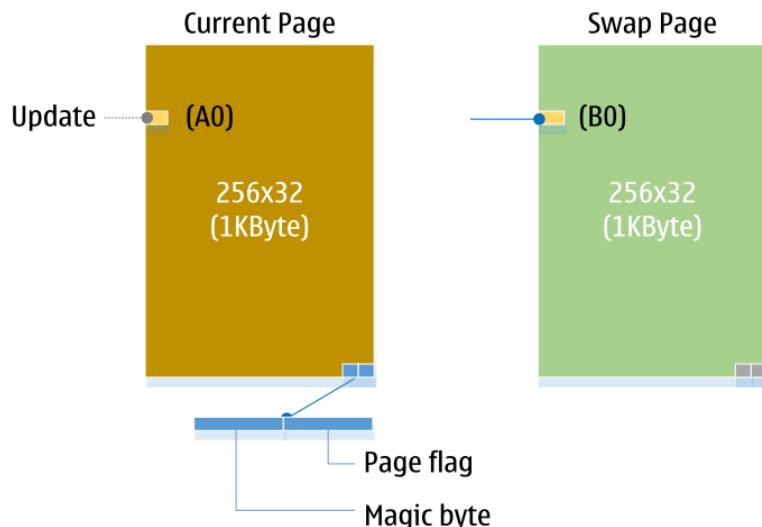


Рис.3.6 Схема E2PCTL на основе алгоритма переключения страниц.

Как показано на рисунке, E2PCTL внутренне использует две страницы для эмулирования одного пространства E2PROM размера страницы. Одна из этих двух страниц отмечена как текущая страница, другая - для обмена страницами. E2PCTL использует последние 2 байта страницы для хранения информации о странице. Когда нам нужно обновить страницу в байте, например, цифру A0 байт. Во-первых, мы не удалим текущую страницу, но сотрим страницу свопинга. Затем текущая страница делится на 3 части. Во-первых, это данные до A0. Давайте сделаем эту часть пространства в CP0, а затем данные после A0. Эта часть пространства CP1. E2PCTL скопирует данные, соответствующие CP0, соответствующему адресу страницы свопинга в соответствии с пользовательской конфигурацией, а затем напишет данные, которые будут обновлены, до адреса, соответствующего странице свопинга (B0), и, наконец, скопирует данные CP1 на страницу свопинга. После завершения вышеуказанной операции данные обмениваются, но статус страницы не обновляется. Поэтому, если до этого времени произошел сбой питания или другие аномалии, операция обновления не будет завершена, потому что предыдущие данные не будут повреждены и целостность данных будет гарантирована. Если все пойдет хорошо, E2PCTL напишет обновленный статус страницы на страницу информации предыдущей страницы свопинга в конце обмена данными CP1, чтобы заменить страницу лица. После этого страница обмена становится текущей страницей. E2PCTL, как показано ниже (1-> 2-> 3-> 4):

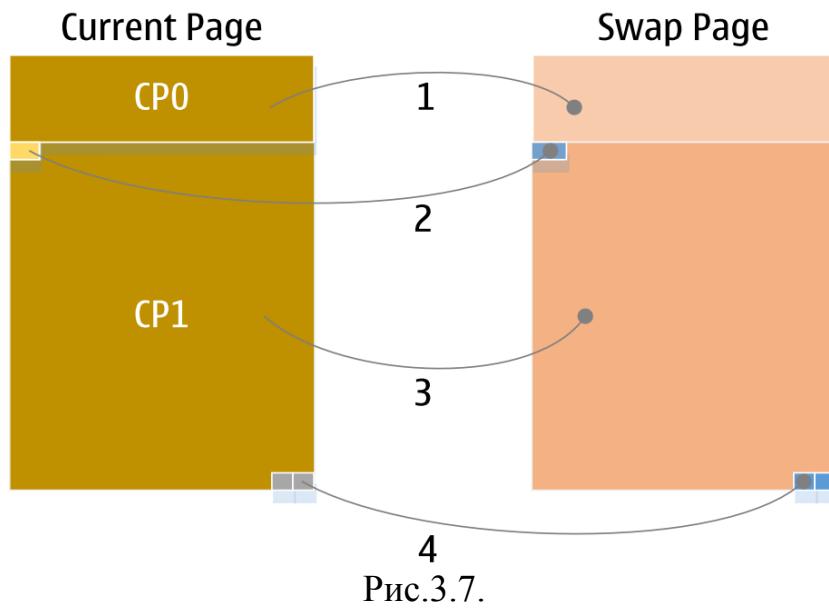


Рис.3.7.

Когда пространство для моделирования системы E2PROM больше, чем 1K, E2PCTL или страница на самый маленький блок для достижения алгоритма пространственного моделирования E2PROM.

Например, если пользователь настроил область 2K E2PROM, E2PCTL фактически занимает до 4 страниц (4K) пространства. Две из этих страниц сгруппированы вместе, чтобы реализовать пространство E2PROM, которое эмулирует размер страницы.

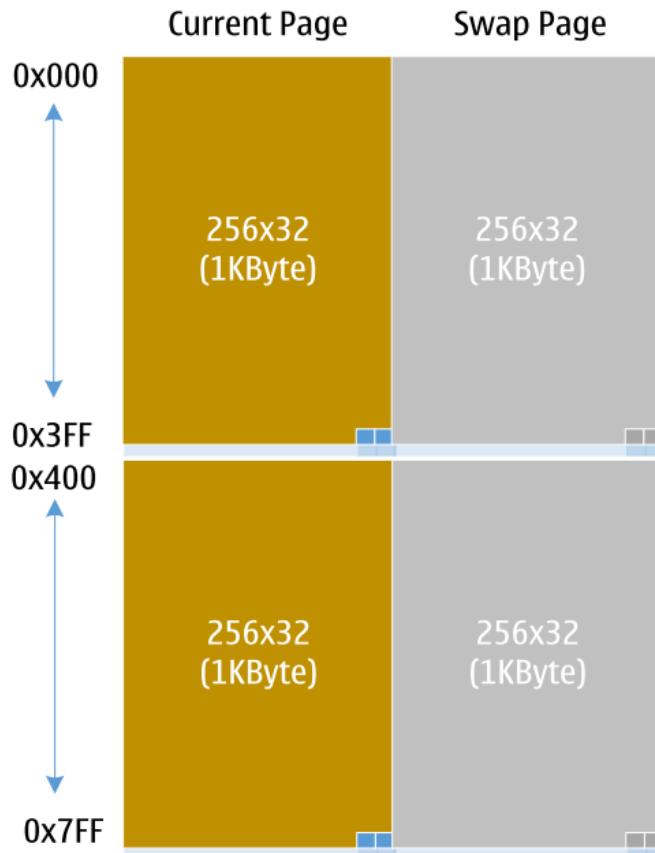


Рис.3.8.

**Обратите внимание**, что сконфигурированные пользователем 2K байты пространства E2PROM не смежны, потому что последние 2 байта каждой страницы

будут использоваться для сохранения информации о статусе страницы.

### 3.6.3. Режим непрерывного программирования E2PCTL

Поскольку обновление E2PCTL приводит к перестановке страниц, страница подкачки будет удалена во время обмена страницей. Сбой страницы не только отнимает много времени, но и увеличивает срок службы FLASH. Поэтому E2PCTL добавляет режим непрерывной записи.

В режиме непрерывной записи пользователь может постоянно обновлять область E2PROM. Операция обмена страницами выполняется только в конце последовательных адресов. Для непрерывного обновления всего блока данных непрерывный режим более эффективен.

Режим непрерывного программирования включает бит SWM в регистр ECCR E2PCTL. После включения непрерывного режима последующие операции записи будут записывать данные непосредственно на адрес, соответствующий странице подкачки. В режиме SWM операция записи не будет выполнять операцию копирования данных области CP0 / 1. Пока не будет записан последний байт, программное обеспечение отключает непрерывный режим с помощью SWM, а затем выполняет запись. Затем E2PCTL выполняет полную копию CP0 / 1 и обновляет информацию о статусе страницы.

### 3.6.4. E2PCTL чтение и запись программного пространства FLASH.

С помощью контроллера E2PCTL вы можете получить доступ к программному FLASH-пространству для чтения и записи. В отличие от эмулируемого E2PROM, доступ к программному пространству FLASH через E2PCTL требует программного обеспечения полностью. Шаги следующие:

1. Удалите целевую страницу, вам нужно сначала удалить целевую страницу перед обновлением данных, адрес страницы задается регистром EEAR. Для управления командой стирания страницы FLASH, пожалуйста, обратитесь к определению регистра EECR.

2. Чтобы записать пространство программы FLASH, 32 бита должны быть минимальным. Задайте данные через E2PD0 ~ 3.

3. Целевой адрес задается регистром EEAR, адрес EEAR [1: 0] будет проигнорирован, E2PCTL программа чтения и записи программы FLASH, онлайн-обновление программы (IAP) В полевых условиях используются данные приложения и приложения, которые должны предоставлять пользовательские обновления для продукта.

### 3.6.5. Рабочий процесс интерфейса E2PCTL

Интерфейс E2PCTL в основном осуществляется через четыре регистра, соответственно E2PCTL, регистр состояния EECR, ECCR, регистры данных EEDR (E2PD0 ~ E2PD3) и адресный регистр EEAR (EEARL / EEARH). Регистр ECCR используется для установки рабочего состояния E2PCTL. Большая часть состояния должна быть установлена до работы E2PCTL. Этот процесс обычно реализуется во

время инициализации системы.

Бит SWM в регистре ECCR используется для включения режима непрерывной записи. Этот бит управления необходимо установить во время операции непрерывной записи. Регистр EECR используется для управления типом операции выбора, используемой для выбора инструкций по эксплуатации, таких как команда чтения, удаления.

Регистр EEDR используется для 8-байтового режима интерфейса, E2PD0 ~ 3 используется для операций чтения и записи в 32-битном режиме, регистр EEAR используется для установки целевого адреса чтения и записи, также используется для установки адреса страницы стирания страницы.

Адрес страницы выравнивается в единицах страницы. Размер страницы составляет 1 Кбайт. Следует отметить, что адрес, указанный EEAR, является адресом байта.

### 3.6.6. Интерфейс E2PCTL для доступа к программному пространству FLASH

Через интерфейс E2PCTL программное пространство FLASH можно читать, записывать и стирать. Чтение и запись в FLASH-пространство поддерживает только 32-битную ширину доступа. Стереть операцию в блоках страниц, 1 Кбайт на страницу в размере (256x32). Прежде чем записывать программное пространство FLASH, сначала удалите страницу, на которой находится целевой адрес. E2PCTL записывать программное пространство FLASH не поддерживает непрерывный режим, пользователю необходимо выполнить последовательность записи.

Ниже приведен процесс стирания и записи программного пространства FLASH:

1. Программирование операции стирания страницы FLASH
  - Установите адрес EEAR [14: 0] в качестве адреса целевой страницы, который должен быть удален, размер страницы FLASH-программы составляет 1 Кбайт, поэтому EEAR [14: 10] будет использоваться как адрес страницы, а EEAR [9: 0] будет установлено на 0.
  - Установите EEPM [3: 0] = 1X01, где EEPM [2] можно установить на 0 или 1
  - Установить EEMPE = 1 и EEPE = 0
  - В Четыре цикла, установите EEPE = 1, запустите процесс FLASH стирания программы.
2. Программирование программирования FLASH.
  - Запишите E2PD0 ~ 3, подготовьте 32-битные данные программирования.
  - Установите EEAR в качестве целевого адреса, где адрес равен 4 байтам.
  - Установите EEPM [3: 0] = 1X10, где EEPM [2] можно установить как 0 или 1
  - Установить EEMPE = 1 и EEPE = 0
  - Установить EEPE = 1 в четыре цикла для запуска потока программирования FLASH

### 3.6.7. Доступ к пространству эмуляции E2PROM через интерфейс E2PCTL

Контроллер E2PCTL логически обращается к пространству FLASH данных через интерфейс эмуляции E2PROM. Аналоговый E2PROM поддерживает 8-битный, 16-разрядный и 32-разрядный доступ к данным для чтения и записи данных.

Режим 8-разрядного байта имеет лучшую совместимость с интерфейсом E2PROM.

32-битный режим способствует повышению эффективности хранения и времени FLASH, поэтому 32-битный режим чтения и записи является рекомендуемым режимом чтения и записи.

Аналоговый интерфейс E2PROM поддерживает непрерывный режим чтения и записи, при необходимости обновлять несколько последовательных приложений с данными адресов, очевидные преимущества, рекомендуется.

Для LGT8F88P / 168P данные FLASH в качестве отдельного пространства для хранения. Нет необходимости настраивать и разрешать пространство данных FLASH через регистр ECCR. LGT8F328P и без отдельных данных FLASH-пространство, FLASH и совместное использование данных FLASH-программа FLASH 32K байтов пространства.

Секция FLASH данных должна быть активирована регистром ECCR, а размер данных FLASH настраивается битами ECS [1: 0] в регистре ECCR. После того, как конфигурация вступает в силу, другой метод использования такой же, как LGT8F88P / 168P. FLASH в реализации интерфейса E2PROM, внутренняя была достигнута при необходимости для автоматического удаления данных FLASH-логика, команда EPROM erase необязательна, эта команда используется только тогда, когда пользователю необходимо выполнить удаление.

Регистр EECR контролирует последовательность стирания / записи флэш-памяти, включая программу FLASH и E2PROM. Специфический тип операции должен быть задан EEPME и EEPML [3: 0] регистра EECR.

Операция чтения E2PROM относительно проста. После установки целевого адреса и режима напишите бит EERE, чтобы прочитать 32-битные данные, соответствующие целевому адресу, в контроллер FLASH. Пользователь может прочитать интересующий байты через регистр EEDR. Контроллер FLASH не выполняет операцию чтения FLASH-памяти программы, пользователь может легко использовать LPM или программу FLASH в едином адресном пространстве адресов данных, используя инструкцию LD / LDD / LDS для чтения.

#### 1. 8-битное программирование режима E2PROM

- Установите целевой адрес в регистр EEARH / L
- Установите новые данные в регистр EEDR
- Установите EEPML [3: 1] = 000, EEPML [0] можно установить на 0 или 1
- Установить EEMPE = 1 и EEPE = 0
- Установите EEPE = 1 в четыре цикла. После завершения настройки контроллер FLASH запустит операцию программирования, и CPU останется в текущем адресе инструкции во время программирования и не будет продолжаться до завершения операции. Во время программирования, если вам нужно стереть данные FLASH, контроллер FLASH автоматически начнет процесс стирания.

## 2. 32-битный режим, программирование E2PROM

- Подготовьте 32-битные данные по E2PD0 ~ 3
- Установите целевой адрес в регистр EEARTH / L. Обратите внимание, что это адрес, выровненный по байтам, FLASH-контроллер с EEAR [15: 2] в качестве адреса для доступа к FLASH.
- Установить EEPM [3: 1] = 010, EEPM [0] можно установить на 0 или 1
- Установить EEMPE = 1 и EEEPE = 0
- Установить EEEPE = 1 в четыре цикла

## 3. 8-битный режим, читать E2PROM

- Установить адрес назначения в регистр EEARTH / L
- Установить EEPM [3: 1] = 000
- Установить EERE = 1 для запуска операции чтения E2PROM
- Подождать 2 цикла (выполнить две операции NOP)
- Данные, соответствующие целевому адресу, обновлены до EEDR  
Регистрация

## 4. 32-битный режим, чтение E2PROM

- Установите EEARTH / L в качестве целевого адреса с 4-байтным адресом
- Установите EEPM [3: 1] = 010, чтобы включить режим 32-битного интерфейса
- Установите EERE = 1, чтобы включить операцию чтения E2PROM
- Подождите 2 тактовых цикла системы (выполните две инструкции NOP)

E2PCTL доступ к аналоговому пространству E2PROM, поддержка режима непрерывного программирования, режим непрерывного доступа для необходимости обновления приложения блока данных очень эффективен, но также помогает улучшить срок службы FLASH.

Режим непрерывного программирования поддерживает только 32-битные операции программирования данных. Режим непрерывного доступа активируется бит SWM в регистре ECCR. После включения SWM следующая операция для записи эмулируемого пространства E2PROM через E2PCTL находится в режиме непрерывного программирования.

В режиме непрерывного программирования контроллер E2PCTL автоматически обрабатывает канал формы на основе данных на целевом адресе. Однако при изменении страницы в режиме непрерывного программирования контроллер не будет автоматически обмениваться данными в области CP0 / 1 или обновлять информацию о странице во время непрерывного программирования.

Перед непрерывным программированием до последней операции E2PCTL автоматически скопирует данные в области CP0 / 1 на страницу подкачки после завершения программирования, отключив бит SWM, чтобы отключить непрерывный режим программирования, а затем запустить последнюю операцию программирования в режиме без SWM. И обновите страницу обмена информацией, сделав ее текущей эффективной страницей, таким образом, завершив все непрерывное программирование.

## 5. Процедура непрерывного программирования:

1. Настраивает размер данных FLASH с помощью ECCR и включает бит SWM.
2. Запрограммируйте область E2PROM с использованием 32-битного режима.
3. Если нет, вернитесь к шагу 2, чтобы продолжить программирование следующих данных
4. Если достигнуто последнее программирование, сначала отключите режим непрерывного программирования с помощью SWM, а затем используйте последовательность операций с шага 2 для завершения последнего программирования

### 3.6.8. E2PCTL Эффективное управление данными FLASH

В дополнение к режиму непрерывного программирования контроллер E2PCTL также может самостоятельно управлять копированием обмена данными об обмене страницей через бит CP0/1 в регистре ECCR.

CP0/1 регистра ECCR используется для управления обменом данными для области CP0 / 1 на текущей странице во время обмена страницей, соответственно. Очистка бит CP0 / 1 не будет обменивать данные соответствующей области на текущей странице во время обмена файлами.

В этом разделе представлен эффективный метод управления, который будет использовать эту функцию. В процессе обновления данных FLASH наиболее длительная операция происходит при обмене процессом стирания страницы. Таким образом, мы можем рассмотреть подход к управлению данными, который минимизирует количество стираний страниц, что повышает эффективность программирования и уменьшает потерю жизни.

Здесь мы предоставляем справочный алгоритм для приложений управления данными на основе блоков данных:

1. Предположим, что пользовательские данные представляют собой только полный блок данных, размер блока данных является целым числом, кратным 4 байтам;
2. Каждое обновление данных будет обновлено Из блока данных
3. Данные блока данных в дополнение к хранению пользовательских данных, но также необходимо сохранить информацию управления блоком.

В вышеуказанных трех условиях мы можем в полной мере использовать режим непрерывного программирования E2PCTL и автоматический механизм обмена страницами для достижения эффективного метода управления данными FLASH.

Поскольку каждая обновленная информация представляет собой блок того же размера, и каждая структура данных содержит информацию о адресе, указывающую на следующую часть данных, мы можем запрограммировать FLASH в адресном порядке каждый раз, когда данные обновляются без необходимости делать CP0 / 1 Копия данных. В то же время, каждый раз, когда вы обновляете данные до удаленной области, это не будет стирать страницы. Когда записывается последний фрагмент данных, следующий фрагмент области данных, на который указывает его информация о структуре, возвращается к начальному адресу страницы. После этого снова будет выполняться операция записи данных. E2PCTL

запустит процесс стирания страницы и обновит текущую активную страницу.

### 3.6.9. Меры по защите от FLASH

Если напряжение V<sub>cc</sub> низкое, операция стирания FLASH может быть вызвана тем, что напряжение слишком низкое и возникает ошибка. FLASH / данные при ошибке с низким уровнем стирания напряжения могут быть по двум причинам.

Прежде всего, нормальная операция стирания FLASH требует минимального рабочего напряжения, ниже этого напряжения, операция завершится неудачно и приведет к ошибке данных.

Вторая причина заключается в том, что для ядра, работающего на определенной частоте, также требуется минимальное напряжение. Когда оно ниже этого напряжения, это приведет к ошибке выполнения команды, которая приведет к ошибке FLASH.

Аналогичную проблему можно избежать следующими простыми способами: когда напряжение питания низкое, пусть система войдет в состояние сброса. Это может быть достигнуто путем настройки внутренней схемы обнаружения низкого напряжения (VDT). Если VDT обнаруживает, что текущее рабочее напряжение ниже установленного порога, VDT выдаст сигнал сброса. Если порог VDT не может удовлетворить потребности приложения, вы можете рассмотреть возможность добавления схемы сброса снаружи.

## 3.7. Регистры Описание

### EEARH/EEARL - Адресный регистр FLASH

EEARH/EEARL		
EEARH: 0x22 (0x42)	По умолчанию: 0x0000	
EEARL: 0x21 (0x41)		
Биты	EEAR[15:0]	
R/W	R/W	
Определение бит		
[7:0]	EEARL	Адрес доступа EFLASH / E2PROM младше 8 бит.
[14:8]	EEARH	Адрес доступа EFLASH / E2PROM старшие семь
[15]	-	Зарезервировано

При использовании контроллера E2PCTL для доступа к области FLASH-программы, EEAR [14: 2] используется для доступа ко всему программному пространству, выровненному в 4 байта.

EEAR [1: 0] Используется только при обращении к регистру данных EEDR.

Пожалуйста, обратитесь к следующему описанию регистра данных EEDR. Контроллер E2PCTL поддерживает режим 8/16/32 бит. EEAR адресуется при выравнивании байтов независимо от режима.

### EEDR/E2PD0 - Регистр данных FLASH

EEDR / E2PD0 - регистр данных FLASH / E2PROM 0	
EEDR/E2PD0: 0x20 (0x40)	По умолчанию: 0x00
Биты	EEDR[7:0]
R/W	R/W

Определение бит		
[7:0]	EEDR/E2PD0	Регистр данных E2PCTL 16/32-битный режим, используемый для доступа к младшему байту

### *E2PD1 - Регистр данных FLASH*

E2PD1 - регистр данных E2PCTL 1		
E2PD1: 0x5A		По умолчанию: 0x00
<b>Биты</b>		E2PD1[7:0]
<b>R/W</b>		R/W
Определение бит		
[7:0]	E2PD1	Младшие 8 бит используются для хранения верхних 16 бит данных в 32-битном режиме

### *E2PD2 - Регистр данных FLASH*

E2PD2 - регистр данных FLASH		
E2PD2: 0x57		По умолчанию: 0x00
<b>Биты</b>		E2PD2[7:0]
<b>R/W</b>		R/W
Определение бит		
[7:0]	E2PD2	Младшие 8 бит используются для хранения верхних 16 бит данных в 32-битном режиме

### *E2PD3 - Регистр данных FLASH*

E2PD3 - регистр данных FLASH		
E2PD3: 0x5c		По умолчанию: 0x00
<b>Биты</b>		E2PD3[7:0]
<b>R/W</b>		R/W
Определение бит		
[7:0]	E2PD3	32-битный режим используется для хранения верхних 8 бит верхних 16 бит данных

### *ECCR - Регистр управления режимом FLASH*

Регистр конфигурации ECCR - FLASH / E2PROM									
ECCR: 0x36 (0x56)			0x0c						
Биты	WEN	EEN	ERN	SWM	CP1	CP0	ECS1	ECS0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Определение бит									
[7]	WEN								Управление разрешением записи ECCR Перед изменением ECCR вы должны записать WEN 1, а затем 6 системных циклов содержимое регистра ECCR будет обновляться.
[6]	EEN								E2PROM действителен только для LGT8F328P 1: Включить эмуляцию E2PROM, сохранит некоторое пространство от 32K FLASH 0: отключить эмуляцию E2PROM, 32K FLASH - все, что используется для программного пространства
[5]	ERN								Запись 1 сбрасывает контроллер E2PCTL

[4]	SWM	Режим непрерывной записи, подходящий для эмуляции работы контроллера E2PROM
[3]	CP1	Управление обменом параметрами страницы CP1
[2]	CP0	Управление обменом параметрами страницы CP0
[1:0]	ECS[1:0]	Конфигурация пространства E2PROM 00: 1 КБ E2PROM, программа 30 КБ FLASH 01: 2 КБ E2PROM, программа 28 КБ FLASH 10: 4 КБ E2PROM, программа 24 КБ FLASH 11: 8 КБ E2PROM, программа 16 КБ FLASH

### EECR - Регистр контроля доступа FLASH

Регистр управления EECR - FLASH / E2PROM																																																																																									
EECR: 0x1F (0x3F)					По умолчанию: 0x00																																																																																				
bits		EEPMS	EEPMS	EEPMS	EEPMS	EERIE	EEMPE	EEPE	EERE																																																																																
R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W																																																																																
Начальное значение		0	0	0	0	0	0	0	0																																																																																
<b>Определение бит</b>																																																																																									
Бит управления режимом доступа EFLASH / EPROM																																																																																									
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <th>[3]</th><th>[2]</th><th>[1]</th><th>[0]</th><th colspan="6">Описание режима</th></tr> <tr> <td>0</td><td>0</td><td>0</td><td>x</td><td colspan="6">8-разрядный режим чтения/записи E2PROM (по умолчанию)</td></tr> <tr> <td>0</td><td>0</td><td>1</td><td>x</td><td colspan="6">16-разрядный режим чтения / записи E2PROM</td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>x</td><td colspan="6">32-разрядный режим чтения / записи E2PROM</td></tr> <tr> <td>1</td><td>x</td><td>0</td><td>0</td><td colspan="6">Стирание E2PROM (дополнительная операция)</td></tr> <tr> <td>1</td><td>x</td><td>0</td><td>1</td><td colspan="6">Стирание программы Flash (стирание страницы)</td></tr> <tr> <td>1</td><td>x</td><td>1</td><td>0</td><td colspan="6">Программирование FLASH-программ</td></tr> <tr> <td>1</td><td>x</td><td>1</td><td>1</td><td colspan="6">Сброс контроллера FLASH / E2PROM</td></tr> </table>										[3]	[2]	[1]	[0]	Описание режима						0	0	0	x	8-разрядный режим чтения/записи E2PROM (по умолчанию)						0	0	1	x	16-разрядный режим чтения / записи E2PROM						0	1	0	x	32-разрядный режим чтения / записи E2PROM						1	x	0	0	Стирание E2PROM (дополнительная операция)						1	x	0	1	Стирание программы Flash (стирание страницы)						1	x	1	0	Программирование FLASH-программ						1	x	1	1	Сброс контроллера FLASH / E2PROM					
[3]	[2]	[1]	[0]	Описание режима																																																																																					
0	0	0	x	8-разрядный режим чтения/записи E2PROM (по умолчанию)																																																																																					
0	0	1	x	16-разрядный режим чтения / записи E2PROM																																																																																					
0	1	0	x	32-разрядный режим чтения / записи E2PROM																																																																																					
1	x	0	0	Стирание E2PROM (дополнительная операция)																																																																																					
1	x	0	1	Стирание программы Flash (стирание страницы)																																																																																					
1	x	1	0	Программирование FLASH-программ																																																																																					
1	x	1	1	Сброс контроллера FLASH / E2PROM																																																																																					
[7:4] EEPM[3:0]																																																																																									
[3] EERIE																																																																																									
Управление включением прерывания FLASH / E2PROM. Запись 1 - включить, запись 0 - запрещено. Прерывание готовности E2PROM действует, когда EEPE автоматически очищается аппаратным обеспечением. Во время операции EPROM это прерывание не произойдет																																																																																									
[2] EEMPE																																																																																									
FLASH / E2PROM управляющий бит разрешения EEMPE используется для управления действительностью EEPE. Когда EEMPE установлен в 1 и EEPE устанавливается одновременно на 0, установка EEPE в 1 в следующие четыре цикла запустит операцию программирования. В противном случае операция программирования недействительна. После четырех циклов EEMPE автоматически очищается																																																																																									
[1] EEPE																																																																																									
FLASH / E2PROM бит разрешения программирования																																																																																									
[0] EERE																																																																																									
Бит разрешения чтения E2PROM, данные будут действительны после двух системных циклов.																																																																																									

### GPIOR2 - Регистр ввода-вывода общего назначения

GPIOR2 - регистр ввода-вывода общего назначения 2		
GPIOR2: 0x2b (0x4b)		По умолчанию: 0x00
Биты		GPIOR2[7:0]
R/W		R/W
0x00		
<b>Определение бит</b>		
[7:0]		Регистр ввода-вывода общего назначения 2 для хранения пользовательских данных

## *GPIOR1 - Регистр ввода-вывода общего назначения*

GPIOR1 - регистр ввода-вывода общего назначения 1	
GPIOR1: 0x2a (0x4a)	По умолчанию: 0x00
Биты	GPIOR1[7:0]
R/W	R/W
	0x00
Определение бит	
[7:0]	GPIOR1 Регистр ввода-вывода общего назначения 1 для хранения пользовательских данных

## *GPIOR0 - Регистр ввода-вывода общего назначения*

GPIOR0 - регистр ввода-вывода общего назначения 0	
GPIOR0: 0x1e (0x3e)	По умолчанию: 0x00
Биты	GPIOR0[7:0]
R/W	R/W
	0x00
Определение бит	
[7:0]	GPIOR0 Регистр ввода-вывода общего назначения 0 для хранения пользовательских данных

## **4. Ускоритель вычислений (uDSC)**

- 16-разрядный режим памяти (LD / ST)
- 32-разрядный аккумулятор (DX)
- Однотактный 16-разрядный множитель (MUL)
- 32-разрядный арифметический и логический блок (ALU)
- 16-разрядная насыщенность (SD)
- 8 циклов 32/16 делитель
- Операция многократного добавления / вычитания с одним циклом (MAC / MSC)

### **4.1. Обзор**

Цифровой вычислительный ускоритель (uDSC), модуль арифметической обработки ядра LGT8XM, работает с режимом LDT ядра LGT8XM, Достигните 16-битного блока обработки цифрового сигнала. Большая часть класса управления соответствует цифровой обработке сигналов.

Функция uDSC Внутренняя и функциональная:

1. 16-разрядный регистр operandов DX / DY
2. 32-разрядный накопительный регистр DA
3. Однотактный 17-битный множитель (16-разрядный с умножением без знака)
4. 32-разрядный ALU (16/32 бит могут быть выполнены сложениями, вычитаниями и сдвигами)
5. 16-разрядная операция накопления (используется для хранения результатов в пространстве оперативной памяти)
6. Делитель 32/16, 8 циклов для завершения операции

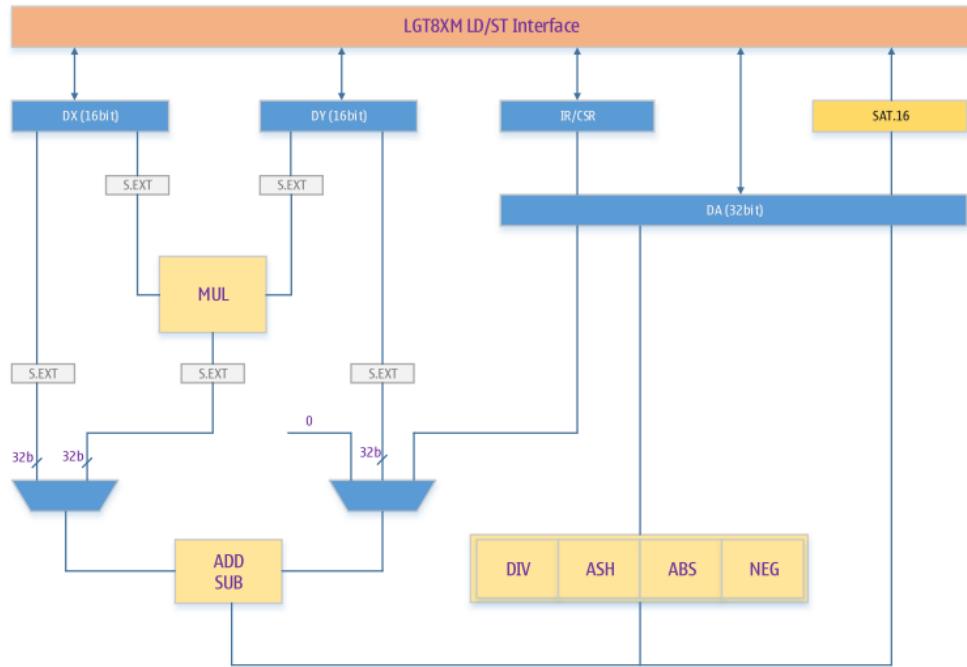


Рис.4.1 Структурная схема uDSC

## 4.2. 16-разрядные режимы работы LD / ST

Для повышения эффективности обработки больших данных uDSC в основном ядре LGT8XM используется выделенный 16-разрядный канал памяти LD / ST, который может быть эффективно использован между файлами регистра uDSC и SRAM и общего назначения с использованием команд LDD / STD 16-разрядный обмен данными.

Чтобы не повредить обычный набор команд LD / ST, ядро LGT8XM переназначает пространство SRAM в 0x2100 ~ 0x28FF. При использовании команды LD / ST для доступа к SRAM от 0x2100 до 0x28FF, ядро автоматически включает 16-разрядную функцию LD / ST, чтобы включить прямой доступ между SRAM и uDSC. На Рис. показано распределение адресов в пространстве ядра LGT8XM

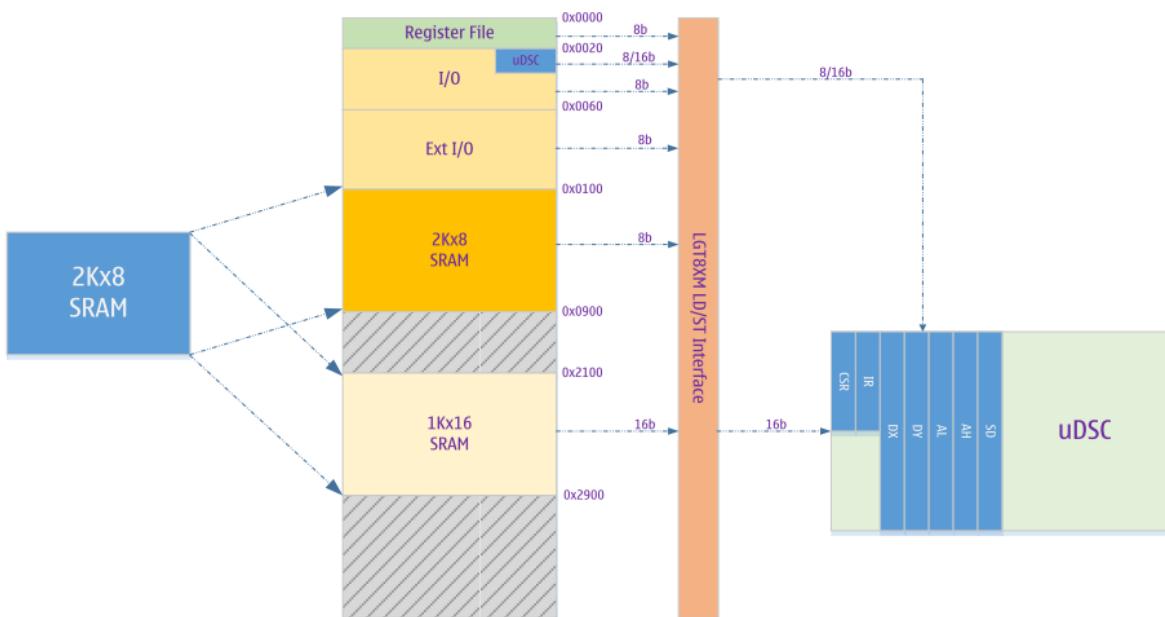


Рис.4.2. Распределение адресов в пространстве ядра LGT8XM.

Как показано на рисунке выше, ядро LGT8XM может напрямую обращаться к

16-разрядным данным между регистрами uDSC DX / DY / DA и SRAM с помощью команды LD / ST. В то же время внутренние регистры uDSC, также сопоставляются с пространством ввода-вывода, доступ к регистру uDSC разделен на 8/16 двух режимов.

Внутреннее дополнение UDSC к работе регистра DX / DY / DA также содержит два дополнительных 8-битных регистра :

Регистр состояния управления UDSC CSR и регистр команд управления IR.

Доступ к CSR / IR возможен только в байтах через пространство ввода/вывода, 16-разрядный режим при доступе к DX / DY / AL / AH. Может использовать IN/OUT и LD/ST/LDD/STD/LDS/STD и другие инструкции для посещения.

Управляющий статус и регистры данных, связанные с uDSC, отображаются в пространство ввода-вывода, прямое использование адресации инструкций IN/OU, могут быть выполнены в течение 8/16-битного доступа к данным.

CSR используется для управления режимом работы uDSC и записи флагов состояния текущих операций выполнения uDSC.

IR-управление uDSC для достижения конкретной операции.

Большинство операций, поддерживаемых uDSC, завершаются за один цикл. Для операции деления требуется семь циклов ожидания. Флаги в регистре CSR также могут определять, завершено ли текущее деление.

Стандартные инструкции LD/ST используют рабочие регистры общего назначения внутри LGT8XM в качестве данных LD/ST и X/Y/Z в качестве адреса назначения.

Когда целевой адрес попадает в 16-битное пространство карты SRAM, значение операнда команды LD / ST изменяется в этой точке, при этом X/Y/Z по-прежнему является целевым адресом.

Значение адресации рабочего регистра общего назначения зависит от режима отображения uDSC **Два способа борьбы**.

Режим отображения uDSC применяется только к адресу доступа 0x2100 ~ 0x28FF.

Режим отображения устанавливается бит 6 (MM) регистра CSR.

Инструкция «LDD Rn, Z + q» в 16-разрядном режиме LD/ST означает, что 16-разрядные данные адреса [Z] загружаются в регистр данных uDSC, а затем значение Z увеличивается на смещение «q».

Здесь значение Rn и режим отображения CSR [MM] соотносятся следующим образом:

LDD Rn, Z/Y+q			
CSR[MM]	[Z+q] Операции Кода операции		
0	0x2100~0x28FF	LDD R0, Z+q	DX = [Z]; Z = Z + q; R0 не изменился
		LDD R1, Z+q	DY = [Z]; Z = Z + q; R1 не изменился
		LDD R2, Z+q	AL = [Z]; Z = Z + q; R2 не изменился
		LDD R3, Z+q	AH = [Z]; Z = Z + q; R3 не изменился
1	0x2100~0x28FF	LDD Rn,Z+q	{Rn} адресуются для DX/DY/AL/AH в области ввода-вывода [DX/DY/AL/AH] = [Z]; Z = Z + q Rn сохраняют неизменными

STD Rn, Z/Y+q			
0	0x2100~0x28FF	STD Z+q, R0	[Z] = DX; Z = Z + q; R0 не изменился
		STD Z+q, R1	[Z] = DY; Z = Z + q; R1 не изменился
		STD Z+q, R2	[Z] = AL; Z = Z + q; R2 не изменился
		STD Z+q, R3	[Z] = AH; Z = Z + q; R3 не изменился
		STD Z+q, R4	[Z] = SD; Z = Z + q; R4 не изменился
1	0x2100~0x28FF	STD Z+q, Rn	{Rn} адресуются для DX/DY/AL/AH/SD в области ввода-вывода [Z] = [DX/DY/AL/AH/SD] адресуемый {Rn} Rn сохраняют неизменными

LDT, LDS/STS в наборе команд LGT8XM могут обращаться к области 0x2100 ~ 0x28FF, но метод адресации LD/STD Y/Z + q более эффективен. LDD/STD адресацию на основе базового адреса мы можем установить Y / Z на базовый адрес данных в ОЗУ. Используемые адресацию LD / STD Y / Z + q, инструкции могут быть выполнены за один цикл , получить доступ к данным и автоматически переместить указатель адреса на следующий целевой адрес.

Режим адресации Y / Z + q смещения команды LDT / STD ядра LGT8XM использует [Y / Z + q] в качестве 8-битного адреса данных для выполнения команды и не увеличивает значение Y / Z после выполнения.

Когда LDD / STD используется для адресации диапазона адресов 0x2100 ~ 0x28FF, изменяется поведение инструкции LDD / STD.

Когда команда выполняется, [Y / Z] используется как адрес для 16-разрядной адресации данных. Увеличьте смещение, указанное «q». Эта функция может повысить эффективность нашей непрерывной адресации путем адресации «q = 2» для непрерывных 16-разрядных данных.

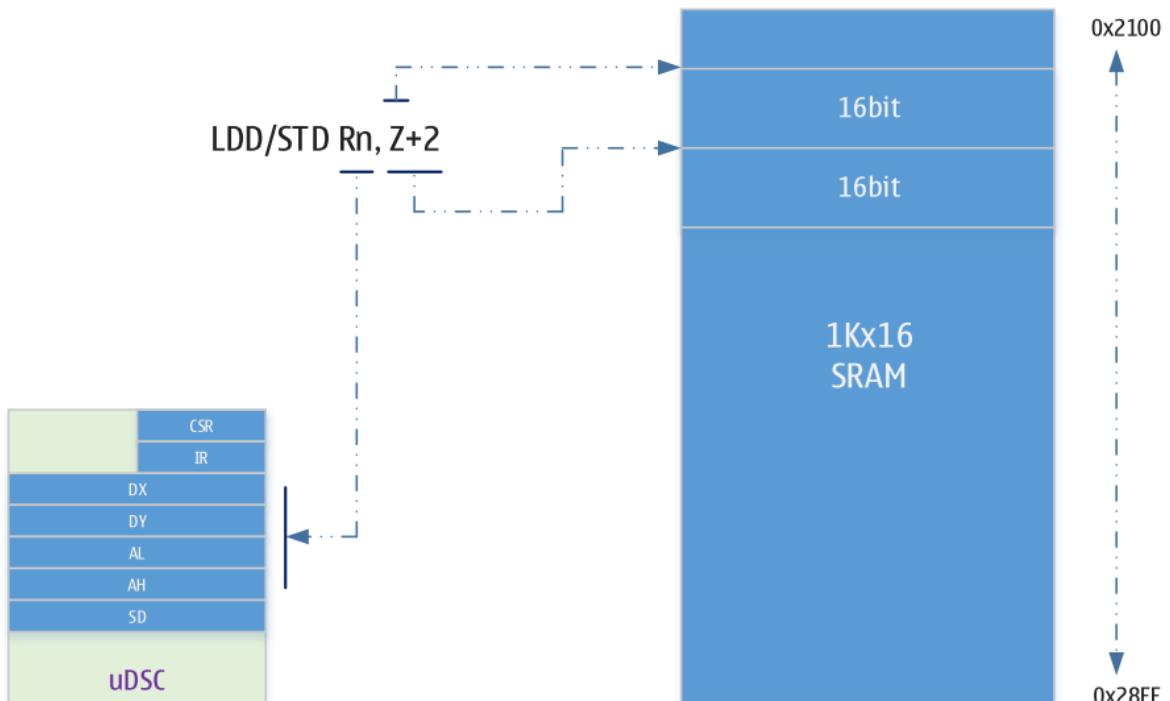


Рис.4.3.

#### 4.2.1. Отображение между адресом переменной и адресом 16-разрядного режима.

LGT8XM - это 8-разрядный процессор с доступом к данным в байтах.

PDN1512A встроенный 2К байт пространства данных. Эта часть пространства отображается на адрес 0x0100 ~ 0x08FF. Компилятор C / C ++ автоматически назначает переменные 0x0100 ~ 0x08FF между ними.

Если мы определим 16-разрядный массив в C / C ++, который требует операции с использованием uDSC, сначала сопоставьте адрес этой переменной с адресной областью (0x2100 ~ 0x28FF) 16-разрядного доступа LD / ST. Нужно увеличить адрес смещения переменной 0x2000.

Программное обеспечение определяет операцию, выполняемую через IR-регистр uDSC.

Все операции uDSC выполняются между DX / DY / DA. Пользователи могут использовать 16-битный LD / ST канал DX / DY / DA и SRAM напрямую и быстро обмениваться данными.

#### *Определение инструкции работы uDSC*

Классификация	IR[7:0]								Функциональное описание
ADD/SUB	0	0	S <sup>1</sup>	0	0	1	0	1	DA = DX + DY
	0	0	S <sup>1</sup>	0	0	0	0	1	DA = DX - DY
	0	0	0	1	1	1	0	1	DA = DY
	0	0	S <sup>1</sup>	1	1	0	0	1	DA = -DY
	0	0	S <sup>1</sup>	1	0	1	1	1	DA = DA + DY
	0	0	S <sup>1</sup>	1	0	0	1	1	DA = DA - DY
MAC/MSC	0	1	S1 <sup>2</sup>	S0 <sup>2</sup>	0	1	0	0	DA = DX * DY
	0	1	S1 <sup>2</sup>	S0 <sup>2</sup>	0	0	0	0	DA = -DX * DY
	0	1	S1 <sup>2</sup>	S0 <sup>2</sup>	1	1	0	0	DA = (DX * DY) » 1
	0	1	S1 <sup>2</sup>	S0 <sup>2</sup>	1	0	0	0	DA = (-DX * DY) » 1
	0	1	S1 <sup>2</sup>	S0 <sup>2</sup>	0	1	1	S	DA = DA + DX * DY
	0	1	S1 <sup>2</sup>	S0 <sup>2</sup>	1	1	1	S	DA = (DA + DX * DY) » 1
	0	1	S1 <sup>2</sup>	S0 <sup>2</sup>	0	0	1	S	DA = DA - DX * DY
	0	1	S1 <sup>2</sup>	S0 <sup>2</sup>	1	0	1	S	DA = (DA - DX * DY) » 1
MISC	1	0	0	0	0	0	0	0	DA = 0
	1	0	0	0	0	1	0	S	DA = NEG(DA)
	1	0	0	0	1	0	0	S	DA = DX^2
	1	0	0	0	1	0	1	S	DA = DY^2
	1	0	1	0	0	0	0	S	DA = ABS(DA)
	1	0	1	1	0	0	0	0	DA = DA/DY
	1	0	1	1	0	0	0	1	DA = DA/DY, DY = DA%DY
SHIFT (СДВИГ)	1	1	0	0	N3	N2	N1	N0	DA = DA « N
	1	1	S	1	N3	N2	N1	N0	DA = DA » N

**Примечание:** 1. S означает, что операция является со знаком операцией или без знака операцией.

2. S1 указывает, является ли DX номером, со знаком, S2 указывает, является

ли DY знаковым числом.

3. N3 ... 0 - это число с четырьмя битами, которое может быть достигнуто Операцией до 15 бит.

4. Указывает, что значение этого бита не имеет смысла, может быть установлен на 0 или 1, рекомендуется установить значение 0.

### 4.3. Регистров Описание

Имя	Адрес вв/вывода	Функциональное описание
DCSR	0x20(0x00)	Регистр состояния управления uDSC
DSIR	0X21(0X01)	Регистр инструкций операций
DSSD	0x22(0x02)	16-разрядный результат переполнения аккумулятора DSA
DSDX	0x10(0x30)	Операнд DSDX, 16-битный доступ для чтения и записи
DSDY	0x11(0x31)	Операнд DSDY, 16-битный доступ для чтения / записи
DSAL	0X38(0X58)	32-р. аккумулятор DSA [15: 0], 16-р. доступ для чтения и записи
DSAH	0x39(0x59)	32-р. аккумулятор DSA [31:16], 16-р. доступ для чтения и записи

## *DSCR - регистр состояния управления*

## *DSIR – операции регистр команд*

### *DSDX - регистр operandов DSDX*

## *DSDY - регистр операнда DSDY*

*DSAL - младшие 16 бит 32-разрядного аккумулятора DA*

*DSAH - старшие 16 бит 32-разрядного аккумулятора DA*

## DSSD - DA Арифметический регистр переполнения

## 4.4. Примеры приложений uDSC

### Пример 1. Основная конфигурация и операции

Ниже приведена простая подпрограмма (AVRGCC), которая реализует 16-битное умножение и возвращает 32-разрядный результат:

```
unsigned long dsu_xmuluu (unsigned short dy, unsigned short dx);
```

Ниже приведен код Ассемблера для функции C:

```
#include "udsc_def.inc" ;определения кода операции
.global dsu_xmuluu ; объявление для вызова из кода С / С ++
dsu_xmuluu:
    out DSDX, r24 ;загрузка DX
    out DSDY, r22 ; загрузка DY
    ldi r20, XMULUU ; загрузка кода операции
    out DSIR, r20 ; умножить
    in r22, DSAL ;{r23, r22} = AL
    in r24, DSAH ;{r25, r24} = AH
    ret
```

## 5. Системные часы и конфигурация

### 5.1. Распределение системных часов

LGT8FX8P поддерживает несколько тактовых входов. Система может работать в трех основных источниках тактового сигнала, а именно на внутреннем 32KHz калибровочном RC-генераторе, внутреннем 32-мегагерцовом калибровочном RC-генераторе и внешнем входном кварцевом генераторе с частотой 400KHz ~ 20MHz.

На Рис.5.1. показано распределение тактовой системы LGT8FX8P. CMU является центром всего управления часами. Он отвечает за разделение системных часов, создание независимых часов для разных модулей и управление часами. В типичном приложении нет необходимости, чтобы все часы работали одновременно. Чтобы снизить энергопотребление системы, модуль управления питанием системы отключает неиспользуемые часы модуля в соответствии с различными режимами ожидания.

Подробные сведения о работе см. В главе «Управление питанием».

#### CPU\_clk

Используется для управления работой ядра LGT8XM и SRAM. Так же управление общим рабочим регистром, регистром состояния.

После того, как такты процессора будут остановлены, ядро не продолжит выполнение инструкций и выполнение вычислений. Система выполняет команду SLEEP, чтобы уснуть.

После этого режима часы ядра будут отключены.

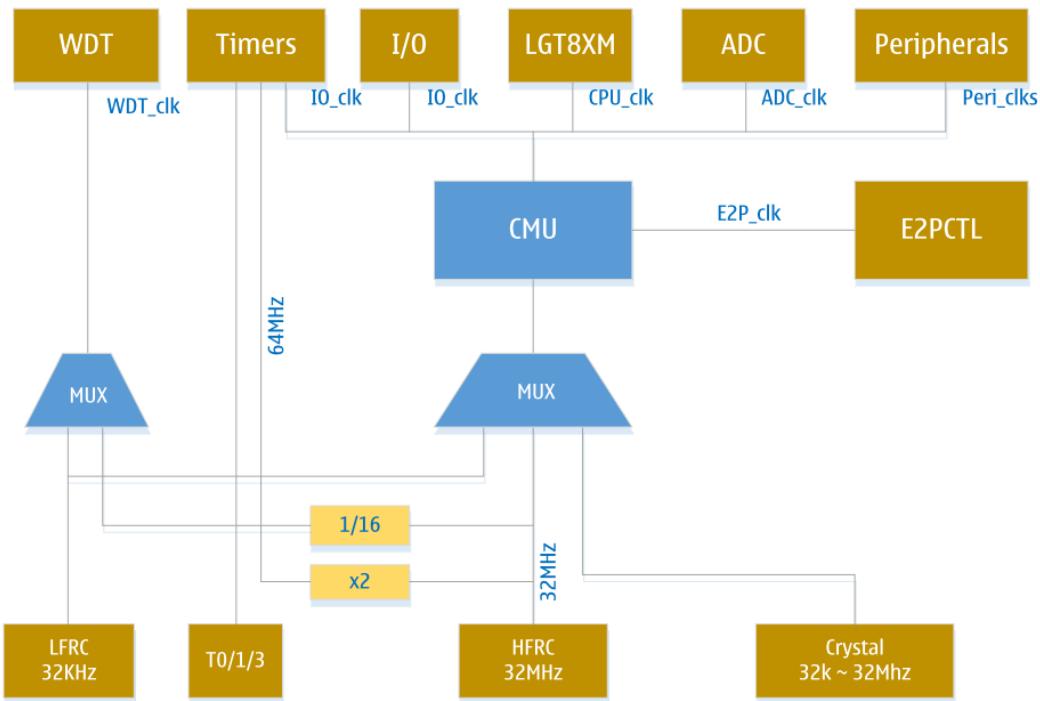


Рис.5.1 Блок – схема распределение тактовой системы LGT8FX8P

### Peri\_clk

Используется для управления большинством периферийных модулей, таких как таймер/счетчик, SPI, USART и т. д. Такты IO также используются для управления внешними Модулями прерывания.

Когда периферийные такты останавливаются из-за спячки, некоторые из периферийных устройств, которые могут быть использованы для пробуждения системы, работают независимо Тактов или асинхронного режима.

Например, функция распознавания адресов TWI может разбудить большинство режимов ожидания, а часть распознавания адресов работает в асинхронном режиме.

### E2P\_clk

Часы E2P\_clk используются для генерации времени доступа к интерфейсу FLASH. E2P\_clk генерирует доступ к E2PCTL для доступа к интерфейсу FLASH Timing. E2P\_clk фиксируется на 32 деления на 1 (1 МГц) от внутреннего 32MHz HFRC-генератора.

Если пользователям необходимо использовать модуль E2PCTL для чтения и записи внутреннего программного FLASH или FLASH-пространства, ему необходимо включить встроенный генератор 32 МГц заранее.

### Asy\_clk

Асинхронный таймер. Таймер / счетчик можно управлять напрямую с помощью внешнего тактового генератора или кварцевого генератора (32,768 К). Этот независимый тактовый режим позволяет таймеру продолжать работать, пока система находится в режиме гибернации.

### WDT\_clk

Внутренний сторожевой таймер источника синхронизации, который может быть сконфигурирован для выбора внутреннего генератора LKRC с частотой 32

кГц или деления на 16 (2 МГц) от внутреннего 32 МГц HFRC.

После того, как система включена, основным источником синхронизации сторожевого таймера является генератор LFRC с частотой 32 кГц.

## 5.2. Выбор источника синхронизации

LGT8FX8P поддерживает четыре типа входа источника синхронизации, пользователь может зарегистрировать регистр PMCR, чтобы включить управление источником синхронизации и завершить главный тактовый переключатель. На Рис. приведена схема структуры управления PMCR:

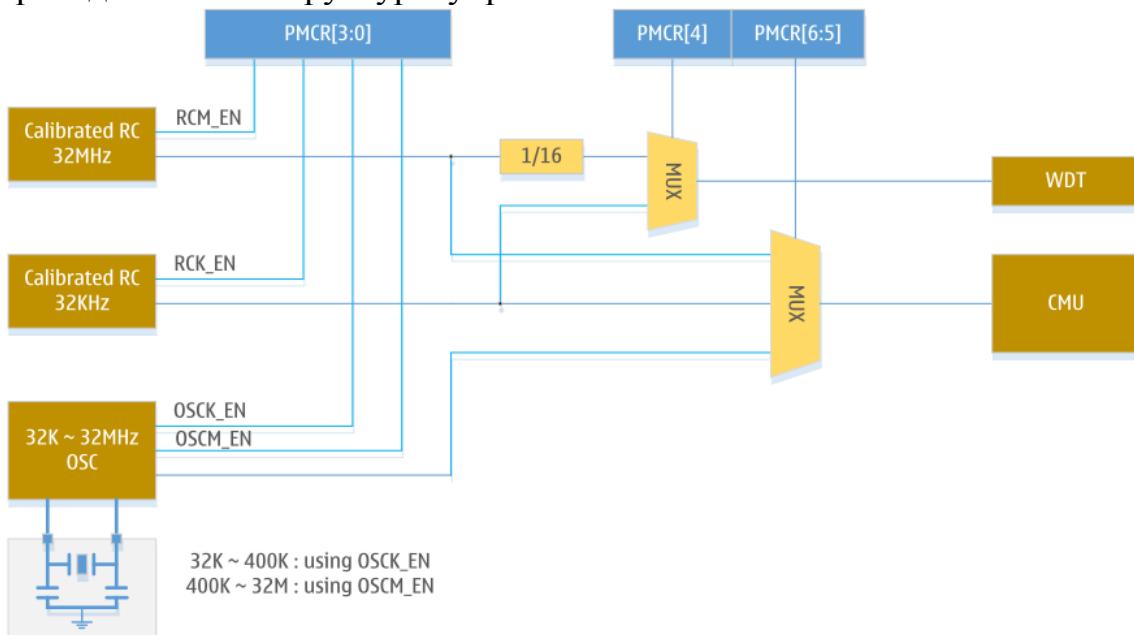


Рис.5.2. Блок – схема выбора источника синхронизации.

Внутренний OSC-генератор LGT8FX8P может работать в режимах с высокой частотой и низкой частотой, пользователю необходимо управлять внутренним OSC-генератором в соответствии с фактическим размером внешнего кристалла в правильном режиме. Тот же внутренний RC-генератор также разделен на два типа высокочастотных и низких частот. Самые низкие 4 бита регистра PMCR управляют этими четырьмя источниками синхронизации. Управляющее соотношение выглядит следующим образом:

PMCR	Соответствующий источник синхронизации
PMCR [0]	32MHz RC Включить управление, 1 Включено, 0 Отключено
PMCR [1]	32KHz RC Включить управление, 1 Включено, 0 Отключено
PMCR [2]	Разрешение 400K ~ 32MHz OSC, 1 разрешение, 0 отключено
PMCR [3]	32K ~ 400K Разрешение режима OSC, 1 разрешение, 0 отключено

Система LGT8FX8P включена, по умолчанию 32 МГц RC в качестве источника системных тактов, основная работа в источнике синхронизации 8 точек (Частота кратная 2 МГц). Пользователь может изменить конфигурацию по умолчанию, установив регистр PMCR и регистр предварительного делителя системы (CLKPR). Если вам нужно изменить конфигурацию основного источника синхронизации, вам необходимо убедиться, что источник синхронизации после

переключения находится в стабильном рабочем состоянии до переключения тактов. Поэтому перед переключением основного источника синхронизации необходимо включить желаемый источник синхронизации через PMCR [3: 0] и дождаться, пока такты стабилизируются перед переключением.

Когда пользователь переключается на внешний ведущий тактовый генератор, хотя пользователь включен внешний кристалл, но не исключает ошибок конфигурации или из-за отказа от кварцевого генератора не может вызвать такты. Если вы переключитесь на внешний кварцевый генератор в это время, система перестанет работать сразу после переключения.

Поэтому, с точки зрения надежности системы, рекомендуется избегать таких проблем с точки зрения разработки программного обеспечения, включив сторожевой таймер.

После того, как источник синхронизации включен и ожидает стабилизации, основной тактовый сигнал может быть переключен PMCR [6: 5]. В котором PMCR [5] используется для выбора внутреннего генератора RC и внешний кристалла, PMCR [6] используется для выбора высокой скорости и источника низкой скорости источника тактовой синхронизации.

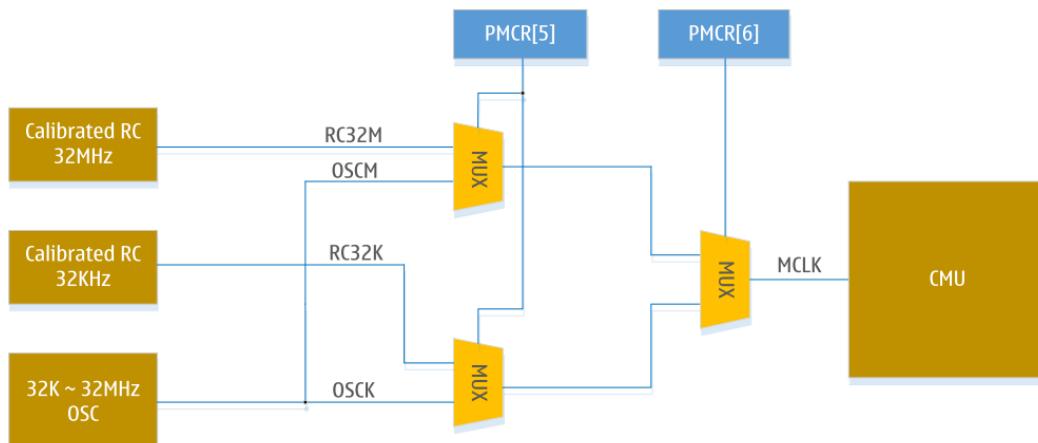


Рис.5.3

#### *Мастер источник выбор синхронизации*

PMCR[6]	PMCR[5]	главные Такты - источник
0	0	внутреннего RC-генератора 32MHz(система по умолчанию)
0	1	внешний 400K ~ 32MHz высокоскоростной осциллятор
1	0	Внутренняя 32 кГц RC-генератор
1	1	внешний 32K ~ 400 кГц низкая скорость кварцевого генератора

Тактового сигнала, чтобы контролировать время для защиты PMCR реестр был случайно изменен, и PMCR регистрировать изменения нужно строго установлены определенные сроки.

В регистре PMCR, старший бит(PMCR[7]), используемых для осуществления контроля времени. Пользователь перед изменением в PMCR другие биты, надо сначала в PMCR[7] установить «1», после этой операции в течении 6 циклов тактов, изменять значения других бит регистра PMCR. 6 циклов После того, как будет изменена прямая модификация PMCR.

Ниже приведен пример перехода на внешний высокоскоростной кристалл, в котором перечислены рекомендуемые шаги:

(1) включить источник синхронизации

- набор PMCR[7] = 1
- в шесть цикла набор PMCR[2] = 1, включение внешнего скоростного режима внешнего кварцевого генератора
- ждать внешнего кристалла стабильный(время ожидания, потому что осциллятор колеблется, в целом, нам приходится ждать) (2)  
Выключатель основного источника синхронизации
- набор PMCR[7] = 1
- в шесть цикла набор PMCR[6:5] = 01, часы системы автоматически переключается на внешний кварцевый генератор
- выполнить несколько НОП операций, улучшить стабильность работы (Дополнительная работа)

**Примечание:** В приведенной выше операции для переключения основных тактов, чтобы обеспечить правильную работу текущих системных тактов, переключатель на внешний кристалл

Можно только отключить предыдущий внутренний RC-генератор.

### 5.3. Управление предделителя системных часов

LGT8FX8P имеет встроенный системный генератор тактовых импульсов, который может управляться регистром предварительного делителя часов (CLKPR).

Эта функция может использоваться для снижения энергопотребления системы, когда система не требует очень высокой вычислительной мощности. Настройка предварительного делителя действительна для источников синхронизации, поддерживаемых системой. Предделитель тактов может влиять на такты выполнения ядра и, следовательно, синхронизировать периферийные устройства.

При переключении между различными настройками предварительного делителя тактов предварительный каскад системных часов гарантирует, что во время переключения не произойдет никаких сбоев, но будет гарантировать, что промежуточное состояние сверхчастоты не будет.

Переключение частотного деления выполняется немедленно. После того, как смена регистра вступает в силу, системные такты переключаются на новые тактовые частоты с частотой от 2 до 3 текущих тактовых циклов системы.

Чтобы избежать ошибочной работы регистров деления времени, для изменений CLKPR также необходимо соблюдать специальную последовательность синхронизации.

- Установите бит разрешения изменения предварительного делителя часов (CLKPCE) в «1» и CLKPR в «0» в противном случае.
- Требуемое значение цикла записывается в CLKPS, CLKPCE write 0

Перед изменением регистра предделителя тактов вам необходимо отключить функцию прерывания, чтобы гарантировать, что время записи может быть завершено.

Для конкретного определения основного генератора тактовых импульсов CLKPR см. Раздел «Описания регистров» в этой главе.

## 5.4. Калибровка внутреннего RC-генератора

LGT8FX8P включает в себя два калибруемых RC-генератора внутри, которые могут быть откалиброваны с точностью  $\pm 1\%$ . 32-мегагерцовый RC по умолчанию установлен на системные такты.

Предварительно подготовленные, внутренние 32MHz HFRC и 32KHz LFRC LGT8FX8P калибруются, а значение калибровки записывается в область информации о конфигурации системы. При отключении питания эти калибровочные значения будут считываться во внутренние регистры и перекалибровать частоту RC через регистры.

Регистр калибровки расположен в адресном пространстве ввода-вывода, и программа пользователя может читать и писать. Для приложений, требующих специальной частоты, частоту внутреннего генератора можно регулировать, изменяя регистр калибровки. Изменение калибровочного регистра не изменяет информацию о заводской конфигурации, не включается система или перезагрузка пользовательского режима перезагрузки, регистр калибровки будет восстановлен до заводских настроек.

## 5.5. Регистры Описание

## *RCMCAL – 32-мегагерцовый регистр калибровки осциллятора HFRC*

RCMCAL - регистр калибровки HFRC на 32 МГц		
RCMCAL: 0x66		По умолчанию: заводская настройка
Биты	RCCAL[7:0]	
R/W	R/W	
<b>Определение бит</b>		
[7:0]	RCCAL	После включения системы значение регистра будет заменено калибровочным значением RC в информации о конфигурации системы.

## *RCKCAL - 32KHz RC калибровочный регистр*

RCKCAL - 32KHz калибровочный регистр RC		
RCMCAL: 0x67		По умолчанию: заводская настройка
Биты	RCCAL[7:0]	
R/W	R/W	
<b>Определение бит</b>		
[7:0]	RCKCAL	Запишите калибровочное значение в регистр RCKCAL, чтобы завершить калибровку 32KHz RC-генератора

*PMCR - Регистр управления синхронизацией тактов*

Определение бит		
[0]	RCMEN	Внутреннее 32-мГц управление мощностью генератора RC, 1 - включено, 0 - отключено
[1]	RCKEN	Встроенный 32KHz RC-генератор позволяет управлять, 1 - включить, 0 - отключить
[2]	OSCMEN	Управление внешним высокочастотным кварцевым генератором, 1 - разрешение, 0 - отключено
[3]	OSCKEN	Управление внешним низкочастотным кварцевым генератором, 1 - разрешение, 0 - отключено
[4]	WCLKS	Выбор источника тактов сторожевого таймера WDT, 0 - выбор 16 делителей для внутреннего 32MHz генератора HFRC 1 - Внутренний генератор LKRC с частотой 32 кГц
[5]	CLKSS	Главный элемент выбора источника синхронизации, выберите тип источника синхронизации, пожалуйста, обратитесь к разделу выбора источника синхронизации
[6]	CLKFS	Регулятор частоты основного источника синхронизации, выберите тип тактовой частоты, см. Раздел выбора источника синхронизации
[7]	PMCE	Контрольный бит разрешения изменения регистра PMCR. Этот бит должен быть установлен первым, прежде чем менять другие биты PMCR, а затем остальные биты устанавливаются в четыре цикла.

## *CLKPR - регистр предделителя тактов*

## CLKPR - Регистр генератора тактов

## 6. Управление питанием

### 6.1. Обзор

Спящий режим снижает энергопотребление системы за счет выключения системных тактов и модуля синхронизации.

LGT8FX8P обеспечивает очень гибкий режим сна и контроллер модуля, позволяет пользователям реализовать идеальную маломощную конфигурацию для их применения.

LGT8FX8P автоматически не отключает аналоговые функциональные модули, такие как АЦП, ЦАП, компараторы (AC), LVD с низким напряжением и т. п.

При входе в спящий режим программное обеспечение необходимо отключить перед сном в соответствии с требованиями приложения Имитация и восстановление правильного состояния после просыпания системы.

LGT8FX8P поддерживает различные режимы спящего режима, включая режим шумоподавления, специфичный для АЦП, который устраниет цифровую часть помех источника питания АЦП при преобразовании АЦП. Кроме того, другие - режимы управления мощностью, которые делятся на пять типов:

Режим сна	Описание Функции
Режим ожидания (IDLE)	Просто отключите основные часы, другие периферийные модули работают, все допустимые источники прерываний могут разбудить ядро
Режим энергосбережения (Save)	То же, что и режим DPS0, режим сохранения совместим с LGT8FX8D
Режим пониженного энергопотребления (DPS0)	Как и в режиме сохранения (Save), поддерживаются источники пробуждения: <ul style="list-style-type: none"><li>• Все изменения выводов</li><li>• Пробуждение от сторожевого таймера</li><li>• Пробуждение от TMR2 в асинхронном режиме</li></ul>
Режим пониженного энергопотребления (DPS1)	Отключите все внутренние и внешние генераторы, включите источник пробуждения: <ul style="list-style-type: none"><li>• Все внешние изменения уровня внешних контактов</li><li>• Внешнее прерывание 0/1</li><li>• Сторожевой таймер работает при 32K LFRC</li></ul>
Режим пониженного энергопотребления (DPS2)	Выключение основного и минимального режимов питания, включает в себя следующее: <ul style="list-style-type: none"><li>• Внешний сброс</li><li>• Изменение выводов PORTD</li><li>• Воспроизведение LPRC (128 мс / 256 мс / 512 мс / 1 с) Обратите внимание, что пробуждение от DPS2 и сброс при включении, тот же самый</li></ul>

LGT8FX8P поддерживает глубокий сон DPS2, в этом режиме внутренний LDO системы находится в состоянии пониженного энергопотребления, основные регистры, все периферийные контроллеры и SRAM находятся в состоянии отключения питания, данные не поддерживаются. Ячейка памяти FLASH также будет находиться в состоянии пониженного энергопотребления, поэтому режим

DPS2 может обеспечить минимальное энергопотребление системы.

Режим отключения питания может быть активирован через изменение вывода порта D (PORTD) или 5-уровневый таймер. Таймер при DPS2 для пробуждения не поддерживает калибровку с точностью около 15% и подходит только для часов с пробуждением с низкой точностью.

Система пробуждается из режима DPS2, сначала включит LDO, процесс и включение одного и того же процесса. Чип выполнит полный запуск сброса при включении, загрузит информацию о конфигурации, а затем запустит программу с адреса, на который указывает вектор сброса.

В режимах, отличных от DPS2, внутреннее питание не отключается, и вся информация о регистре и данные ОЗУ не теряются во время спящего режима. После пробуждения ядро возобновляет выполнение из последней инструкции перед сном.

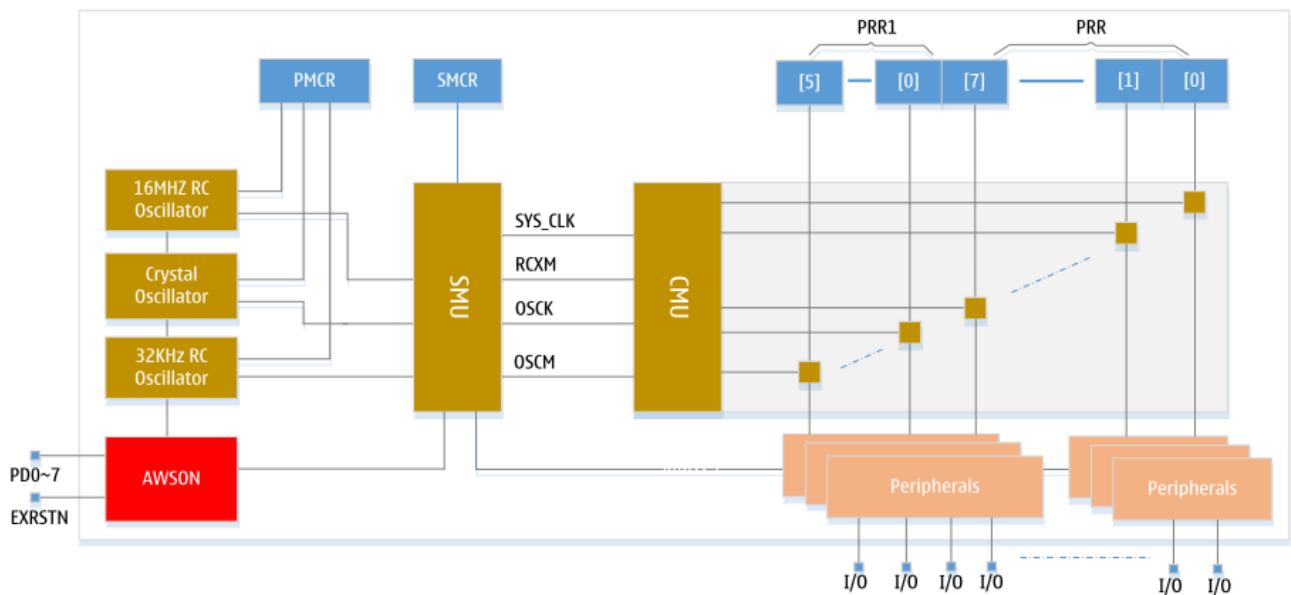


Рис.6.1. Схема управления питанием системы.

Как было показано выше, LGT8FX8P первичного управления энергопотребление всей системой контроллера спящего режима (SMU) и блок управления тактами (CMU). От уровня экономии энергии, можно разделить на четыре уровня питания:

- первый уровень работает модуль управления синхронизацией через регистр PRR, часы не используется, чтобы закрыть модуль, сохраняя динамическую работу системы питания. В нормальных условиях этот уровень может экономить энергию, не очевиден.

- второй уровень является основным источником синхронизации при переключении с часами низкой частоты, и модуль источника тактовых импульсов не закрыт, и использовали другие аналоговые модули, этот режим может быть, по существу, получен весьма значительный расход рабочей мощности от потребляемой мощности системы и сна.

- третий уровень представляет собой систему, войдя в режим пониженного энергопотребления (DPS1), LGT8FX8P может быть достигнуто в полярных DPS1 режиме ожидания мощности, режим пробуждения после сбоя питания, состояние сброса, прежде чем программное обеспечение может быть прочитано MCUSR регистра.

- четвертый уровень - это режим пониженного энергопотребления (DPS2), который отключает основной источник питания для достижения наименьшего

энергопотребления системы. При выключенном ядре все данные в этом режиме будут потеряны. Просыпайтесь сразу же после выполнения процесса сброса при включении, система перезагружается из вектора сброса для запуска.

## 6.2. Управление питанием AWSOM

По сравнению с LGT8FX8D режим пониженного энергопотребления DPS2 - совершенно новый режим питания.

Режим DPS2 используется для приложений, которые требуется более высокое потребление энергии сна. После входа в режим DPS2 система поддерживает только статический модуль (AWSOM) в рабочем состоянии, другие цепи находятся в полностью выключенном состоянии.

Модуль AWSOM предназначен для управления сна и пробуждения, ответственного за режим DPS2. Модуль AWSOM состоит в основном из логики управления пробуждением IO и маломощного LPRC.

Программное обеспечение может управлять AWSOM через регистр IOCWK и регистр DPS2R. Регистр IOCWK используется для управления пробуждением изменений PD0 ~ 7. Регистр DPS2R управляет режимом DPS2 и функциональным режимом LPRC.

Для получения дополнительной информации см. Раздел определения регистров в конце этого раздела.

Перед использованием режима DPS2 программное обеспечение устанавливает IOCWK для включения требуемого входа IO или включает LPRC через регистр DPS2R и настраивает период пробуждения, затем включает режим DPS2 через бит DPS2EN в регистре DPS2R. После настройки программное обеспечение должно установить режим спящего режима DPS2 через регистр SMCR, а затем выполнить команду SLEEP для перехода в спящий режим.

## 6.3. Спящий режим и источник пробуждения

LGT8FX8P поддерживает пять видов спящего режима, пользователь может выбрать соответствующий режим ожидания в соответствии с требованиями приложения.

Регистр SMCR содержит параметры управления режимом спящего режима. После выполнения инструкции SLEEP ядро переходит в спящий режим. Для лучшего потребления энергии сна рекомендуется отключить все неиспользованные часы и аналоговые модули, прежде чем ядро переходит в спящий режим.

Однако следует отметить, что некоторым источникам пробуждения нужны рабочие часы, и если такие источники пробуждения должны использоваться, храните соответствующий источник синхронизации.

Если вам нужно войти в пять режимов спящего режима, бит SE в SMCR должен быть установлен в «1», чтобы включить управление спящим режимом. Тогда выполнить команду SLEEP можно.

SM0 / 1/2 в SMCR используется для выбора другого спящего режима. Пожалуйста, обратитесь к следующему описанию для получения конкретной информации

## Спящий режим и режим ожидания:

Режим сна	Действительные такты					Источник пробуждения						
	Такты ядра	Периферийные такты	Такты ADC	Асинхронные такты	Изменения уровня контакта	Внешнее прерывание 0/1	Соответствие адреса TWI	Прерывание TMR2	АЦП - конец преобразования	Сторожевой таймер	Периферийное прерывание	Изменение уровня PD
Режим ожидания (IDLE)	X	X	X	X	X	X	X	X	X	X	X	X
АЦП Подавление шума		X	X	X	X	X	X	X	X	X		X
Режим энергосбережения (SAVE)				X	X	X	X	X		X		X
Режим пониженного энергопотребления (DPS0) (с RC32K)				X	X	X		X		X		X
Режим пониженного энергопотребления (DPS1) (без RC32K)				X	X	X		X				X
Режим пониженного энергопотребления (DPS2) (без LDO)												X

Когда MCU находится в спящем режиме, если источник пробуждения активен, MCU просыпается после 4 циклов и продолжает выполнять инструкции.

Если прерывание остается действительным, прерывание также немедленно ответит на процедуру обслуживания прерываний.

Если в режиме SLEEP происходит сброс системы, MCU также просыпается и выполняется из вектора сброса.

Когда MCU находится в режиме питания / выключения, система может активироваться внешним прерыванием INT0/1. После пробуждения MCU возобновит работу с предыдущего положения сна.

### 6.3.1. Режим ожидания (IDLE)

Когда SM2...0 установлен на 000, после выполнения команды SLEEP MCU переходит в режим ожидания, а режим IDLE отключает рабочие такты ядра. Все остальные периферийные устройства могут работать нормально.

Режим IDLE может просыпаться от внешнего и внутреннего прерывания. Если вам не нужно использовать компаратор и АЦП в качестве источника пробуждения, рекомендуется отключить его.

IDLE, закрыл только такты, запускающие ядро, поэтому он не может значительно снизить энергопотребление.

В IDLE, ядро также перестанет извлекать и выполнять инструкции, поэтому вы можете уменьшить внутреннюю потребляемую мощность рабочей программы

## FLASH.

Тем не менее, режим IDLE имеет более гибкий режим пробуждения, пользователи могут уменьшить основные такты системы и отключить нежелательные модули для уменьшения энергопотребления.

### 6.3.2. Режим подавления шума АЦП

Когда SM2 ... 0 установлен на 001, MCU переходит в режим уменьшения уровня АЦП после выполнения инструкции SLEEP. В этом режиме, Ядро и большинство периферийных устройств перестанут работать.

АЦП, внешние прерывания, соответствие TWI-адреса, WDT и счетчик таймера/события 2, работающие в режиме асинхронных часов, будут работать нормально. Режим всегда используется для обеспечения хорошей рабочей среды для преобразования ADC. Цифровой модуль уменьшает высокочастотные помехи аналогового преобразования. После входа в этот режим АЦП автоматически запустит преобразование выборки.

После преобразования преобразованных данных в регистр данных АЦП, преобразование АЦП завершается, и MCU просыпается из режима шума АЦП.

### 6.3.3. Режим энергосбережения (SAVE)

Когда SM2...0 установлен в 010, MCU переходит в режим сохранения после выполнения инструкции SLEEP.

В этом режиме система отключит рабочие такты всех модулей. Этот режим может просыпаться только в асинхронном режиме, так как такты всех модулей выключены, а сигналы пробуждения в этом режиме могут генерироваться внешними прерываниями, соответствием TWI-адреса и WDT, работающими в режиме независимого источника синхронизации.

Этот режим отключает все модули, кроме основного источника синхронизации. Чтобы достичь лучшего энергопотребления, рекомендуется переключать основные часы системы на внутренний 32K RC или внешний 32KHz низкочастотный кристалл перед входом в этот режим, а затем отключать неиспользуемый источник синхронизации и аналоговый модуль.

### 6.3.4. Режим пониженного энергопотребления DPS0

Когда SM [2: 0] установлено на 110, MCU переходит в режим DPS0 после выполнения инструкции SLEEP. После входа в DPS0, кроме внутреннего 32KHz RC, другие источники синхронизации закрыты. Этот режим может быть вызван внешним прерыванием INT0/1, если функция прерывания WDT включена, она может активироваться WDT.

### 6.3.5. Режим пониженного энергопотребления DPS1

Когда SM [2: 0] установлено в 011, MCU переходит в режим DPS1 после

выполнения инструкции SLEEP. После ввода DPS1 все источники синхронизации системы закрыты. Этот режим может использовать изменения уровня ввода-вывода, сторожевой таймер.

### 6.3.6. Режим пониженного энергопотребления DPS2

Когда SM [2: 0] установлено в 111 и включает модуль AWSN через DPS2EN в регистр DPSR2.

После выполнения инструкции SLEEP DPS2 переходит в режим DPS2.

После входа в режим DPS2 система отключает питание ядра. Таким образом, данные регистра и ОЗУ будут потеряны. Пробуждение от DPS2 такое же, как процедура сброса при включении.

**DPS2, напряжение ядра отключается, информация о регистре теряется, поэтому состояние управления портом будет восстановлено до состояния входа, все выходы ввода-вывода и выключение управления также будут отключены.**

### 6.3.7. Управление мощностью FLASH и быстрое пробуждение

Когда система находится в режиме SLEEP, ядро не будет продолжать выполнять инструкции, на этот раз вы можете отключить питание FLASH, чтобы получить более низкое энергопотребление в режиме ожидания. Эта функция может управляться битом FPDEN в регистре MCUCR.

В режиме пониженного энергопотребления система может использовать внешние прерывания или пробуждение WDT.

Для фильтрации возможных помех от внешних сигналов внутренняя схема пробуждения включает настраиваемую схему фильтра. В соответствии с необходимостью выбрать соответствующую ширину фильтра. Конфигурация схемы фильтра может быть реализована FWKPN в регистре MCUCR.

*MCUCR [FWKPN] Управление шириной фильтра*

FWKPN	Ширина фильтра
0	260us (по умолчанию)
1	32us

### 6.3.8. Регистры Описание

*SMCR - Регистр управления спящим режимом*

SMCR - регистр управления спящим режимом					
SMCR: 0x33(0x53)		По умолчанию: 0x00			
Биты		SM2	SM1	SM0	SE
R/W	-	R/W	R/W	R/W	R/W
Определение бит					
[0]	SE	Спящий режим включает контрольный бит в «1», выполнение инструкции SLEEP, ядро переходит в спящий режим. Бит SE защищает систему от случайного включения спящего режима. После пробуждения рекомендуется немедленно очистить бит SE.			

		Выбор режима ожидания			
[3:1]	SM	SM2	SM1	SM0	Описание режима
		0	0	0	Режим IDLE
		0	0	1	АЦП режим подавления шума
		0	1	0	Режим энергосбережения (SAVE)
		0	1	1	Режим DPS1
		1	1	0	Режим DPS0
		1	1	1	Режим DPS2
		Другие		Зарезервировано	
[7:4]	-	Зарезервировано			

*PRR - Регистр управления энергосбережением*

*PRRI - Регистр управления энергосбережением*

## *MCUCR - Регистр управления MCU*

MCUCR - регистр управления MCU								
MCUCR: 0x35(0x55)				По умолчанию: 0x00				
MCUCR	FWKEN	FPDEN	EXRFD	PUD	IRLD	IFAIL	IVSEL	WCE

R/W	R/W	R/W	R/W	R/W	W/0	R/0	R/W	R/W
<b>Определение бит</b>								
[0]	WCE	До изменения бит MCUCR, нужно сначала установить этот бит в «1», Затем регистр MCUCR обновляется в течение 6 циклов						
[1]	IVSEL	Бит выбора вектора прерывания. Когда IVSEL установлен в «1», адрес вектора прерывания будет сопоставлен с новым адресом на основе значения регистра IVBASE						
[2]	IFAIL	Флаг сбоя загрузки конфигурации системы, 0 - Контрольная сумма в норме, 1 - Ошибка загрузки						
[3]	IRLD	Запись «1» будет перезагрузка информации о конфигурации системы						
[4]	PUD	Глобальный бит блокировки подтягивающих резисторов. 0 - Включить управление подтягиванием, 1 - Отключить подтягивания всех ИО						
[5]	EXRFD	Отключить бит внешнего сброса. 0 - (190us) цифровой фильтр, который обеспечивает внешний сброс, 1 - отключение схемы цифрового фильтра для внешнего сброса						
[6]	FPDEN	Управление включением/выключением питания FLASH. 0 - FLASH остается включенным после системы SLEEP, 1 - FLASH отключается после системы SLEEP						
[7]	FWKEN	Режим активизации режима пробуждения, только режим питания/ выключения. 0 - задержка фильтра 260us. 1 - задержка фильтра 32us.						

### IOCWK - Регистр управления изменениями на уровне группы PD

IOCWK - PD Group Change Регистрация управления пробуждением								
IOCWK: OxAE		По умолчанию: 0x00						
Биты	IOCD7	IOCD6	IOCD5	IOCD4	IOCD3	IOCD2	IOCD1	IOCD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Определение бит</b>								
[ [7:0] ]	IOCWK	Установите соответствующий бит, включите функцию пробуждения PIN- кода PD-группы IO						

### DPS2R - Регистр управления режимом DPS2

DPS2R - регистр управления режимом DPS2								
DPS2R: 0xAF				По умолчанию: 0x00				
Биты	-	-	-	-	DPS2E	LPRCE	TOS1	TOS0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
<b>Определение бит</b>								
[1:0]	TOS	Настройки пробуждения таймера LPRC: 00 - 128ms, 01 - 256ms, 10 - 512ms, 11 - 1s						
[2]	LPRCE	Управление разрешением LPRC. 0 - отключить таймер LPRC, 1 - включить таймер LPRC						
[3]	DPS2E	Режим DPS2 включение управления. 0 - отключен режим DPS2, 1 - включен режим DPS2						
[7:4]	-	Зарезервировано						

## 7. Система управления и сброса

### 7.1. Обзор

После сброса системы все регистры ввода-вывода устанавливаются в начальные значения, и программа начинает выполнение с вектора сброса.

LGT8FX8P, должна использовать RJMP - относительный переход команды перехода к обработчику сброса. Если программа бесполезна для использования прерывания, источник прерывания не включен, вектор прерывания не будет использоваться, область векторов прерываний может использоваться для хранения кода программы пользователя.

После сброса все порты ввода / вывода сразу переходят в исходное состояние. Большинство входов / выходов инициализируются для ввода и выключения внутреннего подтягивающего резистора. I / O с функцией аналогового ввода также инициализируется цифровой функцией ввода / вывода.

Когда сброс становится недействительным, начинает запускаться внутренний таймер LGT8FX8P, используемый для расширения сброса. Ширина сигнала сброса используется для обеспечения того, чтобы источник питания в системе, такты и другие модули вошли в стабильное состояние.

### 7.2. Источники Сброс.

LGT8FX8P поддерживает в общей сложности шесть источников сброса:

- Сброс при включении: Срабатывание при включении питания, когда рабочее напряжение системы ниже порог сброса внутреннего модуля POR.
- Внешний сброс: импульс низкого уровня с определенной шириной на выводе внешнего вывода сброса микросхемы, действительный внешний сброс.
- Сброс от сторожевого таймера: после включения модуля сторожевого таймера система будет сброшена, если таймер сторожевого таймера переполнится.
- Сброс низкого напряжения: в LGT8FX8P находится модуль обнаружения низкого напряжения (LVD). MCU также будет сброшен, когда напряжение питания системы ниже порога сброса, установленного LVD.
- Сброс программного обеспечения: LGT8FX8P - специальный зарегистрированный программный регистр сброса, пользователь может сбросить MCU в любое время через этот регистр.
- Сброс OCD: сброс OCD выдается модулем отладчика, который используется для прямого сброса ядра MCU.

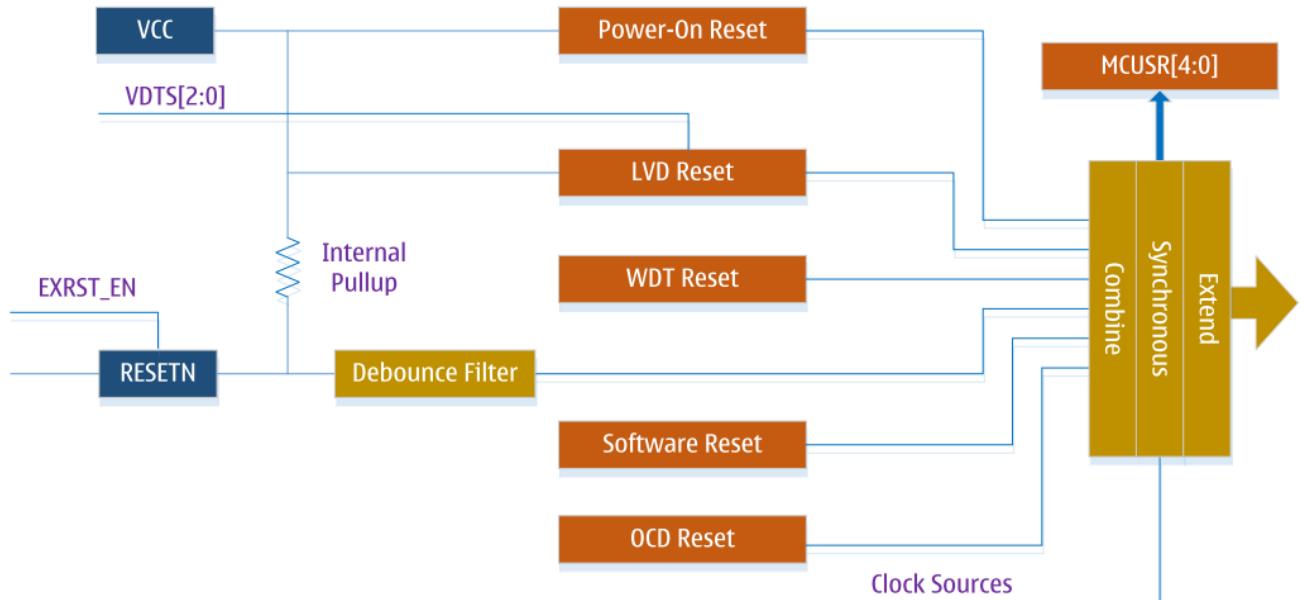


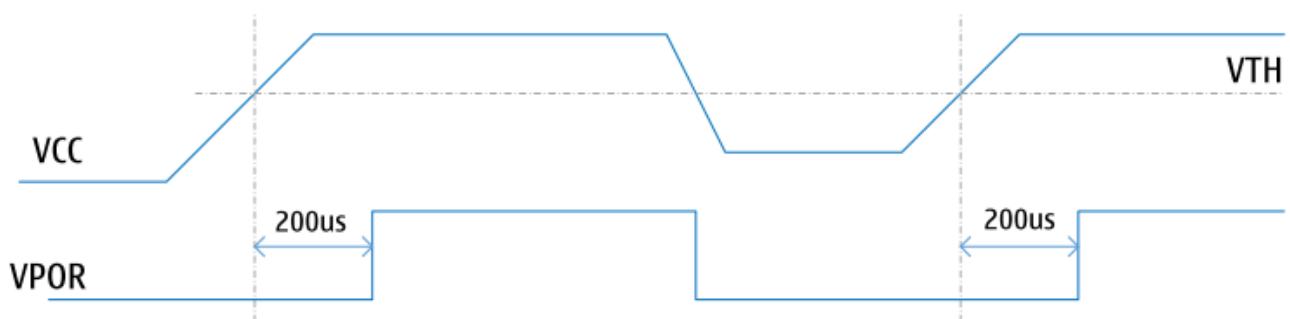
Рис.7.1. Структура системы Сброс.

### 7.2.1. Сброс при включении питания.

Сигнал сброса питания, генерируемый схемой обнаружения внутреннего напряжения. Когда питание системы (VCC) ниже порога обнаружения, действует сигнал сброса включения питания.

Порог обнаружения сброса при включении питания, см. Раздел электрических параметров.

Схема сброса включения питания гарантирует, что чип находится в состоянии сброса во время включения питания и что чип способен работать от известного, стабильного состояния при включении питания. Сигнал сброса при включении также будет расширен с помощью встроенного счетчика, чтобы гарантировать, что различные внутренние аналоговые модули, такие как RC-генераторы, могут войти в стабильное рабочее состояние после включения питания.



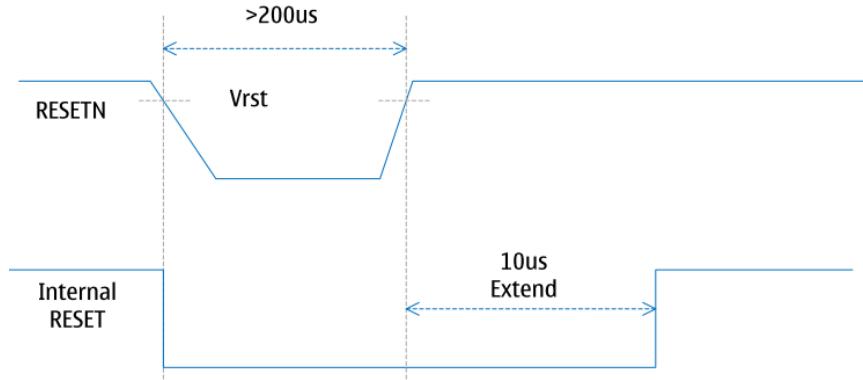
### 7.2.2. Внешний сброс

Применяя низкий уровень к выходу внешнего сброса (RSTN), внешний сброс активируется немедленно.

Длительность низкого уровня больше минимальной требуемой ширины импульса. Внешний сброс - это асинхронный сброс, и внешний сброс может перезагрузить чип, даже если чип не синхронизирован.

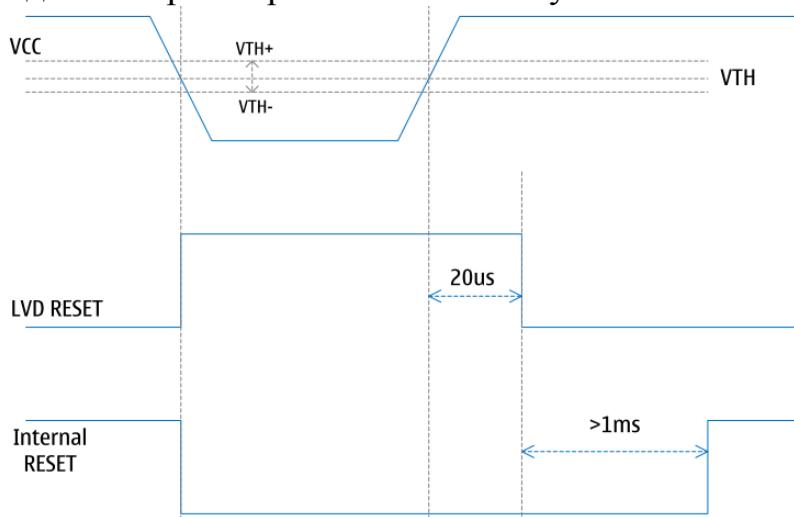
Внешний вывод сброса LGT8FX8P также может использоваться как входы / выходы общего назначения. После включения микросхемы по умолчанию используется функция внешнего сброса. Пользователи могут настраивать по регистру, отключать функцию внешнего сброса этого вывода, которая может использоваться как обычный I / O.

Для получения дополнительной информации см. Описание регистра IOCR.



### 7.2.3. Обнаружение низкого напряжения (LVD) Сброс

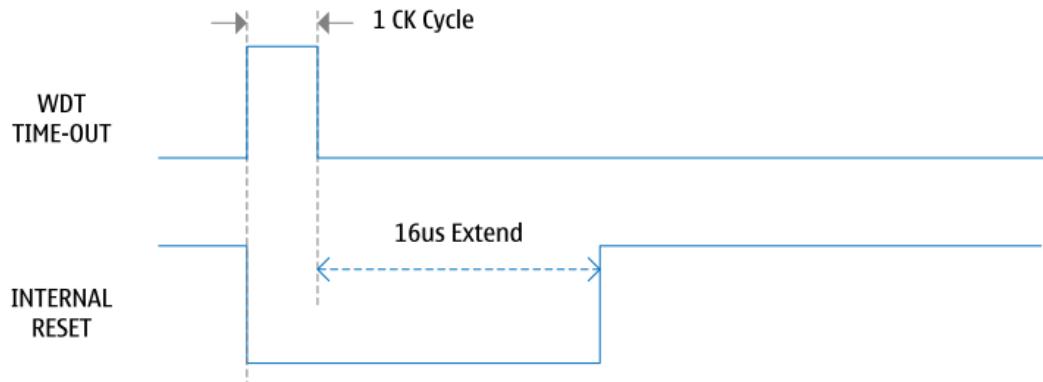
LGT8FX8P включает в себя программируемую схему обнаружения низкого напряжения (LVD). LVD также обнаруживает изменение напряжения VCC, но отличается от сброса включения питания, тем что LVD можно выбрать порог напряжения через регистр VDTCR. Схема обнаружения напряжения LVD имеет характеристику гистерезиса от  $\pm 10$  мВ до  $\pm 50$  мВ для фильтрации дрожания напряжения VCC. Когда LVD включен, сброс LVD будет активен немедленно, если напряжение VCC падает ниже порога установленного сброса. Когда VCC увеличивается выше порога сброса, включается схема расширения внутреннего сброса, и сброс продолжает расширяться как минимум на 1 миллисекунду.



### 7.2.4. Сброс сторожевого таймера

Когда таймер сторожевого таймера переполняется, если включена функция сброса системы сторожевого таймера, сразу же генерируется периодический сигнал сброса системы. Сигнал сброса сторожевого таймера также будет расширен внутренним счетчиком задержки.

Подробнее о работе сторожевого контроллера см. Подробное описание ниже.



### 7.2.5. Сброс программного обеспечения, сброс OCD

Сброс программного обеспечения запускается пользователем, управляя шестым битом регистра VDTCR. Время сброса программного обеспечения точно такое же, как сброс сторожевого таймера. Сигнал сброса будет внутренне растянут 16us. Сброс OCD, сгенерированный внутренним блоком отладчика, сброс OCD обычно контролируется отладчиком, пользовательское программное обеспечение не может инициировать сброс OCD.

## 7.3. Сторожевой таймер

- Выбирается пользователем внутренний генератор 32K RC или 32MHz RC 16-делением (2MHz)
- Режим прерывания поддержки, режим сброса и режим прерывания сброса
- Таймер максимальное время до 8 секунд

LGT8FX8P включает встроенный модуль сторожевого таймера (WDT). Такты WDT могут работать от внутренних 32KHz RC-генератором, или с делением на 16 от внутреннего 32-мегагерцового генератора.

Переполнение счетчика WDT После выхода вы можете вывести сигнал прерывания или сброса системы.

При нормальном использовании программное обеспечение должно выполнять WDR - команда сброса сторожевого таймера перезапускает счетчик перед переполнением. Если система не выполняет даже инструкцию WDR, WDT будет генерировать прерывание или сброс системы.

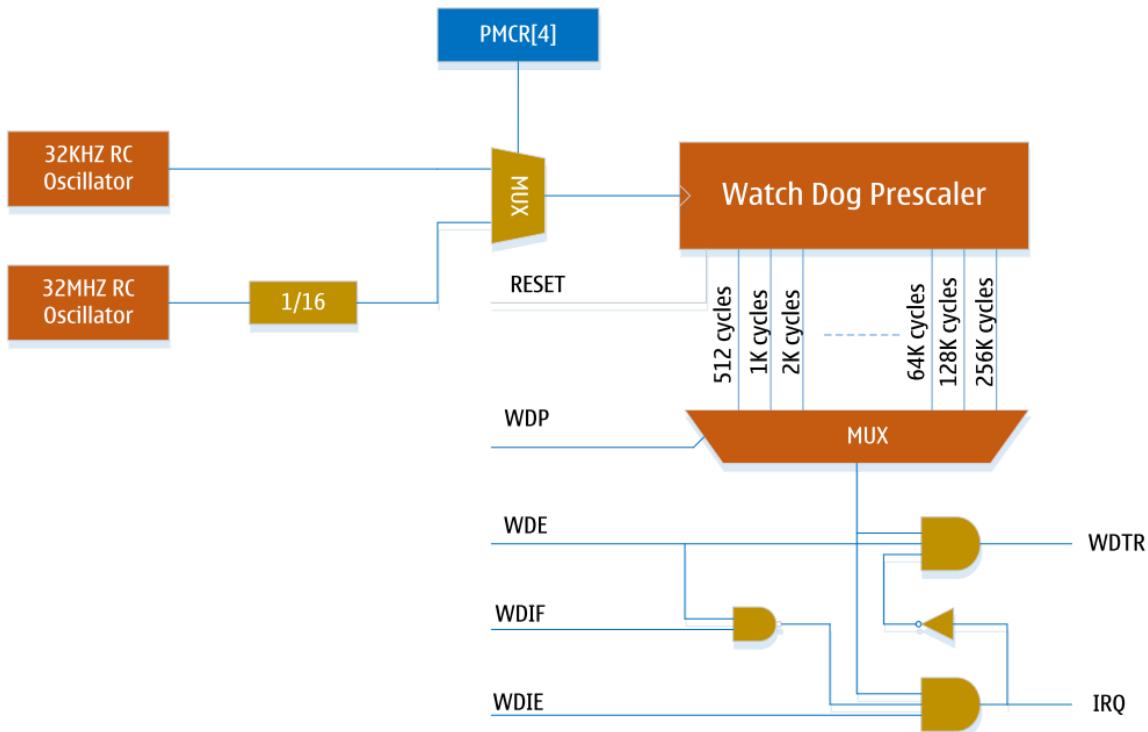


Рис.7.2. Структура сторожевого таймера

В режиме прерывания сигнал запроса прерывания генерируется при переполнении WDT. Это прерывание может использоваться как сигнал пробуждения для спящего режима и как общий системный таймер.

Например, вы можете использовать это прерывание, чтобы ограничить время выполнения операции, в завершении переполнения текущей задачи.

В режиме сброса системы WDT генерирует сигнал сброса системы сразу же после переполнения счетчика. Наиболее типичное использование - предотвращение сбоя или сбой системы.

Третий режим - это режим прерывания сброса, комбинация прерываний и сброс двух функций. Во-первых, система будет реагировать на функцию прерывания WDT, выйти из процедуры сброса прерывания WDT, немедленно переключиться в режим сброса. Эта функция может поддерживать сохранение некоторых из наиболее важных параметров информации перед сбросом. Чтобы предотвратить непреднамеренное отключение WDT, выключение WDT должно выполняться с четко определенным временем. Следующий код описывает, как отключить сторожевой таймер. В следующем примере предполагается, что прерывание отключено, так что весь рабочий поток не будет прерван.

Пример кода для включения и выключения сторожевого таймера:

### Код Ассемблера

WDT\_OFF:

<code>CLI</code>	<i>; Отключить глобальное прерывание</i>
<code>WDR</code>	<i>; Сброс Сторожевого таймера</i>
<code>IN r16, MCUSR</code>	<i>; Очистить WDRF в MCUSR</i>
<code>ANDI r16, ~(1 &lt;&lt; WDRF)</code>	
<code>OUT MCUSR, r16</code>	<i>; Запись «1» в WDCE и WDE</i>
	<i>; Сохранить настройку старого предделителя, чтобы</i>
	<i>предотвратить непреднамеренный тайм-аут</i>
<code>LDS r16, WDTCSR</code>	

```

ORI r16, (1 << WDCE) | (1 << WDE)
STS WDTCSR, r16
LDI r16, (0 << WDE) ; Выключить WDT
STS WDTCSR, r16
SEI ; Включить глобальное прерывание
RET

```

## С Код

```

void WDT_OFF(void)
{
    __disable_interrupt();
    __watchdog_reset();
    /* Очистить WDRF в MCUSR */
    MCUSR &= ~(1 << WDRF);
    /* Запись «1» в WDCE и WDE */
    /* Сохранить настройку старого предделителя, чтобы
     * предотвратить непреднамеренный тайм-аут */
    WDTCSR |= (1 << WDCE) | (1 << WDE);
    /* Выключить WDT */
    WDTCSR = 0x00;
    __enable_interrupt();
}

```

**Примечание:** Если WDT включен неожиданно, например, при запуске программы, чип будет сброшен, но WDT все еще включен.

Если WDT не обрабатывается в коде пользователя, это приведет к циклическому сбросу. Чтобы этого избежать, рекомендуется, чтобы программное обеспечение пользователя очищало контрольные биты сторожевого сброса (WDRF) и WDE в процедуре инициализации.

Следующий код описывает, как изменить значение времени тайм-аута сторожевого таймера.

## Код Ассемблера

WDT\_TOV\_Change:

```

CLI ; Отключить глобальное прерывание
WDR ; Сброс Сторожевого таймера
LDS r16, WDTCSR ; Запустите рассчитанную последовательность
ORI r16, (1 << WDCE) | (1 << WDE)
STS WDTCSR, r16 ; - Получил для циклов, чтобы установить здесь новое
                  ; значение -
                  ; Установить новое значение тайм-аута = 64k циклов
LDI r16, (1 << WDE) | (1 << WDP2) | (1 << WDP0)
STS WDTCSR, r16 ; - Закончена установка нового значения, используется 2
                  ; цикла -
SEI ; Включить глобальное прерывание
RET

```

С Код

```
void WDT_TOV_Change(void)
{
    __disable_interrupt();
    __watchdog_reset();
    /* Запустите рассчитанную последовательность */
    WDTCSR |= (1 << WDCE) | (1 << WDE);
    /* Установить новое значение тайм-аута = 64к циклов */
    WDTCSR |= (1 << WDE) | (1 << WDP2) | (1 << WDP0);
    __enable_interrupt();
}
```

**Примечание:** Перед изменением бит конфигурации WDP рекомендуется сбросить сторожевой таймер. Поскольку изменение бит WDP при относительно небольшом периоде ожидания, может привести к сбросу тайм-аута сторожевого таймера.

## 7.4. Регистры Описания

### VDTCR - Регистр управления нижним пределом напряжения (LVD)

*PMX2 - Регистр мультиплексора функции ввода-вывода*

## *MCUSR - Регистр состояния MCU*

**Примечание:** Для более точного и эффективного использования информации об обновлении сброса рекомендуется, прочитать о инициализации сброса флага и как очистить его.

## **WDTCSR - Регистр состояния контроля Watchdog**

## Watchdog - Список выбора Предделения

WDP3	WDP2	WDP1	WDP0	Период переполнения таймера сторожевого таймера	32KHZ Такты	2MHz Такты
0	0	0	0	2К циклы	64ms	1ms
0	0	0	1	4К циклы	128ms	2ms
0	0	1	0	8К циклы	256ms	4ms
0	0	1	1	16К циклы	512ms	8ms
0	1	0	0	32К циклы	Is	16ms
0	1	0	1	64К циклы	2s	32ms
0	1	1	0	128К циклы	4s	64ms
0	1	1	1	256К циклы	8s	128ms
1	0	0	0	512К циклы	16s	256ms
1	0	0	1	1024К циклы	32s	512ms
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

Зарезервировано

## 8. Прерывания

### 8.1. Вектора Прерываний

- 28 источников прерываний
- Программируемый векторный начальный адрес

Ресурсы прерывания LGT8F88P / 168P / 328P в основном одинаковы, основное различие: вектор прерывания LGT8F88P для командного слова (16) и LGT8F168P / 328P Вектор прерывания - 2 слова инструкций. LGT8F88P Списки прерываний

#### 8.1.1 LGT8F88P Списки прерываний

Таблица прерываний

Номер	Адрес Вектора	Сигнал источника прерывания	Описание источника прерывания
1	0x0000	RESET	Внешний сброс, Сброс при включении, Сброс сторожевого таймера, Сброс отладки SWD, Сброс низкого напряжения
2	0x0001	INT0	Запрос внешнего прерывания 0
3	0x0002	INTI	Запрос внешнего прерывания 1
4	0x0003	PCI0	Прерывание (програмное) от уровня сигнала 0 контакта
5	0x0004	PCI1	Прерывание (програмное) от уровня сигнала 1 контакта
6	0x0005	PCI2	Прерывание (програмное) от уровня сигнала 2 контакта
7	0x0006	WDT	Прерывание переполнения сторожевого таймера
8	0x0007	TC2 COMPA	Прерывание Равенства совпадение А Таймер 2
9	0x0008	TC2 COMPB	Прерывание Равенства совпадение В Таймер 2
10	0x0009	TC2 OVF	Прерывание Переполнения Таймер 2

11	0x000a	TC1 CAPT	Прерывание захвата входного сигнала Таймера 1
12	0x000b	TC1 COMPA	Прерывание Равенства совпадение А Таймер 1
13	0x000c	TC1 COMPB	Прерывание Равенства совпадение В Таймер 1
14	0x000d	TC1 OVF	Прерывание Переполнения Таймер 1
15	0x000e	TC0 COMPA	Прерывание Равенства совпадение А Таймер 0
16	0x000f	TC0 COMPB	Прерывание Равенства совпадение В Таймер 0
17	0x0010	TC0 OVF	Прерывание Переполнения Таймер 0
18	0x0011	SPI STC	Прерывание Последовательное передачи SPI
19	0x0012	USART RXC	Прерывание USART получает конец
20	0x0013	USART UDRE	Прерывание Регистр данных USART пуст
21	0x0014	USART TXC	Прерывание USART отправляет
22	0x0015	ADC	Прерывание Конец преобразования АЦП
23	0x0016	EE_RDY	Прерывание EEPROM Готов
24	0x0017	ANA_COMP	Прерывание аналогового компаратора 0
25	0x0018	TWI	Прерывание двухпроводного последовательного интерфейса
26	0x0019	ANA_C0MP1	Прерывание аналогового компаратора 1
27	0x001a	-	
28	0x001b	PCI3	Прерывание (програмное) от уровня сигнала 3 контакта
29	0x001c	PCI4	Прерывание (програмное) от уровня сигнала 4 контакта
30	0x001d	TC3_JNT	Прерывание Таймер 3

### 8.1.2 LGT8F168P / LGT8F328P Списки прерываний

Таблица прерываний

Номер	Адрес Вектора	Сигнал источника прерывания	Описание источника прерывания
1	0x0000	RESET	Внешний сброс, Сброс при включении, Сброс сторожевого таймера, Сброс отладки SWD, Сброс низкого напряжения
2	0x0002	INT0	Запрос внешнего прерывания 0
3	0x0004	INTI	Запрос внешнего прерывания 1
4	0x0006	PCI0	Прерывание (програмное) от уровня сигнала 0 контакта
5	0x0008	PCI1	Прерывание (програмное) от уровня сигнала 1 контакта
6	0x000a	PCI2	Прерывание (програмное) от уровня сигнала 2 контакта
7	0x000c	WDT	Прерывание переполнения сторожевого таймера
8	0x000e	TC2 COMPA	Прерывание Равенства совпадение А Таймер 2
9	0x0010	TC2 COMPB	Прерывание Равенства совпадение В Таймер 2
10	0x0012	TC2 OVF	Прерывание Переполнения Таймер 2
11	0x0014	TC1 CAPT	Прерывание захвата входного сигнала Таймера 1
12	0x0016	TC1 COMPA	Прерывание Равенства совпадение А Таймер 1
13	0x0018	TC1 COMPB	Прерывание Равенства совпадение В Таймер 1
14	0x001a	TC1 OVF	Прерывание Переполнения Таймер 1
15	0x001c	TC0 COMPA	Прерывание Равенства совпадение А Таймер 0
16	0x001e	TC0 COMPB	Прерывание Равенства совпадение В Таймер 0
17	0x0020	TC0 OVF	Прерывание Переполнения Таймер 0
18	0x0022	SPI STC	Прерывание передача SPI Завершено
19	0x0024	USART RXC	Прерывание приема USART Завершено
20	0x0026	USART UDRE	Прерывание Регистр данных USART пуст

21	0x0028	USART TXC	Прерывание USART отправляет
22	0x002a	ADC	Прерывание Конец преобразования АЦП
23	0x002c	EE_RDY	Прерывание EEPROM Готов
24	0x002e	ANA_COMP	Прерывание аналогового компаратора 0
25	0x0030	TWI	Прерывание двухпроводного последовательного интерфейса
26	0x0032	ANA_CCOMP1	Прерывание аналогового компаратора 1
27	0x0034	-	
28	0x0036	PCI3	Прерывание (програмное) от уровня сигнала 3 контакта
29	0x0038	PCI4	Прерывание (програмное) от уровня сигнала 4 контакта
30	0x003a	TC3_JNT	Прерывание Таймер 3

Вектор сброса LGT8FX8P начинается с адреса 0x0000. За исключением вектора сброса, все другие векторные адреса могут быть перенаправлены на начальный адрес с 512 байтами с использованием IVSEL в регистре MCUCR и регистре IVBASE.

## 8.2. Обработка векторных прерываний

Следующий код только для LGT8F88P, например, для иллюстрации программирования векторов переадресации и прерывания, только для справочных целей:

### Код Ассемблера- LGT8F88P

Адрес	Команда	Комментарий
0x000	RJMP RESET	; Сбросить вектор
0x001	RJMP EXTJNT0	; Внешнее прерывание 0
0x002	RJMP EXTJNT1	; Внешнее прерывание 1
0x003	RJMP PCINT0	; Прерывание смены знака 0
0x004	RJMP PCINT1	; Прерывание смены знака 1
0x005	RJMP PCINT2	; Прерывание смены знака 2
0x006	RJMP WDT	; Прерывание от сторожевого таймера
0x007	RJMP TIM2_COMPA	; Сравнение совпадений A по таймеру 2 Групповое прерывание
0x008	RJMP TIM2_COMPB	; Сравнение совпадений B по таймеру 2 Групповое прерывание
0x009	RJMP TIM2_OVF	; Прерывание переполнения таймера 2
0x00a	RJMP TIM1_CAPT	; Прерывание захвата таймера 1
0x00b	RJMP TIM1_C0MPA	; Сравнение совпадений A по таймеру 1 Групповое прерывание
0x00c	RJMP TIM1_C0MPB	; Сравнение совпадений B по таймеру 1 Групповое прерывание
0x00d	RJMP TIM1_OVFR	; Прерывание переполнения таймера 1
0x00e	RJMP TIM0_COMPA	; Сравнение совпадений A по таймеру 0 Групповое прерывание
0x00f	RJMP TIM0_COMPB	; Сравнение совпадений B по таймеру 0 Групповое прерывание
0x010	RJMP TIM0_OVF	; Прерывание переполнения таймера 0
0x011	RJMP SPI_STC	; Прерывание передача SPI завершено
0x012	RJMP USART_RXC	; Прерывание приема USART завершено
0x013	RJMP USART_UDRE	; Прерывание Регистр данных USART пуст
0x014	RJMP USART_TXC	; Прерывание передача USART завершено
0x015	RJMP ADC	; Прерывание ADC Конец преобразования
0x016	RJMP EE_RDY	; Прерывание EEPROM Готов
0x017	RJMP ANA_COMP	; Прерывание от компаратора
0x018	RJMP TWI	; Прерывание контроллера TWI
0x019	NOP	
0x01a	NOP	
0x01b	RJMP PCI3	; Прерывание смены знака 3

(RESET:)LDI r16, high(RAMEND) ; Начало Основная программа

0x01c

0x01d OUT SPH, r16 ; Установить указатель стека на верхний адрес RAM  
0x01e LDI r16, low(RAMEND)  
0x01f OUT SPL, r16  
0x020 SEI ; Включить глобальные прерывания  
0x021 .....

## 8.2. Регистры Описания

*IVBASE - Регистр базовых адресов вектора прерываний*

IVBASE - регистр адреса базового регистра прерываний		
IVBASE: 0x75		SKUtt: 0x00
IVBASE		IVBASE[7:0]
R/W		R/W
Определение бит		
[7:0]	IVBASE	Если IVSEL равно 1, вектор прерывания (кроме вектора сброса) будет переназначен на 512-байтной странице с базовым адресом IVBASE. Базовый адрес отображаемого вектора прерывания: (IVBASE << 8) + Соответствующий векторный адрес в таблице 1

## 8.3. Внешние прерывания

- 2 внешних источника прерывания
- Конфигурируемые прерывания с уровнем или по фронту
- Источники пробуждения, доступные в виде режима сна.

### 8.3.1. Описание

Внешние прерывания запускаются с помощью контактов INT0 и INT1. Пока включено внешнее прерывание, прерывание может быть запущено, даже если эти два вывода сконфигурированы как выходы. Это можно использовать для генерации программных прерываний. Внешние прерывания могут быть вызваны нарастающим фронтом, спадом или низким уровнем, который настраивается внешним регистром управления прерываниями EICRA.

Когда внешнее прерывание включено и сконфигурировано для чувствительных к уровню (только для контактов INT0 и INT1), прерывания генерируются до тех пор, пока контакт будет низким. Прерывания нарастающего фронта или падения фронтов на контактах INT0 и INT1 запускают нормальную работу тактов ввода-вывода, в то время как прерывания низкого уровня на выводах INT0 и INT1 обнаруживаются асинхронно.

В дополнение к режиму ожидания, такты IO в других режимах ожидания перестают работать. Поэтому оба внешних прерывания могут использоваться как источники пробуждения в других режимах ожидания, кроме режима ожидания.

Если прерывание с использованием уровня используется как источник пробуждения в режиме энергосбережения, измененный уровень должен удерживаться на определенное время, чтобы разбудить MCU, чтобы уменьшить

чувствительность микроконтроллера к шуму. Требуемый уровень должен поддерживаться достаточно долго, чтобы MCU завершил процесс пробуждения и затем инициировал прерывание уровня. Регистр регистрации

### 8.3.2. Регистры Описания

#### *Список регистров*

Регистр	Адрес	Значение по умолчанию	Описание
EICRA	0x69	0x00	Регистр управления внешним прерыванием A
EIMSK	0x3D	0x00	Регистр маски внешнего прерывания
EIFR	0x3C	0x00	Регистр флага внешнего прерывания

#### *EICRA - Регистр управления внешним прерыванием A*

EICRA - Регистр управления внешним прерыванием A								
Адрес: 0x69		Значение по умолчанию: 0x00						
Бит	7	6	5	4	3	2	1	0
Имя	-	-	-	-	ISC11	ISC10	ISC01	ISC00
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Бит	Имя	Описание						
7:4	-	Зарезервировано						
3	ISC11	Бит управления прерыванием INT1 бит высокий.						
2	ISC10	Бит управления запуском прерывания INT1 установлен на низком уровне. Когда глобальное прерывание установлено и соответствующий бит управления маской прерывания в регистре GICR установлен, внешнее прерывание 1 активируется выводом INT1. Метод запуска прерывания описан в таблице. Сначала MCU измеряет уровень на выводе INT1 до обнаружения фронта. Если выбран режим уровня или режим изменения уровня, импульс длительностью более 1 системного тактового сигнала вызовет прерывание. Короткий импульс не гарантирует запуск прерывания. Если выбран режим низкого уровня запуска, низкий уровень должен удерживаться до тех пор, пока текущая команда не будет завершена, чтобы вызвать прерывание.						
1	ISC01	Бит управления прерыванием INT0 бит.						
0	ISC00	Бит бит управления прерыванием INT0 бит. Когда глобальное прерывание установлено и соответствующий бит управления маской прерывания в регистре GICR установлен, внешнее прерывание 0 активируется выводом INT0. Метод запуска прерывания описан в таблице. Перед обнаружением фронта MCU сначала измеряет уровень на выводе INT0. Если выбрано триггерное включение или смена уровня, импульсы дольше, чем один такт системы, вызовут прерывание, в то время как короткие импульсы не вызовут прерывание. Если вы выбираете запуск режим низкого уровня, то низкий уровень должен поддерживаться до тех пор, пока не будет завершено текущее выполнение команды, чтобы вызвать прерывание.						

#### *Внешнее прерывание 1 Запуск управления режимом*

ISC1[1:0]	Описание
0	Внешнего вывода INT1 срабатывает в низкий уровень
1	Внешний контакт INT1 с восходящим или спадающим фронтом
2	Внешний контакт INT1 с спадающим фронтом
3	Внешний контакт INT1 с нарастающим фронтом

## *Внешнее прерывание 0 Запуск управления режимом*

<b>ISC0[1:0]</b>	<b>Описание</b>
0	Внешнего вывода INT0 срабатывает в низкий уровень
1	Внешний контакт INT0 с восходящим или спадающим фронтом
2	Внешний контакт INT0 с спадающим фронтом
3	Внешний контакт INT0 с нарастающим фронтом

## *EIMSK - Регистр маски Внешние прерывания*

## *EIFR - Регистр флага внешнего прерывания*

0	INTF0	Внешний флаг прерывания 0. INTF0 устанавливается, когда фронт вызывает прерывание с внешнего вывода 0. Бит INTF0 не устанавливается, когда прерывание внешнего вывода 0 срабатывает на низком уровне. Если прерывание внешнего вывода 0 прерывается, бит INT0EN устанавливается в 1 и установлен флаг глобального прерывания, тогда будет генерироваться прерывание внешнего вывода 0. INTF0 будет автоматически очищен при выполнении этой процедуры обслуживания прерывания или может быть очищен путем записи «1» в бит INTF0.
---	-------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 9.Общий программируемый порт

### 9.1. Описание

Все MCU, основанные на базовом семействе LGT8XM, имеют возможности чтения / изменения порта ввода-вывода. Это означает, что состояние конкретного порта может быть изменено индивидуально с использованием инструкций SBI и CBI, не затрагивая никаких других операций ввода-вывода. Кроме того, измените направление порта или контролируйте его сопротивление подтягивания.

Большинство входов / выходов LGT8FX8P имеют симметричные характеристики привода, которые приводят и поглощают большие токи. I / O имеет два уровня возможностей привода, и пользователи могут управлять возможностями привода каждого набора операций ввода-вывода. Возможности ввода-вывода могут напрямую управлять некоторыми светодиодами. Большинство входов / выходов LGT8FX8P поддерживают до 30 мА и могут использоваться непосредственно для управления светодиодами сегмента. Все VCC и GND I / O имеют свои собственные диоды защиты от электростатического разряда напрямую и рассчитаны на то, чтобы выдерживать импульсы ESD не менее 5000 В.

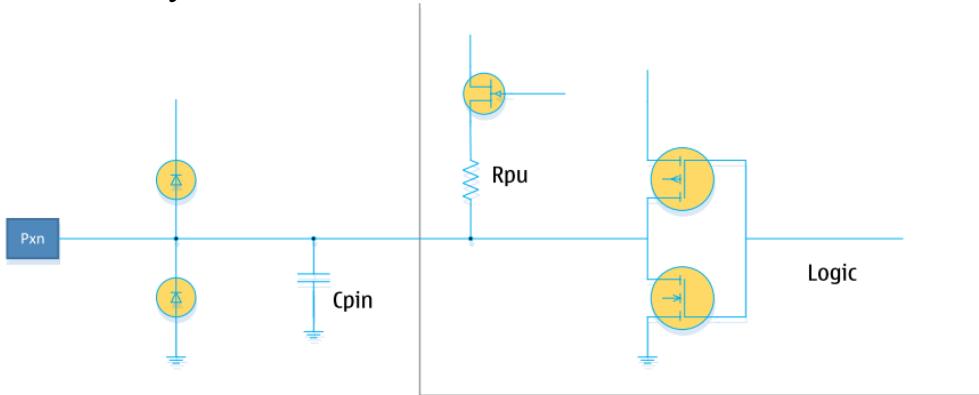


Рис.9.1. Схема эквивалентной схемы ввода / вывода

Все регистры в этой главе принимают единый режим описания, нижний регистр «x» обозначает номер букв порта, а нижний «n» обозначает номер бит в порту. Однако при использовании регистров портов в вашей программе вы должны использовать точное имя регистра. Например, PORTB3, что означает, что третий PORTB, здесь тот же, что и PORTxn.

Подробное определение регистров, связанных с I / O, см. В разделе «Описания регистров».

Каждому порту выделяются три регистра ввода-вывода: PORTx, PORTx (DDRx), регистр ввода данных порта (PINx). Входной регистр порта данных является регистром только для чтения. Регистр выходных данных и регистр

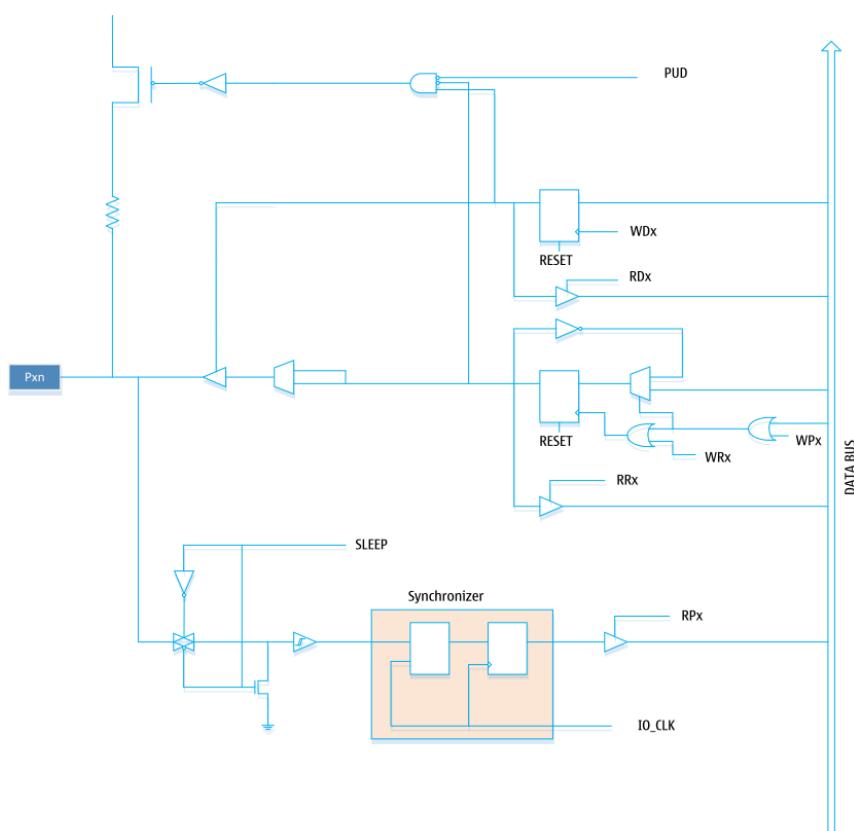
направления порта читаются и также могут быть перезаписаны.

Бит PUD в регистре MCUCR используется для управления подтягиванием всех входов / выходов. Когда бит PUD установлен в «1», подтягивание ввода-вывода отключается. Большинство входов / выходов, помимо их функций ввода / вывода общего назначения, также мультиплексируются в другие периферийные функции. Обратитесь к разделу мультиплексирования функций портов для конкретной функции мультиплексирования.

Следует отметить, что включение мультиплексирования некоторых портов не влияет на использование этих портов в качестве цифрового ввода-вывода. И некоторые функции мультиплексирования могут также нуждаться в управлении направлением ввода-вывода порта через регистры ввода-вывода. Конкретные настройки будут введены в документации каждого модуля мультиплексирования.

## 9.2. Порт ввода / вывода общего назначения

В качестве порта ввода-вывода общего назначения порт представляет собой двунаправленный порт ввода-вывода, внутренний программируемый подтягивание. Рис. показана эквивалентная принципиальная схема порта ввода-вывода общего назначения.



PUD: Подтягивание ОТКЛ

SLEEP: Контроль сна

IO\_CLK: Синхроимпульсы вв/выв

WDx: Запись DDRx

RDX: Читать DDRx

WRx: Запись PORTx

RRx: Читать регистр PORTx

RPx: Читать Вывод PORTx

WPx: Запись регистр PINx

Рис.9.2. Эквивалентная принципиальная схема порта ввода-вывода общего назначения.

## 9.3. Конфигурация использования портов

Каждый порт управляется тремя битами регистров: DDxn, PORTxn и PINxn. Среди них DDxn используется, чтобы иметь возможность посещать регистр DDRx, PORTxn может посещать регистр PORTx, PINxn может посещать регистр PINx.

Биты регистра DDRxn используются для установки направления ввода / вывода порта. Если DDxn установлен в «1», порт Pxh настроен как выходной порт. Если DDxn установлен в 0, Pxh настроен как входной порт.

Если бит PORTxn установлен в «1», и этот порт настроен как входной порт, то резистор подтягивания на этом порту действителен. Если вы хотите отключить нагрузочный резистор порта, PORTxn должен быть записан как «0» или настроить этот порт в качестве выходного порта.

Состояние инициализации сброса порта - это входное состояние, резистор подтягивания недействителен.

PORTxn установлен в «1», тогда как этот порт сконфигурирован как выходной порт, внешний порт будет управляться высоким. Если для параметра PORTxn установлено значение «0», порт будет отключен.

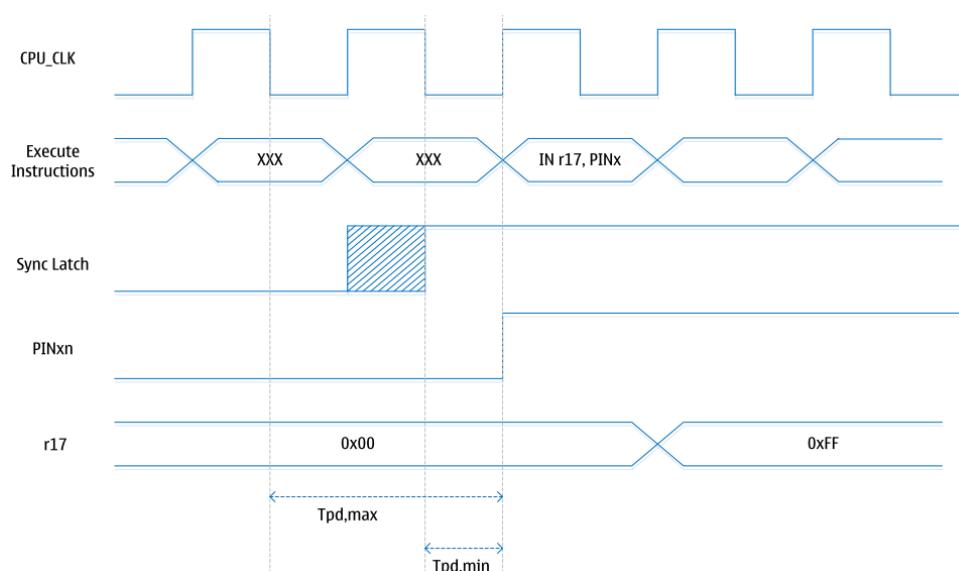


Рис.9.3. Тактировка чтение выводов.

### 9.3.1. Переключение ввода / вывода

Подтягивание или вывод портов произойдет, когда состояние ввода / вывода переключится между трехступенчатым ([DDxn, PORTxn] = 0b00) и высоким выходом ([DDxn, PORTxn] = 0b11). Низкий для середины состояния.

В общем, подтягивающие резисторы приемлемы, так как разница между высоким напряжением и подтягиванием в высокоомной среде не имеет значения. Если это не так, функция pull-up порта может быть отключена бит PUD в регистре MCUCR.

Аналогично, та же проблема возникает при переключении между входами с включенным подтягиванием и низким уровнем выходного сигнала. Пользователь должен использовать tristate ([DDxn, PORTxn] = 0b00) или высокий выход ([DDxn, PORTxn] = 0b11) в качестве промежуточного состояния.

Таблица конфигурации драйвера порта

DDxn	PORTxn	PUD	Состояние порта	Подтянуть	Описание функции
0	0	x	Запись	Запрет	Три состояния (High-Z)
0	1	0	Запись	Включить	Ввод + внутренний режим подтягивания
0	1	1	Запись	Запрет	Три состояния (High-Z)
1	0	x	Экспорт	Запрет	Низкий выход (двигатель)
1	1	x	Экспорт	Запрет	Высокая выходная мощность (двигатель)

### 9.3.2. Чтение значения порта

Независимо от установки бита DDxn направления порта текущее состояние порта может быть прочитано через бит регистра PINxn.

Чтобы избежать метастабильности, вызванной прямым чтением порта, бит регистра PINxn является результатом того, что порт проходит через синхронизатор.

Синхронизатор состоит из защелки и регистра, поэтому между значением PINxn и текущим портом существует небольшая задержка. Эта задержка обусловлена наличием синхронизатора с задержкой до полутора системных циклов. Мы предполагаем, что системный цикл начинается с первого падающего фронта системных тактов, защелка фиксирует данные, когда такты низкие, и данные проходят через защелку, когда такты высоки, как показано в заштрихованной области выше. Когда такта низкие, данные порта фиксируются и синхронизируются в регистре PINxn на переднем фронте следующего такта. Tpd, max и Tpd, min на приведенном выше рисунке представляют собой максимальные и минимальные задержки данных порта и делятся на 1,5 цикла и 0,5 цикла. Чтобы прочитать значение порта, установленное программным обеспечением, в поддержку ввода-вывода и чтения байтов ввода-вывода необходимо вставить команду «нет операции» (NOP). Времена, показаны на Рис. ниже:

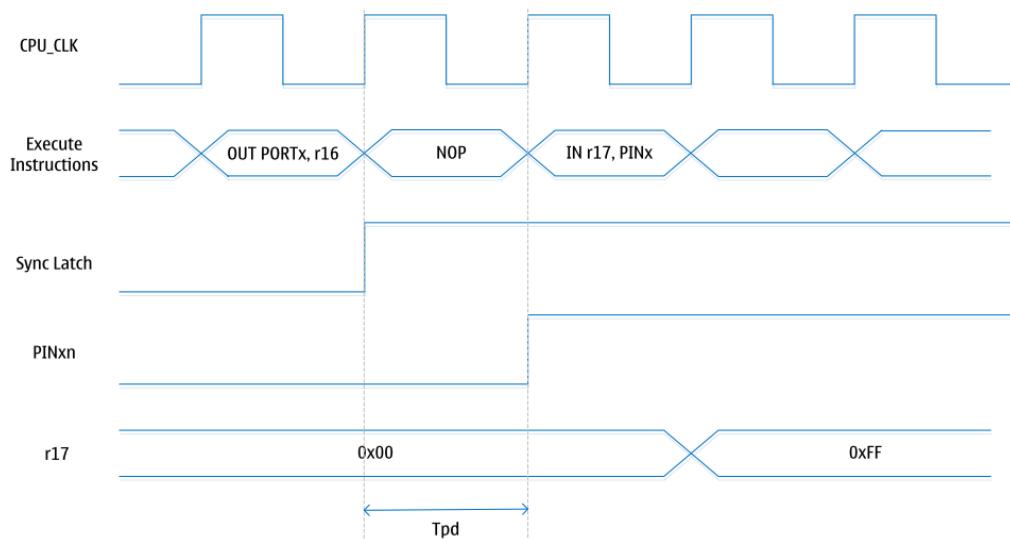


Рис. 9.4.

Следующий код показывает, как установить выводы порта B 0/1 с высоким, 2/3 низким, определить контакты 4 ~ 7 в качестве входных сигналов и разрешить подтягивающие резисторы для контактов 6,7. Значение вывода затем считывается обратно в рабочий регистр общего назначения, и инструкция NOP вставляется непосредственно на вывод вывода и ввод, как описано ранее.

## Сод Ассемблера

```
; Определить вытягивания и установить высокие выходы
; Определить направления для контактов порта
LDI r16, (1<<PB7)|(1<<PB6)|(1<<PB1)|1<<PB0)
LDI r17, (1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0)
OUT PORTB, r16
OUT DDRB, r17
; Вставить NOP для синхронизации
NOP
; Чтение контактов порта
IN r16, PINB
```

## Код C

```
unsigned char I;
/* Определить вытягивания и установить высокие выходы */
/* Определить направления для контактов порта */
PORTB = (1<<PB7)|(1<<PB6)|(1<<PB1)|(1<<PB0);
DDRB = (1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0);
/* Вставить NOP для синхронизации */
__no_operation();
/* Вставить NOP для синхронизации */
I = PINB;
```

### 9.3.3. Включение ввода и управление сном

Из схемы эквивалентной схемы ввода / вывода видно, что цифровой вход может быть зажат на землю под управлением сигнала SLEEP. Сигнал SLEEP управляется контроллером гибернации MCU и различными режимами гибернации. Это гарантирует, что после входа в спящий режим система не будет из-за плавания порта, вызванного утечкой.

Управление SLEEP порта заменяется функцией внешнего прерывания. Управление SLEEP все еще работает, если внешний запрос прерывания недействителен. Функция управления SLEEP также будет заменена некоторыми другими вторыми функциями.

Для получения дополнительной информации см. Следующее введение о второй функции порта.

### 9.3.4. Быстро перевернуть статус порта

Состояние порта установлено на выход IO, состояние порта может быть изменено через регистр PORTn.

Если вам нужно перевернуть статус вывода текущего порта, вам сначала необходимо сначала прочитать текущий статус порта PINx, а затем записать обратно в регистр PORTn, чтобы завершить флип.

LGT8FX8P обеспечивает еще один эффективный способ инвертировать состояние порта, перелистывая указанный статус порта, записывая 1 непосредственно в регистр PINx. Например, мы пишем PINB [3] равным 1, вы

можете достигнуть статуса статуса порта PB3. Этот метод очень полезен для приложений, которым необходимо генерировать выходные такты.

### 9.3.5. Цифровой / аналоговый мультиплексный порт

LGT8FX8P - часть порта для цифрового аналогового гибридного порта. За исключением внутреннего выхода ЦАП PD4, все остальные смешанные порты используются как аналоговые входы.

При использовании в качестве аналогового порта, программное обеспечение должно быть установлено в режим ввода в порт и отключить внутреннюю тягу, необходимую, чтобы избежать влияния на аналоговый доход.

DIDR0 ~ 2 регистрирует порт для закрытия функции микширования входного цифрового канала, для того, чтобы избежать ненужных потерь мощности, вызванные аналоговым входом в цифровую схему. DIDRx не отключает цифровой выход порта.

### 9.3.6. Двухтактный привод с высоким током

LGT8FX8P поддерживает более 6 высокоточных токовых приводных портов, двухтактный привод до 80 мА. Принимая во внимание максимальные ограничения максимальной пропускной способности чипа VCC, не рекомендуется открывать 6-канальный сильноточный привод.

Специально для пакетов QFP32 с одним портом питания не рекомендуется включать и одновременно управлять более 4 токовыми нагрузками.

Нормальный драйвер порта составляет 12 мА, программное обеспечение должно открыть порт высоковольтной функции привода через регистр HDR.

Таблица 9.1 Порты с высокой нагрузочной способностью

Порт HDR	QFP48	QFP32	HDR	Описание функции
PD5	PD5	PD5	HDR[0]	N/A
PD6	PD6	PD6	HDR[1]	N/A
PF1	PF1	PD1 PF1	HDR[2]	PD1 в корпусе QFP32 параллелен PD1 и PF1 в QFP48
PF2	PF2	PD2 PF2	HDR[3]	Порт HDR PD2 в корпусе QFP32 параллелен PD2 и PF2 в QFP48
PF4	PF4	PE4 PF4	HDR[4]	QFP32 внутренний PE4 эквивалентен QFP48 PF4 параллельно с PE4
PF5	PF5	PE5 PF5	HDR[5]	QFP32 внутренний PE5 эквивалентен QFP48 PF5 параллельно с PE5

### 9.3.7. Не активные порты

Если некоторые порты не используются, рекомендуется использовать их на фиксированном уровне. В любом случае плавающие выводы будут потреблять больше энергии и что приводит к неустойчивости системы при сильных помехах.

Самый простой способ дать порту фиксированный уровень - открыть

нагрузочный резистор порта. Следует отметить, что нагрузочный резистор отключен во время сброса при включении.

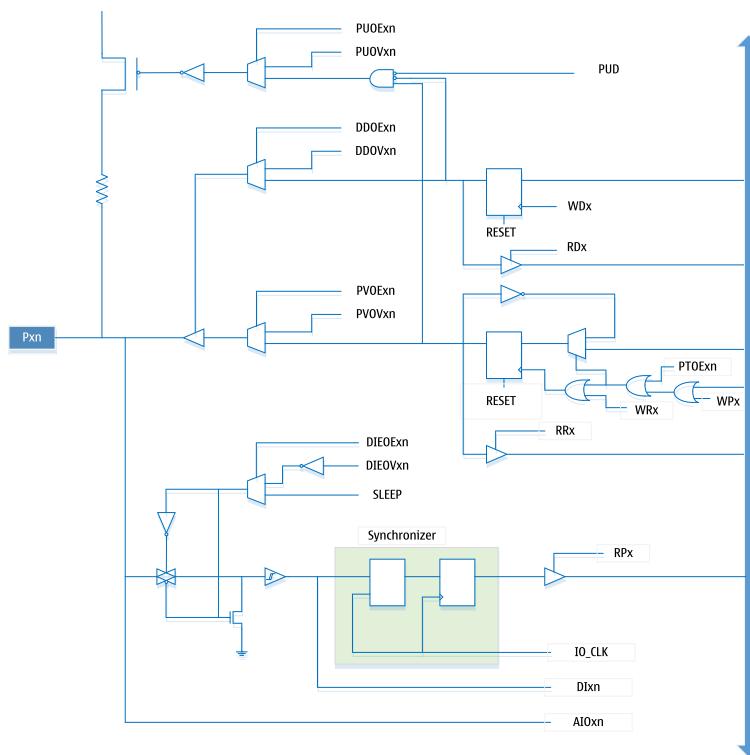
Вытягивание резистора также приведет к избыточной утечке. Поэтому рекомендуется использовать внешнее подтягивающее или выталкивающее резисторы.

Не рекомендуется напрямую подключать порт к источнику питания или заземлению, потому что если эти контакты сконфигурированы как выходы, это может вызвать очень большие токи, проходящие через порт, что приводит к разрушительным последствиям для чипа.

### 9.3.8. Функция мультиплексирования портов

Большинство портов имеют функции мультиплексирования. Следующая эквивалентная схема иллюстрирует функцию мультиплексирования портов для управления портом.

Эти функции мультиплексирования необязательно существуют с выводами порта.



PUOExn:	Pxn подтягивающая отмена устанавливают
PUOVxn:	Pxn подтягивающее значение отмены
DDOExn:	Управление данными Pxn OVERRIDE установить
DDOVxn:	Значение отмены руководства данных Pxn
PVOExn:	Отмена значения PORTA Pxn устанавливает
PVOVxn:	Значение отмены значения PORTA Pxn
DIEOExn:	Входное устройство Pxn - устанавливает отмене, установить
DIEOVxn:	Входное устройство Pxn - установить значению отмены
SLEEP:	Проверка SLEEP
PTOExn:	Отмена бистабильной схемы PORTA Pxn установить

PUD:	Подтягивающий отключает
WDx:	Запись DDRx
RDx:	Читать DDRx
RRx:	Читать регистр PORTx
WRx:	Запись PORTx
RPx:	Читать вывод PORTx
WPx:	Записать PINx
IO_CLK:	Синхроимпульсы ВВ/ВЫВ
DIxn:	Вход ВЫВОД n на PORTx
AI0xn:	АНАЛОГОВЫЙ ВВ/ВЫВ ВЫВОД n в PORTx

Рис. 9.5

Сигналы управления функцией повторного использования обычно описывают:

Сигнал	Полное имя	Функциональное описание
PUOE	Возможность мультиплексного включения	Этот бит равен «1», активирование подтягивания управляет PVOV, если этот бит «0», подтягивание включено совместно DDxn, PORTxn и PUD.
PUOV	Увеличить мультиплексное значение	Если PUOE равен «1», «1» включит нагрузочный резистор на контакте, иначе нагрузочный резистор будет отключен.
DDOE	Включить мультиплексирование портов	Второй бит равен «1», разрешение вывода контактов управляет DDOE, в противном случае оно управляет DDxn
DDOV	Значение мультиплексирования направления порта	Если DDOE равно «1», а второй бит равен «1», выходная функция вывода будет включена, в противном случае вывод вывода будет отключен.
PVOE	Мультиплексирование данных портов включено	Мультиплексирование данных портов включено
PVOV	Мультиплексирование данных портов	Описание функции PVOE Описание
PTOE	Мультиплексирование с инверсией портов	Второй бит равен «1», бит PORTxn будет инвертироваться
DIEOE	Включить мультиплексирование с цифровым входом	Если вторичный бит равен «1», разрешение цифрового входа порта будет контролироваться DIEOV, в противном случае будет выполняться контроль состояния MCU
DIEOV	Мультиплексирование с цифровым входом	Если DIEOE равно «1», функция цифрового входа порта будет управляться вторичным битом, независимо от рабочего состояния MCU
DI	Цифровой вход	Это цифровой входной сигнал, который вводится в альтернативный функциональный модуль. Как видно из нижних схем, таких как I / O, это значение после триггера Шмитта, но перед входным синхронизатором ввода / вывода. Этот сигнал подключается к периферийному модулю, и периферийный модуль будет синхронизироваться по мере необходимости
AIO	Аналоговый вход	Аналоговые входные / выходные сигналы, которые напрямую подключаются к PAD I / O, могут использоваться как аналоговые двунаправленные сигналы. Этот сигнал напрямую подключается к портам аналоговых модулей, таких как внутренние АЦП, компараторы и т. д.

В следующем подразделе кратко описывается функция мультиплексирования и соответствующие управляющие сигналы для каждого вывода.

### 9.3.9. Функция мультиплексирования портов В

Бит	Описание функции мультиплексирования
PB7	XTALI / TOSC2 - внешний основной контакт кварца XI PCINT7 - прерывание по смене уровня контакта 7
PB6	XTALO / TOSC1 - внешний вывод главного генератора XO PCINT6 - прерывание по смене уровня контакта 6

PB5	SCK - входной синхросигнал шины SPI PCINT5 - прерывание по смене уровня контакта 5
PB4	MISO - выход ведущего / ведомого выходной сигнал шины SPI PCINT4 - прерывание по смене уровня контакта 4
PB3	MOSI - вход ведущего / ведомого шины SPI OC2A - таймер / счетчик 2 сравнение, выходной сигнал А PCINT3 - прерывание по смене уровня контакта 3
PB2	SSN - вход выбора подчиненного устройства SPI OC1B - таймер / счетчик 1 сравнение, выходной сигнал В PCINT2 - прерывание по смене уровня контакта 2
PB1	OC1A - таймер / счетчик 1 сравнение, выходной сигнал А PCINT1 - прерывание по смене уровня контакта 1
PB0	ICP1 - вход для записи таймера / счетчика 1 CLKO - выходной сигнал системы PCINT0 - прерывание по смене уровня контакта 0

### *XTAL1 / TOSC2 / PCINT7 - Порт В Бит 7*

**XTAL1:** Внешний кристалл Pin XI. При использовании в качестве кварцевого резонатора тактового сигнала - этот вывод не может использоваться как I / O.

**TOSC2:** Таймер внешнего кристалла Pin 2. Когда внутренний RC сконфигурирован как основной рабочий такт чипа, и включена функция асинхронного таймера (конфигурация регистра ASSR), этот вывод будет внешним кварцевым генератором таймера. Когда AS2 регистра ASSR установлен в «1», а EXCLK установлен на «0», включена функция асинхронного таймера / счетчика событий 2 с использованием внешнего кристалла, и PB7 будет отключен от внутреннего порта ввода-вывода, чтобы стать реверсом внутреннего усилителя генератора. Выходной контакт. В этом режиме внешний кристалл подключается к контакту.

**PCINT7:** Прерывание по смене уровня контакта 7. PB7 является внешним источником прерываний. Если PB7 используется для резонаторов, значения DDB7, PORTB7 и PINB7 не будут иметь никакого значения.

### *XTAL0/TOSC1/PCINT6 - Порт В Бит 6*

**XTAL0:** Внешний кристалл pin XO.

**TOSC1:** Таймер внешнего кристалла 1. Когда внутренний RC сконфигурирован как основной рабочий такт чипа, и включена функция асинхронного таймера (конфигурация регистра ASSR), этот вывод будет внешним кварцевым генератором таймера. Когда AS2 регистра ASSR установлен в 1, а EXCLK установлен на 0, активируется асинхронная функция часов счетчика времени / события 2 с использованием внешнего кристалла. PB6 будет подключен к внутреннему порту ввода / вывода, чтобы стать входным выводом внутреннего генератора. В этом режиме внешний кристалл подключается к штифту.

**PCINT6:** Прерывание по смене уровня контакта 6. PB6 является внешним источником прерываний. Если PB6 используется для резонаторов контактов, значения DDB6, PORTB6 и PINB6 не будут иметь никакого значения.

## *SCK / PCINT5- Порт В Бит 5*

**SCK:** Выходной сигнал главного контроллера SPI, ведомый вход тактов. Когда контроллер SPI настроен как ведомое устройство, этот вывод будет сконфигурирован как входной контакт и не будет управляться DDB5. Когда контроллер SPI настроен как ведущее устройство, направление этого вывода управляется DDB5. Когда этот контакт будет принудительно введен SPI, подтягивающий резистор можно будет контролировать через бит PORTB5.

**PCINT5:** Прерывание по смене уровня контакта 5. PB5 является внешним источником прерываний.

## *MISO / PCINT4- Порт В Бит 4*

**MISO:** Ввод основных данных управления SPI, вывод ведомых данных. Когда SPI сконфигурирован как ведущее устройство, этот вывод будет принудительно вводиться и не будет управляться DDB4. Когда SPI является ведомым устройством, направление данных этого вывода управляется DDB4. Когда этот вывод принудительно вводится контроллером SPI, его нагрузочный резистор все еще может управляться PROTB4.

**PCINT4:** Прерывание по смене уровня контакта. PB4 является внешним источником прерываний.

## *MOSI / OC2A / PCINT3- Порт В Бит 3*

**MOSI:** Вывод данных ведущего устройства контроллера SPI, ввод данных ведомого устройства. Когда SPI настроен как подчиненное устройство, этот вывод будет принудительно введен в качестве входа и не будет управляться DDB3. Когда контроллер SPI сконфигурирован как ведущее устройство, метод этого вывода управляется DDB3. Когда этот штифт принудительно вводится в управление SPI, его нагрузочный резистор можно контролировать через PORTB3.

**OC2A:** Таймера /события 2 сравнение, выходной сигнал А. PB3 можно использовать в качестве внешнего выходного сигнала **сравнения Timer / Counter2**. Вывод должен быть установлен на выход через DDB3. В то же время OC2A также является выходным выводом режима PWM таймера 2.

**PCINT3:** Прерывание по смене уровня контакта. PB3 является внешним источником прерываний.

## *SSN / OC1B / PCINT2- Порт В Бит 2*

**SSN:** SPI Slave Chip Выберите вход. Когда контроллер SPI настроен как подчиненное устройство, этот вывод будет принудительно введен в качестве входа и не будет управляться DDB2. В качестве подчиненного устройства для контроллера SPI допустимо приводить низкое значение в SSN. Когда контроллер SPI сконфигурирован как ведущее устройство, направление этого вывода управляется DDB2. После того, как этот штифт будет принудительно введен контроллером SPI, подтягивающий резистор можно будет контролировать через

## PORTB2.

**OC1B:** Таймера / счетчика 1 сравнение, выходной сигнал В. PB2 можно использовать в качестве таймера / счетчика 1 для сравнения внешнего выхода. Вывод должен быть установлен на выход через DDB2. В то же время OC1B также является выходным выводом режима PWM таймера 1.

**PCINT2:** Прерывание по смене уровня контакта. PB2 является внешним источником прерываний.

## *OC1A / PCINT1- Порт В Бит 1*

**OC1A:** Таймер А / Счетчик 1 сравнение, выходной сигнал А. PB1 может использоваться в качестве таймера / счетчика 1 для сравнения внешнего выхода. Вывод должен быть установлен на выход через DDB1. В то же время OC1A также является выходным выводом режима PWM таймера 1.

**PCINT1:** Прерывание по смене уровня контакта. PB1 является внешним источником прерываний.

## *ICP1 / CLKO / PCINT0- Порт В Бит 0*

**ICP1:** Захват входного вывода таймера / счетчика 1 CLKO: Выходной сигнал системы. Когда бит CLKOE в регистре CLKPR равен 1, этот вывод будет выведен на выход. Под контролем DDB0. Выходная частота - это тактовая частота операционной системы текущей системы.

**PCINT0:** Прерывание по смене уровня контакта. PB0 является внешним источником прерываний.

### 9.3.10. Функция мультиплексирования портов С

Бит	Описание функции мультиплексирования
PC7	ADC8 - входной канал ADC 8 APN2 - инвертированный вход DAP 2 PCINT15 - прерывание по смене уровня контакта 15
PC6	RESETN - вход внешнего сброса PCINT14 - прерывание по смене уровня контакта 14
PC5	ADC5 - входной канал АЦП 5 SCL - линия синхронизации TWI PCINT13 - прерывание по смене уровня контакта 13
PC4	ADC4 - входной канал 4 АЦП SDA - линия данных TWI PCINT12 - прерывание по смене уровня контакта 12
PC3	ADC3 - входной канал 3 АЦП APN1 - инвертированный вход DAP 1 PCINT11 - прерывание по смене уровня контакта 11
PC2	ADC2 - входной канал 2 АЦП PCINT10 - прерывание по смене уровня контакта 10
PC1	ADC1 - входной канал 1 АЦП PCINT9 - прерывание по смене уровня контакта 9

PC0	ADC0 - входной канал 0 АЦП PCINT8 - прерывание по смене уровня контакта 8
-----	------------------------------------------------------------------------------

### *ADC8 / APN2 / PCINT15- Порт С Бит 7*

**ADC8:** Внешний входной канал ADC 8

**APN2:** Инвертирование входного порта дифференциального усилителя 2

**PCINT15:** Прерывание по смене уровня контакта. PC7 является внешним источником прерываний.

### *RESETN / PCINT14- Порт С Бит 6*

**RESETN:** Вход для внешнего сброса. После включения питания этот контакт по умолчанию выполняет функцию внешнего сброса. Функция внешнего сброса может быть отключена регистром IOCR. После выключения функции внешнего сброса этот вывод может использоваться как универсальный ввод-вывод. Однако следует отметить, что при включении питания и других сбросах этот вывод по умолчанию соответствует входу сброса, поэтому, если пользователю необходимо использовать функцию ввода-вывода общего назначения этого вывода, внешняя цепь не может повлиять на включение и сброс микросхемы. В этом процессе рекомендуется настроить этот вывод как ввод-вывод для функции вывода и добавить внешний резистор под нагрузкой.

**PCINT14:** Прерывание по смене уровня контакта. После выключения функции внешнего сброса этого вывода, PC6 можно использовать в качестве внешнего источника прерывания.

### *SCL / ADC5 / PCINT13- Порт С бит 5*

**SCL:** Сигнал интерфейса интерфейса TWI. После того, как бит TWEN в регистре TWCR установлен в 1, интерфейс TWI включен, а PC5 управляет TWI, чтобы стать тактовым сигналом интерфейса TWI.

**ADC5:** Входной канал АЦП 5. Регистр DIDR используется для отключения цифровой функции цифро-аналогового мультиплексированного ввода-вывода, чтобы избежать влияния цифровой части на аналоговую схему. Подробнее см. В соответствующем разделе АЦП.

**PCINT13:** Прерывание по смене уровня контакта. PC5 является внешним источником прерываний.

### *SDA / ADC4 / PCINT12- Порт С Бит 4*

**SDA:** Сигнал данных интерфейса TWI. После того, как бит TWEN в регистре TWCR установлен, интерфейс TWI включен, а PC4 управляет TWI, чтобы стать сигналом данных интерфейса TWI.

**ADC4:** Входной канал 4 АЦП. Регистр DIDR используется для отключения цифровой функции цифро-аналогового мультиплексированного ввода-вывода, чтобы избежать влияния цифровой части на аналоговую схему. Подробнее см. В

соответствующем разделе АЦП.

**PCINT12:** Прерывание по смене уровня контакта. PC4 является внешним источником прерываний.

### *ADC3 / APN1 / PCINT11- Порт С Бит 3*

**ADC3:** Входной канал АЦП 3. Регистр DIDR используется для отключения цифровой функции цифро-аналогового мультиплексированного ввода-вывода, чтобы избежать влияния цифровой части на аналоговую схему. Подробнее см. В соответствующем разделе АЦП.

**APN1:** Инвертирование входного порта дифференциального усилителя 1

**PCINT11:** Прерывание по смене уровня контакта. PC3 является внешним источником прерываний.

### *ADC2 / APN0 / PCINT10- Порт С Бит 2*

**ADC2:** Вход канала АЦП 2. Регистр DIDR используется для отключения цифровой функции цифро-аналогового мультиплексированного ввода-вывода, чтобы избежать влияния цифровой части на аналоговую схему. Подробнее см. В соответствующем разделе АЦП.

**APN0:** Инвертирование входного порта дифференциального усилителя 0

**PCINT10:** Прерывание по смене уровня контакта. PC2 является внешним источником прерываний.

### *ADC1 / APP1 / PCINT9- Порт С Бит 1*

**ADC1:** Вход канала АЦП 1. Регистр DIDR используется для отключения цифровой функции цифро-аналогового мультиплексированного ввода-вывода, чтобы избежать влияния цифровой части на аналоговую схему. Подробнее см. В соответствующем разделе АЦП.

**APP1:** Инвертирование входного порта дифференциального усилителя 1

**PCINT9:** Прерывание по смене уровня контакта. PC1 является внешним источником прерываний.

### *ADC0 / APP0 / PCINT8- Порт С Бит 0*

**ADC0:** Вход канала АЦП 0. Регистр DIDR используется для отключения цифровой функции цифро-аналогового мультиплексированного ввода-вывода, чтобы избежать влияния цифровой части на аналоговую схему. Подробнее см. В соответствующем разделе АЦП.

**APP0:** Инвертирование входного порта дифференциального усилителя 0

**PCINT8:** Прерывание по смене уровня контакта. PC0 является внешним источником прерываний.

### 9.3.11. Функция мультиплексирования портов D

Бит	Описание функции мультиплексирования
PD7	ACXN - аналоговый компаратор 0/1 общий инвертирующий вход PCINT23 - прерывание по смене уровня контакта 23
PD6	AC0P - аналоговый компаратор 0 неинвертирующий вход (корпус QFP32) OC0A - таймер / счетчик 0 сравнение, выходной сигнал А OC3A - таймер / счетчик 3 сравнение, выходной сигнал А (корпус QFP32) PCINT22 - прерывание по смене уровня контакта 22
PD5	T1 - Таймер / Счетчик 1 Вход счетчика внешнего счета OC0B Таймер / Счетчик 0 сравнение, выходной сигнал В PCINT21 - прерывание по смене уровня контакта 21
PD4	XCK - внешний вход / выход внешнего входа USART DAO - 8-разрядный аналоговый выход ЦАП T0 - таймер / счетчик 0 внешних счетных часов PCINT20 - прерывание по смене уровня контакта 20
PD3	INT1 - вход внешнего прерывания 1 OC2B - таймер / счетчик 2 сравнение, выходной сигнал В PCINT19 - прерывание по смене уровня контакта 19
PD2	INT0 - вход внешнего прерывания 0 AC0O - выход компаратора 0 OC3B - таймер / счетчик 3 сравнение, выходной сигнал В (корпус QFP32) PCINT18 - прерывание по смене уровня контакта 18
PD1	TXD - выход данных USART OC3A - таймер / счетчик 3 сравнение, выходной сигнал А (корпус QFP32) PCINT17 - прерывание по смене уровня контакта 17
PD0	RXD - ввод данных USART PCINT16 - прерывание по смене уровня контакта 16

#### ACXN / OC2B / PCINT23- Порт D Бит 7

**ACXN:** Аналоговый компаратор 0/1 Общий инвертирующий вход

**OC2B:** Таймер / счетчик 2 сравнение, выходной сигнал В. PD7 можно использовать в качестве внешнего выходного сигнала Timer / Counter2 Compare Matching. В этом случае вывод должен быть установлен как выход через DDD7. В то же время OC2B также является выходным контактом режима PWM таймера 2.

**PCINT23:** Прерывание по смене уровня контакта. PD7 является внешним источником прерываний.

#### AC0P / OC0A / PCINT22- Порт D Бит 6

**AC0P:** Аналоговый компаратор 0 Положительный вход.

**OC0A:** Таймер / счетчик 2 сравнение, выходной сигнал А. PD6 можно использовать в качестве внешнего выходного сигнала сравнения таймера / счетчика 0. Вывод должен быть настроен на выход через DDD6. В то же время OC0A также является выходным штырем режима PWM таймера 0.

**PCINT22:** Прерывание по смене уровня контакта. PD6 является внешним источником прерываний.

### *T1 / OC0B / PCINT21- Порт D Бит 5*

**T1:** Счетчик таймера / события 1 внешний счетчик часов синхронизации

**OC0B:** Таймер / счетчик 0 сравнение, выходной сигнал B. PD5 может использоваться как внешний выходной сигнал сравнения по таймеру / счетчику 0. Штырь должен быть установлен на выход через DDD5. В то же время OC0B также является выходным выводом режима PWM таймера 0.

**PCINT21:** Прерывание по смене уровня контакта. PD5 является внешним источником прерываний.

### *XCK / T0 / DAO / PCINT20- Порт D Бит 4*

**XCK:** Синхронный режим Внешний сигнал синхронизации USART

**T0:** Внешний вход счетчика времени таймера / счетчика 0

**DAO:** Аналоговый выход 8-разрядный ЦАП

**PCINT20:** Прерывание по смене уровня контакта. PD4 является внешним источником прерываний.

### *INT1 / OC2B / PCINT19- Порт D Бит 3*

**INT1:** Вход внешнего прерывания 1

**OC2B:** Таймер / счетчик 2 сравнение, выходной сигнал B. PD3 может использоваться как внешний выход сравнения по таймеру / счетчику 2. Вывод должен быть установлен на выход через DDD3. В то же время OC2B также является выходным контактом режима PWM для таймера 2.

**PCINT19:** Прерывание по смене уровня контакта. PD3 является внешним источником прерываний.

### *INT0 / OC3B / AC0O / PCINT18- Порт D Бит 2*

**INT0:** Вход внешнего прерывания 0

**OC3B:** Таймер / счетчик 3 сравнение, выходной сигнал B. Только когда QFP32 корпус, PD2 и QFP48 / PF2 сливаются в один IO, поэтому функция OC3B на PF2 также будет выводиться из PD2

**AC0O:** Аналоговый компаратор 0 Результат сравнения выводится напрямую. Контролируется регистром AC0FR

**PCINT18:** Прерывание по смене уровня контакта. PD2 является внешним источником прерываний.

### *TXD / OC3A / PCINT17- Порт D Бит 1*

**TXD:** Передачи данные (вывод данных USART). После того, как передатчик USART включен, PD1 будет принудительно выводиться без Управление DDD1

**OC3A:** Таймер / счетчик 3 сравнение, выходной сигнал A. Только в пакете QFP32 PD1 и QFP48 / PF1 сливаются в один IO, поэтому функция OC3A на PF1 также будет выводиться из PD1

**PCINT17:** Прерывание по смене уровня контакта. PD1 является внешним источником прерываний.

## RXD / PCINT16- Порт D Бит 0

**RXD:** Передача данных (ввод данных USART). После включения приемника USART PD0 будет принудительно вводиться без DDD0. Подтягивающий резистор все еще может управляться через бит PORTD0, когда штифт принудительно подается на вход USART

**PCINT16:** Прерывание по смене уровня контакта. PD0 является внешним источником прерываний.

### 9.3.12. Функция мультиплексирования портов E

Бит	Описание функции мультиплексирования
PE7	ADC11 - входной канал ADC 11 PCINT31 - прерывание по смене уровня контакта 31
PE6	<b>AVREF</b> - внешний источник опорного напряжения АЦП (корпус QFP32) ADC10 - входной канал ADC 10 PCINT30 - прерывание по смене уровня контакта 30
PE5	CLKO - системный тактовый выход AC1O - выход аналогового компаратора 1 PCINT29 - прерывание по смене уровня контакта 29
PE4	OC0A - таймер / счетчик 0 сравнение, выходной сигнал A PCINT28 - прерывание по смене уровня контакта 28
PE3	ADC3 - входной канал 3 АЦП AC1N - аналоговый компаратор 1 инверсный вход PCINT27 - прерывание по смене уровня контакта 27
PE2	SWD - данные подключение отладчика SWD PCINT26 - прерывание по смене уровня контакта 26
PE1	ADC6 - входной канал 6 АЦП ACXP – аналоговый компаратор 0/1, неинверсный вход PCINT25 - прерывание по смене уровня контакта 25
PE0	SWC - входной сигнал тактов отладчика SWD APN4 - инверсный вход 4 дифференциального усилителя PCINT24 - прерывание по смене уровня контакта 24

### ADC11 / PCINT31- Порт E Бит 7

**ADC11:** Вход канала АЦП 11

**PCINT31:** Прерывание по смене уровня контакта. PE7 является внешним источником прерываний.

### AVREF / ADC10 / PCINT30- Порт E Бит 6

**AVREF:** АЦП внешних опорный сигнал источника питание, при использовании в качестве аналогового функции, соответствующий цифровой ввод / вывод предусмотрен в качестве входных данных, а также отключить нагрузочный резистор, чтобы избежать помех от аналоговой схемы цифровой схемы

**ADC10:** Вход канала АЦП 10

**PCINT30:** Прерывание по смене уровня контакта. PE6 является внешним источником прерываний.

#### *CLKO / AC10 / PCINT29- Порт E Бит 5*

**CLKO:** Эта функция аналогична функции CLKO PB0. Может использоваться в качестве запасного вывода для PB0 / CLKO

**AC10:** Выход аналогового компаратора 1

**PCINT29:** Прерывание по смене уровня контакта. PE5 является внешним источником прерываний.

#### *OC0A / PCINT28- Порт E Бит 4*

**OC0A:** Таймер А / Счетчик 0 сравнение, выходной сигнал А. PE4 может использоваться в качестве таймера / счетчика 0, сравниваемого с внешним выходом. Вывод должен быть установлен на выход через DDE4. В то же время OC0A также является выходным выводом режима PWM режима Timer0.

**PCINT28:** Прерывание по смене уровня контакта. PE4 является внешним источником прерываний.

#### *ADC7 / AC1N / PCINT27- Порт E Бит 3*

**ADC7:** Вход канала 7 АЦП. Регистр DIDR используется для отключения цифровой функции цифро-аналогового мультиплексированного ввода-вывода, чтобы избежать влияния цифровой части на аналоговую схему. Для получения дополнительной информации см. Раздел АЦП

**AC1N:** Аналоговый компаратор 1, инверсный вход

**PCINT27:** Прерывание по смене уровня контакта. PE3 является внешним источником прерываний.

#### *SWD / PCINT26- Порт E Бит 2*

**SWD:** Подключение отладчика SWD. PE2 по умолчанию выполняет функцию SWD. Пользователь может отключить функцию отладчика SWD, установив бит SWDD регистра MCUSR. Функция отладки не будет доступна после закрытия SWD.

**PCINT26:** Прерывание по смене уровня контакта. PE2 является внешним источником прерываний.

#### *ADC6 / ACXP / PCINT25- Порт E Бит 1*

**ADC6:** Вход канала 6 АЦП. Регистр DIDR используется для отключения цифровой функции цифро-аналогового мультиплексированного ввода-вывода, чтобы избежать влияния цифровой части на аналоговую схему. Подробную информацию см. В разделе, посвященном АЦП.

**ACXP:** Аналоговый компаратор 0/1 Общий неинвертируемый вход

**PCINT25:** Прерывание по смене уровня контакта. PE1 является внешним источником прерываний.

#### *SWC / APN4 / PCINT24- Порт E Бит 0*

**SWC:** линия таймера отладки SWD. PE0 по умолчанию выполняет функцию SWC. Пользователь может отключить функцию отладчика SWD, установив бит SWDD регистра MCUSR. После отключения SWD функция отладки не сможет использовать

**APN4:** дифференциальный усилитель, инвертирующий вход 4

**PCINT24:** Прерывание по смене уровня контакта. PE0 является внешним источником прерываний.

#### 9.3.13. Функция мультиплексирования портов F

Бит	Описание функции мультиплексирования
PF7	OC2B - таймер / счетчик 2 сравнение, выходной сигнал В PCINT39 - прерывание по смене уровня контакта 39
PF6	T3 - вход таймера / счетчика 3 внешних тактов OC2A - таймер / счетчик 2 сравнение, выходной сигнал А PCINT38 - прерывание по смене уровня контакта 38
PF5	OC1A - таймер / счетчик 1 сравнение, выходной сигнал А PCINT37 - прерывание по смене уровня контакта 37
PF4	OC1B - таймер / счетчик 1 сравнение, выходной сигнал В ICP3 - вход для внешнего захвата таймера / счетчика 3 PCINT36 - прерывание по смене уровня контакта 36
PF3	OC0B - таймер / счетчик 0 сравнение, выходной сигнал В PCINT35 - прерывание по смене уровня контакта 35
PF2	OC3B - таймер / счетчик 3 сравнение, выходной сигнал В PCINT34 - прерывание по смене уровня контакта 34
PF1	OC3A - таймер / счетчик 3 сравнение, выходной сигнал А PCINT33 - прерывание по смене уровня контакта 33
PF0	ADC9 - входной канал ADC 9 APN3 - инверсный вход 3 дифференциального усилителя PCINT32 - прерывание по смене уровня контакта 32

#### *OC2B / PCINT39 - Порт F Бит 7*

**OC2B:** Таймер / счетчик 2 сравнение, выходной сигнал В. Выбор выхода контролируется регистром PMX1

**PCINT39:** Прерывание по смене уровня контакта. PF7 является внешним источником прерываний.

#### *OC2A / T3 / PCINT38 - Порт F Бит 6*

**OC2A:** таймер / счетчик 2 сравнение, выходной сигнал А. Выбор выхода контролируется регистром PMX1

**T3:** Таймер / Счетчик 3 Вход внешних Тактов

**PCINT38:** Прерывание по смене уровня контакта. PF6 является внешним источником прерываний.

#### *OC1A / PCINT37 - Порт F Бит 5*

**OC1A:** Таймер / Счетчик 1 сравнение, выходной сигнал A. Выбор выхода контролируется регистром PMX0

**PCINT37:** Прерывание по смене уровня контакта. PF5 является внешним источником прерываний.

#### *ICP3 / OC1B / PCINT36 - Порт F Бит 4*

**OC1B:** Таймера / счетчика 1 сравнение, выходной сигнал B. Выбор выхода контролируется регистром PMX0

**ICP3:** Таймер / счетчик 3 Вход для внешнего захвата

**PCINT36:** Прерывание по смене уровня контакта. PF4 является внешним источником прерываний.

#### *OC0B / OC3C / PCINT35- Порт F Бит 3*

**OC0B:** Таймера / счетчика 0 сравнение, выходной сигнал B. Выбор выхода контролируется регистром PMX0

**OC3C:** Таймера / счетчика 3 сравнение, выходной сигнал C

**PCINT35:** Прерывание по смене уровня контакта. PF3 является внешним источником прерываний.

#### *OC3B / PCINT34- Порт F Бит 2*

**OC3B:** Таймера / счетчика 3 сравнение, выходной сигнал B

**PCINT34:** Прерывание по смене уровня контакта. PF2 является внешним источником прерываний.

#### *OC3A / PCINT33- Порт F Бит 1*

**OC3A:** Таймером / счетчиком 3 сравнение, выходной сигнал B. Выбор выхода контролируется регистром PMX1

**PCINT33:** Прерывание по смене уровня контакта. PF1 является внешним источником прерываний.

#### *ADC9 / APN3 / PCINT32- Порт F Бит 0*

**ADC9:** Входной канал внешнего входа АЦП 9

**APN3:** обратный вход дифференциального усилителя 3

**PCINT32:** Прерывание по смене уровня контакта. PF0 является внешним источником прерываний.

### 9.3.14. Регистры Описания

### *PORTB - Регистр выходных данных порта B*

## *DDRB - Регистр направления В порта*

### *PINB - Регистр входных данных порта B*

*PORTC - Регистр выходных данных порта С*

## *DDRC - Регистр направления С порта*

DDRC - Регистр направления С порта								
DDRC: 0x07(0x27)		Значение по умолчанию: 0x00						
Биты	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0

### *PINC - Регистр входных данных порта С*

### *PORTD - Регистр выходных данных порта D*

### *DDRD - Регистр направления D порта*

### *PIND - Регистр входных данных порта D*

## *PORTE - Регистр выходных данных порта E*

PORTE - Регистр выходных данных порта E	
PORTE: 0x08(0x28)	Значение по умолчанию: 0x00

<b>Биты</b>	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Определение бит</b>								
[7:0]	PORTE	Регистр выхода порта E						

### *DDRE - Регистр направления E порта*

DDRE - Регистр направления E порта									
<b>DDRE: 0x07(0x27)</b>		Значение по умолчанию: 0x00							
<b>Биты</b>	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0	
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
<b>Определение бит</b>									
[7:0]	DDRE	Биты управления направлением порта E: 1 = выход, 0 = вход							

### *PINE - Регистр входных данных порта E*

PINE - Регистр входных данных порта E									
<b>PINE: 0x06(0x26)</b>		Значение по умолчанию: 0x00							
<b>Биты</b>	PINE7	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0	
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
<b>Определение бит</b>									
[7:0]	PINE	Регистр состояния порта E. Считывается PINE напрямую, чтобы получить текущее состояние порта; запись в бит PINEn «1» переворачивает состояние вывода PORTEn							

### *PORTF - Регистр выходных данных порта F*

PORTF - Регистр выходных данных порта F									
<b>PORTF: 0x0B(0x2B)</b>		Значение по умолчанию: 0x00							
<b>Биты</b>	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
<b>Определение бит</b>									
[7:0]	PORTF	Регистр состояния порта F: режим - ввода, запись «1» в бит откроет внутреннее подтягивание; режим - вывода, запись «1» в бит установит высокую выходную мощность.							

### *DDRF - Регистр направления F порта*

DDRD - Регистр направления D порта									
<b>DDRF: 0x0A(0x2A)</b>		Значение по умолчанию: 0x00							
<b>Биты</b>	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
<b>Определение бит</b>									
[7:0]	DDRF	Биты управления направлением порта F: 1 = выход, 0 = вход							

### *PIND - Регистр входных данных порта D*

PINF - Регистр входных данных порта F									
<b>PINF: 0x09(0x29)</b>		Значение по умолчанию: 0x00							
<b>Биты</b>	PINF7	PINF6	PINF5	PINF4	PINF3	PINF2	PINF1	PINF0	

## *HDR - Регистр управления портами*

## *PMX0 - Регистр 0 управления портовым мультиплексором*

		0 - выход TXD на PD1
0	RXD5	Управление дополнительным входом Serial RXD 1 = вход RXD от PD5 0 = вход RXD от PD0

## *PMX1 - Регистр 1 управления портовым мультиплексором*

## *РМХ2 - Регистр 2 управления портовым мультиплексором*

		автоматически очищается. Программное обеспечение также может быть стабильным в кристалле После настройки вручную очистите STSC0, цепь запуска кристалла выключится, экономя потребление энергии.
[4:3]	-	Зарезервировано
[2]	XIEN	Для включения внешнего тактового сигнала требуется одновременное подключение внешнего кварцевого генератора
[1]	E6EN	Включение универсальной функции ввода-вывода PE6, по умолчанию PE6 - функция AVREF
[0]	C6EN	Включает функцию ввода-вывода общего назначения для PC6, по умолчанию PC6 - вход внешнего сброса

## 10. Прерывание изменения уровня сигнала

- Источники прерываний с изменением уровня 40 контактов
- 5 записей прерываний

### 10.1. Обзор

Прерывания изменением уровня контактов инициируются на контактах PBn, PCn, PDn, PEn и PFn. До тех пор, пока прерывание изменением уровня контактов включено, прерывание может быть запущено, даже если эти контакты сконфигурированы как выходы. Это можно использовать для генерации программных прерываний. Любой разрешенный флеш-пейджер PBn вызовет прерывание PCI-вывода на уровне пин-кода, активирует вывод PIN-кода PCn, запустит PCI1, активирует PDn-щелчок, запустит PCI2, активированное переключение вывода PEn вызовет PCI3 , Разрешение каждого прерывания смены контактов управляет регистрами PCMSK0 ~ 4 соответственно. Все прерывания смены контактов обнаруживаются асинхронно и могут использоваться в качестве источников пробуждения в некоторых режимах ожидания.

### 10.2. Регистры Описания

#### Список регистров прерывания изменением уровня контактов

Регистр	Адрес	Значение по умолчанию	Описание
PCICR	0x68	0x00	Регистр управления прерыванием
PCIFR	0x3B	0x00	Регистр флага прерывания
PCMSK0	0x6B	0x00	Признак прерывания 0 изменения уровня контактов
PCMSK1	0x6C	0x00	Признак прерывания 1 изменения уровня контактов
PCMSK2	0x6D	0x00	Признак прерывания 2 изменения уровня контактов
PCMSK3	0x73	0x00	Признак прерывания 3 изменения уровня контактов
PCMSK4	0x74	0x00	Признак прерывания 4 изменения уровня контактов

*PCICR - регистр управления прерыванием изменением уровня контактов*

*PCIFR - регистр флагов прерывания изменением уровня контактов*

		соответственно. Выполнение процедуры обслуживания прерываний или запись «1» на бит PCIF4 очистит бит PCIF4.
3	PCIF3	Флаг прерывания 3. Изменение уровня любого из разрешенных PEn контактов устанавливает PCIF3. Когда установлены PCIE3 и глобальные прерывания, MCU перейдет к адресу записи прерывания PCI3. Разрешение прерывания PEn можно контролировать с помощью регистра PCMSK3, соответственно. Выполнение процедуры обслуживания прерываний или запись «1» на бит PCIF3 очистит бит PCIF3.
2	PCIF2	Флаг прерывания 2. Изменение уровня любого из разрешенных PDn контактов устанавливает PCIF2. Когда установлены PCIE2 и глобальные прерывания, MCU перейдет к адресу записи прерывания PCI2. Разрешение прерывания PDn можно контролировать с помощью регистра PCMSK2, соответственно. Выполнение процедуры обслуживания прерываний или запись «1» на бит PCIF2 очистит бит PCIF2.
1	PCIF1	Флаг прерывания 1. Изменение уровня любого из разрешенных PCn контактов устанавливает PCIF1. Когда установлены PCIE1 и глобальные прерывания, MCU перейдет к адресу записи прерывания PCI1. Разрешение прерывания PCn можно контролировать с помощью регистра PCMSK1, соответственно. Выполнение процедуры обслуживания прерываний или запись «1» на бит PCIF1 очистит бит PCIF1.
0	PCIF0	Флаг прерывания 0. Изменение уровня любого из разрешенных PBn контактов устанавливает PCIF0. Когда установлены PCIE0 и глобальные прерывания, MCU перейдет к адресу записи прерывания PCI0. Разрешение прерывания PBn можно контролировать с помощью регистра PCMSK0, соответственно. Выполнение процедуры обслуживания прерываний или запись «1» на бит PCIF0 очистит бит PCIF0.

*PCMSK0 - регистр 0 прерывания изменением уровня контактов*

#### **PCMSK0 - регистр 0 прерывания изменения уровня контактов**

*PCMSK1 - регистр 1 прерывания изменением уровня контактов*

PCMSK1 - регистр 1 прерывания изменения уровня контактов								
Адрес 0x6B			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	PCINT15	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7	PCINT15	Бит маски включения бит - 15.						
6	PCINT14	Бит маски включения бит - 14.						
5	PCINT13	Бит маски включения бит - 13.						
4	PCINT12	Бит маски включения бит - 12.						
3	PCINT11	Бит маски включения бит - 11.						
2	PCINT10	Бит маски включения бит - 10.						
1	PCINT9	Бит маски включения бит - 9.						
0	PCINT8	Бит маски включения бит - 8.						

**Пояснение**

При установке бит PCINT $n+8$  в «1», прерывание изменения уровня сигнала PC $n$  включено. Изменение уровня на выводах PC $n$  будет устанавливать PCIF1. Если бит PCIE1 и глобальное прерывание установлены, будет генерироваться прерывание PCI1. При установке бит PCINT  $n+8$  на «0» прерывание изменения уровня сигнала PC $n$  отключено. Бит  $n = 7\dots0$

*PCMSK2 - регистр 2 прерывания изменением уровня контактов*

*PCMSK3 - регистр 3 прерывания изменением уровня контактов*

<b>Бит</b>	<b>Имя</b>	<b>Определение бит</b>
7	PCINT31	Бит маски включения бит - 31.
6	PCINT30	Бит маски включения бит - 30.
5	PCINT29	Бит маски включения бит - 29.
4	PCINT27	Бит маски включения бит - 28.
3	PCINT27	Бит маски включения бит - 27.
2	PCINT26	Бит маски включения бит - 26.
1	PCINT25	Бит маски включения бит - 25.
0	PCINT24	Бит маски включения бит - 24.

**Пояснение**

При установке бит PCINT $n+24$  в «1», прерывание изменения уровня сигнала PEn включено. Изменение уровня на выводах PEn будет устанавливать PCIF3. Если бит PCIE3 и глобальное прерывание установлены, будет генерироваться прерывание PCI3. При установке бит PCINT $n+24$  в «0» прерывание изменения уровня сигнала PEn отключено. Бит  $n = 7\dots0$

#### *PCMSK4 - регистр 4 прерывания изменения уровня контактов*

<b>PCMSK3 - регистр 3 прерывания изменения уровня контактов</b>									
Адрес 0x6B			Значение по умолчанию: 0x00						
<b>Бит</b>	7	6	5	4	3	2	1	0	
<b>Имя</b>	PCINT39	PCINT38	PCINT37	PCINT36	PCINT35	PCINT34	PCINT33	PCINT32	
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
<b>Бит</b>	<b>Имя</b>	<b>Определение бит</b>							
7	PCINT39	Бит маски включения бит - 39.							
6	PCINT38	Бит маски включения бит - 38.							
5	PCINT37	Бит маски включения бит - 37.							
4	PCINT36	Бит маски включения бит - 36.							
3	PCINT35	Бит маски включения бит - 35.							
2	PCINT34	Бит маски включения бит - 34.							
1	PCINT33	Бит маски включения бит - 33.							
0	PCINT32	Бит маски включения бит - 32.							

**Пояснение**

При установке бит PCINT $n+32$  в «1», прерывание изменения уровня сигнала PFn включено. Изменение уровня на выводах PFn будет устанавливать PCIF4. Если бит PCIE4 и глобальное прерывание установлены, будет генерироваться прерывание PCI4. При установке бит PCINT $n+32$  в «0» прерывание изменения уровня сигнала PFn отключено. Бит  $n = 7\dots0$

## 11. Таймер/Счетчик 0 8-разрядный счетчик

- 8-разрядный счетчик
- Два независимых блока сравнения
- Счетчик автоматически очищается при совпадении сравнения и автоматически загружается
- Вывод с чередованием по фазе без прерывания импульсов
- Генератор частоты
- Счетчик внешних событий
- 10 Бит-часы-предделители
- Переполнение и сравнение совпадений с прерываниями

- Контроль мертвого времени.
- Шесть выбираемых источников триггеров автоматически отключает выход PWM.
- Высокоскоростное и высокое разрешение (500KHz @ 7Bit) PWM в режиме высокоскоростной синхронизации

## 11.1. Обзор

TC0 - это 8-разрядный модуль счетчика таймера общего назначения, который поддерживает выходы PWM и может точно генерировать сигналы.

TC0 содержит один генератор тактовых импульсов, один 8-разрядный счетчик, блок управления режимом формирования сигнала и два блока сравнения вывода. В то же время TC0 может совместно использовать 10-разрядный предварительный делитель с TC1, или он может использовать 10-разрядный предделитель независимо. Предделитель делит системные Такты clkio или высокоскоростные Такты rcm2x (2x внутреннего выходного сигнала генератора 32M RC генератора rc32m) для генерации счетчика тактов Clkt0.

Блок управления режимом генерации волны управляет режимом работы счетчика и формированием выходного сигнала сравнения.

В соответствии с различными режимами работы счетчик осуществляет очистку, увеличение или уменьшение каждого из счетчиков часов Clkt0.

Clkt0 может генерироваться внутренним источником синхронизации или внешним источником синхронизации. Когда значение счетчика счетчика TCNT0 достигает максимального значения (равное максимальному значению 0xFF или регистру сравнения OCR0A, определяемому как TOP, а максимальное значение определяется как MAX для отличия), счетчик будет очищен или уменьшен.

Когда значение счетчика TCNT0 достигает минимального значения (равное 0x00, определенное как BOTTOM), счетчик увеличивается. Когда значение счетчика TCNT0 счетчика достигает OCR0A / OCR0B, также известного как совпадение сравнения, выходной сигнал сравнения OC0A / OC0B очищается или устанавливается для генерации формы ШИМ.

Когда мертвое время включено, установленное мертвое время (количество тактов отсчета, соответствующее регистру DTR0) будет вставлено в сформированный сигнал PWM.

Программное обеспечение может очищать выходной сигнал OC0A / OC0B, очищая бит COM0A / COM0B до нуля или устанавливая соответствующий источник триггера.

Когда происходит событие триггера, аппаратное обеспечение автоматически очищает бит COM0A / COM0B, чтобы отключить выходной сигнал OC0A / OC0B.

Счетчик часов может быть сгенерирован внутренним или внешним источником синхронизации. Выбор источника синхронизации и выбор частотного деления контролируются битом CS0, расположенным в регистре TCCR0B. Подробнее см. Разделы предварительного делителя TC0 и TC1.

Счетчик имеет длину 8 бит и поддерживает двунаправленный подсчет. Режим генерации сигнала, который является режимом работы счетчика, управляет битами WGM0, расположенными в регистрах TCCR0A и TCCR0B.

В зависимости от режима работы счетчик очищает, увеличивает или уменьшает каждый счетчик часов Clkt0. Когда счетчик переполняется, устанавливается флаг переполнения счетчика TOV0 в регистре TIFR0.

Прерывание переполнения счетчика TC0 может быть сгенерировано, когда прерывание включено.

Блок сравнения результатов сравнивает значение счетчика TCNT0 с регистрами сравнения OCR0A и OCR0B вывода. Когда TCNT0 равно OCR0A или OCR0B, происходит сравнение и устанавливается флаг сравнения OCF0A или OCF0B вывода в регистре TIFR0.

При разрешенном прерывания может быть сгенерировано прерывание сравнения выходного сигнала TC0.

Следует отметить, что регистры OCR0A и OCR0B дублируются в режиме ШИМ. Двойная буферизация отключена в обычном режиме и режиме СТС. Когда счетчик достигает максимального или минимального значения, значения в регистре буфера синхронно обновляются в регистрах сравнения OCR0A и OCR0B. Подробнее см. Описание раздела режима работы.

Генератор сигналов генерирует выходные сигналы OC0A и OC0B сигналов осцилограммы на выходе на основе режима управления генерацией формы волны и управления режимом выходного сигнала сравнения с использованием сравнения, переполнения счетчика и т.п.

Конкретный метод генерации описывается в описании рабочего режима и описания раздела. Для вывода выходных сигналов OC0A и OC0B сигнала сравнения на соответствующие выводы, также должен быть установлен регистр направления данных этого вывода в качестве выхода. На следующем рисунке показана внутренняя структура TC0.

TC0 содержит один генератор тактовых импульсов счетчика, один 8-разрядный счетчик, два блока сравнения вывода и два блока управления генерацией сигналов.

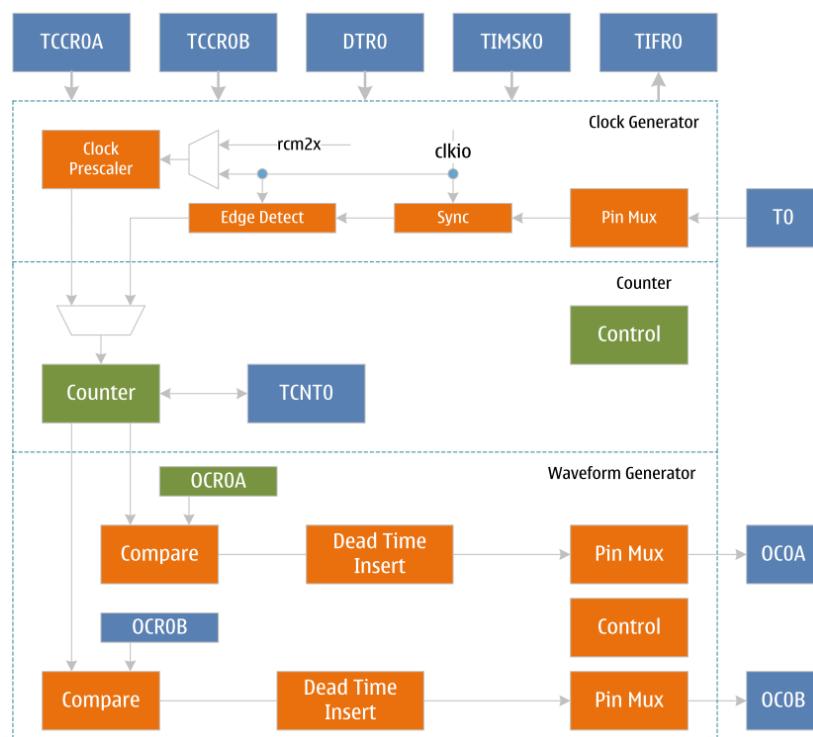


Рис.11.1 Структурная схема ТC0

## 11.2. Режим работы

Счетчик/таймер 0 имеет четыре разных режима работы, включая режим Normal, Compare Compare Clear (CTC), модуляция с быстрой шириной импульсов (FPWM) и фазорегуляторную модуляцию с широтно-импульсной модуляцией (PCPWM). Биты управления режимом WGM0 [2: 0] для выбора.

Ниже описаны эти четыре режима. Так как есть два независимых блока сравнения вывода, обозначенные соответственно «A» и «B», нижний вывод «x» представляет собой два выходных канала сравнения.

### 11.2.1. Нормальный режим

Нормальный режим - это самый простой режим работы счетчика таймера.

В это время биты WGM0[2: 0] = 0, а максимальное значение TOP счетчика - MAX (0xFF). В этом режиме счетчик увеличивается на один для каждого такта счета.

Когда счетчик достигает переполнения TOP, он возвращается в BOTTOM, чтобы снова начать накапливать. Флаг переполнения счетчика таймера TOV0 устанавливается в том же такте счета счетчика TCNT0, значение которого становится равным нулю.

Флаг TOV0 в этом режиме похож на 9-й бит счетчика, он будет установлен и не будет очищен.

Процедура обслуживания прерывания переполнения автоматически очищает флаг TOV0, и программное обеспечение может использовать его для увеличения разрешения счетчика.

В нормальном режиме нет специального случая, и новые значения счета могут быть записаны в любое время.

Форма сигнала выходного сигнала сравнения OC0x может быть получена только тогда, когда регистр направления данных вывода OC0x задан как выход. Когда COM0x = 1, сигнал OC0x будет переключен при совпадении сравнения. В этом случае частоту формы волны можно вычислить по следующей формуле:

$$f_{oc0xnormal} = f_{sys} / (2 * N * 256)$$

где N - обозначает Коэффициент предделителя (1, 8, 64, 256 или 1024).

Модуль сравнения результатов может использоваться для генерации прерывания, но прерывания не рекомендуются в нормальном режиме, который потребляет слишком много времени процессора.

### 11.2.2. Режим СТС

Когда установлено значение WGM0 [2: 0] = 2, счетчик/таймера 0 переходит в режим СТС, а максимальное значение счетчика TOP - OCR0A. В этом режиме счетчик увеличивается на единицу для каждого счетчика часов. Когда значение счетчика TCNT0 равно TOP, счетчик очищается.

OCR0A определяет максимальный счет, который является разрешением

счетчика. Этот режим позволяет пользователю легко контролировать частоту выходного сигнала сравнения, а также упрощает операцию подсчета внешних событий. Когда счетчик достигает максимального значения счетчика, устанавливается флаг соответствия OCF0 на выходе, и прерывание будет генерироваться, когда установлен соответствующий бит разрешения прерывания.

Регистр OCR0A, который является максимальным количеством, может быть обновлен в процедуре обслуживания прерываний. В этом режиме OCR0A не использует двойную буферизацию.

**Внимание:** при обновлении максимального значения OCR0A до минимального значения, когда счетчик работает с предварительным делителем или без него. Если значение, записанное в OCR0A, меньше текущего значения TCNT0, счетчик пропустит равенство сравнения. Перед тем, как произойдет следующее сравнение, счетчик должен просчитывать до TOP, а затем подсчитывать от BOTTOM до OCR0A.

Как и в обычном режиме, значение count возвращается к счету счетчика BOTTOM, чтобы установить флаг TOV0.

Форма сигнала выходного сигнала сравнения OC0x может быть получена только тогда, когда регистр направления данных вывода OC0x задан как выход. Когда COM0x = 1, сигнал OC0x переключается при равенстве сравнения.

В этом случае частоту формы волны можно вычислить по следующей формуле:

$$f_{OC0x} = f_{sys} / (2 * N * (1 + OCR0x))$$

где N - обозначает Коэффициент предделителя (1, 8, 64, 256 или 1024).

Из формулы видно, что когда OCR0A задано равным 0x0, и нет предварительного делителя, может быть получен выходной сигнал с максимальной частотой  $f_{sys} / 2$ .

### 11.2.3. Быстрый режим PWM

Когда установлено значение WGM0 [2: 0] = 3 или 7, счетчик 0 таймера входит в режим быстрой PWM и может использоваться для генерации высокочастотных сигналов ШИМ.

Максимальное значение счета TOP равно MAX (0xFF) или OCR0x соответственно. Разница между быстрым режимом PWM и другими режимами PWM заключается в том, что это односторонняя операция. Счетчик считает до TOP от минимального значения 0x00, а затем сбрасывается в BOTTOM для повторного подсчета. Когда значение счета TCNT0 достигает OCR0x или BOTTOM, выходной сигнал сравнения OC0x устанавливается или очищается в зависимости от настройки режима вывода сравнения COM0x.

Подробности см. В описании регистра.

Из-за односторонней работы рабочая частота режима быстрой PWM в два раза выше, чем в режиме PWM с фазовой модификацией, используя двунаправленную операцию.

Высокочастотные характеристики делают быстрый режим PWM подходящим для регулирования мощности, выпрямления и применения ЦАП. Высокочастотные

сигналы могут уменьшить размер внешних компонентов (индуктивность, емкость и т. д.), Тем самым снижая стоимость системы.

Когда значение count достигает максимального значения, устанавливается флаг переполнения счетчика таймера TOV0 и значение буфера сравнения обновляется до значения сравнения.

Если прерывание включено, регистр OCR0x буфера сравнения может быть обновлен в процедуре обслуживания прерываний. Форма сигнала выходного сигнала сравнения OC0x может быть получена только тогда, когда регистр направления данных вывода OC0x задан как выход.

Частоту формы волны можно вычислить по следующей формуле:

$$f_{oc0xfpw} = f_{sys} / (N * (1 + TOP))$$

где N - обозначает Коэффициент предделителя (1, 8, 64, 256 или 1024).

Когда встречается сравнение между TCNT0 и OCR0x, генератор формы сигнала (очищает) сигнал OC0x. Когда TCNT0 очищается, генератор формы волны очищает (устанавливает) сигнал OC0x для генерации волны ШИМ.

В результате экстремальное значение OCR0x создаст специальную форму ШИМ. Когда OCR0x установлен в 0x00, выходной ШИМ имеет узкий всплеск в каждом (1 + TOP) счетчике часов. Когда OCR0x установлен на максимальное значение, выходной сигнал постоянно высокий или низкий.

#### 11.2.4. Режим PWM с коррекцией фаз

Когда установлено WGM0 [2: 0] = 1 или 5, счетчик 0 таймера входит в режим PWM с фазовой коррекцией, а максимальное значение счетчика равно MAX (0xFF) или OCR0A соответственно. Счетчик работает с двухточечным шагом от BOTTOM до TOP, а затем вниз до BOTTOM и повторяет эту операцию. Когда счетчик достигает TOP и BOTTOM, оба изменяют направление подсчета, а значение счетчика остается только на одном счетчике тактав TOP или BOTTOM.

Во время процесса приращения или уменьшения, когда значение счетчика TCNT0 совпадает с OCR0x, выходной сигнал сравнения OC0x будет очищен или установлен в зависимости от настройки режима вывода COM0x.

По сравнению с односторонней работой максимальная частота, доступная для двунаправленной работы, меньше, но ее отличная симметрия более подходит для управления двигателем.

Фазовая коррекция В режиме PWM флаг TOV0 устанавливается, когда счетчик достигает BOTTOM, а буфер сравнения обновляется до значения сравнения, когда счетчик достигает TOP.

Если прерывание включено, регистр OCR0x буфера сравнения может быть обновлен в процедуре обслуживания прерываний. Форма сигнала выходного сигнала сравнения OC0x может быть получена только тогда, когда регистр направления данных вывода OC0x задан как выход.

Частоту формы волны можно вычислить по следующей формуле:

$$f_{oc0xcpwm} = f_{sys} / (N * TOP * 2)$$

где N - коэффициент предварительной шкалы (1, 8, 64, 256 или 1024).

Во время перерасчета генератор сигналов очищает (устанавливает) сигнал OC0x, когда TCNT0 соответствует OCR0x.

Во время подсчета, генератор сигналов устанавливает (очищает) сигнал OC0x, когда TCNT0 соответствует OCR0x. Из-за этого экстремальное значение OCR0x генерирует специальную волну PWM. Когда OCR0x установлен на максимальное или минимальное значение, выходной сигнал OC0x остается низким или высоким.

Чтобы обеспечить симметричность выходной ШИМ-волны с обеих сторон минимального значения, сигнал OC0x будет перевернут в двух случаях, когда совпадение сравнения не произойдет.

Первый случай - когда значение OCR0x изменяется с максимального значения 0xFF на другие данные. Когда OCR0x является максимальным значением, а значение count достигает максимального значения, выход OC0x совпадает с результатом сравнения в предыдущем счетчике, т.е. OC0x не изменяется. Значение нового OCR0x (не 0xFF) обновляется в это время. Значение OC0x сохраняется до тех пор, пока не произойдет совпадение сравнения, и произойдет сальто. В этом случае сигнал OC0x не является симметричным относительно минимального значения, поэтому сигнал OC0x необходимо переключать, когда TCNT0 достигает максимального значения. То есть, нет первого случая опрокидывания сигнала OC0x при совпадении сравнения.

Во втором случае, когда TCNT0 начинает отсчет с более высокого значения, чем OCR0x, сравнение сравнивается с потерями, что вызывает асимметричную ситуацию. Также необходимо перевернуть сигнал OC0x для достижения симметрии с обеих сторон минимального значения.

#### 11.2.5. Мертвый контроль

Когда бит DTEN0 установлен на «1», функция вставки мертвого времени включена. Выходные сигналы OC0A и OC0B будут вставлены в установленное мертвое время на основе формы сигнала, создаваемой выходным сигналом сравнения канала B. Длительность времени равна DTR0. Значение времени, соответствующее тактам счетчика регистра.

Как показано на [рисунке ниже](#), вставки мертвого времени OC0A и OC0B основаны на форме выходного сигнала сравнения канала B. Когда COM0A и COM0B являются «2» или «3», полярность формы сигнала OC0A совпадает с полярностью сигнала OC0B. Когда COM0A и COM0B являются «2» или «3», форма волны OC0A и форма волны OC0B соответственно, противоположная полярность.

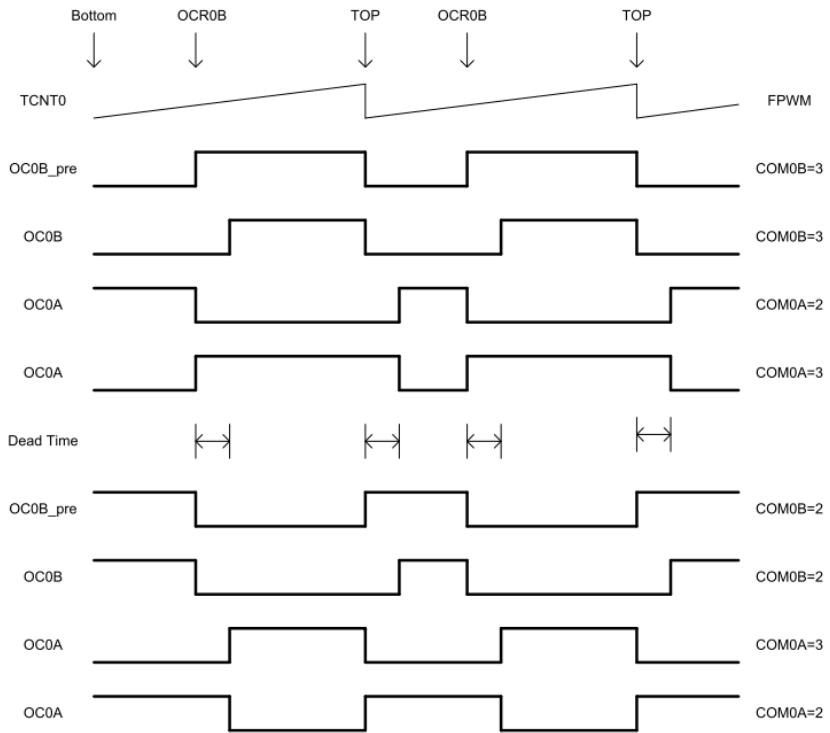


Рис. 11.1 Контроль мертвого времени таймера ТС0 в режиме FPWM

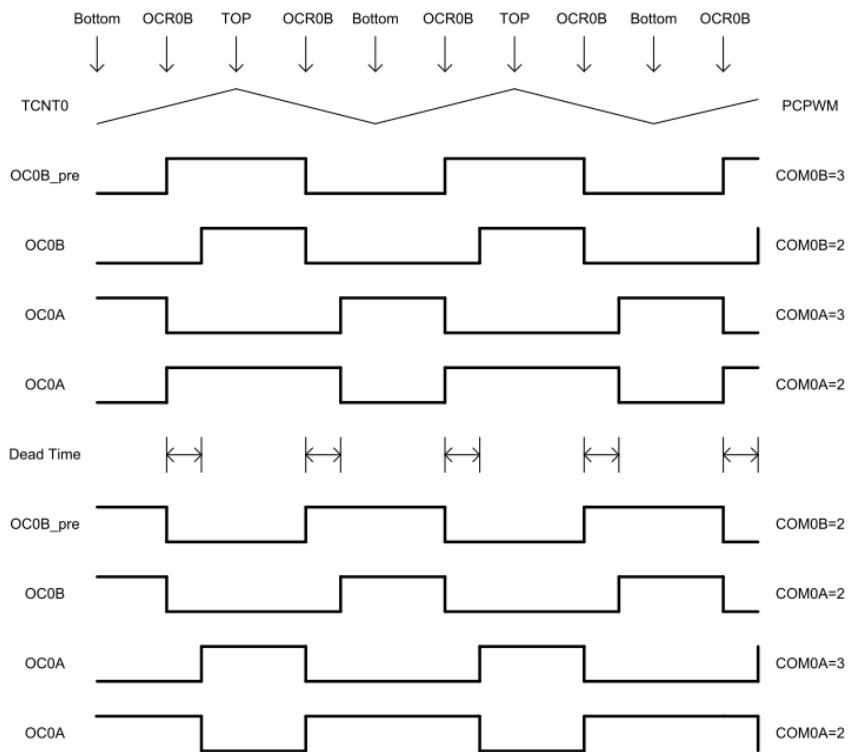


Рис.11. 2. Контроль времени ТС0 в режиме PCPWM

Когда бит DTEN0 установлен на «0», функция вставки мертвого времени отключена, а выходные сигналы OC0A и OC0B являются формами сигналов, генерируемыми соответствующими результатами сравнения.

#### 11.2.6. Режим высокоскоростной синхронизации

В высокоскоростном тактовом режиме в качестве источника синхронизации

для подсчета используется высокочастотный синхросигнал для генерации высокочастотных и высокочастотных сигналов PWM.

Этот высокочастотный тактовый генератор генерируется путем умножения выходного тактового сигнала rc32m внутреннего 32M RC-генератора на 2 раза.

Поэтому перед входом в высокочастотный режим необходимо включить функцию частотного умножения внутреннего 32M RC-генератора, то есть установить бит F2XEN в регистре TCKCSR и дождаться определенного времени до тех пор, пока выходной сигнал умноженного тактового сигнала не станет стабильным. Затем бит TC2XS0 в TCKCSR может быть установлен для установки счетчика таймера в режим высокоскоростных часов.

В этом режиме системные тактовые импульсы и высокоскоростные такты являются асинхронными, а некоторые регистры (см. Список регистров TC0) работают в высокоскоростном тактовом домене, поэтому конфигурация и считывание этих регистров также асинхронны, и при работе необходимо соблюдать осторожность. Нет особых требований к несекретным операциям чтения и записи в регистрах в высокоскоростном домене синхронизации.

При выполнении последовательных чтений и записи подождите системные такты. Выполните следующие действия:

- 1) Запишите регистр A;
- 2) Дождитесь системных тактов (NOP или зарегистрировать в операционных системах такты);
- 3) Чтение или запись регистров A или B.
- 4) Дождитесь системных тактов (зарегистрируйтесь в режиме NOP или тактов операционной системы).

При чтении регистров в высокоскоростном тактовом режиме регистры, отличные от TCNT0, могут считываться напрямую. Когда счетчик все еще подсчитывает, значение TCNT0 изменяется с помощью высокоскоростного тактового сигнала, и счетчик может быть приостановлен (установите CS0 на ноль). Прочтите значение TCNT0.

## 11.2. Регистры Описания

### Список регистров TC0

Регистр	Адрес	Значение по умолчанию	Описание
TCCR0A*	0x44	0x00	Контрольный регистр TC0 А
TCCR0B*	0x45	0x00	Контрольный регистр TC0 В
TCNT0*	0x46	0x00	Регистр значений счетчика TC0
OCR0A*	0x47	0x00	Регистр сравнения выходных данных TC0 А
OCR0B*	0x48	0x00	Регистр сравнения выходных данных TC0 В
DSX0*	0x49	0x00	Регистр управления источником триггера TC0
DTR0*	0x4F	0x00	Регистр мертвый времени TC0
TIMSK0	0x6E	0x00	Регистр маски прерывания TC0
TIFR0	0x35	0x00	Регистр флага прерываний TC0
TCKCSR	0xEC	0x00	TC Clock Control и регистр состояния

**Примечание:** Регистры с «\*» работают в системных тактах и высокоскоростном тактовом домене. Регистры без «\*» работают только в домене системных тактов.

## TCCR0A - Регистр управления ТС0 А

TCCR0A - Регистр управления ТС0 А								
Адрес: 0x44				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	COM0A1	COM0A0	COM0B1	COM0B0	DOC0B	DOC0A	WGM01	WGM00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7	COM0A1	Подробнее см. Описание таблицы управления выходным сравнением.						
6	COM0A0							
5	COM0B1	Подробнее см. Описание таблицы управления выходным сравнением.						
4	COM0B0							
3	DOC0B	При установке бит DOC0B в «1» активируется сигнал источника запуска триггера выключения сравнения OC0B выхода. Когда происходит событие триггера, аппаратное обеспечение автоматически отключает выход осциллографии OC0B. Когда бит DOC0B установлен на «0», сигнал сравнения OC0B выхода выключения источника запуска отключается. Когда происходит событие триггера, выход осциллографии OC0B не отключается.						
2	DOC0A	При установке бит DOC0A в «1» активируется сигнал источника запуска триггера выключения сравнения OC0A выхода. Когда происходит событие триггера, аппаратное обеспечение автоматически отключает выход осциллографии OC0A. Когда бит DOC0A установлен на «0», сигнал сравнения OC0A выхода выключения источника запуска отключается. Когда происходит событие триггера, выход осциллографии OC0A не отключается..						
1	WGM01	Подробнее см. Описание таблицы режимов генерации волны.						
0	WGM00							

## TCCR0B - Регистр управления ТС0 В

TCCR0B - Регистр управления ТС0 В													
Адрес: 0x45				Значение по умолчанию: 0x00									
Бит	7	6	5	4	3	2	1	0					
Имя	FOC0A	FOC0B	OC0AS	DTEN0	WGM02	CS02	CS01	CS00					
R/W	W	W	R/W	R/W	R/W	R/W	R/W	R/W					
Бит	Имя	Определение бит											
7	FOC0A	Бит FOC0A активный только, когда биты WGM определяют режим без PWM.											
		COM0A1	COM0A0	Описание									
		0	0	Нормальных операций порта, OC0A отключено.									
		0	1	Бистабильная схема OC0A на Сравнении Компаратора.									
		1	0	Очистить OC0A на Сравнении Компаратора.									
		1	1	Установить OC0A на Сравнении Компаратора.									
Чтение FOC0A всегда возвращает ноль.													
6	FOC0B	Бит FOC0B активный только, когда биты WGM определяют режим без PWM.											
		COM0B1	COM0B0	Описание									
		0	0	Нормальных операций порта, OC0B отключено.									
		0	1	Бистабильная схема OC0B на Сравнении Компаратора.									
		1	0	Очистить OC0B на Сравнении Компаратора.									
		1	1	Установить OC0B на Сравнении Компаратора.									

		Чтение FOC0B всегда возвращает ноль.																		
5	OC0AS	Контрольный бит выбора выходного порта OC0A. Когда бит OC0AS установлен на «0», форма волны OC0A выводится из вывода PD6. Когда бит OC0AS установлен в «1», форма волны OC0A выводится из вывода PE4 ( <b>действительна в корпусе QFP32</b> ).																		
4	DTEN0	Когда бит DTEN0 установлен на «1», включение мертвого времени включено. Оба OC0A и OC0B вставляют мертвое время в форму волны, генерируемую выходом сравнения канала B. Введенный мертвый временной интервал определяется временем отсчета, соответствующим регистру DTR0. Полярность выходного сигнала OC0A определяется соответствием между COM0 и COM0B. См. Таблицу полярности формы сигнала после мертвого времени вставки OC0A. Когда бит DTEN0 установлен на «0», вставка мертвого времени запрещена. Сигналы OC0A и OC0B являются формами сигналов, генерируемыми соответствующими результатами сравнения.																		
3	WGM02	WGM02 и WGM00 и WGM01 вместе образуют режим WGM0 управления генерацией формы волны [2: 0] для управления режимом подсчета счетчиков и режимом генерации формы сигнала. Подробнее см. В описании таблицы режима генерации волны.																		
2	CS02	Предделение тактов источника синхронизации для счетчика времени 0.																		
1	CS01	<table border="1"> <thead> <tr> <th>CS0[2:0]</th> <th>Описание</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Нет источника синхронизации, стоп-счет</td> </tr> <tr> <td>1</td> <td>clksys</td> </tr> <tr> <td>2</td> <td>clksys / 8 из предделителя</td> </tr> <tr> <td>3</td> <td>clksys / 64 из предделителя</td> </tr> <tr> <td>4</td> <td>clksys / 256 из предделителя</td> </tr> <tr> <td>5</td> <td>clksys / 1024 от предделителя</td> </tr> <tr> <td>6</td> <td>Внешние такты T0, триггер со спадающим фронтом</td> </tr> <tr> <td>7</td> <td>Внешние такты T0, триггер с нарастающим фронтом</td> </tr> </tbody> </table>	CS0[2:0]	Описание	0	Нет источника синхронизации, стоп-счет	1	clksys	2	clksys / 8 из предделителя	3	clksys / 64 из предделителя	4	clksys / 256 из предделителя	5	clksys / 1024 от предделителя	6	Внешние такты T0, триггер со спадающим фронтом	7	Внешние такты T0, триггер с нарастающим фронтом
CS0[2:0]	Описание																			
0	Нет источника синхронизации, стоп-счет																			
1	clksys																			
2	clksys / 8 из предделителя																			
3	clksys / 64 из предделителя																			
4	clksys / 256 из предделителя																			
5	clksys / 1024 от предделителя																			
6	Внешние такты T0, триггер со спадающим фронтом																			
7	Внешние такты T0, триггер с нарастающим фронтом																			
0	CS00																			

#### COM0n - Быстрый Режим PWM Управление сравнением

COM0A1	COM0A0	Описание
0	0	Нормальных операций порта, OC0A разъединяется.
0	1	WGM02 = 0: Нормальная Операция Порта, Разъединенный OC0A WGM02 = 1: Бистабильная схема OC0A на Сравнении Компаратора
1	0	Очистить OC0A на Сравнении, Установить OC0A в основе (неинвертирующий режим)
1	1	Установить OC0A на Сравнении, Очистить OC0A в BOTTOM (инвертирующий режим)

**Примечание:** 1. Особый случай происходит, когда OCR0A равняется TOP, и COM0A1 устанавливается. В этом случае компаратор сравнение игнорируется, но устанавливается или очищается при BOTTOM.

Подробнее см. В разделе Режим быстрый PWM.

#### COM0n - Режим PWM с коррекцией фаз Управление сравнением

COM0A1	COM0A0	Описание
0	0	Нормальных операций порта, OC0A разъединяется.
0	1	WGM02 = 0: Нормальная Операция Порта, Разъединенный OC0A. WGM02 = 1: Бистабильная схема OC0A на Сравнении Компаратора.

1	0	Очистить OC0A на Сравнении Компаратора когда Счет Вверх. Установленный OC0A на Сравнении Компаратора когда Счет Вниз.
1	1	Установленный OC0A на Сравнении Компаратора когда Счет Вверх. Очистить OC0A на Сравнении Компаратора когда Счет Вниз.

**Примечание:** 1. Особый случай происходит, когда OCR0A равняется TOP, и COM0A1 устанавливается. В этом случае, Компаратор Сравнение игнорируется, но установлен или очищен делается в TOP.

Подробности см. в Режим Правильной Фазы PWM.

WGMO - Управление режимом генерации сигнала.

WGM0 [2:0]	Режим работы	Значение TOP	Обновить момент OCR0X	Установите момент TOV0
0	Обычный	0xFF	Непосредственный	MAX
1	PCPWM	0xFF	TOP	BOTTOM
2	CTC	OCR0A	Immediate	MAX
3	FPWM	0xFF	TOP	MAX
4	оговорок	-	-	-
5	PCPWM	OCR0A	TOP	BOTTOM
6	Резервировано	-	-	-
7	FPWM	OCR0A	TOP	TOP

В следующей таблице показано управление полярностью выходного сигнала сигнала ОС0А, когда включено время отключения.

Контроль полярности формы сигнала сигнала ОС0А в режиме включения мертвого времени

<b>DTEN0</b>	<b>COM0A [1:0]</b>	<b>COM0B [1:0]</b>	<b>Описание</b>
0	-	-	полярность сигнала OC0A, контролируемая OC0A, сравнивает выходной режим
1	0	-	OC0A отключен, операция ввода-вывода общего назначения
1	1	-	Зарезервировано
1	2	2	Полярности сигналов OC0A и OC0B одинаковы
		3	Полярности сигналов OC0A и OC0B противоположны
1	3	2	Полярности сигналов OC0A и OC0B противоположны
		3	Полярности сигналов OC0A и OC0B одинаковы

**Примечание:** Полярность выходного сигнала сигнала OC0B управляется режимом вывода сравнения OC0B, который аналогичен тому, когда режим мертвого времени не включен.

## *TCCR0C - Регистр управления ТС0 С*

		источника триггера для выключения сигнала сравнения OC0A/OC0B. Когда бит DOC0A / DOC0B равен «1», нарастающий фронт выбранного флага прерывания автоматически отключает выходной сигнал OC0A/OC0B. При установке бита DSX07 в «0» переполнение <b>TC1</b> отключается как источник триггера для выключения выходного сигнала сравнения OC0A/OC0B.
6	DSX06	При установке бита DSX06 в «1» переполнение <b>TC2</b> активируется в качестве источника триггера для выключения сигнала сравнения OC0A/OC0B. Когда бит DOC0A / DOC0B равен «1», нарастающий фронт выбранного флага прерывания автоматически отключает выходной сигнал OC0A / OC0B. При установке бита DSX06 в «0» переполнение <b>TC2</b> отключается как источник триггера для выключения выходного сигнала сравнения OC0A/OC0B.
5	DSX05	При установке бита DSX05 на «1» уровень изменения <b>Вывода 0</b> включается в качестве источника триггера для выключения выходной формы сигнала сравнения OC0A / OC0B. Когда бит DOC0A / DOC0B установлен в «1», передний фронт бит регистра флага прерывания выбранного источника триггера автоматически отключает выходной сигнал OC0A/OC0B. При установке бита DSX05 в «0», изменение <b>Вывода 0</b> отключается как источник триггера для выключения выходного сигнала сравнения OC0A/OC0B.
4	DSX04	Когда бит DSX04 установлен в «1», внешнее <b>Прерывание 0</b> включается в качестве источника триггера для выключения выходной сигнала сравнения OC0A/OC0B. Когда бит DOC0A/DOC0B равен «1», передний фронт бита регистра флагка выбранного прерывания автоматически отключает выходной сигнал OC0A/OC0B. Когда бит DSX04 установлен в «0», <b>Прерывание 0</b> отключается как источник триггера для выключения выходного сигнала сравнения OC0A/OC0B.
3	-	
2	-	
1	DSX01	Когда бит DSX01 установлен в «1», <b>Аналоговый компаратор 1</b> включен как источник триггера для выключения выходной сигнала сравнения OC0A/OC0B Когда бит DOC0A/DOC0B равен «1», передний фронт флага прерывания выбранного источника триггера автоматически отключает выходной сигнал OC0A/OC0B. Когда бит DSX01 равен «0», <b>Аналоговый компаратор 1</b> отключается как триггер, чтобы отключить выходной сигнал сравнения OC0A/OC0B.
0	DSX00	Когда бит DSX01 установлен в «1», <b>Аналоговый компаратор 0</b> включен как источник триггера для выключения выходной сигнала сравнения OC0A/OC0B Когда бит DOC0A/DOC0B равен «1», передний фронт флага прерывания выбранного источника триггера автоматически отключает выходной сигнал OC0A/OC0B. Когда бит DSX01 равен «0», <b>Аналоговый компаратор 0</b> отключается как триггер, чтобы отключить выходной сигнал сравнения OC0A/OC0B.

#### OC0A / OC0B - Отключить управление Выбор источника

DOC0x	DSX0n = 1	Источник	Описание
0	-	-	Бит DOC0x равен «0», функция выключения источника выходного сигнала отключена.
1	0	Аналоговый компаратор 0	Передний фронт ACIF0 отключает выходной сигнал OC0x
1	1	Аналоговый компаратор 1	Передний фронт ACIF1 отключает выходной сигнал OC0x
1	4	Внешнее прерывание 0	Передний фронт INTF0 отключает выходной сигнал OC0x

1	5	Изменение уровня контакта 0	Передний фронт PCIF0 отключает выходной сигнал OC0x
1	6	Переполнение TC2	Передний фронт TOV2 отключает выходной сигнал OC0x
1	7	Переполнение TC1	Передний фронт TOV1 отключает выходной сигнал OC0x

**Примечание:** 1) Когда DSX0n = 1 указывает, что n-й бит регистра DSX0 равен 1, биты регистра могут быть установлены одновременно.

## *TCNT0 - Регистр счетчиков ТС0*

### *OCR0A - Регистр сравнения выходных данных ТС0 А*

## OCR0B - Регистр сравнения выходных данных ТС0 В

OCR0A - Регистр сравнения выходных данных ТС0 А								
Адрес: 0x48				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	OCR0B7	OCR0B6	OCR0B5	OCR0B4	OCR0B3	OCR0B2	OCR0B1	OCR0B0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	OCR0B	Регистр сравнения выходных данных ТС0. OCR0B содержит 8-битные данные, которые непрерывно сравниваются со значением счетчика TCNT0. Равенство сравнения может использоваться для генерации прерывания сравнения вывода или для генерации формы сигнала на выводе OC0B. При использовании режима PWM регистр OCR0B использует регистр с двойным буфером. Однако двойная буферизация отключена в нормальном режиме и соответствует режиму очистки. Двойная буферизация синхронизирует обновление регистра OCR0B с максимальным или минимальным временем отсчета, предотвращая генерацию асимметричных импульсов ШИМ и устранивая возмущающие импульсы. При использовании функции двойной буферизации CPU обращается к регистру буфера OCR0B. Когда функция двойной буферизации отключена, CPU обращается к самому OCR0B.						

## TIMSK0 - Регистр прерывания ТС0

TIMSK0 - Регистр прерывания ТС0								
Адрес: 0x6E				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	-	-	-	-	-	OCIE0B	OCIE0A	TOIE0
R/W	-	-	-	-	-	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:3	-	Зарезервировано						
2	OCIE0B	Когда бит OCIE0B, установлен в «1» и глобальное прерывание установлено - прерывание Компаратора В Сравнение разрешено. Соответствующее прерывание выполняется, если Сравнение Компаратора в Таймер/Счетчик происходит, то есть, когда бит OCF0B устанавливается в TIFR0. Когда бит OCIE0B равен «0» выход ТС0 прерывания совпадения В - отключен.						
1	OCIE0A	Когда бит OCIE0A, установлен в «1» и глобальное прерывание установлено - прерывание Компаратора А Сравнение разрешено. Соответствующее прерывание выполняется, если Сравнение Компаратора в Таймер/Счетчик происходит, то есть, когда бит OCF0A устанавливается в TIFR0. Когда бит OCIE0A равен «0» выход ТС0 прерывания совпадения А - отключен.						
0	TOIE0	Бит разрешения прерывания переполнения ТС0. Когда бит TOIE0 установлен в «1», а глобальное прерывание установлено, прерывание переполнения ТС0 включено. Прерывание генерируется при переполнении ТС0, т. е. устанавливается бит TOV0 в TIFR. Когда бит TOIE0 равен «0», прерывание переполнения ТС0 отключено.						

## TIFR0 - Регистр флагов прерывания ТС0

TIFR0 - Регистр флагов прерывания ТС0								
Адрес: 0x35				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	OC0A	OC0B	-	-	-	OCF0B	OCF0A	TOV0
R/W	R/0	R/0	-	-	-	R/W	R/W	R/W

Бит	Имя	Определение бит
7	OC0A	На выходе сравнивается сигнал осцилограммы OC0A. Выходной сигнал сигнала осцилограммы OC0A, считываемый программным обеспечением, но не записываемый. Программное обеспечение может считывать значение бит OC0A для получения полярности сигнала формы сигнала, которое должно быть выведено до того, как сигнал OC0A будет включен на соответствующем выводе IO, и может быть изменен путем конфигурирования бит COM0A и установки бит FOC0A. Его полярность позволяет избежать посторонних импульсов помех после включения выхода сигнала OC0A на соответствующем IO-контакте.
6	OC0B	На выходе сравнивается сигнал осцилограммы OC0B. Выходной сигнал сигнала осцилограммы OC0B, считываемый программным обеспечением, но не записываемый. Программное обеспечение может считывать значение бит OC0B для получения полярности сигнала формы сигнала, которое должно быть выведено до того, как сигнал OC0B будет включен на соответствующем выводе IO, и может быть изменен путем конфигурирования бит COM0B и установки бит FOC0B. Его полярность позволяет избежать посторонних импульсов помех после включения выхода сигнала OC0D на соответствующем IO-контакте.
5:3		Зарезервировано
2	OCF0B	Выход TC0 флаг сравнения B. Когда TCNT0 равен OCR0B, блок сравнения дает сигнал равенства и устанавливает флаг сравнения OCF0B. Если Флаг прерывания сравнение B OCIE0B установлено в «1», и установлен флаг глобального прерывания, будет генерироваться прерывание сравнения с результатом B. Флаг OCF0B очищается автоматически при выполнении процедуры обслуживания прерывания или может быть очищен путем записи «1» в бит OCF0B.
1	OCF0A	Выход TC0 флаг сравнения A. Когда TCNT0 равен OCR0A, блок сравнения дает сигнал равенства и устанавливает флаг сравнения OCF0A. Если Флаг прерывания сравнение A OCIE0A установлено в «1», и установлен флаг глобального прерывания, будет генерироваться прерывание сравнения с результатом A. Флаг OCF0A очищается автоматически при выполнении процедуры обслуживания прерывания или может быть очищен путем записи «1» в бит OCF0A.
0	TOV0	Флаг переполнения TC0. Когда счетчик переполняется, устанавливается флаг переполнения TOV0. Если бит разрешения прерывания переполнения TOIE0 установлен на «1» и установлен флаг глобального прерывания, будет генерироваться прерывание переполнения. TOV0 автоматически очищается, когда выполняется эта процедура прерывания, или ее можно очистить, записав «1» в бит TOV0.

*DTR0 - регистр контроля мертвого времени ТС0*

DTR0 - регистр контроля мертвого времени ТС0

3:0	DTR0L	TC0 Dead time register low. Когда бит DTEN0 регистра TCCR0B установлен в «1», OC0A и OC0B образуют дополнительный выход. Управление включением мертвого времени включено. Время задержки, установленное на канале OC0A, определяется DTR0L. Длительность времени - это тактовые импульсы DTR0H. Соответствующее время.
-----	-------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 12. TMR1 - Таймер/Счетчик 1

- 16-разрядный структура, позволяющая формировать 16-битный PWM
- 2 независимых устройства сравнения с выходами
- Двойной буферный регистр сравнения результатов
- 1 блок захвата ввода
- Подавитель шума захвата входного сигнала
- Сравнение соответствия Автоматическая очистка счетчика и автоматическая загрузка
- ШИМ с коррекцией по фазе без прерывистых импульсов
- Переменный период PWM
- Генератор частоты
- Счетчик внешних событий
- Четыре независимых источника прерываний
- ШИМ с тайм-временем управления 6 опций Источник триггера автоматически отключает выход PWM.
- Высокоскоростная (500KHZ @ 7BIT) PWM генерируется в высокоскоростном режиме синхронизации.

### 12.1. Обзор

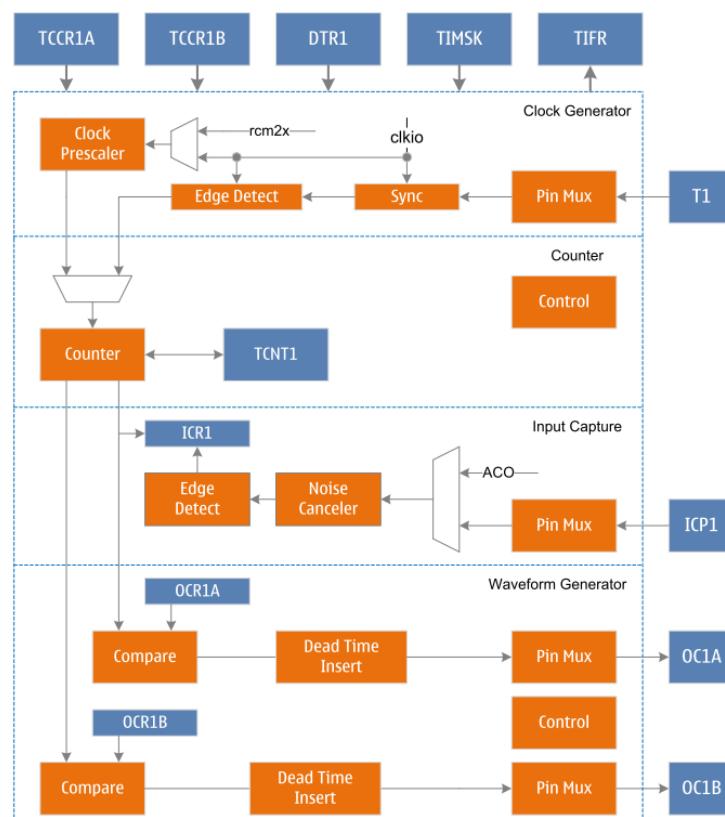


Рис.12.1. Структурная схема ТC1

TC1 - это универсальный 16-разрядный счетчик счетчиков, который поддерживает выходы PWM и может точно генерировать сигналы. TC1 содержит 16-разрядный счетчик, блок управления режимом формирования сигнала, два независимых вывода блока сравнения и блок ввода.

TC1 может совместно использовать 10-битный предварительный делитель с TC0 или может независимо использовать 10-разрядный предварительный делитель. Предделитель делит системные такты clkio или высокоскоростные такты gcm2x (2x внутреннего выходного сигнала генератора 32M RC генератора gc32m) для генерации счетчика тактов Clkt1.

Блок управления режимом генерации волны управляет режимом работы счетчика и формированием формы сигнала сравнения. В соответствии с различными режимами работы счетчик очищает, увеличивает или уменьшает каждый счет тактов Clkt1. Такты Clkt1 могут быть получены от внутреннего источника синхронизации или внешнего источника синхронизации.

Когда значение счета счетчика TCNT1 достигает максимального значения (равное максимальному значению 0xFFFF, фиксированному значению, регистру сравнения OCR1A выходного сигнала или регистру ввода ICR1, определяемому как TOP, для отличия максимальное значение определяется как MAX), счетчик очищается или вычитается при обратном счете.

Когда значение счетчика счетчиков TCNT1 достигает минимального значения (равное 0x0000, определенное как BOTTOM), счетчик увеличивается.

Когда значение счетчика TCNT1 счетчика достигает равенства OCR1A или OCR1B, выходной сигнал сравнения OC1A или OC1B очищается или устанавливается для генерации сигнала PWM.

Когда мертвое время включено, установленное мертвое время (количество тактов отсчета, соответствующее регистру DTR1) будет вставлено в генерируемую форму ШИМ.

Когда функция захвата входа включена, счетчик запускает или прекращает подсчет, когда он запускается, и регистр ICR1 регистрирует значение счета во время периода срабатывания сигнала захвата. Программное обеспечение может отключить вывод осциллографа OC1A/OC1B, очистит биты COM1A/COM1B или установит их в соответствующий источник запуска.

Когда происходит событие триггера, аппаратное обеспечение автоматически очищает бит COM1A/COM1B, чтобы отключить вывод сигнала OC1A/OC1B.

Счетчик тактов может быть сгенерирован внутренним или внешним источником синхронизации. Выбор источника синхронизации и выбор частотного деления контролируются битами CS1, расположенными в регистре TCCR1B. Подробнее см. Разделы предварительного делителя TC0 и TC1.

Счетчик имеет длину 16 бит и поддерживает двунаправленный подсчет. Режим генерации сигнала, то есть режим работы счетчика, управляется битами WGM1, расположенными в регистрах TCCR1A и TCCR1B.

В соответствии с различными режимами работы счетчик осуществляет очистку, увеличение или уменьшение каждого из счетчиков тактов Clkt1.

Когда счетчик переполняется, устанавливается флаг переполнения счетчика TOV1 в регистре TIFR1.

Прерывание переполнения счетчика TC1 может быть сгенерировано, когда прерывание включено.

Модуль сравнения результатов сравнивает значение счетчика TCNT1 с регистрами сравнения OCR1A и OCR1B вывода. Когда TCNT1 равно OCR1A или OCR1B, происходит совпадение и устанавливается флаг сравнения OCF1A или OCF1B вывода в регистре TIFR1.

После прерывания может быть сгенерировано прерывание сравнения выходного сигнала TC1.

Следует отметить, что регистры OCR1A и OCR1B имеют двойную буферизацию в режиме ШИМ. Двойная буферизация отключена в обычном режиме и режиме СТС.

Когда счетчик достигает максимального или минимального значения, значение в регистре буфера обновляется синхронно с регистрами сравнения OCR1A и OCR1B. Подробнее см. Описание раздела режима работы.

Генератор сигналов генерирует выходные сигналы OC1A и OC1B на выводы, используя сравнения и переполнение счетчика в соответствии с режимом генерации сигнала и контролем режима вывода.

Конкретный метод генерации описывается в описании рабочего режима и описания раздела. Для вывода выходных сигналов OC1A и OC1B сигнала сравнения на соответствующие выводы, регистр направления данных этого вывода также должен быть установлен как выходной сигнал.

## 12.2 Режимы работы

Таймер 1 имеет шесть различных режимов работы: Normal, Clear Compare (CTC), Быстродействующую широтно-импульсную модуляцию (FPWM), Фазокорректированную широтно-импульсную модуляцию (PCPWM), Фазу Частотно-модулированная широтно-импульсная модуляция (PFCPWM) и режим захвата ввода (ICP).

Режим выбирается битами управления режимом генерации волны WGM1[3:0].

Ниже описаны эти шесть режимов.

Поскольку существуют два независимых блока сравнения вывода, обозначенные соответственно «А» и «В», нижний выходной «х» используется для представления каналов двух блоков сравнения вывода.

### 12.2.1. Нормальный режим

Обычный режим - это самый простой режим работы счетчика таймера. В это время бит WGM1 [3: 0] = 0, а максимальное значение TOP счетчика - MAX (0xFFFF).

В этом режиме счетчик увеличивается на один для каждого счетчика тактов. Когда счетчик достигает переполнения TOP, он возвращается в BOTTOM, чтобы снова начать накапливать.

Флаг переполнения счетчика таймера TOV1 устанавливается в том же такте счета счетчика TCNT1, значение которого становится равным нулю.

Флаг TOV1 в этом режиме похож на 17-й счет, он будет установлен и не очищается счетом.

Процедура обслуживания прерывания переполнения автоматически очищает

флаг TOV1 и может использоваться программным обеспечением для увеличения разрешения счетчика таймера.

В нормальном режиме нет специального случая, и новые значения счета могут быть записаны в любое время. Форма сигнала выходного сигнала сравнения OC1x может быть получена только тогда, когда регистр направления данных вывода OC1x задан как выход. Когда COM1x = 1, сигнал OC1x переключается, когда происходит совпадение сравнения.

В этом случае частоту формы волны можно вычислить по следующей формуле:

$$f_{oc1xnormal} = f_{sys} / (2 * N * 65536)$$

где N - обозначает Коэффициент предделителя (1, 8, 64, 256 или 1024).

Модуль сравнения результатов может использоваться для генерации прерывания, но прерывания не рекомендуются в нормальном режиме, который потребляет слишком много времени процессора.

## 12.2.2 Режим СТС

Когда установлено WGM1 [3:0] = 4 или 12, счетчик/таймер 1 переходит в режим СТС.

Когда WGM1 [3:0] = 0, максимальное значение TOP - это OCR1A. Когда WGM1 [3] = 1, максимальное значение счета TOP является ICR1.

Ниже описывается режим СТС с использованием WGM1 [3: 0] = 4. В этом режиме режим подсчета увеличивается для каждого счета. Когда значение счетчика TCNT1 равно TOP, счетчик очищается. Этот режим позволяет пользователю легко контролировать частоту выходного сигнала сравнения, а также упрощает операцию подсчета внешних событий. Когда счетчик достигает TOP, устанавливается флаг соответствия OCF1 для сравнения на выходе, и прерывание будет генерироваться, когда установлен соответствующий бит разрешения прерывания.

Регистр OCR1A может быть обновлен в процедуре обслуживания прерываний. В этом режиме OCR1A не использует двойную буферизацию. Будьте внимательны при обновлении максимального или минимального значения, когда счетчик работает без предварительного делителя или очень малого предварительного делителя.

Если значение, записанное в OCR1A, меньше текущего значения TCNT1, счетчик пропустит сравнение. Перед тем, как произойдет следующее сравнение, счетчик должен сначала пересчитать до MAX, а затем начать отсчет с BOTTOM на OCR1A.

Как и в обычном режиме, флаг TOV1 устанавливается в счетчике, который возвращается к 0x0000.

Форма сигнала выходного сигнала сравнения OC1x может быть получена только тогда, когда регистр направления данных вывода OC1x задан как выход.

Частоту формы волны можно вычислить по следующей формуле:

$$f_{oc1xctc} = f_{sys} / (2 * N * (1 + OCR1A))$$

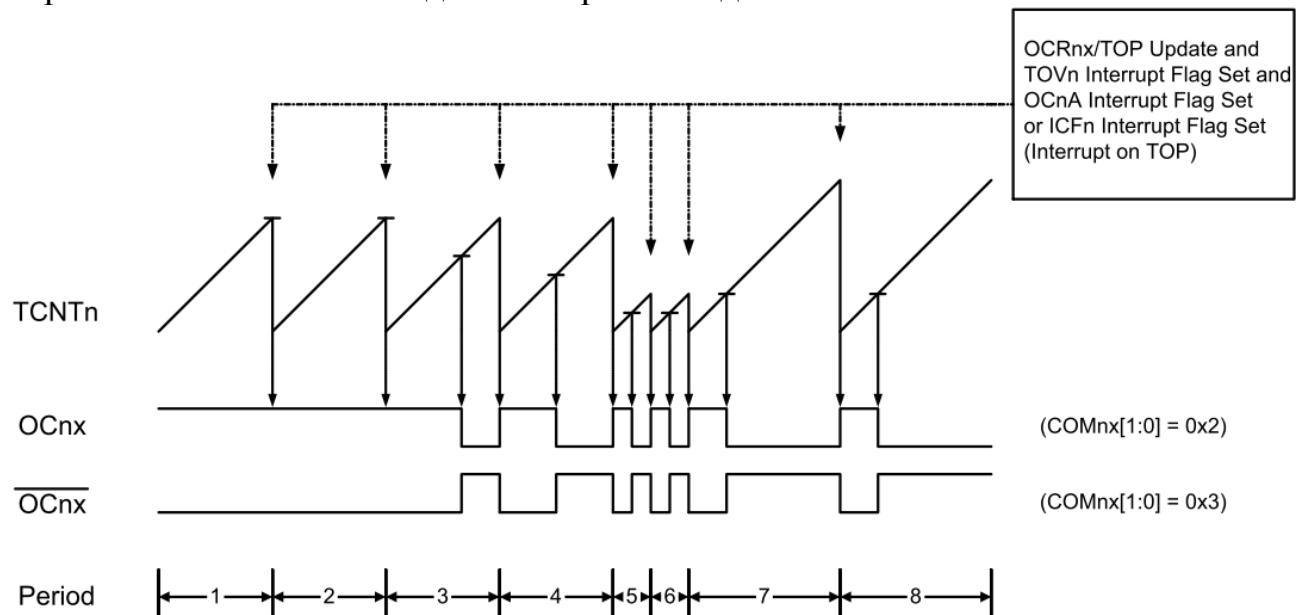
где N - обозначает Коэффициент предделителя (1, 8, 64, 256 или 1024).

Как видно из формулы, когда OCR1A установлено на 0x0000, и нет предварительного делителя, можно получить выходной сигнал с максимальной частотой  $f_{sys} / 2$ .

Когда WGM1 [3: 0] = 12, он аналогичен WGM1 [3: 0] = 4, за исключением того, что соответствующий OCR1A изменен на ICR1.

### 12.2.3. Быстрый режим PWM

Когда установлено WGM1 [3: 0] = 5,6,7,14 или 15, счетчик/таймер 1 переходит в быстрый режим ШИМ, то счетом максимального значения TOP являются 0xFF, 0x1FF, 0x3FF, ICR1 или OCR1A, он может быть использован для получения Высокочастотный ШИМ-сигнал. Разница между быстрым режимом PWM и другими режимами PWM заключается в том, что это односторонняя операция. Счетчик считает до TOP от минимального значения 0x0000, а затем сбрасывается в BOTTOM для повторного подсчета.



Когда значение счетчика достигает TCNT1 верхней или нижней, OC1x выдает сигнал сравнения устанавливается или сбрасывается, в зависимости от выходного сравнения режим установки COM1, а подробное описание регистра. Так как операция односторонняя, рабочая частота быстрого режима PWM в два раза больше двунаправленной операции режима с коррекции PWM фазы.

Высокочастотные характеристики делают быстрый режим PWM подходящим для регулирования мощности, выпрямления и применения ЦАП. Высокочастотные сигналы могут уменьшить размер внешних компонентов (индуктивность, емкость и т. д.), тем самым снижая стоимость системы.

Когда значение счетчика достигает вершины, TOV1 флаг переполнения счетчика/таймера будет установлен, и обновлено значение буфера значение для сравнения на компаратор.

Если прерывание включено, регистр OCR1A может быть обновлен в процедуре обслуживания прерываний.

Форма сигнала выходного сигнала сравнения OC1x может быть получена только тогда, когда регистр направления данных вывода OC1x задан как выход.

Частоту формы волны можно вычислить по следующей формуле:

$$f_{oc1xfpwm} = f_{sys} / (N * (1 + TOP))$$

где N - обозначает Коэффициент предделителя (1, 8, 64, 256 или 1024).

Когда происходит сравнение равенства между TCNT1 и OCR1x, генератор формы сигнала устанавливает (очищает) сигнал OC1x, когда TCNT1 очищается.

В нуле генератор формы сигнала очищает (устанавливает) сигнал OC1x для генерации волны ШИМ. Таким образом, крайности OCR1x будут генерировать специальные формы ШИМ. Когда OCR1x установлен в 0x0000, выходной PWM - это каждый (1 + TOP) счетчик тактов существует узкий выброс. Когда OCR1x установлен в TOP, выходной сигнал постоянно высокий или низкий. Такие, как

Если OCR1A используется как TOP и COM1A = 1, выходной сигнал сравнения OC1A будет генерировать ШИМ с 50% рабочим циклом ШИМ-волны.

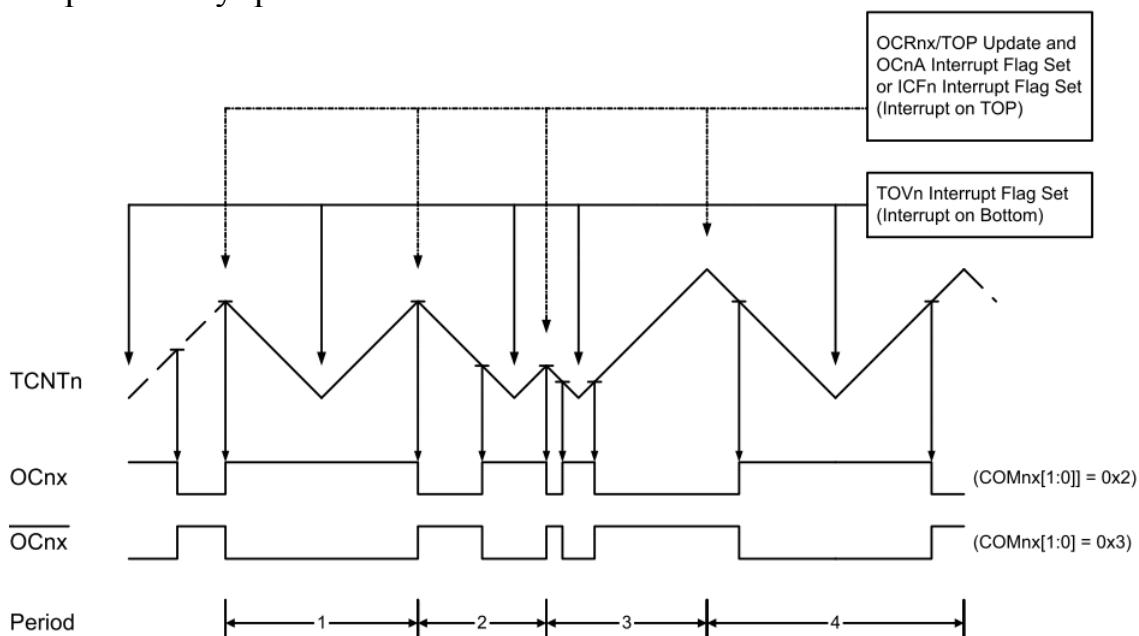
#### 12.2.4. Режим PWM с коррекцией фаз

При установке WGM0 [3: 0] = 1,2,3,10 или 11 счетчик 1 таймера входит в режим PWM с фазовой коррекцией.

Максимальное значение числа TOP составляет 0xFF, 0x1FF, 0x3FF, ICR1 или OCR1A, соответственно.

Счетчик работает в обоих направлениях, увеличиваясь от BOTTOM до TOP, затем вниз до BOTTOM и повторяя эту операцию. Когда счетчик достигает TOP или BOTTOM, изменяет направление счета, а значение счетчика остается только на одном счете тактов TOP или BOTTOM. Значение подсчета при приращении или уменьшении, когда TCNT1 соответствует OCR1x, выходной сигнал сравнения OC1x будет очищен или установлен, в зависимости от режима вывода сравнения COM1 настройки.

По сравнению с односторонней работой максимальная частота, доступная для двунаправленной операции, меньше, но ее отличная симметрия более подходит для электричества управление машиной.



В режиме Фазовой коррекции PWM флаг TOV1 устанавливается, когда счетчик достигает BOTTOM и сравнивается, когда счетчик достигает TOP.

Значение буфера обновляется до значения сравнения.

Если прерывание включено, буфер сравнения OCR1x может быть обновлен в процедуре обслуживания прерываний устройства.

Выходной сигнал OC1x сигнала сравнения может быть получен только тогда, когда регистр направления данных вывода OC1x задан как выход.

Частоту формы волны можно вычислить по следующей формуле:

$$f_{oc1xcpwm} = f_{sys} / (N * TOP * 2)$$

где N - обозначает Коэффициент предделителя (1, 8, 64, 256 или 1024).

Во время отсчета генератор сигналов очищает (устанавливает) сигнал OC1x, когда TCNT1 соответствует OCR1x. Во время подсчета, генератор сигналов устанавливает (очищает) сигнал OC1x, когда TCNT1 соответствует OCR1x.

Из-за этого экстремальное значение OCR1x генерирует специальную волну PWM. Когда OCR1x установлен в TOP или BOTTOM, выходной сигнал OC1x остается низким или высоким.

Если OCR1A используется в качестве TOP и COM1A = 1, выходной сигнал сравнения OC1A будет генерировать PWM-сигнал с 50% -ным рабочим циклом. Чтобы обеспечить симметрию выходной ШИМ-волны с обеих сторон BOTTOM, сигнал OC1x будет перевернут в двух случаях, когда совпадение сравнений не произойдет.

Первый случай - когда значение OCR1x изменяется с TOP на другие данные. Когда OCR1x является TOP, а значение счета достигает TOP, выход OC1x совпадает с тем, когда совпадает предыдущий счетчик убывания, то есть OC1x остается неизменным. Обновленное значение нового значения OCR1x (не TOP) обновляется в это время, а значение OC1x удерживается до тех пор, пока совпадение сравнения не произойдет, когда счетчик увеличивается. В этом случае сигнал OC1x не является симметричным относительно минимального значения, поэтому сигнал OC1x необходимо переключать, когда TCNT1 достигает своего максимального значения. То есть первый случай, когда сигнал OC1x переключается во время сравнения, не возникает.

Во втором случае, когда TCNT1 начинает отсчет с более высокого значения, чем OCR1x, сравнение сравнивается с потерями, что вызывает асимметричную ситуацию. Также необходимо перевернуть сигнал OC1x для достижения симметрии по обе стороны от минимального значения.

## 12.2.5. Режим PWM с коррекцией частоты и фазы

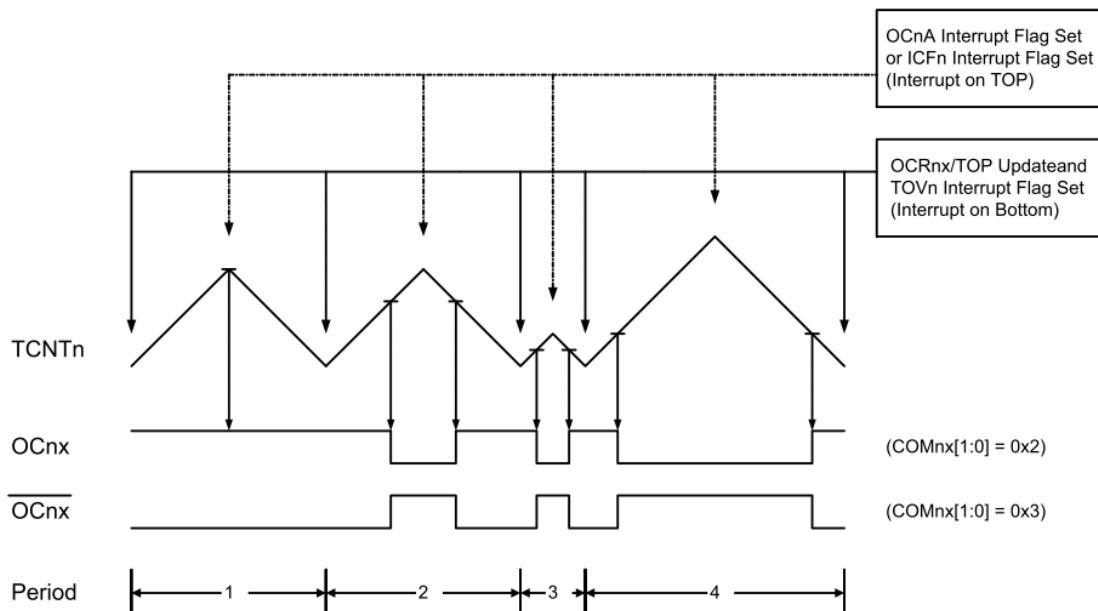
Когда установлено значение WGM0 [3: 0] = 8 или 9, счетчик 1 таймера входит в режим PWM с коррекцией фазовой частоты, а максимальное значение счета TOP - это ICR1 или OCR1A соответственно.

Счетчик работает в обоих направлениях, увеличиваясь от BOTTOM до TOP, затем вниз до BOTTOM и повторяя эту операцию. Когда счетчик достигает TOP или BOTTOM, изменяет направление счета, а значение счетчика остается только на одном счете тактов TOP или BOTTOM.

Во время процесса приращения или уменьшения, когда значение счетчика TCNT1 совпадает с OCR1x, выходной сигнал сравнения OC1x будет очищен или задан в зависимости от установки режима вывода COM1 сравнения. По сравнению с односторонней работой максимальная частота, получаемая в двунаправленной операции, меньше, но ее превосходная симметрия более подходит для управления двигателем.

Коррекция фазовой частоты В режиме PWM флаг TOV1 устанавливается, когда счетчик достигает BOTTOM, а значение буфера сравнения обновляется до значения сравнения.

Время, когда значение сравнения обновляется, является самой большой разницей между PWM-режимом фазовой коррекции частоты и режимом PWM с фазовой коррекцией.



Если прерывание включено, регистр OCR1x буфера сравнения может быть обновлен в подпрограмме обслуживания прерываний. Когда CPU изменяет значение TOP, то есть значение OCR1A или ICR1, оно должно гарантировать, что новое значение TOP не меньше значения TOP, которое уже используется, иначе сравнение не повторится. Сигнал OC1x сигнала сравнения выходного сигнала может быть получен только в том случае, когда в качестве выходного сигнала установлен регистр направления данных OC1x.

Частоту формы волны можно вычислить по следующей формуле:

$$f_{oc1xcpfpwm} = f_{sys} / (N * TOP * 2)$$

где N - обозначает Коэффициент предделителя (1, 8, 64, 256 или 1024).

Во время отсчета генератор сигналов очищает (устанавливает) сигнал OC1x, когда TCNT1 соответствует OCR1x. Во время подсчета, генератор сигналов устанавливает (очищает) сигнал OC1x, когда TCNT1 соответствует OCR1x.

Из-за этого экстремальное значение OCR1x генерирует специальную волну PWM. Когда OCR1x установлен в TOP или BOTTOM, выходной сигнал OC1x остается низким или высоким. Если OCR1A используется в качестве TOP и COM1A = 1, выходной сигнал сравнения OC1A будет генерировать PWM-сигнал с 50% -ным рабочим циклом.

Регистр OCR1x обновляется во время BOTTOM, длины счета восходящей и нисходящей сторон одинаковы, (при постоянных значениях TOP) приводят к симметричным сигналам с правильной частотой и фазой.

При использовании фиксированного значения TOP лучше всего использовать регистр ICR1, как значение TOP, то есть установить WGM1 [3: 0] = 8. В это время регистр OCR1A используется только для генерации выходного сигнала PWM.

Если вы хотите генерировать частотно-изменяющуюся PWM-волну, вы должны изменить значение TOP. Для этого приложения будут более подходящими характеристики двойной буферизации OCR1A.

#### 12.2.6. Режим ввода данных

Входной захват используется для захвата внешних событий и предоставления им отметки времени, указывающей, когда произошло событие. Это можно сделать в предыдущем режиме подсчета, за исключением того, что режим генерации сигналов с использованием значения ICR1 в качестве значения TOP.

Триггерный сигнал для внешнего события вводится на вывод ICP1 и может также реализовываться аналоговым компаратором. Когда изменяется логический уровень на выводе ICP1 или изменяется уровень АСО аналогового компаратора, и это изменение уровня захватывается блоком захвата ввода, запускается входной захват и 16-разрядное значение счета TCNT1. Данные копируются в регистр захвата ввода ICR1 и устанавливается флаг ICF1 ввода.

Если бит ICIE1 установлен в «1», флаг захвата ввода генерирует прерывание ввода. Источник запуска ICP1 или АСО захвата входа выбирается путем установки бита управления захватом аналогового сравнения ACIC аналогового сравнения и регистра ACSR состояния. Следует отметить, что изменение источника триггера может привести к захвату ввода, поэтому после изменения источника триггера для ICF1 необходимо выполнить четкую операцию, чтобы избежать ошибочных результатов.

Сигнал захвата входного сигнала проходит через дополнительный шумоподавитель и отправляется к детектору кромок в соответствии с конфигурацией бита управления выбора входа захвата ICES1, чтобы убедиться, что обнаруженный фронт удовлетворяет условию триггера. Шумоподавитель - это простой цифровой фильтр, который четыре раза проверяет входной сигнал и выводит на выход детектора, только когда четыре образца равны.

Шумоподавитель управляет битом ICNC1 регистра TCCR1B, чтобы включить или отключить его.

При использовании функции захвата ввода значение регистра ICR1 следует читать как можно раньше после того, как установлен ICF1, потому что значение ICR1 будет обновляться после следующего события захвата.

Рекомендуется включить прерывание захвата ввода. В любом режиме захвата ввода не рекомендуется изменять значение TOP во время работы. Введенная метка времени может использоваться для расчета частоты, рабочего цикла и других характеристик сигнала и создания журнала для события триггера. Измерение рабочего цикла внешнего сигнала требует, чтобы фронт триггера был изменен после каждого захвата. Следовательно, фронт сигнала запуска должен быть как можно быстрее изменен после считывания значения ICR1.

### 12.2.7. Управление мертвым временем

Когда бит DTEN1 установлен на «1», функция включения мертвого времени активирована. Выходные сигналы OC1A и OC1B будут вставлены в заданное мертвое время на основе формы сигнала, создаваемого выходным сигналом сравнения B-канала. Длина - это значение времени, соответствующее числу счетчиков в регистре DTR1. Как показано на рисунке ниже, вставки мертвого времени OC1A и OC1B основаны на выходной форме сигнала канала B. Когда COM1A и COM1B являются «2» или «3», полярность сигнала OC1A такая же, как полярность формы сигнала OC1B. Когда COM1A и COM1B «2» или «3» соответственно, форма волны OC1A и форма волны OC1B одинаковы. Полярность противоположна.

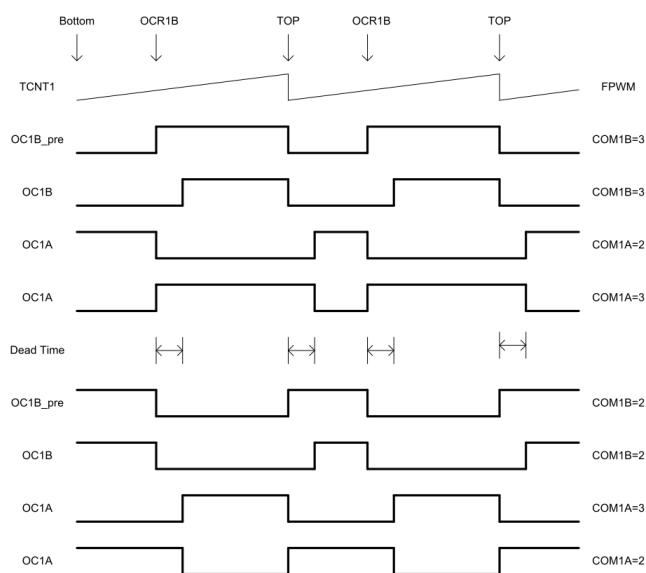


Рис.12. Контроль мертвого времени TC1 в режиме FPWM

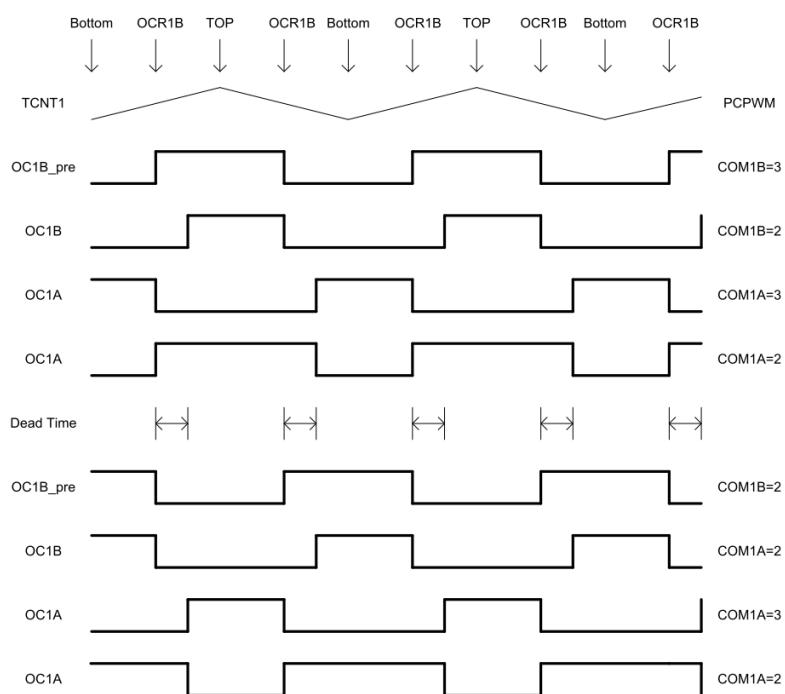


Рис. 12. Тайм-тайм TC1 в режиме PCPWM

Когда бит DTEN1 установлен на «0», функция вставки мертвого времени отключена. Выходные сигналы OC1A и OC1B являются формами сигналов, генерируемыми соответствующими результатами сравнения.

#### 12.2.8. Высокоскоростной режим счета

В режиме высокоскоростной синхронизации в качестве источника синхронизации для счетчика используются высокочастотные такты для генерации высокочастотных и высокочастотных сигналов PWM. Этот высокочастотный тактовый генератор генерируется путем умножения выходного тактового сигнала rc32m внутреннего 32M RC-генератора в 2 раза. Поэтому перед входом в высокочастотный режим необходимо включить функцию частотного умножения внутреннего 32M RC-генератора, то есть установить бит F2XEN в регистре TCKCSR и дождаться определенного времени до тех пор, пока выходной сигнал умноженного тактового сигнала не станет стабильным. Затем бит TC2XS1 в TCKCSR может быть установлен для установки счетчика таймера в режиме высокоскоростной синхронизации. В этом режиме системные тактовые импульсы и высокоскоростные такты являются асинхронными, а некоторые регистры (см. Список регистров TC1) работают в высокоскоростном домене так, что конфигурация и считывание этих регистров также асинхронны, и при работе необходимо соблюдать осторожность.

Нет особых требований к несинхронным операциям чтения и записи в регистрах в высокоскоростном домене синхронизации.

При выполнении последовательных чтений и записи подождите системные такты.

Выполните следующие шаги:

- 5) Запишите регистр A;
- 6) Подождите (NOP или зарегистрироваться под операционными системами);
- 7) Чтение или запись регистров A или B.
- 8) Подождите (NOP или зарегистрироваться под операционными системами);

При чтении регистра в высокоскоростной тактовом домене можно считывать регистры с шириной 8 бит и 16 бит.

Значение регистра (OCR1A, OCR1B, ICR1, TCNT1) сначала считывает значение младшего регистра. После ожидания одного системного такта считывается значение высокого регистра.

При чтении значения TCNT1, когда счетчик все еще подсчитывает, значение TCNT1 изменяется с помощью высокоскоростного тактового сигнала, а счетчик может быть приостановлен (установите CS1 на ноль). Снова прочтайте значение TCNT1.

Чтобы прочитать OCR1A, OCR1B и ICR1, выполните следующие действия:

- 1) Чтение OCR1AL / OCR1BL / ICR1L;
- 2) Ожидание системных часов (NOP);
- 3) Прочтите OCR1AH / OCR1BH / ICR1H.

При чтении TCNT1 выполните следующие действия:

- 1) Установите CS1 в ноль;
  - 2) Ожидание системных часов (NOP);
  - 3) Считать значение TCNT1L;
  - 4) Ожидание системных часов (NOP);
  - 5) Прочтите значение TCNT1H.

## 12.2.9. Регистры Описания

## *Список регистров TCI*

Регистр	Адрес	Значение по умолчанию	Описание
TCCR1A *	0x80	0x00	Регистр управления А
TCCR1B *	0x81	0x00	Регистр управления В
TCCR1C *	0x82	0x00	Регистр управления С
DSX1	0x83	0x00	Регистр управления источником триггера
TCNT1L *	0x84	0x00	Счетчик значений - младший байт
TCNT1H *	0x85	0x00	Счетчик значений - старший байт
ICR1L *	0x86	0x00	Регистр захвата - младший байт
ICR1H *	0x87	0x00	Регистр захвата - старший байт
OCR1AL *	0x88	0x00	Регистр сравнения А выходных данных - младший байт
OCR1AH *	0x89	0x00	Регистр сравнения А выходных данных - старший байт
OCR1BL *	0x8A	0x00	Регистр сравнения В выходных данных - младший байт
OCR1BH *	0x8B	0x00	Регистр сравнения В выходных данных - старший байт
DTR1L *	0x8C	0x00	Регистр контроля мертвого времени - младший байт
DTR1H*	0x8D	0x00	Регистр контроля мертвого времени - старший байт
TIMSK1	0x6F	0x00	Регистр маски прерываний счетчика/таймера
TIFR1	0x36	0x00	Регистр флагов прерываний счетчика/таймера
TCKCSR1	0xEC	0x00	Регистр состояния и контроля счетчика/таймера

**Примечание:** Регистры с «\*» работают в системных и высокоскоростных тактовых доменах. Регистры без «\*» работают только в домене системных тактов.

## *TCCR1A - Регистр управления A*

## *СОМ1х - Режим Управление сравнением без PWM*

СОМ1x [1:0]	Описание
0	OC1x отключен, операция ввода-вывода общего назначения
1	Переключение сигнала OC1x во время сравнения
2	Очистить сигнал OC1x во время сравнения
3	Установите сигнал OC1x при сравнении

СОМ1х - Быстрый Режим PWM Управление сравнением

COM1x [1:0]	Описание
0	OC1x отключен, операция ввода-вывода общего назначения
1	Когда WGM1 равен 15: сигнал OC1A переключается во время сравнения, а OC1B выключается. Когда WGM1 является другим значением: OC1x отключен, общая операция ввода-вывода
2	Очистка сигнала OC1x во время сравнения и установка сигнал OC1x, когда произойдет максимальное совпадение
3	Установка сигнала OC1x во время сравнения и очистка сигнала OC1x, когда произойдет максимальное совпадение

COM1x - Режим PWM с коррекцией фаз Управление сравнением

COM1x [1:0]	Описание
0	OC1x отключен, операция ввода-вывода общего назначения
1	Когда WGM1 равен 9 или 11: сигнал OC1A переключается во время сравнения, OC1B выключается Когда WGM1 является другим значением: OC1x отключен, общая операция ввода-вывода
2	Очистка OC1x при совпадении, когда счет вверх. Установка OC1x при совпадении, когда счет вниз.
3	Установка OC1x при совпадении, когда счет вверх. Очистка OC1x при совпадении, когда счет вниз.

## **TCCR1B - Регистр управления B**

5	-	Зарезервировано
4	WGM13	WGM13, WGM12, WGM11, WGM10 вместе образуют режим WGM1 управления генерацией формы волны [3: 0] для управления режимом подсчета счетчиков и режимом генерации формы сигнала. Подробнее см. В таблице Описания режима генерации волны.
3	WGM12	
2	CS12	Pредделение тактов источника синхронизации для счетчика времени 0.
		<b>CS1[2:0]</b>
		<b>Описание</b>
		0 Нет источника синхронизации, стоп-счет
		1 clksys
		2 clksys / 8 из предделителя
		3 clksys / 64 из предделителя
		4 clksys / 256 из предделителя
1	CS11	5 clksys / 1024 от предделителя
		6 Внешние такты T1, триггер со спадающим фронтом
0	CS10	7 Внешние такты T1, триггер с нарастающим фронтом

### WGM1 - Управление режимом генерации сигнала.

WGM1[3:0]	Режим работы	Значение TOP	Обновить момент OCR1	Установите момент TOV1
0	Normal	0xFFFF	Немедленно	MAX
1	8-битный PCPWM	0x00FF	TOP	BOTTOM
2	9-битный PCPWM	0x01FF	TOP	BOTTOM
3	10-битный PCPWM	0x03FF	TOP	BOTTOM
4	CTC	OCR1A	Немедленно	MAX
5	8-битный PCPWM	0x00FF	BOTTOM	TOP
6	9-битный PCPWM	0x01FF	BOTTOM	TOP
7	10-битный PCPWM	0x03FF	BOTTOM	TOP
8	PFCPWM	ICR1	BOTTOM	BOTTOM
9	PFCPWM	OCR1A	BOTTOM	BOTTOM
10	PCPWM	ICR1	TOP	BOTTOM
11	PCPWM	0CR1A	TOP	BOTTOM
12	CTC	ICR1	Немедленно	MAX
13	Удержание	-	-	-
14	FPWM	ICR1	TOP	TOP
15	FPWM	0CR1A	TOP	TOP

### TCCR1C - Регистр управления С

TCCR1C - Регистр управления С								
Адрес: 0x82				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	FOC1A	FOC1B	DOC1B	DOC1A	DTEN1	-	-	-
R/W	W	W	R/W	R/W	R/W	-	-	-
Бит	Имя	Определение бит						
7	FOC1A	Бит FOC1A активный только, когда биты WGM определяют режим без PWM.						
		COM1A1	COM1A0	Описание				
		0	0	Нормальных операций порта, OC1A отключено.				

		<table border="1"> <tr><td>0</td><td>1</td><td>Бистабильная схема OC1A на Сравнении Компаратора.</td></tr> <tr><td>1</td><td>0</td><td>Очистить OC1A на Сравнении Компаратора.</td></tr> <tr><td>1</td><td>1</td><td>Установить OC1A на Сравнении Компаратора.</td></tr> </table> <p>Чтение FOC1A всегда возвращает ноль.</p>	0	1	Бистабильная схема OC1A на Сравнении Компаратора.	1	0	Очистить OC1A на Сравнении Компаратора.	1	1	Установить OC1A на Сравнении Компаратора.						
0	1	Бистабильная схема OC1A на Сравнении Компаратора.															
1	0	Очистить OC1A на Сравнении Компаратора.															
1	1	Установить OC1A на Сравнении Компаратора.															
6	FOC1B	<p>Бит FOC1B активный только, когда биты WGM определяют режим без PWM.</p> <table border="1"> <thead> <tr> <th>COM1B1</th><th>COM1B0</th><th>Описание</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>Нормальных операций порта, OC1B отключено.</td></tr> <tr><td>0</td><td>1</td><td>Бистабильная схема OC1B на Сравнении Компаратора.</td></tr> <tr><td>1</td><td>0</td><td>Очистить OC1B на Сравнении Компаратора.</td></tr> <tr><td>1</td><td>1</td><td>Установить OC1B на Сравнении Компаратора.</td></tr> </tbody> </table> <p>Чтение FOC1B всегда возвращает ноль.</p>	COM1B1	COM1B0	Описание	0	0	Нормальных операций порта, OC1B отключено.	0	1	Бистабильная схема OC1B на Сравнении Компаратора.	1	0	Очистить OC1B на Сравнении Компаратора.	1	1	Установить OC1B на Сравнении Компаратора.
COM1B1	COM1B0	Описание															
0	0	Нормальных операций порта, OC1B отключено.															
0	1	Бистабильная схема OC1B на Сравнении Компаратора.															
1	0	Очистить OC1B на Сравнении Компаратора.															
1	1	Установить OC1B на Сравнении Компаратора.															
5	DOC1B	Когда бит DOC1B установлен на «1», активируется сигнал сравнения выключения источника запуска OC1B. Когда происходит событие триггера, аппаратное обеспечение автоматически отключает вывод сигнала OC1B. Когда бит DOC1B установлен на «0», выходной сигнал сравнения выключения источника OC1B отключен. Выход сигнала OC1B не отключается при возникновении события триггера.															
4	DOC1A	Когда бит DOC1A установлен на «1», активируется сигнал сравнения выключения источника запуска OC1A. Когда происходит событие триггера, аппаратное обеспечение автоматически отключает вывод сигнала OC1A. Когда бит DOC1A установлен на «0», выходной сигнал сравнения выключения источника OC1A отключен. Выход сигнала OC1A не отключается при возникновении события триггера															
3	DTEN1	Если бит DTEN1 установлен на «1» - мертвое время включено. Оба OC1A и OC1B вставляют мертвое время в форму волны, генерируемую выходным сигналом сравнения В-канала, а вставленный мертвый интервал определяется временем отсчета, соответствующим регистру DTR1. Полярность выходного сигнала OC1A определяется соответствием между COM1A и COM1B. См. OC1A для таблицы полярности сигнала после вставки мертвого времени. Когда бит DTEN1 установлен на «0», вставка мертвого времени запрещена. Сигналы OC1A и OC1B являются формами сигналов, генерируемыми соответствующими результатами сравнения.															
2:0	-	Зарезервировано															

В Таблице показано управление полярностью выходного сигнала OC1A, когда включено время отключения.

Контроль полярности формы сигнала OC1A в режиме включения мертвого времени

DTEN1	COM1A [1:0]	COM1B [1:0]	Описание
0	-	-	полярность сигнала OC1A, контролируемая OC1A, сравнивает выходной режим
1	0	-	OC0A отключен, операция ввода-вывода общего назначения
1	1	-	Зарезервировано
1	2	2	Полярности сигналов OC1A и OC1B одинаковы
		3	Полярности сигналов OC1A и OC1B противоположны
1	3	2	Полярности сигналов OC1A и OC1B противоположны
		3	Полярности сигналов OC1A и OC1B одинаковы



*OC1A / OC1B - Отключить управление Выбор источника*

<b>DOC1x</b>	<b>DSX1n = 1</b>	<b>Источник</b>	<b>Описание</b>
0	-	-	Бит DOC0x равен «0», функция выключения источника выходного сигнала отключена.
1	0	Аналоговый компаратор 0	Передний фронт ACIF0 отключает выходной сигнал OC1x
1	1	Аналоговый компаратор 1	Передний фронт ACIF1 отключает выходной сигнал OC1x
1	4	Внешнее прерывание 1	Передний фронт INTF1 отключает выходной сигнал OC1x
1	5	Изменение уровня контакта 1	Передний фронт PCIF1 отключает выходной сигнал OC1x
1	6	Переполнение TC2	Передний фронт TOV2 отключает выходной сигнал OC1x
1	7	Переполнение TC0	Передний фронт TOV0 отключает выходной сигнал OC1x

**Примечание:** 1) Когда DSX1n = 1 указывает, что n-й бит регистра DSX1 равен 1, биты регистра могут быть установлены одновременно.

## **TCNT1L - Счетчик значений - младший байт**

## *TCNT1H - Счетчик значений Старший байт*

*ICR1L - Регистр захвата Младший байт*

*ICR1L - Регистр захвата Старший байт*

*OCR1AL - Регистр сравнения A выходных данных Младший байт*

*OCR1AH - Регистр сравнения A выходных данных Старший байт*

*OCR1BL - Регистр сравнения В выходных данных Младший байт*

*OCR1BH - Регистр сравнения В выходных данных Старший байт*

*TIMSK1 - Регистр маски прерывания счетчика/таймера*

Регистр маски прерываний счетчика/таймера								
Адрес: 0x6F			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	-	-	TICIE1	-	-	OCIE1B	OCIE1A	TOIE1
R/W	-	-	R/W	-	-	R/W	R/W	R/W
Бит	Имя		Определение бит					

7:6	-	Зарезервировано
5	TICIE1	Когда бит ICIE1 установлен на «1» и глобальное прерывание установлено - прерывание ввода TC1 включено. Когда введен триггер захвата, то есть установлен флаг ICF1 TIFR1, происходит прерывание. Когда бит ICIE1 равен «0», прерывание ввода входного сигнала TC1 отключено.
4:3		Зарезервировано
2	OCIE1B	Когда бит OCIE1B, установлен в «1» и глобальное прерывание установлено - прерывание Компаратора B Сравнение разрешено. Соответствующее прерывание выполняется, если Сравнение Компаратора в Таймер/Счетчик происходит, то есть, когда бит OCF1B устанавливается в TIFR1. Когда бит OCIE1B равен «0» выход TC1 прерывания совпадения B - отключен.
1	OCIE1A	Когда бит OCIE0A, установлен в «1» и глобальное прерывание установлено - прерывание Компаратора A Сравнение разрешено. Соответствующее прерывание выполняется, если Сравнение Компаратора в Таймер/Счетчик происходит, то есть, когда бит OCF0A устанавливается в TIFR0. Когда бит OCIE0A равен «0» выход TC0 прерывания совпадения A - отключен.
0	TOIE1	Бит разрешения прерывания переполнения TC0. Когда бит TOIE0 установлен в «1», а глобальное прерывание установлено, прерывание переполнения TC0 включено. Прерывание генерируется при переполнении TC0, т. е. устанавливается бит TOV0 в TIFR. Когда бит TOIE0 равен «0», прерывание переполнения TC0 отключено.

*TIFR1 - Регистр флагов прерываний счетчика/таймера*

0	TOV1	Флаг переполнения TC1. Когда счетчик переполняется, устанавливается флаг переполнения TOV1. Если бит разрешения прерывания переполнения TOIE1 установлен на «1» и установлен флаг глобального прерывания, будет генерироваться прерывание переполнения. TOV1 автоматически очищается, когда выполняется эта процедура прерывания, или ее можно очистить, записав «1» в бит TOV1.
---	------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## *DTR1L - Регистр контроля мертвого времени Младший байт*

*DTR1H - Регистр контроля мертвого времени Младший байт*

## **TCKCSR -TC Регистр состояния и контроля тактов**

0	TC2XS0	При установке бита TC2XS0 в «1» выбор счетчика/таймера 0 для работы в режиме высокоскоростной синхронизации. При установке бита TC2XS0 на «0» выберите счетчик таймера 0 для работы в режиме системных часов.
---	--------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 13. TMR0 / 1/3 Предделитель

- 3 10-разрядных предделителя
- ТC0, ТC1 и ТC3 мультиплексный предделитель CPS310 в мультиплексном режиме
- Независимый режим предварительного делителя ТC0 CPS310, ТC1 CPS1, ТC3 CPS3
- Сброс программного обеспечения поддержки

### 13.1. Обзор

В мультиплексном режиме ( $PSS1 = 0$  и  $PSS3 = 0$ ) ТC0, ТC1 и ТC3 используют 10-разрядный предварительный делитель CPS310, но у них разные настройки разделения.

В одноразовом режиме ( $PSS1 = 1$  и  $PSS3 = 0$ ) ТC1 независимо использует 10-разрядный предварительный делитель CPS1, а ТC0 и ТC3 используют 10-разрядный предварительный делитель CPS310, но имеют разные настройки деления.

В одноразовом режиме ( $PSS1 = 0$  и  $PSS3 = 1$ ) ТC3 использует 10-разрядный предварительный делитель CPS3 независимо, а ТC0 и ТC1 используют 10-разрядный предварительный делитель CPS310, но имеют разные настройки деления.

В автономном режиме ( $PSS1 = 1$  и  $PSS3 = 1$ ) ТC0 независимо использует 10-разрядный предварительный делитель CPS310. ТC1 независимо использует 10-разрядный предделитель CPS1, а ТC3 независимо использует предварительный делитель CPS3.

Ниже следующее описание используется для ТC0, ТC1 и ТC3, где n представляет 0, 1 или 3.

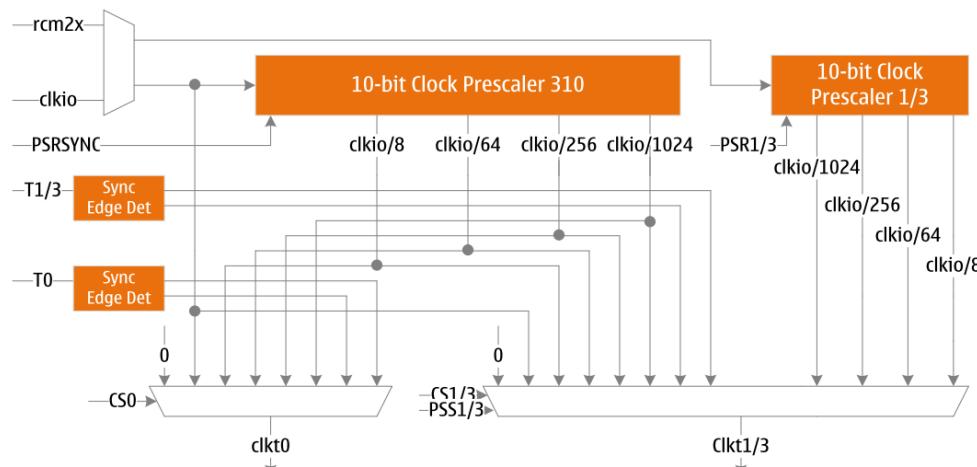


Рис. ТC0 / ТC1 / ТC3 Блок-схема предварительного разделителя

### 13.1.1. Внутренний источник синхронизации

При настройке CSn [2: 0] = 1 таймер 3 может управляться только системными тактами clkio. Счетчик таймера 0 или 1 может быть непосредственно из системного тактового сигнала clkio или высокоскоростного тактового сигнала rcm2x (внутренний выходной сигнал генератора 32M RC на 2 Частотный множитель). Предделитель может выводить четыре разные тактовые частоты: clkio / 8, clkio / 64, clkio / 256 и clkio / 1024.

### 13.1.2. Предделитель Сброс мультиплексированного режима

При установке бита PSS1 в «0», а бит PSS3 - «0», TC0, TC1 и TC3 используют предварительный делитель CPS310. Предделитель работает независимо, и его работа не зависит от логики выбора тактов TC, и она разделяется TC0, TC1 и TC3.

Поскольку управление выбором тактов не влияет, состояние предделителя влияет на применение разделенных тактов. Эффект генерируется, когда таймер включен, а выход предварительного делителя выбран в качестве источника счетчика ( $6 > \text{CSn } [2: 0] > 1$ ). Это может занять от 1 до  $N + 1$  системных тактов с момента включения таймера до первого счета, где  $N$  - коэффициент предварительного масштабирования (8, 64, 256 или 1024). Можно синхронизировать таймер и работу программы, сбросив предварительный делитель. Следует, однако, отметить, что другой таймер использует этот предварительный делитель, и сброс предделителя повлияет на все таймеры, подключенные к нему.

### 13.1.3. Режим однократного использования.

Когда бит PSS1 установлен в «1», TC1 использует предварительный делитель CPS1 независимо, а предварительный делитель сбрасывает бит PSR1. Соответствующие сбросы работают независимо и не влияют на другие предделители.

Когда бит PSS3 установлен в «1», TC3 использует предварительный делитель CPS3 независимо, а сброс предварительного делителя управляется битом PSR3. Соответствующие сбросы работают независимо и не влияют на другие предделители.

Когда бит PSS1 установлен в «1», а бит PSS3 установлен в «1», TC0 независимо использует предварительный делитель CPS310. Сброс предварительного делителя контролируется битом PSRSYNC. TC1 независимо использует предварительный делитель CPS1, а TC3 используется независимо. Предделитель CPS3, каждый сбрасываемый отдельно, не влияет на другие предварительные делители.

#### 13.1.4. Источник внешних часов

Внешний источник синхронизации, обеспечиваемый выводом T0 / T1 / T3, может использоваться в качестве источника тактовых импульсов. Сигнал на выводах T0 / T1 / T3 используется в качестве источника синхросигнала после синхронных логических и фронтов детекторов. Каждый фронт ( $CSn[2:0] = 7$ ) или спадающий фронт ( $CSn[2:0] = 6$ ) будет генерировать импульс счета.

Внешний источник синхронизации не подается в Предделитель.

Из-за наличия схем синхронизации и обнаружения фронтов на выводах, изменение уровня в T0 / T1 / T3 должно быть отложено на 2,5-300 системных часов для обновления счетчика.

Отключение или включение входного тактового сигнала должно занимать, по меньшей мере, один системный тактовый цикл до того, как T0 / T1 / T3 будет стабильным, в противном случае существует возможность генерации ошибочного тактового импульса счетчика.

Для обеспечения правильной выборки ширина внешнего тактового импульса должна быть больше одного тактового цикла системы.

При 50% -ном рабочем цикле внешняя тактовая частота должна быть меньше половины тактовой частоты системы. Из-за разницы в частоте и рабочем цикле системы, вызванных собственной ошибкой генератора, рекомендуется, чтобы максимальная частота внешних часов не превышала  $f_{sys}/2.5$ .

### 13.1.5. Регистры Описания

## *GTCCR - регистр контроля счетчика общего назначения*

## PSSR- Регистр выбора предделителя

GTCCR - регистр контроля счетчика общего назначения								
Адрес: 0xE2				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	PSS1	PSS3	-	-	-	-	PSRASY	PSRSYNC
R/W	R/W	-	-	-	-	-	W	W
Бит	Имя	Определение бит						
7	PSS1	При установке бита PSS1 в «1», TC1 использует только предварительный делитель CPS1. Когда бит PSS1 установлен в «0», преклиратор мультиплексируется. TC0 и TC1 совместно используют Предделитель CPS310. Предделитель CPS1 недействителен и всегда будет сброшен. Если бит PSS3 равен «0» одновременно, TC3 и TC0 и TC1 совместно используют Предделитель CPS310. Предделители CPS1 и CPS3 являются недействительными и будут сбрасываться все время.						
6	PSS3	При установке бита PSS3 на «1», TC3 использует только предварительный делитель CPS3. Если бит PSS3 установлен на «0», преклиратор мультиплексируется. TC0 и TC3 совместно используют Предделитель CPS310. Предделитель CPS3 недействителен и всегда будет сброшен. Если бит PSS1 равен «0» одновременно, TC1 и TC0 и TC3 совместно используют Предделитель CPS310. Предделители CPS1 и CPS3 являются недействительными и будут сбрасываться все время.						
5:2	-	Зарезервировано						
1	PSRASY	Предделитель CPS3 сбрасывает контрольные биты. Бит PSR3 действителен только в одноразовом режиме TC3. Когда бит PSR3 установлен в «1», предварительный делитель CPS3 будет сброшен. Аппарат очистит бит PSR3 после сброса. Если бит PSR3 установлен на «0», установка недействительна. Значение для чтения этого бита всегда будет «0».						
0	PSRSYNC	Предделитель CPS1 сбрасывает бит управления. Бит PSR1 действителен только в одноразовом режиме TC1. Когда бит PSR1 установлен в «1», предварительный делитель CPS1 будет сброшен. Аппарат очистит бит PSR1 после сброса. Если бит PSR1 установлен на «0», установка недействительна. Значение для чтения этого бита всегда будет «0».						

## 14. Таймер / Счетчик 2 (TMR2)

- 8-разрядный счетчик
- Два независимых блока сравнения
- Автоматическая очистка счетчика и автоматическая загрузка при совпадении сравнения
- ШИМ-выход с фазовой коррекцией без импульсов помех
- Генератор частоты
- Счетчик внешних событий
- 10-разрядный предварительный делитель часов
- Переполнение и сравнение совпадений
- Разрешить использование внешнего кристалла RTC с частотой 32,768 кГц

## 14.1. Обзор

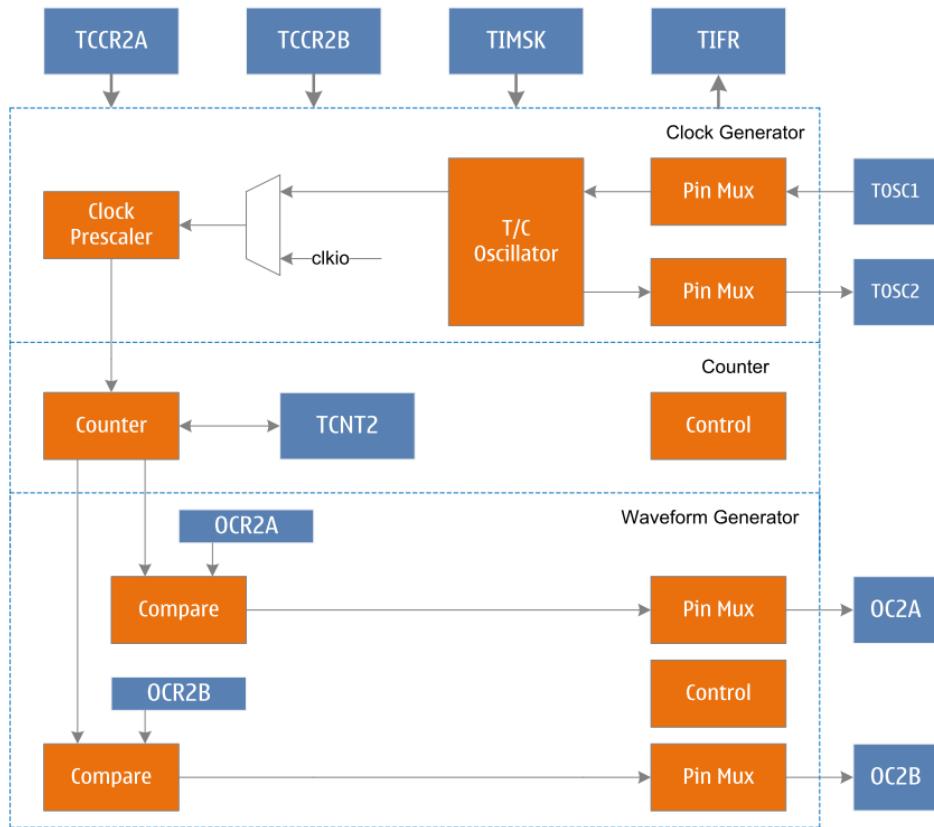


Рис.14.1. Структурная схема TC2

TC2 - это 8-разрядный счетчик счетчиков общего назначения, который поддерживает выход PWM и может точно генерировать формы сигналов.

TC2 содержит 8-разрядный счетчик, блок управления режимом генерации сигнала и два блока сравнения вывода. Блок управления режимом генерации волны управляет режимом работы счетчика и формированием формы сигнала сравнения. В зависимости от режима работы счетчик очищается, увеличивается или уменьшает каждый счет тактов Clkt2. Clkt2 может быть сгенерирован из внутреннего источника синхронизации или внешнего источника синхронизации.

При использовании внешнего кристалла 32,768 кГц TC2 может использоваться как счетчик RTC.

Когда значение счетчика TCNT2 счетчика достигает максимального значения (равное максимальному значению 0xFF или регистровому регистру сравнения OCR2A, определяемому как TOP, максимальное значение определяется как MAX для различия), счетчик очищается или вычитается. Когда значение счетчика счетчиков TCNT2 достигает минимального значения (эквивалентно 0x00, определенное как BOTTOM), счетчик увеличивается. Когда значение счетчика TCNT2 счетчика достигает OCR2A / OCR2B, которое также вызывается, когда происходит сравнение, выходной сигнал сравнения OC2A / OC2B очищается или устанавливается для генерации сигнала ШИМ.

## 14.2. Режимы работы

Счетчик таймера 2 имеет четыре различных режима работы: режим Normal, Compare Compare Clear (CTC), модуляция с быстрой шириной импульсов (FPWM)

и режим фазовой коррекции с широтно-импульсной модуляцией (PCPWM).

Биты управления режимом генерации сигналов WGM2[2: 0] для выбора.

Ниже описаны эти четыре режима. Поскольку существуют два независимых блока сравнения вывода, обозначенные соответственно «A» и «B», нижний выходной «x» используется для представления каналов двух блоков сравнения вывода.

#### 14.2.1. Нормальный режим

Нормальный режим - это самый простой режим работы счетчика таймера.

В это время бит WGM2[2: 0] = 0, а максимальное значение TOP счетчика - MAX (0xFF). В этом режиме счетчик увеличивается на один для каждого такта счета.

Когда счетчик достигает переполнения TOP, он возвращается в BOTTOM, чтобы снова начать накапливать Флаг переполнения счетчика таймера TOV2 устанавливается в том же такте счета счетчика TCNT2, значение которого становится равным нулю.

Флаг TOV2 в этом режиме похож на 9-й бит счетчика, он будет установлен и не будет очищен.

Процедура обслуживания прерывания переполнения автоматически очищает флаг TOV2, и программное обеспечение может использовать его для увеличения разрешения счетчика.

В нормальном режиме нет специального случая, и новые значения счета могут быть записаны в любое время.

Форма сигнала выходного сигнала сравнения OC2x может быть получена только тогда, когда регистр направления данных вывода OC2x задан как выход. Когда COM2x = 1, сигнал OC2x будет переключен при совпадении сравнения. В этом случае частоту формы волны можно вычислить по следующей формуле:

$$f_{OC2x\text{normal}} = f_{\text{sys}} / (2 * N * 256)$$

где N - обозначает Коэффициент предделителя (1, 8, 64, 256 или 1024).

Модуль сравнения результатов может использоваться для генерации прерывания, но прерывания не рекомендуются в нормальном режиме, который потребляет слишком много времени процессора.

#### 14.2.2. Режим СТС

Когда установлено значение WGM0 [2: 0] = 2, счетчик/таймера 2 переходит в режим СТС, а максимальное значение счетчика TOP – OCR2A. В этом режиме счетчик увеличивается на единицу для каждого счетчика часов. Когда значение счетчика TCNT2 равно TOP, счетчик очищается.

OCR2A определяет максимальный счет, который является разрешением счетчика. Этот режим позволяет пользователю легко контролировать частоту выходного сигнала сравнения, а также упрощает операцию подсчета внешних событий. Когда счетчик достигает максимального значения счетчика, устанавливается флаг соответствия OCF2 на выходе, и прерывание будет

генерироваться, когда установлен соответствующий бит разрешения прерывания.

Регистр OCR2A, который является максимальным количеством, может быть обновлен в процедуре обслуживания прерываний. В этом режиме OCR2A не использует двойную буферизацию.

**Внимание:** при обновлении максимального значения OCR2A до минимального значения, когда счетчик работает с предварительным делителем или без него. Если значение, записанное в OCR2A, меньше текущего значения TCNT2, счетчик пропустит равенство сравнения. Перед тем, как произойдет следующее сравнение, счетчик должен просчитывать до TOP, а затем подсчитывать от BOTTOM до OCR2A.

Как и в обычном режиме, значение count возвращается к счету счетчика BOTTOM, чтобы установить флаг TOV2.

Форма сигнала выходного сигнала сравнения OC2x может быть получена только тогда, когда регистр направления данных вывода OC2x задан как выход. Когда COM0x = 1, сигнал OC2x переключается при равенстве сравнения.

В этом случае частоту формы волны можно вычислить по следующей формуле:

$$f_{oc2xctc} = f_{sys} / (2 * N * (1 + OCR2x))$$

где N - обозначает Коэффициент предделителя (1, 8, 64, 256 или 1024).

Из формулы видно, что когда OCR2A задано равным 0x00, и нет предварительного делителя, может быть получен выходной сигнал с максимальной частотой  $f_{sys} / 2$ .

#### 14.2.3. Быстрый режим PWM

Когда установлено значение WGM2[2: 0] = 3 или 7, счетчик 2 таймера входит в режим быстрой PWM и может использоваться для генерации высокочастотных сигналов ШИМ.

Максимальное значение счета TOP равно MAX (0xFF) или OCR2x соответственно. Разница между быстрым режимом PWM и другими режимами PWM заключается в том, что это односторонняя операция. Счетчик считает до TOP от минимального значения 0x00, а затем сбрасывается в BOTTOM для повторного подсчета. Когда значение счета TCNT0 достигает OCR0x или BOTTOM, выходной сигнал сравнения OC0x устанавливается или очищается в зависимости от настройки режима вывода сравнения COM2x.

Подробности см. В описании регистра.

Из-за односторонней работы рабочая частота режима быстрой PWM в два раза выше, чем в режиме PWM с фазовой модификацией, используя двунаправленную операцию.

Высокочастотные характеристики делают быстрый режим PWM подходящим для регулирования мощности, выпрямления и применения ЦАП. Высокочастотные сигналы могут уменьшить размер внешних компонентов (индуктивность, емкость и т. д.). Тем самым снижая стоимость системы.

Когда значение счета достигает максимального значения, устанавливается флаг переполнения счетчика таймера TOV2 и значение буфера сравнения

обновляется до значения сравнения.

Если прерывание включено, регистр OCR2x буфера сравнения может быть обновлен в процедуре обслуживания прерываний. Форма сигнала выходного сигнала сравнения OC2x может быть получена только тогда, когда регистр направления данных вывода OC0x задан как выход.

Частоту формы волны можно вычислить по следующей формуле:

$$f_{oc2xfpwm} = f_{sys} / (N * (1 + TOP))$$

где N - обозначает Коэффициент предделителя (1, 8, 64, 256 или 1024).

Когда встречается сравнение между TCNT2 и OCR2x, генератор формы сигнала (очищает) сигнал OC2x. Когда TCNT2 очищается, генератор формы волны очищает (устанавливает) сигнал OC2x для генерации волны ШИМ.

В результате экстремальное значение OCR2x создаст специальную форму ШИМ. Когда OCR2x установлен в 0x00, выходной ШИМ имеет узкий всплеск в каждом (1 + TOP) счетчике часов. Когда OCR2x установлен на максимальное значение, выходной сигнал постоянно высокий или низкий.

#### 14.2.4. Режим PWM с коррекцией фаз

Когда установлено WGM2[2: 0] = 1 или 5, счетчик 2 таймера входит в режим PWM с фазовой коррекцией, а максимальное значение счетчика равно MAX (0xFF) или OCR2A соответственно. Счетчик работает с двухточечным шагом от BOTTOM до TOP, а затем вниз до BOTTOM и повторяет эту операцию. Когда счетчик достигает TOP и BOTTOM, оба изменяют направление подсчета, а значение счетчика остается только на одном счете тактов TOP или BOTTOM.

Во время процесса приращения или уменьшения, когда значение счетчика TCNT2 совпадает с OCR2x, выходной сигнал сравнения OC2x будет очищен или установлен в зависимости от настройки режима вывода COM2x.

По сравнению с односторонней работой максимальная частота, доступная для двунаправленной работы, меньше, но ее отличная симметрия более подходит для управления двигателем.

Фазовая коррекция В режиме PWM флаг TOV2 устанавливается, когда счетчик достигает BOTTOM, а буфер сравнения обновляется до значения сравнения, когда счетчик достигает TOP.

Если прерывание включено, регистр OCR2x буфера сравнения может быть обновлен в процедуре обслуживания прерываний. Форма сигнала выходного сигнала сравнения OC2x может быть получена только тогда, когда регистр направления данных вывода OC2x задан как выход.

Частоту формы волны можно вычислить по следующей формуле:

$$f_{oc2xcpwm} = f_{sys} / (N * TOP * 2)$$

где N - коэффициент предварительной шкалы (1, 8, 64, 256 или 1024).

Во время перерасчета генератор сигналов очищает (устанавливает) сигнал OC2x, когда TCNT2 соответствует OCR2x.

Во время подсчета, генератор сигналов устанавливает (очищает) сигнал

OC2x, когда TCNT2 соответствует OCR2x. Из-за этого экстремальное значение OCR2x генерирует специальную волну PWM. Когда OCR2x установлен на максимальное или минимальное значение, выходной сигнал OC2x остается низким или высоким.

Чтобы обеспечить симметричность выходной ШИМ-волны с обеих сторон минимального значения, сигнал OC2x будет перевернут в двух случаях, когда совпадение сравнения не произойдет.

Первый случай - когда значение OCR2x изменяется с максимального значения 0xFF на другие данные. Когда OCR2x является максимальным значением, а значение счета достигает максимального значения, выход OC2x совпадает с результатом сравнения в предыдущем счетчике, т.е. OC2x не изменяется. Значение нового OCR2x (не 0xFF) обновляется в это время. Значение OC2x сохраняется до тех пор, пока не произойдет совпадение сравнения, и произойдет сальто. В этом случае сигнал OC2x не является симметричным относительно минимального значения, поэтому сигнал OC2x необходимо переключать, когда TCNT2 достигает максимального значения. То есть, нет первого случая опрокидывания сигнала OC2x при совпадении сравнения.

Во втором случае, когда TCNT2 начинает отсчет с более высокого значения, чем OCR2x, сравнение сравнивается с потерями, что вызывает асимметричную ситуацию. Также необходимо перевернуть сигнал OC2x для достижения симметрии с обеих сторон минимального значения.

#### 14.2.5. Асинхронный режим работы TC2

Когда бит AS2 в регистре ASSR установлен в «1», TC2 работает в асинхронном режиме, а источник счетчика - от генератора внешнего счетчика таймера. При работе TC2 в асинхронном режиме следует учитывать следующие моменты.

**Переключение.** Переход между синхронным и асинхронным режимов может привести к повреждению данных TCNT2, OCR2A, OCR2B, TCCR2A и TCCR2B. Процедура безопасности выглядит следующим образом:

1. Очистите биты регистра OCIE2A, TOIE2 и OCIE2B, чтобы отключить прерывания TC2.
2. Установите бит AS2 для выбора соответствующего источника синхронизации.
3. Запишите новые данные в регистры TCNT2, OCR2A, TCCR2A, OCR2B и TCCR2B.
4. При переключении в асинхронный режим подождите, пока биты TCN2UB, OCR2AUB, TCR2AUB, OCR2BUB и TCR2BUB не будут очищены.
5. Сбросьте флаг прерывания TC2.
6. Включите прерывания, которые необходимо использовать.

**Генератор.** Для генератора лучше всего использовать резонатор с частотой 32,768 кГц. Частота тактовой частоты системы должна быть более чем в 4 раза выше, чем частота резонатора.

Когда CPU записывает TCNT2, OCR2A, TCCR2A, OCR2B и TCCR2B,

аппаратное обеспечение сначала помещает данные в блокнот и защелкивается в соответствующий регистр после нарастающего фронта двух тактов TOSC1.

Новые операции записи данных не могут выполняться до тех пор, пока данные не будут зафиксированы с блокнота в регистре назначения.

Каждый регистр имеет свой собственный регистр, поэтому запись TCNT2 не мешает написанию OCR2.

Асинхронный регистр регистров состояния используется для проверки, были ли данные записаны в регистр назначения.

- Если TC2 используется как условие пробуждения в режиме ожидания MCU, его нельзя ввести до конца каждого обновления регистра.

Спящий режим, иначе MCU может перейти в спящий режим до того, как настройка TC2 вступит в силу, поэтому TC2 не может разбудить систему. Система.

- Если TC2 используется как условие пробуждения для режима ожидания MCU, необходимо принять меры для повторного входа в спящий режим. Для прерывания логики требуется, чтобы тактовый цикл TOSC1 был сброшен, если время от пробуждения до повторного входа в сон меньше одного

Для тактовых циклов TOSC1 прерывание больше не будет происходить, и устройство не может проснуться. Рекомендуется следующий метод работы:

1. Ведите соответствующие данные в каждый регистр;
2. Подождите, пока ASSR обновит соответствующий бит флага занятости.
3. Войдите в спящий режим.

- Если выбран асинхронный режим работы, генератор TC2 будет продолжать работать, если он не переходит в режим выключения. Пользователи должны обратить внимание на

Время установления этого осциллятора может составлять до 1 секунды.

Поэтому пользователю рекомендуется ждать, по крайней мере, после включения генератора TC2.

Через 1 секунду используйте асинхронный режим работы TC2.

**Проснуться.** Пробуждение в спящем режиме в асинхронном режиме работы: после выполнения условия прерывания процесс пробуждения начинается со следующего запуска таймера. То есть счетчик накапливает, по меньшей мере, еще один такт, прежде чем процессор сможет считывать значение счетчика. После пробуждения MCU выполняет процедуру обслуживания прерываний, а затем запускает программу, следующую за оператором SLEEP.

**Читать.** Чтение TCNT2 в течение короткого периода времени после пробуждения из спящего режима может возвращать неверные данные. Поскольку TCNT2 управляет асинхронным тактовым сигналом TOSC1, считывание TCNT2 должно выполняться через внутренний синхронизированный по системе регистр.

Синхронизация происходит на переднем фронте каждого TOSC1. Когда системные часы активируются после пробуждения из спящего режима, значение TCNT2 считается перед входом в спящий режим и не будет обновляться до следующего восходящего фронта TOSC1.

Фаза TOSC1 полностью непредсказуема, когда она просыпается из спящего режима, но зависит от времени пробуждения. Поэтому рекомендуемая

последовательность для считывания значения TCNT2:

1. Напишите произвольное значение OCR2A или TCCR2A;
2. Подождите, пока не будет очищен соответствующий флаг занятости обновления;
3. Прочтите TCNT2.

**Асинхронный.** В асинхронном режиме для синхронизации флага прерывания требуется 3 системных тактовых цикла плюс 1 таймера.

Счетчик накапливает, по меньшей мере, еще один такт, прежде чем MCU сможет прочитать значение счетчика, которое привело к установке флага прерывания. Изменение выходного сигнала сравнения синхронизируется с часами таймера вместо системных часов.

#### 14.2.6. Предделитель TC2.

Входной тактовый сигнал для предварительного делителя TC2 называется clkt2s. Внутренний системный тактовый сигнал clkio или внешний источник синхронизации TOSC1 выбирается битом AS2 в регистре ASSR. По умолчанию используется подключение к системному такту clkio. Если AS2 установлен, TC2 будет асинхронно управляться TOSC1.

Когда внешний монитор 32,768 кГц подключен к контакту TOSC1 и выходу TOSC2, TC2 может использоваться как счетчик RTC. Не рекомендуется применять внешний тактовый сигнал непосредственно к выходу TOSC1.

На рисунке выше показан предварительный делитель TC2. Как показано на рисунке, возможными параметрами предварительного масштабирования являются: clkt2s / 8, clkt2s / 32, clkt2s / 64, Clkt2s / 128, clkt2s / 256 и clkt2s / 1024. Кроме того, вы можете выбрать clkt2s и 0 (стоп-счет).

Бит в PSR2 в SFIOR сбрасывает предварительным делителем, тем самым позволяя пользователю работать с предсказуемой делителем.

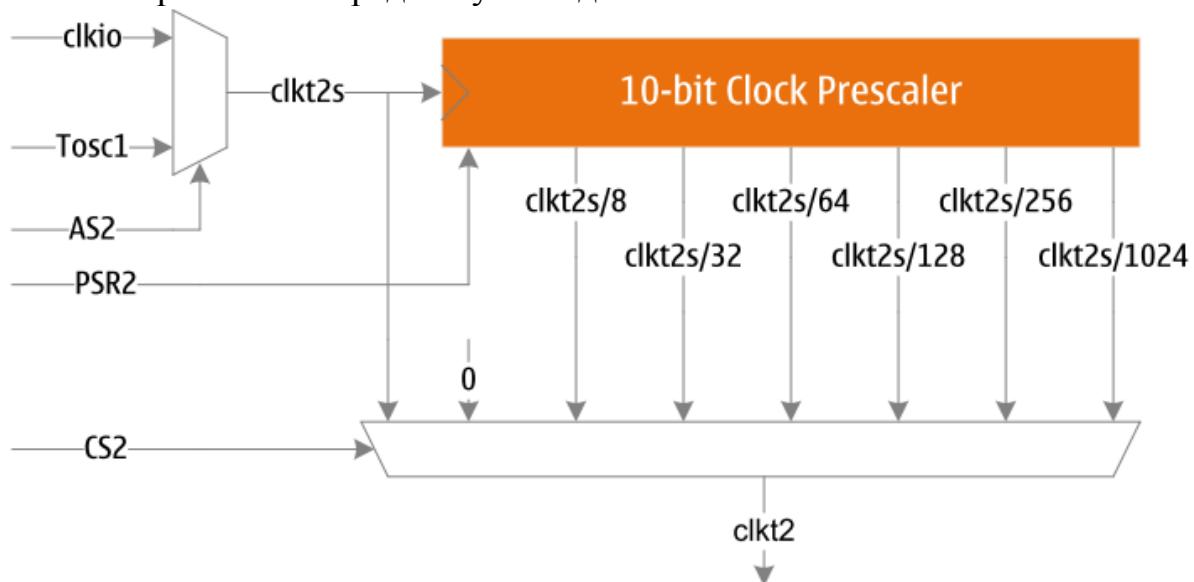


Рис.5 Структура Предделителя TC2

#### 14.2.7. Регистры Описания

## *Список регистров TC2*

Регистр	Адрес	Значение по умолчанию	Описание
TCCR2A	0xB0	0x00	Контрольный регистр TC2 A
TCCR2B	0xB1	0x00	Контрольный регистр TC2 B
TCNT2	0xB2	0x00	Регистр значений счетчика TC2
OCR2A	0xB3	0x00	Регистр сравнения выходных данных TC2 A
OCR2B	0xB4	0x00	Регистр сравнения выходных данных TC2 B
ASSR	0xB6	0x00	Асинхронный регистр состояния TC2
TIMSK2	0x70	0x00	Регистр маски прерывания TC2
TIFR2	0x37	0x00	Регистр флага прерываний TC2

## *TCCR2A - Регистр управления TC2 A*

## *TCCR2B - Регистр управления TC2 B*

		Бит FOC0B активный только, когда биты WGM определяют режим без PWM.																		
		<table border="1"> <thead> <tr> <th>COM2B1</th> <th>COM2B0</th> <th>Описание</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Нормальных операций порта, OC2B отключено.</td> </tr> <tr> <td>0</td> <td>1</td> <td>Бистабильная схема OC2B на Сравнении Компаратора.</td> </tr> <tr> <td>1</td> <td>0</td> <td>Очистить OC2B на Сравнении Компаратора.</td> </tr> <tr> <td>1</td> <td>1</td> <td>Установить OC2B на Сравнении Компаратора.</td> </tr> </tbody> </table>	COM2B1	COM2B0	Описание	0	0	Нормальных операций порта, OC2B отключено.	0	1	Бистабильная схема OC2B на Сравнении Компаратора.	1	0	Очистить OC2B на Сравнении Компаратора.	1	1	Установить OC2B на Сравнении Компаратора.			
COM2B1	COM2B0	Описание																		
0	0	Нормальных операций порта, OC2B отключено.																		
0	1	Бистабильная схема OC2B на Сравнении Компаратора.																		
1	0	Очистить OC2B на Сравнении Компаратора.																		
1	1	Установить OC2B на Сравнении Компаратора.																		
		Чтение FOC2B всегда возвращает ноль.																		
5:4		Резервировано																		
3	WGM02	WGM22,WGM21 и WGM20 вместе образуют режим WGM0 управления генерацией формы волны [2: 0] для управления режимом подсчета счетчиков и режимом генерации формы сигнала. Подробнее см. В описании таблицы режима генерации волны.																		
2	CS22	<table border="1"> <thead> <tr> <th>CS2[2:0]</th> <th>Описание</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Нет источника синхронизации, стоп-счет</td> </tr> <tr> <td>1</td> <td>clksys</td> </tr> <tr> <td>2</td> <td>clksys / 8 из предделителя</td> </tr> <tr> <td>3</td> <td>clksys / 32 из предделителя</td> </tr> <tr> <td>4</td> <td>clksys / 64 из предделителя</td> </tr> <tr> <td>5</td> <td>clksys / 128 из предделителя</td> </tr> <tr> <td>6</td> <td>clksys / 256 из предделителя</td> </tr> <tr> <td>7</td> <td>clksys / 1024 от предделителя</td> </tr> </tbody> </table>	CS2[2:0]	Описание	0	Нет источника синхронизации, стоп-счет	1	clksys	2	clksys / 8 из предделителя	3	clksys / 32 из предделителя	4	clksys / 64 из предделителя	5	clksys / 128 из предделителя	6	clksys / 256 из предделителя	7	clksys / 1024 от предделителя
CS2[2:0]	Описание																			
0	Нет источника синхронизации, стоп-счет																			
1	clksys																			
2	clksys / 8 из предделителя																			
3	clksys / 32 из предделителя																			
4	clksys / 64 из предделителя																			
5	clksys / 128 из предделителя																			
6	clksys / 256 из предделителя																			
7	clksys / 1024 от предделителя																			
1	CS21																			
0	CS20																			

#### COM2n - Быстрый Режим PWM Управление сравнением

COM2x1	COM2x0	Описание
0	0	Нормальных операций порта, OC2A разъединяется.
0	1	Резервировано
1	0	Очистить OC2A на Сравнении, Установить OC2A в основе (неинвертирующий режим)
1	1	Установить OC2A на Сравнении, Очистить OC2A в BOTTOM (инвертирующий режим)

#### COM2n - Режим PWM с коррекцией фаз Управление сравнением

COM2x1	COM2x0	Описание
0	0	Нормальных операций порта, OC2A разъединяется.
0	1	Резервировано.
1	0	Очистить OC2A на Сравнении Компаратора когда Счет Вверх. Установленный OC2A на Сравнении Компаратора когда Счет Вниз.
1	1	Установленный OC2A на Сравнении Компаратора когда Счет Вверх. Очистить OC2A на Сравнении Компаратора когда Счет Вниз.

#### WGM2 - Управление режимом генерации сигнала.

WGM2 [2:0]	Режим работы	Значение TOP	Обновить момент OCR2x	Установите момент TOV2
0	Обычный	0xFF	Немедленно	MAX
1	PCPWM	0xFF	TOP	BOTTOM
2	CTC	OCR0A	Немедленно	MAX
3	FPWM	0xFF	TOP	MAX
4	Резервировано	-	-	-

5	PCPWM	OCR0A	TOP	BOTTOM
6	Резервировано	-	-	-
7	FPWM	OCR0A	TOP	TOP

## *TCNT2 - Регистр счетчиков TC2*

*OCR2A - Регистр сравнения выходных данных ТС2 А*

*OCR2B - Регистр сравнения выходных данных ТС2 В*

которые непрерывно сравниваются со значением счетчика TCNT2. Равенство сравнения может использоваться для генерации прерывания сравнения вывода или для генерации формы сигнала на выводе OC2B. При использовании режима PWM регистр OCR2B использует регистр с двойным буфером. Однако двойная буферизация отключена в нормальном режиме и соответствует режиму очистки. Двойная буферизация синхронизирует обновление регистра OCR2B с максимальным или минимальным временем отсчета, предотвращая генерацию асимметричных импульсов ШИМ и устранивая возмущающие импульсы. При использовании функции двойной буферизации CPU обращается к регистру буфера OCR2B. Когда функция двойной буферизации отключена, CPU обращается к самому OCR2B.

#### **TIMSK2 - Регистр прерывания TC2**

## *TIFR0 - Регистр флагов прерывания ТС0*

		дает сигнал равенства и устанавливает флаг сравнения OCF0A. Если Флаг прерывания сравнение A OCIE0A установлено в «1», и установлен флаг глобального прерывания, будет генерироваться прерывание сравнения с результатом A. Флаг OCF0A очищается автоматически при выполнении процедуры обслуживания прерывания или может быть очищен путем записи «1» в бит OCF0A.
0	TOV0	Флаг переполнения TC0. Когда счетчик переполняется, устанавливается флаг переполнения TOV0. Если бит разрешения прерывания переполнения TOIE0 установлен на «1» и установлен флаг глобального прерывания, будет генерироваться прерывание переполнения. TOV0 автоматически очищается, когда выполняется эта процедура прерывания, или ее можно очистить, записав «1» в бит TOV0.

## *ASSR- Регистр состояния асинхронного интерфейса*

## 15. Таймер / Счетчик 3 (TMR3)

- 16-разрядный структура, позволяющая формировать 16-битный PWM
- 3 независимых блока сравнения выходных данных
- Двойной буферный регистр сравнения выходных данных
- 1 блок захвата ввода
- Подавитель шума для захвата входного сигнала
- Сброс и загрузка счетчика автоматически, когда сравнение
- ШИМ с коррекцией по фазе без импульсов помех
- Переменный период PWM
- Генератор частоты
- Счетчик внешних событий
- 5 независимых источников прерываний
- Контроль мертвого времени
- Шесть выбираемых источников запуска Автоматического отключения выхода ШИМ

### 115.1. Обзор

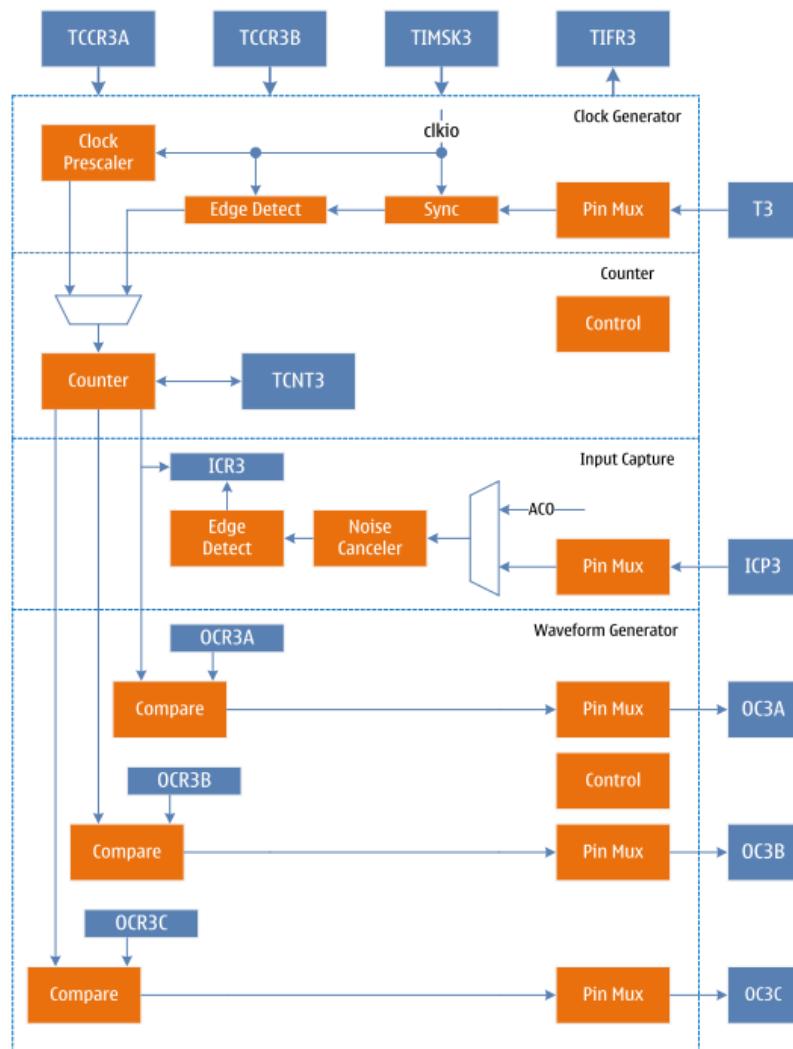


Рис.15.1. Структура TC3

TC3 - это универсальный 16-разрядный счетчик счетчиков, который поддерживает выходы PWM и может точно генерировать сигналы. TC3 содержит 16-разрядный счетчик, блок управления режимом генерации сигнала, три независимых блока сравнения вывода и блок ввода ввода.

Блок управления режимом генерации волны управляет режимом работы счетчика и формированием формы сигнала сравнения. В соответствии с различными режимами работы счетчик очищает, увеличивает или уменьшает каждый из счетчиков часов Clkt3. Clkt3 может быть сгенерирован из внутреннего источника синхронизации или внешнего источника синхронизации.

Когда значение счетчика TCNT3 счетчика достигает максимального значения (равного максимальному значению 0xFFFF или фиксированному значению или регистру сравнения OCR3A или регистру ввода ICR3, определяемому как TOP, максимальное значение определяется как MAX для отличия), счетчик очищается Или вычтите одну операцию.

Когда значение счетчика счетчиков TCNT3 достигает минимального значения (равное 0x0000, определенное как BOTTOM), счетчик будет увеличивать один.

Когда значение счетчика TCNT3 счетчика достигает OCR3A или OCR3B или OCR3C, оно также вызывается при совпадении сравнения. Оно очищает или устанавливает выходной сигнал сравнения OC3A, OC3B или OC3C для генерации сигнала ШИМ.

Когда функция захвата входа включена, счетчик запускает или прекращает подсчет, когда он запускается, и регистр ICR3 регистрирует значение счета во время периода срабатывания сигнала захвата.

## 15.2. Режимы работы

Таймер 1 имеет шесть различных режимов работы, включая Normal, Clear Compare (CTC), Быстродействующую широтно-импульсную модуляцию (FPWM), Фазокорректированную широтно-импульсную модуляцию (PCPWM), Фазу Частотно-модулированная широтно-импульсная модуляция (PFCPWM) и режим захвата ввода (ICP). Он выбирается битом управления режимом генерации волны WGM3 [3: 0]. Ниже описаны эти шесть режимов. Поскольку имеется три независимых блока сравнения вывода, обозначенных соответственно «A», «B» и «C», нижний выходной «x» используется для представления каналов трех блоков сравнения вывода.

### 15.2.1. Нормальный режим

Обычный режим - это самый простой режим работы счетчика таймера это когда бит WGM3[3:0] = 0, а максимальное значение TOP счетчика - MAX (0xFFFF).

В этом режиме счетчик увеличивается на один для каждого счетчика тактов. Когда счетчик достигает переполнения TOP, он возвращается в BOTTOM, чтобы снова начать накапливать.

Флаг переполнения счетчика таймера TOV3 устанавливается в том же такте счета счетчика TCNT3, значение которого становится равным нулю.

Флаг TOV3 в этом режиме похож на 17-й счет, он будет установлен и не

очищается счетом.

Процедура обслуживания прерывания переполнения автоматически очищает флаг TOV3 и может использоваться программным обеспечением для увеличения разрешения счетчика таймера.

В нормальном режиме нет специального случая, и новые значения счета могут быть записаны в любое время. Форма сигнала выходного сигнала сравнения OC3x может быть получена только тогда, когда регистр направления данных вывода OC3x задан как выход. Когда COM3x = 1, сигнал OC3x переключается, когда происходит совпадение сравнения.

В этом случае частоту формы волны можно вычислить по следующей формуле:

$$f_{OC3x\text{normal}} = f_{\text{sys}} / (2 * N * 65536)$$

где N - обозначает Коэффициент предделителя (1, 8, 64, 256 или 1024).

Модуль сравнения результатов может использоваться для генерации прерывания, но прерывания не рекомендуются в нормальном режиме, который потребляет слишком много времени процессора.

### 15.2.2 Режим СТС

Когда установлено WGM3[3:0] = 4 или 12, счетчик/таймер 3 переходит в режим СТС.

Когда WGM3[3:0] = 0, максимальное значение TOP - это OCR3A. Когда WGM3[3] = 1, максимальное значение счета TOP является ICR3.

Ниже описывается режим СТС с использованием WGM3[3:0] = 4. В этом режиме режим подсчета увеличивается для каждого счета. Когда значение счетчика TCNT3 равно TOP, счетчик очищается. Этот режим позволяет пользователю легко контролировать частоту выходного сигнала сравнения, а также упрощает операцию подсчета внешних событий. Когда счетчик достигает TOP, устанавливается флаг соответствия OCF3 для сравнения на выходе, и прерывание будет генерироваться, когда установлен соответствующий бит разрешения прерывания.

Регистр OCR3A может быть обновлен в процедуре обслуживания прерываний. В этом режиме OCR3A не использует двойную буферизацию. Будьте внимательны при обновлении максимального или минимального значения, когда счетчик работает без предварительного делителя или очень малого предварительного делителя.

Если значение, записанное в OCR3A, меньше текущего значения TCNT3, счетчик пропустит сравнение. Перед тем, как произойдет следующее сравнение, счетчик должен сначала пересчитать до MAX, а затем начать отсчет с BOTTOM на OCR3A.

Как и в обычном режиме, флаг TOV3 устанавливается в счетчике, который возвращается к 0x0000.

Форма сигнала выходного сигнала сравнения OC3x может быть получена только тогда, когда регистр направления данных вывода OC3x задан как выход.

Частоту формы волны можно вычислить по следующей формуле:

$$f_{osc3xctc} = f_{sys} / (2 * N * (1 + OCR3A))$$

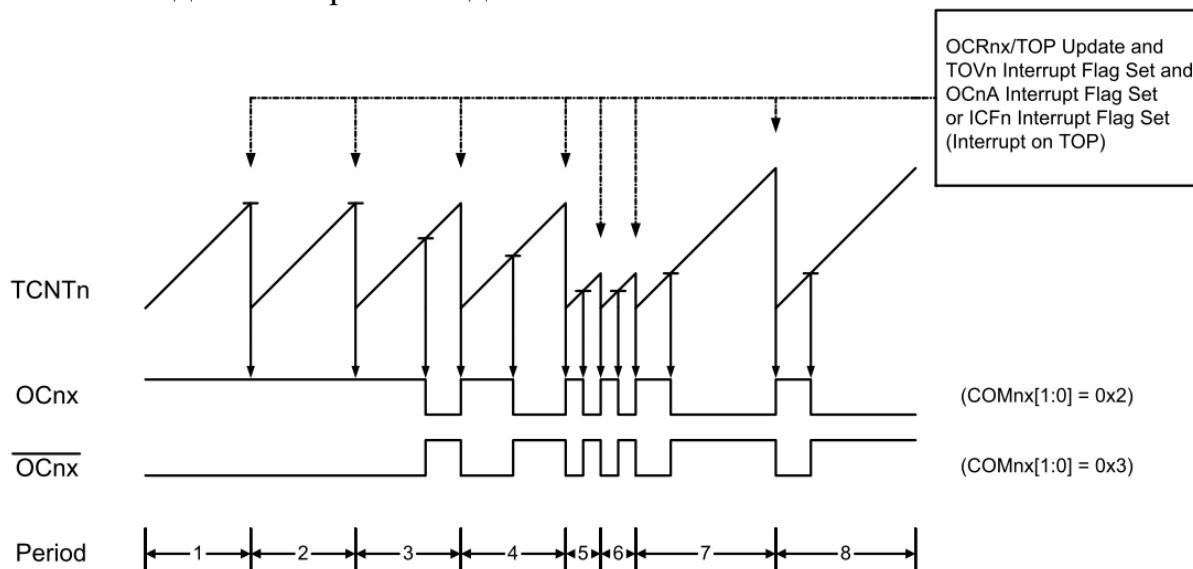
где N - обозначает Коэффициент предделителя (1, 8, 64, 256 или 1024).

Как видно из формулы, когда OCR3A установлено на 0x0000, и нет предварительного делителя, можно получить выходной сигнал с максимальной частотой  $f_{sys} / 2$ .

Когда WGM3[3:0] = 12, он аналогичен WGM3[3:0] = 4, за исключением того, что соответствующий OCR3A изменен на ICR3.

### 15.2.3. Быстрый режим PWM

Когда установлено WGM3[3:0] = 5,6,7,14 или 15, счетчик/таймер 3 переходит в быстрый режим ШИМ, то счетом максимального значения TOP являются 0xFF, 0x1FF, 0x3FF, ICR3 или OCR3A, он может быть использован для получения Высокочастотный ШИМ-сигнал. Разница между быстрым режимом PWM и другими режимами PWM заключается в том, что это односторонняя операция. Счетчик считает до TOP от минимального значения 0x0000, а затем сбрасывается в BOTTOM для повторного подсчета.



Когда значение счетчика достигает TCNT3 верхней или нижней, OC3x выдает сигнал сравнения устанавливается или сбрасывается, в зависимости от выходного сравнения режим установки COM3, а подробное описание регистра. Так как операция односторонняя, рабочая частота быстрого режима PWM в два раза больше двунаправленной операции режима с коррекции PWM фазы.

Высокочастотные характеристики делают быстрый режим PWM подходящим для регулирования мощности, выпрямления и применения ЦАП. Высокочастотные сигналы могут уменьшить размер внешних компонентов (индуктивность, емкость и т. д.), тем самым снижая стоимость системы.

Когда значение счетчика достигает вершины, TOV3 флаг переполнения счетчика/таймера будет установлен, и обновлено значение буфера значение для сравнения на компаратор.

Если прерывание включено, регистр OCR3A может быть обновлен в процедуре обслуживания прерываний.

Форма сигнала выходного сигнала сравнения OC3x может быть получена

только тогда, когда регистр направления данных вывода OC3x задан как выход.

Частоту формы волны можно вычислить по следующей формуле:

$$f_{OC3x} = f_{sys} / (N * (1 + TOP))$$

где N - обозначает Коэффициент предделителя (1, 8, 64, 256 или 1024).

Когда происходит сравнение равенства между TCNT3 и OCR3x, генератор формы сигнала устанавливает (очищает) сигнал OC3x, когда TCNT3 очищается.

В нуле генератор формы сигнала очищает (устанавливает) сигнал OC3x для генерации волны ШИМ. Таким образом, крайности OCR3x будут генерировать специальные формы ШИМ. Когда OCR3x установлен в 0x0000, выходной PWM - это каждый (1 + TOP) счетчик тактов существует узкий выброс. Когда OCR3x установлен в TOP, выходной сигнал постоянно высокий или низкий. Такие, как

Если OCR3A используется как TOP и COM3A = 3, выходной сигнал сравнения OC3A будет генерировать ШИМ с 50% рабочим циклом ШИМ-волны.

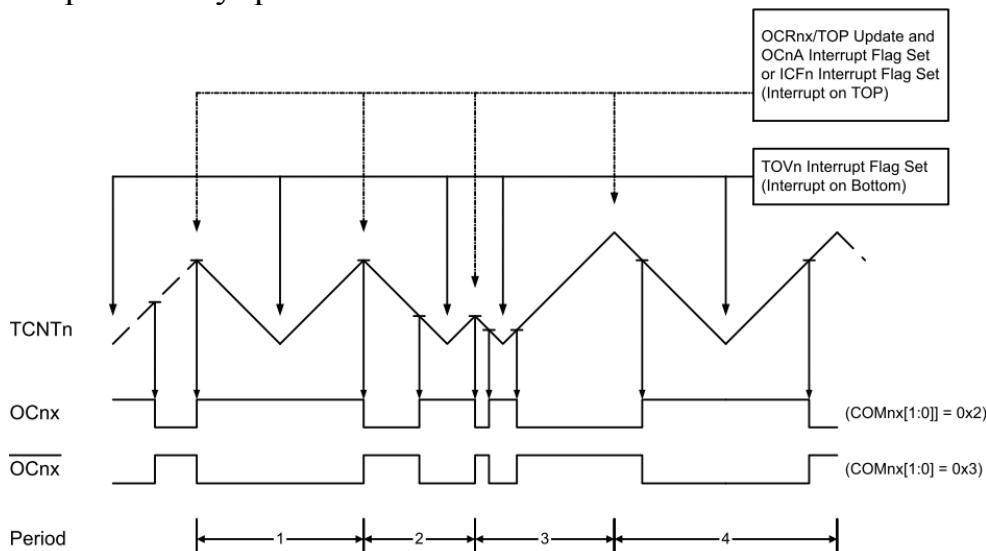
#### 15.2.4. Режим PWM с коррекцией фаз

При установке WGM3[3:0] = 1,2,3,10 или 11 счетчик 1 таймера входит в режим PWM с фазовой коррекцией.

Максимальное значение числа TOP составляет 0xFF, 0x1FF, 0x3FF, ICR3 или OCR3A, соответственно.

Счетчик работает в обоих направлениях, увеличиваясь от BOTTOM до TOP, затем вниз до BOTTOM и повторяя эту операцию. Когда счетчик достигает TOP или BOTTOM, изменяет направление счета, а значение счетчика остается только на одном счете тактов TOP или BOTTOM. Значение подсчета при приращении или уменьшении, когда TCNT3 соответствует OCR3x, выходной сигнал сравнения OC3x будет очищен или установлен, в зависимости от режима вывода сравнения COM3 настройки.

По сравнению с односторонней работой максимальная частота, доступная для двунаправленной операции, меньше, но ее отличная симметрия более подходит для электричества управление машиной.



В режиме Фазовой коррекции PWM флаг TOV3 устанавливается, когда счетчик

достигает BOTTOM и сравнивается, когда счетчик достигает TOP.

Значение буфера обновляется до значения сравнения.

Если прерывание включено, буфер сравнения OCR3x может быть обновлен в процедуре обслуживания прерываний устройства.

Выходной сигнал OC3x сигнала сравнения может быть получен только тогда, когда регистр направления данных вывода OC3x задан как выход.

Частоту формы волны можно вычислить по следующей формуле:

$$f_{OC3xCPWM} = f_{sys} / (N * TOP * 2)$$

где N - обозначает Коэффициент предделителя (1, 8, 64, 256 или 1024).

Во время отсчета генератор сигналов очищает (устанавливает) сигнал OC3x, когда TCNT3 соответствует OCR3x. Во время подсчета, генератор сигналов устанавливает (очищает) сигнал OC3x, когда TCNT3 соответствует OCR3x.

Из-за этого экстремальное значение OCR3x генерирует специальную волну PWM. Когда OCR3x установлен в TOP или BOTTOM, выходной сигнал OC3x остается низким или высоким.

Если OCR3A используется в качестве TOP и COM3A = 1, выходной сигнал сравнения OC3A будет генерировать PWM-сигнал с 50% -ным рабочим циклом. Чтобы обеспечить симметрию выходной ШИМ-волны с обеих сторон BOTTOM, сигнал OC3x будет перевернут в двух случаях, когда совпадение сравнений не произойдет.

Первый случай - когда значение OCR3x изменяется с TOP на другие данные. Когда OCR3x является TOP, а значение счета достигает TOP, выход OC3x совпадает с тем, когда совпадает предыдущий счетчик убывания, то есть OC3x остается неизменным. Обновленное значение нового значения OCR3x (не TOP) обновляется в это время, а значение OC3x удерживается до тех пор, пока совпадение сравнения не произойдет, когда счетчик увеличивается. В этом случае сигнал OC3x не является симметричным относительно минимального значения, поэтому сигнал OC3x необходимо переключать, когда TCNT3 достигает своего максимального значения. То есть первый случай, когда сигнал OC3x переключается во время сравнения, не возникает.

Во втором случае, когда TCNT3 начинает отсчет с более высокого значения, чем OCR3x, сравнение сравнивается с потерями, что вызывает асимметричную ситуацию. Также необходимо перевернуть сигнал OC3x для достижения симметрии по обе стороны от минимального значения.

### 15.2.5. Режим PWM с коррекцией частоты и фазы

Когда установлено значение WGM3[3:0] = 8 или 9, счетчик 3 таймера входит в режим PWM с коррекцией фазовой частоты, а максимальное значение счета TOP - это ICR3 или OCR3A соответственно.

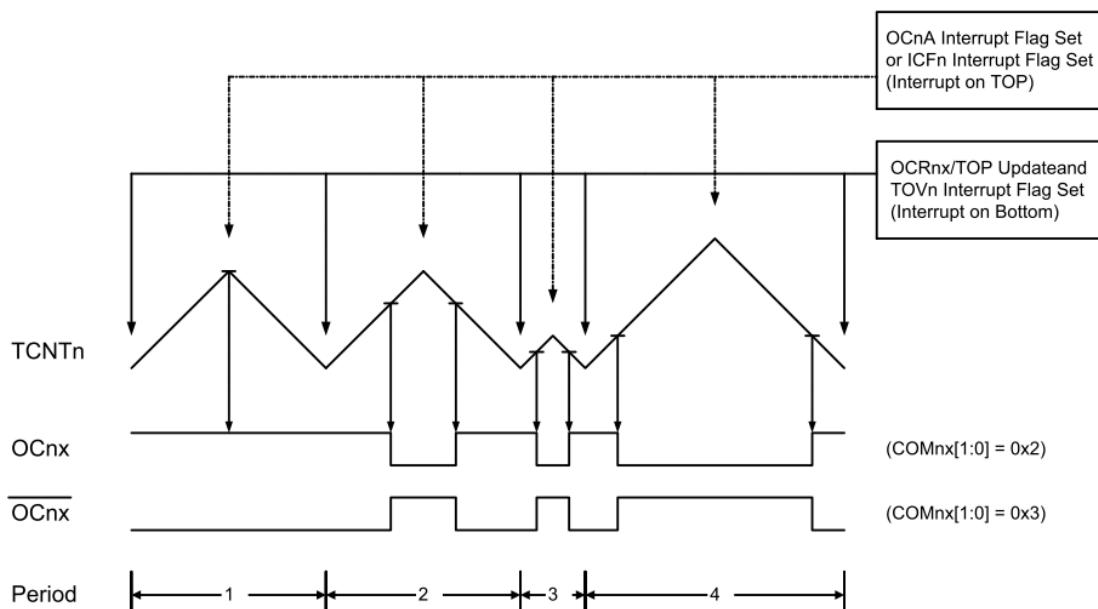
Счетчик работает в обоих направлениях, увеличиваясь от BOTTOM до TOP, затем вниз до BOTTOM и повторяя эту операцию. Когда счетчик достигает TOP или BOTTOM, изменяет направление счета, а значение счетчика остается только на одном счете тактов TOP или BOTTOM.

Во время процесса приращения или уменьшения, когда значение счетчика

TCNT3 совпадает с OCR3x, выходной сигнал сравнения OC3x будет очищен или задан в зависимости от установки режима вывода COM3 сравнения. По сравнению с односторонней работой максимальная частота, получаемая в двунаправленной операции, меньше, но ее превосходная симметрия более подходит для управления двигателем.

Коррекция фазовой частоты В режиме PWM флаг TOV3 устанавливается, когда счетчик достигает BOTTOM, а значение буфера сравнения обновляется до значения сравнения.

Время, когда значение сравнения обновляется, является самой большой разницей между PWM-режимом фазовой коррекции частоты и режимом PWM с фазовой коррекцией.



Если прерывание включено, регистр OCR3x буфера сравнения может быть обновлен в подпрограмме обслуживания прерываний. Когда CPU изменяет значение TOP, то есть значение ORC3A или ICR3, оно должно гарантировать, что новое значение TOP не меньше значения TOP, которое уже используется, иначе сравнение не повторится. Сигнал OC3x сигнала сравнения выходного сигнала может быть получен только в том случае, когда в качестве выходного сигнала установлен регистр направления данных OC3x.

Частоту формы волны можно вычислить по следующей формуле:

$$f_{oc3xcpfcPWM} = f_{sys} / (N * TOP * 2)$$

где N - обозначает Коэффициент предделителя (1, 8, 64, 256 или 1024).

Во время отсчета генератор сигналов очищает (устанавливает) сигнал OC3x, когда TCNT3 соответствует OCR3x. Во время подсчета, генератор сигналов устанавливает (очищает) сигнал OC3x, когда TCNT3 соответствует OCR3x.

Из-за этого экстремальное значение OCR3x генерирует специальную волну PWM. Когда OCR3x установлен в TOP или BOTTOM, выходной сигнал OC3x остается низким или высоким. Если OCR3A используется в качестве TOP и COM3A = 1, выходной сигнал сравнения OC3A будет генерировать PWM-сигнал с 50% -ным рабочим циклом.

Регистр OCR3x обновляется во время BOTTOM, длины счета восходящей и

нисходящей сторон одинаковы, (при постоянных значениях TOP) приводит к симметричным сигналам с правильной частотой и фазой.

При использовании фиксированного значения TOP лучше всего использовать регистр ICR3, как значение TOP, то есть установить WGM3[3:0] = 8. В это время регистр OCR3A используется только для генерации выходного сигнала PWM.

Если вы хотите генерировать частотно-изменяющуюся PWM-волну, вы должны изменить значение TOP. Для этого приложения будут более подходящими характеристики двойной буферизации OCR3A.

#### 15.2.6. Режим ввода данных

Входной захват используется для захвата внешних событий и предоставления им отметки времени, указывающей, когда произошло событие. Это можно сделать в предыдущем режиме подсчета, за исключением того, что режим генерации сигналов с использованием значения ICR3 в качестве значения TOP.

Триггерный сигнал для внешнего события вводится на вывод ICP3 и может также реализовываться аналоговым компаратором. Когда изменяется логический уровень на выводе ICP3 или изменяется уровень АСО аналогового компаратора, и это изменение уровня захватывается блоком захвата ввода, запускается входной захват и 16-разрядное значение счета TCNT3. Данные копируются в регистр захвата ввода ICR3 и устанавливается флаг ICF3 ввода.

Если бит ICIE33 установлен в «1», флаг захвата ввода генерирует прерывание ввода. Источник запуска ICP3 или АСО захвата входа выбирается путем установки бита управления захватом аналогового сравнения ACIC аналогового сравнения и регистра ACSR состояния. Следует отметить, что изменение источника триггера может привести к захвату ввода, поэтому после изменения источника триггера для ICF3 необходимо выполнить четкую операцию, чтобы избежать ошибочных результатов.

Сигнал захвата входного сигнала проходит через дополнительный шумоподавитель и отправляется к детектору кромок в соответствии с конфигурацией бита управления выбора входа захвата ICES3, чтобы убедиться, что обнаруженный фронт удовлетворяет условию триггера. Шумоподавитель - это простой цифровой фильтр, который четыре раза проверяет входной сигнал и выводит на выход детектора, только когда четыре образца равны.

Шумоподавитель управляет битом ICNC3 регистра TCCR3B, чтобы включить или отключить его.

При использовании функции захвата ввода значение регистра ICR3 следует читать как можно раньше после того, как установлен ICF3, потому что значение ICR3 будет обновляться после следующего события захвата.

Рекомендуется включить прерывание захвата ввода. В любом режиме захвата ввода не рекомендуется изменять значение TOP во время работы. Введенная метка времени может использоваться для расчета частоты, рабочего цикла и других характеристик сигнала и создания журнала для события триггера. Измерение рабочего цикла внешнего сигнала требует, чтобы фронт триггера был изменен после каждого захвата. Следовательно, фронт сигнала запуска должен быть как можно быстрее изменен после считывания значения ICR3.

### 15.2.7. Управление мертвым временем

Когда бит DTEN3 установлен на «1», функция включения мертвого времени активирована. Выходные сигналы OC3A и OC3B будут вставлены в заданное мертвое время на основе формы сигнала, создаваемого выходным сигналом сравнения B-канала. Длина - это значение времени, соответствующее числу счетчиков в регистре DTR3. Как показано на рисунке ниже, вставки мертвого времени OC1A и OC1B основаны на выходной форме сигнала канала B. Когда COM3A и COM3B являются «2» или «3», полярность сигнала OC3A такая же, как полярность формы сигнала OC3B. Когда COM3A и COM3B «2» или «3» соответственно, форма волны OC3A и форма волны OC3B одинаковы. Полярность противоположна.

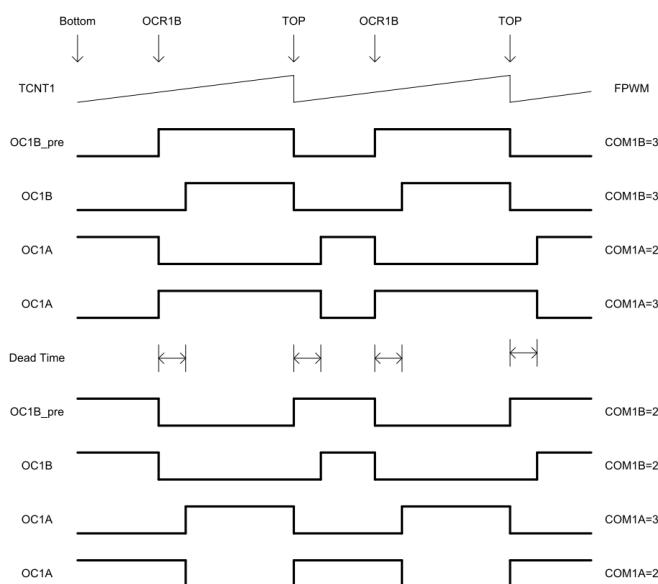


Рис.15 Контроль мертвого времени ТС3 в режиме FPWM

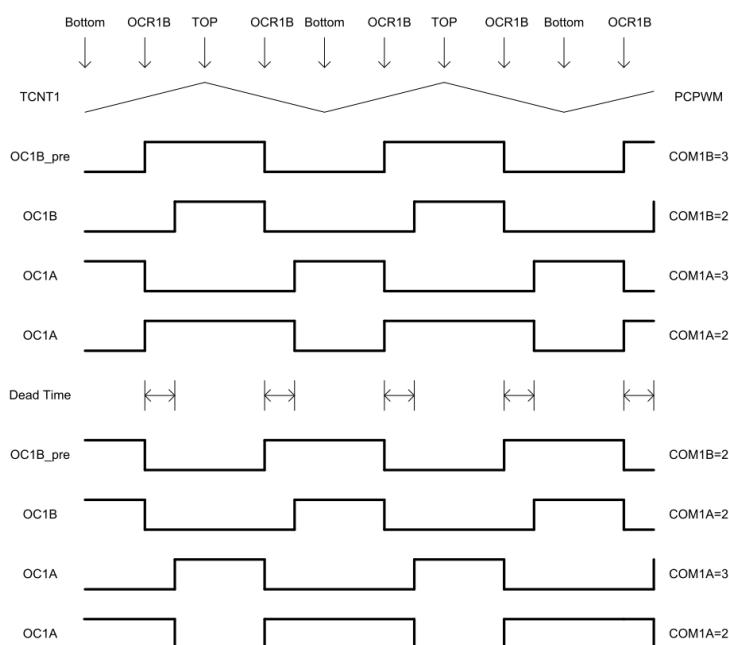


Рис. 15. Тайм-тайм ТС3 в режиме PCPWM

Когда бит DTEN3 установлен на «0», функция вставки мертвого времени отключена. Выходные сигналы OC3A и OC3B являются формами сигналов, генерируемыми соответствующими результатами сравнения.

### 15.2.8. Регистры Описания

## *Список регистров ТС3*

Регистр	Адрес	Значение по умолчанию	Описание
TCCR3A	0x90	0x00	Регистр управления A
TCCR3B	0x91	0x00	Регистр управления B
TCCR3C	0x92	0x00	Регистр управления C
TCCR3D	0x93	0x00	Регистр управления D
TCNT3L	0x94	0x00	Счетчик значений - младший байт
TCNT3H	0x95	0x00	Счетчик значений - старший байт
ICR3L	0x96	0x00	Регистр захвата - младший байт
ICR3H	0x97	0x00	Регистр захвата - старший байт
OCR3AL	0x98	0x00	Регистр сравнения A выходных данных - младший байт
OCR3AH	0x99	0x00	Регистр сравнения A выходных данных - старший байт
OCR3BL	0x9A	0x00	Регистр сравнения B выходных данных - младший байт
OCR3BH	0x9B	0x00	Регистр сравнения B выходных данных - старший байт
DTR3L	0x9C	0x00	Регистр контроля мертвого времени - младший байт
DTR3H	0x9D	0x00	Регистр контроля мертвого времени - старший байт
OCR3CL	0x9E	0x00	Регистр сравнения C выходных данных - младший байт
OCR3CH	0x9F	0x00	Регистр сравнения C выходных данных - старший байт
TIMSK3	0x71	0x00	Регистр маски прерывания счетчика/таймера
TIFR3	0x38	0x00	Регистр флагов прерываний счетчика/таймера

## *TCCR3A - Регистр управления A*

## СОМЗх - Режим Управление сравнением без PWM

СОМ3х [1:0]	Описание
0	ОС3х отключен, операция ввода-вывода общего назначения
1	Переключение сигнала ОС3х во время сравнения
2	Очистить сигнал ОС3х во время сравнения
3	Установите сигнал ОС3х при сравнении

## СОМ3х - Быстрый Режим PWM Управление сравнением

СОМ3x [1:0]	Описание
0	ОС3х отключен, операция ввода-вывода общего назначения
1	Когда WGM3 равен 15: сигнал ОС3А переключается во время сравнения, а ОС3В выключается. Когда WGM3 является другим значением: ОС3х отключен, общая операция ввода-вывода
2	Очистка сигнала ОС3х во время сравнения и установка сигнал ОС3х, когда произойдет максимальное совпадение
3	Установка сигнала ОС3х во время сравнения и очистка сигнал ОС3х, когда произойдет максимальное совпадение

СОМ3х - Режим PWM с коррекцией фаз Управление сравнением

СОМ3x [1:0]	Описание
0	ОС3x отключен, операция ввода-вывода общего назначения
1	Когда WGM3 равен 9 или 11: сигнал ОС3А переключается во время сравнения, ОС3В выключается Когда WGM3 является другим значением: ОС3x отключен, общая операция ввода-вывода
2	Очистка ОС3x при совпадении, когда счет вверх. Установка ОС3x при совпадении, когда счет вниз.
3	Установка ОС3x при совпадении, когда счет вверх. Очистка ОС3x при совпадении, когда счет вниз.

### *TCCR3B - Регистр управления B*

		копируется в регистр ICR3 и устанавливается флаг ICF3 ввода. Если прерывание включено, генерируется прерывание ввода.																		
5	-	Зарезервировано																		
4	WGM33	WGM33, WGM32, WGM31, WGM30 вместе образуют режим WGM3 управления генерацией формы волны [3: 0] для управления режимом подсчета счетчиков и режимом генерации формы сигнала. Подробнее см. В таблице Описания режима генерации волны.																		
3	WGM32																			
2	CS32	<p>Предделение тактов источника синхронизации для счетчика времени 0.</p> <table border="1"> <thead> <tr> <th>CS3[2:0]</th> <th>Описание</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Нет источника синхронизации, стоп-счет</td> </tr> <tr> <td>1</td> <td>clksys</td> </tr> <tr> <td>2</td> <td>clksys / 8 из предделителя</td> </tr> <tr> <td>3</td> <td>clksys / 64 из предделителя</td> </tr> <tr> <td>4</td> <td>clksys / 256 из предделителя</td> </tr> <tr> <td>5</td> <td>clksys / 1024 от предделителя</td> </tr> <tr> <td>6</td> <td>Внешние такты T1, триггер со спадающим фронтом</td> </tr> <tr> <td>7</td> <td>Внешние такты T1, триггер с нарастающим фронтом</td> </tr> </tbody> </table>	CS3[2:0]	Описание	0	Нет источника синхронизации, стоп-счет	1	clksys	2	clksys / 8 из предделителя	3	clksys / 64 из предделителя	4	clksys / 256 из предделителя	5	clksys / 1024 от предделителя	6	Внешние такты T1, триггер со спадающим фронтом	7	Внешние такты T1, триггер с нарастающим фронтом
CS3[2:0]	Описание																			
0	Нет источника синхронизации, стоп-счет																			
1	clksys																			
2	clksys / 8 из предделителя																			
3	clksys / 64 из предделителя																			
4	clksys / 256 из предделителя																			
5	clksys / 1024 от предделителя																			
6	Внешние такты T1, триггер со спадающим фронтом																			
7	Внешние такты T1, триггер с нарастающим фронтом																			
1	CS31																			
0	CS30																			

WGM3 - Управление режимом генерации сигнала.

WGM1[3:0]	Режим работы	Значение TOP	Обновить момент OCR3	Установите момент TOV3
0	Normal	0xFFFF	Немедленно	MAX
1	8-битный PCPWM	0x00FF	TOP	BOTTOM
2	9-битный PCPWM	0x01FF	TOP	BOTTOM
3	10-битный PCPWM	0x03FF	TOP	BOTTOM
4	CTC	OCR1A	Немедленно	MAX
5	8-битный PCPWM	0x00FF	BOTTOM	TOP
6	9-битный PCPWM	0x01FF	BOTTOM	TOP
7	10-битный PCPWM	0x03FF	BOTTOM	TOP
8	PFCPWM	ICR1	BOTTOM	BOTTOM
9	PFCPWM	OCR1A	BOTTOM	BOTTOM
10	PCPWM	ICR1	TOP	BOTTOM
11	PCPWM	OCR1A	TOP	BOTTOM
12	CTC	ICR1	Немедленно	MAX
13	Удержание	-	-	-
14	FPWM	ICR1	TOP	TOP
15	FPWM	OCR1A	TOP	TOP

### *TCCR3C - Регистр управления C*

		<b>СОМЗА1</b>	<b>СОМЗА0</b>	<b>Описание</b>
		0	0	Нормальных операций порта, ОСЗА отключено.
		0	1	Бистабильная схема ОСЗА на Сравнении Компаратора.
		1	0	Очистить ОСЗА на Сравнении Компаратора.
		1	1	Установить ОСЗА на Сравнении Компаратора.
		Чтение FOC3A всегда возвращает ноль.		
		Бит FOC1B активный только, когда биты WGM определяют режим без PWM.		
		<b>СОМЗВ1</b>	<b>СОМЗВ0</b>	<b>Описание</b>
6	FOC3B	0	0	Нормальных операций порта, ОСЗВ отключено.
		0	1	Бистабильная схема ОСЗВ на Сравнении Компаратора.
		1	0	Очистить ОСЗВ на Сравнении Компаратора.
		1	1	Установить ОСЗВ на Сравнении Компаратора.
		Чтение FOC3B всегда возвращает ноль.		
5	DOC3B	Когда бит DOC3B установлен на «1», активируется сигнал сравнения выключения источника запуска ОСЗВ. Когда происходит событие триггера, аппаратное обеспечение автоматически отключает вывод сигнала ОСЗВ. Когда бит DOC3B установлен на «0», выходной сигнал сравнения выключения источника ОСЗВ отключен. Выход сигнала ОСЗВ не отключается при возникновении события триггера.		
4	DOC3A	Когда бит DOC3A установлен на «1», активируется сигнал сравнения выключения источника запуска ОСЗА. Когда происходит событие триггера, аппаратное обеспечение автоматически отключает вывод сигнала ОСЗА. Когда бит DOC3A установлен на «0», выходной сигнал сравнения выключения источника ОСЗА отключен. Выход сигнала ОСЗА не отключается при возникновении события триггера.		
3	DTEN3	Если бит DTEN3 установлен на «1» - мертвое времени включено. Оба ОСЗА и ОСЗВ вставляют мертвое время в форму волны, генерируемую выходным сигналом сравнения В-канала, а вставленный мертвый интервал определяется временем отсчета, соответствующим регистру DTR3. Полярность выходного сигнала ОСЗА определяется соответствием между СОМЗА и СОМЗВ. См. ОСЗА для таблицы полярности сигнала после вставки мертвого времени. Когда бит DTEN3 установлен на «0», вставка мертвого времени запрещена. Сигналы ОСЗА и ОСЗВ являются формами сигналов, генерируемыми соответствующими результатами сравнения.		
2	-	Зарезервировано		
1	DOC3C	Когда бит DOC3C высок, разрешено отключить выходное сравнение С. Если после завершения условия вывода отключено, бит СОМЗС очищается, а выходной контакт ОСЗС отключается. Этот вывод становится универсальной операцией ввода-вывода. Когда бит DOC3C является низким, функция С отключить функцию сравнения вывода недействительна.		
		Бит FOC1B активный только, когда биты WGM определяют режим без PWM.		
		<b>СОМ1В1</b>	<b>СОМ1В0</b>	<b>Описание</b>
0	FOC3C	0	0	Нормальных операций порта, ОС1В отключено.
		0	1	Бистабильная схема ОС1В на Сравнении Компаратора.
		1	0	Очистить ОС1В на Сравнении Компаратора.
		1	1	Установить ОС1В на Сравнении Компаратора.
		Чтение FOC1B всегда возвращает ноль.		

## TCCR3D - Регистр управления TCCR3D

### TCCR3D - Регистр управления D

Адрес: 0x93				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	DSX37	DSX36	DSX35	DSX34	-	-	DSX31	DSX30
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
Бит	Имя	Определение бит						
7	DSX37	При установке бита DSX37 в «1», переполнение <b>TC0</b> активируется в качестве источника триггера для выключения сигнала сравнения OC3x. Когда бит DOC3x равен «1», нарастающий фронт выбранного флага прерывания автоматически отключает выходной сигнал OC3x. При установке бита DSX37 в «0» переполнение <b>TC0</b> отключается как источник триггера для выключения выходного сигнала сравнения OC1A/OC1B.						
6	DSX36	При установке бита DSX36 в «1» переполнение <b>TC2</b> активируется в качестве источника триггера для выключения сигнала сравнения OC3x. Когда бит DOC3x равен «1», нарастающий фронт выбранного флага прерывания автоматически отключает выходной сигнал OC3x. При установке бита DSX36 в «0» переполнение <b>TC2</b> отключается как источник триггера для выключения выходного сигнала сравнения OC3x.						
5	DSX35	При установке бита DSX35 на «1» уровень изменения <b>Вывода 1</b> включается в качестве источника триггера для выключения выходной формы сигнала сравнения OC3x. Когда бит DOC3x установлен в «1», передний фронт бит регистра флага прерывания выбранного источника триггера автоматически отключает выходной сигнал OC3x. При установке бита DSX35 в «0», изменение <b>Вывода 1</b> отключается как источник триггера для выключения выходного сигнала сравнения OC3x.						
4	DSX34	Когда бит DSX34 установлен в «1», внешнее <b>Прерывание 1</b> включается в качестве источника триггера для выключения выходной сигнала сравнения OC3x. Когда бит DOC3x равен «1», передний фронт бита регистра флагка выбранного прерывания автоматически отключает выходной сигнал OC3x. Когда бит DSX34 установлен в «0», <b>Прерывание 1</b> отключается как источник триггера для выключения выходного сигнала сравнения OC3x.						
3	-							
2	-							
1	DSX31	Когда бит DSX31 установлен в «1», <b>Аналоговый компаратор 1</b> включен как источник триггера для выключения выходной сигнала сравнения OC3x. Когда бит DOC3x равен «1», передний фронт флага прерывания выбранного источника триггера автоматически отключает выходной сигнал OC3x. Когда бит DSX31 равен «0», <b>Аналоговый компаратор 1</b> отключается как триггер, чтобы отключить выходной сигнал сравнения OC3x.						
0	DSX30	Когда бит DSX30 установлен в «1», <b>Аналоговый компаратор 0</b> включен как источник триггера для выключения выходной сигнала сравнения OC3x. Когда бит DOC3x равен «1», передний фронт флага прерывания выбранного источника триггера автоматически отключает выходной сигнал OC3x. Когда бит DSX01 равен «0», <b>Аналоговый компаратор 0</b> отключается как триггер, чтобы отключить выходной сигнал сравнения OC3x.						

### OC1A / OC1B - Отключить управление Выбор источника

DOC1x	DSX1n = 1	Источник	Описание
0	-	-	Бит DOC0x равен «0», функция выключения источника выходного сигнала отключена.

1	0	Аналоговый компаратор 0	Передний фронт ACIF0 отключает выходной сигнал OC1x
1	1	Аналоговый компаратор 1	Передний фронт ACIF1 отключает выходной сигнал OC1x
1	4	Внешнее прерывание 1	Передний фронт INTF1 отключает выходной сигнал OC1x
1	5	Изменение уровня контакта 1	Передний фронт PCIF1 отключает выходной сигнал OC1x
1	6	Переполнение TC2	Передний фронт TOV2 отключает выходной сигнал OC1x
1	7	Переполнение TC0	Передний фронт TOV0 отключает выходной сигнал OC1x

**Примечание:** 1) Когда DSX1n = 1 указывает, что n-й бит регистра DSX1 равен 1, биты регистра могут быть установлены одновременно.

### TCNT3L - Регистр счетчика Младший байт

TCNT3L - Регистр счетчика								
Адрес: 0x94				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	TCNT3L7	TCNT3L6	TCNT3L5	TCNT3L4	TCNT3L3	TCNT3L2	TCNT3L1	TCNT3L0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	TCNT3L	Младший байт значения счетчика TC1. Регистры TCNT1H и TCNT1L объединены для формирования 16-разрядного счетчика TCNT1. Чтение и запись 16-битных регистров требует двух операций. При записи 16-битного TCNT1 сначала следует записать TCNT1H. При чтении 16-битного TCNT1 сначала прочтайте TCNT1L. При записи в регистр TCNT1, это предотвратит совпадение сравнения в течение следующего такта таймера, даже если таймер остановлен. Это позволяет не вызывать прерывания при равенстве значений регистра TCNT1 и OCR1x. Если значение, записанное в TCNT1, равно или больше значение OCR1x, сравнение будет потеряно, что приведет к возникновению неправильной формы сигнала. Таймер прекращает отсчет, когда источник синхронизации не выбран, но ЦПУ все еще может получить доступ к TCNT1. Счетчик записи процессора имеет более высокий приоритет, чем операция очистки или добавления или вычитания.						

### TCNT3H - Регистр счетчика Старший байт

TCNT3H - Регистр счетчиков								
Адрес: 0x95				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	TCNT3H7	TCNT3H6	TCNT3H5	TCNT3H4	TCNT3H3	TCNT3H2	TCNT3H1	TCNT3H0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	TCNT3H	Старший байт значения счетчика TCNT3. См. пояснения TCNT3L						

### ICR3L - Регистр захвата Младший байт

ICR3L - Регистр захвата - младший байт								
Адрес: 0x96				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	ICR3L7	ICR3L6	ICR3L5	ICR3L4	ICR3L3	ICR3L2	ICR3L1	ICR3L0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	ICR3L	Младший байт значения ввода входного сигнала TCNT3. ICR3H и ICR1L объединены для формирования 16-битного ICR3. Чтение и запись 16-битных регистров требует двух операций. При записи 16-битного ICR3 сначала должен быть записан ICR3H. При чтении 16-битного ICR3 сначала прочтайте ICR3L. Когда запускается входной захват, значение счетчика TCNT3 обновляется и копируется в регистр ICR3. Регистр ICR3 также может использоваться для определения ТОР-значения счетчика.						

*ICR3H - Регистр захвата Старший байт*

OCR3AL - Регистр сравнения А выходных данных Младший байт

*OCRЗАН - Регистр сравнения А выходных данных Старший байт*

<b>Бит</b>	<b>Имя</b>	<b>Определение бит</b>
7:0	OCR3AH	Старший байт значения регистра OCR3A. См. пояснения OCR3AL

## *OCR3BL - Регистр сравнения В выходных данных Младший байт*

*OCR3BH - Регистр сравнения В выходных данных Старший байт*

*OCR3CL - Регистр сравнения С выходных данных Младший байт*

Двойная буферизация может синхронизировать обновление регистра OCR3C с максимальным или минимальным временем отсчета, предотвращая генерацию асимметричных импульсов ШИМ и устранивая некорректные импульсы. Когда используется функция двойной буферизации, процессор обращается к регистру буфера OCR3C. Когда функция двойной буферизации отключена, процессор обращается к OCR3C.

## *OCRICH - Регистр сравнения С выходных данных Старший байт*

*DTR3L - Регистр контроля мертвого времени Младший байт*

DTR3H - Регистр контроля мертвого времени Старший байт

*TIMSK3 - Регистр маски прерывания счетчика/таймера*



		глобального прерывания, будет генерироваться прерывание сравнения с результатом В. Флаг OCF3B очищается автоматически при выполнении процедуры обслуживания прерывания или может быть очищен путем записи «1» в бит OCF3B.
1	OCF3A	Выход TC3 флаг сравнения А. Когда TCNT3 равен OCR3A, блок сравнения дает сигнал равенства и устанавливает флаг сравнения OCF3A. Если Флаг прерывания сравнение А OCIE3A установлено в «1», и установлен флаг глобального прерывания, будет генерироваться прерывание сравнения с результатом А. Флаг OCF3A очищается автоматически при выполнении процедуры обслуживания прерывания или может быть очищен путем записи «1» в бит OCF3A.
0	TOV3	Флаг переполнения TC3. Когда счетчик переполняется, устанавливается флаг переполнения TOV3. Если бит разрешения прерывания переполнения TOIE3 установлен на «1» и установлен флаг глобального прерывания, будет генерироваться прерывание переполнения. TOV3 автоматически очищается, когда выполняется эта процедура прерывания, или ее можно очистить, записав «1» в бит TOV3.

## 16. Синхронный последовательный периферийный интерфейс (SPI)

- Полнодуплексная трехпроводная синхронная передача данных
- Управление ведущим или ведомым устройством
- Передача с наименьшим или наивысшим приоритетом
- 7 программируемых скоростей передачи данных
- Флаг прерывания передачи
- Запись символа столкновений Механизм защиты
- Пробуждение из режима ожидания
- Режим двойной скорости при работе хоста
- Поддерживает хост с двухпроводным режимом ввода
- 4 буферных регистра для обзора ввода / вывода

### 16.1. Обзор

SPI в основном включает в себя три части: предварительный делитель тактов, детектор тактов, детектор выбора ведомого устройства, передатчик и приемник.

Регистры управления и состояния разделяются этими тремя частями.

Предделитель тактов работает только в режиме работы хоста. Битами управления скоростью передачи выбирают коэффициент деления для генерации соответствующих тактов частоты и выводят его на вывод SPCK. Детектор тактов работает только в режиме ведомого, обнаруживает входной сигнал тайм-сигнала из выводов SPCK и сдвигает в регистры сдвига передачи и приема в соответствии с режимом передачи данных SPI.

Извещатель ведомого детектирует сигнал выбора спутника SPSS и получает состояние передачи для управления работой передатчика и приемника.

Передатчик состоит из сдвигового регистра и логики управления передачей.

Приемник состоит из сдвигового регистра, четырех принимающих буферов и приемной логики управления.

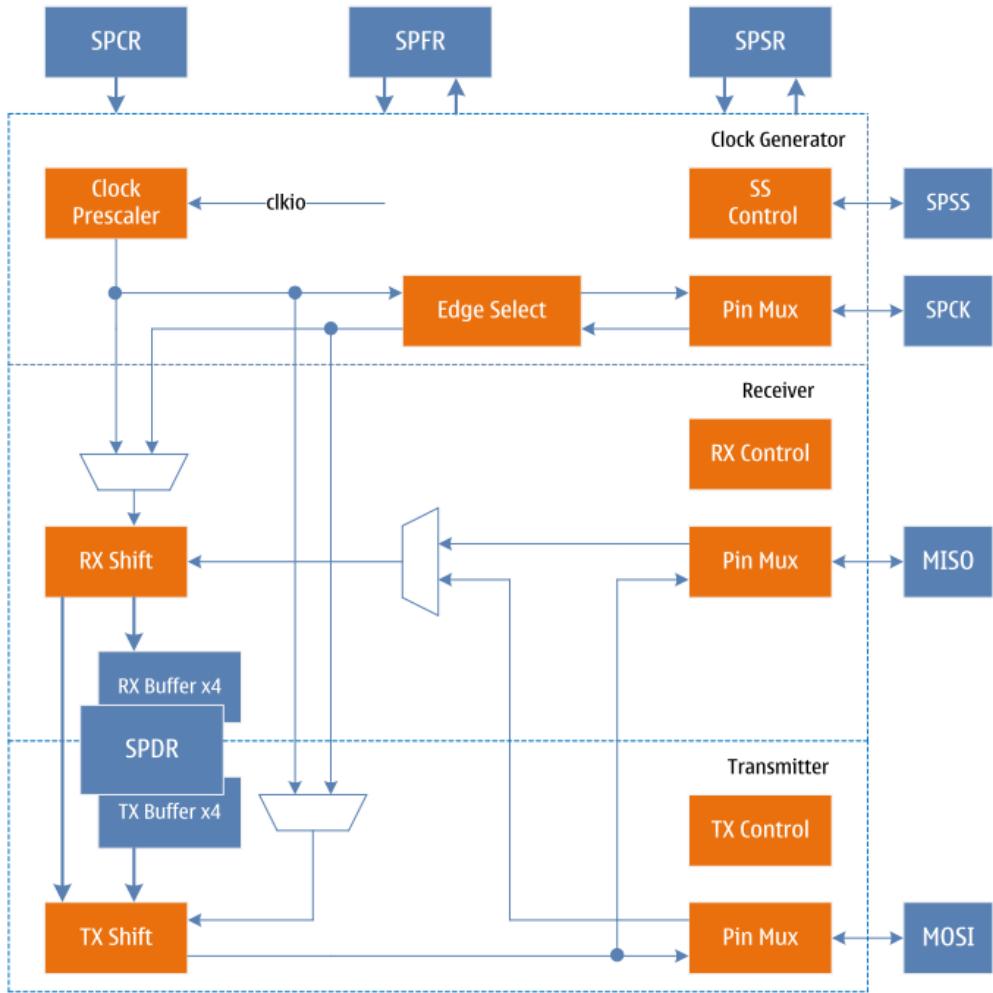


Рис. Структурная схема SPI

## 16.2. Генерация тактов.

Логика генерации тактов делится на предварительный делитель основного тактового генератора и ведомый тактовый детектор, которые работают в режиме ведущего и ведомого режимов соответственно.

Предделитель тактов использует биты управления скоростью передачи битов и биты управления скоростью передачи для выбора коэффициента деления для генерации соответствующих частотно-разделенных тактов (в общей сложности семь выбираемых частотно-делящихся коэффициентов, см. Описание регистра). Выход на вывод **SPCK** Связь обеспечивает такты, обеспечивая при этом синхронизацию сдвига для внутренних регистров сдвига передачи и приема.

Детектор тактов выполняет обнаружение фронтов на входном такте **SPCK** и сдвигает передатчик и приемник в соответствии с режимом передачи данных SPI.

Чтобы обеспечить правильную выборку тактового сигнала, как высокие, так и низкие уровни тактов **SPCK** должны быть больше, чем 2 тактовых цикла системы.

## 16.3. Передача и прием

Модули SPI для передачи и приема поддерживают одновременную передачу и прием в однопроводном режиме, и только двухпроводный прием в

двухпроводном режиме.

#### 16.4. Однолинейная передача и прием

Хост SPI отключает ведомый сигнал SPSS, который должен связываться для начала процесса передачи. Ведущий и ведомый подготавливают данные, которые необходимо передать.

Ведущий генерирует тактовый импульс на синхронизирующем сигнале SPCK для обмена данными. Основные данные сдвигаются из MOSI и передаются из MISO. Ведомые данные удаляются из MISO и перемещаются из MOSI. После данных хост подтягивает сигнал SPSS для завершения связи.

При настройке в качестве хоста модуль SPI не управляет выводом SPSS и должен обрабатываться программным обеспечением пользователя.

Программное обеспечение вытаскивает PIN-код SPSS, выбирает ведомого для связи и инициирует передачу.

Программное обеспечение записывает данные, которые должны быть переданы в регистр SPDR, и запускается генератор тактовых импульсов.

Аппаратное обеспечение генерирует тактовый сигнал связи и сдвигает 8-разрядные данные в ведомый и перемещает данные из подчиненного устройства. После сдвига одного байта данных генератор тактовых импульсов останавливается и устанавливается флаг завершения передачи SPIF. Программное обеспечение может переписать данные в регистр SPDR, чтобы продолжить передачу следующего байта, или может поднять сигнал SPSS для завершения текущей передачи. Последние входящие данные будут сохранены в буфере приема.

При настройке в качестве ведомого устройства модуль SPI будет оставаться спящим до тех пор, пока сигнал SPSS будет высоким, сохраняя триггер MISO. В это время программное обеспечение может обновлять содержимое регистра SPDR. Даже если на выводе SPCK есть импульс входного тактового сигнала, данные SPDR не будут сдвинуты до тех пор, пока сигнал SPSS не будет выведен на низкий уровень.

Когда передача одного байта данных завершена, оборудование устанавливает флаг завершения передачи SPIF. На этом этапе программное обеспечение может продолжать записывать данные в регистр SPDR перед чтением сдвинутых данных. Последние входящие данные сохраняются в буфере приема.

Модуль SPI имеет только четыре буфера в направлении передачи и четыре буфера в направлении приема.

При передаче данных регистр SPDR может быть записан, когда буфер передачи не заполнен (т. е. бит флага полного буфера передачи, WRFULL, является низким). При получении данных, когда буфер приема находится в пустом состоянии (то есть бит пустого флага приема RDSPPF является низким), принятый символ может быть прочитан путем доступа к регистру SPDR.

#### 16.5. Двухпроводный режим

2-проводного приемного модуля SPI действителен только в рабочем режиме хоста. Отличие от однопроводного режима заключается в том, что оба MOSI и MISO используются хостом для приема данных, и каждый синхронизирующий

сигнал SPCK принимает одновременно 2 бита данных (линия MISO Данные до этого, данные в строке MOSI после).

После получения двух байтов данных аппаратное обеспечение устанавливает флаг завершения передачи SPIF, а данные сохраняются в буфере приема и сдвиговом регистре. Программное обеспечение должно дважды считывать регистр SPDR, чтобы получить полученные два байта данных.

Следует отметить, что, хотя хост не передает данные в ведомый в двухпроводном режиме, программному обеспечению по-прежнему необходимо записать данные в регистр SPDR, чтобы запустить генератор тактовых импульсов для генерации тактового сигнала связи.

Запись в регистр SPDR может принимать два байта данных.

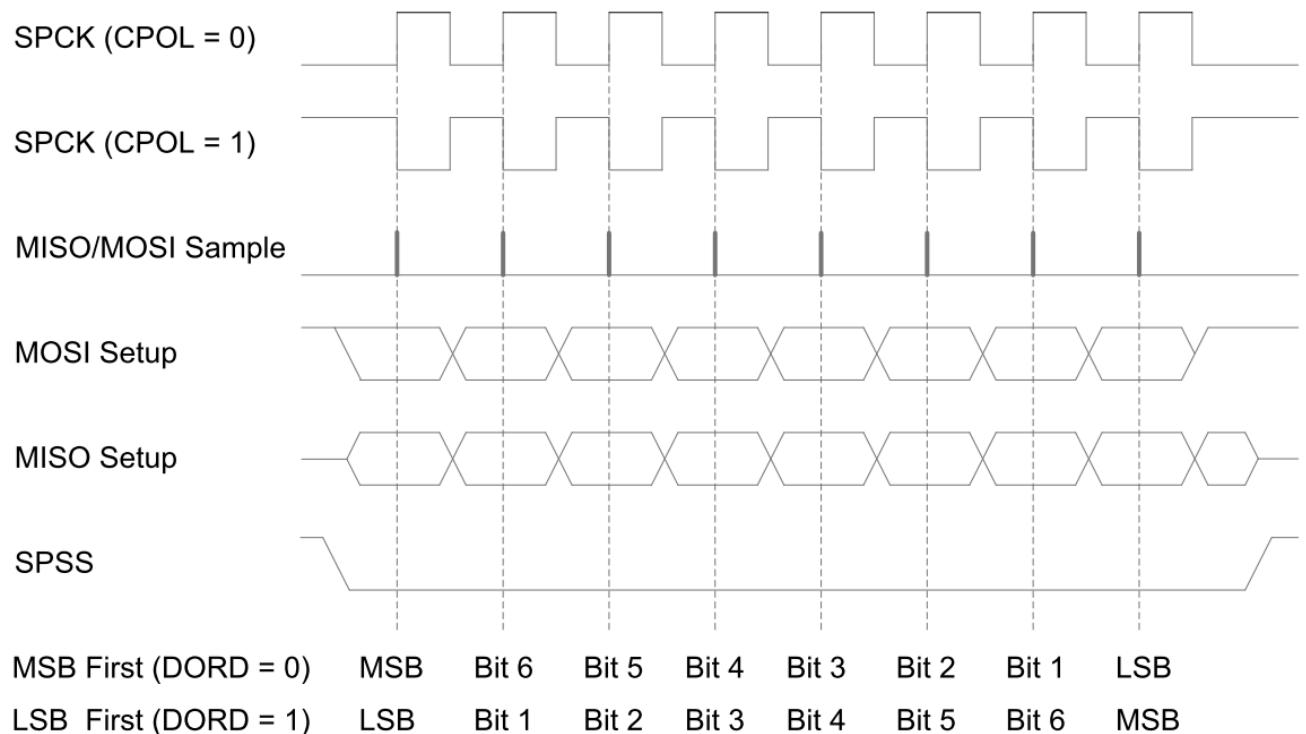
## Режим данных

В однопроводном режиме SPI имеет 4 комбинации фазы и полярности SPCK в отношении последовательных данных и управляется CPHA и CPOL, как показано в следующей таблице. CPHA и CPOL

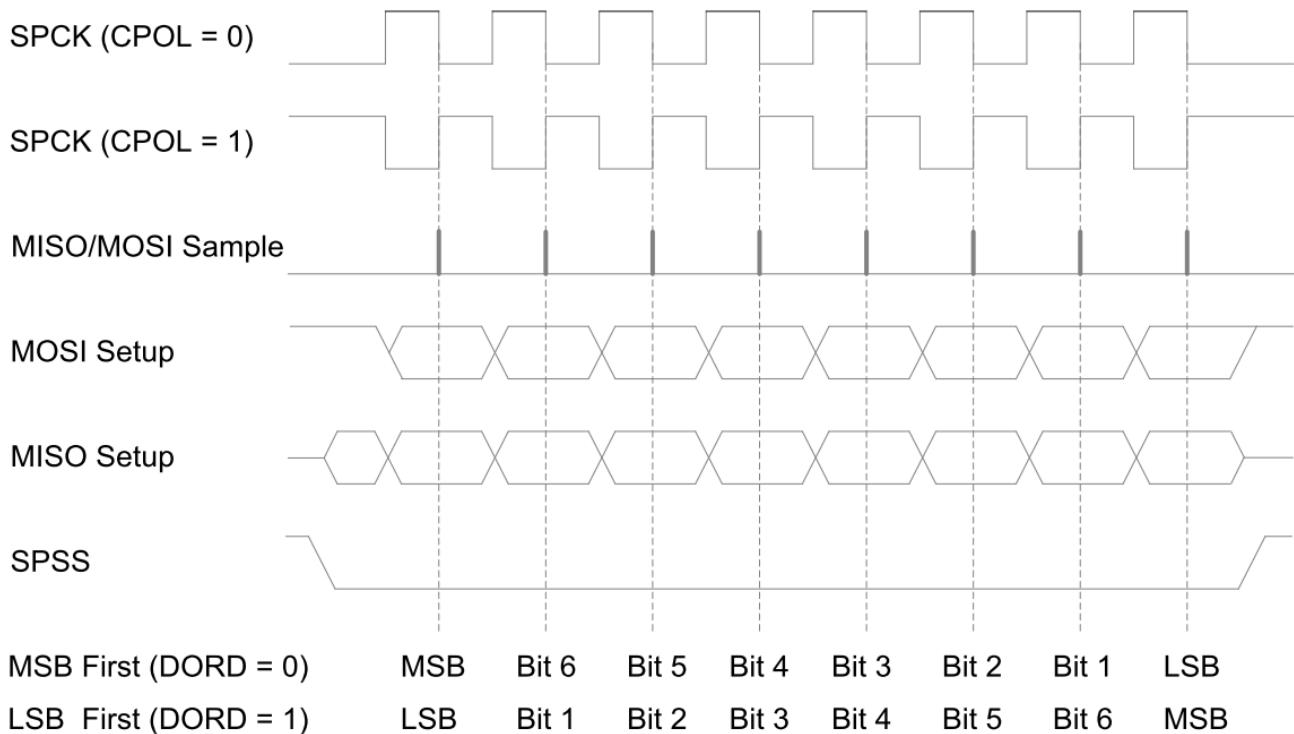
Выбор режима передачи данных

CPOL	CPHA	Исходный край	Конец края	Режим SPI
0	0	Выборка (передний фронт)	Настройка (задний фронт)	0
0	1	Настройка (нарастающий фронт)	Выборка (задний фронт)	1
1	0	Выборка (задний фронт)	Настройка (нарастающий фронт)	2
1	1	Настройка (задний фронт)	Выборка (передний фронт)	3

Когда CPHA = 0, выборки данных и установка краев тактового сигнала следующие:

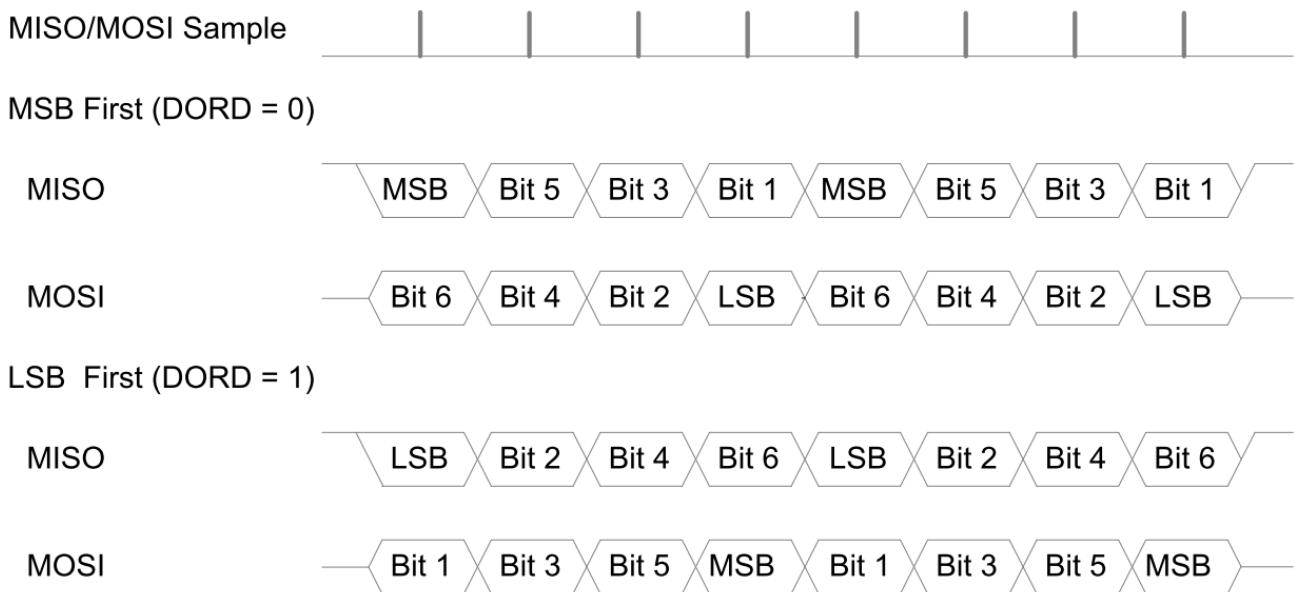


Когда CPHA = 1, выборка данных и установка краев тактового сигнала следующие:



SPI, когда СРНА является «1».

В двухпроводном режиме MISO и MOSI используются в качестве входа хоста. Время выборки данных все еще определяется режимом передачи данных. Режим выборки показан на рисунке ниже:



Режим выборки данных SPI, когда DUAL является «1» в режиме мастера

## 16.6. Функция SPSS Выводов.

При настройке в качестве ведомого устройства всегда вход вывод SPSS сигнала выбора подчиненного устройства. Когда контакт SPSS удерживается на низком уровне, активируется интерфейс SPI, вывод MISO становится выходным выводом (программное обеспечение выполняет соответствующую конфигурацию

порта), а остальные контакты являются входами. Когда контакт SPSS удерживается высоко, модуль SPI сбрасывается и больше не получает данные. Вывод SPSS очень полезен для синхронизации пакетов / байтов, который синхронизирует бит счетчика ведомого устройства с генератором тактов.

Когда SPSS в «1», ведомое устройство SPI немедленно сбрасывает логику приема и передачи и отбрасывает неполные данные в сдвиговом регистре.

При настройке в качестве хоста программное обеспечение пользователя может определять направление вывода SPSS.

Если SPSS настроен как выходной сигнал, его можно использовать для управления выводом SPSS ведомого устройства.

Если SPSS настроен как вход, он должен быть высоким для обеспечения нормальной работы хоста.

Когда сконфигурирован как ведущий, а вывод SPSS является входом, когда внешняя **схема вытягивает вывод** SPSS, модуль SPI предполагает, что другой хост выбрал себя как подчиненный и начал передавать данные.

1. Удалите бит MSTR в регистре SPCR и преобразуйте его в подчиненный, чтобы MOSI и SPCK стали входами.

2. Установите бит SPIF в регистр SPSR и сгенерируйте прерывание SPI, если прерывание включено.

Поэтому при использовании метода прерывания для обработки передачи данных ведущего SPI и существует вероятность того, что SPSS будет выведено на низком уровне, программа обслуживания прерываний должна проверить, является ли бит MSTR в «1». Если оно очищено, программное обеспечение должно установить его для повторного включения основного режима SPI.

## 16.7. Инициализация SPI

Инициализация SPI должна проведена перед общением.

Процесс инициализации обычно включает в себя выбор операции подчиненного устройства, настройку режима передачи данных, выбор скорости передачи битов и управление направлением каждого вывода.

Управление направлением вывода в режиме ведущего и ведомого отличается, как показано в следующей таблице:

Управление направлением выводов

Вывод	Ведущий режим	Ведомый режим
MOSI	Определяется пользовательским ПО	Запись
MISO	Запись	Определяется пользовательским ПО
SPCK	Определяется пользовательским ПО	Запись
SPSS	Определяется пользовательским ПО	Запись

### 16.7.1. Инициализация ведущего SPI

Процесс инициализации хоста SPI выглядит следующим образом:

1. Установите бит MSTR, бит управления выбором бит, режим передачи данных, порядок передачи данных, разрешение прерывания или нет, И активировать двойную линию;

2. Установите выходы MOSI и SPCK в качестве выходов;
  3. Установите бит SPE.

В основном режиме штырь SPSS может быть установлен на выход, если нежелательно, чтобы модуль SPI использовался как подчиненный другой хост.

#### 16.7.2. Инициализация ведомого устройства SPI

Процесс инициализации режима ведомого режима SPI выглядит следующим образом:

1. Очистите бит MSTR, установите режим передачи данных, порядок передачи данных и включено ли прерывание или нет.
  2. Установите вывод MISO в качестве выхода;
  3. Установите бит SPE.

## 16.8. Регистры Описание

## *Список регистров SPI*

Регистр	Адрес	Значение по умолчанию	Описание
SPCR	0x4C	0x00	Регистр управления SPI
SPSR	0x4D	0x00	Регистр состояния SPI
SPDR	0x4E	0x00	Регистр данных SPI
SDFR	0x39	0x00	Регистр контроля буфера SPI

## *SPCR - Регистр управления SPI*

		<b>CPOL</b>	<b>Исходный край</b>	<b>Конец края</b>
0		0	Восходящий фронт	Падение фронт
1		1	Падение фронт	Восходящий фронт
2	СРНА	СРНА	Когда бит СРНА установлен в «1», начальная граница устанавливает данные и завершает данные выборки. Когда бит СРНА установлен в «0», данные выборки начального фронта и данные настройки конечного края.	
		0	Выборка	Установить
		1	Установить	Выборка
1	SPR1	SPR1 и SPR0 используются для выбора тактовой частоты для передач SPI. Для конкретного метода управления см. Таблицу отношений SPCK и системы.		
0	SPR0			

### SPSR - Регистр Состояния SPI

SPSR - Регистр Состояния SPI								
Адрес: 0x4D			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	SPIE	WCOL	-	-	-	DUAL	-	SPI2X
R/W	R/W	R/W	-	-	-	R/W	-	R/W
Бит	Имя	Определение бит						
7	SPIF	Флаг прерывания SPI. Флаг SPIF устанавливается после завершения последовательной передачи. В режиме Ведомого, когда вывод SPSS сконфигурирован как вход и в «0», SPIF в «1». Если бит SPIE в «1» и бит разрешения глобального прерывания в регистре SPCR в «1», генерируется прерывание SPI. Бит SPIF автоматически очищается после ввода процедуры обслуживания прерывания или бит SPIF очищается, сначала считывая регистр SPSR, а затем обращается к регистру SPDR.						
6	WCOL	Запись регистра SPDR во время передачи данных устанавливает бит WCOL. Бит WCOL можно очистить, прочитав регистр SPSR, прежде чем обращаться к регистру SPDR.						
5:3	-	Зарезервировано						
2	DUAL	Двухпроводный бит управления режимом. Когда бит DUAL установлен в «1», режим двухпроводной передачи SPI включен. Если бит DUAL установлен на «0», режим двухпроводной передачи SPI отключается. Режим двухпроводной передачи действителен только в режиме ведущего устройства SPI. В качестве входных данных хоста используются MISO и MOSI. Режим передачи данных описывается в главах двухпроводного приема и режимах передачи данных.						
1	-	Зарезервировано						
0	SPI2X	Бит управления скоростью SPI. Когда бит SPI2X равен «1», скорость передачи SPI удваивается. Когда бит SPI2X установлен на «0», скорость передачи SPI не удваивается. Для конкретного метода управления см. Таблицу отношений SPCK и системы.						

В следующей таблице показана взаимосвязь между SPCK и системными часами.

### Связь между SPCK и системными часами

SPI2X	SPR1	SPR0	Частота SPCK
0	0	0	fsys/4

0	0	1	fsys/16
0	1	0	fsys/64
0	1	1	fsys/128
1	0	0	fsys/2
1	0	1	fsys/8
1	1	0	fsys/32
1	1	1	fsys/64

## *SPDR - Регистр данных SPI*

## *SDFR - Регистр контроля буфера SPI*

		записанные в буфер передачи, переданы, бит WREMPTR в «1», что указывает на то, что буфер передачи пуст. Когда записывается регистр SPDR, буферный адрес передачи накапливается, и когда данные, записанные в буфер передачи, не все передаются, в буфере приема имеется по меньшей мере один байт данных. Бит WREMPTR в «0», указывающим буфер передачи не пустой. Когда бит WREMPTR в «1», адрес буфера передачи будет обнулен.
1	WRPTR1	Недопустимый адрес передачи. При записи в регистр SPDR данные в SPDR записываются в буфер передачи, и буферный адрес передачи накапливается.
0	WRPTR0	Когда бит WREMPTR установлен (запись 1), адрес буфера передачи будет обнулен.

## 17. USART0 - универсальный синхронный / асинхронный последовательный трансивер

- Полнодуплексная работа (независимые последовательные регистры приема и передачи)
- Асинхронная или синхронная работы
- Операция Ведущего или ведомого устройства
- Высокоточный генератор скорости передачи данных
- Поддерживает 5, 6, 7, 8 или 9 бит данных и 1 или 2 стоповых бита
- Механизмы генерации и проверки четности с поддержкой аппаратного обеспечения
- Обнаружение превышения скорости
- Обнаружение ошибок кадров
- Фильтрация шума, включая обнаружение ложного начала бит и цифровой фильтр низких частот
- Три независимых прерывания: отправка конечного прерывания, отправка регистра данных пустым прерыванием и получение конечного прерывания
- Многопроцессорный режим связи
- Режим асинхронной связи с двойной скоростью

### 17.1. Обзор

USART в основном состоит из трех частей: генератора тактовых импульсов, передатчика и приемника. Эти три раздела разделяют регистры управления и состояния.

Генератор тактов состоит из генератора скорости передачи и логики синхронизации внешнего входного тактового сигнала в режиме синхронного подчиненного режима. Вывод XCK используется только для синхронного режима передачи.

Передатчик включает в себя буфер данных записи, регистр последовательного сдвига, генератор четности и логику управления, необходимую для обработки различных форматов кадров.

Буфер данных записи позволяет осуществлять непрерывную передачу данных без введения задержек между кадрами данных.

Приемник имеет блок синхронизации часов и данных для приема

асинхронных данных. В дополнение к блоку восстановления приемник включает в себя четность, логику управления, регистр последовательного сдвига и двухступенчатый UDR буфера приема. Приемник поддерживает тот же формат кадра, что и передатчик, и может обнаруживать ошибки кадрирования, превышение скорости передачи данных и ошибки четности.

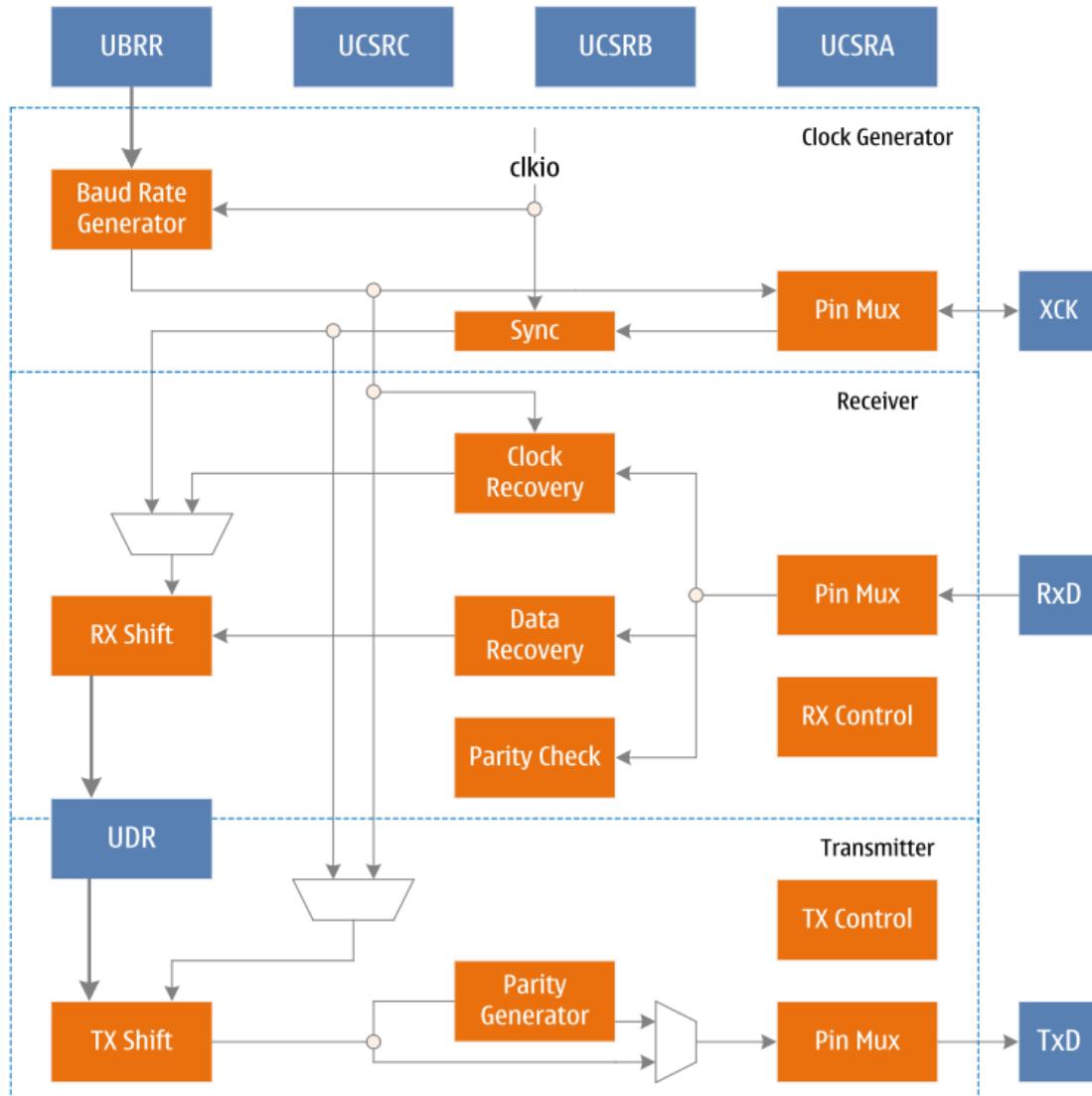


Рис.17.1 Структура USART

## 17.2. Генерация тактов.

Синхронизатор тактов генерирует базовые такты для передатчика и приемника. USART поддерживает четыре режима работы тактов: обычный асинхронный режим, двухскоростной асинхронный режим, ведущий синхронный режим и ведомый синхронный режим.

Бит UMSEL USCRC выбирает либо синхронный, либо асинхронный режим.

Бит U2X UCSRA контролирует двойное разрешение скорости в асинхронном режиме.

Регистр направления данных (мультиплексированный с IO) вывода XCK, который активен только в синхронном режиме, определяет, генерируется ли источник синхронизации внутри (ведущий режим) или извне (режим ведомого).

### 17.3. Генератор скорости передачи.

Регистр скорости передачи, UBRR и счетчик убывания последовательности соединены вместе как программируемый предварительный делитель или генератор скорости передачи для USART. Убывающий счетчик работает на системных тактах ( $f_{sys}$ ) и автоматически загружает значение регистра UBRR, когда оно достигает нуля или записывается регистр UBRL. Такты генерируются, когда счетчик достигает нуля. Такты используются как выходные такты генератора скорости передачи, а частота равна  $f_{sys} / (UBRR + 1)$ .

В следующей таблице приведены формулы для расчета скорости передачи (бит в секунду) и значения UBRR в различных режимах работы.

Режим работы	Расчета скорости передачи (1)	Расчета значения UBRR
Асинхронный нормальный режим	$BAUD = f_{sys} / (16 * (UBRR + 1))$	$UBRR = f_{sys} / (16 * BAUD) - 1$
Асинхронный режим двойной скорости	$BAUD = f_{sys} / (8 * (UBRR + 1))$	$UBRR = f_{sys} / (8 * BAUD) - 1$
Режим синхронного хоста	$BAUD = f_{sys} / (2 * (UBRR + 1))$	$UBRR = f_{sys} / (2 * BAUD) - 1$

#### Примечание:

- Скорость передачи определяется как скорость передачи бит в секунду (бит / с);
- BAUD - скорость передачи,  $f_{sys}$  - системные часы, а UBRR - комбинированное значение регистров скорости передачи данных UBRRH и UBRL.

### 17.4. Режим двойной скорости работы.

Устанавливая бит U2X в регистре UCSRA для удвоения скорости передачи, этот бит действителен только в асинхронном режиме работы, и этот бит устанавливается в «0» в синхронном режиме работы.

Установка этого бита уменьшит вдвое значение делителя скорости передачи, эффективно удваивая скорость передачи асинхронной связи. В этом случае приемник использует только половину числа выборок для выборки данных и восстановления тактов, поэтому требуются более точные настройки скорости передачи и системные такты. Передатчик не изменяется.

### 17.5. Внешние такты

Режим синхронного ведомого режима управляет внешним тиковым сигналом. Внешние часы используются передатчиком и приемником после прохождения через регистр синхронизации и детектор фронта. Этот процесс вводит две системные тиковые задержки. Поэтому максимальная тиковая частота внешнего ХСК ограничена следующей формулой:

$$f_{XCK} < f_{sys} / 4$$

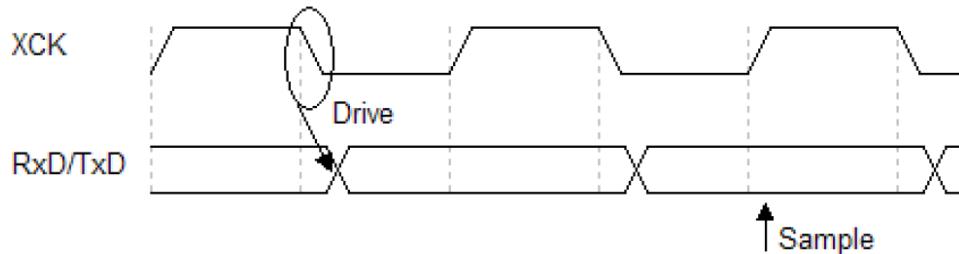
Обратите внимание, что  $f_{sys}$  имеет решение о стабильности для системных тактов. Чтобы предотвратить потерю данных из-за частотного дрейфа, рекомендуется зарезервировать достаточный запас.

## 17.6. Синхронные такты.

В синхронном режиме вывод XCK используется для ввода тактов (ведомый режим) или выход синхронизации (ведомый режим).

Основным правилом связи между фронтом тактового сигнала и выборкой данных и изменением данных является то, что фронт тактового сигнала, используемый для выборки ввода данных (RxD), противоположен фронту тактового сигнала, используемому для изменения выходного конца данных.

UCPOL = 1



UCPOL = 0

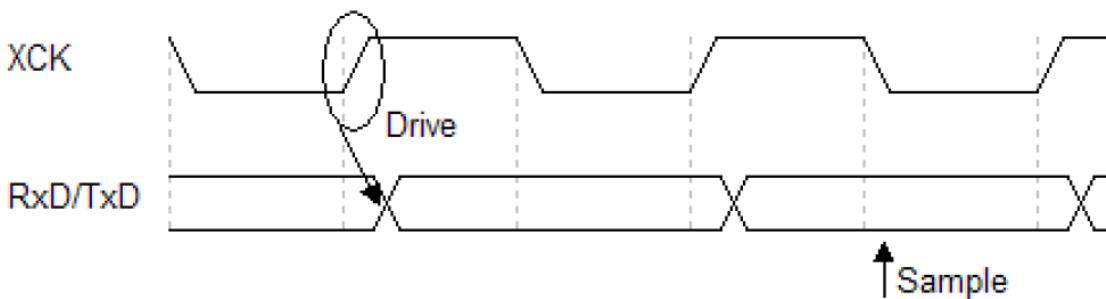


Рис. Время XCK в синхронном режиме

Как показано на рисунке выше, когда значение UCPOL равно «1», выход данных изменяется на заднем фронте XCK, и выборка данных выполняется по нарастающему фронту XCK. Когда значение UCPOL равно «0», выход данных изменяется на передний фронт XCK. Сэмплирование данных происходит на заднем фронте XCK.

## 17.7. Формат кадра.

Кадр последовательных данных состоит из слова данных плюс биты синхронизации (биты начала и окончания) и бит четности для исправления ошибок.

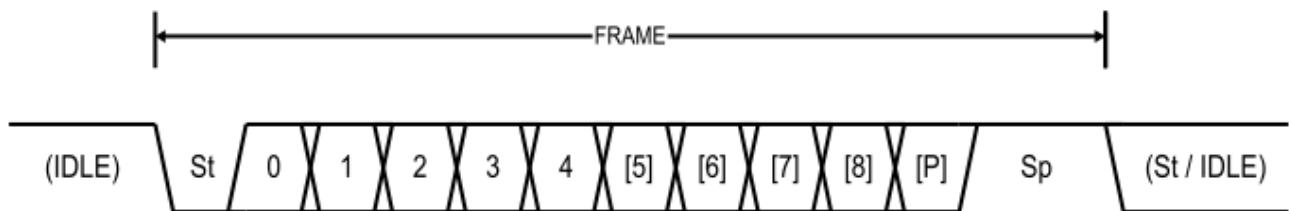
USART принимает следующие 30 комбинаций форматов фреймов данных:

- 1 стартовый бит
- 5, 6, 7, 8 или 9 бит данных
- бит четности, бит четности четности или бит четности
- 1 или 2 стоповых бита

Кадр данных начинается с начального бита, за которым следует младший значащий бит слова данных, за которым следуют другие биты данных с самым высоким битом слова данных. Пакет, успешно передается до 9 бит данных.

Если проверка включена, бит четности будет сопровождаться словом данных, за которым следует стоповый бит.

Когда передается полный кадр данных, следующий новый кадр данных может быть передан немедленно или линия передачи может оставаться бездействующей (высокий уровень). На следующем рисунке показана возможная структура кадра данных. Квадратные скобки необязательны.



Структура кадра USART

St	Стартовый бит, всегда низкий.
(n)	Биты данных (от 0 до 8).
P	Бит четности. Может быть нечетным или четным.
Sp	Стоп бит, всегда высокий.
IDLE	Нет передач на линии связи (RxDn или TxDn). Линия IDLE должна быть высокой.

Структура кадра данных задается UCSZ [2: 0], UPM [1: 0] и USBS в регистрах UCSRB и UCSRC. Приемник и передатчик те же настройки. Любые изменения настроек могут привести к потере текущих передач данных.

Среди них UCSZ [2: 0] определил бит данных кадра данных, UPM [1: 0] используется для включения и подтверждения типа проверки, USBS устанавливает для кадра один или два конечных бита.

Приемник игнорирует второй стоповый бит, поэтому ошибки кадрирования обнаруживаются только тогда, когда первый конечный бит равен «0».

## 17.8. Вычисление битов четности

Бит четности вычисляется путем выполнения операции «исключающее или» всех битов данных. Если используется нечетная четность, результат является исключаительным или инвертированным. Отношение между битом четности и битами данных выглядит следующим образом:

$$\begin{aligned} P_{even} &= d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 0 \\ P_{odd} &= d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 1 \end{aligned}$$

Описание:

- 1)  $P_{even}$  - четный результат проверки
- 2)  $P_{odd}$  - нечетный результат проверки
- 3)  $d_n$  -  $n$ -й бит данных

Если используют бит четности, он располагается между последним информационным разрядом и первым стоповым битом последовательного кадра.

## 17.9. Инициализация USART

Перед передачей должен быть инициализирован USART.

Процесс инициализации обычно включает настройку скорости передачи, настройку структуры кадра и включение приемника или передатчика по мере необходимости.

Для операций USART, управляемых прерываниями, флаг глобального прерывания очищается при инициализации, и все прерывания USART отключены.

При повторной инициализации, такой как изменение скорости передачи или структуры кадра, необходимо обеспечить отсутствие передачи данных.

Флаг TXC может использоваться для определения того, завершил ли передатчик все передачи.

Флаг RXC может использоваться для определения того, считаются ли какие-либо данные в буфере приема. Если для этой цели используется флаг TXC, флаг TXC должен быть очищен перед каждой передачей данных (до регистрации регистра UDR).

## 17.10. Передатчик.

Установка бит TXEN в регистре UCSRB позволит передавать данные USART. Функция IO общего назначения вывода TxD после включения заменяется функцией USART и становится последовательным выходом передатчика.

Перед отправкой данных установите скорость передачи в бодах, рабочий режим и формат кадра.

Если используется синхронный режим передачи, тактовый сигнал, применяемый к выходу XCK, является синхронизирующим сигналом для передачи данных.

### 17.10.1. Передача кадров с 5 до 8 битами данных.

Данные, которые нужно отправить, загружаются в буфер отправки, чтобы начать передачу данных. ЦП загружает данные, записывая их в регистр UDR.

При отправке сдвигового регистра можно отправить новый кадр данных, данные в буфере будут перенесены в регистр сдвига. Когда сдвиговый регистр находится в режиме ожидания (передача данных не выполняется) или последний бит останова предыдущего кадра был отправлен, он будет загружать новые данные.

Как только сдвиговый регистр загрузит новые данные, он будет передавать полный кадр в соответствии с установленными настройками.

### 17.10.2. Передача кадров 9 битами данных.

Если передается кадр из 9-разрядных данных, девятый бит данных должен быть записан в бит TXB8 в регистре UCSRB, прежде чем записывать младшие 8 бит в регистр UDR данных передачи.

9-й бит данных используется для представления адресного фрейма в многомашинной связи и может использоваться для обработки протокола в

синхронной связи.

#### 17.10.3. Передача бита четности.

Схема формирования четности генерирует соответствующий бит четности для кадра последовательных данных.

Когда бит четности активирован ( $UPM1 = 1$ ), логика управления передачей вводит бит четности между последним битом слова данных и первым стоповым битом.

#### 17.10.4. Передача, флаги и обработка прерываний

Передатчик USART имеет два флага: USART Data Register Empty Flag UDRE и Transfer Complete Flag TXC. Оба флага могут генерировать прерывания.

Флаг пустого регистра данных UDRE используется для указания того, может ли буфер отправки записывать новые данные. Этот бит установлен в «1», когда буфер передачи пуст и «0», когда он заполнен. Когда бит UDRE равен «1», CPU может записывать новые данные в UDR регистра данных и наоборот.

Когда бит разрешения запрета прерывания регистра данных UDRIE в регистре UCSRB установлен в «1», запрос пустой запроса прерывания данных USART будет генерироваться всякий раз, когда UDRE установлен (и разрешено глобальное прерывание). Запись в регистр UDR очистит UDRE. При использовании прерывания для передачи данных в UDR должны быть записаны новые данные для очистки UDRE или для отключения запрета на регистрацию регистра данных в процедуре обслуживания пустых прерываний регистра данных. В противном случае, как только процедура обслуживания прерывания закончится, новое прерывание будет сгенерировано снова.

Флаг передачи передачи TXC устанавливается, когда весь кадр данныхмещен из регистра сдвига передачи, и нет новых данных в регистре передачи.

Когда бит разрешения прерывания передачи передачи TXCIE (и разрешено глобальное прерывания) на UCSRB установлен на «1», прерывание передачи USART-передачи будет выполняться с установленным флагом TXC. После того, как введена процедура обслуживания прерываний, флаг TXC автоматически очищается или CPU также может записать «1» на этот бит, чтобы очистить его.

#### 17.10.5. Отключение передатчика

После сброса TXEN передатчик может быть отключен только после того, как все данные будут переданы, то есть нет данных, которые должны быть переданы в регистре сдвига передачи и в регистре буфера передачи.

После того, как передатчик отключен, вывод TxD возобновляет функцию универсального ввода-вывода общего назначения.

## 17.11. Приемник

Установка бит разрешения приема (RXEN) регистра UCSRB запускает приемник USART. Функция IO общего назначения контакта RxD после включения заменяется функцией USART и становится последовательным входом приемника. Скорость передачи, режим работы и формат кадра должны быть установлены до приема данных. Если используется синхронный режим приема, часы на выводе XCK используются как часы передачи.

### 17.11.1. Прием кадров с 5 до 8 битами данных.

Когда приемник обнаруживает действительный стартовый бит, он начинает принимать данные. Каждый бит данных, следующих за стартовым битом, будет приниматься с заданной скоростью передачи данных или синхронизацией XCK до тех пор, пока первый стоповый бит кадра не будет принят, а второй стоповый бит будет проигнорирован приемником.

Каждый бит принятых данных отправляется в регистр сдвига приема. После приема первого стопового бита приемник устанавливает флаг завершения приема данных RXC в регистре UCSRA и сдвигает полный кадр данных в сдвиговом регистре.

В буфере приема ЦП может получать полученные данные, читая регистр UDR.

### 17.11.2. Прием кадров с 9 битами данных.

Если задан кадр данных из 9-битных данных, бит RXB8 регистра UCSRB должен быть сначала прочитан, чтобы получить 9-битные данные, прежде чем читать более низкие 8-битные данные из UDR. Это правило также относится к флагам статуса FE, DOR и PE.

Чтение местоположения памяти UDR изменяет состояние буфера приема, который, в свою очередь, изменяет биты TXB8, FE, DOR и PE, которые также хранятся в буфере.

### 17.11.3. Прием, флаг полного буфера и обработка прерываний

Приемник USART имеет флаг: флаг завершения приема RXC, который указывает, есть ли какой-либо непрочитанный результат в буфере приема.

Если приемник отключен, Буфер приема обновляется и RXC также очищается.

После установки бит RXCIE разрешения прерывания приема UCSRB, пока установлен флаг RXC (и глобальное прерывание включено), при этом произойдет прерывание приема USART.

При приеме данных с использованием режима прерывания конец приема данных прерывает процесс обслуживания.

Данные должны считываться из UDR, чтобы очистить флаг RXC. В противном случае, как только обработка прерывания завершится, будет создано

новое прерывание.

#### 17.11.4. Флаг ошибки приема

Приемник USART имеет три флагка ошибок: ошибка кадрирования FE, переполнение данных DOR и ошибка четности PE. Все они находятся в регистре UCSRA. Флаг ошибки сохраняется в буфере приема вместе с фреймом данных. Все флаги ошибок не могут генерировать прерывания.

Флаг ошибки фрейма FE указывает состояние первого стопового бита следующего считываемого фрейма, сохраненного в буфере приема. Столовый бит является правильным (значение равно «1»), а флаг FE равен «0», иначе флаг FE равен «1». Этот флаг может использоваться для обнаружения потери синхронизации, прерывания передачи и может также использоваться для обработки протокола.

Флаг переполнения данных DOR указывает, что данные были потеряны из-за заполнения буфера приема. Когда буфер приема заполнен, в регистр сдвига приема уже есть данные.

Если в этот момент обнаружен новый стартовый бит, происходит переполнение данных. Флаг DOR установлен для указания того, что один или несколько кадров данных были потеряны между самым последним чтением UDR и следующей прочитанной UDR. Флаг DOR очищается, когда кадр данных успешно смещается из регистра сдвига в буфер приема.

Флаг ошибки четности PE указывает, что следующий кадр данных в буфере приема имеет ошибку четности при получении. Если четность не включена, PE очищается.

#### 17.11.5. Проверка четности приема

Установка бит контролия четности UPM1 начнет проверку четности. Режим проверки (четный или нечетный) определяется UPM0. После проверки четности верификатор вычисляет четность входных данных и сравнивает результат с четностью кадра данных. Результат проверки будет сохранен в буфере приема вместе с данными и стоповыми битами. CPU проверяет полученный фрейм на ошибки четности, читая бит PE.

Если следующие данные, считываемые из буфера приема, имеют ошибку четности и четность включена, UPE устанавливается и остается действительным до тех пор, пока не будет прочитана UDR буфера приема.

#### 17.11.6. Отключение приемника

Отключает приемник сразу по сравнению с передатчиком. Полученные данные будут потеряны. Когда приемник отключен (RXEN очищается), приемник больше не будет занимать вывод RxD, и буфер приема будет обновлен.

### 17.11.7. Асинхронный прием данных

У USART есть блок восстановления тактов и блок восстановления данных для обработки асинхронного приема данных. Логика восстановления тактов используется для синхронизации асинхронных последовательных данных, поступающих с выводов RxD с внутренним часом скорости передачи.

Логика восстановления данных используется для сбора данных и фильтрации каждого бита данных, вводимого через фильтр низких частот, чтобы улучшить устойчивость приемника к помехам.

Рабочий диапазон асинхронного приема зависит от точности внутренних тактовых импульсов в бодах, скорости ввода кадра и количества бит данных, содержащихся в кадре.

### 17.11.8. Асинхронный рабочий диапазон

Рабочий диапазон приемника зависит от степени несоответствия между принятой скоростью передачи данных и внутренней скоростью передачи.

Если передатчик передает данные с чрезвычайно быстрой или медленной скоростью передачи, или если скорость передачи, генерируемая внутренне приемником, не имеет такой же частоты, приемник не может синхронизироваться с стартовым битом.

Чтобы гарантировать, что приемник не пропустил выборку следующего бита начала кадра, скорость ввода данных и скорость передачи в бодах внутреннего приемника не могут быть слишком различны, а соотношение между ними используется для описания диапазона ошибок скорости передачи в бодах.

Следующие две таблицы дают максимальный диапазон ошибок в бодах, который разрешен в нормальном режиме и в режиме двойной скорости.

Максимальный диапазон ошибок передачи в бодах в нормальном режиме

Бит данных + длина бит четности	Максимальный диапазон ошибок (%)	Рекомендуемый диапазон ошибок (%)
5	+6J/-6.8	±3.0
6	+5.8/-5.9	±2.5
7	+5.1/-5.2	±2.0
8	+4.6/-4.5	±3.0
9	+4.1/-4.2	±1.5
10	+3.8/-3.8	±1.5

Максимальный диапазон ошибок передачи в бодах в режиме двойной скорости

Бит данных + длина бит четности	Максимальный диапазон ошибок (%)	Рекомендуемый диапазон ошибок (%)
5	+5.7/-5.9	±2.5
6	+4.9/-5.1	±2.0
7	+4.4/-4.5	±1.5
8	+3.9/-4.0	±1.5
9	+3.5/-3.6	±1.0
10	+3.2/-3.3	±1.0

Как видно из таблицы, скорость передачи в нормальном режиме допускает больший диапазон изменений. Предполагаемый выше допустимый диапазон ошибок в бодах предполагает, что приемник и передатчик вносят одинаковый вклад в максимальную общую ошибку.

Существует два возможных источника ошибки скорости передачи в приемнике.

Во-первых, стабильность часов системы приемника связана с рабочим напряжением и температурой. Эта проблема обычно не возникает при использовании кварцевого генератора для генерации системных часов, но системные часы могут отклоняться при использовании внутреннего генератора.

Вторая причина заключается в том, что генератор скорости передачи может не получить желаемую скорость передачи в бодах путем деления системных часов. На этом этапе вы можете отрегулировать значение UBRR, чтобы ошибка была как можно более низкой.

#### 17.11.9. Установка скорости передачи и ошибка ввода

Для стандартных частот кристалла и резонатора скорость передачи для фактической связи в асинхронном режиме может быть получена с помощью формулы расчета скорости в бодах.

Ошибка между скоростью передачи и общей скоростью обмена данными может быть следующей:

Формула для расчета:

$$\text{Ошибка [%]} = (\text{Baud real} / \text{Baud} - 1) * 100\%,$$

где Baud является часто используемой скоростью передачи данных, а Baud real - скорость передачи в бодах, вычисленная по формуле расчета и введенная в расчет скорости передачи.

Связь между ошибкой скорости в бодах и системными часами  $f_{sys}$  и значением UBRR регистра скорости передачи данных может быть получена по формуле:

Нормальный режим:

$$\text{Ошибка [%]} = (f_{sys} / (16 * (\text{UBRR} + 1)) / \text{Baud} - 1) * 100\%$$

Режим двойной скорости:

$$\text{Ошибка [%]} = (f_{sys} / (8 * (\text{UBRR} + 1)) / \text{Baud} - 1) * 100\%$$

Если ошибка тактов на обеих сторонах связи не учитывается, системные такты  $f_{sys}$  являются стандартными. Во время часов может быть получена связь между значением UBRR ошибки скорости передачи.

В следующей таблице показана ошибка скорости передачи в бодах для разных настроек UBRR в соответствии с системными часами 16 МГц.

Ошибка, вызванная установкой значения UBRR на системных часах 16 МГц

Скорость передачи (бит / с)	fsys = 16.000MHz			
	Нормальный режим (U2X = 0)		Режим двойной скорости (U2X = 1)	
	UBRR	Ошибка	UBRR	Ошибка
2400	416	-0.1%	832	0.0%
4800	207	0.2%	416	-0.1%
9600	103	0.2%	207	0.2%
14.4K	68	0.6%	138	-0.1%
19.2K	51	0.2%	103	0.2%
28.8K	34	-0.8%	68	0.6%
38.4K	25	2.1%	34	-0.8%
57.6K	16	0.2%	51	0.2%
76.8K	12	0.2%	25	0.2%
115.2K	8	-3.5%	16	2.1%
230.4K	3	8.5%	8	-3.5%
250K	3	0%	7	0%
0.5M	1	0%	3	0%
1M	0	0%	1	0%

## 17.12. Многопроцессорный режим связи

Настройка бит мультипроцессорной связи (MPCM) UCSRA фильтрует данные, полученные приемником USART. Кадры, которые не имеют адресной информации, будут игнорироваться и не будут сохранены в буфере приема.

В многопроцессорной системе каждый процессор взаимодействует по одной и той же последовательной шине. Эта фильтрация эффективно уменьшает количество кадров данных, которые должны обрабатываться процессором.

Установка бит MPCM не влияет на работу передатчика, но его использование отличается в многопроцессорных системах связи.

Если кадр данных, принятый приемником, имеет длину от 5 до 8 бит, первый стоповый бит используется для указания, содержит ли текущий кадр данные или информацию о адресе. Если получатель получает длину кадра данных из 9 бит, то 9-й бит определяет, будут ли данные данных или адресов. Если флаг типа фрейма равен «1», это адресный кадр, иначе он является фреймом данных.

В многопроцессорном режиме связи нескольким ведомым процессорам разрешено принимать данные от одного главного процессора.

Первый шаг - определить, какой ведомый процессор устанавливается на связь с ведущим, путем декодирования адресного кадра.

Обработанный подчиненный процессор обычно получает последующие данные, тогда как другие подчиненные процессоры игнорируют эти кадры данных до тех пор, пока не будет принят следующий адресный кадр.

Для ведущего процессора он может использовать 9-битный формат кадра данных и использовать 9-й бит данных для определения формата кадра. В этом режиме связи ведомый процессор также должен работать в формате 9-битного кадра данных.

Ниже приведены шаги для обмена данными в режиме многопроцессорной связи:

1. Все подчиненные процессоры работают в многопроцессорном режиме связи (MPCM);
  2. Основной процессор отправляет адресный кадр, и все подчиненные процессоры получают этот кадр. Ведомого процессор устанавливает бит RXC регистра UCSRA **нормально**.
  3. Каждое ведомое устройство считывает содержимое регистра UDR и декодирует адресный кадр, чтобы определить, выбрано ли оно. Если выбрано - бит MPCM в регистре UCSRA очищается, если не выбран - MPCM удерживается в «1» и ожидает прибытия следующего адресного фрейма.
  4. Адресованное ведомое устройство принимает все кадры данных до тех пор, пока не будет принят новый адресный фрейм. Непринятые подчиненные процессоры игнорируют эти кадры данных.
  5. После того как адресный подчиненный процессор принимает последний кадр данных, он устанавливает бит MPCM и ожидает прибытия следующего адресного кадра. Затем повтор со второго шага.

Возможные форматы кадров, содержат от 5 до 8 бит данных, но непрактичны, потому что приемник должен переключаться между форматами  $n$  и  $n+1$ .

Поскольку приемник и передатчик используют одну и ту же настройку длины символа, этот параметр затрудняет полнодуплексную работу. Если используется формат кадра от 5 до 8 бит данных, передатчик должен установить два стоповых бита, где первый стоповый бит используется для определения типа кадра.

## 17.13. Регистры Описание

## *UCSRA - Регистр A управления и состояния USART*

4	FE	Флаг ошибки кадра. Когда FE в «1», это означает, что данные, полученные буфером приема данных, имеют ошибку кадрирования, то есть первый стоповый бит в «0». Когда FE в «0», это указывает на то, что данные, полученные буфером данных приема, не имеют ошибки кадрирования, то есть первый стоповый бит в «1». После того, как FE в «1», он останется в силе до чтения UDR. При записи в UCSRA бит FE записывается в «0».
3	DOR	Флаг переполнения данных. Когда буфер приема заполнен ( <b>содержит два данных</b> ), в регистре сдвига приема есть данные. Если в это время будет обнаружен новый стартовый бит, произойдет переполнение данных, и будет установлен DOR. Он будет действовать до тех пор, пока не будет прочитана UDR. При записи в UCSRA бит DOR записывается в «0».
2	PE	Флаг ошибки четности. Когда контроль четности включен (UPM1 в «1»), а принятый кадр данных в буфере приема имеет ошибку четности, PE устанавливается в «1», и остается действительным до тех пор, пока не будет прочитана UDR. При записи в UCSRA бит PE записывается в «0».
1	U2X	Бит разрешения передачи с двойной скоростью. Когда U2X в «1», скорость передачи асинхронного режима связи удваивается. Когда U2X в «0», скорость передачи асинхронного режима связи является обычной. Этот бит действителен только в асинхронном режиме работы, этот бит очищается при использовании синхронного режима работы.
0	MPME	Бит разрешения режима мультипроцессорной связи. Установка бита MPME запускает многопроцессорный режим связи. Когда MPME в «1», входные кадры, полученные приемником USART, которые не содержат адресную информацию, будут игнорироваться. Настройки передатчика не зависят от настроек MPME.

## UCSRB - Регистр В управление и состояния USART

## UCSRB - Регистр В управление и состояния USART

2	UCSZ2	Длина символа контролирует бит 2. UCSZ2 объединяется с UCSZ1, UCSZ0 регистра UCSRC для установки количества бит данных в кадре данных.
1	RXB8	Прием 8-го бита данных. Когда длина кадра данных составляет 9 бит, RXB8 является наивысшим битом полученных данных. Прочтите RXB8 перед чтением нижних 8-битных данных, содержащихся в UDR.
0	TXB8	Передача 8-го бита данных. Когда длина кадра данных составляет 9 бит, TXB8 является наивысшим битом передаваемых данных. Напишите TXB8 перед записью более низких 8-битных данных, содержащихся в UDR

### UCSRC - Регистр С управление и состояния USART

UCSRB - Регистр С управление и состояния USART									
Адрес: 0xC2				Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0	
Имя	UMSEL1	UMSEL0	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Бит	Имя	Определение бит							
7:6	UMSEL1:0	Биты выбора режима USART. UMSEL выбор синхронный или асинхронный режим работы.							
		UMSEL	Режим						
		0	Асинхронный режим USART						
		1	Синхронный режим USART						
		2	Режим работы ведомого устройства SPI						
5:4	UPM1:0	Биты выбора четности. Высокий бит UPM1 выбирает включение или отключение контроля четности, а низкий бит UPM0 выбирает нечетные или четные четности.							
		UPM1:0	Режим						
		0	Блокирование паритета						
		1	Зарезервировано						
		2	Включить четность						
3	USBS	Выбор стоп-биты. Количество бит: USBS = 0 – 1 стоп-бит; USBS = 1 – 2 стоп-бита							
		UCSZ1:0	Биты выбора длины символа кадра данных. UCSZ1:0 объединяется с UCSZ2 регистра UCSRB, чтобы установить количество бит данных, содержащихся в кадре данных.						
			UCSZ2:0	Длина кадра данных					
			0	5 бит					
			1	6 бит					
			2	7 бит					
			3	8 бит					
			4	Зарезервировано					
			5	Зарезервировано					
0	UCPOL	Бит выбора полярности бит. В режиме синхронной работы USART UCOPOL устанавливает взаимосвязь между изменением выходных данных и синхронизацией тактов синхронизации ХСК входных данных. Очистите этот бит независимо от UCOPOL в асинхронном режиме работы							
		UCPOL	Изменение передачи данных				Выборка данных		

		0	Положительный фронт ХСК	Отрицательный фронт ХСК
		1	Отрицательный фронт ХСК	Положительный фронт ХСК

*UBRRL - Младший байт скорости передачи данных USART*

*UBRRH - Старший байт скорости передачи данных USART*

UBRRH - Старший байт скорости передачи данных USART									
Адрес: 0xC5				Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0	
Имя	-	-	-	-	UBRR11	UBRR10	UBRR9	UBRR8	
R/W	-	-	-	-	R/W	R/W	R/W	R/W	
Бит	Имя	Определение бит							
7:4	-	Зарезервировано							
3:0	UBRR11:8	Старший байт регистра скорости передачи данных USART. Регистр скорости передачи USART содержит UBRRL и UBRRH, которые используются вместе, чтобы установить скорость передачи в бодах. UBRR = {UBRR [11: 8], UBRRL}							
		Режим работы			Расчет скорости передачи				
		Асинхронный нормальный режим			BAUD = f sys /(16*(UBRR+1))				
		Асинхронный режим скорости			BAUD = f sys /(8*(UBRR+1))				
		Режим синхронного ведущего			BAUD = f sys /(2*(UBRR+1))				

## *UDR - Регистр данных USART*

## 18. USART0 - Режим работы SPI

- Полнодуплексный режим, 3-проводная синхронная передача данных
- Операция ведущего или ведомого устройства
- Поддерживает все четыре режима работы (режимы 0, 1, 2 и 3)
- Первый - Младший бит (LSB) или Старший бит (MSB) при Передачи данных (настраиваемый порядок данных)
- Операция очереди (двойной буфер)
- Генератор скорости передачи данных с высоким разрешением

### 18.1. Обзор

Когда бит UMSEL1 USCRC установлен в «1», режим работы SPI включен и представлен USPI. Этот SPI-модуль имеет трехпроводный режим SPI. По сравнению с четырехпроводным режимом SPI ему не хватает линии выбора подчиненного устройства, а остальные три линии идентичны.

USPI занимает ресурсы USART, включая регистры сдвига и приема сдвига и приема, а также генератор скорости передачи. Логика генерации четности и проверки, логика восстановления данных и тактов недействительны. Адрес регистров управления и состояния одинаковый, но функция битов регистра будет изменяться по мере необходимости в режиме SPI.

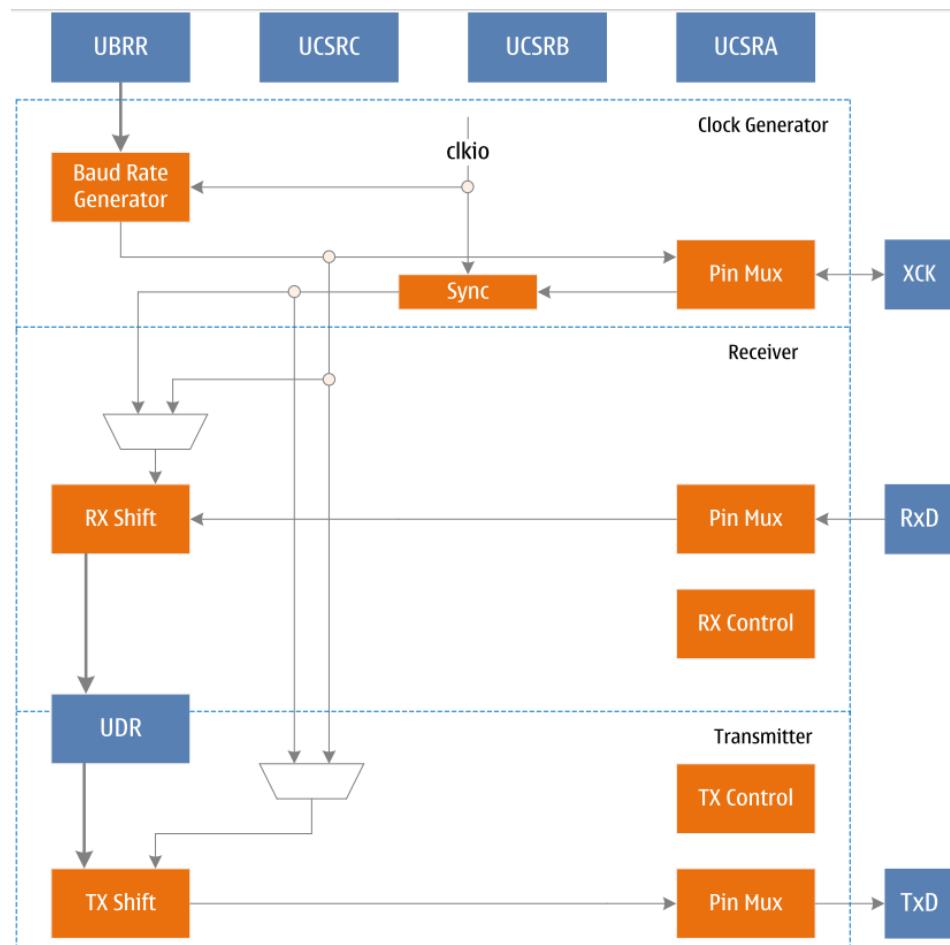


Рис. Блок-схема USART в режиме работы SPI

## 18.2. Генерация тактов

Когда SPI работает в режиме ведомого, ему необходимо предоставить такты для связи и мультиплексировать генератор скорости передачи данных USART для генерации этих тактов. Эти такты выводятся из вывода XCK, поэтому регистр направления данных (DDR\_XCK) на выводе XCK должен быть установлен в «1».

Тактовая частота определяется по следующей формуле:

$$\text{BAUD} = f_{\text{sys}} / (2 * (\text{UBRR} + 1))$$

Когда SPI работает в ведомом режиме, такты связи подаются внешним ведущим и вводятся с помощью выводов XCK, поэтому данные вывода XCK Регистр направления (DDR\_XCK) должен быть установлен в «0».

## 18.3. Режимы и синхронизация данных SPI

SPI имеет четыре комбинации фаз синхронизации и полярности, которые определяются управляющими битами UCPHA и UCPOL. Конкретные элементы управления показаны в следующей таблице и на следующем рисунке:

Режимы работы SPI

SPI Режим	Состояния	Передний фронт	Задний Фронт
0	UCPOLn = 0, UCPHAn = 0	Выборка (Повышение)	Установка (Падение)
1	UCPOLn = 0, UCPHAn = 1	Установка (Повышение)	Выборка (Падение)
2	UCPOLn = 1, UCPHAn = 0	Выборка (Падение)	Установка (Повышение)
3	UCPOLn = 1, UCPHAn = 1	Установка (Падение)	Выборка (Повышение)

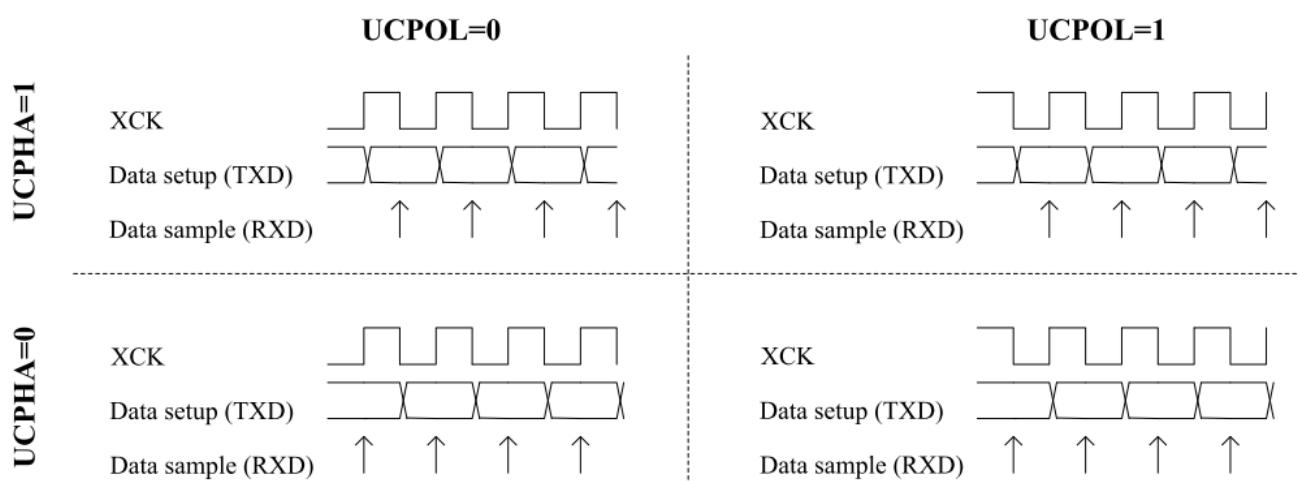


Рис. UCPHAn и UCPOLn временные диаграммы передачи данных.

## 18.4. Формат кадра.

Серийный кадр SPI может начинаться с наименее значимого бита или самого значимого бита и заканчивается самым значительным битом или наименее значимым битом в общей сложности 8 бит. После окончания кадра может быть

передан новый кадр, и после завершения передачи линия данных может быть установлена в состояние бездействия.

## 18.5. Передача данных.

SPI устанавливает бит TXEN регистра UCSRB в «1» для включения передатчика, а вывод TxD используется передатчиком для отправки данных последовательного выхода. В настоящее время приемник не может быть включен. SPI устанавливает бит RXEN в регистре UCSRB в «1» для включения приемника, а приемник RxD используется приемником для приема последовательных входных данных. Передатчик должен быть включен в это время. Оба SPI передают и принимают использование XCK в качестве тактового сигнала передачи.

Сначала SPI должен быть инициализирован. Процесс инициализации обычно включает в себя установку скорости передачи, настройку порядка передачи бит данных кадра и включение приемника или передатчика по мере необходимости.

Для операции SPI, управляемой прерываниями, флаг глобальной прерывания очищается при инициализации, и все прерывания SPI отключены.

При повторной инициализации, такой как изменение скорости передачи или структуры кадра, необходимо обеспечить отсутствие передачи данных. Флаг TXC может использоваться для определения того, завершил ли передатчик все передачи. Флаг RXC может использоваться для определения того, не были ли какие-либо данные в буфере приема не прочитаны. Если для этой цели используется бит флага TXC, флаг TXC должен быть очищен перед каждой передачей данных (до того, как будет записан регистр UDR).

После инициализации SPI данные могут быть записаны в регистр UDR для начала передачи данных. Поскольку передатчик управляет тактовым импульсом передачи, обе передающие и приемные данные работают как таковые.

Когда регистр сдвига передачи готов к отправке нового кадра данных, передатчик переместит данные, записанные в регистр UDR из буфера передачи, в регистр сдвига передачи и отправит его.

Чтобы обеспечить синхронизацию входного буфера с данными передачи, регистр UDR должен считываться один раз после того, как каждый бит данных передан. Когда происходит переполнение данных, самые последние полученные данные будут потеряны вместо самых старых полученных данных.

### 18.5.1. Передача, флаги и обработка прерываний

Передатчик SPI имеет два флага: SPI Data Register Empty Flag UDRE и Transfer Complete Flag TXC. Оба флага могут генерировать прерывания.

Флаг пустого регистра данных UDRE используется для указания того, можно ли в буфер отправки записывать новые данные. Этот бит установлен на «1», когда буфер пуст и «0», когда он заполнен. Когда бит UDRE равен «1», CPU может записывать новые данные в UDR регистра данных и наоборот. **Когда бит запрета прерывания регистра данных**, UDRIE, в регистре UCSRB равен «1», запрос пустого прерывания регистра SPI будет генерироваться всякий раз, когда UDRE установлен (и глобальное прерывание включено). Запись в регистр UDR очистит UDRE.

Когда данные передаются в режиме прерывания, в UDR должны быть

записаны новые данные, чтобы очистить UDRE или регистр данных. Пустое прерывание должно быть записано в процедуре обслуживания пустых прерываний регистра данных. В противном случае, как только процедура обслуживания прерывания закончится, новое прерывание будет сгенерировано снова. Флаг TXC конца передачи устанавливается, когда весь кадр данных смещается из регистра сдвига передачи и в регистре передачи нет новых данных.

Когда бит разрешения завершения передачи прерывания TXCIE (и разрешения глобального прерывания) на UCSRB установлен в «1», полное прерывание передачи SPI будет выполняться с установленным флагом TXC.

После того, как введена процедура обслуживания прерываний, флаг TXC автоматически очищается, а CPU также может записать «1» на этот бит, чтобы очистить его.

### 18.5.2. Отключение передатчика

После сброса TXEN передатчик не сможет отключить его до тех пор, пока все данные не будут переданы. То есть, нет данных для передачи в регистре сдвига передачи и в регистре регистров передачи. После того, как передатчик отключен, вывод TxD возобновляет функцию универсального ввода-вывода общего назначения.

## 18.6. Прием, флаг полного буфера и обработка прерываний

У приемника SPI есть флаг: флаг завершения приема, RXC, указывает, есть ли какие-либо непрочитанные данные в буфере приема.

Когда в буфере приема есть непрочитанные данные, этот бит равен «1», в противном случае «0».

Если приемник отключен, буфер приема обновляется и RXC также очищается. После того, как в UCSRB RXCIE установлен в «1» - разрешения прерываний, полное прерывание получения SPI генерируется до тех пор, пока установлен флаг RXC (и глобальное прерывание включено).

При получении данных в режиме прерывания процедура обслуживания прерывания приема данных должна считывать данные из UDR, чтобы очистить флаг RXC, или новое прерывание будет сгенерировано сразу же после завершения обработчика прерываний.

### 18.6.1. Отключить приемник

Приемник отключается сразу же по сравнению с передатчиком. Полученные данные будут потеряны. Отключить приемник (RXEN clear После нулевого значения) приемник больше не будет занимать вывод RxD, и буфер приема будет обновлен.

## 18.7. Регистры Описание

## *Список регистра USART SPI*

Регистр	Адрес	Значение по умолчанию	Описание
UCSRA	0xC0	0x20	Регистр управления и состояния USPI A
UCSRB	0xC1	0x00	Регистр управления и состояния USPI B
UCSRC	0xC2	0x06	Регистр управления и состояния USPI C
UBRRRL	0xC4	0x0	Младший байт скорости передачи данных USPI
UBRRRH	0xC5	0x0	Старший байт скорости передачи данных USPI
UDR	0xC6	0x0	Регистр данных USPI

#### *UCSRA - Регистр A управления и состояния USART SPI*

#### *UCSRB - Регистр B управления и состояния USART SPI*

5	UDRIE	Регистр данных пуст - бит разрешения прерываний. Включить прерывание: UDRE = «1» и отключить прерывания: UDRE = «0». Когда UDRIE в «1» и глобальное прерывание включено, прерывания Регистр данных USART пуст, могут быть сгенерированы, когда в регистре UCSRA бит UDRE в «1».
4	RXEN	Прием - бит разрешения. Приемник USART активируется после установки. Функция IO общего назначения вывода RxD заменяется на прием USART. Отключение приемника приведет к сбросу буфера приема и аннулированию флагов FE, DOR и PE.
3	TXEN	Передача - бит разрешения. Передатчик USART активируется после утверждения. Функция IO общего назначения вывода TxD заменяется USART. После удаления TXEN передача USART может быть отключена только после завершения всех передач данных.
2:0	-	Зарезервировано

## *UCSRC - Регистр С управления и состояния USART SPI*

UCSRB - Регистр С управление и состояния USART

Адрес: 0xC2			Значение по умолчанию: 0x00										
Бит	7	6	5	4	3	2	1	0					
Имя	UMSEL1	UMSEL0	-	-	-	DORD	UCRPA	UCPOL					
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W					
Бит	Имя	Определение бит											
7:6	UMSEL1:0	Биты выбора режима USART. UMSEL выбор синхронный или асинхронный режим работы.											
		UMSEL	Режим										
		0	Асинхронный режим USART										
		1	Синхронный режим USART										
		2	Режим работы ведомого устройства SPI										
5:3	-	Зарезервировано											
2	DORD	Бит контроля порядка передачи данных. Если бит DORD в «1», в передачи данных LSB передается первой. При установке бит DORD в «0», в передачи данных MSB передается первой.											
1	UCRPA	Выбор фазы синхронизации. UCRPA Выбор выборки данных происходит в начале или в конце.											
		UCRPH	Изменение передачи данных					Выборка данных					
		0	Положительный фронт ХСК					Отрицательный фронт ХСК					
0	UCPOL	1	Отрицательный фронт ХСК					Положительный фронт ХСК					
		Выбор полярности тактов. UCPOL выбирает изменения данных, и образцы возникают на восходящих или падающих фронтах.											
		UCPOL	Изменение передачи данных					Выборка данных					
1	UCPOL	0	Положительный фронт ХСК					Отрицательный фронт ХСК					
		1	Отрицательный фронт ХСК					Положительный фронт ХСК					

*UBRRL - Младший байт скорости передачи данных USART SPI*

**UBRR0 - младший байт скорости передачи данных USART**

Бит	Имя	Определение бит
7:0	UBRR7:0	Младший байт регистра скорости передачи данных USART. Регистр скорости передачи USART содержит UBRRRL и UBRRRH, которые используются вместе, чтобы установить скорость передачи в бодах.

### *UBRRH - Старший байт скорости передачи данных USART*

UBRRH - Старший байт скорости передачи данных USART								
Адрес: 0xC5				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	-	-	-	-	UBRR11	UBRR10	UBRR9	UBRR8
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:4	-	Зарезервировано						
3:0	UBRR11:8	Старший байт регистра скорости передачи данных USART. Регистр скорости передачи USART содержит UBRRRL и UBRRRH, которые используются вместе, чтобы установить скорость передачи в бодах. UBRR = {UBRR [11: 8], UBRRRL}						
Режим работы				Расчет скорости передачи				
Режим ведомого				Скорость передачи определяется ведущим				
Режим ведущего				BAUD = f sys /(2*(UBRR+1))				

### *UDR - Регистр данных USART SPI*

UCSRB - Регистр данных USART								
Адрес: 0xC6				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	UDR7	UDR6	UDR5	UDR4	UDR3	UDR2	UDR1	UDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	UDR	Регистр данных передатчика и приемника USART. Буфер передачи данных USART и буфер приема данных совместно используют регистр UDR данных USART. Запись данных в UDR записывается в буфер данных передачи, а чтение данных из UDR считывает буфер данных приема. В режиме с 5-8 бит-кадрами неиспользуемый 9-й бит игнорируется передатчиком, а приемник устанавливает их в 0. Буфер передачи может быть записан только тогда, когда флаг UDRE регистра UCSRA равен «1», иначе операция передатчика будет ошибочной. Когда регистр сдвига передачи пуст, передатчик загружает данные в буфер передачи в регистр сдвига передачи, и данные выводятся последовательно с вывода TxD. Буфер приема содержит двухуровневый FIFO. После считывания буфера приема FIFO изменяет свое состояние.						

## 19. TWI - двухпроводная последовательная шина (I2C)

- Простой и мощный и гибкий коммуникационный интерфейс требует всего 2 провода.
- Поддерживает ведущее и ведомое устройство.
- Устройство может работать в режиме передатчика или приемника.
- Разрешено 7-битное адресное пространство. 128 ведомых устройств
- Поддержка арбитража с несколькими ведущими устройствами
- Скорость передачи данных до 400 Кбит/с

- Полностью программируемый ведомый адрес и общий адрес
- Пробуждение при совпадении адреса в спящем режиме

## 19.1. Введение TWI Шина

Двухпроводный последовательный интерфейс TWI хорошо подходит для типичных процессорных приложений. Протокол TWI позволяет разработчикам системы подключать 128 различных устройств, используя только две двунаправленные линии передачи.

Две линии - это такты SCL и SDA данных. Внешняя аппаратура требует только двух подтягивающих резисторов на каждой линии. Все устройства, подключенные к шине, имеют свой собственный адрес. Протокол TWI решает проблему арбитража шины.

## 19.2 Терминология TWI

В этом разделе часто появляются следующие определенные термины:

- Ведущее Устройство, которое запускает и останавливает передачу.
- Ведущий также отвечает за генерацию Тактов SCL.
- Ведомое Устройство, к которым адресуются ведущие устройства
- Передатчики, которые размещают данные на шине.
- Приемные устройства, которые принимают данные с шины

## 19.3. Электрическое подключение

Как показано на рисунке ниже, оба провода интерфейса TWI подключаются к положительному источнику питания через подтягивающие резисторы. Драйверы шины для всех TWI-совместимых устройств являются открытыми коллекторами, что позволяет использовать линию и функции интерфейса.

Когда выход TWI-устройства равен «0», шина TWI генерирует низкий уровень.

Когда все выходы устройства TWI в третьем состоянии, шина позволяет подтягивающим резисторам установить высокое напряжение.

Для обеспечения всех операций шины все устройства, подключенные к шине TWI, должны быть включены.

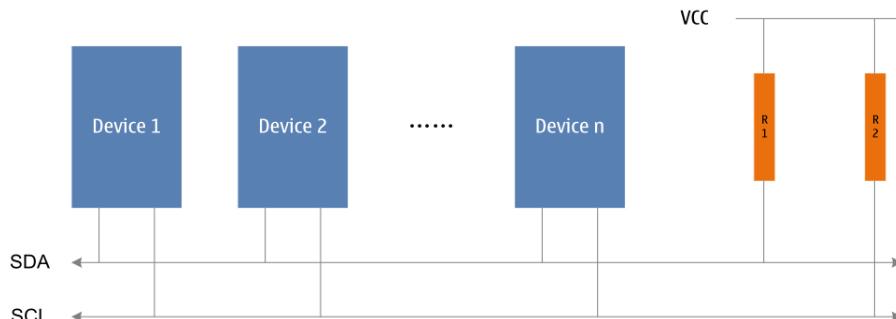


Рис.19.1. Схема подключения шины TWI

## 19.4. Передача данных и структура кадра

Каждый бит передачи данных на шине TWI синхронизируется с часами. Когда линия синхронизации высока, уровень в строке данных должен оставаться стабильным, если только он не должен генерировать условие начала или остановки.

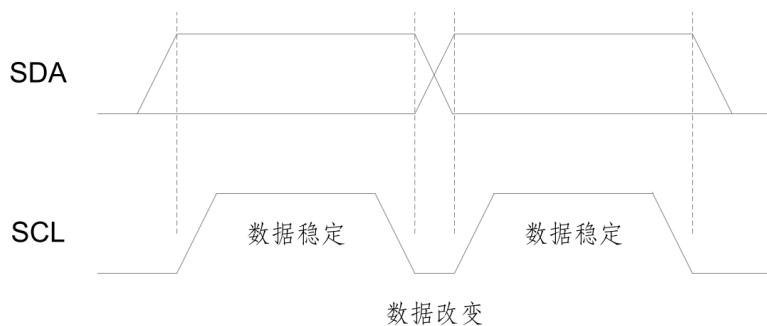


Рис.19.2. Диаграмма достоверности данных TWI

### 19.4.1 Состояние START и STOP

Передачи TWI запускаются и останавливаются ведущим. Ведущий отправляет статус START на шине для отправки данных и статус STOP, чтобы остановить передачу данных. Между состояниями START и STOP шина считается занятой и не позволяет другим ведущим пытаться управлять шиной.

Существует специальный случай, который разрешает только новое состояние START между состояниями START и STOP. Это называется состоянием REPEATED START и применяется, когда текущий ведущий инициирует новую передачу, не отказываясь от управления шиной. Шина по-прежнему считается занятой до следующего STOP после REPEATED START. Это согласуется с START, поэтому в этом документе используются START и REPEATED START, если нет специального объяснения.

Как показано на рисунке ниже, условия START и STOP изменяют уровень линии SDA, когда линия SCL высока.

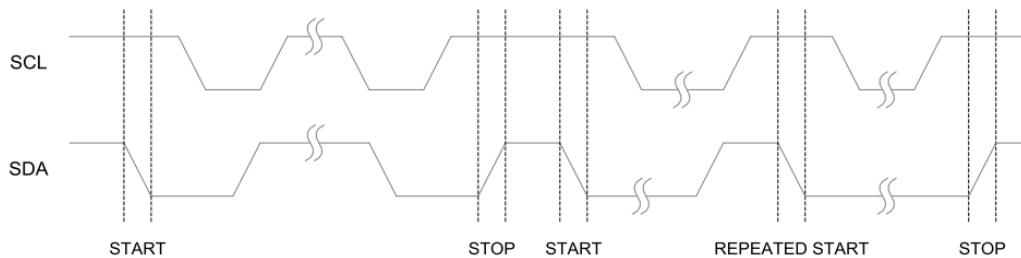


Рис.19.3. Состояния START, REPEATED START и STOP

### 19.4.2. Формат адресного пакета.

Все пакеты адресов, передаваемые по шине TWI, представляют собой 9-битную длину данных и состоят из 7-битного адреса, 1-битового контрольного бита READ / WRITE и 1-битового бита подтверждения. Когда бит READ / WRITE равен

«1», выполняется операция чтения, когда бит READ / WRITE равен «0», выполняется операция записи.

После того, как ведомое устройство получила свой адресован, оно должна подтвердить на линии SDA низким уровнем в течение 9-го цикла SCL (ACK). Если ведомое устройство занято или нет причин для ответа ведущему, держит линию SDA на высоком уровне во время цикла ACK. Затем Ведущий может отправить условие STOP или REPEATED START, чтобы перезапустить передачу.

Пакет адресов включает в себя адрес ведомого и бит управления чтением или записью, представленные SLA + R или SLA + W, соответственно. Сначала бит MSB байта адреса. За исключением того, что зарезервированные адреса «00000000» зарезервированы для широковещательных вызовов, и все адреса в формате «1111xxxx» должны быть зарезервированы для будущего использования, другие подчиненные адреса могут свободно назначаться разработчиком.

Когда происходит широковещательный вызов, все подчиненные устройства должны отвечать, ставить линию SDA на низкий уровень в течение периода ACK.

Функция широковещания может использоваться, когда ведомому необходимо отправить одну и ту же информацию нескольким ведомым устройствам.

После того, как общий адрес вызова плюс бит WRITE отправляется на шину, все подчиненные устройства, которые должны отвечать на общий вызов. Все эти ведомые устройства, отвечающие на широковещательный вызов, получат следующий пакет данных.

Следует отметить, что не имеет смысла посыпать адрес широковещательного вызова плюс бит READ, потому что произойдет столкновение шины, если несколько подчиненных устройств одновременно отправляют разные данные.

Формат адресного пакета показан ниже:

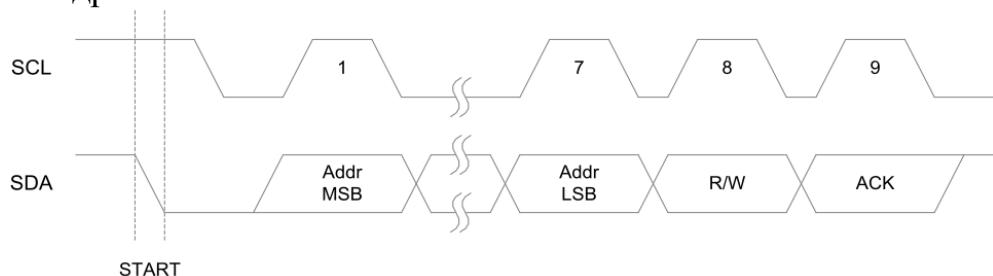


Рис.19.4. Диаграмма формата адресного пакета в формате TWI

#### 19.4.3. Формат пакета данных

Все пакеты, передаваемые по шине TWI, представляют собой 9-битную длину данных и состоят из 1 байт данных и 1 бит подтверждения. Во время передачи данных ведущий отвечает за генерацию тактовых импульсов SCL передачи и состояний START и STOP. Передатчик передает данные байта, которые должны быть переданы, и приемник генерирует ответ приема.

Сигнал подтверждения ACK генерируется приемником в течение 9-го цикла SCL (ACK), устанавливает линию SDA на низкий уровень. Если приемник удерживает линию SDA в верхнем уровне во время цикла ACK, посыпается неподтвержденный сигнал NACK.

Когда получатель получил последний байт или больше не может получать

какие-либо данные по какой-либо причине, отправитель должен быть уведомлен, отправив NACK после получения последнего байта.

Сначала передается бит MSB байта данных.

Формат пакета показан на рисунке ниже:

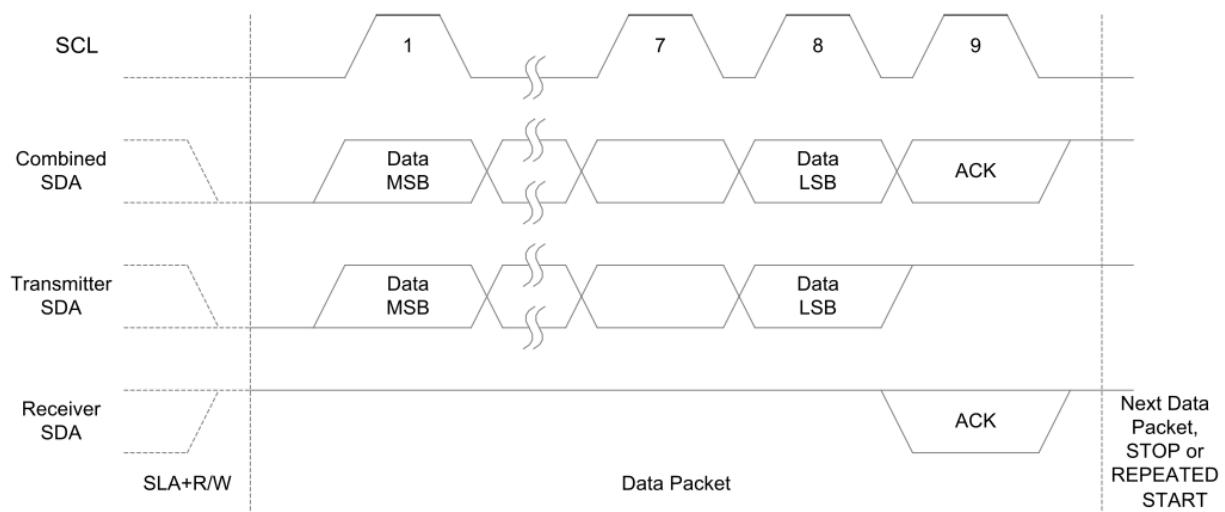


Рис.19.5. Диаграмма формата пакета данных TWI

Комбинированная передача адресов и передачи данных, передача в основном состоит из 1 START, 1 SLA + R / W, 1 или более пакетов данных и 1 STOP. Только пустые сообщения START и STOP являются незаконными.

Вы можете использовать линии и функции линии SCL для реализации рукопожатий ведущий - ведомый.

Ведомое устройство может продлить период низкого уровня SCL. Эта функция полезна, когда ведущий устанавливает тактовую частоту намного быстрее, чем ведомый, или если подчиненному устройству требуется дополнительное время для обработки данных.

Ведомое устройство, расширяющее низкий период SCL, не влияет на высокий цикл SCL. Он все еще определяется хозяином. Можно видеть, что ведомое устройство может уменьшить скорость передачи данных TWI, изменив рабочий цикл SCL.

На следующем рисунке показана типичная передача данных.

Обратите внимание, что через SLA + R / W и STOP можно передавать несколько байтов в зависимости от протокола реализации прикладного программного обеспечения.

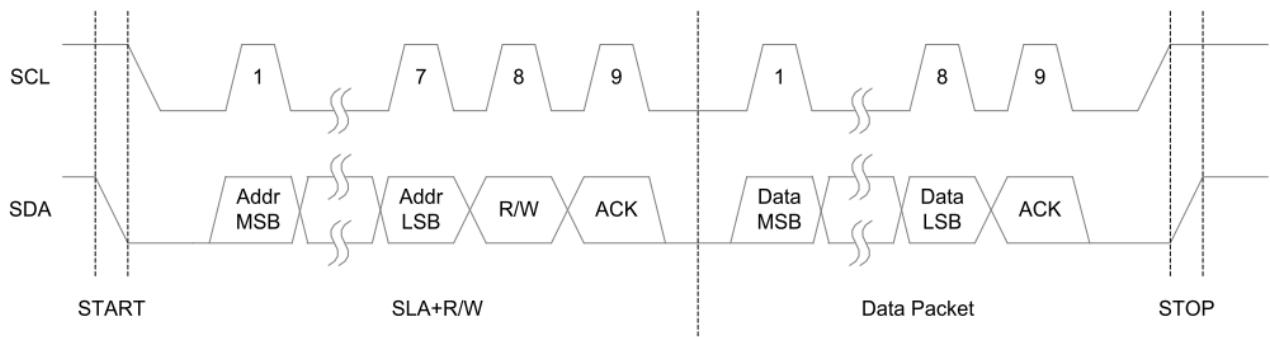


Рис.19.6. Типичная передача TWI

#### 19.4.4. Системы с несколькими хостами и их арбитраж и синхронизация

Протокол TWI позволяет нескольким ведущим на шине и использует специальные меры, чтобы гарантировать, что даже два или более хоста могут инициировать передачи одновременно с нормальными передачами.

Система Multi-хост будет два вопроса:

1. Реализация алгоритма позволяет только один хост, несколько хостов, чтобы завершить передачу. Когда другие хозяева считают, что они теряют право на выбор должен прекратить передачу. Процесс выбора называется арбитражем. Когда конкурирующие мастера арбитраж нашел сбой, адрес должен перейти непосредственно к хосту, является ли это управление режима исполнители шины для обнаружения. На самом деле машина множественного хоста не должна быть обнаружена с самого начала передачи в то же время, то есть уничтожения данных, передаваемый не допускаются на шине.

2. различные хости могут использовать разные частоты SCL. Для обеспечения последовательного перевода, вы должны разработать последовательные тактовые синхронизируются программами хоста. Это облегчит процесс арбитража. Линия шины и функция для решения вышеуказанных проблем. Все хости тактовые линии для получения композиции с часами высокой времени, равный мастер-часы во всех кратчайшим, равную его низком уровне во всех самый длинный хост-часы. Все хости прослушивания SCL, когда объединено SCL часы является высоким или низким, соответственно, они могут эффективно начать переливаться соответствующий высокий и низкий периоды SCL. SCL мульти-мастер механизм синхронизации часов, как показано ниже:

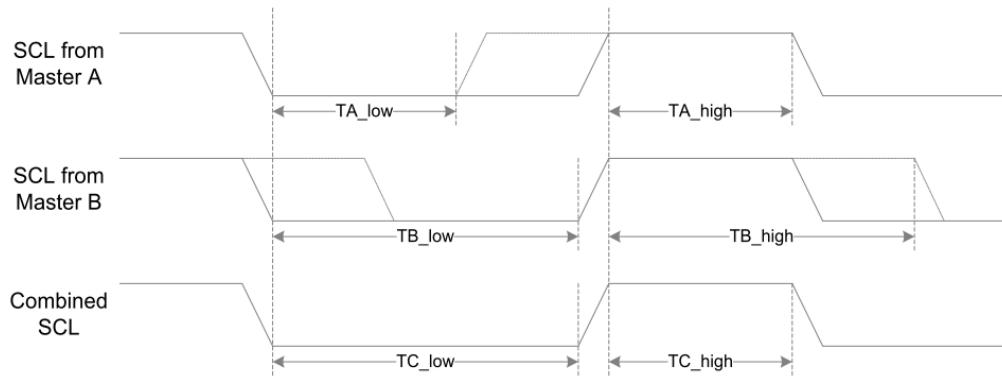


Рис.19.7. Временная диаграмма синхронизации синхронизации SCM

После выходных данных все ведомые продолжают отслеживать линию SDA для достижения арбитража. Если значение, считанное из SDA, не соответствует значению, выводимому с ведомого, ведомый теряет арбитраж.

Следует отметить, что ведомый выдаст высокий SDA, а другой ведомый потеряет арбитраж, если он выдает низкий SDA. Ведомый, который потерял арбитраж, должен немедленно переключиться на подчиненный режим и определить, если он адресован. Ведомый, который теряет арбитраж, должен установить высокий уровень на линии SDA, но он также может генерировать тактовый сигнал до того, как закончится текущий пакет данных или адресов.

Арбитраж будет продолжаться до тех пор, пока в системе не останется только один хост, который может занимать несколько бит. Если несколько мастеров

обращаются к одному подчиненному устройству, арбитраж будет продолжать пакет.

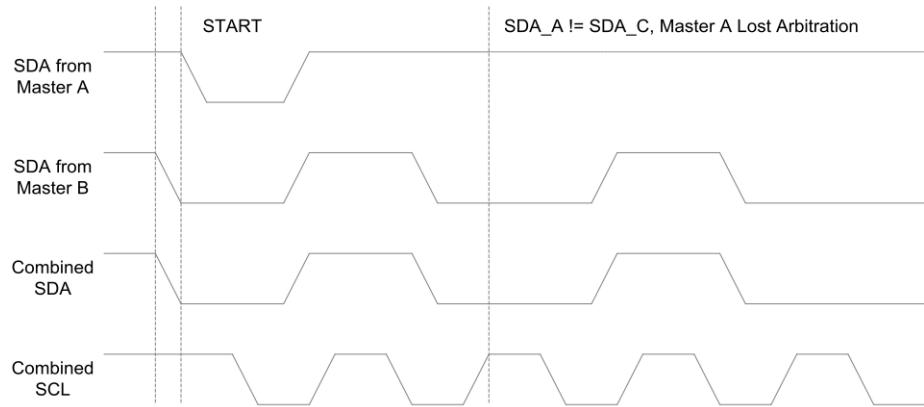


Рис.19.8. Арбитраж между двумя хостами

- Обратите внимание, что арбитраж не допускается в следующих ситуациях: между состоянием REPEATED START и битом данных;
- между состоянием STOP и битом данных;
- между состоянием REPEATED START и STOP;

В прикладном программном обеспечении должны учитываться вышеуказанные обстоятельства и обеспечивать, чтобы эти незаконные ситуации арбитража не возникали. Это означает, что в системе с несколькими ведущими все передачи данных должны состоять из одинаковых SLA + R / W и пакетов данных. Другими словами, все переводы должны содержать одинаковое количество пакетов, иначе результаты арбитража не могут быть определены.

## 19.5 Обзор модуля TWI

Структура модуля TWI показана на рисунке ниже.

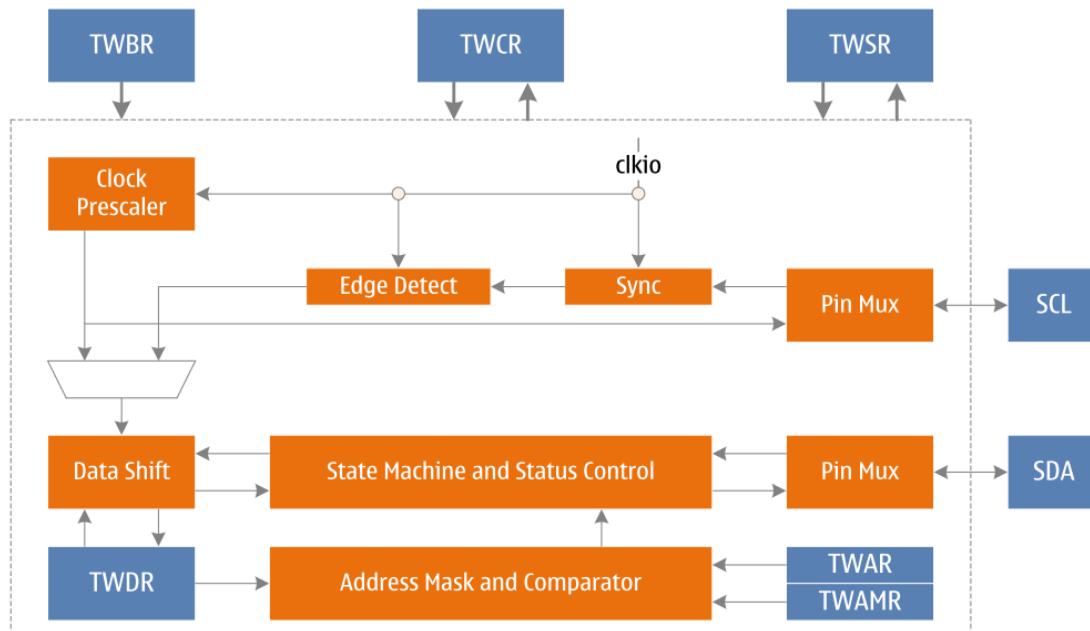


Рис.19.9 Структура модуля TWI

Модуль TWI в основном включает в себя генератор скорости передачи битов, блок интерфейса шины, компаратор адресов и блок управления. См. Следующее подробное описание.

### 19.5.1. Блок генератора скорости передачи данных

Блок генератора скорости передачи данных в основном управляет тактовым циклом SCL в режиме ведущего. Период синхронизации SCL определяется битами управления предварительным делителем в регистре TWB бит TWI и регистром состояния TWI TWSR.

Работа ведомого устройства не зависит от настроек скорости передачи данных или предварительного делителя, но убедитесь, что рабочие такты ведомого устройства по меньшей мере в 16 раз превышают частоту SCL.

Обратите внимание, что ведомое устройство может продлить низкий период SCL, тем самым уменьшая среднюю тактовую частоту шины TWI.

Для генерации тактовой частоты SCL используется следующая формула:

$$f_{SCL} = f_{sys} / (16 + 2 * TWBR * 4 * TWPS)$$

где TWBR - это значение регистра скорости передачи TWI, а TWPS - бит управления предварительного делителя в регистре состояния TWI.

### 19.5.2. Блок интерфейса шины

Блок интерфейса шины состоит из TWDR регистра данных и адресов, контроллера START / STOP и аппаратной схемы арбитражного решения.

TWDR содержит адрес или байты данных, которые должны быть отправлены, или полученный адрес или байты данных. В дополнение к 8-разрядному TWDR модуль интерфейса шины также содержит регистр ACK / NACK, который отправляется или принимается.

Этот ACK / NACK-регистр не может быть напрямую доступен прикладному программному обеспечению. При получении данных он может быть установлен или очищен TWI-регистром TWI.

Когда данные передаются, принятное значение ACK / NACK отражается значением TWS в регистре состояния TWI TWSR.

Контроллер START / STOP генерирует и обнаруживает состояния START, REPEATED START и STOP.

Когда MCU находится в некоторых режимах сна, контроллер START / STOP все еще может обнаруживать состояния START и STOP и разбудить MCU в спящем режиме, когда адресат передается на шину TWI.

Если TWI инициирует передачу данных в режиме ведущего, схема обнаружения арбитража будет продолжать контролировать шину, чтобы определить, все ли она контролирует шину.

Когда модуль TWI теряет управление шиной, блок управления выполнит правильное действие и сформирует соответствующий код состояния для информирования MCU.

### 19.5.3. Блок согласования адресов

Блок согласования адресов используется для проверки - соответствует ли полученный адресный байт 7-битовому адресу в регистре адресов TWI.

Когда бит TWI Radio Call Identification Enable включен (TWGCE) в регистре TWAR установлен, адрес, полученный от шины, также сравнивается с широковещательным адресом. Как только совпадение адреса будет успешным, блок управления выполнит правильное действие.

Модуль TWI может отвечать или не отвечать на адресацию ведущего, в зависимости от настройки регистра TWCR. Даже в спящем режиме блок согласования адресов может сравнивать адреса и, если он адресован ведущим нашине, пробуждает MCU из спящего режима.

### 19.5.4. Блок управления

Блок управления отвечает за прослушивание шины и отвечает в соответствии с настройками TWCR. Флаг прерывания TWI, TWINT, будет установлен, когда нашине TWI возникает событие, требующее участия прикладного программного обеспечения. В следующем такте будет обновлен регистр состояния TWI TWSR, чтобы указать код состояния события. Когда TWINT = «1», TWSR содержит точную информацию о состоянии. В других случаях TWSR представляет собой специальный код состояния, указывающий, что нет точной информации о состоянии. После того, как установлен флаг TWINT, линия SCL удерживается на низком уровне, приостанавливая передачу TWI пошине и позволяя прикладному программному обеспечению обрабатывать событие.

Флаг TWINT будет установлен в следующих ситуациях:

- TWI после отправки START / REPEATED START
- TWI после отправки SLA + R / W
- После того, как TWI передал адресный байт
- Bus Третий арбитраж TWI не прошел
- TWI адресуется хостом (режим подчиненного адреса или широковещательный режим)
- После обращения к ведомому после получения STOP или REPEATED START
- **Шина.** Если ошибка шины, вызванная незаконным состоянием START или STOP

## 19.6. Применение TWI

Интерфейс TWI байт-ориентирован и основан на прерываниях. Все события шины, такие как получение байта или отправка сигнала START, генерируют прерывание TWI. Поскольку TWI основан на прерываниях, прикладное программное обеспечение может свободно выполнять другие операции во время передачи байта TWI. Бит разрешения прерывания TWI в регистре TWCR, TWIE и бит разрешения глобального прерывания совместно определяют, генерируется ли прерывание TWI при установке флага TWINT.

Если бит TWIE очищается, прикладное программное обеспечение должно

использовать флаг TWINT для опроса шины TWI.

Когда установлен флаг TWINT, он указывает, что интерфейс TWI завершил текущую операцию и ожидает, когда прикладное программное обеспечение ответит. В этом случае регистр TWI содержит код состояния, который отражает текущий статус шины.

Прикладное программное обеспечение может устанавливать регистры TWCR и TWDR, чтобы определить, как должен работать интерфейс TWI во время следующего цикла шины TWI.

На следующем рисунке показан пример подключения приложения к интерфейсу TWI. В этом примере ведущий ожидает отправить один байт данных в ведомый. Описание здесь очень просто, в следующей главе будет более подробный экран.

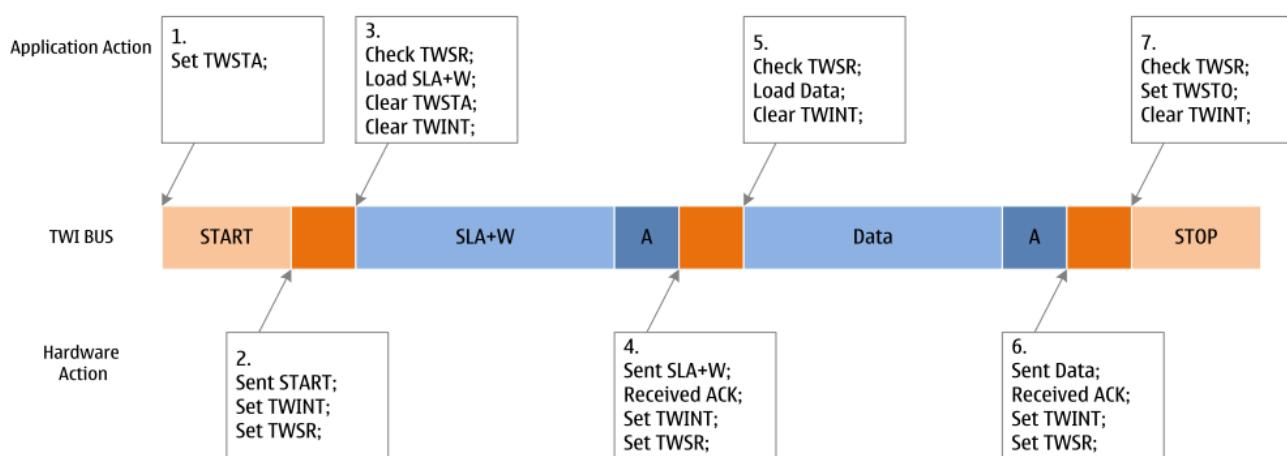


Рис.19.10. Типичная схема процесса передачи данных TWI

Процесс передачи TWI, показанный на рисунке, следующий:

1. Первым шагом в передаче TWI является отправка START. Запишите определенное значение в регистр TWCR, чтобы проинструктировать оборудование TWI для отправки сигнала START. Письменное значение будет подробно описано ниже. **Важно установить TWINT в письменном значении.** Запись «1» в бит TWINT очищает бит. TWI не запускает никаких операций, пока бит TWINT в регистре TWCR установлен. Как только программное обеспечение очистит бит TWINT, модуль TWI немедленно инициирует передачу сигнала START.

2. Когда будет отправлен START, будет установлен флаг TWCR TWINT и TWSR будет обновлен до нового кода состояния, указывающего, что сигнал START был успешно отправлен.

3. Приложение просматривает значение TWSR и определяет, что статус START был успешно отправлен. Если TWSR показывает другие значения, приложение может выполнять специальные операции, такие как вызов обработчика ошибок. Когда определено, что код состояния соответствует ожиданию, программа загружает значение SLA + W в регистр TWDR. Регистр TWDR может использоваться как по адресам, так и по данным. Затем программное обеспечение записывает определенное значение в регистр TWCR, который инструктирует оборудование TWI отправлять значение SLA + W в TWDR. Записанное значение будет описано ниже. Установите TWINT в письменном значении, чтобы очистить флаг TWINT. TWI не запускает никаких операций, пока бит TWINT в регистре

TWCR установлен. После того как программное обеспечение очистит бит TWINT, модуль TWI немедленно инициирует передачу адресного пакета.

4. После отправки адресного пакета будет установлен флаг TWCR TWINT, а TWSR будет обновлен до нового кода состояния, указывающего, что пакет адресов был успешно отправлен. Код состояния также отражает, отвечает ли подчиненный на адресный пакет.

5. Приложение рассматривает значение TWSR и определяет, что пакет адресов был успешно отправлен, а принятый ACK - ожидаемое значение. Если TWSR показывает другие значения, приложение может выполнять специальные операции, такие как вызов обработчика ошибок. Когда определено, что код состояния соответствует ожиданиям, программа загружает значение данных в регистр TWDR. Затем программное обеспечение записывает определенное значение в регистр TWCR, который инструктирует оборудование TWI отправлять значение данных в TWDR. Письменное значение будет подробно описано ниже. Установите TWINT в письменном значении, чтобы очистить флаг TWINT. TWI не запускает никаких операций, пока бит TWINT в регистре TWCR установлен. Когда программное обеспечение очищает бит TWINT, модуль TWI немедленно инициирует передачу пакета.

6. Когда пакет данных отправлен, флаг TWCR TWINT будет установлен, и TWSR будет обновлен с новым кодом состояния, чтобы указать, что пакет данных был успешно отправлен. Код состояния также отражает, отвечает ли ведомый на пакет.

7. Приложение рассматривает значение TWSR, чтобы определить, что пакет был успешно отправлен, а принятый ACK - ожидаемое значение. Если TWSR показывает другие значения, приложение может выполнять специальные операции, такие как вызов обработчика ошибок. Когда определено, что код состояния соответствует ожиданию, программное обеспечение записывает определенное значение в регистр TWCR, чтобы дать указание аппарату TWI отправить сигнал STOP. Записанное значение будет описано ниже. Установите TWINT в письменном значении, чтобы очистить флаг TWINT. TWI не запускает никаких операций, пока бит TWINT в регистре TWCR установлен. Когда программное обеспечение очищает бит TWINT, модуль TWI немедленно инициирует передачу сигнала STOP. Следует отметить, что TWINT не будет установлен после отправки сигнала STOP.

Хотя пример относительно прост, он содержит все правила в процессе передачи данных TWI.

Сводка такова:

- TW Флаг TWINT устанавливается, когда TWI завершает операцию и ожидает обратной связи от приложения. Линия синхронизации SCL будет удерживаться на низком уровне до тех пор, пока TWINT не будет очищен.
- Когда установлен флаг TWINT, пользователь должен обновить значения всех TWI-регистров, связанных со следующим циклом шины TWI. Например, регистр TWDR должен загрузить значение, которое будет отправлено для следующего цикла шины.
- **Применение.** После обновления всех регистров и выполнения других необходимых операций одновременно приложение записывает регистр TWCR. При записи TWCR бит TWINT должен быть установлен, чтобы

очистить флаг TWINT. После того, как TWINT будет очищен, TWI начнет выполнение операции, установленной TWCR.

### 19.6.1. Режим передачи

TWI может работать в следующих 4 основных режимах: Host Transmitter (MT), Host Receiver (MR), ведомый передатчик (ST) и ведомый приемник (SR).

В одном приложении можно использовать несколько режимов. Например, TWI может использовать режим MT для записи данных в TWI EEPROM и использовать режим MR для чтения данных из EEPROM.

Если в системе есть другие ведущие, а некоторые могут также отправлять данные в TWI, используется режим SR. Решайте, какой режим использовать.

Эти режимы будут подробно описаны ниже.

При передаче данных в каждом режиме возможные коды состояния описываются в сочетании с изображениями.

Эти фотографии содержат следующие сокращения:

S: начальный статус

Rs: статус REPEATED START

R: индикатор операции чтения (SDA высокий)

W: Запись флага операции (SDA является низким)

A: бит подтверждения (SDA низкий)

NA: нет ответного бита (высокий уровень SDA)

Данные: 8 битов данных

P: статус STOP

SLA: адрес подчиненного устройства

Окружность на рисунке используется для указания того, что флаг TWINT установлен, а число в круге указывает код состояния в регистре TWSR, где бит управления предварительным делителем маскируется до «0». В этих местах приложение должно выполнить соответствующие действия для продолжения или завершения передачи TWI. Передача TWI будет приостановлена до тех пор, пока флаг TWINT не будет очищен.

Когда флаг TWINT установлен, код состояния в TWSR используется для определения соответствующей операции программного обеспечения.

Подробные сведения о программных операциях и последующих последовательных передачах, требуемых для каждого кода состояния, приведены в таблицах.

Обратите внимание, что биты управления предварительным делителем в TWSR в таблице маскируются до «0».

### 19.6.2. Режим передачи ведущего

В режиме передачи хоста TWI отправляет определенное количество байтов данных подчиненному приемнику. Чтобы войти в режим хоста, необходимо отправить сигнал START. Следующий формат пакета адресов определяет, входит ли TWI в режим хост-отправителя или режим приемника хоста. Если отправляется SLA + W, вводится режим передачи хоста. Если отправляется SLA + R, вводится режим приема хоста. Коды состояния, упомянутые в этом разделе, предполагают,

что бит управления предварительным делителем равен «0».

Сигнал START выдается путем записи в регистр **TWCR** следующего значения:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	X	1	0	X	1	0	X

Бит TWEN должен быть установлен на «1», чтобы включить интерфейс TWI, TWSTA в «1», чтобы отправить сигнал START, и TWINT на «1», чтобы очистить флаг TWINT.

Модуль TWI обнаруживает состояние шины и посыпает сигнал START, как только шина свободна.

После отправки START аппаратное обеспечение устанавливает флаг TWINT и обновляет код состояния TWSR до 0x08.

Чтобы войти в режим отправки ведущего, необходимо отправить SLA + W.

Это можно сделать с помощью следующей операции.

Сначала запишите SLA + W в регистр TWDR, затем напишите «1» в бит TWINT и очистите флаг TWINT, чтобы продолжить передачу.

То есть, напишите следующее значение в регистр TWCR для отправки SLA+W:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	X	0	0	X	1	0	X

Когда передача SLA + W завершена и принимается сигнал ответа, TWINT снова устанавливается и обновляется код состояния TWSR.

Возможные коды состояния: 0x18, 0x20 или 0x38. Соответствующий ответ под каждым кодом состояния подробно описан в таблице кодов состояния.

Когда SLA + W отправляется успешно, он может начать отправлять пакеты. Это можно сделать, записав данные в регистр TWDR. TWDR может записываться только при высоком значении TWINT. В противном случае доступ игнорируется и устанавливается бит конфликта записи: TWWC. После обновления TWDR запись «1» в бит TWINT очищает флаг TWINT для продолжения передачи.

То есть, введите следующие данные в регистр TWCR для отправки данных:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	X	0	0	X	1	0	X

Когда пакет данных передается и принимается сигнал ответа, TWINT снова устанавливается и обновляется код состояния TWSR.

Возможные коды состояния: 0x28 или 0x30. Соответствующий ответ под каждым кодом состояния подробно описан в таблице кодов состояния.

После успешной передачи данных пакет данных может продолжать отправляться. Этот процесс повторяется до тех пор, пока не будет отправлен последний байт. Ведущий будет генерировать сигнал STOP, или сигнал REPEATED START завершит всю передачу.

Сигнал STOP выдается путем записи в регистр TWCR следующего значения:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	X	0	1	X	1	0	X

Выполнить сигнал REPEATED START, введя в регистр TWCR следующее значение:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	X	1	0	X	1	0	X

После отправки REPEATED START (код состояния 0x10) интерфейс TWI может снова получить доступ к тому же подчиненному устройству или получить доступ к новому подчиненному устройству без отправки сигнала STOP.

REPEATED START позволяет ведущему переключаться между различными ведомыми устройствами, ведущим-отправителем и режимами принимающего приемника без потери контроля над шиной.

Коды состояния в режиме передачи хоста и соответствующие операции показаны в следующей таблице:

*Таблица кодов состояния для режимов передачи ведущего*

Код состояния	Состояние шины и оборудования	Ответ прикладного программного обеспечения					Следующий шаг в аппаратном обеспечении	
		Чтение / запись TWDR	Работа на TWCR					
			STA	STO	TWINT	TWEA		
0x08	Отправлен START	Загрузить SLA + W	0	0	1	X	SLA + W будет отправлено: ACK или NACK получены;	
0x10	Отправлен REPEATED START	Загрузить SLA + W	0	0	1	X	SLA + W будет отправлено: ACK или NACK получены;	
		Загрузить SLA + R	0	0	1	X	SLA + W будет отправлено: ACK или NACK получены; Перейдет в режим MR	
0x18	SLA + W отправлено, получено ACK	Загрузка данных	0	0	1	X	Посыпает данные, будет получать ACK или NACK	
		Нет операции	1	0	1	X	Послать REPEATED START	
		Нет операции	0	1	1	X	Отправляет STOP, сброс флага TWSTO	
		Нет операции	1	1	1	X	Отправляет STOP, сброс флага TWSTO; START отправлен	
0x20	SLA + W отправлено, получено NACK	Загрузка данных	0	0	1	X	Посыпает данные, будет получать ACK или NACK	
		Нет операции	1	0	1	X	Послать REPEATED START	
		Нет операции	0	1	1	X	Отправляет STOP, сброс флага TWSTO	
		Нет операции	1	1	1	X	Отправляет STOP, сброс флага TWSTO; START отправлен	
0x28	Байт данных отправлен.	Загрузка данных	0	0	1	X	Посыпает данные, будет получать ACK или NACK	

	Получен ACK	Нет операции	1	0	1	X	Послать REPEATED START
		Нет операции	0	1	1	X	Отправляет STOP, сброс флага TWSTO
		Нет операции	1	1	1	X	Отправляет STOP, сброс флага TWSTO; START отправлен
0x30	Байты данных отправлены. Получен NACK	Загрузка данных	0	0	1	X	Посыпает данные, будет получать ACK или NACK
		Нет операции	1	0	1	X	Послать REPEATED START
		Нет операции	0	1	1	X	Отправляет STOP, сброс флага TWSTO
		Нет операции	1	1	1	X	Отправляет STOP, сброс флага TWSTO; START отправлен
0x38	Отказ SLA+W или арбитраж данных	Нет операции	0	0	1	X	Выпустит шину, войдет в неадресный подчиненный режим
		Нет операции	1	0	1	X	Отправляет START в режиме простоя

Формат и состояния режима передачи ведущего приведены ниже:

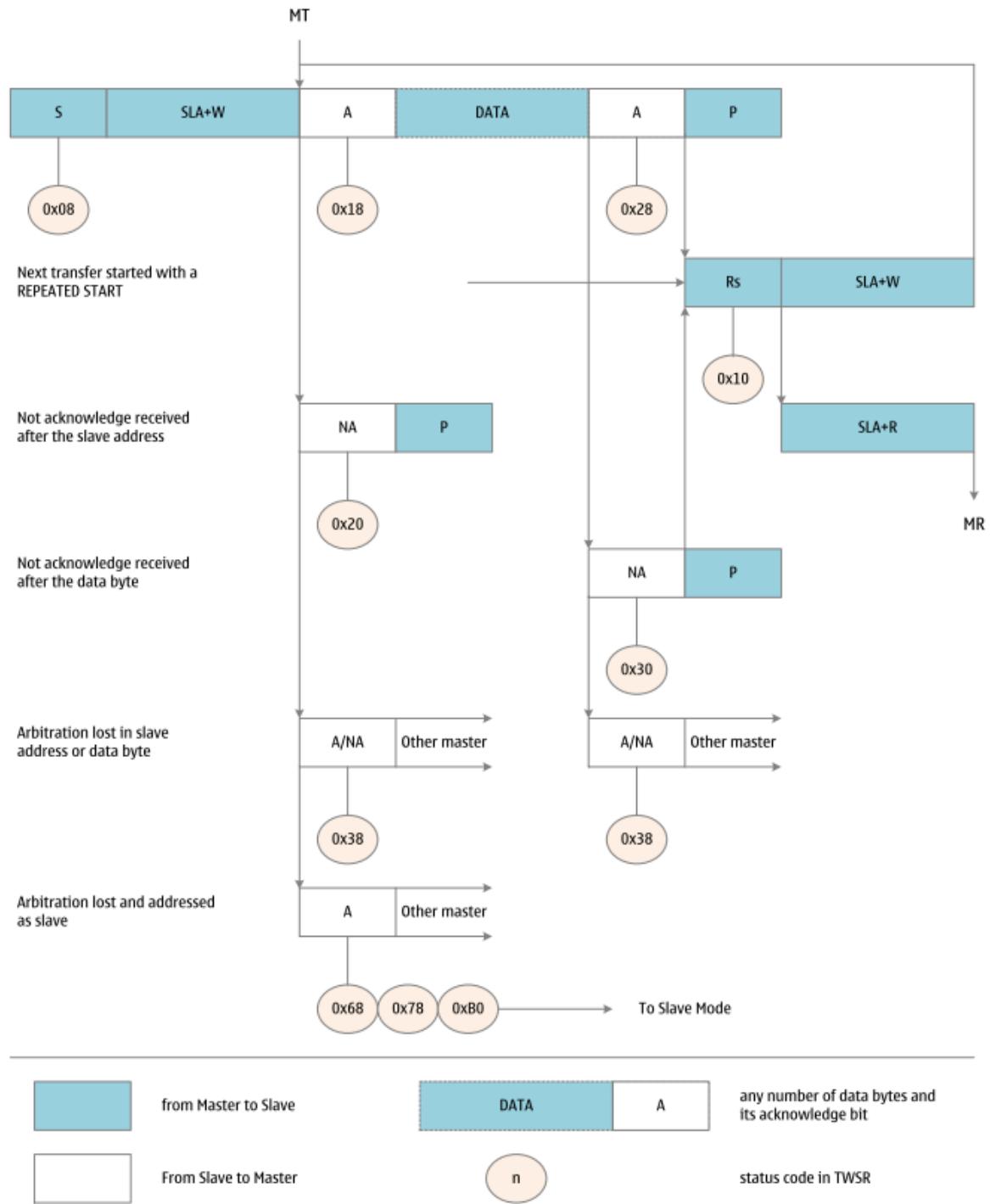


Рис.19.11. Формат режима передачи и диаграмма состояний

### 19.6.3. Режим приема ведущего

В режиме приема ведущего TWI получает определенное количество байтов данных от ведомого передатчика. Чтобы войти в режим ведущего, необходимо отправить сигнал START.

Следующий формат пакета адресов определяет, входит ли TWI в режим ведущего-отправителя или режим приемника ведущего. Если отправляется SLA + W, вводится режим передачи хоста. Если отправляется SLA + R, вводится режим приема хоста. Коды состояния, упомянутые в этом разделе, предполагают, что бит управления предварительным делителем равен «0».

Сигнал START выдается путем записи в регистр TWCR следующего значения:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	X	1	0	X	1	0	X

Бит TWEN должен быть установлен в «1», чтобы включить интерфейс TWI, TWSTA в «1», чтобы отправить сигнал START, и TWINT в «1», чтобы очистить флаг TWINT.

Модуль TWI обнаруживает состояние шины и посыпает сигнал START, как только шина свободна. После отправки START аппаратное обеспечение устанавливает флаг TWINT и обновляет код состояния TWSR до 0x08. Чтобы войти в режим приема ведомого, необходимо отправить SLA + R.

Это можно сделать с помощью следующей операции. Сначала запишите SLA + R в регистр TWDR, затем напишите «1» в бит TWINT и очистите флаг TWINT, чтобы продолжить передачу. То есть, напишите следующее значение в регистр TWCR для отправки SLA + R:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	X	0	0	X	1	0	X

Когда передача SLA + R завершена и принимается сигнал ответа, TWINT снова устанавливается и обновляется код состояния TWSR. Возможные коды состояния: 0x38, 0x40 или 0x48.

Соответствующий ответ под каждым кодом состояния подробно описан в таблице кодов состояния.

Когда SLA + R успешно передается, он может начать прием пакетов. Прием продолжается, очищая бит флага TWINT, записывая «1» в бит TWINT. То есть, запись следующего значения в регистр TWCR начинает прием:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	X	0	0	X	1	0	X

Когда пакет данных принимается и передается ответный сигнал, TWINT снова устанавливается, и код состояния TWSR обновляется. Возможные коды состояния: 0x50 или 0x58.

Соответствующий ответ под каждым кодом состояния подробно описан в таблице кодов состояния.

После успешного получения данных вы можете продолжать получать пакеты данных. Этот процесс повторяется до тех пор, пока не будет получен последний байт.

После приема последнего байта мастер должен отправить сигнал подтверждения NACK на ведомый передатчик. Хост принимает сигнал STOP или сигнал REPEATED START завершает прием.

Сигнал STOP выдается путем записи в регистр TWCR следующего значения:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	X	0	1	X	1	0	X

Выполните сигнал REPEATED START, введя в регистр TWCR следующее значение:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	X	1	0	X	1	0	X

После отправки REPEATED START (код состояния 0x10) интерфейс TWI может снова получить доступ к тому же ведущему или получить доступ к новому ведущему без отправки сигнала STOP.

REPEATED START позволяет ведущему переключаться между различными ведомыми устройствами, ведущими-отправителями и режимами принимающего приемника без потери контроля над шиной.

Коды состояния в режиме приема ведущего и соответствующие операции показаны в следующей таблице:

*Таблица кодов состояния для режима приема ведущего*

Код состояния	Состояние шины и оборудования	Ответ прикладного программного обеспечения					Следующий шаг в аппаратном обеспечении	
		Чтение / запись TWDR	Работа на TWCR					
			STA	STO	TWINT	TWEA		
0x08	Отправлен START	Загрузить SLA + W	0	0	1	X	SLA + W будет отправлено: ACK или NACK получены;	
0x10	Отправлен REPEATED START	Загрузить SLA + W	0	0	1	X	SLA + W будет отправлено: ACK или NACK получены;	
		Загрузить SLA + R	0	0	1	X	SLA + W будет отправлено: ACK или NACK получены; Перейдет в режим MR	
0x38	Отказ SLA+R или арбитраж данных	Нет операции	0	0	1	X	Выпустит шину, войдет в неадресный подчиненный режим	
		Нет операции	1	0	1	X	Отправляет START в режиме простоя	
0x40	SLA + R отправлено, получено ACK	Нет операции	0	0	1	0	Получат данные, NACK будет отправлен	
		Нет операции	0	0	1	1	Получат данные, ACK будет отправлен	
0x48	SLA + R отправлено, получено NACK	Нет операции	1	0	1	X	Послать REPEATED START	
		Нет операции	0	1	1	X	Отправляет STOP, сброс флага TWSTO	
		Нет операции	1	1	1	X	Отправляет STOP, сброс флага TWSTO; START отправлен	
0x50	Байт данных получен, ACK отправлен	Загрузка данных	0	0	1	0	Получат данные, NACK будет отправлен	
		Нет операции	0	0	1	1	Получат данные, ACK будет отправлен	
0x58	Получен байт данных, отправлен NACK	Нет операции	1	0	1	X	Послать REPEATED START	
		Нет операции	0	1	1	X	Отправляет STOP, сброс флага TWSTO	
		Нет операции	1	1	1	X	Отправляет STOP, сброс флага TWSTO;	

Формат и состояние принимающего режима, как показано ниже:

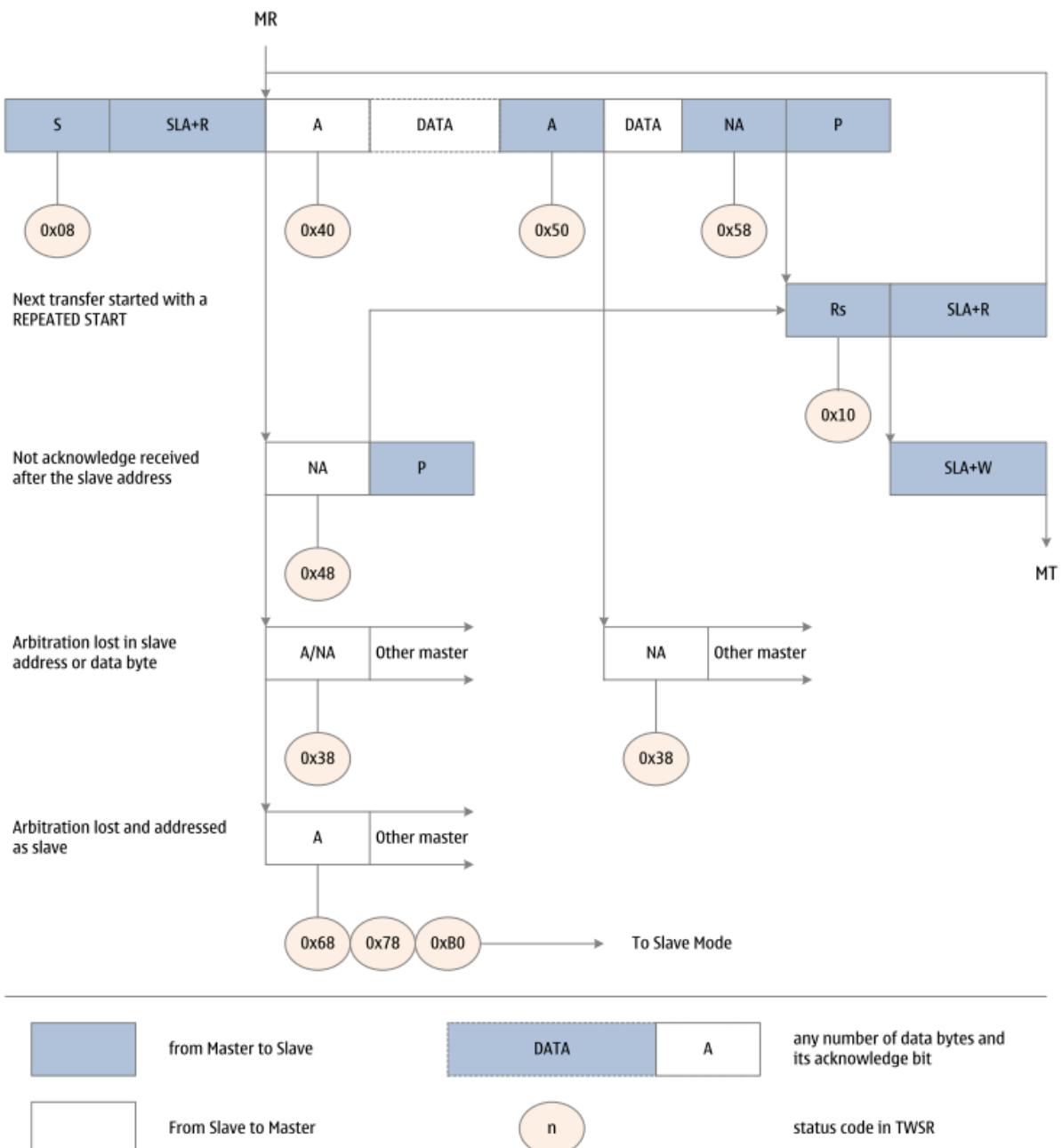


Рис.19.12. Формат принимающего режима и диаграмма состояний

#### 19.6.4. Режим ведомого приемника

В режиме ведомого приемника из передатчика хоста может быть получено определенное количество байтов данных. Коды состояния, упомянутые в этом разделе, предполагают, что бит управления предварительным делителем равен «0».

Чтобы запустить режим приема ведомого, установите регистры TWAR и TWCR.

TWAR необходимо установить следующим образом:

Старшие 7 бит TWAR - это ведомый адрес, на который будет реагировать интерфейс TWI при обращении к узлу. Если установлен LSB, TWI будет отвечать на адрес широковещательного вызова (0x00), в противном случае общий адрес вызова будет проигнорирован.

TWCR необходимо установить следующим образом:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
0	1	0	0	0	1	0	X

TWEN должен быть настроен на включение интерфейса TWI. TWEA должен быть настроен на возврат подтверждения ACK, когда ведущий адресован (адрес ведомого или широковещательный вызов) сам по себе. TWSTA и TWSTO должны быть очищены.

После инициализации TWAR и TWCR интерфейс TWI ожидает, пока не будет адресован собственный адрес подчиненного устройства (или широковещательный адрес).

Когда бит направления данных, следующий за ведомым адресом, равен «0» (указывает на операцию записи), TWI переходит в режим приема ведомого.

Когда бит направления данных равен «1» (с указанием операции чтения), TWI переходит в режим подчиненной передачи.

После получения собственного подчиненного адреса и символа операции записи установлен флаг TWINT, а действительный код состояния обновляется до TWSR.

Соответствующий ответ под каждым кодом состояния подробно описан в таблице кодов состояния.

Следует отметить, что режим приема подчиненного устройства также может быть введен после завершения арбитража TWI в режиме ведущего (см. Коды состояния 0x68 и 0x78).

Если бит TWEA сбрасывается во время передачи, TWI вернет NACK (высокий уровень) в линию SDA после получения байта. Это можно использовать для указания того, что ведомый не может получать больше данных.

Когда бит TWEA равен «0», TWI не будет отвечать на собственный адрес подчиненного устройства. Тем не менее, TWI все еще слушает шину.

После установки TWEA он может возобновить распознавание адреса и ответить. Другими словами, вы можете использовать TWEA для временной изоляции интерфейса TWI от шины.

**В других режимах ожидания, чем в режиме ожидания, такты интерфейса TWI могут быть отключены.**

Если режим приема ведомого включен, интерфейс будет продолжать отвечать на подчиненный адрес или широковещательный адрес, используя такты шины. Совпадение адресов пробудит MCU.

Во время пробуждения интерфейс TWI будет удерживать SCL до тех пор, пока флаг TWINT не будет очищен. Когда такты интерфейса TWI вернутся в нормальное состояние, он может получить больше данных.

Код состояния режима подчиненного приема отображается в следующей таблице:

Таблица кодов состояния режима подчиненного приема

Код состояния	Состояние шины и оборудования	Ответ прикладного программного обеспечения					Следующий шаг в аппаратном обеспечении	
		Чтение / запись TWDR	Работа на TWCR					
			STA	STO	TWINT	TWEA		
0x60	Получено SLA+W; ACK отправлено	Нет операции	X	0	1	0	Получат данные, NACK будет отправлен	
		Нет операции	X	0	1	1	Получат данные, ACK будет отправлен	
0x68	Арбитраж не удался при отправке SLA+R/W; получено SLA+W; отправлено ACK	Нет операции	X	0	1	0	Получат данные, NACK будет отправлен	
		Нет операции	X	0	1	1	Получат данные, ACK будет отправлен	
0x70	Полученный широковещательный адрес, отправленный ACK	Нет операции	X	0	1	0	Получат данные, NACK будет отправлен	
		Нет операции	X	0	1	1	Получат данные, ACK будет отправлен	
0x78	Арбитраж не удался при отправке SLA+R/W; получено SLA+W; отправлено ACK	Нет операции	X	0	1	0	Получат данные, NACK будет отправлен	
		Нет операции	X	0	1	1	Получат данные, ACK будет отправлен	
0x80	Собственные данные, полученные ACK	Чтение данных	X	0	1	0	Получат данные, NACK будет отправлен	
		Чтение данных	X	0	1	1	Получат данные, ACK будет отправлен	
0x88	Получены данные о себе, отправлен NACK	Чтение данных	0	0	1	0	Переключится на неактивный ведомый режим, не ответит на подчиненный адрес и трансляцию	
		Чтение данных	0	0	1	1	Переключится на неактивный ведомый режим, ответит на подчиненный адрес, TWGCE = 1 ответит на трансляцию	
		Чтение данных	1	0	1	0	Перейдет в неадресный ведомый режим, не будет отвечать на ведомый адрес и широковещательную рассылку, отправит START, когда шина свободна	
		Чтение данных	1	0	1	1	Переключится на неактивный режим ведомого, ответит на ведомый адрес, TWGCE = 1 ответит на	

							широковещательную передачу, шина отправит START в режиме ожидания
0x90	Получены широковещательные данные, отправлен NACK	Чтение данных	X	0	1	0	Получат данные, NACK будет отправлен
		Чтение данных	X	0	1	1	Получат данные, ACK будет отправлен
0x98	Получены широковещательные данные, отправлен NACK	Чтение данных	0	0	1	0	Переключится на неактивный ведомый режим, не ответит на подчиненный адрес и трансляцию
		Чтение данных	0	0	1	1	Переключится на неактивный ведомый режим, ответит на подчиненный адрес, TWGCE = 1 ответит на трансляцию
		Чтение данных	1	0	1	0	Перейдет в неадресный ведомый режим, не будет отвечать на ведомый адрес и широковещательную рассылку, отправит START, когда шина свободна
		Чтение данных	1	0	1	1	Переключится на неактивный режим ведомого, ответит на ведомый адрес, TWGCE = 1 ответит на широковещательную передачу, шина отправит START в режиме ожидания
0xA0	Получен STOP или REPEATED START при работе с ведомыми	Нет операции	0	0	1	0	Переключится на неактивный ведомый режим, не ответит на подчиненный адрес и трансляцию
		Нет операции	0	0	1	1	Переключится на неактивный ведомый режим, ответит на подчиненный адрес, TWGCE = 1 ответит на трансляцию
		Нет операции	1	0	1	0	Перейдет в неадресный ведомый режим, не будет отвечать на ведомый адрес и широковещательную рассылку, отправит START, когда шина свободна
		Нет операции	1	0	1	1	Переключится на неактивный режим ведомого, ответит на ведомый адрес, TWGCE = 1 ответит на широковещательную передачу, шина отправит START в режиме ожидания

Диаграмма формата и состояния режима подчиненного приема выглядит

следующим образом:

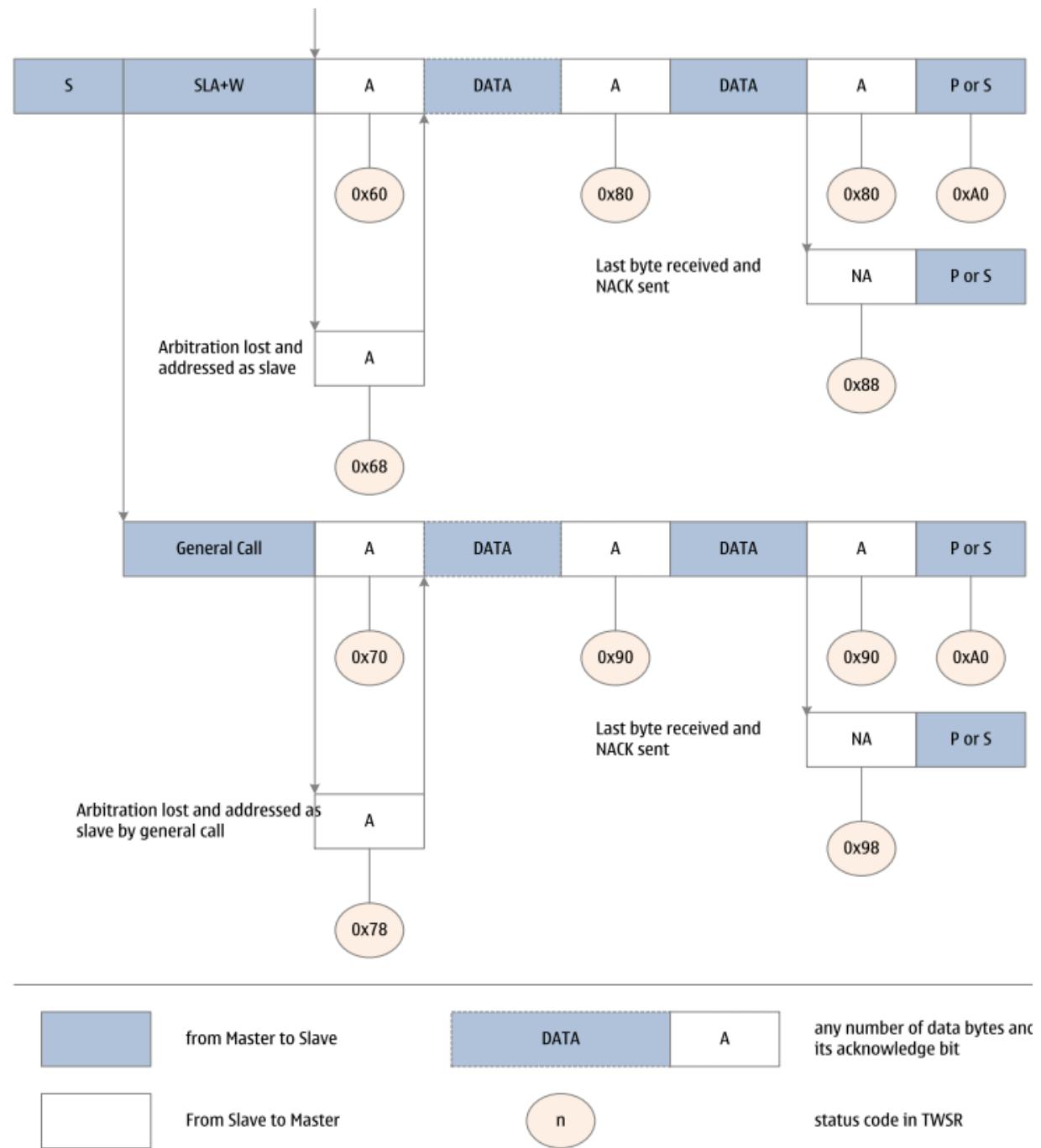


Рис.19.13. Формат режима приема и состояния устройства

#### 19.6.5. Режим ведомой передачи.

В режиме ведомой передачи определенное количество байтов данных может быть отправлено на приемник ведущего. Коды состояния, упомянутые в этом разделе, предполагают, что бит управления предварительным делителем равен «0». Чтобы запустить режим приема ведомого, установите регистры TWAR и TWCR. TWAR необходимо установить следующим образом:

TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
Адрес устройства подчиненного устройства							

Старшие 7 бит TWAR - это ведомый адрес, на который будет реагировать

интерфейс TWI при обращении к узлу. Если установлен LSB, TWI будет отвечать на адрес широковещательного вызова (0x00), в противном случае общий адрес вызова будет проигнорирован.

TWCR необходимо установить следующим образом:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
0	1	0	0	0	1	0	X

TWEN должен быть настроен на включение интерфейса TWI. TWEA должен быть настроен на возврат подтверждения ACK, когда ведущий адресован (адрес ведомого или широковещательный вызов) сам по себе. TWSTA и TWSTO должны быть очищены. После инициализации TWAR и TWCR интерфейс TWI ожидает, пока не будет адресован собственный адрес ведомого устройства (или широковещательный адрес).

Когда бит направления данных, следующий за ведомым адресом, равен «0» (указывает на операцию записи), TWI переходит в режим приема ведомого.

Когда бит направления данных равен «1» (указывающий операцию чтения), TWI переходит в режим подчиненной передачи.

После получения своего собственного подчиненного адреса и флага операции чтения установлен флаг TWINT и обновленный код состояния обновляется в TWSR.

Соответствующий ответ под каждым кодом состояния подробно описан в таблице кодов состояния.

Следует отметить, что режим подчиненной передачи также может быть введен после завершения арбитража TWI в режиме ведущего (см. Код состояния 0xB0).

Если бит TWEA сбрасывается во время передачи, TWI переключится на неадресный подчиненный режим после отправки последнего байта.

После того, как главный приемник выдаст NACK или ACK для передачи последнего байта, код состояния в регистре TWSR будет обновлен до 0xC0 или 0xC8.

Если главный приемник продолжит передачу, ведомый передатчик не будет отвечать, и ведущий получит все данные «1» (т. е. 0xFF).

Когда подчиненный отправляет последний байт данных (TWEA очищается) и ожидает ответа NACK, и хост хочет получить больше данных и отправляет ACK в ответ, TWSR обновляется до 0xC8.

Когда бит TWEA равен «0», TWI не будет отвечать на собственный адрес подчиненного устройства. Тем не менее, TWI все еще слушает шину.

После установки TWEA он может возобновить распознавание адреса и ответить. Другими словами, вы можете использовать TWEA для временной изоляции интерфейса TWI от шины.

В других режимах ожидания, чем в режиме ожидания, можно отключить синхронизацию интерфейса TWI. Если режим приема ведомого включен, интерфейс будет продолжать отвечать на ведомый адрес или широковещательный адрес, используя такты шины. Совпадение адресов пробудит MCU. Во время пробуждения интерфейс TWI будет удерживать SCL до тех пор, пока флаг TWINT не будет очищен.

Когда такты интерфейса TWI вернутся в нормальное состояние, он может получить больше данных.

Код состояния режима ведомой передачи показан в следующей таблице:

*Таблица кодов состояния режима ведомой передачи*

Код состояния	Состояние шины и оборудования	Ответ прикладного программного обеспечения					Следующий шаг в аппаратном обеспечении	
		Чтение / запись TWDR	Работа на TWCR					
			STA	STO	TWINT	TWEA		
0xA8	SLA + R был получен, ACK отправлен	Загрузка данных	X	0	1	0	Пошлют последние данные, ожидаем получить NACK	
		Загрузка данных	X	0	1	1	Посыпает данные, получить ACK	
0xB0	Арбитраж не удался при отправке SLA+R / W; получено SLA+R; отправлено ACK	Загрузка данных	X	0	1	0	Пошлют последние данные, ожидаем получить NACK	
		Загрузка данных	X	0	1	1	Посыпает данные, получить ACK	
0xB8	Данные были отправлены, получено ACK	Загрузка данных	X	0	1	0	Пошлют последние данные, ожидаем получить NACK	
		Загрузка данных	X	0	1	1	Посыпает данные, получить ACK	
0xC0	Данные отправлены, NACK получен	Нет операции	0	0	1	0	Переключится на неактивный ведомый режим, не ответит на подчиненный адрес и трансляцию	
		Нет операции	0	0	1	1	Переключится на неактивный ведомый режим, ответит на подчиненный адрес, TWGCE = 1 ответит на трансляцию	
		Нет операции	1	0	1	0	Перейдет в неадресный ведомый режим, не будет отвечать на ведомый адрес и широковещательную рассылку, отправит START, когда шина свободна	
		Нет операции	1	0	1	1	Переключится на неактивный режим ведомого, ответит на ведомый адрес, TWGCE = 1 ответит на широковещательную передачу, шина отправит START в режиме ожидания	
0xC8	Данные отправлены, ACK получен	Чтение данных	0	0	1	0	Переключится на неактивный ведомый режим, не ответит на подчиненный адрес и трансляцию	
		Чтение данных	0	0	1	1	Переключится на неактивный ведомый режим, ответит на подчиненный адрес, TWGCE = 1 ответит	

							на трансляцию
	Чтение данных	1	0	1	0		Перейдет в неадресный ведомый режим, не будет отвечать на ведомый адрес и широковещательную рассылку, отправит START, когда шина свободна
	Чтение данных	1	0	1	1		Переключится на неактивный режим ведомого, ответит на ведомый адрес, TWGCE = 1 ответит на широковещательную передачу, шина отправит START в режиме ожидания

Формат и состояние режима ведомой передачи следующий:

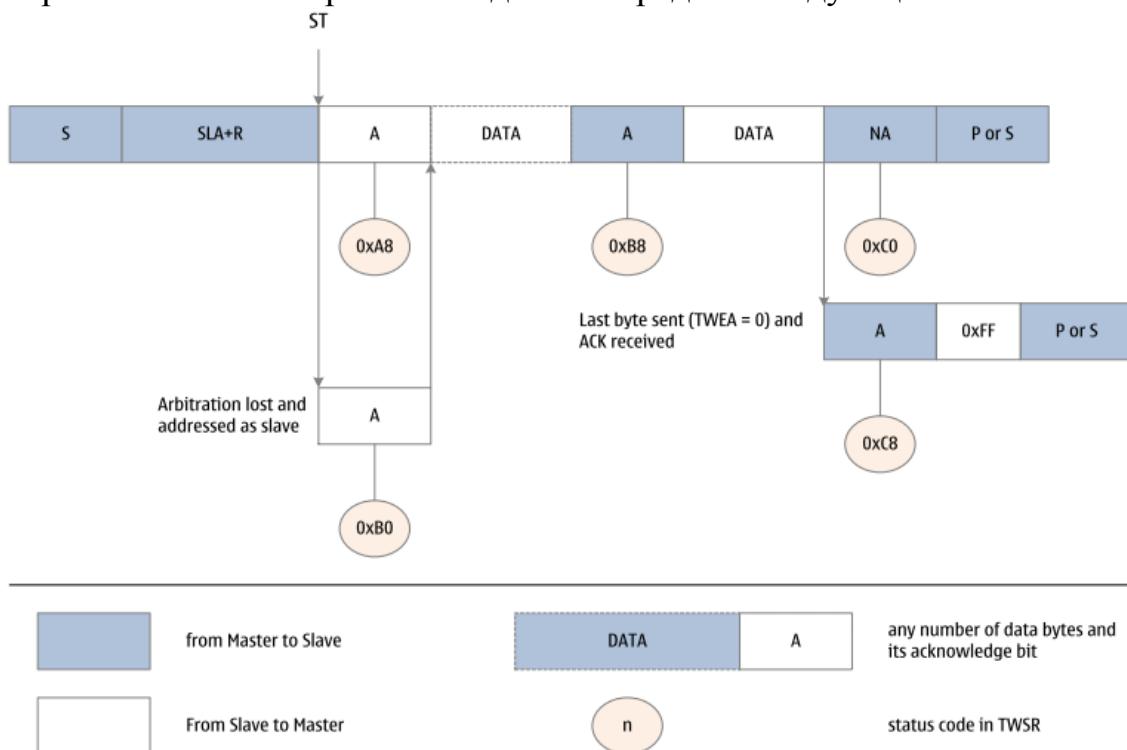


Рис.19.14. Формат режима слайдов и диаграмма состояний

#### 19.6.6. Другие состояния

Существует два кода состояния, которые не имеют соответствующих определений состояния TWI, как показано в следующей таблице:

Таблица Другие коды состояния

Код состояния	Состояние шины и оборудования	Ответ прикладного программного обеспечения				Следующий шаг в аппаратном обеспечении	
		Чтение / запись TWDR		Работа на TWCR			
		STA	STO	TWINT	TWEA		

0xF8	Информация распределенная; TWINT = 0	Нет операции	Не используйте TWCR				Ожидание или выполнение текущей операции
0x00	Ошибка шины, вызванная незаконным START или STOP	Нет операции	0	1	1	X	Только влияет на внутреннее оборудование; STOP не отправляется на шину; шина выпускает и очищает бит TWSTO

Код состояния 0xF8 указывает, что в настоящее время нет соответствующей информации, поскольку флаг TWINT равен «0». Это состояние может возникнуть, когда интерфейс TWI не участвует в последовательной передаче или текущая передача еще не завершена.

Состояние 0x00 указывает на то, что во время последовательной передачи произошла ошибка шины. Ошибка шины возникает, когда происходит незаконный START или STOP. Например, START или STOP появляется между адресом и данными, адресом и ACK. Ошибка шины установит TWINT. Чтобы восстановить ошибку, TWSTO необходимо установить и очистить, написав «1». Это приведет к тому, что интерфейс TWI войдет в незаписанный подчиненный режим без STOP, отпустите SCL и SDA и очистите бит TWSTO.

### *Комбинированный режим*

В некоторых случаях несколько режимов TWI должны быть объединены для выполнения желаемого задания.

Например, чтобы считывать данные из последовательной EEPROM, типичная передача включает в себя следующие шаги:

1. Перенос должен начаться;
2. Необходимо указать место, где EEPROM должен прочитать данные;
3. Чтение должно быть завершено;
4. Передача должна завершиться.

Обратите внимание, что данные могут передаваться от ведущего к ведомому и наоборот. Ведущий сообщает подчиненному устройству, где читать данные, используя режим передачи ведущего. Затем прочтайте данные с ведомого устройства, используя режим приема ведущего.

Направление передачи изменится. Ведущий должен поддерживать управление шиной на всех этапах, и все этапы выполняются. Если в системе с несколькими ведущими, ведущий изменяет местоположение считываемых данных между шагами 2 и 3, этот принцип нарушается, и ведущий будет считывать данные не в том месте.

Изменение направления передачи данных достигается путем отправки REPEATED START между байтом адреса передачи и полученными данными. После отправки REPEATED START у ведущего все еще есть управление шиной. На следующем рисунке показан этот процесс передачи:

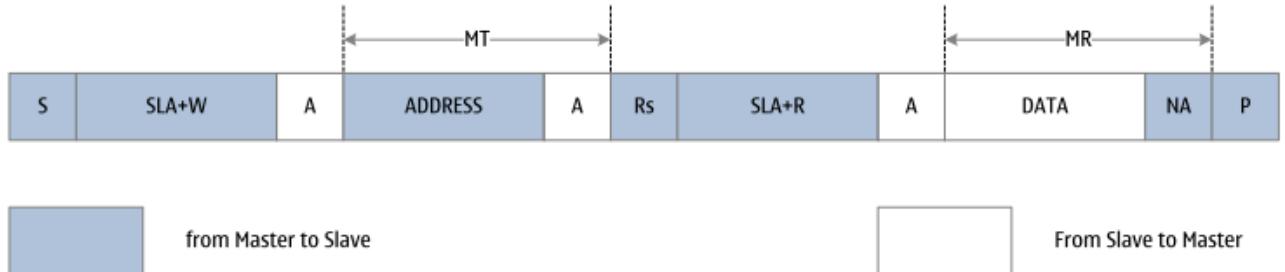


Рис.19.15. Объединяет несколько режимов TWI для доступа к серийным EEPROM-изображениям

## 19.7. Системы с несколькими мастерами и арбитраж

Если несколько **ведущих** подключены к одной и той же шине TWI, один или несколько из них могут одновременно запускать передачу данных. Протокол TWI гарантирует, что в этом случае одному из ведущих разрешается передавать без потери данных через арбитражный процесс.

Ниже описывается процесс арбитража шины с двумя ведущими, которые пытаются отправить данные в ведомый. Существует несколько различных ситуаций, которые могут создать арбитражный процесс шины:

- Два или более хоста одновременно взаимодействуют с одним ведомым. В этом случае как ведущий, так и ведомый не знают, что на шине есть конкуренция;
- Два или более ведущих одновременно выполняют разные данные или операционные обращения к одному и тому же ведомому. Арбитраж происходит в этом случае, в бит или бит READ / WRITE. Когда другие хосты отправляют «0» в строку SDA, хост, который отправляет «1» на линию SDA, не сможет выполнить арбитраж. Неисправный хост переключится на подчиненный режим, который не будет адресован, или дождитесь, пока шина отправит новый сигнал START в режиме ожидания, в зависимости от операции прикладного программного обеспечения.
- Два или более ведущих доступ к другим подчиненным устройствам. В этом случае арбитраж шины возникает во время фазы SLA. Когда другие ведущие отправляют «0» в линию SDA, ведущий, который отправляет «1» на линию SDA, будет арбитраж. Ведущий, который выходит из строя во время арбитража SLA-шины, переключится на подчиненный режим и проверяет, адресован ли он ведущим, который получает контроль над шиной. Если он адресован, он войдет в режим SR или ST, в зависимости от бит READ / WRITE, следующего за SLA. Если он не адресован, он переключится на неадресованный ведомый режим или дождаясь, пока шина отправит новый сигнал START, в зависимости от работы программного обеспечения.

На следующем рисунке описывается процесс арбитража шины:

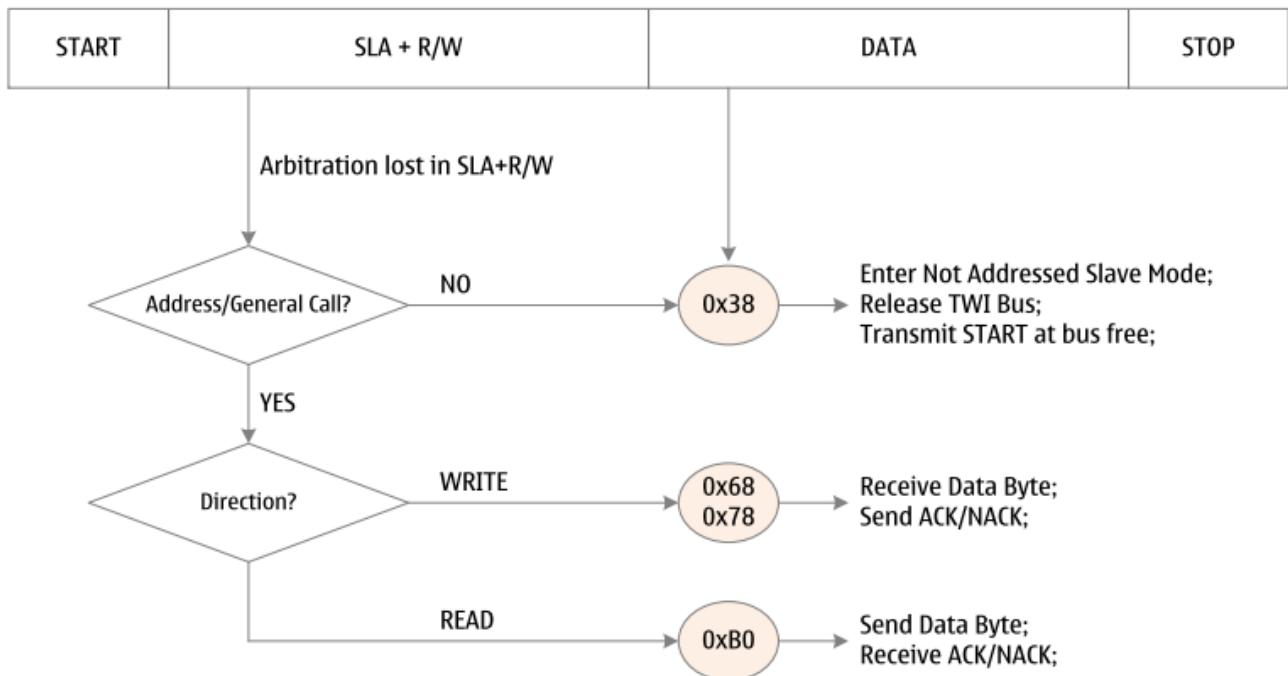


Рис.19.16. Схема арбитражного процесса шины

## 19.8. Регистры Описание

### Список регистров TWI

Регистр	Адресс	Значение по умолчанию	Описание
TWBR	0xB8	0x00	Регистр скорости передачи данных TWI
TWSR	0xB9	0x00	Регистр состояния TWI
TWAR	0xBA	0x00	Регистр адресов TWI
TWDR	0xBB	0xFF	Регистр данных TWI
TWCR	0xBC	0x00	Регистр управления TWI
TWAMR	0xBD	0x00	Регистр маски адреса TWI

### TWBR - Регистр скорости передачи данных TWI

TWBR - регистр скорости передачи данных TWI								
Адрес: 0xB8				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	TWBR7	TWBR6	TWBR5	TWBR4	TWBR3	TWBR2	TWBR1	TWBR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	TWBR	Биты управления битрейтом TWI. TWBR - коэффициент делителя генератора битовых скоростей. Генератор скорости передачи бит - это делитель, который используется для генерации тактов SCL в режиме ведущий. Битовая скорость вычисляется следующим образом: $f_{scl} = f_{sys} / (16 + 2 * TWBR * 4^{TWPS})$						

### TWSR - Регистр состояния TWI

### TWSR - регистр состояния TWI

Адрес: 0xB9			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	TWS7	TWS6	TWS5	TWS4	TWS3	-	TWPS1	TWPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:3	TWS[7:3]	Флаг состояния TWI. 5-битный TWS реагирует на логику TWI и состояние шины. Различные значения статуса имеют разные значения. <a href="#">Подробнее см.</a> <b>Описание режима работы TWI.</b> Значение, считываемое из TWSR, включает в себя 5-битное значение состояния и бит бит предварительного делителя 2 бит, а бит предварительного каскада должен быть замаскирован в «0» при обнаружении состояния. Это состояние, независимое от настроек предварительного делителя.						
2	-	Зарезервировано						
1	TWPS1	Регулятор предварительного делителя TWI высокий. TWPS1 и TWPS0 вместе образуют TWPS [1: 0] для управления предварительным делителем скорости передачи битов и для управления скоростью передачи битов вместе с TWBR.						
0	TWPS0	TWPS[1:0]		Коэффициент предделения				
		0		1				
		1		4				
		2		16				
		3		64				

## *TWAR - Регистр адресов TWI*

## *TWDR - Регистр данных TWI*

7:0	TWDR	Регистр данных TWI. TWD является следующим байтом на передаваемой шине или последним байтом, который был только что получен на шине.
-----	------	--------------------------------------------------------------------------------------------------------------------------------------

## *TWCR - Регистр управления TWI*

		<p>Бит TWEN обеспечивает работу TWI и активирует интерфейс TWI. Когда бит TWEN в «1», вывод IO управления TWI подключается к контактам SCL и SDA.</p> <p>Когда бит TWEN в «0», интерфейсный модуль TWI отключается и все передачи прекращаются, включая текущие операции.</p>
1	-	Зарезервировано
0	TWIE	<p>Бит управления прерыванием TWI.</p> <p>Когда бит TWIE установлен в «1», и глобальное прерывание установлено, запрос прерывания TWI активируется всякий раз, когда флаг TWINT в «1».</p>

## *TWAMR - Регистр масок адресов TWI*

## 20. Аналоговый компаратор 0 (AC0)

- Точность сравнения 10 мВ
  - Заводская калибровка смещения
  - Поддержка 3 аналоговых входов с защелкой
  - Поддержка мультиплексированного входа ADC (ADMUX)
  - Поддержка внутреннего дифференциального входа усилителя (DFFO)
  - Поддержка внутренних 8 бит Вход ЦАП (DAO)
  - Программируемый выход Цифровая фильтрация

## 20.1. Обзор

Аналоговый компаратор сравнивает уровень неинверсный и инверсный входных компараторов.

Когда напряжение неинверсного входа выше напряжения инверсного входа, устанавливается аналоговый выходной сигнал компаратора АС0.

Когда уровень АС0 изменяется, фронт сигнала может использоваться для запуска прерывания. Выходной сигнал АС0 также может использоваться для запуска ввода входного сигнала счетчика 1 и управления выходным сигналом ШИМ с таймером.

LGT8FX8P AC0 встроен аналоговый компаратор, аналоговый мультиплексор содержит входной селектор, компаратор неинверсный и инверсный вывод источника входного сигнала может быть выбран источник опорного сигнала генерируется из различных внутренних или внешних портов.

Аналоговый компаратор сам по себе поддерживает калибровку смещения, что обеспечивает согласованность работы компаратора.

Компаратор поддерживает дополнительную функцию аппаратного гистерезиса для улучшения стабильности выхода компаратора. В то же время на выходе компаратора может быть запрограммирован цифровой аппаратный фильтр.

В соответствии с требованиями приложения для выбора более стабильного результата сравнения можно выбрать подходящую настройку фильтра.

Состояние выхода компаратора можно считывать непосредственно через регистры, а запросы прерывания также могут быть сгенерированы для обеспечения более эффективного захвата событий в реальном времени.

Выход компаратора также может выводиться непосредственно на внешний порт ввода-вывода. Ниже показана структурная схема операционного усилителя / аналогового компаратора 0.

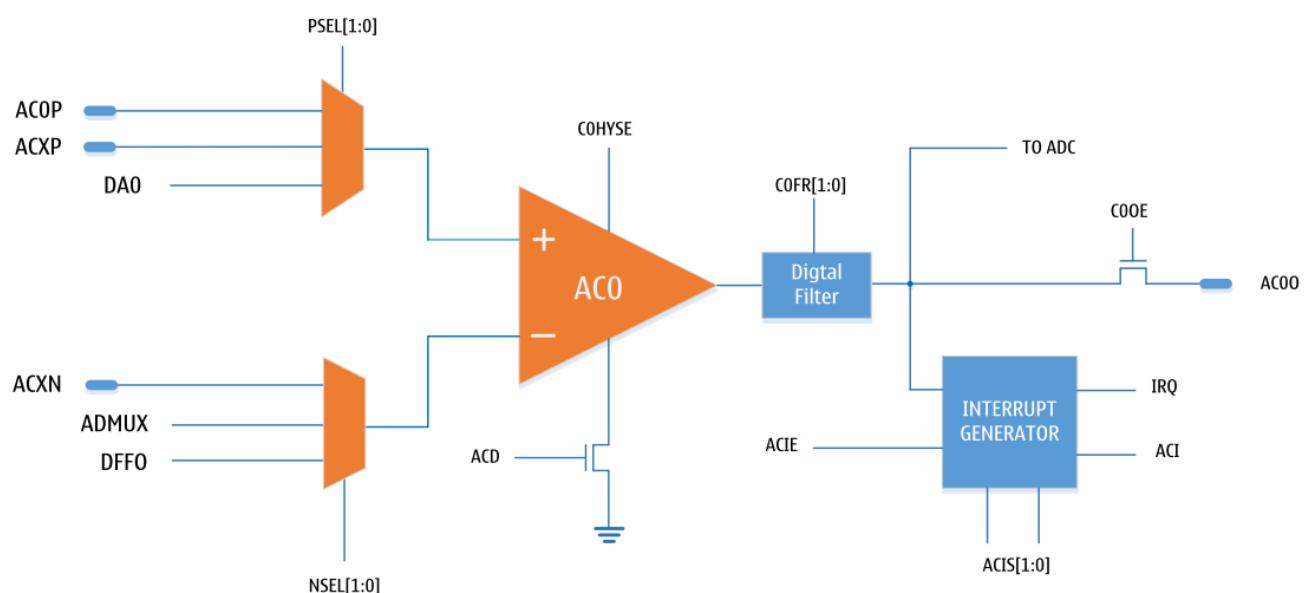


Рис.20.1 Аналоговый компаратор 0 Функциональная Схема

## 20.2. Входы аналогового компаратора

Оба входа аналогового компаратора поддерживают несколько выбираемых

источников входного сигнала. Неинверсный вход выбирается тремя способами:

1. Внешний независимый аналоговый вход AC0P
2. Аналоговый компаратор 0/1 Общий аналоговый вход ACXP
3. Выход внутреннего 8-разрядного ЦАП DAO

Выбор входного источника управляется бит C0BG в регистре состояния управления C0SR и бит C0PS0 в регистре C0XR. Подробнее см. В разделе описания регистра в этом разделе.

AC0P - входной канал положительного режима AC0. **Обратите внимание**, что существует небольшая разница в положении штыря AC0P в разных упаковках. AC0P в пакете QFP48 является независимым портом. QFP32 инкапсулирует этот порт AC0P параллельно с PD6 в порт.

ACXP является общим неинверсным входом компаратора 0/1.

В LGT8FX8P есть два аналоговых компаратора. ACXP одновременно подключается к неинверсным мультиплексорам обоих компараторов, что облегчает работу двух компараторов.

DAO - это выход из внутреннего 8-разрядного ЦАП. Исходный источник ЦАП может быть выбран из мощности системы, внутреннего задания или ввода из внешней ссылки. Для конфигурации ЦАП см. Разделы, относящиеся к КСР.

C0BG	C0PS0	Источник входного сигнала AC0
0	0	AC0P
0	1	ACXP
1	0	DAO
1	1	Отключить неинверсный входной канал компаратора

Инверсный вход также может выбирать три разных аналоговых входа:

1. Компаратор 0/1 Общий аналоговый вход ACXN
2. Выход мультиплексора ADC ADMUX
3. Выход внутреннего дифференциального усилителя DFFO-компаратор

Инверсный выбор входного канала от АЦП Управление бит СМЕ00 / 01 в регистре ADCSRB модуля. Когда инверсный вход компаратора выбирается как ADMUX, аналоговый входной канал должен выбираться через бит CHMUX регистра ADMUX модуля ADC. В этом режиме вход компаратора может расширяться более гибко.

ACXN является общим инверсным входом компаратора 0/1, что облегчает совместное использование компаратора 0/1; DFFO получается из выхода внутреннего дифференциального усилителя.

Дифференциальный усилитель доступен с контролем усиления x1 / x8 / x16 / x32 для обнаружения и измерения малого сигнала.

СМЕ01	СМЕ00	Источник входного сигнала AC0
0	0	ACXN
0	1	ADMUX
1	0	DFFO
1	1	Отключите инверсный входной канал компаратора

## 20.3. Выход компаратора и управление PWM

LGT8FX8P поддерживает многоканальные ШИМ-выходы, которые могут использоваться с модулем компаратора. Выход компаратора может использоваться для прямого отключения сигнала ШИМ для реализации более гибкой схемы защиты ШИМ. Для управления, связанного с выходом ШИМ, обратитесь к соответствующему разделу секции таймера.

Фильтрация выходного сигнала компаратора

### 20.3.1. Выходной фильтр компаратора

Выход компаратора поддерживает встроенный гистерезис. Пользователь может включить схему гистерезиса через бит C0HYSE регистра C0XR. Контур гистерезиса может устранить неустойчивое состояние процесса изменения состояния компаратора и получить функцию выходного фильтра.

**Рекомендуется**, чтобы пользователь включил схему гистерезиса при использовании компаратора для получения стабильного выхода компаратора. Как показано на рисунке ниже, схема гистерезиса расположена между аналоговым выходом компаратора и цифровым выходом.

Когда положительное входное напряжение V IN + компаратора больше (VIN+ - V H +), выходной сигнал компаратора COUT высок, когда напряжение V IN + меньше (V IN - - V H -), выходной сигнал компаратора низкий. Схема гистерезиса позволяет избежать дрожания, вызванного самой схемой, когда положительное напряжение компаратора приближается к отрицательному напряжению.

Напряжение гистерезиса компаратора и диаграмма выходного сигнала компаратора:

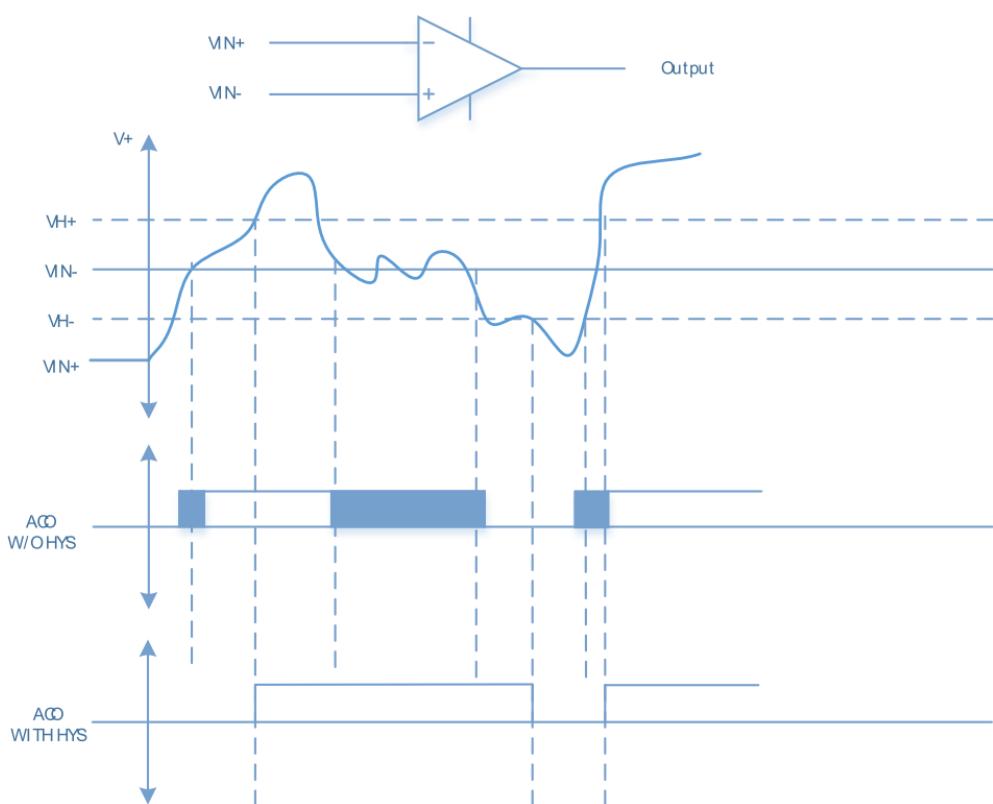


Рис.20.2

Хотя схема гистерезиса очень эффективна для подавления пульсации напряжения вблизи порога компаратора, входной сигнал может быть нарушен различными сильными сторонами в реальной среде приложения. Более сильные помехи могут привести к тому, что входной уровень мгновенно повысится за пределы порогового диапазона схемы гистерезиса и не может быть эффективно подавлен.

LGT8FX8P объединяет программируемый цифровой фильтр на выходе компаратора для фильтрации эффектов переходных помех на выходе компаратора. Цифровой фильтр может выбирать соответствующую ширину фильтра времени в соответствии с требованиями приложения. Схема фильтра обновляет выход компаратора только тогда, когда выход компаратора продолжает соответствовать временному пределу фильтра. Для достижения более стабильной производительности.

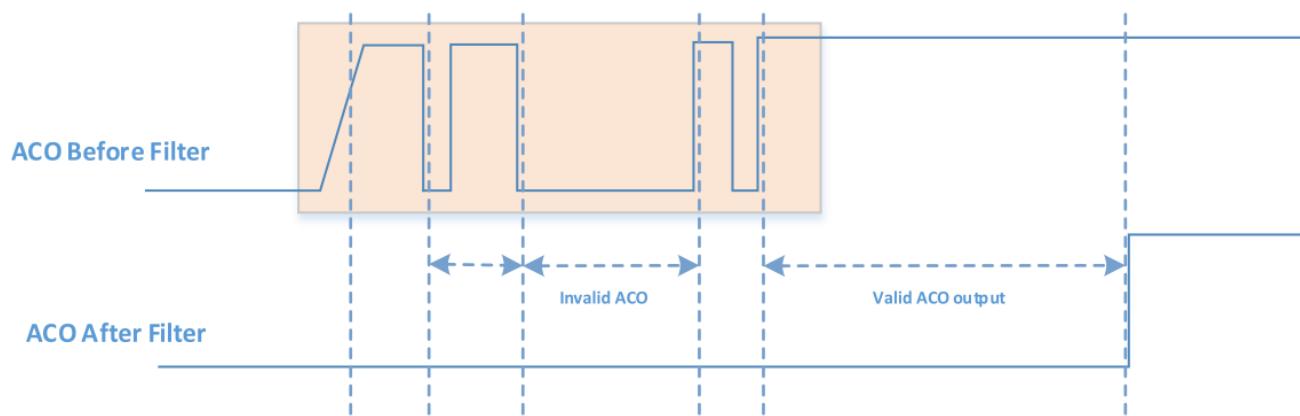


Рис.20.3. Синхронизация выходного фильтра компаратора

Цифровая фильтрация AC0 управляется битами C0FEN и C0FS регистра C0XR. Для конкретного метода настройки см. Раздел определения регистра этой главы.

## 20.4. Регистры Описание

### *COSR - Регистр управления и состояния AC0*

		C0BG	C0PS0	Неинвертируемого входа															
		0	0	AC0P в качестве неинвертируемого входа															
		0	1	ACXP в качестве неинвертируемого входа															
		1	0	Выход внутреннего ЦАП в качестве неинвертируемого входа															
		1	1	Отключите источник неинвертируемого входного сигнала AC0															
5	C0O	Бит состояния выхода аналогового компаратора. Выход аналогового компаратора напрямую связан с битом C0O после синхронизации. Программное обеспечение может считывать значение бит C0O для получения выходного значения аналогового компаратора.																	
4	C0I	Флаг прерывания Аналогового компаратора. Когда выходное событие аналогового компаратора запускает режим прерывания, определенный бит C0IS, устанавливается бит C0I. Прерывание генерируется, когда бит разрешения прерывания C0IE в «1», и установлено глобальное прерывание. Когда выполняется процедура обслуживания прерывания аналогового компаратора, C0I автоматически очищается, или может быть записано в бит C0I «1», чтобы очистить бит.																	
3	C0IE	Бит разрешения прерывания от аналогового компаратора. Когда бит C0IE установлен в 1 и глобальное прерывание включено, прерывания AC0 включены. Когда бит C0IE установлен в 0, прерывания AC0 отключены.																	
2	C0IC	Аналоговый бит ввода-вывода аналогового компаратора C0IC = 1, источник входного захвата счетчика/таймера 1 поступает от выхода аналогового компаратора. C0IC = 0, источник входного сигнала счетчика/таймера 1 поступает от внешнего контакта ICP1.																	
1	C0IS1	Биты режима запуска прерывания от аналогового компаратора. C0IS0 и C0IS1 вместе образуют C0IS [1: 0], который используется для управления режимом запуска прерывания аналогового компаратора.																	
0	C0IS0	<table border="1"> <thead> <tr> <th>C0IS1</th> <th>C0IS0</th> <th>Режим прерывания</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>На каждый фронт</td> </tr> <tr> <td>0</td> <td>1</td> <td>Зарезервировано</td> </tr> <tr> <td>1</td> <td>0</td> <td>Падающий фронт</td> </tr> <tr> <td>1</td> <td>1</td> <td>Настающий фронт</td> </tr> </tbody> </table>			C0IS1	C0IS0	Режим прерывания	0	0	На каждый фронт	0	1	Зарезервировано	1	0	Падающий фронт	1	1	Настающий фронт
C0IS1	C0IS0	Режим прерывания																	
0	0	На каждый фронт																	
0	1	Зарезервировано																	
1	0	Падающий фронт																	
1	1	Настающий фронт																	

### *ADCSRБ - АЦП и регистр состояния B*

		10: Внутренний делитель напряжения АЦП, - инверсный вход AC1 11: Выход дифференциального усилителя - инверсный вход AC1
3	ACTS	Выбор канала источника переменного тока 0 - Выход AC0 как источник запуска автоматического преобразования АЦП 1 - Выход AC1 как источник запуска автоматического преобразования АЦП
2	ADTS2	
1	ADTS1	
0	ADTS0	См. Описание регистра ADC.

### C0XR - Регистр вспомогательного контроля AC0

C0XR - регистр вспомогательного контроля AC0													
Адрес: 0x51				Значение по умолчанию: 0x00									
Бит	Имя	Определение бит											
7	-	Зарезервировано											
6	C0OE	Выход компаратора AC0 для управления выходом через внешний порт C0OE = 1, выход компаратора AC0 для внешнего порта PD2 C0OE = 0, отключить вывод компаратора на внешний порт											
5	C0HYSE	Функция гистерезиса выходного сигнала AC0. C0HYSE = 1 - включить гистерезис выхода C0HYSE = 0 - отключить гистерезис выхода											
4	C0PS0	Положительный источник входного сигнала AC0 выбирает низкий бит. C0PS0 и C0BG совместно управляют положительным источником входного сигнала AC0. См. Определение регистра C0SR.											
3	C0WKE	Используется для управления пробуждением сна AC0. C0WKE = 1 - активировать функцию пробуждения выхода компаратора C0WKE = 0 - отключить функцию пробуждения выхода компаратора											
2	C0FEN	Разрешить управление цифровым фильтром компаратора. C0FEN = 1 - цифровой фильтр включен C0FEN = 0 - цифровой фильтр отключен											
1:0	C0FS[1:0]	Установка ширины цифрового фильтра компаратора 00 = выкл. 01 = 32us 10 = 64us 11 = 96us											

## 21. Аналоговый компаратор 1 (AC1)

- Точность сравнения 10 мВ
- Заводская калибровка смещения
- Поддержка 4-стороннего аналогового входа вне чипа
- Поддерживает внутренний вход 1/5 делителя (VDO)
- Поддерживает вход внутреннего дифференциального усилителя (DFFO)
- Поддерживает внутренний 8-разрядный вход ЦАП (DAO)
- Программируемое управление выходным фильтром

## 21.1. Обзор

Аналоговый компаратор сравнивает уровень неинвертируемого и инвертируемого входов компаратора.

Когда напряжение инвертируемого входа выше напряжения неинвертируемого входа, устанавливается аналоговый выходной сигнал компаратора АСО.

Когда уровень АСО изменяется, фронт сигнала может использоваться для запуска прерывания.

Выходной сигнал АСО также может использоваться для запуска захвата входного сигнала счетчика 1 и для управления выходным сигналом PWM с таймером.

Интегрированный аналоговый компаратор LGT8FX8P - AC1, включает в себя мультиплексированный селектор аналогового входа. Входы неинвертируемый и инвертируемый компаратора могут быть выбраны из внешних портов или из нескольких встроенных исходных источников.

Аналоговый компаратор сам по себе поддерживает калибровку смещения, что обеспечивает согласованность работы компаратора.

Компаратор поддерживает дополнительную функцию аппаратного гистерезиса для улучшения стабильности выхода компаратора. В то же время на выходе компаратора может быть запрограммирован цифровой аппаратный фильтр.

В соответствии с требованиями приложения для выбора более стабильного результата сравнения можно выбрать подходящую настройку фильтра.

Состояние выхода компаратора можно считывать непосредственно через регистр, а запросы прерывания также могут быть сгенерированы для обеспечения более эффективного захвата событий в реальном времени.

Выход компаратора также может выводиться непосредственно на внешний порт ввода-вывода.

Структура аналогового компаратора 1 показана на рисунке ниже.

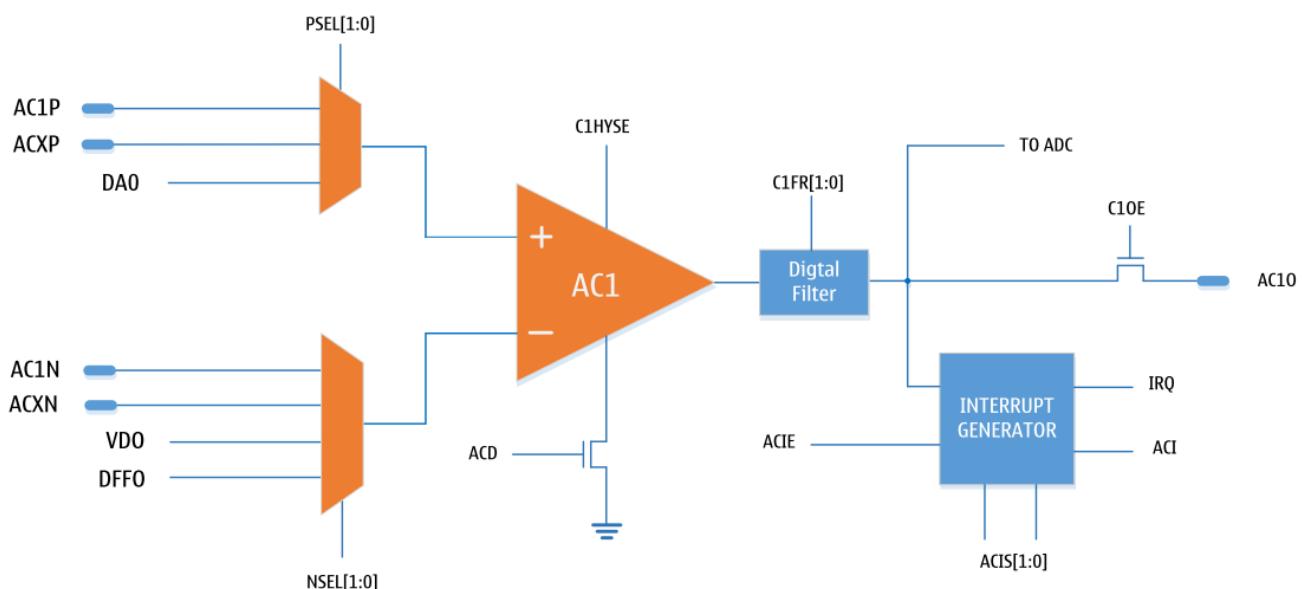


Рис. 21.1. Аналоговый компаратор 1 Функциональная Схема

## 21.2. Входы аналогового компаратора

Оба аналоговых входа имеют несколько источников входного сигнала.  
Неинвертируемый вход трехпозиционный:

1. Внешний независимый аналоговый вход AC1P
2. Аналоговый компаратор 0/1 Общий аналоговый вход ACXP
3. Выход внутреннего 8-битного ЦАП DAO

Выбор источника входного сигнала управляется битом C1BG в регистре состояния управления C1SR и битом C1PS0 в регистре C1XR. Подробнее см. В разделе описания регистра в этом разделе.

AC1P - это входной канал неинвертируемый режима AC1.

ACXP является общим неинвертируемый входом компаратора 0/1.

В LGT8FX8P есть два аналоговых компаратора. ACXP одновременно подключается к неинвертируемый мультиплексору двух компараторов, что облегчает сотрудничество двух компараторов.

DAO - это выход из внутреннего 8-разрядного ЦАП.

Исходный источник ЦАП может быть выбран из мощности системы, внутреннего задания или ввода из внешней ссылки. Для конфигурации ЦАП см. Разделы, относящиеся к КСР.

C1BG	C1PS0	Источник входного сигнала AC1
0	0	AC1P
0	1	ACXP
1	0	DAO
1	1	Отключить неинверсный входной канал компаратора

Инверсный вход может также выбирать 4 различных аналоговых входа:

1. Внешний аналоговый вход AC1N - инверсный вход AC1
2. Компаратор 0/1 - общий инверсный вход ACXN
3. Внутренний выход 1/5 делителя АЦП - инверсный вход AC1
4. Выход внутреннего дифференциального усилителя DFFO - отрицательный вход AC1

Выбор инверсного входного канала компаратора управляется битами СМЕ11/10 в регистре ADCSRB от модуля ADC.

Когда инверсный вход компаратора выбирается как выход внутреннего делителя АЦП, входной сигнал многоканального делителя напряжения должен выбираться через **бит VDS регистра ADCRSC модуля ADC**.

ACXN является общим инверсным входом компаратора 0/1, что облегчает совместное использование компаратора 0/1.

DFFO подается из выхода внутреннего дифференциального усилителя. Дифференциальный усилитель выбирает регуляторы усиления x1 / x8 / x16 / x32 для обнаружения и измерения малого сигнала.

СМЕ11	СМЕ10	Источник входного сигнала AC1
0	0	ACXN
0	1	AC1N
1	0	VDO

## 21.3. Выход компаратора и управление PWM

LGT8FX8P поддерживает многоканальные ШИМ-выходы, которые могут использоваться с модулем компаратора. Выход компаратора может использоваться для прямого отключения сигнала ШИМ для реализации более гибкой схемы защиты ШИМ. Для управления, связанного с выходом ШИМ, обратитесь к соответствующему разделу секции таймера.

### 21.3.1. Выходной фильтр компаратора

Выход компаратора поддерживает встроенный гистерезис. Пользователь может включить схему гистерезиса через бит C1HYSE регистра C1XR. Контур гистерезиса может устранить неустойчивое состояние процесса изменения состояния компаратора и получить функцию выходного фильтра.

**Рекомендуется**, чтобы пользователь включил схему гистерезиса при использовании компаратора для получения стабильного выхода компаратора. Как показано на рисунке ниже, схема гистерезиса расположена между аналоговым выходом компаратора и цифровым выходом.

Когда положительное входное напряжение V IN + компаратора больше (VIN+ + VH+), выходной сигнал компаратора C1UT высок, когда напряжение V IN + меньше (VIN+ - VH-), выходной сигнал компаратора низкий. Схема гистерезиса позволяет избежать дрожания, вызванного самой схемой, когда положительное напряжение компаратора приближается к отрицательному напряжению.

Напряжение гистерезиса компаратора и диаграмма выходного сигнала компаратора:

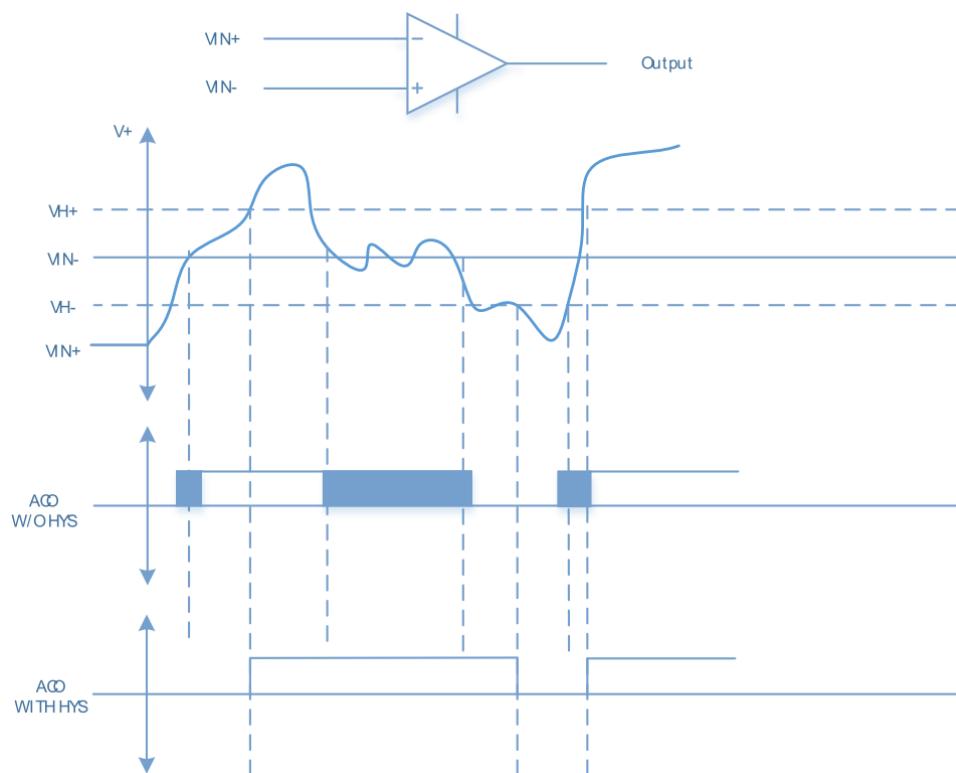


Рис.21.1

Хотя схема гистерезиса очень эффективна для подавления пульсации напряжения вблизи порога компаратора, входной сигнал может быть нарушен различными сильными сторонами в реальной среде приложения. Более сильные помехи могут привести к тому, что входной уровень мгновенно повысится за пределы порогового диапазона схемы гистерезиса и не может быть эффективно подавлен.

LGT8FX8P объединяет программируемый цифровой фильтр на выходе компаратора для фильтрации эффектов переходных помех на выходе компаратора. Цифровой фильтр может выбирать соответствующую ширину фильтра времени в соответствии с требованиями приложения. Схема фильтра обновляет выход компаратора только тогда, когда выход компаратора продолжает соответствовать временному пределу фильтра. Для достижения более стабильной производительности.

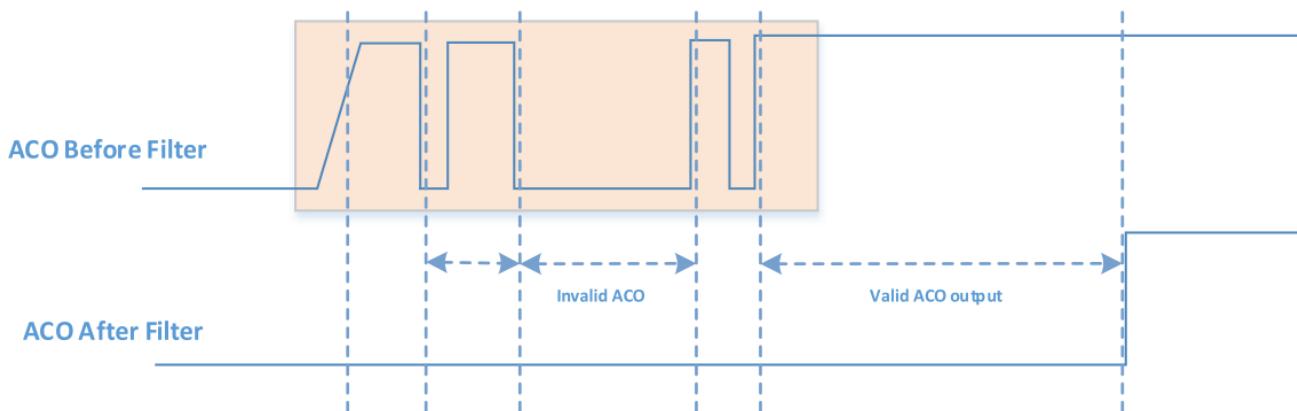


Рис.21.2 Синхронизация выходного фильтра компаратора

Цифровая фильтрация AC0 управляется битами C0FEN и C0FS регистра C0XR. Для конкретного метода настройки см. Раздел определения регистра этой главы.

## 21.4. Регистры Описание

*C1SR - Регистр управления и состояния АС1*

		C1BG	C1PS0	Неинвертируемого входа
		0	0	AC0P в качестве неинвертируемого входа
		0	1	ACXP в качестве неинвертируемого входа
		1	0	Выход внутреннего ЦАП в качестве неинвертируемого входа
		1	1	Отключите источник неинвертируемого входного сигнала AC0
5	C1O	Бит состояния выхода аналогового компаратора. Выход аналогового компаратора напрямую связан с битом C1O после синхронизации. Программное обеспечение может считывать значение бит C1O для получения выходного значения аналогового компаратора.		
4	C1I	Флаг прерывания Аналогового компаратора. Когда выходное событие аналогового компаратора запускает режим прерывания, определенный бит C1IS, устанавливается бит C1I. Прерывание генерируется, когда бит разрешения прерывания C1IE в «1», и установлено глобальное прерывание. Когда выполняется процедура обслуживания прерывания аналогового компаратора, C1I автоматически очищается, или может быть записано в бит C1I «1», чтобы очистить бит.		
3	C1IE	Бит разрешения прерывания от аналогового компаратора. Когда бит C1IE установлен в 1 и глобальное прерывание включено, прерывания AC1 включены. Когда бит C1IE установлен в 0, прерывания AC1 отключены.		
2	C1IC	Аналоговый бит ввода-вывода аналогового компаратора C1IC = 1 - источник входного захвата счетчика/таймера 1 поступает от выхода аналогового компаратора. C1IC = 0 - источник входного сигнала счетчика/таймера 1 поступает от внешнего контакта ICP1.		
1	C1IS1	Биты режима запуска прерывания от аналогового компаратора. C1IS0 и C1IS1 вместе образуют C1IS [1: 0], который используется для управления режимом запуска прерывания аналогового компаратора.		
0	C1IS0	C1IS1	C1IS0	Режим прерывания
		0	0	На каждый фронт
		0	1	Зарезервировано
		1	0	Падающий фронт
		1	1	Настающий фронт

### *C1XR - Регистр вспомогательного контроля AC1*

		сигнала AC1. См. Определение регистра C1SR.
3	C1WKE	Используется для управления пробуждением сна AC1. C1WKE = 1 - активировать функцию пробуждения выхода компаратора C1WKE = 0 - отключить функцию пробуждения выхода компаратора
2	C1FEN	Разрешить управление цифровым фильтром компаратора. C1FEN = 1 - цифровой фильтр включен C1FEN = 0 - цифровой фильтр отключен
1:0	C1FS[1:0]	Установка ширины цифрового фильтра компаратора 00 = выкл. 01 = 32us 10 = 64us 11 = 96us

## 22. Цифро-аналоговый преобразователь (ЦАП).

- 8-разрядный цифровой аналоговый преобразователь.
- Выход ЦАП может использоваться в качестве аналогового входного сигнала сравнения.
- Поддерживаемый выход ЦАП на внешний порт (DAO).
- Дополнительный источник напряжения VCC / AVREF / IVREF

### 22.1. Обзор

LGT8FX8P имеет 8-битный программируемый цифро-аналоговый преобразователь (ЦАП).

Эталонная входное питание ЦАП может быть выбрана из источника питания системы, внутреннего задания или входа AVREF от внешнего порта чипа.

Выход ЦАП может быть выбран в качестве источника входного сигнала для внутреннего компаратора AC0 / 1 или может быть непосредственно выведен на внешний вывод микросхемы для использования в качестве внешней ссылки.

Когда ЦАП выводится на внешний контакт, он не может быть напрямую использован для управления нагрузкой. Ему необходимо передать последовательное напряжение или другую аналогичную схему возбуждения.

Внутренняя структура ЦАП показана на рисунке ниже:

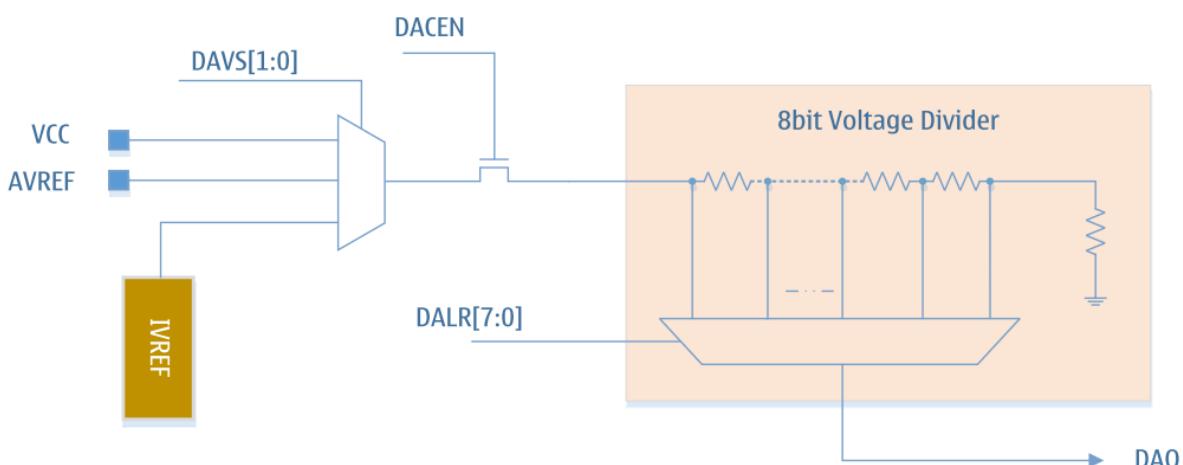


Рис. 22.1. Структурная схема ЦАП.

## 20.2. Регистры Описание

### DACON - регистр управления ЦАП

DACON - регистр управления ЦАП								
Адрес: 0xA0				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	-	-	-	-	DACEN	DAOE	DAVS1	DAVS0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:4	-	Зарезервировано						
3	DACEN	Бит управления разрешения ЦАП DACEN = 1 - Включить модуль ЦАП DACEN = 0 - Отключить модуль ЦАП						
2	DAOE	Выход ЦАП для управления доступом к внешнему порту DAOE = 1 - Включить вывод ЦАП для внешнего PD4 DAOE = 0 - Отключить вывод ЦАП на внешний порт						
1	DAVS1	Источник напряжения ЦАП опорного 0. [DVS1, DVS0] = 00: - Рабочее напряжение VCC 01: - Внешний вход AVREF 10: - Внутреннее опорное напряжение 11: - Отключить опорный источник ЦАП, а также выключить модуль ЦАП						
0	DAVS0							

### DALR - регистр данных DAC

DALR - регистр данных DAC								
Адрес: 0xA1				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	DALR7	DALR6	DALR5	DALR4	DALR3	DALR2	DALR1	DALR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	DALR	Регистр данных ЦАП. Настройка ЦАП выходной режим соотношения между величиной напряжения ЦАП выходным напряжением DALR из: $V_{DAO} = V_{REF} * (DALR + 1) / 256$ где: $V_{DAO}$ - ЦАП выходные аналоговое напряжение $V_{REF}$ - источник опорного напряжения ЦАП, в регистре DACON выбор бит DAVS						

## 23. 12-разрядный аналого-цифровой преобразователь (АЦП)

- 12-разрядное разрешение с  $\pm 1\text{LSB}$  DNL и  $\pm 1.5\text{LSB}$  INL
- Частота дискретизации до 500KSPS с наивысшим разрешением
- 12 мультиплексированных односторонних входных каналов
- Многоканальные каналы с дифференциальным усилителем с программируемым усилением
- Диапазон входного напряжения равен 0 - VCC
- Внутреннее опорное напряжение 1.024V / 2.048V / 4.096V
- Поддержка AVCC и внешний вход опорного напряжения

- Внутренняя многопозиционная схема делителя напряжения 1/5, 4/5
- Поддержка положительной и отрицательной калибровки смещения
- Режим запуска автоматического запуска, основанный на источнике прерывания
- Поддержка автоматического контроля канала переполнения / уменьшения
- Результат преобразования поддерживает дополнительный режим выравнивания
- Запрос на прерывание преобразования

### 21.3. Обзор

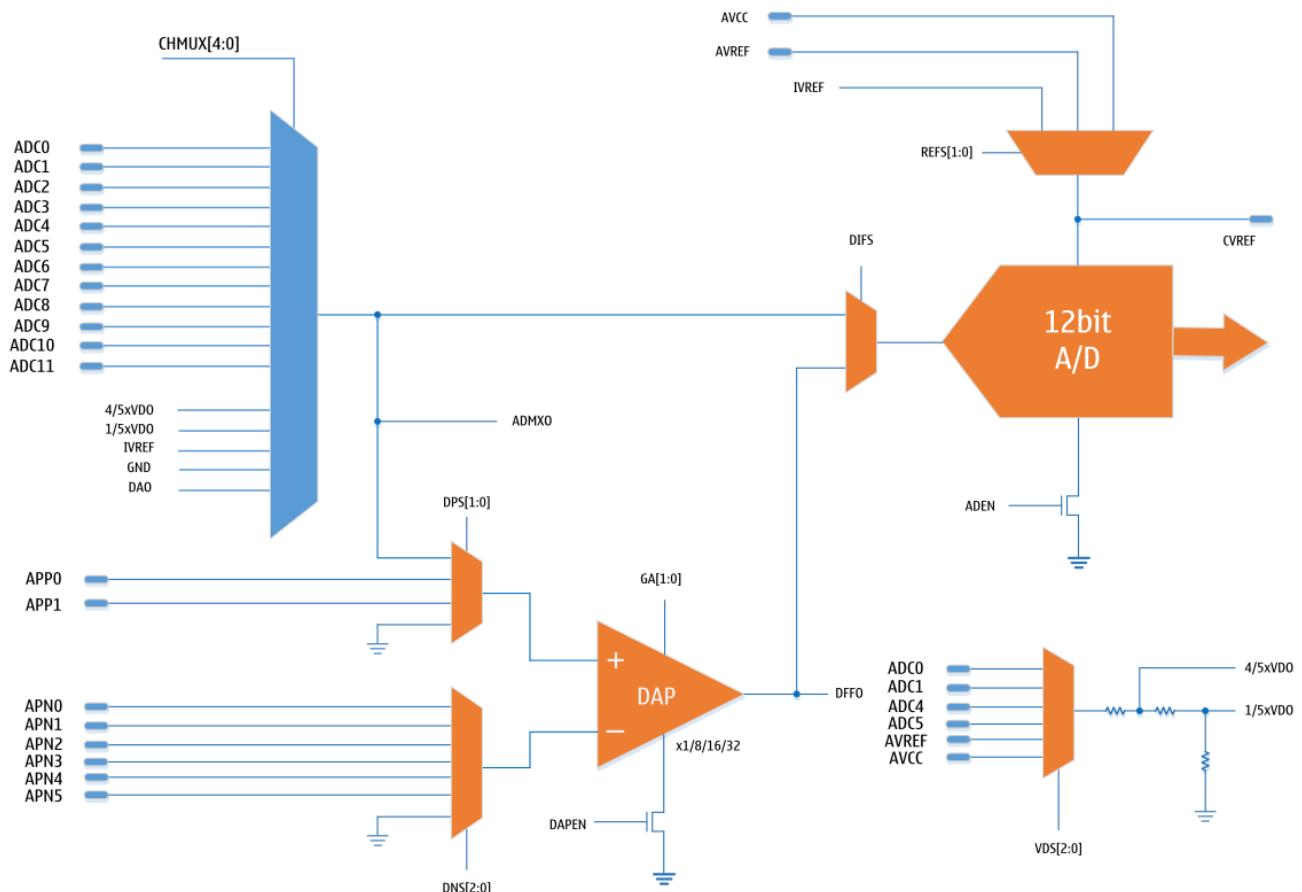


Рис. 23.1. Блок-схема АЦП

Аналогово-цифровой преобразователь представляет собой 12-разрядный АЦП с последовательным приближением. АЦП подключен к 17-канальному аналоговому мультиплексору, который может подключать 12 аналоговых входов и 5 внутренних источников напряжения из внешних портов чипа.

В АЦП интегрирован внутренний дифференциальный операционный усилитель с программируемым усилением  $x1 / x8 / x16 / x32$ . Вход усилителя может поступать от внешнего порта или выхода мультиплексора ADC. Выход дифференциального усилителя можно использовать подключить к аналоговому входу АЦП.

Аналоговый источник входного сигнала АЦПА содержит внутренний мультиплексированный вход от внутреннего АЦПА делителя напряжения, источника опорного напряжения внутреннего, внутреннего аналогового заземления и аналогового **выходного сигнала от модуля ключа касания**.

Внутренний множественный делитель входного напряжения одновременно выдает два 4/5, 1/5 напряжения, а вход делителя напряжения может выбирать уровень от внешнего порта или от источника питания системы.

АЦП поддерживает калибровку смещения. Процесс калибровки смещения контролируется программным обеспечением. Калибровка смещения включает как положительную, так и отрицательную калибровку. После того, как калибровка смещения активирована, контроллер ADC автоматически откалибрует результаты выборки АЦП с использованием как положительных, так и отрицательных значений калибровки. Для метода калибровки смещения см. Соответствующую часть этой главы.

## 23.2. Работа АЦП

АЦП преобразует входное аналоговое напряжение в 12-разрядное цифровое значение путем последовательного приближения. Минимальное значение представляет GND, максимальное значение представляет собой опорное напряжение минус 1 LSB.

Источник опорного напряжения может быть выбран в качестве напряжения питания AVCC АЦП, опорное напряжение AVREF внешнее или внутреннее опорное напряжение **1.024V / 2.048V** путем записи битов в **ADMUX** работах.

Аналоговый входной канал можно выбрать, записав бит CHMUX в регистре **ADMUX**.

Любой входной разъем ADC, внешний опорный вывод напряжения и внутренний опорный сигнал могут использоваться как односторонний вход АЦП.

Входной канал АЦП можно переключить на внутренний дифференциальный усилитель, установив регистр DITM в регистр **ADTMR**. Дифференциальные источники входного сигнала и коэффициенты усиления могут быть установлены через регистр **DAPCR**.

АЦП можно запустить, установив бит ADEN в регистр **ADCSRA**. АЦП не потребляет электроэнергию, когда ADEN очищается. Поэтому рекомендуется отключать АЦП перед выходом в спящий режим.

Результат преобразования АЦП составляет 12 бит и хранится в регистрах данных ADC **ADCH** и **ADCL**. Результат преобразования выравнивается по правому краю по умолчанию, но его можно выровнять по левому краю, установив бит **ADLAR** регистра **ADMUX**. Если вы установите результат преобразования на выравнивание по левому краю и потребуете только 8-битной точности преобразования, то достаточно читать **ADCH**. В противном случае сначала прочтите **ADCL**, а затем прочтите **ADCH**, чтобы убедиться, что содержимое регистра данных является результатом одного и того же преобразования. После чтения **ADCL** регистры данных **ADCL** и **ADCH** фиксируются.

После чтения **ADCH** результат преобразования может быть обновлен в регистрах данных **ADCL** и **ADCH**.

Конец преобразования АЦП может вызвать прерывание. Даже если конец преобразования происходит между чтением **ADCL** и **ADCH**, прерывание все равно будет запущено.

### 23.2.1. Запуск преобразования

Чтобы запустить бит преобразования ADC, напишите «1» в бит ADSC, чтобы начать одно преобразование. Этот бит остается высоким во время преобразования и очищается аппаратным обеспечением после завершения преобразования.

Если канал изменяется во время преобразования, ADC изменит канал после завершения преобразования.

Преобразования АЦП имеют разные источники запуска. Установка бита разрешения автоматического запуска АЦП ADATE регистра ADCSRA, включает автоматический запуск.

Источник запуска можно выбрать, установив бит выбора запуска АЦП ADTS регистра ADCSRB.

Когда возникает нарастающий фронт выбранного запускающего сигнала, предделитель ADC сбрасывается и начинает преобразование. Это дает возможность начать преобразование с фиксированным интервалом времени.

После преобразования, даже если сигнал запуска еще существует, новое преобразование не будет запущено. Если запускающий сигнал в процессе преобразования создает нарастающий фронт, этот передний фронт также будет проигнорирован. Даже если конкретное прерывание отключено или бит разрешения глобального прерывания равен «0», флаг прерывания будет установлен. Это вызовет преобразование без прерываний. Однако, чтобы инициировать новое преобразование в следующем событии прерывания, флаг прерывания должен быть очищен.

Используя флаг прерывания АЦП в качестве источника запуска, следующее преобразование АЦП может быть запущено после завершения текущего преобразования.

Затем АЦП работает в режиме непрерывного преобразования, непрерывно отбирает и обновляет регистры данных АЦП.

Первое преобразование инициируется путем записи «1» в бит ADSC регистра ADCSRA. В этом режиме последующие преобразования АЦП не зависят от того, установлен ли флаг прерывания ADC ADIF. Если включен автоматический запуск, ADSC устанавливает регистр ADCSRA для инициирования одного преобразования.

Флаг ADSC также может использоваться для проверки того, выполняется ли преобразование. Независимо от того, как инициируется преобразование, ADSC всегда «1» во время процесса преобразования.

### 23.2.2. Предделитель и время преобразования АЦП

По умолчанию для последовательной схемы аппроксимации для максимальной точности требуется входной тактовый сигнал от 300 кГц до 3 МГц. Если требуемая точность преобразования меньше 12 бит, входная тактовая частота может превышать 3 МГц для достижения более высокой частоты дискретизации.

Модуль ADC включает предусилитель, который может использоваться системным тактовым генератором для генерации приемлемого входного тактового сигнала АЦП.

Предделитель задается битом ADPS регистра ADCSRA.

Установка ADEN в регистре ADCSRA позволяет АЦП и предварительный

делитель начать подсчет. Пока бит ADEN равен «1», предделитель продолжает подсчет до тех пор, пока ADEN не будет очищен.

После того, как установлен ADSC регистра ADCSRA, однократное преобразование начинается с возрастающего фронта следующего тактового цикла АЦП. Для нормального преобразования требуется **15 тактовых циклов** АЦП.

После включения АЦП (бит ADEN регистра ADCSRA установлен) для инициализации аналоговой схемы до первого преобразования может потребоваться 50 тактовых импульсов АЦП.

Во время преобразования АЦП начало отсчета и удержания при входных тактовых импульсах 1,5 АЦП после начала преобразования, а результирующий выход первого преобразования АЦП происходит при 14,5 тактовых импульсах АЦП после запуска.

В конце преобразования результат АЦП подается в регистр данных АЦП и устанавливается флаг ADIF. ADSC очищается одновременно. Затем программное обеспечение может снова устанавливать флаг ADSC или автоматически запускать, чтобы инициировать новое преобразование.

### 23.2.3. Каналы выборки и опорные напряжения

MUX и REFS в регистрах ADMUX реализуют одиночную буферизацию через временные регистры. ЦП может осуществлять произвольный доступ к временными регистрам. Перед началом преобразования ЦП может в любой момент настроить выбор каналов и источников ссылок.

Для обеспечения АЦП имеет достаточно времени выборки, после запуска преобразования, вы не можете настроить выбранный канал и ссылки.

После того, как (АДИФ в ADCSRA) При преобразовании завершен, выбранный канал и ссылки будут обновлены.

Начало преобразования - это нарастающий фронт следующих входных тактов АЦП после установки ADSC. Таким образом, пользователю рекомендуется после ADSC вход АЦП в пределах ADMUX тактового цикла не работают, чтобы выбрать новый канал и ссылки.

При использовании автоматического запуска время, в течение которого происходит событие запуска, является неопределенным. Чтобы контролировать влияние новых настроек на преобразование, необходимо соблюдать особую осторожность при обновлении регистров ADMUX.

Если установлены оба параметра ADATE и ADEN, время прерывания может быть выполнено в любое время, автоматически активируя преобразование АЦП. Если вы измените содержимое регистра ADMUX в течение этого времени, пользователь не сможет определить, будет ли следующее преобразование основано на старой или новой конфигурации.

**Рекомендуется**, чтобы пользователь обновил ADMUX в следующие моменты безопасности:

- 1) бит ADATE или ADEN равен «0»;
- 2) по крайней мере один тактовый цикл входа ADC во время преобразования, но после события триггера;
- 3) после завершения преобразования, но после, перед тем как флаг прерывания источника запущен.

Если обновлен ADMUX, в любом из вышеупомянутых случаев, новая конфигурация вступит в силу до следующего преобразования.

При выборе входного канала АЦП **важно отметить**, что канал выбран до начала преобразования. Новый аналоговый входной канал может быть выбран после одного периода синхронизации АЦП после утверждения ADSC, но самый **простой способ** - дождаться завершения преобразования до изменения канала.

В АЦП источник опорного напряжения VREF указывает диапазон преобразования АЦП. Если уровень одностороннего канала превышает VREF, результат преобразования будет близок к максимальному значению 0xFFFF. VREF может быть AVCC, внешнее напряжение AREF, внутреннее задание. Используйте внутреннюю ссылку (1.024V / 2.048V / 4.096V)

**Примечание:** После включения чипа внутренняя ссылка откалибрована по умолчанию до 1.024 В. Если пользователь использует внутреннюю ссылку 1.024V, ее можно использовать напрямую без какой-либо другой операции. Но если вам нужно использовать внутреннее опорное напряжение 4.096V или 2.048V, самокалибровка требует более нового внутреннего опорного значения.

Калибровочное значение 2.048V / 4.096V загружается в регистр VCAL2 / 3 (0xCE / 0xCC) после включения питания. Во время инициализации программы значение VCAL2 / 3 считывается и записывается в регистр VCAL (0XC8) для завершения калибровки.

#### 23.2.4. Автоматический мониторинг каналов.

Режим автоматического контроля канала используется для контроля изменений напряжения выбранных входных каналов АЦП в режиме реального времени.

Программное обеспечение позволяет автоматически контролировать канал, устанавливая бит AMEN в регистре ADCSRC.

АЦП автоматически преобразует напряжение выбранного канала. Когда результат преобразования выходит за пределы заданного диапазона переполнения, будет установлен бит флага прерывания АЦП (ADIF). В то же время прекратите автоматический мониторинг.

Программное обеспечение может реагировать на события переполнения прерываниями или запросами. Бит AMOF регистра ADMSC используется для указания типа события переполнения. Флаг ADIF автоматически очищается аппаратным обеспечением в ответ на сброс прерывания, его можно очистить программным обеспечением 1 в режиме опроса.

Режим автоматического мониторинга можно повторно включить, только если ADIF очищен и установлен бит AMEN в регистре ADCSRC.

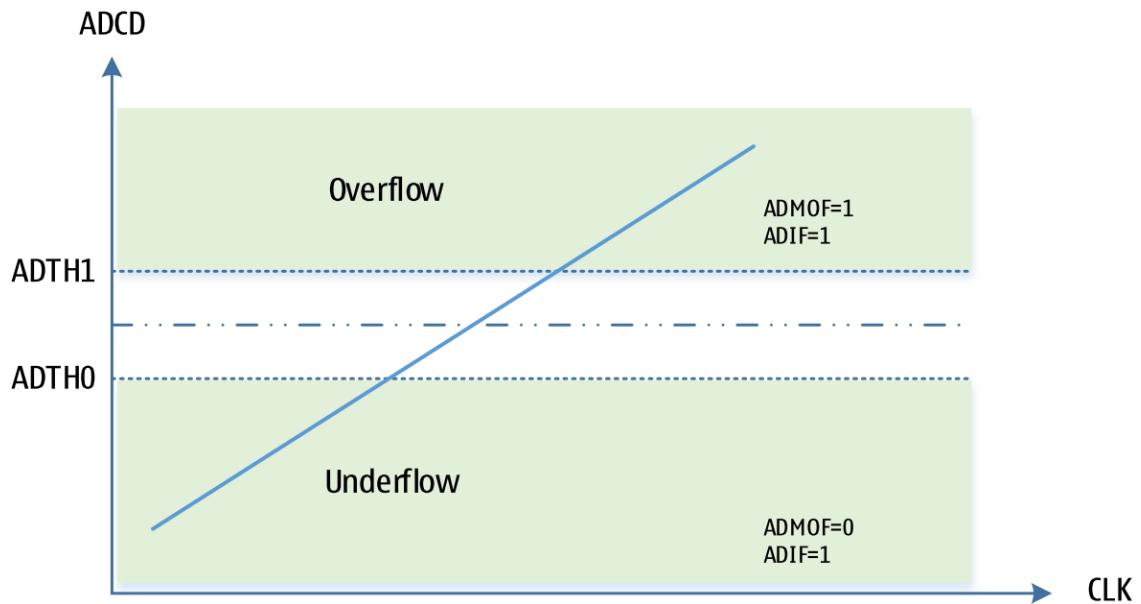


Рис. 23.2.

Чтобы преодолеть нестабильность результатов преобразования одного АЦП, автоматическое обнаружение поддерживает настраиваемую функцию цифровой фильтрации.

Цифровая фильтрация обнаруживает результаты непрерывного преобразования и запускает событие переполнения только в том случае, если получается последовательный результат для ограниченного числа последовательных преобразований.

Количество последовательных преобразований может быть установлено битами AMFC [3: 0] в регистре ADMSC. Автоматический контроль канала контролируется битом AMEN в регистре ADCSRC. Регистр ADT0 используется для установки порога нижнего потока: ADT1 используется для установки порога переполнения. ADT0 / 1 - это 16-разрядный регистр.

Программное обеспечение устанавливает бит AMEN и немедленно остановит текущее преобразование АЦП и сбросит состояние управления АЦП перед входом в режим автоматического преобразования.

Перед запуском режима автоматического определения канала вам необходимо установить обнаруженный канал и другие связанные с ним конфигурации.

Программное обеспечение может отключить режим автоматического обнаружения в любое время, очистив регистр AMEN.

### 23.2.5. Делитель с несколькими входами напряжения (VDS)

АЦП внутренне содержит модуль делителя входного напряжения. Разделив входное напряжение от внешнего источника дополнительного АЦПА входного канала (АЦП0 / 1/4/5), внешнего опорного AVREF или аналогового источника питания.

Модуль делителя напряжения одновременно выводит 4/5 и 1/5 напряжения на внутренние 12 и 13 входных каналов АЦП. Среди них 4/5 в основном используется для калибровки смещения АЦП, 1/5 используется для калибровки внутреннего

смещения и в основном используется для определения напряжения питания и подобных приложений.

Функции, связанные с делителем напряжения, в основном управляются регистром ADCSRD.

### 23.2.6. Калибровка смещения АЦП

Из-за изменений в производственном процессе и присущих характеристикам структуры схемы внутренняя схема компаратора АЦП будет производить разные уровни ошибки смещения. Поэтому компенсация напряжения смещения имеет решающее значение для создания высокоточной структуры преобразования АЦП.

АЦП внутри чипа LGT8FX8P поддерживает интерфейс, связанный с напряжением смещения, который может использоваться для завершения измерения и калибровки смещения.

**Принцип калибровки смещения:**

Калибровка смещения, в основном, предназначена для проверки результата преобразования АЦП, как в положительном, так и в отрицательном направлениях, путем изменения входной полярности внутреннего компаратора.

Поскольку напряжение смещения, как в положительном, так и в отрицательном направлениях, также выражается в двух полярностях, среднее значение ошибки смещения может быть получено путем вычитания двух результатов преобразования.

В нормальном приложении результат преобразования может быть скорректирован в соответствии с этим напряжением смещения.

**Смещение процедуры калибровки:**

1. Установить модуль VDS, выбранный источник входного сигнала аналогового VDS мощность (AVCC)

2. Выбор АЦП опорного напряжения в аналоговый источник питания (AVCC)

3. ADCSRC [SPN] = 0, АЦП чтение 4 / 5VDO канала Преобразованное значение записывается как PVAL

4. ADCSRC [SPN] = 1, АЦП считывает канал 4 / 5VDO, бит записи с преобразованной величиной NVAL

5. Сохраняет значение (NVAL - PVAL) >> 1 - регистр OFR0

6. ADCSRC [SPN] = 1, АЦП считывает канал 1 / 5VDO, результат преобразования записывается как NVAL

7. ADCSRC [SPN] = 0, АЦП читает канал 1 / 5VDO, бит записи результата преобразования PVAL

8. Будет ли значение (NVAL - PVAL) >> 1 Сохранить в регистр OFR1

9. Установите ADCSRC [OFEN] = 1, чтобы включить функцию компенсации смещения.

**Специальное примечание:**

Поскольку ошибка смещения имеет положительные и отрицательные направления, вышеуказанные данные и операции являются **полностью  
подписанными**.

Конфигурация, связанная с АЦП, должна быть изменена во время калибровки

смещения, поэтому рекомендуется выполнить калибровку смещения до нормальной конфигурации.

Чтобы повысить точность калибровки, рекомендуется, чтобы АЦП считывал многопроходную фильтрацию во время преобразования канала.

Калибровка смещения после завершения конфигурации OFR0 / 1 автоматическая компенсация смещения активируется через бит OFEN.

После нормального преобразования в будущем контроллер АЦП будет автоматически использовать OFR0 / 1 для компенсации в соответствии с результатом преобразования АЦП.

### 23.2.6. Динамическая калибровка АЦП.

Метод калибровки смещения, описанный выше, основан на смещении в тестовой среде и тестовом входе. При изменении системной среды смещение АЦП также изменится. Поэтому, если возможна компенсация калибровки в реальном времени, очень важно улучшить точность измерения АЦП, чтобы преодолеть разницу в производительности, вызванную изменениями в устройстве с рабочей средой. Здесь используется предлагаемый алгоритм.

Основываясь на принципе алгоритма калибровки смещения, он может динамически компенсировать ошибку смещения, вызванную рабочей средой, и получать согласованные и точные результаты испытаний. Этот метод исключает необходимость вычисления напряжения смещения и отключения компенсации смещения (OFEN).

Алгоритм должен только контролировать полярность преобразования АЦП через SPN.

Два результата измерения отбираются под разными SPN. Ошибка, введенная смещением в обоих результатах, является положительной и отрицательной, поэтому мы можем просто добавить среднее значение.

Метод противодействует ошибке, вызванной дисбалансом. Мы предполагаем, что когда АЦП преобразует, смещение вводит тестовую ошибку VOFS, поэтому SPN управления выполняет два последовательных преобразования АЦП.

Результат преобразования АЦП может быть выражен как:

Когда  $SPN = 1$ ,  $VADC1 = VREL + VOFS1$   $SPN = 0$ ,  $VADC0 = VREL - VOFS0$ .

Мы добавляем два результата измерения, чтобы исключить влияние VOFS на фактический вход для образца VREL.

Из-за совпадающих характеристик схемы VOFS1 и VOFS0 могут быть не совсем одинаковыми, но эффект компенсации смещения может быть достигнут в целом.

Алгоритм динамического смещения коррекции:

1. В соответствии с необходимостью инициализации параметров преобразования АЦП

2. Установите  $SPN = 1$ , запустите выборку АЦП, запишите результат выборки АЦП в виде  $VADC1$

3. Установите  $SPN = 0$ , запустите выборку АЦП, запишите результат выборки АЦП в виде  $VADC2$  ( $VADC1 + VADC2) >> 1$

Это практическое применение результата преобразования АЦП.

Этот алгоритм можно комбинировать с алгоритмом выборки, чтобы получить результат, более идеальный.

### 23.3. Регистры Описание

#### Список регистров ADC

Регистр	Адрес	Значение по умолчанию	Описание
ADCL	0x78	0x00	Регистр данных - младший байт АЦП
ADCH	0x79	0x00	Регистр данных - старший байт АЦП
ADCSRA	0x7A	0x00	Регистр A управления и состояния АЦП
ADCSR <sub>B</sub>	0x7B	0x00	Регистр B управления и состояния АЦП
ADMUX	0x7C	0x00	Регистр управления мультиплексора АЦП
ADCSRC	0x7D	0x01	Регистр C управления и состояния АЦП
DIDR <sub>0</sub>	0x7E	0x00	Регистр управления 0 Цифровым входом (Откл)
DIDR <sub>1</sub>	0x7F	0x00	Регистр управления 1 Цифровым входом (Откл)
DAPCR	0xDC	0x00	Регистр управления дифференциальным усилителем
OFR <sub>0</sub>	0xA3	0x00	Регистр компенсации смещения 0
OFR <sub>1</sub>	0xA4	0x00	Регистр компенсации смещения 1
ADT0L	0xA5	0x00	Автоматический контроль нижнего порога младший байт
ADT0H	0xA6	0x00	Автоматический контроль нижнего порога старший байт
ADT1L	0xAA	0x00	Автоматический контроль верхнего порога младший байт
ADT1H	0xAB	0x00	Автоматический контроль верхнего порога старший байт
ADMSC	0xAC	0x01	Регистр контроля состояния канала АЦП
ADCSR <sub>D</sub>	0xAD	0x00	Регистр D управления и состояния АЦП

#### ADCL - Регистр данных - младший байт АЦП

ADCL - Регистр данных - младший байт АЦП								
Адрес: 0x78			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя 0	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0
Имя 1	ADC3	ADC2	ADC1	ADC0				
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	ADC[7:0]/ADC[3:0]	Регистр младшего байта данных ADC. Когда бит ADLAR в «0», выходные данные АЦП сохраняются в регистре с выравниванием ADC0 по биту 0, то есть ADCL является ADC [7: 0], как показано в Name 0. Когда бит ADLAR в «1», выходные данные АЦП находятся в регистре с выравниванием ADC0 по биту 4, то есть верхние 4 бита ADCL - это ADC [3: 0], а нижние 4 бита не имеют смысла, как показано в Name 1.						

#### ADCH - Регистр данных - старший байт АЦП

ADCH - Регистр данных - старший байт АЦП								
Адрес: 0x78			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0

*ADCSRA - Регистр А управления и состояния АЦП*

ADCSRA - Регистр А управления и состояния АЦП									
Адрес: 0x7A			Значение по умолчанию: 0x05						
Бит	7	6	5	4	3	2	1	0	
Имя	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Бит	Имя	Определение бит							
7	ADEN	Бит управления включением АЦП. Когда бит ADEN установлен на «1», АЦП включен. Когда бит ADEN установлен на «0», АЦП отключен.							
6	ADSC	Бит начала преобразование АЦП. В режиме одиночного преобразования настройка ADSC инициирует преобразование. В режиме непрерывного преобразования ADSC настроен на инициирование первого преобразования.							
5	ADATE	Бит запуска включением автоматически управления АЦП. Когда бит ADATE установлен на «1», активируется функция автоматического запуска. Передний фронт выбранного запускающего сигнала включает преобразование. Выбор источника запуска контролируется ADTS регистра ADCSRB. Когда бит ADATE установлен на «0», функция автоматического запуска отключена.							
4	ADIF	Флаг прерывания АЦП. ADIF устанавливается, когда ADC завершает преобразование и обновляет регистр данных. Если бит разрешения прерывания АЦП ADIE равен «1», а глобальное прерывание установлено, генерируется прерывание АЦП. Выполнение прерывания АЦП очищает бит ADIF и также можно записать «1» в этот бит, чтобы очистить его.							
3	ADIE	Бит управления разрешением прерывания АЦП. Когда бит ADIE установлен на «1», и глобальное прерывание установлено, прерывание АЦП включено. Когда бит ADIE установлен на «0», прерывание АЦП отключается.							
2:0	ADPS[2:0]	Бит выбора коэффициент деления предделителя ADC. ADPS выбирает коэффициент деления системных тактов для генерации тактов АЦП.							
		ADPS[2:0]		Коэффициент деления					
		0		2					
		1		2					
		2		4					

		3	8
		4	16
		5	32 (по умолчанию)
		6	64
		7	128

### ADCSRB - Регистр В управления и состояния АЦП

ADCSRB - Регистр В управления и состояния АЦП									
Адрес: 0x7B				Значение по умолчанию: 0x00					
Бит	Имя	7	6	5	4	3	2	1	0
Имя	CME01	CME00	CME11	CME10	ACTS	ADTS2	ADTS1	ADTS0	
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит							
7	CME01	Выбор инверсного входа AC0, CME0 = {CME01, CME00} 00: Внешний порт ACXN - инверсный вход AC0 01: Мультиплексированный выход ADC - инверсный вход AC0 10: Выход дифференциального усилителя - инверсный вход AC0 11: Отключите источник инверсный входного сигнала AC0							
6	CME00								
5	CME11	Выбор инверсного входа AC1, CME1 = {CME11, CME10} 00: Внешний порт ACXN - инверсный вход AC1 01: Внешний порт AC1N используется - инверсный вход AC1 10: Внутренний делитель напряжения АЦП, - инверсный вход AC1 11: Выход дифференциального усилителя - инверсный вход AC1							
4	CME10								
3	ACTS	Выбор канала источника переменного тока 0 - Выход AC0 как источник запуска автоматического преобразования АЦП 1 - Выход AC1 как источник запуска автоматического преобразования АЦП							
2	ADTS2	АЦП автоматически запускает бит управления выбора источника. Когда бит ADATE установлен в «1», функция автоматического запуска включена, а выбор источника триггера управляемся ADTS. Если бит ADATE установлен на «0», параметр ADTS недействителен. Передняя кромка выбранного интервала прерывания сигнала промежуточного триггера включает преобразование. Когда источник триггера переключается с флага прерывания на чистый источник триггера на источник запуска триггера прерывания, триггерный сигнал будет генерировать нарастающий фронт.							
1	ADTS1	Если ADEN установлен, АЦП также начнет преобразование. При переключении в режим непрерывного преобразования (ADTS = 0) функция автоматического триггера отключена.							
ADTS[2:0]					Источник запуска				
0					Непрерывный режим преобразования				
1					Компаратор 0/1				
2					Внешнее прерывание				
3					Счетчик таймера 0 Сравнить Равенство				
4					Счетчик таймера 0 Переполнение				
5					Счетчик таймера 1 Равенство Сравнить В				
6					Счетчик таймера 1 Переполнение				
7					Счетчик таймера 1 Событие захвата ввода				

## ADMUX - Регистр управления мультиплексора АЦП

ADMUX - Регистр управления мультиплексора АЦП												
Адрес: 0x7C				Значение по умолчанию: 0x00								
Бит	7	6	5	4	3	2	1	0				
Имя	REFS1	REFS0	ADLAR	CHMUX4	CHMUX3	CHMUX2	CHMUX1	CHMUX0				
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W				
Бит	Имя	Определение бит										
7	REFS1	С битом REFS2 регистра ADCSRD используется в АЦП для выбора источника опорного напряжения путем установки бита управления для выбора REFS. Настройка не вступает в силу до тех пор, пока текущее преобразование не будет завершено.										
		REFS2	REFS1	REFS0	Выбор опорного напряжения							
		0	0	0	AREF							
		0	0	1	AVCC							
		0	1	0	2.048V							
		0	1	1	1.024V							
		1	0	0	4.096V							
5	ADLAR	Результат преобразования выравнивается по левому или по правому краю. Когда бит ADLAR установлен в «1», результат преобразования выравнивается по левому краю в регистре данных АЦП. Когда бит ADLAR установлен в «0», результат преобразования выравнивается по правому краю в регистре данных ADC.										
4	CHMUX4	Биты управления выбором источника входного сигнала АЦП.										
			CHMUX[3:0]	Источник входного сигнала	Описание							
3	CHMUX3		0	0000	PC0	Вход внешнего порта						
			0	0001	PC1							
			0	0010	PC2							
			0	0011	PC3							
			0	0100	PC4							
			0	0101	PC5							
2	CHMUX2		0	0110	PE1	Вход внешнего порта						
			0	0111	PE3							
			0	1001	PC7							
			0	1010	PF0							
			0	1011	PE6							
			0	1100	PE7							
1	CHMUX1		0	1101	4/5VDO	Схема внутреннего деления						
			0	1000	1/5VDO							
			0	1110	IVREF	Внутреннее опорное напряжение						
			0	1111	AGND							
0	CHMUX0		1	xxxx	DACO	Имитированная земля Внутренний вывод ЦАП						

## ADCSRC - Регистр С управления и состояния АЦП

ADCSRC - Регистр С управления и состояния АЦП								
Адрес: 0x7D				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0

### DIDR0 - Регистр управления 0 Цифровым входом (Откл)

*DIDR1 - Регистр управления 1 Цифровым выходом (Откл)*

1	PD7D	1 = Выкл. Функция цифрового входа PD7
0	PD6D	1 = Выкл. Функция цифрового входа PD6

DAPCR - Регистр управления дифференциальным усилителем

### *OFR0 - Регистр компенсации смещения 0*

### *OFR1 - Регистр компенсации смещения 1*

R/W		R/W
Бит	Имя	Определение бит
7:0	OFR1	Компенсация смещения регистра 1; в OFR1 как знакового числа. Хранить в двоичном дополнительном формате

### *ADTOL - Автоматический контроль нижнего порога - младший байт*

ADT0L - Автоматический контроль нижнего порога - младший байт								
Адрес: 0xA5				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	ADT0L[7:0]							
R/W	R/W							
Бит	Имя	Определение бит						
7:0	ADT0L	Автоматический контроль нижнего порога переполнения регистра младшие 8 бит						

### *ADTOH - Автоматический контроль нижнего порога старший байт*

ADT0H - Автоматический контроль нижнего порога - старший байт								
Адрес: 0xA6				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	ADT0L[7:0]							
R/W	R/W							
Бит	Имя	Определение бит						
7:0	ADT0H	Автоматический контроль нижнего порога переполнения регистра старшие 8 бит						

### *ADT1L - Автоматический контроль верхнего порога младший байт*

ADT1L - Автоматический контроль верхнего порога - младший байт								
Адрес: 0xAA				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	ADT0L[7:0]							
R/W	R/W							
Бит	Имя	Определение бит						
7:0	ADT1L	Автоматический контроль верхнего порога переполнения регистра младшие 8 бит						

### *ADT1H - Автоматический контроль верхнего порога старший байт*

ADT1H - Автоматический контроль верхнего порога - старший байт								
Адрес: 0xAB				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	ADT0L[7:0]							
R/W	R/W							
Бит	Имя	Определение бит						
7:0	ADT1H	Автоматический контроль верхнего порога переполнения регистра старшие 8 бит						

### *ADMSC - Регистр контроля состояния канала АЦП*

ADCSRD - Регистр контроля состояния канала АЦП								
Адрес: 0xAC				Значение по умолчанию: 0x01				

*ADCSRД - Регистр D управления и состояния АЦП*

## **VCAL - Регистр внутренний контрольный калибровочный**

		напишите VCAL2 в регистр, чтобы завершить калибровку внутренней ссылки 2.048V.
--	--	--------------------------------------------------------------------------------

### VCAL1 – 1,024 - калибровочный регистр

VCAL1 - калибровочный регистр								
Адрес: 0xCD				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	VCAL1[7:0]							
R/W	R/W							
Бит	Имя	Определение бит						
7:0	VCAL1	Внутренний опорный калибровочный коэффициент для 1.024V						

### VCAL2 – 2,048 - калибровочный регистр

VCAL2 - калибровочный регистр								
Адрес: 0xCE				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	VCAL2[7:0]							
R/W	R/W							
Бит	Имя	Определение бит						
7:0	VCAL2	Внутренний опорный калибровочный коэффициент для 2.048V						

### VCAL3 – 4,096 - калибровочный регистр

VCAL3 - калибровочный регистр								
Адрес: 0xCC				Значение по умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	VCAL3[7:0]							
R/W	R/W							
Бит	Имя	Определение бит						
7:0	VCAL3	Внутренний опорный калибровочный коэффициент для 4.096V						

## 24. Регистры Краткий Справочник Таблица

Адр	Имя	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
<b>Расширенный журнал ввода-вывода</b>									
F6	GUID3								GUID Байт 3
F5	GUID2								GUID Байт 2
F4	GUID1								GUID Байт 1
F3	GUID0								GUID Байт 0
F2	PMCR	PMCE	CLKFS	CLKSS	WCLKS	OSCKEN	OSCMEN	RCKEN	RCMEN
F0	PMX2	WCE	STOSC1	STOSC0	-	-	XIEN	E6EN	C6EN
EE	PMX0	PMXCE	C1BF4	C1AF5	C0BF3	C0AC0	SSB1	TXD6	RXD5
ED	PMX1	-	-	-	-	-	C3AC	C2BF7	C2AF6
EC	TCKSR	-	F2XEN	TC2XF1	TC2XF0	-	AFCKS	TC2XS1	TC2XS0
E2	PSSR	PSS1	PSS3	-	-	-	-	PSR3	PSR1
E1	OCPUE	PUE7	PUE6	PUE5	PUE4	PUE3	PUE2	PUE1	PUE0
E0	HDR	-	-	HDR5	HDR4	HDR3	HDR2	HDR1	HDR0







22	DSSD	DSSD порт доступа uDSC							
21	DSIR	Реестр инструкций для uDSC							
20	DSCR	DSUEN	MM	D1	D0	-	DSN	DSZ	DSC

## 25. Набор инструкций краткий Справочник Таблица

См. **LGT8FX8P Series – Programming Manual 1.0.1**

