

1. Микроконтроллер RISC с Программируемой внутренней Флэш-памятью

LGT8FX8D Series – FLASH MCU Overview v1.0.1

От переводчика: Переводил для себя, так что некоторые места не редактировались. Явные ошибки были исправлены. Предполагаемые ошибки не редактировались. Жмаев А.В.

1.1. Обзор функций

- Высокопроизводительное и маломощное 8-битное ядро LGT8XM
 - Расширенная архитектура RISC
 - 131 инструкция, более 80% для однократного исполнения
 - 32x8 рабочих регистров общего назначения
 - Эффективность выполнения до 16MIPS при работе 16 МГц
 - Внутренний умножитель одиночного цикла (8x8)
- Энергонезависимая программа и пространство для хранения данных
 - 4K / 8K / 16K / 32Kbytes встроенная программируемая встроенная программная память FLASH
 - 512 / 1K / 1K / 2Kbytes внутренний SRAM
 - Программируемый аналоговый интерфейс E2PROM с байтовым доступом
 - Новый алгоритм шифрования программ для обеспечения безопасности кода пользователя
- Периферийный контроллер
 - Два 8-битных таймера с независимой поддержкой предварительного делителя и режим сравнения выхода;
 - 16-разрядный таймер с независимым предварительным делителем, ввода и сравнения вывода;
 - Встроенный калибровочный RC-генератор (32KHz) для функции счетчика реального времени;
 - Поддерживает до 6 выходных сигналов PWM, программируемый тайм-контроль мертвого времени;
 - 8-канальный, 12-битный высокоскоростной аналого-цифровой преобразователь (АЦП);
 - Два аналоговых компаратора (АС) с расширением от входных каналов АЦП;
 - Два операционных усилителя с фиксированным усилением (ОРА) в качестве интерфейсных входов для ADC / АС;
 - Внутренний калиброванный опорный источник напряжения 1,25V/2,56V $\pm 1\%$;
 - Два 8-разрядных ЦАП могут использоваться для генерации опорных источников напряжения;
 - Программируемый сторожевой таймер (WDT);
 - Программируемый синхронный / асинхронный последовательный

- интерфейс (USART);
- Синхронный периферийный интерфейс (SPI), программируемая ведущая/ведомая операция;
 - Программируемый двухпроводный последовательный интерфейс (TWI), совместимый с ведущим-ведомым режимом I2C;
 - Специальные функции процессора
 - SWD двухпроводный интерфейс для отладки / производства;
 - Поддержка внешнего прерывания и поддержка прерывания уровня ввода/вывода;
 - Встроенная схема сброса питания (POR) и трехуровневая схема обнаружения низкого напряжения (LVD);
 - Встроенный 1% откалиброванный 32-мегагерцовый генератор;
 - Встроенный 1% откалиброванный 32KHz RC-генератор;
 - Внешняя поддержка 32,768KHz и 400K ~ 20MHz Crystal Inputs
 - I / O и упаковка
 - QFP32L (доступно до 30 входов / выходов);
 - SSOP28 / 24/20/16
 - Рабочая среда
 - Рабочее напряжение: 1,8 В ~ 5,5 В;
 - Рабочая частота: 0 ~ 20 МГц;
 - Рабочая температура: -40C ~ + 85C;
 - HBM ESD:> ± 4000 В.

1.2. Структура микросхемы

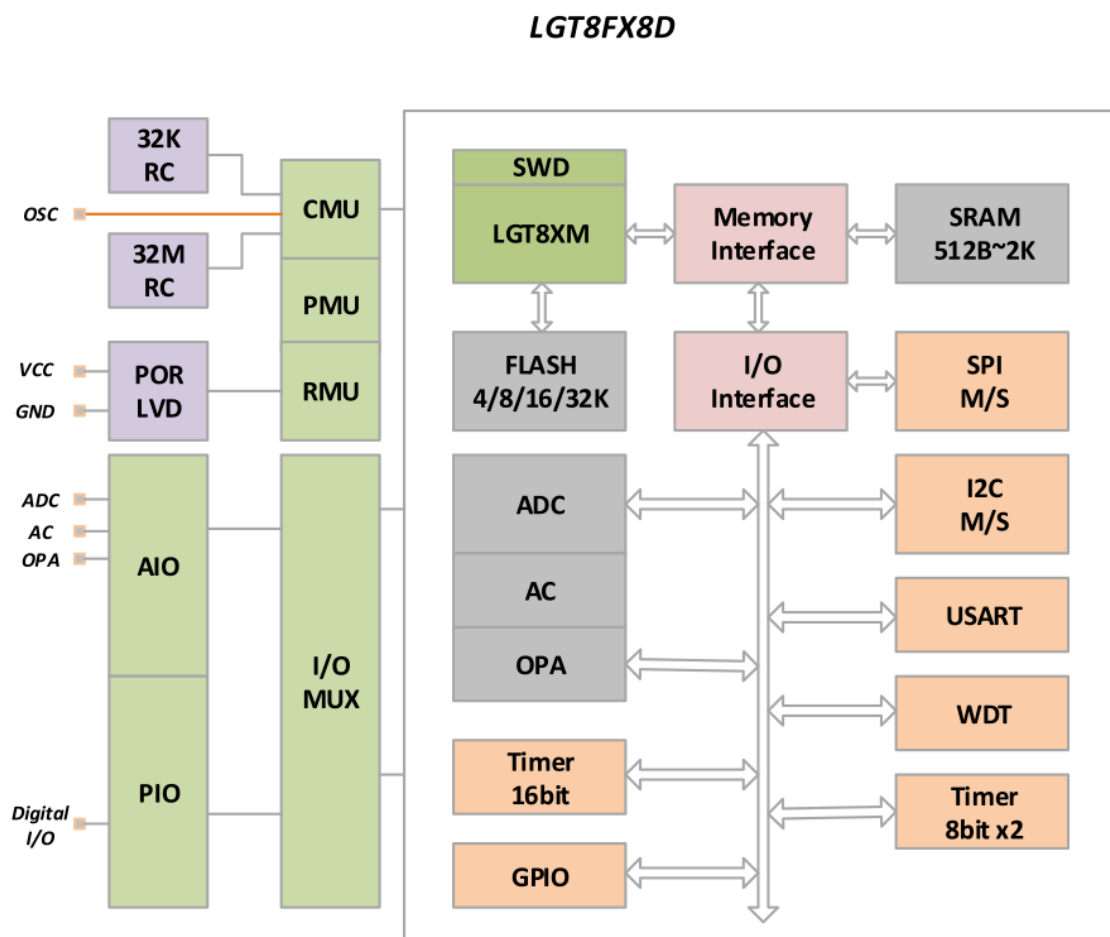


Рис.1.1. Структура микросхемы

Таблица 1.1. Описание структуры микросхемы

Имя модуля	Функция модуля
SWD	Модуль отладки и одновременно выполнять онлайн-отладку и функцию ISP
LGT8XM	8-разрядное высокопроизводительное ядро RISC
CMU	Модуль управления тактами генерирует различные рабочие такты, требуемые системой
PMU	Модуль управления питанием, ответственный за управление переходом между рабочими состояниями системы
RMU	Модуль сброса генерации
POR/LVD	Модуль сброса при включении и схема обнаружения низкого напряжения
ADC	8-канальный 10-разрядный аналого-цифровой преобразователь
AC	Аналоговый компаратор
OPA	Операционный усилитель
Timer	Таймер / Счетчик
WDT	Модуль сторожевого таймера
SPI M/S	Контроллер SPI Ведущий - Водомый
I2C M/S	Контроллер I2C Ведущий - Водомый
USART	Синхронные / асинхронные последовательные трансиверы
AIO	Входной канал АЦП
PIO	Программируемый цифровой ввод-вывод

1.3. Корпуса микросхемы

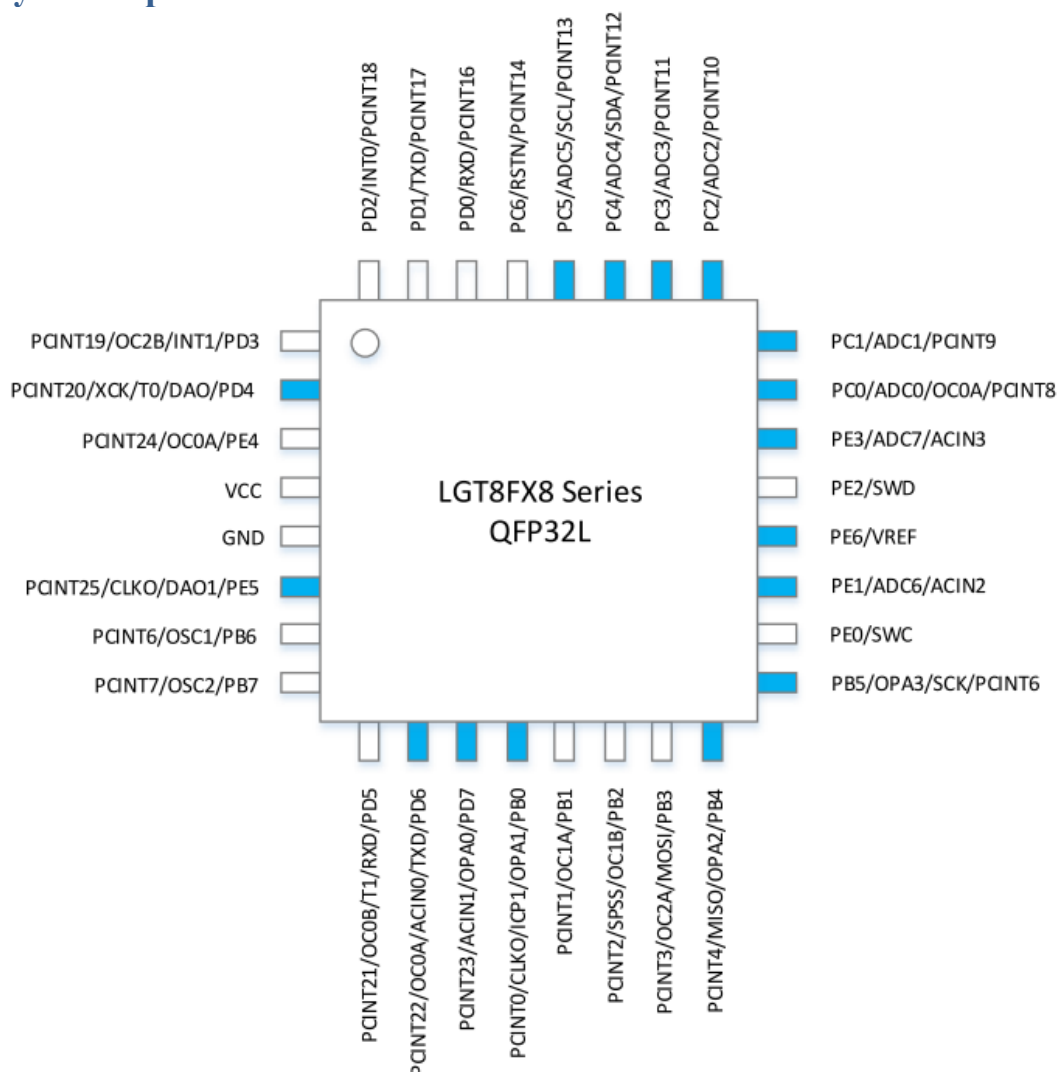


Рис.1.2. Корпус QFP32L

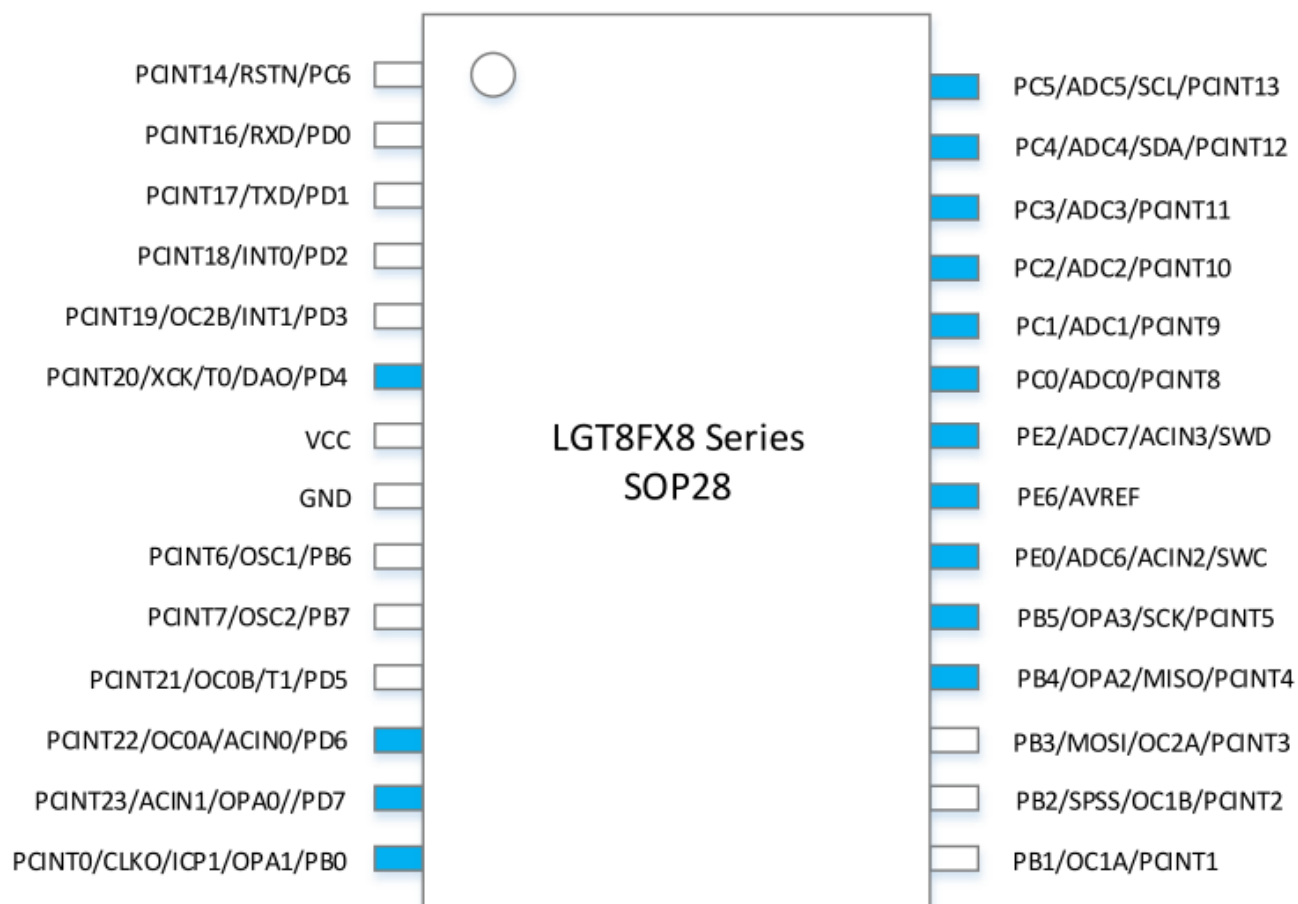


Рис.1.3. Корпус SOP28

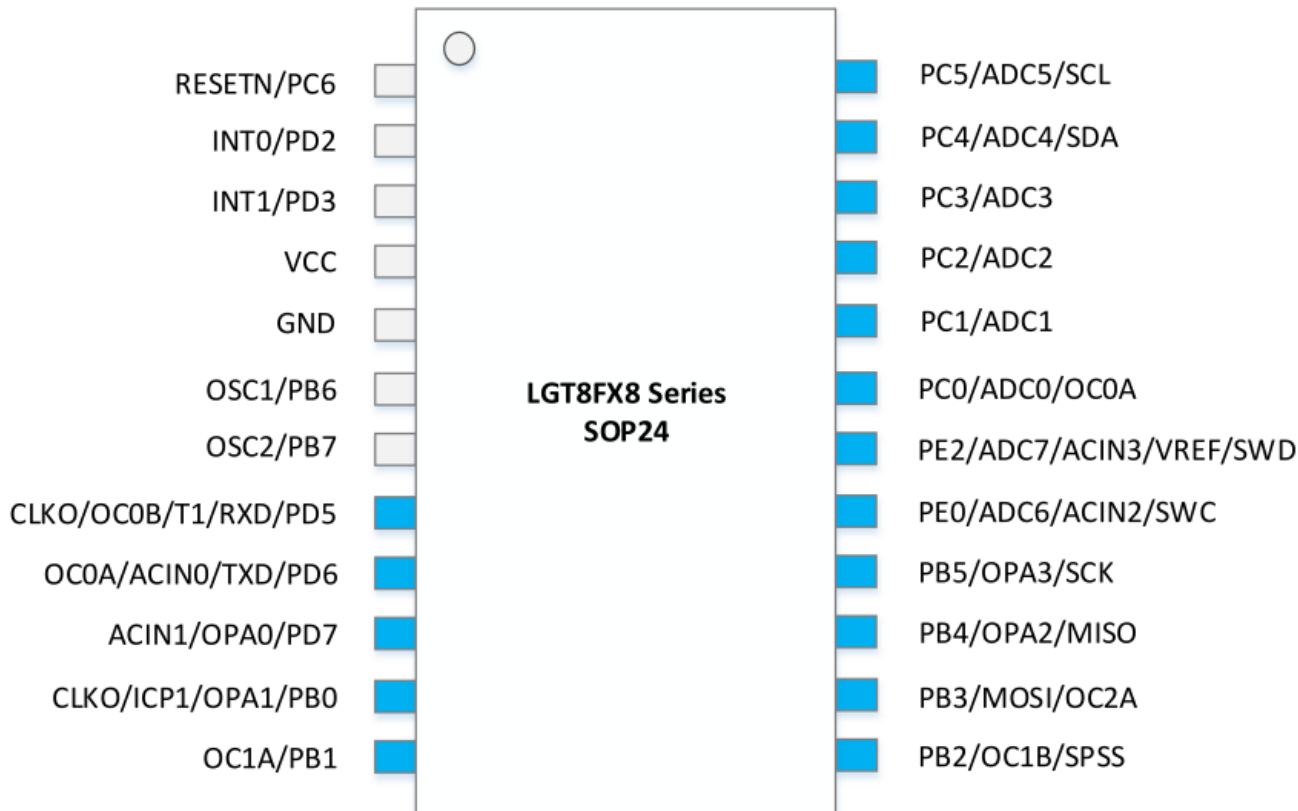


Рис.1.4. Корпус SOP24

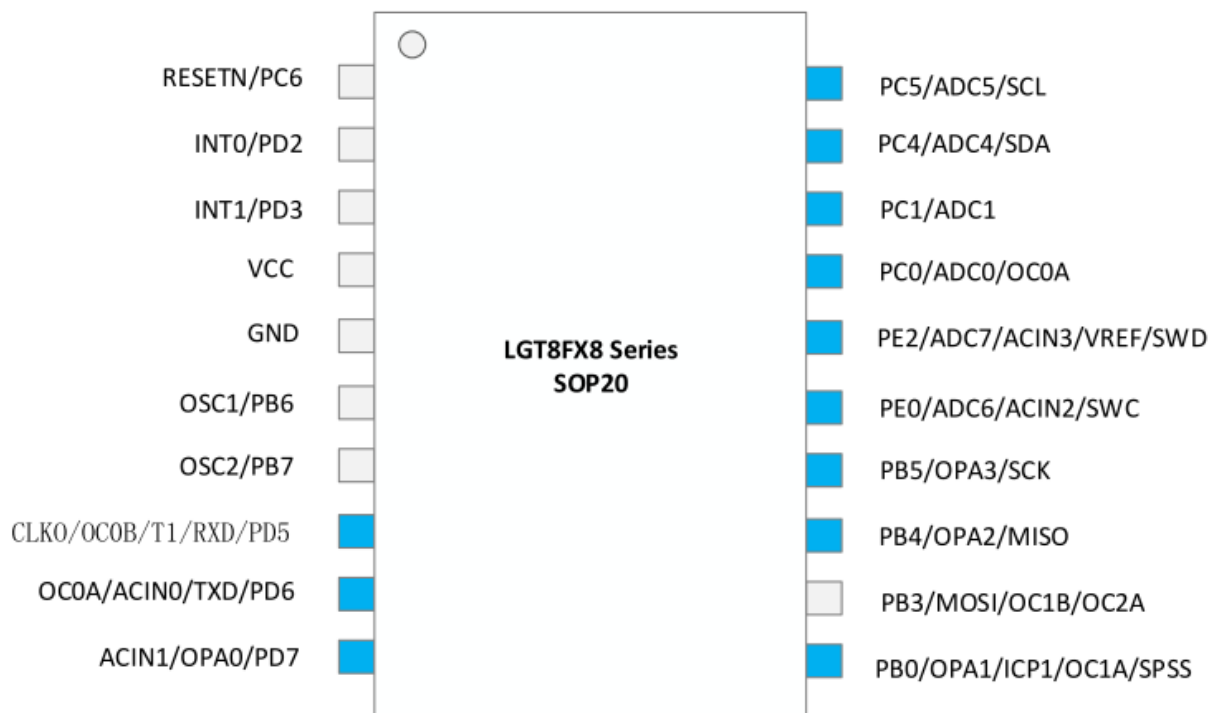


Рис.1.5. Корпус SOP20

Таблица 1.2. Описание контакта

Имя контакта	Функциональное описание
Vcc	Питание системы (1,8 В ~ 5,5 В)
GND	Общий Питания Система
OSC1	Вход внешнего кристалла
OSC2	Выход внешнего кристалла
RSTN	Вход внешнего асинхронного сброса
RXD	Синхронный / асинхронный интерфейс UART
TXD	
XCK	
INT0/1	Вход внешнего прерывания, источник асинхронного пробуждения
OC0A/B	Таймер 0 выход сравнения (PWM0A/B)
OC1A/B	Таймер 1 выход сравнения (PWM1A/B)
OC2A/B	Таймер 2 выход сравнения (PWM2A/B)
SCL	TWI двухпроводный интерфейс данных (I2C)
SDA	
SCK	
SPSS	Интерфейс SPI
MISO	
MOSI	
T0	
T1	Таймер 1 Вход внешних тактов
ICP1	Таймер 1 Вход для внешнего захвата
SWD/SWC	Интерфейс отладки SWD
PCINTX	Функция прерывания изменения уровня сигнала
ADC7...0	Входные каналы АЦП
DAO0/1	Выходные каналы ЦАП
VREF	Внешний источник опорного напряжения вход АЦП

AIN0/1	Аналоговый компаратор 0 Внешний вход
AIN2/3	Аналоговый компаратор 1 Внешний вход
OPA0/1	Операционный усилитель 0 Внешний вход
OPA2/3	Операционный усилитель 1 Внешний вход
CLKO	Выходной сигнал тактов системы
PB7...0	Программируемый ввод-вывод
PD7...0	Программируемый ввод-вывод
PC6...0	Программируемый ввод-вывод
PE6...0	Программируемый ввод-вывод

2. Ядро LGT8XM

- Маломощная конструкция
- Высокоэффективная архитектура RISC □
- 131 инструкций, 80% из которых одноктактные
- Поддержка встроенной внутрисхемной отладки (OCD)

2.1. Обзор

В этой главе в основном описывается архитектура и функции ядра LGT8XM. Ядро является мозгом MCU и отвечает за правильное выполнение программы, поэтому ядро должно быть в состоянии выполнять вычисления точно, контролировать периферийные устройства и обрабатывать различные прерывания. На Рис.2.1. показана структура ядра LGT8XM.

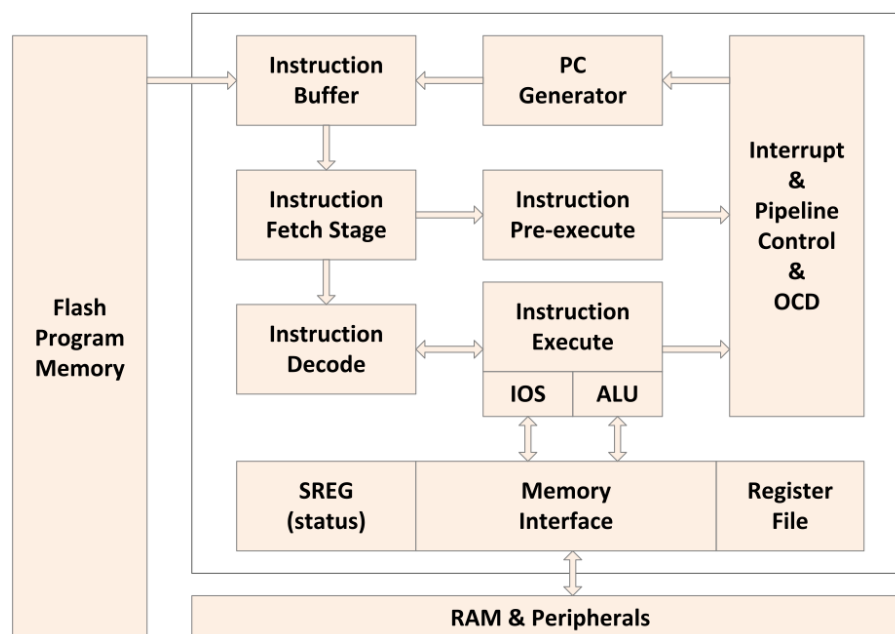


Рис.2.1. Структура ядра LGT8XM

Для достижения большей эффективности и параллелизма ядро LGT8XM использует архитектуру Navel - независимую шину данных и программ.

Инструкции выполняются с помощью оптимизированного двухступенчатого конвейера. Двухступенчатый конвейер может уменьшить количество недопустимых инструкций в конвейере и уменьшить объем доступа к памяти в

программную память FLASH, поэтому энергопотребление работы ядра может быть уменьшено.

В то же время в ядро LGT8XM добавлен буфер команд (может одновременно выполнять заказы 2 инструкций) на предыдущем этапе извлечения команды. Через модуль предварительного выполнения в цикле выборки команд он дополнительно уменьшает частоту доступа к программной памяти FLASH, после множества тестов LGT8XM

Он может уменьшить доступ к FLASH примерно на 50% по сравнению с другими ядрами той же архитектуры, что значительно снижает потребление энергии всей системой.

Ядро LGT8XM имеет файл из 32 универсальных 8-битным регистра с высокоскоростным доступом, которые облегчают одно цикловые арифметические логические операции (ALU).

В общем случае два операнда операции ALU поступают из общего рабочего регистра, и результат операции ALU также записывается в файл регистра за один цикл.

Шесть, из тридцати двух рабочих регистров, используются в комбинации по два для формирования трех 16-разрядных регистров, которые могут использоваться в качестве указателей адресов косвенной адресации для доступа к внешнему пространству памяти и программному пространству FLASH.

LGT8XM поддерживает одноктактные 16-разрядные арифметические операции, что значительно повышает эффективность косвенной адресации.

Три специальных 16-разрядных регистра в ядре LGT8XM называются X, Y, Z-регистрами и будут подробно описаны ниже.

ALU поддерживает арифметические логические операции между регистрами и между константами и регистрами.

Работа отдельного регистра также может выполняться в ALU. По завершении работы ALU эффект результата операции в состоянии ядра обновляется в регистре состояния (SREG).

Управление потоком программ осуществляется условным и безусловным переходом/вызовом и может быть адресовано всем программным областям.

Большинство команд LGT8XM являются 16-разрядными.

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 по 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), а SP можно получить через пространство ввода-вывода.

Доступ к SRAM данных можно получить через 5 различных режимов адресации.

Внутреннее пространство памяти LGT8XM линейно отображено в едином адресном пространстве. Для получения дополнительной информации см. Введение раздела Памяти.

Ядро LGT8XM содержит гибкий контроллер прерываний, который можно контролировать с помощью бита разрешения глобального прерывания в регистре

состояния. Все прерывания имеют отдельный вектор прерывания. Приоритет прерывания имеет соответствующую связь с адресом вектора прерывания. 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 фактически контролируют основные регистры управления и состояния, SPI и другие периферийные устройства ввода-вывода. Эта часть пространства может быть напрямую доступна через инструкции IN/OUT, или к ней можно получить доступ через адрес, который они сопоставляют с пространством памяти (0x20 - 0x5F).

Кроме того, LGT8FX8D также содержит расширенное пространство ввода/вывода. Они сопоставляются с пространством памяти 0x60 - 0xFF, доступ к которому возможен только с помощью инструкций ST/STS/STD и LD/LDS/LDD.

2.2. Арифметический логический блок (ALU)

LGT8XM внутренне содержит 16-разрядный арифметический логический блок, который выполняет 16-разрядные арифметические операции данных за один цикл.

Эффективное ALU подключено к 32 универсальным рабочим регистрам.

Арифметическая логическая операция между двумя регистрами или регистрами и константами может быть выполнена за один цикл.

Существует три типа операций ALU: арифметические, логические и битовые операции. В то же время секция ALU также содержит одноктактный аппаратный множитель, который может реализовать два 8-битных регистра непосредственно со знаковыми или без знаковыми операциями за один цикл.

Пожалуйста, обратитесь к подробному описанию раздела набора инструкций.

2.2.1. SREG - Регистр состояния

Регистра состояния (SREG) в основном сохраняет информацию о результате, полученную в результате выполнения самой последней операции ALU.

Эта информация используется для управления потоком выполнения программы. Регистр состояния обновляется после завершения операции ALU, что устраняет необходимость в отдельной команде сравнения и приводит к более компактной и эффективной реализации кода.

Значение регистра состояния автоматически не сохраняется и не восстанавливается, когда оно реагирует на прерывание и выходит из прерывания. Это требует реализации программного обеспечения.

SREG - Регистр состояния системы

Регистр состояния системы SREG								
Адрес: 0x3F (0x5F)				По умолчанию: 0x00				
Бит	7	6	5	4	3	2	1	0
Имя	I	T	H	S	V	N	Z	C
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Определение бит								
[0]	C	Флаг переноса – переполнение при арифметической или логической операции.						

[1]	Z	Флаг Нуля - результат арифметической или логической операции равен нулю.
[2]	N	Флаг - отрицательной арифметической или логической операции.
[3]	V	Флаг переполнения - результатов операции бинарного переполнения.
[4]	S	Битовый знак эквивалентен результату операции XOR между N и V.
[5]	H	Флаг половинного переноса из 3 в 4 разряд байта, полезный в операциях BCD.
[6]	T	Временные биты, бит-копирование (BLD) и бит-хранилище (BST)
[7]	I	Флаг разрешения прерывания (I=1), чтобы позволить ядру реагировать на события прерывания. Различные источники прерываний управляются независимыми битами управления. Глобальный бит разрешения прерывания является последним барьером, который управляет сигналом прерывания в ядре. Бит I автоматически очищается аппаратным обеспечением после того, как ядро реагирует на вектор прерывания и устанавливается автоматически после выполнения команды возврата прерывания (RETI). Бит I также можно изменить с помощью инструкций SEI и CLI, см. Раздел описания инструкций

2.2.2 Общие рабочие регистры

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. Общие рабочие 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.

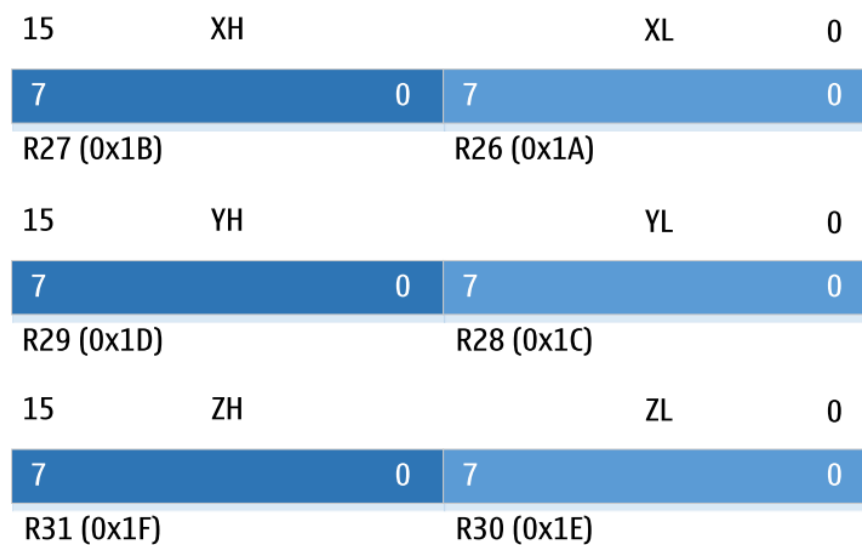


Рис.2.3. Регистры X, Y, Z.

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 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 связанные с указателем 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

2.2.5. Время выполнения инструкции

В этом разделе описываются общие временные концепции для выполнения команд. На этот раз ядро LGT8XM управляется тактами ядра (CLK_{cpu}) Такты поступают непосредственно из схемы выбора источника синхронизации системы.

На Рис.2.4. показана синхронизация выполнения конвейера команд на основе концепции архитектуры Navte и файла регистра быстрого доступа.

Это физическая гарантия, которая позволяет ядру достигать эффективности 1MIPS / МГц. Из рисунка видно, что вторая команда будет считываться одновременно во время выполнения первой команды. Когда вторая команда входит в период выполнения, третья команда будет считана одновременно. Это не требует дополнительных циклов для чтения инструкций на протяжении всего исполнения и на конвейерной основе эффективности выполнения одной инструкции каждый понедельник.

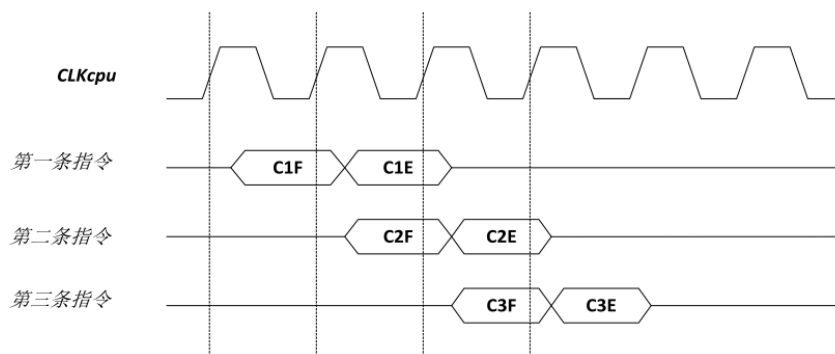


Рис.2.4. Диаграмма синхронизации выполнения конвейера команд

На Рис.2.5. показано время доступа для рабочего регистра общего назначения. В течение одного цикла операция ALU использует два регистра в качестве операндов и записывает результат выполнения ALU в регистр назначения в течение этого цикла.

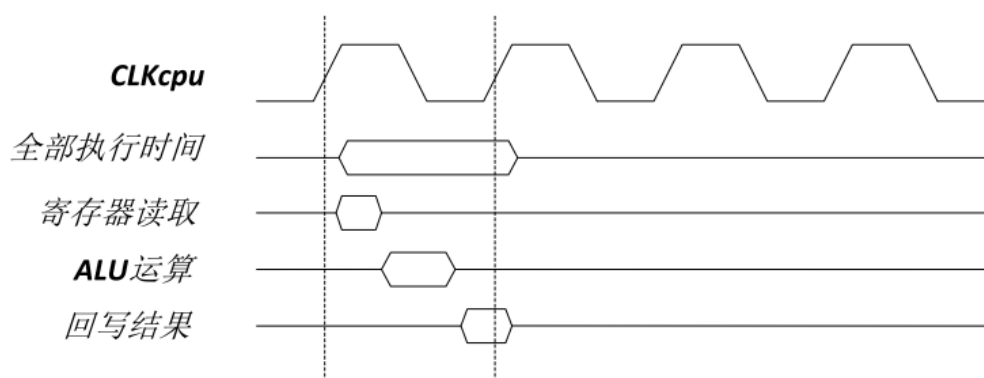


Рис.2.5. Диаграмма доступа к рабочему регистру общего назначения

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 C

```
char cSREG;  
cSREG = SREG;           /* хранить значение SREG */  
/* отключить прерывания во время временной последовательности */
```

```

_cli ();
EECR |= (1 << EEMPE);      /* начать запись EEPROM */
EECR |= (1 << EEPF);
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

```

Примеры кода C

```

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

```

2.2.7.Время отклика прерывания

Ядро LGT8XM оптимизировано для ответа прерывания, поэтому любое прерывание должно обслуживаться 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. Система Памяти

3.1.Обзор

В этой главе в основном описываются различные модули памяти внутри серии LGT8FX8D.

Архитектура LGT8XM поддерживает два основных пространства внутренней памяти: пространство памяти данных и пространство памяти программ.

Кроме того, LGT8FX8D также содержит данные FLASH внутри, а внутренний контроллер может реализовать функцию хранения данных интерфейса EEPROM.

Кроме того, система LGT8FX8D также содержит специальный блок памяти для хранения информации о конфигурации системы и базового номера устройства чипа (GUID).

Серия LGT8FX8D включает в себя четыре различные модели LGT8F48D/88D/168D/328D, четыре типа периферийных устройств и пакетов полностью совместимы.

Разница заключается в пространстве памяти программ FLASH и внутренних SRAM данных. В Табл.3.1 описываются различные конфигурации чипов LGT8FX8D.

Таблица 3.1. Разная конфигурация хранилища

Микросхема	FLASH	SRAM	E2PROM	Вектор прерывания
LGT8F48D	4KB	512B	1KB	1 слово инструкции
LGT8F88D	8KB	1KB	2KB	1 слово инструкции
LGT8F168D	16KB	1KB	4KB	2 слово инструкции
LGT8F328D	32KB	2KB	Настраивается как 0K/1K/2K/4K/8K (совместно с FLASH)	2 слово инструкции

LGT8F328D не имеет независимого пространства FLASH для эмуляции интерфейса E2PROM, оно используется для эмуляции пространства памяти E2PROM и совместного использования программы FLASH. Пользователь может выбрать соответствующую конфигурацию в соответствии с требованиями приложения.

Из-за уникальной реализации интерфейса E2PROM система требует в два раза больше пространства флэш-памяти программы для эмуляции пространства памяти E2PROM. Например, если пользователь настроил пространство E2PROM на 1 КБ для LGT8F328D, зарезервировано 2 Кбайт байтов программного пространства, оставив 30 КБ. Пространство FLASH используется для хранения программ.

Таблица 3.2. Программая FLASH и конфигурации E2PROM LGT8F328D:

DEVICE	FLASH	E2PROM 1
LGT8F328D	32KB	0KB
	30KB	1KB
	28KB	2KB
	24KB	4KB
	16KB	8KB

3.2. Программируемый программный модуль памяти FLASH

Семейство микроконтроллеров LGT8FX8D включает в себя 4K/8K/16K/32K байтов встроенных программируемых программных модулей памяти FLASH.

Программа FLASH может гарантировать не менее 20 000 циклов записи и стирания.

LGT8FX8D имеет интегрированный контроллер интерфейса FLASH, который позволяет осуществлять внутрисистемное программирование (ISP) и

самообновление программ.

Подробные сведения о реализации см. В описании раздела контроллера FLASH Interface в этой главе.

Данные программного пространства также можно получить (прочитать) напрямую через инструкцию LPM. Эта функция позволяет зависящие от приложения постоянные таблицы поиска.

В то же время программное пространство FLASH также отображается в пространстве хранения системных данных. Пользователи также могут использовать LD/LDD/LDS для доступа к пространству FLASH. Программное пространство отображается в диапазон адресов, начиная с области хранения данных 0x4000. Как показано на Рис.3.1.

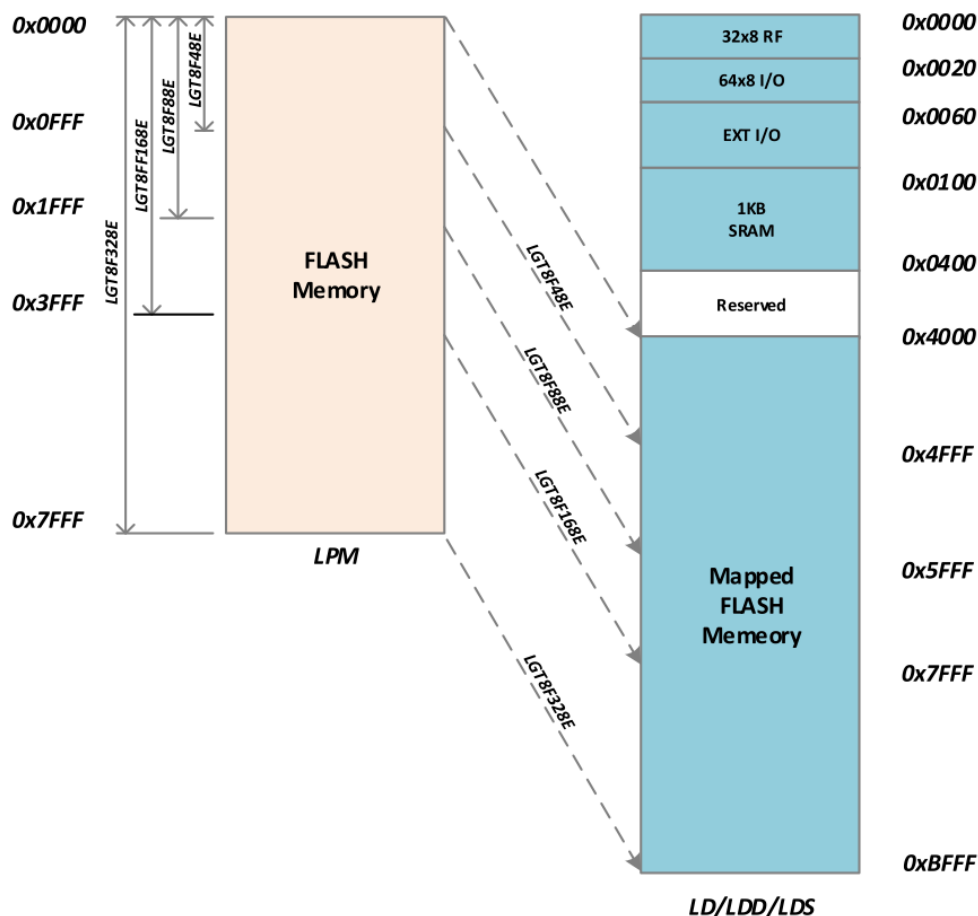


Рис.3.1. Программное пространство

3.3. SRAM Блок хранения данных

Семейство микроконтроллеров LGT8FX8D является относительно сложным микроконтроллером, который поддерживает множество различных типов периферийных устройств, контроллеры которых распределены в 64 регистрах пространствах ввода/вывода. Доступ к нему возможен напрямую с помощью инструкций IN/OUT.

Другие регистры периферийного управления выделяются в области 0x60 ~ 0xFF. Поскольку эта часть пространства отображается в пространство памяти данных, к ней можно получить доступ только через инструкции ST/STS/STD и LD/LDS/LDD. Объем памяти системных данных LGT8FX8D начинается с адреса 0 и сопоставляется с общим файлом рабочего регистра, пространством ввода-вывода,

расширенным пространством ввода-вывода и пространством SRAM для внутренних данных.

Первые 32-байтные адреса соответствуют 32 рабочим регистрам общего назначения ядра LGT8XM.

Следующие 64 адреса являются стандартными пространствами ввода/вывода, к которым можно напрямую обращаться через инструкции IN/OUT.

Следующие 160 адресов - это расширенное пространство ввода-вывода, за которым следуют 1024 байта SRAM данных.

Часть пространства, начинающаяся с 0x4000 и заканчивающаяся на 0xBFFF, отображает местоположение памяти программы FLASH.

32 通用工作寄存器	0x0000 - 0x001F
64 标准I/O空间	0x0020 - 0x005F
160 扩展I/O空间	0x0060 - 0x00FF
1024 SRAM数据空间	0x0100 - 0x03FF
Reserved	0x0400 - 0x3FFF
4K~32KB FLASH Memory	0x4000 - 0xBFFF

Рис. 3.2. Единое пространство сопоставления системных данных

Система поддерживает пять различных режимов адресации для охвата всего пространства данных: прямой доступ, косвенный доступ со смещением, косвенный доступ, косвенный доступ для уменьшенных адресов перед доступом и косвенный доступ для дополнительных адресов после доступа.

Общие рабочие регистры R26 - R31 адресные указатели для непрямого доступа.

Косвенный доступ может адресовать все пространство для хранения данных.

Косвенный доступ со смещенными адресами может адресовать до 63 адресных пространств с регистром Y / Z в качестве базового адреса. При использовании режима косвенного доступа в регистре, который поддерживает автоматическое увеличение/уменьшение адреса, регистр адресов X/Y/Z автоматически уменьшается/увеличивается на аппаратное обеспечение до/после доступа. Подробнее см. В разделе описания набора инструкций.

3.4. Пространство регистров ввода-вывода

Подробное определение пространства ввода-вывода см. В разделе «Описание регистров» в листе данных LGT8FX8D.

Все периферийные устройства LGT8FX8D распределены по пространству ввода-вывода. Доступ ко всем адресам пространства ввода/вывода можно получить инструкциями LD/LDS/LDD и ST/STS/STD.

Доступ к данным осуществляется через 32 общих рабочих регистра. Регистры ввода-вывода от 0x00 до 0x1F могут быть доступны через инструкции бит-адресации SBI и CBI. В этих регистрах значение бит может быть определено с помощью инструкций SBIS и SBIC для управления потоком выполнения программы.

Подробнее см. В разделе описания набора инструкций.

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

Когда для доступа к пространству ввода-вывода используются LD или ST-инструкции, к ним следует обращаться через пространство ввода-вывода на отображаемом адресе унифицированного пространства карты памяти системных данных (плюс смещение 0x20).

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

Чтобы быть совместимыми с будущими устройствами, в зарезервированные биты должны быть установлены «0» при записи

3.5. GPIOR2/1/0 Регистры ввода-вывода общего назначения

В пространстве ввода-вывода LGT8FX8D есть три универсальных регистра ввода-вывода, GPIOR2/1/0, к которым можно получить доступ, используя инструкцию IN/OUT для хранения пользовательских данных.

3.6. Интерфейсный контроллер EFLASH / E2PROM

LGT8FX8D внутренне реализует гибкий и надежный контроллер интерфейса E2PROM, который может использовать существующие данные в пространстве памяти FLASH в системе, чтобы реализовать пространство для хранения байтов для чтения / записи и реализовать приложения хранения, похожие на E2PROM, интерфейс E2PROM

Используя алгоритм выравнивания записи, срок службы данных FLASH может быть увеличен в 1 раз, и более 50 000 циклов записи / записи могут быть гарантированы. Контроллер интерфейса FLASH реализует онлайн-операции стирания и записи в программном пространстве FLASH и может использоваться для автоматического обновления прошивки в Интернете. Доступ к программному пространству FLASH через контроллер FLASH поддерживает только 16-разрядные запись и чтение. Подробнее о доступе к E2PROM и программном флэш-памяти см. Подробное описание ниже.

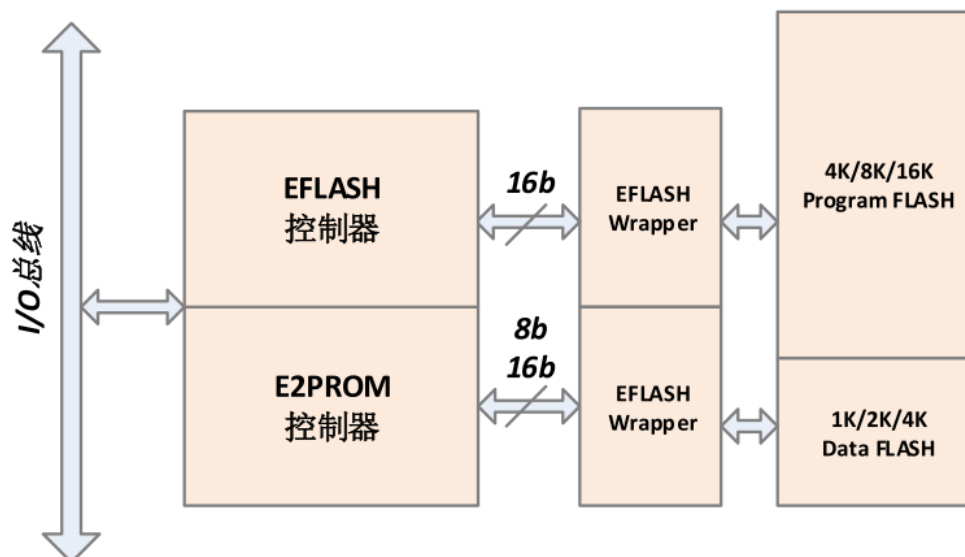


Рис.3.3 Структура контроллера LGT8F48D/88D/168D FLASH/E2PROM

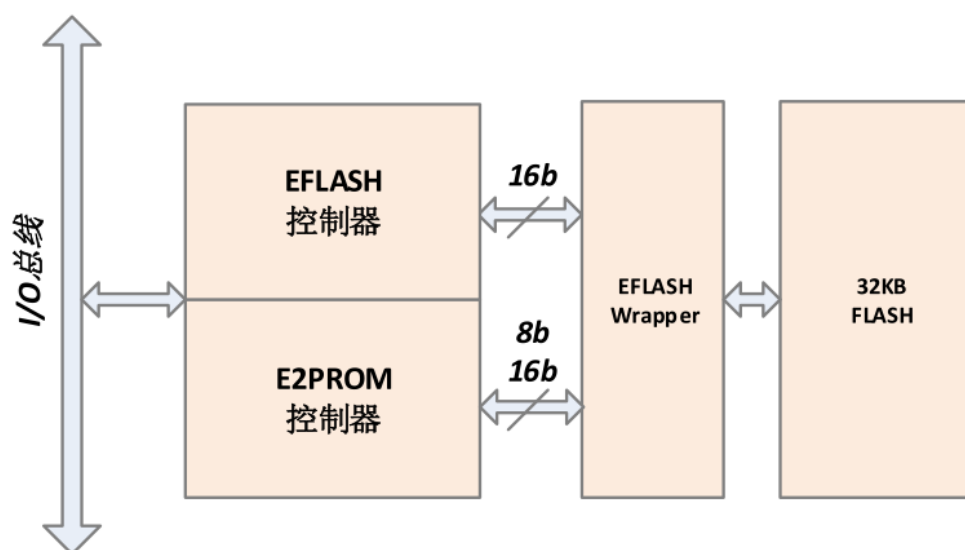


Рис.3.3 Структура контроллера LGT8F328D FLASH/E2PROM

3.6.1. Интерфейс чтения и записи FLASH/E2PROM

Все регистры контроллера FLASH могут быть доступны через пространство ввода/вывода. Работа FLASH и E2PROM-пространства также осуществляется путем настройки и управления этими регистрами. Подробный метод использования будет отдельно описан в разделе описания регистра.

Время доступа для записи FLASH / E2PROM см. Таблицу, приведенную ниже.

Контроллер FLASH может автоматически обновлять состояние текущей операции. Программное обеспечение пользователя может определить, завершена ли текущая операция, обнаружив эти состояния и запустив работу следующего байта. Если код пользователя содержит операции над FLASH / E2PROM, необходимо следовать некоторым рекомендациям.

Во-первых, при подаче питания или отключении питания V_{cc} медленно поднимается и падает, в результате чего устройство работает на низком напряжении в течение определенного периода времени. Это будет влиять на минимальное напряжение, необходимое для максимальной частоты работы текущей системы, а также повлияет на FLASH. Операция программирования будет

имеет дефект. На этом этапе вам необходимо принять необходимые защитные меры. Это будет подробно описано в следующем разделе.

Во избежание неправильной работы E2PROM, необходимо для работы E2PROM соблюдать специальную процедуру.

Пожалуйста, обратитесь к описанию регистров управления EFLASH / E2PROM в конце этого раздела. При работе с EFLASH / E2PROM выполнение ядра LGT8XM будет выполняться до тех пор, пока ядро не сможет возобновить работу после завершения операции.

3.6.2 Защита работы FLASH / E2PROM

Если напряжение Vcc низкое, работа с данными FLASH / E2PROM может быть ошибочной, поскольку напряжение слишком низкое. Это тот же дизайн, что и чип E2PROM на уровне платы.

Ошибки в работе данных FLASH / E2PROM при низком напряжении могут быть вызваны двумя причинами.

Во-первых, для нормальной работы FLASH / E2PROM требуется минимальное рабочее напряжение. Ниже этого напряжения операция не будет приводить к ошибкам данных.

Вторая причина заключается в том, что ядро работает на определенной частоте и также требует минимального напряжения. Когда напряжение ниже этого напряжения, а ЦП продолжает работать на этой частоте, это вызовет ошибку выполнения команды, которая сделает FLASH / E2PROM.

В операции произошла ошибка. Аналогичную проблему можно избежать с помощью следующего простого метода: 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, подумайте о добавлении схемы сброса извне.

Невозможно выполнить операции записи в зарезервированном пространстве ввода-вывода.

Некоторые регистры включают флаг состояния, в который необходимо записать «1», чтобы очистить его. Обратите внимание, что инструкции CBI и SBI поддерживают только определенные биты, поэтому CBI / SBI может работать только с регистрами, которые содержат эти флаги состояния. Кроме того, инструкции CBI / SBI могут работать только в регистрах в диапазоне адресов от 0x00 до 0x1F.

3.6.3. FLASH / E2PROM Контроль времени чтения / записи

Регистр EECR управляет реализацией всех операций FLASH.

EEPROM в основном управляет режимом работы и типом операции выбора.

Основным выбором EEPROM [3] является использование данных FLASH (E2PROM) или программы FLASH. Когда объектом операции является программа FLASH, интерфейс данных фиксируется в 32-битном режиме. Когда объектом операции являются данные FLASH (E2PROM), можно выбрать различные ширины

данных.

По умолчанию используется 8-битный режим, который является самым простым и интуитивно понятным.

Когда контроллер FLASH реализует интерфейс E2PROM, логика автоматического стирания данных FLASH реализуется внутренне, когда это необходимо. Поэтому команда стирания EPROM является необязательной. Эта команда используется только тогда, когда пользователю необходимо выполнить операцию стирания в одиночку. EEMPE контролирует время стирания / записи FLASH, включая программу FLASH и E2PROM. Оба должны выполнить ответ под контролем EEMPE.

EEPE может запускать все операции стирания и программы во время действия EEMPE. Тип конкретной операции определяется EEPM [3: 0].

Чтение операции E2PROM относительно просто. После установки целевого адреса и режима запись EERE-бита считывает 32-битные данные, соответствующие целевому адресу, в контроллер FLASH. Пользователь может прочитать интересующий байты через регистр EEDR.

Контроллер FLASH не выполняет операции чтения в программном пространстве FLASH. Пользователь может легко использовать LPM или читать инструкцию LD / LDD / LDS по адресу унифицированного пространства отображения данных через программу FLASH.

Пример потока данных **FLASH / E2PROM**

1. 8-битный режим, программирование E2PROM

- Обнаружение бит EEPE, ожидание работы контроллера FLASH
- Установка целевого адреса в EEAR [8: 0]
- Установка новых данных в EEDR
- Установка EEPM [3: 1] = 000, EEPM [0] может быть установлен на 0 или 1
- установлен EEMPE = 1, а EEPE = 0
- EEPE = 1 устанавливается в течение четырех циклов.

Когда настройка будет завершена, контроллер FLASH начнет программирование во время программирования.

ЦП останется на текущем адресе инструкции и не продолжит работу до завершения операции. Во время процесса программирования контроллер FLASH автоматически инициирует процесс стирания, если ему нужно стереть данные FLASH.

2. 16-разрядный режим, программирование E2PROM

- Обнаружение бит EEPE и ожидание остановки контроллера FLASH
- Установите 16-битные данные через EEAR [0] и EEDR, см. Раздел определения регистров EEDR
- Установите целевой адрес в EEAR [12: 0]. Обратите внимание, что это адрес, выровненный по байтам. Контроллер FLASH использует EEAR [14: 1] в качестве адреса для доступа к FLASH.
- Установите EEPM [3: 1] = 001, EEPM [0] можно установить на 0 или 1

- Установить EEMPE = 1 и EEPPE = 0
- Установить EEPPE = 1 в четыре цикла

3. 8-битный режим, чтение E2PROM

- Обнаружение бит EEPPE, ожидание контроллера FLASH в режиме ожидания
- Установка целевого адреса в EEAR [8: 0]
- Установка EEPM [3: 1] = 000
- Установка EEPPE = 1 запуск E2PROM операция чтения
- Ожидание 2 Цикл (выполнить две операции NOP)
- Данные, соответствующие целевому адресу, обновляются в регистре EEDR

4. 16-разрядный режим, чтение E2PROM

- Обнаружение бит EEPPE, ожидание, когда контроллер FLASH неактивен;
- Установить EEAR [12: 0] в качестве целевого адреса, адрес равен 2 байтам;
- Установить EEPM [3: 1] = 001, включить 16 бит Режим интерфейса
- Настройка EEPPE = 1, запуск операции чтения E2PROM
- Ожидание двух системных тактовых циклов (выполнение двух инструкций NOP)
- Данные, соответствующие целевому адресу, обновляются во внутренний 16-разрядный регистр контроллера, и пользователь может использовать EEAR [0] и EEDR считывает данные указанного байта или всех данных.

5. Программа FLASH Операция стирания

- Определите бит EEPPE и дождитесь, пока контроллер FLASH не будет работать.
- Установите EEAR [14: 0] в качестве адреса целевой страницы для стирания. Размер страницы FLASH-программы составляет 1 Кбайт, поэтому EEAR [14:10] В качестве адреса страницы EEAR [9: 0] установлено значение 0.
- Установлен EEPM [3: 0] = 1X01, где EEPM [2] может быть установлен на 0 или 1
- EEMPE = 1, а EEPPE = 0
- В четырех циклах Внутри установите EEPPE = 1, запустите программу FLASH стирание.

6. Программирование программирования FLASH

- Обнаружение бит EEPPE и ожидание работы контроллера FLASH
- Установите 16-битные данные программирования через EEAR [0] и EEDR
- Установите EEAR [14: 1] в качестве целевого адреса, адрес которого выровнен на 2 байта
- Настройка EEPM [3: 0] = 1X10, где EEPM [2] может быть установлен на 0 или 1
- Установите EEMPE = 1 и EEPPE = 0
- В четырех циклах установите EEPPE = 1, чтобы запустить поток программирования FLASH

[Важно] Для LGT8F328D необходимо включить контроллер E2PROM и

настроить размер E2PROM через регистр ECCR перед выполнением операции E2PROM. Последующая операция такая же, как и у LGT8F48D/88D/168D.

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			
		Моде ь	FLASH	E2PR0M	Размер EEAR
		LGT8F48D	4KB	1KB	EEAR[9:0]
[14:8]	EEARH	LGT8F88D	8KB	2KB	EEAR[10:0]
		LGT8F168D	16KB	4KB	EEAR[11:0]
		LGT8F328D	32/30/28/24/16K	0/1/2/4/8K	EEAR[12:0]
[15]	-	Зарезервировано			

При использовании контроллера EFLASH/E2PROM для доступа к области FLASH программы - EEAR [14: 1] используется для доступа ко всему программному пространству, выровненному по 2 байта. EEAR [0] используется только при обращении к регистру данных EEDR.

Подробнее см. Описание регистра данных EEDR. При использовании контроллера EFLASH / E2PROM для доступа к области FLASH данных (E2PROM) EEAR [12: 0] используется для доступа к максимальным 8 Кбайтам пространства E2PROM. Доступ в это время поддерживает 8/16-битный режим, а EEAR выровнен по байтам независимо от режима.

EEDR - 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	Регистр данных EFLASH / E2PROM

Важное примечание: Внутренняя уставка LGT8FX8D - это 16-разрядный интерфейс, а минимальная единица данных чтения / записи - 16 бит. Поэтому регистр данных внутри контроллера FLASH составляет 16 бит. EEAR [0] используется для решения старших и младших 8 бит.

EEDR - это 8-разрядный регистр данных, значение которого зависит от режима доступа. Когда контроллер FLASH используется для доступа к внутренней программе FLASH, контроллер FLASH работает в 16-разрядном режиме. В это время EEDR служит в качестве интерфейса для доступа к внутренним 16-

разрядным регистрам данных и будет работать с EEAR [0].

При использовании контроллера FLASH для доступа к данным FLASH интерфейс доступа контроллера FLASH может работать в 8/16-битном режиме.

При работе в 8-битном режиме EEDR - это фактические данные, которые необходимо прочитать / записать, а EEAR [12: 0] используется для адресации пространства E2PROM с максимальным объемом 8 Кбайт.

Аппаратное обеспечение автоматически завершит преобразование 8-битных данных в 16-разрядный интерфейс доступа к данным без каких-либо дополнительных операций пользователя.

При работе с 16-битным режимом EEDR, также будет служить интерфейсом для доступа к внутренним данным и будет работать с EEAR [0]. В этих двух режимах пользователю необходимо настроить данные, которые должны быть записаны в FLASH через EEAR [0] и EEDR, или прочитать требуемые байтовые данные через EEAR [0] и EEDR.

На следующем Рис. 3.4. показана взаимосвязь между регистром ввода / вывода EEAR / EEDR и внутренним интерфейсом контроллера FLASH

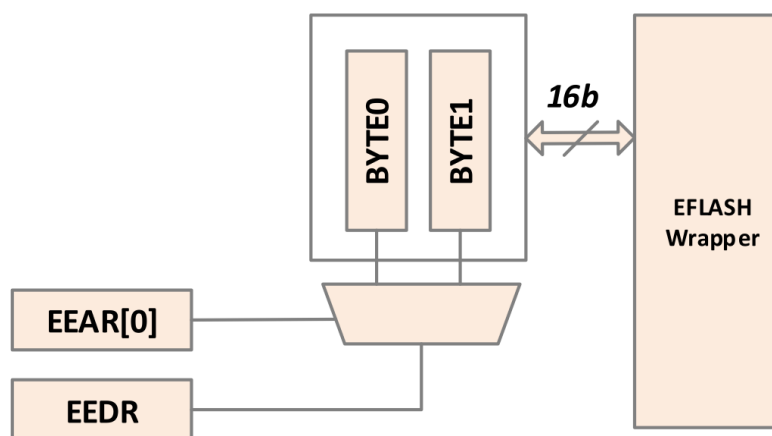


Рис. 3.4. Взаимосвязь между регистром ввода/вывода EEAR/EEDR и внутренним интерфейсом контроллера FLASH

Когда используется 8-разрядный режим, EEAR [12: 0] обновляет данные указанной позиции байта вместе с EEDR. Остальные данные автоматически объединяются логикой внутреннего управления контроллера FLASH.

Пользователям не нужно заботиться о конкретной реализации. При использовании 16-разрядного режима пользователю необходимо обновить 16-разрядные данные, которые составляют 2 байта данных. Аппаратное обеспечение решает, следует ли обновлять верхние 8 бит или нижние 8 бит согласно EEAR [0]. Метод обновления данных выглядит следующим образом:

```
OUT EEARL, $0
OUT EEDR, BYTE0
OUT EEARL, $1
OUT EEDR, BYTE1
# Задайте целевой адрес программирования
OUT EEARL, ADDR1
OUT EEARH, ADDR1H
... ..
```

ЕССР - Регистр управления режимом FLASH

Регистр конфигурации ECCR - FLASH / E2PROM (Только для LGT8F328D)								
ECCR: 0x36 (0x56)			0x00					
Биты	7	6	5	4	3	2	1	0
Имя	WEN	EEN	-	-	-	-	EC1	EC0
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						
[5:2]	-	Зарезервировано						
[1:0]	ES[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			
Биты	7	6	5	4	3	2	1	0
Имя	EPM3	EPM2	EPM1	EPM0	EERIE	EEMPE	EEPE	EERE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Определение бит								
[7:4]	EPM[3:0]	Бит управления режимом доступа EFLASH / EPROM						
		[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]	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
Определение бит		
[7:0]	GPIOR2	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
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
Определение бит		
[7:0]	GPIOR0	Регистр ввода-вывода общего назначения 0 для хранения пользовательских данных

4. Системные Такты и конфигурация

4.1. Распределение часов системы и синхронизации системы.

LGT8FX8D поддерживает несколько тактовых входов. Система может работать на трех основных источниках синхронизации:

- внутреннем 32KHz калибровочном RC-генераторе,
- внутреннем 32-мегагерцовом калибровочном RC-генераторе,
- внешнем входном кристалле мощностью 400KHz ~ 20MHz.

На Рис4.1. показано распределение тактовой системы LGT8FX8D.

CMU является центром всего управления Тактами. Он отвечает за деление системных тактов, создание независимых тактов для разных модулей и контроль тактов. В обычных приложениях все такты не должны работать одновременно. Чтобы снизить энергопотребление системы, управление питанием системы отключает неиспользуемые такты модуля в соответствии с различными режимами ожидания. Подробные сведения о работе см. В соответствующих разделах управления питанием.

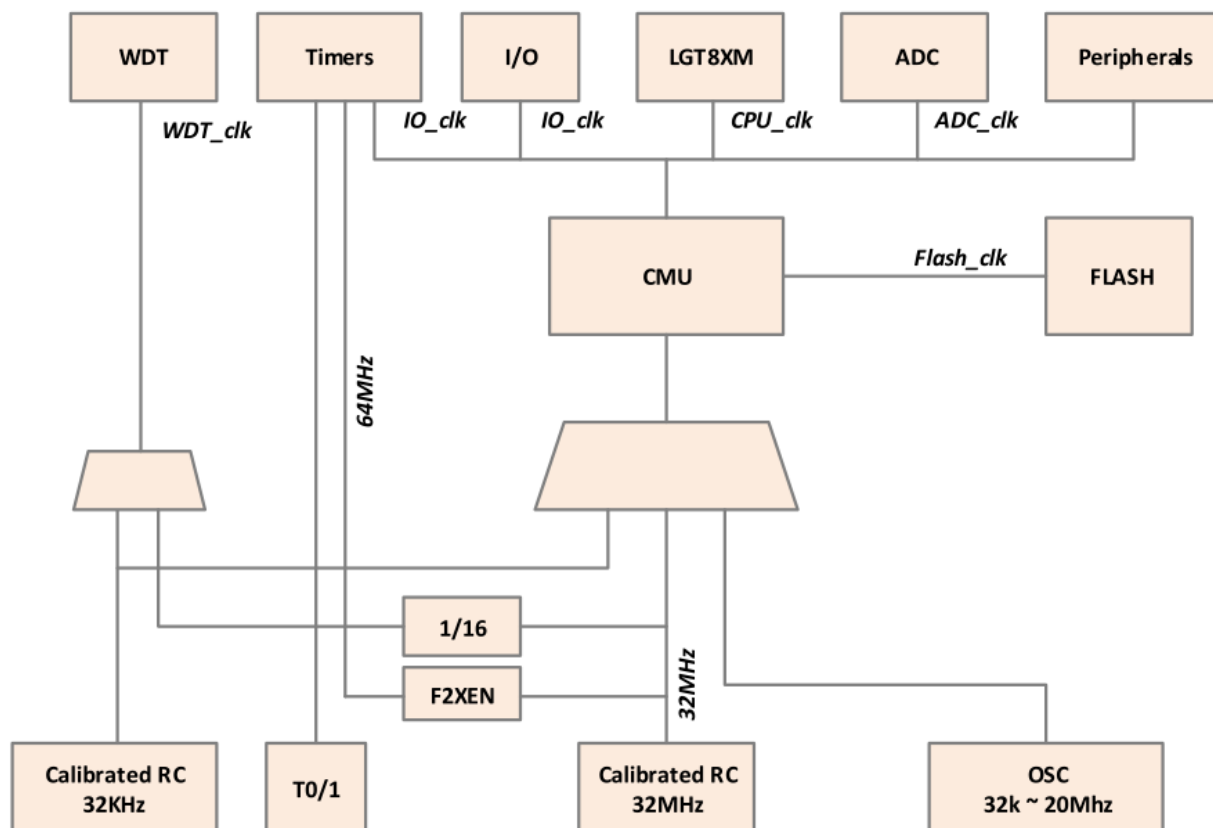


Рис4.1. Распределение тактовой системы LGT8FX8D.

CPU_clk

CPU_clk используется для управления ядром LGT8XM и SRAM. Такие, как управление общими рабочими регистрами, регистрами состояния и т. Д. После того, как Такты процессора остановлены, ядро не будет выполнять инструкции и вычисления.

IO_clk

Такты используются для управления большинством периферийных модулей, таких как таймер / счетчик, SPI, USART и т. Д. Такты ввода-вывода также используются для управления внешним модулем прерывания.

Когда Такты IO останавливаются из-за сна, некоторые периферийные части, которые могут использовать систему пробуждения, работают в отдельном такте или асинхронном режиме.

Например, функция распознавания адреса TWI может разбудить большинство режимов ожидания. В это время часть распознавания адреса работает в асинхронном режиме.

Flash_clk

FLASH Такты используется для генерации времени доступа к интерфейсу FLASH. Такты FLASH такие же, как и системные такты. Такты FLASH в основном используются для доступа к программе FLASH и FLASH через контроллер FLASH.

Asy_clk

Асинхронный таймер. Таймер / счетчик может управляться непосредственно от внешних Тактом или кристаллов (32,768 K). Этот независимый режим синхронизации может поддерживать работу таймера во время обработки режима

сна.

Выбор источника Тактов

LGT8FX8D поддерживает четыре типа входов источника синхронизации.

Пользователи могут использовать регистр PMCR для включения управления источником синхронизации и полного переключения главного тактового генератора (См. Таблица 4.1.). На Рис.4.2. структурная схема управления PMCR.

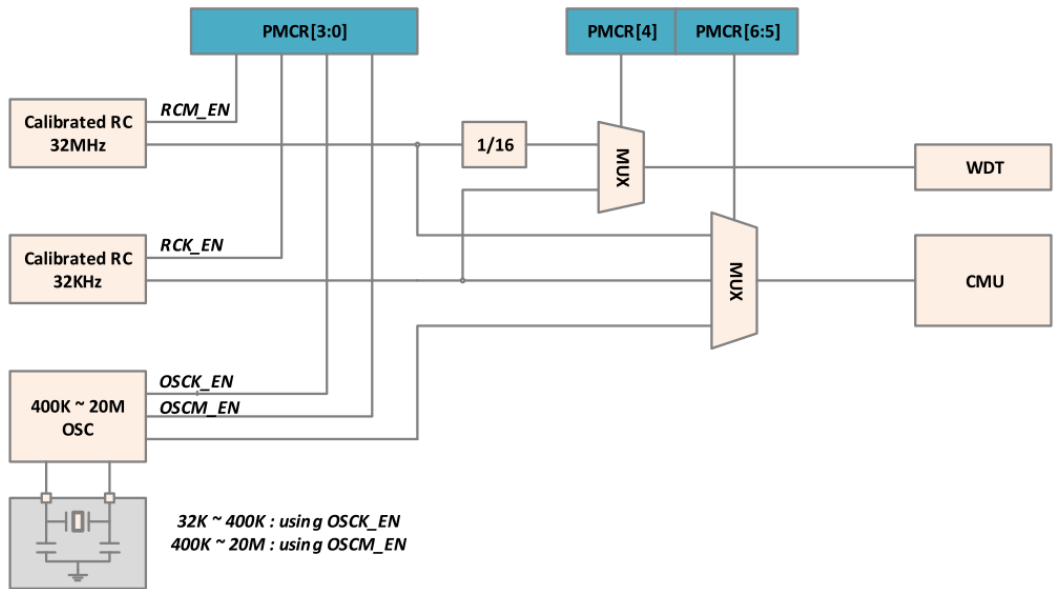


Рис.4.2. Структурная схема управления PMCR.

Внутренний OSC-генератор LGT8FX8D может работать как в высокочастотном, так и в низкочастотном режимах.

Пользователь должен управлять внутренним OSC-генератором для работы в режиме, в соответствии с фактическим размером внешнего кварцевого генератора. Один и тот же внутренний RC-генератор также разделен на высокую частоту и низкую частоту. Младшие 4 бита регистра PMCR используются для управления этими четырьмя источниками синхронизации.

Таблица 4.1. Включения источников синхронизации

PMCR	Соответствующий источник синхронизации
PMCR[0]	32MHZ включают управление RC: 1 включают, 0 Выключений
PMCR[1]	32KHz включают управление RC: 1 включают, 0 Выключений
PMCR[2]	400K ~ 20MHz включить режим OSC: 1 Включить, 0 Выкл.
PMCR[3]	32K ~ 400K включить режим OSC: 1 Включить, 0 Выкл.

После включения системы LGT8FX8D в качестве источника системных тактов по умолчанию используется 32 МГц RC, а ядро работает на частоте 8 МГц (2 МГц).

Пользователь может изменить конфигурацию по умолчанию, установив регистр PMCR и регистр предварительного делителя системы (CLKPR). Если вам необходимо изменить конфигурацию первичного источника синхронизации, вы должны убедиться, что источник синхронизируемого синхросигнала находится в стабильном рабочем состоянии перед переключением часов. Поэтому перед

переключением основного источника синхронизации необходимо включить желаемый источник синхронизации через PMCR [3: 0] и подождать, пока такты не стабилизируются до переключения.

Когда пользователь переключает основные такты на внешний кристалл, хотя пользователь включил внешний кристалл, он не исключает, что кристалл не может быть запущен из-за неправильной конфигурации или отказа кристалла.

Если вы переключитесь на внешний кристалл в это время, система сразу же прекратит работу после переключения. Поэтому, учитывая надежность системы, рекомендуется избегать таких проблем с точки зрения разработки программного обеспечения, включив сторожевой таймер.

Выбор основного источника синхронизации в Табл. 4.2. и на Рис.4.3.

После того, как источник синхронизации включен и ожидает стабилизации, главный тактовый сигнал может быть переключен PMCR [6: 5]. PMCR [5] используется для выбора внутреннего RC-генератора и внешнего кристалла.

PMCR [6] используется для выбора высокоскоростного источника синхронизации и низкоскоростного источника синхронизации.

Таблица 4.2. Выбор основного источника синхронизации:

PMCR[6]	PMCR[5]	Источник первичных Тактов
0	0	Внутренний 32-мегагерцовый RC генератор (по умолчанию)
0	1	Внешний высокочастотный генератор с частотой 400К ~ 20 МГц
1	0	Внутренний 32KHz RC-генератор
1	1	Внешний 32К ~ 400KHz низкочастотный кристалл

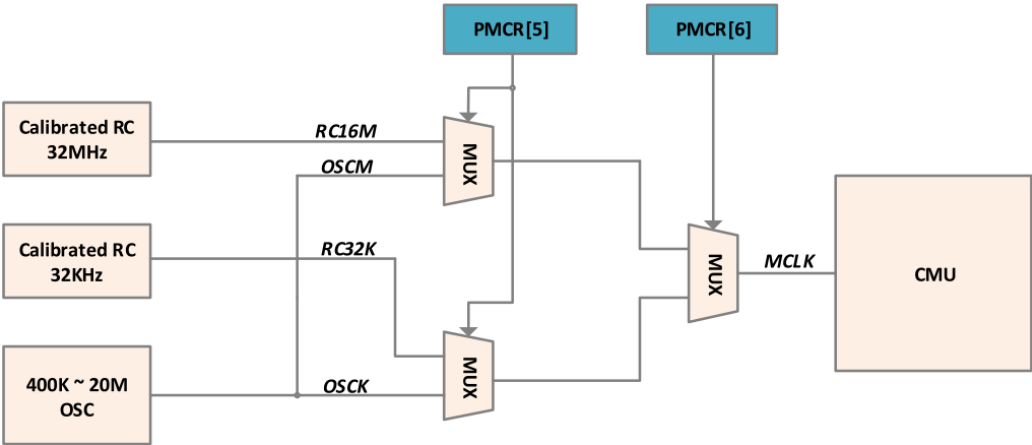


Рис.4.3. Структурная схема выбора основного источника синхронизации

4.2. Синхронизация источника синхронизации тактов.

Чтобы защитить регистр PMCR от случайного изменения, изменения в регистре PMCR должны быть строго установлены в указанное время.

Самый старший бит регистра PMCR (PMCR [7]) используется для реализации управления синхронизацией.

Прежде чем пользователи смогут модифицировать другие биты PMCR, они должны сначала установить PMCR [7], чтобы изменить значение других регистров PMCR в течение 6 циклов после установки операции. После 6 циклов прямые изменения PMCR будут признаны недействительными. Ниже приведен пример

переключения на внешний высокоскоростной кварцевый генератор и перечисление рекомендуемых этапов работы:

1. Включить источник синхронизации
 - Установить PMCR [7] = 1
 - Установить PMCR [3] = 1 в шесть циклов, чтобы включить внешние Высокоскоростной внешний кварцевый генератор
 - Подождите, пока внешний кварцевый генератор стабилизируется (время ожидания меняется в зависимости от кварцевого генератора. Обычно уровень может ждать)
2. Переключатель основного источника синхронизации
 - Установите PMCR [7] = 1
 - В шести циклах установите PMCR [6: 5] = 01. Система автоматически переключит рабочие такты на внешний кварцевый генератор.
 - Выполните несколько операций NOP для повышения стабильности. Выберите операцию)

[Осторожно] В приведенной выше операции переключения основных Тактов, чтобы была гарантия, что текущие системные такты работают нормально, после переключения на внешний кварцевый генератор, предыдущий внутренний RC-генератор можно отключить.

4.3. Управление предварительным делением системы

LGT8FX8D имеет предусилитель системных тактов, который можно контролировать через регистр предварительного делителя часов (CLKPR).

Эта функция может использоваться для снижения энергопотребления системы, когда система не требует очень высокой вычислительной мощности.

Настройки предварительного делителя действительны для источников синхронизации, поддерживаемых системой. Предделитель Тактов может влиять на такты выполнения ядра и, таким образом, синхронизировать периферийные устройства.

При переключении между различными настройками предварительного делителя тактов предварительный каскад системных часов гарантирует, что во время переключения не возникнут глюки, и что никогда не будет промежуточного состояния переключения.

После того, как изменения регистра вступят в силу, системные такты переключаются на новые разделенные такты после (максимум) 2 - 3 текущих тактовых циклов системы.

Чтобы избежать ошибочной работы регистра делителя тактовых импульсов, модификация CLKPR также должна следовать специальной временной последовательности:

- Установите бит разрешения изменения предварительного делителя часов (CLKPCE) на 1 и другие биты CLKPR на 0
- В четвертом цикле требуемое значение записывается в CLKPS, а CLKPCE записывается в 0.

Перед изменением регистра предварительного делителя часов отключите функцию прерывания, чтобы гарантировать, что время записи может быть

завершено. Для конкретного определения основного генератора тактовых импульсов CLKPR см. Раздел описания регистра этого раздела.

[Важно] Максимальная рабочая частота LGT8FX8D составляет 20 МГц. Поэтому, когда в качестве основного источника синхронизации выбран внутренний 32-мегагерцовый RC, необходимо убедиться, что CLKPR настроен на правильную конфигурацию частотного деления (**минимум 2-х частотное деление**).

4.4. Калибровка внутреннего RC-генератора

LGT8FX8D содержит два калибровочных RC-генератора, внутренне откалиброванных с точностью $\pm 1\%$. Среди них 32 МГц RC используется по умолчанию для часов работы системы.

Перед каждым изготовлением LGT8FX8D внутренний 32-мегагерцовый RC был откалиброван, и значение калибровки было записано в область информации о конфигурации системы.

Пользователи получают доступ к этому калибровочному значению через регистр RCCAL в пространстве регистров ввода-вывода.

Внутренний 32КHz RC не откалиброван при его изготовлении.

Если вам нужны очень точные низкочастотные такты, вы можете самостоятельно калибровать RC-генератор и записать калибровочное значение в собственные данные LGT8FX8D FLASH.

После каждого запуска микросхемы программное обеспечение считывает это калибровочное значение и записывает его в регистр RCKCAL для завершения калибровки 32-КГц генератора RC. Для конкретного метода калибровки см. п.п.4.4.1.

4.4.1. Внутренний 32КHz RC Метод калибровки

Перед калибровкой внутреннего 32КHz RC, вы должны иметь возможность измерять текущую внутреннюю тактовую частоту 32 кГц.

Более простой способ - переключить такты работы системы на внутренний 32КHz RC-генератор. Затем выведите меандр, который легче измерить частоту с помощью ввода-вывода. Частота внутреннего RC получается путем измерения частоты меандра извне.

Вышеуказанный метод калибровки для RC, и метод калибровки RC-генератора 32 кГц тот же. После получения RCKCAL значение может быть записано в зарезервированную область данных FLASH (E2PROM) для последующей калибровки программного обеспечения.

4.5. Регистры Описание

RCCAL - 32-мегагерцовый калибровочный регистр RC

RCCAL - 32-мегагерцовый калибровочный регистр RC	
RCCAL: 0x66	По умолчанию: 0x00
Биты	RCCAL[7:0]
R/W	R/W

Определение бит		
[7:6]	-	Зарезервировано
[5:0]	RCCAL	После включения системы значение регистра будет заменено калибровочным значением RC в информации о конфигурации системы. См. п.п.4.4.

PMCR - регистр управления синхронизацией Тактов

PMCR - регистр управления синхронизацией тактов								
PMCR: 0xF2		По умолчанию: 0x01						
Биты	7	6	5	4	3	2	1	0
Имя	PMCE	CLKFS	CLKSS	WCLKS	OSCKEN	OSCMEN	RCKEN	RCMEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Определение бит								
[7]	PMCE	Регистр PMCR изменяет бит управления включением. Перед изменением других положений PMCR этот бит должен быть установлен первым, тогда остальные значения бит должны быть установлены в течение четырех циклов.						
[6]	CLKFS	CLKFS	CLKSS	Источник первичных Тактов				
		0	0	Внутренний 32-мегагерцовый RC генератор (по умолчанию)				
		0	1	Внешний ВЧ генератор с частотой 400К ~ 20 МГц				
[5]	CLKSS	1	0	Внутренний 32KHz RC-генератор				
		1	1	Внешний 32К ~ 400KHz низкочастотный кристалл				
[4]	WCLKS	Выбор внутреннего RC источника тактов WDT, «0» - 1 МГц, «1» - 32 кГц						
[3]	OSCKEN	Внешний 32К ~ 400KHz низкочастотный кристалл						
[2]	OSCMEN	Внутренний 32KHz RC-генератор						
[1]	RCKEN	Внешний высокочастотный генератор с частотой 400К ~ 20 МГц						
[0]	RCMEN	Внутренний 32-мегагерцовый RC генератор (по умолчанию)						

CLKPR - Регистр основного предделителя тактов

CLKPR - Регистр основного предделителя тактов								
CLKPR: 0x61					По умолчанию: 0x03			
Биты	7	6	5	4	3	2	1	0
Имя	CLKPCE	CLKOEN1	CLKOEN0	-	CLKPS3	CLKPS2	CLKPS1	CLKPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Определение бит								
[7]	CLKPCE	Предделитель тактов. Изменение управления тактами. Перед заменой других бит регистра CLKPR вы должны сначала установить CLKPCE в «1», а затем установить остальные биты в течение следующих четырех системных циклов. По завершении четырех циклов CLKPCE автоматически очищается.						
[6]	CLKOEN1	Установить в «1» - будут системные такты выводиться на вывод PE5						
[5]	CLKOEN0	Установить в «1» - будут системные такты выводиться на вывод PB0						
[4]	-	Зарезервировано						
[0:3]	CLKPS	Биты выбора предварительного делителя тактов						
		CLKPS3	CLKPS2	CLKPS1	CLKPS0	Параметры деления		
		0	0	0	0	1		
		0	0	0	1	2		
		0	0	1	0	4		
		0	0	1	1	8 (Настройка по умолчанию)		

		0	1	0	0	16
		0	1	0	1	32
		0	1	1	0	64
		0	1	1	1	128
		1	0	0	0	256
		Зарезервировано				Другие значения

RCKCAL - 32-килогерцовый калибровочный регистр RC

RCKCAL - 32-килогерцовый калибровочный регистр RC		
RCKCAL: 0x67		По умолчанию: 0x00
Биты		RCKCAL [7:0]
R/W		R/W
Определение бит		
[7:6]	-	Зарезервировано
[5:0]	RCKCAL	После включения системы значение регистра будет заменено калибровочным значением RC. См. п.п.4.4.1.

5. Управление питанием

5.1. Обзор

Спящий режим используется для отключения неиспользуемых модулей в MCU, что снижает энергопотребление системы.

LGT8FX8D обеспечивает очень гибкий режим ожидания и контроллер модуля, позволяет пользователю достичь наилучшей конфигурации с малой мощностью потребления.

Когда модуль LVD включен, LVD продолжит работу в спящем режиме.

Для дальнейшего снижения энергопотребления системы его можно отключить перед тем, как перейти в спящий режим.

LGT8FX8D поддерживает режим пониженного энергопотребления.

В режиме пониженного энергопотребления цифровая часть системы, большинство входов/выходов и аналоговых модулей будут отключены.

Режим пониженного энергопотребления минимизирует энергопотребление системы.

Режим отключения питания может быть пробужден только указанным внешним выводом.

Процесс пробуждения согласуется с процессом включения системы.

Программное обеспечение может считывать регистр состояния MCU, чтобы получить предыдущий статус системы.

LGT8FX8D содержит внутренне точный 32-КГц RC-генератор, который позволяет пользователю переключать системные такты на 32KHz RC и использовать надлежащий режим сна, не имея дело со сложными задачами. Это также обеспечивает идеальное энергопотребление системы в нерабочем режиме.

Если пользователь подключает внешний кварцевый генератор 32768 Гц, вы также можете переключать основные такты на внешний кварцевый генератор, затем отключать неиспользуемый источник синхронизации и другие аналоговые

модули и перейти в спящий режим для дальнейшего сохранения энергии.

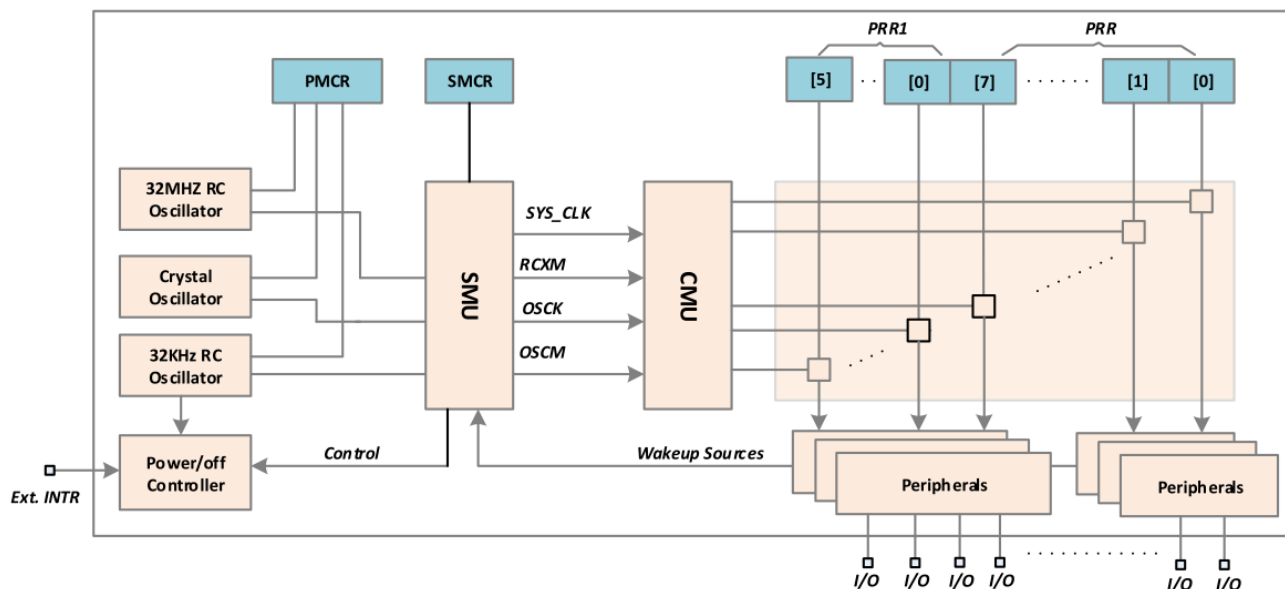


Рис.5.1. Схема управления питанием системы.

Как показано на Рис.5.1., LGT8FX8D в основном контролирует потребление энергии всей системы через контроллер режима ожидания (SMU) и блок управления тактовой частотой (CMU).

С уровня энергосбережения мы можем разделить энергопотребление на три уровня.

Первый уровень - контролировать рабочие такты модуля через регистр PRR и экономить динамическое энергопотребление системы, отключив такты без использования модуля. В нормальных условиях энергопотребление, сэкономленное на этом уровне, не является очевидным.

Второй этап заключается в переключении основного источника синхронизации на низкочастотный тактовый генератор и отключении неиспользуемого модуля источника синхронизации и других аналоговых модулей. Этот режим может в основном получить экономию очень значительной мощности работы системы и потребления энергии сна.

Третий уровень - позволить системе перейти в режим пониженного энергопотребления. В режиме пониженного энергопотребления LGT8FX8D работает с минимальным потреблением энергии. Этот режим может быть пробужден только внешним выводом прерывания. После пробуждения в режиме пониженного энергопотребления программное обеспечение может считывать состояние предварительного сброса через регистр MCUSR.

5.2. Спящий режим

LGT8FX8D поддерживает пять режимов ожидания (Табл. 5.1).

Пользователь может выбрать подходящий режим ожидания в соответствии с требованиями приложения.

Регистр SMCR содержит параметры управления режимом спящего режима. После выполнения команды SLEEP ядро переходит в спящий режим.

Для лучшего потребления энергии сна рекомендуется отключить все неиспользованные такты и аналоговые блоки, прежде чем ядро переходит в спящий

режим.

Однако следует отметить, что для некоторых источников пробуждения требуются рабочие такты. Если вам нужно использовать такой источник пробуждения, сохраните источник синхронизации.

Таблица 5.1. Спящий режим и режим ожидания

Режим Сна	Эффективные такты				Источник пробуждения							
	CPU_CLK FLASH_CLK	IO_CLK	ADC_CLK	ASY_CLK	PIN Change	INT0 INT1	TWI Address Match	Timer2	ADC	WDT	Other INT	Other I/O
<i>Бездействующий</i>		X	X	X	X	X	X		X	X	X	X
<i>Режим подавления шума при АЦП</i>			X	X		X	X	X	X	X		
<i>Питание / Вниз</i>					X	X	X			X		
<i>Питание / выключение S0 (C RC32K)</i>						X				X		
<i>Питание / выключение S1 (C RC32K)</i>						X						

Если вам нужно войти в пять выше перечисленных режимов ожидания, бит SE в SMCR должен быть установлен в 1 для включения режима спящего режима. Затем выполните инструкцию SLEEP.

SM0 / 1/2 в SMCR используется для выбора другого спящего режима. Для получения дополнительной информации см. Описание ниже.

Когда MCU находится в спящем режиме, если источник пробуждения действителен, MCU проснется через 4 цикла и продолжит выполнение инструкции.

Если прерывание остается в силе, прерывание немедленно ответит на процедуру обслуживания прерываний.

Если сброс системы происходит в режиме SLEEP, MCU также проснется и начнет выполнение из вектора сброса.

Когда MCU находится в режиме питания/выключения, система может просыпаться через внешнее прерывание INT0/1, и MCU возобновит работу с предыдущей позиции после сна.

5.2.1. Режим IDLE (Бездействующий)

Когда SM2 ... 0 установлено значение 000, после выполнения команды SLEEP MCU переходит в режим IDLE, режим IDLE отключает такты работы ядра, а все остальные периферийные устройства будут работать нормально.

Режим IDLE может быть разбужен внешними прерываниями, внутренними прерываниями и т. д. Если вам не нужно использовать компаратор и АЦП в качестве источника пробуждения, рекомендуется отключить его.

Режим IDLE не может получить очевидную экономию энергопотребления, потому что выключено только работающее на нем такты.

В режиме IDLE основное ядро также прекратит выполнение инструкций и выборки, что уменьшит рабочую мощность внутренней программы FLASH.

Режим IDLE имеет более гибкий режим пробуждения. Пользователи могут получать более идеальную рабочую мощность за счет снижения основных тактов системы и закрытия ненужных модулей.

5.2.2. Режим подавления шума при АЦП

Когда SM2 ... 0 установлен в 001, команда SLEEP выполняет MCU в режиме подавления помех при АЦП.

В этом режиме основное и большинство периферийных устройств перестанут работать. Абонентский интерфейс, внешнее прерывание, соответствие TWI-адреса, WDT и таймер/счетчик2, работающие в режиме асинхронных часов, будут работать.

Режим используется для обеспечения хорошей рабочей среды для преобразования АЦП. Уменьшите высокочастотные помехи от цифровых модулей до аналоговых преобразований. После ввода этого режима АЦП автоматически начнет преобразование. После преобразования данных в регистр данных АЦП прерванное прерывание преобразования АЦП разбудит MCU из режима АЦП.

5.2.3. Режим Питание/Вниз

При SM2 ... 0 до 010, после того, как команда SLEEP выполняется, MCU входит в режим Питание/ Выключить.

В этом режиме система отключит всю работу модуля синхронизации. Этот режим работы тактов в связи с закрытием всех модулей, поэтому можно только разбудить асинхронный режим, внешнее прерывание, и работу WDT совпадение адресов режим независимого источника синхронизации TWI в сигнале пробуждения может быть сгенерирована в этом режиме. Эта модель может отключить все модули, за исключением того источника тактового генератора. Для достижения более желательного расхода рабочей мощности, рекомендуется здесь, перед входом в режим, системные такты переключаются на ведущий внутренний или внешний 32 кГц 32K RC низкой частоты генератора, и, чтобы закрыть источник синхронизации не используется, и аналоговый модуль.

5.2.4. Режим Питание/Выключить S0

Когда SM ... 0 установлен на 110, после выполнения инструкции SLEEP, MCU переходит в режим Питание/Выключить S0.

После входа в Питание/Выключить S0 все источники синхронизации отключены, за исключением внутреннего 32KHz RC. Этот режим можно разбудить внешним прерыванием INT0/1, если функция прерывания WDT включена, WDT также может использоваться для реализации синхронизации тактов.

5.2.5. Режим Питание/Выключить S1

Когда SM ... 0 до 011, после выполнения инструкции SLEEP, то MCU переходит в режим Питание/Выключить S1.

После ввода Питание/Выключить S1, все исходные системы тактов закрыты. Этот режим может быть только через внешнее прерывание INT0 / 1 после.

Перед входа в режиме Питание/Выключить, вам нужно установить заранее и пробуждение источника и установить соответствующее пробуждение условие, для внешних прерываний, вам необходимо установить прерывание включить и настроить пороговый уровень, необходимый для применения.

5.2.6. Управление мощностью FLASH и быстрое пробуждение

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

В режиме Питание/Выключить система может использовать внешние прерывания или пробуждение от WDT.

Чтобы отфильтровать возможные помехи от внешних сигналов, внутренняя схема пробуждения включает конфигурируемую схему фильтра. Вы можете выбрать соответствующую ширину фильтра по мере необходимости.

Конфигурация схемы фильтра может быть реализована FWKPEN регистра MCUCR. Управление шириной фильтра MCUCR [FWKPEN] (См. Табл. 5.2.)

Таблица 5.2. Управление шириной фильтра

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

5.3. Регистры Описание

SMCR - Регистр управления спящим режимом

SMCR - Регистр управления спящим режимом								
SMCR: 0x33(0x53)		По умолчанию: 0x00						
Биты	7	6	5	4	3	2	1	0
Имя	-	-	-	-	SM2	SM1	SM0	SE
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Определение бит								
[7:4]	-	Зарезервировано						
[3]	SM2	Выбор режима ожидания						
		SM2	SM1	SM0	Описание режима			
[2]	SM1	0	0	0	Режим IDLE (Бездействующий)			
		0	0	1	Режим подавления шума при АЦП			
		0	1	0	Режим Питание/Вниз			

[1]	SM0	0	1	1	Режим Питание/Выключить S
		1	1	0	Режим Питание/Выключить S0
		1	0	0	Включения/Выключения Блокировки
		Другие			Зарезервировано
[0]	SE	Бит разрешения режима ожидания установлен в «1» - выполняется команда SLEEP, и ядро переходит в спящий режим. Бит SE защищает систему от случайного входа в спящий режим. Рекомендуется установить этот бит в «1» и сразу же выполнить инструкцию SLEEP. После пробуждения рекомендуется немедленно очистить бит SE.			

PRR - Регистр управления энергосбережением								
PRR: 0x64		По умолчанию: 0x00						
Биты	7	6	5	4	3	2	1	0
Имя	PRTWI	PRTIM2	PRTIM0	-	PRTIM1	PRSPI	PRUSART0	PRADC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Определение бит								
[7]	PRTWI	1 - Отключить такты для модуля TWI						
[6]	PRTIM2	1 - Отключить такты для таймер/счетчика 2						
[5]	PRTIM0	1 - Отключить такты для таймер/счетчика 0						
[4]	-	Зарезервировано						
[3]	PRTIM1	1 - Отключить такты для таймер/счетчика 1						
[2]	PRSPI	1 - Отключить такты для модуля SPI						
[1]	PRUSART0	1 - Отключить такты для модуля USART0						
[0]	PRADC	1 - Отключить такты контроллера АЦП						

PRR1 - Регистр управления энергосбережением								
PRR1: 0x65		По умолчанию: 0x00						
Биты	7	6	5	4	3	2	1	0
Имя	-	-	PRWDT	-	-	PREFL	PRPCI	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Определение бит								
[7:6]	-	Зарезервировано						
[5]	PRWDT	1 - Отключить такты WDT						
[4:3]	-	Зарезервировано						
[1]	PRPCI	1 - Отключить внешнее изменение выводов и такты работы внешнего прерывания						
[0]	-	Зарезервировано						

MCUCR - регистр управления MCU								
MCUCR: 0x35(0x55)			По умолчанию: 0x00					
Биты	7	6	5	4	3	2	1	0
Имя	FWKEN	FPDEN	-	PUD	-	-	IVSEL	IVCE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Определение бит								

[7]	FWKEN	Режим быстрого пробуждения активирует управление, действующее только для режима питания/выключения 0: 260us задержка фильтра 1: задержка фильтра 32us
[6]	FPDEN	Управление включением питания / выключения FLASH 0: FLASH остается включенным после системы SLEEP 1: отключение питания FLASH после системы SLEEP
[5]	-	Зарезервировано
[4]	PUD	Глобальные подтягивающие резисторы отключить
[3:2]	-	Зарезервировано
[1]	IVSEL	Бит выбора вектора прерывания. После того, как этот бит установлен, адрес вектора прерывания будет сопоставлен с новым адресом на основе значения регистра IVBASE. Подробный адрес отображения, пожалуйста, обратитесь к описанию регистра IVBASE
[0]	IVCE	Перед изменением IVSEL необходимо установить бит разрешения изменения вектора прерывания, а затем установить IVSEL в течение 6 циклов.

6. Системное управление и сброс

6.1. Обзор

После сброса системы все регистры ввода/вывода устанавливаются на их начальные значения, и программа запускает выполнение из вектора сброса.

На векторном адресе прерывания LGT8FX8D для перехода к обработчику сброса необходимо использовать инструкцию RJMP - относительного перехода.

Если программа не используется для прерывания, а источник прерывания не включен, вектор прерывания не будет использоваться.

Область векторов прерываний может использоваться для хранения кода программы пользователя.

После того, как сброс действителен, все порты ввода/вывода сразу же вступают в исходное состояние. Большинство входов/выходов имеют начальное состояние входа и выключение внутреннего подтягивающего резистора.

Входы/выходы с функциями аналогового ввода также инициализируются функциями цифрового ввода/вывода. Когда сброс становится неактивным, внутренний счетчик таймера LGT8FX8D начинает использовать для сброса напряжения. Ширина сигнала сброса растяжения используется для обеспечения того, чтобы модули мощности и синхронизации в системе вошли в стабильное состояние.

6.2. Источник сброса

LGT8FX8D поддерживает в общей сложности шесть источников сброса:

- Сброс при включении питания: Срабатывание при включении включено, когда рабочее напряжение системы низкое, а порог сброса внутреннего модуля POR сбрасывается.

- Внешний сброс: когда на выводе RESETN микросхемы появляется импульс, превышающий минимальную ширину сброса, действительный внешний сброс.
- Сброс сторожевого таймера: после того, как модуль сторожевого таймера включен, если таймер сторожевого таймера отключился, система перезагрузится.
- Сброс низкого напряжения: LGT8FX8D имеет модуль обнаружения низкого напряжения (LVD) внутри. Когда источник питания системы ниже порогового значения сброса, установленного LVD, MCU также будет сброшен.
- Сброс программного обеспечения: LGT8FX8D имеет специальный программный сгенерированный регистр сброса. Пользователи могут сбросить MCU в любое время через этот регистр.
- Сброс OCD: сброс OCD выдается модулем отладчика и используется для прямого сброса ядра MCU.

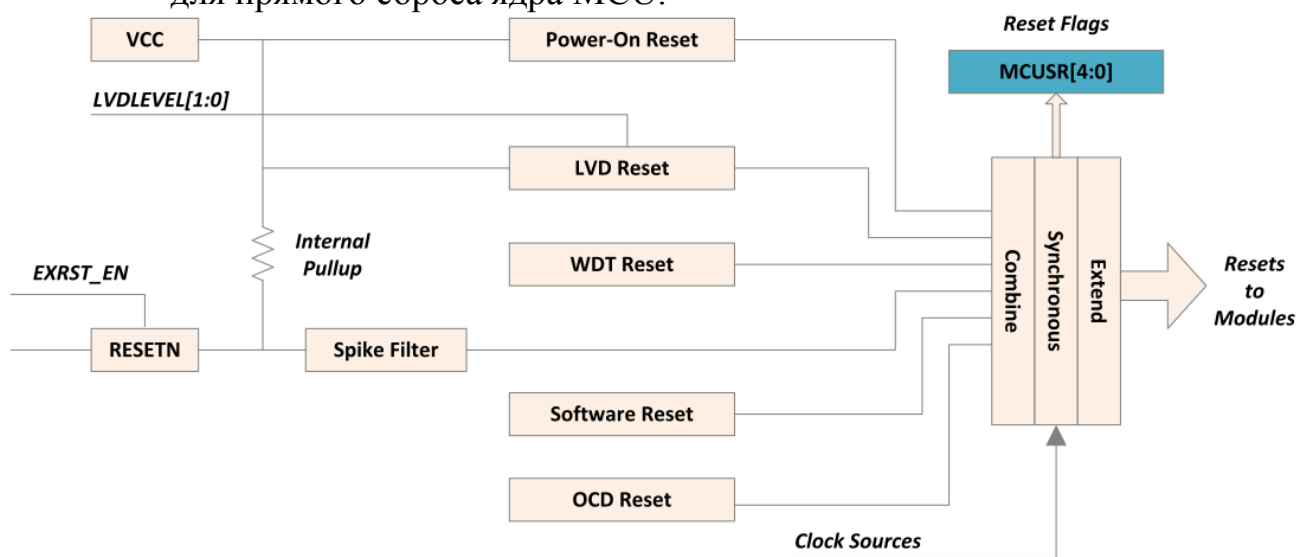


Рис.6.1 Структура системы Сброс

6.2.1. Сброс при включении.

Сигнал сброса при включении генерируется схемой обнаружения внутреннего напряжения. Когда системный источник питания (Vcc) ниже порогового значения обнаружения, действует сигнал сброса включения питания.

Порог обнаружения сброса при включении питания см. в разделе Электрических параметров.

Схема сброса при включении гарантирует, что чип находится в состоянии перезапуска во время включения питания. После включения микросхемы он может начать работать из известного стабильного состояния.

Сигнал сброса включения питания также растягивается внутренним счетчиком микросхемы, чтобы гарантировать, что различные внутренние аналоговые модули, такие как RC-генератор, могут войти в стабильное рабочее состояние после включения питания.

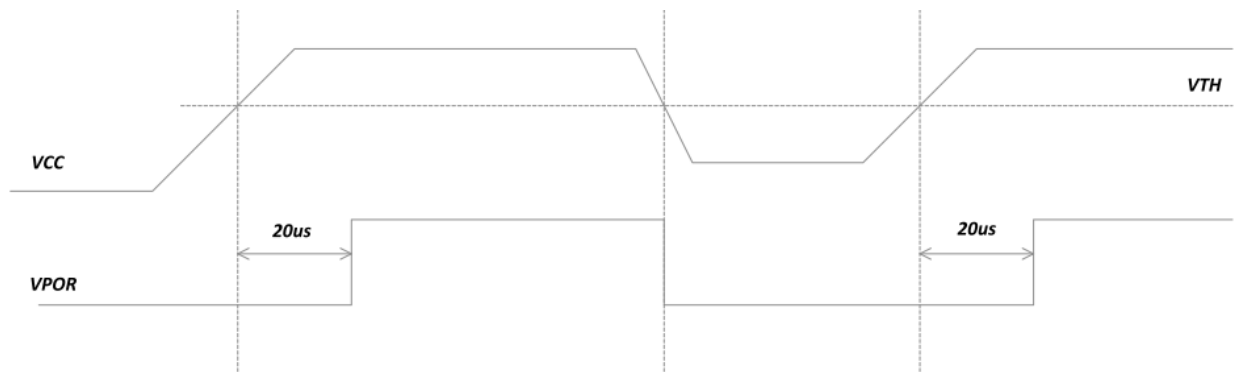


Рис.6.2. Диаграмма Сброса при включении питания.

6.2.2. Внешний сброс

Низкий уровень применяется к выходу внешнего сброса (RSTN), и внешний сброс активен немедленно. Ширина нижнего уровня должна быть больше, чем минимальная ширина импульса импульса. Внешний сброс - это асинхронный сброс. Даже если чип не имеет тактов, внешний сброс может по-прежнему перезагружать чип. Вывод внешнего сброса LGT8FX8D также может использоваться как входы/выходы общего назначения.

После включения микросхемы по умолчанию используется внешняя функция сброса. Пользователь может отключить функцию внешнего сброса этого вывода через конфигурацию регистра, чтобы он мог использоваться как обычный ввод-вывод. Подробнее см. Описание регистра IOCR.

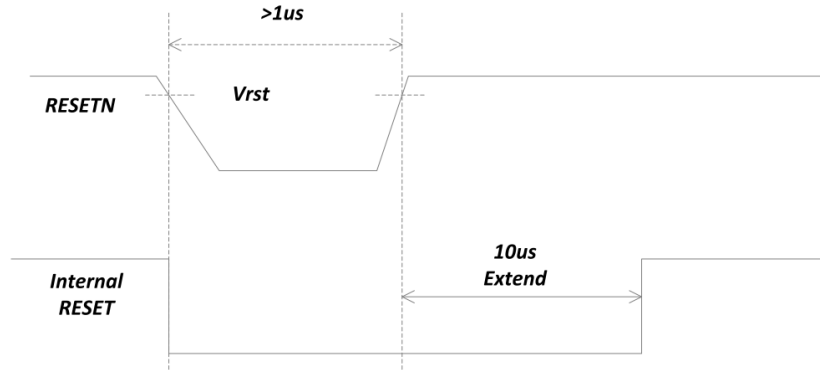


Рис.6.3. Диаграмма Внешнего сброса

6.2.3. Обнаружение низкого напряжения (LVD)

Сброс LGT8FX8D содержит программируемую схему низковольтного обнаружения (LVD). LVD также обнаруживает изменение напряжения VCC, но в отличие от сброса при включении, LVD может выбрать порог обнаружения напряжения. Пользователь может выбирать между тремя различными порогами напряжения через биты конфигурации системы или напрямую, управляя регистром VDTCSR. Схема обнаружения напряжения LVD имеет характеристику гистерезиса от ± 10 мВ до ± 50 мВ, которая используется для фильтрации джиттера напряжения VCC. Когда LVD включен, сброс LVD будет действовать немедленно, если напряжение VCC падает до порога установленного сброса. Когда VCC увеличивается выше порога сброса, начинается схема расширения внутреннего

сброса, и сброс продолжает расширяться как минимум на 1 миллисекунду.

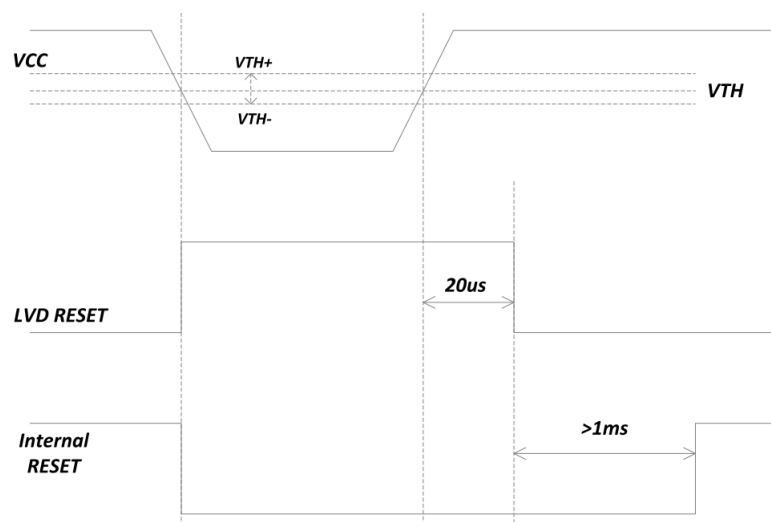


Рис.6.4. Диаграмма Сброса при обнаружение низкого напряжения (LVD)

6.2.4. Сброс сторожевого таймера

Если функция сброса системы сторожевого таймера включена, когда таймер сторожевого таймера переполняется, сразу будет генерироваться сигнал с одним циклом сброса системы. Сигнал сброса сторожевого таймера также будет расширен внутренним счетчиком задержки. Подробную информацию о работе сторожевого контроллера см. В подробном описании ниже.

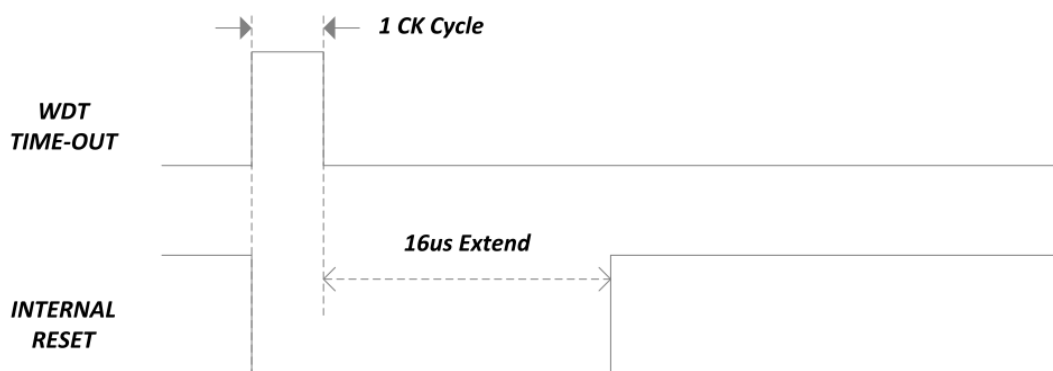


Рис.6.5. Диаграмма Сброс сторожевого таймера

6.2.5. Сброс программного обеспечения, сброс OCD

Сброс программного обеспечения - это шестой бит-запуск пользователя через работу регистра VDTCCR.

Время сброса программного обеспечения точно такие же, как и сброс сторожевого таймера. Внутренний сигнал сброса увеличивается на 16us.

Сброс OCD генерируется внутренним отладчиком. Сброс OCD обычно контролируется отладчиком. Программное обеспечение пользователя не может инициировать сброс OCD.

6.3. Управление мультиплексированием контактов

(В оригинале нет описания.)

6.4. Сторожевой таймер

- Выбор по времени внутреннего 32KHz RC или внутреннего 32MHz RC 16-точечного деления (2MHz)
- Поддержка режима прерывания, режима сброса и режима прерывания возврата
- Тайм-аут таймера до 8 секунд LGT8FX8D внутренне содержит улучшенный внешний вид Дверной доводчик (WDT).

Рабочие такты таймера WDT могут быть внутренним 32KHz RC генератором или внутренним 32MHz RC генератором, разделенным на 16.

После переполнения счетчика WDT может быть выведен сигнал прерывания или сброса системы.

При нормальном использовании программное обеспечение требуется для выполнения команды сброса WDR - сторожевого таймера, чтобы перезапустить счетчик перед переполнением. Если в системе нет команды выполнения WDR, WDT будет генерировать прерывание или сброс системы. Структура сторожевого таймера, как показано на Рис.7.1.

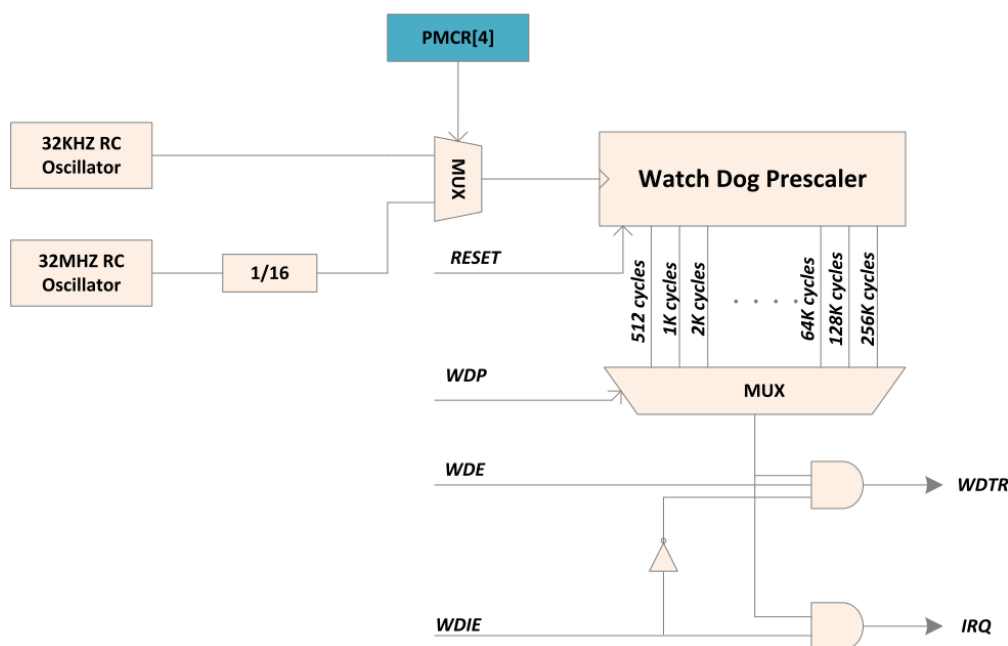


Рис.7.1 Структура сторожевого таймера

В режиме прерывания сигнал запроса прерывания генерируется после переполнения WDT. Это прерывание может использоваться как сигнал пробуждения в режиме ожидания или как общий системный таймер.

Например, это прерывание может использоваться для ограничения времени выполнения операции и завершения текущей задачи при переполнении.

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

Третий режим, режим прерывания сброса, объединяет функции прерывания и

сброса.

Во-первых, система ответит на функцию прерывания WDT и выйдет из программы сброса прерываний WDT, сразу переключившись в режим сброса. Эта функция может поддерживать сохранение некоторых параметров ключа перед сбросом.

Чтобы предотвратить случайное отключение WDT, закрытие WDT должно выполняться с строго определенным временем.

Следующий код описывает, как отключить сторожевой таймер.

В следующем примере предполагается, что прерывание отключено, так что весь процесс операции не будет прерван.

Код Ассемблера

WDT_OFF:

; Выключить глобальное прерывание

CLI; Таймер сторожа сброса

WDR

; Очистить WDRF в MCUSR

IN r16, MCUSR

ANDI r16, ~ (1 << WDRF)

OUT MCUSR, r16

; Записать логическую единицу в WDCE и WDE

; Сохраните старое установку Делителя частоты, чтобы предотвратить неумышленный тайм-аут

LDS r16, WDTCSR

ORI r16, (1 << WDCE) | (1 << WDE)

STS WDTCSR, r16

; Выключить WDT

LDI r16, (0 << WDE)

STS WDTCSR, r16

; Включить глобальное прерывание

SEI

RET

Код C

void WDT_OFF (void)

{

__disable_interrupt ();

__watchdog_reset ();

/ Очистить WDRF в MCUSR */*

MCUSR &= ~ (1 << WDRF);

/ Записать логическую единицу в WDCE и WDE */*

/ Сохраните старое установку Делителя частоты, чтобы предотвратить неумышленный тайм-аут */*

WDTCSR |= (1 << WDCE) | (1 << WDE);

/ Выключить WDT */*

WDTCSR = 0x00;

__enable_interrupt ();

}

[Подсказка] 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 тайм-аута сторожевого таймера.

Код Ассемблера

WDT TOV Change:

; Выключить глобальное прерывание

CLI

; Таймер сторожа сброса

WDR

; Пуск засек время последовательности

LDS r16, WDTCSR

ORI r16, (1 <<WDCE) | (1 <<WDE)

STS WDTCsr, r16

; - Извлеч для в циклов устанавливает новое значение отсюда -

; Установить новое значение паузы = циклы 64K циклов

LDI r16, (1 <<WDE) | (1 <<WDP2) | (1 <<WDP0)

STS WDTC SR, r16

; - Завершено задание нового значения, используется 2 цикла-

; Включить глобальное прерывание

SEI

RET

Код С

```
void WDT_TOV_Change (void)
```

 $\{$

```
__disable_interrupt ();
```

```
__ watchdog_reset ();
```

/ Начальная синхронизированная последовательность */*

$$\text{WDTCsr} \mid = (1 \ll \text{WDCE}) \mid (1 \ll \text{WDE});$$

/ Установить новое значение тайм-аута = 64К циклов */*

$$\text{WDTCsr} = (1 \ll \text{WDE}) \mid (1 \ll \text{WDP2}) \mid (1 \ll \text{WDP0});$$

```
__enable_interrupt();
```

}

[Инструкции по использованию] Перед изменением бит конфигурации WDP рекомендуется сбросить сторожевой таймер. Поскольку изменение бит WDP до небольшого периода ожидания очень вероятно приведет к сбросу тайм-аута сторожевого таймера.

6.5. Регистры Описание

VDTCR - управляющий регистр LVD

VDTCR - управляющий регистр LVD								
VDTCR: 0x62		Значение по умолчанию: 0x00 или загружено из информации о конфигурации системы						
Биты	7	6	5	4	3	2	1	0
Имя	CE	SWR	-	-	-	VDTS1	VDTS0	VDTEN
R/W	R/W	W				R/W	R/W	R/W
Определение бит								
[7]	CE	Бит разрешения изменения значения VDTCR. Прежде чем пользователь сможет						

		установлен, когда бит WDE должен быть очищен, иначе WDT не будет отключен. Когда бит WDTOE установлен, аппаратное обеспечение очищает бит WDTOE после 4 тактов.
[3]	WDE	<p>WDT активирует бит управления. Когда бит WDE установлен в «1», WDT включен. Когда бит WDE установлен на «0», WDT отключен. WDE может быть очищен только при установке бит WDTOE.</p> <p>Чтобы отключить WDT, который уже включен, необходимо выполнить следующую последовательность:</p> <ol style="list-style-type: none"> 1. Установив биты WDTOE и WDE одновременно, даже если WDE был установлен, необходимо записать «1» в бит WDE перед запуском операции выключения. 2. Записать бит WDE в следующие 4 такта. «0» Это отключит WDT. WDT сбрасывает флаг WDRF системы (в регистре MCUSR), когда бит WDE установлен на «1» и переполнение WDT для сброса системы. Бит WDE устанавливается, когда бит WDRF установлен. Поэтому, чтобы очистить бит WDE, бит WDRF должен быть очищен первым.
[2:0]	WDP[2:0]	Контроль выбора предварительного делителя WDT. Используется для установки цикла переполнения WDT. Рекомендуется изменить значение WDP, когда WDT не считает. Изменение значения WDP во время подсчета приведет к непредсказуемому переполнению WDT.

Таблиц 8.1. Список выбора предварительной выборки Watchdog

WDP3	WDP2	WDP1	WDP0	Циклы переполнения сторожевого таймера	Время при 32кГц	Время при 2МГц
0	0	0	0	2K cycles	64ms	1ms
0	0	0	1	4K cycles	128ms	2ms
0	0	1	0	8K cycles	256ms	4ms
0	0	1	1	16K cycles	512ms	8ms
0	1	0	0	32K cycles	1s	16ms
0	1	0	1	64K cycles	2s	32ms
0	1	1	0	128K cycles	4s	64ms
0	1	1	1	256K cycles	8s	128ms
1	0	0	0	512K cycles	16s	256ms
1	0	0	1	1024K cycles	32s	512ms
1	0	1	0	Зарезервировано		
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

7. Прерывания и вектора прерывания

- 28 источников прерываний
- Программируемые начальные адреса вектора

LGT8F48D / 88D / 168D / 328D имеют одинаковые ресурсы прерывания.

Основное различие заключается в том, что вектор прерывания LGT8F48D/88D - это 1 слово инструкции (16 бит). Вектор прерывания LGT8F168D/328D - 2 слова.

7.1. Вектора прерываний LGT8F48D/88D

Список векторов прерываний LGT8F48D/88D

Номер	Адрес вектора	Источник 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	INT1	Запрос внешнего прерывания 1
4	0x0003	PCI0	Прерывание от изменения уровня сигнала 0
5	0x0004	PCI1	Прерывание от изменения уровня сигнала 1
6	0x0005	PCI2	Прерывание от изменения уровня сигнала 2
7	0x0006	WDT	Прерывание переполнения сторожевого таймера
8	0x0007	TC2 COMPA	Таймер 2 Сравнение А Прерывание при равенстве
9	0x0008	TC2 COMPB	Таймер 2 Сравнение А Прерывание при равенстве
10	0x0009	TC2 OVF	Прерывание при переполнении таймера 2
11	0x000A	TC1 CAPT	Прерывание от входного сигнала таймера 1
12	0x000B	TC1 COMPA	Таймер 1 Сравнение А Прерывание при равенстве
13	0x000C	TC1 COMPB	Таймер 1 Сравнение В Прерывание при равенстве
14	0x000D	TC1 OVF	Прерывание при переполнении таймера 1
15	0x000E	TC0 COMPA	Таймер 0 Сравнение А Прерывание при равенстве
16	0x000F	TC0 COMPB	Таймер 0 Сравнение В Прерывание при равенстве
17	0x0010	TC0 OVF	Прерывание при переполнении таймера 0
18	0x0011	SPI STC	SPI Прерывание конец передачи серии
19	0x0012	USART RXC	USART Прерывание после получения
20	0x0013	USART UDRE	USART Прерывание Освобождения регистра данных
21	0x0014	USART TXC	USART Прерывание Конец передачи
22	0x0015	ADC	АЦП Прерывание Конец преобразования
23	0x0016	EE_RDY	EEPROM Прерывание Готов
24	0x0017	ANA_COMP	Аналоговый компаратор 0 Прерывание
25	0x0018	TWI	TWI Прерывание
26	0x0019	ANA_COMP1	Аналоговый компаратор 1 Прерывание
27	0x001A	-	
28	0x001B	PCI3	Прерывание от изменения уровня сигнала 3
29	0x001C	OPA0_COMP	OPA0 встроенный Таймер совпадение прерываний
30	0x001D	OPA1_COMP	OPA1 встроенный Таймер совпадение прерываний

7.2. Вектора прерываний LGT8F168D/328D

Список векторов прерываний LGT8F168D/328D

Номер	Адрес вектора	Источник 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	INT1	Запрос внешнего прерывания 1
4	0x0006	PCI0	Прерывание от изменения уровня сигнала 0

5	0x0008	PCI1	Прерывание от изменения уровня сигнала 1
6	0x000a	PCI2	Прерывание от изменения уровня сигнала 2
7	0x000c	WDT	Прерывание переполнения сторожевого таймера
8	0x000e	TC2 COMPA	Таймер 2 Сравнение А Прерывание при равенстве
9	0x0010	TC2 COMPB	Таймер 2 Сравнение А Прерывание при равенстве
10	0x0012	TC2 OVF	Прерывание при переполнении таймера 2
11	0x0014	TC1 CAPT	Прерывание от входного сигнала таймера 1
12	0x0016	TC1 COMPA	Таймер 1 Сравнение А Прерывание при равенстве
13	0x0018	TC1 COMPB	Таймер 1 Сравнение В Прерывание при равенстве
14	0x001a	TC1 OVF	Прерывание при переполнении таймера 1
15	0x001c	TC0 COMPA	Таймер 0 Сравнение А Прерывание при равенстве
16	0x001e	TC0 COMPB	Таймер 0 Сравнение В Прерывание при равенстве
17	0x0020	TC0 OVF	Прерывание при переполнении таймера 0
18	0x0022	SPI STC	SPI Прерывание конец передачи серии
19	0x0024	USART RXC	USART Прерывание после получения
20	0x0026	USART UDRE	USART Прервать Освобождения регистра данных
21	0x0028	USART TXC	USART Прерывание Конец передачи
22	0x002a	ADC	АЦП Прерывание Конец преобразования
23	0x002c	EE_RDY	EEPROM Прерывание Готов
24	0x002e	ANA_COMP	Аналоговый компаратор 0 Прерывание
25	0x0030	TWI	TWI Прерывание
26	0x0032	ANA_COMP1	Аналоговый компаратор 1 Прерывание
27	0x0034	-	
28	0x0036	PCI3	Прерывание от изменения уровня сигнала 3
29	0x0038	OPA0_COMP	OPA0 встроенный Таймер совпадение прерываний
30	0x003a	OPA1_COMP	OPA1 встроенный Таймер совпадение прерываний

Вектор сброса LGT8FX8D начинается с адреса 0x0000. В дополнение к вектору сброса другие векторные адреса могут быть перенаправлены на начальный адрес с 512 байтами с помощью регистров **IVSEL** и **IVBASE** в регистре **MCUCR**.

7.3. Обработка векторных прерываний

Следующий код использует LGT8F48D / 88D в качестве примера для иллюстрации программирования векторов сброса и прерывания, только для справки:

Пример кода Ассемблера - LGT8F48D / 88D

Адрес	Команда	Комментарий
0x0000	RJMP RESET	<i>;Внешний сброс, Сброс при включении, Сброс сторожевого таймера, Сброс отладки SWD, Сброс низкого ;напряжения</i>
0x0001	RJMP EXT_INT0	<i>;Запрос внешнего прерывания 0</i>
0x0002	RJMP EXT_INT1	<i>;Запрос внешнего прерывания 1</i>
0x0003	RJMP PCIO	<i>;Прерывание от изменения уровня сигнала 0</i>
0x0004	RJMP PCI1	<i>;Прерывание от изменения уровня сигнала 1</i>
0x0005	RJMP PCI2	<i>;Прерывание от изменения уровня сигнала 2</i>
0x0006	RJMP WDT	<i>;Прерывание переполнения сторожевого таймера</i>
0x0007	RJMP TC2_COMPA	<i>;Таймер 2 Сравнение А Прерывание при равенстве</i>

0x0008	RJMP TC2_COMPB	;Таймер 2 Сравнение A Прерывание при равенстве
0x0009	RJMP TC2_OVF	;Прерывание при переполнении таймера 2
0x000A	RJMP TC1_CAPT	;Прерывание от входного сигнала таймера 1
0x000B	RJMP TC1_COMPA	;Таймер 1 Сравнение A Прерывание при равенстве
0x000C	RJMP TC1_COMPB	;Таймер 1 Сравнение B Прерывание при равенстве
0x000D	RJMP TC1_OVF	;Прерывание при переполнении таймера 1
0x000E	RJMP TC0_COMPA	;Таймер 0 Сравнение A Прерывание при равенстве
0x000F	RJMP TC0_COMPB	;Таймер 0 Сравнение B Прерывание при равенстве
0x0010	RJMP TC0_OVF	;Прерывание при переполнении таймера 0
0x0011	RJMP SPI_STC	;SPI Прерывание конец передачи серии
0x0012	RJMP USART_RXC	;USART Прерывание после получения
0x0013	RJMP USART_UDRE	;USART Прервать Освобождения регистра данных
0x0014	RJMP USART_TXC	;USART Прерывание Конец передачи
0x0015	RJMP ADC	;АЦП Прерывание Конец преобразования
0x0016	RJMP EE_RDY	;EEPROM Прерывание Готов
0x0017	RJMP ANA_COMP	;Аналоговый компаратор 0 Прерывание
0x0018	RJMP TWI	;TWI Прерывание
0x0019	NOP	;Аналоговый компаратор 1 Прерывание
0x001A	NOP	
0x001B	RJMP PCI3	;Прерывание от изменения уровня сигнала 3
		;
0x001C	LDI r16, high(RAMEND)	;Основная программа Начало
0x001D	OUT SPH, r16	;Установка указателя стека
0x001E	LDI r16, low(RAMEND)	
0x001F	OUT SPL, r16	
0x0020	SEI	;Включить глобальные прерывания
0x0021	

7.4. Регистры Описание

IVBASE - Регистр базового адреса вектора прерываний

IVBASE - регистр базового адреса вектора прерываний		
IVBASE: 0x75		По умолчанию: 0x00
Биты		IVBASE [7:0]
R/W		R/W
Определение бит		
[7:0]	IVBASE	Если IVSEL равен 1, вектор прерывания (кроме вектора сброса) будет переназначен на 512-байтные страницы с использованием IVBASE в качестве базового адреса. Базовый адрес отображаемого вектора прерывания: (IVBASE << 8) + соответствующий векторный адрес в таблице 1

8. Внешние прерывания

- Два внешних источника прерывания
- Конфигурируемые прерывания с уровнем или по фронту
- Может использоваться как источник пробуждения в спящем режиме

8.1. Обзор

Внешние прерывания запускаются с помощью контактов INT0 и INT1. Пока внешнее прерывание включено, 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, настроенным внешним регистром управления прерываниями, EICRA.

Когда внешнее прерывание включено и настроено как инициированное уровнем (только контакты INT0 и INT1), прерывание будет генерироваться до тех пор, пока контакт не будет низким. Восходящий или падающий фронт контактов INT0 и INT1 запускает нормальную работу тактов ввода-вывода, а прерывания с низким уровнем срабатывания контактов INT0 и INT1 регистрируются асинхронно.

В дополнение к режиму ожидания, такты IO в других режимах ожидания перестают работать. Поэтому эти 2 внешних прерывания могут использоваться в качестве источников пробуждения в режимах сна, отличных от режима ожидания.

Если прерывание с использованием уровня сигнала используется в качестве источника пробуждения в энергосберегающем режиме, измененный уровень должен удерживаться в течение определенного промежутка времени, чтобы разбудить MCU, чтобы уменьшить чувствительность микроконтроллера к шуму. Требуемый уровень должен быть достаточно длительным, чтобы MCU завершил процесс пробуждения, а затем инициировал прерывание уровня.

8.2. Регистры Описание

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

IOCR - IO Регистр регистрации особых функций								
EICRA: 0x69		По умолчанию: 0x00						
Биты	7	6	5	4	3	2	1	0
Имя	-	-	-	-	ISC11	ISC10	ISC01	ISC00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Определение бит								
[7:4]	-	Зарезервировано						
[3]	ISC11	INT1 - Биты управления запуском прерывания						
		Когда глобальное прерывание разрешено и бит управления маской прерывания регистра GICR установлен, внешнее прерывание активируется выводом INT1.						
[2]	ISC10	ISC1 [1: 0]	Описание					
		0	Прерывание по низкому уровню					
		1	Прерывание по нарастающий или спадающий фронт					
		2	Прерывание по спадающему фронту					
		3	Прерывание по нарастающему фронту					

[1]	ISC01	INT0 - Биты управления запуском прерывания Когда глобальное прерывание разрешено и бит управления маской прерывания регистра GICR установлен, внешнее прерывание активируется выводом INT0.	
		ISC0 [1: 0]	Описание
[0]	ISC00	0	Прерывание по низкому уровню
		1	Прерывание по нарастающий или спадающий фронт
		2	Прерывание по спадающему фронту
		3	Прерывание по нарастающему фронту

EIMSK - Регистр маски внешнего прерывания

EIMSK - регистр маски внешнего прерывания								
EIMSK: 0x3D			По умолчанию: 0x00					
Биты	7	6	5	4	3	2	1	0
Имя	-	-	-	-	-	-	INT1	INT0
R/W	-	-	-	-	-	-	R/W	R/W
Определение бит								
[7:3]	-		Зарезервировано					
[1]	INT1		Бит разрешения прерывания внешнего вывода 1. Если бит INT1 установлен в «1», а глобальное прерывание установлено, прерывание внешнего вывода 1 включено и активирована функция пробуждения. Даже если вывод INT1 сконфигурирован как выход, прерывание будет генерироваться до тех пор, пока уровень штыря соответственно изменится. Когда бит INT1 установлен на «0», прерывание внешнего вывода 1 отключено, а функция пробуждения также отключена.					
[0]	INT0		Бит разрешения прерывания внешнего вывода 0. Если бит INT0 установлен в «1», а глобальное прерывание установлено, прерывание внешнего вывода 0 включено и активирована функция пробуждения. Даже если вывод INT0 сконфигурирован как выход, прерывание будет генерироваться до тех пор, пока уровень штыря соответственно изменится. Когда бит INT0 установлен на «0», прерывание внешнего вывода 0 отключено, а функция пробуждения также отключена.					

EIFR - Регистр флага внешнего прерывания

EIFR - регистр флага внешнего прерывания								
EIFR: 0x3C			По умолчанию: 0x00					
Биты	7	6	5	4	3	2	1	0
Имя	-	-	-	-	-	-	INTF1	INTF0
R/W	-	-	-	-	-	-	R/W	R/W
Определение бит								
[7:3]	-		Зарезервировано					
[1]	INTF1		Внешний флаг прерывания INTF1 устанавливается по прерыванию от внешнего вывода 1. Если прерывание внешнего вывода 1 включено с битом INT1EN, установленным на «1», и установлен флаг глобального прерывания, будет генерироваться прерывание внешнего вывода 1. INTF1 автоматически очищается, когда выполняется эта процедура прерывания, или ее можно очистить, записав «1» в бит INTF1.					
[0]	INTF0		Внешний флаг прерывания INTF0 устанавливается по прерыванию от внешнего вывода 0. Если прерывание внешнего вывода 0 включено с битом INT0EN, установленным на «1», и установлен флаг глобального прерывания, будет генерироваться прерывание внешнего вывода 0. INTF0 автоматически очищается, когда выполняется эта процедура прерывания, или ее можно очистить, записав «1» в бит INTF0					

9. Подсистем ввода / вывода

9.1. Обзор

Все микроконтроллеры, основанные на реализации семейства базовых семейств LGT8, имеют возможности чтения / изменения порта ввода-вывода. Это означает, что статус порта можно индивидуально изменить с помощью инструкций SBI и CBI, не затрагивая никаких других операций ввода-вывода.

Аналогичным образом, изменение направления порта или управление его нагрузочным резистором может быть одинаковым.

Большинство входов / выходов LGT8FX8D имеют симметричные характеристики привода, которые могут управлять и выдавать большие токи.

I / O имеет два уровня возможностей привода, и пользователь может управлять возможностями привода каждой группы операций ввода-вывода.

Возможности привода ввода-вывода могут напрямую управлять некоторыми светодиодами.

Большинство входов / выходов LGT8FX8D могут работать до 16 мА и могут использоваться непосредственно для управления светодиодами сегментного кода.

Все Vcc и GND I / O напрямую имеют отдельные диоды защиты от электростатического разряда, предназначенные для выдерживания импульсов ESD не менее 5000 В.

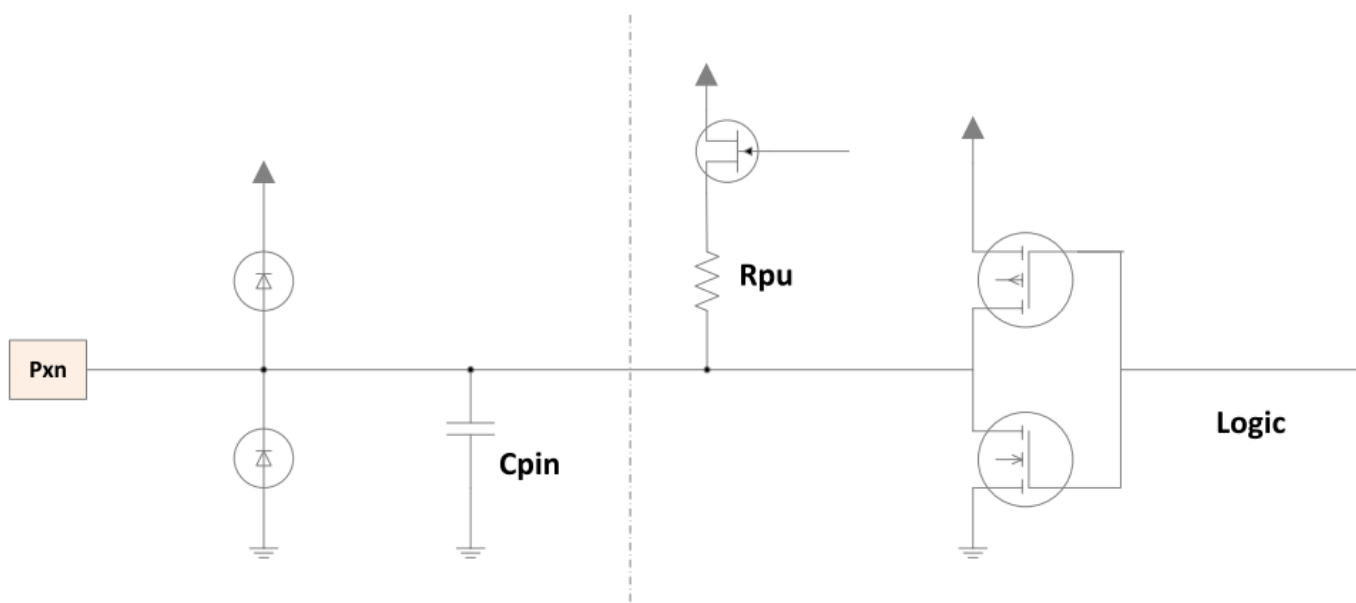


Рис.9.1 Эквивалентной схемы ввода / вывода

Обратитесь к главе о мультиплексировании функций портов для конкретных функций повторного использования. Следует отметить, что включение функций мультиплексирования для определенных портов не влияет на использование этих портов в качестве цифрового ввода-вывода.

Кроме того, для некоторых альтернативных функций могут потребоваться регистры ввода / вывода для управления направлением ввода/вывода порта. Конкретные настройки будут введены в документации каждого мультиплексного модуля.

9.2. Порты ввода/вывода общего назначения

В качестве входов/выходов общего назначения порты представляют собой двунаправленные порты ввода-вывода с внутренними программируемыми подтягиваниями.

9.3. Конфигурация использования портов

Каждый порт управляется тремя битами регистров: DDxn, PORTxn и PINxn. Среди них DDxn можно использовать для настройки через 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 Pxn настроен как выходной порт. Если DDxn установлен в 0, Pxn настроен как входной порт.

Если бит PORTxn записывается в 1, и этот порт сконфигурирован как входной порт, то подтягивающий резистор для этого порта действителен.

Если вы хотите отключить нагрузочный резистор порта, PORTxn должен быть записан как 0 или настроить этот порт как выходной порт.

Состояние инициализации сброса порта - это входное состояние, а нагрузочный резистор недействителен. PORTxn установлен в 1, и этот порт сконфигурирован как выходной порт, а внешний порт будет управляться высоким. Если для параметра PORTxn установлено значение 0, порт будет отключен.

9.4. Переключение ввода/вывода

Когда состояние ввода/вывода переключается между трехступенчатым ([DDxn, PORTxn] = 0b00) и высоким выходом ([DDxn, PORTxn] = 0b11), появится подтягивание или вывод портов. Среднее состояние низкое.

В общем, подтягивающие резисторы приемлемы, потому что в условиях с высоким импедансом разница между высокой скоростью и подъемом не важна. Если это не так, функция вытягивания порта может быть отключена через бит PUD в регистре MCUCR. Аналогично, та же проблема возникает при низком переключении между входами и выходами с включенным подтягиванием.

Пользователь должен использовать три-состояние ([DDxn, PORTxn] = 0b00) или высокий выход ([DDxn, PORTxn] = 0b11) в качестве промежуточного состояния.

Таблица 9.1. 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	Вытекание тока при внешнем 0
0	1	1	Запись	Запрет	Три состояния (high-z)
1	0	x	Выход	Запрет	Вывод низкий
1	1	x	Выход	Запрет	Высокий выход

9.5. Читать значение порта

Независимо от установки DDxn бита порта, текущее состояние порта можно считывать с помощью битов регистра PINxn.

Чтобы избежать метастабильного состояния, вызванного 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, а время задержки составляет не более 1 полусистемного периода. Время задержки Tpdmax и Tpdmin показано на Рис.9.3.

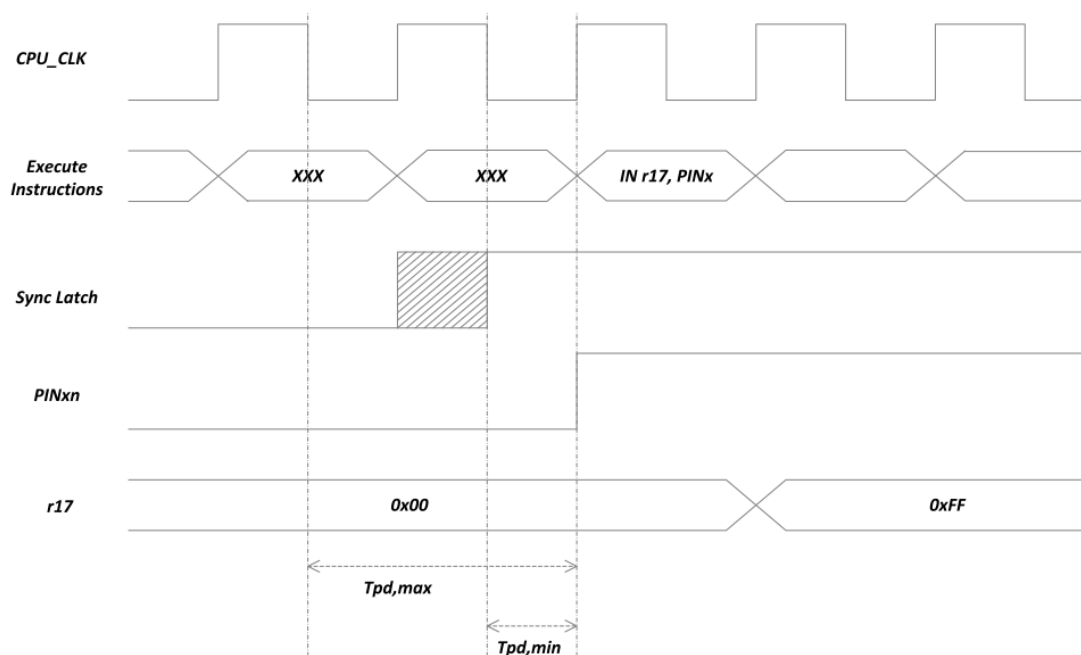


Рис.9.3. Диаграмма Время задержки Tpdmax и Tpdmin.

Мы предполагаем, что системный цикл начинается с первого падающего фронта системных тактов. Защелка фиксирует данные, когда такты низкие, и данные фиксируются, когда такты высоки, как показано заштрихованной частью на рисунке выше.

Когда Такты низкие, данные порта фиксируются и синхронизируются в регистре PINxn на переднем фронте следующих тактов.

На приведенном Рис.9.3. Tpdmax и Tpdmin являются максимальными и минимальными задержками данных порта и делятся на 1,5 периода и 0,5 периода.

Если вы хотите прочитать значение порта для программного обеспечения, вам необходимо вставить инструкцию NOP в поддержку записи и чтения байтов ввода-вывода. Временной интервал T_{pd} показан на Рис.9.4.

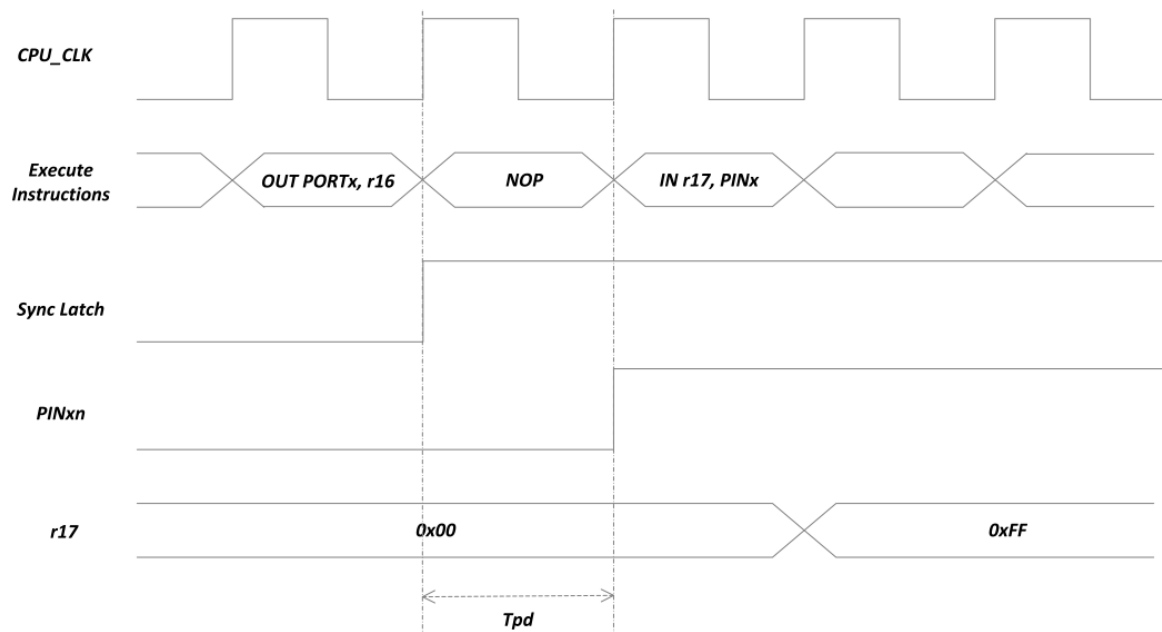


Рис.9.4. Диаграмма Временной интервал T_{pd}

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

Код Ассемблера

```
; Определите подтягивания и установите выходы высоко
; Определение направлений для выводов портов
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 пор для синхронизации
NOP
; Чтение порта
IN r16, PINB
```

Код Си

```
unsigned char I;
/* Определите подтягивания и установите выходы высоко */
/* Определение направления для выводов портов */
PORTB = (1<<PB7)|(1<<PB6)|(1<<PB1)|(1<<PB0);
DDRB = (1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0);
/* Вставить пор для синхронизации */
__no_operation();
/* Чтение порта */
I = PINB;
```

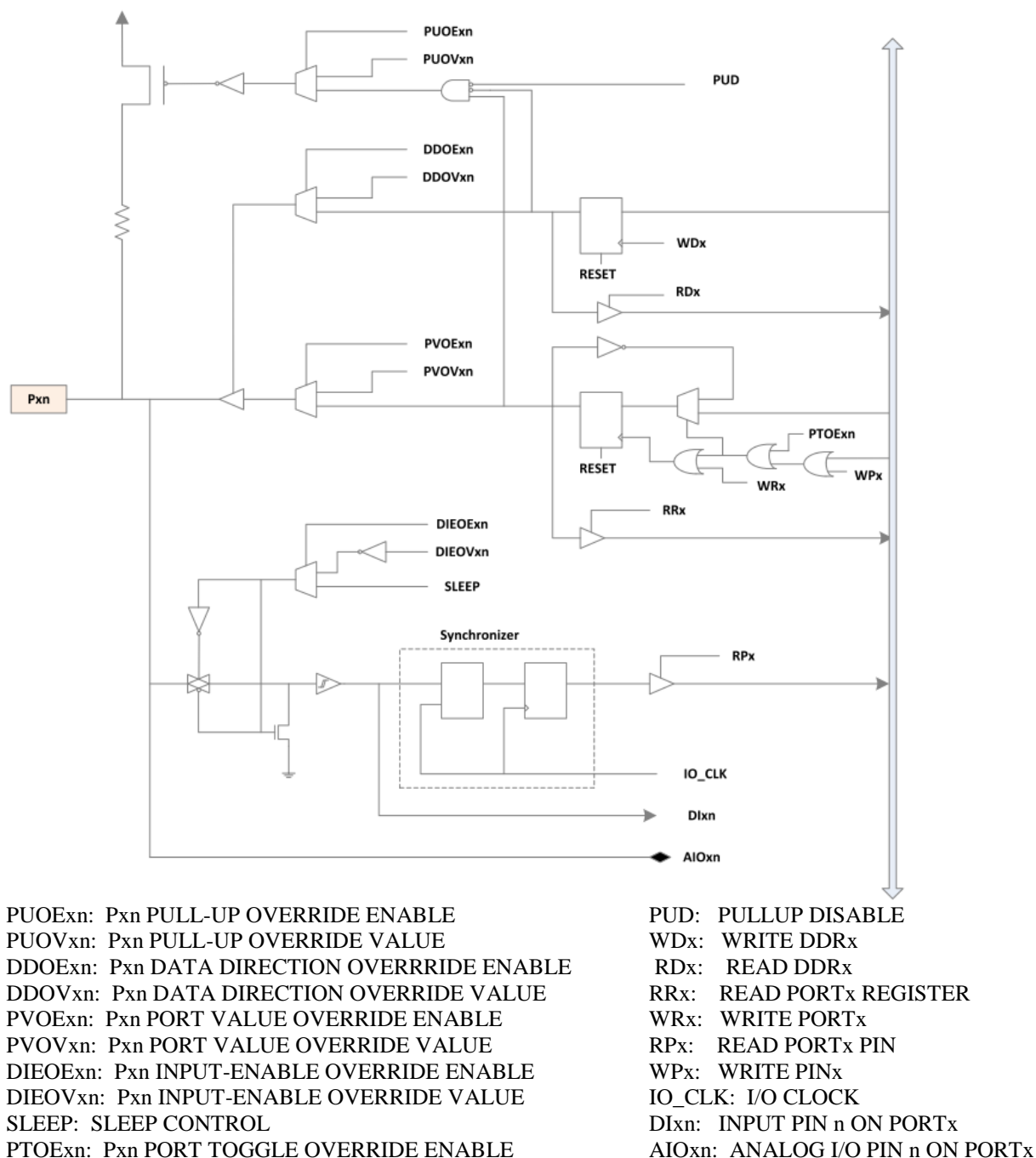


Рис.9.5. Эквивалентная схема мультиплексирования портов

9.6. Включение ввода и управление сном

Из эквивалентной принципиальной схемы ввода/вывода видно, что цифровой вход может быть зажат до уровня земли под управлением сигнала SLEEP. Сигнал SLEEP управляется контроллером сна MCU и различными режимами ожидания. Это гарантирует, что система не будет течь из-за входа порта, плавающего после входа в спящий режим. Управление SLEEP порта заменяется внешней функцией прерывания. Если запрос внешнего прерывания недействителен, управление SLEEP все еще работает. Функция управления SLEEP также будет заменена другими вторыми функциями. Подробнее см. Следующее введение о второй функции порта.

9.6.1. Управление свободными портами

Если некоторые порты не используются, рекомендуется использовать их на фиксированном уровне. В любом случае плавающие штыри будут потреблять больше энергии и будут приводить 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 сопротивление.

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.

9.7. Функции мультиплексирования портов

Большинство портов имеют функции мультиплексирования. Следующая эквивалентная схема на Рис.9.5. иллюстрирует функцию мультиплексирования портов для управления портом. Эти функции мультиплексирования необязательно существуют с выводами порта.

9.7.1. Функция мультиплексирования портов В

Вывод	Альтернативные Функции Описание
PB7	XTAL2 / TOSC2 (внешний основной кристалл контакт 2) PCINT7 (прерывание 7 смены уровня контакта)
PB6	XTAL1 / TOSC1 (внешний основной кристалл контакт 1) PCINT6 (прерывание 6 смены уровня контакта)
PB5	SCK (входной тактовый импульс шины SPI) PCINT5 (прерывание 5 смены уровня контакта)
PB4	MISO (главный входной сигнал ведущего / ведущего устройства шины SPI) PCINT4 (прерывание 4 смены уровня контакта)
PB3	MOSI (главный выход шины SPI / ведомый вход) OC2A (Timer / Counter 2 Compare Match Output A) PCINT3 (прерывание 3 смены уровня контакта)
PB2	SSN (вход для выбора ведомого устройства SPI) OC1B (таймер / счетчик 1 Сравнить совпадение выхода B) PCINT2 (прерывание 2 смены уровня контакта)
PB1	OC1A (Таймер / Счетчик 1 сравнивает выходной результат A) PCINT1 (прерывание 1 смены уровня контакта)
PB0	ICP1 (вход таймера / счетчика 1) CLKO (выход системного тактового сигнала) PCINT0 (прерывание 0 смены уровня контакта)

XTAL2 / TOSC2 / PCINT7 - Порт В Контакт 7

XTAL2: Внешний Вывод 2 кристалла. При использовании в качестве резонатора тактового сигнала этот вывод не может использоваться как I/O.

TOSC2: Таймер внешнего кристалла Вывод 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 7 смены уровня контакта. PB7 является внешним источником прерываний. Если PB7 используется для кристаллов, значения DDB7, PORTB7 и PINB7 не будут иметь никакого значения.

XTAL1 / TOSC1 / PCINT6- Порт В Контакт 6

XTAL1: Внешний Вывод 1 кристалла.

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 генератора. В этом режиме внешний кристалл подключается к контакту.

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

SCK / PCINT5- Порт В Контакт 5

SCK: Сигнал тактов главного контроллера SPI, ведомый вход. Когда контроллер SPI настроен как подчиненное 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 5 смены уровня контакта. PB5 является внешним источником прерываний.

MISO / PCINT4-Port В Контакт 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 4 смены уровня контакта. PB4 является внешним источником прерываний.

MOSI / OC2A / PCINT3- Порт В Контакт 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 Контакт принудительно вводится в управление SPI, его нагрузочный резистор можно контролировать через PORTB3.

OC2A: Группа сравнения результатов сравнения счетчика Таймера / Счетчика 2. PB3 можно использовать в качестве внешнего выходного сигнала сравнения Таймера / Счетчика 2. Вывод должен быть установлен на выход через DDB3. В то же время OC2A также является выходным выводом режима PWM таймера2.

PCINT3: Прерывание 3 смены уровня контакта. PB3 является внешним источником прерываний.

SSN / OC1B / PCINT2- Порт В Контакт 2

SSN: Вывод входа данных SPI Ведущего контроллера. 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 контакт будет принудительно введен контроллером SPI, подтягивающий резистор можно будет контролировать через PORTB2.

OC1B: Сравнение результатов сравнения по счетчику таймера / счетчика 1 группы 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 2 смены уровня контакта. PB2 является внешним источником прерываний.

OC1A / PCINT1 - Порт В Контакт 1

OC1A: Сравнить выходное совпадение группы А счетчика таймера / события 1. 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 1 смены уровня контакта. PB1 является внешним источником прерываний.

ICP1 / CLK0 / PCINT0- Порт В Контакт 0

ICP1: Захват входного вывода таймера / счетчика 1

CLK0: 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 текущей операционной системы.

PCINT0: Прерывание 0 смены уровня контакта. PB0 является внешним источником прерываний.

PB7 ... PB4 мультиплексная логическая таблица управления энергией

Название сигнала	PB7/XTAL2/ TOSC2/PCINT7	PB6/XTAL1/ TOSC1/PCINT6	PB5/SCK PCINT5	PB4/MISO PCINT4
PUOE	OSCEN AS2	OSCEN AS2	SPE&MSTR	SPE&MSTR
PUOV	0	0	PORTB5&PUD	PORTB4&PUD

DDOE	OSCEN AS2	OSCEN AS2	SPE&MSTR	SPE&MSTR
DDOV	0	0	0	0
PVOE	0	0	SPE&MSTR	SPE&MSTR
PVOV	0	0	SCK Выход	SPI Ведомый Выход
DIEOE	PCINT7 Enable	PCINT6 Включить	PCINT5 Включить	PCINT4 Включить
DIEOV	1	1	1	1
DI	PCINT7 Вход	PCINT6 Вход	PCINT5 Вход SCK Вход	PCINT4 Вход SPI Ведущий Вход
AIO	XTAL2 TOSC2	XTAL1 TOSC1	-	-

[Объяснение]: OSCEN включает OSCK_EN и OSCM_EN, см. Описание регистра

PMCR PB3 ... PB0 мультиплексная функция управления логической таблицей

Название сигнала	PB3/MOSI/ OC2A/PCINT3	PB2/SSN/ OC1B/PCINT2	PB1/OC1A/ PCINT1	PB0/ICP1/ CLKO/PCINT0
PUOE	SPE&MSTR	SPE&MSTR	0	0
PUOV	PORTB3&PUD	PORTB2&PLJD	0	0
DDOE	SPE&MSTR	SPE&MSTR	0	CLKO УСТАНОВИТЬ 0
DDOV	0	0	0	1
PVOE	SPE&MSTR+OC2A УСТАНОВИТЬ	OC1B УСТАНОВИТЬ	OC1A УСТАНОВИТЬ	CLKO УСТАНОВИТЬ 0
PVOV	SPI Ведущий Выход OC2A	OC1B	OC1A	CLKO
DIEOE	PCINT3 Включить	PCINT2 Включить	PCINT1 Включить	PCINT0 Включить
DIEOV	1	1	1	1
DI	PCINT3 Вход SPI Ведомый Вход	PCINT2 Вход SPI Ведомый Выбор	PCINT1 Input	PCINT0 Вход ICP1 Вход
AIO	-	-	-	-

9.7.2. Функция мультиплексирования портов C

Вывод	Альтернативные Функции Описание
PC6	RESETN (вход внешнего сброса) PCINT14 (прерывание 14 смены уровня контакта)
PC5	ADC5 (входной канал АЦП 5) SCL (линия синхронизации TWI) PCINT13 (прерывание 13 смены уровня контакта)
PC4	ADC4 (Входной канал ADC 4) SDA (линия передачи данных TWI) PCINT12 (прерывание 12 смены уровня контакта)
PC3	ADC3 (входной канал АЦП 3) PCINT11 (прерывание 11 смены уровня контакта)
PC2	ADC2 (входной канал ADC 2) PCINT10 (прерывание 6 смены уровня контакта10)
PC1	ADC1 (входной канал АЦП 1) PCINT9 (прерывание 9 смены уровня контакта)

RESETN / PCINT4- Порт C Контакт 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 сконфигурировать этот вывод как функцию ввода-вывода и добавить соответствующий нагрузочный резистор извне.

PCINT14: Прерывание 14 смены уровня контакта. После выключения функции внешнего сброса этого вывода, PC6 можно использовать в качестве внешнего источника прерывания.

SCL / ADC5 / PCINT13- Порт C Контакт 5

SCL: Сигнал интерфейса интерфейса TWI. После того, как бит TWEN в регистре TWCR установлен в 1, интерфейс TWI включен и ПК5 управляется 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 13 смены уровня контакта.

SDA / ADC4 / PCINT12- Порт C контакт 4

SDA: сигнал данных интерфейса TWI. После того, как бит TWEN в регистре TWCR установлен, интерфейс TWI включен, а PC4 управляется 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 12 смены уровня контакта.

ADC3 / PCINT11- Порт C Контакт 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.

PCINT11: Прерывание 11 смены уровня контакта.

ADC2 / PCINT1- Порт C Контакт 2

ADC2: Входной канал 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.

PCINT10: Прерывание 10 смены уровня контакта.

ADC1 / PCINT9- Порт С Контакт 1

ADC1: Входной канал 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.

PCINT9: Прерывание 9 смены уровня контакта.

ADC0 / PCINT8- Порт С Контакт 0

ADC0: Входной канал 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.

PCINT8: Прерывание 8 смены уровня контакта.

PC6 ... PC4 мультиплексная логическая таблица управления

Название сигнала	PC6/RESETN/ PCINT14	PC5 /ADC5/TK9/SCL/ PCINT13	PC4/ADC4/TK8/SDA/ PCINT12
PUOE	RSTIOEN	TWI Включить	TWI Включить
PUOV	1	PORTC4&RJD	PORTC4&PUD
DDOE	RSTIOEN	TWI Включить	TWI Включить
DDOV	0	SCL Выход	SDA Выход
PVOE	RSTIOEN	TWI Включить	TWI Включить
PVOV	1	0	0
DIEOE	PCINT14 Включить + RSTIOEN	PCINT13 Включить + TWI Включить	PCINT12 Включить + TWI Включить
DIEOV	1	1	1
DI	PCINT14 Вход Внешний ввод сброса	PCINT13 Вход SCL Вход	PCINT12 Вход SDA Вход
AIO	-	ADC5	ADC4

PC3 ... PC0 мультиплексная логическая таблица управления

Название сигнала	PC3/ADC3/ PCINT11	PC2/ADC2/ PCINT10	PC1/ADC1/ PCINT9	PC0/ADC0/ PCINT8
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	0	0	0
PVOV	0	0	0	0
DIEOE	PCINT11 Включить	PCINT10 Включить	PCINT9 Включить	PCINT8 Включить
DIEOV	1	1	1	1
DI	PCINT11 Вход	PCINT10 Вход	PCINT9 Вход	PCINT8 Вход
AIO	ADC3	ADC2	ADC1	ADC0

9.7.3. Функция мультиплексирования портов D

Вывод	Альтернативные Функции Описание
PD7	AIN1 (отрицательный отрицательный аналоговый компаратор) PCINT23 ((прерывание 23 смены уровня контакта)
PD6	AIN0 (положительный положительный положительный аналоговый компаратор) OC0A (таймер / счетчик 0 сравнивает выходной сигнал A) PCINT22 (прерывание 22 смены уровня контакта)
PD5	T1 (таймер / счетчик 1 внешний счетчик часов) OC0B (таймер / счетчик 0 сравнивает выходной результат B) PCINT21 (прерывание 21 смены уровня контакта)
PD4	XCK (внешний вход / выход внешнего входа USART) T0 (таймер / счетчик 0 внешних счетчиков часов) PCINT20 (прерывание 20 смены уровня контакта)
PD3	INT1 (вход внешнего прерывания 1) OC2B (таймер / счетчик 2 сравнивает выходной результат B) PCINT19 (прерывание 19 смены уровня контакта)
PD2	NT0 (вход внешнего прерывания 0) PCINT18 (прерывание 18 смены уровня контакта)
PD1	TXD (вывод данных USART) PCINT17 (прерывание 17 смены уровня контакта)
PD0	RXD (ввод данных USART) PCINT16 (прерывание 16 смены уровня контакта)

AIN1 / OC2B / PCINT23- Порт D Контакт 7

AN1: Инверсный вход аналогового компаратора. Функция цифрового ввода вывода PD7 отключается регистром DIDR1, а нагрузочный резистор порта отключается, чтобы предотвратить повреждение цифрового порта в аналоговой цепи.

OC2B: Выходные данные сравнения группа В Таймера/Счетчика 2. PD7 можно использовать в качестве внешнего выходного сигнала сравнения Таймера/Счетчика 2. В этом случае контакт должен быть установлен как выход через DDD7. В то же время OC2B также является выходным выводом режима PWM режима Таймера/Счетчика 2.

PCINT23: Прерывание 23 смены уровня контакта.

AIN0 / OC0A / PCINT22- Порт D Контакт 6

AN0: Прямой вход аналогового компаратора. Функция цифрового входа на вывод PD6 отключается регистром DIDR1, а нагрузочный резистор порта отключается, чтобы предотвратить повреждение цифрового порта в аналоговой цепи.

OC0A: Выходные данные сравнения группа А Таймера/Счетчика 0. PD6 можно использовать в качестве внешнего выходного сигнала сравнения Таймера/Счетчика 0. В этом случае контакт должен быть установлен на выход через DDD6. В то же время OC0A также является выходным выводом режима PWM режима Таймера/Счетчика 0.

PCINT22: Прерывание 22 смены уровня контакта.

T1 / OC0B / PCINT21- Порт D Контакт 5

T1: Таймера/Счетчика 1 внешний вход тактов синхронизации

OC0B: Таймера/Счетчика сравнивают выходные данные группы В. PD5 может

использоваться как внешний выходной сигнал сравнения по Таймера/Счетчика 0. Контакт должен быть установлен на выход через DDD5. В то же время OC0B также является выходным выводом режима PWM Таймера/Счетчика 0.

PCINT21: прерывание смены знака 21.

XCK / T0 / PCINT20- Порт D Контакт 4

XCK: Синхронный режим Внешний тактовый сигнал USART

T0: Вход Таймера/счетчика 0 внешний счетчик тактов

PCINT20: Прерывание 20 смены уровня контакта.

INT1 / OC2B / PCINT19- Порт D Контакт 3

INT1: Вход внешнего прерывания 1

OC2B: Вывод сравнение группы В Таймера/Счетчика 2. PD3 может использоваться как внешний выход сравнения Таймера/Счетчика 2. Контакт должен быть установлен на выход через DDD3. В то же время OC2B также является выходным выводом режима PWM режима Таймера/Счетчика 2.

PCINT19: Прерывание 19 смены уровня контакта.

INT0 / PCINT18- Порт D Контакт 2

INT0: Вход внешнего прерывания 0

PCINT18: Прерывание 18 смены уровня контакта.

TXD / PCINT17- Порт D Контакт 1

TXD: Вывод Передачи данных USART. После того, как передатчик USART включен, PD1 будет принудительно выводиться без управления DDD1.

PCINT17: Прерывание 17 смены уровня контакта.

RXD / PCINT16- Порт D Контакт 0

RXD: Вывод Приема данных USART. После того, как приемник USART включен, PD0 будет принудительно введен в качестве входа и не будет управляться DDD0.

После того, как контакт принудительно установлен на вход USART, подтягивающий резистор можно контролировать через бит PORTD0.

PCINT16: Прерывание 16 смены уровня контакта.

PD7 ... PD4 мультиплексная логическая таблица управления:

Название сигнала	PD7/AIN1/ PCINT23	PD6/AIN0/ OC0A/PCINT22	PD5/OC0B/ PCINT21	PD4/XCK/ T0/PCINT20
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	OC0AEN&OC0AS	OC0B Включить	XCKOEN
PVOV	0	OC0A	OC0B	XCK Выход
DIEOE	PCINT23 Включить	PCINT22 Включить	PCINT21 Enable + TIEN	PCINT20 Включить +XCKIEN+T0EN
DIEOV	1	1	1	1

DI	PCINT23 Вход	PCINT22 Вход	PCINT21 Вход T1 Вход	PCINT20 Вход XCK Вход T0 Вход
AIO	-	-	-	-

PD3 ... PD0 мультиплексная логическая таблица управления:

Название сигнала	PD3/OC2B/ INT1/PCINT19	PD2/INT0/ PCINT18	PD1/TXD/ PCINT17	PDO/RXD/ PCINT16
PUOE	0	0	TXEN	RXEN
PUOV	0	0	0	PORTDO&PUD
DDOE	0	0	TXEN	RXEN
DDOV	0	0	1	0
PVOE	OC2B Включить	0	TXEN	0
PVOV	OC2B	0	TXD	0
DIEOE	PCINT19 Включить + INT1 Включить	PCINT18 Включить + INT0 Включить	PCINT17 Включить	PCINT16 Включить + RXEN
DIEOV	1	1	1	1
DI	PCINT19 Вход INT1 Вход	PCINT18 Вход INT0 Вход	PCINT17 Вход	PCINT16 Вход RXD
AIO	-	-		-

9.7.4. Функция мультиплексирования портов E

Вывод	Альтернативные Функции Описание
PE6	VREF (АЦП внешний источник опорного напряжения) PCINT30 (прерывание 30 смены уровня контакта)
PE5	CLKo (выход системного тактового сигнала) PCINT29 (прерывание 29 смены уровня контакта)
PE4	OC0A (Таймер/Счетчик 0 Сравнение Выход A) PCINT28 (прерывание 28 смены уровня контакта)
PE3	ADC7 (входной канал АЦП 7) PCINT27 (прерывание 27 смены уровня контакта)
PE2	SWD (строка данных отладчика SWD) PCINT26 (прерывание 26 смены уровня контакта)
PE1	ADC6 (входной канал АЦП 6) PCINT25 (прерывание 25 смены уровня контакта)
PE0	SWC (вход синхронизации отладчика SWD) PCINT24 (прерывание 24 смены уровня контакта)

VREF / PCINT30- Порт E Контакт 6

VREF: внешний входной сигнал источника питания АЦП. При использовании в качестве аналоговой функции необходимо установить соответствующий цифровой ввод-вывод в качестве входного сигнала и выключить нагрузочный резистор, чтобы предотвратить нарушение цифровой схемы аналоговой схемой.

PCINT30: Прерывание 30 смены уровня контакта.

CLKO / PCINT29- Порт E Контакт 5

CLKO: Эта функция аналогична функции CLKO PB0. Может использоваться как резервный Контакт для PB0 / CLKO.

PCINT29: Прерывание 29 смены уровня контакта.

OC0A / PCINT28- Порт E Контакт 4

OC0A: Выхода группы А счетчика Таймера/Счетчика 0. PE4 может использоваться в качестве, выхода сравнения Таймера/Счетчика 0. В этом случае Контакт должен быть установлен как выход через DDE4. В то же время OC0A также является выходным выводом режима PWM режима Таймера/Счетчика 0.

PCINT28: Прерывание 28 смены уровня контакта.

ADC7 / PCINT27- Порт E Контакт 3

ADC7: Входной канал АЦП 7. Регистр DIDR используется для отключения цифровой функции цифроаналогового мультиплексированного ввода-вывода, чтобы избежать влияния цифровой части на аналоговую схему. Подробнее см. В соответствующем разделе АЦП.

PCINT27: Прерывание 27 смены уровня контакта.

SWD / PCINT26- Порт E Контакт 2

SWD: Линия данных отладчика SWD. После сброса при включении PE2 по умолчанию переключается на функцию SWD. Пользователь может отключить функцию отладчика SWD, установив бит SWDD регистра MCUSR. Функция отладки не будет доступна после закрытия SWD.

PCINT26: Прерывание 26 смены уровня контакта.

ADC6 / PCINT25- Порт E Контакт 1

ADC6: Входной канал АЦП 6. 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.

PCINT25: Прерывание 25 смены уровня контакта.

SWC / PCINT24- Порт E Контакт 0

SWC: Линия таймера отладки SWD. После сброса при включении PE0 по умолчанию выполняет функцию SWC. Пользователь может отключить функцию отладчика SWD, установив бит SWDD регистра MCUSR. Функция отладки не будет доступна после закрытия SWD.

PCINT24: Прерывание 24 смены уровня контакта.

PE6 ... PE4 мультиплексная логическая таблица управления

Название сигнала	PE6/VREF/ PCINT30	PE5/CLKO/ PCINT29	PE4/OC0A/ PCINT28
PUOE	REFIOEN	0	0
PUOV	0	0	0
DDOE	REFIOEN	CLKO Установить 1	0
DDOV	0	1	0
PVOE	REFIOEN	CLKO Установить 1	OC0AEN&OC0AS
PVOV	1	CLKO	OC0A

PIND - Регистр входных данных порта D

PIND - Регистр входных данных порта D								
PIND: 0x09(0x29)		Значение по умолчанию: 0x00						
Биты	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Определение бит								
[7:0]	PIND	Регистр состояния порта D. Считывается PIND напрямую, чтобы получить текущее состояние порта; запись в бит PINDn «1» переворачивает состояние вывода PORTDn						

PORTE - Регистр выходных данных порта E

PORTE - Регистр выходных данных порта E								
PORTE: 0x08(0x28)		Значение по умолчанию: 0x00						
Биты	-	PE6	PE5	PE4	PE3	PE2	PE1	PE0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Определение бит								
[7]	-	Зарезервировано						
[6:0]	PORTE	Регистр выхода порта E						

DDRE - Регистр направления E порта

DDRE - Регистр направления E порта								
DDRE: 0x07(0x27)		Значение по умолчанию: 0x00						
Биты	-	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Определение бит								
[7]	-	Зарезервировано						
[6:0]	DDRE	Биты управления направлением порта E: 1 = выход, 0 = вход						

PINE - Регистр входных данных порта E

PINE - Регистр входных данных порта E								
PINE: 0x06(0x26)		Значение по умолчанию: 0x00						
Биты	-	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Определение бит								
[7]	-	Зарезервировано						
[6:0]	PINE	Регистр состояния порта E. Считывается PINE напрямую, чтобы получить текущее состояние порта; запись в бит PINEn «1» переворачивает состояние вывода PORTEn						

10. Прерывание изменения уровня сигнала

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

10.1. Обзор

Прерывание смены контактов инициируется выводами PBn, PCn, PDn и PEn.

PCICR - регистр управления прерыванием изменения уровня контактов								
Адрес 0x68			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	-	-	-	-	PCIE3	PCIE2	PCIE1	PCIE0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:4	-	Зарезервировано						
3	PCIE3	Когда бит PCIE3 установлен в «1» и глобальное прерывание включено, прерывание 3 переключения контактов активировано. Изменение уровня любого из активированного PEn-контактов приведет к прерыванию PCI3. Разрешение прерывания вывода PEn может управляться регистром PCMSK3, соответственно. Когда бит PCIE3 установлен в «0», прерывание 3 смены уровня контактов отключено.						
2	PCIE2	Когда бит PCIE2 установлен в «1» и глобальное прерывание включено, прерывание 2 переключения контактов активировано. Изменение уровня любого из активированного вывода PDn приведет к прерыванию PCI2. Разрешение прерывания вывода PDn может управляться регистром PCMSK2, соответственно. Когда бит PCIE2 установлен на «0», прерывание 2 смены уровня контактов отключено.						
1	PCIE1	Когда бит PCIE1 установлен на «1» и глобальное прерывание включено, прерывание 1 переключения контактов активировано. Изменение уровня любого из активированного вывода PCn приведет к прерыванию PCI1.						

7	PCINT7	Бит маски включения бит - 7.
6	PCINT6	Бит маски включения бит - 6
5	PCINT5	Бит маски включения бит - 5
4	PCINT4	Бит маски включения бит -4
3	PCINT3	Бит маски включения бит - 3.
2	PCINT2	Бит маски включения бит - 2.
1	PCINT1	Бит маски включения бит - 1.
0	PCINT0	Бит маски включения бит - 0.
Пояснение		
<p>При установке бит PCINTn в «1», прерывание изменения уровня сигнала PBn включено. Изменение уровня на выводах PBn будет устанавливать PCIF0. Если бит PCIE0 и глобальное прерывание установлены, будет генерироваться прерывание PCIF0. При установке бит PCINTn в «0» прерывание изменения уровня сигнала PBn отключено. Бит n = 7...0</p>		

PCMSK1 - регистр 1 прерывания изменения уровня контактов

PCMSK2 - регистр 2 прерывания изменения уровня контактов

2	PCINT18	Бит маски включения бит - 10.
1	PCINT17	Бит маски включения бит - 9.
0	PCINT16	Бит маски включения бит - 8.
Пояснение		
При установке бит PCINT _{n+16} в «1», прерывание изменения уровня сигнала PD _n включено. Изменение уровня на выводах PD _n будет устанавливать PCIF2. Если бит PCIE2 и глобальное прерывание установлены, будет генерироваться прерывание PCI2. При установке бит PCINT _{n+16} в «0» прерывание изменения уровня сигнала PD _n отключено. Бит n = 7...0		

PCMSK3 - регистр 3 прерывания изменения уровня контактов

PCMSK3 - регистр 3 прерывания изменения уровня контактов								
Адрес 0x6B			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	-	PCINT30	PCINT29	PCINT28	PCINT27	PCINT26	PCINT25	PCINT24
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7	-	Зарезервировано						
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...0								

11. Таймер/Счетчик 0 - 8-разрядный

- 8-разрядный счетчик
- Два независимых блока сравнения
- Автоматическая очистка счетчика и загрузка при равенстве сравнения
- С Фазовой коррекцией ШИМ-выход без импульсных помех
- Генератор частоты
- Счетчик внешних событий
- 10-разрядный предварительный делитель тактов
- Прерывание при переполнении и сравнениях совпадений
- Управление мертвым временем
- 8 дополнительных источников запуска для автоматического выключения выхода PWM
- Высокоскоростное и высокое разрешение (500KHz @ 7Bit) PWM в режиме высоких тактовых импульсов

11.1. Обзор

TC0 - это 8-разрядный счетчик 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 предварительный делитель независимо. Предделитель делит системные такты clk_{io} или высокоскоростные такты $gsm2x$ (2x внутреннего выходного сигнала генератора 32M RC генератора $gsc32m$) для генерации счетчика часов $Clkt0$.

Блок управления режимом генерации волны управляет режимом работы счетчика и формированием сравнительного выходного сигнала. В соответствии с различными режимами работы счетчик осуществляет очистку, увеличение или уменьшение каждого из счетчиков тактов $Clkt0$.

$Clkt0$ может генерироваться внутренним источником синхронизации или внешним источником синхронизации.

Когда значение счета 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 $OCR0A / OCR0B$, которое также вызывается, когда происходит сравнение, выходной сигнал сравнения $OC0A / OC0B$ очищается или устанавливается для генерации сигнала ШИМ.

Когда мертвое время включено, установленное мертвое время (количество тактом 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 запуска. Когда происходит событие запуска, аппаратное обеспечение автоматически очищает бит $COM0A / COM0B$, чтобы отключить выходной сигнал $OC0A / OC0B$.

Счетчик тактов 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, управляется битами $WGM0$, расположенными в регистрах $TCCR0A$ и $TCCR0B$.

В соответствии с различными режимами работы счетчик осуществляет очистку, увеличение или уменьшение каждого из счетчиков 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, формируется сигнал сравнения и устанавливается флаг сравнения OCF0A или OCF0B вывода в регистре TIFR0. После 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 CTC. Когда счетчик достигает максимального или минимального значения, значения в регистре буфера обновляются синхронно с регистрами сравнения OCR0A и OCR0B. Подробнее см. Описание раздела режима работы.

Генератор сигналов генерирует выходные сигналы OC0A и OC0B 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 Рис.11.1. 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.

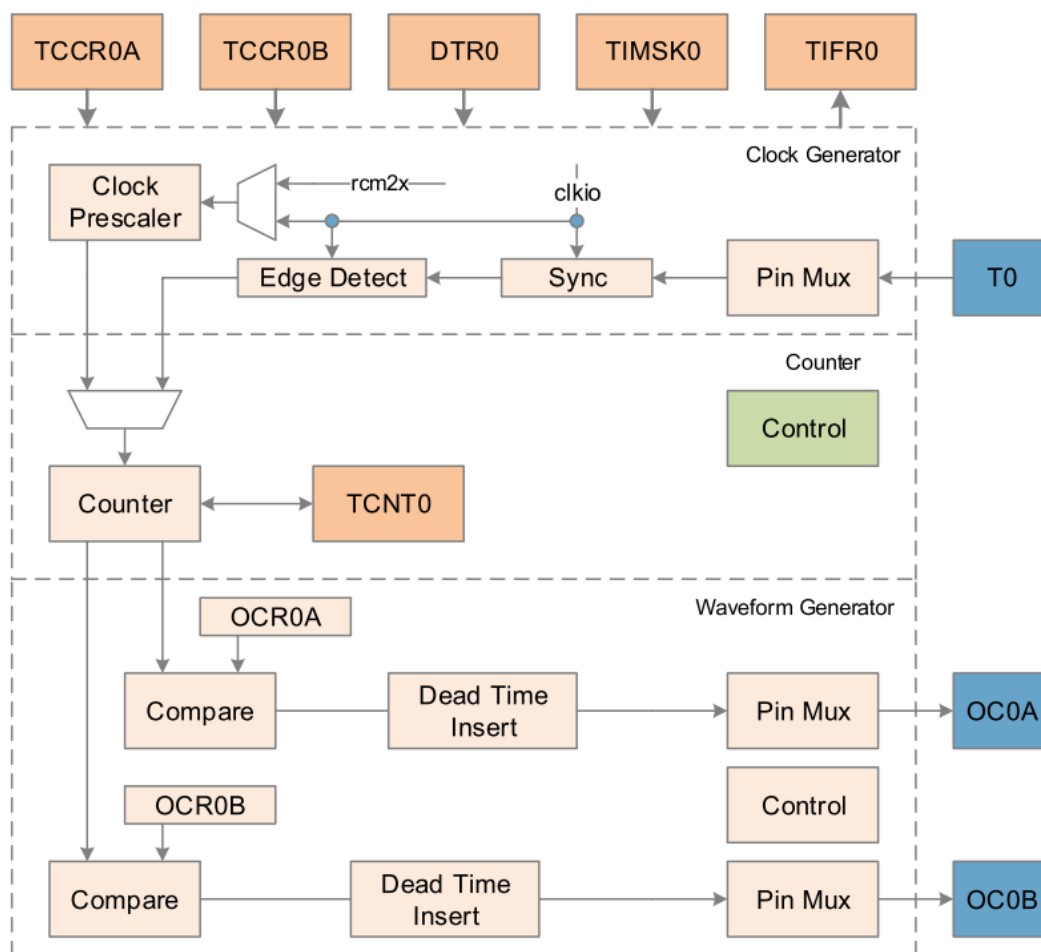


Рис.11 .1. Структурная схема TC0

11.2. Режим работы

Счетчик таймера 0 имеет четыре различных режима работы, включая нормальный режим, режим сравнения (СТС), режим быстрой широтно-импульсной модуляции (FPWM) и режим импульсной модуляции фазовой коррекции (PCPWM). Биты WGM0 [2: 0] для выбора управления режимом. Ниже описаны эти четыре режима. Так как есть два независимых блока сравнения вывода, обозначенные соответственно «А» и «В», нижний вывод «х» представляет собой два выходных канала сравнения.

11.2.1. Нормальный режим

Обычный режим - это самый простой режим работы счетчика таймера.

В это время биты управления режимом генерации сигнала WGM0 [2: 0] = 0, а максимальное значение счетчика TOP - MAX (0xFF).

В этом режиме счетчик увеличивается на один такт счета счетчика.

Когда счетчик достигает переполнения TOP, он сбрасывается в BOTTOM, чтобы снова начать накапливать.

Флаг переполнения счетчика таймера TOV0 устанавливается в тот же счет такта, когда значение счетчика TCNT0 становится равным нулю. Флаг TOV0 в этом режиме похож на 9-й бит счетчика, он устанавливается, но не очищается.

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

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

Форма выходного сигнала сравнения OC0x может быть получена только тогда, когда регистр направления данных вывода OC0x задан как выход.

Когда COM0x = 1, сигнал OC0x будет переключается при равенстве сравнения.

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

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

где N - Коэффициент делителя (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 0 таймера переходит в режим СТС, а максимальное значение счетчика - TOP OCR0A.

В этом режиме счетчик увеличивается на единицу для каждого счета такта.

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, устанавливает

флаг соответствия OCF0 на выходе, и прерывание будет генерироваться, когда установлен соответствующий бит разрешения прерывания.

Регистр OCR0A, который является максимальным числом, может быть обновлен в процедуре обслуживания прерываний. В этом режиме OCR0A не использует двойную буферизацию, и следует соблюдать осторожность при обновлении максимума вблизи минимального значения, когда счетчик работает без предварительного делителя или очень низкого предварительного делителя.

Если значение, записанное в OCR0A, меньше текущего значения TCNT0, счетчик пропустит равенство сравнения. Перед тем, как произойдет следующее сравнение, счетчик должен посчитывать до TOP, а затем посчитать с BOTTOM до OCR0A.

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

Форма 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 совпадении сравнения.

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

$$f_{oc0xctc} = f_{sys} / (2 * N * (1 + OCR0x))$$

N - Коэффициент предделителя (1, 8, 64, 256 или 1024). Из формулы видно, что когда OCR0A задано равным 0x0000, 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 добавляется в TOP от минимального значения 0x00, а затем возвращается 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 COM0x. Подробнее см. Описание регистра.

Рабочая частота, счетчика в режиме быстрой PWM, в два раза выше, чем двунаправленная операция в режиме PWM с коррекцией фаз.

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, будет установлен флаг переполнения счетчика таймера TOV0, а значение буфера сравнения будет обновлено 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:

$$f_{oc0x\text{pwm}} = f_{\text{sys}} / (N * (1 + \text{TOP}))$$

N - Коэффициент делителя (1, 8, 64, 256 или 1024).

Когда происходит равенство 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 двунаправленным тактом (увеличение и уменьшение) от BOTTOM до TOP, а затем вниз до BOTTOM. Эта операция повторяется.

Когда счетчик достигает TOP и BOTTOM, оба изменяют направление подсчета, а значение счетчика остается только на одном такте TOP счетчика 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 однонаправленной работой максимальная частота, доступная для двунаправленной работы, мала, но ее отличная симметрия более подходит для управления двигателем.

Фазовая коррекция в режиме PWM флаг TOV0 устанавливается, когда счетчик достигает BOTTOM, а значение буфера 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:

$$f_{oc0x\text{pcpwm}} = f_{\text{sys}} / (N * \text{TOP} * 2)$$

N - Коэффициент делителя (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 счетчика 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, а значение OC0x сохраняется до тех пор, пока не произойдет совпадение сравнения, и произойдет переворот. В этом случае сигнал OC0x не является симметричным относительно минимального значения, поэтому сигнал OC0x необходимо переключать, когда TCNT0 достигает максимального значения. То есть первый случай опрокидывания сигнала OC0x во время сравнения не возникает.

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. Контроль Мертвого Времени

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 Рис.11 .2. и Рис.11 .3., временные вставки OC0A и OC0B основаны на выходной форме сигнала канала B. Когда COM0A и COM0B являются «2» или «3», полярность сигнала OC0A такая же, как полярность формы сигнала OC0B. Когда COM0A и COM0B «2» или «3», форма OC0A и форма OC0B соответственно. Противоположная полярность.

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

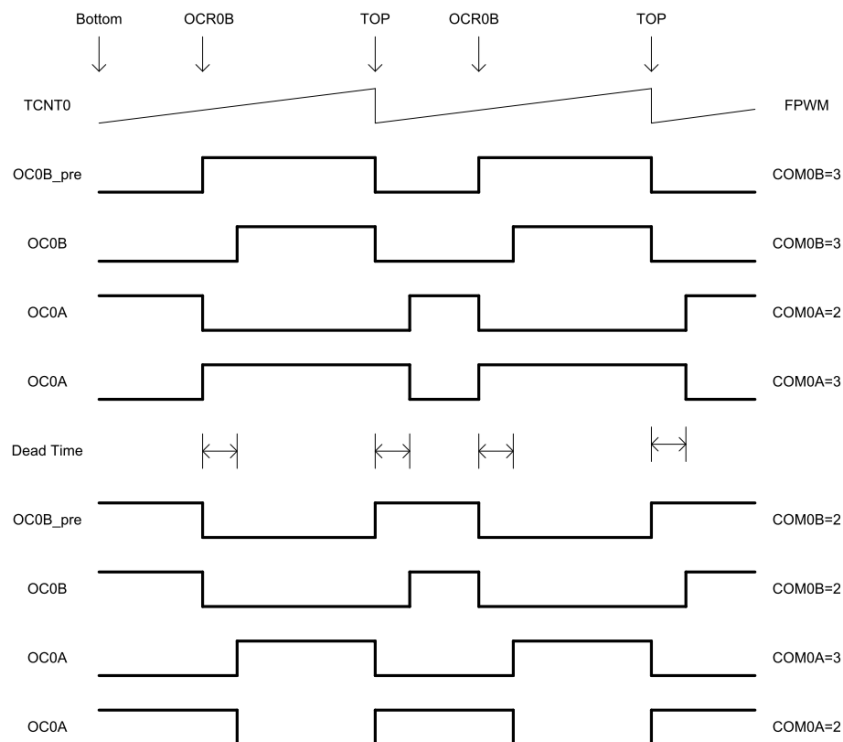


Рис.11 .2. TC0 Диаграмма мертвого времени в режиме FPWM

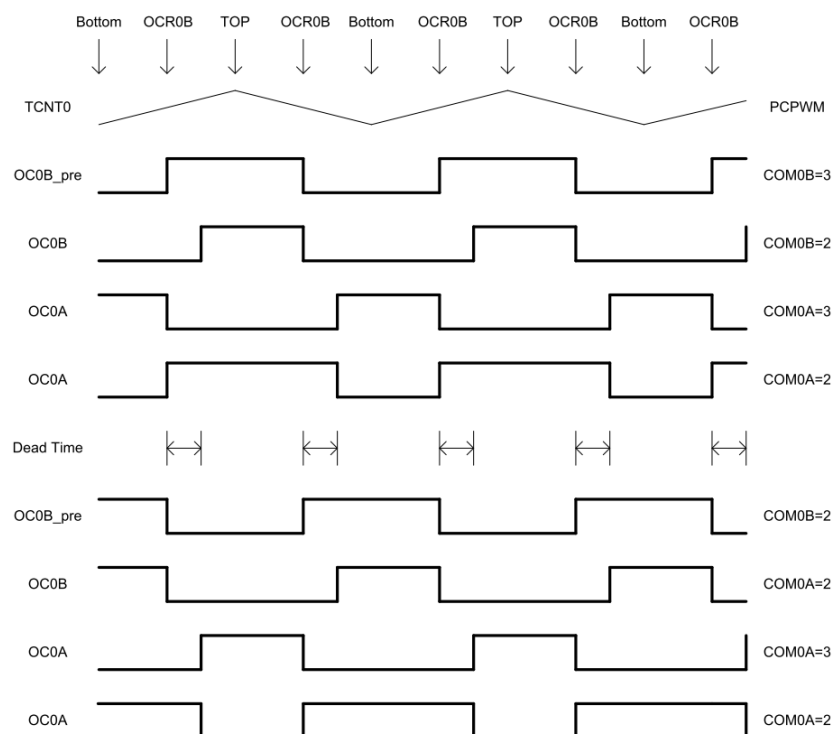


Рис.11 .3. TC0 Диаграмма мертвого времени в режиме PCPWM

11.2.6. Режим высокоскоростной синхронизации.

В высокоскоростном тактовом режиме в качестве источника синхронизации используется счетчик высоких частот для подсчета для генерации высокочастотных и высокочастотных сигналов PWM.

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

Поэтому перед входом в высокочастотный режим необходимо включить функцию частотного умножения внутреннего 32М 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 тактов. В этом режиме системные такты и высокоскоростные такты асинхронны, а некоторые регистры (см. Список регистров TC0) работают в высокоскоростном домене так, что 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 (задержка в режиме NOP или тактов операционной системы);
- 3) Чтение или запись регистров A или B.
- 4) Дождитесь системных тактов (задержка 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). Прочитайте значение TCNT0.

11.2. Регистры Описания

Список регистров TC0

Регистр	Адрес	Значение по умолчанию	Описание
TCCR0A*	0x44	0x00	Контрольный регистр TC0 A
TCCR0B*	0x45	0x00	Контрольный регистр TC0 B
TCNT0*	0x46	0x00	Регистр значений счетчика TC0
OCR0A*	0x47	0x00	Регистр сравнения выходных данных TC0 A
OCR0B*	0x48	0x00	Регистр сравнения выходных данных TC0 B
DSX0*	0x49	0x00	Регистр управления источником Запуска
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 Регистр состояния и контроля тактов

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

TCCR0A - Регистр управления TC0 A

TCCR0A - Регистр управления TC0 A								
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

		генерации волны.	
2	CS02	Предделение тактов источника синхронизации для счетчика времени 0.	
1	CS01		
0	CS00		
</			

COM0n - Нормальный режим и режим CTC Управление сравнением

COM0x1	COM0x0	Описание
0	0	Нормальных операций порта, OC0x отключены.
0	1	Сигнал Переключить OC0x при равенстве
1	0	Очистить сигнал OC0x при Сравнении
1	1	Установить сигнал OC0x на Сравнении

COM0n - Быстрый Режим PWM Управление сравнением

COM0x1	COM0x0	Описание
0	0	Нормальных операций порта, OC0x отключены.
0	1	Зарезервировано
1	0	Очистить сигнал OC0x при Сравнении, и установить сигнал OC0x в BOTTOM
1	1	Установить OC0x при Сравнении, Очистить OC0x в BOTTOM (инвертирующий режим)

COM0n - Режим PWM с коррекцией фаз Управление сравнением

COM0x1	COM0x0	Описание
0	0	Нормальных операций порта, OC0A отключены.
0	1	Зарезервировано
1	0	Очистить OC0x на Сравнении Компаратора когда Счет Вверх. Установленный OC0x на Сравнении Компаратора когда Счет Вниз.
1	1	Установленный OC0x на Сравнении Компаратора когда Счет Вверх. Очистить OC0x на Сравнении Компаратора когда Счет Вниз.

WGM0 - Управление режимом генерации сигнала.

WGM0 [2:0]	Режим работы	Значение TOP	Обновить момент OCR0X	Установите момент TOV0
0	Нормальный	0xFF	Немедленно	MAX
1	PCPWM	0xFF	TOP	BOTTOM
2	CTC	OCR0A	Немедленно	MAX
3	FPWM	0xFF	TOP	MAX
4	Зарезервировано	-	-	-
5	PCPWM	OCR0A	TOP	BOTTOM
6	Зарезервировано	-	-	-
7	FPWM	OCR0A	TOP	TOP

Полярности формы сигнала OC0A в режиме включения мертвого времени

DTEN0	COM0A [1:0]	COM0B [1:0]	Описание
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 OC0B управляется режимом вывода сравнения OC0B, который аналогичен тому, когда режим мертвого времени не включен.

DSX0- Регистр управления TC0 Управления источниками запуска

DSX0- Регистр управления TC0 C								
Адрес: 0x49			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	DSX07	DSX06	DSX05	DSX04	DSX03	DSX02	DSX01	DSX00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7	DSX07	При установке бита DSX07 в «1», переполнение TC1 активируется в качестве источника запуска для выключения сигнала сравнения OC0A/OC0B.						
6	DSX06	При установке бита DSX06 в «1» переполнение TC2 активируется в качестве источника запуска для выключения сигнала сравнения OC0A/OC0B.						
5	DSX05	При установке бита DSX05 на «1» уровень изменения Вывода 0 включается в качестве источника запуска для выключения выходной формы сигнала сравнения OC0A / OC0B.						
4	DSX04	Когда бит DSX04 установлен в «1», внешнее Прерывание 0 включается в качестве источника запуска для выключения выходной сигнала сравнения OC0A/OC0B.						
3	DSX03	Когда бит DSX03 установлен в «1», Аналоговый компаратор 1 Канал 1 включен как источник запуска для выключения выходной сигнала сравнения OC0A/OC0B						
2	DSX02	Когда бит DSX03 установлен в «1», Аналоговый компаратор 1 Канал 0 включен как источник запуска для выключения выходной сигнала сравнения OC0A/OC0B						
1	DSX01	Когда бит DSX01 установлен в «1», Аналоговый компаратор 0 Канал 1 включен как источник запуска для выключения выходной сигнала сравнения OC0A/OC0B						
0	DSX00	Когда бит DSX01 установлен в «1», Аналоговый компаратор 0 Канал 0 включен как источник запуска для выключения выходной сигнала сравнения OC0A/OC0B						

OC0A / OC0B - Отключить управление Выбор источника

DOC0x	DSX0n=1	Источник	Описание
0	-	-	Бит DOC0x = «0», функция запуска выключения источника выходного сигнала отключена.
1	0	Аналоговый	Передний фронт ACIF00 отключает выходной

TIFR0 - Регистр флагов прерывания TCO

TIFR0 - Регистр флагов прерывания TC0								
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	Чтение сигнала осциллограммы OC0A. Выходной сигнал осциллограммы OC0A, считывается программным обеспечением, но не записывается. 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 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	Чтение сигнал осциллограммы OC0B. Выходной сигнал осциллограммы OC0B, считывается программным обеспечением, но не записывается. 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 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 = «1» и флаг глобального прерывания в «1», будет генерироваться прерывание равенства 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 = «1» и флаг глобального прерывания в «1», 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 TCO

DTR0 - регистр контроля мертвого времени TC0								
Адрес: 0x4F			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	DTR07	DTR06	DTR05	DTR04	DTR03	DTR02	DTR01	DTR00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						

7:0	DTR0H	TC0 мертвый регистр времени. Когда бит DTEN0 регистра TCCR0B равен «1», управление мертвым временем включения включено, а установленное мертвое время определяется DTR0. Длительность времени - это время, соответствующее тактам счетчика DTR0.
-----	-------	--

ТСККСР - TC Регистр состояния и контроля тактов

ТСККСР - TC Контроль тактов и регистр состояния								
Адрес: 0xEC			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	-	F2XEN	TC2XF1	TC2XF0	-	AFCKS	TC2XS1	TC2XS0
R/W	-	R/W	R	R		R/W	R/W	R/W
Бит	Имя	Определение бит						
7	-	Зарезервировано						
6	F2XEN	Контрольный бит разрешения выходного сигнала RC 32М. Когда бит F2XEN в «1», выходной сигнал RC-генератора с частотой 32М и умножения на 2 включен, выводятся высокоскоростные такты 64М. Когда бит F2XEN установлен в «0», частотный умножитель 32М RC-генератора отключается, а высокоскоростные такты 64М не могут быть выведены.						
5	TC2XF1	Флаг режима высокоскоростной синхронизации TC1. Когда бит TC2XF1 считывается как «1», он указывает, что счетчик таймера 1 работает в режиме высокоскоростной синхронизации. Когда он равен «0», это означает, что счетчик 1 таймера работает в режиме системных тактов.						
4	TC2XF0	Флаг режима высокоскоростной синхронизации TC0. Когда бит TC2XF0 считывается как «1», это он указывает, что счетчик 0 таймера работает в режиме высокоскоростной синхронизации. Когда он равен «0», это означает, что счетчик 0 таймера работает в режиме системных тактов.						
3	-	Зарезервировано						
2	AFCKS	Выбор тактов фильтра OP/AC, пожалуйста, обратитесь к главе OP/AC для подробного определения						
1	TC2XS1	Бит управления скоростью высокоскоростного тактового генератора TC 1. При установке бита TC2XS1 в «1» счетчик/таймера 1 работает в режиме высокоскоростной синхронизации. Когда бит TC2XS1 установлен в «0», счетчик/таймера 1 работает в режиме системных тактов. Чтобы установить TC2XS[1: 0], сначала записать регистр ECCPR в 0xD8, а затем записать значение для записи в бит TC2XS в течение 6 системных тактов						
0	TC2XS0	Бит управления скоростью высокоскоростного тактового генератора TC0. При установке бита TC2XS0 в «1» счетчик/таймера 0 работает в режиме высокоскоростной синхронизации. Когда бит TC2XS0 установлен в «0», счетчик/таймера 0 работает в режиме системных тактов. Чтобы установить TC2XS[1: 0], сначала записать регистр ECCPR в 0xD8, а затем записать значение для записи в бит TC2XS в течение 6 системных тактов						

12. Таймер/счетчик 1 - 16-разрядный

- Полная 16-разрядная конструкция, позволяющая использовать 16-разрядный PWM
- 2 независимых блока сравнения выходов
- Двойной буферный регистр сравнения результатов
- 1 блок захвата ввода
- Подавитель шума входного захвата

- Автоматическое сравнение соответствия
- Сброс счетчика и загрузите автоматически
- Исправлена фаза PWM без возмущающих импульсов
- Переменный период PWM
- Генератор частоты
- Счетчик внешних событий
- Четыре независимых источника прерываний
- ШИМ с тайм-контролем
- Четыре дополнительных запуска Источник с автоматическим выводом ШИМ-выход
- Высокоскоростной (500KHZ @ 7BIT) PWM в режиме высокоскоростной синхронизации

12.1. Обзор

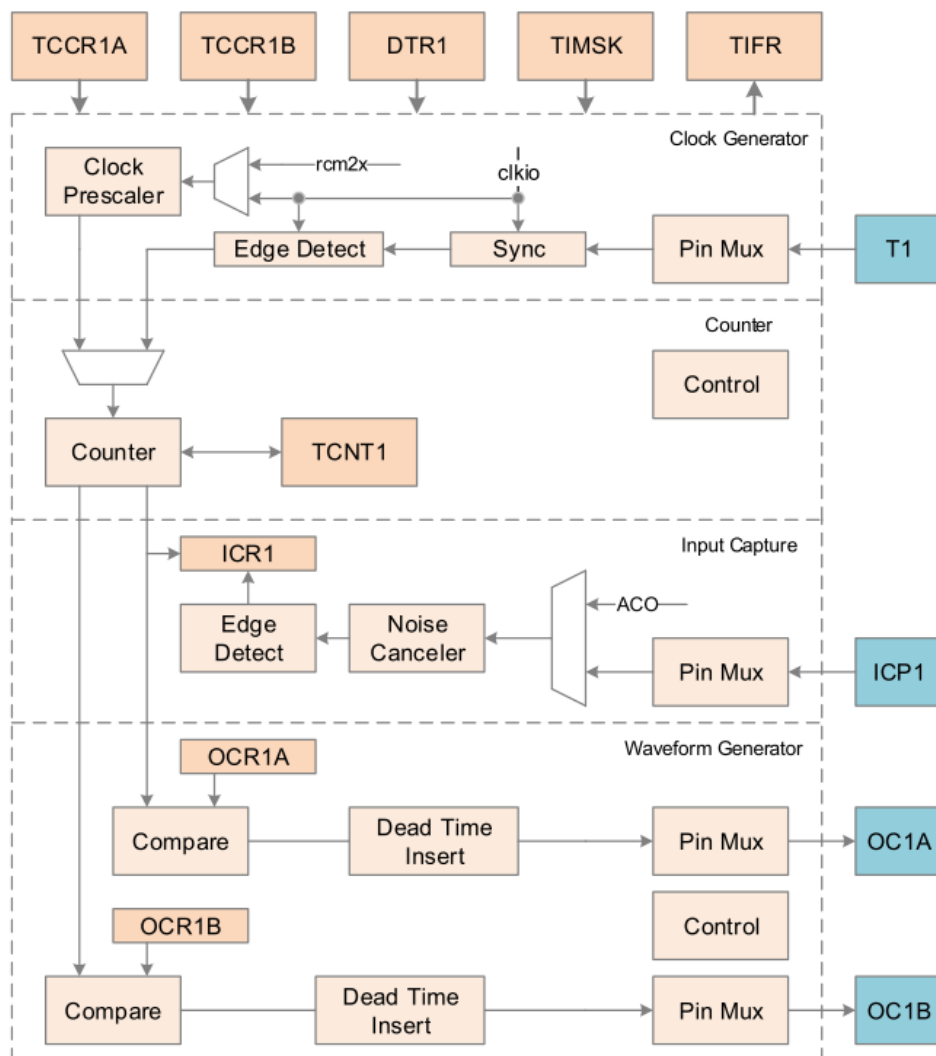


Рис. .1. Структурная схема TC1

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

В то же время TC1 может совместно использовать 10-битный

предварительный делитель с TC0 или он может независимо использовать 10-разрядный предварительный делитель.

Предделитель делит системные такты clk_{io} или высокоскоростные такты $rcm2x$ (2х внутреннего выходного сигнала генератора 32М RC генератора $rc32m$) для генерации счетчика часов $Clkt1$.

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 из счета тактов $Clkt1$. $Clkt1$ может быть сгенерирован из внутреннего источника синхронизации или внешнего источника синхронизации.

Когда значение счетчика TCNT1 счетчика достигает максимального значения (равное максимальному значению $0xFFFF$ или фиксированному значению, или регистру сравнения OCR1A, или регистру регистра ввода ICR1, определяемому как TOP, максимальное значение определяется как MAX для отличия), счетчик очищается или вычитает далее.

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

Когда значение счетчика TCNT1 счетчика достигает равенства с OCR1A или OCR1B, формируется 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 сформированный сигнал PWM.

Когда функция захвата входа активирована, счетчик запускает или прекращает подсчет при срабатывании, а регистр ICR1 регистрирует значение счета во время периода срабатывания сигнала захвата.

Программное обеспечение может очистить выходной сигнал OC1A / OC1B, очистив бит COM1A/COM1B до нуля или установить соответствующий источник запуска.

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

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

Счетчик имеет длину 16 бит и поддерживает двунаправленный подсчет. Режим генерации сигнала, который является режимом 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 часов $Clkt1$.

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, создается совпадение сравнения 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 CTC.

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

Генератор сигналов генерирует выходные сигналы OC1A и OC1B сигналов осциллограммы вывода, используя совпадение сравнения и переполнение счетчика в соответствии с режимом режима 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, 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 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 счетчика TOP - 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 тот же счет такта, когда значение счетчика TCNT1 становится равным нулю. Флаг TOV1 в этом режиме похож на 17-й бит, но он будет установлен и не будет очищен.

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 представляет собой 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 CTC

Когда установлено $WGM1 [3: 0] = 4$ или 12, счетчик 1 таймера переходит в режим CTC.

Когда $WGM1 [3] = 0$, максимальное значение TOP TOP - это OCR1A.

Когда $WGM1 [3] = 1$, максимальное значение счета TOP является ICR1.

Ниже описывается режим CTC с использованием $WGM1 [3: 0] = 4$. В этом режиме режим подсчета увеличивается для каждого счетчика. 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, и следует соблюдать осторожность при обновлении максимального минимального значения, когда счетчик работает без предварительного делителя или очень низкого предварительного делителя.

Если значение, записанное в OCR1A, меньше текущего значения TCNT1, счетчик пропустит сравнение. Перед тем, как произойдет следующее сравнение, счетчик должен рассчитывать до MAX перед подсчетом 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 - 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. Режим быстрый PWM (FPWM)

Когда установлен на WGM1 [3: 0] = 5, 6, 7, 14 или 15, счетчик 1 таймера переходит в режим быстрой PWM, а максимальный счетчик TOP - 0xFF, 0x1FF, 0x3FF, ICR1 или OCR1A, соответственно, который может использоваться для генерации Высокочастотный ШИМ-сигнал.

Разница между быстрым режимом PWM и другими режимами PWM заключается в том, что это односторонняя операция.

Счетчик увеличивается с BOTTOM на TOP и обратно до BOTTOM для повторного подсчета. Когда значение счетчика TCNT1 достигает TOP или BOTTOM, выходной сигнал сравнения OC1x устанавливается или очищается в зависимости от установки режима вывода сравнения COM1. Подробности см. В описании регистра.

Из-за односторонней операции рабочая частота режима быстрой PWM в два раза больше, чем в режиме PWM с фазовой коррекцией, используя двунаправленную операцию.

Высокочастотные характеристики делают быстрый режим PWM подходящим для регулирования мощности, выпрямления и применения ЦАП. Высокочастотные сигналы могут уменьшить размер внешних компонентов (емкость индуктора и т. Д.), Что снижает стоимость системы.

Когда значение счетчика достигнет TOP, будет установлен флаг переполнения счетчика таймера TOV1, а значение буфера сравнения будет обновлено до значения сравнения.

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 - Коэффициент делителя (1, 8, 64, 256 или 1024).

Когда встречается сравнение между TCNT1 и OCR1x, генератор формы сигнала устанавливает (очищает) сигнал OC1x.

Когда TCNT1 очищается, генератор формы волны очищает (устанавливает) сигнал OC1x для генерации волны ШИМ. В результате экстремальное значение OCR1x генерирует специальную форму ШИМ.

Когда OCR1x установлен в 0x0000, выходной PWM имеет узкий импульс в каждом (1 + TOP) счетчике часов.

Когда OCR1x установлен в TOP, выходной сигнал постоянно высокий или низкий.

Если OCR1A используется в качестве TOP и COM1A = 1, выходной сигнал сравнения OC1A будет генерировать PWM-сигнал с 50% -ным рабочим циклом.

12.2.4. Режим PWM с Фазовой коррекцией (PCPWM)

Когда установлено значение WGM0 [3: 0] = 1, 2, 3, 10 или 11, счетчик 1

таймера входит в режим PWM с фазовой коррекцией, а максимальное значение 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 и BOTTOM, оба изменяют направление счета, а значение счетчика остается только на одном счетчике часов TOP или BOTTOM. Во время процесса приращения или уменьшения, когда значение счетчика TCNT1 совпадает с OCR1x, выходной сигнал сравнения OC1x будет очищен или установлен в зависимости от установки режима вывода COM сравнения.

По сравнению с односторонней работой максимальная частота, получаемая двунаправленной операцией, меньше, но ее превосходная симметрия более подходит для управления двигателем.

В режиме PWM с фазовой коррекцией флаг TOV1 устанавливается, когда счетчик достигает BOTTOM, а буфер сравнения обновляется до значения сравнения, когда счетчик достигает TOP.

Если прерывание включено, регистр OCR1x буфера сравнения может быть обновлен в подпрограмме обслуживания прерываний.

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

Частоту формы волны можно вычислить по следующей формуле:

$$f_{oc1xcrpwm} = f_{sys} / (N * TOP * 2)$$

где N - Коэффициент делителя (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 счетчика 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 подсчитывается счетчик возрастания. 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 Фазовой коррекцией частоты

Когда установлено значение WGM0 [3: 0] = 8 или 9, счетчик /таймер 1 входит в режим PWM с коррекцией фазовой частоты, а максимальное значение счета TOP - это ICR1 или OCR1A соответственно. Счетчик использует двустороннюю операцию, которая увеличивается от BOTTOM до TOP, а затем уменьшается 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 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 двунаправленной работой, меньше, но ее отличная симметрия более подходит для управления двигателем.

В режиме PWM с Коррекцией фазовой частоты 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 режимом PWM с коррекцией фазовой 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 ORC1A 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 тогда, когда регистр направления данных вывода OC1x задан как выход.

Частоту формы волны можно вычислить по следующей формуле:

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

N - Коэффициент делителя (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.

Поскольку регистр OCR1х обновляется на время BOTTOM, длина отсчета для восходящей и нисходящей сторон значения TOP одинакова, что приводит 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 захвата ввода.

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. Запускающий сигнал для внешнего события вводится через вывод ICP1, и он также может быть реализован аналоговым компаратором.

Когда изменяется логический уровень на выводе ICP1 или изменяется уровень АСО аналогового компаратора, и это изменение уровня захватывается блоком захвата ввода, происходит входной захват и 16-разрядные данные значения счета TCNT1 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 ACIS аналогового сравнения и регистра ACSR состояния.

Следует отметить, что изменение источника запуска может привести к захвату ввода, поэтому после изменения источника запуска ICF1 необходимо удалить один раз, чтобы избежать ошибочных результатов.

Сигнал захвата входного сигнала проходит через дополнительный шумоподавитель и отправляется к детектору фронта в соответствии с конфигурацией бита управления выбора входа захвата ICES1, чтобы убедиться, что обнаруженный фронт удовлетворяет условию запуска.

Шумоподавитель - это простой цифровой фильтр, который четыре раза проецирует входной сигнал и выводит только детектор фронта, когда четыре образца равны. Шумоподавитель управляется 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 запуска.

Измерение рабочего цикла внешнего сигнала требует, чтобы фронт запуска 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 В-канала. Длина - это значение времени, соответствующее числу счетчиков в регистре DTR1.

Как показано на рисунках, вставки мертвого времени OC1A и OC1B основаны на выходной форме сигнала канала В. 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. Противоположная полярность.

Когда бит DTEN1 установлен на «0», функция вставки мертвого времени отключена. Выходные сигналы OC1A и OC1B являются формами сигналов, генерируемыми соответствующими результатами сравнения.

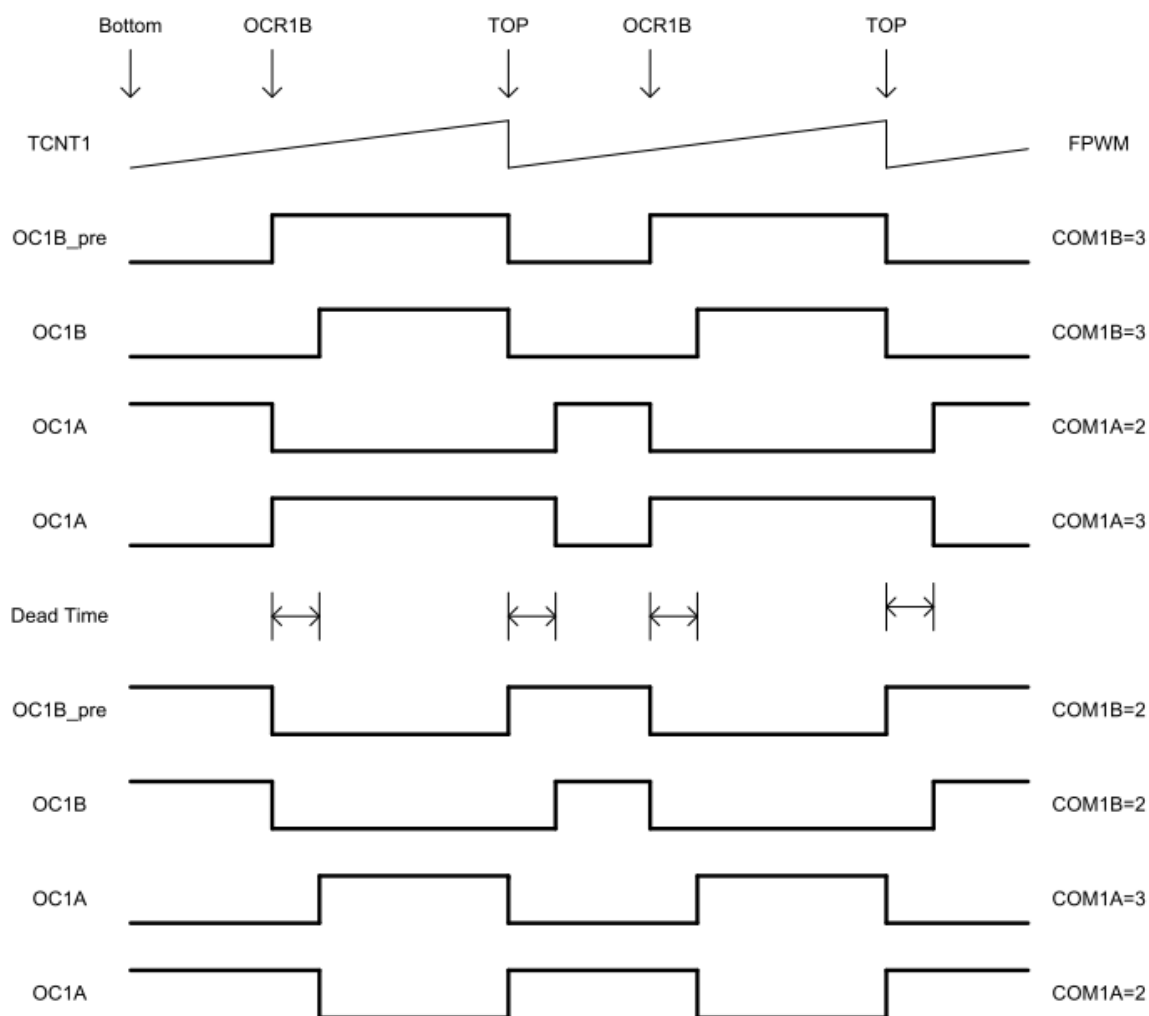


Рис.12.2. Контроль мертвого времени таймера TC1 в режиме FPWM

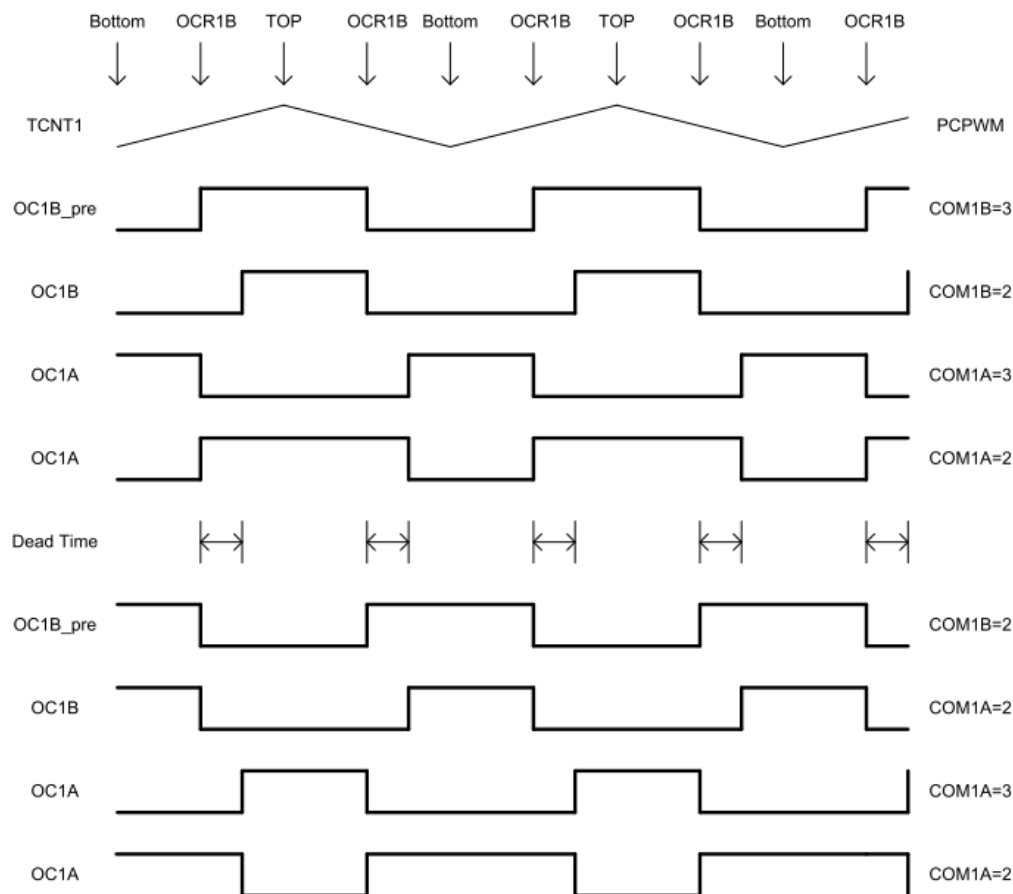


Рис.12.3. Контроль мертвого времени таймера TC1 в режиме RCPWM

12.2.8. Режим высокоскоростного счетчика

В режиме высокоскоростной синхронизации в качестве источника синхронизации для подсчета используется высокочастотный синхроимпульс, который используется для генерации высокочастотных и высокочастотных сигналов PWM.

Этот высокочастотный тактовый генератор генерируется путем умножения выходного тактового сигнала $tc32m$ внутреннего 32M RC-генератора на 2 раза. Поэтому перед входом в высокочастотный режим необходимо включить функцию частотного умножения внутреннего 32M RC-генератора, т. е. установить бит F2XEN регистра 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 такты 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 во время работы 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 в операционных системах);
- 7) Чтение или запись регистров А или В.
- 8) Дождитесь системного такта (зарегистрируйтесь в режиме NOP или тактов операционной системы)

Когда считывается регистр в высокоскоростной тактовом домене, может быть непосредственно прочитан регистр с 8-битными данными, и когда считывается значение 16-разрядного регистра (OCR1A, OCR1B, ICR1, TCNT1), первое значение регистра регистрируется первым.

После ожидания одного системного такта считывается значение старшего 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 прочитайте значение TCNT1.

Чтобы прочитать OCR1A, OCR1B и ICR1, выполните следующие действия:

- 1) Прочтите OCR1AL / OCR1BL / ICR1L;
- 2) Дождитесь системных часов (NOP);
- 3) Прочтите OCR1AH / OCR1BH / ICR1H.

Чтобы прочитать TCNT1, выполните следующие действия:

- 1) Установите CS1 в ноль,
- 2) Дождитесь системного тактового сигнала (NOP);
- 3) Прочитайте значение TCNT1L;
- 4) Дождитесь системного такта (NOP); Прочтите значение TCNT1H ,

12.3. Регистры Описание

Список регистров TC1

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 запуска
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 тактов

Примечание: Регистры с «*» работают в системных и высокоскоростных

тактовых доменах. Регистры без «*» работают только в домене системных тактов.

TCCR1A - Регистр управления A

TCCR1A - Регистр управления A								
Адрес: 0x80			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	COM1A1	COM1A0	COM1B1	COM1B0	-	-	WGM11	WGM10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7	COM1A1	Подробнее см. Описание таблицы управления выходным сравнением.						
6	COM1A0							
5	COM1B1	Подробнее см. Описание таблицы управления выходным сравнением.						
4	COM1B0							
3:2	-	Зарезервировано						
1	WGM11	Подробнее см. Описание таблицы режимов генерации волны.						
0	WGM10							

COM1x - Режим Управление сравнением без PWM

COM1x [1:0]	Описание
0	OC1x отключен, операция ввода-вывода общего назначения
1	Переключение сигнала OC1x при равенстве сравнения
2	Очистить сигнал OC1x при равенстве сравнения
3	Установите сигнал OC1x при равенстве сравнения

COM1x - 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 сигнал 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

TCCR1B - Регистр управления B								
Адрес: 0x81			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7	ICNC1	Когда бит ICNC1 установлен в «1», подавитель шума входного сигнала включен. В это время вход внешнего контакта ICP1 фильтруется, и входной сигнал действителен, когда четыре последовательных выборки равны. Эта функция задерживает захват ввода на 4 Циклы тактов. При установке бита ICNC1 в «0» подавление шума захвата входного сигнала отключается, а вход внешнего вывода ICP1 захватывает прямым действием.						
6	ICES1	Когда бит ICES1 установлен в «1», передний фронт выбранного уровня запускает захват ввода, а бит ICES1 установлен на «0», задний фронт выбранного уровня запускает захват ввода. Когда событие захватывается, значение счетчика копируется в регистр ICR1 и устанавливается флаг ICF1 ввода. Если прерывание включено, генерируется прерывание ввода.						
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 Описании режима генерации волны.						
3	WGM12							
2	CS12	Предделение тактов источника синхронизации для счетчика времени 0.						
		CS1[2:0]	Описание					
		0	Нет источника синхронизации, стоп-счет					
1	CS11	1	clk _{sys}					
		2	clk _{sys} / 8 из предделителя					
		3	clk _{sys} / 64 из предделителя					
		4	clk _{sys} / 256 из предделителя					
0	CS10	5	clk _{sys} / 1024 от предделителя					
		6	Внешние такты T1, запуск со спадающим фронтом					
		7	Внешние такты T1, запуск 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	OCR1A	TOP	BOTTOM
12	CTC	ICR1	Немедленно	MAX

13	Зарезервировано	-	-	-
14	FPWM	ICR1	TOP	TOP
15	FPWM	OCR1A	TOP	TOP

TCCR1C - Регистр управления C

TCCR1C - Регистр управления C								
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 управления принудительной установки выхода сравнения. При работе в режиме без PWM равенство сравнения может быть сгенерировано путем записи «1» в бит FOC1A. Принудительное сравнение не устанавливает флаг OCF1A и не перезагружает или не очищает таймер, но выходной вывод OC1A будет обновляться в соответствии с настройкой COM1A, как только произойдет истинное сравнение. Чтение FOC1A всегда возвращает ноль.						
6	FOC1B	Бит управления принудительной установки выхода сравнения. При работе в режиме без PWM равенство сравнения может быть сгенерировано путем записи «1» в бит FOC1B. Принудительное сравнение не устанавливает флаг OCF1B и не перезагружает или не очищает таймер, но выходной вывод OC1A будет обновляться в соответствии с настройкой COM1B, как только произойдет истинное сравнение. Чтение FOC1B всегда возвращает ноль.						
5	DOC1B	Когда бит DOC1B в «1», активируется сигнал сравнения выключения источника запуска OC1B. Когда происходит событие запуска, 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 запуска.						
4	DOC1A	Когда бит DOC1A в «1», активируется сигнал сравнения выключения источника запуска OC1A. Когда происходит событие запуска, 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 запуска						
3	DTEN1	Если бит DTEN1 в «1» - мертвое времени включено. Оба OC1A и OC1B вставляют мертвое время в форму волны, генерируемую выходным сигналом сравнения В-канала, а вставленный мертвый интервал определяется временем отсчета регистра 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

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	Перепополнение TC2	Передний фронт TOV2 отключает выходной сигнал OC1x
1	7	Перепополнение 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

TCNT1L - Регистр счетчиков TC1								
Адрес: 0x84			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	TCNT1L7	TCNT1L6	TCNT1L5	TCNT1L4	TCNT1L3	TCNT1L2	TCNT1L1	TCNT1L0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	TCNT1L	<p>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 прочитайте TCNT1L.</p> <p>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 значения 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.</p>						

TCNT1H - Счетчик значений Старший байт

TCNT1H - Регистр счетчиков TC1								
Адрес: 0x85			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	TCNT1H7	TCNT1H6	TCNT1H5	TCNT1H4	TCNT1H3	TCNT1H2	TCNT1H1	TCNT1H0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	TCNT1H	Старший байт значения счетчика TCNT1. См. пояснения TCNT1L						

ICR1L - Регистр захвата Младший байт

ICR1L - Регистр захвата - младший байт								
Адрес: 0x86			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	ICR1L7	ICR1L6	ICR1L5	ICR1L4	ICR1L3	ICR1L2	ICR1L1	ICR1L0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	ICR1L	Младший байт значения ввода входного сигнала TCNT1. ICR1H и ICR1L объединены для формирования 16-битного ICR1. Чтение и запись 16-битных регистров требует двух операций. При записи 16-битного ICR1 сначала должен быть записан ICR1H. При чтении 16-битного ICR1 сначала прочитайте ICR1L. Когда запускается входной захват, значение счетчика TCNT1 обновляется и копируется в регистр ICR1. Регистр ICR1 также может использоваться для определения TOP-значения счетчика.						

ICR1H - Регистр захвата Старший байт

ICR1H - Регистр захвата - старший байт								
Адрес: 0x87			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	ICR1H7	ICR1H6	ICR1H5	ICR1H4	ICR1H3	ICR1H2	ICR1H1	ICR1H0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	ICR1H	Старший байт значения счетчика ICR1. См. пояснения ICR1L						

OCR1AL - Регистр сравнения A выходных данных Младший байт

OCR1AL - Регистр сравнения A выходных данных - младший байт								
Адрес: 0x88			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	OCR1AL7	OCR1AL6	OCR1AL5	OCR1AL4	OCR1AL3	OCR1AL2	OCR1AL1	OCR1AL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	OCR1AL	<p>Младший байт регистра сравнения A. OCR1AL и OCR1AH объединены для формирования 16-разрядного OCR1A. Чтение и запись 16-битных регистров требует двух операций. При записи 16-битного OCR1A сначала необходимо записать OCR1AH. При чтении 16-битного OCR1A сначала прочитайте OCR1AL. OCR1A непрерывно сравнивается со значением счетчика TCNT1. Сопоставление сравнения может использоваться для генерации прерывания сравнения или для генерации формы сигнала на выводе OC1A. При использовании режима PWM регистр OCR1A использует двойной буферный регистр. Однако функция двойной буферизации отключается в обычном режиме работы и режиме согласования. Двойная буферизация может синхронизировать обновление регистра OCR1A с максимальным или минимальным временем отсчета, предотвращая генерацию асимметричных импульсов ШИМ и устраняя некорректные импульсы. Когда используется функция двойной буферизации, процессор обращается к регистру буфера OCR1A. Когда функция двойной буферизации отключена, процессор обращается к OCR1A.</p>						

OCR1AH - Регистр сравнения A выходных данных - старший байт								
Адрес: 0x89			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	OCR1AH7	OCR1AH6	OCR1AH5	OCR1AH4	OCR1AH3	OCR1AH2	OCR1AH1	OCR1AH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	OCR1AH	Старший байт значения регистра OCR1A. См. пояснения OCR1AL						

OCR1BL - Регистр сравнения В выходных данных - младший байт								
Адрес: 0x8A			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	OCR1BL7	OCR1BL6	OCR1BL5	OCR1BL4	OCR1BL3	OCR1BL2	OCR1BL1	OCR1BL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	OCR1BL	<p>Младший байт регистра сравнения А. OCR1BL и OCR1BH объединены для формирования 16-разрядного OCR1B. Чтение и запись 16-битных регистров требует двух операций. При записи 16-битного OCR1B сначала необходимо записать OCR1BH. При чтении 16-битного OCR1B сначала прочитайте OCR1BL. OCR1B непрерывно сравнивается со значением счетчика TCNT1. Сопоставление сравнения может использоваться для генерации прерывания сравнения или для генерации формы сигнала на выводе OC1B. При использовании режима PWM регистр OCR1B использует двойной буферный регистр. Однако функция двойной буферизации отключается в обычном режиме работы и режиме согласования. Двойная буферизация может синхронизировать обновление регистра OCR1B с максимальным или минимальным временем отсчета, предотвращая генерацию асимметричных импульсов ШИМ и устраняя некорректные импульсы. Когда используется функция двойной буферизации, процессор обращается к регистру буфера OCR1B. Когда функция двойной буферизации отключена, процессор обращается к OCR1B.</p>						

OCR1BH - Регистр сравнения В выходных данных - старший байт								
Адрес: 0x8B			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	OCR1BH7	OCR1BH6	OCR1BH5	OCR1BH4	OCR1BH3	OCR1BH2	OCR1BH1	OCR1BH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	OCR1BH	Старший байт значения регистра OCR1B. См. пояснения OCR1BL						

DTR1L - Регистр контроля мертвого времени								
Адрес: 0x8C			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	DTR1L							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	DTR1L	Когда бит DTEN1 установлен в »1«, OC1A и OC1B являются дополнительными выходами, мертвое время, вставленное на выход OC1A, определяется тактами счетчика DTR1L.						

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 запуск 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 В 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 В - отключен.						
1	OCIE1A	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 TC0 прерывания совпадения А - 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

TIFR1 - Регистр флагов прерываний счетчика/таймера								
Адрес: 0x36			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	-	-	ICF1	-	-	OCF1B	OCF1A	TOV1
R/W	-	-	R/W	-	-	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:6	-	Зарезервировано						
5	ICF1	Флаг захвата. Флаг ICF1 устанавливается, когда происходит событие ввода. Когда ICR1 используется как верхнее значение счетчика, а значение счета достигает значения TOP, устанавливается флаг ICF1. Если для ICIE1 установлено значение «1» и установлен флаг глобального прерывания, будет генерироваться прерывание ввода. ICF1 будет очищен автоматически при выполнении этой процедуры обслуживания прерываний или может быть очищен путем записи «1» в бит ICF1.						
4:3	-	Зарезервировано						
2	OCF1B	Выход TC1 флаг сравнения В. Когда TCNT1 равен OCR1B, блок сравнения дает сигнал равенства и устанавливает флаг сравнения OCF1B. Если Флаг прерывания сравнение В OCIE1B установлено в «1», и установлен флаг глобального прерывания, будет генерироваться прерывание сравнения с результатом В. Флаг OCF1B очищается автоматически при выполнении процедуры обслуживания прерывания или может быть очищен путем записи «1» в бит OCF1B.						

13. Таймер / счетчик 0/1 Предделитель

- Два 10-разрядных предварительных делителя
- TC0 и TC1 мультиплексный предделитель CPS10 в мультиплексном режиме
- Одноточечный предварительный делитель TC0 CPS10 TC1 в одиночном режиме
- Сброс программного обеспечения предделитель CPS1

13.1. Обзор

В мультиплексном режиме ($PSS1 = 0$) TC0 и TC1 используют 10-разрядный предварительный делитель CPS10, но имеют разные настройки деления.

В режиме однократного использования ($PSS1 = 1$) TC0 независимо использует предварительный делитель CPS10, а TC1 независимо использует предварительный делитель CPS1. Они имеют разные настройки частотного деления.

Нижеследующее описание используется для TC0 и TC1, где n представляет 0 или 1.

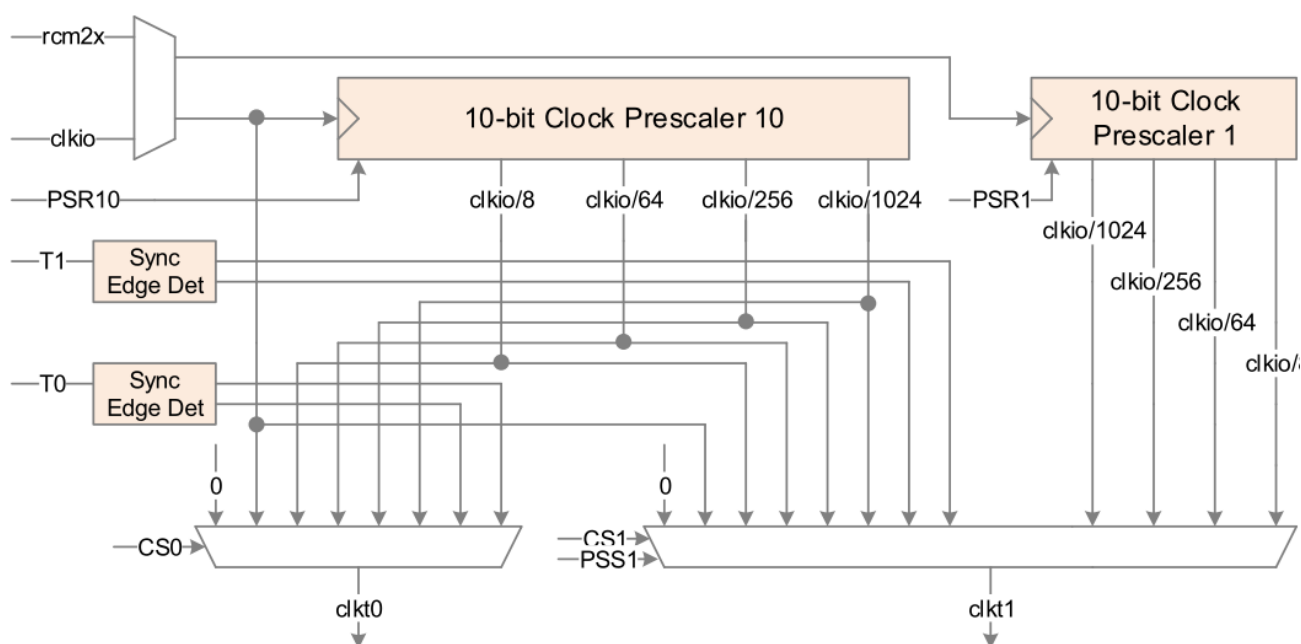


Рис.13.1. Структурная схема блока предделителя TC0 / TC1

13.2. Внутренний источник синхронизации

Когда $CSn [2: 0] = 1$ установлен, счетчик таймера может управляться непосредственно системным тактовым сигналом $clkio$ или высокоскоростным синхронизирующим импульсом $rcm2x$ (внутренний выходной сигнал генератора 32M RC, умноженный на 2).

Предделитель может выводить 4 разные тактовые частоты, $clkio / 8$, $clkio / 64$, $clkio / 256$ и $clkio / 1024$.

13.3. Сброс делителя

13.3.1. Мультиплексный режим

Когда бит PSS1 установлен на «0», TC0 и TC1 совместно используют предделитель CPS10.

Предделитель работает независимо, его работа не зависит от логики выбора тактов TC, и она разделяется TC0 и TC1. Поскольку управление не влияет на выбор тактов, состояние предделителя влияет на применение разделенных тактов.

Эффект возникает, когда таймер включен, а выход предварительного каскада выбран в качестве источника тактовых импульсов ($6 > CSn[2:0] > 1$). Это может занять от 1 до $N + 1$ системных часов, чтобы включить таймер к первому счету, где N - коэффициент предварительной 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.

13.3.2. Режим однократного использования.

Когда бит PSS1 установлен в «1», TC0 независимо использует предварительный каскад CPS10, а предварительный делитель сбрасывается бит PSR10.

TC1 использует предварительный делитель CPS1 независимо, а сброс предварительного делителя управляется битом PSR1.

Соответствующие сбросы работают независимо и не влияют на другие предделители.

13.4. Источник внешних Тактов.

Внешний источник синхронизации, обеспечиваемый выводом T0/T1, может использоваться в качестве источника тактовых импульсов.

Сигнал на выводе T0/T1 используется как источник счетчика часов после прохождения логики синхронизации и детектора фронта.

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 границ на выводах, изменение уровня в T0 / T1 требует задержки от 2,5 до 3,5 системных часов для обновления счетчика.

Включение или включение входного тактового сигнала должно занимать по меньшей мере один системный тактовый цикл до того, как T0 / T1 будет стабильным, в противном случае существует возможность импульсного счетчика ошибок.

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

PSSR - Выбор предделителя								
Адрес: 0x83			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	PSS1	-	-	-	-	-	-	PSR1
R/W	R/W	-	-	-	-	-	-	W
Бит	Имя	Определение бит						
7	PSS1	<p>Бит управления выбрать предделителя.</p> <p>Когда бит PSS1 установлен в «1», предварительный делитель используется один. TC0 использует только предделитель CPS10, а TC1 использует только предделитель CPS1. Когда бит PSS1 установлен в «0», предделитель мультиплексируется. TC0 и TC1 разделяют предделитель CPS10.</p> <p>Предделитель CPS1 недействителен и всегда будет сброшен.</p>						
6:1	-	Зарезервировано						
0	PSR1	<p>Бит управления сброса Предделителя CPS1.</p> <p>Бит PSR1 действителен только в одноразовом режиме. Когда бит PSR1 установлен в «1», предварительный делитель CPS1 будет сброшен. Аппарат очищает бит PSR1 после сброса. Если бит PSR1 установлен на «0», установка недействительна. Значение для чтения этого бита всегда будет «0».</p>						

14. Таймер/Счетчик 2 - 8-разрядный

- 8-разрядный счетчик
- Два независимых блока сравнения
- Автоматическая очистка счетчика и загрузка при равенстве сравнения
- С Фазовой коррекцией ШИМ-выход без импульсных помех
- Генератор частоты
- Счетчик внешних событий
- Предварительные делители тактов - 10 бит
- Прерывания при переполнении и равенстве сравнения
- Подключение внешнего 32,768KHz RTC часового резонатора

14.1. Обзор

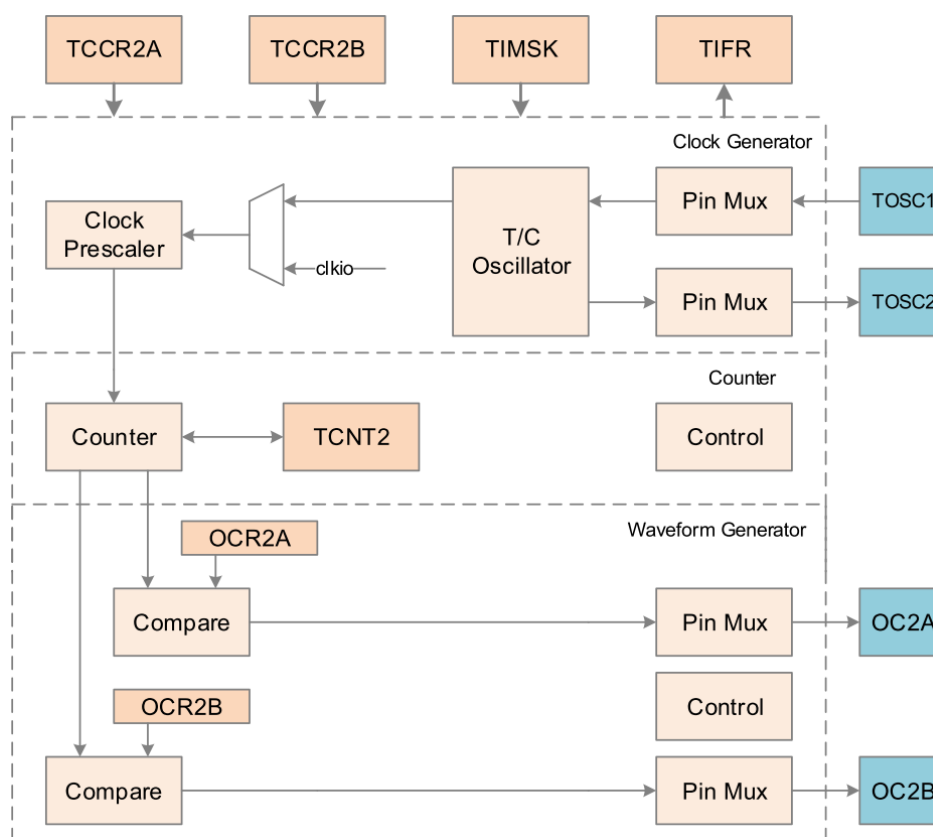


Рис.14.1. Структурная схема TC2

TC2 - это 8-разрядный счетчик 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 очищает, увеличивает или уменьшает в каждый такт Clkt2.

Clkt2 может быть сгенерирован из внутреннего источника синхронизации или внешнего источника синхронизации. При использовании внешнего резонатора 32,768 кГц TC2 может использоваться как счетчик RTC времени.

Когда значение счетчика TCNT2 счетчика достигает максимального значения

(равное максимальному значению 0xFF или регистру сравнения OCR2A/OCR2B, определяемому как TOP, максимальное значение определяется как MAX для различия), счетчик очищается или вычитается.

Когда значение счетчика TCNT2 достигает минимального значения (эквивалентно 0x00, определенное как BOTTOM), счетчик увеличивается.

Когда значение счетчика TCNT2 счетчика достигает OCR2A/OCR2B, которое также вызывается, когда происходит совпадение сравнения, выходной сигнал сравнения OC2A / OCR2B очищается или устанавливается для генерации формы ШИМ.

14.2. Режим 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 «А» и «В», нижний выходной «х» используется для представления каналов двух блоков сравнения вывода.

14.2.1. Нормальный режим

Обычный режим - это самый простой режим работы счетчика таймера. В это время биты управления режимом генерации сигнала WGM2 [2: 0] = 0, а максимальное значение TOP счетчика - MAX (0xFF). В этом режиме счетчик увеличивается на один для каждого счета тактов.

Когда счетчик достигает переполнения TOP, он возвращается в BOTTOM, чтобы снова начать накапливать.

Флаг переполнения счетчика таймера TOV2 устанавливается на одном и том же счете, значение счетчика TCNT2 которого становится равным нулю. Флаг TOV2 в этом режиме похож на 9-й бит, но он будет устанавливаться и не будет очищаться. Процедура обслуживания прерывания переполнения автоматически очищает флаг TOV2 и может использоваться программным обеспечением для увеличения разрешения счета времени.

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

Форма выходного сигнала сравнения OC2x может быть получена только тогда, когда регистр направления данных вывода OC2x задан, как выход.

Когда COM2x = 1, сигнал OC2x переключается при совпадении сравнения. В этом случае частоту формы волны можно вычислить по следующей формуле:

$$f_{oc2xnormal} = f_{sys} / (2 * N * 256)$$

N - Коэффициент делителя (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 CTC

Когда установлен параметр WGM2 [2: 0] = 2, счетчик 2 таймера переходит в режим CTC, а максимальное значение счетчика TOP - OCR2A. В этом режиме счетчик увеличивается на единицу для каждого счета тактов.

Когда значение счетчика TCNT2 равно TOP, счетчик очищается.

OCR2A определяет максимальное значение, которое 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 для сравнения на выходе, и прерывание будет генерироваться, когда установлен соответствующий бит разрешения прерывания. Регистр OCR2x, который является максимальным количеством, может быть обновлен в процедуре обслуживания прерываний. В этом режиме OCR2x не использует двойную буферизацию.

Будьте осторожны, когда счетчик обновляется до минимального значения без предварительного делителя или очень низкого предварительного делителя. Если значение, записанное в OCR2x, меньше текущего значения TCNT2, счетчик теряет сравнение. Перед тем, как произойдет следующее сравнение, счетчик должен рассчитывать до TOP, а затем рассчитывается с BOTTOM на OCR2x. Как и в обычном режиме, значение счетчика возвращается к счету BOTTOM для установки флага TOV2.

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 переключается при совпадении сравнения. В этом случае частоту формы волны можно вычислить по следующей формуле:

$$f_{oc2xctc} = f_{sys} / (2 * N * (1 + OCR2A))$$

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

Как видно из формулы, когда OCR2x установлен в 0x00, и нет предделения, можно получить выходной сигнал с максимальной частотой $f_{sys} / 2$.

14.2.3. Режим быстрой PWM

Когда задано значение WGM2 [2: 0] = 3 или 7, счетчик 2 таймера переходит в быстрый режим PWM и может использоваться для генерации высокочастотных сигналов ШИМ. Максимальное значение TOP должно быть MAX (0xFF) или OCR2A соответственно. Разница между быстрым режимом PWM и другими режимами PWM заключается в том, что это односторонняя операция. Счетчик добавляется до TOP от минимального значения 0x00 и возвращается в BOTTOM для повторного подсчета.

Когда значение счетчика TCNT2 достигает OCR2x или BOTTOM, выходной сигнал сравнения OC2x устанавливается или очищается в зависимости от настройки режима вывода сравнения COM2x. Подробнее см. Описание регистра.

Рабочая частота, счетчика в режиме быстрой PWM, в два раза выше, чем двунаправленная операция в режиме PWM с коррекцией фаз.

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 достигнет максимального значения, будет установлен флаг переполнения счетчика таймера TOV2, а значение буфера сравнения будет обновлено 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 OC2x задан как выход. Частоту формы волны можно вычислить по следующей формуле:

$$f_{oc2xfpwm} = f_{sys} / (N * (1 + TOP))$$

N - Коэффициент делителя (1, 8, 64, 256 или 1024).

Когда встречается сравнение между TCNT2 и OCR2x, генератор сигналов устанавливает (очищает) сигнал OC2x.

Когда TCNT2 очищается, генератор формы сигнала очищает (устанавливает) сигнал OC2x для генерации волны ШИМ. В результате экстремума OCR2x генерирует специальную форму ШИМ. Когда OCR2x установлен в 0x00, выходной PWM имеет узкий всплеск в каждом (1 + TOP) счете такта. Когда OCR2x установлен на максимальное значение, выходная форма сигнала постоянно высокая или низкая.

14.2.4. Режим PWM с фазовой коррекцией

Когда установлено значение WGM2 [2: 0] = 1 или 5, таймер/счетчик 2 переходит в режим PWM с фазовой коррекцией, а максимальное значение TOP равно MAX (0xFF) или OCR2A соответственно. Счетчик работает с двунаправленным счетом от BOTTOM до TOP, а затем вниз до BOTTOM и повторяет эту операцию.

Когда счетчик достигает TOP и BOTTOM, счетчик изменяют направление подсчета, а значение счетчика остается только на одном счете такта 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 однонаправленной работой максимальная частота, доступная для двунаправленной работы, меньше, но ее отличная симметрия более подходит для управления двигателем. В режиме PWM с фазовой коррекцией флаг TOV2 устанавливается, когда счетчик достигает BOTTOM, а значение буфера 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:

$$f_{OC2x} = f_{sys} / (N * TOP * 2)$$

N - Коэффициент делителя (1, 8, 64, 256 или 1024).

Во время подсчета 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 крайняя величина 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 сравнений не произойдет.

Первый случай - когда значение OCR2x изменяется с максимального значения 0xFF на другие данные.

Когда OCR2x является максимальным значением, а значение счет достигает максимального значения, вывод OC2x совпадает с результатом, когда сравнивается предыдущий счетчик убывания, то есть OC2x остается неизменным. Значение нового OCR2x обновляется (а не 0xFF) в это время, а значение OC2x сохраняется до тех пор, пока не произойдет совпадение сравнения и не произойдет опрокидывание. В это время сигнал OC2x не симметричен относительно минимального значения, поэтому ему нужно перевернуть сигнал OC2x, когда TCNT2 достигнет своего максимального значения.

То есть первый случай, когда сигнал OC2x переключается во время сравнения, не возникает.

Во втором случае, когда TCNT2 начинает отсчет с более высокого значения, чем OCR2x, сравнительное совпадение теряется, вызывая асимметричную ситуацию. Также необходимо перевернуть сигнал OC2x для достижения симметрии с обеих сторон минимального значения.

14.2.5. Асинхронный режим работы TC2

Когда в регистре ASSR бит AS2 равен «1», TC2 работают в асинхронном режиме, источник тактовых импульсов таймера/счетчика от внешнего генератора.

При работе TC2 в асинхронном режиме следует учитывать следующие моменты.

- Преобразование между синхронными и асинхронными режимами может привести к TCNT2, OCR2A, OCR2B и повреждение данных TCCR2A TCCR2B.

Процедура безопасности выглядит следующим образом:

1. Очистите биты OCIE2A, TOIE2 и OCIE2B, чтобы отключить прерывания TC2;
2. Установите бит AS2 для выбора соответствующего источника синхронизации;
3. Записать 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. Удалить флаги прерывания TC2;

6. Включить прерывания, которые необходимо использовать.

- **Генератор:** Лучше всего использовать резонатор часов с частотой 32,768 кГц. Частота тактовой частоты системы должна быть более чем в 4 раза выше, чем частота кристалла.
- 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 TOSC1-тактов. 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 буфер, поэтому запись TCNT2 не мешает написанию OCR2. Асинхронный регистр состояния регистров используется для проверки, были ли данные записаны в регистр назначения.
- Если TC2 используется как условие пробуждения в режиме ожидания MCU, устройство не может войти в режим SLEEP до конца каждого обновления регистра. В противном случае MCU может войти в режим SLEEP до того, как настройка TC2 вступит в силу, а TC2 не сможет разбудить систему.
- Если TC2 используется как условие пробуждения для режима ожидания MCU, необходимо принять меры для повторного входа в спящий режим. Для логики прерывания требуется сброс тактового цикла TOSC1. Если время от пробуждения до повторного входа в сон меньше одного тактового цикла TOSC1, прерывание больше не будет происходить, и устройство не сможет проснуться. Рекомендуются следующие методы:
 1. Введите соответствующие данные в каждый регистр.
 2. Подождите, пока не будет сброшен соответствующий флажок «Ожидание загрузки» ASSR.
 3. Войдите в спящий режим.
- Если выбран асинхронный режим работы, генератор TC2 будет продолжать работать, если он не переходит в режим выключения. Пользователь должен знать, что время установления этого осциллятора может составлять до 1 секунды. Поэтому пользователю рекомендуется ждать не менее 1 секунды после включения генератора TC2 перед использованием асинхронного режима работы TC2.
- **Проснуться:** Пробуждение в спящем режиме в асинхронном режиме работы: после выполнения условия прерывания процесс пробуждения начинается со следующего такта 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.

- **Чтение:** Чтение TCNT2 в течение короткого периода времени после пробуждения из спящего режима может возвращать неверные данные. Поскольку TCNT2 управляется асинхронным тактовым сигналом TOSC1, считывание TCNT2 должно выполняться через внутренний синхронизатор по системе регистр. Синхронизация происходит на переднем фронте каждого TOSC1.

Когда системные такты активируются после пробуждения из спящего режима, значение TCNT2 считывается перед входом в спящий режим и не будет обновляться до следующего восходящего фронта TOSC1.

Фаза TOSC1 полностью непредсказуема, когда она пробуждается из спящего режима и связана с временем пробуждения. Поэтому рекомендуемая последовательность для считывания значения TCNT2:

1. Запишите произвольное значение в OCR2A или TCCR2A;
 2. Подождите, пока не будет очищен соответствующий флаг загрузки 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 тактами таймера вместо системных тактов.

14.3. TC2 Предделитель.

Входной тактовый сигнал для предварительного делителя TC2 называется clk_{2s} . Внутренний системный тактовый сигнал clk_{io} или внешний источник синхронизации TOSC1 выбирается битом AS2 в регистре ASSR. По умолчанию используется подключение к системному такту clk_{io} .

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.

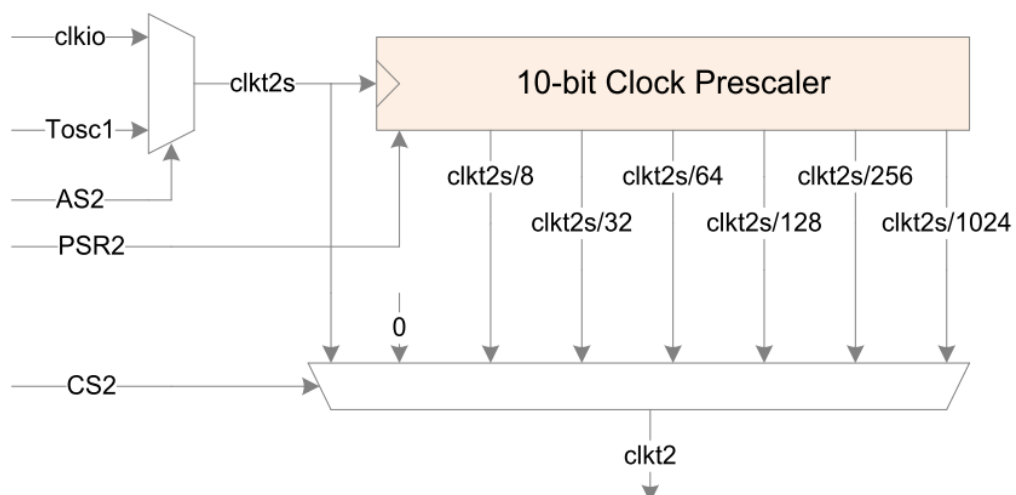


Рис.14.2. Структурная схема предварительного делителя TC2

На Рис.14.2. показан предварительный делитель TC2. Как показано на рисунке, возможными параметрами предварительного масштабирования являются: $\text{clkt2s} / 8$, $\text{clkt2s} / 32$, $\text{clkt2s} / 64$, $\text{clkt2s} / 128$, $\text{clkt2s} / 256$ и $\text{clkt2s} / 1024$.

Кроме того, вы можете выбрать `clk2s` и `0` (стоп-счет). Установка бита `PSR2` в регистре `SFIOR` сбрасывает предварительный делитель, позволяя пользователю начинать с предсказуемого предделителя.

14.4. Регистры Описания

Список регистров TC2

Регистр	Адрес	Значение по умолчанию	Описание
TCCR2A	0xB0	0x00	Контрольный регистр TC2 А
TCCR2B	0xB1	0x00	Контрольный регистр TC2 В
TCNT2	0xB2	0x00	Регистр значений счетчика TC2
OCR2A	0xB3	0x00	Регистр сравнения выходных данных TC2 А
OCR2B	0xB4	0x00	Регистр сравнения выходных данных TC2 В
ASSR	0xB6	0x00	Асинхронный регистр состояния TC2
TIMSK2	0x70	0x00	Регистр маски прерывания TC2
TIFR2	0x37	0x00	Регистр флага прерываний TC2

ТССР2А - Регистр управления ТС2 А

TCCR2A - Регистр управления TC2 A								
Адрес: 0xB0			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	COM2A1	COM2A0	COM2B1	COM2B0	-	-	WGM21	WGM20
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
Бит	Имя	Определение бит						
7	COM2A1	Подробнее см. Описание таблицы управления выходным сравнением.						
6	COM2A0							
5	COM2B1	Подробнее см. Описание таблицы управления выходным сравнением.						
4	COM2B0							
3:2	-	Зарезервировано						
1	WGM21	Подробнее см. Описание таблицы режимов генерации волны.						
0	WGM20							

ТССР2В - Регистр управления ТС2 В

TCCR2B - Регистр управления TC2B								
Адрес: 0xB1			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	FOC2A	FOC2B	-	-	WGM22	CS22	CS21	CS20
R/W	W	W	-	-	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7	FOC2A	Бит управления принудительной установки выхода сравнения. При работе в режиме без PWM равенство сравнения может быть сгенерировано путем записи «1» в бит FOC2A. Принудительное сравнение не устанавливает флаг OCF2A и не перезагружает или не очищает таймер, но выходной вывод OC2A будет						

		обновляться в соответствии с настройкой COM2A, как только произойдет истинное сравнение. Чтение FOC2A всегда возвращает ноль.																		
6	FOC2B	Бит управления принудительной установки выхода сравнения. При работе в режиме без PWM равенство сравнения может быть сгенерировано путем записи «1» в бит FOC2B. Принудительное сравнение не устанавливает флаг OCF2B и не перезагружает или не очищает таймер, но выходной вывод OC2B будет обновляться в соответствии с настройкой COM2B, как только произойдет истинное сравнение. Чтение FOC2B всегда возвращает ноль.																		
5:4		Зарезервировано																		
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	<div>Предделение тактов источника синхронизации для счетчика времени 0.</div> <table><tr><th>CS2[2:0]</th><th>Описание</th></tr><tr><td>0</td><td>Нет источника синхронизации, стоп-счет</td></tr><tr><td>1</td><td>clk_{sys}</td></tr><tr><td>2</td><td>clk_{sys} / 8 из предделителя</td></tr><tr><td>3</td><td>clk_{sys} / 32 из предделителя</td></tr><tr><td>4</td><td>clk_{sys} / 64 из предделителя</td></tr><tr><td>5</td><td>clk_{sys} / 128 из предделителя</td></tr><tr><td>6</td><td>clk_{sys} / 256 из предделителя</td></tr><tr><td>7</td><td>clk_{sys} / 1024 от предделителя</td></tr></table>	CS2[2:0]	Описание	0	Нет источника синхронизации, стоп-счет	1	clk _{sys}	2	clk _{sys} / 8 из предделителя	3	clk _{sys} / 32 из предделителя	4	clk _{sys} / 64 из предделителя	5	clk _{sys} / 128 из предделителя	6	clk _{sys} / 256 из предделителя	7	clk _{sys} / 1024 от предделителя
CS2[2:0]	Описание																			
0	Нет источника синхронизации, стоп-счет																			
1	clk _{sys}																			
2	clk _{sys} / 8 из предделителя																			
3	clk _{sys} / 32 из предделителя																			
4	clk _{sys} / 64 из предделителя																			
5	clk _{sys} / 128 из предделителя																			
6	clk _{sys} / 256 из предделителя																			
7	clk _{sys} / 1024 от предделителя																			
1	CS21																			
0	CS20																			

COM2n - Нормальный режим и режим CTC Управление сравнением

COM2x1	COM2x0	Описание
0	0	Нормальных операций порта, OC2x отключены.
0	1	Сигнал Переключить OC2x при равенстве
1	0	Очистить сигнал OC2x при Сравнении
1	1	Установить сигнал OC2x на Сравнении

COM2n - Быстрый Режим PWM Управление сравнением

COM2x1	COM2x0	Описание
0	0	Нормальных операций порта, OC2x разъединяется.
0	1	Зарезервировано
1	0	Очистить OC2x на Сравнении, Установить OC2x в основе (неинвертирующий режим)
1	1	Установить OC2x на Сравнении, Очистить OC2x в ВОТТОМ (инвертирующий режим)

COM2n - Режим PWM с коррекцией фаз Управление сравнением

COM2x1	COM2x0	Описание
0	0	Нормальных операций порта, OC2x разъединяется.
0	1	Резервировано.
1	0	Очистить OC2x на Сравнении Компаратора когда Счет Вверх. Установленный OC2x на Сравнении Компаратора когда Счет Вниз.
1	1	Установленный OC2x на Сравнении Компаратора когда Счет Вверх. Очистить OC2x 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	OCR2A	Немедленно	MAX
3	FPWM	0xFF	TOP	MAX
4	Зарезервировано	-	-	-
5	PCPWM	OCR2A	TOP	BOTTOM
6	Зарезервировано	-	-	-
7	FPWM	OCR2A	TOP	TOP

ТСНТ2 - Регистр счетчиков ТС2

TCNT2 - Регистр счетчиков TC2								
Адрес: 0xB2			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	TCNT27	TCNT26	TCNT25	TCNT24	TCNT23	TCNT22	TCNT21	TCNT20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	TCNT2	Регистр значений счетчика TC2. Доступ к считыванию / записи 8-битного значения счетчика выполняется непосредственно через регистр TCNT2. Запись ЦП в регистр TCNT2, предотвратит появление совпадения сравнения в следующем такте таймера, даже если таймер остановлен. Это позволяет инициализировать значение регистра TCNT2 в соответствии с значением OCR2, не вызывая прерывания. Если значение, записанное в TCNT2, равно или обходит значение OCR2, сравнение теряется, что приводит к возникновению неправильной формы сигнала. Таймер прекращает отсчет, когда источник синхронизации не выбран, но CPU все еще может получить доступ к TCNT2. Счетчик записи процессора имеет более высокий приоритет, чем операция очистки или добавления или вычитания.						

OCR2A - Регистр сравнения выходных данных TC2 A

OCR2A - Регистр сравнения выходных данных TC2 A								
Адрес: 0xB3			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	OCR2A7	OCR2A6	OCR2A5	OCR2A4	OCR2A3	OCR2A2	OCR2A1	OCR2A0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	OCR2A	Регистр сравнения выходных данных TC2. OCR2A содержит 8-битные данные, которые непрерывно сравниваются со значением счетчика TCNT2. Равенство сравнения может использоваться для генерации прерывания сравнения вывода или для генерации формы сигнала на выводе OCR2A. При использовании режима PWM регистр OCR2A использует регистр с двойным буфером. Однако двойная буферизация отключена в нормальном режиме и соответствует режиму очистки. Двойная буферизация синхронизирует обновление регистра OCR2A с максимальным или минимальным временем отсчета, предотвращая генерацию асимметричных импульсов ШИМ и устраняя возмущающие импульсы. При использовании функции двойной буферизации CPU обращается к регистру буфера OCR2A. Когда функция двойной буферизации отключена, CPU обращается к самому OCR2A.						

OCR0A - Регистр сравнения выходных данных TC0 A								
Адрес: 0xB4			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	OCR2B7	OCR2B6	OCR2B5	OCR2B4	OCR2B3	OCR0B2	OCR2B1	OCR2B0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	OCR2B	Регистр сравнения выходных данных TC2. OCR2B содержит 8-битные данные, 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 устраняя возмущающие импульсы. При использовании функции двойной буферизации CPU обращается к регистру буфера OCR2B. Когда функция двойной буферизации отключена, CPU обращается к самому OCR2B.						

ASSR - Регистр состояния асинхронного интерфейса								
Адрес: 0xB6			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	INTCK	-	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7	INTCK	Асинхронный бит управления выбором тактов. Когда бит INTCK установлен в 1, внутренний RC32K выбирается как источник асинхронных тактов. Когда бит INTCK установлен в 0, кварцевый резонатор выбираются в качестве источника асинхронных тактов.						
6	-	Зарезервировано						
5	AS2	Бит управления асинхронным режимом таймера 2. Когда бит AS2 установлен в 1, таймер 2 работает в асинхронном режиме, а его источник синхронизации выбирается бит INTCK. Когда бит AS2 установлен в 0, таймер 2 работает в синхронном режиме, а его источником синхронизации является Clkio. Когда значение AS2 изменяется, значения регистров TCNT2, OCR2A, OCR2B, TCCR2A и TCCR2B могут быть неправильными и нуждаются в переконфигурации.						
4	TCN2UB	Регистр TCNT2 обновляет флаг. Когда таймер 2 работает в асинхронном режиме, бит TCN2UB будет установлен при записи в TCNT2. Когда значение TCNT2 обновляется, аппаратное обеспечение очищает бит TCN2UB. TCNT2 может быть обновлен только тогда, когда бит TCN2UB равен нулю.						
3	OCR2AUB	OCR2A регистрирует флаг обновления. Когда таймер 2 работает в асинхронном режиме, бит OCR2AUB будет установлен при записи на OCR2A. Когда значение OCR2A обновляется, аппаратное обеспечение очищает бит OCR2AUB. OCR2A может обновляться только тогда, когда бит OCR2AUB равен нулю.						
2	OCR2BUB	OCR2B регистрирует флаг обновления. Когда таймер 2 работает в асинхронном режиме, бит OCR2BUB будет установлен при записи на OCR2B. Когда значение OCR2B обновляется,						

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.

15. Последовательный периферийный интерфейс SPI

- Полнодуплексная, трехпроводная синхронная передача данных
- Операция ведущего или ведомого устройства
- Передача с наименьшим или наивысшим приоритетом
- 7 программируемых скоростей передачи данных
- Флаг прерывания Конец передачи
- Запись символа столкновений Механизм защиты
- Пробуждение из режима ожидания
- Хост работает с режимом двойной скорости
- Поддерживает хост с двухпроводным режимом ввода.

15.1. Обзор.

SPI в основном состоит из трех частей: предварительный делитель тактов, детектор часов, детектор выбора подчиненного устройства, передатчик и приемник.

Регистры управления и состояния разделяются этими тремя частями.

Предделитель тактов работает только в режиме работы хоста. Биты управления скоростью передачи битов выбирают коэффициент частотного деления для генерации соответствующих тактов и вывода на вывод SPCK.

Детектор тактов работает только в режиме ведомого, обнаруживает входной сигнал тайм-сигнала из выводов SPCK и сдвигает регистры сдвига передачи и приема в соответствии с режимом передачи данных SPI. Извещатель подчиненного выбора 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, буфера приема и логики управления приемом.

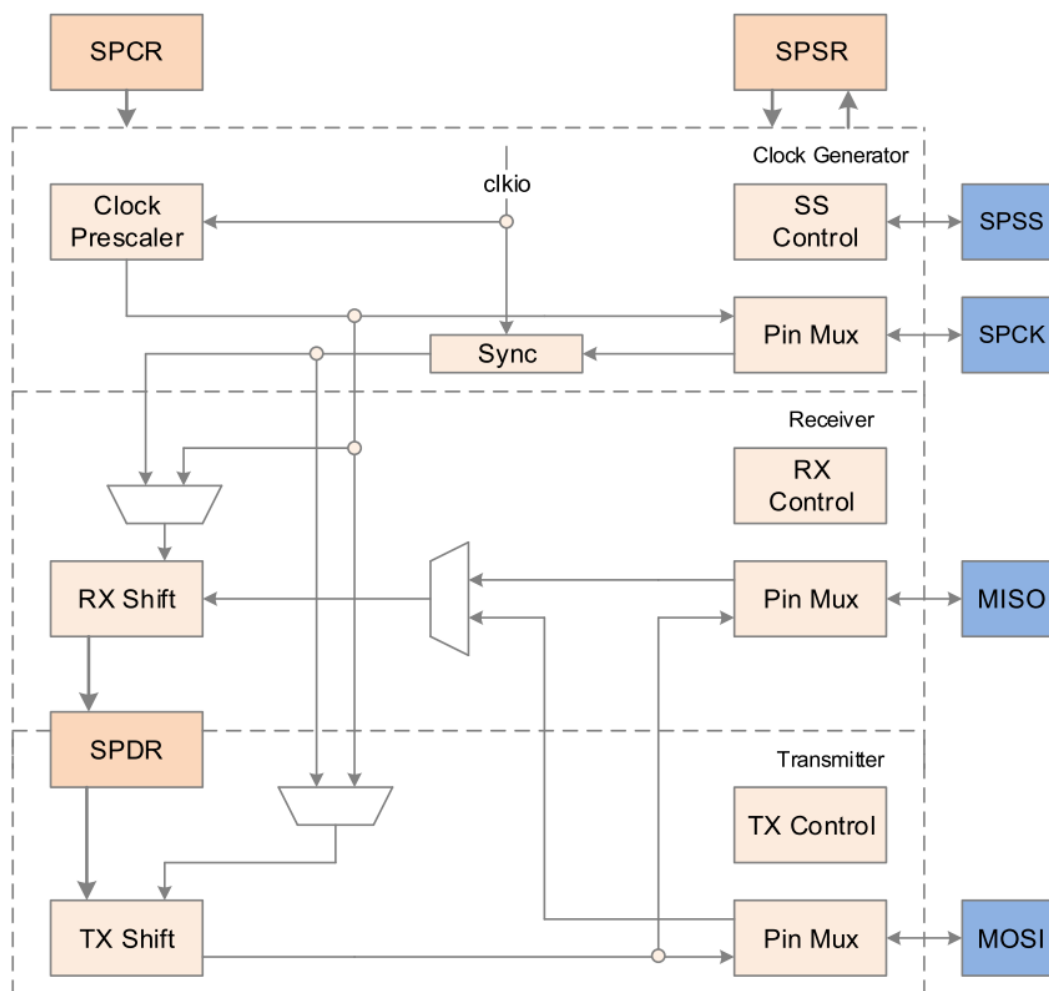


Рис.15.1. Структурная схема SPI

15.2. Генерация Тактов.

Логика генерации часов делится на предварительный делитель основного тактового генератора и ведомый тактовый детектор, которые работают в режиме ведущего и ведомого режимов соответственно. Предварительный делитель тактов использует биты управления скоростью передачи битов и биты управления скоростью передачи для выбора коэффициента деления частоты и генерации соответствующих тактовых импульсов частоты (доступно всего семь выбираемых коэффициентов частотного деления. Подробнее см. Описание регистра).

Выход на контакт SPCK Связь обеспечивает такты, обеспечивая при этом синхронизацию сдвига для внутренних регистров сдвига передачи и приема. Детектор часов выполняет обнаружение кромок 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 часов SPCK должны быть больше, чем 2 тактовых цикла системы.

15.3. Передача и прием.

Модуль SPI поддерживает одновременную передачу и прием в однопроводном режиме и только двухпроводный прием только в двухпроводном режиме..

15.3.1. Однолинейная передача и прием

Хост SPI устанавливает «0» сигнал на выводе SPSS, который должен указывать начало процесса передачи.

Ведущий и подчиненный подготавливают данные, которые необходимо передать.

Ведущий генерирует тактовый импульс синхронизирующего сигнала на выводе SPCK для обмена данными.

Основные данные сдвигаются из MOSI и передаются в MISO. Ведомые данные удаляются из MISO и перемещаются в MOSI.

После передачи данных хост устанавливает сигнал на выводе 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 ведомый 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 на ведомый и перемещает данные из ведомого 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 на выводе 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 запуск MISO. Теперь 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 на выводе 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 один буфер в направлении передачи и два буфера в направлении приема. При передаче данных регистр SPDR не должен записываться до завершения процесса сдвига.

При получении данных полученный символ должен быть прочитан путем доступа к регистру SPDR до завершения следующего процесса смещения байта, иначе предыдущий байт будет потерян.

15.3.2. Принимающий двухпроводный прием

Двухпроводный режим модуля SPI действителен только в режиме работы с хостом.

Разница в однопроводном режиме заключается в том, что как MOSI, так и

MISO используются для хоста для приема данных, и каждый тактовый импульс SPCK одновременно принимает 2 бита данных (Сначала выводятся данные по линии MISO, данные по линии MOSI поступают позже)

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.

15.4. Структура 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.

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:

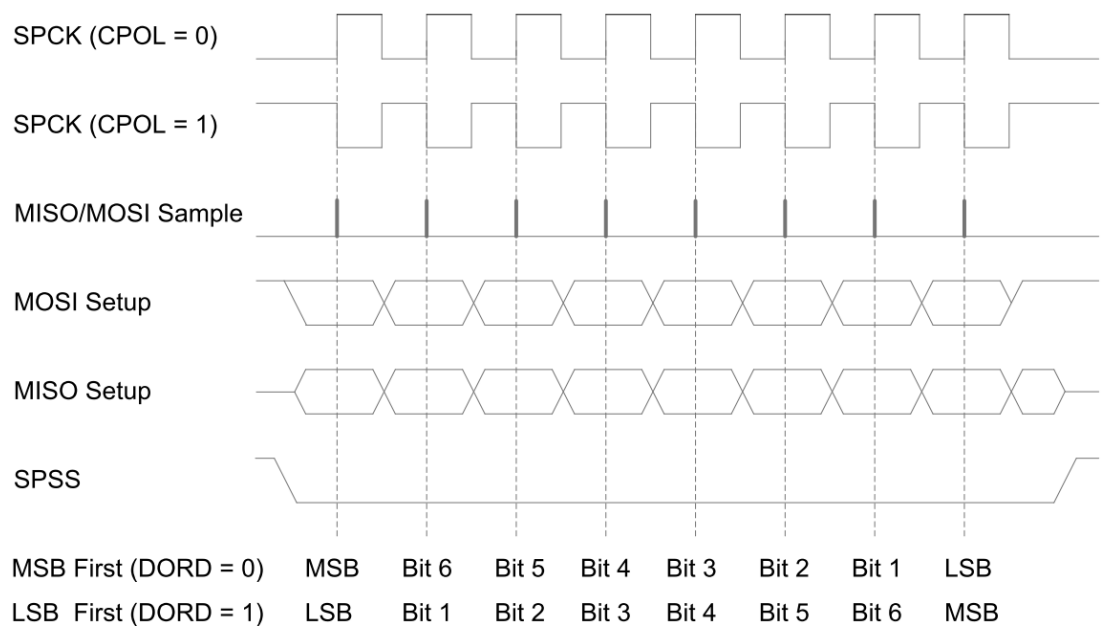


Рис15.2. Диаграмма передачи данных SPI, когда CPHA является «0»

Когда CPHA = 1, выборка данных и установка краев тактового сигнала следующие:

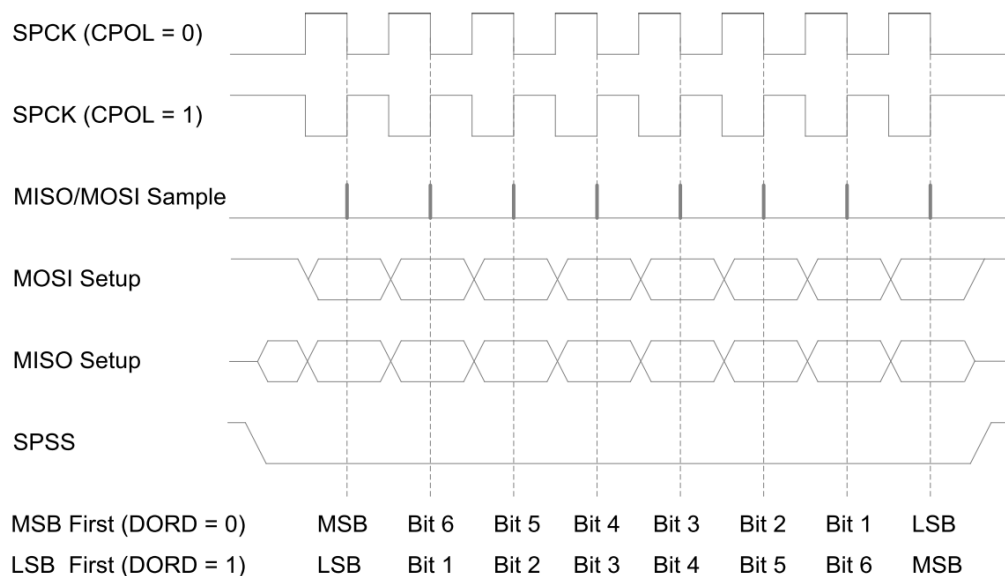


Рис15.3. Диаграмма передачи данных SPI, когда CPOL является «1»

В двухпроводном режиме MISO и MISO используются в качестве входа хоста. Время выборки данных все еще определяется режимом передачи данных. Режим выборки показан на рисунке ниже:

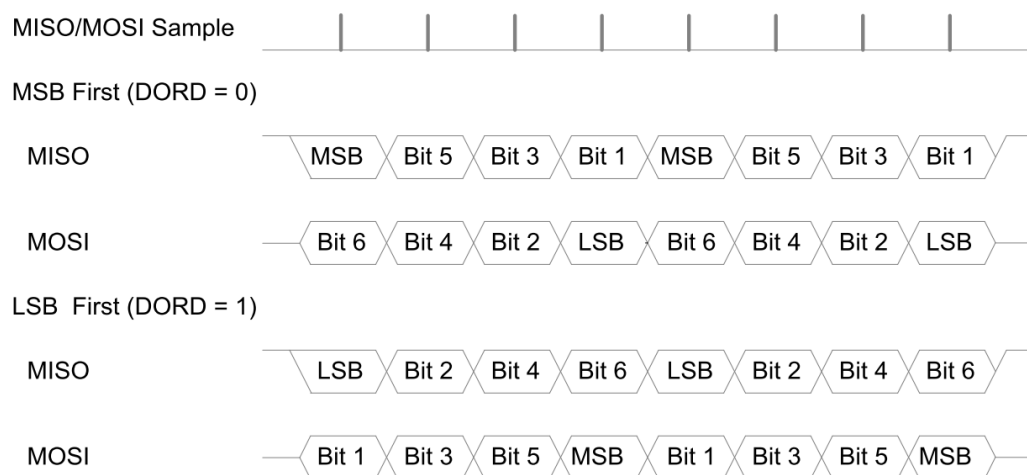


Рис15.3. Диаграмма передачи данных SPI, когда DUAL является «1» в режиме ведущего

15.5. 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.

15.6. 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

15.6.1. 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.

15.6.2. Инициализация ведомого устройства SPI

Процесс инициализации режима ведомого режима SPI выглядит следующим

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**.

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

SPCR - регистр управления SPI								
Адрес: 0x4C			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7	SPIE	Бит разрешения прерывания SPI. Прерывание SPI генерируется, когда бит SPIF в регистре SPSR установлен в «1» и глобальное прерывание включено. Когда бит SPIE установлен на «0», прерывание SPI отключено.						
6	SPE	Бит разрешения SPI. Когда бит SPE установлен в «1», модуль SPI включен. SPE должен быть установлен до любой операции SPI. Когда бит SPE установлен на «0», модуль SPI отключается.						
5	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.						
4	MSTR	Выбор Ведущего или Ведомого. Когда бит MSTR в «1», выбирается режим Ведущего. Когда бит MSTR в «0», выбирается режим ведомого. В главном режиме, когда вывод SPSS сконфигурирован как вход и выведен в «0», бит MSTR будет очищен и будет установлен SPIF в регистре SPSR. Пользователь должен сбросить MSTR в мастер-режим.						
3	CPOL	Бит контроля полярности часов. Когда бит CPOL установлен в «1», SPCK находится в состоянии ожидания. Когда бит CPOL установлен в «0», SPCK в режиме ожидания низкий.						
		CPOL	Исходный край		Конец края			
		0	Восходящий фронт		Падение фронт			
		1	Падение фронт		Восходящий фронт			
2	CPHA	Когда бит CPHA установлен в «1», начальная граница устанавливает данные и завершает данные выборки. Когда бит CPHA установлен в «0», данные выборки начального фронта и данные настройки конечного края.						
		CPHA	Исходный край		Конец края			
		0	Выборка		Установить			
		1	Установить		Выборка			
1	SPR1	SPR1 и SPR0 используются для выбора тактовой частоты для 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 обращаясь 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

Буферный регистр SPI								
Адрес: 0x4E			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	SPDR7	SPDR6	SPDR5	SPDR4	SPDR3	SPDR2	SPDR1	SPDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Бит	Имя	Определение бит
7:0	SPDR	SPI отправляет и принимает данные. Запись данных в SPDR записывается в регистр сдвига данных передачи, а чтение данных из SPDR считывает буфер данных приема.

16. 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

16.1. 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.

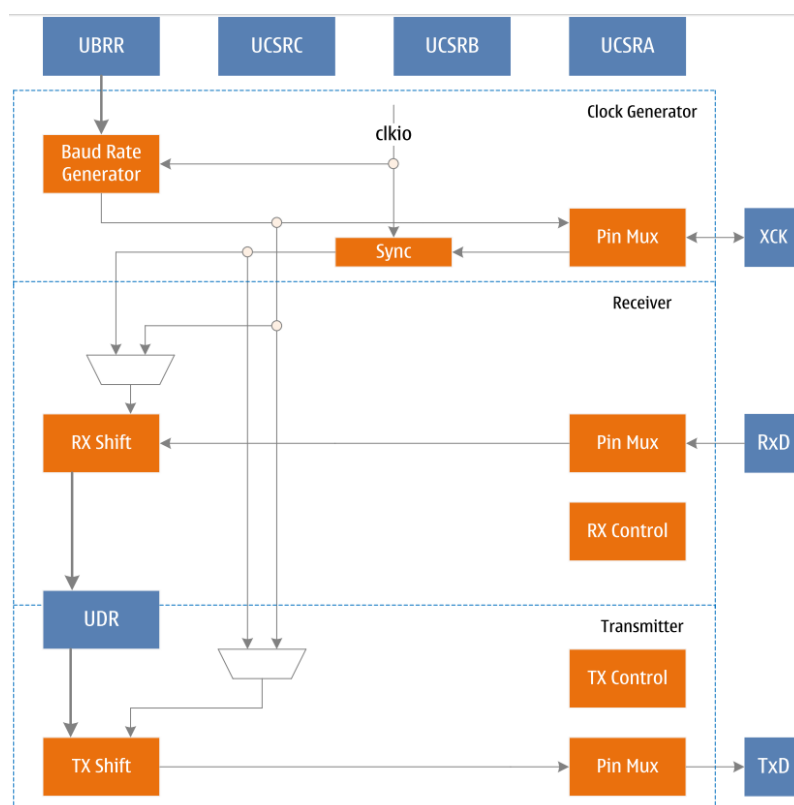


Рис. Блок-схема USART в режим работы SPI

16.2. 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:

$$BAUD = f_{sys} / (2 * (UBRR + 1))$$

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».

16.3. 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)

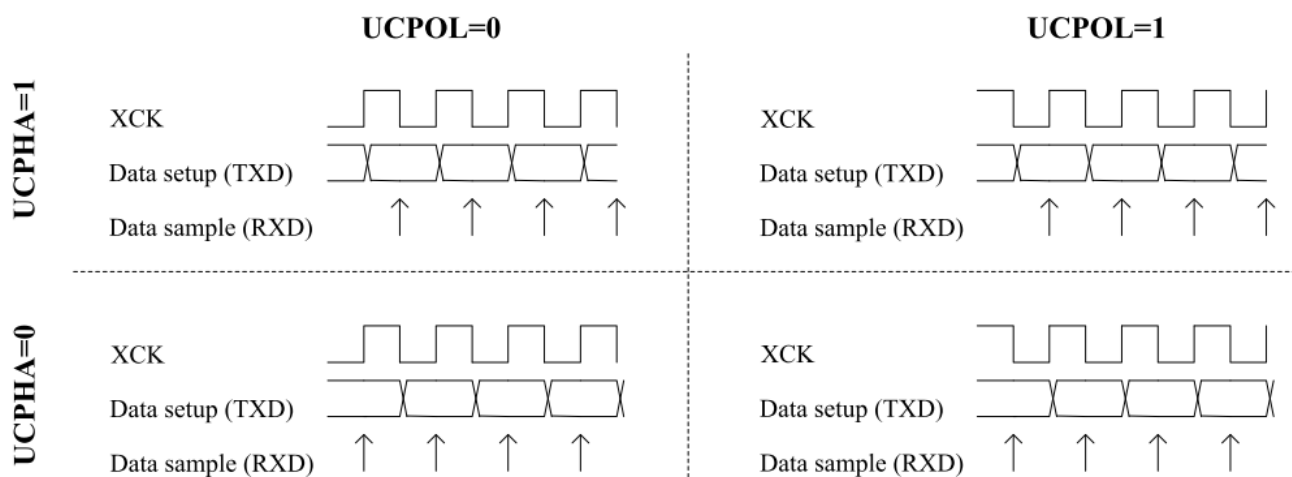


Рис. UCPHAN и UCPOLn временные диаграммы передачи данных.

16.4. 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.

16.5. 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 ХСК 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.

16.5.1. 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.

16.5.2. 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.

16.6. 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.

16.7. 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
UBRRL	0xC4	0x0	Младший байт скорости передачи данных USPI
UBRRH	0xC5	0x0	Старший байт скорости передачи данных USPI
UDR	0xC6	0x0	Регистр данных USPI

UCSRA - Регистр А управление и состояния USART SPI

UCSRA - Регистр А управление и состояния USART								
Адрес: 0xC0			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	RXC	TXC	UDRE	-	-	-	-	-
R/W	R	R/W	R	-	-	-	-	-
Бит	Имя	Определение бит						
7	RXC	Когда RXC в «1» - наличие непрочитанных данных в буфере приема. Когда значение RXC в «0», в буфере приема нет непрочитанных данных. Когда приемник отключен, буфер приема обновляется, в результате чего RXC = «0». Когда бит разрешения прерывания RXCIE в «1», RXC может использоваться для генерации прерывания конца приема.						
6	TXC	Когда TXC в «1» после передачи данных из регистра сдвига и когда буфер передачи пуст. TXC автоматически очищается, когда выполняется прерывание передачи, и его также можно очистить, записав «1» в TXC. TXC может использоваться для генерации прерывания передачи, когда бит разрешения завершения передачи, TXCIE, равен «1».						
5	UDRE	Когда UDRE в «1», это означает, что буфер данных передачи USART пуст, и данные могут быть записаны. Когда UDRE в «0», это означает, что буфер данных передачи USART заполнен и данные не могут быть записаны. UDRE может использоваться для создания пустого прерывания регистра данных, когда бит разрешения пустого регистра данных, UDRIE в «1».						
4:0	-	Зарезервировано						

UCSRB - Регистр В управление и состояния USART SPI

UCSRB - Регистр В управление и состояния USART								
Адрес: 0xC1			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	RXCIE	TXCIE	UDRIE	RXEN	TXEN	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	-	-	-
Бит	Имя	Определение бит						
7	RXCIE	Окончание приема - бит разрешения прерываний. Включить прерывание: RXC = «1» и отключить прерывание: RXC = «0». Если RXCIE в «1» и глобальное прерывание разрешено, прерывание конечного выхода USART может быть сгенерировано, когда в регистре UCSRA бит RXC в «1».						
6	TXCIE	Окончание передачи - бит разрешения прерываний. Включить прерывание TXC: = «1» и отключить прерывание TXC после очистки. Когда TXCIE = «1», глобальное прерывание включено и полное прерывание передачи USART может быть сгенерировано, если в регистре UCSRA TXC в «1».						
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						

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	UCPHA	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					
		3	Режим работы ведущего 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	UCPHA	Выбор фазы синхронизации. UCPHA Выбор выборки данных происходит в начале или в конце.						
		UCPHN	Изменение передачи данных		Выборка данных			
		0	Положительный фронт ХСК		Отрицательный фронт ХСК			
		1	Отрицательный фронт ХСК		Положительный фронт ХСК			
0	UCPOL	Выбор полярности тактов. UCPOL выбирает изменения данных, и образцы возникают на восходящих или падающих фронтах.						
		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								
Адрес: 0xC4			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	UBRR7	UBRR6	UBRR5	UBRR4	UBRR3	UBRR2	UBRR1	UBRR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	UBRR7:0	Младший байт регистра скорости передачи данных USART. Регистр скорости передачи USART содержит UBRRL и UBRRH, которые используются вместе, чтобы установить скорость передачи в бодах.						

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 UBRRH и UBRRL, которые используются вместе, чтобы установить скорость передачи в бодах. $UBRR = \{UBRR [11: 8], UBRRL\}$	
		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 битами 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.						

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 нижних 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 ХСК 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.

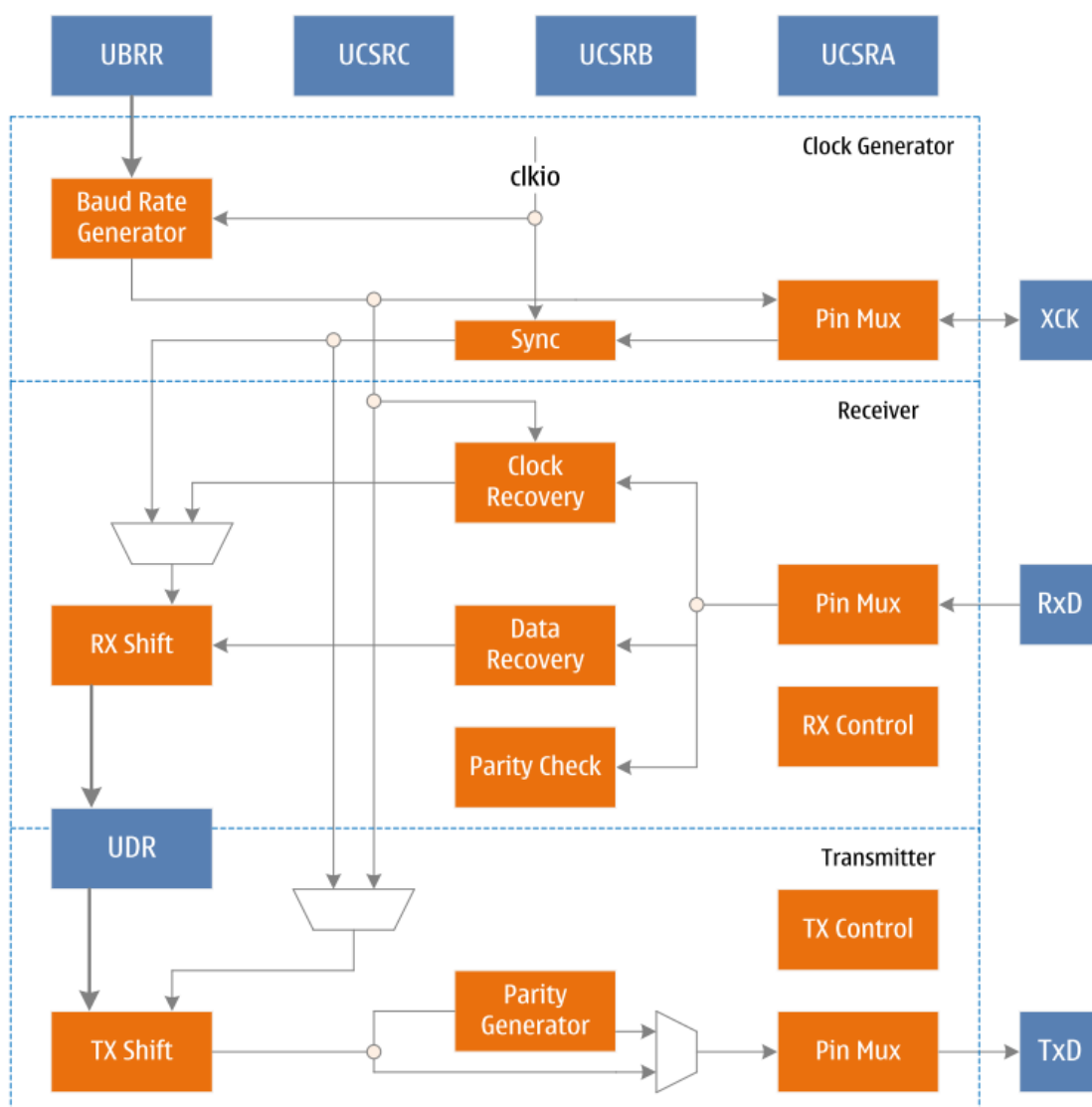


Рис.17.1 Структура USART

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 USCRA контролирует двойное разрешение скорости в асинхронном режиме.

Регистр направления данных (мультиплексированный с IO) вывода ХСК, 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 UBRR. 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:

1. Скорость передачи определяется как скорость передачи бит в секунду (бит / с);
2. BUAD - скорость передачи, f_{sys} - системные такты, а UBRR - комбинированное значение регистров скорости передачи данных UBRRH и UBRRL.

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 управляется внешним тактовым сигналом. Внешние такты используются передатчиком и приемником после прохождения через регистр синхронизации и детектор фронта. Этот процесс вводит две системные тактовые задержки. Поэтому максимальная тактовая частота внешнего ХСК ограничена следующей формулой:

$$f_{\text{ХСК}} < f_{\text{sys}} / 4$$

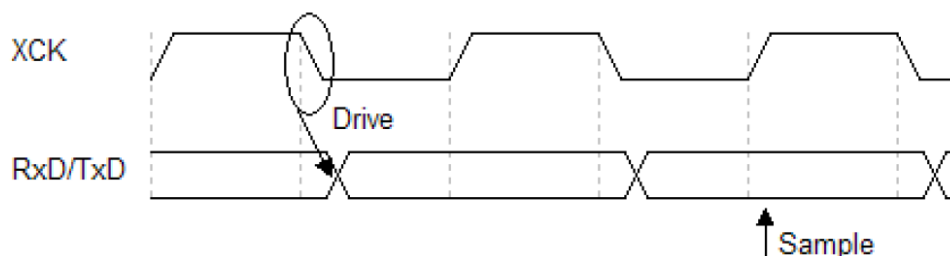
Обратите внимание, что f_{sys} имеет решение о стабильности для системных тактов. Чтобы предотвратить потерю данных из-за частотного дрейфа, рекомендуется зарезервировать достаточный запас.

17.6. Синхронные такты.

В синхронном режиме вывод ХСК 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), противоположно фронту тактового сигнала, используемому для изменения выходного конца данных.

UCPOL = 1



UCPOL = 0

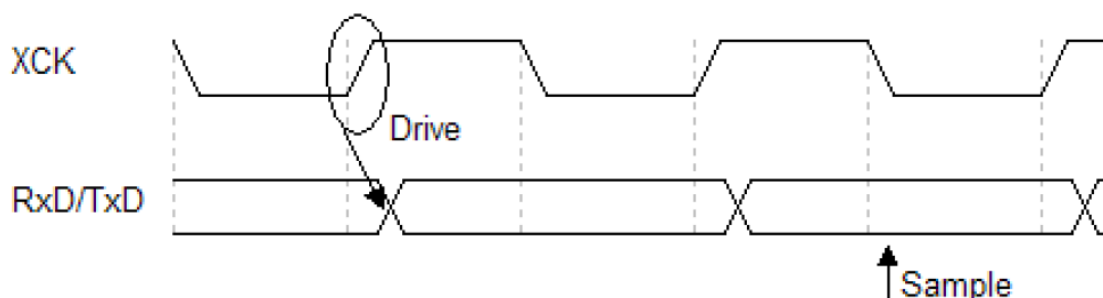


Рис. Время ХСК в синхронном режиме

Как показано на рисунке выше, когда значение UC POL равно «1», выход данных изменяется на заднем фронте ХСК, и выборка данных выполняется по нарастающему фронту ХСК. Когда значение UC POL равно «0», выход данных изменяется на передний фронт ХСК. Сэмплирование данных происходит на заднем фронте ХСК.

17.7. Формат кадра.

Кадр последовательных данных состоит из слова данных плюс биты синхронизации (биты начала и окончания) и бит четности для исправления ошибок.

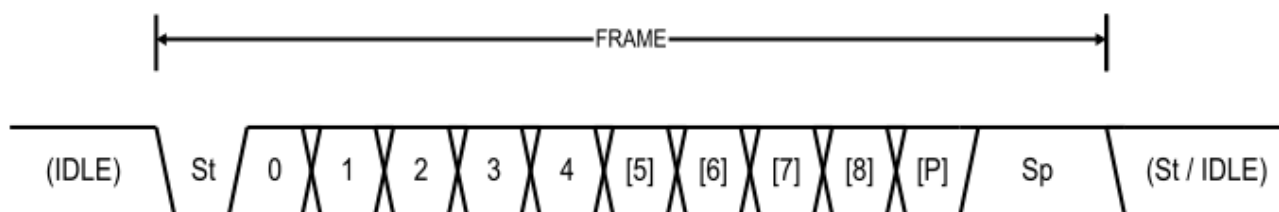
USART принимает следующие 30 комбинаций форматов фреймов данных:

- 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 не обязательны.



Структура кадра USART

St	Стартовый бит, всегда низкий.
(n)	Биты данных (от 0 до 8).
P	Бит четности. Может быть нечетным или четным.
Sp	Стор 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 «исключительное или» всех битов данных. Если используется нечетная четность, результат является

исключительным или инвертированным. Отношение между битом четности и битами данных выглядит следующим образом:

$$P_{\text{even}} = d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 0$$

$$P_{\text{odd}} = d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 1$$

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, считываются 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 ХСК, 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

Передачик 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 регистра 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 TXC устанавливается, когда весь кадр данных смещен из регистра сдвига передачи, и нет новых данных в регистре передачи.

Когда бит разрешения прерывания передачи TXCIE (и разрешено глобальное прерывание) 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, такты на выводе ХСК используются как такты 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 ХСК 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 флажка 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. Стоповый 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 контроля 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 нижних 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 большой 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 такты 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:

$$\text{Ошибка [\%]} = (f_{\text{sys}} / (16 * (\text{UBRR} + 1)) / \text{Baud} - 1) * 100\%$$

Режим двойной скорости:

$$\text{Ошибка [\%]} = (f_{\text{sys}} / (8 * (\text{UBRR} + 1)) / \text{Baud} - 1) * 100\%$$

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

2. Основной процессор отправляет адресный кадр, и все подчиненные процессоры получают этот кадр. Ведомого процессор устанавливает бит RXC регистра UCSRA нормально.

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.

Возможные форматы кадров, содержат от 5 до 8 бит данных, но непрактичны, потому что приемник должен переключаться между форматами n и $n+1$.

17.13. Регистры Описание

UCSRA - Регистр А управление и состояния USART								
Адрес: 0xC0			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPME
R/W	R	R/W	R	R	R	R	R/W	R/W
Бит	Имя	Определение бит						
7	RXC	Когда RXC в «1» - наличие непрочитанных данных в буфере приема. Когда значение RXC в «0», в буфере приема нет непрочитанных данных. Когда приемник отключен, буфер приема обновляется, в результате чего RXC = «0». Когда бит разрешения прерывания RXCIE в «1», RXC может использоваться для генерации прерывания конца приема.						
6	TXC	Когда TXC в «1» после передачи данных из регистра сдвига и когда буфер передачи пуст. TXC автоматически очищается, когда выполняется прерывание передачи, и его также можно очистить, записав «1» в TXC. TXC может использоваться для генерации прерывания передачи, когда бит разрешения завершения передачи, TXCIE, равен «1».						
5	UDRE	Когда UDRE в «1», это означает, что буфер данных передачи USART пуст, и данные могут быть записаны. Когда UDRE в «0», это означает, что буфер данных передачи USART заполнен и данные не могут быть записаны. UDRE может использоваться для создания пустого прерывания регистра данных, когда бит разрешения пустого регистра данных, UDRIE в «1».						
4	FE	Флаг ошибки кадра. Когда FE в «1», это означает, что данные, полученные						

1	RXB8	Прием 8-го бита данных. Когда длина кадра данных составляет 9 бит, RXB8 является наивысшим битом полученных данных. Прочитайте 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					
		3	Режим работы ведущего 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	Включить нечетность					
3	USBS	Выбор стоп-биты. Количество бит: USBS = 0 – 1 стоп-бит; USBS = 1 – 2 стоп-бита						
2:1	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					
		6	Зарезервировано					
7	9 бит							
0	UCPOL	Бит выбора полярности бит. В режиме синхронной работы USART UCPOL устанавливает взаимосвязь между изменением выходных данных и синхронизацией тактов синхронизации ХСК входных данных. Очистите этот бит независимо от UCPOL в асинхронном режиме работы						
		UCPOL	Изменение передачи данных		Выборка данных			
		0	Положительный фронт ХСК		Отрицательный фронт ХСК			

UBRRL - Младший байт скорости передачи данных USART

UBRR1 - младший байт скорости передачи данных USART								
Адрес: 0xC4			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	UBRR7	UBRR6	UBRR5	UBRR4	UBRR3	UBRR2	UBRR1	UBRR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	UBRR7:0	Младший байт регистра скорости передачи данных USART. Регистр скорости передачи USART содержит UBRR1 и UBRR0, которые используются вместе, чтобы установить скорость передачи в бодах.						

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 UBRR11 и UBRR8, которые используются вместе, чтобы установить скорость передачи в бодах. $UBRR = \{UBRR [11: 8], UBRR8\}$						
		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				$BAUD = f_{sys} / (2 * (UBRR + 1))$		

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	R/W
Бит	Имя	Определение бит						
7:0	UDR	<p>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.</p>						

18. 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

18.1. 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.

18.2 Терминология TWI

В этом разделе часто появляются следующие определенные термины:

- Ведущее Устройство - которое запускает и останавливает передачу.
- Ведущий - 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

18.3. 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.

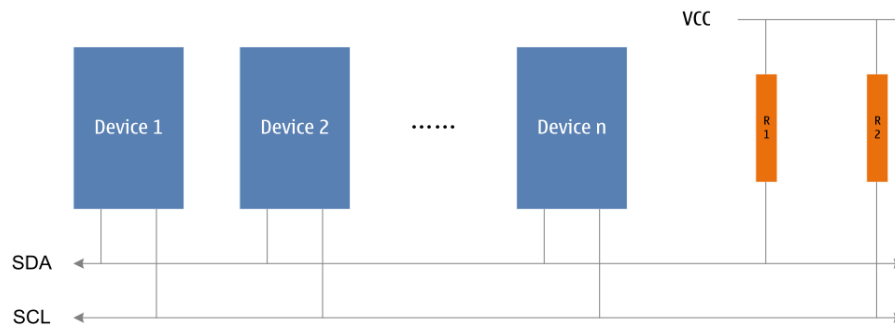


Рис.18.1. Схема подключения шины TWI

18.4. 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.

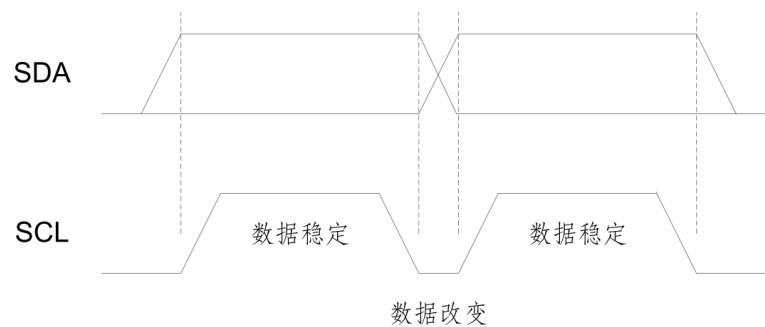


Рис.18.2. Диаграмма достоверности данных TWI

18.4.1 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.

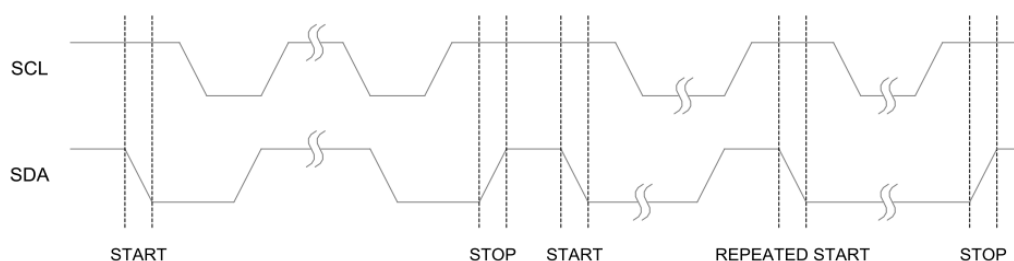


Рис.18.3. Состояния START, REPEATED START и STOP

18.4.2. 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 ширококестельных вызовов, и все адреса в формате «1111xxxx» должны быть зарезервированы для будущего использования, другие подчиненные адреса могут свободно назначаться разработчиком.

Когда происходит ширококестельный вызов, все подчиненные устройства должны отвечать, ставить линию SDA на низкий уровень в течение периода ACK.

Функция ширококестельного вызова 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 ширококестельный вызов, получают следующий пакет данных.

Следует отметить, что не имеет смысла посылать адрес ширококестельного вызова плюс бит READ, потому что произойдет столкновение шины, если несколько подчиненных устройств одновременно отправляют разные данные.

Формат адресного пакета показан ниже:

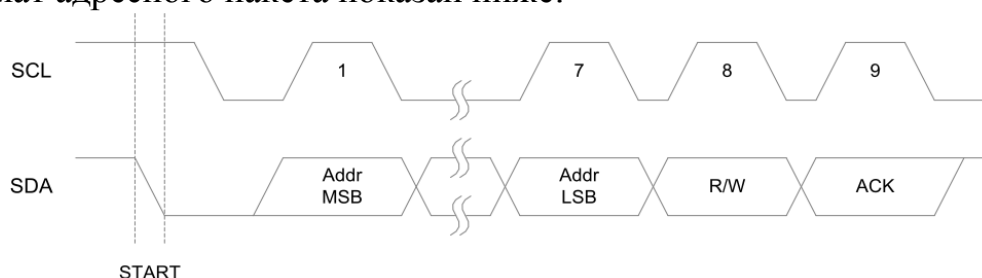


Рис.18.4. Диаграмма формата адресного пакета в формате TWI

18.4.3. 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, посылается 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:

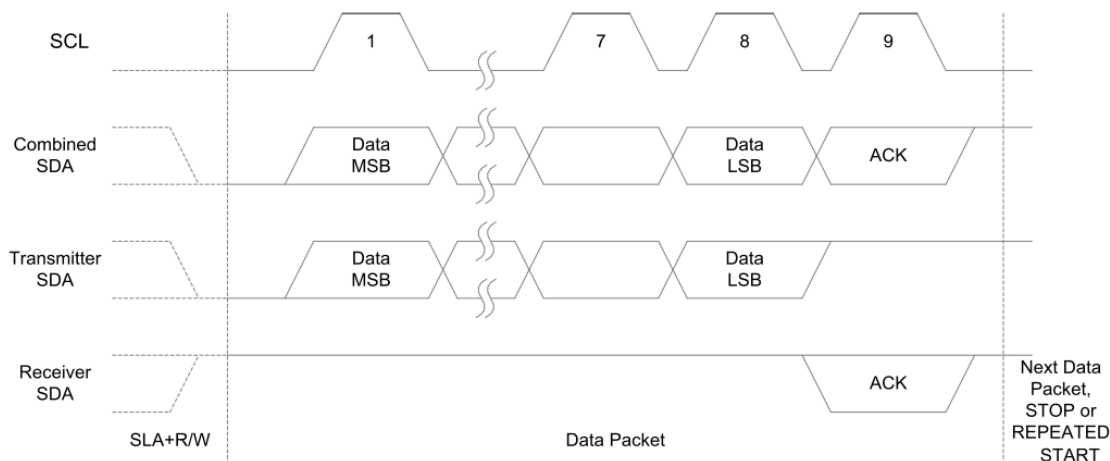


Рис.18.5. Диаграмма формата пакета данных TWI

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.

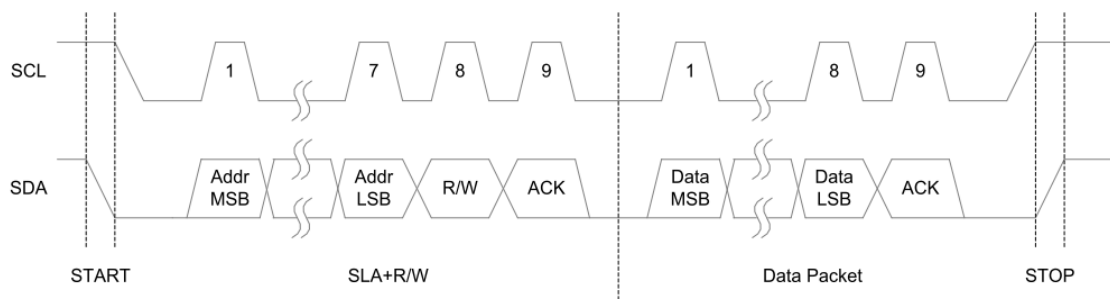


Рис.18.6. Типичная передача TWI

18.4.4. 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 хосты могут использовать разные частоты SCL. Для обеспечения последовательного перевода, вы должны разработать последовательные тактовые синхронизируются программы хоста. Это облегчит процесс арбитража. Линия шины и функция для решения вышеуказанных проблем. Все хосты тактовые линии для получения композиции с часами высокой времени, равный мастер-такты во всех кратчайшим, равную его низком уровне во всех самый длинный хост-такты. Все хосты прослушивания SCL, когда объединено SCL такты 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:

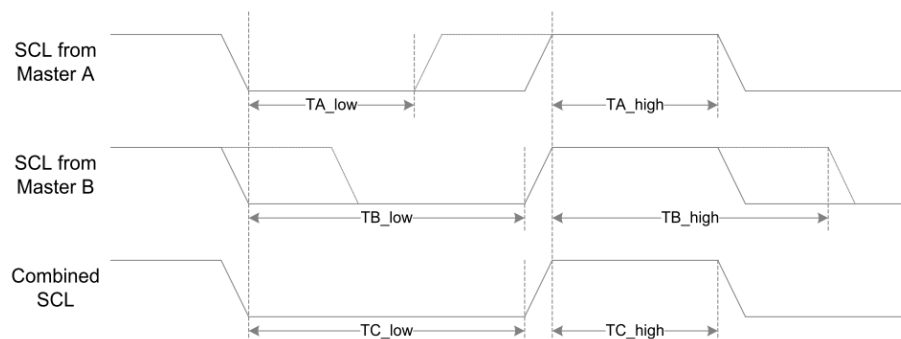


Рис.18.7. Временная диаграмма синхронизации синхронизации SCM

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.

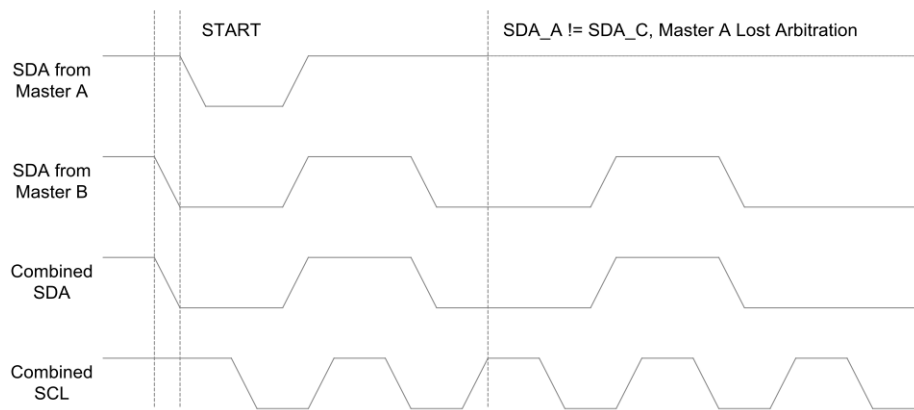


Рис.18.8. Арбитраж между двумя хостами

- 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.

18.5 Обзор модуля TWI

Структура модуля TWI показана на рисунке ниже.

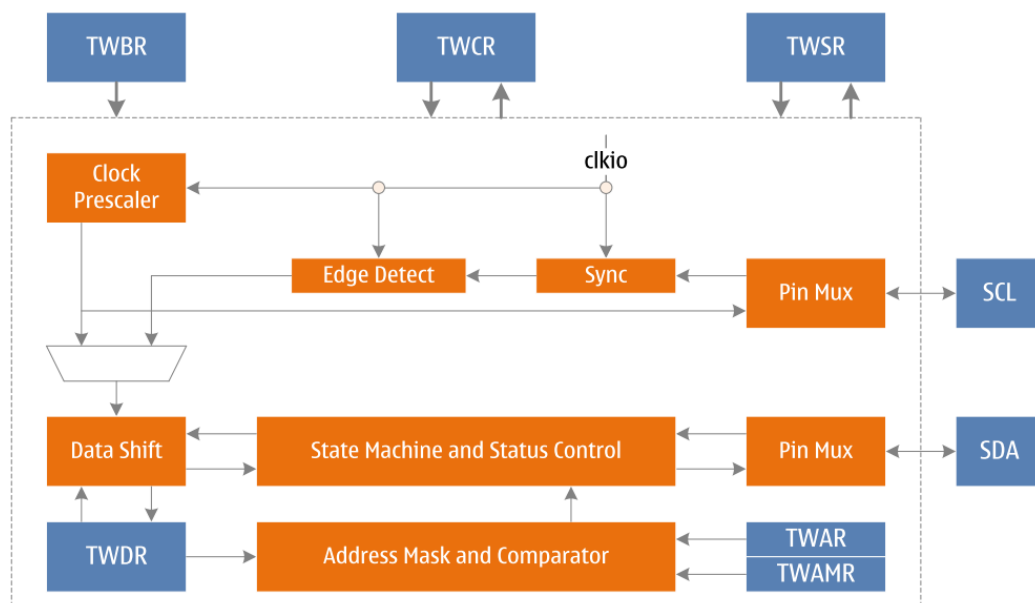


Рис.18.9 Структура модуля TWI

Модуль TWI в основном включает в себя генератор скорости передачи битов, блок интерфейса шины, компаратор адресов и блок управления. См. Следующее подробное описание.

18.5.1. 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.

18.5.2. 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, принятое 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.

18.5.3. 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 ширококестельным 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 на шине, пробуждает MCU из спящего режима.

18.5.4. Блок управления

Блок управления отвечает за прослушивание шины и отвечает в соответствии с настройками TWCR. Флаг прерывания TWI, TWINT, будет установлен, когда на шине 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 по шине 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 ширококестельный режим)
- После обращения к ведомому после получения STOP или REPEATED START
- **Шина.** Если ошибка шины, вызванная незаконным состоянием START или STOP

18.6. 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.

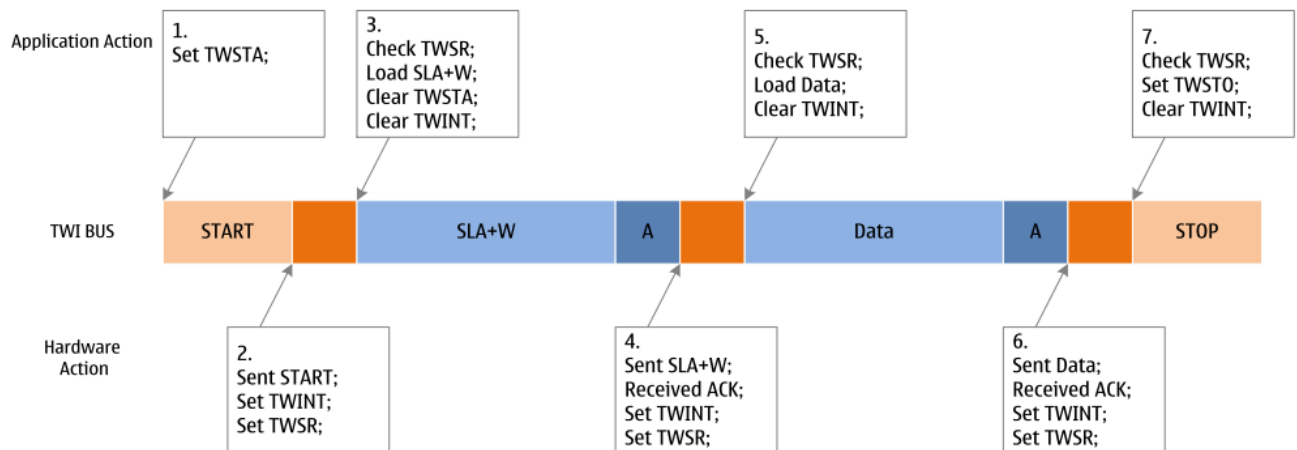


Рис.18.10. Типичная схема процесса передачи данных TWI

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.

18.6.1. 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 бит байтов 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».

18.6.2. 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 посылает 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	Посылает 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	Посылает 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. Получен ACK	Загрузка данных	0	0	1	X	Посылает 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	1	1	1	X	Отправляет STOP,

		операции					сброс флага TWSTO; START отправлен
0x30	Байты данных отправлены. Получен NACK	Загрузка данных	0	0	1	X	Посылает 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:

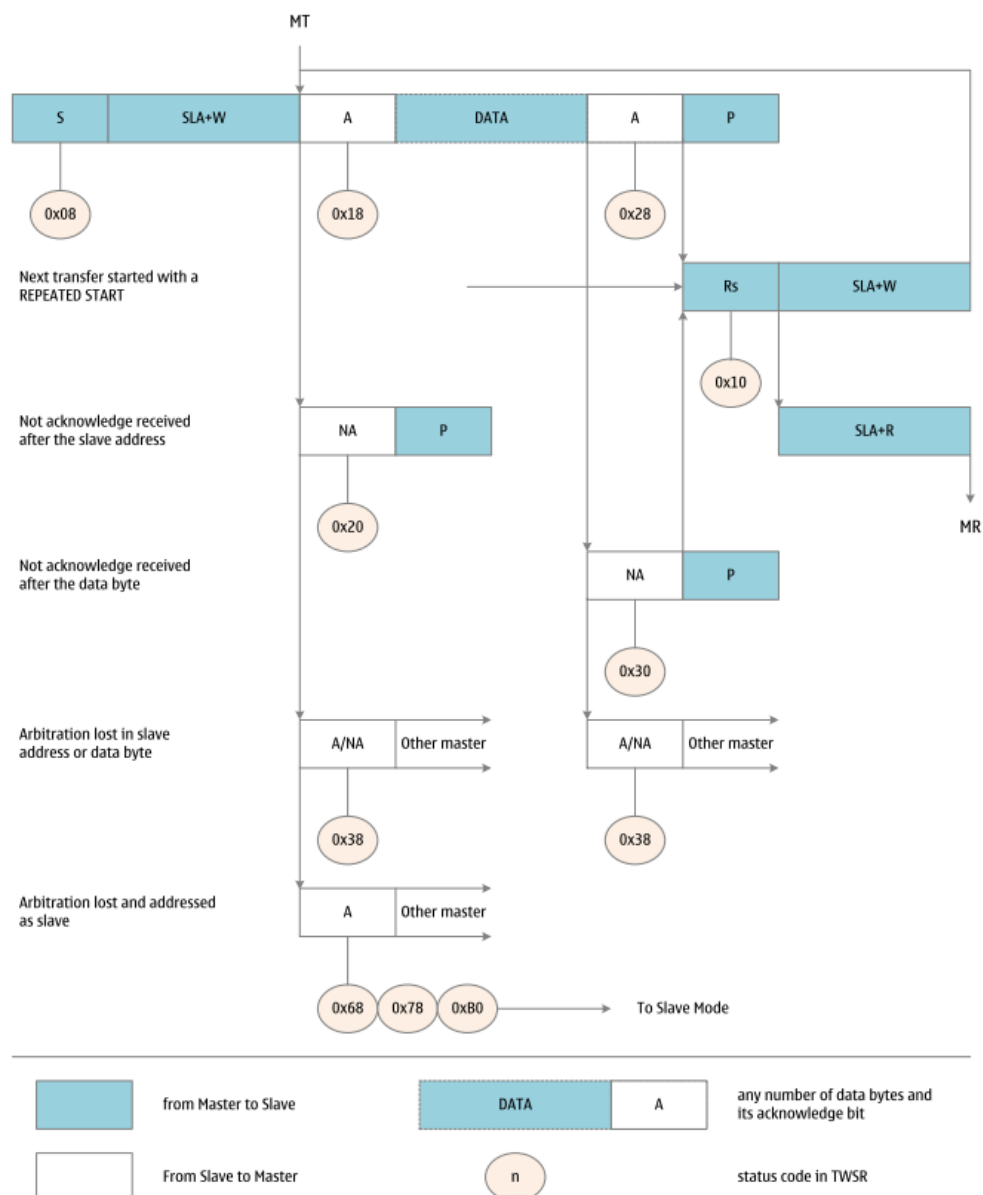


Рис.18.11. Формат режима передачи и диаграмма состояний

18.6.3. 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 посылает 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, АСК отправлен	Загрузка данных	0	0	1	0	Получат данные, NACK будет отправлен
		Нет операции	0	0	1	1	Получат данные, АСК 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; START отправлен

Формат и состояние принимающего режима, как показано ниже:

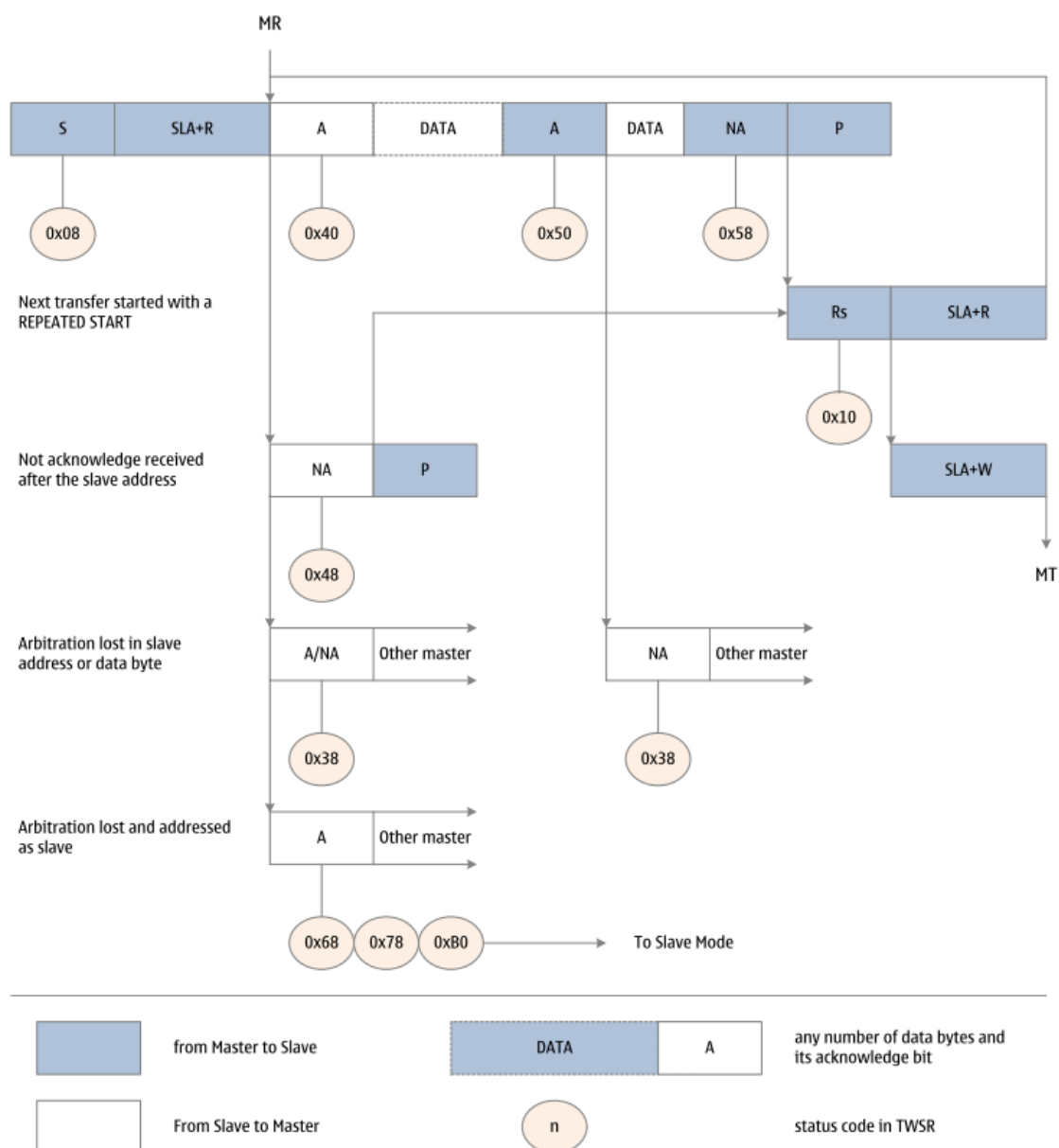


Рис.18.12. Формат принимающего режима и диаграмма состояний

18.6.4. 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 TWA_R и TWCR.

TWA_R необходимо установить следующим образом:

TWA ₆	TWA ₅	TWA ₄	TWA ₃	TWA ₂	TWA ₁	TWA ₀	TWGCE
Адрес устройства подчиненного устройства							

Старшие 7 бит TWA_R - 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 TWA_R 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 широковещате льный 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 неак- тивный 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 ширококешательную 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 ширококешательную передачу, шина отправит START в режиме ожидания
0x90	Получены ширококешательные 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 ширококешательные 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 ширококешательную 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 ширококешательную 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:

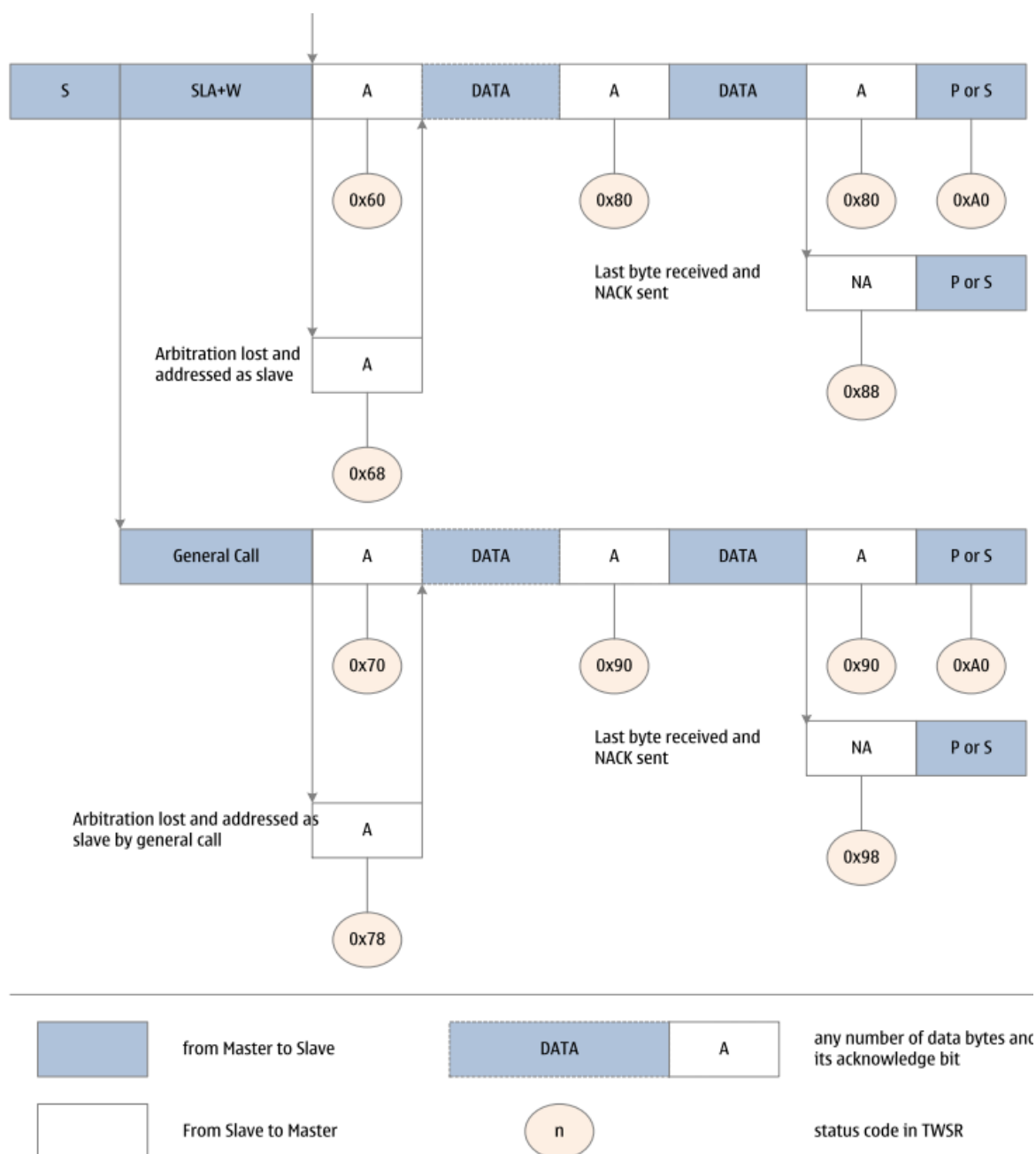


Рис.18.13. Формат режима приема и состояния устройства

18.6.5. 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 TWA_R и TWC_R. TWA_R необходимо установить следующим образом:

TWA ₆	TWA ₅	TWA ₄	TWA ₃	TWA ₂	TWA ₁	TWA ₀	TWGCE
Адрес устройства подчиненного устройства							

Старшие 7 бит TWA_R - 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.

TWC_R 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 TWA_R и TWC_R 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	Посылает 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	Посылает 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	Посылает 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, АСК 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:

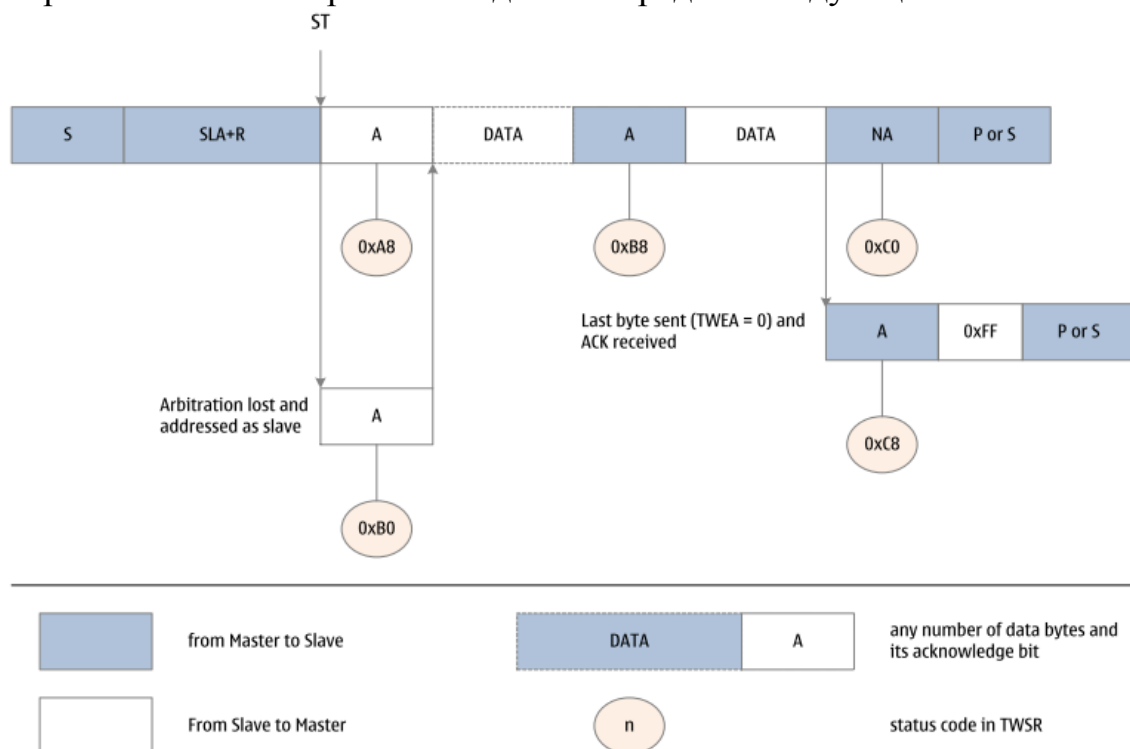


Рис.18.14. Формат режима слайдов и диаграмма состояний

18.6.6. Другие состояния

Существует два кода состояния, которые не имеют соответствующих определений состояния TWI, как показано в следующей таблице:

Таблица Другие кодов 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 прочитайте 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:

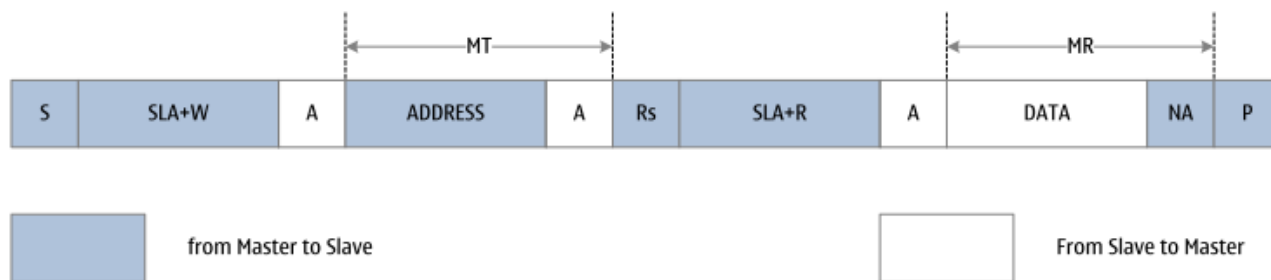


Рис.18.15. Объединяет несколько режимов TWI для доступа к серийным EEPROM-изображениям

18.7. 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 дожидаясь, пока шина отправит новый сигнал START, в

зависимости от работы программного обеспечения.

На следующем рисунке описывается процесс арбитража шины:

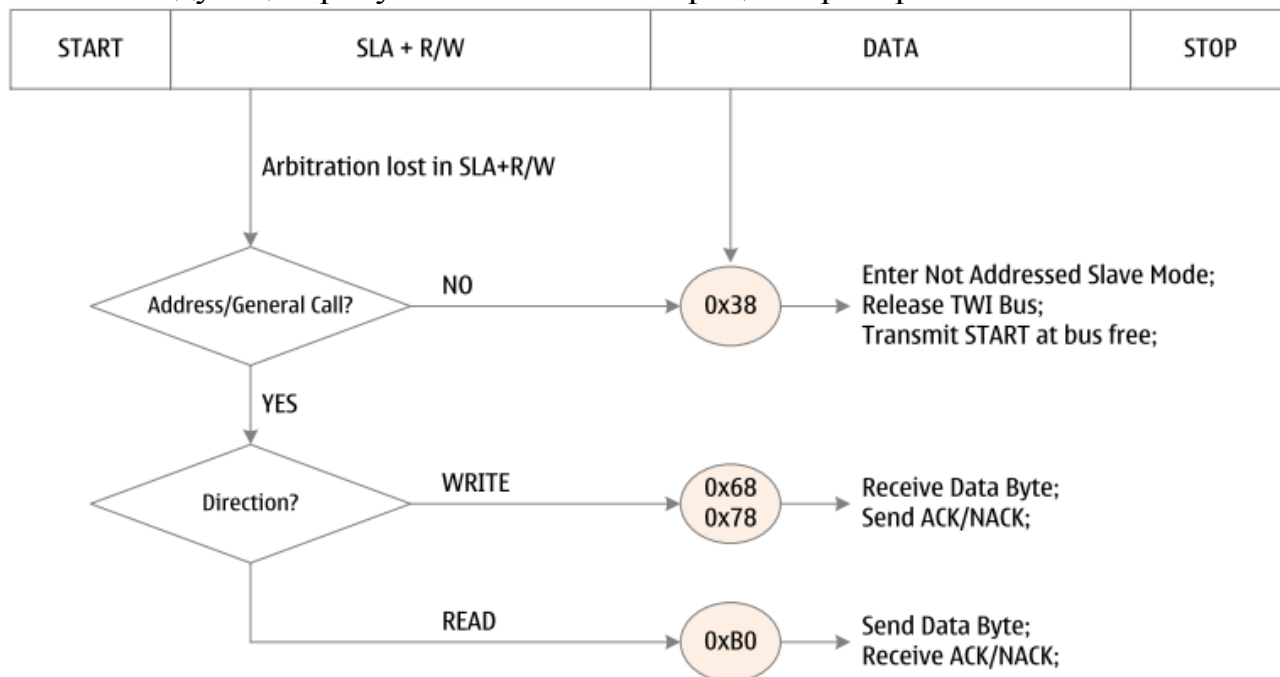


Рис.18.16. Схема арбитражного процесса шины

18.8. Регистры Описание

Список регистров TWI

Регистр	Адрес	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								
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]	<p>Флаг состояния TWI. 5-битный TWS реагирует на логику TWI и состояние шины. Различные значения статуса имеют разные значения. Подробнее см. Описание режима работы TWI. 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.</p>						
2	-	Зарезервировано						
1	TWPS1	<p>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.</p>						
0	TWPS0	TWPS[1:0]			Коэффициент предделения			
		0			1			
		1			4			
		2			16			
		3			64			

TWAR - регистр адресов TWI								
Адрес: 0xBA			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	TWAR6	TWAR5	TWAR4	TWAR3	TWAR2	TWAR1	TWAR0	TWGCE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:1	TWA[6:0]	Битовый адрес ведомого устройства TWI. TWA - это ведомый адрес TWI. Когда TWI работает в подчиненном режиме, TWI будет реагировать на этот адрес. Режим ведущего не требует этого адреса. Однако в системах с несколькими ведущими также необходимо установить для подчиненного адреса другие ведущего.						
0	TWGCE	Бит управления TWI разрешения ширококвещательного распознавания. Когда бит TWGCE установлен в «1», распознавание ширококвещательной передачи TWI включено. Когда бит TWGCE установлен в «0», распознавание ширококвещательной передачи шины TWI отключено. Когда TWGCE установлен, и принятый адресный фрейм равен 0x00, модуль TWI будет отвечать на эту передачу по шине.						

[illegible]

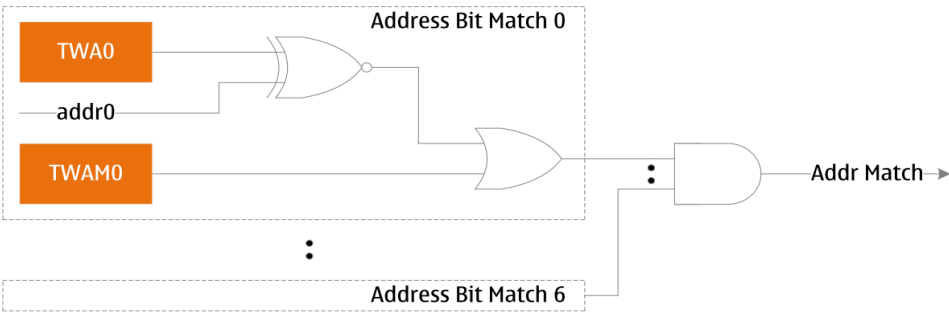
Бит	Имя	Определение бит
7:0	TWDR	Регистр данных TWI. TWD является следующим байтом на передаваемой шине или последним байтом, который был только что получен на шине.

TWCR - Регистр управления TWI

TWCR - регистр управления TWI								
Адрес: 0xBC			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
R/W	R/W	R/W	R/W	R/W	R	R/W	-	R/W
Бит	Имя	Определение бит						
7	TWINT	<p>Флаг прерывания TWI.</p> <p>Когда TWI завершает текущее задание и хочет, чтобы прикладное программное обеспечение вмешалось, аппаратное обеспечение установит бит TWINT. Если глобальное прерывание установлено и бит TWIE установлен, будет генерироваться прерывание TWI, и MCU выполнит процедуру обслуживания прерывания TWI. Когда установлен флаг TWINT, будет увеличен низкий уровень сигнала SCL.</p> <p>Флаг TWINT может быть очищен только путем записи «1» в этот бит. Аппаратное обеспечение не автоматически очищает этот бит, даже если выполняется процедура обслуживания прерывания.</p> <p>Также обратите внимание, что очистка этого бита немедленно включит операцию TWI. Поэтому перед очисткой бит TWINT должен быть завершен доступ к регистрам TWAR, TWAMR, TWSR и TWDR.</p>						
6	TWEA	<p>TWI разрешает бит подтверждения.</p> <p>Бит TWEA управляет генерацией импульса ответа. Когда бит TWEA установлен на «1», и выполняется одно из следующих условий, на шине TWI будет генерироваться импульс ответа:</p> <ol style="list-style-type: none"> 1) Принят ведомый адрес устройства; 2) Широковещательный вызов принимается при установке TWGCE; 3) Один байт данных принимается в режиме приема основного приема или подчиненного устройства. <p>Когда бит TWEA установлен на «0», устройство временно отключается от шины TWI. После установки устройства возобновляется распознавание адресов.</p>						
5	TWSTA	<p>Бит начального состояния TWI.</p> <p>Бит TWSTA необходимо установить, когда CPU хочет стать ведущим на шине TWI. Аппаратное обеспечение обнаружит, доступна ли шина, и когда шина свободна, он начнет работу на шине. Когда шина не работает, TWI ждет, пока не будет обнаружено условие остановки, а затем генерирует начальное состояние, чтобы объявить себя желающим стать ведущим. Программное обеспечение должно очистить бит TWSTA после отправки исходного состояния.</p>						
4	TWSTO	<p>Бит контроля состояния остановки TWI.</p> <p>Когда бит TWSTO установлен в «1» в режиме мастера, TWI генерирует условие остановки на шине и затем автоматически очищает бит TWSTO.</p> <p>В режиме ведомого, установка бит TWSTO в «1» может привести к восстановлению TWI из условия ошибки. Это не приведет к возникновению условия останова, но приведет к тому, что TWI вернется к определенному неадресационному подчиненному режиму и отпустит сигнальные линии SCL и SDA в состояние с высоким импедансом.</p>						
3	TWWC	<p>Конфликтный флаг записывает TWI.</p> <p>Когда флаг TWINT в «0», в регистре TWDR устанавливает флаг TWWC.</p> <p>Когда флаг TWINT в «1», запись «1» в регистр TWDR очищает флаг TWWC.</p>						
2	TWEN	<p>Бит управления разрешен 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. Когда бит TWEN в «0», интерфейсный модуль TWI отключается и все передачи прекращаются, включая текущие операции.
1	-	Зарезервировано
0	TWIE	Бит управления прерыванием TWI. Когда бит TWIE установлен в «1», и глобальное прерывание установлено, запрос прерывания TWI активируется всякий раз, когда флаг TWINT в «1».

TWAMR - Регистр масок адресов TWI

TWAMR - регистр масок адресов TWI								
Адрес: 0xBD			Значение по умолчанию: 0xFF					
Бит	7	6	5	4	3	2	1	0
Имя	TWAR6	TWAR5	TWAR4	TWAR3	TWAR2	TWAR1	TWAR0	TWGCE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:1	TWAM[6:0]	<p>Бит управления маскировкой адреса TWI.</p> <p>TWAM - это 7-битное управление маскировкой адреса подчиненного устройства TWI. Каждый бит TWAM используется для маскировки (отключения) соответствующего бита адреса в TWAR. Когда бит маски установлен, логика совпадения адресов будет игнорировать сравнение полученного бита адреса с соответствующим битом в TWA. На следующем рисунке показаны детали логики согласования адресов.</p> 						
0	-	Зарезервировано						

19. Операционный Усилитель/Компаратор 0

- Точность сравнения 12 мВ
- Поддержка двухстороннего аналогового входа
- Поддержка мультиплексированного выхода с АЦП
- Поддерживает вывод из внутренних ОРАМР операционных усилителей
- Поддержка внутреннего входного напряжения 1.25 / 2.56V
- Встроенный 8-разрядный ЦАП
- Защита от перегрузки по току/перенапряжению достигается с помощью схемы ОУ
- Программируемое управление выходным фильтром
-

19.1. Обзор

Модуль операционного усилителя / компаратора объединяет конфигурируемый предусилитель входного режима и аналоговый компаратор со встроенным 8-разрядным ЦАПом.

Операционный усилитель поддерживает настраиваемые режимы ввода с фиксированным прямым входом (12х усилением), инверсным входом (11х усилением) усилителя, который может использоваться с аналоговым компаратором и 8-разрядным ЦАП для достижения гибкого обнаружения сигнала.

Выход 0 (0-ампер) также может быть обработан более точно аналого-цифровым преобразователем (АЦП), аналоговым компаратором.

Величина прямого электрода компаратором сравнивается со значением инверсного электрода.

Когда напряжение на положительном электроде выше напряжения на отрицательном электроде, устанавливается аналоговый выход АСО компаратора.

Когда уровень АСО изменяется, фронт сигнала может использоваться для запуска прерывания.

Выходной сигнал АСО также может использоваться для инициирования входного захвата счетчика 1 таймера и для управления выходом ШИМ, генерируемым таймером.

Аналоговый компаратор интегрированный внутри с 8-битным цифро-аналоговым преобразователем (ЦАП), внутреннее опорное напряжение может быть разбита на 256 различных уровней опорного напряжения.

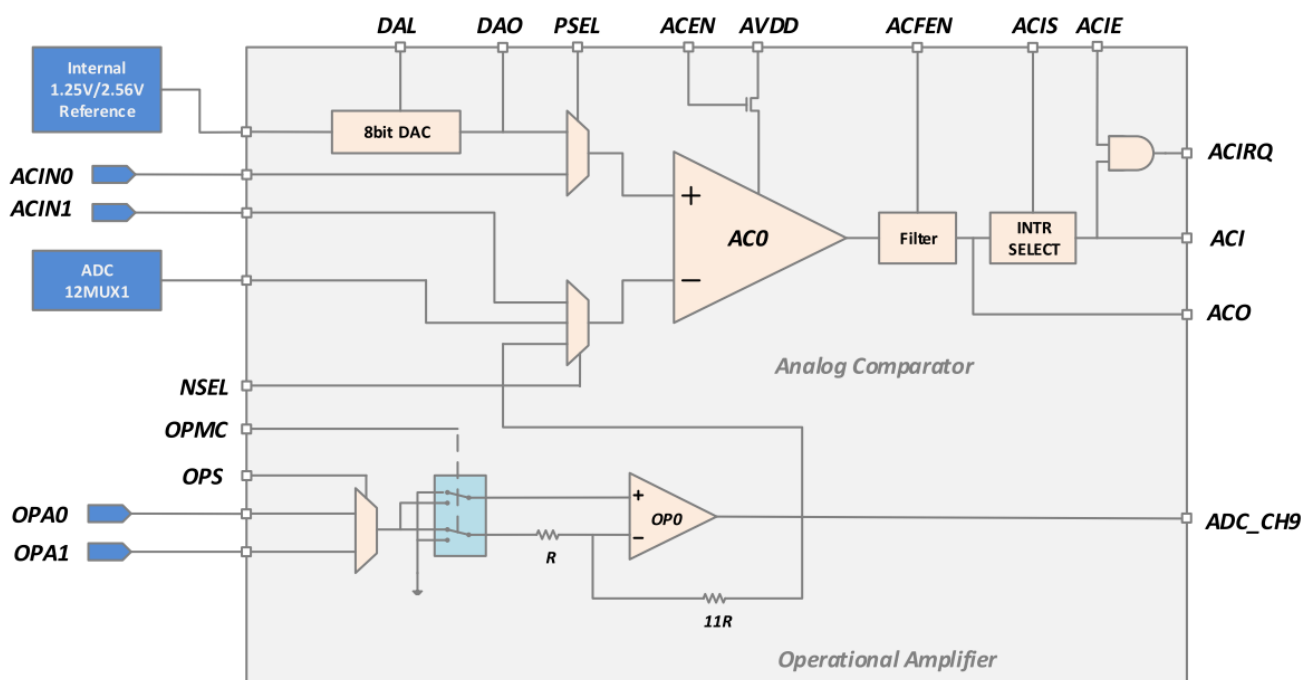


Рис.19.1. Структурная схема модуля операционного усилителя / компаратора 0

19.2. Входной коммутатор

Входной коммутатор OPA0 отвечает за мультиплексированный вход аналоговых входных каналов, переключение режима прямого/инверсного входа и усиление входного сигнала.

Усиленный сигнал ОРА0 делится на два канала, которые соответственно подключены к входному каналу АЦП и отрицательному входу аналогового компаратора (АС). То есть сигналы, обрабатываемые операционными усилителями, могут обрабатываться непосредственно с помощью компараторов или они могут быть преобразованы АЦП для получения точных результатов, так что могут быть реализованы более сложные функции.

ОРА0 всей аналоговой частью работы под контролем ОРА0 контроллера; ОРА0 аналоговая часть во время переключения ввода или изменений режима ввода, выход которого есть стабильный процесс, контроллер ОРА0 отвечает за генерацию контроллера синхронизации, чтобы избежать неустойчивости на выходе. Сценарные схемы имеют эффект.

Предшествующий операционный усилитель поддерживает два аналоговых входных канала: ОРА0/1. Два канала могут регулярно переключаться программным обеспечением или автоматически переключаться через контроллер ОРА внутри чипа.

Время переключения канала управляется 8-разрядным таймером. Источник синхронизации таймера может быть выбран в качестве системного тактового сигнала или 32-битной (1 МГц) внутренней RC32M, что обеспечивает более гибкий цикл переключения каналов и отвечает требованиям различных приложений.

После операционного усилителя включить функцию, чтобы установить зарегистрировать DIDR1 ОУ входной контакт, связанных управляющих битов, чтобы избежать влияния функции цифровой секции ввода/вывода на операционный усилитель аналоговых входных каналов. Подробности см. В разделе определения регистров этой главы.

19.2.1. Коммутатор переключения каналов усилителя

В режиме переключения автоматического переключения таймера во избежание помех, возникающих при переключении каналов, действие переключения канала связано с выходным фильтром аналогового компаратора для обеспечения плавного переключения каналов.

Пользователи также могут полностью контролировать процесс переключения, настраивая соответствующие регистры.

Коммутационный аппарат переключения каналов ОРА0 отвечает за коммутацию входных каналов в реальном времени и может реализовывать эффекты двух входов на основе только одного операционного усилителя.

ОРА0 также переключит соответствующий режим ввода канала при переключении канала управления. Таким образом, мы можем достичь трех комбинаций входных сигналов двух входов вперед, двух инвертированных входов и одной инверсии вперед.

Во время коммутации каналов ОРА0 произойдет неустойчивое изменение выходного сигнала операционного усилителя.

Одновременно переключающие каналы создают управляющий сигнал, который используется для управления следующим этапом схемы, которая обрабатывает выходной сигнал ОУ.

Например, аналоговые компараторы, аналоговые компараторы могут

использовать эту управляющую информацию для соответствующей фильтрации своего выхода, чтобы избежать неустойчивости и помех, вызванных переключением каналов.

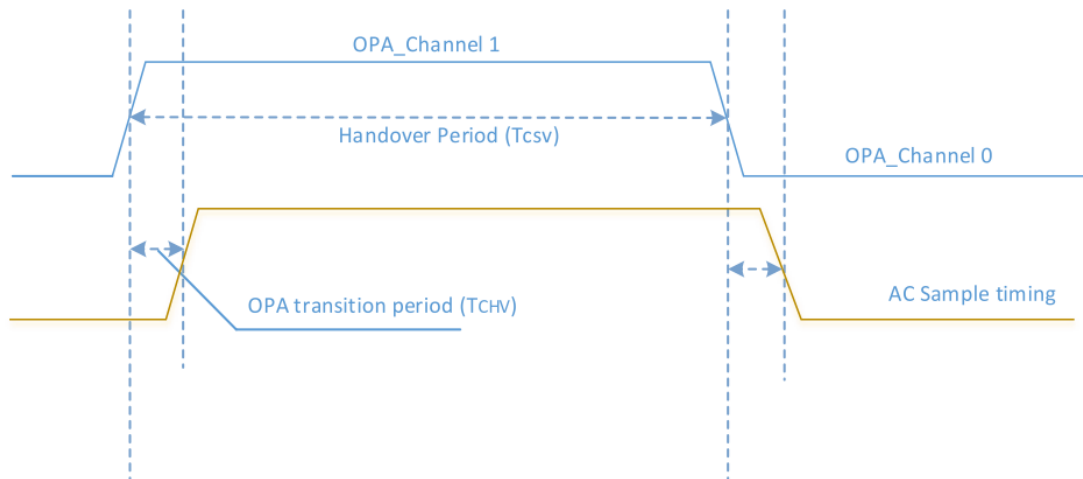


Рис.19.2. Время переключения каналов

Как показано на Рис.19.2., Tcsv - это цикл переключения каналов OPA0. После того, как OPA0 запускает переключение каналов, существует встроенное время установления (Tchv).

За это время выход OPA0 может быть нестабильным или может не сразу отражать состояние текущего канала. Таким образом, цикл Tchv может быть сконфигурирован так, чтобы позволить следующему уровню обработки отфильтровывать помехи, введенные каналом переключения.

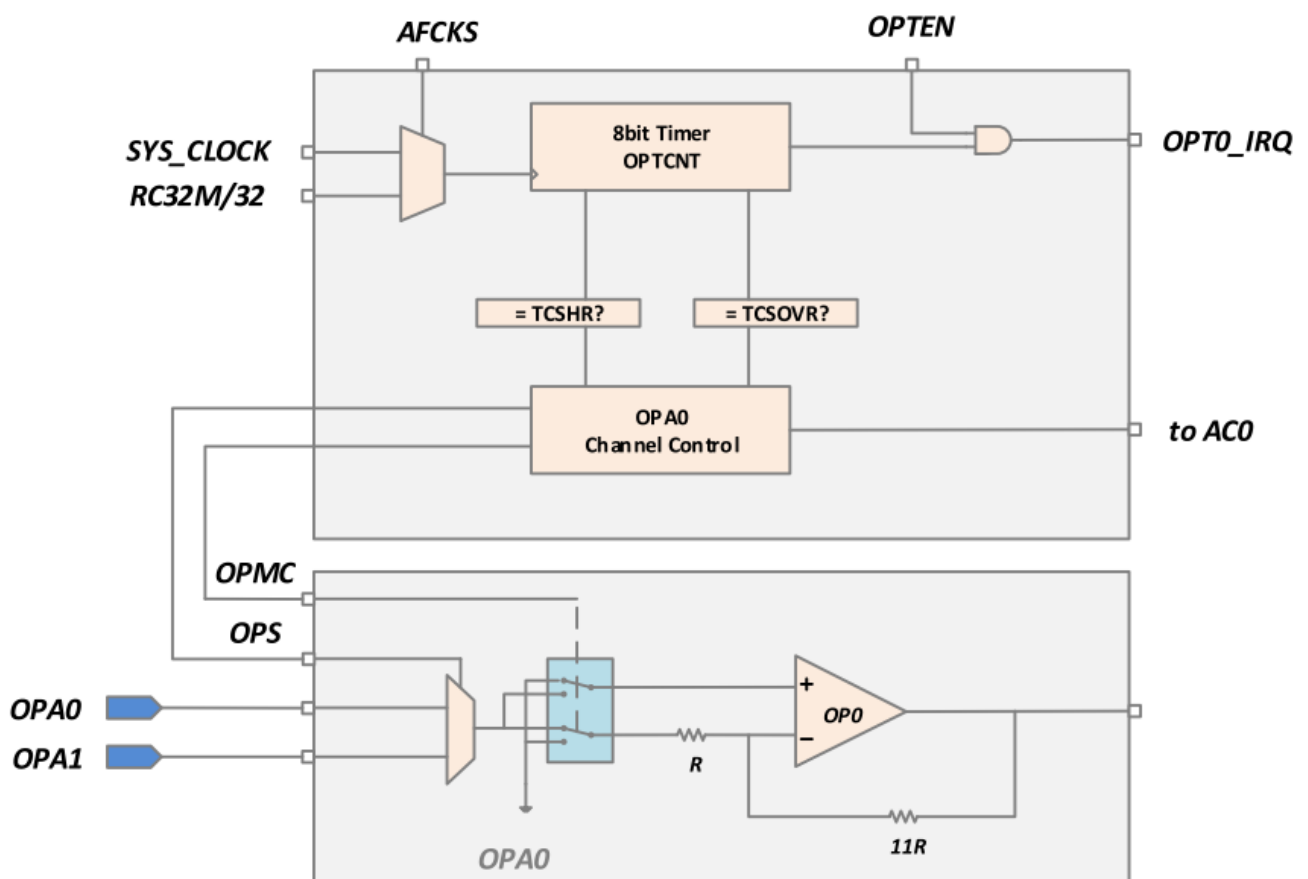


Рис19.3 Коммутационный аппарат переключения каналов OPA0

Коммутационный аппарат переключения каналов OPA0 внутренне содержит 8-разрядный счетчик для установки периода переключения каналов (T CSV). Считанные такты могут быть системными тактами по умолчанию, или может быть выбран внутренний генератор RC 32 МГц, деленный на 32. (1 МГц).

Таким образом, пользователь может выбрать соответствующий период переключения в соответствии с конкретными требованиями приложения.

Для приложений, для которых требуется быстрый ответ запуска, например защита от перегрузки по току, вы можете выбрать более быстрый счет тактов и сконфигурировать меньший цикл счета.

Таймер используется для генерации цикла Tschv в дополнение к Tcsv.

Пользователи могут контролировать время переключения каналов через регистры.

Таймер переключения каналов также может использоваться в качестве независимого 8-битового таймера в дополнение к синхронизации для переключения канала.

Сигнал прерывания генерируется после переполнения таймера.

Пользователи могут использовать программное обеспечение для переключения входных каналов путем прерывания служб. Для того, чтобы программное обеспечение переключило канал на более удобный.

Контроллер OPA0 реализует управляющий регистр (OP0CRA), выделенный для канала переключения программного обеспечения. Пользователю нужно только записать 1 в бит SCSW в регистре для достижения переключения канала. При использовании таймера программного обеспечения для переключения каналов управление синхронизацией TCHV может быть достигнуто путем настройки OP0CRB.

Пользователь может также использовать таймер переключения каналов как таймер, полностью независимый от контроллера OPA0, для выполнения задач синхронизации, требуемых другими системами.

Программное обеспечение может включать только таймер, устанавливая бит OPTEN регистра OP0CRB. Выбор таймера подсчета таймера также контролируется битом AFCKS регистра TCKCSR. Предварительная настройка переполнения таймера устанавливается регистром OP0TCNT.

19.3. Входы аналогового компаратора

Оба входа аналогового компаратора поддерживают несколько выбираемых источников входного сигнала. Прямой вход поддерживает выключение микросхемы ACIN0 и 8-разрядный ЦАП, на которые подается внутреннее опорное напряжение. Выбор источника входного сигнала управляется битом ACBG в ACSR регистра AC Control Status. Подробнее см. В описании регистра.

Инверсный вход поддерживает выключение микросхемы ACIN1, выход мультиплексора ADC и выход OPAMP. Выбор инверсного входного канала компаратора контролируется битом ACME00/01 в регистре ADCCSRB из модуля АЦП.

Когда конфигурация выбирает ACIN1, бит AIND1 регистра DIDR1 должен быть установлен **одновременно**, иначе вход ACIN1 не будет выбран для инверсного входа аналогового компаратора.

АСМЕ01	АСМЕ00	DIDR1[1]	MUX[2:0]	АС Инверсный вход
0	0	0	XXX	ADC[xxx]
0	0	1	XXX	AIN1
0	1	1	XXX	ADC[xxx]
0	1	X	000	ADC0
0	1	X	001	ADC1
0	1	X	010	ADC2
0	1	X	011	ADC3
0	1	X	100	ADC4
0	1	X	101	ADC5
0	1	X	110	ADC6
0	1	X	111	ADC7
1	X	X	X	OPA0

19.4. Выходная фильтрация компаратора

Аналоговый компаратор реагирует на изменения входного сигнала в реальном времени. При наличии помех на входном сигнале, аналогичные нарушения или мгновенные переходы также происходят на выходе аналогового компаратора. Это вмешательство может привести к ошибочным результатам работы. Поэтому на выходе аналогового компаратора последовательно соединена схема фильтра, которая разделена на две части.

Схема удержания вывода, используемая совместно с операционным усилителем OPA0, а выход OPA0 подключен ко входу аналогового компаратора.

Если включена функция автоматического переключения каналов OPA0, выход аналогового компаратора будет удерживаться здесь во время интервала переключения канала OPA0, чтобы избежать помех, вызванных нестабильным выходом при переключении OPA0. За этой цепью удерживания находится схема фильтра, которая удерживает ширину выхода компаратора. Пользователь может отфильтровывать тревожный сигнал с небольшим временем установления после изменения конфигурации регистра.

Сигнал синхронизации (Рис.19.4.), используемый фильтром, может быть от системного тактового сигнала, или его можно выбрать как внутреннюю тактовую частоту 32 МГц с частотой 32 МГц (1 МГц).

Пользователи могут выбирать соответствующие параметры фильтрации в соответствии с характеристиками помех среды приложения.

Выходная фильтрация AC0 активируется битом ACFEN регистра OP0CRA.
Частота фильтра выбирается битом AFCKS регистра TCKCSR.

Ширина фильтра может быть установлена регистром AFTCNT.

Для конкретного метода настройки обратитесь к разделу определения регистров этой главы.

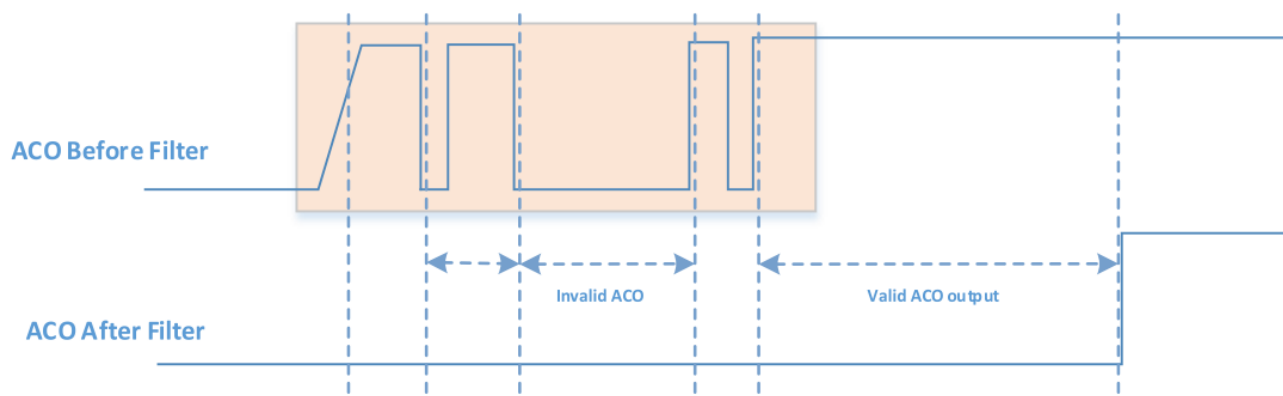


Рис.19.4. Синхронизация выходного фильтра компаратора

19.5. Выход компаратора и управление PWM

Серия LGT8FX8D может выводить до шести каналов PWM-сигналов, а сигналы PWM, генерируемые таймером и таймером 1, могут использоваться с операционным усилителем/компаратором.

Выход операционного усилителя/компаратора может использоваться для прямого отключения сигнала ШИМ для реализации более гибкой схемы защиты ШИМ.

Вход операционного усилителя имеет два канала. Вы можете выбрать любой канал или два канала в качестве источника сигнала для управления выходом ШИМ через конфигурацию программного обеспечения.

Для связывания битов конфигурации обратитесь к счетчику таймера 0/1 для определения регистра DSX0/1.

Если вам не нужно использовать функцию ОУ в вашем приложении, вы также можете использовать выход компаратора в качестве управляющего сигнала для выключения выхода ШИМ. Это позволяет использовать другие каналы компаратора для управления выходом ШИМ. В этом случае функция управления выходным сигналом PWM для компаратора усилитель канал 0 должна быть активирована через регистр DSX0/1.

Внутренний источник опорного откалиброван интегрированный источник опорного напряжения (DAC0)

LGT8FX8D внутренний 8-битный серии аналогового преобразования, выходное напряжение 1.25V и 2.56V конфигурируемый; внутренний источник опорного напряжения обеспечивает опорное напряжение для АЦП и аналоговый компаратор. Аналоговый компаратор может использовать внутренний 8-разрядный прецизионный цифро-аналоговый преобразователь (ЦАП). Цифро - аналоговый преобразователь может использоать источник опорного внутреннего напряжения для генерирования максимального выходного напряжения с кратностью деления – 256.

Выходной сигнал ЦАП может быть использован в качестве опорного напряжения аналогового компаратора инверсного входа и может быть выведен к выводам чипа в качестве внешнего опорного напряжения.

При использовании выходного сигнала ЦАП для управления другими периферийными цепями требуется внешнее устройство слежения за напряжением.

Выход ЦАП управляется индивидуально с помощью бит DACEN0/1 регистра IOCR, а напряжение на выходе ЦАП контролируется регистром DALR0.

Подробное определение см. В разделе «Определение регистра» этой главы.

19.6. Регистры Описание

ОР0СРА - Регистр управления ОРА0

OP0CRA - Регистр управления OPА0								
Адрес: 0x58			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	OPАEN	АССН	АСFEN	-	CH1IM	CH0IM	CH1EN	CH0EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7	OPАEN	Включите OPА0-модуль с OPАEN, установленным в 1. Аналоговый интерфейс OPА0 входит в активное состояние. Пользователю также необходимо настроить CH0EN / CH1EN, чтобы обеспечить правильную работу OPА0. Ниже приведено определение режима работы OPА0:						
		CH0EN	CH1EN	OPАEN	Описание функций			
		0	0	1	Канал 0 работает независимо			
		0	1	1	Канал 1 работает независимо			
		1	0	1	Канал 0 работает независимо			
		1	1	1	Двухканальный режим работы			
		x	x	0	OPА0 не работает			
6	АССН	Считать текущий выбранный канал OPА Запись 1 для переключение каналов						
5	АСFEN	Включает функцию фильтрации выходного сигнала аналогового фильтра. 1: Включает функцию фильтрации аналогового компаратора. См. Раздел Аналоговый компаратор.						
4	-	Зарезервировано						
3	CH1IM	Режим обратного входа канала 1 включен, 1: обратный вход, 0: прямой вход						
2	CH0IM	Режим обратного входа канала 0 включен, 1: обратный вход, 0: прямой вход						
1	CH1EN	OPА0 Канал 1 Включить управление, 1: Включить						
0	CH0EN	OPА0 Канал 0 Включить управление, 1: Включить						

OP0CRB - Регистр управления внешним таймером OPA0

OP0CRB - Регистр управления внешним таймером OPA0								
Адрес: 0x59			Значение по умолчанию: 0x07					
Бит	7	6	5	4	3	2	1	0
Имя	OPTEN	TCSH6	TCSH5	TCSH4	TCSH3	TCSH2	TCSH1	TCSH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7	OPTEN	Управление включением таймера, 1: включить таймер и прерывания от таймера. В режиме двухканального режима OPA0, таймер, также автоматически включается, но функция прерывания таймера не включена.						
6:0	TCSH[6:0]	Для определения количества циклов Tcsn используется время генерации времени Tcsn. Его можно установить в 0, чтобы отключить генерацию синхронизации Tcsnv, так что переключение каналов будет полностью контролироваться Tcsv.						

Регистр управления синхронизацией каналов OPA0 - OP0TCNT								
Адрес: 0x5a			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	OP0TCNT[7:0] (GPIOR4[7:0])							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	OP0TCNT[7:0] (GPIOR4[7:0])	<p>Управление генерацией времени TCSV используется для установки количества циклов TCSV, когда счетчик счетчиков достигает OP0TCNT, начинается переключение каналов, и счетчик очищается для повторного запуска.</p> <p>Когда модуль OPA0 не включен, этот регистр может использоваться как регистр ввода-вывода общего назначения GPIOR4 для временного хранения одного байта пользовательских данных для быстрого доступа к чтению и записи.</p>						

AC0SR - AC0 Регистр управления и состояния								
Адрес: 0x50			Значение по умолчанию: 0x80					
Бит	7	6	5	4	3	2	1	0
Имя	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7	ACD	<p>Бит отключает аналоговый компаратор.</p> <p>Когда бит ACD установлен в «1», аналоговый компаратор выключается.</p> <p>Когда бит ACD установлен на «0», аналоговый компаратор включен.</p>						
6	ACBG	<p>Аналоговые компаратора управляющие биты выбора опорного напряжения.</p> <p>При установке бита ACBG в «1» выбирает на прямой вход опорное напряжение.</p> <p>Когда бит ACBG установлен в «0», прямой вход выбирает внешний контакт AIN0 в качестве входного сигнала.</p>						
5	ACO	<p>Бит состояния выхода аналогового компаратора. Выход аналогового компаратора напрямую связан с битом ACO после синхронизации.</p> <p>Программное обеспечение может считывать значение бит ACO для получения выходного значения аналогового компаратора.</p>						
4	ACI	<p>Аналоговый флаг прерывания компаратора. Когда выходное событие аналогового компаратора запускает режим прерывания, определенными битами ACIS[1:0], устанавливается бит ACI. Прерывание генерируется, когда бит разрешения прерывания ACIE равен «1», и устанавливается глобальное прерывание. Когда выполняется процедура обслуживания прерывания аналогового компаратора, ACI будет автоматически очищен или бит ACI будет очищен записью в него «1».</p>						
3	ACIE	<p>Бит разрешения прерывания от аналогового компаратора. Когда бит ACIE установлен в «1» и глобальное прерывание установлено - прерывания аналогового компаратора включено. Когда бит ACIE установлен на «0», прерывания аналогового компаратора отключено.</p>						
2	ACIC	<p>Бит разрешения захвата от аналогового компаратора.</p> <p>Когда бит ACIC установлен в «1», источник входного сигнала счетчика/таймера 1 поступает из выходного ACO аналогового компаратора.</p> <p>Когда бит ACIC установлен на «0», источник входного сигнала счетчика/таймера 1 поступает от внешнего контакта ICP1.</p>						

1:0	ACIS[1:0]	Режим прерывания аналогового компаратора. ACIS0 и ACIS1 вместе образуют ACIS [1: 0] для управления режимом запуска от прерывания аналогового компаратора.	
		ACIS[1:0]	Режим прерывания
		0	Запуск от повышения или падения фронта сигнала АСО
		1	Зарезервирован
		2	Запуск от спадающего фронта сигнала АСО
		3	Запуск от нарастающего фронта АСО

ADCSRB - Регистр В управления АЦП и состояния

ADCSRB - Регистр АЦП и состояния								
Адрес: 0x7B			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	ACME01	ACME00	ACME11	ACME10	ACTS	ADTS2	ADTS1	ADTS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7	ACME01	Компаратор 0 - выбор ввода для инверсного входа. 00: - выбор внешнего входа ACIN0 01: - выбор мультиплексированного выхода с АЦП 1x: - выбор выхода операционного усилителя 0						
6	ACME00							
5	ACME11	Компаратор 1 - выбор ввода для инверсного входа. 00: - выбор внешнего входа ACIN2 01: - выбор мультиплексированного выхода с АЦП 1x: - выбор выхода операционного усилителя 1						
4	ACME10							
3	ACTS	Выбор запуска источника питания переменного тока 0 - Выход АС0 в качестве источника запуска преобразования АЦП 1 - Выход АС1 в качестве источника запуска преобразования АЦП						
2:0	ADTS[2:0]	<p>Биты выбора источника управления автоматического запуска АЦП. Когда бит ADATE установлен в «1», функция автоматического запуска включена.</p> <p>При установке бит ADATE в «0» параметр ADTS недействителен. Фронт выбранного сигнала запуска включает преобразование. Когда источник запуска переключается с флага прерывания на источник запуска прерывания, запускающий сигнал генерирует нарастающий фронт. Если ADEN установлен в это время, АЦП также начнет преобразование. При переключении в режим непрерывного преобразования (ADTS = 0) функция автоматического запуска отключена.</p>						
		ADTS [2:0]	Источник запуска					
		0	Режим непрерывной преобразования					
		1	Компаратор 0/1					
		2	Внешнее прерывание 0					
		3	Счетчик/таймера 0 Сравнить совпадение					
		4	Счетчик/таймера 0 Переполнение					
		5	Счетчик/таймера 1 Сравнить совпадение В					
		6	Счетчик/таймера 1 Переполнение					
		7	Счетчик/таймера 1 Вход Событие захвата					

AFTCNT0 - Регистр конфигурации фильтра Компаратор 0

AFTCNT0 - Регистр конфигурации фильтра Компаратор 0								
Адрес: 0x51			Значение по умолчанию: 0xff					
Бит	7	6	5	4	3	2	1	0
Имя	AFTCNT0[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	AFTCNT0[7:0]	<p>При настройке периода сбора фильтра, выходной сигнал компаратора должен оставаться установленным для длины периода, прежде чем его можно считать действительным изменением компаратора, иначе изменение будет отфильтровано. Отфильтрованные отсчетные такты поступают от системных тактов или внутреннего 12KHz RC-генератора</p>						

DALR0 - Регистр управления выходным напряжением DAC0

DALR0 - Регистр управления выходным напряжением DAC0								
Адрес: 0x52			Значение по умолчанию: 0xff					
Бит	7	6	5	4	3	2	1	0
Имя								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	DAL0[7:0]	DAL0[7:0]		Выходное напряжение DAC0				
		0x00		IVREF/256				
		0x01		2*IVREF/256				
		0x02		3*IVREF/256				
					
					
		0xFC		253*IVREF/256				
		0xFD		254* IVREF/256				
		0xFE		255*IVREF/256				
		0xFF		IVREF				

DIDR1 - Регистр отключение Цифровых входов

DIDR1 - Регистр отключение Цифровых входов								
Адрес: 0x7F			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	OPAD3	OPAD2	OPAD1	OPAD0	AIND3	AIND2	AIND1	AIND0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7	OPAD3	Блокировка управляющего бита цифрового входа OPA3						
6	OPAD2	Блокировка управляющего бита цифрового входа OPA2						
5	OPAD1	Блокировка управляющего бита цифрового входа OPA1						
4	OPAD0	Блокировка управляющего бита цифрового входа OPA0						
3	AIND3	Отключить управляющий бит ACIN3						
2	AIND2	Отключить управляющий бит ACIN2						
1	AIND1	<p>Отключить управляющий бит ACIN1.</p> <p>Когда бит AIND1 установлен на «1», цифровой вход AIN1 отключается и остается на нуле. Когда аналоговый компаратор включен, функция цифрового входа AIN1 не требуется, поэтому необходимо установить AIND1.</p> <p>Когда бит AIND1 установлен на «0», цифровой вход с разъемом AIN1</p>						

		активируется, и сигнал на выводе может быть введен во внутреннюю цифровую логику. Бит ACD должен быть установлен для отключения аналогового компаратора.
0	AIND0	Отключить управляющий бит ACIN0. Когда бит AIND0 установлен на «1», цифровой вход AIN0 отключается и остается на нуле. Когда аналоговый компаратор включен, функция цифрового входа AIN0 не требуется, поэтому необходимо установить AIND0. Когда бит AIND0 установлен на «0», цифровой вход с разъемом AIN0 активируется, и сигнал на выводе может быть введен во внутреннюю цифровую логику. Бит ACD должен быть установлен для отключения аналогового компаратора.

20. Операционный Усилитель/Компаратор 1

- Точность сравнения 12 мВ
- Поддержка двухстороннего аналогового входа
- Поддержка мультиплексированного выхода с АЦП
- Поддерживает вывод из внутренних ОРАМР операционных усилителей
- Поддержка внутреннего входного напряжения 1.25 / 2.56V
- Встроенный 8-разрядный ЦАП
- Защита от перегрузки по току/перенапряжению достигается с помощью схемы ОУ
- Программируемое управление выходным фильтром

20.1. Обзор

Модуль операционного усилителя / компаратора объединяет конфигурируемый предусилитель входного режима и аналоговый компаратор со встроенным 8-разрядным ЦАПом.

Операционный усилитель поддерживает настраиваемые режимы ввода с фиксированным прямым входом (12х усилением), инверсным входом (11х усилением) усилителя, который может использоваться с аналоговым компаратором и 8-разрядным ЦАП для достижения гибкого обнаружения сигнала.

Выход 0 (0-ампер) также может быть обработан более точно аналого-цифровым преобразователем (АЦП), аналоговым компаратором.

Величина прямого электрода компаратором сравнивается со значением инверсного электрода.

Когда напряжение на положительном электроде выше напряжения на отрицательном электроде, устанавливается аналоговый выход АСО компаратора.

Когда уровень АСО изменяется, фронт сигнала может использоваться для запуска прерывания.

Выходной сигнал АСО также может использоваться для инициирования входного захвата счетчика/таймера 1 и для управления выходом ШИМ, генерируемым таймером.

Аналоговый компаратор интегрированный внутри с 8-битным цифро-аналоговым преобразователем (ЦАП), внутреннее опорное напряжение может быть разбита на 256 различных уровней опорного напряжения.

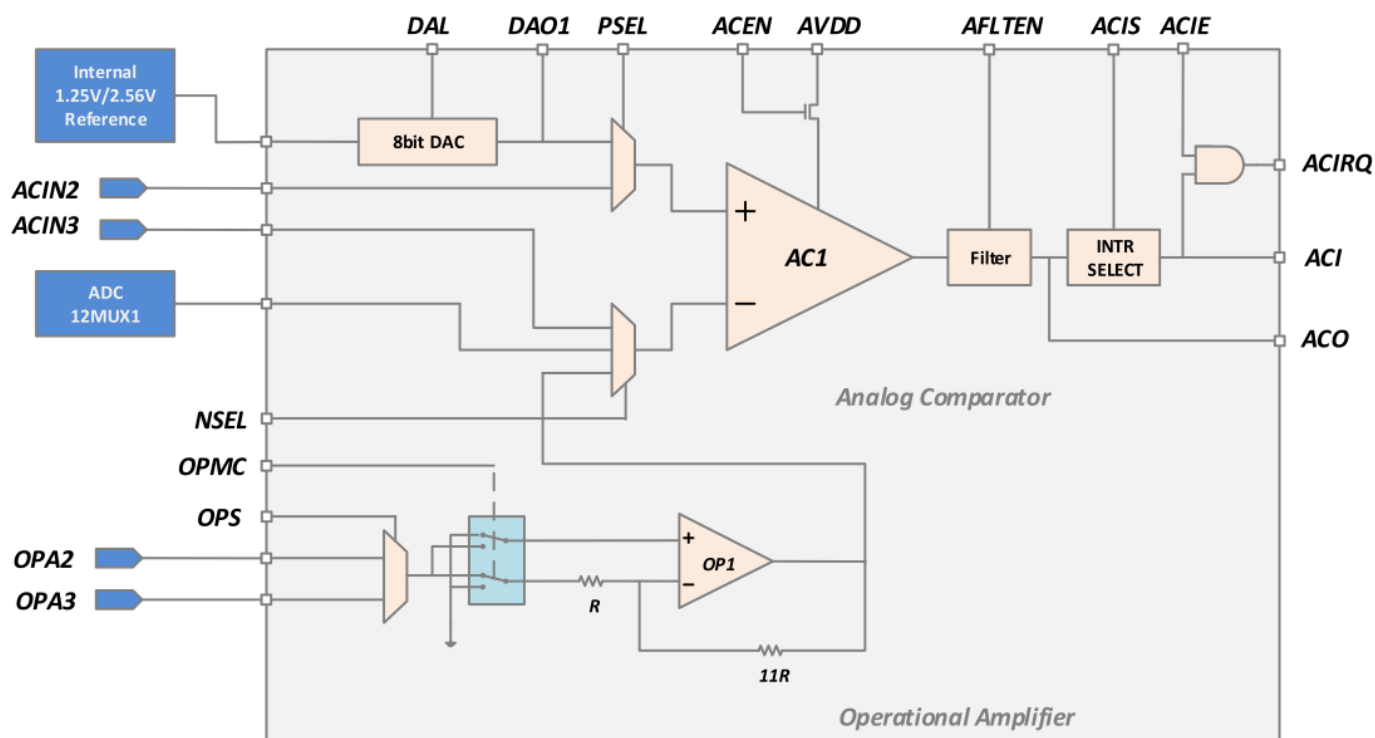


Рис.20.1. Структурная схема модуля операционного усилителя/компаратора 1

20.2. Входной коммутатор

Входной коммутатор OPA1 отвечает за мультиплексированный вход аналоговых входных каналов, переключение режима прямого/инверсного входа и усиление входного сигнала.

Усиленный сигнал OPA1 делится на два канала, которые соответственно подключены к входному каналу АЦП и отрицательному входу аналогового компаратора (AC). То есть сигналы, обрабатываемые операционными усилителями, могут обрабатываться непосредственно с помощью компараторов или они могут быть преобразованы АЦП для получения точных результатов, так что могут быть реализованы более сложные функции.

OPA1 всей аналоговой частью работы под контролем OPA1 контроллера; OPA1 аналоговая часть во время переключения ввода или изменений режима ввода, выход которого есть стабильный процесс, контроллер OPA1 отвечает за генерацию контроллера синхронизации, чтобы избежать неустойчивости на выходе. Сценарии имеют эффект.

Предыдущий операционный усилитель поддерживает два аналоговых входных канала: OPA0/1. Два канала могут регулярно переключаться программным обеспечением или автоматически переключаться через контроллер OPA внутри чипа.

Время переключения канала управляется 8-разрядным **таймером**. Источник синхронизации таймера может быть выбран в качестве системного тактового сигнала или 32-битной (1 МГц) внутренней RC32M, что обеспечивает более гибкий цикл переключения каналов и отвечает требованиям различных приложений.

После операционного усилителя включить функцию, чтобы установить зарегистрировать DIDR1 ОУ входной контакт, связанных управляющих битов, чтобы избежать влияния функции цифровой секции ввода/вывода на операционный

усилитель аналоговых входных каналов. Подробности см. В разделе определения регистров этой главы.

20.2.1. Коммутатор переключения каналов усилителя

В режиме переключения автоматического переключения таймера во избежание помех, возникающих при переключении каналов, действие переключения канала связано с выходным фильтром аналогового компаратора для обеспечения плавного переключения каналов.

Пользователи также могут полностью контролировать процесс переключения, настраивая соответствующие регистры.

Коммутационный аппарат переключения каналов ОРА1 отвечает за коммутацию входных каналов в реальном времени и может реализовывать эффекты двух входов на основе только одного операционного усилителя.

ОРА1 также переключит соответствующий режим ввода канала при переключении канала управления. Таким образом, мы можем достичь трех комбинаций входных сигналов двух входов вперед, двух инвертированных входов и одной инверсии вперед.

Во время коммутации каналов ОРА1 произойдет нестабильное изменение выходного сигнала операционного усилителя.

Одновременно переключающие каналы создают управляющий сигнал, который используется для управления следующим этапом схемы, которая обрабатывает выходной сигнал ОУ.

Например, аналоговые компараторы, аналоговые компараторы могут использовать эту управляющую информацию для соответствующей фильтрации своего выхода, чтобы избежать нестабильности и помех, вызванных переключением каналов.

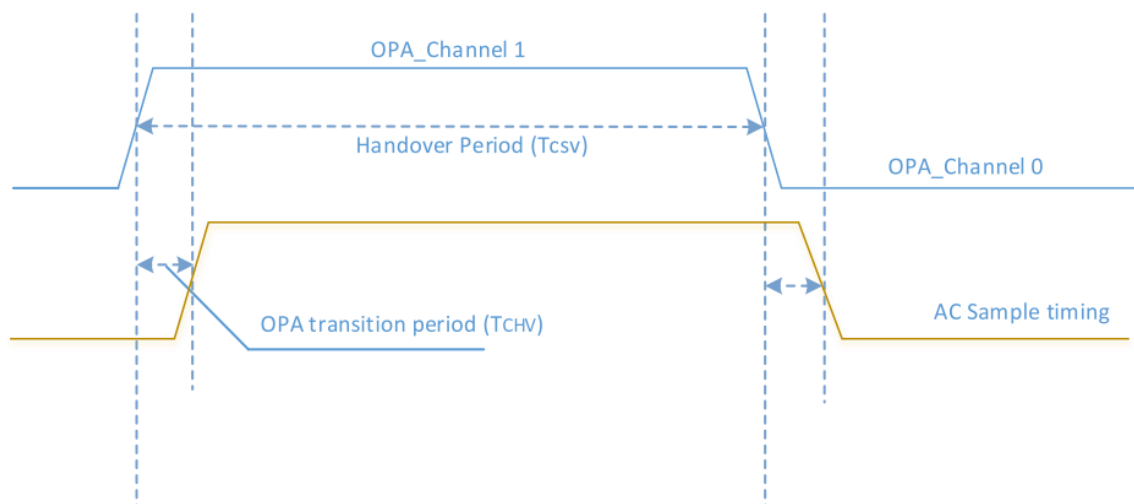


Рис.20.2. Время переключения каналов

Как показано на Рис.20.2., Tcsv - это цикл переключения каналов ОРА1. После того, как ОРА1 запускает переключение каналов, существует встроенное время установления (Tschv).

За это время выход ОРА1 может быть нестабильным или может не сразу отражать состояние текущего канала. Таким образом, цикл Tschv может быть сконфигурирован так, чтобы позволить следующему уровню обработки отфильтровывать помехи, введенные каналом переключения.

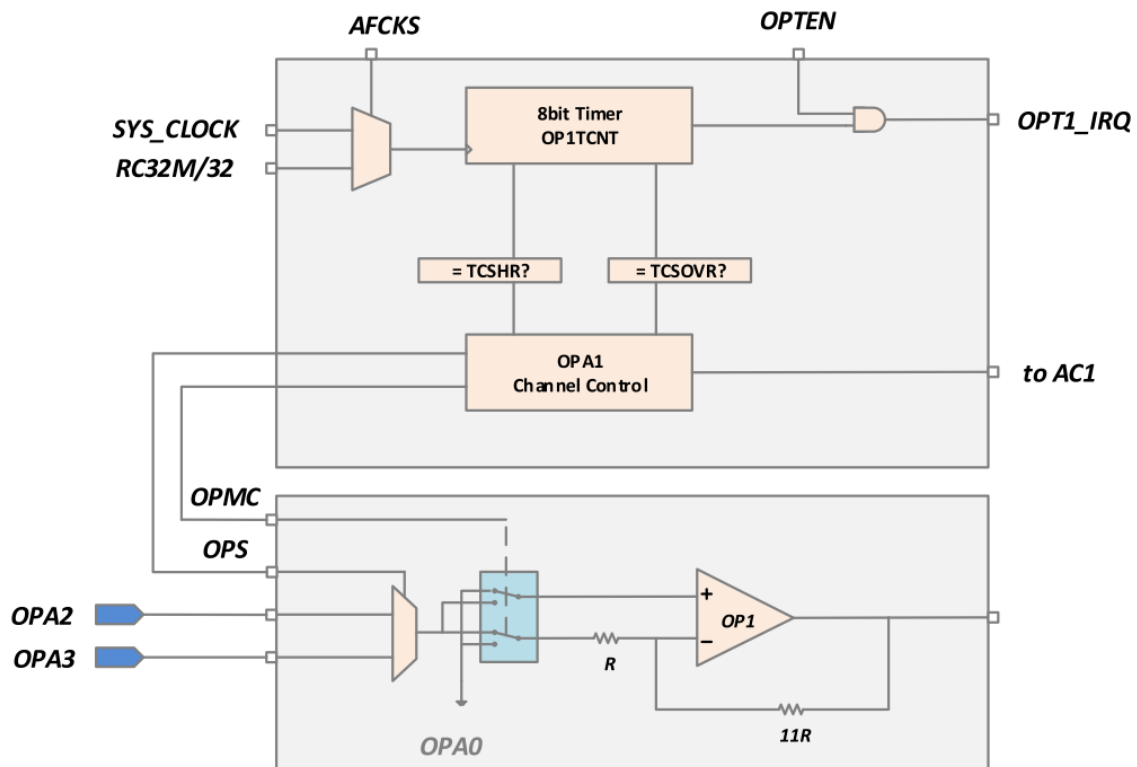


Рис20.3 Коммутационный аппарат переключения каналов OPA1

Коммутационный аппарат переключения каналов OPA1 внутренне содержит 8-разрядный счетчик для установки периода переключения каналов (Tcsv). Считанные такты могут быть системными тактами по умолчанию, или может быть выбран внутренний генератор RC 32 МГц, деленный на 32. (1 МГц).

Таким образом, пользователь может выбрать соответствующий период переключения в соответствии с конкретными требованиями приложения.

Для приложений, для которых требуется быстрый ответ запуска, например защита от перегрузки по току, вы можете выбрать более быстрый счет тактов и сконфигурировать меньший цикл счета.

Таймер используется для генерации цикла Tschv в дополнение к Tcsv.

Пользователи могут контролировать время переключения каналов через регистры.

Таймер переключения каналов также может использоваться в качестве независимого 8-битового таймера в дополнение к синхронизации для переключения канала.

Сигнал прерывания генерируется после переполнения таймера.

Пользователи могут использовать программное обеспечение для переключения входных каналов путем прерывания служб. Для того, чтобы программное обеспечение переключило канал на более удобный.

Контроллер OPA1 реализует управляющий регистр (OP1CRA), выделенный для канала переключения программного обеспечения. Пользователю нужно только записать 1 в бит SCSW в регистре для достижения переключения канала. При использовании таймера программного обеспечения для переключения каналов управление синхронизацией Tschv может быть достигнуто путем настройки OP1CRB.

Пользователь может также использовать таймер переключения каналов как таймер, полностью независимый от контроллера OPA1, для выполнения задач синхронизации, требуемых другими системами.

Программное обеспечение может включать только таймер, устанавливая бит OPTEN регистра OP1CRB. Выбор таймера подсчета таймера также контролируется битом AFCKS регистра TCKCSR. Предварительная настройка переполнения таймера устанавливается регистром OP1TCNT.

20.3. Входы аналогового компаратора

Оба входа аналогового компаратора поддерживают несколько выбираемых источников входного сигнала. Прямой вход поддерживает выключение микросхемы ACIN0 и 8-разрядный ЦАП, на которые подается внутреннее опорное напряжение. Выбор источника входного сигнала управляется битом ACBG в ACSR регистра AC Control Status. Подробнее см. В описании регистра.

Инверсный вход поддерживает выключение микросхемы ACIN1, выход мультиплексора ADC и выход OPAMP. Выбор инверсного входного канала компаратора контролируется битом ACME00/01 в регистре ADCCSRB из модуля АЦП.

Когда конфигурация выбирает ACIN1, бит AIND1 регистра DIDR1 должен быть установлен **одновременно**, иначе вход ACIN1 не будет выбран для инверсного входа аналогового компаратора.

Управление инверсным входом AC1

ACME11	ACME10	DIDR1[3]	MUX[2:0]	AC Инверсный вход
0	0	0	XXX	ADC[xxx]
0	0	1	XXX	AIN1
0	1	1	XXX	ADC[xxx]
0	1	X	000	ADC0
0	1	X	001	ADC1
0	1	X	010	ADC2
0	1	X	011	ADC3
0	1	X	100	ADC4
0	1	X	101	ADC5
0	1	X	110	ADC6
0	1	X	111	ADC7
1	X	X	X	OPA1

20.4. Выходная фильтрация компаратора

Аналоговый компаратор реагирует на изменения входного сигнала в реальном времени. При наличии помех на входном сигнале, аналогичные нарушения или мгновенные переходы также происходят на выходе аналогового компаратора. Это вмешательство может привести к ошибочным результатам работы. Поэтому на выходе аналогового компаратора последовательно соединена схема фильтра, которая разделена на две части.

Схема удержания вывода, используемая совместно с операционным усилителем OPA1, а выход OPA1 подключен ко входу аналогового компаратора.

Если включена функция автоматического переключения каналов OPA1,

выход аналогового компаратора будет удерживаться здесь во время интервала переключения канала ОРА1, чтобы избежать помех, вызванных нестабильным выходом при переключении ОРА1. За этой цепью удерживания находится схема фильтра, которая удерживает ширину выхода компаратора. Пользователь может отфильтровывать тревожный сигнал с небольшим временем установления после изменения конфигурации регистра.

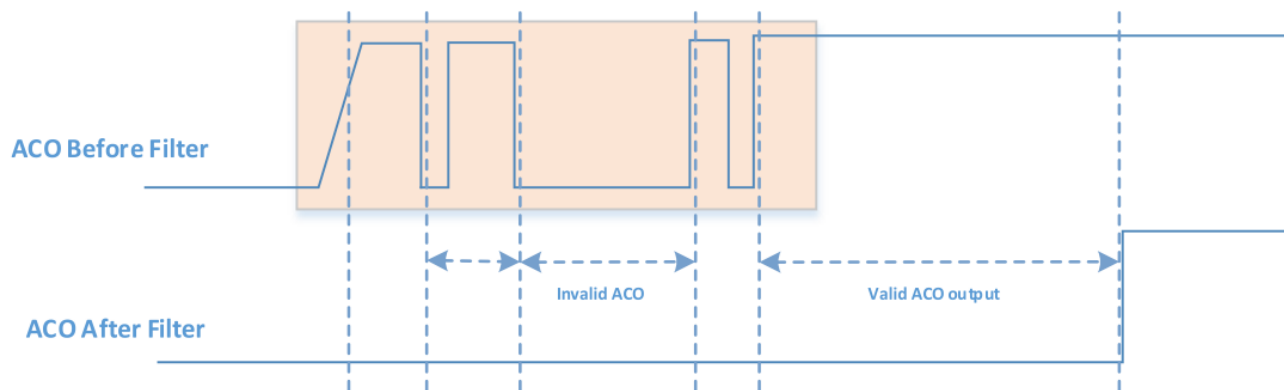


Рис.20.4. Синхронизация выходного фильтра компаратора

Сигнал синхронизации (Рис.20.4.), используемый фильтром, может быть от системного тактового сигнала, или его можно выбрать как внутреннюю тактовую частоту 32 МГц с частотой 32 МГц (1 МГц).

Пользователи могут выбирать соответствующие параметры фильтрации в соответствии с характеристиками помех среды приложения.

Выходная фильтрация АС1 активируется битом ACFEN регистра OP1CRA.

Частота фильтра выбирается битом AFCKS регистра TCKCSR.

Ширина фильтра может быть установлена регистром AFTCNT.

Для конкретного метода настройки обратитесь к разделу определения регистров этой главы.

20.5. Выход компаратора и управление PWM

Серия LGT8FX8D может выводить до шести каналов PWM-сигналов, а сигналы PWM, генерируемые таймером и таймером 1, могут использоваться с операционным усилителем/компаратором.

Выход операционного усилителя/компаратора может использоваться для прямого отключения сигнала ШИМ для реализации более гибкой схемы защиты ШИМ.

Вход операционного усилителя имеет два канала. Вы можете выбрать любой канал или два канала в качестве источника сигнала для управления выходом ШИМ через конфигурацию программного обеспечения.

Для связывания битов конфигурации обратитесь к счетчику таймера 0/1 для определения регистра DSX0/1.

Если вам не нужно использовать функцию ОУ в вашем приложении, вы также можете использовать выход компаратора в качестве управляющего сигнала для выключения выхода ШИМ. Это позволяет использовать другие каналы компаратора для управления выходом ШИМ. В этом случае функция управления выходным сигналом PWM для компаратора усилитель канал 0 должна быть

активирована через регистр DSX0/1.

20.6. Внутренний источник опорного откалиброван интегрированный источник опорного напряжения (DAC1)

LGT8FX8D внутренний 8-битный серии аналогового преобразования, выходное напряжение 1.25V и 2.56V конфигурируемый; внутренний источник опорного напряжения обеспечивает опорное напряжение для АЦП и аналоговый компаратор. Аналоговый компаратор может использовать внутренний 8-разрядный прецизионный цифро-аналоговый преобразователь (ЦАП). Цифро - аналоговый преобразователь может использовать источник опорного внутреннего напряжения для генерирования максимального выходного напряжения с кратностью деления – 256.

Выходной сигнал ЦАП может быть использован в качестве опорного напряжения аналогового компаратора инверсного входа и может быть выведен к выводам чипа в качестве внешнего опорного напряжения.

При использовании выходного сигнала ЦАП для управления другими периферийными цепями требуется внешнее устройство слежения за напряжением.

Выход ЦАП управляется индивидуально с помощью бит DACEN0/1 регистра IOCR, а напряжение на выходе ЦАП контролируется регистром DALR0.

Подробное определение см. В разделе «Определение регистра» этой главы.

20.7. Регистры Описание

OP1CRA - Регистр управления ОРА1

OP1CRA - Регистр управления OP1								
Адрес: 0x32			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	OPAEN	ACCH	ACFEN	-	CH1IM	CH0IM	CH1EN	CH0EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7	OPAEN	Включите OP1-модуль с OPAEN, установленным в 1. Аналоговый интерфейс OP1 входит в активное состояние. Пользователю также необходимо настроить CH0EN / CH1EN, чтобы обеспечить правильную работу OP1. Ниже приведено определение режима работы OP1:						
		CH0EN	CH1EN	OPAEN	Описание функций			
		0	0	1	Канал 0 работает независимо			
		0	1	1	Канал 1 работает независимо			
		1	0	1	Канал 0 работает независимо			
		1	1	1	Двухканальный режим работы			
		x	x	0	OP1 не работает			
6	ACCH	Считать текущий выбранный канал ОРА Запись 1 для переключение каналов						
5	ACFEN	Включает функцию фильтрации выходного сигнала аналогового фильтра. 1: Включает функцию фильтрации аналогового компаратора. См. Раздел Аналоговый компаратор.						
4	-	Зарезервировано						

		<p>Фронт выбранного сигнала запуска включает преобразование. Когда источник запуска переключается с флага прерывания на источник запуска прерывания, запускающий сигнал генерирует нарастающий фронт. Если ADEN установлен в это время, АЦП также начнет преобразование. При переключении в режим непрерывного преобразования (ADTS = 0) функция автоматического запуска отключена.</p>	
		ADTS [2:0]	Источник запуска
		0	Режим непрерывной преобразования
		1	Компаратор 0/1
		2	Внешнее прерывание 0
		3	Счетчик/таймера 0 Сравнить совпадение
		4	Счетчик/таймера 0 Переполнение
		5	Счетчик/таймера 1 Сравнить совпадение В
		6	Счетчик/таймера 1 Переполнение
		7	Счетчик/таймера 1 Вход Событие захвата

AFTCNT1 - Регистр конфигурации фильтра Компаратор 1

AFTCNT1 - Регистр конфигурации фильтра Компаратор 1								
Адрес: 0x30				Значение по умолчанию: 0xff				
Бит	7	6	5	4	3	2	1	0
Имя	AFTCNT1[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	AFTCNT1[7:0]	<p>При настройке периода сбора фильтра, выходной сигнал компаратора должен оставаться установленным для длины периода, прежде чем его можно считать действительным изменением компаратора, иначе изменение будет отфильтровано. Отфильтрованные отсчетные такты поступают от системных тактов или внутреннего 12KHz RC-генератора</p>						

DALR1 - Регистр управления выходным напряжением DAC1

DALR1 - Регистр управления выходным напряжением DAC1								
Адрес: 0x31				Значение по умолчанию: 0xff				
Бит	7	6	5	4	3	2	1	0
Имя								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	DAL1[7:0]	DAL1[7:0]			Выходное напряжение DAC1			
		0x00			IVREF/256			
		0x01			2*IVREF/256			
		0x02			3*IVREF/256			
				
				
		0xFC			253*IVREF/256			
		0xFD			254* IVREF/256			
		0xFE			255*IVREF/256			
		0xFF			IVREF			

DIDR1 - Регистр отключение Цифровых входов

DIDR1 - Регистр отключение Цифровых входов								
Адрес: 0x7F			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	OPAD3	OPAD2	OPAD1	OPAD0	AIND3	AIND2	AIND1	AIND0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7	OPAD3	Блокировка управляющего бита цифрового входа OPA3						
6	OPAD2	Блокировка управляющего бита цифрового входа OPA2						
5	OPAD1	Блокировка управляющего бита цифрового входа OPA1						
4	OPAD0	Блокировка управляющего бита цифрового входа OPA0						
3	AIND3	Отключить управляющий бит ACIN3						
2	AIND2	Отключить управляющий бит ACIN2						
1	AIND1	<p>Отключить управляющий бит ACIN1.</p> <p>Когда бит AIND1 установлен на «1», цифровой вход AIN1 отключается и остается на нуле. Когда аналоговый компаратор включен, функция цифрового входа AIN1 не требуется, поэтому необходимо установить AIND1.</p> <p>Когда бит AIND1 установлен на «0», цифровой вход с разъемом AIN1 активируется, и сигнал на выводе может быть введен во внутреннюю цифровую логику. Бит ACD должен быть установлен для отключения аналогового компаратора.</p>						
0	AIND0	<p>Отключить управляющий бит ACIN0.</p> <p>Когда бит AIND0 установлен на «1», цифровой вход AIN0 отключается и остается на нуле. Когда аналоговый компаратор включен, функция цифрового входа AIN0 не требуется, поэтому необходимо установить AIND0.</p> <p>Когда бит AIND0 установлен на «0», цифровой вход с разъемом AIN0 активируется, и сигнал на выводе может быть введен во внутреннюю цифровую логику. Бит ACD должен быть установлен для отключения аналогового компаратора.</p>						

21. ADC-аналого-цифровой преобразователь

- 12-разрядное разрешение с $\pm 1\text{LSB DNL}$ и $\pm 1.5\text{LSB INL}$
- Частота дискретизации до 250KSPS при самом высоком разрешении
- 8 мультиплексированных односторонних входных каналов
- Диапазон входного напряжения АЦП равен 0-V_{CC}
- Режим непрерывного преобразования или однократного преобразования
- Дополнительные внутренние 1.25V / 2.56V опорные напряжения
- Поддержка AV_{CC} и внешнего входа опорного напряжения
- Канал обнаружения напряжения питания 1 / 4V_{CC}
- Входные каналы, поддерживающие внутренний операционный усилитель 0
- Режим запуска автоматического запуска, основанный на источнике прерывания
- Результат преобразования поддерживает дополнительный режим с левым выравниванием
- Прерывание прерывания преобразования АЦП

21.1. Обзор

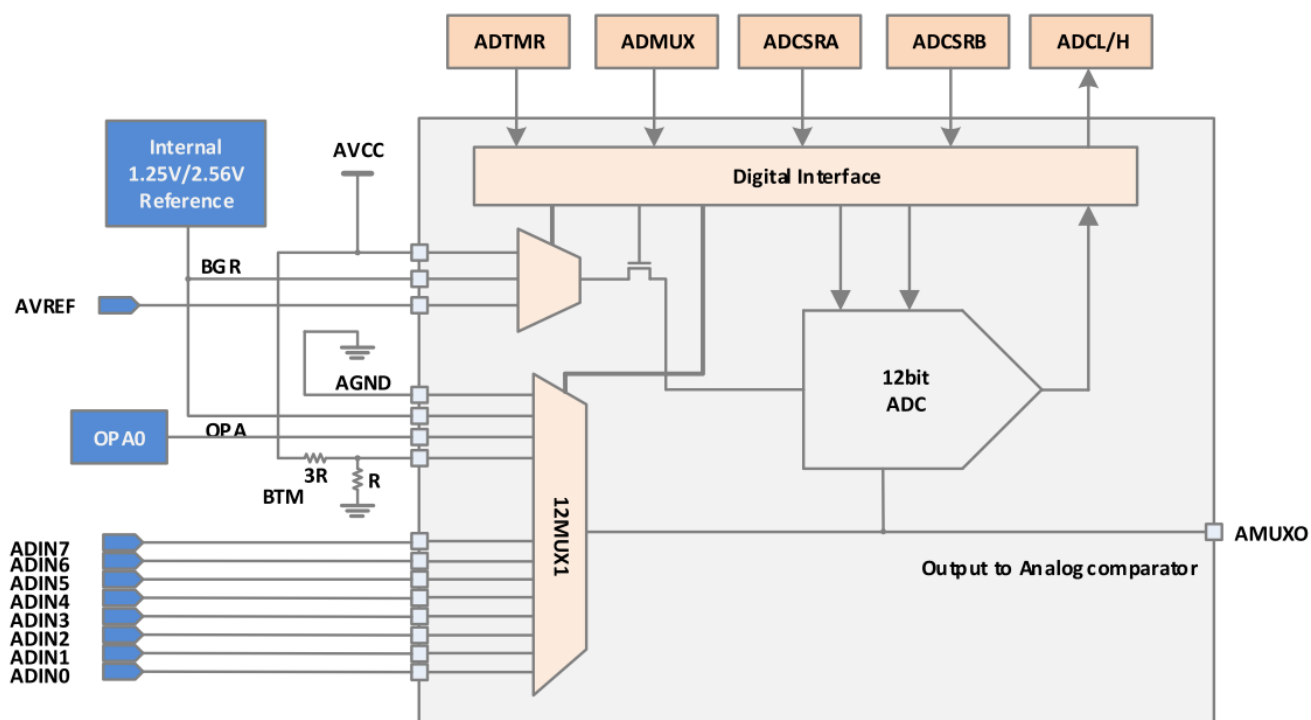


Рис.21 .1. Структурная схема АЦП

АЦП представляет собой 12-битный АЦП с последовательным приближением.

АЦП подключен к 8-канальному аналоговому мультиплексору, который отображает восемь однополярных входных напряжений от порта A. В однополярные входное напряжение относительно 0 В (GND).

21.2. 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.

Опорным напряжением может служить напряжение питания АЦП AVCC, внешний опорный AREF или внутренний опорный сигнал 1.25V / 2.56V, который выбирается путем записи в бит REFS в регистре ADMUX.

Канал аналогового ввода может быть выбран путем записи в бит MUX регистра ADMUX. Любой входной контакт АЦП, внешний опорный вывод и внутренний опорный сигнал могут использоваться как входы с однополярного АЦП.

Входные выводы ADC 0-5 могут использоваться как дифференциальные входы для АЦП. Дифференциальное усиление можно выбрать, записав бит GAIN в регистре ADTMR.

АЦП можно запустить, установив бит ADEN в регистр ADCSRA.

Не потребляет электроэнергию АЦП, когда бит ADEN очищен, поэтому рекомендуется отключить АЦП перед тем, как перейти 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 влево и выберете только 8-битную точность преобразования, то достаточно читать ADCH. В противном случае сначала прочитайте ADCL, а затем прочитайте 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.

21.2.1. Запуск преобразования

Чтобы запустить преобразования ADC, в бит ADSC записывается «1». Этот бит остается высоким во время преобразования, пока он не будет очищен аппаратным обеспечением после завершения преобразования.

Если канал изменяется во время преобразования, ADC завершит это преобразование перед изменением канала. Преобразования АЦП имеют разные источники запуска. Установка бита активации автоматического запуска АЦП, ADSC, регистра ADCSRA позволяет автоматически запускать. Источник запуска можно выбрать, установив бит выбора запуска АЦП ADIFSC регистра ADCSRA.

Когда возникает нарастающий фронт выбранного запускающего сигнала, пределитель ADC сбрасывается и запускает преобразование. Это дает возможность начать преобразование с фиксированным интервалом времени.

После преобразования, даже если сигнал запуска еще существует, новое преобразование не будет запущено.

Если во время преобразования возникает запуск, передний фронт также игнорируется. 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 при 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 иницируется 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 начинается конвертация, ADSC всегда «1» во время процесса преобразования

21.2.2. 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 системными тактами для создания 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 нарастающего фронта следующего тактового цикла АЦП.

Для нормального преобразования требуется 15 тактовых циклов АЦП. После того, как ADC включен (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 начинается с 1,5 тактов ввода АЦП после начала преобразования, а результат вывода первого преобразования АЦП происходит при 14,5 часах ввода 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 установить флаг ADSC или автоматически запускать, чтобы инициировать новое преобразование.

21.2.3. Изменение канала или источника опорного напряжения

Изменение каналов MUX или источника опорного напряжения REFS в регистре ADMUX реализуют единую буферизацию через временные регистры.

ЦП может осуществлять произвольный доступ к временным регистрам. Перед началом преобразования ЦП может в любой момент настроить выбор каналов и источника опорного напряжения.

Для того, чтобы обеспечить достаточное количество времени дискретизации АЦП, после запуска преобразования, вы не можете настроить выбранный канал и ссылки.

После завершения преобразования (ADIF устанавливается в регистре ADCSRA) обновляется выбор каналов и ссылок. Начало преобразования - это нарастающий фронт следующих входных тактов АЦП после установки ADSC. Поэтому рекомендуется в такте, не работает ADMUX пользователя после ввода набора ADSC АЦПА, чтобы выбрать новый канал и ссылку.

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

2:0	ADPS[2:0]	ADC управление предделителем. ADPS - выбор коэффициента деления системных тактов для генерации тактов АЦП.	
		ADPS[2:0]	Коэффициента деления
		0	2
		1	2
		2	4
		3	8
		4	16
		5	32
		6	64
		7	128

ADCSRB - Регистр В управления АЦП и состояния

ADCSRB - Регистр АЦП и состояния								
Адрес: 0x7B			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	ACME01	ACME00	ACME11	ACME10	ACTS	ADTS2	ADTS1	ADTS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7	ACME01	Компаратор 0 - выбор ввода для инверсного входа.						
6	ACME00	00: - выбор внешнего входа ACIN0 01: - выбор мультиплексированного выхода с АЦП 1x: - выбор выхода операционного усилителя 0						
5	ACME11	Компаратор 1 - выбор ввода для инверсного входа.						
4	ACME10	00: - выбор внешнего входа ACIN2 01: - выбор мультиплексированного выхода с АЦП 1x: - выбор выхода операционного усилителя 1						
3	ACTS	Выбор запуска источника питания переменного тока 0 - Выход AC0 в качестве источника запуска преобразования АЦП 1 - Выход AC1 в качестве источника запуска преобразования АЦП						
2:0	ADTS[2:0]	Биты выбора источника управления автоматического запуска АЦП. Когда бит ADATE установлен в «1», функция автоматического запуска включена. При установке бит ADATE в «0» параметр ADTS недействителен. Фронт выбранного сигнала запуска включает преобразование. Когда источник запуска переключается с флага прерывания на источник запуска прерывания, запускающий сигнал генерирует нарастающий фронт. Если ADEN установлен в это время, АЦП также начнет преобразование. При переключении в режим непрерывного преобразования (ADTS = 0) функция автоматического запуска отключена.						
		ADTS [2:0]	Источник запуска					
		0	Режим непрерывной преобразования					
		1	Компаратор 0/1					
		2	Внешнее прерывание 0					
		3	Счетчик/таймера 0 Сравнить совпадение					
		4	Счетчик/таймера 0 Переполнение					
		5	Счетчик/таймера 1 Сравнить совпадение В					
		6	Счетчик/таймера 1 Переполнение					
		7	Счетчик/таймера 1 Вход Событие захвата					

ADMUX - Регистры управления мультиплексором

ADMUX - Регистры управления мультиплексором									
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							
7:6	REFS[1:0]	Выбор опорного напряжения путем установки бит REFS управления. Для выбора опорного напряжения, если изменение в работе в процессе преобразования, надо выждать время для нового преобразования.							
		REFS[1:0]		Опорное напряжение					
		0		AREF					
		1		AVCC					
		2		2,56V (Внутреннее)					
		3		1,25V (Внутреннее)					
5	ADLAR	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 АЦП.							
4:0	CHMUX[4:0]	Биты управления выбором источника входного сигнала АЦП.							
		CHMUX[4:0]	Односторонний источник входного сигнала						
		0	PC0			Внешние источники входного сигнала			
		1	PC1						
		2	PC2						
		3	PC3						
		4	PC4						
		5	PC5						
		6	PE1						
		7	PE3						
		8	Обнаружение напряжения питания 1 / 4VCC			Внутренние источники входного сигнала			
		9...12	Зарезервировано						
		13	Внутренний выход ОУ 0						
		14	Внутренний источник опорного напряжения						
		15	GND						

ADTMR - Регистры управления режимом ADC

ADTMR- Регистры управления режимом ADC								
Адрес: 0x7D			Значение по умолчанию: 0x01					
Бит	7	6	5	4	3	2	1	0
Имя	-	-	-	-	-	-	-	ADTM
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:1	-	Зарезервировано						
0	ADTM	Тестовый режим, внутреннее опорное напряжение от выходного порта AVREF						

DIDR0 - Регистр отключение Цифровых входов

DIDR0 - Регистр отключение Цифровых входов								
Адрес: 0x7E			Значение по умолчанию: 0x00					
Бит	7	6	5	4	3	2	1	0
Имя	ADC7D	ADC6D	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	Имя	Определение бит						
7:0	ADCD[7:0]	<p>Бит отключения Цифрового вход.</p> <p>Когда бит ADCxD установлен в «1», цифровой вход вывода ADCx отключается и остается на нуле.</p> <p>Когда аналоговый компаратор включен, функция цифрового входа ADCx не требуется, поэтому установите ADCxD.</p> <p>Когда бит ADCxD установлен в «0», цифровой вход вывода ADCx включен и сигнал на выводе может быть введен во внутреннюю цифровую логику. В это время бит ADEN должен быть очищен, а аналоговый компаратор должен быть отключен.</p>						

22. Электрические характеристики

22.1. Абсолютная рабочая среда

Рабочая температура	-40 - 85°C	<p>Важное примечание. Когда чип работает при применении внешней рабочей среды, превышающей указанные максимальные пределы, это может привести к физическому повреждению чипа. Нормальная функция, гарантированная чипом, должна гарантировать, что рабочая среда находится в заданных максимальных пределах. Когда чип работает в течение длительного времени и максимальные предельные параметры, это повлияет на срок службы чипа и стабильность устройства.</p>
Температура хранения	-60 - 150°C	
Уровень контакта	0V - VCC	
Макс.рабочее напряжение	6,0 В	
Макс. ток ввода/вывода	30 мА VCC/GND	
Максимальный ток	200 мА	
ESD Характеристики	$\geq \pm 4$ кВ	

22.2. DC Характеристики

Типичные характеристики постоянного тока $T_A = -40 - + 85^\circ\text{C}$, $V_{CC} = 1,8 \text{ В} - 5,5 \text{ В}$

Символ	Параметр	Состояние	Мин.	Тип.	Мак.	Еден.
V _{IL}	Низкий порог входного уровня			V _{CC} /3		V
V _{IH}	Входной порог высокого уровня			V _{CC} /2		V
V _{OL}	Низкий порог выхода	I _{OL} =40mA, V _{CC} =5V			0.8	V
		I _{OL} =25mA, V _{CC} =3.3V			0.7	
V _{OH}	Высокий порог выхода	I _{OH} =20mA, V _{CC} =5V	4.4V			V
		I _{OH} =12mA, V _{CC} =3.3V	2.6V			
I _{IL}	Низкий уровень утечки I/O				1	μA
I _{IH}	Высокий уровень утечки I/O				1	μA
R _{WPU}	Слабое сопротивление подтягиванию I/O			80		кΩ
R _{PU}	Высокое сопротивление подтягиванию I/O			15		кΩ
I _{CC}	Активный	1MHz@3.3V 4MHz@3.3V		0.56 1.25		mA
	Холостой	4MHz@3.3V		0.30		mA
	Питание/Откл SO	w/o WDT@3.3V		12.0		μA
	Питание/Откл SI	V _{CC} =3.3V		7.4		μA

Примечания:

1. Вышеупомянутый тест энергопотребления настраивает драйвер питания в режим низкого драйвера. Подробную конфигурацию см. В документах, связанных с управлением питанием.

2. Тест-код управляет вводом-выводом для вывода квадратной волны с фиксированной частотой и циклически считывает состояние порта;

22.3. Внутренние характеристики тактов

Заводская калибровка

Частота	VCC	Температура	Точность
32MHz	2,5V- 5V	-40 - 85C	±1%
32KHz	2.5V- 5V	-40 - 85C	±1%

22.4. Характеристики внутреннего опорного напряжения

Заводская калибровка

Напряжение	VCC	Температура	Точность
1.25V	2.0V-5V	-40 - 85C	±1%
2.56V	3.3V-5V	-40 - 85C	±1%

22.5. Характеристики схемы низковольтного детектора

VDTs	Мин. V _{ВОН}	Типовое. V _{ВОН}	Макс. V _{ВОН}	Единицы
111	VDT Отключено			
110	1.6	1.7	2.0	V
101	2.4	2.5	2.8	
100	3.8	4.0	4.5	
011	Зарезервировано			
....				
000				

22.6. Характеристики схемы АЦП

Символ	Параметр	Условие	Мин.	Тип.	Макс.	Еден.
	Разрешение			12		Бит
V _{IN}	Входное напряжение		GND		V _{REF}	V
A _{VCC}	Источник питания		2.0		V _{CC}	V
V _{REF}	Опорное напряжение		1.0		A _{VCC}	V
CLK	Частота тактов		50K		10M	Hz
	Время преобразования	Свободная работа		19		CLK
INL	Интегральная нелинейность			3		LSB
DNL	Дифференцированный нелинейность			3		LSB
RAIN	Входное сопротивление			100		MΩ

22.7. Характеристики схемы компаратора

Символ	Параметр	Условие	Мин.	Тип.	Макс.	Еден.
	Разрешение			5		mV
V _{IN}	Входное напряжение		GND		V _{CC}	V
A _{VCC}	Источник питания		2.0		V _{CC}	V
RAIN	Входное сопротивление			100		MΩ

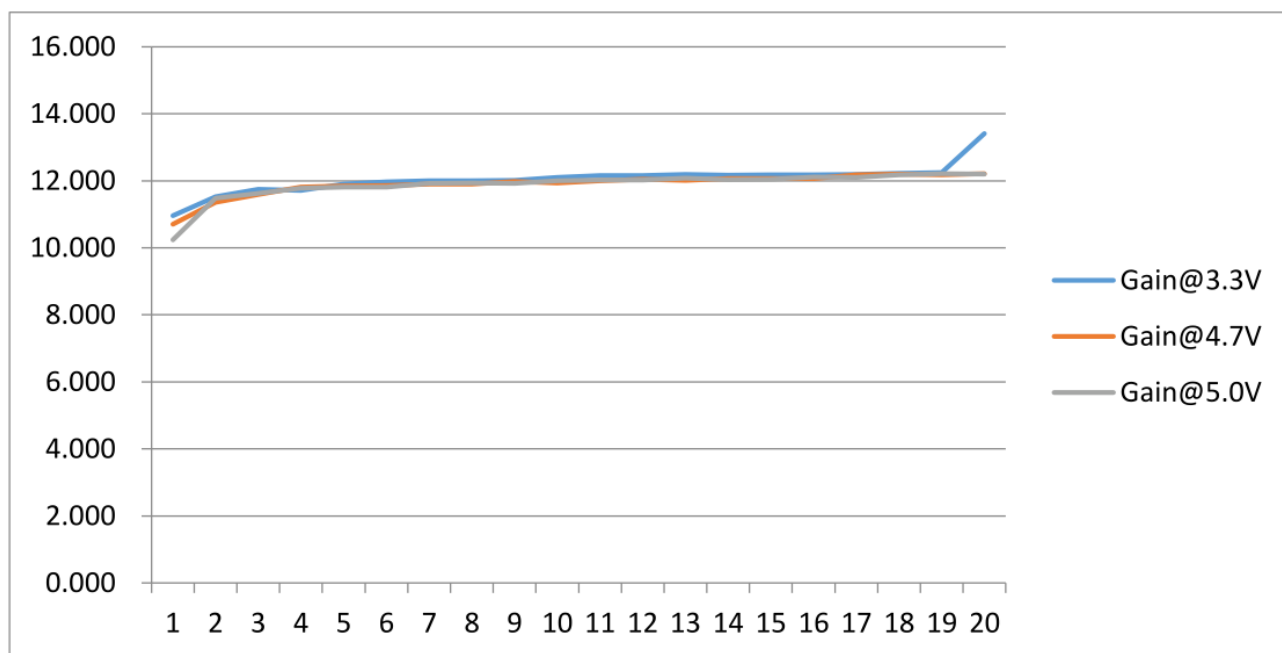
22.8. Характеристики схемы рабочего усилителя

Символ	Параметр	Условие	Мин.	Тип.	Макс.	Еден.
Усиление	Усиление усиления	V _{IN} = 100mV~220mV		12		
V _{IN}	Входное напряжение				(V _{CC} -0.9)/Усиление	V
A _{VCC}	Источник питания		2.0		V _{CC}	V
RAIN	Входное сопротивление			100		MΩ

22.8.1. Функции ввода и усиления усилителя:

1	2	3	4	5	6	7	8	9	10	Еден.
20	40	60	80	90	100	110	120	130	140	mV

1	2	3	4	5	6	7	8	9	10	Еден.
150	160	170	180	190	200	210	220	230	240	mV



23. Регистры Краткий Справочник Таблица

Адрес	Имя	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Расширенные IO Регистры									
\$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
\$F1									
\$F0	IOCR	IOCE	ST0SC1	ST0SC0	DACEN1	DACEN0	XIEN	REFIOEN	RSTIOEN
\$EE	PMXCR					OC0E4	SSB1	TDD6	RDD5
\$EC	TKCSR	-	F2XEN	TC2XF1	TC2XF0	-	AFCKS	TC2XS1	TC2XS0
\$E2	PSSR	PSS1	-	-	-	-	-	-	PSR1
\$CF	LDOCR	WEN				PDEN	VSEL2	VSEL1	VSEL0
\$CE	VCAL2	Калибровочное значение для внутреннего опорного напряжения 2,56 В							
\$CD	VCAL1	Калибровочное значение для внутреннего опорного напряжения 1,2 В							
\$C8	VCAL	Справочный калибровочный регистр внутреннего напряжения							
\$C6	UDR0	USART Данные							
\$C5	UBRR0H	-	-	-	-	Регистр скорости передачи USART Старший			
\$C4	UBRR0L	Регистр скорости передачи USART Младший							
\$C2	UCSR0C	UMSEL0		UPM0		USBS0	UCSZ01/	UCSZ00/	UCPOL0

							UDORD0	UCPHA0	
\$C1	UCSR0B	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80
\$C0	UCSR0A	RXC0	TXC0	UDRE0	FE0	DORO	UPEO	U2X0	MPCMO
\$BD	TWAMR	TWI Адрес Маска							-
\$BC	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
\$BB	TWDR	TWI Данные							
\$BA	TWAR	TWI Адрес							TWGCE
\$B9	TWSR	TWI Состояние					-	TWPS	
\$B8	TWBR	TWI Скорость передачи битов							
\$B6	ASSR	-	EXCLK	AS2	TCN2UB	0CR2AUB	0CR2BUB	TCR2AUB	TCR2BUB
\$B4	OCR2B	Таймер/Счетчик 2 Выход Сравнить В							
\$B3	OCR2A	Таймер/Счетчик 2 Выход Сравнить А							
\$B2	TCNT2	Регистр счетчика таймера/счетчика 2							
\$B1	TCCR2B	FOC2A	FOC2B	-	-	WGM22	CS2		
\$B0	TCCR2A	COM2A		COM2B		-	-	WGM21	WGM20
\$A9	PORTE	Выход порта E							
\$A8	DDRE	Направление данных E							
\$A7	PINE	Вход в порт E							
\$8B	OCR1BH	Таймер / Счетчик 1 Выход Сравнить В Старший							
\$8A	OCR1BL	Таймер / Счетчик 1 Выход Сравнить В Младший							
\$89	OCR1AH	Таймер / Счетчик 1 Выход Сравнить А Старший							
\$88	OCR1AL	Таймер / Счетчик 1 Выход Сравнить А Младший							
\$87	ICR1H	Захват таймера / счетчика 1 Старший							
\$86	ICR1L	Захват таймера / счетчика 1 Младший							
\$85	TCNT1H	Счетчик таймера / счетчика 1 Старший							
\$84	TCNT1L	Счетчик таймера / счетчика 1 Младший							
\$82	TCCR1C	FOC1A	FOC1B	-	-	-	-	-	-
\$81	TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS1		
\$80	TCCR1A	COM1A		COM1B		-	-	WGM11	WGM10
\$7F	DIDR1	OPA3D	OPA2D	OPA1D	OPA0D	AIN3D	AIN2D	AIN1D	AIN0D
\$7E	DIDRO	ADC7D	ADC6D	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D
\$7D	ADTMR	-	-	-	-	-	-	-	ADTM
\$7C	ADMUX	REFS		ADLAR	-	MUX			
\$7B	ADCSRB	ACME01	ACME00	ACME11	ACME10	-	ADTS		
\$7A	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS		
\$79	ADCH	ADC Данные Старший							
\$78	ADCL	ADC Данные Младший							
\$75	IVBASE	Базовый адрес векторного прерывания							
\$73	PCMSK3		PCINT30	PCINT29	PCINT28	PCINT27	PCINT26	PCINT25	PCINT24
\$70	TIMSK2	-	-	-	-	-	OCIE2B	OCIE2A	TOIE2
\$6F	TIMSK1	-	-	ICIE1	-	-	OCIE1B	OCIE1A	TOIE1
\$6E	TIMSK0	-	-	-	-	-	OCIE0B	OCIE0A	TOIE0
\$6D	PCMSK2	PCINT[23:16]							
\$6C	PCMSK1	PCINT[15:8]							
\$6B	PCMSK0	PCINT[7:0]							
\$69	EICRA	-	-	-	-	ISC1		ISC0	
\$68	PCICR	-	-	-	-	PCIE3	PCIE2	PCIE1	PCIE0

\$66	OSCCAL	-	-	OSC Калибровка					
\$65	PRR1	-	-	PRWDT	-	-	PREFL	PRPCI	-
\$64	PRR	PRTWI	PRTIM2	PRTIMO	-	PRTIMI	PRSPI	PRUSART0	PRADC
\$62	VDTCR	VDTCE	SWRSTN	-	-	-	VDTSEL		VDTEN
\$61	CLKPR	CLKPCE	CLKOEN1	CLKOEN0	-	CLKPS			
\$60	WDTCSR	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0
IO Регистры									
\$5F(\$3F)	SREG	1	T	H	S	V	N	Z	C
\$5E(\$3E)	SPH	Stack Point High							
\$5D(\$3D)	SPL	Stack Point Low							
\$5A(\$3A)	OP0TCNT	Переключатель каналов OPA0 Period Register							
\$59(\$39)	OP0CRB	OPTEN	OPA1 Регистр регистрации времени переключения канала						
\$58(\$38)	OP0CRA	OP1EN	ACCH	ACFEN	-	CHUM	CH0IM	CH1EN	CH0EN
\$56(\$36)	ECCR	WEN	EEN	-	-	-	-	EC1	EC0
\$55(\$35)	MCUCR	FWKEN	FPDEN	-	PUD	-	-	IVSEL	IVCE
\$54(\$34)	MCUSR	SWDD	-	-	OCDRF	WDRF	BORF	EXTRF	PORF
\$53(\$33)	SMCR	-	-	-	-	SM			SE
\$52(\$32)	DAL0	DAC0 Выходные данные Регистр							
\$51(\$31)	AFTCNT0	Регистр регистрации фильтра AC0							
\$50(\$30)	AC0SR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS	
\$4E(\$2E)	SPDR	SPI Данные							
\$4D(\$2D)	SPSR	SPIF	WCOL	-	-	-	DUAL	-	SPI2X
\$4C(\$2C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR	
\$4B(\$2B)	GPIOR2	Регистр 2 общего назначения IO							
\$4A(\$2A)	GPIOR1	Регистр 1 общего назначения IO							
\$48(\$28)	OCROB	Таймер/Счетчик 0 Выход Сравнить B							
\$47 (\$27)	OCROA	Таймер/Счетчик 0 Выход Сравнить A							
\$46(\$26)	TCNT0	Таймер/Счетчик 0 Счетчик							
\$45(\$25)	TCCR0B	FOC0A	FOC0B	OC0AS	-	WGM02	CS0		
\$44(\$24)	TCCR0A	COM0A		COM0B		-	-	WGM01	WGM00
\$43(\$23)	GTCCR	TSM	-	-	-	-	-	PSRASYS	PSRSYNC
\$42(\$22)	EEARH	EEPROM Адрес Старший							
\$41(\$21)	EEARL	EEPROM Адрес Младший							
\$40(\$20)	EEDR	EEPROM Данные Младший							
\$3F(\$1F)	EECR	EEPM2	-	EEPM1	EEPM0	EERIE	EEMWE	EEWE	EERE
\$3E(\$1E)	GPIOR0	Регистр 0 общего назначения IO							
\$3D(\$1D)	EIMSK	-	-	-	-	-	-	INT1	INT0
\$3C(\$1C)	EIFR	-	-	-	-	-	-	INTF1	INTF0
\$3B(\$1B)	PCIFR	-	-	-	-	PCIF3	PCIF2	PCIF1	PCIF0
\$37(\$17)	TIFR2	-	-	-	-	-	OCF2B	OCF2A	TOV2
\$36(\$16)	TIFR1	-	-	ICF1	-	-	OCF1B	OCF1A	TOV1
\$35(\$15)	TIFR0		-	-	-	-	OCF0B	OCF0A	TOV0
\$34(\$14)	OP1TCNT	OP1 Регистр переключения каналов							
\$33(\$13)	OP1CRB	OPTEN	OP1 Регистр задержки удержания канала						
\$32(\$12)	OP1CRA	OP1EN	ACCH	ACFEN	-	CHUM	CH0IM	CH1EN	CH0EN
\$31(\$11)	DALI	DAC1 Выбор уровня выходного сигнала							
\$30(\$10)	AFTCNT1	Регистр синхронизации фильтра AC1							

\$2F(\$0F)	AC1CSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS
\$2B(\$0B)	PORTD	Выход порта D						
\$2A(\$0A)	DDRD	Направление данных D						
\$29(\$09)	PIND	Вход в порт D						
\$28(\$08)	PORTC	Выход порта C						
\$27(\$07)	DDRC	Направление данных C						
\$26(\$06)	PINC	Вход в порт C						
\$25(\$05)	PORTB	Выход порта B						
\$24(\$04)	DDRB	Направление данных B						
\$23(\$03)	PINB	Вход в порт B						

24. Набор инструкций

25. Параметры корпусов

26. История версий