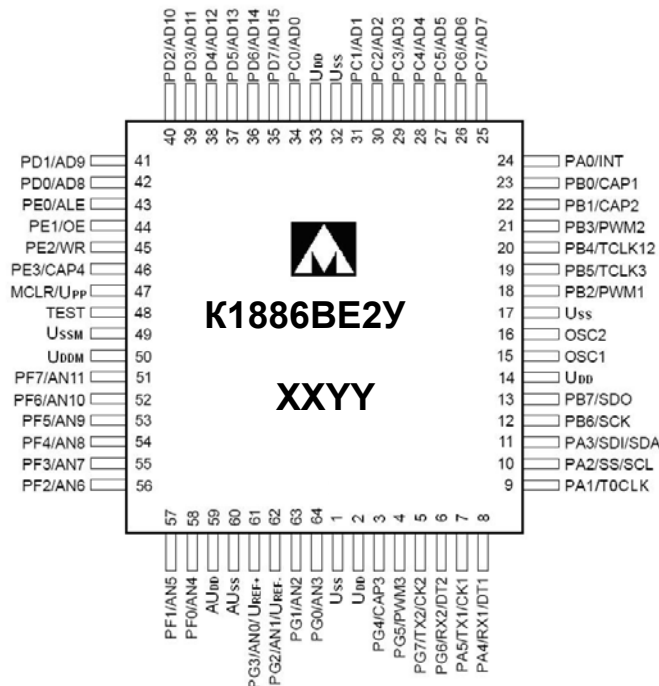




## Микросхема однокристалльной микро-ЭВМ с ЭСПЗУ (Flash-типа)

### Основные параметры микросхемы



- Тактовая частота до 33 МГц.
- Минимальная длительность цикла выполнения команды от 121 нс.
- 58 однословных инструкций (коды инструкций 16-ти разрядные).
- 8-ми разрядное АЛУ.
- Параллельные: 16-ти разрядная шина команд и 8-ми разрядная шина данных.
- Инструкция 8х8 битного аппаратно реализованного умножения.

XX – год выпуска

YY – неделя выпуска

#### Примечание

Микросхема К1886BE2АУ является полным аналогом микросхемы К1886BE2У с другим температурным диапазоном.

Обозначение	Диапазон
1886BE2У	минус 60 – 85 °С
К1886BE2У	минус 60 – 85 °С
К1886BE2АУ	0 – 70 °С

#### Тип корпуса:

- 64-х выводной металлокерамический корпус Н18.64-1В

- Поддержка прямого, косвенного и относительного режимов адресации.
- Наличие инструкций одновременно работающих с двумя регистрами.
- Все команды (включая команду 8х8 битное умножение) выполняются за один цикл (от 121 нс), кроме команд ветвления и чтения/записи таблиц, выполняемых за два цикла.
- 4-х векторный контроллер прерываний поддерживающий 18 источников прерываний (внешних и внутренних).
- 16-ти уровневый аппаратный стек.
- Возможность работы только с внутренней, с внутренней и внешней и только с внешней памятью программ (режимы: микроконтроллер, расширенный микроконтроллер и микропроцессор).
- Суммарный адресуемый объем памяти программ 128 КБайт (64К x 16 бит).

- Объем внутренней памяти программ 64 КБайта (32К x 16 бит).
- Внутренняя память программ перепрограммируемая FLASH типа.
- Объем внутренней памяти данных 902 байта.
- Все регистры «специального назначения» находятся в адресном пространстве памяти данных.
- 50 универсальных линий ввода/вывода с индивидуальной настройкой направления и высокой нагрузочной способностью, в том числе две высоковольтные (до 12 вольт) высокоточковые (до 60 мА) линии с открытым стоком.
- 12-ти канальный 10-ти разрядный АЦП последовательного приближения.
- Четыре входа захвата (регистрации события) 16-ти разрядные, с максимальным разрешением до 121 нс.
- Три выхода ШИМ (с разрешающей способностью до 10 бит).
- 16-ти разрядный таймер/счетчик с 8-ми разрядным программируемым предделителем (таймер 0).
- Два 8-ми разрядных таймер/счетчиков (таймер 1 и 2).
- 16-ти разрядный таймер/счетчик (таймер 3).
- Два универсальных синхронно-асинхронных приемопередатчика (USART/SCI) с независимыми генераторами скорости передачи данных.
- Последовательный синхронный порт (SSP) с режимами последовательного периферийного интерфейса (SPI) и двухпроводного интерфейса (I2C) (включая режим ведущего I2C).
- Микроконтроллер имеет систему сброса по включению и снижению напряжения питания (с таймерами отсрочки включения и запуска тактового генератора).
- Сторожевой таймер с собственным RC генератором на кристалле, для обеспечения высоконадежной защиты от сбоев.
- Защищенный режим (защита кода программы).
- Последовательный режим программирования памяти программ, возможность внутрисхемного программирования.
- Поддерживает режим энергосбережения SLEEP (с возможностью выхода с помощью внешних и внутренних прерываний и сброса).
- Четыре режима работы встроенного тактового генератора (RC генератор, LF - низкочастотный кварцевый генератор с низким энергопотреблением, XT – стандартный генератор с кварцевым резонатором, EC – вход внешнего тактового сигнала).
- Высокоскоростная энергосберегающая КМОП технология 0.6 мкм.
- Полностью статическая схемотехника.
- Количество циклов стирания/записи внутренней памяти программ не менее 10 000.
- Время хранения информации, записанной во внутреннюю память программ не менее 10 лет.
- Диапазон напряжений питания от 4.5 до 5.5 В.
- Максимальный ток потребления не более 50 мА.
- 64 выводной металлокерамический корпус Н18.64-1В.

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

Микросхема предназначена для широкого применения в аппаратуре общего назначения, в качестве периферийного контроллера организующего сбор и первичную обработку информации. Может использоваться для организации малопроизводительных вычислительных систем и в качестве устройства совмещения различных типов интерфейсов, так же предназначена для обработки информации от группы датчиков.

### **Основные области применения:**

- интеллектуальные датчики;
- промышленные системы управления;
- телекоммуникационное оборудование;
- системы безопасности;
- измерительное оборудование;
- малопроизводительные вычислительные устройства.

## Содержание

Описание выводов.....	7
Структурная блок-схема микросхемы .....	11
Описание функционирования микросхемы.....	12
Встроенный тактовый генератор .....	12
Использование кварцевого или керамического резонатора .....	12
Внешний тактовый генератор .....	13
Режим RC генератора.....	13
Синхронизация выполнения команды.....	14
Схема сброса микроконтроллера .....	15
Сброс по включению питания.....	16
Таймер включения питания PWRT .....	17
Таймер запуска генератора.....	17
Последовательность удержания микроконтроллера в состоянии сброса.....	18
Время задержки при различных видах сброса.....	18
Биты статуса и их значение после «сброса».....	19
Сброс по снижению напряжения питания .....	23
Прерывания.....	24
Регистр состояния прерываний (INTSTA) .....	25
Регистры разрешения периферийных прерываний PIE1 и PIE2.....	27
Регистры запроса периферийных прерываний PIR1 и PIR2 .....	29
Обработка прерываний .....	32
Прерывание от вывода PA0/INT .....	33
Прерывание от вывода PA1/T0CLK .....	33
Периферийные прерывания .....	33
Сохранение регистров при прерывании .....	34
Организация памяти .....	37
Память программ .....	38
Память данных .....	39
Регистры общего назначения (GPR).....	40
Регистры специального назначения (SFR) .....	40
Функционирование стека .....	49
Косвенная адресация .....	50
Регистры для чтения/записи таблиц.....	52
Счетчик команд .....	52
Регистр выбора банка (BSR).....	53
Считывание и запись таблиц данных.....	54
Запись таблиц во внутреннюю память.....	55
Запись таблиц во внешнюю память.....	57
Чтение таблиц .....	58
Аппаратный умножитель.....	59
Порты ввода-вывода.....	63
Порт А.....	63
Регистр порта В и регистр направления данных DDRB .....	64
Регистр порта С и регистр направления данных DDRC.....	65

Регистр порта D и регистр направления данных DDRD .....	66
Регистр порта E и регистр направления данных DDRE .....	67
Регистр порта F и регистр направления данных DDRF .....	67
Регистр порта G и регистр направления данных DDRG .....	68
Блок «таймер 0» .....	69
Таймер 1, таймер 2, таймер 3, ШИМ, захват (регистрация событий) .....	72
«Таймер 1» и «таймер 2». «Таймер 1» и «таймер 2» в 8-ми разрядном режиме	76
«Таймер 1» и «таймер 2» в 16-ти разрядном режиме .....	77
Использование выходов широтно-импульсных модуляторов (ШИМ) .....	78
«Таймер 3» .....	81
Режим трех входов захвата и регистра периода для таймера .....	82
Режим четырех входов захвата .....	84
Модули универсальных синхронно-асинхронных приемопередатчиков .....	85
Генератор скорости передачи данных .....	87
Асинхронный режим .....	88
Асинхронный передатчик .....	89
Асинхронный приемник .....	90
Синхронный ведущий режим .....	92
Передача данных в синхронном ведущем режиме .....	92
Прием данных в синхронном ведущем режиме .....	93
Синхронный ведомый режим .....	94
Передача данных в синхронном ведомом режиме .....	94
Прием данных в синхронном ведомом режиме .....	95
Модуль ведущего синхронного последовательного порта (СПП) .....	96
Режим SPI .....	101
Функционирование модуля СПП в режиме SPI .....	101
Настройка выводов при работе с модулем СПП в режиме SPI. Типовое включение .....	102
Режим ведущего .....	103
Режим ведомого .....	104
Выбор синхронного режима ведомого .....	104
Функционирование в режиме покоя (Sleep режим). Эффект сброса .....	105
Работа СПП в режиме I2C .....	106
Режим ведомого .....	108
Адресация .....	108
Прием данных ведомым .....	110
Передача данных ведомым .....	110
Поддержка общего вызова .....	111
Работа модуля в спящем режиме. Эффект при сбросе .....	112
Режим ведущего .....	112
Режим с несколькими ведущими (конкуренция) .....	113
Поддержка режима ведущего I2C .....	113
Генератор скорости обмена .....	115
Формирование флага СТАРТ в режиме ведущего I2C .....	115
Формирование флага повторный СТАРТ в режиме ведущего I2C .....	117
Передача данных в режиме ведущего I2C .....	119
Прием данных в режиме ведущего I2C .....	121

Формирование подтверждения .....	123
Формирование флага СТОП .....	125
Арбитраж линии тактового сигнала (SCL) .....	126
Функционирование в спящем режиме. Эффект сброса.....	127
Режим работы при наличии на шине более одного ведущего, арбитраж и конфликт шины .....	127
Подключение к шине I2C .....	132
Аналогово-цифровой преобразователь.....	133
Вычисление минимального времени задержки .....	137
Уравнение вычисления минимального времени заряда емкости $C_{HOLD}$ . .....	137
Специальные модули микроконтроллера .....	140
Регистры конфигурации микроконтроллера .....	140
Внутрисхемное программирование микроконтроллера .....	142
Сторожевой таймер.....	142
Режим энергосбережения (SLEEP).....	143
Схема подключения напряжения питания .....	144
Система команд .....	145
Предельные и предельно-допустимые режимы работы .....	154
Электрические параметры микросхемы .....	157
Типовые зависимости .....	159
Габаритный чертеж микросхемы .....	161
Информация для заказа.....	161
Лист регистрации изменений .....	162

**Описание выводов**

**Таблица 1**

Обозн. вывода	Номер вывода	Тип вывода	Тип вх. буфера	Назначение
OSC1	15	вход	ST	Вход генератора в режимах RC генератора и генератора с кварцевым резонатором. Вход внешнего тактового сигнала.
OSC2	16	выход	-	Выход генератора. Подключается к кварцевому резонатору (в соответствующем режиме генератора). В режимах RC генератора и внешнего тактового сигнала - выход сигнала с частотой $F_c/4$ (частота циклов команд).
MCLR/Upp	47	вход/ напряжение питания	ST	Вход «сброса» микроконтроллера с активным низким уровнем сигнала. Вход напряжения программирования.
<b>Выводы порта А</b>				<b>Дополнительное назначение выводов:</b>
PA0/INT	24	вход	ST	PA0 может использоваться как вход внешнего прерывания. Можно настроить: фронт или спад сигнала вызовет прерывание. Только входной контакт.
PA1/T0CLK	9	вход	ST	PA1 может использоваться как вход внешнего прерывания. Можно настроить: фронт или спад сигнала вызовет прерывание. PA1 - может быть выбран как вход тактового сигнала для таймера 0. Только входной контакт.
PA2/SS/SCL	10	вход/ выход(2)	ST	PA2 может быть также использован как вход SPI в режиме ведомого или вход тактового сигнала для шины I2C. Позволяет подключать повышенное напряжение, обеспечивает высокий выходной ток, выход с открытым стоком.
PA3/SDI/SDA	11	вход/ выход(2)	ST	PA3 может быть также использован как вход данных для SPI или как данные для шины I2C. Позволяет подключать повышенное напряжение, обеспечивает высокий

## Спецификация 1886BE2У, К1886BE2У

Обозн. вывода	Номер вывода	Тип вывода	Тип вх. буфера	Назначение
				выходной ток, выход с открытым стоком.
PA4/RX1/DT1	8	вход/ выход(1)	ST	PA4 может быть выбран в качестве входа асинхронного приемника, или как линия данных в синхронном режиме (USART1). Режим вывода доступен только из USART.
PA5/TX1/CK1	7	вход/ выход(1)	ST	PA5 может быть выбран в качестве выхода асинхронного передатчика, или как линия тактовых импульсов в синхронном режиме (USART1). Режим вывода доступен только из USART.
<b>Порт В</b> - параллельный двунаправленный порт ввода/вывода. Выводы имеют программно включаемые резисторы, подтягивающие к питанию.				<b>Дополнительное назначение выводов:</b>
PB0/CAP1	23	вход/ выход	ST	PB0 может быть входом захвата 1.
PB1/CAP2	22	вход/ выход	ST	PB1 может быть входом захвата 2.
PB2/PWM1	18	вход/ выход	ST	PB2 может быть выходом ШИМ 1.
PB3/PWM2	21	вход/ выход	ST	PB3 может быть выходом ШИМ 2.
PB4/TCLK12	20	вход/ выход	ST	PB4 может быть входом внешнего тактового сигнала для таймера 1 и таймера 2.
PB5/TCLK3	19	вход/ выход	ST	PB5 может быть входом внешнего тактового сигнала для таймера 3.
PB6/SCK	12	вход/ выход	ST	PB6 может быть также использован как тактовый сигнал в режимах ведущий/ведомый для SPI.
PB7/SDO	13	вход/ выход	ST	PB7 может быть также использован как выход данных для SPI.
<b>Порт С</b> - параллельный двунаправленный порт ввода/вывода.				<b>Дополнительное назначение выводов:</b>
PC0/AD0	34	вход/выход	TTL	Это младший байт 16-ти битной системной шины в режиме микропроцессора или режиме расширенного микроконтроллера. Для реализации системной шины эти выводы могут являться выходами адреса и входами/выходами данных.
PC1/AD1	31	вход/выход	TTL	
PC2/AD2	30	вход/выход	TTL	
PC3/AD3	29	вход/выход	TTL	
PC4/AD4	28	вход/выход	TTL	
PC5/AD5	27	вход/выход	TTL	
PC6/AD6	26	вход/выход	TTL	
PC7/AD7	25	вход/выход	TTL	



## Спецификация 1886BE2У, К1886BE2У

Обозн. вывода	Номер вывода	Тип вывода	Тип вх. буфера	Назначение
<b>Порт D</b> - параллельный двунаправленный порт ввода/вывода.				<b>Дополнительное назначение выводов:</b>
PD0/AD8	42	вход/выход	TTL	Это старший байт 16-ти битной системной шины в режиме микропроцессора или режиме расширенного микроконтроллера. Для реализации системной шины эти выводы могут являться выходами адреса и входами/выходами данных.
PD1/AD9	41	вход/выход	TTL	
PD2/AD10	40	вход/выход	TTL	
PD3/AD11	39	вход/выход	TTL	
PD4/AD12	38	вход/выход	TTL	
PD5/AD13	37	вход/выход	TTL	
PD6/AD14	36	вход/выход	TTL	
PD7/AD15	35	вход/выход	TTL	
<b>Порт E</b> - параллельный двунаправленный порт ввода/вывода.				<b>Дополнительное назначение выводов:</b>
PE0/ALE	43	вход/выход	TTL	В режиме микропроцессора или расширенного микроконтроллера PE0 становится выходом сигнала фиксации адреса внешней памяти. Адрес должен быть зафиксирован на спаде сигнала ALE.
PE1/OE	44	вход/выход	TTL	В режиме микропроцессора или расширенного микроконтроллера PE1 становится сигналом управления, для чтения данных из внешней памяти (активный уровень - низкий).
PE2/WR	45	вход/выход	TTL	В режиме микропроцессора или расширенного микроконтроллера PE2 становится сигналом разрешения записи во внешнюю память (активный уровень - низкий).
PE3/CAP4	46	вход/выход	TTL	PE3 может быть входом захвата 4.
<b>Порт F</b> - параллельный двунаправленный порт ввода/вывода.				<b>Дополнительное назначение выводов:</b>
PF0/AN4	58	вход/выход	ST	Может быть аналоговым входом 4.
PF1/AN5	57	вход/выход	ST	Может быть аналоговым входом 5.
PF2/AN6	56	вход/выход	ST	Может быть аналоговым входом 6.
PF3/AN7	55	вход/выход	ST	Может быть аналоговым входом 7.
PF4/AN8	54	вход/выход	ST	Может быть аналоговым входом 8.
PF5/AN9	53	вход/выход	ST	Может быть аналоговым входом 9.
PF6/AN10	52	вход/выход	ST	Может быть аналоговым входом 10.
PF7/AN11	51	вход/выход	ST	Может быть аналоговым входом 11.
<b>Порт G</b> - параллельный двунаправленный порт ввода/вывода.				<b>Дополнительное назначение выводов:</b>
PG0/AN3	64	вход/выход	ST	Может быть аналоговым входом 3.
PG1/AN2	63	вход/выход	ST	Может быть аналоговым входом 2.
PG2/AN1/ <sub>UREF-</sub>	62	вход/выход	ST	Может быть аналоговым входом 1, или входом опорного напряжения

## Спецификация 1886BE2У, К1886BE2У

Обозн. вывода	Номер вывода	Тип вывода	Тип вх. буфера	Назначение
				«земли» для АЦП.
PG3/AN0/U <sub>REF</sub> +	61	вход/выход	ST	Может быть аналоговым входом 0, или входом положительного опорного напряжения для АЦП.
PG4/CAP3	3	вход/выход	ST	Может быть входом захвата 3.
PG5/PWM3	4	вход/выход	ST	Может быть выходом ШИМ 3.
PG6/RX2/DT2	6	вход/выход	ST	Может быть выбран в качестве входа асинхронного приемника, или как линия данных в синхронном режиме (USART2).
PG7/TX2/CK2	5	вход/выход	ST	Может быть выбран в качестве выхода асинхронного передатчика, или как линия тактовых импульсов в синхронном режиме (USART2).
TEST	48	вход	ST	Управляющий вход для перехода в тестовый режим. Для обычного режима работы должен быть подключен к USS.
U <sub>SS</sub>	1, 17, 32	напряжение питания	-	«Земляной» вывод напряжения питания.
U <sub>SSM</sub>	49	напряжение питания	-	«Земляной» вывод напряжения питания FLASH памяти.
U <sub>CC</sub>	2, 14, 33	напряжение питания	-	Положительный вывод напряжения питания.
U <sub>CCM</sub>	50	напряжение питания	-	Положительный вывод напряжения питания FLASH памяти.
AU <sub>SS</sub>	60	напряжение питания	-	«Земляной» вывод для АЦП. Этот вывод должен иметь тот же потенциал что и USS.
AU <sub>CC</sub>	59	напряжение питания	-	Положительный вывод напряжения питания для АЦП. Этот вывод должен иметь тот же потенциал что и UCC.

### Обозначения:

TTL - входной буфер TTL, ST - вход с триггером Шмитта.

### Примечания:

1. Режим выхода доступен только из периферийных устройств.
2. Вывод входа/выхода с открытым стоком. Вывод переводится в состояние входа при любом типе сброса устройства.

Структурная блок-схема микросхемы

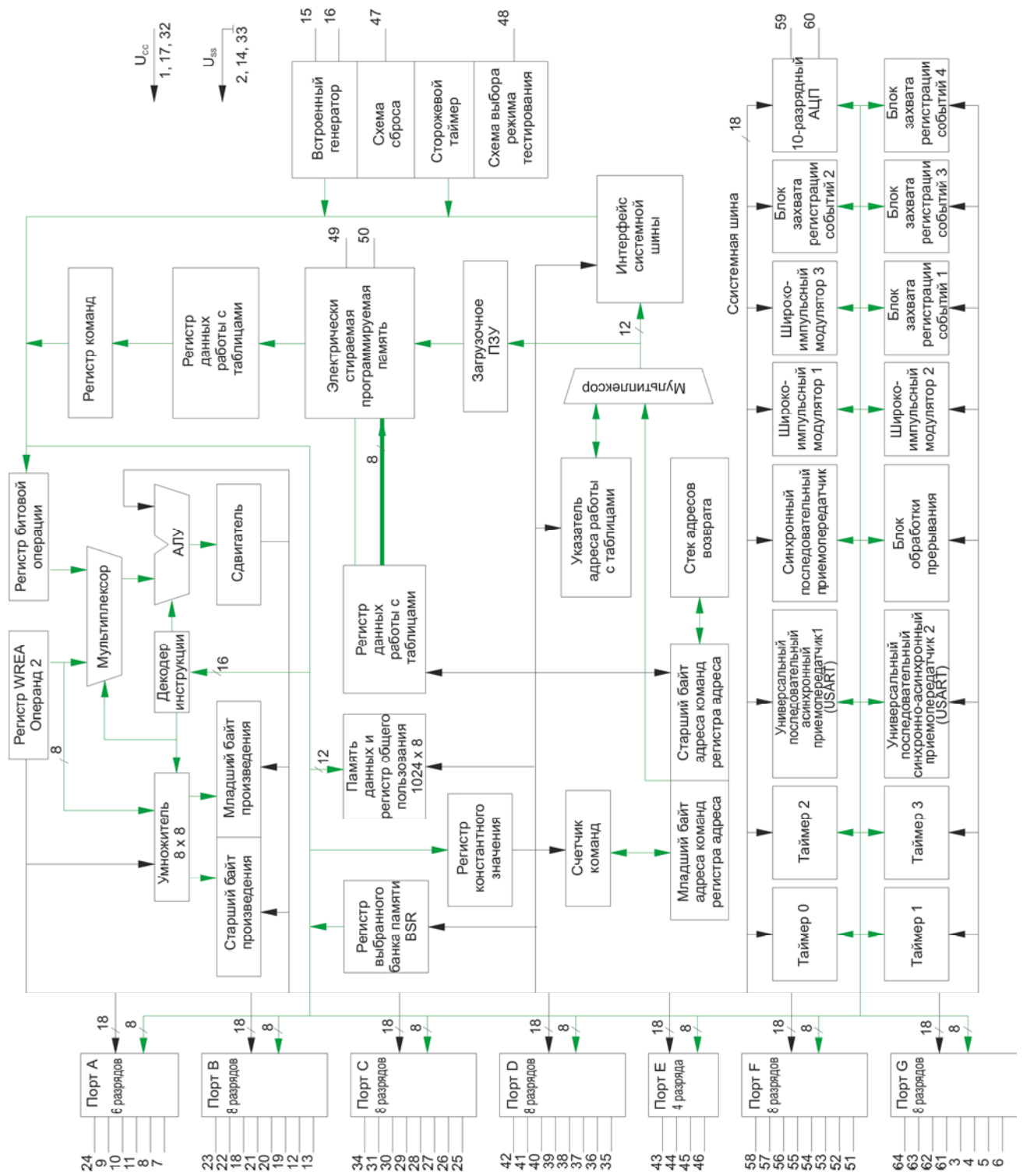


Рис. 1 Структурная блок-схема

### **Описание функционирования микросхемы**

#### **Встроенный тактовый генератор**

Генератор для формирования тактовых сигналов содержится на кристалле микроконтроллера. Четыре периода тактовых сигналов генератора составляют цикл выполнения команды.

Генератор микроконтроллера может работать в четырех режимах. Режимы выбираются программированием двух битов конфигурации FOSC1 и FOSC0 при программировании микроконтроллера. Возможен выбор следующих режимов:

- LF – генератор с внешним низкочастотным кварцевым резонатором ( $\leq 2$  МГц), обеспечивает низкое энергопотребление;
- XT – генератор с внешним кварцевым или керамическим резонатором (частота от 2 МГц до 33 МГц);
- EC – режим подачи внешнего тактового сигнала (конфигурация генератора по умолчанию);
- RC – RC генератор с частотой до 4 МГц (подключается внешняя частотозадающая RC цепочка).

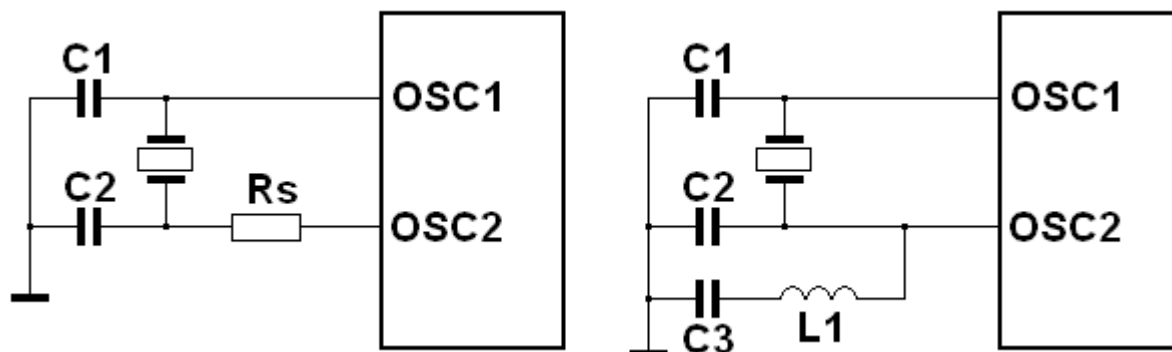
При выполнении команды SLEEP тактовый генератор выключается, уменьшая потребляемый ток. Состояние внутреннего тактового сигнала соответствует такту Q1.

При поступлении сигнала «сброс» от вывода MCLR при нормальной работе микроконтроллера тактовый генератор не выключается.

#### ***Использование кварцевого или керамического резонатора***

В режимах тактового генератора XT или LF, кварцевый или керамический резонатор подсоединяется к выводам OSC1 и OSC2. Генератор требует использования кварцевых резонаторов с параллельным резонансом. Использование резонаторов с последовательным резонансом может привести к получению тактовой частоты не соответствующей параметрам резонатора. Для частот превышающих 24 МГц, кварцевый резонатор обычно работает на частоте гармоники. В этом случае требуется резонансный контур, чтобы уменьшить усиление на частоте основной гармоники. На Рис. 1 показаны примеры схем подключения резонаторов.

При включении напряжения питания, тактовый генератор начнет генерацию сигнала. Время необходимое для запуска генератора зависит от большого количества факторов. В их число входят: частота резонатора, емкость используемых конденсаторов ( $C_1$  и  $C_2$ ), скорость нарастания напряжения питания, рабочая температура, сопротивление резистора, если он подключен, режим тактового генератора (который выбирает коэффициент усиления внутреннего инвертора). Напряжение полного размаха выхода тактового генератора может быть достаточно малым (менее 50% от VDD) пока временная диаграмма тактового сигнала центрируется к VDD/2.



**Рис. 1** Подключение резонатора.

Схема справа - для резонатора, работающего на гармониках.

Примечания:

- Резистор  $R_S$  может потребоваться для некоторых типов резонаторов.
- Параллельный резонансный контур  $L_1C_2$  отфильтровывает основную частоту:  $(2\pi f)^2 = 1/(L_1 \cdot C_2)$ .
- $C_3$  (0.1 мкф) препятствует протеканию постоянного тока на землю.
- Для резонаторов необходимы внешние конденсаторы  $C_1$  и  $C_2$  (смотрите справочные параметры). При расчете емкости конденсатора необходимо учитывать емкость печатной платы. Большая емкость увеличивает стабильность генератора, но увеличивается время запуска и ток генератора.

### **Внешний тактовый генератор**

В режиме внешнего генератора (EC), на ввод OSC1 может быть подан внешний тактовый сигнал с КМОП уровнями. В этом режиме, вход OSC1 имеет высокое входное сопротивление, а вывод OSC2 является выходом CLKOUT ( $F_C/4$ ). В качестве генераторов могут быть использованы готовые модули генераторов, обеспечивающие широкий набор тактовых частот и стабильные параметры.

### **Режим RC генератора**

В приложениях, не требующих высоко стабильной тактовой частоты, может быть использован режим RC генератора, который позволяет уменьшить стоимость устройства. Частота RC генератора зависит от напряжения питания, значения подключенных внешних сопротивления и емкости, и от рабочей температуры. Дополнительно частота будет варьироваться в некоторых пределах из-за технологического разброса параметров кристалла. Также на частоту будут влиять емкости между выводами корпуса и дорожками печатной платы, особенно при малых значениях емкости внешнего конденсатора. Необходимо учитывать и технологический разброс параметров внешних компонентов R и C. На Рис. 2 показана схема подключения RC цепочки к микроконтроллеру. Для сопротивления резистора меньше 2.2 кОм частота тактового генератора может быть нестабильна или генерация может прекратиться. Для очень большого сопротивления

резистора (более 1 МОм) генератор тактового сигнала становится чувствителен к внешним помехам, влажности и утечки тока. Поэтому, рекомендуется выбирать величину сопротивления резистора от 3 до 100 кОм. Генератор может работать без внешнего конденсатора, но рекомендуется использовать конденсатор с емкостью более 20 пФ для стабильной работы генератора. Без внешнего конденсатора, или если конденсатор имеет очень малую емкость, частота генератора может варьироваться из-за изменений во внешних емкостях, таких как емкости проводников печатной платы или выводов компонентов.

В режиме RC генератора на выводе OSC2 формируется тактовый сигнал с частотой  $F_C/4$ . Генератор в режиме RC начинает формировать тактовый сигнал сразу после достижения напряжением питания порогового уровня. Время запуска RC генератора зависит от ряда факторов: сопротивления резистора, емкости конденсатора, скорости нарастания напряжения питания, температуры и т.д.

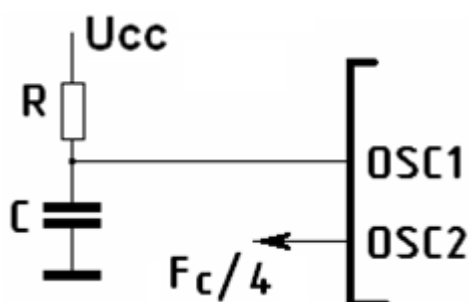


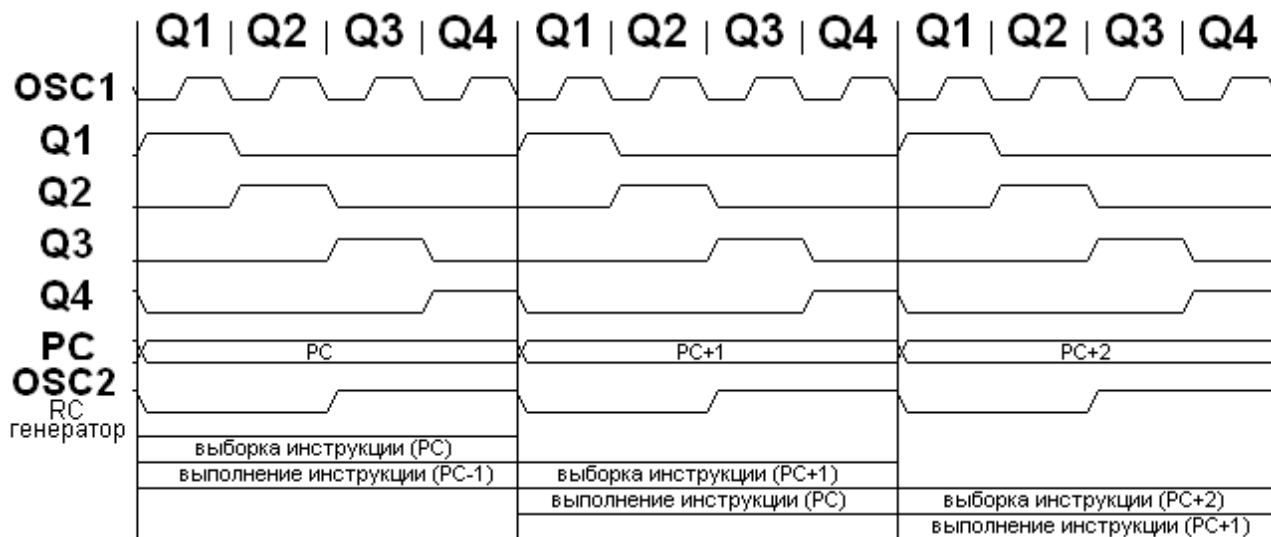
Рис. 2 Схема включения в режиме RC генератора

### Синхронизация выполнения команды

Выход тактового сигнала разделяется на 4 непересекающихся квадратурных тактовых сигнала, именуемых Q1, Q2, Q3, Q4. Программный счетчик (PC) увеличивается в такте Q1, а выборка команды из программной памяти и сохранение ее в регистре команд происходит по синхросигналу Q4. Команда декодируется и выполняется в течение циклов Q1-Q4. Тактовые сигналы и выполнение команд показано на Рис. 3.

Цикл выполнения команды состоит из четырех Q циклов (Q1, Q2, Q3, Q4). Выборка и выполнение команд происходят конвейерным способом, т.е. выборка одной команды использует тот же цикл, что и декодирование и выполнение другой команды. Благодаря конвейерной обработке команд, каждая инструкция выполняется за один цикл. Если команда изменяет счетчик команд (команды ветвления), то для выполнения команды требуется два цикла, так как необходимо удалить выбранную команду из конвейера (см. Рис. 1). Во время удаления выбирается новая команда, и затем она выполняется.

Цикл выборки команды начинается с приращения счетчика команд в такте Q1. В цикле выполнения команды, код загруженной команды помещается в регистр команд IR на такте Q1. Декодирование и выполнение команды происходит в тактах Q2, Q3, Q4. Операнд из памяти данных читается в такте Q2, а результат выполнения команды записывается в такте Q4.



**Рис. 3** Синхронизация выполнения команды



**Рис. 4** Конвейерное выполнение команд

### Схема сброса микроконтроллера

Микроконтроллеры различают следующие виды сброса:

- сброс по включению питания;
- сброс по снижению напряжения питания;
- сброс по внешнему сигналу MCLR;
- сброс по переполнению сторожевого таймера.

Некоторые регистры не изменяются после сброса: после сброса по включению питания они содержат неизвестное значение, а после любого другого сброса их состояние остается неизменным. Большинство других регистров переводятся в определенное состояние по сбросу. Биты TO и PD принимают определенные значения при различных видах сброса, как показано в Таблица 3. Эти биты в соединении с битами POR и BOR, используются в программном обеспечении для определения вида сброса. В Таблица 5 представлено описание всех видов сброса для всех регистров.

При поступлении сигнала «сброс» регистры направления передачи сигналов (DDR) устанавливаются в «1», переводя выходы портов в состояние высокоимпедансных входов. Состояние сброса некоторых периферийных

модулей может перевести выводы портов в другие состояния, например, такие как аналоговые входы или системная шина.

В состоянии «сброс» выход внутреннего тактового сигнала соответствует такту Q1. Если микроконтроллер находится в режиме «расширенного микроконтроллера» или «микропроцессора», то во время «сброса» на выводе PE0/ALE будет присутствовать низкий логический уровень выходного сигнала, а на PE1/OE и PE2/WR высокий уровень.

Упрощенная блок-схема схемы сброса на чипе (см. Рис. 5):

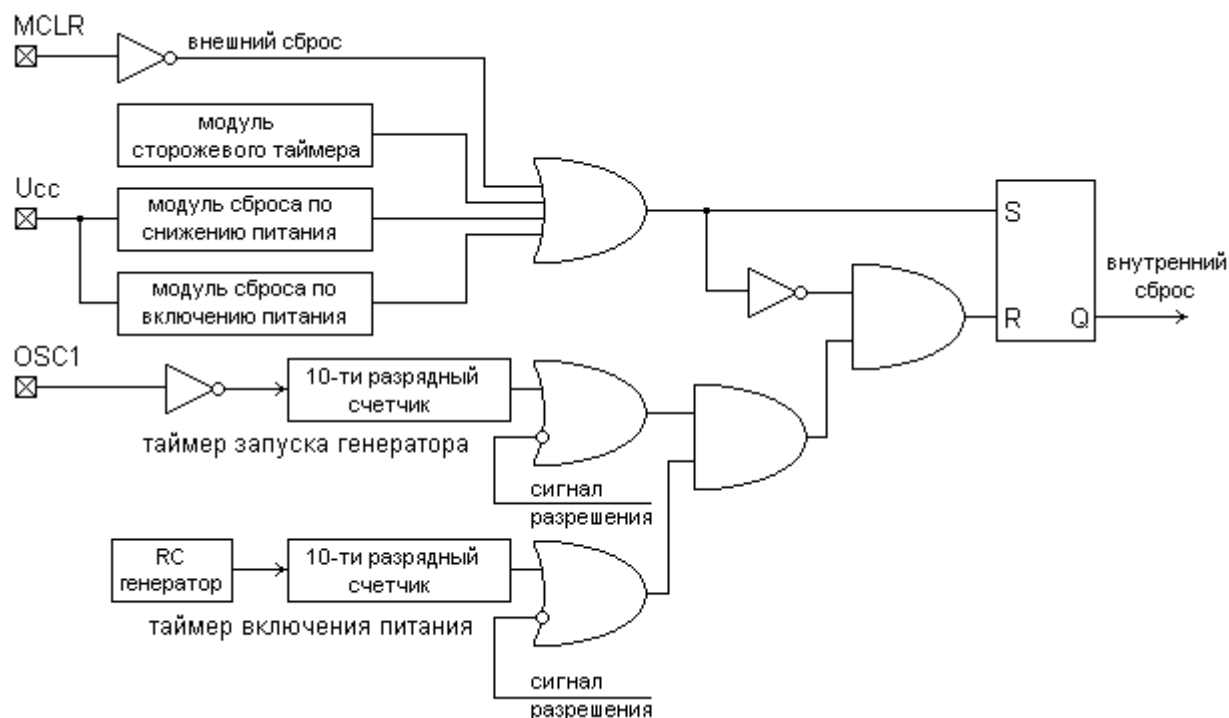


Рис. 5 Упрощенная блок-схема встроенной схемы сброса.

### Сброс по включению питания

Схема сброса по включению питания удерживает микроконтроллер в состоянии «сброс» до тех пор, пока напряжение питания не достигнет определенного уровня (примерно 1.4 – 2.3 В). Благодаря этой схеме, в ряде приложений можно обойтись без внешней RC цепочки, подключаемой к выводу MCLR. В этом случае вывод MCLR подключается через резистор или напрямую к напряжению питания. Внешняя схема «сброса» (см. Рис. 6) потребуется только в случае низкой скорости нарастания напряжения питания.



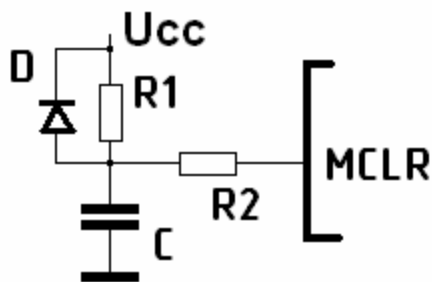


Рис. 6 Внешняя схема «сброса» по включению питания.

Примечание:

Диод D предназначен для быстрой разрядки конденсатора при снижении напряжения питания. Сопротивление резистора R1 рекомендуется выбирать не более 40 кОм (чтобы падение напряжения на резисторе, из-за токов утечки вывода MCLR/U<sub>pp</sub>, не превышало 0.2 В). Резистор R2 предназначен для ограничения тока через вывод MCLR, рекомендуемая величина 100 Ом – 1 кОм.

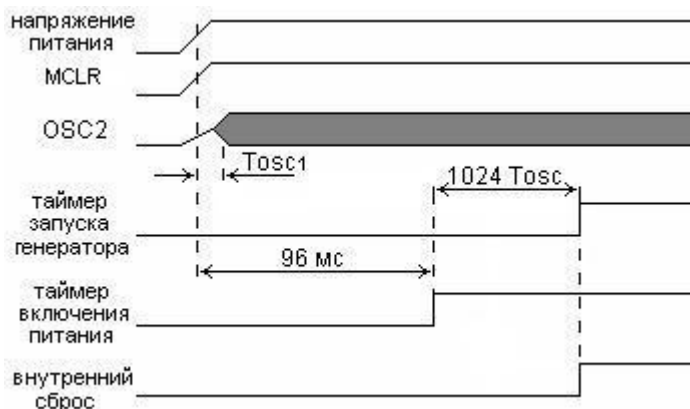
### Таймер включения питания PWRT

Таймер включения питания обеспечивает задержку включения (номинальное значение 96 мс) по сигналу схемы сброса включения питания. Это происходит после фронта внутреннего сигнала «сброса», или после фронта сигнала MCLR. Таймер включения питания работает на внутреннем RC генераторе. В течение этого времени микроконтроллер удерживается в состоянии сброса. В большинстве случаев эта задержка позволяет напряжению питания достигнуть номинального значения. Время задержки варьируется от микроконтроллера к микроконтроллеру и зависит от величины напряжения питания и температуры. Смотрите таблицу параметров.

### Таймер запуска генератора

Таймер запуска генератора обеспечивает дополнительную задержку в 1024 такта генератора после окончания задержки от таймера включения питания или выхода микроконтроллера из режима SLEEP в режимах XT или LF. Таймер запуска генератора считает каждый импульс генератора на входе OSC1. Счетчик начинает инкрементироваться после того, как амплитуда сигнала генератора достигнет порога входного буфера. Задержка гарантирует стабилизацию частоты генератора с кварцевым резонатором прежде, чем устройство выйдет из режима сброса. Длительность задержки зависит от частоты резонатора.

На рисунке ниже показана работа схемы таймера запуска генератора (распределение времени при запуске генератора). На этом рисунке показан низкочастотный генератор, время запуска которого превышает задержку таймера по включению питания. На Рис. 7: T<sub>OSC1</sub> - время, требуемое кварцевому генератору для запуска.



**Рис. 7** Время запуска генератора.

### Последовательность удержания микроконтроллера в состоянии сброса

При включении питания выполняется следующая последовательность удержания микроконтроллера в состоянии сброса: во-первых, внутренний сигнал «сброса» по включению питания увеличивается, пока не достигнет порогового уровня. Если сигнал MCLR находится в высоком уровне, то начинает работать таймер включения питания, после того, как он отсчитает 96 мс, включается таймер запуска генератора, если MCLR в низком уровне то таймеры запускаются после фронта этого сигнала. Обычно задержка от таймера включения питания больше, за исключением низкочастотных кварцевых резонаторов. Общее время задержки также изменяется в зависимости от конфигурации генератора. Ниже показано время задержки, в зависимости от конфигурации генератора (см. Таблица 2). На Рис. 8 отображены последовательности задержек.

Если напряжение питания устройства не соответствует спецификации электрических характеристик после окончания задержки от таймеров, то на выводе MCLR/Upp должен присутствовать низкий логический уровень, пока напряжение питания не достигнет номинального значения. Для большинства схем достаточно использования внешней RC цепочки.

Если сигнал сброса от MCLR подается во время нормальной работы микроконтроллера, то после его окончания таймеры включения питания и запуска генератора не работают, но запускается таймер задержки на обновление конфигурационных бит (типовое значение 1 мс).

Ниже (см. Таблица 2, Таблица 3) показаны состояния после сброса для некоторых битов и специальных регистров, в то время как (см. Таблица 4) показано состояние при инициализации для всех регистров.

**Таблица 2**  
Время задержки при различных видах сброса

Конфигурация генератора	Включение или снижение напряжения питания	Выход из режима SLEEP	Сброс от MCLR
XT, LF	сумма 96 мс и $1024 \cdot T_C$	$1024 \cdot T_C$	1 мс
EC, RC	сумма 96 мс и $1024 \cdot T_C$	-	1 мс

**Таблица 3**

Биты статуса и их значение после «сброса»

POR	BOR (если разрешен сброс по снижению питания, иначе значение не известно)	TO	PD	Тип «сброса»
0	0	1	1	Сброс по включению питания.
1	1	1	0	Выход из режима SLEEP по прерыванию (см. примечание).
1	1	0	1	Сброс от WDT при нормальном режиме работы (см. примечание).
1	1	0	0	Выход из режима SLEEP от WDT (см. примечание).
1	1	1	1	Сброс от MCLR (см. примечание).
1	0	1	1	Сброс по снижению напряжения питания.
x	x	1	1	Выполнение команды CLRWDT.

**Примечание:**

Для отмеченных видов «сброса», состояния битов статуса будут соответствовать приведенным в таблице, для случая если биты предварительно установлены в единицу.

**Таблица 4**

Условия сброса программного счетчика и регистра CPUSTA

Тип «сброса»	PCN:PCL	CPUSTA(3)	Задержка включения	
Сброс по включению питания	0000h	--11 1100	Сумма 96 мс и 1024*TC	
Сброс по снижению напряжения питания	0000h	--11 1110	Сумма 96 мс и 1024*TC	
Сброс от MCLR в режиме нормальной работы	0000h	--11 1111(4)	1 мс	
Сброс от MCLR в режиме SLEEP	0000h	--11 1111(4)	Большее из 1 мс или (только для режимов XT и LF) 1024*TC	
Сброс от WDT в режиме нормальной работы	0000h	--11 0111(4)	Нет	
Сброс от WDT во время режима SLEEP	0000h	--11 0011(4)	Для режимов XT и LF: 1024*TC	
Выход из режима SLEEP по прерыванию	GLINTD установлен	PC + 1	--11 1011(4)	Для режимов XT и LF: 1024*TC
	GLINTD сброшен	PC + 1(1)	--10 1011(4)	Для режимов XT и LF: 1024*TC

**Обозначения:**

и = не изменяется, x = не известно, - = не реализовано, читается как «0»

## Примечание:

1. При «пробуждении», выполняется эта команда. Далее команда выбирается в соответствии с вектором прерывания, а затем выполняется.
2. Программный счетчик = 0, то есть устройство переходит к вектору сброса и устанавливает регистры в состояние сброса по WDT.
3. Значение бита BOR известно только если разрешен сброс по снижению питания.
4. Состояние статусных битов соответствует приведенным в таблице для случая их предварительной установки в единицу.

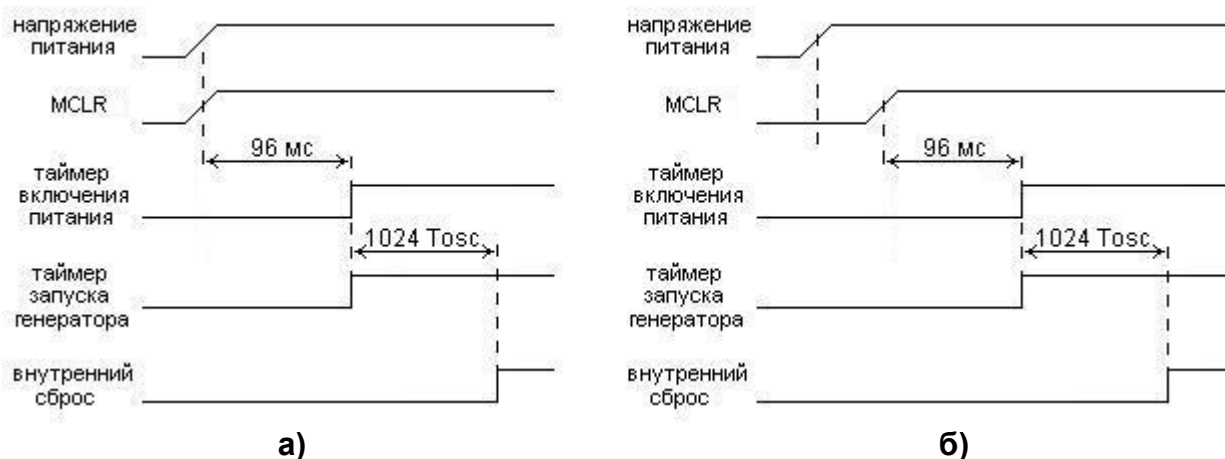


Рис. 8

Последовательность удержания в режиме сброса по включению питания:

- а) MCLR подсоединен к напряжению питания
- б) MCLR не подключен к напряжению питания

Таблица 5  
Значения регистров при «сбросе»

Регистр	Адрес	Сброс по включению или снижению напряжения питания	Сброс от MCLR или от сторожевого таймера	Выход из режима SLEEP по прерыванию
<b>Вне банка</b>				
INDF0	00h	N/A	N/A	N/A
FSR0	01h	0000 0000	uuuu uuuu	uuuu uuuu
PCL	02h	0000h	0000h	PC+1 <sup>(2)</sup>
PCLATH	03h	0000 0000	0000 0000	uuuu uuuu
ALUSTA	04h	1111 0000	1111 0000	1111 uuuu
T0STA	05h	0000 000-	0000 000-	0000 000-
CPUSTA <sup>(3)</sup>	06h	--11 11q0	--11 qq1u	--uu qquu
INTSTA	07h	0000 0000	0000 0000	uuuu uuuu <sup>(1)</sup>
INDF1	08h	N/A	N/A	N/A
FSR1	09h	0000 0000	uuuu uuuu	uuuu uuuu
WREG	0Ah	0000 0000	0000 0000	uuuu uuuu
TMR0L	0Bh	0000 0000	0000 0000	uuuu uuuu

## Спецификация 1886BE2У, К1886BE2У

Регистр	Адрес	Сброс по включению или снижению напряжения питания	Сброс от MCLR или от сторожевого таймера	Выход из режима SLEEP по прерыванию
TMR0H	0Ch	0000 0000	0000 0000	uuuu uuuu
TBLPTRL	0Dh	0000 0000	0000 0000	uuuu uuuu
TBLPTRH	0Eh	0000 0000	0000 0000	uuuu uuuu
BSR	0Fh	0000 0000	0000 0000	uuuu uuuu
<b>Банк 0</b>				
PORTA <sup>(4,6)</sup>	10h	0-xx 11xx	0-uu 11uu	u-uu uuuu
DDRB	11h	1111 1111	1111 1111	uuuu uuuu
PORTB <sup>(4)</sup>	12h	xxxx xxxx	uuuu uuuu	uuuu uuuu
RCSTA1	13h	0000 -000	0000 -000	uuuu -uuu
RCREG1	14h	0000 0000	0000 0000	uuuu uuuu
TXSTA1	15h	0000 --10	0000 --10	uuuu --uu
TXREG1	16h	0000 0000	0000 0000	uuuu uuuu
SPBRG1	17h	0000 0000	0000 0000	uuuu uuuu
<b>Банк 1</b>				
DDRC <sup>(5)</sup>	10h	1111 1111	1111 1111	uuuu uuuu
PORTC <sup>(4,5)</sup>	11h	xxxx xxxx	uuuu uuuu	uuuu uuuu
DDRD <sup>(5)</sup>	12h	1111 1111	1111 1111	uuuu uuuu
PORTD <sup>(4,5)</sup>	13h	xxxx xxxx	uuuu uuuu	uuuu uuuu
DDRE <sup>(5)</sup>	14h	---- 1111	---- 1111	---- uuuu
PORTE <sup>(4,5)</sup>	15h	---- xxxx	---- uuuu	---- uuuu
PIR1	16h	x000 0010	u000 0010	uuuu uuuu <sup>(1)</sup>
PIE1	17h	0000 0000	0000 0000	uuuu uuuu
<b>Банк 2</b>				
TMR1	10h	0000 0000	0000 0000	uuuu uuuu
TMR2	11h	0000 0000	0000 0000	uuuu uuuu
TMR3L	12h	0000 0000	0000 0000	uuuu uuuu
TMR3H	13h	0000 0000	0000 0000	uuuu uuuu
PR1	14h	1111 1111	1111 1111	uuuu uuuu
PR2	15h	1111 1111	1111 1111	uuuu uuuu
PR3/CA1L	16h	0000 0000	0000 0000	uuuu uuuu
PR3/CA1H	17h	0000 0000	0000 0000	uuuu uuuu
<b>Банк 3</b>				
PW1DCL	10h	00-- ----	00-- ----	uu-- ----
PW2DCL	11h	000- ----	000- ----	uuu- ----
PW1DCH	12h	0000 0000	0000 0000	uuuu uuuu
PW2DCH	13h	0000 0000	0000 0000	uuuu uuuu
CA2L	14h	0000 0000	0000 0000	uuuu uuuu
CA2H	15h	0000 0000	0000 0000	uuuu uuuu
TCON1	16h	0000 0000	0000 0000	uuuu uuuu
TCON2	17h	0000 0000	0000 0000	uuuu uuuu
<b>Банк 4</b>				
PIR2	10h	000- 0010	000- 0010	uuu- uuuu <sup>(1)</sup>

## Спецификация 1886BE2У, К1886BE2У

Регистр	Адрес	Сброс по включению или снижению напряжения питания	Сброс от MCLR или от сторожевого таймера	Выход из режима SLEEP по прерыванию
PIE2	11h	000- 0000	000- 0000	uuu- uuuu
-	12h	---- ----	---- ----	---- ----
RCSTA2	13h	0000 -000	0000 -000	uuuu -uuu
RCREG2	14h	0000 0000	0000 0000	uuuu uuuu
TXSTA2	15h	0000 --10	0000 --10	uuuu --uu
TXREG2	16h	0000 0000	0000 0000	uuuu uuuu
SPBRG2	17h	0000 0000	0000 0000	uuuu uuuu
<b>Банк 5</b>				
DDRF	10h	1111 1111	1111 1111	uuuu uuuu
PORTF <sup>(4)</sup>	11h	0000 0000	0000 0000	uuuu uuuu
DDRG	12h	1111 1111	1111 1111	uuuu uuuu
PORTG <sup>(4)</sup>	13h	0000 0000	0000 0000	uuuu uuuu
ADCON0	14h	0000 -0-0	0000 -0-0	uuuu uuuu
ADCON1	15h	0000- 0000	000- 0000	uuuu uuuu
ADRESL	16h	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESH	17h	xxxx xxxx	uuuu uuuu	uuuu uuuu
<b>Банк 6</b>				
SSPADD	10h	0000 0000	0000 0000	uuu uuuu
SSPCON1	11h	0000 0000	0000 0000	uuuu uuuu
SSPCON2	12h	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	13h	0000 0000	0000 0000	uuuu uuuu
SSPBUF	14h	0000 0000	0000 0000	uuuu uuuu
-	15h	---- ----	---- ----	---- ----
-	16h	---- ----	---- ----	---- ----
-	17h	---- ----	---- ----	---- ----
<b>Банк 7</b>				
PW3DCL	10h	000- ----	000- ----	uuu- ----
PW3DCH	11h	0000 0000	0000 0000	uuuu uuuu
CA3L	12h	0000 0000	0000 0000	uuuu uuuu
CA3H	13h	0000 0000	0000 0000	uuuu uuuu
CA4L	14h	0000 0000	0000 0000	uuuu uuuu
CA4H	15h	0000 0000	0000 0000	uuuu uuuu
TCON3	16h	-000 0000	-000 0000	-uuu uuuu
WRDIV	17h	0000 0101	0000 0101	uuuu uuuu
<b>Вне банка</b>				
PRODL	18h	0000 0000	0000 0000	uuuu uuuu
PRODH	19h	0000 0000	0000 0000	uuuu uuuu

**Обозначение:**

u = не изменяется, x = неизвестно, - = не реализовано, читается как «0»,  
q = значение зависит от условия.

**Примечание:**

1. Один бит или более в INTSTA, PIR1, PIR2 будет изменен (чтобы произошел выход).
2. Когда выход из режима SLEEP происходит по прерыванию и бит GLINTD сброшен, РС загружается вектором прерывания.  
См. Таблица 4 для значений по сбросу в особых условиях.
3. Это значение, которое будет в триггере-защелке порта вывода.
4. Когда запрограммирован режим микропроцессора или расширенного микроконтроллера, работа этого порта не зависит от этих регистров.
5. При любом типе сброса устройства эти выводы конфигурируются как входы.

### Сброс по снижению напряжения питания

Микроконтроллеры имеют на кристалле схему сброса по снижению напряжения питания. Эта схема переводит микроконтроллер в режим сброса, когда напряжение питания опускается ниже установленного уровня, что гарантирует прекращение выполнения программ при выходе напряжения питания за установленные нормы. Прежде чем использовать схему сброса по снижению напряжения питания, проверьте электрические характеристики, чтобы удостовериться в том, что она отвечает вашим требованиям. Включение или выключение схемы сброса производится битом BODEN в слове конфигурации.

Работа схемы сброса: если напряжение питания опускается ниже  $U_{bor}$  (типичное значение 4.0 В), произойдет сброс по снижению напряжения питания. Микроконтроллер находится в состоянии сброса, пока напряжение питания не поднимется выше  $U_{bor}$ . Затем начинает работать таймер включения питания, после отсчёта 96 мс включается таймер запуска генератора (для режимов LP и XT). Это удерживает микроконтроллер в состоянии сброса 96 мс плюс  $1024 \cdot T_C$ . Если напряжение питания опускается ниже  $U_{bor}$  во время работы таймера включения питания, то микроконтроллер возвращается в состояние сброса и таймеры инициализируются заново. После подъема питания, таймеры начнут отсчёт временной задержки. На Рис. 9 показаны типовые ситуации сброса.

В некоторых приложениях параметры внутренней схемы сброса по снижению питания не удовлетворяют требованиям. В этом случае должна быть применена внешняя схема сброса по снижению напряжения питания.

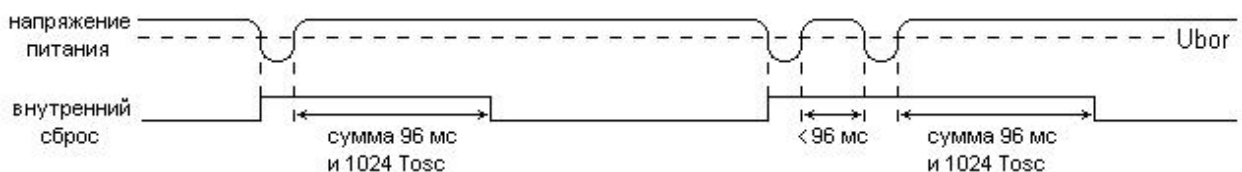


Рис. 9 Сброс по снижению напряжения питания

### Прерывания

Микроконтроллеры имеют 18 источников прерывания:

- внешнее прерывание от вывода PA0/INT;
- изменение уровня сигнала на выводах PB7:PB0;
- переполнение таймера TMR0;
- переполнение таймера TMR1;
- переполнение таймера TMR2;
- переполнение таймера TMR3;
- буфер передатчика USART1 пуст;
- буфер приемника USART1 заполнен;
- буфер передатчика USART2 пуст;
- буфер приемника USART2 заполнен;
- прерывания от модуля SSP;
- прерывания от модуля SSP при возникновении коллизии на шине I2C;
- завершение преобразования АЦП;
- захват сигнала 1;
- захват сигнала 2;
- захват сигнала 3;
- захват сигнала 4;
- изменение сигнала на выводе PA1/T0CLK.

Прерываниями управляют шесть регистров: CPUSTA, INTSTA, PIE1, PIR1, PIE2, PIR2.

Регистр CPUSTA содержит бит глобального запрещения прерываний GLINTD (Global Interrupt Disable). Если этот бит установлен, все прерывания запрещены. Этот бит является частью функциональных возможностей ядра микроконтроллера и описывается в разделе «Обработка прерываний».

Когда происходит прерывание, бит GLINTD автоматически устанавливается для запрета дальнейших прерываний, адрес возврата записывается в стек, а в РС загружается адрес вектора прерываний. Существует 4 вектора прерываний. Каждый адрес вектора прерываний предназначен для определенного источника прерываний (кроме периферийных прерываний, у которых один и тот же адрес).

Эти источники следующие:

- внешнее прерывание от вывода PA0/INT;
- переполнение таймера TMR0;
- изменение сигнала на выводе PA1/T0CLK;
- любое периферийное прерывание.

Переход по адресу вектора прерывания не сбрасывает флаг запроса прерывания. Флаг запроса прерывания должен быть сброшен в программе обработки прерывания перед разрешением прерываний, чтобы предотвратить повторный переход на обработку прерываний. В программе обработки периферийных прерываний источник прерывания можно идентифицировать проверкой флагов запроса прерываний.

Когда выполняется условие прерывания, индивидуальные флаги запросов прерываний устанавливаются независимо от состояния бита GLINTD и соответствующих битов маски.



При внешнем прерывании происходит задержка прерывания. Для команд, выполняющихся за два машинных цикла, задержка длиннее на один машинный цикл.

Возврат из программы обработки прерываний производится по команде RETFIE. При выполнении команды происходит восстановление программного счетчика (PC) из стека и сбрасывается бит GLINTD (чтобы разрешить прерывания).

### Регистр состояния прерываний (INTSTA)

Регистр INTSTA содержит флаги запроса прерываний и биты разрешений для не периферийных прерываний. Бит PEIF (флаг запроса периферийных прерываний) только читается, и объединяет по «ИЛИ» все не замаскированные флаги запросов периферийных прерываний в регистрах PIR (регистры 5-4 и 5-5). Биты флагов запросов прерываний устанавливаются по заданным условиям, даже если соответствующий бит разрешения прерывания сброшен (прерывание запрещено), или бит GLINTD установлен (все прерывания запрещены).

Следует с осторожностью сбрасывать любой разрешающий бит регистра INTSTA, когда прерывания разрешены (бит GLINTD сброшен). Если какие-либо флаги запроса прерывания (T0IF, INTF, T0CKIF, или PEIF) устанавливаются в том же машинном цикле, в котором соответствующие биты разрешения прерывания сбрасываются, устройство переходит по адресу сброса (0x00). Прежде, чем запретить какое-либо прерывание, сбросом разрешающего бита регистра INTSTA, необходимо предварительно установить бит GLINTD, т.е. запретить все прерывания.

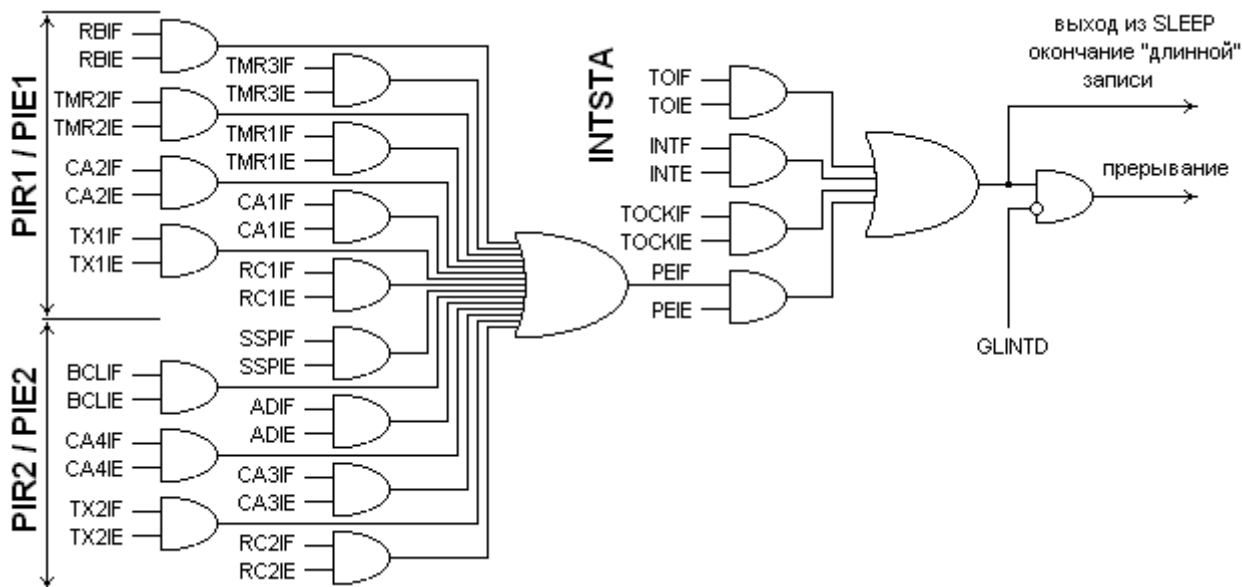


Рис. 10 Структурная схема логики прерываний

**Таблица 6**

Регистр INTSTA (адрес: 07h, не зависит от номера банка)

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	INTE	T0IE
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7</b>	<p><b>PEIF:</b> Флаг запроса периферийного прерывания. Этот бит является объединением по «ИЛИ» всех флагов запросов периферийных прерываний логически умноженных по «И» на соответствующие биты разрешения прерываний. Логика прерываний направляет выполнение программы по адресу 20h), когда происходит задержка периферийного прерывания. 1 = есть не обработанное периферийное прерывание; 0 = нет не обработанного периферийного прерывания.</p>						
<b>бит 6</b>	<p><b>T0CKIF:</b> Флаг внешнего запроса прерывания от вывода PA1/T0CLK. Сбрасывается программно программой обработки прерывания. Адрес вектора 18h. 1 = на выводе PA1/T0CLK обнаружен заданный фронт сигнала; 0 = на выводе PA1/T0CLK не обнаружен заданный фронт сигнала.</p>						
<b>бит 5</b>	<p><b>T0IF:</b> Флаг запроса прерывания по переполнению таймера 0. Сбрасывается программно программой обработки прерывания. Адрес вектора 10h. 1 = таймер TMR0 переполнен; 0 = таймер TMR0 не переполнен.</p>						
<b>бит 4</b>	<p><b>INTF:</b> Флаг внешнего запроса прерывания от вывода PA0/INT. Сбрасывается программно программой обработки прерывания. Адрес вектора 08h. 1 = на выводе PA0/INT обнаружен заданный фронт сигнала; 0 = на выводе PA0/INT не обнаружен заданный фронт сигнала.</p>						
<b>бит 3</b>	<p><b>PEIE:</b> Бит разрешение периферийных прерываний. Этот бит действует, как бит глобального разрешения периферийных прерываний, чьи соответствующие биты разрешения прерываний также установлены. 1 = периферийные прерывания разрешены; 0 = периферийные прерывания запрещены.</p>						
<b>бит 2</b>	<p><b>T0CKIE:</b> Бит разрешения внешнего прерывания от вывода PA1/T0CLK. 1 = прерывание разрешено; 0 = прерывание запрещено.</p>						
<b>бит 1</b>	<p><b>T0IE:</b> Бит разрешения прерывания по переполнению таймера 0. 1 = прерывание разрешено; 0 = прерывание запрещено.</p>						

<b>бит 0</b>	<b>INTE:</b> Бит разрешения внешнего прерывания на выводе PA0/INT. 1 = прерывание разрешено; 0 = прерывание запрещено.
--------------	--

Обозначения:

R = бит для чтения; W = бит с возможностью записи;

U = бит не реализован, читается как 0;

-п = значение бита после сброса по включению питания:

1 = установлен;

0 = сброшен;

x = значение не известно.

**Регистры разрешения периферийных прерываний PIE1 и PIE2**

Эти регистры содержат индивидуальные биты разрешения периферийных прерываний.

**Таблица 7**  
Регистр PIE1 (адрес: 17h, банк 1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TX1IE	RC1IE
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7</b>		<b>RBIE:</b> Бит разрешения прерывания по изменению уровня сигнала на входах PORTB. 1 = прерывание разрешено; 0 = прерывание запрещено.					
<b>бит 6</b>		<b>TMR3IE:</b> Бит разрешения прерывания по переполнению таймера 3. 1 = прерывание разрешено; 0 = прерывание запрещено.					
<b>бит 5</b>		<b>TMR2IE:</b> Бит разрешения прерывания по переполнению таймера 2. 1 = прерывание разрешено; 0 = прерывание запрещено.					
<b>бит 4</b>		<b>TMR1IE:</b> Бит разрешения прерывания по переполнению таймера 1. 1 = прерывание разрешено; 0 = прерывание запрещено.					
<b>бит 3</b>		<b>CA2IE:</b> Бит разрешения прерывания по Захвату 2. 1 = прерывание разрешено; 0 = прерывание запрещено.					
<b>бит 2</b>		<b>CA1IE:</b> Бит разрешения прерывания по Захвату 1. 1 = прерывание разрешено; 0 = прерывание запрещено.					

бит 1	<b>TX1IE:</b> Бит разрешения прерывания от передатчика USART1 (буфер передатчика пуст). 1 = прерывание разрешено; 0 = прерывание запрещено.
бит 0	<b>RC1IE:</b> Бит разрешения прерывания от приемника USART1 (в буфере приемника есть данные). 1 = прерывание разрешено; 0 = прерывание запрещено.

Обозначения:

R = бит для чтения; W = бит с возможностью записи; U = бит не реализован, читается как 0;

-n = значение бита после сброса по включению питания: 1 = установлен; 0 = сброшен; x = значение не известно.

**Таблица 8**  
Регистр PIE2 (адрес: 11h, банк 4)

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
SSPIE	BCLIE	ADIE	—	CA4IE	CA3IE	TX2IE	RC2IE
бит 7	6	5	4	3	2	1	бит 0
бит 7	<b>SSPEI:</b> Бит разрешения прерываний от последовательного синхронного порта. 1 = прерывание от SSP разрешено; 0 = прерывание от SSP запрещено.						
бит 6	<b>BCLIE:</b> Бит разрешения прерываний при возникновении коллизий шины. 1 = прерывание при возникновении коллизий разрешено; 0 = прерывание при возникновении коллизий запрещено.						
бит 5	<b>ADIE:</b> Бит разрешения прерываний по окончанию преобразования АЦП. 1 = прерывание от модуля АЦП разрешено; 0 = прерывание от модуля АЦП запрещено.						
бит 4	Не реализовано: читается как «0».						
бит 3	<b>CA4IE:</b> Бит разрешения прерывания по Захвату 4. 1 = прерывание разрешено; 0 = прерывание запрещено.						
бит 2	<b>CA3IE:</b> Бит разрешения прерывания по Захвату 3. 1 = прерывание разрешено; 0 = прерывание запрещено.						
бит 1	<b>TX2IE:</b> Бит разрешения прерывания от передатчика USART2 (буфер передатчика пуст). 1 = прерывание разрешено; 0 = прерывание запрещено.						
бит 0	<b>RC2IE:</b> Бит разрешения прерывания от приемника USART2 (в буфере приемника есть данные). 1 = прерывание разрешено; 0 = прерывание запрещено.						

Обозначения:

R = бит для чтения; W = бит с возможностью записи; U = бит не реализован, читается как 0;  
 -n = значение бита после сброса по включению питания: 1 = установлен; 0 = сброшен; x = значение не известно.

### Регистры запроса периферийных прерываний PIR1 и PIR2

Эти регистры содержат индивидуальные флаги запросов периферийных прерываний.

Примечание:

Флаги запроса прерываний устанавливаются при возникновении условий прерываний, вне зависимости от состояния флага общего запрета прерываний GLINTD и соответствующих флагов разрешения периферийных прерываний. Перед разрешением прерывания, пользователь должен сбросить флаги запросов прерываний, чтобы программа не перешла незамедлительно к подпрограмме обработки периферийных прерываний.

**Таблица 9**  
Регистр PIR1 (адрес: 16h, банк 1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R-0
RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TX1IF	RC1IF
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7</b>		<b>RBIF:</b> Флаг запроса прерывания по изменению уровня сигнала на входах PORTB. 1 = один из сигналов на входе PORTB изменился (программа должна устранить условие несовпадения); 0 = ни один из сигналов на входе PORTB не изменился.					
<b>бит 6</b>		<b>TMR3IF:</b> Флаг запроса прерывания по переполнению таймера 3. <u>Если Захват 1 разрешен (CA1/PR3=1):</u> 1 = произошло переполнение таймера 3; 0 = переполнения таймера 3 не произошло. <u>Если Захват 1 запрещен (CA1/PR3= 0):</u> 1 = значение таймера 3 перешло к 0000h от значения, равного величине из регистра периода PR3H:PR3L; 0 = значение таймера 3 не перешло к 0000h от значения, равного величине из регистра периода PR3H:PR3L.					
<b>бит 5</b>		<b>TMR2IF:</b> Флаг запроса прерывания по переполнению таймера 2. 1 = значение таймера 2 перешло к 0000h от значения, равного величине из регистра периода PR2; 0 = значение таймера 2 не перешло к 0000h от значения, равного величине из регистра периода PR2.					

<b>бит 4</b>	<b>TMR1IF:</b> Флаг запроса прерывания по переполнению таймера 1. <u>Если таймер 1 в 8-разрядном режиме (бит T16=0):</u> 1 = значение таймера 1 перешло к 0000h от значения, равного величине из регистра периода PR1; 0 = значение таймера 1 не перешло к 0000h от значения, равного величине из регистра периода PR1. <u>Если таймер 1 в 16-ти разрядном режиме (бит T16=1):</u> 1 = значение TMR2:TMR1 перешло к 0000h от значения, равного величине из регистра периода PR2:PR1; 0 = значение TMR2:TMR1 не перешло к 0000h от значения, равного величине из регистра периода PR2:PR1.
<b>бит 3</b>	<b>CA2IF:</b> Флаг запроса прерывания по Захвату 2. 1 = произошел захват по изменению сигнала на выводе PB1/CAP2; 0 = захвата не было.
<b>бит 2</b>	<b>CA1IF:</b> Флаг запроса прерывания по Захвату 1. 1 = произошел захват по изменению сигнала на выводе PB0/CAP1; 0 = захвата не было.
<b>бит 1</b>	<b>TX1IF:</b> Флаг запроса прерывания от передатчика USART1. Флаг устанавливается и сбрасывается аппаратно, доступен только для чтения. 1 = буфер передатчика USART1 пуст; 0 = буфер передатчика USART1 заполнен.
<b>бит 0</b>	<b>RC1IF:</b> Флаг запроса прерывания от приемника USART1. Флаг устанавливается и сбрасывается аппаратно, доступен только для чтения. 1 = в буфере приемника USART1 есть данные; 0 = буфер приемника USART1 пуст.

Обозначения:

R = бит для чтения; W = бит с возможностью записи; U = бит не реализован, читается как 0;

-n = значение бита после сброса по включению питания: 1 = установлен; 0 = сброшен; x = значение не известно.

Таблица 10

Регистр PIR2 (адрес: 10h, банк 4)

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R-1	R-0
SSPIF	BCLIF	ADIF	—	CA4IF	CA3IF	TX2IF	RC2IF
бит 7	6	5	4	3	2	1	бит 0
бит 7	<p><b>SSPIF:</b> Флаг запроса прерывания от последовательного синхронного порта  <b>1 = выполнено условие возникновения прерывания от модуля СПП.</b> Бит должен быть программно сброшен перед возвратом из подпрограммы обработки прерываний. Следующие условия устанавливают этот бит:  <u>SPI:</u>                      произошла прием/передача.  <u>Режим ведущего/ведомого I2C:</u>                      произошла прием/передача.  <u>режим ведущего I2C:</u>                      условие START было выполнено модулем;                      условие STOP было выполнено модулем;                      условие повторный START было выполнено модулем;                      передача бита подтверждения было выполнено модулем;                      принято условие START, пока модуль не работал (система с несколькими ведущими);                      принято условие STOP, пока модуль не работал (система с несколькими ведущими).  <b>0 = условие прерывания от модуля СПП не возникло.</b></p>						
бит 6	<p><b>BCLIF:</b> Флаг запроса прерываний при возникновении коллизий на шине.  <b>1 = возникла коллизия на шине в модуле SSP, при конфигурации I2C в режиме ведущего;</b>  <b>0 = коллизия не возникла.</b></p>						
бит 5	<p><b>ADIF:</b> Флаг запроса прерывания от модуля АЦП.  <b>1 = преобразование в АЦП завершено;</b>  <b>0 = преобразование в АЦП не завершено.</b></p>						
бит 4	Не реализовано: читается как «0».						
бит 3	<p><b>CA4IF:</b> Флаг запроса прерывания по Захвату 4.  <b>1 = произошел захват по изменению сигнала на выводе PE3/CAP4;</b>  <b>0 = захвата не было.</b></p>						
бит 2	<p><b>CA3IF:</b> Флаг запроса прерывания по Захвату 3.  <b>1 = произошел захват по изменению сигнала на выводе PG4/CAP3;</b>  <b>0 = захвата не было.</b></p>						
бит 1	<p><b>TX2IF:</b> Флаг запроса прерывания от передатчика USART2. Флаг устанавливается и сбрасывается аппаратно, доступен только для чтения.  <b>1 = буфер передатчика USART2 пуст;</b>  <b>0 = буфер передатчика USART2 заполнен.</b></p>						

<b>бит 0</b>	<b>RC2IF:</b> Флаг запроса прерывания от приемника USART2. Флаг устанавливается и сбрасывается аппаратно, доступен только для чтения. 1 = в буфере приемника USART2 есть данные; 0 = буфер приемника USART2 пуст.
--------------	---

### Обозначения:

R = бит для чтения; W = бит с возможностью записи; U = бит не реализован, читается как 0;

-n = значение бита после сброса по включению питания: 1 = установлен; 0 = сброшен; x = значение не известно.

## Обработка прерываний

Бит глобального запрещения прерываний GLINTD(CPUSTA<4>) разрешает все немаскированные прерывания (если сброшен), или запрещает все прерывания (если установлен). Индивидуальные прерывания запрещены через соответствующий бит разрешения в регистре INTSTA. Для запрета периферийных прерываний необходимо, чтобы был сброшен бит глобального разрешения периферийных прерываний PEIE или сброшен бит разрешения соответствующего периферийного прерывания. Запрещение периферийных прерываний через бит глобального разрешения периферийных прерываний, запрещает все периферийные прерывания. Бит GLINTD устанавливается при сбросе микроконтроллера (прерывания запрещены).

Команда RETFIE сбрасывает бит GLINTD для разрешения прерываний и загружает счетчик команд значением из вершины стека. Когда происходит прерывание, бит GLINTD автоматически устанавливается для запрета дальнейших прерываний, адрес возврата записывается в стек и счетчик команд загружается адресом вектора прерывания. Существует 4 вектора прерывания, что способствует сокращению задержки при обработке прерываний. Флаги запросов прерываний должны быть сброшены в программе перед разрешением прерываний, чтобы избежать повторного вызова.

Вектор периферийных прерываний имеет множество источников прерываний. В программе обслуживания периферийного прерывания, источник прерывания можно определить проверкой флагов запроса прерывания.

Микроконтроллер имеет 4 вектора прерываний. Адреса векторов и их приоритеты показаны в Таблица 6. Если происходит запрос двух прерываний одновременно, прерывание с большим приоритетом будет обслуживаться в первую очередь. Это означает, что адрес вектора именно этого прерывания будет загружен в счетчик команд (PC).



**Таблица 11**

Приоритеты и адреса векторов прерываний

Адрес	Вектор	Приоритет
0008h	Внешнее прерывание на выводе PA0/INT (INTF)	1 (самый высокий)
0010h	Прерывание по переполнению TMR0 (T0IF)	2
0018h	Внешнее прерывание по PA1/T0CLK (T0CKIF)	3
0020h	Периферийные прерывания (PEIF)	4 (самый низкий)

**Примечание:**

- Индивидуальные флаги запроса прерывания устанавливаются независимо от состояния соответствующего маскирующего бита или бита GLINTD.
- Прежде, чем запретить какое-либо прерывание, сбросом разрешающего бита регистра INTSTA, бит GLINTD должен быть установлен (общий запрет прерываний).

**Прерывание от вывода PA0/INT**

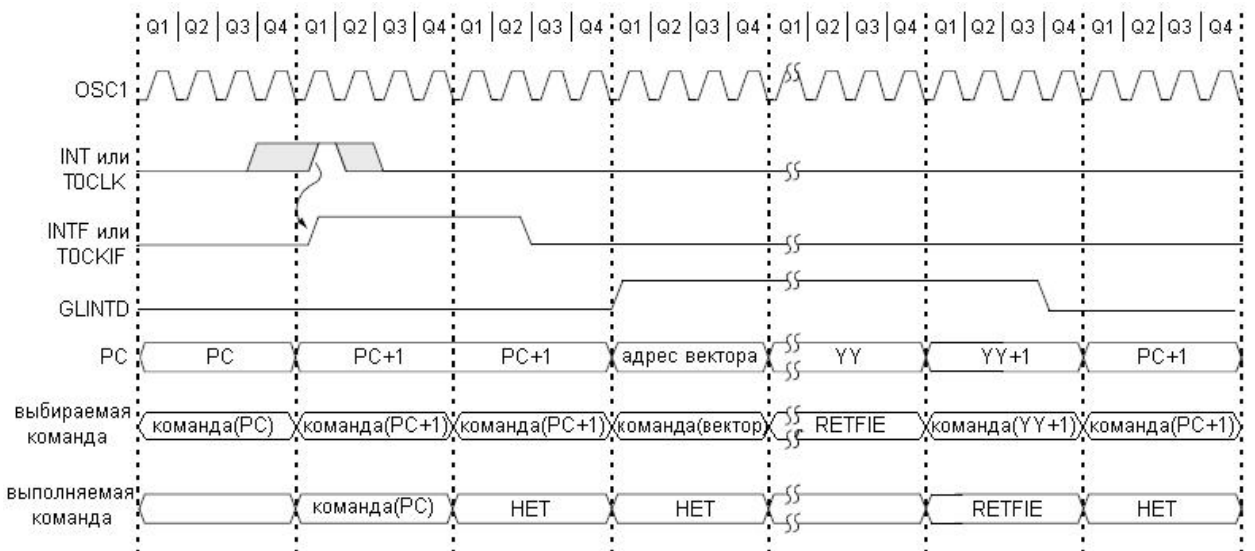
Внешнее прерывание от вывода PA0/INT происходит либо по переднему фронту сигнала, если бит INTEDG (T0STA<7>) установлен, либо по заднему фронту, если бит INTEDG сброшен. Когда активный фронт сигнала появляется на входе PA0/INT, бит INTF (INTSTA<4>) устанавливается. Это прерывание может быть запрещено сбросом бита INTE (INTSTA<0>). Прерывание на выводе INT может выводить микроконтроллер из режима SLEEP. Смотрите раздел описания режима SLEEP.

**Прерывание от вывода PA1/T0CLK**

Внешнее прерывание от вывода PA1/T0CLK происходит либо по переднему фронту сигнала, если бит T0SE (T0STA<6>) установлен, либо по заднему фронту, если бит T0SE сброшен. Когда активный фронт сигнала появляется на выводе PA1/T0CLK, бит T0CKIF (INTSTA<6>) устанавливается. Прерывание может быть запрещено сбросом бита T0CKIE (INTSTA<2>). Прерывание PA1/T0CLK может выводить микроконтроллер из режима SLEEP. Смотрите раздел описания режима SLEEP.

**Периферийные прерывания**

Установленный флаг запроса периферийных прерываний PEIF показывает, что произошло, по крайней мере, одно периферийное прерывание. Бит PEIF только для чтения и является объединением по «ИЛИ» всех флагов запросов периферийных прерываний, логически умноженных по «И» на соответствующие биты разрешения прерываний в регистрах PIE. Некоторые периферийные прерывания могут выводить микроконтроллер из режима SLEEP. Смотрите раздел описания режима SLEEP.



**Рис. 11** Временная диаграмма обработки прерываний от выводов INT или TOCLK

### Сохранение регистров при прерывании

Во время прерывания, в стеке сохраняется только значение PC для возврата. Сохранение других регистров необходимо реализовать программно. В Пример 1 показано сохранение и восстановление информации в подпрограмме обработки прерывания. Этот пример приведен для простой схемы прерываний, где не может произойти вложенности прерываний. Содержимое регистров сохраняется в области GPR вне банков.

В Пример 2 показано сохранение и восстановление информации для случая когда требуется вложение прерываний. В приведенном примере может выполняться до максимум 6 уровней вложения прерываний. BSR хранится в области GPR вне банков, в то время как другие регистры будут храниться в определенном банке. Таким образом, эта программа может осуществить 6 записей блоков со значениями сохраняемых регистров. Для работы программа требует выделенного регистра косвенной адресации FSR0.

Сегменты кода PUSH и POP (запись в стек и чтение стека) могут быть либо в каждой подпрограмме обработки прерывания, либо могут быть вызываемыми подпрограммами. В зависимости от конкретного приложения, также может потребоваться сохранение и других регистров.

#### Пример 1

##### Сохранение регистров при прерывании (простой вариант)

- ; Адреса, которые используются для хранения значений регистров находятся во
- ; внутренней памяти данных. Диапазон адресов 1Ah – 1Fh (вне банков).
- ; При помощи команды MOVFP может быть сохранено и восстановлено до 6
- ячеек.
- ; Эта команда не влияет на биты состояний и не нарушает значение регистра
- WREG.

```
UNBANK1 EQU 0x01A; адрес для сохранения первой ячейки
UNBANK2 EQU 0x01B; адрес для сохранения второй ячейки
UNBANK3 EQU 0x01C; адрес для сохранения третьей ячейки
UNBANK4 EQU 0x01D; адрес для сохранения четвертой ячейки
```

```

UNBANK5 EQU 0x01E; адрес для сохранения пятой ячейки
           ; (метка не используется в программе)
UNBANK6 EQU 0x01F; адрес для сохранения шестой ячейки
           ; (метка не используется в программе)

           ; Адрес вектора прерывания
PUSH MOVFP ALUSTA, UNBANK1 ; Запись в стек значения ALUSTA
      MOVFP BSR, UNBANK2   ; Запись в стек значения BSR
      MOVFP WREG, UNBANK3  ; Запись в стек значения WREG
      MOVFP PCLATH, UNBANK4 ; Запись в стек значения PCLATH
      :
; Код программы обработки прерывания
      :
POP  MOVFP UNBANK4, PCLATH ; Восстановление значения PCLATH
     MOVFP UNBANK3, WREG   ; Восстановление значения WREG
     MOVFP UNBANK2, BSR    ; Восстановление значения BSR
     MOVFP UNBANK1, ALUSTA ; Восстановление значения ALUSTA

      RETFIE ; Возврат из прерывания (разрешает прерывания)
    
```

**Пример 2**

Сохранение регистров при прерывании (вариант с поддержкой вложенных прерываний)

```

; Адреса, которые используются для хранения значений регистров находятся во
; внутренней памяти данных. Для сохранения значений регистра BSR
используется
; область данных с адресами 1Ah - 1Fh (вне банков). Таким образом может быть
; сохранено до 6 блоков значений регистров. Программа использует регистр FSR0
; (и биты его управления FS1 и FS0 в регистре ALUSTA).

Nobank_FSR EQU 0x40
Bank_FSR   EQU 0x41
ALU_Temp   EQU 0x42
WREG_TEMP  EQU 0x43
BSR_S1     EQU 0x01A; адрес первой ячейки для сохранения BSR
BSR_S2     EQU 0x01B; 0x1Ah-0x1Fh - адреса ячеек для сохранения BSR
BSR_S3     EQU 0x01C
BSR_S4     EQU 0x01D
BSR_S5     EQU 0x01E
BSR_S6     EQU 0x01F

; Инициализация
      CALL CLEAR_RAM ; Очистка ОЗУ данных
;
INIT_POINTERS ; подготовка параметров для процедур POP и
PUSH
      CLRF BSR, F ; установка банков в 0
      CLRF ALUSTA, F ; переключение FSR0 в режим
      BSF ALUSTA, FS1 ; автоинкрементирования
    
```

	CLRF	WREG, F	; сброс WREG
	MOVLW	BSR_S1	; загрузка FSR0 значением первого адреса
для	MOVWF	FSR0	; сохранения BSR
	MOVWF	Nobank_FSR	
	MOVLW	0x20	
	MOVWF	Bank_FSR	
	:		
;		Код Вашей программы	
	:		
			; Адрес вектора прерывания
PUSH	BSF	ALUSTA, FS0	; FSR0 - режим автоинкрементирования
	BCF	ALUSTA, FS1	
	MOVFP	BSR, INDF0	; сохранение регистра BSR
	CLRF	BSR, F	; установка банка 0 для периферийных
регистров и			
			; ОЗУ данных
	MOVFP	ALUSTA, ALU_Temp	
используемого	MOVFP	FSR0, Nobank_FSR	; сохранение значения FSR,
			; для сохранения BSR
	MOVFP	WREG, WREG_TEMP	
используемого	MOVFP	Bank_FSR, FSR0	; восстановление значения FSR,
			; для сохранения других значений
	MOVFP	ALU_Temp, INDF0	; запись в стек значения ALUSTA
	MOVFP	WREG_TEMP, INDF0	; запись в стек значения WREG
	MOVFP	PCLATH, INDF0	; запись в стек значения PCLATH
	MOVFP	FSR0, Bank_FSR	; сохранение значения FSR, используемого
			; для сохранения других значений
используемого	MOVFP	Nobank_FSR, FSR0	; восстановление значения FSR,
			; для сохранения BSR
	:		
;		Код программы обработки прерывания	
	:		
POP	CLRF	ALUSTA, F	; FSR0 - режим автодекрементирования
используемого	MOVFP	Bank_FSR, FSR0	; восстановление значения FSR,
			; для сохранения других значений
	DECf	FSR0, F	
	MOVFP	INDF0, PCLATH	; чтение значения PCLATH
	MOVFP	INDF0, WREG	; чтение значение WREG
регистра	BSF	ALUSTA, FS1	; FSR0 - режим без изменения значения
	MOVFP	INDF0, ALU_Temp	; чтение значения ALUSTA
	MOVFP	FSR0, Bank_FSR	; сохранение значения FSR, используемого
			; для сохранения других значений
	DECf	Nobank_FSR, F	

MOVFP	Nobank_FSR, FSR0	; восстановление значения FSR, используемого
		; для сохранения BSR
MOVFP	ALU_Temp, ALUSTA	
MOVFP	INDF0, BSR	
		;
RETFIE		; возврат из прерывания (разрешает прерывания)

### Организация памяти

В микроконтроллере есть два блока памяти: память программ и память данных. Каждый блок имеет свою собственную шину, так что доступ к каждому блоку возможен во время одного и того же цикла генератора. Память данных делится на RAM общего назначения и регистры специальных функций (SFRs). Функционирование SFR-регистров, которые управляют ядром микроконтроллера, описываются в этой главе. SFR-регистры, используемые для управления модулями периферии, описываются в разделах, посвященных этим конкретным модулям периферии.

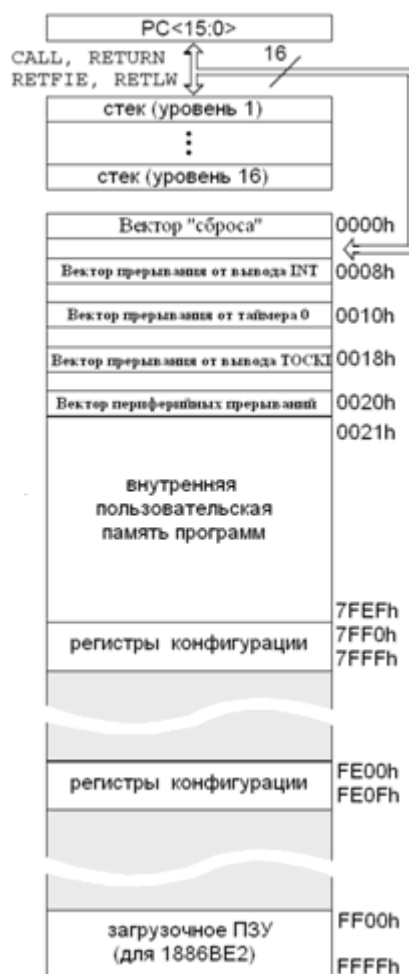


Рис. 12 Карта памяти программ и стека

### Память программ

Микроконтроллеры имеют 16-ти битный счетчик команд, способный адресовать область памяти программ размером 64К x 16 бит. Вектор «сброса» имеет адрес 0000h, вектора прерывания находятся по адресам 0008h, 0010h, 0018h и 0020h (см. Рис. 12).

Объем внутренней памяти программ составляет 32К x 16 бит. Память перепрограммируемая FLASH типа.

Микроконтроллер может функционировать в одной из 4 возможных конфигураций памяти программ. Конфигурация выбирается битами конфигурации. Возможны следующие режимы: **микропроцессор, микроконтроллер, расширенный микроконтроллер и защищенный микроконтроллер.**

Режимы «микроконтроллера» и «защищенного микроконтроллера» обеспечивают доступ только к внутренней памяти программ. Любая попытка доступа по адресам вне памяти программ приводит к чтению не известного значения. Режим защищенного микроконтроллера еще имеет функцию защиты кода.

В режиме «расширенного микроконтроллера» имеется доступ как к внутренней, так и внешней памяти программ. Выполнение автоматически переключается между внутренней и внешней памятью. 16-ти битный адрес позволяет адресовать диапазон памяти программ в 64К слов.

В режиме «микропроцессор» возможен доступ только к внешней памяти программ. Встроенная (внутренняя) память программ игнорируется. 16-ти битный адрес позволяет адресовать диапазон памяти программ в 64К слов. Микропроцессорный режим исходно установлен в незапрограммированных приборах.

Микроконтроллеры могут функционировать, когда память программ находится вне кристалла. Это режимы «микропроцессора» и «расширенного микроконтроллера». Микропроцессорный режим исходно установлен в незапрограммированных приборах.

Независимо от режима процессора память данных всегда находится на кристалле.

Когда выбран режим «микропроцессора» или «расширенного микроконтроллера», PORTC, PORTD и PORTE работают как системная шина. PORTC и PORTD – мультиплексированная шина адреса/данных, а PORTE<2:0> – сигналы управления. Форма сигналов адреса и данных показана на Рис. 14. Временные характеристики приведены в электрической спецификации. В режиме «расширенного микроконтроллера» когда прибор работает с внутренней памятью программ, внешние управляющие сигналы продолжают быть активными. Таким образом, они индицируют действия, которые происходят во внутренней памяти. Данные из внешней памяти игнорируются.

Режим микропроцессора	Режим расширенного микроконтроллера	Режим микроконтроллера	
0000h Внешняя память программ FFFFh вне кристалла	0000h внутренняя память программ 7FFFh Внешняя память программ 8000h FFFFh вне кристалла	0000h внутренняя память программ 7FFFh 8000h Config. Bits FFFFh вне кристалла	память программ
00h на кристалле 120h 220h 320h FFh 1FFh 2FFh 3FFh	00h на кристалле 120h 220h 320h FFh 1FFh 2FFh 3FFh	00h на кристалле 120h 220h 320h FFh 1FFh 2FFh 3FFh	память данных

**Рис. 13** Распределение памяти в различных режимах



**Рис. 14** Диаграммы сигналов доступа к внешней памяти программ

### Память данных

Память данных разделена на 2 области. Первая – это область регистров общего назначения (GPR), и вторая – это область регистров специальных функций (SFR). SFR-регистры контролируют и обеспечивают режимы функционирования прибора.

Память данных разделена на банки, так организованы обе области. Область GPR сгруппирована в банки для того, чтобы получить объем памяти более 232 байта. Организация банками требует использования управляющих битов для выбора банка. Эти управляющие биты находятся в регистре выбора банков BSR. Если произведен доступ к области вне банков, значение регистра BSR игнорируется. Таблица 12 показывает организацию карты памяти данных.

Команды MOVPF и MOVFP обеспечивают перенос значений данных из области периферии («P») в любое место в области регистров («F») и наоборот. «P»-диапазон определен адресами от 00h до 1Fh, диапазон «F» – от 00h до FFh. Диапазон «P» имеет 6 регистров, которые могут быть использованы как регистры общего назначения. Это может быть удобно для некоторых применений, где переменные необходимо скопировать в другие ячейки в ОЗУ общего назначения, (такие как запоминание информации о статусе во время прерывания).

Ко всей памяти данных можно обращаться используя либо прямой доступ, либо косвенный (используя регистры указателя адреса FSR0 и FSR1). Косвенная адресация использует соответствующие управляющие биты BSR-регистра для доступа в области памяти данных, организованные в банки. BSR-регистр подробно рассматривается в разделе «Регистр выбора банка (BSR)».

### Регистры общего назначения (GPR)

Микроконтроллеры имеют область регистров общего назначения (ОЗУ) объемом 902 байта. Эти регистры 8-ми битные. ОЗУ разбито на банки. Для облегчения переключения между этими банками существует команда «MOVLR bank». Регистр GPR не изменяется при всех типах сбросов.

### Регистры специального назначения (SFR)

Регистры специального назначения (SFR) используются процессором и периферийными устройствами для управления работой прибора (см. Таблица 12). Эти регистры представляют собой статическое ОЗУ.

Регистры SFR могут быть разделены на 2 группы, те, которые связаны с функцией «ядра», и те, которые связаны с функциями периферии. Те регистры, которые связаны с «ядром», описываются ниже, а те, которые связаны с функциями периферии, описываются в соответствующем разделе для каждого модуля периферии. Регистры периферии организованы в банки, регистры «ядра» представляют собой область, не организованную в банки. Для облегчения переключения между периферийными банками используется команда «MOVLB bank».

**Таблица 12**  
Схема адресов регистров

Адрес	Не зависит от номера адресуемого банка.							
00h	INDF0							
01h	FSR0							
02h	PCL							
03h	PCLATH							
04h	ALUSTA							
05h	TOSTA							
06h	CPUSTA							
07h	INTSTA							
08h	INDF1							
09h	FSR1							
0Ah	WREG							
0Bh	TMR0L							
0Ch	TMR0H							
0Dh	TBLPTRL							
0Eh	TBLPTRH							
0Fh	BSR							
	Банк 0	Банк 1	Банк 2	Банк 3	Банк 4	Банк 5	Банк 6	Банк 7



## Спецификация 1886BE2У, К1886BE2У

10h	PORTA	DDRC	TMR1	PW1DC L	PIR2	DDRF	SSPAD D	PW3DC L
11h	DDRB	PORTC	TMR2	PW2DC L	PIE2	PORTF	SSPCO N1	PW3DC H
12h	PORTB	DDRD	TMR3L	PW1DC H	-	DDRG	SSPCO N2	CA3L
13h	RCSTA 1	PORTD	TMR3H	PW2DC H	RCSTA 2	PORTG	SSPST AT	CA3H
14h	RCREG 1	DDRE	PR1	CA2L	RCREG 2	ADCON 0	SSPBU F	CA4L
15h	TXSTA 1	PORTE	PR2	CA2H	TXSTA 2	ADCON 1	-	CA4H
16h	TXREG 1	PIR1	PR3L/C A1L	TCON1	TXREG 2	ADRES L	-	TCON3
17h	SPBRG 1	PIE1	PR3H/ CA1H	TCON2	SPBRG 2	ADRES H	-	WRDIV
<b>Не зависит от номера адресуемого банка.</b>								
18h	PRODL							
19h	PRODH							
1Ah	<b>Регистры общего назначения.</b>							
1Fh	<b>Регистры общего назначения.</b>							
	<b>Банк 0</b>		<b>Банк 1</b>		<b>Банк 2</b>		<b>Банк 3</b>	
20h	Регистры общего назначения.		Регистры общего назначения.		Регистры общего назначения.		Регистры общего назначения.	
FFh	Регистры общего назначения.		Регистры общего назначения.		Регистры общего назначения.		Регистры общего назначения.	

**Примечание:**

- Регистры SFR в области адресов 10h-17h разбиты на банки. Младший полубайт регистра BSR определяет выбранный номер банка. Все не организованные в банки регистры игнорируют значения битов регистра BSR.
- Область памяти GPR с адресами 20h-FFh, 120h-1FFh, 220h-2FFh и 320h-3FFh разбита на банки. Старший полубайт регистра BSR определяет выбранный номер банка. Другие регистры памяти игнорируют значения битов регистра BSR.
- Чтение любого не существующего регистра дает значение равное нулю.

**Таблица 13**  
Регистры специального назначения

Адрес	Наз-вание	бит 7	бит 6	бит 5	бит 4	бит 3	бит 2	бит 1	бит 0	POR, BOR	MCLR, WDT
<b>Не зависит от номера адресуемого банка.</b>											
00h	INDF0	использует содержимое FSR0 для адресации памяти данных (физически не реализован)								----	----
01h	FSR0	указатель 0 адреса, для косвенной адресации памяти данных								0000 0000	uuuu uuuu
02h	PCL	младшие 8 бит счетчика команд								0000 0000	0000 0000
03h	PCLA	регистр-защелка для старших 8 бит счетчика команд								0000	0000

## Спецификация 1886BE2У, К1886BE2У

Адрес	Название	бит 7	бит 6	бит 5	бит 4	бит 3	бит 2	бит 1	бит 0	POR, BOR	MCLR, WDT
	TH									0000	0000
04h	ALUSTA	FS3	FS2	FS1	FS0	OV	Z	DC	C	1111 0000	1111 0000
05h	T0STA	INTEDG	T0SE	T0CS	T0PS3	T0PS2	T0PS1	T0PS0	-	0000 000-	0000 000-
06h	CPUSTA	-	-	STKAV	GLINTD	TO	PD	POR	BOR	--11 11q0	--11 qq1u
07h	INTSTA	PEIF	TOCKIF	TOIF	INTF	PEIE	TOCKIE	TOIE	INTE	0000 0000	0000 0000
08h	INDF1	использует содержимое FSR1 для адресации памяти данных (физически не реализован)								----	----
09h	FSR1	указатель 1 адреса, для косвенной адресации памяти данных								0000 0000	uuuu uuuu
0Ah	WREG	рабочий регистр								0000 0000	0000 0000
0Bh	TMR0L	младший байт регистра таймера 0								0000 0000	0000 0000
0Ch	TMR0H	старший байт регистра таймера 0								0000 0000	0000 0000
0Dh	TBLPTL	младший байт табличного указателя памяти программ								0000 0000	0000 0000
0Eh	TBLPTRH	старший байт табличного указателя памяти программ								0000 0000	0000 0000
0Fh	BSR	регистр выбора банка памяти данных								0000 0000	0000 0000
<b>Банк 0</b>											
10h	PORTA(3,5)	RBPU	-	PA5/ TX1/ CK1	PA4/ RX1/ DT1	PA3/ SDI/ SDA	PA2/ S/ SCL	PA1/ T0CLK	PA0/ INT	0-xx 11xx	0-uu 11uu
11h	DDRB	регистр выбора направления данных для PORTB								1111 1111	1111 1111
12h	PORTB(3)	PB7/ SDO	PB6/ SCK	PB5/ TCLK3	PB4/ TCLK2	PB3/ PWM2	PB2/ PWM1	PB1/ CAP2	PB0/ CAP1	xxxx xxxx	uuuu uuuu
13h	RCSTA1	SPEN1	RX91	SREN1	CREN1	-	FERR1	OERR1	RX9D1	0000 - 000	0000 - 000
14h	RCREG1	регистр принимаемых данных последовательного порта USART1								0000 0000	0000 0000
15h	TXSTA1	CSRC1	TX91	TXEN1	SYNC1	-	-	TRMT1	TX9D1	0000 -- 10	0000 -- 10
16h	TXREG1	регистр передаваемых данных последовательного порта USART1								0000 0000	0000 0000
17h	SPBRG1	регистр управления генератором последовательного порта USART1								0000 0000	0000 0000
<b>Банк 1</b>											
10h	DDRC(4)	регистр выбора направления данных для PORTC								1111 1111	1111 1111

## Спецификация 1886BE2У, К1886BE2У

Адрес	Название	бит 7	бит 6	бит 5	бит 4	бит 3	бит 2	бит 1	бит 0	POR, BOR	MCLR, WDT
11h	PORTC(3,4)	PC7/AD7	PC6/AD6	PC5/AD5	PC4/AD4	PC3/AD3	PC2/AD2	PC1/AD1	PC0/AD0	xxxx xxxx	uuuu uuuu
12h	DDRD(4)	регистр выбора направления данных для PORTD								1111 1111	1111 1111
13h	PORTD(3,4)	PD7/AD15	PD6/AD14	PD5/AD13	PD4/AD12	PD3/AD11	PD2/AD10	PD1/AD9	PD0/AD8	xxxx xxxx	uuuu uuuu
14h	DDRE(4)	регистр выбора направления данных для PORTE								---- 1111	---- 1111
15h	PORTE(3,4)	-	-	-	-	PE3/CAP4	PE2/WR	PE1/OE	PE0/ALE	---- xxxx	---- uuuu
16h	PIR1	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TX1IF	RC1IF	x000 0010	u000 0010
17h	PIE1	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TX1IE	RC1IE	0000 0000	0000 0000
<b>Банк 2</b>											
10h	TMR1	регистр таймера 1								0000 0000	0000 0000
11h	TMR2	регистр таймера 2								0000 0000	0000 0000
12h	TMR3L	младший байт регистра таймера 3								0000 0000	0000 0000
13h	TMR3H	старший байт регистра таймера 3								0000 0000	0000 0000
14h	PR1	регистр периода таймера 1								1111 1111	1111 1111
15h	PR2	регистр периода таймера 2								1111 1111	1111 1111
16h	PR3L/CA1L	младший байт регистра периода таймера 3 / младший байт регистра захвата 1								0000 0000	0000 0000
17h	PR3H/CA1H	старший байт регистра периода таймера 3 / старший байт регистра захвата 1								0000 0000	0000 0000
<b>Банк 3</b>											
10h	PW1DCL	DC1	DC0	-	-	-	-	-	-	00-- ----	00-- ----
11h	PW2DCL	DC1	DC0	TM2PW2	-	-	-	-	-	000- --	000- --
12h	PW1DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	0000 0000	0000 0000
13h	PW2DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	0000 0000	0000 0000
14h	CA2L	младший байт регистра захвата 2								0000 0000	0000 0000
15h	CA2H	старший байт регистра захвата 2								0000 0000	0000 0000
16h	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h	TCON	CA2O	CA1O	PWM2	PWM1	CA1/P	TMR3	TMR2	TMR1	0000	0000

## Спецификация 1886BE2У, К1886BE2У

Адрес	Название	бит 7	бит 6	бит 5	бит 4	бит 3	бит 2	бит 1	бит 0	POR, BOR	MCLR, WDT
	2	VF	VF	ON	ON	R3	ON	ON	ON	0000	0000
<b>Банк 4</b>											
10h	PIR2	SSPIF	BCLIF	ADIF	-	CA4IF	CA3IF	TX2IF	RC2IF	000-0010	000-0010
11h	PIE2	SSPIE	BCLIE	ADIE	-	CA4IE	CA3IE	TX2IE	RC2IE	000-0000	000-0000
12h	нет	-	-	-	-	-	-	-	-	-----	-----
13h	RCSTA2	SPEN2	RX92	SREN2	CREN2	-	FERR2	OERR2	RX9D2	0000-000	0000-000
14h	RCREG2	регистр принимаемых данных последовательного порта USART2								0000 0000	0000 0000
15h	TXSTA2	CSRC2	TX92	TXEN2	SYNC2	-	-	TRMT2	TX9D2	0000-010	0000-010
16h	TXREG2	регистр передаваемых данных последовательного порта USART2								0000 0000	0000 0000
17h	SPBRG2	регистр управления генератором последовательного порта USART2								0000 0000	0000 0000
<b>Банк 5</b>											
10h	DDRF	регистр выбора направления данных для PORTF								1111 1111	1111 1111
11h	PORTF(3)	PF7/AN11	PF6/AN10	PF5/AN9	PF4/AN8	PF3/AN7	PF2/AN6	PF1/AN5	PF0/AN4	0000 0000	0000 0000
12h	DDRG	регистр выбора направления данных для PORTG								1111 1111	1111 1111
13h	PORTG(3)	PG7/TX2/CK2	PG6/RX2/DT2	PG5/PWM3	PG4/CAP3	PG3/AN0	PG2/AN1	PG1/AN2	PG0/AN3	0000 0000	0000 0000
14h	ADCON0	CHS3	CHS2	CHS1	CHS0	-	GO/DONE	-	ADON	0000-0-0	0000-0-0
15h	ADCON1	ADCS1	ADCS0	ADFM	-	PCFG3	PCFG2	PCFG1	PCFG0	000-0000	000-0000
16h	ADRESL	младший байт регистра результата преобразования АЦП								xxxx xxxx	uuuu uuuu
17h	ADRESH	старший байт регистра результата преобразования АЦП								xxxx xxxx	uuuu uuuu
<b>Банк 6</b>											
10h	SSPADD	регистр адреса в режиме ведомого I2C, регистр скорости обмена в режиме ведущего I2C								0000 0000	0000 0000
11h	SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
12h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000
13h	SSPSDAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	0000 0000
14h	SSPBUF	регистр буфера приема/передачи синхронного последовательного порта								0000 0000	0000 0000
15h	нет	-	-	-	-	-	-	-	-	-----	-----

## Спецификация 1886BE2У, К1886BE2У

Адрес	Название	бит 7	бит 6	бит 5	бит 4	бит 3	бит 2	бит 1	бит 0	POR, BOR	MCLR, WDT
16h	нет	-	-	-	-	-	-	-	-	-----	-----
17h	нет	-	-	-	-	-	-	-	-	-----	-----
<b>Банк 7</b>											
10h	PW3D CL	DC1	DC0	TM2P W3	-	-	-	-	-	000- -- --	000- -- --
11h	PW3D CH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	0000 0000	0000 0000
12h	CA3L	младший байт регистра захвата 3								0000 0000	0000 0000
13h	CA3H	старший байт регистра захвата 3								0000 0000	0000 0000
14h	CA4L	младший байт регистра захвата 4								0000 0000	0000 0000
15h	CA4H	старший байт регистра захвата 4								0000 0000	0000 0000
16h	TCON 3	-	CA4O VF	CA3O VF	CA4E D1	CA4E D0	CA3E D1	CA3E D0	PWM3 ON	-000 0000	-000 0000
17h	WRDI V	коэффициент деления частоты генератора для записи во флеш								0000 0101	0000 0101
<b>Не зависит от номера адресуемого банка.</b>											
18h	PROD L	младший байт 16-ти битного результата (8x8 битное аппаратное умножение)								0000 0000	0000 0000
19h	PROD H	старший байт 16-ти битного результата (8x8 битное аппаратное умножение)								0000 0000	0000 0000

### Обозначения:

x=не известно; u=не изменяется; -=не реализовано, читается «0»;q=зависит от условий.

### Примечания:

1. К старшему байту счетчика команд нет прямого доступа. PCLATH - это регистр-защелка для PC<15:8>. При выполнении некоторых команд значение регистра PCLATH переписывается в PC<15:8>, или наоборот: значение PC<15:8> переписывается в PCLATH.
2. Внешний сброс от вывода MCLR не влияет на статусные биты TO и PD регистра CPUSTA.
3. Это то значение, которое будет на выходной защелке порта.
4. Когда прибор сконфигурирован для «Микропроцессорного» и «Расширенного микроконтроллерного» режимов, функционирование этого порта не связано с этими регистрами.
5. При любом сбросе прибора эти выводы конфигурируются как входы.

**Регистр статуса процессора (ALUSTA)** - регистр содержит биты статуса арифметического и логического блоков и биты управления режимом для режима косвенной адресации.

Как в случае со всеми другими регистрами, в регистр ALUSTA может быть загружен результат выполнения любой команды. Если регистр ALUSTA является местом назначения для результата определенной команды, которая может изменять биты Z, DC, C и OV, то запись в эти 4 бита запрещается. Эти биты

устанавливаются или сбрасываются в соответствии с результатом выполнения команды. Следовательно, результат выполнения команды с регистром ALUSTA в качестве места назначения результата может стать отличным от того, что надеялись получить. Следовательно, рекомендуется для изменения ALUSTA-регистра использовать только следующие команды: BCF, BSF, SWAPF и MOVWF, т.к. эти команды не влияют на какие-либо статусные биты. Чтобы посмотреть, как другие команды влияют на статусные биты, смотрите описание системы команд.

Арифметический и логический блок (АЛУ) может производить арифметические и логические операции над двумя операндами или с одним операндом. Все команды с одним операндом производятся либо с WREG-регистром либо с данным файловым регистром. В командах с двумя операндами один операнд – это WREG-регистр, другой – либо файловый регистр, либо 8-ми битная константа.

**Таблица 14**

Регистр ALUSTA (адрес: 04h, не зависит от номера банка)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
FS3	FS2	FS1	FS0	OV	Z	DC	C
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7,6</b>		<b>FS3:FS2:</b> биты выбора режима FSR1 00 = автодекремент величины FSR1 после выполнения команды 01 = автоинкремент величины FSR1 после выполнения команды 1x = значение FSR1 не изменяется					
<b>бит 5,4</b>		<b>FS1:FS0:</b> биты выбора режима FSR0 00 = автодекремент величины FSR0 после выполнения команды 01 = автоинкремент величины FSR0 после выполнения команды 1x = значение FSR0 не изменяется					
<b>бит 3</b>		<b>OV:</b> бит переполнения Этот бит используется для знаковой арифметики (дополнение до 2). Он показывает переполнение, когда знаковый бит (7 бит) изменяет состояние. 1 = произошло переполнение для знаковой арифметики в арифметических операциях (т.е. результат для знаковой арифметики превысил +127, или стал меньше чем -128) 0 = не произошло переполнение					
<b>бит 2</b>		<b>Z:</b> флаг нуля 1 = результат арифметической или логической операции равен 0 0 = результат арифметической или логической операции не равен 0					
<b>бит 1</b>		<b>DC:</b> флаг десятичного переноса/заема Для команд ADDWF и ADDLW. 1 = произошел перенос из 4-го снизу бита результата 0 = не было переноса из 4-го снизу бита результата Примечание: для заема значение инверсное (для команд вычитания).					

<b>бит 0</b>	<b>С:</b> флаг переноса/заема Для команд ADDWF и ADDLW. Отметим, что вычитание выполняется дополнением до 2 второго операнда. Для команд сдвига RRCF и RLCF этот бит загружается либо старшим, либо младшим битом регистра-источника. 1 = произошел перенос из самого значащего бита результата 0 = нет переноса из самого значащего бита результата. Примечание: для заема значение инверсное (для команд вычитания).
--------------	---

Обозначения:

R = бит для чтения; W = бит с возможностью записи; U = бит не реализован, читается как 0;

-n = значение бита после сброса по включению питания: 1 = установлен; 0 = сброшен; x = значение не известно.

**Регистр статуса ЦПУ (CPUTA)** содержит статусный и управляющий биты для ЦПУ. Этот регистр содержит бит, который используется для глобального разрешения/запрещения прерываний. Если необходимо разрешить/запретить только определенное прерывание, используются регистры статуса прерываний (INTSTA) или регистры разрешения прерываний от периферии (PIE). Регистр CPUTA также показывает, доступность стека, и содержит флаги включения питания (PD) и переполнения сторожевого таймера (TO). Биты TO, PD и STKAV доступны только для чтения. Они устанавливаются и сбрасываются в соответствии с логикой прибора. Следовательно, результат выполнения команды с регистром CPUTA в качестве места назначения результата выполнения может быть отличным, нежели ожидалось.

Бит POR позволяет отличить сброс при включении питания от внешнего MCLR сброса или сброса по переполнению сторожевого таймера. Бит BOR является индикатором сброса по снижению напряжения питания (только в случае если схема сброса по снижению напряжения питания включена в регистре конфигурации).

**Таблица 15**

Регистр CPUSTA (адрес: 06h, не зависит от номера банка)

U-0	U-0	R-1	R/W-1	R-1	R-1	R/W-0	R/W-0
-	-	<b>STKAV</b>	<b>GLINTD</b>	<b>TO</b>	<b>PD</b>	<b>POR</b>	<b>BOR</b>
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7,6</b>		Не реализованы: читаются значения равные нулю.					
<b>бит 5</b>		<p><b>STKAV:</b> флаг доступа к стеку  Этот флаг показывает, что величина 4-х битного указателя стека равна Fh, или произошел переход от Fh к 0h, т.е. переполнение стека.  1 = стек доступен  0 = стек полон, или произошло переполнение стека (с тех пор, как этот бит был сброшен при переполнении стека, только «сброс» (RESET) прибора может установить этот бит)</p>					
<b>бит 4</b>		<p><b>GLINTD:</b> бит глобального запрета прерываний  Этот бит запрещает все прерывания. Когда прерывания разрешены, то вызвать прерывания могут только источники с установленными битами разрешения прерывания.  1 = запрещены все прерывания  0 = разрешены все немаскированные прерывания</p>					
<b>бит 3</b>		<p><b>TO:</b> флаг переполнения сторожевого таймера  1 = устанавливается после включения питания или выполнения команд CLRWDT или SLEEP  0 = после переполнения сторожевого таймера</p>					
<b>бит 2</b>		<p><b>PD:</b> флаг включения питания  1 = устанавливается после включения питания или выполнения команды CLRWDT  0 = после выполнения команды SLEEP</p>					
<b>бит 1</b>		<p><b>POR:</b> флаг сброса при включении питания  1 = не было сброса при включении питания  0 = произошел сброс при включении питания (бит должен быть установлен программно)</p>					
<b>бит 0</b>		<p><b>BOR:</b> флаг сброса по снижению напряжения питания  <i>Когда бит BODEN в регистре конфигурации установлен (разрешено):</i>  1 = сброса при снижении питания не было  0 = произошел сброс при снижении питания (бит должен быть установлен программно)  <i>Когда бит BODEN в регистре конфигурации сброшен (запрещено):</i>  значение бита безразлично.</p>					

Обозначения:

R = бит для чтения; W = бит с возможностью записи; U = бит не реализован, читается как 0;

-n = значение бита после сброса по включению питания: 1 = установлен; 0 = сброшен; x = значение не известно.



**Регистр статуса управления TMR0 (T0STA).** Этот регистр содержит различные биты управления. Бит 7 (INTEDG) используется для выбора управляющего перепада сигнала (т.е. фронт или срез сигнала), при котором на выводе PA0/INT будет устанавливаться флаг запроса прерывания PA0/INT. Остальные биты конфигурируют таймер 0, его предделитель и источник тактовых сигналов.

**Таблица 16**

Регистр T0STA (адрес: 05h, не зависит от номера банка)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
INTEDG	TOSE	TOCS	TOPS3	TOPS2	TOPS1	TOPS0	-
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7</b>		<b>INTEDG:</b> бит выбора управляющего перепада сигнала на выводе PA0/INT для прерывания. Этот бит выбирает, на фронте или спаде сигнала будет происходить прерывание. 1 = фронт сигнала на выводе PA0/INT генерирует прерывание 0 = спад сигнала на выводе PA0/INT генерирует прерывание					
<b>бит 6</b>		<b>TOSE:</b> бит выбора управляющего перепада сигнала при внешнем тактировании таймера 0. Этот бит выбирает, на фронте или спаде сигнала таймер 0 будет инкрементироваться. <u>Когда TOCS=0 (внешнее тактирование):</u> 1 = фронт сигнала на выводе PA1/T0CLK инкрементирует таймер 0 и/или устанавливает T0CKIF - бит 0 = спад сигнала на выводе PA1/T0CLK инкрементирует таймер 0 и/или устанавливает T0CKIF - бит <u>Когда TOCS=1 (внутреннее тактирование):</u> значение бита безразлично					
<b>бит 5</b>		<b>TOCS:</b> бит выбора источника тактирования для таймера 0. Этот бит выбирает источник синхронизации для таймера 0. 1 = внутренняя тактовая частота с генератора циклов (Tcy) 0 = внешнее тактирование с вывода PA1/T0CLK					
<b>бит 4-1</b>		<b>TOPS3:TOPS0:</b> биты выбора предделителя для таймера 0. Эти биты позволяют выбрать величину деления предделителя: 0000 - 1:1    0001 - 1:2    0010 - 1:4    0011 - 1:8 0100 - 1:16    0101 - 1:32    0110 - 1:64    0111 - 1:128 1xxx - 1:256					
<b>бит 0</b>		Не реализовано: читается значение равное нулю.					

### Функционирование стека

Микроконтроллеры имеют 16x16 бит аппаратный стек (см. Рис. 12). Стек не является частью области памяти программ или данных, указатель стека не является ни считываемым, ни записываемым. Значение счетчика команд PC помещается (PUSH) в стек, когда выполняются команды CALL и LCALL или произошло прерывание. Стек восстанавливает значение PC (POP) в случае

выполнения команд RETURN, RETLW или RETFIE. Операции «PUSH» и «POP» не влияют на PCLATH (защелку).

Стек работает как круговой буфер с указателем стека, сброшенным в нулевое значение после любых типов сбросов. В стеке существует определенный бит (STKAV), позволяющий программно убедиться в том, что не произошло переполнения стека. Бит STKAV устанавливается после сброса прибора. Когда указатель стека становится равен Fh, STKAV сбрасывается. Если указатель стека проходит адреса от Fh к 0h, бит STKAV будет оставаться сброшенным до тех пор, пока не произойдет сброс прибора.

### Примечание.

1. Не предусмотрен специальный статусный бит для заполненного стека. STKAV-бит может быть использован для обнаружения того, что стек заполнен, в результате чего указатель стека находится на его вершине.
2. Здесь нет командной мнемоники, называемой «PUSH» или «POP». Это действия, которые происходят при выполнении команд CALL, RETURN, RETLW и RETFIE или обращении к вектору прерывания.
3. После сброса если операция «POP» имеет место до операции «PUSH», бит STKAV будет сброшен. Это выглядит так же, как в случае когда стек полон. Если следующей выполняется операция «PUSH» (перед следующим «POP»), то бит STKAV зафиксируется сброшенным. И только сброс прибора устанавливает этот бит.

После того, как прибор 16 раз осуществил операцию «PUSH» (без операции «POP»), 17-й «push» записывает значение поверх первого. 18-й «push» записывает сверху второго «push» (и т.д.).

## Косвенная адресация

Косвенная адресация – это режим адресации памяти данных, когда адрес памяти данных в команде не фиксирован. Таким образом, адрес регистра, из которого будет производиться чтение или в который будет производиться запись, может быть модифицирован программой. Это может быть удобно в случае таблиц данных, размещенных в памяти данных. Ниже показан принцип косвенной адресации (см. Рис. 15). Там показана модификация значения адреса памяти данных, значением регистра.

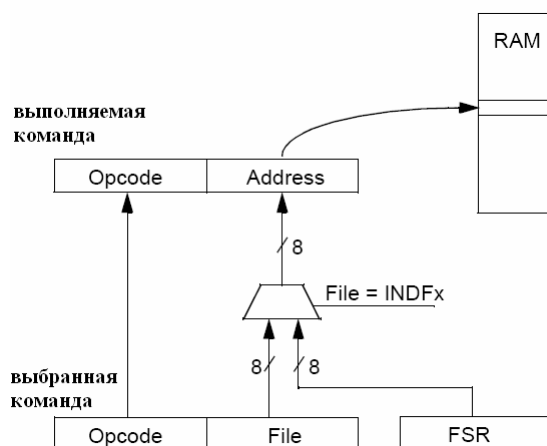


Рис. 15 Косвенная адресация

Пример 3 показывает использование косвенной адресации для очистки ОЗУ данных (от 20h до FFh) с минимальным количеством команд. Подобная концепция может быть использована для переноса определенного числа байт данных в передающий регистр USART (TXREG).

Микроконтроллер имеет две пары регистров для реализации косвенной адресации. Это: INDF0, FSR0 и INDF1, FSR1. Регистры INDF0 и INDF1 физически не реализованы. Чтение и запись в эти регистры активирует косвенную адресацию со значением адреса из соответствующего регистра FSR, который является адресом данных. FSR - это 8-ми битный регистр, позволяющий адресовать область памяти данных объемом 256 байт. Для памяти, организованной в банки, банк, к которому осуществляется доступ, определяется величиной в регистре BSR.

Если косвенно через FSR читается сам файл INDF0 или INDF1, то читаются все нули. Подобным образом, если в INDF0 (или INDF1) идет косвенная запись, операция будет эквивалентна команде NOP, и она не оказывает влияния на статусные биты.

Существуют 2 управляющих бита, связанных с каждым регистром FSR. Эти 2 бита конфигурируют FSR-регистр, чтобы:

- производить автодекремент значения (адреса) в регистре FSR после доступа к памяти с косвенной адресацией;
- производить автоинкремент значения (адреса) в регистре FSR после доступа к памяти с косвенной адресацией;
- не изменять значение (адрес) в регистре FSR после доступа к памяти с косвенной адресацией.

Эти управляющие биты находятся в регистре ALUSTA. Регистр FSR1 управляется битами FS3 и FS2, а регистр FSR0 управляется битами FS1 и FS0. Когда используются автоинкрементный или автодекрементный режимы, то изменение регистра FSR не отражается на регистре ALUSTA. Например, если при косвенной адресации регистр FSR станет равен нулю, то бит Z устанавливаться не будет. Если регистр FSR содержит величину 00h, косвенное чтение будет давать значение 00h (бит Z установлен), в то время как косвенная запись будет эквивалентна команде NOP (это не влияет на статусные биты).

Косвенная адресация позволяет за один цикл передавать данные по всему адресному пространству памяти данных. Это возможно с использованием команд MOVVPF и MOVVFP, где либо «P» либо «F» задано как INDF0 или INDF1. Если источник или приемник при косвенной адресации – это память, организованная в банки, то ячейка доступа будет определяться значением в регистре BSR.

### Пример 3 Косвенная адресация

	MOVLW	0x20	
	MOVWF	FSR0	; FSR0 = 20h
	BCF	ALUSTA,FS1	; Задание режима
	BSF	ALUSTA,FS0	; автоинкрементирования FSR
	BCF	ALUSTA,C	; C = 0
	MOVLW	END_RAM + 1	
LP	CLRF	INDF0,F	; очистка ячейки памяти (FSR-указатель адреса)
	CPFSEQ	FSR0	; сравнение: FSR0 = END_RAM+1?
	GOTO	LP	; Нет, очистка продолжается
	:		; Да, вся память очищена.

## Регистры для чтения/записи таблиц

Регистры указателя таблиц TBLPTRL и TBLPTRH формируют 16-ти битное значение для адресации пространства памяти программ размером 64 Кслов. Указатель таблиц используется командами TABLWT и TABLRD. Эти команды позволяют осуществить передачу данных между областями данных и программ. Указатель таблиц служит в качестве 16-ти битного адреса слова внутри программной памяти. Регистр защелки таблиц – 16-ти разрядный регистр. Старший байт регистра TBLATH, младший байт TBLATL. Регистры не относятся ни к области памяти программ, ни к области памяти данных. Защелка таблиц используется для временной фиксации данных во время их передачи между памятью программ и памятью данных (см. описания команд TABLRD, TABLWT, TLRD и TLWT). Для более полного описания этих регистров и функционирования чтения/записи таблиц см. раздел « Считывание и запись таблиц данных».

## Счетчик команд

Счетчик команд PC - это 16-ти битный регистр. PCL - младший байт счетчика команд находится в области памяти данных. PCL можно читать и записывать точно так же как и любой другой регистр. PCH – это старший байт счетчика команд, и он не имеет прямой адресации. Т.к. PCH находится вне памяти программ и данных, то используется 8-ми битный регистр PCLATH в качестве удерживающей защелки для старшего байта счетчика команд. PCLATH находится в памяти данных. Пользователь может считывать или записывать PCH через PCLATH.

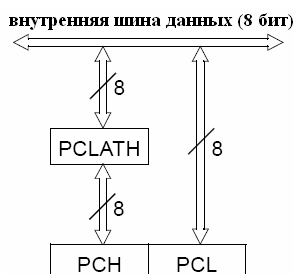


Рис. 16 Функционирование счётчика команд

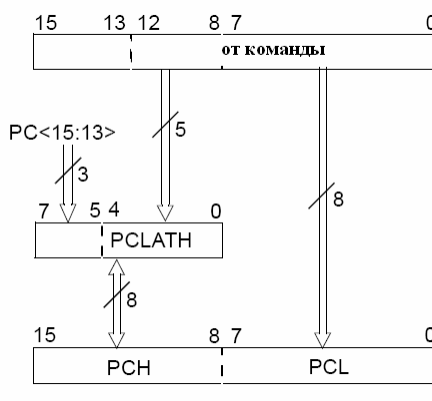


Рис. 17 Счетчик команд при выполнении инструкций CALL и GOTO

16-ти битный счетчик команд инкрементируется после выборки команды в течение цикла Q1 до тех пор пока:

- не изменится следующими командами: GOTO, CALL, LCALL, RETURN, RETLW или RETFIE;
- не изменится при переходе к вектору прерывания;
- не изменится в результате записи в регистр PCL результата выполнения команды.

Эти «переходы» эквивалентны вынужденному циклу «NOP» с переходом по адресу. Рис. 16 и Рис. 17 показывают функционирование счетчика команд в различных ситуациях.

Работа счетчика команд (PC) и регистра PCLATH для различных команд:

- **Команда LCALL:**

8-ми битный адрес указан в коде команды, PCLATH не изменяется.

PCLATH → PCH; биты команды <7: 0> → PCL

- **Любая команда чтения из PCL:**

PCL → шина данных → ALU или приемник; PCH → PCLATH

- **Любая команда записи в PCL:**

8-ми битные данные → шина данных → PCL; PCLATH → PCH

- **Любая команда чтения – модификации – записи PCL (например ADDWF PCL,F):**

Чтение: PCL → шина данных → ALU

Запись: 8-ми битный результат → шина данных → PCL

PCLATH → PCH

- **Команда RETURN:**

Содержимое стека → PC<15:0>

- **Команды CALL, GOTO:**

13-ти битный адрес указан в коде команды

биты команды <12:0> → PC<12:0>

PC<15:13> → PCLATH<7:5>

биты команды <12:8> → PCLATH<4:0>

Команды чтения – модификации – записи воздействуют только на PCL. PCH загружается значением из PCLATH. Для примера ADDWF PCL,F приведет к переходу в пределах текущей страницы. Если PC=03F0h, WREG=30h и PCLATH=03h до начала действия команды, то после ее действия PC=0320h. Чтобы выполнить правильный 16-ти байтный переход, необходимо вычислить 16-ти битный адрес приемника, записать старший байт в PCLATH и тогда записать младший в PCL.

Следующие команды, связанные со счетчиком команд, не изменяют PCLATH:

- LCALL, RETLW и RETFIE;
- переход к вектору прерывания;
- команды чтения – модификации - записи и записи для PCL.

### Регистр выбора банка (BSR)

Регистр выбора банка BSR используется для переключения между банками в области памяти данных (см. Рис. 18). Младший полубайт используется для

выбора банка периферийного регистра, для его записи используется команда «MOVLB bank». Старший полубайт используется для выбора банка памяти общего назначения (ОЗУ), для его записи используется команда «MOVLR bank». Если выбранный банк физически не реализован, то при его чтении будут считываться все нули. Любая запись в область памяти будет соответственно устанавливать или сбрасывать биты состояния АЛУ.

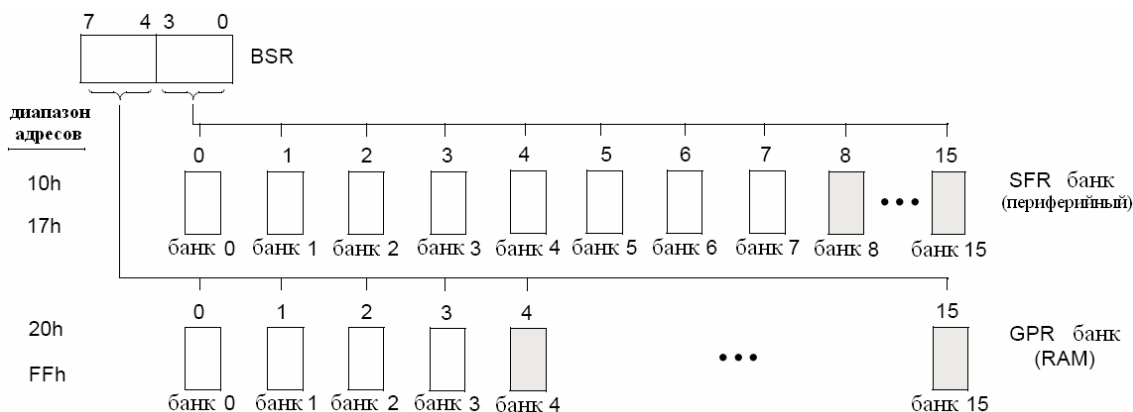
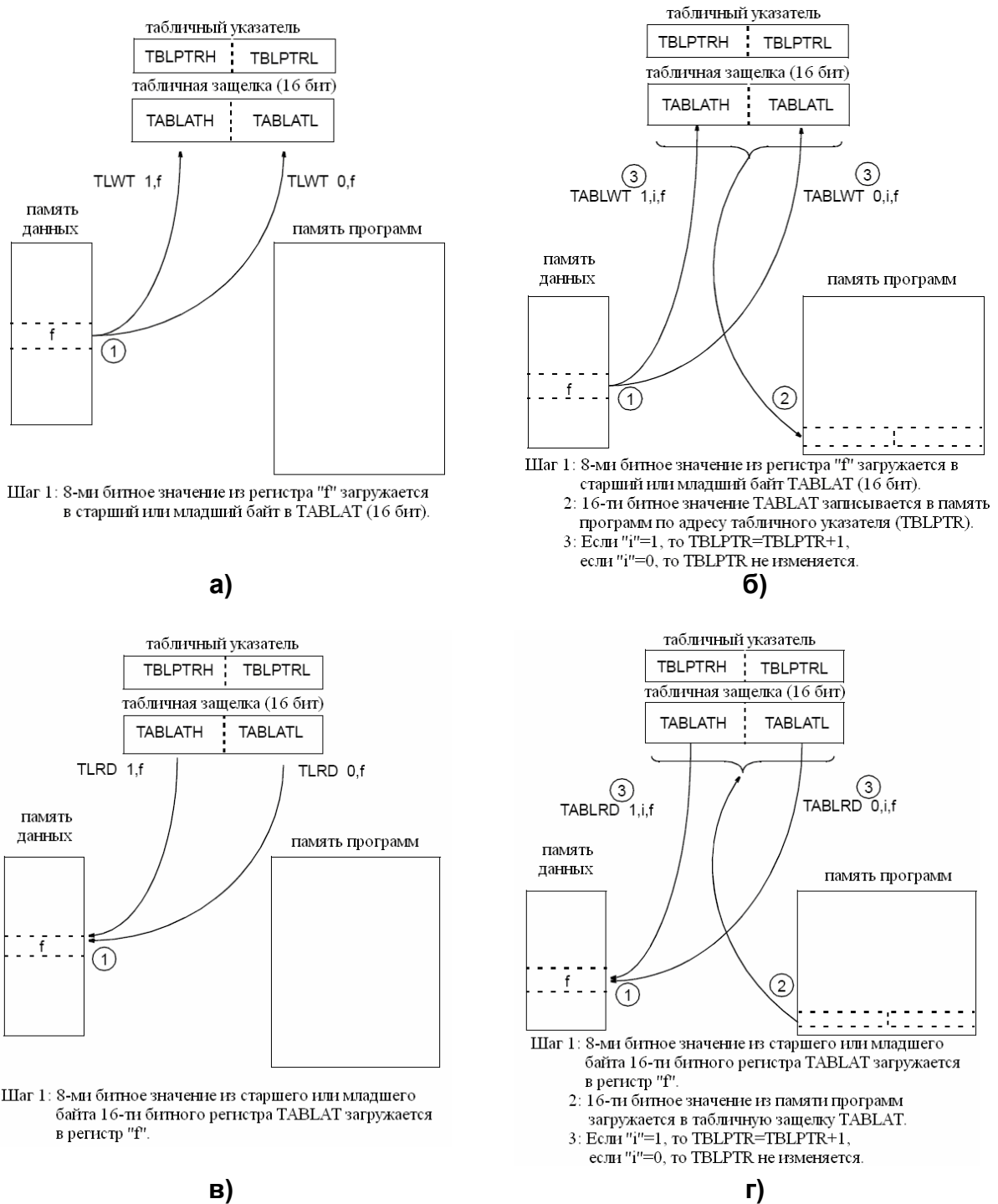


Рис. 18 Функционирование BSR

### Считывание и запись таблиц данных

Микроконтроллер имеет 4 команды, которые дают возможность переносить данные из области памяти данных в область памяти программ и наоборот. Т.к. память программ 16-ти битная, а память данных 8-ми битная, то для переноса 16-ти битной величины данных в память данных или из памяти данных требуется 2 операции. Для записи данных из памяти данных в память программ используются 2 следующие команды: TLWT t,f и TABLWT t,i,f. Для записи данных из памяти программ в память данных используются 2 следующие команды: TLRD t,f и TABLRD t,i,f. Операнд команды TABLWT - «i» определяет: требуется ли автоматически инкрементировать величину 16-ти битного регистра TBLPTR (для следующей записи). В Пример 4 регистр TBLPTR автоматически не инкрементируется. Рис. 19 показывает выполнение этих 4 команд. Шаги показывают последовательность операций. Память программ может быть как внутренней, так и внешней (для режима «микропроцессор» или «расширенный микроконтроллер»).



**Рис. 19** Выполнение команд чтения/записи таблиц

**а)** Выполнение команды TLWT  
**в)** Выполнение команды TLRD

**б)** Выполнение команды TABLWT  
**г)** Выполнение команды TABLRD

**Запись таблиц во внутреннюю память**

Записи таблиц во внутреннюю память запускает операцию «длинной записи». «Длинная запись» необходима для программирования внутренней FLASH памяти. Выполнение команд останавливается во время цикла «длинной

записи». «Длинная запись» будет закончена любым разрешенным прерыванием. Чтобы гарантировать, что ячейка памяти запрограммирована, требуется время программирования не менее: см. спецификацию. Для окончания «длинной записи» обычно разрешается только одно прерывание, чтобы гарантировать, что никакие другие прерывания преждевременно не закончат «длинную запись». Последовательность событий для программирования ячейки внутренней памяти программ будет следующей:

1. Запретить все источники прерываний, за исключением прерывания для окончания записи.
2. Подать на вывод MCLR/Upp напряжение программирования.
3. Сбросить сторожевой таймер (WDT).
4. Произвести запись таблицы. Прерывание закончит длинную запись.
5. Верифицировать ячейку памяти (чтение таблицы).

При программировании необходимо выполнять требования спецификации. Нарушение спецификации (включая температуру) может привести к тому, что ячейка памяти будет запрограммирована не полностью и со временем может стереться. Если напряжение программирования (Upp) не подано, то команда записи таблицы будет выполнена за 2 цикла, и память программ не изменится. Закончить операцию «длинной записи» могут только следующие события: источник прерывания или «сброс». Для окончания «длинной записи» по прерыванию, требуется чтобы было разрешено прерывание и был установлен флаг запроса прерывания.

Если для окончания «длинной записи» используется периферийное прерывание, то это прерывание должно быть разрешено и бит флага запроса прерывания должен быть установлен. Флаг запроса прерывания не сбрасывается при переходе по адресу вектора прерывания. Бит GLINTD определяет будет ли программа переходить к вектору прерывания после окончания «длинной записи». Если GLINTD сброшен, то программа переходит к вектору прерывания, если GLINTD установлен – не переходит. **Исключение составляет окончание «длинной записи» по прерыванию от «таймера 0».** В этом случае независимо от бита GLINTD программа продолжит выполнение, т.е. перехода к вектору прерывания не будет.

Длительность импульсов записи во флеш память задается 8-ми разрядным регистром **WRDIV** (коэффициент деления частоты генератора). Требуемое значение длительности 5 мкс. Коэффициент деления выбирается:  $K = 5 \text{ мкс} / TC$ , где TC в мкс. Значение регистра после сброса равно 05h. Если предполагается производить запись во флеш память и частота генератора отличается от 1 МГц, то необходимо предварительно загрузить в этот регистр необходимое значение. Регистр доступен по чтению и записи.



**Таблица 17**

Воздействие прерываний на операцию «длинной записи»

источник прерываний	GLINTD	бит разрешения	флаг запроса	действие
РА0/INT, РА1/Т0СLК, ТМR0 <sup>(прим.)</sup> , периферийные прерывания	0	1	1	Заканчивает длинную запись таблиц во внутреннюю память программ, переходит к вектору прерывания.
	0	1	0	Нет.
	1	0	х	Нет.
	1	1	1	Заканчивает длинную запись таблиц, не переходит к вектору прерывания.

Примечание:

В случае «таймера 0» переход к вектору прерывания не происходит.

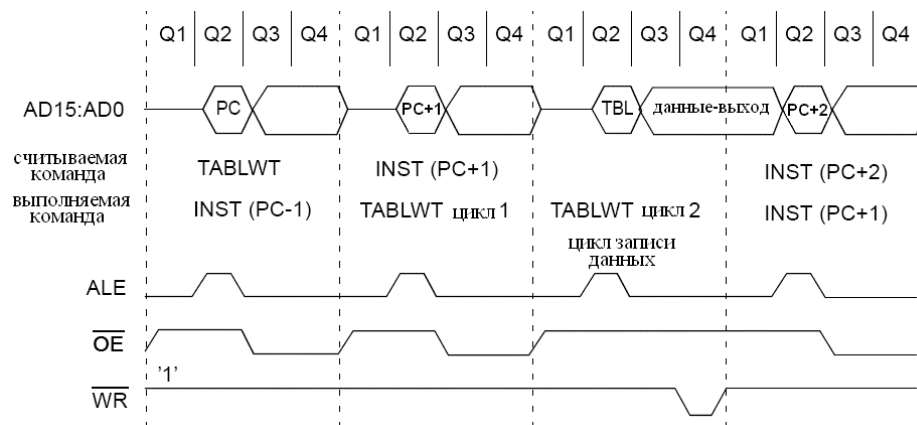
**Запись таблиц во внешнюю память**

Команда записи таблиц во внешнюю память всегда выполняется за 2 цикла. Второй цикл записывает данные в ячейку внешней памяти. Последовательность событий для записи во внешнюю память та же самая, что и для записи во внутреннюю.

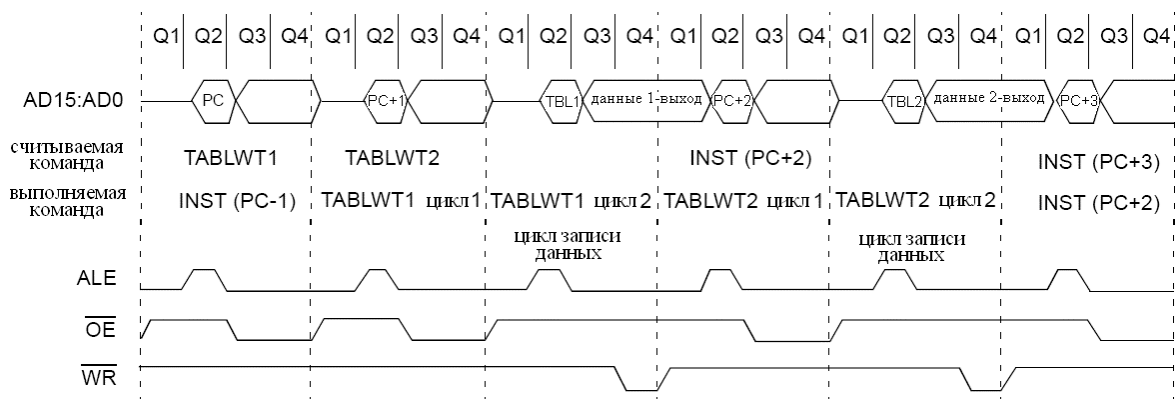
**Пример 4**

Запись таблицы

CLRWDТ		; Очистка сторожевого таймера
MOVLW	HIGH (TBL_ADDR)	; Загрузка адреса таблицы
MOVWF	TBLPTRH	
MOVLW	LOW (TBL_ADDR)	
MOVWF	TBLPTRL	
MOVLW	HIGH (DATA)	; Загрузка старшего байта в
TLWT	1,WREG	; табличную защелку TABLATH
MOVLW	LOW (DATA)	; Загрузка младшего байта в табличную
защелку		
TABLWT	0,0,WREG	; TABLATL и запись в память программ (внешнюю)



**Рис. 20** Временная диаграмма выполнения команды TABLWT



**Рис. 21** Временная диаграмма последовательности команд TABLWT

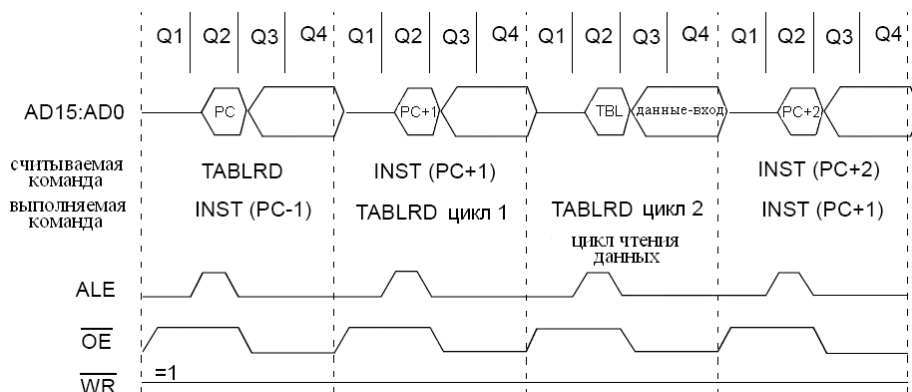
## Чтение таблиц

Операция чтения таблиц осуществляет чтение памяти программ. Это позволяет хранить константы в памяти программ и считывать их в память данных при необходимости. Пример 5 показывает считывание 16-ти битной величины из памяти программ с адресом из TBLPTR. После того, как незначащий байт был считан из TABLATH, в TABLATH загружается 16-ти битная величина из памяти программ с адресом из TBLPTR, и значение TBLPTR инкрементируется. При первом чтении данные из памяти программ загружаются в защелку, а данные считываемые из защелки рассматриваются как незначащее (пустое) чтение (в «f» были загружены неизвестные данные). Режим косвенной адресации через INDF0 должен быть сконфигурирован либо с автоинкрементированием либо с автодекрементированием регистра указателя адреса FSR0.

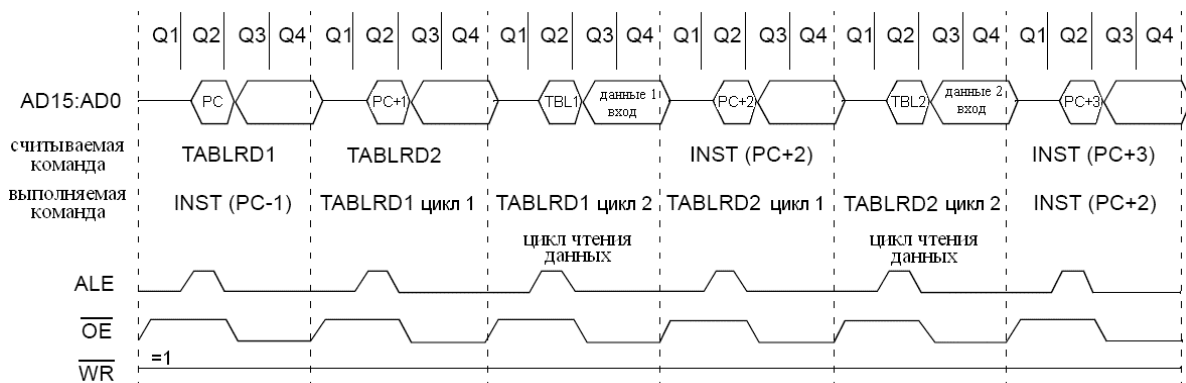
### Пример 5 Чтение таблицы

MOVLW	HIGH (TBL_ADDR); Загрузка адреса таблицы
MOVWF	TBLPTRH
MOVLW	LOW (TBL_ADDR)
MOVWF	TBLPTRL
TABLRD	0,1,DUMMY ; Пустое чтение из табличной защелки, чтение памяти прог-

TBLPTR		; рамм в табличную защелку, инкрементирование
TBLRD	1,INDF0	; Чтение старшего байта из табличной защелки
TABLATH		
TABLRD	0,1,INDF0	; Чтение младшего байта из табличной защелки
TABLATL,		; чтение памяти программ в табличную защелку и инкрементирование TBLPTR



**Рис. 22** Временная диаграмма выполнения команды TABLRD (внешняя память программ)



**Рис. 23** Временная диаграмма последовательности команд TABLRD (внешняя память программ)

## Аппаратный умножитель

Микроконтроллеры имеют 8x8 битный аппаратный умножитель, включенный в АЛУ прибора. Из-за того, что умножение реализовано аппаратно, оно выполняется за один цикл. Беззнаковое умножение дает 16-ти битный результат. Результат хранится в 16-ти битном регистре PRODH:PRODL. Умножение не влияет ни на какие флаги в регистре ALUSTA. Регистры PRODH и PRODL доступны только для чтения. Реализация выполнения умножения за один цикл обеспечивает более высокую вычислительную производительность и уменьшает требования к размеру кода для алгоритмов умножения. Увеличение

производительности позволяет использовать прибор для применений, ранее предназначенных только для цифровых сигнальных процессоров.

Ниже приведено сравнение быстродействия микроконтроллеров, использующих аппаратно реализованное умножение с выполнением за один цикл, и производящих те же самые вычисления но без аппаратно реализованного умножения (см. Таблица 18).

Пример 6 показывает последовательность действий при 8x8 беззнаковом умножении. Требуется только одна команда, когда один аргумент для умножения уже загружен в регистр WREG.

Пример 7 приводит последовательность действий при 8x8 знаковом умножении. Для вычисления знаковых битов аргументов тестируется знаковый бит каждого аргумента и производится соответствующее вычитание. Результат хранится в регистрах RESH:PRODL.

Пример 9 приводит последовательность действий при 16x16 беззнаковом умножении. В примере 8 приводится используемый алгоритм. 32-х битный результат хранится в 4-х регистрах, RES3:RES0.

Пример 11 приводит последовательность действий при 16x16 знаковом умножении. В примере 10 приводится используемый алгоритм. 32-х битный результат хранится в 4-х регистрах, RES3:RES0. Для вычисления знаковых битов аргументов тестируется знаковый бит каждого аргумента и производится соответствующее вычитание.

**Таблица 18**  
Сравнение производительности

	Метод умножения	Объем программы (слов)	Кол-во циклов (максимум)	Время выполнения (мкс)		
				33 МГц	16 МГц	8 МГц
8 x 8 без знака	без аппаратного умножителя	13	69	8.364	17.25	34.50
	аппаратный умножитель	1	1	0.121	0.25	0.50
8 x 8 со знаком	без аппаратного умножителя	-	-	-	-	-
	аппаратный умножитель	7	7	0.848	1.75	3.5
16 x 16 без знака	без аппаратного умножителя	21	242	29.33 3	60.50	121.0
	аппаратный умножитель	24	24	2.91	6.0	12.0
16 x 16 со знаком	без аппаратного умножителя	52	254	30.78 8	63.50	127.0
	аппаратный умножитель	36	36	4.36	9.0	18.0

**Пример 6**  
Программа 8 x 8 битного беззнакового умножения

MOVFP	ARG1,WREG	;
MULWF	ARG2	; ARG1 * ARG2 -> PRODH:PRODL

**Пример 7**

Программа 8 x 8 битного умножения со знаком

MOVFP	ARG1,WREG	
MULWF	ARG2	; ARG1 * ARG2 -> PRODH:PRODL
MOVFP	PRODH,RESH	; PRODH -> RESH
BTFS	ARG2,SB	; Тест бита знака
SUBWF	RESH,F	; RESH = RESH - ARG1
MOVFP	ARG2,WREG	
BTFS	ARG1,SB	; Тест бита знака
SUBWF	RESH,F	; RESH = RESH - ARG2

**Пример 8**

Алгоритм 16 x 16 битного беззнакового умножения

RES3:RES0 = ARG1H:ARG1L * ARG2H:ARG2L
= (ARG1H * ARG2H * 2 <sup>16</sup> ) + (ARG1H * ARG2L * 2 <sup>8</sup> ) + (ARG1L * ARG2H * 2 <sup>8</sup> ) + (ARG1L * ARG2L)

**Пример 9**

Программа 16 x 16 битного беззнакового умножения

MOVFP	ARG1L,WREG	
MULWF	ARG2L	; ARG1L * ARG2L -> PRODH:PRODL
MOVFP	PRODH,RES1	
MOVFP	PRODL,RES0	
MOVFP	ARG1H,WREG	
MULWF	ARG2H	; ARG1H * ARG2H -> PRODH:PRODL
MOVFP	PRODH,RES3	
MOVFP	PRODL,RES2	
MOVFP	ARG1L,WREG	
MULWF	ARG2H	; ARG1L * ARG2H -> PRODH:PRODL
MOVFP	PRODL,WREG	
ADDWF	RES1,F	; Сложение промежуточных результатов
MOVFP	PRODH,WREG	
ADDWFC	RES2,F	
CLRF	WREG,F	
ADDWFC	RES3,F	
MOVFP	ARG1H,WREG	
MULWF	ARG2L	; ARG1H * ARG2L -> PRODH:PRODL
MOVFP	PRODL,WREG	
ADDWF	RES1,F	; Сложение промежуточных результатов
MOVFP	PRODH,WREG	
ADDWFC	RES2,F	
CLRF	WREG,F	
ADDWFC	RES3,F	

**Пример 10**

Алгоритм 16 x 16 битного умножения со знаком

$$\begin{aligned} & \text{RES3:RES0} = \text{ARG1H:ARG1L} * \text{ARG2H:ARG2L} \\ & = (\text{ARG1H} * \text{ARG2H} * 2^{16}) + (\text{ARG1H} * \text{ARG2L} * 2^8) + (\text{ARG1L} * \text{ARG2H} * 2^8) + (\text{ARG1L} * \\ & \text{ARG2L}) + \\ & (-1 * \text{ARG2H} < 7 > * \text{ARG1H:ARG1L} * 2^{16}) + (-1 * \text{ARG1H} < 7 > * \text{ARG2H:ARG2L} * 2^{16}) \end{aligned}$$

**Пример 11**

Программа 16 x 16 битного умножения со знаком

```

MOVFP ARG1L,WREG
MULWF ARG2L           ; ARG1L * ARG2L -> PRODH:PRODL
MOVFP PRODH,RES1
MOVFP PRODL,RES0

MOVFP ARG1H,WREG
MULWF ARG2H           ; ARG1H * ARG2H -> PRODH:PRODL
MOVFP PRODH,RES3
MOVFP PRODL,RES2

MOVFP ARG1L,WREG
MULWF ARG2H           ; ARG1L * ARG2H -> PRODH:PRODL
MOVFP PRODL,WREG
ADDWF RES1,F           ; Сложение промежуточных результатов
MOVFP PRODH,WREG
ADDWFC RES2,F
CLRF WREG,F
ADDWFC RES3,F

MOVFP ARG1H,WREG
MULWF ARG2L           ; ARG1H * ARG2L -> PRODH:PRODL
MOVFP PRODL,WREG
ADDWF RES1,F           ; Сложение промежуточных результатов
MOVFP PRODH,WREG
ADDWFC RES2,F
CLRF WREG,F
ADDWFC RES3,F

BTFSS ARG2H,7           ; ARG2H:ARG2L отрицательно?
GOTO SIGN_ARG1         ; нет, проверка ARG1
MOVFP ARG1L,WREG
SUBWF RES2,F
MOVFP ARG1H,WREG
SUBWFB RES3,F

SIGN_ARG1
BTFSS ARG1H,7           ; ARG1H:ARG1L отрицательно?
GOTO CONT_CODE         ; нет, окончание
MOVFP ARG2L,WREG
SUBWF RES2,F
    
```

MOVFP	ARG2H,WREG
SUBWFB	RES3,F

CONT\_CODE

### Порты ввода-вывода

Микроконтроллеры имеют семь портов ввода-вывода. Порты от «В» до «G» имеют регистр направления данных DDR, который используется для конфигурации выводов порта на вход или на выход. Некоторые выводы портов могут иметь дополнительное назначение.

Когда выводы портов сконфигурированы как выводы периферийного устройства, значение, содержащееся в регистре DDR неизвестно. После окончания работы с периферийным модулем пользователю желательно заново установить значение регистра DDR. Для некоторых других периферийных устройств (которые требуют входных выводов) требуется выставление битов направления передачи данных в регистре DDR.

Сигнал «сброса» переводит выводы в режим входа с высоким входным сопротивлением. Но некоторые периферийные модули могут внести изменения, такие например как перевод в режим аналогового входа или системной шины.

### Порт А

«Порт А» 6-ти разрядный. Этот порт не имеет регистра направления данных (DDR). По сигналу «сброс», выводы «порта А» принудительно конфигурируются, как «вход» с высоким входным сопротивлением. Направление данных на выводах PA4 и PA5, контролируется периферийным модулем. По сигналу «сброс», периферийный модуль неактивен, при этом выводы переведены в состояние «вход». При чтении «порта А» считывается состояние с выводов.

Вывод PA0/INT может работать как обычный вход или как вход внешнего прерывания. Вывод PA1/T0CLK может работать как обычный вход или как вход тактового сигнала для «таймера 0». Выводы PA2 и PA3 мультиплексированы с периферийным модулем СПП. Выводы PA4 и PA5 мультиплексированы с периферийным модулем USART1. Настройка PA2, PA3, PA4 и PA5 как выходов, проводится автоматически периферийным модулем.

Выводы PA2 и PA3 могут быть выходами. Они имеют «открытый сток». Для их использования как выходов необходимо записать в регистр «порта А» (PORTA) нужное значение. Значение «0» переводит вывод в низкий уровень, а значение «1» переводит вывод, в состояние высокого входного сопротивления. Необходимо использование внешнего резистора для формирования на выходе высокого уровня. Запись в PORTA влияет только на состояние выводов PA2 и PA3.

Не рекомендуется использовать команды чтение-модификация-запись (например, BCF, BSF, BTG) над регистром «порта А». Такие операции могут изменить состояние выходного триггера-защелки, что приведет к переключению от состояния выхода на вход или наоборот. Для того чтобы избежать этого, используйте дополнительный регистр, а затем переместите его значение в регистр «порта А».

**Таблица 19**

Название	Бит	Тип входного буфера	Функция
PA0/INT	0	Триггер Шмитта	Вход или вход внешнего прерывания.
PA1/T0CLK	1		Вход или вход тактового сигнала «таймера 0».
PA2/SS/SCL	2		Вход/выход, выход – открытый сток, или вход для SPI в режиме ведомого, или вход синхросигнала для шины I2C.
PA3/SDI/SDA	3		Вход/выход, выход – открытый сток, или вход данных для SPI, или вход данных для шины I2C.
PA4/RX1/DT1	4		Вход или вход приемника асинхронного USART1, или вход/выход данных синхронного USART1.
PA5/TX1/CK1	5		Вход или выход асинхронного передатчика USART1, или вход/выход синхросигнала синхронного USART1.
RBPU	7		Бит управления резисторами подтяжки «порта В». Активный уровень – низкий.

### **Регистр порта В и регистр направления данных DDRB**

«Порт В» - это 8-ми разрядный двунаправленный порт ввода/вывода. Направление данных управляется регистром направления DDRB. Значение «1» в DDRB конфигурирует соответствующие выводы порта как ввод. Значение «0» в регистре DDRB конфигурирует соответствующие выводы порта как выход. При чтении «порта В» считывается состояние с выводов.

Каждый вывод «порта В» имеет внутренний резистор подтяжки к напряжению питания. Все резисторы подтяжки одновременно управляются битом RBPU (седьмой бит регистра PORTA). Резисторы подтяжки включаются при установке RBPU=0. Подтяжка автоматически выключается, если вывод порта конфигурируется, как выход. По любому сигналу «сброс» резисторы подтяжки включаются.

«Порт В» может использоваться для внешних прерываний. Для этого могут быть использованы любые его выводы, которые сконфигурированы как входы, то есть любой вывод сконфигурированный как выход исключается из обработки прерывания. Сигналы на входах «порта В» сравниваются со значением, записанным в регистре порта (PORTB). При несовпадении выставляется флаг запроса прерывания RBIF(PIR1<7>). Выставление этого флага может например вывести устройство из SLEEP режима.

В программе обработки прерывания требуется следующая последовательность для сброса флага запроса прерывания:

- Прочитайте и запишите обратно значение порта «В». Это приведет к окончанию несовпадения.
- После этого обнулите флаг запроса прерывания RBIF.



По сигналу «Сброс» состояние бита RBIF неопределенно, до тех пор, пока значение регистра защелки «порта В» отличается от значения на выводах.

**Таблица 20**

Название	Бит	Тип входного буфера	Функция
PB0/CAP1	0	Триггер Шмита	Вход/выход или вход «захвата 1» (регистрации событий). Программируемая подтяжка к $U_{CC}$ и возможность прерывания по изменению состояния.
PB1/CAP2	1		Вход/выход или вход «захвата 2» (регистрации событий). Программируемая подтяжка к $U_{CC}$ и возможность прерывания по изменению состояния.
PB2/PWM1	2		Вход/выход или выход «ШИМ 1». Программируемая подтяжка к $U_{CC}$ и возможность прерывания по изменению состояния.
PB3/PWM2	3		Вход/выход или выход «ШИМ 2». Программируемая подтяжка к $U_{CC}$ и возможность прерывания по изменению состояния.
PB4/TCLK12	4		Вход/выход или вход внешнего тактового сигнала «таймера 1» и «таймера 2». Программируемая подтяжка к $U_{CC}$ и возможность прерывания по изменению состояния.
PB5/TCLK3	5		Вход/выход или вход внешнего тактового сигнала «таймера 3». Программируемая подтяжка к $U_{CC}$ и возможность прерывания по изменению состояния.
PB6/SCK	6		Вход/выход или тактовый сигнал в режимах ведущий/ведомый для SPI. Программируемая подтяжка к $U_{CC}$ и возможность прерывания по изменению состояния.
PB7/SDO	7		Вход/выход или выход данных для SPI. Программируемая подтяжка к $U_{CC}$ и возможность прерывания по изменению состояния.

### Регистр порта С и регистр направления данных DDRC

«Порт С» - это 8-ми разрядный двунаправленный порт ввода/вывода. Направление данных управляется регистром направления DDRC. Значения «1» в регистре DDRC конфигурирует соответствующие выводы порта как вход. Значения «0» в регистре DDRC конфигурирует соответствующие выводы порта как выход. При чтении регистра «порта С» (PORTC) считывается состояние с выводов, а при записи в регистр, значение записывается в регистр защелки. В

режимах «микропроцессор» и «расширенный микроконтроллер» «порт С» является младшим байтом шины адреса/данных (AD7-AD0) системной шины.

**Таблица 21**

Название	Бит	Тип входного буфера	Функция
PC0/AD0	0	ТТЛ	Вход/выход или вывод системной шины адреса/данных
PC1/AD1	1		Вход/выход или вывод системной шины адреса/данных
PC2/AD2	2		Вход/выход или вывод системной шины адреса/данных
PC3/AD3	3		Вход/выход или вывод системной шины адреса/данных
PC4/AD4	4		Вход/выход или вывод системной шины адреса/данных
PC5/AD5	5		Вход/выход или вывод системной шины адреса/данных
PC6/AD6	6		Вход/выход или вывод системной шины адреса/данных
PC7/AD7	7		Вход/выход или вывод системной шины адреса/данных

### Регистр порта D и регистр направления данных DDRD

«Порт D» - это 8-ми разрядный двунаправленный порт ввода/вывода. Направление данных управляется регистром направления DDRD. Значения «1» в регистре DDRD конфигурирует соответствующие выходы порта как вход. Значения «0» в регистре DDRD конфигурирует соответствующие выходы порта как выход. При чтении регистра «порта D» (PORTD) считывается состояние с выводов, а при записи в регистр, значение записывается в регистр защелку. В режимах «микропроцессор» и «расширенный микроконтроллер» «порт D» является старшим байтом шины адреса/данных (AD15-AD8) системной шины.

**Таблица 22**

Название	Бит	Тип входного буфера	Функция
PD0/AD8	0	ТТЛ	Вход/выход или вывод системной шины адреса/данных
PD1/AD9	1		Вход/выход или вывод системной шины адреса/данных
PD2/AD10	2		Вход/выход или вывод системной шины адреса/данных
PD3/AD11	3		Вход/выход или вывод системной шины адреса/данных

PD4/AD12	4		Вход/выход или вывод системной шины адреса/данных
PD5/AD13	5		Вход/выход или вывод системной шины адреса/данных
PD6/AD14	6		Вход/выход или вывод системной шины адреса/данных
PD7/AD15	7		Вход/выход или вывод системной шины адреса/данных

### Регистр порта E и регистр направления данных DDRE

«Порт E» - это 4 разрядный двунаправленный порт ввода/вывода. Направление данных определяется регистром направления DDRE. Значение «1» в регистре DDRE конфигурирует соответствующий вывод порта как вход. Значение «0» в регистре DDRE конфигурирует соответствующий вывод порта как выход. При чтении регистра «порта E» (PORTE) считывается состояние с выводов, а при записи в регистр, значение записывается в регистр защелку. В режимах «микропроцессор» и «расширенный микроконтроллер» «порт E» выводит управляющие сигналы для системной шины: разрешение защелки адреса (ALE), разрешение выхода (OE) и запись (WR). Активный уровень для сигналов OE и WR - низкий. Четвертый вывод «порта E» является универсальным входом/выходом или входом «захвата 4» (регистрации событий).

**Таблица 23**

Название	Бит	Тип входного буфера	Функция
PE0/ALE	0	ТТЛ	Вход/выход или выход сигнала фиксации адреса внешней памяти.
PE1/OE	1		Вход/выход или выход сигнала чтения данных из внешней памяти.
PE2/WR	2		Вход/выход или выход сигнала разрешения записи во внешнюю память.
PE3/CAP4	3		Вход/выход или вход «захвата 3» (регистрации событий).

### Регистр порта F и регистр направления данных DDRF

«Порт F» - это 8-ми разрядный двунаправленный порт ввода/вывода. Направление данных управляется регистром направления DDRF. Значения «1» в регистре DDRF конфигурирует соответствующие выводы порта как вход. Значения «0» в регистре DDRF конфигурирует соответствующие выводы порта как выход. При чтении регистра «порта F» (PORTF) считывается состояние с выводов, а при записи в регистр, значение записывается в регистр защелку. Все восемь бит «порта F» могут быть аналоговыми входами 8 каналов 10- битного АЦП.

По сигналу «сброс», все выводы порта автоматически конфигурируются как аналоговые входы. Конфигурирование их как цифровых входов/выходов проводится программно.

**Таблица 24**

Название	Бит	Тип входного буфера	Функция
PF0/AN4	0	Триггер Шмитта	Вход/выход или аналоговый вход 4.
PF1/AN5	1		Вход/выход или аналоговый вход 5.
PF2/AN6	2		Вход/выход или аналоговый вход 6.
PF3/AN7	3		Вход/выход или аналоговый вход 7.
PF4/AN8	4		Вход/выход или аналоговый вход 8.
PF5/AN9	5		Вход/выход или аналоговый вход 9.
PF6/AN10	6		Вход/выход или аналоговый вход 10.
PF7/AN11	7		Вход/выход или аналоговый вход 11.

### Регистр порта G и регистр направления данных DDRG

«Порт G» - это 8-ми разрядный двунаправленный порт ввода/вывода. Направление данных управляется регистром направления DDRG. Значения «1» в регистре DDRG конфигурирует соответствующие выводы порта как вход. Значения «0» в регистре DDRG конфигурирует соответствующие выводы порта как выход. При чтении регистра «порта G» (PORTG) считывается состояние с выводов, а при записи в регистр, значение записывается в регистр защелку. Четыре младших разрядов «порта G» могут быть аналоговыми входами 4 каналов 10- битного АЦП. Остальные биты могут быть использованы периферийными устройствами (см. Таблица 25).

По сигналу «сброс», четыре младших вывода порта автоматически конфигурируются как аналоговые входы. Конфигурирование их как цифровых входов/выходов проводится программно.

**Таблица 25**

Название	Бит	Тип входного буфера	Функция
PG0/AN3	0	Триггер Шмитта	Вход/выход или аналоговый вход 3.
PG1/AN2	1		Вход/выход или аналоговый вход 2.
PG2/AN1/U <sub>REF-</sub>	2		Вход/выход или аналоговый вход 1 или отрицательный вход опорного напряжения АЦП.
PG3/AN0/U <sub>REF+</sub>	3		Вход/выход или аналоговый вход 0 или положительный вход опорного напряжения АЦП.
PG4/CAP3	4		Вход/выход или вход «захвата 3» (регистрации событий).
PG5/PWM3	5		Вход/выход или выход «ШИМ 3».

Название	Бит	Тип входного буфера	Функция
PG6/RX2/DT2	6		Вход/выход или вход асинхронного приемника или вход/выход синхронных данных (USART2).
PG7/TX2/CK2	7		Вход/выход или выход асинхронного передатчика или линия тактовых импульсов в синхронном режиме (USART2).

### Блок «таймер 0»

Блок «таймер 0» состоит из 16-разрядного таймер/счетчика. Старший байт представлен регистром TMR0H, а младший байт – регистром TMR0L. Оба регистра доступны по чтению и записи. Программно-управляемый 8-разрядный делитель частоты позволяет создать на основе блока 24-разрядный счетчик. Источник тактовых импульсов задается битом T0CS регистра T0STA. Счетчик может изменять свое состояние или от внутренних тактовых импульсов, или от внешних, подаваемых на вход PA1/T0CLK. Управление «таймером 0» осуществляется с помощью регистра T0STA.

**Таблица 26**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
INTEDG	TOSE	TOCS	TOPS3	TOPS2	TOPS1	TOPS0	-
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7</b>	<b>INTEDG:</b> бит выбора управляющего перепада сигнала на выводе PA0/INT для запроса прерывания. Этот бит выбирает, на фронте или спаде сигнала будет происходить запрос прерывания. 1 = фронт сигнала на выводе PA0/INT генерирует прерывание 0 = спад сигнала на выводе PA0/INT генерирует прерывание						
<b>бит 6</b>	<b>TOSE:</b> бит выбора управляющего перепада сигнала при внешнем тактировании «таймера 0». Этот бит выбирает, на фронте или спаде сигнала «таймер 0» будет инкрементироваться. <u>Когда T0CS=0 (внешнее тактирование):</u> 1 = фронт сигнала на выводе PA1/T0CLK инкрементирует «таймер 0» и/или устанавливает T0CKIF - бит 0 = спад сигнала на выводе PA1/T0CLK инкрементирует «таймер 0» и/или устанавливает T0CKIF - бит <u>Когда T0CS=1 (внутреннее тактирование):</u> значение бита безразлично						

<b>бит 5</b>	<b>TOCS:</b> бит выбора источника тактирования для «таймера 0». Этот бит выбирает источник синхронизации для «таймера 0». 1 = внутренняя тактовая частота с генератора циклов 0 = внешнее тактирование с вывода PA1/T0CLK
<b>бит 4-1</b>	<b>TOPS3:TOPS0:</b> биты выбора предделителя для таймера 0. Эти биты позволяют выбрать величину деления входного тактового сигнала предделителем: 0000 - 1:1    0001 - 1:2    0010 - 1:4    0011 - 1:8 0100 - 1:16    0101 - 1:32    0110 - 1:64    0111 - 1:128 1xxx - 1:256
<b>бит 0</b>	не реализовано: читается значение равное нулю.

Обозначения:

R = бит для чтения; W = бит с возможностью записи; U = бит не реализован, читается как 0;

-n = значение бита после сброса по включению питания: 1 = установлен; 0 = сброшен; x = значение не известно.

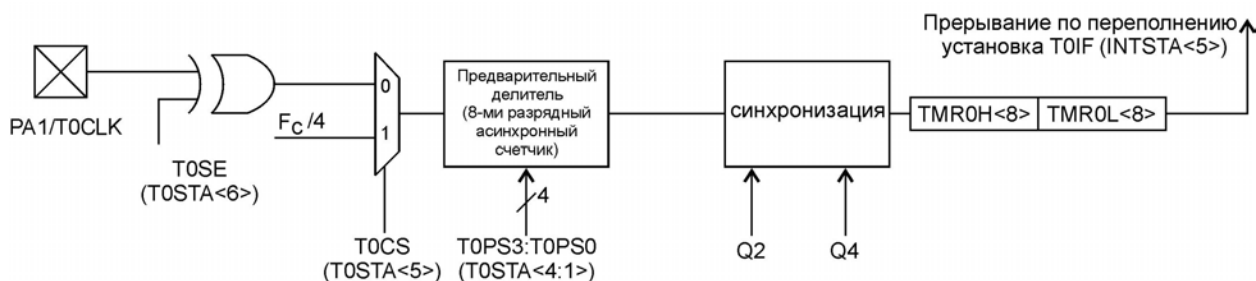
Если бит TOCS установлен в «1», то инкрементирование счетчика «таймера 0» происходит от тактовых импульсов внутреннего генератора, который является источником синхронизации для всего микроконтроллера. Если бит TOCS сброшен в «0», то инкрементирование счетчика происходит от тактовых импульсов с входа PA1/T0CLK (внешний источник тактовых импульсов). В случае использования внешнего источника тактовых импульсов, бит T0SE определяет полярность фронта, по которому изменяется состояние счетчика. Если бит T0SE установлен в «1», счетчик будет изменять свое состояние по переднему фронту сигнала PA1/T0CLK, а если бит T0SE сброшен в «0», то по заднему фронту (спаду) сигнала PA1/T0CLK. Предварительный 8-разрядный делитель с программируемым коэффициентом деления частоты (ПДПКД) осуществляет деление частоты тактовых импульсов в диапазоне от 1:1 до 1:256, в зависимости от состояния битов TOPS3:TOPS0. Таймер циклически изменяет свое состояние в диапазоне значений от 0000h до FFFFh с шагом 1. При достижении максимального значения (FFFFh) (состояние переполнения) устанавливается флаг (T0IF) запроса прерывания по переполнению «таймера 0». Этот запрос на обработку прерывания может быть замаскирован, путем сброса в «0» соответствующего бита разрешения запроса прерывания (T0IE). Флаг запроса на обработку прерывания от «таймера 0» (T0IF) сбрасывается программно программой обработки прерывания.

Если для «таймера 0» используется внешний источник тактовых импульсов, то осуществляется синхронизация внешнего тактового сигнала и внутренней тактовой частоты. Рис. 25 иллюстрирует механизм синхронизации. Тактовый сигнал синхронизируется после предделителя (ПДПКД). Сигнал с выхода предделителя (ПДПКД) сэмплируется два раза в каждом командном цикле с целью детектирования появления переднего или заднего фронта. Требования к параметрам внешнего сигнала синхронизации приведены в таблице электрических параметров. Процедура синхронизации внешних тактовых импульсов, вносит задержку от времени прихода активного фронта до момента изменения таймером 0 своего состояния. На рис.10-2 показано, что эта задержка может составлять от 3Tc до 7Tc.

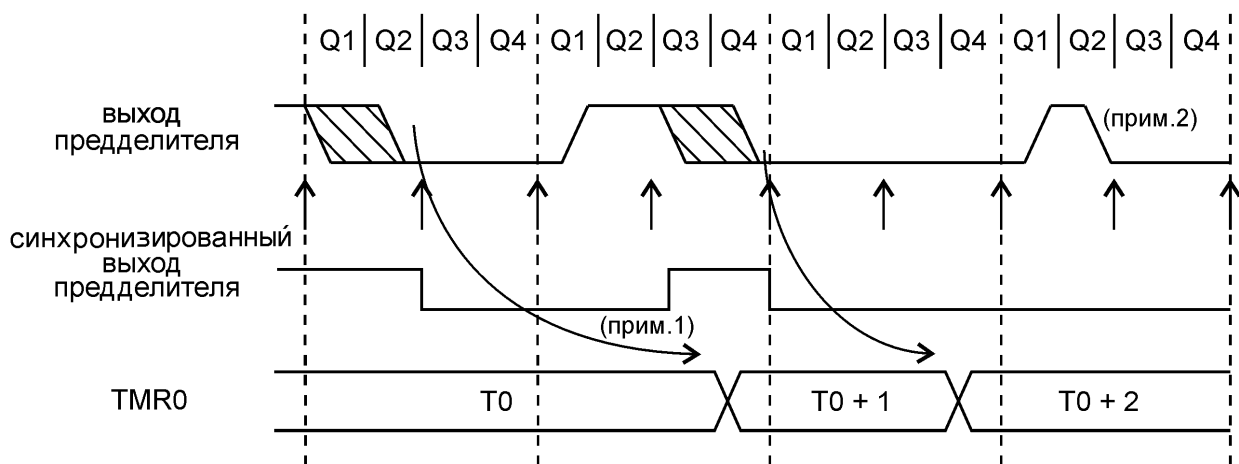
Проблема считывания 16-разрядного значения регистров TMR0L и TMR0H заключается в том, что после считывания младшего (или старшего) байта, его значение может измениться от FFh к 00h. Для обеспечения однозначного считывания состояния счетчика рекомендуется маскировать сигнал запроса на обработку прерывания.

Запись в любой из регистров TMR0L и TMR0H блокирует изменение соответствующей части счетчика «таймера 0» в последующем после записи цикле микроконтроллера, при этом запись не оказывает влияния на другую часть счетчика. Поэтому рекомендуется последовательно производить запись сначала регистра TMR0L, а затем TMR0H. Запись в любой из регистров TMR0L или TMR0H сбрасывает в исходное состояние предделитель (ПДПКД).

Установка коэффициента деления предделителя (ПДПКД) полностью зависит от состояния регистра T0STA, то есть значение коэффициента может быть изменено «на лету» во время исполнения программы. Перед изменением коэффициента деления рекомендуется сбрасывать ПДПКД.



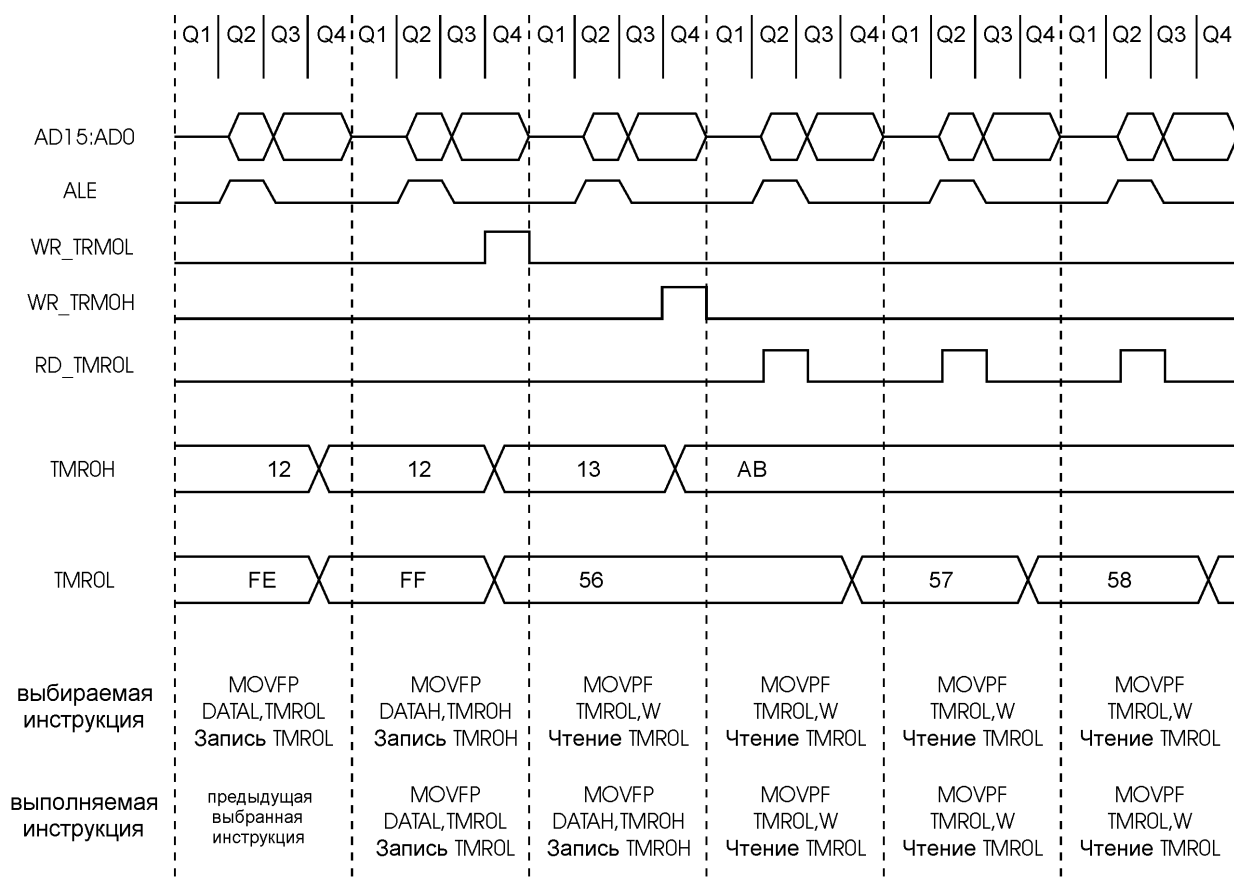
**Рис. 24** Блок-схема модуля «таймер 0»



**Рис. 25** Функционирование «таймер 0» от внешнего источника тактовых импульсов

**Примечания:**

1. Задержка от времени прихода активного фронта до момента изменения TMR0 своего состояния составляет от  $3 \cdot T_c$  до  $7 \cdot T_c$ .
2. Длительность импульса на выходе ПДПКД меньше частоты синхронизации. В этом случае состояние счетчика TMR0 не изменится.



**Рис. 26** Функционирование «таймер 0» при обращении к регистрам TMR0L и TMR0H

Примечание:

В примере записывается значение TMR0 равное AB56h.

**Таймер 1, таймер 2, таймер 3, ШИМ, захват (регистрация событий)**

«Таймер 1» и «таймер 2» представляют собой два 8-ми разрядных таймер/счетчика (TMR1 и TMR2), каждый с 8-ми разрядным регистром периода (PR1 и PR2) и отдельными флагами запроса прерывания при переполнении. Таймеры 1 и 2 могут работать как таймеры, инкрементирующиеся от внутренних тактовых импульсов  $F_c/4$ , либо инкрементирующиеся на заднем фронте внешнего тактового сигнала с вывода PB4/TCLK12. Они могут быть программно объединены в единый 16-ти разрядный таймер/счетчик. Эти таймеры также используются как опорные для модулей широтно-импульсных модуляторов.

«Таймер 3» является 16-разрядным таймером/счетчиком (регистры TMR3H и TMR3L). Он содержит два дополнительных регистра (PR3H/CA1H:PR3L/CA1L), которые используются как 16-ти разрядный регистр периода, или как 16-ти разрядный регистр «захвата 1». Для приращения «таймера 3» может использоваться или внутренний тактовый сигнал  $F_c/4$ , или внешний тактовый сигнал с вывода PB5/TCLK3. Этот таймер используется как опорный для всех 16-ти разрядных захватов (регистраций событий). Шесть других дополнительных регистров включают регистры захвата (регистрации событий): 2 (CA2H:CA2L), 3 (CA3H:CA3L) и 4 (CA4H:CA4L).



Микроконтроллер имеет три выхода ШИМ (широотно-импульсных модуляторов) и четыре выхода захвата (регистрации событий). Таблица 27 показывает использование ресурсов таймеров для реализации функций ШИМ и захватов.

**Таблица 27**

Требуемые ресурсы таймеров для различных функций

Функция	Ресурсы таймера
ШИМ 1	Таймер 1
ШИМ 2	Таймер 1 или таймер 2
ШИМ 3	Таймер 1 или таймер 2
Захват 1	Таймер 3
Захват 2	Таймер 3
Захват 3	Таймер 3
Захват 4	Таймер 3

**Таблица 28**

Регистр TCON1 (адрес: 16h, банк 3)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7,6</b>		<b>CA2ED1:CA2ED0:</b> биты выбора режима «захвата 2». 00 = Захват на каждом заднем фронте (спаде) 01 = Захват на каждом переднем фронте 10 = Захват на каждом 4-м переднем фронте 11 = Захват на каждом 16-м переднем фронте					
<b>бит 5,4</b>		<b>CA1ED1:CA1ED0:</b> биты выбора режима «захвата 1». 00 = Захват на каждом заднем фронте (спаде) 01 = Захват на каждом переднем фронте 10 = Захват на каждом 4-м переднем фронте 11 = Захват на каждом 16-м переднем фронте					
<b>бит 3</b>		<b>T16:</b> бит выбора режима «таймера 1 и 2». 1 = «таймер 2» и «таймер 1» образуют 16-разрядный таймер 0 = «таймер 2» и «таймер 1» являются двумя 8-ми разрядными таймерами					
<b>бит 2</b>		<b>TMR3CS:</b> бит выбора источника тактовых импульсов «таймера 3» 1 = внешний тактовый сигнал с вывода PB5/TCLK3 (приращение по заднему фронту сигнала) 0 = внутренний тактовый сигнал FC/4					
<b>бит 1</b>		<b>TMR2CS:</b> бит выбора источника тактовых импульсов «таймера 2». 1 = внешний тактовый сигнал с вывода PB4/TCLK12 (приращение по заднему фронту сигнала) 0 = внутренний тактовый сигнал FC/4					

<b>бит 0</b>	<p><b>TMR1CS:</b> бит выбора источника тактовых импульсов «таймера 1».</p> <p>1 = внешний тактовый сигнал с вывода PB4/TCLK12 (приращение по заднему фронту сигнала)</p> <p>0 = внутренний тактовый сигнал FC/4</p>
--------------	---

Обозначения:

R = бит для чтения; W = бит с возможностью записи; U = бит не реализован, читается как 0;

-n = значение бита после сброса по включению питания: 1 = установлен; 0 = сброшен; x = значение не известно.

**Таблица 29**  
Регистр TCON2 (адрес: 17h, банк 3)

R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/PR3	TMR3ON	TMR2ON	TMR1ON
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7</b>		<p><b>CA2OVF:</b> бит переполнения «захвата 2».</p> <p>Этот бит означает, что значение регистров захвата (CA2H:CA2L) не было считано до того как произошел следующий захват. Регистр захвата сохраняет старое несчитанное значение захвата (последний захват перед переполнением). Последующие захваты не обновят значение регистра захвата до тех пор, пока регистр захвата не будет считан (оба байта).</p> <p>1 = произошло переполнение</p> <p>0 = нет переполнения</p>					
<b>бит 6</b>		<p><b>CA1OVF:</b> бит переполнения «захвата 1».</p> <p>Этот бит означает, что значение регистров захвата (PR3H/CA1H:PR3L/CA1L) не было считано до того как произошел следующий захват. Регистр захвата сохраняет старое несчитанное значение захвата (последний захват перед переполнением). Последующие захваты не обновят значение регистра захвата до тех пор, пока регистр захвата не будет считан (оба байта).</p> <p>1 = произошло переполнение</p> <p>0 = нет переполнения</p>					
<b>бит 5</b>		<p><b>PWM2ON:</b> бит включения «ШИМ 2».</p> <p>1 = «ШИМ 2» включен, вывод PB3/PWM2 игнорирует состояние бита DDRB&lt;3&gt;</p> <p>0 = «ШИМ 2» выключен, вывод PB3/PWM2 использует состояние бита DDRB&lt;3&gt; для направления передачи данных.</p>					
<b>бит 4</b>		<p><b>PWM1ON:</b> бит включения «ШИМ 1».</p> <p>1 = «ШИМ 1» включен, вывод PB2/PWM1 игнорирует состояние бита DDRB&lt;2&gt;</p> <p>0 = «ШИМ 1» выключен, вывод PB2/PWM1 использует состояние бита DDRB&lt;2&gt; для направления передачи данных.</p>					

<b>бит 3</b>	<b>CA1/PR3:</b> бит выбора режима регистра CA1/PR3 1 = активирует «захват 1», регистр PR3H/CA1H:PR3L/CA1L является регистром «захвата 1». «Таймер 3» работает без регистра периода. 0 = включает регистр периода. PR3H/CA1H:PR3L/CA1L является регистром периода для «таймера 3».
<b>бит 2</b>	<b>TMR3ON:</b> бит включения «таймера 3». 1 = «таймер 3» включен 0 = «таймер 3» остановлен
<b>бит 1</b>	<b>TMR2ON:</b> бит включения «таймера 2». Этот бит контролирует приращение «таймера 2». Если таймеры 1 и 2 объединены в единый 16-ти разрядный таймер (бит T16 установлен), то должен быть установлен TMR2ON. Это позволяет изменяться старшему байту 16-ти разрядного таймера. 1 = запускает «таймер 2» 0 = останавливает «таймер 2»
<b>бит 0</b>	<b>TMR1ON:</b> бит включения «таймера 1». <u>Если бит T16=1 (16-ти разрядный режим таймера 2 и таймера 1):</u> 1 = запускает 16-ти разрядный таймер TMR2:TMR1 0 = останавливает 16-ти разрядный таймер <u>Если бит T16=0 (8-ми разрядный режим таймеров):</u> 1 = запускает 8-ми разрядный «таймер 1» 0 = останавливает 8-ми разрядный «таймер 1»

Обозначения:

R = бит для чтения; W = бит с возможностью записи; U = бит не реализован, читается как 0;

-n = значение бита после сброса по включению питания: 1 = установлен; 0 = сброшен; x = значение не известно.

**Таблица 30**  
Регистр TCON3 (адрес: 16h, банк 7)

U-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	CA4OVF	CA3OVF	CA4ED1	CA4ED0	CA3ED1	CA3ED0	PWM3ON
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7</b>	не реализовано: читается значение равное нулю.						
<b>бит 6</b>	<b>CA4OVF:</b> бит переполнения «захвата 4». Этот бит означает, что значение регистров захвата (CA4H:CA4L) не было считано до того как произошел следующий захват. Регистр захвата сохраняет старое несчитанное значение захвата (последний захват перед переполнением). Последующие захваты не обновят значение регистра захвата до тех пор, пока регистр захвата не будет считан (оба байта). 1 = произошло переполнение 0 = нет переполнения						

<b>бит 5</b>	<p><b>CA3OVF:</b> бит переполнения «захвата 3».</p> <p>Этот бит означает, что значение регистров захвата (CA3H:CA3L) не было считано до того как произошел следующий захват. Регистр захвата сохраняет старое несчитанное значение захвата (последний захват перед переполнением). Последующие захваты не обновят значение регистра захвата до тех пор, пока регистр захвата не будет считан (оба байта).</p> <p>1 = произошло переполнение 0 = нет переполнения</p>
<b>бит 4,3</b>	<p><b>CA4ED1:CA4ED0:</b> биты выбора режима «захвата 4».</p> <p>00 = захват на каждом заднем фронте (спаде) 01 = захват на каждом переднем фронте 10 = захват на каждом 4-м переднем фронте 11 = захват на каждом 16-м переднем фронте</p>
<b>бит 2,1</b>	<p><b>CA3ED1:CA3ED0:</b> биты выбора режима «захвата 3».</p> <p>00 = захват на каждом заднем фронте (спаде) 01 = захват на каждом переднем фронте 10 = захват на каждом 4-м переднем фронте 11 = захват на каждом 16-м переднем фронте</p>
<b>бит 0</b>	<p><b>PWM3ON:</b> бит включения «ШИМ 3».</p> <p>1 = «ШИМ 3» включен, вывод PG5/PWM3 игнорирует состояние бита DDRG&lt;5&gt; 0 = «ШИМ 3» выключен, вывод PG5/PWM3 использует состояние бита DDRG&lt;5&gt; для направления передачи данных.</p>

Обозначения:

R = бит для чтения; W = бит с возможностью записи; U = бит не реализован, читается как 0;

-n = значение бита после сброса по включению питания: 1 = установлен; 0 = сброшен; x = значение не известно.

**«Таймер 1» и «таймер 2». «Таймер 1» и «таймер 2» в 8-ми разрядном режиме**

Если бит T16=0, то «таймер 1» и «таймер 2» работают в 8-разрядном режиме. Источники тактовых сигналов для таймеров могут быть настроены индивидуально для каждого таймера, это или внутренняя тактовая частота  $F_C/4$ , или внешний сигнал с вывода PB4/TCLK12. Источник импульсов таймера конфигурируется битом TMRxCS (где: x=1 для «таймера 1» или =2 для «таймера 2»). Если бит TMRxCS=0, то источник тактовых импульсов внутренний и таймер инкрементируется каждый командный цикл (частота  $F_C/4$ ). Если бит TMRxCS установлен, то источник импульсов внешний сигнал с вывода PB4/TCLK12 и таймер инкрементируется на каждом заднем фронте (спаде) сигнала TCLK12. Сигнал с входа PB4/TCLK12 синхронизируется с внутренним тактовым сигналом, это вызывает задержку между спадом сигнала на выводе и приращением таймера. Временные требования к внешнему тактовому сигналу смотрите в «электрических параметрах».

Значение таймера инкрементируется от 00h до момента равенства с значением регистра периода (PRx). В следующем цикле приращения он

сбрасывается в 00h. Флаг запроса прерывания от таймера устанавливается, когда таймер сбрасывается. «Таймер 1» и «таймер 2» имеют индивидуальные биты флагов запроса прерывания (соответственно TMR1IF и TMR2IF).

Каждый таймер имеет индивидуальный бит разрешения прерываний (TMRxIE). Прерывание от таймера может быть разрешено установкой и запрещено очисткой этого бита. Также для разрешения прерывания должен быть установлен бит разрешения прерываний от периферийных устройств (PEIE=1) и сброшен бит глобального запрещения прерываний (GLINTD=0).

Таймеры могут включаться и выключаться программно установкой или сбросом соответствующего управляющего бита TMRxON.

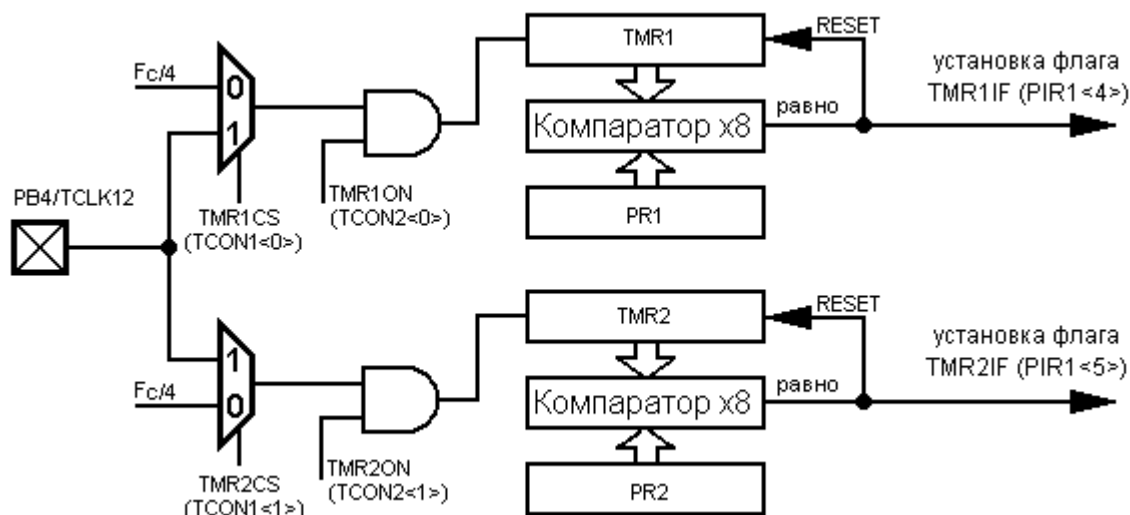


Рис. 27 «Таймер 1» и «таймер 2» в режиме двух 8-разрядных таймер/счетчиков

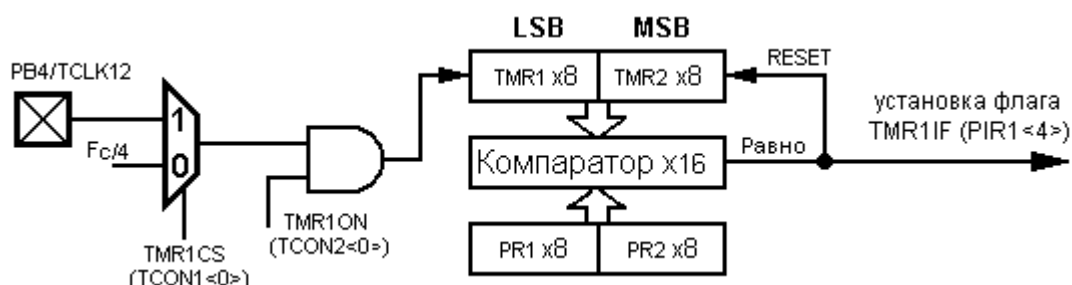


Рис. 28 «Таймер 1» и «таймер 2» в режиме 16-ти разрядного таймер/счетчика

### «Таймер 1» и «таймер 2» в 16-ти разрядном режиме

Если бит T16=1, то таймеры 1 и 2 объединяются в один 16-ти разрядный таймер TMR2:TMR1. 16-ти разрядный таймер инкрементируется до тех пор, пока его значение не совпадет с значением 16-ти разрядного регистра периода (PR1:PR2). В следующем цикле приращения значение таймера сбрасывается в 0000h, и устанавливается бит запроса прерывания TMR1IF.

Для увеличения «таймера 2» бит TMR2ON должен быть установлен, т.е. для работы 16-ти разрядного таймера должен быть установлен как бит TMR1ON, так и TMR2ON (см. Таблица 31).

Источник тактовых импульсов для 16-разрядного таймера задается битом TMR1CS, значение бита TMR2CS безразлично. Если бит TMR1CS=0, то источник тактовых импульсов внутренний и таймер инкрементируется каждый командный цикл (частота  $F_C/4$ ). Если бит TMR1CS установлен, то источник импульсов внешний сигнал с вывода PB4/TCLK12 и таймер инкрементируется на каждом заднем фронте (спаде) сигнала TCLK12. Сигнал с входа TCLK12 синхронизируется (дважды в каждом цикле микроконтроллера) с внутренним тактовым сигналом, это вызывает задержку между спадом сигнала на выводе и приращением таймера. Временные требования к внешнему тактовому сигналу смотрите в «электрических параметрах».

**Таблица 31**  
Включение 16-ти разрядного таймера

T16	TMR2ON	TMR1ON	Результат
1	1	1	16-ти разрядный таймер работает
1	0	1	Увеличивается только «таймер 1»
1	x	0	16-ти разрядный таймер выключен
0	1	1	Таймеры работают в 8-ми разрядном режиме

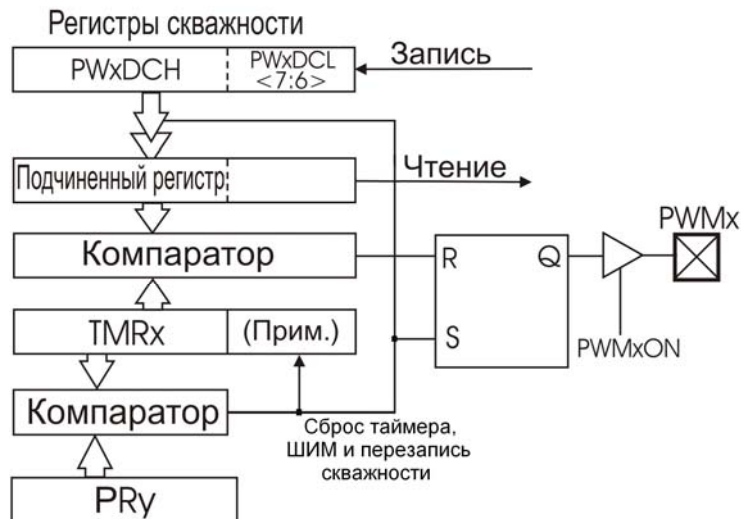
### **Использование выходов широтно-импульсных модуляторов (ШИМ)**

Микроконтроллер содержит три высокоскоростных выхода ШИМ. Выход «ШИМ 1» использует как опорный «таймер 1», а «ШИМ 2» и «ШИМ 3» могут быть независимо сконфигурированы для использования в качестве опорного «таймер 1» или «таймер 2». Выходы ШИМ подключены к выводам PB2/PWM1, PB3/PWM2 и PG5/PWM3.

Каждый выход ШИМ имеет максимальное разрешение 10 бит. При 10-ти битовом разрешении выходная частота ШИМ равняется 32.2 КГц (при тактовой частоте 33 МГц), а при 8-ми битовом разрешении частота выхода ШИМ равна 128.9 КГц. Сквозность сигнала на выходе может варьироваться от 0% до 100%. На Рис. 29 показана упрощенная блок-схема модуля ШИМ.

Регистры сквозности имеют двойную буферизацию для работы без помех. На Рис. 30 продемонстрировано появление помех для случая отсутствия двойной буферизации регистров сквозности. Пунктирная линия показывает выход ШИМ, для случая отсутствия двойной буферизации. Если новое значение сквозности записывается после того, как таймер прошел данное значение, ШИМ не сбрасывается в текущем цикле. В этом примере период ШИМ равен 50. прежнее значение сквозности равно 30, новое значение равно 10.

Для включения «ШИМ 1» необходимо установить бит PWM1ON (TCON2<4>). Когда бит PWM1ON=1, вывод PB2/PWM1 становится выходом «ШИМ 1» и независимо от бита направления передачи данных (DDRB<2>) конфигурируется как выход. Если бит PWM1ON=0, то направление передачи данных вывода задается битом направления передачи (DDRB<2>). Подобным образом, бит PWM2ON (TCON2<5>) задает конфигурацию вывода PB3/PWM2, а бит PWM3ON (TCON3<0>) - конфигурацию вывода PG5/PWM3.



**Рис. 29** Упрощенная блок-схема модуля ШИМ

**Примечание:**

8-ми разрядный таймер объединен с 2-х разрядным значением фазы Q для создания 10-ти разрядного значения.

Период выхода «ШИМ 1» определяется «таймером 1» и его регистром периода PR1. Для «ШИМ 2», если бит TM2PW2 (PW2DCL<5>)=0, период определяется «таймером 1» и PR1, а если TM2PW2=1, то «таймером 2» и PR2. Для «ШИМ 3», если бит TM2PW3 (PW3DCL<5>)=0, определяется «таймером 1» и PR1, а если TM2PW3=1, то «таймером 2» и PR2.

Работа двух разных выходов ШИМ на двух разных таймерах допускает различные периоды ШИМ. Работа всех ШИМ от «таймера 1» позволяет лучше использовать ресурсы, освобождая «таймер 2» для работы в качестве 8-ми разрядного таймера. «Таймер 1» и «таймер 2» не могут использоваться как 16-ти разрядный таймер, если используется любой ШИМ.

Периоды ШИМ могут высчитываться следующим образом:

$$\begin{aligned} \text{период «ШИМ 1»} &= [(PR1) + 1] * 4 * T_c \\ \text{период «ШИМ 2»} &= [(PR1) + 1] * 4 * T_c \text{ или } [(PR2) + 1] * 4 * T_c \\ \text{период «ШИМ 3»} &= [(PR1) + 1] * 4 * T_c \text{ или } [(PR2) + 1] * 4 * T_c \end{aligned}$$

Скважность ШИМ определяется 10-ти битным значением DCx<9:0>. Старшие 8 бит находятся в регистре PWxDCH, а младшие 2 бита в регистре PWxDCL<7:6>. Таблица 32 демонстрирует максимальную частоту ШИМ в зависимости от значения в регистре периода. Количество битов разрешения, которого может достигнуть ШИМ, зависит от тактовой частоты микроконтроллера и частоты ШИМ.

Максимальное разрешение ШИМ (биты) для заданной частоты ШИМ: =  $\log(FC/FPWM) / \log(2)$ , где: FPWM = 1/ период ШИМ.

Длительность импульса ШИМ = (DCx)\*Tc (где DCx представляет 10-ти битное значение из PWxDCH:PWxDCL).

Если DCx = 0, тогда длительность импульса равна 0. Если PRx = PWxDCH, тогда выход ШИМ будет низким от одного до четырех тактов тактового генератора (в зависимости от состояния битов PWxDCL<7:6>). Чтобы скважность была 100%, значение PWxDCH должно быть больше, чем значение PRx.

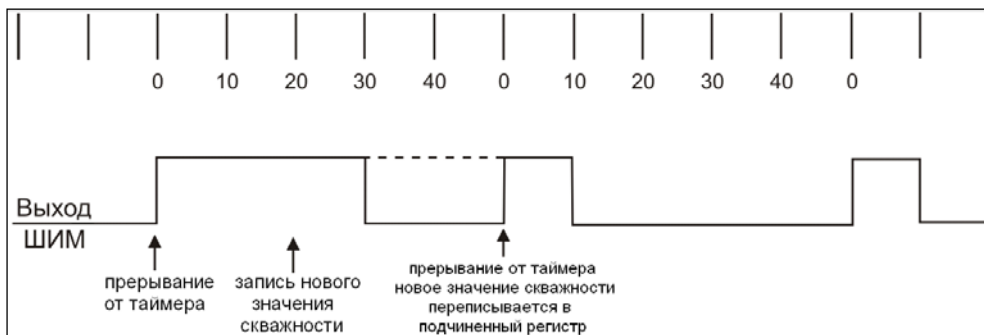


Рис. 30 Выход ШИМ

Регистры скважности для всех ШИМ имеют двойную буферизацию. При записи в регистры значение сохраняется в основных защелках, а при переполнении опорного таймера и начале нового периода ШИМ, значения из основной защелки переписываются в регистры подчиненной защелки, и вывод ШИМ переходит в высокий уровень.

Для регистров PW1DCH, PW1DCL, PW2DCH, PW2DCL, PW3DCH и PW3DCL операция записи записывает в «основные защелки», в то время как операция чтения считывает «подчиненные защелки». Поэтому желательно избегать операций типа чтение-модификация-запись с регистрами скважности.

Модули ШИМ используют прерывания от «таймера 1» и/или «таймера 2» (флаги запроса прерывания соответственно TMR1IF и TMR2IF). Прерывание от таймера генерируется, когда значение регистра таймера совпадет со значением регистра периода и во время следующего приращения сбросится в 0. Это прерывание также отмечает начало цикла ШИМ. В этот момент можно записать новые значения скважности. Флаги запроса прерывания должны быть сброшены программно.

ШИМ может работать с внешним генератором тактовых импульсов, но при этом необходимо учесть ряд особенностей. Так как внешний тактовый сигнал с входа PB4/TCLK12 синхронизируется с внутренним (выбирается один раз за командный цикл), то задержка между изменением сигнала TCLK12 и увеличением таймера, будет изменяться в пределах  $T_{су}$  (одного командного цикла). Это вызовет дрожание скважности и ошибку в периоде ШИМ. Дрожание будет приблизительно  $+1T_{су}$ , если внешний генератор не синхронизирован с генератором процессора. При использовании внешнего источника тактовых импульсов для ШИМ, его частота должна быть много меньше, чем тактовая частота микроконтроллера ( $F_c$ ).

Использование внешнего генератора тактовых импульсов для тактирования опорного таймера ШИМ (таймер 1 или 2) ограничивает максимальное разрешение ШИМ до 8 бит. Биты  $PWxDCL<7:6>$  должны быть сброшены. Использование любого другого значения вызовет искажение выхода ШИМ. Максимально достижимая частота ШИМ также более низкая. Максимальная частота ШИМ, когда источником тактовых импульсов является вывод PB4/TCLK12, показана в Таблица 32 (режим стандартного разрешения).



**Таблица 32**

Частота ШИМ (при тактовой частоте 33МГц)

Частота ШИМ	32.2 КГц	64.5 КГц	90.66 КГц	128.9 КГц	515.6 КГц
Значение PR	0xFF	0x7F	0x5A	0x3F	0x0F
Высокое разрешение	10 бит	9 бит	8.5 бит	8 бит	6 бит
Стандартное разрешение	8 бит	7 бит	6.5 бит	6 бит	4 бит

### «Таймер 3»

«Таймер 3» является 16-ти разрядным таймером, состоящим из регистров TMR3H (старший байт) и TMR3L (младший байт). Таймер имеет 16-ти разрядный регистр периода, который может также быть 16-ти разрядным регистром захвата (регистрации событий) PR3H/CA1H:PR3L/CA1L. Таймер может тактироваться внутренними импульсами  $F_c/4$  (если бит TMR3CS(TCON1<2>)=0), или внешними – задним фронтом (спадом) сигнала на выводе PB5/TCLK3 (если TMR3CS=1). Для работы таймера должен быть установлен бит TMR3ON.

При внешнем тактировании таймера, сигнал с вывода PB5/TCLK3 синхронизируется внутренними тактовыми импульсами дважды во время каждого цикла команды. Это вызывает задержку от момента появления заднего фронта на TCLK3 до фактического приращения таймера. Рис. 33 показывает временную диаграмму при работе таймера от внешнего генератора тактовых импульсов.

«Таймер 3» является 16-ти разрядным, поэтому необходимо осторожно считывать или записывать его во время работы, так как эти операции 8-ми разрядные. Лучше остановить таймер на время выполнения считывания/записи, а затем вновь запустить (используя бит TMR3ON). Однако, если нет возможности остановить таймер, можно использовать пример 13 для записи и пример 14 для считывания (во время данного процесса прерывания должны быть запрещены).

Таймер может работать в двух режимах (зависит от состояния бита CA1/PR3(TCON2<3>)):

- Режим трех входов захвата, таймер работает с регистром периода
- Режим четырех входов захвата.

Всего имеется до четырех 16-ти разрядных регистров захвата, которые захватывают 16-ти разрядное значение таймера, при фиксации события на выводах захвата. Есть четыре входа захвата PB0/CAP1, PB1/CAP2, PG4/CAP3 и PE3/CAP4, по одному для каждой пары регистров захвата. Событиями захвата могут быть (определяется битами CAxED1 и CAxED0):

- Передний фронт сигнала на входе захвата.
- Задний фронт (спад).
- Каждый четвертый передний фронт.
- Каждый шестнадцатый передний фронт.

Каждый 16-ти разрядный регистр захвата имеет связанный с ним флаг запроса прерывания, который устанавливается, когда происходит захват. Модули захвата являются частью блока «таймера 3».

### Режим трех входов захвата и регистра периода для таймера

Этот режим выбирается, если управляющий бит CA1/PR3=0. В режиме трех входов захвата регистры PR3H/CA1H и PR3L/CA1L составляют 16-ти разрядный регистр периода, и соответственно «захват 1» отключен и бит прерывания CA1IF никогда не устанавливается. Остальные входы захватов работают. Блок-схема показана на Рис. 31. Таймер инкрементируется до тех пор, пока не сравняется с значением регистра периода, а затем сбрасывается в 0000h в следующем цикле прерывания. При этом устанавливается флаг запроса прерывания от таймера (TMR3IF). Это прерывание может быть запрещено сбросом бита разрешения прерываний (TMR3IE). Флаг TMR3IF должен быть сброшен программно.

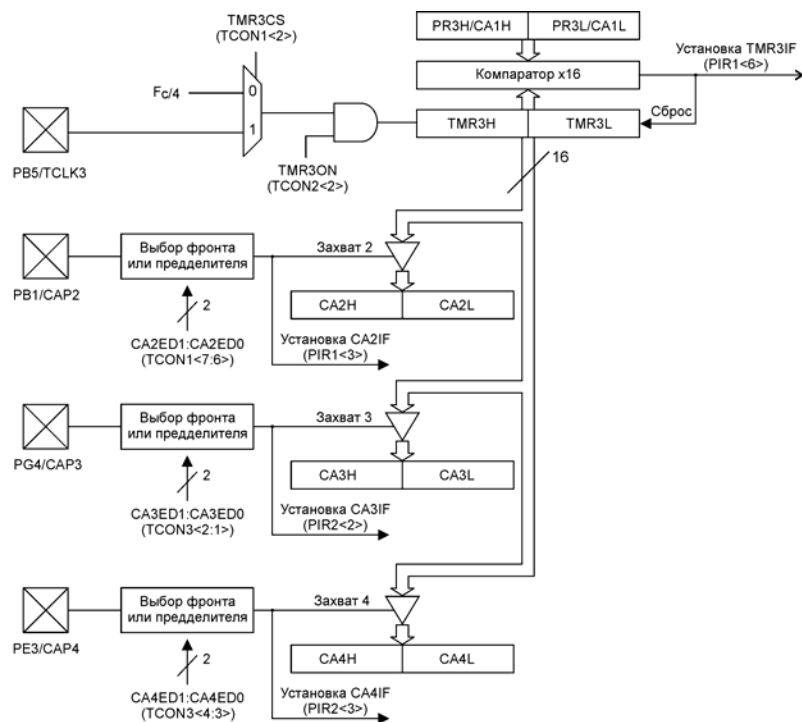
Биты CAxED1 и CAxED0 определяют событие, по которому произойдет захват. Когда происходит захват, устанавливается флаг запроса прерывания (CAxIF). Прерывание будет разрешено, если бит маски CAxIE установлен, бит разрешения прерывания от периферийного устройства (PEIE) установлен, а бит глобального запрета прерываний (GLINTD) должен быть сброшен. Флаг запроса прерывания CAxIF сбрасывается программно.

При изменении выбранного предварительного делителя, содержимое делителя частоты не сбрасывается. Поэтому, первый захват после подобного изменения будет неопределенным. Однако последующие захваты будут верными. Предварительный делитель частоты сбрасывается сигналом «сброс».

При использовании вывода захвата CAPx в качестве выхода порта режим захвата не отключается. Можно просто отключить прерывание от захвата посредством сброса CAxIE. Если вывод CAPx используется как выход, то можно активировать захват записью в порт. Это может быть полезным во время разработки для эмуляции прерывания от захвата.

Сигнал на входе захвата синхронизируется с внутренними тактовыми импульсами. Это накладывает определенные ограничения на форму входного сигнала.

Флаг переполнения имеет двойную буферизацию. Основной флаг устанавливается если одно захваченное значение уже находится в регистре захвата CAxH:CAxL, и произошло другое событие захвата на выводе CAPx. Новое значение не будет записываться в регистр захвата, защищая предыдущее нечитанное значение. При считывании старшего и младшего байта регистра захвата (в любом порядке), основной флаг переполнения передается в подчиненный флаг переполнения CAxOVF, и затем основной флаг сбрасывается. После этого можно считать регистр TCONx для определения значения CAxOVF. Рекомендуемая последовательность для считывания регистров захвата и флагов переполнения захвата показана в Пример 12.



**Рис. 31** Блок-схема «таймера 3»  
в режиме с тремя входами захвата и регистром периода

### Пример 12

Последовательность для чтения регистров захвата

```

MOVLB 3 ; выбрать банк 3
MOVFP CA2L, LO_BYTE ; считать младший байт регистра «захвата 2»
MOVFP CA2H, HI_BYTE ; считать старший байт регистра «захвата 2»
MOVFP TCON2, STAT_VAL ; считать регистр TCON2
    
```

### Пример 13

Запись в «таймер 3»

```

BSF CPUSTA, GLINTD ; отключить прерывания
MOVFP RAM_L, TMR3L
MOVFP RAM_H, TMR3H
BCF CPUSTA, GLINTD ; сделано, активировать прерывания
    
```

### Пример 14

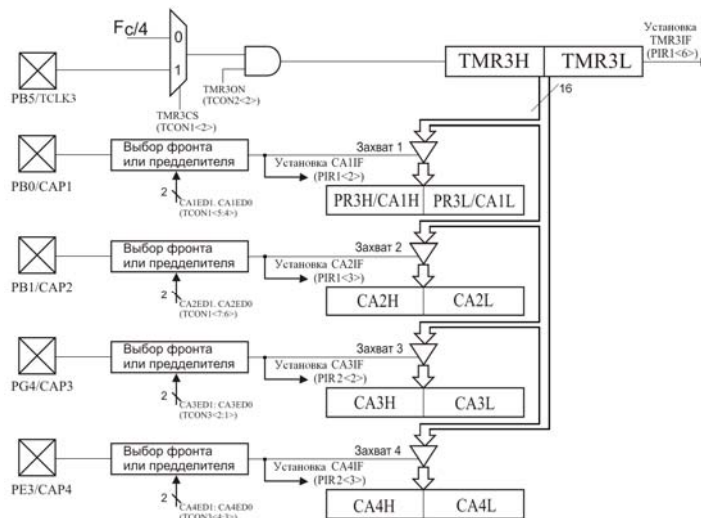
Считывание из «таймера 3»

```

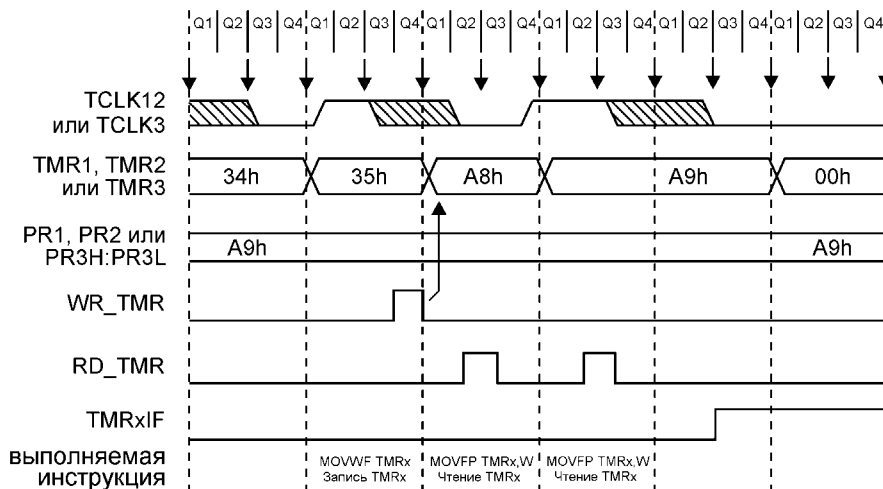
MOVFP TMR3L, TMPLO ; считывать младший байт TMR3
MOVFP TMR3H, TMPHI ; считывать старший байт TMR3
MOVFP TMPLO, WREG ; младший байт в WREG
CPFSLT TMR3L ; сравнить TMR3L < WREG
RETURN ; возврат
MOVFP TMR3L, TMPLO ; считать младший байт TMR3
MOVFP TMR3H, TMPHI ; считать старший байт TMR3
RETURN ; возврат
    
```

**Режим четырех входов захвата**

Этот режим выбирается если бит CA1/PR3=1. Блок-схема показана на Рис. 32. В данном режиме таймер работает без регистра периода, инкрементируясь от 0000h до FFFFh, и затем сбрасываясь в 0000h (с установкой флага запроса прерывания TMR3IF). Бит TMR3IF должен быть сброшен программно. Регистры PR3H/CA1H и PR3L/CA1L образуют 16-ти разрядный регистр «захвата 1». Соответствующий ему вход захвата - вывод PB0/CAP1. Режим захвата задается битами CA1ED1 и CA1ED0. Все захваты работают аналогично (см. Рис. 32, раздел «Режим трех входов захвата и регистра периода для таймера»).



**Рис. 32** Блок-схема «таймера 3» в режиме с четырьмя входами захвата



**Рис. 33** Работа таймеров 1, 2 и 3 от внешнего сигнала синхронизации

**Примечания:**

TCLK12 (TCLK3) выбирается в Q2 и Q4, «стрелка вниз» обозначает точку выборки. Задержка от спада сигнала на TCLK12 (TCLK3) до приращения таймера от 2\*Тс до 6\*Тс.

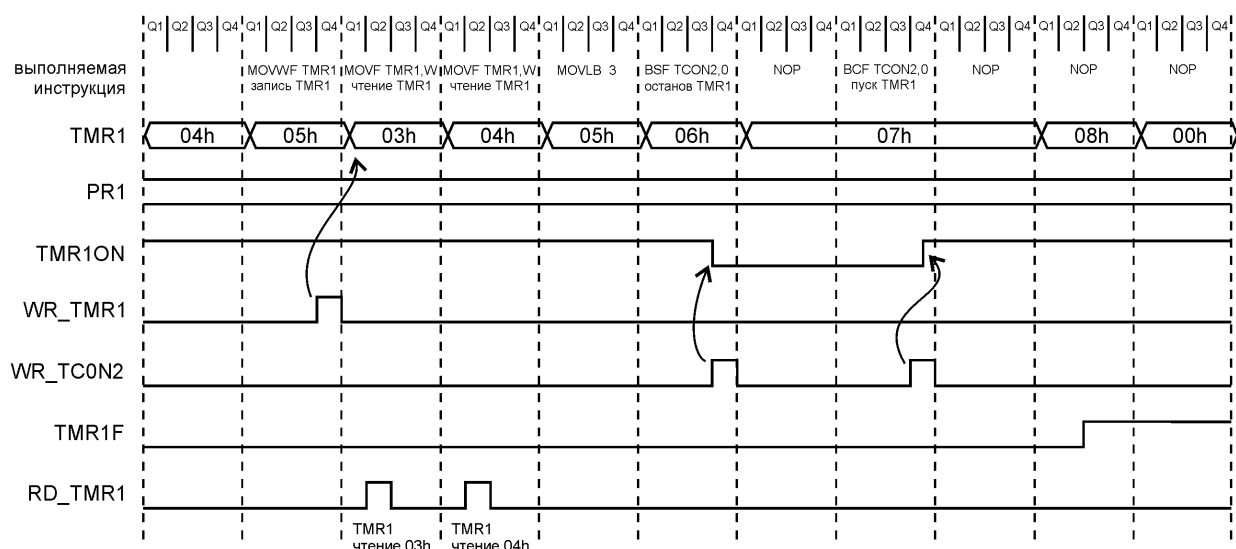


Рис. 34 Работа таймеров 1, 2 и 3 от внутреннего сигнала синхронизации

## Модули универсальных синхронно-асинхронных приемопередатчиков

Микроконтроллер содержит два идентичных модуля синхронно-асинхронных приемопередатчиков USART1 и USART2. Для описания работы этих модулей будут использованы общие названия регистров и выводов, т.к. они имеют одинаковое назначение в обоих модулях (см. Таблица 33).

Таблица 33

	Используемое название	USART1	USART2
<b>Регистры</b>	RCSTA	RCSTA1	RCSTA2
	TXSTA	TXSTA1	TXSTA2
	SPBRG	SPBRG1	SPBRG2
	RCREG	RCREG1	RCREG2
	TXREG	TXREG1	TXREG2
<b>Биты прерывания</b>	RCIE	RC1IE	RC2IE
	RCIF	RC1IF	RC2IF
	TXIE	TX1IE	TX2IE
	TXIF	TX1IF	TX2IF
<b>Выводы</b>	RX/DT	RA4/RX1/DT1	RG6/RX2/DT2
	TX/CK	RA5/TX1/CK1	RG7/TX2/CK2

Универсальный синхронно-асинхронный приемопередатчик может работать в следующих режимах:

- асинхронный (полный дуплекс),
- синхронный ведущий (полудуплекс),
- синхронный ведомый (полудуплекс).

Бит SPEN (RCSTA<7>) должен быть установлен, чтобы выводы RX/DT и TX/CK сконфигурировались как выводы последовательного интерфейса. Модуль USART будет управлять направлением выводов RX/DT и TX/CK, в зависимости от

состояния битов конфигурации в регистрах RCSTA и TXSTA. Следующие биты контролируют направление выводов: SPEN, TXEN, SREN, CREN, CSRC.

**Таблица 34**

Регистры TXSTA1 (адрес: 15h, банк 0) и TXSTA2 (адрес: 15h, банк 4)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	-	-	TRMT	TX9D
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7</b>		<b>CSRC:</b> выбор источника тактовых импульсов <u>Синхронный режим:</u> 1 = режим ведущего (внутренний тактовый сигнал из BRG) 0 = режим ведомого (от внешнего источника тактового сигнала) <u>Асинхронный режим:</u> Не имеет значения					
<b>бит 6</b>		<b>TX9:</b> выбор 9-ти разрядной передачи. 1 = выбирает 9-ти разрядную передачу 0 = выбирает 8-ми разрядную передачу					
<b>бит 5</b>		<b>TXEN:</b> разрешение передачи 1 = передача разрешена 0 = передача отключена Бит SREN/CREN блокирует TXEN в синхронном режиме.					
<b>бит 4</b>		<b>SYNC:</b> бит выбора режима USART (синхронный/асинхронный) 1 = синхронный режим 0 = асинхронный режим					
<b>бит 3,2</b>		Не реализованы, читаются как «0»					
<b>бит 1</b>		<b>TRMT:</b> флаг заполненности сдвигового регистра передатчика (TSR) 1 = регистр пуст 0 = регистр заполнен					
<b>бит 0</b>		<b>TX9D:</b> 9-й бит передаваемых данных (может использоваться для программной реализации проверки четности)					

Обозначения:

R = бит для чтения; W = бит с возможностью записи; U = бит не реализован, читается как 0;

-n = значение бита после сброса по включению питания: 1 = установлен; 0 = сброшен; x = значение не известно.

**Таблица 35**

Регистры RCSTA1 (адрес: 13h, банк 0) и RCSTA2 (адрес: 13h, банк 4)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R-0	R-0	R-0
SPEN	RX9	SREN	CREN	-	FERR	OERR	RX9D
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7</b>		<b>SPEN:</b> бит разрешения работы последовательного порта 1 = конфигурирует TX/CK и RX/DT как выводы последовательного порта USART 0 = последовательный порт отключен					

<b>бит 6</b>	<b>RX9:</b> выбор 9-ти разрядного приема 1 = выбирает 9-ти разрядный прием 0 = выбирает 8-ми разрядный прием
<b>бит 5</b>	<b>SREN:</b> бит разрешения однократного приема. Этот бит разрешает прием одного байта и после его приема автоматически сбрасывается. Синхронный режим: 1 = разрешить прием 0 = запретить прием Примечание: бит игнорируется в синхронном режиме ведомого. Асинхронный режим: Не имеет значения
<b>бит 4</b>	<b>CREN:</b> бит разрешения продолжительного приема. Этот бит разрешает непрерывный прием последовательно передаваемых данных. Асинхронный режим: 1 = разрешает непрерывный прием 0 = запрещает непрерывный прием Синхронный режим: 1 = разрешает непрерывный прием до момента сброса CREN (CREN отменяет SREN) 0 = отключает непрерывный прием
<b>бит 3</b>	Не реализовано, читается как «0»
<b>бит 2</b>	<b>FERR:</b> бит ошибки кадрирования 1 = есть ошибка (сбрасывается при чтении регистра RCREG) 0 = нет ошибки
<b>бит 1</b>	<b>OERR:</b> бит ошибки переполнения внутреннего буфера. 1 = есть ошибка (сбрасывается при сбросе бита CREN) 0 = нет ошибки
<b>бит 0</b>	<b>RX9D:</b> 9-й бит принятых данных (может использоваться для программной реализации проверки четности)

Обозначения:

R = бит для чтения; W = бит с возможностью записи; U = бит не реализован, читается как 0;

-n = значение бита после сброса по включению питания: 1 = установлен; 0 = сброшен; x = значение не известно.

### Генератор скорости передачи данных

Генератор скорости передачи данных (BRG) поддерживает асинхронный и синхронный режимы USART. Это отдельный 8-ми разрядный таймер/счетчик. Его период задается значением в регистре SPBRG. Скорость передачи рассчитывается по следующей формуле:

- скорость передачи для асинхронного режима =  $F_c / (64 * ((SPBRG) + 1))$ ,
- скорость передачи для синхронного режима =  $F_c / (4 * ((SPBRG) + 1))$ , значение (SPBRG) от 0 до 255.

Запись нового значения в SPBRG приводит к сбросу таймера BRG. Это обеспечивает то, что генератор сразу переключается на новую скорость передачи. После сигнала «сброс» регистр SPBRG очищается, поэтому его необходимо загружать требуемым значением после каждого сброса.

### Асинхронный режим

В этом режиме USART использует стандартный формат NRZ (один стартовый бит, восемь или девять информационных битов и один стоповый бит). Самый распространенный формат данных – это 8-ми битный. Внутрипроцессорный генератор скорости передачи может использоваться для получения стандартных частот скорости передачи. Приемник и передатчик USART являются функционально независимыми, но используют одинаковый формат данных и скорость передачи. Проверка четности не поддерживается аппаратными средствами, но может быть реализована программно (используя девятый бит данных). Модуль USART в асинхронном режиме останавливается во время SLEEP (в режиме ожидания). Асинхронный режим выбирается сбросом бита SYNC (TXSTA<4>).

Модуль USART в асинхронном режиме состоит из следующих компонентов:

- Генератор скорости передачи
- Схема выборки
- Асинхронный приемник
- Асинхронный передатчик

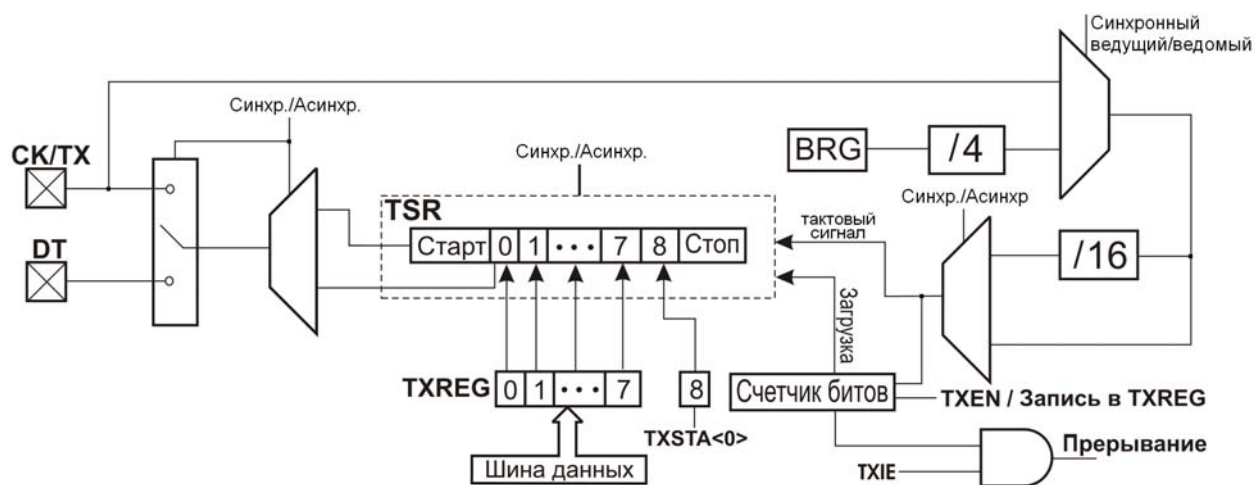
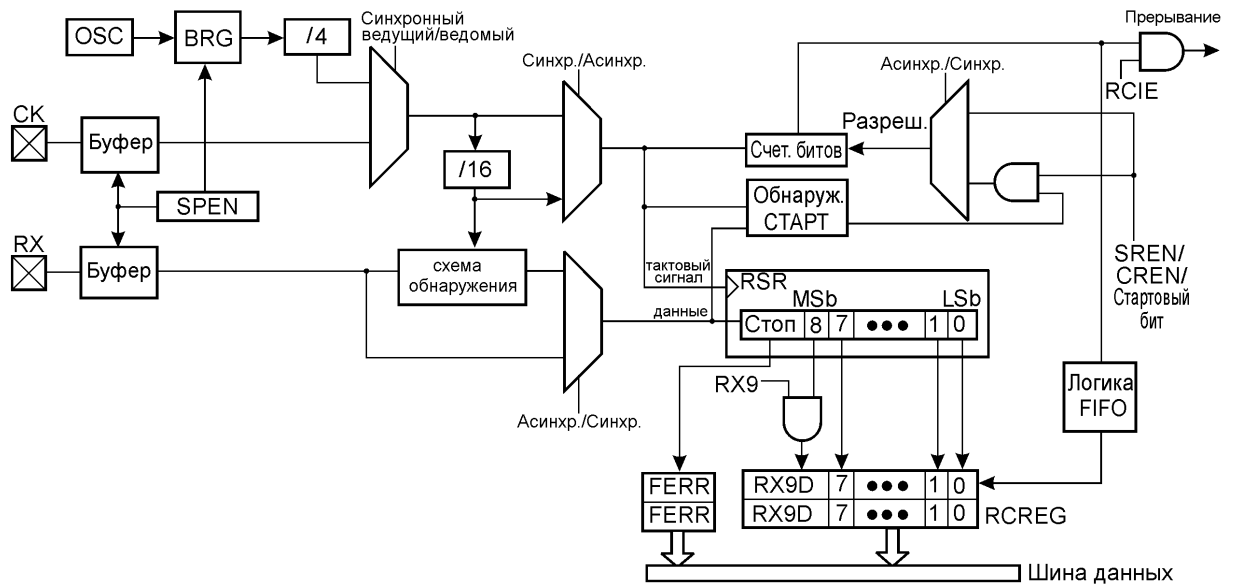


Рис. 35 Блок-схема передатчика





**Рис. 36** Блок-схема приемника

### Асинхронный передатчик

Блок-схема передатчика представлена на Рис. 35. Основой передатчика является сдвиговый регистр передачи (TSR). Он получает информацию из буфера передатчика (TXREG). В буфер TXREG данные загружаются программно. Сдвиговый регистр TSR загружается новыми данными из TXREG (если они есть) как только передастся стоповый бит предыдущей посылки данных. Это происходит в последнем командном цикле периода BRG. При этом устанавливается флаг запроса прерывания TXIF индицирующий, что TXREG пуст. Это прерывание может быть разрешено или запрещено соответственно установкой или сбросом бита TXIE. Флаг запроса прерывания TXIF устанавливается независимо от значения TXIE. Он не может быть сброшен программно. Флаг сбрасывается аппаратно при загрузке новых данных в TXREG. Бит TRMT (TXSTA<1>) индицирует состояние сдвигового регистра TSR. Бит устанавливается когда TSR пуст. TRMTx доступен только для чтения и не может вызывать прерывания. Регистр TSR не отображается в памяти данных, т.е. не доступен для чтения/записи.

Передача разрешается, когда устанавливается бит TXEN (TXSTA<5>). Фактически передача не начнется: пока данные не будут загружены в TXREG и генератор скорости передачи не выдаст сдвиговый синхрои импульс (см. Рис. 37). Передачу также можно начать сначала загрузив TXREG, а потом установив бит TXEN. Обычно, если передача начинается в первый раз, TSR пуст, поэтому запись данных в TXREG повлечет немедленную их передачу в TSR, освободив TXREG. Поэтому возможна неразрывная последовательная передача (см. Рис. 38). Сброс TXEN во время передачи вызовет отмену передачи, сброс передатчика.

Для выбора 9-ти битной передачи, бит TX9 (TXSTA<6>) должен быть установлен в единицу. Девятое значение бита записывается в TX9D (TXSTA<0>) перед записью 8-ми битных данных в TXREG, так как запись данных в TXREG может вызвать немедленную передачу данных в TSR (если TSR пуст).

Шаги для настройки асинхронной передачи следующие:

1. записать значение в регистр SPBRG для задания скорости передачи.
2. включить асинхронный последовательный порт (бит SYNC=0 и бит SPEN=1).
3. если требуются прерывания, тогда установите бит TXIE.
4. если требуется 9-ти битная передача, тогда установите бит TX9.
5. если выбрана 9-ти битная передача, девятый бит должен быть загружен в TX9D.
6. загрузите данные в регистр TXREG.
7. запустите передачу установкой бита TXEN.

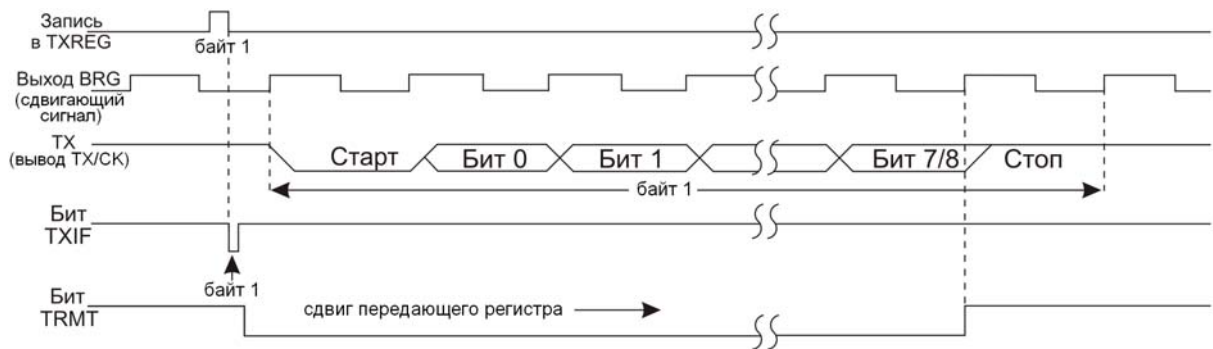


Рис. 37 Асинхронная передача

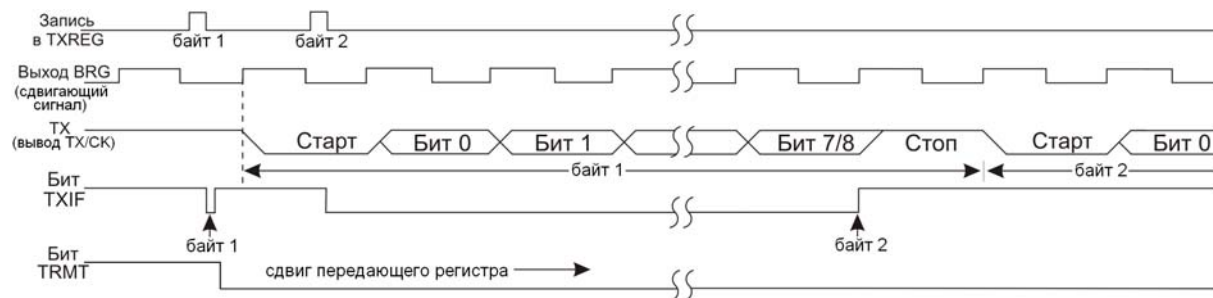


Рис. 38 Асинхронная неразрывная последовательная передача

### Асинхронный приемник

Блок-схема приемника показана на Рис. 36. Данные с вывода RX/DT подаются в блок восстановления данных. Это высокоскоростной сдвиговый регистр, работающий на частоте в 16 раз выше скорости передачи, в то время как основной сдвиговый регистр принимаемых данных работает со скоростью передачи.

Если выбран асинхронный режим, то приемник включается установкой бита CREN (RCSTA<4>).

Основой приемника является сдвиговый регистр приема (RSR). После захвата стопового бита, принятые данные из RSR передаются в RCREG (если он пуст), после чего устанавливается флаг запроса прерывания RCIF. Прерывание может быть разрешено или запрещено соответственно установкой или сбросом бита RCIE. RCIF доступен только для чтения, он сбрасывается аппаратно когда считываются данные из RCREG и регистр пуст. Регистр RCREG имеет двойную буферизацию, т.е. можно принять два байта данных в RCREG FIFO и третий байт

начать принимать в RSR. При обнаружении стопового бита третьего байта, если RCREG по-прежнему не считан, устанавливается бит ошибки переполнения приемника OERR (RCSTA<1>). Данные в RSR будут потеряны. Для извлечения двух байт RCREG должен считываться дважды. Бит OERR должен быть очищен программно сбросом приема (сбросом бита CREN). Пока бит OERR установлен, приемник не работает. Флаг ошибки кадрирования FERR (RCSTA<2>) устанавливается, если невозможно обнаружить стоповый бит. Флаг FERR и девятый бит данных буферизуются также как и принятые данные. Поэтому необходимо считать регистр RCSTA перед считыванием RCREG, чтобы не потерять прежнюю информацию FERR и RX9D.

Данные с вывода RX/DT сканируются, три раза в каждом такте приема, мажоритарной схемой обнаружения, для выявления уровня сигнала на выводе RX/DT. Сканирование осуществляется на седьмом, восьмом и девятом заднем фронте (спаде) импульсов частотой в 16 раз превышающей частоту приема (см. Рис. 39).

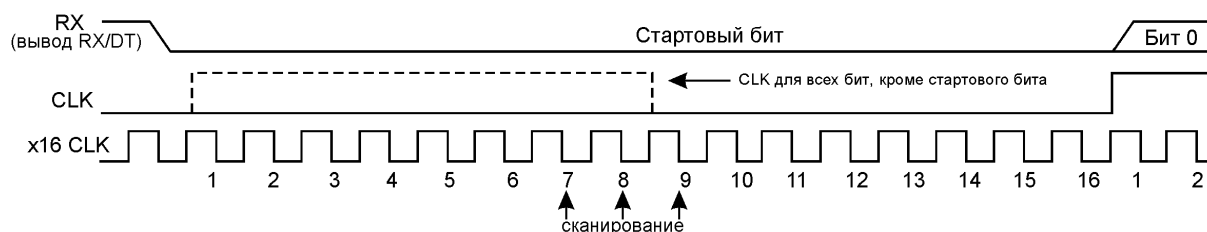


Рис. 39 Схема сканирования вывода RX/DT

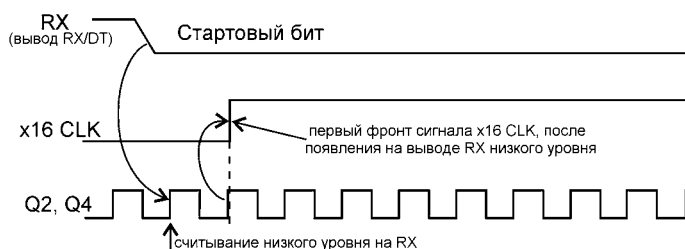


Рис. 40 Обнаружение стартового бита

Шаги для настройки асинхронного приема следующие:

1. Записать значение в регистр SPBRG для задания скорости приема.
2. Включить асинхронный последовательный порт (бит SYNC=0 и бит SPEN=1).
3. Если требуются прерывания, тогда установите бит RCIE.
4. Если требуется 9-ти битный прием, тогда установите бит RX9.
5. Разрешите прием установкой бита CREN.
6. При завершении приема установится бит RCIF, и произойдет прерывание (если установлен бит RCIE).
7. Считайте RCSTA для получения значения девятого бита (для 9-ти битного приема) и бит FERR для определения ошибки.
8. Считайте 8-ми битные принятые данные из регистра RCREG.
9. Если произошла ошибка переполнения, сбросьте бит OERR.

Для отмены приема, сбросьте либо биты SREN и CREN, либо бит SPEN. Это сбросит логику приема, но не изменит настройки.

## Синхронный ведущий режим

В синхронном ведущем режиме данные передаются полудуплексным способом, то есть прием и передача происходят не одновременно: при передаче данных прием запрещен, и наоборот. Синхронный режим включается при установке бита SYNC (TXSTA<4>). Бит SPEN (RCSTA<7>) устанавливается, чтобы сконфигурировать выводы: СК - линия тактовых импульсов и DT - линия данных. Ведущий режим означает, что процессор формирует тактовые импульсы на линии СК. Ведущий режим выбирается установкой бита CSRC (TXSTA<7>).

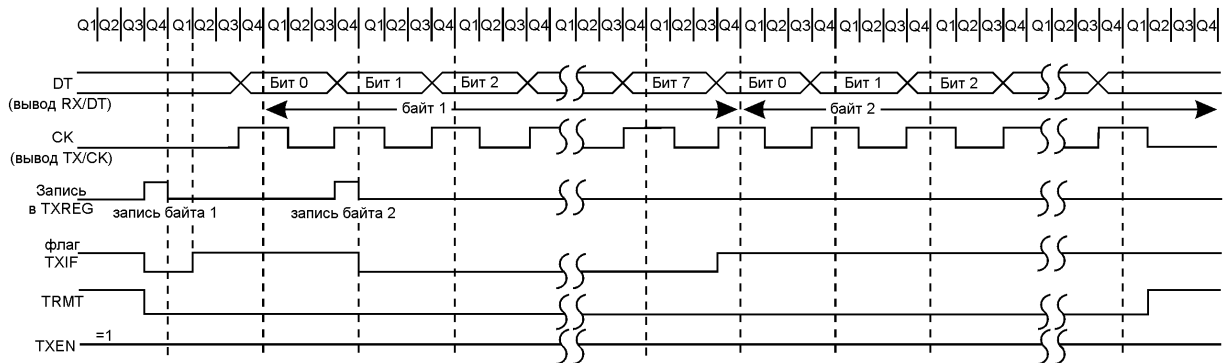


Рис. 41 Синхронная передача в ведущем режиме

## Передача данных в синхронном ведущем режиме

Блок-схема передатчика показана на Рис. 35. Основой передатчика является сдвиговый регистр передачи (TSR). Сдвиговый регистр получает данные из буфера передатчика TXREG. Данные загружаются в TXREG программно. После передачи последнего бита предыдущей посылки, TSR загружается новыми данными из TXREG (если они есть). Это происходит в последнем командном цикле периода BRG. При этом устанавливается флаг запроса прерывания TXIF индицирующий, что TXREG пуст. Это прерывание может быть разрешено или запрещено соответственно установкой или сбросом бита TXIE. Флаг запроса прерывания TXIF устанавливается независимо от значения TXIE. Он не может быть сброшен программно. Флаг сбрасывается аппаратно при загрузке новых данных в TXREG. Бит TRMT (TXSTA<1>) индицирует состояние сдвигового регистра TSR. Бит устанавливается когда TSR пуст. TRMT доступен только для чтения и не может вызывать прерывания. Регистр TSR не отображается в памяти данных, т.е. не доступен для чтения/записи.

Передача разрешается, при установке бита TXEN (TXSTA<5>). Фактически передача не начнется пока данные не будут загружены в TXREG. Первый бит данных появится на первом переднем фронте тактовых импульсов с вывода TX/СК. Данные стабилизируются к заднему фронту тактовых импульсов (см. Рис. 41). Передачу также можно начать сначала загрузив TXREG, а потом установив бит TXEN. Это удобно, когда выбраны низкие скорости передачи. Генератор BRG остановлен, когда биты TXEN, CREN, SREN сброшены. Установка бита TXEN запустит генератор BRG, который сразу же выдаст сдвиговые тактовые импульсы. Обычно, при первом разрешении передачи регистр TSR пуст, поэтому записанные в TXREG данные будут сразу переданы в TSR, опустошая TXREG. Поэтому возможна неразрывная последовательная передача данных. Сброс TXEN во

время передачи вызовет отмену передачи, сброс передатчика. Если во время передачи будут установлены биты CREN и SREN, передача будет отменена, и вывод RX/DT вернется в третье состояние (для приема). Вывод TX/CK останется выходом, если установлен бит CSRC (внутренний источник тактовых импульсов от BRG). Логика передатчика не сбрасывается, хотя и отсоединяется от выводов. Чтобы сбросить передатчик - необходимо сбросить бит TXEN. Если бит SREN установлен для прерывания осуществляемой передачи и получения одного байта, то после получения одного байта SREN сбросится и последовательный порт снова вернется к передаче, так как бит TXEN попрежнему установлен. Линия DT сразу же будет переключена в режим выхода. Для избежания этого, TXEN должен быть сброшен.

Чтобы выбрать 9-ти битную передачу, необходимо установить бит TX9 (TXSTA<6>). Девятый бит должен записываться (в TX9D (TXSTA<0>)) до записи 8-ми битных данных в TXREG, так как запись данных в TXREG может вызвать немедленную передачу данных в TSR (если TSR пуст).

Шаги для настройки передачи в синхронном ведущем режиме:

1. Записать значение в регистр SPBRG для задания скорости передачи.
2. Включить синхронный последовательный порт в ведущем режиме (биты SYNC=1, SPEN=1 и CSRC=1).
3. Проверить что биты CREN и SREN сброшены, если эти биты установлены, то они отменяют передачу.
4. Если требуются прерывания, тогда установите бит TXIE.
5. Если требуется 9-ти битная передача, тогда установите бит TX9.
6. Если выбрана 9-ти битная передача, девятый бит должен быть загружен в TX9D.
7. Загрузите данные в регистр TXREG.
8. Запустите передачу установкой бита TXEN.

Для отмены передачи необходимо сбросить бит SPEN или бит TXEN. Это сбросит логику передачи, но сохранит настройки, когда передача возобновится.

### **Прием данных в синхронном ведущем режиме**

Если выбран синхронный режим, прием разрешается установкой бита SREN (RCSTA<5>) или бита CREN (RCSTA<4>). Данные с вывода RX/DT опрашиваются на заднем фронте (спаде) тактовых импульсов. Если установлен SREN, то принимается только один байт. Если установлен CREN, то прием продолжается пока CREN не сбросится. Если установлены оба бита, то CREN имеет приоритет. После приема последнего бита полученные данные из сдвигового регистра приема (RSR) передаются в RCREG (если он пуст), и устанавливается флаг запроса прерывания RCIF. Прерывание может быть разрешено или запрещено соответственно установкой или сбросом бита RCIE. RCIF доступен только для чтения, и сбрасывается аппаратно, когда RCREG считан и пуст.

Регистр RCREG имеет двойную буферизацию, т.е. можно принять два байта данных в RCREG FIFO и третий байт начать принимать в RSR. При приеме последнего бита третьего байта, если RCREG по-прежнему не считан, устанавливается бит ошибки переполнения приемника OERR (RCSTA<1>). Данные в RSR будут потеряны. Для извлечения двух байт RCREG должен считываться дважды. Бит OERR должен быть очищен программно сбросом

приема (сбросом бита CREN). Пока бит OERR установлен, приемник не работает. Девятый бит данных буферизуется также как и принятые данные. Поэтому необходимо считать регистр RCSTA перед считыванием RCREG, чтобы не потерять прежние значение RX9D.

Шаги для настройки приема в синхронном ведущем режиме:

1. записать значение в регистр SPBRG для задания скорости приема.
2. включить синхронный последовательный порт в ведущем режиме (биты SYNC=1, SPEN=1 и CSRC=1).
3. если требуются прерывания, тогда установите бит RCIE.
4. если требуется 9-ти битный прием, тогда установите бит RX9.
5. если требуется прием единичного байта установите бит SREN, для продолжительного приема установите бит CREN.
6. при завершении приема установится бит RCIF, и произойдет прерывание (если установлен бит RCIE).
7. считайте RCSTA для получения значения девятого бита (для 9-ти битного приема) и бит определения ошибки.
8. считайте 8-ми битные принятые данные из регистра RCREG.
9. если произошла ошибка переполнения, сбросьте бит OERR.

Для отмены приема, сбросьте либо биты SREN и CREN, либо бит SPEN. Это сбросит логику приема, но не изменит настройки.

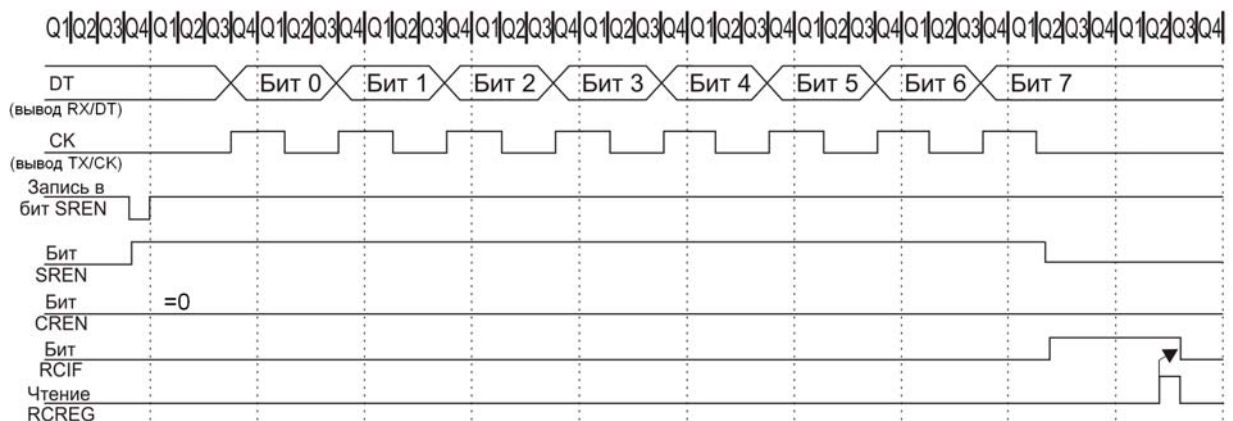


Рис. 42 Синхронный прием

### Синхронный ведомый режим

Синхронный ведомый режим отличается от ведущего тем, что тактовые импульсы подаются от внешнего источника. Это позволяет устройству передавать или получать данные в режиме SLEEP. Ведомый режим включается сбросом бита CSRC (TXSTA<7>).

### Передача данных в синхронном ведомом режиме

Работа ведущего и ведомого режимов идентична, за исключением работы в режиме SLEEP. Если 2 байта записываются в TXREG и затем выполняется команда SLEEP, то произойдет следующее. Первый байт сразу переносится в TSR и будет передаваться по тактовым импульсам. Второй байт останется в

TXREG. Флаг TXIF не будет установлен. Когда закончится передача первого байта, второй байт будет перенесен из TXREG в TSR и установится флаг TXIF. Если TXIE=1, прерывание выведет чип из режима SLEEP, и если разрешено периферийное прерывание, тогда программа переходит к вектору прерывания (0020h).

Шаги для настройки передачи в синхронном ведомом режиме:

1. Записать значение в регистр SPBRG для задания скорости передачи.
2. Включить синхронный последовательный порт в ведомом режиме (биты SYNC=1, SPEN=1 и CSRC=0).
3. Сбросить бит CREN.
4. Если требуются прерывания, тогда установите бит TXIE.
5. Если требуется 9-ти битная передача, тогда установите бит TX9.
6. Если выбрана 9-ти битная передача, девятый бит должен быть загружен в TX9D.
7. Загрузите данные в регистр TXREG.
8. Запустите передачу установкой бита TXEN.

Для отмены передачи необходимо сбросить бит SPEN или бит TXEN. Это сбросит логику передачи, но сохранит настройки, когда передача возобновится.

### **Прием данных в синхронном ведомом режиме**

Работа ведущего и ведомого режимов идентична, за исключением работы в режиме SLEEP. Также безразлично значение бита SREN.

Если прием разрешен (CREN=1) до команды SLEEP, то данные могут быть получены в режиме SLEEP. При завершении приема, данные из RSR передаются в RCREG и устанавливается флаг запроса прерывания RCIF, а если бит RCIE=1 прерывание выведет чип из режима SLEEP. Если разрешено периферийное прерывание, программа перейдет к вектору прерывания (0020h).

Шаги для настройки приема в синхронном ведомом режиме:

1. Записать значение в регистр SPBRG для задания скорости приема.
2. Включить синхронный последовательный порт в ведомом режиме (биты SYNC=1, SPEN=1 и CSRC=0).
3. Если требуются прерывания, тогда установите бит RCIE.
4. Если требуется 9-ти битный прием, тогда установите бит RX9.
5. Для разрешения приема установите бит CREN.
6. При завершении приема установится бит RCIF, и произойдет прерывание (если установлен бит RCIE).
7. Считайте RCSTA для получения значения девятого бита (для 9-ти битного приема) и бит определения ошибки.
8. Считайте 8-ми битные принятые данные из регистра RCREG.
9. Если произошла ошибка переполнения, сбросьте бит OERR.

Для отмены приема, сбросьте либо бит CREN, либо бит SPEN. Это сбросит логику приема, но не изменит настройки.

**Модуль ведущего синхронного последовательного порта (СПП)**

Модуль ведущего синхронного последовательного порта представляет собой последовательный интерфейс, применяемый для соединения с другими микроконтроллерами или периферийными микросхемами. Такими микросхемами могут быть последовательные EEPROM, сдвиговые регистры, драйверы дисплея, аналого-цифровые преобразователи и т. д. Модуль может функционировать в одном из двух режимов: интерфейс SPI и интерфейс I2C.

**Таблица 36**

Регистр SSPSTAT, статусный регистр СПП (адрес: 13h, банк 6)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	СКЕ	D/A	P	S	R/W	UA	BF
бит 7	6	5	4	3	2	1	бит 0
бит 7		<p><b>SMP: выбор режима сэмплирования.</b>  <u>Режим ведущего SPI:</u>                      1 = входные данные захватываются в конце периода выходной частоты синхронизации                      0 = входные данные захватываются в середине периода выходной частоты синхронизации  <u>Режим ведомого SPI: SMP должен быть сброшен в «0» во время использования SPI в режиме ведомого.</u>  <u>Режим I2C ведомого или ведущего:</u>                      1 = управление длительностью фронта выключено (в стандартном режиме 100 кГц и 1 МГц)                      0 = управление длительностью фронта включено (в скоростном режиме 400 кГц)</p>					
бит 6		<p><b>СКЕ: Выбор фронта тактового сигнала SPI</b> (см. Рис. 44, Рис. 45, Рис. 46).  <u>При СКР = 0: 1= Данные передаются по переднему фронту SCK</u>                      0 = Данные передаются по заднему фронту SCK  <u>При СКР = 1: 1= Данные передаются по переднему фронту SCK</u>                      0 = Данные передаются по заднему фронту SCK</p>					
бит 5		<p><b>D/A: указатель данные/адрес</b> (используется только в режиме I2C).                      1 = Последний принятый или переданный байт является данными                      0 = Последний принятый или переданный байт является адресным</p>					



бит 4	<b>P: флаг обнаружения состояния «СТОП»</b> (используется только в режиме I2C). Этот бит сброшен в «0», когда модуль СПП выключен, SSPEN =0. 1 = указывает, что состояние СТОП было обнаружено последним (по отношению к состоянию СТАРТ). 0 = состояние СТОП не обнаружено.
бит 3	<b>S: флаг обнаружения состояния «СТАРТ»</b> (используется только в режиме I2C). Этот бит сброшен в «0» когда модуль MSSP выключен, SSPEN =0. 1 = указывает, что состояние СТАРТ было обнаружено последним ( по отношению к состоянию СТОП). 0 = состояние СТАРТ не обнаружено.
бит 2	<b>R/W: указатель чтение/запись (используется только в режиме I2C).</b> Разряд сохраняет информацию о направлении обмена данными (запись/чтение) начиная с последнего момента совпадения адреса. Значение разряда действительно только после совпадения адреса и до приема следующих состояний СТАРТ, СТОП или ACK. В режиме ведомого I2C: 1 = Чтение данных 0 = Запись данных В режиме ведущего I2C: 1 = Передача данных выполняется 0 = Передачи данных не происходит Результат операции «логическое ИЛИ» этого бита с битами SEN, RSEN, PEN, RCEN, или ACKEN укажет на неактивное состояние модуля СПП.
бит 1	<b>UA: Обновление адреса</b> (используется только в режиме I2C с 10-ти разрядными адресами). 1 = Указывает на необходимость обновления адреса в регистре SSPADD 0 = Адрес не нужно обновлять
бит 0	<b>BF: флаг заполнения буфера</b> <u>Прием данных ( режимы SPI и I2C ):</u> 1 = Прием завершен, в SSPBUF находятся принятые данные 0 = Прием не завершен, SSPBUF пуст <u>Передача данных (только режим I2C):</u> 1 = Передача данных осуществляется, в SSPBUF находятся передающиеся данные 0 = Передача данных завершена, SSPBUF пуст

Обозначения:

R = бит для чтения; W = бит с возможностью записи; U = бит не реализован, читается как 0;

-n = значение бита после сброса по включению питания: 1 = установлен; 0 = сброшен; x = значение не известно.

**Таблица 37**

Регистр SSPCON1, регистр 1 управления модулем СПП (адрес: 11h, банк 6)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	СКР	SSPM3	SSPM2	SSPM1	SSPM0
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7</b>		<p><b>WCOL: Флаг обнаружения конфликта записи.</b>  <u>Режим ведущего:</u>                      1 = попытка записи в регистр SSPBUF в условиях не готовности блока I2C к передаче данных                      0 = Нет конфликта  <u>Режим ведомого:</u>                      1= попытка записи в SSPBUF в момент передачи ранее записанных данных (установка в «0» осуществляется программно)                      0 = Нет конфликта</p>					
<b>бит 6</b>		<p><b>SSPOV: Флаг обнаружения переполнения буфера при приеме данных.</b>  <u>В режиме SPI:</u>                      1 = прием нового байта произошел до момента, когда ранее полученные данные не были считаны из регистра SSPBUF. В случае переполнения данные в SSPBUF теряются. Переполнение может происходить только в режиме ведомого. Чтобы предотвратить установку флага в режиме ведомого, пользователь должен считывать SSPBUF, даже если предполагается только передача данных. В режиме ведущего бит переполнения не устанавливается, т.к. каждый новый прием и передача данных инициализируется записью в SSPBUF. Установка в «0» осуществляется программно.                      0 = переполнения нет  <u>В режиме I2C:</u>                      1 = прием нового байта произошел до момента, когда ранее полученные данные были считаны из регистра SSPBUF. Значение SSPOV не имеет смысла при передаче данных. Установка в «0» осуществляется программно.                      0 = переполнения нет</p>					
<b>бит 5</b>		<p><b>SSPEN: включение модуля СПП.</b>  <u>В режиме SPI:</u>                      1 = модуль СПП включен, выводы SCK, SDO, SDI и SS конфигурируются для использования модулем                      0 = модуль СПП выключен, выводы работают как порты ввода/вывода  <u>В режиме I2C:</u>                      1 = модуль СПП включен, выводы SDA и SCL конфигурируются для использования модулем                      0 = модуль СПП выключен, выводы работают как порты ввода/вывода  <u>Примечание: В режиме SPI выводы SDA и SCL работают как порты ввода/вывода.</u></p>					

бит 4	<p><b>СКР: Управление сигналом тактовой частоты.</b>  <u>В режиме SPI:</u>          1 = высокий уровень сигнала является не активным          0 = низкий уровень сигнала является не активным  <u>В режиме ведомого I2C: управление тактовым сигналом SCK:</u>          1 = тактовый сигнал разрешен          0 = тактовый сигнал удерживается в состоянии 0 (используется для подготовки данных).  <u>В режиме ведущего I2C:</u>          Не используется в этом режиме.</p>
бит 3, 2, 1, 0	<p><b>SSPM3--SSPM0: Выбор режима синхронного последовательного порта:</b>          0000 - SPI режим ведущего, тактовый сигнал = <math>F_c/4</math>          0001 - SPI режим ведущего, тактовый сигнал = <math>F_c/16</math>          0010 - SPI режим ведущего, тактовый сигнал = <math>F_c/64</math>          0011 - SPI режим ведущего, тактовый сигнал = «выход таймера 2»/2          0100 - SPI режим ведомого, тактовый сигнал с вывода SCK, разрешено управление выводом SS.          0101 - SPI режим ведомого, тактовый сигнал с вывода SCK, запрещено управление выводом SS, SS может использоваться как вывод порта входа/выхода          0110 = I2C режим ведомого, 7-разрядный адресный режим          0111 = I2C режим ведомого, 10-разрядный адресный режим          1000 = I2C режим ведущего, тактовый сигнал = <math>F_c/(4 * (SSPAD+1))</math>          1xx1 = не используемая конфигурация          1x1x = не используемая конфигурация</p>

Обозначения:

R = бит для чтения; W = бит с возможностью записи; U = бит не реализован, читается как 0;

-n = значение бита после сброса по включению питания: 1 = установлен; 0 = сброшен; x = значение не известно.

**Таблица 38**

Регистр SSPCON2, регистр 2 управления модулем СПП (адрес: 12h, банк 6)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
бит 7	6	5	4	3	2	1	бит 0
бит 7	<p><b>GCEN: Разрешение поддержки общего вызова.</b>            Используется только в режиме ведомого I2C.            1 = разрешить установку сигнала запроса на обработку прерывания при приеме в регистр SSPSR адреса общего вызова (0000h)            0 = установка сигнала запроса на обработку прерывания при приеме в регистр SSPSR адреса общего вызова (0000h) запрещена.</p>						

бит 6	<p><b>ACKSTAT: Флаг состояния получения подтверждения обмена данными.</b> Используется только в режиме ведущего I2C при передаче данных.          1 = подтверждение обмена данными не было получено от ведомого          0 = подтверждение было получено от ведомого</p>
бит 5	<p><b>ACKDT: бит подтверждения данных.</b> Используется только в режиме ведущего I2C при приеме данных. Значение этого разряда передается при формировании сигнала подтверждения о приеме данных.          1 = нет подтверждения          0 = есть подтверждение</p>
бит 4	<p><b>ACKEN: Разрешение передачи сигнала подтверждения.</b>          Используется только в режиме ведущего I2C при приеме данных.          1 = инициализировать передачу сигнала подтверждения (передается значение разряда ACKDT). Установка в «0» производится автоматически после передачи сигнала.          0 = передача сигнала подтверждения не инициализирована</p>
бит 3	<p><b>RCEN: Разрешение приема данных.</b>          Используется только в режиме ведущего I2C.          1 = разрешить прием данных          0 = приемник выключен</p>
бит 2	<p><b>PEN: Разрешение формирования состояния СТОП.</b>          Используется только в режиме ведущего I2C.          1 = инициализировать формирование состояния СТОП на выводах SDA и SCL. Аппаратно устанавливается в «0».          0 = формирование состояния СТОП не инициализировано</p>
бит 1	<p><b>RSEN: Разрешение формирования состояния повторный СТАРТ.</b>          Используется только в режиме ведущего I2C.          1 = инициализировать формирование состояния повторный СТАРТ на выводах SDA и SCL. Аппаратно устанавливается в «0».          0 = формирование состояния повторный СТАРТ не инициализировано</p>
бит 0	<p><b>SEN: Разрешение формирования состояния СТАРТ.</b>          Используется только в режиме ведущего I2C.          1 = инициализировать формирование состояния СТАРТ на выводах SDA и SCL. Аппаратно устанавливается в «0».          0 = формирование состояния СТАРТ не инициализировано</p>

Обозначения:

R = бит для чтения; W = бит с возможностью записи; U = бит не реализован, читается как 0;

-n = значение бита после сброса по включению питания: 1 = установлен; 0 = сброшен; x = значение не известно.

### **Режим SPI**

Режим SPI позволяет синхронно передавать и одновременно получать 8-разрядные данные. Поддерживаются 4 режима SPI. Для приема/передачи данных используются 3 вывода микроконтроллера: «выход последовательных данных» (SDO), «вход последовательных данных» (SDI) и «тактовый сигнал» (SCK). Для работы в режиме ведомого дополнительно может использоваться еще один вывод: «выбор ведомого» (SS).

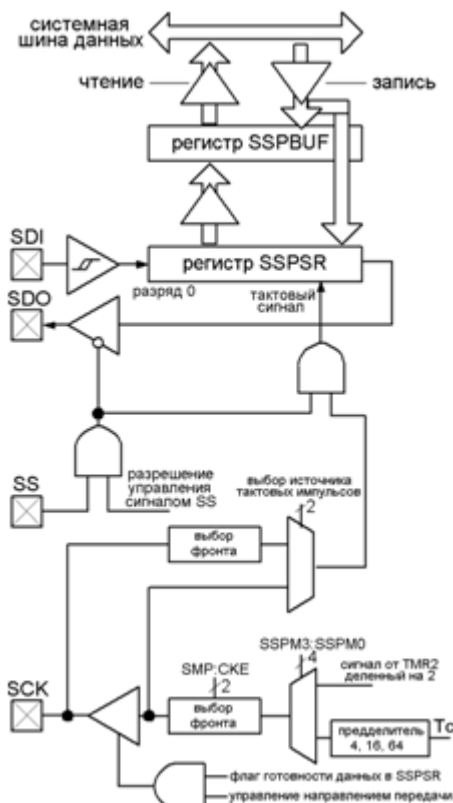
### **Функционирование модуля СПП в режиме SPI**

Для инициализации режима SPI необходимо установить несколько контрольных разрядов регистра SSPCON1 (SSPCON1<5:0>) и SSPSTAT<7:6>. Выбранная конфигурация определяет следующие варианты функционирования:

- Режим ведущего (SCK- выход)
- Режим ведомого (SCK - вход)
- Активный уровень тактового сигнала (активный уровень SCK)
- Фаза выборки входных данных
- Активный фронт тактового сигнала (передний/задний фронт SCK)
- Скорость передачи данных (только в режиме ведущего)
- Режим выбора ведомого (только в режиме ведомого)

Рис. 43 показывает блок-схему модуля СПП в режиме SPI.

СПП состоит из приемо-передающего регистра сдвига SSPSR и буферного регистра SSPBUF. В SSPSR выполняется сдвиг данных в/из микроконтроллера старшим битом вперед. SSPBUF хранит данные, которые были записаны в SSPSR, до получения новых данных. Принятый байт данных перемещается в регистр SSPBUF. Далее устанавливаются флаг заполнения буфера BF (SSPSTAT<0>) и выставляется сигнал запроса на обработку прерывания SSPIF (PIR2<7>). Структура регистров SSPBUF и SSPSR позволяет хранить полученные данные в течение приема следующего байта данных. Любое обращение по записи к регистру SSPBUF во время передачи/приема данных будет игнорироваться, кроме того, будет установлен флаг обнаружения конфликта записи WCOL (SSPCON1<7>). Очистка флага WCOL в «0» должна быть осуществлена программно (обращением по записи к регистру SSPCON1).



**Рис. 43** Структурная схема модуля СПП в SPI режиме

При приеме данных в режиме SPI, регистр SSPBUF должен быть считан до момента окончания приема следующего байта. Установка флаг заполнения буфера, BF (SSPSTAT<0>), в «1» указывает, что SSPBUF загружен полученными данными (передача завершена). После обращения к регистру SSPBUF по чтению, бит BF автоматически сбрасывается в «0». Как правило, для определения окончания передачи или приема используется прерывание от модуля СПП. SSPBUF должен быть считан и записан в программе обработки прерываний. Прямого доступа по чтению и записи к регистру SSPSR не существует, обращение к нему осуществляется через обращение к регистру SSPBUF.

### **Настройка выводов при работе с модулем СПП в режиме SPI.**

#### **Типовое включение**

Для включения модуля СПП необходимо установить разряд SSPEN (регистр SSPCON1<5>) в «1». Для сброса или перенастройки модуля, рекомендуется сбросить в «0» разряд SSPEN, реинициализировать регистры SSPCON, затем заново установить бит SSPEN в «1». Это позволит сконфигурировать выводы SDI, SDO, SCK и SS для использования в качестве выводов интерфейса SPI. Разряды регистров управления направлением выводов (DDR) должны быть настроены следующим образом:

- SDI, автоматически настраивается при включении модуля в режиме SPI
- SDO, DDRB<7> должен быть сброшен в «0»
- SCK (режим ведущего), DDRB<6> должен быть сброшен в «0»
- SCK (режим ведомого), DDRB<6> должен быть установлен в «1»
- SS, PORTA<2> должен быть установлен в «1»

Главный контроллер инициализирует передачу данных, формируя тактовый сигнал SCK. Данные сдвигаются по установленному фронту тактового сигнала и принимаются на противоположном фронте тактового сигнала. Оба микроконтроллера должны иметь одинаковую конфигурацию режима SPI, в части состояния разряда СКР (управление сигналом тактовой частоты).

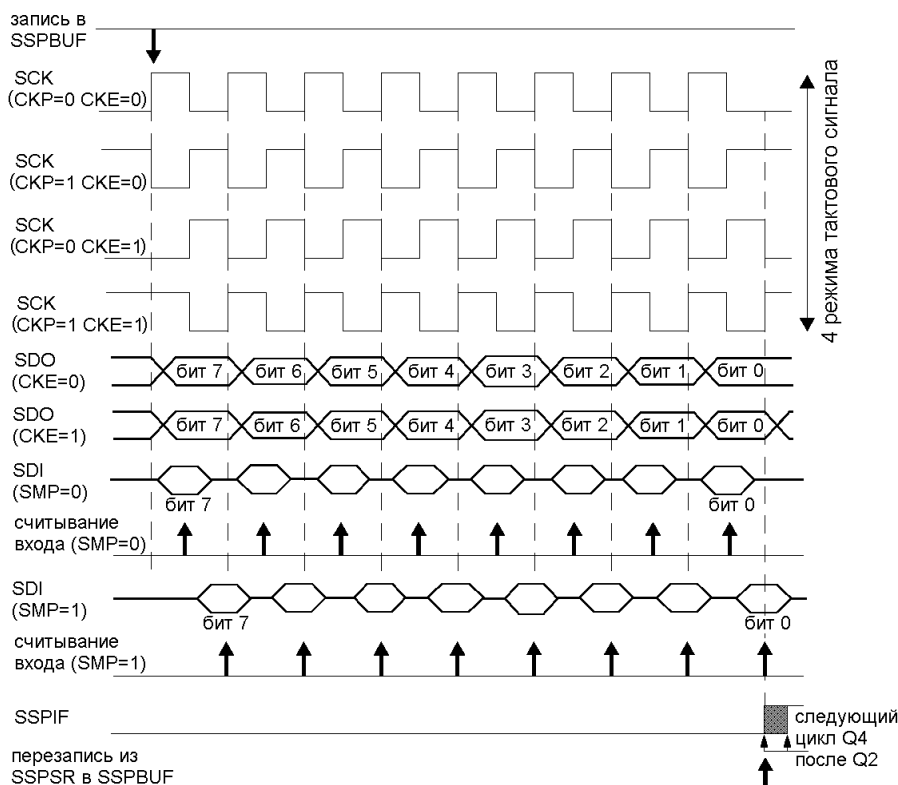
### **Режим ведущего**

Модуль СПП в режиме ведущего может инициировать передачу данных в любое время, т.к. в этом случае он контролирует линию SCK. Обмен данными иницируется путем обращения по записи к регистру SSPBUF. Ведущий определяет, когда ведомый может пересылать данные в соответствии с используемым протоколом. В режиме ведущего данные считаются переданными/принятыми, как только произошла запись в регистр SSPBUF из сдвигового регистра. Данные со входа SDI последовательно (потактно) вдвигаются в регистр SSPSR со скоростью, определяемой регистрами управления. При получении 8 бит (байта), полученный байт загружается в регистр SSPBUF, при этом формируются сигнал запроса на обработку прерывания и определенные флаги.

Управление сигналом тактовой частоты, выбор активного фронта, моменты передачи и приема данных производится с помощью записи разрядов СКР регистра SSPCON1, СКЕ и SMP регистра SSPSTA, возможные варианты передачи/приема данных показаны на рисунках Рис. 43, Рис. 45 и Рис. 46. Рис. 44 показывает варианты передачи данных для режима ведущего. Например, если разряд СКЕ установлен в «1», то данные на выходе SDO устанавливаются еще до момента первого фронта на линии SCK. Момент приема входных данных определяется разрядом SMP. Кроме того, рисунок демонстрирует момент загрузки SSPBUF полученными данными. Данные всегда передаются старшим разрядом вперед. Существуют следующие варианты выбора тактовой частоты SPI в режиме ведущего:

- $F_c/4$  (или  $T_{ic}$ )
- $F_c/16$  (или  $4 \cdot T_{ic}$ )
- $F_c/64$  (или  $16 \cdot T_{ic}$ )
- Частота с выхода Таймера 2 ( $TMR2IF\_set$ ) / 2

Таким образом, максимальная частота передачи данных составляет 8.25 МГц при тактовой частоте микроконтроллера 33 МГц.



**Рис. 44** Работа модуля СПП в режиме ведущего SPI

## Режим ведомого

В режиме ведомого данные передаются и принимаются по внешнему тактовому сигналу поступающего со входа SCK. При приеме последнего разряда, устанавливается флаг прерывания SSPIF (PIR2<7>). Параметры внешнего тактового сигнала должны удовлетворять требованиям, описанным в разделе электрических характеристик. Ведомый может передавать/получать данные, находясь в режиме покоя (Sleep режим). При получении байта данных микроконтроллер выходит из режима покоя.

## Выбор синхронного режима ведомого

Наличие вывода SS позволяет использовать модуль СПП в режиме синхронного ведомого. В данном случае разряды регистра SSPCON1<3:0> должны иметь значение 04h, то есть установлена конфигурация «SPI режим ведомого, тактовый сигнал с вывода SCK, разрешено управление выводом SS». При таком использовании модуля PORTA<2> должен быть установлен в «1». Если на выводе SS подан низкий логический уровень, передача и прием разрешены, вывод SDO управляет модулем. Когда на вывод SS подан высокий уровень сигнала, вывод SDO отключен и находится в «третьем» состоянии. При таком использовании модуля рекомендуется применять внешние подтягивающие резисторы. Кроме того, подача логической единицы на вход SS фактически означает сброс модуля. Сброс модуля в режиме SPI означает, сброс счетчика битов сдвигового регистра.

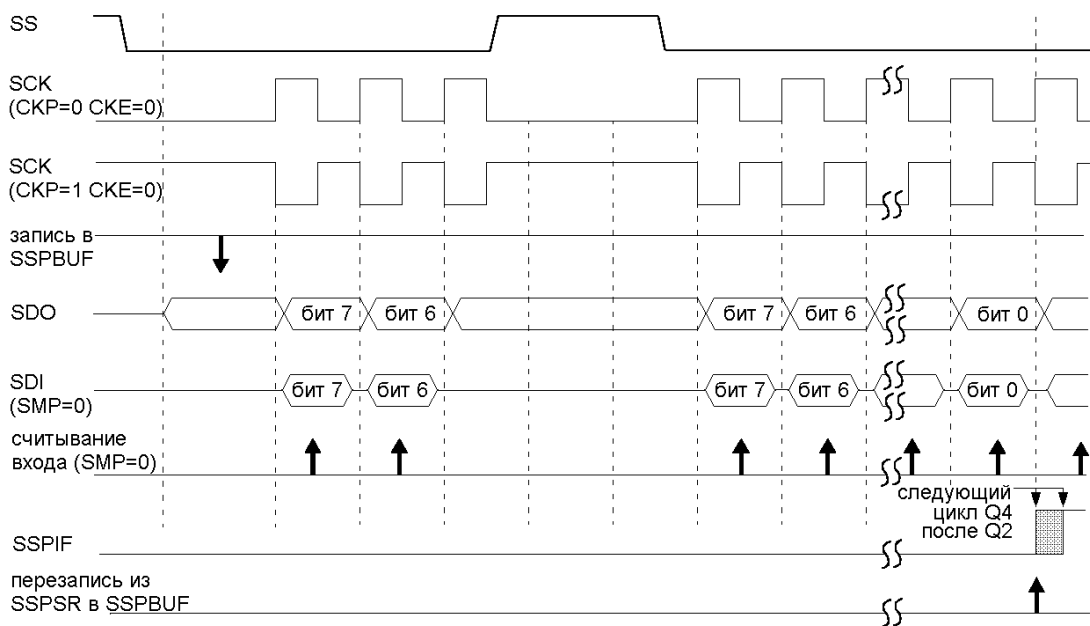


**Функционирование в режиме покоя (Sleep режим). Эффект сброса**

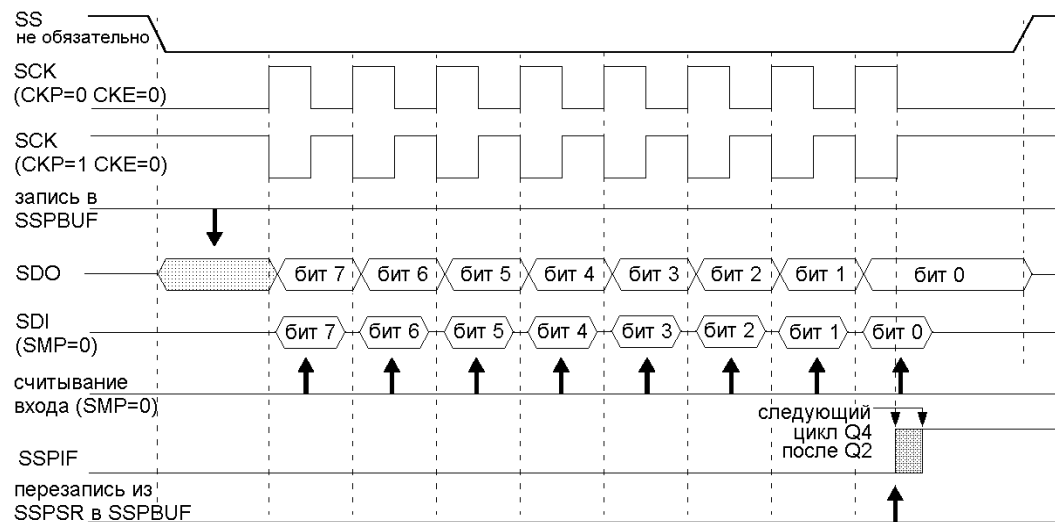
Так как в режиме покоя системный тактовый сигнал отсутствует, модуль СПП, включенный в режиме SPI ведущий, не может передавать или получать данные и будет сохранять свое текущее состояние до выхода микроконтроллера из спящего режима.

В режиме SPI ведомый сдвиговый регистр функционирует асинхронно по отношению к системной частоте микроконтроллера, что позволяет осуществлять прием/передачу данных в момент нахождения микроконтроллера в режиме покоя. По приему 8 бит, устанавливается флаг прерывания от модуля СПП, который выводит микроконтроллер из спящего режима.

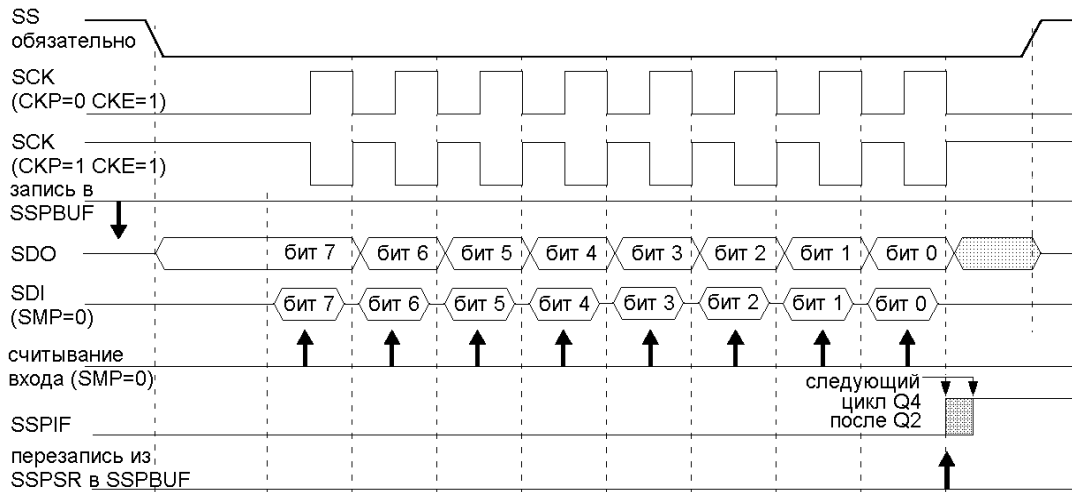
Любой сброс отключает модуль СПП и прерывает текущую передачу.



**Рис. 45** Демонстрация работы в режиме синхронного ведомого



**Рис. 46** Функционирование модуля в режиме SPI ведомый (CKE = 0)



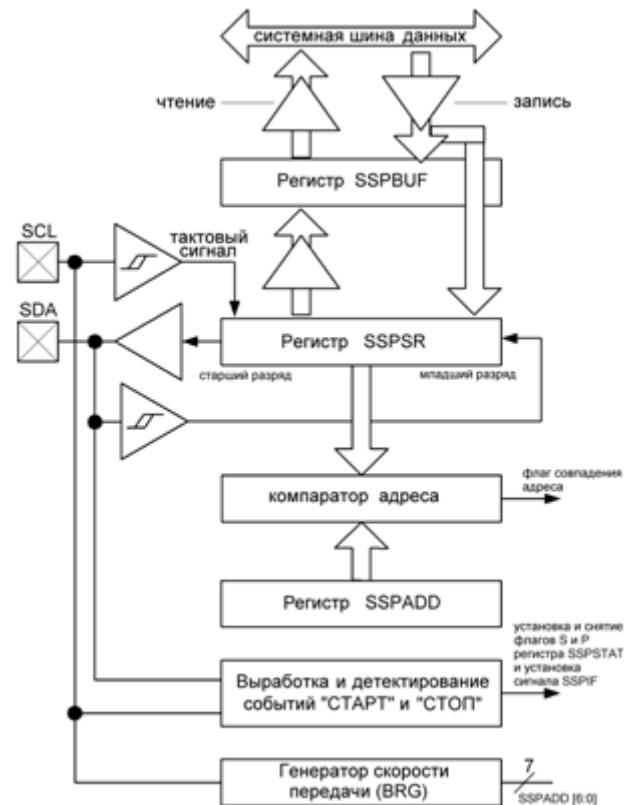
**Рис. 47** Функционирование модуля в режиме SPI ведомый (SCKE = 1)

**Работа СПП в режиме I2C**



**Рис. 48**

Структурная схема модуля СПП в режиме ведущего I2C



**Рис. 49**

Структурная схема модуля СПП в режиме ведомого I2C

Модуль СПП в режиме I2C полностью поддерживает все функции ведущего и ведомого, включая поддержку общего вызова, и обеспечивает аппаратные прерывания при обнаружении битов СТАРТ и СТОП для определения занятости шины (при конкуренции на шине). Модуль СПП поддерживает режимы с 7-ми и 10-ти разрядной адресацией.

Фильтр «glitch» установлен на выводах SCL и SDA, когда они настроены на вход. Этот фильтр функционирует в двух режимах 100 КГц и 400 КГц. В режиме 100 КГц, когда эти выводы являются выходами, этот фильтр контролирует длительности формируемых сигналов независимо от тактовой частоты микроконтроллера.

Для передачи данных используются два вывода. Это вывод SCL, который является сигналом синхронизации и вывод SDA, являющийся шиной данных. Выводы SCL и SDA автоматически настраиваются, когда включается режим I2C. Модуль СПП включается установкой в «1» бита разрешения SSPEN (SSPCON1<5>).

Модуль СПП имеет 6 регистров управления для функционирования в режиме I2C:

- регистр управления 1 - SSPCON1;
- регистр управления 2 - SSPCON2;
- регистр статуса – SSPSTAT;
- буфер приема/передачи – SSPBUF;
- сдвиговый регистр - SSPSR (не имеет прямого доступа);
- регистр адреса – SSPADD.

Регистр SSPCON1 позволяет управлять режимом функционирования I2C. С помощью битов (SSPCON1<3:0>) можно выбрать один из следующих режимов I2C:

- I2C режим ведомого (7-разрядная адресация);
- I2C режим ведомого (10-разрядная адресация);
- I2C режим ведущего, тактовый сигнал =  $F_{cl}/(4*(SSPADD + 1))$ .

Перед выбором любого режима I2C, выводы SCL и SDA должны быть запрограммированы на вход путем установки соответствующих битов регистра DDR. После выбора режима I2C, установка бита SSPEN в «1», включает выводы SCL и SDA в качестве линии синхронизации и линии данных в режиме I2C.

Регистр SSPSTAT содержит информацию о статусе передачи данных. Эта информация включает флаги обнаружения битов СТАРТ и СТОП, определяет, был ли полученный байт данными или адресом, является ли следующий байт завершением 10-разрядного адреса и флаг операции прием или передача данных.

SSPBUF является регистром, из которого считываются принятые данные, и в который записываются передаваемые данные. Регистр SSPBUF осуществляет сдвиг принимаемых и передаваемых данных. При операциях приема данных SSPBUF и SSPSR образуют двойной буферизированный приемник. Это позволяет начать прием следующего байта до считывания последнего байта полученных данных. После получения полного байта он перемещается из регистра SSPSR в регистр SSPBUF и устанавливается флаг SSPBUF. Если другой полный байт принят до того, как считан регистр SSPBUF, происходит переполнение приемника, устанавливается бит SSPOV (SSPCON1<6>), а байт из SSPSR теряется.

В регистр SSPADD записывается адрес ведомого устройства. В режиме 10-ти разрядного адреса, необходимо сначала записать старший байт адреса (1111 0 A9 A8 0), а после совпадения старшего байта необходимо загрузить младший байт адреса (A7:A0).

### Режим ведомого

В режиме ведомого выводы SCL и SDA должны быть сконфигурированы как входы. Модуль СПП заменит при необходимости направление вывода при передаче данных ведомым.

Когда адрес совпал или получен байт данных после совпадения адреса, аппаратно генерируется бит подтверждения (ACK) и затем данные из регистра SSPSR загружаются в SSPBUF.

При определенных условиях модуль СПП не формирует бит подтверждения. Это условия (одно или оба одновременно):

- Установлен бит BF (SSPSTAT<0>) перед приемом данных.
- Установлен бит переполнения SSPOV (SSPCON1<6>) перед приемом данных.

Если бит BF установлен, значение регистра SSPSR не загружается в SSPBUF, но биты SSPIF и SSPOV устанавливаются. Таблица 39 показывает, что происходит, когда принят байт данных, при различных значениях битов BF и SSPOV. Затененные ячейки показывают случаи, когда пользователь не сбросил в «0» бит переполнения. Флаг BF аппаратно сбрасывается в «0» при считывании регистра SSPBUF, в то время как бит SSPOV сбрасывается в «0» программно.

Минимальная длительность логических уровней входного сигнала синхронизации SCL должна удовлетворять спецификации.

**Таблица 39**  
Действия после приема байта данных

Статусные биты		Загрузка данных из SSPSR в SSPBUF	Формирование бита ACK	Установка флага запроса прерываний SSPIF
BF	SSPOV			
0	0	да	да	да
1	0	нет	нет	да
1	1	нет	нет	да
0	1	да	нет	да

### Адресация

После включения модуля СПП, ожидается прием бита СТАРТ. После этого производится прием 8-ми битов в регистр SSPSR. Выборка битов происходит по переднему фронту тактового сигнала с вывода SCL. Значение регистра SSPSR<7:1> сравнивается со значением регистра SSPADD. Адрес сравнивается по заднему фронту 8 такта сигнала SCL. Если адреса совпадают, а биты BF и SSPOV =0, выполняются следующие действия:

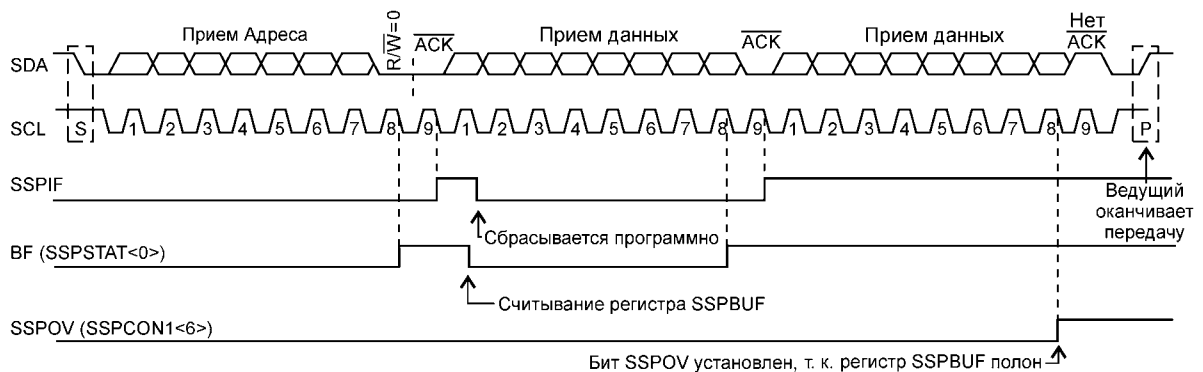
1. Значение регистра SSPSR загружается в регистр SSPBUF по 8 заднему фронту сигнала SCL.
2. Устанавливается флаг BF в «1» (буфер полон) по 8 заднему фронту сигнала SCL.
3. Генерируется бит подтверждения ACK.
4. Устанавливается флаг запроса прерывания SSPIF (PIR2<7>) и, если разрешено, генерируется прерывание (по 9 заднему фронту сигнала SCL).

В режиме ведомого при 10-ти разрядном адресе необходимо принять два байта адреса. Пять старших бит первого байта определяют, является ли

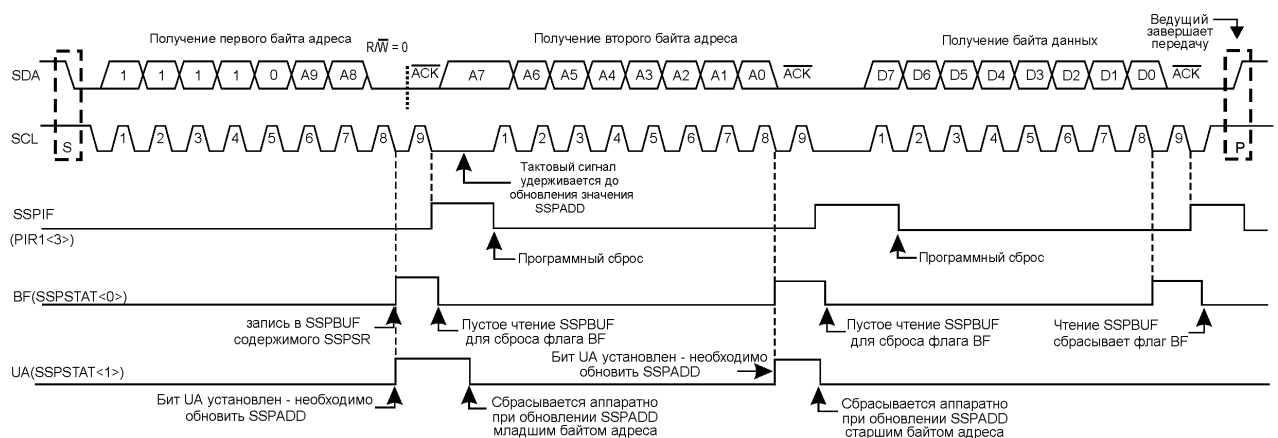
полученный байт первым байтом 10-ти разрядного адреса. Бит R/W (SSPSTAT<2>) должен быть настроен для приема второго байта адреса. Для 10-ти разрядной адресации первый байт адреса должен иметь формат «1111 0 A9 A8 0», где A9 и A8 - 2 старших бита адреса. Последовательность событий для 10-ти разрядной адресации (шаги 7-9 для ведомого передатчика) следующая:

1. Получить первый (старший) байт адреса (биты SSPIF, BF и UA (SSPSTAT<1>) устанавливаются в «1»).
2. Записать в регистр SSPADD второй (младший) байт адреса (аппаратно сбрасывается бит UA и «отпускается» линия SCL).
3. Считать регистр SSPBUF (бит BF сбрасывается) и сбросить флаг SSPIF.
4. Получить второй (младший) байт адреса (биты SSPIF, BF и UA устанавливаются в «1»).
5. Записать в регистр SSPADD первый (старший) байт адреса. Это сбрасывает бит UA и «отпускается» линия SCL.
6. Считать регистр SSPBUF (сбрасывает бит BF) и сбросить флаг SSPIF.
7. Принять бит повторный СТАРТ.
8. Получить первый (старший) байт адреса (биты SSPIF и BF устанавливаются в «1»).
9. Считать регистр SSPBUF (сбрасывает бит BF) и сбросить флаг SSPIF.

При приеме бита повторного СТАРТА (этап 7) в 10-разрядном режиме, пользователю нужно согласовать только первый байт адреса. Обновлять значение SSPADD для второй половины адреса не требуется.



**Рис. 50 Прием данных ведомым I2C (7-ми разрядная адресация)**



**Рис. 51 Прием данных ведомым I2C (10-ти разрядная адресация)**

### **Прием данных ведомым**

Когда бит R/W в адресном байте сброшен и принятый адрес совпадает с адресом устройства, то бит R/W в регистре SSPSTAT сбрасывается. Принятый адрес загружается в регистр SSPBUF. Если при приеме байта адреса есть условие переполнения (бит BF (SSPSTAT<0>) или бит SSPOV (SSPCON1<6>) установлен в «1»), то бит подтверждения ACK не формируется.

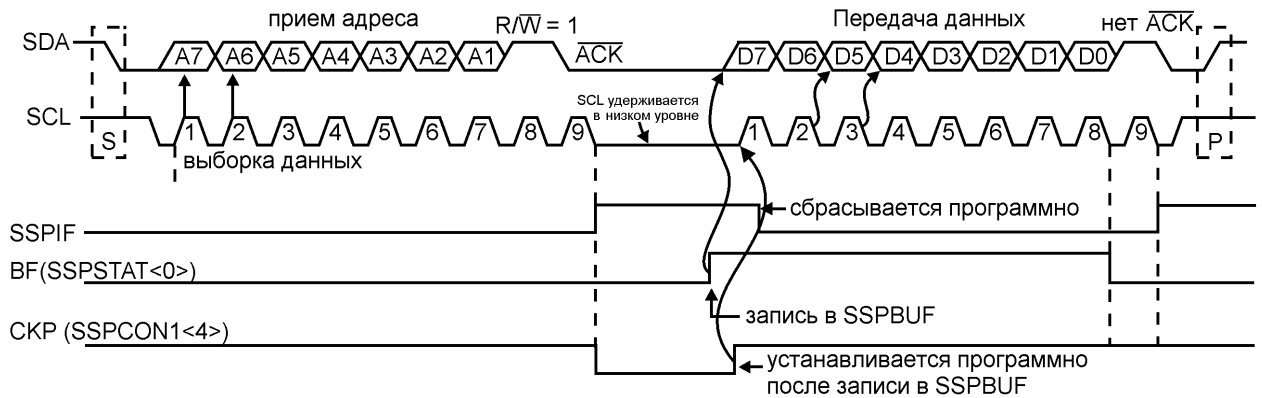
Прерывание от СПП генерируется для каждого принятого байта данных. Флаг запроса прерывания SSPIF (PIR2<7>) сбрасывается программно. Регистр SSPSTAT используется для определения типа принятого байта.

Регистр SSPBUF будет загружен значением из SSPSR, если бит SSPOV установлен в «1», а бит BF сброшен. Если было выполнено чтение SSPBUF, но не был сброшен бит SSPOV до следующего приема, ACK не формируется, но значение SSPBUF обновляется.

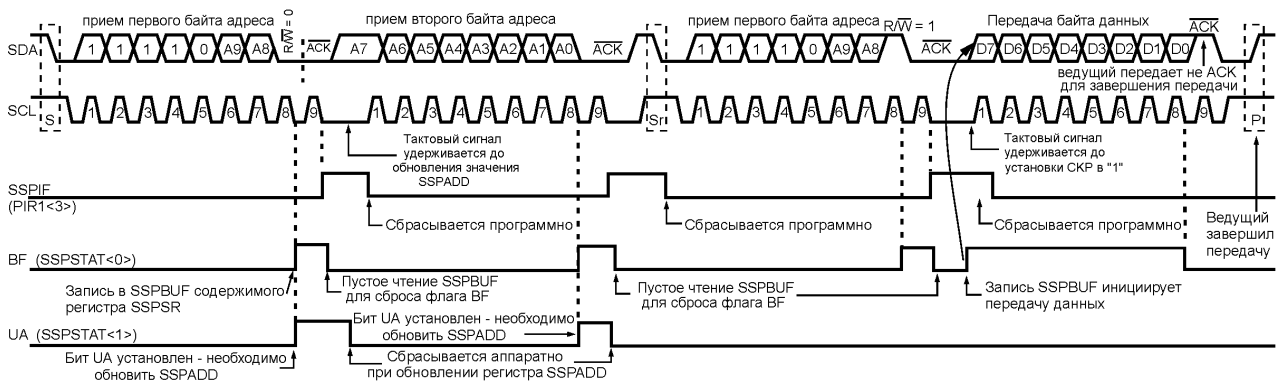
### **Передача данных ведомым**

Когда бит R/W в адресном байте установлен в «1» и принятый адрес совпадает, устанавливается бит R/W в регистре SSPSTAT. Полученный адрес загружается в регистр SSPBUF. Бит ACK формируется 9 битом, вывод SCL удерживается в низком логическом уровне. Передаваемые данные должны быть записаны в регистр SSPBUF, из которого они перегружаются в регистр SSPSR. После этого вывод SCL включается установкой бита СКР (SSPCON1<4>). Ведущий должен контролировать вывод SCL, ожидая смены уровня сигнала. Ведомые устройства могут задержать ведущих удлинением тактового сигнала. Восемь бит данных последовательно сдвигаются по заднему фронту сигнала SCL, что гарантирует достоверное значение данных на линии SDA (см. Рис. 52).

Прерывание от СПП генерируется для каждого переданного байта данных. Флаг запроса прерывания SSPIF должен быть сброшен программно, регистр SSPSTAT используется для определения статуса передачи. Флаг SSPIF устанавливается по заднему фронту 9 такта сигнала. Ведущее устройство формирует бит ACK на 9 такте сигнала SCL для каждого принятого байта. Если на линии SDA высокий уровень сигнала, т.е. ACK не сформирован, передача данных завершается. Логика ведомого сбрасывается и настраивается на обнаружение бита СТАРТ. Если на линии SDA низкий уровень сигнала, т.е. бит подтверждения ACK получен, то в регистр SSPBUF записывается новый байт для передачи, который перегружается в регистр SSPSR. Затем линию SCL необходимо разрешить установкой бита СКР.



**Рис. 52** Передача данных ведомым I2C (7-ми разрядная адресация)



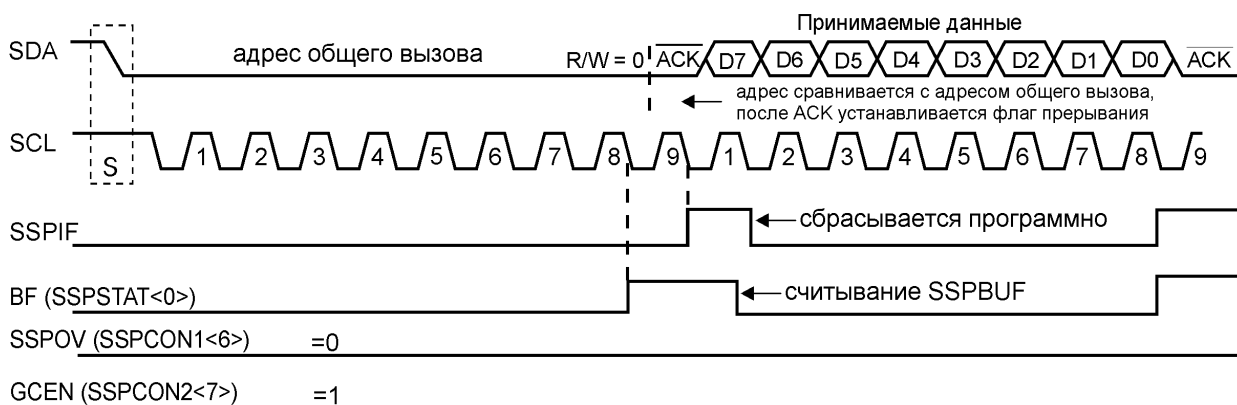
**Рис. 53** Передача данных ведомым I2C (10-ти разрядная адресация)

## Поддержка общего вызова

Процедура адресации для шины I2C состоит в том, что первый байт после условия СТАРТ обычно определяет, какое ведомое устройство адресуется ведущим. Исключение, это адрес общего вызова, который адресует все устройства. Когда используется этот адрес, все устройства, должны ответить подтверждением. Адрес общего вызова это один из восьми зарезервированных адресов шины I2C. Он состоит из всех нулей, включая бит R/W = 0.

Адрес общего вызова распознается если установлен бит GCEN (SSPCON2<7>). Следующий за флагом СТАРТ, байт помещается в SSPSR и сравнивается с SSPADD, а также с адресом общего вызова, заданным аппаратно. Если адрес совпал с адресом общего вызова то: значение из SSPSR переписывается в SSPBUF, по заднему фронту 8 такта устанавливается флаг BF, на 9 такте формируется флаг ACK и устанавливается флаг SSPIF. Считыванием содержимого регистра SSPBUF можно определить произошло обращение по адресу устройства или общий вызов.

В 10-ти разрядном режиме, после установки бита UA (SSPSTAT<1>), требуется обновление значения SSPADD, для проверки младшего байта адреса. Если ведомый, в режиме 10-ти разрядной адресации, получает адрес общего вызова и бит GCEN установлен, то вторая половина адреса не требуется, бит UA не устанавливается, а ведомый начнет получать данные после формирования флага подтверждения ACK (см. Рис. 54).



**Рис. 54** Получение общего вызова ведомым I2C (7-ми или 10-ти разрядная адресация)

### Работа модуля в спящем режиме. Эффект при сбросе

Находясь в спящем режиме, модуль I2C может принимать адреса и данные, и когда совпадает адрес или происходит прием полного байта принимаемых данных, процессор выходит из спящего режима (если разрешено прерывание от СПП).

Любой сброс выключает модуль СПП и прекращает передачу данных.

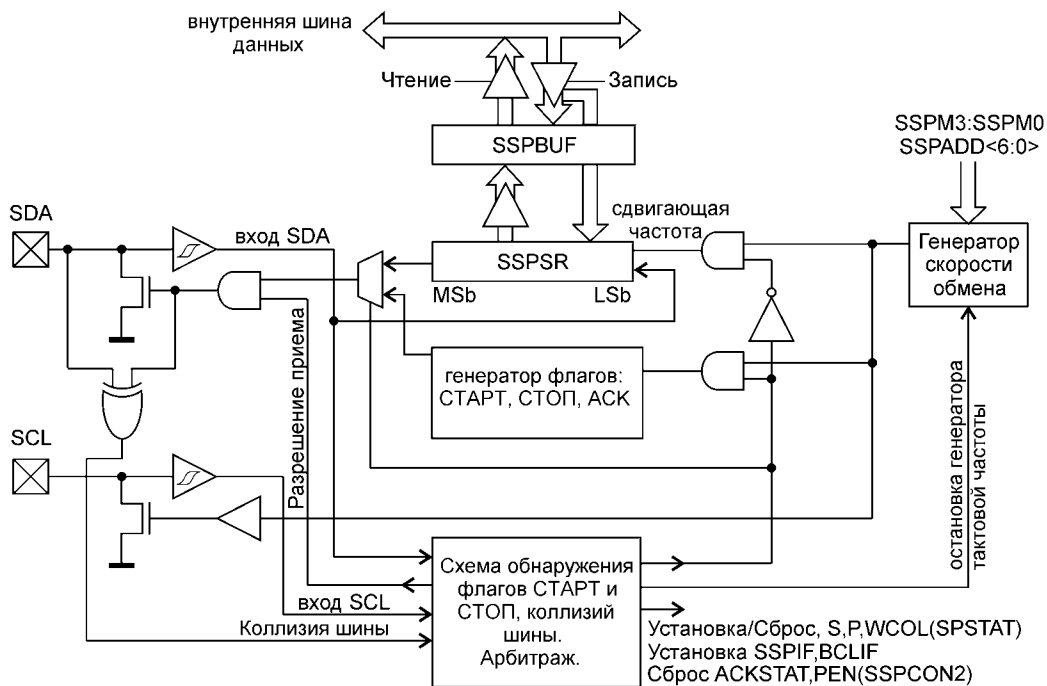
### Режим ведущего

В режиме ведущего поддерживается генерация прерываний при выявлении условий СТАРТ и СТОП. Биты Р «СТОП» и S «СТАРТ» сброшены после сброса микроконтроллера, или при выключенном модуле СПП. Управление шиной I2C может производиться, когда бит Р установлен, или когда шина свободна (оба бита S и Р сброшены).

В режиме ведущего линии SCL и SDA аппаратно управляются модулем СПП. Следующие события приведут к установке флага запроса прерывания от СПП SSPIF и прерыванию, если оно разрешено:

- Произошло условие СТАРТ.
- Произошло условие СТОП.
- Передан/принят байт данных.
- Передан флаг подтверждения ACK.
- Произошло условие повторный СТАРТ.





**Рис. 55** Структурная схема модуля СПП в режиме ведущего I2C

### **Режим с несколькими ведущими (конкуренция)**

В режиме конкуренции генерация прерывания при выявлении условий СТАРТ и СТОП позволяет определить, когда канал свободен. Биты P «СТОП» и S «СТАРТ» сбрасываются при сбросе микроконтроллера или при выключенном модуле СПП. Управление шиной I2C можно начать когда установлен бит P (SSPSTAT<4>) или шина свободна, т.е. биты S и P сброшены. Когда шина занята, разрешение прерывания от СПП можно использовать для обнаружения условия СТОП на шине.

В случае нескольких ведущих, линия SDA должна проверяться на соответствие уровня сигнала на выходе с ожидаемым уровнем. Проверка производится аппаратно, результат помещается в бит BCLIF. Состояния, когда арбитраж может быть потерян:

- передача адреса;
- передача данных;
- Условие СТАРТ;
- Условие повторный СТАРТ;
- Передача флага подтверждения АСК.

### **Поддержка режима ведущего I2C**

Режим ведущего включается программированием соответствующих битов SSPM в SSPCON1 и установкой бита SSPEN. В режиме ведущего пользователю предоставляются шесть функций:

- Формирование условия СТАРТ на линиях SDA и SCL.
- Формирование условия повторный СТАРТ на линиях SDA и SCL.
- Запись в регистр SSPBUF, инициирующая передачу данных/адреса.

- Генерация условия СТОП на линиях SDA и SCL.
- Конфигурирование порта I2C для приема данных.
- Генерирование флага подтверждения ACK в конце принятого байта данных.

Модуль СПП в режим ведущего I2C, не имеет стек событий. Например, пользователь не может, запустив формирование условия СТАРТ, сразу же не дожидаясь выполнения формирования условия, произвести запись в регистр SSPBUF данных для передачи. В этом случае, регистр SSPBUF не будет записан, а бит WCOL установится в «1», указывая, что запись в SSPBUF не выполнена. Ведущий генерирует на шине тактовый сигнал и флаги СТАРТ и СТОП. Передача заканчивается с флагом СТОП или флагом повторный СТАРТ. Т.к. повторный СТАРТ является началом следующей последовательной передачи данных, то шина I2C не будет освобождена.

Передатчик ведущего выдает последовательные данные на линию SDA и тактовый сигнал на линию SCL. Первый передаваемый байт содержит 7-ми разрядный адрес ведомого устройства и бит чтения/записи данных R/W. Бит R/W равен «0» если ведущий производит передачу данных, и равен «1» если прием. Данные передаются по 8 бит, после которых принимается флаг подтверждения ACK. Флаги СТАРТ и СТОП указывают начало и конец передачи.

Генератор скорости обмена, используемый для функционирования режима SPI, сейчас используется для установления частоты тактового сигнала на линии SCL: 100 КГц, 400 КГц или 1 МГц. Значение для перезагрузки генератора скорости обмена берется из 7 младших бит регистра SSPADD. Генератор скорости обмена автоматически начнет работу после записи данных в SSPBUF. Когда операция завершена (передан последний бит байта и получен флаг подтверждения ACK), генератор тактового сигнала останавливается и вывод SCL остается в последнем состоянии.

Обычная последовательность этапов при передаче данных:

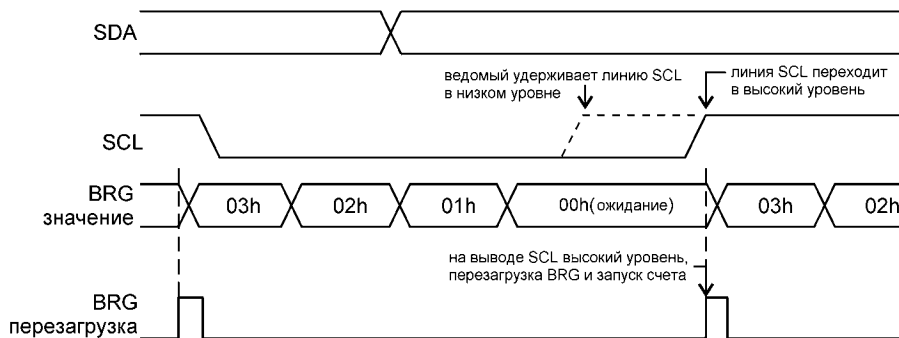
1. Пользователь генерирует условие СТАРТ установкой бита разрешения SEN в регистре SSPCON2.
2. После выполнения операции модулем СПП устанавливается бит SSPIF и генерируется прерывание (если оно разрешено).
3. Пользователь загружает регистр SSPBUF значением адреса.
4. На вывод SDA передаются все 8 бит.
5. Модуль СПП принимает флаг подтверждения ACK от ведомого устройства и записывает его значение в регистр SSPCON2 (SSPCON2<6>).
6. По заднему фронту 9 такта устанавливается бит SSPIF и генерируется прерывание (если оно разрешено).
7. Пользователь загружает регистр SSPBUF байтом данных.
8. Данные выводятся на вывод SDA (все 8 бит).
9. Модуль СПП принимает флаг подтверждения ACK от ведомого устройства и записывает его значение в регистр SSPCON2 (SSPCON2<6>).
10. По заднему фронту 9 такта устанавливается бит SSPIF и генерируется прерывание (если оно разрешено).
11. Пользователь генерирует условие СТОП установкой бита разрешения PEN в регистре SSPCON2.
12. После выполнения операции модулем СПП устанавливается бит SSPIF и генерируется прерывание (если оно разрешено).

**Генератор скорости обмена**

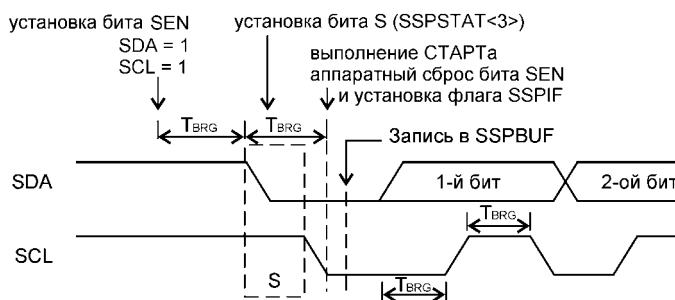
В режиме ведущего I2C, значение для перезагрузки генератора скорости обмена берется из младших 7 бит регистра SSPADD (см. Рис. 56). Счетчик генератора загружается этим значением, далее считает, декрементируясь до 0 (дважды в каждом командном цикле, в тактах Q2 и Q4), и останавливается до следующей перезагрузки, которая не всегда производится автоматически. Если имеет место арбитраж шины, то после окончания счета, счетчик перезагрузится только когда сигнал на линии SCL перейдет к требуемому высокому уровню (см. Рис. 57).



**Рис. 56** Структурная схема генератора скорости обмена



**Рис. 57** Работа генератора скорости обмена с арбитражем линии SCL

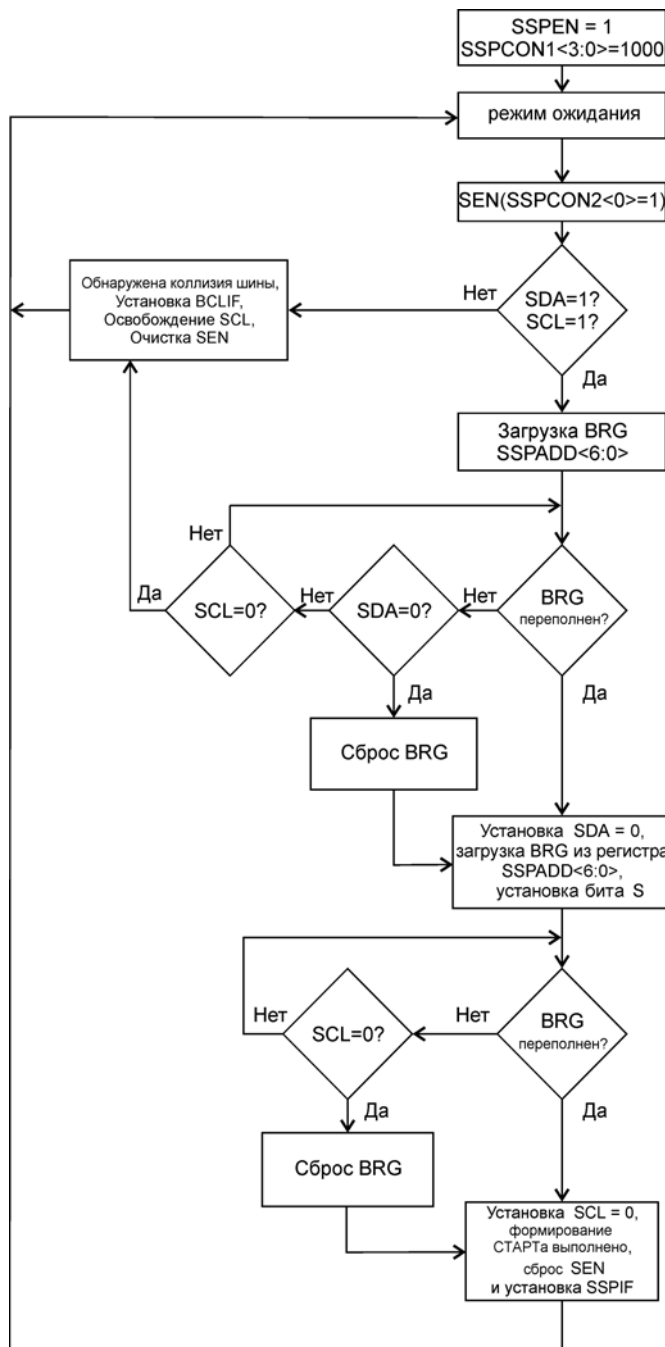


**Рис. 58** Формирование флага СТАРТ

**Формирование флага СТАРТ в режиме ведущего I2C**

Чтобы запустить формирование флага СТАРТ на шине I2C, нужно установить бит разрешения условия СТАРТ - SEN (SSPCON2<0>). Если на линиях SDA и SCL высокий уровень сигнала, генератор скорости обмена загружается значением из регистра SSPADD<6:0> и начинает счет. Если по окончании отсчета времени счетчиком генератора (TBRG) сохраняется высокий уровень на линиях SCL и SDA, сигнал SDA переводится в низкий логический уровень. Перевод SDA в

низкий уровень, в то время, как на SCL высокий, является флагом СТАРТ на шине I2C. После этого устанавливается бит S (SSPSTAT<3>).



**Рис. 59** Блок схема формирования флага СТАРТ

Счетчик генератора загружается новым значением из регистра SSPADD<6:0> и начинает счет. По окончании счета бит SEN (SSPCON2<0>) аппаратно сбрасывается, генератор останавливается, на линии SDA остается низкий уровень сигнала. Формирование флага СТАРТ окончено.

Если в начале формирования флага СТАРТ, на выводах SDA и SCL присутствует низкий уровень или во время формирования СТАРТ низкий уровень на SCL появляется раньше, чем на SDA, устанавливается флаг прерывания при конфликте шины (BCLIF), формирование флага СТАРТ прекращается, модуль I2C сбрасывается в неактивное состояние.

Если производится запись в регистр SSPBUF до окончания формирования флага СТАРТ, то устанавливается статусный флаг WCOL, а значение буфера не изменяется (т.е. запись не производится). Так как модуль не имеет стека событий, то до завершения формирования флага СТАРТ, нельзя устанавливать любой из 5 младших битов регистра SSPCON2.

### **Формирование флага повторный СТАРТ в режиме ведущего I2C**

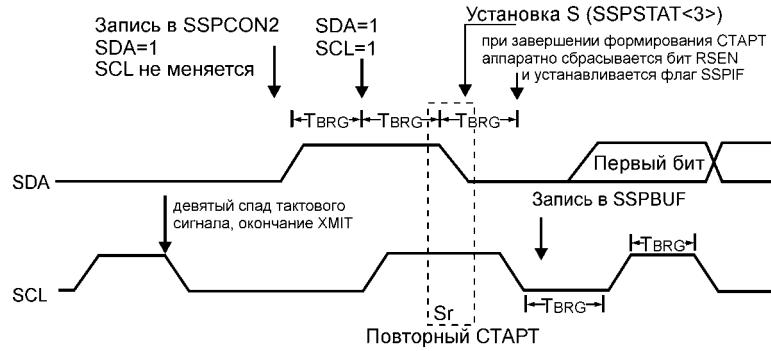
Для запуска формирования флага повторный СТАРТ, необходимо установить в «1» бит RSEN (SSPCON2<1>), при этом модуль I2C должен находиться в режиме ожидания. Если установка бита RSEN производится, когда модуль выполняет какое-либо действие, то это не приведет ни к какому эффекту. При установке бита RSEN, вывод SCL переводится в низкий логический уровень. Когда на SCL устанавливается низкий уровень, генератор скорости обмена загружается содержимым SSPADD<6:0> и начинает счет. Линия SDA отпускается в высокий уровень. Если по окончании счета генератора скорости обмена, на линии SDA сохраняется высокий уровень, то линия SCL также отпускается в высокий уровень. Генератор скорости перезагружается и начинает отсчет. Если по окончании отсчета времени TBRG на линиях SDA и SCL сохраняется высокий уровень, то сигнал SDA переводится в низкий уровень. После формирования на линиях SDA и SCL флага повторный СТАРТ, устанавливается бит S (SSPSTAT<3>). Генератор скорости перезагружается и отсчитывает TBRG, далее бит RSEN в регистре SSPCON2 аппаратно сбрасывается, а бит SSPIF устанавливается.

Конфликт шины во время формирования флага повторный СТАРТ происходит, если:

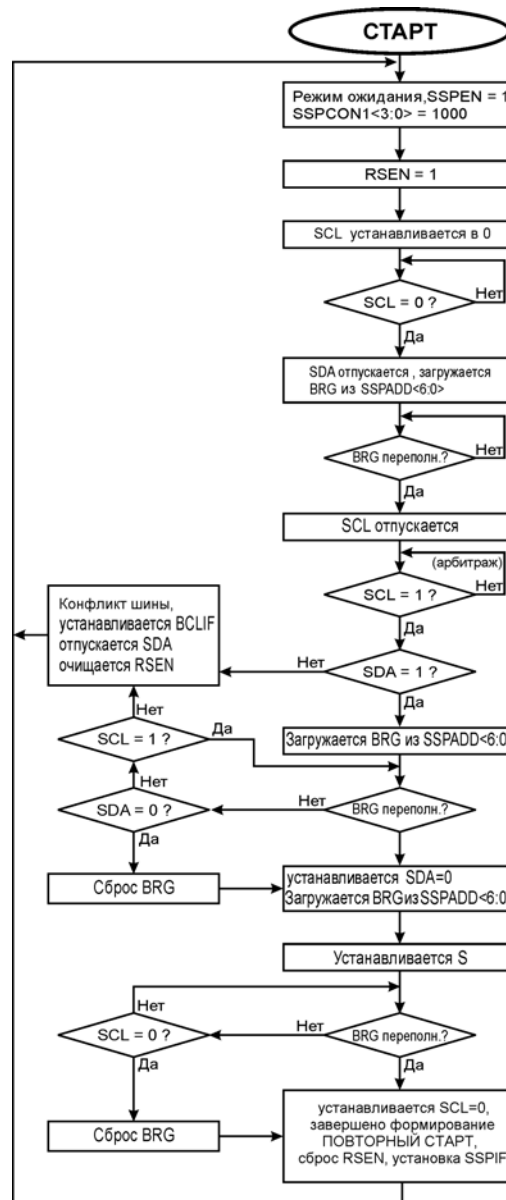
- На SDA низкий уровень, когда SCL переходит из низкого в высокий.
- Низкий уровень на SCL появляется раньше, чем на SDA.. Это может показывать, что другой ведущий пытается передать данные.

Сразу после установки бита SSPIF пользователь может загрузить в регистр SSPBUF 7-ми разрядный адрес в режиме 7-ми разрядной адресации, или старший байт 10-ти разрядного адреса. По завершению передачи 8 бит и получения флага подтверждения ACK, можно передать байт данных или младший байт адреса.

Если, во время формирования флага повторный СТАРТ, производится запись в регистр SSPBUF, то устанавливается WCOL и содержимое буфера остается неизменным (запись не производится). Так как модуль не имеет стека событий, то до завершения формирования флага повторный СТАРТ, нельзя устанавливать любой из 5 младших битов регистра SSPCON2.



**Рис. 60** Формирование флага повторный СТАРТ



**Рис. 61** Блок-схема формирования флага повторный СТАРТ

### **Передача данных в режиме ведущего I2C**

Для передачи байта данных, 7-ми разрядного адреса, или части 10-ти разрядного адреса, нужно просто записать байт в регистр SSPBUF. Это действие установит бит BF и запустит генератор скорости обмена для передачи данных. Каждый передаваемый бит будет выдаваться на SDA по заднему фронту сигнала SCL. Низкий уровень на SCL удерживается в течение одного периода генератора (TBRG). Данные должны устанавливаться на SDA до прихода переднего фронта на SCL (см раздел временных характеристик). После перехода SCL в высокий уровень, данные на SDA не должны изменяться до и в течение некоторого времени после перехода SCL в низкий уровень. По окончании передачи 8 бит, бит BF сбрасывается, и ведущий отпускает линию SDA для приема флага подтверждения ACK. По заднему фронту 9 такта значение ACK записывается в бит AKSTAT. После этого бит SSPIF устанавливается в «1», генератор скорости обмена останавливается, линия SCL остается в низком уровне, а линия SDA не изменяется (см. Рис. 63).

При передаче бит BF (SSPSTAT<0>) устанавливается, когда CPU записывает данные в SSPBUF и сбрасывается, когда переданы все 8 бит. Если происходит запись в SSPBUF, когда уже происходит передача (SSPSR все еще сдвигает байт данных), то устанавливается бит WCOL и содержимое буфера остается неизменным (запись не происходит). WCOL должен быть сброшен программно.

В режиме передачи, бит AKSTAT (SSPCON2<6>) сбрасывается, когда ведомый послал подтверждение (ACK= 0) и устанавливается, когда ведомый не подтверждает (ACK = 1). Ведомый посылает подтверждение, когда распознал адрес (включая общий вызов) или корректно принял данные.

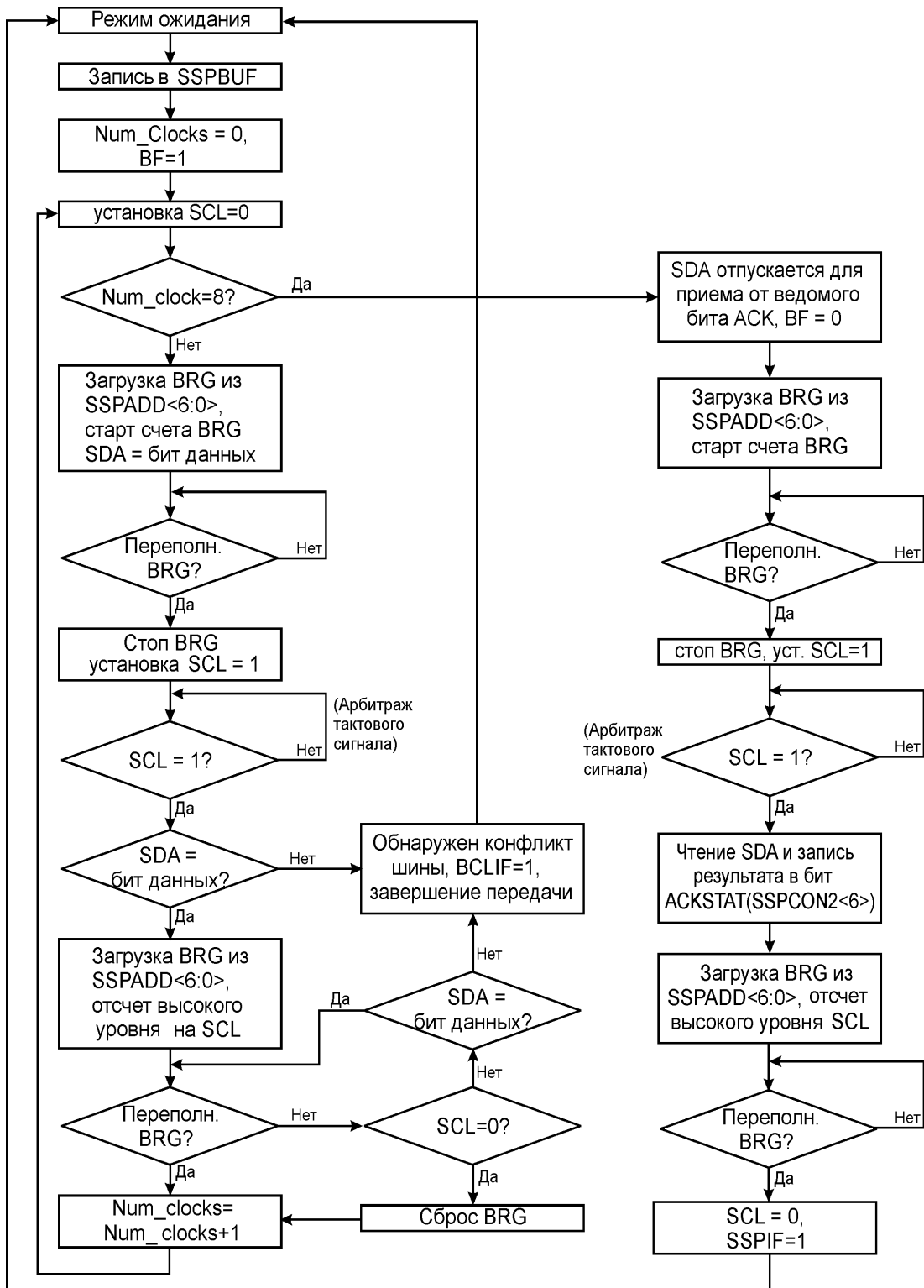
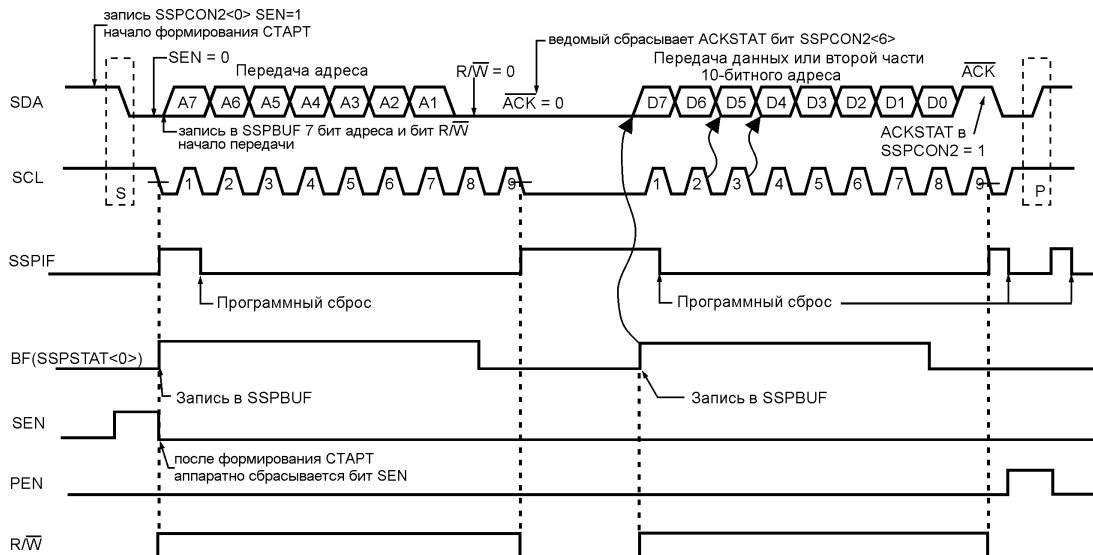


Рис. 62 Блок схема передачи данных в режиме ведущего I2C





**Рис. 63** Передача данных в режиме ведущего I2C (7 или 10-разрядная адресация)

### Прием данных в режиме ведущего I2C

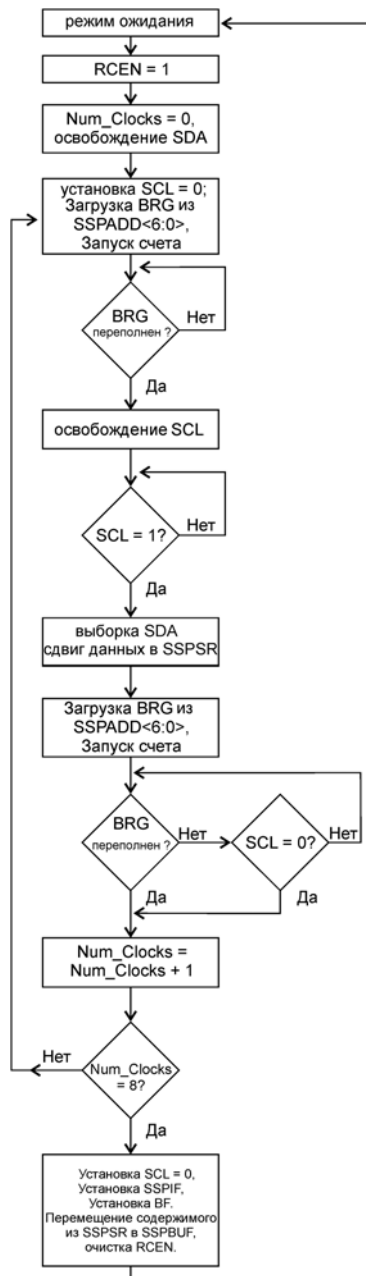
Прием данных в режиме ведущего разрешается установкой в «1» разряда разрешения приема RCEN регистра SSPCON2<3>. Модуль СПП должен находиться в режиме ожидания в момент установки разряда RCEN, в противном случае его установка будет проигнорирована.

Генератор скорости обмена начинает формировать тактовые сигналы SCL для приема данных в сдвиговый регистр SSPSR. Каждый бит данных будет приниматься со входа SDA по заднему фронту SCL. По заднему фронту 8 такта, значение из SSPSR переписывается в SSPBUF разряд разрешения приема данных автоматически сбрасывается в «0», после чего устанавливается флаг заполнения буфера BF, сигнал запроса на обработку прерывания SSPIF, генератор останавливается, удерживая линию SCL в низком уровне, а модуль SSP переходит в режим ожидания для получения следующей команды. После обращения к регистру SSPBUF по чтению, флаг BF аппаратно сбрасывается в «0». Далее, например, для окончания приема ведущий может, передать бит подтверждения установкой разряда ACKEN регистра SSPCON2<4>.

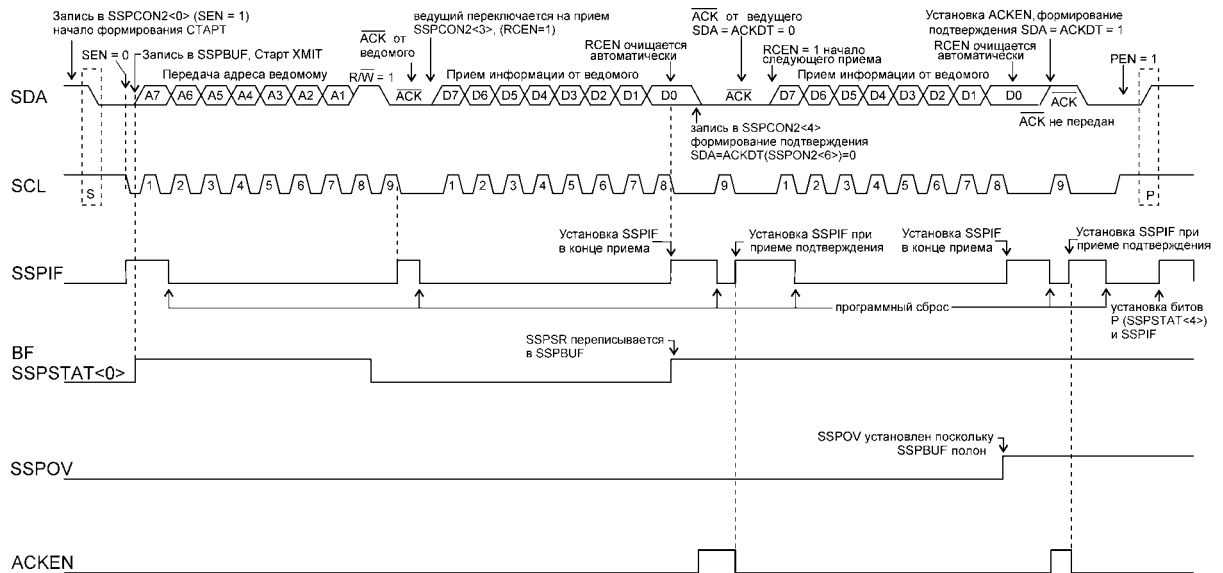
При приеме данных флаг BF устанавливается в «1», в момент загрузки байта адреса или данных из SSPSR в SSPBUF. Он сбрасывается в «0», по обращению к регистру SSPBUF по чтению.

При приеме данных флаг обнаружения переполнения буфера SSPOV устанавливается в «1», когда 8 бит получены в SSPSR, а регистр SSPBUF не был считан после предыдущего приема.

Если пользователь совершает обращение по записи к регистру SSPBUF во время приема данных (SSPSR принимает данные), устанавливается флаг обнаружения конфликта записи WCOL и содержимое буфера остается неизменным (запись не выполняется).



**Рис. 64** Блок схема приема данных в режиме ведущего I2C

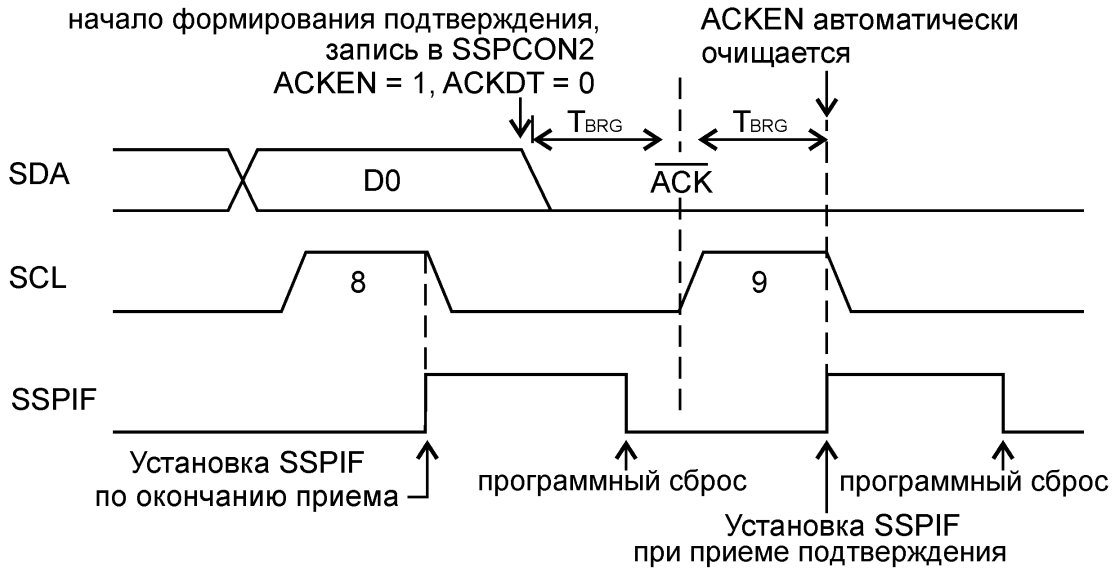


**Рис. 65** Прием данных в режиме ведущего I2C (7-разрядная адресация)

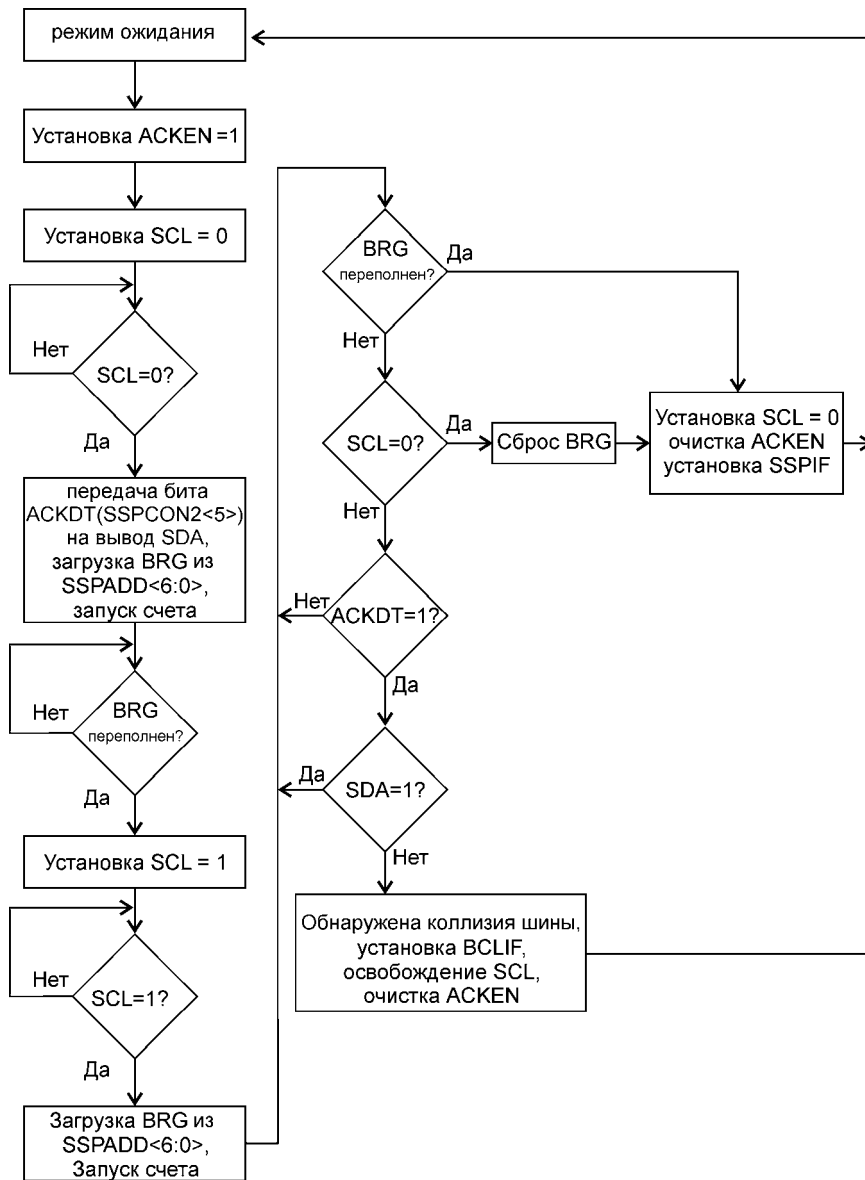
## Формирование подтверждения

Последовательность формирования сигнала подтверждения вводится в действие установкой в «1» разряда разрешения последовательности подтверждения ACKEN регистра SSPCON2<4>. Если этот разряд установлен, на вывод SCL выдается низкий уровень сигнала, а на линию SDA устанавливается содержимое разряда ACKDT. Если нужно подтвердить прием, разряд ACKDT должен быть равен «0». По окончании периода работы генератора скорости передачи (BRG) линия SCL «отпускается». Как только состояние линии SCL изменится с низкого уровня на высокий, генератор опять начнет отсчет периода. По окончании периода на линии SCL будет установлен низкий уровень, а бит ACKEN автоматически сброшен в «0», кроме того в этот момент устанавливается сигнал запроса на обработку прерывания, модуль SSP переходит в режим ожидания (см. Рис. 66).

Если пользователь осуществляет доступ по записи к регистру SSPBUF, когда последовательность передачи подтверждения уже в действии, устанавливается флаг обнаружения конфликта записи WCOL и содержимое буфера остается неизменным.



**Рис. 66** Формирование подтверждения



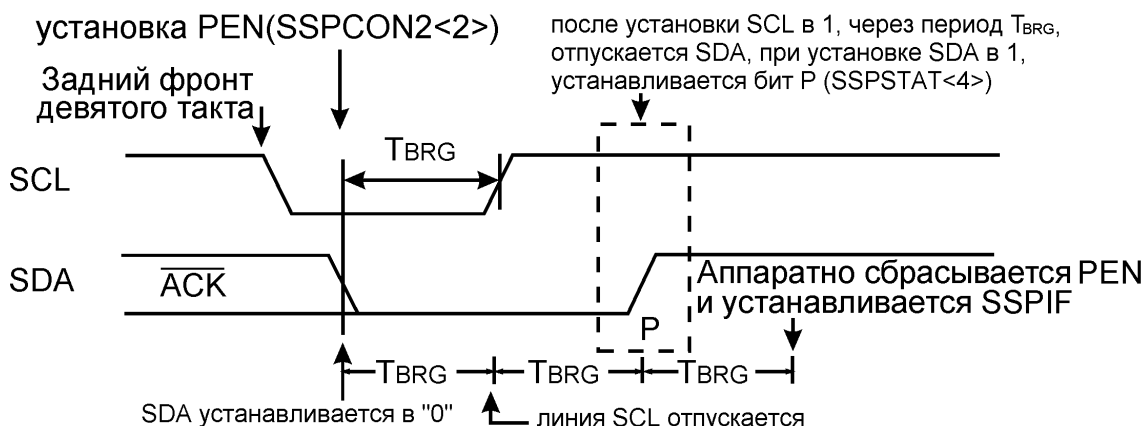
**Рис. 67** Блок схема формирования подтверждения

**Формирование флага СТОП**

Состояние «СТОП» может быть передано в конце приема/передачи данных, путем установки в «1» разряда разрешения формирования состояния «СТОП» PEN регистра SSPCON2<2>. По окончании приема или передачи данных, после прохождения заднего фронта 9 тактового сигнала, на SCL удерживается низкий уровень сигнала. Если разряд PEN установлен, модуль установит низкий уровень сигнала на линии SDA и запустит генератор скорости обмена для отсчета периода. По окончании отсчета периода линия SCL «отпускается» (выход SCL модуля переходит в «третье состояние»). После установки высокого уровня на SCL, и по истечении очередного периода генератора «отпускается» SDA. После появления на линии SDA высокого уровня, устанавливается флаг P регистра SSPSTAT. Далее разряд PEN аппаратно будет сброшен в «0» и установлен сигнал запроса на обработку прерывания SSPIF (см. Рис. 68).

Перед началом обмена по шине программное обеспечение должно определить, свободна ли шина, путем проверки флагов S и P регистра SSPSTAT. Линия может считаться свободной в случае если установлен флаг P (последним было обнаружено состояние «СТОП»).

Если пользователь осуществляет доступ по записи к регистру SSPBUF, когда последовательность передачи состояния «СТОП» уже в действии, устанавливается флаг обнаружения конфликта записи WCOL и содержимое буфера остается неизменным.



**Рис. 68** Формирование флага СТОП

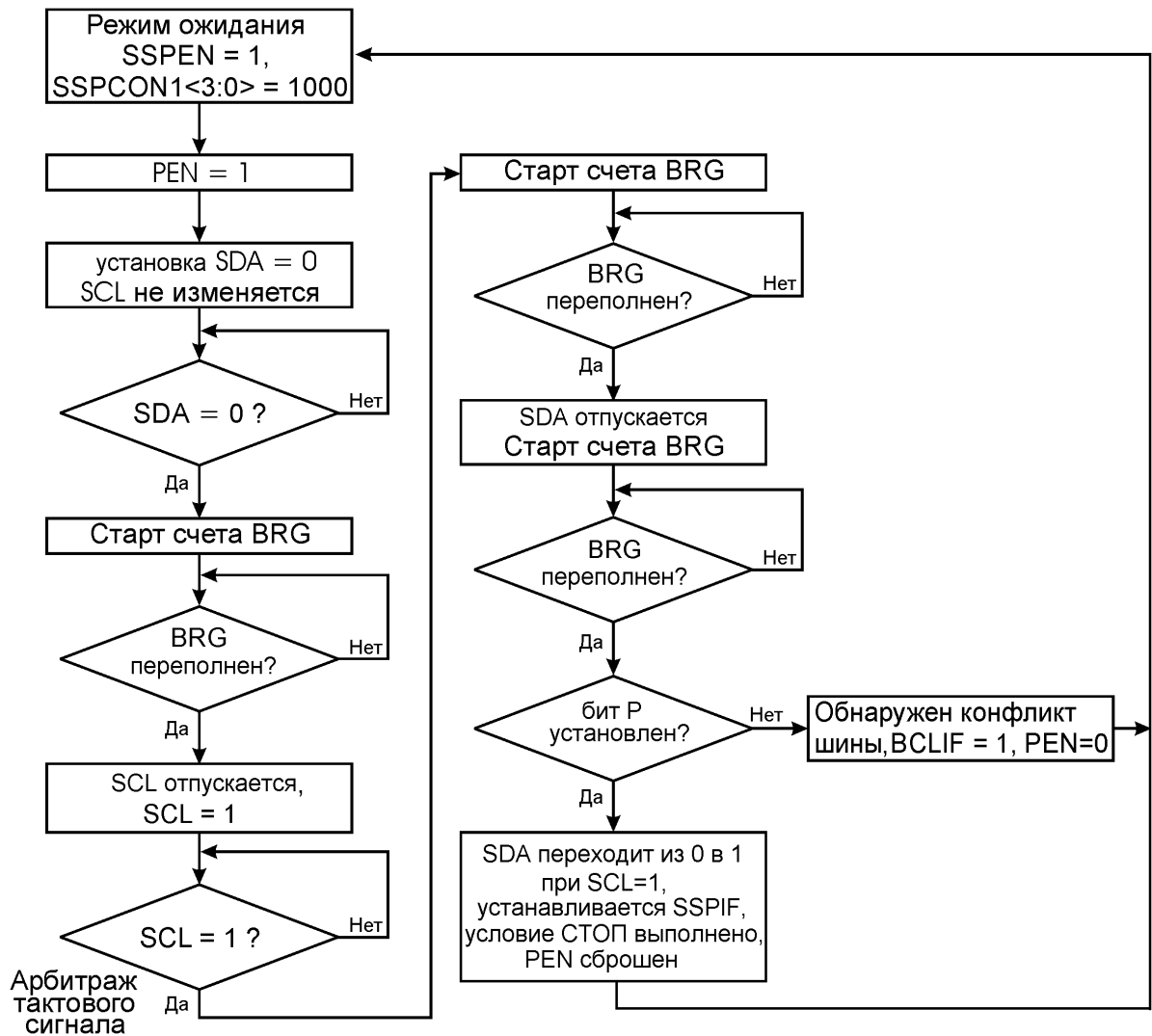


Рис. 69 Блок схема формирования флага СТОП

### Арбитраж линии тактового сигнала (SCL)

Арбитраж (проверка состояния) линии тактового сигнала производится ведущим каждый раз во время приема, передачи данных, формирования состояния «СТАРТ»/ «СТОП», при «отпускании» линии SCL (на линии SCL должен появиться высокий уровень). Генератор скорости обмена (BRG) приостанавливает свою работу до тех пор, пока на выводе SCL не появится высокий уровень сигнала. При появлении высокого уровня на SCL, BRG начинает отсчет периода (загружается значение SSPADD<6:0>). Такое функционирование позволяет гарантировать длительность высоко уровня сигнала на SCL всегда не менее периода работы генератора (BRG), даже если другое устройство удерживает тактовый сигнал (см. Рис. 70).

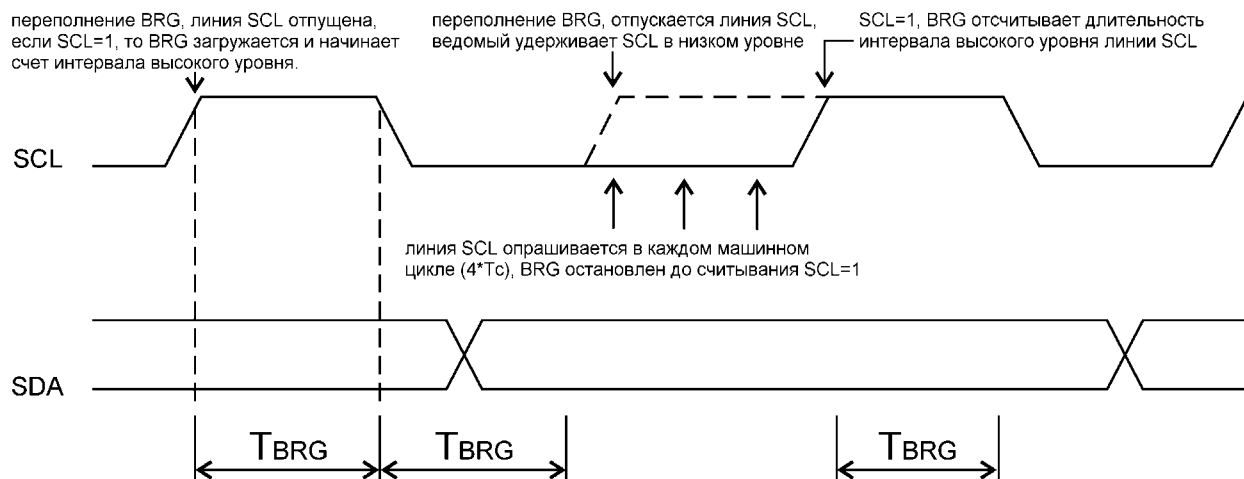


Рис. 70 Арбитраж линии тактового сигнала в режиме ведущий

## Функционирование в спящем режиме. Эффект сброса

Находясь в спящем режиме, модуль I2C может получать адрес и данные. При приеме «своего» адреса или при приеме байта данных, процессор выходит из спящего режима (если разрешена установка запроса на обработку прерывания от модуля СПП).

Любой сброс выключает модуль MSSP и прерывает текущий обмен данными.

## Режим работы при наличии на шине более одного ведущего, арбитраж и конфликт шины

Поддержка режима работы при наличии на шине более одного ведущего достигается за счет арбитража шины. При передаче адреса или данных ведущим, арбитраж линии SDA происходит за счет проверки состояния линии при передаче состояния логической «1». Если в этот момент состояние линии остается равным «0» (удерживается другим ведущим), то на шине имеет место конфликт. Обнаружив конфликт шины, ведущий устанавливает флаг прерывания (флаг обнаружения конфликта шины) BCLIF, прекращает текущую операцию и переводит модуль в режим ожидания (см. Рис. 71). Если при возникновении конфликта шины выполнялась передача данных, она прерывается, устанавливается бит BF, а линии SDA и SCL «отпускаются» в высокое состояние. В регистр SSPBUF могут быть записаны данные. После освобождения шины обмен данными может быть продолжен, путем формирования состояния «СТАРТ».

Если при возникновении конфликта, выполнялось формирование состояния «СТАРТ», «СТОП» или подтверждения (ACK), выполняемая операция обрывается, линии SDA и SCL «отпускаются», а соответствующий разряд управления регистра SSPCON2 сбрасывается в «0». После освобождения шины обмен данными может быть продолжен, путем формирования состояния «СТАРТ».

После обнаружения конфликта ведущий продолжает следить за состоянием шины, и при появлении состояния «СТОП» устанавливается сигнал запроса на обработку прерывания SSPIF.

Обращение по записи к регистру SSPBUF будет инициировать передачу данных, в не зависимости от состояния при котором был обнаружен конфликт.

Использование сигнала прерывания модуля при обнаружении состояний «СТАРТ» и «СТОП» позволяет определить состояние шины (свободна или занята). Управление шиной может быть перехвачено при установлении флага P. Конфликт шины при формировании состояния «СТАРТ» происходит, если:

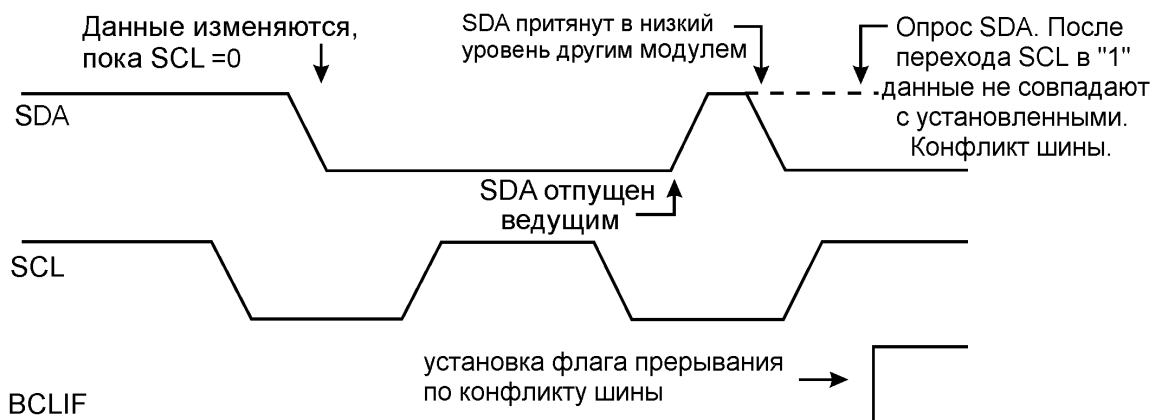
1. на SDA или SCL низкий уровень сигнала в начале формирования состояния «СТАРТ» (см. Рис. 72);
2. на SCL низкий уровень появляется раньше, чем на SDA (см. Рис. 73)

Во время формирования состояния «СТАРТ» проверяется состояние линий SDA и SCL.

Если: линия SDA имеет низкий уровень сигнала или вывод SCL имеет низкий уровень сигнала, то формирование состояния «СТАРТ» прекращается, и устанавливается флаг запроса на обработку прерывания BCLIF, и модуль СПП возвращается в состояние ожидания (см. Рис. 71).

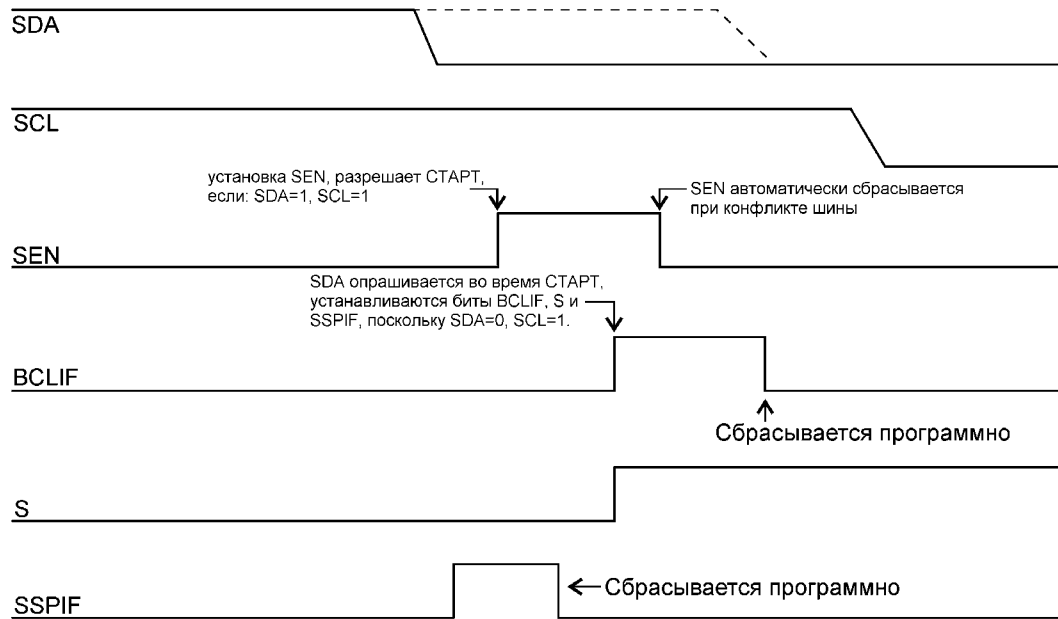
Формирование состояния «СТАРТ» начинается, когда SDA и SCL «отпущены» (находятся в высоком состоянии). Если на линии SDA высокий уровень сигнала, генератор скорости обмена загружается значение регистра SSPADD<6:0> и начинается отсчет периода. Если на линии SCL низкий уровень сигнала, в то время как на SDA высокий, происходит детектирования конфликта на шине, потому что предполагается, что другой ведущий пытается удерживать шину.

Если во время отсчета периода (BRG) на SDA появляется низкий уровень сигнала, генератор сбрасывается, а на SDA формируется низкий уровень раньше времени (см. Рис. 74). Если на SDA высокий уровень, низкий уровень формируется по окончании отсчета периода. Генератор скорости обмена в этом случае перезагружается и начинает отсчет нового периода. Если на линии SCL в этот момент времени детектируется низкий уровень, то такая ситуация не считается конфликтом на шине, это объясняется тем фактом, что два ведущих не могут одновременно формировать состояние «СТОП» на шине и кто-то из них захватит линию SDA первым. В конце периода (BRG), на линии SCL устанавливается низкий уровень.

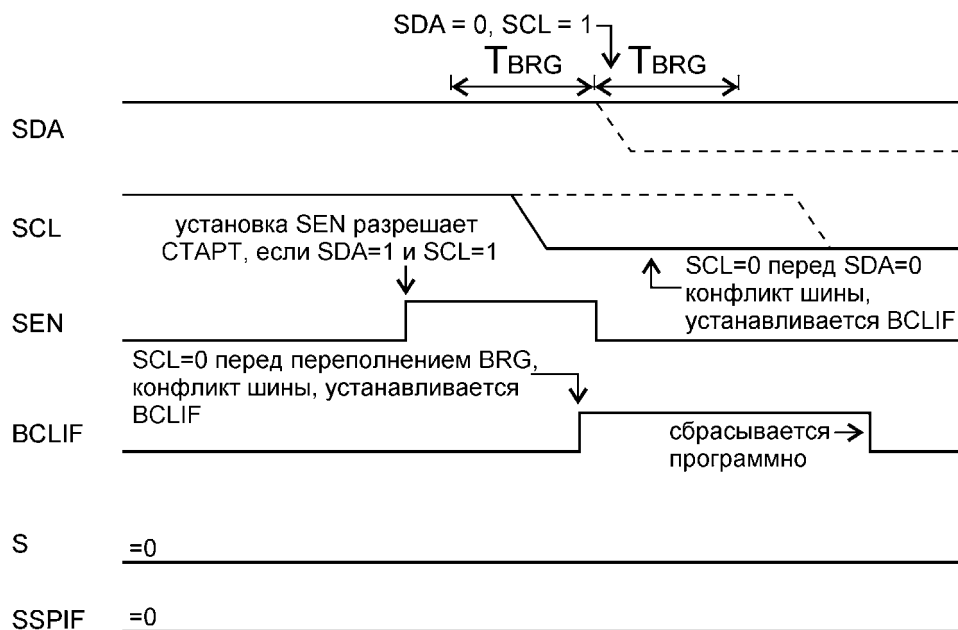


**Рис. 71** Конфликт шины при передаче данных и бита подтверждения

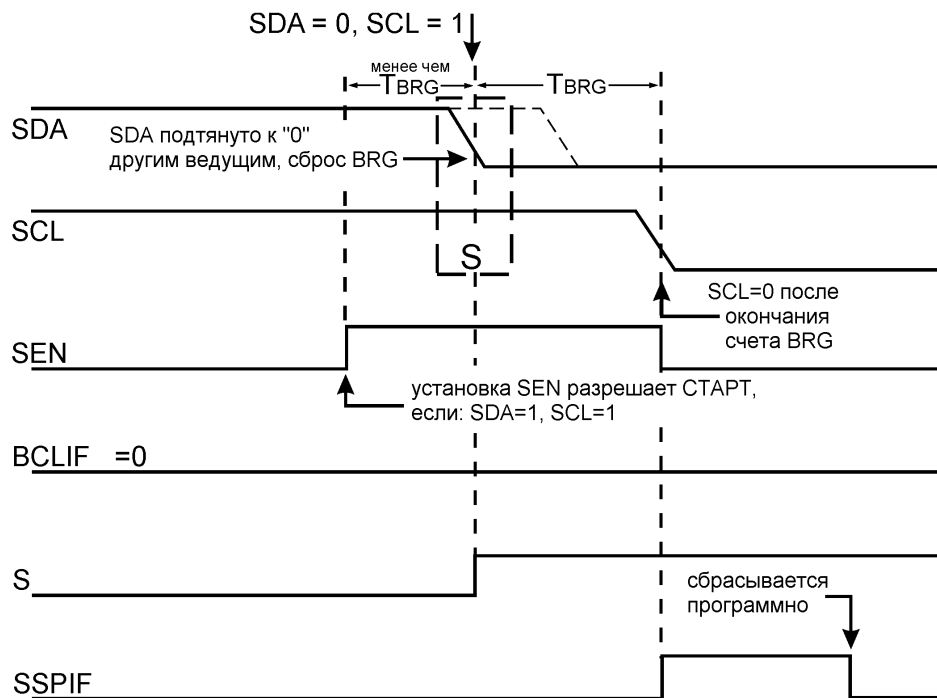




**Рис. 72** Конфликт шины во время формирования состояния «СТАРТ» (SCL=1)



**Рис. 73** Конфликт шины во время формирования состояния «СТАРТ» (SCL=0)



**Рис. 74** Сброс BRG при проверке линии SDA во время формирования состояния «СТАРТ»

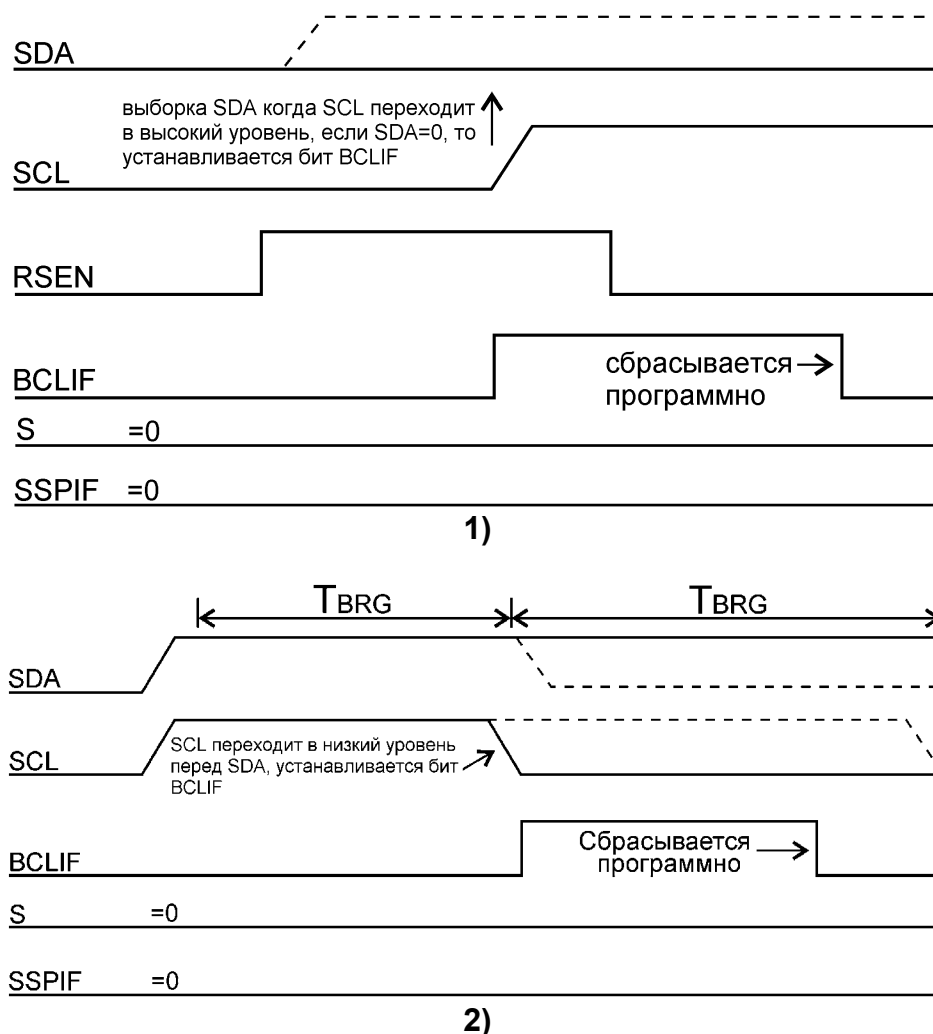
Конфликт на шине при формировании состояния «Повторный СТАРТ» происходит, если:

1. присутствует низкий уровень на SDA, когда SCL меняет свое состояние с низкого на высокий;
2. SCL переходит в состояние низкого уровня раньше чем SDA, указывая на ситуацию, что другой ведущий пытается передать данные.

После того как линия SDA «отпущена» ведущим и на линии появился высокий уровень, в генератор скорости передачи (BRG) загружается значение регистра SSPADD<6:0> и начинает отсчет периода. По окончании отсчета периода «отпускается» линия SCL и при появлении на ней высокого уровня проверяется состояние линии SDA. Наличие на SDA низкого уровня сигнала означает конфликт на шине, т. е. другой ведущий пытается передать данные. В противном случае (если на SDA высокий уровень) генератор (BRG) начинает отсчет нового периода. Ситуация при которой SDA переходит в низкий уровень до окончания отсчета, не признается как конфликт на шине, поскольку два ведущих не могут одновременно пытаться захватить линию SDA.

Если, однако, линия SCL меняет свое состояние на низкий уровень до окончания отсчета, при этом SDA сохраняет высокий уровень, значит произошел конфликт на шине, т. е. другой ведущий пытается передать данные.

Если по окончании отсчета периода генератором (BRG), на SCL и SDA высокие уровни, то SDA устанавливается низкий уровень, а генератор начинает отсчет нового периода. По окончании счета, независимо от состояния линии SCL, на ней устанавливается низкий уровень и формирования состояния «Повторный СТАРТ» считается выполненным (см. Рис. 75).

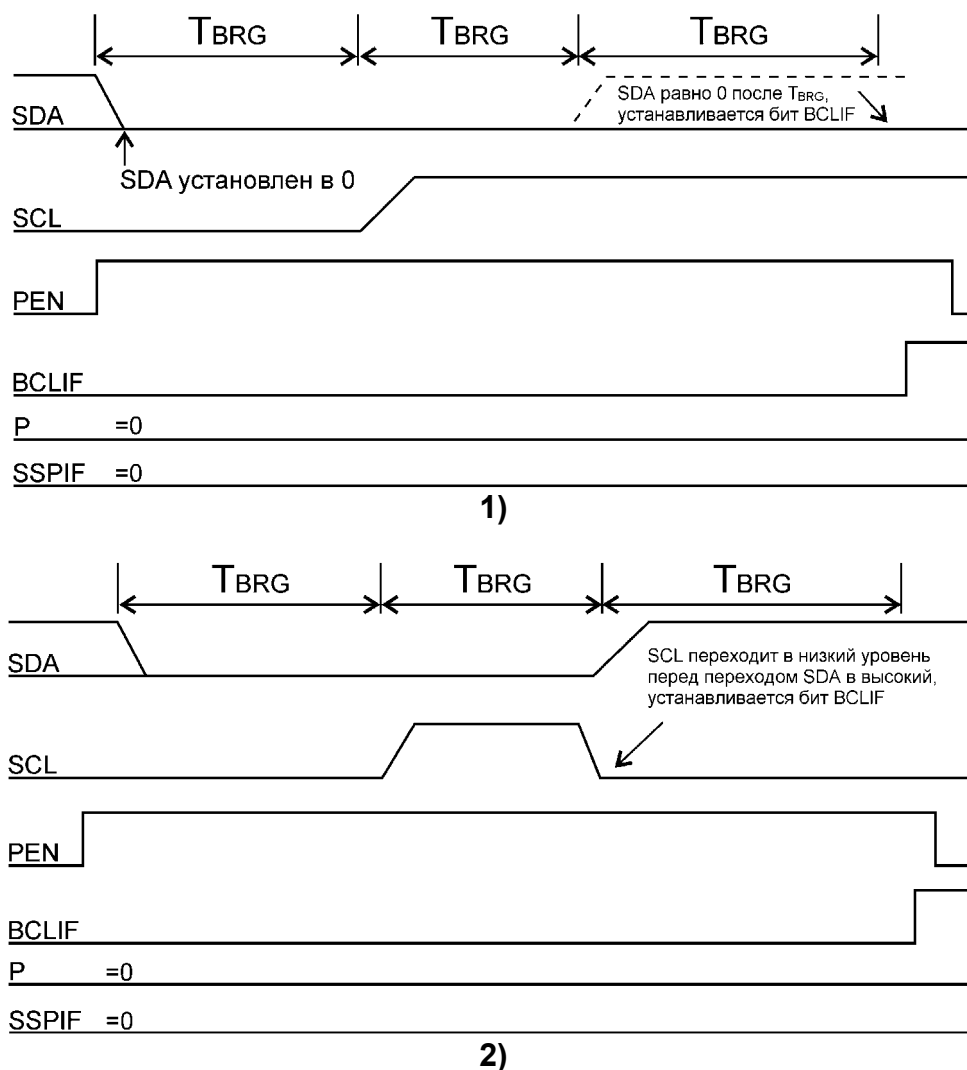


**Рис. 75** Конфликт шины во время формирования состояния «Повторный СТАРТ» (случай 1 и 2).

Конфликт на шине при формировании состояния «СТОП» происходит, если: после «отпускания» линии SDA и окончания отсчета периода генератором (BRG) на SDA по-прежнему остается низкий уровень сигнала.

после «отпускания» линии SCL сигнал на SCL остается в состоянии «низкий уровень» в тот момент, когда SDA меняет свое состояние с низкого на высокое.

Формирование состояния «СТОП» начинается с установки на линии SDA низкого уровня, после проверки линии SDA на наличие на ней низкого уровня, линия SCL «отпускается». После появления на линии SCL высокого уровня в генератор скорости обмена загружается значение регистра SSPADD<6:0> и начинается отсчет периода. По окончании отсчета SDA "отпускается", генератор (BRG) перезагружается и снова начинается отсчет периода, по окончании которого проверяется линия SDA. Если на линии обнаружен низкий уровень, такая ситуация означает конфликт на шине. Кроме того ситуация при которой на линии SCL обнаружен низкий уровень до того как линия SDA «отпущена», будет также признана конфликтом (см. Рис. 76).



**Рис. 76** Конфликт шины во время формирования состояния «СТОП» (случай 1 и 2)

## Подключение к шине I2C

Для стандартного режима I2C значение резисторов RP и RS зависит от следующих параметров:

- Напряжение питания. (Напряжение питания ограничивает минимальное значение сопротивления RP, из-за ограничения минимального тока стока при UOL max. Желаемый запас помехоустойчивости ограничивает максимальное значение RS.)
- Емкость шины. (Это полная емкость проводников, соединений и выводов. Эта емкость ограничивает максимальное значение RP из-за требований к длительности фронтов сигнала.)
- Количество устройств на шине (входной ток + ток утечки).

Находящийся в регистре SSPSTAT бит SMP включает управление длительностью фронтов сигналов на выводах в режиме I2C (ведущем или ведомом).

Устройства на шине I2C должны иметь один источник питания, к которому подключаются все подтягивающие резисторы.

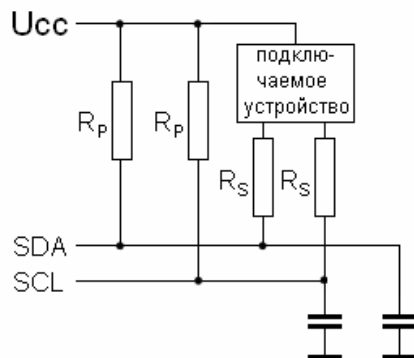


Рис. 77 Схема подключения к шине I2C

### Аналогово-цифровой преобразователь

Аналогово-цифровой преобразователь (АЦП) имеет 12 аналоговых входов. Входной аналоговый сигнал через коммутатор каналов поступает в модуль АЦП. Модуль АЦП преобразует напряжение методом последовательного приближения. Результат преобразования 10-ти разрядный. Положительный и отрицательный входы опорного напряжения могут быть выбраны программно или с выводов AUCC и AUSS, или с выводов PG3/AN0/ $U_{REF+}$  и PG2/AN1/ $U_{REF-}$ . АЦП может работать в спящем режиме микроконтроллера, при этом в качестве источника тактовых импульсов для АЦП должен быть выбран RC генератор. Модуль АЦП имеет 4 регистра:

- регистр результата, старший байт (ADRESH);
- регистр результата, младший байт (ADRESL);
- регистр управления 0 (ADCON0);
- регистр управления 1 (ADCON1).

Регистр ADCON0 (регистр 14-1) используется для настройки работы модуля АЦП. С помощью регистра ADCON1 (регистр 14-2) устанавливается, какие входы микроконтроллера будут использоваться модулем АЦП и в каком режиме. Входы настраиваются в качестве аналоговых входов (PG3 и PG2 могут быть входами опорного напряжения) или цифровых входов/выходов.

**Таблица 40**

Регистр 14-1: Регистр ADCON0 (адрес: 14h, банк 5)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0
CHS3	CHS2	CHS1	CHS0	—	GO/DONE	—	ADON
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7-4</b>		<b>CHS3 - CHS0:</b> биты выбора аналогового канала: 0000 = канал 0, (AN0) 0001 = канал 1, (AN1) 0010 = канал 2, (AN2) 0011 = канал 3, (AN3) 0100 = канал 4, (AN4) 0101 = канал 5, (AN5) 0110 = канал 6, (AN6) 0111 = канал 7, (AN7) 1000 = канал 8, (AN8) 1001 = канал 9, (AN9) 1010 = канал 10, (AN10) 1011 = канал 11, (AN11) 11xx = зарезервировано					
<b>бит 3</b>		Не используется, читается как «0».					
<b>бит 2</b>		<b>GO/DONE:</b> бит состояния модуля АЦП. Если ADON = 1, т.е. АЦП включен: 1 = модуль АЦП выполняет преобразование (установка бита вызывает начало преобразования, аппаратно сбрасывается по завершению преобразования), 0 = в модуле АЦП преобразования нет.					
<b>бит 1</b>		Не используется, читается как «0».					
<b>бит 0</b>		<b>ADON:</b> включение модуля АЦП. 1 = модуль АЦП включен, 0 = модуль АЦП выключен (снижает общий ток потребления микроконтроллера).					

Обозначения:

R = бит для чтения; W = бит с возможностью записи; U = бит не реализован, читается как 0;

-n = значение бита после сброса по включению питания: 1 = установлен; 0 = сброшен; x = значение не известно.

**Таблица 41**

Регистр 14-2: Регистр ADCON1 (адрес: 15h, банк 5)

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCS1	ADCS0	ADFM	—	PCFG3	PCFG2	PCFG1	PCFG0
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7, 6</b>		<b>ADCS1:ADCS0:</b> выбор источника импульсов преобразования модуля АЦП: 00 = FC/8 01 = FC/32 10 = FC/64 11 = источник импульсов внутренний RC генератор (FRC)					

<b>бит 5</b>	<b>ADFM:</b> Формат сохранения 10-битного результата: 1 = правое выравнивание, 6 старших бит ADRESH читаются как «0». 0 = левое выравнивание, 6 младших бит ADRESL читаются как «0».
<b>бит 4</b>	Не используется, читается как «0».
<b>бит 3-1</b>	<b>PCFG3:PCFG1:</b> биты управления конфигурацией порта АЦП.
<b>бит 0</b>	<b>PCFG0:</b> Выбор источника опорного напряжения: 1 = опорное напряжение берется с выводов $U_{REF+}$ и $U_{REF-}$ . 0 = опорное напряжение берется с выводов AUDD и AUSS.

Обозначения:

R = бит для чтения; W = бит с возможностью записи; U = бит не реализован, читается как 0;

-n = значение бита после сброса по включению питания: 1 = установлен; 0 = сброшен; x = значение не известно.

**Таблица 42**

<b>PCFG3: PCFG0</b>	<b>AN11</b>	<b>AN10</b>	<b>AN9</b>	<b>AN8</b>	<b>AN7</b>	<b>AN6</b>	<b>AN5</b>	<b>AN4</b>	<b>AN3</b>	<b>AN2</b>	<b>AN1</b>	<b>AN0</b>
<b>000x</b>	A	A	A	A	A	A	A	A	A	A	A	A
<b>001x</b>	A	A	A	A	D	A	A	A	A	A	A	A
<b>010x</b>	A	A	A	A	D	D	A	A	A	A	A	A
<b>011x</b>	A	A	A	A	D	D	D	A	A	A	A	A
<b>100x</b>	A	A	A	A	D	D	D	D	A	A	A	A
<b>101x</b>	D	A	A	A	D	D	D	D	D	A	A	A
<b>110x</b>	D	D	A	A	D	D	D	D	D	D	A	A
<b>111x</b>	D	D	D	D	D	D	D	D	D	D	D	D

Обозначения: A - аналоговый вход, D - цифровой вход/выход.

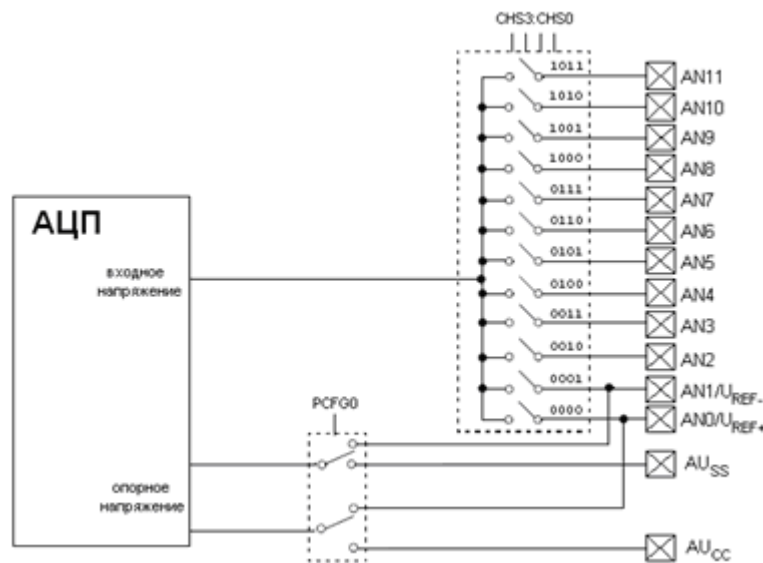
Когда преобразование завершено, 10-ти разрядный результат преобразования записывается в регистры ADRESH:ADRESL, бит GO/DONE (ADCON0<2>) сбрасывается и устанавливается флаг запроса прерывания ADIF. Блок-схема модуля АЦП показана на Рис. 78.

Рекомендованная последовательность действий для работы с АЦП:

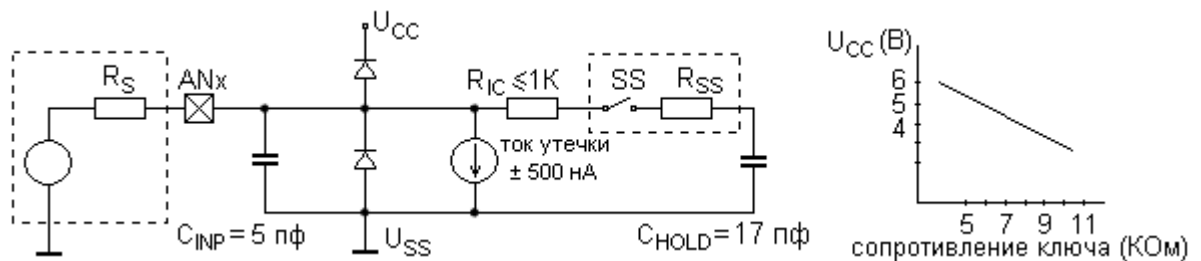
1. Настроить модуль АЦП: выбрать аналоговые входы, источник опорного напряжения, цифровые входы/выходы (ADCON1); выбрать источник импульсов преобразования (ADCON1); выбрать входной канал АЦП (ADCON0); включить модуль АЦП (ADCON0). С помощью регистров DDR все выходы, которые будут использоваться как аналоговые входы, должны быть настроены на вход.
2. Настроить прерывание от модуля АЦП (если необходимо): сбросить флаг запроса прерывания по окончании преобразования ADIF; установить бит ADIE (маска прерывания от АЦП); сбросить бит GLINTD.
3. Выдержать паузу, необходимую для зарядки конденсатора CHOLD (до уровня входного напряжения).
4. Начать аналогово-цифровое преобразование, установив бит GO/DONE (ADCON0).
5. Ожидать окончания преобразования (длительность преобразования  $12 \cdot TAD$ ), т.е. ожидать или сброс бита GO/DONE, или (если разрешено) прерывание по окончании преобразования.

6. Считать результат преобразования из регистров ADRESH:ADRESL и сбросить бит ADIF (если необходимо).
7. Для запуска следующего преобразования необходимо выполнить шаги, начиная с пункта 1, 2 или 3. Время преобразования одного бита определяется как время TAD. Время ожидания перед следующим преобразованием должно быть не менее  $2 \cdot TAD$ .

Для обеспечения необходимой точности преобразования конденсатор CHOLD должен успевать заряжаться до уровня входного напряжения. Схема модели входа АЦП приведена на Рис. 79. Выходное сопротивление  $R_S$  и сопротивление ключа выборки  $R_{SS}$  влияют на время зарядки емкости CHOLD. Величина сопротивления ключа выборки ( $R_{SS}$ ) зависит от напряжения питания  $U_{CC}$  (см. Рис. 79). Значение выходного сопротивления источника аналогового сигнала  $R_S$  влияет на значение входного смещения напряжения из-за тока утечки вывода. Выходное сопротивление  $R_S$  должно быть не более 10 кОм. При его уменьшении время заряда емкости  $C_{HOLD}$  уменьшается. После выбора аналогового входного канала до начала преобразования, должно пройти определенное время для заряда емкости  $C_{HOLD}$ . Для расчета этого времени воспользуются уравнением, описанным далее. Уравнение дает ошибку в пределах  $\frac{1}{2} L_{Sb}$  (шага АЦП). Ошибка в  $\frac{1}{2} L_{Sb}$ , это максимальная погрешность, позволяющая работать модулю АЦП с необходимой точностью вычисления.



**Рис. 78** Блок-схема модуля АЦП



**Рис. 79** Схема аналогового входа АЦП



### Вычисление минимального времени задержки

$T_{ACQ}$  = время установки схемы + время зарядки  $C_{HOLD}$  + температурный коэффициент

$$T_{ACQ} = T_{AMP} + T_{CHOLD} + T_{COFF}$$

### Уравнение вычисления минимального времени заряда емкости

$C_{HOLD}$ .

$$U_{HOLD} = ( U_{REF} - ( U_{REF} / 2048 ) ) * ( 1 - e^{(-T_{CHOLD} / C_{HOLD} (R_{IC} + R_{SS} + R_S))} )$$

или:  $T_{CHOLD} = C_{HOLD} (R_{IC} + R_{SS} + R_S) \ln(1/2047)$

Для  $C_{HOLD} = 17$  пФ,  $R_S = 10$  КОм, ошибки  $\frac{1}{2}$  LSB,  $U_{CC} = 5$  В ( $R_{SS} = 6$  КОм),  $T_{emp} = 50^\circ\text{C}$ ,  $U_{HOLD} = 0$ :

$$T_{CHOLD} = -17 * ( 1 + 6 + 10 ) * \ln(1/2047) = 2.2 \text{ мкс}$$

$$T_{ACQ} = 2 \text{ мкс} + 2.2 \text{ мкс} + [(Temp - 25^\circ\text{C}) (0.05 \text{ мкс}/^\circ\text{C})] = 5.45 \text{ мкс}$$

#### Примечание:

1. Опорное напряжение  $U_{REF}$  не влияет на уравнение.
2. Конденсатор  $C_{HOLD}$  после каждого преобразования не разряжается.
3. После того как преобразование завершено необходимо программно обеспечить задержку не менее  $2 * T_{AD}$ , прежде чем начать следующее преобразование. В течение этого времени конденсатор  $C_{HOLD}$  не подключен к выбранному входному каналу АЦП.

Время преобразования АЦП зависит от частоты тактовых импульсов преобразования ( $T_{AD}$ ). Для 10-ти разрядного аналогово-цифрового преобразования требуется время  $12 * T_{AD}$ . Источники импульсов тактирования АЦП выбираются программно. Для достоверного аналогово-цифрового преобразования должен быть выбран источник импульсов, обеспечивающий время  $T_{AD}$  не менее 1.6 мкс. Возможны четыре варианта:

- $F_C/8$  ( $T_{AD}=8 * T_C$ ) - используется при тактовой частоте микроконтроллера до 5 МГц.
- $F_C/32$  ( $T_{AD}=32 * T_C$ ) - используется при тактовой частоте микроконтроллера до 20 МГц.
- $F_C/64$  ( $T_{AD}=64 * T_C$ ) - используется при тактовой частоте микроконтроллера до 33 МГц.
- Внутренний RC генератор - если тактовая частота микроконтроллера больше 1 МГц, то RC генератор рекомендуется использовать только в SLEEP режиме.

Регистры ADCON1 и DDR управляют настройкой выводов АЦП. Если выводы микросхемы конфигурируются как аналоговые входы, то необходимо установить в единицу соответствующие биты в регистре DDR. Если соответствующий бит сброшен, то вывод настраивается как цифровой выход. Модуль АЦП работает независимо от установленного состояния битов CHS2:CHS0 и битов регистра DDR. При считывании порта, результат чтения разрядов, соответствующих выводам, настроенным как аналоговые входы, будет всегда равен нулю. Аналоговые уровни на цифровом входе не влияют на корректность преобразования. Аналоговый сигнал, подаваемый на цифровой вход

(включая AN0-AN11), влияет на ток потребления входного буфера, это приводит к повышению энергопотребления микроконтроллера.

Для запуска аналогово-цифрового преобразования необходимо включить АЦП и установить в «1» бит GO/DONE. Эти биты должны устанавливаться разными командами. После определенного времени на выбранном канале начнется преобразование (см. Рис. 80), длительность первого такта от TCY до TAD. Сброс в «0» бита GO/DONE во время преобразования останавливает преобразование. При этом регистры ADRESH:ADRESL не изменяют своего значения. После досрочного завершения преобразования необходимо обеспечить временную задержку  $2 \cdot T_{AD}$ , т.к. выбранный канал не подключен к внутреннему конденсатору в течение этого времени.

10-ти разрядный результат преобразования АЦП записывается в 16-ти разрядный регистр ADRESH:ADRESL. Модуль АЦП позволяет преобразовывать 10-ти разрядное значение в 16-ти разрядное, выравниванием влево или вправо (см. Рис. 81). Выбор формата преобразования осуществляется программно. Не задействованные биты имеют значение «0». При выключенном АЦП значения регистров ADRESH:ADRESL не изменяются и регистры могут быть использованы как универсальные 8-разрядные регистры.



**Рис. 80** Работа модуля АЦП по тактам

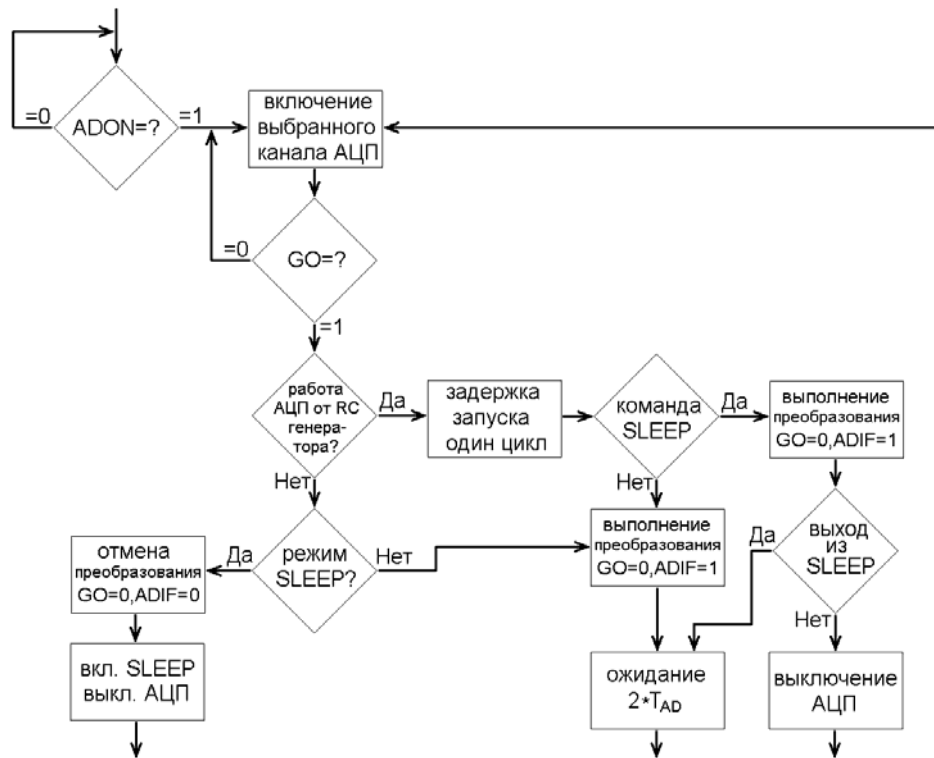
**Выравнивание вправо:**

ADRESH								ADRESL								
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	10-ти разрядный результат преобразования										

**Выравнивание влево:**

ADRESH								ADRESL							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
10-ти разрядный результат преобразования										0	0	0	0	0	0

**Рис. 81** Выравнивание результата преобразования АЦП



**Рис. 82** Блок-схема работы АЦП

Модуль АЦП может работать в спящем режиме микроконтроллера в случае если источником тактовых импульсов для АЦП будет внутренний RC генератор (ADCS1:ADCS0=11). Если выбран этот режим тактирования, то модуль АЦП, прежде чем начать преобразование, произведет задержку в течение одного цикла команд. Это позволяет программе выполнить команду SLEEP, для уменьшения цифрового шума во время преобразования (команда SLEEP должна следовать непосредственно за командой, устанавливающей бит GO/DONE). После завершения преобразования бит GO/DONE сбрасывается, результат преобразования записывается в регистры ADRESH:ADRESL. Если разрешено прерывание от АЦП, то микроконтроллер будет выведен из режима SLEEP. Если прерывание запрещено, то после преобразования модуль АЦП будет выключен, хотя бит ADON останется установленным в «1». Если был выбран не RC генератор, а другой источник тактовых импульсов для АЦП, то выполнение команды SLEEP прервет преобразование и выключит модуль АЦП, уменьшив ток потребления микроконтроллера, оставив ADON=1.

При сбросе микроконтроллера модуль АЦП выключается и останавливается преобразование (если оно было начато). Регистры ADRESH:ADRESL не меняют значения, а после сброса по включению питания их значение не определено.

Предпочтительно использовать АЦП с  $T_{AD}$  не больше 8 мкс, но не меньше рекомендованного нижнего предела. В системах с низкой рабочей частотой и в случае использования АЦП в режиме SLEEP микроконтроллера, источником тактового сигнала должен быть встроенный RC генератор. В других случаях используется тактовый сигнал от основного тактового генератора. Использование тактового сигнала от основного генератора позволяет снизить влияние шумов от переключения внутренних вентилях, т.к. переключение логики АЦП происходит синхронно с другими устройствами, что невозможно при использовании

встроенного RC генератора. Если каналы цифрового ввода/вывода постоянно активны, потеря точности из-за шумов при переключении может быть значительной. При работе АЦП в SLEEP режиме отсутствуют цифровые шумы, т.к. другие узлы микроконтроллера остановлены, поэтому точность преобразования получается высокой.

Если значение входного напряжения АЦП превышает на 0.3 В величину питающих напряжений ( $U_{SS}$  и  $U_{CC}$ ), то точность преобразования выйдет за пределы значений, оговоренных в спецификации.

Для сглаживания пульсаций входного сигнала на вход АЦП может добавляться внешняя RC цепочка. Значение сопротивления  $R_S$  должно выбираться, чтобы общее сопротивление источника сигнала было в пределах рекомендованной величины 10 кОм. Любой внешний компонент, подключенный к аналоговому входу, должен иметь низкий ток утечки через выводы.

## **Специальные модули микроконтроллера**

### **Регистры конфигурации микроконтроллера**

Регистры конфигурации микроконтроллера находятся в памяти программ и записываются при программировании микроконтроллера (смотрите спецификацию по программированию).

Запись битов конфигурации производится побитно, по соответствующим этим битам адресам слов (см. Таблица 43). Чтение битов PM1, PM0, WDTPS1, WDTPS0, FOSC1 и FOSC0 возможно по любому адресу в диапазоне FE00h-FE07h, а битов PM2 и BODEN по любому адресу в диапазоне FE08h-FE0Fh.

Значения битов конфигурации дублируются в ПЗУ программ по адресам 7FF0h-7FFFh. Значению бита =0 соответствует значение ячейки ПЗУ равное FFFFh, а значению бита =1 - значение ячейки равное 0000h. Соответствие адресов ячеек ПЗУ битам конфигурации указано в таблице. Область памяти программ 7FF0h-7FFFh доступна только для чтения.

В микроконтроллере с масочным ПЗУ область памяти 7FF0h-7FFFh доступна для размещения кодов программы, а чтение регистров конфигурации не реализовано.

#### **Примечание:**

Если в ячейки с адресами 7FF4h, 7FF6h и 7FFFh записано значение FFFFh, то при «сбросе» микроконтроллер перейдет в режим защиты кода программ, т.е. считывание и стирание памяти программ заблокируется. Для предотвращения этого необходимо произвести стирание ячеек памяти до поступления сигнала «сброс».

Режим микроконтроллера с защитой кодов программы (режим «защищенного микроконтроллера») позволяет защитить содержимое внутренней памяти программ микроконтроллера от считывания или модификации внешним устройством (программатором). После установки этого режима блокируется доступ программатора к внутренней памяти программ, т.е. выполнение операций стирания, чтения, верификации и записи памяти программ, а также регистров конфигурации, становится не возможным. Микроконтроллер, находящийся в

«защищенном» режиме, игнорирует все команды, поступающие от программатора, и не отвечает на них. Программа управления программатором в этом случае не может определить наличие программируемого микроконтроллера и выдает соответствующую ошибку.

Установка режима «защищенного микроконтроллера» не влияет на выполнение команд микроконтроллера, осуществляющих чтение/запись таблиц в памяти программ (TABLRD и TABLWT). Установка режима «защищенного микроконтроллера» не влияет на функционирование программы, записанной в память программ микроконтроллера.

Перевод микроконтроллера в режим «защищенного микроконтроллера» производится записью соответствующей комбинации битов в регистры конфигурации микроконтроллера (смотрите таблицу 43).

**Внимание!**

**Перед программированием микроконтроллера проверьте правильность установленной конфигурации микроконтроллера. Установка режима «защищенного микроконтроллера» заблокирует возможность изменения содержимого внутренней памяти программ микроконтроллера и его конфигурации.**

**Таблица 43**

Регистры конфигурации микроконтроллера

	<b>FE0Fh</b>	<b>FE0Eh</b>					
бит 15 - 8	бит 7	6	5	4	3	2	1
-	<b>PM2</b>	<b>BODEN</b>	-	-	-	-	-

		<b>FE06h</b>		<b>FE04h</b>	<b>FE03h</b>	<b>FE02h</b>	<b>FE01h</b>
бит 15 - 8	бит 7	6	5	4	3	2	1
-	-	<b>PM1</b>	-	<b>PM0</b>	<b>WDTPS1</b>	<b>WDTPS0</b>	<b>FOSC1</b>

<p><b>PM2:</b> бит 7, адрес FE0Fh (ячейка ПЗУ: 7FFFh)</p> <p><b>PM1:</b> бит 6, адрес FE06h (ячейка ПЗУ: 7FF6h)</p> <p><b>PM0:</b> бит 4, адрес FE04h (ячейка ПЗУ: 7FF4h)</p>	<p><b>PM2, PM1, PM0</b> - биты выбора режима микроконтроллера:                  111 = режим микропроцессора                  110 = режим микроконтроллера                  101 = расширенный режим микроконтроллера                  000 = режим микроконтроллера с защитой кодов программы (т.е. запрет считывания содержимого внутренней памяти программ)</p>
<p><b>BODEN:</b> бит 6, адрес FE0Eh (ячейка ПЗУ: 7FFEh)</p>	<p><b>BODEN</b> - включение схемы сброса по снижению напряжения питания:                  1 = схема включена                  0 = схема выключена</p>

<b>WDTPS1:</b> бит 3, адрес FE03h (ячейка ПЗУ: 7FF3h) <b>WDTPS0:</b> бит 2, адрес FE02h (ячейка ПЗУ: 7FF2h)	<b>WDTPS1, WDTPS0</b> - выбор предделителя «сторожевого таймера»: 11 = «сторожевой таймер» включен, предделитель = 1 10 = «сторожевой таймер» включен, предделитель = 256 01 = «сторожевой таймер» включен, предделитель = 64 00 = «сторожевой таймер» выключен, работает как 16-ти разрядный таймер переполнения
<b>FOSC1:</b> бит 1, адрес FE01h (ячейка ПЗУ: 7FF1h) <b>FOSC0:</b> бит 0, адрес FE00h (ячейка ПЗУ: 7FF0h)	<b>FOSC1, FOSC0</b> - выбор режима тактового генератора: 11 = EC – режим подачи внешнего тактового сигнала 10 = XT – генератор с внешним кварцевым или керамическим резонатором (частота от 2МГц до 33 МГц) 01 = RC генератор с частотой до 4 МГц 00 = LF – генератор с внешним низкочастотным кварцевым резонатором ( $\leq 2$ МГц)

Обозначение:

- = зарезервировано, читается как 0.

### Внутрисхемное программирование микроконтроллера

Для программирования внутренней FLASH памяти микроконтроллера используется последовательный интерфейс. Для этого задействуются семь выводов микросхемы (см. Таблица 44). Это позволяет производить программирование микроконтроллеров установленных на печатную плату. Подробнее смотрите в спецификации по программированию.

**Таблица 44**

Выводы, используемые для программирования

Обозначение	В режиме программирования		
	Назначение	Тип	Описание
PA4/RX1/DT1	DT	I/O	последовательные данные
PA5/TX1/CK1	CK	I	последовательные тактовые импульсы
PA1/T0CLK	OSCI	I	вход синхронизации микроконтроллера
TEST	TEST	I	вход выбора тестового режима
MCLR/UPP	MCLR/UPP	P	напряжение программирования
UDD	UDD	P	напряжение питания
USS	USS	P	«земля»

### Сторожевой таймер

Сторожевой таймер (WDT) предназначен для восстановления микроконтроллера при сбоях или сброса устройства во время его нахождения в режиме SLEEP. Для повышения надежности сторожевой таймер имеет собственный RC генератор. Он работает даже при отсутствии тактовой частоты микроконтроллера. Режим сторожевого таймера программируется битами

конфигурации в регистрах конфигурации микроконтроллера. Во время нормальной работы WDT должен очищаться через определенные интервалы времени. Это время должно быть меньше, чем минимальное время переполнения WDT, в противном случае переполнение WDT произведет сброс устройства.

Номинальный период сторожевого таймера (с предделителем = 1) составляет около 12 мс. Это время зависит от температуры и напряжения питания. Для увеличения периода таймера можно включить предделители с большим коэффициентом деления. Сторожевой таймер и его предделитель сбрасываются командами CLRWDT и SLEEP, сигналом «сброса» и при выходе из режима SLEEP по прерыванию. Таймер начинает счет сразу же после окончания сигнала «сброс». Бит TO в регистре CPUSTA будет сброшен при переполнении сторожевого таймера.

Если сторожевой таймер включен в режиме обычного таймера, то на него подаются импульсы с генератора тактовой частоты микроконтроллера. Время переполнения составляет 65536 тактов TC. При переполнении сбрасывается бит TO в регистре CPUSTA, но устройство не сбрасывается. Команда CLRWDT устанавливает этот бит. Регистры сторожевого таймера и его предделителя не доступны для чтения/записи. Таймер в этом режиме останавливается в режиме SLEEP.

### Режим энергосбережения (SLEEP)

Микроконтроллер переходит в режим энергосбережения при выполнении команды SLEEP. При этом сбрасывается сторожевой таймер и его предделитель (если они включены), бит PD сбрасывается, бит TO устанавливается (регистр CPUSTA), выключается генератор тактовой частоты микроконтроллера, порты ввода/вывода сохраняют свое состояние.

Следующие события могут вывести микроконтроллер из режима SLEEP:

- Сброс при включении или сброс при снижении питания.
- Подача сигнала сброс на внешний вход сброса MCLR.
- Сброс от сторожевого таймера (если он включен).
- Прерывание с вывода RA0/INT, прерывание от изменения сигналов на порте B, прерывание с RA1/T0CLK, или некоторые периферийные прерывания (от модуля захвата (регистрации событий), от приемника и передатчика USART в синхронном ведомом режиме, завершение преобразования АЦП, завершение приема/передачи SPI в ведомом режиме, от приемника I2C в ведомом режиме).

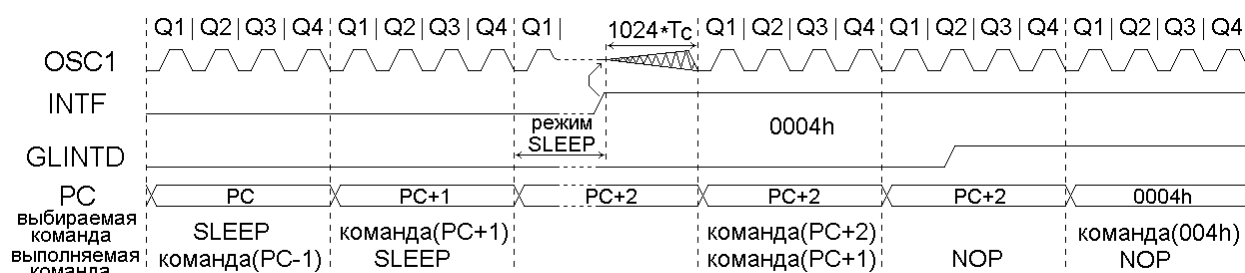
Другие периферийные устройства не могут генерировать прерывания в режиме SLEEP, так как выключен тактовый генератор микроконтроллера.

Любой сигнал «сброса» вызовет сброс устройства. Прерывания продолжат выполнение программы. Биты TO и PD в регистре CPUSTA могут быть использованы для определения причины сброса устройства. Устанавливаемый при включении бит PD, сбрасывается при переходе в режим SLEEP. Бит TO сбрасывается при переполнении сторожевого таймера.

При выполнении команды SLEEP, предварительно выбирается следующая команда (PC+1). Чтобы пробудить устройство прерыванием, должен быть установлен соответствующий бит разрешения прерывания. Это происходит независимо от состояния бита GLINTD. Если бит GLINTD установлен, устройство продолжает выполнение программы. Если бит GLINTD сброшен, то устройство

выполняет команду, следующую за SLEEP, и затем переходит к адресу вектора прерывания. В случаях, где исполнение команды после SLEEP нежелательно, необходимо ставить NOP после команды SLEEP. Сторожевой таймер сбрасывается при выходе устройства из режима SLEEP, независимо от источника пробуждения.

Если установлен XT или LF режим генератора тактовой частоты микроконтроллера, то при выходе из SLEEP происходит запуск «таймера запуска генератора», который будет держать устройство в состоянии «сброса» в течение 1024 тактов TC.



**Рис. 83** Выход из режима SLEEP по прерыванию

### Примечания:

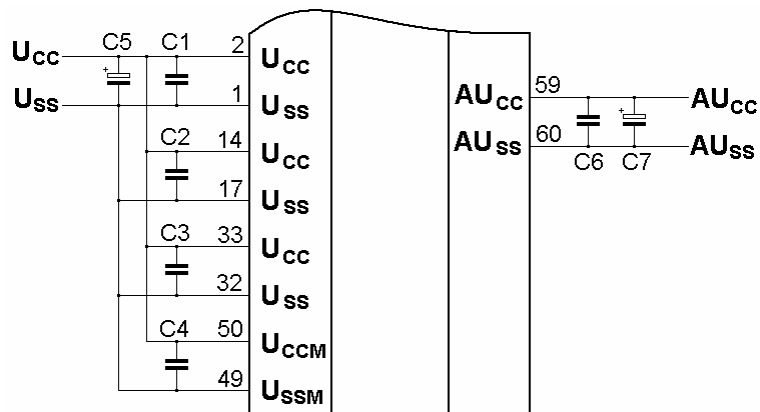
1. Режим генератора XT или LF. Задержки запуска генератора ( $1024 \cdot TC$ ) не будет в режиме RC генератора.
2. Если  $GLINTD=0$ , процессор переходит к программе обработки прерываний, если  $GLINTD=1$ , то продолжится выполнение программы.

### Схема подключения напряжения питания

Схема представлена на Рис. 84. Линии питания  $U_{CC}$  (положительный вывод напряжения питания цифровой части микроконтроллера) и  $U_{SS}$  («земляной» вывод напряжения питания микроконтроллера) подключаются к соответствующим линиям питания на печатной плате. Линии питания  $AU_{CC}$  (положительный вывод напряжения питания АЦП микроконтроллера) и  $AU_{SS}$  («земляной» вывод напряжения питания АЦП микроконтроллера) для снижения влияния помех рекомендуется подключать к соответствующим аналоговым линиям питания на печатной плате (если таковые имеются). Потенциалы положительных выводов напряжения питания  $U_{CC}$  и  $AU_{CC}$  должны быть одинаковы. Аналогично потенциалы «земляных» выводов  $U_{SS}$  и  $AU_{SS}$  также должны быть одинаковыми.

Рекомендуемые значения номиналов конденсаторов, установленных в цепи питания: C1-C4 и C6 не менее 0.1 мкФ., C5 и C7 - не менее 47 мкФ.





**Рис. 84** Схема подключения напряжения питания

### Система команд

Микроконтроллер поддерживает 58 команд (см. Таблица 45). Все команды 16-ти разрядные. Команды выполняются за один цикл, состоящий из четырех периодов тактовой частоты, за исключением выполняемых за два цикла команд переходов и команд изменяющих значение программного счетчика PC (т.е. результат операции записывается в PCL), а также команд чтения/записи таблиц в памяти программ (запись во внутреннюю FLASH память имеет большую длительность). Коды команд приведены в Таблица 46. Неиспользуемые коды команд зарезервированы, их применение не рекомендуется. Есть некоторые особенности использования команд:

- если результат выполнения операции записывается в регистр ALUSTA, то флаги Z, C, DC и OV меняя свое значение после выполнения команды, изменят записанный результат.
- операции с регистром PCL: чтение PCL приводит к загрузке в PCLATH значения регистра PCH, запись и чтение-модификация-запись приводит к загрузке в PCH значения регистра PCLATH.
- необходимо учитывать, что команды битовых операций производят операцию «чтение-модификация-запись» целого регистра.

**Таблица 45**  
Набор команд

Мнемоника команды	Описание команды	Изменяемые флаги	Кол-во циклов
<b>ADDLW k</b>	Содержимое регистра WREG складывается с 8-ми битной константой «k» (k=0...255) и результат помещается в регистр WREG.	OV, C, DC, Z	1
<b>ADDWF f,d</b>	Сложение содержимого регистров WREG и «f» (f = 0...255). Если d=0, то результат сохраняется в регистре WREG, если d=1, то в регистре «f».	OV, C, DC, Z	1

<b>ADDWFC f,d</b>	Сложение содержимого регистров WREG, бита переноса и содержимого регистра «f» (f = 0...255). Если d=0, то результат сохраняется в регистре WREG, если d=1, то в регистре «f».	OV, C, DC, Z	1
<b>ANDLW k</b>	Логическая операция «И» 8-ми битной константы «k» (k = 0...255) и содержимого регистра WREG. Результат помещается в регистр WREG.	Z	1
<b>ANDWF f,d</b>	Логическая операция «И» содержимого регистров WREG и «f». Если d=0, то результат сохраняется в регистре WREG, если d=1, то в регистре «f» (f = 0...255).	Z	1
<b>BCF f,b</b>	Сброс бита «b» в регистре «f» (b = 0...7, f = 0...255).	-	1
<b>BSF f,b</b>	Установка в единицу бита «b» в регистре «f» (b=0...7, f = 0...255).	-	1
<b>BTFSC f,b</b>	Если бит «b» в регистре «f» равен 0, тогда следующая команда пропускается, т.е. вместо следующей команды выполняется операция NOP, в этом случае выполнение команды занимает 2 цикла (b = 0...7, f = 0...255).	-	1(2)
<b>BTFSS f,b</b>	Если бит «b» в регистре «f» равен 1, тогда следующая команда пропускается, т.е. вместо следующей команды выполняется операция NOP, в этом случае выполнение команды занимает 2 цикла (b = 0...7, f = 0...255).	-	1(2)
<b>BTG f,b</b>	Инвертирование бита «b» в регистре «f» (b = 0...7, f = 0...255).	-	1
<b>CALL k</b>	Вызов подпрограммы находящейся в пределах страницы 8Кслов. Адрес следующей после CALL команды (PC+1) помещается в стек. 13-ти битный адрес, содержащийся в коде команды, загружается в счетчик команд PC <12:0>. Затем старшие 8 бит PC копируются в PCLATH. Команда CALL выполняется за два цикла. Для вызова подпрограмм за пределами 8Кслов, смотрите команду LCALL.	-	2
<b>CLRF f,s</b>	Сбрасывает (обнуляет) регистр «f» (f = 0...255). Если s=0: сбрасываются регистры «f» и WREG, если s=1: сбрасывается регистр «f».	-	1
<b>CLRWDT</b>	Сбрасывает «Сторожевой таймер» и его предделитель. Устанавливает биты TO, PD в «1».	TO=1, PD=1	1
<b>COMF f,d</b>	Инвертирование битов регистра «f» (f = 0...255). Если d=0, то результат сохраняется в регистре WREG, если d=1, то результат сохраняется в регистре «f».	Z	1

## Спецификация 1886BE2У, К1886BE2У

<b>CPFSEQ f</b>	Сравнение содержимого регистра «f» ( $f = 0...255$ ) с содержимым регистра WREG путем беззнакового вычитания. Если $(f) = (WREG)$ , то вместо следующей команды выполняется операция NOP, в этом случае выполнение команды занимает 2 цикла.	-	1(2)
<b>CPFSGT f</b>	Сравнение содержимого регистра «f» ( $f = 0...255$ ) с содержимым регистра WREG путем беззнакового вычитания. Если $(f) > (WREG)$ , то вместо следующей команды выполняется операция NOP, в этом случае выполнение команды занимает 2 цикла.	-	1(2)
<b>CPFSLT f</b>	Сравнение содержимого регистра «f» ( $f = 0...255$ ) с содержимым регистра WREG путем беззнакового вычитания. Если $(f) < (WREG)$ , то вместо следующей команды выполняется операция NOP, в этом случае выполнение команды занимает 2 цикла.	-	1(2)
<b>DAW f,s</b>	Команда производит десятичную коррекцию результата сложения (регистр WREG) двух чисел в упакованном формате BCD. Если $s=0$ , то результат десятичной коррекции помещается в «f» и в WREG, если $s=1$ , то результат помещается в «f» ( $f = 0...255$ ). Выполнение операции: Если: $[ [WREG<7:4> > 9].OR.[C = 1] ].AND.[WREG<3:0> > 9]$ , то: $WREG<7:4> + 7 \rightarrow f<7:4>, s<7:4>$ . Если: $[WREG<7:4> > 9].OR.[C = 1]$ , то $WREG<7:4> + 6 \rightarrow f<7:4>, s<7:4>$ , иначе $WREG<7:4> \rightarrow f<7:4>, s<7:4>$ . Если: $[WREG<3:0> > 9].OR.[DC = 1]$ , то $WREG<3:0> + 6 \rightarrow f<3:0>, s<3:0>$ , иначе $WREG<3:0> \rightarrow f<3:0>, s<3:0>$ .	C	1
<b>DECf f,d</b>	Уменьшение значения регистра «f» на единицу ( $f = 0...255$ ). Если $d=0$ , то результат сохраняется в регистре WREG, если $d=1$ , то в регистре «f».	OV, C, DC, Z	1
<b>DECFSZ f,d</b>	Уменьшение значения регистра «f» на единицу и пропуск следующей команды если результат равен нулю, т.е. вместо следующей команды выполняется операция NOP, в этом случае выполнение занимает 2 цикла. Если $d=0$ , то результат операции сохраняется в регистре WREG, если $d=1$ , то в регистре «f» ( $f = 0...255$ ).	-	1(2)

## Спецификация 1886BE2У, К1886BE2У

<b>DCFSNZ f,d</b>	Уменьшение значения регистра «f» на единицу и пропуск следующей команды если результат не равен нулю, т.е. вместо следующей команды выполняется операция NOP, в этом случае выполнение занимает 2 цикла. Если d=0, то результат операции сохраняется в регистре WREG, если d=1, то в регистре «f» (f = 0...255).	-	1(2)
<b>GOTO k</b>	Безусловный переход по программе в пределах страницы 8 Кслов. 13-ти битный адрес, содержащийся в коде команды, загружается в счетчик команд PC<12:0>. Затем старшие 8 бит PC копируются в PCLATH. Команда выполняется за 2 цикла.	-	2
<b>INCF f,d</b>	Увеличение значения регистра «f» на единицу (f = 0...255). Если d=0, то результат сохраняется в регистре WREG, если d=1, то в регистре «f».	OV, C, DC, Z	1
<b>INCFSZ f,d</b>	Увеличение значения регистра «f» на единицу и пропуск следующей команды если результат равен нулю, т.е. вместо следующей команды выполняется операция NOP, в этом случае выполнение занимает 2 цикла. Если d=0, то результат операции сохраняется в регистре WREG, если d=1, то в регистре «f» (f = 0...255).	-	1(2)
<b>INFSNZ f,d</b>	Увеличение значения регистра «f» на единицу и пропуск следующей команды если результат не равен нулю, т.е. вместо следующей команды выполняется операция NOP, в этом случае выполнение занимает 2 цикла. Если d=0, то результат операции сохраняется в регистре WREG, если d=1, то в регистре «f» (f = 0...255).	-	1(2)
<b>IORLW k</b>	Логическая операция включающего «ИЛИ» 8-ми битной константы «k» (k = 0...255) и содержимого регистра WREG. Результат помещается в регистр WREG.	Z	1
<b>IORWF f,d</b>	Логическая операция включающего «ИЛИ» содержимого регистров WREG и «f». Если d=0, то результат сохраняется в регистре WREG, если d=1, то в регистре «f» (f = 0...255).	Z	1
<b>LCALL k</b>	Вызов подпрограммы находящейся в любом месте памяти в диапазоне 64Кслов. Адрес следующей после LCALL команды (PC+1) помещается в стек. 16-ти битный адрес загружается в счетчик команд PC. Младшие 8 бит загружаются из кода команды, а старшие 8 бит из регистра PCLATCH. LCALL выполняется за два цикла.	-	2

## Спецификация 1886BE2У, K1886BE2У

<b>MOVFP f,p</b>	Пересылка данных из области памяти «f» в область памяти «p». Адрес области «f» может быть от 00h до FFh, а области «p» от 00h до 1Fh. И «f» и «p» могут быть регистром WREG, а также косвенно адресованы.	-	1
<b>MOVLB k</b>	Загрузка 4х битной константы «k» в младшие 4 бита регистра выбора банка (BSR). Старшие 4 бита регистра BSR не изменяются.	-	1
<b>MOVLR k</b>	Загрузка 4-х битной константы «k» в старшие 4 бита Регистра выбора банка (BSR). Младшие 4 бита регистра BSR не изменяются.	-	1
<b>MOVLW k</b>	8-ми битная константа «k» загружается в регистр WREG.	-	1
<b>MOVFP p,f</b>	Пересылка данных из области памяти «p» в область памяти «f». Адрес области «f» может быть от 00h до FFh, а области «p» от 00h до 1Fh. И «f» и «p» могут быть регистром WREG, а также косвенно адресованы.	Z	1
<b>MOVWF f</b>	Пересылка содержимого регистра WREG в регистр «f» (f = 0...255).	-	1
<b>MULLW k</b>	Беззнаковое перемножение 8-ми битной константы «k» и содержимого регистра WREG. 16-ти битный результат записывается в паре регистров PRODH:PRODL. Регистр WREG не изменяется. Операция не изменяет флаги.	-	1
<b>MULWF f</b>	Беззнаковое перемножение содержимого регистров «f» (f = 0...255) и WREG. 16-ти битный результат записывается в паре регистров PRODH:PRODL. Регистры WREG и «f» не изменяются. Операция не изменяет флаги.	-	1
<b>NEGW f,s</b>	Изменение знака содержимого регистра WREG путем двоичного дополнения. Если s=0, то результат помещается в «f» и в WREG, если s=1, то результат помещается в «f» (f = 0...255).	OV, C, DC, Z	1
<b>NOP</b>	Нет операции.	-	1
<b>RETFIE</b>	Возврат из прерывания. Содержимое счетчика команд восстанавливается из стека. Разрешаются глобальные прерывания сбросом бита GLINTD(CPUSTA<4>). PCLATCH не изменяется. Команда выполняется за 2 цикла.	GLINTD=0	2
<b>RETLW k</b>	Возврат из подпрограммы. В регистр WREG загружается значение 8-ми битной константы «k». В счетчик команд загружается из стека адрес возврата. PCLATCH не изменяется. Команда выполняется за два цикла.	-	2

<b>RETURN</b>	Возврат из подпрограммы. В счетчик команд загружается из стека адрес возврата. PCLATCH не изменяется. Команда выполняется за два цикла.	-	2
<b>RLCF f,d</b>	Операция циклического сдвига содержимого регистра «f» влево через флаг переноса «C». Если d=0, то результат операции сохраняется в регистре WREG, если d=1, то в регистре «f» (f = 0...255).	C	1
<b>RLNCF f,d</b>	Операция циклического сдвига содержимого регистра «f» влево. Если d=0, то результат операции сохраняется в регистре WREG, если d=1, то в регистре «f» (f = 0...255).	-	1
<b>RRCF f,d</b>	Операция циклического сдвига содержимого регистра «f» вправо через флаг переноса «C». Если d=0, то результат операции сохраняется в регистре WREG, если d=1, то в регистре «f» (f = 0...255).	C	1
<b>RRNCF f,d</b>	Операция циклического сдвига содержимого регистра «f» вправо. Если d=0, то результат операции сохраняется в регистре WREG, если d=1, то в регистре «f» (f = 0...255).	-	1
<b>SETF f,s</b>	Установка всех битов регистра «f» в единицы. Если s=0, то значение 0FFh помещается в «f» и в WREG, если s=1, то результат помещается только в «f» (f = 0...255).	-	1
<b>SLEEP</b>	Сбрасывает «сторожевой таймер» и его предделитель. Бит «ТО» устанавливается, а «РО» сбрасывается. Процессор переходит в режим «сна» (SLEEP) с остановкой тактового генератора.	TO=1, PD=0	1
<b>SUBLW k</b>	Содержимое регистра WREG вычитается из 8-ми битной константы «k». Результат помещается в регистр WREG.	OV, C, DC, Z	1
<b>SUBWF f,d</b>	Вычитание содержимого регистра WREG из содержимого регистра «f». Если d=0, то результат операции сохраняется в регистре WREG, если d=1, то в регистре «f» (f = 0...255).	OV, C, DC, Z	1
<b>SUBWFB f,d</b>	Вычитание содержимого регистра WREG и флага переноса (заема) из содержимого регистра «f» (метод двоичного дополнения). Если d=0, то результат операции сохраняется в регистре WREG, если d=1, то в регистре «f» (f = 0...255).	OV, C, DC, Z	1

<b>SWAPF f,d</b>	Обмен местами полубайтов регистра «f». Верхняя половина регистра и нижняя меняются местами. Если d=0, то результат операции сохраняется в регистре WREG, если d=1, то в регистре «f» (f = 0...255).	-	1
<b>TABLRD t,i,f</b>	1. Содержимое байта «табличной защелки» (TBLAT) пересылается в регистр «f» (f = 0...255). Если t=1, то пересылается старший байт, если t=0, то младший байт. 2. Содержимое области памяти программ, указываемой 16-ти битным «табличным указателем» (TBLPTR) загружается в 16-ти битную «табличную защелку» (TBLAT). 3. Если i=1, то значение TBLPTR увеличивается на единицу, если i=0, то значение TBLPTR не изменяется. Команда выполняется 2 цикла, а если регистр «f» является регистром PCL, то 3 цикла.	-	2(3)
<b>TABLWT t,i,f</b>	1. Загрузка содержимого регистра «f» (f = 0...255) в 16-ти битную «табличную защелку» (TBLAT). Если t=1, то загружается в старший байт, если t=0, то в младший байт. 2. Содержимое «табличной защелки» (TBLAT) записывается в область памяти программ, указываемой «табличным указателем» (TBLPTR). Если TBLPTR указывает на область внешней памяти программ, то команда выполнится за 2 цикла. Если TBLPTR указывает на внутреннюю область FLASH памяти, то выполнение команды происходит до прерывания (т.е. >2 циклов). 3. Если i=1, то значение TBLPTR увеличивается на единицу, если i=0, то значение TBLPTR не изменяется. Примечание: Для записи во внутреннюю память программ должно быть подано напряжение программирования. Если это условие не выполнено, то команда выполнится за 2 цикла и значение памяти не изменится.	-	2(>2)
<b>TLRD t,f</b>	Считывание данных из 16-ти битной «табличной защелки» в регистр «f» (f = 0...255). «Табличная защелка» при этом не изменяется. Команда используется совместно с TABLRD для пересылки данных из памяти программ в память данных. Если t=1, то считывается старший байт, если t=0, то младший байт.	-	1

<b>TLWT t,f</b>	Содержимое регистра «f» (f = 0...255) записывается в 16-ти битную «табличную защелку» (TBLAT). Команда используется совместно с командой TABLWT для пересылки данных из памяти данных в память программ. Если t=1, то записывается старший байт, если t=0, то младший байт.	-	1
<b>TSTFSZ f</b>	Сравнение содержимого регистра «f» (f = 0...255) с нулем. Если (f) = 0, то вместо следующей команды выполняется операция NOP, в этом случае выполнение команды занимает 2 цикла.	-	1(2)
<b>XORLW k</b>	Логическая операция исключающего «ИЛИ» 8-ми битной константы «k» (k = 0...255) и содержимого регистра WREG. Результат помещается в регистр WREG.	Z	1
<b>XORWF f,d</b>	Логическая операция исключающего «ИЛИ» содержимого регистров WREG и «f». Если d=0, то результат сохраняется в регистре WREG, если d=1, то в регистре «f» (f = 0...255).	Z	1

**Таблица 46**  
Коды команд

Мнемоника команды	Код команды	Мнемоника команды	Код команды	Мнемоника команды	Код команды
ADDLW k	1011 0001 kkkk kkkk	DCFSNZ f,d	0010 011d ffff ffff	RETURN	0000 0000 0000 0010
ADDWF f,d	0000 111d ffff ffff	GOTO k	110k kkkk kkkk kkkk	RLCF f,d	0001 101d ffff ffff
ADDWFC f,d	0001 000d ffff ffff	INCF f,d	0001 010d ffff ffff	RLNCF f,d	0010 001d ffff ffff
ANDLW k	1011 0101 kkkk kkkk	INCFSZ f,d	0001 111d ffff ffff	RRCF f,d	0001 100d ffff ffff
ANDWF f,d	0000 101d ffff ffff	INFSNZ f,d	0010 010d ffff ffff	RRNCF f,d	0010 000d ffff ffff
BCF f,b	1000 1bbb ffff ffff	IORLW k	1011 0011 kkkk kkkk	SETF f,s	0010 101s ffff ffff
BSF f,b	1000 0bbb ffff ffff	IORWF f,d	0000 100d ffff ffff	SLEEP	0000 0000 0000 0011
BTFSC f,b	1001 1bbb ffff ffff	LCALL k	1011 0111 kkkk kkkk	SUBLW k	1011 0010 kkkk kkkk
BTFSS f,b	1001 0bbb ffff ffff	MOVFP f,p	011p pppp ffff ffff	SUBWF f,d	0000 010d ffff ffff
BTG f,b	0011 1bbb ffff ffff	MOVLB k	1011 1000 uuuu kkkk	SUBWFB f,d	0000 001d ffff ffff
CALL k	111k kkkk kkkk kkkk	MOVLR k	1011 101x kkkk uuuu	SWAPF f,d	0001 110d ffff ffff
CLRF f,s	0010 100s	MOVLW k	1011 0000	TABLWD t,i,f	1010 10ti



## Спецификация 1886BE2У, К1886BE2У

	ffff ffff		kkkk kkkk		ffff ffff
CLRWDТ	0000 0000 0000 0100	MOVPF p,f	010p pppp ffff ffff	TABLWT t,i,f	1010 11ti ffff ffff
COMF f,d	0001 001d ffff ffff	MOVWF f	0000 0001 ffff ffff	TLRD t,f	1010 00tx ffff ffff
CPFSEQ f	0011 0001 ffff ffff	MULLW k	1011 1100 kkkk kkkk	TLWT t,f	1010 01tx ffff ffff
CPFSGT f	0011 0010 ffff ffff	MULWF f	0011 0100 ffff ffff	TSTFSZ f	0011 0011 ffff ffff
CPFSLT f	0011 0000 ffff ffff	NEGW f,s	0010 110s ffff ffff	XORLW k	1011 0100 kkkk kkkk
DAW f,s	0010 111s ffff ffff	NOP	0000 0000 0000 0000	XORWF f,d	0000 110d ffff ffff
DECF f,d	0000 011d ffff ffff	RETFIE	0000 0000 0000 0101		
DECFSZ f,d	0001 011d ffff ffff	RETLW k	1011 0110 kkkk kkkk		

### Обозначения:

f - адрес регистра от 00h до FFh.

p - адрес периферийного регистра от 00h до 1Fh.

k - поле константы (данные или адрес).

b - адрес бита в 8-ми разрядном регистре.

d - выбор места назначения для размещения результата: если =0 - результат помещается в регистр WREG, если =1 - в указанный регистр.

s - выбор места назначения для размещения результата: если =0 - результат помещается в указанный регистр и регистр WREG, если =1 - только в указанный регистр.

i - управление «табличным указателем»: если =1 - значение указателя инкрементируется после выполнения операции, если =0 - не изменяется.

t - выбор байта в 16-ти разрядной «табличной защелке»: если =1 - старший байт, если =0 - младший байт.

x,u - не используются, имеют значение 0.

**Предельные и предельно-допустимые режимы работы**

**Таблица 47**

Наименование параметра, единица измерения	Буквенное обозначение	Норма параметра			
		Предельно-допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Напряжение программирования и стирания, В на входах: 47, 48	$U_{PR}$	11,75	12,25	-	-
Частота следования импульсов тактовых сигналов, МГц,	$f_C$	-	33	-	-
Длительность программирования одного слова, мкс	$t_{CYW}$	500	-	-	-
Напряжение питания АЦП, В	$U_{CCA}$	4,5	5,5	-	7,0
Длительность периода преобразования одного разряда АЦП, мкс при: $U_{CC}=5,12$ В	$T_{AD}$	1,6	-	-	-
Длительность сигнала высокого уровня синхронизации на входах PA1(9) и OSC1 (15), нс при: $U_{CC}=4,5$ В	$t_{WH(PA\_OSC)}$	10	-	-	-
Время установления данных относительно OSC1 (15), нс при: $U_{CC}=4,5$ В	$t_{SU(OSC-D)}$	25	-	-	-
Время удержания данных относительно OSC1 (15), нс при: $U_{CC}=5,5$ В	$t_H(OSC-D)$	15	-	-	-
Длительность сигнала высокого уровня прерывания INT/PA0 (24), нс, при: $U_{CC}=4,5$ В	$t_{WH(INT)}$	25	-	-	-
Длительность сигнала высокого уровня прерывания PB7 (13):PB0 (23), нс при: $U_{CC}=4,5$ В	$t_{WH(PB)}$	25	-	-	-
Длительность сигнала низкого уровня системного сброса, MCLR/ $U_{PP}$ (47), нс при: $U_{CC}=4,5$ В	$t_{WL(MCLR)}$	100	-	-	-
<b>Параметры синхронного последовательного приемопередатчика в режиме I2C</b>					
Время установления данных относительно SCL (10) на линии SDA (11), нс, при: $U_{CC}=4,5$ В	$t_{SU(SDA-SCL)}$	$2 \cdot TC \cdot (BRG+1)$		-	-
Время удержания данных относительно SCL (10) на линии SDA (11), нс, при: $U_{CC}=4,5$ В	$t_H(SDA-SCL)$	0	-	-	-
<b>Параметры синхронного последовательного приемопередатчика в режиме SPI</b>					

## Спецификация 1886BE2У, К1886BE2У

Наименование параметра, единица измерения	Буквенное обозначение	Норма параметра			
		Предельно-допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Длительность сигнала высокого уровня на линии SCK (12), нс при: $U_{CC}=4,5\text{ В}$	$t_{WH(SCK)}$	40	-	-	-
Длительность сигнала низкого уровня на линии SCK (12), нс при: $U_{CC}=4,5\text{ В}$	$t_{WL(SCK)}$	40	-	-	-
Время установления данных SDI (11) относительно SCK (12), нс при: $U_{CC}=4,5\text{ В}$	$t_{SU(SCK-SDI)}$	100	-	-	-
Время удержания данных SDI (11) относительно SCK (12), нс при: $U_{CC}=4,5\text{ В}$	$t_{H(SCK-SDI)}$	100	-	-	-
<b>Параметры универсального последовательного синхронно-асинхронного приемопередатчика</b>					
Время установления данных RX/DT (5, 8) относительно синхронизации TX/CK (6, 7), нс при: $U_{CC}=4,5\text{ В}$	$t_{SU(TX-RX)}$	17	-	-	-
Время удержания данных RX/DT (5, 8) относительно синхронизации TX/CK (6, 7), нс при: $U_{CC}=4,5\text{ В}$	$t_{H(TX-RX)}$	17	-	-	-
<b>Параметры Таймера 0</b>					
Длительность сигнала высокого уровня на линии PA1/T0CLK (22), нс, прескалер включен при: $U_{CC}=4,5\text{ В}$	$t_{WH(PA1)}$	10	-	-	-
Длительность сигнала низкого уровня на линии PA1/T0CLK (22), нс прескалер включен при: $U_{CC}=4,5\text{ В}$	$t_{WL(PA1)}$	10	-	-	-
Период сигнала на линии PA1/T0CLK (22), нс при: $U_{CC}=4,5\text{ В}$	$T_C(PA1)$	20 или $(4 \cdot T_C + 25)/N$	-	-	-
<b>Параметры Таймера 1, 2 и 3</b>					
Период сигнала на линии PB4/TCLK12 (20), нс при: $U_{CC}=4,5\text{ В}$	$T_C(PB4)$	$4 \cdot T_C + 25$	-	-	-
Период сигнала на линии PB5/TCLK3 (19), нс при: $U_{CC}=4,5\text{ В}$	$T_C(PB5)$	$4 \cdot T_C + 25$	-	-	-
<b>Параметры схем регистрации событий</b>					
Длительность сигнала высокого уровня на линиях PB0 (23), PB1 (22), PG4 (3), PE3 (46), нс при: $U_{CC}=4,5\text{ В}$	$t_{WH(PB,PG,PE)}$	10	-	-	-

## Спецификация 1886BE2У, К1886BE2У

Наименование параметра, единица измерения	Буквенное обозначение	Норма параметра			
		Предельно-допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Длительность сигнала низкого уровня на линиях PB0 (23), PB1 (22), PG4 (3), PE3 (46), нс при: $U_{CC}=4,5$ В	$t_{WL}(PB,PG,PE)$	10	-	-	-
Период сигнала на линиях PB0 (23), PB1 (22), PG4 (3), PE3 (46), нс при: $U_{CC}=4,5$ В	$T_C(PB,PG,PE)$	$2 \cdot (4 \cdot T_C) / N1$	-	-	-
Длительность фронта нарастания и спада входного сигнала на выводе OSC1 (15), нс при: $f_c=33$ МГц	$t_r$ $t_f$	-	5	-	7
Емкость нагрузки, пФ	$C_L$	-	40	-	60
Напряжение источника питания, В	$U_{CC}$	4,5	5,5	-	7,0
Входное напряжение высокого уровня, В - входы с КМОП уровнями	$U_{IH}$	$0,8 \cdot U_{CC}$	$U_{CC}$		$U_{CC}+0,3$
- входы с ТТЛ уровнями		2,0	$U_{CC}$	-	$U_{CC}+0,3$
Входное напряжение низкого уровня, В - входы с КМОП уровнями	$U_{IL}$	0	$0,2 \cdot U_{CC}$	минус 0,3	-
- входы с ТТЛ уровнями		0	0,8	минус 0,3	-
Длительность фронта сигнала синхронизации на OSC1, PA1, нс	$t_R$	-	5	-	-
Длительность спада сигнала синхронизации на OSC1, PA1, нс	$t_F$	-	5	-	-
Максимальная рассеиваемая мощность, Вт	$P_C$	-	0,3	-	1,0

Примечание:

1. Не допускается одновременное воздействие двух и более предельных режимов.
2. п - в названии вывода - обозначает инверсию.

Буквенные обозначения, приведенные в формулах:

$T_C$  - период сигнала внешнего синхронизатора на входе OSC1 (15);

BRG - значение регистра SPBRGX блока универсального последовательного синхронно-асинхронного приемопередатчика (см. ТСКЯ.431295.001РЭ);

N - значение регистра TOSTA блока таймера 0 (см. ТСКЯ.431295.001РЭ);

N1 - значение прескалера блоков захвата и регистрации событий

Стойкость к воздействию статического электричества – не менее 1000 В.

**Электрические параметры микросхемы**

**Таблица 48**

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра	
		не менее	не более
Выходное напряжение низкого уровня, В: на выводах I/O (3...8, 12, 13, 18...23, 51...58, 61...64) при: $U_{CC}=4,5\text{ В}$ , $I_{OL}=4\text{ мА}$	$U_{OL}$	-	0,45
на выводах буферов ТТЛ (25...31, 34...46) при: $U_{CC}=4,5\text{ В}$ , $I_{OL}=6,0\text{ мА}$		-	0,40
на выводах PA2 (10) и PA3 (11) при $U_{CC}=4,5\text{ В}$ , $I_{OL}=60,0\text{ мА}$		-	1,0
на выводах OSC2 (16) при: $U_{CC}=4,5\text{ В}$ , $I_{OL}=1,0\text{ мА}$		-	0,45
Выходное напряжение высокого уровня, В: на выводах I/O (3...8, 12, 13, 18...23, 51...58, 61...64) при: $U_{CC}=4,5\text{ В}$ , $I_{OH}=\text{минус }4\text{ мА}$	$U_{OH}$	4,05	-
на выводах буферов ТТЛ (25...31, 34...46) при: $U_{CC}=4,5\text{ В}$ , $I_{OH}=\text{минус }6,0\text{ мА}$		2,40	-
на выводах OSC2 (16) при: $U_{CC}=4,5\text{ В}$ , $I_{OH}=\text{минус }1,0\text{ мА}$		4,05	-
Уровень напряжения срабатывания схемы генерации сброса, В	$U_{BOR}$	3,6	4,3
Входной ток утечки низкого уровня, мкА при: $U_I=0\text{ В}$	$I_{ILL}$	-	$ \pm 1 $
на выводах (3...8, 12, 13, 18...23, 51...58, 61...64)		-	$ \pm 2 $
- PA2 (10), PA3(11) - MCLR/ $U_{PP}$ (47), Test (48)		-	$ \pm 7 $
Входной ток утечки высокого уровня, мкА при: $U_I=U_{CC}=5,5\text{ В}$	$I_{ILH}$	-	$ \pm 1 $
на выводах (3...8, 12, 13, 18...23, 51...58, 61...64))		-	$ \pm 2 $
- PA2 (10), RA3 (11) - MCLR/ $U_{PP}$ (47), Test (48)		-	$ \pm 7 $
Входной ток утечки высокого уровня, мкА на выводах nMCLR/ $U_{PP}$ (47), Test (48) при: $U_I=12\text{ В}$	$I_{ILH1}$	-	$ \pm 90 $
Статический ток потребления в режиме покоя, мкА при: $U_{CC}=5,5\text{ В}$	$I_{CCS}$	-	50

## Спецификация 1886BE2У, К1886BE2У

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра	
		не менее	не более
Динамический ток потребления, мА, на выводах (2, 14, 33, 50, 59) при: $U_{CC}=5,5\text{ В}$ , $f_C=33\text{ МГц}$	$I_{OCC}$	-	50
Ток низкого уровня выводов порта В, подключенного к $U_{CC}$ , мкА на выводах (12, 13, 18...23) при $U_{CC}=5,5\text{ В}$ , $U_I=0\text{ В}$	$I_{ILB}$	$ \pm 80 $	$ \pm 270 $
Время задержки данных относительно OSC1(15), нс, на выводах (3...8, 12, 13, 18...23, 25...31, 34...46, 51...58, 61...64) при: $U_{CC}=4,5\text{ В}$	$t_{d(Osc-D)}$	-	90
Период работы внутреннего RC - генератора сторожевого таймера, мкс при: $U_{CC}=5,5\text{ В}$	$T_{RC}$	60	140
Время задержки данных на линии SDA (11) относительно SCL (10), нс при: $U_{CC}=4,5\text{ В}$ Скорость передачи данных      100 КГц 400 КГц 1 МГц	$t_{d(SCL-SDA)}$	- - -	3500 1000 400
Время задержки данных SDO (13) относительно SCK (12), нс при: $U_{CC}=4,5\text{ В}$	$t_{d(SCK-SDO)}$	-	50
Время задержки данных RX/DT (6, 8) от фронта TX/SK (5, 7), нс при: $U_{CC}=4,5\text{ В}$	$t_{d(TX-RX)}$	-	50
Дифференциальная нелинейность, единица младшего разряда на выводах: (51...58, 61...64) при: $U_{CCA}=5,12\text{ В}$ , $0 \leq U_{IN} \leq U_{CCA}$	$E_{DL}$	-	$ \pm 1 $
Разрешающая способность, разрядов на выводах: (51...58, 61...64) при: $U_{CCA}=5,12\text{ В}$ , $0 \leq U_{IN} \leq U_{CCA}$	$E_N$	-	10
Интегральная нелинейность, единица младшего разряда на выводах: (51...58, 61...64) при: $U_{CCA}=5,12\text{ В}$ , $0 \leq U_{IN} \leq U_{CCA}$	$E_{IL}$	-	$ \pm 2 $
Ошибка смещения, единица младшего разряда на выводах: (51...58, 61...64) при: $U_{CCA}=5,12\text{ В}$ , $0 \leq U_{IN} \leq U_{CCA}$	$E_{OFF}$	-	$ \pm 1 $

### Примечание

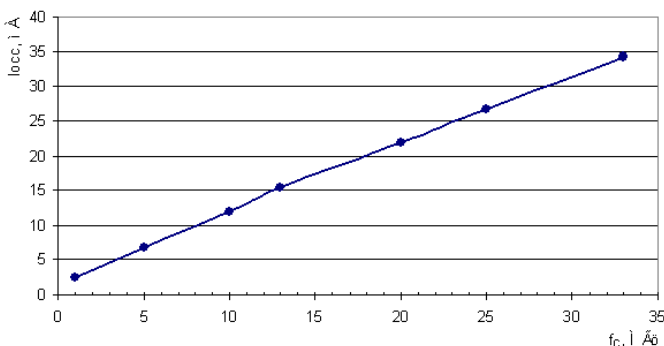
n - в названии вывода - обозначает инверсию.

## Типовые зависимости

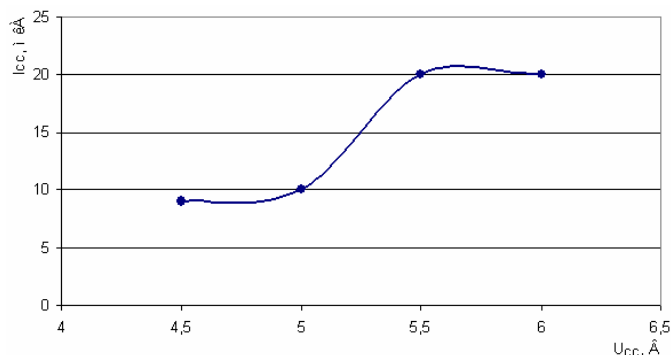
Зависимость частоты RC генератора от внешних времязадающих элементов (R, C) на входе OSC1 (см. ТСКЯ.431295.001ТО).

Условия измерения:  $U_{CC}=5\text{ В}$ ,  $T=25^\circ\text{C}$

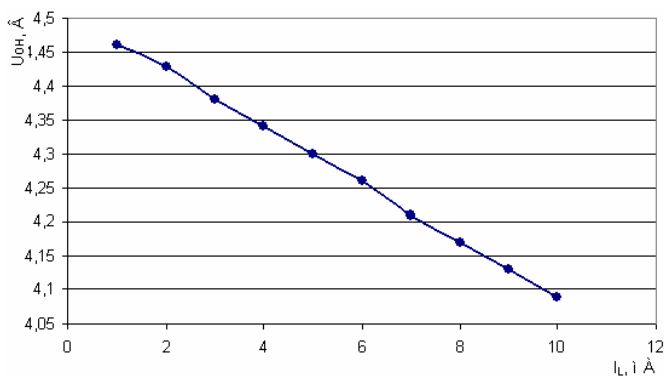
СЕХТ	РЕХТ	fC
22 пФ	10 кОм	2676 кГц
	100 кОм	278 кГц
100 пФ	3.3 кОм	2856 кГц
	5.1 кОм	2000 кГц
	10 кОм	1080 кГц
	100 кОм	116 кГц
300 пФ	3.3 кОм	1120 кГц
	5.1 кОм	756 кГц
	10 кОм	404 кГц
	100 кОм	27,6 кГц



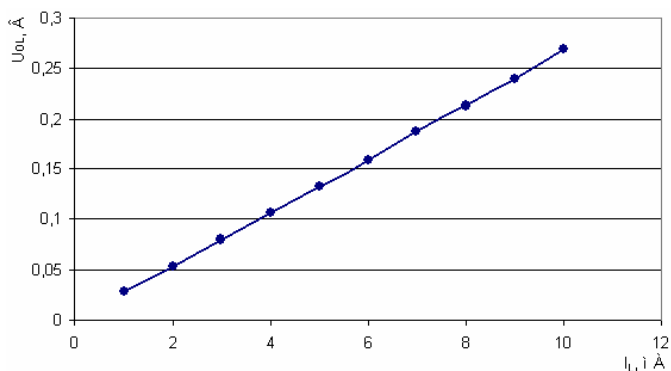
**Рис. 85** Зависимость динамического тока потребления от частоты следования синхросигнала ( $f_c$ ). Нормальные условия,  $U_{CC}=5,5\text{ В}$



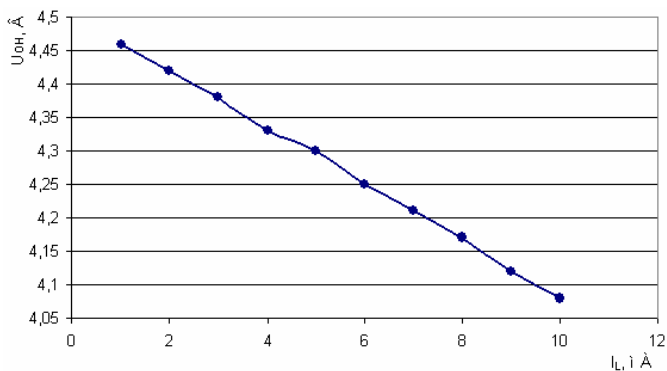
**Рис. 86** Зависимость статического тока потребления от напряжения питания. Нормальные условия.



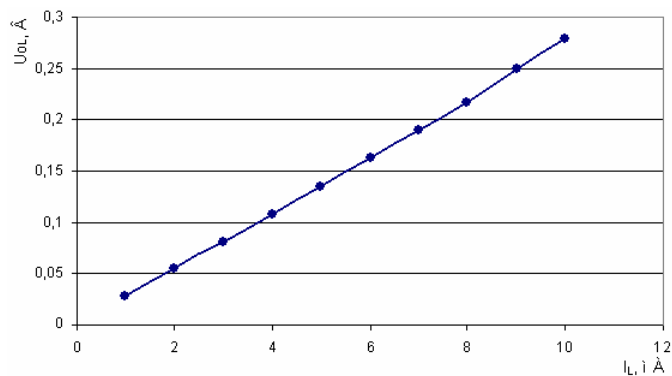
**Рис. 87** Зависимость выходного напряжения логической «1» ТТЛ выходов от тока нагрузки. Нормальные условия,  $U_{CC}=4,5\text{ В}$



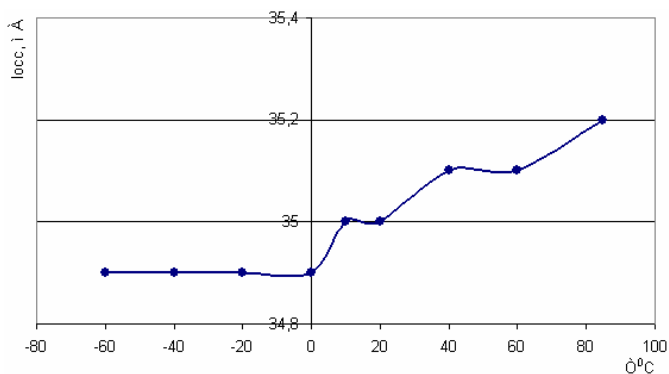
**Рис. 85** Зависимость выходного напряжения логического «0» ТТЛ выходов от тока нагрузки. Нормальные условия,  $U_{CC}=4,5\text{ В}$



**Рис. 86** Зависимость выходного напряжения логической «1» выходов с триггером Шмидта от тока нагрузки. Нормальные условия,  $U_{CC} = 4,5$  В



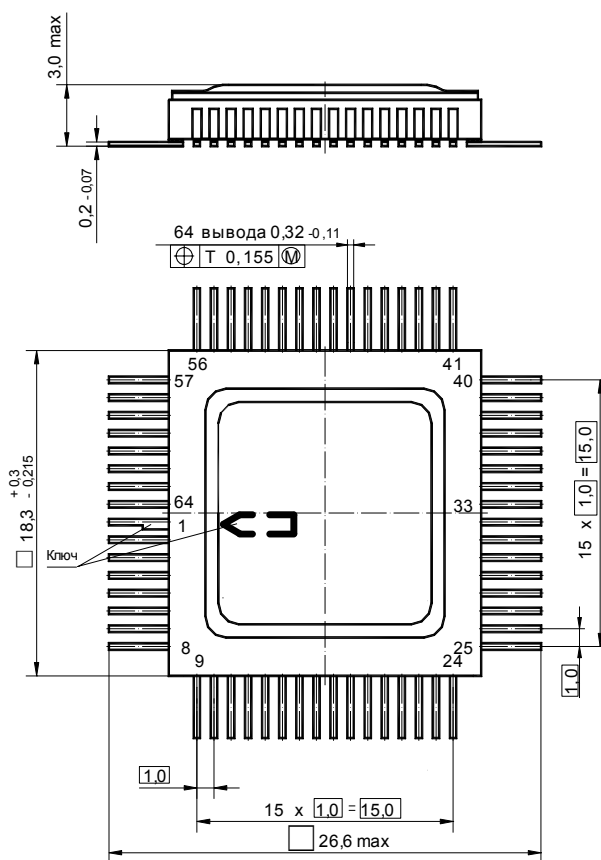
**Рис. 870** Зависимость выходного напряжения логической «1» выходов с триггером Шмидта от тока нагрузки. Нормальные условия,  $U_{CC} = 4,5$  В



**Рис. 88** Зависимость динамического тока потребления от температуры.  $U_{CC} = 5,5$  В,  $f_c = 34$  МГц



**Габаритный чертеж микросхемы**



**Корпус Н18.64-1В**

**Информация для заказа**

Обозначение	Маркировка	Тип корпуса	Температурный диапазон
1886BE2У	1886BE2У	Н18.64-1В	минус 60 – 85 °С
К1886BE2У	К1886BE2У	Н18.64-1В	минус 60 – 85 °С
К1886BE2АУ	К1886BE2У*	Н18.64-1В	0 – 70 °С

Микросхемы с приемкой «ВП» маркируются ромбом.  
Микросхемы с приемкой «ОТК» маркируются буквой «К».

## **Спецификация 1886BE2У, К1886BE2У**

---

### Лист регистрации изменений

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов	№№ новых листов
1	27.02.2010	2.2	Добавлен лист регистрации	-	163
2	24.03.2010	2.3	Корректировка на основании планового пересмотра документации	1, 161	-
3	27.04.2010	2.4	1. Замена логотипа 2. Табл.37 изменены 1, 2 строки	1 98	-