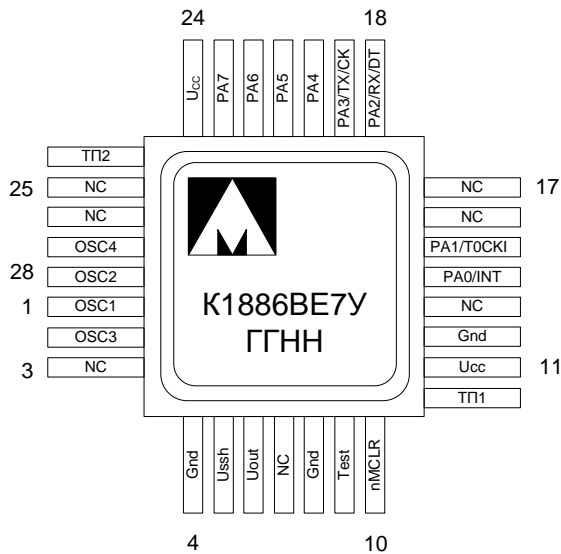




**Микропотребляющая микросхема  
однокристалльной микро-ЭВМ с ЭСППЗУ EEPROM-типа  
(2К x 16) и встроенным регулятором напряжения  
1886BE7(71)У, К1886BE7(71)У, К1886BE71Н4**



ТП – технологическая перемычка

ГГ – неделя выпуска

НН – год выпуска

**Тип корпуса:**

- 28-выводной металлокерамический корпус Н09.28-1В;
- микросхемы К1886BE71Н4 поставляются в бескорпусном исполнении.

**Основные параметры  
микросхемы:**

- Тактовая частота до 10 МГц;
- 58 однословных инструкций (коды инструкций 16-ти разрядные);
- 8-ми разрядное АЛУ;
- Аппаратно реализованного инструкция 8x8 битного умножения;
- Поддержка прямого, косвенного и относительного режимов адресации;
- Наличие инструкций одновременно работающих с двумя регистрами;
- EEPROM память данных емкостью 256 байт;
- Аппаратная поддержка интерфейса LIN;
- Функция самостоятельной модификации кода программ;
- Встроенный линейный регулятор с диапазоном входных напряжений до 16 В и током нагрузки до 50 мА;
- Уменьшенное до 14 мс время старта микроконтроллера 1886BE71У (типичное значение);
- Диапазон напряжения питания 4,5 – 5,5 В;
- Температурный диапазон:

Обозначение	Диапазон
1886BE7(71)У	минус 60 – 125 °С
К1886BE7(71)У	минус 60 – 125 °С
К1886BE7(71)УК	0 – 70 °С

**Примечание** – Микросхема К1886BE7(71)АУ является полным аналогом К1886BE7(71)У с другим температурным диапазоном.

**Особенности ядра микроконтроллера:**

- 58 однословных инструкций.
- Все инструкции выполняются за один цикл, за исключением инструкций переходов и инструкций чтения/записи таблиц выполняемых за два цикла.
- Скорость работы: тактовая частота до 10 МГц, минимальная длительность цикла 400 нс.
- 8 x 8 битный аппаратный умножитель, за цикл.
- Поддержка прерываний.
- 16 словный аппаратный стек.
- Прямая, косвенная и относительная модель адресации.

ОЗУ данных емкостью 902 байта.

Внутренняя память программ типа EEPROM размером 2К x 16.

Встроенный умножитель напряжения для программирования/стирания EEPROM-памяти.

**Краткое описание периферии:**

- 8 пользовательских выводов.
- 16-ти битный таймер/счетчик с 8-ми битным предварительным делителем.
- Один Универсальный Синхронный Асинхронный Приемник Передатчик (USART) с программируемой скоростью передачи и поддержкой режима LIN.
- Универсальный контроллер внутренней памяти данных типа CMOS EEPROM размером 256 x 8.
- Встроенный регулятор напряжения.

**Специализированные особенности:**

- Сброс по снижению питания.
- Отложенный запуск по подаче питания и тактовой частоты.
- Сторожевой таймер.
- Защищенный режим.
- Режим энергосбережения (SLEEP).
- Уровень напряжения питания микроконтроллера от 4,5 В до 5,5 В.

**Общее описание и области применения микросхемы**

Микросхема предназначена для широкого применения в аппаратуре общего назначения, автомобильной технике, железнодорожном, водном и воздушном транспорте в качестве малопотребляющего контроллера электронных блоков.

**Основные области применения:**

- интеллектуальные датчики;
- автомобильная техника;
- промышленные системы управления;
- телекоммуникационное оборудование;
- системы безопасности;
- горнорудная электронная аппаратура.

## **Содержание**

Описание выводов.....	6
Структурная блок-схема микросхемы .....	7
Указания по применению и эксплуатации .....	7
Описание функционирования микросхемы.....	8
Встроенный тактовый генератор .....	8
Использование кварцевого или керамического резонатора .....	8
Внешний тактовый генератор .....	9
Режим RC генератора.....	9
Синхронизация выполнения команды .....	10
Схема сброса микроконтроллера .....	11
Сброс по включению питания.....	12
Таймер включения питания PWRT .....	13
Таймер запуска генератора.....	13
Сброс по снижению напряжения питания .....	18
Прерывания.....	19
Регистр состояния прерываний (INTSTA) .....	19
Регистр состояния прерываний .....	20
Регистр разрешения периферийных прерываний PIE1 .....	22
Регистр запроса периферийных прерываний PIR1 .....	22
Регистры адреса точки останова DBL и DBH .....	23
Обработка прерываний .....	24
Прерывание от вывода PA0/INT .....	25
Прерывание от вывода PA1/T0CLK .....	25
Периферийные прерывания .....	26
Сохранение регистров при прерывании .....	26
Организация памяти .....	29
Память программ .....	29
Память данных .....	30
Регистры общего назначения (GPR).....	30
Регистры специального назначения (SFR) .....	30
Функционирование стека .....	38
Косвенная адресация .....	39
Регистры для чтения/записи таблиц.....	40
Модуль счетчика команд.....	41
Регистр выбора банка (BSR).....	43
Считывание и запись таблиц данных .....	43
Запись таблиц во внутреннюю память.....	44
Чтение таблиц .....	46
Аппаратный умножитель.....	46
Порты ввода-вывода.....	50
Регистр порта А и регистр направления данных DDRA .....	50
Блок «таймер 0» .....	52
Модуль универсального синхронно-асинхронных приемопередатчика с поддержкой LIN интерфейса .....	56
Регистр режима и статуса работы приемника.....	57

Регистр режима и статуса работы передатчика .....	58
Регистр режима и статуса работы приемника LIN заголовка.....	59
Регистр данных приемника .....	59
Регистр данных передатчика .....	60
Регистр задания скорости приема и передачи .....	61
Регистр скорости поля SYNCH в LIN фрейме .....	61
Генератор скорости передачи данных .....	63
Асинхронный режим.....	64
Асинхронный передатчик.....	66
Асинхронный приемник .....	68
Режим автоматического приема LIN заголовка .....	70
Передача LIN фрейма .....	71
Синхронный ведущий режим .....	72
Передача данных в синхронном ведущем режиме.....	73
Прием данных в синхронном ведущем режиме.....	75
Синхронный ведомый режим.....	77
Передача данных в синхронном ведомом режиме .....	77
Прием данных в синхронном ведомом режиме .....	77
Блок внутренней памяти данных EEPROM.....	79
Основные выполняемые функции и возможности .....	80
Регистр режима работы контроллера .....	81
Регистр контроля и тестирования .....	82
Регистр режима работы .....	84
Регистр данных.....	85
Регистр адреса обращения .....	86
Работа блока по стиранию, записи и чтению данных .....	87
Включение EEPROM .....	87
Очистка всей EEPROM.....	87
Запись всей EEPROM одним значением .....	87
Очистка строки EEPROM .....	88
Запись слова в EEPROM .....	89
Чтение слова из EEPROM .....	90
Описание блока управления EEPROM – памятью программ.....	91
Описание регистров .....	92
Регистр коэффициента деления частоты генератора .....	92
Регистр управления EEPROM памяти.....	93
Регистр режима работы EEPROM памяти .....	93
Регистр старшего адреса EEPROM памяти.....	95
Регистр младшего адреса EEPROM памяти .....	95
Регистр конфигурационных бит .....	95
Регистр младшего байта данных EEPROM памяти.....	96
Регистр старшего байта данных EEPROM памяти .....	96
Выполнение операций с блоком EEPROM – памятью программ.....	97
Выполнение операций записи/стирания памяти .....	97
Режимы адресации памяти .....	97
Выполнение операции чтения памяти .....	98
Чтение/запись конфигурационных бит .....	98
Специальные модули микроконтроллера .....	99

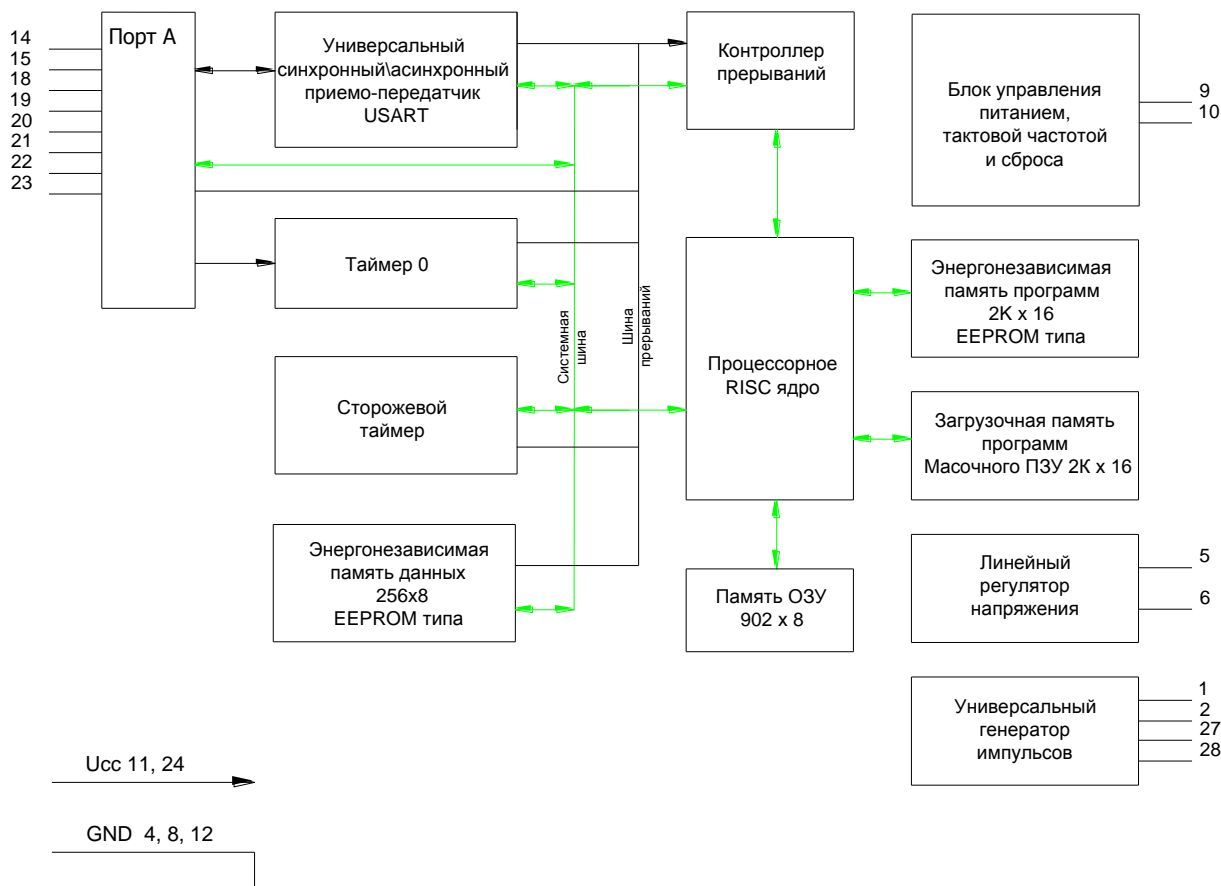
Регистры конфигурации микроконтроллера .....	99
Внутрисхемное программирование микроконтроллера .....	100
Сторожевой таймер.....	101
Режим энергосбережения (SLEEP).....	101
Схема подключения напряжения питания .....	103
Система команд .....	104
Предельные и предельно-допустимые режимы работы .....	114
Электрические параметры микросхемы .....	117
Электрические параметры микросхемы К1886BE71Н4 (бескорпусное исполнение) .....	120
Типовые зависимости .....	121
Габаритный чертеж микросхемы .....	124
Информация для заказа.....	126
Лист регистрации изменений.....	127

**Описание выводов**

**Таблица 1**

<b>№ вывода корпуса</b>	<b>№ контактной площадки кристалла</b>	<b>Условное обозначение</b>	<b>Тип вывода</b>	<b>Назначение вывода</b>
<b>Генератор</b>				
27	24	OSC4	выход	Выход генератора для подключения RC цепи (в режиме RC генератора)
28	28	OSC2	выход	Выход обратной связи для внешнего кварцевого резонатора
1	1	OSC1	вход	Вход для сигналов тактовой синхронизации, от внешнего кварцевого генератора или резонатора
2	2	OSC3	выход	Выход генератора для подключения RC цепи (в режиме RC генератора)
<b>Порта А</b>				
14	13	PA0/INT	вход	Вывод порта А, разряд 0/ Вход внешнего прерывания. Только входной контакт
15	14	PA1/T0CLK	вход	Вывод порта А, разряд 1/ Вход тактового сигнала для таймера 0 и внешнего прерывания (T0CKIF). Только входной контакт
18	16	PA2/RX/DT	вход/выход	Вывод порта А, разряд 2/ Вход асинхронного приемника/ Вход (выход) линии данных в синхронном режиме USART
19	17	PA3/TX/CK	вход/выход	Вывод порта А, разряд 3/ Выход асинхронного передатчика/ Вход (выход) тактовых импульсов в синхронном режиме USART
20	18	PA4	вход/выход	Вывод порта А, разряд 4
21	19	PA5	вход/выход	Вывод порта А, разряд 5
22	21	PA6	вход/выход	Вывод порта А, разряд 6
23	22	PA7	вход/выход	Вывод порта А, разряд 7
<b>Линейный регулятор</b>				
5	6	Ussh	вход	Вход линейного регулятора
6	7	Uout	выход	Выход линейного регулятора
<b>Питание и управление</b>				
11, 24	11, 23	U <sub>cc</sub>	питание	Питание микроконтроллера (электрические несвязанное с линейным регулятором)
4, 8, 12	5, 8, 12	GND	общий	Общий
9	9	TEST	вход	Вывод используемый при программировании и тестировании микросхемы
10	10	MCLRn	вход	Вход внешнего сброса кристалла
3, 7, 13, 16, 17, 25, 26	3, 4, 15, 20	NC	–	Не используемые выводы/контактные площадки

## Структурная блок-схема микросхемы



**Рис. 1 – Структурная блок-схема**

### Указания по применению и эксплуатации

Незадействованные в схмотехническом решении входы микросхемы необходимо подключать к питанию или земле, при этом рекомендуется использовать токоограничивающие резисторы номиналом от 10 до 55 кОм.

Технологическая перемычка ТП1, расположенная между выводами 10-11, электрически соединена с крышкой корпуса, технологическая перемычка ТП2, расположенная между выводами 24-25, электрически соединена с обратной стороной кристалла микросхемы.

Технологические перемычки рекомендуется соединить с шиной «Общий».

При монтаже бескорпусных микросхем K1886BE71H4 на металлизированную монтажную площадку рекомендуется соединить ее с шиной "Общий".

## **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 ( $\leq 1$  МГц), обеспечивает низкое энергопотребление;
- XT – генератор с внешним кварцевым или керамическим резонатором (частота от 1 МГц до 10 МГц);
- EC – режим подачи внешнего тактового сигнала (конфигурация генератора по умолчанию);
- RC – RC генератор с частотой от 30 кГц до 2 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 MCLRn при нормальной работе микроконтроллера тактовый генератор не выключается.

Примечание – От момента подачи напряжения питания до загрузки конфигурационных разрядов (около 94 мс от подачи напряжения питания) генератор находится в следующем режиме:

- 1886BE7У – EC-режим;
- 1886BE71У – LF-режим.

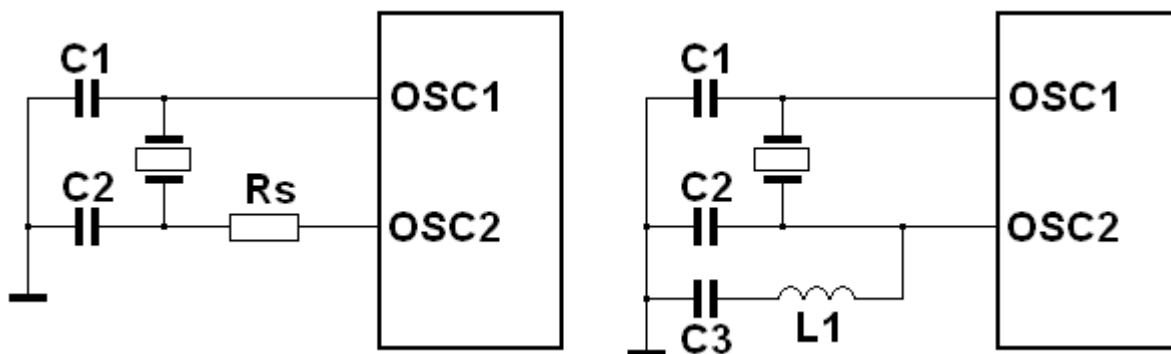
Выбор начального режима у 1886BE71У в пользу LF позволяет быстрее запустить низкочастотный кварцевый резонатор.

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

В режимах тактового генератора XT или LF, кварцевый или керамический резонатор подсоединяется к выводам OSC1 и OSC2 (выводы OSC3 и OSC4 допускается оставлять не подключенными). Генератор требует использования кварцевых резонаторов с параллельным резонансом. Использование резонаторов с последовательным резонансом может привести к получению тактовой частоты не соответствующей параметрам резонатора.

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. 2** – 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.

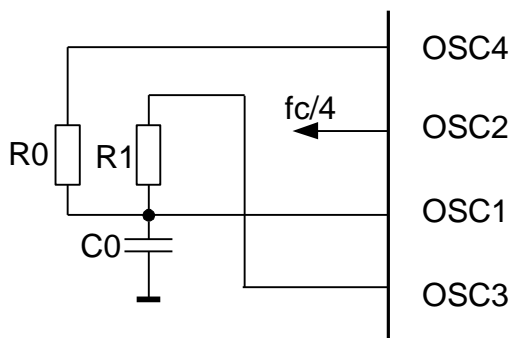
**Внешний тактовый генератор**

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**

На кристалле реализована схема температурно стабильного RC генератора (температурная стабильность не хуже  $50 \cdot 10^{-6} \text{ 1}^{\circ}\text{C}$ ). Генератор обладает стабильностью по напряжению питания в диапазоне от 4,5 до 5,5 В не хуже 0,4%. Частота RC генератора зависит от значений подключенных внешних сопротивления и емкости. 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. 3 показана схема подключения RC цепочки к микроконтроллеру. Рекомендуется выбирать величину сопротивления резистора от 3 до 100 кОм. Рекомендуется использовать конденсатор с емкостью более 20 пФ для стабильной работы генератора. Рекомендуемые значения емкости и сопротивления для получения различных рабочих частот приведены в разделе Типовые зависимости.

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 и т.д.



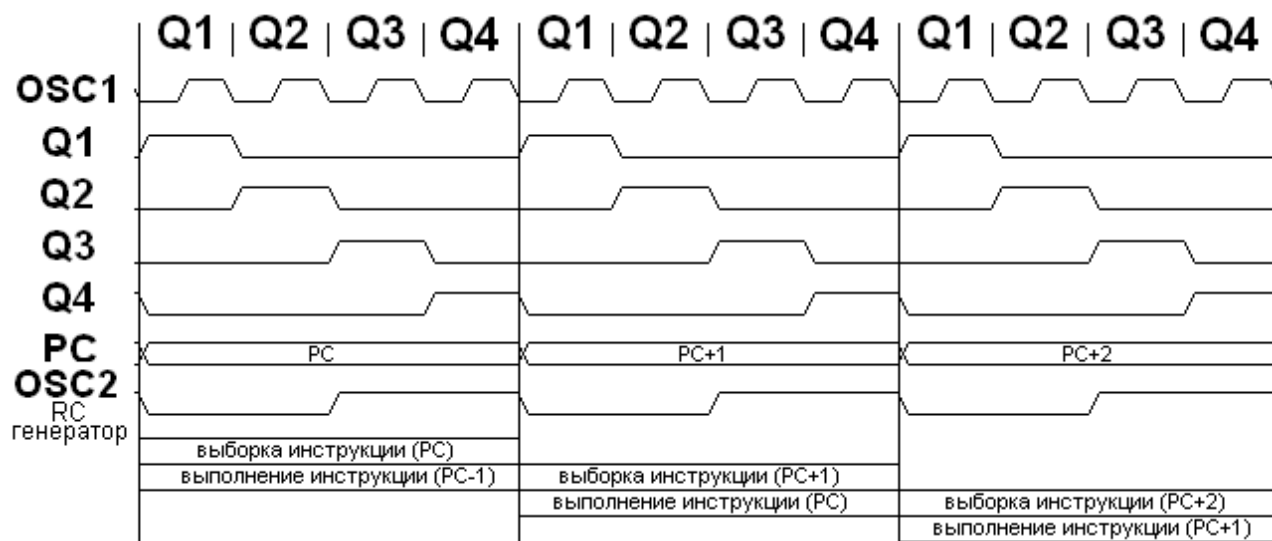
**Рис. 3** – Схема включения в режиме RC генератора

### Синхронизация выполнения команды

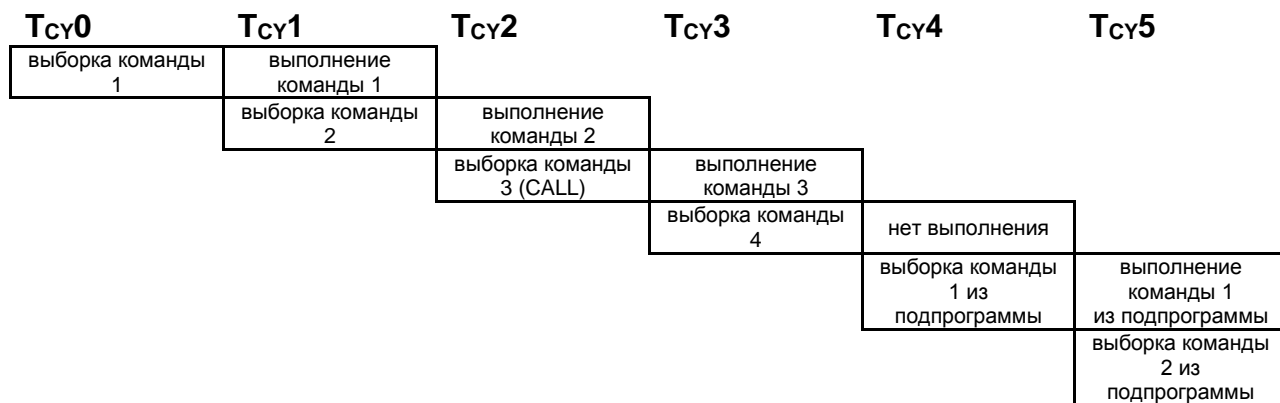
Выход тактового сигнала разделяется на 4 непересекающихся квадратурных тактовых сигнала, именуемых Q1, Q2, Q3, Q4. Программный счетчик (PC) увеличивается в такте Q1, а выборка команды из программной памяти и сохранение ее в регистре команд происходит по синхросигналу Q4. Команда декодируется и выполняется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

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. 2). 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. 4** – Синхронизация выполнения команды



**Рис. 5 – Конвейерное выполнение команд**

### Схема сброса микроконтроллера

Микроконтроллеры различают следующие виды сброса:

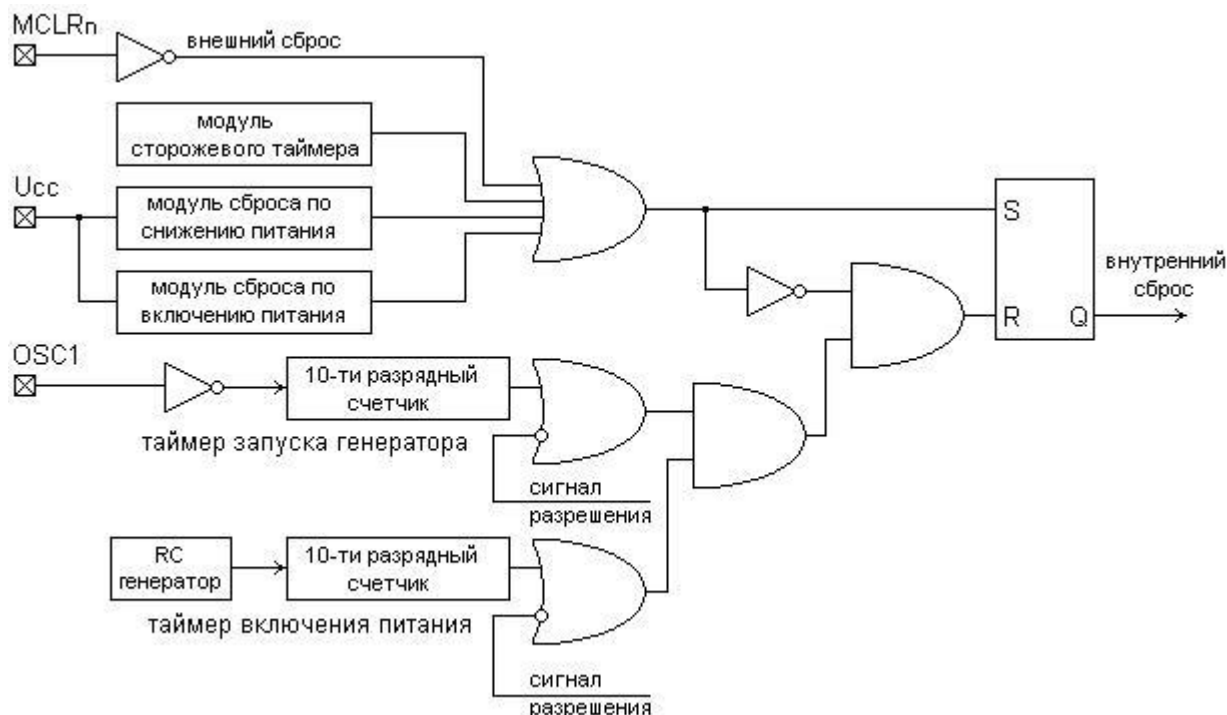
- сброс по включению питания;
- сброс по снижению напряжения питания;
- сброс по внешнему сигналу MCLRn;
- сброс по переполнению сторожевого таймера.

*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 показывает Таблица 3. Эти биты в соединении с битами POR и BOR, используются в программном обеспечении для определения вида сброса. Таблица 5 представляет 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.

В состоянии «сброс» выход внутреннего тактового сигнала соответствует такту Q1.

Упрощенная блок-схема схемы сброса приведена на Рис. 6.

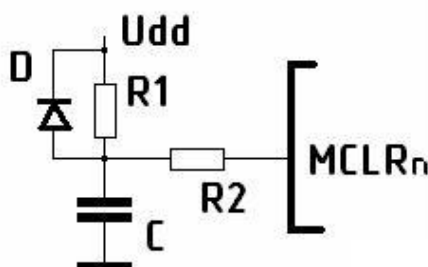


**Рис. 6** – Упрощенная блок-схема встроенной схемы сброса

### 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 MCLRn. В этом случае вывод MCLRn подключается через резистор или напрямую к напряжению питания. Внешняя схема «сброса» (см. Рис. 7) потребуется только в случае низкой скорости нарастания напряжения 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 MCLRn, не превышало 0.2 В). Резистор R2 предназначен для ограничения тока через вывод MCLRn, рекомендуемая величина 100 Ом – 1 кОм.



**Рис. 7** – Внешняя схема «сброса» по включению питания

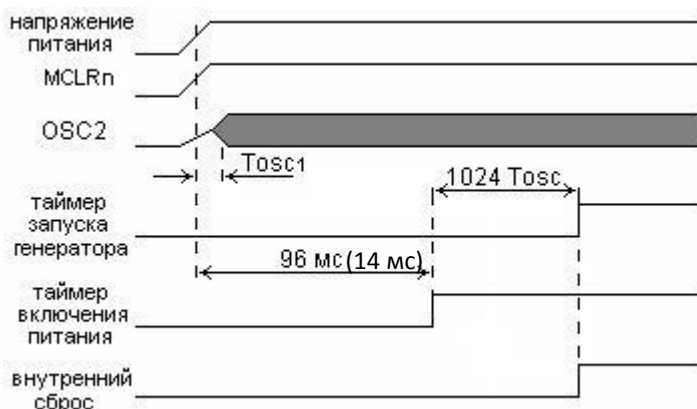
### **Таймер включения питания PWRT**

Таймер включения питания обеспечивает задержку включения (номинальное значение 96 мс – для 1886BE7 и 14 мс – для 1886BE71) по сигналу схемы сброса включения питания. Это происходит после фронта внутреннего сигнала «сброса», или после фронта сигнала MCLRn. 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 включения питания и таймер запуска генератора работают последовательно. Сначала запускается таймер включения питания, затем таймер запуска генератора. 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 Рис. 8 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.  $T_{osc1}$  – время, требуемое кварцевому генератору для запуска.



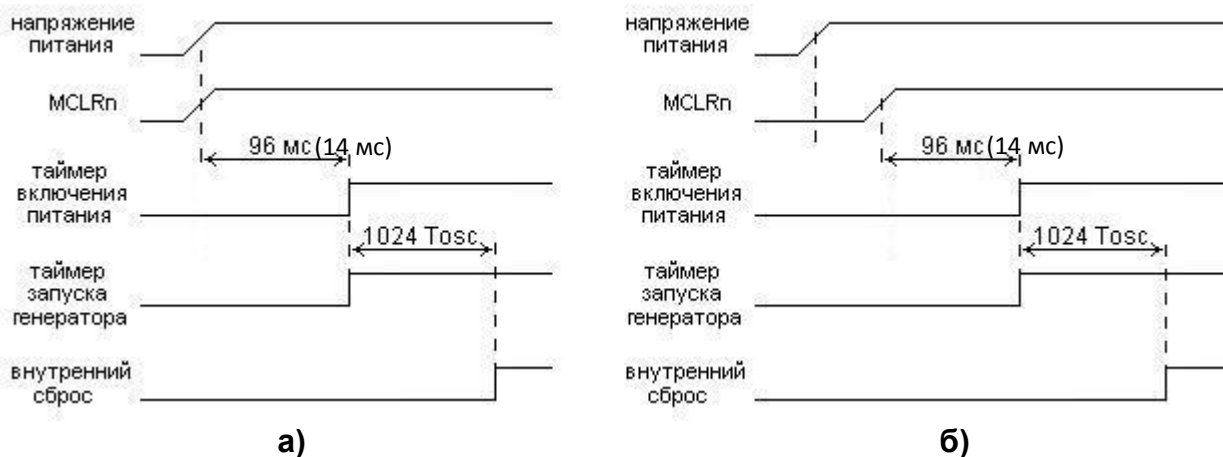
**Рис. 8** – Время запуска генератора. Последовательность удержания микроконтроллера в состоянии сброса

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 MCLRn находится в высоком уровне, тогда начинает работать таймер включения питания, когда он отсчитает 96 мс (для 1886BE71У

это время составляет 14 мс), включается таймер запуска генератора, если MCLRn 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 (Таблица 2). На Рис. 9 отображены последовательности задержек.

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

Если сигнал сброса от MCLRn 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 2 мс).



**Рис. 9** – Последовательность удержания микроконтроллера в режиме сброса по включению питания:

- а)** MCLRn подсоединен к напряжению питания
- б)** MCLRn не подключен к напряжению питания

Ниже (Таблица 3, Таблица 4) показаны состояния после сброса для некоторых битов и специальных регистров, в то время как Таблица 5 показывает 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 MCLRn
XT, LF	сумма 96** мс и 1024·T <sub>c</sub>	1024·T <sub>c</sub>	2 мс
EC, RC	сумма 96** мс	-	2 мс

*Примечание* – \*\* - для 1886BE71У это время составляет 14 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 MCLRn (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 <sup>(3)</sup>	Задержка включения
Сброс по включению питания		0000h	--11 1100	Сумма 96 <sup>(5)</sup> мс и 1024*TC
Сброс по снижению напряжения питания		0000h	--11 1110	Сумма 96 <sup>(5)</sup> мс и 1024*TC
Сброс от nMCLR в режиме нормальной работы		0000h	--11 1111 <sup>(4)</sup>	2 мс
Сброс от nMCLR в режиме SLEEP		0000h	--11 1111 <sup>(4)</sup>	Большее из 2 мс или (только для режимов XT и LF) 1024*TC
Сброс от WDT в режиме нормальной работы		0000h	--11 0111 <sup>(4)</sup>	Нет
Сброс от WDT во время режима SLEEP		0000h	--11 0011 <sup>(4)</sup>	Для режимов XT и LF: 1024*TC
Выход из режима SLEEP по прерыванию	GLINTD установлен	PC + 1	--11 1011 <sup>(4)</sup>	Для режимов XT и LF: 1024*TC
	GLINTD сброшен	PC + 1 <sup>(1)</sup>	--10 1011 <sup>(4)</sup>	Для режимов XT и LF: 1024*TC

*Обозначения:*

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

*Примечания:*

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.
5. Для 1886BE71У это время составляет 14 мс

**Таблица 5**  
Значения регистров при «сбросе»

Регистр	Адрес	Сброс по включению или снижению напряжения питания	Сброс от MCLR <sub>n</sub> 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	iiii iiiii	iiii iiiii
PCL	02h	0000h	0000h	PC+1 <sup>(2)</sup>
PCLATH	03h	0000 0000	0000 0000	iiii iiiii
ALUSTA	04h	1111 0000	1111 0000	1111 iiiii
T0STA	05h	0000 000-	0000 000-	0000 000-
CPUSTA <sup>(3)</sup>	06h	--11 11q0	--11 qq1u	--uu qqii
INTSTA	07h	0000 0000	0000 0000	iiii iiiii <sup>(1)</sup>
INDF1	08h	N/A	N/A	N/A
FSR1	09h	0000 0000	iiii iiiii	iiii iiiii
WREG	0Ah	0000 0000	0000 0000	iiii iiiii
TMR0L	0Bh	0000 0000 <sup>(5)</sup>	0000 0000	iiii iiiii
TMR0H	0Ch	0000 0000 <sup>(5)</sup>	0000 0000	iiii iiiii
TBLPTRL	0Dh	0000 0000	0000 0000	iiii iiiii
TBLPTRH	0Eh	0000 0000	0000 0000	iiii iiiii
BSR	0Fh	0000 0000	0000 0000	iiii iiiii
<b>Банк 0</b>				
-	10h	-	-	-
LINCNTR	11h	0000 0000	0000 0000	iiii iiiii
LINBRG	12h	0000 0000	0000 0000	iiii iiiii
RCSTA	13h	0000 -000	0000 -000	iiii -iiii
RCREG	14h	0000 0000	0000 0000	iiii iiiii
TXSTA	15h	0000 --10	0000 --10	iiii --ii
TXREG	16h	0000 0000	0000 0000	iiii iiiii
SPBRG	17h	0000 0000	0000 0000	iiii iiiii
<b>Банк 1</b>				
DDRA	10h	1111 11--	1111 11--	iiii ii--
PORTA	11h	xxxx xxxx	iiii iiiii	iiii iiiii
<b>Банк 2</b>				
-	-	-	-	-
<b>Банк 3</b>				
-	-	-	-	-
<b>Банк 4</b>				
-	-	-	-	-



## Спецификация 1886BE7(71)У, K1886BE7(71)У, K1886BE71H4

Регистр	Адрес	Сброс по включению или снижению напряжения питания	Сброс от MCLRn или от сторожевого таймера	Выход из режима SLEEP по прерыванию
<b>Банк 5</b>				
PIR1	10h	0000 0010	0000 0010	iiii iiiii(1)
PIE1	11h	0000 0000	0000 0000	iiii iiiii
-	12h			
-	13h			
EE_CONT	14h	0000 0000	0000 0000	iiii iiiii
EE_MODE	15h	0000 0000	0000 0000	iiii iiiii
EE_DATA	16h	0000 0000	0000 0000	iiii iiiii
EE_ADR	17h	0000 0000	0000 0000	iiii iiiii
<b>Банк 6</b>				
DBH	10h	0000 0000	0000 0000	iiii iiiii
DBL	11h	0000 0000	0000 0000	iiii iiiii
-	12h	-	-	iiii iiiii
EEDIV	13h	0000 0000	0000 0000	iiii iiiii
<b>Банк 7</b>				
-	-	-	-	-
<b>Банк 14</b>				
EECON	11h	0000 0000	0000 0000	iiii iiiii
<b>Банк 15</b>				
-	10h	-	-	-
-	11h	-	-	-
EDLSB	12h	0000 0000	0000 0000	iiii iiiii
EDMSB	13h	0000 0000	0000 0000	iiii iiiii
EEMOD	14h	0000 0000	0000 0000	iiii iiiii
EAMSB	15h	0--- -000	0--- -000	u--- -iiii
EALSB	16h	0000 0000	0000 0000	iiii iiiii
CFREG	17h	-111 1111	-111 1111	-iiii iiiii
<b>Вне банка</b>				
PRODL	18h	0000 0000	0000 0000	iiii iiiii
PRODH	19h	0000 0000	0000 0000	iiii iiiii

**Обозначение:**

i = не изменяется, x = неизвестно, - = не реализовано, читается как «0»,

q = значение зависит от условия.

**Примечание:**

1. Один бит или более в INTSTA, PIR1 будет изменен (чтобы произошел выход).
2. Когда выход из режима SLEEP происходит по прерыванию и бит GLINTD сброшен, PC загружается вектором прерывания.  
Таблица 3 приводит значения по сбросу в особых условиях.
3. Это значение, которое будет в триггере-защелке порта вывода.
4. При любом типе сброса устройства эти выводы конфигурируются как входы.
5. Счетчик TMR0 в 1886BE71У после старта контроллера при выборе режима генератора LF или XT содержит значение равное количеству тактов встроенного RC генератора прошедших от момента загрузки конфигурационных разрядов (это происходит на 14 мс от подачи питания) до

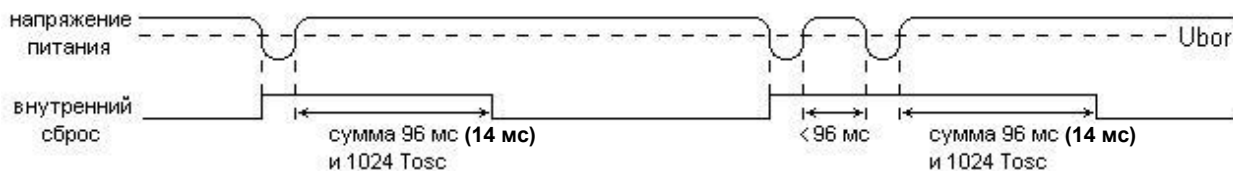
начала исполнения программы. По данному значению можно оценить время понадобившееся на старт кварцевого генератора.

### **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 схема 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 включаются таймер включения питания и таймер запуска генератора (для режимов LF и XT). Это удерживает микроконтроллер в состоянии сброса время, равное сумме 96 мс (для 1886BE71У это время составляет 14 мс) и  $1024 \cdot T_{osc}$ . 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. 10 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. 10 – Сброс по снижению напряжения питания**

## **Прерывания**

Микроконтроллеры имеют 6 источников прерывания:

- Внешнее прерывание от вывода PA0/INT.
- Переполнение таймера TMRO.
- Буфер передатчика USART пуст.
- Буфер приемника USART заполнен.
- Прерывания от модуля EERPOM.
- Изменение сигнала на выводе PA1/T0CLK(только в случае тактирования таймера TMRO от этого вывода).

Прерываниями управляют четыре регистра: CPUSTA, INTSTA, PIE1, PIR1. 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 PC 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 TMRO.
- Изменение сигнала на выводе PA1/T0CLK.
- Любое периферийное прерывание.

Флаг запроса прерывания должен быть сброшен в программе 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.

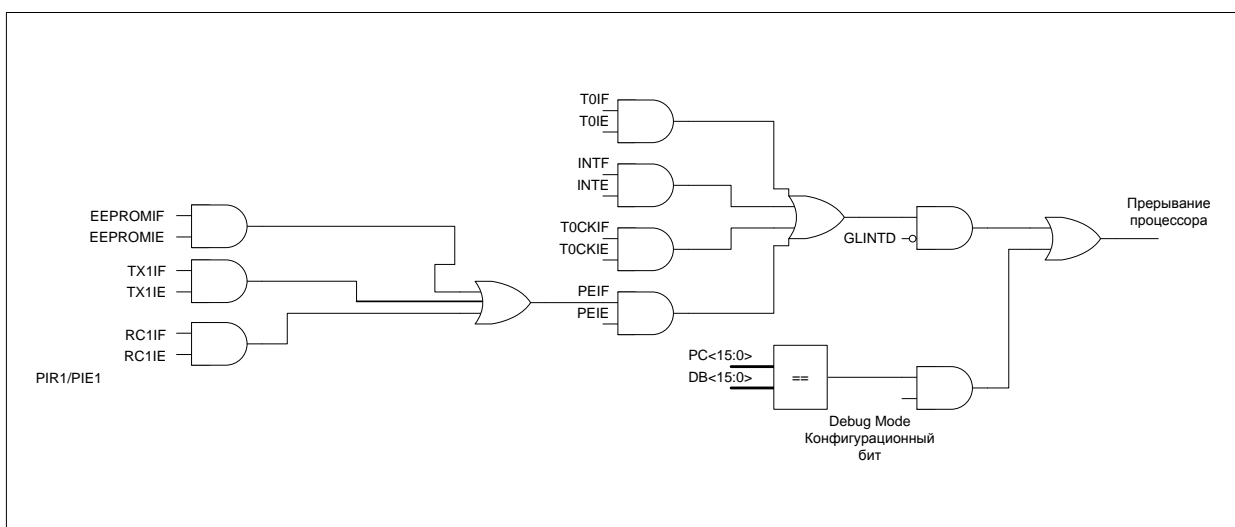
Возврат из программы обработки прерываний производится по команде RETFIE. При выполнении команды происходит восстановление программного счетчика (PC) из стека и сбрасывается бит GLINTD (чтобы разрешить прерывания).

Логика работы контроллера прерываний представлена на Рис. 10.

## **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.

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. 11 – Структурная схема логики прерываний**

### Регистр состояния прерываний

**Таблица 6**

INTSTA.ADR = 0x07, Банк доступа BANK = UNBANKED

Номер	7	6	5	4	3	2	1	0
Доступ*	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса**	0	0	0	0	0	0	0	0
	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	TOIE	INTE

\*

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

**Таблица 7**

<b>№</b>	<b>Функциональное имя бита</b>	<b>Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений</b>
7	PEIF	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 00201 = есть не обработанное периферийное прерывание; 0 = нет не обработанного периферийного прерывания.
6	T0CKIF	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.
5	T0IF	Флаг запроса прерывания по переполнению таймера 0. Сбрасывается программно программой обработки прерывания. Адрес вектора 10h. 1 = таймер TMR0 переполнен; 0 = таймер TMR0 не переполнен.
4	INTF	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.
3	PEIE	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.
2	T0CKIE	Бит разрешения внешнего прерывания от вывода PA1/T0CLK. 1 = прерывание разрешено; 0 = прерывание запрещено.
1	T0IE	Бит разрешения прерывания по переполнению таймера 0. 1 = прерывание разрешено; 0 = прерывание запрещено.
0	INTE	Бит разрешения прерывания от вывода PA0/INT. 1 = прерывание разрешено; 0 = прерывание запрещено.

**Регистр разрешения периферийных прерываний PIE1**

**Таблица 8**

PIE1.ADR = 0x11, Банк доступа BANK = 0x05

Номер	7	6	5	4	3	2	1	0
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса**	0	0	0	0	0	0	0	0
	-	-	-	-	-	EEPROMIE	TXIE	RCIE

\*

R/W - бит доступен на чтение и запись;

RO - бит доступен только на чтение;

U - бит физически не реализован или зарезервирован.

**Таблица 9**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7-3	-	Не используются
2	EEPROMIE	Бит разрешения прерывания от EEPROM. 1 = прерывание разрешено; 0 = прерывание запрещено.
1	TXIE	Бит разрешения прерывания от передатчика USART (буфер передатчика пуст). 1 = прерывание разрешено; 0 = прерывание запрещено.
0	RCIE	Бит разрешения прерывания от приемника USART (в буфере приемника есть данные). 1 = прерывание разрешено; 0 = прерывание запрещено.

Примечание:

\* Кроме данного бита в регистрах контроллера есть флаги разрешения прерываний для различных событий в контроллере.

**Регистр запроса периферийных прерываний PIR1**

Этот регистр содержит 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 10**

PIR1.ADR = 0x10, Банк доступа BANK = 0x05

Номер	7	6	5	4	3	2	1	0
Доступ*	U	U	U	U	R/W	R/W	R/W	R/W
Значение после сброса**	0	0	0	0	0	0	1	0
	-	-	-	-	DEBUGIF	EEPROMIF	TXIF	RCIF

\*

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

**Таблица 11**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7-4		Не используются
3	DEBUGIF	Флаг прерывания при совпадении содержимого регистра РС с регистрами DBH и DBL (функционирует только в режиме отладки см. раздел Регистры конфигурации микроконтроллера) 1 = есть прерывание 0 = нет прерывания
2	EEPROMIF	Флаг прерывания от EEPROM 1 = есть прерывание 0 = нет прерывания
1	TXIF	Флаг запроса прерывания от передатчика USART. Флаг устанавливается и сбрасывается аппаратно, доступен только для чтения. 1 = буфер передатчика USART пуст; 0 = буфер передатчика USART заполнен.
0	RCIF	Флаг запроса прерывания от приемника USART. Флаг устанавливается и сбрасывается аппаратно, доступен только для чтения. 1 = в буфере приемника USART есть данные; 0 = буфер приемника USART пуст.

### Регистры адреса точки останова DBL и DBH

Эти регистры содержат адрес точки останова. При совпадении содержимого РС со значением этих регистров процессор останавливается и переходит в програму отладчик.

**Таблица 12**

DBL.ADR = 0x11, Банк доступа BANK = 0x06

Номер	7	6	5	4	3	2	1	0
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса**	0	0	0	0	0	0	0	0
	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0

\*

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

**Таблица 13**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7	DB<7:0>	Младший байт адреса точки остановки и перехода в программу отладчик

**Таблица 14**

DBH.ADR = 0x10, Банк доступа BANK = 0x06

Номер	7	6	5	4	3	2	1	0
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса**	0	0	0	0	0	0	0	0
	DB15	DB14	DB13	DB12	DB11	DB10	DB9	DB8

\*

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

**Таблица 15**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7	DB<15:9>	Старший байт адреса точки остановки и перехода в программу отладчик

### 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 5 векторов прерываний, что способствует сокращению задержки при обработке прерываний.



Вектор периферийных прерываний имеет множество источников прерываний. В программе обслуживания периферийного прерывания, источник прерывания можно определить проверкой флагов запроса прерывания. Флаги запросов периферийных прерываний должны быть сброшены в программе перед разрешением прерываний, чтобы избежать повторного вызова.

Микроконтроллер 1886BE7У имеет 5 векторов прерываний. Адреса векторов и их приоритеты показаны в Таблица 16. 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 16**

Приоритеты и адреса векторов прерываний

<b>Адрес</b>	<b>Вектор</b>	<b>Приоритет</b>
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)
0FE0h	Прерывание отладчика по совпадению PC=DB	-

**Примечания:**

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

**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 PA0/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. Формирование флага запроса прерывания T0CKIF происходит только в случае тактирования таймера TMR0 от этого вывода, т.е. это прерывание от внешнего тактового сигнала таймера. Если TMR0 тактируется от внутренней тактовой частоты, то прерывание не формируется. Прерывание может быть запрещено сбросом бита T0CKIE (INTSTA<2>). Прерывание от вывода 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 доступен только для чтения и является объединением по «ИЛИ» всех флагов запросов периферийных прерываний, логически умноженных по «И» на соответствующие биты разрешения прерываний в регистре PIE1. Некоторые периферийные прерывания могут выводить микроконтроллер из режима SLEEP. Смотрите раздел описания режима SLEEP.

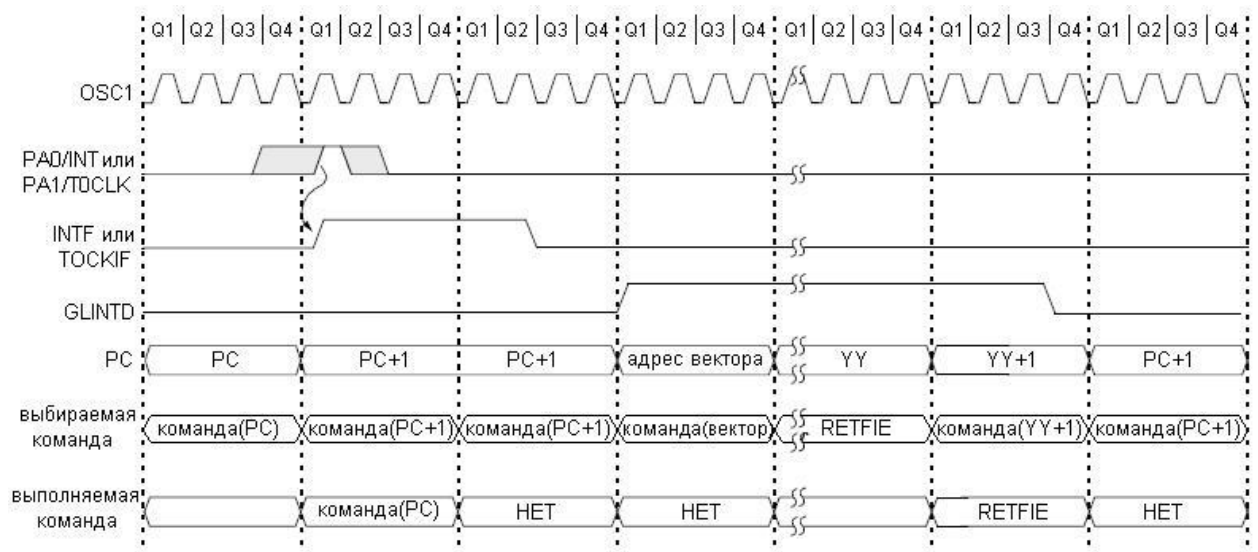
## Сохранение регистров при прерывании

Во время прерывания, в стеке сохраняется только значение PC для возврата. Сохранение других регистров необходимо реализовать программно.

Пример 1 показывает 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.

Пример 2 показывает 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.



**Рис. 12 – Временная диаграмма обработки прерываний от выводов PA0/INT или PA1/T0CLK**

**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_TEMPEQU 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	
	MOVFP FSR0, Nobank_FSR	; сохранение значения FSR, используемого
		; для сохранения BSR
	MOVFP WREG, WREG_TEMP	
	MOVFP Bank_FSR, FSR0	; восстановление значения FSR, используемого
		; 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
	MOVFP Nobank_FSR, FSR0	; восстановление значения FSR, используемого
		; для сохранения BSR
:		
;	Код программы обработки прерывания	
:		
POP	CLRF ALUSTA, F	; FSR0 - режим автодекрементирования
	MOVFP Bank_FSR, FSR0	; восстановление значения FSR, используемого
		; для сохранения других значений
	DECF FSR0, F	
	MOVFP INDF0, PCLATH	; чтение значения PCLATH
	MOVFP INDF0, WREG	; чтение значение WREG
	BSF ALUSTA, FS1	; FSR0 - режим без изменения значения регистра
	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	
;		
	RETIE	; 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, посвященным этим конкретным модулям периферии.

Вектор прерывания от вывода PA0/INT

Вектор прерывания от вывода PA1/T0CLK

0x7F0  
0x7FF

**Рис. 13** – 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. Таблица 16).

Объем внутренней памяти программ составляет 2К x 16 бит. Память перепрограммируемая EEPROM. Микроконтроллер может функционировать в

одной из 2 возможных конфигураций памяти программ. Конфигурация выбирается битами конфигурации. Возможны следующие режимы: микроконтроллер и защищенный микроконтроллер.

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

### **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 17 показывает организацию карты памяти данных.

Команды MOVPF и MOVFP обеспечивают перенос значений данных из области периферии («Р») в любое место в области регистров («F») и наоборот. «Р»-диапазон определен адресами от 00h до 1Fh, диапазон «F» – от 00h до FFh. Диапазон «Р» 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 переключение между этими банками существует команда «MOVL 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 18). 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 17**

Адрес	Не зависит от номера адресуемого банка.											
00h	INDF0											
01h	FSR0											
02h	PCL											
03h	PCLATH											
04h	ALUSTA											
05h	T0STA											
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	Банк 8	Банк 9	Банк 14	Банк 15
10h	-	DDRA				PIR1	DBH				-	-
11h	LIN CNTR	PORTA				PIE1	DBL				EECON	-
12h	LIN BRG						-					EDLSB
13h	RCSTA						EEDIV				-	EDMSB
14h	RCREG					EE_CONT					-	EEMOD
15h	TXSTA					EE_MODE					-	EAMSB
16h	TXREG					EE_DATA					-	EALSB
17h	SPBRG					EE_ADR					-	CFREG
	Не зависит от номера адресуемого банка.											
18h	PRODL											
19h	PRODH											
1Ah	Регистры общего назначения.											
1Fh	Регистры общего назначения.											
	Банк 0			Банк 1			Банк 2			Банк 3		
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.		

**Примечания:**

1. 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.
2. 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.
3. Чтение любого не существующего регистра дает значение равное нулю.

**Таблица 18**

Регистры специального назначения

Адрес	Название	бит 7	бит 6	бит 5	бит 4	бит 3	бит 2	бит 1	бит 0	POR, BOR	MCLRn, WDT
Не зависит от номера адресуемого банка.											
00h	INDF0	использует содержимое FSR0 для адресации памяти данных (физически не реализован)									
01h	FSR0	указатель 0 адреса, для косвенной адресации памяти данных								xxxx xxxx	uuuu uuuu
02h	PCL	младшие 8 бит счетчика команд								0000 0000	0000 0000

## Спецификация 1886BE7(71)У, K1886BE7(71)У, K1886BE71H4

03h	PCLATH <sup>1</sup>	регистр-защелка для старших 8 бит счетчика команд								0000 0000	uuuu uuuu
04h	ALUSTA	FS3	FS2	FS1	FS0	OV	Z	DC	C	1111 xxxx	1111 uuuu
05h	T0STA	INTEDG	T0SE	T0CS	T0PS3	T0PS2	T0PS1	T0PS0	-	0000 000-	0000 000-
06h	CPUSTA <sup>2</sup>	-	ESTBY	STKAV	GLINTD	TO	PD	POR	BOR	--11 11qq	--11 qquu
07h	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	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								xxxx xxxx	uuuu uuuu
0Ah	WREG	рабочий регистр								xxxx xxxx	uuuu uuuu
0Bh	TMR0L	младший байт регистра таймера 0								0000 0000 <sup>3</sup>	uuuu uuuu
0Ch	TMR0H	старший байт регистра таймера 0								0000 0000 <sup>3</sup>	uuuu uuuu
0Dh	TBLPTRL	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										xxxx xxxx	uuuu uuuu
11h	LINCNTR	BRK CNT3	BRK CNT2	BRK CNT1	BRK CNT0	BRK	SYNCH	ERR	LINEN	0000 0000	0000 0000
12h	LINBRG	Коэффициент скорости при приеме поля SYNCH								0000 0000	0000 0000
13h	RCSTA	SPEN	RX9	SREN	CREN	-	FERR	OERR	RX9D	0000 -000	0000 -000
14h	RCREG	регистр принимаемых данных последовательного порта USART								0000 0000	0000 0000
15h	TXSTA	CSRC	TX9	TXEN	SYNC	-	-	TRMT	TX9D	0000 --00	0000 --00
16h	TXREG	регистр передаваемых данных последовательного порта USART								0000 0000	0000 0000
17h	SPBRG	регистр управления генератором последовательного порта USART								0000 0000	0000 0000
<b>Банк 1</b>											
10h	DDRA	DDRA 7	DDRA 6	DDRA 5	DDRA 4	DDRA 3	DDRA 2	-	-	1111 11--	1111 11--
11h	PORTA	PORTA[7:0]								xxxx xxxx	uuuu uuuu
12h											
13h											
14h											
15h											
16h											
17h											
<b>Банк 2</b>											
10h											
11h											
12h											
13h											
14h											
15h											
16h											
17h											
<b>Банк 3</b>											
10h											



## Спецификация 1886BE7(71)У, K1886BE7(71)У, K1886BE71H4

11h														
12h														
13h														
14h														
15h														
16h														
17h														
<b>Банк 4</b>														
10h														
11h														
12h														
13h														
14h														
15h														
16h														
17h														
<b>Банк 5</b>														
10h	PIR1	-	-	-	-	DEBUG IF	EEPROM IF	TXIF	RCIF	----	0010	----	0010	
11h	PIE1	-	-	-	-	-	EEPROM IE	TXIE	RCIE	----	-000	----	-000	
12h														
13h														
14h	EE_CON T	TEST_P	EETEST	CPTTEST	VEE2	VEE1	BRG2	BRG1	BRG0	0000	0000	0000	0000	
15h	EE_MODE	EN_EE	-	IEBUSY	BUSY	-	MODE2	MODE1	MODE0	0-00	-000	0-00	-000	
16h	EE_DATA	Данные / команда									0000	0000	0000	0000
17h	EE_ADR	Адрес записи / чтения данных									0000	0000	0000	0000
<b>Банк 6</b>														
10h	DBH	Старший байт адреса останова									0000	0000	0000	0000
11h	DBL	Младший байт адреса останова									0000	0000	0000	0000
12h	-													
13h	EEDIV	Регистр коэффициента деления схемы записи/стирания памяти программ									0000	0000	0000	0000
14h														
15h														
16h														
17h														
<b>Банк 14</b>														
10h	-													
11h	ECON	LOCK	ERROR	EET	EBW	EBE	EER	EWR	ERD	0000	0000	0000	0000	
12h	-													
13h	-													
14h	-													
15h	-													
16h	-													
17h	-													
<b>Банк 15</b>														
10h	-													
11h	-													
12h	EDLSB	ED[7:0]									0000	0000	0000	0000
13h	EDMSB	ED[15:8]									0000	0000	0000	0000
14h	EEMOD	STATE	RESET	CPEN	HWS	AM	TM2	TM1	ESTBY	0000	0000	0000	0000	
15h	EAMSB	CPRDY	-	-	-	-	EA10	EA9	EA8	0---	-000	0---	-000	
16h	EALSB	EA[7:0]									0000	0000	0000	0000
17h	CFREG	-	DBG_E N	BODEN	PM0	WDT1	WDT0	FOSC1	FOSC0	-111	1111	-111	1111	
Не зависит от номера адресуемого банка.														
18h	PRODL	младший байт 16-ти битного результата (8x8 битное аппаратное умножение)									xxxx	xxxx	uuuu	uuuu
19h	PRODH	старший байт 16-ти битного результата (8x8 битное аппаратное умножение)									xxxx	xxxx	uuuu	uuuu

### 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>, его содержимое обновляется от, или записывается в старший байт счетчика команд.

2. Внешний сброс от вывода MCLRn не влияет на статусные биты TO и PD регистра CPUSTA.
3. Счетчик TMR0 в 1886BE7У после старта контроллера равен 16'h0000. Счетчик TMR0 в 1886BE71У после старта контроллера при выборе режима генератора LF или XT содержит значение равное количеству тактов встроенного RC генератора (см. раздел Таймер 0), а при выборе режима генератора RC или EC равен 16'h0000.

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 19**

ALUSTA.ADR = 0x04, Банк доступа BANK = UNBANKED

Номер	7	6	5	4	3	2	1	0
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса	1	1	1	1	0	0	0	0
	FS3	FS2	FS1	FS0	OV	Z	DC	C

\*

R/W - бит доступен на чтение и запись;

RO - бит доступен только на чтение;

U - бит физически не реализован или зарезервирован.

**Таблица 20**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7..6	FS[3:2]	биты выбора режима FSR1 00 = автодекремент величины FSR1 после выполнения команды 01 = автоинкремент величины FSR1 после выполнения команды 1x = значение FSR1 не изменяется
5..4	FS[1:0]	биты выбора режима FSR0 00 = автодекремент величины FSR0 после выполнения команды 01 = автоинкремент величины FSR0 после выполнения команды 1x = значение FSR0 не изменяется
3	OV	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
2	Z	флаг нуля 1 = результат арифметической или логической операции равен 0 0 = результат арифметической или логической операции не равен 0
1	DC	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).
0	C	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).

Регистр статуса ЦПУ (CPUSTA) содержит статусный и управляющий биты для ЦПУ. Этот регистр содержит бит GLINTD, 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 CPUSTA также показывает, заполнен ли стек 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 CPUSTA в качестве места назначения результата выполнения может быть отличным, нежели ожидалось.

Бит POR позволяет отличить сброс при включении питания от внешнего MCLRn 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 21**

CPUSTA.ADR = 0x06, Банк доступа BANK = UNBANKED

Номер	7	6	5	4	3	2	1	0
Доступ*	U	R/W	RO	R/W	RO	RO	R/W	R/W
Значение после сброса**	0	0	1	1	1	1	0	1
	-	ESLP	STKAV	GLINTD	TO	PD	POR	BOR

\*

R/W - бит доступен на чтение и запись;

RO - бит доступен только на чтение;

U - бит физически не реализован или зарезервирован.

**Таблица 22**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7	-	Зарезервировано
6	ESLP	Бит управления EEPROM памяти программ в режиме SLEEP. 0 = память не отключается при переходе в SLEEP режим, при выходе из режима память сразу готова к работе. 1 = память отключается при переходе в SLEEP режим, что уменьшает статический ток потребления, при выходе из режима требуется дополнительное время порядка 10 мкс для запуска схемы памяти.
5	STKAV	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)
4	GLINTD	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
3	TO	флаг переполнения сторожевого таймера 1 = устанавливается после включения питания или выполнения команд CLRWDT или SLEEP 0 = после переполнения сторожевого таймера
2	PD	флаг включения питания 1 = устанавливается после включения питания или выполнения команды CLRWDT 0 = после выполнения команды SLEEP
1	POR	флаг сброса при включении питания 1 = не было сброса при включении питания 0 = произошел сброс при включении питания (бит должен быть установлен программно)
0	BOR	флаг сброса по снижению напряжения питания Когда бит BODEN в регистре конфигурации установлен (разрешено): 1 = сброса при снижении питания не было 0 = произошел сброс при снижении питания (бит должен быть установлен программно) Когда бит BODEN в регистре конфигурации сброшен (запрещено): значение бита безразлично

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 INTF. Остальные биты конфигурируют таймер 0, его предделитель и источник тактовых сигналов.

**Таблица 23**

T0STA.ADR = 0x05, Банк доступа BANK = UNBANKED

Номер	7	6	5	4	3	2	1	0
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	U
Значение после сброса**	0	0	0	0	0	0	0	0
	INTEDG	T0SE	T0CS	T0PS3	T0PS2	T0PS1	T0PS0	-

\*

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

**Таблица 24**

<b>№</b>	<b>Функциональное имя бита</b>	<b>Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений</b>
7	INTEDG	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
6	T0SE	бит выбора управляющего перепада сигнала при внешнем тактировании таймера 0. Этот бит выбирает, на фронте или спаде сигнала таймер 0 будет инкрементироваться. Когда T0CS=0 (внешнее тактирование): 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 Когда T0CS=1 (внутреннее тактирование): значение бита безразлично
5	T0CS	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
4..1	T0PS[3:0]	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
0	-	Зарезервировано

### **Функционирование стека**

Микроконтроллеры имеют 16x16 бит аппаратный стек. 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.

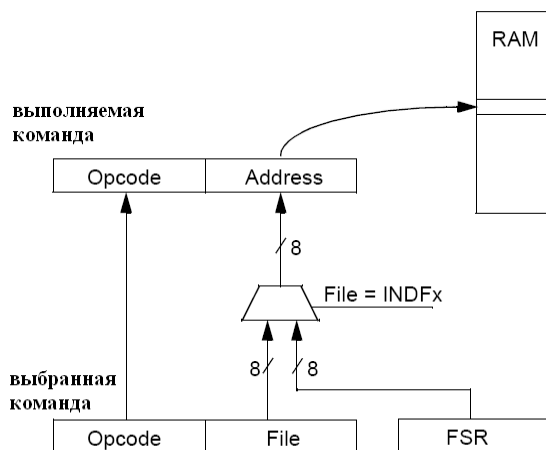
**Примечания:**

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. На Рис. 14 показан принцип косвенной адресации. Там показана модификация значения адреса памяти данных, значением регистра.



**Рис. 14 – 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 FS1и 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».

### **Модуль счетчика 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.

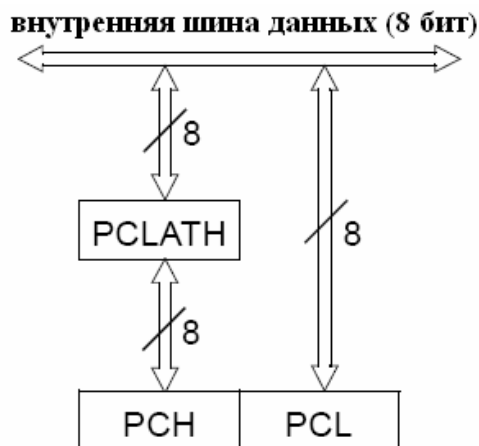
16-ти битный счетчик команд инкрементируется после выборки команды в течение цикла Q1 до тех пор пока:

- 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.

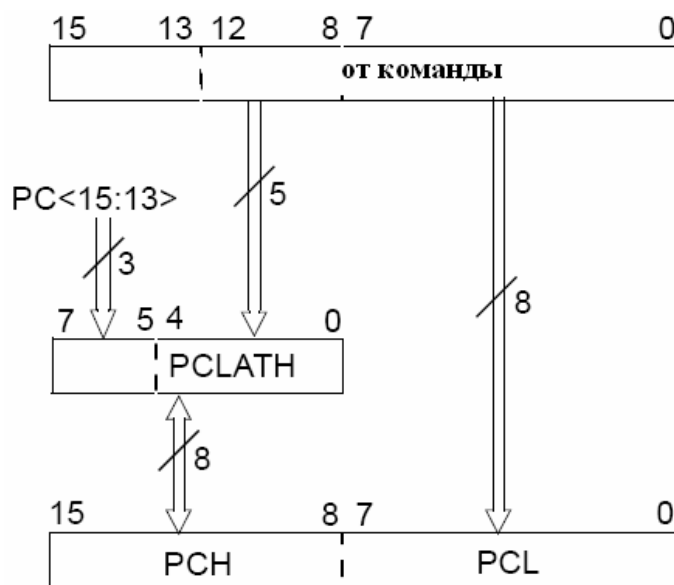
Эти «переходы» эквивалентны вынужденному циклу «NOP» с переходом по адресу. Рис. 15 и Рис. 16 показывают функционирование счетчика команд в различных ситуациях.

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>



**Рис. 15** – Функционирование счетчика команд



**Рис. 16** – Счетчик команд при выполнении инструкций Call и GOTO

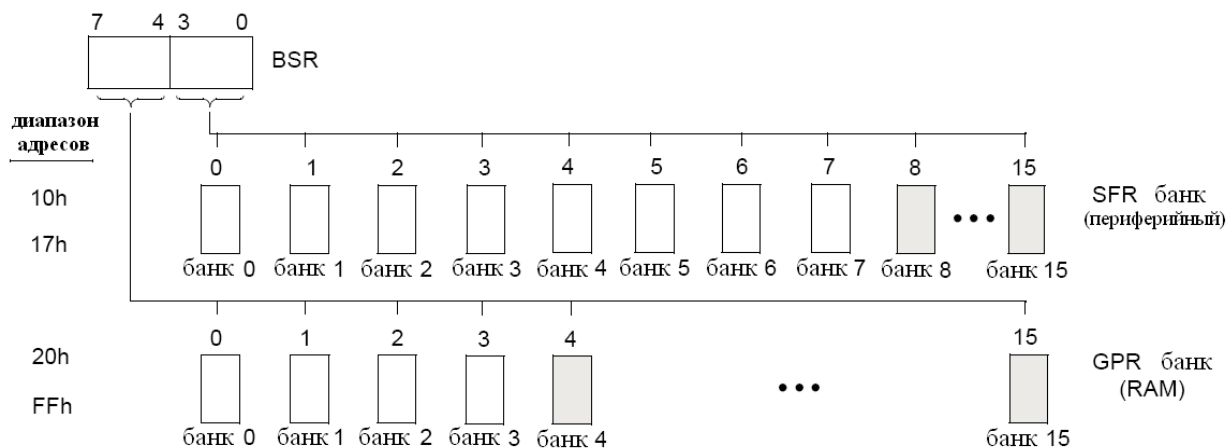
Команды чтения – модификации – записи воздействуют только на PCL. PCH загружается значением из PCLATH. Для примера ADDWF PCL,F приведет к переходу в пределах текущей страницы. Если PC=03F0h, WREG=30h и PCLATH=03h до начала действия команды, то после ее действия PC=0320h. Чтобы выполнить правильный 16-ти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

Следующие команды, связанные с 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)

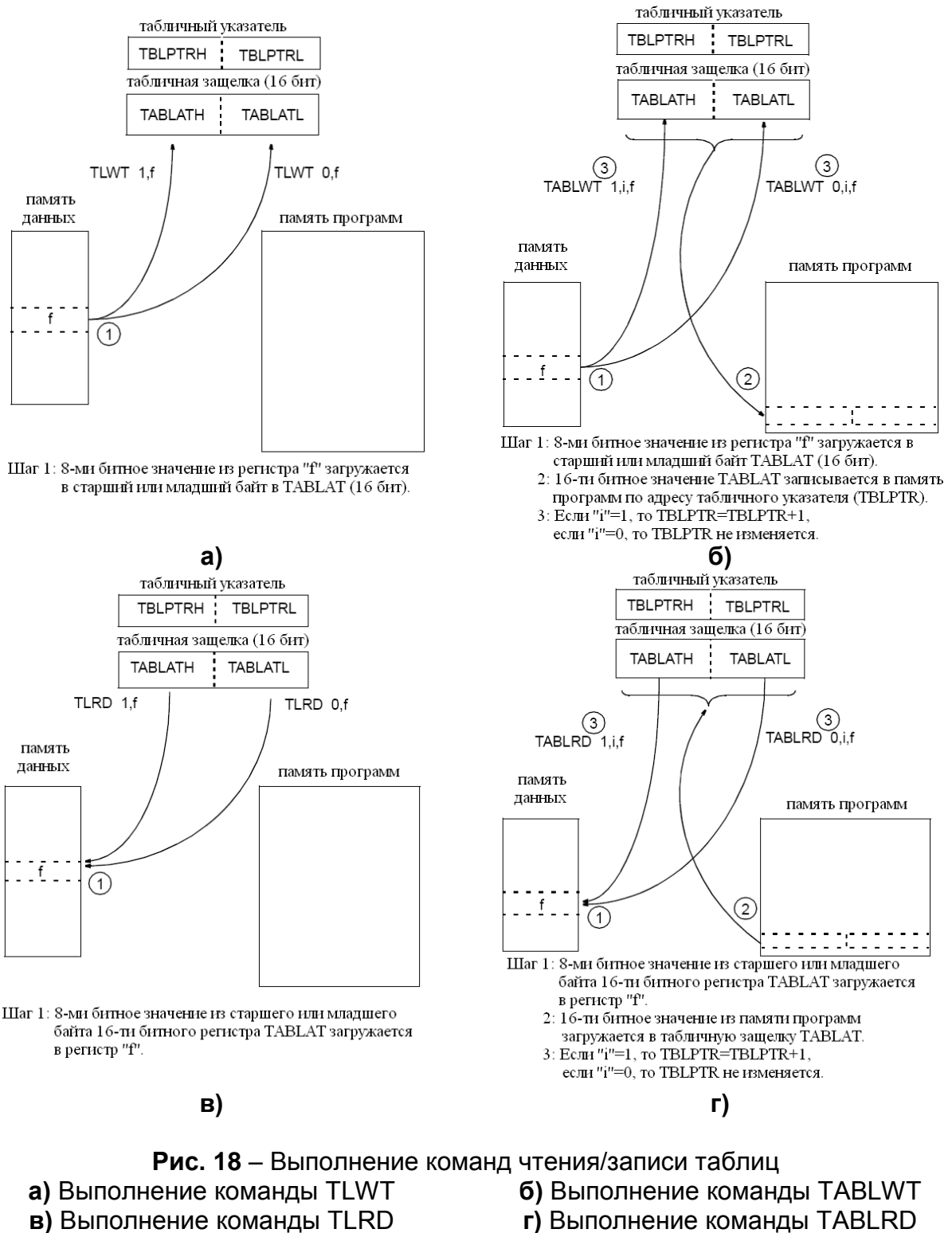
Регистр выбора банка BSR используется для переключения между банками в области памяти данных (Рис. 17). Младший полубайт используется для выбора банка периферийного регистра, для его записи используется команда «MOVLB bank». Старший полубайт используется для выбора банка памяти общего назначения (ОЗУ), 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. 17 – 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). Рис. 18 показывает выполнение этих 4 команд. Шаги показывают последовательность операций. Память программ может быть как внутренней, так и внешней (для режима «микропроцессор» или «расширенный микроконтроллер»).



**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 MCLRn 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 адресуется слово за пределами памяти программ, 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 **EEDIV** (коэффициент деления частоты генератора). Требуемое значение длительности 5 мс. Коэффициент деления выбирается:

$K = 5 \text{ мс} / 651 * T_{osc}$ , где  $T_{osc}$  в мс. Значение регистра после сброса равно 00h. Регистр доступен по чтению и записи.

**Таблица 25**

Воздействие прерываний на операцию «длинной записи»

источник прерываний	GLINTD	бит разрешения	флаг запроса	действие
РА0/INT, РА1/T0CLK, ТМRO(прим.), Периферийные прерывания	0	1	1	Заканчивает длинную запись таблиц во внутреннюю память программ, переходит к вектору прерывания.
	0	1	0	Нет.
	1	0	x	Нет.
	1	1	1	Заканчивает длинную запись таблиц, не переходит к вектору прерывания.

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

**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 4 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 4**  
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 программ в табличную защелку, инкрементирование TBLPTR
TLRD	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

**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 26).

Пример 5 показывает последовательность действий при 8x8 беззнаковом умножении. Требуется только одна команда, когда один аргумент для умножения уже загружен в регистр WREG.

Пример 6 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 7 и пример 8 приводят последовательность действий при 16x16 беззнаковом умножении. В примере 8 приводится используемый алгоритм. 32-х битный результат хранится в 4-х регистрах, RES3:RES0.

Пример 9 и пример 10 приводят 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 26**  
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	29.333	60.50	121.0
	аппаратный умножитель	24	24	2.91	6.0	12.0
16 x 16 со знаком	без аппаратного умножителя	52	254	30.788	63.50	127.0
	аппаратный умножитель	36	36	4.36	9.0	18.0

**Пример 5**

Программа 8 x 8 битного беззнакового умножения

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

**Пример 6**

Программа 8 x 8 битного умножения со знаком

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

**Пример 7**

Алгоритм 16 x 16 битного беззнакового умножения

RES3:RES0 = ARG1H:ARG1L * ARG2H:ARG2L
---------------------------------------

$$= (\text{ARG1H} * \text{ARG2H} * 2^{16}) + (\text{ARG1H} * \text{ARG2L} * 2^8) + (\text{ARG1L} * \text{ARG2H} * 2^8) + (\text{ARG1L} * \text{ARG2L})$$

**Пример 8**

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 9**

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 10**

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**

Микроконтроллер имеет один порт ввода-вывода. Порт имеет регистр направления данных DDR, который используется для конфигурации выводов порта на вход или на выход.

Когда выводы порта 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.

## **Регистр порта А и регистр направления данных DDRA**

«Порт А» 8-ми разрядный. По сигналу «сброс», выводы «порта А» принудительно конфигурируются, как «вход» с высоким входным сопротивлением. Выводы PA0 и PA1 всегда сконфигурированы как вход. Направление данных на выводах PA2 и PA3, контролируется периферийным модулем или регистром DDRA. Направление данных на выводах PA4 – PA7 контролируется регистром DDRA. По сигналу «сброс», периферийный модуль неактивен, при этом выводы переведены в состояние «вход». При чтении «порта А» считывается состояние с выводов.

Вывод PA0 может работать как обычный вход или как вход внешнего прерывания. Вывод PA1 может работать как обычный вход или как вход тактового сигнала для «таймера 0». Выводы PA2 и PA3 мультиплексированы с периферийным модулем USART.

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 27**

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/RX/DT	2		Пользовательский вывод/Вход или вход приемника асинхронного USART, или вход/выход данных синхронного USART.
PA3/TX/CK	3		Пользовательский вывод/Вход или выход асинхронного передатчика USART, или вход/выход синхросигнала синхронного USART.
PA4	4		Пользовательский вывод
PA5	5		Пользовательский вывод
PA6	6		Пользовательский вывод
PA7	7		Пользовательский вывод

### 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.

Счетчик TMR0 в 1886BE71У позволяет также оценить время понадобившееся на старт кварцевого генератора. В начале исполнения программы контроллера (только при выборе режима генератора LF или XT) счетчик содержит значение равное количеству тактов встроенного RC генератора прошедших от момента загрузки конфигурационных разрядов (загрузка конфигурации происходит на 14 мс от подачи питания) до начала исполнения программы. За это время при старте микроконтроллера происходит запуск кварцевого резонатора и переполнение счетчика подсчитывающего 1024 такта частоты кварцевого резонатора. После начала исполнения программ функциональность таймера 0 будет определяться регистром T0STA. При выборе режима генератора EC или RC в начале исполнения программы контроллера счетчик равен 0, также как и во всех режимах генератора микроконтроллера 1886BE7У.

**Таблица 28**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
INTEDG	T0SE	T0CS	T0PS3	T0PS2	T0PS1	T0PS0	-
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>T0SE:</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, установка бита T0CKIF не производится						

<b>бит 5</b>	<b>T0CS:</b> бит выбора источника тактирования для «таймера 0». Этот бит выбирает источник синхронизации для «таймера 0». 1 = внутренняя тактовая частота с генератора циклов, прерывание от вывода RA1/T0CKI не формируется; 0 = внешнее тактирование с вывода RA1/T0CLK, формирование прерывания от вывода RA1/T0CKI разрешено
<b>бит 4-1</b>	<b>T0PS3:T0PS0:</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.

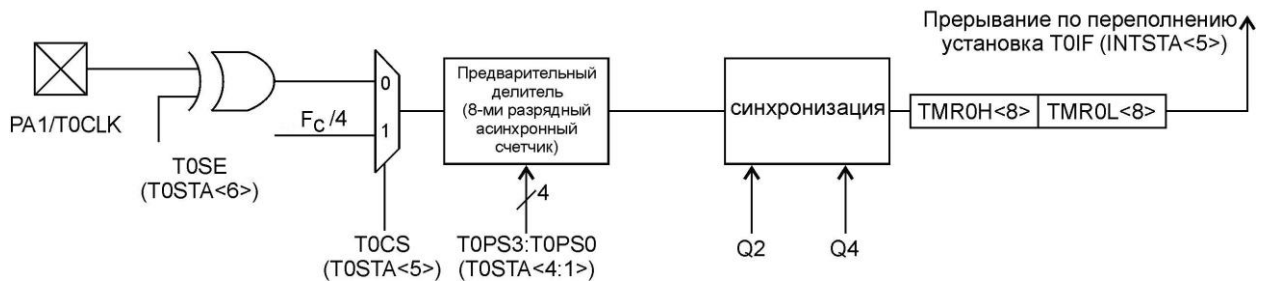
Если бит T0CS установлен в «1», то инкрементирование счетчика «таймера 0» происходит от тактовых импульсов внутреннего генератора, который является источником синхронизации для всего микроконтроллера. Если бит T0CS сброшен в «0», то инкрементирование счетчика происходит от тактовых импульсов с входа RA1/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 RA1/T0CLK, а если бит T0SE сброшен в «0», то по заднему фронту (спаду) сигнала RA1/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 T0PS3:T0PS0. 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. 19 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. 20 показано, что эта задержка может составлять от 3 T<sub>c</sub> до 7 T<sub>c</sub>.

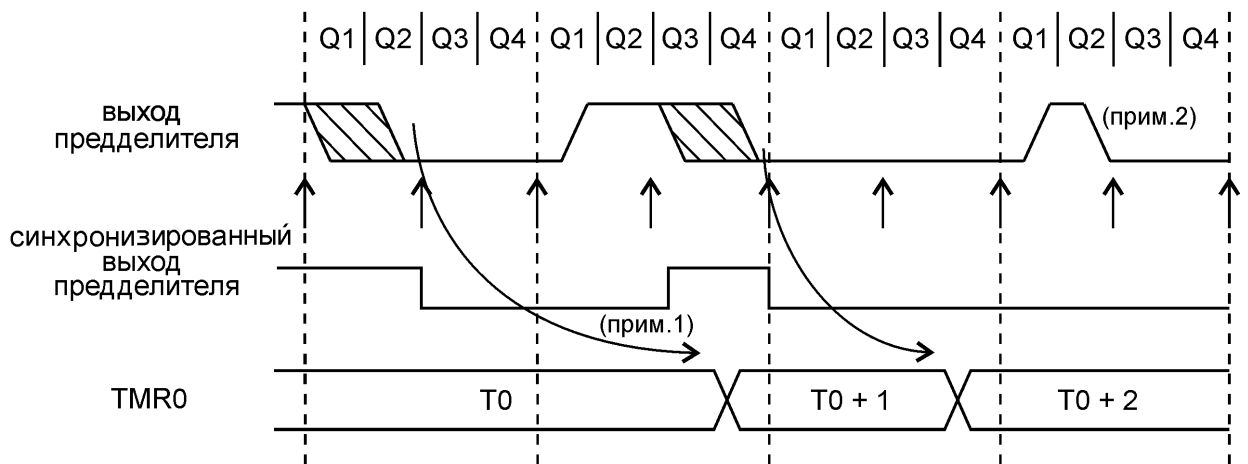
Проблема считывания 16-разрядного значения регистров TMR0L и TMR0H заключается в том, что после считывания младшего (или старшего) байта, его значение может измениться от FFh к 00h. Для обеспечения однозначного считывания состояния счетчика рекомендуется маскировать сигнал запроса на обработку прерывания.

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 TOSTA, то есть значение коэффициента может быть изменено «на лету» во время исполнения программы. Перед изменением коэффициента деления рекомендуется сбрасывать ПДПКД.



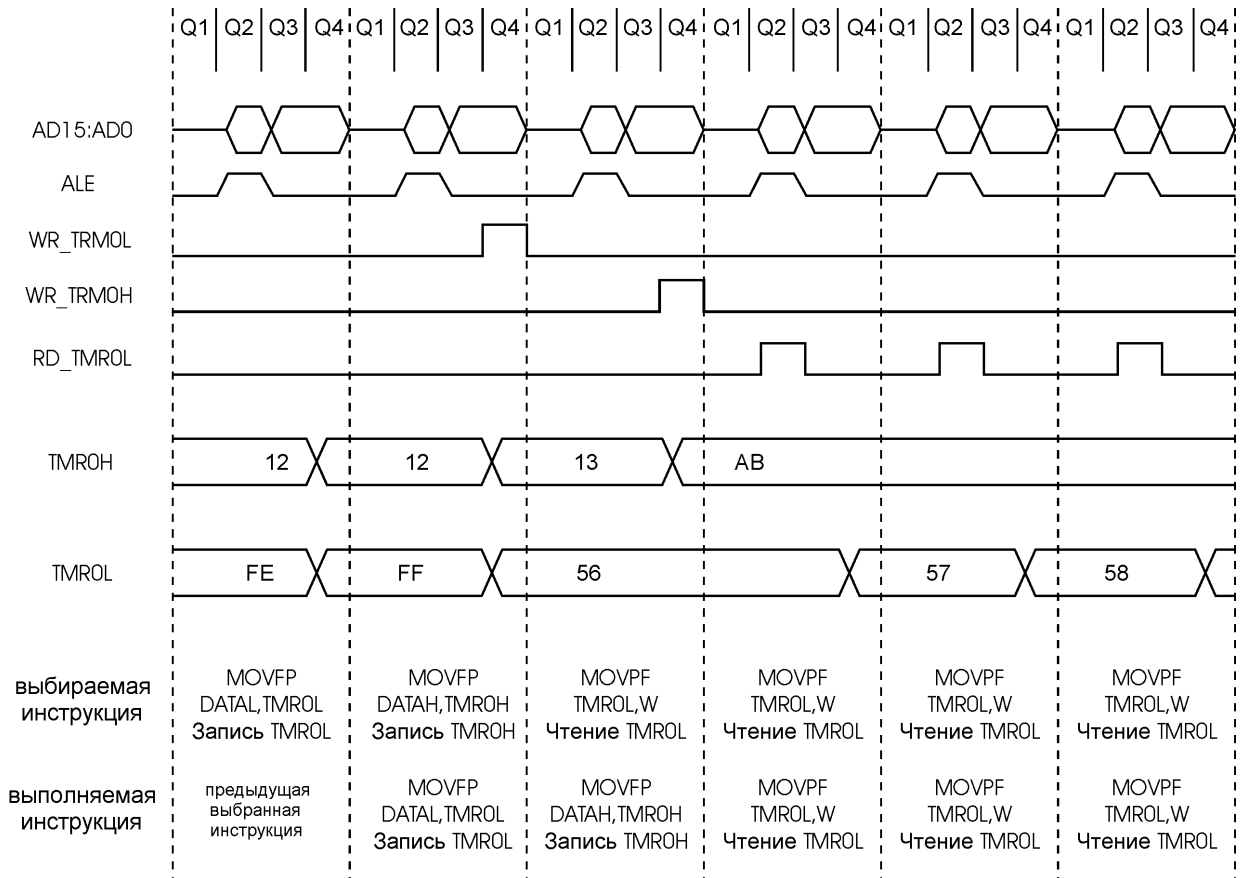
**Рис. 19** Блок-схема модуля «таймер 0»



**Рис. 20** 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. 21** Функционирование «таймер 0» при обращении к регистрам TMR0L и TMR0H

Примечание:

В примере записывается значение TMR0 равное AB56h.

## **Модуль универсального синхронно-асинхронных приемопередатчика с поддержкой LIN интерфейса**

Микроконтроллер содержит один модуль синхронно-асинхронных приемопередатчика с поддержкой LIN интерфейса USART/LIN. Универсальный синхронно-асинхронный приемопередатчик может работать в следующих режимах:

- асинхронный (полный дуплекс);
- асинхронный (полный дуплекс) с автоматическим приемом LIN заголовка;
- синхронный ведущий (полудуплекс);
- синхронный ведомый (полудуплекс).

Бит SPEN (RCSTA<7>) должен быть установлен, чтобы выводы PA2/RX/DT и PA3/TX/CK сконфигурировались как выводы последовательного интерфейса. Модуль USART/LIN будет управлять направлением выводов PA2 и PA3, в зависимости от состояния битов конфигурации в регистрах RCSTA и TXSTA. Следующие биты контролируют направление выводов: SPEN, TXEN, SREN, CREN, CSRC. При установке бита LINEN в асинхронном режиме работы, модуль автоматически принимает заголовок пакета (поля BREAK и SYNCH) с вычислением скорости передачи.



**Регистр режима и статуса работы приемника**

**Таблица 28**

RCSTA.ADR = 0x13, Банк доступа BANK = 0x00

Номер	7	6	5	4	3	2	1	0
Доступ*	R/W	R/W	R/W	R/W	U	RO	RO	R/W
Значение после сброса**	0	0	0	0	0	0	0	X
	SPEN	RX9	SREN	CREN	-	FERR	OERR	RX9D

\*

R/W - бит доступен на чтение и запись;

RO - бит доступен только на чтение;

U - бит физически не реализован или зарезервирован.

**Таблица 29**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7	SPEN	бит разрешения работы последовательного порта 1 = конфигурирует PA3/TX/CK и PA2/RX/DT как выводы последовательного порта USART 0 = последовательный порт отключен
6	RX9	выбор 9-ти разрядного приема 1 = выбирает 9-ти разрядный прием 0 = выбирает 8-ми разрядный прием
5	SREN	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
4	CREN	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
3	-	Не реализовано, читается как «0»
2	FERR	бит ошибки кадрирования 1 = есть ошибка (сбрасывается при чтении регистра)

		RCREG) 0 = нет ошибки
1	OERR	бит ошибки переполнения внутреннего буфера. 1 = есть ошибка (сбрасывается при сбросе бита CREN) 0 = нет ошибки
0	RX9D	9-й бит принятых данных (может использоваться для программной реализации проверки четности)

**Регистр режима и статуса работы передатчика**

**Таблица 30**

TXSTA.ADR = 0x15, Банк доступа BANK = 0x00

Номер	7	6	5	4	3	2	1	0
Доступ*	R/W	R/W	R/W	R/W	U	U	RO	R/W
Значение после сброса**	0	0	0	0	0	0	1	X
	CSRC	TX9	TXEN	SYNC	-	-	TRMT	TX9D

\*

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

**Таблица 31**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7	CSRC	выбор источника тактовых импульсов <u>Синхронный режим:</u> 1 = режим ведущего (внутренний тактовый сигнал из BRG) 0 = режим ведомого (внешнего источника тактового сигнала) <u>Асинхронный режим:</u> Не имеет значения
6	TX9	выбор 9-ти разрядной передачи. 1 = выбирает 9-ти разрядную передачу 0 = выбирает 8-ми разрядную передачу
5	TXEN	разрешение передачи 1 = передача разрешена 0 = передача отключена Бит SREN/CREN подменяет TXEN в синхронном режиме.
4	SYNC	бит выбора режима USART (синхронный/ асинхронный) 1 = синхронный режим 0 = асинхронный режим
3..2	-	Не реализованы, читаются как «0»
1	TRMT	флаг заполненности сдвигового регистра передатчика (TSR) 1 = регистр пуст 0 = регистр заполнен

0	TX9D	9-й бит передаваемых данных (может использоваться для программной реализации проверки четности)
---	------	---

**Регистр режима и статуса работы приемника LIN заголовка**

**Таблица 32**

LINCNTR.ADR = 0x11, Банк доступа BANK = 0x00

Номер	7	6	5	4	3	2	1	0
Доступ*	RO	RO	RO	RO	R/W	R/W	R/W	R/W
Значение после сброса**	0	0	0	0	0	0	0	0
	BRK CNT3	BRK CNT2	BRK CNT1	BRK CNT0	BRK	SYNCH	ERR	LINEN

\*

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

**Таблица 33**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7..4	BRKCNT[3:0]	Длительность принятого поля BREAK, При приеме поля BREAK короче, чем 11 битовых интервалов вырабатывается сигнал ошибки (ERR)
3	BRK	Флаг окончания приема поля BREAK 1 = BREAK получен 0 = не получен
2	SYNCH	Флаг окончания приема поля SYNCH 1 = SYNCH получен 0 = не получен
1	ERR	Флаг возникновения ошибки при приеме заголовка LIN пакета. 1 = есть ошибка 0 = нет ошибки
0	LINEN	Сигнал разрешения работы приемника LIN заголовка <u>Асинхронный режим:</u> 1 = принимает заголовок LIN пакета 0 = отключен <u>Синхронный режим:</u> Не важно.

**Регистр данных приемника**

**Таблица 34**

RCREG.ADR = 0x14, Банк доступа BANK = 0x00

Номер	7	6	5	4	3	2	1	0
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса**	0	0	0	0	0	0	0	0

DATA7	DATA6	DATA 5	DATA4	DATA3	DATA 2	DATA1	DATA 0
-------	-------	-----------	-------	-------	-----------	-------	-----------

\*

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

**Таблица 35**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7..0	DATA[7:0]	Регистр данных: Данные полученные по USART.

**Регистр данных передатчика**

**Таблица 36**

TXREG.ADR = 0x16, Банк доступа BANK = 0x00

Номер	7	6	5	4	3	2	1	0
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса**	0	0	0	0	0	0	0	0
	DATA7	DATA6	DATA 5	DATA4	DATA3	DATA 2	DATA1	DATA 0

\*

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

**Таблица 37**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7..0	DATA[7:0]	Регистр данных: Данные полученные по USART.

**Регистр задания скорости приема и передачи**

**Таблица 38**

SPBRG.ADR = 0x17, Банк доступа BANK = 0x00

Номер	7	6	5	4	3	2	1	0
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса**	0	0	0	0	0	0	0	0
	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0

\*

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

**Таблица 39**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7..0	BRG[7:0]	SPBRG регистр определяет период (скорость) работы приемопередатчика

**Регистр скорости поля SYNCH в LIN фрейме**

**Таблица 40**

LINBRG.ADR = 0x12, Банк доступа BANK = 0x00

Номер	7	6	5	4	3	2	1	0
Доступ*	RO	RO	RO	RO	RO	RO	RO	RO
Значение после сброса**	0	0	0	0	0	0	0	0
	LIN BRG7	LIN BRG6	LIN BRG5	LIN BRG4	LIN BRG3	LIN BRG2	LIN BRG1	LIN BRG0

\*

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

**Таблица 41**

<b>№</b>	<b>Функциональное имя бита</b>	<b>Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений</b>
7..0	LINBRG[7:0]	LINBRG регистр содержит значение скорости определенное при приеме поля SYNCH заголовка LIN фрейма

## **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 (в том числе LIN) =  $F_{osc} / (32 * ((SPBRG) + 1))$ ;
- скорость передачи для синхронного режима =  $F_{osc} / (4 * ((SPBRG) + 1))$ , значение (SPBRG) от 0 до 255.

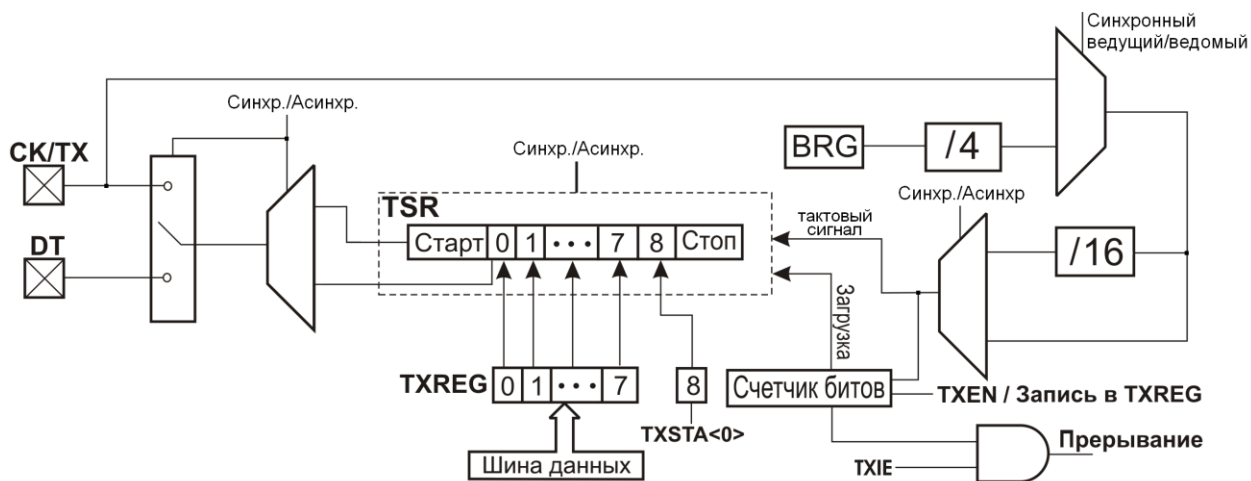
Запись нового значения в SPBRG приводит к сбросу таймера BRG. Это обеспечивает то, что генератор сразу переключается на новую скорость передачи. После сигнала «сброс» регистр SPBRG очищается, поэтому его необходимо загружать требуемым значением после каждого сброса.

## 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.
- Детектор LIN заголовка.



**Рис. 22** Блок-схема передатчика



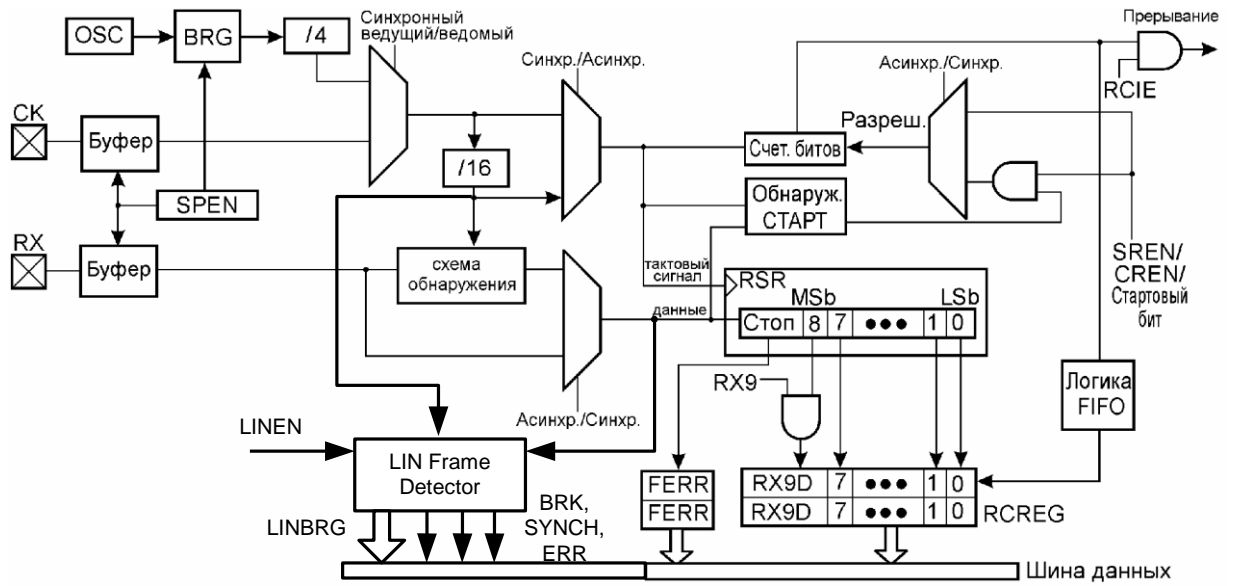


Рис. 23 Блок-схема приемника

## Асинхронный передатчик

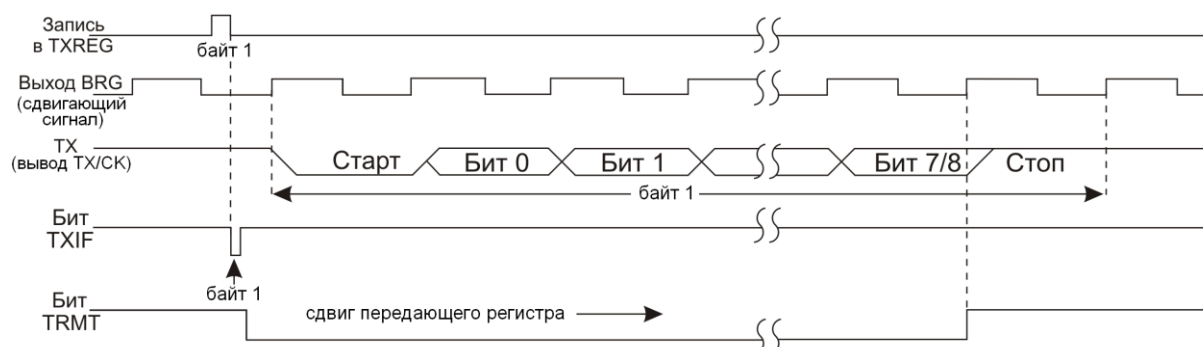
Блок-схема передатчика представлена на Рис. 22. Основой передатчика является сдвиговый регистр передачи (TSR). Он получает информацию из буфера передатчика (TXREG). В буфер TXREG данные загружаются программно. Сдвиговый регистр TSR загружается новыми данными из TXREG (если они есть) как только передастся 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. 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, когда устанавливается бит TXEN (TXSTA<5>). Фактически передача не начнется: пока данные не будут загружены в TXREG и генератор скорости передачи не выдаст сдвиговый синхроимпульс (см. Рис. 24). 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. 25).

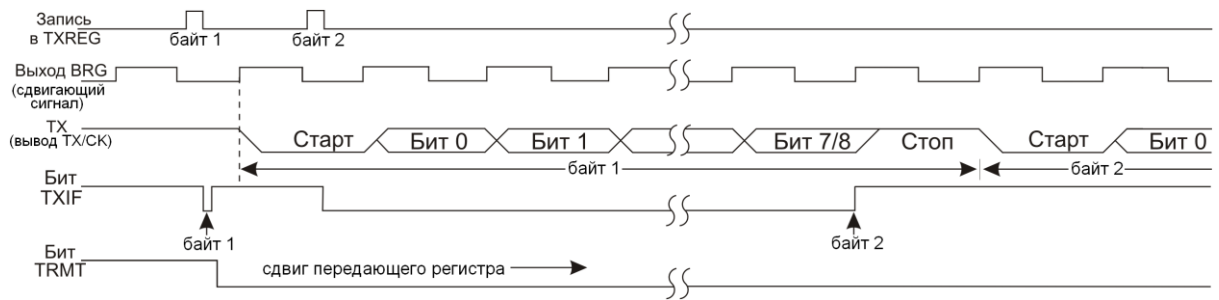
Для выбора 9-ти битной передачи, бит TX9 (TXSTA<6>) должен быть установлен в единицу. Девятое значение бита записывается в TX9D (TXSTA<0>) перед записью 8-ми битных данных в TXREG, так как запись данных в TXREG может вызвать немедленную передачу данных в TSR (если TSR пуст).

Шаги для настройки асинхронной передачи следующие:

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



**Рис. 24** Асинхронная передача



**Рис. 25** Асинхронная неразрывная последовательная передача

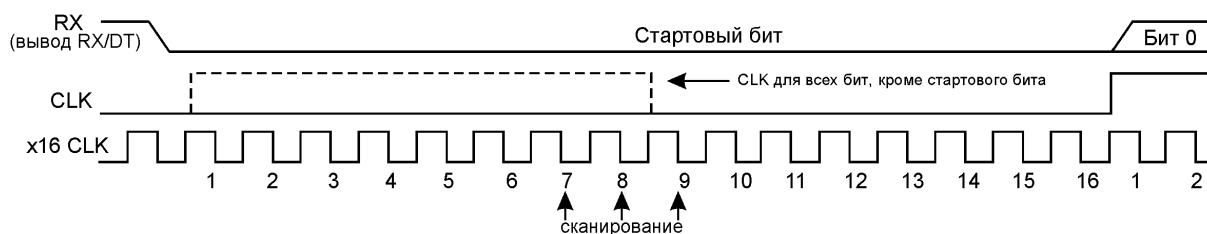
## Асинхронный приемник

Блок-схема приемника показана на Рис. 23. Данные с вывода PA2/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.

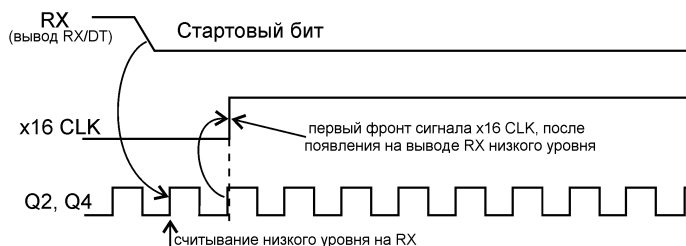
Если выбран асинхронный режим, то приемник включается установкой бита CREN (RCSTA<4>).

Основой приемника является сдвиговый регистр приема (RSR). После обнаружения 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 PA2/RX/DT сканируются, три раза в каждом такте приема, мажоритарной схемой обнаружения, для выявления уровня сигнала на выводе PA2/RX/DT. Сканирование осуществляется на седьмом, восьмом и девятом заднем фронте (спаде) импульсов частотой в 16 раз превышающей частоту приема (Рис. 26).



**Рис. 26** Схема сканирования вывода PA2/RX/DT



**Рис. 27** Обнаружение стартового бита

Шаги для настройки асинхронного приема следующие:

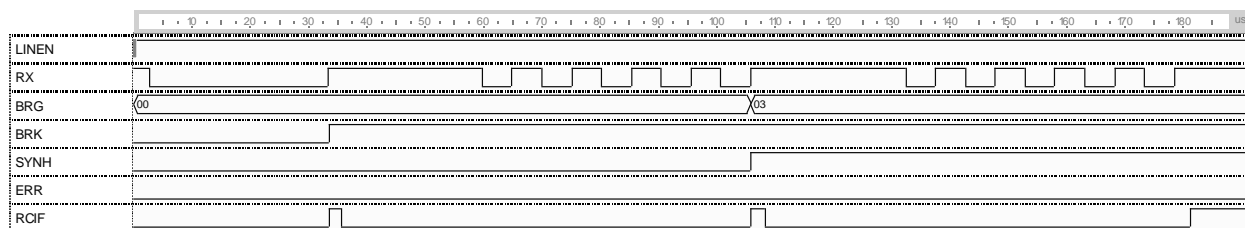
- записать значение в регистр SPBRG для задания скорости приема.
- включить асинхронный последовательный порт (бит SYNC=0 и бит SPEN=1).
- если требуются прерывания, тогда установите бит RCIE.
- если требуется 9-ти битный прием, тогда установите бит RX9.
- разрешите прием установкой бита CREN.
- при завершении приема установится бит RCIF, и произойдет прерывание (если установлен бит RCIE).
- считайте RCSTA для получения значения девятого бита (для 9-ти битного приема) и бит FERR для определения ошибки.
- считайте 8-ми битные принятые данные из регистра RCREG.
- 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.

## Режим автоматического приема LIN заголовка

При работе в асинхронном режиме модуль USART может автоматически принимать и распознавать заголовок LIN фрейма. Для этого необходимо разрешить работу блока LIN Frame Detector установкой бита LINEN. Прием поля BREAK производится на основании скорости передачи задаваемой в регистре SPBRG. При приеме поля BREAK вырабатывается прерывание RCIF и выставляется флаг BRK. В регистре LINCNTR биты BRKCNT[3:0] содержат длительность принятого поля BREAK. Если длительность поля BREAK меньше 11, автоматически вырабатывается флаг возникновения ошибки ERR. Максимальное значение BRKCNT – 16, независимо от реальной длительности поля BREAK. Для сброса сигнала прерывание RCIF после приема поля BREAK в бит BRK регистр LINCNTR необходимо записать «0». При этом прерывание снимается, но значение бита не изменяется.

После приема поля BREAK автоматически принимается поле SYNCH. При приеме поля SYNCH автоматически вычисляется скорость передачи. При приеме поля SYNCH вырабатывается прерывание RCIF и выставляется флаг SYNCH. В регистре LINBRG содержится скорость передачи поля SYNCH. В случае необходимости эта скорость может быть использована для задания скорости передачи SPBRG для всего блока USART.



**Рис. 28** Заголовок LIN фрейма

Для сброса сигнала прерывание RCIF после приема поля SYNCH в бит SYNCH регистр LINCNTR необходимо записать «0». При этом прерывание снимается, но значение бита не изменяется. В случае если при приеме поля SYNCH распознаваемая скорость будет ниже, чем скорость при SPBRG == 0xFF, автоматически вырабатывается флаг возникновения ошибки ERR. После приема поля SYNCH модуль USART переходит в обычный режим работы в асинхронном режиме, все последующие принимаемые данные отображаются в регистре RCREG. После приема всего LIN фрейма для подготовки к приему следующего необходимо сбросить блок LIN Frame Detector. Для этого необходимо установить бит LINEN сначала в «0», а затем в «1».

## **Передача LIN фрейма**

Формирование LIN фрейма осуществляется программным путем. Для передачи поля BREAK в регистр SPBRG должна быть записана скорость меньшая чем реальна скорость передачи, таким образом, что бы при отправке байта 0x00 приемник воспринял минимум 11 битов равных 0 (с учетом старт бита). После передачи поля BREAK в регистре SPBRG задается нормальная скорость работы и для отправки поля SYNCH отправляется байт 0x55. Затем остальные байты фрейма.

## 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: PA3/СК - линия тактовых импульсов и PA2/DT - линия данных. Ведущий режим означает, что процессор формирует тактовые импульсы на линии PA3/СК. Ведущий режим выбирается установкой бита CSRC (TXSTA<7>).

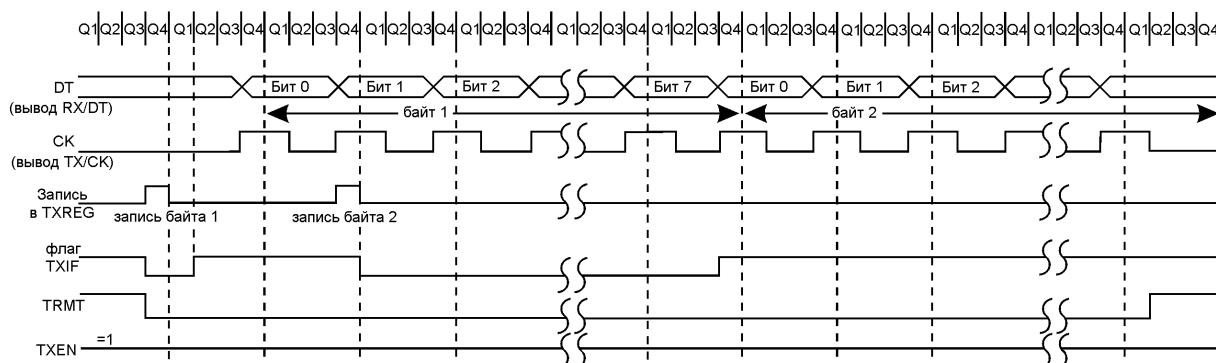


Рис. 29 Синхронная передача в ведущем режиме



## **Передача данных в синхронном ведущем режиме**

Блок-схема передатчика показана на Рис. 22. 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 PA3/TX/CK. Данные стабилизируются по заднему фронту тактовых импульсов (см. Рис. 29). 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 и переведет выводы PA2/RX/DT и PA3/TX/CK в третье состояние. Если во время передачи будут установлены биты CREN и SREN, передача будет отменена, и вывод PA2/RX/DT вернется в третье состояние (для приема). Вывод PA3/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 PA2/RX/DT сразу же будет переключена из третьего состояния 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:

- записать значение в регистр SPBRG для задания скорости передачи.
- включить синхронный последовательный порт в ведущем режиме (биты SYNC=1, SPEN=1 и CSRC=1).
- проверить что биты CREN и SREN сброшены, если эти биты установлены, то они отменяют передачу.
- если требуются прерывания, тогда установите бит TXIE.

- если требуется 9-ти битная передача, тогда установите бит TX9.
- если выбрана 9-ти битная передача, девятый бит должен быть загружен в TX9D.
- загрузите данные в регистр TXREG.
- 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.

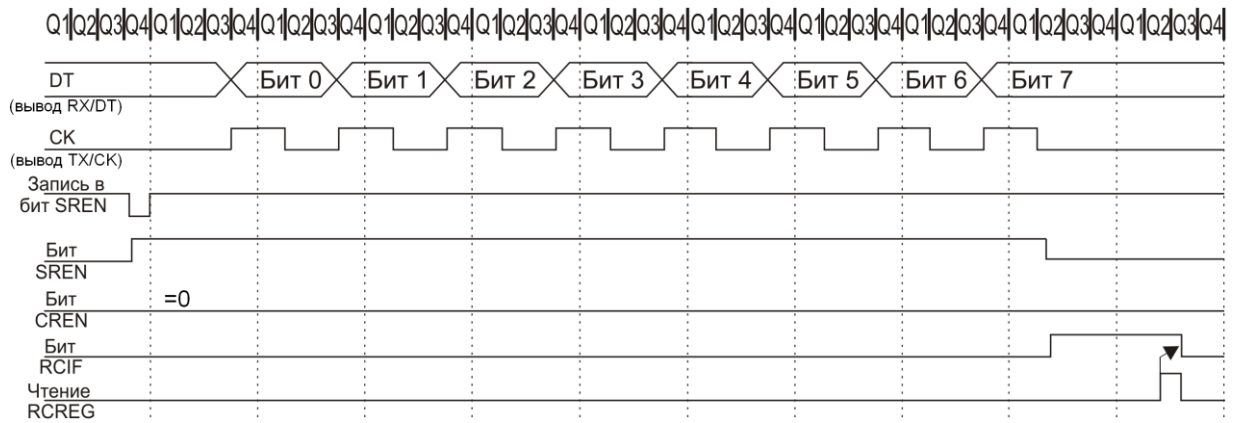
## **Прием данных в синхронном ведущем режиме**

Если выбран синхронный режим, прием разрешается установкой бита SREN (RCSTA<5>) или бита CREN (RCSTA<4>). Данные с вывода PA2/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 прежнее значение RX9D.

Шаги для настройки приема в синхронном ведущем режиме:

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



**Рис. 30 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 исполняется 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 микроконтроллер из режима SLEEP, и если разрешено периферийное прерывание, тогда программа переходит к вектору прерывания (0020h).

Шаги для настройки передачи в синхронном ведомом режиме:  
записать значение в регистр SPBRG для задания скорости передачи.

- включить синхронный последовательный порт в ведомом режиме (биты SYNC=1, SPEN=1 и CSRC=0).
- сбросить бит CREN.
- если требуются прерывания, тогда установите бит TXIE.
- если требуется 9-ти битная передача, тогда установите бит TX9.
- если выбрана 9-ти битная передача, девятый бит должен быть загружен в TX9D.
- загрузите данные в регистр TXREG.
- 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:

- записать значение в регистр SPBRG для задания скорости приема;
- включить синхронный последовательный порт в ведомом режиме (биты SYNC=1, SPEN=1 и CSRC=0);
- если требуются прерывания, тогда установите бит RCIE;
- если требуется 9-ти битный прием, тогда установите бит RX9;
- для разрешения приема установите бит CREN;
- при завершении приема установится бит RCIF, и произойдет прерывание (если установлен бит RCIE);

- считайте RCSTA для получения значения девятого бита (для 9-ти битного приема) и бит определения ошибки;
- считайте 8-ми битные принятые данные из регистра RCREG;
- 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.

## **Блок внутренней памяти данных EEPROM**

Блок контроллера EEPROM памяти предназначен для возможности работы ядра микроконтроллера с встроенной памятью типа EEPROM размером 256 8-ми битных слов. Память EEPROM имеет организацию 16 строк по 16 8-ми битных слов. Минимально доступной для чтения и запись является одно 8-ми битное слово. Минимально доступной для стирания является одна строка. Перед записью необходимо провести стирание строк, в которых будет расположена записываемая информация. Запись возможна только "1" поверх "0". В очищенном состоянии память содержит все "0".

При переходе в SLEEP режим блок контроллера должен быть программным образом выключен для обеспечения энергосберегающего режима. После выхода из SLEEP режима блок должен включаться заново.

**Основные выполняемые функции и возможности**

**Таблица 42**

<b>Наименование</b>	<b>Краткое описание, качественно-количественные характеристики</b>
Запись 8-ми битного слова в EEPROM	Позволяет записать в заданный байт, значение битов отличное от нуля. Запись в TEST режиме возможна всегда
Чтение 8-ми битного слова из EEPROM	Позволяет считать из памяти байт
Очистка всей EEPROM	Позволяет стереть содержимое всей памяти
Очистка строки EEPROM	Позволяет очистить заданную строку памяти.
Запись всей EEPROM одним значением	Позволяет проинициализировать всю память одним значением



## **Регистр режима работы контроллера**

Работа с памятью EEPROM осуществляется по средствам чтения и записи регистров контроллера. Описание регистров приведено ниже.

**Регистр контроля и тестирования**

**Таблица 43**

EE\_CONT.ADR = 0x14, Банк доступа BANK = 0x05

Номер	7	6	5	4	3	2	1	0
Доступ*	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса**	1	0	0	0	0	0	0	0
	TEST_P	EETEST	CPTEST	VEE2	VEE1	BRG2	BRG1	BRG0

\*

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

**Таблица 44**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7	TEST_P	Сигналы для отбраковочных испытаний микросхемы.
6	EETEST	
5	CPTEST	
4	VEE2	
3	VEE1	
2..0	BRG[2:0]	<p>Режим определения временных характеристик:                      В зависимости от частоты работы ядра, необходимо указать контроллеру эту частоту, исходя из которой, контроллер будет формировать необходимые длительности сигналов Программирования и Стирания заданных длительностей.</p> <p>000, 001, 010 – зарезервировано                      011 – частота Fc = 1 – 10 МГц                      100 – частота Fc = 500 кГц – 1 МГц                      101 – частота Fc = 250 кГц – 500 кГц                      110 – частота Fc = 0 – 250 кГц                      111 – частота Fc = 40 – 60 кГц</p>

**Регистр режима работы**

**Таблица 45**

EE\_MODE.ADR = 0x15, Банк доступа BANK = 0x05

Номер	7	6	5	4	3	2	1	0
Доступ*	R/W	RO	R/W	RO	RO	R/W	R/W	R/W
Значение после сброса**	0	0	0	0	0	0	0	0
	EN_EE	-	IEBUSY	BUSY	-	MODE2	MODE1	MODE0

\*

R/W - бит доступен на чтение и запись;

RO - бит доступен только на чтение;

U - бит физически не реализован или зарезервирован.

**Таблица 46**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7	EN_EE	Разрешение работы контроллера EEPROM 0 – Контроллер выключен 1 – Контроллер включен
6	-	Зарезервировано
5	IEBUSY	Разрешения прерывания по окончании занятости контроллера EEPROM
4	BUSY	Флаг занятости контроллера EEPROM
3	-	Зарезервировано
2..0	MODE[2:0]	Режим работы контроллера: 000 – Нет работы 001 – Чтение слова из EEPROM 010 – Запись слова в EEPROM 011 – Отчистка строки EEPROM 100 – Отчистка всей EEPROM 101 – Запись всей EEPROM одним значением 110 – Запрещено 111 – Запрещено После каждой операции, блок должен быть переведен в режим «Нет работы»

**Регистр данных**

**Таблица 47**

EE\_DATA.ADR = 0x16, Банк доступа BANK = 0x05

Номер	7	6	5	4	3	2	1	0
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса**	0	0	0	0	0	0	0	0
	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

\*

R/W - бит доступен на чтение и запись;

RO - бит доступен только на чтение;

U - бит физически не реализован или зарезервирован.

**Таблица 48**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7..0	DATA[7:0]	Регистр данных: При выдаче команды в этот регистр записывается команда, при выдаче адреса в регистр записывается адрес, при выдаче или приеме данные

**Регистр адреса обращения**

**Таблица 49**

EE\_ADR.ADR = 0x17, Банк доступа BANK = 0x05

Номер	7	6	5	4	3	2	1	0
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса**	0	0	0	0	0	0	0	0
	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0

\*

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

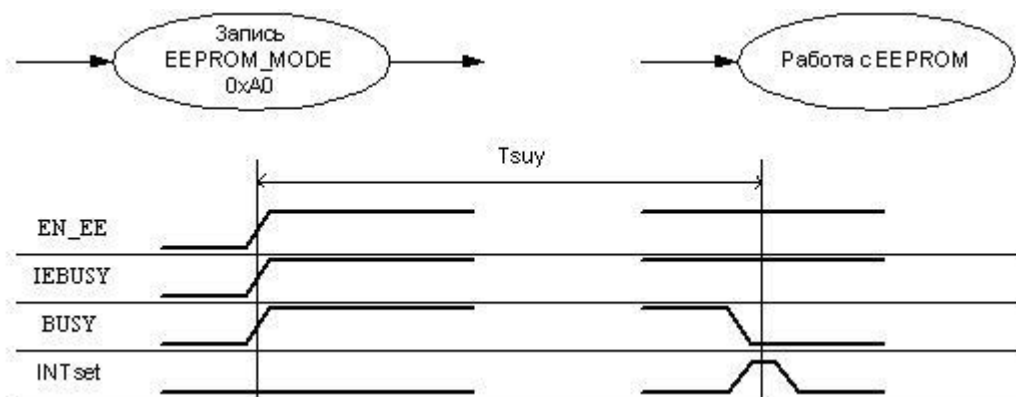
**Таблица 50**

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7..0	ADR[7:0]	Регистр адреса данных: При чтении и записи одного слова используются все разряды адреса, при очистке строки имеют значения только ADR[7:4], при блочной очистке и записи (в TEST режиме) содержимое регистра значения не имеет. При выполнении операций установления флагов защиты от записи или стирания, номер строки для которой будет устанавливаться флаг определяется данным регистром.

## Работа блока по стиранию, записи и чтению данных

### Включение EEPROM

После выставления бита EN\_EE в регистре EE\_MODE производится включение памяти EEPROM. Процесс включения при тактовой частоте микроконтроллера 10 МГц занимает порядка 150 мкс ( $T_{suy}$ ). Последовательность действий при включении EEPROM представлена на рисунке 31.

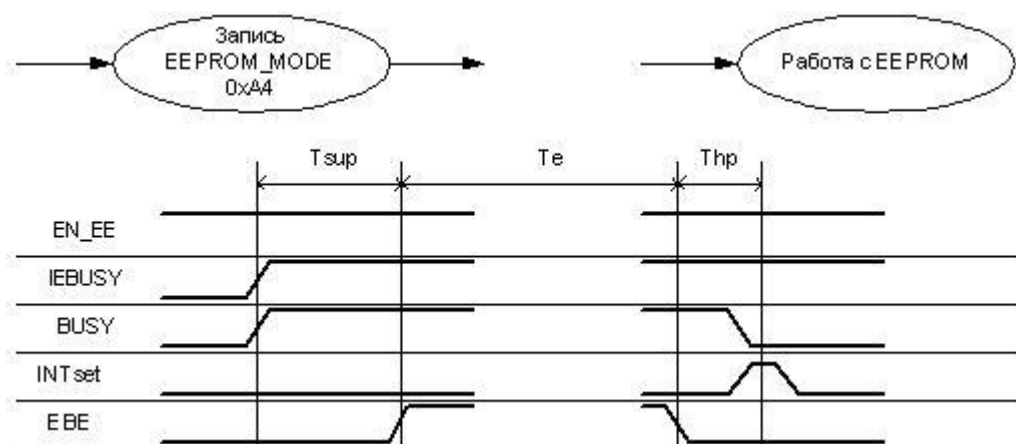


**Рис. 31** Последовательность действий при включении EEPROM

### Очистка всей EEPROM

Для очистки всего содержимого EEPROM используется команда очистки всей EEPROM. Для очистки всей памяти необходимо что бы был включен контроллер EEPROM. Процесс очистки всей EEPROM при тактовой частоте микроконтроллера 10 МГц занимает порядка 5 мс ( $T_{sup}$ ,  $T_e$  и  $T_{hp}$  )

Последовательность действий при очистке всей EEPROM представлена на рисунке 32.



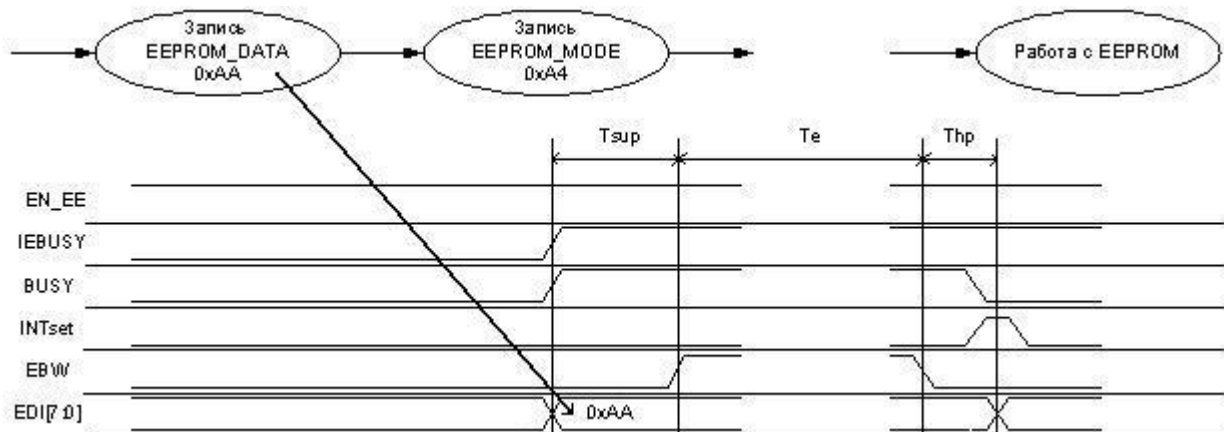
**Рис. 32** Последовательность действий при очистке всей EEPROM

### Запись всей EEPROM одним значением

Для заполнения всего содержимого EEPROM используется команда записи всей EEPROM. Перед подачей этой команды в регистр EE\_DATA необходимо записать значение которым будет заполнена вся память. Для заполнения всей

памяти необходимо что бы был включен контроллер EEPROM. Процесс заполнения всей EEPROM при тактовой частоте микроконтроллера 10 МГц занимает порядка 5 мс ( $T_{sup}$ ,  $T_e$  и  $T_{hp}$ ).

Последовательность действий при заполнении всей EEPROM представлена на рисунке 33.

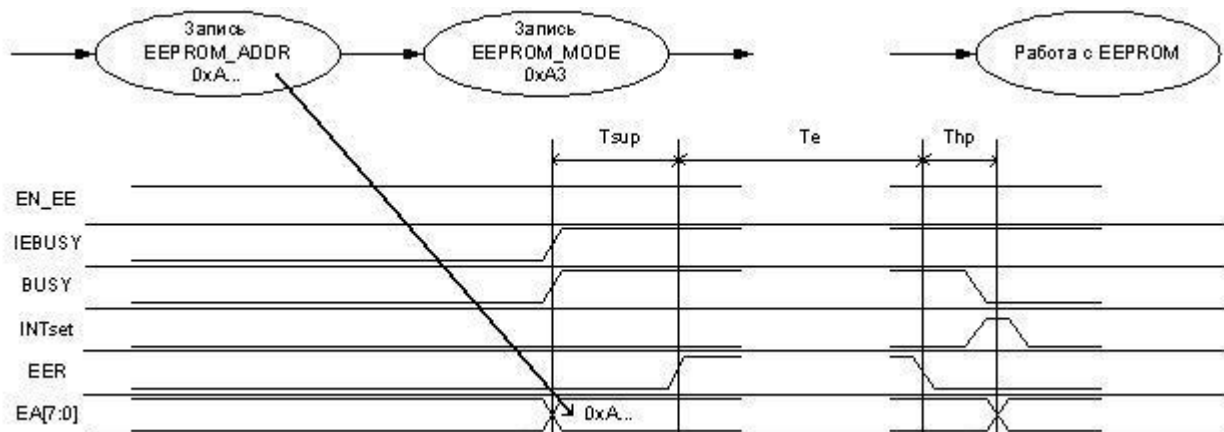


**Рис. 33** Последовательность действий при заполнении всей EEPROM

### **Очистка строки EEPROM**

Для очистки содержимого одной строки EEPROM используется команда очистки одной строки EEPROM. Перед подачей этой команды в регистр EE\_ADR[7:4] необходимо записать адрес строки для очистки (младшие разряды регистра EE\_ADR значения не имеют). Для очистки заданной строки памяти необходимо чтобы контроллер EEPROM был включен. Процесс очистки одной строки EEPROM при тактовой частоте микроконтроллера 10 МГц занимает порядка 5 мс ( $T_{sup}$ ,  $T_e$  и  $T_{hp}$ ).

Последовательность действий при стирании одной строки EEPROM представлена на рисунке 34.



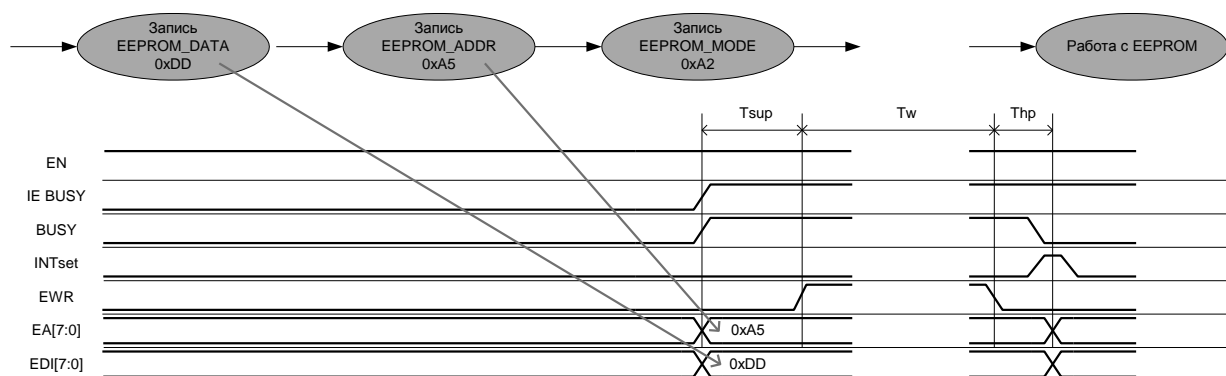
**Рис. 34** Последовательность действий при стирании одной строки EEPROM



## Запись слова в EEPROM

Для записи одного 8-ми битного слова в память EEPROM используется команда запись слова в EEPROM. Перед подачей этой команды в регистр EE\_ADR[7:0] необходимо записать адрес для записи, а в регистр EE\_DATA записать значение для сохранения в памяти. Для записи слова в память необходимо, чтобы контроллер EEPROM был включен. Процесс записи строки EEPROM при тактовой частоте микроконтроллера 10 МГц занимает порядка 5 мс ( $T_{sup}$ ,  $T_e$  и  $T_{hp}$ )

Последовательность действий при записи одного слова в EEPROM представлена на рисунке 35.

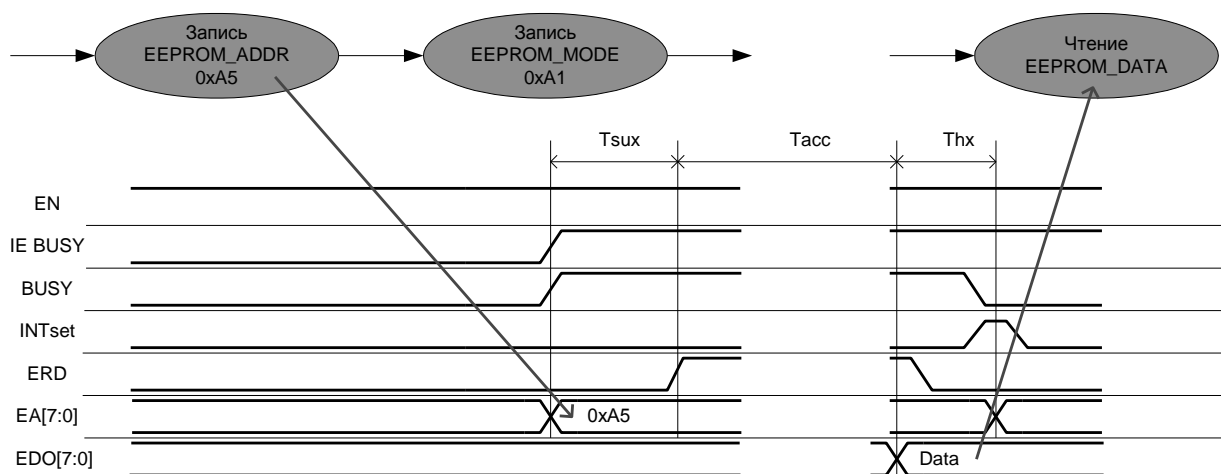


**Рис. 35** Последовательность действий при записи одного слова в EEPROM

## Чтение слова из EEPROM

Для чтения одного 8-ми битного слова из памяти EEPROM используется команда чтение слова в EEPROM. Перед подачей этой команды в регистр EE\_ADR[7:0] необходимо записать адрес для чтения. Для чтения слова из памяти необходимо, чтобы контроллер EEPROM был включен. Процесс чтения строки EEPROM при тактовой частоте микроконтроллера 10 МГц занимает порядка 1 мкс ( $T_{sux}$ ,  $T_{acc}$  и  $T_{hx}$ ).

Последовательность действий при чтении одного слова из EEPROM представлена на рисунке:



**Рис. 36**

При переходе в SLEEP режим блок контроллера должен быть программным образом выключен для обеспечения энергосберегающего режима. После выхода из SLEEP режима блок должен включаться заново.

## **Описание блока управления EEPROM – памятью программ**

Блок позволяет осуществлять чтение, запись и стирание памяти различными способами:

- с помощью команд микроконтроллера TABLWD, TABLRD (в этом случае используются сигналы чтения и записи от ядра).
- с помощью установки/сброса битов периферийных регистров блока управления EEPROM памяти программ.

**Внимание!** Из пользовательской программы, в случае работы через периферийные регистры блока управления EEPROM, допускается осуществлять **только** операцию стирания всей EEPROM памяти. Операции чтения/записи необходимо осуществлять с помощью команд чтения/записи таблиц (TABLWD, TABLRD). Обратите внимание, что при стирании EEPROM памяти удаляются и конфигурационные биты. После стирания их необходимо восстановить до момента выключения напряжения питания или подачи сигнала сброс.

Адресное пространство 2 Кслов или от 16`h0000 до 16`h07FF.

При работе через установку/сброс битов управления периферийных регистров возможно использование функции аппаратной поддержки. Функция реализует автоматическую выработку длительностей сигналов записи / стирания, а также автоматический сброс бит записи/стирания периферийных регистров. По умолчанию включен режим аппаратной поддержки.

В блоке реализованы следующие защитные функции:

- при доступе по чтению данных по адресу, выходящему за пределы адресного пространства (0000 – 07FF), получаемые данные будут равны 0000. Исключением являются адреса FE00 – FE0F. При обращении по чтению к этим адресам данные будут содержать текущее конфигурационное слово (см. подраздел «Регистры конфигурации микроконтроллера»);
- все попытки записи по адресам, выходящим за пределы адресного пространства блока, будут заблокированы. Исключения: попытка записи по адресам 16`h07F0 -- 16`h07FF блокируется, так как это область хранения конфигурационных бит. Запись конфигурационных бит осуществляется по адресам FE00 – FE0F;
- изменение содержимого регистра хранения конфигурации в рабочем режиме не возможно, т.е. доступ по записи заблокирован;
- реализован контроль правильности адресации при выполнении операций записи слова и стирания строки, в случае если режим аппаратной поддержки включен.

**Описание регистров**

**Таблица 51**

Общее описание регистров блока EEPROM\_interface

Обозначение	Адрес	Доступ	Значение после сброса
EEDIV	13h, банк 6	RW RW RW RW RW RW RW RW	0000 0000
EECON	11h, банк 14	R- R- RW RW RW RW RW RW	0000 0000
EDLSB	12h, банк 15	R- R- R- R- R- R- R- R-	0000 0000
EDMSB	13h, банк 15	R- R- R- R- R- R- R- R-	0000 0000
EEMOD	14h, банк 15	R- RW RW RW RW RW RW RW	0000 0000
EAMSB	15h, банк 15	R- U- U- U- RW RW RW RW	0--- 0000
EALSB	16h, банк 15	RW RW RW RW RW RW RW RW	0000 0000
CFREG	17h, банк 15	U- RW RW RW RW RW RW RW	-111 1111

**Регистр коэффициента деления частоты генератора**

**Таблица 52**

Регистр EEDIV (адрес: 17h, банк 7)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EEDIV	EEDIV	EEDIV	EEDIV	EEDIV	EEDIV	EEDIV	EEDIV
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7-0</b>		<p>Коэффициента деления частоты генератора с целью выработки импульсов записи и стирания длительностью <math>\geq 5</math> мс. Делитель состоит из прескалера и постскалера. Значение к-та деления прескалера фиксировано и равно 651. Значение к-та постскалера равно значению регистра. Таким образом при измени значения регистра на 1, реальный к-нт деления изменится на 651. Правило выбора коэффициента деления <math>K = 5 \text{ мс} / 651 \cdot T_{\text{osc}}</math>. В случае дробного результата, округление проводить в большую сторону, для получения времени импульса больше 5 мс.</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 неизвестно.

**Регистр управления EEPROM памяти**

**Таблица 53**

Регистр EECON (адрес: 11h, банк 14)

R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LOCK	ERROR	EET	EBW	EBE	EER	EWR	ERD
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7</b>		Флаг режима EEPROM 1 – режим standby 0 – рабочий режим					
<b>бит 6</b>		Флаг ошибки данных EEPROM 1 – указывает на наличие ошибки 0 – указывает на отсутствие ошибки					
<b>бит 5</b>		Установка тестового режима работы EEPROM. Не доступен для записи в рабочем режиме микроконтроллера 1 – тестовый режим 0 – рабочий режим					
<b>бит 4</b>		Одновременная запись данными всей EEPROM. Описание работы с этим разрядом приведено в подразделе «Выполнение операций записи/стирания памяти».					
<b>бит 3</b>		Одновременное стирание всей EEPROM. Описание работы с этим разрядом приведено в подразделе «Выполнение операций записи/стирания памяти».					
<b>бит 2</b>		Стирание строки EEPROM. Адрес строки определяется 10-4 разрядами адресной шины. Таким образом, строка содержит 16 слов. Описание работы с этим разрядом приведено в подразделе «Выполнение операций записи/стирания памяти».					
<b>бит 1</b>		Запись одного слова данных в EEPROM. Адрес слова определяется 10-0 разрядами адресной шины. Описание работы с этим разрядом приведено в подразделе «Выполнение операций записи/стирания памяти».					
<b>бит 0</b>		Чтение слова данных EEPROM. Адрес слова определяется 10-0 разрядами адресной шины. Описание работы с этим разрядом приведено в подразделе «Выполнение операции чтения памяти».					

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 режима работы EEPROM памяти**

**Таблица 54**

Регистр EEMOD (адрес: 14h, банк 15)

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STATE	RESET	CPEN	HWS	AM	TM2	TM1	ESTBY
бит 7	6	5	4	3	2	1	бит 0

<b>бит 7 STATE</b>	Флаг состояния автомата аппаратной поддержки 1 – автомат находится в состоянии работы 0 – автомат находится в состоянии «выключен»
<b>бит 6 RESET</b>	Сброс автомата аппаратной поддержки и делителя. 1 – схемы находятся в состоянии сброса
<b>бит 5 CPEN</b>	Разрешение работы блока Charge Pumpe. 1 – блок включен. Важно: установка высокого напряжения необходимого для операций записи и стирания происходит через 10 - 20 мкс после установки этого бита в 1. Признаком появления высокого напряжения является EAMSB[7] = 1. 0 – блок выключен
<b>бит 4 HWS</b>	Отключение режима аппаратной поддержки операций записи/стирания памяти. Описание работы с этим разрядом приведено в подразделе «Выполнение операций записи/стирания памяти». 1 – режим отключен 0 – режим включен (значение по умолчанию)
<b>бит 3 AM</b>	Режим адресации EEPROM памяти. Описание работы с этим разрядом приведено в подразделе «Режимы адресации памяти». 1 – адрес поступает с регистров EAMSB[3:0] и EALSB[7:0] 0 – адрес поступает с с регистров TBLPTRH и TBLPTRL
<b>бит 2 TM2</b>	Выбор тестового подрежима обеих половин EEPROM памяти. Состояние разряда имеет значение, только если EECON[5] равен 1. Не доступен для записи в рабочем режиме микроконтроллера. 1 – выбран тестовый подрежим «High». 0 – выбран тестовый подрежим «Low».
<b>бит 1 TM1</b>	Выбор тестового режима обеих половин EEPROM памяти. Состояние разряда имеет значение, только если EECON[5] равен 1. Не доступен для записи в рабочем режиме микроконтроллера. 1 – выбран тестовый режим «Internal» 0 – выбран тестовый режим «External».
<b>бит 0 ESTBY</b>	Отключение (перевод в режим standby) EEPROM памяти. Не доступен для записи в рабочем режиме микроконтроллера. 1 – EEPROM отключена. Признаком этого состояния служат EECON[7] равный 1. 0 – EEPROM находится в рабочем состоянии. Значение по умолчанию.

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 старшего адреса EEPROM памяти**

**Таблица 55**

Регистр EAMSB (адрес: 15h, банк 15)

R-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
CPRDY	-	-	-	-	EA[10]	EA[9]	EA[8]
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7 CPRDY</b>		Флаг состояния блока Charge Pump 1 – блок включен и на выходе установлено высокое напряжение. Память готова к операции запись или стирание 0 – блок выключен или на выходе еще нет высокого напряжения					
<b>биты 6 - 3</b>		Не реализованы.					
<b>бит 2-0 EA[10:8]</b>		Разряды 10 – 8 шины адреса EEPROM памяти. Адресуют слова в пределах одной из половин EEPROM памяти. Состояние имеет значение только при EEMOD[3] = 1. Описание работы с этим разрядом приведено в подразделе «Режимы адресации памяти».					

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 младшего адреса EEPROM памяти**

**Таблица 56**

Регистр EALSB (адрес: 16h, банк 15)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EA[7]	EA[6]	EA[5]	EA[4]	EA[3]	EA[2]	EA[1]	EA[0]
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7-0 EA[7:0]</b>		Разряды 7 – 0 шины адреса EEPROM памяти. Адресуют слова в пределах одной из половин EEPROM памяти. Состояние имеет значение только при EEMOD[3] = 1. Описание работы с этим разрядом приведено в подразделе «Режимы адресации памяти».					

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 конфигурационных бит**

**Таблица 57**

Регистр CFREG (адрес: 17h, банк 15)

U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
-	DBG_EN	BODEN	PM0	WDT1	WDT0	FOSC1	FOSC0
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7</b>		Не реализован					
<b>бит 6 DBG_EN</b>		Конфигурационный бит разрешения режима отладки.					

<b>бит 5 BODEN</b>	Конфигурационный бит разрешения работы BOR.
<b>бит 4 PM0</b>	Конфигурационный бит установки защиты памяти программ.
<b>бит 3 WDT1</b>	Конфигурационный бит режима WDT.
<b>бит 2 WDT0</b>	Конфигурационный бит режима WDT.
<b>бит 1 FOSC1</b>	Конфигурационный бит режима генератора.
<b>бит 0 FOSC0</b>	Конфигурационный бит режима генератора.

Примечания:

- 1 Регистр не доступен для записи в рабочем режиме микроконтроллера.
- 2 Подробнее описание конфигурационных бит приведено в подразделе «Чтение/запись конфигурационных бит».

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 младшего байта данных EEPROM памяти**

**Таблица 58**

Регистр EDLSB (адрес: 12h, банк 15)

R -0	R -0	R -0	R -0	R -0	R-0	R-0	R-0
ED[7]	ED[6]	ED[5]	ED[4]	ED[3]	ED[2]	ED[1]	ED[0]
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7-0 EA[7:0]</b>		Разряды 7 – 0 шины данных EEPROM памяти (чтение). Описание работы с этим разрядом приведено в подразделе «Выполнение операции чтения памяти».					

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 старшего байта данных EEPROM памяти**

**Таблица 59**

Регистр EDMSB (адрес: 13h, банк 15)

R -0	R -0	R -0	R -0	R -0	R-0	R-0	R-0
ED[15]	ED[14]	ED[13]	ED[12]	ED[11]	ED[10]	ED[9]	ED[8]
бит 7	6	5	4	3	2	1	бит 0
<b>бит 7-0 ED[15:8]</b>		Разряды 15 – 8 шины данных EEPROM памяти (чтение). Описание работы с этим разрядом приведено в подразделе «Выполнение операции чтения памяти».					

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.



## **Выполнение операций с блоком EEPROM – памятью программ**

### **Выполнение операций записи/стирания памяти**

Выполнение операций записи/стирания возможно в двух режимах: с аппаратной поддержкой и без. Выбор режима задается битом EEMOD[4].

1 В случае, если **включен режим аппаратной поддержки**, т.е. EEMOD[4]=0, то для запуска требуемой операции необходимо установить соответствующий бит регистра EECONH/EECONL в 1. При этом включение и отключение блока Charge Pumpe, отсчет длительности импульса (обязательна предварительная инициализация регистра EE\_DIV), а также сброс бита в 0 будут произведены автоматически. Признаком окончания операции могут служить:

- равенство данного бита 0;
- бит EEMOD[7] = 0.

При выполнении операции будет использоваться адрес согласно выбранному режиму адресации памяти (см. ниже), а для операции записи данные из регистров «табличной защелки».

Примечание – регистры «табличной защелки» TBLATH и TBLATL загружаются с помощью команд TLWT.

Важно! В данном режиме реализована функция защиты от неверной адресации. Если адресована, например, младшая половина памяти, и в то же время пользователь устанавливает бит управления записью или стирания старшей половины, то данное действие будет заблокировано.

2 В случае, если **режим аппаратной поддержки выключен**, т.е. EEMOD[4]=1, то перед установкой соответствующего требуемой операции бита в 1 необходимо:

- согласно выбранному режиму адресации памяти загрузить адрес, а для операции записи загрузить в регистры «табличной защелки» требуемые данные;
- включить Charge Pumpe (EEMOD[5]=1);
- убедиться в наличии высокого напряжения на выходе Charge Pumpe (EAMSB[7] должен быть в состоянии 1).

Далее необходимо отсчитать временной интервал равный 5 мс и сбросить соответствующий требуемой операции бит в 0. Затем отключить Charge Pumpe (EEMOD[5]=0) если не требуется повторных операций записи или стирания.

### **Режимы адресации памяти**

Режим адресации памяти выбирается битом EEMOD[3].

1 Если, **бит EEMOD[3] сброшен в 0** (это значение по умолчанию), то в качестве адреса используется содержимое регистров табличного указателя памяти программ, т.е. TBLPTRH и TBLPTRL.

Физическое адресное пространство – 12 бит или 4096 слов (11-0). Старший бит фактически выбирает старшую или младшую половины памяти.

Адреса от 0FF0 до 0FFF заняты конфигурационными битами. При прямой адресации к ним возможно обращение только по чтению. Запись в эти адреса блокируется.

Область адресов FE00 – FE0F – виртуальное пространство конфигурационных бит. При обращении по записи в любой из указанных адресов, будет записано FFFF, не зависимо от состояния шины данных.

При попытке чтения адресов выходящих за пределы адресного пространства блока (кроме FE00 – FE0F) выходная шина данных примет состояние 0000.

Запись в адреса, выходящие за пределы адресного пространства блока (кроме FE00 – FE0F), будет блокироваться.

2 Если бит **EEMOD[3]** установлен в 1, то адресация происходит аналогично предыдущему варианту, но в качестве адреса используется содержимое регистров EAMSB и EALSB. Этот режим рекомендуется для выполнения операций стирания.

### ***Выполнение операции чтения памяти***

Для чтения содержимого старшей или младшей половин памяти при доступе через регистры, достаточно установить сигнал EECONH[0] или EECONL[0] соответственно в 1 на один системный цикл (после чего сбросить в 0). Содержимое адреса памяти будет находиться в регистрах EDMSB и EDLSB. При выполнении данной операции необходимо следить за соответствием адреса и использованием бит регистров EECONH и EECONL.

### ***Чтение/запись конфигурационных бит***

Конфигурационные биты физически находятся по адресам 0FF0 – 0FFF. Кроме того, существует виртуальная область памяти конфигурационных бит (FE00 – FE0F).

При чтении конфигурационных битов по любому адресу в диапазоне FE00h–FE0Fh, блок будет возвращать конфигурационное слово, как оно представлено в таблице 61.

При чтении конфигурационных битов по любому адресу в диапазоне 0FF0h–0FFFh, блок будет возвращать фактическое содержимое ячейки памяти по соответствующему адресу.

Как было сказано ранее, значения битов конфигурации дублируются в памяти программ по адресам 0FF0h–0FFFh. Значению бита =0 соответствует значение ячейки ПЗУ равное FFFFh, а значению бита =1 – значение ячейки равное 0000h. В таблице 61 приведено соответствие адресов ячеек памяти битам конфигурации.

Область памяти программ 0FF0 – 0FFF доступна только для чтения. Для записи конфигурационных бит необходимо обращаться к адресам FE00 – FE0F. Для оперативной работы с конфигурационными битами (чтение и запись) в области периферийных регистров реализован регистр CFREG (см. описание регистров). При работе в рабочем режиме микроконтроллера доступ по записи к нему заблокирован.

## **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 61). Чтение битов DBG\_EN, BODEN, PM0, WDTPS1, WDTPS0, FOSC1 и FOSC0 возможно по любому адресу в диапазоне FE00h-FE07h.

Значения битов конфигурации дублируются в ПЗУ программ по адресам 07F0h-07FFh. 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 07F0h-07FFh доступна только для чтения.

Примечание – Если в ячейки с адресами 07F4h 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, за исключением команды «СТЕРЕТЬ ВСЮ ПАМЯТЬ» из расширенного набора команд.

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.

### **Внимание!**

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

**Таблица 61**

Регистры конфигурации микроконтроллера

		FE06h	FE05h	FE04h	FE03h	FE02h	FE01h
-	-	DBG_EN	BODEN	PM0	WDTPS1	WDTPS0	FOSC1
бит 15 - 8	бит 7	6	5	4	3	2	1

<b>DBG_EN:</b> бит 6, адрес FE06h (ячейка ПЗУ: 07F6h)	<b>DBG_EN</b> – включение отладочного режима 1 – обычный режим 0 – отладочный режим
<b>BODEN:</b> бит 5, адрес FE05h (ячейка ПЗУ: 07F5h)	<b>BODEN</b> - включение схемы сброса по снижению напряжения питания: 1 = схема включена 0 = схема выключена
<b>PM0:</b> бит 4, адрес FE04h (ячейка ПЗУ: 07F4h)	<b>PM0</b> - биты выбора режима микроконтроллера: 1 = режим микроконтроллера 0 = режим микроконтроллера с защитой кодов программы (т.е. запрет считывания содержимого внутренней памяти программ)
<b>WDTPS1:</b> бит 3, адрес FE03h (ячейка ПЗУ: 07F3h) <b>WDTPS0:</b> бит 2, адрес FE02h (ячейка ПЗУ: 07F2h)	<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: 07F1h) <b>FOSC0:</b> бит 0, адрес FE00h (ячейка ПЗУ: 07F0h)	<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 1МГц до 10 МГц) 01 = RC генератор с частотой до 2 МГц 00 = LF – генератор с внешним низкочастотным кварцевым резонатором ( $\leq 1$ МГц)

Обозначение:

- = зарезервировано, читается как 0.

### **Внутрисхемное программирование микроконтроллера**

Для программирования внутренней EEPROM памяти программ микроконтроллеров 1886BE7У используется последовательный интерфейс ISP (Interface Serial Program). В этом режиме программирования задействованы выводы микроконтроллера и напряжения питания, приведённые в таблице 62.

**Таблица 62**  
Выводы ISP интерфейса

Обозначение	В режиме программирования		
	Назначение	Тип	Описание
PA2/RX/DT	DT	вход/выход	Последовательные данные
PA3/TX/CK	CK	вход	Последовательный синхросигнал
PA1/T0CLK	CLK	вход	Источник синхронизации микроконтроллера
TEST	TEST	вход	Вход для выбора режима
MCLRn	MCLRn	питание	Внешний сброс
U <sub>CC</sub>	U <sub>CC</sub>	питание	Напряжение питания
GND	GND	общий	Общий

### **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 T<sub>C</sub>. 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 nMCLR.

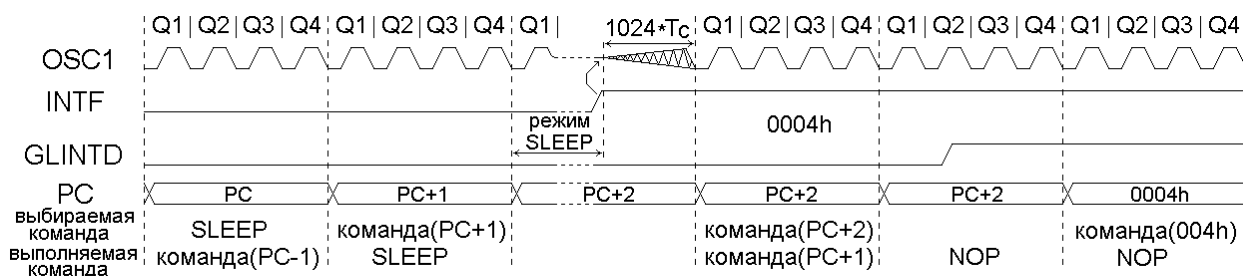
- Сброс от сторожевого таймера (если он включен).
- Прерывание с вывода PA0/INT, прерывание с PA1/T0CLK, или некоторые периферийные прерывания, от приемника и передатчика USART в синхронном ведомом режиме.

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 переходе в режим SLEEP необходимо отключить блоки EEPROM памяти программ и памяти данных. Для отключения EEPROM памяти программ необходимо установить бит ESLP в регистре CPUSTA. Для отключения EEPROM памяти данных необходимо очистить бит EN\_EE в регистре EE\_MODE. 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  $T_c$ .

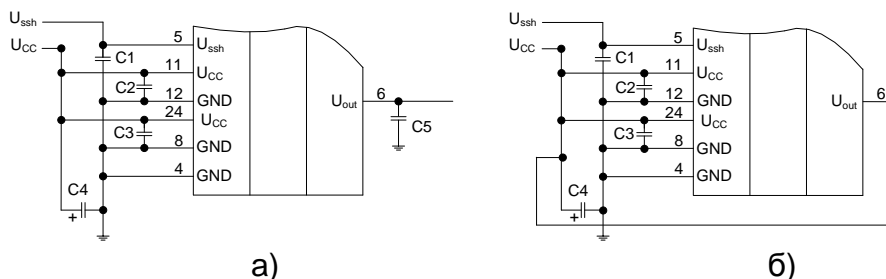


**Рис. 37** Выход из режима SLEEP по прерыванию

**Примечания:**

1. Режим генератора XT или LF. Задержки запуска генератора ( $1024 \cdot T_c$ ) 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**



**Рис. 38** Схема подключения напряжения питания

а) питание микроконтроллера от внешнего напряжения  $U_{cc} = 4,5-5,5$  В; Если встроенный линейный регулятор напряжения не используется, установка конденсаторов С1 и С5 не требуется.

б) питание микроконтроллера от встроенного линейного регулятора напряжения

Рекомендуемые значения номиналов конденсаторов:

- С1 - не менее 10,0 мкФ, не менее 20 В;
- С2...С3 - не менее 0,1 мкФ, не менее 7В, конденсаторы должны располагаться максимально близко к соответствующим выводам питания.
- С4 - не менее 10 мкФ, не менее 7 В;
- С5 = 1,0 мкФ.

## **Система команд**

Микроконтроллер поддерживает 58 команд (см. Таблица 63). 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 РС (т.е. результат операции записывается в PCL), а также команд чтения/записи таблиц в памяти программ (запись во внутреннюю EEPROM память имеет большую длительность). Коды команд приведены в Таблица 64. 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, изменяют записанный результат.
- операции с регистром PCL: чтение PCL приводит к загрузке в PCLATH значения регистра РСН, запись и чтение-модификация-запись приводит к загрузке в РСН значения регистра PCLATH.
- необходимо учитывать, что команды битовых операций производят операцию «чтение-модификация-запись» целого регистра.



**Таблица 63**  
Набор команд

<b>Мнемоника команды</b>	<b>Описание команды</b>	<b>Изменяемые флаги</b>	<b>Кол-во циклов</b>
<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
<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 → f<7:4>, s<7:4>. Если: [WREG<7:4> > 9].OR.[C = 1], то WREG<7:4> + 6 → f<7:4>, s<7:4>, иначе WREG<7:4> → f<7:4>, s<7:4>. Если: [WREG<3:0> > 9].OR.[DC = 1], то WREG<3:0> + 6 → f<3:0>, s<3:0>, иначе WREG<3:0> → 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

**Спецификация 1886BE7(71)У, K1886BE7(71)У, K1886BE71H4**

<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)
<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

**Спецификация 1886BE7(71)У, K1886BE7(71)У, K1886BE71H4**

<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
<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 f,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 «ПО» 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</b> <b>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</b> <b>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</b> <b>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</b> <b>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 64**  
Коды команд

<b>Мнемоника команды</b>	<b>Код команды</b>	<b>Мнемоника команды</b>	<b>Код команды</b>	<b>Мнемоника команды</b>	<b>Код команды</b>
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 ffff ffff	MOVLW k	1011 0000 kkkk kkkk	TABLRD t,i,f	1010 10ti ffff ffff
CLRWDT	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 65**

№ п/п	Наименование параметра, единица измерения	Об-ие параметра	Предельно-допустимые значения		Предельные значения	
			не менее	не более	не менее	не более
1	Напряжение источника питания, В	$U_{CC}$	4,5	5,5	–	7,0
2	Напряжение линейного регулятора, В	$U_{CC\_R}$	6,0	16,0	–	17,0
3	Входное напряжение высокого уровня, В, на выводах PA, OSC1	$U_{IH}$	$0,8 \cdot U_{CC}$	–	–	$U_{CC} + 0,3$
4	Входное напряжение низкого уровня, В, на выводах PA, OSC1	$U_{IL}$	–	$0,2 \cdot U_{CC}$	минус 0,3	–
5	Выходной ток линейного регулятора, мА	$I_{O\_R}$	–	50	–	60
6	Выходной ток высокого уровня, мА, - на выводах PA - на выводе OSC2	$I_{OH}$	–	4,0	–	10
			–	1,0	–	2
7	Выходной ток низкого уровня, мА, - на выводах PA - на выводе OSC2	$I_{OL}$	минус 4,0	–	минус 10	–
			минус 1,0	–	минус 2	–
8	Частота следования импульсов тактовых сигналов на выводах PA1 и OSC1, МГц,	$f_c$	–	10	–	–
9	Частота следования импульсов тактовых сигналов на выводах PA1 при программировании, стирании и чтении EEPROM программ, МГц,	$f_{C\_PROG}$	–	8,0	–	–
10	Длительность программирования одного слова, мс	$t_{CYW}$	4,0	–	–	–
11	Длительность сигнала высокого уровня синхронизации на входах PA1 и OSC1, нс, при: $U_{CC} = 4,5$ В	$t_{WH(PA\_OSC)}$	40	–	–	–
12	Время установления данных относительно OSC1 (15), нс, при: $U_{CC} = 4,5$ В	$t_{SU(OSC-D)}$	25	–	–	–

## Спецификация 1886BE7(71)У, K1886BE7(71)У, K1886BE71H4

№ п/п	Наименование параметра, единица измерения	Об-ие параметра	Предельно-допустимые значения		Предельные значения	
			не менее	не более	не менее	не более
13	Длительность сигнала высокого уровня прерывания INT/PA0, нс, при: $U_{CC}=4,5 В$	$t_{WH(INT)}$	25	–	–	–
14	Длительность сигнала низкого уровня системного сброса, $nMCLR$ , нс, при: $U_{CC}=4,5 В$	$t_{WL(MCLR)}$	100	–	–	–
<b>Параметры универсального последовательного синхронно-асинхронного приемопередатчика</b>						
15	Время установления данных RX/DT относительно синхронизации TX/CK, нс, при: $U_{CC}=4,5 В$	$t_{SU(TX-RX)}$	17	–	–	–
16	Время удержания данных RX/DT относительно синхронизации TX/CK, нс, при: $U_{CC}=4,5 В$	$t_{H(TX-RX)}$	17	–	–	–
<b>Параметры Таймера 0</b>						
17	Длительность сигнала высокого уровня на линии PA1/T0CK1, нс, прескалер включен при: $U_{CC}=4,5 В$	$t_{WH(PA1)}$	40	–	–	–
18	Длительность сигнала низкого уровня на линии PA1/T0CK1, нс, прескалер включен при: $U_{CC}=4,5 В$	$t_{WL(PA1)}$	40	–	–	–
19	Период сигнала на линии PA1/T1CLK(22), нс, при: $U_{CC}=4,5 В$	$T_{C(PD)}$	100 или $(4 \bullet T_{C} + 25)/N$	–	–	–
20	Длительность фронта входного сигнала линейного регулятора, мс	$t_{r\_R}$	0,1	100	–	–
21	Длительность фронта входного сигнала, нс, - на выводах: OSC1	$t_r$	–	5,0	–	–
	- на выводах: PA		–	20	–	–
22	Длительность спада входного сигнала, нс, - на выводах: OSC1	$t_f$	–	5,0	–	–
	- на выводах: PA		–	20	–	–
23	Емкость нагрузки, пФ	$C_L$	–	40	–	–

**Примечание:**

1. Не допускается одновременное воздействие двух и более предельных режимов.

2. n - в названии вывода - обозначает инверсию.
3. Стойкость к воздействию статического электричества – не менее 2 кВ.

**Электрические параметры микросхемы**

**Таблица 66**

№	Наименование параметр, Единица измерения	Обозначение параметра	Норма		Температура среды (корпуса), °С
			Не менее	Не более	
1	Выходное напряжение низкого уровня, В, на выводах PA (14, 15, 18–23), при: UCC= 4,5 В, IOL= 4,0 мА на выводе OSC2 (28) при: UCC= 4,5 В, IOL= 1,0 мА	UOL	–	0,45	25, 125, минус 60
2	Выходное напряжение высокого уровня, В, на выводах PA (14, 15, 18–23), при: UCC= 4,5 В, IOL= 4,0 мА на выводе OSC2 (28) при: UCC= 4,5 В, IOL= 1,0 мА	UOH	4,05	–	25, 125, минус 60
3	Уровень напряжения для срабатывания схемы генерации сброса, В	UBOR	3,6	4,3	25, 125, минус 60
4	Выходное напряжение регулятора, В, при: UCC= 5,5 В, UCC_S=16 В, UCC_S=6 В, IO_S=0 мА, IO_R=50 мА, CL=4,7мкФ	UO_R	4,7	5,3	25, 125, минус 60
5	Статический ток потребления в режиме покоя (без учета потребления регулятора, память программ выключена), мкА, UCC= 5,5 В	ICCS1	–	5,0	25, минус 60
			–	20	125
6	Статический ток потребления в режиме покоя (без учета потребления стабилизатора, память программ включена), мкА, UCC= 5,5 В	ICCS2	–	30	25, 125, минус 60
7	Динамический ток потребления, мА, при: UCC= 5,5 В, fC= 10 МГц при: UCC= 5,5 В, fC= 200 кГц	IOCC	–	5,0	25, 125,
			–	0,16	минус 60
8	Входной ток утечки высокого/низкого уровня, мкА на цифровых выводах	IILH IILL	–	±1,0	25, 125, минус 60
9	Входной ток утечки в состоянии «выключено», мкА при: UI=0, и 5,5 В на выводах OSC3 (2) и OSC4 (27)	ILOH ILOL	–	±1,0	25, 125, минус 60

## Спецификация 1886BE7(71)У, K1886BE7(71)У, K1886BE71H4

№	Наименование параметр, Единица измерения	Обозначение параметра	Норма		Температура среды (корпуса), °С
			Не менее	Не более	
10	Ток потребления регулятора в режиме «холостого хода», мкА UCC_R= 16 В, IO_R=0 мА	ICC_R	–	8,0	25, 125, минус 60
11	Период работы внутреннего RC – генератора сторожевого таймера, мкс при: UCC= 5,5 В	TRC	60	140	25, 125, минус 60
12	Время старта микроконтроллера (от подачи питания до начала исполнения программ), мс - 1886BE7У - 1886BE71У	TST	63 9	146 21	25, 125, минус 60
13	Время задержки данных относительно OSC1(1), нс, на выводах (14, 15, 18–23) при: UCC= 4,5 В	td(OSC–D)	–	100	25, 125, минус 60
14	Длительность спада выходных сигналов, нс, при: CL= 10 пФ при: CL= 40 пФ	tf	–	2 5	25, 125, минус 60
15	Длительность фронта выходных сигналов, нс, при: CL= 10 пФ при: CL= 40 пФ	tr	–	2 5	25, 125, минус 60
16	Время включения регулятора, мс, при: UCC_R =6 В, IO_R =50 мА, CL=1 мкФ	tON	–	2,5	25, 125, минус 60
Параметры универсального последовательного синхронно-асинхронного приемопередатчика					
17	Время задержки данных RX/DT (18) от фронта TX/CK (19), нс при: UCC= 4,5 В	td(TX–RX)	–	50	25, 125, минус 60
Параметры встроенного тактового генератора в режиме RC					
18	Диапазон изменения частоты при изменении напряжения питания в диапазоне 4,5 – 5,5 В, %, при R0=12 кОм, R1=12кОм, C0=30 пФ, н.у.	ΔF	–	0,5	25, 125, минус 60
19	Температурный коэффициент, ppm*/°С, (без учета параметров внешних RC компонентов)	K	–	25	25, 125, минус 60

Примечание: \* - ppm – 1 \* 10<sup>-6</sup>



**Электрические параметры микросхемы К1886BE71Н4  
(бескорпусное исполнение)****Таблица 67**

№ п/п	Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра	Температура среды, °С
1	Выходное напряжение регулятора, В при: UCC_S=16 В, UCC_S=6В, ILVR=0 мА, ILVR=50 мА, CL=4,7 мкФ	UO_R	4,7 – 5,3	25
2	Статический ток потребления в режиме покоя (без учета потребления стабилизатора, память программ выключена), мкА не более	ICC1	5,0	25
3	Статический ток потребления в режиме покоя (без учета потребления стабилизатора, память программ включена), мкА не более	ICC2	30	25
4	Функциональный контроль	ФК	–	25



**Типовые зависимости**

**Таблица 68**

Зависимость частоты встроенного тактового генератора в режиме RC  
от величин внешних RC компонентов при  $U_{CC}=5,5$  В,  $T=25^{\circ}\text{C}$

Номинал резисторов $R0^*$ , $R1^*$ , кОм	Номинал конденсатора $C0^*$ , пФ	Частота генератора на OSC1, кГц
12	30	1870,8
12	120	485,2
12	180	330,2
12	280	212,8
6	280	421,6

*Примечание:* \* - см. рис. 3.

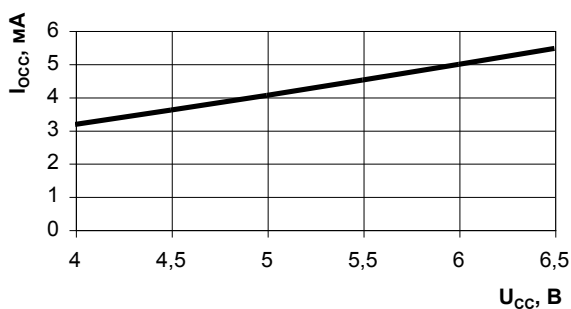


Рисунок 39 Зависимость динамического тока потребления  $I_{CC}$  от напряжения питания, при:  $T=25^{\circ}\text{C}$ ,  $f_c = 10$  МГц

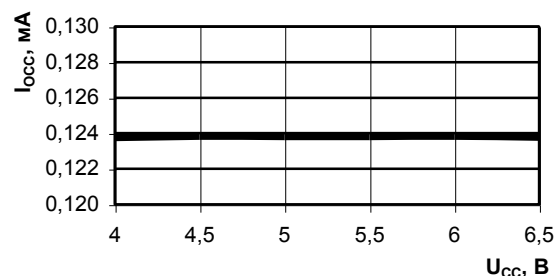


Рисунок 40 Зависимость динамического тока потребления  $I_{CC}$  от напряжения питания, при:  $T=25^{\circ}\text{C}$ ,  $f_c = 200$  кГц

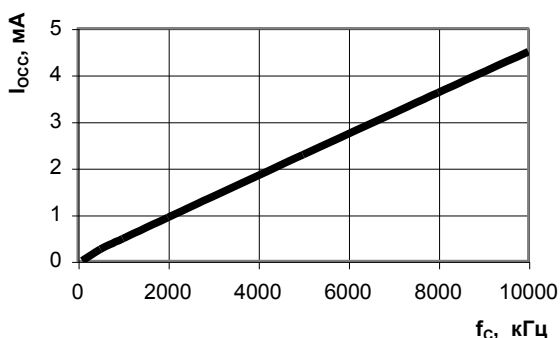


Рисунок 41 Зависимость динамического тока потребления  $I_{CC}$  от частоты следования импульсов тактовых сигналов, при  $T=25^{\circ}\text{C}$ ,  $U_{CC} = 5,5$  В

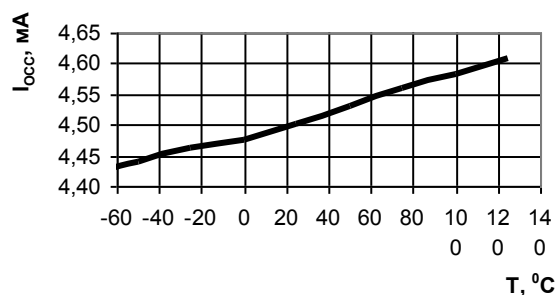


Рисунок 42 Зависимость динамического тока потребления  $I_{CC}$  от температуры, при:  $f_c=10$  МГц,  $U_{CC}= 5,5$  В

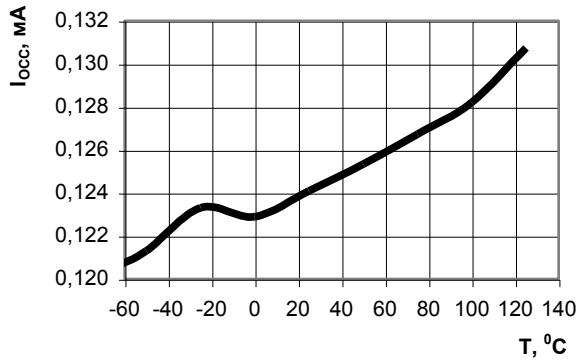


Рисунок 43 Зависимость динамического тока потребления  $I_{OCC}$  от температуры, при:  $f_c=200$  кГц,  $U_{CC}=5,5$  В

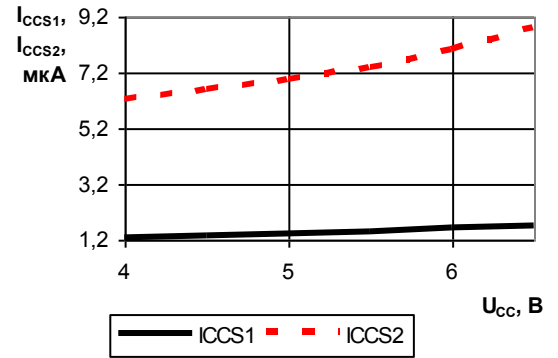


Рисунок 44 Зависимость статического тока потребления  $I_{OCC1}$  и  $I_{OCC2}$  от напряжения питания, при:  $T=25$  °C

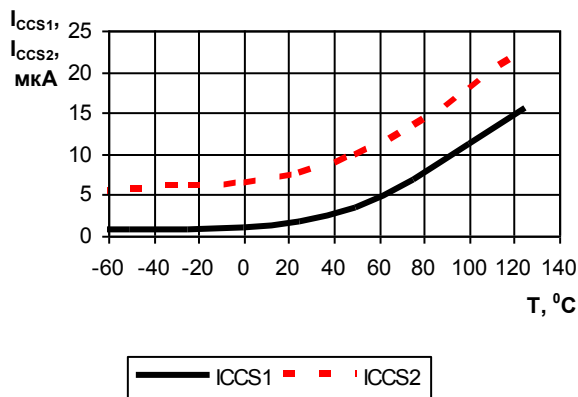


Рисунок 45 Зависимость статического тока потребления  $I_{OCC1}$ ,  $I_{OCC2}$  от температуры, при:  $U_{CC} = 5,5$  В

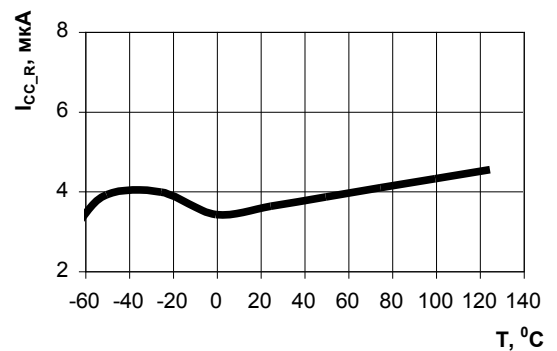


Рисунок 46 Зависимость тока потребления регулятора в режиме «холостого хода»  $I_{OCC\_R}$  от температуры, при:  $U_{CC\_R}=16$  В

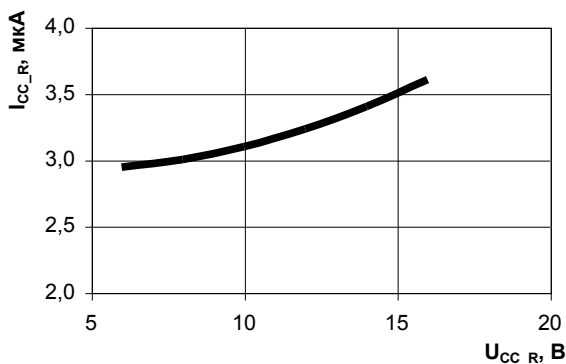


Рисунок 47 Зависимость тока потребления регулятора в режиме «холостого хода»  $I_{OCC\_R}$  от напряжения линейного регулятора, при:  $T=25$  °C

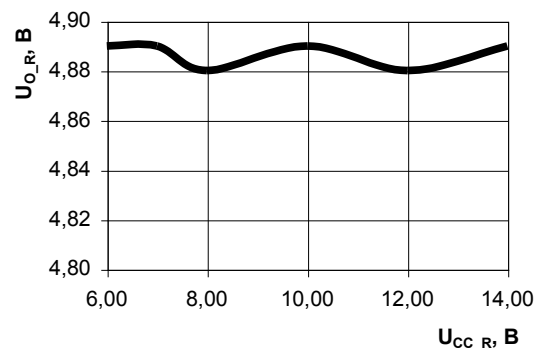


Рисунок 48 Зависимость выходного напряжения регулятора от входного напряжения линейного регулятора

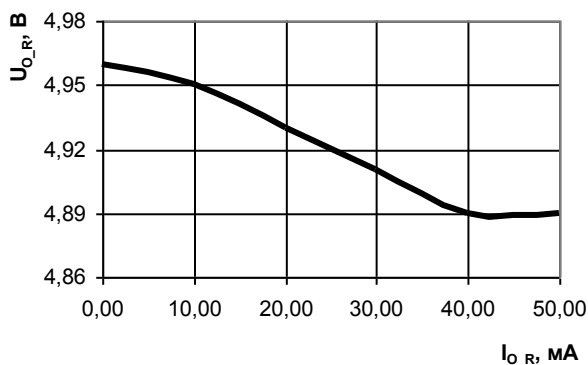


Рисунок 49 Зависимость выходного напряжения регулятора от выходного тока линейного регулятора

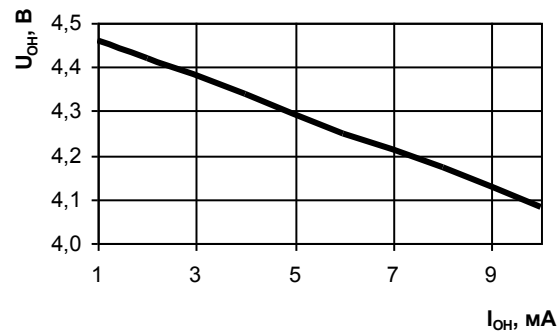


Рисунок 50 Зависимость выходного напряжения высокого уровня  $U_{OH}$  на выводах PA от выходного тока высокого уровня, при:  $T = 25\text{ }^{\circ}\text{C}$ ,  $U_{CC} = 4,5\text{ В}$

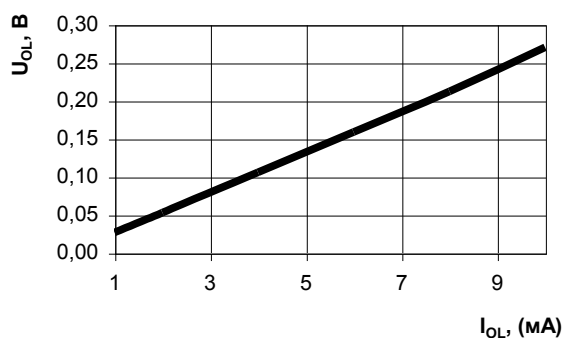
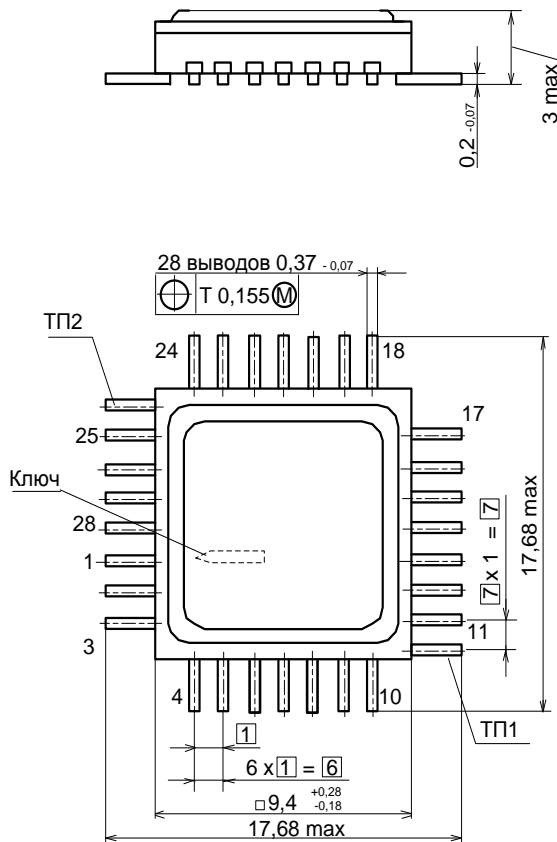
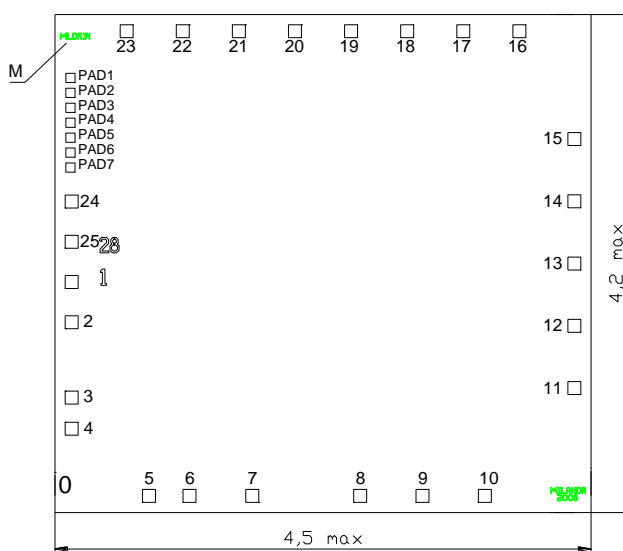


Рисунок 51 Зависимость выходного напряжения высокого уровня  $U_{OH}$  на выводах PA от выходного тока низкого уровня, при:  $T = 25\text{ }^{\circ}\text{C}$ ,  $U_{CC} = 4,5\text{ В}$

**Габаритный чертеж микросхемы**



**Рис. 39 Корпус H09.28-1B**



**Рис. 40 Кристалл (бескорпусное исполнение)**

**Примечания:**

- 1 Номера контактным площадкам (КП) присвоены условно, кроме первой. Расположение КП соответствует топологическому чертежу;
- 2 PAD1...PAD7 – КП для пережигания перемычек (размеры КП 60 × 60 мкм)
- 3 М – маркировка кристалла V2 MLDR34.

**Спецификация 1886BE7(71)У, К1886BE7(71)У, К1886BE71Н4**

№ КП	Координаты X	Координаты Y	№ КП	Координаты X	Координаты Y
1	0.00	1717.80	14	4032.00	2366.00
2	0.00	1394.80	15	4032.00	2866.00
3	0.00	789.50	16	3591.00	3732.00
4	0.00	539.50	17	3141.00	3732.00
5	620.15	0.00	18	2691.00	3732.00
6	945.10	0.00	19	2241.00	3732.00
7	1448.55	0.00	20	1791.00	3732.00
8	2134.00	0.00	21	1341.00	3732.00
9	2814.00	0.00	22	891.00	3732.00
10	3314.00	0.00	23	441.00	3732.00
11	4032.00	866.00	24	0.00	2363.80
12	4032.00	1366.00	25	0.00	2040.80
13	4032.00	1866.00	26		
<b>Площадки пережигаемых перемычек</b>					
RM1	-10.00	3357.00			
RM3	-10.00	3237.00			
RP3	-10.00	2877.00			
RP1	-10.00	2757.00			
RP0	-10.00	2637.00			

**Информация для заказа**

<b>Обозначение</b>	<b>Маркировка</b>	<b>Тип корпуса</b>	<b>Температурный диапазон</b>
1886BE7У	1886BE7У	Н09.28-1В	минус 60 – 125 °С
К1886BE7У	К1886BE7У	Н09.28-1В	минус 60 – 125 °С
К1886BE7УК	К1886BE7У•	Н09.28-1В	0 – 70 °С
1886BE71У	1886BE71У	Н09.28-1В	минус 60 – 125 °С
К1886BE71У	К1886BE71У	Н09.28-1В	минус 60 – 125 °С
К1886BE71УК	К1886BE71У•	Н09.28-1В	0 – 70 °С

*Примечание:*

Микросхемы в бескорпусном исполнении поставляются в виде отдельных кристаллов, получаемых разделением пластины. Микросхемы поставляются в таре (кейсах) без потери ориентации. Маркировка микросхемы – К1886BE71Н4, наносится на тару.

Микросхемы с приемкой «ВП» маркируются ромбом.

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

**Лист регистрации изменений**

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
1	03.02.2010	1.6	1. Исправлены названия выводов, регистров, битов, адресов. 2. Введен лист регистрации изменений	1, 10, 13, 14, 16-19, 31-35, 51, 59, 68, 87
2	26.03.2010	1.7	Корректировка в связи с плановым пересмотром документации	1, 98
3	27.04.2010	1.8	Замена логотипа	1
4	25.05.2010	1.9	Добавлено исполнение 1886BE71У	1, 8, 14, 16, 17, 32, 34, 52, 99, 101
5	08.06.2010	1.10	1. Изменен рис.38. 2. Приведены в соответствие с ТУ табл. 65, 66	87 96-99
6	01.10.2010	2.0	1. Введены типовые зависимости. 2. Приведены в соответствие с ТУ: - рис.10 - табл.65 - табл.68.	102-104  18 97 101
7	04.04.2011	2.1	Уточнение описания работы микросхемы	18, 25, 52
8	05.08.2011	2.2	Исправлены ошибки	1, 105
9	10.10.2011	2.3	Уточнение наименования микросхем	По тексту
10	22.07.2014	2.4	Введение бескорпусного исполнения микросхемы	По тексту
11	16.04.2015	2.5.0	Добавлен ГЧ кристалла. Дополнена таблица 1	105, 106 6
12	14.10.2015	2.6.0	Исправлен рисунок 40	103
13	12.09.2016	2.7.0	Добавлен раздел «Указания по применению» Отредактированы и дополнены подразделы «Описание блока управления EEPROM – памятью программ» и «Описание регистров»	7 91 – 98
14	25.07.2017	2.7.1	Исправлена опечатка	121