

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра радиотехнических систем

В.Н. Левкович, А.И. Бурак

***ВЫЧИСЛИТЕЛЬНЫЕ И КОММУНИКАЦИОННЫЕ
СРЕДСТВА РАДИОСИСТЕМ***

Лабораторный практикум
для студентов специальности
39 01 02 «Радиоэлектронные системы»
дневной формы обучения

В 2-х частях

Часть 1

Минск 2004

УДК 004.31(075.8)
ББК 32.973 я 73
Л 34

Р е ц е н з е н т:

доцент кафедры сетей и устройств телекоммуникаций БГУИР,
канд. техн. наук И.И. Астровский

Левкович В.Н.

Л 34

Вычислительные и коммуникационные средства радиосистем: Лаб. практикум для студ. спец. 39 01 02 «Радиоэлектронные системы» дневн. формы обуч. В 2 ч. Ч. 1/ В.Н. Левкович, А.И. Бурак. -Мн: БГУИР, 2004. - 59 с.: ил.

ISBN 985-444-745-6 (ч.1)

Лабораторный практикум содержит краткие теоретические сведения для выполнения лабораторных работ по следующим темам: 1) «Последовательный синхронный ввод/вывод данных», 2) «Последовательный асинхронный ввод/вывод данных», 3) «Цифроаналоговое преобразование на основе широтно-импульсной модуляции», 4) «Аналого-цифровое преобразование».

В часть 1, кроме того, вошло описание двух лабораторных работ (1-я и 2-я темы), приводятся также варианты заданий для выполнения на лабораторных занятиях.

УДК 004.31(075.8)
ББК 32.973 я 73

ISBN 985-444-745-6 (ч.1)
ISBN 985-444-744-8

©Левкович В.Н., Бурак А.И., 2004
© БГУИР, 2004

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

1. ОБЩИЕ СВЕДЕНИЯ ОБ АРХИТЕКТУРЕ ОДНОКРИСТАЛЬНОГО МИКРОКОНТРОЛЛЕРА PIC16F877

1.1. Структурная схема

1.2. Организация памяти программ

1.2.1. Регистры PCN и PCLATH

1.2.2. Вычисляемый переход

1.2.3. Стек

1.2.4. Страницы памяти программ

1.3. Организация памяти данных

1.3.1. Регистры общего назначения

1.3.2. Регистры специального назначения

1.3.2.1. Регистр STATUS

1.3.2.2. Регистр OPTION_REG

1.3.2.3. Регистр INTCON

1.3.2.4. Регистр PIE1

1.3.2.5. Регистр PIR1

1.3.2.6. Регистр PIE2

1.3.2.7. Регистр PIR2

1.3.3. Косвенная адресация, регистры INDF и FSR

1.4. Прерывания

1.4.1. Внешнее прерывание с входа RB0/INT

1.4.2. Прерывание по переполнению TMR0

1.4.3. Прерывание по изменению уровня сигнала на входах RB7:RB4

1.4.4. Сохранение контекста при обработке прерываний

1.5. Система команд

2. ЛАБОРАТОРНАЯ РАБОТА

«Последовательный синхронный ввод/вывод данных»

- 2.1. Структурная схема аппаратной части
- 2.2. Управляющие регистры
 - 2.2.1. Регистр SSPSTAT статуса модуля MSSP
 - 2.2.2. Регистр SSPCON управления модуля MSSP
 - 2.2.3. Настройка выводов в режиме SPI
 - 2.2.4. Типовое включение
 - 2.2.5. Режим ведущего SPI
 - 2.2.6. Работа в SLEEP-режиме микроконтроллера
 - 2.2.7. Эффект сброса
- 2.3. Примеры процедур инициализации модуля MSSP, передачи и приема байта данных
- 2.4. Варианты лабораторных заданий

3. ЛАБОРАТОРНАЯ РАБОТА

«Последовательный асинхронный ввод/вывод данных»

- 3.1. Управляющие регистры
 - 3.1.1. TXSTA - регистр управления и статуса передатчика
 - 3.1.2. RCSTA - регистр управления и статуса приемника
- 3.2. Генератор частоты обмена USART BRG
- 3.3. Выборка
- 3.4. Асинхронный режим USART
- 3.5. Асинхронный передатчик USART
- 3.6. Асинхронный приемник USART
- 3.7. Настройка 9-разрядного асинхронного приема с детектированием адреса
- 3.8. Пример управляющей программы для USART
- 3.9. Варианты лабораторных заданий

ЛИТЕРАТУРА

ВВЕДЕНИЕ

Лабораторный практикум содержит учебно-методические материалы для выполнения лабораторных работ по следующим темам:

- 1) «Последовательный синхронный ввод/вывод данных»,
- 2) «Последовательный асинхронный ввод/вывод данных»,
- 3) «Цифроаналоговое преобразование на основе широтно-импульсной модуляции»,
- 4) «Аналого-цифровое преобразование».

Даны краткие теоретические сведения по аппаратной организации ядра однокристального микроконтроллера PIC16F877, его системе команд, а также по встроенным в однокристальный микроконтроллер периферийным модулям: аналого-цифрового преобразования, широтно-импульсной модуляции (ШИМ), последовательного синхронного порта и универсального синхронно-асинхронного последовательного порта.

Даны примеры программ настройки и использования вышеперечисленных модулей для решения практических задач. Приводятся также варианты заданий студентам по каждой теме для выполнения на лабораторных работах.

Темы изучаются на базе периферийных модулей, размещенных на одном кристалле с ядром микроконтроллера.

В ч. 1 практикума приведено описание двух лабораторных работ (темы 1, 2).

1. ОБЩИЕ СВЕДЕНИЯ ОБ АРХИТЕКТУРЕ ОДНОКРИСТАЛЬНОГО МИКРОКОНТРОЛЛЕРА PIC16F877

1.1. Структурная схема

Семейство однокристальных микроконтроллеров (ОМК) PIC16F87X включает четыре модели: PIC16F873, PIC16F874, PIC16F876, PIC16F877. Они различаются между собой объемами памяти программ, памяти данных, количеством внешних портов, периферийными устройствами на кристалле. Рассмотрим самую сложную и совершенную среди них модель - PIC16F877.

Структурная схема Омк показана на рис.1. Назначение выводов микроконтроллера PIC16F877 отражено в табл. 1.

Основные параметры и характеристики ядра микроконтроллера:

- высокоскоростная RISC архитектура;
- 35 инструкций (команд);
- все команды выполняются за один цикл, кроме команд переходов, выполняемых за два цикла;
- максимальная тактовая частота – 20 МГц;
- длительность одного машинного цикла – 200 нс;
- память программ 8К x 14 слов FLASH;
- память данных (ОЗУ) 368 x 8 байтов;
- EEPROM память данных - 256 x 8 байтов;
- система прерываний - 14 источников;
- 8-уровневый аппаратный стек;
- прямой, косвенный и относительный способы адресации;
- сброс по включению питания (POR);
- таймер сброса (PWRT) и таймер ожидания запуска генератора (OST) после включения питания;
- сторожевой таймер WDT с собственным RC-генератором;
- программируемая защита памяти программ;

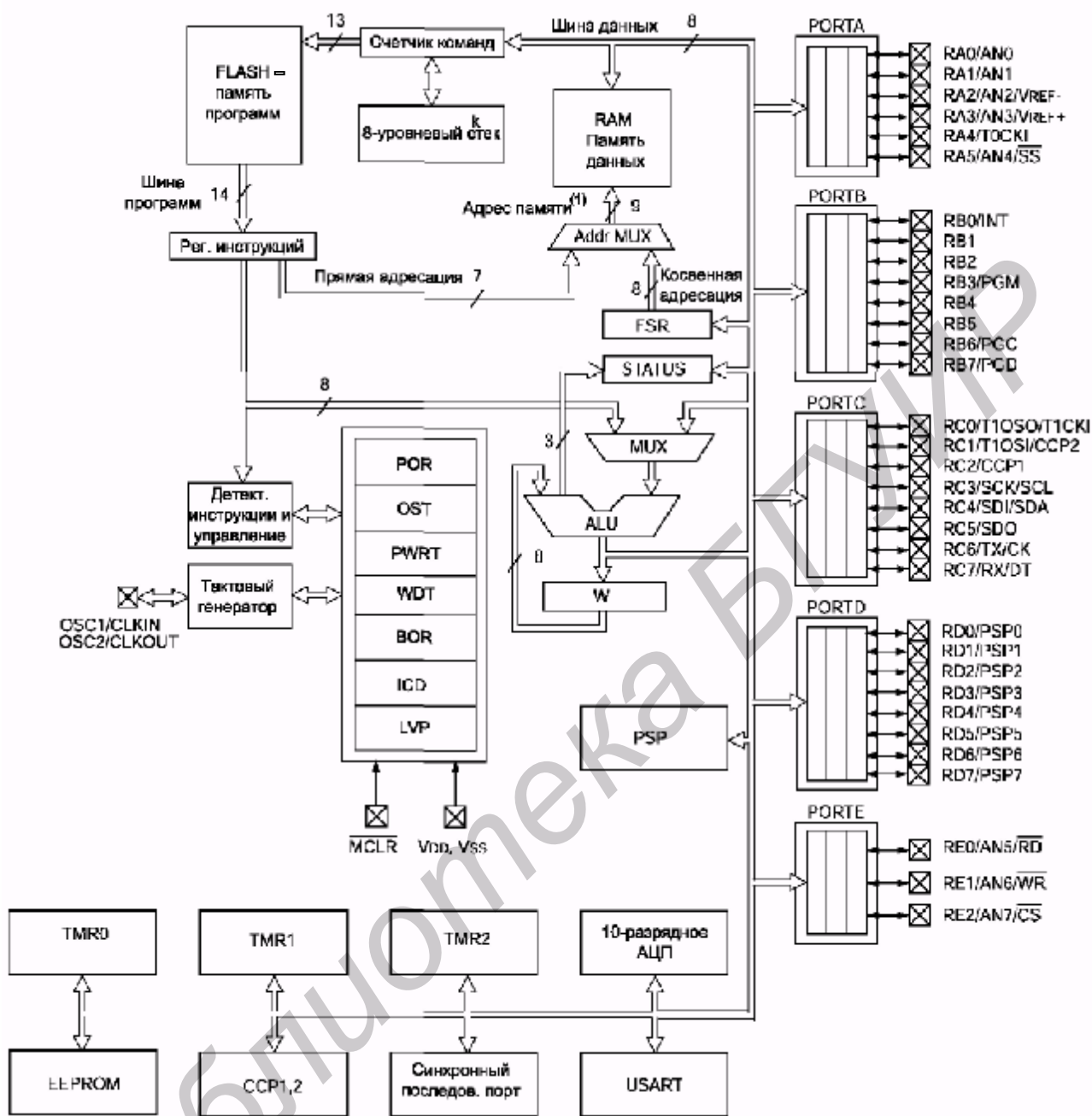


Рис. 1. Структурная схема однокристалльного микроконтроллера PIC16F877

- режим энергосбережения SLEEP;
- выбор параметров тактового генератора;
- все элементы памяти статического типа, состояние ОМК не искажается даже при останове тактового генератора;

Таблица 1

| Обозначение вывода | № вывода | Тип I/O/P | Тип буфера в режиме ввода | Описание |
|-----------------------------------------------------------------------------------------------------------------------|----------|-----------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | 2 | 3 | 4 | 5 |
| OSC1/CLKIN | 13 | 1 | ST/CMOS | Вход генератора/ вход внешнего тактового сигнала |
| OSC2/CLKOUT | 14 | 0 | - | Выход генератора. Подключается кварцевый или керамический резонатор. В RC-режиме тактового генератора на выходе OSC2 присутствует тактовый сигнал CLKOUT, равный $F_{osc}/4$ |
| -MCLR/VPP | 1 | I/P | ST | Вход сброса микроконтроллера или вход напряжения программирования. Сброс микроконтроллера происходит при низком логическом уровне сигнала на входе |
| <i>Двунаправленный порт ввода/вывода PORTA</i> | | | | |
| RA0/AN0 | 2 | I/O | TTL | RA0 может быть настроен как аналоговый канал 0 |
| RA1/AN1 | 3 | I/O | TTL | RA1 может быть настроен как аналоговый канал 1 |
| RA2/AN2/VREF- | 4 | I/O | TTL | RA2 может быть настроен как аналоговый канал 2 или вход отрицательного опорного напряжения |
| RA3/AN3/V _{REF+} | 5 | I/O | TTL | RA3 может быть настроен как аналоговый канал 3 или вход положительного опорного напряжения |
| RA4/T0CKI | 6 | I/O | ST | RA4 может использоваться в качестве входа внешнего тактового сигнала для TMR0. Выход с открытым стоком |
| RA5/-SS/AN4 | 7 | I/O | TTL | RA1 может быть настроен как аналоговый канал 1 или вход выбора микросхемы в режиме ведомого SPI |
| <i>Двунаправленный порт ввода/вывода PORTB. PORTB имеет программно подключаемые подтягивающие резисторы на входах</i> | | | | |
| RB0/INT | 33 | I/O | TTL/ST | RB0 может использоваться в качестве входа внешних прерываний |
| RB1 | 34 | I/O | TTL | |
| RB2 | 35 | I/O | TTL | |
| RB3/PGM | 36 | I/O | TTL | RB3 может использоваться в качестве входа для режима низковольтного программирования |
| RB4 | 37 | I/O | TTL | Прерывания по изменению уровня входного сигнала |
| RB5 | 38 | I/O | TTL | Прерывания по изменению уровня входного сигнала |
| RB6/PGC | 39 | I/O | TTL/ST | Прерывания по изменению уровня входного сигнала или вывод для режима внутрисхемной отладки ICD. Тактовый вход в режиме программирования |
| RB7/PGD | 40 | I/O | TTL/ST | Прерывания по изменению уровня входного сигнала или вывод для режима внутрисхемной отладки ICD. Вывод данных в режиме программирования |

Окончание табл. 1

| 1 | 2 | 3 | 4 | 5 |
|-------------------------------------------------|-------|-----|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Двунаправленный порт ввода/вывода PORTC.</i> | | | | |
| RC0/ T1OSO/ T1CKI | 15 | I/O | ST | RC0 может использоваться в качестве выхода генератора TMR1 или входа внешнего тактового сигнала для TMR1 |
| RC1/T1OSI/ CCP2 | 16 | I/O | ST | RC1 может использоваться в качестве входа генератора для TMR1 или вывода модуля CCP2 |
| RC2/CCP1 | 17 | I/O | ST | RC2 может использоваться в качестве вывода модуля CCP1 |
| RC3/SCK/ SCL | 18 | I/O | ST | RC3 может использоваться в качестве входа/выхода тактового сигнала в режиме SPI и I ² C |
| RC4/SDI/ SDA | 23 | I/O | ST | RC4 может использоваться в качестве входа данных в режиме SPI или вход/выход данных в режиме I ² C |
| RC5/SDO | 24 | I/O | ST | RC5 может использоваться в качестве выхода данных в режиме SPI |
| RC6/TX/CK | 25 | I/O | ST | RC6 может использоваться в качестве вывода передатчика USART в асинхронном режиме или вывода синхронизации USART в синхронном режиме |
| RC7/RX/DT | 26 | I/O | ST | RC6 может использоваться в качестве вывода приемника USART в асинхронном режиме или вывода данных USART в синхронном режиме. Двунаправленный порт ввода/вывода PORTD или ведомый параллельный порт для подключения к шине микропроцессора |
| <i>Двунаправленный порт ввода/вывода PORTD</i> | | | | |
| RD0/PSP0 | 19 | I/O | ST/TTL | |
| RD1/PSP1 | 20 | I/O | ST/TTL | |
| RD2/PSP2 | 21 | I/O | ST/TTL | |
| RD3/PSP3 | 22 | I/O | ST/TTL | |
| RD4/PSP4 | 27 | I/O | ST/TTL | |
| RD5/PSP5 | 28 | I/O | ST/TTL | |
| RD6/PSP6 | 29 | I/O | ST/TTL | |
| RD7/PSP7 | 30 | I/O | ST/TTL | |
| <i>Двунаправленный порт ввода/вывода PORTE</i> | | | | |
| RE0/ -RD/AN5 | 8 | I/O | ST/TTL | RE0 может использоваться в качестве управляющего входа чтения PSP или аналогового канала 5 |
| RE1/-WR/ AN6 | 9 | I/O | ST/TTL | RE1 может использоваться в качестве управляющего входа записи PSP или аналогового канала 6 |
| RE2/-CS/ AN7 | 10 | I/O | ST/TTL | RE2 может использоваться в качестве управляющего входа выбора PSP или аналогового канала 7 |
| V _{SS} | 12,31 | p | | Общий вывод для внутренней логики и портов ввода/вывода |
| V _{DD} | 11,32 | p | | Положительное напряжение питания для внутренней логики и портов ввода/вывода |

Обозначения: I - вход, O - выход, I/O - вход/выход, P - питание,
TTL - входной буфер TTL, ST - вход с триггером Шмитта.

- возможность программирования в готовом устройстве через два вывода микроконтроллера;
- низковольтный режим программирования;
- режим внутрисхемной отладки (используется два вывода микроконтроллера);
- широкий диапазон напряжений питания от 2,0 В до 5,5 В;
- повышенная нагрузочная способность портов ввода/вывода (25 мА);
- малое энергопотребление: менее 0,6 мА при питании 3 В и тактовой частоте 4 МГц; 20 мкА при питании 3 В и частоте 32 кГц; менее 1 мкА в режиме энергосбережения.

Основные характеристики встроенных периферийных модулей:

- таймер 0: 8-разрядный таймер/счетчик с 8-разрядным программируемым предварительным делителем;
- таймер 1: 16-разрядный таймер/счетчик с возможностью подключения внешнего резонатора;
- таймер 2: 8-разрядный таймер/счетчик с 8-разрядным программируемым предварительным делителем и выходным делителем;
- два модуля сравнение/захват/ШИМ (ССР). 16-разрядный захват (максимальная разрешающая способность 12,5 нс), 16-разрядное сравнение (максимальная разрешающая способность 200 нс), 10-разрядный ШИМ;
- многоканальное 10-разрядное аналого-цифровое преобразование (АЦП);
- последовательный синхронный порт MSSP: ведущий/ведомый режима SPI, ведущий/ведомый режима I²C;
- последовательный синхронно-асинхронный приемопередатчик USART с поддержкой детектирования адреса;
- ведомый 8-разрядный параллельный порт PSP с поддержкой внешних сигналов -RD, -WR, -CS;
- детектор пониженного напряжения (BOD) для сброса по снижению напряжения питания (BOR).

1.2. Организация памяти программ

В микроконтроллерах PIC16F87X имеется три вида памяти: программ, данных и энергонезависимая память данных EEPROM.

Память программ и память данных имеют отдельные шины данных и адреса, что позволяет иметь к ним параллельный доступ.

Микроконтроллер PIC16F877 имеет 13-разрядный счетчик команд PC, способный адресовать 8К 14-разрядных слов памяти программ, и 8-уровневый стек. Адрес вектора сброса – 0000h. Адрес вектора прерываний – 0004h.

1.2.1. Регистры PCH и PCLATH

Счетчик команд PC указывает адрес выполняемой инструкции (команды). Младший байт счетчика команд PCL доступен для чтения и записи. Старший байт PCH, содержащий <12:8> биты счетчика команд PC, не доступен для чтения и записи. Все операции с регистром PCH происходят через дополнительный регистр PCLATH. При любом виде сброса микроконтроллера счетчик команд PC очищается.

На рис. 2 показано две ситуации загрузки значения в счетчик команд PC. В примере сверху запись в счетчик команд PC происходит при записи значения в регистр PCL (PCLATH <4:0> → PCH). В примере снизу запись значения в счетчик команд PC происходит при выполнении команды CALL или GOTO (PCLATH <4:3> → PCH).

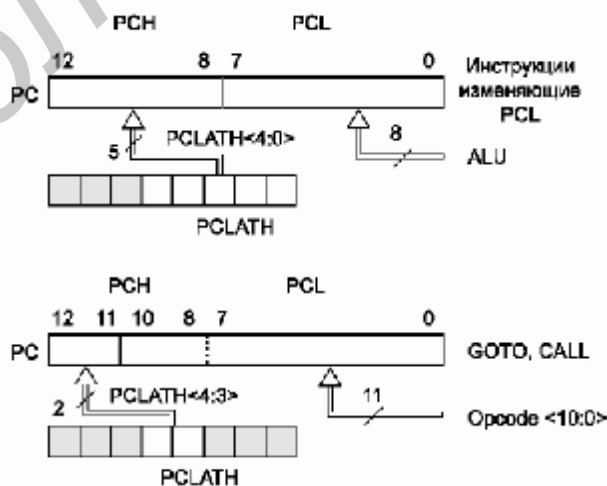


Рис. 2. Запись значения в счетчик команд PC

1.2.2. Вычисляемый переход

Вычисляемый переход может быть выполнен командой приращения к регистру PCL (например ADDWF PCL). При выполнении табличного чтения вычисляемым переходом следует заботиться о том, чтобы значение PCL не пересекло границу блока памяти (каждый блок 256 байтов).

1.2.3. Стек

PIC16F87X имеет 8-уровневый 13-разрядный аппаратный стек. Стек не имеет отображения на память программ и память данных, нельзя записать или прочитать данные из стека. Значение счетчика команд заносится в вершину стека при выполнении инструкций перехода на подпрограмму (CALL) или обработки прерываний. Чтение из стека и запись в счетчик команд PC происходит при выполнении инструкций возвращения из подпрограммы или обработки прерываний (RETURN, RETLW, RETFIE), при этом значение регистра PCLATH не изменяется.

Стек работает как циклический буфер. После восьми записей в стек, девятая запись запишется на место первой, а десятая запись заменит вторую и так далее.

В микроконтроллере не имеется никаких указателей о переполнении стека.

В микроконтроллере не предусмотрено команд записи/чтения из стека, кроме команд вызова/возврата из подпрограмм (CALL, RETURN, RETLW и RETFIE) или условий перехода по вектору прерываний.

1.2.4. Страницы памяти программ

Микроконтроллер PIC16F877 способен адресовать 8К слов памяти программ. Инструкции переходов (CALL и GOTO) имеют 11-разрядное поле для указания адреса, что позволяет непосредственно адресовать только 2К слов памяти программ. Для адресации верхних страниц памяти программ используются 2 бита в регистре PCLATH<4:3>. Перед выполнением команды

перехода (CALL или GOTO) необходимо запрограммировать биты регистра PCLATH<4:3> для адресации требуемой страницы.

При выполнении инструкций возврата из подпрограммы 13-разрядное значение для счетчика программ PC берется из вершины стека, поэтому манипуляция битами регистра PCLATH<3:4> не требуется.

Содержимое регистра PCLATH не изменяется после выполнения инструкции RETURN или RETFIE. Пользователь должен сам изменить значения регистра PCLATH для последующего выполнения команд GOTO и CALL.

В примере ниже показан переход со страницы 0 на страницу 1 памяти программ. Этот пример предполагает, что в подпрограмме сохраняется и восстанавливается значение регистра PCLATH.

```
ORG      0x500
BSF      PCLATH,3      ; Выбор страницы 1 (800h-FFFh)
CALL     SUB1_P1       ; Переход на страницу 1 (800h-FFFh)
...
ORG      0x900
SUB1_P1: ; Страница 1 (800h-FFFh)
...
RETURN   ; Возврат на страницу 0 (000h-7FFh)
```

1.3. Организация памяти данных

Память данных разделена на четыре банка, которые содержат регистры общего и специального назначения.

Биты RP1 (STATUS<6>) и RP0 (STATUS<5>) предназначены для управления банками данных.

В табл. 2 показано состояние управляющих битов при обращении к банкам памяти данных.

Объем банков памяти данных - до 128 байт (7Fh). В начале банка размещаются регистры специального назначения, затем регистры общего назначения, выполненные как статическое ОЗУ.

Таблица 2

| RP1:RP0 | Банк |
|---------|------|
| 00 | 0 |
| 01 | 1 |
| 10 | 2 |
| 11 | 3 |

Все реализованные банки содержат регистры специального назначения. Некоторые часто используемые регистры специального назначения могут отображаться и в других банках памяти.

1.3.1. Регистры общего назначения

Обратиться к регистрам общего назначения можно прямой или косвенной адресацией через регистр FSR.

Карта памяти данных микроконтроллера PIC16F877 показана в табл. 3.

1.3.2. Регистры специального назначения

С помощью регистров специального назначения выполняется управление функциями ядра и периферийными модулями микроконтроллера. Регистры специального назначения реализованы как статическое ОЗУ. Перечень специальных регистров и их составы приведены в табл. 4. Здесь использованы следующие условные обозначения:

- не используется, читается как '0';
- и - не изменяется;
- х - не известно;
- q - зависит от условий,
- г - резервные ячейки, читаются как '0'.

Таблица 3

| Регистр | Адрес | Регистр | Адрес | Регистр | Адрес | Регистр | Адрес |
|---------------------------------------------------|-------|---------------------------------------------------|------------|---------------------------------------------------|--------------|---------------------------------------------------|--------------|
| INDF | 00h | INDF | 80h | INDF | 100h | INDF | 180h |
| TMR0 | 01h | OPTION REG | 81h | TMR0 | 101h | OPTION REG | 181h |
| PCL | 02h | PCL | 82h | PCL | 102h | PCL | 182h |
| STATUS | 03h | STATUS | 83h | STATUS | 103h | STATUS | 183h |
| FSR | 04h | FSR | 84h | FSR | 104h | FSR | 184h |
| PORTA | 05h | TRISA | 85h | | 105h | | 185h |
| PORTB | 06h | TRISB | 86h | PORTB | 106h | TRISB | 186h |
| PORTC | 07h | TRISC | 87h | | 107h | | 187h |
| PORTD | 08h | TRISD | 88h | | 108h | | 188h |
| PORTED | 09h | TRISE | 89h | | 109h | | 189h |
| PCLATH | 0Ah | PCLATH | 8Ah | PCLATH | 10Ah | PCLATH | 18Ah |
| INTCON | 0Bh | INTCON | 8Bh | INTCON | 10Bh | INTCON | 18Bh |
| PIR1 | 0Ch | PIE1 | 8Ch | EEDATA | 10Ch | EECON1 | 18Ch |
| PIR2 | 0Dh | PIE2 | 8Dh | EEADR | 10Dh | EECON2 | 18Dh |
| TMR1L | 0Eh | PCON | 8Eh | EEDATH | 10Eh | Резерв | 18Eh |
| TMR1H | 0Fh | | 8Fh | EEADRH | 10Fh | Резерв | 18Fh |
| T1CON | 10h | | 90h | | 110h | | 190h |
| TMR2 | 11h | SSPCON2 | 91h | | 111h | | 191h |
| T2CON | 12h | PR2 | 92h | | 112h | | 192h |
| SSPBUF | 13h | SSPADDD | 93h | | 113h | | 193h |
| SSPCON | 14h | SSPSTAT | 94h | | 114h | | 194h |
| CCPR1L | 15h | | 95h | | 115h | | 195h |
| CCPR1H | 16h | | 96h | Регистры общего назначения | 116h | Регистры общего назначения | 196h |
| CCP1CON | 17h | | 97h | | 117h | | 197h |
| RCSTA | 18h | TXSTA | 98h | | 118h | | 198h |
| TXREG | 19h | SPBRG | 99h | | 119h | | 199h |
| RCREG | 1Ah | | 9Ah | 16 байтов | 11Ah | 16 байтов | 19Ah |
| CCPR2L | 1Bh | | 9Bh | | 11Bh | | 19Bh |
| CCPR2H | 1Ch | | 9Ch | | 11Ch | | 19Ch |
| CCP2CON | 1Dh | | 9Dh | | 11Dh | | 19Dh |
| ADRESH | 1Eh | ADRESL | 9Eh | | 11Eh | | 19Eh |
| ADCON0 | 1Fh | ADCON1 | 9Fh | | 11Fh | | 19Fh |
| | 20h | | A0h | | 120h | | 1A0h |
| Регистры общего назначения 96 байтов | | Регистры общего назначения 80 байтов | | Регистры общего назначения 80 байтов | | Регистры общего назначения 80 байтов | |
| | | Доступ к 70h-7Fh | EFh F0h | Доступ к 70h-7Fh | 16Fh 170h | Доступ к 70h-7Fh | 1EFh 1F0h |
| | 7Fh | | FFh | | 17Fh | | 1FFh |

Примечание. Регистр косвенной адресации INDF – не физический регистр. Обращение к нему по содержимому FSR= 00h дает нулевой результат.

Таблица 4

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Состояние после сброса POR, BOR |
|---------------|---------|----------------------------------------------------------------------------|---------|-----------------------------------------------------------------|---------------------------------|---------|----------|---------|---------|---------------------------------|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Банк 0 | | | | | | | | | | |
| 00h | INDF | Обращение к регистру, адрес которого записан в FSR (не физический регистр) | | | | | | | | 0000 0000 |
| 01h | TMR0 | Регистр таймера 0 | | | | | | | | xxxx xxxx |
| 02h | PCL | Младшие биты счетчика команд PC | | | | | | | | 0000 0000 |
| 03h | STATUS | IRP | RP1 | RP0 | -TO | -PD | Z | DC | C | 0001 1xxx |
| 04h | FSR | Регистр адреса при косвенной адресации | | | | | | | | xxxx xxxx |
| 05h | PORTA | - | - | Запись в выходную защелку PORTA, чтение состояния выводов PORTA | | | | | | -- 0x 0000 |
| 06h | PORTB | Запись в выходную защелку PORTB, чтение состояния выводов PORTB | | | | | | | | xxxx xxxx |
| 07h | PORTC | Запись в выходную защелку PORTC, чтение состояния выводов PORTC | | | | | | | | xxxx xxxx |
| 08h | PORTD | Запись в выходную защелку PORTD, чтение состояния выводов PORTD | | | | | | | | xxxx xxxx |
| 09h | PORTE | - | - | - | - | - | RE2 | RE1 | RE0 | ---- -xxx |
| 0Ah | PCLATH | - | - | - | Старшие биты счетчика команд PC | | | | | --- 0 0000 |
| 0Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x |
| 0Ch | PIR1 | PSPIF | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 |
| 0Dh | PIR2 | - | - | - | EEIF | BCLIF | - | - | CCP2IF | -r-0 0--0 |
| 0Eh | TMR1L | Младший байт 16-разрядного таймера 1 | | | | | | | | xxxx xxxx |
| 0Fh | TMR1H | Старший байт 16-разрядного таймера 1 | | | | | | | | xxxx xxxx |
| 10h | T1CON | - | - | T1CKPS1 | T1CKPS0 | T1OSCEN | T1SYNC | TMR1CS | TMR1ON | -- 00 0000 |
| 11h | TMR2 | Регистр таймера 2 | | | | | | | | 0000 0000 |
| 12h | T2CON | - | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 |
| 13h | SSPBUF | Буфер приемника MSSP / регистр передатчика | | | | | | | | xxxx xxxx |
| 14h | SSPCON | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 |
| 15h | CCPR1L | Младший байт захвата/сравнения/ШИМ CCP1 | | | | | | | | xxxx xxxx |
| 16h | CCPR1H | Старший байт захвата/сравнения/ШИМ CCP1 | | | | | | | | xxxx xxxx |
| 17h | CCP1CON | - | - | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | -- 00 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x |
| 19h | TXREG | Регистр данных передатчика USART | | | | | | | | 0000 0000 |
| 1Ah | RCREG | Регистр данных приемника USART | | | | | | | | 0000 0000 |
| 1Bh | CCPR2L | Младший байт захвата/сравнения/ШИМ CCP2 | | | | | | | | xxxx xxxx |
| 1Ch | CCPR2H | Старший байт захвата/сравнения/ШИМ CCP2 | | | | | | | | xxxx xxxx |
| 1Dh | CCP2CON | - | - | CCP2X | CCP2Y | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 | -- 00 0000 |
| 1Eh | ADRESH | Старший байт результата преобразования АЦП | | | | | | | | xxxx xxxx |
| 1Fh | ADCON0 | ADCS1 | ADCS0 | CHS2 | CHS1 | CHS0 | GO/-DONE | - | ADON | 0000 00-0 |

| Банк 1 | | | | | | | | | | |
|---------------|------------|-------------------------------------------------------------------------------|---------|---------------------------|---------------------------------|-------|--------------------------|--------|------------|---------------|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 80h | INDF | Обращение к регистру, адрес которого записан в FSR (не физический регистр) | | | | | | | | 0000 0000 |
| 81h | OPTION_REG | -RBPU | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 |
| 82h | PCL | Младшие биты счетчика команд PC | | | | | | | | 0000 0000 |
| 83h | STATUS | IRP | RP1 | RP0 | -TO | -PD | Z | DC | C | 0001 1xxx |
| 84h | FSR | Регистр адреса при косвенной адресации | | | | | | | | xxxx xxxx |
| 85h | TRISA | - | - | Направление выводов PORTA | | | | | | -- 11 1111 |
| 86h | TRISB | Направление выводов PORTB | | | | | | | | 1111 1111 |
| 87h | TRISC | Направление выводов PORTC | | | | | | | | 1111 1111 |
| 88h | TRISD | Направление выводов PORTD | | | | | | | | 1111 1111 |
| 89h | TRISE | IBF | OBF | IBOV | PSPMODE | - | Направление вывода PORTE | | | 0000 -111 |
| 8Ah | PCLATH | - | - | - | Старшие биты счетчика команд PC | | | | --- 0 0000 | |
| 8Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x |
| 8Ch | PIE1 | PSPIE | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 |
| 8Dh | PIE2 | - | - | - | EEIE | BCLIE | - | - | CCP2IE | - r- 0 0 -- 0 |
| 8Eh | PCON | - | - | - | - | - | - | -POR | -BOR | ---- -- qq |
| 8Fh | - | Не реализовано | | | | | | | | - |
| 90h | - | Не реализовано | | | | | | | | - |
| 91h | SSPCON2 | GCEN | ACKSTAT | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN | 0000 0000 |
| 92h | PR2 | Регистр периода таймера 2 | | | | | | | | 1111 1111 |
| 93h | SSPADDD | Регистр адреса / Регистр генератора скорости обмена | | | | | | | | 0000 0000 |
| 94h | SSPSTAT | SMP | CKE | D/-A | P | S | R/-W | UA | BF | 0000 0000 |
| 95h | - | Не реализовано | | | | | | | | - |
| 96h | - | Не реализовано | | | | | | | | - |
| 97h | - | Не реализовано | | | | | | | | - |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | - | BRGH | TRMT | TX9D | 0000 - 010 |
| 99h | SPBRG | Регистр генератора скорости USART | | | | | | | | 0000 0000 |
| 9Ah | - | Не реализовано | | | | | | | | - |
| 9Bh | - | Не реализовано | | | | | | | | - |
| 9Ch | - | Не реализовано | | | | | | | | - |
| 9Dh | - | Не реализовано | | | | | | | | - |
| 9Eh | ADRESL | Младший байт результата преобразования АЦП | | | | | | | | xxxx xxxx |
| 9Fh | ADCON1 | ADFM | - | - | - | PCFG3 | PCFG2 | PCFG1 | PCFG0 | 0 --- 0000 |
| Банк 2 | | | | | | | | | | |
| 100h | INDF | Обращение к регистру, адрес которого записан в FSR (не физический регистр) | | | | | | | | 0000 0000 |
| 101h | TMR0 | Регистр таймера 0 | | | | | | | | xxxx xxxx |
| 102h | PCL | Младшие биты счетчика команд PC | | | | | | | | 0000 0000 |
| 103h | STATUS | IRP | RP1 | RP0 | -TO | -PD | Z | DC | C | 0001 1xxx |
| 104h | FSR | Регистр адреса при косвенной адресации | | | | | | | | xxxx xxxx |
| 105h | - | Не реализовано | | | | | | | | - |
| 106h | PORTB | Запись в выходную защелку PORTB, чтение состояния выводов PORTB | | | | | | | | xxxx xxxx |
| 107h | - | Не реализовано | | | | | | | | - |
| 108h | - | Не реализовано | | | | | | | | - |

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---------------|------------|----------------------------------------------------------------------------|--------|------------------------------|---------------------------------|-------|------|------|-----------|------------|
| 109h | - | Не реализовано | | | | | | | | - |
| 10Ah | PCLATH | - | - | - | Старшие биты счетчика команд PC | | | | | --- 0 0000 |
| 10Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | TOIF | INTF | RBIF | 0000 000x |
| 10Ch | EEDATA | Регистр данных, младший байт | | | | | | | | xxxx xxxx |
| 10Dh | EEADR | Регистр адреса, младший байт | | | | | | | | xxxx xxxx |
| 10Eh | EEDATH | - | - | Регистр данных, старший байт | | | | | xxxx xxxx | |
| 10Fh | EEADRH | - | - | - | Регистр адреса, старший байт | | | | | xxxx xxxx |
| Банк 3 | | | | | | | | | | |
| 180h | INDF | Обращение к регистру, адрес которого записан в FSR (не физический регистр) | | | | | | | | 0000 0000 |
| 181h | OPTION_REG | -RBPU | INTEDG | T0C S | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 |
| 182h | PCL | Младшие биты счетчика команд PC | | | | | | | | 0000 0000 |
| 183h | STATUS | IRP | RP1 | RP0 | -TO | -PD | Z | DC | C | 0001 1xxx |
| 184h | FSR | Регистр адреса при косвенной адресации | | | | | | | | xxxx xxxx |
| 185h | - | Не реализовано | | | | | | | | - |
| 186h | TRISB | Направление выводов PORTB | | | | | | | | 1111 1111 |
| 187h | - | Не реализовано | | | | | | | | - |
| 188h | - | Не реализовано | | | | | | | | - |
| 189h | - | Не реализовано | | | | | | | | - |
| 18Ah | PCLATH | - | - | - | Старшие биты счетчика команд PC | | | | | --- 0 0000 |
| 18Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | TOIF | INTF | RBIF | 0000 000x |
| 18Ch | EECON1 | EESPD | - | - | - | WREER | WREN | WR | RD | x --- x000 |
| 18Dh | EECON2 | Регистр управления 2 (физически не реализован) | | | | | | | | ----- |
| 18Eh | - | Резерв | | | | | | | | - |
| 18Fh | - | Резерв | | | | | | | | - |

1.3.2.1. Регистр STATUS

В регистре STATUS содержатся флаги состояния АЛУ, флаги причины сброса микроконтроллера и биты управления банками памяти данных.

Регистр STATUS может быть адресован любой командой, как и любой другой регистр памяти данных. Если обращение к регистру STATUS выполняется командой, которая воздействует на флаги Z, DC и C, то изменение этих трех битов командой заблокировано. Эти биты сбрасываются или устанавливаются согласно логике ядра микроконтроллера. Команды изменения регистра STATUS также не воздействуют на биты -TO и -PD. Поэтому результат выполнения команды с регистром STATUS может отличаться от ожидаемого. Например, команда CLRFSR STATUS сбросит три старших бита и установит бит Z (состояние регистра STATUS после выполнения команды 000uuuuu, где u - неизменяемый бит).

При изменении битов регистра STATUS рекомендуется использовать команды, не влияющие на флаги ALU (SWAPF, MOVWF, BCF и BSF).

Флаги C и DC используются как биты заема и десятичного заема соответственно, например, при выполнении команд вычитания SUBLW и SUBWF.

Регистр STATUS доступен по адресам 03h, 83h, 103h или 183h.

Структура регистра отражена в табл. 5.

Таблица 5

| Номер бита | Имя бита | Доступ и состояние после сброса | Назначение |
|------------|-------------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | IRP | R/W-0 | Выбор банка при косвенной адресации: 1 банк 2, 3(100h-1FFh) 0 банк 0, 1 (000h - 0FFh) |
| 6, 5 | RP1, RP0 | R/W-0 | Выбор банка при непосредственной адресации: 11 банк 3 (180h-1FFh) 10 банк 2 (100h-17Fh) 01 банк 1 (080h - 0FFh) 00 банк 0 (000h - 07Fh) |
| 4 | -TO | R-1 | Флаг переполнения сторожевого таймера: 1 после POR или выполнения команд CLRWDT, SLEEP, 0 после переполнения WDT |
| 3 | -PD | R-1 | Флаг включения питания: 1 после POR или выполнения команды CLRWDT, 0 после выполнения команды SLEEP |
| 2 | Z | R/W-x | Флаг нулевого результата: 1 нулевой результат выполнения арифметической или логической операции, 0 ненулевой результат выполнения арифметической или логической операции |
| 1 | DC | R/W-x | Флаг десятичного переноса/заема (для команд ADDWF, ADDWL, SUBWF, SUBWL), заем имеет инверсное значение: 1 был перенос/заем из младшего полубайта, 0 не было переноса/заема из младшего полубайта |
| 0 | C | R/W-x | Флаг переноса/заема (для команд ADDWF, ADDWL, SUBWF, SUBWL): 1 был перенос/заем из старшего бита, 0 не было переноса/заема из старшего бита. Заем имеет инверсное значение. Вычитание выполняется путем прибавления дополнительного кода второго операнда. При выполнении команд сдвига (RRF, RLF) бит C загружается старшим или младшим битом сдвигаемого регистра |

1.3.2.2. Регистр OPTION_REG

Регистр OPTION доступен для чтения и записи, содержит биты управления предварительным делителем TMR0/WDT, активным фронтом внешнего прерывания RB0/INT, подтягивающими резисторами на входах PORTB.

Если предварительный делитель включен перед WDT, то коэффициент деления тактового сигнала для TMR0 равен 1:1.

Регистр OPTION_REG доступен по адресам 81h и 181h.

Структура регистра отражена в табл. 6.

Таблица 6

| Номер бита | Имя бита | Доступ и состояние после сброса | Назначение | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------|-------------------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|----------|---------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|-----|-----|------|------|-----|------|------|-----|-------|------|-----|-------|-------|
| 7 | -RBPU | R/W-1 | Включение подтягивающих резисторов на входах PORTB: 1 подтягивающие резисторы отключены, 0 подтягивающие резисторы включены | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | INTEDG | R/W-1 | Выбор активного фронта сигнала на входе внешнего прерывания INT: 1 прерывания по фронту сигнала, 0 прерывания по срезу сигнала | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | TOCS | R/W-1 | Выбор тактового сигнала для TMR0: 1 внешний тактовый сигнал с вывода RA4/TOCKI, 0 внутренний тактовый сигнал CLKOUT | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | TOSE | R/W-1 | Выбор фронта приращения TMR0 при внешнем тактовом сигнале: 1 приращение по срезу сигнала (с высокого к низкому уровню) на выводе RA4/TOCKI, 0 приращение по фронту сигнала (с низкого к высокому уровню) на выводе RA4/TOCKI | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | PSA | R/W-1 | Выбор включения предделителя: 1 предделитель включен перед WDT, 0 предделитель включен перед TMR0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 1 0 | PS2 PS1 PS0 | R/W-1 | Установка коэффициента деления предделителя: <table style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th></th> <th>для TMR0</th> <th>для WDT</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>1:2</td> <td>1:1</td> </tr> <tr> <td>001</td> <td>1:4</td> <td>1:2</td> </tr> <tr> <td>010</td> <td>1:8</td> <td>1:4</td> </tr> <tr> <td>011</td> <td>1:16</td> <td>1:8</td> </tr> <tr> <td>100</td> <td>1:32</td> <td>1:16</td> </tr> <tr> <td>101</td> <td>1:64</td> <td>1:32</td> </tr> <tr> <td>110</td> <td>1:128</td> <td>1:64</td> </tr> <tr> <td>111</td> <td>1:256</td> <td>1:128</td> </tr> </tbody> </table> | | для TMR0 | для WDT | 000 | 1:2 | 1:1 | 001 | 1:4 | 1:2 | 010 | 1:8 | 1:4 | 011 | 1:16 | 1:8 | 100 | 1:32 | 1:16 | 101 | 1:64 | 1:32 | 110 | 1:128 | 1:64 | 111 | 1:256 | 1:128 |
| | для TMR0 | для WDT | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 000 | 1:2 | 1:1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 001 | 1:4 | 1:2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 010 | 1:8 | 1:4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 011 | 1:16 | 1:8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 100 | 1:32 | 1:16 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 101 | 1:64 | 1:32 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 110 | 1:128 | 1:64 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 111 | 1:256 | 1:128 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

1.3.2.3. Регистр INTCON

Регистр INTCON доступен для чтения и записи, содержит биты разрешений и флаги прерываний по переполнению TMR0, по изменению уровня сигнала на выводах PORTB и по внешнему источнику прерываний RB0/INT. Флаги прерываний устанавливаются при возникновении условий прерываний вне зависимости от соответствующих битов разрешения и бита общего разрешения прерываний GIE (INTCON<7>).

Регистр INTCON доступен по адресам 0Bh, 1Bh, 10Bh и 18Bh.

Структура регистра отражена в табл. 7.

Таблица 7

| <i>Номер бита</i> | <i>Имя бита</i> | <i>Доступ и состояние после сброса</i> | <i>Назначение</i> |
|-------------------|-----------------|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | GIE | R/W-0 | Глобальное разрешение прерываний: 1 разрешены все немаскированные прерывания, 0 все прерывания запрещены |
| 6 | PEIE | R/W-0 | Разрешение прерываний от периферийных модулей: 1 разрешены все немаскированные прерывания периферийных модулей, 0 прерывания от периферийных модулей запрещены |
| 5 | TOIE | R/W-0 | Разрешение прерывания по переполнению TMR0: 1 прерывание разрешено, 0 прерывание запрещено |
| 4 | INTE | R/W-0 | Разрешение внешнего прерывания INT: 1 прерывание разрешено, 0 прерывание запрещено |
| 3 | RBIE | R/W-0 | Разрешение прерывания по изменению сигнала на входах RB7:RB4 PORTB: 1 прерывание разрешено, 0 прерывание запрещено |
| 2 | TOIF | R/W-0 | Флаг прерывания по переполнению TMR0: 1 произошло переполнение TMR0 (сбрасывается программно), 0 переполнения TMR0 не было |
| 1 | INTF | R/W-0 | Флаг внешнего прерывания INT: 1 выполнено условие внешнего прерывания на выводе RB0/INT (сбрасывается программно), 0 внешнего прерывания не было |
| 0 | RBIF | R/W-x | Флаг прерывания по изменению уровня сигнала на входах RB7:RB4 PORTB: 1 зафиксировано изменение уровня сигнала на одном из входов (сбрасывается программно), 0 не было изменения уровня сигнала |

1.3.2.4. Регистр PIE1

Регистр PIE1 доступен для чтения и записи, содержит биты разрешения периферийных прерываний.

Чтобы разрешить периферийные прерывания, необходимо установить в '1' бит PEIE(INTCON<6>).

Регистр PIE1 доступен по адресу 8Ch.

Структура регистра отражена в табл. 8.

Таблица 8

| Номер бита | Имя бита | Доступ и состояние после сброса | Назначение |
|------------|----------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| 7 | PSPIE | R/W-0 | Разрешение прерывания записи/чтения ведомого параллельного порта: 1 прерывание разрешено, 0 прерывание запрещено |
| 6 | ADIE | R/W-0 | Разрешение прерывания по окончании преобразования АЦП: 1 прерывание разрешено, 0 прерывание запрещено |
| 5 | RCIE | R/W-0 | Разрешение прерывания от приемника USART: 1 прерывание разрешено, 0 прерывание запрещено. |
| 4 | TXIE | R/W-0 | Разрешение прерывания от передатчика USART: 1 прерывание разрешено, 0 прерывание запрещено |
| 3 | SSPIE | R/W-0 | Разрешение прерывания от модуля синхронного последовательного порта: 1 прерывание разрешено, 0 прерывание запрещено |
| 2 | CCP1IE | R/W-0 | Разрешение прерывания от модуля CCP1: 1 прерывание разрешено, 0 прерывание запрещено |
| 1 | TMR2IE | R/W-0 | Разрешение прерывания по переполнению TMR2: 1 прерывание разрешено, 0 прерывание запрещено |
| 0 | TMR1IE | R/W-0 | Разрешение прерывания по переполнению TMR1: 1 прерывание разрешено, 0 прерывание запрещено |

1.3.2.5. Регистр PIR1

Регистр PIR1 доступен для чтения и записи, содержит флаги прерываний периферийных модулей.

Флаги прерываний устанавливаются при возникновении условий прерываний вне зависимости от соответствующих битов разрешения и бита общего разрешения прерываний GIE (INTCON<7>). Программное обеспечение пользователя должно сбрасывать соответствующие флаги при обработке прерываний от периферийных модулей.

Регистр PIR1 доступен по адресу 0Ch.

Структура регистра отражена в табл. 9.

1.3.2.6. Регистр PIE2

Регистр PIE2 доступен для чтения и записи, содержит биты разрешения прерываний от модуля CCP2, возникновения коллизий на шине и окончания записи в EEPROM-память данных.

Регистр PIE2 доступен по адресу 8Dh.

Структура регистра отражена в табл. 10.

1.3.2.7. Регистр PIR2

Регистр PIR2 доступен для чтения и записи, содержит флаги прерываний от модуля CCP2, возникновения коллизий на шине и окончания записи в EEPROM-память данных.

Флаги прерываний устанавливаются при возникновении условий прерываний вне зависимости от соответствующих битов разрешения и бита общего разрешения прерываний GIE (INTCON<7>). Программное обеспечение пользователя должно сбрасывать соответствующие флаги при обработке прерываний от периферийных модулей.

Регистр PIR2 доступен по адресу 0Dh.

Структура регистра отражена в табл. 11.

Таблица 9

| Номер бита | Имя бита | Доступ и состояние после сброса | Назначение |
|------------|----------|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | PSPIF | R/W-0 | Флаг прерывания от ведомого параллельного порта: 1 произошла операция чтения или записи (сбрасывается программно), 0 операции чтения или записи не происходило |
| 6 | ADIF | R/W-0 | Флаг прерывания от модуля АЦП: 1 преобразование АЦП завершено, 0 преобразование АЦП не завершено |
| 5 | RCIF | R/W-0 | Флаг прерывания от приемника USART: 1 буфер приемника USART полон, 0 буфер приемника USART пуст |
| 4 | TXIF | R/W-0 | Флаг прерывания от передатчика USART: 1 буфер передатчика USART пуст, 0 буфер передатчика USART полон |
| 3 | SSPIF | R/W-0 | Флаг прерываний от модуля MSSP: 1 выполнено условие возникновения прерывания от модуля MSSP прием/передача данных (сбрасывается программно). 0 условие прерывания от модуля MSSP не выполнено |
| 2 | CCP1IF | R/W-0 | Флаг прерывания от модуля CCP1: <u>Режим захвата</u> 1 выполнен захват значения TMR1 (сбрасывается программно), 0 захвата значения TMR1 не происходило. <u>Режим сравнения</u> 1 значение TMR1 достигло указанного в регистрах CCP1H:CCP1L (сбрасывается программно), 0 значение TMR1 не достигло указанного в регистрах CCP1H:CCP1L. <u>ШИМ-режим.</u> Не используется |
| 1 | TMR2IF | R/W-0 | Флаг прерывания по переполнению TMR2: 1 произошло переполнение TMR2 (сбрасывается программно), 0 переполнения TMR2 не было |
| 0 | TMR1IF | R/W-0 | Флаг прерывания по переполнению TMR1: 1 произошло переполнение TMR1 (сбрасывается программно), 0 переполнения TMR1 не было |

Таблица 10

| Номер бита | Имя бита | Доступ и состояние после сброса | Назначение |
|------------|----------|---------------------------------|-----------------------------------------------------------------------------------------------------------|
| 7 | - | 0 | Не реализован: читается как '0' |
| 6 | - | R/W-0 | Резерв: всегда должен равняться нулю |
| 5 | - | 0 | Не реализован: читается как '0' |
| 4 | EEIE | R/W-0 | Разрешение прерывания по окончании записи в EEPROM данных: 1 прерывание разрешено, 0 прерывание запрещено |
| 3 | BCLIE | R/W-0 | Разрешение прерывания при возникновении коллизий на шине: 1 прерывание разрешено, 0 прерывание запрещено |
| 2 | - | 0 | Не реализован: читается как '0' |
| 1 | - | 0 | Не реализован: читается как '0' |
| 0 | CCP2IE | R/W-0 | Разрешение прерывания от модуля CCP2: 1 прерывание разрешено, 0 прерывание запрещено |

Таблица 11

| Номер бита | Имя бита | Доступ и состояние после сброса | Назначение |
|------------|----------|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | - | 0 | Не реализован: читается как '0' |
| 6 | - | R/W-0 | Резерв: всегда должен равняться нулю |
| 5 | - | 0 | Не реализован: читается как '0' |
| 4 | EEIF | R/W-0 | Флаг прерывания по окончании записи в EEPROM данных: 1- запись в EEPROM завершена (сбрасывается программно), 0 - запись в EEPROM не завершена или не начата |
| 3 | BCLIF | R/W-0 | Флаг прерывания возникновения коллизий на шине: 1 - на шине обнаружены коллизии (только в режиме ведущего I ² C), 0 - коллизий не обнаружено |
| 2 | - | 0 | Не реализован: читается как '0' |
| 1 | - | 0 | Не реализован: читается как '0' |
| 0 | CCP2IF | R/W-0 | Флаг прерывания от модуля CCP2: <u>Режим захвата</u> 1 - выполнен захват значения TMR1 (сбрасывается программно), 0 - захвата значения TMR1 не происходило. <u>Режим сравнения</u> 1 - значение TMR1 достигло указанного в регистрах CCP2H:CCP2L (сбрасывается программно), 0 - значение TMR1 не достигло указанного в регистрах CCP2H:CCP2L. <u>ШИМ-режим</u> не используется |

1.3.3. Косвенная адресация, регистры INDF и FSR

Для выполнения косвенной адресации необходимо обратиться к физически не реализованному регистру INDF. Обращение к регистру INDF фактически вызовет действие с регистром, адрес которого указан в FSR. Косвенное чтение регистра INDF (FSR=0) даст результат 00h. Косвенная запись в регистр INDF вызывает только воздействия на флаги АЛУ в регистре STATUS. 9-й бит косвенного адреса IRP сохраняется в регистре STATUS<7>. Пример 9-разрядной косвенной адресации показан на рис. 3.

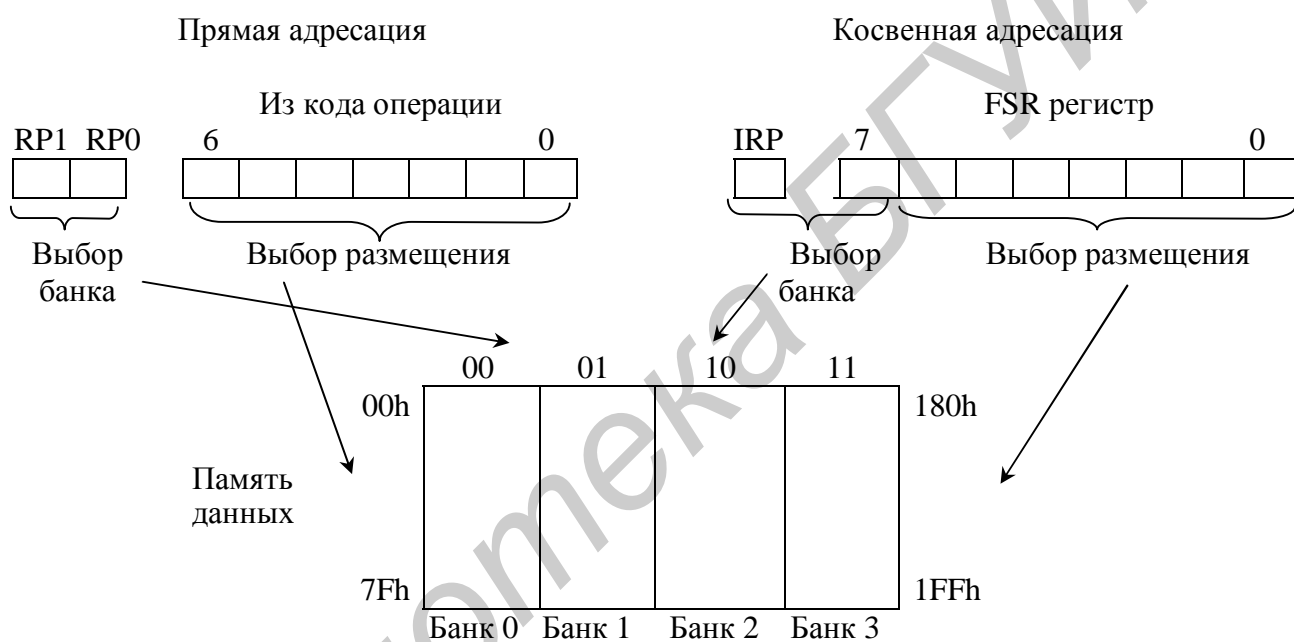


Рис. 3. Схема прямой и косвенной адресации в PIC16F877

В примере ниже показано использование косвенной адресации для очистки памяти данных в диапазоне адресов 20h-2Fh.

```
BCF      STATUS, IRP ; Установить банк 0,1
MOVLW   0x20        ; Указать первый регистр в ОЗУ
MOVWF   FSR
NEXT:
  CLRF   INDF       ; Очистить регистр
  INCF   FSR, F     ; Увеличить адрес
  BTFSS FSR, 4     ; Завершить?
  GOTO  NEXT        ; Нет, продолжить очистку
CONTINUE:
  ; Да
```

1.4. Прерывания

PIC16F877X имеет 14 источников прерываний. Регистр INTCON содержит флаги отдельных прерываний, биты разрешения этих прерываний и бит глобального разрешения прерываний.

Если бит GIE (INTCON<7>) установлен в '1', то разрешены все немаскированные прерывания. Если GIE=0, то все прерывания запрещены. Каждое прерывание в отдельности может быть разрешено или запрещено установкой/сбросом соответствующего бита в регистрах INTCON, PIE1 и PIE2. При сбросе микроконтроллера бит GIE сбрасывается в '0'.

При возвращении из подпрограммы обработки прерывания, по команде RETFIE бит GIE аппаратно устанавливается в '1', разрешая все немаскированные прерывания.

В регистре INTCON находятся флаги следующих прерываний: внешнего сигнала INT, изменения уровня сигнала на входах RB7:RB4, переполнения TMR0.

В регистрах PIR1, PIR2 содержатся флаги прерываний периферийных модулей микроконтроллера, а в регистрах PIE1, PIE2 – соответствующие биты разрешения прерываний. В регистре INTCON находится бит разрешения прерываний от периферийных модулей. Структурная схема логики прерываний показана на рис. 4.

При переходе на подпрограмму обработки прерываний бит GIE аппаратно сбрасывается в '0', запрещая прерывания, адрес возврата из подпрограммы обработки прерываний помещается в стек, а в счетчик команд PC загружается вектор прерывания 0004h. Источник прерываний может быть определен проверкой флагов прерываний, которые должны быть сброшены программно перед разрешением прерываний, чтобы избежать повторного вызова.

Для внешних источников прерываний (сигнал INT, изменения уровня сигнала на входах RB7:RB4) время перехода на подпрограмму обработки прерываний будет составлять 3-4 машинных цикла. Точное время перехода

зависит от конкретного случая, оно одинаково для 1- и 2- цикловых команд. Флаги прерываний устанавливаются независимо от состояния соответствующих битов маски и бита GIE.

Индивидуальные флаги прерываний устанавливаются независимо от состояния соответствующих битов маски и бита GIE.

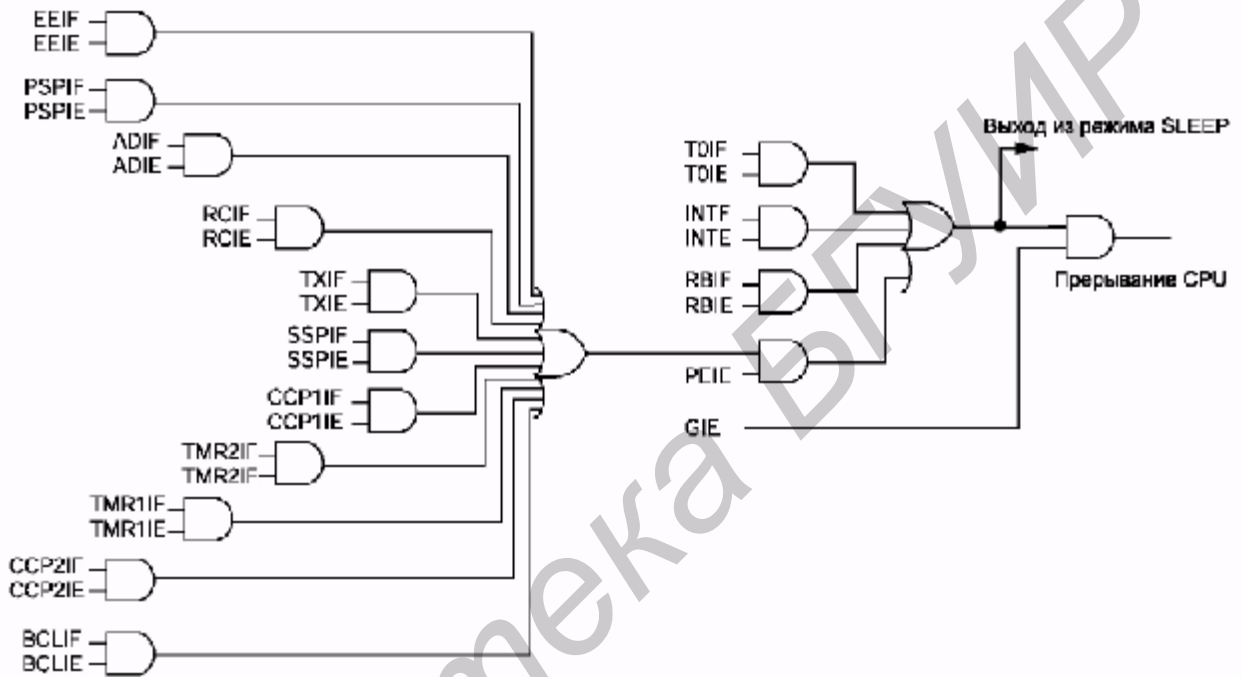


Рис. 4. Структурная схема логики прерываний

1.4.1. Внешнее прерывание с входа RB0/INT

Внешнее прерывание с входа RB0/INT происходит: по переднему фронту сигнала, если бит INTEDG (OPTION_REG<6>) установлен в '1'; по заднему фронту сигнала, если бит INTEDG сброшен в '0'. Когда активный фронт сигнала появляется на входе RB0/INT, бит INTF (INTCON<1>) устанавливается в '1'. Прерывание может быть запрещено сбросом бита INTE (INTCON<4>) в '0'. Флаг прерывания INTF должен быть сброшен программно в подпрограмме обработки прерываний. Прерывание INT может вывести микроконтроллер из режима SLEEP, если бит INTE=1 до перехода в режим SLEEP. Состояние бита

GIE определяет, переходить ли на подпрограмму обработки прерываний после выхода из режима SLEEP.

1.4.2. Прерывание по переполнению TMR0

Переполнение таймера TMR0 (FFh -> 00h) устанавливает флаг T0IF (INTCON<2>) в '1'. Прерывание от TMR0 можно разрешить/запретить установкой/сбросом бита T0IE(INTCON<5>).

1.4.3. Прерывание по изменению уровня сигнала на входах RB7:RB4

Изменение уровня сигнала на входах RB7:RB4 вызывает установку флага RBIF(INTCON<0>). Прерывание можно разрешить/запретить установкой/сбросом бита RBIE(INTCON<4>).

1.4.4. Сохранение контекста при обработке прерываний

При переходе на подпрограмму обработки прерываний в стеке сохраняется только адрес возврата. Как правило, необходимо сохранять значения ключевых регистров при обработке прерываний (например, регистр W и STATUS), что выполняется программным способом.

Так как старшие 16 байтов каждого банка микроконтроллера PIC16F877 доступны во всех банках, то регистры STATUS_TEMP, PCLATH_TEMP и W_TEMP могут быть размещены в этой области. Ниже показан пример текста программы сохранения контекста.

Пример: Сохранение и восстановление регистров STATUS, W и PCLATH

```
MOVWF    W_TEMP          ;Сохранить W в регистре текущего банка
SWAPF   STATUS,W        ;Поменять местами полубайты и сохранить в W
CLRF    STATUS          ;Выбрать банк 0
MOVWF   STATUS_TEMP     ;Сохранить регистр STATUS
MOVF    PCLATH,W        ;
MOVWF   PCLATH_TEMP    ;Сохранить регистр PCLATH
...
...                    ;Код программы обработки прерываний
...
MOVF    PCLATH_TEMP,W  ;
MOVWF   PCLATH         ;Восстановить регистр PCLATH
```

| | | |
|-------|---------------|------------------------------------------------------------------------------|
| SWAPF | STATUS_TEMP,W | ;Прочитать регистр STATUS_TEMP ; в W, восстанавливая банк памяти программ |
| MOVWF | STATUS | ;Переписать W в регистр STATUS |
| SWAPF | W_TEMP,F | ;Поменять местами полубайты в W_TEMP |
| SWAPF | W_TEMP,W | ;Поменять местами полубайты в W_TEMP и ;записать в W |

В примере для пересылки содержимого регистра в рабочий регистр W используется команда SWAPF. Это единственный способ пересылки без искажения состояния регистра STATUS.

1.5. Система команд

Каждая команда микроконтроллера PIC16F87X состоит из одного 14-разрядного слова, содержащего код операции (OPCODE), определяющий тип команды, и один или несколько операндов, указывающих операцию команды. Описание полей команд дается в табл. 12. Полный список команд приведен в табл. 13. Команды разделены на следующие группы: байт-ориентированные команды, бит-ориентированные команды, команды управления и операций с константами.

Таблица 12

| <i>Поле</i> | <i>Описание</i> |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| f | Адрес регистра (от 0x00 до 0x7F) |
| w | Рабочий регистр (аккумулятор) |
| b | Номер бита в 8-разрядном регистре |
| k | Константа (данные или метка) |
| d | Указатель адресата результата операции: d = 0 - результат сохраняется в регистре w, d = 1 - результат сохраняется в регистре f, По умолчанию d = 1 |
| C | Флаг переполнения или заема |
| Z | Флаг нулевого результата |
| DC | Флаг десятичного переноса (из 3-го в 4-й разряд) |
| PC | Счетчик команд |
| GIE | Бит глобального разрешения прерываний |
| WDT | Сторожевой таймер |
| -TO | Флаг переполнения WDT |
| -PD | Флаг сброса по включению питания |

Для байт-ориентированных команд f является указателем регистра, а d — указателем адреса результата. Указатель регистра определяет, какой регистр должен использоваться в команде. Указатель адресата определяет, где будет сохранен результат. Если $d=0$, результат сохраняется в регистре W . Если $d=1$, результат сохраняется в регистре, который используется в команде.

В бит-ориентированных командах b определяет номер бита, участвующего в операции, а f — указатель регистра, который содержит этот бит.

В командах управления или операциях с константами k представляет восемь или одиннадцать битов константных значений или значения литералов.

Все команды выполняются за один машинный цикл, кроме команд условия, в которых получен истинный результат, и инструкций, изменяющих значение счетчика команд PC. В случае выполнения команды за два машинных цикла во втором цикле выполняется инструкция NOP. Один машинный цикл состоит из четырех тактов генератора. Для тактового генератора с частотой 4 МГц все команды выполняются за 1 мкс; если условие истинно или изменяется счетчик команд PC, команда выполняется за 2 мкс.

Таблица 13

| Мнемоника команды | Описание | Кол-во циклов | Изм. флаги | Прим. |
|-------------------------------------|--------------------------------------------------------------------|---------------|------------|-------|
| 1 | 2 | 3 | 4 | 5 |
| <i>Байт-ориентированные команды</i> | | | | |
| ADDWF f,d | Сложение W и f | 1 | C,DC,Z | 1,2 |
| ANDWF f,d | Побитное 'И' W и f | 1 | Z | 1,2 |
| CLRF f | Очистить f | 1 | Z | 2 |
| CLRW | Очистить W | 1 | Z | |
| COMF f,d | Инвертировать f | 1 | Z | 1,2 |
| DECF f,d | Вычесть 1 из f | 1 | Z | 1,2 |
| DECFSZ f,d | Вычесть 1 из f и пропустить следующую команду, если результат 0 | 1(2) | | 1,2,3 |
| INCF f,d | Прибавить 1 к f | 1 | Z | 1,2 |
| INCFSZ f,d | Прибавить 1 к f и пропустить следующую команду, если результат 0 | 1(2) | | 1,2,3 |
| IORWF f,d | Побитное 'ИЛИ' W и f | 1 | Z | 1,2 |
| MOVF f,d | Переслать f | 1 | Z | 1,2 |
| MOVWF f | Переслать W в f | 1 | | |
| NOP | Нет операции | 1 | | |

| 1 | 2 | 3 | 4 | 5 |
|----------------------------------------------------|------------------------------------------------------------------------------|------|---------|-----|
| RLF f,d | Циклический сдвиг f влево через перенос | 1 | C | 1,2 |
| RRF f,d | Циклический сдвиг f вправо через перенос | 1 | C | 1,2 |
| SUBWF f,d | Вычесть W из f | 1 | C,DC,Z | 1,2 |
| SWAPF f,d | Поменять местами полубайты в регистре f | 1 | | 1,2 |
| XORWF f,d | Побитное 'исключающее ИЛИ' W и f | 1 | Z | 1,2 |
| <i>Бит-ориентированные команды</i> | | | | |
| BCF f,b | Очистить бит b в регистре f | 1 | | 1,2 |
| BSF f,b | Установить бит b в регистре f | 1 | | 1,2 |
| BTFSC f,b | Проверить бит b в регистре f, пропустить следующую команду, если результат 0 | 1(2) | | 3 |
| BTFSS f,b | Проверить бит b в регистре f, пропустить следующую команду, если результат 1 | 1(2) | | 3 |
| <i>Команды управления и операций с константами</i> | | | | |
| ADDLW k | Сложить константу с W | 1 | C,DC,Z | |
| ANDLW k | Побитное 'И' константы и W | 1 | Z | |
| CALL k | Вызов подпрограммы | 2 | | |
| CLRWDT | Очистить WDT | 1 | -TO,-PD | |
| GOTO k | Безусловный переход | 2 | | |
| IORLW k | Побитное 'ИЛИ' константы и W | 1 | Z | |
| MOVLW k | Переслать константу в W | 1 | | |
| RETFIE | Возврат из подпрограммы с разрешением прерываний | 2 | | |
| RETLW k | Возврат из подпрограммы с загрузкой константы k в W | 2 | | |
| RETURN | Возврат из подпрограммы | 2 | | |
| SLEEP | Перейти в режим SLEEP | 1 | -TO,-PD | |
| SUBLW k | Вычесть W из константы k | 1 | C,DC,Z | |
| XORLW k | Побитное 'исключающее ИЛИ' константы и W | 1 | Z | |

Примечания:

1. При выполнении операции «чтение - модификация – запись» с портом ввода/вывода исходные значения считываются с выводов порта, а не из выходных защелок. Например, если в выходной защелке была записана '1', а на соответствующем выходе низкий уровень сигнала, то обратно будет записано значение '0'.

2. При выполнении записи в TMR0 (и d=1) предделитель TMR0 сбрасывается, если он подключен к модулю TMR0.

3. Если условие истинно или изменяется значение счетчика команд PC, то инструкция выполняется за два цикла. Во втором цикле выполняется команда NOP.

2. ЛАБОРАТОРНАЯ РАБОТА

«Последовательный синхронный ввод/вывод данных»

Цель работы:

- 1) изучение принципов организации последовательного синхронного обмена данными в микропроцессорных системах;
- 2) приобретение навыков программирования процедур последовательного синхронного обмена данными.

2.1. Структурная схема аппаратной части

В ОМК PIC16F877 последовательный синхронный ввод/вывод данных производится с помощью встроенного модуля MSSP — ведущего последовательного синхронного порта (рис. 5). Он может использоваться для связи с периферийными микросхемами или другими микроконтроллерами.

Периферийными микросхемами могут быть: EEPROM-память, сдвиговые регистры, драйверы ЖКИ, АЦП и др.

Модуль MSSP может работать в одном из двух режимов: SPI — последовательный периферийный интерфейс и I²C — Inter-Integrated Circuit.

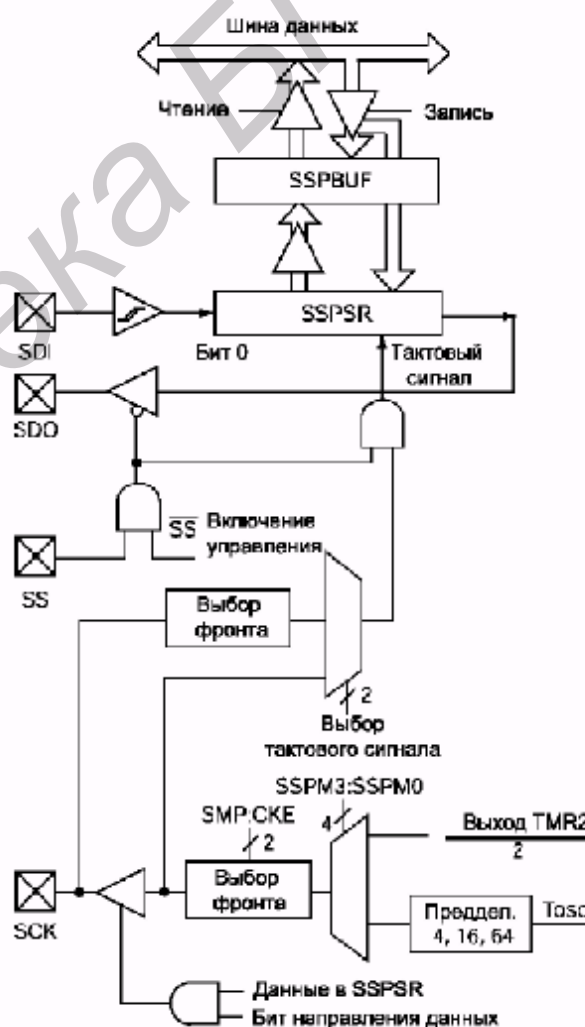


Рис. 5. Структурная схема модуля MSSP в SPI

В рамках данной лабораторной работы ограничимся изучением только первого режима.

2.2. Управляющие регистры

2.2.1. Регистр SSPSTAT статуса модуля MSSP

Регистр SSPSTAT доступен для чтения и записи, содержит флаги настройки режимов работы, а также состояния модуля MSSP.

Регистр SSPSTAT доступен по адресу 94h.

Структура регистра отражена в табл. 14.

Таблица 14

| Номер бита | Имя бита | Доступ и состояние после сброса | Назначение |
|------------|----------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | SMP | R/W-0 | Фаза выборки бита: <u>Ведущий режим SPI</u> 1 – опрос входа в конце периода вывода данных 0 – опрос входа в середине периода вывода данных <u>Ведомый режим SPI</u> Для режима ведомого SPI этот бит всегда должен быть сброшен в «0» |
| 6 | SCKE | R/W-0 | Выбор фронта тактового сигнала: <u>SPI режим. CKP=0</u> 1 – данные передаются по переднему фронту сигнала на выводе SCK 0 – данные передаются по заднему фронту сигнала на выводе SCK <u>SPI режим. CKP=1</u> 1 – данные передаются по заднему фронту сигнала на выводе SCK 0 – данные передаются по переднему фронту сигнала на выводе SCK |
| 5 | D/A | R-0 | Бит Данные/Адрес (только для режима I ² C) |
| 4 | P | R-0 | Бит STOP (только для режима I ² C) |
| 3 | S | R-0 | Бит START (только для режима I ² C) |
| 2 | R/W | R-0 | Бит чтения/записи (только для режима I ² C) |
| 1 | UA | R-0 | Флаг обновления адреса устройства (только для режима 10-разрядного I ² C) |
| 0 | BF | R-0 | Бит статуса буфера: <u>Прием (SPI и I²C режимы)</u> 1 – прием завершен, буфер SSPBUF полон 0 – прием не завершен, буфер SSPBUF пуст <u>Передача (только для I²C режима)</u> |

2.2.2. Регистр SSPCON управления модуля MSSP

Регистр SSPCON доступен для чтения и записи, содержит флаги настройки режимов работы MSSP. Регистр SSPSTAT доступен по адресу 14h.

Структура регистра отражена в табл. 15.

Таблица 15

| Номер бита | Имя бита | Доступ и состояние после сброса | Назначение |
|------------------|----------------------------------|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | WCOL | R/W-0 | Бит конфликта записи (только для режима I ² C) |
| 6 | SSPOV | R/W-0 | Бит переполнения приемника: <u>SPI режим</u> 1 – принят новый байт, а SSPBUF содержит предыдущие данные (байт в SSPSR будет потерян). В ведомом режиме пользователь должен прочитать содержимое регистра SSPBUF, даже если только передает данные. В ведущем режиме бит в '1' не устанавливается, т.к. каждая операция инициализируется записью в SSPBUF (сбрасывается в '0' программно), 0 – нет переполнения |
| 5 | SSPEN | R/W-0 | Бит включения модуля MSSP. Когда модуль включен, соответствующие порты ввода/вывода настраиваются на выход или вход. <u>SPI режим:</u> 1 – модуль MSSP включен, выходы SCK, SDO, SDI, -SS используются модулем MSSP, 0 – модуль MSSP выключен, выходы работают как цифровые порты ввода/вывода |
| 4 | CKP | R/W-0 | Бит выбора полярности тактового сигнала. <u>SPI режим:</u> 1 – пассивный высокий уровень сигнала, 0 - пассивный низкий уровень сигнала |
| 3 2 1 0 | SSPM3 SSPM2 SSPM1 SSPM0 | R/W-0 R/W-0 R/W-0 R/W-0 | Режим работы модуля MSSP: 0000 – ведущий режим SPI, тактовый сигнал = Fosc/4, 0001 – ведущий режим SPI, тактовый сигнал = Fosc/16, 0010 – ведущий режим SPI, тактовый сигнал = Fosc/64, 0011 – ведущий режим SPI, тактовый сигнал = выход TMR2 / 2, 0100 – ведомый режим SPI, тактовый сигнал с вывода SCK. Вывод -SS подключен к MSSP, 0101 – ведомый режим SPI, тактовый сигнал с вывода SCK. Вывод -SS не подключен к MSSP |

В SPI-режиме возможен одновременный синхронный прием и передача 8-разрядных данных. Модуль SSP поддерживает четыре режима SPI с типовым использованием трех выводов микроконтроллера:

- вход последовательных данных (SDI);
- выход последовательных данных (SDO);
- тактовый сигнал (SCK).

Дополнительно может быть задействован четвертый вывод для работы в режиме ведомого:

- выбор ведомого (-SS).

При инициализации SPI необходимо определить параметры работы модуля SPI битами SSPCON<5:0>, SSPSTAT<7:6>. Управляющие биты определяют следующие параметры работы:

- ведущий режим (SCK-выход);
- ведомый режим (SCK-вход);
- полярность тактового сигнала (пассивный уровень SCK);
- фаза выборки входных данных (в середине или в конце передачи бита);
- активный фронт тактового сигнала (передний, задний);
- частота тактового сигнала (только в ведущем режиме);
- режим выбора ведомого (только в режиме ведомого).

Модуль MSSP состоит из приемного/передающего регистра сдвига (SSPSR) и буферного регистра (SSBUF). В регистре SSPSR выполняется сдвиг данных из микроконтроллера или в микроконтроллер старшим битом вперед. В регистре SSPBUF сохраняются записанные данные, пока не будут получены новые. После приема 8 бит данных в регистр SSPSR они переписываются в SSPBUF, устанавливается в '1' флаг полного приемного буфера BF (SSPSTAT<0>) и флаг прерывания SSPIF. Двойная буферизация принимаемых данных позволяет принимать следующий байт до чтения предыдущего. Любая запись в регистр SSPBUF во время выполнения операции приема/передачи данных будет игнорироваться, при этом устанавливается в '1' флаг WCOL

(SSPCON<7>). Пользователь должен программно сбросить бит WCOL в '0', чтобы была возможность проверки выполнения записи в регистр SSPBUF. При приеме данных в режиме SPI регистр SSPBUF должен быть прочитан до момента окончания приема следующего байта. Бит статуса приемного буфера BF (SSPSTAT<0>) указывает на получение нового байта данных. Бит BF аппаратно сбрасывается в '0' при чтении регистра SSPBUF. Принятые данные могут быть недостоверными, если режим SPI используется только для передачи данных. Прерывания от модуля MSSP используются для определения завершения приема/передачи данных (в подпрограмме обработки прерываний необходимо прочитать/записать регистр SSPBUF). Если не планируется использовать прерывания от модуля MSSP, то необходимо предусмотреть программную проверку выполнения записи в регистр SSPBUF для передачи данных.

Ниже приведен пример загрузки данных в регистр SSPBUF (SSPSR) для передачи данных. Команда, отмеченная меткой aa, требуется только тогда, когда принимаемые данные имеют какое-то значение (в некоторых приложениях модуль MSSP в режиме SPI используется только для передачи данных).

Пример: Загрузка данных в регистр SSPBUF (SSPSR)

```

        BCF      STATUS, RPI      ;Банк 1
        BSF      STATUS, RP0
LOOP
        BTFSS   SSPSTAT, BF      ;Данные приняты?
        GOTO    LOOP            ;Нет
        BCF      STATUS, RP0     ;Банк 0
        MOVF    SSPBUF, W        ;Загрузить в W значение из SSPBUF
aa      MOVWF   RXDATA           ;Если необходимо, сохранить значение в памяти
        MOVF    TXDATA, W       ;Загрузить в W значение из TXDATA
        MOVWF   SSPBUF          ;Передать новые данные

```

Регистр SSPSR не доступен для непосредственного чтения или записи, все операции выполняются через регистр SSPBUF. В регистре SSPSTAT находятся биты, указывающие текущее состояние модуля MSSP.

2.2.3. Настройка выводов в режиме SPI

Для включения модуля MSSP необходимо установить бит SSPEN (SSPCON<5>) в '1'. Для сброса или перенастройки режима SPI рекомендуется сбросить бит SSPEN в '0', выполнить изменения параметров работы, а затем вновь установить бит SSPEN в '1'. После включения MSSP в режиме SPI выводы SDI, SDO, SCK, -SS используются последовательным портом. Для правильной работы последовательного порта биты регистров TRIS должны быть настроены следующим образом:

- SDI, бит TRIS должен быть установлен в '1';
- SDO, бит TRIS должен быть сброшен в '0';
- SCK (ведущий режим), бит TRIS должен быть сброшен в '0';
- SCK (ведомый режим), бит TRIS должен быть установлен в '1';
- -SS, бит TRIS должен быть установлен в '1'.

Любая нежелательная функция последовательного порта может быть выключена установкой/сбросом соответствующих битов регистров направления данных TRIS. Например, если в режиме ведущего SPI выполняется только передача данных, то выводы SDI и -SS могут быть использованы как цифровые выходы, при этом соответствующие биты в регистре TRIS должны быть сброшены в '0'.

2.2.4. Типовое включение

На рис. 6 показано типовое соединение двух микроконтроллеров. Главный микроконтроллер (процессор 1) инициализирует передачу, формируя тактовый сигнал SCK. Данные сдвигаются по установленному битом SMP фронту тактового сигнала. Для одновременного приема/передачи данных (фиктивных данных) оба микроконтроллера должны иметь одинаковую полярность тактового сигнала (бит СКР). Всего существует три сценария передачи данных:

- ведущий передает данные - ведомый передает фиктивные данные;
- ведущий передает данные - ведомый передает данные;

- ведущий передает фиктивные данные - ведомый передает данные.

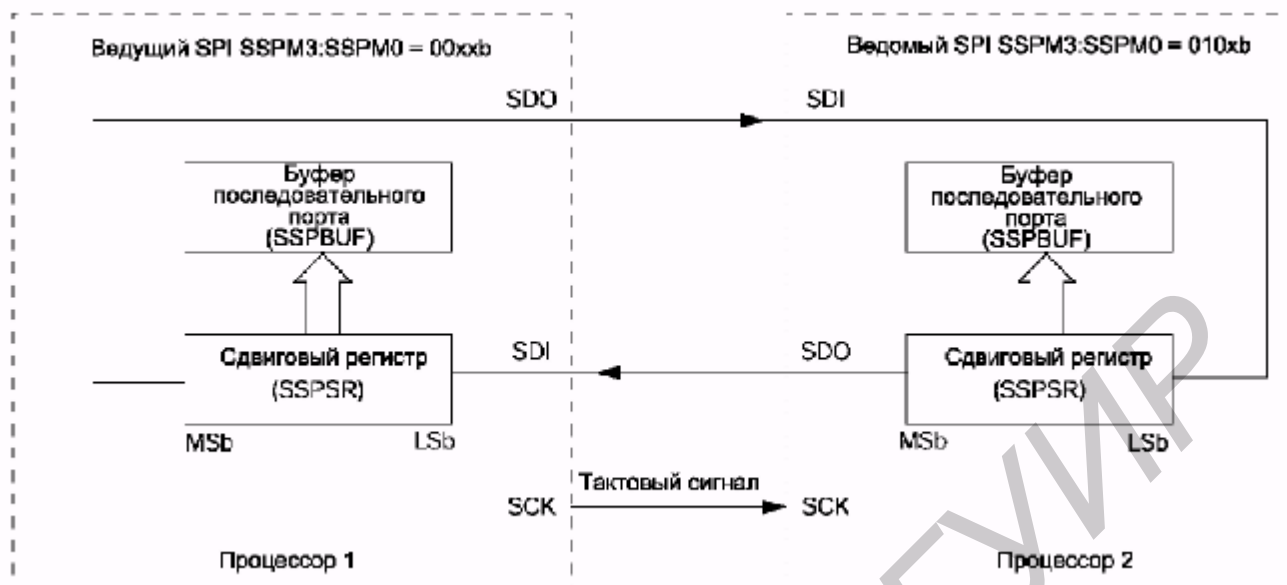


Рис. 6. Типовое соединение ведущего и ведомого SPI

2.2.5. Режим ведущего SPI

Ведущий шины может инициализировать передачу данных в любой момент, поскольку он генерирует тактовый сигнал и определяет, когда ведомый (процессор 2) должен передать данные в соответствии с используемым протоколом.

В режиме ведущего данные передаются/приняты после их записи/чтения из регистра SSPBUF. Если в SPI-режиме требуется только принимать данные, вывод SDO может быть заблокирован (настроен как вход). Данные с вывода SDI последовательно сдвигаются в регистр SSPSR с установленной скоростью. Каждый принятый байт загружается в регистр SSPBUF (как нормально полученный байт) с формированием прерываний и воздействием на соответствующие биты статуса. Эта функция может быть полезна при реализации «монитора шины».

Полярность тактового сигнала устанавливается битом СКР (SSPCON<4>), что позволяет получить различные методы передачи данных. Данные всегда передаются старшим битом вперед. В ведущем режиме частота тактового сигнала выбирается программно:

- $F_{osc}/4$ (или TCY);
- $F_{osc}/16$ (или 4 x TCY);
- $F_{osc}/64$ (или 16 x TCY);
- выход таймера TMR2 / 2.

Максимальная частота передачи данных 5МГц при тактовой частоте микроконтроллера 20МГц.

Временная диаграмма передачи данных в режиме ведущего SPI показана на рис.7. Бит SCK определяет, по какому фронту тактового сигнала необходимо выполнять прием данных. Параметры выборки входных данных устанавливаются битом SMP. Поле загрузки принятых данных в регистр SSPBUF устанавливается флаг прерываний SSPIF в '1'.

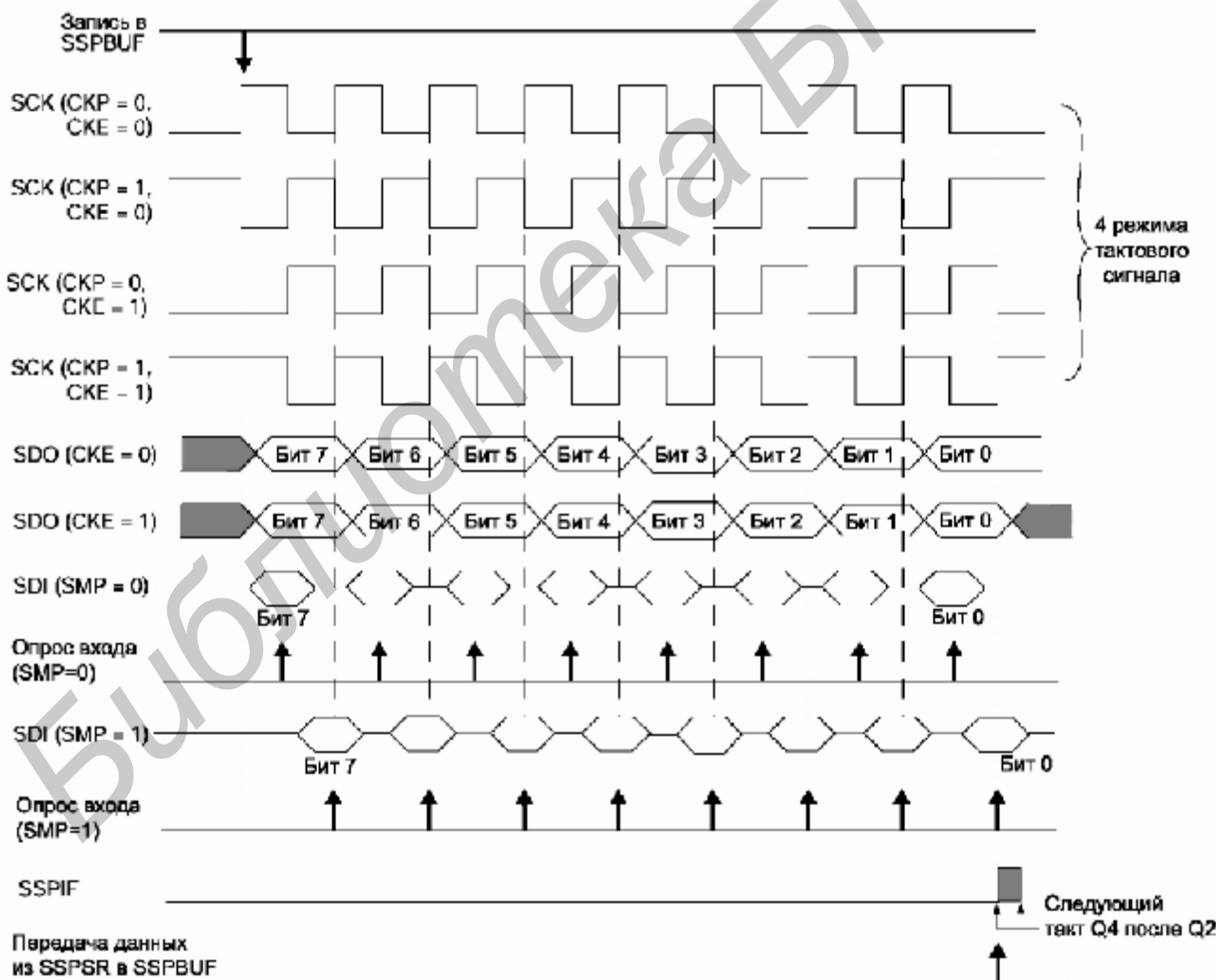


Рис. 7. Временная диаграмма работы модуля MSSP в режиме ведущего SPI

2.2.6. Работа в SLEEP-режиме микроконтроллера

В режиме ведущего SPI тактовый сигнал модуля MSSP отсутствует, состояние приема/передачи данных не изменяется до выхода микроконтроллера из режима SLEEP. После выхода микроконтроллера из режима SLEEP модуль SSP продолжит передачу/прием данных.

В режиме ведомого SPI данные могут быть приняты/переданы, т.к. сдвиговый регистр работает асинхронно. Это позволяет в SLEEP-режиме микроконтроллера принять данные в сдвиговый регистр или передать из него. Как только будут приняты все 8 бит данных, устанавливается в '1' флаг прерывания от модуля MSSP, и если прерывания разрешены, микроконтроллер выйдет из SLEEP-режима.

2.2.7. Эффект сброса

Любой сброс микроконтроллера выключает модуль MSSP, прием/передача данных прекращается.

В табл. 16 приведены регистры, влияющие на работу последовательного синхронного порта.

Таблица 16

| Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит3 | Бит 2 | Бит1 | Бит0 | Сброс FOR, BOR | Другие сбросы |
|---------|-------------------------------------------|-------|-------|-------|------|-------|------|------|----------------|---------------|
| INTCON | GIE | PEIE | TOIE | INTE | RBIE | TOIF | INTF | RBIF | 0000 | 0000 |
| PIR | SSPIF | | | | | | | | 0 | 0 |
| PIE | SSPIE | | | | | | | | 0 | 0 |
| SSPBUF | Буфер приемника SSP / регистр передатчика | | | | | | | | xxxx | iiii |
| SSPCON | WCO | SSPO | SSPE | СКР | SSPM | SSPM | SSPM | SSPM | 0000 | 0000 |
| SSPSTAT | SMP | CKE | D/-A | P | S | | UA | BF | 0000 | 0000 |

Примечание. Затененные биты на работу не влияют.

2.3. Примеры процедур инициализации модуля MSSP, передачи и приема байта данных

SPI_Init

```

bsf    STATUS,RP0
bcf    TRISC,5      ;
bcf    TRISC,3      ;
bsf    SSPSTAT,SMP ;Опрос входа в конце вывода данных
bsf    SSPSTAT,CKE  ; Данные передаются по переднему фронту
                        ; сигнала SCK (СКР=0)

```

```

bcf    STATUS,RP0
bsf    SSPCON,SSPEN    ;Включение модуля MSSP
bcf    SSPCON,CKP      ;Пассивный низкий уровень сигнала на SCK
bcf    SSPCON,SSPM3    ;Ведущий режим модуля MSSP,
                    ;тактовый сигнал Fosc/64 = 57600
bcf    SSPCON,SSPM2    ;(при Fosc = 3,6864 МГц)
bsf    SSPCON,SSPM1    ;
bcf    SSPCON,SSPM0    ;
return

```

;Процедура передачи байта. Байт данных находится в регистре W

```

btfss  SSPSTAT,BF ;ожидание окончания предыдущей передачи/приема
goto   $-1
bcf    PIR1,SSPIF
movwf  SSPBUF
return

```

;Процедура приема байта. По окончании приема байт данных находится в регистре W

```

btfss  SSPSTAT,BF ;ожидание окончания предыдущей передачи/приема
goto   $-1
bcf    PIR1,SSPIF
movwf  SSPBUF      ;Запуск приема данных
btfss  PIR1,SSPIF  ;Ожидание окончания приема байта данных
goto   $-1
bcf    PIR1,SSPIF
movf   SSPBUF,w
return

```

2.4. Варианты лабораторных заданий

Составить и отладить программу, осуществляющую запись данных в EEPROM 25LC64 (микросхема энергонезависимой памяти) с параметрами, заданными преподавателем.

Проконтролировать запись последующим чтением данных из EEPROM.

Рекомендуемые скорости работы: 0,1-3МГц.

Рекомендуемые объемы передаваемых данных: 4-60 байт.

Диапазон стартовых адресов для записи в EEPROM: 0-255.

3. ЛАБОРАТОРНАЯ РАБОТА

«Последовательный асинхронный ввод/вывод данных»

Цель работы:

- 1) изучение принципов организации последовательного асинхронного обмена данными в микропроцессорных системах;
- 2) приобретение навыков программирования процедур последовательного асинхронного обмена данными.

В микроконтроллере PIC16F877 последовательный асинхронный ввод/вывод данных производится с помощью встроенного модуля USART, который может работать в полнодуплексном асинхронном режиме для связи с терминалами, персональными компьютерами или в синхронном полудуплексном режиме для связи с микросхемами ЦАП, АЦП, последовательными EEPROM и т.д.

USART может работать в трех режимах:

- асинхронный, полный дуплекс;
- ведущий синхронный, полудуплекс;
- ведомый синхронный, полудуплекс.

Биты SPEN (RCSTA<7>) и TRISC<7:6> должны быть установлены в '1' для использования выводов RC6/TX/CK и RC7/RX/DT в качестве портов универсального синхронно-асинхронного приемопередатчика. Модуль USART поддерживает режим детектирования 9-разрядного адреса для работы в сетевом режиме.

3.1. Управляющие регистры

3.1.1. TXSTA - регистр управления и статуса передатчика

Регистр TXSTA доступен для чтения и записи, содержит флаги настройки режимов работы, а также состояния модуля USART.

Регистр TXSTA доступен по адресу 98h.

Структура регистра отражена в табл. 17.

Таблица 17

| Номер бита | Имя бита | Доступ и состояние после сброса | Назначение |
|------------|----------|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | CSRC | R/W-0 | Выбор источника тактового сигнала. <u>Синхронный режим:</u> 1 - ведущий, внутренний тактовый сигнал от BRG, 0 - ведомый, внешний тактовый сигнал с входа СК. <u>Асинхронный режим.</u> Не имеет значения |
| 6 | TX9 | R/W-0 | Разрешение 9-разрядной передачи: 1 - 9-разрядная передача, 0 - 8-разрядная передача |
| 5 | TXEN | R/W-0 | Разрешение передачи: 1 – разрешена, 0 – запрещена. <i>Примечание.</i> В синхронном режиме биты SREN/CREN отменяют действие бита TXEN |
| 4 | SYNC | R/W-0 | Режим работы USART: 1 – синхронный, 0 – асинхронный |
| 3 | - | U-0 | Не используется: читается как «0» |
| 2 | BRGH | R/W-0 | Выбор высокоскоростного режима. <u>Синхронный режим.</u> Не имеет значения. <u>Асинхронный режим:</u> 1 – высокоскоростной режим, 0 – низкоскоростной режим |
| 1 | TRMT | R-1 | Флаг очистки сдвигового регистра передатчика TSR: 1 - TSR пуст, 0 - TSR полон |
| 0 | TX9D | R/W-0 | 9-й бит передаваемых данных (может использоваться для программной проверки четности) |

3.1.2. RCSTA - регистр управления и статуса приемника

Регистр RCSTA доступен для чтения и записи, содержит флаги настройки режимов работы, а также состояния модуля USART.

Регистр RCSTA доступен по адресу 18h.

Структура регистра отражена в табл. 18.

Таблица 18

| Номер бита | Имя бита | Доступ и состояние после сброса | Назначение |
|------------|----------|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | SPEN | R/W-0 | Разрешение работы последовательного порта: 1 - модуль USART включен (выводы RC7/RX/DT, RC6/TX/CK подключены к USART), 0 - модуль USART выключен |
| 6 | RX9 | R/W-0 | Разрешение 9-разрядного приема: 1 - 9-разрядный прием, 0 - 8-разрядный прием |
| 5 | SREN | R/W-0 | Разрешение одиночного приема. <u>Синхронный режим:</u> 1 - разрешен одиночный прием, 0 - запрещен одиночный прием. Сбрасывается в '0' по завершению приема. В режиме ведомого не имеет значения. <u>Асинхронный режим.</u> Не имеет значения |
| 4 | CREN | R/W-0 | Разрешение приема. <u>Синхронный режим:</u> 1 - прием разрешен (при установке бита CREN автоматически сбрасывается бит SREN), 0 - прием запрещен. <u>Асинхронный режим:</u> 1 - прием разрешен, 0 - прием запрещен |
| 3 | ADDEN | R-0 | Разрешение детектирования адреса. <u>Асинхронный 9-разрядный прием (RX9=1):</u> 1 - детектирование адреса разрешено. Если бит RSR<8>=1, то генерируется прерывание и загружается приемный буфер, 0 - детектирование адреса запрещено. Принимаются все байты, девятый бит может использоваться для проверки четности. <u>Асинхронный 8-разрядный прием (RX9=0).</u> Не имеет значения. <u>Синхронный режим.</u> Не имеет значения |
| 2 | FERR | R-0 | Ошибка кадра, сбрасывается при чтении регистра RCREG: 1 - произошла ошибка кадра, 0 - ошибки кадра не было |
| 1 | OERR | R-x | Ошибка переполнения внутреннего буфера, устанавливается в '0' при сбросе бита CREN: 1 - произошла ошибка переполнения, 0 - ошибки переполнения не было |
| 0 | RX9D | | 9-й бит принятых данных (может использоваться для программной проверки четности) |

3.2. Генератор частоты обмена USART BRG

BRG используется для работы USART в синхронном ведущем и асинхронном режимах. BRG представляет собой отдельный 8-разрядный генератор скорости обмена в бодах, период которого определяется значением в регистре SPBRG. В асинхронном режиме бит BRGH (TXSTA<2>) тоже влияет на скорость обмена (в синхронном режиме бит BRGH игнорируется). В табл. 19 указаны формулы для вычисления скорости обмена в бодах при различных режимах работы модуля USART (относительно внутреннего тактового сигнала микроконтроллера).

Учитывая требуемую скорость и F_{OSC} , выбирается самое близкое целое значение для записи в регистр SPBRG, рассчитанное по формулам, приведенным в табл. 19. Затем рассчитывается ошибка скорости обмена.

В некоторых случаях может быть выгодно использовать высокоскоростной режим работы USART (BRGH=1), поскольку уравнение $F_{OSC} / (16 (X + 1))$ позволяет уменьшить погрешность скорости. Запись нового значения в регистр SPBRG сбрасывает таймер BRG, гарантируя немедленный переход на новую скорость.

Таблица 19

| SYNC | BRGH = 0 | BRGH = 1 |
|------|--------------------------------------------------------------|---------------------------------------------------------|
| 0 | (Асинхронный) Скорость обмена = $=F_{osc} / (64 (X + 1))$ | (Асинхронный) Скорость обмена = $=F_{osc}/(16(X+1))$ |
| 1 | (Синхронный) Скорость обмена = $=F_{osc} / (4 (X + 1))$ | (Синхронный) Скорость обмена = $=F_{osc}/(4(X+1))$ |

X = значение регистра SPBRG (от 0 до 255)

В табл. 20 — 23 приведены результаты расчетов скорости передачи при различных значениях параметров настройки модуля USART.

3.3. Выборка

Сигнал с входа RC7/RX/DT тестируется цепью мажоритарного детектора три раза за такт передачи, чтобы определить, высокого или низкого уровня

сигнал присутствует на входе. Если выбран низкоскоростной режим (BRGH=0), то выборка производится по седьмому, восьмому и девятому заднему фронту тактового сигнала x16. Если BRGH = 1 (выбран высокоскоростной режим), выборка производится на втором такте сигнала x4 тремя запросами).

Таблица 20

| BRGH=0 | | | | | | |
|-----------------------------------|---------------------------|-----------|----------------|---------------------------|-----------|----------------|
| Стандартная скорость обмена, Кбит | F _{OSC} = 20 МГц | | | F _{OSC} = 10 МГц | | |
| | Реальная скорость, Кбит | Ошибка, % | Значение SPBRG | Реальная скорость, Кбит | Ошибка, % | Значение SPBRG |
| 0,3 | - | - | - | - | - | - |
| 1,2 | 1,221 | 1,75 | 255 | 1,202 | 0,17 | 129 |
| 2,4 | 2,404 | 0,17 | 129 | 2,404 | 0,17 | 64 |
| 9,6 | 9,766 | 1,73 | 31 | 9,766 | 1,73 | 15 |
| 19,2 | 19,531 | 1,72 | 15 | 19,531 | 1,72 | 7 |
| 28,8 | 31,250 | 8,51 | 9 | 31,25 | 8,51 | 4 |
| 33,6 | 34,722 | 3,34 | 8 | 31,25 | 6,99 | 4 |
| 57,6 | 62,500 | 8,51 | 4 | 52,083 | 9,58 | 2 |
| Максимум | 312,5 | - | 0 | 156,250 | - | 0 |
| Минимум | 1,221 | - | 255 | 0,610 | - | 255 |

Таблица 21

| BRGH=0 | | | | | | |
|-----------------------------------|--------------------------|-----------|----------------|-------------------------------|-----------|----------------|
| Стандартная скорость обмена, Кбит | F _{OSC} = 4 МГц | | | F _{OSC} = 3,6864 МГц | | |
| | Реальная скорость, Кбит | Ошибка, % | Значение SPBRG | Реальная скорость, Кбит | Ошибка, % | Значение SPBRG |
| 0,3 | 0,3 | 0 | 207 | 0,3 | 0 | 191 |
| 1,2 | 1,202 | 0,17 | 51 | 1,2 | 0 | 47 |
| 2,4 | 2,404 | 0,17 | 25 | 2,4 | 0 | 23 |
| 9,6 | 8,929 | 6,99 | 6 | 9,6 | 0 | 5 |
| 19,2 | 20,833 | 8,51 | 2 | 19,2 | 0 | 2 |
| 28,8 | 31,250 | 8,51 | 1 | 28,8 | 0 | 1 |
| 33,6 | - | - | - | - | - | - |
| 57,6 | 62,500 | 8,51 | 0 | 57,6 | 0 | 0 |
| Максимум | 62,500 | - | 0 | 57,6 | - | 0 |
| Минимум | 0,244 | - | 255 | 0,225 | - | 255 |

Таблица 22

| BRGH=1 | | | | | | |
|-----------------------------------|---------------------------|-----------|----------------|---------------------------|-----------|----------------|
| Стандартная скорость обмена, Кбит | F _{osc} = 20 МГц | | | F _{osc} = 10 МГц | | |
| | Реальная скорость, Кбит | Ошибка, % | Значение SPBRG | Реальная скорость, Кбит | Ошибка, % | Значение SPBRG |
| 0,3 | - | - | - | - | - | - |
| 1,2 | - | - | - | - | - | - |
| 2,4 | - | - | - | 2,441 | 1,71 | 255 |
| 9,6 | 9,615 | 0,16 | 129 | 9,615 | 0,16 | 64 |
| 19,2 | 19,231 | 0,16 | 64 | 19,531 | 1,72 | 31 |
| 28,8 | 29,070 | 0,94 | 42 | 28,409 | 1,36 | 21 |
| 33,6 | 33,784 | 0,55 | 36 | 32,895 | 2,10 | 18 |
| 57,6 | 59,524 | 3,34 | 20 | 56,818 | 1,36 | 10 |
| Максимум | 1250 | - | 0 | 625 | - | 0 |
| Минимум | 4,883 | - | 255 | 2,441 | - | 255 |

Таблица 23

| BRGH=1 | | | | | | |
|-----------------------------------|--------------------------|-----------|----------------|-------------------------------|-----------|----------------|
| Стандартная скорость обмена, Кбит | F _{osc} = 4 МГц | | | F _{osc} = 3,6864 МГц | | |
| | Реальная скорость, Кбит | Ошибка, % | Значение SPBRG | Реальная скорость, Кбит | Ошибка, % | Значение SPBRG |
| 0,3 | - | - | - | - | - | - |
| 1,2 | 1,202 | 0,17 | 207 | 1,2 | 0 | 191 |
| 2,4 | 2,404 | 0,17 | 103 | 2,4 | 0 | 95 |
| 9,6 | 9,615 | 0,16 | 25 | 9,6 | 0 | 23 |
| 19,2 | 19,231 | 0,16 | 12 | 19,2 | 0 | 11 |
| 28,8 | 27,798 | 3,55 | 8 | 28,8 | 0 | 7 |
| 33,6 | 35,714 | 6,29 | 6 | 32,9 | 2,04 | 6 |
| 57,6 | 62,500 | 8,51 | 3 | 57,6 | 0 | 3 |
| Максимум | 250 | - | 0 | 230,4 | - | 0 |
| Минимум | 0,977 | - | 255 | 0,9 | - | 255 |

3.4. Асинхронный режим USART

В этом режиме USART использует стандартный формат NRZ: один стартовый бит, восемь или девять битов данных и один стоповый бит. Наиболее часто встречается 8-разрядный формат передачи данных.

Интегрированный 8-разрядный генератор BRG позволяет получить стандартные скорости передачи данных. Генератор скорости обмена может работать в одном из двух режимов: высокоскоростном (x16 BRGH=1

TXSTA<2>) и низкоскоростном (x64 BRGH=0 TXSTA<2>). Приемник и передатчик последовательного порта работают независимо друг от друга, но используют один и тот же формат данных и одинаковую скорость обмена. Бит четности аппаратно не поддерживается, но может быть реализован программно, применяя 9-разрядный формат данных. Все данные передаются младшим битом вперед.

В SLEEP-режиме микроконтроллера модуль USART (асинхронный режим) выключен.

Выбор асинхронного режима USART выполняется сбросом бита SYNC в '0' (TXSTA<4>).

Модуль USART в асинхронном режиме состоит из следующих элементов: генератора скорости обмена, цепи опроса, асинхронного передатчика, асинхронного приемника.

В табл. 24 показаны регистры и биты, связанные с работой передатчика и приемника USART в асинхронном режиме.

Таблица 24

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Состояние после сброса POR, BOR |
|----------------------|--------|-----------------------------------|-------|-------|-------|-------|--------|--------|--------|---------------------------------|
| 0Bh, 8Bh, 10Bh, 18Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | TOIF | INTF | RBIF | 0000 000x |
| 0Ch | PIR1 | PSPIF | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 |
| 8Ch | PIE1 | PSPIE | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x |
| 19h | TXREG | Регистр данных передатчика USART | | | | | | | | 0000 0000 |
| 1Ah | RCREG | Регистр данных приемника USART | | | | | | | | 0000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | - | BRGH | TRMT | TX9D | 0000 - 010 |
| 99h | SPBRG | Регистр генератора скорости USART | | | | | | | | 0000 0000 |

3.5. Асинхронный передатчик USART

Структурная схема асинхронного передатчика USART показана на рис. 8. Главным в передатчике является сдвиговый регистр TSR, который получает данные из буфера передатчика TXREG. Данные в регистр TXREG загружаются программно. После передачи стопового бита предыдущего байта в последнем

машинном такте цикла BRG TSR загружается новым значением из TXREG (если оно есть), после чего устанавливается флаг прерывания TXIF (PIR1<4>). Прерывание может быть разрешено или запрещено битом TXIE (P1E1<4>). Флаг TXIF устанавливается независимо от состояния бита TXIE и не может быть сброшен в '0' программно. Очистка флага TXIF происходит только после загрузки новых данных в регистр TXREG. Аналогичным образом бит TRMT (TXSTA<1>) отображает состояние регистра TSR. Бит TRMT доступен только на чтение и не может вызвать генерацию прерывания.

Регистр TSR не отображается на память и недоступен для чтения.

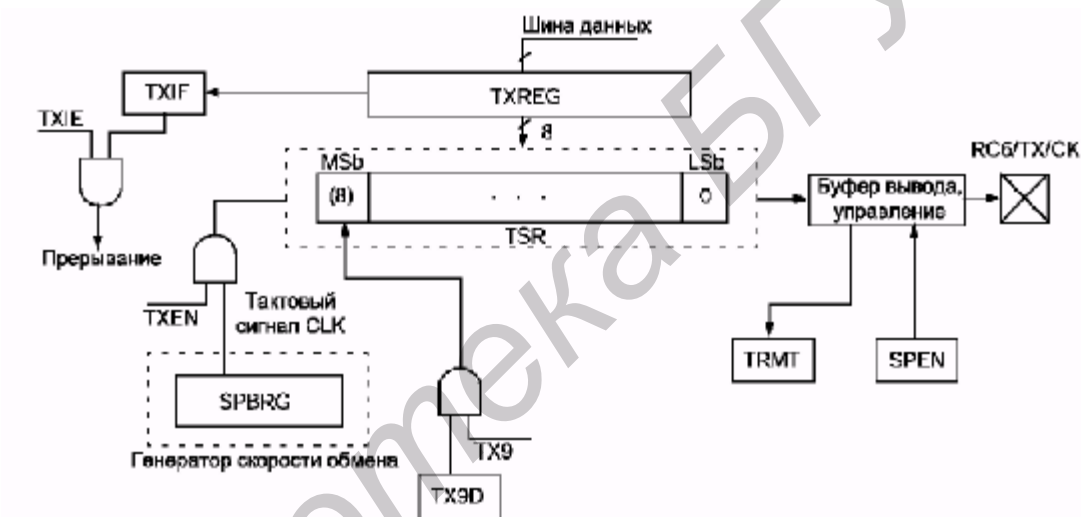


Рис. 8. Структурная схема асинхронного передатчика USART

Флаг TXIF устанавливается в '1' только тогда, когда бит TXEN=1, и сбрасывается автоматически в '0' после загрузки новых данных в регистр TXREG.

Для разрешения передачи необходимо установить бит TXEN (TXSTA<5>) в '1'. Передача данных не начнется до тех пор, пока в TXREG не будут загружены новые данные; не придет очередной тактовый импульс от генератора BRG (рис. 9). Можно сначала загрузить данные в TXREG, а затем установить бит TXEN. Как правило, после разрешения передачи регистр TSR пуст, таким образом, данные, записываемые в TXREG, сразу передаются в TSR, а TXREG остается пустым. Это позволяет реализовать слитную передачу

данных (рис. 10). Сброс бита TXEN в '0' вызовет немедленное прекращение передачи, сброс передатчика и перевод вывода RC6/TX/CK в третье состояние.

Для разрешения 9-разрядной передачи необходимо установить бит TX9 (TXSTA<6>) в '1'. Девятый бит данных записывается в бит TX9D (TXSTA<0>). Девятый бит данных должен быть указан до записи в регистр TXREG, потому что данные, записанные в регистр TXREG, могут быть сразу загружены в сдвиговый регистр TSR (если он пуст).

Рекомендованная последовательность действий для передачи данных в асинхронном режиме:

- установить требуемую скорость передачи с помощью регистра SPBRG и бита BRGH;
- выбрать асинхронный режим сбросом бита SYNC в '0' и установкой бита SPEN в '1';
- если необходимо, разрешить прерывания установкой бита TXIE в '1';
- если передача 9-разрядная, установить бит TX9 в '1';
- разрешить передачу установкой бита TXEN в '1', — автоматически устанавливается флаг TXIF;
- если передача 9-разрядная, записать 9-й бит данных в TX9D;
- записать данные в регистр TXREG;
- если используются прерывания, то биты GIE и PEIE в регистре INTCON должны быть установлены в '1'.

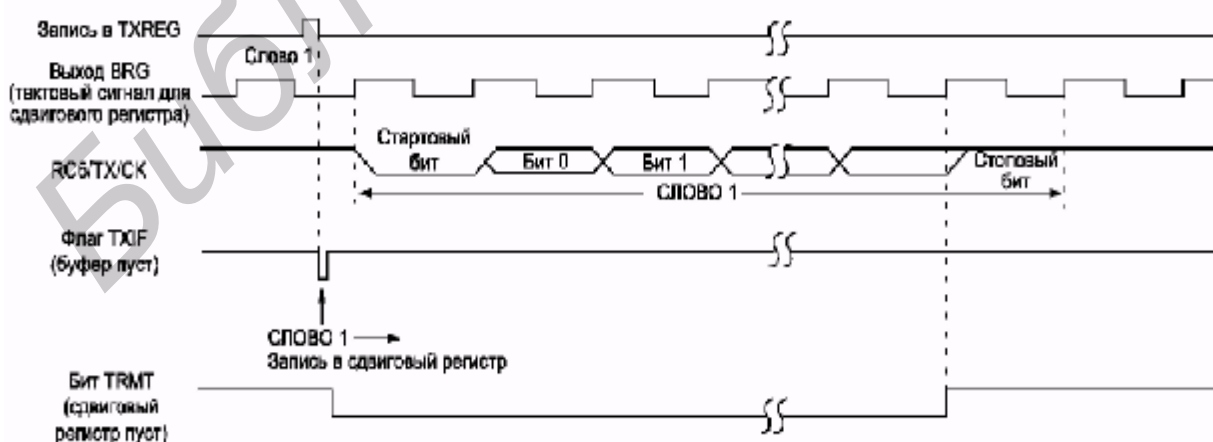


Рис. 9. Временная диаграмма асинхронной передачи данных

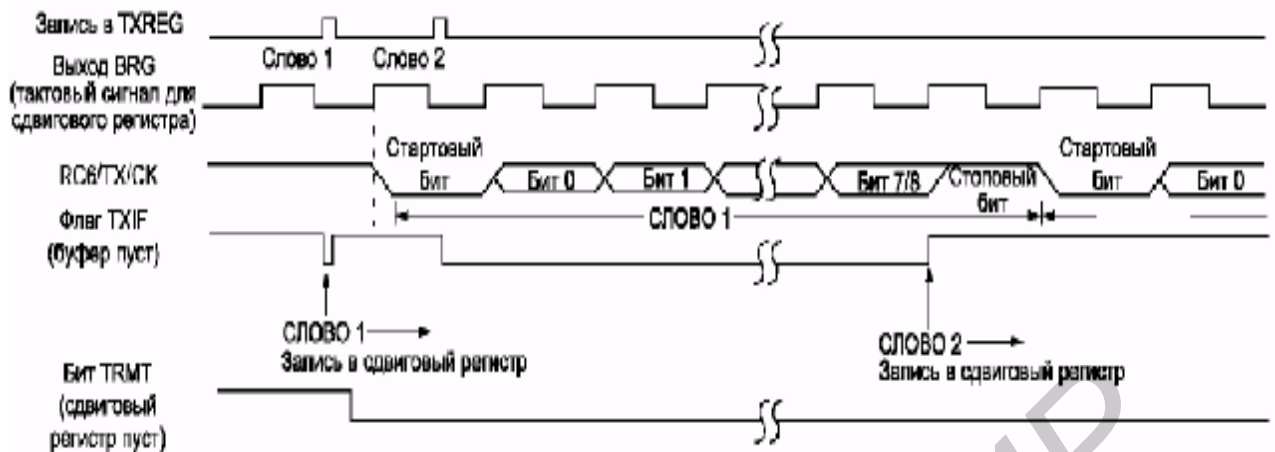


Рис. 10. Временная диаграмма слитной асинхронной передачи (последовательная передача двух байтов)

3.6. Асинхронный приемник USART

Структурная схема асинхронного приемника USART показана на рис. 11. Данные подаются на вход RC7/RX/DT в блок восстановления данных, представляющий собой скоростной сдвиговый регистр, работающий на частоте, в 16 раз превышающей скорость передачи или F_{osc} .

Включение приемника производится установкой бита CREN регистра RCSTA в '1'.

Главным в приемнике является сдвиговый регистр RSR. После получения стопового бита данные переписываются в регистр RCREG, если он пуст. После записи в регистр RCREG выставляется флаг прерывания RCIF (PIR1<5>). Прерывание можно разрешить/запретить установкой/сбросом бита RCIE (PIE1<5>). Флаг RCIF доступен только для чтения, сбрасывается аппаратно при чтении из регистра RCREG. Регистр RCREG имеет двойную буферизацию, т.е. представляет собой двухуровневый буфер FIFO. Поэтому можно принять 2 байта данных в FIFO RCREG и третий в регистр RSR. Если FIFO заполнен и обнаружен стоповый бит третьего байта, — устанавливается бит переполнения приемника OERR (RCSTA<1>). Байт, принятый в RSR, будет потерян. Для

извлечения двух байтов из FIFO необходимо дважды прочитать регистр RCREG. Бит OERR нужно программно очистить сбросом бита CREN, т.е. запрещением приема. В любом случае, если бит OERR установлен, логика приемника выключена.

Бит ошибки кадра FERR (RCSTA<2>) устанавливается в '1', если не обнаружен стоповый бит. FERR и девятый бит принятых данных буферизируются так же, как принятые данные. Рекомендуется сначала прочитать регистр RCSTA, затем RCREG, чтобы не потерять информацию RX9D и FERR.

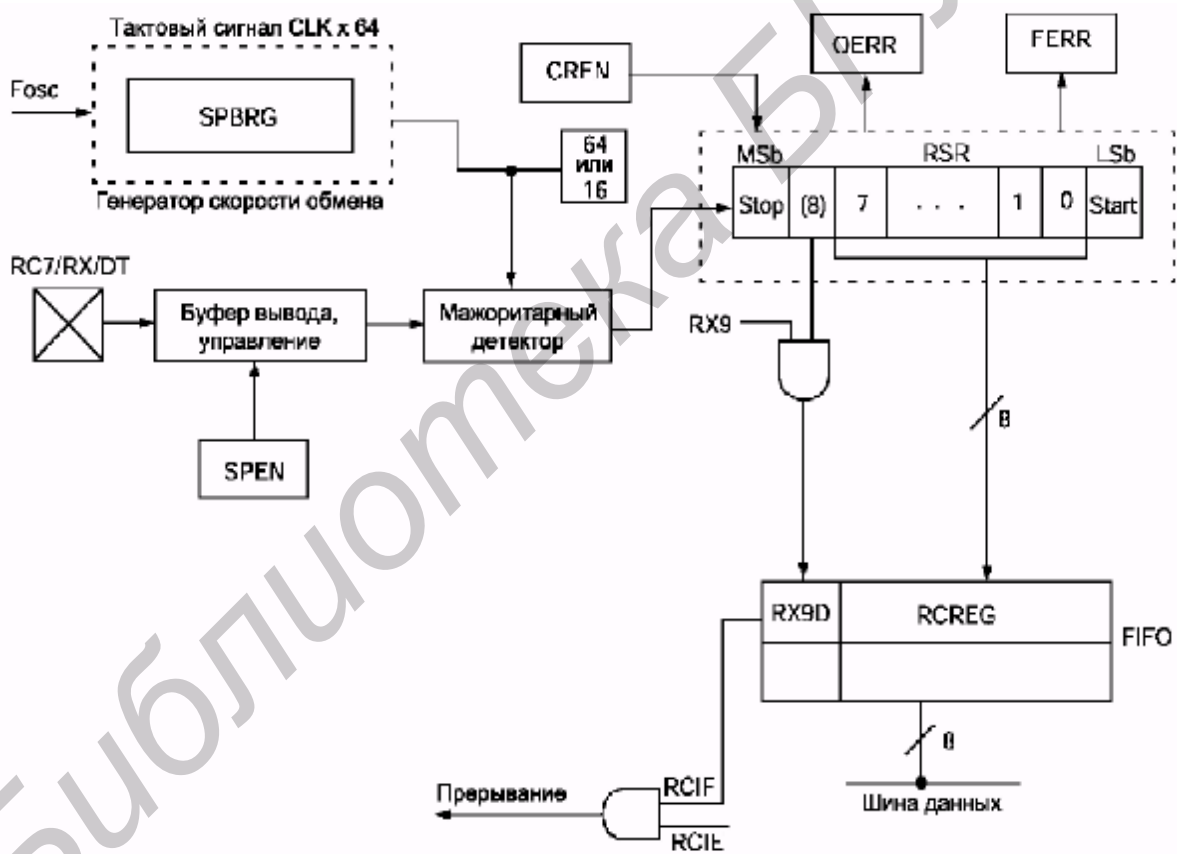


Рис. 11. Структурная схема асинхронного приемника USART

На временной диаграмме рис. 12 показан последовательный прием трех байтов. Регистр RCREG (приемный буфер) читается после приема трех байтов, поэтому устанавливается бит OERR в '1'.

Рекомендованные действия при приеме данных в асинхронном режиме:

- установить требуемую скорость передачи с помощью регистра SPBRG и бита BRGH;
- выбрать асинхронный режим сбросом бита SYNC в '0' и установкой бита SPEN в '1';
- если необходимо, разрешить прерывания установкой бита RCIE в '1';
- если прием 9-разрядный, установить бит RX9 в '1';
- разрешить прием установкой бита CREN в '1';
- ожидать установку бита RCIF или прерывание, если оно разрешено битом RCIE;
- считать 9-й бит данных (если разрешен 9-разрядный прием) из регистра RCSTA и проверить возникновение ошибки;
- считать восемь битов данных из регистра RCREG;
- при возникновении ошибки переполнения сбросить бит CREN в '0';
- если используются прерывания, то биты GIE и PEIE в регистре INTCON должны быть установлены в '1'.

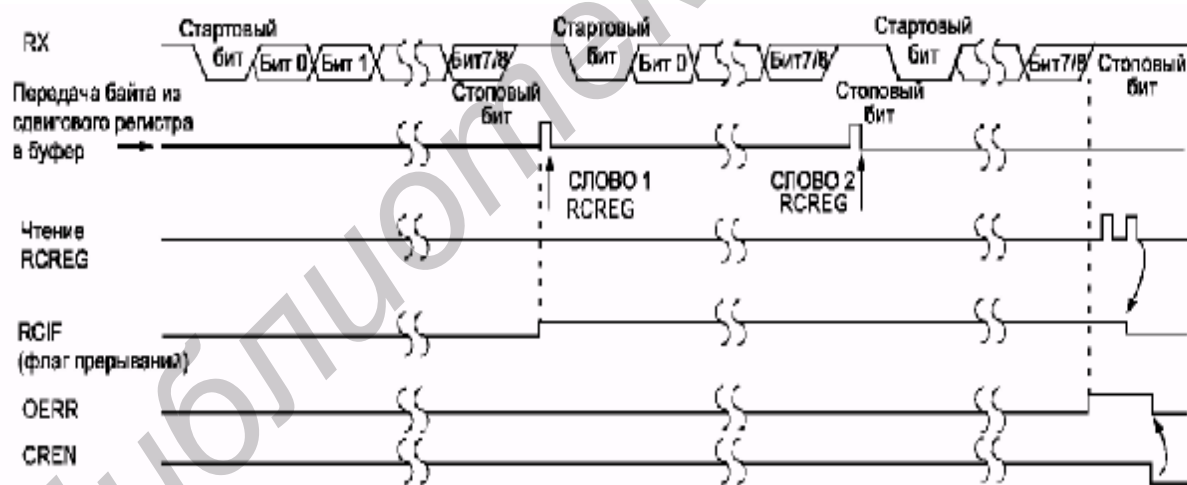


Рис. 12. Временная диаграмма приема данных

3.7. Настройка 9-разрядного асинхронного приема с детектированием адреса

Структурная схема асинхронного приемника USART в режиме детектирования адреса показана на рис. 13.

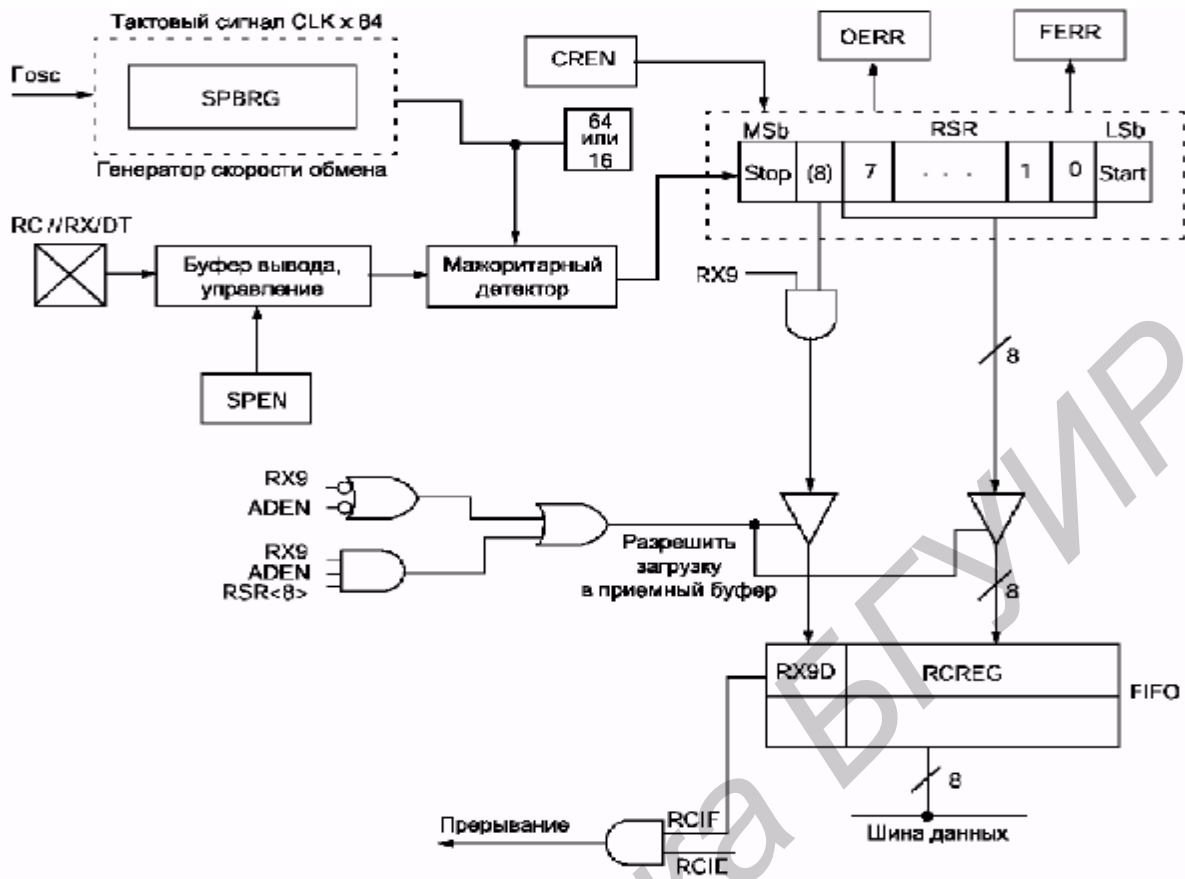


Рис. 13. Структурная схема асинхронного приемника USART в режиме детектирования адреса

Рекомендуется следующая последовательность действия при использовании детектора адреса:

- установить требуемую скорость передачи с помощью регистра SPBRG и бита BRGH;
- выбрать асинхронный режим сбросом бита SYNC в '0' и установкой бита SPEN в '1';
- если необходимо, разрешить прерывания установкой бита RCIE в '1';
- установить бит RX9 в '1' для включения 9-разрядного приема;
- установить бит ADDEN в '1' для разрешения детектирования адреса;
- разрешить прием установкой бита CREN в '1';
- ожидать установку бита RCIF или прерывание, если оно разрешено битом RCIE;

- считать восемь битов данных из регистра RCREG для проверки адресации устройства;
- при возникновении ошибки переполнения сбросить бит CREN в '0';
- если принятый адрес соответствует адресу устройства, сбросить биты ADDEN и RCIF в '0' для начала приема данных.

На рис. 14 показана временная диаграмма последовательного приема байта данных и байта адреса. Байт данных не записывается в RCREG, т.к. ADDEN=1, а бит 8 = 0.

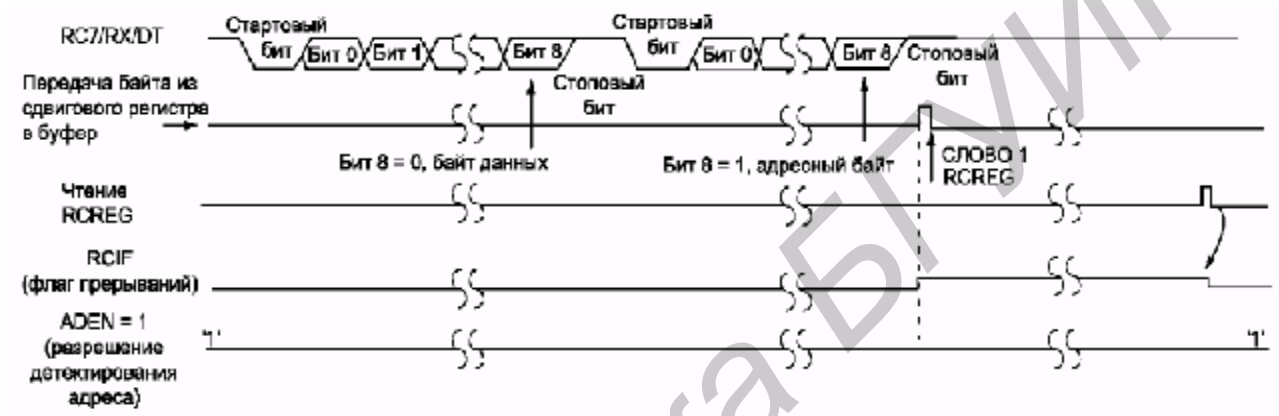


Рис. 14. Временная диаграмма приема байта с детектированием адреса

На рис. 15 показана временная диаграмма последовательного приема байта адреса и байта данных. Байт данных не записывается в RCREG, т.к. ADDEN не был сброшен в '0' (ADDEN=1), а бит 8 = 0.

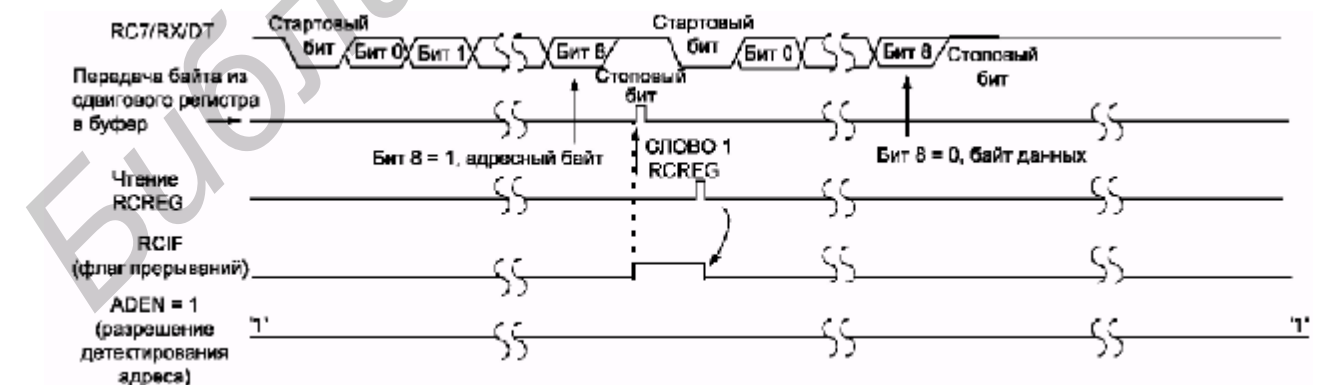


Рис. 15. Временная диаграмма приема данных с первым адресным байтом

3.8. Пример управляющей программы для USART

```
; Установка скорости обмена 28800 бод (при Fosc = 3,6864 МГц)
    bsf     STATUS,RP0      ; Выбор 1-го банка
    movlw  0x07
    movwf  SPBRG

; Инициализация передатчика USART
    bcf     TXSTA,TX9      ; 8-разрядная передача
    bsf     TXSTA,TXEN     ; Разрешение передачи
    bcf     TXSTA,SYNC     ; Асинхронный режим работы USART
    bsf     TXSTA,BRGH     ; Высокоскоростной режим

; Инициализация приемника USART
    bcf     STATUS,RP0      ; Выбор 0-го банка
    bsf     RCSTA,SPEN     ; Разрешение работы последовательного порта
    bcf     RCSTA,RX9      ; 8-разрядный прием
    bsf     RCSTA,CREN     ; Разрешение приема

wait_byte
    btfss  PIR1,RCIF      ; Ожидание приема байта данных
    goto  $-1
    movf   RCREG,w        ; Считывание принятого байта данных
    movwf  TXREG          ; Передача принятого байта данных
    goto  wait_byte
```

3.9. Варианты лабораторных заданий

1. Составить и отладить программу, осуществляющую побайтный прием данных от компьютера, инверсию принятого байта и передачу его обратно на компьютер. Использовать прерывание от приемника USART.

Скорость связи и режим работы определяются преподавателем.

2. По команде от компьютера осуществить передачу массива данных от микроконтроллера компьютеру.

Скорость связи, режим работы и объем данных определяются преподавателем.

Возможные скорости связи: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 56800 бит/с.

Возможные режимы работы: 8 битов данных, 9 битов данных.

Возможный объем данных: 4-60 байтов.

ЛИТЕРАТУРА

1. PIC16F87X. Однокристалльные 8-разрядные FLASH CMOS микроконтроллеры компании Microchip technology incorporated: Пер. с англ. –М.: ООО «Микрочип», 2002. – 184 с. www.microchip.ru

2. Бурак А.И., Левкович В.Н. Интегрированная среда MPLab IDE разработки программ для микроконтроллеров PICmicro фирмы Microchip: Метод. пособие к лабораторным работам по курсу «Цифровые и микропроцессорные устройства». – Мн.: БГУИР, 2003. – 31 с.

3. Левкович В.Н. и др. Конструирование программ на Ассемблере для микроконтроллеров семейства PICmicro: Учеб. пособие по курсу “Цифровые и микропроцессорные устройства» для студ. спец. 39 01 01 «Радиотехника» и 39 01 02 «Радиоэлектронные системы» всех форм обуч. -Мн.: БГУИР, 2004. - 80 с.

Учебное издание

Левкович Василий Николаевич,
Бурак Андрей Иосифович

**ВЫЧИСЛИТЕЛЬНЫЕ И КОММУНИКАЦИОННЫЕ
СРЕДСТВА РАДИОСИСТЕМ**

Лабораторный практикум
для студентов специальности
39 01 02 «Радиоэлектронные системы»
дневной формы обучения
В 2-х частях
Часть 1

Редактор Т. Н. Крюкова
Корректор Е.Н. Батурчик

Подписано в печать 16.11.2004.
Гарнитура «Таймс».
Уч.-изд. л. 2,5.

Формат 60x84 1/16.
Печать ризографическая.
Тираж 100 экз.

Бумага офсетная.
Усл. печ. л. 3,6.
Заказ 618.

Издатель и полиграфическое исполнение: Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
Лицензия на осуществление издательской деятельности №02330/0056964 от 01.04.2004.
Лицензия на осуществление полиграфической деятельности №02330/0133108 от 30.04.2004.
220013, Минск, П. Бровки, 6