

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

Кафедра радиотехнических систем

В.Н. Левкович

ЦИФРОВЫЕ И МИКРОПРОЦЕССОРНЫЕ УСТРОЙСТВА

Лабораторный практикум
для студентов специальностей
I-39 01 02 «Радиоэлектронные системы»,
I-39 01 03 «Радиоинформатика»,
I-39 01 04 «Радиоэлектронная защита информации»

В 2-х частях

Часть 1

Минск 2005

УДК 681.325.5-181.48(075.8)

ББК 32.973.26-04 я 73

Л 37

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 37 Цифровые и микропроцессорные устройства: Лаб. практикум для студ. спец. I-39 01 02 «Радиоэлектронные системы», I-39 01 03 «Радиоинформатика», I-39 01 04 «Радиоэлектронная защита информации». В 2 ч. Ч. 1. –Мн.: БГУИР, 2005. - 38 с.: ил.

ISBN 985-444-817-7 (ч. 1)

В первую часть практикума вошли краткие теоретические сведения по аппаратной организации ядра, портов ввода/вывода и периферийного модуля счетчика-таймера однокристалльного микроконтроллера PIC16F628, приводится также его система команд. Приведены материалы, относящиеся к лабораторной работе по теме «Архитектура ядра однокристалльного микроконтроллера PIC16F628», лабораторное задание, контрольные вопросы для самопроверки.

УДК 681.325.5-181.48(075.8)

ББК 32.973.26-04 я 73

ISBN 985-444-817-7 (ч. 1)

ISBN 985-444-818-5

© Левкович В.Н., 2005

© БГУИР, 2005

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

1. АРХИТЕКТУРА ЯДРА ОМК PIC16F628

1.1. Общие сведения

1.2. Структурная организация

1.3. Организация памяти

1.3.1. Организация памяти программы

1.3.2. Организация памяти данных

1.4. Система команд

1.5. Регистр состояния STATUS

1.6. Регистр OPTION_REG

1.7. Регистр INTCON

1.8. Косвенная адресация данных

1.9. Прерывания

1.9.1. Внешнее прерывание с входа RB0/INT

1.9.2. Прерывание по переполнению TMR0

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

1.9.4. Сохранение контекста при обработке прерываний

1.10. Порты ввода/вывода

1.10.1. PORTA

1.10.2. PORTB

1.11. Особенности программирования портов

1.11.1 Организация двунаправленных портов

1.11.2 Обращение к портам ввода/вывода

1.12. Модуль таймера TMR0

1.12.1. Прерывание от таймера

1.12.2. Использование TMR0 с внешним сигналом

1.12.3. Предварительный делитель

1.13. Лабораторное задание

1.14. Контрольные вопросы

ЛИТЕРАТУРА

ВВЕДЕНИЕ

Однокристалльные микроконтроллеры (ОМК) содержат в одном корпусе интегральной микросхемы все функциональные блоки электронной вычислительной машины: арифметико-логическое устройство (АЛУ), устройство управления (УУ), операционные регистры, управляющие регистры, память программ, память данных, порты ввода/вывода. Кроме этого, на кристалле микросхемы часто размещают и периферийные устройства, такие, как таймеры, аналого-цифровой преобразователь (АЦП), цифроаналоговый преобразователь (ЦАП), компараторы, энергонезависимая память, драйверы стандартных последовательных интерфейсов и др. Однокристалльные микроконтроллеры предназначены для работы по одной программе, записываемой в энергонезависимую память программ и не меняемой в течение всего цикла эксплуатации. Они, как правило, являются встроенными в аппаратуру вычислительными машинами и выполняют функции управления, а также формирования и обработки сигналов и информации.

В настоящее время ОМК производят несколько десятков фирм. Среди них микроконтроллеры PICmicro американской фирмы Microchip Technology Inc. отличаются высокой производительностью, низким энергопотреблением, гибкой и развитой архитектурой, широкими функциональными возможностями, развитой периферией, простотой в освоении, низкой стоимостью. В зависимости от производительности и функциональных возможностей PICmicro подразделяются на семейства: PIC12, PIC16, PIC17 и PIC18. Самые простые модели выпускаются в 8-выводных, а сложные – в 80-выводных корпусах. При этом все семейства имеют общую базовую архитектуру и систему команд, что упрощает их изучение. Такое сочетание потребительских свойств сделало PICmicro одними из самых популярных ОМК у разработчиков радиоэлектронной аппаратуры.

Для лабораторного цикла выбрана модель PIC16F628. Она является почти идеальной для учебных целей: отличается простотой, но при этом несет в себе

важнейшие черты всех семейств, что облегчает освоение других моделей микроконтроллеров. Наличие встроенной электрически перепрограммируемой памяти программ позволяет легко тестировать разрабатываемые схемы и программы. В рамках учебного курса студенту предоставляется возможность пройти полный путь от постановки задачи до работающей конструкции.

В лабораторном практикуме описание ОМК PIC16F628 дано лишь в объеме, необходимом для начального изучения и выполнения лабораторных работ по следующим темам:

«Архитектура ядра однокристального микроконтроллера PIC16F628»;

«Алгоритмы и программирование процедур формирования импульсных сигналов с заданными временными параметрами»;

«Алгоритмы и программирование процедур отображения цифровой информации»;

«Алгоритмы и программирование процедур ввода данных с клавиатуры».

1. АРХИТЕКТУРА ЯДРА ОМК PIC16F628

1.1. Общие сведения

Ядро однокристального микроконтроллера PIC16F628 разработано в соответствии с модифицированной гарвардской RISC-архитектурой и изготавливается по высокоскоростной КМОП-технологии. Оно имеет внутреннюю энергонезависимую электрически перепрограммируемую память программ (FLASH) емкостью 2048 14-разрядных слов, 8-битную длину машинного слова и 224-байтный внутреннюю память данных. Система команд включает 35 инструкций. Все команды имеют длину в одно слово шириной 14 бит и исполняются за один машинный цикл (200 нс при максимальной тактовой частоте 20 МГц), кроме команд перехода, которые выполняются за два цикла (400 нс).

PIC16F628 имеет прерывание, срабатывающее от десяти источников, и восьмиуровневый аппаратный стек.

Периферия включает в себя три таймера, два аналоговых компаратора, источник опорного напряжения, модуль широтно-импульсной модуляции, последовательный синхронно-асинхронный приемопередатчик USART, энергонезависимую память данных EEPROM, 15 линий двунаправленного ввода/вывода и др.

В настоящей работе мы ограничимся изучением лишь цифровых портов ввода/вывода и 8-битного таймера/счетчика TMR0 с 8-битным программируемым предварительным делителем.

Особенность портов в высокой нагрузочной способности: 25 мА максимальный втекающий и вытекающий токи. Это упрощает схемы внешних устройств, за счет чего уменьшается общая стоимость разрабатываемых систем.

Микроконтроллер способен работать в широком диапазоне тактовых частот – от 0 до 20 МГц, широком диапазоне питающих напряжений – от 2 до 5,5 В, в широком температурном диапазоне – от -40 до +125 °С. Он отличается

также низким энергопотреблением: менее 2 мА при напряжении питания 5 В и тактовой частоте 4 МГц, 15 мкА при питании 3 В и тактовой частоте 32 кГц.

Разработки на базе PIC-контроллеров поддерживаются ассемблером, программным симулятором, внутрисхемным эмулятором и программатором.

1.2. Структурная организация

Укороченная структурная схема PIC16F628 показана на рис. 1.1. В ней отражены функциональные блоки, составляющие ядро микроконтроллера, а также порты ввода/вывода PORTA и PORTB, счетчик-таймер TMR0 и некоторые блоки, повышающие надежность работы в реальных условиях: схема сброса по включению питания POR, таймер включения питания PWRT, таймер запуска генератора OST и сторожевой таймер WDT.

Назначение выводов микроконтроллера PIC16F877 отражено в табл. 1.1.

Архитектура микроконтроллера основана на концепции отдельных шин и областей памяти для данных и для программ (гарвардская архитектура). Это увеличивает скорость обмена по сравнению с традиционной прынстонской архитектурой, в которой команды и данные передаются по одной и той же шине. Разделение шин команд и данных позволяет увеличить разрядность команды по сравнению с разрядностью данных. Шина данных и память данных (ОЗУ) имеют ширину 8 битов, а программная шина и программная память (ПЗУ) - ширину 14 битов. Такая концепция обеспечивает простую, но мощную систему однословных команд, разработанную так, что битовые, байтовые и регистровые операции работают с высокой скоростью и с перекрытием по времени выборки команд и циклов выполнения. 14-битная ширина программной памяти обеспечивает выборку 14-битной команды за один цикл. Двухступенчатый конвейер обеспечивает одновременную выборку следующей и исполнение текущей команды. Все команды выполняются за один машинный цикл, исключая команды переходов, которые выполняются за два цикла. В PIC16F628 программная па-

мять расположена внутри кристалла. Исполняемая программа может находиться только во внутреннем ПЗУ.

Регистры PIC16F628 разделяются на две функциональные группы: специальные и общего назначения. Специальные регистры используются для управления режимами работы функциональных блоков контроллера, регистры общего назначения (память данных или ОЗУ) - для хранения переменных.

Микроконтроллер PIC16F628 использует прямую и косвенную адресацию всех регистров и ячеек памяти. Все специальные регистры и счетчик команд также адресуются как память данных. Ортогональная (симметричная) система команд позволяет выполнять любую операцию с любым регистром, используя любой из названных выше методов адресации. Это облегчает программирование для PIC16F628 и значительно уменьшает время, необходимое на обучение работе с микроконтроллером.

В микроконтроллере PIC16F628 имеется 8-разрядное арифметико-логическое устройство (АЛУ) и рабочий регистр (аккумулятор) W. АЛУ выполняет сложение, вычитание, сдвиг, битовые и логические операции. В командах, обрабатывающих два операнда, один из операндов содержится в рабочем регистре W. Второй операнд может быть константой или содержимым любого регистра ОЗУ. В командах с одним операндом операнд может быть содержимым рабочего регистра или любого регистра ОЗУ. Для выполнения всех операций АЛУ используется рабочий регистр W, который не может быть прямо адресован. Результат операции в АЛУ помещается либо в рабочий регистр W, либо в регистр ОЗУ.

В зависимости от результата выполнения операции изменяются значения битов переноса (C), десятичного переноса (DC) и нулевого результата (Z) в регистре состояния STATUS. При вычитании биты C и DC работают как биты заема и десятичного заема соответственно.

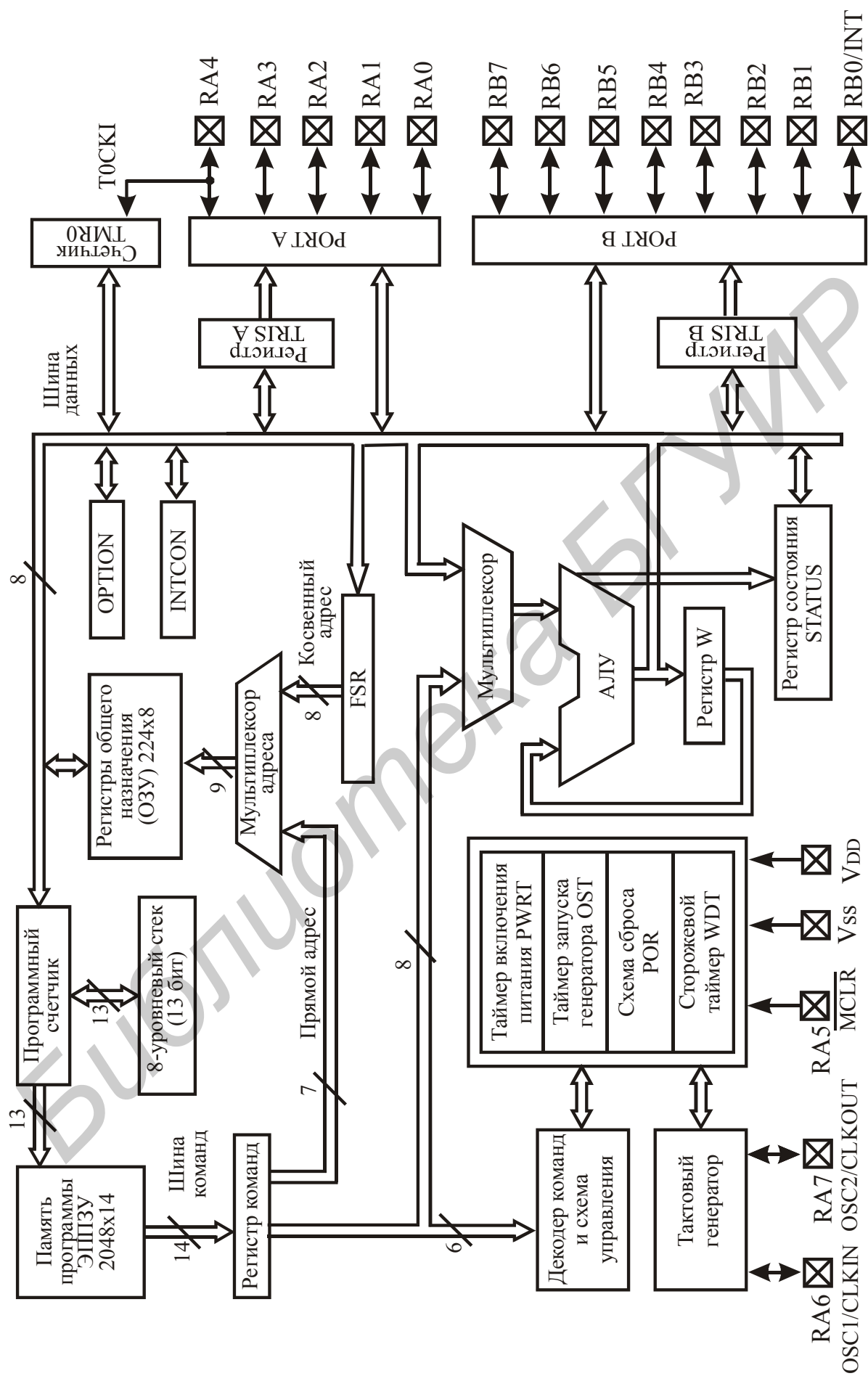


Рис. 1.1.1. Укороченная структурная схема PIC16F628

Таблица 1.1

Обозначение вывода	№ вывода DIP	Тип I/O/P	Тип буфера в режиме ввода	Описание
1	2	3	4	5
RA0/AN0	17	I/O	ST	Двунаправленный порт ввода/вывода, аналоговый вход компаратора
RA1/AN1	18	I/O	ST	Двунаправленный порт ввода/вывода, аналоговый вход компаратора
RA2/AN2/ V _{REF}	1	I/O	ST	Двунаправленный порт ввода/вывода, аналоговый вход компаратора, выход источника опорного напряжения V _{REF}
RA3/AN3/ CPM1	2	I/O	ST	Двунаправленный порт ввода/вывода, аналоговый вход компаратора, выход компаратора
RA4/T0CKI/ CPM2	3	I/O	ST	Двунаправленный порт ввода/вывода (выход с открытым стоком), вход внешнего тактового сигнала для TMR0, выход компаратора
RA5/-MCLR/ THV	4	I	ST	Вход сброса микроконтроллера, вход напряжения программирования. Когда вывод настроен как -MCLR, то по низкому уровню сигнала производится сброс микроконтроллера. При нормальной работе напряжение на выводе не должно превышать V _{DD}
RA6/OSC2/ CLKOUT	15	I/O	ST	Двунаправленный порт ввода/вывода, выход генератора для подключения резонатора. В режиме ER-генератора на выводе формируется сигнал с частотой 1/4 OSC1, обозначая циклы команд
RA7/OSC1/ CLKIN	16	I/O	ST	Двунаправленный порт ввода/вывода, вход генератора для подключения резонатора, вход внешнего тактового сигнала, вывод ER-смещения
RB0/INT	6	I/O	TTL/ST	Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора, вход внешнего прерывания
RB1/RX/DT	7	I/O	TTL/ST	Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора, вход приемника USART, линия данных в синхронном режиме USART
RB2/TX/CK	8	I/O	TTL/ST	Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора, выход передатчика USART, линия тактового сигнала в синхронном режиме USART
RB3/CCP1	9	I/O	TTL/ST	Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора, вывод модуля CCP
RB4/PGM	10	I/O	TTL/ST	Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора. Изменение сигнала на входе может вывести микроконтроллер из режима SLEEP

1	2	3	4	5
RB5	11	I/O	TTL	Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора. Изменение сигнала на входе может вывести микроконтроллер из режима SLEEP
RB6/T1OSO/T1CKI	12	I/O	TTL/ST	Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора. Изменение сигнала на входе может вывести микроконтроллер из режима SLEEP. Выход генератора таймера 1
RB7/T1OSI	13	I/O	TTL/ST	Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора. Изменение сигнала на входе может вывести микроконтроллер из режима SLEEP. Вход генератора таймера 1
V _{SS}	5	P	-	Общий вывод для внутренней логики и портов ввода/вывода
V _{DD}	14	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.

Входная тактовая частота, поступающая с вывода OSC1/CLKIN, внутри делится на четыре, и из нее формируются четыре циклические неперекрывающиеся тактовые последовательности Q1, Q2, Q3 и Q4. Счетчик команд увеличивается в такте Q1, команда считывается из памяти программы и защелкивается в регистре команд в такте Q4. Команда декодируется и выполняется в течение последующего цикла в тактах Q1...Q4. Временные диаграммы тактирования и выполнения команд изображены на рис. 1.2.

Цикл выполнения команды состоит из четырех тактов: Q1...Q4. Выборка команды и ее выполнение совмещены по времени таким образом, что выборка команды занимает один цикл, а выполнение – следующий цикл. Эффективное время выполнения команды составляет один цикл. Если команда изменяет счетчик команд (например команда CALL), то для выполнения этой команды потребуется два цикла. Цикл выборки начинается с увеличения счетчика команд в такте Q1. В цикле выполнения команды выбранная команда защелкива-

ется в регистр команд в такте Q1. В течение тактов Q2, Q3 и Q4 происходит декодирование и выполнение команды. В такте Q2 считывается память данных (чтение операнда), а запись происходит в такте Q4.

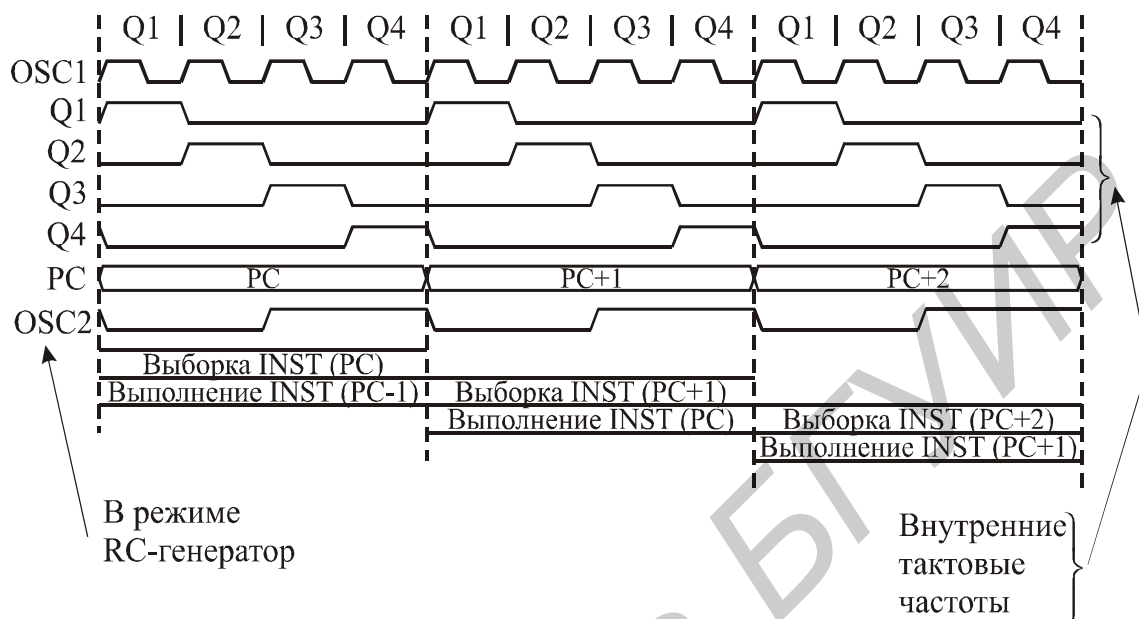


Рис. 1.2. Временные диаграммы тактирования и выполнения команд

1.3. Организация памяти

Внутренняя память в микроконтроллере PIC16F628 состоит из двух частей: памяти программы и памяти данных. Память программы и память данных имеют отдельные шины, поэтому доступ к ним может происходить одновременно. Память данных делится на регистры общего назначения (ОЗУ) и специальные регистры.

1.3.1. Организация памяти программы

Микроконтроллер PIC16F628 имеет 13-разрядный счетчик команд, способный адресовать до $8K \times 14$ слов памяти программы. В PIC16F628 присутствуют только первые $2K \times 14$ (0000-07FFh) слов памяти программы, остальные адреса зарезервированы для будущих модификаций. При сбросе процессор запускается с адреса 0000h, вектор прерывания расположен по адресу 0004h.

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. 1.3 показаны 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).

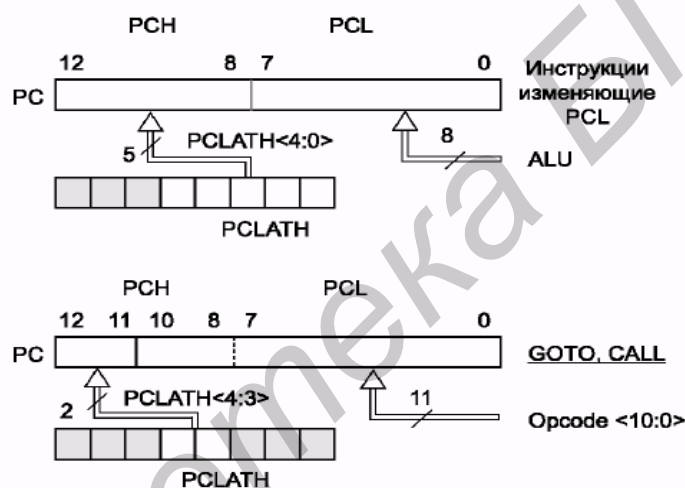


Рис. 1.3. Запись значения в счетчик команд PC

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).

PIC16F628 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 8 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.

1.3.2. Организация памяти данных

Память данных разделяется на две области. Первая представляет собой регистры специальных функций, вторая – регистры общего назначения (ОЗУ). Специальные регистры включают в себя регистр таймера/счетчика (TMR0), счетчик команд (PC), регистр состояния (STATUS), регистры ввода/вывода (PORTA и PORTB), регистр косвенной адресации (FSR) и др. Специальные регистры TRISA и TRISB управляют конфигурацией (направлением передачи) портов ввода/вывода, а OPTION — режимами работы предварительного делителя.

Регистры общего назначения используются для хранения переменных по усмотрению пользователя.

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. 1.2 приведена организация памяти данных PIC16F628.

Все регистры могут быть адресованы прямо или косвенно, с использованием регистра косвенной адресации FSR. Непосредственная адресация поддерживается специальными командами, загружающими данные из памяти программы в рабочий регистр W.

Таблица 1.2

<i>Регистр</i>	<i>Адрес</i>	<i>Регистр</i>	<i>Адрес</i>	<i>Регистр</i>	<i>Адрес</i>	<i>Регистр</i>	<i>Адрес</i>
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
	07h		87h		107h		187h
	08h		88h		108h		188h
	09h		89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch		10Ch		18Ch
	0Dh		8Dh		10Dh		18Dh
TMR1L	0Eh	PCON	8Eh		10Eh		18Eh
TMR1H	0Fh		8Fh		10Fh		18Fh
T1CON	10h		90h		110h		190h
TMR2	11h		91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
	13h		93h		113h		193h
	14h		94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah	EEDATA	9Ah		11Ah		19Ah
	1Bh	EEADR	9Bh		11Bh		19Bh
	1Ch	EECON1	9Ch		11Ch		19Ch
	1Dh	EECON2	9Dh		11Dh		19Dh
	1Eh		9Eh		11Eh		19Eh
CMCON	1Fh	VRCON	9Fh		11Fh		19Fh
	20h		A0h	Регистры общего на- значения 48 байтов	120h		1A0h
Регистры общего на- значения 96 байтов		Регистры об- щего назна- чения 80 байтов			14Fh		
		Доступ к 70h-7Fh	EFh		150h		
			F0h	Доступ к 70h-7Fh	16Fh	Доступ к 70h-7Fh	1EFh
	7Fh		FFh		170h		1F0h
					17Fh		1FFh

Банк 0**Банк 1****Банк 2****Банк 3**

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

1.4. 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. 1.3. Полный список команд приведен в табл. 1.4. Команды разделены на следующие группы: байт-ориентированные команды, бит-ориентированные команды, команды управления и операций с константами.

Таблица 1.3

<i>Поле</i>	Описание
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.

Таблица 1.4

Мнемоника команды	Описание	Кол-во циклов	Изм. флаги	Прим.
<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		
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	GIE	
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, ес-

ли 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.

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.

1.5. Регистр состояния STATUS

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

Структура регистра отражена в табл. 1.5. В регистре STATUS содержатся флаги состояния АЛУ, флаги причины сброса микроконтроллера и биты управления банками памяти данных.

Таблица 1.5

Номер бита	Имя бита	Доступ и состояние после сброса	Назначение
1	2	3	4
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

1	2	3	4
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

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 CLRf STATUS сбросит три старших бита и установит бит Z (состояние регистра STATUS после выполнения команды $000uu1uu$, 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.

1.6. Регистр OPTION_REG

Регистр OPTION_REG 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. 1.6.

Таблица 1.6

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	T0CS	R/W-1	Выбор тактового сигнала для TMR0: 1 внешний тактовый сигнал с вывода RA4/T0CKI, 0 внутренний тактовый сигнал CLKOUT																											
4	T0SE	R/W-1	Выбор фронта приращения TMR0 при внешнем тактовом сигнале: 1 приращение по срезу сигнала (с высокого к низкому уровню) на выводе RA4/T0CKI, 0 приращение по фронту сигнала (с низкого к высокому уровню) на выводе RA4/T0CKI																											
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																												

1.7. 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. 1.7.

Таблица 1.7

Номер бита	Имя бита	Доступ и состояние после сброса	Назначение
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

1.8. Косвенная адресация данных

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 битов косвенного адреса IRP сохраняется в регистре STATUS<7>. Пример 9-разрядной косвенной адресации показан на рис. 1.4.

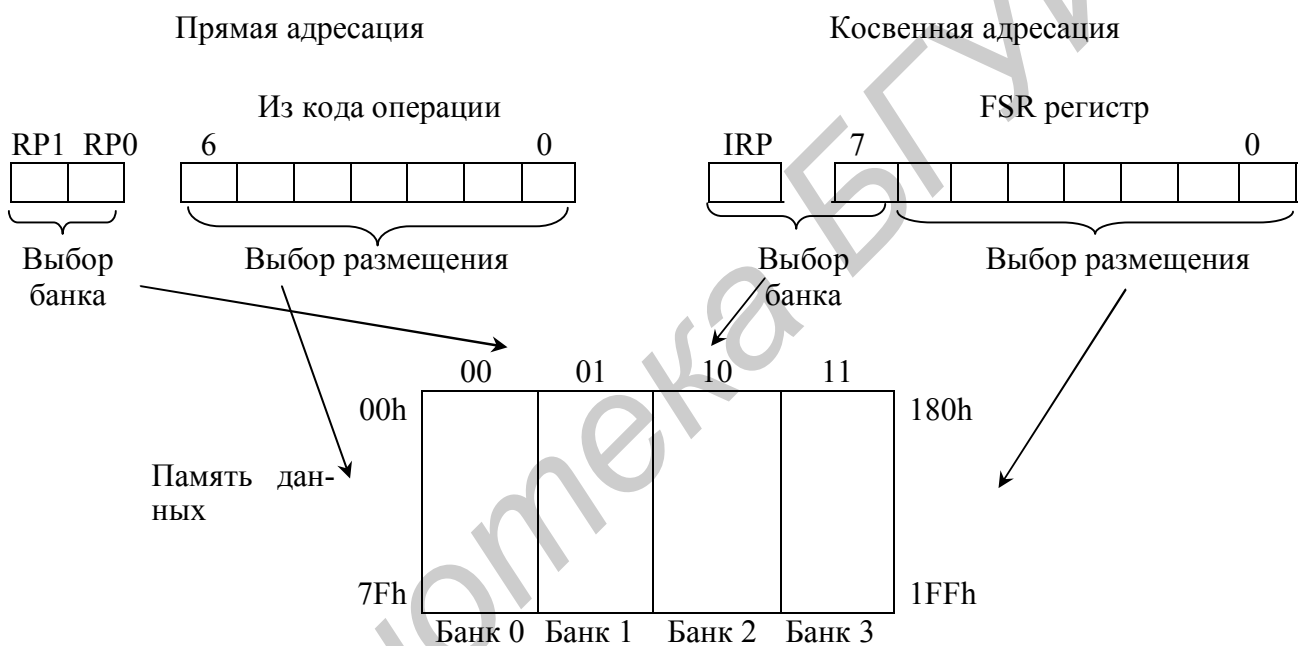


Рис. 1.4. Схема прямой и косвенной адресации в PIC16F628

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:
CLRFB   INDF        ; Очистить регистр
INCF    FSR, F      ; Увеличить адрес
BTFSS   FSR, 4     ; Завершить?
GOTO    NEXT       ; Нет, продолжить очистку
CONTINUE:
                ; Да
    
```

1.9. Прерывания

PIC16F628 имеет 10 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 и PIE1. 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 регистре INTCON находится бит разрешения прерываний от периферийных модулей. Упрощенная структурная схема логики прерываний (без прерываний от периферийных модулей) показана на рис. 1.5.

При переходе на подпрограмму обработки прерываний бит GIE аппаратно сбрасывается в '0', запрещая прерывания, текущее значение счетчика команд (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 одного и двух 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.

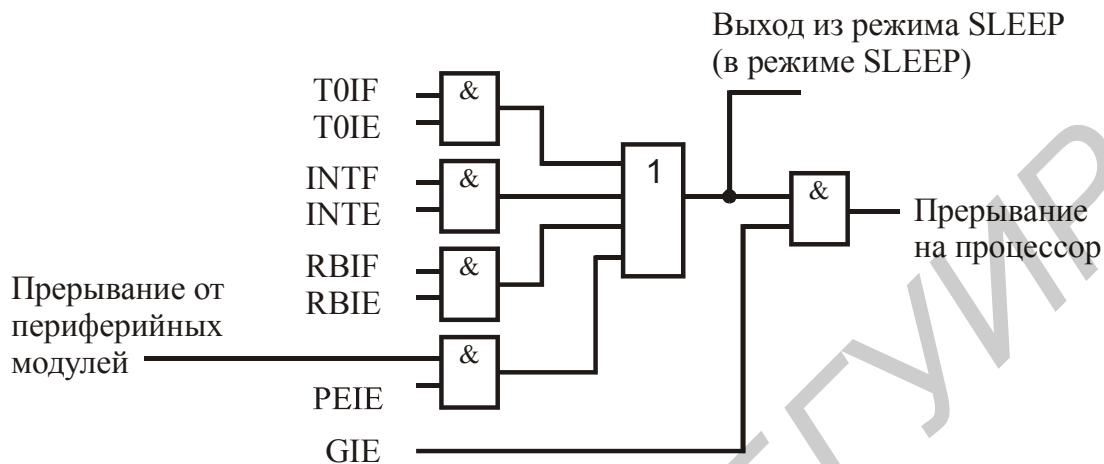


Рис. 1.5.. Логика прерываний

1.9.1. 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.

1.9.2. 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>).

1.9.3. 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>).

1.9.4. 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 PIC16F628 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.

1.10. Порты ввода/вывода

Микроконтроллеры PIC16F628 имеют два порта ввода/вывода, PORTA и PORTB. Некоторые выходы портов мультиплексированы с периферийными модулями микроконтроллера. Когда периферийный модуль включен, вывод не может использоваться как универсальный канал ввода/вывода.

Программа может считывать и записывать данные в регистры ввода/вывода аналогично регистрам общего назначения. При чтении всегда считывается действительное состояние выводов, независимо от того, запрограммированы отдельные биты как входы или как выходы. После сброса все разряды программируются как входы (выводы находятся в высокоимпедансном состоянии), поскольку регистры управления портами TRISA и TRISB устанавливаются в '1'. Разряд порта ввода-вывода определяется как выход, если соответствующий бит в регистре управления портом установлен в '0'.

1.10.1. PORTA

Регистр ввода-вывода PORTA имеет разрядность 8 битов. Разряд RA4 имеет вход с триггером Шмитта и выход с открытым стоком. Он объединен с входом таймера TMR0. Все остальные разряды PORTA имеют триггеры Шмитта на входах и выходные КМОП-буферы.

Описание выводов регистра PORTA приведено в табл. 1.1. Упрощенная функциональная организация разрядов RA0...RA3 регистра показана на рис. 1.6, а разряда RA4 – на рис.1.7.

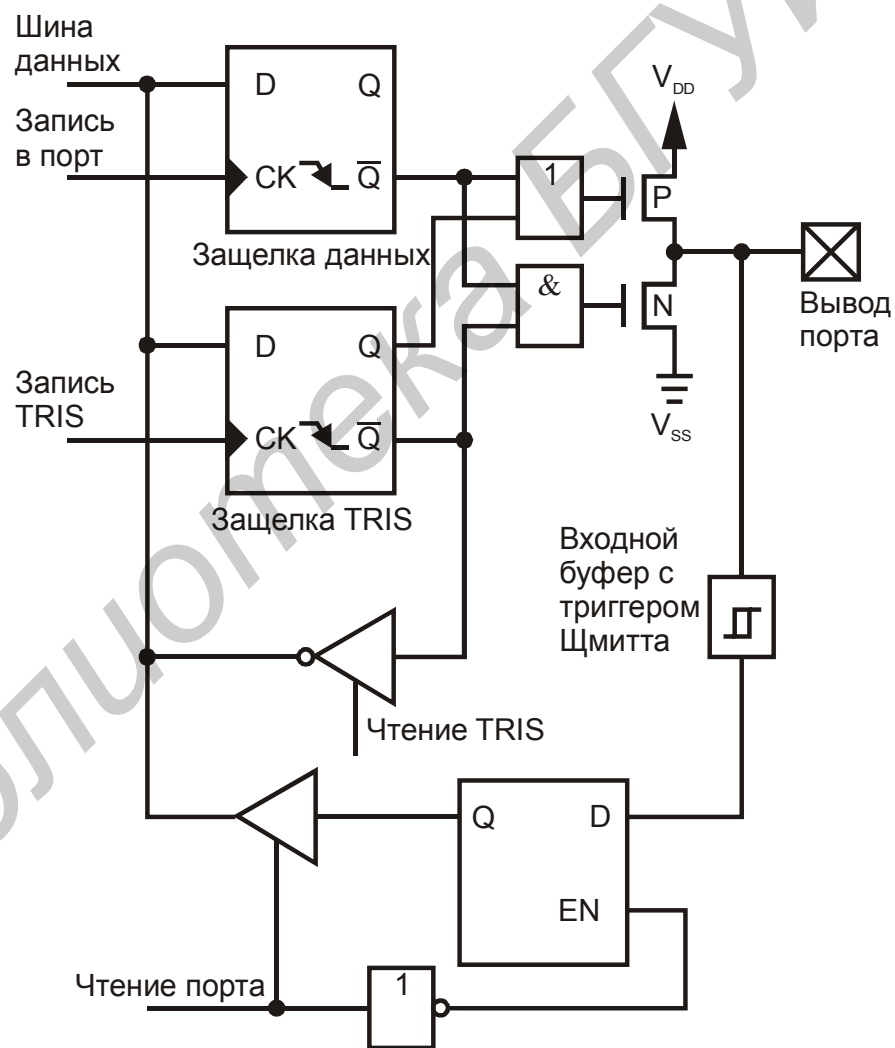
Ниже приведем пример программы инициализации (настройки) PORTA.

CLRF	PORTA	;Очистка (сброс в «0») выходных защёлок PORTA.
BSF	STATUS,RP0	;Выбор банка 1.
MOVLW	0x 0F	;Значение константы для выбора режимов работы
		;разрядов.
MOVWF	TRISA	;Установить RA<3:0> как входы и RA4 как
		;выход.
BCF	STATUS,RP0	;Выбор банка 0.

1.10.2. PORTB

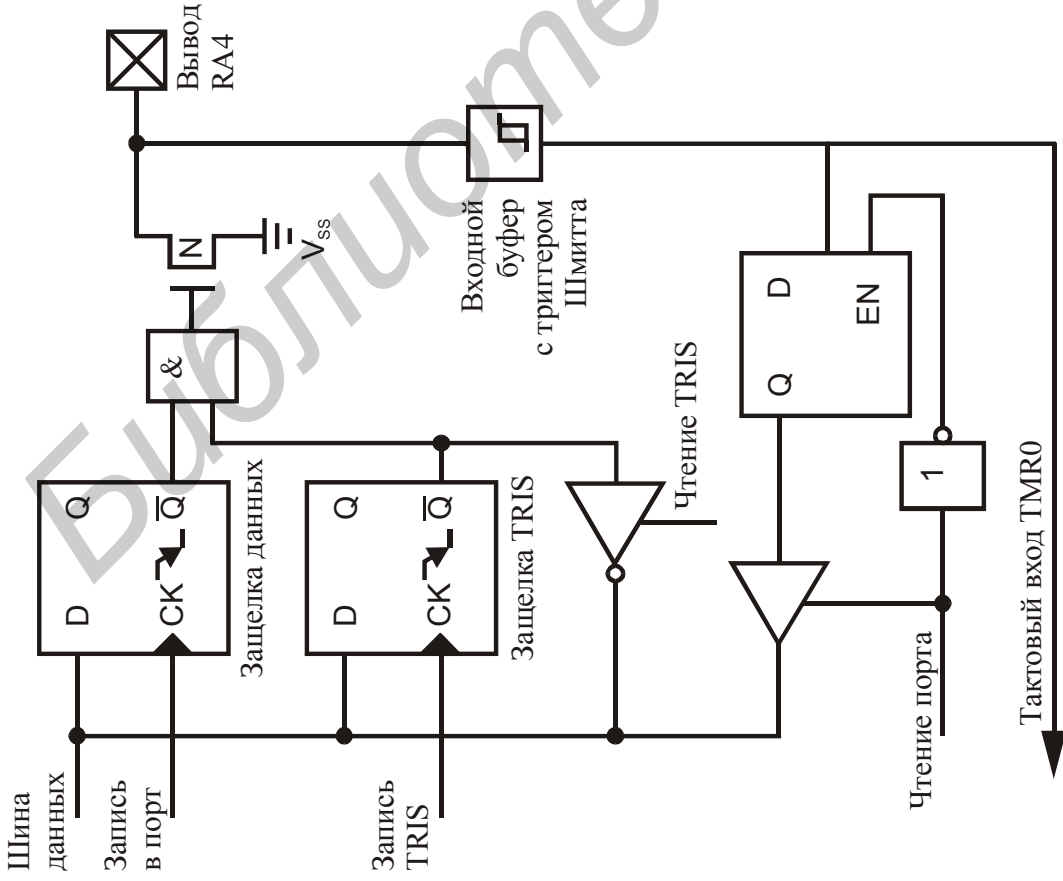
Регистр ввода/вывода PORTB 8-разрядный. Все разряды PORTB имеют внутренние подтягивающие резисторы, которые могут быть включены установкой в '0' бита RBPU (OPTION). Подтягивающие резисторы автоматически отключаются, если соответствующий разряд программируется как выход. По включении питания подтягивающие резисторы отключаются.

Упрощенная функциональная организация выводов RB7...RB4 PORTB показана на рис. 1.8, а выводов RB3...RB0 – на рис.1.9.



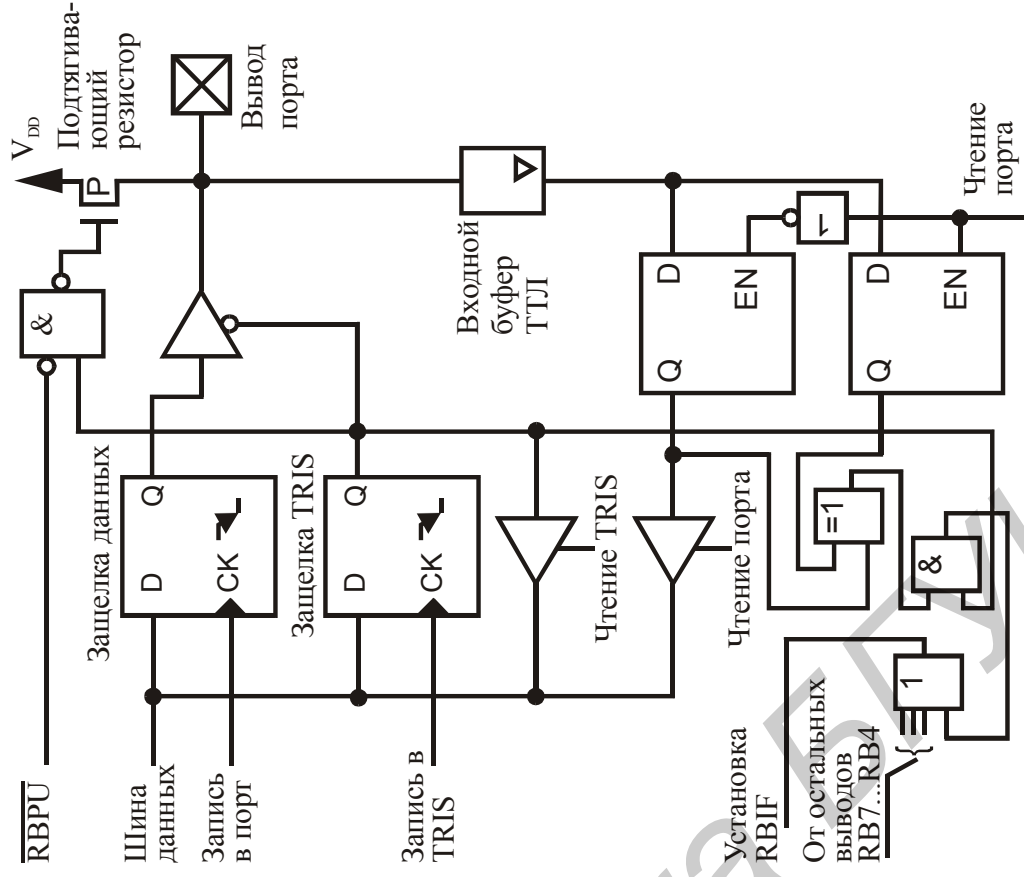
Примечание. Все выводы имеют защитные диоды на V_{DD} и V_{SS}.

Рис. 1.6.. Функциональная организация выводов RA0...RA3 регистра PORTA



Примечание. Вывод имеет защитный диод только на V_{SS} .

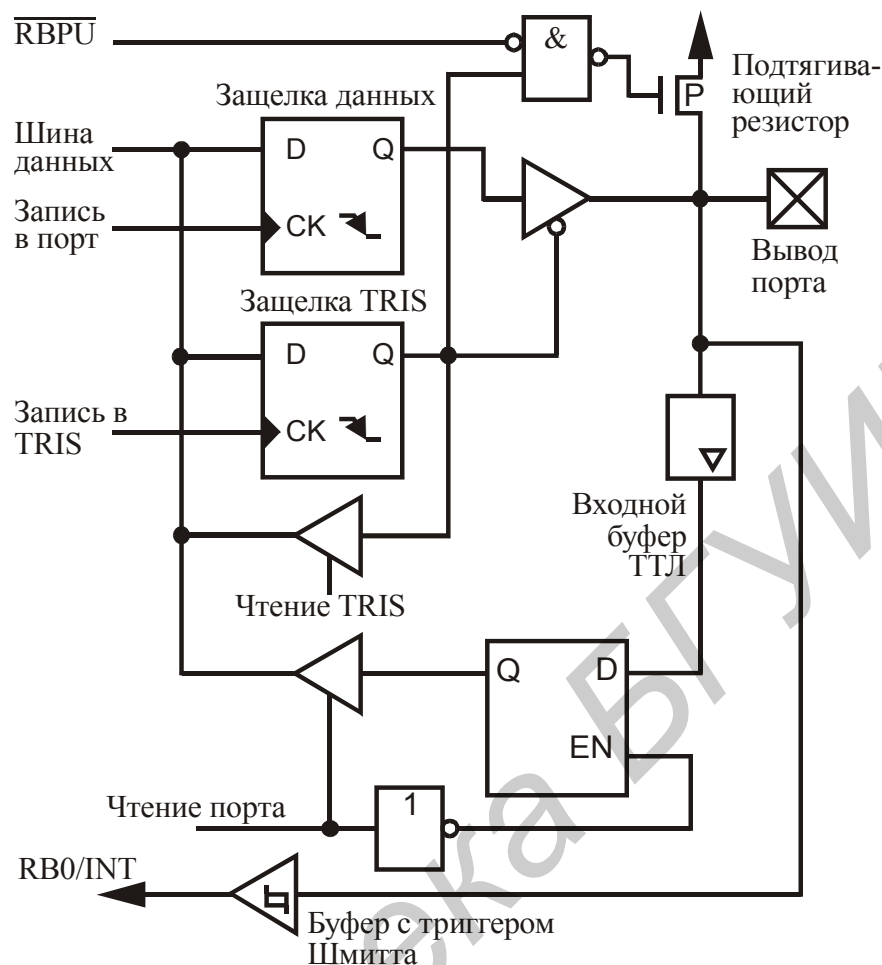
Рис. 1.7. Функциональная организация вывода RA4 регистра PORTA



Примечания:

1. Бит $TRISB=1$ разрешает подтягивающий резистор, если в регистре $OPTION\ RBPU=0$.
2. Все выходы имеют защитные диоды на V_{DD} и V_{SS} .

Рис. 1.8. Функциональная организация выводов RB7...RB4



Примечания:

1. Бит $TRISB=1$ разрешает подтягивающий резистор, если в регистре $OPTION\ RBPUS=0$.
2. Все выходы имеют защитные диоды на V_{DD} и V_{SS} .

Рис. 1.9. Функциональная организация выводов $RB3...RB0$

Имеется возможность прерывания по изменению состояния четырех разрядов $PORTB$ (выводы $RB7...RB4$). Прерывание может возникнуть только от тех разрядов $PORTB<7:4>$, которые запрограммированы как входы; выходы не включаются в процедуру сравнения. Текущее состояние разрядов $PORTB<7:4>$, запрограммированных как входы, сравнивается с состоянием, зашелкнутым в регистр $PORTB$ при последнем считывании. При несовпадении возникает прерывание по изменению состояния. Это прерывание может вывести микрокон-

троллер из режима пониженного энергопотребления SLEEP. Для сброса прерывания в подпрограмме обработки необходимо выполнить следующие действия:

- 1) считать PORTB (это сбросит условие несовпадения);
- 2) сбросить флаг RBIF.

Прерывание по несовпадению совместно с программно управляемыми подтягивающими резисторами позволяет легко реализовать интерфейс клавиатуры и обеспечить выход из режима пониженного энергопотребления по нажатию клавиши. Для возникновения прерывания по изменению состояния минимальная длительность импульса должна быть не менее длительности цикла команды.

Ниже приведем пример инициализации (настройки) PORTB.

CLRF	PORTB	;Обнуление выходных регистров PORTB.
BSF	STATUS,RP0	;Выбор банка 1.
MOVLW	0xCF	;Значение для задания направления.
MOVWF	TRISB	;Установить RB<3:0> как входы, RB<5:4> ; как выходы и RB<7:6> как входы

1.11. Особенности программирования портов

1.11.1. Организация двунаправленных портов

Некоторые команды выполняются в режиме «чтение — модификация — запись».

Например, команды BCF и BSF считывают содержимое порта целиком, модифицируют один бит и выводят результат обратно. При использовании этих команд с портами, в которых некоторые разряды запрограммированы как входы, а некоторые как выходы, необходима осторожность. Например, команда BSF для пятого бита регистра PORTB сначала считывает все восемь битов, затем выполняется установка пятого бита и новое значение байта целиком записывается в выходную защелку. Если другой бит регистра PORTB используется в качестве двунаправленного входа/выхода (скажем, бит 0) и в данный момент он определен как вход, входной сигнал на этом выводе будет считан и записан обратно в выходную защелку этого же вывода, затирая ее предыдущее состоя-

ние. До тех пор пока этот вывод остается в режиме входа, никаких проблем не возникает. Однако, если позднее линия 0 переключится в режим выхода, ее состояние будет неопределенным. Команда считывания порта считывает состояние вывода, а не выходных регистров. Например, если разряд порта запрограммирован как выход и установлен в '1', но внешняя схема поддерживает низкий уровень на выводе, порт будет считываться как '0'. На вывод, работающий в режиме выхода, не должны подключаться внешние нагрузки по схеме "монтажное И" либо "монтажное ИЛИ". Возникающие при этом большие токи могут повредить кристалл.

1.11.2. Обращение к портам ввода/вывода

Запись в порт вывода происходит в конце цикла выполнения команды. При чтении данные должны быть стабильны в начале цикла выполнения команды. Надо быть внимательным при операциях чтения, следующих сразу же за записью в тот же порт. Здесь надо учитывать инерционность установления напряжения на выводах. Может потребоваться программная задержка, чтобы напряжение на выводе успело стабилизироваться до начала исполнения следующей команды чтения. Время установления напряжения на выводе зависит от подключенной к нему нагрузки и может меняться в широких пределах.

Ниже рассмотрен пример выполнения операции «чтение — модификация — запись» с портом ввода/вывода.

```

;Начальные установки порта: PORTB<7...4> - входы,
;                               PORTB<3...0> - выходы.
;Разряды PORTB<7...6> имеют внешние нагрузки.
;
;                               Защелка PORTB  Выводы PORTB
BCF  PORTB,7                    ;01pp rppp      11pp rppp
BCF  PORTB,6                    ;10pp rppp      11pp rppp
BSF  STATUS,RP0 ;
BCF  TRISB,7                    ;10pp rppp      11pp rppp
BCF  TRISB,6                    ;10pp rppp      10pp rppp
; Примечание: пользователь мог бы ожидать, что результирующее значение будет
; 00pp rppp.
; Однако вторая команда BCF вызывает защелкивание в RB7 высокого уровня в
; соответствии с состоянием вывода RB7.

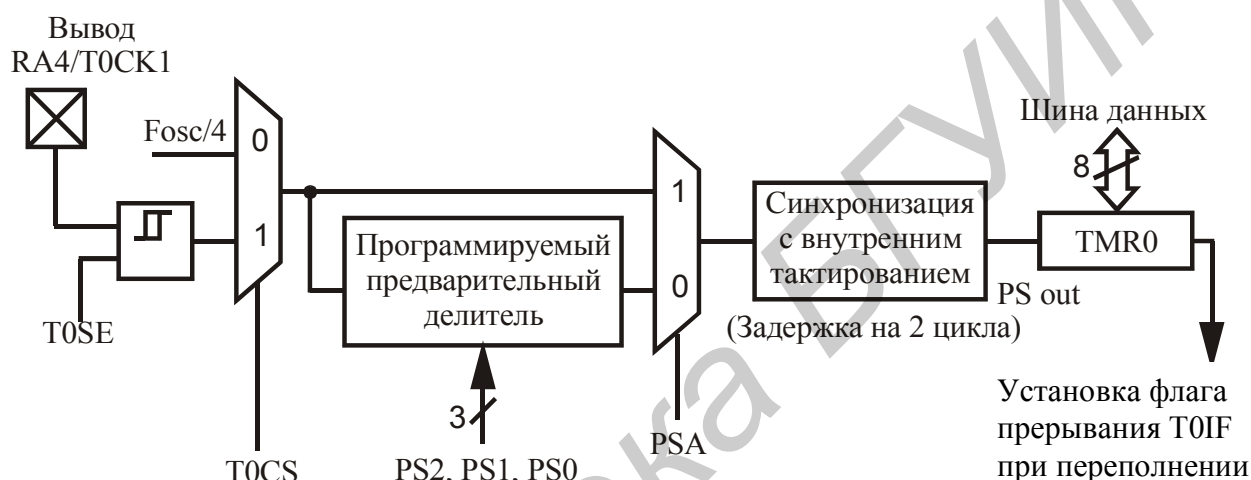
```

1.12. Модуль таймера TMR0

Модуль таймера TMR0 имеет следующие особенности:

- 8-разрядный таймер/счетчик, доступен по чтению и записи;
- 8-разрядный программируемый предварительный делитель;
- внутреннее или внешнее тактирование;
- прерывание по переполнению счетчика (переход от 0FFh к 00h);
- выбор фронта тактирующего импульса при внешнем тактировании.

Упрощенная структурная схема модуля таймера приведена на рис. 1.10.



Примечания:

1. Биты T0CS, T0SE, PSA, PS2, PS1 и PS0 находятся в регистре OPTION.
2. Предварительный делитель используется совместно со сторожевым таймером.

Рис. 1.10. Структурная схема таймера

Режим **таймера** выбирается установкой в '0' бита T0CS (OPTION<5>). В режиме таймера TMR0 увеличивается в каждом командном цикле (в отсутствие предварительного делителя). Если происходит запись в TMR0, то увеличение счетчика задерживается на два последующих цикла выполнения команды. Запись в TMR0 должна вестись с учетом этой задержки. При необходимости проверки регистра TMR0 на ноль без влияния на процесс счета рекомендуется пользоваться командой **MOVF TMR0,W**.

Режим **счетчика** выбирается установкой в '1' бита T0CS (OPTION<5>). В этом режиме TMR0 увеличивается по каждому перепаду 1/0 или 0/1 на выводе T0CKI. Перепад, увеличивающий значение TMR0, выбирается битом выбора

фронта переключения T0SE (OPTION<4>). Установка этого бита в '0' вызывает увеличение TMR0 по перепаду 0/1.

Предварительный делитель может использоваться модулем сторожевого таймера WDT или модулем таймера. Подключение предварительного делителя задается битом PSA (OPTION<3>). Установка бита PSA в '1' подключает предварительный делитель к модулю WDT и устанавливает коэффициент деления для TMR0 1:1. Установка бита PSA в '0' подключает предварительный делитель к модулю таймера. Коэффициент деления предварительного делителя может быть установлен битами PS0-PS2 регистра OPTION_REG. Сам предварительный делитель недоступен для чтения и записи.

1.12.1. Прерывание от таймера

Прерывание от TMR0 вырабатывается при переполнении счетчика (переходе от 0FFh к 00h). При переполнении устанавливается в '1' бит T0IF (INTCON<2>). Прерывание может быть замаскировано установкой в '0' бита T0IE (INTCON<5>). Бит T0IF должен быть сброшен в '0' в процедуре обработки прерывания от TMR0 до того, как прерывания снова будут разрешены. Прерывание от TMR0 не может вывести микроконтроллер из режима пониженного энергопотребления SLEEP, поскольку в режиме SLEEP таймер TMR0 выключен.

1.12.2. Использование TMR0 с внешним сигналом

Если для тактирования TMR0 используется внешний сигнал, то он должен удовлетворять определенным требованиям для синхронизации с внутренней тактовой частотой. Кроме того, между перепадом на выводе T0CKI и реальным увеличением счетчика TMR0 есть некоторая задержка.

Если предварительный делитель не используется, внешний тактовый сигнал на входе T0CKI должен сохранять как высокий, так и низкий уровень в течение не менее двух периодов тактового генератора.

Когда используется предварительный делитель, входной сигнал TMR0 делится асинхронным счетчиком предварительного делителя, поэтому выходной сигнал делителя является симметричным. Период сигнала на входе TMR0 должен быть не менее четырех периодов тактового генератора. Сигнал же на входе T0СКІ должен иметь высокие и низкие уровни длительностями не менее 10 нс.

Так как выход предварительного делителя синхронизирован с внутренней тактовой частотой, то возможна небольшая задержка между перепадом сигнала на выводе T0СКІ и моментом увеличения содержимого TMR0.

1.12.3. Предварительный делитель

Встроенный 8-разрядный счетчик может использоваться как предварительный делитель для TMR0 или как дополнительный делитель для сторожевого таймера WDT. Необходимо учесть, что делитель может быть использован либо с TMR0, либо со сторожевым таймером WDT, но не одновременно.

Схема использования предварительного делителя отражена на рис 1.11.

Биты PSA и PS0-PS2 в регистре OPTION<3:0> задают режим использования предварительного делителя и его коэффициент деления.

Когда предварительный делитель используется с TMR0, все команды, производящие запись в регистр TMR0 (например, **CLRF TMR0**, **MOVWF TMR0**, **BSF TMR0,b** и т.д.), обнуляют предварительный делитель.

Когда предварительный делитель используется со сторожевым таймером WDT, команда **CLRWDT** очищает предварительный делитель одновременно со сбросом сторожевого таймера WDT. Предварительный делитель не может быть считан или записан программно. По сбросу предварительный делитель содержит все '0'.

Назначение предварительного делителя задается программно и может быть изменено в процессе выполнения программы. Чтобы избежать непредусмотренного сброса контроллера, при **переключении** предварительного дели-

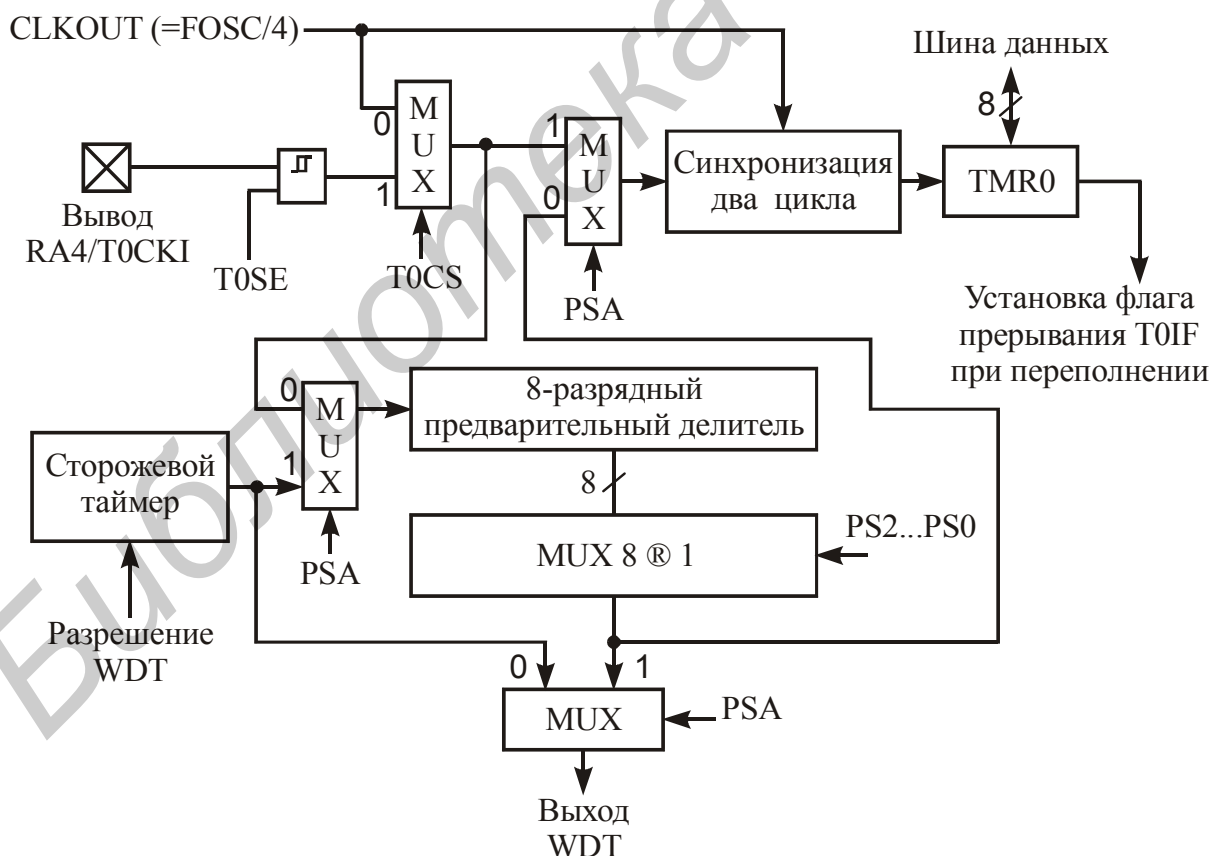
теля с TMR0 на WDT должна быть выполнена такая последовательность команд:

```
BCF      STATUS,RP0      ;Установка банка 0.
CLRF    TMR0             ;Сброс TMR0.
BSF     STATUS,RP0      ;Установка банка 1.
CLRWDT                      ;Сброс WDT и предварительного делителя.
MOVLW   b'xxxx1xxx'     ;Укажите новое значение предварительного
MOVWF   OPTION           ; делителя.
BCF     STATUS,RP0      ; Установка банка 0.
```

Для переключения предварительного делителя с WDT на TMR0 должна быть выполнена последовательность команд:

```
CLRWDT                      ;Сброс WDT и предварительного делителя.
BSF     STATUS,RP0      ;Установка банка 1.
MOVLW   b'xxxx0xxx'     ;Указать новое значение предварительного делителя,
MOVWF   OPTION           ;источник тактирования и фронт переключения TMR0.
BCF     STATUS,RP0      ;Установка банка 0.
```

Эта последовательность должна быть выполнена даже в том случае, если сторожевой таймер WDT запрещен.



Примечание. Биты T0CS, T0SE, PSA, PS2...PS0 - биты регистра OPTION <6...0>.

Рис. 1.11. Схема использования предварительного делителя

1.13. Лабораторное задание

1. Изучить структурную схему, систему команд, организацию памяти программ и данных, организацию прерываний, назначение и состав регистров STATUS, OPTION_REG и INTCON микроконтроллера PIC16F628.

2. Загрузить на персональном компьютере интегрированную среду MPLab IDE разработки программ для микроконтроллеров PICmicro.

3. Руководствуясь указаниями инструкции по работе с MPLab IDE [2], ввести, скомпилировать, устранить ошибки и загрузить в симулятор программу, приведенную в п. 1.4 этой инструкции.

4. Провести моделирование работы программы в пошаговом режиме. При этом обратить внимание на изменение состояний программного счетчика и ячейки памяти, хранящей переменную C1.

5. Подготовить отчет по работе, который должен содержать: титульный лист, структурную схему ядра микроконтроллера PIC16F628, систему команд, организацию памяти данных, составы регистров STATUS, OPTION_REG и INTCON.

1.14. Контрольные вопросы

1. Назначение памяти программ и памяти данных в компьютерах. Различия гарвардской и принстонской (фон-неймановской) архитектур компьютеров.

2. Поясните назначение программного счетчика, стека, регистра команд, декодера команд, арифметико-логического устройства, рабочего регистра W и регистра состояния STATUS.

3. Поясните, как изменяется состояние программного счетчика в процессе выполнения линейного участка программы.

4. Как изменяется состояние программного счетчика в процессе выполнения команды перехода GOTO?

5. Как изменяются состояния программного счетчика, регистра команды и стека в процессе выполнения команд вызова подпрограммы CALL и возврата из подпрограммы RETURN?
6. Поясните назначение и функции регистров OPTION_REG и INTCON.
7. Прокомментируйте действия, выполняемые микроконтроллером по каждой из его команд.

ЛИТЕРАТУРА

1. PIC16F62X. Однокристалльные 8-разрядные FLASH CMOS микроконтроллеры компании Microchip technology incorporated: Пер. с англ. –М.: ООО «Микрочип», 2001. – 148 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

ЦИФРОВЫЕ И МИКРОПРОЦЕССОРНЫЕ УСТРОЙСТВА

Лабораторный практикум
для студентов специальностей
I-39 01 02 «Радиоэлектронные системы»,
I-39 01 03 «Радиоинформатика»,
I-39 01 04 «Радиоэлектронная защита информации»
В 2-х частях
Часть 1

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

Подписано в печать 19.05.05.
Гарнитура «Таймс».
Уч.-изд. л. 1,7.

Формат 60x84 1/16.
Печать ризографическая.
Тираж 200 экз.

Бумага офсетная.
Усл. печ. л.
Заказ 159.

Издатель и полиграфическое исполнение: Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
Лицензия на осуществление издательской деятельности №02330/0131518 от 01.04.2004.
Лицензия на осуществление полиграфической деятельности №02330/0133108 от 30.04.2004.
220013, Минск, П. Бровки, 6