

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
ДГТУ

Кафедра «Приборостроение»

РЕГИСТРЫ СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ  
МИКРОКОНТРОЛЛЕРА PIC16F628A

Методические указания к лабораторной работе по дисциплинам  
«Автоматизированные системы сбора и обработки измерительной информации», «Микропроцессорная техника»,  
«Микропроцессорные системы»,  
«Микропроцессорные измерительные системы»

Ростов-на-Дону 2012

Составители: к.т.н., доц. Литвин А.В.  
к.т.н., доц. Мороз К.А.  
ст. преподаватель Нестеренко И.Н.  
ст. преподаватель Сыроватка В.Н.

УДК 004.382.7

Регистры специального назначения микроконтроллера PIC16F628A.

Метод. указания к лабораторной работе по дисциплинам «Автоматизированные системы сбора и обработки измерительной информации», «Микропроцессорная техника», «Микропроцессорные системы», «Микропроцессорные измерительные системы» – Ростов н/Д: Издательский центр ДГТУ. 2012. – 20 с.

В методических указаниях приводятся цель работы, краткое описание порядка составления программы, содержание и порядок выполнения лабораторной работы, рассмотрен пример написания программы для микроконтроллера. Предназначены для студентов очной и заочной форм обучения специальностей «Приборостроение», «Стандартизация и сертификация», «Метрология и метрологическое обеспечение», а также направлений «Биомедицинская инженерия» и «Приборостроение».

Печатается по решению методической комиссии факультета «Приборостроение и техническое регулирование»

Рецензент к.т.н., доцент П. С. Обухов

Научный редактор к.т.н., проф. В.Н. Ананченко

© Издательский центр ДГТУ, 2012

## 1. ЦЕЛЬ РАБОТЫ

Изучение регистров специального назначения (РСН) микроконтроллера (МК) PIC16F628A. Приобретение практических навыков по управлению работой МК путем конфигурирования РСН.

## 2. КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ

Регистры специального назначения (РСН) – это специально зарезервированные ячейки памяти с фиксированными адресами и именами, обращаясь к которым можно настраивать режимы работы МК и его модулей. Собственно, все управление МК сводится к изменению состояния тех или иных битов в РСН.

РСН МК PIC16F628A расположены в 0,1,2,3 банках памяти (см. Приложение Б) по адресам 00H – 20H, 80H – 9FH, 100H – 110H, 180H – 190H. Закрашенные участки памяти данных не реализованы. Часть РСН представлены во всех банках, менять их содержимое можно, находясь в любом банке, при этом их значение отображается и во всех остальных банках. Это регистры INDF, PCL, STATUS, FSR, PCLATH, INTCON. Остальные РСН представлены только в одном из банков памяти, и для работы с их содержимым необходимо перейти в тот банк, где они расположены. Например: так как большая часть работы МК происходит в банке 0, то при обращении к регистрам банка 1 (OPTION, TRISA, и т.д.) необходимо сначала перейти в банк 1, произвести действие с их содержимым, а затем вернуться обратно в банк 0.

**Регистр STATUS.** (Приложение В). Расположен по адресам 03H, 83H, 103H, 183H. Выбор банка памяти осуществляется битами 5,6,7, если не используется косвенная адресация – битами 5 и 6. Поскольку основная роль в работе МК отводится банкам 0 и 1, то 6 бит можно сразу установить в ноль, а переключение между банками 0 и 1 осуществлять изменением бита 5. По умолчанию он установлен в 0 и МК работает в банке 0. Меняя бит 5 на единицу, переходим в банк 1.

Для переключения банков в программе написанной на языке ассемблера обычно используют команды bcf и bsf, которые позволяют изменить какой-либо один бит в регистре, сохранив неизменными все остальные биты. Данные команды бит ориентированные, о чем свидетельствует буква b в начале команды.

---

Формат команды:

bcf f,b – очистить (уст. в 0) содержимое бита номер b в регистре по имени f.

bsf f,b - установить в 1 бит номер b в регистре по имени f.

Пример перехода из нулевого банка в первый и обратно:

bcf STATUS,6 ; установка бита 6 в ноль, выбор банков 0 и 1.

bsf STATUS,5 ; установка бита 5 в единицу, выбор банка 1.

XXXXXXX ; команды изменяющие содержимое регистров банка 1.

bcf STATUS,5 ; установка бита 5 в ноль, выбор банка 0.

---

Остальные биты регистра STATUS являются флагами. Изменение флагов не влияет на работу МК. Этим они отличаются от рабочих битов, изменение которых влияет на работу МК. Флаги выполняют информационную функцию. Проверая тот или иной флаг можно узнать результат операции или факт наступления того или иного события. Наиболее часто обращаются к содержимому флагов Z (2 бит регистра STATUS) и C (0 бит регистра STATUS). Флаг нулевого результата - Z информирует, был ли результат операции нулевым, если это так, то флаг устанавливается в 1. Флаг переноса – заема - C показывает, был ли перенос из старшего бита. Проверка флага C дает возможность узнать, было ли переполнение регистра после выполнения операции, например, сложения.

Пример сложения чисел 254 и 3.

$$254_{10} = 11111110_2$$

Результат операции вышел за пределы разрядной емкости регистра в 8 бит, и единица помещается во флаг C, который служит «временным 9м разрядом регистра». Перед началом использования флага рекомендуется привести его в однозначное исходное состояние. Например, с помощью команды `bcf STATUS,0`.

$$+ \quad 3_{10} = 00000011_2$$

$$257_{10} = 100000001_2$$

Также, флаг C задействуется в операциях «сдвига», в которых он используется для временного хранения состояния седьмого или нулевого бита сдвигаемого регистра.

**Регистры PortA, TrisA и PortB, TrisB.** Для работы с портами ввода-вывода существует четыре РСН, по два на каждый порт. Порты А и В содержат по 8 линий, поэтому их регистры тоже 8-ми разрядные. Линии портов могут работать как на ввод, так и на вывод информации из МК. Также, любую линию можно конфигурировать отдельно, получая, таким образом, требуемую комбинацию линий, работающих на прием или выдачу сигнала. В процессе работы программы также возможно изменение направления работы линий портов. Это бывает необходимо, когда выводы используются в одном устройстве для нескольких различных задач.

За направление работы выводов портов отвечают регистры **TrisA** (85H) и **TrisB** (86H, 106H, 186H). Они находятся в банке 1, и для работы с их содержимым необходимо перейти в банк 1. Если в какой-либо разряд этих регистров заносится единица, то данный вывод порта настраивается на вход, ввод сигнала с вывода порта в МК. Если заносится 0 – то данный вывод работает на выход, вывод сигнала из МК. При этом, МК является источником цифрового сигнала ТТЛ уровня ('0' = 0В, '1' = 5В). За каждым выводом порта закреплен свой бит в регистре. Например: бит 0 регистра TrisA управляет выводом порта под названием RA0, а бит 7 регистра TrisB – выводом порта RB7.

Изменять содержимое регистров TrisA и TrisB можно разными способами. Например, с помощью команд `bcf` и `bsf`. Эти команды особенно удобны для изменения режима работы отдельных линий, при сохранении режимов остальных. Если же необходимо работать со многими линиями, или со всем портом сразу, удобнее использовать команду `mov`. Работа этой команды тесно связана с

работой аккумулятора W. Например, если необходимо в TrisB записать 0, и настроить все линии порта B на выход, используется команда `movwf TrisB`, которая копирует данные из W в регистр TrisB. Но, сначала, необходимо данные, тот же ноль, занести в W. Сделать это можно с помощью команды `movlw .0`, которая копирует константу 0 в аккумулятор W.

---

Формат команды:

`movwf f` - копировать содержимое W в регистр с названием f.

`movlw k` – копировать константу k в аккумулятор (W).

Корень команды `mov` означает – копировать. Буквосочетания WF – копировать содержимое аккумулятора W в регистр. Буквосочетания LW – копировать константу (число) в аккумулятор.

Пример настройки вывода RB1 порта B на вход, остальных выводов на выход:

`movlw .2` ; копирование константы 2 в аккумулятор W

`movwf TrisB` ; копирование содержимого аккумулятора W в регистр TrisB

---

- \* О форме представления чисел при программировании на языке ассемблера. Наиболее часто используют представление чисел в десятичной, шестнадцатеричной и двоичной форме исчисления. Точка перед числом – признак использования числа в десятичной системе счисления. В примере - .2 или d'2'. При использовании шестнадцатеричной системы запись была бы вида – h'02' или 0x02, а двоичной – b'00000010'.

Если регистры TrisA и TrisB отвечают за настройку направления работы линий портов, то регистры **PortA** (05H) и **PortB** (06H,106H) за их состояние. Выводы портов непосредственно связаны с регистрами PortA и PortB. Если поменять какой-либо бит регистра с 0 на 1, то на соответствующей линии порта изменится напряжение с 0В до 5В. (При условии что порт ранее был настроен на выход). Таким образом, копируя в PCN PortA и PortB те или иные числа, можно достигать заданного сочетания нулей и единиц на линиях портов. Достоинством этих регистров (ячеек памяти) является то, что записанное число может в них оставаться сколь угодно долго (пока включено питание), при этом будут сохраняться и состояния линий портов. Это рационально, поскольку в это время центральный процессор МК может исполнять любые команды, не поддерживая состояние портов. Данное свойство портов незаменимо при организации индикации на семисегментных индикаторах.

Менять содержимое регистров **PortA** и **PortB** можно всегда, но не всегда это приведет к изменению сигналов на линиях портов. Важно, чтобы при этом соответствующие линии портов работали на выход.

Изменять содержимое регистров **PortA** и **PortB** можно с помощью команд `bcf` и `bsf`, например: `bsf PortB,1` – установить бит 1 регистра PortB в единицу. Однако на практике, более удобно копировать в регистр содержимое акку-

мулятора W, например: `movwf PortB` – копирование содержимого аккумулятора W в регистр PortB. Можно сказать, что содержимое аккумулятора W выводится на порт В.

Если порт (или часть его линий) работает на вход, то состояние его линий будет отображаться в регистрах PortA и PortB, и будет меняться, так же, как меняется состояние соответствующих линий. В любой момент времени можно скопировать содержимое регистров PortA и PortB в аккумулятор W и проанализировать сигналы на линиях, а также, при необходимости, программно проанализировать их изменение. Для считывания состояния линий портов в аккумулятор W используется команда `movf`.

---

Формат команды:

`movf f,d` – копировать содержимое регистра в аккумулятор W (указатель d = 0); или же обратно в регистр f (указатель d = 1), то есть сам в себя. Копирование регистра самого в себя используется при проверке содержимого регистра на ноль. Если содержимое регистра f равно нулю, то и результат операции будет нулевым, что повлечет изменение состояния флага Z регистра STATUS.

Пример копирования содержимого регистра PortB в аккумулятор W:  
`movf PortB,0` ; копирование содержимого регистра в W. (d = 0).

---

Все линии портов, кроме линии RA4 порта А, имеют на выходе триггеры, которые позволяют выводить на линии порта сигналы ТТЛ уровней. Линия RA4 не может являться источником сигнала, поскольку вывод RA4 МК подключен к внутренней схеме с транзистором с открытым стоком. При этом при установке 4 бита регистра PortA в единицу происходит подключение линии порта к нулю источника питания ('земле'). Данное обстоятельство необходимо учитывать при проектировании схемы с использованием данного вывода.

К выводам порта В работающим на вход, могут быть программно подключены внутренние подтягивающие резисторы. Данные резисторы подключаются между выводом МК и плюсом источника питания. Они служат нагрузкой для схем, выполненных по схеме с открытым коллектором или открытым стоком. Данные схемы могут быть подключены к выводам МК, и являться для него источниками сигнала. Резисторы могут быть подключены или отключены только все 8 сразу. Управление ими осуществляется с помощью бита 7 РСН OPTION (Приложение Г). Порт А подтягивающих резисторов не имеет.

По причине ограниченного количества выводов МК, и присутствия в составе МК PIC16F628A большого числа периферийных модулей, большинство выводов микросхемы могут использоваться для работы с несколькими периферийными модулями (Приложение К, рисунок 2), при этом они не могут быть одновременно использованы как выводы портов. Чтобы использовать необходимые линии как выводы портов, необходимо программно отключить перифе-

рийные модули, способные использовать эти выводы в своей работе. Необходимо помнить, что исходное состояние выводов, заданное производителем микросхемы может не соответствовать задачам разработчика. Порядок использования и назначение выводов необходимо определить в начальный момент работы программы, настроив их, тем самым, произведя инициализацию МК.

Периферийных модулей в составе МК PIC16F628A влияющих на назначение выводов МК – семь. Список модулей и управляющих ими регистров приведен в таблице 1, содержимое регистров – в Приложениях В – Л.

Таблица 1 - Периферийные модули МК PIC16F628A

№	Модуль	Название модуля	Регистры управляющие отключением модуля
1	TMR0	Таймер 8 разрядный	OPTION
2	TMR1	Таймер 16 разрядный	T1CON
3	TMR2	Таймер 8 разрядный	T2CON
4	Компараторы	Модуль компараторов	CMCON
5	CCP	Сравнения/захвата/шим	CCP1CON
6	V <sub>ref</sub>	Источник опорного напряжения	VRCON
7	USART	Универсальный приемо-передатчик	RCSTA

\* Модуль таймера TMR0 не требует специального отключения.

\* Конфигурирование выводов МК как линий портов RA5, RA6, RA7 производится в слове конфигурации – специальном регистре, определяющем основную настройку МК и не отображенном в памяти. Слово конфигурации программно для МК не доступно, определяется пользователем в начале программы до её основной части, записывается в МК при программировании.

### 3. ЗАДАНИЕ НА ВЫПОЛНЕНИЕ ЛАБОРАТОРНОЙ РАБОТЫ

Настроить регистры специального назначения в порядке, предусмотренном вариантом (от пункта а до пункта л таблицы 3). Результаты выполнения работы представить по форме таблицы 2.

Таблица 2 – Форма отчета о выполнении задания

Пункт задания	Команда с использ. чисел в системе счисл.			Комментарии к команде
	Десятичной	Двоичной	16-тиричной	
X	<i>movlw .15</i>	<i>movlw b'00001111'</i>	<i>movlw h'0f'</i>	Копировать 15 в W
	<i>movwf PortA</i>			Записать содержимое W в регистр PortA
Y	<i>bcf PortB,3</i>			Установить 3 бит порта B в 0
.....				

### 4. СОДЕРЖАНИЕ ОТЧЕТА

4.1. Цель работы.

4.2. Перечень регистров спец. назначения с описанием их функций.

4.3. Перечень команд ассемблера, используемых для изменения состояния



регистров специального назначения.

4.4. Задание на выполнение лабораторной работы.

4.5. Результаты выполнения работы по форме таблицы 2.

Таблица 3 – Задания на выполнение лабораторной работы

Вар.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
а	Отключить периферийные модули в указанном порядке														
TMR1	1	3	6	4	5	4	6	6	3	4	3	6	3	5	1
TMR2	2	4	5	3	1	1	5	1	5	2	2	1	2	4	3
Компараторы	3	6	4	5	2	6	2	3	2	3	6	3	1	2	6
CCP	4	1	3	6	3	3	4	2	6	6	1	4	5	6	2
V <sub>ref</sub>	5	2	2	1	6	2	3	4	1	5	5	2	6	1	5
USART	6	5	1	2	4	5	1	5	4	1	4	5	4	3	4
б	Сконфигурировать линии портов (+ на вход, * – на выход)														
RA0	*	*	+	+	*	+	*	*	+	*	*	*	*	*	+
RA1	*	+	*	+	*	*	*	*	*	+	*	+	*	+	*
RA2	+	*	*	*	*	*	+	+	*	*	*	*	+	*	*
RA3	+	*	+	*	+	*	+	*	+	*	+	*	*	*	+
RA4	*	+	*	+	*	+	*	*	*	+	*	+	*	+	*
RA5	+	*	*	+	*	+	*	+	*	*	*	+	+	*	*
RA6	*	+	*	*	*	+	+	*	+	*	*	*	*	+	*
RA7	*	*	+	+	+	*	*	*	+	*	+	*	*	+	+
RB0	+	+	*	*	+	+	*	*	+	*	+	+	*	*	+
RB1	*	+	+	*	*	*	+	+	*	+	*	*	*	*	*
RB2	+	*	*	+	+	*	*	*	+	*	+	*	*	*	*
RB3	+	+	*	*	+	*	+	*	*	+	*	*	+	+	*
RB4	+	+	*	+	*	*	+	*	+	*	*	*	*	+	*
RB5	*	+	+	+	*	+	*	+	*	*	*	+	*	+	*
RB6	*	*	*	*	*	*	+	+	*	*	*	+	*	*	*
RB7	*	*	+	*	*	+	+	+	+	+	*	*	*	+	*
в	Переконфигурировать линии портов (+ на вход, * – на выход)														
RA0	+	+	*	*	+	*	*	*	*	+	*	*	*	*	*
RA1	+	*	+	*	*	*	*	+	*	*	*	*	*	*	*
RA2	+	*	*	*	+	*	*	*	*	*	+	+	*	*	+
RA3	*	+	+	*	*	*	+	+	+	+	+	*	*	+	*
RA4	+	*	*	*	+	*	*	*	+	*	*	*	+	+	*
RA5	*	+	*	*	*	+	*	*	*	+	*	*	*	*	*
RA6	+	*	*	+	*	*	+	+	*	*	*	+	+	*	*
RA7	*	*	+	*	*	+	*	*	*	*	+	*	*	*	+
RB0	*	*	*	*	+	*	+	*	+	*	*	+	*	*	*
RB1	+	+	*	+	*	*	*	*	*	+	*	*	+	*	*
RB2	+	*	+	*	*	*	+	*	*	*	*	*	*	+	*
RB3	*	*	*	*	+	*	+	*	+	+	*	*	+	*	+
RB4	*	+	*	*	*	*	*	*	*	*	*	+	*	*	*

RB5	+	+	*	*	+	*	*	*	+	*	*	*	*	*	+
RB6	+	*	*	+	*	*	+	*	+	*	*	*	+	+	*
RB7	+	*	+	*	+	+	*	*	*	+	*	+	*	*	*

Таблица 3 – продолжение.

Вар.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
г	Настроить порты А и В на выход														
д	Установить сигналы на линиях портов														
RA0	0	0	1	1	0	1	0	0	0	0	0	1	0	1	1
RA1	1	0	0	1	0	1	1	0	1	0	1	0	1	0	0
RA2, RA3	0	1	0	1	1	1	1	0	1	1	0	1	0	1	0
RA4	0	1	0	0	0	0	0	1	1	1	0	1	0	0	0
RA5	0	0	1	0	0	0	0	1	0	0	1	0	1	0	0
RA6, RA7	1	1	0	0	1	1	0	1	0	0	0	0	0	0	1
RB0	0	1	1	1	0	0	1	1	0	0	0	1	1	1	0
RB1	0	1	1	0	1	0	0	0	0	0	1	0	1	0	0
RB2	0	0	1	1	1	1	1	0	0	0	0	1	0	0	0
RB3, RB4	1	0	1	0	1	0	0	1	1	1	0	0	1	0	0
RB5	1	1	1	0	1	0	0	0	0	1	0	0	0	1	1
RB6	1	1	0	0	0	0	0	1	0	1	0	0	1	0	1
RB7	1	1	0	0	1	0	1	0	1	1	0	1	0	1	0
е	Переустановить сигналы на линиях портов														
RA0	1	1	0	1	1	0	1	1	1	0	0	0	1	0	0
RA1	0	1	1	0	0	1	0	1	0	1	0	1	1	1	0
RA2, RA3	1	0	0	1	1	0	1	0	1	1	0	0	1	0	1
RA4	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
RA5	0	1	0	0	0	1	0	0	1	0	0	1	0	0	0
RA6, RA7	0	0	1	0	1	1	1	1	1	1	0	0	1	0	1
RB0	0	0	0	1	1	0	1	0	0	0	1	1	0	0	0
RB1	0	1	1	0	0	0	0	1	1	0	1	0	1	0	0
RB2	1	1	1	1	1	1	1	1	0	1	1	0	0	1	0
RB3, RB4	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1
RB5	0	1	0	1	1	0	1	1	0	1	0	1	0	1	0
RB6	1	1	1	0	1	1	1	0	1	0	0	1	0	1	0
RB7	0	1	0	1	0	1	0	0	1	1	0	0	0	0	1
ж	Записать число в порт (числа даны в десятичной системе счисл.)														
А	255	13	154	125	211	3	2	16	56	26	24	45	124	45	52
В	34	153	24	66	47	189	169	151	155	201	125	1	42	154	255
з	Настроить порты А и В на вход														
и	Скопировать состояние линий порта (+) в аккумулятор W														
А	+		+	+			+			+	+			+	
В		+			+	+		+	+			+	+		+
к	Настроить порты														
А	На вход					На выход					На вход				

В	На выход			На вход			На вход	
л	Состояние линий порта вывести на линии другого порт							
	А в В	В в А	А в В	А в В	В в А	В в А	А в В	

#### 6. ЛИТЕРАТУРА.

1. Сайт компании MICROCHIP - [www.microchip.ru](http://www.microchip.ru)
2. Корабельников Е.А. Руководство по конструированию устройств на микроконтроллерах. 2006. [www.ikarab.narod.ru](http://www.ikarab.narod.ru)
3. Предько М.Справочник по PIC-микроконтроллерам.— М.:ДМК Пресс,2002
4. Ульрих В.А. Микроконтроллеры PIC16X7XX. – СПб.: Наука и техника,2002

#### ПРИЛОЖЕНИЕ А.

Структурная схема микроконтроллера PIC16F628A





Бит 7	IRP	– Бит выбора банка при косвенной адресации 1 = банк 2,3 (100h – 1FFh) 0 = банк 0,1 (000h – 0FFh)
Бит 6-5	RP1:RP0	– Биты выбора банка при непосредственной адресации 11 = банк 3 (180h – 1FFh) 10 = банк 2 (100h – 17Fh) 01 = банк 1 (080h – 0FFh) 00 = банк 0 (000h – 07Fh)
Бит 4	-TO	– Флаг переполнения сторожевого таймера WDT 1 = после POR или выполнения команд CLRWDT, SLEEP 0 = после переполнения WDT
Бит 3	-PD	– Флаг включения питания 1 = после POR или выполнения команды CLRWDT 0 = после выполнения команды SLEEP
Бит 2	Z	– Флаг нулевого результата 1 = нулевой результат выполнения арифм. или логич. операции 0 = не нулевой результат выполн. арифм. или логич. операции
Бит 1	DC	– Флаг десятичного переноса-заема (для команд ADDWF, ADDWL, SUBWF, SUBWL), заем имеет инверсное знач. 1 = был перенос из младшего полубайта 0 = не было переноса из младшего полубайта
Бит 0	C	– Флаг переноса-заема (для команд ADDWF, ADDWL, SUBWF, SUBWL), заем имеет инверсное значение 1 = был перенос из старшего бита 0 = не было переноса из старшего бита

**Примечание** Флаг заема имеет инверсное значение. Вычитание выполняется путем прибавления дополнительного кода второго операнда. При выполнении команд сдвига бит C загружается старшим или младшим битом сдвигаемого регистра. При помощи флагов –TO и –PD (биты 4 и 3) можно определить причину сброса микроконтроллера

## ПРИЛОЖЕНИЕ Г

Регистр **OPTION** (адрес 81h, 181h)

-RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
7	6	5	4	3	2	1	Бит 0

- Бит 7      -RBPU    – Включение подтягивающих резисторов на входах PortB  
                          1 = подтягивающие резисторы отключены  
                          0 = подтягивающие резисторы включены
- Бит 6      INTEDG    – Выбор активного фронта сигнала на входе внешнего прерывания INT  
                          1 = прерывание по переднему фронту сигнала  
                          0 = прерывание по заднему фронту сигнала
- Бит 5      T0CS        – Выбор тактового сигнала для TMR0  
                          1 = внешний тактовый сигнал с вывода RA4/T0CKI  
                          0 = внутренний тактовый сигнал CLKOUT
- Бит 4      T0SE        – Выбор фронта приращения TMR0 при внешнем тактовом сигнале  
                          1 = приращение по заднему фронту сигнала (с высокого к низкому уровню) на выводе RA4/T0CKI  
                          0 = приращение по переднему фронту сигнала (с низкого к высокому уровню) на выводе RA4/T0CKI
- Бит 3      PSA         – Выбор включения предделителя  
                          1 = предделитель включен перед WDT  
                          0 = предделитель включен перед TMR0
- Биты 2-0 PS2- PS0    – Установка коэффициента деления предделителя

Значение	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

## ПРИЛОЖЕНИЕ Д

Регистр **T1CON** (адрес 10h)

-	-	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
7	6	5	4	3	2	1	Бит 0

Бит 7-6 Не реализованы: читаются как '0'

Бит 5-4 T1CKPS1: T1CKPS0 – Выбор коэффициента деления предделителя TMR1

11 = 1 : 8

10 = 1 : 4

01 = 1 : 2

00 = 1 : 1

Бит 3 T1OSCEN – Включение внутреннего тактового генератора TMR1

1 = генератор включен

0 = генератор выключен

Примечание. Инвертирующий элемент и резистивная обратная связь выключены для уменьшения тока потребления.

Бит 2 T1SYNC – Синхронизация внешнего тактового сигнала

TMR1CS = 1

1 = не синхронизировать внешний тактовый

0 = синхронизировать внешний тактовый

TMR1CS = 0

Значение бита игнорируется

Бит 1 TMR1CS – Выбор источника тактового сигнала

1 = внешний источник с вывода RB6/T1OSO/T1CKI

0 = внутренний источник Fosc/4

Бит 0 TMR1ON – Включение модуля TMR1

1 = включен

0 = выключен

---

## ПРИЛОЖЕНИЕ Е

Регистр **CMCON** (адрес 1Fh)

C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM2	CM0
7	6	5	4	3	2	1	Бит 0

Бит 7 C2OUT – Выход компаратора 2



Если C2INV = 0

1 = C2 V<sub>IN+</sub> > C2 V<sub>IN-</sub>

0 = C2 V<sub>IN+</sub> < C2 V<sub>IN-</sub>

Если C2INV = 1

0 = C2 V<sub>IN+</sub> > C2 V<sub>IN-</sub>

1 = C2 V<sub>IN+</sub> < C2 V<sub>IN-</sub>

Бит 6 C1OUT – Выход компаратора 1

Если C1INV = 0

1 = C1 V<sub>IN+</sub> > C1 V<sub>IN-</sub>

0 = C1 V<sub>IN+</sub> < C1 V<sub>IN-</sub>

Если C1INV = 1

0 = C1 V<sub>IN+</sub> > C1 V<sub>IN-</sub>

1 = C1 V<sub>IN+</sub> < C1 V<sub>IN-</sub>

Бит 5 C2INV – Инверсный выход компаратора 2

1 = C2 инверсный выход

0 = C2 не инверсный выход

Бит 4 C1INV – Инверсный выход компаратора 1

1 = C1 инверсный выход

0 = C1 не инверсный выход

Бит 3 CIS – Подключение входов компараторов

Если CM2:CM0 = 001

1 = C1 V<sub>IN-</sub> подключен к RA3

0 = C1 V<sub>IN-</sub> подключен к RA0

Если CM2:CM0 = 010

1 = C1 V<sub>IN-</sub> подключен к RA3

C2 V<sub>IN-</sub> подключен к RA2

0 = C1 V<sub>IN-</sub> подключен к RA0

C2 V<sub>IN-</sub> подключен к RA1

Биты 2-0 CM2:CM0 Режим работы компараторов (8 режимов, см. рисунок в документации на мк.) 111 – компараторы выключены

## ПРИЛОЖЕНИЕ Ж

Регистр VRCON (адрес 9Fh)

VREN	VROE	VRR	-	VR3	VR2	VR1	VR0
7	6	5	4	3	2	1	Бит 0

Бит 7 VREN – Включение источника опорного напряжения

- 1 = источник опорного напряжения включен  
 0 = источник опорного напряжения выключен и не потребляет тока

Бит 6 VROE – Включение выхода  $V_{REF}$   
 1 = выход  $V_{REF}$  подключен к RA2  
 0 = выход  $V_{REF}$  не подключен к RA2

Бит 5 VRR – Диапазон выходного напряжения  $V_{REF}$   
 1 = нижний диапазон  
 0 = верхний диапазон

Бит 4 Не используется: читается как '0'

Биты 3-0 VR3: VR0 – Выбор выходного напряжения  $V_{REF}$   $0 \leq VR[3:0] \leq 15$   
 Если VRR = 1 :  $V_{REF} = (VR[3:0] / 24) \times V_{DD}$   
 Если VRR = 0 :  $V_{REF} = (V_{DD} \times 1/4) + (VR[3:0] / 32) \times V_{DD}$

### ПРИЛОЖЕНИЕ 3

Регистр PCON (адрес 8Eh)

-	-	-	-	OSCF	-	-POR	-BOD
7	6	5	4	3	2	1	Бит 0

Биты 7-4,2 Не реализованы: читаются как '0'

Бит 3 OSCF – Выбор частоты тактового генератора в режиме INTOSC/RC  
 1 = типовое значение 4 МГц <sup>(1)</sup>  
 0 = типовое значение 32 кГц

Бит 2 -POR – Флаг сброса по включению питания  
 1 = сброса по снижению питания не было  
 0 = произошел сброс микроконтроллера по включению питания

Бит 1 -BOD – Флаг сброса по снижению напряжения питания  
 1 = Сброса по снижению напряжения питания не было  
 0 = Произошел сброс микроконтроллера по снижению напряжения питания

Примечание. <sup>(1)</sup> В режиме RC генератора и OSCF = 1 частота тактового сигнала зависит от внешней RC цепочки, подключенной к выводу RA7/OSC1/CLKIN

### ПРИЛОЖЕНИЕ И

Регистр RCSTA (адрес 18h) Регистр управления и статуса приемника

SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
7	6	5	4	3	2	1	Бит 0

Бит 7 SPEN – Разрешение работы последовательного порта

		(вывод RB1/RX/DT и RB2/TX/CK используются USART когда биты TRIS<2:1> установлены в '1')
		1 = модуль последовательного порта включен 0 = модуль последовательного порта выключен
Бит 6	RX9	– Разрешение 9-битного приема 1 = 9-битный прием 0 = 8-битный прием
Бит 5	SREN	– Разрешение одиночного приема <u>Синхронный режим</u> 1 = разрешен одиночный прием 0 = запрещен одиночный прием Сбрасывается в '0' по завершению приема <i>Примечание.</i> В режиме ведомого не имеет значения. <u>Асинхронный режим</u> Не имеет значения.
Бит 4	CREN	– 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
Бит 3	ADDEN	– Разрешение детектирования адреса <u>Асинхронный 9-битный 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-битный прием (RX9=0), Синхронный режим</u> Не имеет значения
Бит 2	FERR	– Ошибка кадра, сбрасывается при чтении регистра RCREG 1 = произошла ошибка кадра 0 = ошибки кадра не было
Бит 1	OERR	– Ошибка переполнения внутреннего буфера, устанавливается в '0' при сбросе бита CREN 1 = произошла ошибка переполнения 0 = ошибки переполнения не было
Бит 0	RX9D	9-й бит принятых данных (может использоваться для программной проверки четности)

#### ПРИЛОЖЕНИЕ К

Регистр **T2CON** (адрес 12h)

-	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
7	6	5	4	3	2	1	Бит 0

Бит 7 Не реализован: читается как '0'

Бит 6-3 TOUTPS3 : TOUTPS0 – Выбор коэффициента выходного делителя TMR2

0000 = 1 : 1	0110 = 1 : 7	1100 = 1 : 13
0001 = 1 : 2	0111 = 1 : 8	1101 = 1 : 14
0010 = 1 : 3	1000 = 1 : 9	1110 = 1 : 15
0011 = 1 : 4	1001 = 1 : 10	1111 = 1 : 16
0100 = 1 : 5	1010 = 1 : 11	
0101 = 1 : 6	1011 = 1 : 12	

Бит 2 TMR2ON – Включение модуля TMR2

1 = включен

0 = выключен

Биты 1-0 T2CKPS1 : T2CKPS0 – Выбор коэффициента деления предделителя TMR2

00 = 1 : 1

01 = 1 : 4

1x = 1 : 16

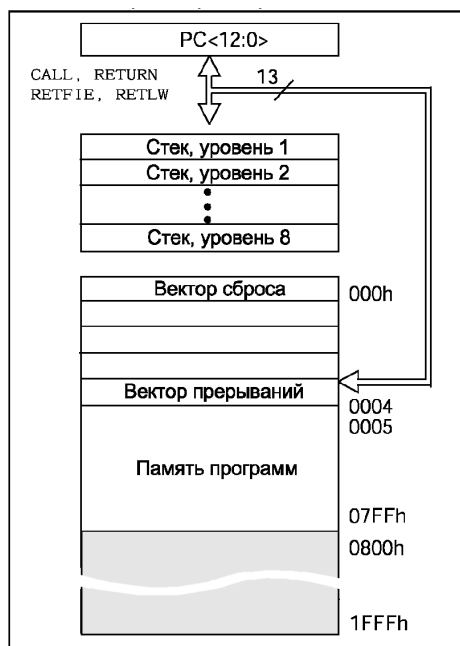


Рисунок 1 – Организация памяти в микроконтроллере PIC16F628A

PDIP, SOIC

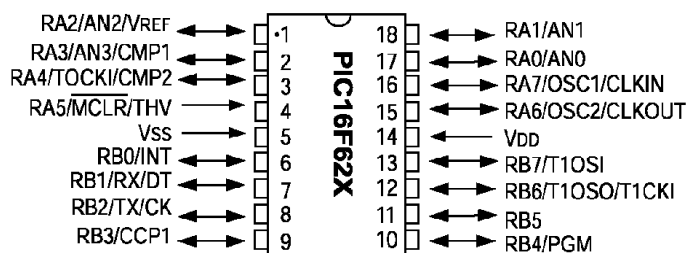


Рисунок 2 – Расположение выводов микроконтроллера PIC16F628A

## ПРИЛОЖЕНИЕ Л

Регистр CCP1CON (адрес 17h)

-	-	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0
7	6	5	4	3	2	1	Бит 0

Бит 7-6 Не реализованы: читаются как '0'

Биты 5-4 CCR1X : CCR1Y – Младшие биты скважности ШИМ

Режим захвата: не используются

Режим сравнения: не используются

Режим ШИМ : два младших бита скважности. Восемь старших находятся в CCR1L

Биты 3-0 CCR1M3 : CCR1M0 – Режим работы модуля CCR1

0000 = модуль CCR1 выключен (сброс модуля CCR1)

0100 = захват по каждому заднему фронту сигнала

0101 = захват по каждому переднему фронту сигнала

0110 = захват по каждому 4-му переднему фронту сигнала

0111 = захват по каждому 16-му переднему фронту сигнала

1000 = Сравнение, устанавливает выходной сигнал (устанавливается флаг CCR1IF в '1')

1001 = Сравнение, сбрасывает выходной сигнал (флаг CCR1IF в '1')

1010 = Сравнение, на выходной сигнал не влияет (флаг CCR1IF в '1')

1011 = Сравнение, триггер специальных функций (устанавливается флаг CCR1IF в '1', CCR1 сбрасывает TMR1 )

11xx = ШИМ режим

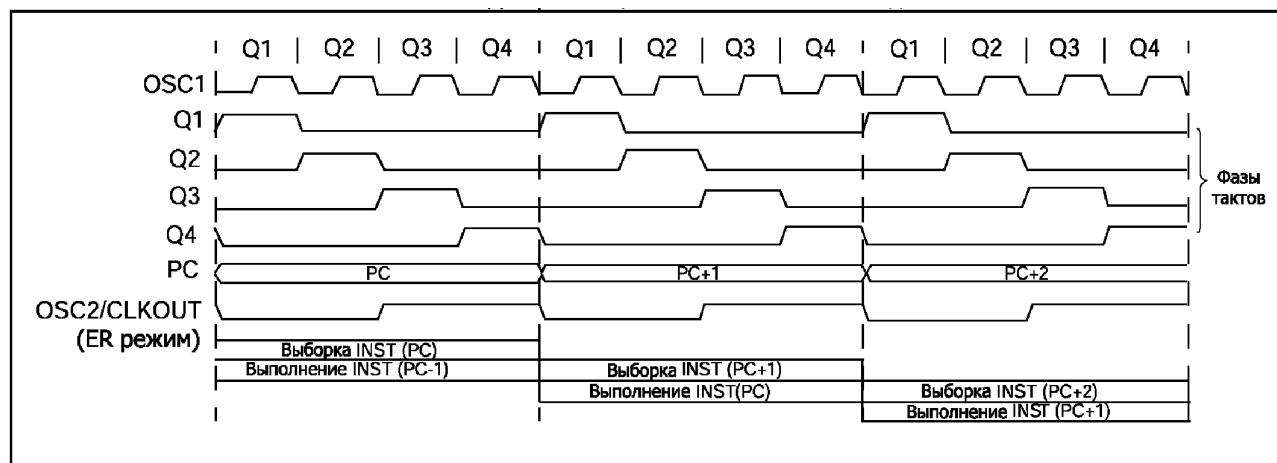


Рисунок 3 – Диаграмма циклов выполнения команд

## ПРИЛОЖЕНИЕ М.

### 1. КОНТРОЛЬНЫЕ ВОПРОСЫ ДЛЯ ПРОВЕРКИ ГОТОВНОСТИ СТУДЕНТОВ К ВЫПОЛНЕНИЮ ЛАБОРАТОРНОЙ РАБОТЫ.

- 1.Определение микроконтроллера,особенности как класса устройств,структура.
- 2.Сущность, состав, особенности, достоинства и недостатки «принстонской» и

«гарвардской» архитектур микроконтроллеров.

3. Типы памяти микроконтроллеров, особенности каждого типа памяти.
4. Системы команд. CISC и RISC.
5. Архитектура микроконтроллеров семейства PIC16.
6. Тактовый генератор, машинный цикл микроконтроллеров семейства PIC16.
7. Организация памяти программ микроконтроллеров PIC16.
8. Сброс микроконтроллера серии PIC16, схемы POR, PWRT, OST, BOR, WDT.
9. Периферийные модули ( и их назначение) в составе МК серии PIC16.
10. Прерывания в МК серии PIC16, стек.
11. Основные характеристики микроконтроллера серии PIC16.

## II. КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ИТОГАМ ЛАБОРАТОРНОЙ РАБОТЫ.

1. Какие регистры называют регистрами специального назначения?
2. Какие регистры специального назначения изучены в лабораторной работе?
3. Опишите назначение и состав регистров PortA, PortB, TrisA, TrisB.
4. Опишите состав и назначение регистра STATUS.
5. В чем отличие рабочих битов от флагов? Перечислите известные Вам флаги.
6. Перечислите бит-ориентированные команды для изменения содержимого регистров, укажите их формат и особенности применения.
7. Перечислите байт-ориентированные команды для изменения содержимого регистров, укажите их формат и особенности применения.
8. Чем отличаются команды *movwf*, *movf*, *movlw*?

-----  
ЛР N            от    .    .12.

Объем 1.00 усл.п.л.

Формат 60x80/16

Заказ N        . Тираж        . Цена

-----  
©Центр Дистанционного образования ДГТУ

Адрес университета :

344010, Ростов-на-Дону, пл. Гагарина, 1