

## **8 ОРГАНИЗАЦИЯ 8-РАЗРЯДНЫХ МИКРОКОНТРОЛЛЕРОВ**

### **На примере семейства МК PICmicro фирмы MicroChip**

#### **ОСОБЕННОСТИ:**

- 8-разрядное слово данных.
- Гарвардская архитектура.
- RISC-система команд.
- Устройство управления на принципе схемной логики.
- 4-х тактовый машинный цикл, тактовая частота до 20 МГц (маш. цикл 200 нс).
- 4 вида памяти:
- Малое энергопотребление (около 0,6 мА/кристалл) и менее 1 мкА в режиме энергосбережения.
- Широкий набор периферийных модулей.
- Наличие как цифровых, так и аналоговых средств ввода/вывода.

Функциональная схема микроконтроллеров семейства PicMicro показана на рисунке 1.

#### **СОСТАВ:**

- центральное процессорное устройство (ЦПУ) – «Ядро», включающее систему управления, синхронизации и сброса, операционное устройство на основе АЛУ, набор управляющих регистров специального назначения;
- система памяти, включающая память программ, память данных, постоянную память данных (EEPROM), стек;
- набор периферийных модулей;
- порты ввода/вывода.

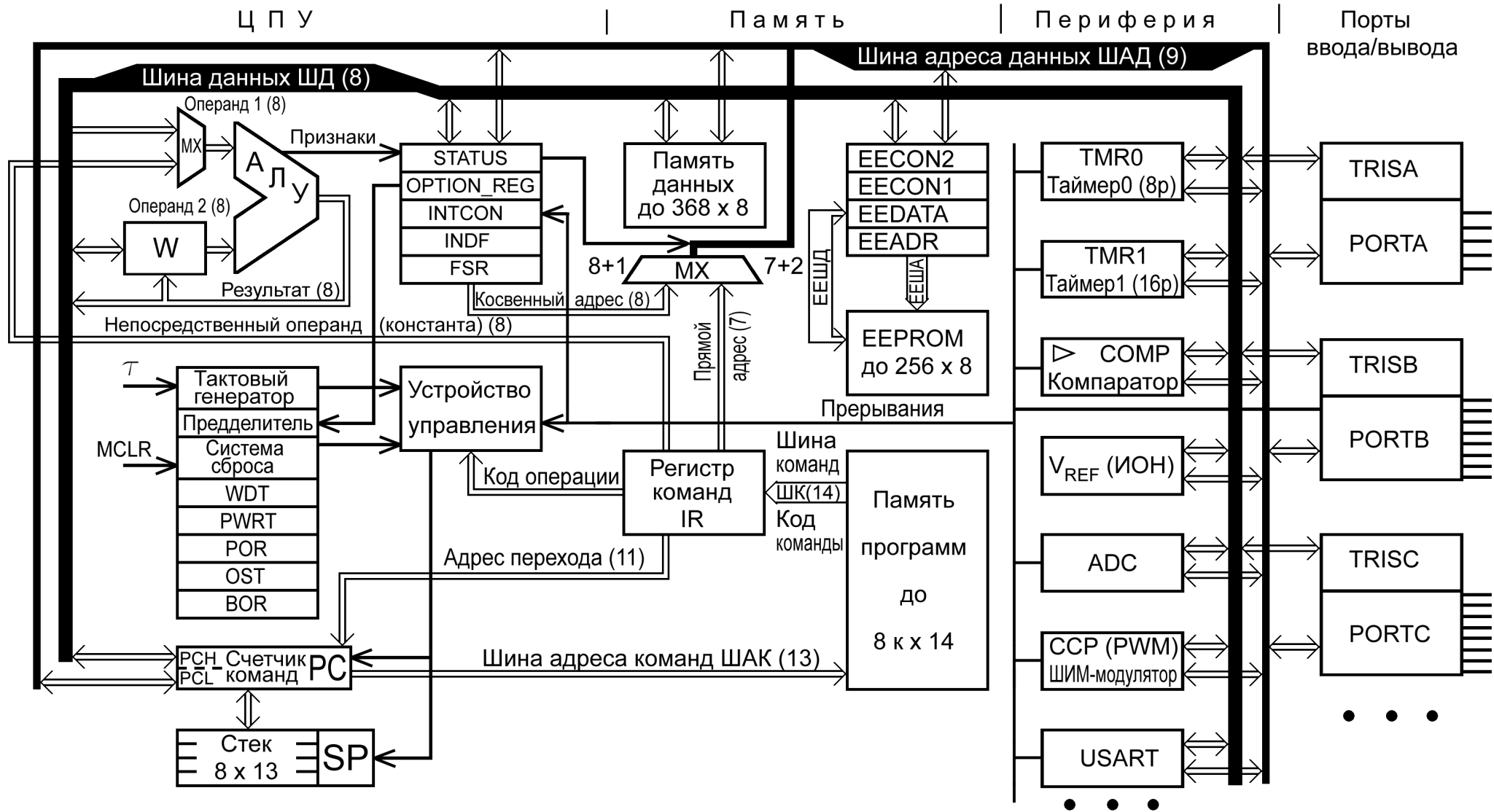


Рисунок 1 - Функциональная схема микроконтроллеров семейства PICmicro серии PIC16

## **СИСТЕМА УПРАВЛЕНИЯ СИНХРОНИЗАЦИИ И СБРОСА**

Устройство управления формирует последовательность управляющих сигналов, обеспечивающих выполнение действий, предписанных кодом команды. Помимо комбинационного узла, регистра состояния и дешифратора команд, входящих в состав управляющего устройства, к системе управления относится программно недоступный регистр команд IR, в который загружается код исполняемой команды. Код команды содержит поле кода операции и, в зависимости от типа команды, может содержать поле непосредственного операнда (константа) (8 разрядов), поле адреса операнда (7 разрядов), поле приемника результата (1 разряд), поле номера бита (3 разряда), поле адреса перехода (11 разрядов).

Выборка кода команды производится из ячейки памяти программ, на которую указывает счетчик команд ( $PC = \langle PCN:PCL \rangle$ ). При выполнении операций над данными возможно применение одного из трех способов адресации операнда:

- непосредственная – операнд указывается в коде команды в виде константы;
- прямая – в коде команды указывается адрес источника операнда;
- косвенная – в качестве источника операнда указывается виртуальный регистр INDF, через который происходит обращение к ячейке памяти данных, адрес которой содержится в регистре косвенного адреса FSR.

УУ допускает выполнение безусловных и условных программных переходов. При этом 11-разрядный адрес перехода указывается в коде команды, а два недостающие разряда должны быть загружены в регистр PCLATH предварительно.

### **Тактирование МК.**

Тактовый сигнал определяет темп работы МК и необходим для выполнения команд и работы периферийных устройств. Для формирования тактового сигнала предусмотрен внутренний генератор. Тактовый сигнал формируется тактовым генератором на основе следующих событий:

- внешний сигнал;
- сигнал внутреннего генератора с внешним кварцевым резонатором;
- сигнал внутреннего генератора с внешним керамическим резонатором;

- сигнал внутреннего генератора с внешней RC-цепью;
- сигнал внутреннего генератора с внутренней RC-цепью (в PIC16F84A нет).

Тактовый сигнал разделяется внутренней схемой МК на четыре последовательных неперекрывающихся такта Q1, Q2, Q3, Q4, образующих машинный цикл (МЦ) (Рисунок 2). Команда в МК выполняется за два машинных цикла. В первом цикле, который действует на фоне выполнения предыдущей команды, в такте Q1 счетчик команд (PC) увеличивается на единицу, а выборка кода команды из памяти и программ происходит в такте Q4. Во втором цикле в такте Q1 происходит декодирование кода, в такте Q2 – выборка операнда, в такте Q3, собственно, реализация предусмотренного командой алгоритма, в такте Q4 сохранение полученного результата. Каждый цикл требует четырех тактов генератора. Следовательно, на весь цикл команды тратится восемь тактов, но пользователь «видит», что выполнение команды происходит в течении четырех тактов генератора. Исключение составляют команды перехода. Так как их действие связано с изменением содержимого счетчика команд, их выполнение и первый цикл следующей команды происходят в разные моменты времени и занимают восемь тактов.

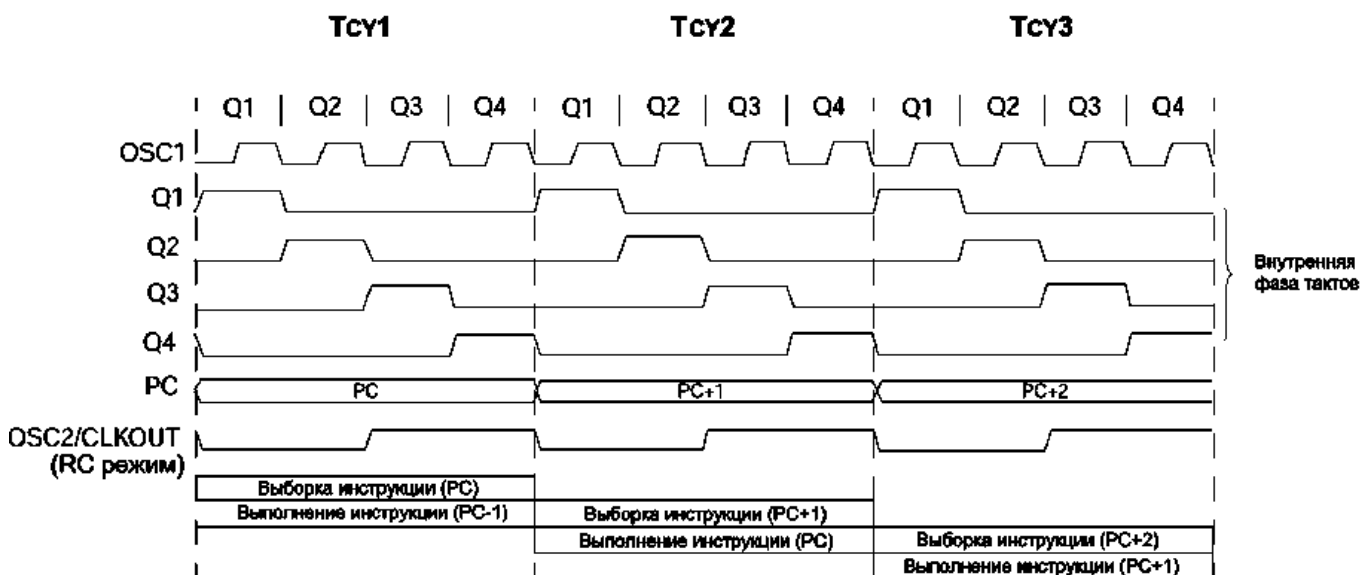


Рисунок 2

В систему тактирования входит предделитель – счетчик – делитель частоты тактового генератора с фиксированным набором коэффициентов деления. Может

быть использован для тактирования либо системных таймеров либо сторожевого таймера.

МК может быть переведен в режим пониженного энергопотребления (режим «сна» SLEEP) специальной командой SLEEP. При этом тактовый генератор выключается, выполнение программы приостанавливается, ЦПУ не работает, все регистры сохраняют свои данные. Выход из этого режима происходит по внешнему сбросу, переполнению сторожевого таймера или возникновении события, вызывающего разрешенное внешнее прерывание.

### **Система сброса.**

Логика сброса предназначена для перевода МК в исходное состояние с заведомо известными параметрами работы. Счетчик команд обнуляется, что обеспечивает начало выполнения программы с нулевого адреса. Большинство регистров не меняют значения при любых видах сброса, но после сброса по включению питания они содержат случайные значения.

Сброс предусмотрен внешним сигналом MCLR и по внутренним процессам МК:

- POR – при включении питания (Power-on Reset), удерживает МК в состоянии сброса, пока напряжение питания  $V_{DD}$  не достигнет требуемого уровня;

- PWRT – с задержкой при включении питания (Power-up Timer), срабатывает после включения питания и удерживает МК в состоянии сброса в течение 72 мс, пока не стабилизируется напряжение питания;

- OST – обеспечивает задержку в 1024 периодов тактового генератора после включения питания (Oscillator Start-up Timer), это гарантирует, что частота генератора стабилизировалась;

- BOR – при снижении напряжения питания ниже установленного значения (Brown Out Reset);

- WDT – сброс при переполнении сторожевого таймера (Watchdog Timer).

## ОПЕРАЦИОННОЕ УСТРОЙСТВО

Построено на основе арифметическо-логического устройства (АЛУ), выполняющего операции с данными, содержит рабочий регистр-аккумулятор (W) и регистр признаков и состояния (STATUS) (рисунок 3).

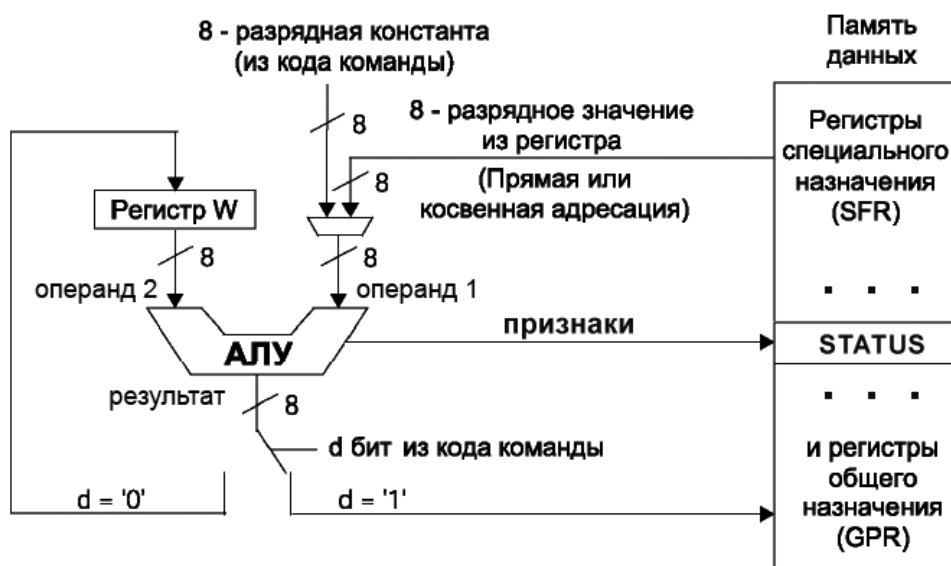


Рисунок 3

8-разрядное АЛУ может выполнять арифметические сложение и вычитание, прибавление и вычитание единицы, логические функции конъюнкции (AND), дизъюнкции (OR), инверсию (COM), ИСКЛЮЧАЮЩЕЕ ИЛИ (XOR), сдвиги. Первый из операндов находится в любом из регистров специального или общего назначения, а второй всегда в аккумуляторе. Результат может быть сохранен либо в регистре – источнике первого операнда, либо в аккумуляторе. Соответствующий операнд при этом теряется.

По результатам выполнения арифметической или логической операции в регистре STATUS формируются следующие признаки (флаги):

**C** – флаг переноса из старшего разряда. Устанавливается, если результат сложения превысил число 255. Кроме того выполняет функцию флага заёма при выполнении операции вычитания. Сбрасывается, если был заём, устанавливается, если не был. При выполнении сдвига, в него попадает выдвигаемый бит;

**DC** – флаг десятичного переноса/заёма. Действует аналогично флагу C для полубайта;

**Z** – флаг нулевого результата. Устанавливается, если в результате операции получился ноль и сбрасывается, если нет.

## СИСТЕМА ПАМЯТИ

**Память программ** - энергонезависимая память для хранения кода программы однократно или многократно программируемая (FLASH). Содержит до 8 к 14-разрядных ячеек (в PIC16F84A 1 к). Для адресации используется 13-разрядный счетчик команд (PC), состоящий из двух регистров PCH (5 разрядов) и PCL (8 разрядов).

**Память данных** – 8-разрядная энергозависимая память для хранения операндов и промежуточных результатов. Поделена на четыре страницы (банка) по 128 байт каждая (Рисунок 4) (в PIC16F84A всего 68 байт). Специальные регистры входят в адресное пространство памяти данных.

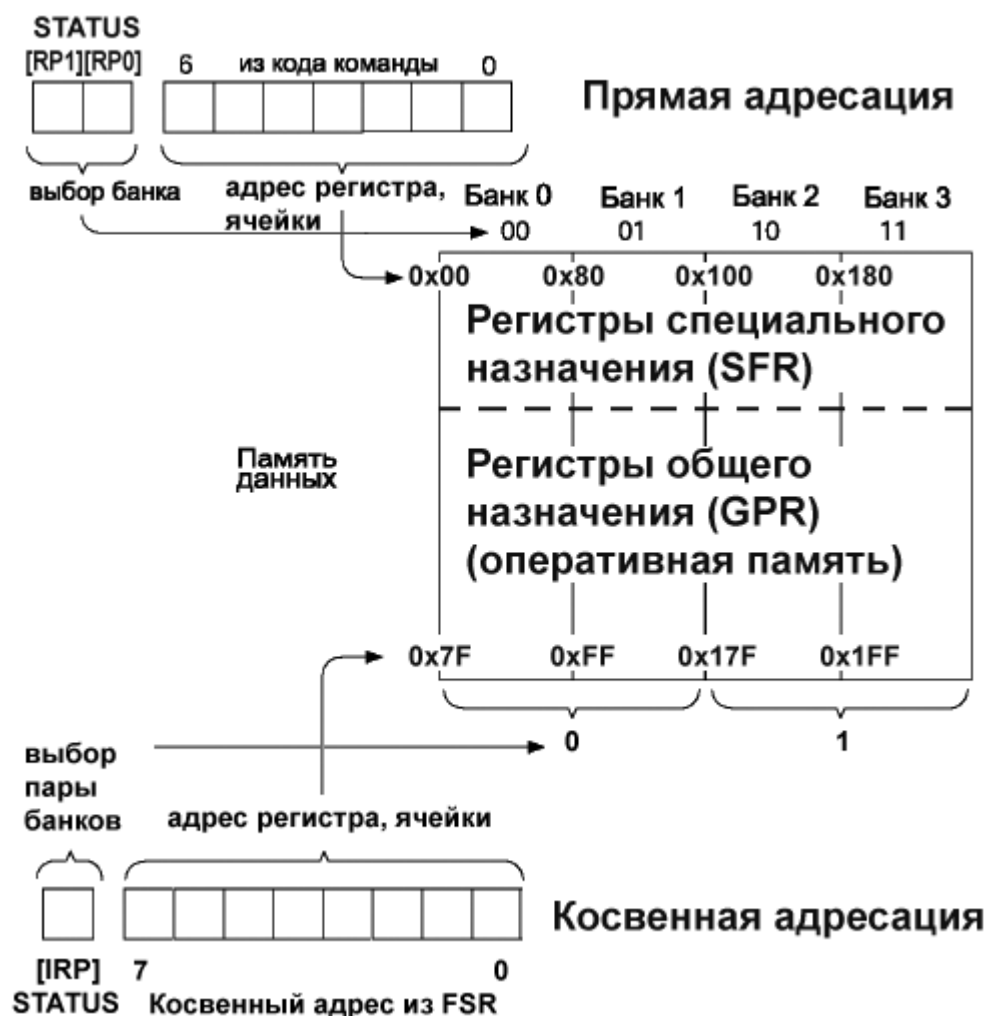


Рисунок 4

Регистры специального назначения (управляющие) SFR – special function registers:

- регистр W рабочий регистр – аккумулятор;
- счетчик команд PC -> PCL, PCH -> PCLATH;
- регистр признаков STATUS;
- регистр управления OPTION\_REG;
- регистр прерываний INTCON;
- регистр косвенного адреса FSR;
- виртуальный регистр доступа к памяти данных INDF (Indirect Addressing);
- широкий набор регистров, управляющих периферийным оборудованием

(TRISx, ADCON и др.).

Регистры общего назначения GPR (General Purpose Register), представляющие собой ячейки ОЗУ.

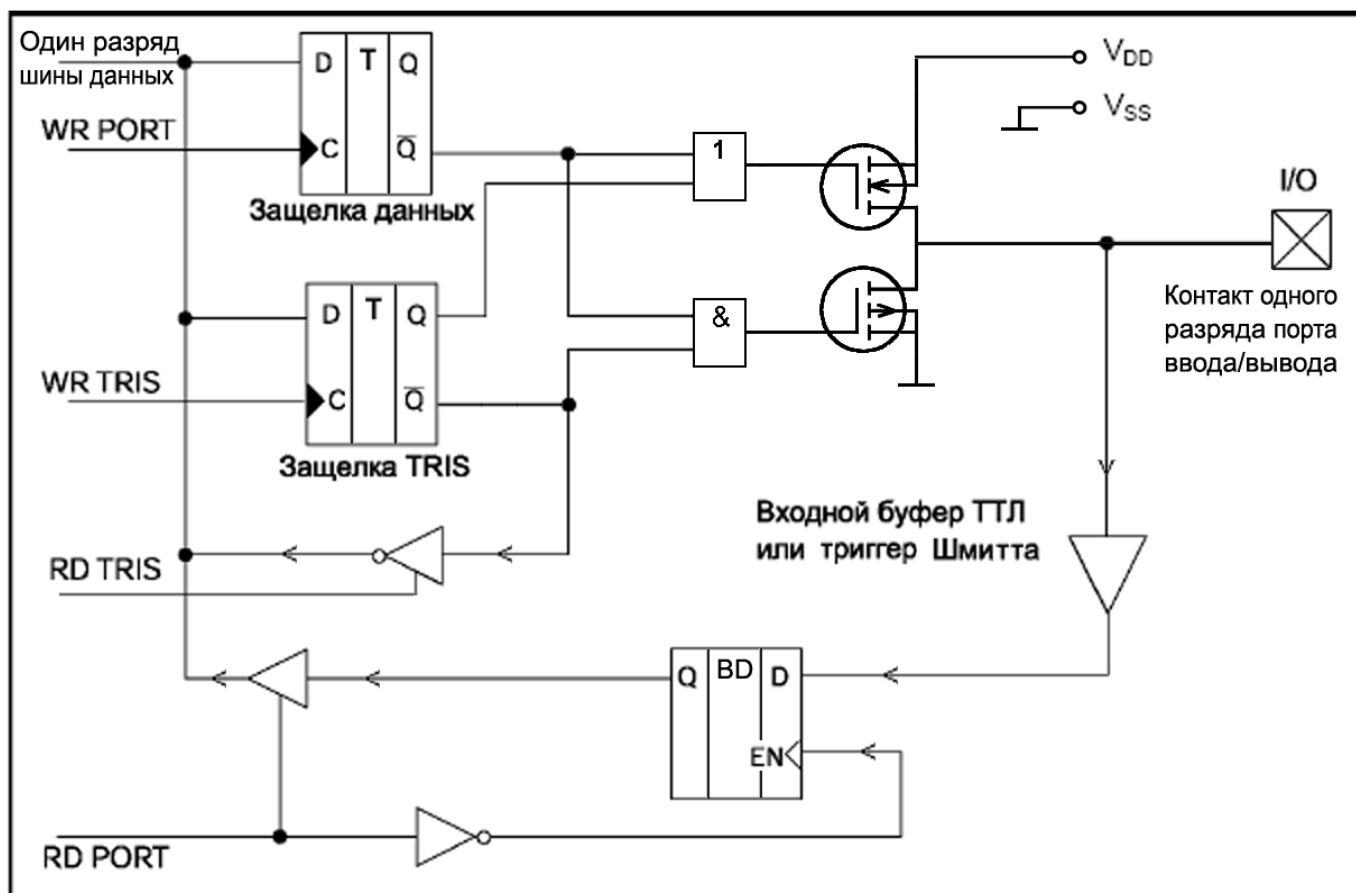
**EEPROM** – 8-разрядная энергонезависимая память для хранения редко меняющихся данных, которые должны быть сохранены после выключения питания. Содержит до 256 к (в PIC16F84A 64 к). По отношению к ячейкам EEPROM возможна только косвенная адресация через регистр EEADR, а обмен данными через регистр EEDATA.

**Стек** – отдельный модуль из восьми 13-разрядных ячеек, снабженных программно недоступным указателем стека. Предназначен исключительно для хранения адресов возврата из подпрограмм.



## ПОРТЫ ВВОДА/ВЫВОДА

МК содержит параллельные порты ввода/вывода, каждый разряд которых может быть запрограммирован на ввод или вывод сигнала. Порты могут быть сгруппированы по разрядности (обычно 8) и представляют собой регистр PORT<x> (где x – буквенное имя порта: A, В и т.д., например, PORTB) доступный для чтения и записи, через который осуществляется обмен информацией. На рисунке 5 показана типовая функциональная схема одного канала (разряда) порта ввода/вывода. В состав порта входит управляющий регистр (TRIS<x>), определяющий направление передачи данных. Настройка порта заключается в записи управляющей комбинации нулей и единиц в регистр TRIS<x>. Показанный на рисунке D-триггер «Защелка TRIS» представляет собой 1/8 регистра TRIS<x>, а D-триггер «Защелка данных» - 1/8 регистра PORT<x>.



Примечание. Все выводы портов имеют защитные диоды, подключенные к  $V_{DD}$  и  $V_{SS}$ .

Рисунок 5

Запись единицы в «Защелку TRIS» приводит к тому, что, независимо от состояния «Защелки данных», на выходе элемента 2ИЛИ формируется единица, а на выходе элемента 2И – ноль. Оба транзистора закрыты, выходные схемы соответствующего разряда порта устанавливаются в Z-состояние и этот вывод микроконтроллера доступен только для ввода информации.

Запись нуля в «Защелку TRIS» приводит к тому, что состояние выхода определяет информация, записанная в «Защелку данных», то есть подлежащая выводу. Если записан ноль, на выходах элементов формируется единица, нижний транзистор (с каналом n-типа) открыт, а верхний (с каналом p-типа) закрыт, на выходе ноль. Если записана единица, на выходах элементов формируется ноль, нижний транзистор закрыт, а верхний открыт, на выходе единица. То есть на выходе присутствует логический уровень сигнала, соответствующий выводимой информации. и этот вывод микроконтроллера доступен только для ввода информации.

Чтение порта возможно при любом состоянии регистра TRIS. Информация считывается в шину данных через шинный формирователь BD непосредственно с линии порта. Даже если в порт выведена единица, а линия закорочена на «землю», то при чтении будет прочитан ноль.

Есть простой способ запомнить направление канала ввода/вывода и состояние битов регистра TRIS<x>: "1" напоминает "Input" (ввод), а "0" напоминает "Output" (вывод).

## ПЕРИФЕРИЙНЫЕ МОДУЛИ

В состав МК входят периферийные модули, обеспечивающие выполнение таких спец. функций, как формирование временных интервалов, формирование сигналов ввода/вывода, аналоговый ввод/вывод данных, сравнение аналоговых сигналов и др. функции.

**Системный таймер/счетчик.** Предназначен для формирования временных интервалов, подсчета событий и участия в работе периферийных модулей, использующих синхронизацию по времени (например, ШИМ-модулятор). В зависимости от модели, контроллер серии PICmicro может содержать от одного до трех 8- и 16-разрядных таймеров. На рисунке 6 показана функциональная схема 8-разрядного таймера TMR0, имеющегося в составе всех МК семейства.

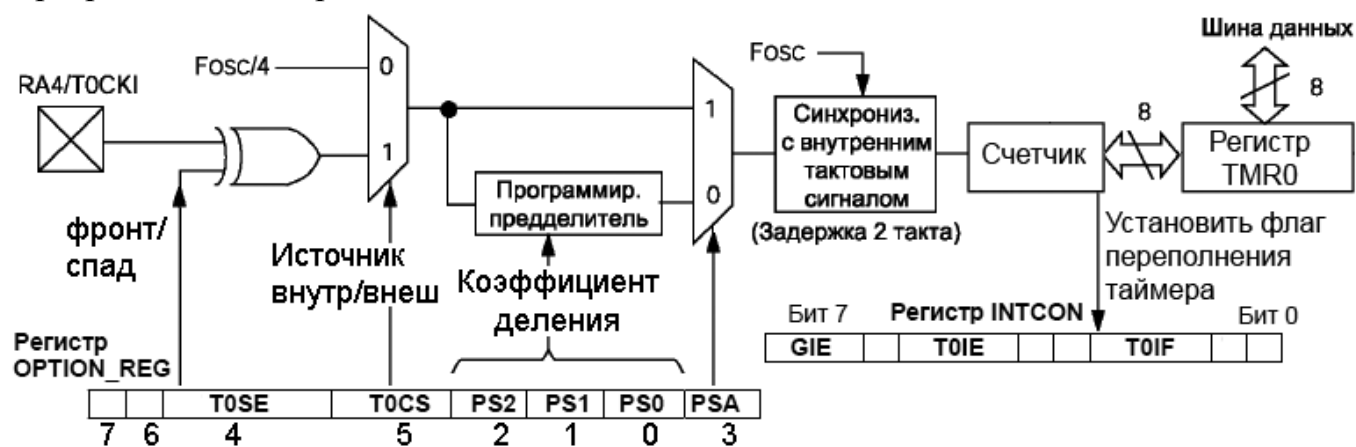


Рисунок 6

Основой таймера является 8-разрядный двоичный счетчик. Возможность предварительной записи числа позволяет получить коэффициент счета от 2 до 256. Остальные элементы слева от счетчика (по схеме рис. 6) служат для снабжения счетчика счетными импульсами. Источником счетных импульсов может быть фронт или спад (в зависимости от значения бита T0SE) внешнего сигнала T0CKI или каждый четвертый импульс внутреннего генератора (в зависимости от значения бита T0CS). Частота полученных импульсов может быть поделена предделителем, коэффициент деления N которого задается битами PS2 – PS0 по формуле:

$$N=2^{<PS2 - PS0>+1}, \text{ где } <PS2 - PS0> - \text{трехзначное двоичное число.}$$

На использование предделителя указывает нулевое состояние бита PSA. Управляющие биты входят в состав специального регистра OPTION\_REG и их значения выступают в роли управляющих сигналов. Переполнение счетчика происходит при переходе от состояния «255» к состоянию «0». При этом устанавливается бит TOIF регистра INTCON, управляющего прерываниями. Состояние бита TOIF может быть считано и сброшено программно, либо вызвать прерывание, если оно разрешено установкой битов GIE (глобальное разрешение прерываний) и TOIE (разрешение прерывания от таймера).

## СИСТЕМА КОМАНД

Одноадресная RISC система команд МК содержит 35 14-разрядных команд. Код команды обычно включает в себя поле кода операции и поле параметра. В качестве параметра может быть указан операнд (8 разрядов), адрес операнда (7 разрядов), приемник результата (1 разряд) или адрес перехода (11 разрядов).

Всю систему команд можно условно поделить на четыре группы:

- Команды пересылки информации;
- Логические и арифметические операции
- Команды ветвления
- Команды управления

При рассмотрении формата команд будем использовать следующие обозначения:

W – рабочий регистр-аккумулятор (или просто, аккумулятор);

F – любой регистр (ячейка) памяти данных, SFR или GPR,

f f f f f f – его семиразрядный адрес;

d – бит, указывающий место сохранения результата. Если  $d = 0$ , результат сохраняется в аккумуляторе, если  $d = 1$ , результат сохраняется в памяти данных на месте первого операнда;

b b b – трехразрядный номер бита в байте;

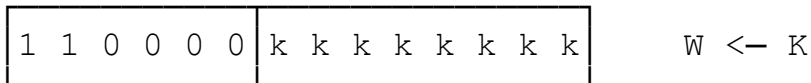
K – константа

k k k k k k k k – 8 бит в командах пересылки, арифметических и логических,

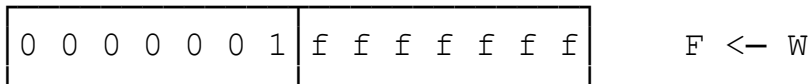
k k k k k k k k k k – адрес перехода 11 бит в командах ветвления.

## Группа команд пересылки информации

MOVLW K Переслать константу в W

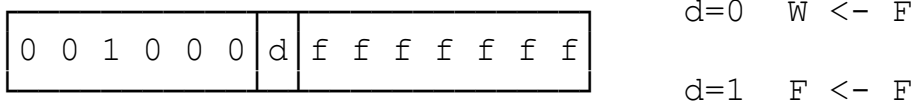


MOVWF F переслать W в F



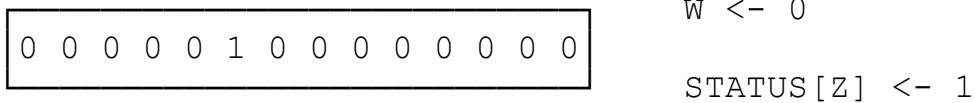
Содержимое W копируется в регистр с адресом F

MOVF F,d Переслать из F

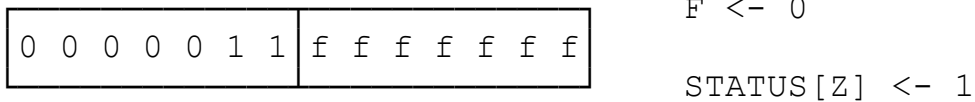


Если d=0 содержимое регистра с адресом F копируется в W, или если d=1 содержимое регистров F и W не меняется, устанавливается признак Z в STATUS в соответствии с содержимым F

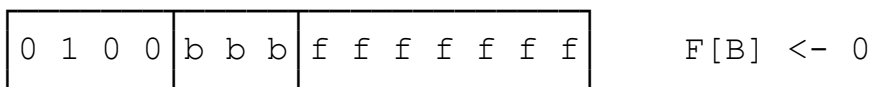
CLRW Очистить W



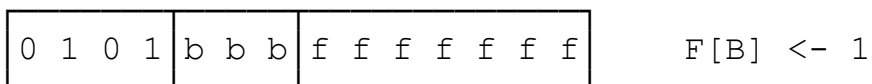
CLRF F Очистить F



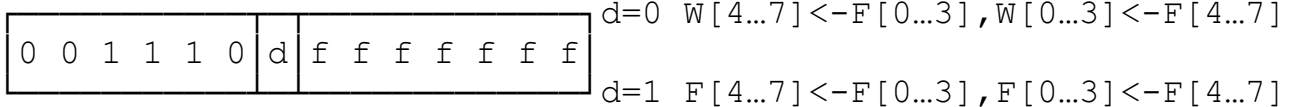
BCF F,b Очистить бит b в регистре F



BSF F,b Установить бит b в регистре F

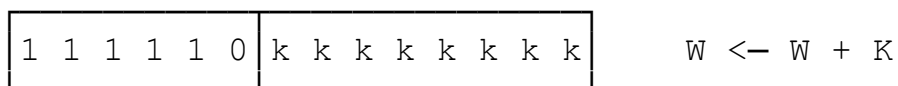


SWAPF F,d Поменять местами полубайты в регистре F

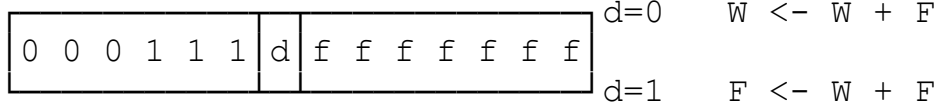


## Логические и арифметические операции

ADDLW K Сложить W с константой



ADDWF F,d Сложение W и F



SUBLW K Вычесть W из константы

SUBWF F,d вычесть W из F

ANDLW K Побитное «И» константы и W

ANDWF F,d Побитное «И» W и F

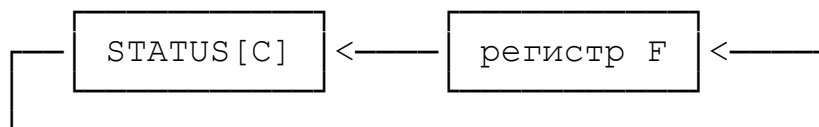
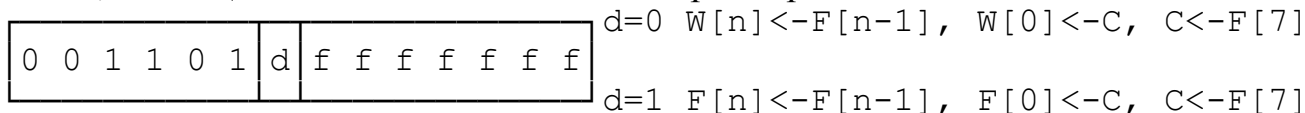
IORLW K Побитное «ИЛИ» константы и W

IORWF F,d Побитное «ИЛИ» W и F

XORLW K Побитное «ИСКЛЮЧАЮЩЕЕ ИЛИ» константы и W

XORWF F,d Побитное «ИСКЛЮЧАЮЩЕЕ ИЛИ» W и F

RLF F,d Циклический сдвиг F влево через перенос



RRF F,d Циклический сдвиг F вправо через перенос

DEC F,d Вычесть 1 из F

INC F,d Прибавить 1 к F

COM F,d Инвертировать F

## Команды ветвления

**GOTO K** Безусловный переход по адресу K

1 0 1	k k k k k k k k k k k	PC ← K
-------	-----------------------	--------

**DECFSZ F,d** Вычесть 1 из F и пропустить следующую команду, если 0

0 0 1 0 1 1	d	f f f f f f f f	F-1=0 PC←PC+2
			F-1≠0 PC←PC+1

**INCFSZ F,d** Прибавить 1 к F и пропустить следующую команду, если 0

**BTFSZ F,b** Проверить бит b в регистре F, пропустить следующую команду, если 0

0 1 1 0	b b b	f f f f f f f f	F[b]=0 PC←PC+2
			F[b]≠0 PC←PC+1

**BTSS F,b** Проверить бит b в регистре F, пропустить следующую команду, если 1

**CALL K** Вызов подпрограммы с адреса K

1 0 0	k k k k k k k k k k k	(SP) ← PC
		PC ← K

**RETURN** Возврат из подпрограммы

0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	PC ← (SP)
-------------------------------	-----------

**RETFIE** Возврат из подпрограммы с разрешением прерываний

**RETLW K** Возврат из подпрограммы с загрузкой константы в W

## Команды управления

**NOP** Нет операции

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	не выполняются никакие действия
-------------------------------	---------------------------------

**CLRWDT** Очистить WDT

**SLEEP** Перейти в режим SLEEP