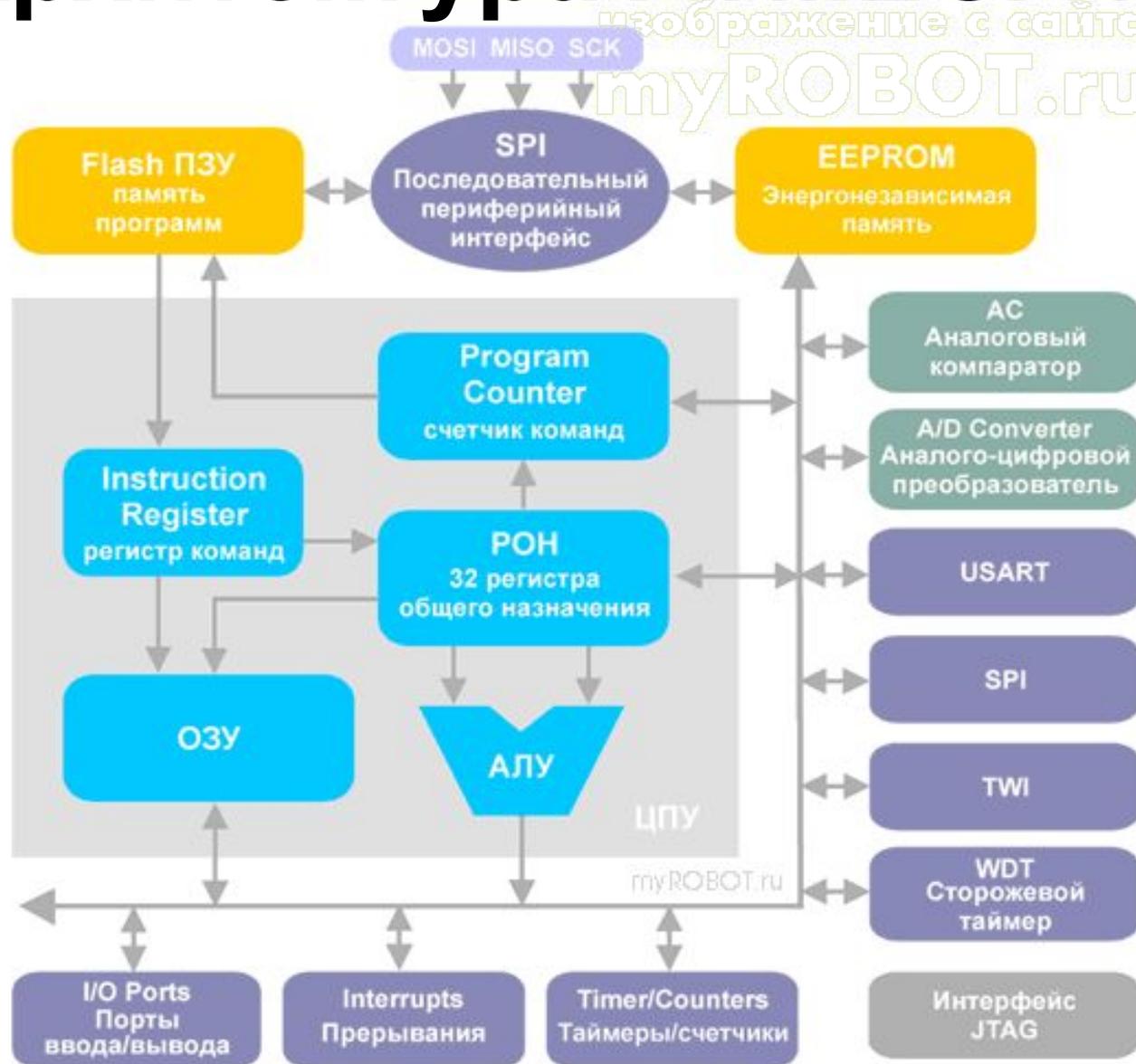
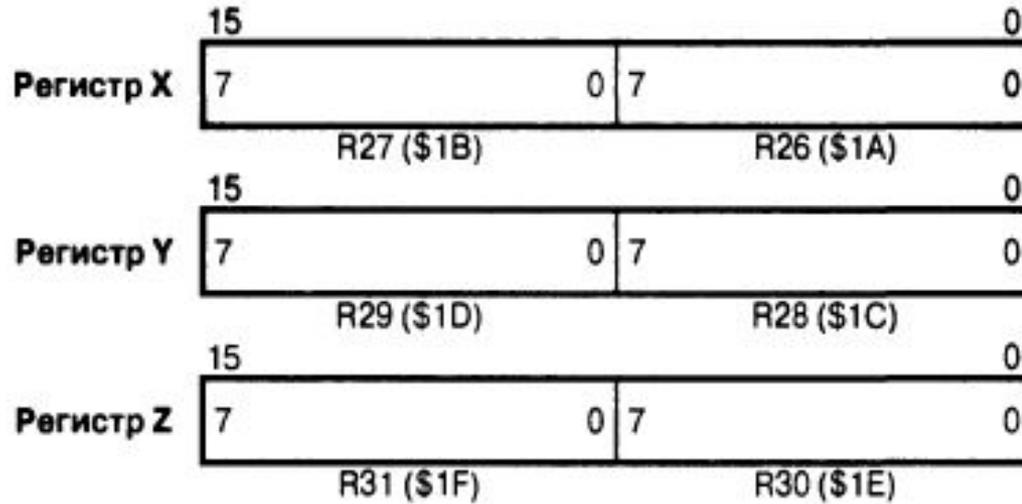


# Микроконтроллеры AVR семейства Mega

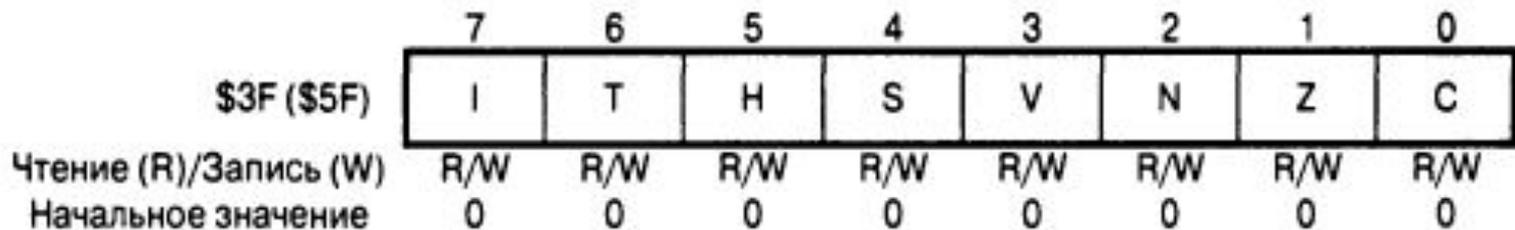
# Архитектура ATMEGA16



# Регистры-указатели X, Y, Z

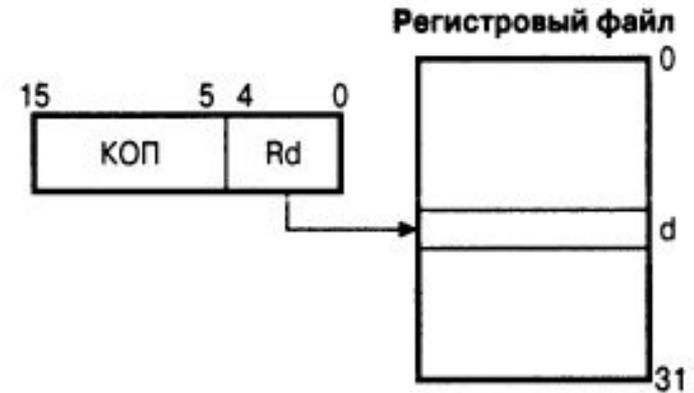


# Регистр SREG



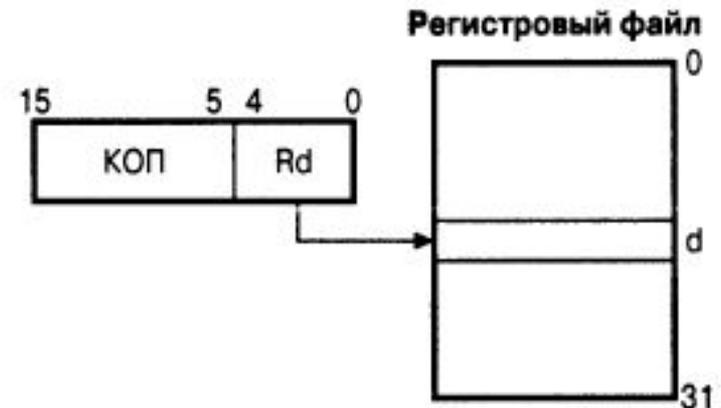
# Способы адресации памяти данных

- Прямая адресация одного регистра общего назначения



Операнд — регистр d (Rd)

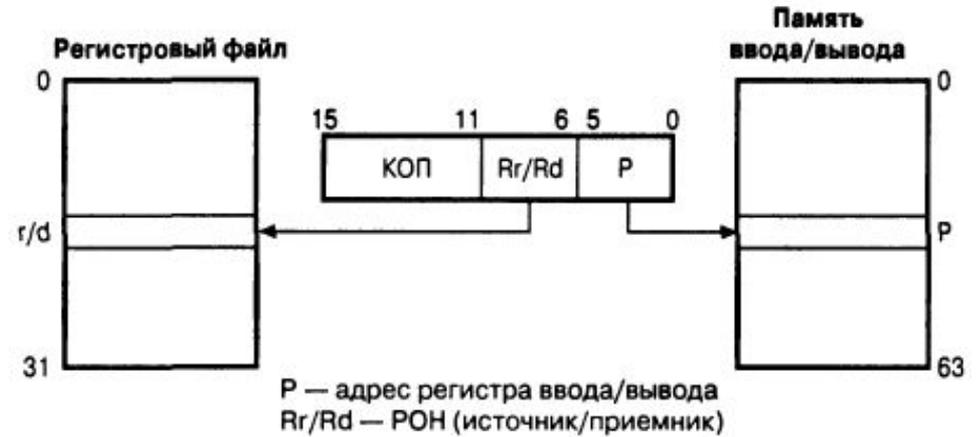
- Прямая адресация двух регистров общего назначения



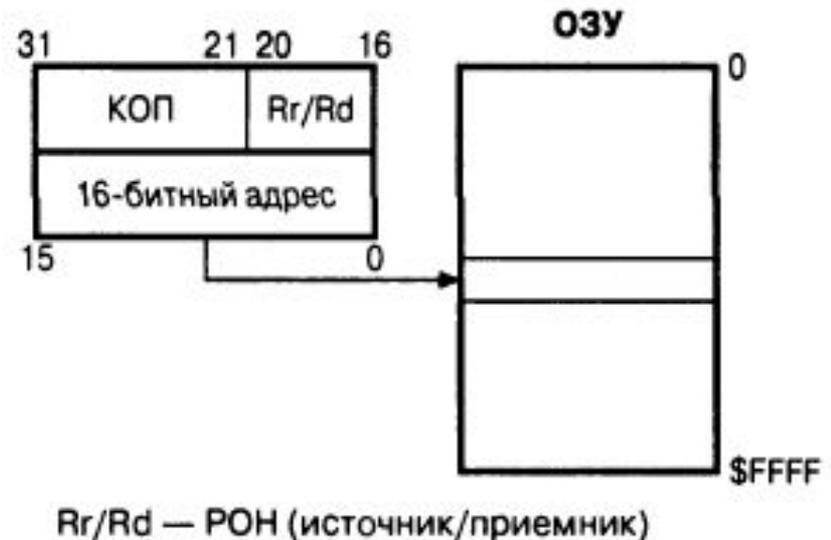
Операнд — регистр d (Rd)

# Способы адресации памяти данных

- Прямая адресация регистра ввода/вывода

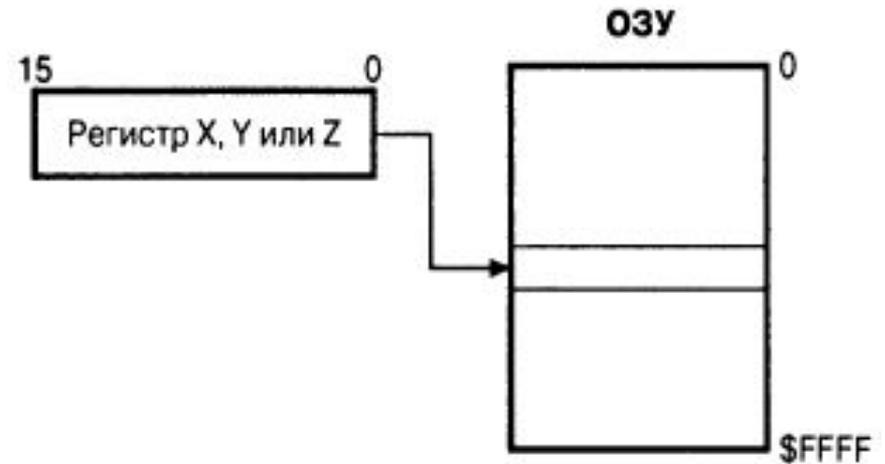


- Прямая адресация ОЗУ

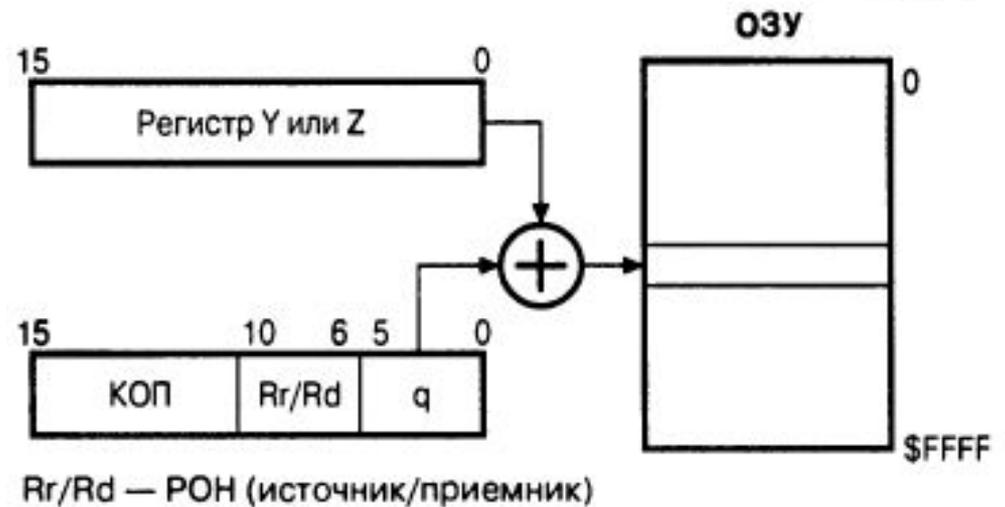


# Способы адресации памяти данных

- Простая косвенная адресация

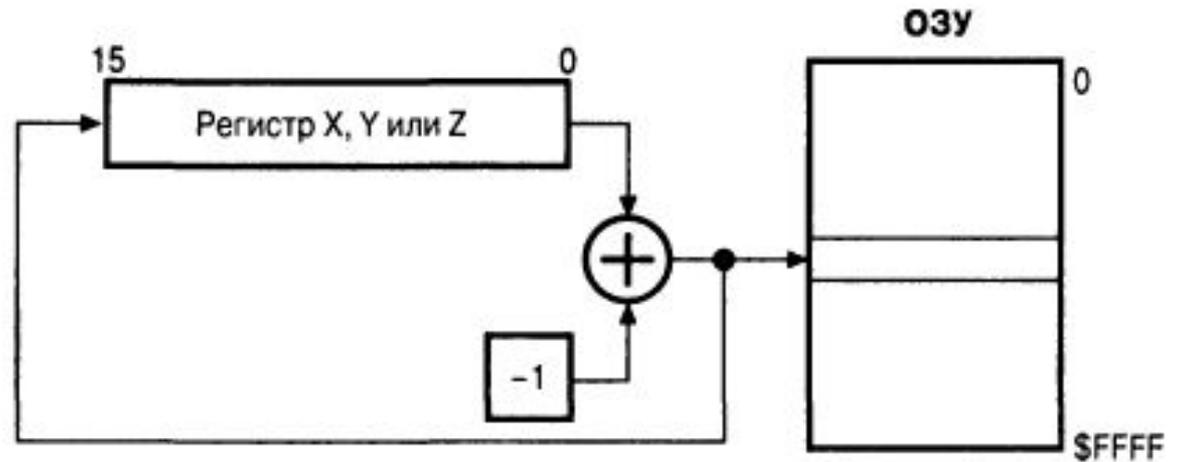


- Относительная косвенная адресация

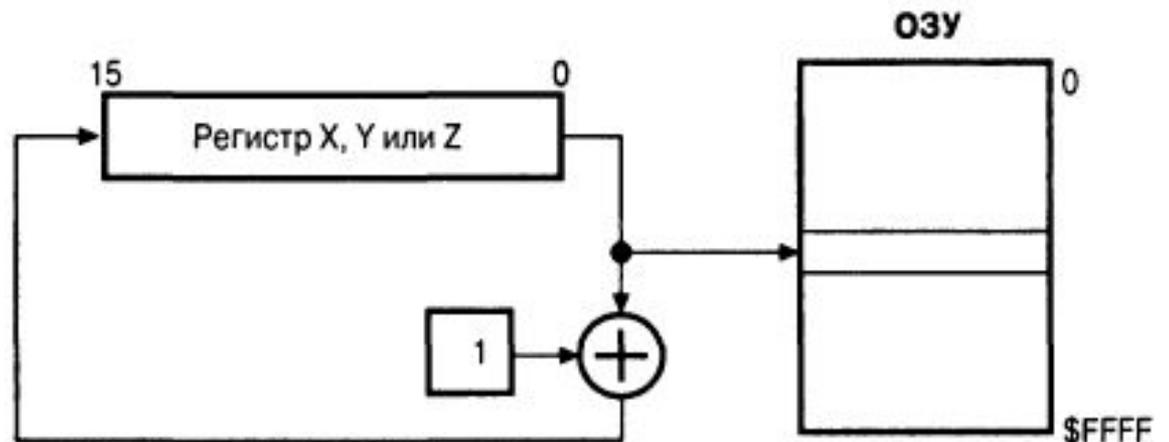


# Способы адресации памяти данных

- Косвенная адресация с преддекрементом



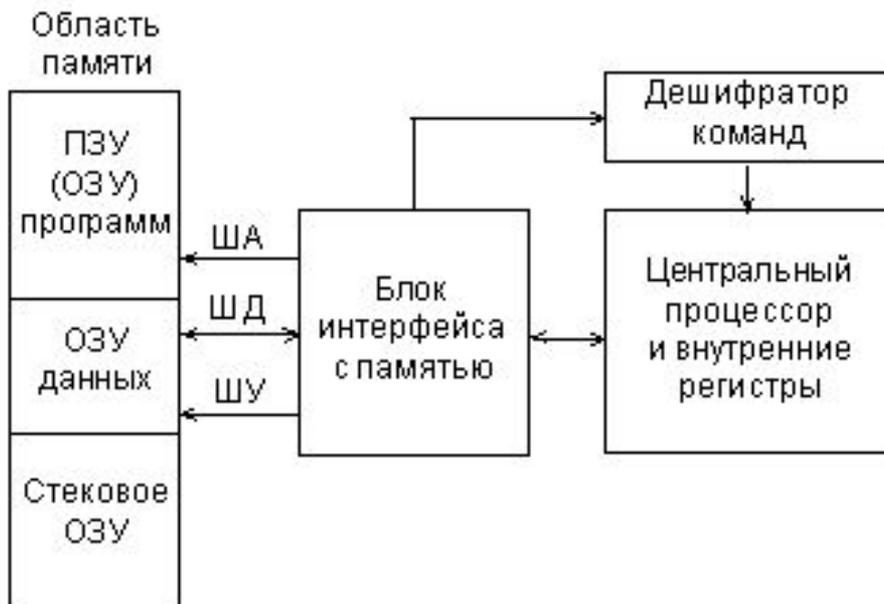
- Косвенная адресация с постинкрементом



# Архитектура Фон-Неймана

С точки зрения организации процессов выборки и исполнения команды в современных МПС применяется одна из двух архитектур: **фон-неймановская (принстонская)** или **гарвардская**.

Основной особенностью **фон-неймановской архитектуры** является использование общей памяти для хранения программ и данных.

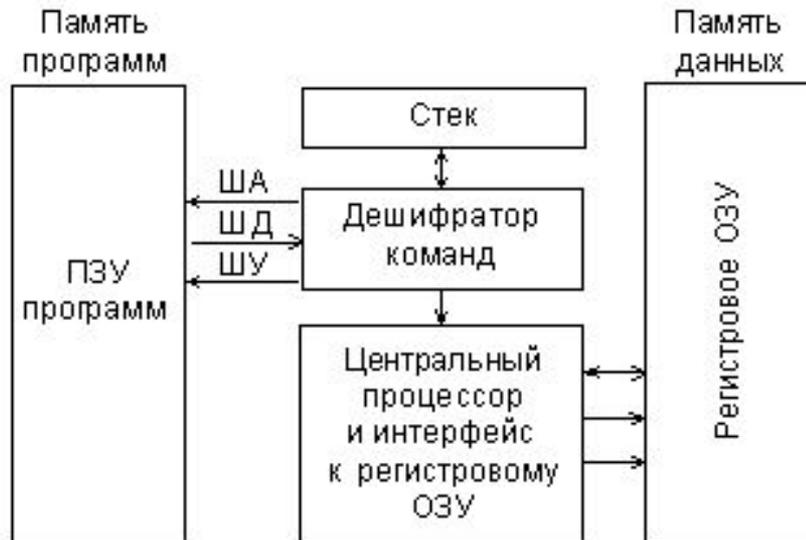


Основное преимущество архитектуры Фон-Неймана – упрощение устройства МПС, так как реализуется обращение только к одной общей памяти. Кроме того, использование единой области памяти позволяло оперативно перераспределять ресурсы между областями программ и данных, что существенно повышало гибкость МПС с точки зрения разработчика программного обеспечения. Размещение стека в общей памяти облегчало доступ к его содержимому. Неслучайно поэтому фон-неймановская архитектура стала основной архитектурой универсальных компьютеров, включая персональные компьютеры.

Рис. 1.2. Структура МПС с фон-неймановской архитектурой.

# Гарвардская архитектура

Основной особенностью **гарвардской архитектуры** является использование отдельных адресных пространств для хранения команд и данных, как показано на рис. 1.3.



Кроме того, гарвардская архитектура обеспечивает потенциально более высокую скорость выполнения программы по сравнению с фон-неймановской за счет возможности реализации параллельных операций. Выборка следующей команды может происходить одновременно с выполнением предыдущей, и нет необходимости останавливать процессор на время выборки команды. Этот метод реализации операций позволяет обеспечивать выполнение различных команд за одинаковое число тактов, что дает возможность более просто определить время выполнения циклов и критичных участков программы.

Рис. 1.3. Структура МПС с гарвардской архитектурой.

# Архитектура с общей шиной

Так же эти два типа архитектуры различаются по количеству используемых шин, и в силу этого обстоятельства они имеют другие названия - **одношинная**, или принстонская, фон-неймановская архитектура и **двухшинная**, или гарвардская, архитектура.

Архитектура с общей шиной (рис. 1.4) распространена гораздо больше, она применяется, например, в персональных компьютерах и в сложных микрокомпьютерах. Архитектура с раздельными шинами (рис. 1.5) применяется в основном в однокристалльных микроконтроллерах.



Рис. 1.4. Архитектура с общей шиной данных и команд.

Архитектура с общей шиной (принстонская, фон-неймановская) проще, она не требует от процессора одновременного обслуживания двух шин, контроля обмена по двум шинам сразу. Наличие единой памяти данных и команд позволяет гибко распределять ее объем между кодами данных и команд. Например, в некоторых случаях нужна большая и сложная программа, а данных в памяти надо хранить не слишком много. В других случаях, наоборот, программа требуется простая, но необходимы большие объемы хранимых данных. Перераспределение памяти не вызывает никаких проблем, главное — чтобы программа и данные вместе помещались в памяти системы. Как правило, в системах с такой архитектурой память бывает довольно большого объема (до десятков и сотен мегабайт). Это позволяет решать самые сложные задачи.

# Архитектура с раздельными шинами данных и команд

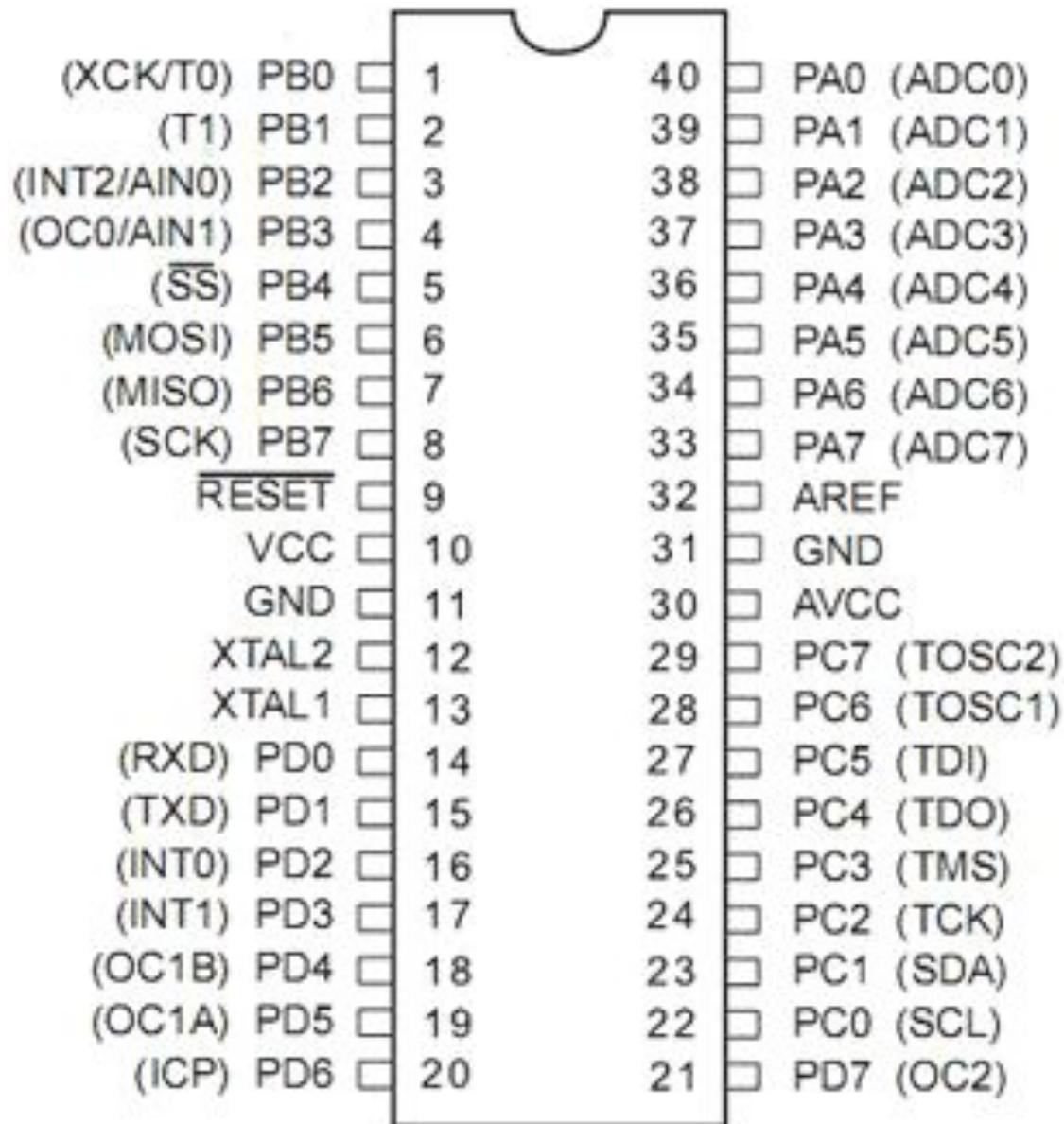


Рис. 1.5. Архитектура с раздельными шинами данных и команд

Архитектура с раздельными шинами данных и команд сложнее, она заставляет процессор работать одновременно с двумя потоками кодов, обслуживать обмен по двум шинам одновременно. Программа может размещаться только в памяти команд, данные — только в памяти данных. Такая узкая специализация ограничивает круг задач, решаемых системой, так как не дает возможности гибкого перераспределения памяти. Память данных и память команд в этом случае имеют не слишком большой объем, поэтому применение систем с данной архитектурой ограничивается обычно не слишком сложными задачами.

- В случае двухшинной архитектуры обмен по обеим шинам может быть независимым, параллельным во времени. Соответственно, структуры шин (количество разрядов кода адреса и кода данных, порядок и скорость обмена информацией и т.д.) могут быть выбраны оптимально для той задачи, которая решается каждой шиной. Поэтому при прочих равных условиях переход на двухшинную архитектуру ускоряет работу микропроцессорной системы, хотя и требует дополнительных затрат на аппаратуру, усложнения структуры процессора. Память данных в этом случае имеет свое распределение адресов, а память команд — свое.
- Проще всего преимущества двухшинной архитектуры реализуются внутри одной микросхемы. В этом случае можно также существенно уменьшить влияние недостатков этой архитектуры. Поэтому основное ее применение — в микроконтроллерах, от которых не требуется решения слишком сложных задач, но зато необходимо максимальное быстродействие при заданной тактовой частоте.

## PDIP



# Таблица векторов прерывания

№	Адрес	Источник	Описание
1	\$0002	INT0	Внешнее прерывание 0
2	\$0004	INT1	Внешнее прерывание 1
3	\$0006	TIMER2 COMP	Совпадение таймера/счетчика T2
4	\$0008	TIMER2 OVF	Переполнение таймера/счетчика T2
5	\$000A	TIMER1 CAPT	Захват таймера/счетчика T1
6	\$000C	TIMER1 COMPA	Совпадение «А» таймера/счетчика T1
7	\$000E	TIMER1 COMPB	Совпадение «В» таймера/счетчика T1
8	\$0010	TIMER1OVF	Переполнение таймера/счетчика T1
9	\$0012	TIMER0OVF	Переполнение таймера/счетчика T0
10	\$0014	SPI, STC	Передача по SPI завершена
11	\$0016	USART, RXC	USART, прием завершен
12	\$0018	USART, UDRE	Регистр данных USART пуст
13	\$001A	USART, TXC	USART, передача завершена
14	\$001C	ADC	Преобразование АЦП завершено
15	\$001E	EE_RDY	EEPROM, готово
16	\$0020	ANA_COMP	Аналоговый компаратор
17	\$0022	TWI	Прерывание от модуля TWI
18	\$0024	INT2	Внешнее прерывание 2
19	\$0026	TIMER0 COMP	Совпадение таймера/счетчика T0
20	\$0028	SPM_RDY	Готовность SPM

# Формирование прерываний в AVR Studio

```
.CSEG  
.ORG $000  
RJMP Reset  
.ORG INT0addr ; Внешнее прерывание 0  
RJMP INT_0;  
.ORG OVF0addr; Переполнение таймера/счетчика T0  
RJMP OVF_0;  
.ORG $030  
Reset:
```

# Формирование прерываний в AVR Studio

```
#include <avr/io.h>
#include <avr/interrupt.h>
ISR(TIMER1_COMPA_vect)
{
}
ISR(INT0_vect)
{
}
int main (void)
```

# Регистры внешних прерываний

## Регистр управления прерываниями – GICR

7	6	5	
INT1	INT0	INT2	
R/W	R/W	R/W	
0	0	0	

## Регистр флагов внешних прерываний – GIFR

7	6	5	
INTF1	INTF0	INTF2	
R/W	R/W	R/W	
0	0	0	

# Регистр управления микроконтроллера – MCUCR

3	2	1	0
ISC11	ISC10	ISC01	ISC00
R/W	R/W	R/W	R/W
0	0	0	0

## Определение условий генерации внешних прерываний

Разряд	Название	Описание		
3,2	ISC11, ISC10	Определяют условие генерации внешнего прерывания INT1 следующим образом:		
		ISC11	ISC10	Условие
		0	0	По НИЗКОМУ уровню на выводе INT1
		1	0	По спадающему фронту сигнала на выводе INT1
		1	1	По нарастающему фронту сигнала на выводе INT1
1,0	ISC01, ISC00	Определяют условие генерации внешнего прерывания INT0 следующим образом:		
		ISC01	ISC00	Условие
		0	0	По НИЗКОМУ уровню на выводе INT0
		1	0	По спадающему фронту сигнала на выводе INT0
		1	1	По нарастающему фронту сигнала на выводе INT0

# Порты таймеров

Таймер	Обозначение	Описание	Вывод порта
T0	T0	Внешний вход таймера T0	PB0
	OC0	Внешний выход таймера T0	PB3
T1	T1	Внешний вход таймера T1	PB1
	OC1A	Внешний выход А таймера T1	PD5
	OC1B	Внешний выход В таймера T1	PD4
T2	TOSC1	Внешний вывод 1 для подключения резонатора таймера T2	PC6
	TOSC2	Внешний вывод 2 для подключения резонатора для таймера T2	PC7
	OC2	Внешний выход таймера T2	PD7

## Регистры таймеров 0 и 2

- счетный регистр TCNTn (Timer CouNT);
- регистр сравнения OCRn (Output Compare Register);
- регистр управления TCCRn (Timer/Count Control Register<sup>18</sup>).

# Режимы работы таймеров

- формирователь временных интервалов;
- счетчик внешних событий;
- широтно-импульсный модулятор.

# Регистр разрешения прерываний TIMSK

Название	Бит	Описание флагов разрешения прерываний
OSIE2	7	По событию «Совпадение» таймера-счетчика T2
TOIE2	6	По переполнению таймера-счетчика T2
TICIE1	5	По событию «Захват» таймера-счетчика T1
OSIE1A	4	По событию «Совпадение А» таймера-счетчика T1
OSIE1B	3	По событию «Совпадение В» таймера-счетчика T1
TOIE1	2	По переполнению таймера-счетчика T1
OSIE0	1	По событию «Совпадение» таймера-счетчика T0

# Регистр флагов прерываний TIFR

Название	Бит	Описание флагов прерываний
OSCF2	7	По событию «Совпадение» таймера-счетчика T2
TOV2	6	По переполнению таймера-счетчика T2
ICF1	5	По событию «Захват» таймера-счетчика T1
OSCF1A	4	По событию «Совпадение А» таймера-счетчика T1
OSCF1B	3	По событию «Совпадение В» таймера-счетчика T1
TOV1	2	По переполнению таймера-счетчика T1
OSCF0	1	По событию «Совпадение» таймера-счетчика T0
TOV0	0	По переполнению таймера-счетчика T0

# регистр управления TCCR0

7	6	5	4	3	2	1	0
FOC0	WGM00	COM01	COM00	WGM01	SC02	SC01	SC00

3, 6	WGMn1:WGMn0	<p>Режим работы таймера</p> <p><i>WGMn1 WGMn0</i></p> <table> <tr> <td>0</td> <td>0</td> <td>Режим работы</td> </tr> <tr> <td>0</td> <td>1</td> <td>Normal</td> </tr> <tr> <td>1</td> <td>0</td> <td>Фазовый ШИМ</td> </tr> <tr> <td>1</td> <td>1</td> <td>СТС (сброс при совпадении)</td> </tr> <tr> <td></td> <td></td> <td>Быстрый ШИМ</td> </tr> </table>	0	0	Режим работы	0	1	Normal	1	0	Фазовый ШИМ	1	1	СТС (сброс при совпадении)			Быстрый ШИМ																					
0	0	Режим работы																																				
0	1	Normal																																				
1	0	Фазовый ШИМ																																				
1	1	СТС (сброс при совпадении)																																				
		Быстрый ШИМ																																				
2...0	CS02...CS00	<p>Управление предделителем.</p> <table> <thead> <tr> <th><i>SC02</i></th> <th><i>SC01</i></th> <th><i>SC00</i></th> <th>Коэффициент предделителя</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Таймер отключен</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Кдел=1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Кдел=8</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Кдел=64</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Кдел=256</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Кдел=1024</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Подключен вход T0 (счет осуществляется по спадающему фронту импульсов)</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Подключен вход T0 (счет осуществляется по нарастающему фронту импульсов)</td> </tr> </tbody> </table>	<i>SC02</i>	<i>SC01</i>	<i>SC00</i>	Коэффициент предделителя	0	0	0	Таймер отключен	0	0	1	Кдел=1	0	1	0	Кдел=8	0	1	1	Кдел=64	1	0	0	Кдел=256	1	0	1	Кдел=1024	1	1	0	Подключен вход T0 (счет осуществляется по спадающему фронту импульсов)	1	1	1	Подключен вход T0 (счет осуществляется по нарастающему фронту импульсов)
<i>SC02</i>	<i>SC01</i>	<i>SC00</i>	Коэффициент предделителя																																			
0	0	0	Таймер отключен																																			
0	0	1	Кдел=1																																			
0	1	0	Кдел=8																																			
0	1	1	Кдел=64																																			
1	0	0	Кдел=256																																			
1	0	1	Кдел=1024																																			
1	1	0	Подключен вход T0 (счет осуществляется по спадающему фронту импульсов)																																			
1	1	1	Подключен вход T0 (счет осуществляется по нарастающему фронту импульсов)																																			

# Normal, CTC

COM01	COM00	Описание
0	0	Нормальная работа порта PB3, OC0 отсоединен.
0	1	Переключение OC0 в противоположное состояние при событии совпадение
1	0	Сброс OC0 при событии совпадение.
1	1	Установка OC0 при событии совпадение.

# Фазовый ШИМ - Phase Correct PWM

## Fast PWM – Быстродействующий ШИМ

COM01	COM00	Описание
0	0	Нормальная работа порта PB3, OC0 отсоединен.
0	1	Зарезервировано.
1	0	Сброс OC0 при событии compare match когда счет вверх, установка при счете вниз (PWM без инверсии).
1	1	Установка OC0 при событии compare match когда счет вверх, сброс при счете вниз (PWM с инверсией).

# Регистры таймера 1 (TC1)

- Счетный регистр TCNT1 (H:L) - 16 p
- Регистр сравнения OCR1A (H:L) - 16 p
- Регистр сравнения OCR1B (H:L) - 16 p
- Регистры управления TCCR1A и TCCR1B - 8p
- Регистр захвата ICR1 (H:L) - 16 p

Bit	7	6	5	4	3	2	1	0	
	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	W	W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

# Аналого-цифровой преобразователь (АЦП)

8-канальный 10-разрядный АЦП последовательного приближения

РА0-РА7 – ADC0-ADC7

Регистры АЦП:

Регистр данных – ADC (ADCL и ADCH)

Регистр управления и состояния – ADCSRA

Регистр выбора мультиплексора – ADMUX

Регистр специальных функций – SFIOR

# Регистр управления и состояния – ADCSRA

Бит	7	6	5	4	3	2	1	0
Название	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0

# Регистр выбора мультиплексора – ADMUX

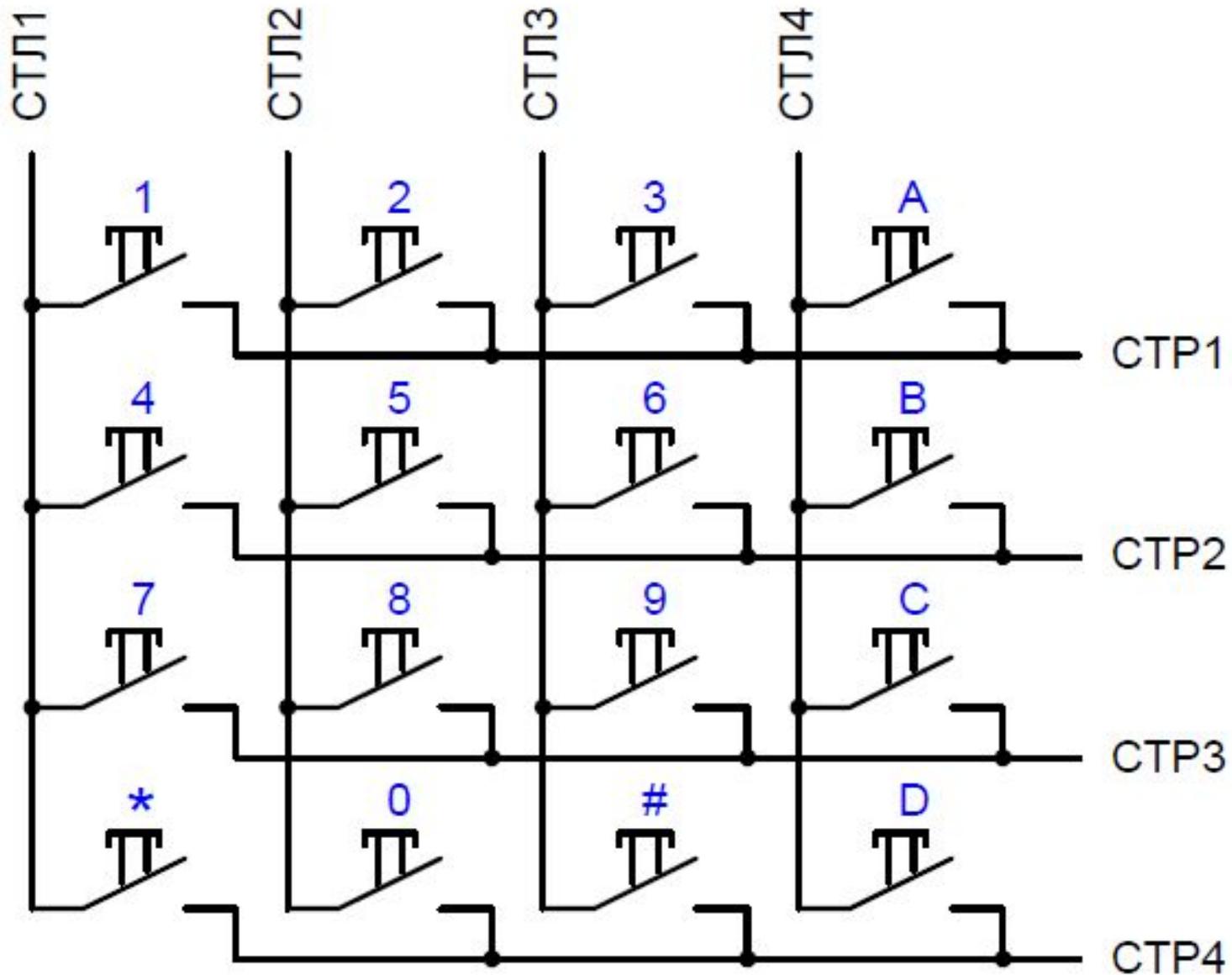
Бит	7	6	5	4	3	2	1	0
Название	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0

REFS1	REFS1	Источник опорного напряжения
0	0	Внешний ИОН, подключенный к выводу AREF; внутренний ИОН отключен
0	1	Напряжение питания
1	0	Зарезервировано
1	1	Внутренний ИОН напряжением 2.56 В

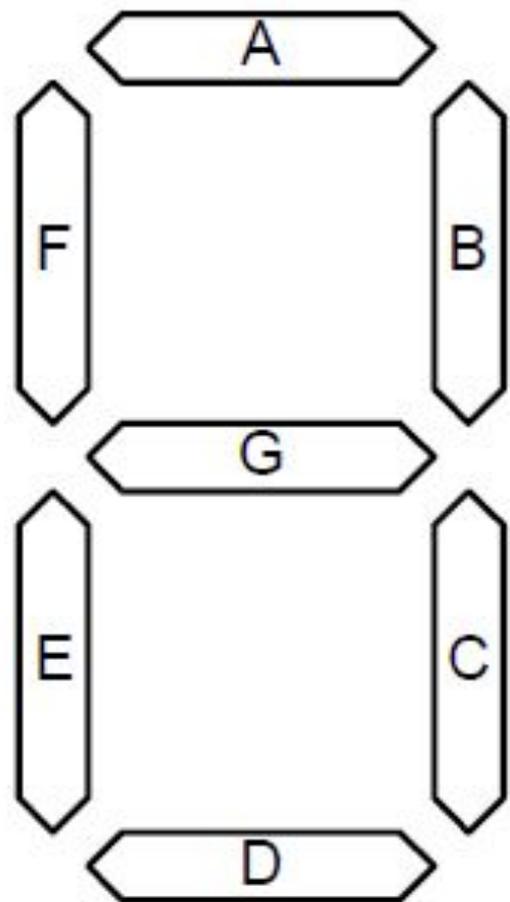
# Регистр специальных функций – SFIOR

7	6	5	
ADTS2	ADTS1	ADTS0	<b>Источник стартового сигнала</b>
0	0	0	Режим непрерывного преобразования
0	0	1	Прерывание от аналогового компаратора
0	1	0	Внешнее прерывание INT0
0	1	1	Событие «Совпадение» таймера/счетчика T0
1	0	0	Переполнение таймера/счетчика T0
1	0	1	Событие «Совпадение В» таймера/счетчика T1
1	1	0	Переполнение таймера/счетчика T1
1	1	1	Событие «Захват» таймера/счетчика T1

# Матричная клавиатура



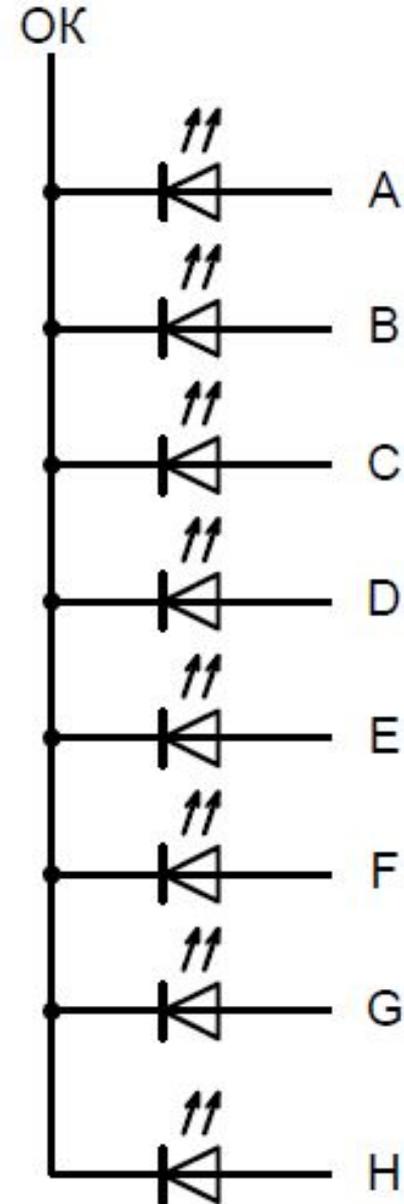
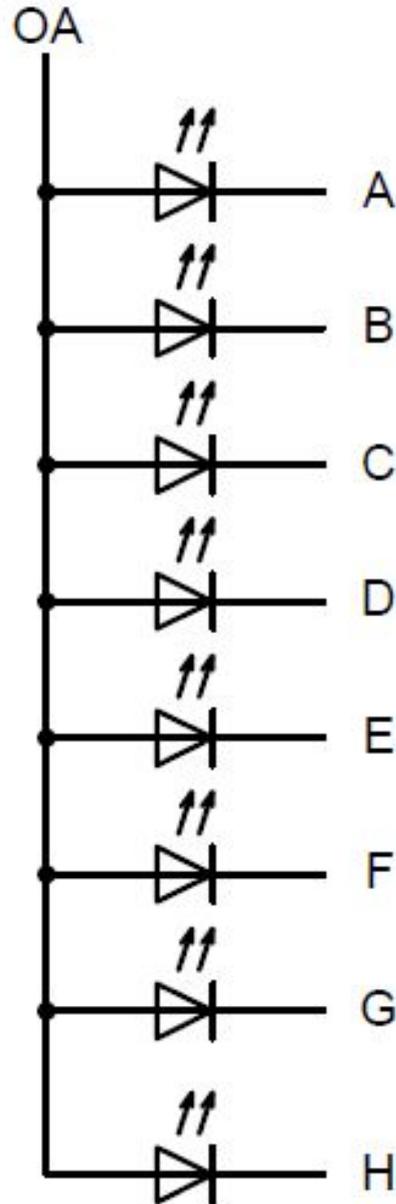
# Семисегментный четырехразрядный индикатор



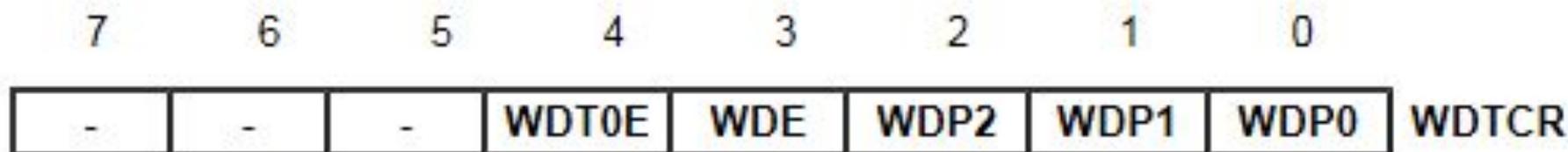
	0		4		8		C
	1		5		9		d
	2		6		A		E
	3		7		b		F

(H)

# Схемы включения светодиодов



# Сторожевой таймер



WDP2	WDP1	WDP0	Длительность цикла сторожевого таймера
0	0	0	16 тыс. циклов
0	0	1	32 тыс. циклов
0	1	0	64 тыс. циклов
0	1	1	128 тыс. циклов
1	0	0	256 тыс. циклов
1	0	1	512 тыс. циклов
1	1	0	1024 тыс. циклов
1	1	1	2048 тыс. циклов

# Аналоговый компаратор

AIN0 – PB2 – неинвертирующий

AIN1 – PB3 - инвертирующий

7            6            5            4            3            2            1            0

ACD	-	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	ACSR
-----	---	-----	-----	------	------	-------	-------	------

ACIS1	ACIS0	Режим прерывания
0	0	Прерывание по переключению выхода компаратора
	01	Зарезервировано
1	0	Прерывание по падающему фронту на выходе компаратора
1	1	Прерывание по нарастающему фронту на выходе компаратора

# ЖКИ дисплей WH1604A

Дисплей (видеопамять DDRAM):

Количество строк: **4**

Количество символов в строке: **16**

Отображаемые символы:

- русские буквы
- латинские буквы
- цифры
- знаки препинания
- пользовательские

Интерфейс дисплея:

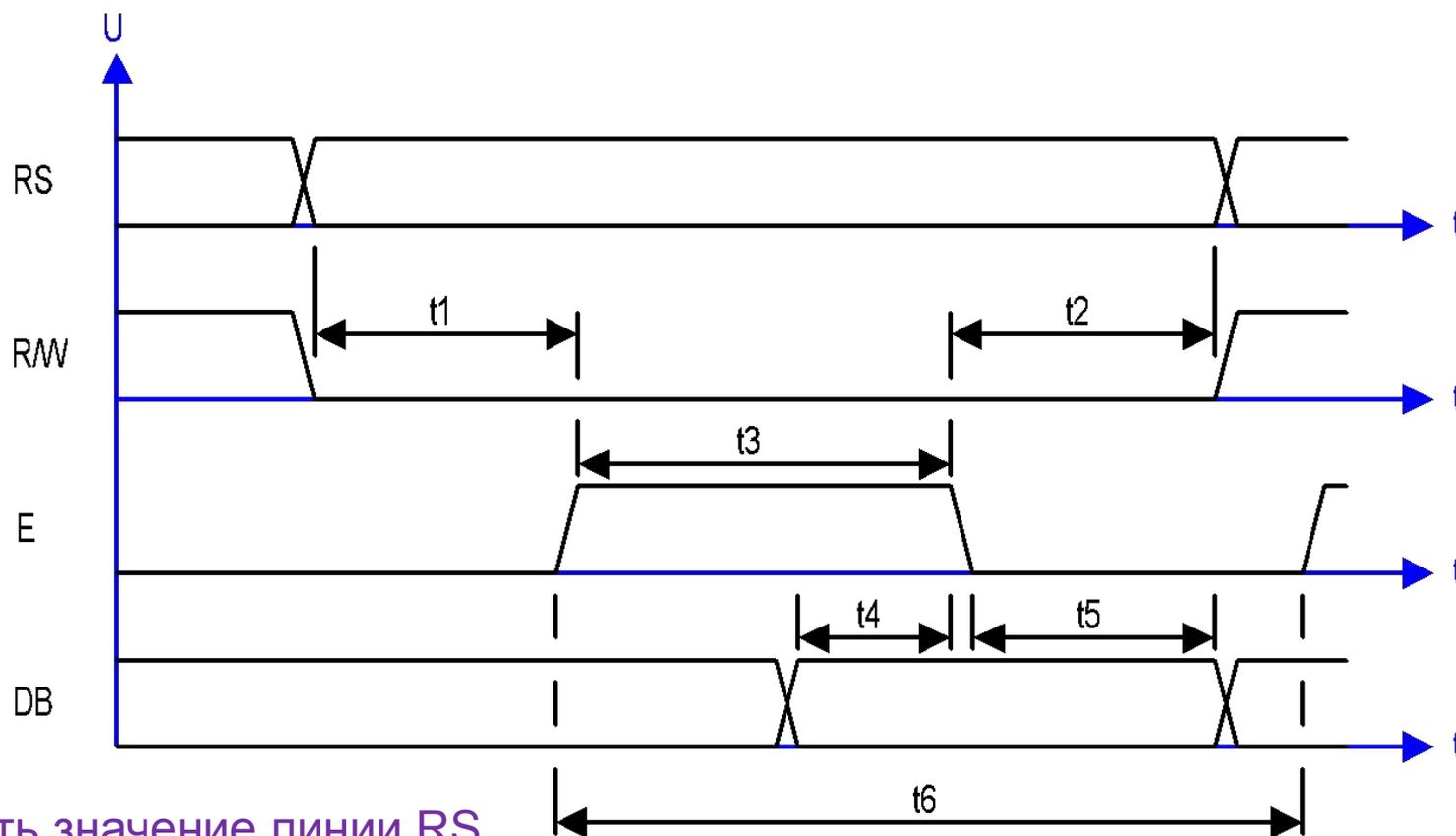
- 3 линии управления (RS, R/W, E)
- 8 линий данных (DB0-DB7), используется DB4-DB7

RS - выбора регистра, 0 - регистр команд и состояния, 1 – регистр данных

R/W - сигнал чтения/записи, 0 – запись, 1 – чтение

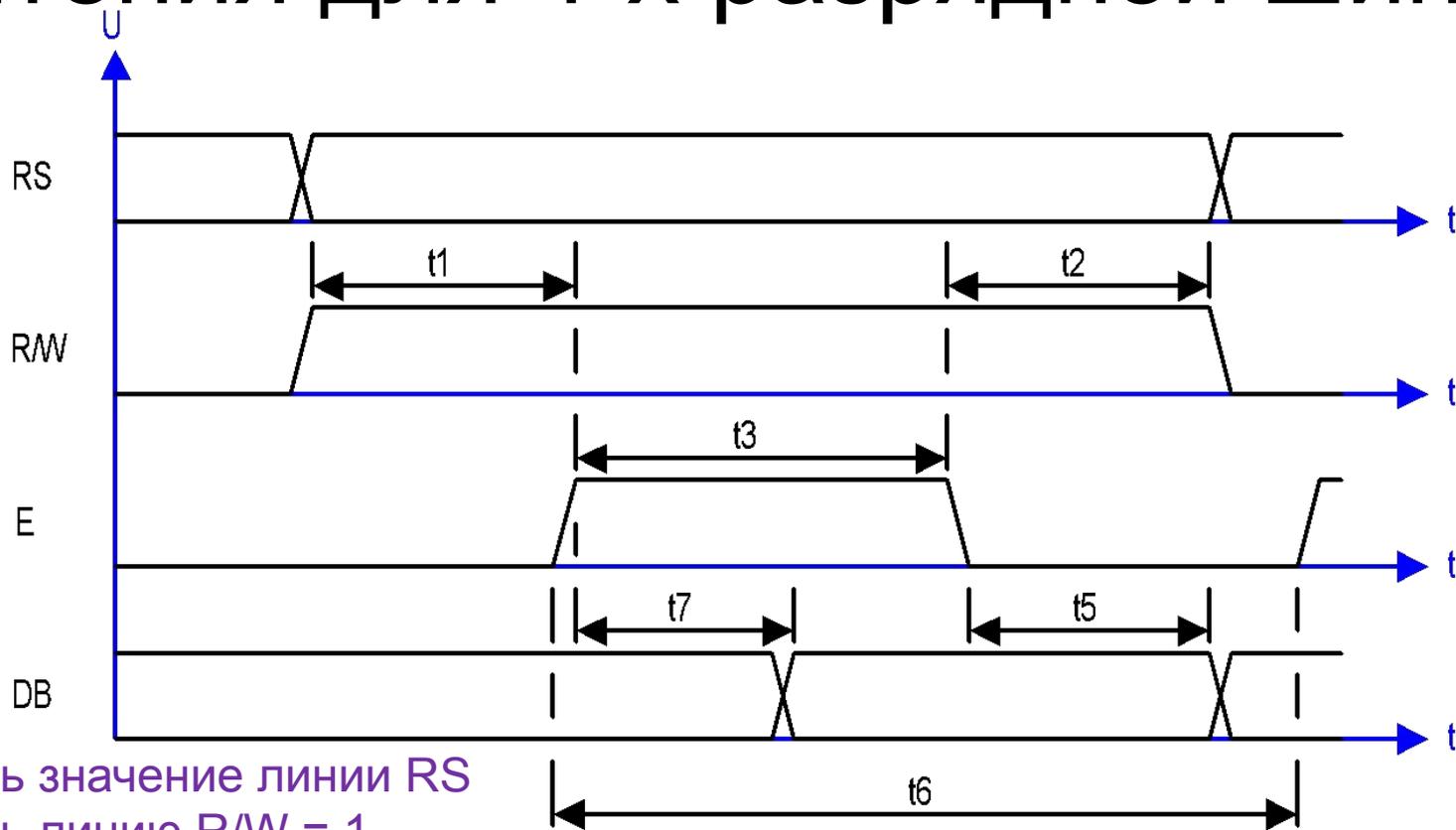
E – сигнал стробирования, по которому осуществляется операция записи или чтения, активный уровень у него логическая 1

# Цикл записи для 4-х разрядной шины



1. Установить значение линии RS
2. Вывести значение старшей тетрады байта данных на линии шины DB4...DB7
3. Установить линию E = 1
4. Установить линию E = 0
5. Вывести значение младшей тетрады байта данных на линии шины DB4...DB7
6. Установить линию E = 1
7. Установить линию E = 0

# Цикл чтения для 4-х разрядной шины



1. Установить значение линии RS
2. Установить линию R/W = 1
3. Установить линию E = 1
4. Считать значение старшей тетрады байта данных с линий шины DB4...DB7
5. Установить линию E = 0
6. Установить линию E = 1
7. Считать значение младшей тетрады байта данных с линий шины DB4...DB7
8. Установить линию E = 0
9. Установить линию R/W = 0

# Временные характеристики

Параметр	Описание	Значение
t1	Установление адреса	Не менее 40 нс
t2	Удержание адреса	Не менее 10 нс
t3	Активность строба	Не менее 230 нс
t4	Установление данных	Не менее 80 нс
t5	Удержание данных	Не менее 10 нс / 5 нс (запись/чтение)
t6	Период строба	Не менее 500 нс
t7	Готовность данных при чтении	Не более 160 нс

# Контроллер HD44780

- регистр данных (DR) RS = 1
- регистр команд (IR) RS = 0
- видеопамять (DDRAM)
- ОЗУ знакогенератора (CGRAM)
- счетчик адреса памяти (AC)  
(IR – чтение 0-6 бит)
- флаг занятости контроллера  
(IR – чтение 7 бит)

# регистр команд

Команда	Биты								Описание	Время выполнения
	D7	D6	D5	D4	D3	D2	D1	D0		
Очистка дисплея	0	0	0	0	0	0	0	1	Очистка дисплея и установка адреса видеопамяи в 0	1.52 мс
Возврат в начало	0	0	0	0	0	0	1	-	Установка адреса видеопамяи в 0 и сдвиг экрана в начало	1.52 мс
Задание режима	0	0	0	0	0	1	I/D	S	Задание направления движения курсора и сдвига экрана	37 мкс
Включение дисплея и курсора	0	0	0	0	1	D	C	B	Включение/выключение дисплея, курсора и задание вида курсора	37 мкс
Сдвиг курсора и экрана	0	0	0	1	S/C	R/L	-	-	Сдвиг курсора и экрана, видеопамяи не изменяется	37 мкс
Задание настроек	0	0	1	DL	N	F	-	-	Задание базовых настроек дисплея	37 мкс
Задание адреса CGRAM	0	1	A5	A4	A3	A2	A1	A0	Задание текущей позиции в памяти знакогенератора	37 мкс
Задание адреса DDRAM	1	A6	A5	A4	A3	A2	A1	A0	Задание текущей позиции в видеопамяи	37 мкс

# Биты регистра команд

I/D	0 - декремент, 1 - инкремент счетчика адреса видеопамати
S	1 - разрешение сдвига экрана
D	0 - дисплей выключен, 1 - включен
C	0 - курсор-подчеркивание выключен, 1 - включен
B	0 - курсор-знакоместо выключен, 1 - включен
S/C	0 - сдвиг курсора, 1 - сдвиг экрана
R/L	0 - сдвиг влево, 1 - сдвиг вправо
DL	0 - шина данных 4 бита, 1 - шина данных 8 бит
N	0 - развертка в 1 строку, 1 - развертка в 2 строки
F	0 - знакоместо 5*8 точек (стандартный режим), 1 - знакоместо 5*10 точек
A6-A0	Адрес

# Значения управляющих флагов после подачи питания

I/D = 1:	режим увеличения сетчика на 1
S = 0:	без сдвига изображения
D/L = 1:	8-ми разрядная шина данных
N = 0:	режим развертки одной строки
F = 0:	символы с матрицей 5 x 8 точек
D = 0:	отображение выключено
C = 0:	курсор в виде подчеркика выключен
B = 0:	курсор в виде мерцающего знакоместа выключен

# Пользовательские символы

Адрес	D7	D6	D5	D4	D3	D2	D1	D0
000	*	*	*	1	0	0	1	0
001	*	*	*	1	0	1	0	1
010	*	*	*	1	0	1	0	1
011	*	*	*	1	1	1	0	1
100	*	*	*	1	0	1	0	1
101	*	*	*	1	0	1	0	1
110	*	*	*	1	0	0	1	0
111	*	*	*	0	0	0	0	0

# Инициализация 4-битной передачи данных

Power on

Ждем больше 15мс, пока Vcc установится до 4,5 В

RS	R $\bar{W}$	DB7	DB6	DB5	DB4
0	0	0	0	1	1

BF не можно прочесть до этой операции  
Function set ( интерфейс – 8-бит )

Ждем больше 4,1 мс

RS	R $\bar{W}$	DB7	DB6	DB5	DB4
0	0	0	0	1	1

BF не можно прочесть до этой операции  
Function set ( интерфейс – 8-бит )

Ждем больше 100 мкс

RS	R $\bar{W}$	DB7	DB6	DB5	DB4
0	0	0	0	1	1

BF не можно прочесть до этой операции  
Function set ( интерфейс – 8-бит )

RS	R/W	DB7	DB6	DB5	DB4
0	0	0	0	1	0
0	0	0	0	1	0
0	0	N	F	*	*
0	0	0	0	0	0
0	0	1	0	0	0
0	0	0	0	0	0
0	0	0	0	0	1
0	0	0	0	0	0
0	0	0	1	I/D	S

*Теперь можно читать BF или выставлять задержки между посылками данных в ЖКИ больше, чем время исполнения команд контроллера ЖКИ*

- Function set : интерфейс 8-бит
- Function set : интерфейс 4-бит, задать число строк и шрифт
- Display off
- Display clear
- Entry mode set

Инициализация закончена

Посл. команд	DB7	DB6	DB5	DB4	
1	0	0	0	0	Готовность ЖК модуля Выдержка 50 мс
2	0	0	1	0	
3	0	0	1	0	Шина 4 бит, LCD - 2 строки
	1	0	0	0	
4	0	0	0	0	Полное выключение дисплея
	1	0	0	0	
5	0	0	0	0	Очистка дисплея
	0	0	0	1	
Выдержка 100мкс					
6	0	0	0	0	Сдвиг курсора вправо
	0	1	1	0	
Выдержка 10мс					
7	0	0	0	0	Включение дисплея, курсор не видим
	1	1	0	0	

# EEPROM

**EEDR** – регистр данных 8 бит

**EEAR (H:L)** – регистр адреса

**EECR** – регистр управления



Запись в EEPROM

1. **EEAR** – задать адрес
2. **EEDR** – поместить данные
3. **EEMWE** <- 1
4. **EEWE** <- 1

Чтение из EEPROM

1. **EEAR** – задать адрес
2. **EERE** <- 1
3. **EEDR** – считать данные