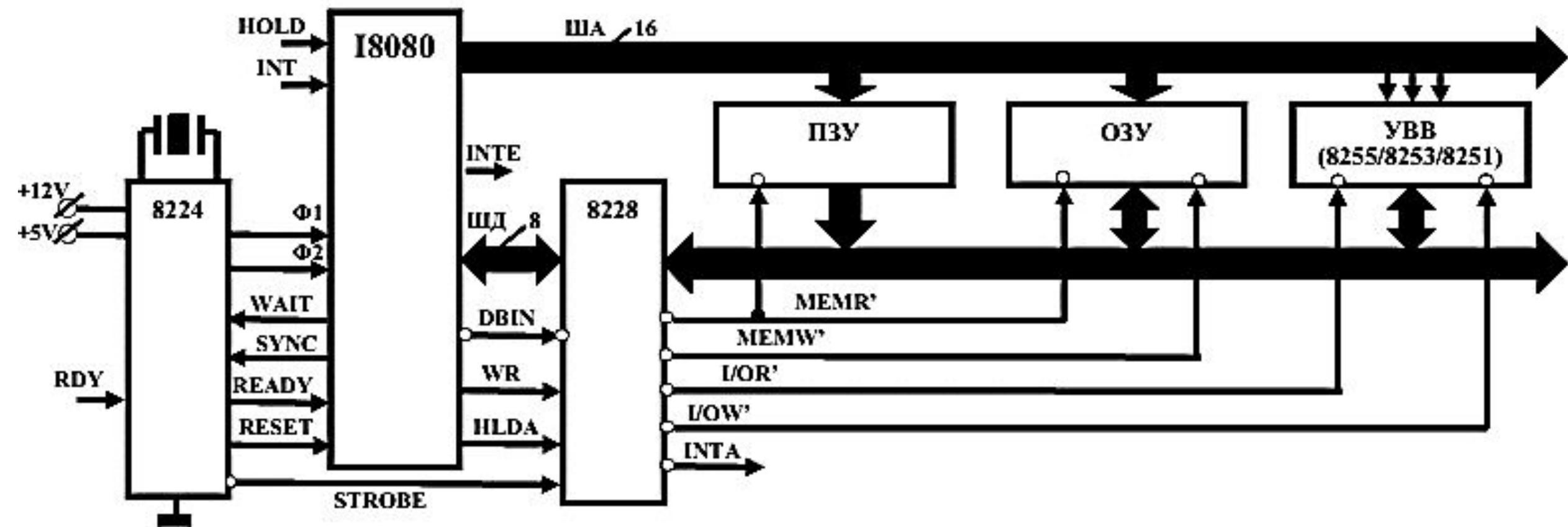


# **Универсальные микропроцессоры**

# МПС на базе I8080



I8080 – микропроцессор

8224 – генератор тактовых импульсов

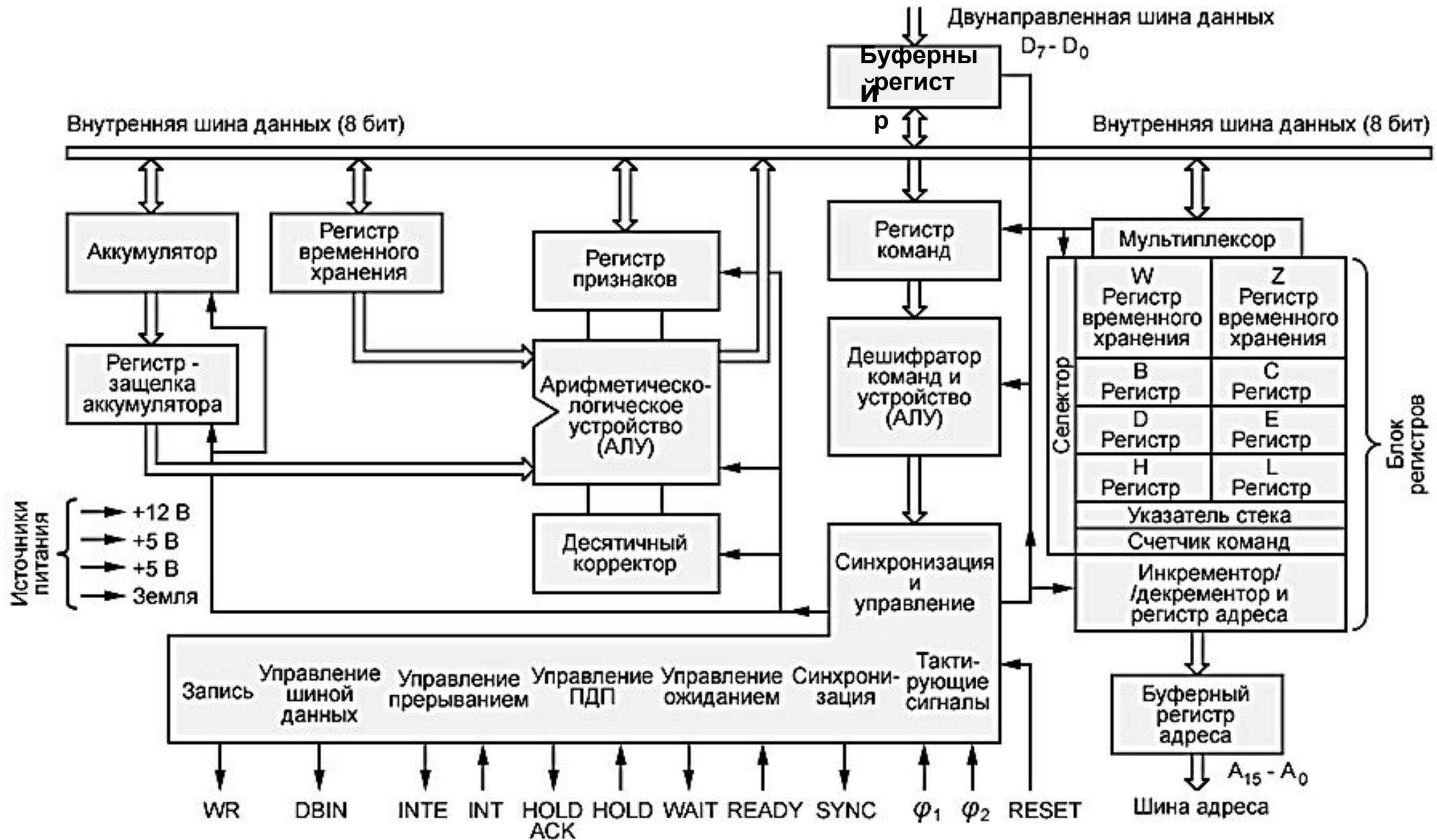
8228 – системный контроллер

ПЗУ (ROM) – память программ

ОЗУ (RAM) – память данных

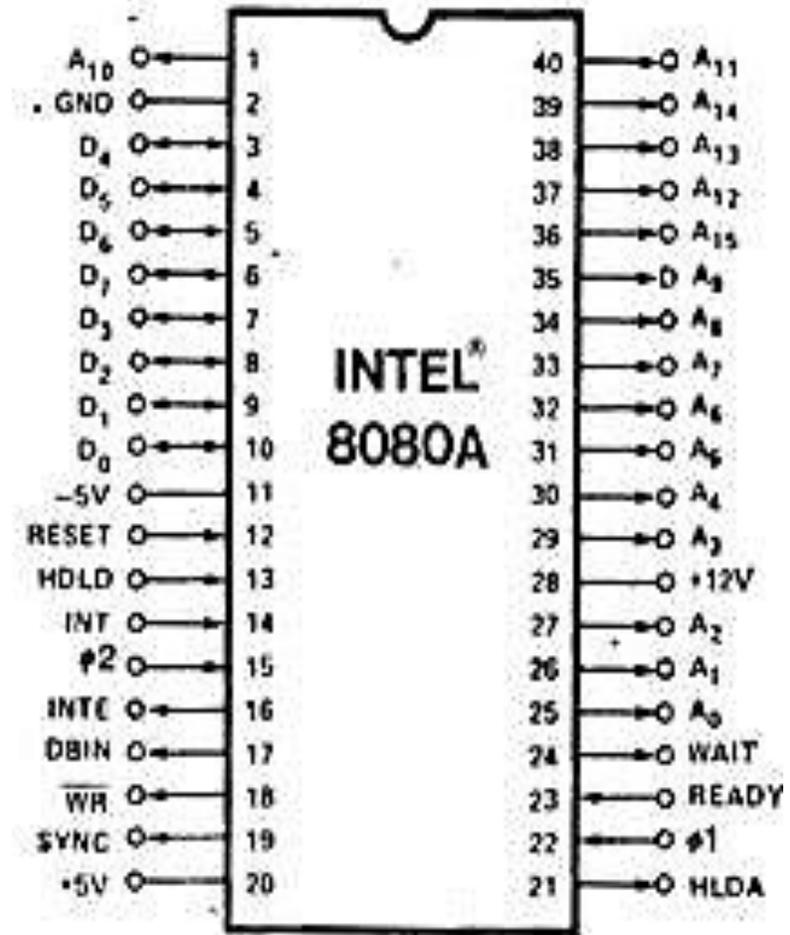
ВВ (IO) – микросхемы ввода-вывода

# Архитектура I8080



# Назначение выводов

**A0 - A15:** выводы шины адреса с тремя состояниями;  
**D0 - D7:** двунаправленная шина данных с тремя состояниями;  
**F1(22) и F2(15):** входы для подачи тактовых сигналов;  
**RESET (12):** сигнал на начальную установку микропроцессора;  
**READY (23):** сигнал готовности данных на шине к вводу в микропроцессор;  
**DBIN (17):** сигнал приема (чтения) данных;  
**WR (18):** сигнал выдачи (записи) данных;  
**SYNC (19):** сигнал синхронизации в начале каждого машинного цикла выполнения команды.  
**HOLD (13):** захват шин, сигнал запроса на прямой доступ к памяти.  
**HLDA (21):** подтверждение захвата шин, (шины адреса и шины данных микропроцессора перешли в высокоомное состояние).  
**WAIT (24):** выход сигнала, указывающего на то, что МП находится в состоянии "Ожидание" (ожидание готовности внешнего устройства или памяти, к обмену данными).



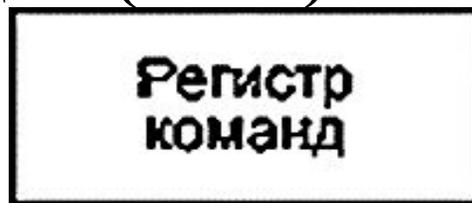
**INT (14):** вход, используемый для аппаратного запроса прерывания;

**INTA (16):** выход сигнала, указывающего состояние внутреннего триггера разрешения прерывания. Если INTA=0, прерывания запрещены. INTA=1 указывает на разрешение прерываний.

## Формат регистра признаков (F)

|    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|
| D7 | D5 | D6 | D4 | D3 | D2 | D1 | D0 |
| S  | Z  | 0  | AC | 0  | P  | 1  | C  |

Регистр команд принимает  
первый байт – код операции (КОП)  
любой команды



Счётчик команд (РС)  
содержит адрес команды  
следующей за выполняемой

Счетчик команд

Указатель стека (SP)  
содержит адрес последнего  
обращения в стек – вершину стека

Указатель стека

Универсальный регистр (А)  
участвует в большинстве  
операций, накапливает  
результат операции в АЛУ



Регистры общего назначения  
(РОН)

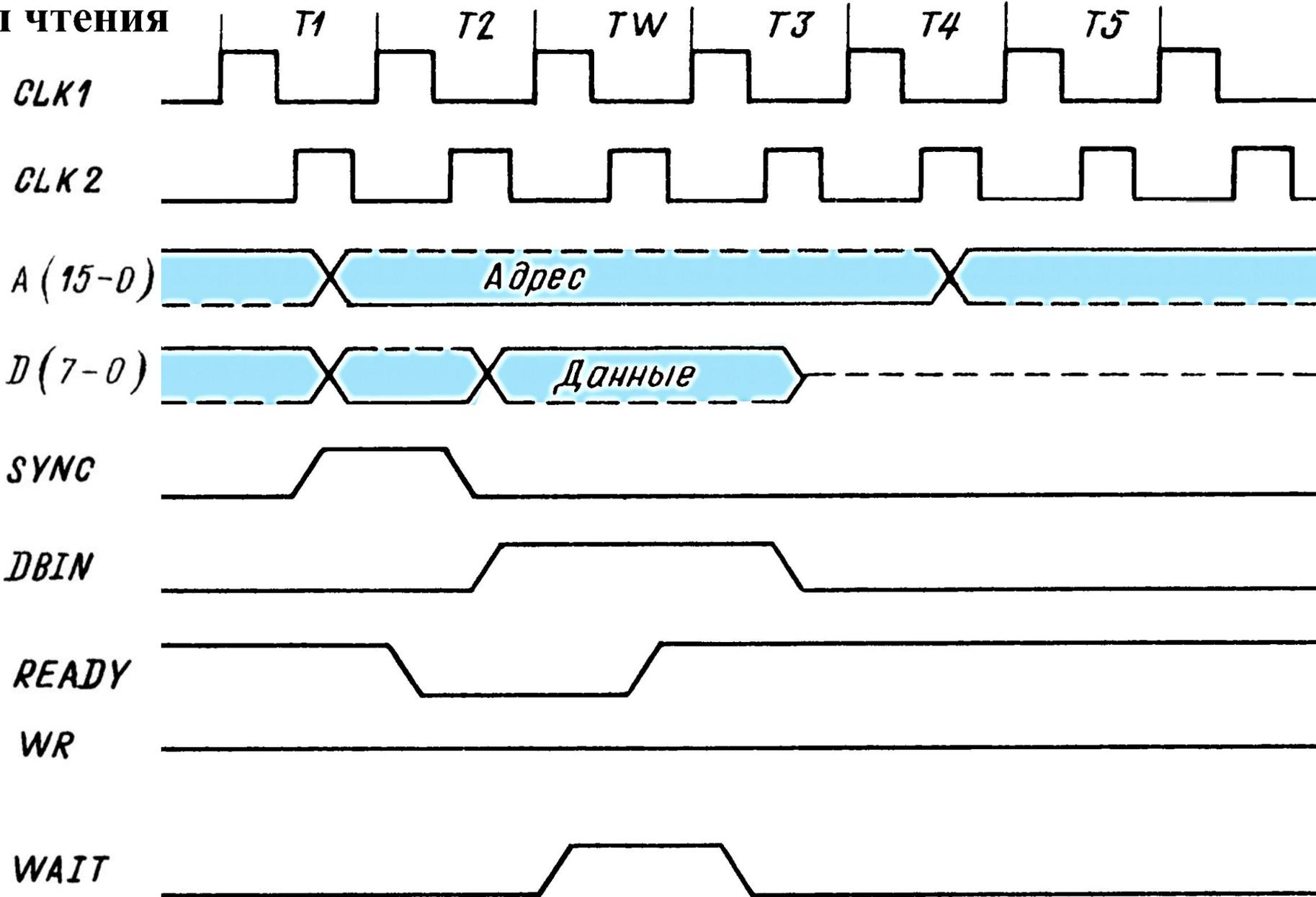
|              |              |
|--------------|--------------|
| В<br>Регистр | С<br>Регистр |
| D<br>Регистр | E<br>Регистр |
| H<br>Регистр | L<br>Регистр |

# Машинные циклы

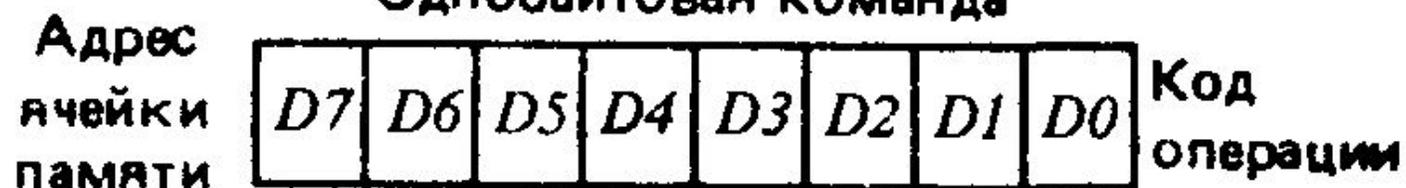
| Машинный цикл                    | Обозначение сигнала слова-состояния процессора |           |              |             |            |           |            |             |
|----------------------------------|--|-----------|--------------|-------------|------------|-----------|------------|-------------|
|                                  | <i>INTA</i>                                    | <i>WO</i> | <i>STACK</i> | <i>HLTA</i> | <i>OUT</i> | <i>M1</i> | <i>INP</i> | <i>MEMR</i> |
| Цикл <i>M1</i> — выборка команды | 0  | 1         | 0            | 0           | 0          | 1         | 0          | 1           |
| Цикл чтения из ЗУ                | 0  | 1         | 0            | 0           | 0          | 0         | 0          | 1           |
| » записи в ЗУ                    | 0  | 0         | 0            | 0           | 0          | 0         | 0          | 0           |
| » чтения из стека                | 0  | 1         | 1            | 0           | 0          | 0         | 0          | 1           |
| Цикл записи в стек               | 0  | 0         | 1            | 0           | 0          | 0         | 0          | 0           |
| Цикл ввода                       | 0  | 1         | 0            | 0           | 0          | 0         | 1          | 0           |
| » вывода                         | 0  | 0         | 0            | 0           | 1          | 0         | 0          | 0           |
| » прерывания                     | 1  | 1         | 0            | 0           | 0          | 1         | 0          | 0           |
| » останова                       | 0  | 1         | 0            | 1           | 0          | 0         | 0          | 1           |
| » прерывания при останове        | 1  | 1         | 0            | 1           | 0          | 1         | 0          | 0           |

# Временные диаграммы работы МП

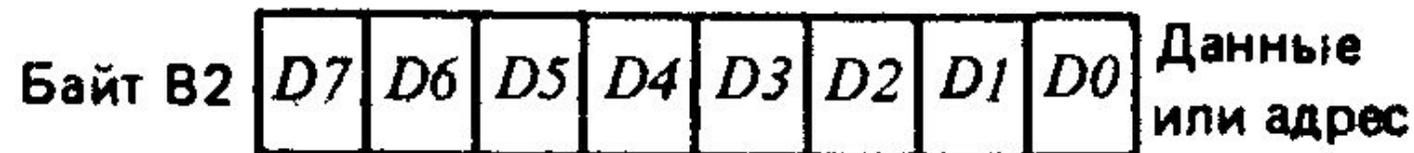
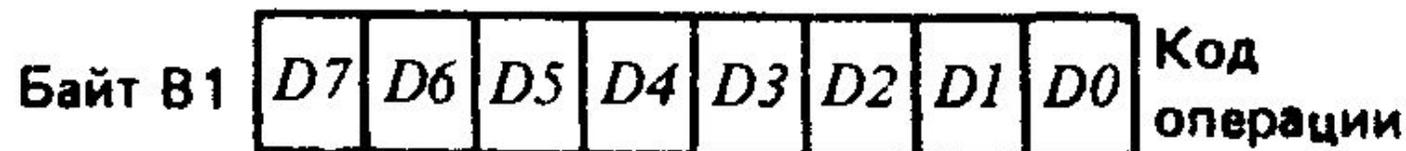
Цикл чтения



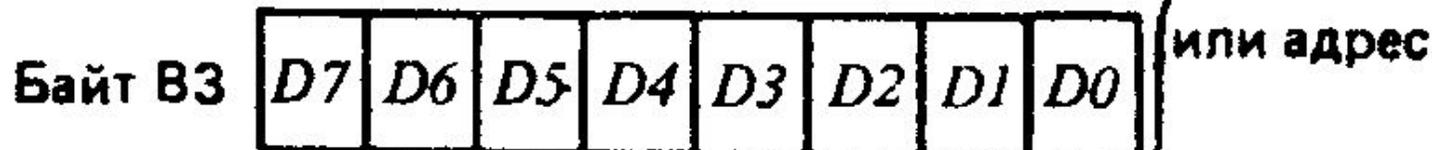
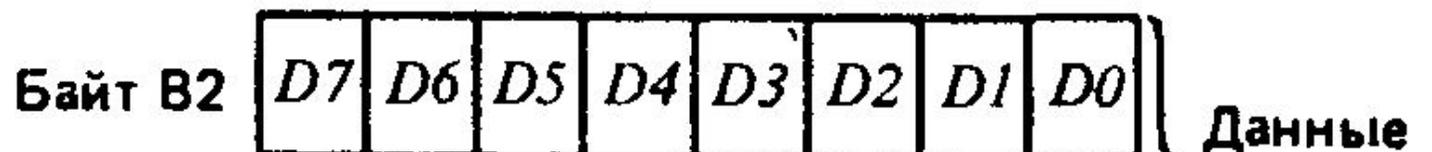
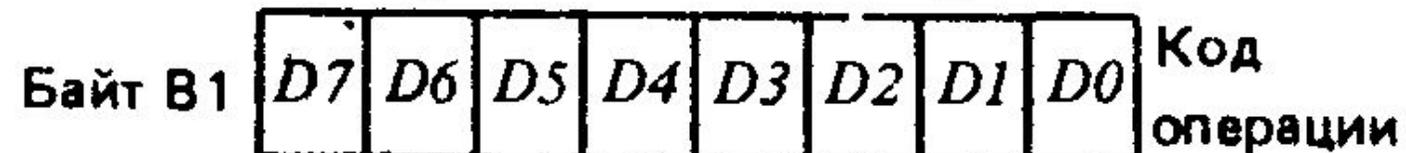
### Однобайтовая команда



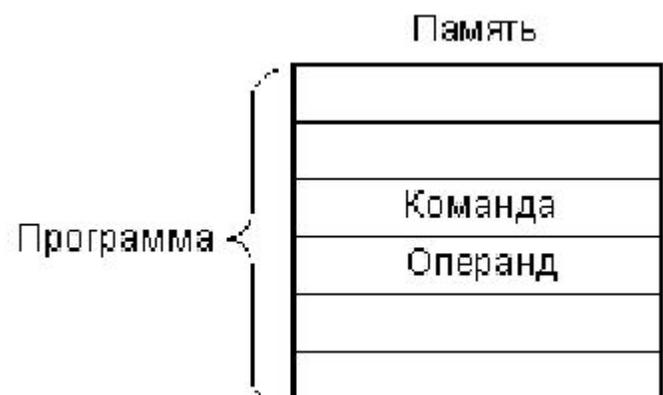
### Двухбайтовая команда



### Трехбайтовая команда



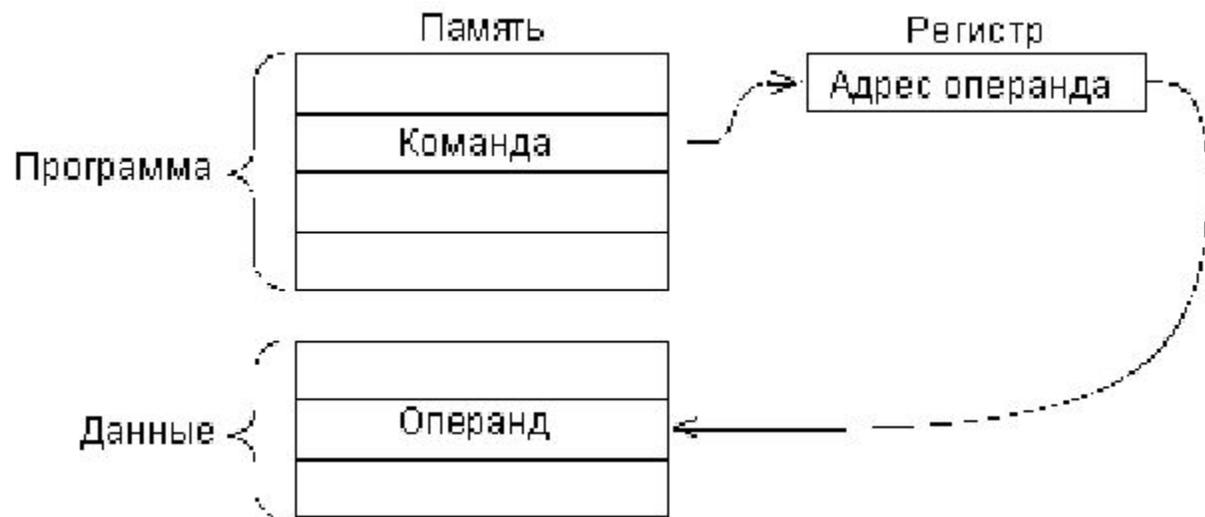
# Способы адресации



Непосредственная



Прямая

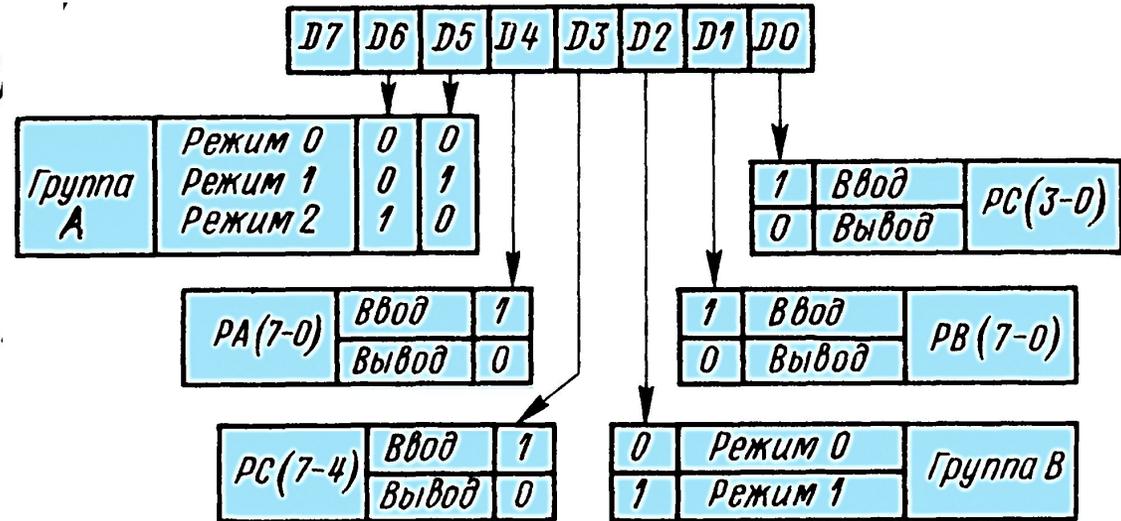
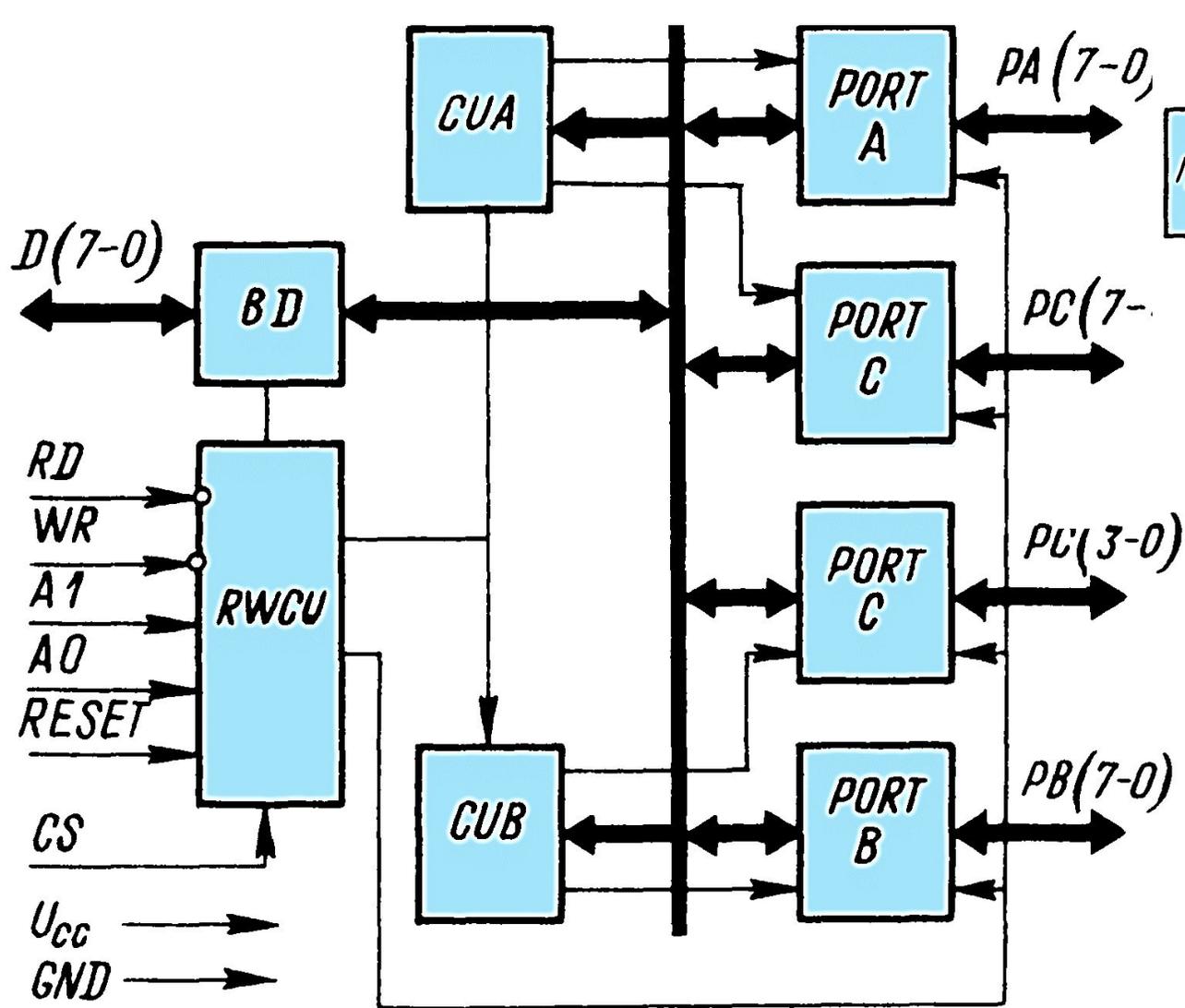


Косвенная



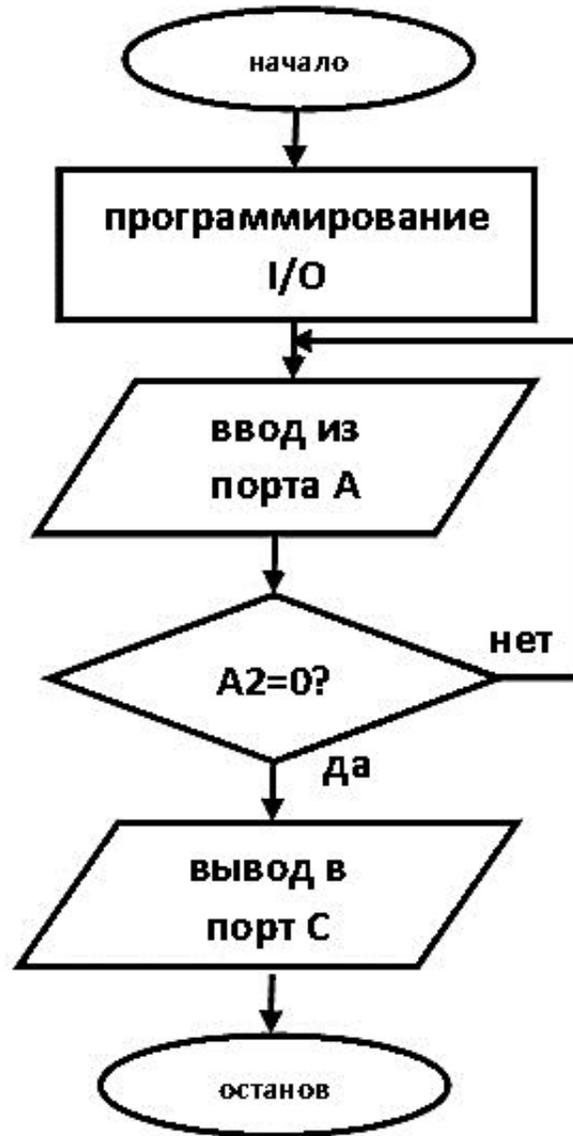
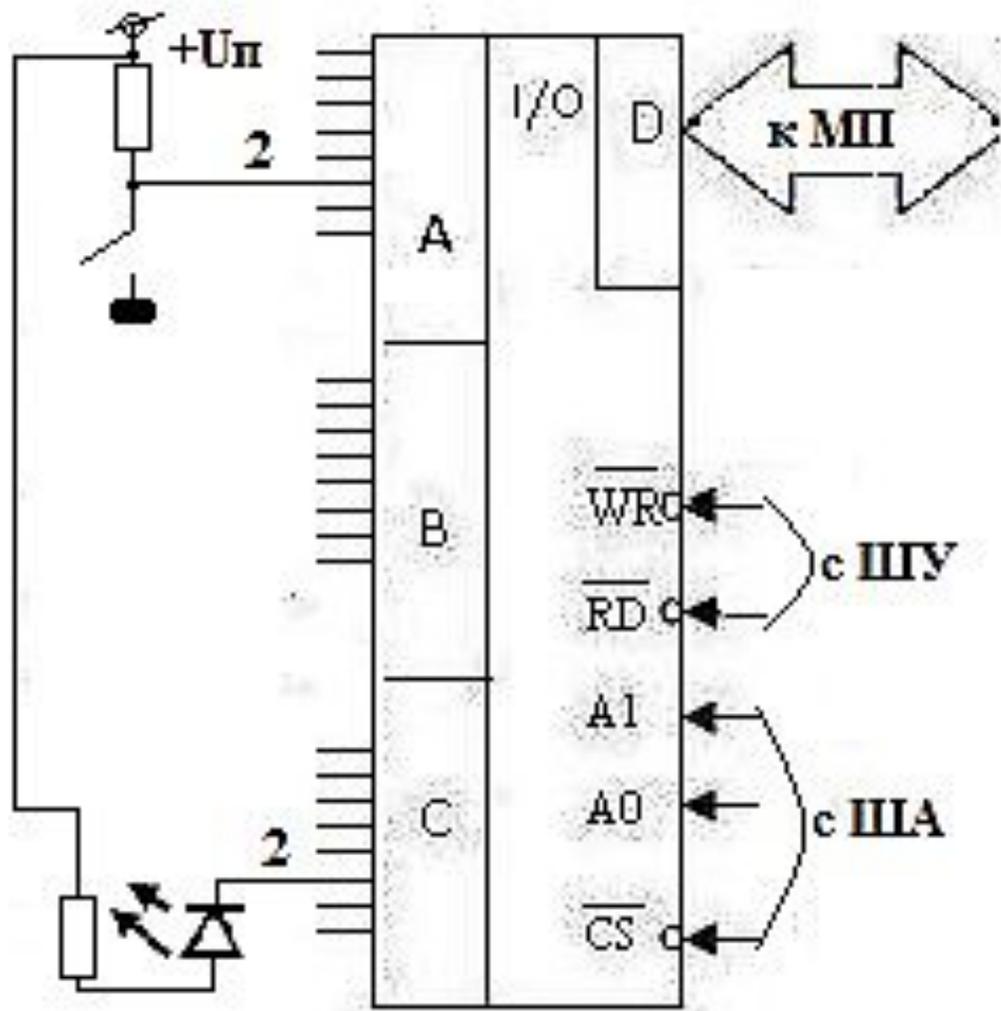
Регистровая

# БИС параллельного интерфейса I8255



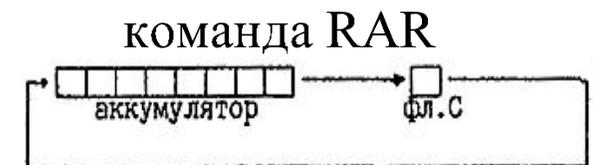
| A1 | A0 | D      |
|----|----|--------|
| 0  | 0  | port A |
| 0  | 1  | port B |
| 1  | 0  | port C |
| 1  | 1  | RCW    |

# Подключение контактного датчика

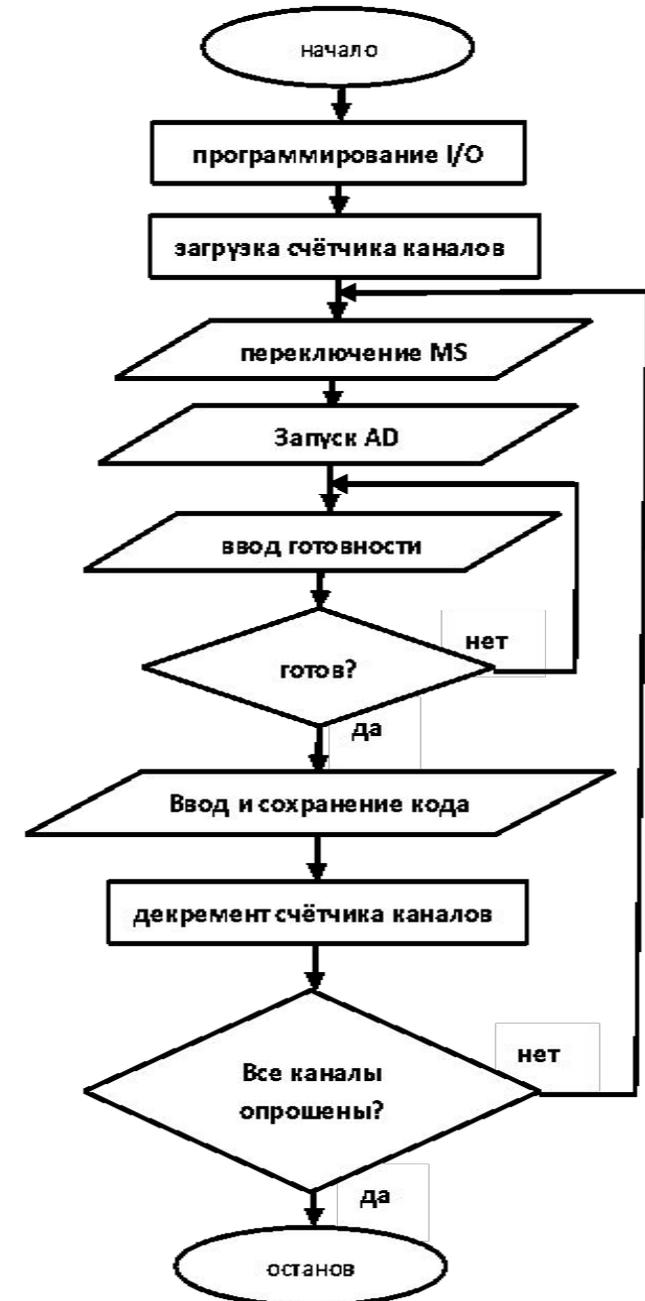
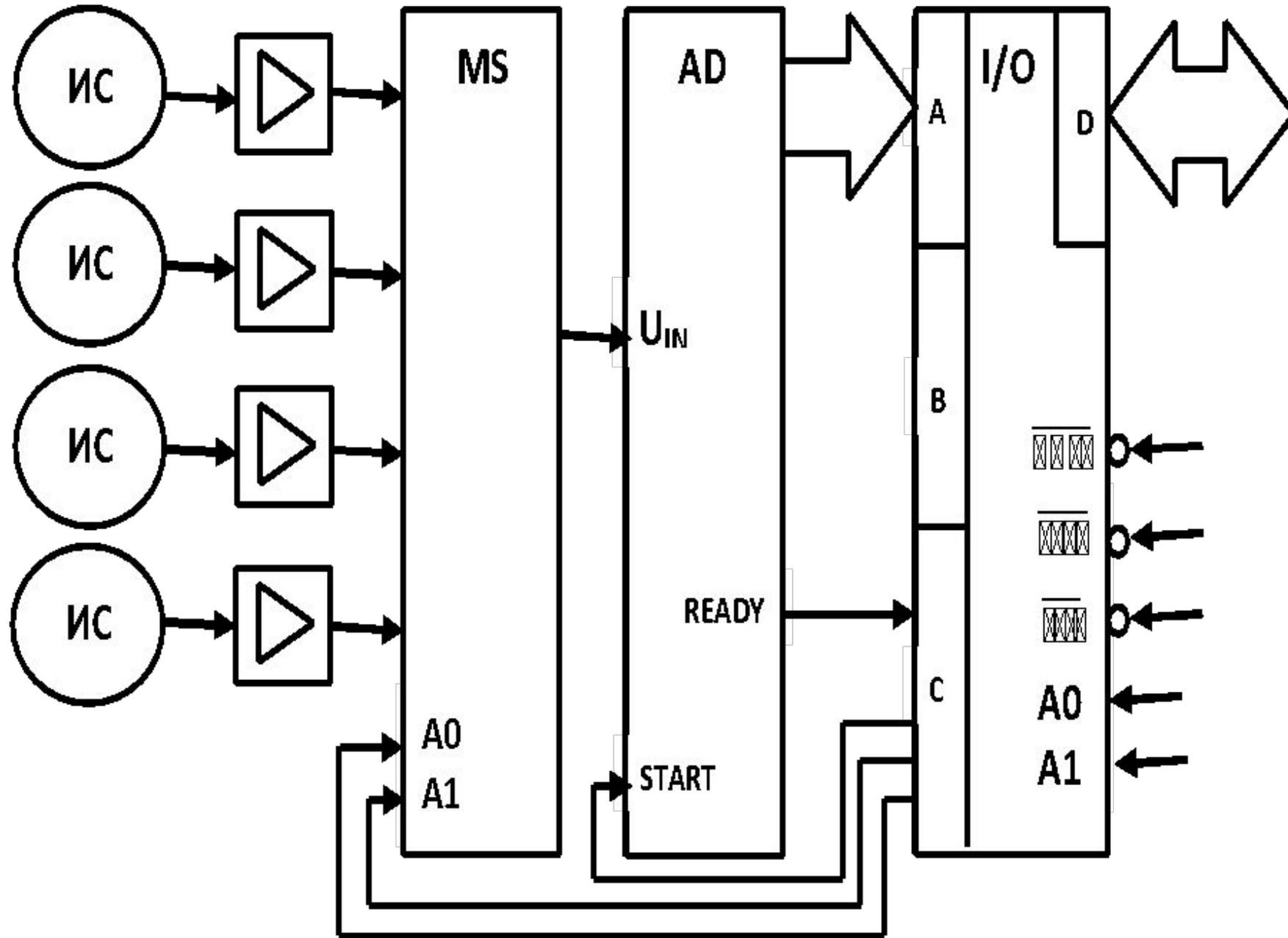


```

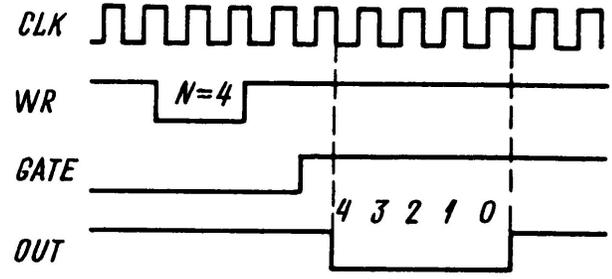
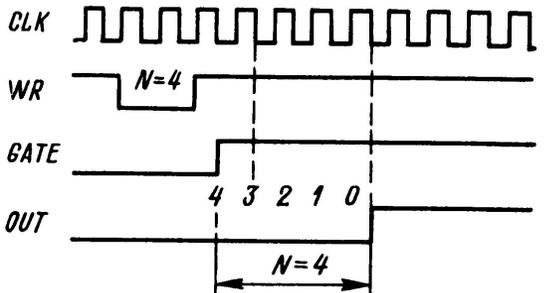
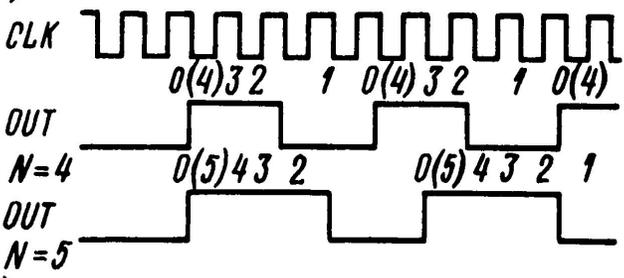
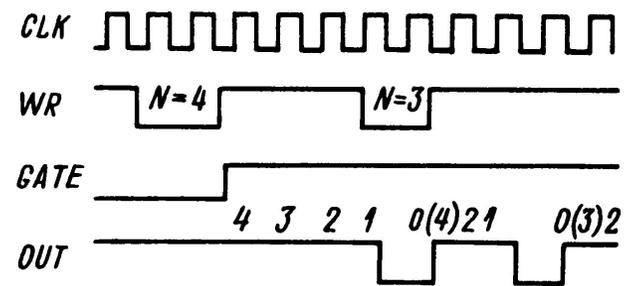
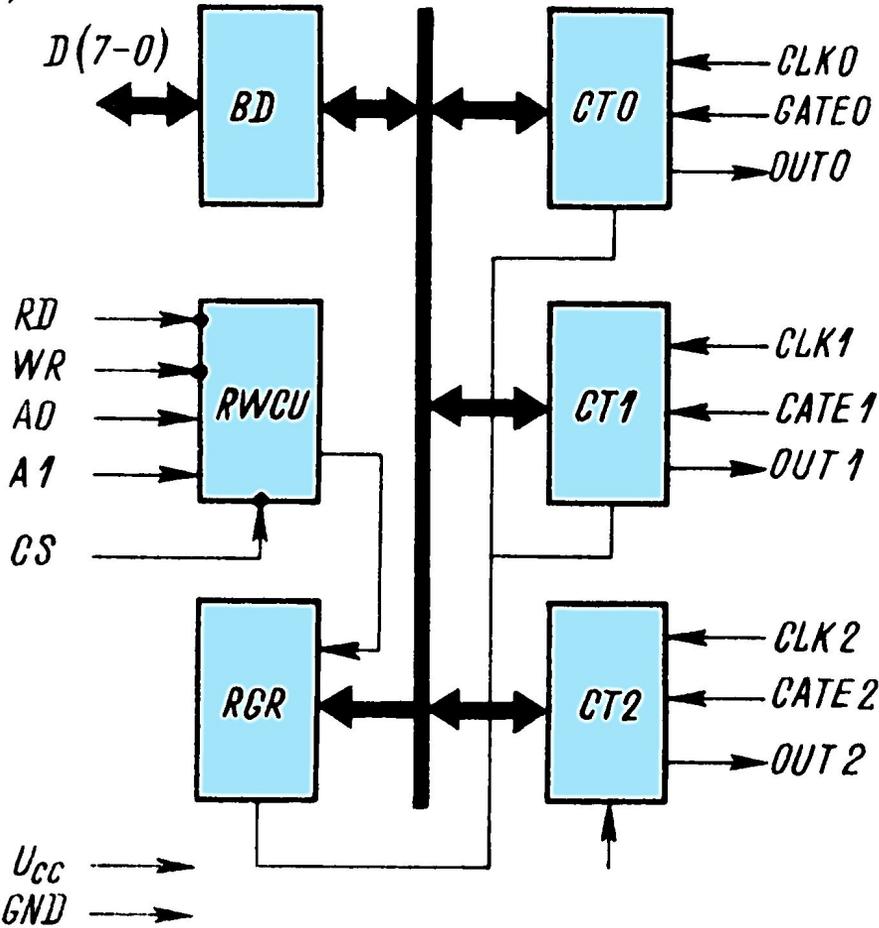
MVI A, CW
OUT RCW
INP: IN portA
    RAR
    RAR
    RAR
    IC INP
    OUT portC
    ANI 04
    JNZ INP
  
```



# Подключение аналоговых датчиков к МП



# БИС программируемого таймера I8253



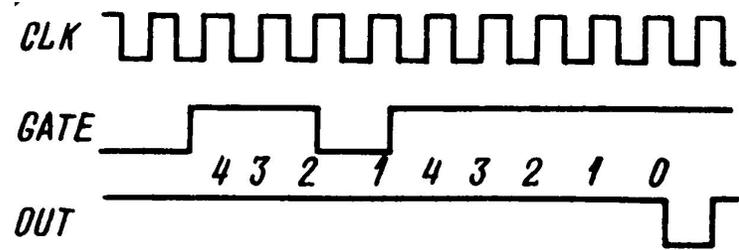
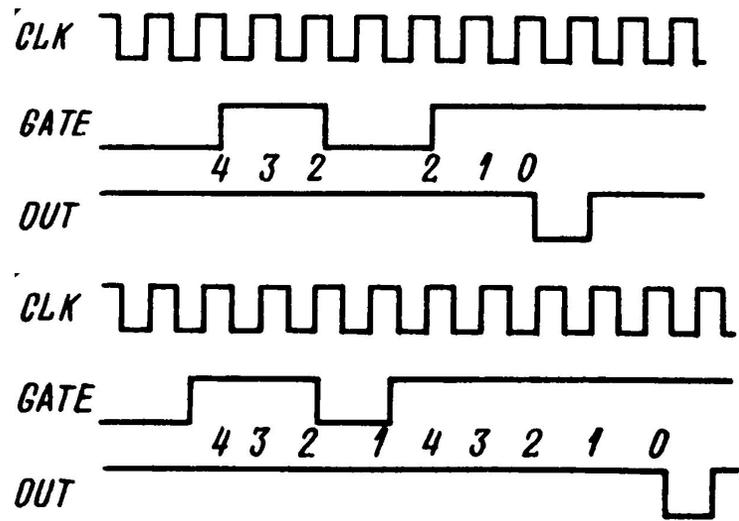
|                    |   |
|--------------------|---|
| Двоичный код       | 0 |
| Двоично-десятичный | 1 |

|    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|----|----|----|----|----|----|----|----|

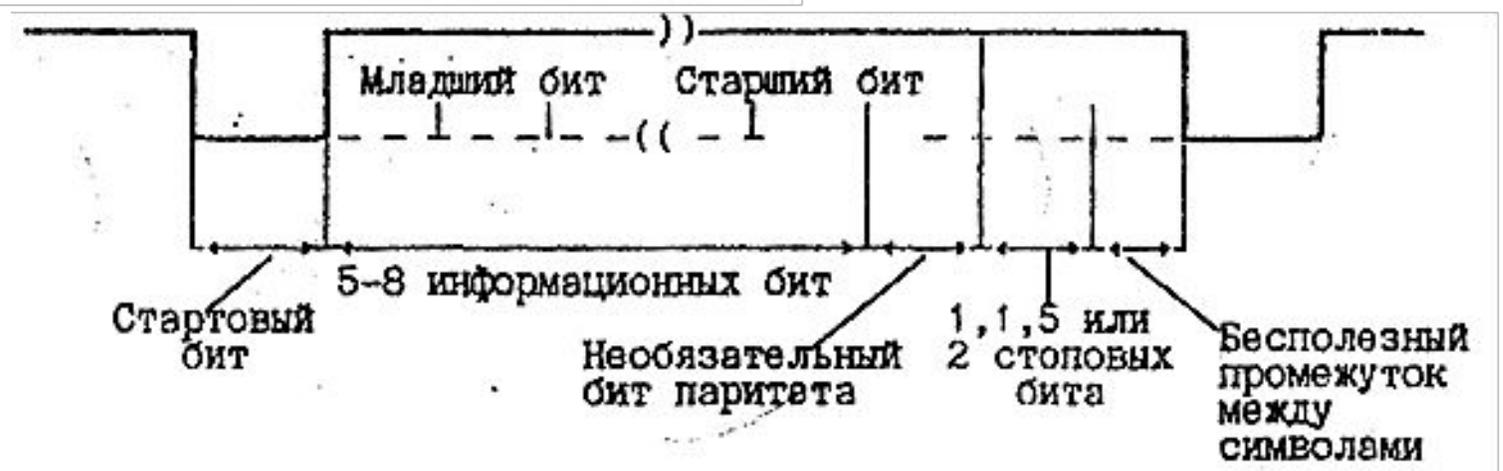
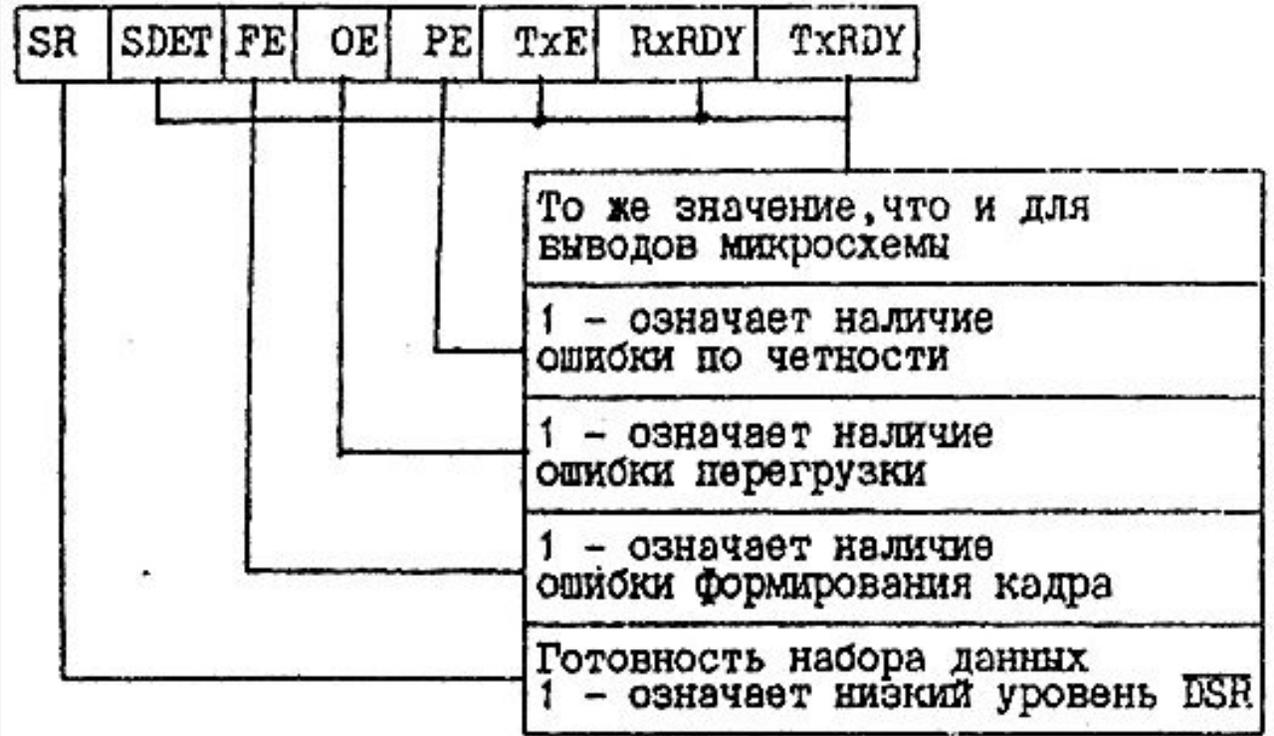
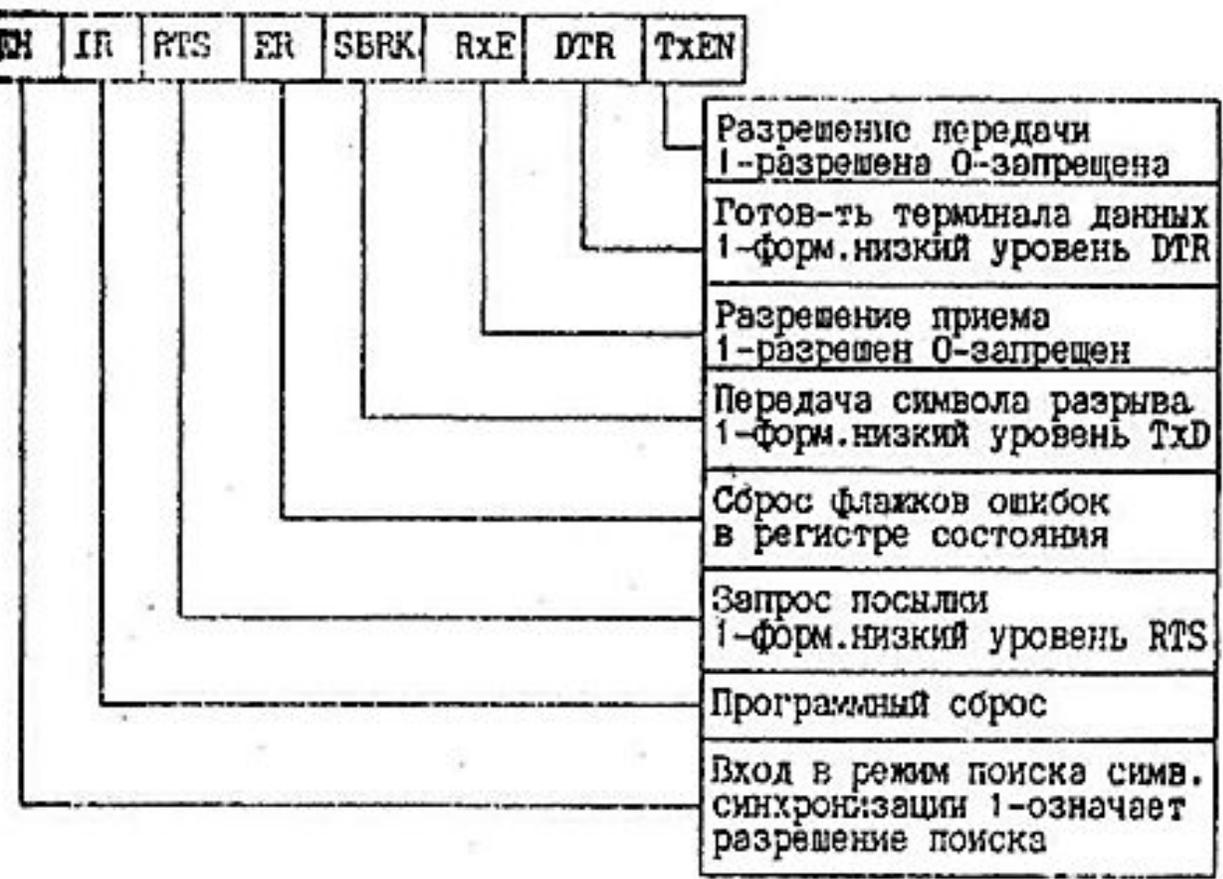
|        |   |   |
|--------|---|---|
| CT0    | 0 | 0 |
| CT1    | 0 | 1 |
| CT2    | 1 | 0 |
| Запрет | 1 | 1 |

|   |   |   |         |
|---|---|---|---------|
| 0 | 0 | 0 | Режим 0 |
| 0 | 0 | 1 | Режим 1 |
| X | 1 | 0 | Режим 2 |
| X | 1 | 1 | Режим 3 |
| 1 | 0 | 0 | Режим 4 |
| 1 | 0 | 1 | Режим 5 |

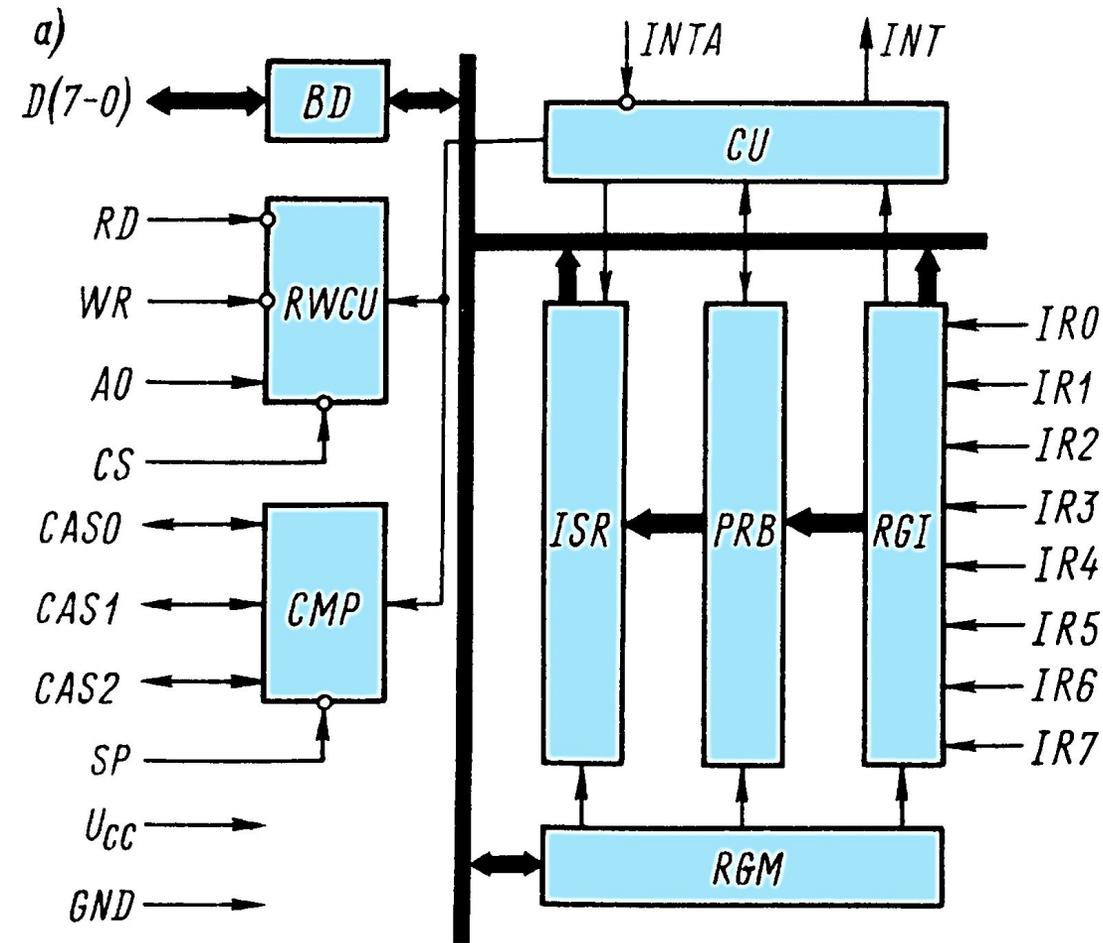
|              |   |   |
|--------------|---|---|
| Защелкивание | 0 | 0 |
| Младший байт | 0 | 1 |
| Старший байт | 1 | 0 |
| Два байта    | 1 | 1 |





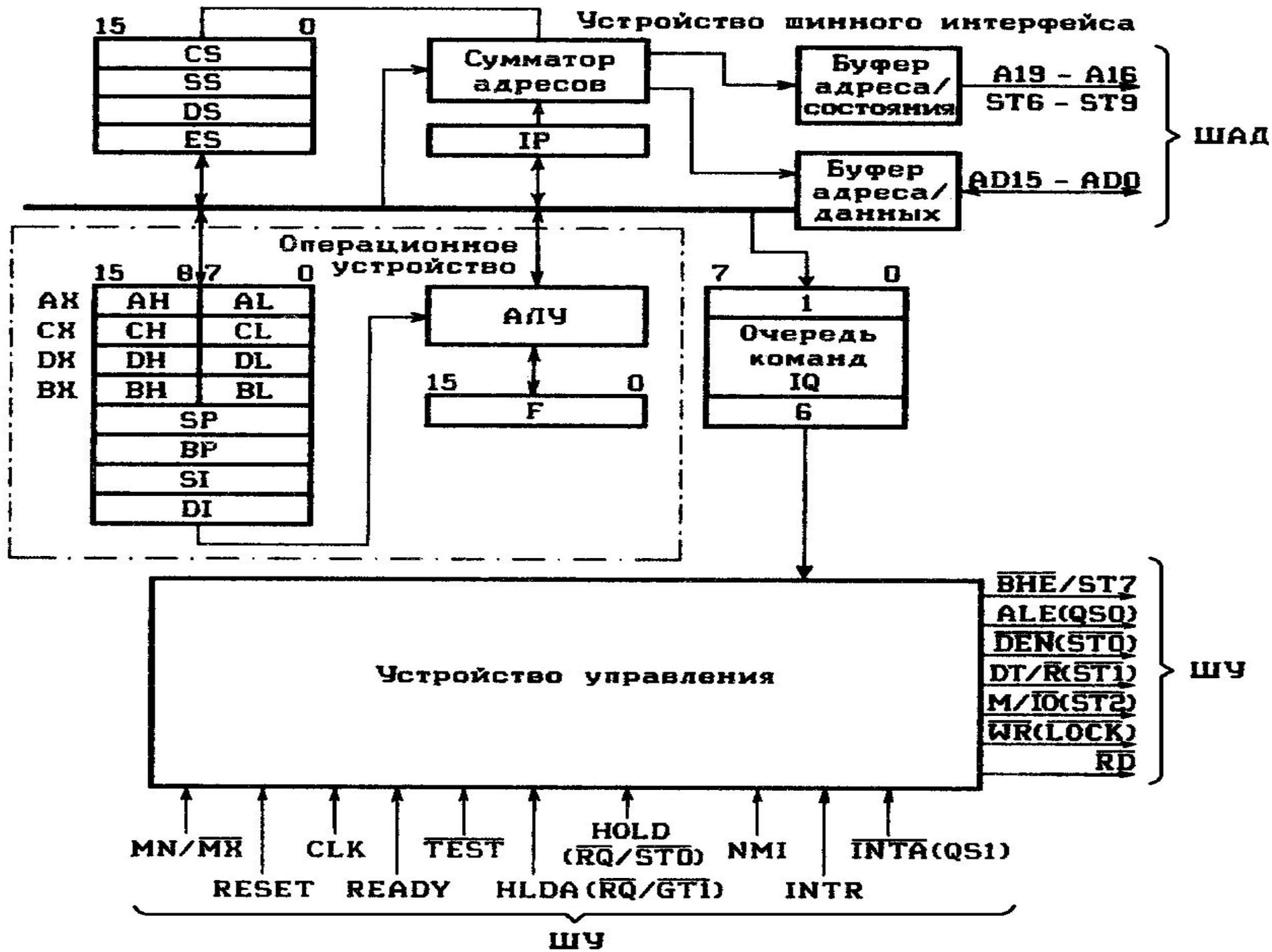


# БИС контроллера прерываний I8259

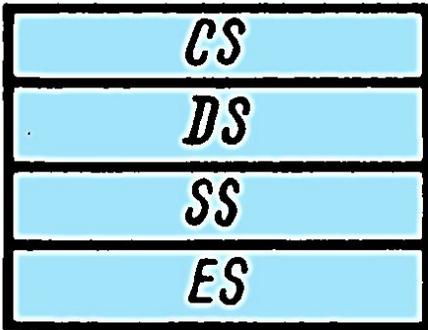


| Обозначение вывода | Номер контакта                 | Назначение вывода   |
|--------------------|--------------------------------|---|
| $D(7-0)$           | 4; 5; 6; 7; 8; 9; 10; 11       | Вход/выход данных   |
| $RD$               | 3                              | Вход строба чтения  |
| $WR$               | 2                              | » » записи  |
| $A0$               | 27                             | Вход нулевого разряда адреса, используемый при загрузке команд и считывании состояния ПКП   |
| $CS$               | 1                              | Вход выбора микросхемы  |
| $CAS0-CAS2$        | 12; 13; 15                     | Входы/выходы каскадирования   |
| $SP$               | 16                             | Признак подчинения: напряжение $H$ -уровня указывает, что ПКП является старшим (ведущим) контроллером; напряжение $L$ -уровня определяет ПКП подчиненным (ведомым) контроллером |
| $INTA$             | 26                             | Подтверждение прерывания – входное напряжение $H$ -уровня указывает о выдаче ПКП команды $CALL$ на шину данных  |
| $INT$              | 17                             | Прерывание – напряжение $H$ -уровня указывает о запросе на обслуживание прерывания  |
| $IR0-IR7$          | 18; 19; 20; 21; 22; 23; 24; 25 | Входы запросов прерываний (положительный фронт)   |
| $U_{CC}$           | 28                             | Напряжение питания (+5 В)   |
| $GND$              | 14                             | » » (0 В)   |

# Intel 8086



# Сегментные и индексные регистры



*CS* – регистр сегмента программы, определяет начальный адрес сегмента памяти, в котором располагается программа. Выборка очередной команды осуществляется относительно содержимого *CS* с использованием значения указателя команд *IP*.

*DS* – регистр сегмента данных, определяет начальный адрес текущего сегмента данных. Вычисление физического адреса в этом сегменте зависит от способа адресации.

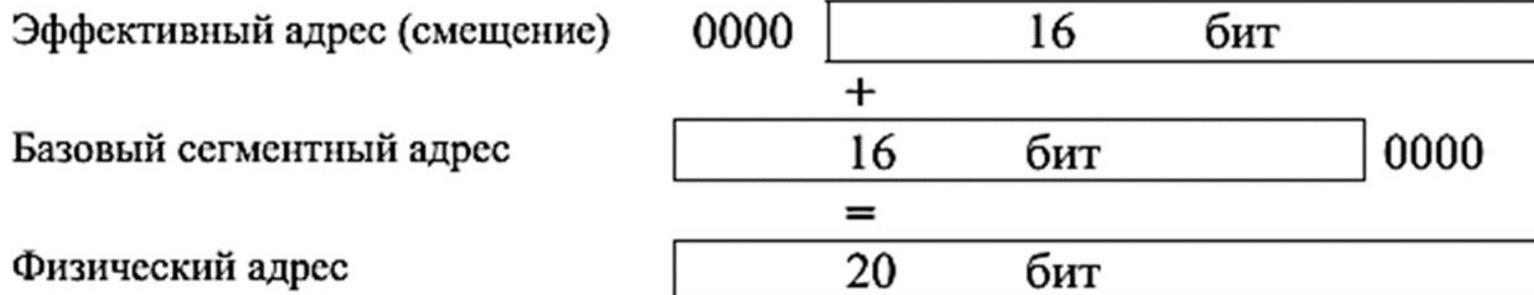
*ES* – регистр дополнительного сегмента данных

*SS* – регистр сегмента стека, определяет начало стекового сегмента и используется в командах обращения к стеку, при обработке прерываний и подпрограмм. (Стек – это сегмент памяти, работающий по принципу LIFO (Last in first out – вошедший последним выходит первым)).



*IP* – указатель команд, является смещением в сегменте относительно начала сегмента

## Формирование физического адреса



Указатель стека

Указатель базы

Индекс источника

Индекс приемника



**CF** (Carri Flag) – флаг переноса (заёма) старшего бита в арифметических операциях;

**PF** (Pariti Flag) – флаг паритета, устанавливается при чётном числе единиц в результате;

**AF** (Auxiliary Flag) – флаг дополнительного переноса (заёма) в тетраде для десятичной арифметики;

**ZF** (Zero Flag) – флаг нулевого результата.

**SF** (Sign Flag) – флаг знака. Если в триггере устанавливается единица, флаг отрицательный.

**TF** (Trap Flag) – флаг трассировки (пошагового режима). При установке этого флага после выполнения каждой команды вызывается внутреннее прерывание 1-го типа (INT 1).

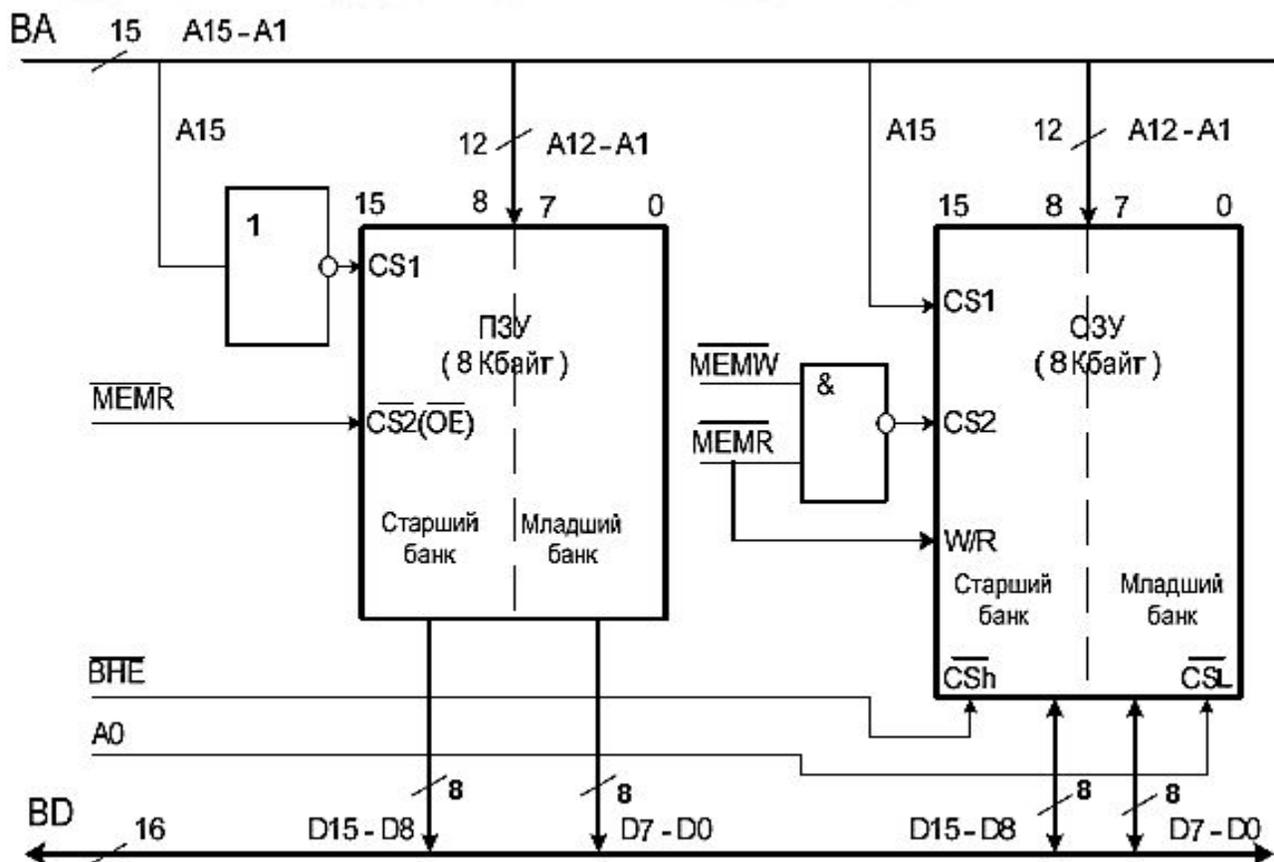
**IF** (Interrupt Flag) – флаг управления прерываниями. При единичном значении разрешается выполнение маскируемых аппаратных прерываний.

**DF** (Direction Flag) – флаг управления направлением в строковых операциях. При единичном значении индексные регистры, участвующие в строковых операциях, автоматически декрементируются на количество байт операнда, при нулевом – инкрементируются.

**OF** (Overflow Flag) – флаг переполнения. Устанавливается, если результат арифметической команды не помещается в операнде назначения.

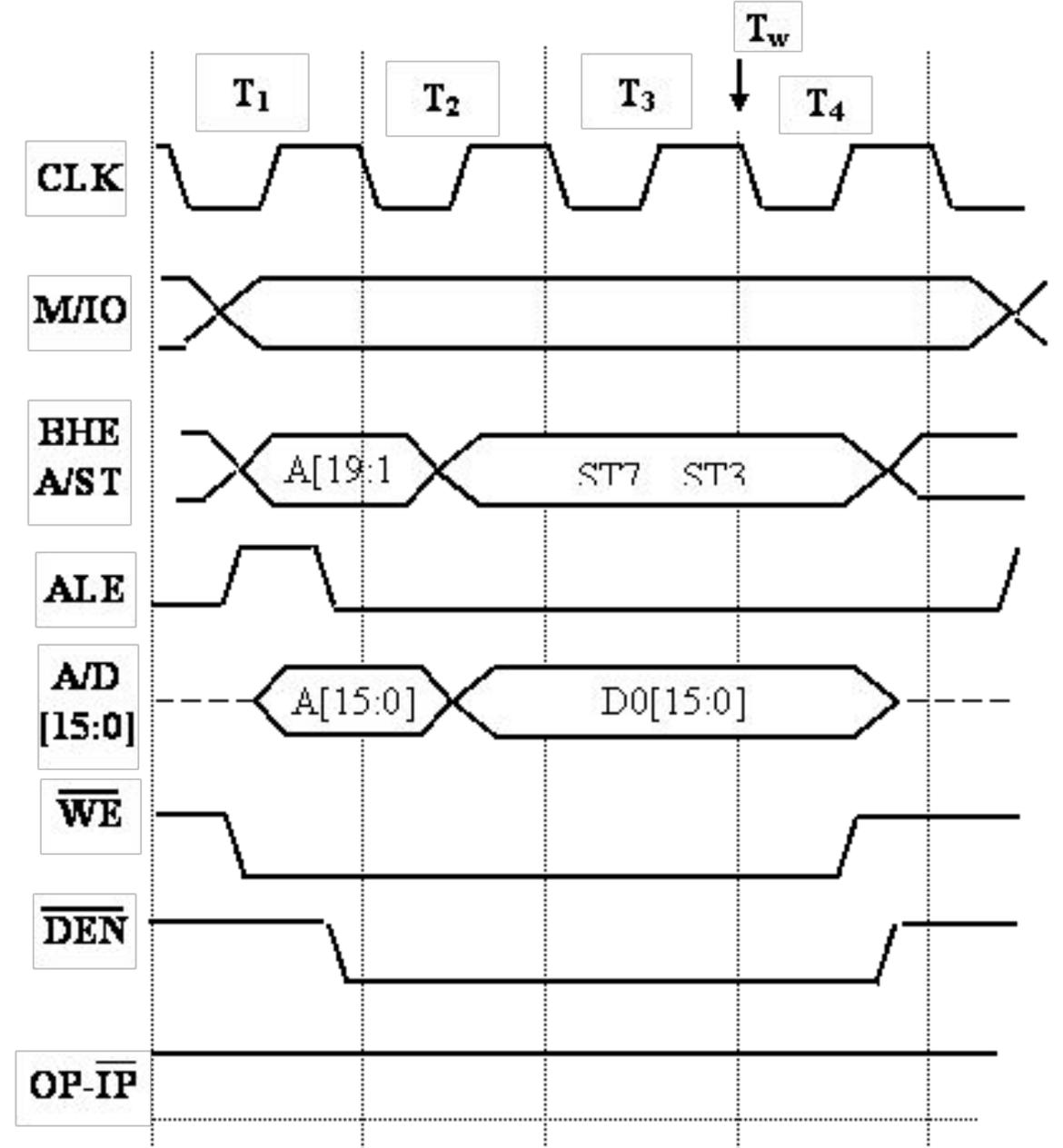
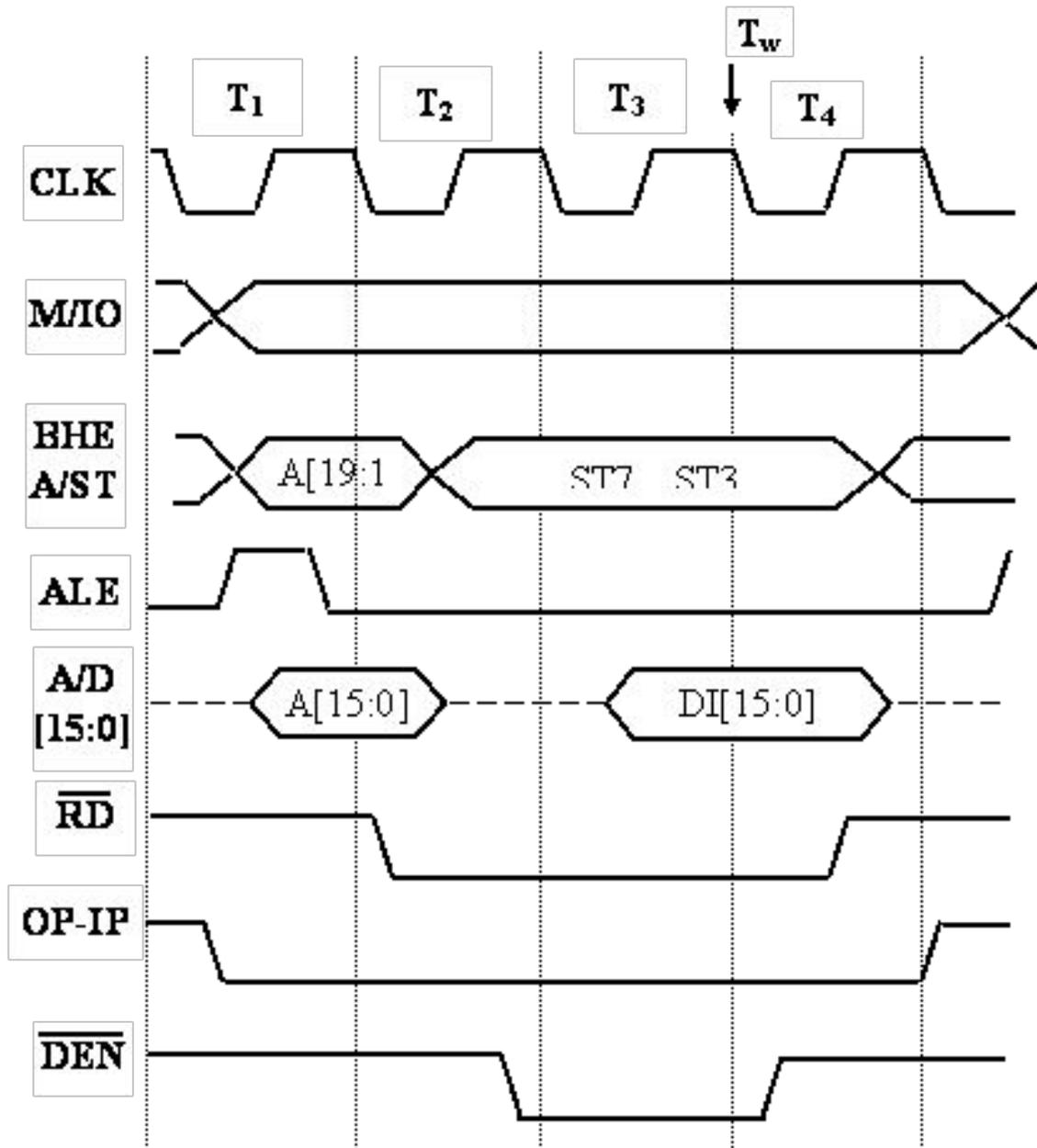
| Тип ссылки к памяти    | Сегмент по умолчанию | Альтернативный сегмент | Смещение |
|------------------------|----------------------|------------------------|----------|
| Выборка команды        | CS                   | Нет                    | IP       |
| Стековая операция      | SS                   | Нет                    | SP       |
| Переменная             | DS                   | CS, SS, ES             | EA       |
| Цепочка-источник       | DS                   | CS, SS, ES             | SI       |
| Цепочка-приёмник       | ES                   | Нет                    | DI       |
| BP как базовый регистр | SS                   | CS, SS, DS             | EA       |

## Схема подключения банков памяти



- A0 = 0, BHE = 0 – пересылается слово;
- A0 = 0, BHE = 1 – пересылается только младший байт;
- A0 = 1, BHE = 0 – пересылается только старший байт;
- A0 = 1, BHE = 1 – устройство не выбрано.

# ВРЕМЕННЫЕ ДИАГРАММЫ РАБОТЫ В МИНИМАЛЬНОМ РЕЖИМЕ



# ХРОНОЛОГИЯ МИКРОПРОЦЕССОРОВ INTEL

- |                                  |   |
|----------------------------------|---|
| 1971, 15 ноября: 4004            | 1991, 22 апреля: 80486SX                    |
| 1972, 4-й квартал: 4040          | 1992, 3 марта: 80486DX2                     |
| 1972, 1 апреля: 8008             | 1992, 9 ноября: 80486SL                     |
| 1974, 1 апреля: 8080             | 1993, 22 марта: Pentium                     |
| 1976, март: 8085                 | 1994, 7 марта: 80486DX4                     |
| 1978, 8 июня: 8086               | 1994, август: Intel386 EX                   |
| 1979, 1 июня: 8088               | 1995, 1 ноября: Pentium Pro                 |
| 1981, 1 января: iAPX 432         | 1997, 8 января: Pentium MMX                 |
| 1982: 80186                      | 1997, 7 мая: Pentium II                     |
| 1982, 1 февраля: 80286           | 1998, 15 апреля: Celeron (Pentium II-based) |
| 1985, 17 октября: 80386DX        | 1998, 29 июня: Pentium II Xeon              |
| 1988, 5 апреля: i960 aka 80960   | 1999, 26 февраля: Pentium III               |
| 1988, 16 июня: 80386SX           | 1999, 25 октября: Pentium III Xeon          |
| 1989, 16 января: 80376           | 2000, 23 августа: XScale                    |
| 1989, 27 февраля: i860 aka 80860 | 2000, 20 ноября: Pentium 4                  |
| 1989, 10 апреля: 80486DX         | 2001: Itanium                               |
| 1990, 15 октября: 80386SL        | 2001, 21 Мая: Pentium 4 Xeon                |

|   |   |
|---|---|
| 2002, июль: Itanium 2   | 2014, 3-й квартал: 14 нм, Core M, i3, i5, i7 — Broadwell                                  |
| 2003, март: Pentium M   | 2015, 3-й квартал: 14 нм, Core M, i3, i5, i7 — Skylake                                    |
| 2003, март: Celeron M   | 2017, 1-й квартал: 14 нм, Celeron, Pentium G, Core i3, i5, i7 — Kaby Lake (7-е поколение) |
| 2003, сентябрь: Pentium 4EE                                       | 2017, 3-й квартал: 14 нм, Core i9 — Skylake   |
| 2004, весна: EM64T  | 2017, 4-й квартал: 14 нм, Core i3, i5, i7 — Coffee Lake, i9 Skylake.                      |
| 2005, Q2 (?): Pentium D   | 2018, конец: 14 нм — Cooper Lake[3]   |
| 2006, осень: Conroe, <u>Merom</u>                                 | 2018, 4-й квартал: 14 нм — i3, i5, i7, i9 Coffee Lake Refresh (8-е поколение)             |
| 2007, зима: Core 2 Extreme QX6700                                 | 2018, конец: 10 нм — Cannon Lake  |
| 2007, зима: Core 2 Quad – Четырёхъядерный                         | 2019, 3-й квартал: 14 нм - Comet Lake   |
| 2008, зима: Core 2 обновление линейки                             | 2019, конец: 10 нм — Ice Lake (10-е поколение)  |
| 2008, весна: Centrino Atom  | 2020: 10 нм — Tiger Lake (11-е поколение)   |
| 2008, осень: Core i7  |   |
| 2009, осень: Core i5  |   |
| 2010, 4 января: Core i3   |   |
| 2011, весна: Celeron Sandy Bridge, Pentium Sandy Bridge           |   |
| 2011, 3 квартал: Core i3, i5, i7, i7 Extreme Edition Sandy Bridge |   |
| 2012, 1 квартал: Core i3, i5, i7 Ivy Bridge                       |   |
| 2013, 2 квартал: Core i5, i7 Haswell                              |   |

# КЛАССИФИКАЦИЯ МИКРОПРОЦЕССОРОВ

## Классификация Флинна:

1. ОКОД (**SISD** – single instruction stream / single data stream) одиночный поток команд, одиночный поток данных. – Чисто скалярный, последовательный МП.
2. ОКМД (**SIMD** – single instruction stream / multiple data stream) – одиночный поток команд, множественный поток данных. Это шаг к векторным процессорам (когда под 1 командой обрабатывается несколько данных). Появились в мультимедийных процессорах.
3. МКОД (**MISD** – multiple instruction stream / single data stream) – множественный поток команд и один поток данных. Конвейерный процессор.
4. МКМД (**MIMD** – multiple instruction stream / multiple data stream) – множественный поток команд и множественный поток данных. Параллельные конвейеры. Мультипроцессорная система. (начиная с 5 поколения).

Структура современного микропроцессора предполагает наличие порядка десяти обрабатывающих устройств, каждое из которых представляет собой конвейер. Загрузка параллельно функционирующих конвейеров обеспечивается:

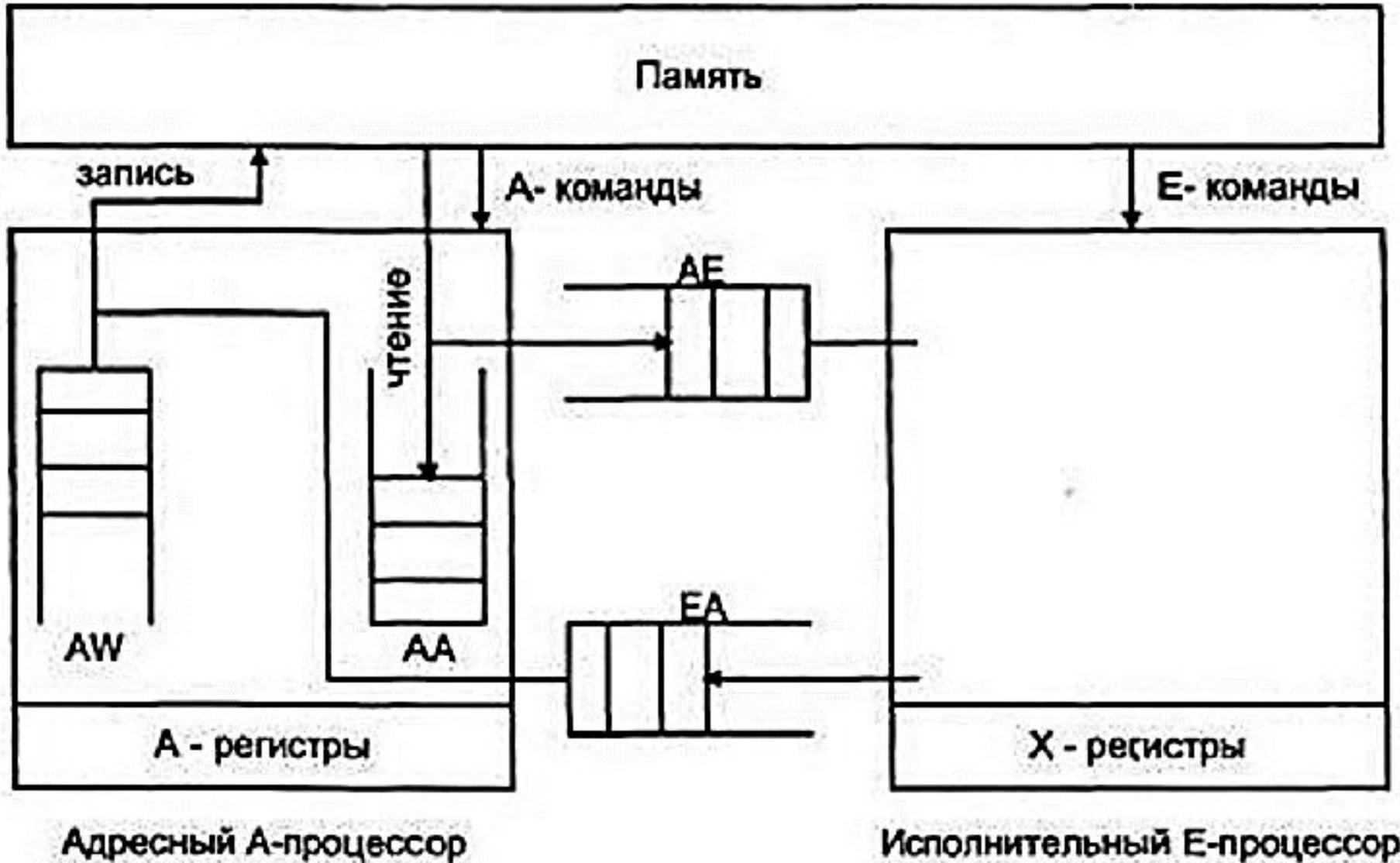
- либо аппаратурой процессора
- либо компилятором, на вход которого поступают программы на традиционном последовательном языке программирования
- либо совместно аппаратурой и компилятором

В компиляторах используется техника извлечения параллелизма из последовательных программ.

Аппаратура МП-в ориентирована на выделение более простых форм параллелизма, в том числе естественного.

В большинстве программ присутствует, так называемый, естественный параллелизм вычисления целочисленных адресных выражений и собственно обработки данных в формате с плавающей точкой.

# Микропроцессор с разнесенной архитектурой



Расщепление общей программы на программы для А- и Е- процессоров осуществляется на уровне компилятора или специальным блоком – расщепителем.

# RISC и CISC

## два направления развития микропроцессоров

### RISC (Reduced Instruction Set Computer)

- удалены сложные и редко используемые инструкции;
- все инструкции имеют одинаковую длину, что позволяет уменьшить сложность управления процессором и увеличить скорость обработки команд;
- отсутствуют инструкции, работающие с памятью напрямую, все данные загружаются только из памяти в регистр и наоборот;
- отсутствуют операции работы со стеком;
- применение конвейера параллельных вычислений;
- большинство операций производятся за один такт микропроцессора.

### CISC (Complete Instruction Set Computer)

- малое количество регистров общего назначения
- большое количество машинных команд
- сложное декодирование инструкций и, как следствие, расходование аппаратных ресурсов

Тактовая частота RISC-процессоров (при прочих равных условиях) выше тактовой частоты CISC-процессоров

# Обзор на примере семейства x86

1978 - Intel первый 16-разрядный процессор с маркировкой «8086», тактовая частота 5 МГц (позже появились процессоры 8 и 10 МГц), производительность 0,33 MIPS, технология 3 мкм, 29 тыс. транзисторов составляли кристалл, 1 Мб адресуемой оперативной памяти, конвейер имеет 6-байтную очередь инструкций.

1979 - 8088. Сущность ядра не изменилась, тактовая частота 10 МГц внешняя шина данных 8-битная. Это позволило, используя имеющуюся восьмиразрядную периферию, выпустить первый персональный компьютер — IBM PC.

1982 - i80286. 134000 транзисторов, технология 1,5 мкм, 16 Мбайт физической памяти, виртуальная память 1 Гбайт, частота от 12,5 МГц, защищенный режим. Появление псевдомногозадачности позволило создавать многозадачные ОС. Новые инструкции для работы с защищенным режимом и инструкции, использующие PIO (Programmable Input/Output). Deskriptory сегментов и deskriptорные таблицы. Уровни привилегий

1985 - 80386, первый полностью 32-битным процессор. 275000 транзисторов, частота от 16 до 32 МГц. Максимальный размер 4 гигабайта.

Виртуальная память 64 терабайт. Очередь команд 16 байт. Три режима работы.

1990 - Intel386S со средствами управления энергопотреблением.

1989 – 80486, 1,25 млн. транзисторов, 25 - 50 МГц, кэш первого уровня - 8 Кб, второго уровня - 512 Кб; средства, необходимые для построения мультипроцессорных систем; буферы отложенной записи; инструкции выполняются за 1 такт.

Арифметический сопроцессор на одном кристалле с центральным процессором.

1993 – Pentium, 3.1 млн. транзисторов, технология 0.8 мкм, частота 60-66 МГц; кэш первого уровня: 16 Кб; кэш второго уровня на материнской плате (до 1 Мб), шина данных 64 разрядная (60-66 МГц); адресная шина 32-разрядная. Socket 4.

Суперскалярность.

1995 - Pentium Pro, кэш-памяти второго уровня, работающая на полной частоте ядра; 5,5 млн. транзисторов в ядре, 15,5-31 млн. транзисторов на кэш-память, технология от 0,5 до 0,35 мкм, тактовая частота 150 - 200 МГц, кэш первого уровня 16 Кб, кэш второго уровня до 2 Мб, адресная шина 32-разрядная, шина данных 64.

1997 - Pentium MMX., встроенные средства поддержки мультимедиа, ускоренная обработка изображения и аудиоданных; новые инструкции; тактовая частота 233 МГц.

май 1997 - Pentium 2, ядро Pentium Pro с увеличенной тактовой частотой; модульная конструкция с разъемом типа Slot 1 и картриджем SECC (Single Edge Contact Cartridge); тактовая частота 233 - 450 МГц, технология 0,25 микрон, 7,5 миллионов транзисторов в ядре, кэш-память 2 уровня 512 Кб, первый уровень 32 Кб. В кристалл встраивается термодатчик, что позволяет мерить температуру прямо на ядре.

# Pentium 3

1999 - ядро **Katmai** - введено расширение SSE (Streaming SIMD Extensions), инструкции одновременно над группой операндов с плавающей точкой. 128-разрядные регистры XMM для инструкций класса SSE. 0,18 мкм технология, 28 млн. транзисторов в ядре, 256 Кб вторичного кэш, работающего на частоте ядра, тактовая частота 1 ГГц.

конец 1999 – ядро **Coppermine**, интегрированная в ядро кэш второго уровня 256 Кб (Advanced Transfer Cache), 0,18 мкм технология, 28 млн. транзисторов; расширение SSE (Streaming SIMD Extensions) - ориентировано на инструкции, выполняемые одновременно над группой операндов с плавающей точкой и 128-разрядные регистры XMM для SSE.

2001 – ядро **Tualatin**, новая 0.13 мкм технология.

# Pentium 4

2000 - Pentium 4 (**Willamette**), 0,18 мкм технология, 42 млн. транзисторов на кристалле, частоты от 1.4 до 2.0 ГГц. новый набор инструкций - SSE2; новые принципы микроархитектуры ядра – **NetBurst** ориентированные на высокие частоты Основная направленность - интернет и мультимедийные приложения. Суперскалярность.

Конвейерное исполнение и динамическое изменение последовательности команд. Усовершенствованное предсказание направления ветвлений обеспечивает ~ 90-процентную вероятность правильности предсказания.

Кэш третьего уровня, предусмотренный для серверных вариантов процессора

2001— ядро **Northwood**, 0.13 мкм технология, тактовая частота –2.6 ГГц, кэш второго уровня 512 Кб, 55 млн транзисторов. Поддержка наборов инструкций MMX, SSE SSE2

2002 - процессор с технологией Hyper-Threading, частота 3.06 ГГц.

Псевдомногопроцессорность: один физический процессор видится системой как два.

2003 -семейство **Prescott**, 0.09 мкм (90 нм) технология, площадь кристалла 112 мм<sup>2</sup>, 125 млн. транзисторов, тактовые частоты 3,4; 3,2; 3,0 ГГц. Кэш 2-го уровня 1 Мбайт, 13 новых инструкций SSE3. Усовершенствованная микроархитектура NetBurst, поддержка технологии **Hyper-Threading**.

2011 - **Extreme Edition**, технология Hyper-Threading, частота ядра 3.2 ГГц. Интегрированная в кристалл кэш-память третьего уровня L3 объемом 2 Мб, дополняющая стандартный кэш L2 512 кбайт и работающая также на частоте ядра.



**БФА** (блок формирования адреса) – формирует адреса выбираемых из памяти операндов, организуя связь с кэш-памятью данных 1-го уровня (L1). При обращении к памяти параллельно выставляет адрес для загрузки операнда в заданный регистр

**БРЗ** (блок регистров замещения) и адрес для пересылки результата из БРЗ в память.

Таким образом реализуется процедура предварительного чтения данных для последующей их обработки в исполнительных блоках, т.е. спекулятивная выборка.

Таким же образом организуется параллельная работа блоков SSE, FPU, MMX

**SSE** (Streaming SIMD Extension) – 128-битная векторная арифметика

**FPU** (Floating point unit) – модуль операций с плавающей запятой (с плавающей точкой)

**MMX** (MultiMedia eXtension) – SIMD-технология

# Система команд 80x86

подразделяется на группы:

- команды передачи данных;
- команды арифметических операций над целыми числами;
  - логические команды;
  - команды сдвига;
  - команды обработки строк;
- команды передачи управления;
  - команды прерываний;
  - команды управления флагами;
- команды управления состоянием процессора;
  - команды плавающей арифметики;
- команды мультимедийных расширений (MMX – MultiMedia eXtension);
  - команды потокового расширения (SSE – Streaming SIMD Extension).

# **Основные способы адресаций 80x86**

## **Регистровая адресация –**

операнды могут находиться в любых регистрах общего назначения и сегментных.

## **Непосредственная адресация –**

операнды приводятся непосредственно в операторе программы.

## **Прямая адресация –**

адрес операнда, располагающегося в памяти, присутствует в команде.

## **Косвенная адресация –**

адрес операнда в памяти содержится в регистре, а в команде присутствует имя этого регистра

## **Адресация по базе –**

в регистре находится адрес начала структуры данных.

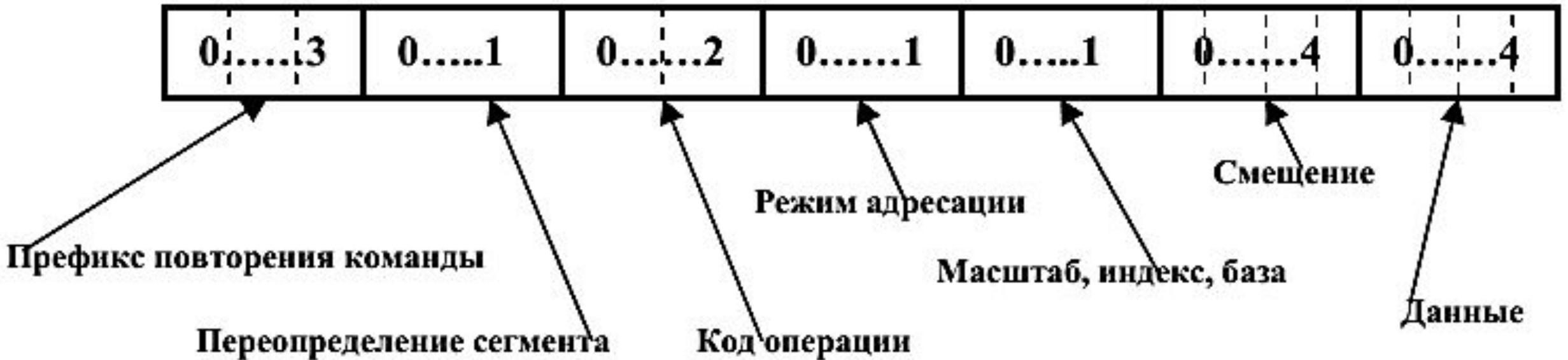
## **Косвенная адресация с масштабированием –**

этот способ адресации идентичен предыдущему, но используется, если массив состоит не из байт, а из слов, двойных слов, т. е. содержимое базового регистра умножается на два, а если из двойных слов – на четыре.

## **Адресация по базе с индексированием и масштабированием –**

самая полная возможная схема адресации, в которую входят как частные все случаи, рассмотренные ранее.

# Формат команд



**Поле префиксов** может содержать префикс повторения команды или префикс запрета доступа к шине на время выполнения команды, а также префиксы размеров адреса и операнда.

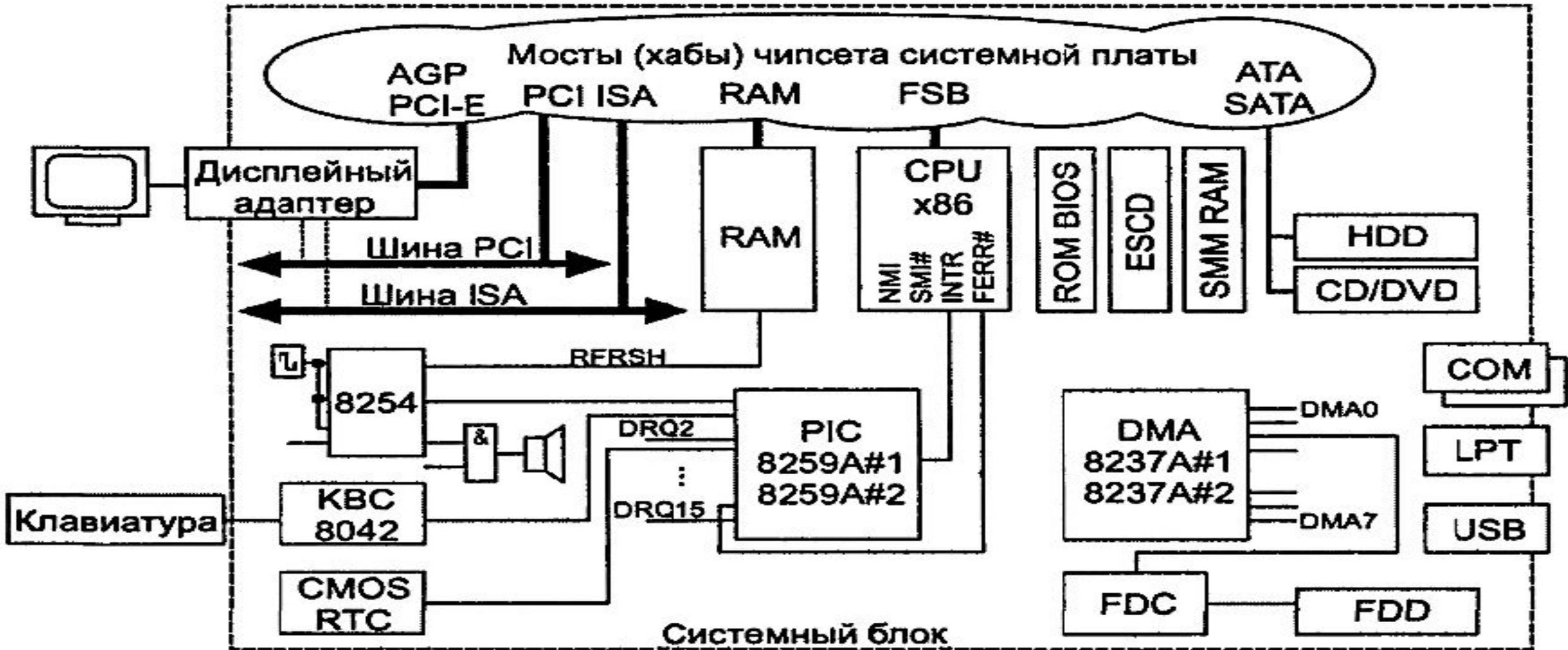
**Поле переопределения сегмента** используется для задания сегмента, вместо того, который определен по умолчанию.

**Поле кода операции** задает вид действия для данной команды.

**Поле режима адресации** определяет местонахождения операнда – в регистре или в памяти.

**Поле МИБ** (масштаб, индекс, база) является расширением поля режима адресации и используется при формировании исполнительного адреса для задания масштабного коэффициента, индексного и базового регистров.

# Архитектурные признаки IBM PC-совместимого компьютера



ESCD (Extended System Configuration Data) — спецификация стандарта конфигурирования компьютеров x86

SMRAM — оперативная память хранения специальной программы режима системного управления

(System Management Mode, SMM) для привилегированного исполнения

## **Ядро:**

один или несколько микропроцессоров, программно совместимых с x86;  
оперативная память; ПЗУ с BIOS; связующие их интерфейсы.

В наполнения ядра возможны варианты

## **вне Ядра:**

Набор обязательных средств  
ввода/вывода  
и средств поддержки периферии

### **Обязательные компоненты системной платы:**

- ✓ DRAM;
- ✓ системные средства ввода/вывода:
- ✓ контроллеры клавиатуры, прерываний, DMA:
- ✓ таймеры, CMOS RTC, средства управления динамиком;
- ✓ Интерфейсные схемы и разъемы шин расширения;
- ✓ кварцевый генератор синхронизации;
- ✓ схема формирования сброса системы по сигналу Power Good от блока питания или кнопки Reset;
- ✓ схема управления блоком питания;
- ✓ регуляторы-преобразователи напряжения (Voltage Regulation Module, VRM).
- ✓ средства мониторинга состояния системного блока:
- ✓ измерители скорости вращения вентиляторов и температуры процессора и т. д.;
- ✓ измерители питающих напряжений;
- ✓ сигнализаторы несанкционированного доступа и т. п.
- ✓ интерфейсы COM- и LPT-портов,
  - ✓ 4–6 портов USB,
- ✓ каналы ATA и/или 2–4 порта SATA.
  - ✓ контроллер FireWire,
- ✓ адаптер локальной сети (Ethernet)

# Архитектурные построения системных плат

## □ Шинно-мостовая —

наличие центральной магистральной шины и подключение к ней через мосты компонентов схемы; два основных моста — северный (Northbridge) и южный (Southbridge). северный мост обеспечивает взаимодействия центрального процессора, оперативной памяти и видеоподсистемы; южный мост для связи с внешними устройствами

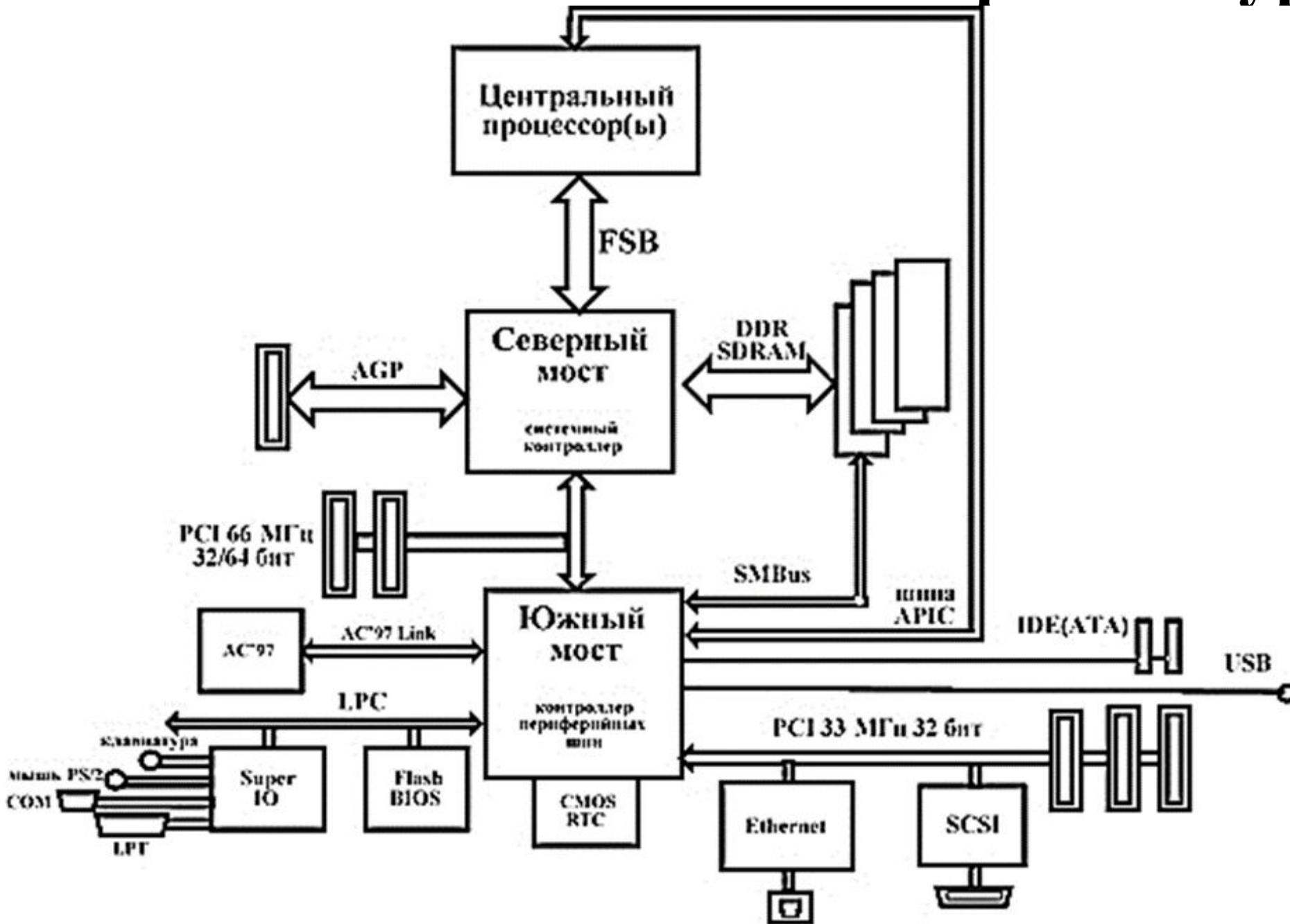
## □ Хабовая —

хабы это специализированные микросхемы, обеспечивающие передачу данных между интерфейсами; соединение основывается на паре основных хабов (северного и южного), связанных между собой высокопроизводительным каналом

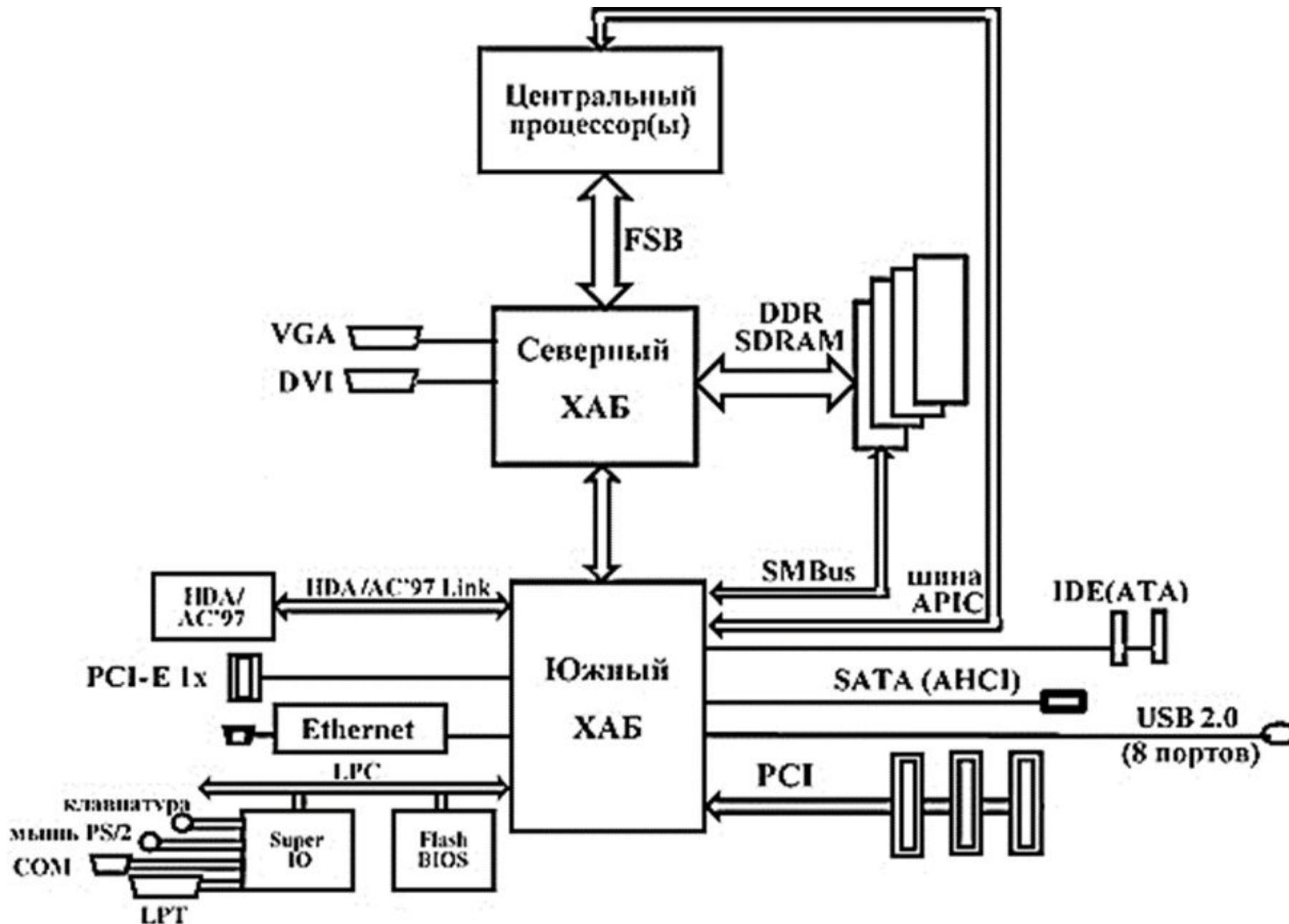
## □ Hyper Transport —

замена шинного соединения периферийных устройств двухточечными встречно направленными соединениями. более высокая тактовая частота интерфейсов и, соответственно, пропускная способность. обеспечивает все типы транзакций

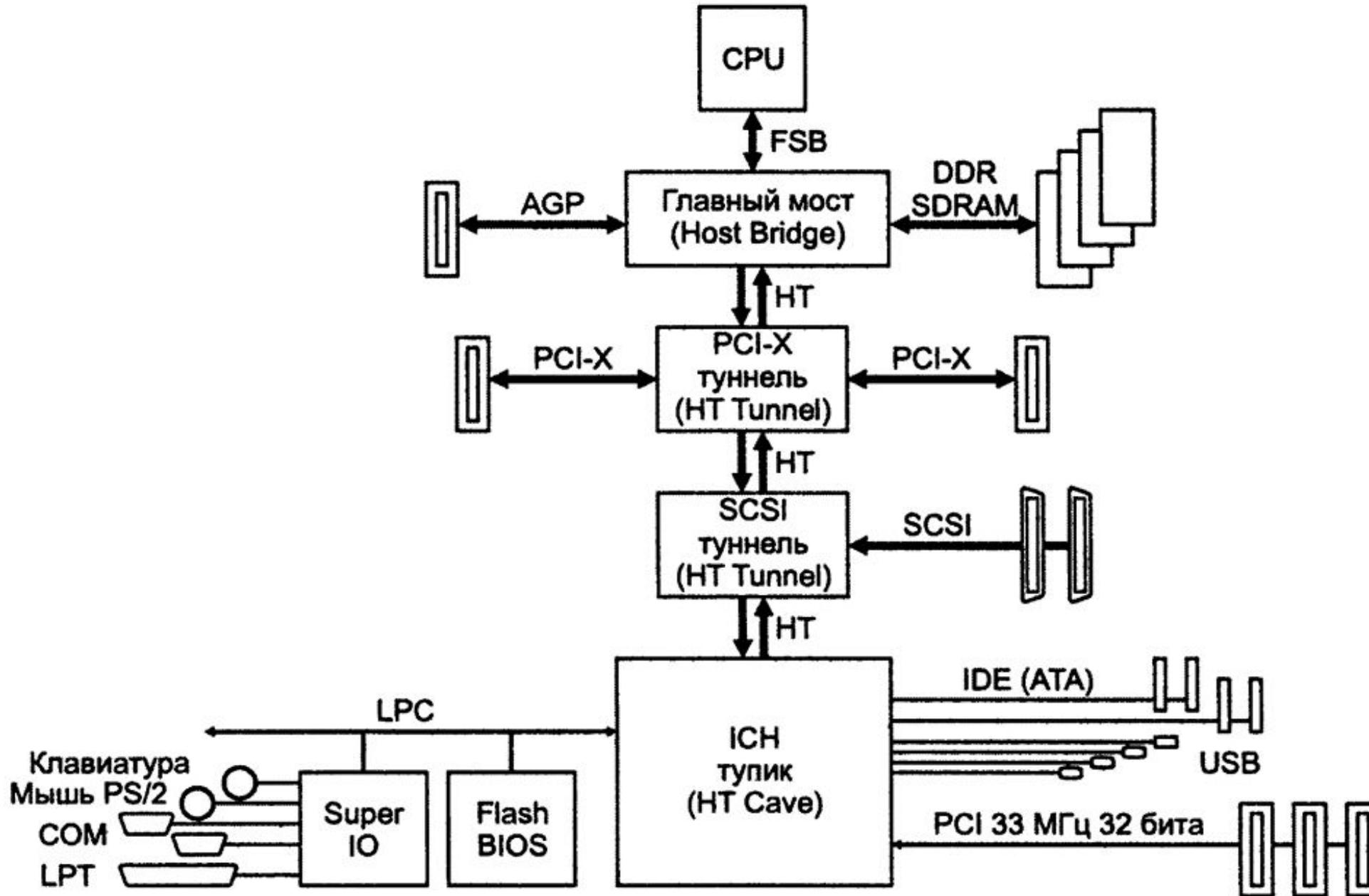
# Шинно-мостовая архитектура



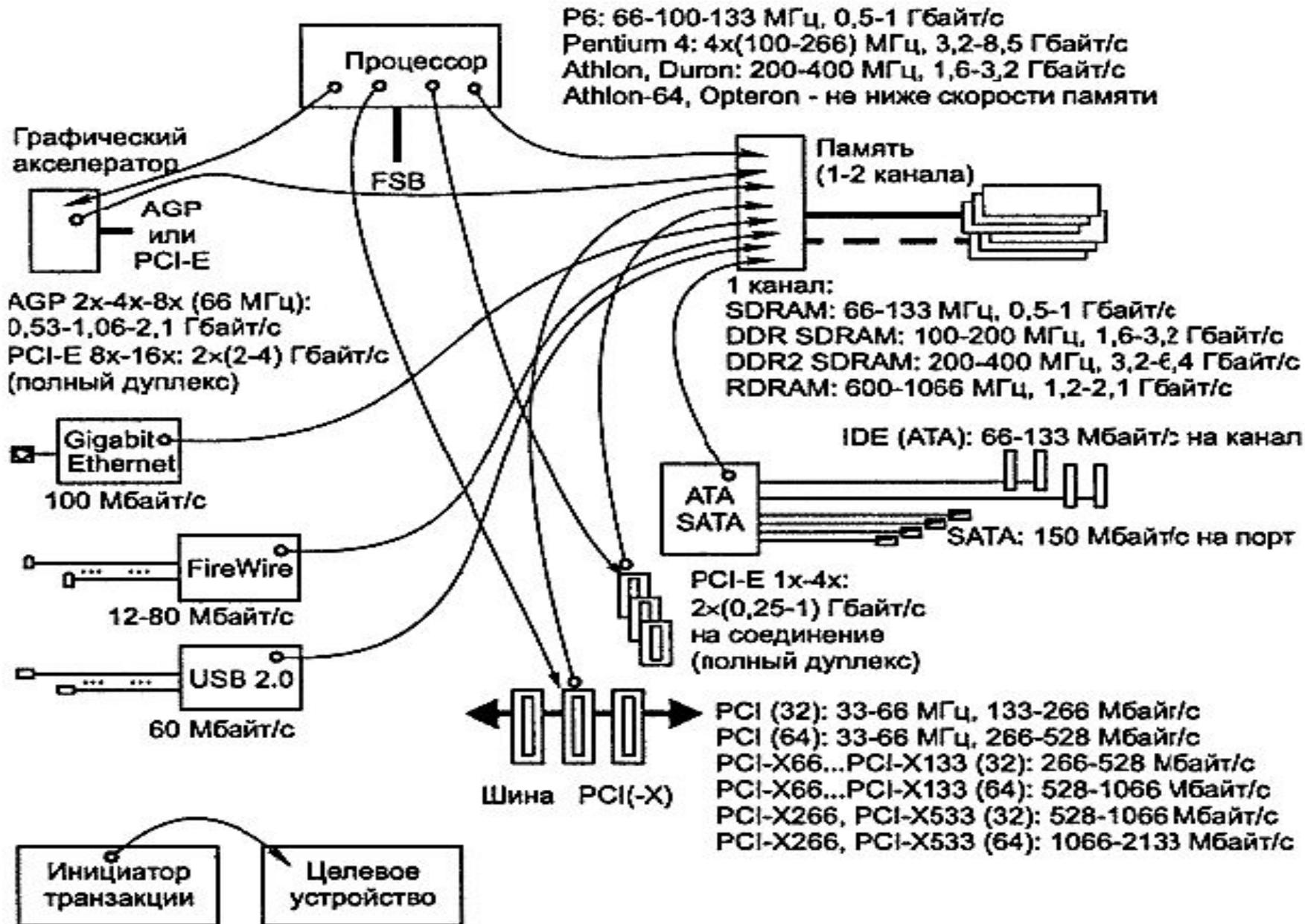
# Хабовая архитектура



# Архитектура Hyper Transport



# Компоненты и потоки данных на системной плате



# Типы и характеристики интерфейсов

## Интерфейс –

это аппаратное и программное обеспечение (элементы соединения и вспомогательные схемы управления, их физические, электрические и логические параметры), предназначенное для сопряжения систем или частей системы (программ или устройств)

### выполняемые функции:

- выдача и прием информации;
- управление передачей данных;
- согласование источника и приемника информации.

### технические характеристики интерфейсов:

- вместимость (максимально возможное количество абонентов, одновременно подключаемых к контроллеру интерфейса без расширителей);
- пропускная способность или скорость передачи (длительность выполнения операций установления и разъединения связи и степень совмещения процессов передачи данных);
- максимальная длина линии связи;
  - разрядность;
- топология соединения.

# Архитектура системных интерфейсов

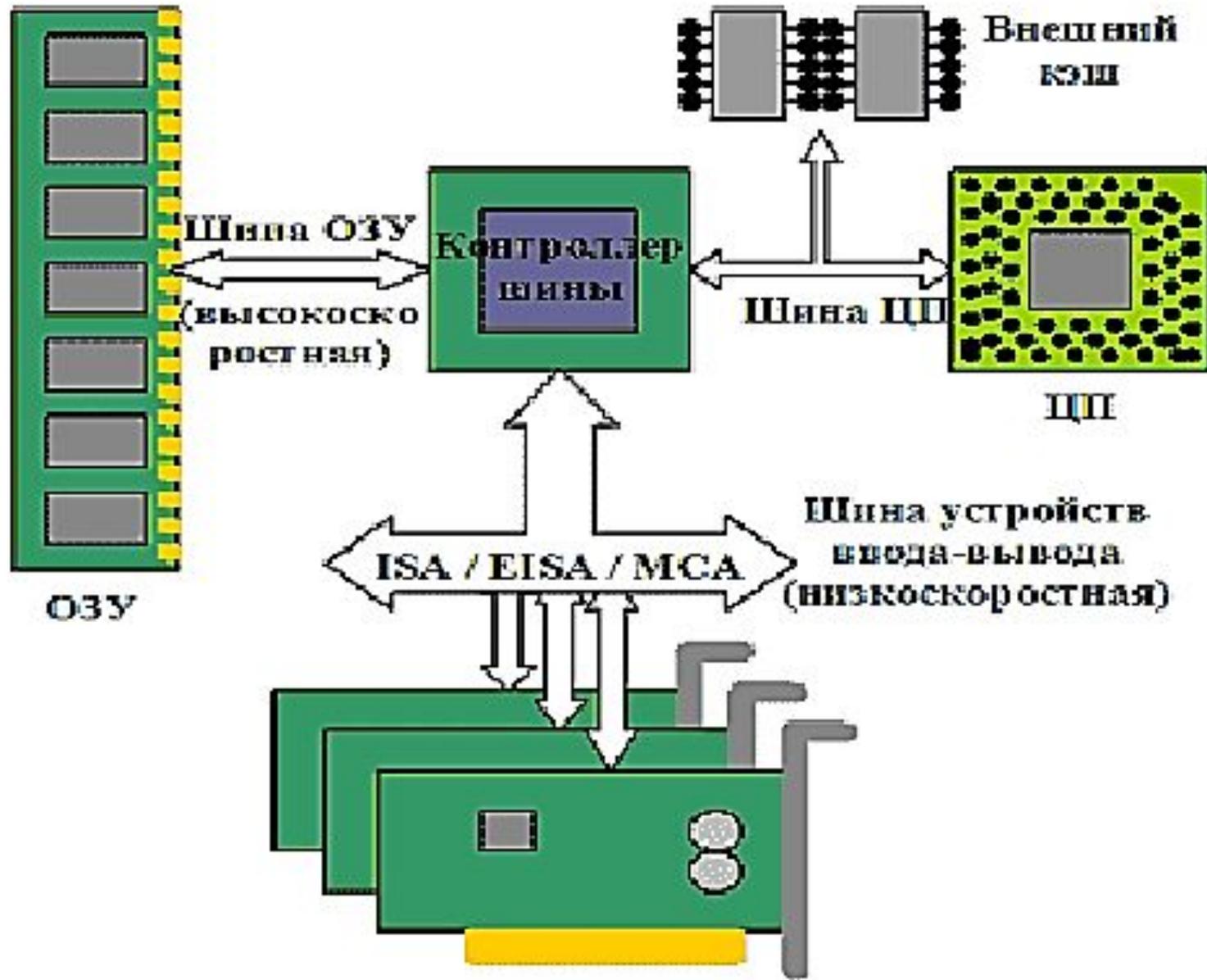
Системный интерфейс выполняется в виде стандартизированных системных шин. Возможно внедрение сетевого взаимодействия в архитектуру системных интерфейсов.

Два класса системных интерфейсов:

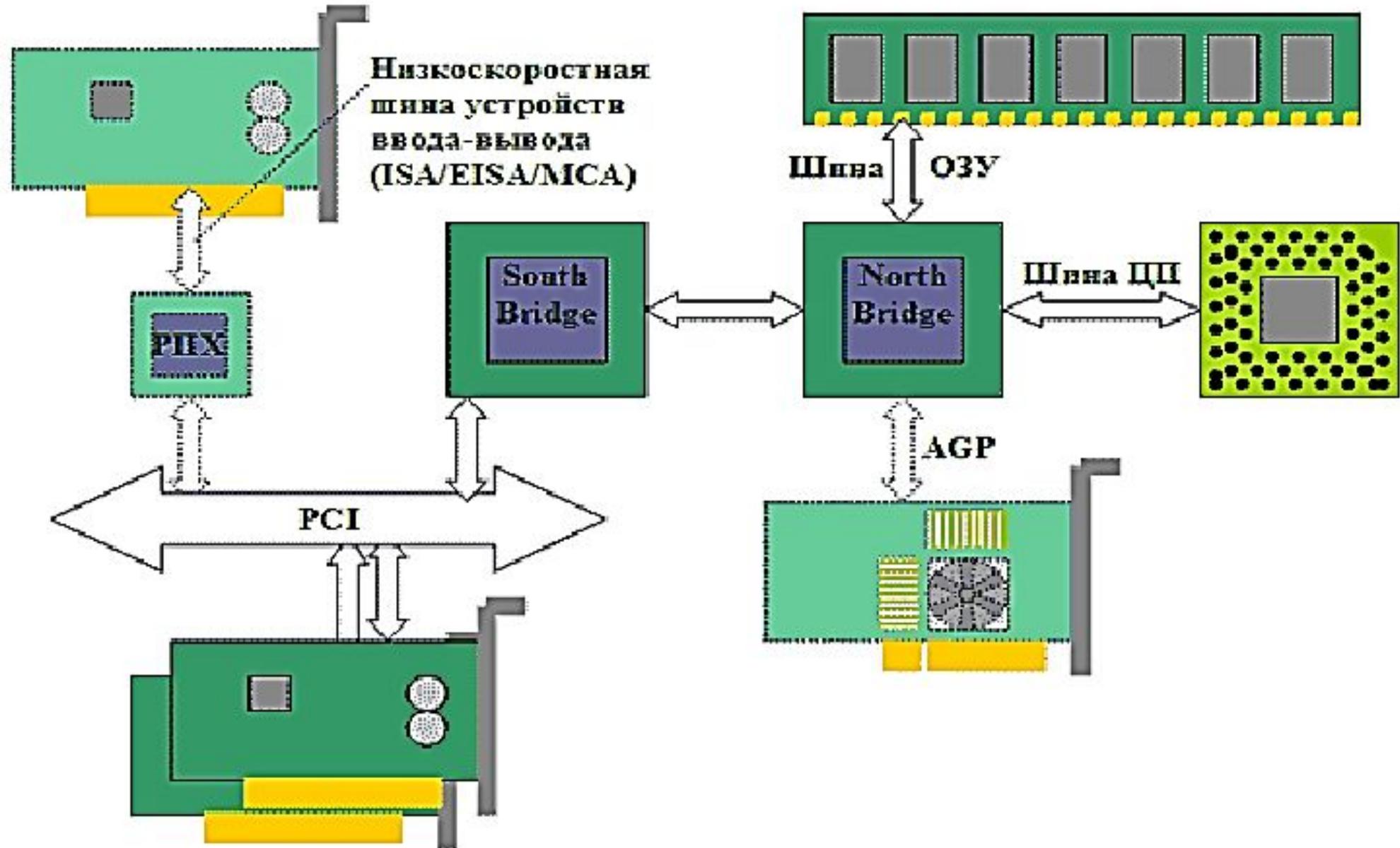
с общей шиной (сигналы адреса и данных мультиплексируются) - Unibus,  
с изолированной шиной (раздельные сигналы данных и адреса) - Multibus.

- ❖ ISA (Industry Standard Architecture) – архитектура промышленного стандарта; первый стандартный системный интерфейс в IBM PC-совместимых ПК для обеспечения питания и взаимодействия плат расширения с системной платой.
  - ❖ EISA (Extended ISA) – эволюционное развитие архитектуры ISA.
  - ❖ MCA (Micro Channel Architecture) – микроканальная архитектура; альтернативная системная архитектура не зависит от типа процессора, является асинхронной, предусмотрена автоматическая конфигурация системы.
- ❖ PCI (Peripheral Component Interconnect) – взаимодействие периферийных компонентов; поддерживает Plug&Play  
(PnP – технология, предназначенная для быстрого определения и конфигурирования устройств).
  - ❖ AGP (Accelerated Graphics Port) – высокоскоростной графический порт; выделенный интерфейс для подключения видеокарты
  - ❖ PCI Express – 3GIO (Third Generation Input/Output Interconnection), 3-е поколение шины ввода/вывода, межэлементный последовательный интерфейс.

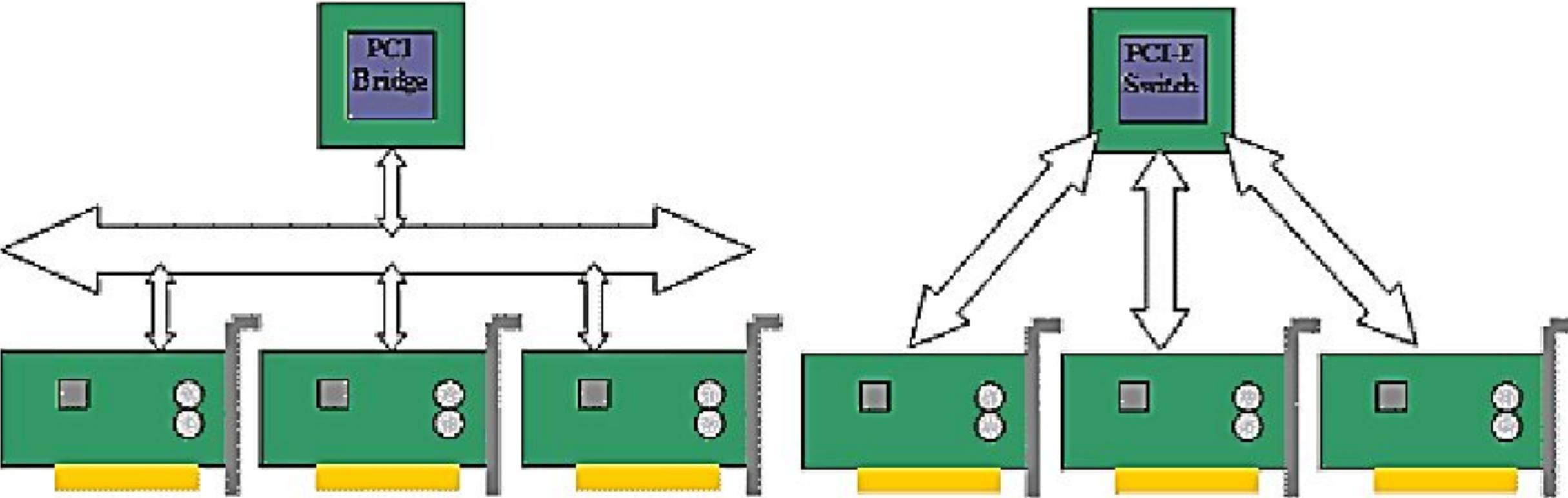
# Система с низкоскоростной шиной устройств ввода/вывода



# Система на основе PCI



# Сравнение топологий PCI и PCI Express



# Порт параллельного интерфейса

был введен в РС для подключения принтера –**LPT**-порт (Line PrinTer – построчный принтер); традиционный LPT-порт, он же стандартный SPP (Standard Parallel Port, SPP), ориентирован на вывод данных, через него программно реализуется протокол обмена Centronics (международный стандарт параллельного интерфейса для подключения периферийных устройств персонального компьютера);

Стандарт IEEE 1284, стандартизует модификации параллельного порта (SPP, EPP и ECP);

определяет 5 режимов обмена, метод согласования режима, физический и электрический интерфейсы.

- Режим совместимости (Compatibility Mode) – однонаправленный, вывод по протоколу Centronics. (для SPP)
- Полубайтный режим (Nibble Mode) – ввод байта в два цикла (по 4 бита) с использованием для приема линии состояния; для любых адаптеров, т. к. задействует только возможности стандартного порта.
- Байтный режим (Byte Mode) – ввод байта целиком с использованием для приема линии данных; работает только на портах, допускающих чтение выходных данных.
  - Режим EPP (EPP Mode) – двунаправленный обмен данными, EPP (Enhanced Parallel Port) – улучшенный параллельный порт; управляющие сигналы интерфейса генерируются аппаратно во время цикла обращения к порту; для внешней памяти и адаптеров локальных сетей.
- Режим ECP (ECP Mode) – двунаправленный обмен с возможностью аппаратного сжатия данных по методу RLE (Run Length Encoding) и использованием FIFO-буферов и DMA; ECP (Extended Capability Port) – порт с расширенными возможностями.

управляющие сигналы генерируются аппаратно. Для принтеров, сканеров и устройств блочного обмена

# Универсальный внешний последовательный интерфейс

**COM-** порт (Communications Port) – коммуникационный порт;

обеспечивает асинхронный обмен по стандарту RS-232C;

реализуется на микросхемах универсальных асинхронных приемопередатчиков (UART);

для связи с другими компьютерами, сетями и периферийными устройствами с последовательным интерфейсом (например, для подключения мыши)

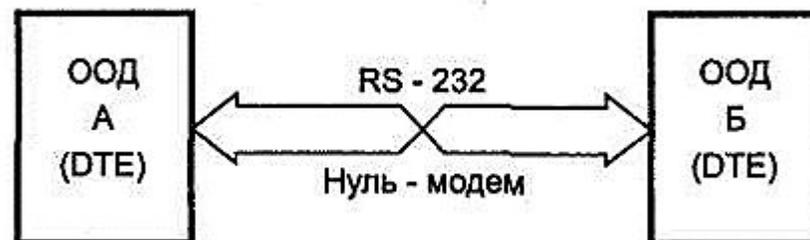
**Интерфейс RS-232C** (CCITT V.24/ V.28; X.20bis/X.21bis; ISOIS2110)

для подключения аппаратуры, передающей или принимающей данные

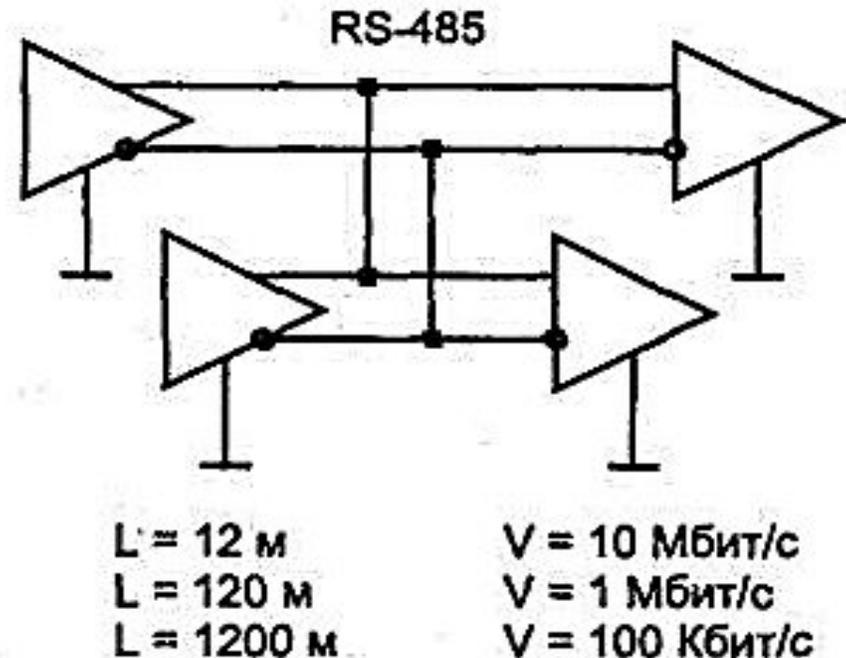
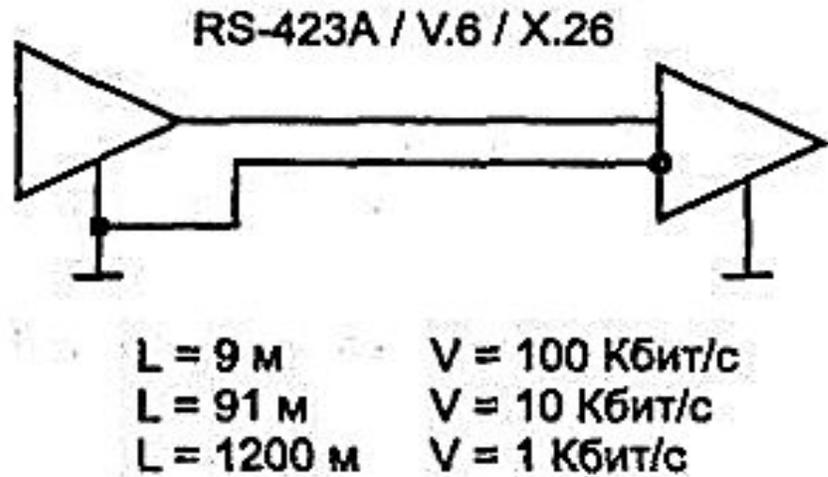
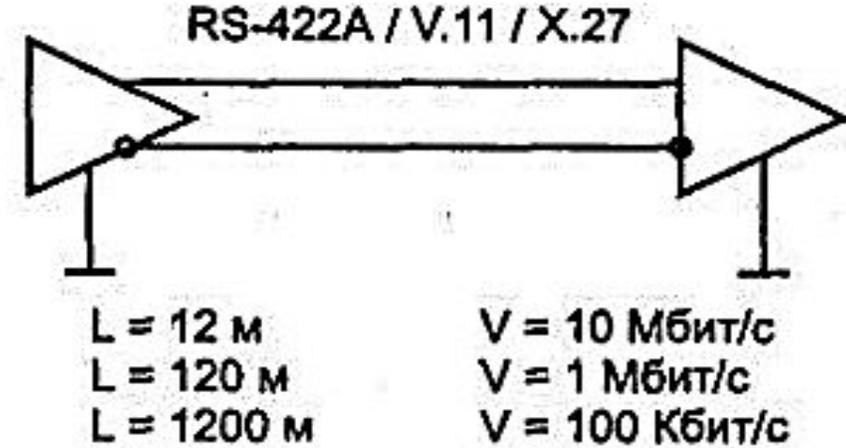
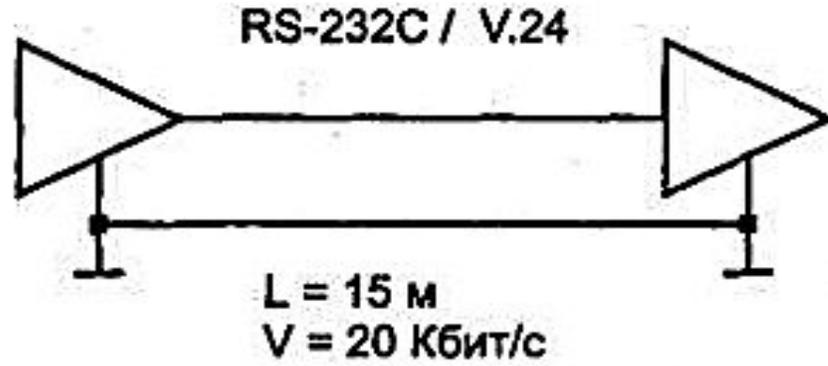


DTE (Data Terminal Equipment) – окончное оборудование данных (ООД), например, компьютер

DCE (Data Communication Equipment) – аппаратура передачи/приема данных (АКД), обычно модем



# Стандарты последовательных интерфейсов



# FireWire и USB

- последовательные соединения с возможностью коммутации при работающей системе большого количества периферийных устройств – 127 для USB и 63 для FireWire;
  - способны передавать изохронный трафик аудио- и видеоданных;
    - поддерживают технологию Plug&Play;
    - все протокольные функции выполняются посредством одной пары сигнальных проводов или двух – в FireWire;
    - шинный протокол строится на основе пересылки пакетов.
  - FireWire допускает большую свободу в топологии и пространственную протяженность, а также для FireWire значительно выше допустимая мощность;
  - FireWire и USB различаются пропускной способностью и управлением.

# USB

(Universal Serial Bus) – универсальная последовательная шина, промышленный стандарт расширения архитектуры PC

- кабель USB – экранированная витая пара сигнальных линий с импедансом 90 Ом и неэкранированная для подачи питания (+5 В), длиной до 5 м;
  - дифференциальный способ передачи сигналов;
- взаимодействие приложений с устройствами USB через программный интерфейс контроллера USB;

Для обмена информацией с хост-контроллером устройство, подключенное к шине USB, получает уникальный адрес и логически представляется как набор независимых конечных точек (EndPoint) со следующими параметрами:

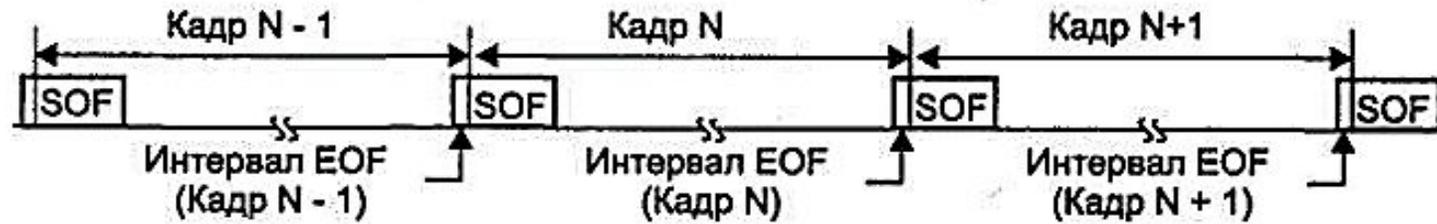
- требуемая частота доступа к шине и допустимые задержки обслуживания;
  - требуемая полоса пропускания канала;
  - требования к обработке ошибок;
- максимальные размеры передаваемых и принимаемых пакетов;
  - тип передачи;
- направление передачи (для передач массивов и изохронного обмена).

Конечная точка с нулевым номером используется для инициализации, общего управления и опроса состояния устройства; сконфигурирована при включении питания и подключении устройства к шине,

# Транзакции с устройствами USB

## Поток (stream)

однонаправленно доставляет данные от одного конца канала к другому, реализует передачу массивов изохронную, а также в случае прерывания;



SOF (Start Of Frame) –  
маркер синхронизации

EOF (End Of Frame) –  
завершение кадра

## Сообщение (message)

имеет формат, определенный спецификацией USB;

хост посылает запрос к конечной точке, после которого передается/принимается пакет сообщения, за которым следует пакет с информацией состояния конечной точки.



# Интерфейсы локальных сетей

организуются посредством сетевых адаптеров,  
или сетевых интерфейсных карт, Network Interface Card (NIC)

**Ethernet** – пакетная технология компьютерных сетей,

определяет проводные соединения и электрические сигналы на физическом уровне, формат пакетов и протоколы управления доступом к среде на канальном уровне модели OSI31; следующие поколения – 40 Gigabit Ethernet и 100 Gigabit Ethernet, Terabit Ethernet; вытеснил технологии Arcnet, FDDI и Token ring.

**OSI** (Open System Interconnection) –

набор протоколов стандарта сетевой архитектуры для семиуровневой функциональной модели сети

# Взаимодействие процессора с памятью

## Иерархическая организация памяти

помогает компенсировать разницу в быстродействии процессоров и скорости доступа к DRAM.

Когда процессору нужно выполнить команду:

1. анализируется состояние внутренних регистров;
2. если необходимые данные в регистрах не обнаружены, происходит обращение к кэш-памяти первого, второго и т. д. уровней...  
если обнаруживаются данные в одном из кэш, это называют «попаданием» (~ 90 % в оптимально организованных системах),  
неудачу (отсутствие данных) называют «промахом»;
3. если данных нет ни в одной кэш-памяти, процессор обращается к оперативной памяти;
4. при отсутствии нужных данных в оперативной памяти, они считываются с жесткого диска.

# Организация кэш-памяти (Cache memory)

кэш для инструкций и данных может быть отдельный и общий

□ в смешанной кэш-памяти  
высокая вероятность попаданий

□ в отдельной кэш-памяти выборка инструкций и данных производится одновременно, что исключает возможные конфликты, используется в системах с конвейеризацией команд при извлечении с опережением

**с прямым отображением (direct-mapped)** –

блок основной памяти занимает одно фиксированное место;

адрес обращения к кэшу из трёх составляющих:

старшие разряды (тег) определяют одну из закреплённых за данной строкой страниц оперативной памяти, средние разряды – это адрес строки кеша,

младшие – смещение, указывающее на номер ячейки в строке кеша;

**полностью ассоциативный (fully associative)** –

блок основной памяти может быть расположен в любом месте кэша,

для нахождения требуемой строки теговый адрес одновременно сравнивается с тегами всех строк кэша, сложная схемная реализация;

**наборно-ассоциативный (set-associative)** –

блок основной памяти может располагаться на ограниченном множестве мест;

младшие разряды адреса определяют номер ячейки памяти в строке (поле смещения),

средние задают номер набора (поле индекса),

старшие – тег, определяют выбранную страницу оперативной памяти, бит L для выбора строки.

# Логическое распределение пространства оперативной и постоянной физической памяти

□ • 00000h-9FFFFh (640 Кбайт) – стандартная, или базовая, память (conventional, или base, memory), доступна DOS и программам реального режима, 80000h-9FFFFh – расширенная базовая память (extended conventional memory);

□ A0000h-FFFFFFh (384 Кбайт) – верхняя память (Upper Memory Area, UMA), для системных нужд, блоки верхней памяти UMB (Upper Memory Blok):

A0000h – BFFFFh для видеопамяи,

C0000h – DFFFFh для адаптеров с собственными модули ROM BIOS и/или разделяемое с системной шиной,

E0000h – EFFFFh свободная область под системные модули BIOS,

F0000h – FFFFFh system BIOS,

FD000h – FDFFFFh энергонезависимая память ESCD (Extended System Configuration Data), при наличии PnP;

□ Память выше 100000h – дополнительная, или расширенная, память (extended memory), доступна только в защищенном (и в «большом реальном») режиме:

100000h-10FFEFh – высокая память (HMA) –

единственная область расширенной памяти, доступная в реальном режиме.

# Режим прямого доступа к памяти (Direct Memory Access, DMA)

**пассивный доступ (Slave DMA) –**

устройство при обращении к памяти использует управляющие возможности общего контроллера DMA

**активный доступ (Bus Master DMA) –**

для устройства, являющегося ведущим на шине и имеющего возможность организовать обмен с памятью

Принцип организации работы внешних устройств с памятью в режиме DMA изменился с введением шины PCI, где применяется технология захвата управления шиной внешним устройством (Bus Mastering – BM), реализующая прямой доступ к памяти, минуя процессор, для каждого устройства, которое может быть главным абонентом системной шины.

# Внешняя память

энергонезависимые устройства хранения данных, позволяющие сохранять информацию для последующего использования и реализующие различные физические принципы хранения информации – магнитный, оптический, электронный и их сочетания;

при доступе процессора посредством исполняемой программы к внешней памяти (в отличие от внутренней) оперируют блоками информации фиксированного и произвольного размера;

управление, в том числе избыточное кодирование и декодирование, исправление ошибок или/и организация повторных обращений к носителю и другие вспомогательные операции осуществляются посредством контроллера устройства хранения.

# Виртуальная память

использование иерархической организации системы памяти  
с непрерывным адресным пространством;

системное программное обеспечение связывает  
каждое указанное пользователем символьное имя с физической ячейкой памяти,  
т. е. отображает пространство имен на физическую память компьютера;

## **страничная организация –**

разбиение пространства логических адресов на блоки фиксированной длины – страницы. адрес обращения  
состоит из адреса (номера) страницы на диске  
и смещения в пределах страницы,  
отсутствует проблема внешней фрагментации;

## **сегментная организация –**

разбиение пространства логических адресов на крупные блоки информации переменной длины  
– сегменты (код программы, массив данных и т. д.)

## **сегментно-страничной организация –**

виртуальное пространство процесса делится на сегменты,  
а каждый сегмент – на страницы, которые нумеруются в пределах сегмента,  
преобразование виртуального адреса в физический – двухуровневое или трехкомпонентное,  
включающее номера сегмента и страницы, а также смещение в пределах страницы.

# ТИПЫ ПРЕРЫВАНИЙ

□ **внутренние прерывания** – реакция на возникшие внутренние проблемные события;

□ **аппаратные прерывания** – запрос посылает внешнее устройство,

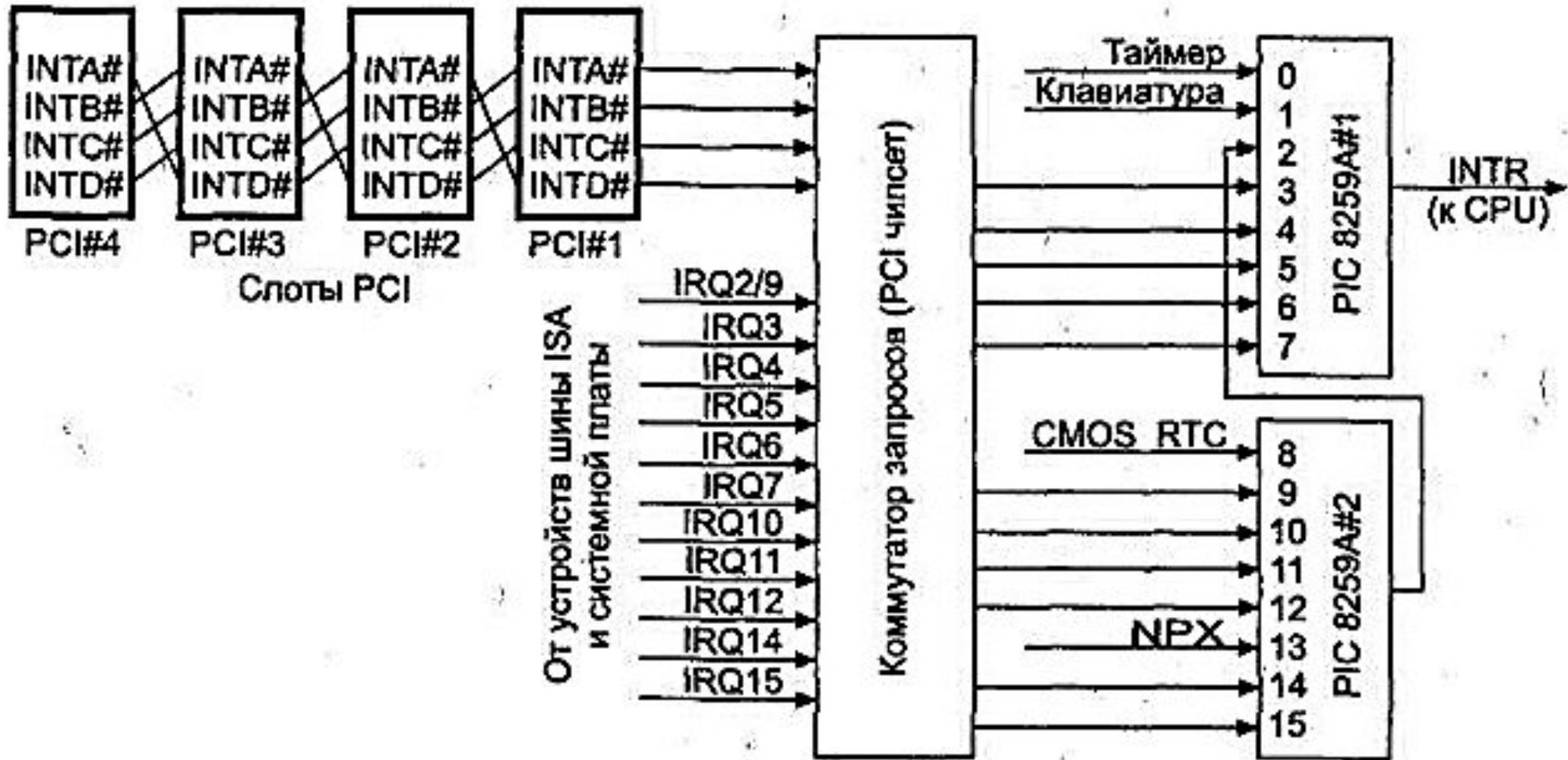
бывают **маскируемые** (запрещаемые) и **немаскируемые** (незапрещаемые),  
запрещение и разрешение маскируемых прерываний осуществляется инструкциями STI (Set Interrupt-Enable Flag) и CLI (Clear Interrupt Enable Flag),  
отражается состоянием флага IF (Interrupt Flag);

□ **программные прерывания** – обработка процедур по вектору, определяемому командой.

# Аппаратные прерывания в порядке убывания приоритета

| Имя    | Вектор | Контроллер/маска | Описание                                   |
|--------|--------|------------------|--|
| NMI    | 02h    |                  | Контроль канала, паритет                   |
| IRQ0   | 08h    | #1/1h            | Таймер (канал £8253/8254)                  |
| IRQ1   | 09h    | #1/2h            | Клавиатура                                 |
| IRQ2   | 0Ah    | #1/4h            | Резерв<br>(подключается каскад IRQ8-IRQ15) |
| IRQ8   | 70h    | #2/1 h           | CMOS RTC — часы реального                  |
| IRQ9   | 71h    | #2/2h            | Резерв                                     |
| IRQ10  | 72h    | #2/4h            | Резерв                                     |
| IRQ11  | 73h    | #2/8h            | Резерв                                     |
| IRQ12  | 74h    | #2/10h           | PS/2-Mouse (резерв)                        |
| IRQ 13 | 75h    | #2/20h           | Математический сопроцессор                 |
| IRQ14  | 76h    | #2/40h           | НОС  |
| IRQ15  | 77h    | #2/80h           | Резерв                                     |
| IRQ3   | 0B     | #1/4h            | COM2, COM4                                 |
| IRQ4   | 0C     | #1/10h           | COM1, COM3                                 |
| IRQ5   | 0D     | #1/20h           | XT — НОС, AT — LPT2,                       |
| IRQ6   | 0E     | #1/40h           | FDC  |
| IRQ7   | 0F     | #1/80h           | LPT1 — принтер                             |

# Коммутация запросов прерываний



для реализаций системы прерываний процессоры Pentium и выше имеют встроенный контроллер прерываний APIC (Advanced Programmable Interruption Controller); внутренние контроллеры процессоров связаны между собой по шине APIC, к которой подключена и часть чипсета, преобразующая запросы аппаратных прерываний в сигналы протокола APIC.

# Методы оценки производительности

производительность – скорость появления некоторого числа событий в секунду.

время выполнения заданной программы включает:

- ✓ временные затраты на работу процессора,
  - ✓ обращения к диску и к памяти,
  - ✓ ввод/вывод,
- ✓ действия операционной системы.

относительная производительность определяется:

- ✓ частотой синхронизации,
- ✓ средним числом тактов на команду,
- ✓ количеством выполняемых команд

Единицы измерения производительности:

**KOPS** (Kilo Operations Per Second) –тысячи операций в секунду,

**MIPS** (Million Instructions Per Second) – миллионы инструкций в секунду,

**MFLOPS** (Million Floating Point Operations Per Second) –

миллионы операций с плавающей точкой в секунду

# Что влияет на производительность

Применение конвейерной обработки,  
когда несколько последовательных команд находятся на разных стадиях выполнения,  
увеличивает производительность,  
но возникшее прерывание заставляет сбрасывать  
загруженную в конвейер очередь последовательных команд, готовых к выполнению  
и заполнять очередь другими командами из подпрограммы обработки прерывания.  
т.е. конвейерные прерывания ограничивают быстродействие.

Производительность зависит от  
быстродействия оперативной и дисковой памяти,  
кэширования,  
эффективности программного обеспечения.

## Для оценки и сравнения систем используются следующие критерии:

- Пиковая производительность – теоретический максимум быстродействия компьютера при идеальных условиях; это количество операций в единицу времени, выполняемых всеми имеющимися в микропроцессоре арифметико-логическими устройствами, параметр не зависит от типа выполняемой программы, т. к. его при выявлении обработке подвергается последовательность не связанных между собой и не конфликтующих при доступе в память команд.
- Максимальная пользовательская нагрузка – уровень, после которого производительность снижается недопустимо.
- Индекс производительности утилит определяет возможности системы по выполнению универсальных утилит.
- Пропускная способность системы.

# **Тестовая оценка производительности**

## **Тесты производителей –**

предназначены для оценки выпускаемых процессоров, ориентированы на сравнение ограниченного множества однотипных устройств.

## **Стандартные тесты независимых аналитиков –**

сравнение процессоров разных производителей.

## **Пользовательские тесты –**

ориентированы на оценку процессоров в ракурсе определенной задачи, комплексное их применение дает усредненную общую картину производительности

# **ОДНОКРИСТАЛЬНЫЕ МИКРОКОНТРОЛЛЕРЫ**

# Представители семейства

| МК   | ПЗУ (кБ)  | ОЗУ (Б) | stanby реж. | линий вв | КОЛ-ВО ВЫВОДОВ | примечания |
|------|-----------|---------|-------------|----------|----------------|------------|
| 8020 | 1         | 64      | н/д         | 13       | 20             |            |
| 8021 | 1         | 64      | н/д         | 21       | 28             |            |
| 8022 | 2         | 64      | н/д         | 21       | 28             | АЦП        |
| 8035 | -         | 64      | да          | 24       | 40             |            |
| 8039 | -         | 128     | да          | 24       | 40             |            |
| 8040 | -         | 256     | да          | 24       | 40             |            |
| 8048 | 1         | 64      | да          | 24       | 40             |            |
| 8049 | 2         | 128     | да          | 24       | 40             |            |
| 8050 | 4         | 256     | да          | 24       | 40             |            |
| 8748 | 1 (EPROM) | 64      | -           | 24       | 40             |            |
| 8749 | 2 (EPROM) | 128     | -           | 24       | 40             |            |
| 8648 | 1         | 64      | н/д         | 24       | 40             |            |

# Универсальный периферийный интерфейс (UPI)

| Обозначение | Встроенное ПЗУ (кВ) | Встроенное ОЗУ (В) | Примечания                |
|-------------|---------------------|--------------------|---------------------------|
| 8041        | 1                   | 64                 |                           |
| 8041АН      | 1                   | 128                |                           |
| 8741А       | 1                   | 64                 | Версия 8041 с EPROM       |
| 8741АН      | 1                   | 128                | Версия 8041АН с OTP EPROM |
| 8042АН      | 2                   | 256                |                           |
| 8742        | 2                   | 128                | Версия 8042АН с EPROM     |
| 8742АН      | 2                   | 256                | Версия 8042АН с OTP EPROM |

## Микросхемы расширения памяти и портов ввода вывода

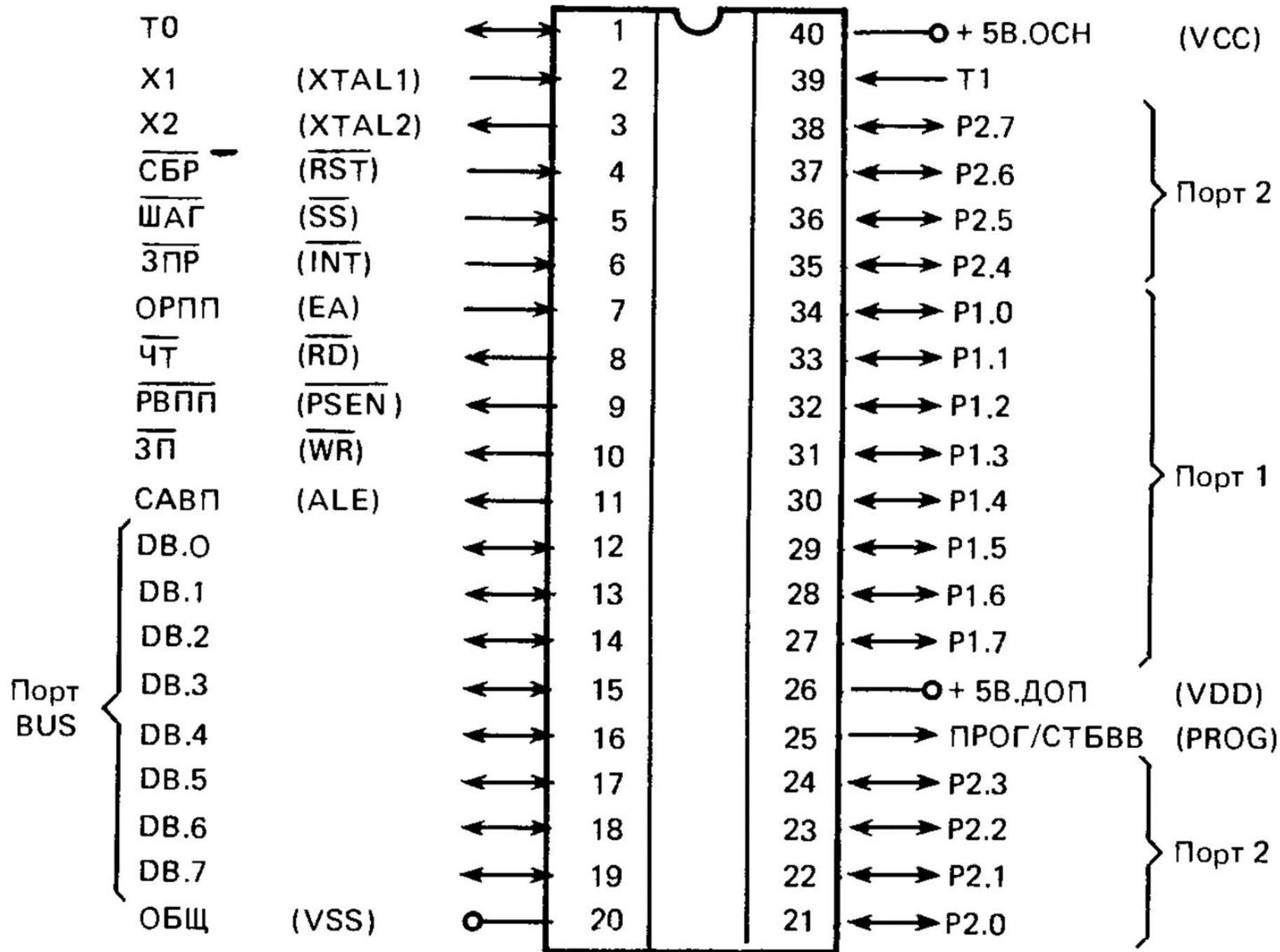
8355 – 2 Кб ROM, 16 линий ввода/вывода

8755А – 2 Кб EPROM, 16 линий ввода/вывода

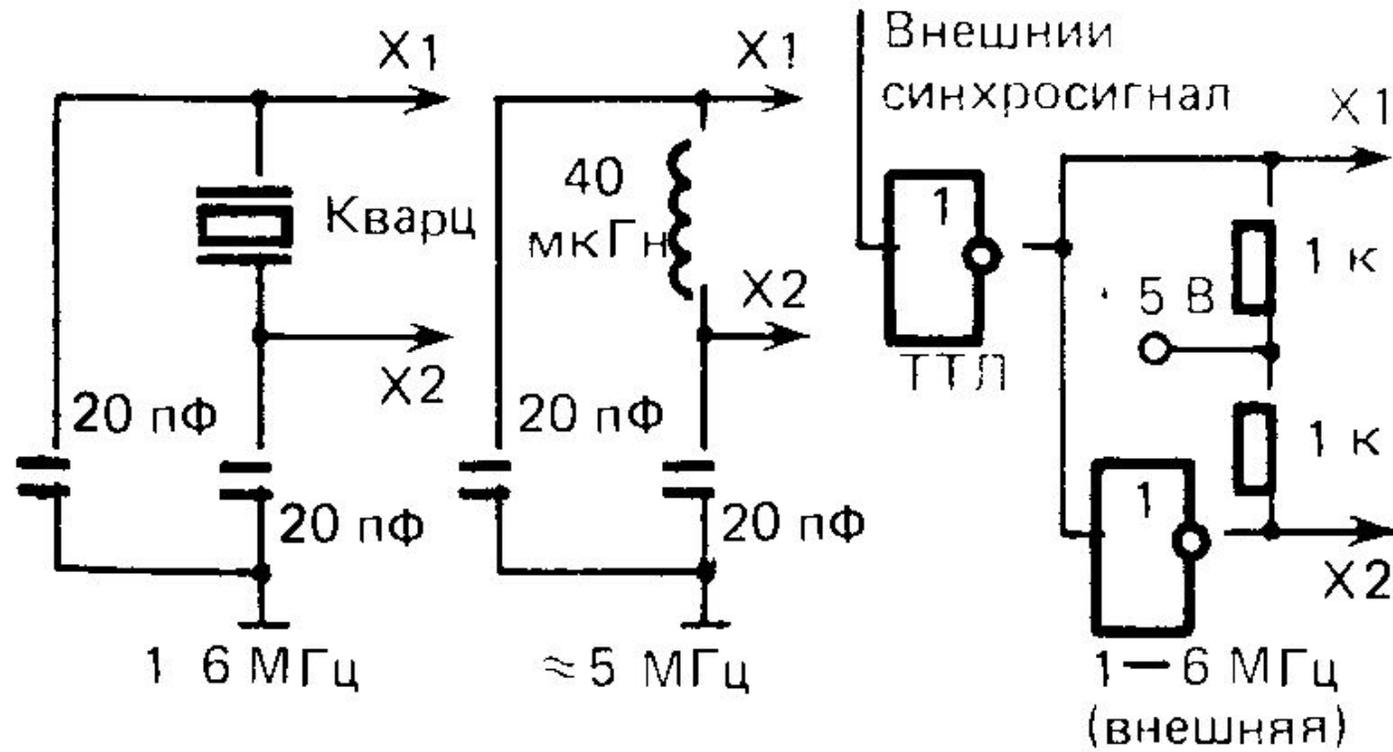
8155 – 256 байт RAM, 22 линий ввода/вывода, 1 таймер

8156 – 256 байт RAM, 22 линий ввода/вывода, 1 таймер

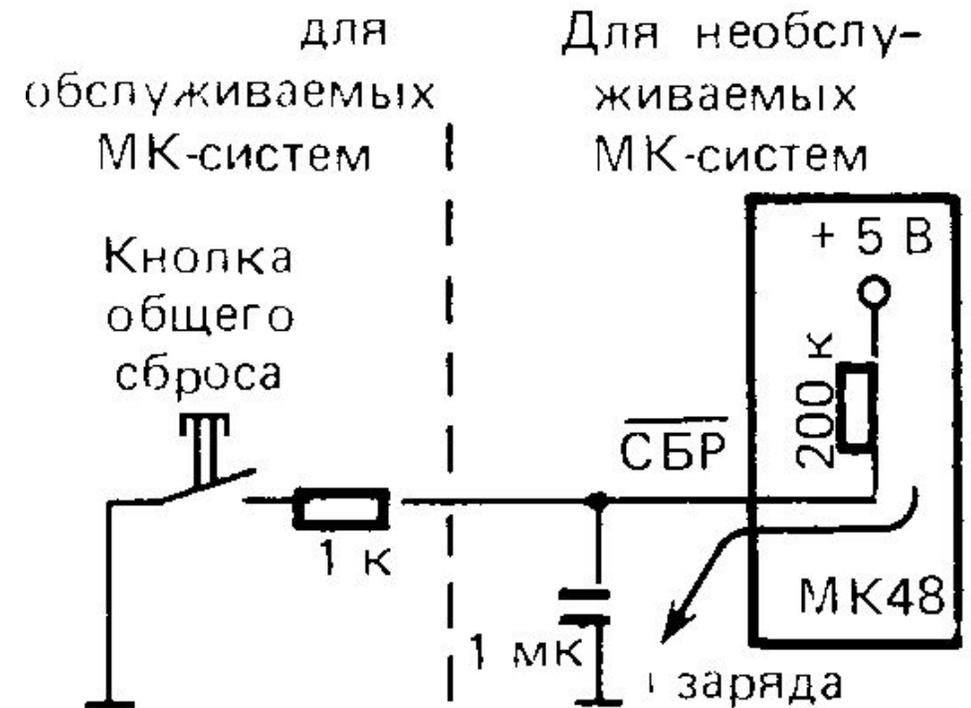
8243 – 16 линий ввода/вывода



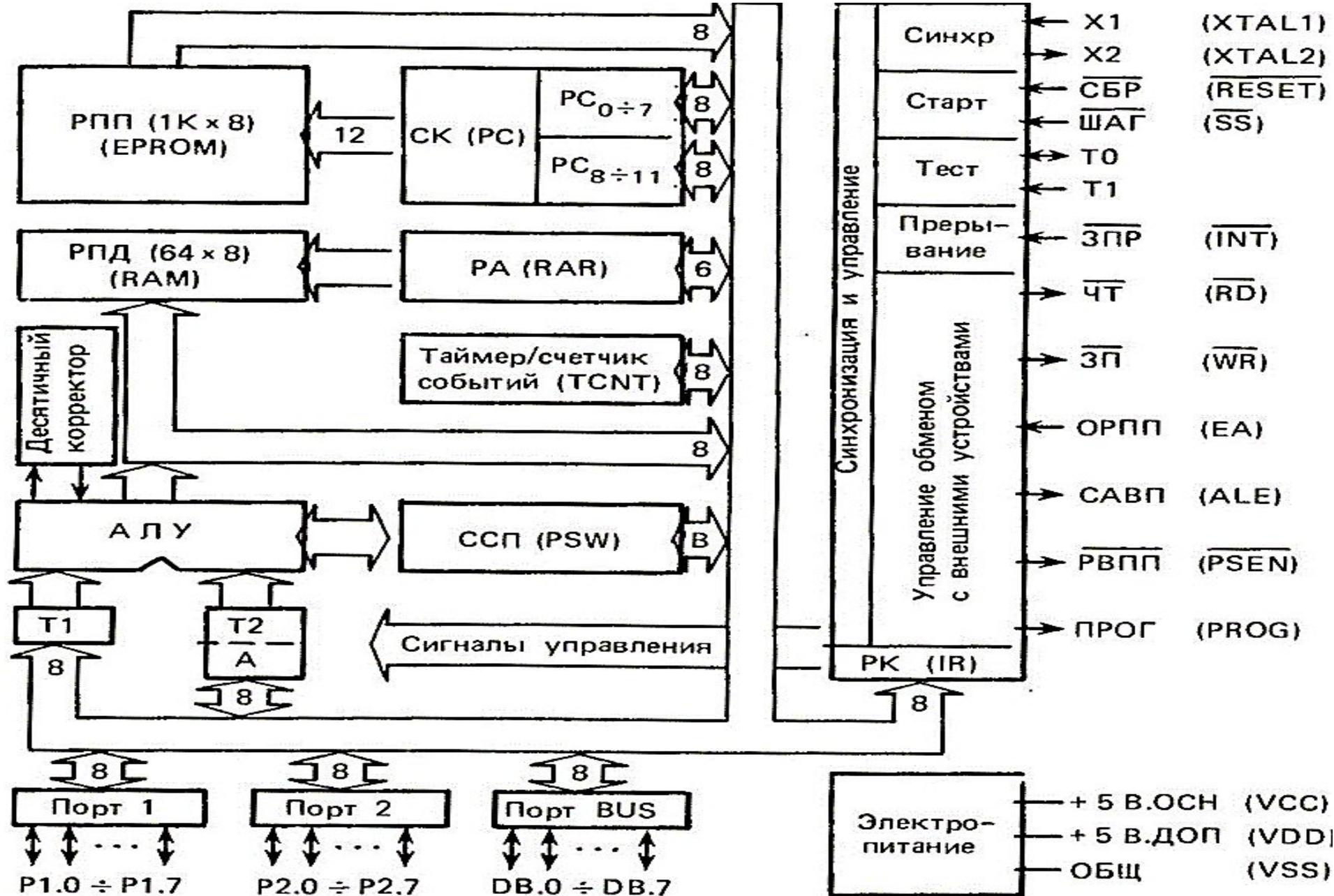
# Схемы синхронизации



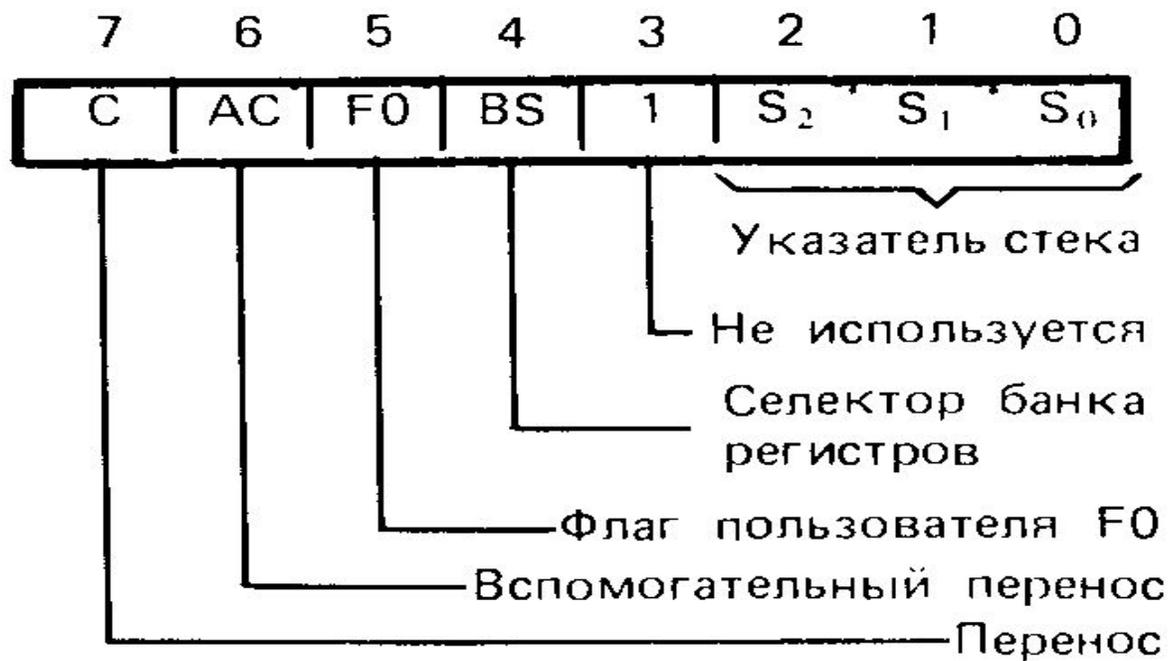
## Начальная установка



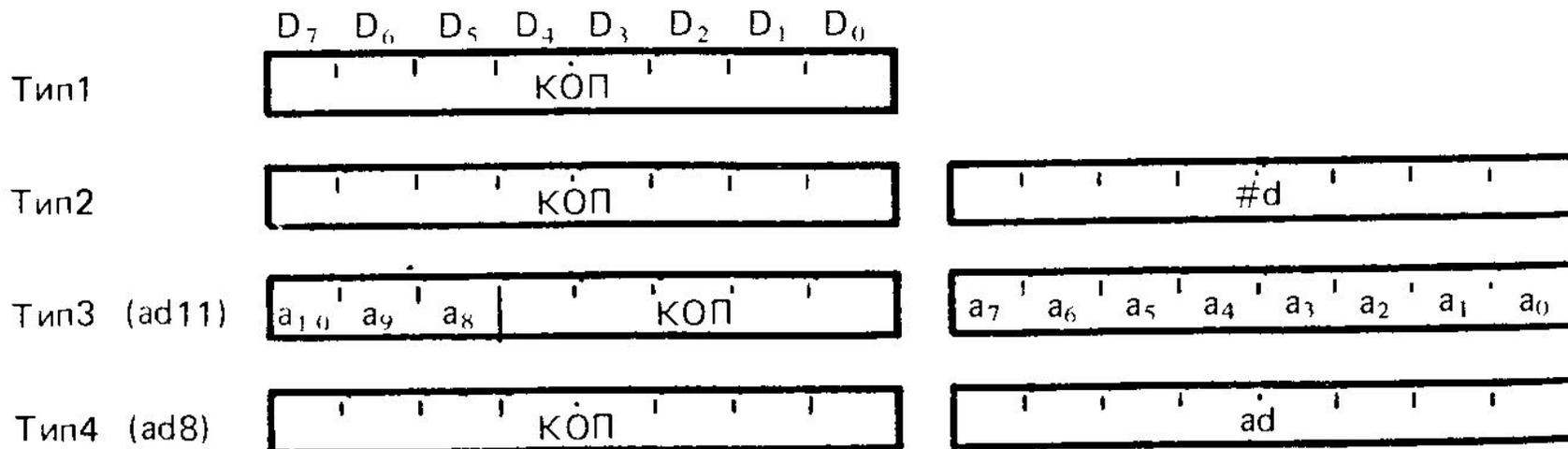
# Архитектура I8748



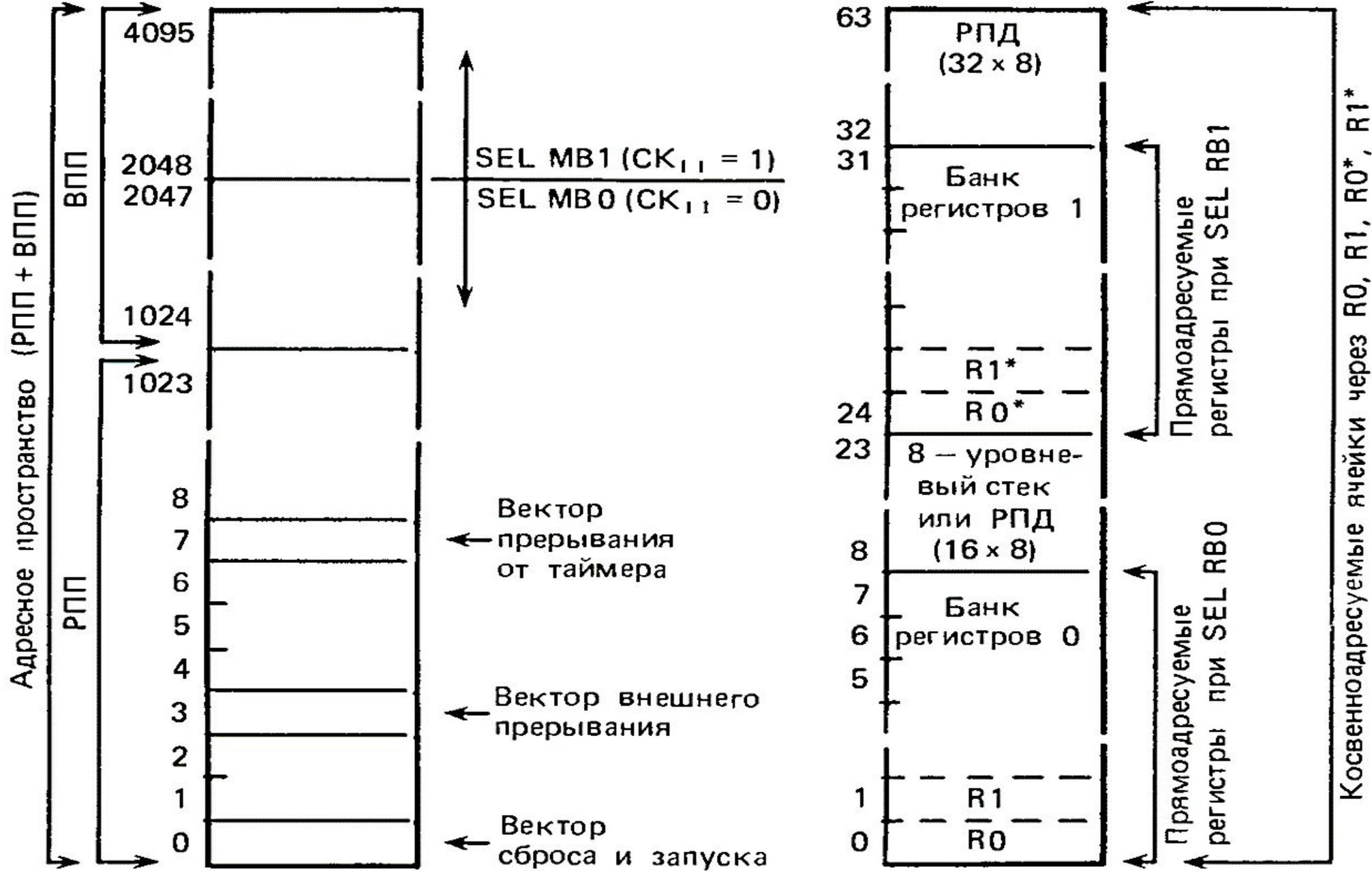
# Слово состояния



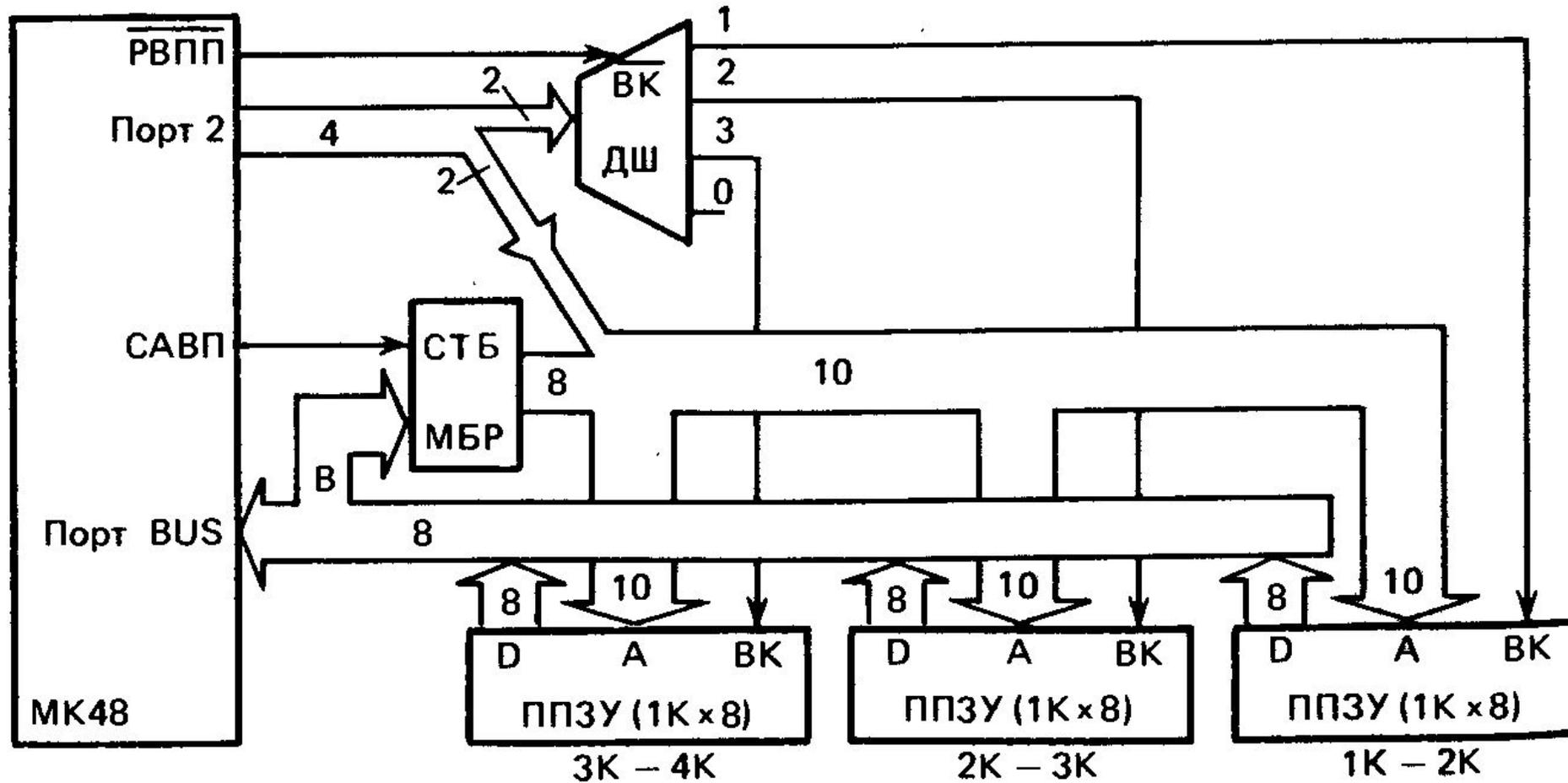
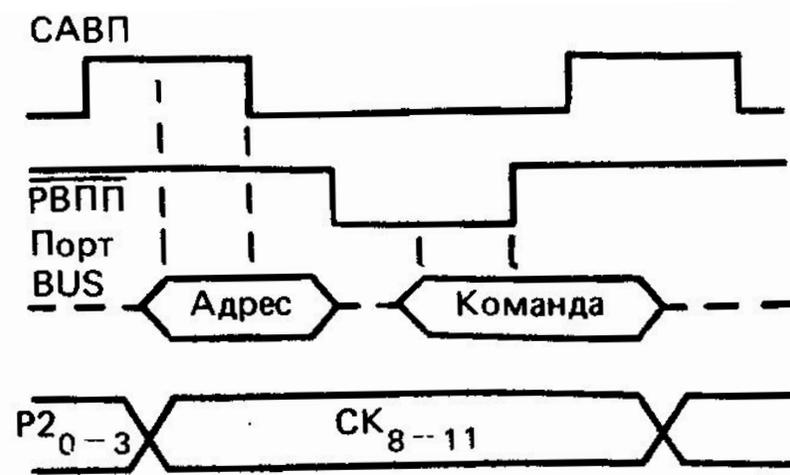
# Типы команд



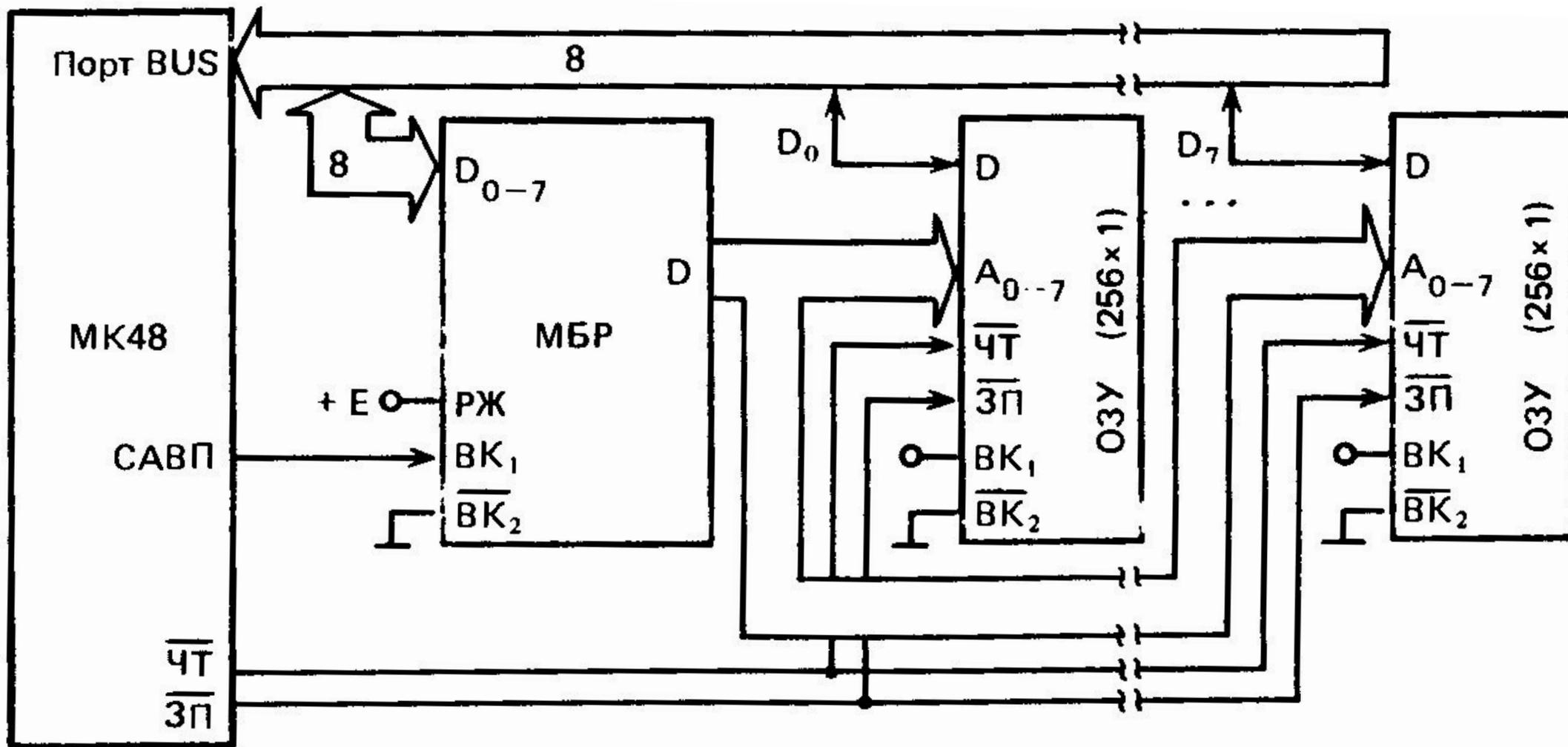
# Карта памяти



# Подключение внешней памяти программ



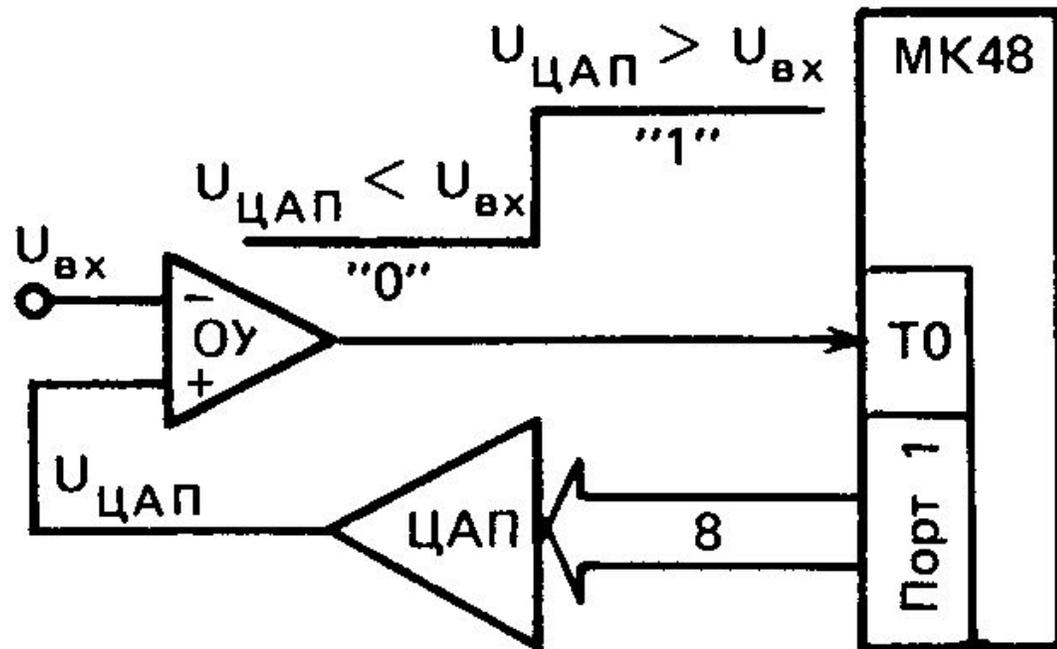
# Подключение внешней памяти данных



# Ввод информации от аналогового датчика

```

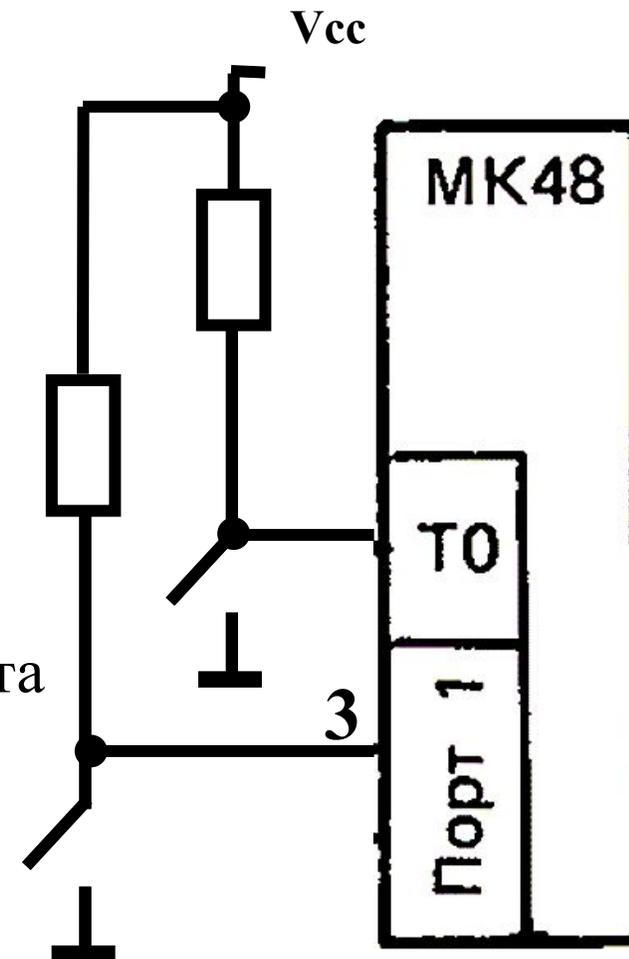
;R4 - РЕГИСТР ЦИФРОВОГО ЭКВИВАЛЕНТА
;R3 - РЕГИСТР БЕГУЩЕЙ ЕДИНИЦЫ ДЛЯ УКАЗАНИЯ
;      ТЕКУЩЕГО ВЗВЕШИВАЕМОГО БИТА
CONVRT: MOV     R5,#08H ;ЗАГРУЗКА СЧЕТЧИКА  ЦИКЛОВ
        MOV     R3,#1
        MOV     R4,#0
LOOP:   MOV     A,R3      ;СЛИВИГ (R3) ВПРАВО НА 1 РАЗРЯД
        RR     A
        MOV     R3,A
        ORL    A,R4      ;ВЗВЕШИВАНИЕ ТЕКУЩЕГО РАЗРЯДА
        OUTL   P1,A      ;ВЫДАЧА ПРОМЕЖУТОЧНОГО ЦИФРОВОГО
                          ; ЭКВИВАЛЕНТА НА ЦАП
        JTO    ENOUGH    ;ЕСЛИ TO=1, ТО ВЫДАННЫЙ БАЙТ БОЛЬШЕ
                          ; ДВОИЧНОГО ЭКВИВАЛЕНТА UВХ, И
                          ; СОХРАНЯЕТСЯ СТАРОЕ ЗНАЧЕНИЕ R4
                          ; ЕСЛИ TO=0, ТО УСТАНОВЛЕННЫЙ БИТ
        MOV    R4,A      ; ЗАКОМИНАЕТСЯ В РЕГИСТРЕ R4
ENOUGH: DJNZ   R5,LOOP   ;ДЕКРЕМЕНТ R5 И, ЕСЛИ НЕ НУЛЬ, ТО
                          ; ПЕРЕХОД К АНАЛИЗУ СЛЕДУЮЩЕГО (J-1)-ГО БИТА
    
```



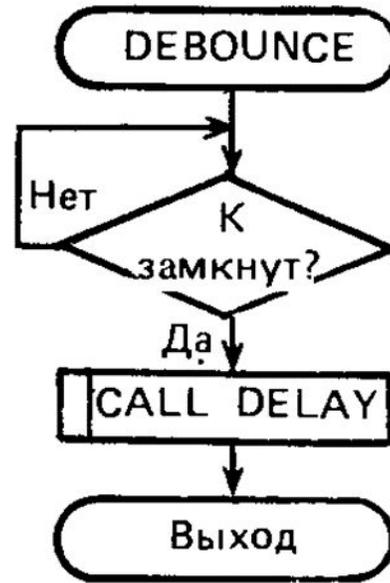
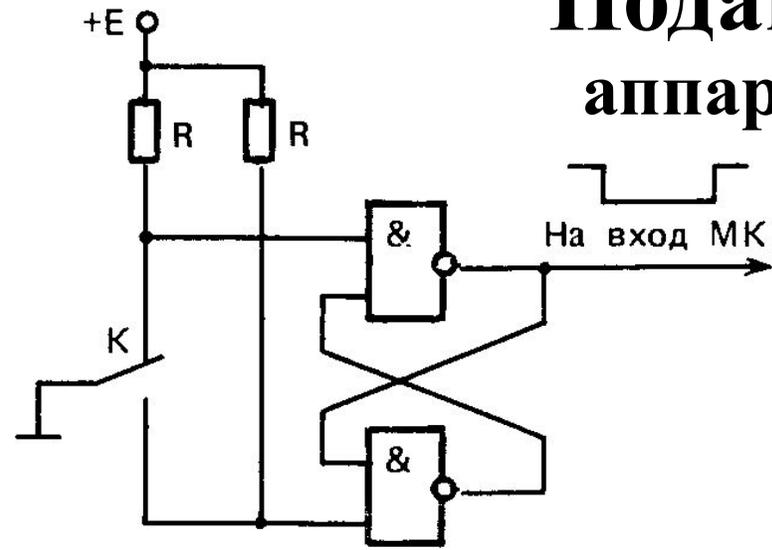
# Подключение контактных датчиков

**Inp:** JT0 Inp ;цикл ожидания  
замыкания контакта

**Inp:** MOV A, 08 ;настройка линии порта  
OUTL P1, A на ввод информации от контакта  
IN A, P1 ;цикл ожидания  
JB3 Inp замыкания контакта



# Подавление дребезга контакта аппаратный и программные способы

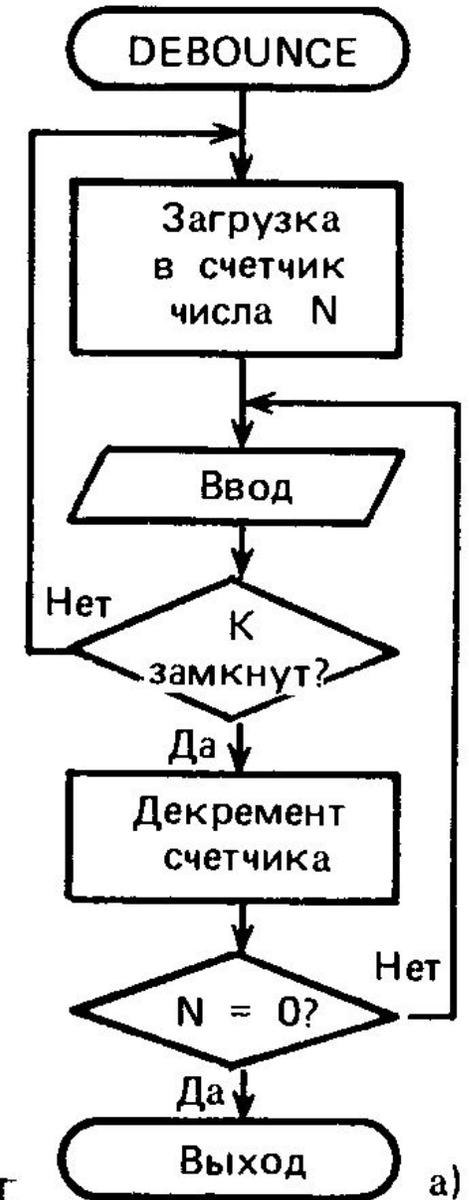


```

DBNCDL: JTO      DBNCDL ;ОЖИДАНИЕ НУЛЯ НА ВХОДЕ ТО
          CALL    DELAY  ;ВЫЗОВ ПОДПРОГРАММЫ ЗАДЕРЖКИ
EXIT:    ...      ;ВЫХОД ИЗ ПРОЦЕДУРЫ
  
```

```

DBNC:    MOV     R3,#20    ;ИНИЦИАЛИЗАЦИЯ СЧЕТЧИКА
DBNC1:   JTO     DBNC     ;ЕСЛИ КОНТАКТ РАЗОМКНУТ, ТО НАЧАТЬ
          ; ОТСЧЕТ СПРОСОВ СНАЧАЛА
          DJNZ   R3,DBNC1  ;ДЕКРЕМЕНТ СЧЕТЧИКА, И ЕСЛИ СОДЕРЖИМОЕ
          ; СЧЕТЧИКА НЕ РАВНО 0, ТО
          ; ПОВТОРИТЬ АНАЛИЗ СОСТОЯНИЯ КОНТАКТА
  
```

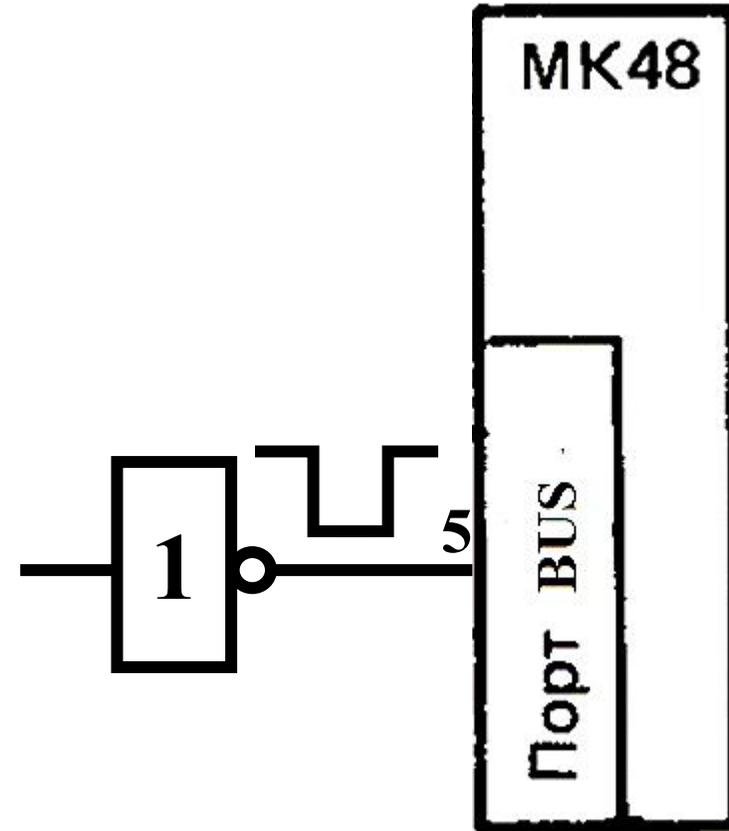


a)

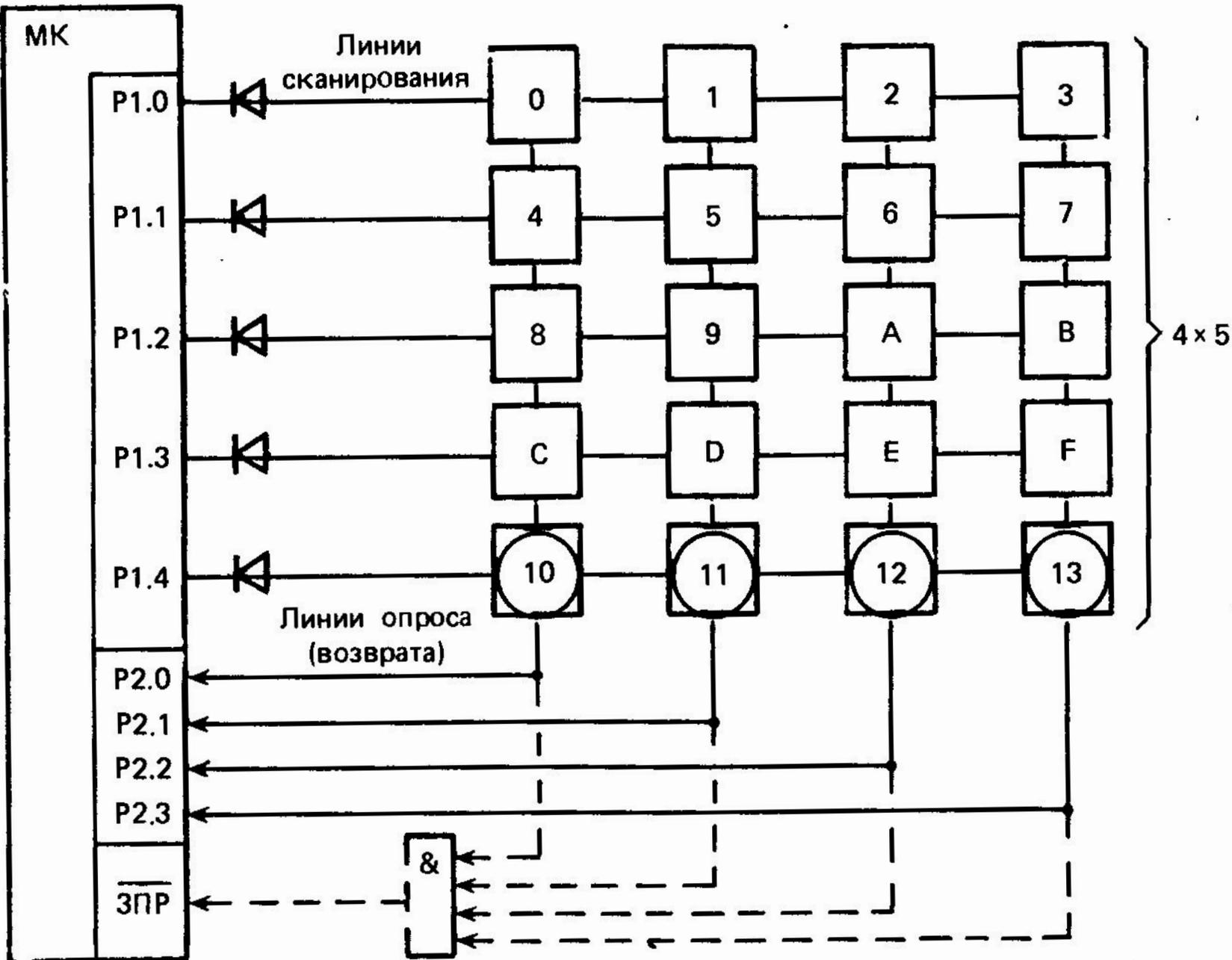
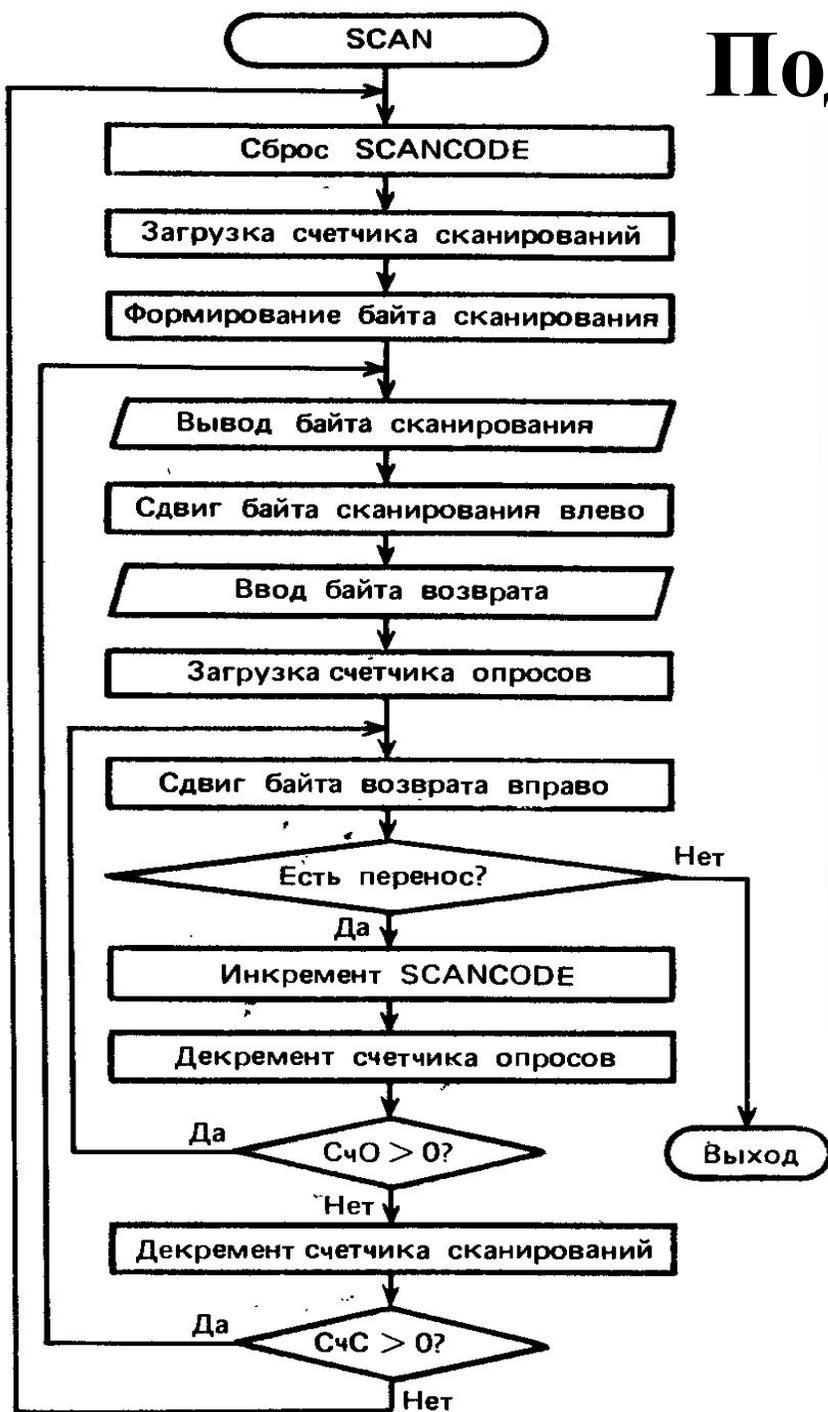
# Подключение импульсного датчика

**Inp:** INS A, BUS ;цикл ожидания  
JB5 Inp нулевого уровня

**Inp1:** INS A, BUS ;цикл ожидания  
CPL A единичного уровня  
JB5 Inp

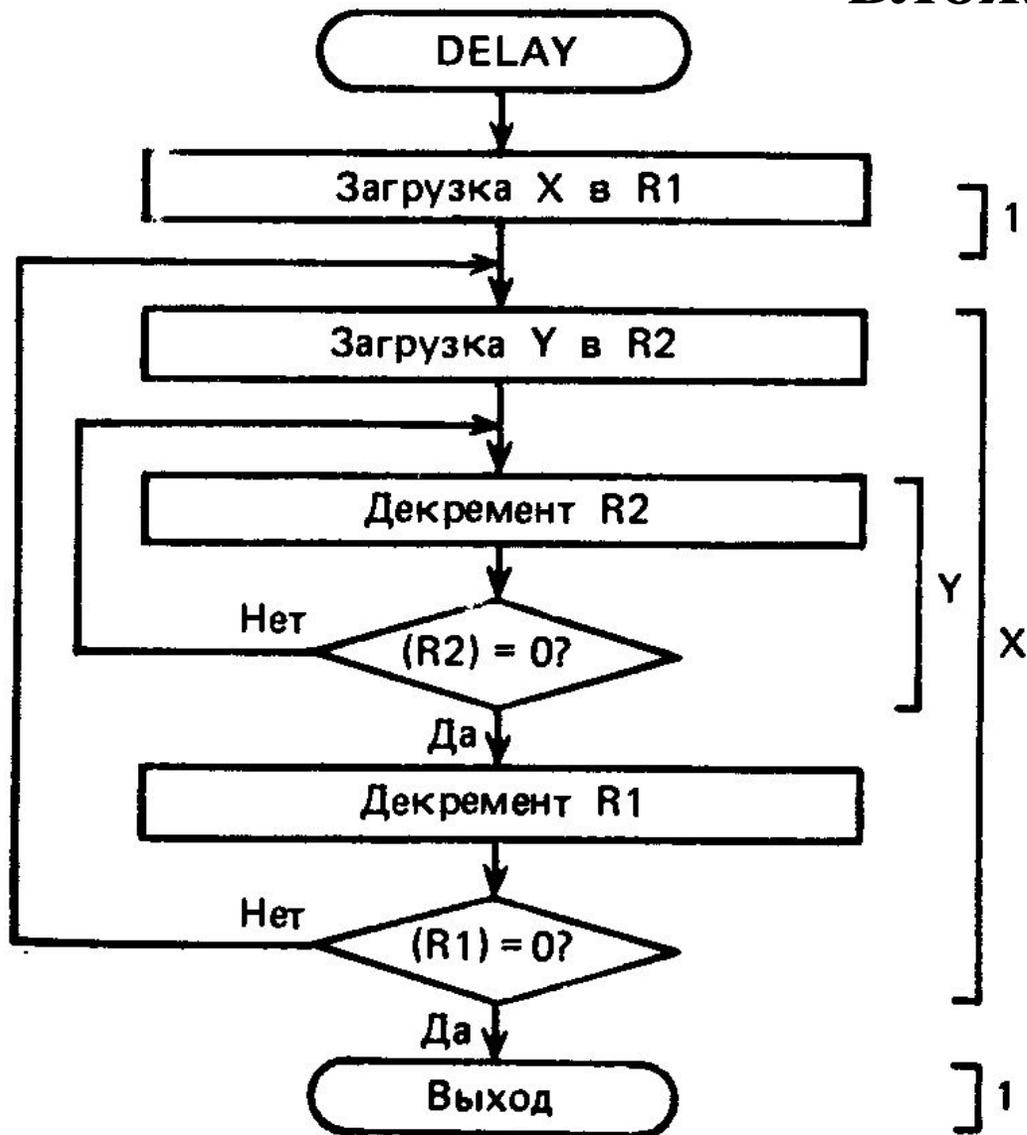


# Подключение клавиатуры



# Формирование и измерение временных интервалов

## ВЛОЖЕННЫМ ЦИКЛОМ



```
DELAY:  MOV    R1,#84      ;ЗАГРУЗКА X
LOOPEX: MOV    R2,#236    ;ЗАГРУЗКА Y
LOOPIN: DJNZ   R2,LOOPIN  ;ДЕКРЕМЕНТ R2 И ВНУТРЕННИЙ ЦИКЛ,
                          ; ЕСЛИ (R2) НЕ РАВНО НУЛЮ
                          DJNZ   R1,LOOPEX  ;ДЕКРЕМЕНТ R1 И ВНЕШНИЙ ЦИКЛ,
                          ; ЕСЛИ (R1) НЕ РАВНО НУЛЮ
LOOPAD: MOV    R3,#4      ;ТОЧНАЯ ПОДСТРОЙКА
        DJNZ   R3,LOOPAD  ; ВРЕМЕННОЙ ЗАДЕРЖКИ
        RET
```

$$T = 5 + 5 + X(5 + 5Y + 5) + 5,$$

где  $T$  – реализуемый временной интервал  
в микросекундах = 100 000 мкс

# Формирование и измерение временных интервалов с использованием таймера

```
START:  DIS    TCNT1    ;ЗАПРЕТ ПРЕРЫВАНИЯ ОТ ТАЙМЕРА
        CLR    A        ;СБРОС АККУМУЛЯТОРА
        MOV    T,A      ;СБРОС ТАЙМЕРА
        MOV    R5,A     ;СБРОС РЕГИСТРА 5
        STRT   T        ;ЗАПУСК ТАЙМЕРА
M1:     JTF    COUNT    ;ЕСЛИ TF=1, ТО ПЕРЕХОД К COUNT И СБРОС TF
        JMP    M1      ;ЦИКЛ
COUNT: INC    R5      ;ИНКРЕМЕНТ РЕГИСТРА 5
        MOV    A,R5    ;ПЕРЕСЫЛКА СОДЕРЖИМОГО R5 В АККУМУЛЯТОР
        JB3   INT      ;ПЕРЕХОД К ПОДПРОГРАММЕ ОБСЛУЖИВАНИЯ ПРЕРЫВАНИЯ INT,
        ; ЕСЛИ БИТ A.3 РАВЕН 1
        JMP    M1      ;ПЕРЕХОД, ЕСЛИ БИТ A.3 НЕ РАВЕН 1
        ...
        ...
        ...
INT:    STOP   TCNT    ;ОСТАНОВ ТАЙМЕРА
        JMP    07H    ;ПЕРЕХОД К ЯЧЕЙКЕ 7 (ВЕКТОР ПРЕРЫВАНИЯ
        ; ОТ СЧЕТЧИКА СОБЫТИЯ)
```

# Передача в последовательном коде

```

        ANL      P1, #0F7H      ;ВЫДАЧА СТАРТ-БИТА
        CALL    DELAYT         ;ВРЕМЕННАЯ ЗАДЕРЖКА T
PSCONV: MOV      R7, #8        ;ИНИЦИАЛИЗАЦИЯ СЧЕТЧИКА БИТ
ROTATE: RRC     A              ;СДВИГ ВПРАВО, Т.Е. (C) (←← A0
        JC      OFF           ;ЕСЛИ ПЕРЕНОС, ТО УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ
                                ; ПРОЦЕДУРЕ ВЫДАЧИ 1
ON:     ANL      P1, #11110111B ;ВЫДАЧА НА P1.3 СИГНАЛА 0
        JMP     DLY
OFF:    ORL      P1, #00001000B ;ВЫДАЧА НА P1.3 СИГНАЛА 1
        JMP     DLY
DLY:    CALL    DELAYT         ;ВРЕМЕННАЯ ЗАДЕРЖКА T
        DJNZ   R7, ROTATE     ;ДЕКРЕМЕНТ СЧЕТЧИКА БИТ И ЦИКЛА,
                                ; ЕСЛИ НЕ НУЛЬ
        ORL      P1, #08H     ;ВЫДАЧА СТОП-БИТА
        CALL    DELAYT         ;ВРЕМЕННАЯ ЗАДЕРЖКА T
EXIT:   ...                  ;ВЫХОД ИЗ ПРОЦЕДУРЫ
```

# Приём последовательного кода

|         |      |         |   |
|---------|------|---------|---|
| WAIT:   | JTO  | WAIT    | :ОЖИДАНИЕ СТАРТ-БИТА  |
|         | CALL | DELAYX  | :ЗАДЕРЖКА T/2   |
|         | JTO  | WAIT    | :ЕСЛИ TO=1, ТО ПОВТОРИТЬ  |
| SPCONV: | MOV  | R7,#8   | :ЗАГРУЗКА СЧЕТЧИКА БИТ, N=8   |
|         | CLR  | A       | :СБРОС АККУМУЛЯТОРА   |
| LOOP:   | CLR  | C       | :СБРОС ФЛАГА ПЕРЕНОСА   |
|         | CALL | DELAYT  | :ЗАДЕРЖКА НА ВРЕНЯ T  |
|         | JNTO | ROTATE  | :ЕСЛИ TO=0, ТО СОХРАНЕНИЕ НУЛЕВОГО<br>: СОСТОЯНИЯ ФЛАГА ПЕРЕНОСА, ИНАЧЕ |
|         | CPL  | C       | : УСТАНОВКА ФЛАГА ПЕРЕНОСА  |
| ROTATE: | RRC  | A       | :САМЫЙ ПАРАЛЛЕЛЬНЫЙ КОДА  |
|         | DJNZ | R7,LOOP | :ДЕКРЕМЕНТ СЧЕТЧИКА БИТ,<br>: И ЕСЛИ НЕ НУЛЬ, ТО ЦИКЛ                   |
| EXIT:   | ...  |         | :ВХОД ИЗ ПРОЦЕДУРЫ  |

# Контроль паритета

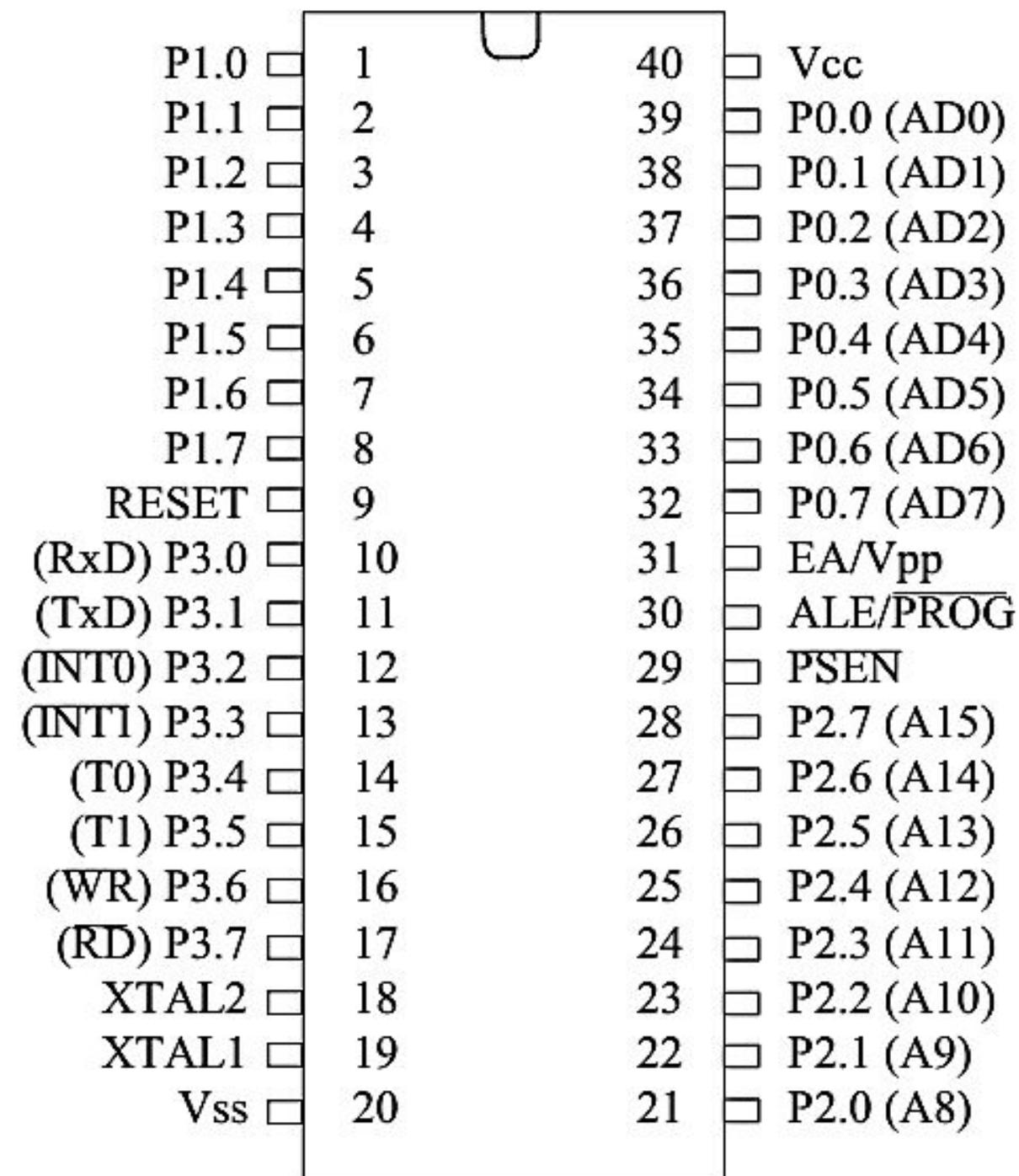
после выполнения программы  
аккумулятор сохранит своё значение,  
флаг пользователя (в состава PSW) будет установлен,  
если число единиц в аккумуляторе было нечетно

```
      CLR      FO      ;СБРОС FO
      MOV      R7,#8   ;ЧИСЛО ПОВТОРОВ
LOOP:  RRC      A      ;ПЕРЕСЫЛКА БИТА A.0 В ПЕРЕНОС
      JNC      NEXT   ;ПРОПУСТИТЬ, ЕСЛИ БИТ РАВЕН 0
      CPL      FO     ;ПОДСЧЕТ ПАРИТЕТА
NEXT:  DJNZ    R7,LOOP ;ПОВТОРИТЬ 8 РАЗ
```

**MCS 51**

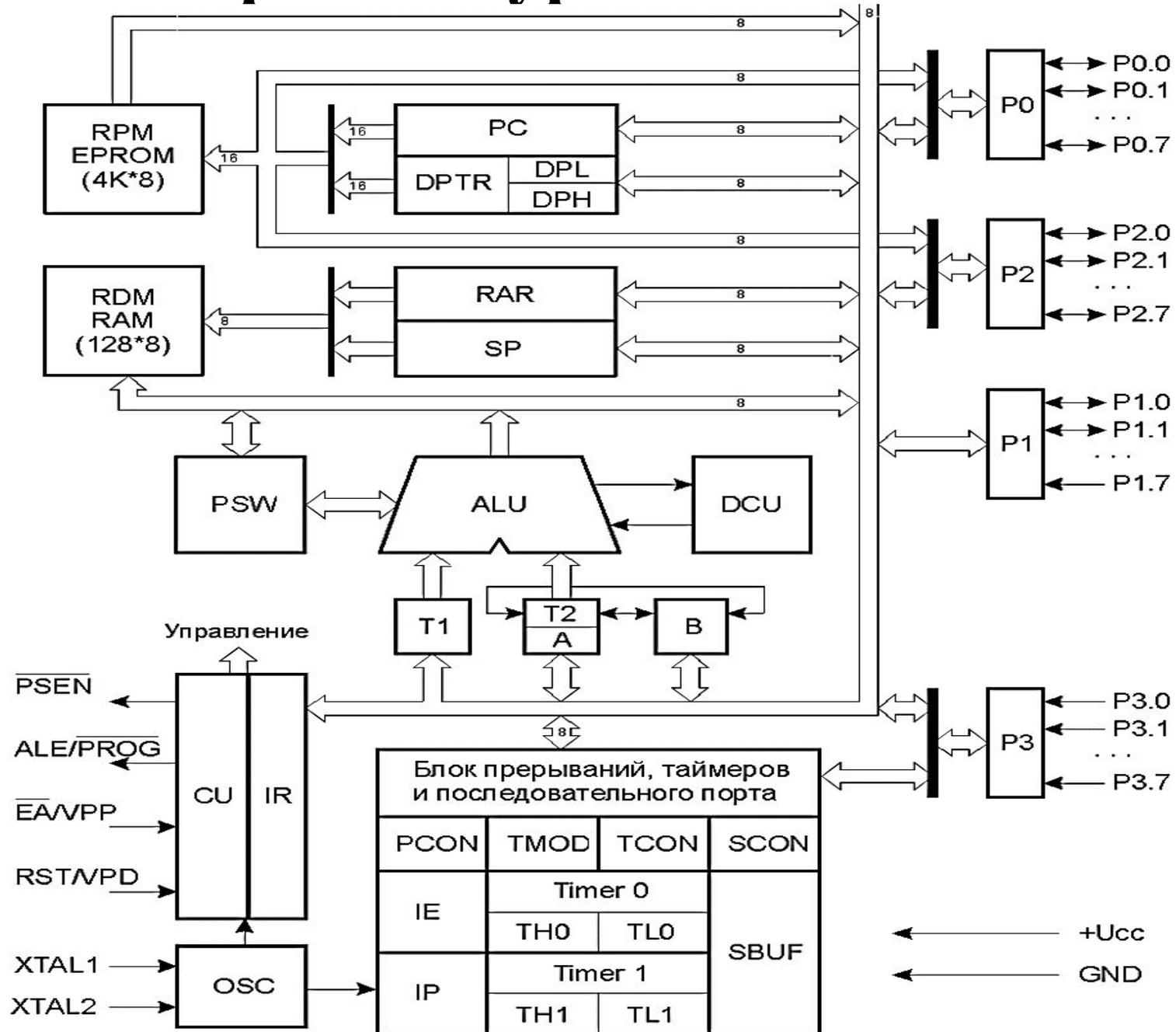
# Микроконтроллеры семейства MCS 51 и его аналоги

| Type        | Manufacturer | Max. Clock Rate | ROM (Byte)     | RAM / XRAM (Byte) | Max. I/O Lines | Timer (16-Bit) | Serial I/O | ADC-In/Res | Others                     | PDF-Size (Data Sheet) |
|-------------|--------------|-----------------|----------------|-------------------|----------------|----------------|------------|------------|----------------------------|-----------------------|
| AT-89C51    | Atmel        | 24              | 4k Flash       | 128               | 32             | 2              | 1          | -          | -                          | <a href="#">122k</a>  |
| AT-89C52    | Atmel        | 24              | 8k Flash       | 256               | 32             | 3              | 1          | -          | -                          | <a href="#">173k</a>  |
| AT-89C1051  | Atmel        | 24              | 1k Flash       | 64                | 15             | 2              | 1          | -          | Analog Comp.               | <a href="#">219k</a>  |
| AT-89C2051  | Atmel        | 24              | 2k Flash       | 128               | 15             | 2              | 1          | -          | Analog Comp.               | <a href="#">246k</a>  |
| DS-80C310   | Dallas       | 33              | -              | 256               | 32             | 3              | 1          | -          | 2 Data Pointer             | <a href="#">327k</a>  |
| DS-80C320   | Dallas       | 33              | -              | 256               | 32             | 3              | 2          | -          | Low Power, WDT             | <a href="#">397k</a>  |
| DS-80C323   | Dallas       | 18              | -              | 256               | 32             | 3              | 2          | -          | Low Power, WDT             | <a href="#">397k</a>  |
| DS-87C520   | Dallas       | 33              | 16k            | 256 / 1k          | 32             | 3              | 2          | -          | WDT                        | <a href="#">470k</a>  |
| DS-87C530   | Dallas       | 33              | 16k            | 256 / 1k          | 32             | 3              | 2          | -          | RTC, WDT                   | <a href="#">343k</a>  |
| I-8XC51     | Intel        | 24              | 4k             | 128               | 32             | 2              | 1          | -          | -                          | <a href="#">288k</a>  |
| I-8XC51FX   | Intel        | 20              | 8k-32k         | 256               | 32             | 2              | 1          | -          | PCA                        | <a href="#">284k</a>  |
| SC-8XC51    | Philips      | 33              | 4k             | 128               | 32             | 2              | 1          | -          | -                          | <a href="#">411k</a>  |
| P-8XC52     | Philips      | 33              | 8k             | 256               | 32             | 3              | 1          | -          | -                          | <a href="#">380k</a>  |
| SC-8XC451   | Philips      | 16              | 4k             | 128               | 56             | 2              | 1          | -          | Parallel Interface         | <a href="#">315k</a>  |
| P-8XC550    | Philips      | 16              | 4k             | 128               | 24+8in         | 2              | 1          | 8 / 8      | WDT                        | <a href="#">292k</a>  |
| PCB-8XC552  | Philips      | 24              | 8k             | 256               | 40+8in         | 2              | 2          | 8 / 10     | I2C, WDT, PWM              | <a href="#">403k</a>  |
| PCB-8XC562  | Philips      | 16              | 8k             | 256               | 40+8in         | 3              | 1          | 8 / 8      | WDT, PWM, CAPCOM           | <a href="#">247k</a>  |
| P-8XC575    | Philips      | 16              | 8k             | 256               | 32             | 3              | 1          | -          | WDT, PCA, 4Analog Comp.    | <a href="#">508k</a>  |
| P-8XC576    | Philips      | 16              | 8k             | 256               | 30             | 3              | 1          | 6 / 10     | WDT, PWM, 4Analog Comp.    | <a href="#">615k</a>  |
| SAB-80C5X5  | Siemens      | 20              | 8k             | 256               | 40+8in         | 3              | 1          | 8 / 8      | WDT, PWM, WDU              |                       |
| SAB-80C515A | Siemens      | 18              | -              | 256 / 1k          | 40+8in         | 3              | 1          | 8 / 10     | WDT, PWM, WDU              |                       |
| SAB-80C5X7  | Siemens      | 16              | 8k             | 256               | 56+12in        | 4              | 2          | 12 / 8     | WDT, PWM, WDU              |                       |
| SAB-80C517A | Siemens      | 18              | -              | 256 / 1k          | 56+12in        | 4              | 2          | 12 / 10    | WDT, WDU, PWM              |                       |
| SAB-C501    | Siemens      | 40              | 8k             | 256               | 32             | 3              | 1          | -          | -                          |                       |
| SAB-C502    | Siemens      | 20              | 16k            | 256 / 256         | 32             | 3              | 1          | -          | progr. XRAM-Start Adr, WDU |                       |
| SAB-C504    | Siemens      | 40              | 16k            | 256 / 256         | 32             | 3              | 1          | 8 / 10     | PWM for motor, WDU         |                       |
| SAB-C509    | Siemens      | 16              | -              | 256 / 3k          | 64+15in        | 5              | 2          | 15 / 10    | WDT, PWM, WDU              |                       |
| SAB-C511    | Siemens      | 12              | 2,5k           | 128               | 32             | 2              | 1 SSC      | -          | -                          |                       |
| SAB-C511A   | Siemens      | 12              | 4k             | 256               | 32             | 2              | 1 SSC      | -          | -                          |                       |
| SAB-C513    | Siemens      | 12              | 8k             | 256               | 32             | 3              | 2          | -          | SSC                        |                       |
| SAB-C513A   | Siemens      | 12              | 16k            | 256 / 256         | 32             | 3              | 2          | -          | SSC, EEPROM available      |                       |
| SAB-C515C   | Siemens      | 10              | ROMie 10 / 64k | 256 / 1k          | 49+8in         | 3              | 2          | 8 / 10     | Full CAN Interface,        |                       |



- **U** — потенциал общего провода ("земли");
- **U** — основное напряжение литания +5 В;
- **X1, X2** — подкл. кварцевого резонатора;
- **RST** — вход сброса микроконтроллера;
- **PSEN** — разрешение ВПП;
- **ALE** — строб адреса внешней памяти;
- **EA** — отключение внутренней ПП;
- **P1** — 8-битный квазидвунаправленный порт ввода/вывода;
- **P2** — 8-битный квазидвунаправленный порт ввода/вывода;
- **P3** — 8-битный квазидвунаправленный порт ввода/вывода; может выполнять альтернативные функции;
- **P0** — 8-битный двунаправленный порт ввода-вывода

# Архитектура Intel 8751



# Устройство управления и синхронизации

Кварцевый резонатор,  
подключаемый к внешним выводам микроконтроллера,  
управляет работой внутреннего генератора,  
который в свою очередь формирует сигналы синхронизации.

Устройство управления (CU) на основе сигналов синхронизации формирует машинный цикл фиксированной длительности, равной 12 периодам резонатора.

Большинство команд микроконтроллера выполняется за один машинный цикл.

Некоторые команды, оперирующие с 2-байтными словами или связанные с обращением к внешней памяти, выполняются за два машинных цикла.

Команды деления и умножения требуют четырех машинных циклов.

К устройству управления примыкает регистр команд

# Организация ОЗУ, ПЗУ и регистров

Объем резидентной ПП – 4 Кбайт.

При обращении к внешней памяти программ все микроконтроллеры семейства 8051 используют 16-разрядный адрес, что обеспечивает доступ к 64 Кбайт ПЗУ.

Объем резидентной ПД – 128 байт.

Объем внешней памяти данных до 64 Кбайт.

первые 32 байта организованы в четыре банка РОН (0 – 3)

каждый банк из восьми регистров R0 — R7.

В любой момент программе доступен только один банк регистров, номер которого содержится в третьем и четвертом битах слова состояния PSW

# Регистры специальных функций (Special Function Register)

| Адрес | Символ | Наименование   |
|-------|--------|--|
| 0E0H  | *ACC   | Аккумулятор (Accumulator)  |
| 0F0H  | *B     | Регистр расширитель аккумулятора (Multiplication Register)               |
| 0D0H  | *PSW   | Слово состояния программы (Program Status Word)                          |
| 080H  | *P0    | Порт 0 (SFR P0)  |
| 090H  | *P1    | Порт 1 (SFR P1)  |
| 0A0H  | *P2    | Порт 2 (SFR P2)  |
| 0B0H  | *P3    | Порт 3 (SFR P3)  |
| 081H  | SP     | Регистр указатель стека (Stack Pointer)                                  |
| 083H  | DPH    | Старший байт регистра указателя данных DPTR (Data Pointer High)          |
| 082H  | DPL    | Младший байт регистра указателя данных DPTR (Data Pointer Low)           |
| 08CH  | TH0    | Старший байт таймера 0 ( )   |
| 08AH  | TL0    | Младший байт таймера 0 ( )   |
| 08DH  | TH1    | Старший байт таймера 1 ( )   |
| 08BH  | TL1    | Младший байт таймера 1 ( )   |
| 089H  | TMOD   | Регистр режимов таймеров счетчиков (Timer/Counter Mode Control Register) |
| 088H  | *TCON  | Регистр управления статуса таймеров (Timer/Counter Control Register)     |
| 0B8H  | *IP    | Регистр приоритетов (Interrupt Priority Control Register)                |
| 0A8H  | *IE    | Регистр маски прерывания (Interrupt Enable Register)                     |
| 087H  | PCON   | Регистр управления мощностью (Power Control Register)                    |
| 098H  | *SCON  | Регистр управления приемопередатчиком (Serial Port Control Register)     |
| 099H  | SBUF   | Буфер приемопередатчика (Serial Data Buffer)                             |

# Карта адресуемых бит в РПД

Адреса

(D<sub>7</sub>)

(D<sub>0</sub>)

7FH

2FH

2EH

2DH

2CH

2BH

2AH

29H

28H

27H

26H

25H

24H

23H

22H

21H

20H

1FH

18H

17H

10H

0FH

08H

07H

00H

|        |    |    |    |    |    |    |    |
|--------|----|----|----|----|----|----|----|
|        |    |    |    |    |    |    |    |
| 7F     | 7E | 7D | 7C | 7B | 7A | 79 | 78 |
| 77     | 76 | 75 | 74 | 73 | 72 | 71 | 70 |
| 6F     | 6E | 6D | 6C | 6B | 6A | 69 | 68 |
| 67     | 66 | 65 | 64 | 63 | 62 | 61 | 60 |
| 5F     | 5E | 5D | 5C | 5B | 5A | 59 | 58 |
| 57     | 56 | 55 | 54 | 53 | 52 | 51 | 50 |
| 4F     | 4E | 4D | 4C | 4B | 4A | 49 | 48 |
| 47     | 46 | 45 | 44 | 43 | 42 | 41 | 40 |
| 3F     | 3E | 3D | 3C | 3B | 3A | 39 | 38 |
| 37     | 36 | 35 | 34 | 33 | 32 | 31 | 30 |
| 2F     | 2E | 2D | 2C | 2B | 2A | 29 | 28 |
| 27     | 26 | 25 | 24 | 23 | 22 | 21 | 20 |
| 1F     | 1E | 1D | 1C | 1B | 1A | 19 | 18 |
| 17     | 16 | 15 | 14 | 13 | 12 | 11 | 10 |
| 0F     | 0E | 0D | 0C | 0B | 0A | 09 | 08 |
| 07     | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| Банк 3 |    |    |    |    |    |    |    |
| Банк 2 |    |    |    |    |    |    |    |
| Банк 1 |    |    |    |    |    |    |    |
| Банк 0 |    |    |    |    |    |    |    |

# Карта адресуемых бит

## в блоке регистров специальных функций

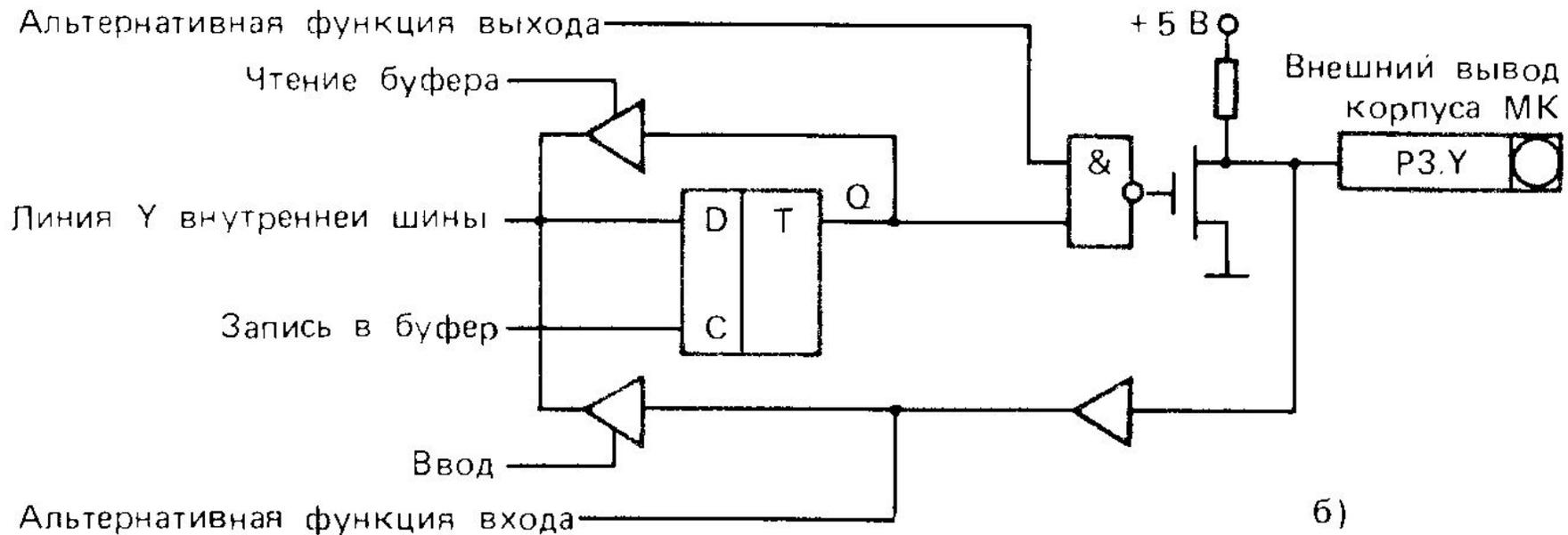
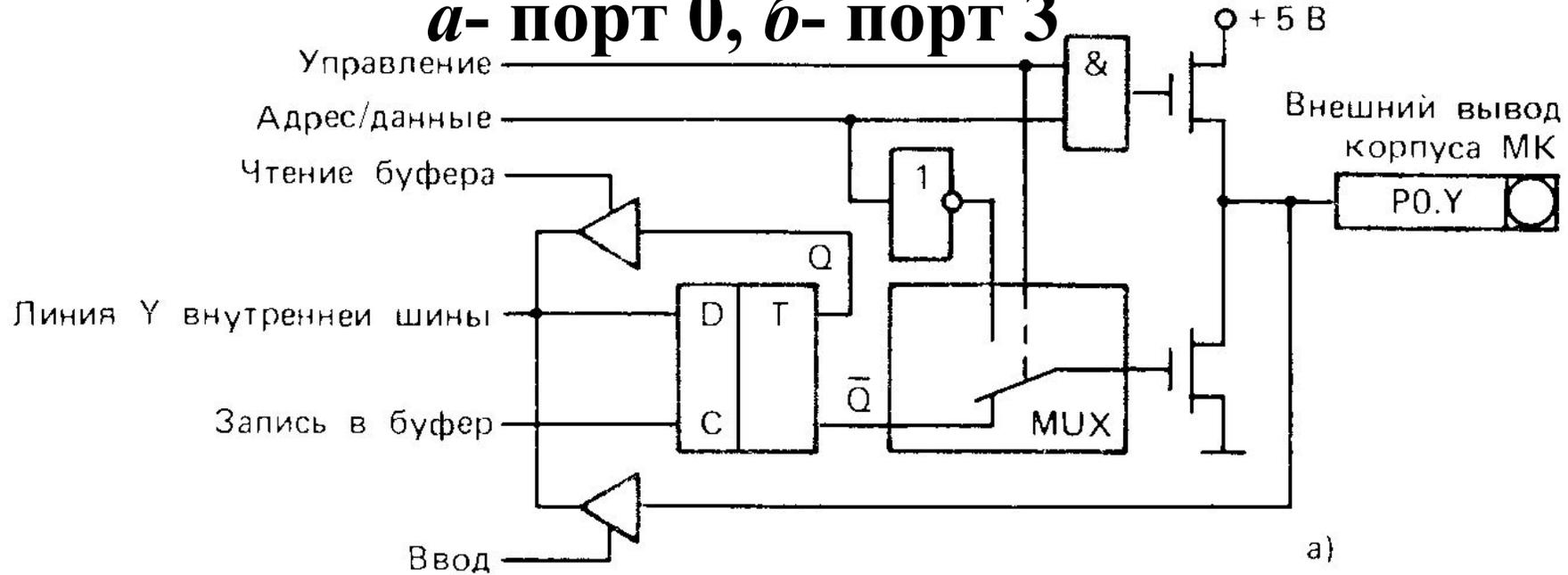
| Адрес<br>байта | Адреса битов по разрядам |    |    |    |    |    |    |    | Имя<br>регистра |
|----------------|--------------------------|----|----|----|----|----|----|----|-----------------|
|                | Adr                      | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0              |
| F0H            | F7                       | F6 | F5 | F4 | F3 | F2 | F1 | F0 | B               |
| ...            | ...                      |    |    |    |    |    |    |    | ...             |
| E0H            | E7                       | E6 | E5 | E4 | E3 | E2 | E1 | E0 | ACC             |
| ...            | ...                      |    |    |    |    |    |    |    | ...             |
| D0H            | D7                       | D6 | D5 | D4 | D3 | D2 | D1 | D0 | PSW             |
| ...            | ...                      |    |    |    |    |    |    |    | ...             |
| B8H            | -                        | -  | -  | BC | BB | BA | B9 | B8 | IP              |
| ...            | ...                      |    |    |    |    |    |    |    | ...             |
| B0             | B7                       | B6 | B5 | B4 | B3 | B2 | B1 | B0 | P3              |
| ...            | ...                      |    |    |    |    |    |    |    | ...             |
| A8H            | AF                       | -  | -  | AC | AB | AA | A9 | A8 | IE              |
| ...            | ...                      |    |    |    |    |    |    |    | ...             |
| A0H            | A7                       | A6 | A5 | A4 | A3 | A2 | A1 | A0 | P2              |
| ...            | ...                      |    |    |    |    |    |    |    | ...             |
| 98H            | 9F                       | 9E | 9D | 9C | 9B | 9A | 99 | 98 | SCON            |
| ...            | ...                      |    |    |    |    |    |    |    | ...             |
| 90H            | 97                       | 96 | 95 | 94 | 93 | 92 | 91 | 90 | P1              |
| ...            | ...                      |    |    |    |    |    |    |    | ...             |
| 88H            | 8F                       | 8E | 8D | 8C | 8B | 8A | 89 | 88 | TCON            |
| ...            | ...                      |    |    |    |    |    |    |    | ...             |
| 80H            | 87                       | 86 | 85 | 84 | 83 | 82 | 81 | 80 | P0              |

# Регистр флагов (PSW)

| Символ    | Позиция       | Имя и назначение  |                     |     |      |                     |   |   |   |           |   |   |   |           |   |   |   |           |   |   |   |           |
|-----------|---------------|---|---------------------|-----|------|---------------------|---|---|---|-----------|---|---|---|-----------|---|---|---|-----------|---|---|---|-----------|
| P         | PSW.0         | Флаг паритета. Устанавливается и сбрасывается аппаратно в каждом цикле команды и фиксирует нечетное/четное число единичных бит в аккумуляторе   |                     |     |      |                     |   |   |   |           |   |   |   |           |   |   |   |           |   |   |   |           |
| -         | PSW.1         | Не используется   |                     |     |      |                     |   |   |   |           |   |   |   |           |   |   |   |           |   |   |   |           |
| OV        | PSW.2         | Флаг переполнения. Устанавливается и сбрасывается аппаратно при выполнении арифметических операций  |                     |     |      |                     |   |   |   |           |   |   |   |           |   |   |   |           |   |   |   |           |
| RS0 - RS1 | PSW.3 - PSW.4 | <p>Биты выбора используемого банка регистров. Могут быть изменены программным путем</p> <table border="1"> <thead> <tr> <th>RS0</th> <th>RS1</th> <th>Банк</th> <th>Границы адресов ОЗУ</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>00H - 07H</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>08H - 0FH</td> </tr> <tr> <td>0</td> <td>1</td> <td>2</td> <td>10H - 17H</td> </tr> <tr> <td>1</td> <td>1</td> <td>3</td> <td>18H - 1FH</td> </tr> </tbody> </table> | RS0                 | RS1 | Банк | Границы адресов ОЗУ | 0 | 0 | 0 | 00H - 07H | 1 | 0 | 1 | 08H - 0FH | 0 | 1 | 2 | 10H - 17H | 1 | 1 | 3 | 18H - 1FH |
| RS0       | RS1           | Банк  | Границы адресов ОЗУ |     |      |                     |   |   |   |           |   |   |   |           |   |   |   |           |   |   |   |           |
| 0         | 0             | 0   | 00H - 07H           |     |      |                     |   |   |   |           |   |   |   |           |   |   |   |           |   |   |   |           |
| 1         | 0             | 1   | 08H - 0FH           |     |      |                     |   |   |   |           |   |   |   |           |   |   |   |           |   |   |   |           |
| 0         | 1             | 2   | 10H - 17H           |     |      |                     |   |   |   |           |   |   |   |           |   |   |   |           |   |   |   |           |
| 1         | 1             | 3   | 18H - 1FH           |     |      |                     |   |   |   |           |   |   |   |           |   |   |   |           |   |   |   |           |
| F0        | PSW.5         | Флаг пользователя. Может быть установлен, сброшен или проверен программой пользователя  |                     |     |      |                     |   |   |   |           |   |   |   |           |   |   |   |           |   |   |   |           |
| AC        | PSW.6         | Флаг вспомогательного переноса. Устанавливается и сбрасывается только аппаратными средствами при выполнении команд сложения и вычитания и сигнализирует о переносе или заеме в бите 3 аккумулятора  |                     |     |      |                     |   |   |   |           |   |   |   |           |   |   |   |           |   |   |   |           |
| C         | PSW.7         | Флаг переноса. Устанавливается и сбрасывается как аппаратно, так и программным путем  |                     |     |      |                     |   |   |   |           |   |   |   |           |   |   |   |           |   |   |   |           |

# Схемотехника портов ввода-вывода мс51,

## а- порт 0, б- порт 3



# Альтернативные функции порта P3

активируются предварительной записью «1»  
в соответствующие биты порта

| Символ            | Позиция | Имя и назначение   |
|-------------------|---------|--|
| $\overline{RD}$   | P3.7    | Чтение. Активный сигнал низкого уровня формируется аппаратно при обращении к ВПД                           |
| $\overline{WR}$   | P3.6    | Запись. Активный сигнал низкого уровня формируется аппаратно при обращении к ВПД                           |
| T1                | P3.5    | Вход таймера/счетчика 1 или тест-вход  |
| T0                | P3.4    | Вход таймера/счетчика 0 или тест-вход  |
| $\overline{INT1}$ | P3.3    | Вход запроса прерывания 1. Воспринимается сигнал низкого уровня или срез                                   |
| $\overline{INT0}$ | P3.2    | Вход запроса прерывания 0. Воспринимается сигнал низкого уровня или срез                                   |
| TXD               | P3.1    | Выход передатчика последовательного порта в режиме УАПП. Выход синхронизации в режиме сдвигающего регистра |
| RXD               | P3.0    | Вход приемника последовательного порта в режиме УАПП. Ввод/вывод данных в режиме сдвигающего регистра      |

# Таймеры / счетчики мс51

два программируемых 16-битных таймера/счетчика (Т/С0 и Т/С1),  
могут быть использованы как в качестве

- **таймеров** – содержимое Т/С инкрементируется в каждом машинном цикле (через каждые 12 периодов колебаний кварцевого резонатора)
- **счетчиков внешних событий** – инкрементируются под воздействием перехода из 1 в 0 внешнего входного сигнала, подаваемого на соответствующий (Т0,Т1) вывод. на распознавание требуется два машинных цикла, максимальная частота подсчета входных сигналов равна  $1/24$  частоты резонатора.

для управления режимами работы Т/С  
и для организации их взаимодействия с системой прерываний  
используются два регистра специальных функций – **TMOD** и **TCON**

# Регистр режима работы таймера/счетчика (TMOD)

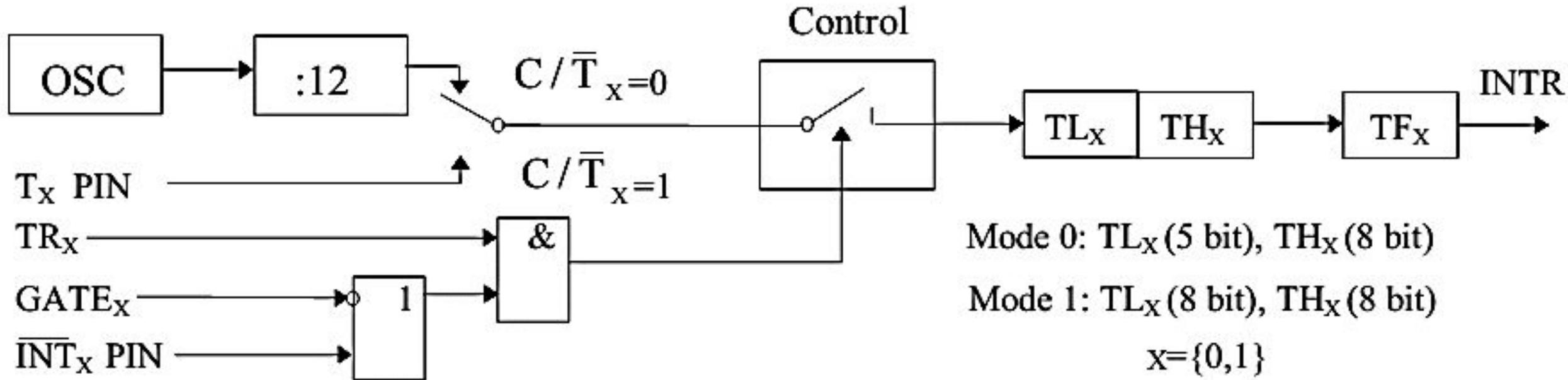
| Символ | Позиция                           | Имя и назначение  |    |   |
|--------|-----------------------------------|---|----|---|
| GATE   | TMOD.7 для T/C1 и TMOD.3 для T/CO | Управление блокировкой. Если бит установлен, то таймер/счетчик "x" разрешен до тех пор, пока на входе "INTx" высокий уровень и бит управления "TRx" установлен. Если бит сброшен, то T/C разрешается, как только бит управления "TRx" устанавливается |    |   |
| C/T    | TMOD.6 для T/C1 и TMOD.2 для T/CO | Бит выбора режима таймера или счетчика событий. Если бит сброшен, то работает таймер от внутреннего источника сигналов синхронизации. Если; установлен, то работает счетчик от внешних сигналов на входе "Tx"   |    |   |
| M1     | TMOD.5 для T/C1 и TMOD.1 для T/CO | <b>Режим работы</b>   |    |   |
|        |                                   | M1  | M0 |   |
| M0     | TMOD.4 для T/C1 и TMOD.0 для T/CO | 0   | 0  | Таймер BE48. "TLx" работает как 5-битный предделитель   |
|        |                                   | 0   | 1  | 16 битный таймер/счетчик. "THx" и "TLx" включен последовательно   |
|        |                                   | 1   | 0  | 8-битный авто перезагружаемый таймер/счетчик. "THx" хранит значение, которое должно быть перезагружено в "TLx" каждый раз по переполнению   |
|        |                                   | 1   | 1  | Таймер/счетчик 1 останавливается. Таймер/счетчик 0: TLO работает как 8-битный таймер/счетчик, и его режим определяется управляющими битами таймера 0. TH0 работает только как 8 битный таймер, и его режим определяется управляющими битами таймера 1 |

# Регистр управления/статуса таймера (TCON)

| Символ | Позиция | Имя и назначение   |
|--------|---------|--|
| TF1    | TCON.7  | <b>Флаг переполнения таймера 1.</b> Устанавливается аппаратно при переполнении таймера/счетчика. Сбрасывается при обслуживании прерывания аппаратно  |
| TR1    | TCON.6  | <b>Бит управления таймера 1.</b> Устанавливается, / сбрасывается программой для пуска/останова   |
| TF0    | TCON.5  | <b>Флаг переполнения таймера 0.</b> Устанавливается аппаратно. Сбрасывается при обслуживании прерывания  |
| TR0    | TCON.4  | <b>Бит управления таймера 0.</b> Устанавливается / сбрасывается программой для пуска/останова таймера/счетчика                                       |
| IE1    | TCON.3  | <b>Флаг фронта прерывания 1.</b> Устанавливается аппаратно, когда детектируется срез внешнего сигнала INT1. Сбрасывается при обслуживании прерывания |
| IT1    | TCON.2  | <b>Бит управления типом прерывания 1.</b> Устанавливается / сбрасывается программно для спецификации запроса INT1 (срез/низкий уровень)              |
| IE0    | TCON.1  | <b>Флаг фронта прерывания 0.</b> Устанавливается по срезу сигнала INT0. Сбрасывается при обслуживании прерывания                                     |
| IT1    | TCON .0 | <b>Бит управления типом прерывания 0.</b> Устанавливается / сбрасывается программно для спецификации запроса INT0 (срез/низкий уровень)              |

# Режимы работы таймеров-счетчиков

для T/C0 и T/C1 режимы работы 0, 1 и 2 одинаковы, режимы 3 различны



**Режим 0:** 8-битный счетчик, к входу которого подключен пяти-битный предделитель частоты (на 32). таймерный регистр имеет разрядность 13 бит.

При переполнении устанавливается флаг прерывания от таймера – TF.

Входной синхросигнал таймера разрешен (поступает на вход T/C), когда управляющий бит TR установлен в 1 либо управляющий бит GATE(блокировка) равен 0, либо на внешний запрос INT поступает уровень 1.

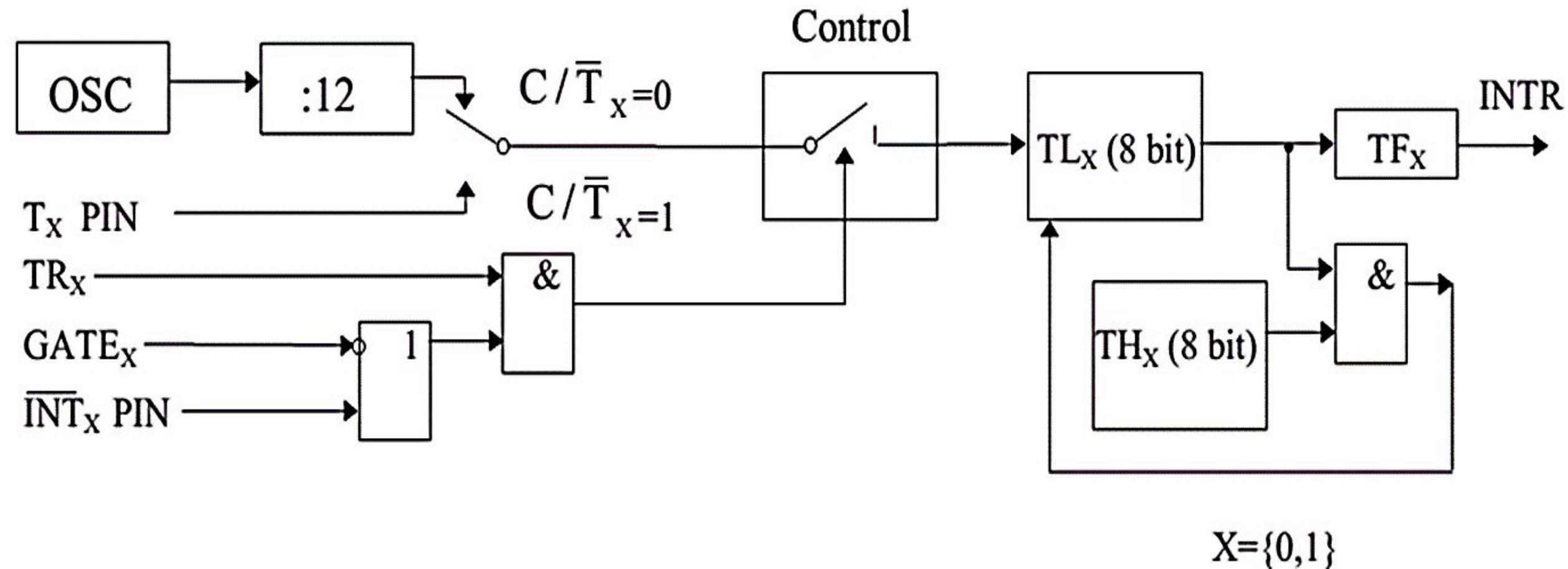
установка бита GATE в 1 позволяет использовать таймер для измерения длительности импульсного сигнала подаваемого на вход запроса прерывания.

**Режим 1:** Работа T/C такая же, как и в режиме 0, но таймерный регистр имеет разрядность 16 бит

## Режим 2:

работа организована таким образом, что переполнение TL приводит не только к установке флага TF,

но и автоматически перезагружает в TL содержимое старшего байта (ТН) таймерного регистра, которое предварительно было задано программным путем; перезагрузка оставляет содержимое ТН неизменным



## Режим 3:

T/C1 сохраняет неизменным свое текущее содержимое, так же как и при сбросе управляющего бита TR1 в 0..

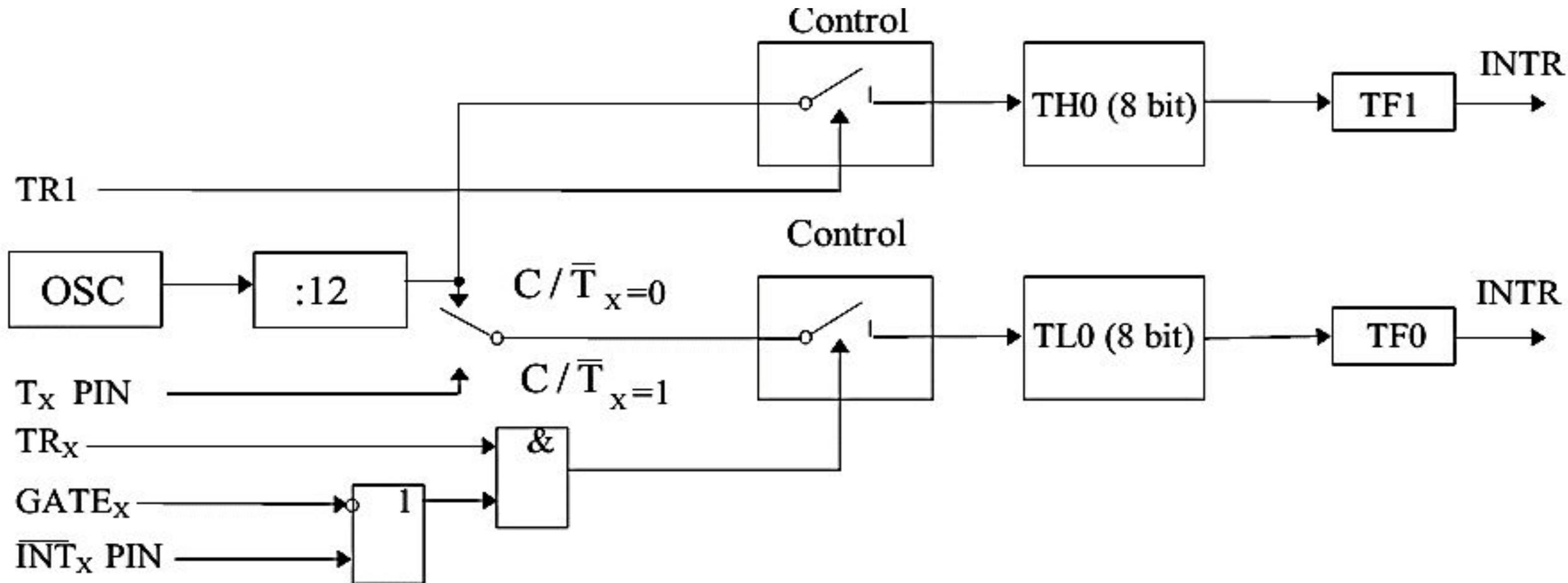
TL0 и TH0 функционируют как два независимых восьмибитных счетчика.

Работу TL0 определяют управляющие биты T/C0 (C/T, GATE TR0), INT0 и флаг переполнения TF0.

Работу TH0, который может выполнять только функции таймера определяет управляющий бит TR1.

При этом TH0 использует флаг переполнения TF1.

Можно считать, что в этом режиме имеется три таймера/счетчика.



# Универсальный асинхронный приемопередатчик UART (Universal Asynchronous Receiver Transmitter)

прием и передача информации в последовательном коде младшими битами вперед, в дуплексном режиме обмена.

В состав приемопередатчика входят принимающий и передающий сдвигающие регистры, а также специальный буферный регистр (**SBUF**) приемопередатчика; запись байта в буфер приводит к автоматической переписи байта в сдвигающий регистр передатчика и инициирует начало передачи;

работой последовательного порта управляют:

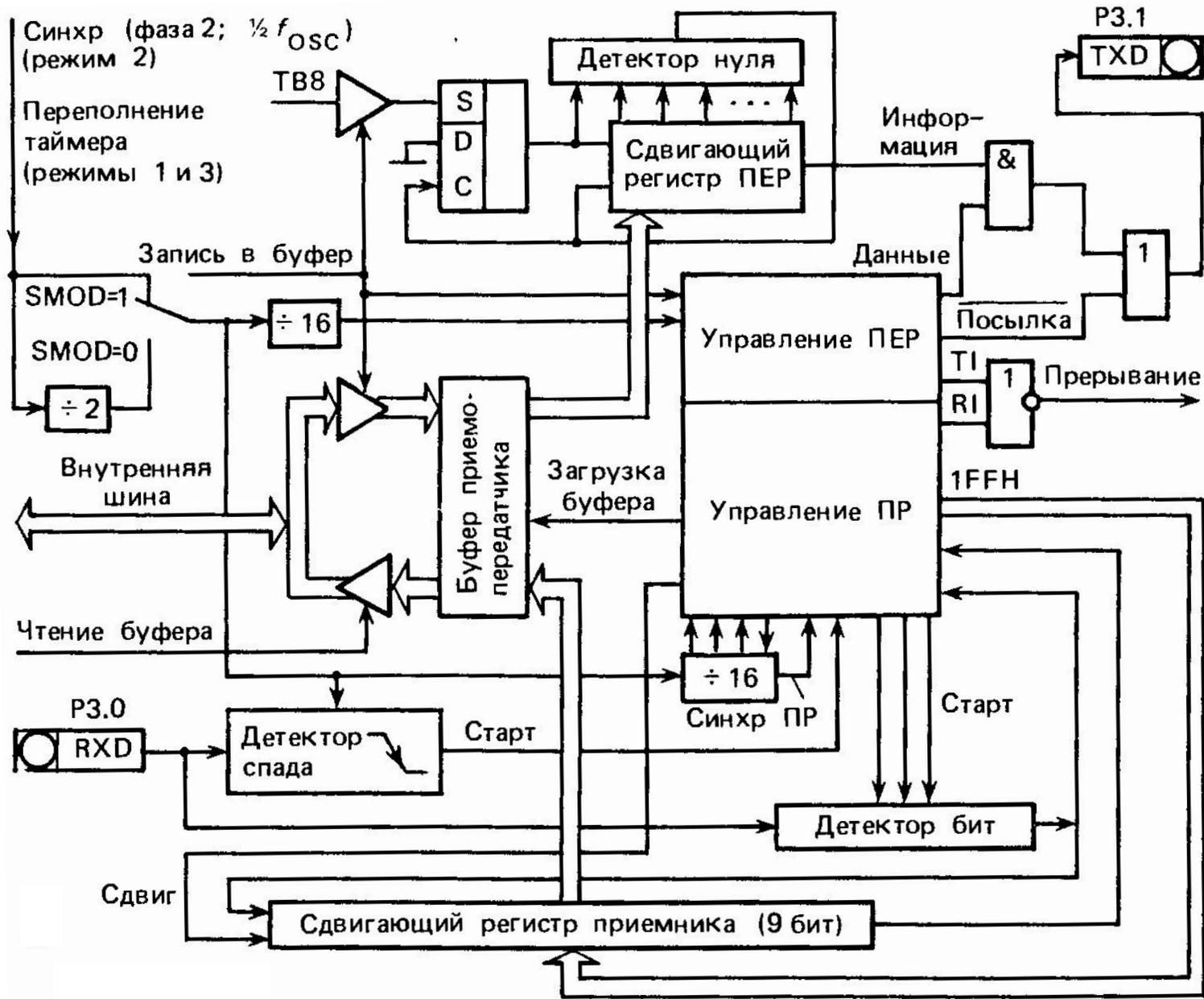
- Регистр управления/статуса приемопередатчика **SCON**
- Бит **SMOD** регистра управления мощностью (**PCON**)

# Режимы работы УАПП

- Режим **0**. Информация и передается, и принимается через вывод входа приемника (RxD). Принимаются или передается 8 бит данных. Через вывод выхода передатчика (TxD) выдаются импульсы сдвига, которые сопровождают каждый бит. Частота передачи бита информации равна  $1/12$  частоты кварцевого резонатора
- Режим **1**. В этом режиме передаются через вывод TxD или принимаются через RxD 10 бит информации: старт-бит (0), 8 бит данных и стоп-бит (1) при приеме информации в бит RB8 регистра управления/статуса приемопередатчика SCON заносится стоп-бит Скорость приема/передачи — величина переменная и задается таймером.
- Режим **2**. В этом режиме через вывод TxD передаются или через RxD принимаются 11 бит информации: старт-бит, 8 бит данных, программируемый девятый бит и стоп-бит. При передаче девятый бит данных может принимать значение 0 или 1 для повышения достоверности передачи путем контроля по четности в него помещается значение признака паритета из слова состояния программы (PSW.0). При приеме девятый бит данных помещается в бит RB8 SCON, а стоп-бит, в отличие от режима 1, теряется. Частота приема/передачи выбирается программой и может быть равна либо  $1/32$ , либо  $1/64$  частоты резонатора в зависимости от управляющего бита SMOD.
- Режим **3**. совпадает с режимом 2, но частота приема/передачи является величиной переменной и задается таймером.



# УАПШ в режимах 1, 2, 3



# Регистр управления/статуса приемопередатчика (SCON)

| Символ | Позиция | Имя и назначение  |     |  |                                |   |   |  |   |   |  |   |   |   |   |   |  |
|--------|---------|---|-----|--|--------------------------------|---|---|--|---|---|--|---|---|---|---|---|--|
| SM0    | SCON.7  | Биты управления режимом работы приемопередатчика. Устанавливаются/сбрасываются программно см. примечание 1  |     |  |                                |   |   |  |   |   |  |   |   |   |   |   |  |
| SM1    | SCON.6  | <table border="1"> <thead> <tr> <th>SM0</th> <th>SM1</th> <th>Режим работы приемопередатчика</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Сдвигающий регистр расширения ввода/вывода</td> </tr> <tr> <td>0</td> <td>1</td> <td>8 битовый приемопередатчик, изменяемая скорость передачи</td> </tr> <tr> <td>1</td> <td>0</td> <td>9 битовый приемопередатчик. Фиксированная скорость передачи</td> </tr> <tr> <td>1</td> <td>1</td> <td>9 битовый приемопередатчик, изменяемая скорость передачи</td> </tr> </tbody> </table> | SM0 | SM1  | Режим работы приемопередатчика | 0 | 0 | Сдвигающий регистр расширения ввода/вывода | 0 | 1 | 8 битовый приемопередатчик, изменяемая скорость передачи | 1 | 0 | 9 битовый приемопередатчик. Фиксированная скорость передачи | 1 | 1 | 9 битовый приемопередатчик, изменяемая скорость передачи |
|        |         | SM0   | SM1 | Режим работы приемопередатчика                           |                                |   |   |  |   |   |  |   |   |   |   |   |  |
|        |         | 0   | 0   | Сдвигающий регистр расширения ввода/вывода               |                                |   |   |  |   |   |  |   |   |   |   |   |  |
|        |         | 0   | 1   | 8 битовый приемопередатчик, изменяемая скорость передачи |                                |   |   |  |   |   |  |   |   |   |   |   |  |
| 1      | 0       | 9 битовый приемопередатчик. Фиксированная скорость передачи   |     |  |                                |   |   |  |   |   |  |   |   |   |   |   |  |
| 1      | 1       | 9 битовый приемопередатчик, изменяемая скорость передачи  |     |  |                                |   |   |  |   |   |  |   |   |   |   |   |  |
| SM2    | SCON.5  | Бит управления режимом приемопередатчика. Устанавливается программно для запрета приема сообщения, в котором девятый бит имеет значение 0   |     |  |                                |   |   |  |   |   |  |   |   |   |   |   |  |
| REN    | SCON.4  | Бит разрешения приема. Устанавливается/сбрасывается программно для разрешения/запрета приема последовательных данных  |     |  |                                |   |   |  |   |   |  |   |   |   |   |   |  |
| TB8    | SCON. 3 | Передача бита 8. Устанавливается/сбрасывается программно для задания девятого передаваемого бита в режиме 9-битового передатчика  |     |  |                                |   |   |  |   |   |  |   |   |   |   |   |  |
| RB8    | SCON.2  | Прием бита 8. Устанавливается/сбрасывается аппаратно для фиксации девятого принимаемого бита в режиме 9-битового приемника  |     |  |                                |   |   |  |   |   |  |   |   |   |   |   |  |
| TI     | SCON. 1 | Флаг прерывания передатчика. Устанавливается аппаратно при окончании передачи байта. Сбрасывается программно после обслуживания прерывания  |     |  |                                |   |   |  |   |   |  |   |   |   |   |   |  |
| RI     | SCON.0  | Флаг прерывания приемника. Устанавливается аппаратно при приеме байта. Сбрасывается программно после обслуживания прерывания  |     |  |                                |   |   |  |   |   |  |   |   |   |   |   |  |

# Скорость приема/передачи информации через последовательный порт.

частота работы приемопередатчика в режиме 0:  
зависит только от резонансной частоты кварцевого резонатора  $f_{PE3}$ :

$$f = f_{PE3}/12$$

за машинный цикл последовательный порт передает один бит информации;  
в режимах 1, 2 и 3 скорость приема/передачи зависит  
от значения управляющего бита **SMOD** в регистре специальных функций **PCON**  
в режиме 2 частота передачи определяется выражением

$$f = 2SMODf_{PE3}/64$$

т.е., при  $SMOD = 0$  частота передачи равна  $1/64$  частоты  $f_{PE3}$ ,  
а при  $SMOD = 1$  —  $1/32$  частоты  $f_{PE3}$ .

в режимах 1 и 3 частота передачи, кроме бита **SMOD**, зависит от таймера 1;  
при этом частота передачи  $f$  зависит от частоты переполнения  $f_{OVT1}$ :

$$f = 2SMODf_{OVT1}/32$$

Прерывание от таймера 1 в этом случае должно быть запрещено.

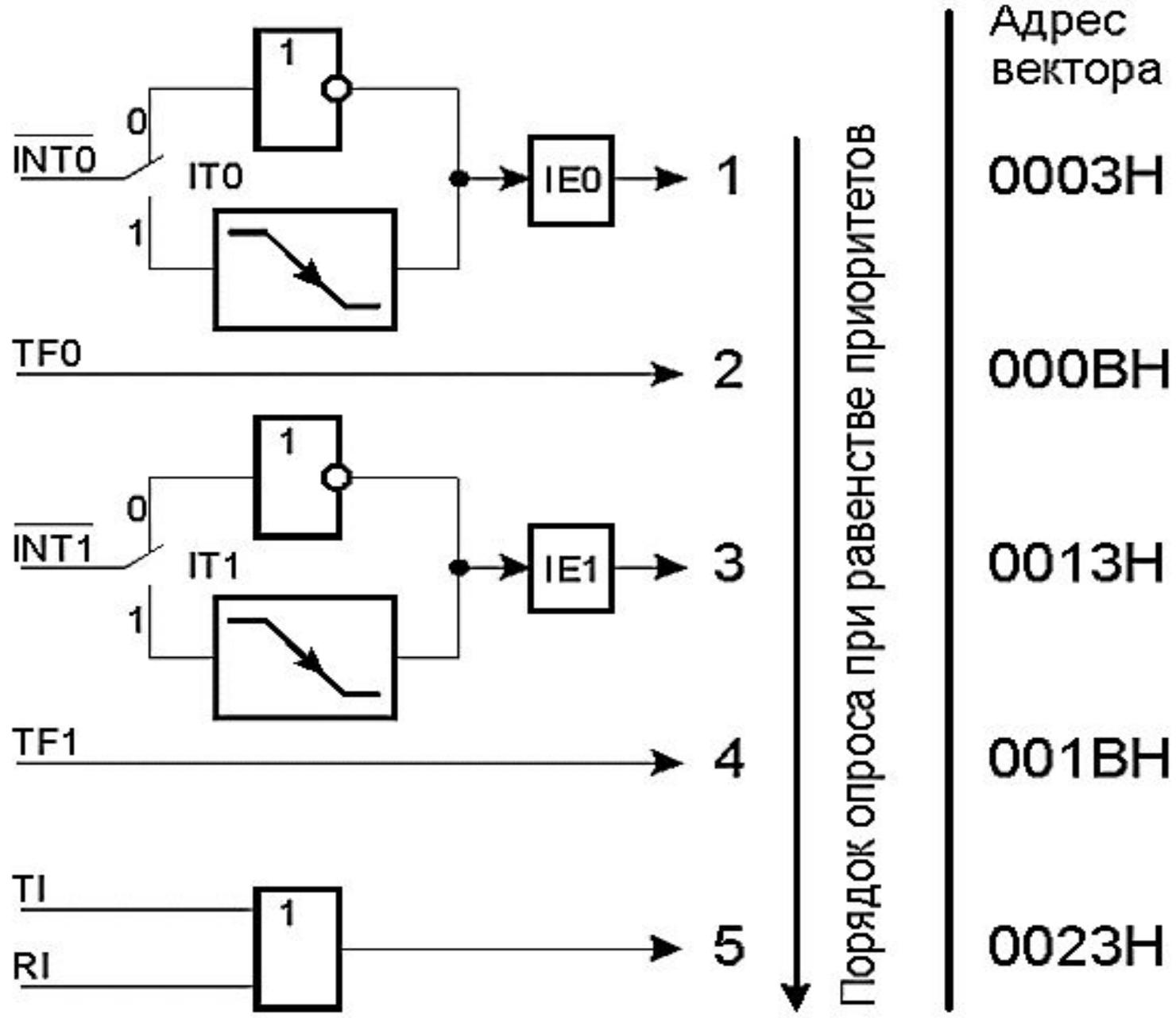
# Настройка таймера 1 для управления частотой работы приемопередатчика

| Частота приема/передачи (BAUD RATE) | Частота резонатора МГц | Таймер/счетчик 1 |     |              |                       |
|-------------------------------------|------------------------|------------------|-----|--------------|-----------------------|
|                                     |                        | SMOD             | C/T | Режим (MODE) | Перезагружаемое число |
| Режим 0, макс: 1 МГц                | 12                     | X                | X   | X            | X                     |
| Режим 2, макс: 375 КГц              | 12                     | 1                | X   | X            | X                     |
| Режим 1, 3: 62,2 КГц                | 12                     | 1                | 0   | 2            | 0FFH                  |
| 19,2 КГц                            | 11,059                 | 1                | 0   | 2            | 0FDH                  |
| 9,6 КГц                             | 11,059                 | 0                | 0   | 2            | 0FDH                  |
| 4,8 КГц                             | 11,059                 | 0                | 0   | 2            | 0FAH                  |
| 2,4 КГц                             | 11,059                 | 0                | 0   | 2            | 0F4H                  |
| 1,2 КГц                             | 11,059                 | 0                | 0   | 2            | 0F4H                  |
| 137,5 Гц                            | 11,059                 | 0                | 0   | 2            | 1DH                   |
| 110 Гц                              | 6                      | 0                | 0   | 2            | 72H                   |
| 110 Гц                              | 12                     | 0                | 0   | 1            | 0FEEBH                |

# Регистр управления мощностью (PCON)

| Символ     | Позиция          | Наименование и функция   |
|------------|------------------|--|
| SMOD       | PCON.7           | Удвоенная скорость передачи. Если бит установлен в 1, то скорость передачи вдвое больше, чем при SMOD = 0. По сбросу SMOD = 0. |
|            | PCON.6           | Не используется  |
|            | PCON.5           | Не используется  |
|            | PCON.4           | Не используется  |
| GF1<br>GF0 | PCON.3<br>PCON.2 | Флаги, специфицируемые пользователем (флаги общего назначения)   |
| PD         | PCON.1           | Бит пониженной мощности. При установке бита в 1 микро-ЭВМ переходит в режим пониженной потребляемой мощности                   |
| IDL        | PCON.0           | Бит холостого хода. Если бит установлен в 1, то микро-ЭВМ переходит в режим холостого хода                                     |

# Система прерываний mc51



# Регистр масок прерывания (IE)

| Символ | Позиция | Имя и назначение   |
|--------|---------|--|
| EA     | IE.7    | Снятие блокировки прерывания. Сбрасывается, программно для запрета всех прерываний независимо от состояний IE.4 - IE.0             |
|        | IE.6    | Не используется  |
|        | IE.5    | Не используется  |
| ES     | IE.4    | Бит разрешения прерывания, от приемопередатчика Установка/сброс программой для разрешения/запрета прерываний от флагов TI или RI . |
| ET1    | IE.3    | Бит разрешения прерывания от таймера. Установка/сброс программой для разрешения/запрета прерываний от таймера 1                    |
| EX1    | IE.2    | Бит разрешения внешнего прерывания 1. Установка/сброс программой для разрешения/запрета прерывания 1                               |
| ET0    | IE.1    | Бит разрешения прерывания от таймера 0. Установка/сброс программой для разрешения/запрета прерываний от таймера 0 .                |
| EX0    | IE.0    | Бит разрешения внешнего прерывания 0. Установка/сброс программой для разрешения/запрета прерывания 0                               |

# Регистр приоритетов прерываний (IP)

| Символ | Позиция        | Имя и назначение   |
|--------|----------------|--|
| -      | IP.7 -<br>IP.5 | Не используется  |
| PS     | IP.4           | Бит приоритета приемопередатчика. Установка/сброс программой для присваивания прерыванию от приемопередатчика высшего/низшего приоритета |
| PT1    | IP.3           | Бит приоритета таймера 1. Установка/сброс программой для присваивания прерыванию от таймера 1 высшего/низшего приоритета                 |
| PX1    | IP.2           | Бит приоритета внешнего прерывания 1. Установка/сброс программой для присваивания высшего/низшего приоритета внешнему прерыванию INT1    |
| PT0    | IP.1           | Бит приоритета таймера 0. Установка/сброс программой для присваивания прерыванию от таймера 0 высшего/низшего приоритета                 |
| PX0    | IP.0           | Бит приоритета внешнего прерывания 0. Установка/сброс программой для присваивания высшего/низшего приоритета внешнему прерыванию INT0    |

# Система команд mc51

111 базовых команд,

по функциональному признаку подразделяются на группы команд:

- ✓ пересылки данных;
- ✓ арифметических операций;
- ✓ логических операций;
- ✓ операций над битами;
- ✓ передачи управления.

# Режимы адресации

## **Прямая адресация (Direct Addressing)**

операнд определяется 8-битным адресом в инструкции, используется только для внутренней памяти данных и регистров SFR.

## **Косвенная адресация (Indirect Addressing)**

инструкция адресует регистр, содержащий адрес операнда, для указания 8-битных адресов используются регистры R0 и R1 или указатель стека SP.

## **Регистровая адресация (Register Instruction)**

для доступа к регистрам R0+R7, в байте кода операции трехбитовое поле, определяющее номер регистра.

## **Непосредственная адресация (*Immediate constants*)**

операнд содержится в поле команды вслед за кодом операции – один или два байта (data8, data16).

## **Индексная адресация (Indexed Addressing)**

используется при обращении к памяти программ и только при чтении, осуществляется просмотр таблиц в памяти программ, базовый адрес требуемой таблицы в DPTR или PC, а аккумулятор указывает на точку входа в нее;

адрес элемента таблицы находится сложением базы с индексом (содержимым аккумулятора).

Другой тип индексной адресации применяется в командах "перехода по выбору" (Case Jump),

адрес перехода вычисляется как сумма указателя базы и аккумулятора.

## **Неявная адресация (Register-Specific Instructions)**

# Типы команд

|    | $D_7 \dots D_0$        | $D_7 \dots D_0$ |                        |
|----|------------------------|-----------------|------------------------|
| 1  | КОП                    |                 |                        |
| 2  | КОП                    | # d             |                        |
| 3  | КОП                    | ad              |                        |
| 4  | КОП                    | bit             |                        |
| 5  | КОП                    | rel             |                        |
| 6  | $a_{10} a_9 a_8$   КОП | $a_7 \dots a_0$ |                        |
| 7  | КОП                    | ad              | $D_7 \dots D_0$<br># d |
| 8  | КОП                    | ad              | rel                    |
| 9  | КОП                    | ads             | add                    |
| 10 | КОП                    | # d             | rel                    |
| 11 | КОП                    | bit             | rel                    |
| 12 | КОП                    | ad16h           | ad16l                  |
| 13 | КОП                    | # d16h          | # d16l                 |

# PCA (Programmable Counter Array)

имеется у моделей 80C51FA, 80C51FB, 80C51 PC и 80C51GB

обеспечивает большие "временные" возможности по сравнению с обычными счетчиками-таймерами, которые имеются на кристалле, при этом достигается большая точность с меньшими затратами ресурсов.

PCA рекомендуется использовать для измерения ширины импульса, частоты, скважности, И для формирования прямоугольных импульсов на внешних выводах микроконтроллера.

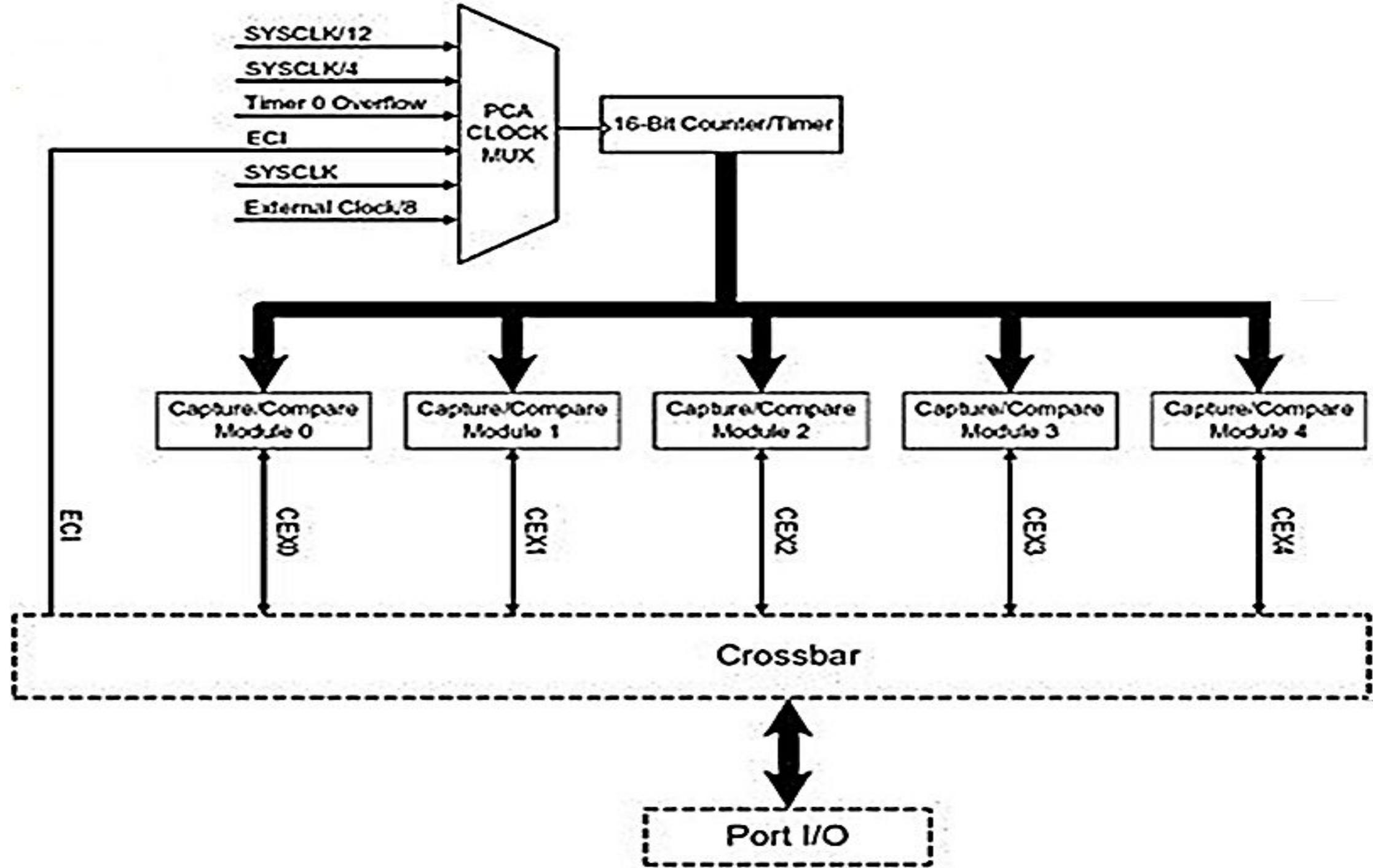
PCA состоит из 16-разрядного таймера-счетчика с времязадающим узлом (TimeBase) и пятью модулями захвата/сравнения ССМ (Capture/Compare Module).

TimeBase может подавать на вход PCA один из четырех тактовых сигналов:

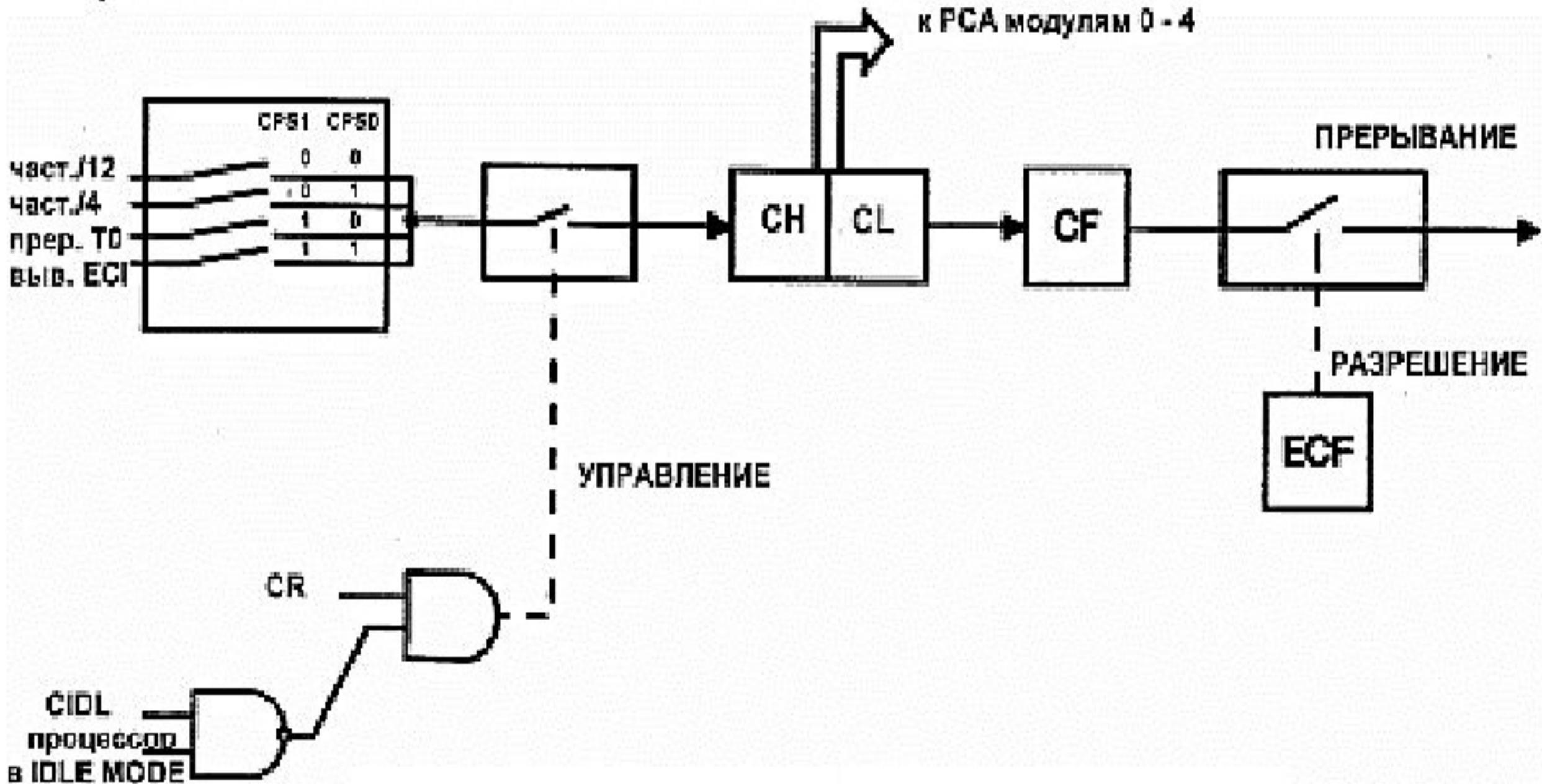
- системную частоту тактирования, деленную на 12 или 4,
- сигнал с выхода переполнения таймера 0 или
- сигнал с внешнего счетного входа ECI (External Clock Input).

Каждый из пяти ССМ может быть запрограммирован на выполнение одной из четырех функций:

- переключаемая по фронту защелка,
  - программный счетчик,
  - высокоскоростной выход и
- широто-импульсный модулятор PWM (Pulse Width Modulator).



# Структурная схема РСА таймера-счетчика



# Регистр режимов PCA таймера-счетчика (CMOD)

Адрес=0D9H

Значение после сброса 00XX X000B адресация к отдельным битам невозможна

| Символ                                     | Позиция | Имя и назначение  |   |  |  |      |      |   |   |   |                     |   |   |                    |   |   |                           |   |   |                                     |
|--|---------|---|---|--|--|------|------|---|---|---|---------------------|---|---|--------------------|---|---|---------------------------|---|---|-------------------------------------|
| CIDL                                       | CMOD.7  | CIDL=0 разрешает работу PCA в "режиме холостого хода" (Idle Mode), CIDL=1 запрещает работу PCA в режиме "холостого хода".   |   |  |  |      |      |   |   |   |                     |   |   |                    |   |   |                           |   |   |                                     |
| WDTE                                       | CMOD.6  | WDTE=0 запрещает режим сторожевого таймера (watchdog timer) четвертому модулю сравнения-захвата, WDTE=1 разрешает этот режим.   |   |  |  |      |      |   |   |   |                     |   |   |                    |   |   |                           |   |   |                                     |
| -  | CMOD.5  | -   |   |  |  |      |      |   |   |   |                     |   |   |                    |   |   |                           |   |   |                                     |
| -  | CMOD.4  | -   |   |  |  |      |      |   |   |   |                     |   |   |                    |   |   |                           |   |   |                                     |
| -  | CMOD.3  | -   |   |  |  |      |      |   |   |   |                     |   |   |                    |   |   |                           |   |   |                                     |
| CPS1                                       | CMOD.2  | <table border="1"> <thead> <tr> <th colspan="3">Источник сигналов для таймера-счетчика PCA</th> </tr> <tr> <th>CPS1</th> <th>CPS0</th> <th>сигнал на счетном входе PCA таймерасчетчика</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>тактовая частота/12</td> </tr> <tr> <td>0</td> <td>1</td> <td>тактовая частота/4</td> </tr> <tr> <td>1</td> <td>0</td> <td>переполнение от таймера 0</td> </tr> <tr> <td>1</td> <td>1</td> <td>внешний сигнал на контакте EC1/P1.2</td> </tr> </tbody> </table> | Источник сигналов для таймера-счетчика PCA  |  |  | CPS1 | CPS0 | сигнал на счетном входе PCA таймерасчетчика | 0 | 0 | тактовая частота/12 | 0 | 1 | тактовая частота/4 | 1 | 0 | переполнение от таймера 0 | 1 | 1 | внешний сигнал на контакте EC1/P1.2 |
| Источник сигналов для таймера-счетчика PCA |         |   |   |  |  |      |      |   |   |   |                     |   |   |                    |   |   |                           |   |   |                                     |
| CPS1                                       | CPS0    |   | сигнал на счетном входе PCA таймерасчетчика |  |  |      |      |   |   |   |                     |   |   |                    |   |   |                           |   |   |                                     |
| 0  | 0       |   | тактовая частота/12                         |  |  |      |      |   |   |   |                     |   |   |                    |   |   |                           |   |   |                                     |
| 0  | 1       | тактовая частота/4  |   |  |  |      |      |   |   |   |                     |   |   |                    |   |   |                           |   |   |                                     |
| 1  | 0       | переполнение от таймера 0   |   |  |  |      |      |   |   |   |                     |   |   |                    |   |   |                           |   |   |                                     |
| 1  | 1       | внешний сигнал на контакте EC1/P1.2   |   |  |  |      |      |   |   |   |                     |   |   |                    |   |   |                           |   |   |                                     |
| CPS0                                       | CMOD.1  |   |   |  |  |      |      |   |   |   |                     |   |   |                    |   |   |                           |   |   |                                     |
| ECF  | CMOD.0  | ECF=1 разрешает флагу CF в SCON генерировать запрос на прерывание. ECF=0 запрещает данную функцию.  |   |  |  |      |      |   |   |   |                     |   |   |                    |   |   |                           |   |   |                                     |

# Регистр управления PCA таймером-счетчиком (CCON)

Адрес=0D8H

Значение после сброса 00X0 0000В возможна адресация к отдельным битам

| Символ | Позиция | Имя и назначение   |
|--------|---------|--|
| CF     | CCON.7  | Флаг переполнения PCA таймера-счетчика, устанавливается аппаратно при переполнении счетчика. CF генерирует запрос на прерывание только при установленном бите ECP в CMOO. CP можно установить как программно, так и аппаратно, но сбрасывается он только программно. |
| CR     | CCON.6  | Бит запуска PCA таймера-счетчика. При программной установке включает PCA таймер-счетчик. При программном сбросе выключает.   |
| -      | CCON.5  | -  |
| CCF4   | CCON.4  | Флаг прерывания модуля 4. Устанавливается аппаратно, когда происходит соответствующее событие. Сбрасывается программно.  |
| CCF3   | CCON.3  | Флаг прерывания модуля 3. Устанавливается аппаратно, когда происходит соответствующее событие. Сбрасывается программно.  |
| CCF2   | CCON.2  | Флаг прерывания модуля 2. Устанавливается аппаратно, когда происходит соответствующее событие. Сбрасывается программно.  |
| CCF1   | CCON.1  | Флаг прерывания модуля 1. Устанавливается аппаратно, когда происходит соответствующее событие. Сбрасывается программно.  |
| CCFO   | CCON.0  | Флаг прерывания модуля 0. Устанавливается аппаратно, когда происходит соответствующее событие. Сбрасывается программно.  |

# Регистр режимов модуля сравнения захвата (CCAPMn)

Значение после сброса X000

Адреса CCAPMn (n=0..4)

|        |      |
|--------|------|
| CCAPM0 | 0DAH |
| CCAPM1 | 0DBH |
| CCAPM2 | 0DCH |
| CCAPM3 | 0DDH |
| CCAPM4 | 0DEH |

адресация к отдельным битам не возможна

| Символ | Позиция  | Имя и назначение  |
|--------|----------|---|
| -      | CCAPMn.7 | -   |
| ECOMn  | CCAPMn.6 | ECOMn=1 разрешение сравнения.   |
| CAPPn  | CCAPMn.5 | CAPPn=1 разрешение захвата по фронту.   |
| CAPNn  | CCAPMn.4 | CAPNn=1 разрешение захвата по спаду.  |
| MATn   | CCAPMn.3 | Если MATn=1, то при совпадении значений в PCA таймере-счетчике и в регистрах CCAPnN, CCAPnL устанавливается флаг CCFn в регистре CCON.                              |
| TOGn   | CCAPMn.2 | Если TOGn=1, то при совпадении значения PCA в таймере-счетчике и в регистрах CCAPnH, CCAPnL на внешнем контакте CEXn происходит переключение сигнала (0-1 или 1-0). |
| PWMn   | CCAPMn.1 | PWMn=1 разрешает использовать внешний контакт CEXn для генерации импульсов заданной скважности.   |
| ECCFn  | CCAPMn.0 | Разрешение флагу CCFn в регистре CCON генерировать запрос на прерывание.  |

# Режимы работы PCA

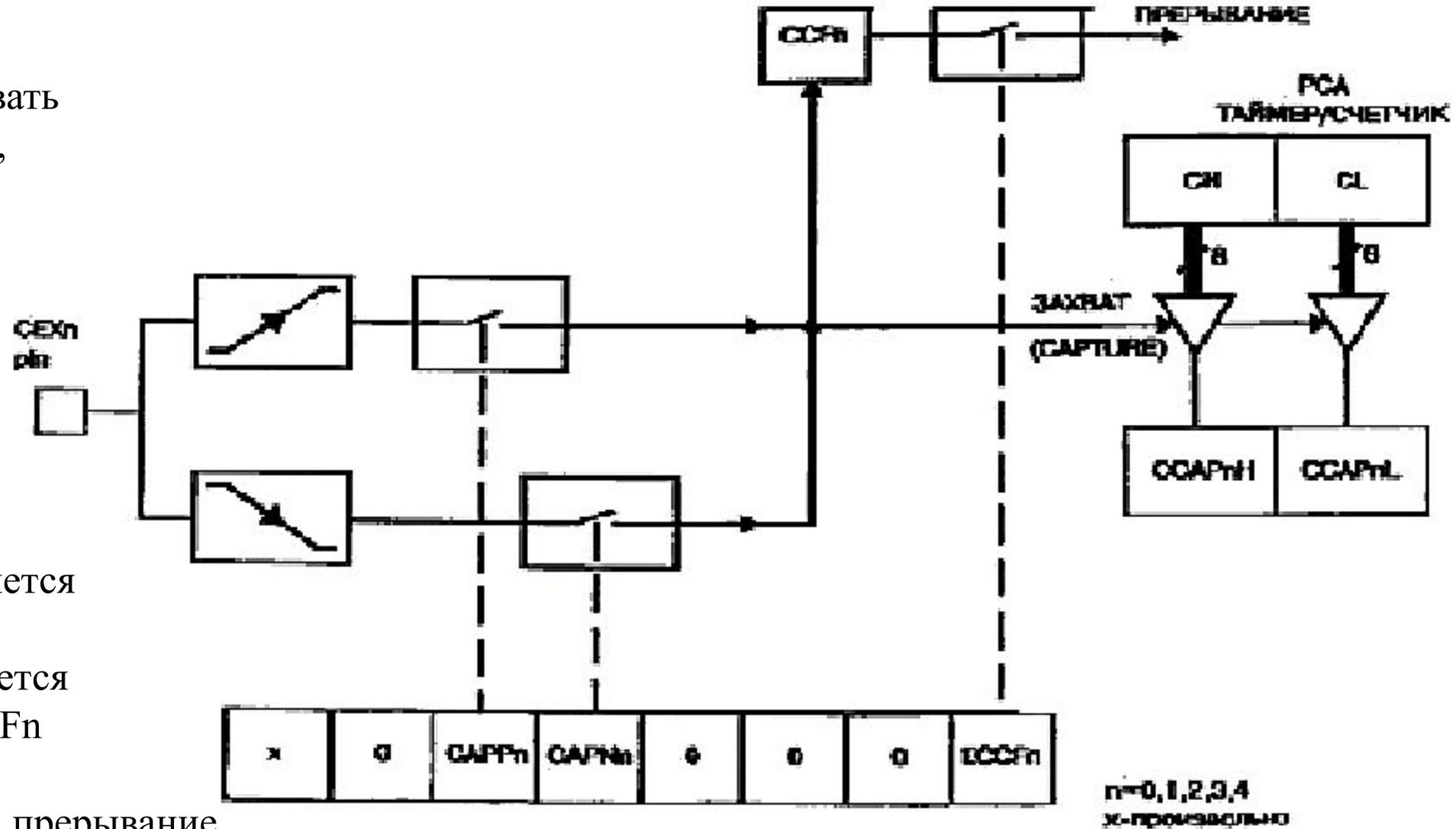
комбинации битов регистра CCPMn,  
соответствующие различным режимам работы модуля сравнения-захвата

|   | ECOMn | CAPPn | CAPNn | MATn | TOGn | PWMn | ESSCF | Функции модуля                      |
|---|-------|-------|-------|------|------|------|-------|-------------------------------------|
| X | 0     | 0     | 0     | 0    | 0    | 0    | 0     | нет операций                        |
| X | X     | 1     | 0     | 0    | 0    | 0    | X     | захват по фронту на CEXn            |
| X | X     | 0     | 1     | 0    | 0    | 0    | X     | захват по спаду на CEXn             |
| X | X     | 1     | 1     | 0    | 0    | 0    | X     | захват по перепаду на CEXn          |
| X | 1     | 0     | 0     | 1    | 0    | 0    | X     | программируемый таймер              |
| X | 1     | 0     | 0     | 1    | 1    | 0    | X     | скоростной вывод                    |
| X | 1     | 0     | 0     | 0    | 0    | 1    | 0     | модулирование длительности импульса |
| X | 1     | 0     | 0     | 1    | X    | 0    | X     | сторожевой таймер                   |

# Режим захвата

рекомендуется использовать при измерении периодов, длительности, скважности импульсов, разности фаз между различными входами; Установка битов  $CA\overline{P}P_n$  и/или  $CA\overline{P}N_n$  в регистре  $CCAPM_n$  определяет перепад (0-1 или 1-0), по которому осуществляется захват.

При захвате устанавливается флаг события модуля  $CCF_n$  в регистре  $CCON$ , и генерируется запрос на прерывание, если установлен бит  $ECCF_n$  в регистре  $CCAPM_n$ .





## Режим скоростного вывода

формируется сигнал на внешнем выводе СЕХ<sub>n</sub>,  
когда происходит совпадение РСА таймера со значением,  
предварительно загруженным в регистры ССАР<sub>nH</sub>, ССАР<sub>nL</sub>;  
режим скоростного вывода является более точным по сравнению с переключением  
контактов параллельного порта программным путем,  
поскольку формирование сигнала на внешнем выводе происходит до обращения к  
подпрограмме обработки прерывания.

Это значит, что задержка, связанная с выполнением подпрограммы обработки прерывания,  
не будет влиять на формирование сигнала.

Если подпрограмма обработки прерывания не изменит содержимого ССАР<sub>nH</sub>, ССАР<sub>nL</sub>,  
то следующий сигнал "совпадение" будет сформирован после того  
как значение РСА таймера-счетчика снова совпадет со старым сравниваемым значением

# Режим сторожевого таймера (watchdog timer)

Сторожевой таймер - это схема, которая автоматически сбрасывает микроконтроллер, если не получает от управляемой системы сигнала подтверждения, что не произошло никакого сбоя;

это используется в системах, где есть электрические помехи или сбои по питанию; в режиме сторожевого таймера может работать только четвертый модуль.

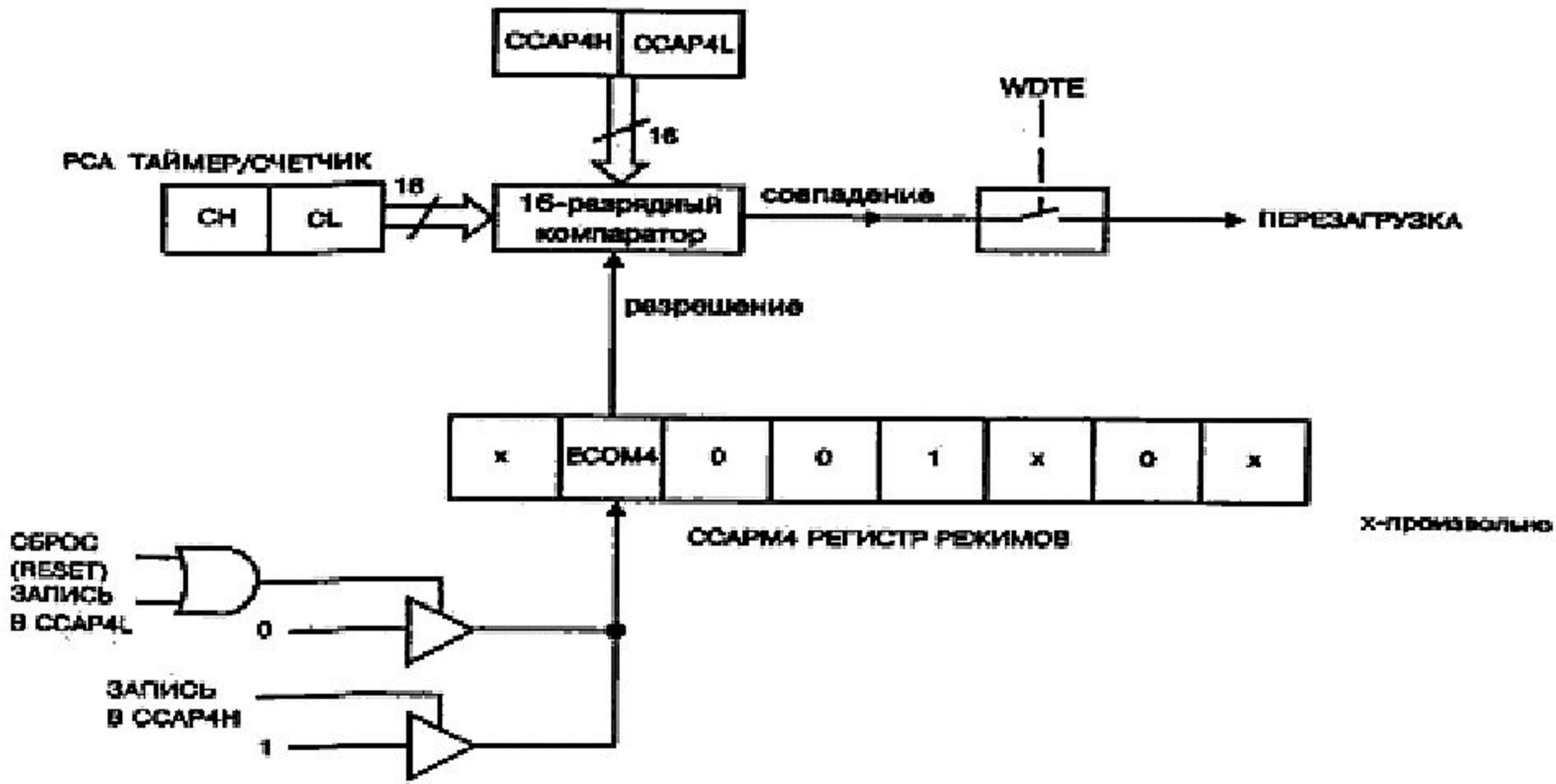
Сигнал сброса срабатывает всякий раз, когда происходит совпадение значения PCA таймера-счетчика со значением CCAP4H, CCAP4L

этот режим устанавливает бит WDTE в регистре CMOD; при этом четвертый модуль должен находиться в режиме программируемого таймера или скоростного вывода;

Предотвратить сброс можно тремя способами:

1. периодически изменять значение в регистрах CCAP4H, CCAP4L, так что бы оно никогда не совпало с PCA таймером-счетчиком;
2. периодически изменять регистры CH, CL PCA таймера-счетчика, чтобы они никогда не совпали со значением в регистрах CCAP4H, CCAP4L;
3. выключить режим охранного таймера путем сброса бита WDTE перед тем, как произойдет совпадение и затем снова включить его.

# Сторожевой таймер





# Аналого-цифровой преобразователь

АЦП микроконтроллера в семействе MCS-51/52  
(например, типа SAB 80515 фирмы Siemens или 80C51GB)

- обеспечивает 8 битное преобразование
- имеет восемь мультиплексных каналов аналогового входного сигнала "на чипе"
  - имеет схему выборки-хранения
  - возможность программирования опорных напряжений, что позволяет увеличивать точность преобразования, сужая пределы измерения.

Преобразование осуществляется методом последовательного приближения с использованием конденсаторной цепи.

Длительность цикла преобразования от 15 до 29 машинных циклов.

Для использования встроенного АЦП есть специальные функциональные регистры:

- ADCON- регистр управления аналого-цифрового преобразователя,
- ADDAT- регистр данных аналого-цифрового преобразователя,
- DAPR- регистр программирования опорных напряжений.

# ADCON

(регистр управления преобразователем)

адрес - 0D8H, возможна побитовая адресация

|      |      |      |      |      |      |      |      |       |
|------|------|------|------|------|------|------|------|-------|
| BD   | CLK  | -    | BSY  | ADM  | MX2  | MX1  | MX0  | Бит   |
| 0DFH | 0DEH | 0D0H | 0DCH | 0DBH | 0DAH | 0D9H | 0D8H | Адрес |

используется, чтобы

- выбрать один из восьми каналов аналогового входного сигнала, которые будут преобразованы,
- определять однократное или циклическое преобразование,
- проверять бит состояния BSY, который сообщает, происходит ли преобразование или нет.

| Символ | Позиция | Функция  |                             |                             |           |                 |           |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |
|--------|---------|--|-----------------------------|-----------------------------|-----------|-----------------|-----------|---|---|---|-----------------------------|-----|---|---|---|-----------------------------|-----|---|---|---|-----------------------------|-----|---|---|---|-----------------------------|-----|---|---|---|-----------------------------|-----|---|---|---|-----------------------------|-----|---|---|---|-----------------------------|-----|---|---|---|-----------------------------|-----|
| MX0    | ADCON.0 | <p>Выбор канала аналогового входного сигнала, см. таблицу.</p> <table border="1"> <thead> <tr> <th>MX2</th> <th>MX1</th> <th>MX0</th> <th>Выбранный Канал</th> <th>Вывод ИМС</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Аналоговый входной сигнал 0</td> <td>AN0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Аналоговый входной сигнал 1</td> <td>AN1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Аналоговый входной сигнал 2</td> <td>AN2</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Аналоговый входной сигнал 3</td> <td>AN3</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Аналоговый входной сигнал 4</td> <td>AN4</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Аналоговый входной сигнал 5</td> <td>AN5</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Аналоговый входной сигнал 6</td> <td>AN6</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Аналоговый входной сигнал 7</td> <td>AN7</td> </tr> </tbody> </table> | MX2                         | MX1                         | MX0       | Выбранный Канал | Вывод ИМС | 0 | 0 | 0 | Аналоговый входной сигнал 0 | AN0 | 0 | 0 | 1 | Аналоговый входной сигнал 1 | AN1 | 0 | 1 | 0 | Аналоговый входной сигнал 2 | AN2 | 0 | 1 | 1 | Аналоговый входной сигнал 3 | AN3 | 1 | 0 | 0 | Аналоговый входной сигнал 4 | AN4 | 1 | 0 | 1 | Аналоговый входной сигнал 5 | AN5 | 1 | 1 | 0 | Аналоговый входной сигнал 6 | AN6 | 1 | 1 | 1 | Аналоговый входной сигнал 7 | AN7 |
| MX2    | MX1     |  | MX0                         | Выбранный Канал             | Вывод ИМС |                 |           |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |
| 0      | 0       |  | 0                           | Аналоговый входной сигнал 0 | AN0       |                 |           |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |
| 0      | 0       |  | 1                           | Аналоговый входной сигнал 1 | AN1       |                 |           |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |
| 0      | 1       |  | 0                           | Аналоговый входной сигнал 2 | AN2       |                 |           |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |
| 0      | 1       |  | 1                           | Аналоговый входной сигнал 3 | AN3       |                 |           |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |
| 1      | 0       |  | 0                           | Аналоговый входной сигнал 4 | AN4       |                 |           |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |
| 1      | 0       |  | 1                           | Аналоговый входной сигнал 5 | AN5       |                 |           |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |
| 1      | 1       | 0  | Аналоговый входной сигнал 6 | AN6                         |           |                 |           |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |
| 1      | 1       | 1  | Аналоговый входной сигнал 7 | AN7                         |           |                 |           |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |
| MX1    | ADCON.1 |  |                             |                             |           |                 |           |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |
| MX2    | ADCON.2 |  |                             |                             |           |                 |           |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |
| ADM    | ADCON.3 | Режим аналого-цифрового преобразования. Если ADM = 1 - непрерывное преобразование. Если ADM = 0, преобразователь останавливается после одного преобразования.  |                             |                             |           |                 |           |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |
| BSY    | ADCON.4 | Флаг занятости. Этот флаг указывает, происходит преобразование (BSY = 1) или нет (BSY = 0).  |                             |                             |           |                 |           |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |
| -      | ADCON.5 | Зарезервирован (должен быть 0).  |                             |                             |           |                 |           |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |
| CLK    | ADCON.6 | (Используется не для управления АЦП) Включение синхронизации системы. Если равно 1, синхросигнал с $t/12$ частотой генератора подается на вывод P1. 6/CLKOUT. CLK = 0 блокирует синхронизирующий вывод.  |                             |                             |           |                 |           |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |
| BD     | ADCON.7 | (Используется не для управления АЦП) Включение режима передачи со скоростью в бодах. Если равно 1, то осуществляется прием в режиме 1 и 3 последовательного порта из внутреннего генератора скоростей в бодах.   |                             |                             |           |                 |           |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |   |   |   |                             |     |

# DAPR

(регистр программирования опорных напряжений АЦП).

Регистр DAPR позволяет менять внутренние опорные напряжения IVAREF и IVAGND.

Они могут программироваться с шагом в 1/16

относительно внешних опорных напряжений (VAREF-VAGND).

Биты с 0 по 3 регистра DAPR определяют IVAGND, биты с 4 по 7 определяют IVAREF.

Для нормальной работы АЦП требуется как минимум разность в 1 вольт между внутренними опорными напряжениями.

Поэтому, внутреннее опорное напряжение IVAREF должно всегда программироваться на четыре шага выше, чем IVAGND

Значения IVAGND и IVAREF определяются по формуле:

$$IVAGND = VAGND + DAPR(0 - 3) / 16 (VAREF - VAGND)$$

причем  $DAPR(0 - 3) < 0$  и  $DAPR(0 - 3) < 13$ ;

$$IVAREF = VAGND + DAPR(4 - 7) / 16 (VAREF - VAGND)$$

где  $DAPR(0 - 3)$  - содержание младшего полубайта,

$DAPR(4 - 7)$  - содержание старшего полубайта DAPR, принимаемого как десятичное целое число без знака.

Если  $DAPR(0 - 3)$  или  $DAPR(4 - 7) = 0$ ,

внутренние опорные напряжения соответствуют внешним опорным напряжениям

Если  $VAINPUT > IVAREF$ , то результат преобразования будет равен 0FFH, если  $VAINPUT < IVAGND$ ,

то результат преобразования - 00H ( $VAINPUT$  - напряжение аналогового входного сигнала).

# Регистр программирования АЦП DAPR (0DAH)

|                              |   |   |   |                              |   |   |   |
|------------------------------|---|---|---|------------------------------|---|---|---|
| 7                            | 6 | 5 | 4 | 3                            | 2 | 1 | 0 |
| Цифровое Значение для IVAREF |   |   |   | Цифровое Значение для IVAGND |   |   |   |

Если используются внешние опорные напряжения  $V_{AGND} = 0\text{ V}$  и  $V_{AREF} = +5\text{ V}$  (относительно GND и VCC), то при помощи регистра DAPR могут быть установлены следующие значения внутренних опорных напряжений IVAGND и IVAREF:

| Ступень | DAPR (0-3) | DAPR (4-7) | IVAGND (V) | IVAREF (V) |
|---------|------------|------------|------------|------------|
| 0       | 0000       | 0000       | 0.0        | 5.0        |
| 1       | 0001       | 0001       | 0.3125     | -          |
| 2       | 0010       | 0010       | 0.625      | -          |
| 3       | 0011       | 0011       | 0.9375     | -          |
| 4       | 0100       | 0100       | 1.25       | 1.25       |
| 5       | 0101       | 0101       | 1.5625     | 1.5625     |
| 6       | 0110       | 0110       | 1.875      | 1.875      |
| 7       | 0111       | 0111       | 2.1875     | 2.1875     |
| 8       | 1000       | 1000       | 2.5        | 2.5        |
| 9       | 1001       | 1001       | 2.8125     | 2.8125     |
| 10      | 1010       | 1010       | 3.125      | 3.125      |
| 11      | 1011       | 1011       | 3.4375     | 3.4375     |
| 12      | 1100       | 1100       | 3.75       | 3.75       |
| 13      | 1101       | 1101       | -          | 4.0625     |
| 14      | 1110       | 1110       | -          | 4.375      |
| 15      | 1111       | 1111       | -          | 4.6875     |

