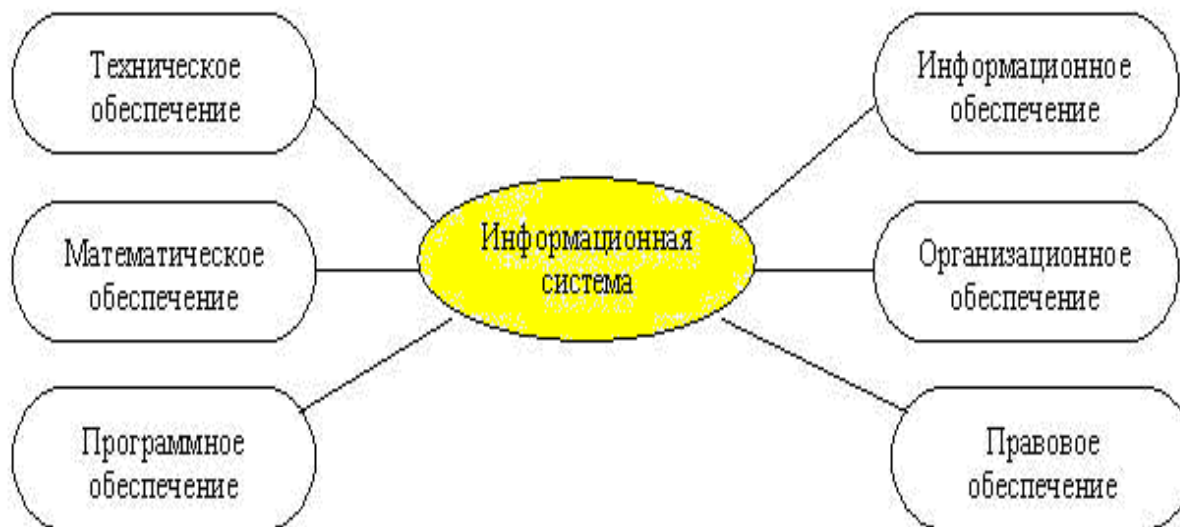


## СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

1. **Общая характеристика информационных систем и технических средств. Преобразование информации.**
2. **Технические средства обработки информации. 8-разрядный процессор.**
3. **Запоминающие устройства информационных систем**
4. **Архитектура 16-разрядных процессоров**
5. **Архитектура 32-разрядных процессоров**
6. **Архитектура 64-разрядных и многоядерных процессоров**
7. **Архитектура персональных компьютеров**
8. **Программно-технические средства ввода информации в компьютер**
9. **Программно-технические средства вывода информации**

## Структура информационной системы как совокупность обеспечивающих подсистем

**Информационная система** — это взаимосвязанная совокупность информационных, технических, программных, математических, организационных, правовых, эргономических, лингвистических, технологических и других средств, а также персонала, предназначенная для сбора, обработки, хранения и выдачи информации и принятия управленческих решений.

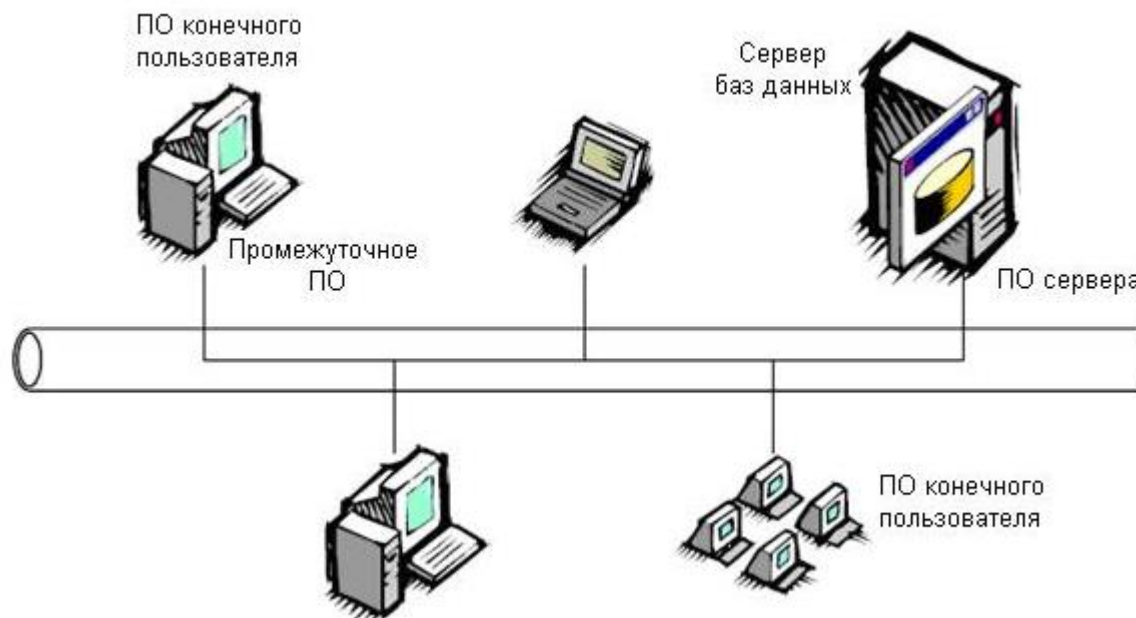


Информационная система предназначена для своевременного обеспечения определенных людей необходимой информацией, то есть для удовлетворения конкретных информационных потребностей в рамках определенной предметной области.

## Классификация информационных систем



## Клиент-серверная двухуровневая архитектура



Достоинства этой архитектуры:

1. Полная поддержка многопользовательской работы;
2. Простота обеспечения целостности данных.

## Технические средства преобразования аналоговых процессов в цифровые

В процессе преобразования напряжения в цифровой код осуществляются три независимые операции:

- дискретизация по времени;
- квантование по уровню;
- кодирование.

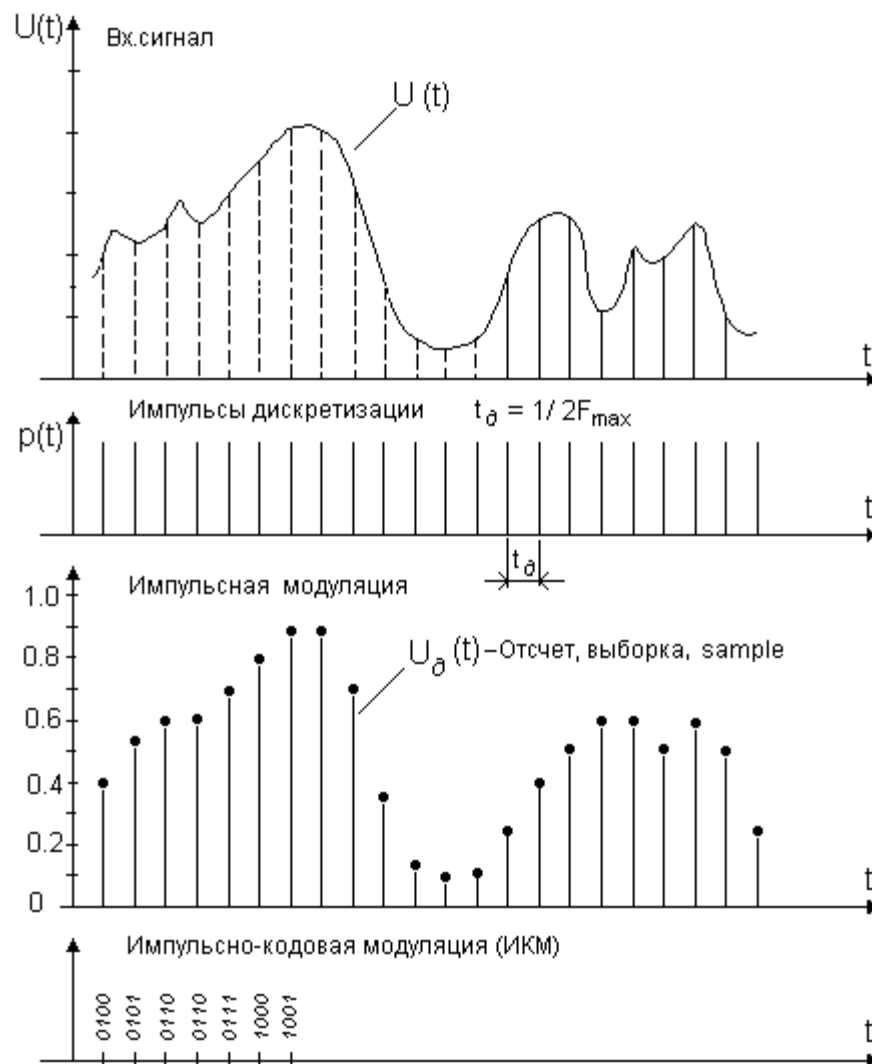
Период дискретизации

$$\Delta t_d = 1/(2F_m)$$

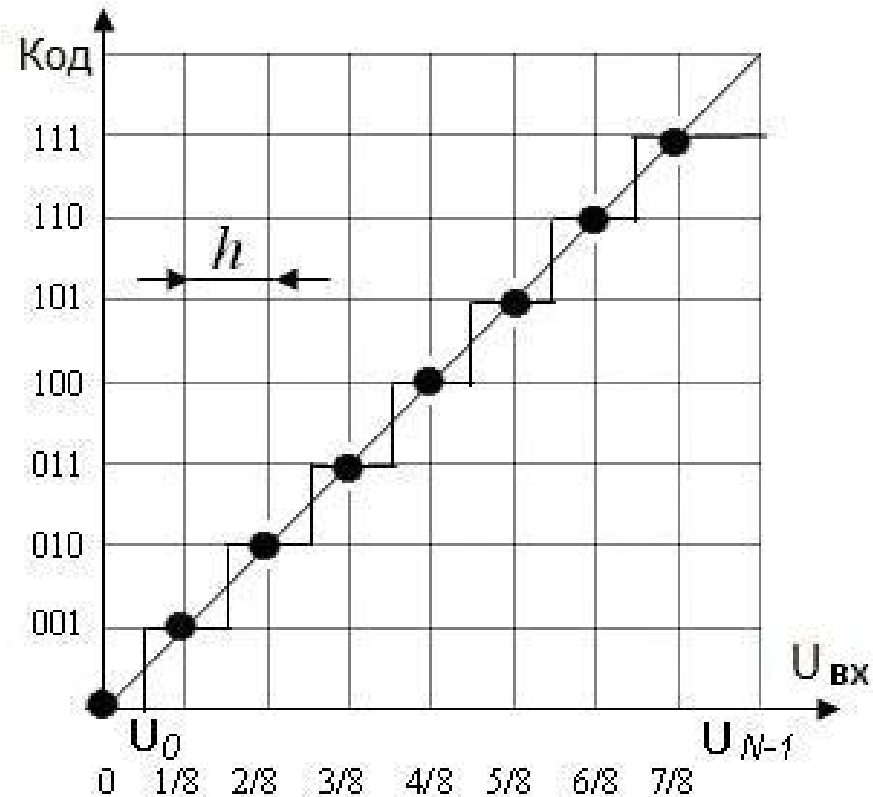
$$U_\delta(t) = U(t)P(t)$$

$$P(t) = \sum \delta(t - nT_o)$$

## Технические средства преобразования аналоговых процессов в цифровые



## Статическая характеристика преобразования АЦП



$h = U_{ВХ.макс} / (2^N - 1)$  - шаг квантования;

$DR = 20 \lg(U_{макс} / \Delta U) = 20 \lg(2^N)$  динамический диапазон.

## Основные параметры АЦП

**Шаг квантования  $h$**  — разность между двумя соседними значениями квантованной величины:  $h = U_{\text{вх.макс}} / (2^N - 1)$ .

**Количество разрядов.**

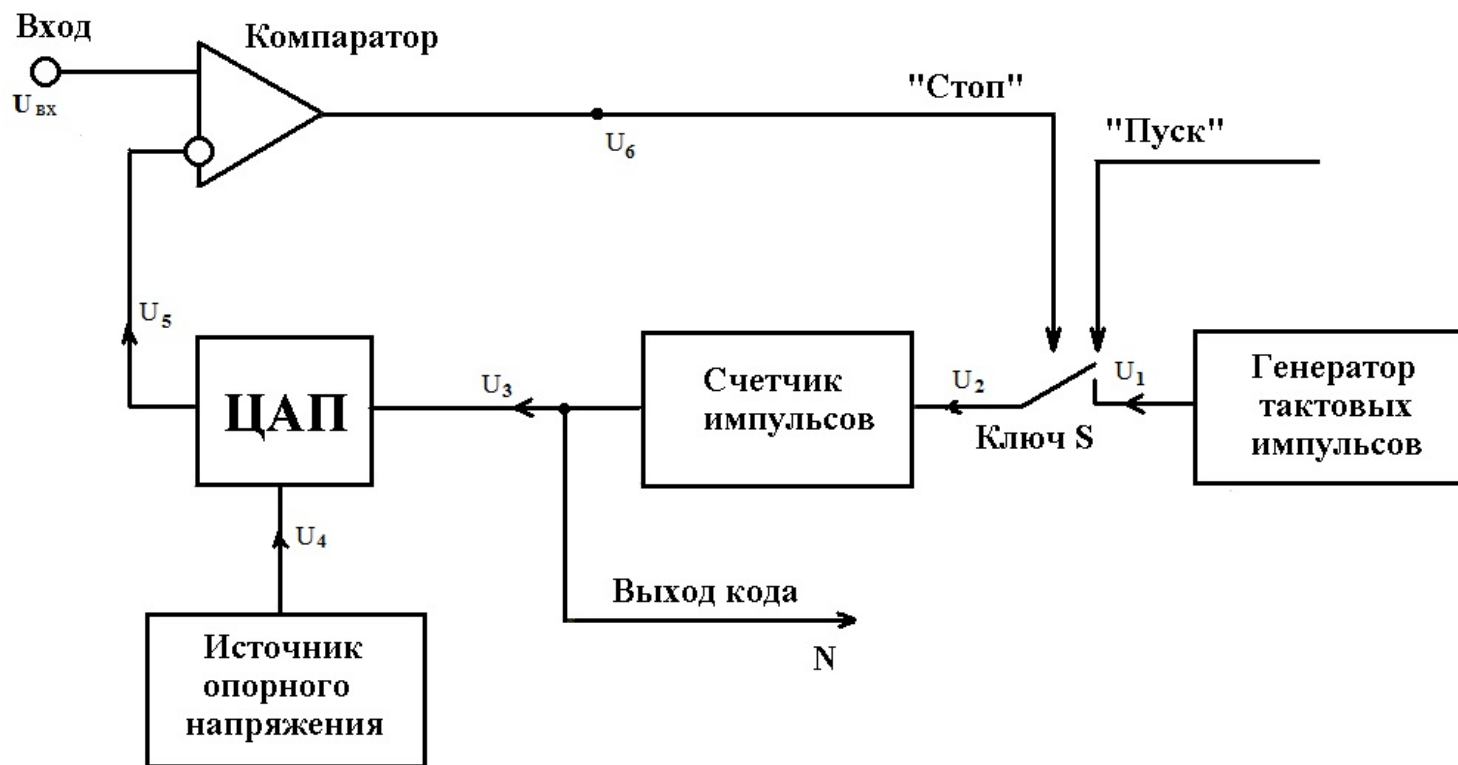
**Время преобразования  $t_{\text{пр}}$**  — это время от начала преобразования до появления на выходе устойчивого кода, соответствующего данной выборке.

**Динамический диапазон АЦП (DR — *Dynamic Range*)** называется отношение максимального воспринимаемого уровня входного напряжения к минимальному, выраженное в дБ:  $DR = 20 \lg(U_{\text{max}} / \Delta U) = 20 \lg(2^N)$ .

**Погрешность смещения нуля** — значение  $U_{\text{вх}}$ , при котором выходной код АЦП равен нулю.



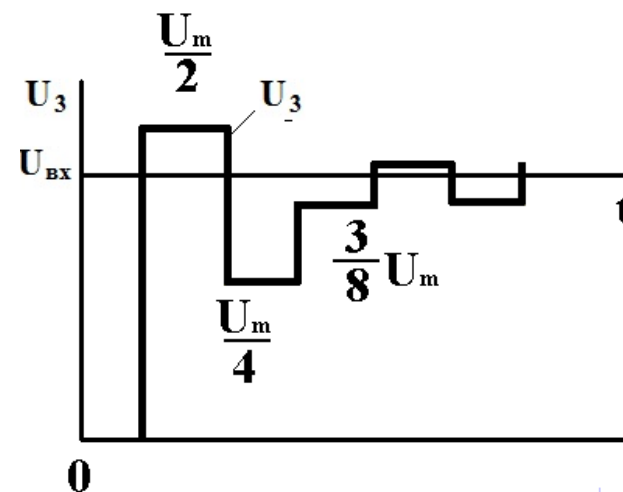
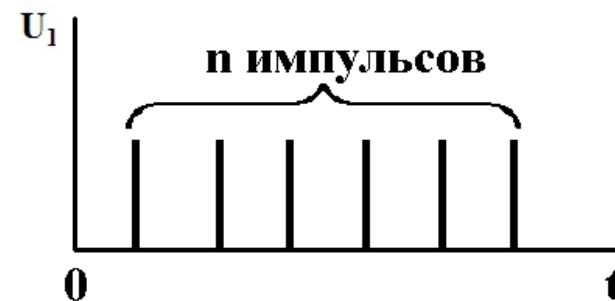
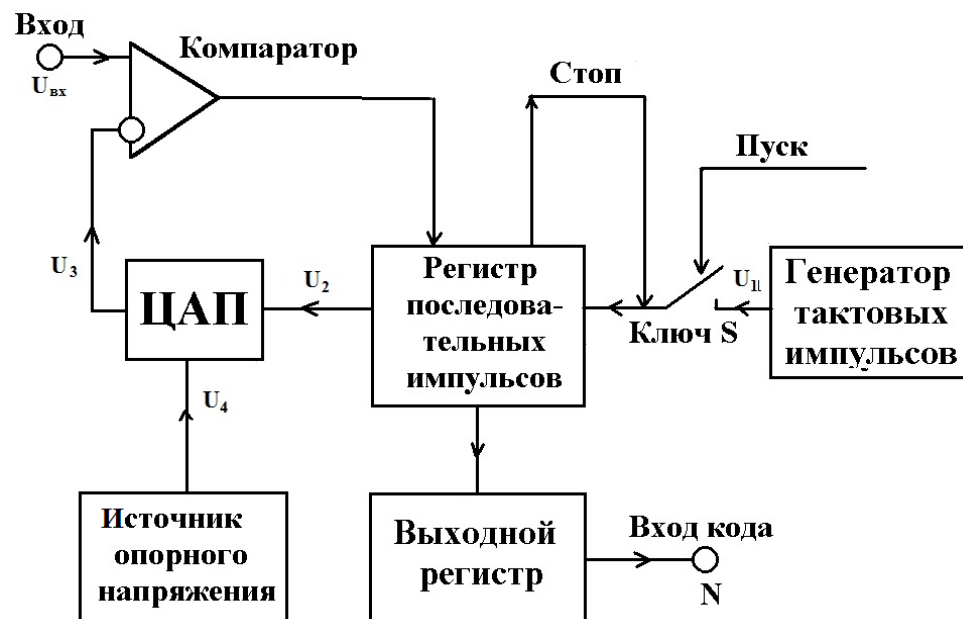
## Структурная схема АЦП последовательного счета



$T_{\text{пр}} = (2^n - 1)T$  – время преобразования

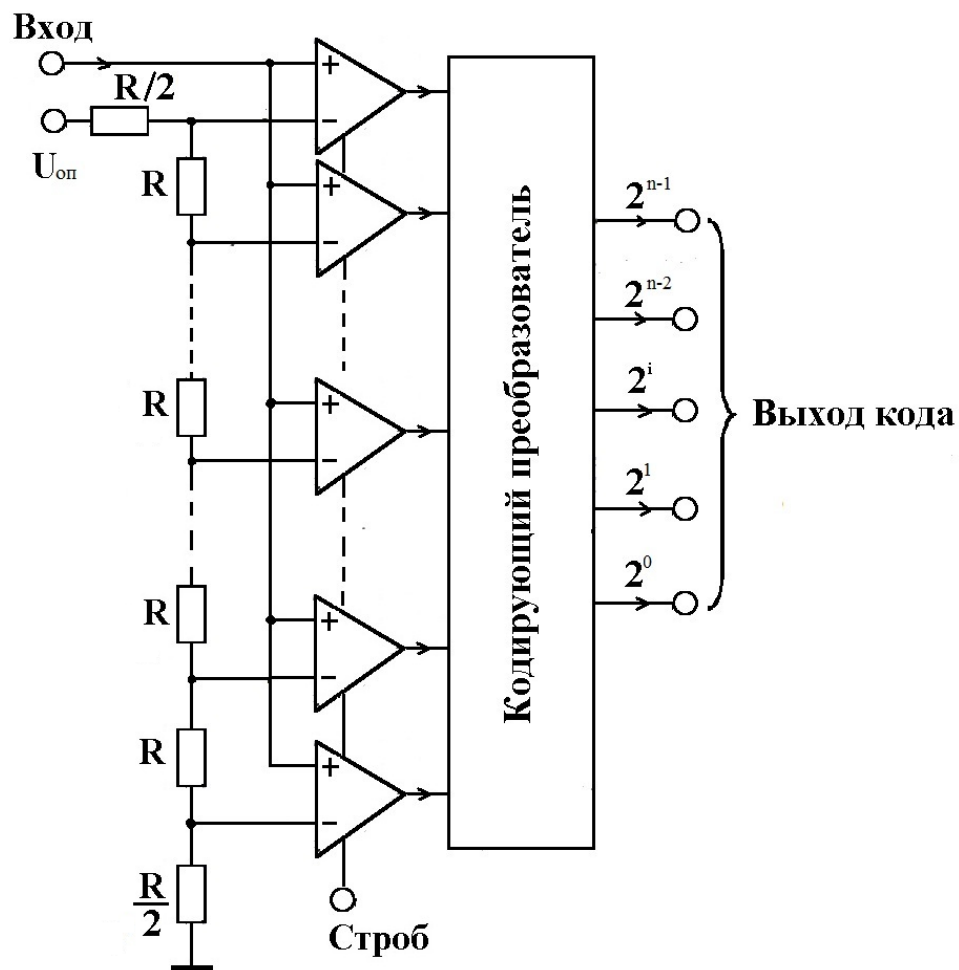
$Kh = U_{\text{вх}}$  – условие окончания преобразования

## АЦП последовательного приближения



$T_{пр} = nT$ ,  $T$  – период тактовых импульсов

## Структурная схема параллельного АЦП



$T_{пр} = T$ ,  $T$  – период тактовых импульсов

## Цифро-аналоговые преобразователи (ЦАП)

Основная характеристика - Функция преобразования ЦАП



Аналитическая запись функции преобразования:

$$U_{\text{ВЫХ}} = \frac{U_{\text{МАХ}}}{N_{\text{МАХ}}} N_{\text{ВХ}}$$

Отношение  $K_{\text{цап}} = U_{\text{МАХ}} / N_{\text{МАХ}}$  называется **коэффициентом цифро-аналогового преобразования**.

## Цифро-аналоговые преобразователи (ЦАП)

Принцип работы большинства ЦАП состоит в суммировании долей аналоговых сигналов (веса разряда), в зависимости от входного кода.

$$U_{\text{вых}} = K_{\text{цат}} \sum_1^n A_i U_i$$

**Разрешающая способность** (вес разряда) определяется для конкретной разрядности, и вычисляется по следующей формуле

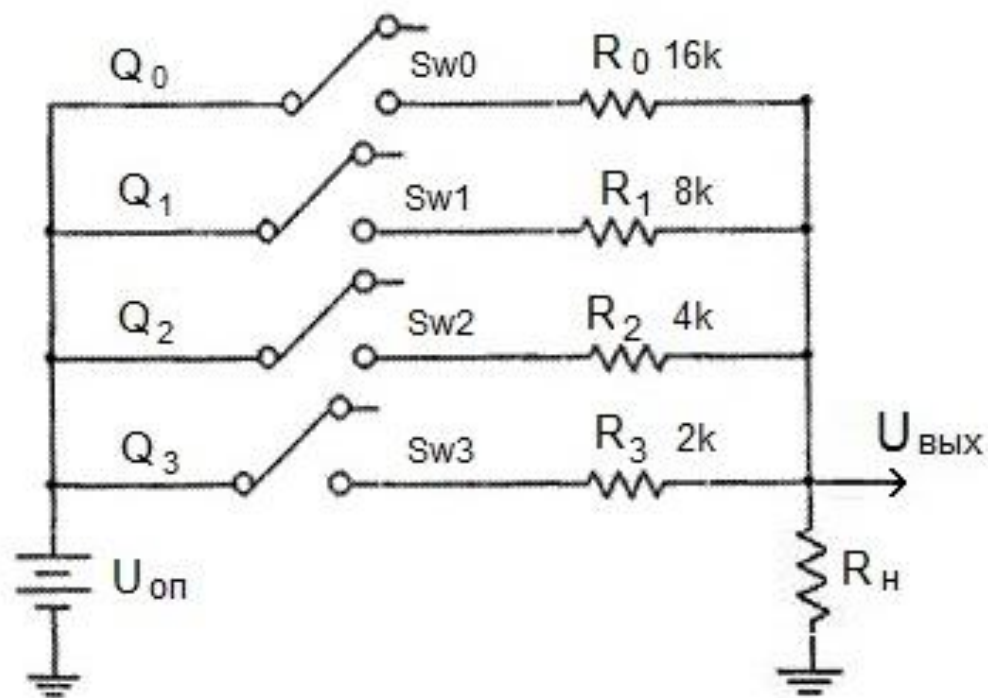
где  $U_{\text{оп}}$  - опорное напряжение ЦАП.

$$U_i = \frac{U_{\text{оп}}}{2^n}$$

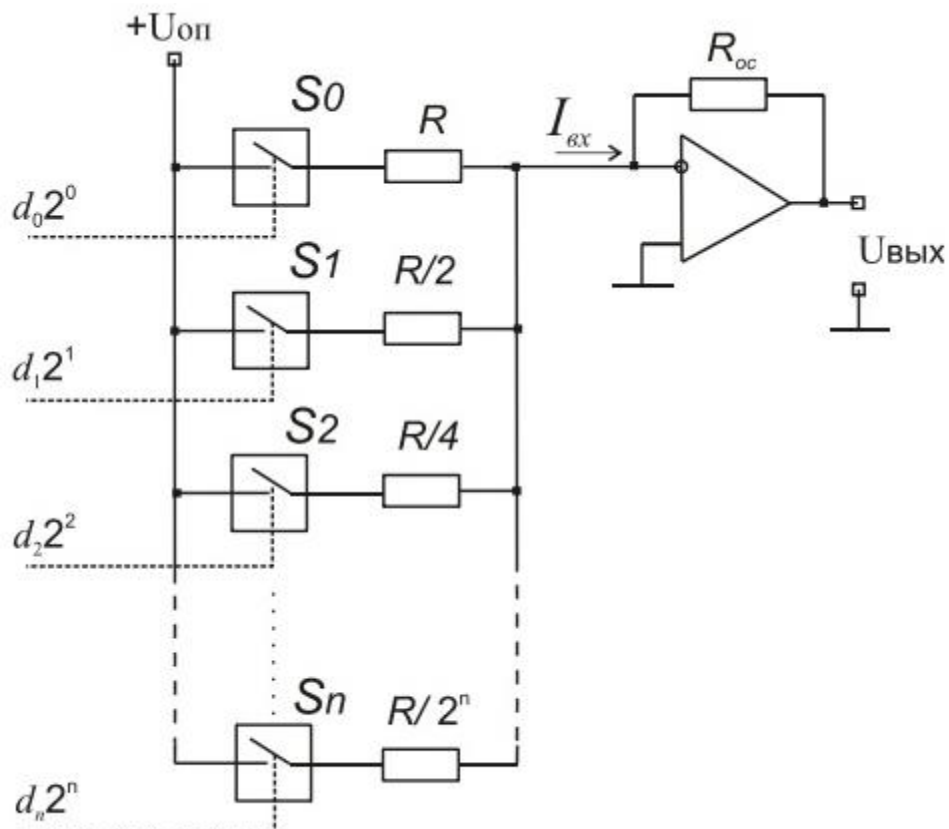
**Абсолютная погрешность** преобразования - относительная разность между реальным и идеальным значениями предела шкалы преобразования при отсутствии смещения нуля.

**Погрешность смещения нуля** - значение  $U_{\text{вых}}$ , когда входной код ЦАП равен нулю.

## Схема ЦАП с весовыми сопротивлениями



## Схема ЦАП с весовыми сопротивлениями



$$U_{\text{вых}} = -I_{\text{вх}} \cdot R_{\text{oc}}$$

$$I_{\text{вх}} = U_{\text{оп}} (d_0/R + d_1 \cdot 2/R + d_2 \cdot 4/R + \dots + d_n \cdot 2^n/R)$$

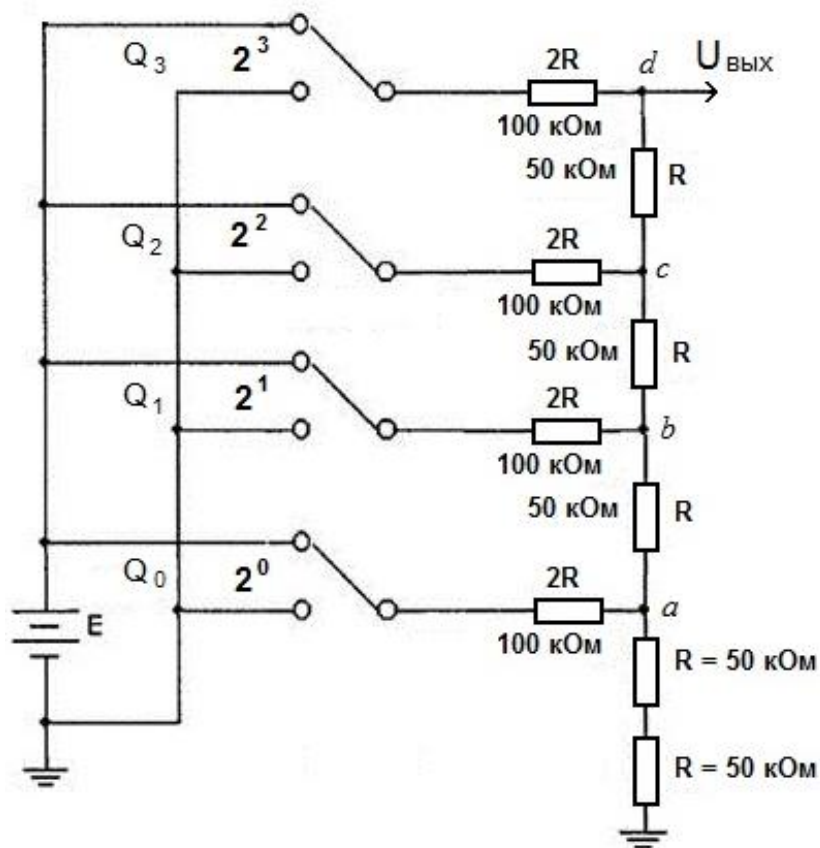
## Схема ЦАП с весовыми сопротивлениями. Недостатки.

- При высокой разрядности сопротивления резисторов должны быть согласованы с высокой точностью.
- Особо жесткие требования предъявляются к резисторам старших разрядов, поскольку разброс тока в них не должен превышать тока младшего разряда. Разброс сопротивления в  $n$ -м разряде должен быть меньше, чем:
$$\frac{\Delta R}{R} = \frac{1}{2^n}$$
- При различных входных кодовых состояниях потребляемый от источника опорного напряжения (ИОН) ток будет также различным, что, несомненно, повлияет на величину выходного напряжения ИОН.
- Сопротивления весовых резисторов могут отличаться в тысячи раз, что затрудняет реализацию таких резисторов в полупроводниковых ИС. Помимо этого, сопротивления резисторов старших разрядов могут быть соизмеримы с сопротивлением замкнутого ключа, а это приведет к погрешностям преобразования.

*Такая схема ЦАП применяется при небольшом числе разрядов ( $n < 8$ ).*



## Схема ЦАП типа R-2R



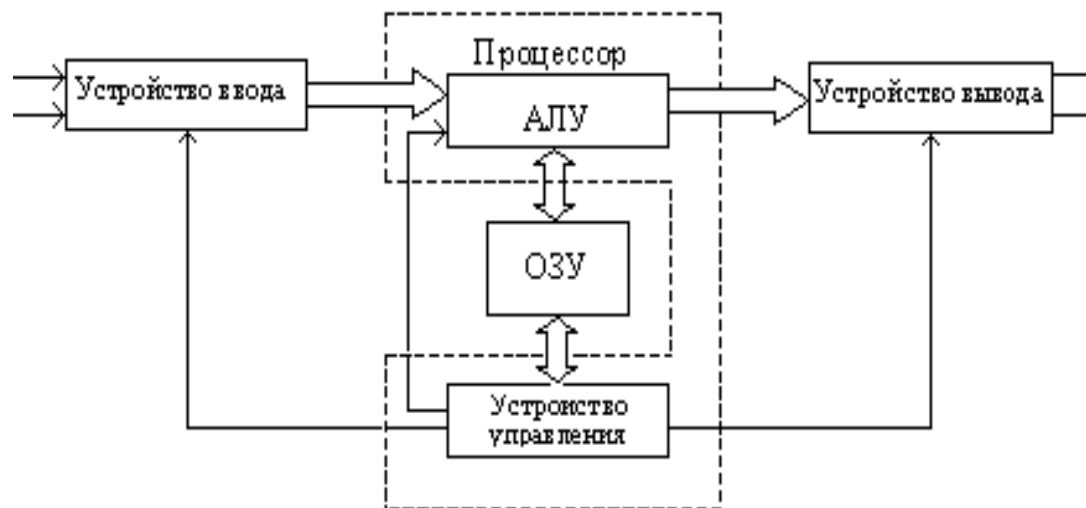
На выходе каждого элемента матрицы получается вдвое меньшее напряжение, чем на его входе. Отсюда

$$U_{\text{вых}} = E * (1 * 1/2 + 0 * 1/4 + 0 * 1/8 + 1 * 1/16 + 0 * 1/32 + 1 * 1/64 + 0 * 1/128 + 1 * 1/256)$$

## Схема ЦАП с сопротивлениями R-2R. Преимущества.

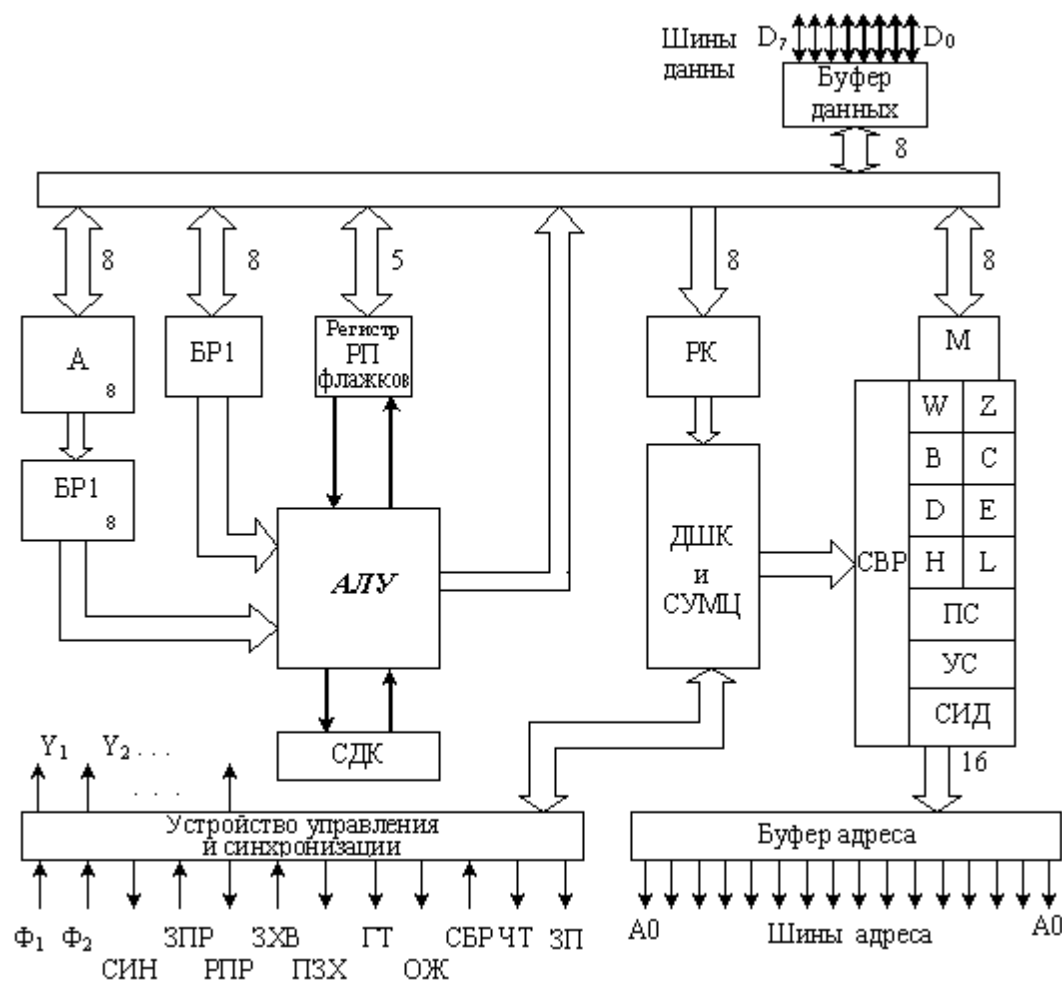
- 1) Используются всего два номинала сопротивлений резисторов: R и 2R.
- 2) Нагрузка источника опорного напряжения ( $R_{вх}$ ) не изменяется при любом числе последовательно включенных элементов матрицы и остается равной 2R.
- 3) Возможность увеличения разрядности без изменения точности.

## Обобщенная структурная схема компьютера

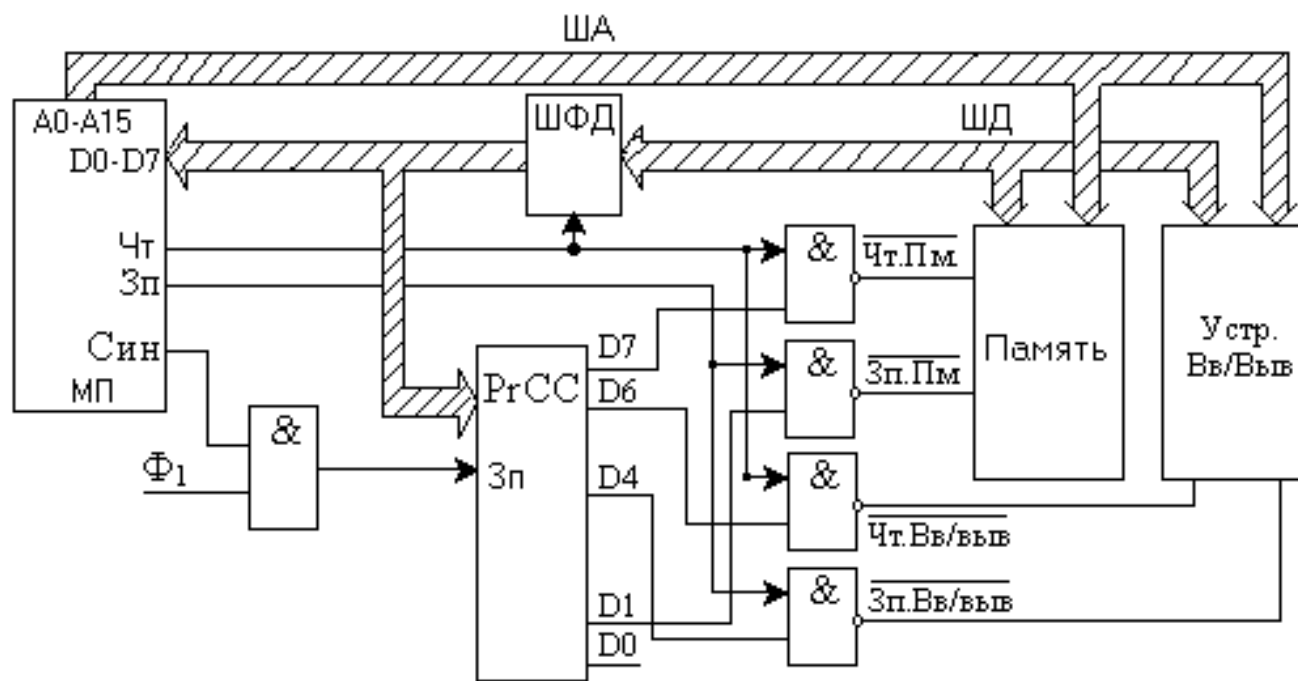


2 типа архитектуры: *Неймановская* и *Гарвардская*.

## Структурная схема 8-разрядного однокристалльного МП

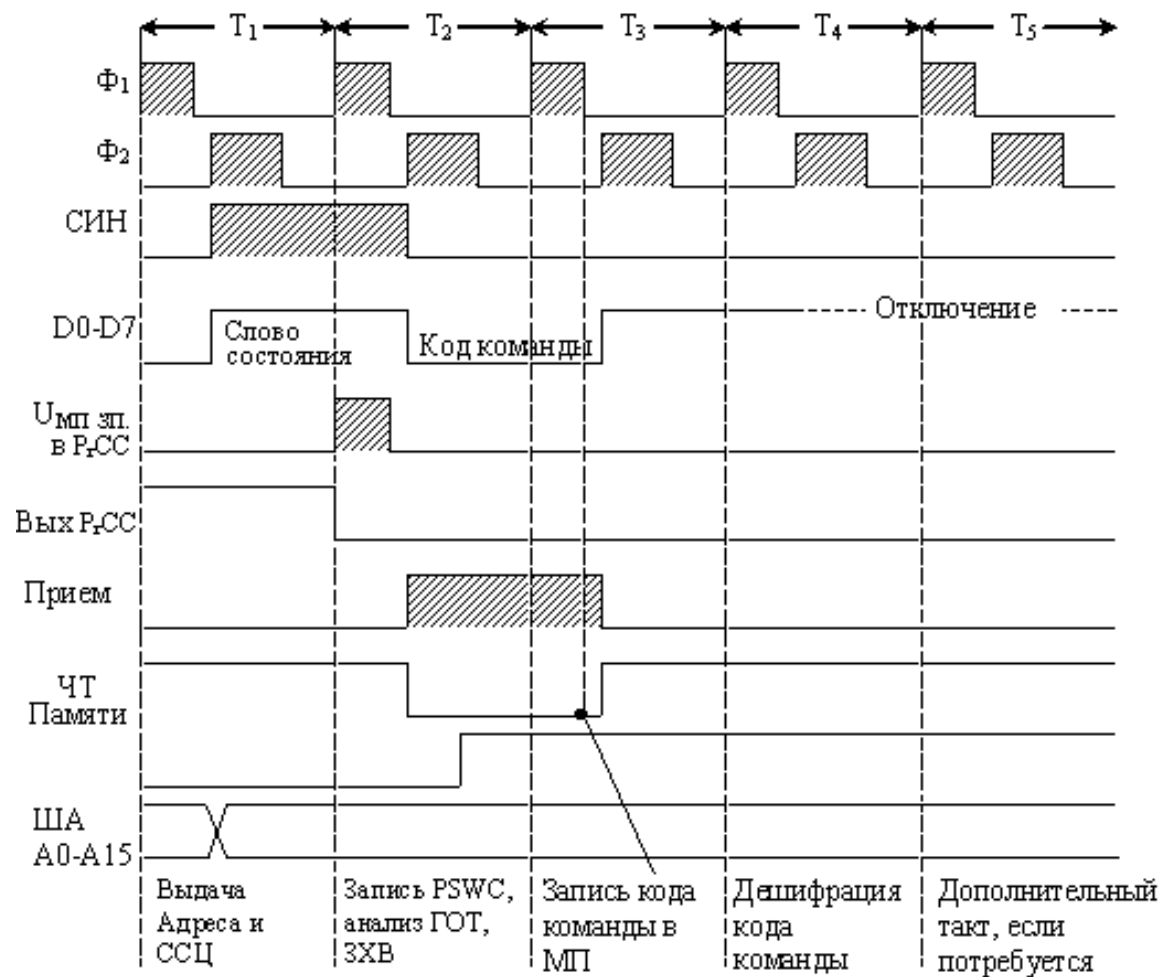


## Структурная схема 8-разрядной микро-ЭВМ



INTA	- D <sub>0</sub>
WO	- D <sub>1</sub>
STACK	- D <sub>2</sub>
HLTA	- D <sub>3</sub>
OUT	- D <sub>4</sub>
M1	- D <sub>5</sub>
INP	- D <sub>6</sub>
MEMR	- D <sub>7</sub>

## Временная диаграмма функционирования 8-разрядного МП



## Система команд 8-разрядного МП

- |                             |                            |
|-----------------------------|----------------------------|
| 1) Передачи данных;         | 5) Регистровых операций;   |
| 2) Арифметических операций; | 6) Передачи управления;    |
| 3) Логических операций      | 7) Работа со стеком;       |
| 4) Сдвига;                  | 8) Ввода/вывода;           |
|                             | 9) Управление процессором. |

Метка: **MOV**  $r_1, r_2$  ; Комментарий.  $r_1 = A, B, C, D, E, H, L, M$   $r_2 = A, B, C, D, E, H, L, M$ .  
**MVI**  $r$ , число ; Комментарий

Метка: **INR**  $r$  ; **DCR**  $r$   
**LDA** Адрес ; **STA** Адрес

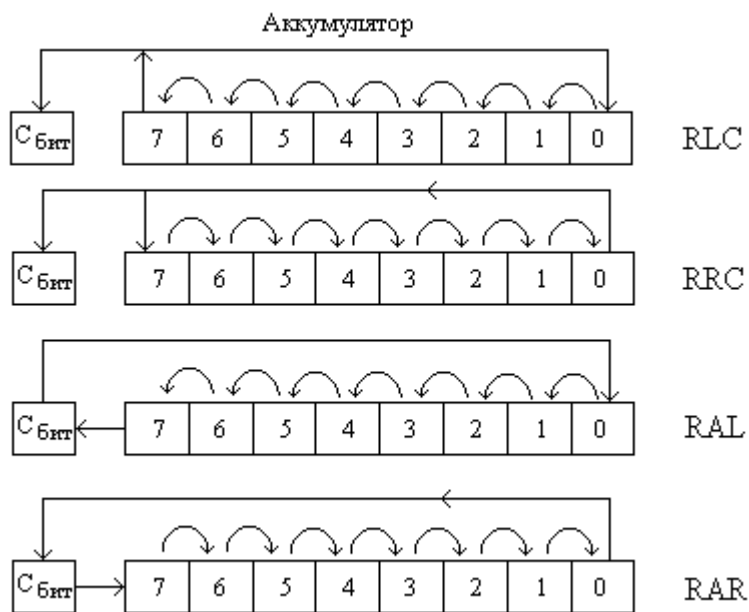
**PUSH**  $rp$  ( $rp = B, C; D, E; H, L, PSW$ ); **POP**  $rp$   
**ADD**  $r$  ; **ADD**  $M$  ; **ADC**  $r$  ; **ADI** число; **DAD**  $rp$   
**SUB**  $r$  ; **SUB**  $M$  ; **SBB**  $r$  ;

**ANA**  $r (M)$ ; **ANI** число; **ORA**  $r (M)$ ; **XRA**  $r (M)$ ; **CMP**  $r (M)$

**INR**  $r$  ; **DCR**  $r$  ; **INX**  $rp$  ; **DCX**  $rp$

**IN**  $Nport$  ; **OUT**  $Nport$  ;

## Система команд 8-разрядного МП. Команды сдвига и передачи управления

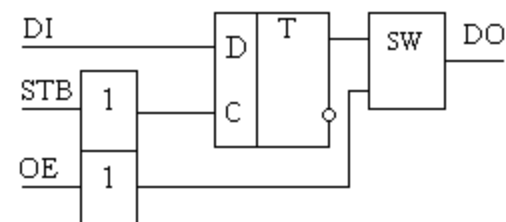
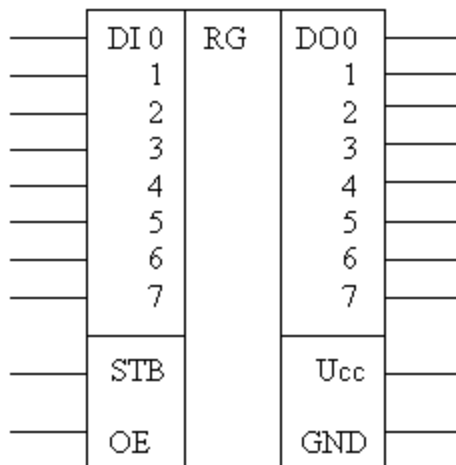
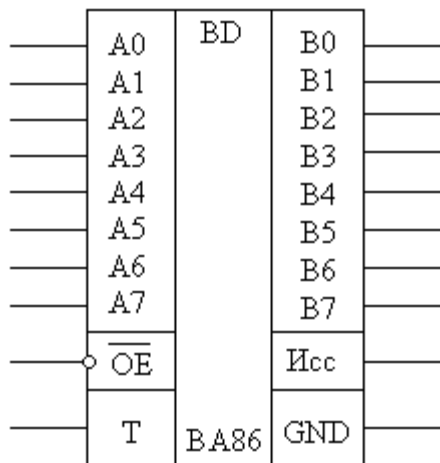


Команды безусловной и условной передачи управления:

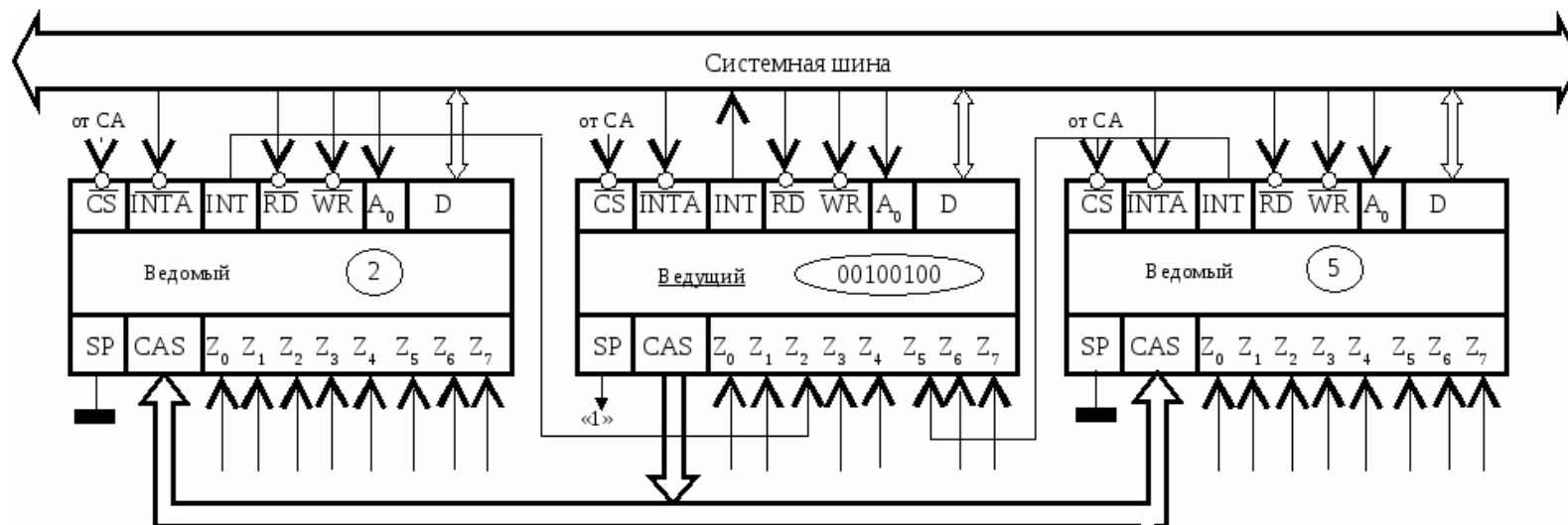
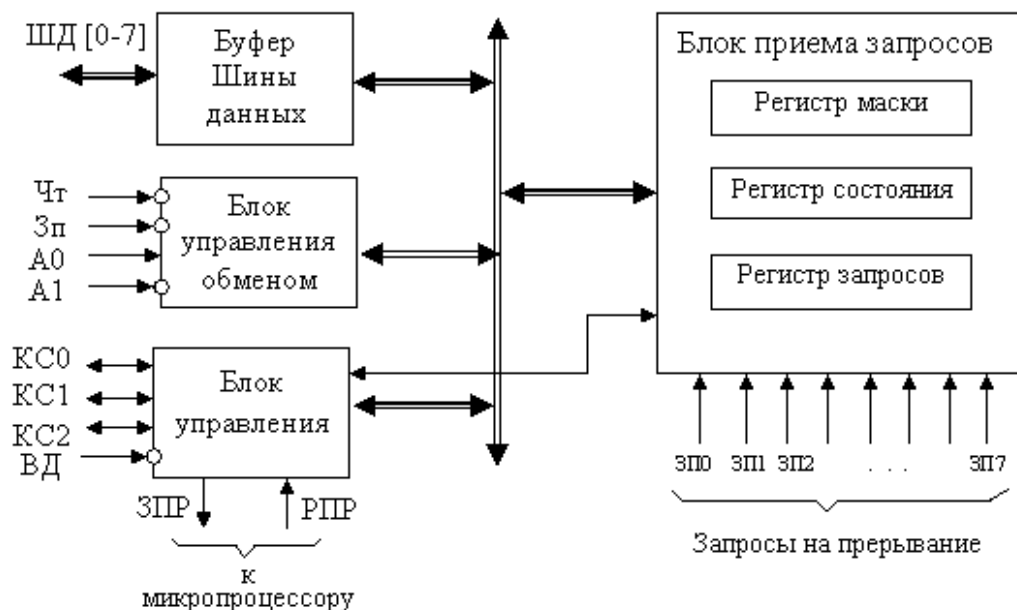
**JMP addr;    JC addr;    JNC addr;    JZ addr;    JP addr;    JM maddr**



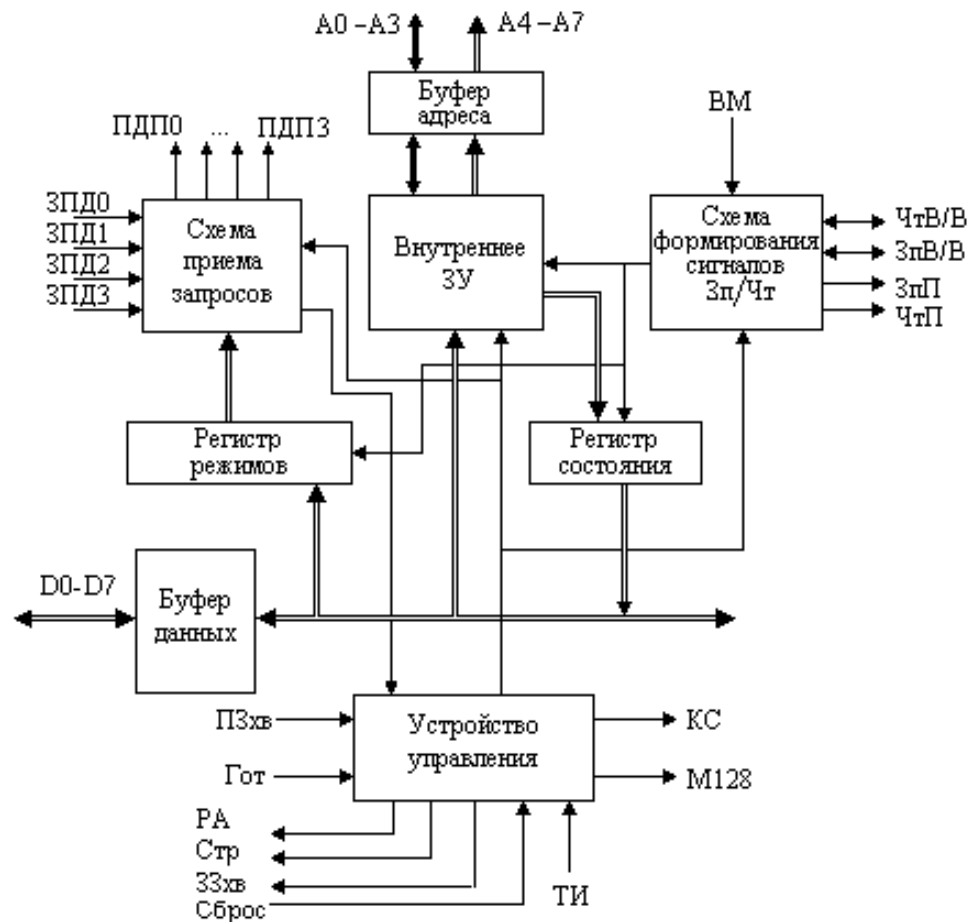
## Шинные формирователи и Порты ввода-вывода



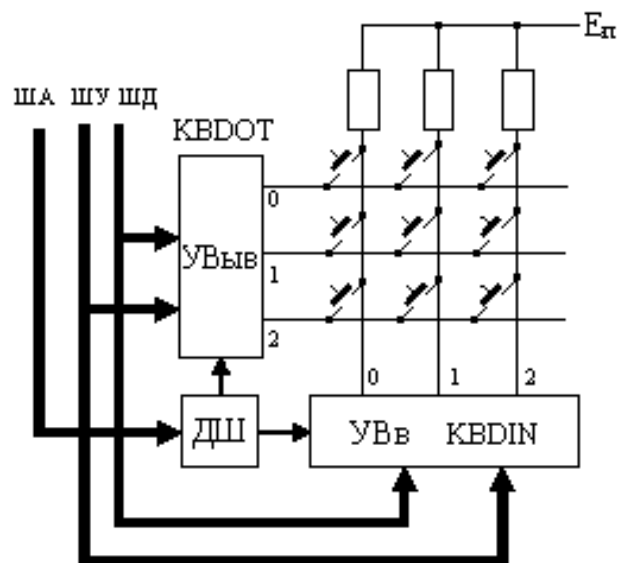
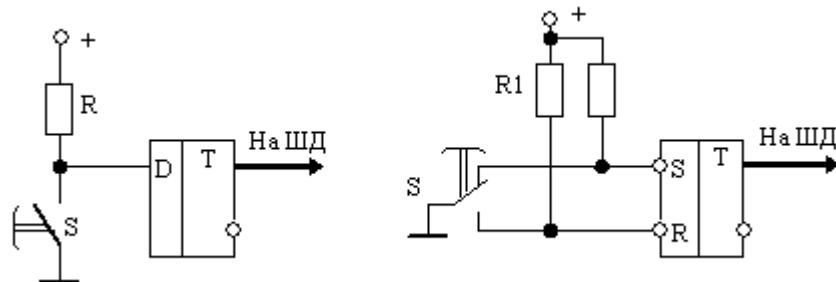
## Программируемый контроллер прерываний



## Программируемый контроллер прямого доступа в память

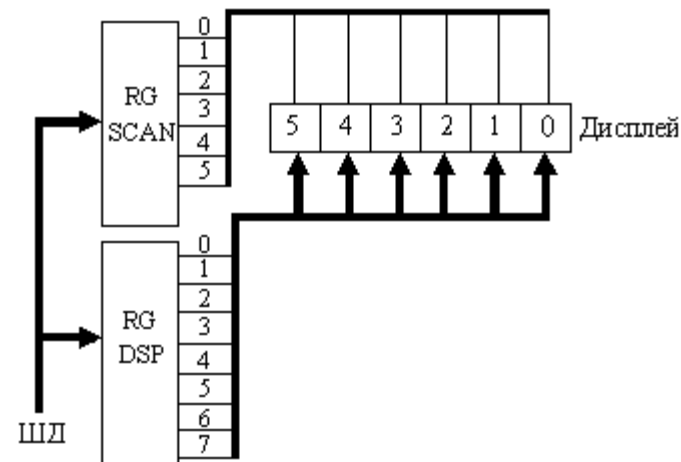
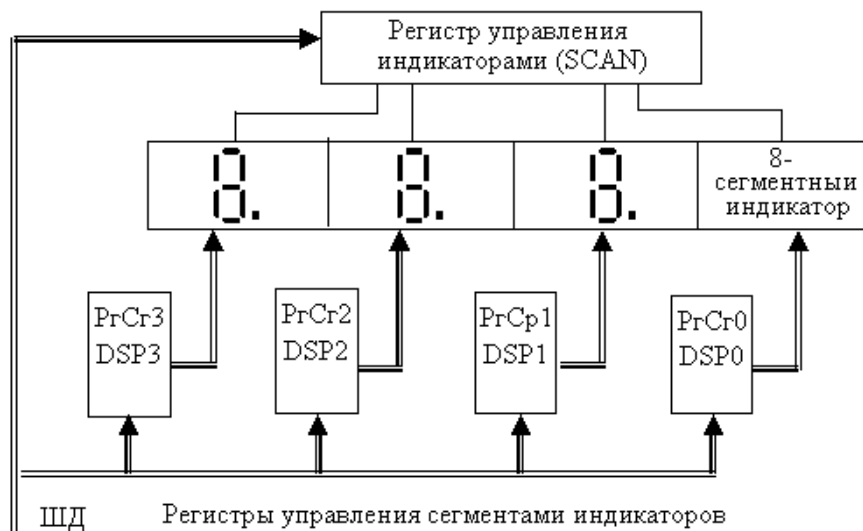


## ПОДКЛЮЧЕНИЕ КЛАВИАТУРЫ К ПРОЦЕССОРУ

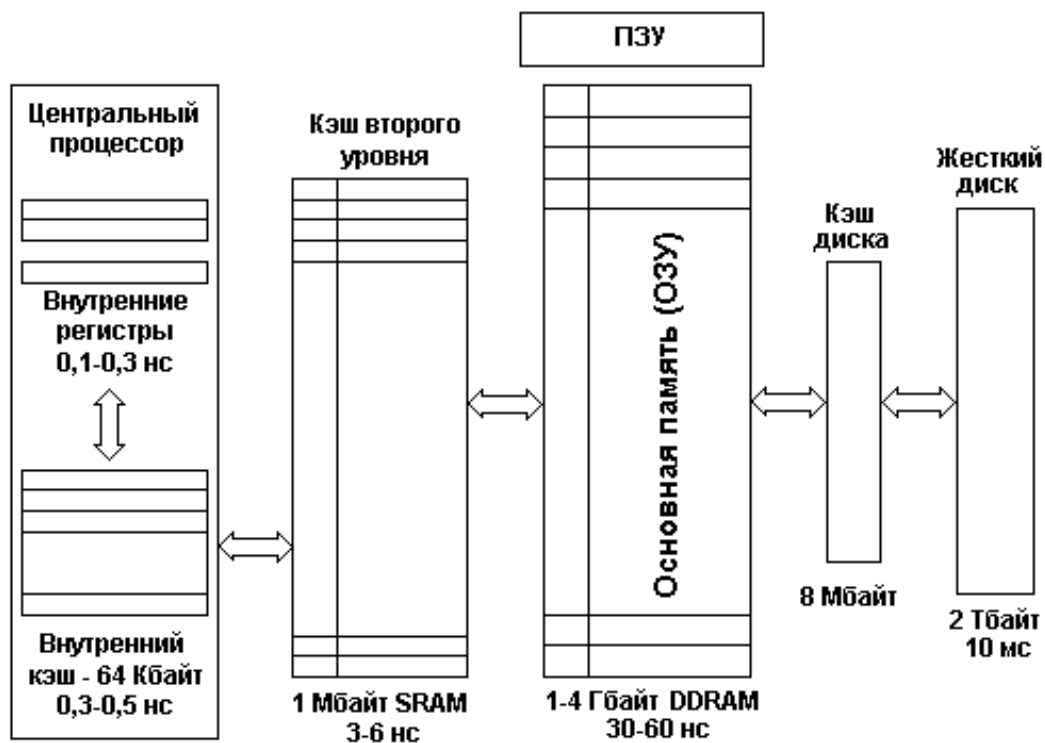


Адрес	Машинный код	Метка	Мнемокод	Комментарий
0800	3E F8	M1	MVI A, 11111000	Занести 0 в младшие три разряда аккумулятора
0802	D3 KBDOT		OUT KBDOT	Записать 0 в выходное устройство с адресом KBDOT
0804	DB KBDIN		IN KBDIN	Получить число со входного устройства с адресом KBDIN
0806	E6 07		ANI 00000111	Очистить старшие пять разрядов аккумулятора
0808	FE 07		CPI 00000111	Проверить, есть ли в младших трех разрядах аккумулятора 0
080A	CA 0008		JZ M1	Если нет, то идти на M1
080B	C3 0D08	DONE	JMP DONE	Конец

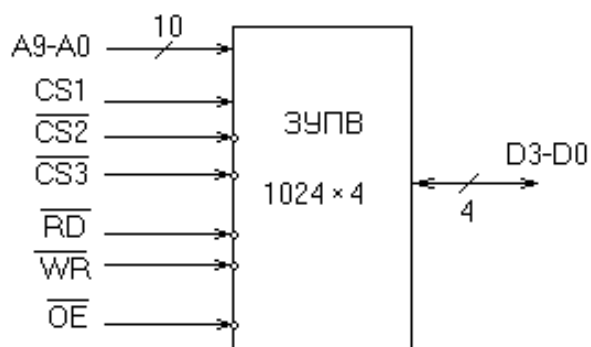
## СТАТИЧЕСКАЯ И ДИНАМИЧЕСКАЯ ИНДИКАЦИЯ



## ИЕРАРХИЯ ПАМЯТИ КОМПЬЮТЕРА



## ПОЛУПРОВОДНИКОВЫЕ ЗУ С ПРОИЗВОЛЬНЫМ ДОСТУПОМ



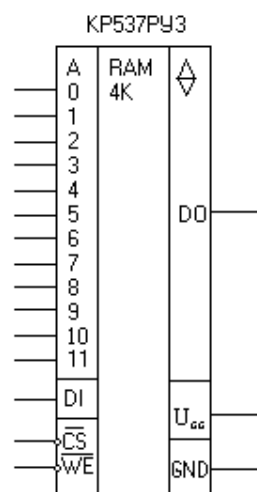
A9-A0 – Адресные шины

CS1 } выбор кристалла  
CS2 }  
CS3 } (объединенные схемой И)

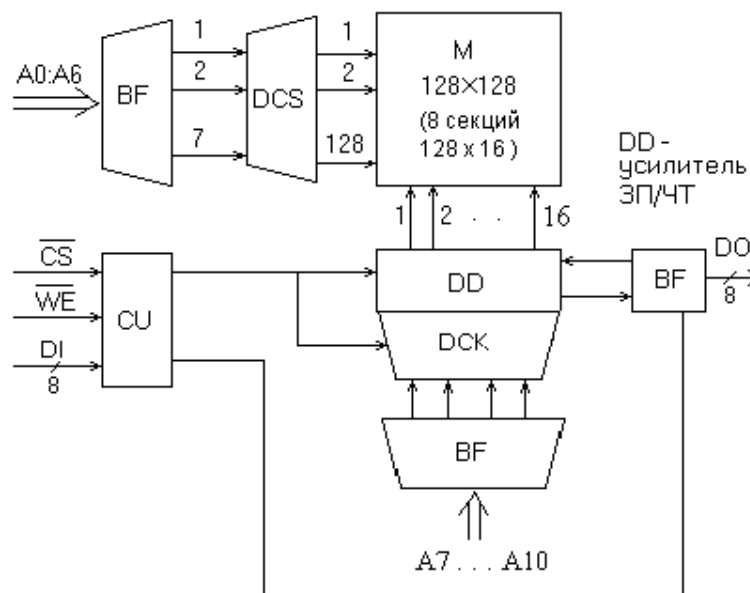
RD – чтение, в ряде устройств может быть RD/WR

WR – запись

OE – (output enable) разрешение выдачи (стробируемый выход)



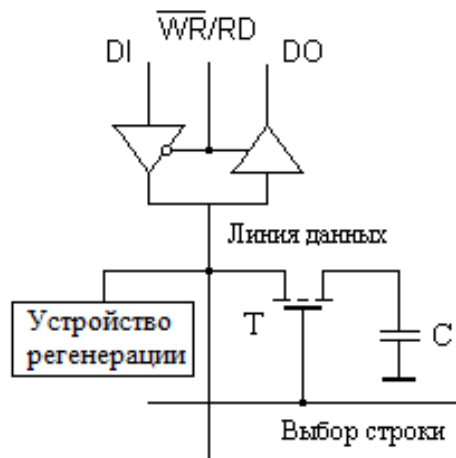
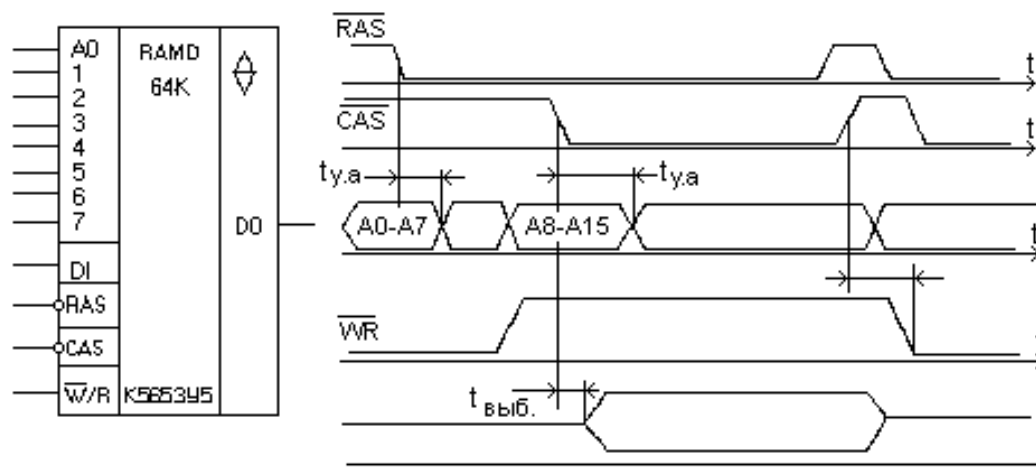
a)



б)

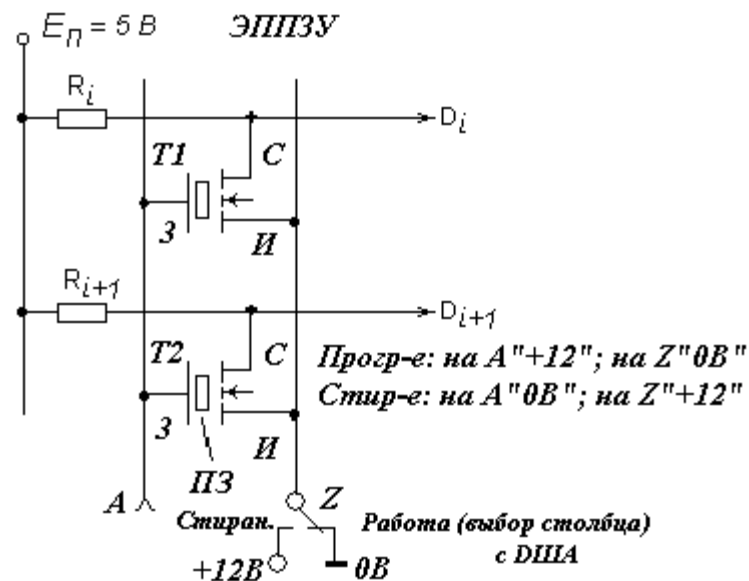
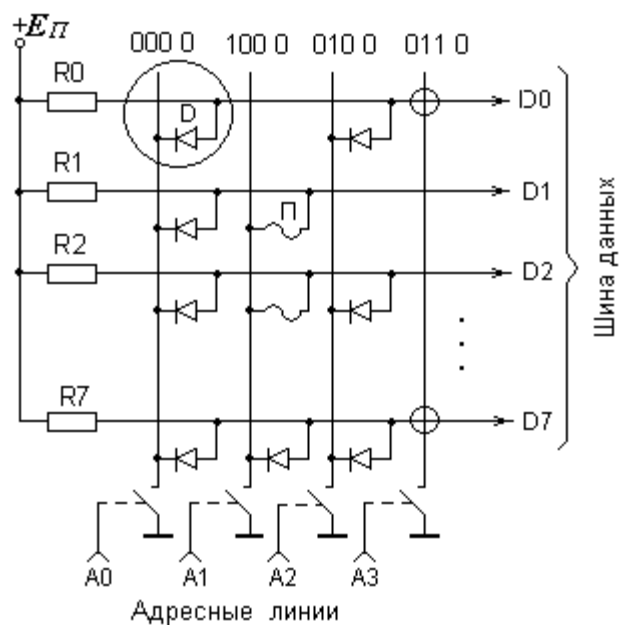
KP537PY10

## ДИНАМИЧЕСКИЕ ЗУ С ПРОИЗВОЛЬНЫМ ДОСТУПОМ

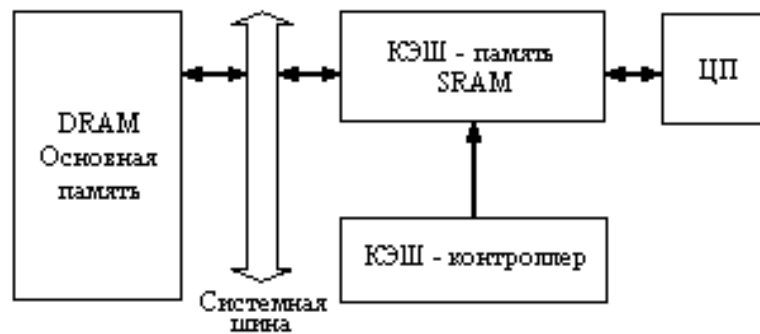




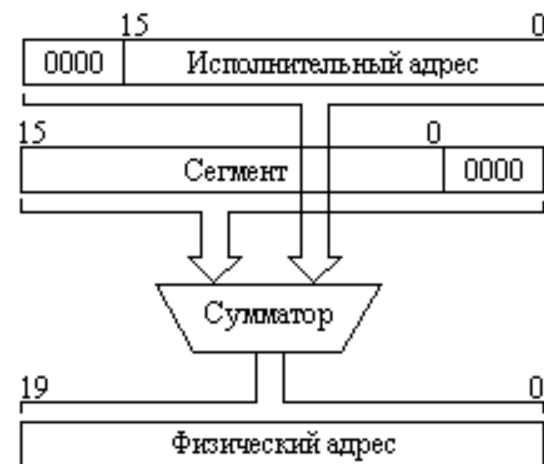
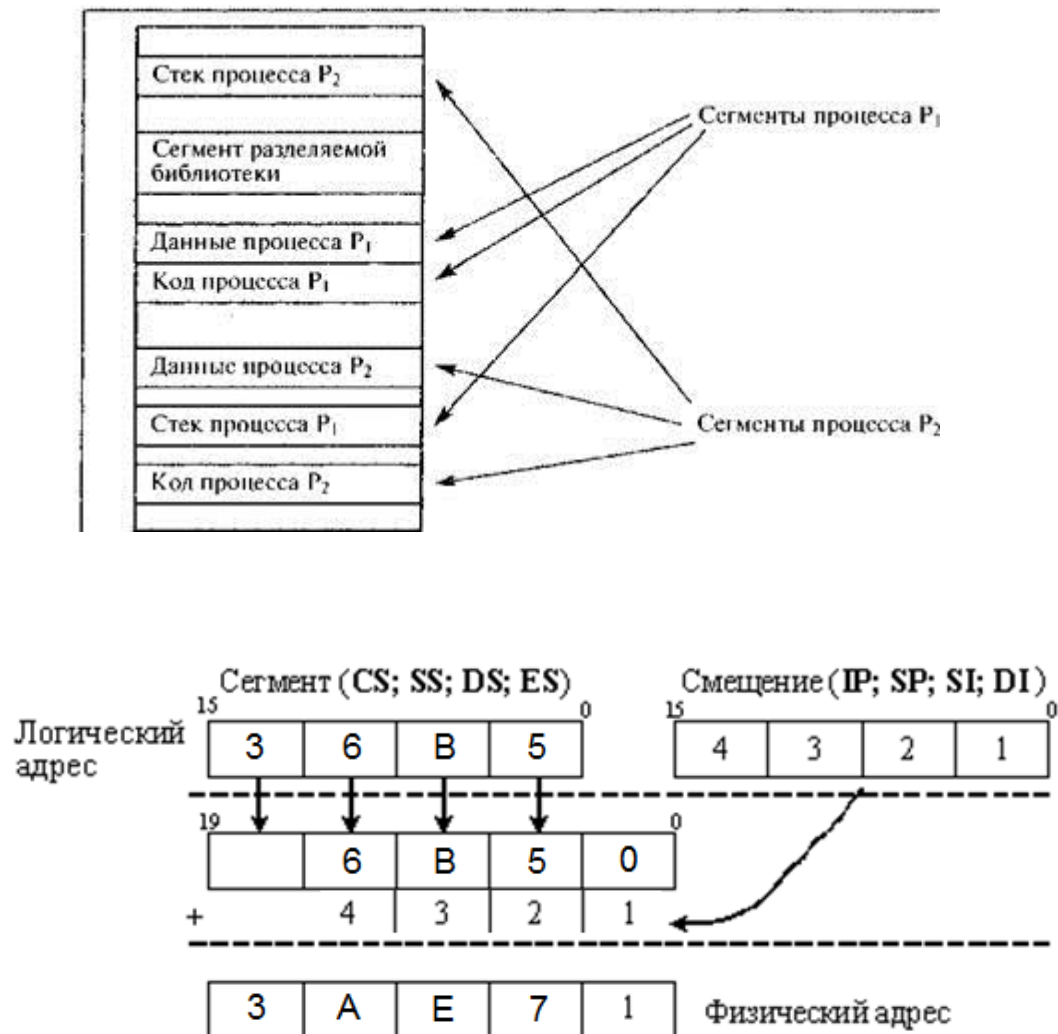
## ПОСТОЯННЫЕ ЗАПОМИНАЮЩИЕ УСТРОЙСТВА



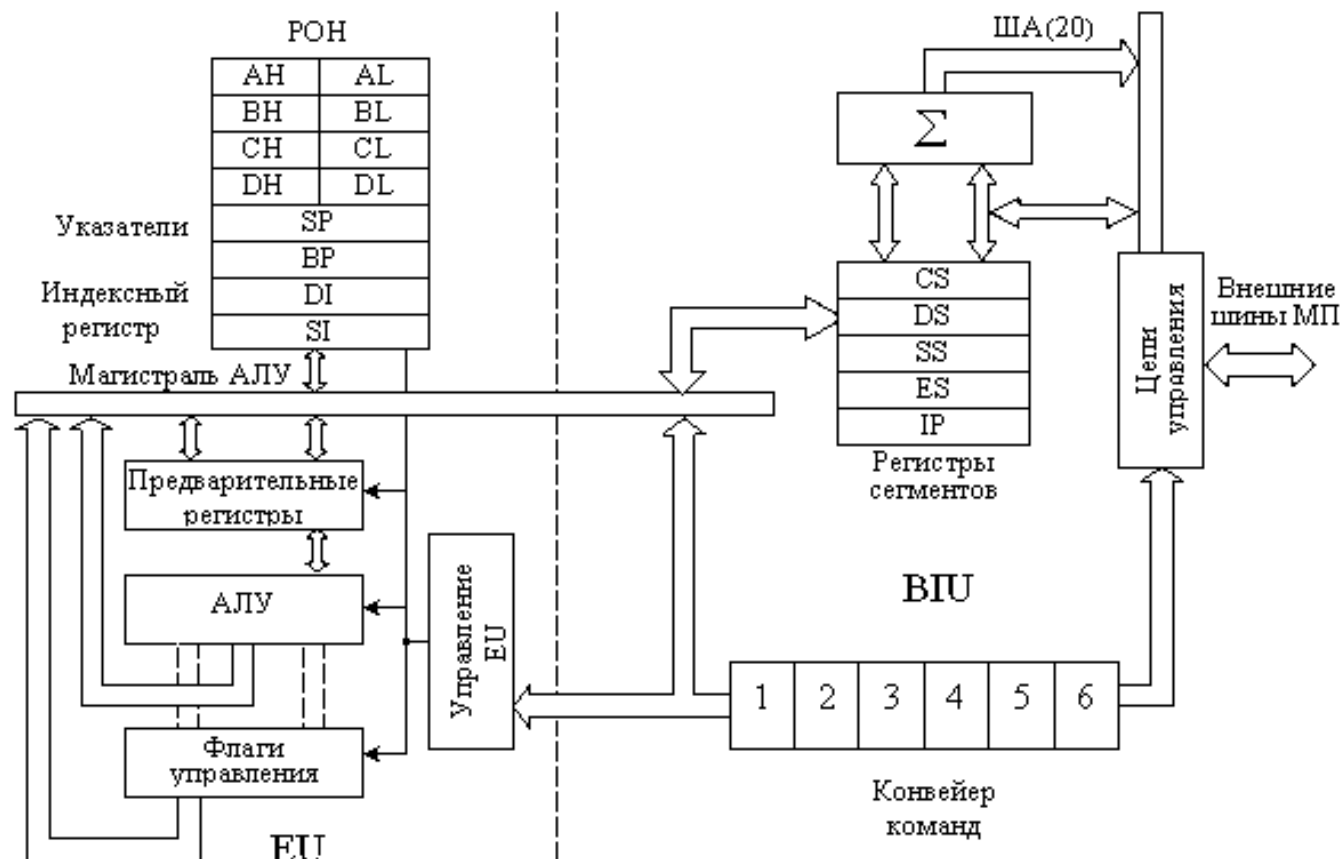
## КЭШ-ПАМЯТЬ КОМПЬЮТЕРА



## 16-РАЗРЯДНЫЕ ПРОЦЕССОРЫ. СЕГМЕНТАЦИЯ ПАМЯТИ



## СТРУКТУРА 16-РАЗРЯДНОГО ПРОЦЕССОРА



## РЕГИСТРЫ 16-РАЗРЯДНОГО ПРОЦЕССОРА

Intel 8080			Intel 8086			
	7	0		7	0	
		A	AX	AH	AL	Акумулятор
HL	H	L	BX	BH	BL	База
BC	B	C	CX	CH	CL	Счетчик
DE	D	E	DX	DH	DL	Данные
	15	0		15	0	
SP			SP			Указатель стека
			BP			Указатель базы
			DI			Индекс приемника
			SI			Индекс источника
	15	0		15	0	
			CS			Код
			DS			Данные
			SS			Стек
			ES			Дополнительные
	15	0		15	0	
PC			IP			Instruction Pointer
Флаги		Z S A P C	Флаги			
						O D I T S Z A P C

## БАЗОВЫЕ КОМАНДЫ 16-РАЗРЯДНОГО ПРОЦЕССОРА

Пере-сылки данных	1	MOV r, r	Пере-сылка данных	27	LEA r	Ариф-метиче-ские	74	DIV src	98	AND r1, r2		
	2	MOV r, mem		28	LDS r, mem		52	SUB r, data	75	IDIV src	99	AND r, mem
	3	MOV mem, r		29	LES r, mem		53	SUB mem, data	76	DAA	100	AND mem, r
	4	MOV mem, data		30	LAHF		54	SUB a, data	77	DAS	101	AND r, data
	5	MOV r, data		31	SAHF		55	SBB r1, r2	78	AAA	102	AND mem, data
	6	MOV a, mem		32	PUSHF		56	SBB r, mem	79	AAS	103	AND a, data
	7	MOV mem, a	Арифметические			57	SBB mem, r	80	AAM	104	OR r1, r2	
	8	MOV seg, r	34	ADD r1, r2	58	SBB r, data	81	AAD	105	OR r, mem		
	9	MOV seg, mem	35	ADD r, mem	59	SBB mem, data	82	CBW	106	OR mem, r		
	10	MOV r, seg	36	ADD mem, r	60	SBB a, data	83	CWD	107	OR r, data		
	11	MOV mem, seg	37	ADD r, data	Сдвиги	84	SHL/SAL r	108	OR mem, data			
	12	PUSH r	38	ADD mem, data				85	SHL/SAL mem	109	OR a, data	
	13	PUSH mem	39	ADD a, data				86	SHR r	110	XOR r1, r2	
	14	PUSH r	40	ADC r1, r2				87	SHR mem	111	XOR r, mem	
	15	PUSH seg						88	SAR r	112	XOR mem, r	
	16	POP r						89	SAR mem	113	XOR r, data	
	17	POP mem						90	ROL r	114	XOR mem, data	
	18	POP r						91	ROL mem	115	XOR a, data	
	19	POP seg						92	ROR r	116	TEST r1, r2	
	20	XCHG r, mem	41	ADC r, mem				93	ROR mem	117	TEST r, mem	
	21	XCHG r, r	42	ADC mem, r	94	RCL r	118	TEST r, data				
	22	XCHG AX, r	43	ADC r, data	95	RCL mem	120	TEST a, data				
	23	IN port	44	ADC mem, data	96	RCR r			121	NOT r		
	24	IN	45	ADC a, data	97	RCR mem			122	NOT meme		
	25	OUT port	46	INC r	70	CMP mem, data						
	26	OUT	47	INC mem	71	CMP a, data						
		48	INC r	72	MUL src							
		49	SUB r1, r2	73	IMUL src							
		50	SUB r, mem									

```

MOV AX, 8000H
MOV DS, AX
ADC AX, 2C25h

```

## БАЗОВЫЕ КОМАНДЫ 16-РАЗРЯДНОГО ПРОЦЕССОРА

Обра- ботка строк	123	MOVS
	124	CMPS
	125	SCAS
	126	LODS
	127	STOS
	128	REPNE/REP NZ
	129	REP/REPE/PERZ
Безус- ловные переходы	130	JMP label
	131	CALL name
	132	RET
Услов- ные пере- ходы	133	JZ/JE label
	134	JL/JNGE label
	135	JLE/JNG label
	136	LOOP label
	137	LOOPNZ/LOOPNE label
	138	JCXZ label

## СПОСОБЫ АДРЕСАЦИИ ПАМЯТИ 16-РАЗРЯДНОГО ПРОЦЕССОРА

1. Регистровая адресация:      MOV AX, SI      ; <SI>→<AX>  
ADD DI, BX      ; <BX>+<DI>
2. Непосредственная адресация:      SUB AL, 30H      ; <AL>- 48 (30H = 48D)  
MOV CL, 10      ; (10→<CL>)
3. Прямая адресация:      MOV AX, GAMMA  
ADD TEMP, BL ; <BL>+ <<DS><sup>↑</sup><sup>4</sup> + TEMP>
4. Косвенная регистровая:      ADD AX, [DI]
5. Базовая адресация:      MOV AX, [BX]  
MOV AX, [BP]10
6. Индексная адресация:      MOV ADRM [SI], AX
7. Базово-индексная адресация:      MOV AX, [BX+2][DI]
8. Адресация портов ввода-вывода:      IN AL, 40H  
OUT DX, AX



## Примеры программирования 16-разрядных процессоров

**Пример:** Скопировать 100 байтов из строки SRC в строку DST.

CLD ; Установить DF=0 для обработки строки  
;слева направо

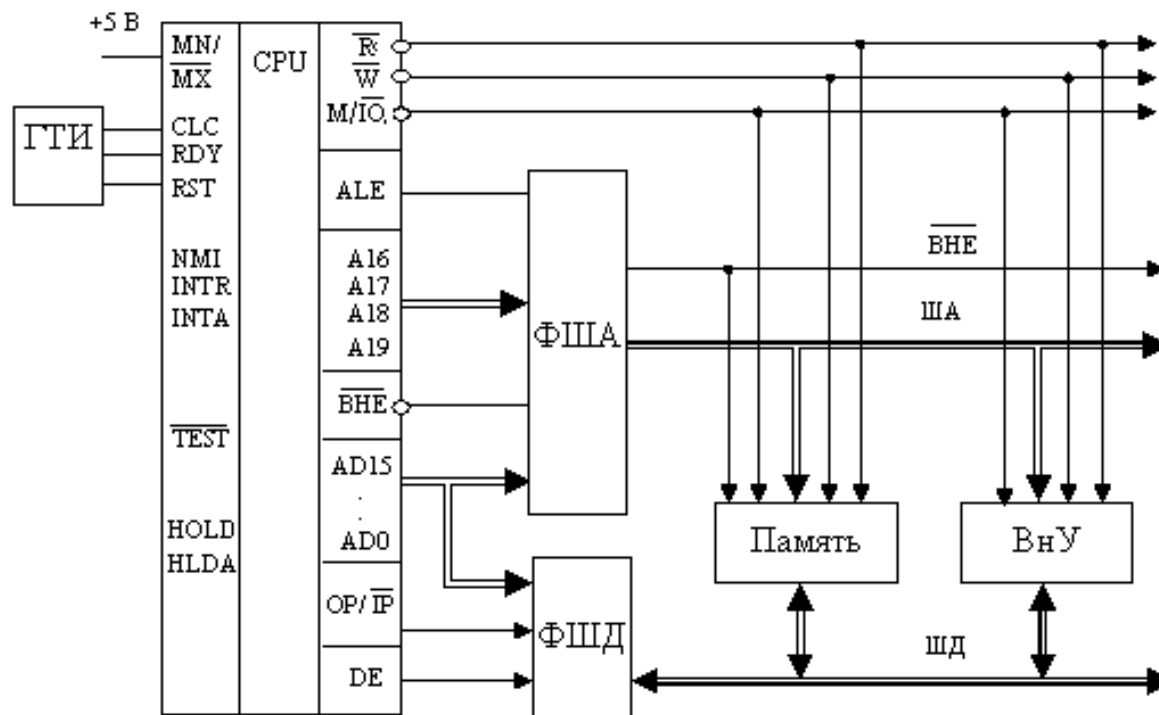
LEA SI, SRC ; Занести смещение адреса SRC в SI, а

LEA DI, ES: DST ; смещение адреса DST в DI

MOV CX, 100 ; Установить счетчик элементов

REP MOVSB, SRC ; Скопировать байты.

## СХЕМА МИКРО-ЭВМ НА БАЗЕ 16-РАЗРЯДНОГО ПРОЦЕССОРА 8086



MN/MX#- вход, задающий режим работы МП (минимальный или максимальный);

CLK- вход тактовой синхронизации

READY- входной сигнал готовности

INTA# -подтверждение запроса на прерывание.

HLDA, HOLD -запрос/разрешение захвата.

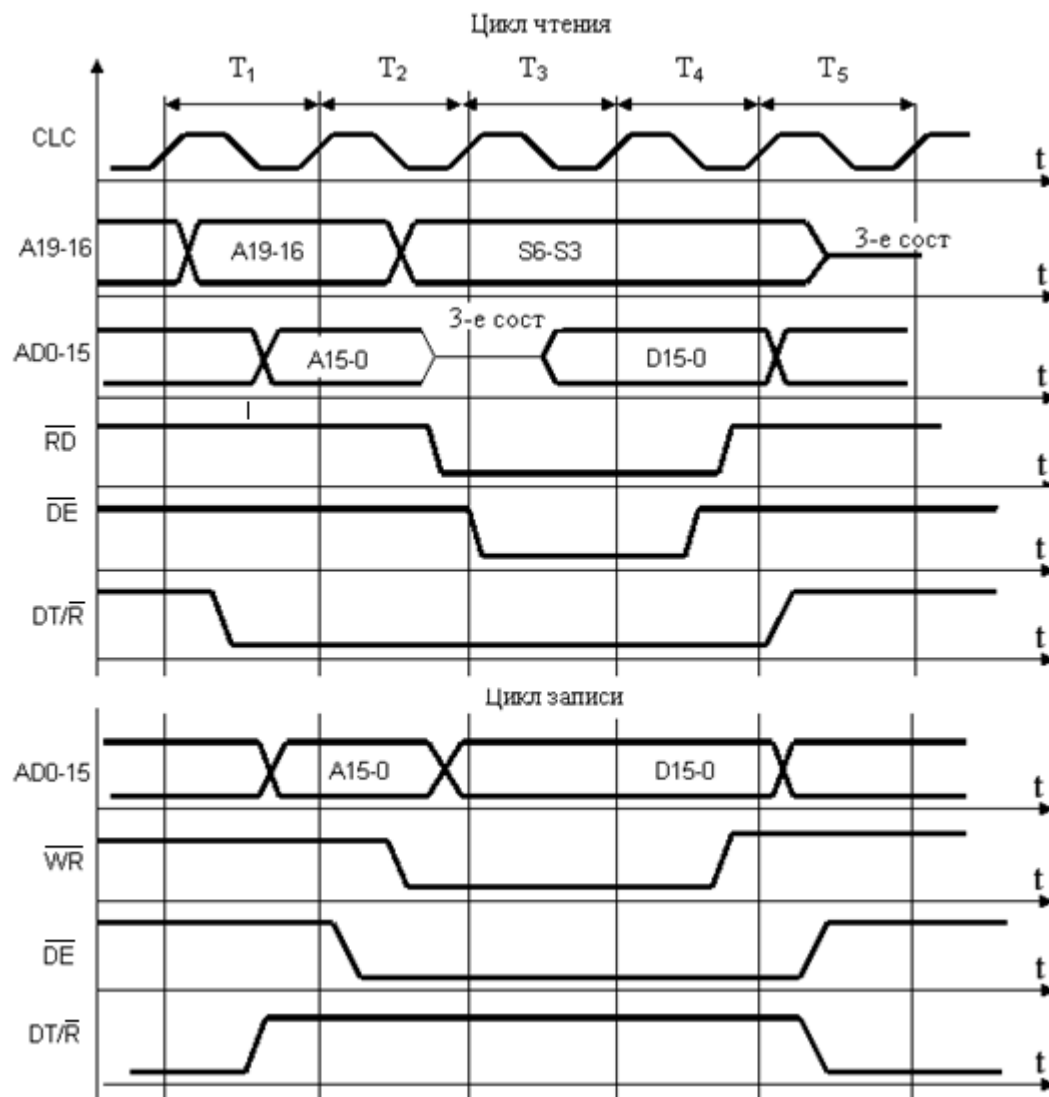
TEST#- проверка уровня сигнала, используется вместе с командой ожидания WAIT,

ALE строб адреса;

BHE#- разрешение старшего байта (Byte High Enable)

DE –разрешение передачи данных; OP/IP# - выдача/прием данных

## ВРЕМЕННАЯ ДИАГРАММА ФУНКЦИОНИРОВАНИЯ МИКРО-ЭВМ НА МП 8086

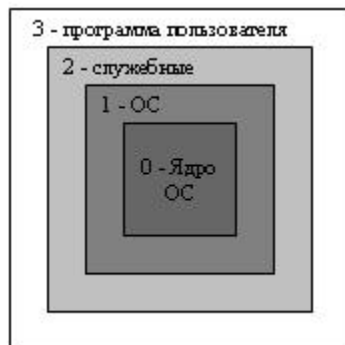


DE – переключение из 3-го состояния.

DF/R (IO/OP) – направление передачи

Обозначение вывода	Вход/выход	Назначение вывода
AD15—AD0	Вх/вых	Мультиплексированная ША/Д
A16/S <sub>3</sub> —A19/S <sub>0</sub>	Вых	Линии адреса или состояния
BHE/S <sub>7</sub>	Вых	Разрешение старшего байта шины
RD	Вых	Управление чтением
WR	Вых	Управление записью
M/IO	Вых	Выбор памяти или ВУ
ALE	Вых	Разрешение фиксации адреса
DT/R	Вых	Управление пересылкой данных
DEN	Вых	Разрешение пересылки данных
MN/MX	Вх	Установка режима
TEST	Вх	Сигнал окончания режима ожидания
HOLD	Вх	Запрос захвата шин
HLDA	Вых	Подтверждение захвата
INTR	Вх	Запрос прерывания
NMI	Вх	Запрос немаскируемого прерывания
INTA	Вых	Подтверждение прерывания
READY	Вх	Готовность памяти или ВУ
RESET	Вх	Сброс (начальная установка)
CLK	Вх	Такты ГТИ
GND, +5B	Вх	Общий (земля), питание

## ЗАЩИТА ПАМЯТИ В КОМПЬЮТЕРАХ



3 - Программы - приложения

2 - Служебные программы

1 - Служебные программы

0 - Ядро ОС (инициализация работы,  
управление доступом к памяти, защита  
и др.)

1) Защита памяти.

2) Поддержки **многозадачного**  
(мультипрограммного) режима.

3) Поддержка виртуальной памяти.

Для осуществления защиты памяти кроме базового адреса вводится *указатель размера и атрибут сегмента*. В этом случае ситуация, когда величина смещения превышает размер сегмента, считается аварийной, т.к. возможно проникновение в область соседнего сегмента.

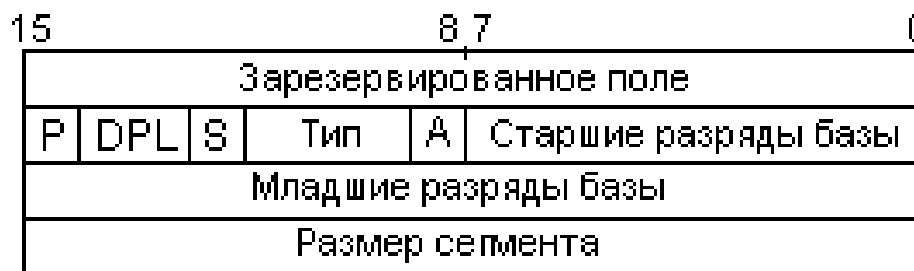
Атрибуты сегментов могут быть самыми разными. Наиболее широко применяются следующие:

- разделение на **системную область** и **область пользователя**;
- разделение на **область программ** и **данных**;
- в случае работы с областью программы производят разделение ее на участки, допускающие только считывание данных или также и запись.

## ЗАЩИТА ПАМЯТИ В КОМПЬЮТЕРАХ

**Правила доступа** для сегментов программ и данных:

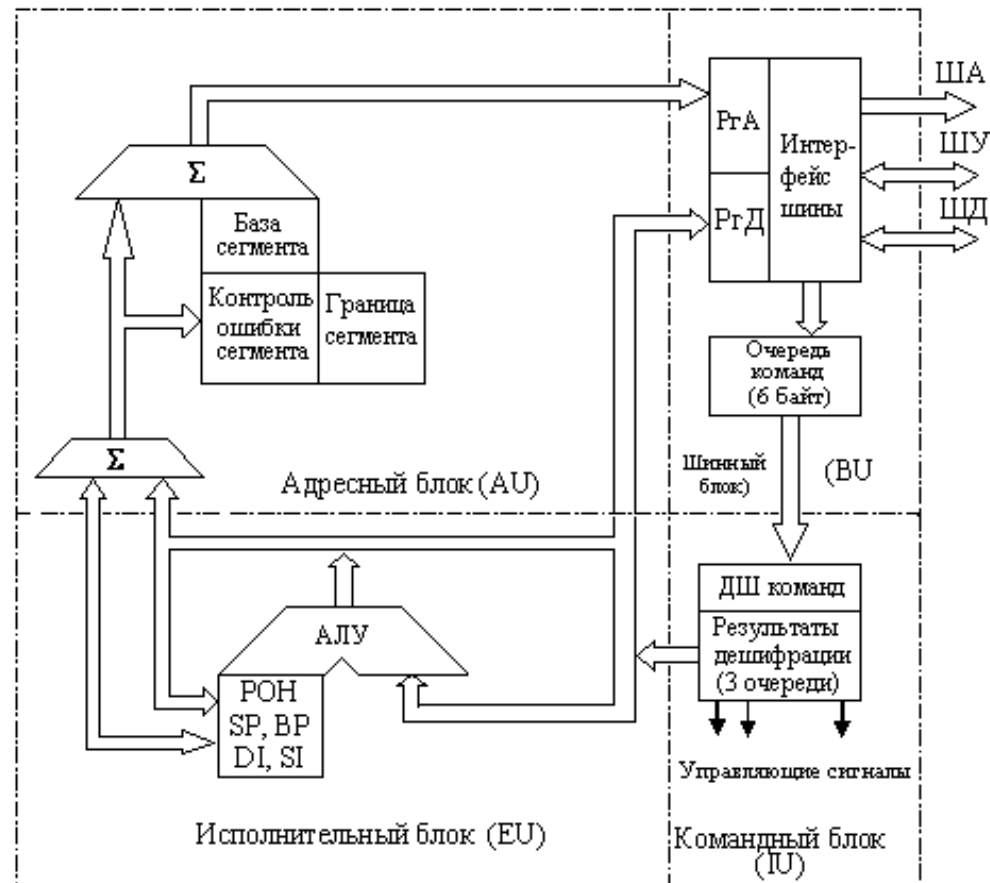
- 1) **Данные** из сегмента, имеющий уровень привилегий могут быть выбраны программой **такого** или **более высокого уровня** привилегий.
- 2) **Сегмент программ**, имеющий уровень защиты PL, может быть вызван программой, имеющий **такой** или **более низкий уровень** привилегий.
- 3) Уровень защиты и привилегий определяется двумя битами, значение которых указывает номер кольца защиты или уровня. Эти биты размещаются в байте доступа дескриптора (для привилегий дескриптора) или в селекторе (уровня привилегий запроса).
- 4) Выполнение команды **ветвления** ограничивается **сегментами в пределах одной программы**. В командах вызова и перехода допускается переход в сегменты другой программы при условии, что уровень ее привилегий равен текущему уровню.



Дескриптор сегмента

## АРХИТЕКТУРА ПРОЦЕССОРОВ 2-ГО ПОКОЛЕНИЯ

Номер бита	Назначение
0 - CF	Флаг переноса
1 - 1	Зарезервировано и равно 1
2 - PF	Флаг чётности
3 - 0	Зарезервировано и равно 0
4 - AF	Флаг вспомогательного переноса
5 - 0	Зарезервировано и равно 0
6 - ZF	Флаг нуля
7 - SF	Флаг знака
8 - TF	Флаг ловушки
9 - IF	Флаг разрешения прерываний
10 - DF	Флаг направления
11 - OF	Флаг переполнения
12-13 - IOPL	Уровень привилегий ввода/вывода
14 - NT	Флаг вложенной задачи
15 - 0	Зарезервировано и равно 0



## ЗАЩИТА ПАМЯТИ В КОМПЬЮТЕРАХ. СЕЛЕКТОР АДРЕСА.

CS:IP; DS:BP.

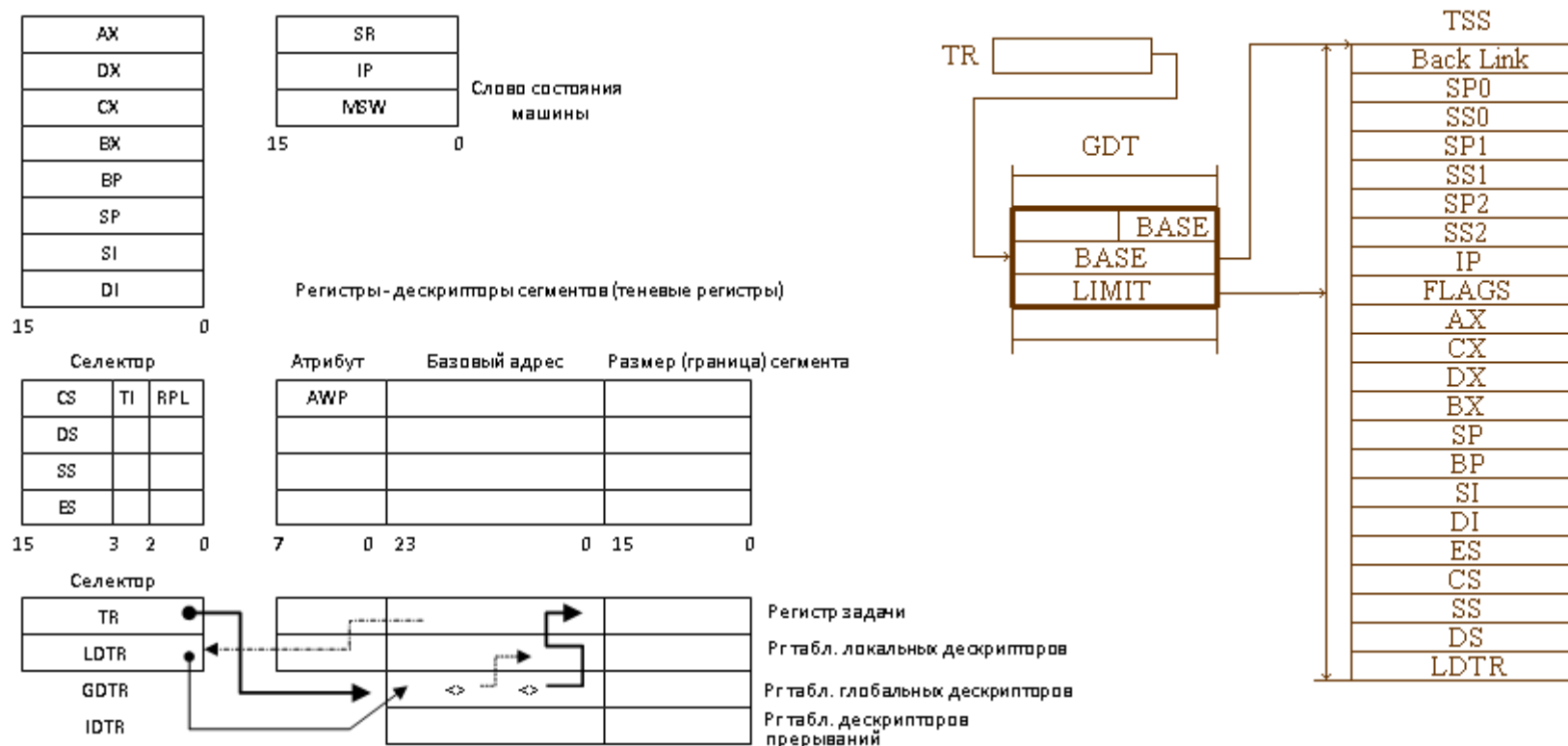


Формат селектора адреса

Два младших бита обозначены как **RPL** (**Requested Privilege Level**). Это поле является запрошенным программой уровнем привилегий и его мы будем обсуждать позже. Поле **TI** (**Table Indicator**) состоит из одного бита. Если этот бит равен нулю, для преобразования адреса используется так называемая глобальная таблица дескрипторов **GDT** (**Global Descriptor Table**), в противном случае - локальная таблица дескрипторов **LDT** (**Local Descriptor Table**).

Таблица GDT - единственная в системе. Обычно в ней находятся описания сегментов операционной системы. Таблиц LDT может быть много. Эти таблицы содержат описания сегментов программ, работающих под управлением ОС, т.е. отдельных задач. В каждый данный момент времени процессор может использовать только одну таблицу LDT.

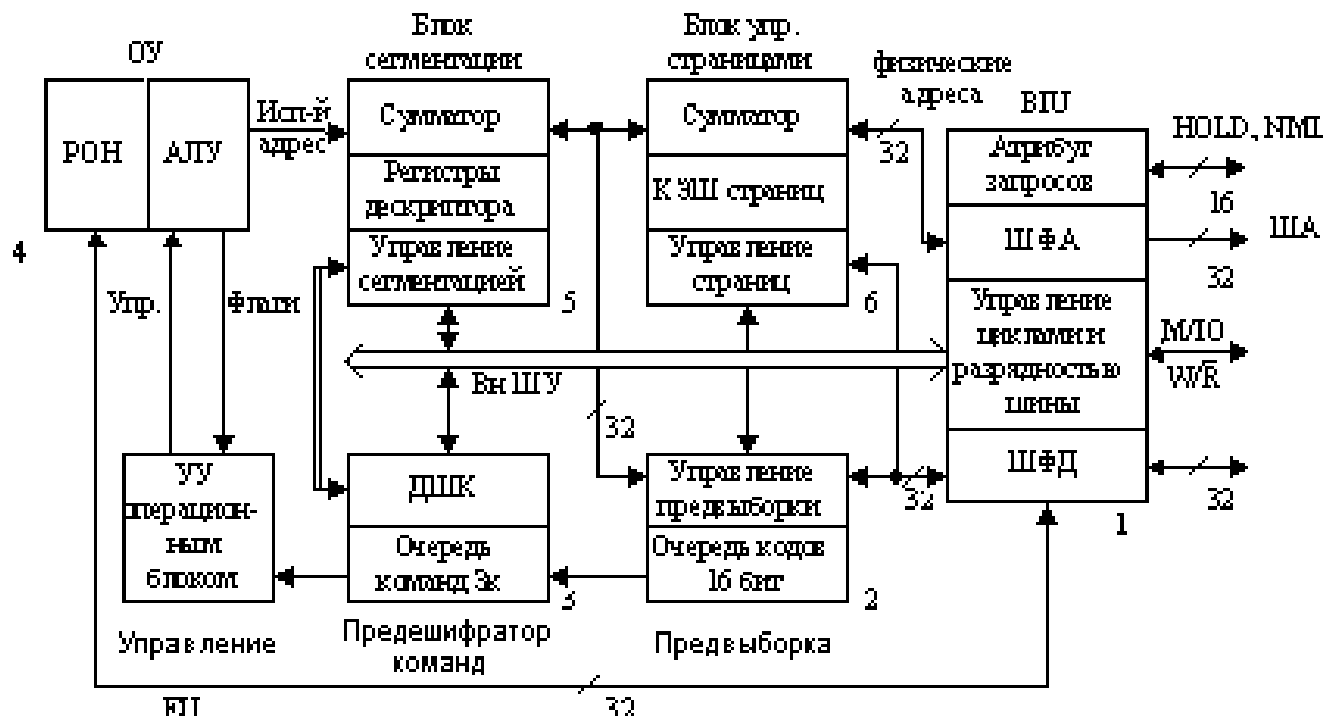
## РЕГИСТРЫ ПРОЦЕССОРА 80286



TSS определяется специальным дескриптором. Такой дескриптор является системным объектом и может находиться только в GDT. Для переключения на определенную задачу, нужно в регистр задачи TR занести селектор TSS. Дескриптор TSS в свою очередь определяет адрес сегмента состояния задачи.



## АРХИТЕКТУРА 32-РАЗРЯДНЫХ ПРОЦЕССОРОВ



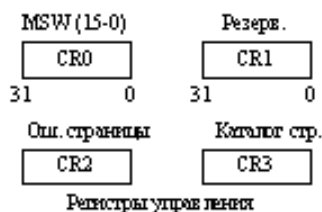
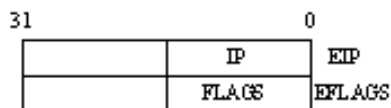
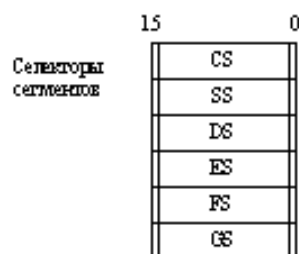
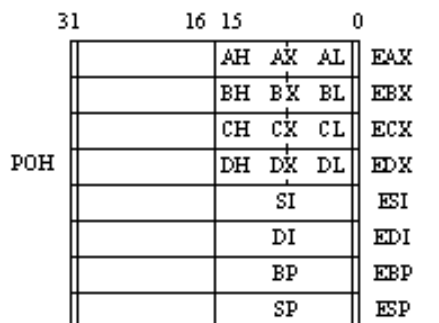
**Центральный процессор** включает в себя операционное ОУ и управляющее УУ устройства. ОУ состоит из АЛУ и восьми 32-разрядных РОН. В АЛУ введен *64-разрядный сдвигатель*, используемый при быстрых арифметических и циклических сдвигах, умножении и делении. В связи с этим 32-разрядное умножение выполняется менее чем за 1 мкс.

Имеется два режима работы:

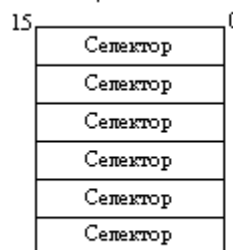
В *реальном режиме* МП 386 работает как очень быстрый 8086, но при необходимости с расширением разрядности операндов и адресов до 32.

В *защищенном режиме* могут осуществляться переключения и выполнения нескольких задач, предназначенных для режима виртуального МП 8086.

## РЕГИСТРЫ 32-РАЗРЯДНОГО ПРОЦЕССОРА



Регистры сегментов



Регистры дескрипторов



Добавлены флаги:

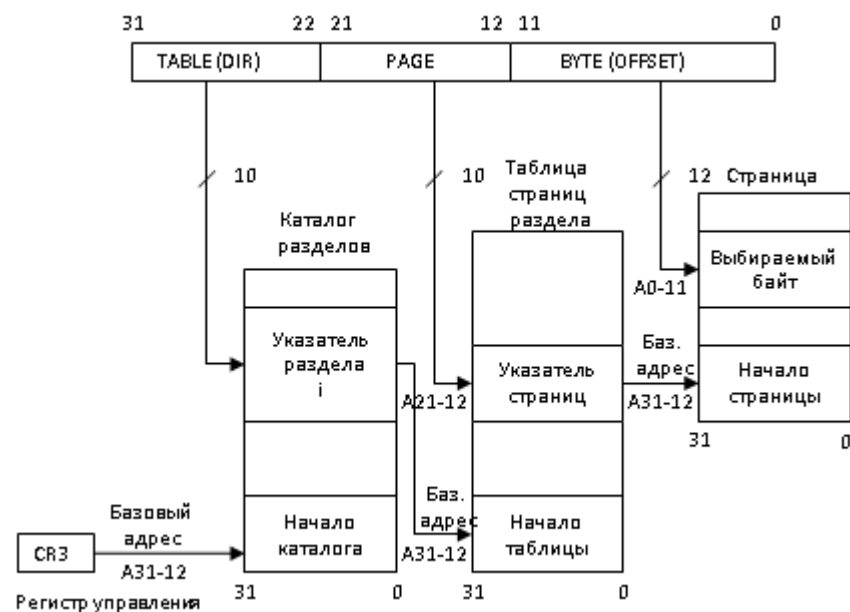
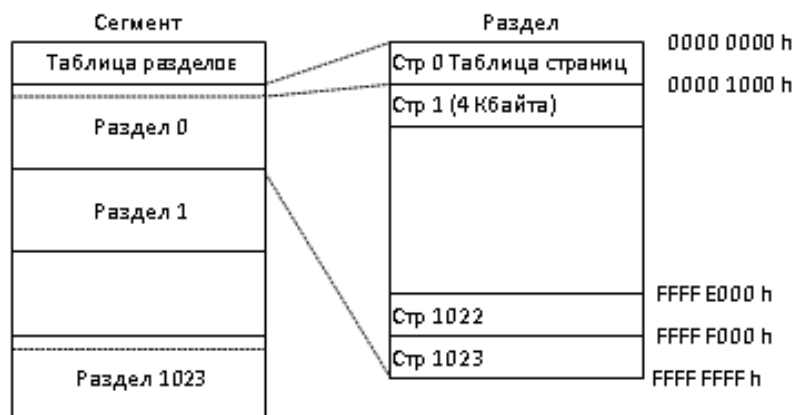
**IOPL** – уровень привилегии ввода – вывода (биты 12 – 13);

**NT** – вложенная задача (бит 14);

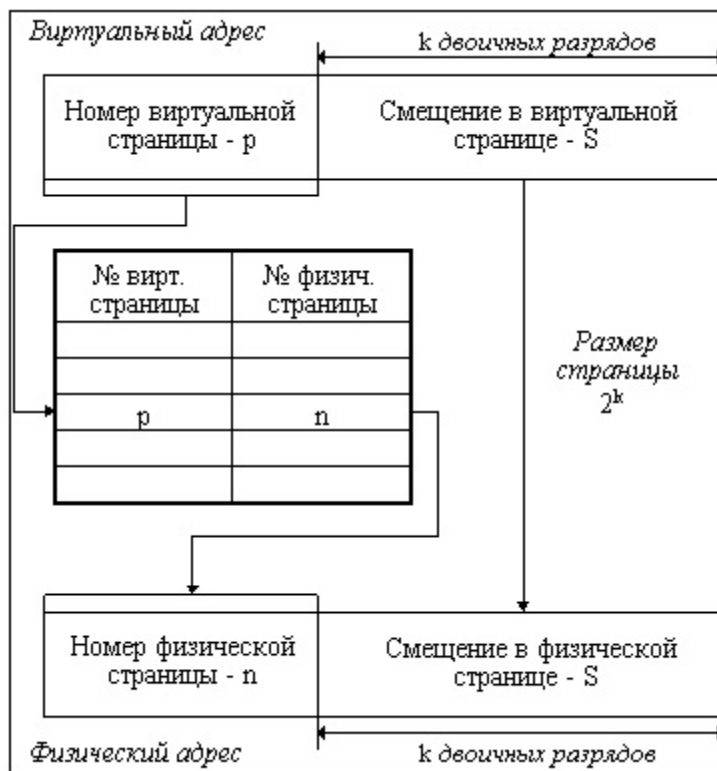
**RF** – флаг итога (бит 16) используется в пошаговом режиме при отладке. Если бит установлен, то любая ошибка отладки в следующей команде игнорируется;

**VM** – виртуальный режим 8086 (бит 17).

## СТРАНИЧНАЯ ОРГАНИЗАЦИЯ ПАМЯТИ



## СХЕМА ПРЕОБРАЗОВАНИЯ АДРЕСОВ В 32-РАЗРЯДНЫХ ПРОЦЕССОРАХ

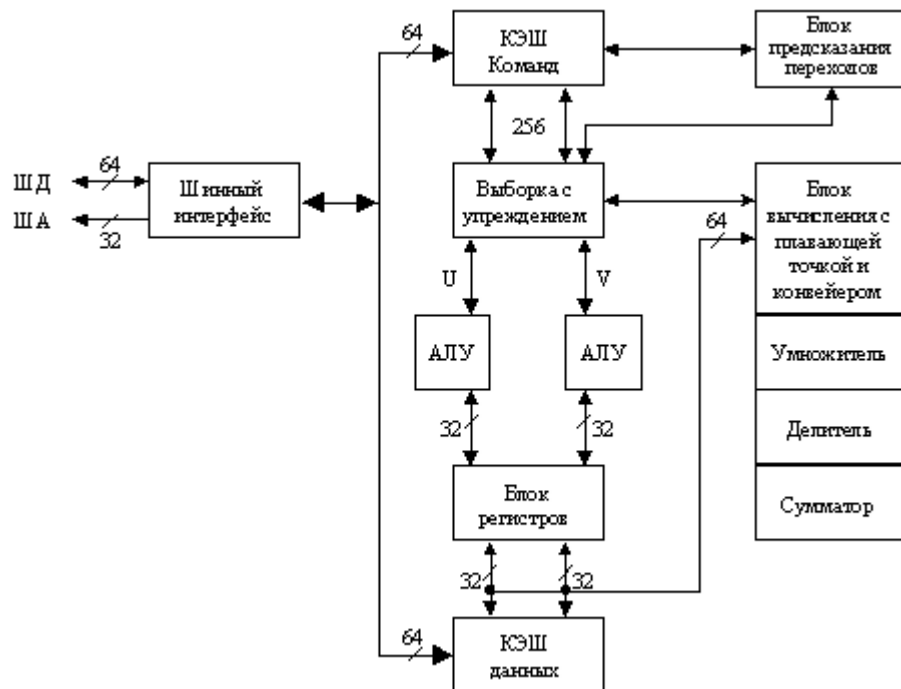


## СУПЕРСКАЛЯРНЫЕ ПРОЦЕССОРЫ

- 1) Одновременное выполнение нескольких команд.
- 2) Конвейерная обработка команд.
- 3) Разделение памяти на память команд и память данных.
- 4) RISC команды (*reduced instruction set computer* — компьютер с набором коротких (простых, быстрых) команд) — быстродействие процессора увеличивается за счёт упрощения инструкций, чтобы их декодирование было более простым, а время выполнения — меньшим.

Такты→	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Команда 1	ВК ↔	ДК ↔	ВА ↔	ВО ↔	ИК ↔	ЗР ↔								
Команда 2		ВК ↔	ДК ↔	ВА ↔	ВО ↔	ИК ↔	ЗР ↔							
Команда 3			ВК ↔	ДК ↔	ВА ↔	ВО ↔	ИК ↔	ЗР ↔						
Команда 4				ВК ↔	ДК ↔	ВА ↔	ВО ↔	ИК ↔	ЗР ↔					
Команда 5					ВК ↔	ДК ↔	ВА ↔	ВО ↔	ИК ↔	ЗР ↔				
Команда 6						ВК ↔	ДК ↔	ВА ↔	ВО ↔	ИК ↔	ЗР ↔			
Команда 7							ВК ↔	ДК ↔	ВА ↔	ВО ↔	ИК ↔	ЗР ↔		
Команда 8								ВК ↔	ДК ↔	ВА ↔	ВО ↔	ИК ↔	ЗР ↔	
Команда 9									ВК ↔	ДК ↔	ВА ↔	ВО ↔	ИК ↔	ЗР ↔

## АРХИТЕКТУРА ПРОЦЕССОРА PENTIUM



Процессор Pentium выполнен по 0.8 мк – BICMOS технологии и реализован на 3.1 млн. транзисторах. Корпус микросхемы имеет 352 вывода. Тактовая частота 66 МГц. В процессоре на одном кристалле реализованы:

- Два блока работы с целыми числами;
- Блок с плавающей точкой;
- Отдельное 8 Кбайтовое устройство КЭШ – памяти программ и отдельное 8 Кбайтовое устройство КЭШ – памяти данных;
- Блок предобработки переходов;
- Блок управления памятью.

Технология Intel – MMX (*Multi Media eXtention*), поддерживается в процессорах Pentium MMX и представляет из себя набор новых команд, созданных для повышения эффективности работы современных Multimedia – приложений и приложений связи. **SIMD** (**S**ingle **I**nstruction stream / **M**ultiple **D**ata stream) – одиночный поток команд и множественный поток данных.

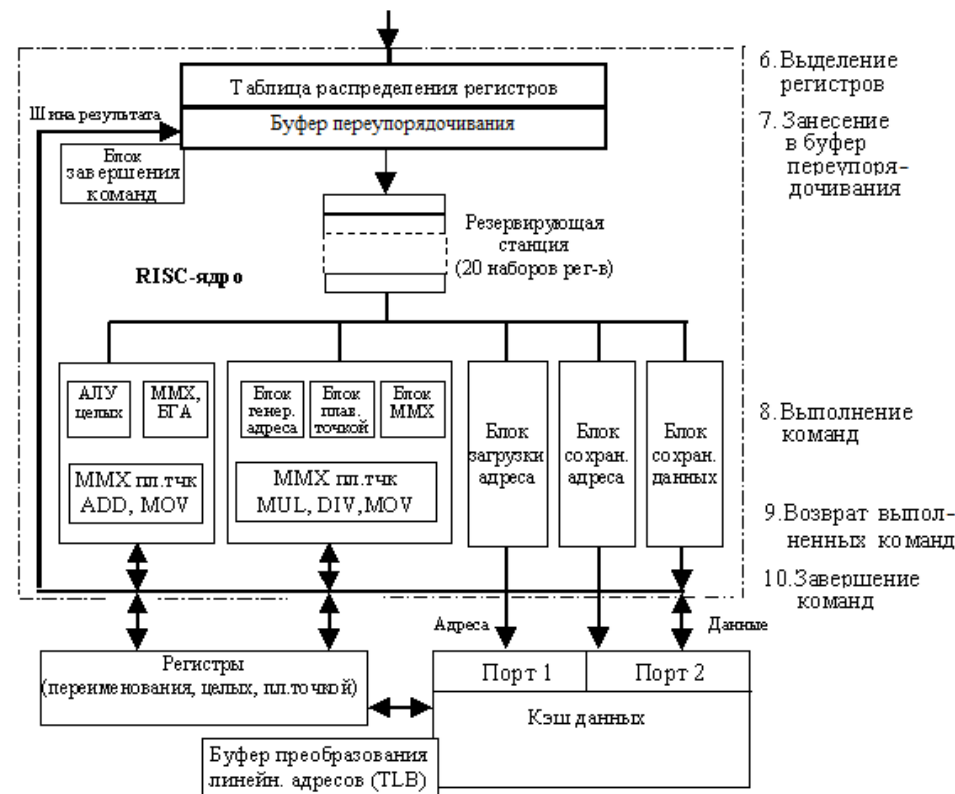
Добавлено:

- восемь 64-разрядных регистров (MM0 – MM7);
- 4 типа данных (упакованные байты, упакованные слова, упакованные двойные слова и четырехсловный тип);
- В новом MMX – процессоре введен дополнительный набор из 57 команд, связи с этим их общее число достигло 277.

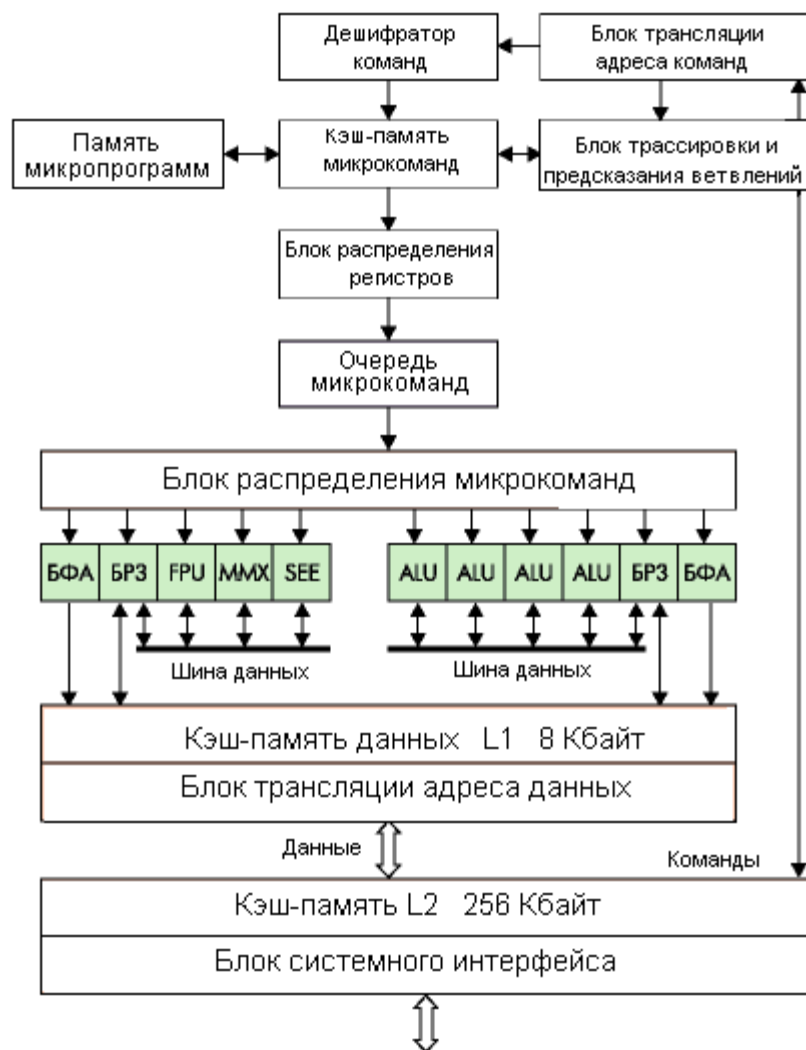
## Процессор PENTIUM III

### Ступени конвейера

1. Выборка с упреждением
2. Опред. длины команды
3. Распределение команд по ПК
4. Преобразование кодов
5. Буферизация команд



## АРХИТЕКТУРА ПРОЦЕССОРА PENTIUM 4

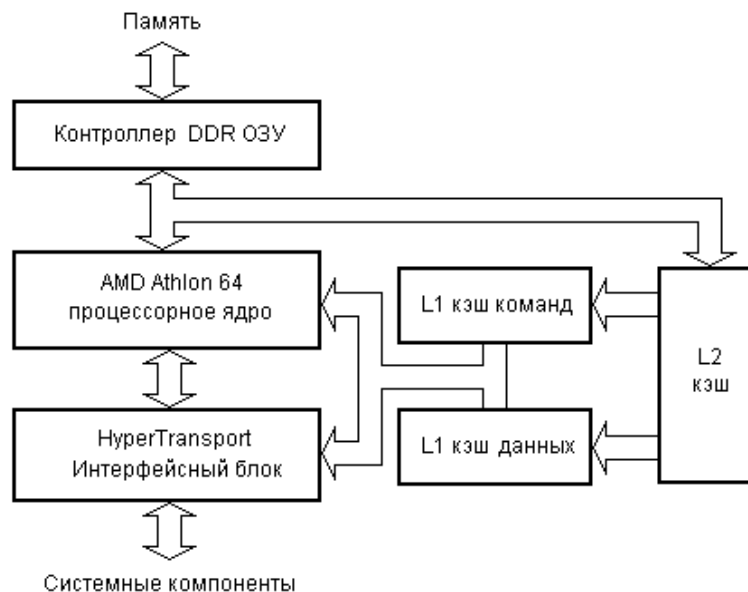


Отличительными особенностями Pentium 4 от его предшественников являются:

- 1) введение гиперконвейера (Hyper Pipelining), позволяющего размещать и выполнять одновременно 20 микроопераций;
- 2) усовершенствование динамического исполнения позволяет в единицу времени выполнять в 3 раза больше микроопераций;
- 3) частота процессора превысила 3 ГГц, а системной шины 800 МГц;
- 4) в два раза повышена скорость выполнения операций в АЛУ целых чисел;
- 5) введено дополнительно 144 новых мультимедийных команд.



## АРХИТЕКТУРА 64-РАЗРЯДНЫХ ПРОЦЕССОРОВ



Структура процессора Athlon 64 фирмы AMD

Шина **HyperTransport (HT)**— это двунаправленная последовательно/параллельная компьютерная шина, с высокой пропускной способностью и малыми задержками. Она работает на частотах от 200 МГц до 2,6 ГГц. HyperTransport поддерживает автоматическое определение ширины шины, от 2-х битных линий до 32-х битных. Шина HyperTransport основана на передаче пакетов.

## 64-РАЗРЯДНЫЕ ПРОЦЕССОРЫ

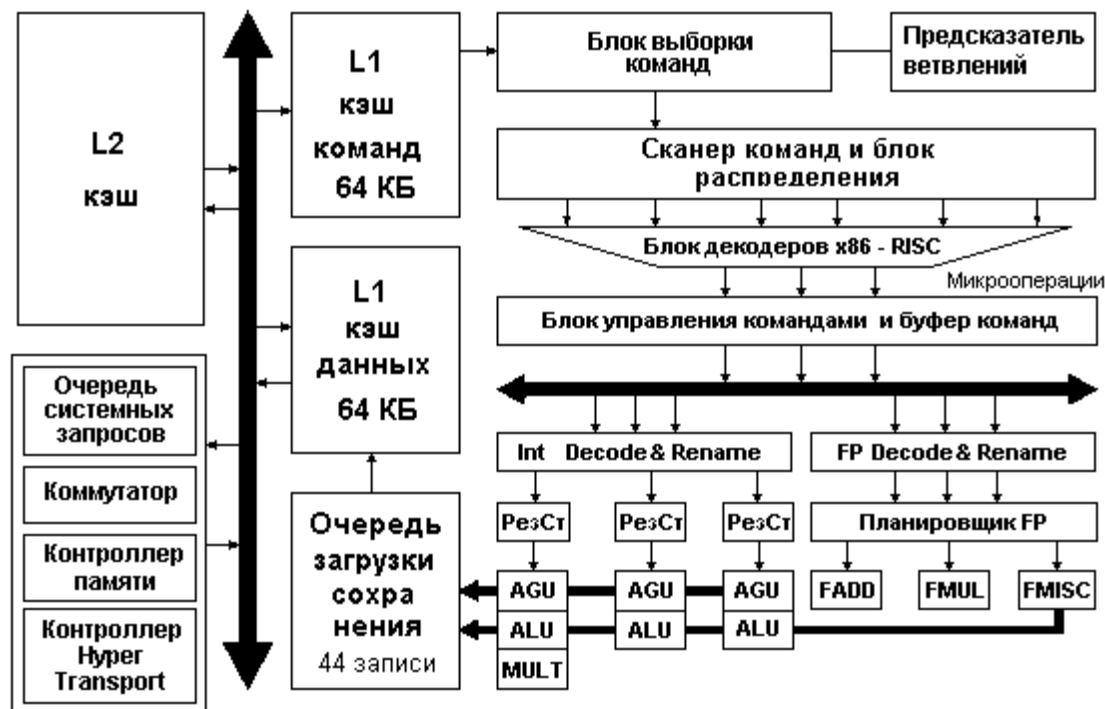
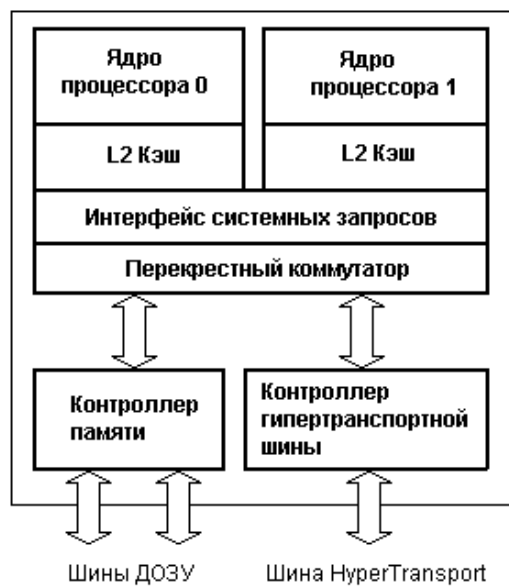
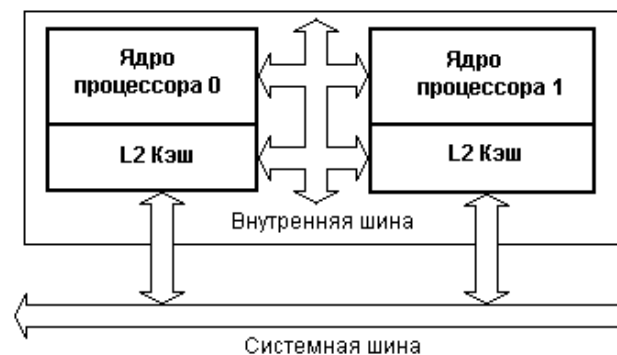


Схема процессорного ядра Athlon 64

## Многоядерные процессоры AMD и Intel



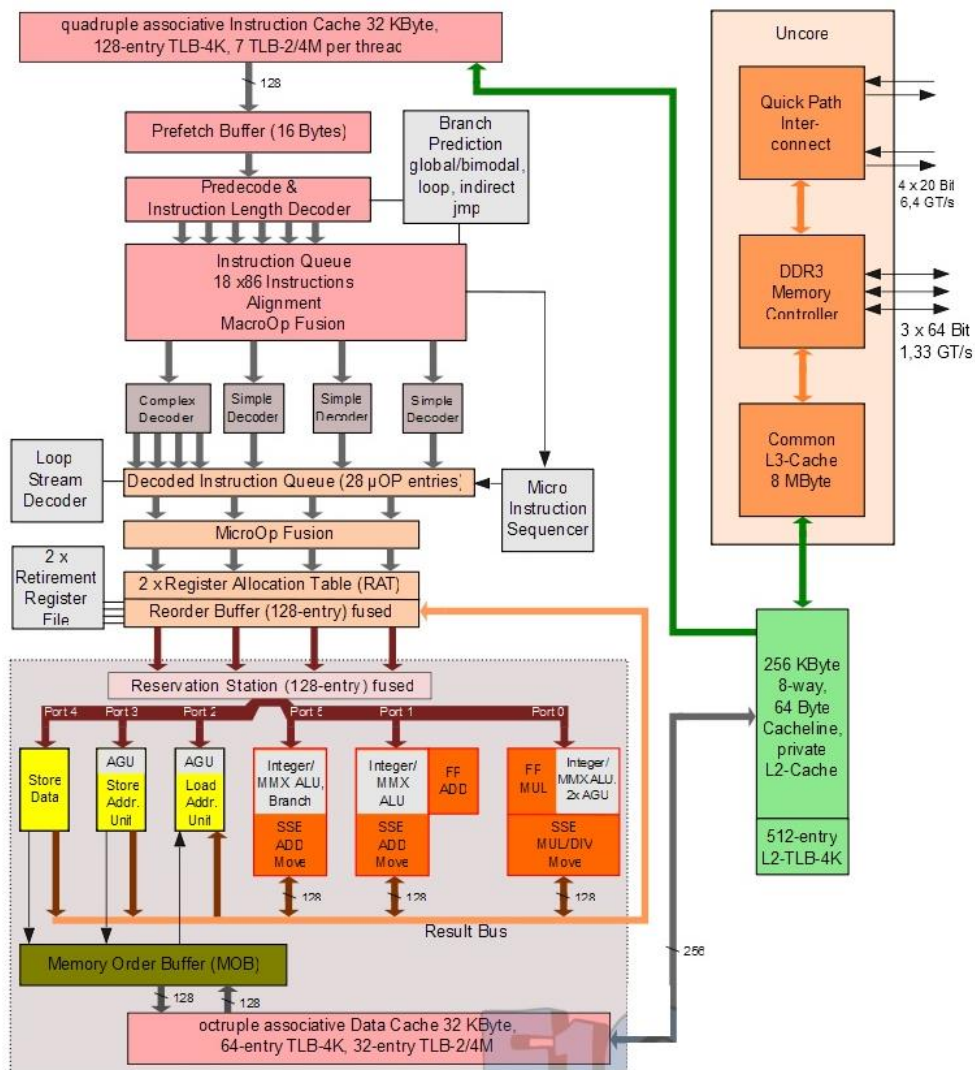
а)



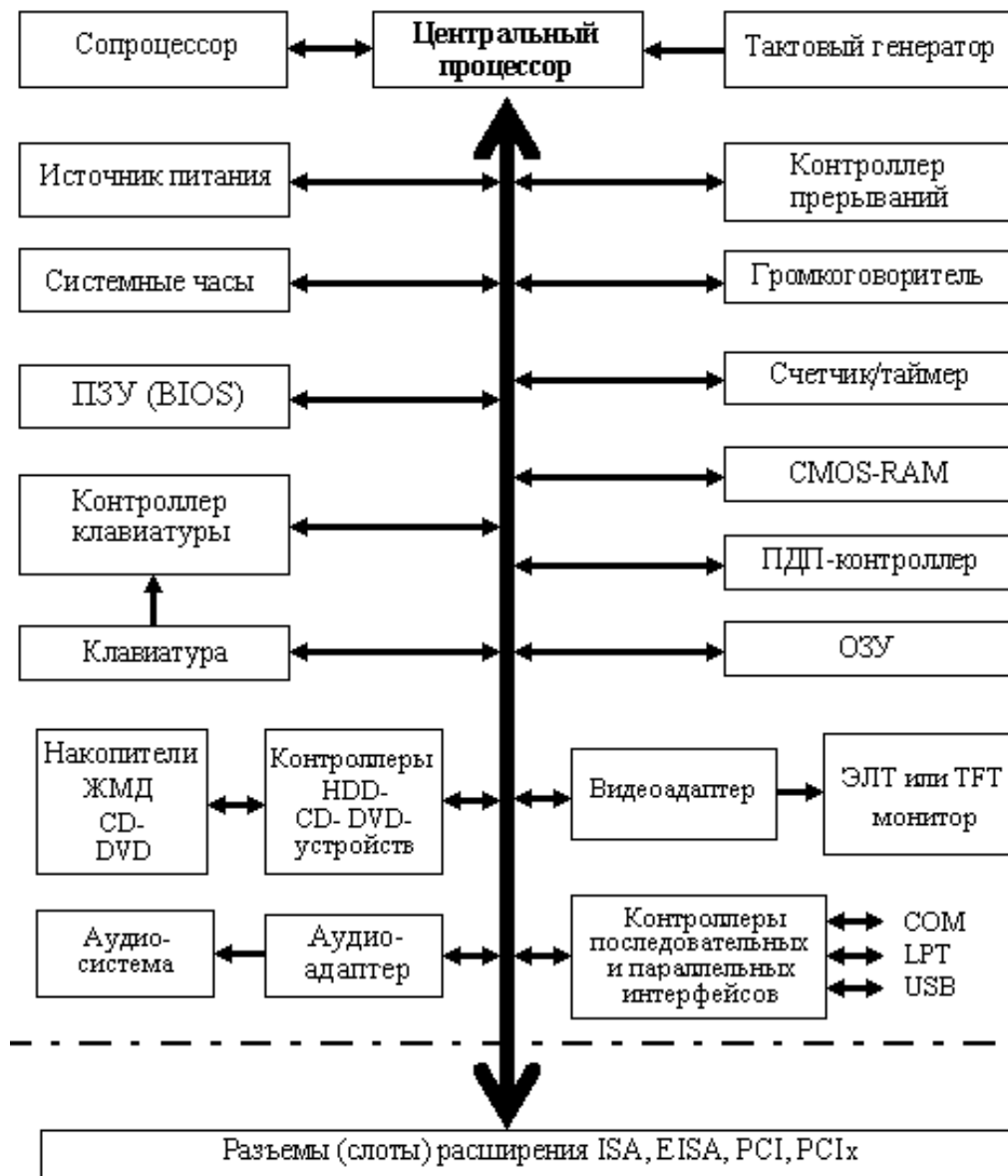
б)

Структура двухядерного процессора типа **Athlon 64 X2** (а) и **Pentium D9xx** (б)

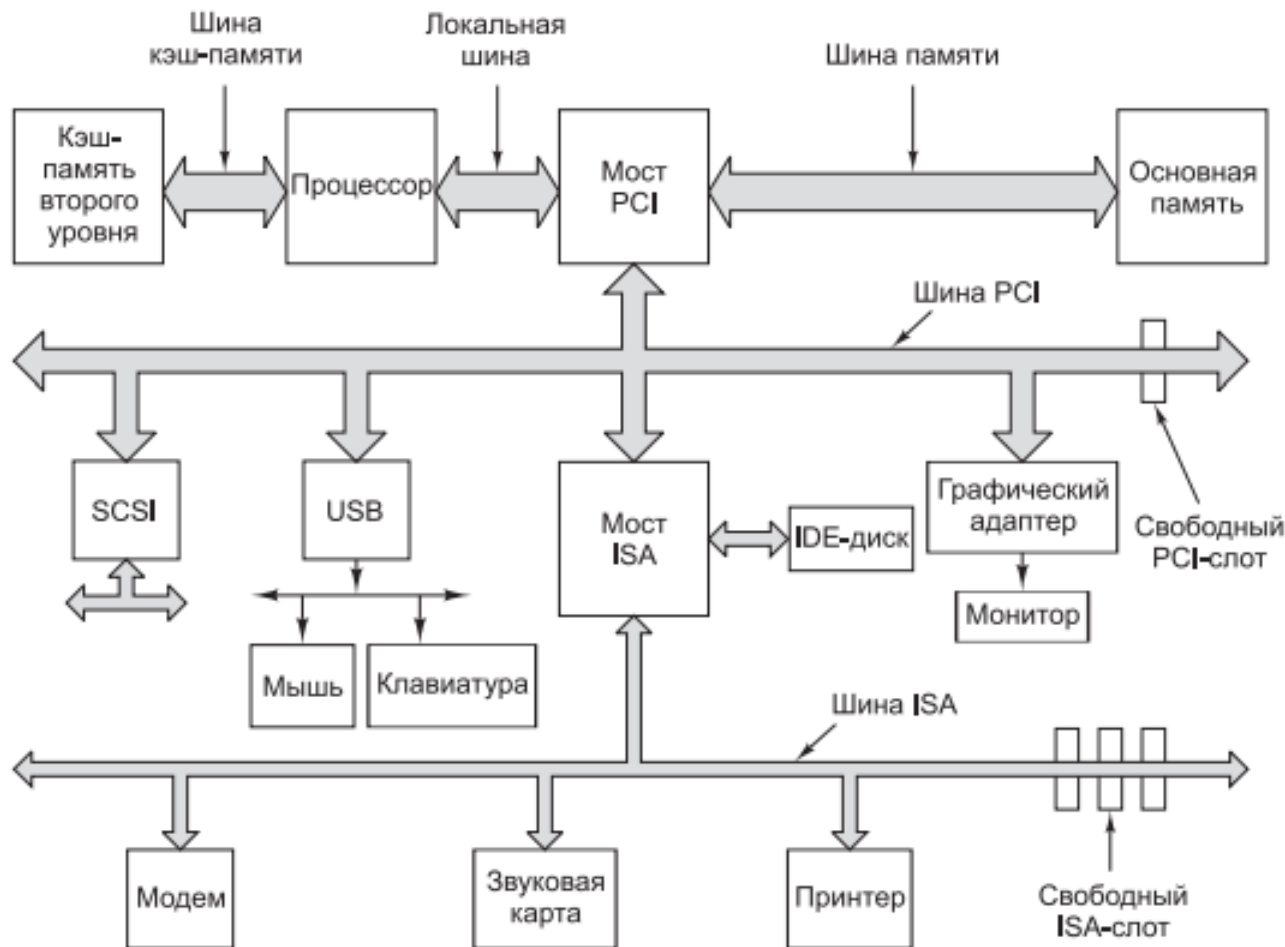
## Ядро процессора AMD Core i7



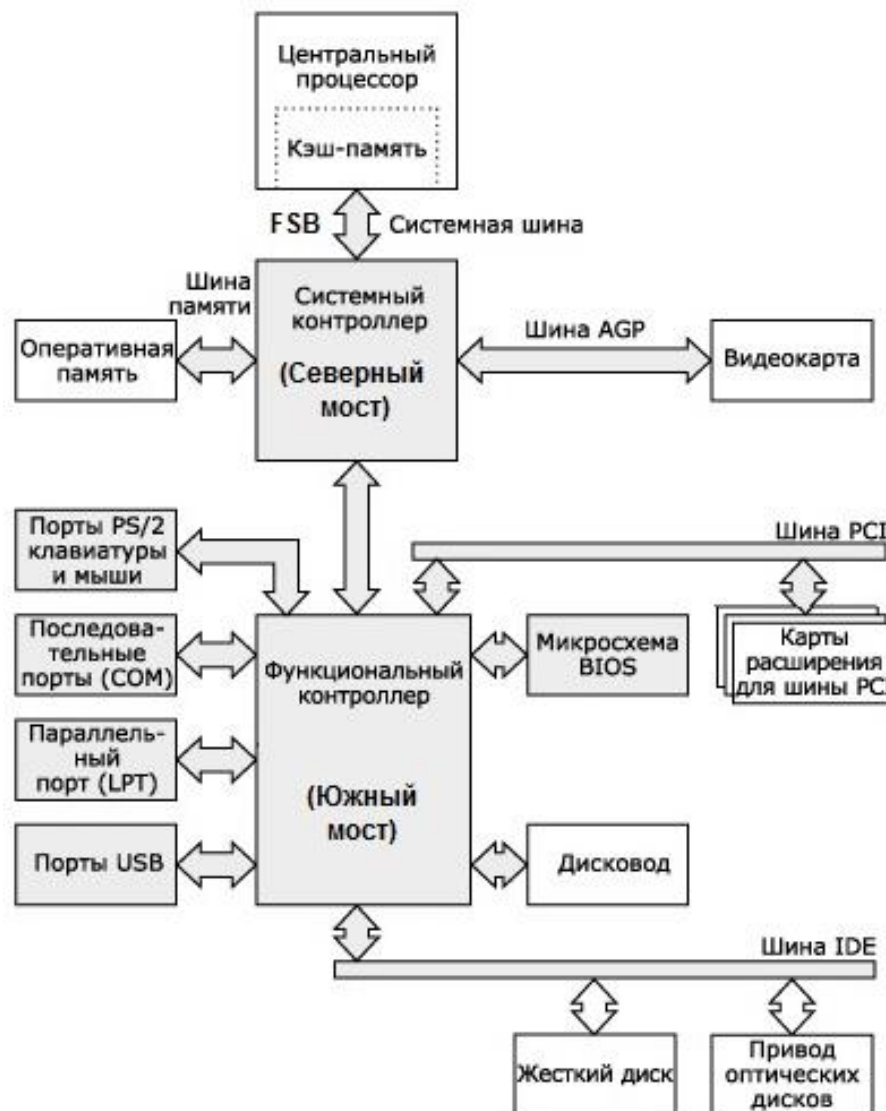
## Функциональная схема ПЭВМ



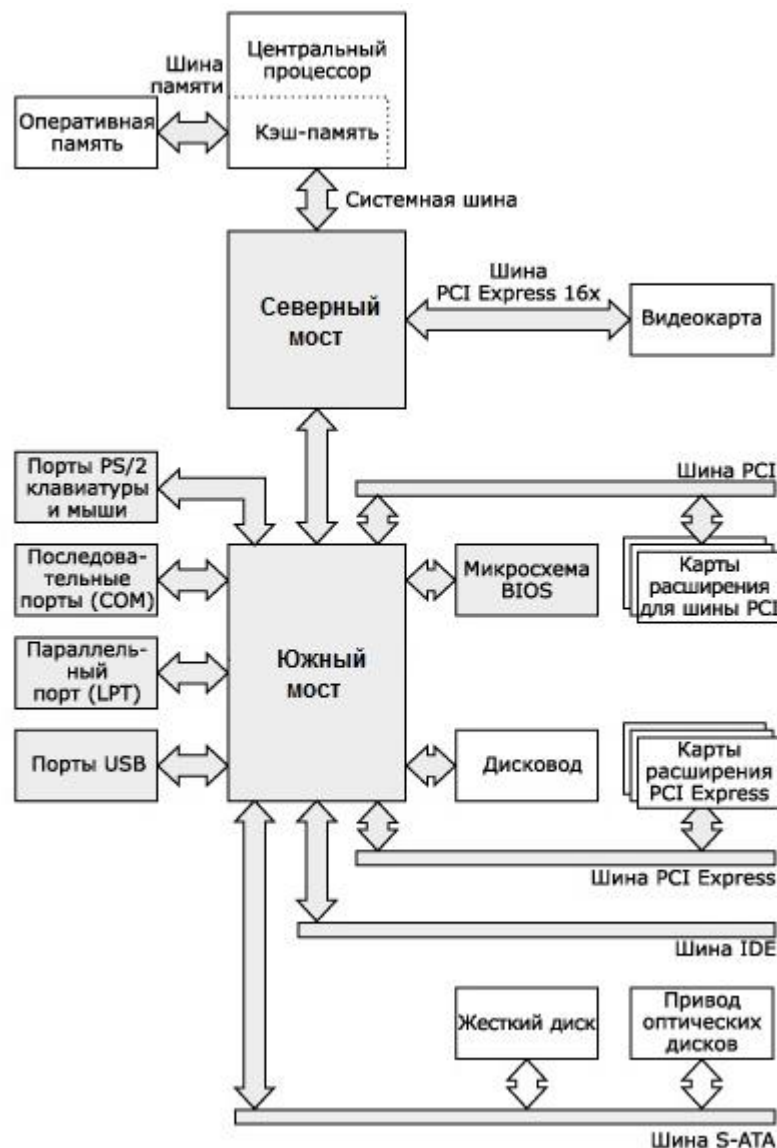
## Архитектура первых ПК на базе процессоров Pentium



## Компьютер на основе системного и функционального контроллеров



## Компьютер на основе послед-х шин и встроенного контроллера ДОЗУ



**FSB (Front Site Bus)** –  
50 – 400 МГц

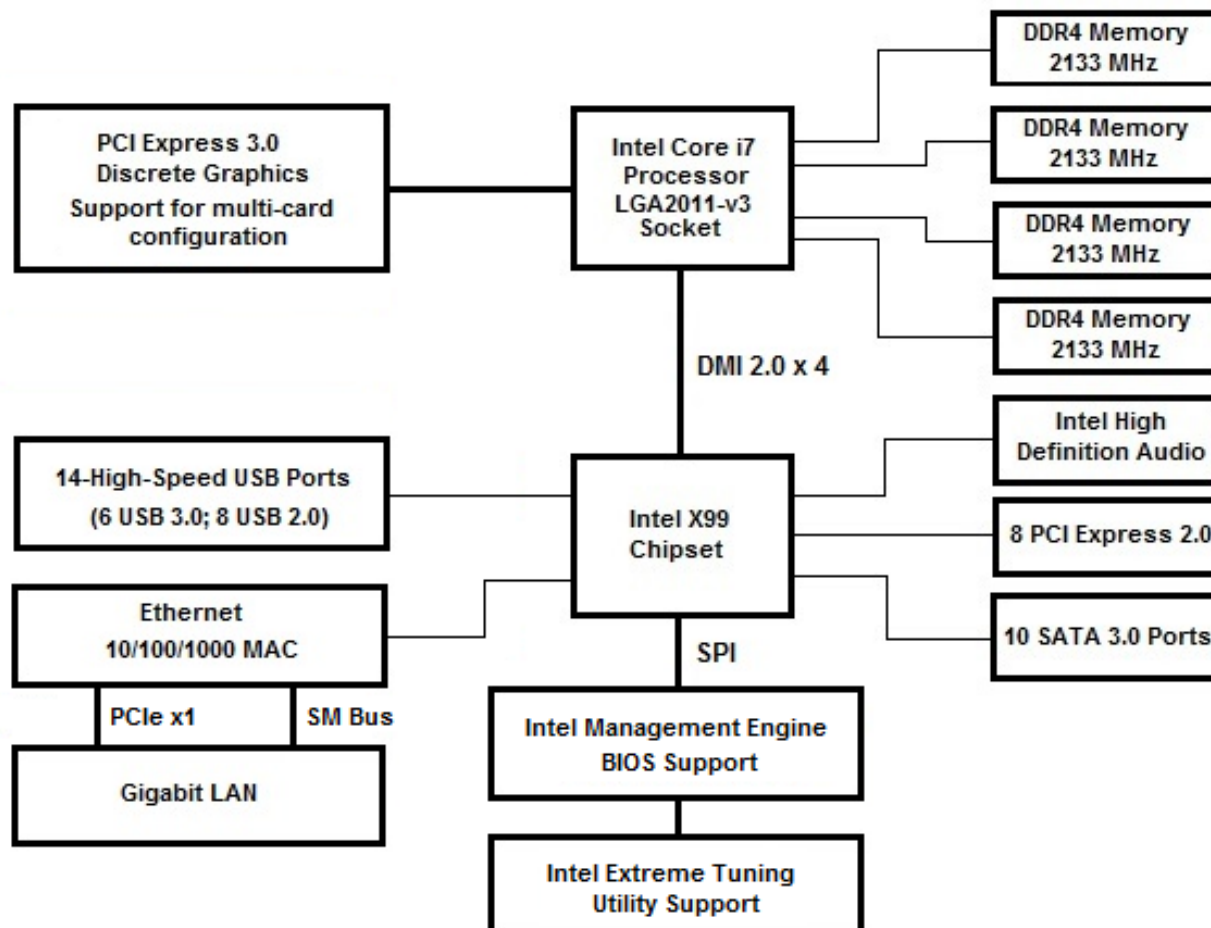
PCI Express:  
PCIe-1.0 – 2 Гбит/с  
PCIe-2.0 – 4 Гбит/с  
PCIe-3.0 – 8  
PCIe-4.0 – 16  
симплекс

IDE – 32-58 Мгбит/с;

SATA-1 - 150 Мгбит/с  
SATA-2 - 300 Мгбит/с;  
SATA-3 - 600 Мгбит/с.



## Компьютер с однокристальным системным контроллером



## Распределение DOS-адресного пространства ПЭВМ

Размер и адрес	Название области памяти	
до 15 Мб 10FFF0h	XMSРасширенная память	Расширенная память (XMS-eXtended Memory Area)
64 Кб 100000h	HMA Высокая память	Высокая память (HMA-High Memory Area)
128 Кб E0000h	ПЗУ BIOS	Область верхней памяти (UMA - Upper Memory Area)
64 Кб D0000h	Верхняя память (UMB-Upper Memory Block)	
64 Кб C0000h	ПЗУ - расширения BIOS	
32 Кб B8000h	Текстовый буфер EGA	
32 Кб B0000h	UMB Верхняя память	
64 Кб A0000h	Графический буфер EGA	
640 Кб	Command.com	7FFF - 9FFFh Stack DOS  Стандартная память  (CMA-Conventional Memory Area)
	Свободная память для прикладных программ	
	Command.com (резидентная часть)	
	Загружаемые драйверы	
0700h	IO.SYS и MSDOS.SYS	
0500h	Область данных DOS	
0400h	Область данных BIOS	
0000h	Векторы прерываний	

MS DOS: набор программ обработки прерываний, в частности прерывания INT 21H

Область данных BIOS:  
входной буфер клавиатуры с указателями;  
адреса последовательных и параллельных портов;  
данные настройки видеосистемы (форма курсора, его нахождение на экране, текущий видеорежим и т.п.);  
ячейки для отсчета текущего времени.

## Распределение Windows-адресного пространства ПЭВМ

Низкоуровневые компоненты Windows (Модули ядра. Работают в кольце защиты с номером 0)	4 Гбайт
	3 Гбайт
DLL Win32, другие совместно используемые процессы	2 Гбайт
Прикладные программы Win32 Виртуальные машины Win16	4 Мбайт
Почти не используется	1 Мбайт
Память MS-DOS	0 Мбайт

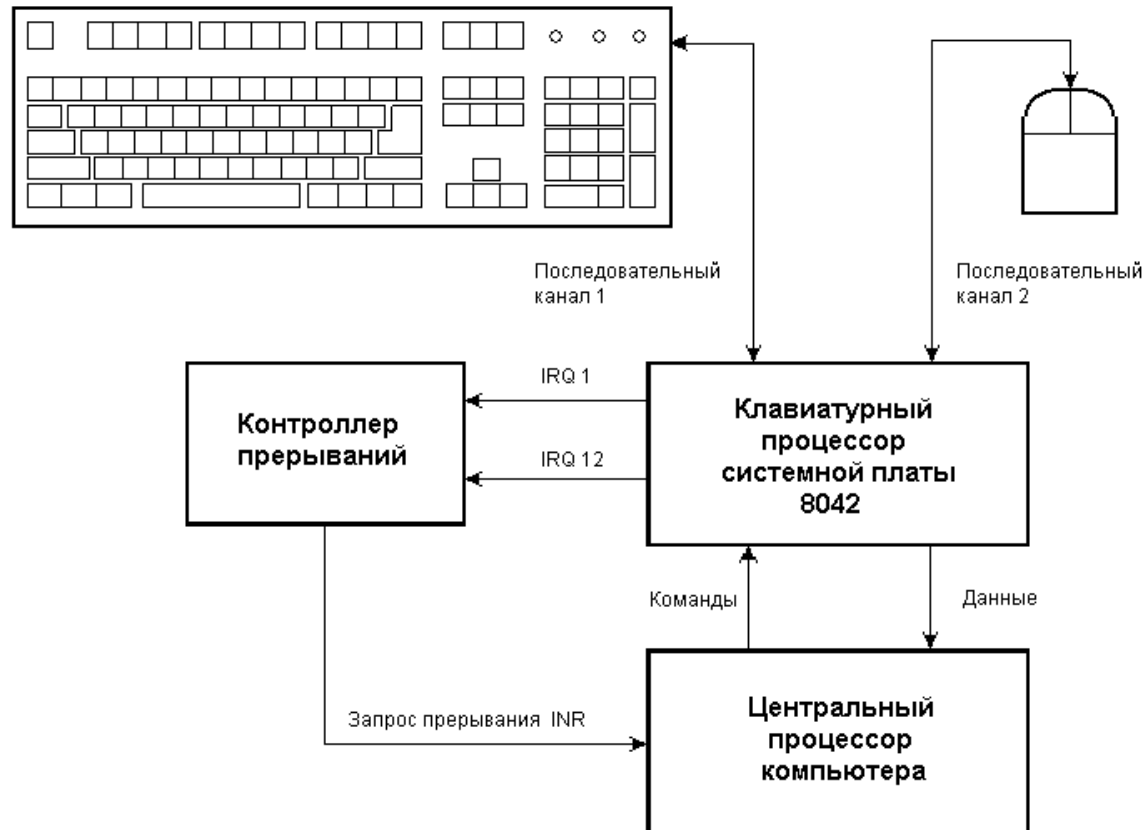
## Особенности архитектуры серверных компьютеров

- 1) память обеспечивает повышенную устойчивость к сбоям за счет введения коррекции ошибок ECC (Error Checking and Correction,);
- 2) вводится дублирование процессоров;
- 3) используется горячее резервирование (Hot-swap) важных компонентов.
- 4) дублирование накопителей на жестких магнитных дисках в составе массива RAID, контроллеров дисков и блоков питания;
- 5) дублирование групп вентиляторов, обеспечивающих охлаждение компонентов сервера.

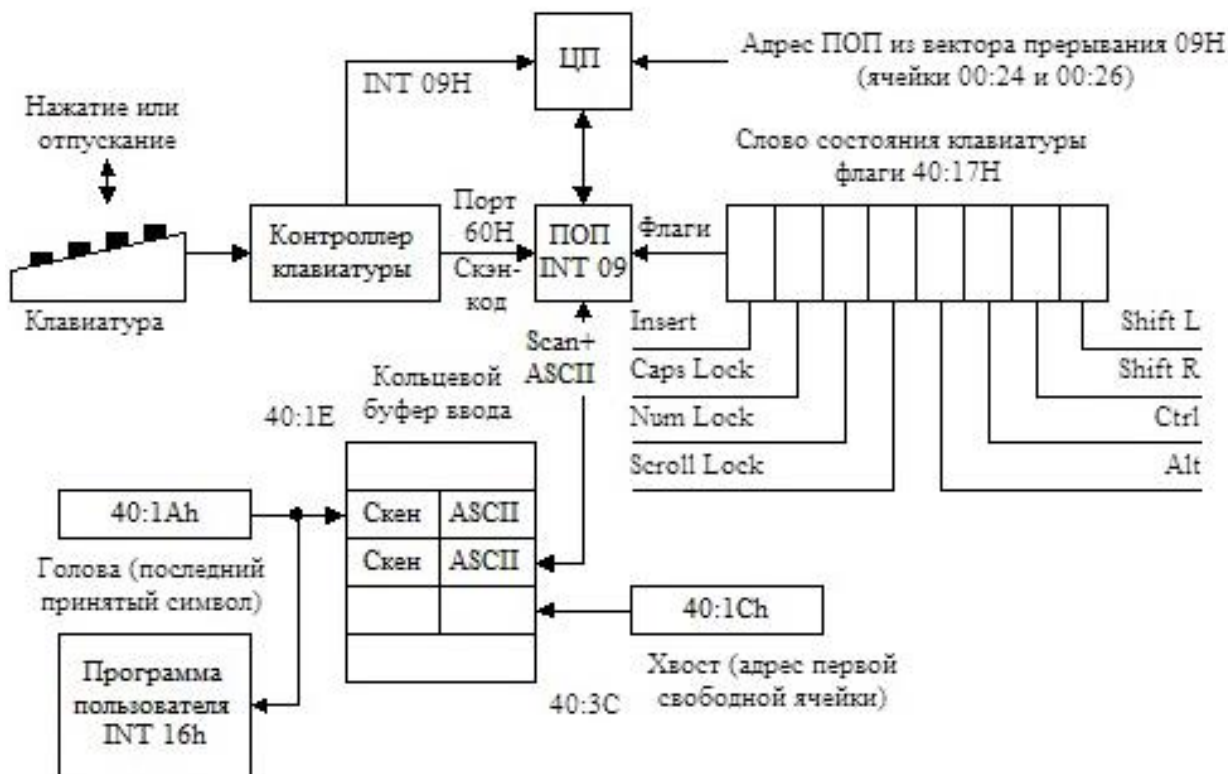
Аппаратный мониторинг:

- 1) датчики температуры контролируют температурные режимы всех процессоров, модулей памяти, температуру в отсеках с установленными жёсткими дисками; электронные счётчики импульсов, встроенные в вентиляторы, выполняют функции тахометров и позволяют, в зависимости от температуры, регулировать скорость их вращения;
- 2) постоянный контроль напряжения питания компонентов сервера позволяет сигнализировать об эффективности работы блоков питания;
- 3) сторожевой таймер не позволяет остаться незамеченным зависанию системы, автоматически производя принудительную перезагрузку сервера.

## Клавиатура ПЭВМ и ее взаимодействие с процессором



## Схема взаимодействия клавиатуры с процессором



По таблице трансляции скен – кодов в коды ASCII формируется двухбайтовый код. Старший байт содержит **скен – код**, а младший – **код ASCII**.

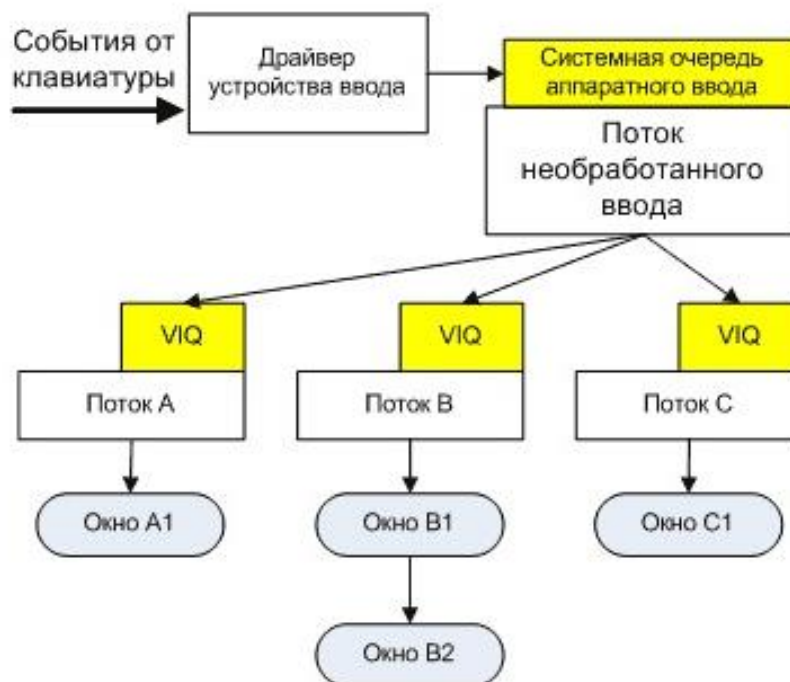
## Клавиатура ПЭВМ. Клавиатурные функции BIOS.

Функция 0	<p>– прочесть текущий символ.</p> <p>В результате выполнения в AL – ASCII код (либо 0). Символ помещается также в буфер клавиатуры в текущую ячейку 041E – 043E.</p>
Функция 1	<p>– проверить готовность.</p> <p>Готов ли текущий символ к передаче в программу. Это связано с различными скоростями ввода и обработки символов. Результат выполнения – состояние флага ZF:</p> <p>ZF = 0 символ не готов</p> <p>ZF = 1 готов</p> <p>Если ZF = 0, то в AL возвращается готовый символ, в AH – scan – code.</p>

Пример ввода с клавиатуры INT21h, программная функция 3Fh:

IN_AREA DB 20 DUP ( ' ' )	;область ввода
MOV AH, 3Fh	;запрос на ввод
MOV BX, 00	;номер для клавиатуры
MOV CX, 20	;максимальное число вводимых байт
LEA DX, IN_AREA	;адрес области ввода
INT 21h	

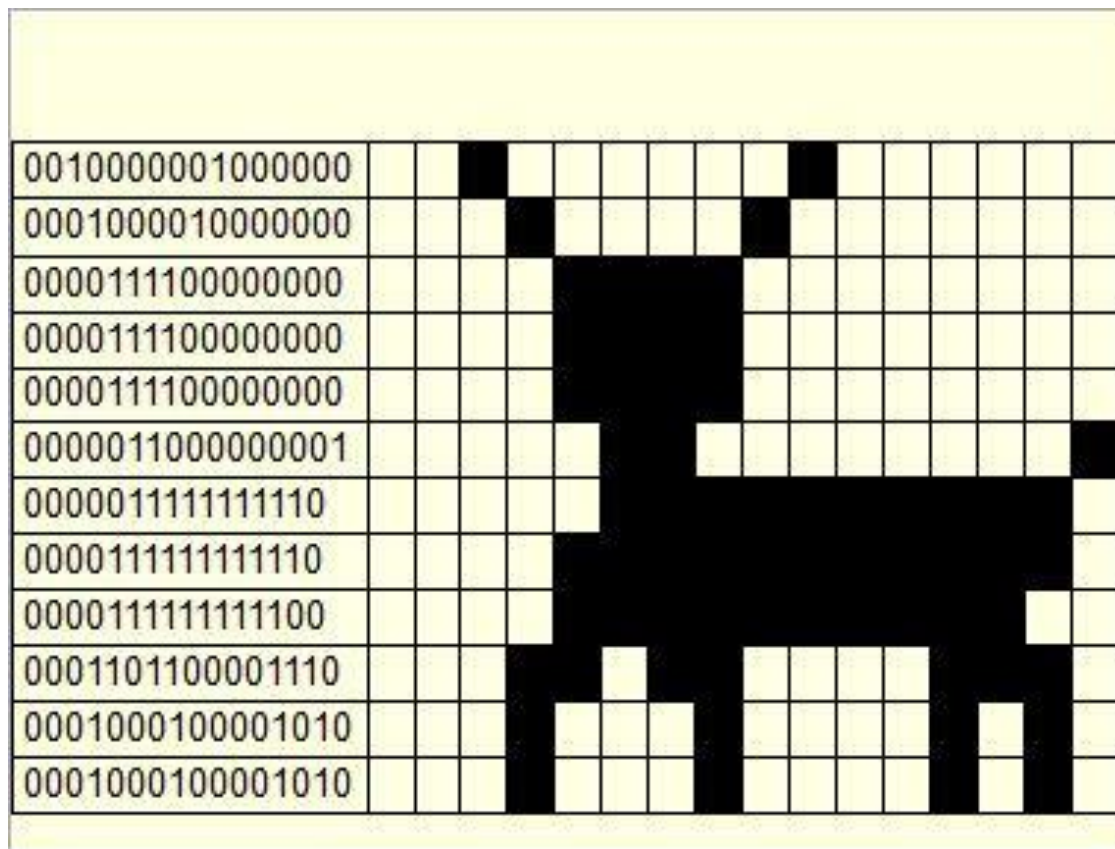
## Клавиатура ПЭВМ. Взаимодействие с Windows.



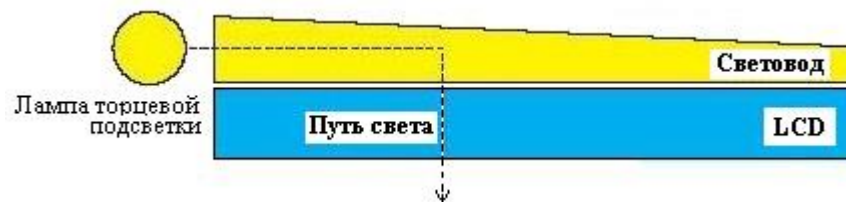
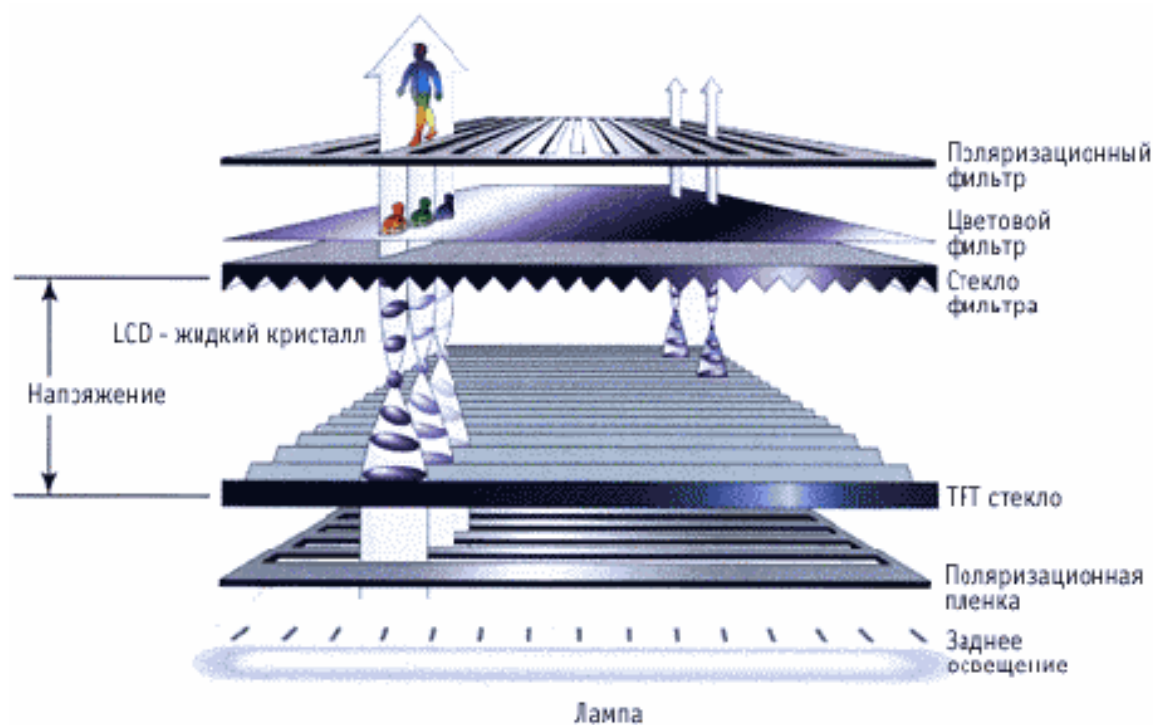
Виртуальная очередь  
потока VIQ –  
Virtualized Input Queue



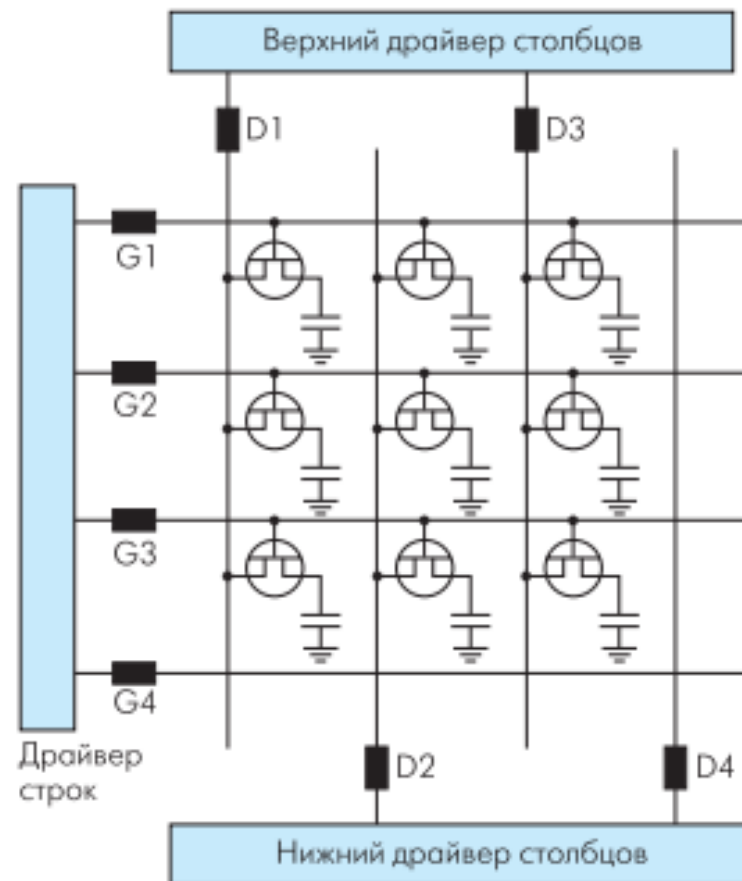
## Видеосистема компьютера. Растровые изображения



## Видеосистема компьютера. TFT-монитор.



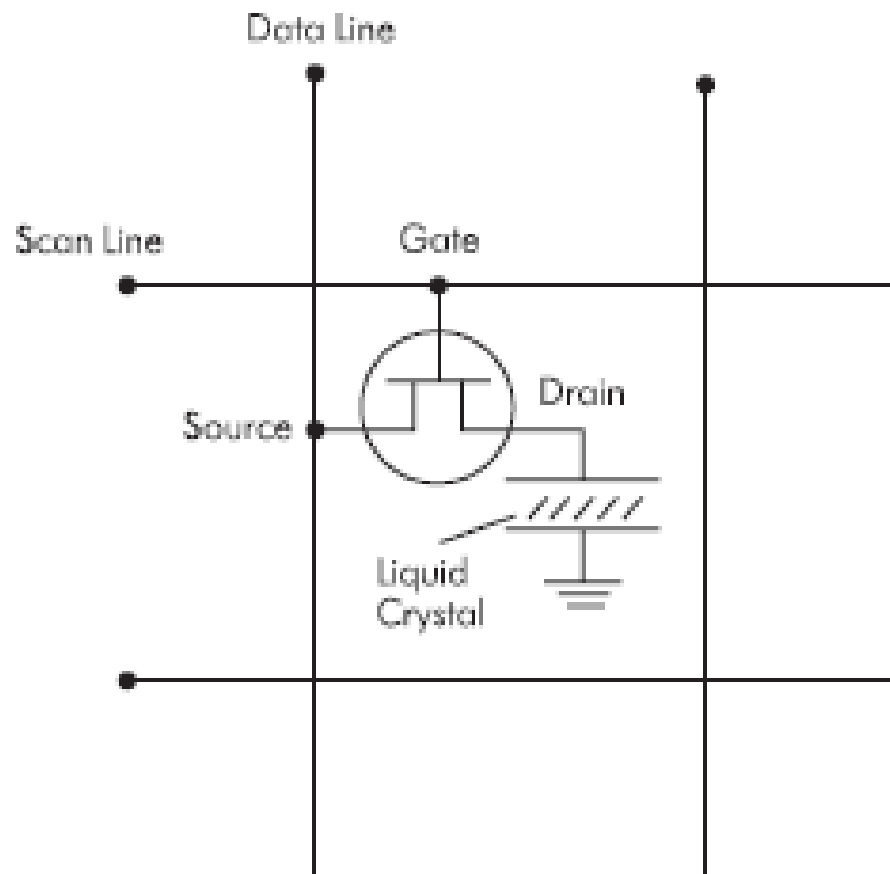
## Видеосистема компьютера. TFT-монитор.



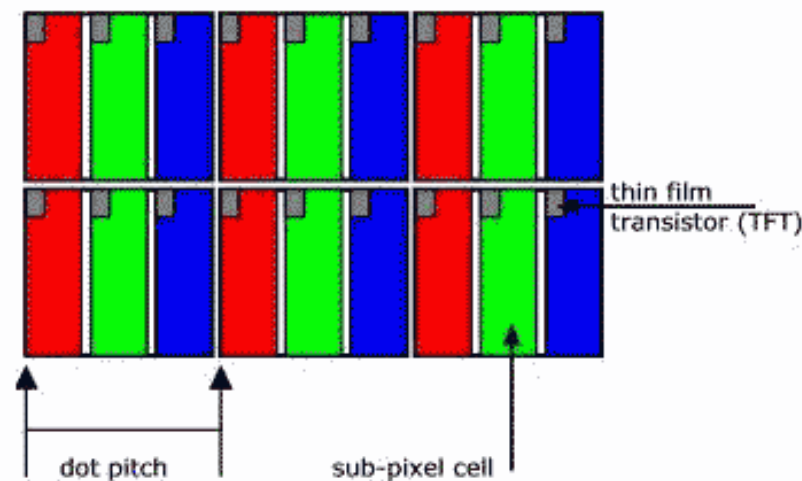
$D_i$  (Data) - шина столбцов

$G_i$  (Gate) - шина строк (управление затворами транзисторов)

## Видеосистема компьютера. Схема ячейки пиксела ЖК-монитора.



## Видеосистема компьютера. Формирование цвета



Stripe

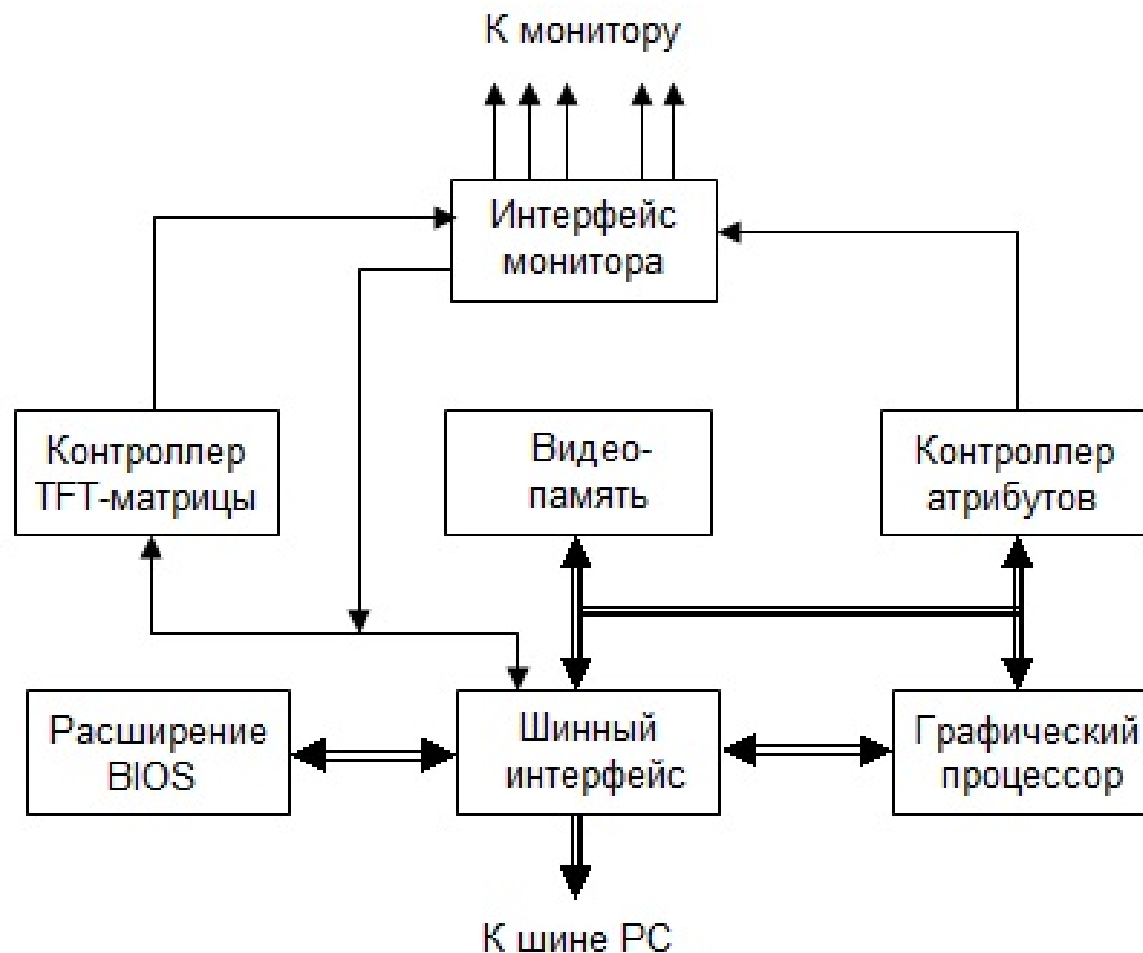


Mosaic



Delta Triad

## Видеосистема компьютера. Схема графического адаптера.



## Видеосистема компьютера. Интерфейсы VGA и SVGA.

**VGA** ( Video Graphics Array) - графическая система для дисплеев персональных компьютеров, разработанная фирмой IBM.

В VGA используются аналоговые сигналы. Разрешение 640×480 пикселей.

Дальнейшая модификация SVGA - **800 × 600** и **1024 × 768** .

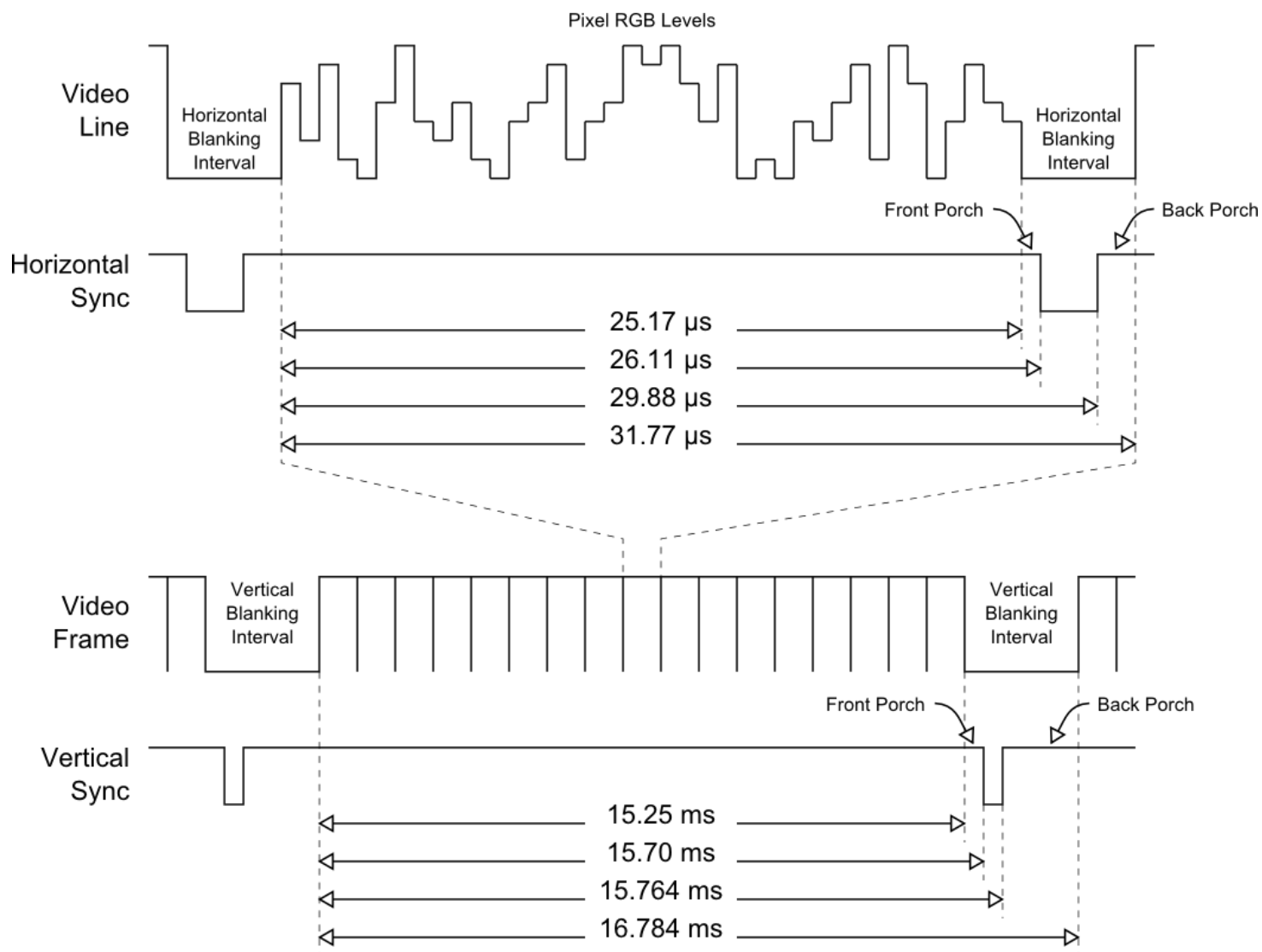


## Видеосистема компьютера. Интерфейсы VGA и SVGA.

Пин	Название	Направление	Описание
1	Красный	>	Красный видео (75 Ом, 0.7 В)
2	Зеленый	>	Зеленый видео (75 Ом, 0.7 В)
3	Синий	>	Синий видео (75 Ом, 0.7 В)
4	RES		Не используется
5	GND	-----	Земля
6	RGND	-----	Земля красного
7	GGND	-----	Земля зеленого
8	BGND	-----	Земля Синего
9	+5V	>	Дополнительные +5В от в/карты
10	SGND	-----	Земля Синхронизации
11	ID0	<	ID монитора Бит 0 (опционально)
12	SDA	<	I2C двунаправленная линия данных
13	HSYNC or CSYNC	>	Горизонтальная синхронизация (или Композитная синхронизация)
14	VSNC	>	Вертикальная синхронизация
15	SCL	<	Тактовая частота 15 SCL I2C в DDC2, Monitor ID3 в DDC1



## Видеосистема компьютера. Сигналы VGA.



## Видеосистема компьютера. Функции BIOS.

Функции BIOS делятся по назначению на семь групп :

Функции контроля видеорежима 0H и 0FH (регистры AH,BX,CX,DX. INT10h):

0 – устанавливать видеорежим (выбирается один из восьми режимов).

Функции управления координатами 01H – 05H:

01 – установить конфигурацию курсора;

02 – установить координаты курсора;

03 – прочитать координаты курсора;

04 – прочитать координаты фотоселектора;

05 – установить активную страницу;

Функции управления окнами 06H и 07H:

06 – прокрутка окна вверх;

07 – прокрутка окна вниз.

Функции записи и чтения символов 08H, 09H, 0AH, 0EH:

08 – прочитать символ и атрибут;

09 – записать символ и атрибут;

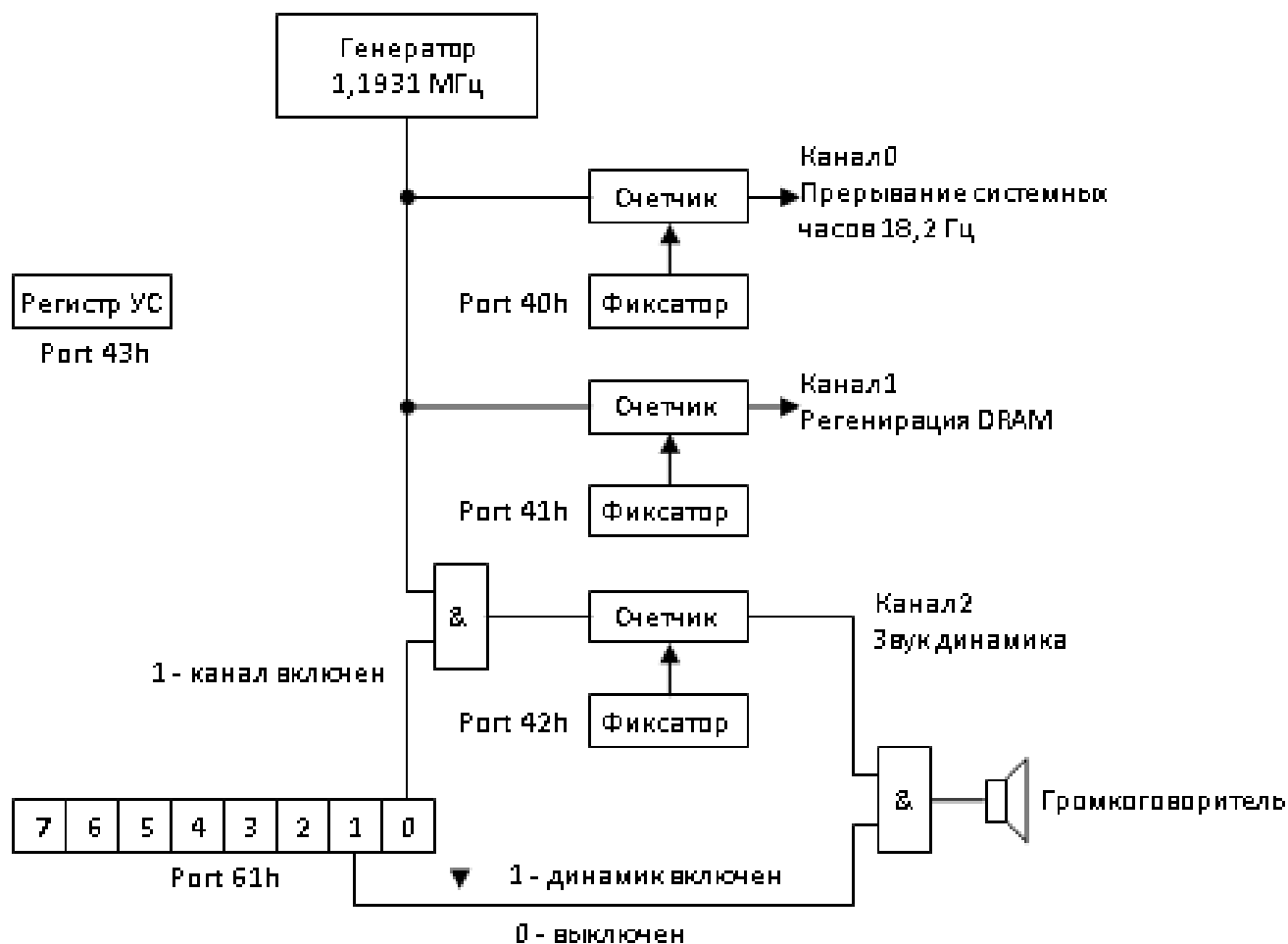
0A – записать только символ.

Функции выбора палитры цвета 0BH:

0B – установить цветовую палитру.

Функция записи и чтения пикселя 0CH, 0DH.

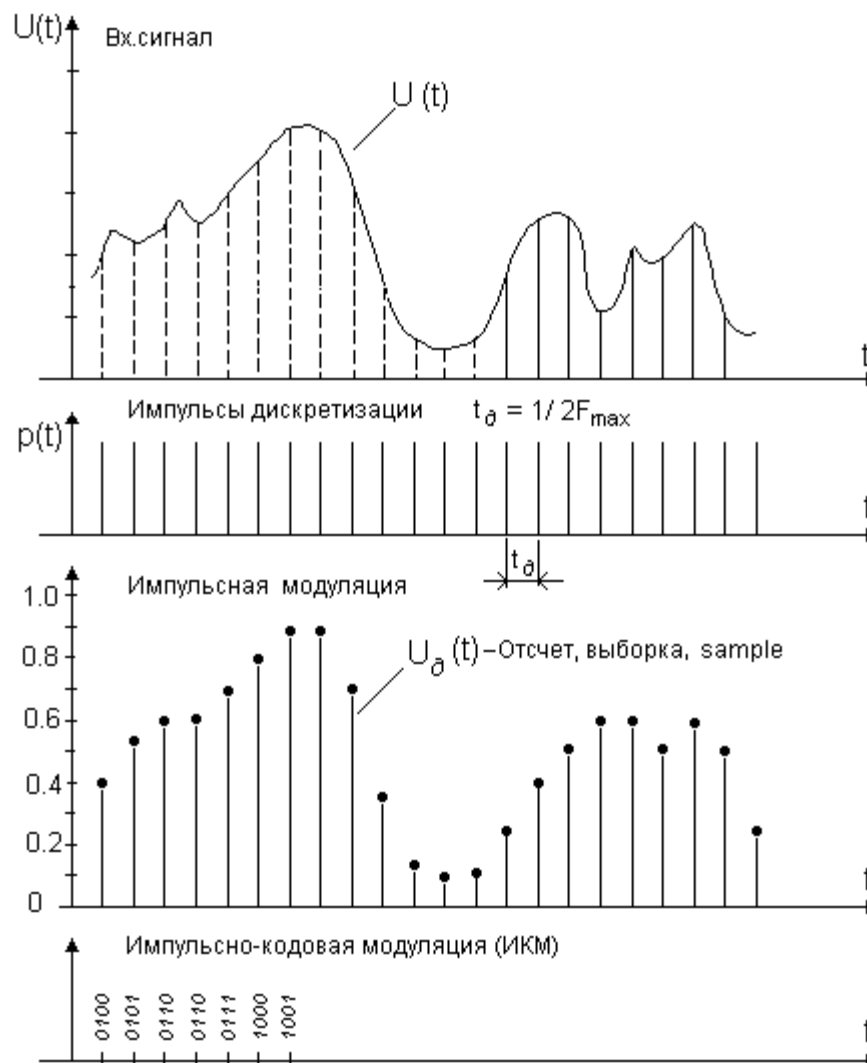
## Схема формирования звука.



## Аудиосистема компьютера.



## Аудиосистема компьютера. Принцип АЦП.



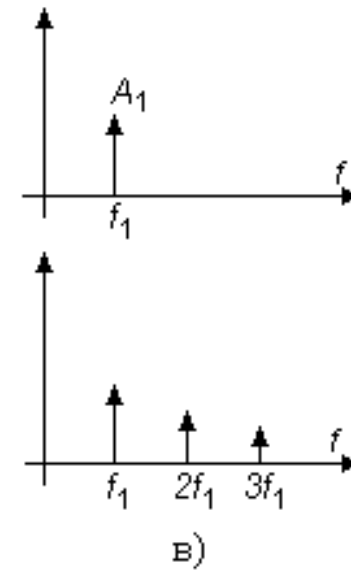
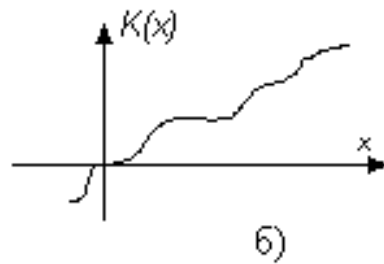
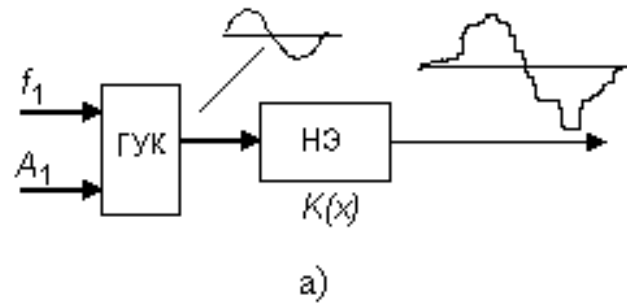
## Аудиосистема компьютера. Синтез звука.



Таблица музыкальных нот и соответствующих им частот

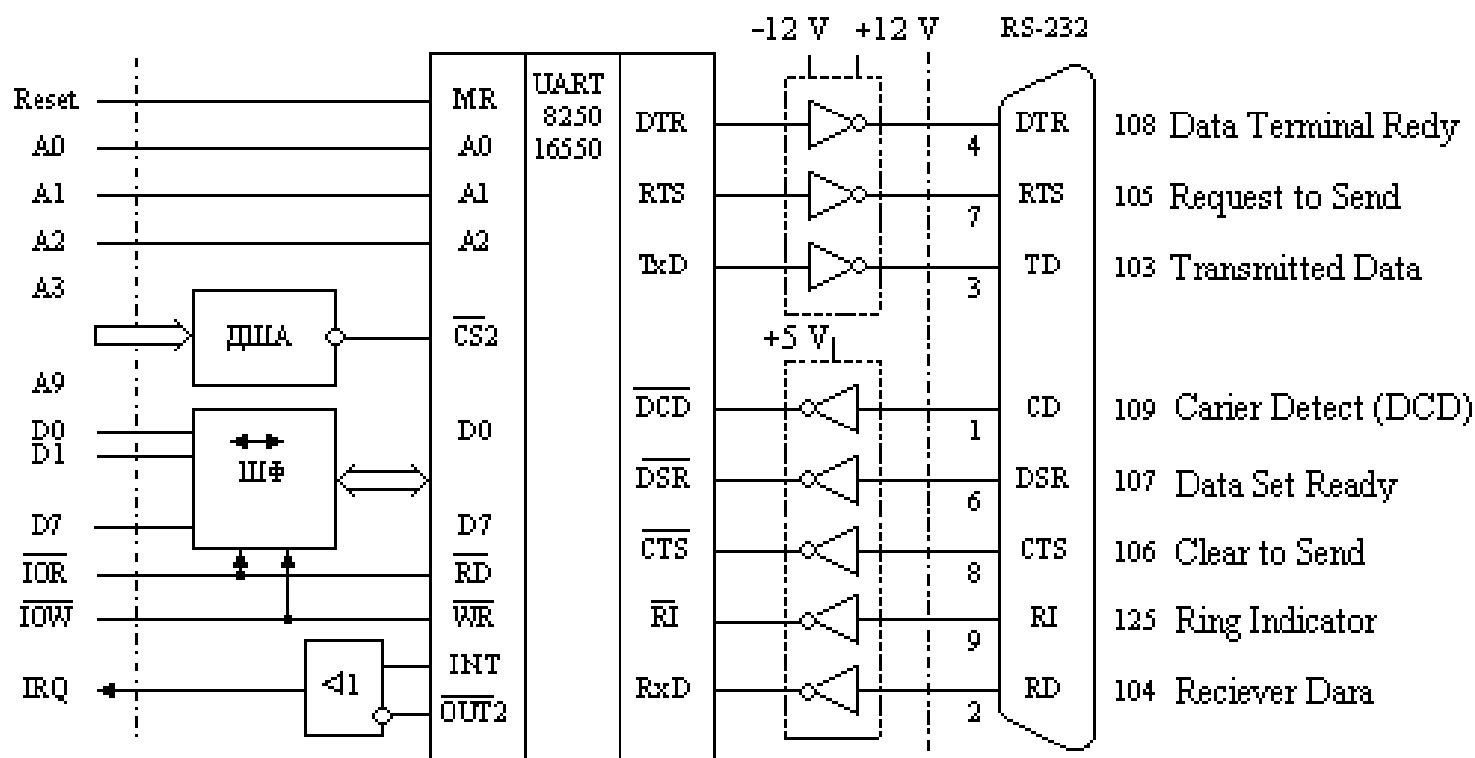
Нота	Частота, Гц.
До	261,63
До-диез	277,18
Ре	293,67
Ре-диез	311,13
Ми	329,63
Фа	349,22
Фа-диез	369,99
Соль	391,99
Соль-диез	415,30
Ля	440,00
Ля-диез	466,16
Си	493,88

## Аудиосистема компьютера. Синтез звука.



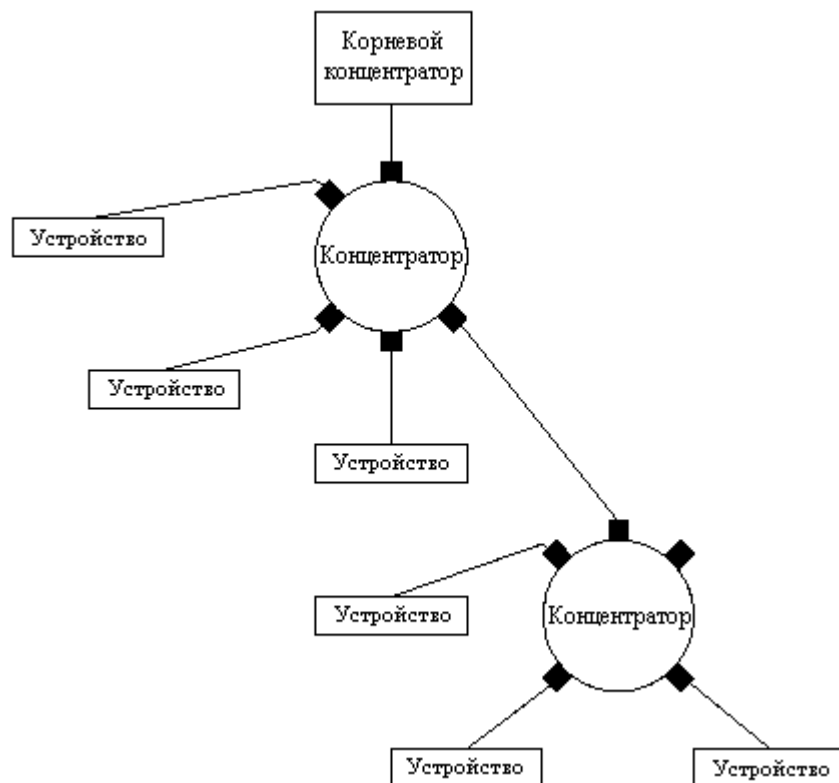
## Внешние интерфейсы компьютера.

### Последовательный интерфейс RS-232





## Универсальный интерфейс USB.



USB – 1,5 Мбит/с; 12 Мбит/с

USB-2 – 480 Мбит/с

USB-3 – 5 Гбит/с

Типы передаваемых данных:

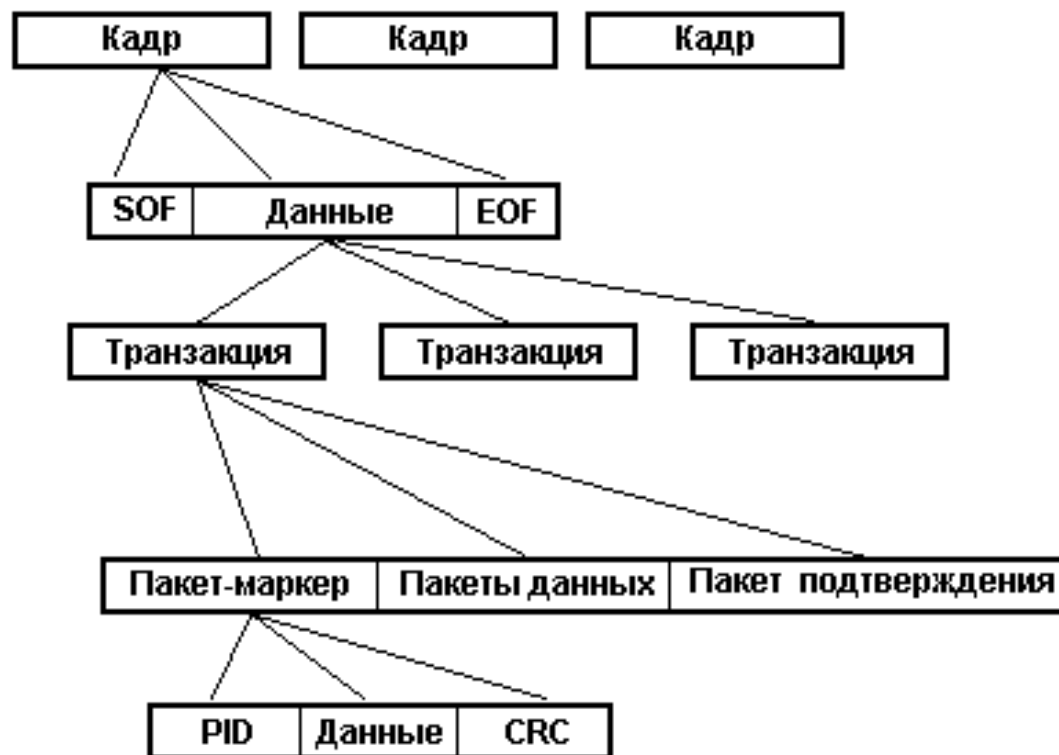
- 1) Упр. послыки до 64 байт; для конфигурирования устройств;
- 2) Передача пакетов 8-64 байта
- 3) Прерывания – передача типа вв-х символов или координат;
- 4) Изохронная передача (мультимедийные данные, без повтора при наличии ошибок).

Транзакция: 3 пакета

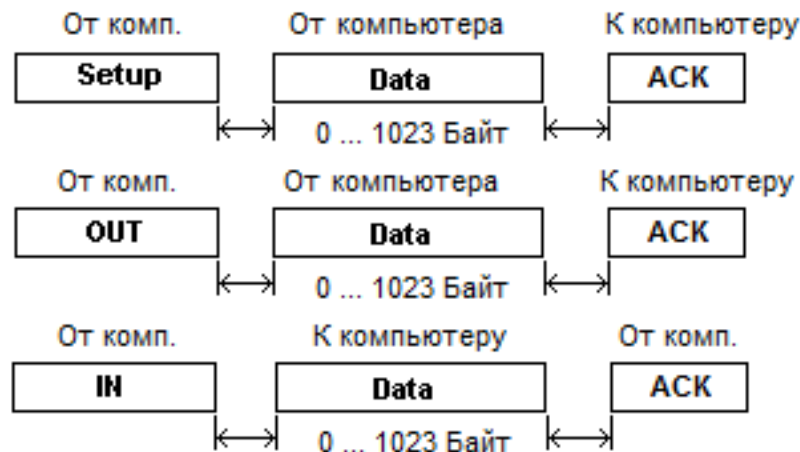
1. Маркер (тип, направление, адр.)
2. Данные
3. Подтверждение

## Универсальный интерфейс USB.

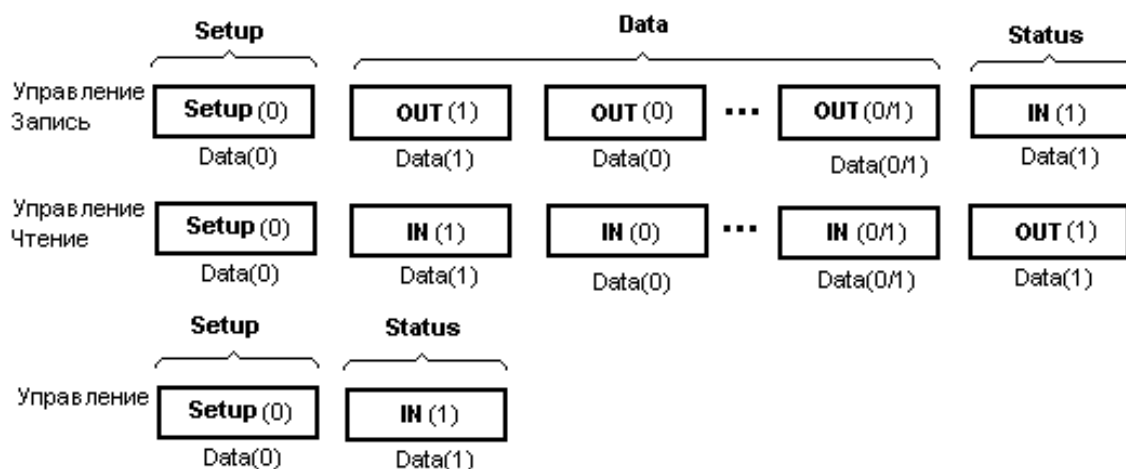
Обмен данными по шине USB инициируется хост-контроллером.  
Хост-контроллер формирует кадры из наиболее приоритетных данных.



## Универсальный интерфейс USB.



### Виды транзакций

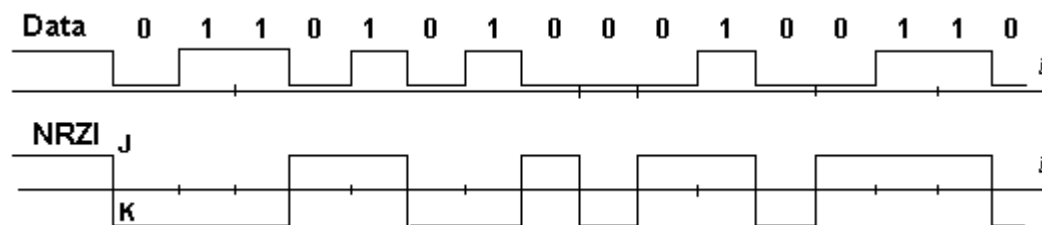


Управляющие последовательности при записи и чтении данных

## Универсальный интерфейс USB.



Пакеты маркеры	PID
OUT	0001
IN	1001
SETUP	1101
SOF	0101
PING	0100
Пакеты данных	
DATA0	0011
DATA1	1011
DATA2	0111
MDATA	1111
Пакеты подтверждения (Handshake)	
ACK	0010
NAK	1010



Сигналы для передачи данных

## Мультимедийный интерфейс высокого разрешения **HDMI** (*High-Definition Multimedia Interface*)

Пропускная способность: от 4,9 до 10,2 Гбит/с.

Входят следующие цепи (19):

**TMDS** (*Transition-Minimized Differential Signaling*). Использует три канала (TMDS Data0 - TMDS Data2), передающие потоки аудио/видео и дополнительные данные (TMDS Clock), с пропускной способностью до 3,4 Гбит/с на канал.

**CEC** (*Consumer Electronics Control*). Используются для передачи команд и управляющих команд между участниками связи. Среди команд есть включение/выключение, воспроизведение, переход в режим ожидания, запись и другие.

**SCL** (*Serial Data Clock*). Применяется для сигналов синхронизации передачи данных.

**SDA** (*Serial Data Access*). Для доступа передачи данных.

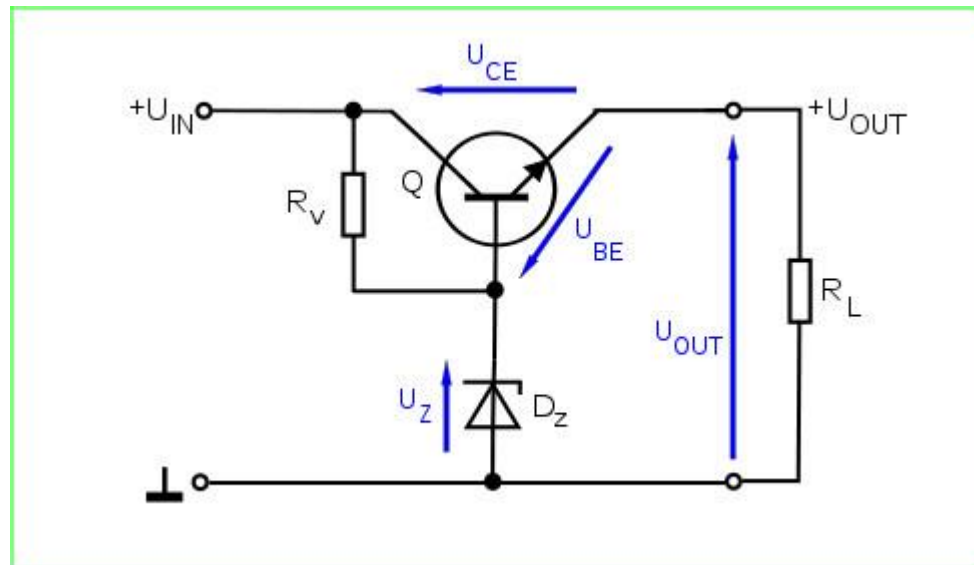
**DDC** (*Display Data Channel*). Служит для передачи спецификации дисплея, в частности, название производителя, номер модели, поддерживаемые форматы и разрешения и т.д.

**Цепи питания** (+ 5 В), контроля подсоединения и экранов.

## Мультимедийный интерфейс высокого разрешения HDMI (*High-Definition Multimedia Interface*)



## КЛАССИЧЕСКИЙ ИСТОЧНИК ПИТАНИЯ ЭЛЕКТРОННЫХ УСТРОЙСТВ



Недостаток: большие габариты и вес из-за сетевого трансформатора

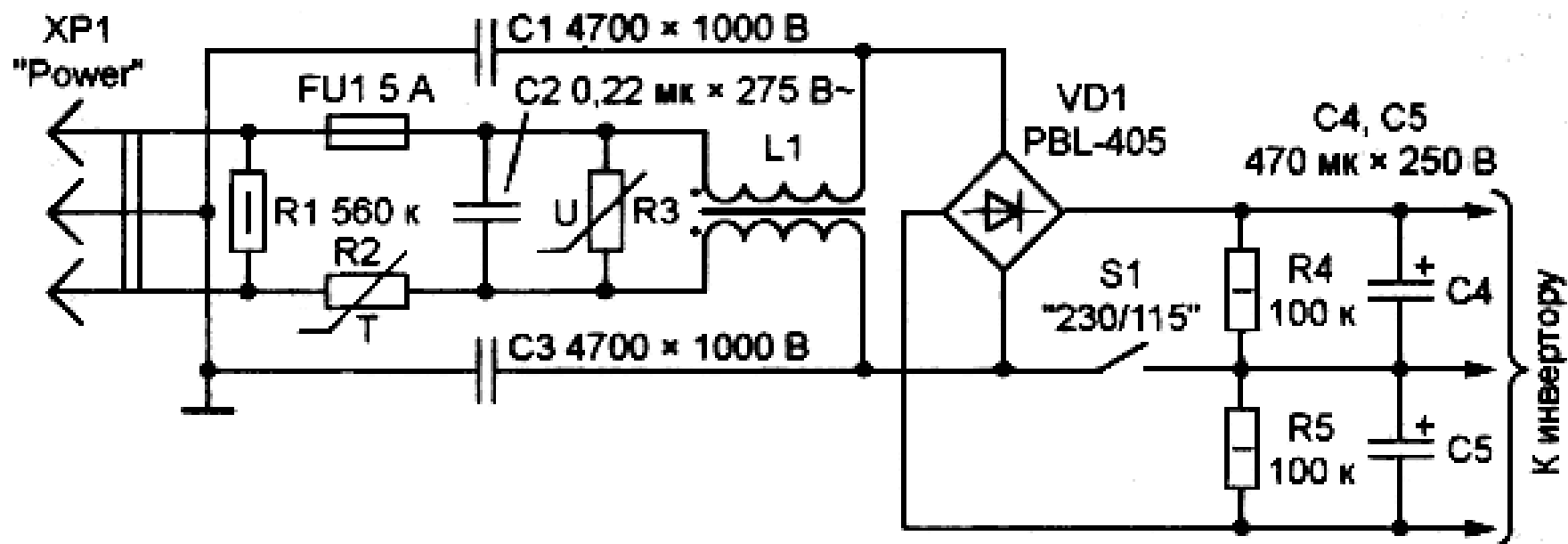
## ИСТОЧНИКИ ПИТАНИЯ КОМПЬЮТЕРОВ

Типовыми вторичными напряжениями источника питания являются (+3,3 В; 20 А), (+5 В; 32 А), (+12 В; 16А), (-5 В; 0,5 А), (-12 В; 0,5 А).

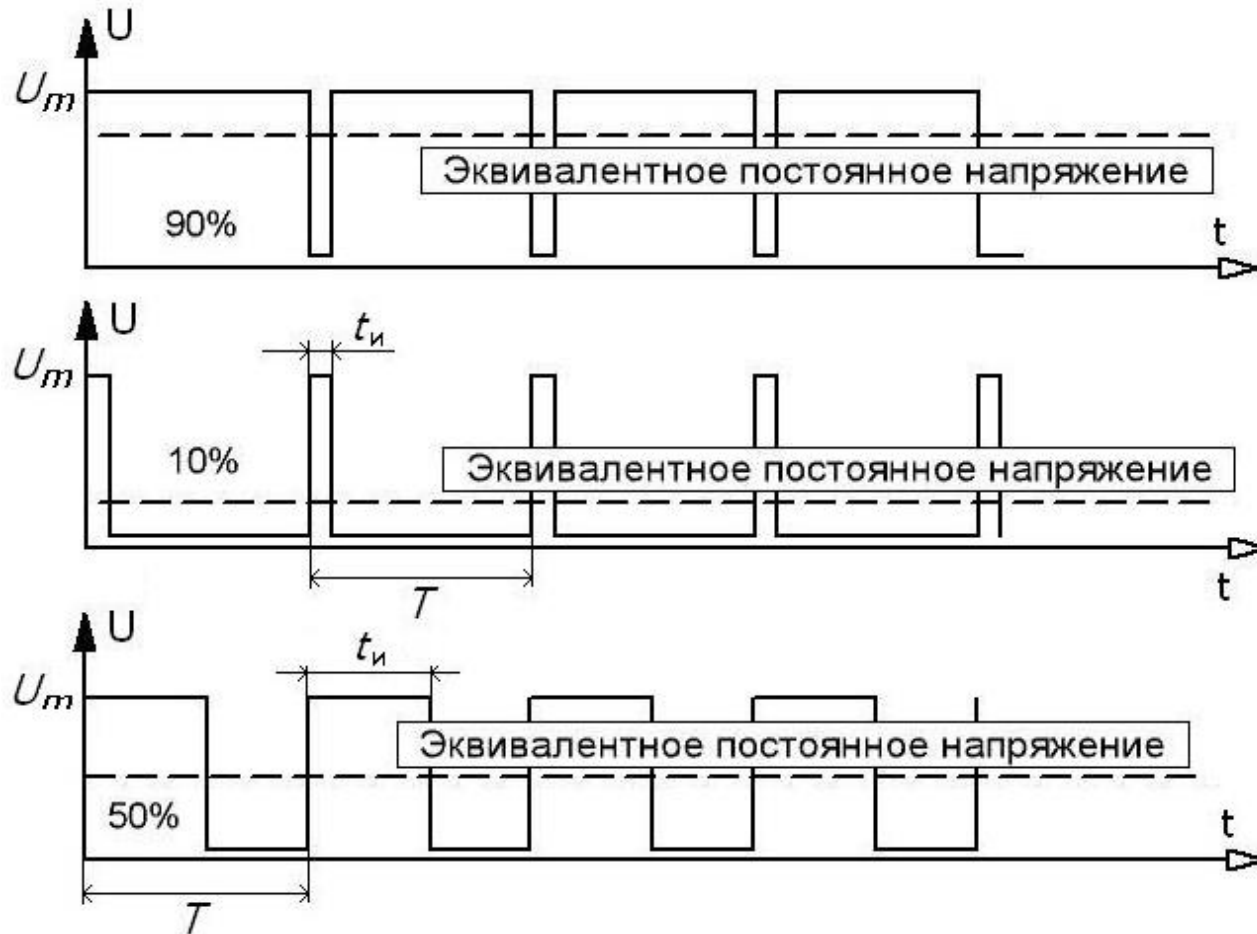




## ИСТОЧНИКИ ПИТАНИЯ КОМПЬЮТЕРОВ



## ИСТОЧНИКИ ПИТАНИЯ КОМПЬЮТЕРОВ



Среднее значение выходного напряжения преобразователя при ШИМ

$$U_{cp} = (t_{и}/T)U_m = gU_m$$

## ТЕХНИЧЕСКИЕ СРЕДСТВА ВВОДА ИНФОРМАЦИИ В КОМПЬЮТЕР

Измерительные преобразователи подразделяются на **первичные** и **вторичные**.

Измерительный преобразователь, на который непосредственно воздействует измеряемая физическая (или иная) величина называют первичным измерительным преобразователем. *Вторичный измерительный преобразователь* выполняет функцию звена, связывающего первичный измерительный преобразователь со вторичными устройствами (АЦП).

**Нормирующие преобразователи** служат для:

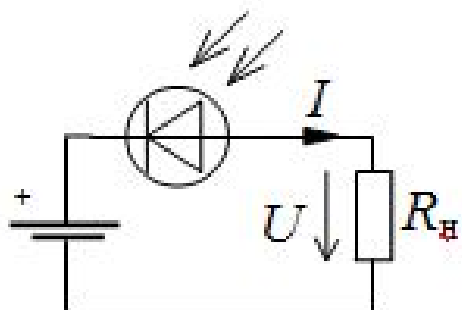
- 1) усиления слабых сигналов и формирования унифицированных уровней сигналов напряжения или тока;
- 2) выполнения (при необходимости) линеаризации нелинейных характеристик первичного преобразователя;
- 3) осуществления термокомпенсации, если первичный преобразователь подвержен сильному влиянию температуры, как, например, в случае с термопарами и емкостными датчиками влажности;
- 4) снижения влияния электромагнитных помех.

### Унифицированные сигналы

по напряжению:  $0 - 0,01 \text{ В}$ ;  $0 - 1 \text{ В}$  или  $0 - 10 \text{ В}$ ;

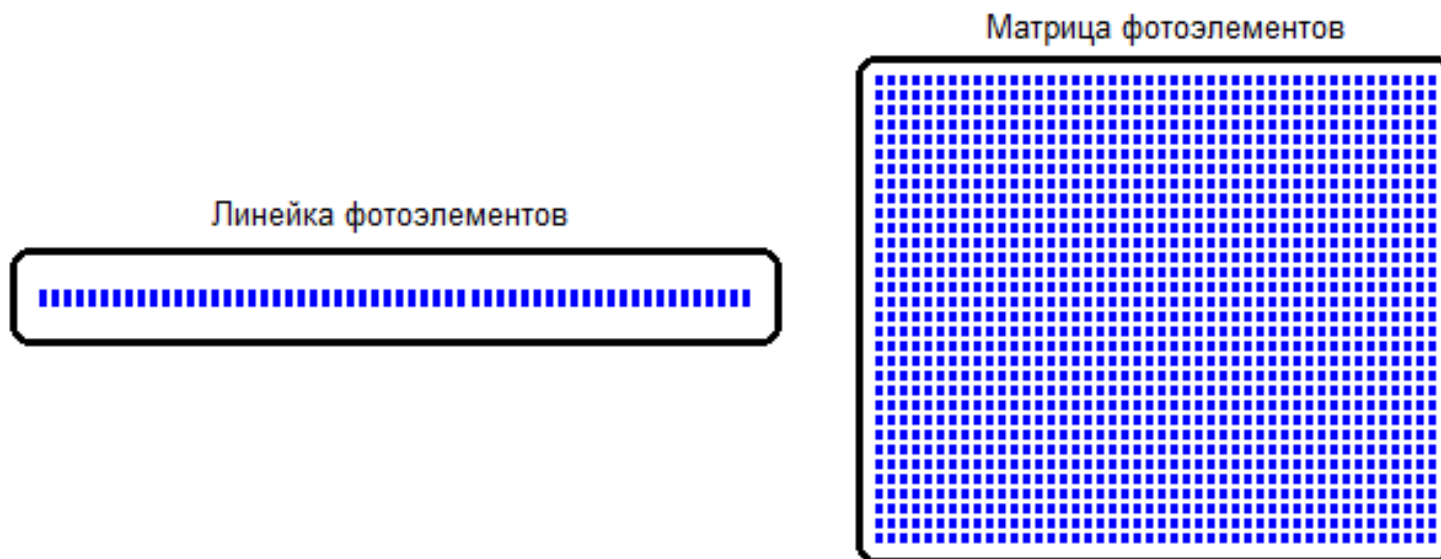
по току:  $0 - 5 \text{ мА}$ ;  $0 - 20 \text{ мА}$  или  $4 - 20 \text{ мА}$ .

## ПРЕОБРАЗОВАТЕЛИ ИЗОБРАЖЕНИЙ В ЭЛЕКТРИЧЕСКИЙ СИГНАЛ

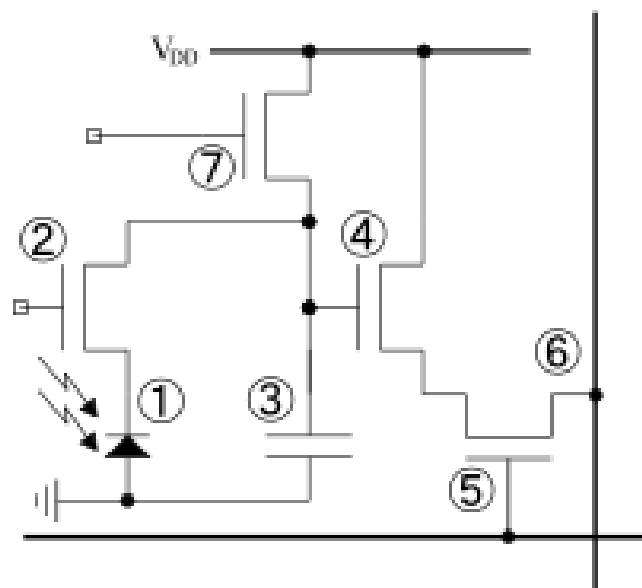


Преобразователь на основе фотодиода

Преобразователь на основе фотоэлементов ПЗС или КМОП-транзисторов



## КМОП-сенсоры

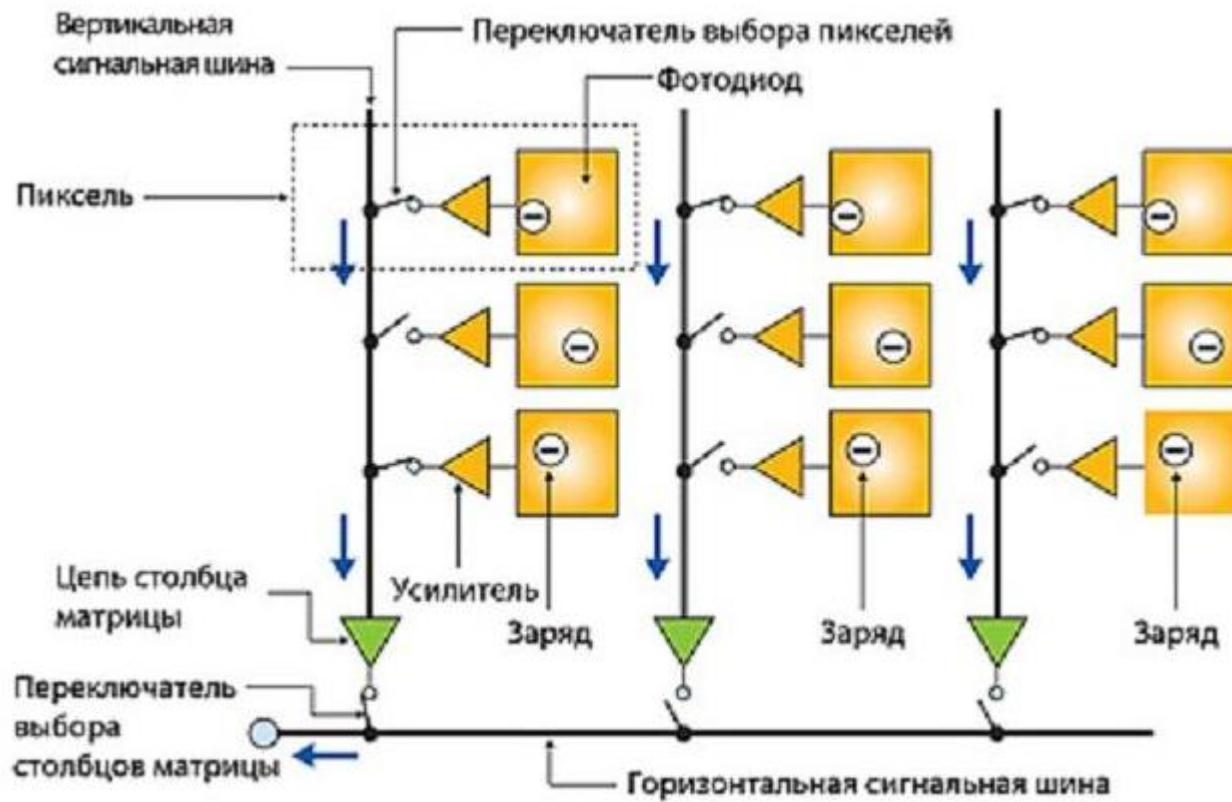


- 1 — светочувствительный элемент (фотодиод);
- 2 — затвор МОП-транзистора;
- 3 — конденсатор, сохраняющий заряд с фотодиода;
- 4 — усилитель;
- 5 — шина выбора строки;
- 6 — вертикальная шина, передающая сигнал процессору;
- 7 — сигнал сброса.

**Преимущества CMOS матриц** — малая стоимость в производстве, низкое энергопотребление (почти в 100 раз по сравнению с ПЗС) и высокое быстродействие.

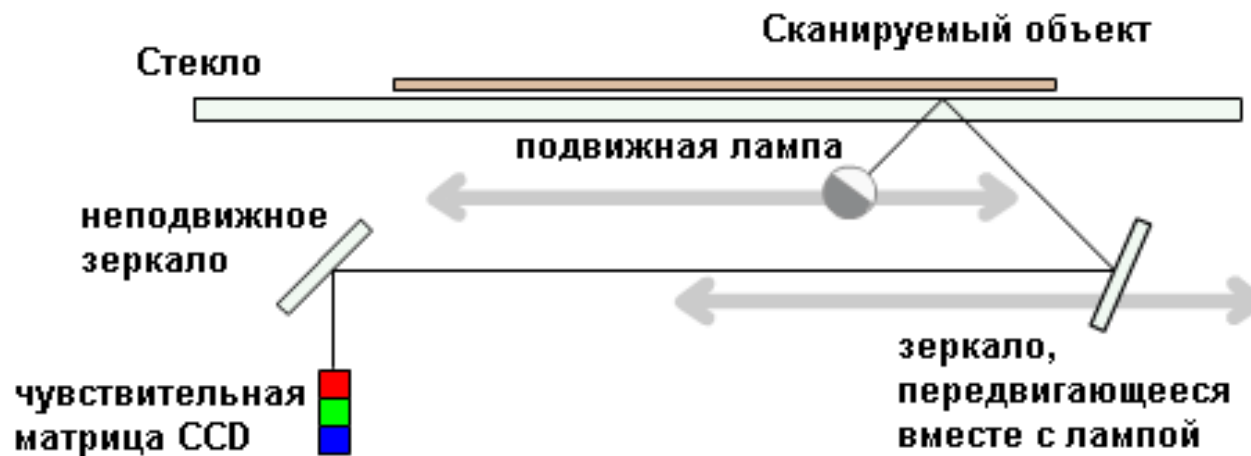
**Недостатки:** малый размер светочувствительного элемента по сравнению с площадью пикселя. Большая часть площади занимает электроника, встроенная в пиксель.

## Матрица КМОП-фотосенсоров



Структура CMOS-матрицы

## СХЕМА ПЛАНШЕТНОГО СКАНЕРА



## СЕНСОРНЫЕ ЭКРАНЫ

Сенсорный экран состоит из следующих элементов: собственно экран (панель), контроллер и интерфейс.

Существуют несколько типов построения сенсорных экранов:

1) **Резистивные:**

- 4-проводные;
- 5-проводные;
- 8-проводные;

2) **Емкостные;**

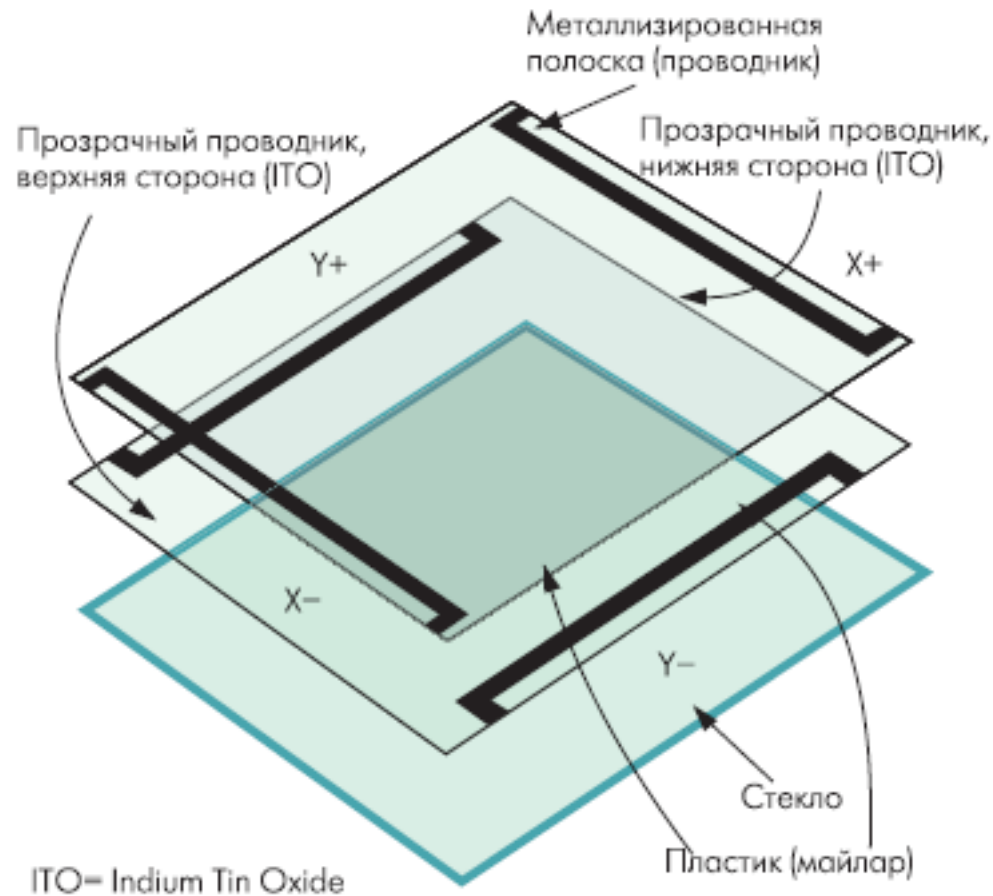
3) **Цифровые;**

4) **На поверхностных акустических волнах** (ПАВ или SAW);

5) **Индукционные.**

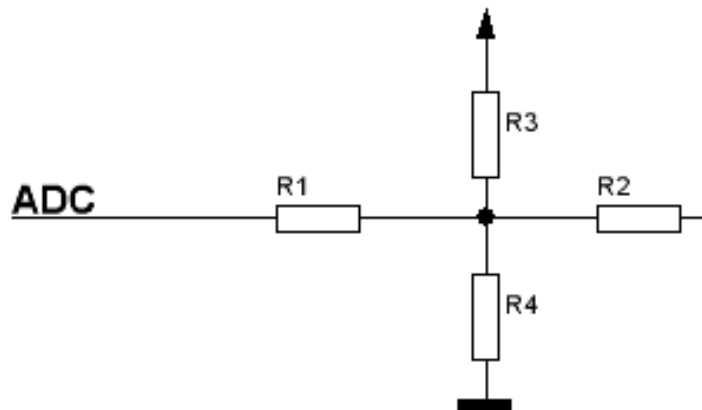


## СЕНСОРНЫЕ РЕЗИСТИВНЫЕ ЭКРАНЫ



Конструкция 4-проводного сенсорного экрана

## СЕНСОРНЫЕ РЕЗИСТИВНЫЕ ЭКРАНЫ



Обычно полное сопротивление одной плоскости экрана колеблется в диапазоне от 100 до 900 Ом.

## СЕНСОРНЫЕ РЕЗИСТИВНЫЕ ЭКРАНЫ

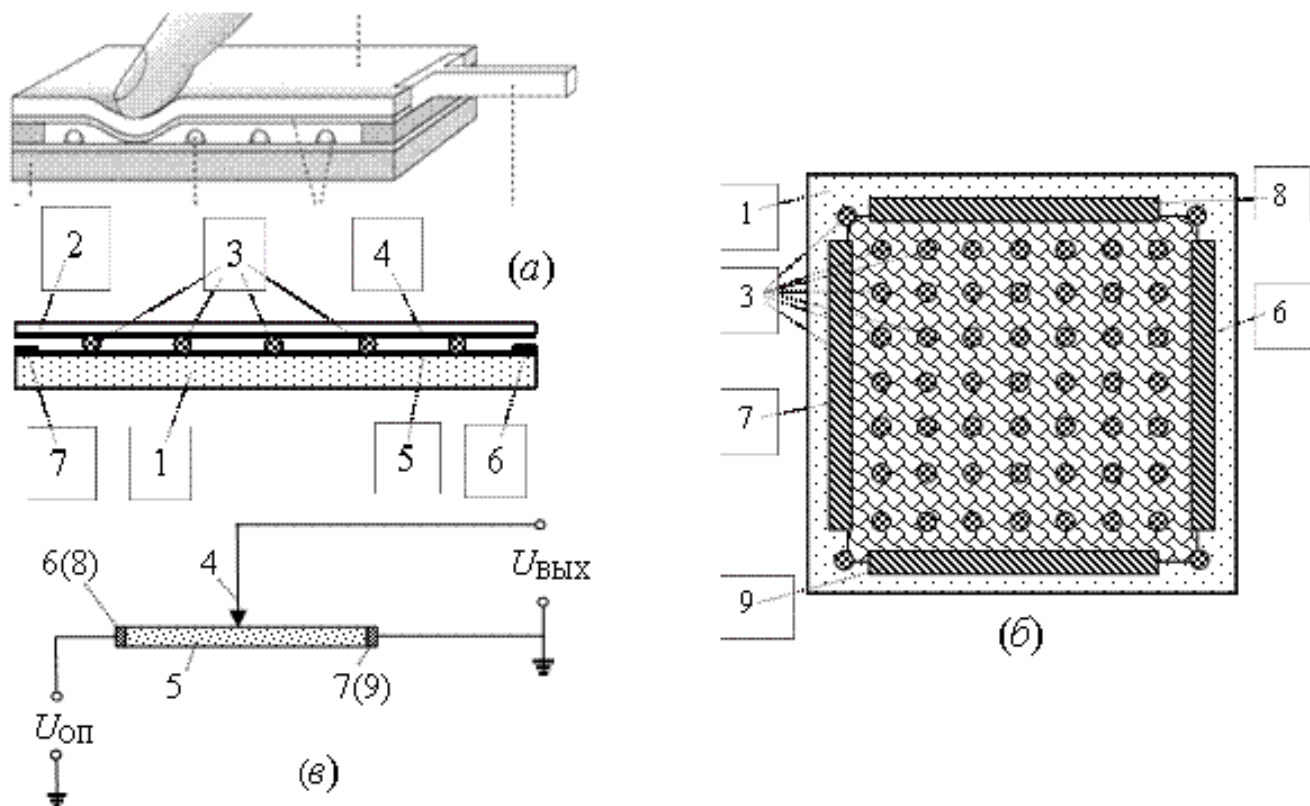
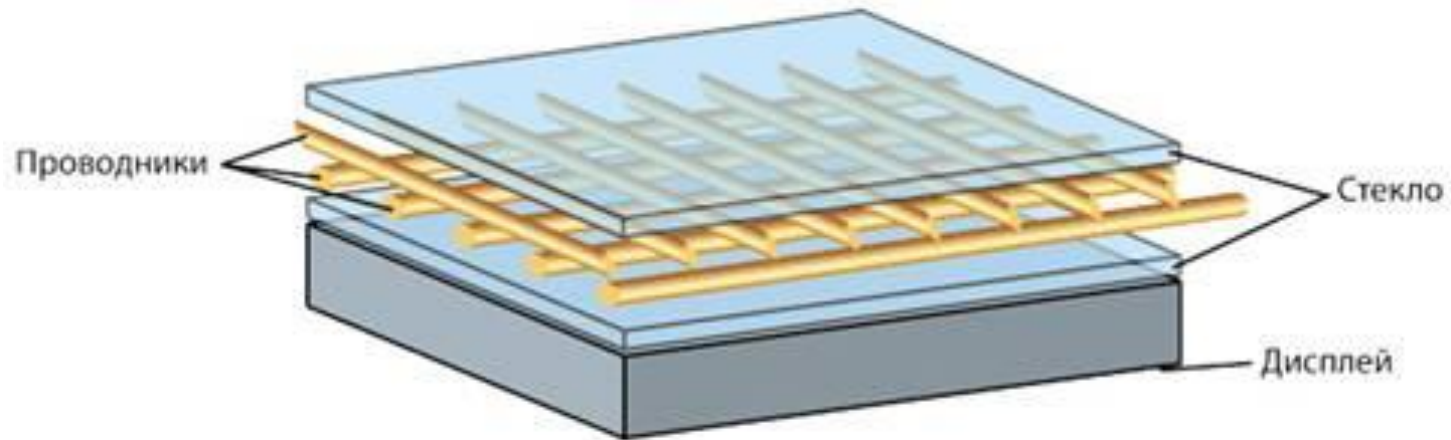


Схема 4-проводного сенсорного экрана

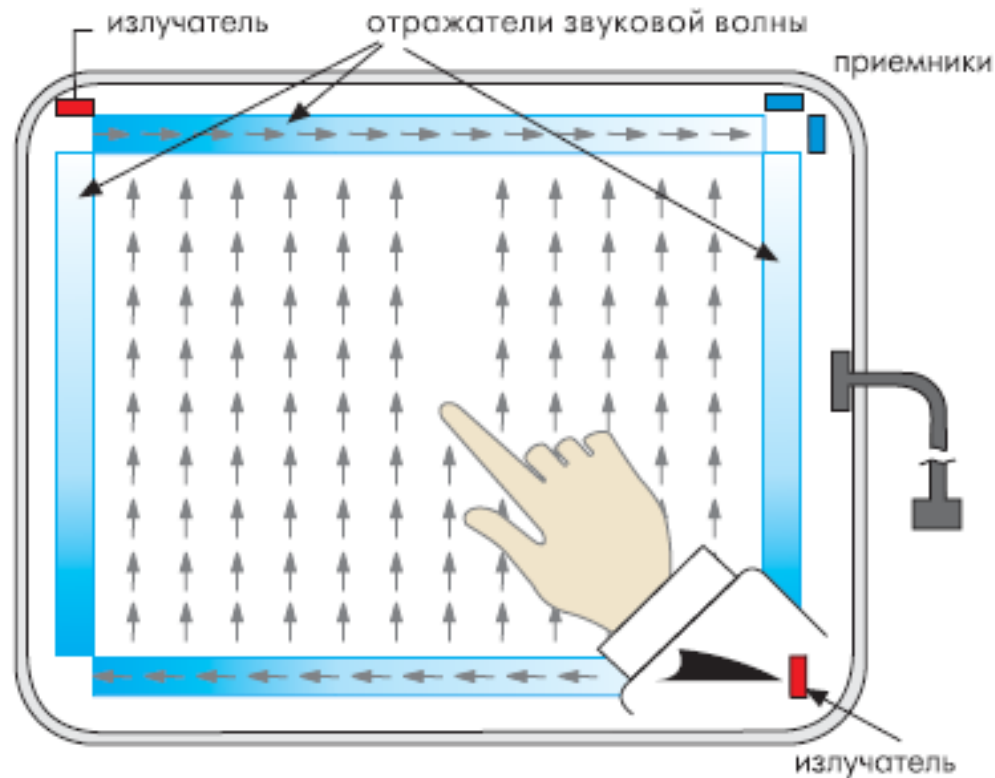
## СЕНСОРНЫЕ ЕМКОСТНЫЕ ЭКРАНЫ



Контроллер последовательно подает на каждый из электродов напряжение и измеряет амплитуду возникающего импульса тока.

Важное преимущество ёмкостных сенсорных экранов — это способность их сохранять практически 90 % от изначальной яркости дисплея. Из-за этого изображения на ёмкостном экране смотрятся более чёткими, чем на экранах, имеющих резистивную конструкцию.

## СЕНСОРНЫЕ ЭКРАНЫ НА ПОВЕРХНОСТНЫХ АКУСТИЧЕСКИХ ВОЛНАХ



У панелей, сделанных по этой технологии, точность определения координат высока, но при этом они чувствительны к качеству поверхности, наличию жира, грязи.

## ИНДУКЦИОННЫЕ СЕНСОРНЫЕ ЭКРАНЫ

Принцип работы основан на использовании резонансных катушек индуктивности, расположенных внутри сенсорной панели и катушки, помещенной внутрь стилуса. Внутри сенсорной панели размещается матрица индуктивных катушек, которые формируют на ее поверхности электромагнитное поле.

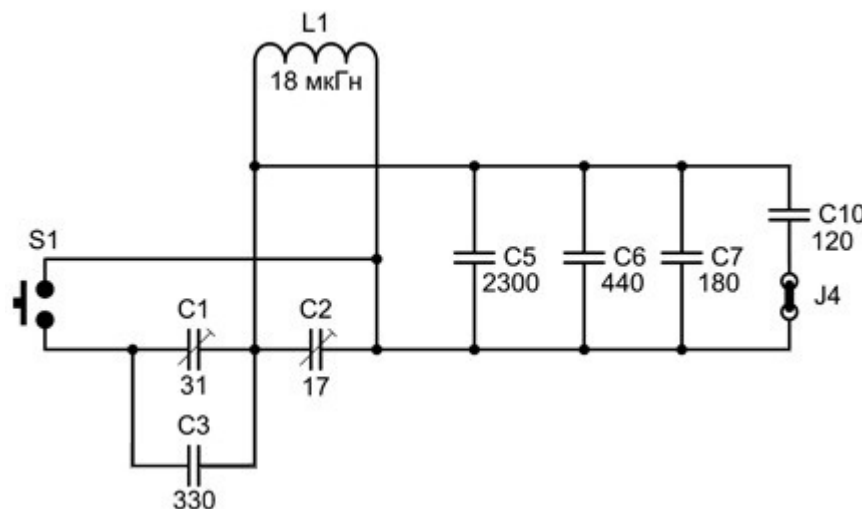
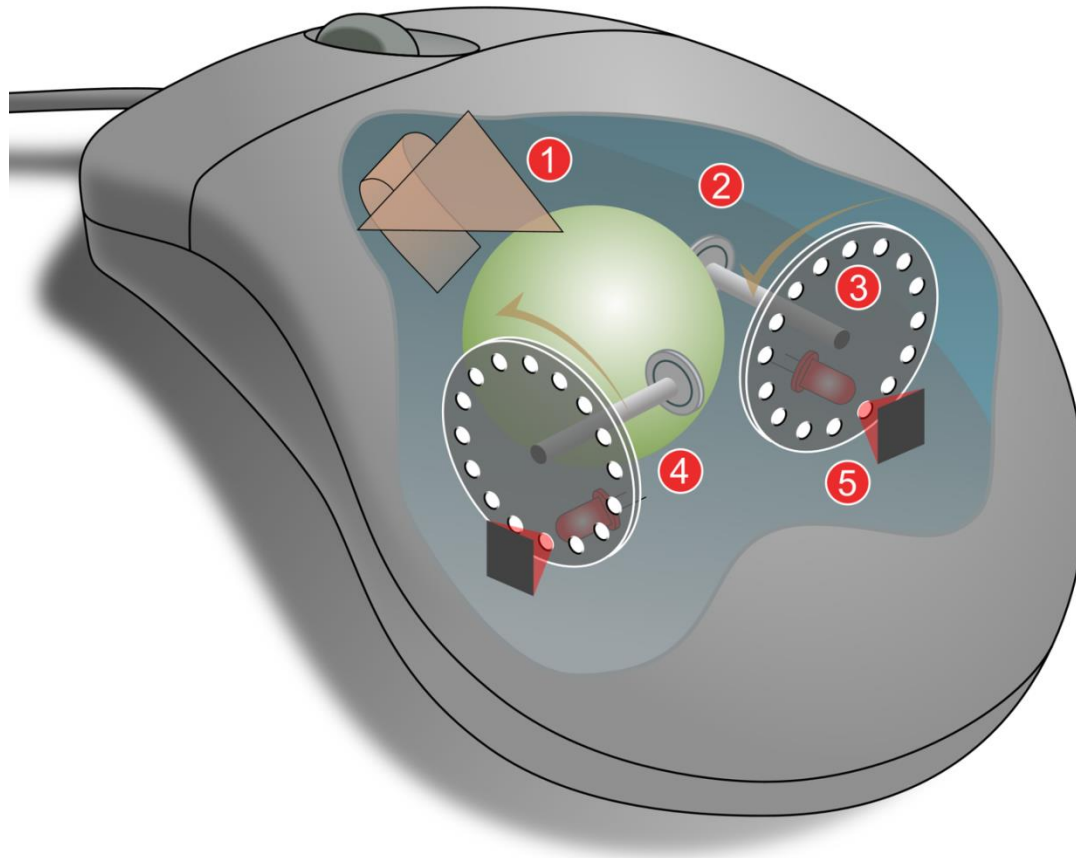
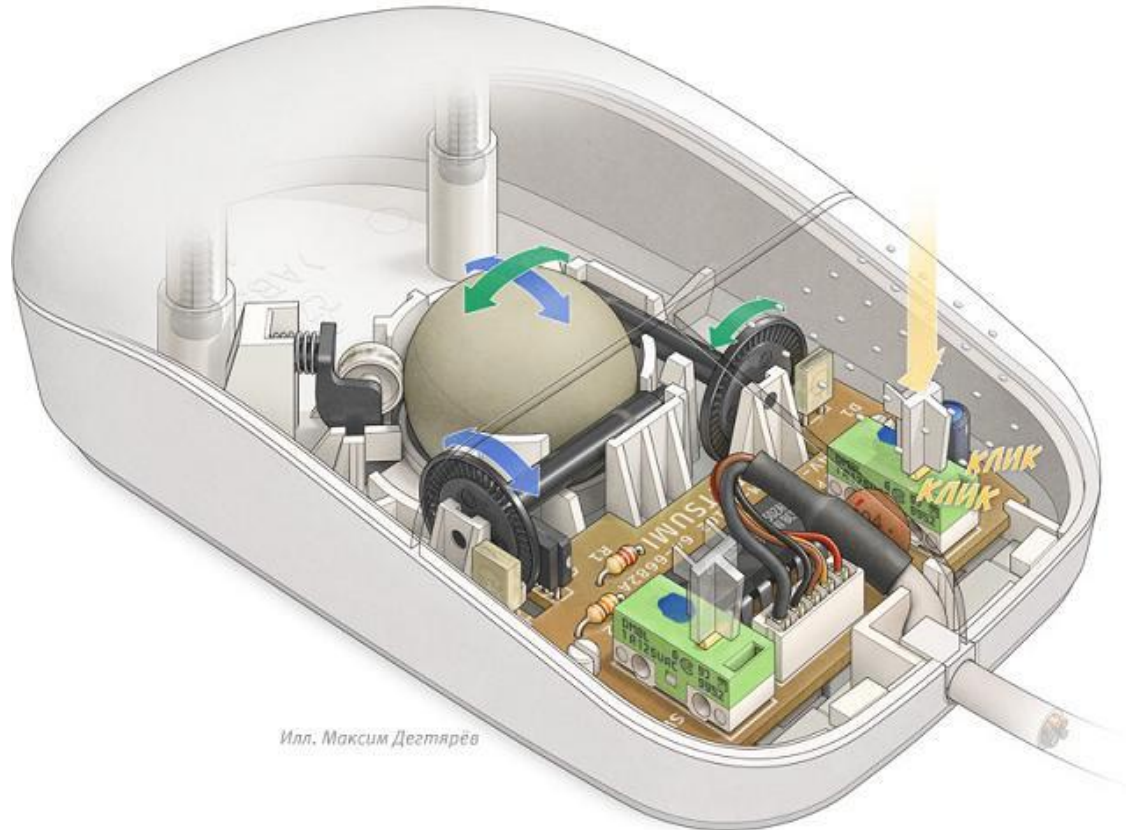


Схема внутреннего устройства стилуса для индуктивного сенсорного экрана

## ШАРИКОВАЯ МЫШЬ



## ШАРИКОВАЯ МЫШЬ



Илл. Максим Дегтярёв



## ОПТИЧЕСКАЯ МЫШЬ

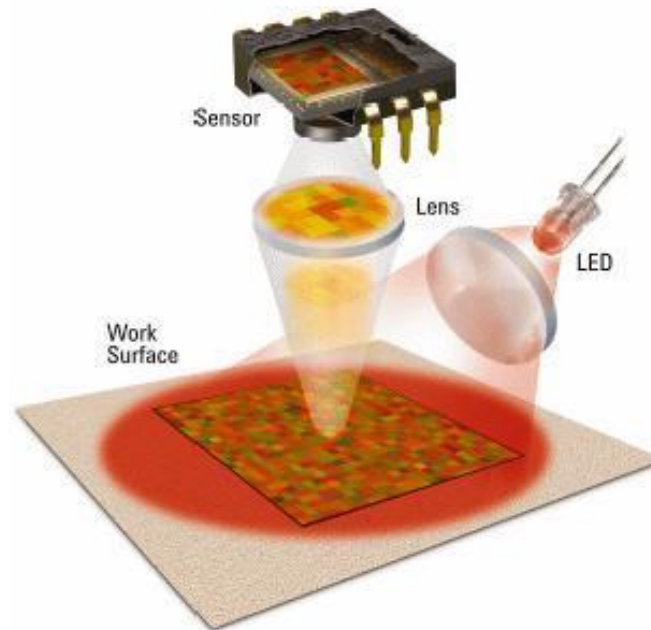


Содержит излучатель и процессор обработки изображений. Процессор делает снимки поверхности под мышью с частотой несколько кГц.

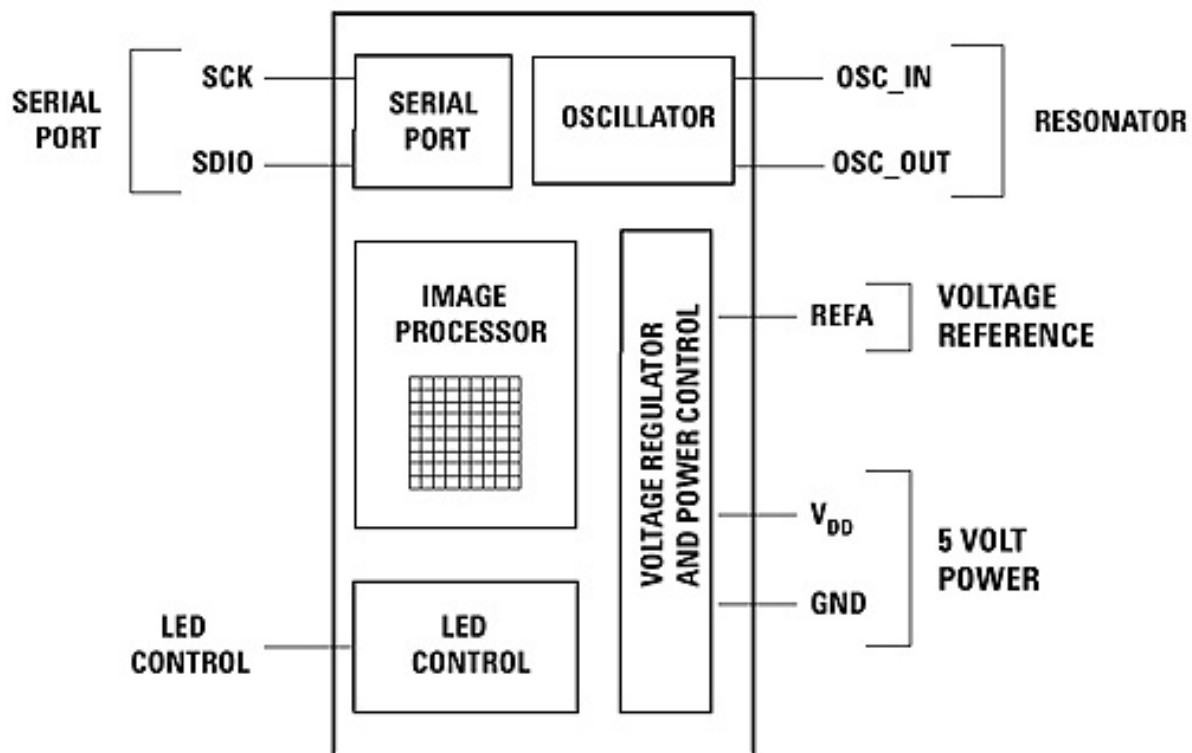
На основании анализа ряда последовательных снимков, интегрированный DSP-процессор высчитывает результирующие показатели, свидетельствующие о направлении перемещения мыши вдоль осей X и Y

В оптической мыши со светодиодом используется видимое излучение (красный). В лазерной мыши – инфракрасное невидимое излучение. Лазерные более точные, меньше потребляют энергии, не критичны к поверхности сканирования.

## ОПТИЧЕСКАЯ МЫШЬ



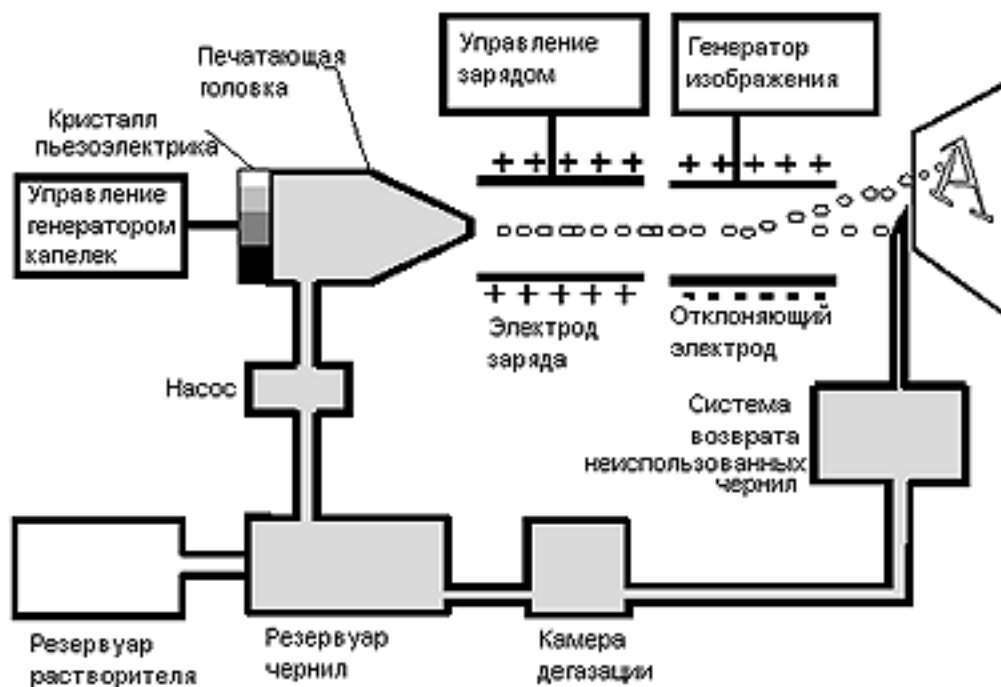
## ОПТИЧЕСКАЯ МЫШЬ. СТРУКТУРА ОПТИЧЕСКОГО СЕНСОРА



## ТЕХНИЧЕСКИЕ СРЕДСТВА ВЫВОДА ИНФОРМАЦИИ

### Струйные принтеры

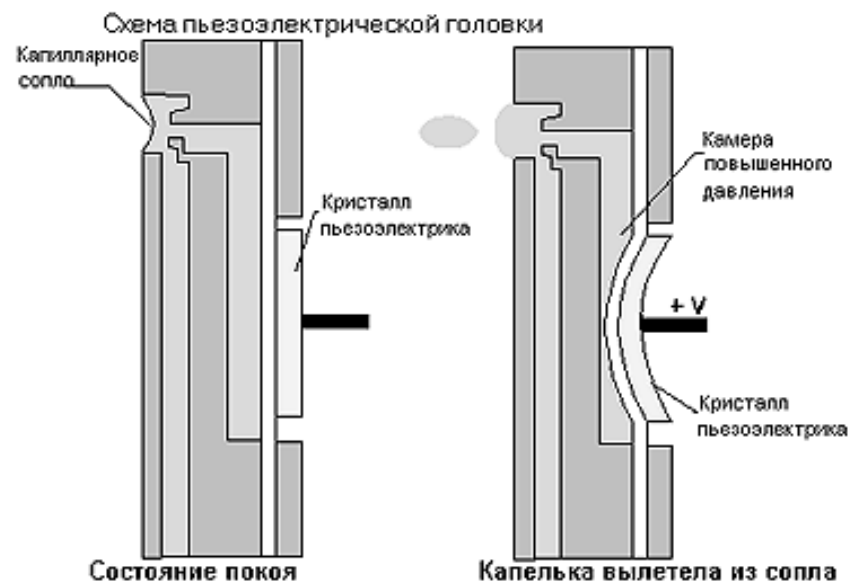
Схема устройства непрерывной струйной печати



## ТЕХНИЧЕСКИЕ СРЕДСТВА ВЫВОДА ИНФОРМАЦИИ

### Струйные принтеры

#### Пьезоэлектрическая струйная печать



## ТЕХНИЧЕСКИЕ СРЕДСТВА ВЫВОДА ИНФОРМАЦИИ

### Лазерные принтеры

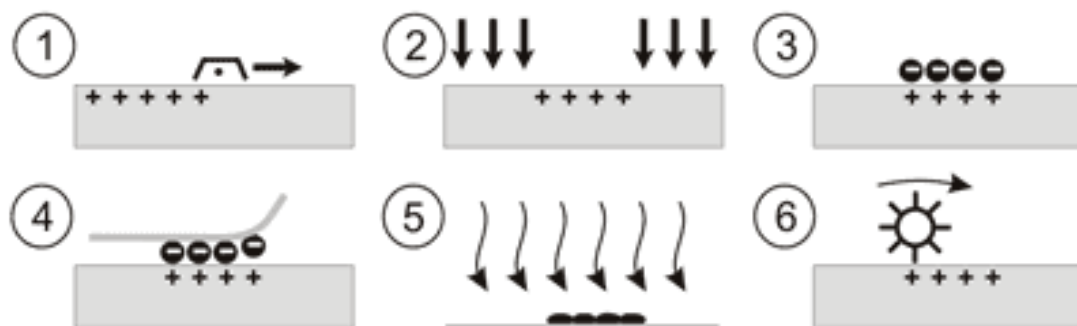
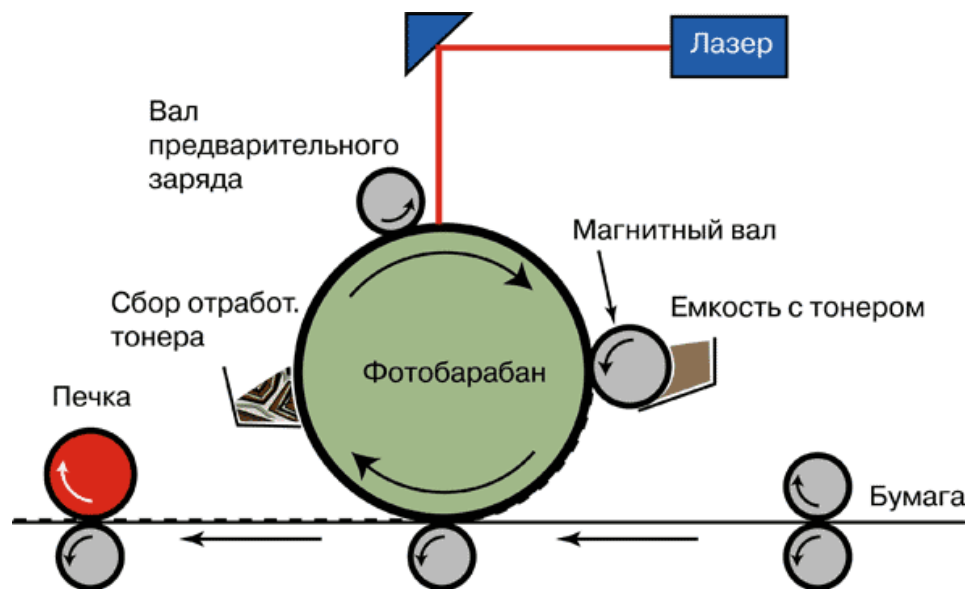


Схема лазерной печати

- 1) зарядка светочувствительного барабана коронирующим устройством;
- 2) экспонирование светом (создание скрытого изображения);
- 3) проявление скрытого изображения тонером (частицы которого имеют заряд, противоположный заряду на светочувствительном барабане);
- 4) перенос изображения на бумагу;
- 5) термическое закрепление тонера на бумаге;
- 6) нейтрализация заряда на барабане и его очистка.

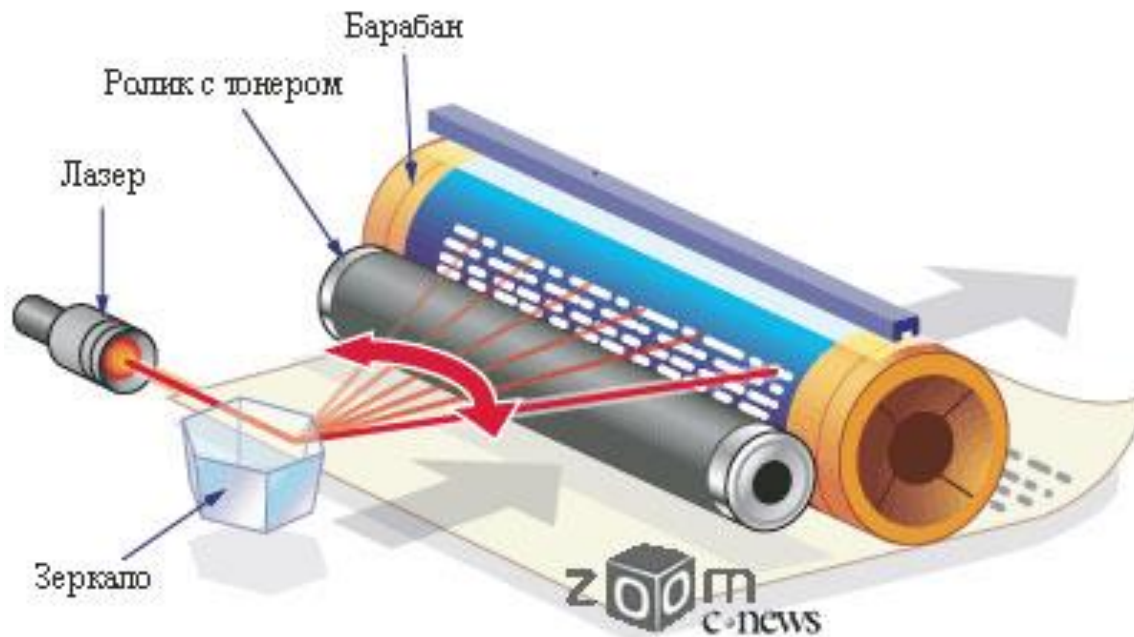
## ТЕХНИЧЕСКИЕ СРЕДСТВА ВЫВОДА ИНФОРМАЦИИ

### Лазерные принтеры



После формирования строки изображения, специальный прецизионный шаговый двигатель поворачивает барабан так, чтобы можно было формировать следующую строку. Это смещение равняется разрешающей способности принтера и обычно составляет  $1/300$  или  $1/600$  дюйма. В каждой строке на каждый дюйм приходится по 300 или 600 точек. Таким образом и получается "лазерное" разрешение в  $300 \times 300$  ( $600 \times 600$ ) dpi.

## ЛАЗЕРНЫЕ ПРИНТЕРЫ





## ТЕХНИЧЕСКИЕ СРЕДСТВА ВЫВОДА ИНФОРМАЦИИ

### Лазерные цветные принтеры

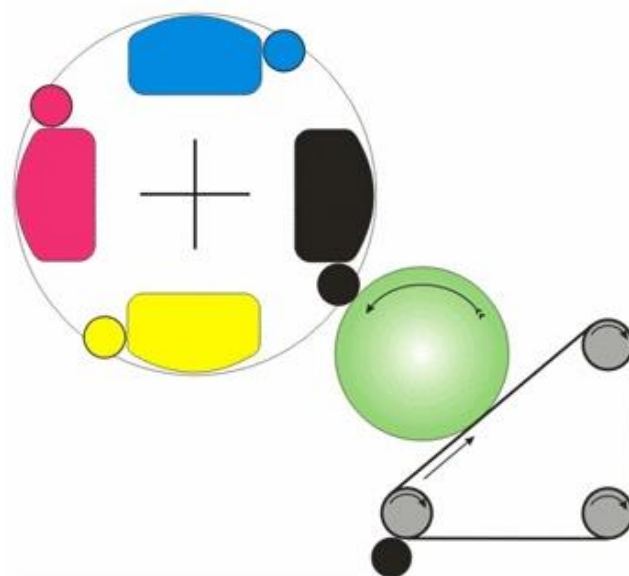


Схема многопроходного цветного принтера

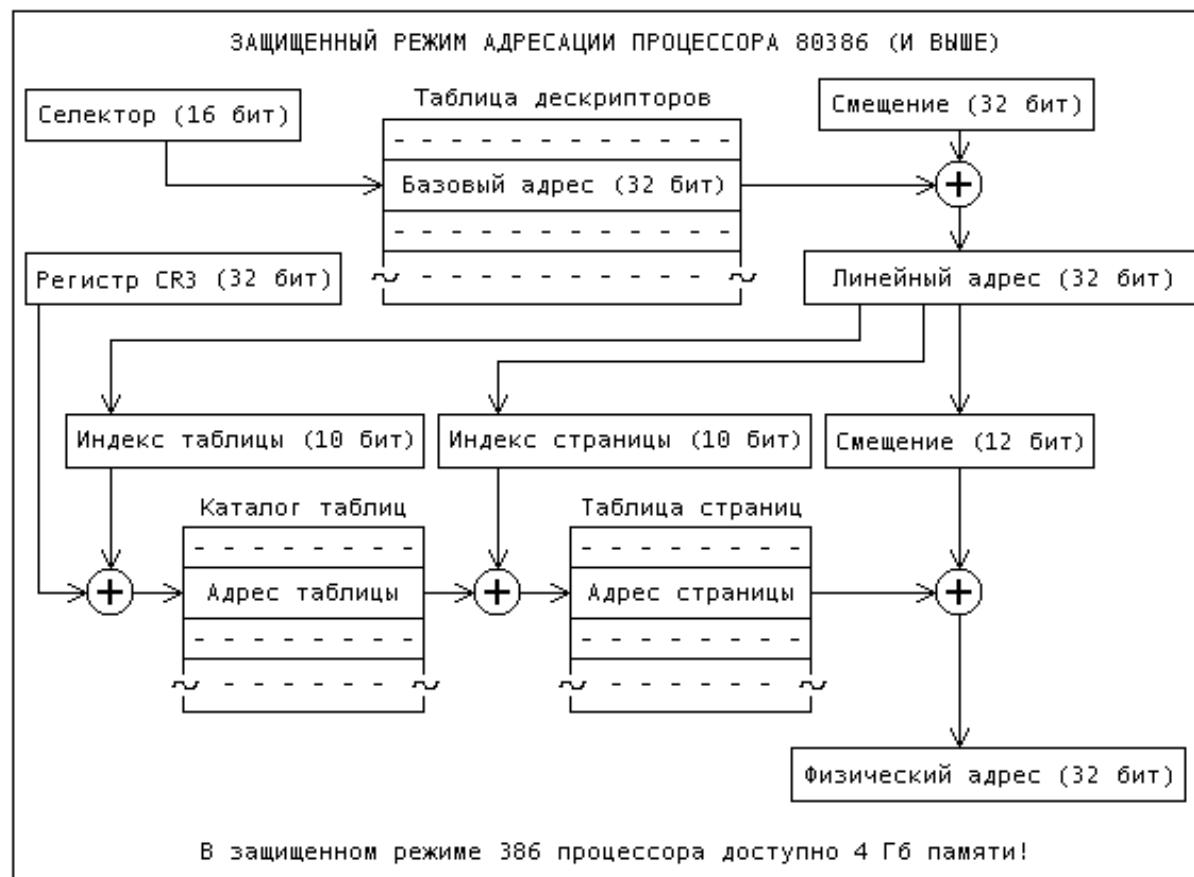
## ЗАЩИЩЕННЫЙ РЕЖИМ РАБОТЫ ПРОЦЕССОРА. Сегмент состояния задачи. (Task State Segment)

0	Link	
2	Stack 0	
6	Stack 1	
10	Stack 2	
14	IP	
16	FLAGS	
18	AX	
20	CX	
22	DX	
24	BX	
26	SP	
28	BP	
30	SI	
32	DI	
34	ES	
36	CS	
38	SS	
40	DS	
42	LDTR	

Поле Link представляет собой поле обратной связи и используется для организации вложенных вызовов задач.

Поля Stack 0, Stack 1, Stack 2 хранят логические адреса (селектор:смещение) отдельных для каждого кольца защиты стеков.

## Защищенный режим адресации процессора



## Пример работы с памятью

Для того, чтобы указать, какой размер данных необходимо переслать, надо указать перед адресом оператор размера

```
mov eax, dword ptr [0000509Ah]  
mov rbx, qword ptr [00A056F1h]  
mov dword ptr [0F81562Dh], edx  
mov qword ptr [0F8156F7h], 0DF899564FEh]
```

Операторы размера	
Оператор размера	Размер в байтах
byte	1
word	2
dword	4
fword	6
pword	6
qword	8
tbyte	10
tword	10
dqword	16

## Директивы объявления данных

```
db 67h
db 5dh, 0f6h
db "z" ; то же самое, что и db 7ah
db "w", "k", "y"
dw 8a34h, 0c51h, 8bh
du 9e3ah, 07deh
dw "WE" ; то же самое, что и db 57h, 45h
dd 01F243D5Eh
dd "WEGa" ; то же самое, что и db 57h, 45h, 47h, 61h
dq 1122334455667788h
```

## Синтаксис команд Ассемблера

Для работы с процессорами x86 используются два типа синтаксиса ассемблера — синтаксисы компаний AT&T и Intel. Эти синтаксисы представляют одни и те же команды по-разному. Например, команда в синтаксисе **Intel** выглядит так:

**mov eax,ebx**

В синтаксисе же **AT&T** уже будет несколько иной вид:

**movl %ebx,%eax**

**l** (от *long*) — операнды размером в 4 байта.

Числовые константы имеют следующую форму записи:

**20h** (Intel) **\$0x20** (AT&T, знак доллар в начале)

В среде ОС UNIX более популярен синтаксис типа AT&T. Для UNIX-систем есть два основных ассемблера — это NASM (Netwide Assembler) и FASM (Flat Assembler). Для линейки Windows популярностью пользуются FASM и MASM (Macro Assembler) от фирмы Microsoft, и также существовал еще TASM (Turbo Assembler) фирмы Borland (TASM в настоящее время не поддерживается).

## Наиболее часто используемые команды Ассемблера

**NOP** (NO OPERATION) - Это инструкция, которая при запуске не производит никаких изменений в регистрах, стеке или памяти.

Применяется, если требуется заменить одну инструкцию на другую, более короткую. Лишнее место заполняется NOP'ами.

Также она служит для полного уничтожения другой инструкции, для этого нужно заменить её соответствующим количеством NOP'ов.

**PUSH; POP** - Сохранение и извлечение операндов в стеке

PUSH imm16 / imm32 / r16 / r32 / m16 / m32

POP r16 / r32 / m16 / m32

**PUSHAD** (PUSH All) – размещает в стеке все регистры общего назначения в следующей последовательности: ax, cx, dx, bx, sp, bp, si, di.

PUSH DWORD PTR DS:[401008] – косвенная адресация

**POPAD**

## Наиболее часто используемые команды Ассемблера

**MOV Erd, Ers** (MOV EAX, EBX) или **MOV rd, rs** (MOV AX, BX, MOV AH, BL)

MOV DWORD PTR DS:[400500],EAX – вид записи в отладчике

MOV AX,WORD PTR DS:[405008]

MOV AL, BYTE PTR DS:[405008]



## Команды работы с битами

**Bt < операнд>, <номер бита>** сохраняет значение одного бита из первого операнда во флаге CF.

**BT (Bit Test)** — проверка бита     **BT AX,0** ;Проверка младшего бита AX

**MOV AX, 10h** ; 00010000

**BT AX, 4** ; CF = 1

**BTS (Bit Test and Set)** — проверка бита и установка указанного бита в 1

**BTC (Bit Test and Complement)** — проверка бита и инвертирование указанного бита в операнде

**BTR (Bit Test and Reset)** - тоже самое и сброс указанного бита в операнде

```
mov ax,4 ; ax=100b
```

```
bt ax,2 ;
```

```
jc metka ;
```

```
. . . .
```

```
metka:
```

```
. . . .
```

## Организация циклов

XOR ECX,ECX

ADD ECX,15h; инициализация счетчика циклов

DEC ECX

; тело цикла

TEST ECX,ECX

JNE метка\_начала\_цикла

;или

CMP ECX,0

JNE метка\_начала\_цикла

LOOP ; уменьшение ECX

LOOPZ, LOOPE Цикл повторяется пока флаг Z установлен

LOOPNZ, LOOPNE Цикл повторяется пока флаг Z сброшен

## Команды арифметических операций

ADD EAX, 1 ;или

INC EAX

SUB EAX,ECX

SUB EAX,ECX ;

SUB EAX,DWORD PTR DS:[405000]

MUL ECX ; ECX × EAX результат в EDX:EAX

MUL DWORD PTR DS:[405000]

imul ebp, dword ptr [esi+74], FF800002

mov ax,10234

mov bl,154

div bl ;ah=остаток, al=частное

mov EAX, 0FFFFFFFh ;Младшая часть делимого

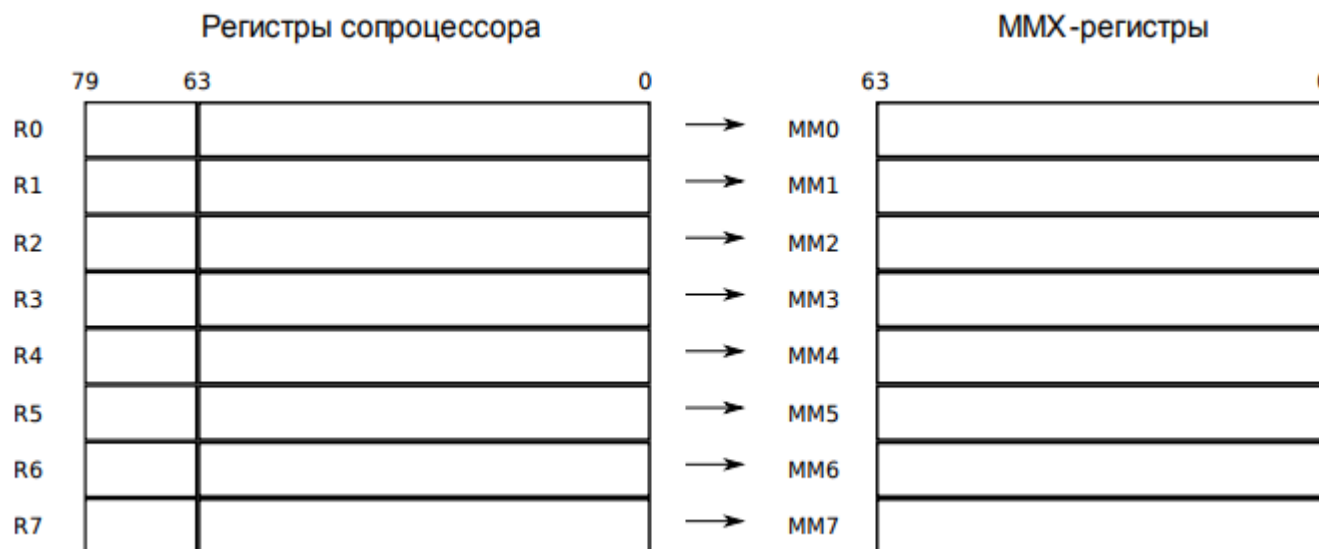
mov EDX,40h ;Старшая часть делимого

mov EBX,256 ;Делитель

div EBX ;Частное в EAX ;Остаток в EDX

## Команды мультимедийных операций

**MMX (Multimedia Extensions** — мультимедийные расширения) —набор SIMD инструкций, разработанного компанией Intel и впервые представленного в 1997 году одновременно с линией процессоров Pentium MMX. Набор инструкций был предназначен для ускорения процессов обработки потоковых аудио и видеоданных. Расширение MMX включает в себя восемь 64-битных РОН MM0—MM7. Для совместимости со способами переключения контекста процессора в существующих ОС Intel была вынуждена объединить в программной модели процессора восемь регистров MMX с мантиссами восьми регистров FPU.



## Типы данных мультимедийных операций

### Типы данных:

**упакованные байты** (8 байтов в одном 64-разрядном регистре) (*packed byte*);

**упакованные слова** (четыре 16-разрядных слова в 64-разрядном регистре) (*packed word*);

**упакованные двойные слова** (два 32-разрядных слова в 64-разрядном регистре) (*packed doubleword*);

**64-разрядные слова** (*quadword*).

## Синтаксис и типы данных мультимедийных команд

**instruction [dest, src]**

**instruction** — имя команды, **dest** место назначения, **src** — источник.

**MOVD mmreg1, reg32 | mem32**

**PADDB mmreg1, mmreg2 | mem64**

Систему команд образуют 57 инструкций, позволяющих параллельно обрабатывать несколько элементов данных. Большинство команд имеют суффикс, который определяет тип данных и используемую арифметику:

**US** (unsigned saturation) — арифметика с насыщением, данные без знака.

**S** или **SS** (signed saturation) — арифметика с насыщением, данные со знаком.

**B, W, D, Q** указывают тип данных. Если в суффиксе есть две из этих букв, первая соответствует входному операнду, а вторая — выходному.

## Расширение мультимедийных команд

**SSE** (*Streaming SIMD Extensions*, потоковое SIMD-расширение процессора) — это SIMD ( *Single Instruction, Multiple Data*, Одна инструкция — множество данных) набор инструкций, разработанный Intel и впервые представленный в процессорах серии Pentium III как ответ на аналогичный набор инструкций 3DNow! от AMD, который был представлен годом раньше.

Технология **SSE** позволяла преодолеть две основные проблемы MMX: при использовании MMX невозможно было одновременно использовать инструкции сопроцессора, так как его регистры были общими с регистрами MMX, и возможность MMX работать только с целыми числами.

**SSE** отличается прежде всего тем, что оперирует с данными вещественного типа, которые используются в геометрических расчётах, то есть, приложениях трёхмерной графики, компьютерных играх, редакторах вроде 3DStudioMax, и многих других.

В **SSE** добавлены восемь (шестнадцать для x64) 128-битных регистров, которые называются xmm0 — xmm7 (-xmm15).

SIMD SSE2; SSE3 и др.

## Примеры MMX-команд Ассемблера

### Команды сложения:

paddb, paddw, paddd;

paddsw — знаковое насыщение paddusb;

paddusw — беззнаковое насыщение;

Приемник — всегда MMX-регистр,  
источник — память или MMX-регистр.

### Команды вычитания:

psubb, psubw, psubd ;

psubsw — знаковое насыщение;

psubusb, psubusw — беззнаковое насыщение.

### MMX-команды сдвига:

psllw, pslld, psllq — логический сдвиг 16-битных, 32-битных или 64-битных операндов влево, при этом освободившиеся младшие разряды заполняются нулями



## Примеры команд с плавающей точкой

Имя	Описание
FADD	$a = b + c$
FNADD	$a = -(b + c)$
FSUB	$a = b - c$
FMUL	$a = b \times c$
FDIV	$a = b / c$
FMIN	$a = \min(b, c)$
FMAX	$a = \max(b, c)$
FAMIN	$a = \min(\text{abs}(b), \text{abs}(c))$
FAMAX	$a = \max(\text{abs}(b), \text{abs}(c))$
FCPS	$a = \text{sign}(b) \mid \text{fraction}(c) \mid \text{exponent}(c)$
FCPE	$a = \text{sign}(b) \mid \text{fraction}(b) \mid \text{exponent}(c)$
FCMPEQ	float vector compare EQ
FCMPNE	float vector compare NE
FCMPLT	float vector compare LT
FCMPLE	float vector compare LE
FCMPBD	float vector compare bounds: $-b \leq a \leq b$
FCMPUN	float vector compare unordered