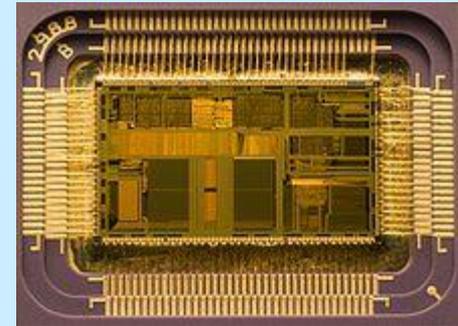




# *Микропроцессорные системы* ©

Мещеряков В.А. , РФФ, ТГУ.



**Система – множество взаимосвязанных элементов.**  
(целое, составленное из частей)

# Микропроцессорные системы (МПС)

МПС – радиоэлектронные системы:

их основой являются **микропроцессоры**, **микроконтроллеры**, ИС;  
они функционируют по заданным программам;  
предназначены для **получения**, **хранения**, **обработки**, **передачи**  
информации и **управления** процессами.

**Микропроцессор** – цифровая интегральная микросхема **исполняющая код программы**

**Интегральная микросхема** – радиоэлектронная схема изготовленная на полупроводниковом кристалле помещённом в защитный корпус.

**МПС - совокупность **технических** и **программных** средств.**

Технические средства (Hard)	Программные средства (Soft)
	<pre data-bbox="1174 972 1474 1193">i = 0 do while (i.lt.10) write(*,*) i i=i+1 enddo</pre>

**Периферийные устройства** – технические устройства выполняющие функции получения, визуализации, хранения, передачи информации

# Основные типы микропроцессорных систем

**1. Компьютеры** – **универсальные** высокопроизводительные МПС для решения задач цифровой обработки информации.



**2. Микроконтроллеры** – **специализированные** МПС

встраиваемые в системы управления процессами.

Они выполнены в виде одной интегральной схемы

<http://www.gaw.ru/html.cgi/txt/app/micros/avr>



Микроконтроллеры  
встраивают в различные  
устройства



**3. Программируемые логические интегральные схемы (ПЛИС, ПЛМ)** – программируемые электронные цифровые конструкторы .

<http://habrahabr.ru/post/80056/>

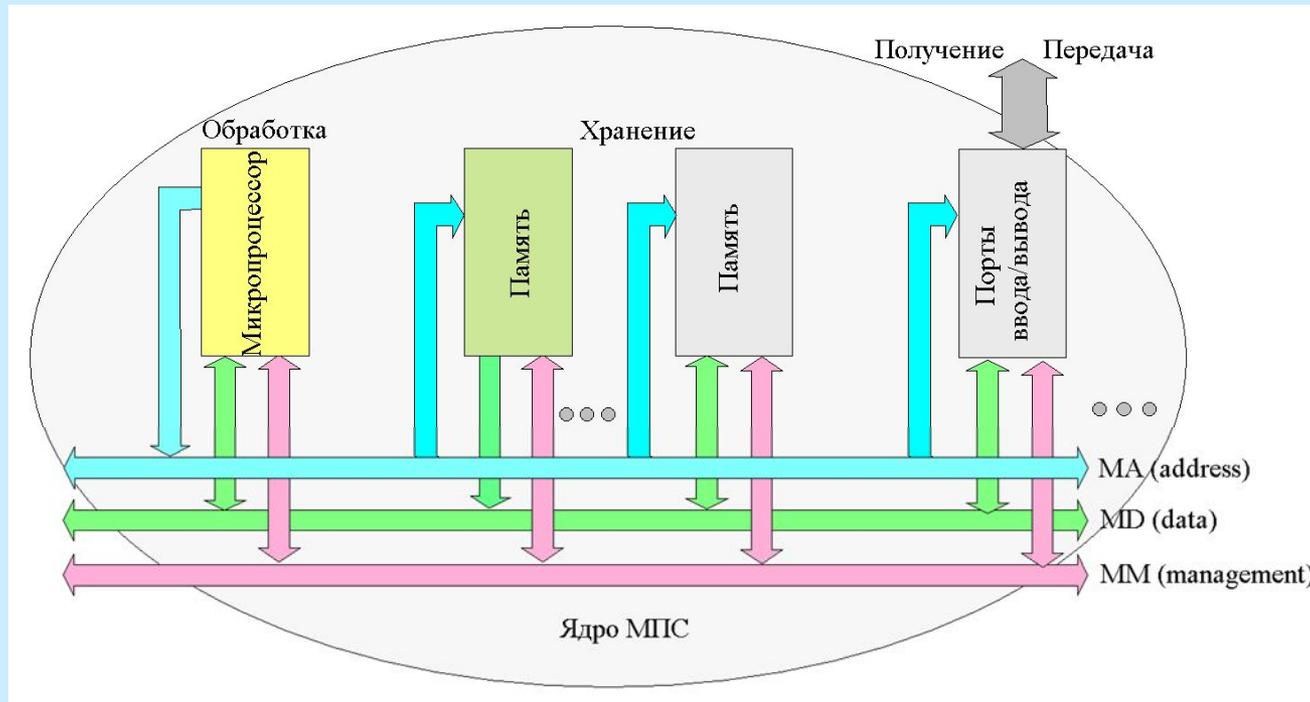
Максфилд Клайв. Проектирование на ПЛИС: Курс молодого бойца. –М: Додэка-XXI, 2007



# Структура микропроцессорной системы

МПС – совокупность связанных информационными **магистралями** функциональных **модулей**.

## Магистрально – модульная структура МПС.

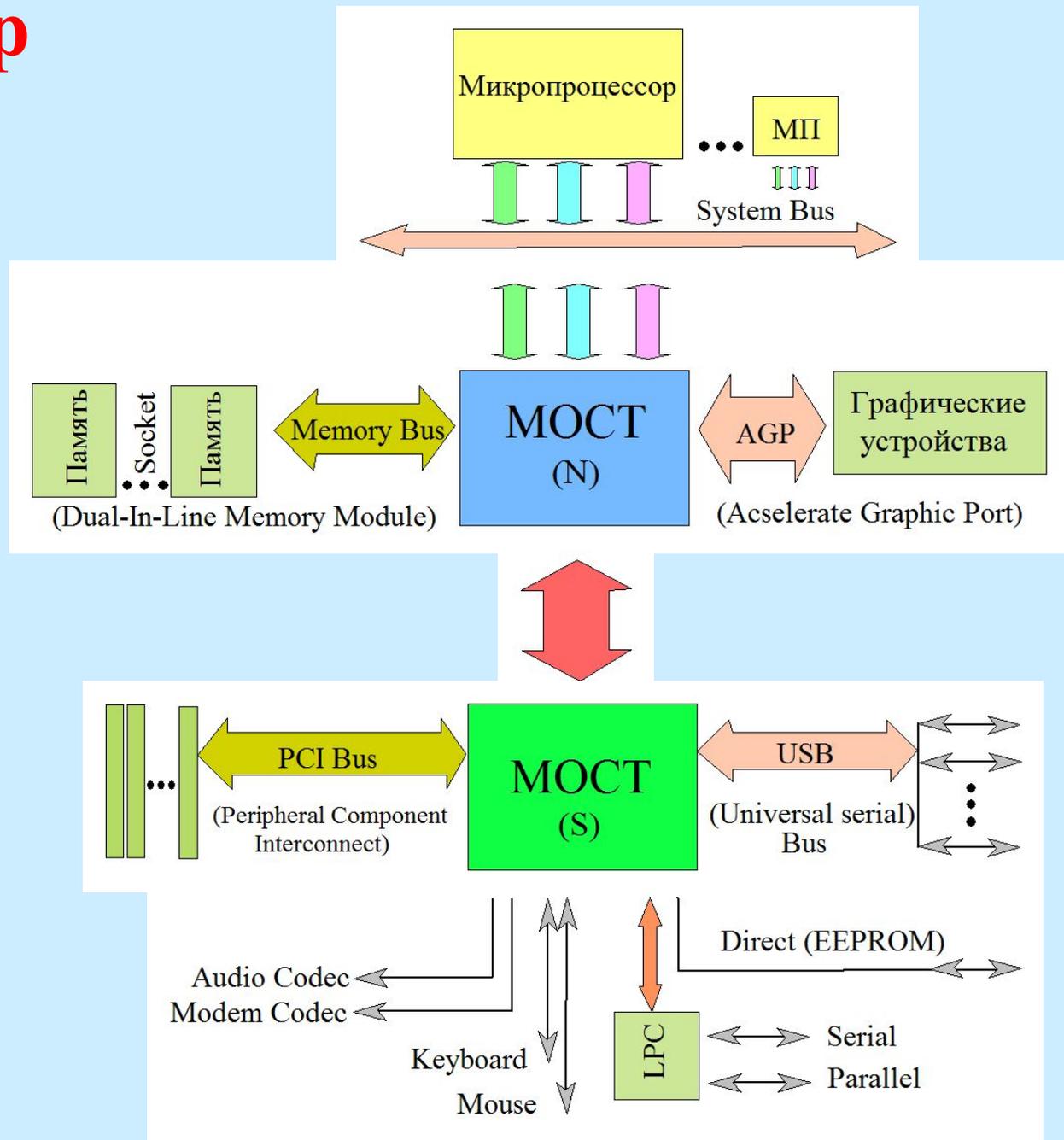


Wi-Fi, Bluetooth ...

**Магистрали** – электрические, оптические, радиоволновые или какие либо еще каналы связи, предназначенные для обмена информацией между модулями.

**Интерфейс** – совокупность правил (требований) определяющих электрическую, логическую и конструктивную совместимость модулей посредством магистралей.

# Компьютер



**Микропроцессор** – цифровая интегральная микросхема исполняющая код программы ;

**System Bus** – основная магистраль системы (**параллельного типа**);

**Северный мост (Hub)** – разветвитель-концентратор информации между основной магистралью системы и другими магистралями (например: Memory Bus, AGP) (ПЛИС);

**Memory Bus** –магистраль для связи с памятью (**параллельного типа**);

**Память** – модуль для хранения информации (**оперативная память**);

**Socket** – электрические разъемы для подключения модулей;

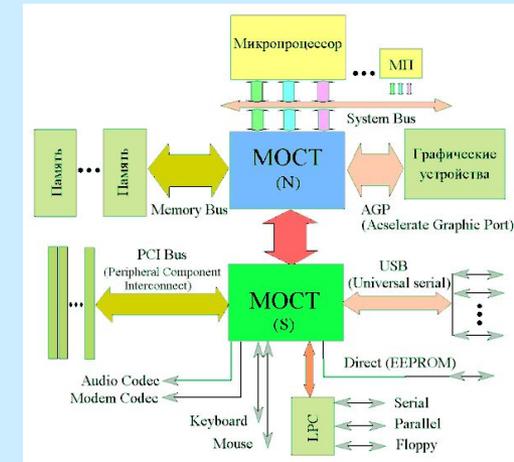
**AGP** – магистраль для подключения графических устройств отображения информации (**параллельного типа**);

**Южный мост (Hub)** – разветвитель-концентратор информации между основной магистралью системы и вспомогательными магистралями (ПЛИС);

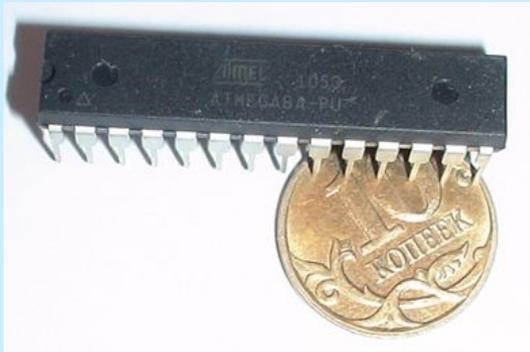
**USB** – магистраль системы для связи с различными внешними устройствами (**последовательного типа**);

**PCI Bus** –магистраль системы для связи с различными внешними устройствами (**параллельного типа**);

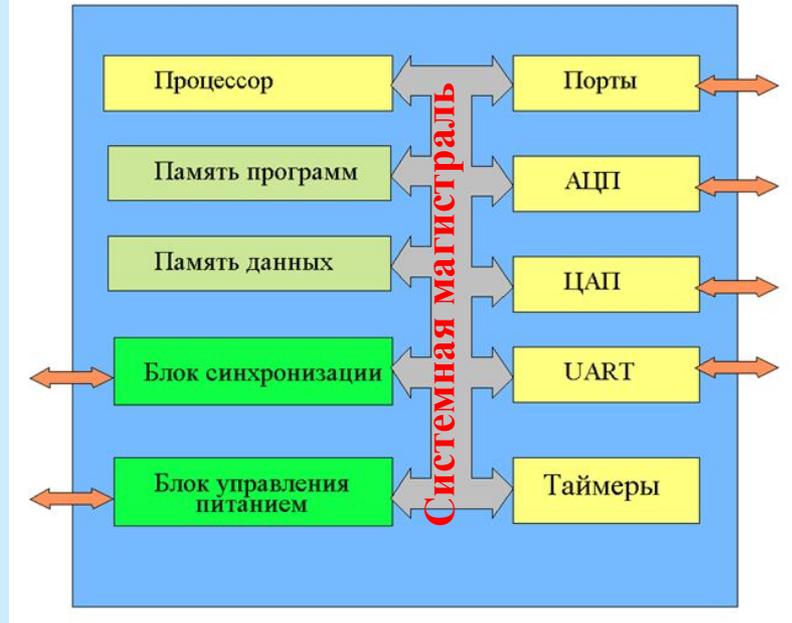
**Keyboard, mouse, ..., floppy** – периферийные устройства и их для связи с компьютером.



# Микроконтроллер



## Кристалл полупроводника



**Процессор** – модуль исполняющий код программы ;

**Память программ** – модуль для хранения **программы** (энергонезависимая память);

**Память данных** – модуль для хранения **данных** участвующих в обработке;

**Блок синхронизации** – модуль, обеспечивающий синхронную работу системы;

**Блок управления питанием** – модуль, обеспечивающий энергосбережение;

**Порт** – модуль для электрической связи контроллера с внешней средой.

**АЦП, ЦАП** – преобразователи аналоговой и цифровой информации;

**UART** – формирователь пакетов **цифровых данных для обмена** с внешней средой;

**Таймер** – устройство контроля интервалов времени.

# **Информация**

**Кодирование информации**

**Мера количества информации**

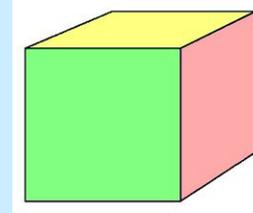
## Понятие информации

**Классификация** - это **логическое** разделение множества на части по ряду признаков.  
- это осмысленный порядок вещей.

(находиться в порядке – **быть пронумерованными !!!** )

Рассмотрим простые системы:

**Системы описываются**  
**множеством элементов и/или состояний.**



Предположим, что система может находиться в одном из  $N$  возможных состояний.

Сведения о состоянии системы – **информация.**

**Информация** – сведения о состоянии системы,  
зафиксированные на каком-либо материальном носителе.

**Информацию:**

**получают;**

**записывают;**

**хранят;**

**обрабатывают;**

**используют;**

**передают.**

датчики (сенсоры), преобразователи, каналы связи;

кодеры ;

запоминающие устройства (память);

процессоры;

декодеры, преобразователи, исполнители;

передатчики.

# Информация характеризуется:

**Количеством - L** (Lenght - длина, размер).

**Содержанием - D** (Date - данные)

Для **записи** содержания нужен **алфавит**

(запись это **кодирование** с помощью алфавита)

Алфавит – **упорядоченное** множество символов (букв, знаков и других **графем**) для системы письма.

**Самый простой алфавит состоит из 2 символов.**

Информация о состоянии системы «игральная кость»

Примеры алфавитов:

0, 1;

□, ■;

♂, ♀.

Состояние	Содержание	
	1	
	10	
	11	
	100	
	101	

# Для хранения информации необходимо знать её количество ( $L$ )

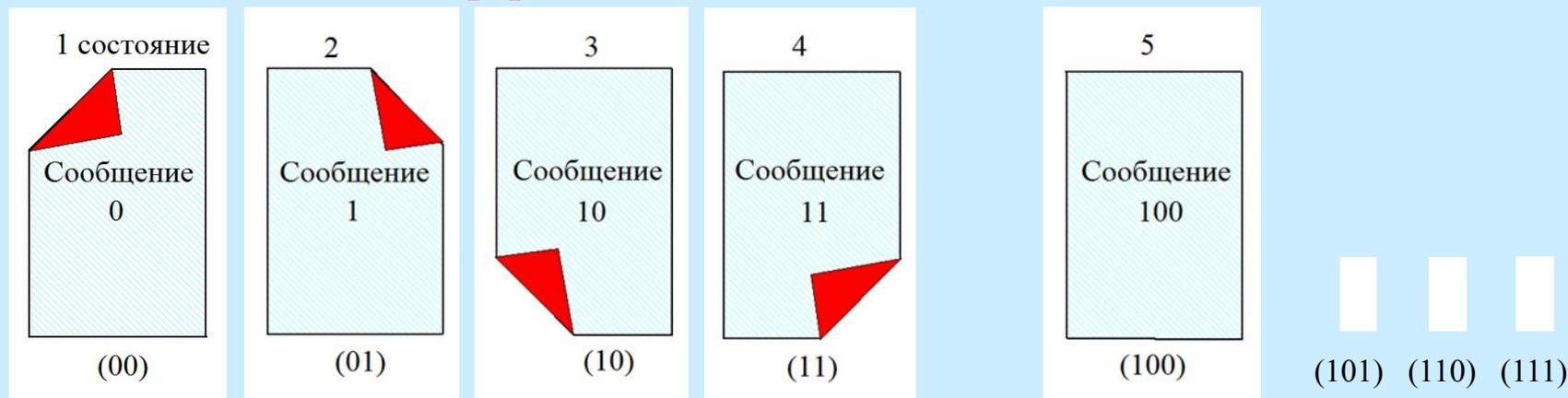
## Мера информации

Рассмотрим систему находящуюся в одном из 4 состояний.

Например: лист бумаги с загнутым углом ( или предмет в углу комнаты).

Нас интересует  $D$  - информация о состоянии листа  
и количество  $L$  этой информации.

Если выберем алфавит: 0,1



Запись длиной в  $L=2$  символа может содержать информацию об одном из 4 состояний

Запись длиной в  $L=N$  символов может содержать информацию об одном из  $2^N$  состояний

Мера веса килограмм [кг],

мера количества информации – ?.

Количество информации определяющее состояние системы связано с вероятностью нахождения системы в этом состоянии

Если состояния равновероятны, то

$$P = 1 / N.$$

**Количество информации** определяется формулой:

$$L = \text{Log}_2 1/P = \text{Log}_2 N \quad (\text{Хартли})$$

Минимальное число состояний системы  $N=2$ ,  $P=1/2$

(системы: выключатель, монета).

$$L = \text{Log}_2 1 / 0.5 = 1 \quad [\text{бит}]$$

**[бит] (binary digits)** 1 бит – 1 место – 1 позиция – 1 разряд ...

Клод Шеннон. **Бит** – количество информации, занимаемое ответом на вопрос, поставленный в альтернативной форме.

(H<sub>2</sub>O это химическая формула воды? Да.).

Система «игральный кубик» имеет шесть возможных состояний,  $N=6$  и  $P=1/6$ .

Длина сообщения  $L = \text{Log}_2 1 / 6 = 2.585$  [бит]

Всегда округляем в большую сторону,  $L = 3$  [бит]



Энтропия (S) - мера беспорядка (неопределенности) системы.  $S$  [эрг/град].

$$1 \text{ эрг} = 0.0000001 \text{ Дж}$$

Если беспорядком системы назвать всё множество возможных её состояний ( $N$ ), то можно связать меру беспорядка  $S$  и меру количества информации  $L$ :

$$S = k \ln N = \quad - \text{ формула Планка-Больцмана,}$$

$$k = 1.38 \text{E-}24 \text{ [эрг/град]}$$

$$S_{\text{[эрг/град]}} = k \ln 2^L \quad L \text{ [бит].}$$

Чем меньше вероятность предсказания состояния системы, тем больше её энтропия.

$$8 \text{ бит} = 1 \text{ байт.}$$

$$1024 \text{ бит} = 1 \text{ килобит} = 2^{10} \text{ бит}$$

$$1 \text{ мегабит} = 2^{20} \text{ бит}$$

Процедура классификации:



Кодирование информации

10011011 В  
10101001 e

## Хранение информации (содержание D и количество L задано).

Некоторые параметры минерала изумруда

Параметры	значение	тип параметра
химическая формула	$Be_3Al_2Cr_2Si_6O_{18}$	символьный
число атомов берилла в молекуле	3	целый без знака
плотность	2.8 [г/см <sup>3</sup> ]	вещественный
цвет	зеленый	тип цвет
...		

**Тип** ( в программировании) — дискретное множество или упорядоченный набор различных значений и множество операций над ними ...

Для **кодирования** информации используют **минимальный алфавит (0,1)**

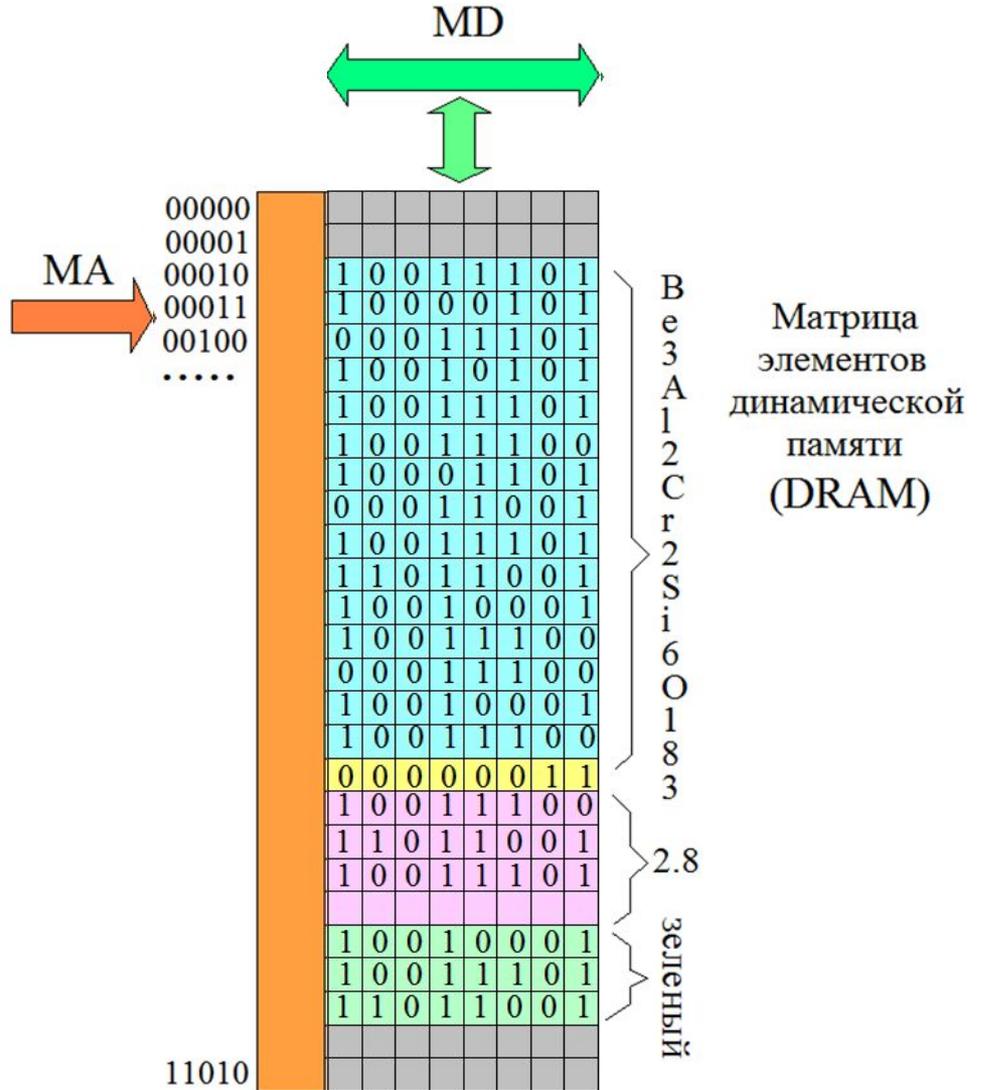
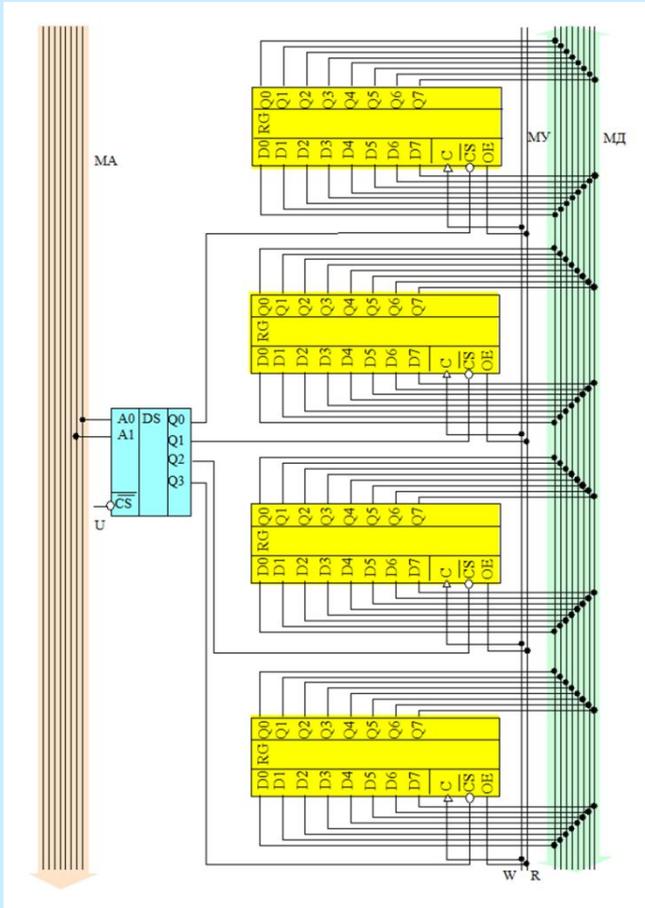
Предварительно необходимо выбрать хранилище, способ кодирования и записи информации.

Лист бумаги - неединственное хранилище кодов информации

Информацию (данные) хранят в запоминающих устройствах (**в памяти**). 13

# Структура хранилища информации

При создании хранилища информации решается ряд вопросов:  
 об объеме хранилища;  
 о его технической организации;  
 о способах доступа к нему;  
 и т.д.



# Системы счисления

**Система счисления** – совокупность:  
 символов (цифр, знаков);  
 правил записи чисел с помощью цифр;  
 законов арифметики.

Системы счисления	
позиционные	непозиционные

0	1	2	3	+	.
4	5	6	7	-	
8	9	A	B	*	
C	D	E	F	/	

Римская:  
 Символы I, V, X, M, ...  
 Числа: XXV, VII, III, ...

**Основание системы** – количество цифр используемое для записи чисел (B).

B = 2	0	1														
B = 3	0	1	2													
B = 4	0	1	2	3												
B = 8	0	1	2	3	4	5	6	7								
B = 10	0	1	2	3	4	5	6	7	8	9						
B = 16	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Количество	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15



60 лет РФФ  
 (1953 - 2013)

$$101 \quad 111100 = 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$$

$$12 \quad 2020 = 2 \cdot 3^3 + 0 \cdot 3^2 + 2 \cdot 3^1 + 0 \cdot 3^0$$

$$11 \quad 330 = 3 \cdot 4^2 + 3 \cdot 4^1 + 0 \cdot 4^0$$

$$5 \quad 74 = 7 \cdot 8^1 + 4 \cdot 8^0$$

$$5 \quad 60 = 6 \cdot 10^1 + 0 \cdot 10^0$$

$$5 \quad 3C = 3 \cdot 16^1 + 12 \cdot 16^0$$

**Вопрос.** Можно ли с помощью цифр из каждой системы счисления записать информацию о числе пальцев на руке и прошедшую годовщину РФФ?

## Запись чисел (кодирование)

Число:  $\pm d^{(i)} d^{(i-1)} \dots d^{(1)} d^{(0)} d^{(-1)} d^{(-2)} \dots d^{(-m)}$       Пример:  $\pm 4121.937$   
 $\pm 1001.001$   
 $\pm 1F5C.0E1$

позиция цифры

## Перевод в десятичное:

$$M = \left[ d^{(i)} B^{n-1} + d^{(i-1)} B^{n-2} + \dots + d^{(1)} B^1 + d^{(0)} B^0 \right] + \left[ d^{(-1)} B^{-1} + \dots + d^{(-m)} B^{-m} \right]$$

$$1101.001_{(2)} \quad 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^0 + 1 \cdot 2^{-3} = 13.125_{(10)}$$

$$3170.002_{(8)} \quad 3 \cdot 8^3 + 1 \cdot 8^2 + 7 \cdot 8^1 + 2 \cdot 8^{-3} = X_{(10)}$$

# Способы перевода чисел

Таблица перевода

(10)	(2)	(8)	(16)
0	00000	00	0
1	00001	01	1
2	00010	02	2
3	00011	03	3
4	00100	04	4
5	00101	05	5
6	00110	06	6
7	00111	07	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16

Перевод делением

$136_{(10)} = 10001000_{(2)} = 1*2^7 + 1*2^3 = 128 + 8 = 136$

## Арифметика двоичной системы исчисления.

		a + b				a * b		
a \ b	0	1	a \ b	0	1	a \ b	0	1
0	0	1	0	0	0	0	0	0
1	1	0	1	0	1	1	0	1

$136_{(10)} = 210_{(8)} = 2*8^2 + 1*8^1 = 136$

$136_{(10)} = 88_{(16)} = 8*16^1 + 8*16^0 = 136$

$123_{(10)} = 7B_{(16)} = 7*16^1 + 11*16^0 = 123$

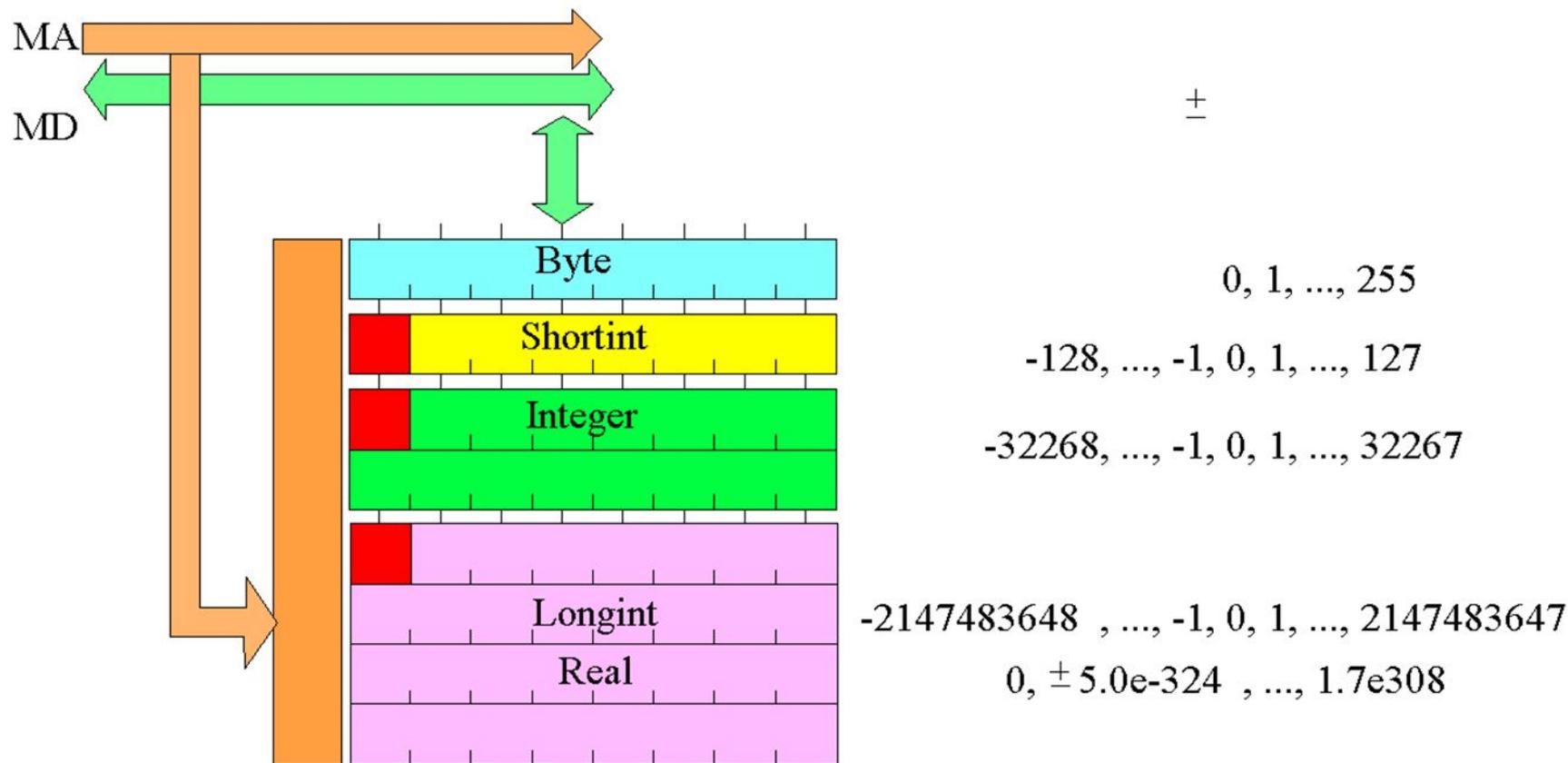
Перевод для (2), (8), (16) систем

$136_{(10)} = \overbrace{10001000}^{(2)}_{\substack{2 \ 1 \ 0 \\ (8)}} \quad 136_{(10)} = \overbrace{10001000}^{(2)}_{\substack{8 \ 8 \\ (16)}}$

# Кодирование чисел.

Языки программирования высокого уровня оперируют данными различных типов.

Например: - Язык Паскаль оперирует числами:



Как переводят целые числа в двоичное представление и помещают на место в памяти?

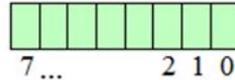
Существует ряд способов (видов кодирования):

прямой код; обратный код; дополнительный код; двоично-десятичный код;

# Прямой код

Алгоритм кодирования числа  $M_{(10)}$

Размер хранилища 8 бит



Тип **Byte**

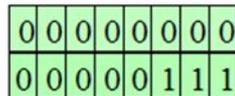
- 1)  $M_{(10)} \rightarrow M_{(2)}$        $7_{(10)} = 111_{(2)}$
- 2) добавляем ведущие нули       $00000111_{(2)}$
- 3) помещаем в хранилище      

0	0	0	0	0	0	1	1	1
---	---	---	---	---	---	---	---	---

Диапазон: 0 - 255

Тип **Word**

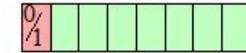
Размер хранилища 2 байта



Диапазон: 0 - 65535

Тип **Shortint**

- 1) Кодируем знак: 0 (+), 1 (-)



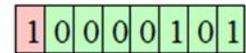
- 2)  $|M_{(10)}| \rightarrow M_{(2)}$

$$|-5_{(10)}| = 101_{(2)}$$

- 3) добавляем ведущие нули

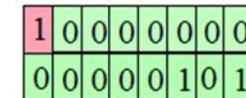
$$10000111_{(2)}$$

- 4) помещаем в хранилище



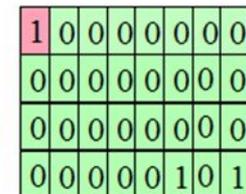
Диапазон: -127 ... +127

Тип **Integer**



Диапазон: -32767 ... + 32767

Тип **Longint**      4 байта



Диапазон:  
-2147483647 ... + 2147483647

# Обратный код

(только для отрицательных чисел)

## Алгоритм кодирования

1)  $|M_{(10)}| \longrightarrow M_{(2)} \quad | -6_{(10)} | = 110_{(2)}$

2) добавляем ведущие нули  $00000110_{(2)}$

3) инвертируем  $11111001_{(2)}$

4) помещаем в хранилище 

1	1	1	1	1	0	0	1
---	---	---	---	---	---	---	---

# Дополнительный двоичный код

## Алгоритм кодирования

1)  $|M_{(10)}| \longrightarrow M_{(2)} \quad | -6_{(10)} | = 110_{(2)}$

2) добавляем ведущие нули  $00000110_{(2)}$

3) инвертируем  $11111001_{(2)}$

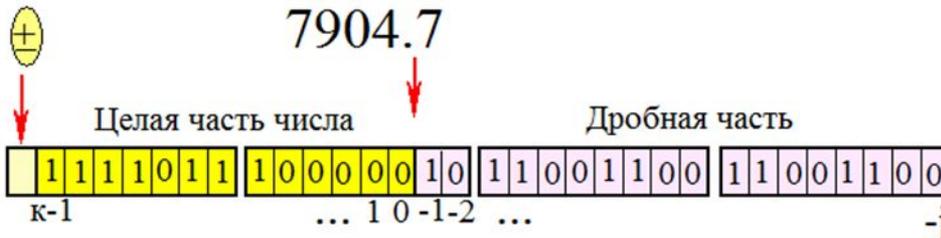
4) добавляем 1  $11111001_{(2)} + 1$

5) помещаем в хранилище 

1	1	1	1	1	0	1	0
---	---	---	---	---	---	---	---

# Фиксированная точка

(Место - 4 байта)

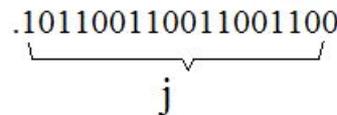
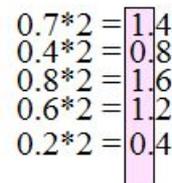


Целую часть переводим прямым кодом.

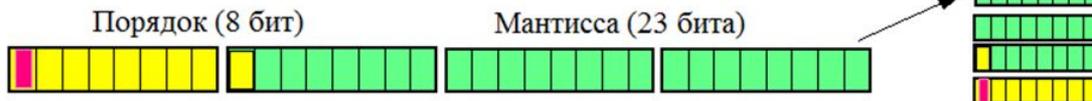
## Алгоритм перевода дробной части

Пример:  $0.7_{(10)}$

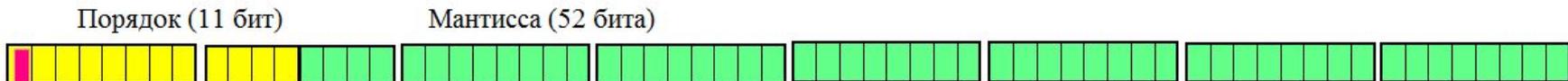
- умножаем на 2
- запоминаем целую часть
- вычитаем 1
- если не 0, то переходим на пункт 1)
- запомненные целые части переписываем в память



Одинарная точность (4 байта)



Двойная точность (8 байт)



# Кодирование вещественных чисел.

## Плавающая точка

(Место - 4 байта)

Научная нотация:

$$1320 \cdot 10^{-2} = 132 \cdot 10^{-1} = 13.2 = 1.32 \cdot 10^{+1} = 0.132 \cdot 10^{+2} = 0.0132 \cdot 10^{+3} \dots$$

## Нормализация:

$$13.2 = 0.132 \cdot 10^{+2},$$

$$-0.00573 = -0.573 \cdot 10^{-2}$$

## Мантисса. Порядок.

$$\begin{matrix} 132 & 2, \\ 573 & -2 \end{matrix}$$

Алгоритм перевода:

Пример:  $-0.0390625_{(10)}$

- переводим в двоичную форму;  $-0.0000101_{(2)}$
- нормализуем;  $-0.101E-100$

- выделяем части для записи
- сдвигаем мантиссу влево на 1 разряд с удалением 1
- упаковываем в хранилище

# Кодирование символов.

## Таблица кодов ASCII

(American Standard Code for Information Interchange)

10	16	2	Симв	10	16	2	Симв	10	16	2	Симв	10	16	2	Симв
0	0	00000000	NUL	32	20	00100000	(sp)	64	40	001000000	@	96	60	01100000	`
1	1	00000001	SOH	33	21	00100001	!	65	41	001000001	A	97	61	01100001	a
2	2	00000010	STX	34	22	00100010	"	66	42	001000010	B	98	62	01100010	b
3	3	00000011	ETX	35	23	00100011	#	67	43	001000011	C	99	63	01100011	c
4	4	00000100	EOF	36	24	00100100	\$	68	44	001000100	D	100	64	01100100	d

## Таблица кодов

Windows (Win-1251)

10	16	2	Симв	10	16	2	Симв	10	16	2	Симв	10	16	2	Симв
0	0	00000000	NOP	32	20	00100000	(sp)	64	40	001000000	@	96	60	01100000	`
1	1	00000001	SOH	33	21	00100001	!	65	41	001000001	A	97	61	01100001	a
2	2	00000010	STX	34	22	00100010	"	66	42	001000010	B	98	62	01100010	b
3	3	00000011	ETX	35	23	00100011	#	67	43	001000011	C	99	63	01100011	c

### Распространенные кодировки:

ASCII, ISO 8859, Windows -1250 -1258, KOI8, ISO-2022, Unicode.

### Форматированное кодирование.

Помимо кода символа, для его отображения используют тип шрифта.

Поэтому символ может иметь различное **графическое изображение**.

# Кодирование изображений.

**Графика** - вид изобразительного искусства, средствами которого являются линии, штрихи, пятна и точки на «холсте».

**Компьютерная графика** – техника создания, кодирования и обработки изображений.

**Типы:** растровая, векторная, фрактальная.

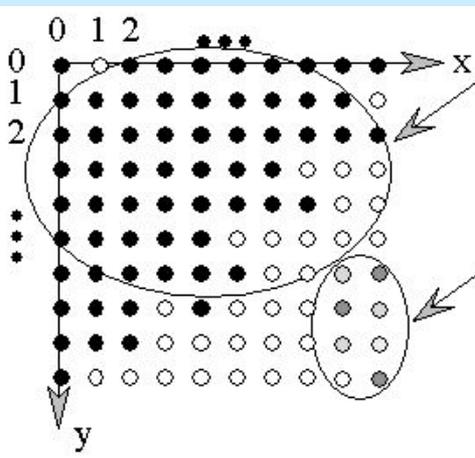
## Растровая графика

**Растр** (rastrum, лат.) – массив **пикселей** (мельчайших единиц изображения).

**Picse**l - **p**ictures **e**lement.

**Разрешение растра** – число пикселей по вертикали и горизонтали ( $N * M$ ).

**Атрибуты пикселя:** координаты, цвет.

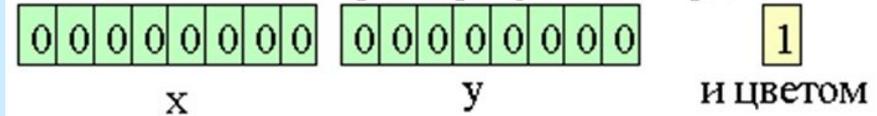


**Монохромное**

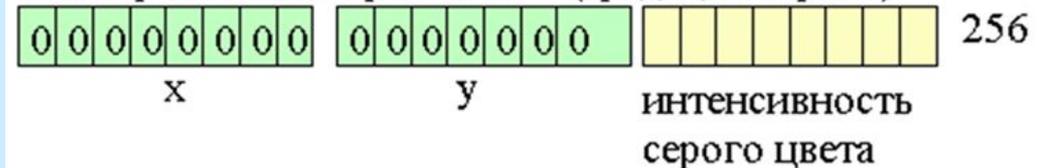
**Полихромное**

Монохромное изображение (двухцветное).

Каждый пиксел характеризуется координатами



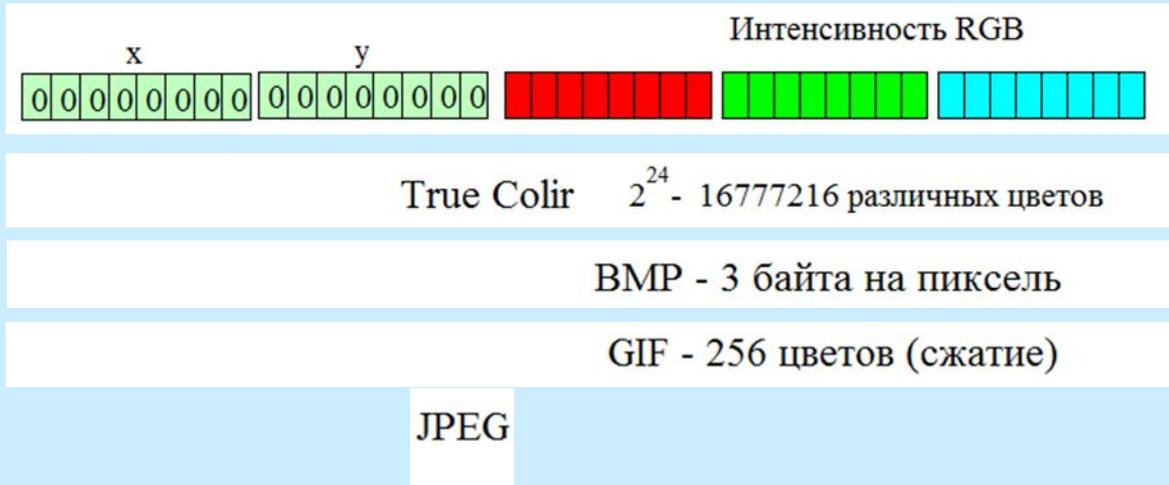
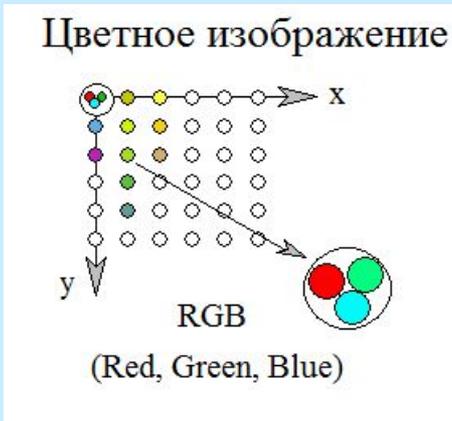
Полихромное изображение (градации серого)



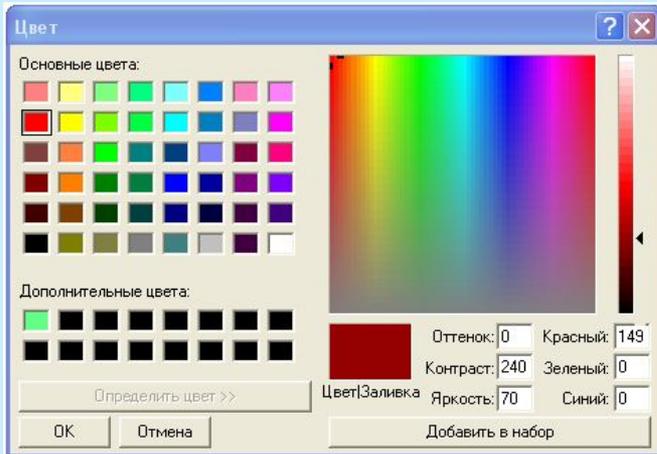
Сотовый телефон имеет монохромный экран с 8 градациями серого цвета и разрешением растра  $100 * 50$ . Определить объем памяти для хранения кадра изображения.

# Растровая графика

# Кодирование изображений.

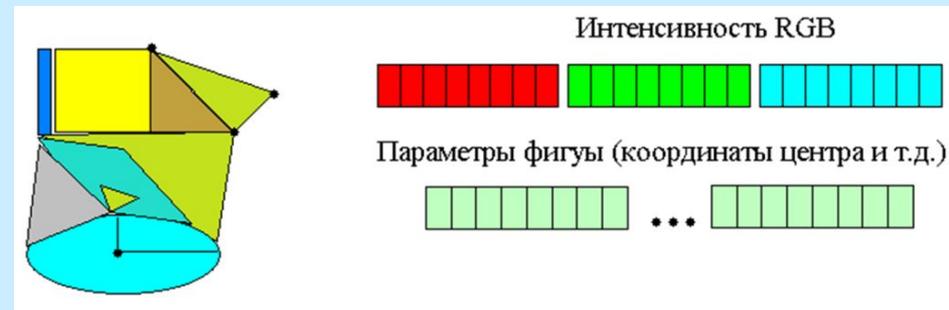


## Шкала интенсивности цвета (палитра)

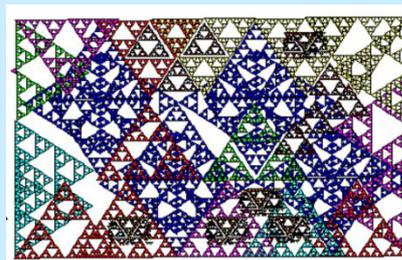
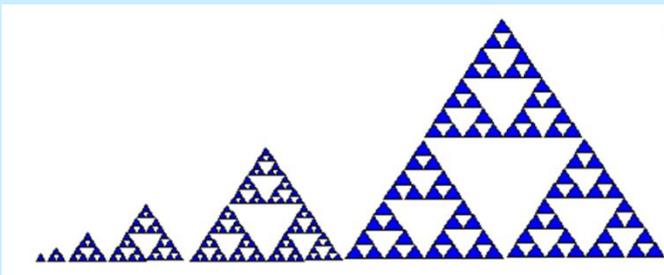


$$2^8 = 256$$

# Векторная графика



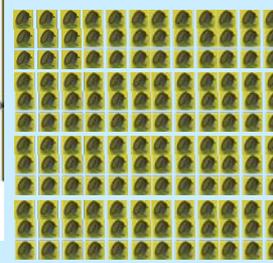
# Фрактальная графика



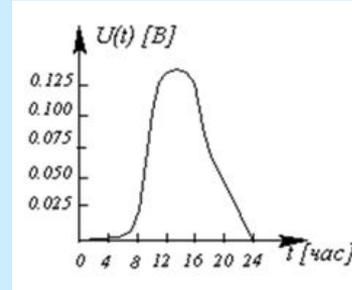
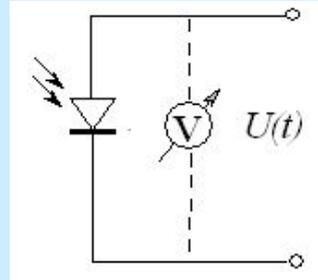
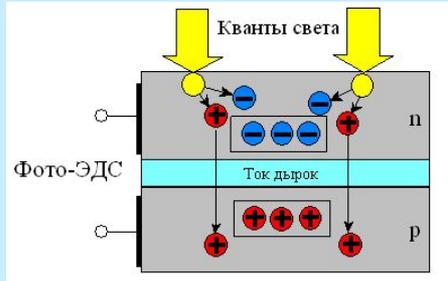
См. «Канторово множество»

# Получение видеoinформации.

## Фотодиод

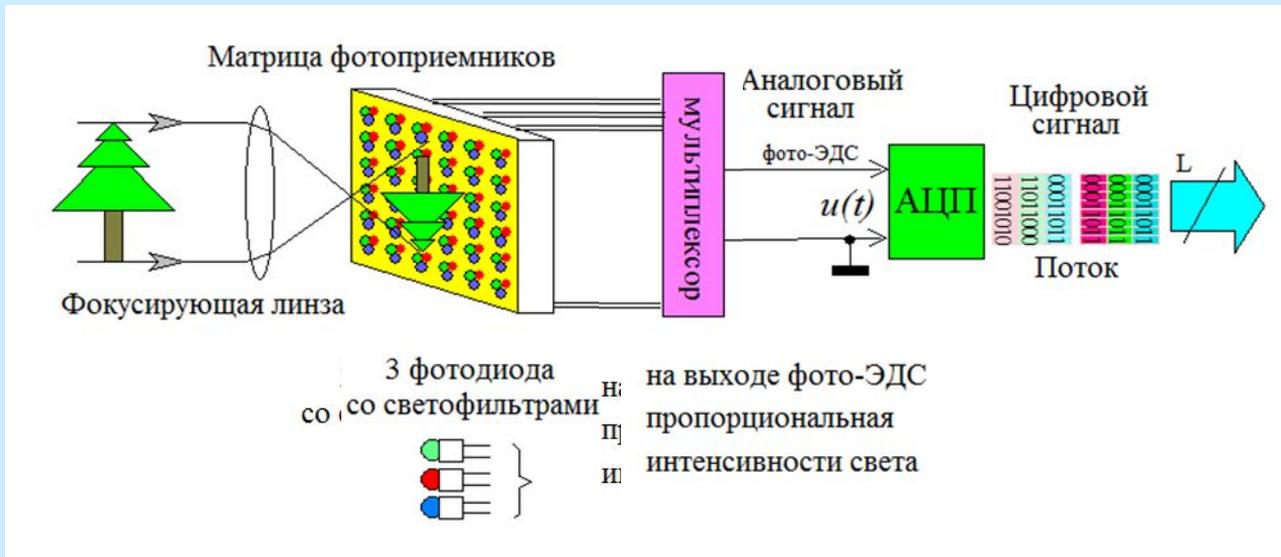


## Кодирование изображений.



## Аналоговый сигнал

Сигнал  $U(t)$ , у которого информационный параметр  $U$  изменяется в интервале времени наблюдения непрерывно, называют **аналоговым**.



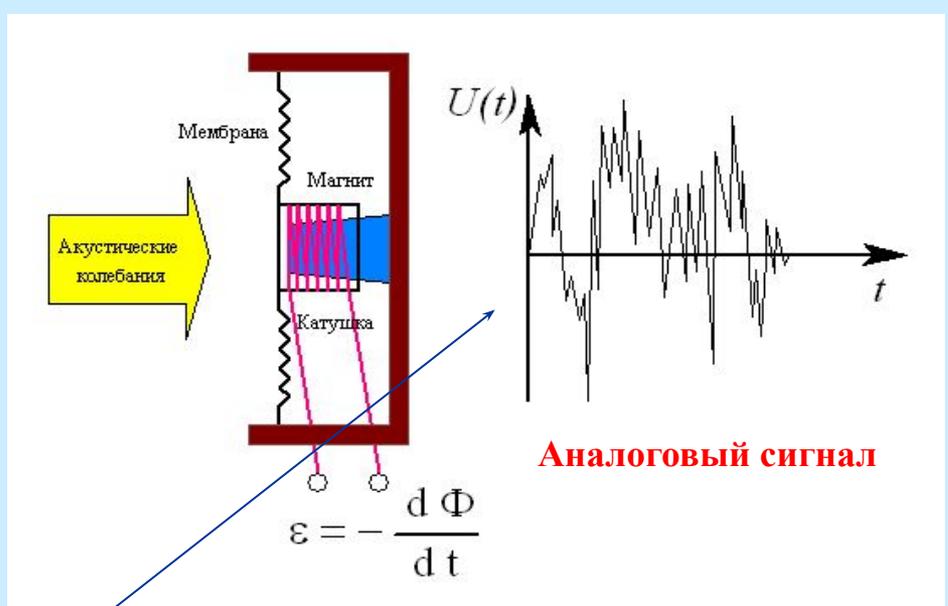
## В память (данные)

Сигнал, у которого информационный параметр  $U$  представлен набором дискретных значений (таблицей значений), называют **цифровым**

# Кодирование звука.

## Получение аудиоинформации.

### Микрофон



Микрофон

Аналоговый сигнал

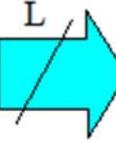
$u(t)$



1001010  
10010101  
1000110  
10010101  
1000110

Поток

Цифровой сигнал



В память  
(данные)

## Вопрос из тестов.

Если числа в двоичной системе счисления имеют вид 111 и 1001, то их сумма в десятичной системе счисления равна...

1. 222
2. 14
3. 16
4. 8

# Аналоговые и цифровые сигналы

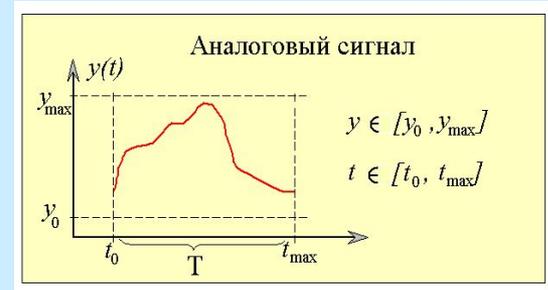
Обмен информацией осуществляется с помощью **сигналов**.

$U(t)$  – амплитуда;  $\varphi(t)$  – фаза;  $\omega(t)$  – частота.

Сигнал, у которого информационный параметр в интервале времени наблюдения изменяется непрерывно и может принимать любые значения, называют **аналоговым** ( $y(t)$ ).

Сигнал, у которого информационный параметр задан таблицей значений (набором дискретных значений), называют **цифровым** ( $\{y_k = y(t_k)\}_{k=0,m}$ ).

Устройства формирующие, передающие, принимающие и преобразующие информацию в аналоговой форме называют **аналоговыми устройствами**.



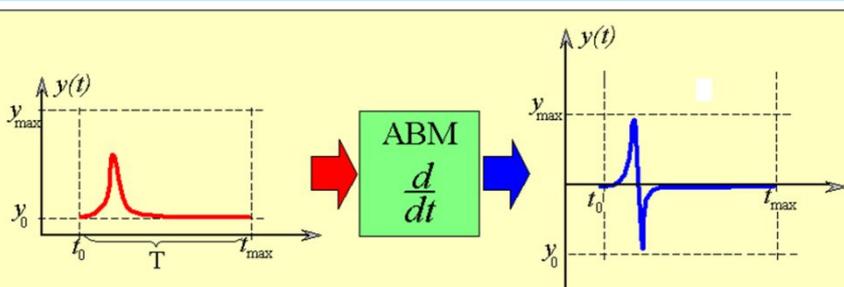
Аналоговая форма представления информации

$t$	$y$
0	0
1	4
2	5
3	6
4	6
$\vdots$	$\vdots$
128	1

Цифровая форма представления информации

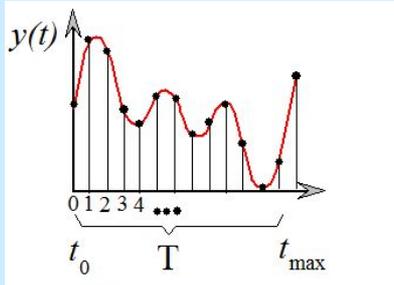
Положительные качества АВМ: **высокое быстродействие** (реальный масштаб времени); **малые габариты**.

Недостатки АВМ: влияние **внешних и внутренних шумов**; необходимость использования **прецизионных электронных компонентов**.



# Преобразование аналогового сигнала в цифровой

1. **Дискретизация сигнала** – разбиение интервала времени наблюдения сигнала на равные части  $\Delta T$ , с последующей нумерацией частей.

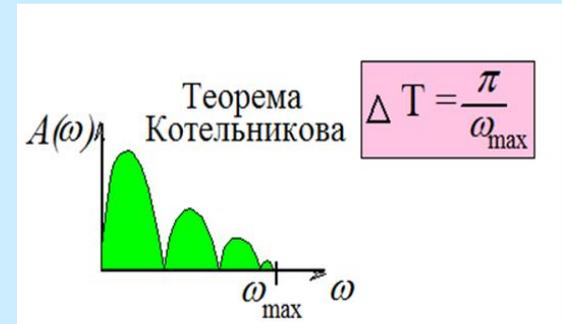
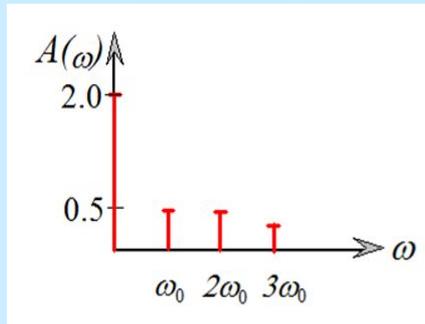


Дискретизованный сигнал

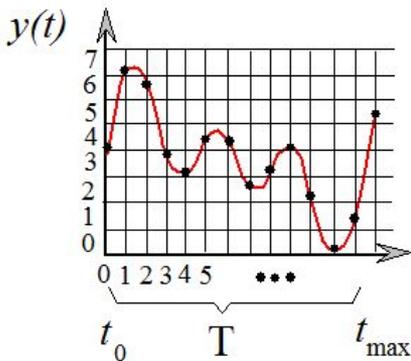
$$\{y_k = y(t_k)\}_{k=0}^m$$

К вопросу о частоте дискретизации (как выбрать  $\Delta T$ ?).

$$y(t) = \sin(\omega_0 t)/2 + \sin(2\omega_0 t)/2 + \sin(3\omega_0 t)/3 + 2$$

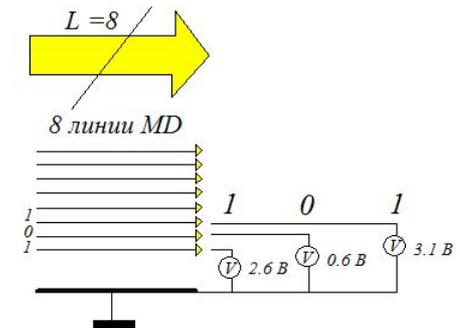
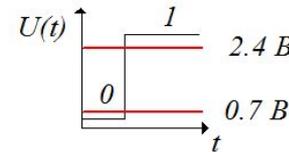


2. **Квантование сигнала** – построение шкалы, по которой реальному параметру сигнала  $y_k = y(t_k)$  ставится в соответствие числовой эквивалент (номер деления шкалы).

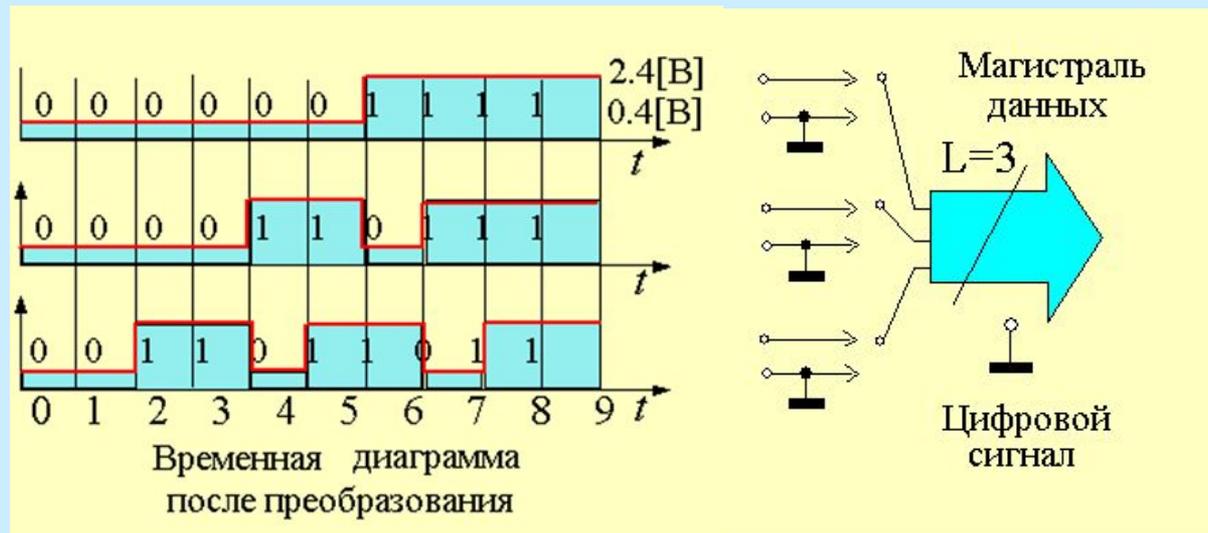
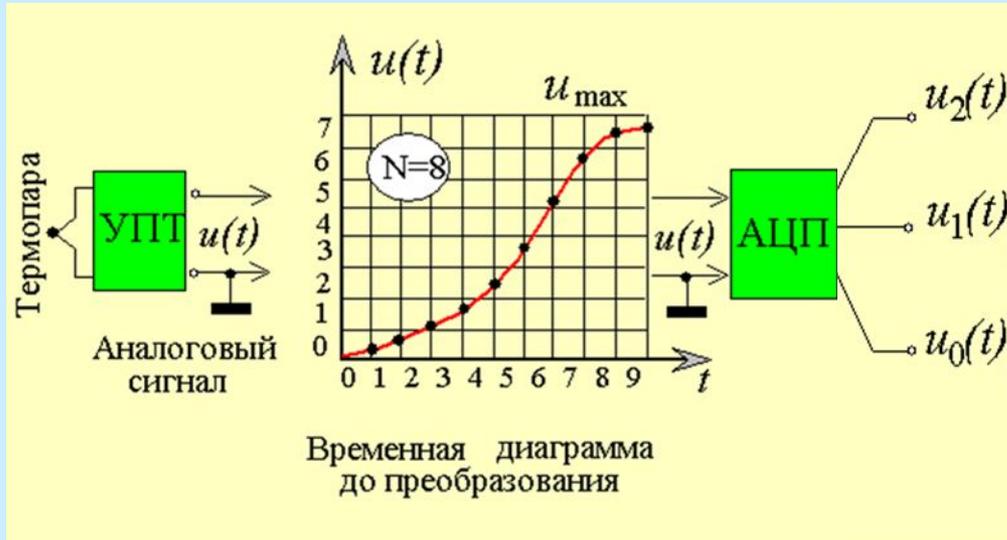


$y(t)$   
 00000111  
 00000110  
 00000101  
 00000100  
 00000011  
 00000010  
 00000001  
 00000000

$t$	$y$	двоичное представление
0	4	00000100
1	7	00000111
2	6	00000110
3	3	00000011
4	3	00000011
5	4	00000100
⋮	⋮	⋮
⋮	⋮	⋮

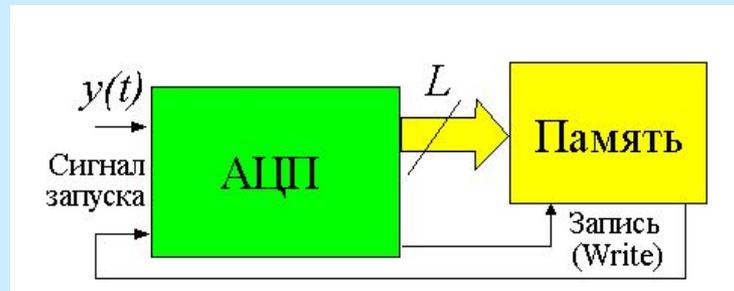
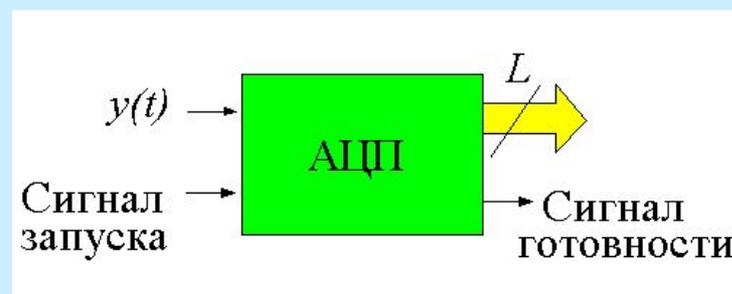


# Преобразование аналогового сигнала в цифровой



# Аналого-цифровые преобразователи (АЦП)

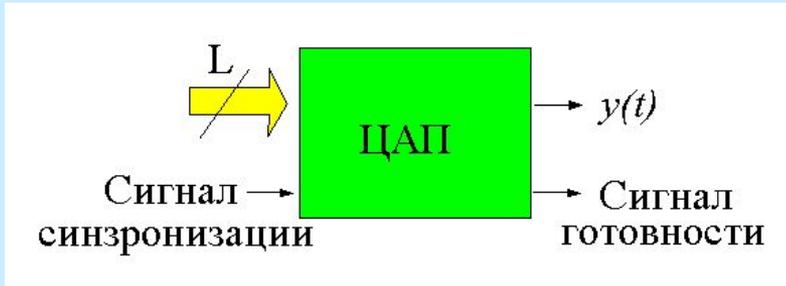
Аналого-цифровой преобразователь – электронное устройство, осуществляющее преобразование **аналогового** электрического сигнала  $y(t)$  в **двоичный цифровой**  $L$ -разрядный электрический сигнала.



Основные характеристики АЦП: 1. Разрядность ( $L$ ); 2. Скорость (частота) преобразования.

# Цифро-аналоговые преобразователи (ЦАП)

Цифро-аналоговый преобразователь – электронное устройство, осуществляющее преобразование **двоичного цифрового  $L$  разрядного электрического сигнала** в **аналоговый электрический сигнал  $y(t)$** .



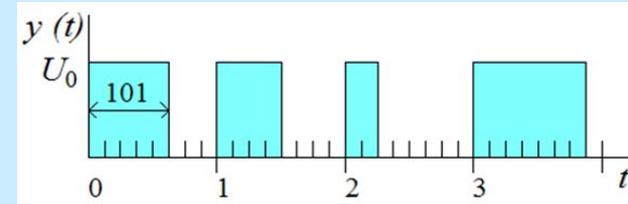
Основные характеристики АЦП:

Разрядность и частота преобразования

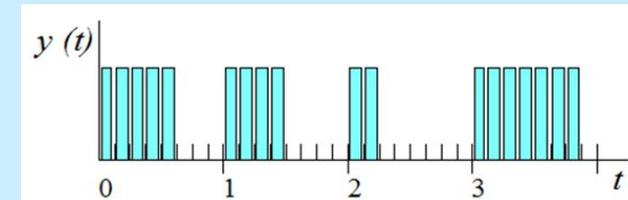
$t$	$y(t)$
0	101
1	100
2	010
3	111
$\vdots$	$\vdots$

## Типы ЦАП

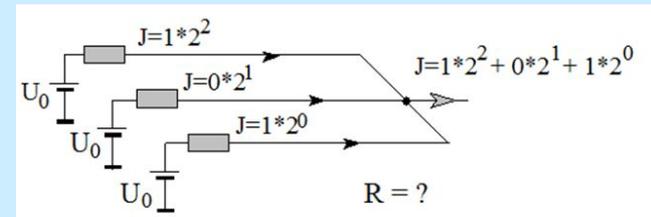
**ЦАП широтно-импульсного типа.** Внутренний источник опорного напряжения  $U_0$  подключается к выходным элементам устройства на время, пропорциональное входному двоичному цифровому сигналу.



**ЦАП передискретизации,** основан на изменяемой плотности импульсов.

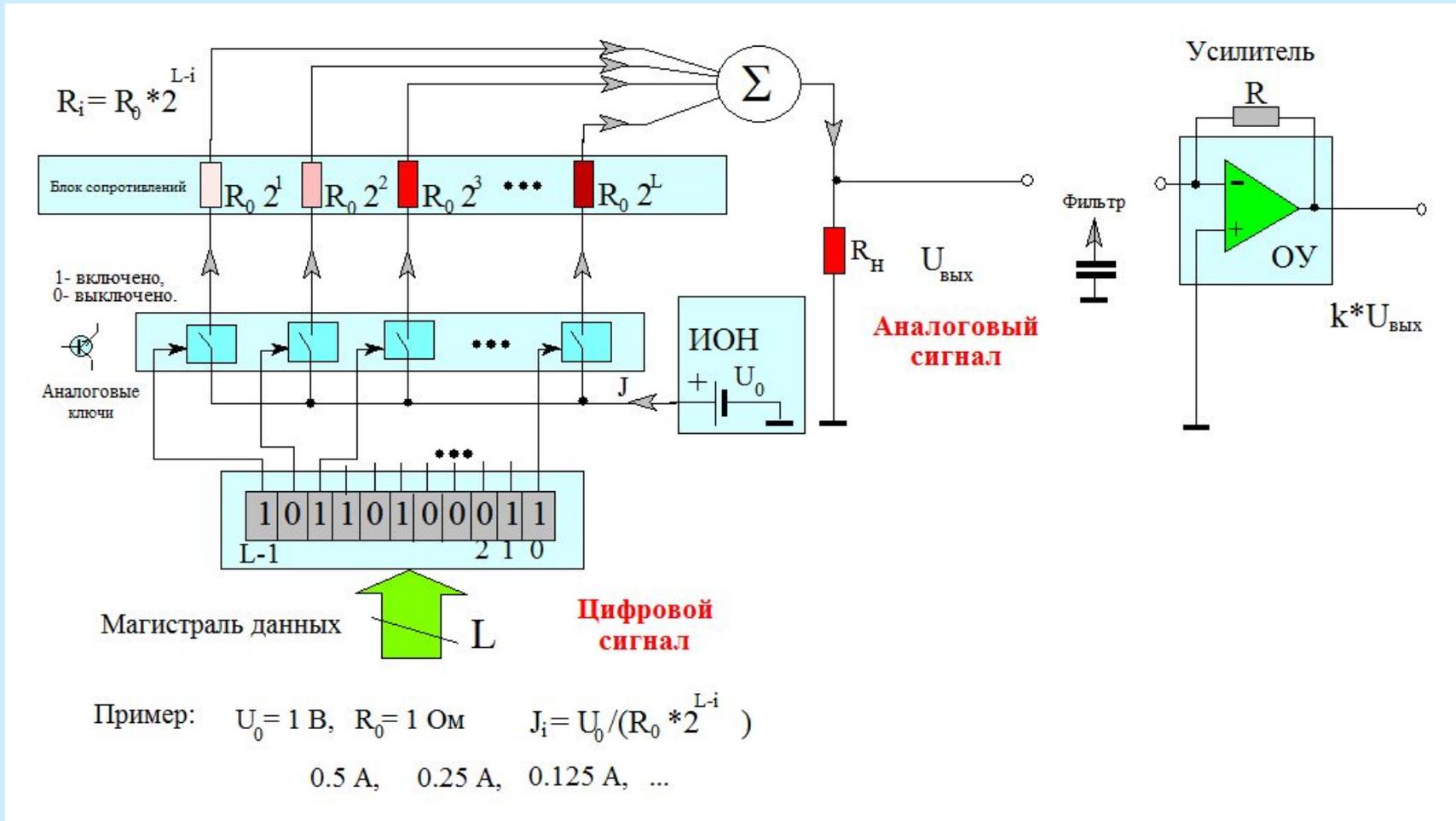


**ЦАП взвешивающего типа.** Каждая цифра двоичного сигнала в соответствии со своим весом задает ток на участке цепи. Токи всех участков суммируются и определяют выходной аналоговый сигнал.

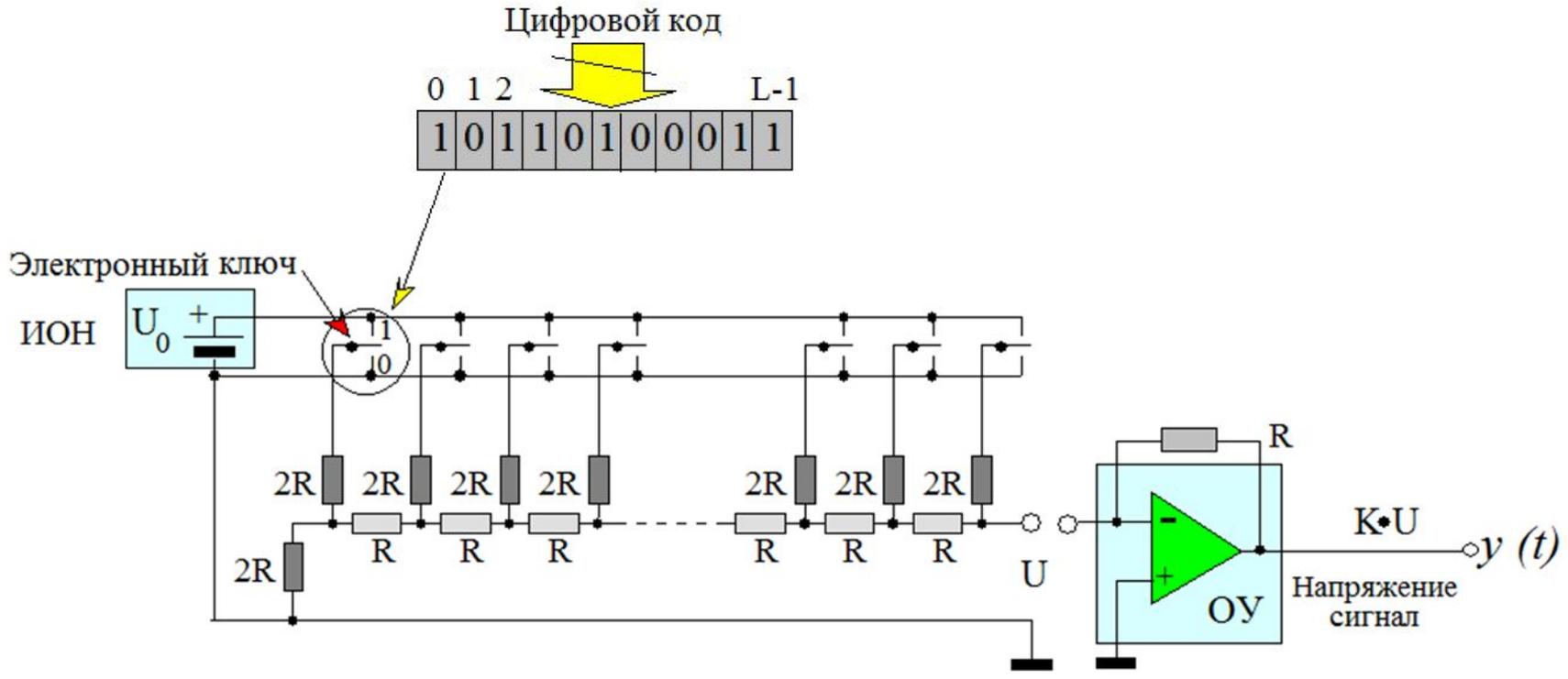


**ЦАП лестничного типа.** Суммируемые токи создаются набором резисторов подключаемых к источнику опорного напряжения по лестничной схеме (R-2R).

# ЦАП взвешивающего типа.



## ЦАП лестничного типа.



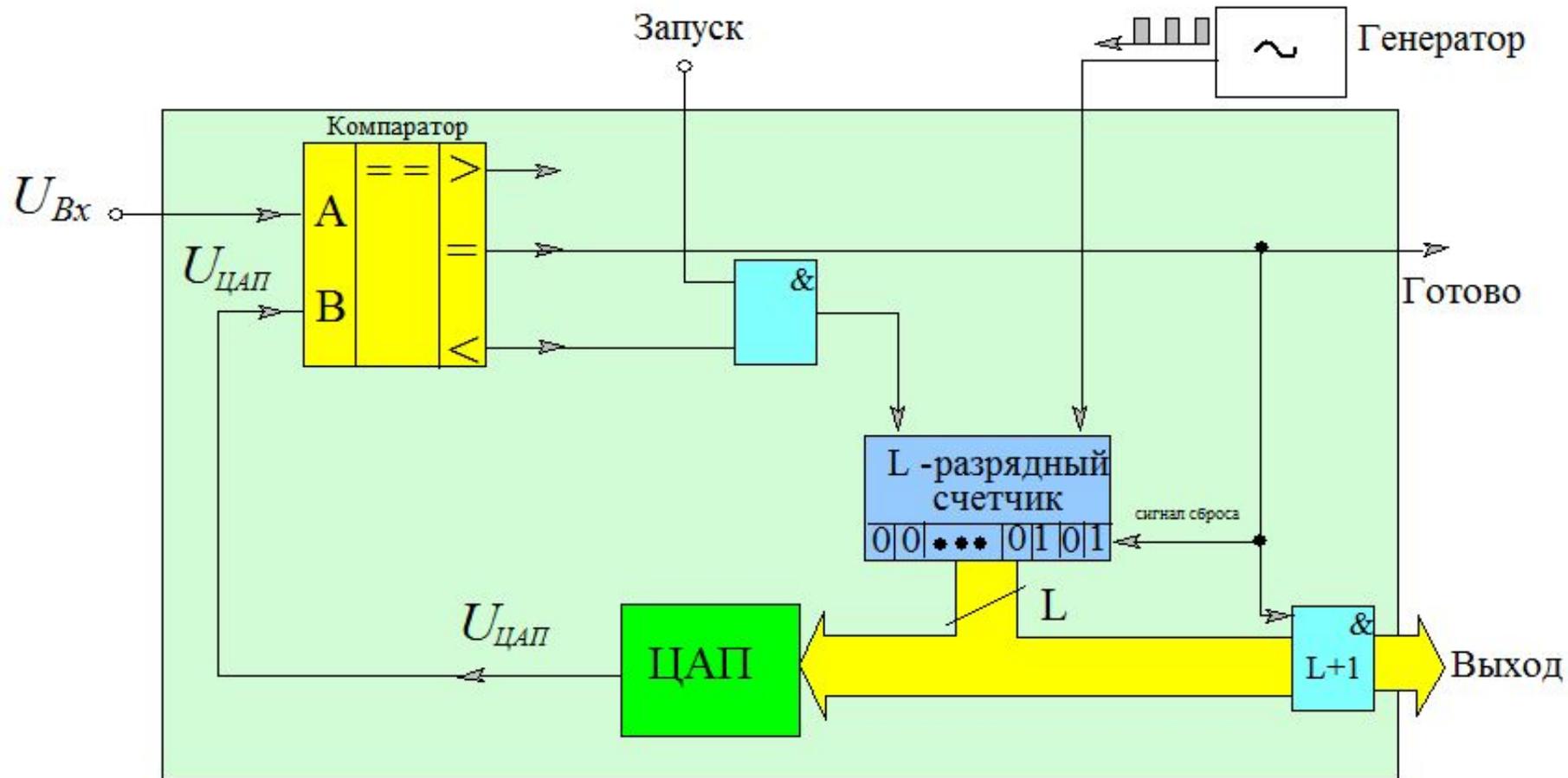
Диапазон  $0 \leq U < U_0$

Дискрет  $\Delta = \frac{U_0}{2^L}$

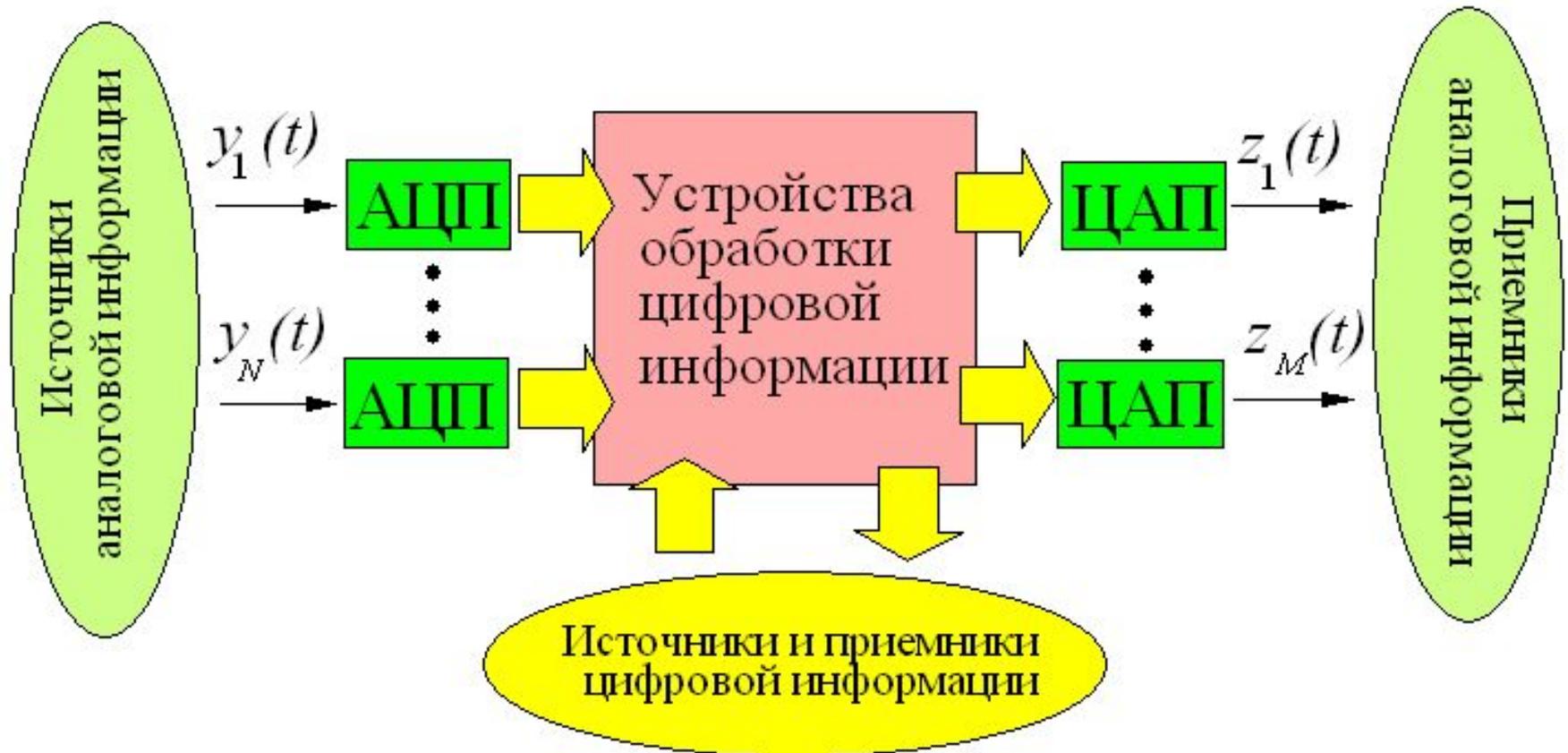
Пример:

$U_0 = 1.024 \text{ В}$   
 $0 \leq U < 1.024 \text{ В}$   
 $L = 10,$   
 $2^L = 1024$   
 $\Delta = 0.001 \text{ В}$

# Аналого-цифровой преобразователь



# Общая система сбора и обработки информации



# Датчики

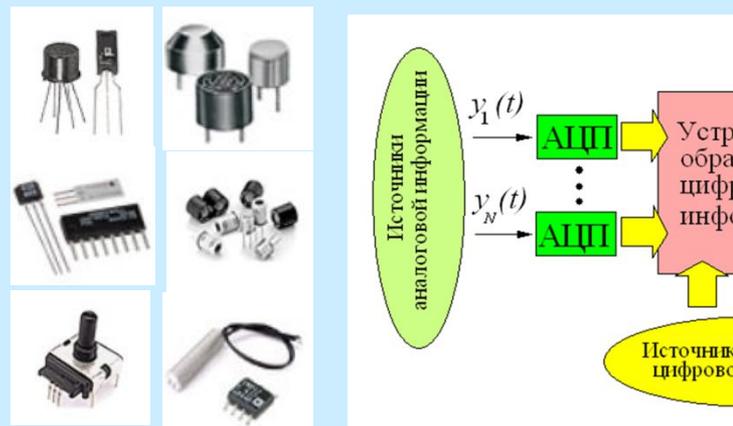
Датчик (сенсор) - чувствительный элемент, преобразующий изменения параметра среды в пригодный для технического использования сигнал (обычно электрический).

## Классификация по виду активности

- Пассивные** (параметрические). Например: фоторезистор; термистор  
(изменяется сопротивление).
- Активные** (генераторные). Например: фотодиод; термопара  
(изменяется напряжение).

## Классификация по типу внешнего воздействия и принципу преобразования

- Оптические датчики (фотодатчики)
- Магнитные датчики (На основе эффекта Холла)
- Пьезоэлектрические датчики. Тензодатчики
- Потенциометрические датчики
- Ёмкостные датчики
- Индуктивные датчики. ...



## Классификация по типу измеряемого параметра

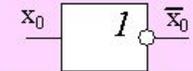
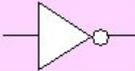
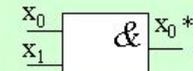
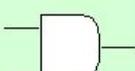
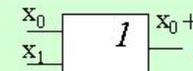
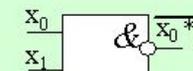
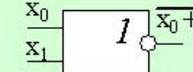
- Давления
- Уровня
- Расхода
- Температуры
- Концентрации
- Перемещения
- Положения. ...

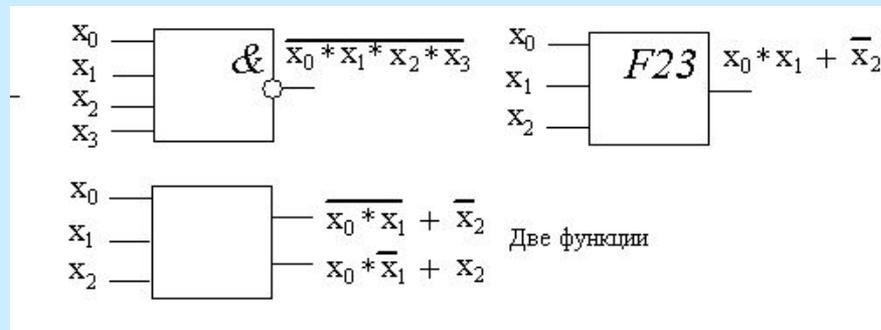
## Исполнительные устройства

# Элементарные логические функции.

Обработка информации осуществляется **цифровыми устройствами**, функционирование которых построено на двоичной логике.

Логические функции

функция	название	графическое обозначение	название элемента	графическое обозначение
$F_0(x_0) = \overline{x_0}$	<i>инверсия</i>		<i>не (NOT)</i>	
$F_1(x_0, x_1) = x_0 * x_1$	<i>конъюнкция</i>		<i>и (AND)</i>	
$F_2(x_0, x_1) = x_0 + x_1$	<i>дизъюнкция</i>		<i>или (OR)</i>	
$F_3(x_0, x_1) = \overline{x_0 * x_1}$	<i>Шеффера</i>		<i>не-и (NAND)</i>	
$F_4(x_0, x_1) = \overline{x_0 + x_1}$	<i>Пирса</i>		<i>не-или (NOR)</i>	
* * *				
$F_{15}(x_0, x_1) = \dots$				

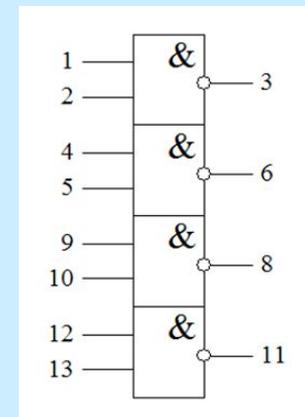
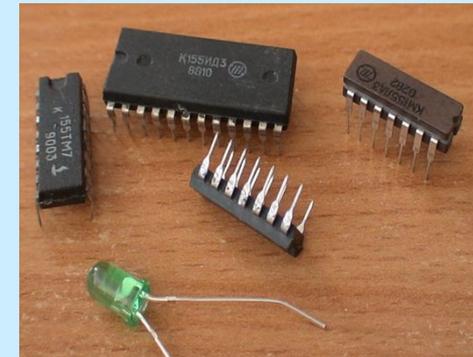


# Таблица истинности функций

$x_0$	$x_1$	$\overline{x_0}$	$x_0 * x_1$	$x_0 + x_1$	$\overline{x_0 * x_1}$	$\overline{x_0 + x_1}$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	0
1	1	0	1	1	0	0

## Базовые логические элементы

элементы	базис <i>или-не</i>	базис <i>и-не</i>



# Электрические схемы базовых логических элементов

В цифровой электронной технике параметры схемы и её техническое решение носят название "логика схемы" или просто "логика".

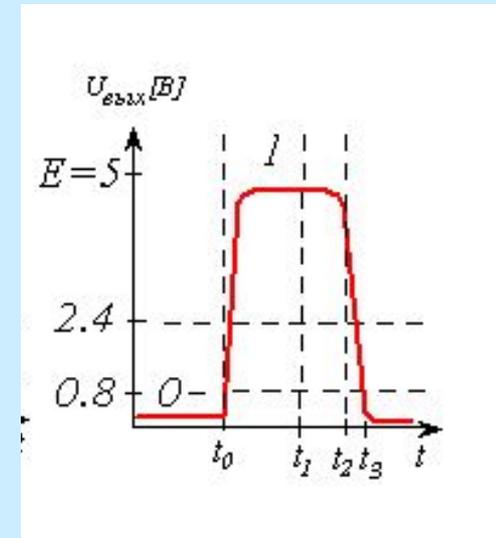
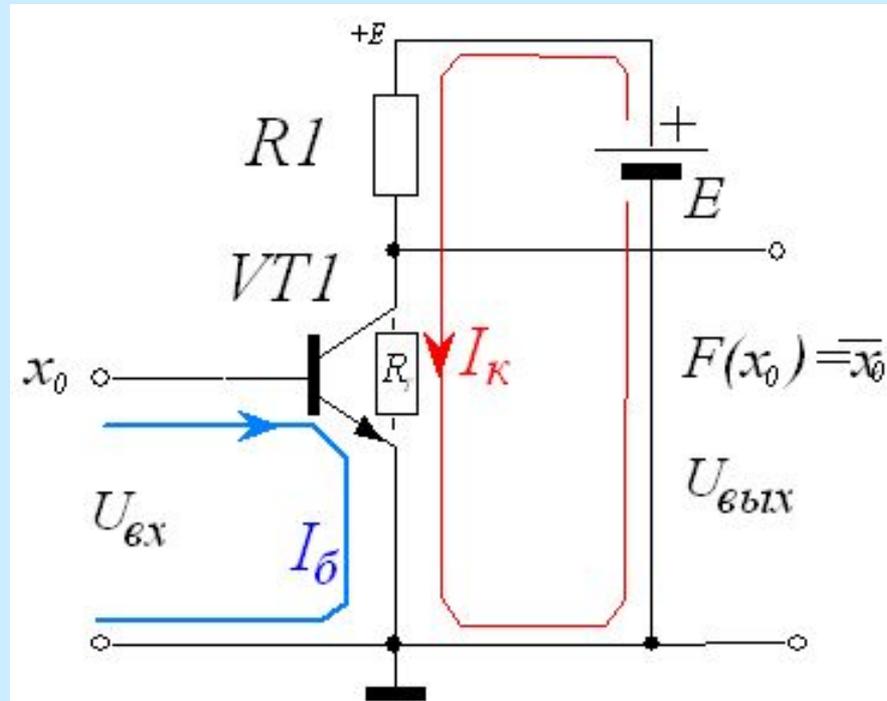
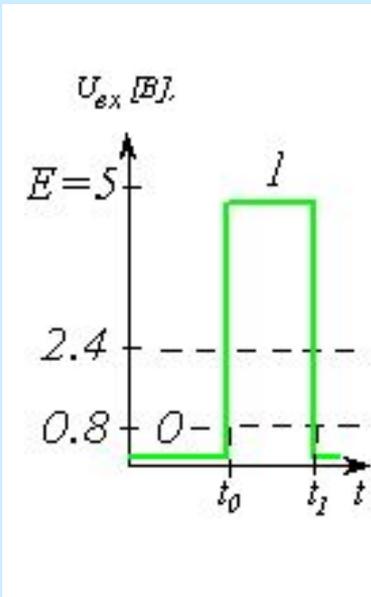
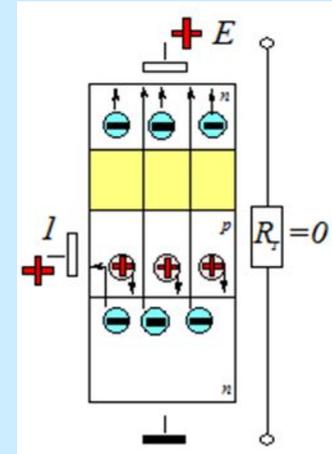
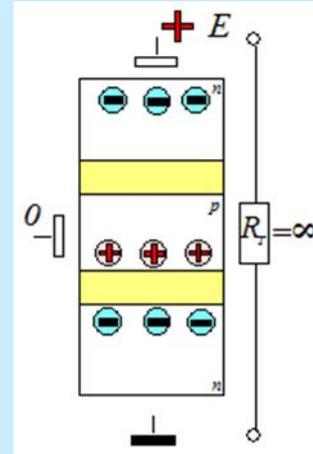
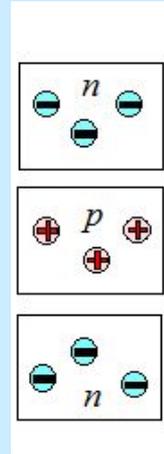
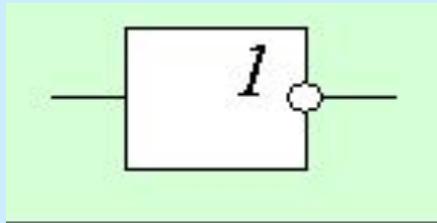
электрические уровни напряжений питания схемы;  
электрические уровни напряжений, обозначающих двоичные 0 и 1 и их допустимые пределы, обеспечивающие помехозащищенность информации;  
входные и выходные нагрузочные особенности схемы;  
параметры быстродействия схемы (задержки сигналов, крутизна фронтов);  
компонентный состав схемы (типы диодов, транзисторов).

Существуют электронные схемы на базе следующих логик:

транзисторно-транзисторной (ТТЛ);  
эмиттерно-связанной (ЭСЛ);  
р- и п-канальной металл-окисел-полупроводниковой (МОП);  
комплиментарной металл-окисел-полупроводниковой (КМОП);  
инжекционной (И2Л).

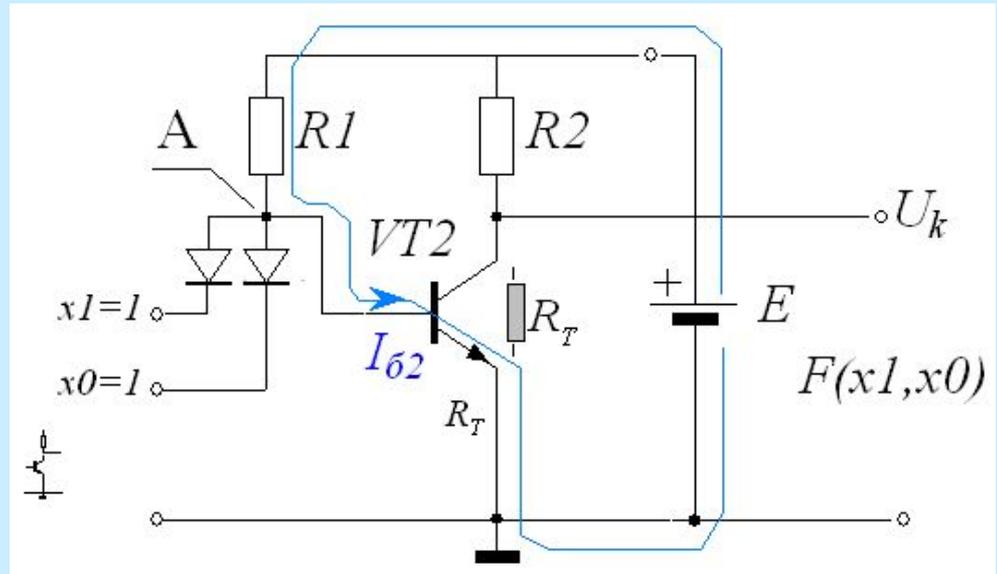
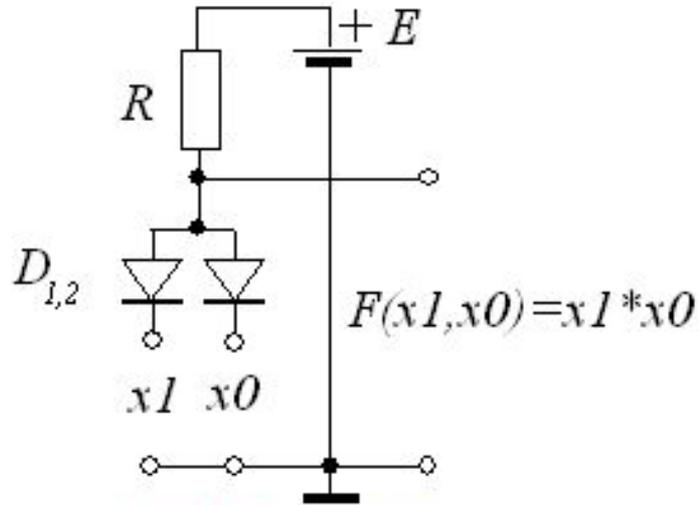
Транзисторно-транзисторная логика (ТТЛ).

Элемент НЕ.

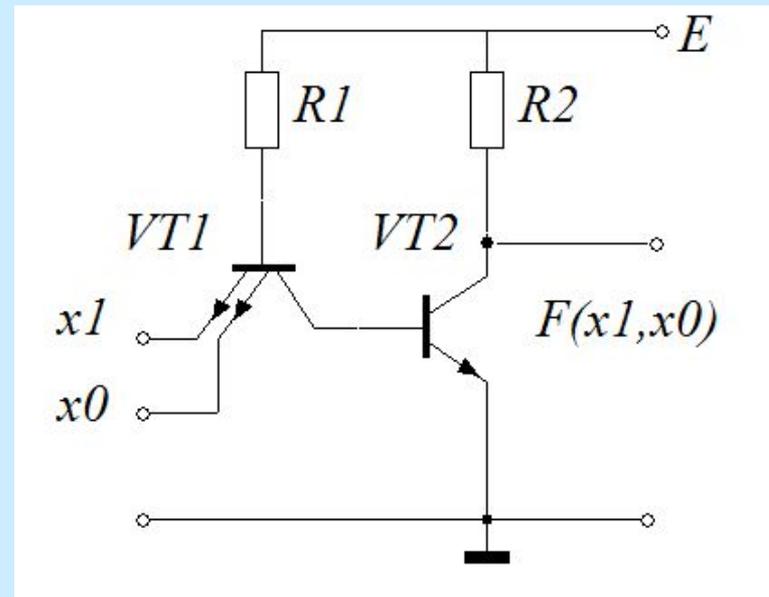


(ТТЛ).

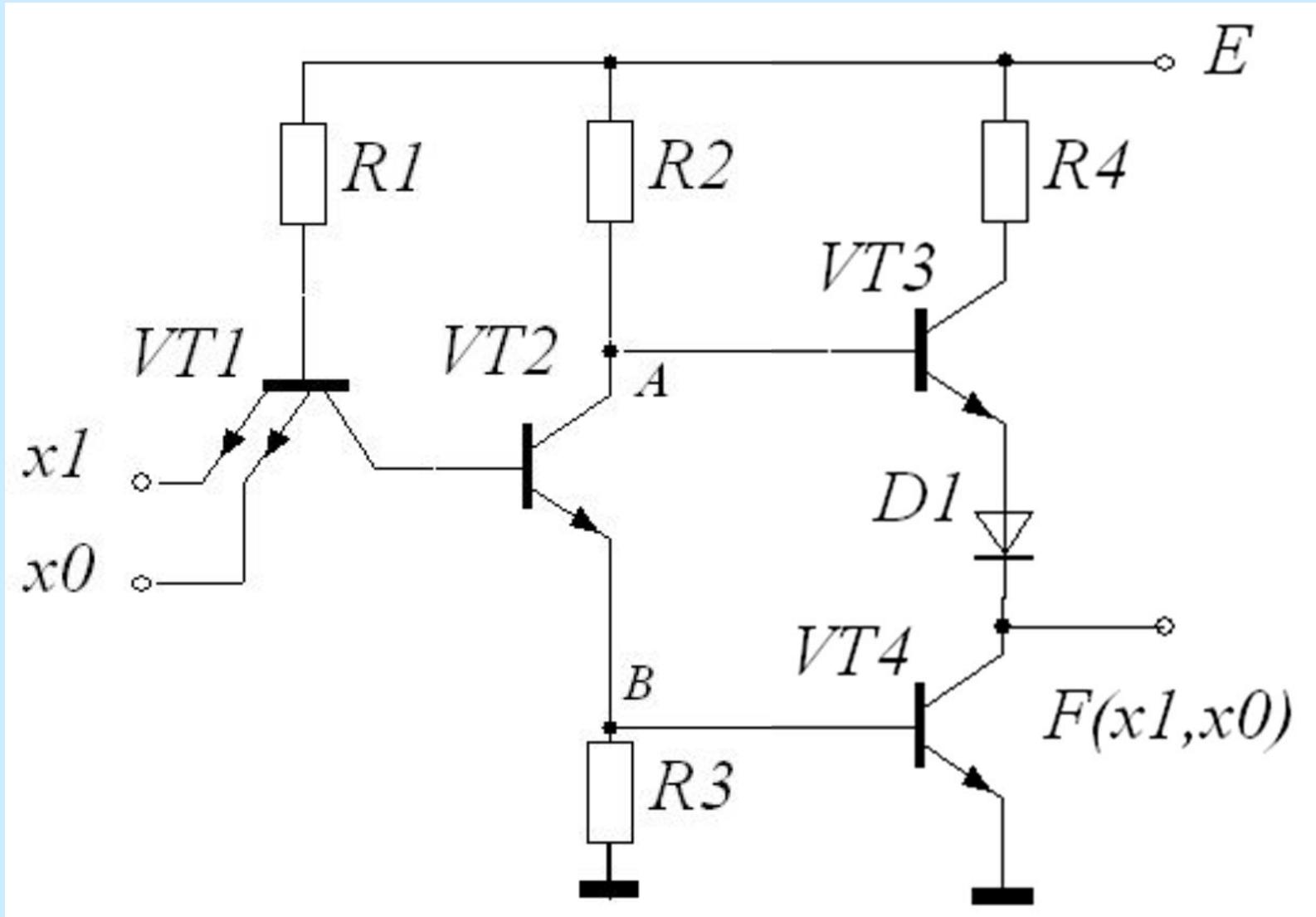
# Элемент И-НЕ.



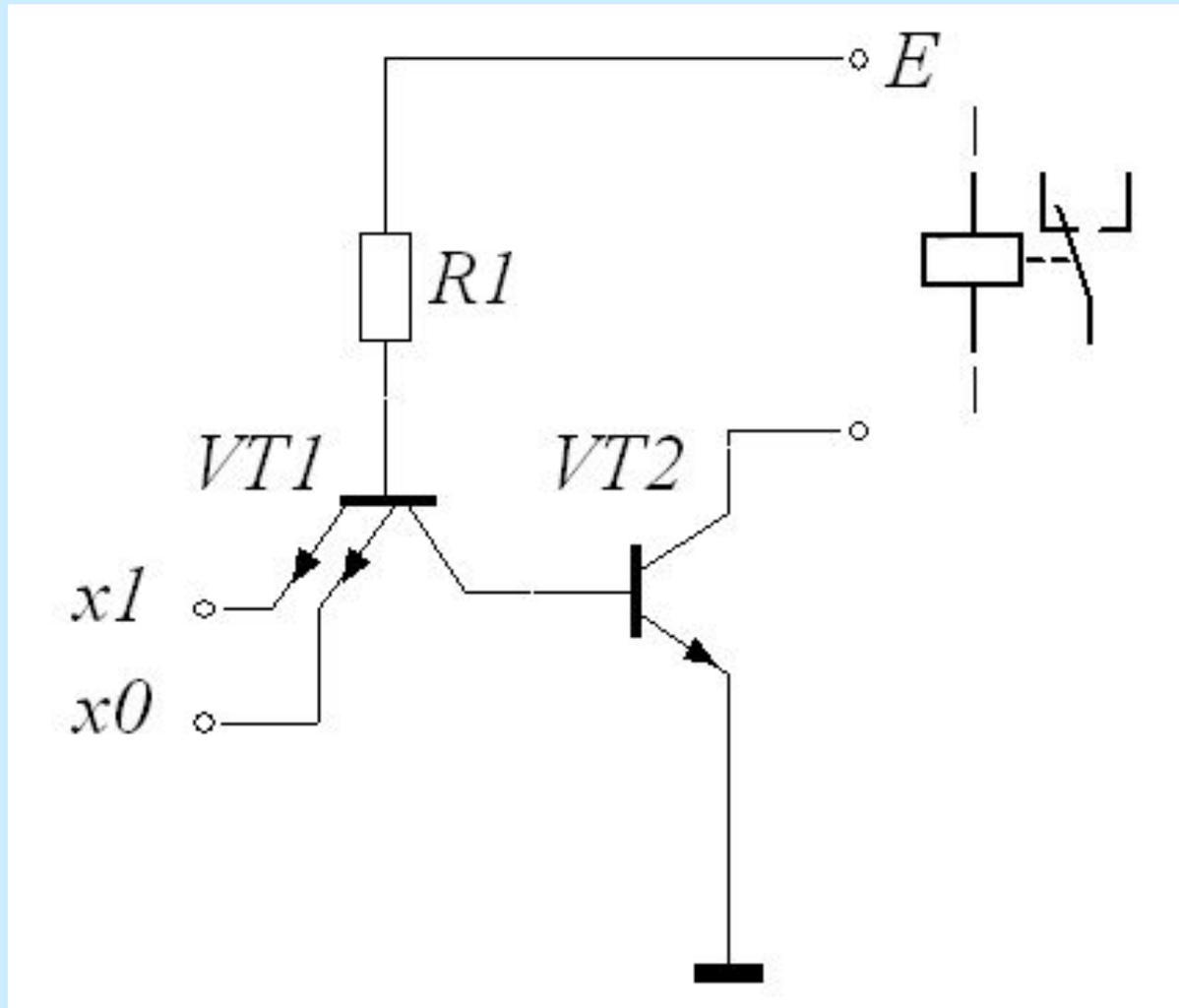
$x1$	$x0$	$F$
0	0	0
0	1	0
1	0	0
1	1	1



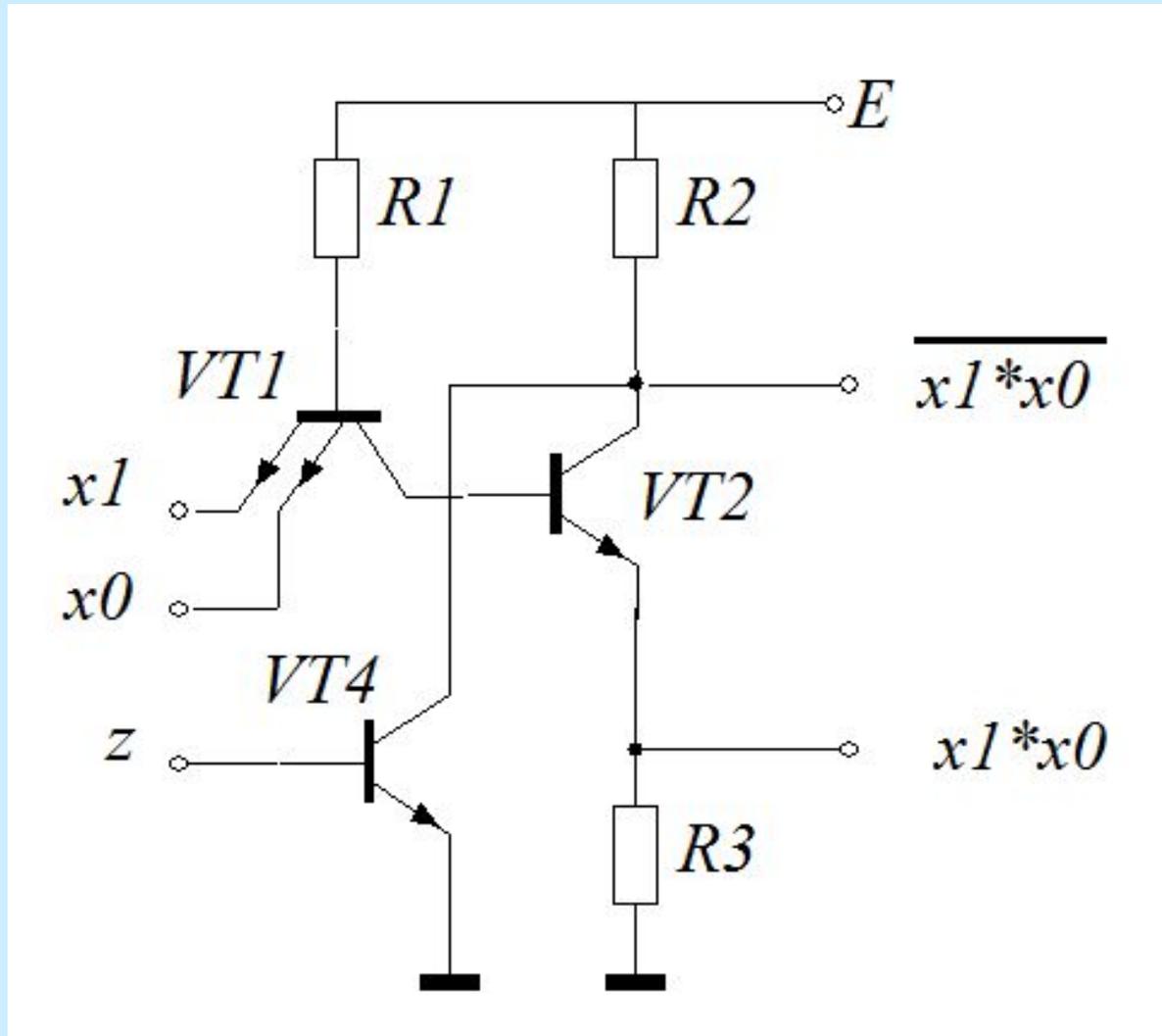
# Ограничение тока (ТТЛ).



# Открытый коллектор. (ТТЛ).

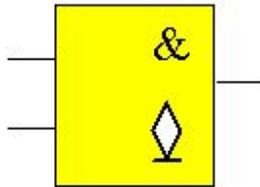


## Три стабильных состояния (ТТЛ).

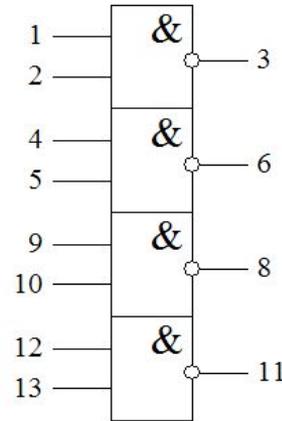
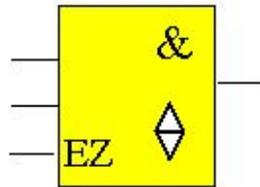


# Обозначения на схемах (ТТЛ).

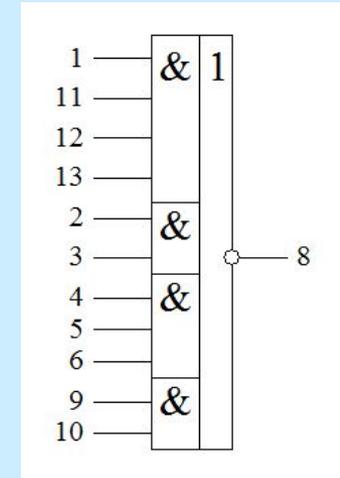
Открытый коллектор



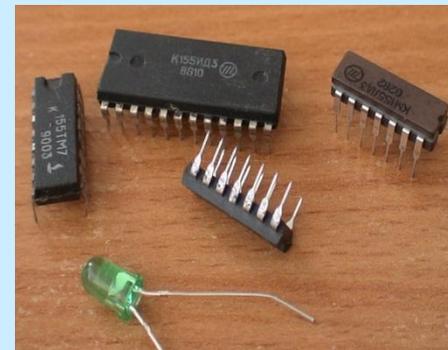
Трехстабильный выход



2\_4И-НЕ\_ЛР9



4\_2\_3\_2И\_4ИЛИ-НЕ\_ЛР9

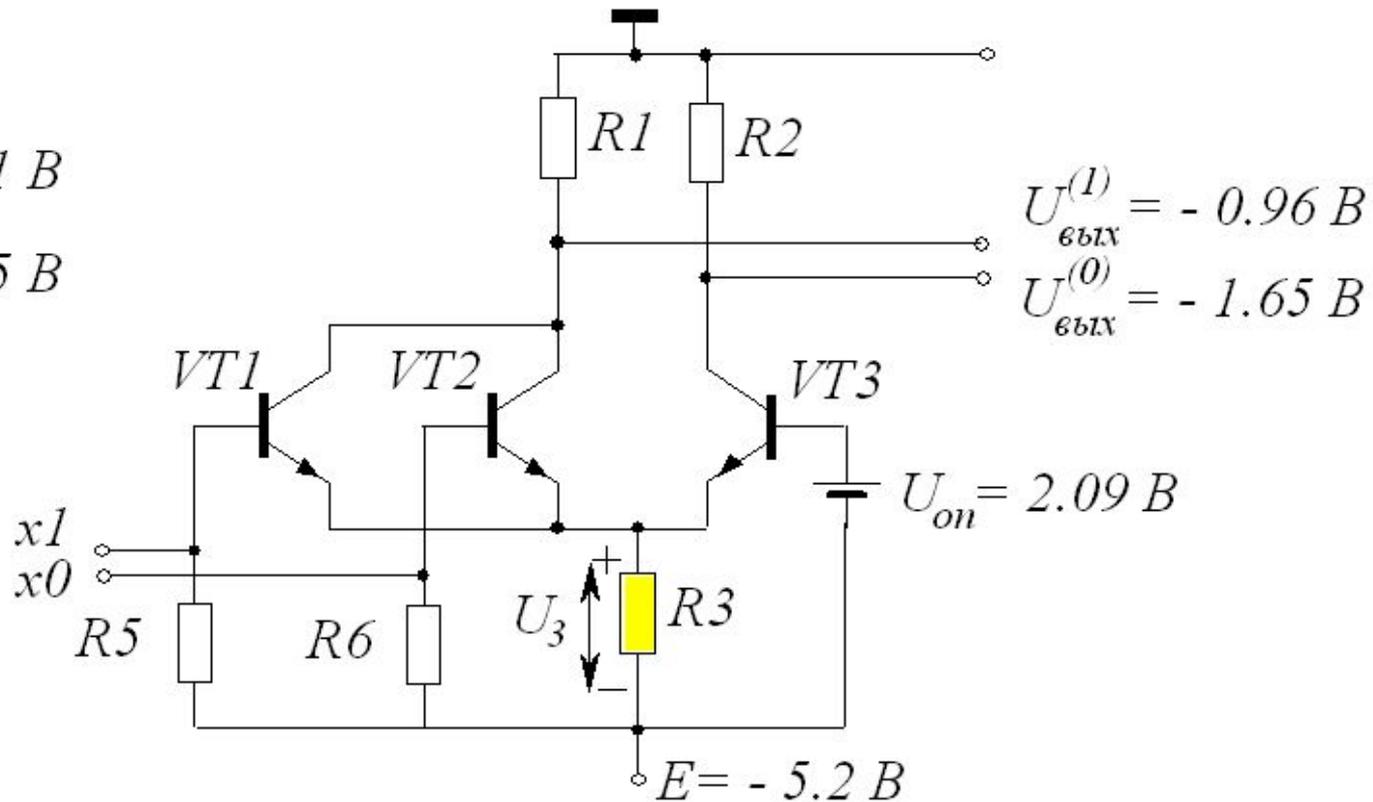


Серии: К155, К555, К531, КР1531, КР1533....

## Эмиттерно-связанная логика (ЭСЛ).

$$U_{\text{вх}}^{(1)} = -0.81 \text{ В}$$

$$U_{\text{вх}}^{(0)} = -1.85 \text{ В}$$



$$U_{\text{вых}}^{(1)} = -0.96 \text{ В}$$

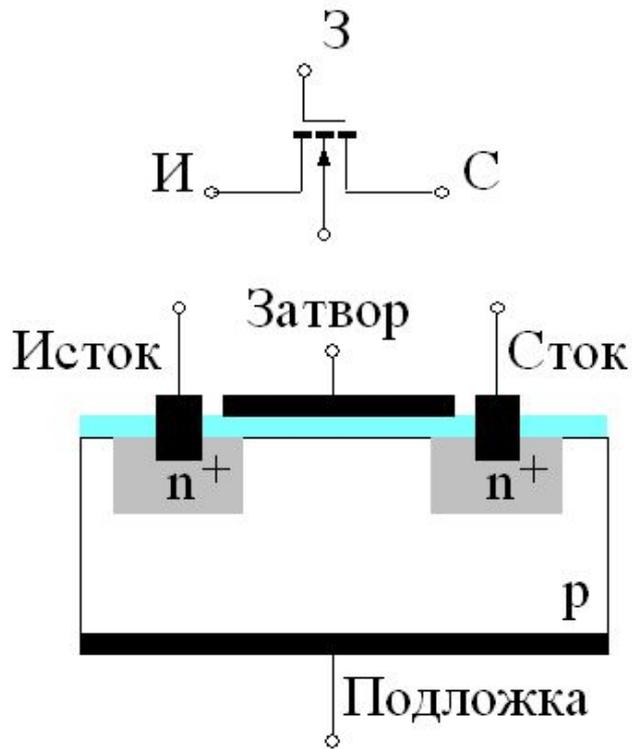
$$U_{\text{вых}}^{(0)} = -1.65 \text{ В}$$

$$U_{\text{бэ}}(VT2) = U_{x1} - U_3$$

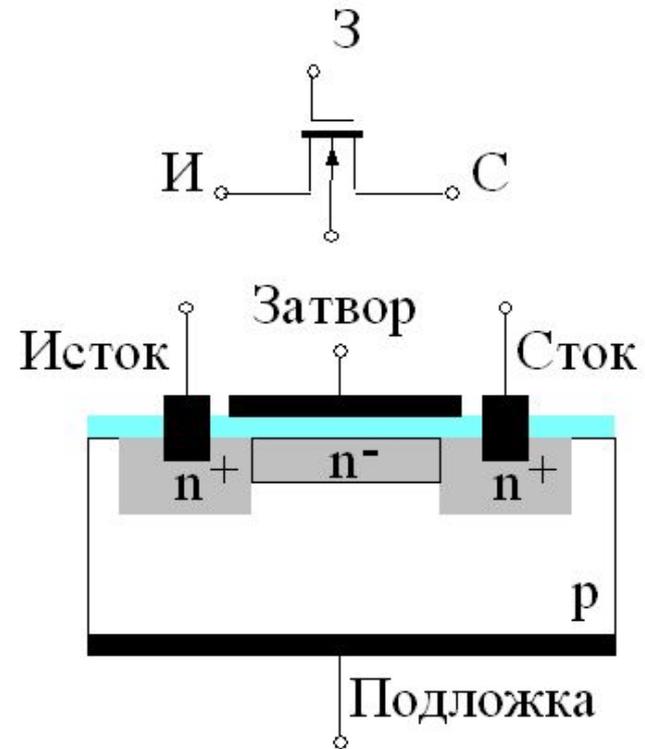
отрицательная обратная связь

# Металл-окисел-полупроводниковая (МОП) логика.

## МОП-транзисторы.

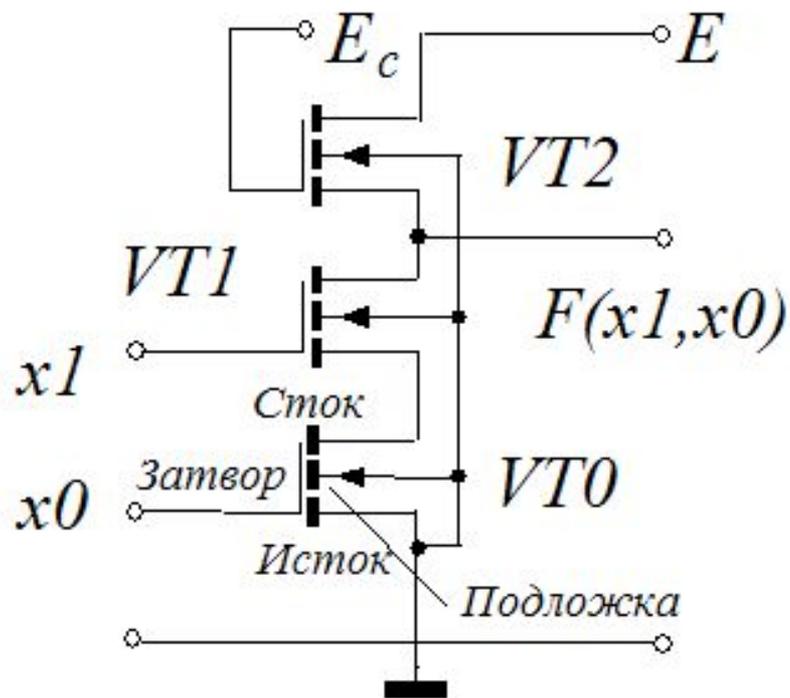


с индуцированным каналом  
(нормально закрытый)

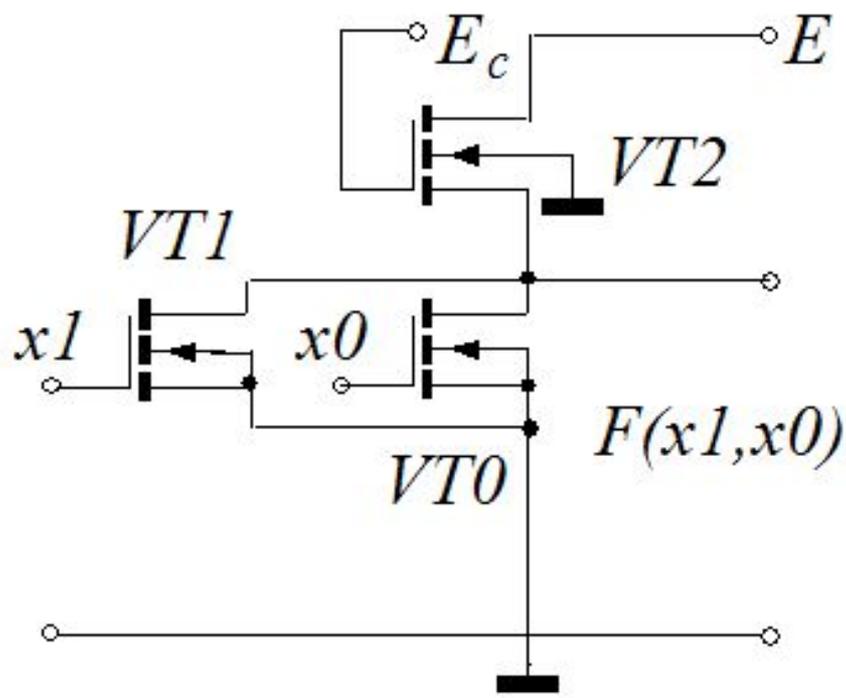


и встроенным каналом  
(нормально открытый)

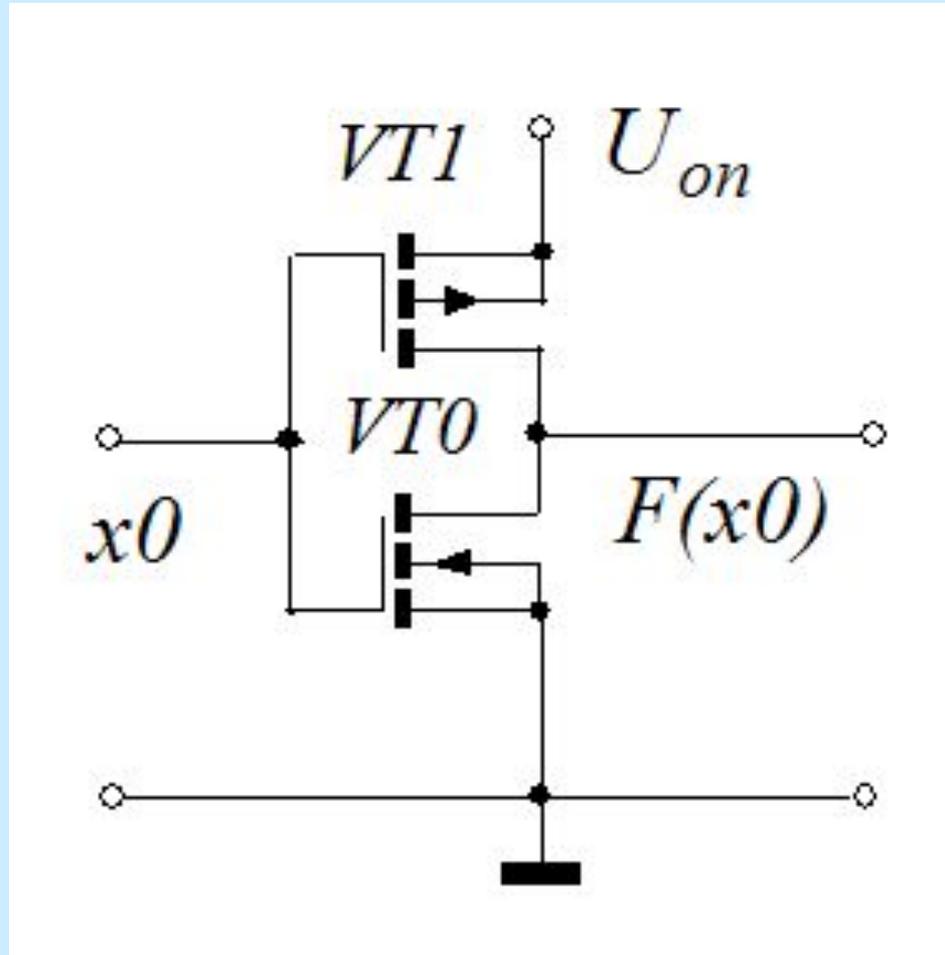
Элемент И-НЕ (МОП).



Элемент ИЛИ-НЕ (МОП).

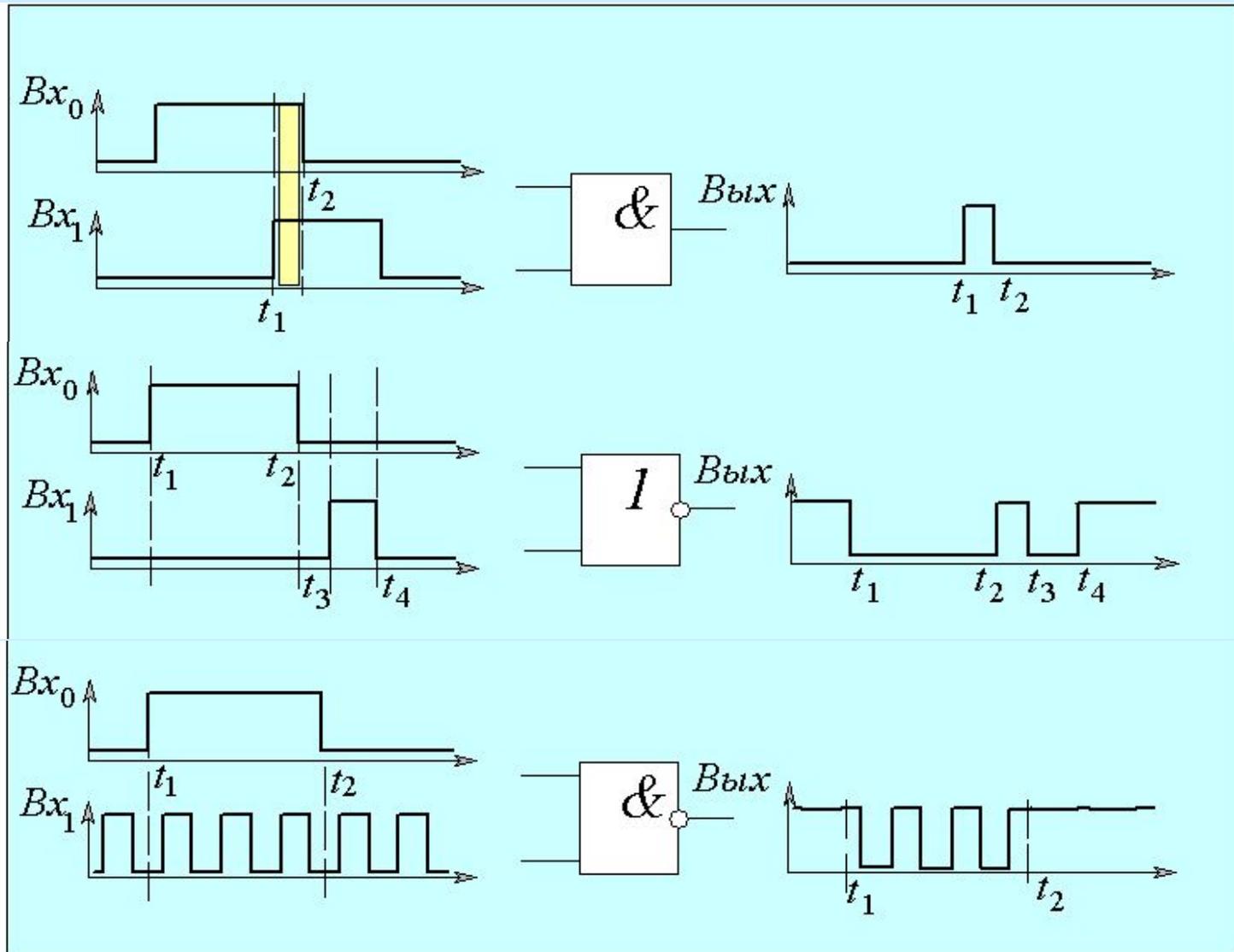


# Комплементарная МОП (КМОП) логика.

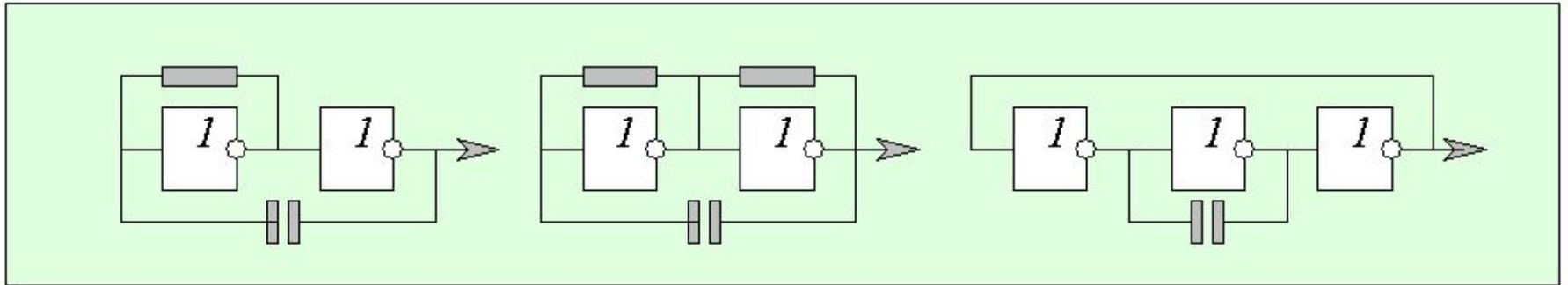


**Элемент НЕ**

# Временные диаграммы



## Генераторы на логических элементах



# Синтез цифровых устройств

Цифровые устройства

Комбинационные

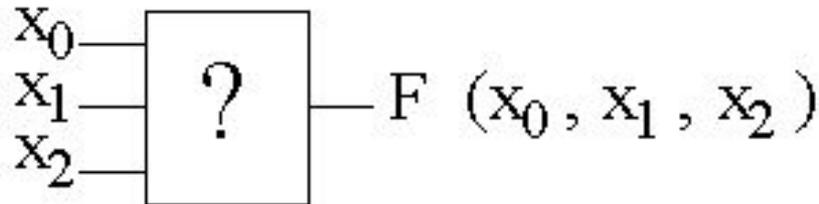
таблицы истинности  
булевы функции

Последовательностные

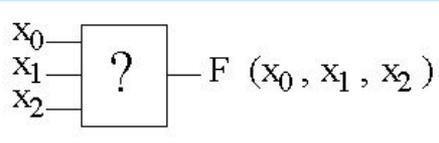
таблицы переходов  
уравнения состояния

Устройства с памятью

## Синтез комбинационных устройств

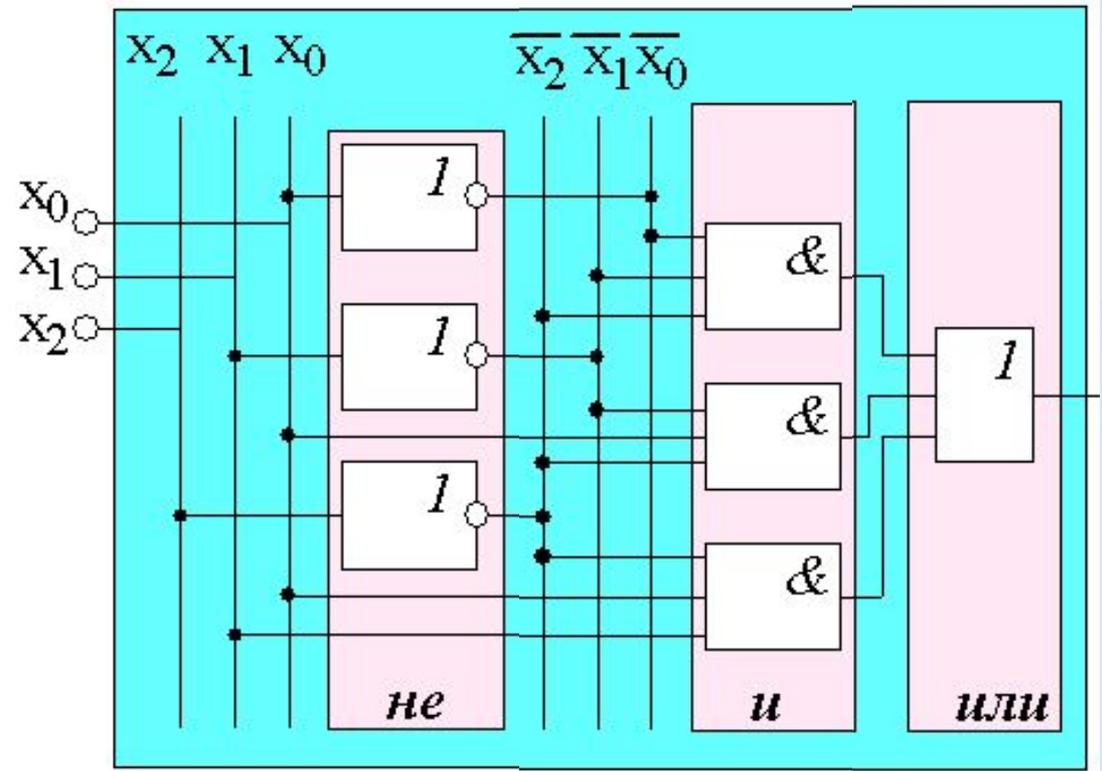


Желаемое поведение синтезируемой комбинационной логической схемы зададим таблицей истинности.



$$F(x_0, x_1, x_2) = \overline{x_2} * \overline{x_1} * \overline{x_0} + \overline{x_2} * \overline{x_1} * x_0 + \overline{x_2} * x_1 * x_0$$

$x_2$	$x_1$	$x_0$	F	Conjunction
0	0	0	1	$\overline{x_2} * \overline{x_1} * \overline{x_0}$
0	0	1	1	$\overline{x_2} * \overline{x_1} * x_0$
0	1	0	0	0
0	1	1	1	$\overline{x_2} * x_1 * x_0$
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0



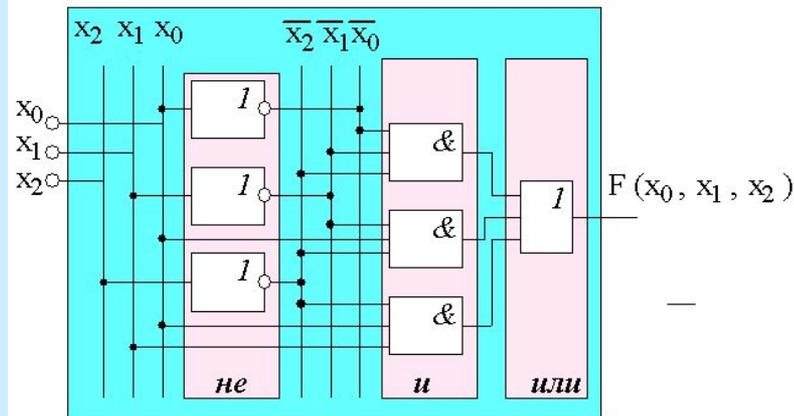
# Синтез цифровых устройств

## Минимизации выражений

$$F(x_0, x_1, x_2) = \bar{x}_2 * \bar{x}_1 * \bar{x}_0 + \bar{x}_2 * \bar{x}_1 * x_0 + \bar{x}_2 * x_1 * x_0$$

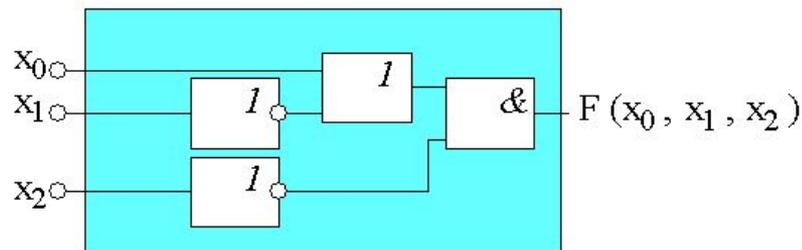
Некоторые правила и теоремы

$x+1=1$	$x*0=0$	Идемпотентность
$x+x=x$	$x*x=x$	
$x+\bar{x}=1$	$x*\bar{x}=0$	
$\bar{\bar{x}}=x$	$\bar{\bar{x}}=x$	Двойное отрицание
$x+y=y+x$	$x*y=y*x$	Коммутативность
$x+x*y=x$	$x*(x+y)=x$	Поглощение
$x+\bar{x}*y=x+y$	$x*(\bar{x}+y)=x*y$	Склеивание
$\overline{x+y} = \bar{x} * \bar{y}$	$\overline{x*y} = \bar{x} + \bar{y}$	Закон Де Моргана
$x+y*z = (x+y)*(x+z)$	$x*(y+z) = x*y+x*z$	Дистрибутивность



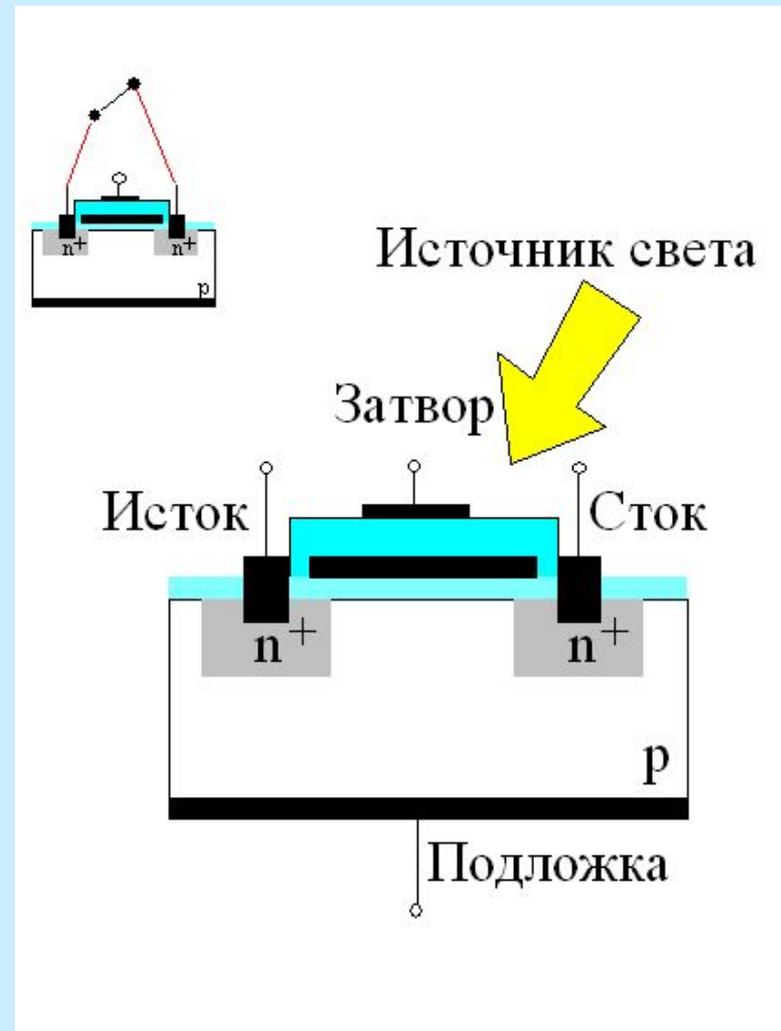
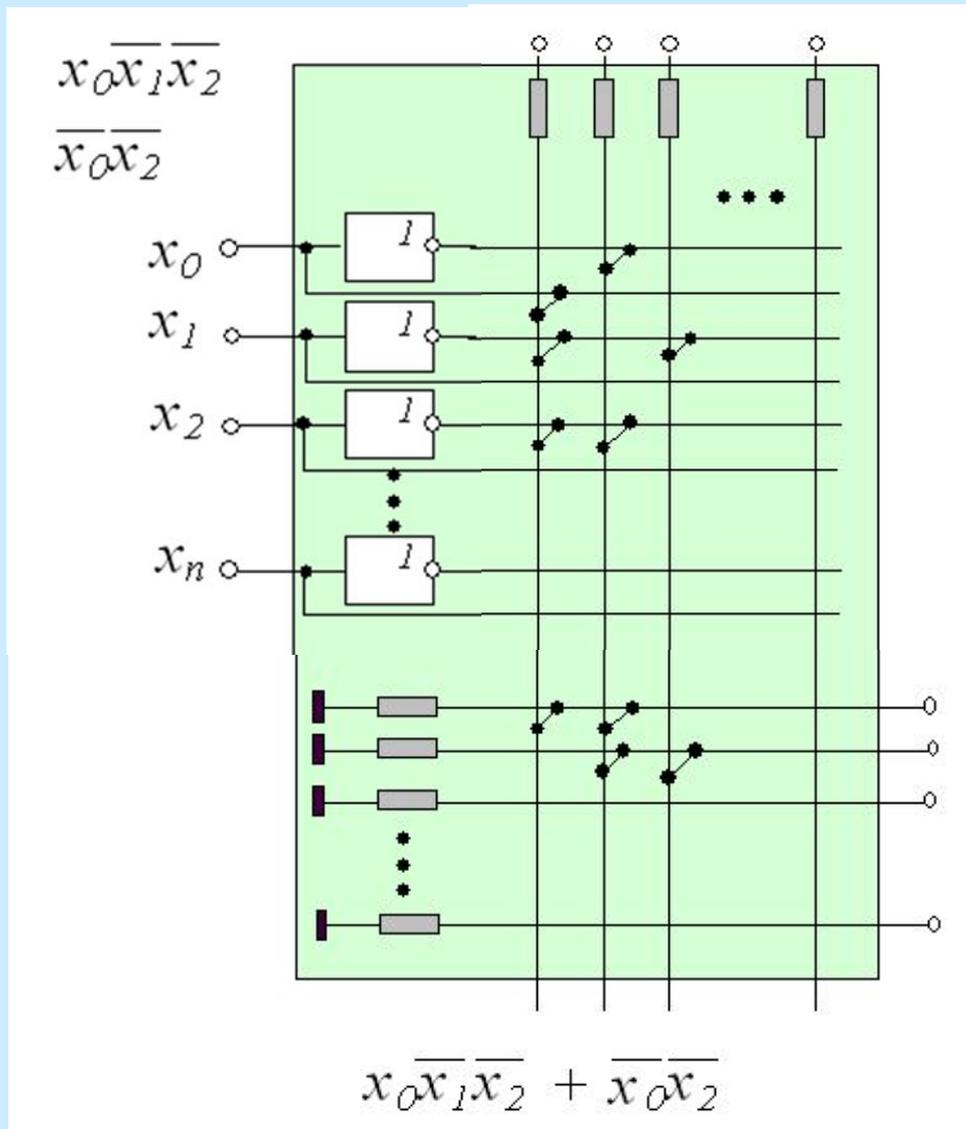
Минимизация схемы логического устройства

$$F(x_0, x_1, x_2) = \bar{x}_2 * (\bar{x}_1 + x_0)$$



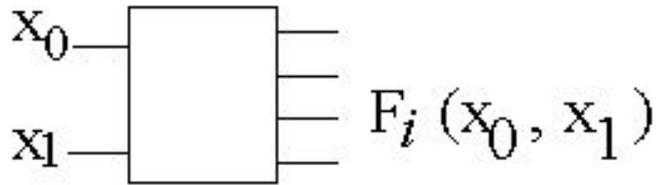
# Логическая матрица

ПЛИС, *programmable logic device*, PLD

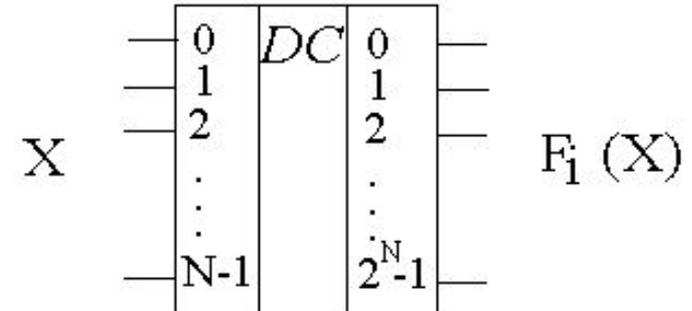


# Цифровые устройства комбинационного типа

*Дешифратор* (декодер) – устройство, преобразующее  $N$ - разрядное двоичное число в  $M$ - разрядное двоичное число.

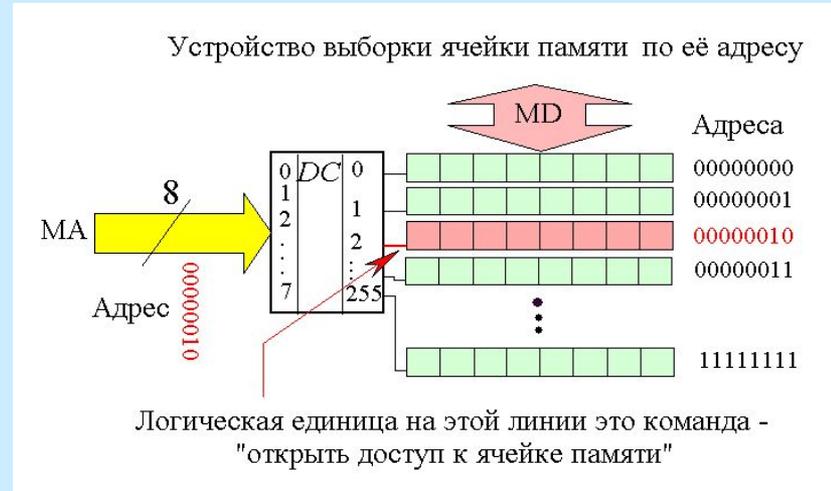
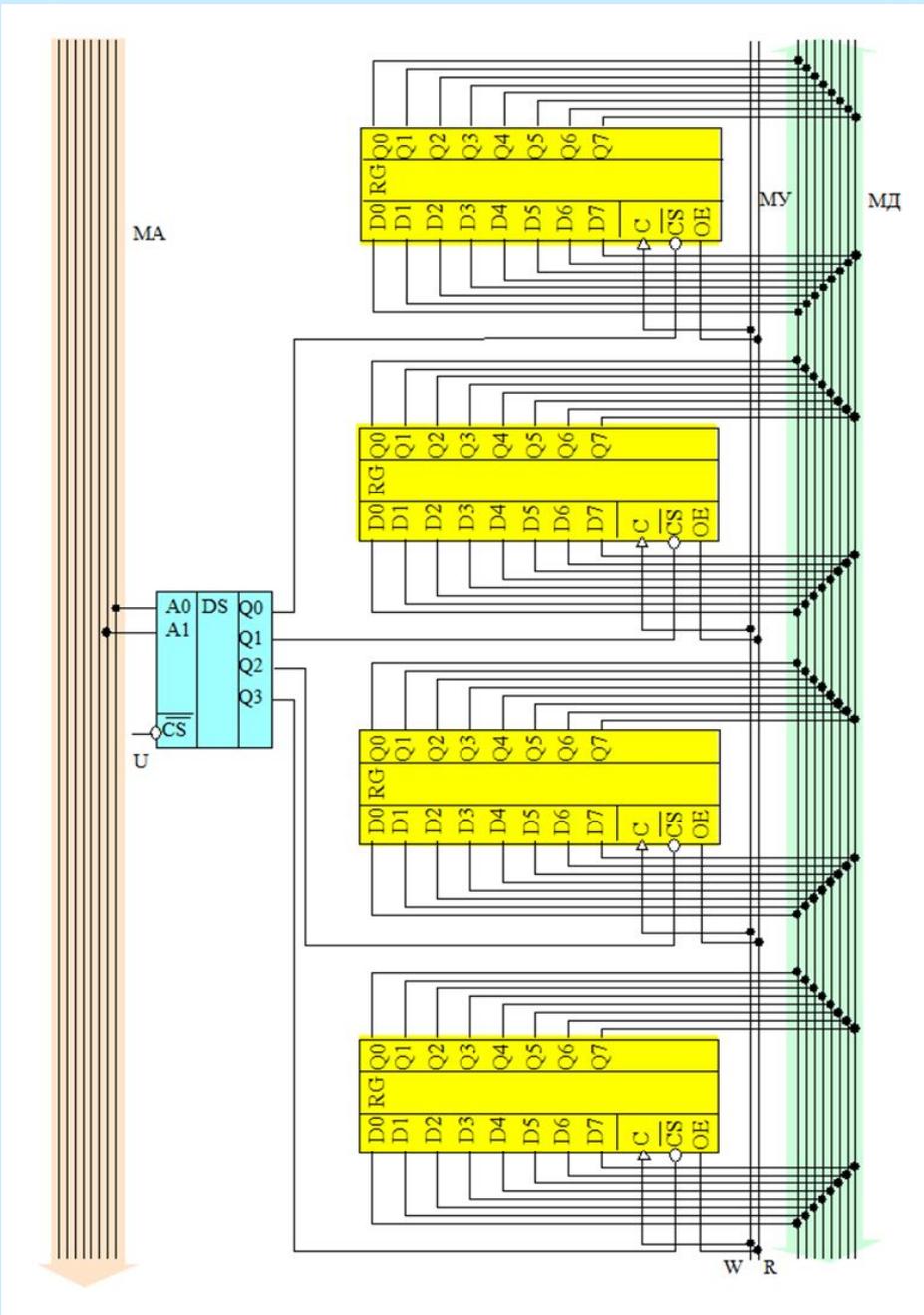


$x_1$	$x_0$	$F_0$	$F_1$	$F_2$	$F_3$
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



# Адресный доступ к памяти.

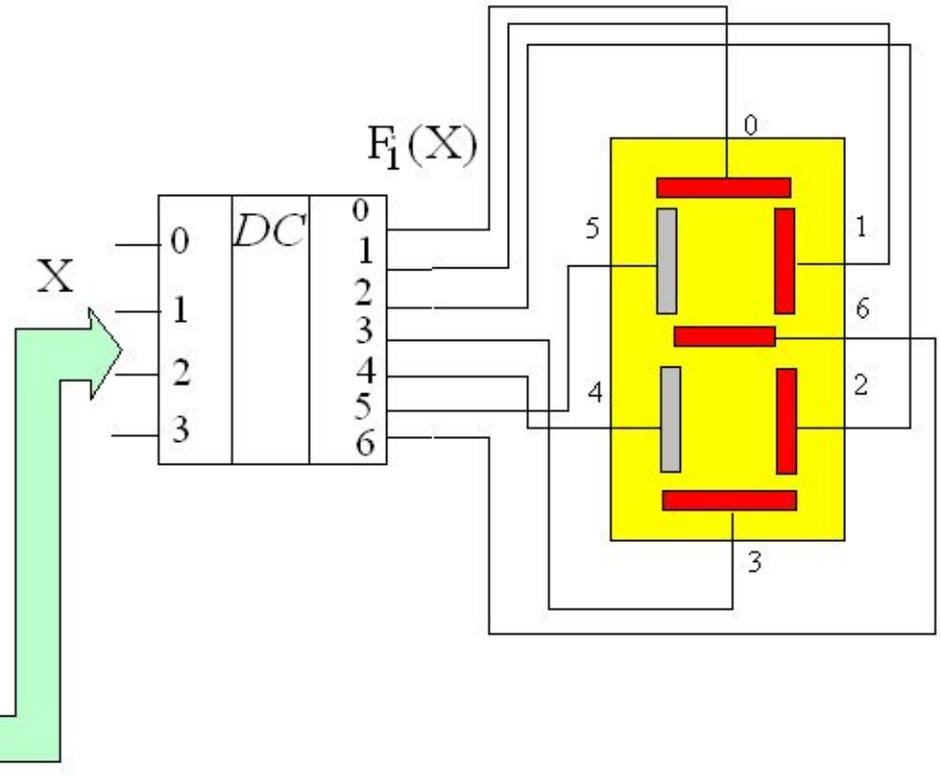
# Дешифратор адреса ячейки памяти



Дешифратор для управления матрицами светодиодных излучателей.

Дешифратор для индикатора

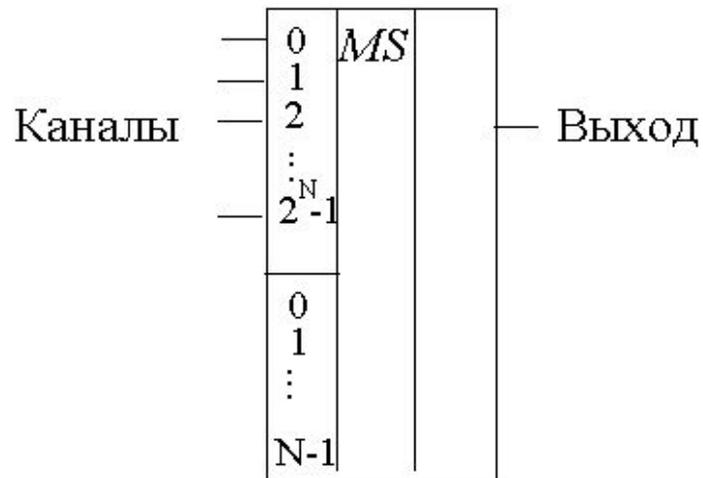
X <sub>3</sub>	X <sub>2</sub>	X <sub>1</sub>	X <sub>0</sub>		F <sub>0</sub>	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>	F <sub>4</sub>	F <sub>5</sub>	F <sub>6</sub>
0	0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	0	1	1	0	0	0	0
0	0	1	0	2	1	1	0	1	1	0	1
0	0	1	1	3	1	1	1	1	0	0	1
0	1	0	0	4	0	1	1	0	0	1	1
0	1	0	1	5	1	0	1	1	0	1	1
0	1	1	0	6	1	0	1	1	1	1	1
0	1	1	1	7	1	1	1	0	0	0	0
1	0	0	0	8	1	1	1	1	1	1	1
1	0	0	1	9	1	1	1	1	0	1	1



# Мультиплексор

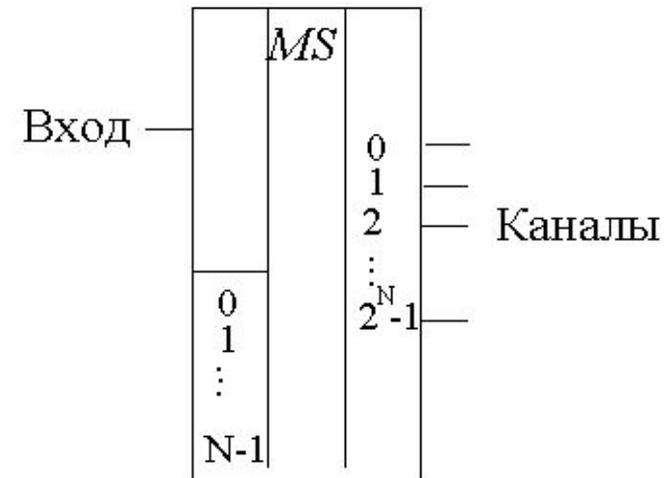
Мультиплексор (переключатель каналов) – устройство, обеспечивающее соединение одного из входных каналов с выходным каналом.

Мультиплексор



N- разрядный код управления

Демультимплексор



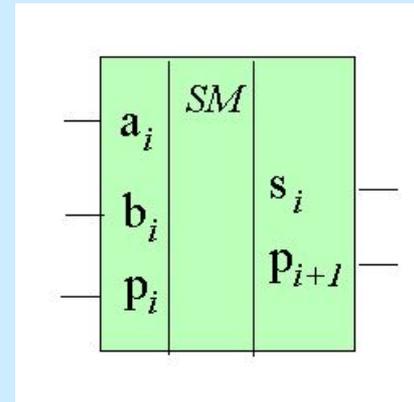
N- разрядный код управления

# Двоичный сумматор

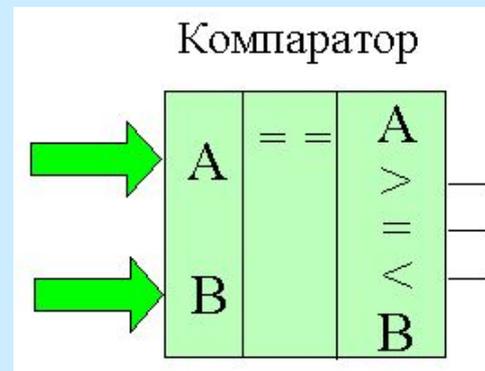
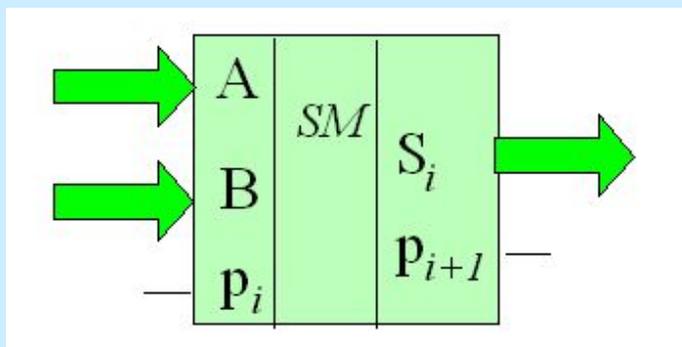
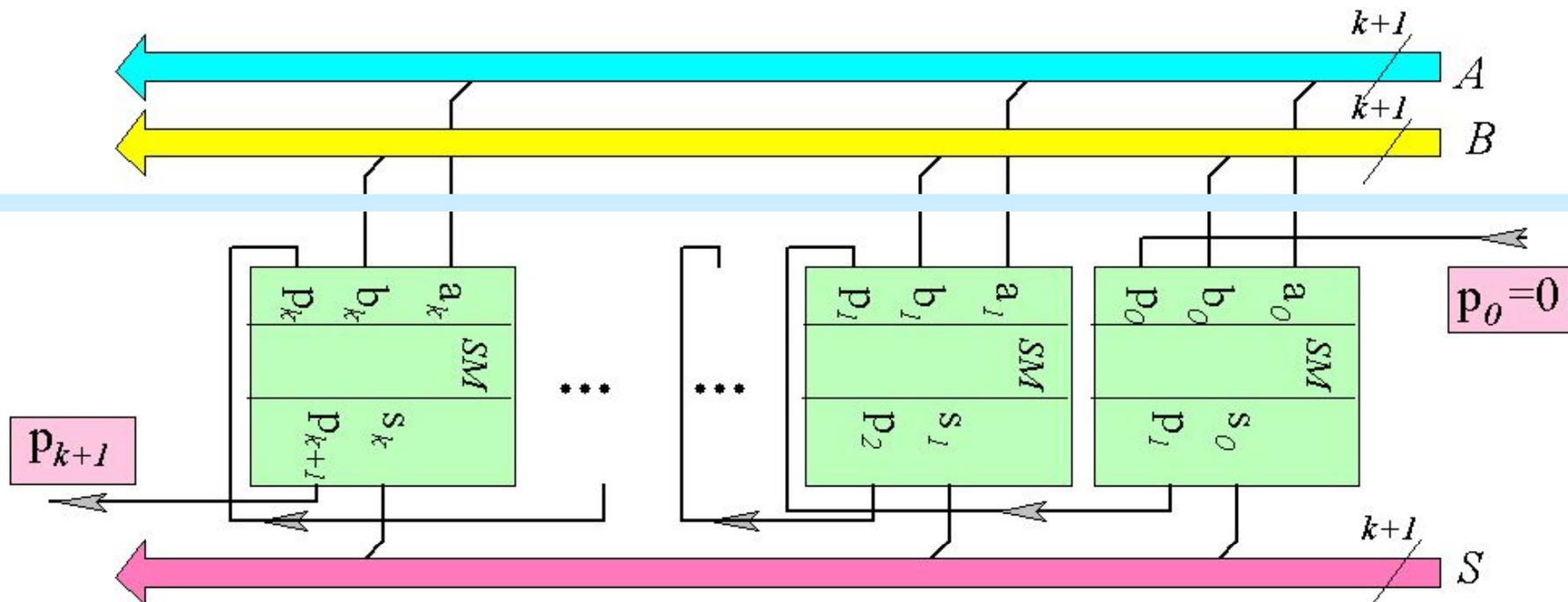
Двоичный сумматор – устройство, осуществляющее сложение двоичных чисел  $A = \{a_k, \dots, a_0\}$  и  $B = \{b_k, \dots, b_0\}$  и получение их суммы  $S = \{s_k, \dots, s_0\}$ .

$$\begin{array}{r}
 a_k a_{k-1} \dots a_1 a_0 \\
 + \quad b_k b_{k-1} \dots b_1 b_0 \\
 \hline
 s_k s_{k-1} \dots s_1 s_0
 \end{array}
 \begin{array}{l}
 A \\
 B \\
 S
 \end{array}$$

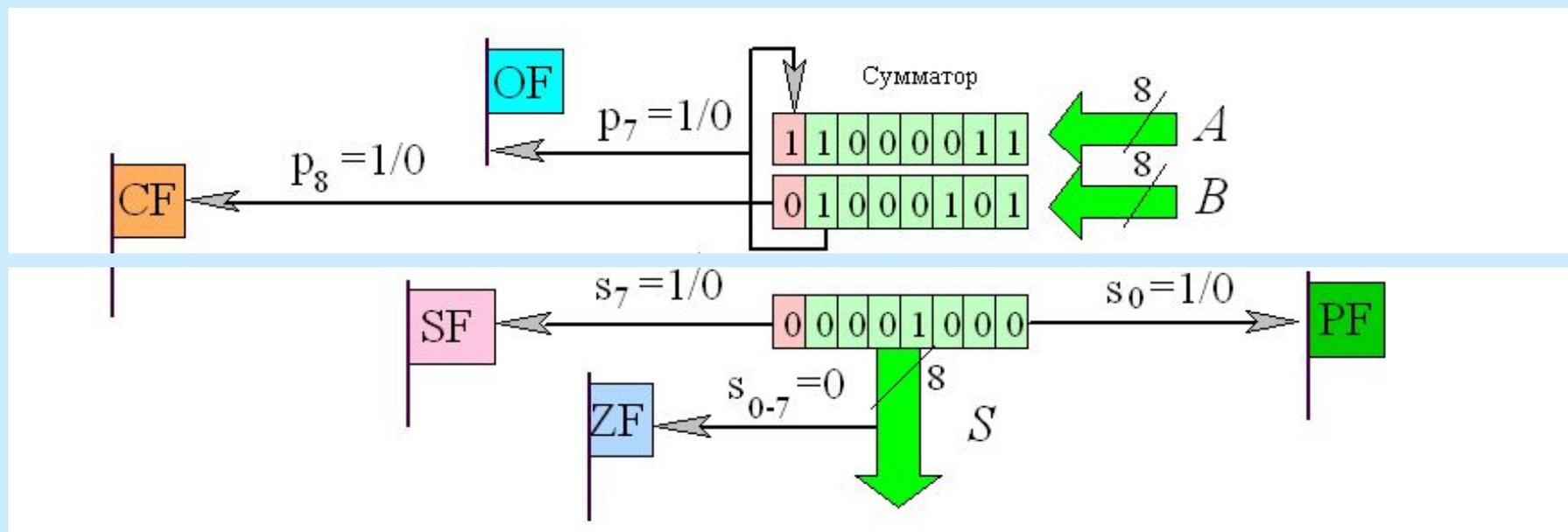
$$p_{i+1} \begin{cases} + 1 \\ + 0 \end{cases}$$



# N – разрядный двоичный сумматор



# Сумматор как элемент АЛУ



## Состояния (флаги) АЛУ

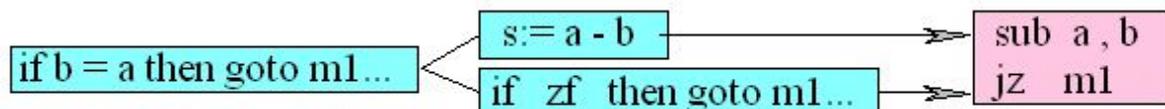
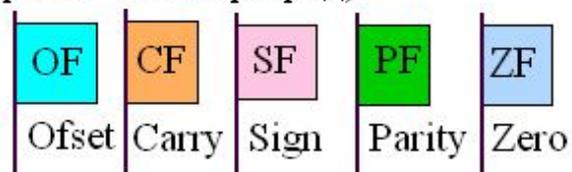
O - переполнение (перенос 1 в зн. разряд)

C - перенос

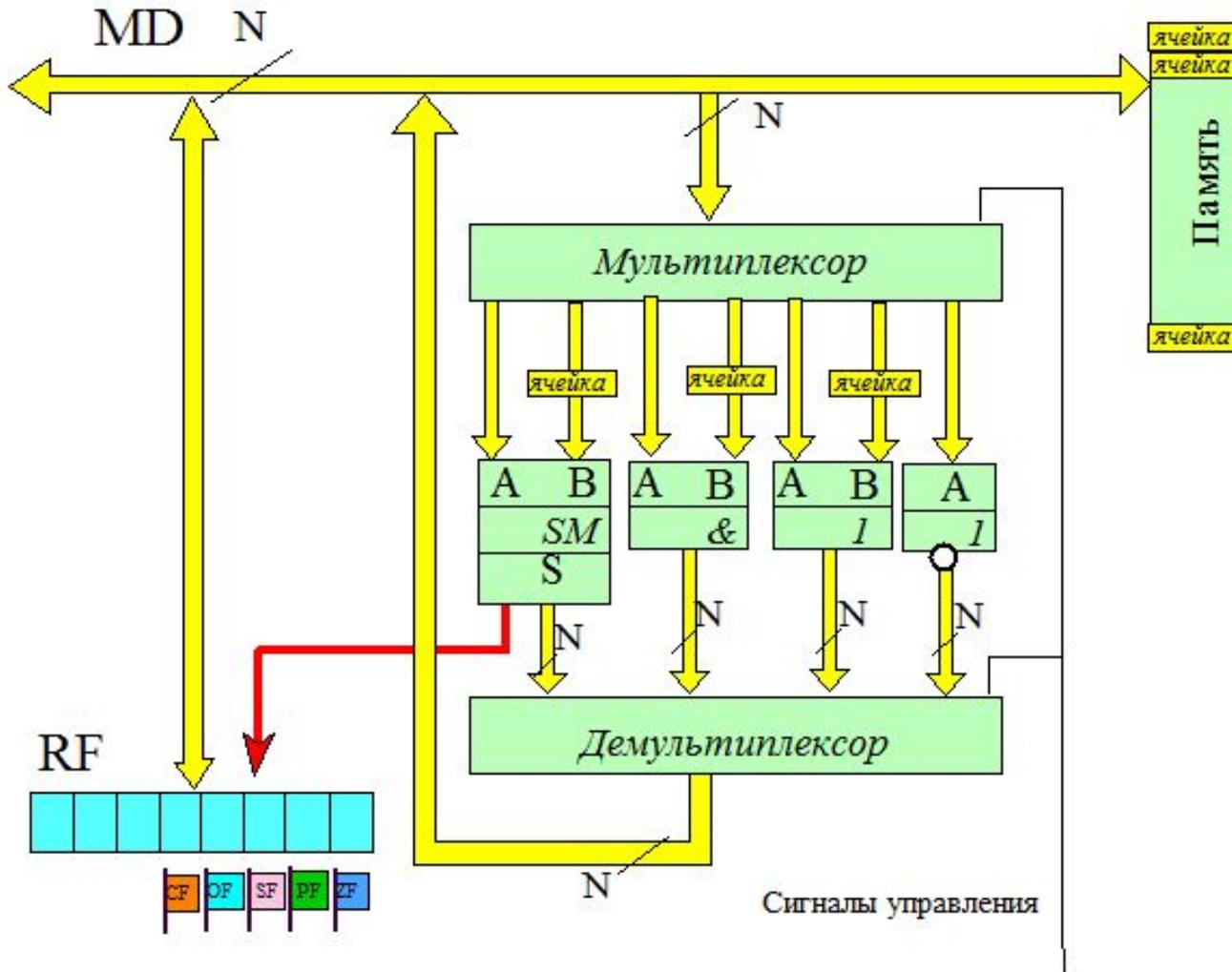
S - знак

P - четность

Z - результат равен 0



Фрагмент АЛУ



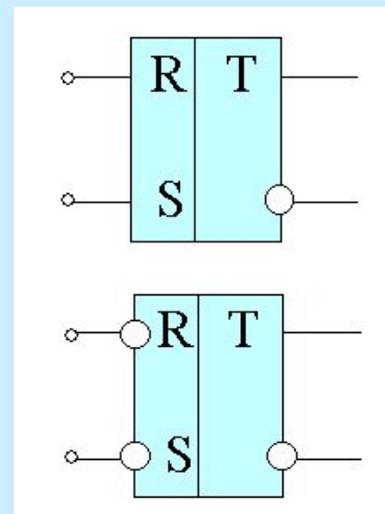
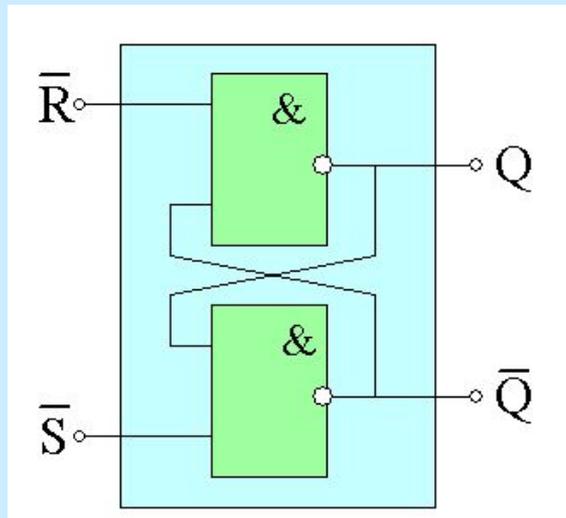
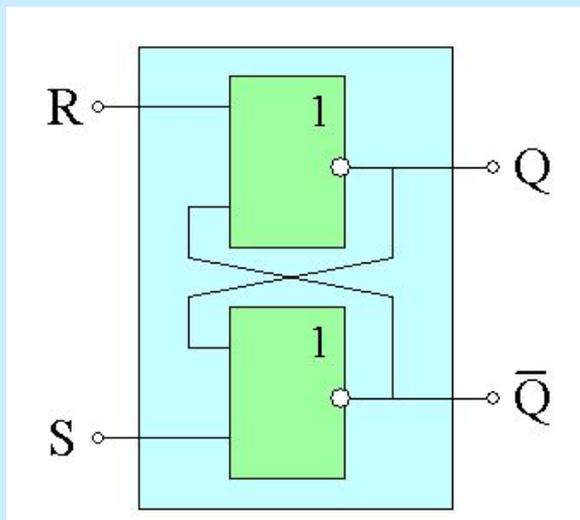
```

add a , b
sub a , b
inc a
dec a
and a , b
or a , b
not a
    
```

# Цифровые устройства последовательного типа

## Триггер

Триггер – электронное устройство, способное находится в одном из двух устойчивых состояний



Q (Quit) - выход триггера (прямой).

$\bar{Q}$  - выход триггера (инверсный).

R (Reset) - вход установки триггера в состояние 0.

S (Set) - вход установки триггера в состояние 1.

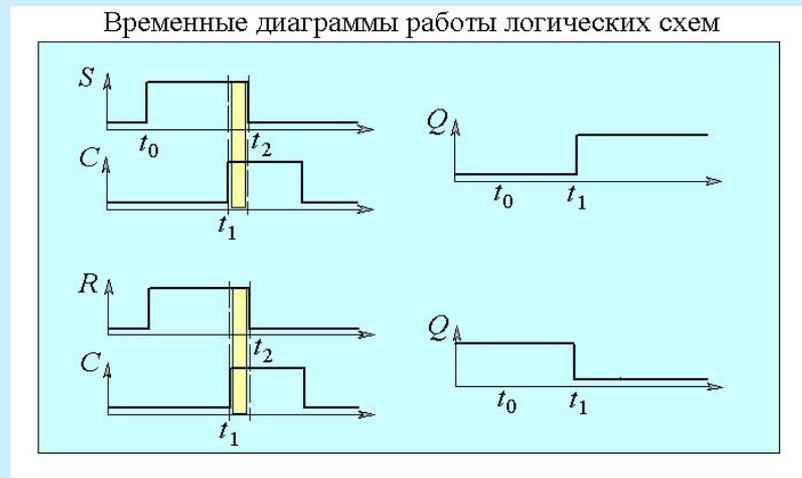
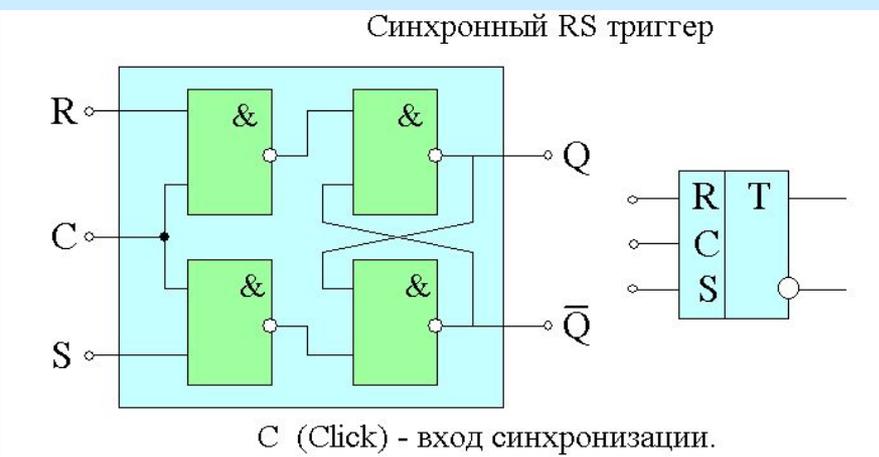
Уравнение состояния

$$Q_{t+1} = S + \bar{R} * Q_t$$

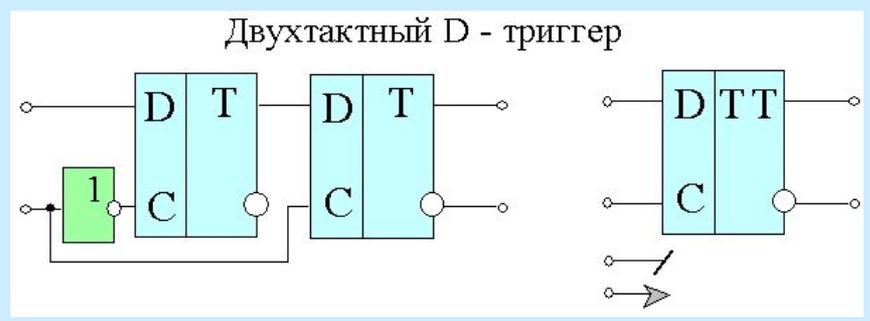
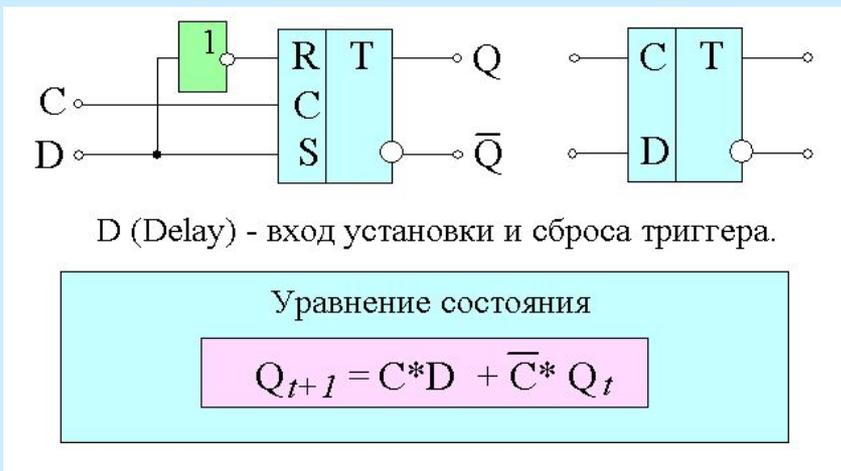
Таблица переходов

S	R	$Q_{t+1}$	действие
0	0	$Q_t$	хранить
0	1	0	сбросить
1	0	1	установить
1	1		запрещено

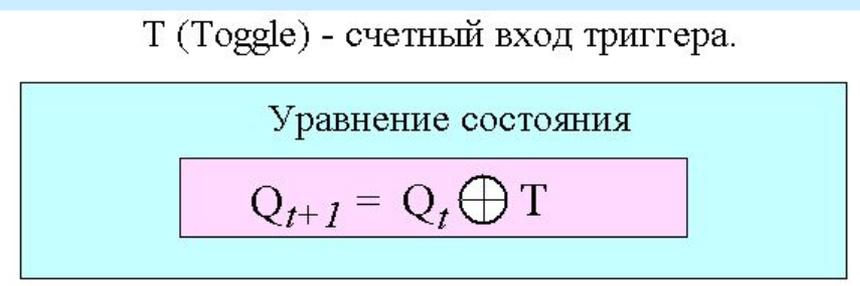
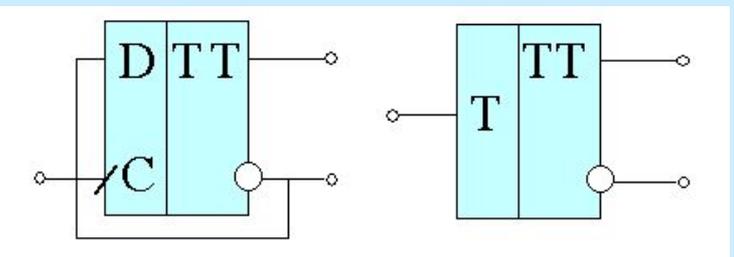
# Синхронный RS-триггер



# D-триггер



# T-триггер



## Основные параметры триггеров

1. нагрузочная способность,
2. задержка переключения.

## Классификация триггеров

### По функциональному назначению

1. с установочным входом –  $RS$ ,
2. задержки –  $D$ ,
3. счетные –  $T$ ,
4. универсальные –  $JK$ .

### По способу управления

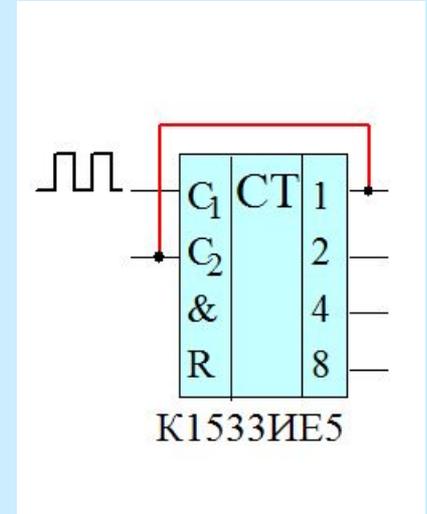
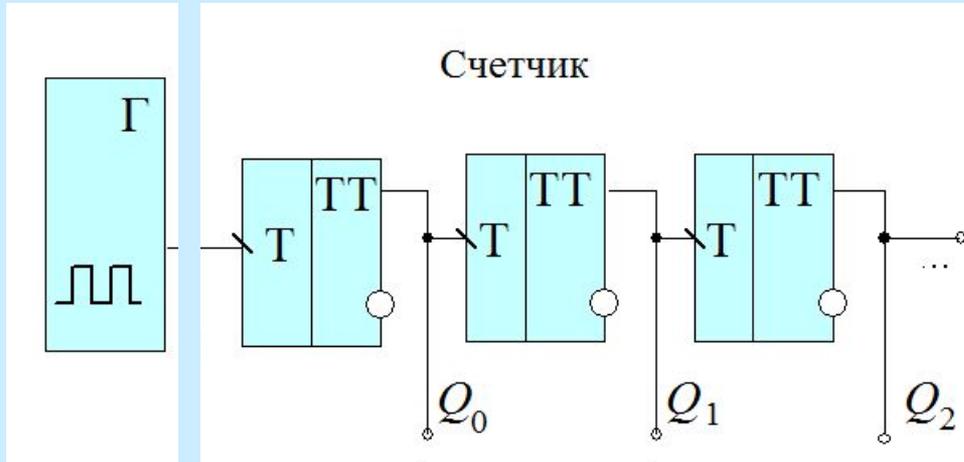
1. статические,
2. динамические.

### По способу записи/чтения

1. асинхронные,
2. синхронные.

# Счетчики

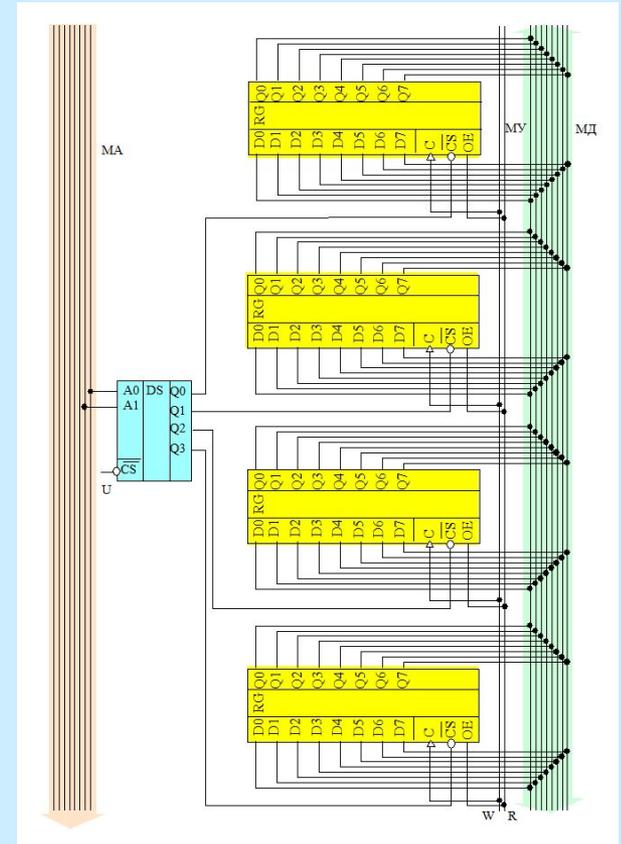
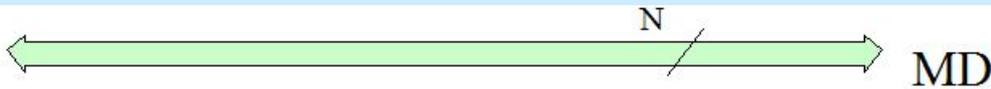
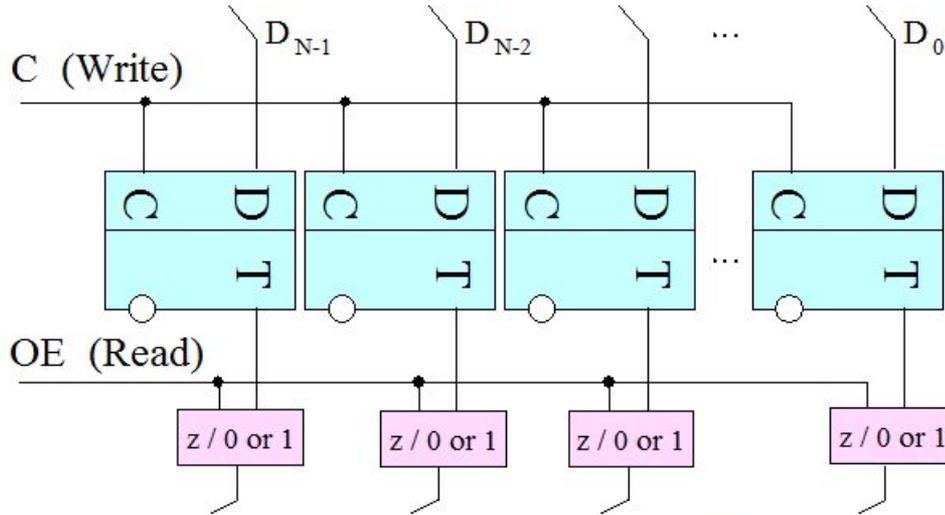
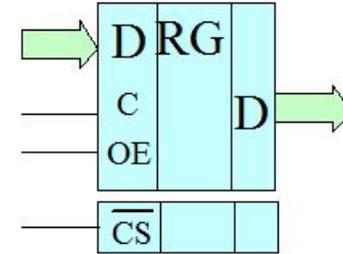
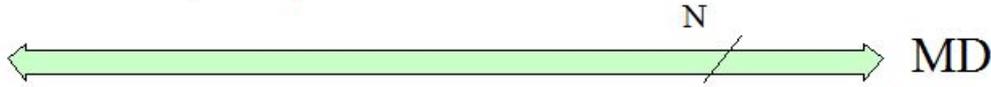
*Счетчики* – устройства, запоминающие количество поступивших на их вход импульсов.



	0	0	0	0	
■	1	0	0	1	
	0	■	1	0	
■	1	■	1	0	
	0		0	■	1
■	1		0	■	1
	0	■	1	■	1
■	1	■	1	■	1

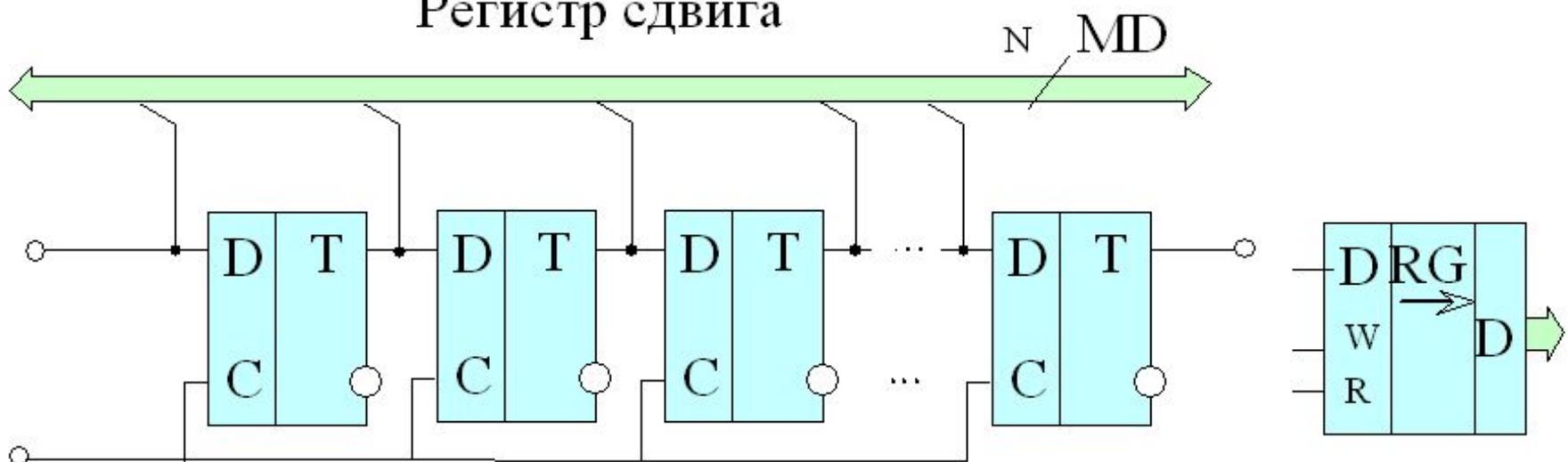
# Регистры

Регистр с параллельной записью/чтением



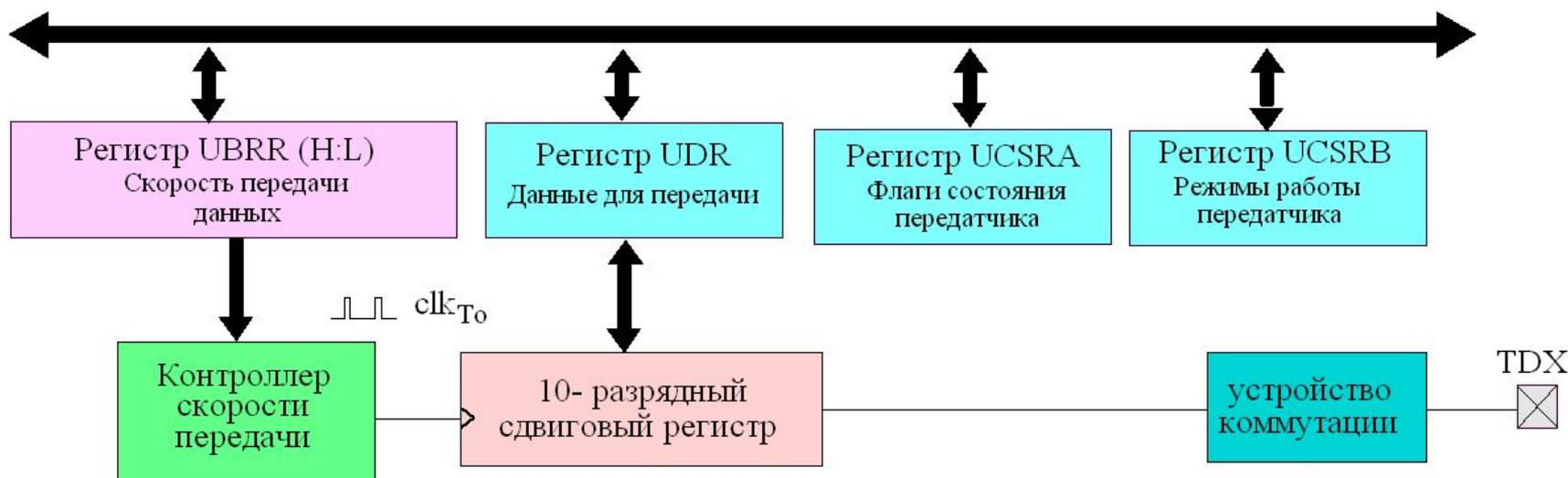
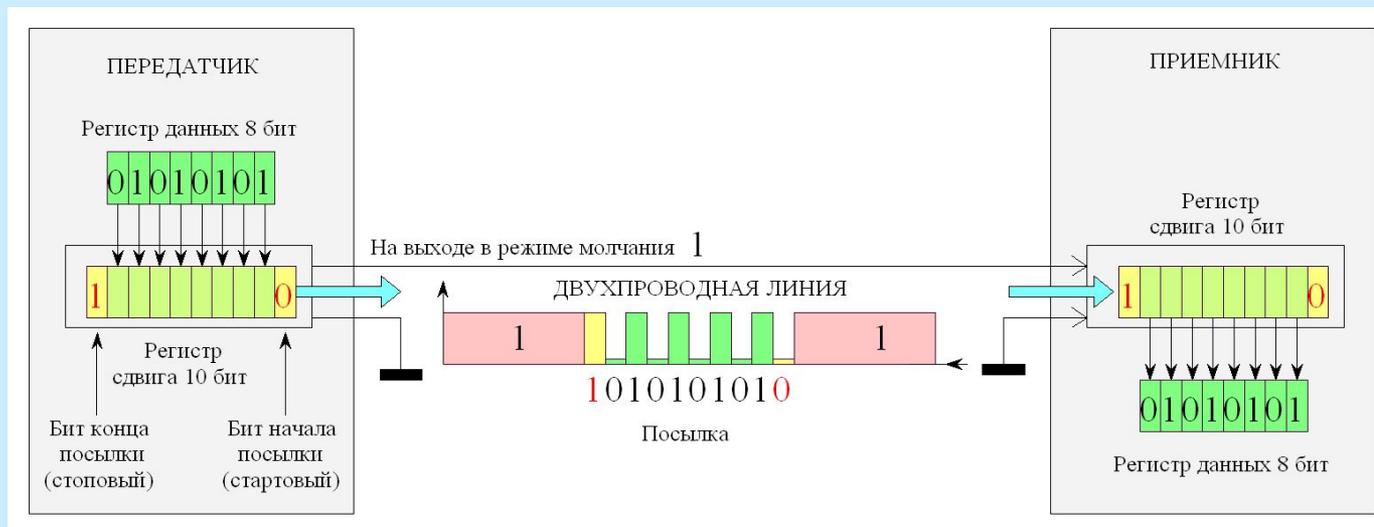
# Регистры

## Регистр сдвига



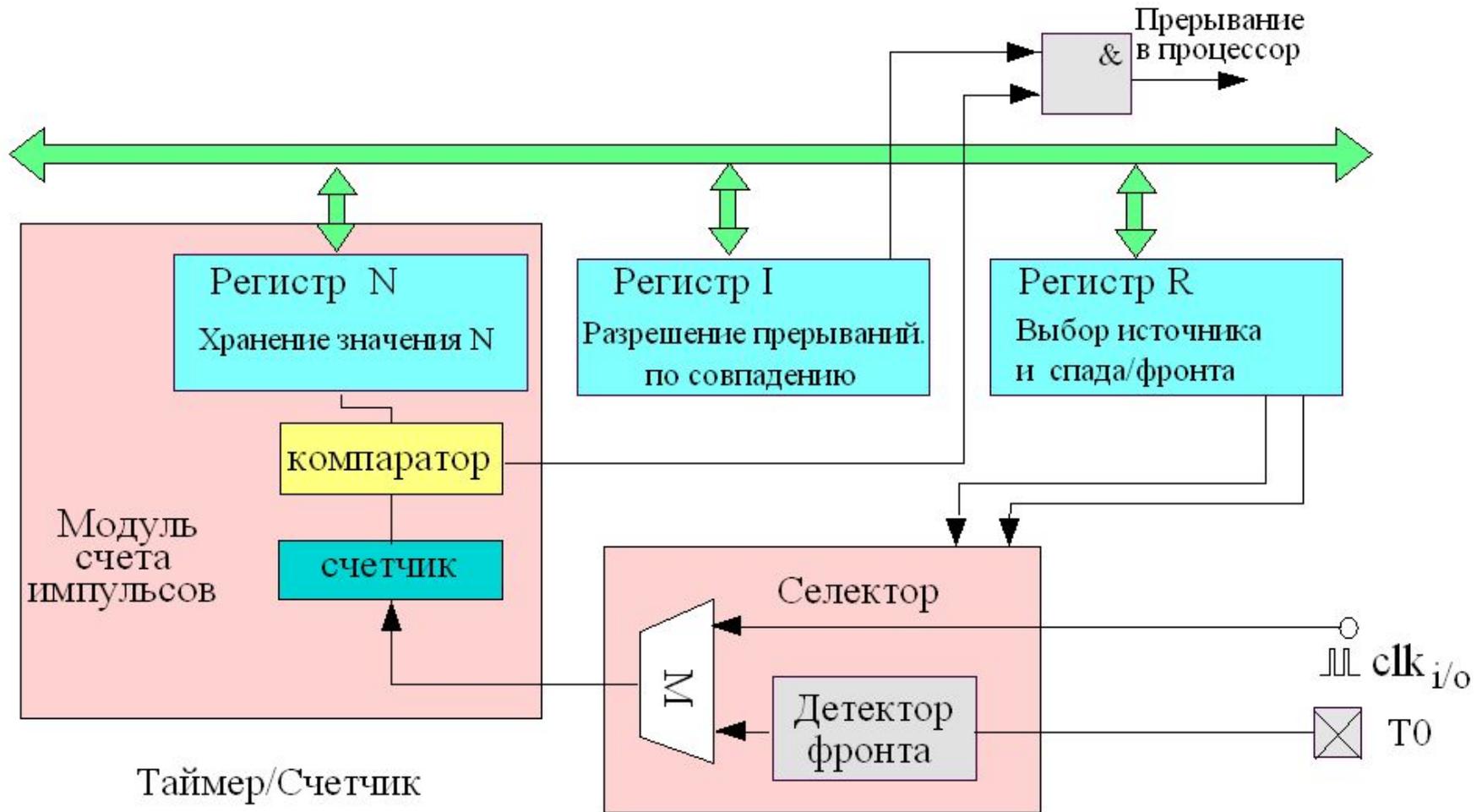
# Универсальный приемник-передатчик

Приемник-передатчик – устройства подготовки цифровой информации для пересылки между удаленными абонентами по последовательному каналу.



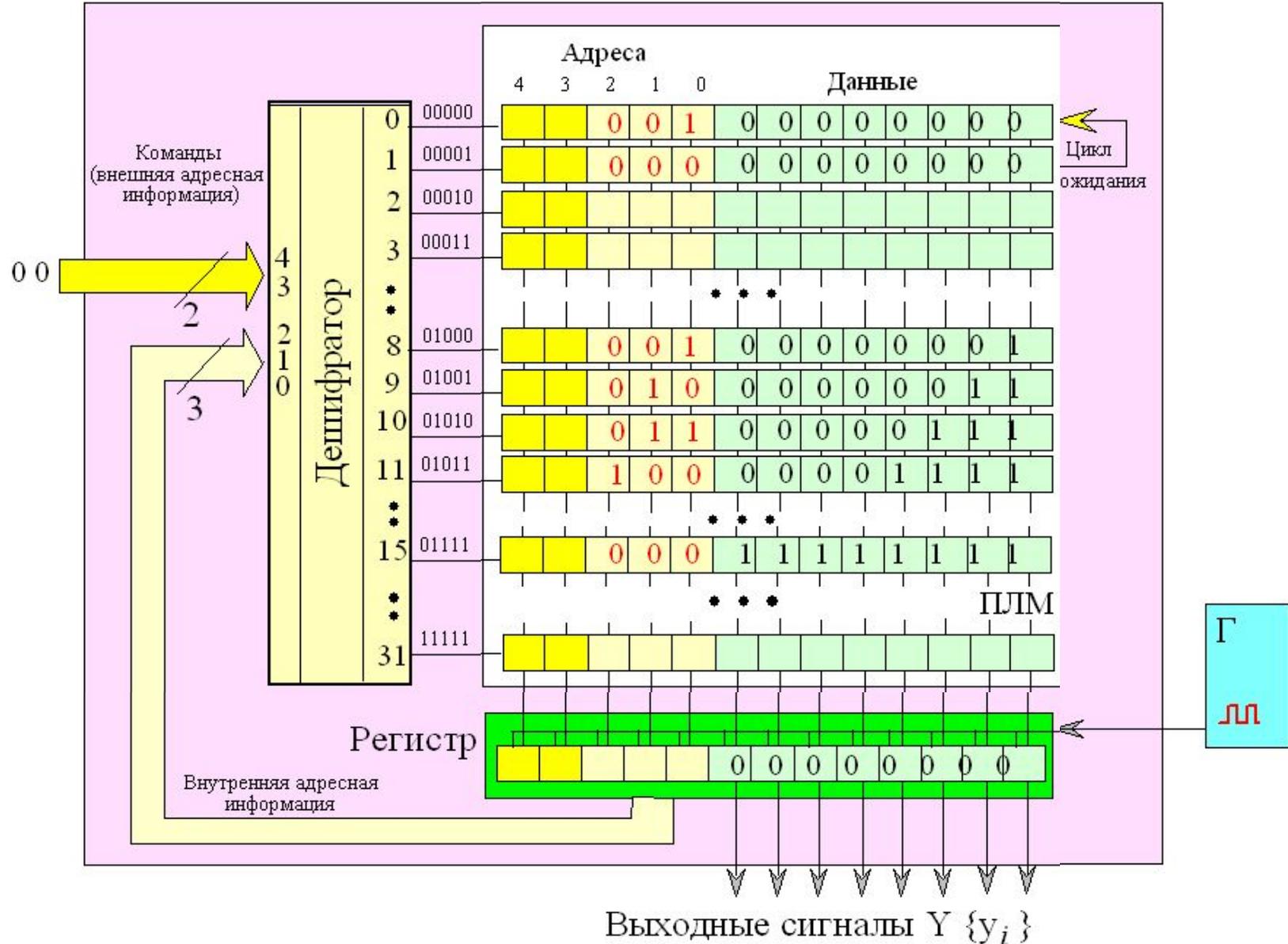
# Таймер

Таймер – устройство контроля интервалов времени.



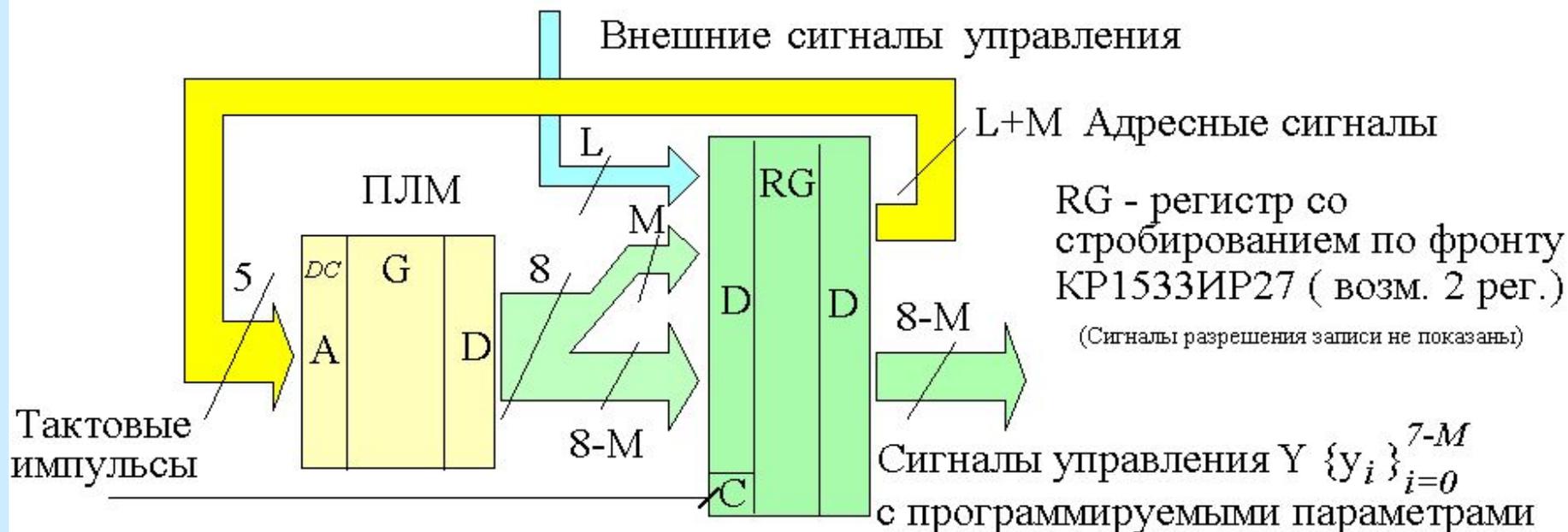
# Микропрограммный автомат

– устройство формирования сигналов требуемой длительности и последовательности.



# Микропрограммный автомат на интегральных элементах

## Микропрограммный автомат (пример)

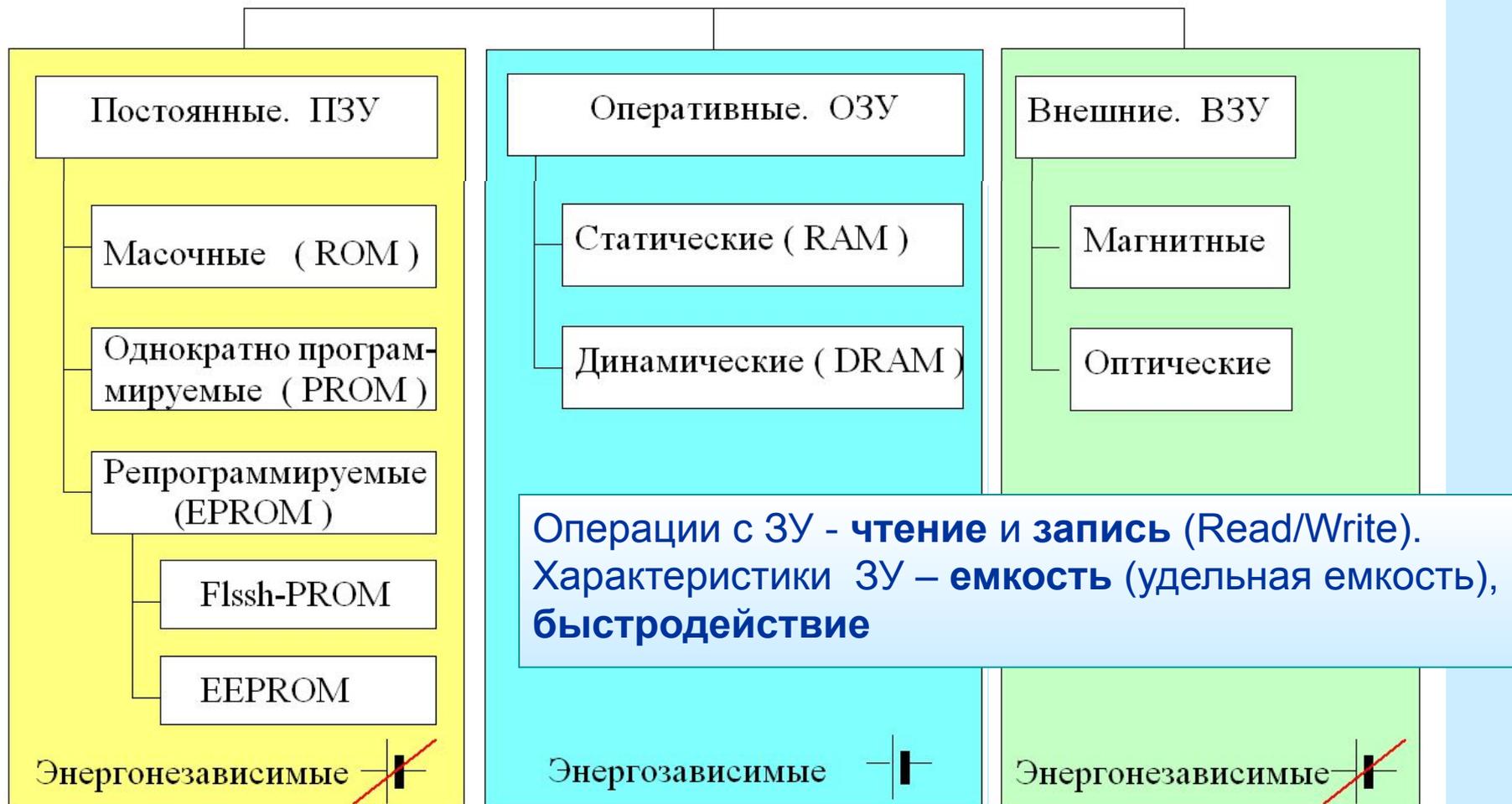


ПЛМ К155РЕЗ с организацией 32x8

Если  $L \leq 2$ , то  $M \leq 3$

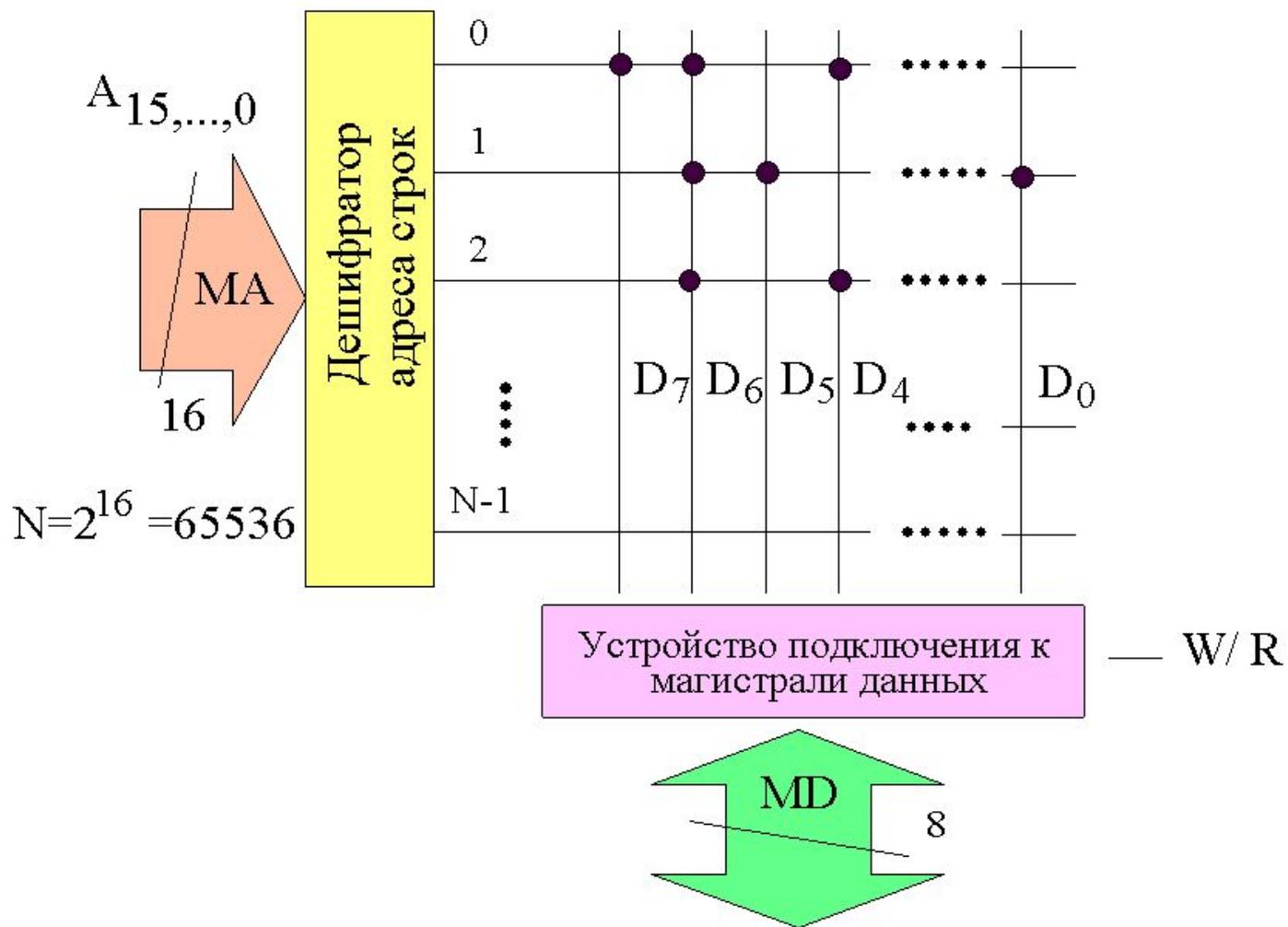
# Память. Запоминающие устройства

## Основные типы запоминающих устройств

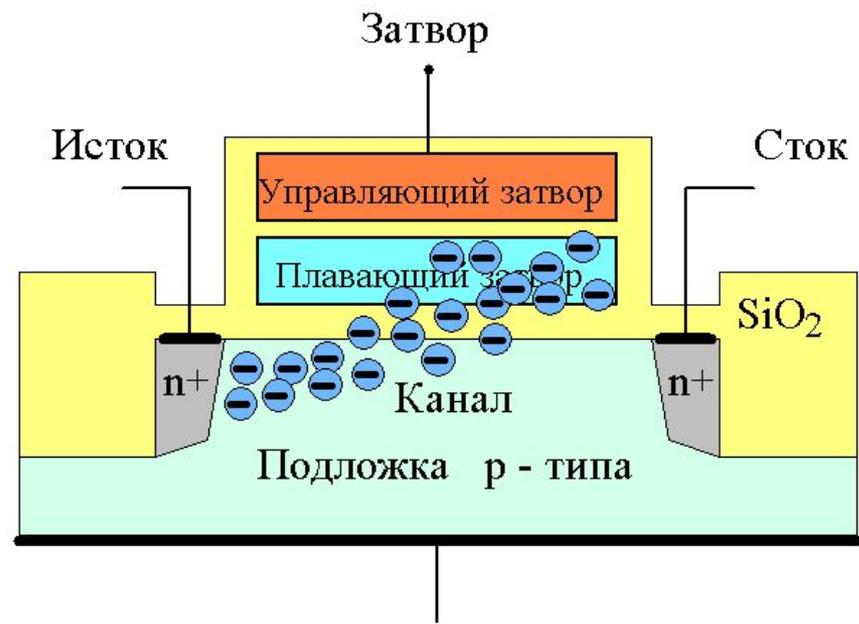
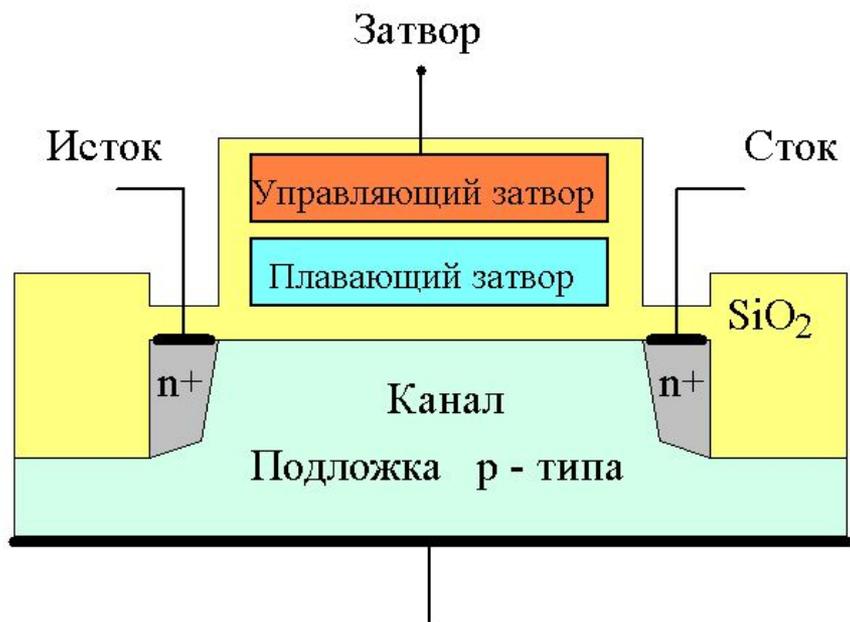


ПЗУ (ROM – Read Only Memory);  
ОЗУ (RAM – Random Access Memory);  
ППЗУ (PROM – Programmable Read Only Memory);  
РППЗУ (EPROM – Erasable Programmable Read Only Memory);  
(EEPROM – Electrically Erasable Programmable Read Only Memory);  
(Flash EEPROM (вспышка, молниеносно стираемая)).

# Масочные ЗУ



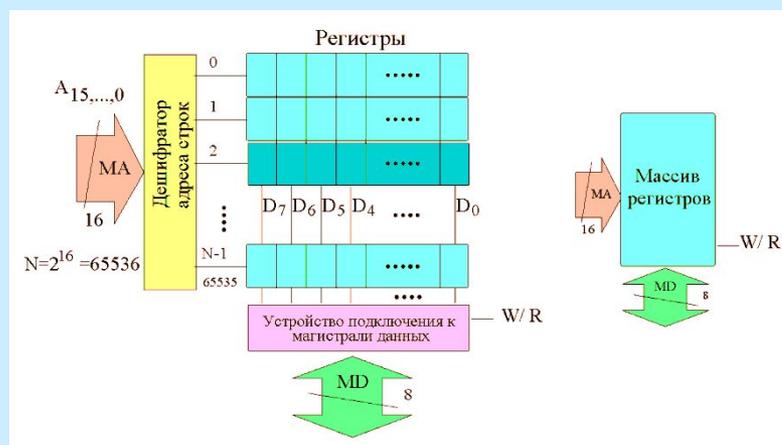
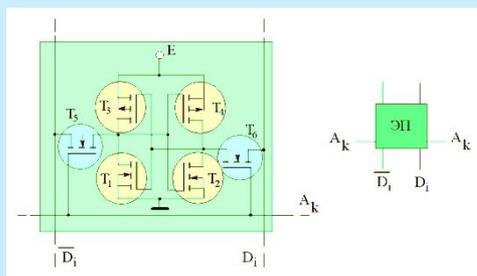
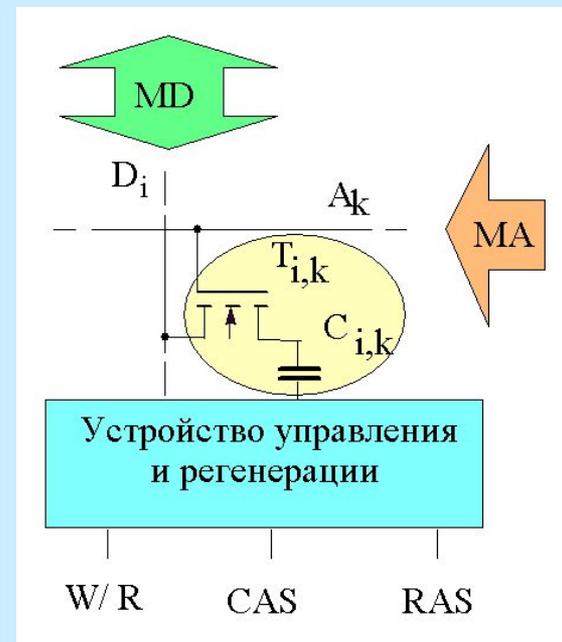
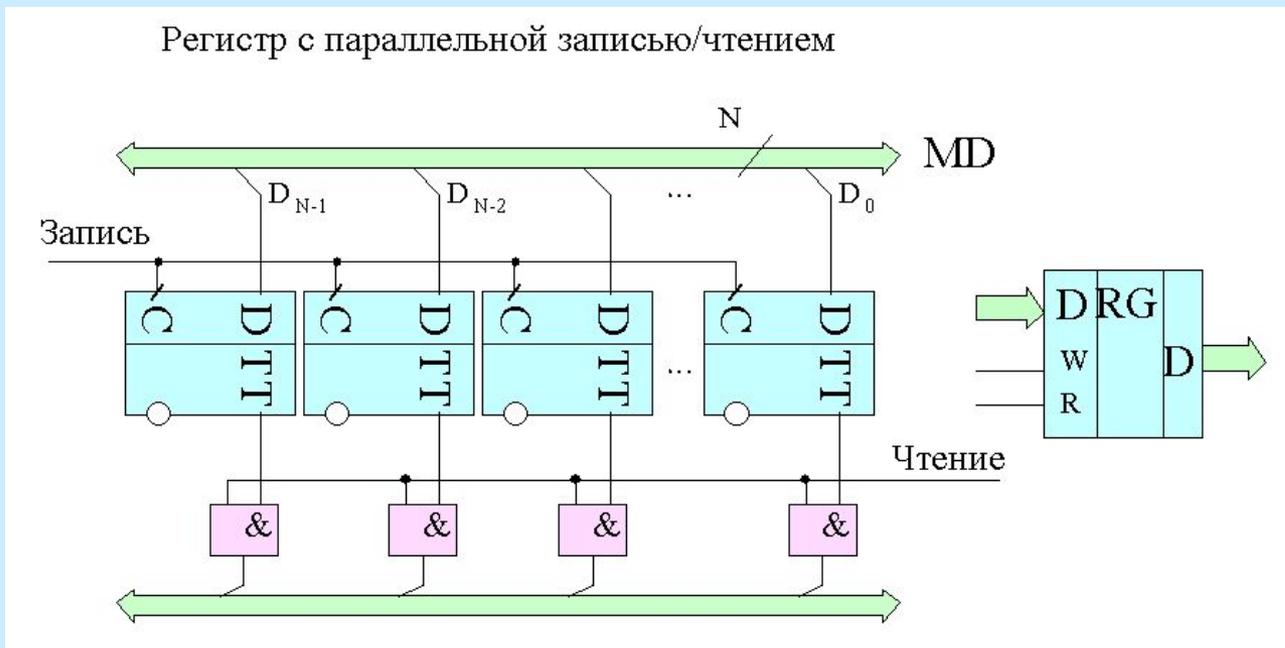
# Репрограммируемые ЗУ



# Оперативные ЗУ

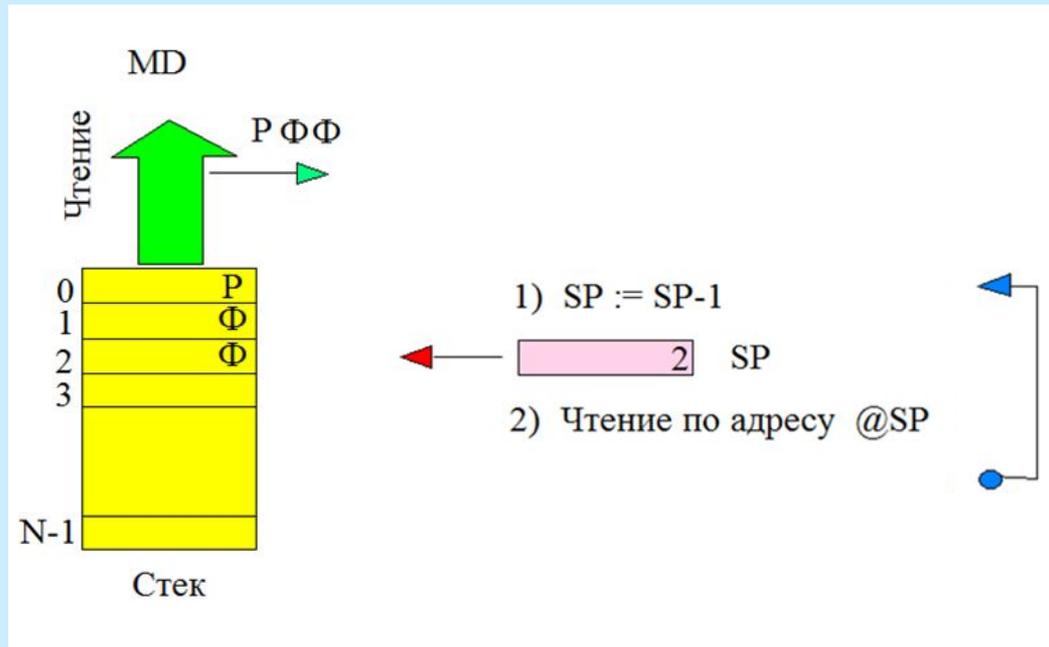
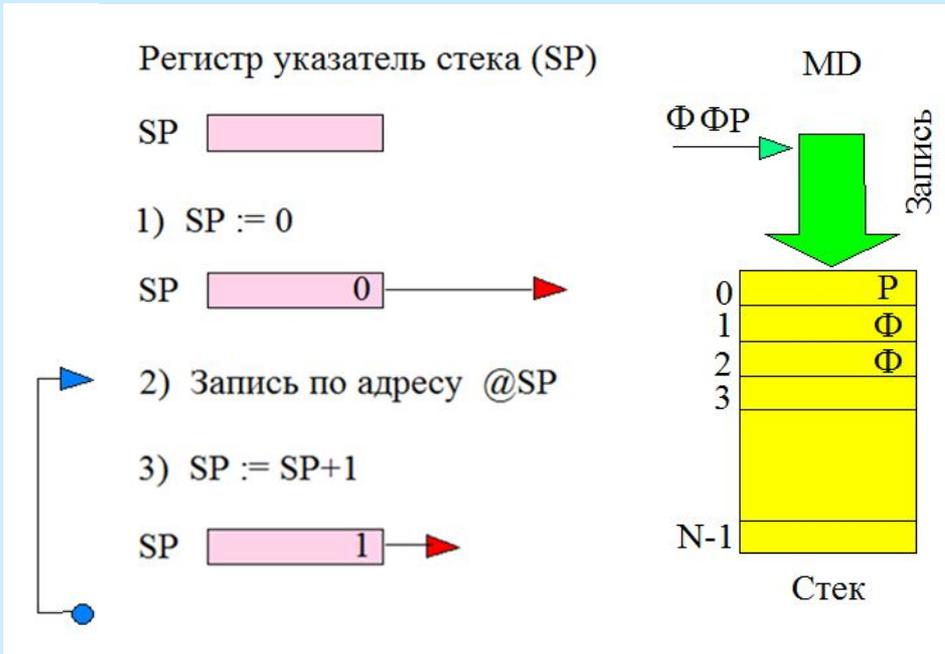
## Статические (SRAM)

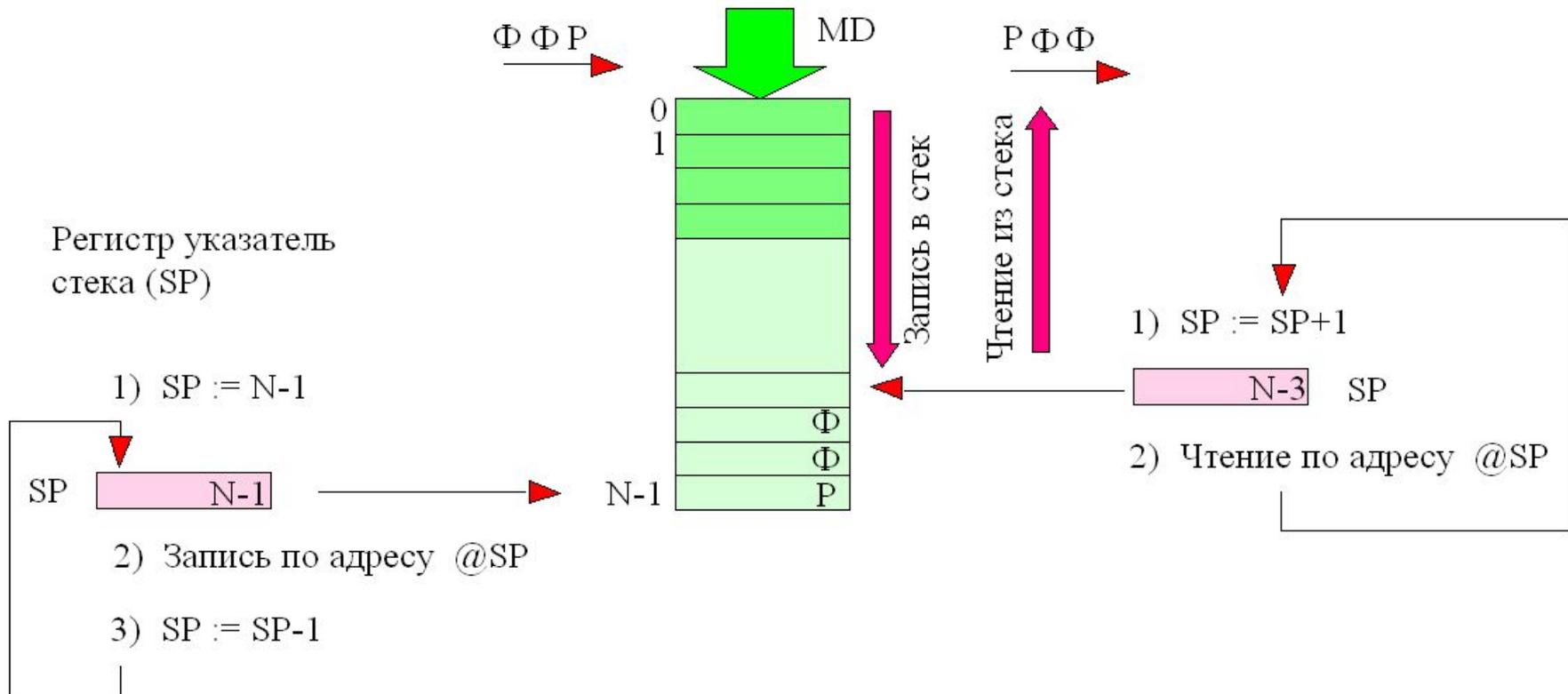
## Динамические (DRAM)





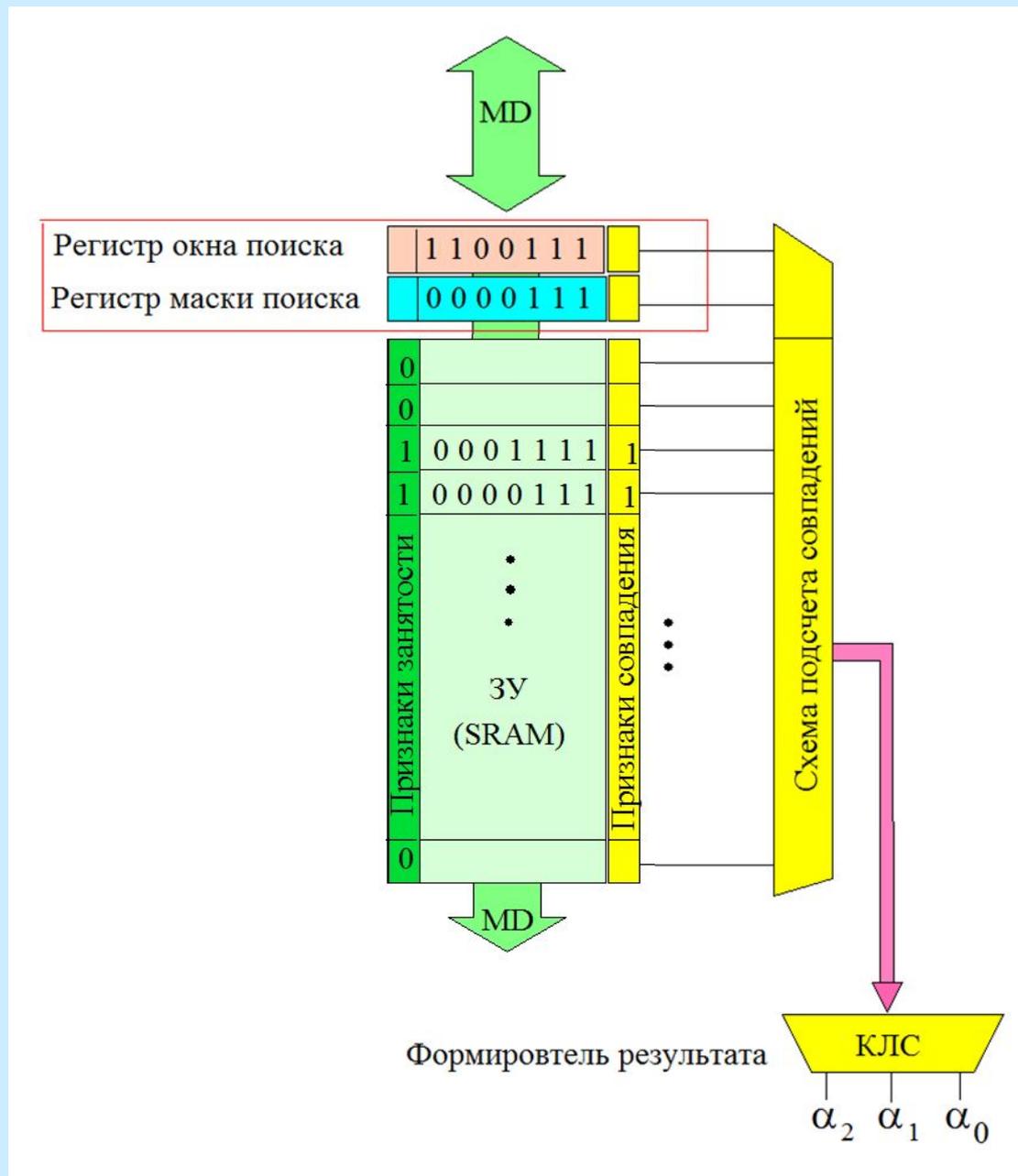
# Стековый (FIFO-LIFO)





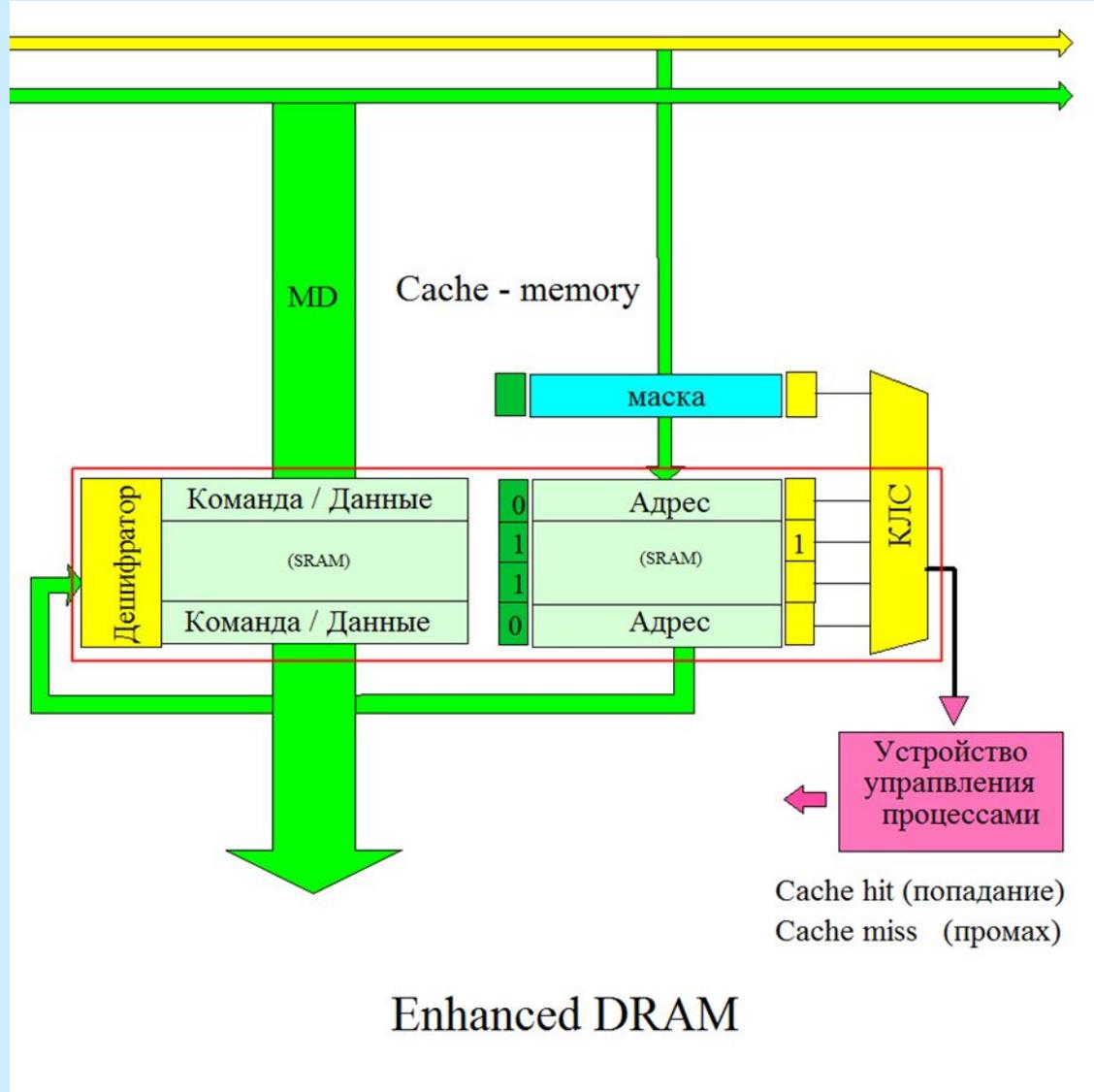
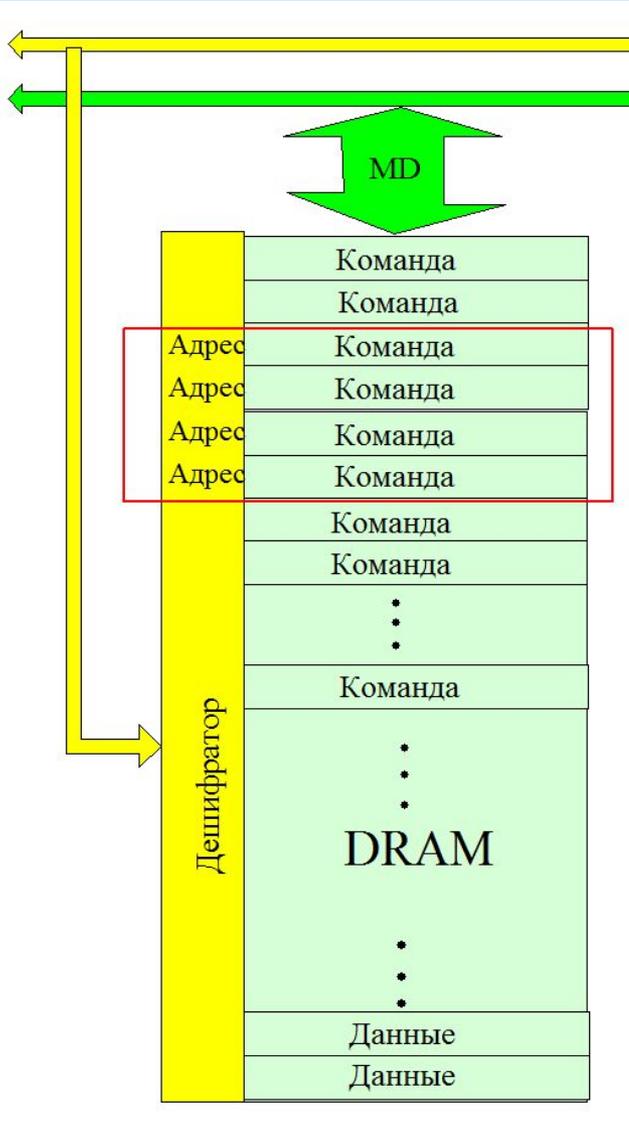
# Ассоциативная память

## Способы доступа к памяти



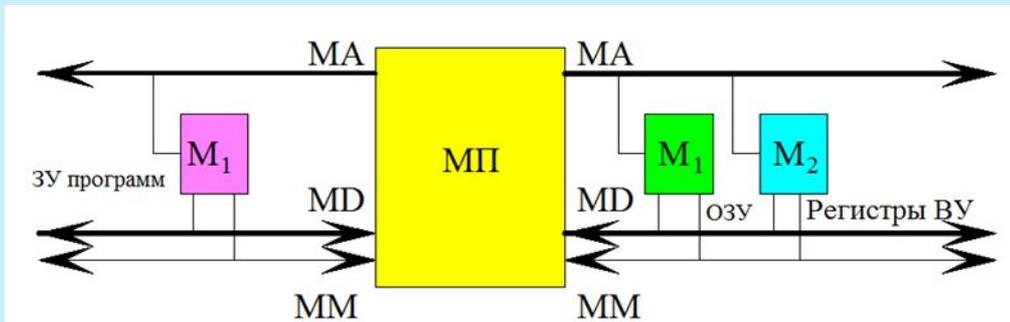
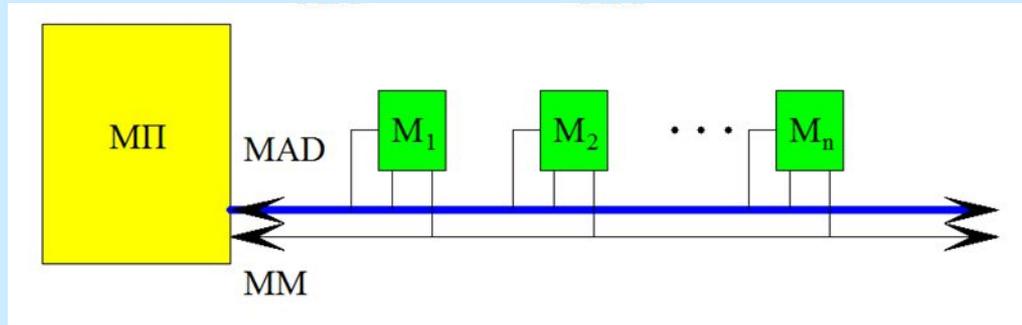
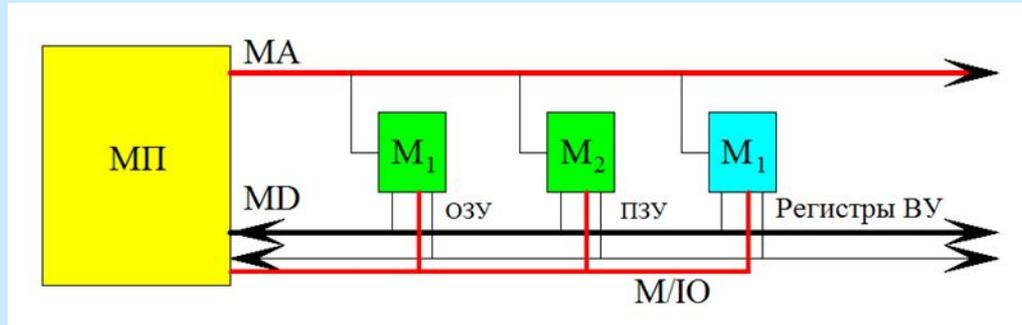
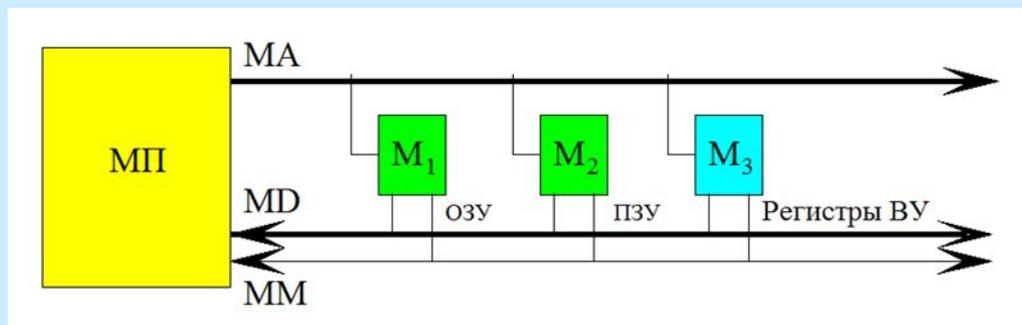
# Cache - память

# Способы доступа к памяти



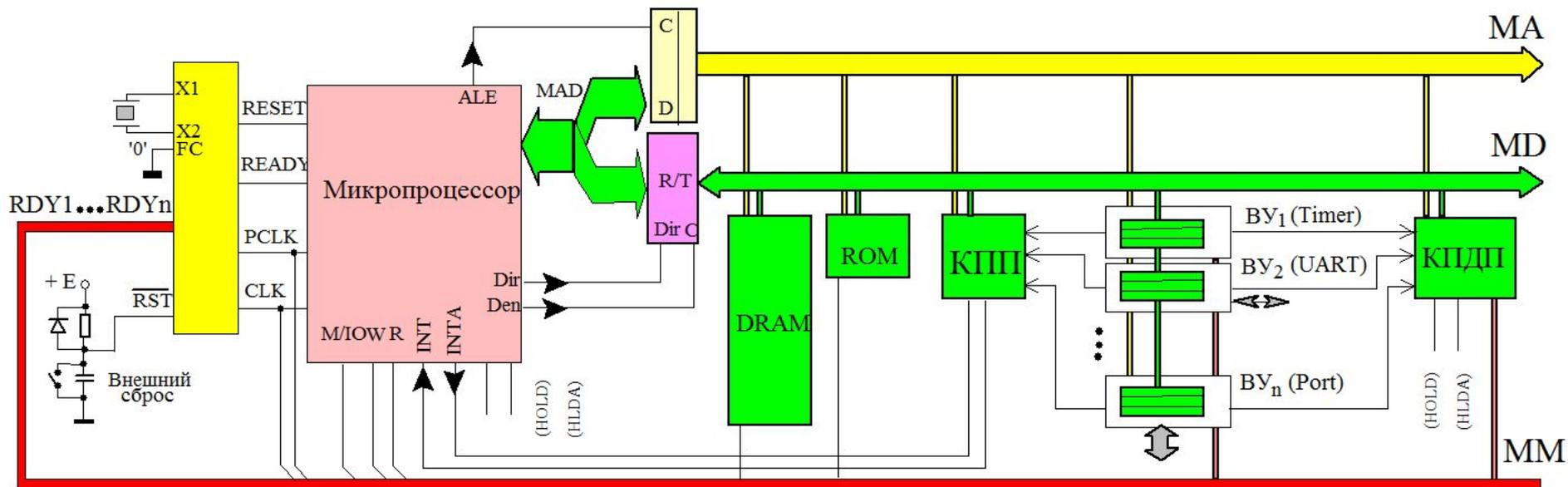
# Архитектура МПС

## Принстон

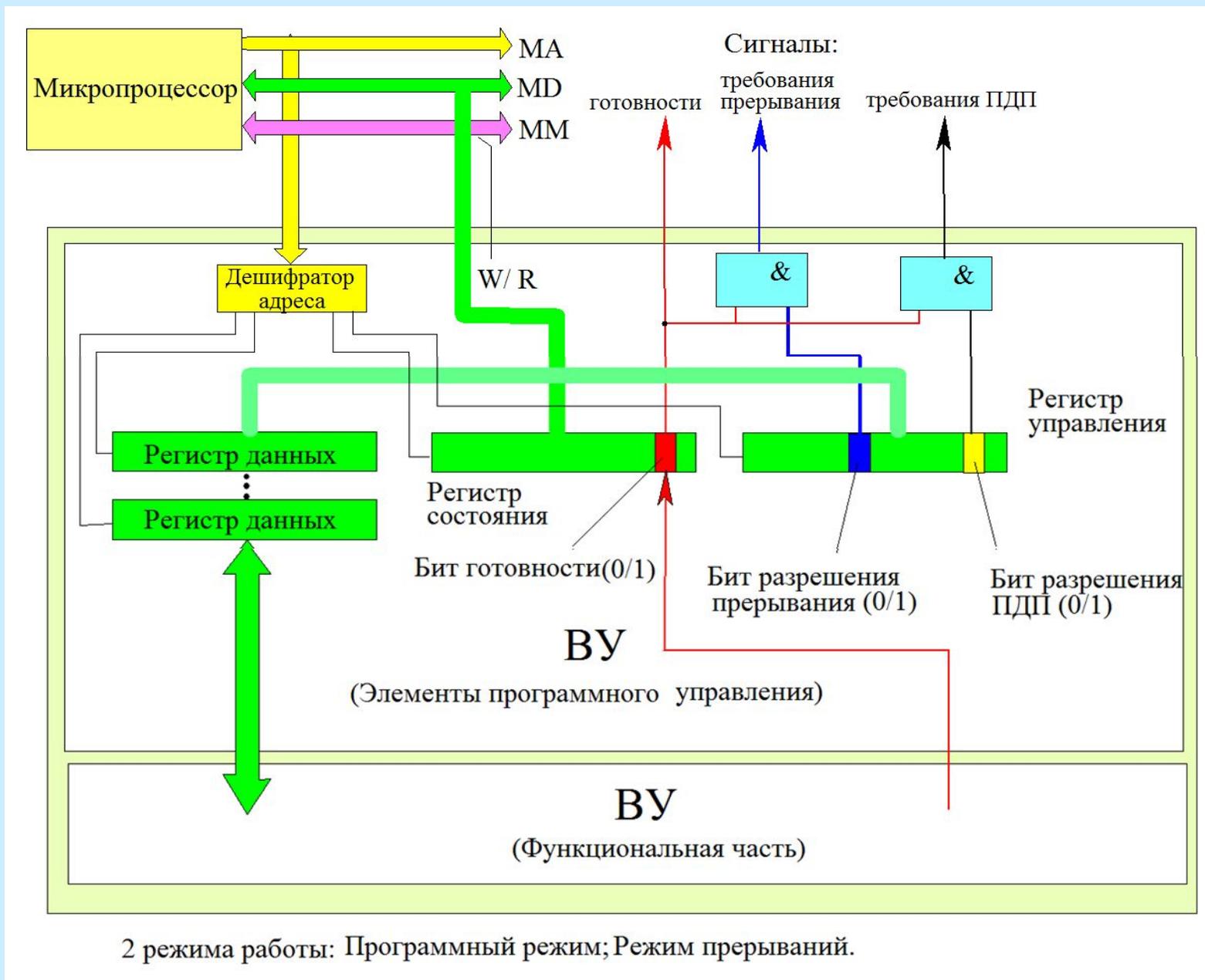


## Гарвард

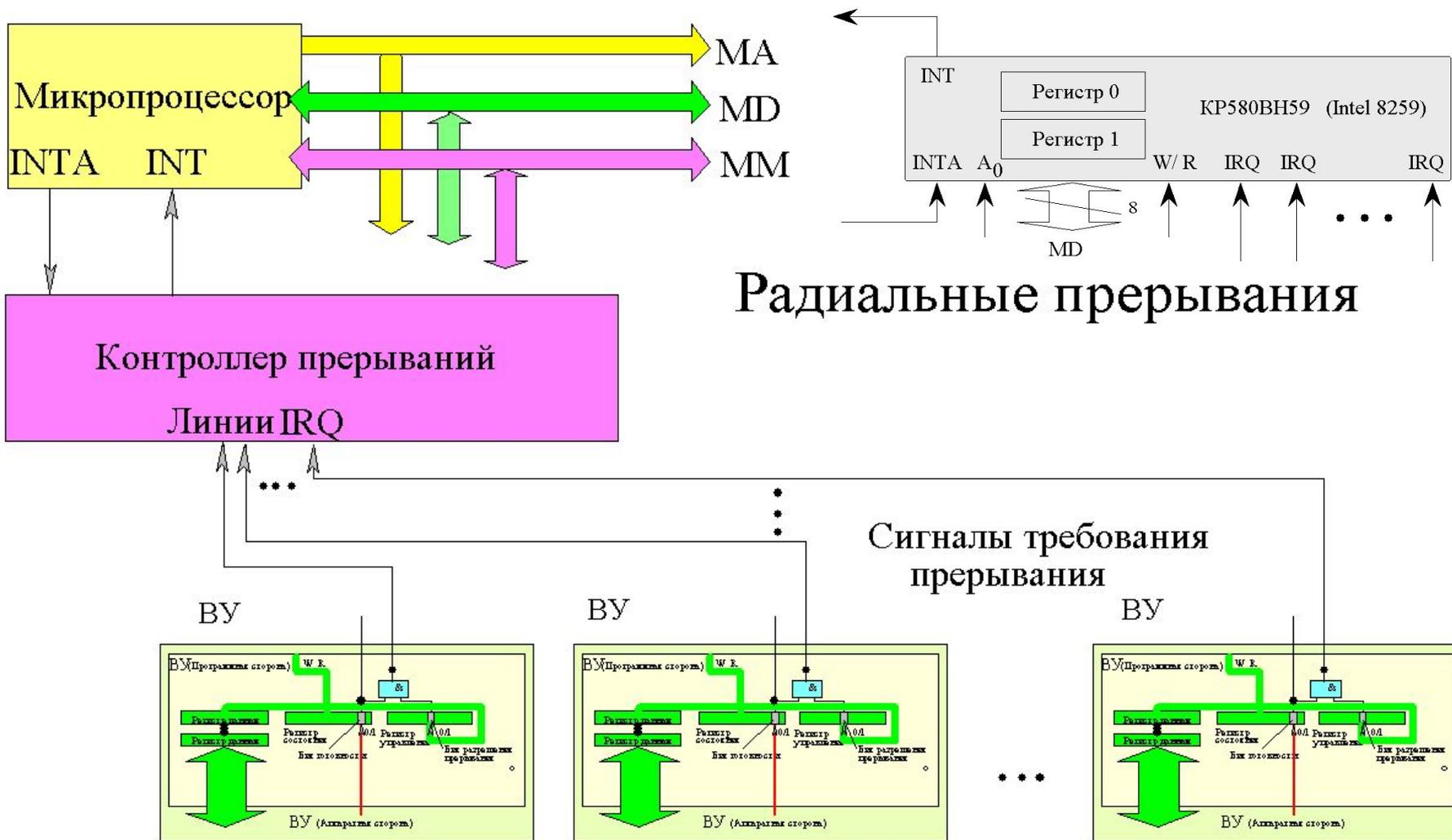
# Структура МПС (Принстон)



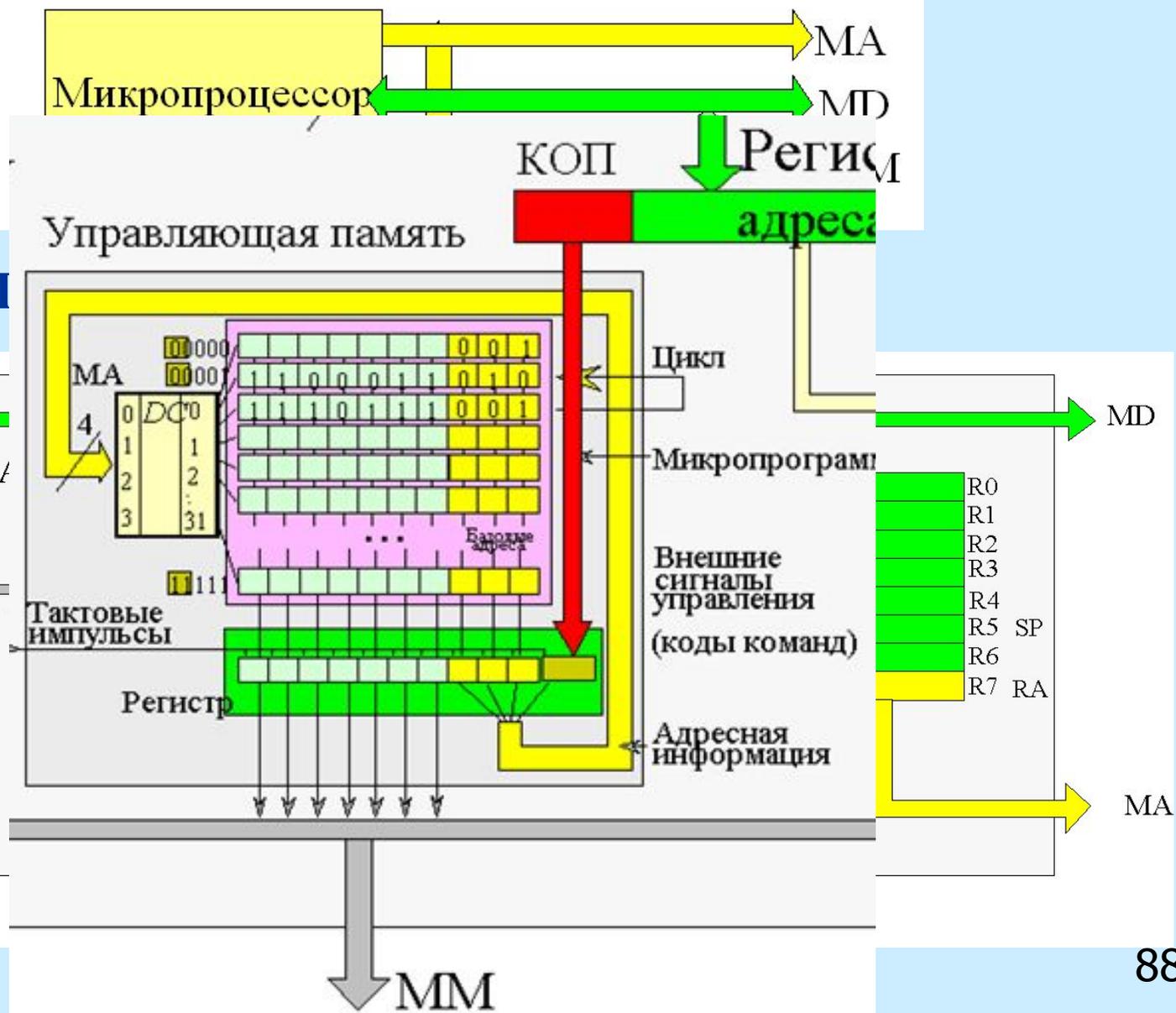
# Модель внешнего устройства (ВУ) МПС.



# Радиальные прерывания.

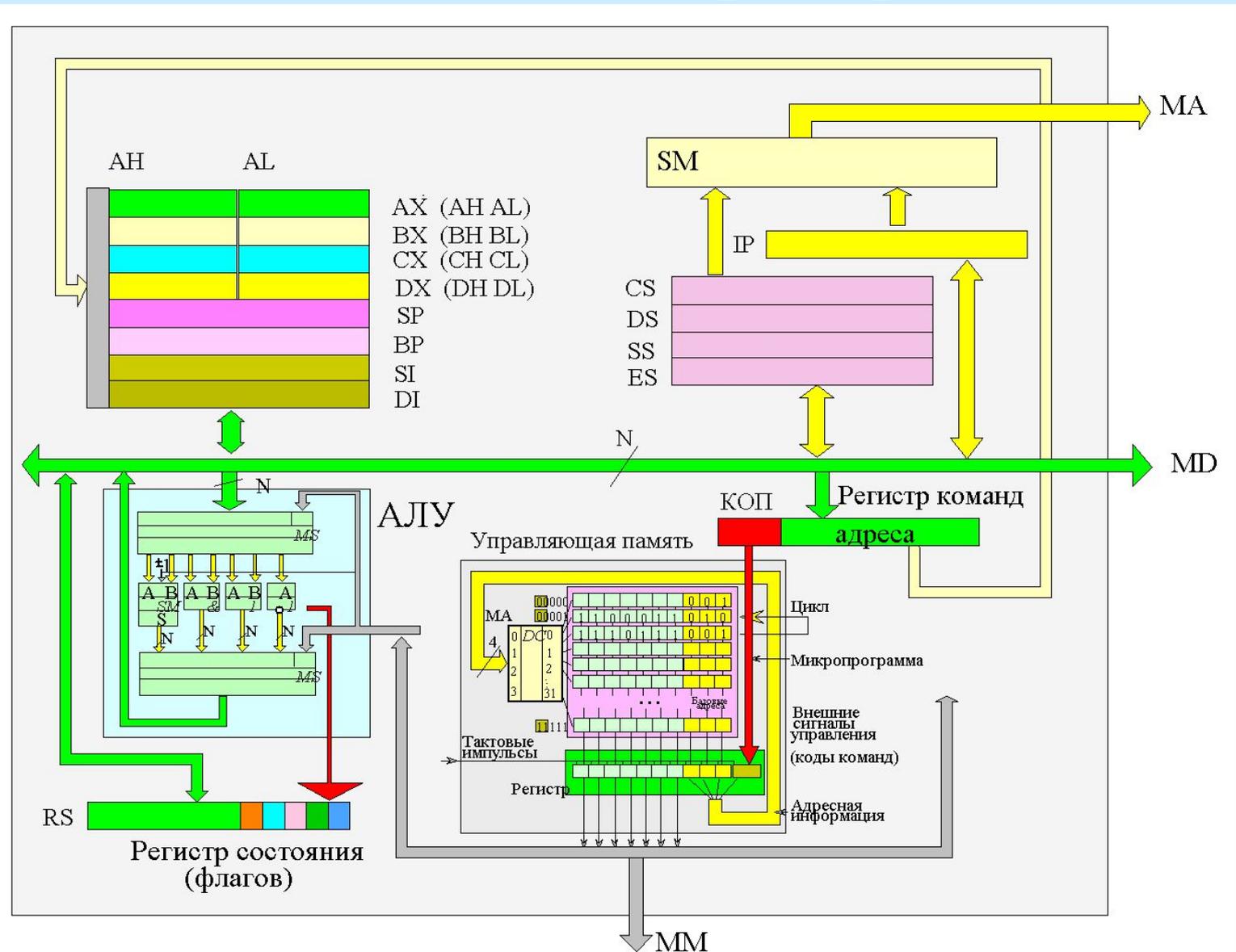


# Архитектура микропроцессора

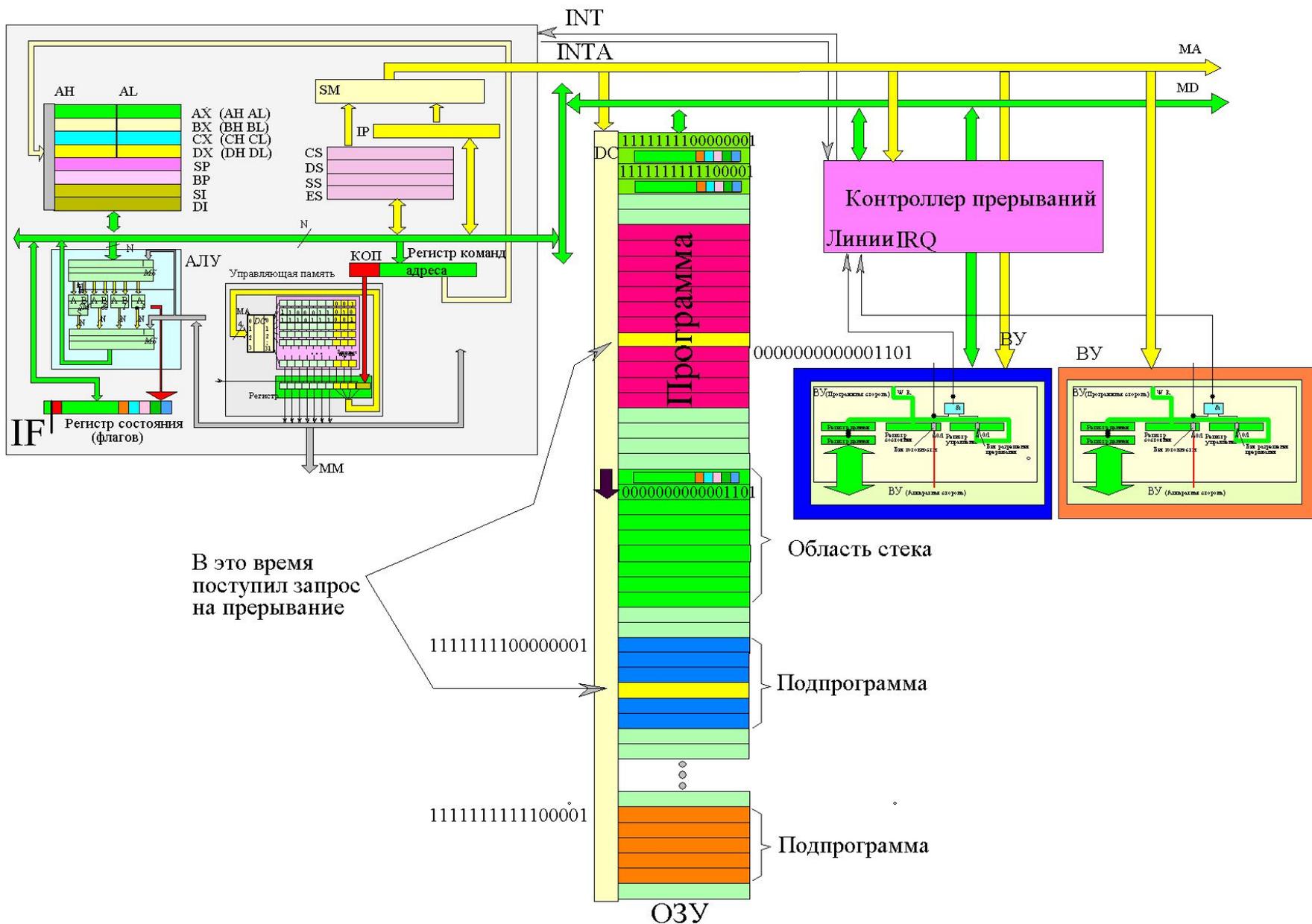


# Архитектура микропроцессора

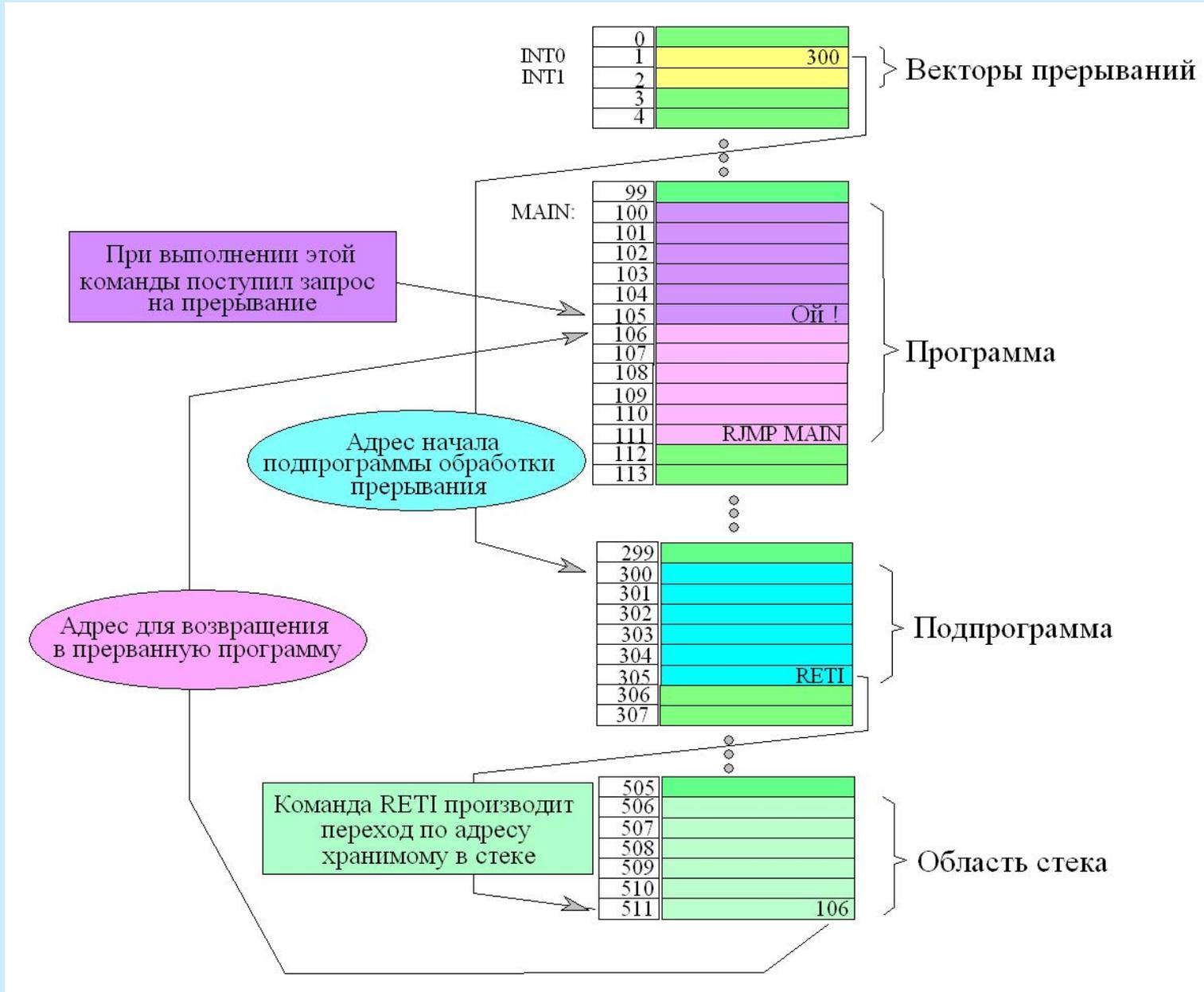
## МП со специальными регистрами



# Микропроцессорная система



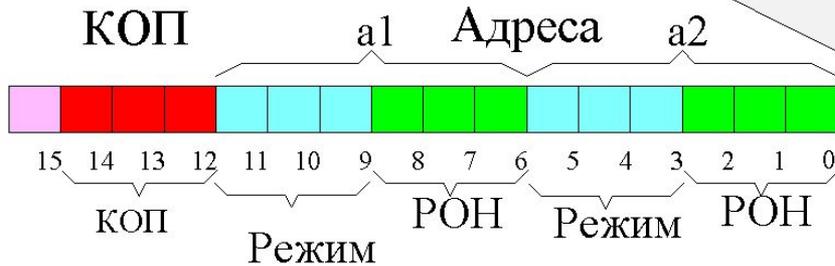
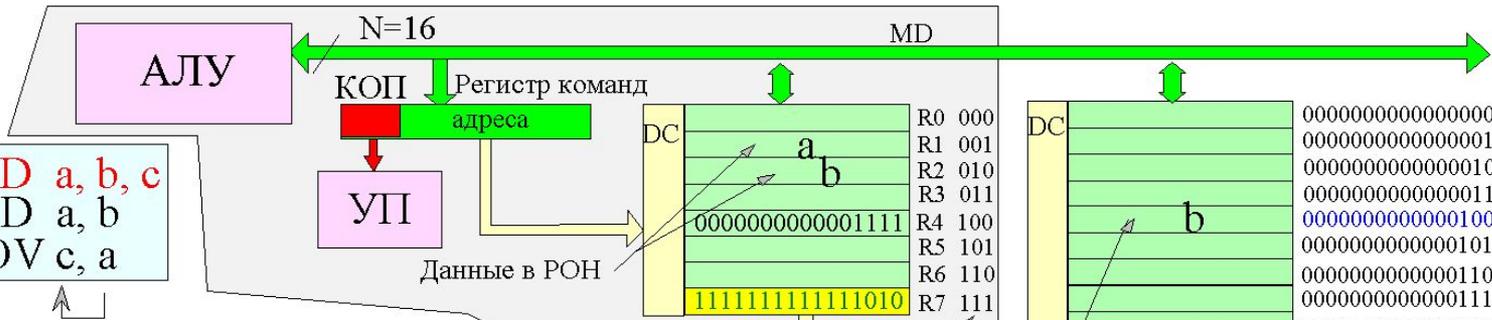
# Распределение памяти. Прерывания



Пример

```

c := a+b  ADD a, b, c
a := a+b  ADD a, b
c := a    MOV c, a
    
```



Некоторые двухадресные команды и их коды

MOV a1,a2	0	0	0
ADD a1,a2	0	0	1
SUB a1,a2	0	1	0
CMP a1,a2	0	1	1
AND a1,a2	1	0	0
OR a1,a2	1	0	1
...	1	1	0
...	1	1	1

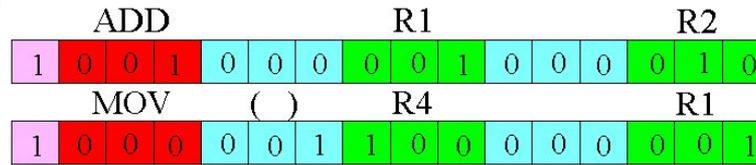
Мнемоники и коды режимов

	0	0	0
( )	0	0	1
( )+	0	1	0
-( )	0	1	1
&( )	1	0	0
...			
	1	1	0

Пример

```

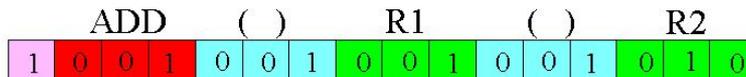
a := a+b  ADD R1, R2  ! Данные в POH
c := a    MOV (R4), R1
    
```



R1 = 0000000000001010  
R2 = 0000000000000100

```

a := a+b  ADD (R1), (R2)  ! Данные в O3Y
    
```

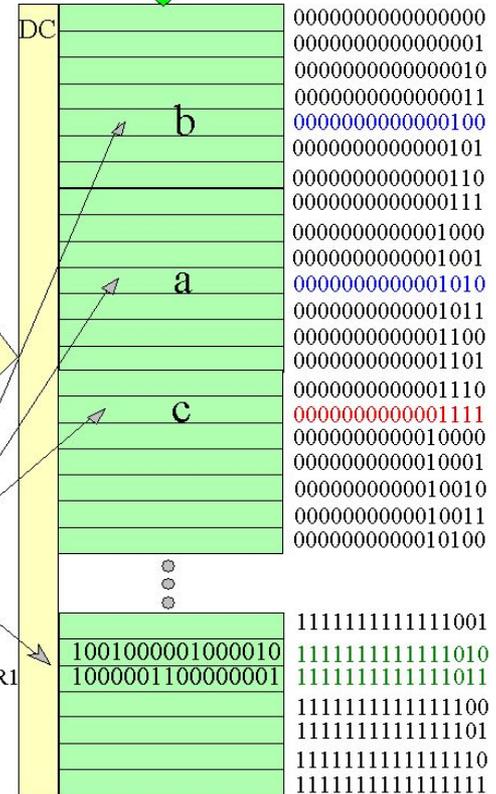


Экзотика

```

MOV &(R4), (R1)+
    
```

Данные в ОЗУ  
Программа в ОЗУ  
ADD R1, R2  
MOV (R4), R1

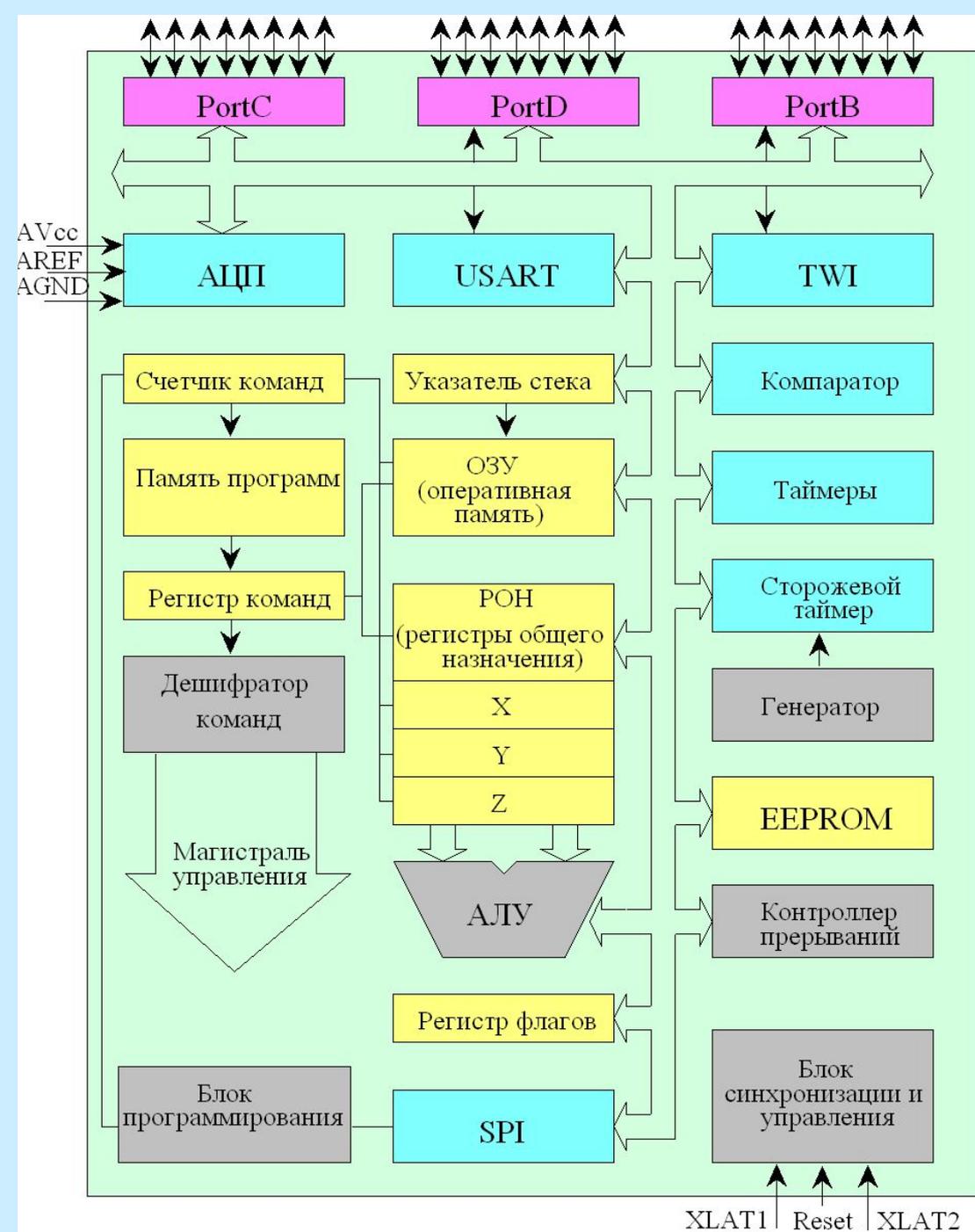


O3Y

адреса ячеек памяти

Команды программы

# Архитектура микроконтроллера



(RESET)	PC <sub>6</sub>	1	28	PC <sub>5</sub>	(ADC <sub>5</sub> /SCL)
(RXD)	PD <sub>0</sub>	2	27	PC <sub>4</sub>	(ADC <sub>4</sub> /SDA)
(TXD)	PD <sub>1</sub>	3	26	PC <sub>3</sub>	(ADC <sub>3</sub> )
(INT <sub>0</sub> )	PD <sub>2</sub>	4	25	PC <sub>2</sub>	(ADC <sub>2</sub> )
(INT <sub>1</sub> )	PD <sub>3</sub>	5	24	PC <sub>1</sub>	(ADC <sub>1</sub> )
(XCK/T <sub>0</sub> )	PD <sub>4</sub>	6	23	PC <sub>0</sub>	(ADC <sub>0</sub> )
	VCC	7	22		AGND
	GND	8	21		AREF
					AVCC
(XTAL <sub>1</sub> /TOSQ)	PB <sub>6</sub>	9	20		PB <sub>5</sub> (SCK)
(XTAL <sub>2</sub> /TOSQ)	PB <sub>7</sub>	10	19		PB <sub>4</sub> (MISO)
(T <sub>1</sub> )	PD <sub>5</sub>	11	18		PB <sub>3</sub> (MOSI/OC <sub>2</sub> )
(AIN <sub>0</sub> )	PD <sub>6</sub>	12	17		PB <sub>2</sub> (SS/OC <sub>1B</sub> )
(AIN <sub>1</sub> )	PD <sub>7</sub>	13	16		PB <sub>1</sub> (OC <sub>1A</sub> )
(ICP)	PB <sub>0</sub>	14	15		

