

# Модуль 3. Функциональные узлы последовательного типа.

Триггерные схемы. Бистабильная ячейка. Таблицы истинности триггерных схем. Схема устранения дребезга контактов. Асинхронные и синхронные триггеры. Однотактные и двухтактные триггеры.

Регистры. Классификация регистров. Параллельные и последовательные регистры. Парафазные и однофазные регистры. Сдвигающие регистры.

**Счетчики импульсов. Синтез счетчиков. Двоичные счетчики.**

**Счетчики с переменным модулем счета. Суммирующие, вычитающие и реверсивные счетчики.**

**Счётчиком наз. последовательностное устройство для счета входных импульсов и фиксации их числа и двоичном коде.** Состоит из запоминающих ячеек – триггеров обычно *D*- или *JK*-типа. Между собой ячейки счётчика соединяют таким образом, чтобы каждому числу импульсов соответствовали состояния 1 или 0 определенных ячеек. При этом **совокупность единиц и нулей на выходах  $n$  ячеек, наз. разрядами счетчика, представляет собой  $n$ -разрядное двоичное число, которое однозначно определяет количество прошедших через входы импульсов.**

Каждый разряд счётчика может находиться в двух состояниях. Число устойчивых состояний, которое может принимать данный счётчик, называют коэффициентом пересчёта  $K_{сч}$ .

**В цифровых схемах счётчики выполняют следующие микрооперации над кодовыми словами:**

- установка в исходное состояние (запись нулевого кода);**
- запись входной информации в параллельной форме;**
- хранение информации;**
- выдача хранимой информации в параллельной форме;**
- инкремент—увеличение хранящегося кодового слова на единицу;**
- декремент — уменьшение хранящегося кодового слова на единицу.**

Если с каждым входным импульсом "записанное" в счётчике число увеличивается, счётчик является *суммирующим*, если же оно уменьшается, то – *вычитающим*. Счётчик, работающий на сложение и на вычитание, наз. *реверсивным*.

Счётчики входят в состав разнообразных цифровых устройств: электронных часов, делителей частоты, распределителей импульсов, вычислительных и управляющих устройств. Выпускаемые промышленностью интегральные счётчики

Каждый разряд счётчика может находиться в двух состояниях. Число устойчивых состояний, которое может принимать счётчик, называют коэффициентом (модулем) счёта  $K_{сч}$ .

Максимальное число  $N$ , которое может быть записано в счётчике, равно  $(2^n - 1)$ , где  $n$  – число разрядов счётчика.

По способу кодирования последовательных состояний различают двоичные счётчики с коэффициентами счёта  $K_{сч} = 2^n$ , у которых порядок смены состояний триггеров соответствует последовательности двоичных чисел, и недвоичные (двоично-кодированные) с  $K_{сч} < 2^n$  (десятичные с коэффициентом  $K_{сч} = 10$  или делители частоты с коэффициентом деления  $K_{сч} \neq 2^n$ ).

По направлению счёта счётчики подразделяют на: суммирующие, выполняющие микрооперацию инкремента над хранящимся кодовым словом;

вычитающие, выполняющие микрооперацию декремента над хранящимся кодовым словом;

реверсивные, выполняющие в зависимости от значения управляющего сигнала над хранящимся кодовым словом либо микрооперацию декремента, либо инкремента.

Счётчики, у которых под воздействием входного импульса переключение соответствующих разрядов происходит последовательно друг за другом, называют

**счетчики с последовательным переносом, в которых переключение триггеров разрядных схем осуществляется последовательно один за другим;**  
**счетчики с параллельным переносом, в которых переключение всех триггеров разрядных схем осуществляется одновременно по сигналу синхронизации; ~**  
**счетчики с комбинированным последовательно-параллельным переносом, при котором используются различные комбинации способов переноса.**

- По способу кодирования внутренних состояний счётчики делятся на:
- двоичные счётчики;
- двоично-десятичные (или декадные) счётчики;
- счетчики, работающие в коде Грея;
- счётчики Джонсона.

- Обычный счетчик перебирает свои состояния в возрастающем порядке (суммирующий счетчик).
- Если наоборот, то это вычитающий счетчик.
- Если можно менять направление перебора- реверсивный счетчик.
- Если для переключения нужен синхросигнал, счетчик называется синхронным, если только входной сигнал, то асинхронным.

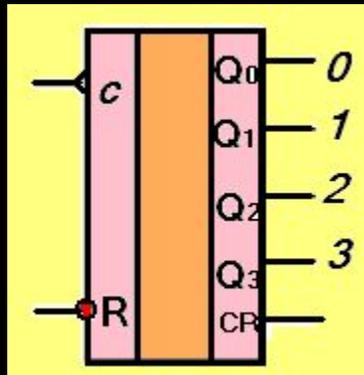
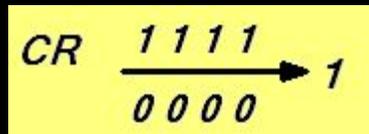


Рис. Схемное обозначение счетчика

- CR – выход переноса, который используется для соединения со следующим счетчиком.

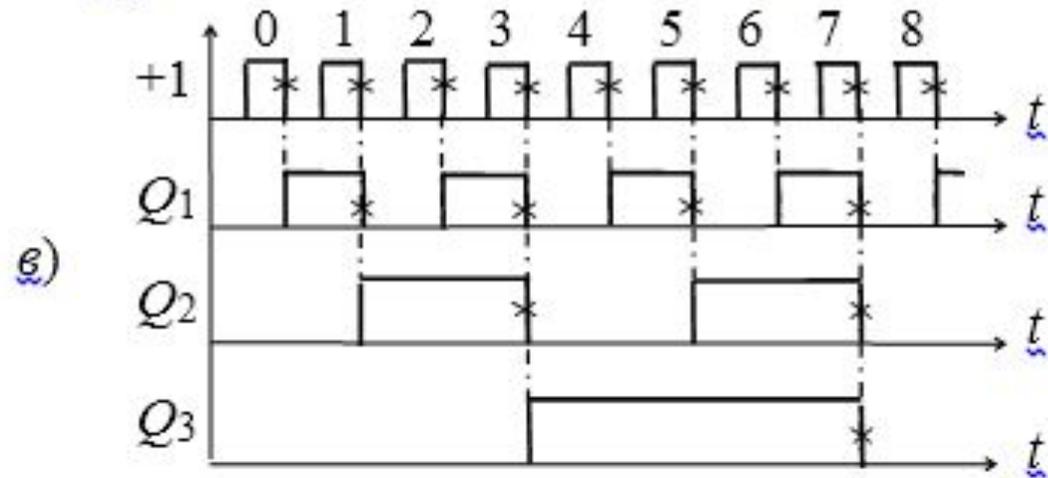
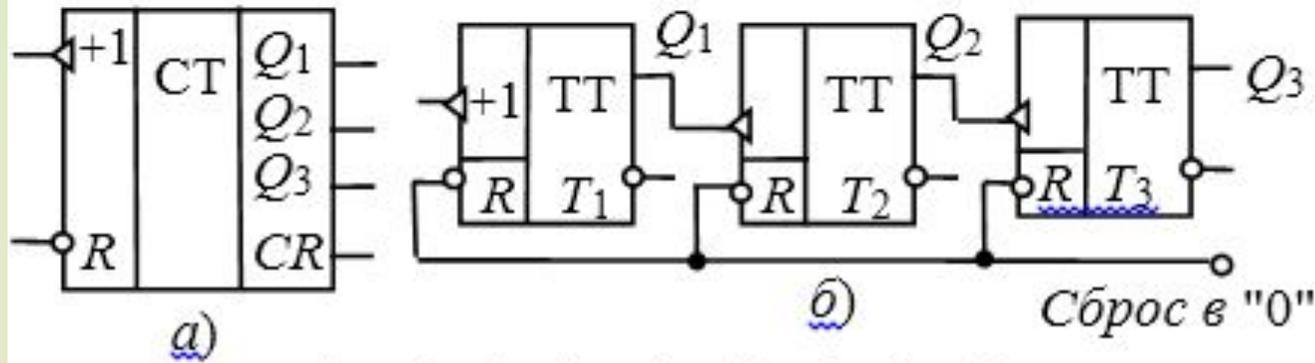
### Виды связи между триггерами счетчиками

- **непосредственная связь** - счетчик последовательного переноса
- **тракт последовательного переноса** - счетчик последовательного переноса
- **тракт параллельного переноса** – счетчик параллельного переноса.

## СЧЁТЧИК С НЕПОСРЕДСТВЕННЫМИ СВЯЗЯМИ

R - вход общего сброса, Q1, Q2 и Q3 – выходы счетчика, CR – выход переноса единицы.

Суммирующий вход счетчика обозначается +1, вычитающий -1. Это счетные входы. У асинхронных счетчиков эти входы помечены специальными символами: или, указывающими полярность перепада входного сигнала: 1/0 или 0/1, при которой происходит переключение триггеров счетчика происходит переключение триггеров счетчика



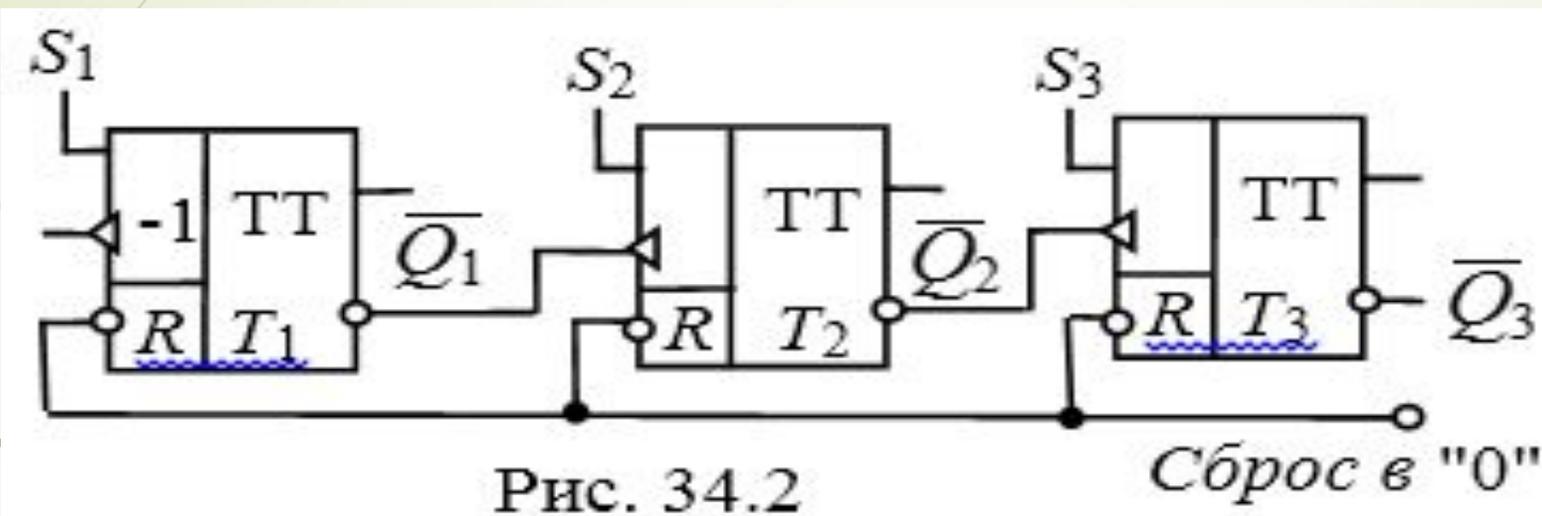
Т а б л и ц а 34.1

Номер входного импульса	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	CR
0	0	0	0	
1	0	0	1	
2	0	1	0	
3	0	1	1	
4	1	0	0	
5	1	0	1	
6	1	1	0	
7	1	1	1	
				1
0	0	0	0	

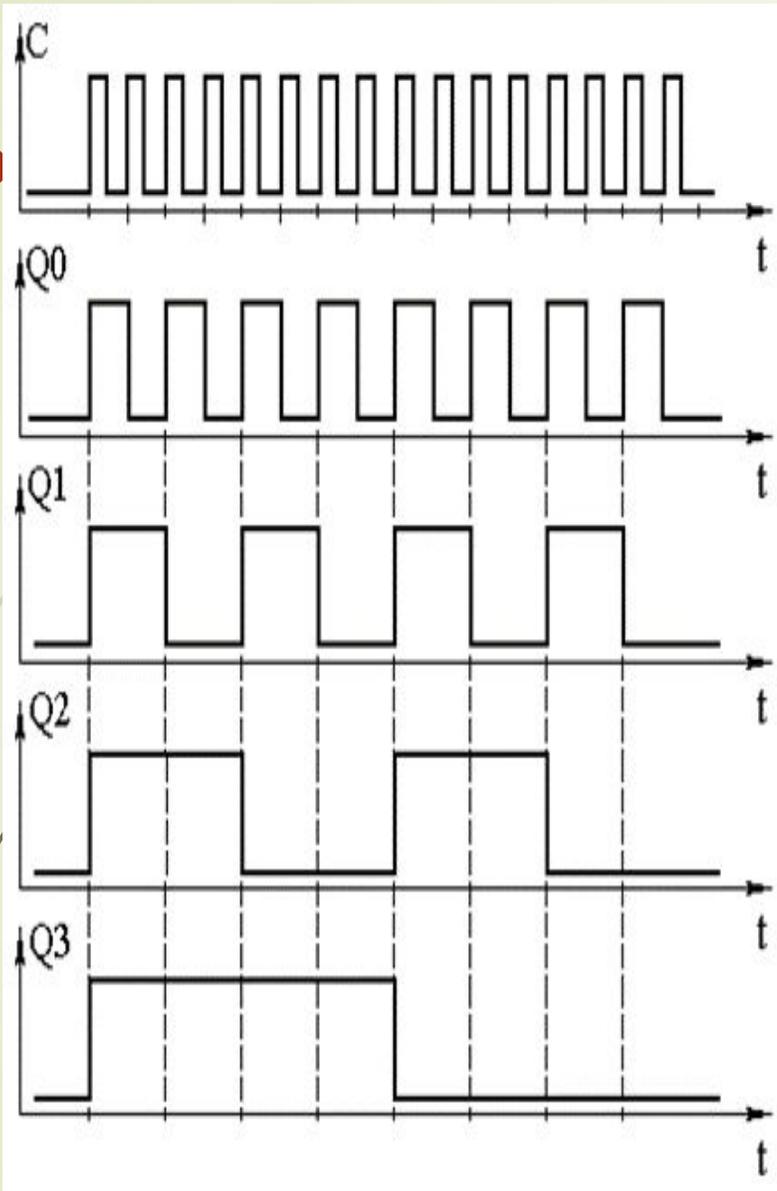
Для переключения триггеров в счетчиках используют следующие связи: непосредственную, тракт последовательного переноса, тракт параллельного переноса. Первый триггер счетчика T<sub>1</sub> образует младший разряд. Он пересчитывает входные импульсы по модулю 2, а состояние его выхода воспринимается следующим T<sub>2</sub> триггером как входные сигналы и снова пересчитываются на 2 и т. д.



Схема *вычитающего* счётчика приведена на рис. 34.2, в которой по входам  $S$  в разряды счётчика заносят двоичное число, из которого нужно вычесть число, представляемое количеством входных импульсов. Пусть, например, в счётчик (рис. 34.2) занесено число  $5_{10} = 101_2$ . Первым входным импульсом триггер  $T_1$  переключится из 1 в 0 (по основному выходу); при этом на инверсном выходе  $\overline{Q}_1$  возникает перепад 0/1, которым триггер  $T_2$  переключиться не может; в счётчике останется число  $100_2 = 4_{10}$ .



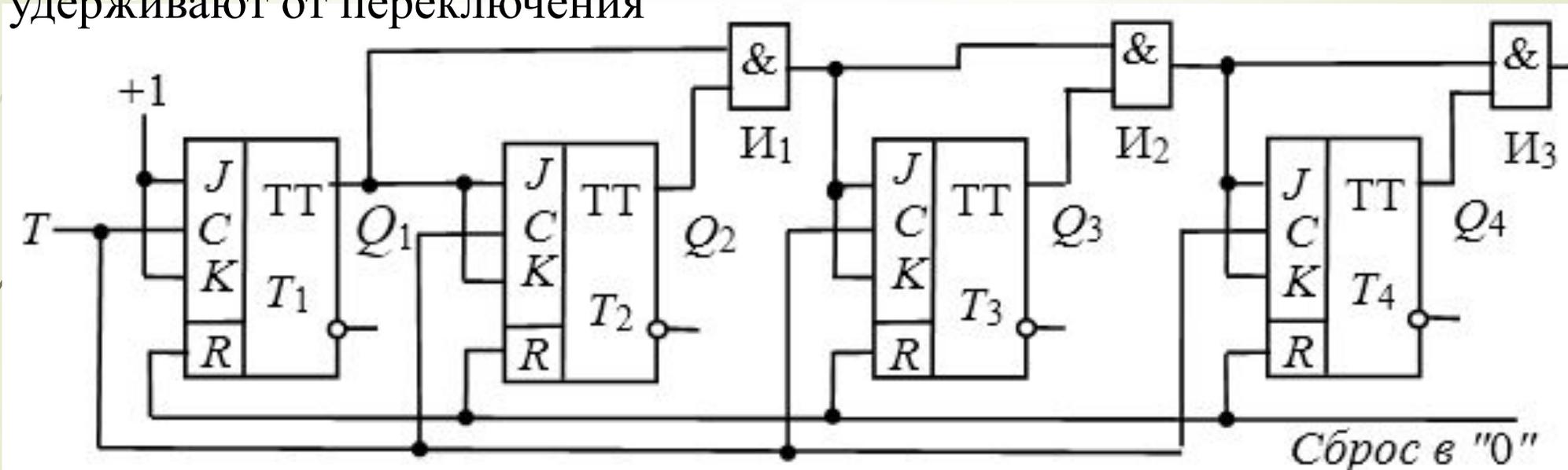
Второй входной импульс устанавливает триггер  $T_1$  в состояние 1, на выходе  $\overline{Q}_1$  появляется перепад 1/0, который переключает  $T_2$  в состояние 1, а формирующийся при этом на  $\overline{Q}_2$  перепад 1/0 переключает  $T_3$  в состояние 0. В счётчике остается число  $011_2 = 3_{10}$ . Аналогично можно рассмотреть действие последующих входных импульсов.



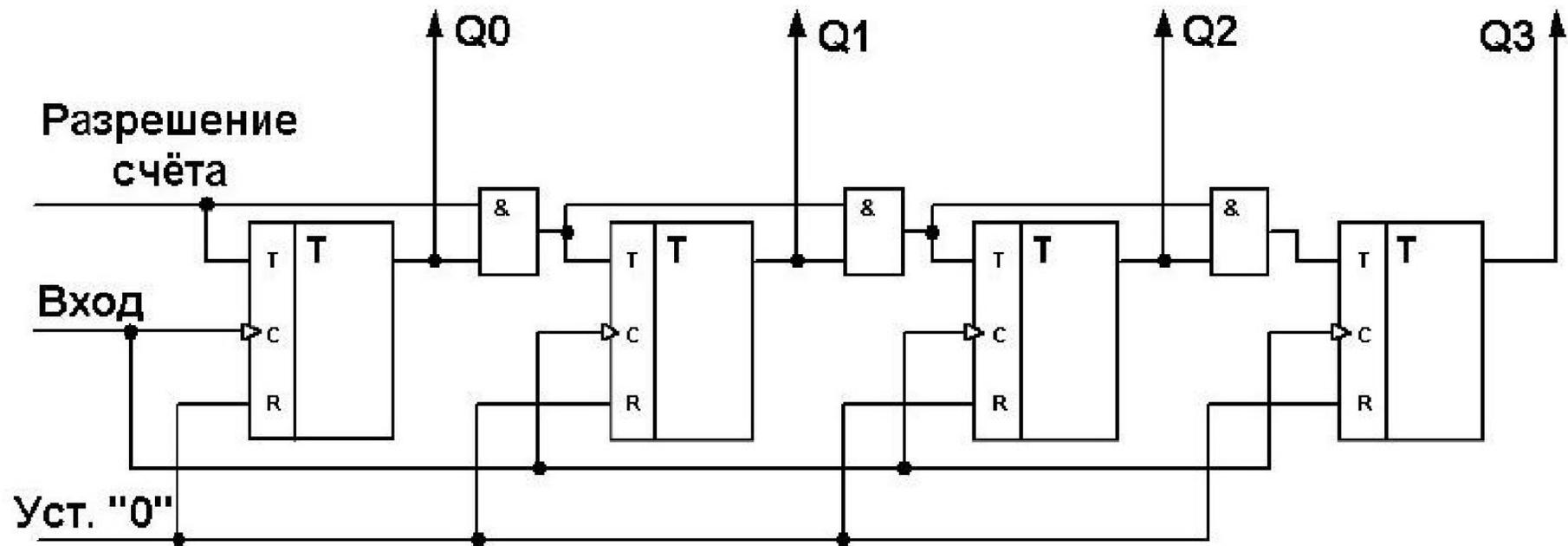
В счётчике с непосредственной связью переключение триггеров, вызванное срезом входного сигнала, происходит один за другим, *последовательно*, и задержка распространения *n*-разрядного счётчика, оцениваемая задержкой самого худшего случая – сменой всех 1 на все 0, – в *n* раз больше задержки одного *T*-триггера. Если разрядов много, то большая задержка может оказаться серьёзным недостатком такого счётчика. Из-за невозможности выполнить смену состояния всего счётчика в единый момент времени, счётчики с непосредственной связью бывают только *асинхронными*, т. е. сигналом, переключающим их, является сам входной сигнал.

# СУММИРУЮЩИЙ СИНХРОННЫЙ СЧЁТЧИК

В *синхронном* счётчике переключающиеся разряды переходят в новое состояние одновременно (синхронно). Для того чтобы на входы всех разрядов каждый счётный импульс поступал одновременно, а переключение разрядов происходило в нужной последовательности, в схему добавляют логические цепи, которые обеспечивают переключение одних разрядов, а другие удерживают от переключения

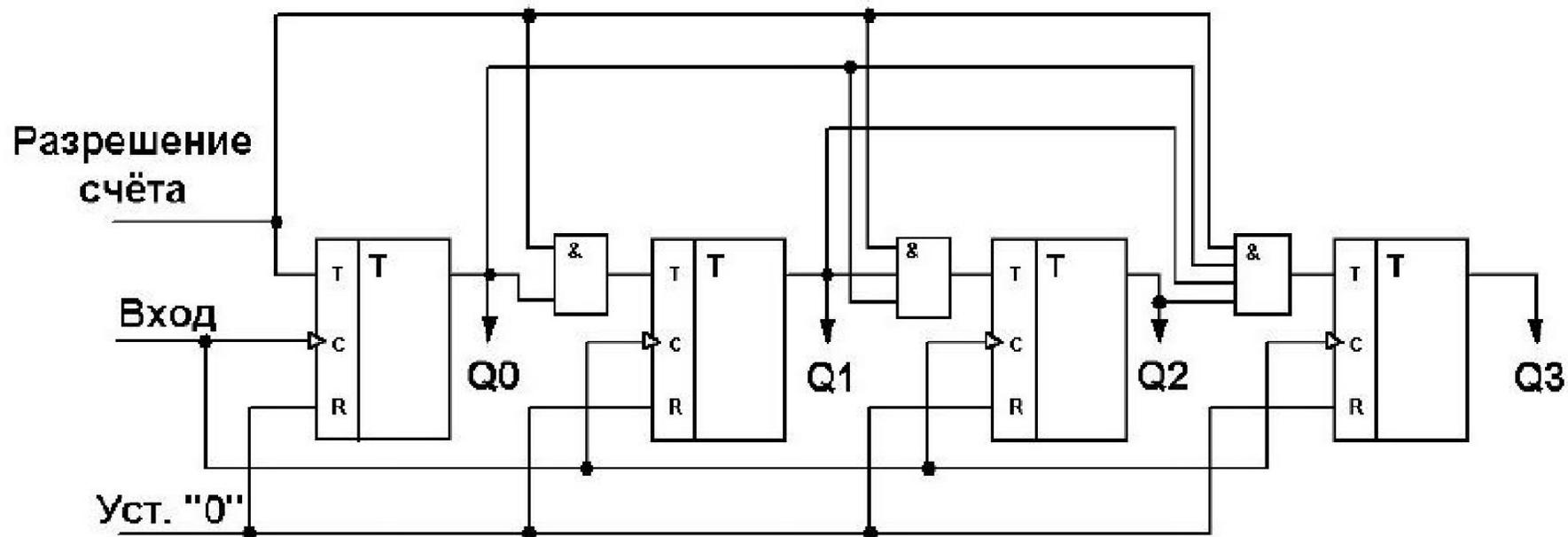


В схеме четырехразрядного синхронного счётчика на  $JK$ -триггерах на тактовые входы  $C$  всех триггеров счётные импульсы поступают одновременно с входа  $T$ . Информационные входы  $J$  и  $K$  каждого триггера объединены. Триггер  $T_1$  переключается каждым счётным импульсом, так как на его входы  $J$  и  $K$  постоянно подаётся 1. Остальные триггеры переключаются счётными импульсами при следующих условиях  $T_2$  – при  $Q_1 = 1$ ;  $T_3$  – при  $Q_1 = 1$  и  $Q_2 = 1$ ;  $T_4$  – при  $Q_1 = 1$ ,  $Q_2 = 1$  и  $Q_3 = 1$ .



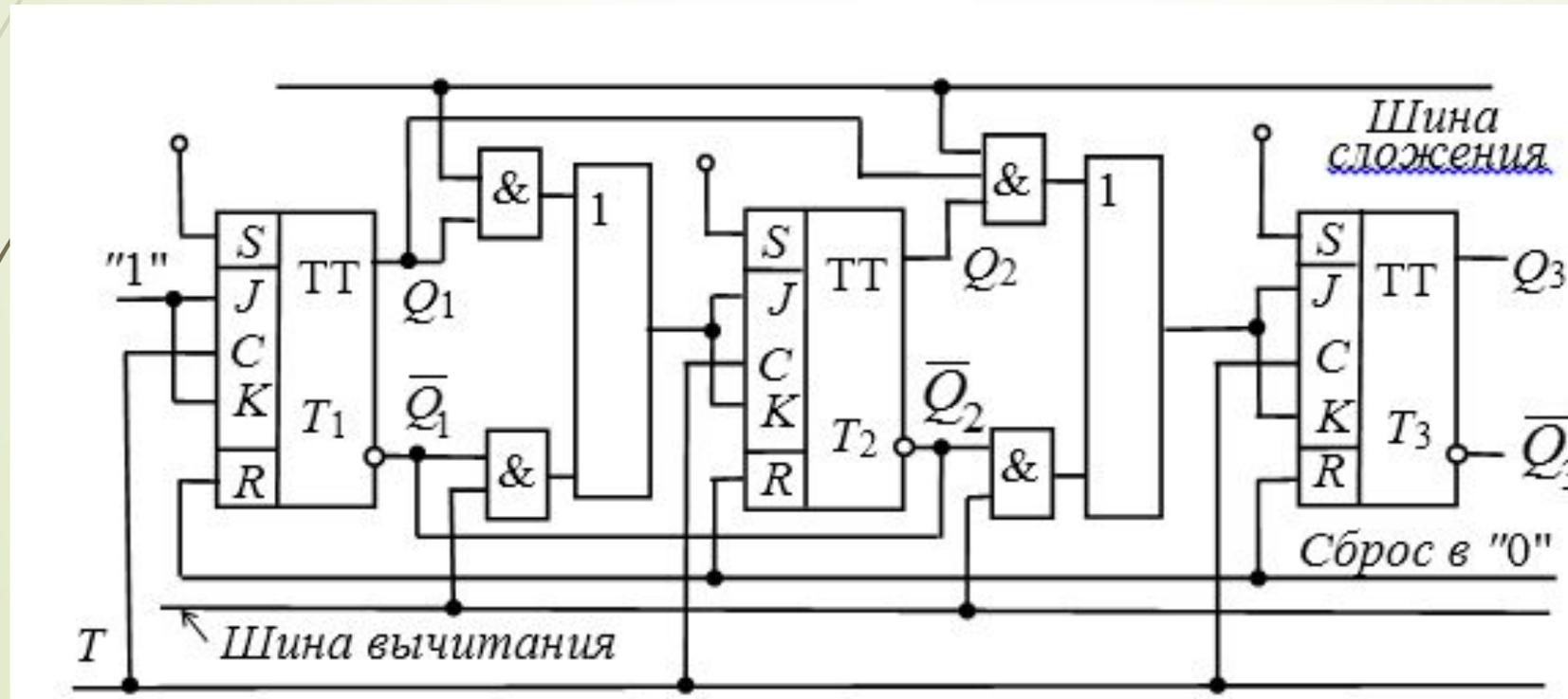
*Синхронный счетчик с последовательным переносом*

*Синхронные счетчики с параллельным переносом*



Реверсивный счётчик, фрагмент которого изображен на рис. 34.4, работает как на сложение, так и на вычитание. Для перехода от сложения к вычитанию и обратно изменяют подключение входа последующего триггера к выходам предыдущего.

На объединённые входы  $J$  и  $K$  каждого триггера подаётся через дизъюнкторы конъюнкция сигналов с выходов предыдущих триггеров: основные выходы предыдущих триггеров присоединяются через конъюнкторы верхнего ряда (при сложении), а инверсные выходы – через конъюнкторы нижнего ряда (при вычитании). При сложении подают 1 на шину сложения, которой вводятся в действие конъюнкторы верхнего ряда; при этом на шине вычитания присутствует 0, вследствие чего конъюнкторы нижнего ряда выключены. Вычитание осуществляется при подаче 1 на шину вычитания и 0 на шину сложения. Счетные импульсы поступают на вход  $T$ .



## Двоично-кодированные счетчики

Наибольшее распространение среди недвоичных счётчиков, у которых коэффициент пересчёта  $K_{сч} < 2^n$ , имеют десятичные счётчики, у которых  $K_{сч} = 10$ . При проектировании недвоичного счётчика вначале определяют количество его разрядов  $n$  так, чтобы  $2^n$  было бóльшим ближайшим к  $K_{сч}$  числом. Затем тем или иным способом (например, принудительной установкой некоторых разрядов счётчика в 1) исключают избыточные состояния счётчика, число которых равно  $2^n - K_{сч}$ .

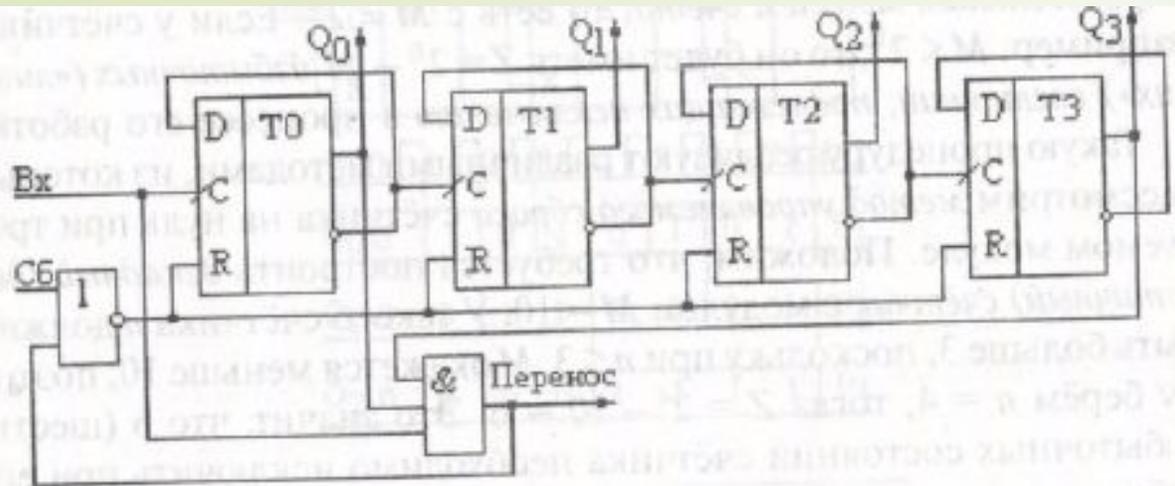
Так, для получения  $K_{сч} = 10$  одноразрядный счётчик должен содержать четыре триггера, а избыточные состояния  $2^n - K_{сч} = 16 - 10 = 6$  исключают.

При проектировании десятичного счётчика чаще используют двоично-десятичное кодирование чисел. В этой системе, например, число 375 записывается как 0011 0111 0101, где сохранены позиции десятичных разрядов:  $0011_2 = 3_{10}$ ,  $0111_2 = 7_{10}$ ,  $0101_2 = 5_{10}$ .

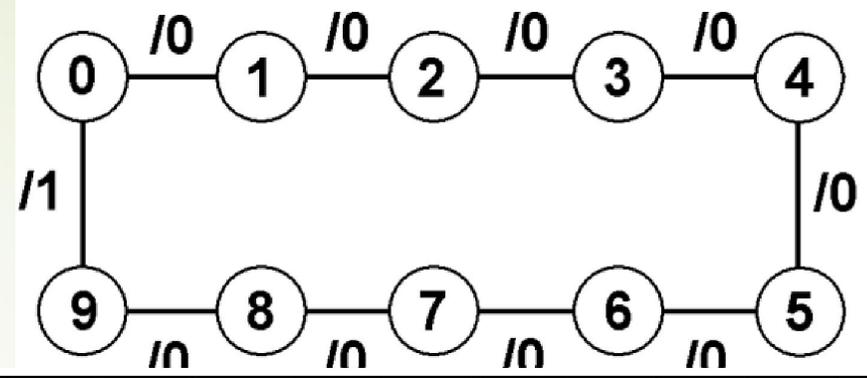
При построении таких счетчиков получили распространение в основном два метода: метод исключения лишних состояний; метод управляемого сброса.

Десятичный счётчик должен состоять из последовательно соединенных декад, информация о каждом из девяти импульсов накапливается в декаде, а десятым импульсом она обнуляется, и единица переносится в следующую декаду. Каждая декада работает в натуральном двоичном коде с весами двоичных разрядов, начиная со старшего, соответственно равными 8, 4, 2, 1, т.е. декада работает в коде 8-4-2-1. Если к выводам декад подключить индикаторы, то они будут показывать записанные числа в декадах в привычном десятичном

# Двоично-десятичные (декадные) счётчики

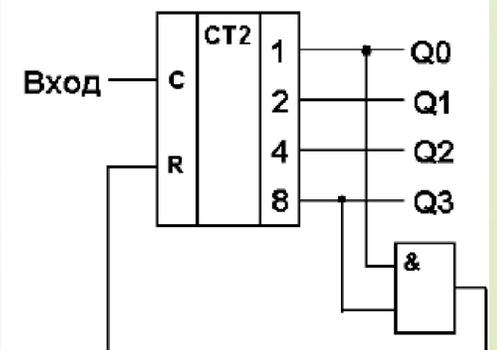


T	0	1	2	3	4	5	6	7	8	9	10	11	t
$Q_n$	0	1	0	1	0	1	0	1	0	1	0	1	t
$\bar{Q}_0^*$	1	0	1	0	1	0	1	0	1	0	0	0	t
$Q_1$	0	0	1	1	0	0	1	1	0	0	0	0	t
$\bar{Q}_1^*$	1	1	0	0	1	1	0	0	1	1	0	1	t
$Q_2$	0	0	0	0	1	1	1	1	0	0	0	0	t
$\bar{Q}_2^*$	1	1	1	1	0	0	0	0	1	1	0	1	t
$Q_3$	0	0	0	0	0	0	0	0	1	1	0	0	t
$\bar{Q}_3^*$	1	1	1	1	1	1	1	1	0	0	0	1	t



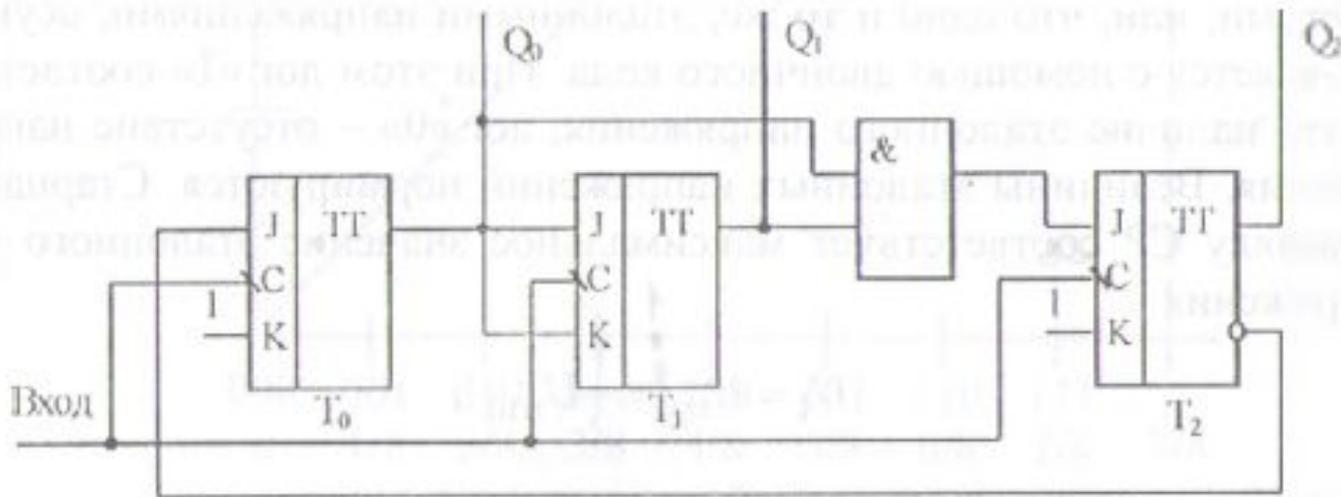
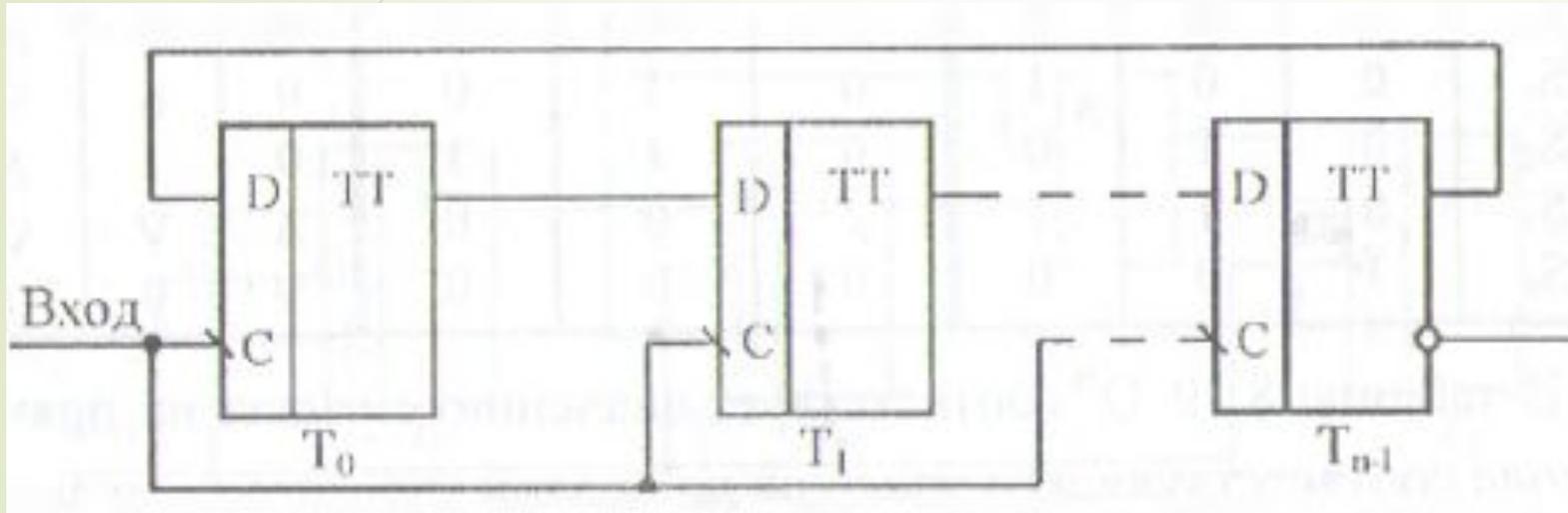
i	$Q_3$	$Q_2$	$Q_1$	$Q_0$	$Q_3^+$	$Q_2^+$	$Q_1^+$	$Q_0^+$	P4	$T_3$	$T_2$	$T_1$	$T_0$
0	0	0	0	0	0	0	0	1	0	0	0	0	1
1	0	0	0	1	0	0	1	0	0	0	0	1	1
2	0	0	1	0	0	0	1	1	0	0	0	0	1
3	0	0	1	1	0	1	0	0	0	0	1	1	1
4	0	1	0	0	0	1	0	1	0	0	0	0	1
5	0	1	0	1	0	1	1	0	0	0	0	1	1
6	0	1	1	0	0	1	1	1	0	0	0	0	1
7	0	1	1	1	1	0	0	0	0	1	1	1	1
8	1	0	0	0	1	0	0	1	0	0	0	0	1
9	1	0	0	1	0	0	0	0	1	1	0	0	1

$$T_0 = 1; \quad T_1 = \bar{Q}_3 Q_0; \quad T_2 = \bar{Q}_3 Q_1 Q_0; \quad T_3 = Q_3 Q_2 Q_1 Q_0 + Q_3 \bar{Q}_2 \bar{Q}_1 \bar{Q}_0$$



## Недвоичные счетчики.

Недвоичные счетчики имеют коэффициент пересчёта не кратный 2. Наиболее просто недвоичный счетчик можно реализовать па базе сдвигового регистра, если выход последнего  $T_{n-1}$  триггера соединить с D входом первого  $T_0$  триггера. В результате имеет место замкнутая **КОЛЬЦЕВАЯ** система



счетчик с  $K_{сч} = 5$



# Счётчики в коде Грея

$$q_i = Q_i \oplus Q_{i+1} \text{ (для } i = 0, 1, \dots, n-2),$$

$$q_{i-1} = Q_{i-1}.$$

Двоичный код				Код Грея			
Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	q <sub>3</sub>	q <sub>2</sub>	q <sub>1</sub>	q <sub>0</sub>
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

q <sub>3</sub>	q <sub>2</sub>	q <sub>1</sub>	q <sub>0</sub>	q <sub>3</sub> <sup>+</sup>	q <sub>2</sub> <sup>+</sup>	q <sub>1</sub> <sup>+</sup>	q <sub>0</sub> <sup>+</sup>	P4	T <sub>3</sub>	T <sub>2</sub>	T <sub>1</sub>	T <sub>0</sub>
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	1	1	0	0	0	1	0
0	0	1	1	0	0	1	0	0	0	0	0	1
0	0	1	0	0	1	1	0	0	0	1	0	0
0	1	1	0	0	1	1	1	0	0	0	0	1
0	1	1	1	0	1	0	1	0	0	0	1	0
0	1	0	1	0	1	0	0	0	0	0	0	1
0	1	0	0	1	1	0	0	0	1	0	0	0
1	1	0	0	1	1	0	1	0	0	0	0	1
1	1	0	1	1	1	1	1	0	0	0	1	0
1	1	1	1	1	1	1	0	0	0	0	0	1
1	1	1	0	1	0	1	0	0	0	1	0	0
1	0	1	0	1	0	1	1	0	0	0	0	1
1	0	1	1	1	0	0	1	0	0	0	1	0
1	0	0	1	1	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	1	1	0	0	0

$$T_0 = \overline{Q_3} \overline{Q_2} \overline{Q_1} \overline{Q_0} + \overline{Q_3} \overline{Q_2} \overline{Q_1} Q_0 + \overline{Q_3} \overline{Q_2} Q_1 \overline{Q_0} + \overline{Q_3} \overline{Q_2} Q_1 Q_0 + \overline{Q_3} Q_2 \overline{Q_1} \overline{Q_0} + \overline{Q_3} Q_2 \overline{Q_1} Q_0 + \overline{Q_3} Q_2 Q_1 \overline{Q_0} + \overline{Q_3} Q_2 Q_1 Q_0 + Q_3 \overline{Q_2} \overline{Q_1} \overline{Q_0} + Q_3 \overline{Q_2} \overline{Q_1} Q_0 + Q_3 \overline{Q_2} Q_1 \overline{Q_0} + Q_3 \overline{Q_2} Q_1 Q_0 = \overline{Q_3} \overline{Q_2} (\overline{Q_1} \overline{Q_0} + \overline{Q_1} Q_0) + Q_3 \overline{Q_2} (Q_1 \overline{Q_0} + Q_1 Q_0) + Q_3 Q_2 (\overline{Q_1} \overline{Q_0} + \overline{Q_1} Q_0) + Q_3 Q_2 (Q_1 \overline{Q_0} + Q_1 Q_0) = \overline{Q_3} \overline{Q_2} (\overline{Q_0} \oplus Q_1) + \overline{Q_3} \overline{Q_2} (Q_0 \oplus Q_1) + Q_3 \overline{Q_2} (\overline{Q_0} \oplus Q_1) + Q_3 \overline{Q_2} (Q_0 \oplus Q_1) = (\overline{Q_0} \oplus Q_1) (\overline{Q_3} \overline{Q_2} + Q_3 \overline{Q_2}) + (Q_0 \oplus Q_1) (\overline{Q_3} \overline{Q_2} + Q_3 \overline{Q_2}) = (\overline{Q_0} \oplus Q_1) (\overline{Q_2} \oplus Q_3) + (Q_0 \oplus Q_1) (Q_2 \oplus Q_3) = \overline{Q_0} \oplus Q_1 \oplus Q_2 \oplus Q_3.$$

$$T_1 = \overline{Q_3} \overline{Q_2} \overline{Q_1} \overline{Q_0} + \overline{Q_3} \overline{Q_2} \overline{Q_1} Q_0 + \overline{Q_3} \overline{Q_2} Q_1 \overline{Q_0} + \overline{Q_3} \overline{Q_2} Q_1 Q_0 = \overline{Q_3} \overline{Q_2} (\overline{Q_1} \overline{Q_0} + \overline{Q_1} Q_0) + \overline{Q_3} \overline{Q_2} (Q_1 \overline{Q_0} + Q_1 Q_0) = \overline{Q_3} \overline{Q_2} (\overline{Q_0} \oplus Q_1) + \overline{Q_3} \overline{Q_2} (Q_0 \oplus Q_1) = \overline{Q_3} \overline{Q_2} (\overline{Q_0} \oplus Q_1) + \overline{Q_3} \overline{Q_2} (Q_0 \oplus Q_1) = \overline{Q_3} \overline{Q_2} (\overline{Q_0} \oplus Q_1 \oplus Q_0 \oplus Q_1) = \overline{Q_3} \overline{Q_2} (\overline{Q_0} \oplus Q_0 \oplus Q_1 \oplus Q_1) = \overline{Q_3} \overline{Q_2} (Q_3 \oplus Q_2).$$

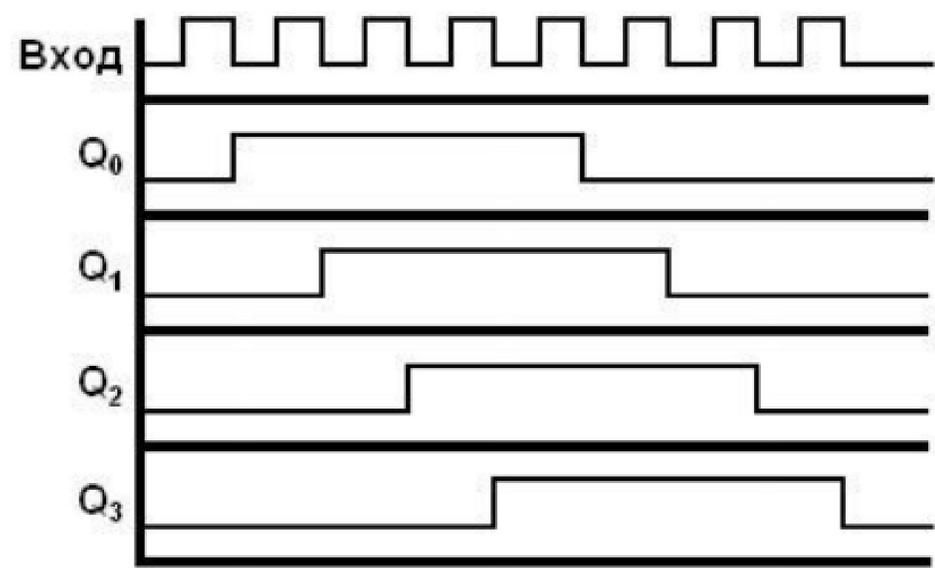
$$T_2 = \overline{Q_3} \overline{Q_2} \overline{Q_1} \overline{Q_0} + \overline{Q_3} \overline{Q_2} \overline{Q_1} Q_0 = \overline{Q_3} \overline{Q_2} (\overline{Q_1} \overline{Q_0} + \overline{Q_1} Q_0) = \overline{Q_3} \overline{Q_2} (\overline{Q_0} \oplus Q_1).$$

$$T_3 = \overline{Q_3} \overline{Q_2} \overline{Q_1} \overline{Q_0} + \overline{Q_3} \overline{Q_2} \overline{Q_1} Q_0 = \overline{Q_3} \overline{Q_2} (\overline{Q_1} \overline{Q_0} + \overline{Q_1} Q_0) = \overline{Q_3} \overline{Q_2} (\overline{Q_0} \oplus Q_1).$$

$$P4 = \overline{Q_3} \overline{Q_2} \overline{Q_1} \overline{Q_0}.$$

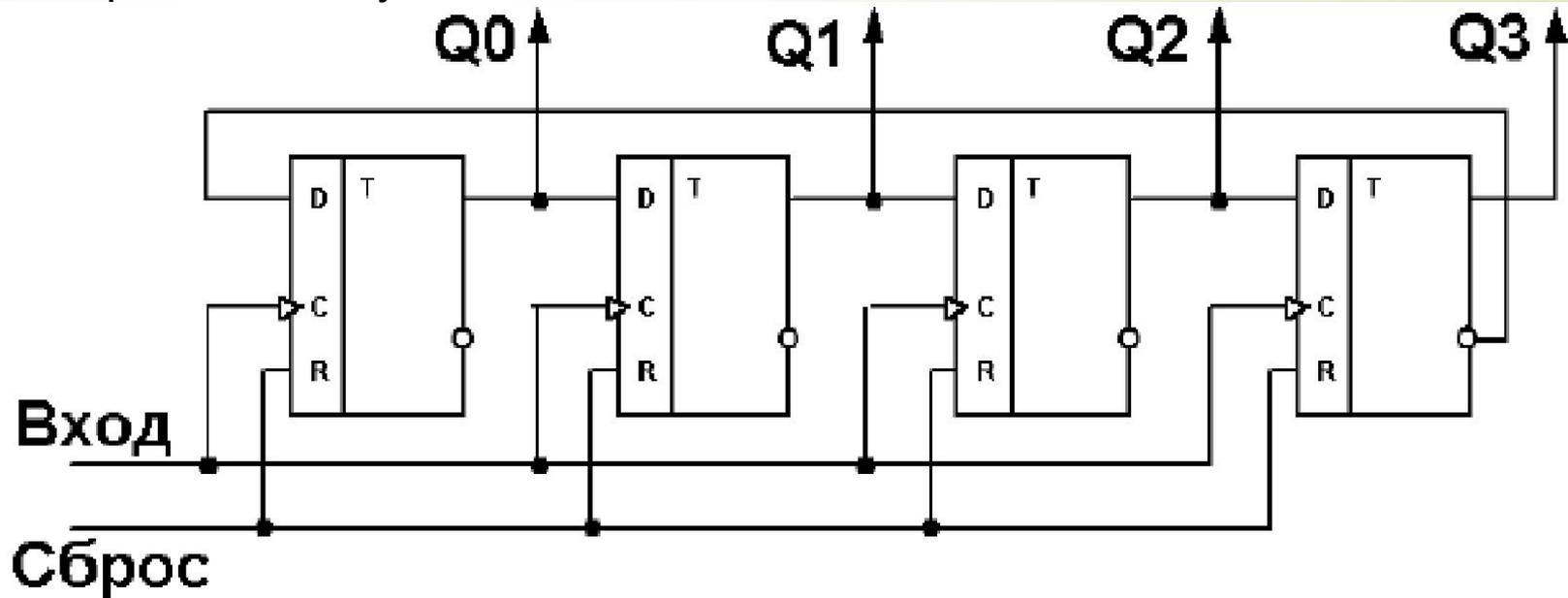
# Счётчик Джонсона

$Q_3$	$Q_2$	$Q_1$	$Q_0$	$Q_3^+$	$Q_2^+$	$Q_1^+$	$Q_0^+$
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	1
0	0	1	1	0	1	1	1
0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	0
1	1	1	0	1	1	0	0
1	1	0	0	1	0	0	0
1	0	0	0	0	0	0	0



первые четыре такта счётчик заполняется «единицами», а в следующие четыре такта - «нулями».

	Q1Q0			
	00	01	11	10
00	1	1	1	-
01	-	-	1	-
11	0	-	0	0
10	0	-	-	-



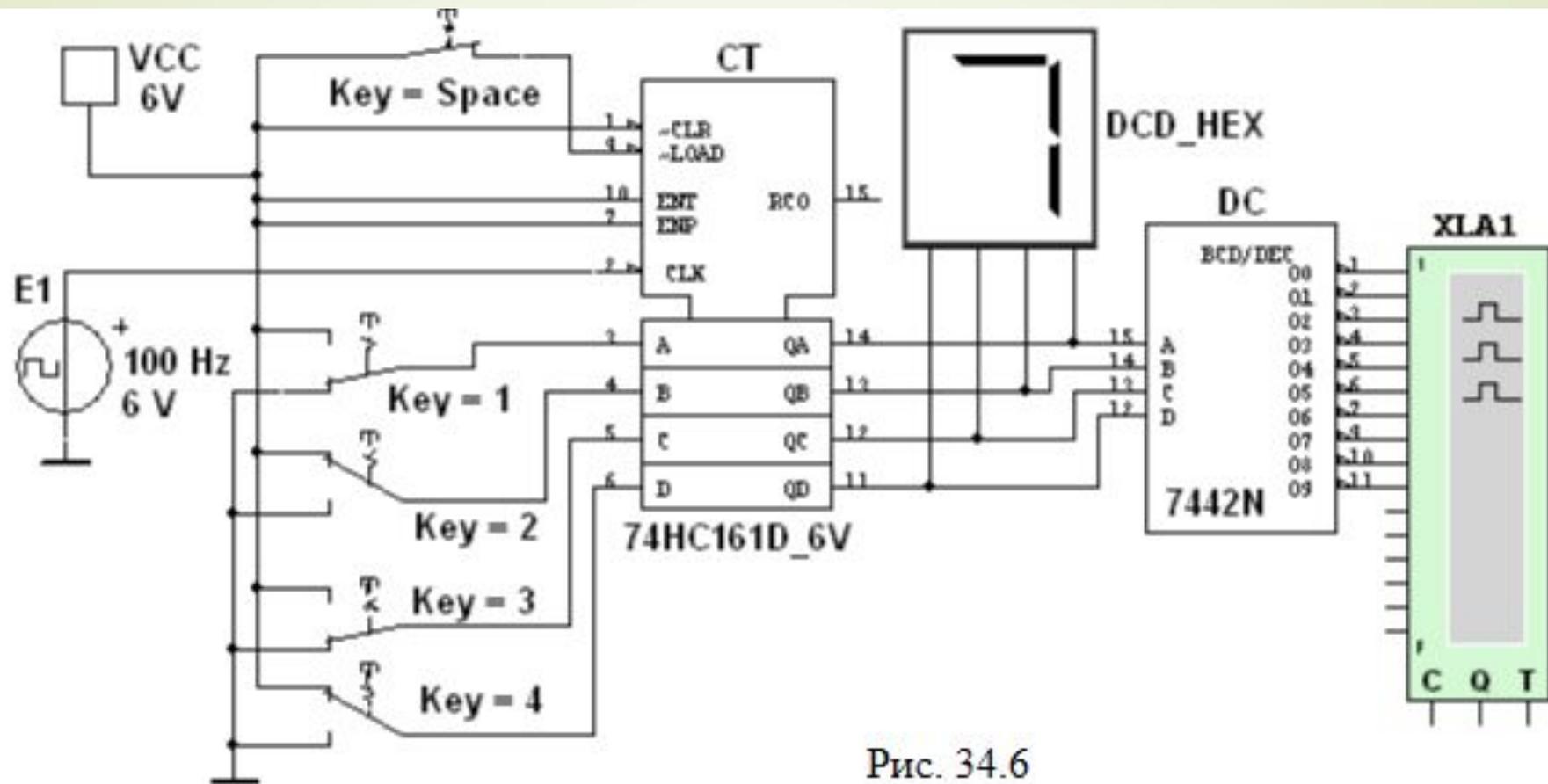
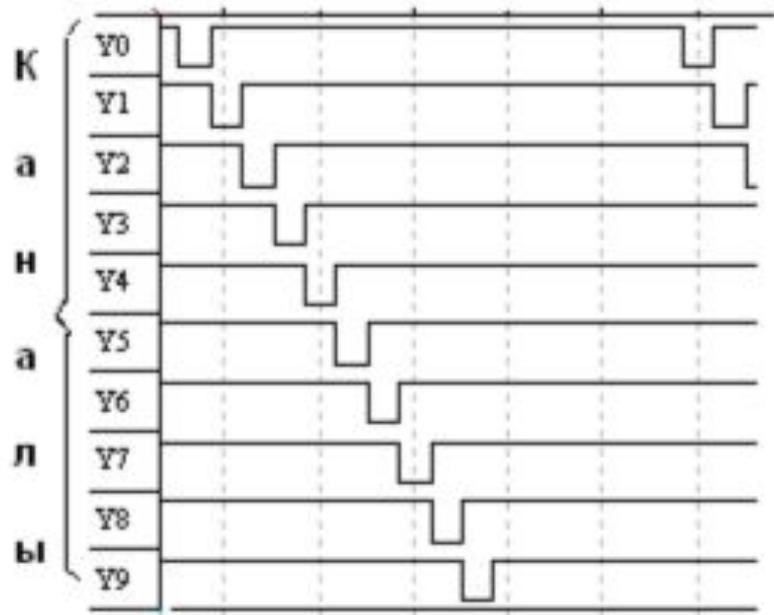
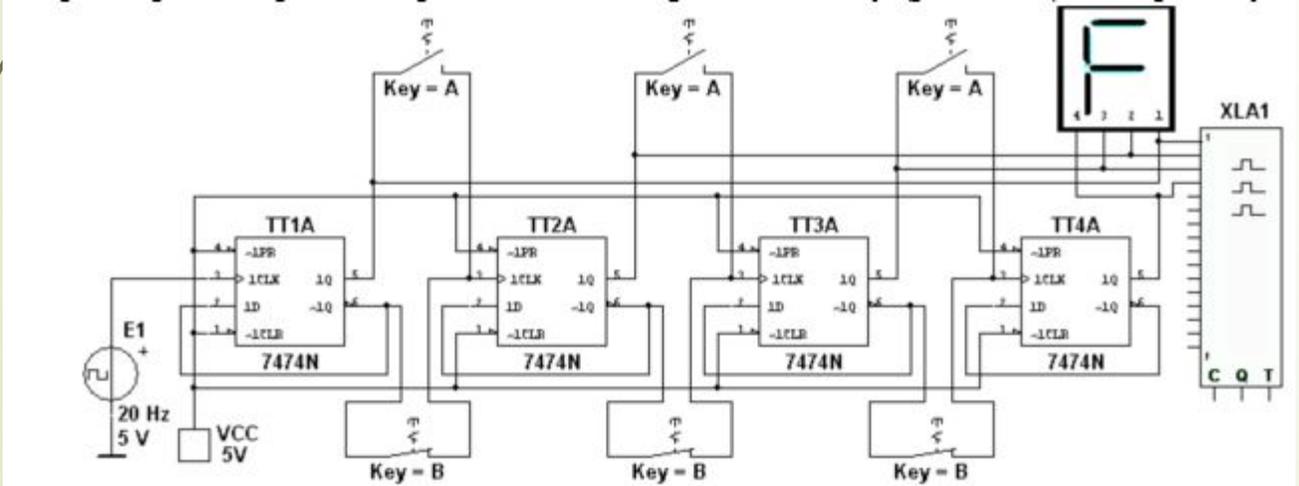
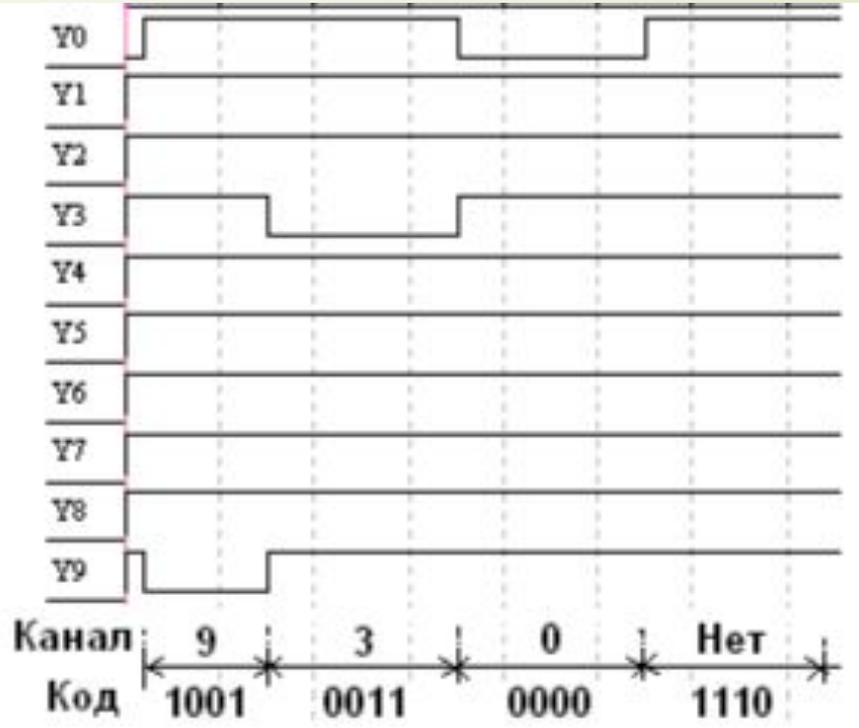


Рис. 34.6



a)



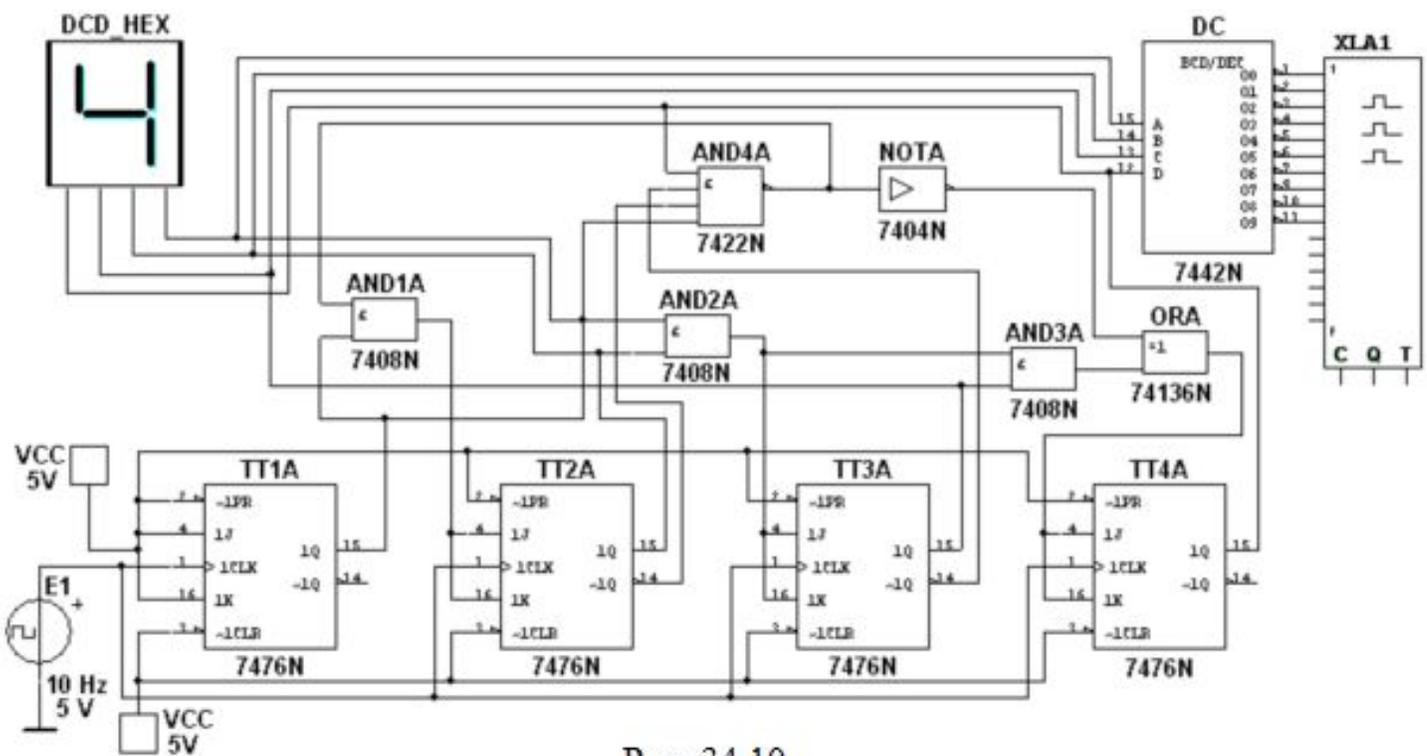
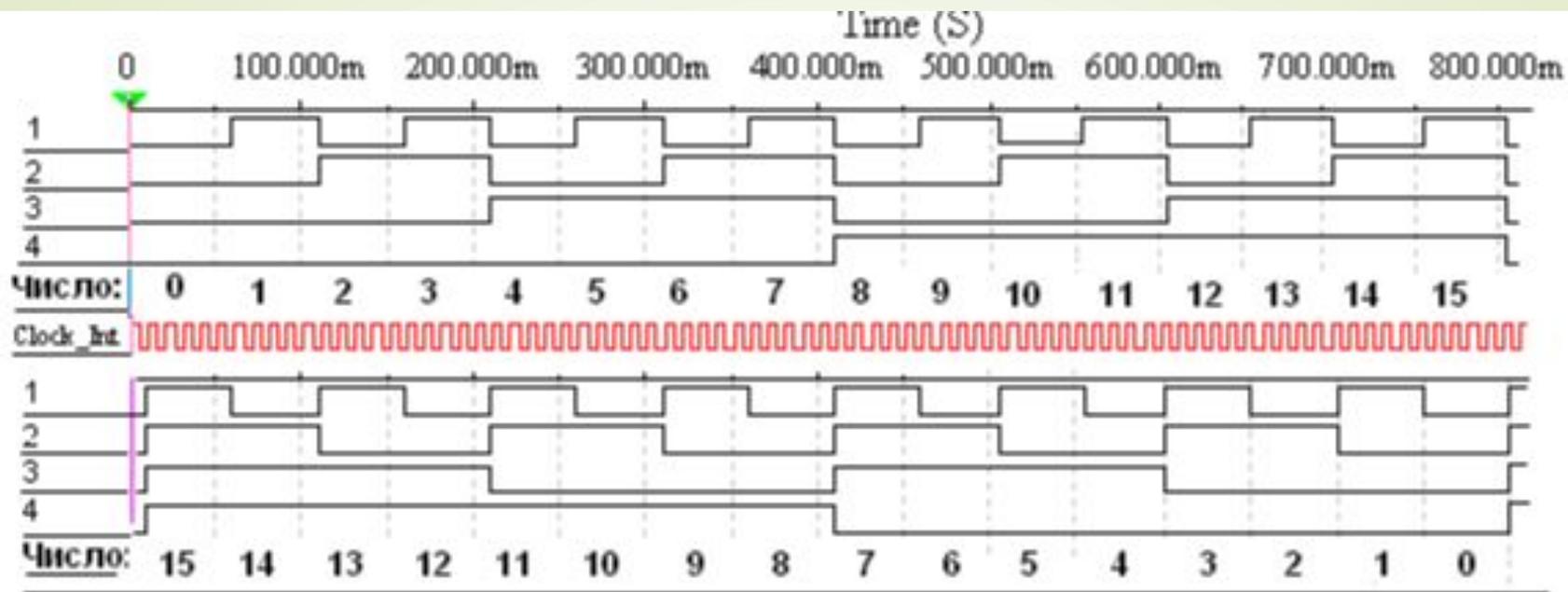
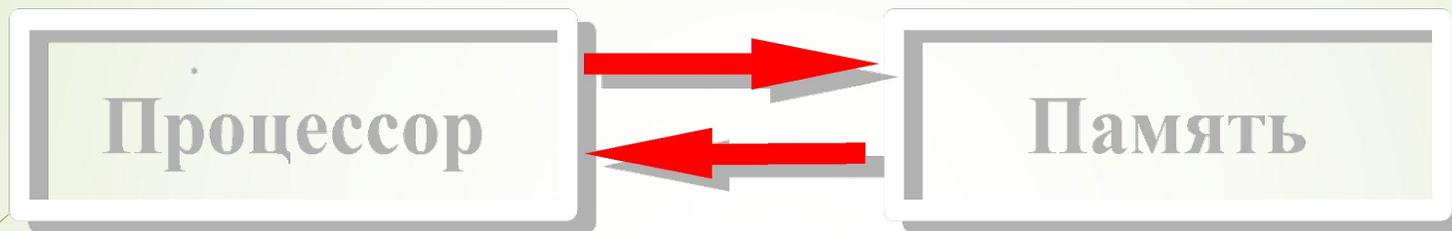


Рис. 34.10

# Архитектура памяти цифрового устройства

□ Любая сложная цифровая система (ЭВМ) с точки зрения ее логической структуры состоит как минимум из 2-х компонентов:

- блока вычислений (центрального процессора)
- блока памяти.



Для кратковременного хранения небольших объемов кодовых слов обычно используют регистры. При необходимости длительного хранения или хранения больших объемов информации применяют запоминающие устройства (ЗУ), выполненные на специализированных ИС. Применение ЗУ, использующих ИС, позволяет максимально упростить аппаратную часть электронных устройств.

**Запоминающим устройством (ЗУ) наз. отдельное устройство вычислительного комплекса или процессора, реализующее функцию памяти: запись, хранение и выдача цифровой информации в процессе её обработки.**

**Память - memory** - способность устройства обеспечивать хранение информации. Хранение осуществляется в запоминающих устройствах (ЗУ).

# Классификация ЗУ

Все существующие типы ЗУ делят на два класса: оперативные запоминающие устройства (ОЗУ) и постоянные запоминающие устройства (ПЗУ).

К ОЗУ относятся устройства, кратковременного хранения информации с возможностью ее быстрой записи и чтения. Отсюда и название – оперативное ЗУ. Выключение напряжения питания приводит к потере хранящейся информации.

ПЗУ предназначены для длительного хранения информации, которая не исчезает при снятии напряжения питания. Поэтому ОЗУ, обычно, используются для хранения обрабатываемых данных, а ПЗУ - для хранения алгоритмов работы устройства.



## ПАРАМЕТРЫ ЗУ :

- **Емкость ЗУ (M) -максимально возможный объем хранимой в нем информации.**

Единицей измерения количества информации является один *бит*. Это количество информации, соответствующее одному разряду двоичного кодового слова или одной логической константе. Численно бит может принимать значения лог. 0 или лог. 1. Обычно информация, равная одному биту, хранится в одном элементарном запоминающем элементе (ЭЗЭ). Поэтому емкость ЗУ принято определять в битах или количестве кодовых слов с указанием их разрядности. При этом 8-разрядное кодовое слово называют *байтом*.

Для определения больших объемов информации используют приставки кило и мега, означающие соответственно  $2^{10} = 1024$  бит = 1 Кбит и  $2^{20} = 1048576$  бит = 1 Мбит.

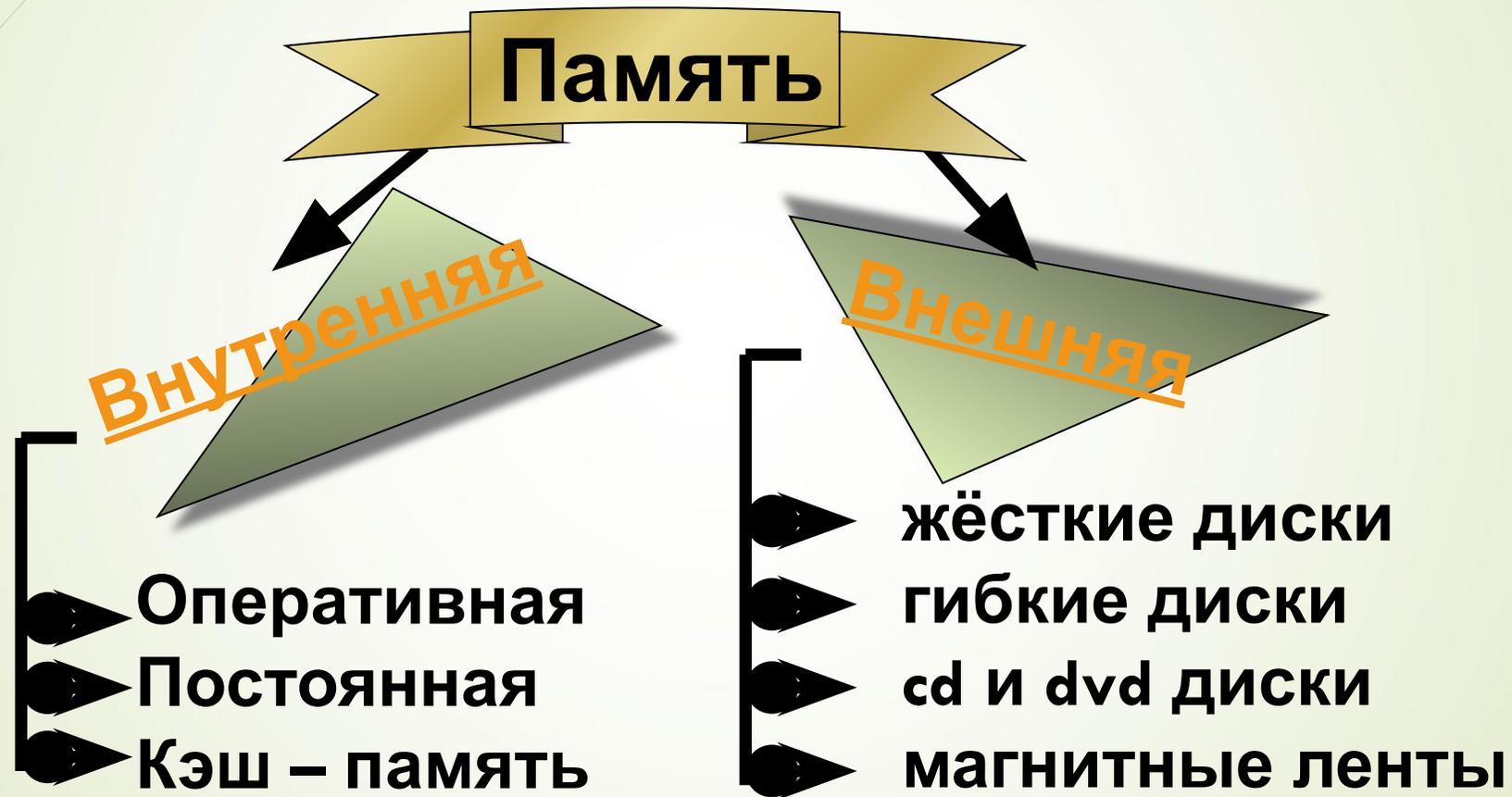
- **«Организация ЗУ»  $M = NL$**

**где N-число кодовых слов, хранимых в ЗУ , L- их длина (разрядность).**

Очевидно, что емкость ЗУ связана с параметрами, характеризующими его организацию, соотношением  $M = NL$ . Из сказанного следует, что при одном и том же объеме хранимой информации память может иметь различную организацию. Так, два ЗУ с организацией  $32 \times 8$  и  $256 \times 1$  имеют одинаковый объем памяти, равный 256 бит.

- **Время выборки  $t_A$  - временной интервал между подачей на вход памяти заданного сигнала и получением на выходе данных при условии, что все остальные сигналы поданы.**
- **Время цикла адреса в режиме записи - минимальное время совпадения сигналов на управляющих входах», необходимое для надежной записи в нее информации.**
- **Время цикла адреса для режима считывания.**
- **Энергонезависимость - способность ЗУ сохранять/не сохранять данные при**

По выполняемой функции ЗУ (память)  
можно классифицировать на:



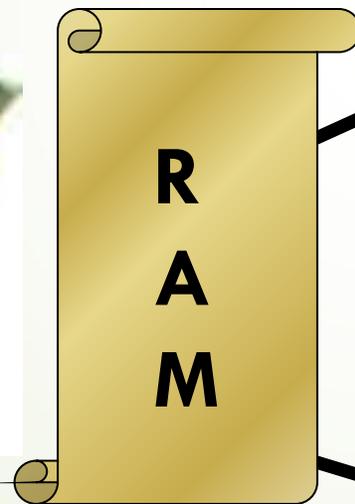
# Оперативно запоминающие устройства (ОЗУ) – статические и динамические

предназначены для хранения переменной информации и допускающая изменение своего содержимого в ходе выполнения процессором операций по обработке информации



Оперативная память представляет собой совокупность ячеек, разделенных на разряды для хранения в каждом бите информации. Все ячейки памяти пронумерованы, номер ячейки - это ее адрес. Он позволяет отличать ячейки друг от друга, обращаться к любой ячейке, чтобы записать в нее новую информацию или воспользоваться уже хранящейся в ячейке.

Поскольку в любой момент времени доступ может осуществляться к произвольно выбранной ячейке, то этот вид памяти также называют **памятью с произвольной выборкой - RAM** (Random Access Memory).



**статическая**  
(SRAM - Static RAM)

**динамическая**  
(DRAM - Dynamic RAM)

В статических ОЗУ записанная информация постоянно хранится в выделенном для нее месте и не разрушается при ее считывании. Разрушение информации возможно только при ее принудительном стирании или отключении напряжения источника питания.

В **статической памяти** элементы (ячейки) построены на различных вариантах триггеров - схем с двумя устойчивыми состояниями. После записи бита в такую ячейку она может пребывать в этом состоянии сколь угодно долго - необходимо только наличие питания. **Ячейки статической памяти имеют малое время срабатывания (единицы-десятки наносекунд), но микросхемы на их основе имеют низкую удельную плотность данных и высокое энергопотребление.**

**Статическая память используется в основном в качестве буферной.**

В динамических ОЗУ информация постоянно циркулирует в массиве, отведенном для ее хранения. При этом считывание информации сопровождается ее разрушением. Для сохранения информации ее необходимо перезаписать заново.

В **динамической памяти** ячейки построены на основе областей с накоплением зарядов, занимающих гораздо меньшую площадь, нежели триггеры, и практически не потребляющих энергии при хранении. При записи бита в такую ячейку в ней формируется электрический заряд, который сохраняется в течение нескольких миллисекунд; для постоянного сохранения заряда ячейки необходимо регенерировать – перезаписывать содержимое для восстановления зарядов. Ячейки динамической памяти имеют большее время срабатывания (десятки-сотни наносекунд), но большую удельную плотность (порядка десятков Мбит на корпус) и меньшее энергопотребление.

**Динамическая память используется в качестве основной.**



**SRAM(статическая) и**

**DRAM(динамическая)**

памяти называют также **асинхронными** - могут выполняться в произвольные моменты времени.



Существуют также **синхронные** виды памяти, получающие внешний синхросигнал, к импульсам которого жестко привязаны моменты подачи адресов и обмена данными; помимо экономии времени на охранных интервалах, они позволяют более полно использовать внутреннюю конвейеризацию и блочный доступ.

## FPM DRAM

(Fast Page Mode DRAM - динамическая память с быстрым страничным доступом) активно используется в последние несколько лет. Она позволяет ускорить блочные передачи и снизить накладные расходы на регенерацию памяти.



## EDO

(Extended Data Out - расширенное время удержания данных на выходе). Этот вид памяти позволяет примерно на 15 % по сравнению с FPM ускорить процесс считывания последовательных массивов данных. При случайной адресации такая память ничем не отличается от обычной.



## BEDO

(Burst EDO - EDO с блочным доступом) - память на основе EDO, работающая не одиночными, а пакетными циклами чтения/записи.

## PB SRAM

(Pipelined Burst SRAM - статическая память с блочным конвейерным доступом) - разновидность синхронных SRAM с внутренней конвейеризацией, за счет которой примерно вдвое повышается скорость обмена блоками данных.

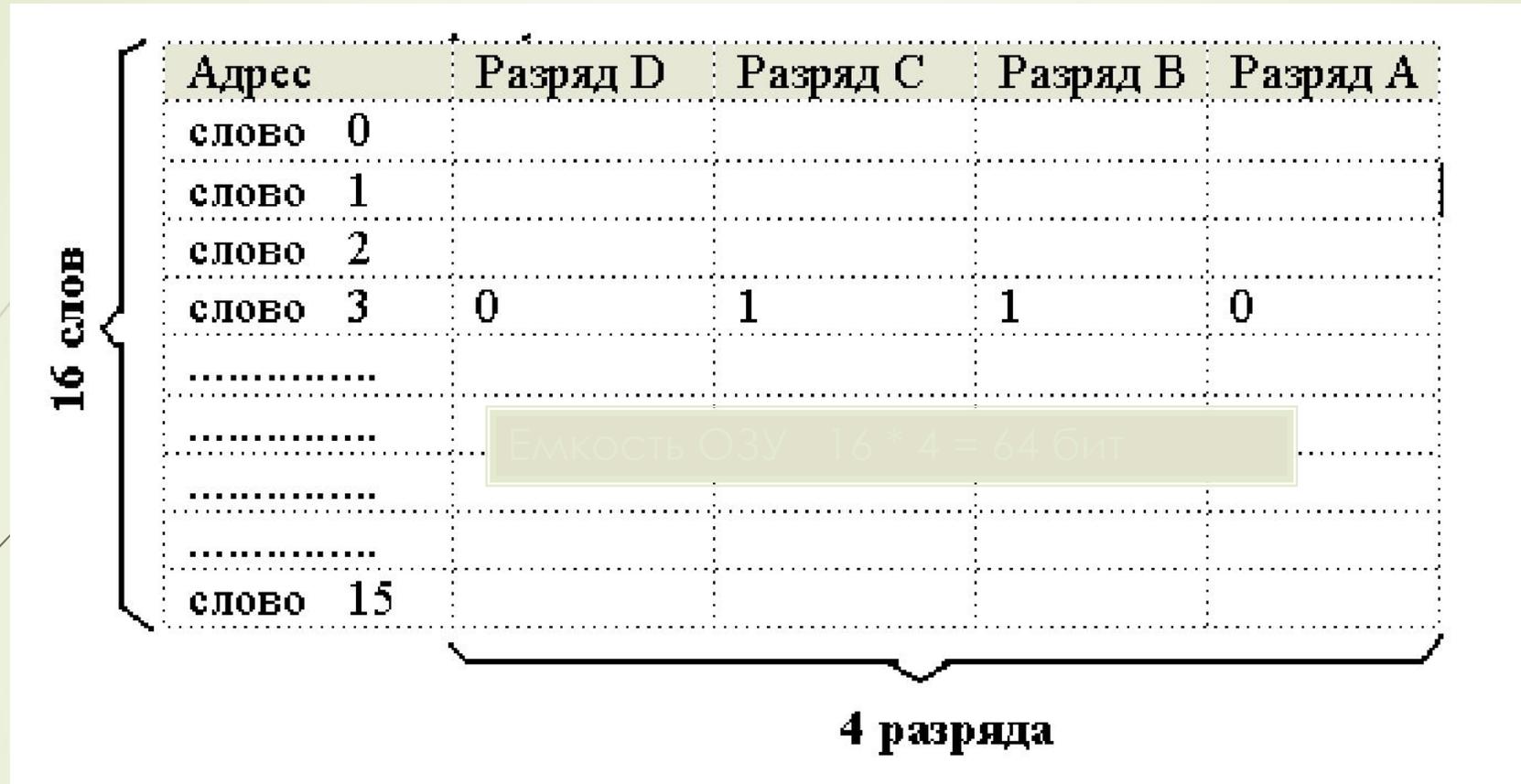


## SDRAM

(Synchronous DRAM - синхронная динамическая память) – память с синхронным доступом, работающая быстрее обычной асинхронной (FPM/EDO/BEDO). SDRAM также поддерживает блочный обмен. Основная выгода от использования SDRAM состоит в поддержке последовательного доступа в синхронном режиме, где не требуется дополнительных тактов ожидания. При случайном доступе SDRAM работает практически с той же скоростью, что и FPM/EDO.

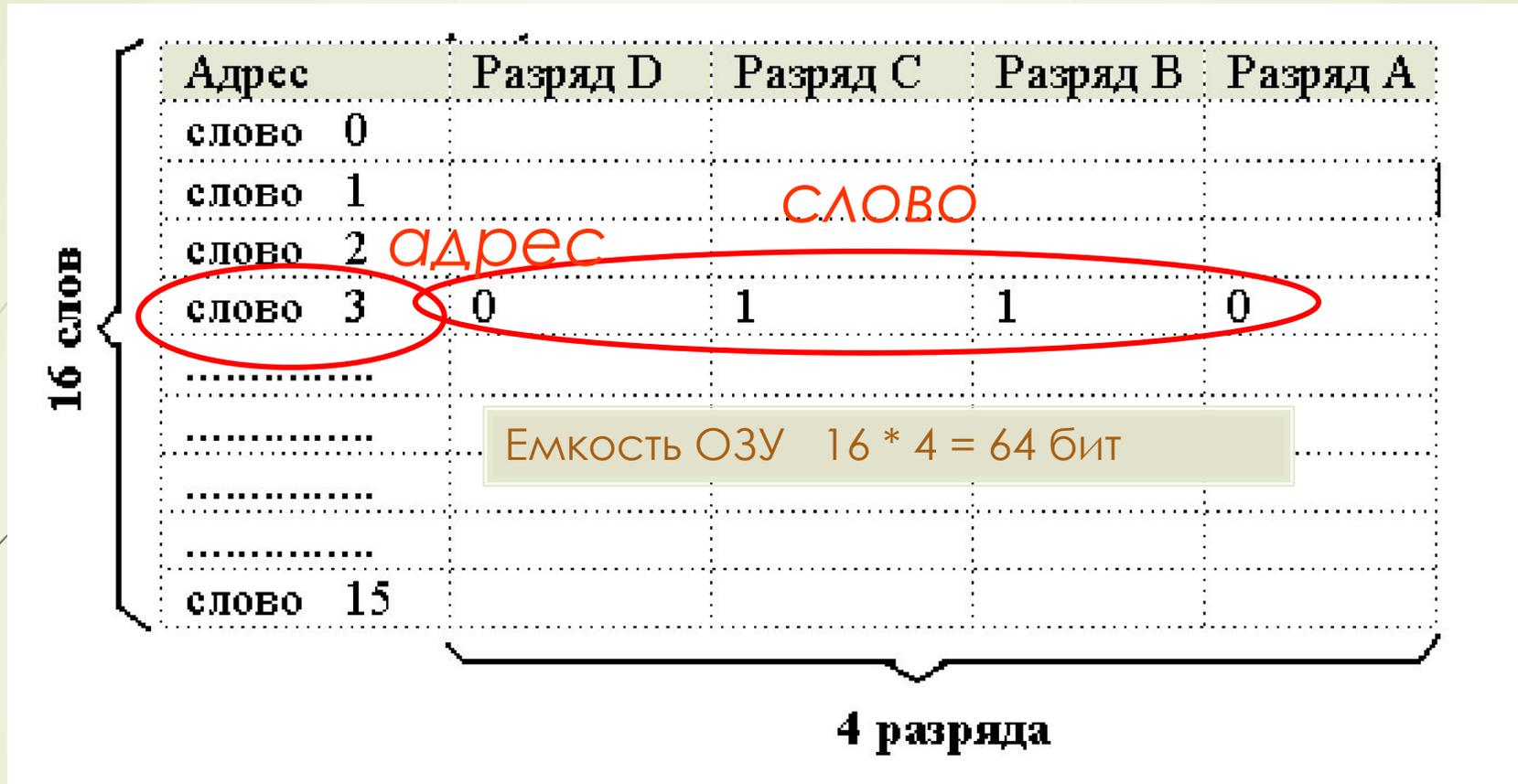
**После выключения компьютера  
оперативная память  
очищается.**

# Логическая организация ОЗУ



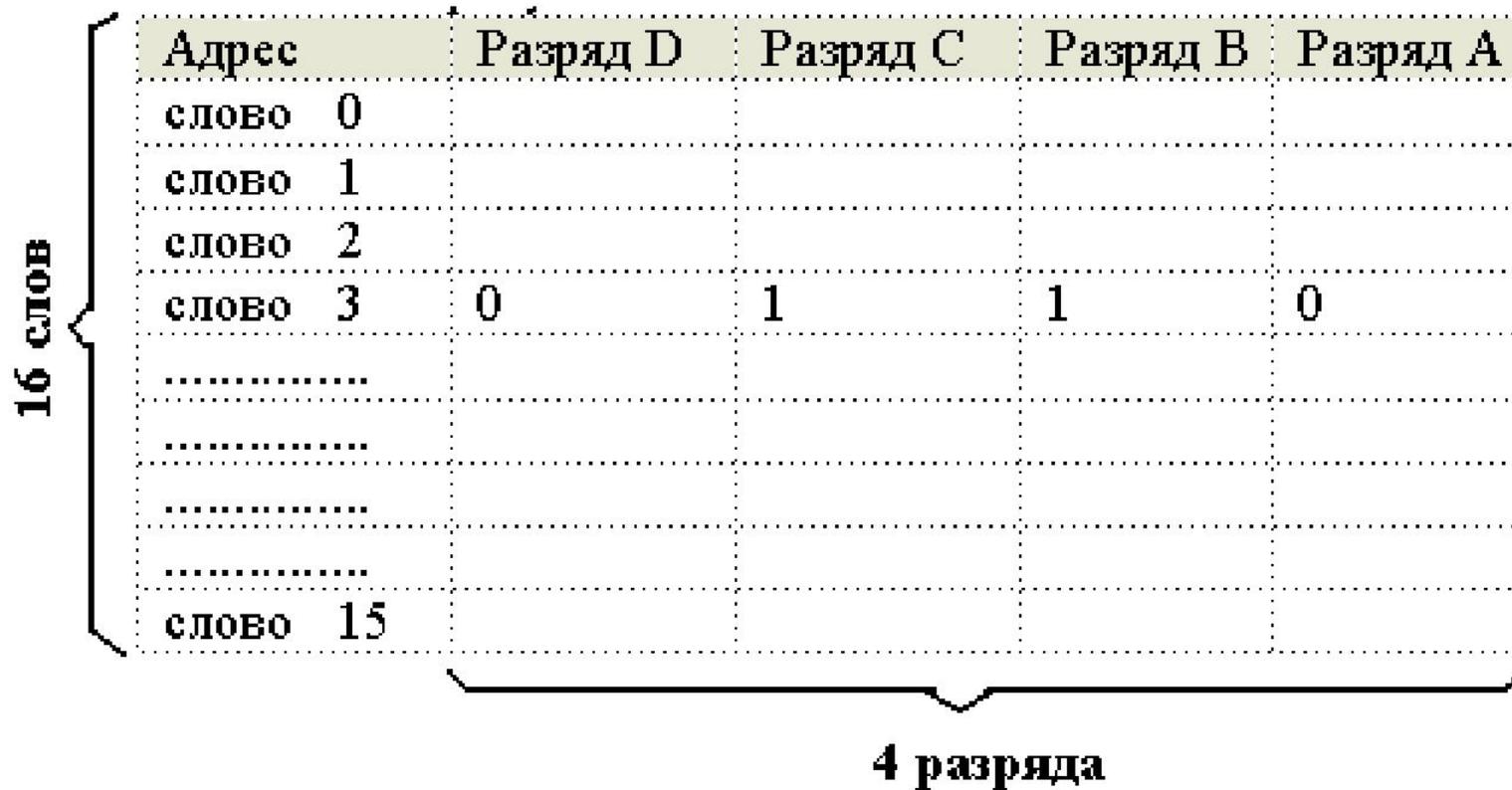
- Поле записи ОЗУ имеет 64 позиции, которые организованы в 16 групп (слов) по 4 разряда в каждой, т.е. ОЗУ 16x4. В такое ОЗУ можно поместить 16 слов по 4 разряда каждое.

# Логическая организация ОЗУ



- Двоичные слова одинаковой длины (данные и команды программы) помещаются в ячейки памяти по их адресам. Адреса можно представить себе как целые двоичные числа из некоторого интервала. Каждый адрес определяет в точности одну запоминающую ячейку памяти.

# Логическая организация ОЗУ

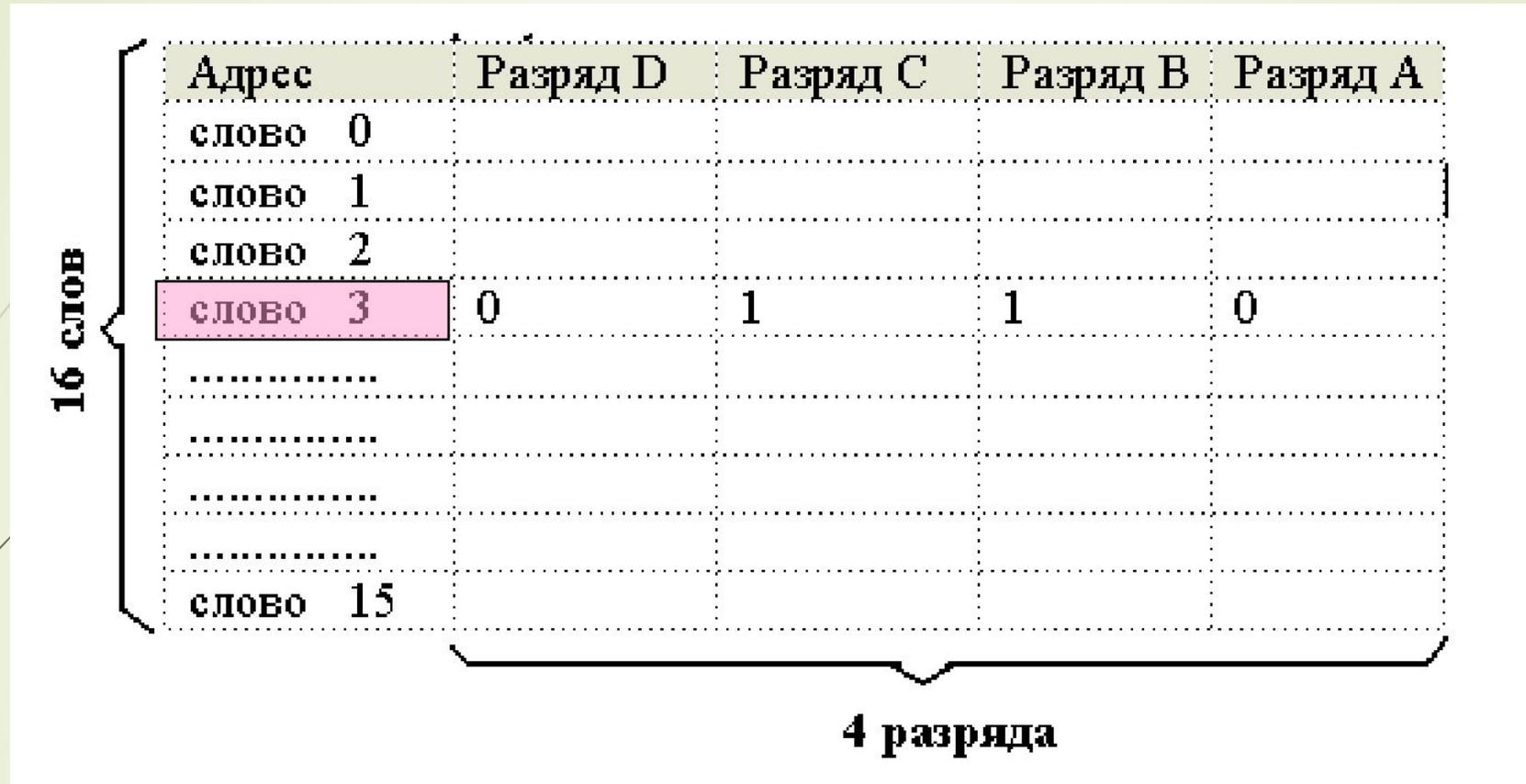


The diagram illustrates the logical organization of memory. It consists of a grid with 16 rows and 4 columns. The columns are labeled 'Разряд D', 'Разряд C', 'Разряд B', and 'Разряд A'. The rows are labeled 'слово 0' through 'слово 15'. A bracket on the left side of the grid indicates '16 слов'. A bracket at the bottom of the grid indicates '4 разряда'. The cell for 'слово 3' contains the binary value '0110'.

Адрес	Разряд D	Разряд C	Разряд B	Разряд A
слово 0				
слово 1				
слово 2				
слово 3	0	1	1	0
.....				
.....				
.....				
.....				
слово 15				

- Ячейку памяти, отведенную под нужное нам слово (т.е. номер слова) называют адресом этого слова. Например, адрес слова 3 - 0011 (в двоичной системе счисления). По этому адресу записана информация 0110 - это значение слова 3.

# Логическая организация ОЗУ



Адрес	Разряд D	Разряд C	Разряд B	Разряд A
слово 0				
слово 1				
слово 2				
слово 3	0	1	1	0
.....				
.....				
.....				
.....				
слово 15				

16 слов

4 разряда

- Ячейку памяти, отведенную под нужное нам слово (т.е. номер слова) называют адресом этого слова. Например, адрес слова 3 - 0011 (в двоичной системе счисления). По этому адресу записана информация 0110 - это значение слова 3.

# Логическая организация ОЗУ

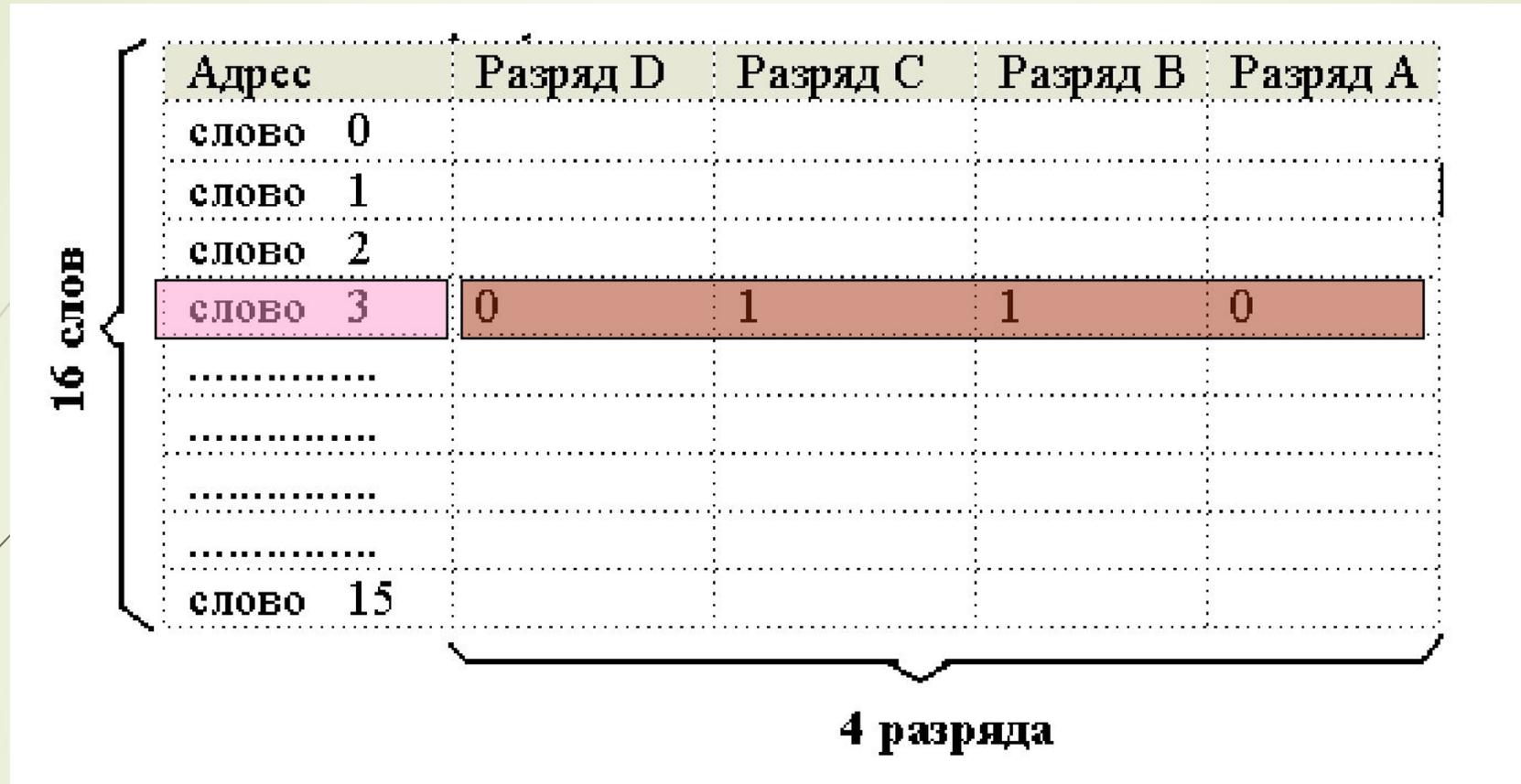
Адрес	Разряд D	Разряд C	Разряд B	Разряд A
слово 0				
слово 1				
слово 2				
слово 3	0	1	1	0
.....				
.....				
.....				
.....				
слово 15				

16 слов

4 разряда

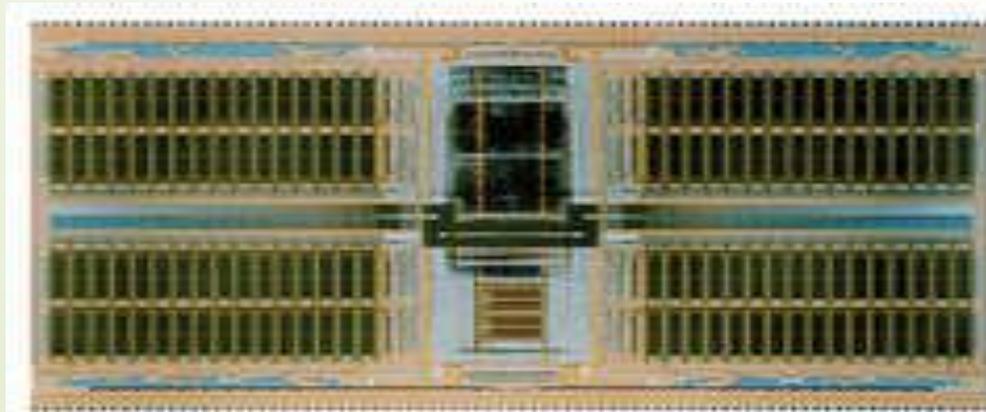
- Ячейку памяти, отведенную под нужное нам слово (т.е. номер слова) называют адресом этого слова. Например, адрес слова 3 - 0011 (в двоичной системе счисления). По этому адресу записана информация 0110 - это значение слова 3.

# Логическая организация ОЗУ



- Такие ОЗУ называются ЗУ с прямым доступом, т.к. можно по номеру слова и номеру разряда обратиться непосредственно к данной ячейке памяти и записать или считать информацию из этой ячейки.

- В ОЗУ записанная информация теряется при отключении питания. Поэтому такие ОЗУ называются энергозависимыми и используются для временного хранения информации.
- ОЗУ современных ЭВМ - это большие интегральные схемы (БИС).



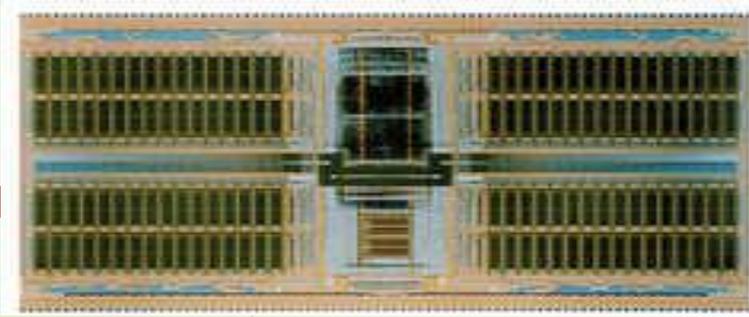
Микросхема  
DRAM

# По способу хранения информации различают ОЗУ

- а) Статические (SRAM) - каждый бит информации (0 или 1) хранится в элементе типа электронной защелки (триггер), состояние которого остается неизменным до тех пор, пока не будет сделана новая запись в этот элемент или не будет выключено питание. Статическая память является более дорогой и имеет высокое энергопотребление, но она имеет высокое быстродействие (единицы-десятки наносекунд) и низкую удельную плотность данных (порядка единиц Мбит на корпус). Поэтому статическая память используется в основном в качестве буферной кэш-память).

## По способу хранения информации различают ОЗУ

- б) динамические (DRAM) - каждый информационный бит хранится в виде заряда конденсатора (ПЗС - структуры, МДП-структуры). Из-за наличия токов утечки заряд конденсатора необходимо периодически восстанавливать (каждую миллисекунду) и, как следствие, эта память работает медленнее, чем статическая. Каждый ее элемент занимает на кристалле Интегральной Схемы (ИС) меньше места, чем в SRAM.



# DRAM

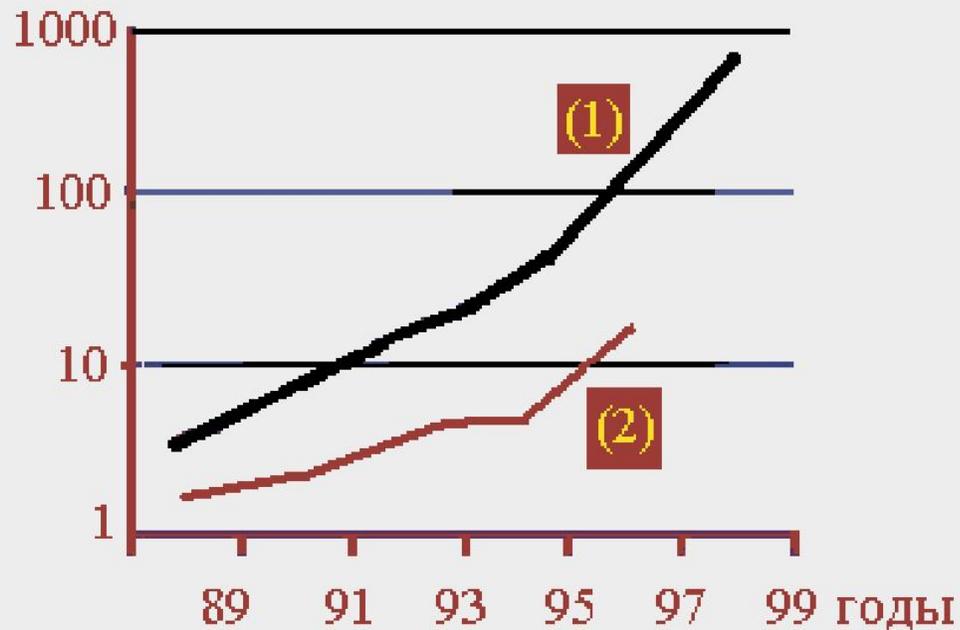
- В динамической памяти ячейки построены на основе областей с накоплением зарядов, занимающих гораздо меньшую площадь, нежели триггеры, и практически не потребляющих энергии при хранении. При записи бита в такую ячейку в ней формируется электрический заряд, который сохраняется в течение нескольких миллисекунд; для постоянного сохранения заряда ячейки необходимо регенерировать - перезаписывать содержимое для восстановления зарядов. Ячейки микросхем динамической памяти организованы в виде прямоугольной (обычно - квадратной) матрицы. Ячейки динамической памяти имеют большее время срабатывания (десятки-сотни наносекунд) и большую удельную плотность (порядка десятков Мбит на корпус) и меньшее энергопотребление. DRAM в 4-5 раз дешевле статической.

# Сравнение SRAM и DRAM

- Наиболее эффективным является использование динамической памяти вместе с небольшой по размерам статической памятью.

	DRAM	SRAM
емкость	до 64 - 128 Мб	до 512 Кбайт
себестоимость 1 Мб	30-35 \$	100 \$
время доступа	60-150 нс	15-20 нс

## Рост пропускной способности памяти DRAM (2) и производительности центральных процессоров CPU (1).



Десять лет усилий разработчиков привели к росту:

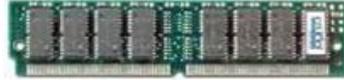
- производительности процессоров в 100 раз,
- производительности основной памяти только в 12 раз.

Для обработки качественной графики и видео в ПЭВМ нужна более быстрая основная память. Ее пропускная способность, как предсказывает компания Intel, в 1999 года достигнет 1,6 Гбайт/с.

# Конструктивы модулей ОЗУ

- *DIP, Dual In-line Package* 

- *SIMM, Single In-line Memory Module*



- *DIMM, Dual In-line Memory Module*



- *RIMM<sup>TM</sup>, «Rambus In-line Memory Module»*

- *mobile*



## Разновидности ОЗУ

### **SIMM:**

- *FPM DRAM, Fast Page Mode DRAM*
- *RAMEDO, RAM Extended Data Out*
- *BEDO DRAM, Burst Extended Data Output*
- ...

### **DIMM:**

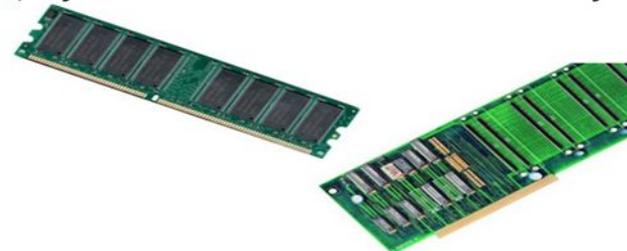
- *SDRAM, Synchronous DRAM*
- *DDR 1,2 SDRAM, Double Data Rate SDRAM*
- *RDRAM, Rambus DRAM*
- *DRDRAM, Direct Rambus DRAM*
- *GDDR 1,2,3,4 SDRAM, Graphics DDR*
- *SDR SDRAM*
- ...

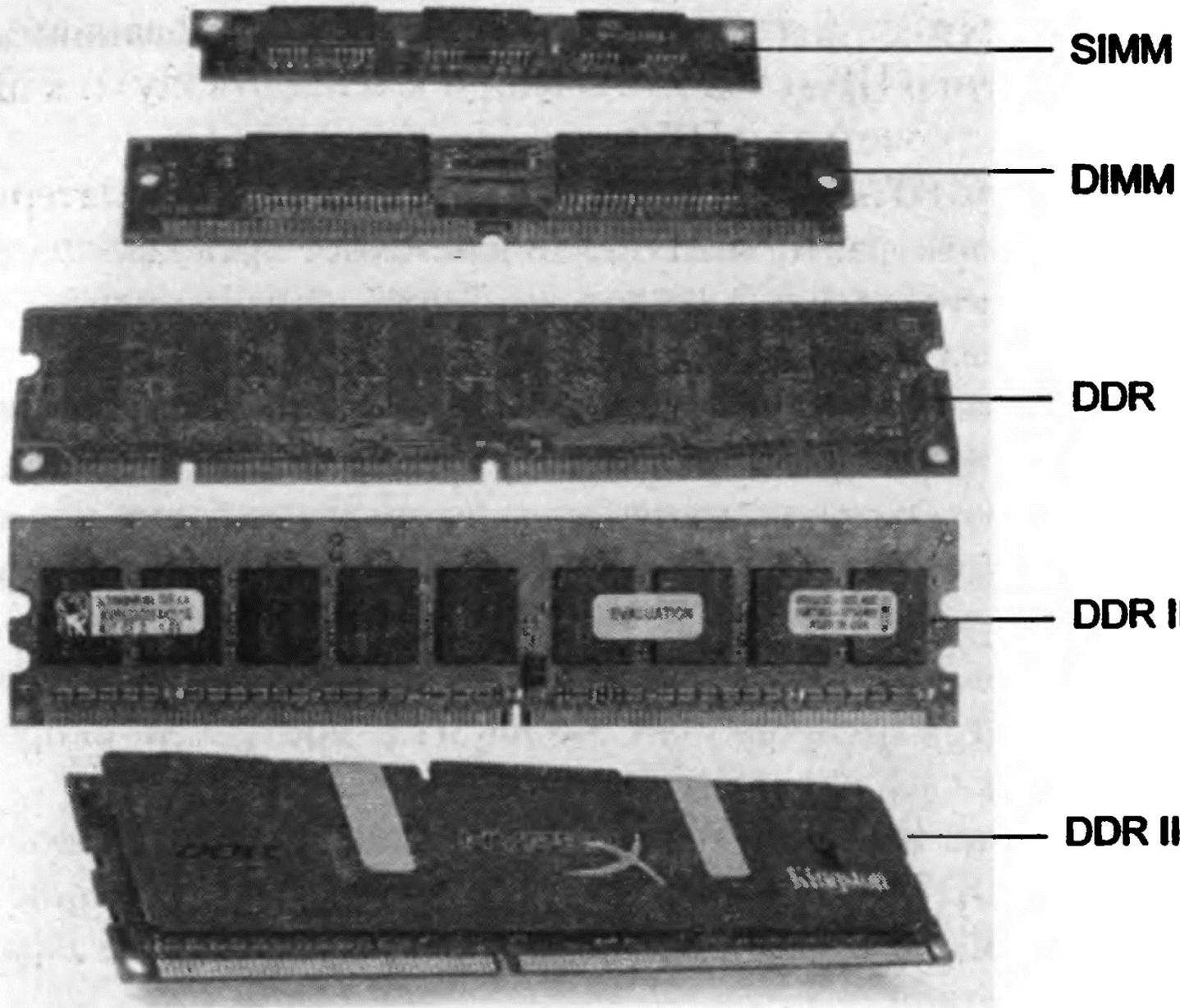
### **Mobile:**

- *Mobile SDRAM*
- ...

## Типы микросхем ОЗУ

- *SRAM, Static Random Access Memory*
- *DRAM, Dynamic Random Access Memory*





SIMM

DIMM

DDR

DDR II

DDR III

Индексация модуля памяти

DDR3 Частота, МГц)	Индекс
1066	PC3-8500
1333	PC3-10664
1600	PC3-12800
1800	PC3-14400
2000	PC3-16000

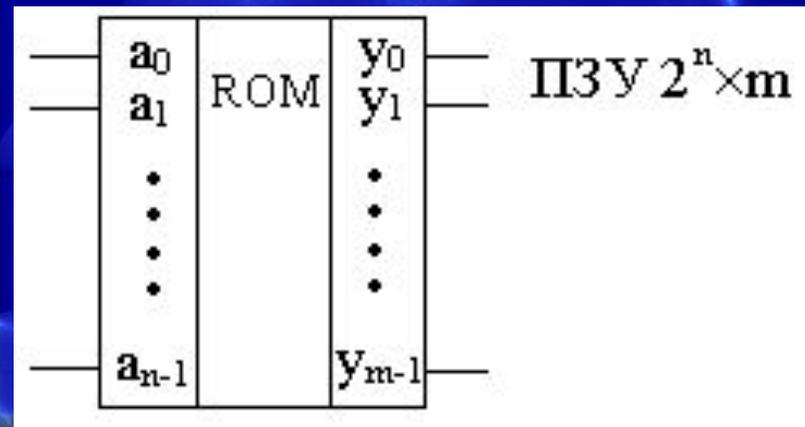
**Рис. 2.7. Эволюция развития модулей памяти**

# Постоянные запоминающие устройства - ПЗУ

- Это полупроводниковые энергонезависимые ЗУ. Ячейки памяти в ПЗУ, как правило, не триггерного типа: это специальные элементы (диоды, транзисторы), которые при изготовлении ПЗУ устанавливаются в состоянии логического "0" или "1".
- В запрограммированное ПЗУ нельзя занести новую информацию, а можно только многократно считывать уже записанную - (ROM-Read-Only Memory).
- В ПЭВМ ПЗУ используют для хранения программ начального запуска (самозагрузки) и других обслуживающих системных программ. Программы, хранимые в ПЗУ, образуют программно-аппаратное обеспечение ЭВМ.

# Постоянные запоминающие устройства (ПЗУ)

ПЗУ представляет собой чисто комбинационную схему, имеющую  $n$  адресных входов и  $m$  выходов.



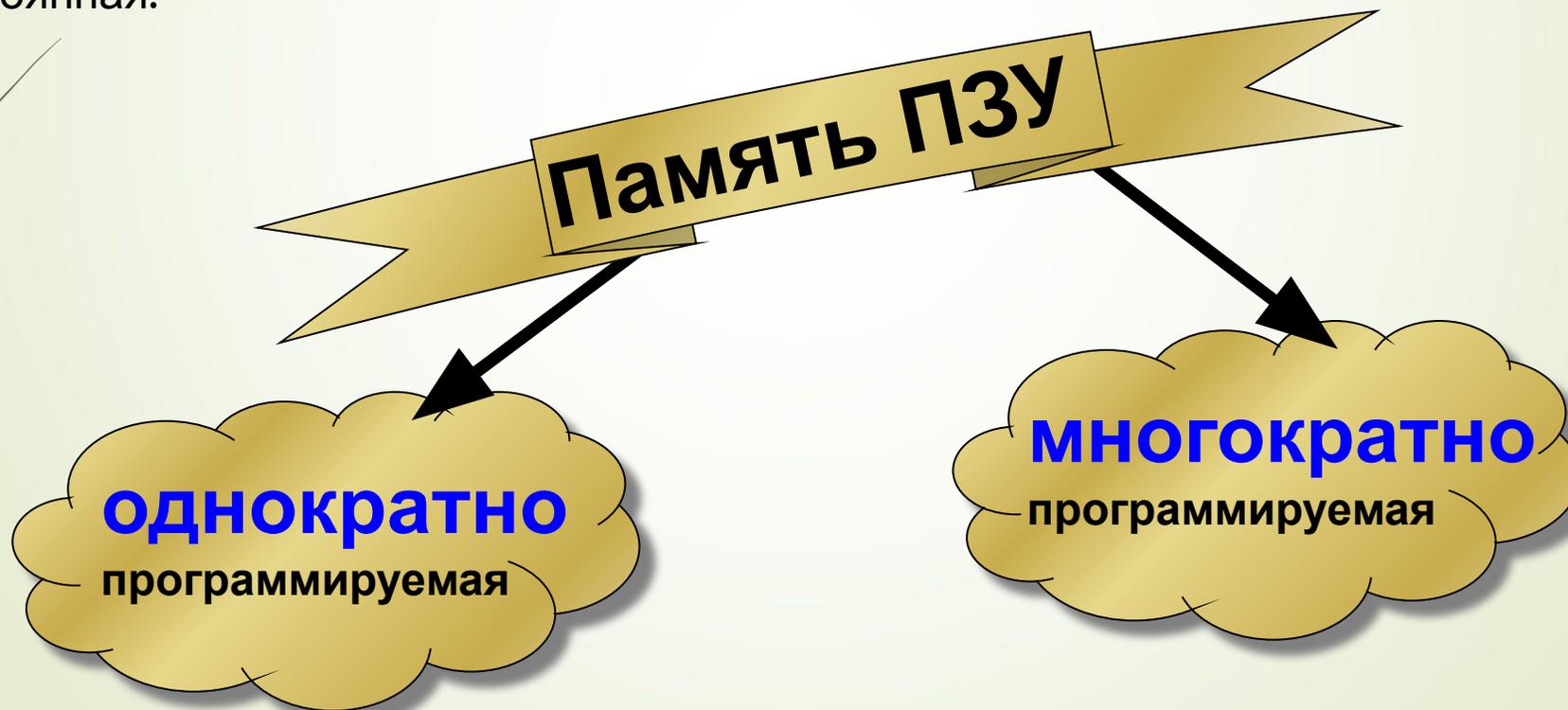
Схемное обозначение ПЗУ

# Постоянное запоминающее устройство (ПЗУ)

предназначено только для хранения и чтения хранящейся в них информации

полупроводниковые ЗУ, магнитные диски, накопители на магнитных лентах

**ПЗУ** (ROM - read only memory ) служит для хранения неизменяемой программной и справочной информации. Просмотр этой памяти невозможен так как эта память постоянная.



## 3 вида ПЗУ по принципу действия :

- 1) масочное ПЗУ (MROM) - информация "защита" по специальному шаблону или маске при его изготовлении и изменению не подлежит.
- 2) программируемое ПЗУ (PROM) - пользователю предоставляется возможность самостоятельно запрограммировать ПЗУ после его изготовления. Программирование осуществляется специальным устройством программатором и состоит в пережигании плавких перемычек. После такого программирования содержимое памяти не может быть изменено.

## 3 вида ПЗУ по принципу действия :

- 3) стираемые программируемые ПЗУ (EPROM) - записанная информация может стираться с помощью интенсивного ультрафиолетового излучения (за 10-15 мин). Новая запись осуществляется с помощью специального программатора. Фирмы Toshiba и NEC выпустили кристаллы емкостью 512 Кбайт, обеспечивающие время доступа 150-120 нс.

## По способу занесения информации (программирования) полупроводниковых ПЗУ различают:

- масочные ПЗУ, программируемые изготовителем в процессе производства,
- программируемые (ППЗУ), в которых пользователь может посредством программатора один раз, как в микросхеме K155PE3, разрушить переключки, исходя из собственной программы или кода,
- репрограммируемые ЗУ (РПЗУ), допускающие многократное перепрограммирование, как в микросхеме K573PPФ (стирание осуществляется ультрафиолетовым облучением кристалла) или KP557PP (стирание — под воздействием электрического сигнала). Сюда же относят *Flash-память* (например, микросхемы 28F0165S и 28F032SA ёмкостью 432 Мбит при временах доступа 70150 нс фирмы Intel), обладающую свойством быстрого стирания массивов данных со всего кристалла одним сигналом.



# Кэш-память cache memory - буферное ЗУ

- Необходимость в создании кэш-памяти возникла потому, что появились процессоры, работающие с очень большим быстродействием и для выполнения сложных прикладных процессов возникла потребность в очень большой памяти.
- Использование большой сверхскоростной памяти было невыгодно. Поэтому между оперативной памятью и процессором стали устанавливать меньший по размерам высокоскоростной буфер, названный кэш-память.





# КЭШ-память (CACHE)

- Эффективным способом увеличения скорости обмена информацией между процессором и ОЗУ является использование кеш-памяти, вводимой сначала непосредственно в центральный процессор (первый уровень), а затем помещенной на материнскую плату (второй уровень).

-

# КЭШ-память (CACHE)

□ КЭШ-память логически представляет собой промежуточный буфер, через который

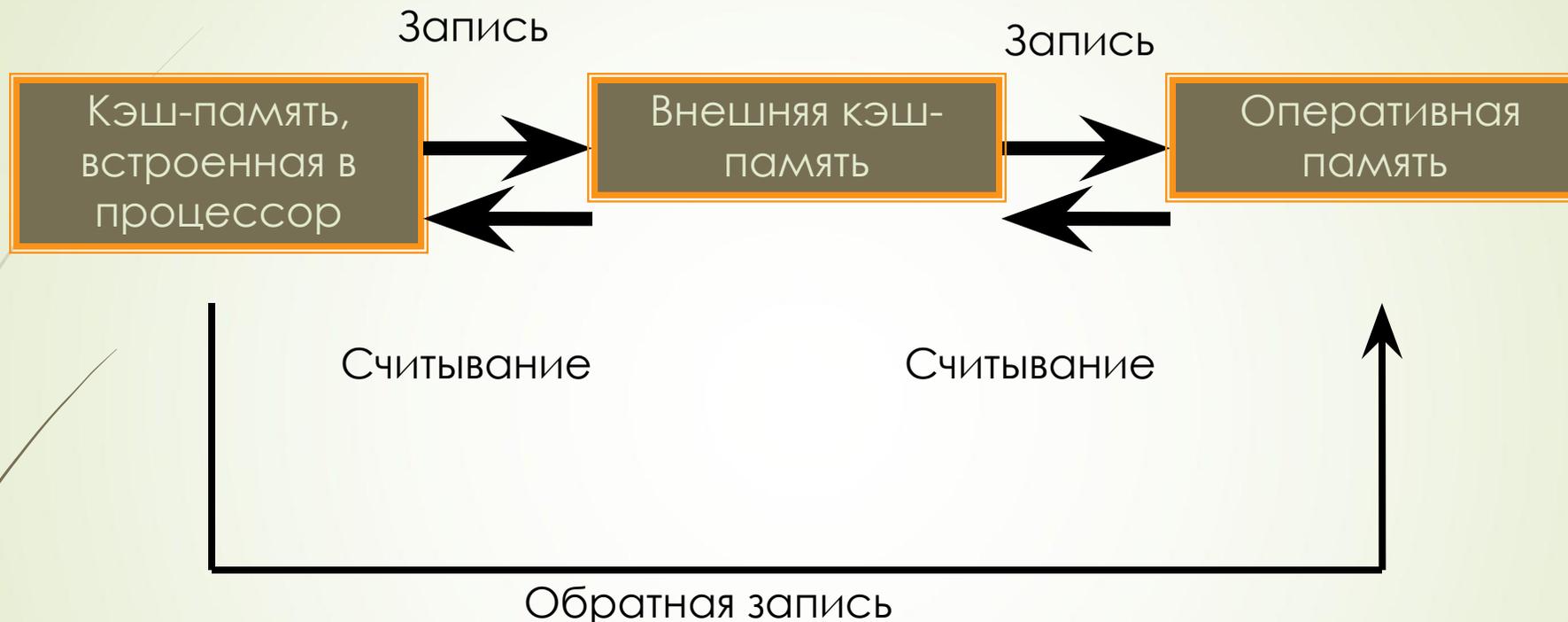
прокачиваются данные. Область кеш-памяти, предназначенная для временного хранения данных перед или после их пересылки из одного компонента системы в другой, либо из одной системы в другую называют **буферной** или **буфером**. Копирование может осуществляться как по чтению, так и по записи. Выигрыш по быстродействию достигается благодаря тому, что часто используемые данные находятся в КЭШе, а потому доступ к ним со стороны микропроцессора ускоряется.

# Структура КЭШ-памяти



Результаты эталонных тестов показали, что при работе кэш-памяти производительность центрального процессора возрастает более, чем в три раза.

# Структура КЭШ-памяти



Объема встроенной кэш-памяти 1-го уровня (емкостью 8 Кбайт) недостаточно для крупных частей программы или количеств данных, поэтому на современных материнских платах размещают дополнительную внешнюю кэш-память 2-го уровня (Second-level-Cache), среднее время обращения порядка 25 нс.

## *Cache Levels*

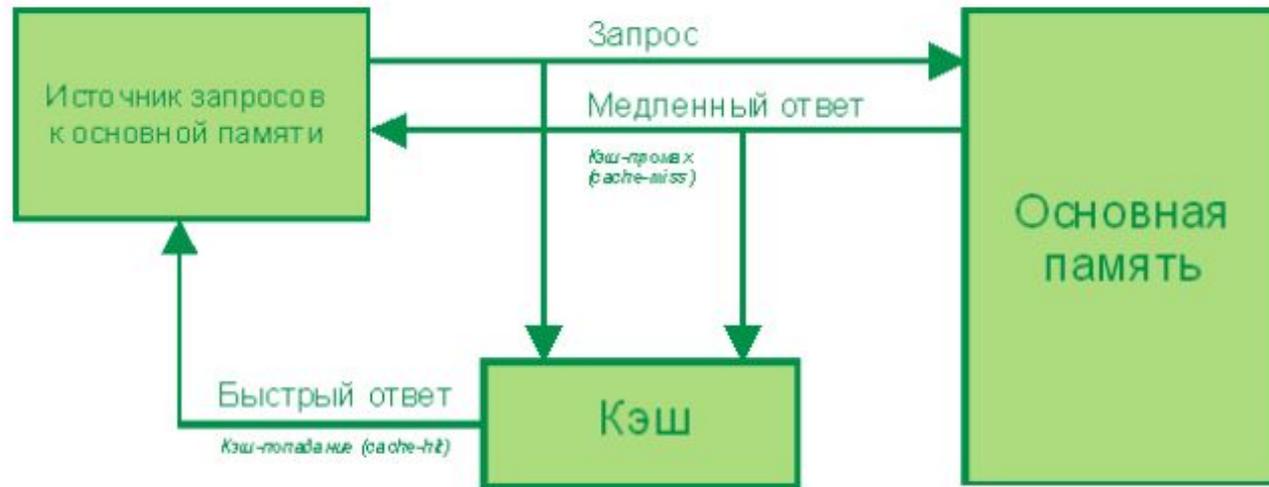
- **L1**, встроен в основное ядро МП
- **L2**, на плате МП / на материнской плате
- **L2/3**, на материнской плате
- **L3/4**, в поле ОП / в модуле ВЗУ

# Структура КЭШ-памяти



Если процессор не находит желаемые данные во внутренней кэш-памяти, то он, в первую очередь, пытается найти их во внешней кэш-памяти и только после этого производит выборку данных из более медлительной оперативной памяти.

# Принцип действия кэш-памяти



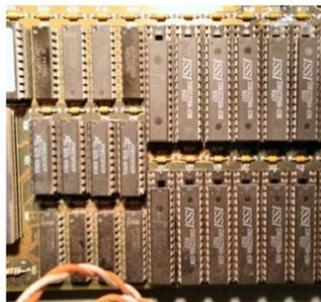
## Cache Levels

- L1, встроен в основное ядро МП
- L2, на плате МП / на материнской плате
- L2/3, на материнской плате
- L3/4, в поле ОП / в модуле ВЗУ

## Cache

### Cache memory

Собственно кэш-память - физический конструктив

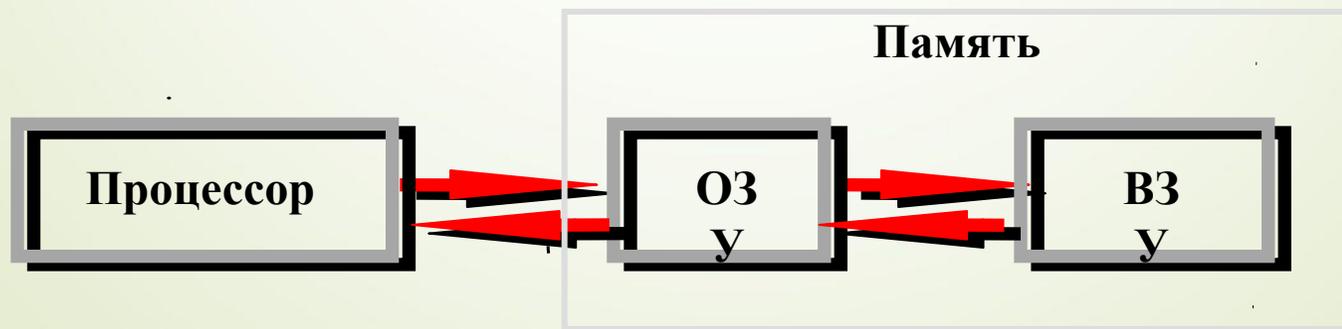


**Кэширование (хеширование)** - способ совместного функционирования двух типов запоминающих устройств, отличающихся временем доступа и стоимостью хранения данных, который за счёт динамического копирования в «быстрое» ЗУ наиболее часто используемой информации из «медленного» ЗУ позволяет:

- с одной стороны, **уменьшить среднее время доступа** к данным,
- а с другой стороны, **экономить** более дорогую более быстродействующую память.

# Внешняя память - ВЗУ

- Большая часть информации хранится *вне ЭВМ*, во **внешних ЗУ** большой емкости (перфокарты, перфоленты, магнитные ленты, барабаны, диски, оптические диски) - такая память называется **внешней (ВЗУ)**.
- Для ВЗУ сообщения переносятся только между запоминающей средой и ОЗУ, поэтому такую память часто называют **память с непрямым доступом**.



# Уровни памяти / Иерархия ЗУ

УРОВЕНЬ	Время доступа	Объём	Удельная стоимость хранения
<b>МПП</b> - сверхоперативное ЗУ, регистры (доступ за 1 такт процессора).	~ 2-3 нс	Десятки байт	МАХ
<b>Кэш-память</b> – буферное ЗУ (доступ 1-2, 3-5 тактов).	~ 5-8 нс	Десятки-сотни Кбайт	
<b>Основная память (ОП)</b> <b>ОЗУ</b> (до 20 тактов). <b>ПЗУ</b> – «постоянное ЗУ».	~ 5-20 нс ~ 35-100 нс	Сотни-тысячи Мбайт, Сотни Кбайт	
<b>Внешняя память (ВЗУ)</b> НГМД, НЖМД, НМОД, НМЛ ...	Десятки мс	Сотни-тысячи Гбайт	Min

## По способу доступа к информации устройства памяти разделяют на

- **адресные** (доступ к ячейке памяти по адресу коду (по коду номера ячейки),
- **с последовательным доступом** (*доступ* просмотром ячеек памяти, расположенных в последовательном порядке по очереди);
- **ассоциативные** (поиск и извлечение информации производятся по некоторому признаку самой информации, содержащемуся в хранимом слове, т. е. по содержанию (например, по выделенному тэгу (полю) слов).

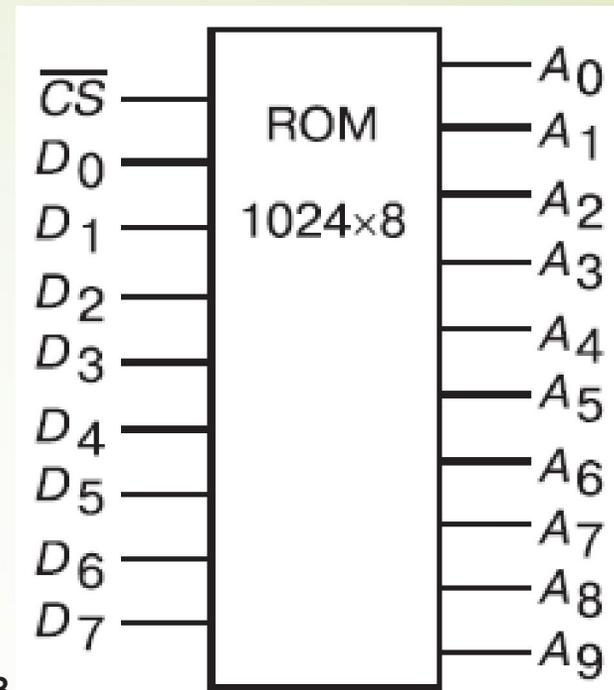
Основная область применения ассоциативного доступа к данным — *сверхоперативная кэш-память*, построенная на основе быстродействующих ЗУ и предназначенная для ускорения доступа к данным и часто используемым подпрограммам с целью повышения скорости выполнения основной программы.

Основными показателями ЗУ являются объём памяти (в кило, мега или гигабайтах), *быстродействие* (производительность) — время цикла записи или считывания информации, *временные соотношения* между различными сигналами и *свойство энергонезависимости*, т. е. способность ЗУ сохранять/не сохранять данные при аварийном отключении напряжения питания.

**Структурно функциональная  
организация ОЗУ и ПЗУ зависит от  
способа доступа (одномерной или  
двумерной адресации) к нужному  
элементу или нужной ячейке памяти.**

Каждое ЗУ характеризуется набором адресов и содержимым ячеек. Число адресов должно быть равно числу ячеек памяти. Обратившись к ЗУ по тому или иному адресу, определяют содержимое соответствующей ячейки, которая может содержать либо единственный бит (0 или 1), либо несколько битов информации.

Десять линий адреса  $A_9A_0$  обеспечивают адресацию  $2^{10} = 1024 = 1$  кбит ячеек памяти, а восемь линий  $D_7D_0$  данных показывают, что в каждой ячейке памяти хранятся 8 бит информации. Для этой микросхемы организацию памяти записывают как 1024×8. Байт считывается с микросхемы, когда сигнал считывания с кристалл  $\overline{CS} = 0$  (черта над буквами CS означает инверсную логику, т. е. сигнал действует, когда он имеет значение логического нуля).



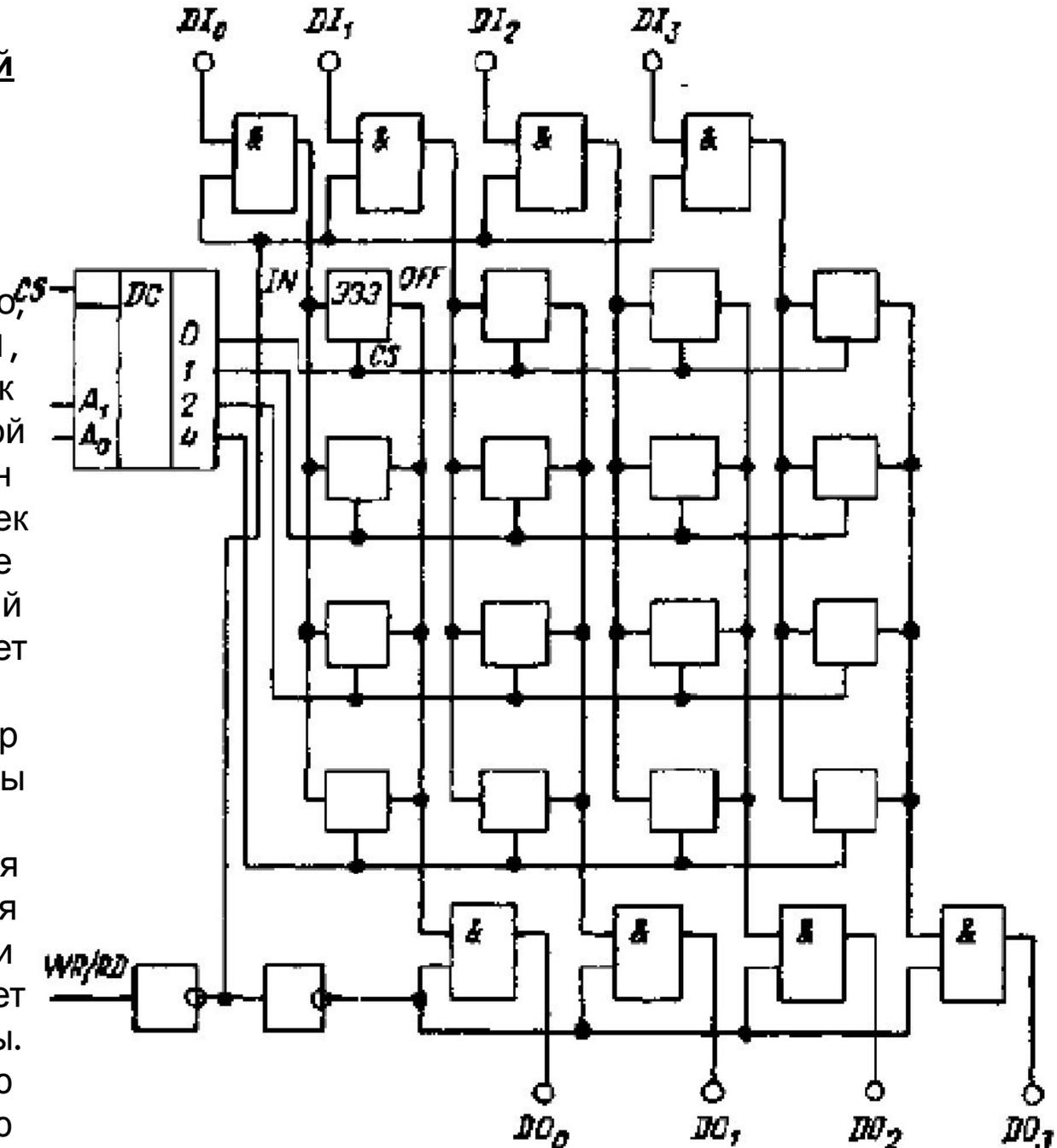
в ЗУ с произвольным доступом для хранения каждого бита информации используется отдельный ЭЗЭ.

Структурная схема ЗУ (ЛА=16, организация 4X4) с **одномерной адресацией** содержит матрицу 4X4 ЭЗЭ, каждый из которых снабжен тремя выводами: входом информации (IN), выходом информации (OFF) и выходом разрешения работы — выбора кристалла (CS — chip select), дешифратор адреса, входные и выходные ключи на элементах 2И.

Если на вход дешифратора подано некоторое адресное слово, на соответствующем его выходе формируется сигнал лог. 1, который выбирает псе ЭЗЭ, расположенные в одной из строк матрицы. Дальнейшая работа устройства зависит от того, какой режим работы задан управляющими сигналами. Если подан сигнал «чтение» ( $WR/RD=1$ ), то информация из выбранных ячеек подается на первые входы выходных элементов 2И. На вторые входы этих ключей подан единичный (пассивный) логический сигнал. Информация, хранившаяся в выбранных ЭЗЭ, попадает на выходные выводы  $DO_0, DO_1, DO_2, DO_3$ .

Если на схему подан сигнал «запись» ( $WR/RD=0$ ), то информация с входных шин  $DI_0, DI_1, DI_2, DI_3$  через входные элементы 2И подается на входы ЭЗЭ и записывается в них.

Особенностью данной структурной схемы является одновременная запись и считывание информации, хранящейся в строке матрицы ЭЗЭ. Поэтому при заданной разрядности хранимого слова наращивание объема памяти может выполняться только за счет увеличения числа строк матрицы. Это, в свою очередь, ведет к значительному увеличению количества выходов дешифратора, т. е. к усложнению аппаратной реализации



# Структурная схема ЗУ с селекторами

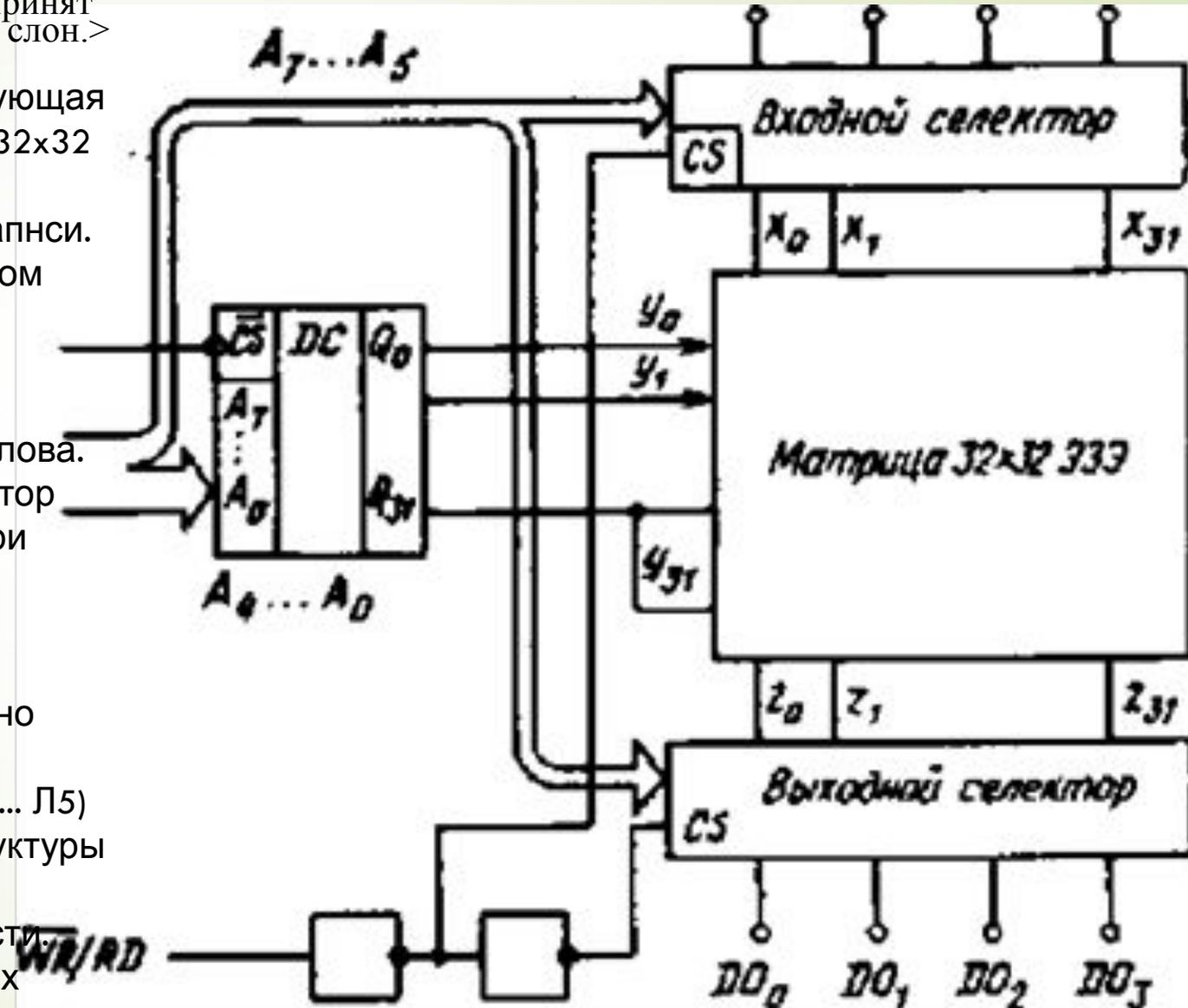
С точки зрения упрощения дешифратора желательно выполнять матрицу ЭЗЭ прямоугольной. Однако при использовании рассмотренной структуры это предполагает увеличение разрядности выходного слова.

Решить это противоречие можно, если в качестве входных и выходных ключей ИС использовать соответственно схемы демультиплексоров и мультиплексоров. Применительно к ЗУ их принято называть селекторами. При этом часть разрядов адресного слова используется для управления селекторами.

Структурная схема ЗУ с организацией 256x4, реализующая описанный принцип, включает матрицу ЭЗЭ размером 32x32 элемента, дешифратор адреса, входной и выходной селекторы и элементы управления режимами чтения/записи. Каждый ЭЗЭ снабжен тремя выводами входом и выходом информации и выводом разрешения работы (выборки).

Управление доступом к такому объему информации требует подачи на вход ЗУ 256 различных адресов, т. е. предполагает использование 8-разрядного адресного слова. Младшие его разряды ( $A_4 \dots A_0$ ), поступая на дешифратор адреса, выбирают из матрицы одну из 32 строк ЭЗЭ. При этом информация в зависимости от заданного режима работы может либо считываться ( $WR/RD = 1$ ), либо записываться ( $WR/RD = 0$ ) во все элементы выбранной строки. Выбор нужных ЭЗЭ производится соответственно либо входным, либо выходным селекторами, которые в зависимости от значения старших разрядов адреса ( $A_7 \dots A_5$ ) выбирают из строки нужные элементы. Описанные структуры выгодно использовать при одновременной записи/считывании информационных слов большой разрядности.

В противном случае практическая реализация данных структур приводит к большим аппаратным затратам.



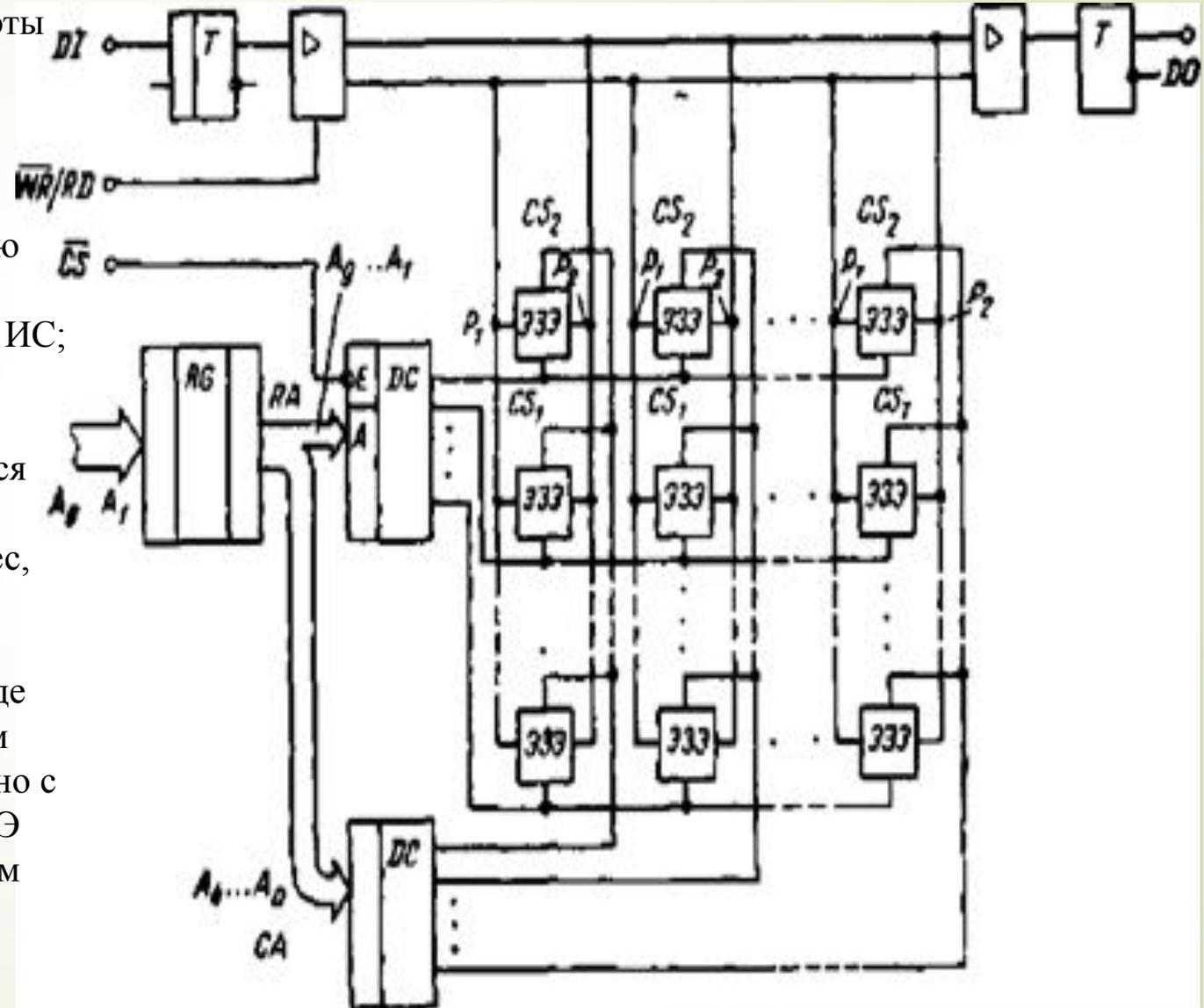
структура содержит матрицу ЭЗЭ, статический регистр адреса, дешифраторы строки и столбца, усилители записи и считывания, входной и выходной буферные триггеры. Однако, в отличие от схемы одномерной адресации, каждый ЭЗЭ матрицы содержит не один, а два вывода разрешения работы (CS1 и CS2). При этом информационные выходы  $p_1$  и  $p_2$  обратимые, т. е. позволяя как записывать так и считывать информацию. Для выбора нужной ячейки на оба входа CS необходимо подать активные логические уровни.

Цепи управления матрицей ЭЗЭ обеспечивают реализацию одного из трех режимов работы:

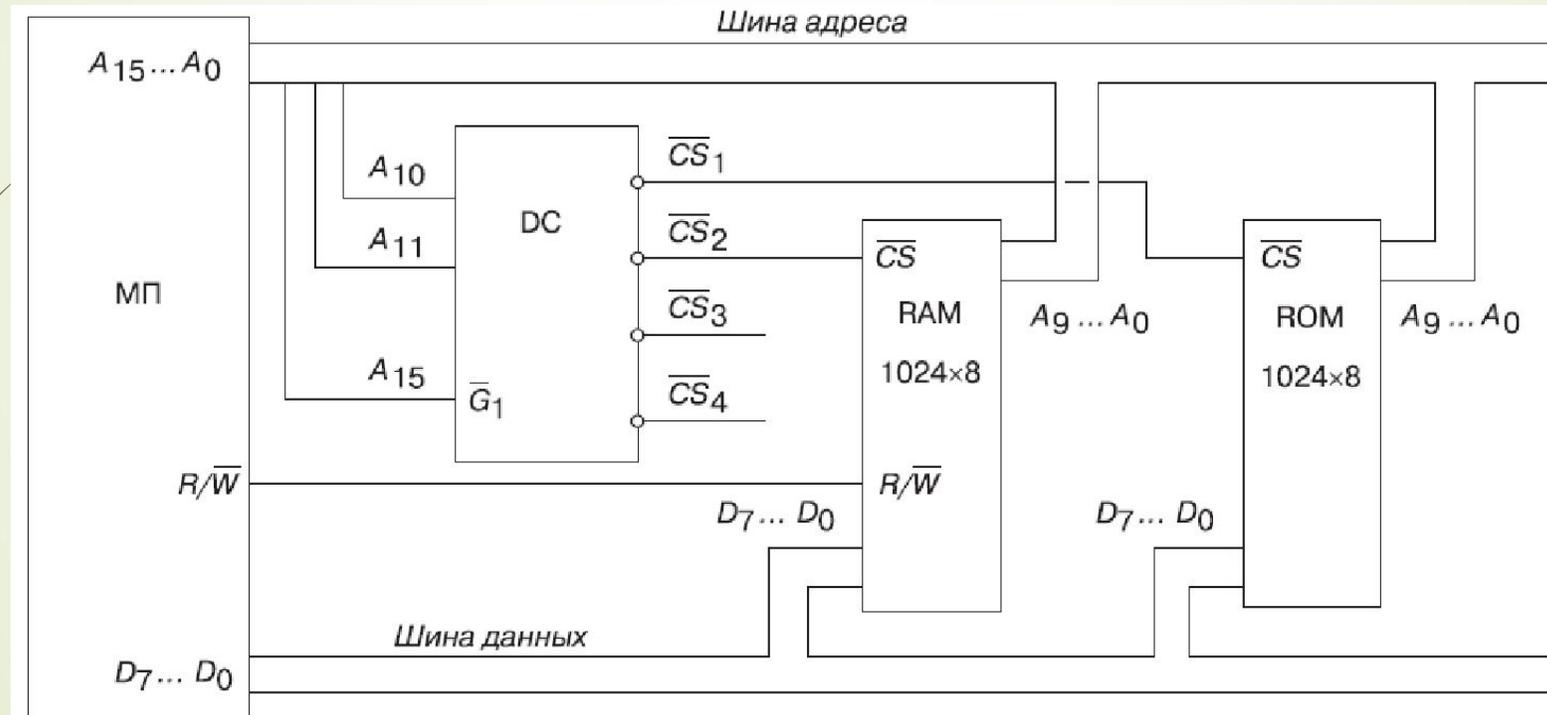
- хранения, при котором ЭЗЭ отключены от входа и выхода ИС;
- чтения, при котором информация из ЭЗЭ, выбранного по соответствующему адресу, выдается на выход ИС;
- записи, при которой информация со входа ИС записывается указанному адресу.

Каждому ЭЗЭ матрицы присваивается определенный адрес, поиск которого производится указанием номеров соответствующих строки и столбца. Эти номера формируются на выходах дешифраторов. Адрес ЭЗЭ в виде двоичного числа принимается по адресной шине регистром адреса. Число разрядов регистра адреса однозначно связано с объемом памяти ИС. Число строк и столбцов матрицы ЭЗЭ обычно выбирается равным целой степени числа 2, причем общий объем памяти определяется произведением числа строк на число столбцов :

**ЗУ С ДВУМЕРНОЙ АДРЕСАЦИЕЙ** применяют при необходимости побитовой записи-считывания информации



На рис. 6.30 изображена схема памяти простейшего микропроцессорного устройства. Разводка контактов ОЗУ (RAM) аналогична ПЗУ, но в ней добавлен контакт считывания/записи  $R/\bar{W}$  для *шины управления* микропроцессора (МП). Сигнал  $R/\bar{W}$  управляет направлением передачи байтов: при  $R/\bar{W} = 1$  ЗУ работает в режиме записи в ячейки памяти слов данных  $D7D0$ , при  $R/\bar{W} = 0$  ЗУ работает в режиме чтения содержимого ячеек.





Микросхемы ОЗУ и ПЗУ подключаются к *шине данных* и необходимому числу младших линий *шины адреса*. Сигналы выборки  $8_i$  микросхемы формирует дешифратор DC, обеспечивая одномерный выбор только одной микросхемы. В данный момент времени только на одном из выходов  $8_i$  (из четырёх) дешифратора DC действует сигнал логического нуля при условии, что на его входе сигнал разрешения  $G_1 = 0$ .

При  $G_1 = 1$  ни на одном из выходов DC не будет сигнала логического нуля. Итак, сигнал на линии  $A_{15}$  активизирует дешифратор, а сигналы на линиях  $A_{10}$  и  $A_{11}$  дешифруются и определяют выбор микросхемы.

Если в схеме памяти больше четырёх микросхем, то вместо дешифратора  $2 \times 4$  по требуется дешифратор  $3 \times 8$ , а на его входы подключают неиспользованные старшие линии шины адреса.

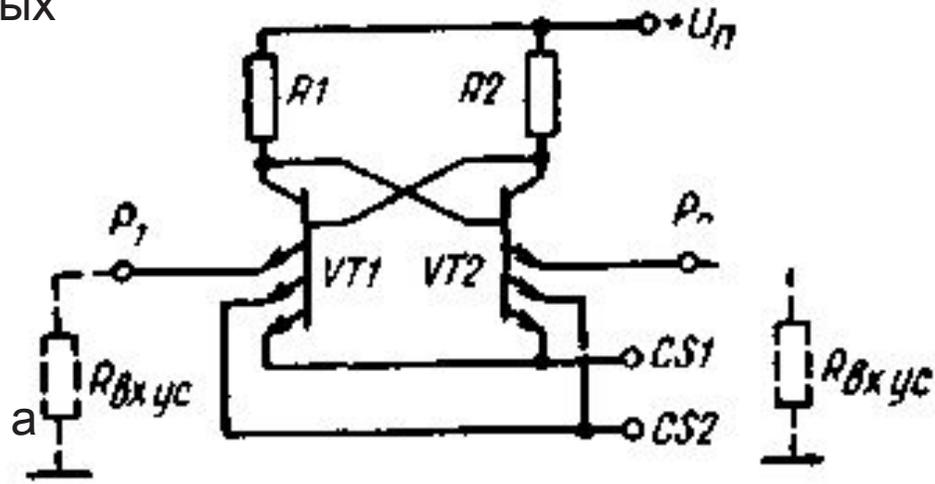
Очевидно, что для хранения бита информации нужен *элемент памяти*, а для хранения бинарного слова — *ячейка* памяти. В качестве элементов, хранящих информацию в ОЗУ, используют *триггеры*, как в микросхеме К155РУ2 (*статическое* наиболее распространённое ОЗУ, имеющее, как правило, одно питающее напряжение), или *ёмкость транзистора*, как в микросхеме КР565РУ1 (*динамическое* ОЗУ, в которой информация хранится в форме наличия или отсутствия заряда). Интегральные схемы ОЗУ содержат на одном кристалле матрицу запоминающих элементов, дешифраторы адреса, формирователи сигналов, усилители считывания и другие схемы управления, необходимые для функционирования ОЗУ.

Элемент памяти на биполярных транзисторах с одномерной адресацией (рис. 6.31) представляет собой триггер, выполненный на двухэмиттерных транзисторах. Нижние эмиттеры транзисторов  $VT1$  и  $VT2$  подключены к адресной шине, верхние эмиттеры — к разрядным шинам 1 и 2.

В режиме *записи* на адресную шину с выхода дешифратора поступает высокий уровень сигнала, поэтому нижние эмиттеры не оказывают влияния на работу транзисторов. На разрядные шины 1 и 2 поступают с устройства управления МП разные уровни напряжения. Будем считать, что в элементе записан 0, если транзистор  $VT1$  открыт, а транзистор  $VT2$  закрыт. Если нужно записать в элемент единицу (1), то высокий уровень напряжения подаётся через шину 1 на верхний эмиттер транзистора  $VT1$ , а через шину 2 на верхний эмиттер транзистора  $VT2$  — низкий уровень. Этот случай соответствует закрытому состоянию транзистора  $VT1$  и открытому состоянию транзистора  $VT2$ .

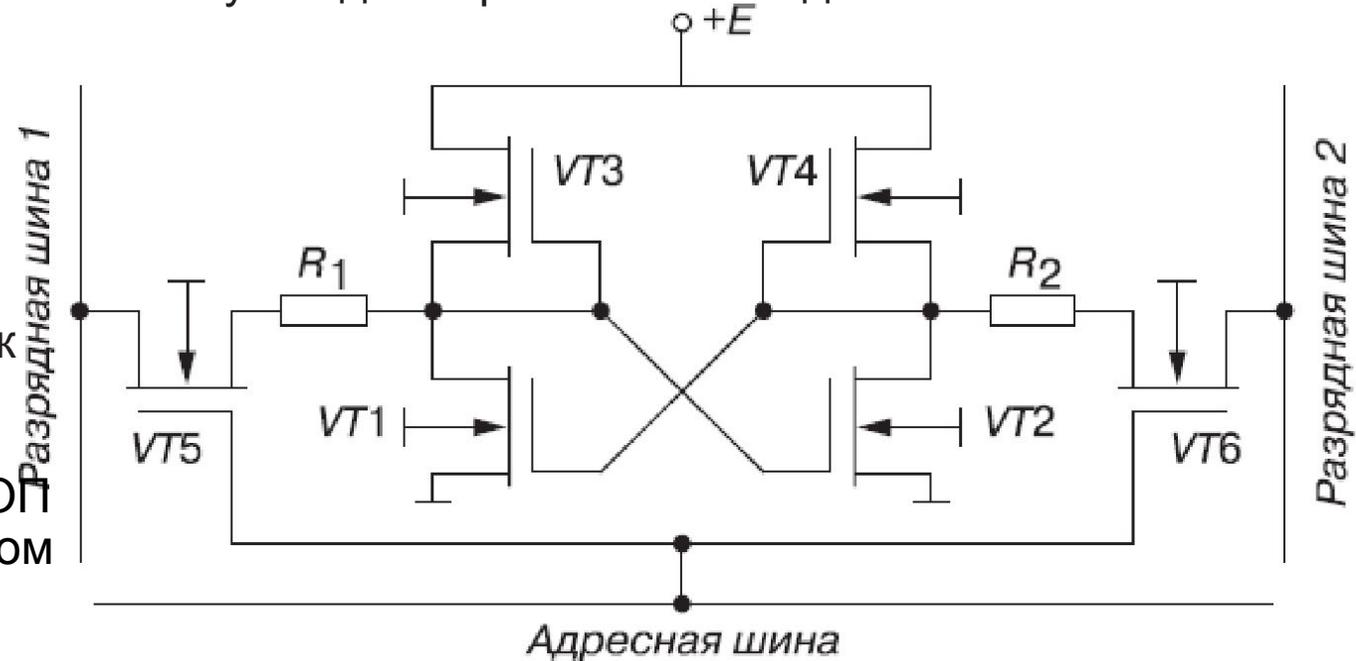
В режиме *считывания* протекающий через верхний эмиттер транзистора  $VT2$  ток в шине 2 создаёт на сопротивлении усилителя считывания напряжение, соотв. логической единице. Этот сигнал (1) передаётся на выход ЗУ. Если транзистор  $VT1$  открыт, а транзистор  $VT2$  закрыт, то на выход ЗУ подается сигнал 0.

В режиме *хранения* на адресную шину поступает низкий уровень напряжения, поэтому можно считать, что нижние эмиттеры обоих транзисторов подключены к общей точке схемы. В этом режиме сигналы на верхних эмиттерах не оказывают влияние на состояние транзисторов, т. е. состояние триггера остаётся неизменным. Запоминающий элемент (триггер) с двухкоординатной выборкой выполняют на двух трёхэмиттерных транзисторах. Третьи эмиттеры каждого транзистора соединяют с выводами *второй* адресной шины.



элемент памяти на МОП-транзисторах с однокоординатной выборкой приведен на рис. 6.32. Триггер образован транзисторами VT1-VT4 (транзисторы VT3 и VT4 с нулевыми напряжениями на затворах по отношению к своим истокам служат нагрузками в триггере). Резисторы R1 и R2 служат для ограничения входных токов в момент отпирания ключевых транзисторов VT1 и VT2.

Управление триггером для записи и считывания осуществляется переключением транзисторов VT5 и VT6. В исходном состоянии напряжения на обеих разрядных шинах 1 и 2 данных, как и на адресной шине, равны нулю. При этом транзисторы VT5 и VT6 закрыты, так как разность потенциалов между затворами и истоками по абсолютной величине меньше порогового напряжения отпирания. Отметим, что МОП-транзисторы со встроенным каналом (*n*-типа в данном случае) в открытом состоянии способны пропускать сигналы (ток) в обоих направлениях.



В режиме *записи* на адресную шину (на затворы транзисторов VT5 и VT6) с выхода дешифратора поступает высокий уровень напряжения (логическая 1), подключая триггеры к разрядным шинам 1 и 2, на которые поступают разные уровни напряжения: на шину 1 подаётся логическая 1, а на шину 2 — логический 0. При этом транзистор VT5 открывается, положительный потенциал поступает на затвор транзистора VT2. Транзистор VT2 переходит в открытое состояние, а транзистор VT1 — в закрытое, независимо от его предшествующего состояния. Состоянию триггера с записанной 1 соответствует высокое напряжение на стоке транзистора VT1 и близкое к нулю напряжение на стоке транзистора VT2. В режиме *считывания* на адресную шину с дешифратора поступает логическая 1, а шины 1 и 2 подключаются к входам усилителей считывания, поэтому единичное состояние триггера передаётся на выход ЗУ. В режиме *хранения* на адресную шину поступает низкий уровень напряжения, закрытые транзисторы VT5 и VT6 "изолируют" триггер от разрядных шин, и его состояние остаётся неизменным.

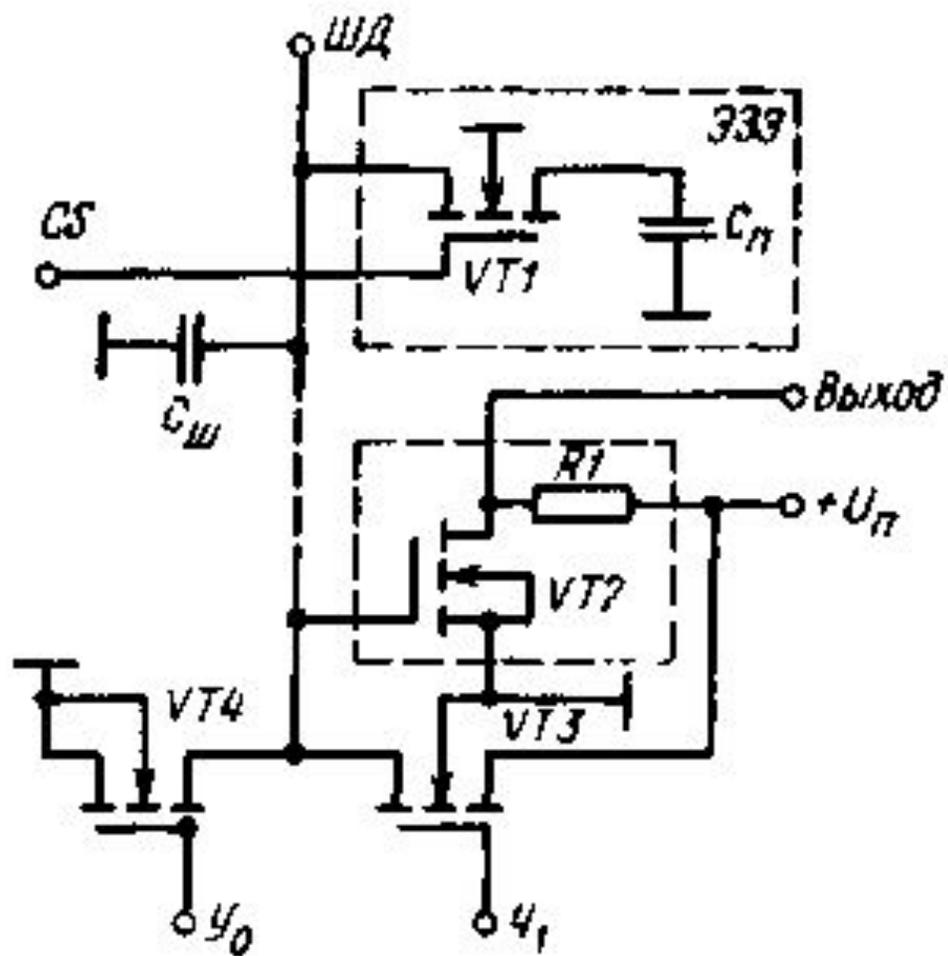


Рис. 21.11. Фрагмент структурной схемы динамического ЗУ

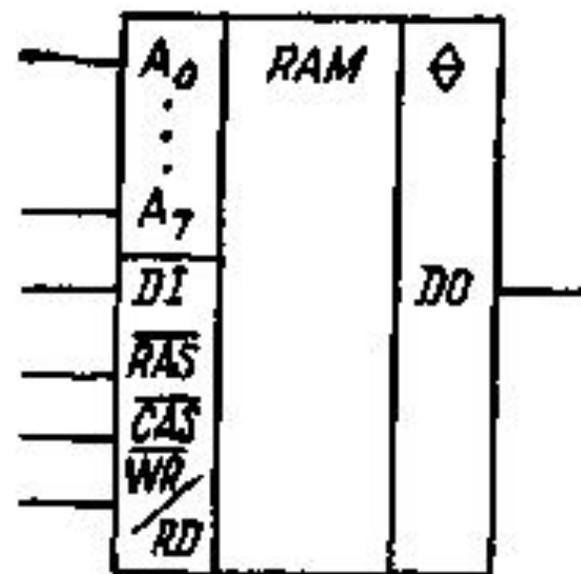
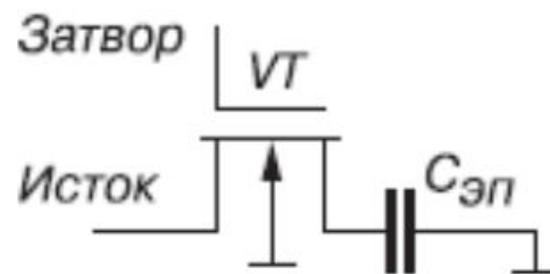


Рис. 21.12. Условное обозначение ИС динамического ОЗУ типа 565РУ5

Элемент памяти ПЗУ должен хранить заранее записанную изготовителем информацию. В этом случае говорят, что программа жёстко "зашита" в ЗУ. Из ПЗУ можно только считывать хранимые там слова, но нельзя вносить новые, стирать и заменять записанные слова другими. Полупроводниковые ПЗУ могут быть *диодными* и *транзисторными*.

На рис. 6.33 изображено ПЗУ, состоящее из диодной матрицы и дешифратора адреса DC. Горизонтальные линии матрицы — адресные, вертикальные — разрядные, с которых можно снимать восьмиразрядные двоичные числа записанные в ПЗУ.



Если, например, с выхода дешифратора DC возбуждается верхняя адресная линия, то сигналы 1 проходят с неё через диоды следующих линий разрядной шины: 8, 6, 4, 2 и 1, т. е. считывается двоичное число 10101011. Аналогично, при возбуждении других линий адресной шины на выходе ПЗУ будут считываться три других двоичных числа. Заметим, что в данном примере диодная матрица является совокупностью элементов ИЛИ: входами элементов являются линии адресной шины, а выходом — одна из линий разрядной шины.

В ПЗУ (рис. 6.34, а) записаны три 4разрядных числа (1001, 0011 и 1010), которые образуются на его выходе при возбуждении соответствующей линии адресной шины, вследствие того, что в соответствии с кодами программы расплавлены импульсами тока перемычки в отдельных эмиттерах транзисторов. Это обеспечивает появление на выходе ПЗУ нулей в линиях разрядной шины, где разрушены эмиттеры.

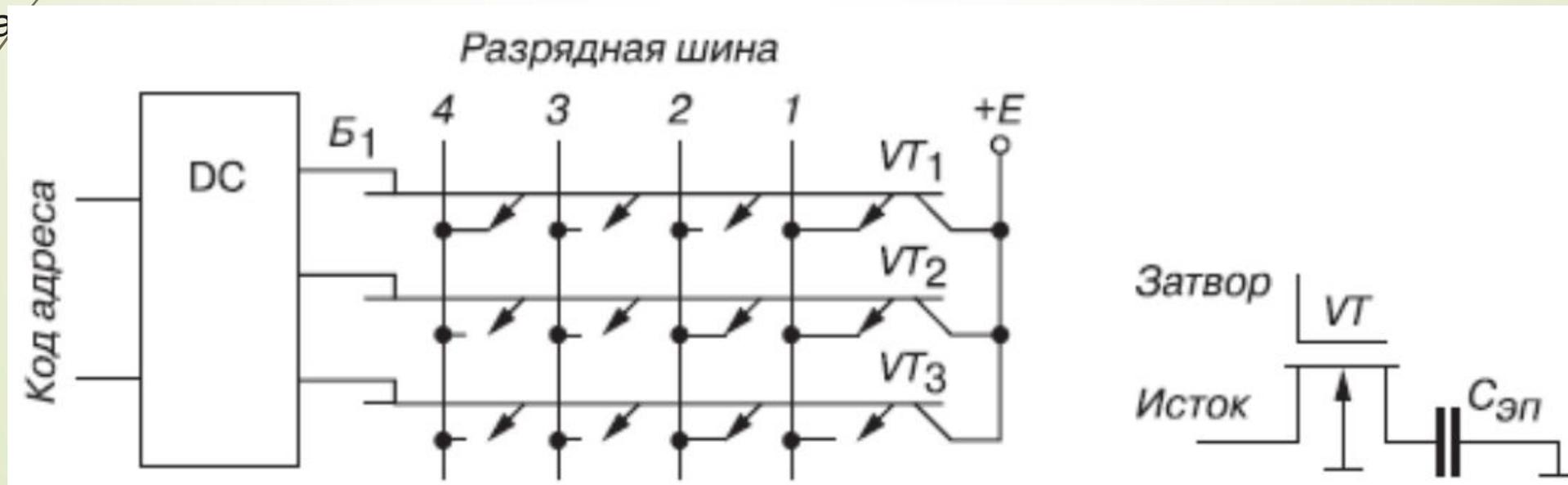
Основой для построения ёмкостного элемента динамической памяти служит МОП транзистор, в котором сток образует одну обкладку конденсатора, а подложка — другую. Диэлектриком между обкладками служит тонкий слой оксида кремния  $\text{SiO}_2$ .

Электрическая схема элемента памяти приведена на рис. 6.34, б. Логическому нулю соответствует нулевое значение напряжения на конденсаторе  $C_{эл}$ , логической единице — высокий уровень напряжения. Структура истокзатворсток выполняет функцию транзисторного ключа. Затвор транзистора  $V_T$  подключают к линии адресной шины, исток — к линии разрядной шины. Параллельно представленной на рис. 6.34, б структуре подключают дополнительные элементы для нормального функционирования элемента.

Ячейки памяти статических ПЗУ выполняются также на *многоэмиттерных* транзисторах (рис. 6.34, а). При возбуждении одной из линий адресной шины отпирается транзистор  $VT_i$ , база  $B_i$  которого присоединена к этой линии. Благодаря этому возбуждаются те линии разрядной шины, к которым присоединены эмиттеры этого транзистора.

При отсутствии напряжения на затворе транзистор  $VT$  заперт, конденсатор  $C_{эп}$  элемента памяти отключен от разрядной шины и находится в режиме хранения информации. Чтобы уменьшить изменение напряжения на элементе памяти при считывании информации используют высокочувствительные дифференциальные усилители с положительной обратной связью — усилители регенерации напряжения до уровня  $+E$  на конденсаторе  $C_{эп}$ .

В связи с небольшим значением ёмкости  $C_{эп}$  элемента памяти заряд в ней не может храниться более 115 мс. Поэтому с помощью устройства управления микропроцессора обеспечен в ЗУ режим принудительной регенерации с периодом повторения цикла, равным обычно 12 мс, посредством специального кварцевого генератора импульсов регенерации, а также синхронизированного тактовыми импульсами счётчика и мультиплексора



Говоря о быстродействии памяти, необходимо помнить, что прежде чем считать информацию, требуется найти ее местоположение в ЗУ. При разработке полупроводниковых ЗУ нашел применение метод произвольного доступа, при котором время выборки постоянно и не зависит от местоположения информации в хранящемся массиве.

- В зависимости от характера использования, выделяют
  - ✱ кэш-память,
  - ✱ оперативную память
  - ✱ внешнюю память (память, предоставляемую буферами).

