

**Память**

## Основная память

- **Память** — это тот компонент компьютера, в котором хранятся программы и данные.
- Также часто встречается термин «запоминающее устройство».
- Без памяти, откуда процессоры считывают и куда записывают информацию, не было бы современных цифровых компьютеров.

# Память

Основной единицей хранения данных в памяти является двоичный разряд, который называется **битом** (может содержать 0 или 1).

□ Двоично-десятичный код (например мэйнфреймы IBM).

Число 1944 в двоично-десятичной и в чисто двоичной системах счисления (в обоих случаях используется 16 бит):

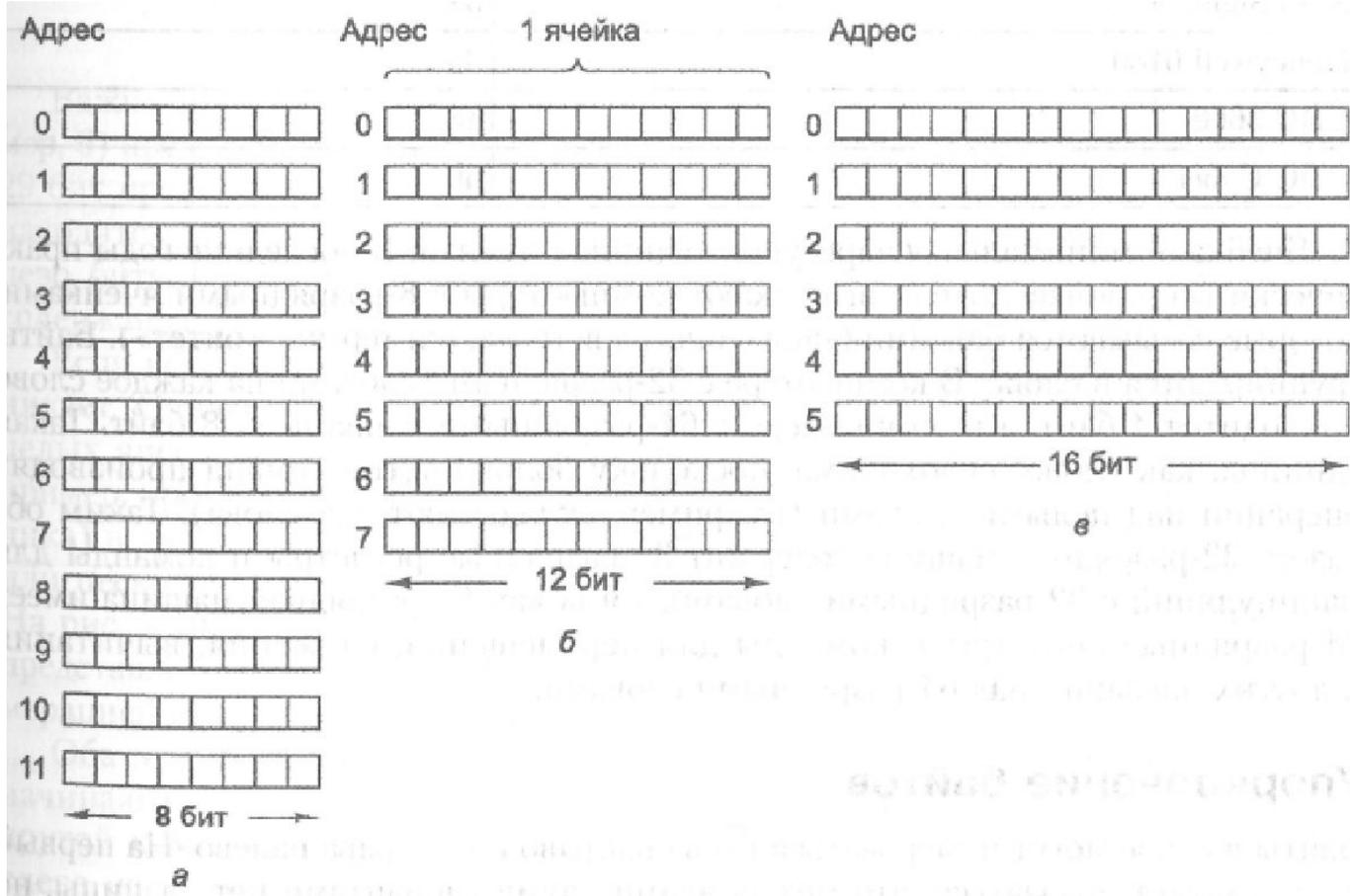
+ двоично-десятичное представление – 0001 1001 0100 0100;

+ двоичное представление – 0000 0111 1001 1000.

В двоично-десятичном представлении 16 бит достаточно для хранения числа от 0 до 9999, то есть доступно всего 10 000 различных комбинаций,

В двоичном представлении те же 16 бит позволяют получить 65 536 комбинаций.

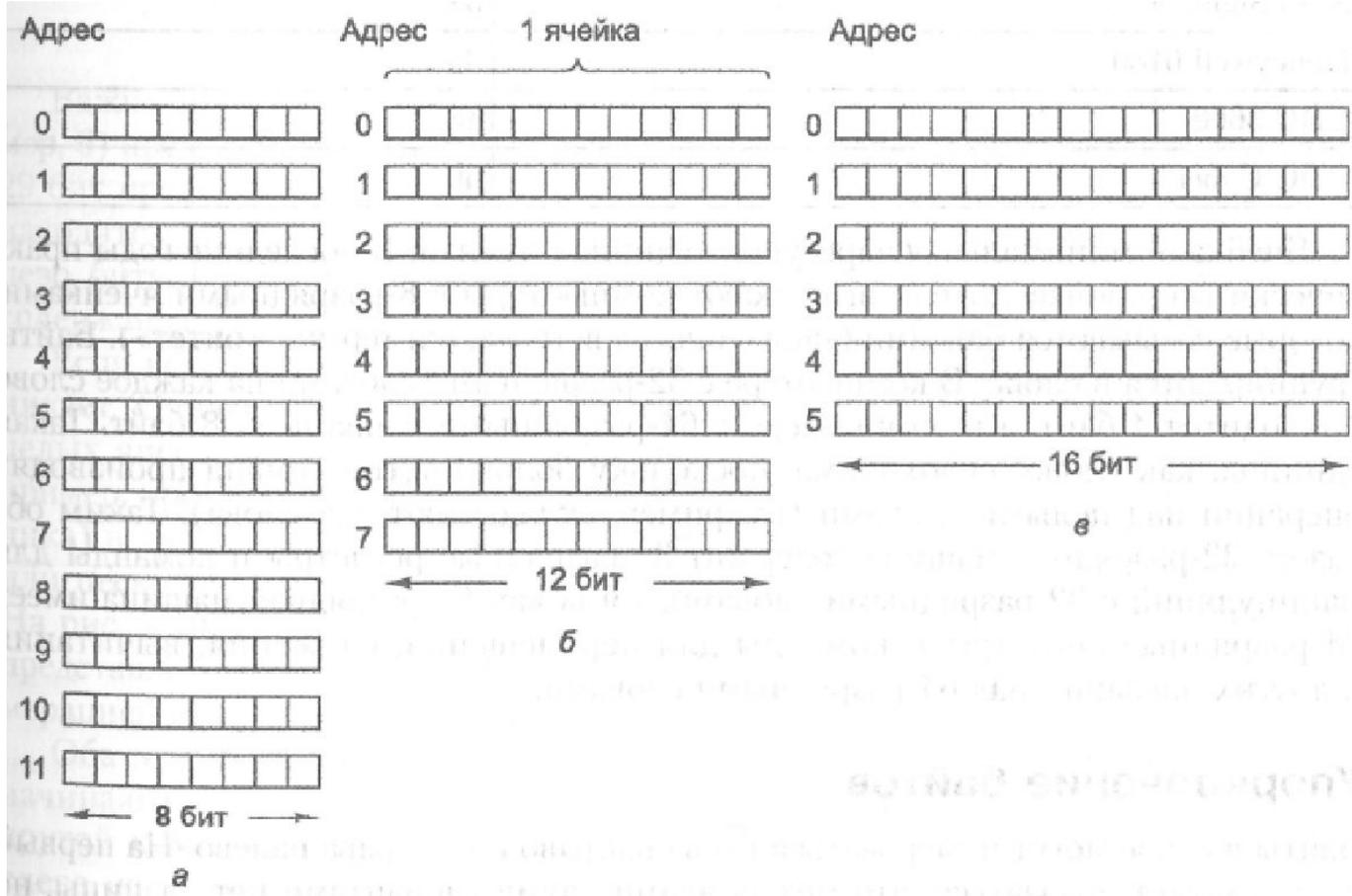
## Три варианта организации 96-разрядной памяти



## Адреса памяти

- Память состоит из **ячеек**
- Каждая ячейка имеет номер, который называется **адресом**.
- По адресу программы могут ссылаться на определенную ячейку.
- Если память содержит  $n$  ячеек, они будут иметь адреса от 0 до  $n-1$ .
- Все ячейки памяти содержат одинаковое число битов. Если ячейка состоит из  $k$  бит, она может содержать любую из  $2^k$  комбинаций.

## Три варианта организации 96-разрядной памяти



## Адреса памяти

- Адреса памяти также выражаются в двоичных числах. Если адрес состоит из  $m$  бит, максимальное число адресуемых ячеек составит  $2^m$
- Число битов в адресе определяет максимальное количество адресуемых ячеек памяти и не зависит от числа битов в ячейке

- Ячейка — минимальная адресуемая единица памяти.
- В последние годы практически все производители выпускают компьютеры с 8-разрядными ячейками, которые называются **байтами** (также иногда встречается термин «**октет**»).
- Байты группируются в **слова**.
- В компьютере с 32-разрядными словами на каждое слово приходится 4 байт, а в компьютере с 64-разрядными словами — 8 байт.
- 32-разрядная машина содержит 32-разрядные регистры и команды для манипуляций с 32-разрядными словами, тогда как 64-разрядная машина имеет 64-разрядные регистры и команды для перемещения, сложения, вычитания и других операций над 64-разрядными словами.

# Упорядочение байтов

Байты в слове могут нумероваться слева направо или справа налево.

Область памяти 32-разрядного компьютера, в котором байты пронумерованы слева направо (как у компьютеров SPARC или мэйнфреймов IBM)



Аналогичная область памяти 32-разрядного компьютера с нумерацией байтов справа налево (как у компьютеров Intel)

Память с **прямым порядком следования байтов** (big endian) (нумерация начинается с высшего порядка)

Память с **обратным порядком следования байтов** (little endian)

## Jim Smith, 21 год, отдел 260» (1 X 256 + 4 = 260)

Обратный  
порядок  
следования  
байтов

0	J	I	M	
4	S	M	I	T
8	H	0	0	0
12	0	0	0	21
16	0	0	1	4

*a*

Прямой  
порядок  
следования  
байтов

	M	I	J
T	I	M	S
0	0	0	H
0	0	0	21
0	0	1	4

*b*

Передача  
с преобразованием  
прямого порядка  
следования  
байтов в обратный

0		M	I	J
4	T	I	M	S
8	0	0	0	H
12	21	0	0	0
16	4	1	0	0

*в*

Передача  
с перестановкой  
байтов

0	J	I	M	
4	S	M	I	T
8	H	0	0	0
12	0	0	0	21
16	0	0	1	4

*г*

возраст при передаче  
 $21 \times 2^{24}$

Решение проблемы —  
использование  
программы, которая бы  
инвертировала байты в  
слове после создания  
копии.

Запись данных о сотруднике для машины с прямым порядком следования байтов (*a*); та же запись для машины с обратным порядком следования байтов (*b*); результат передачи записи с машины с прямым порядком следования байтов на машину с обратным порядком следования байтов (*в*); результат перестановки байтов в предыдущем случае (*г*)

**Отсутствие стандарта упорядочивания байтов является серьезной проблемой при обмене информацией между разными машинами.**

# Код исправления ошибок

- В памяти компьютера из-за всплесков напряжения и по другим причинам время от времени могут возникать ошибки.
- Для борьбы с ними используются специальные коды, умеющие обнаруживать и исправлять ошибки. В этом случае к каждому слову в памяти особым образом добавляются дополнительные биты. Когда слово считывается из памяти, эти дополнительные биты проверяются, что и позволяет обнаруживать ошибки

# Код исправления ошибок

Пример: 10001001 и 10110001

- Предположим, что слово состоит из  $m$  бит данных, к которым мы дополнительно прибавляем  $r$  бит (контрольных разрядов).
- Пусть общая длина слова составит  $n$  бит (то есть  $n = m + r$ ).
- Единицу из  $n$  бит, содержащую  $m$  бит данных и  $r$  контрольных разрядов, называют **КОДОВЫМ СЛОВОМ**

- Определим количество отличающихся битов (исключающее ИЛИ) и сосчитаем число битов со значением 1 в полученном результате
- Число битовых позиций, по которым отличаются два слова, называется *интервалом Хэмминга* [Hamming, 1950]
- Если интервал Хэмминга для двух слов равен  $d$  значит, достаточно  $d$  одноразрядных ошибок, чтобы превратить одно слово в другое

# Метод Ричарда Хэмминга

Закодируем в качестве примера слово из 4 бит 1100 в секторы  $AB$ ,  $ABC$ ,  $AC$  и  $BC$ , по одному биту в каждом секторе (в алфавитном порядке) используя диаграмму Венна



Кодирование числа 1100 (*a*); добавляются биты четности (*б*); ошибка в секторе AC (*в*)



- Добавим бит четности к каждому из трех пустых секторов, чтобы сумма битов в каждом из трех кругов,  $A$ ,  $B$ , и  $C$ , получилась четной
- Предположим, что бит в секторе  $AC$  изменился с 0 на 1
- Компьютер обнаруживает, что круги  $A$  и  $C$  являются нечетными. Единственный способ исправить ошибку, изменив только один бит, — возвращение значения 0 биту в секторе  $AC$ .

Таким способом компьютер может исправлять одиночные ошибки автоматически.

# Метод Ричарда Хэмминга

Алгоритм Хэмминга при создании кодов исправления ошибок для слов любого размера:

- В коде Хэмминга к слову, состоящему из  $m$  бит, добавляются  $r$  бит четности, при этом образуется слово длиной  $m + r$  бит.
- Все биты, номера которых — степени двойки, являются битами четности, остальные используются для данных.

Например, к 16-разрядному слову нужно добавить 5 бит четности. Биты с номерами 1, 2, 4, 8 и 16 — биты четности, все остальные — биты данных. Всего слово содержит 21 бит (16 бит данных и 5 бит четности).

# Метод Ричарда Хэмминга

Ниже указаны позиции проверки для каждого бита четности:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
0	0	1	0	1	1	1	0	0	0	0	0	1	0	1	0	0	1	1	1	0	
x		x		x		x		x		x		x		x		x		x		x	1
	x	x			x	x			x	x			x	x			x	x			2
			x	x	x	x					x	x	x	x					x	x	4
							x	x	x	x	x	x	x								8
															x	x	x	x	x	x	16

Код Хэмминга для 16-разрядного слова 1111000010101110

(до вычисления самих контрольных бит, мы присвоили им значение «0».)

Каждый бит четности позволяет проверять определенные битовые позиции. Общее число битов со значением 1 в проверяемых позициях должно быть четным.

В общем случае бит  $b$  проверяется битами  $b_1, b_2, \dots, b_j$ , такими что  $b_1 + b_2 + \dots + b_j = b$ . Например, бит 5 проверяется битами 1 и 4, поскольку  $1+4 = 5$ . Бит 6 проверяется битами 2 и 4, поскольку  $2 + 4 = 6$  и т. д.

# Метод Ричарда Хэмминга

Вычисляем значение каждого контрольного бит: общее число битов со значением 1 в проверяемых позициях должно быть четным.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
0	0	1	0	1	1	1	0	0	0	0	0	1	0	1	1	0	1	1	1	0	
x		x		x		x		x		x		x		x		x		x		x	1
	x	x			x	x			x	x			x	x			x	x			2
			x	x	x	x					x	x	x	x					x	x	4
							x	x	x	x	x	x	x	x							8
															x	x	x	x	x	x	16

Пусть бит 5 изменит значение: **001011100000101101110** => **001001100000101101110**

Заново вычисляем все контрольные биты: **101101100000101101110**

Сложив номера позиций неправильных контрольных бит (1 + 4 = 5) получаем позицию ошибочного бита

# Кэш-память

Процессоры всегда работали быстрее, чем память:

- Разработчики процессоров помещают все больше и больше транзисторов на микросхемы, создают конвейерные и суперскалярные архитектуры.
- Разработчики памяти обычно используют новые технологии для увеличения емкости, а не быстродействия, что делает разрыв еще большим.

На практике такое несоответствие в скорости работы приводит к тому, что когда процессор обращается к памяти, проходит несколько машинных циклов, прежде чем он получит запрошенное слово. Чем медленнее работает память, тем дольше процессору приходится ждать, тем больше циклов проходит.

# Кэш-память

## Пути решения проблемы:

- ✓ Считывать информацию из памяти и при этом продолжать выполнение команд, но если какая-либо команда попытается использовать слово до того, как оно считано из памяти, процессор должен приостановить работу => простой процессора.
  - ✓ Сконструировать машину, которая не приостанавливает работу, но следит, чтобы программы-компиляторы не использовали слова до того, как они считаны из памяти => задержка на уровне программного обеспечения.
- ✓ Создать память которая работает так же быстро, как процессор. Память можно разместить прямо на микросхему процессора => сильно увеличивает размер процессора, а следовательно и стоимость.

# Кэш-память

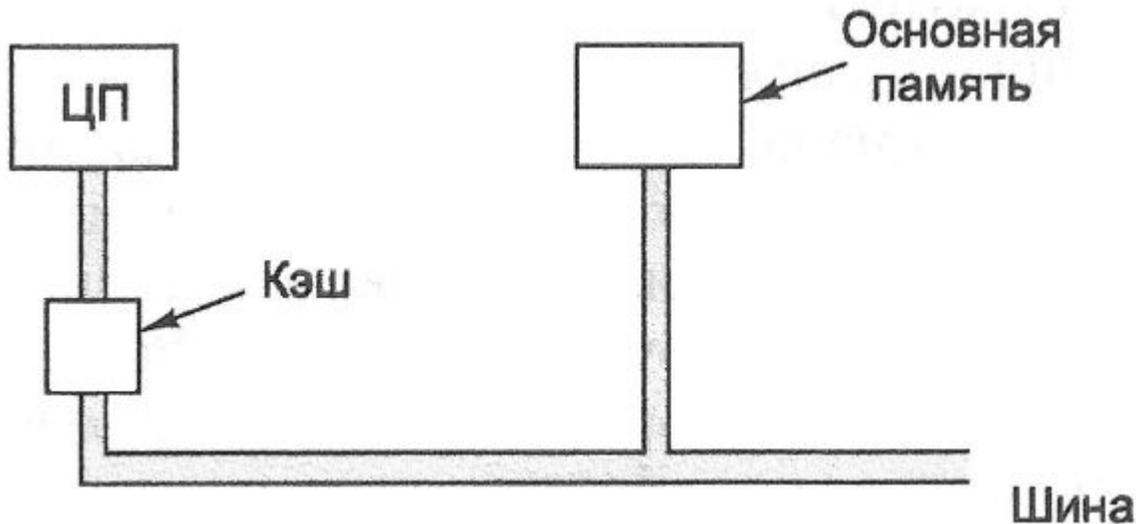
Память небольшого объема с высокой скоростью работы называется **кэш-памятью**

Основная идея кэш-памяти: в ней находятся слова, которые чаще всего используются.

- Если процессору нужно какое-нибудь слово, сначала он обращается к кэш-памяти. Только в том случае, если слова там нет, он обращается к основной памяти.

# Принцип локальности

Когда определенное слово вызывается из памяти, оно вместе с соседними словами переносится в кэш-память, что позволяет при очередном запросе быстро обращаться к следующим словам.



Если слово считывается или записывается  $k$  раз, компьютеру требуется сделать одно обращение к медленной основной памяти и  $k - 1$  обращений к быстрой кэш-памяти. Чем больше  $k$ , тем выше общая производительность.

Общее устройство процессора, кэш-памяти и основной памяти

$c$  – время доступа к кэш-памяти,

$m$  – время доступа к основной памяти

$h$  – **коэффициент кэш-попаданий** (hit ratio), который показывает соотношение числа обращений к кэш памяти и общего числа всех обращений к памяти (в нашем примере  $h = (k - 1)/k$ ).

Выделяют **коэффициент кэш-промахов** (miss ratio), равный  $1 - h$ .

Таким образом, мы можем вычислить среднее время доступа:

$$\text{Среднее время доступа} = c + (1 - h) m.$$

- $h \rightarrow 1$
- $h \rightarrow 0$

- Основная память и кэш-память делятся на блоки фиксированного размера с учетом принципа локальности.
- Блоки внутри кэш-памяти обычно называют **строками кэша** (cache lines).
- При кэш-промахе из основной памяти в кэш-память загружается вся строка, а не только необходимое слово.

Например, если строка состоит из 64 байт, обращение к адресу 260 влечет за собой загрузку в кэш-память всей строки (байты с 256 по 319) на случай, если через некоторое время понадобятся другие слова из этой строки.

- Такой путь обращения к памяти более эффективен, чем вызов каждого слова по отдельности, потому что однократный вызов  $k$  слов происходит гораздо быстрее, чем вызов одного слова  $k$  раз.
- Многие компьютеры (даже 32-разрядные) способны передавать 64 и 128 бит параллельно за один цикл шины.

- Объем кэш-памяти?

Чем больше объем, тем лучше работает память, но тем дороже она стоит.

- Размер строки кэша?

Кэш-память объемом 16 Кбайт можно разделить на 1024 строки по 16 байт, 2048 строк по 8 байт и т. д.

- Механизм организации кэш-памяти, то есть то, как она определяет, какие именно слова находятся в ней в данный момент?
- Должны ли команды и данные находиться вместе в общей кэш-памяти?
- Количество блоков кэш-памяти?

В настоящее время очень часто кэш-память первого уровня располагается прямо на микросхеме процессора, кэш-память второго уровня – не на самой микросхеме, но в корпусе процессора, а кэш-память третьего уровня – еще дальше от процессора.

**Объединенная кэш-память (unified cache)**, в которой хранятся и данные, и команды. В этом случае вызов команд и данных автоматически уравнивается.

**Разделенная кэш-память (split cache)**, когда команды хранятся в одной кэш-памяти, а данные — в другой. Такая архитектура называется **гарвардской (Harvard architecture)**

### **Разделенная кэш-память**

- больше подходит к широко используемым в настоящее время конвейерным архитектурам, т.к. при конвейерной организации обращения и к командам, и к данным (операндам) должны осуществляться одновременно.
- позволяет осуществлять параллельный доступ, а общая память — нет.
- поскольку команды обычно не меняются во время выполнения программы, содержание кэша команд не приходится записывать обратно в основную память.

# Оперативная память

Оперативную память часто называют оперативным запоминающим устройством (**ОЗУ**), или памятью с произвольным доступом (Random Access Memory (**RAM**))

- Энергозависимая часть системы компьютерной памяти, в которой во время работы компьютера хранится выполняемый машинный код (программы), а также входные, выходные и промежуточные данные, обрабатываемые процессором.
- «Энергозависимая» - оперативная память компьютера обнуляется при выключении питания
- Передача данных в оперативную память процессором производится непосредственно, либо через сверхбыструю память (кэш)
- Два типа ОЗУ: статическое и динамическое

## Кэш-память

- построена на триггерах, которые, в свою очередь, состоят из транзисторов
- группа транзисторов входящая в триггер занимает гораздо больше места, чем конденсаторы, соответственно объемы такой памяти ограничены

## Статическое ОЗУ (SRAM):

- основано на триггерах
- информация в ОЗУ сохраняется на протяжении всего времени, пока к нему подается питание: секунды, минуты, часы и даже дни;
- работает очень быстро (обычно время доступа составляет несколько наносекунд, часто используется в качестве кэш-памяти второго уровня)

## Динамическое ОЗУ

### (Dynamic RAM, DRAM):

- представляет собой массив ячеек, каждая из которых содержит транзистор и крошечный конденсатор
- конденсаторы могут быть заряженными и разряженными, что позволяет хранить нули и единицы
- электрический заряд имеет тенденцию исчезать, каждый бит в динамическом ОЗУ должен **обновляться** (перезаряжаться) каждые несколько миллисекунд, чтобы предотвратить утечку данных
- об обновлении заботится внешняя логика
- требует более сложного сопряжения, чем статическое ОЗУ, **но!** имеет больший объем

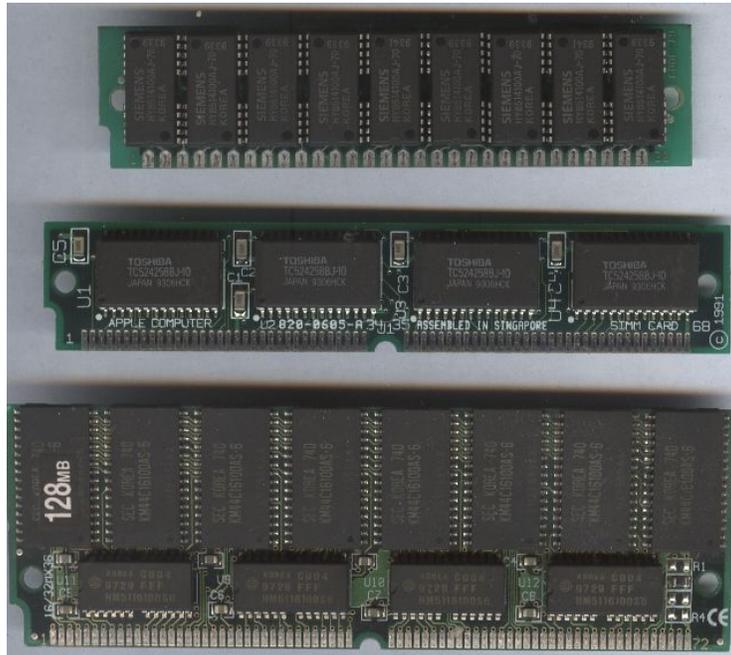
## Динамическое ОЗУ

- имеет очень высокую плотность записи (много битов на одну микросхему) => основная память почти всегда строится на основе динамических ОЗУ
- работает очень медленно (время доступа занимает десятки наносекунд)

Сочетание кэш-памяти на основе статического ОЗУ и основной памяти на основе динамического ОЗУ соединяет в себе преимущества обоих устройств

# Сборка модулей памяти и их типы

- **SIMM** (Single Inline Memory Module) - модуль памяти с односторонним расположением выводов)



один краевой разъем с 72 контактами  
скорость передачи данных за один тактовый  
цикл 32 бит

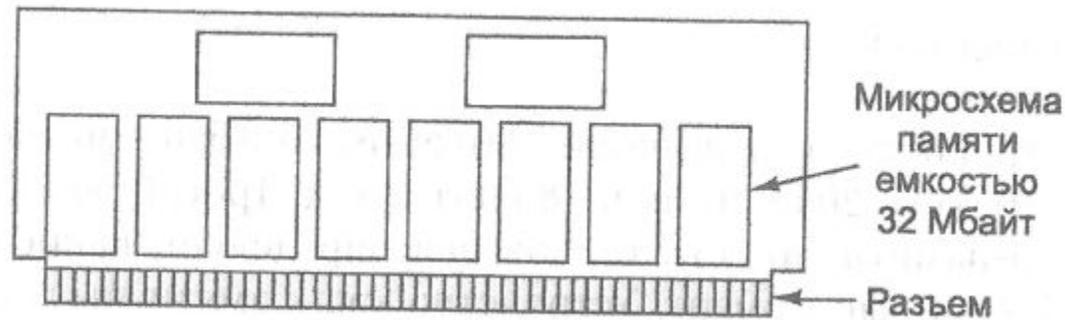
- **DIMM** (Dual Inline Memory Module) - модуль памяти с двухсторонним расположением выводов)



два краевых разъема (по одному на каждой  
стороне платы) с 120 контактами (всего 240)  
скорость передачи данных - 64 бит за цикл

# Сборка модулей памяти и их типы

В настоящее время наиболее распространенными являются **DDR3 DIMM** - третья версия двух-скоростных модулей памяти



В портативных компьютерах обычно используется модуль DIMM меньшего размера, который называется **SO-DIMM** (Small Outline DIMM)

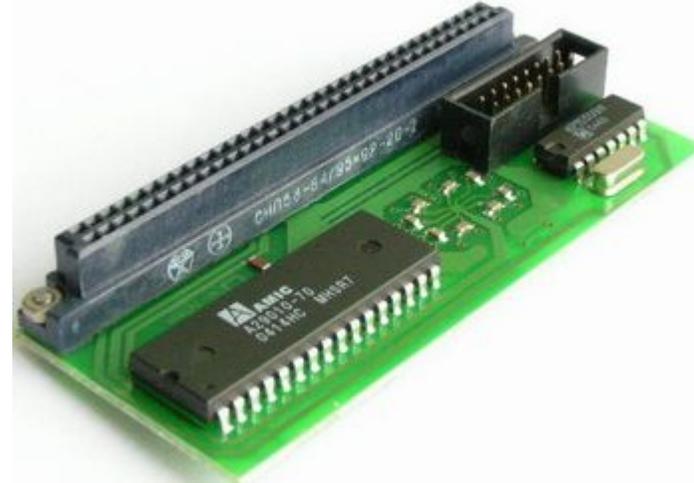
4-гигабайтный модуль DIMM с восемью 256-мегабайтными микросхемами с каждой стороны.  
Другая сторона выглядит аналогично

DIMM содержат 8 микросхем по 256 Мбайт каждая. Таким образом, весь модуль вмещает 2 Гбайт информации. Во многих компьютерах предусматривается возможность установки четырех модулей; следовательно при использовании модулей по 2 Гбайт общий объем памяти достигает 8 Гбайт (и более при использовании модулей большей емкости).

# ПЗУ

Дополнительно к оперативной памяти многие компьютеры оснащены небольшой по объему неизменяемой памятью с произвольным доступом — постоянным запоминающим устройством (**ПЗУ**), оно же память, предназначенная только для чтения (Read Only Memory (**ROM**)).

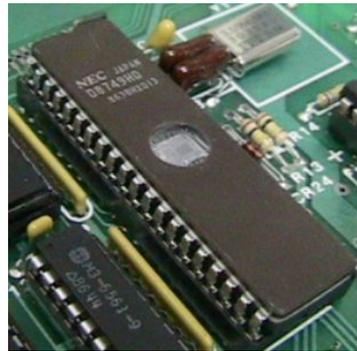
- ПЗУ не утрачивает своего содержимого при отключении питания, то есть является энергонезависимой.
- ПЗУ программируется на предприятии-изготовителе и впоследствии не подлежит изменению.



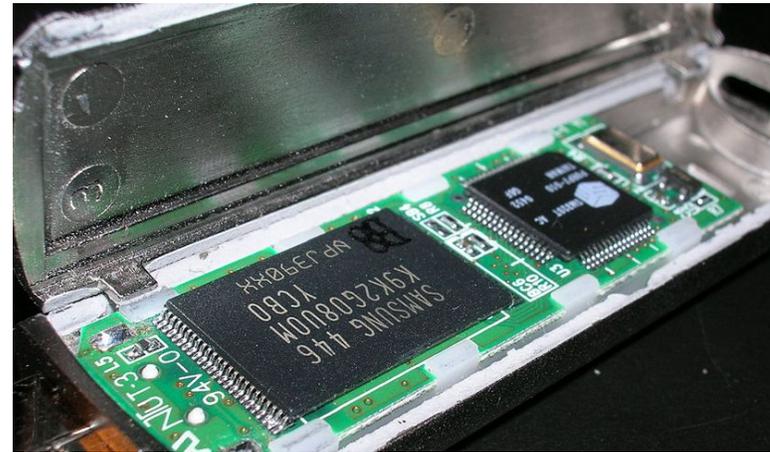
В ПЗУ размещается начальный загрузчик, используемый для их запуска. ПЗУ предназначена для осуществления низкоуровневого управления устройством, поэтому ей оснащаются некоторые контроллеры ввода-вывода

## Другие разновидности энергонезависимой памяти

Электрически стираемые программируемые постоянные запоминающие устройства (**ЭСПЗУ**), они же **EEPROM** (Electrically Erasable PROM).



Запись в них занимает на несколько порядков больше времени, чем запись в ОЗУ, поэтому они используются для тех же целей, что и ПЗУ.



Флеш-память – используется как носитель информации в портативных электронных устройствах. Она служит «пленкой» в цифровых фотоаппаратах и «диском» в переносных музыкальных плеерах и т.д.

ПЗУ стирается под воздействием ультрафиолетовых лучей или под действием электронного импульса, стирается по байтам. Флэш-память стирается и записывается блоками

# Вспомогательная память

## Иерархическая структура памяти



По мере продвижения сверху вниз по иерархии меняются три параметра:

- увеличивается время доступа
- растет объем памяти
- увеличивается количество битов, которое вы получаете за один доллар.

# Магнитные (жесткие) диски

- Жесткий диск на два порядка дешевле, чем ОЗУ, а его емкость зачастую на два порядка выше, но время произвольного доступа к данным примерно на три порядка медленнее.
- Является механическим устройством.
- Состоит из одной или нескольких металлических пластин, вращающихся со скоростью 5400, 7200, 10 800 и более оборотов в минуту.
- Механический привод поворачивается на определенный угол над пластинами, подобно звукоснимателю старого проигрывателя виниловых пластинок на 33 оборота в минуту.

- Информация записывается на диск в виде последовательности концентрических окружностей.
- В каждой заданной позиции привода каждая из головок может считывать кольцеобразный участок, называемый дорожкой. Из совокупности всех дорожек в заданной позиции привода составляется цилиндр.
- Каждая дорожка поделена на определенное количество секторов, обычно по 512 байт на сектор. На современных дисках внешние цилиндры содержат больше секторов, чем внутренние. Перемещение привода с одного цилиндра на другой занимает около 1 мс. Перемещение к произвольно выбранному цилиндру обычно занимает от 5 до 10 мс в зависимости от конкретного накопителя. Когда привод расположен над нужной дорожкой, накопитель должен выждать, когда нужный сектор попадет под головку.
- Это приводит к возникновению еще одной задержки от 5 до 10 мс в зависимости от скорости вращения диска. После попадания требуемого сектора под головку производится операция чтения или записи со скоростью от 50 Мбайт/с (для низкоскоростных дисков) до 160 Мбайт/с (для высокоскоростных).

# Магнитные (жесткие) диски



Магнитный диск состоит из одной или нескольких алюминиевых поверхностей, покрытых магнитным слоем. Изначально их диаметр составлял 50 см, сейчас — от 3 до 9 см, у портативных компьютеров — меньше 3 см, причем это значение продолжает уменьшаться. Головка диска, содержащая индукционную катушку, движется над поверхностью диска, опираясь на воздушную подушку. Когда через головку проходит положительный или отрицательный ток, он намагничивает поверхность под головкой. При этом магнитные частицы намагничиваются направо или налево в зависимости от полярности тока. Когда головка проходит над намагниченной областью, в ней (в головке) возникает положительный или отрицательный ток, что дает возможность считывать записанные ранее биты. Поскольку диск вращается под головкой, поток битов может записываться, а потом считываться.

На сегодняшний момент диски имеют около 50 000 дорожек на сантиметр, то есть ширина каждой дорожки составляет около 200 нанометров (1 нанометр = 1/1 000 000 мм).

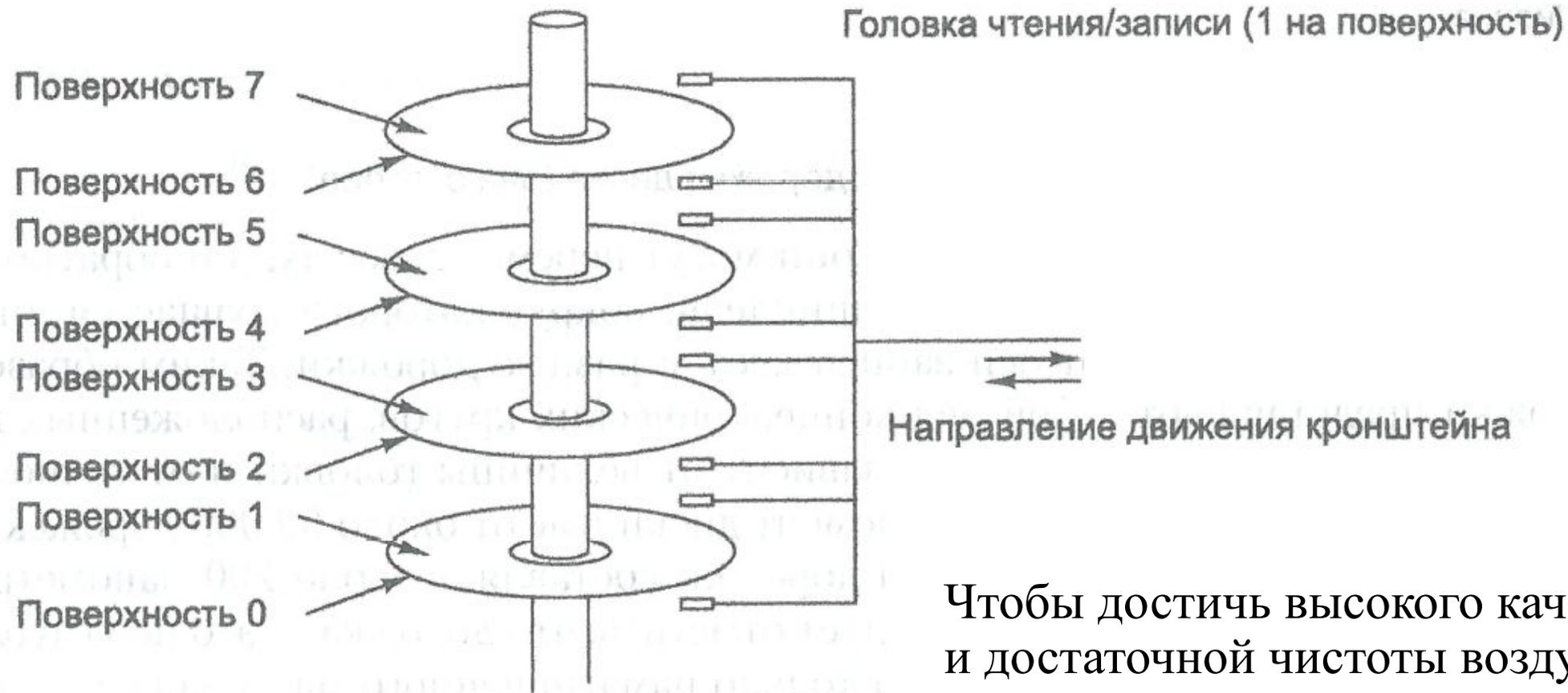
Дорожка — это не углубление на поверхности диска, а просто кольцо намагниченного материала, которое отделяется от других дорожек небольшими пограничными областями.

- **Дорожкой** называется круговая последовательность битов, записанных на диск за его полный оборот.
- Каждая дорожка делится на **секторы** фиксированной длины. Каждый сектор обычно содержит 512 байт данных.
- Перед данными располагается **преамбула** (preamble), которая позволяет головке синхронизироваться перед чтением или записью.
- После данных идет код исправления ошибок (Error-Correcting Code, ECC), в качестве которого используется код Хэмминга или чаще **код Рида-Соломона**, позволяющий исправлять множественные ошибки, а не только одиночные.
- Между соседними секторами находится **межсекторный интервал**.

Многие производители указывают размер неформатированного диска (как будто каждая дорожка содержит только данные), хотя честнее было бы указывать вместимость форматированного диска, на котором не учитываются преамбулы, ECC-коды и межсекторные интервалы. Емкость форматированного диска обычно на 15 % меньше неформатированного.

# Винчестеры

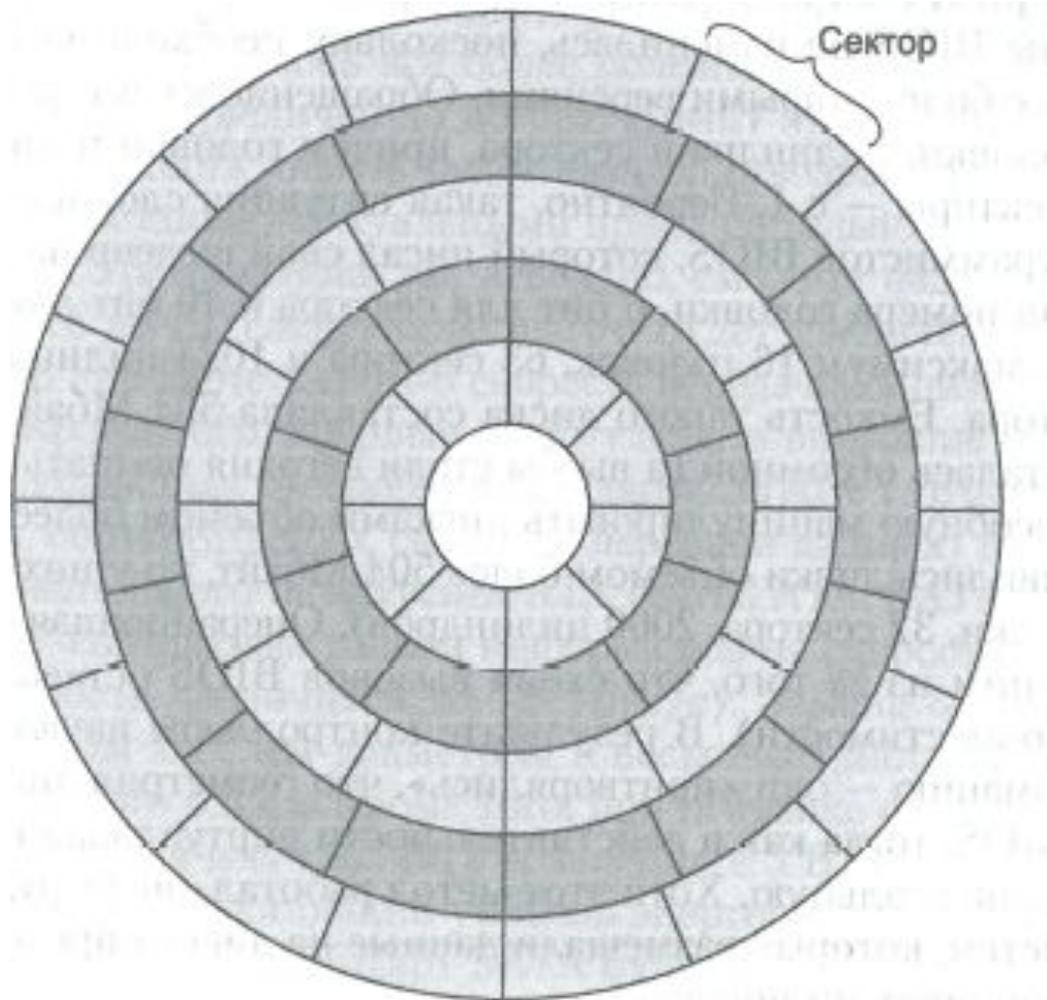
## Винчестер с четырьмя дисками



Чтобы достичь высокого качества поверхности и достаточной чистоты воздуха, диски герметично закрываются. Такие диски называются **винчестерами («жесткими дисками»)**.

- Совокупность дорожек, расположенных на одном расстоянии от центра, называется **цилиндром**.
- В современных моделях дисков для ПК устанавливается от 1 до 12 пластин, содержащих от 12 до 24 рабочих поверхностей.
- На одной пластине современных высокопроизводительных дисков может храниться до 1 Тбайт данных.
- Чтобы считать или записать сектор, головка должна переместиться на нужное расстояние от оси. Этот процесс называется **позиционированием**.
- Когда головка помещается на нужное расстояние от центра, выжидается некоторое время (оно называется **временем ожидания сектора**), пока нужный сектор не окажется под головкой.

*Время передачи информации определяется в основном временем поиска и временем ожидания сектора.*



Раньше при производстве дисков изготовители создавали максимально возможную плотность записи на внутренней дорожке, и при продвижении от центра диска плотность записи постепенно снижалась.

В настоящее время цилиндры делятся на зоны (на диске их обычно от 10 до 30). При продвижении от центра диска число секторов на дорожке в каждой зоне возрастает. Это усложняет структуру информации на дорожке, но зато повышает емкость диска, что считается более важным. Все секторы имеют одинаковый размер.

**Контроллер** - микросхема, которая управляет диском

Задачи контроллера:

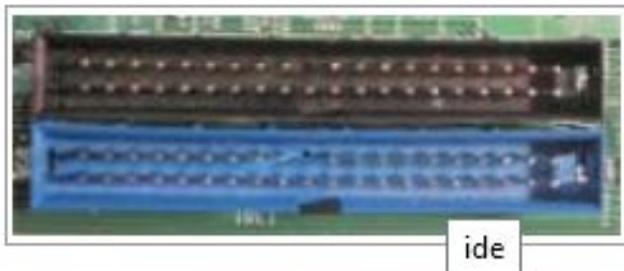
- получение от программного обеспечения таких команд, как READ, WRITE и FORMAT (то есть запись всех преамбул);
- управление перемещением кронштейна;
- обнаружение и исправление ошибок;
- преобразование байтов, считываемых из памяти, в непрерывный поток битов, и наоборот.
- Некоторые контроллеры производят буферизацию и кэширование нескольких секторов на случай их дальнейшего использования, а также пропускают поврежденные секторы.

# IDE-диски (стандарт проектирования дисков)

- Диск Seagate на 10 Мбайт, управляемый контроллером Хебес на встроенной карте.
- 80-х годах устройства **IDE** (Integrated Drive Electronics — **устройство со встроенным контроллером**) контроллер стал встраиваться в материнскую плату, объем 504 Мбайт.
- Стандарт **EIDE** (Extended IDE - **усовершенствованные устройства со встроенным контроллером**), поддерживающие дополнительную схему адресации **LBA (Logical Block Addressing** линейная адресация блоков).
- Стандарт интерфейса диска **ATA-3**
- Стандарт **ATAPI-4** (ATA Packet Interface - пакетный интерфейс АТА), отличался скоростью 33 Мбит/с
- Стандарт **ATAPI-7**, появилась спецификация последовательного интерфейса АТА (**Serial ATA, SATA**), позволившего передавать через 7-контактный разъем информацию на скоростях от 150 Мбит/с

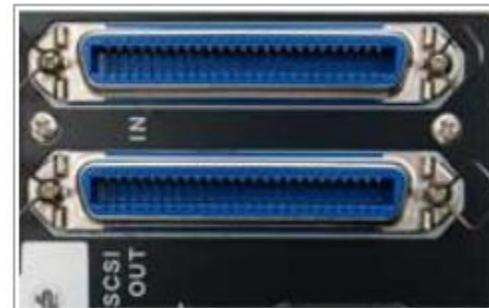
# SCSI-диски

- **SCSI**-диски с точки зрения расположения цилиндров, дорожек и секторов не отличаются от IDE-дисков, но они имеют другой интерфейс и более высокую скорость передачи данных.
- **SCSI** (Small Computer System Interface - интерфейс малых вычислительных систем).



**IDE**

**SCSI** — это не просто интерфейс жесткого диска, но и шина, к которой могут подсоединяться SCSI-контроллер и до семи дополнительных устройств. Ими могут быть один или несколько жестких SCSI-дисков, дисководы CD-ROM, сканеры, накопители на магнитной ленте и другие периферийные устройства. Каждое устройство имеет свой идентификационный код от 0 до 7 (до 15 для 16-разрядных версий). У каждого устройства есть два разъема: один — входной, другой — выходной.



**SCSI**

# RAID-массивы

## **Redundant Array of Inexpensive Disks** (избыточный массив недорогих дисков)

Рядом с компьютером (обычно большим сервером) устанавливается бокс с дисками, контроллер диска замещается RAID-контроллером, данные копируются в RAID-массив, а затем производятся обычные действия.

Поскольку SCSI-диски обладают высокой производительностью при довольно низкой цене и при этом один контроллер может управлять несколькими дисками (до 7 дисков на 8-разрядных моделях SCSI и до 15 на 16-разрядных).

Большинство RAID-устройств состоит из:

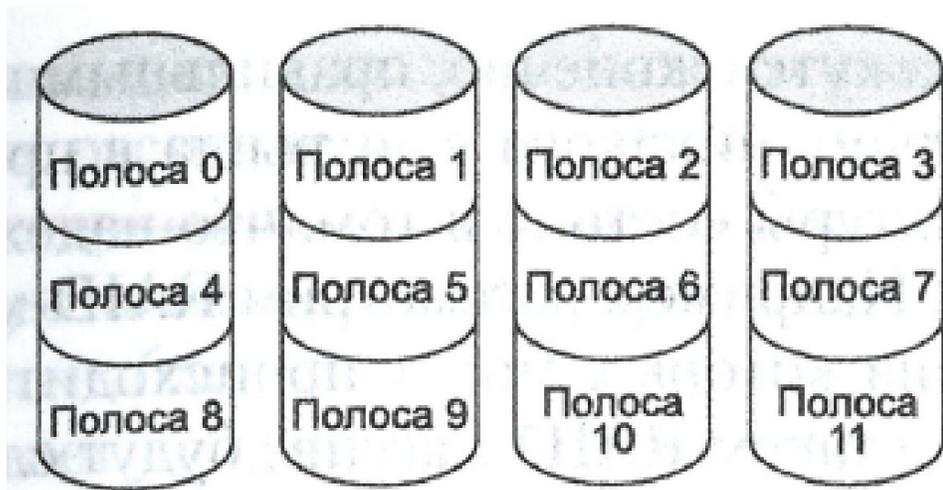
- ✓ SCSI-контроллера, предназначенного для управления RAID-массивом;
- ✓ бокса SCSI-дисков, которые операционная система воспринимает как один большой диск.

## Достоинства RAID-систем:

- программное обеспечение воспринимает RAID-массив как один большой диск;
- данные на всех дисках RAID-массива распределены по дискам таким образом, чтобы можно было осуществлять **параллельные операции**.

Существуют 6 разных вариантов организации дисков с разными характеристиками надежности и производительности, называемых «уровнями»

## RAID-массив уровня 0



RAID-массив с четырьмя

дисками

В RAID-массиве уровня 0 полосы последовательно записываются по кругу, это называется **распределением данных** (striping) по дискам.

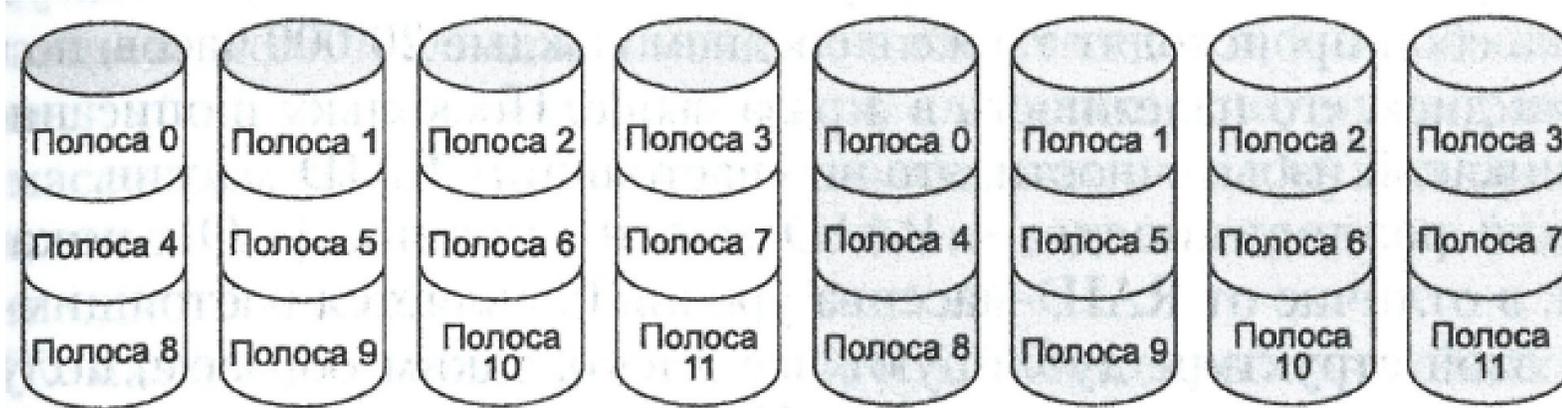
Если в запросе требуется задействовать полос больше, чем дисков в RAID-массиве, то некоторые диски получают по несколько запросов, и как только такой диск завершает выполнение первого запроса, он приступает к следующему.

**Хорошо работает с большими запросами. Чем больше запрос – тем лучше!**

**Если небольшие порции данных – то пропадает эффект параллелизма.**

- Представляет собой виртуальный диск, разделенный на полосы (strips) по  $k$  секторов каждая, при этом секторы с 0 по  $k - 1$  занимают полосу 0, секторы с  $k$  по  $2k - 1$  — полосу 1 и т. д. Для  $k = 1$  каждая полоса — это сектор, для  $k = 2$  каждая полоса — это два сектора и т. д.

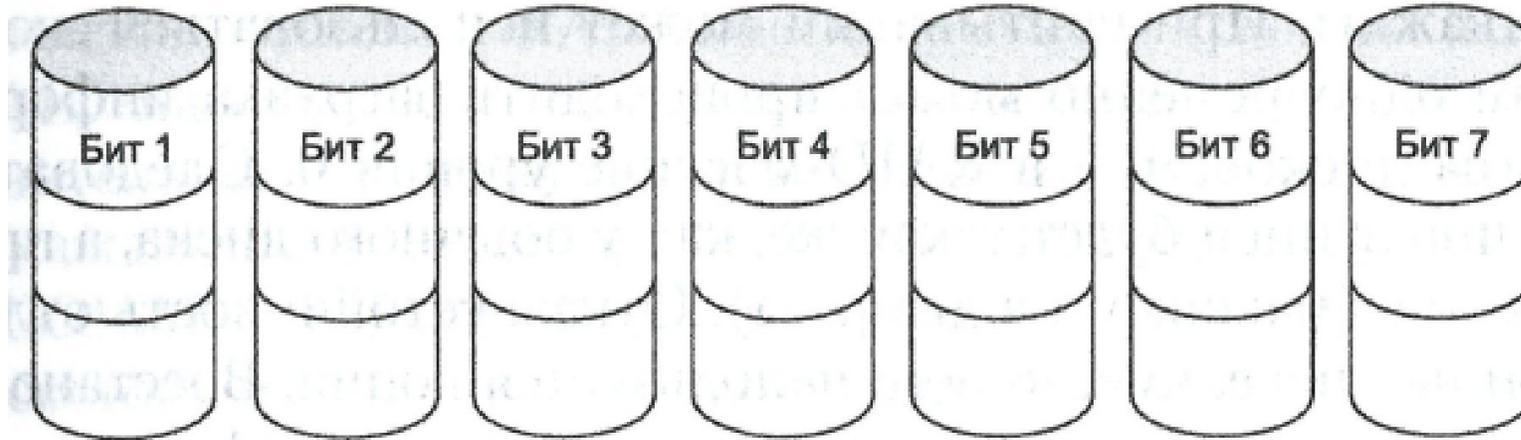
# RAID-массив уровня 1



- В этой структуре дублируют все диски, т. о, получается 4 исходных диска и 4 резервные копии.

- При записи информации каждая полоса записывается дважды. При считывании может использоваться любая из двух копий.
- Производительность при записи будет такая же, как у обычного диска, а при считывании — гораздо выше (максимум в два раза).
- Отказоустойчивость отличная: если происходит сбой на диске, вместо него используется копия. Восстановление состоит просто в установке нового диска и копировании всей информации с резервной копии на него.

## RAID-массив уровня 2

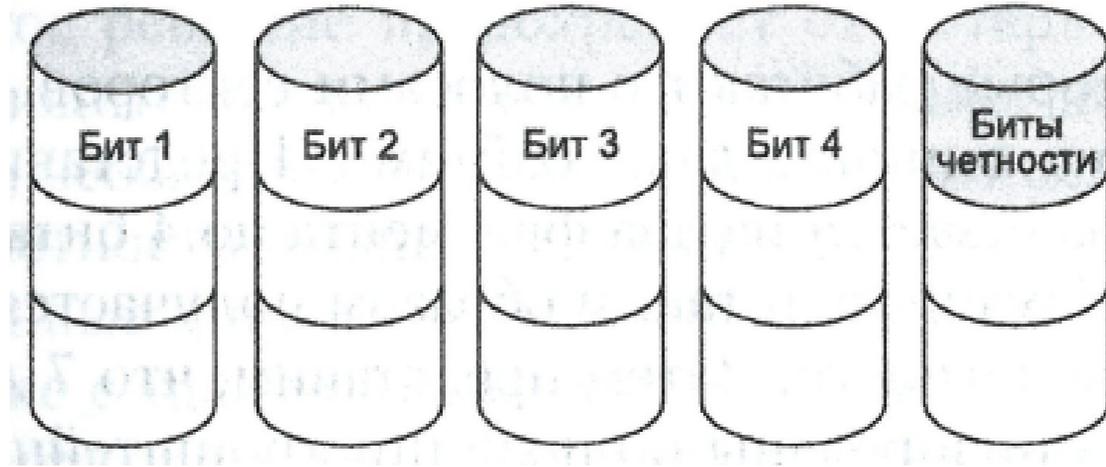


- RAID-массив уровня 2 оперирует словами, а иногда даже байтами
- Все диски должны быть синхронизированы по позиции кронштейна и позиции вращения.

1 байт разбивается на два фрагмента по 4 бита, затем к каждому из них добавляется код Хэмминга => слово из 7 бит, где 1, 2 и 4 – биты четности.

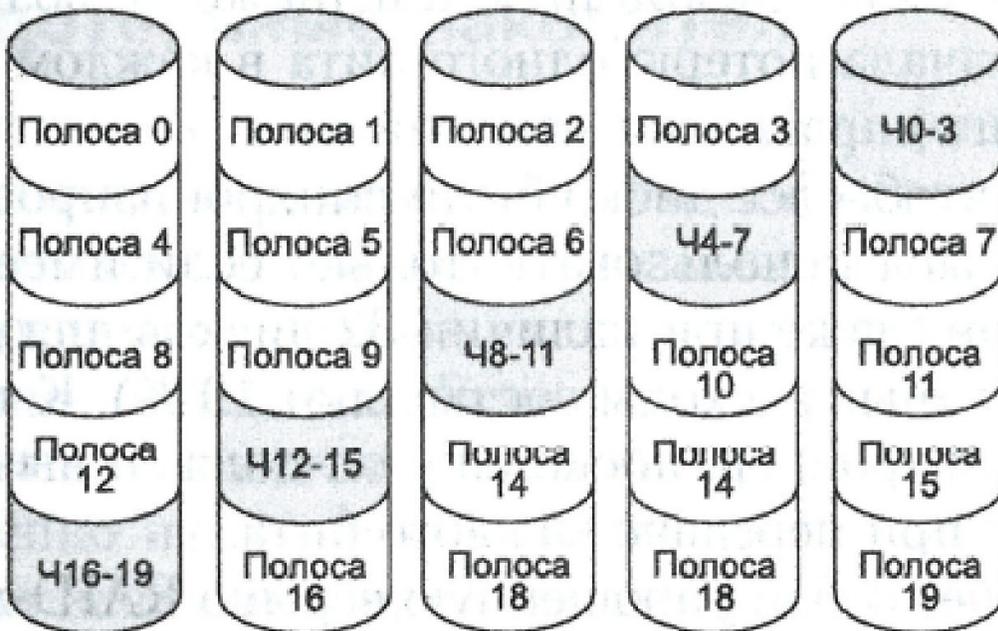
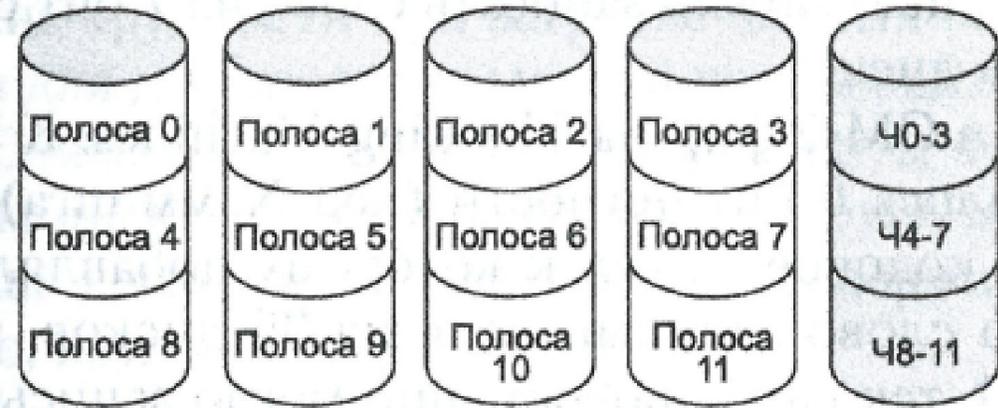
**Т.о за одну операцию можно записать слово из 7 бит с кодом Хэмминга на 7 дисков по 1 биту на диск.**

## RAID-массив уровня 3



- Для каждого слова данных вычисляется один бит четности и записывается на диск четности.
- Как и в RAID-массиве уровня 2, диски должны быть точно синхронизированы, поскольку каждое слово данных распределено по нескольким дискам.
- Один бит четности позволяет только обнаружить ошибки, но не исправлять их.

## RAID-массивы уровней 4 и 5

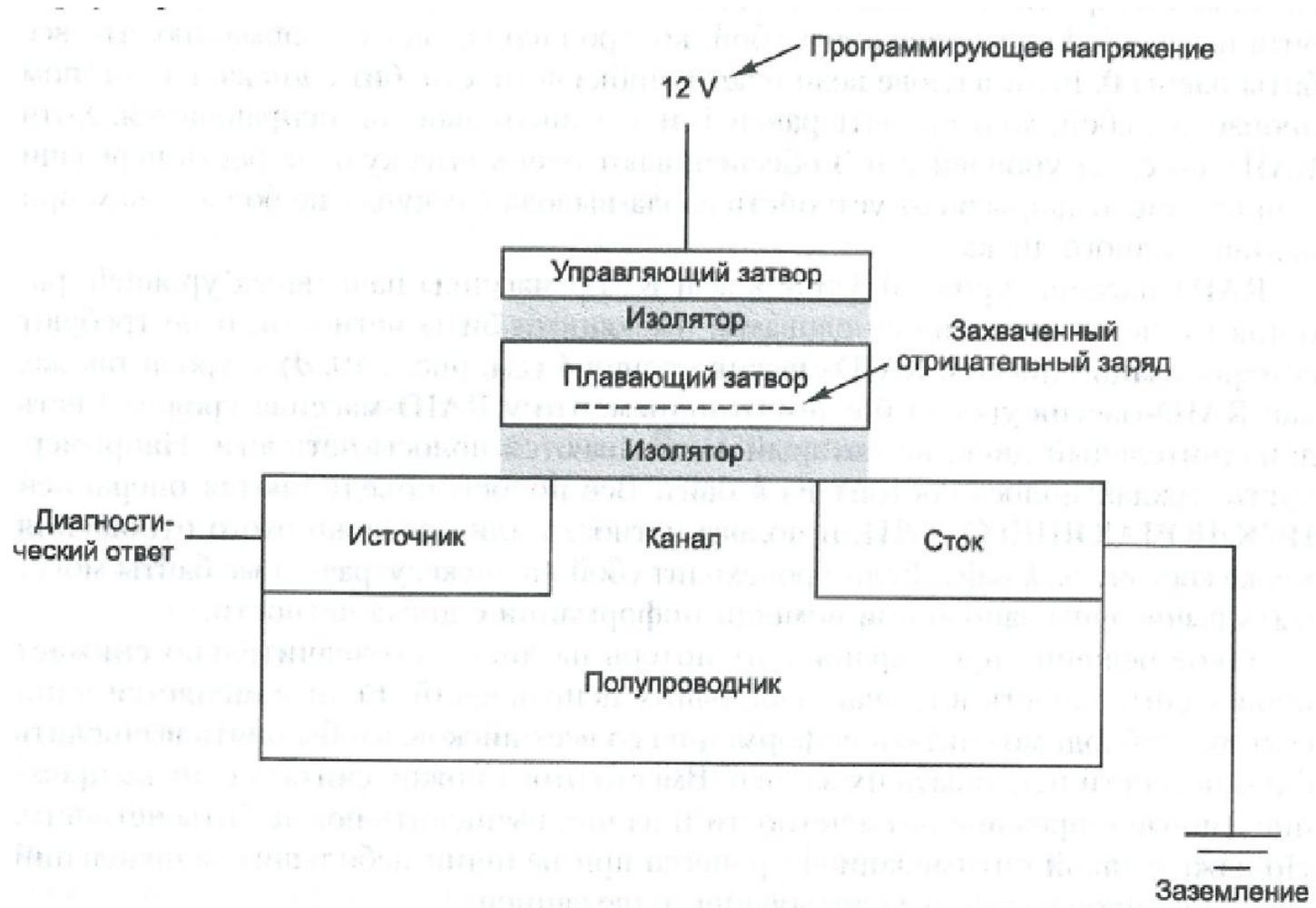


- RAID-массивы уровней 4 и 5, как и RAID-массивы начальных уровней, работают с полосами, а не со словами, имеющими биты четности, и не требуют синхронизации дисков.
- RAID-массива уровня 4 есть дополнительный диск, на который записываются полосы четности.
- Например, пусть каждая полоса состоит из  $k$  байт. Все полосы объединяются операцией исключающего ИЛИ, и полоса четности для проверки этого отношения также состоит из  $k$  байт. Если происходит сбой на диске, утраченные байты могут быть вычислены заново при помощи информации с диска четности.

# Твердотельные накопители

- Устройства на базе энергонезависимой флэш-памяти, часто называемые **твердотельными накопителями** или **SSD-дисками** (Solid State Disk), постепенно начинают рассматриваться как высокоскоростная альтернатива традиционным технологиям магнитных дисков.
- Фудзио Масуока (Fujio Masuoka), начале 1980-х годов изобрел первую флэш-память
- У **SSD-дисков** нет движущихся частей, дисковых пластин, а данные хранятся во флэш-памяти. Они напоминают диски только тем, что содержат большой объем данных, который при отключении питания не теряется.

- ✓ Флэш-память образуется из множества твердотельных ячеек, состоящих из одного специального флэш-транзистора.
- ✓ В транзистор встроен плавающий затвор, который может заряжаться и разряжаться при помощи высоких напряжений.



## Ячейка флэш-памяти

- Для программирования ячейки флэш-памяти на управляющий затвор подается высокое напряжение (в компьютерном мире напряжение в 12 В считается высоким), ускоряющее процесс инжекции горячих электронов в плавающий затвор.
- Электроны внедряются в плавающий затвор, что приводит к появлению отрицательного заряда внутри транзистора.
- Внедренный отрицательный заряд увеличивает напряжение, необходимое для включения транзистора;
- Проверяя, включается ли канал при высоком или низком напряжении, можно определить, заряжен ли плавающий затвор, и таким образом получить результат 0 или 1 для ячейки флэш-памяти.
- Внедренный заряд остается в

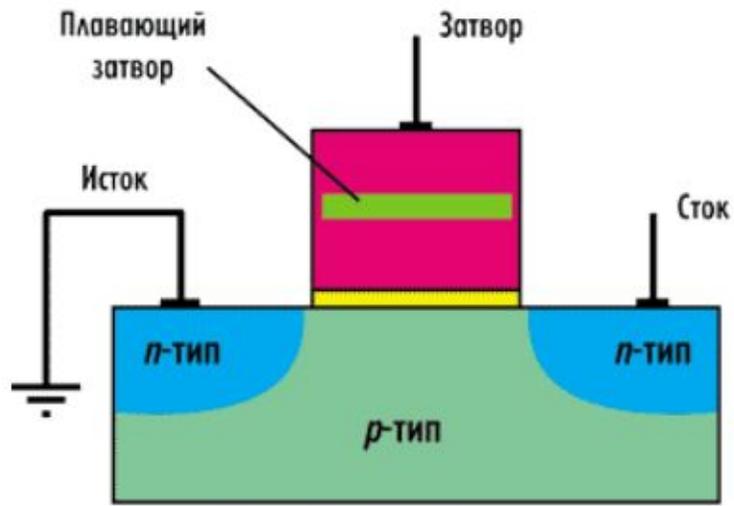
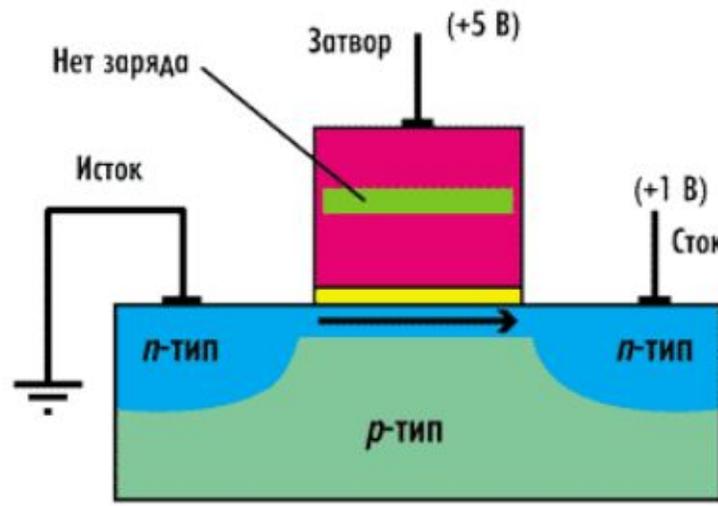
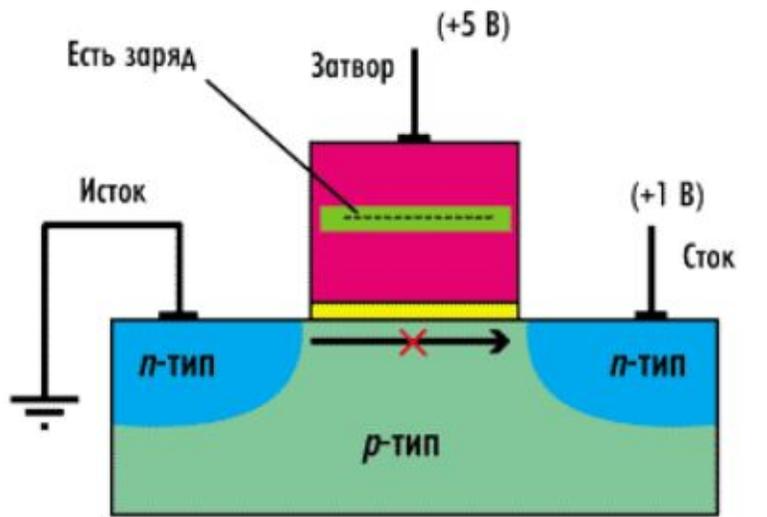


Схема MOSFET-транзистора с плавающим затвором



При отсутствии заряда на плавающем затворе ведет себя как обычный MOSFET-транзистор



При наличии заряда на плавающем затворе нет канала проводимости даже при подаче напряжения на управляющий затвор

- В открытом состоянии напряжение между стоком и истоком близко к нулю, а в закрытом может достигать высокого значения.
- Сам по себе транзистор не способен сохранять информацию
- Плавающий затвор предназначен для хранения информации:
  - ✓ выполнен из поликристаллического кремния и полностью окружен слоем диэлектрика, что обеспечивает ему полное отсутствие электрического контакта с элементами транзистора;
  - ✓ расположен между управляющим затвором и подложкой из p-n-переходов;
  - ✓ способен сохранять помещенный на него заряд (отрицательный) в течение неограниченного времени (до 10 лет);
- Наличие или отсутствие избыточного отрицательного заряда (электронов) на плавающем затворе может трактоваться как логические единица и ноль.

## Плюсы

- обладают более высокой производительностью по сравнению с вращающимися магнитными дисками (скорость в два-три раза выше)
- не имеет подвижных частей, оно особенно хорошо подходит для ноутбуков
- Цена несколько центов за гигабайт

## Минусы

- стоимость гигабайта составляет от одного до трех долларов
- ресурс безотказной работы (флэш-ячейка перестает функционировать примерно через 100 000 операций перезаписи)

## SSD диски

Для повышения срока жизни SSD была разработана методика **нивелирования износа**, основанная на распределении записи по всем ячейкам на диске. Каждый раз, когда на устройство записывается новый блок, для записи выбирается блок, относительно давно не использовавшийся.

# Диски

- **Диски CD-ROM** – оптические диски, которые изначально использовались для записи телевизионных программ, позже стали одними из основных средств хранения информации в компьютерной индустрии.
- **Диски CD-R** (CD-Recordable – записываемый компакт-диск).
- **Диски CD-RW** (CD-ReWritable – перезаписываемый компакт-диск)
- **DVD-диски**, раньше расшифровывалась как Digital Video Disk (цифровой видеодиск), сейчас аббревиатура официально превратилась в Digital Versatile Disk (цифровой многоцелевой диск)
- **Диски Blu-Ray**, предусматривающую технологию применения синего лазера вместо красного