Организация памяти

Память

- Основной единицей хранения данных в памяти является двоичный разряд - бит.
- Набор бит объединяется в ячейку памяти – байт
- Байты объединяются в слова
- 16 – разрядное слово (двойное слово -2 байта)
- 32 – разрядное слово (4 байта)
- 64 – разрядное слово (8 байт)
- Каждая ячейка имеет номер, который называется адресом.

Память

- Память выполняет три операции:
- а) хранение информации;
- б) запись информации;
- в) чтение информации.

Характеристики памяти

- **Емкость памяти** - определяет максимальное количество хранимой в памяти информации (в битах, байтах, килобайтах, мегабайтах, гигабайтах, терабайтах и т.д.)

- **Время доступа к памяти**

  Время доступа при записи / чтении складывается из времени поиска ячейки памяти по заданному адресу и времени записи/чтения в ячейку.

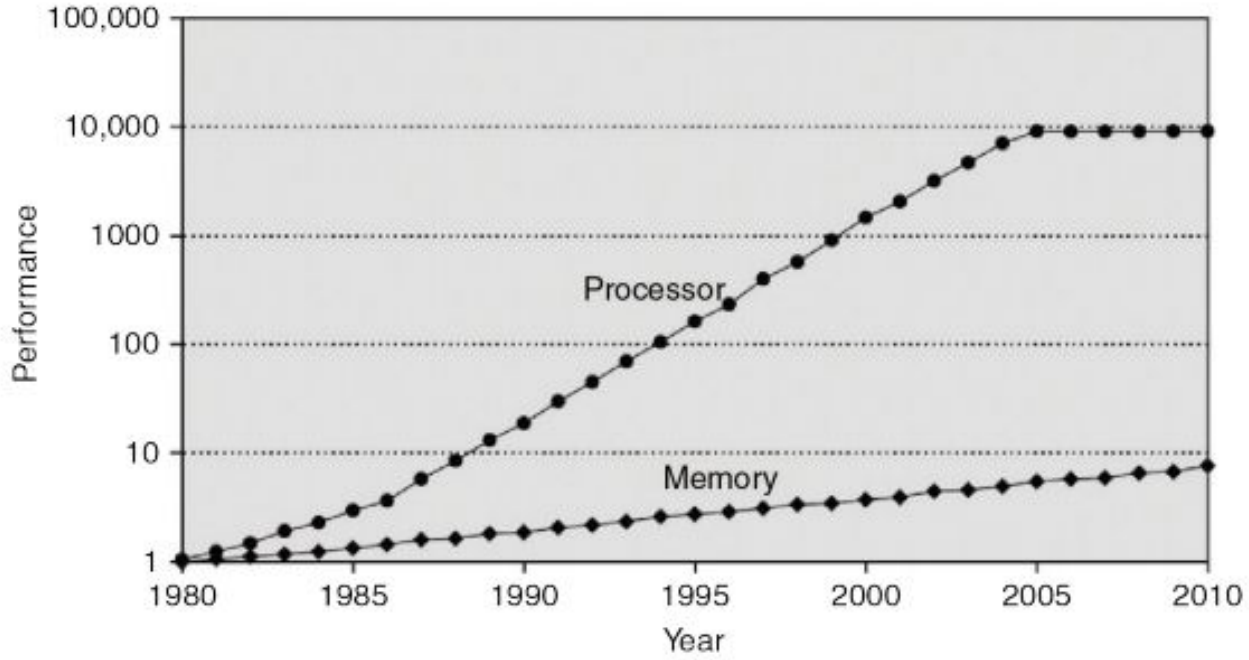
- $t_{обращ.зап} = t_{поиска} + t_{записи}$
- $t_{обращ.чтен} = t_{поиска} + t_{чтения}$

Характеристики памяти

- **Пропускная способность шины памяти (Bandwidth)**
  - Количество данных, переданных памятью по шине памяти за единицу времени (КБ, МБ, ГБ /сек.)

- **Стоимость** памяти - стоимостью хранения одного бита информации.

Сравнение производительности памяти и процессора

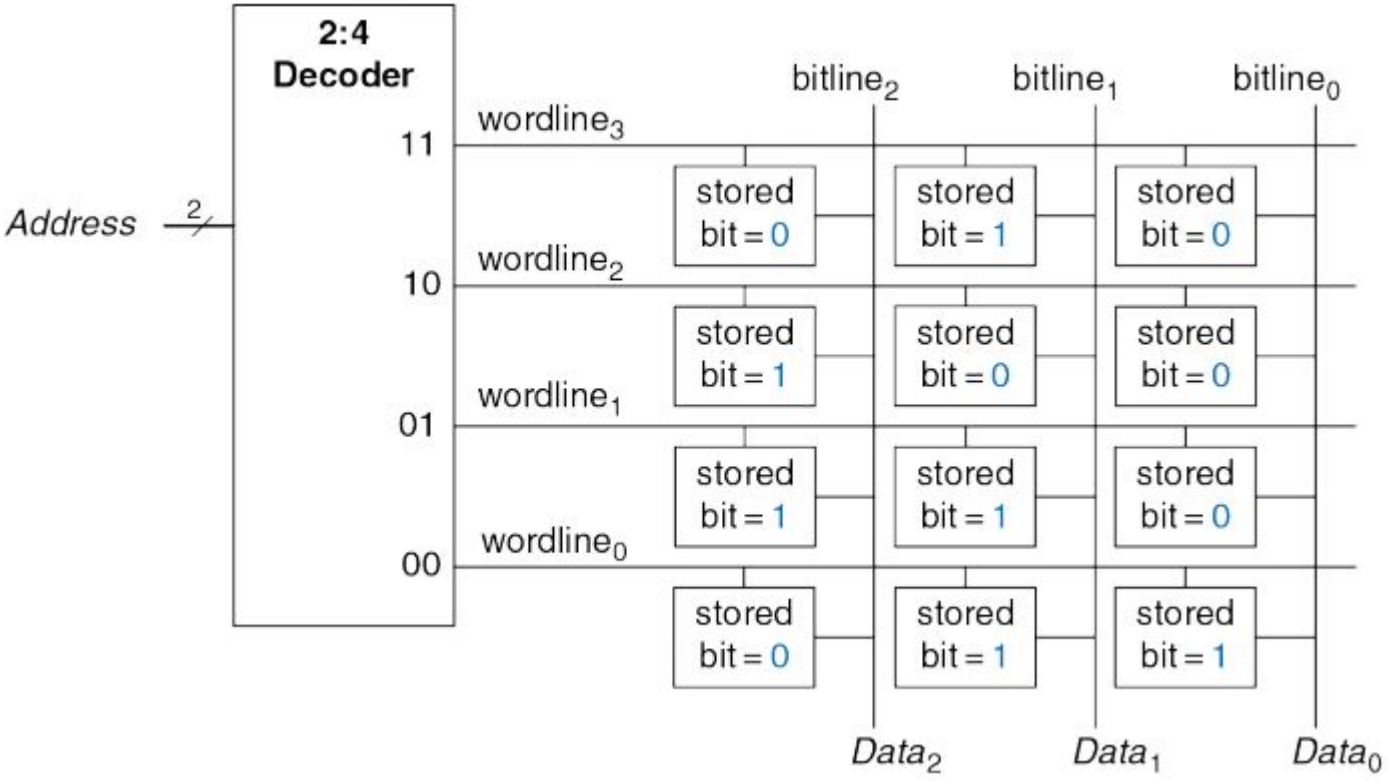


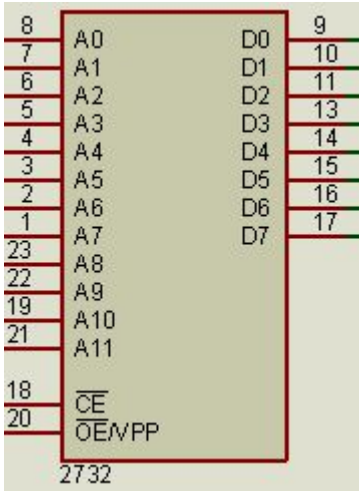
Разрыв в производительности процессоров и памяти.

Организация микросхем памяти

Байт ориентированная память

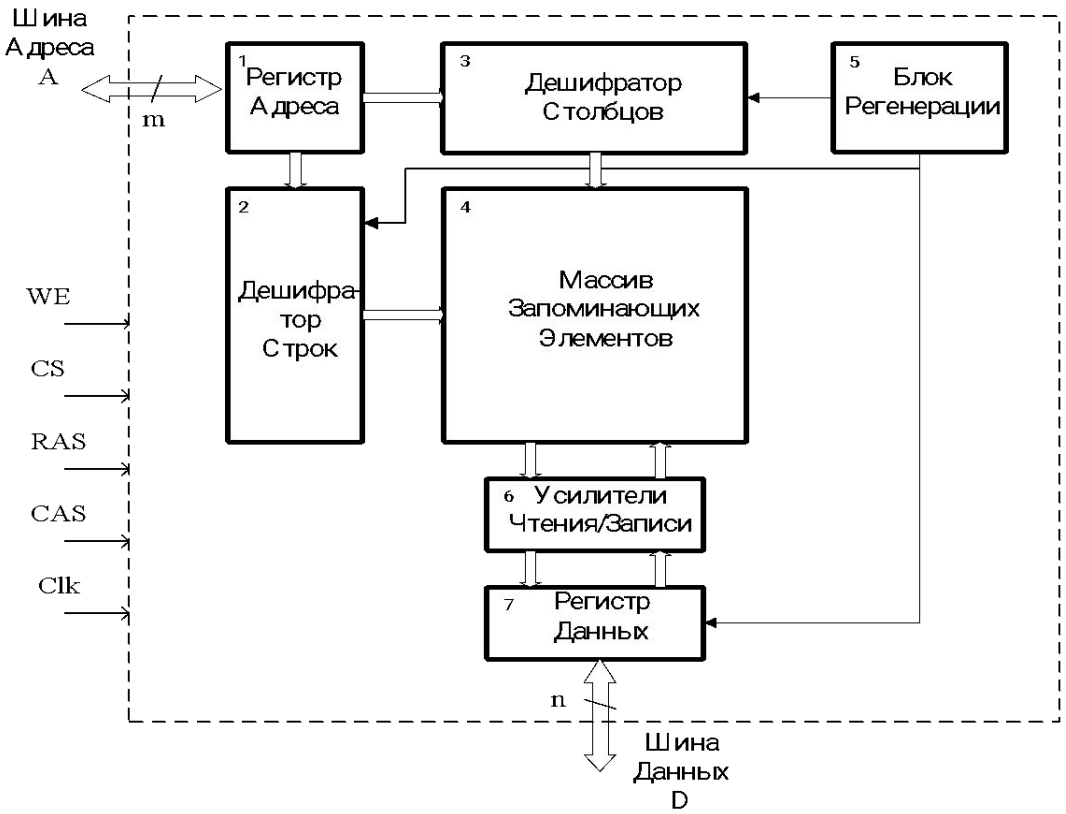
- Матрица памяти (прямоугольная)





- Ячейки выбираются построчно

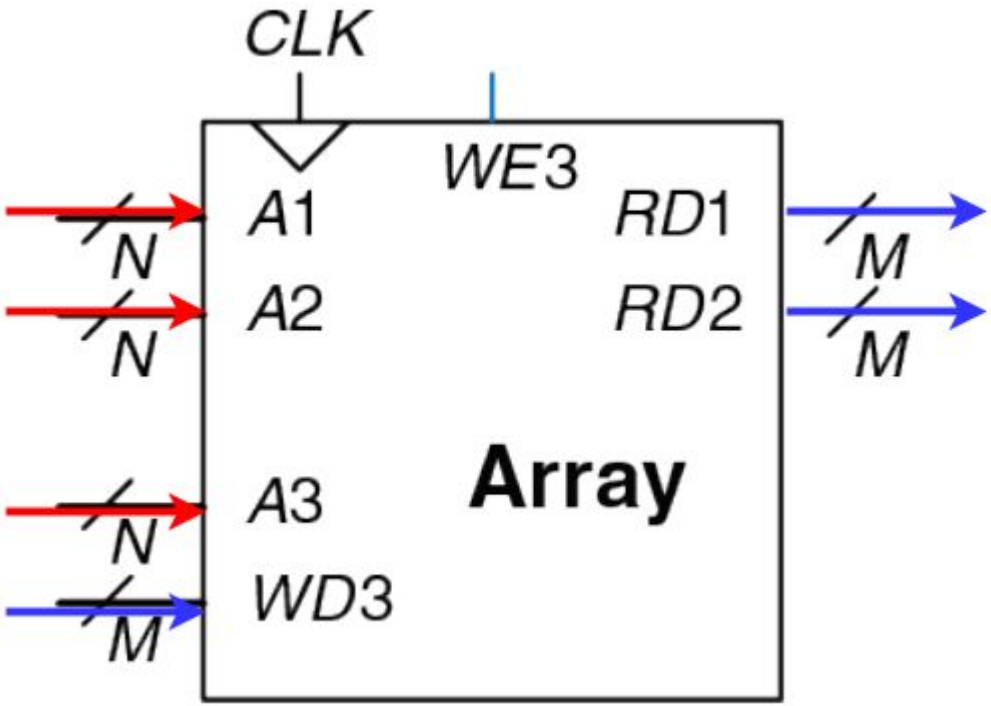
Бит/байт ориентированная память (квадратная)



Ячейка может хранить один или несколько бит

- Ячейка выбирается на пересечении строки и столбца
- Для уменьшения количества выводов по одним и тем линиям адреса сначала подается адрес строки, который дешифрируется по сигналу RAS (row address strobe), а потом адрес столбца, дешифрируется по сигналу CAS (column address strobe - строб адреса столбца) .
- Данные считываются по линии столбцов

Многопортовая память

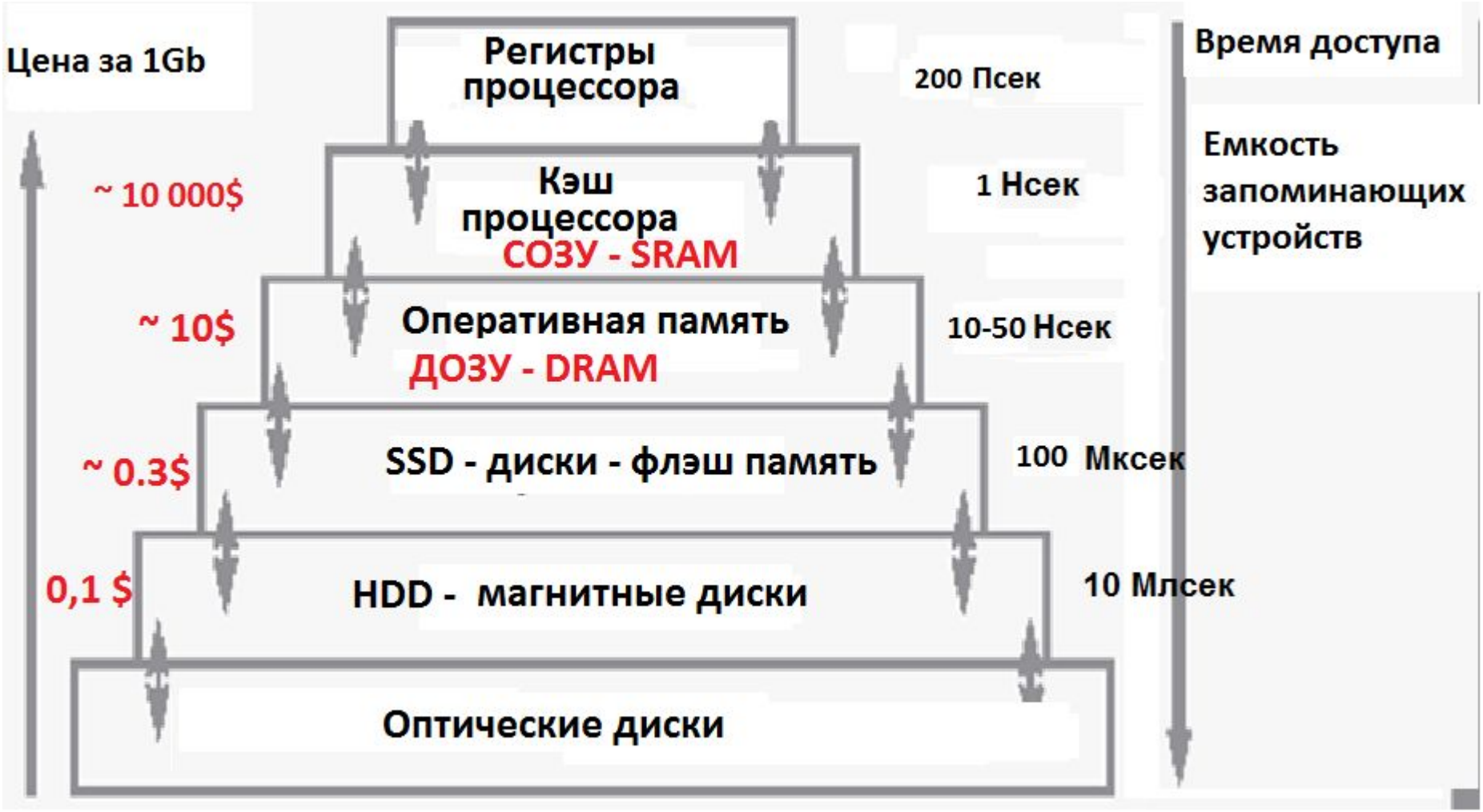


- RD1 – данные считанные по адресу A1
- RD2 – данные считанные по адресу A2
- WD3 – данные записываемые по адресу A3

Типы памяти

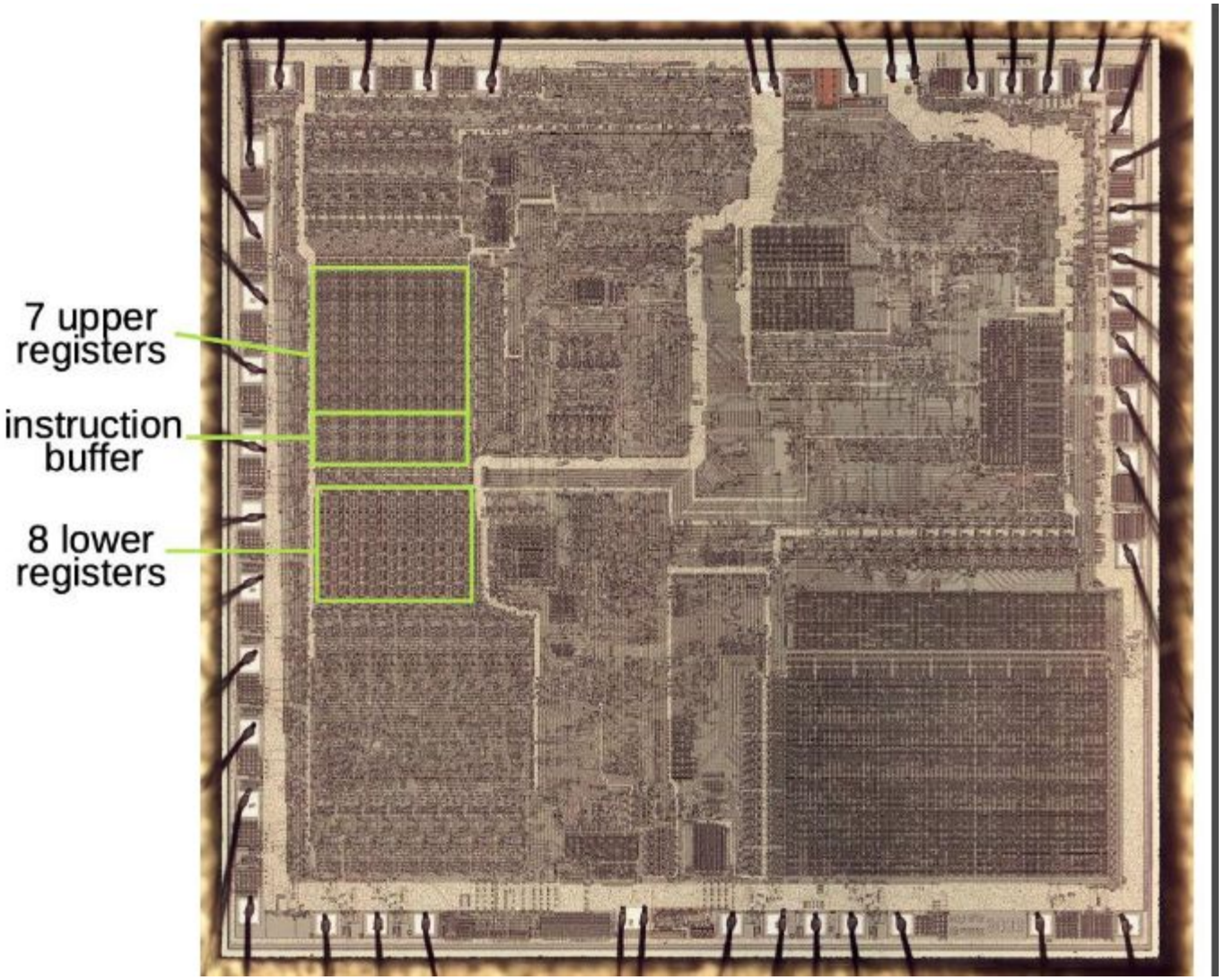
- Статическая память (СОЗУ);
- Динамическая память (ДОЗУ);
- Энергонезависимая память :
  - Однократно программируемая;
  - Многократно программируемая память;
- Память на жестких дисках;
- Память на оптических дисках.

Иерархия памяти



Задержка в регистрах процессора зависит от тактовой частоты

Регистры x8086



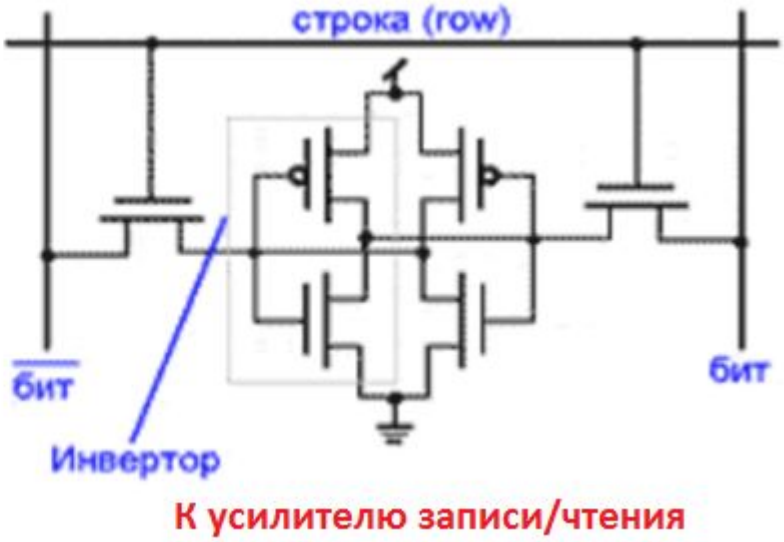
Регистры процессора

- Работают на частоте процессора
- В качестве запоминающего элемента используются триггеры с количеством транзисторов от 7 до 20
- Регистры располагается ближе к ядру процессора, поэтому работает быстрее кэша.
- Время переключения от 2 до 500 Пксек

Площадь и задержки

Тип памяти	Количество транзисторов в запоминающем элементе	Задержка
Триггер	~20	Малая
Статическое ОЗУ	6	Средняя
Динамическое ОЗУ	1	Большая

Статическая память - СОЗУ



- Элемент – асинхронный RS-тиггер - защелка
- Содержит 6 транзисторов

кэш

Кэш и основная память

Один раз прочитать большой блок из медленной оперативной памяти в кэш, а потом много раз обращаться к быстрому кэшу.

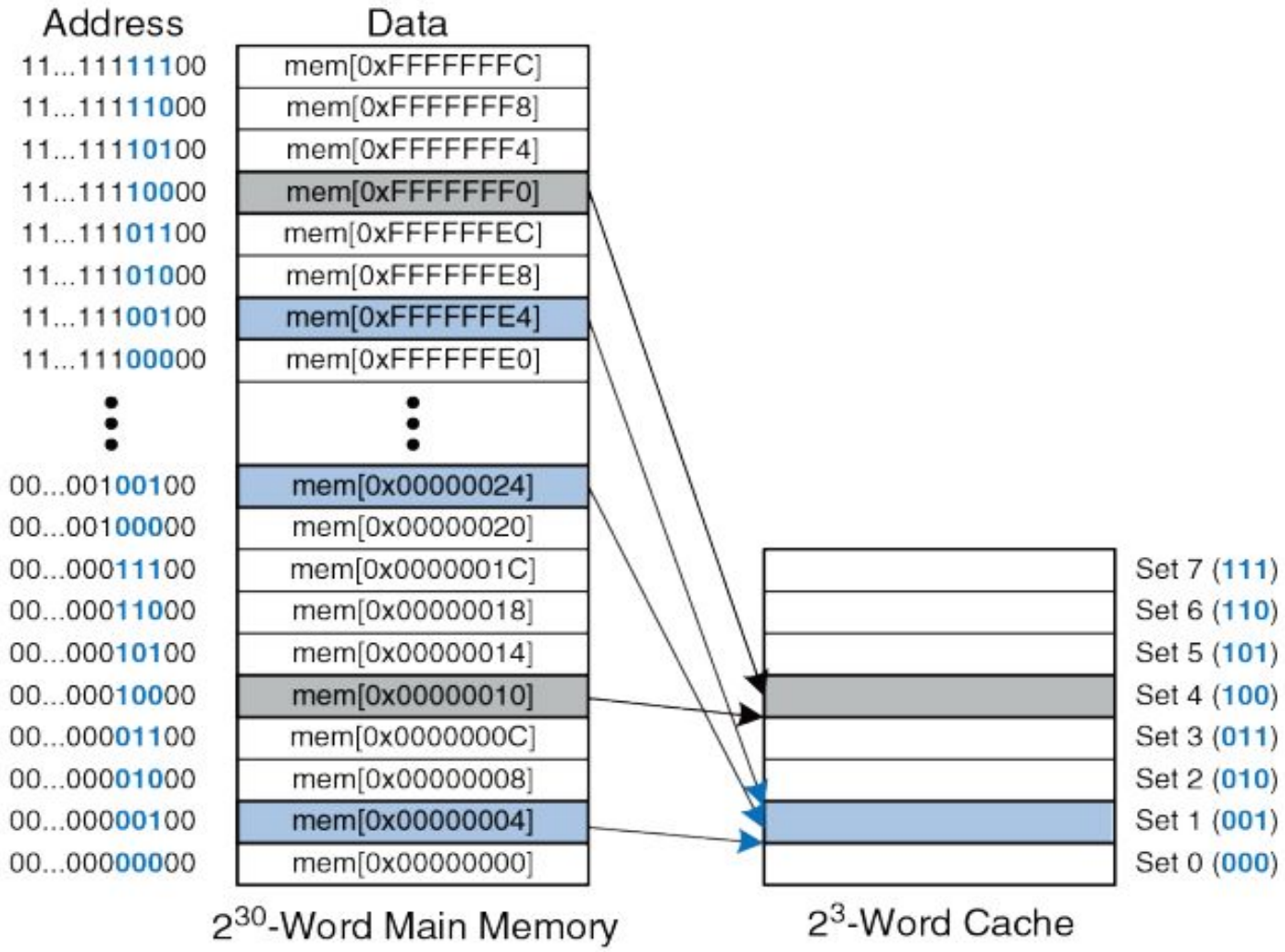
КЭШ

- Кэш должен предсказать, какие данные понадобятся процессору, и выбирать их из оперативной памяти
- Для предсказания кэш использует временную и пространственную локальность
- **Временная локальность**: процессор, вероятно, еще раз обратится к тем данным, которые он недавно использовал поэтому эти данные переписываются из ОЗУ в кэш
- Пространственная локальность: при обращении процессора к каким либо данным ему, вероятно, понадобятся и расположенные рядом данные .
- Поэтому, читая одно слово данных из памяти, он заодно читает и группу соседних слов.
- ***Строка* кэша -** группа слов считываемых в кэш за одно обращение ОП.

кэш

- Кэш разбит на наборы (Set), каждый из которых состоит из одной или нескольких строк кэша.
- Взаимосвязь между адресом данных в оперативной памяти и адресом этих данных в кэше называется отображением.
- Каждый адрес памяти всегда отображается в один и тот же набор кэша.
- Кэш-память классифицируется по числу строк в наборе.
  - *Кэш прямого отображения* (англ.: direct mapped cache) каждый набор содержит только одну строку
  - Наборно-ассоциативный кэш с *N* секциями - каждый набор состоит из *N* строк.
    - Если C – это емкость памяти, а b - длина строки кэша, то количество строк кэша B = C/b
  - *Полностью ассоциативный кэш* имеет только один набор (*S*=1), и данные могут оказаться в любой из *B* строк этого набора.

Кэш прямого отображения

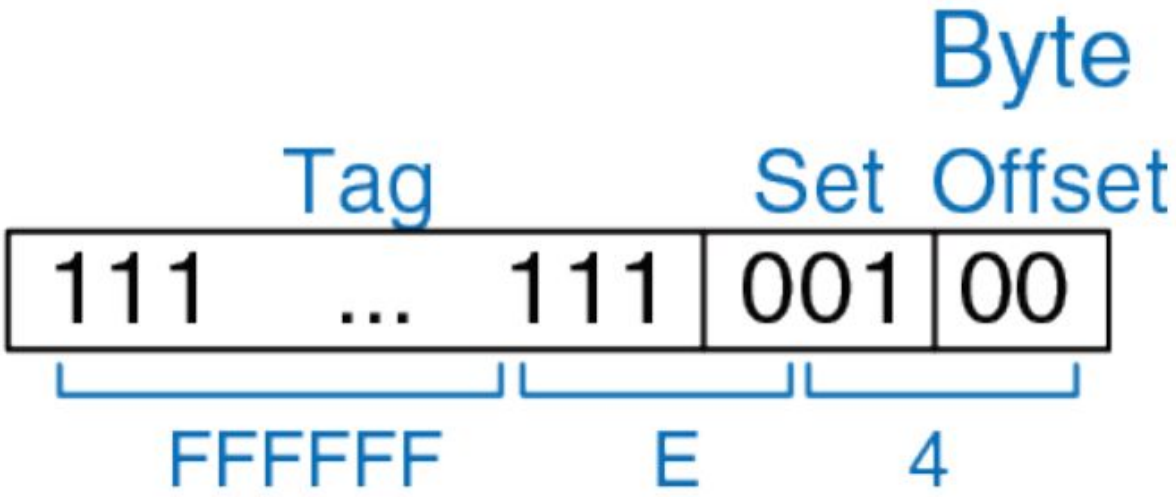


Отображение оперативной памяти на кэш прямого отображения

КЭШ прямого отображения

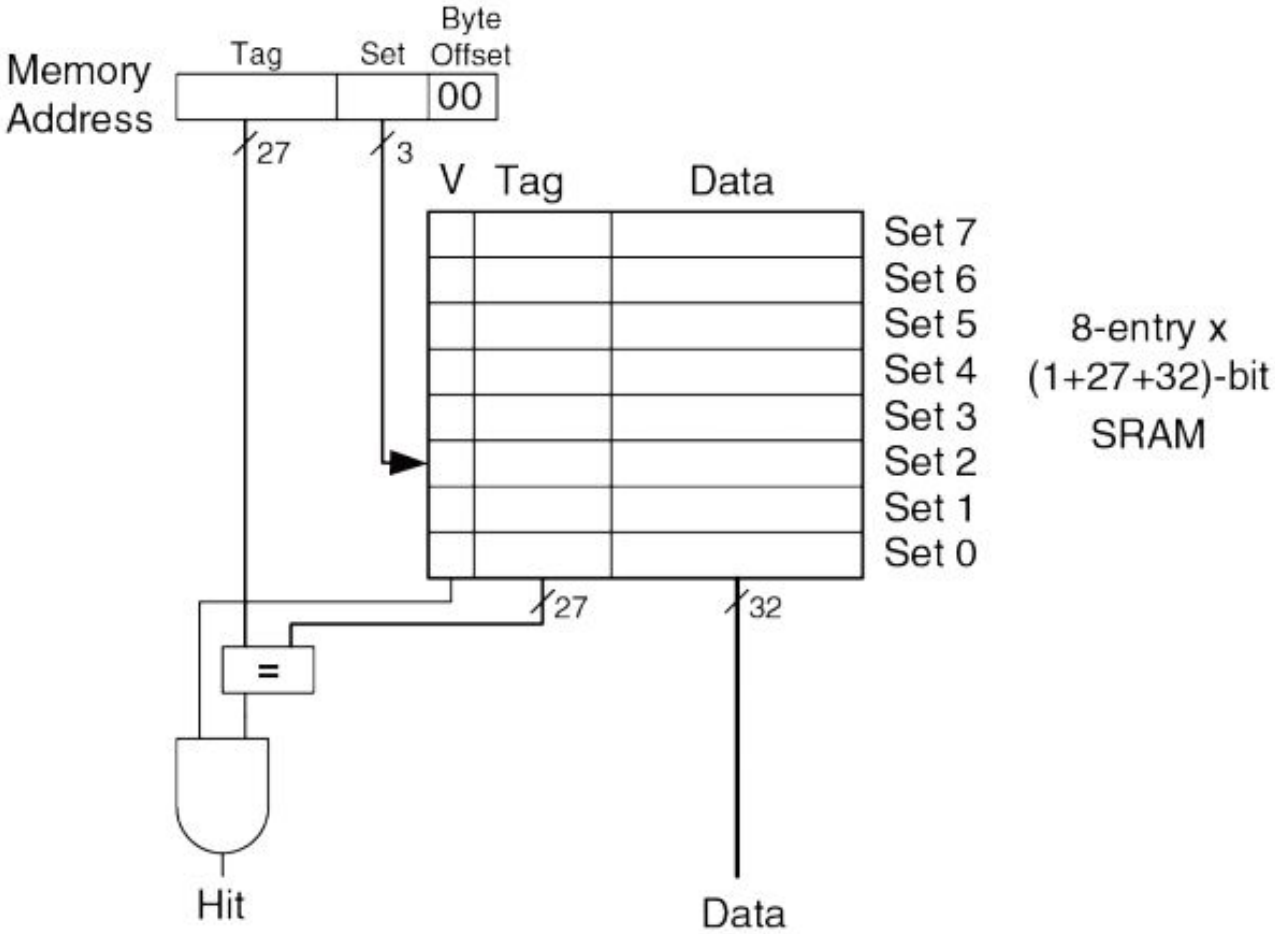
- В кэш-памяти прямого отображения каждый набор содержит только одну строку кэша
- Оперативная память условно поделена на блоки, равные длине строки кэша, а кэш память также поделена на наборы такого же размера.
- Для кэша из 8-ми слов первые 8 блоков памяти отображаются в 8 наборов кэша, потом вторые 8 наборов ОП отображаются в те же наборы кэша
- При этом может быть ситуация когда несколько блоков памяти отображаются в один и тот же набор кэша.

Формат адреса обращения к кэш



- Set – номер набора
- Qffset – номер байта внутри набора
- Tag – Признак (оставшаяся старшая часть адреса)
  - служит для определения какой из возможных адресов блоков ОП отображен в кэш.
  - Заносится при записи строки кэша.

Схема выборки данных



Кэш прямого отображения с восемью наборами

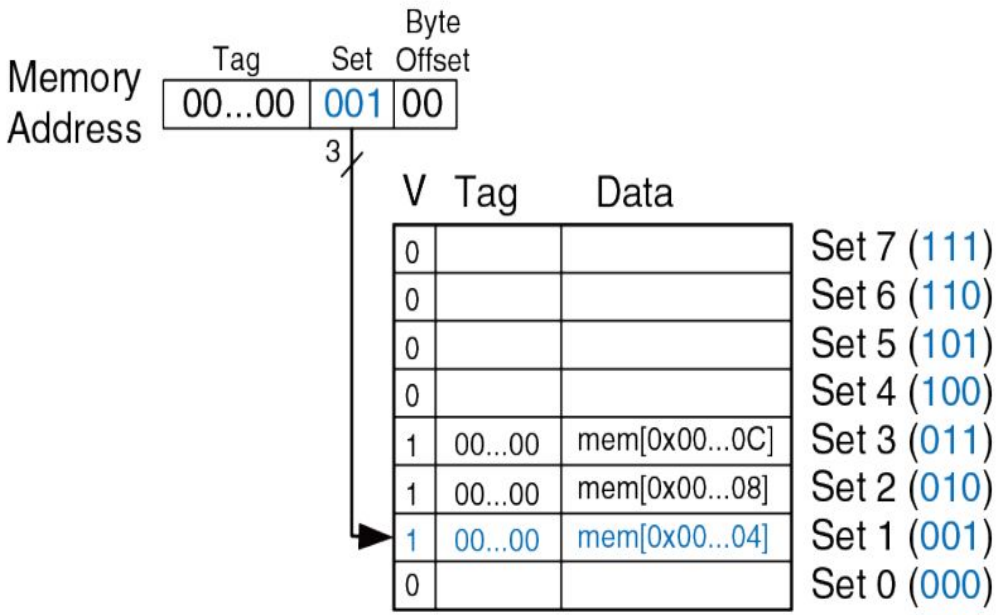
Запись в наборе состоит из: Строка кэша + Тэг + V (61 бит)

V- бит присутствия (достоверности) блока в кэше

Старшая часть адреса обращения сравнивается с признаком,

Пример работы кэша

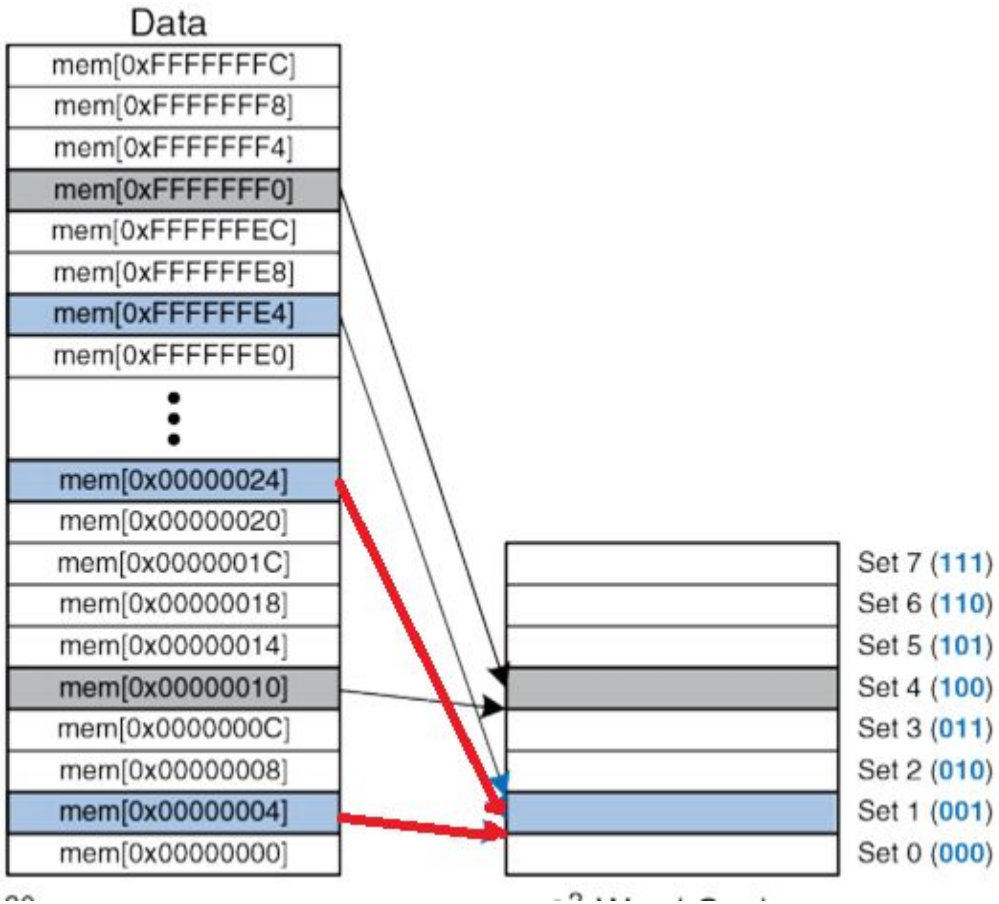
```
        addi $t0, $0, 5
loop:   beq  $t0, $0, done
        lw   $t1, 0x4($0)
        lw   $t2, 0xC($0)
        lw   $t3, 0x8($0)
        addi $t0, $t0, -1
        j    loop
done:
```



- Всего в цикле 15 обращений к памяти по адресам 0x4, 0xC, 0x8.
- При первом проходе цикла данных в кэше нет - 3 промаха
- Данные подгружаются в кэш и в остальные проходы процессор обращается в кэш
- Процент промахов 3/15 = 20%

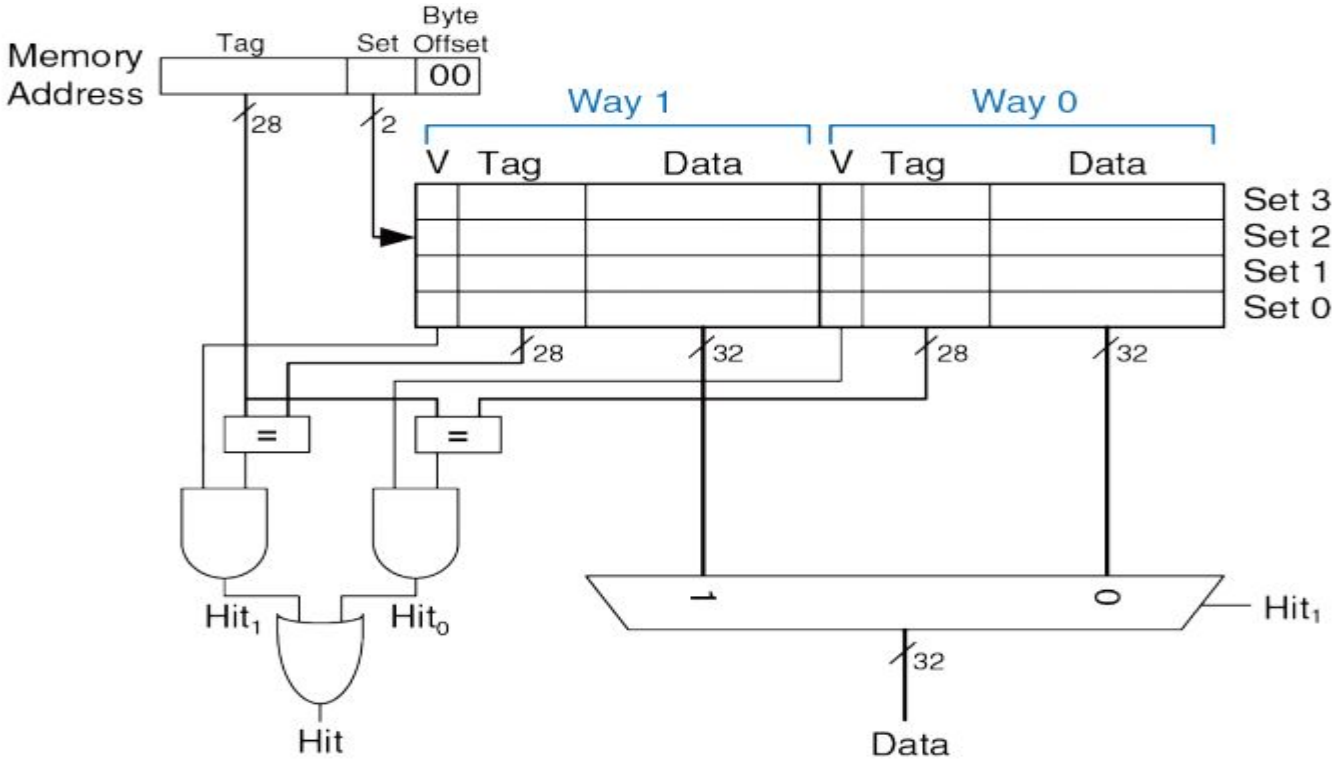
Пример работы кэша

```
      addi $t0, $0, 5
loop: beq  $t0, $0, done
      lw   $t1, 0x4($0)
      lw   $t2, 0x24($0)
      addi $t0, $t0, -1
      j    loop
done:
```



- Две команды обращаются в один и тот же набор кэша
- 100% промахов

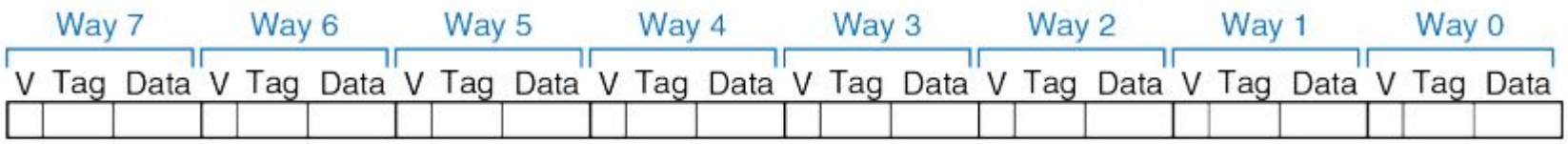
Многосекционный наборно-ассоциативный кэш



Двухсекционный наборно-ассоциативный кэш

- Кэш состоит из 4-х наборов, каждый из которых вмещает две секции.
- При отображении двух адресов на один набор кэша конфликтов не возникает, так как каждый адрес ОП будет отображен в одну из секций набора.
- При чтении происходит сравнение адреса обращения с признаком сразу во всех записях набора кэша (ассоциативный поиск)

Полностью ассоциативный кэш



Полностью ассоциативный кэш с восемью строками

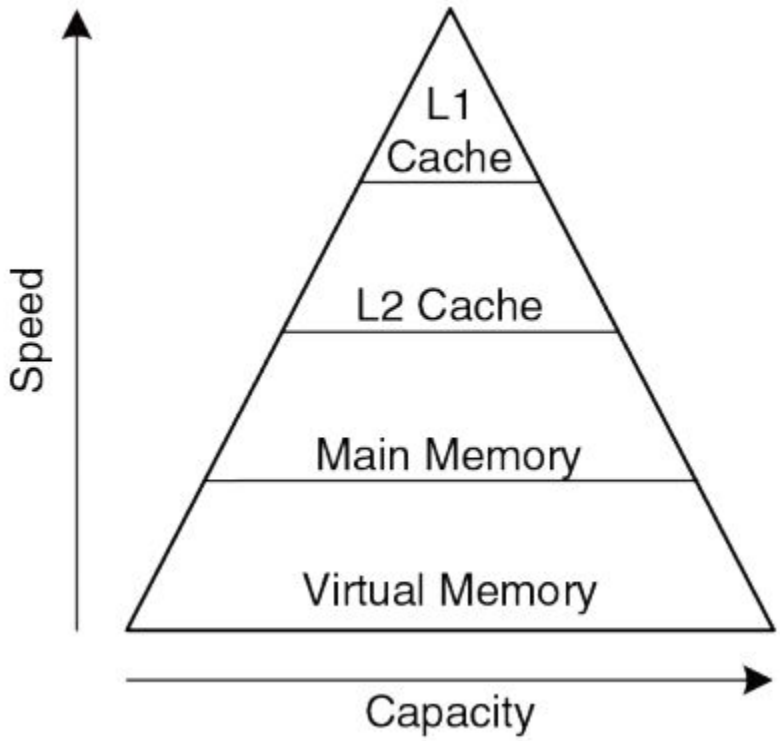
- Кэш состоит из 8 секций
- Адрес памяти может быть отображен в строку любой из этих секций
- Минимальное количество конфликтов
- Большие аппаратные затраты

Запись в кэш

- *Сквозная запись* (write-through)
  - Данные, записываемые в кэш, одновременно записываются и в оперативную память
  - Недостаток - большее количество операций записи в память

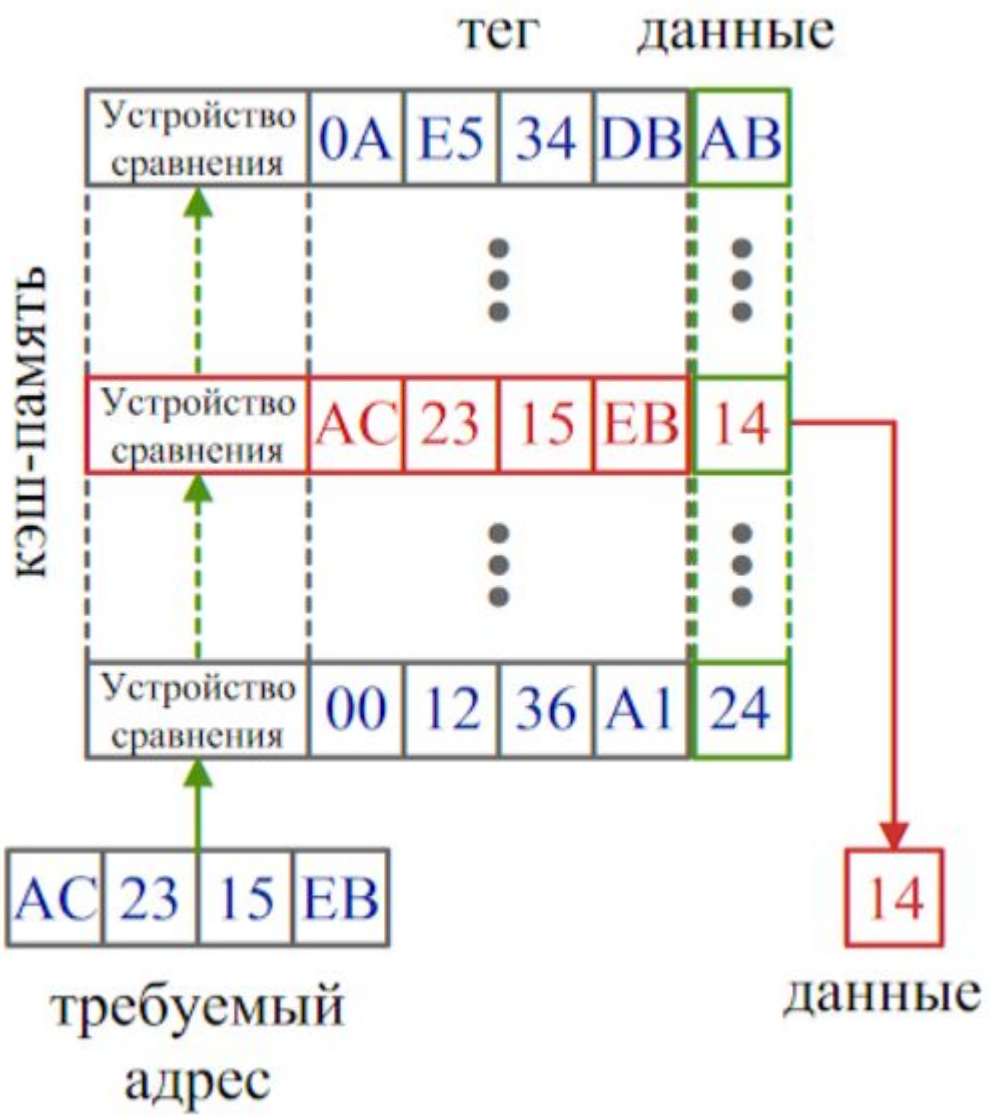
- *Отложенная запись* (write-back).
  - У каждой строки есть бит изменения (*D*, от англ. dirty). Если в строку производилась запись, то этот бит равен 1, в противном случае он равен 0.
  - Измененные строки записываются обратно в оперативную память только тогда, когда они вытесняются из кэша.
  - Используется чаще

Многоуровневые кэши



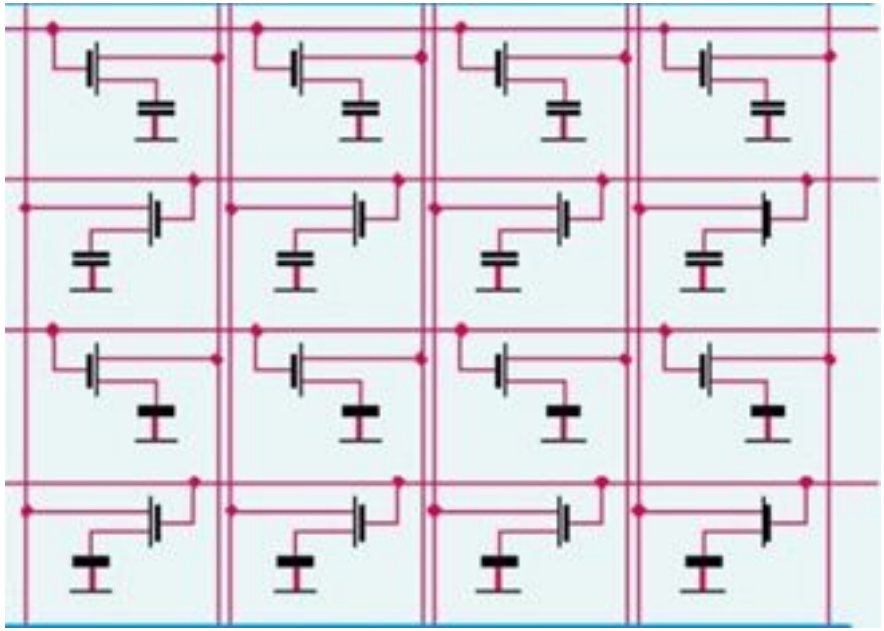
- Чем больше кэш, тем он медленнее
- Используют многоуровневые кэши

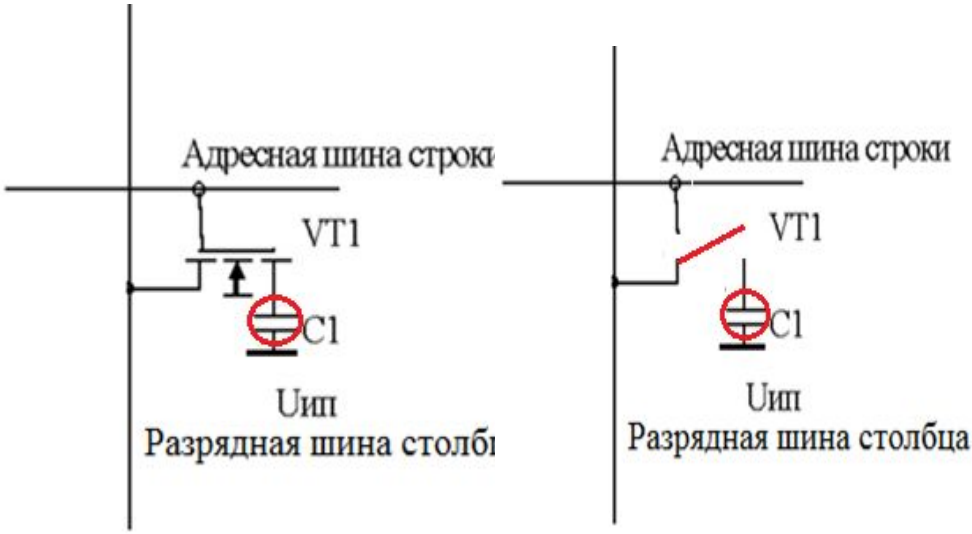
Ассоциативная память



Динамическая память

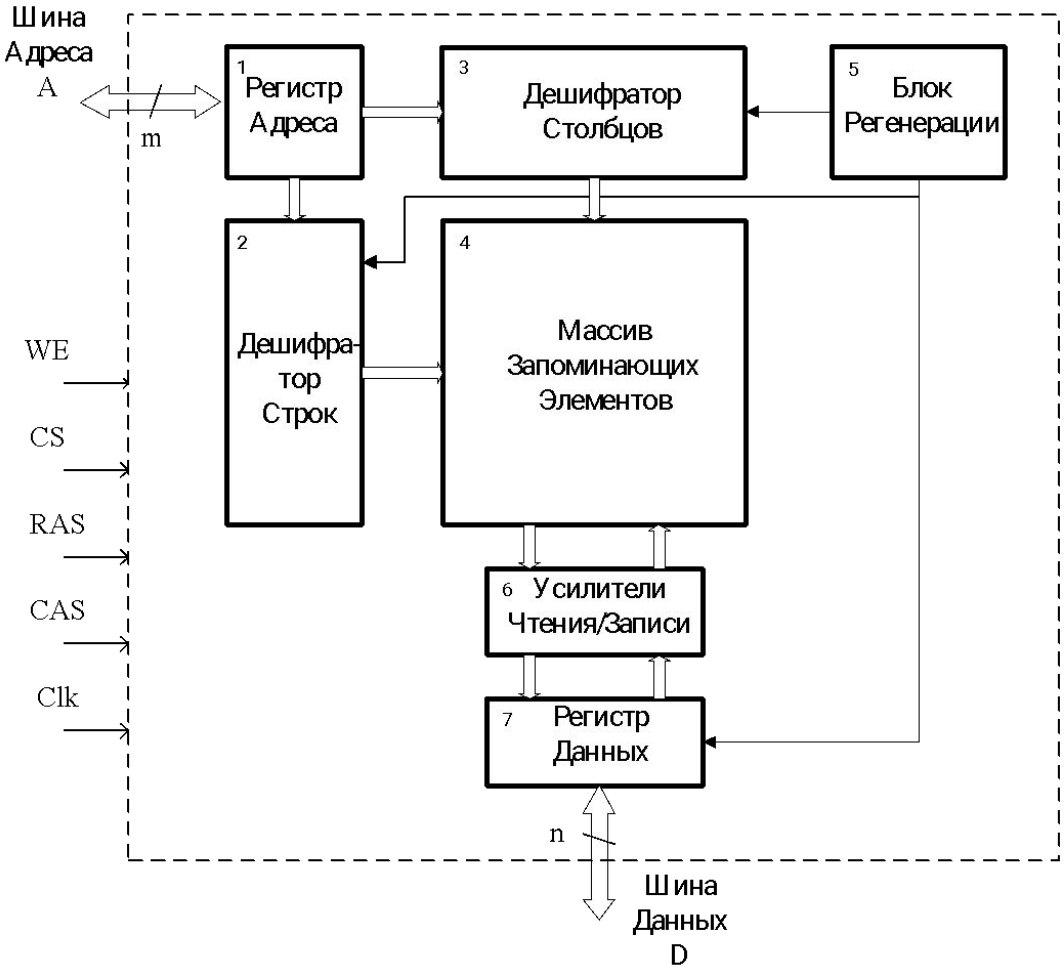
Динамическая память





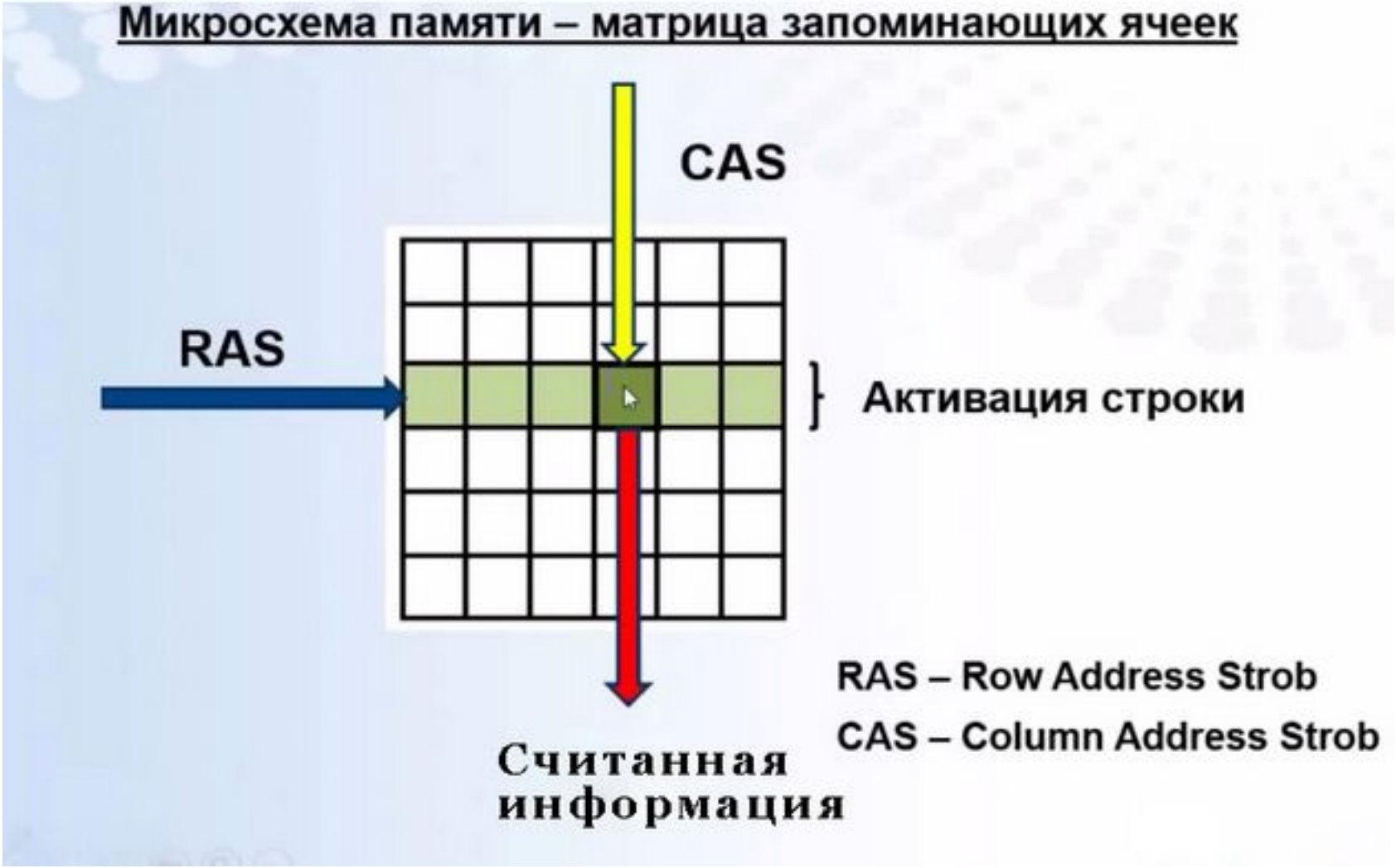
- **Матрица** запоминающих элементов (конденсаторов)
- Низкая стоимость. (один транзистор)
- Низкое быстродействие (конденсатор надо периодически подзаряжать специальными циклами регенерации. Во время подзарядки обращение к ячейке не возможно).
- При чтении конденсатор так же разряжается

Организация ДОЗУ

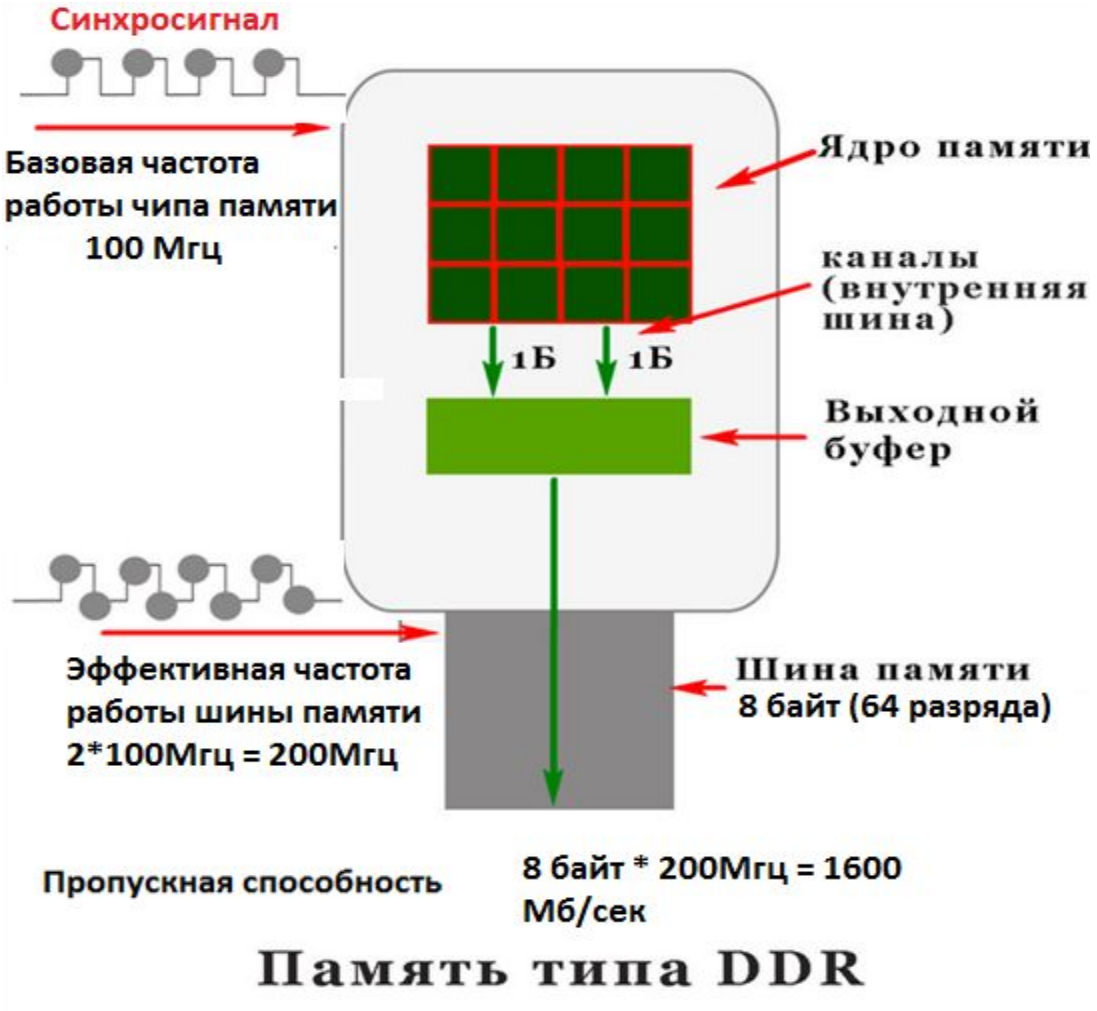


- Сначала адрес строки дешифрируется по сигналу RAS (row address strobe), а потом адрес столбца, дешифрируется по сигналу CAS (column address strobe - строб адреса столбца) .

Структура динамической памяти



Эффективная частота и пропускная способность



Эффективная частота = 2 * базовая частота потому, что данные передаются по внешней шине памяти по переднему и заднему фронту синхросигнала

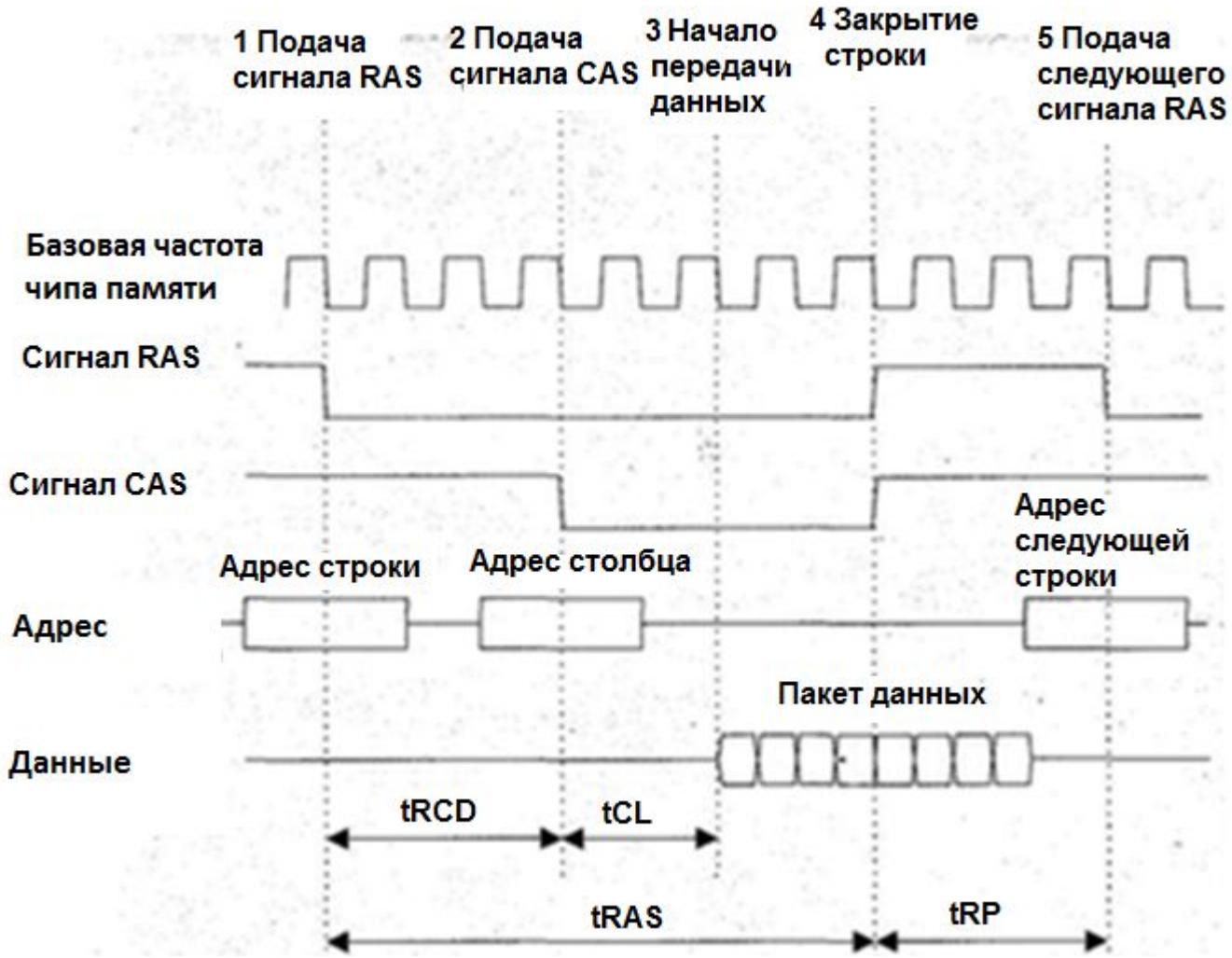
DDR3

DDR3 SDRAM – третье поколение памяти DDR.

Стандарт памяти DDR3 SDRAM	Рабочая (базовая) частота шины, МГц	Тип модуля: 240 pin DDR3 SDRAM DIMM	Пропускная способность шины, Мбайт/с
DDR3-800	800 (400)	PC3-6400	6400
DDR3-1067	1067 (533)	PC3-8500	8500
DDR3-1333	1333 (667)	PC3-10600	10600
DDR3-1600	1600 (800)	PC3-12800	12800
DDR3-1866	1866 (933)	PC3-14900	14900
DDR3-2133	2133 (1066)	PC3-17000	17000

- 6400Мбайт = 800*8байт(64бита внешняя шина)

Диаграмма работы динамической памяти



Тайминг памяти - количество тактов базовой частоты матрицы памяти между фронтами сигналов управления памятью

Основные тайминги динамической памяти

- Четыре основных тайминга, :
- **tCL (time of CAS Latency) -** задержка между сигналом CAS и выдачей данных (записью данных) из памяти;
- -**tRCD (time of RAS to CAS Delay) -** задержка от импульса RAS до импульса CAS;
- - **tRP (time of Row Precharge),-** задержку между завершением обработки одной строки и началом обработки другой;
- - **tRAS (time of Active to Precharge Delay) -** длительность сигнала RAS. Время задержки обработки одной строки.

  - Так для тайминга 10-11-10-30 DDR3 1866 время задержки выдачи данных можно определить:
  - T = 1/933 = 1,07 нсек
  - tCL = 10* 1.07 = 10.7 нсек

Тайминги динамической памяти

- Идеально – все единицы.
- Для памяти с большей частотой внешней шины(DDR3 и DDR4) памяти значения таймингов будет выше.
- Выбирают минимальное значение в своем типе.
- Некоторые версии BIOS позволяют ручную настройку таймингов.
- Если используются разные планки памяти, желательно чтобы тайминги были одинаковы.

Тайминги памяти

Эволюция динамической памяти

- SDRAM-(*Synchronous Dynamic Random Access Memory*) динамическое ОЗУ, работало в синхронном режиме с контроллером памяти.

- Double Data Rate SDRAM- синхронная динамическая память с удвоенной частотой передачи данных:
  - Обмен данными по внешней шине по переднему и заднему фронту тактового импульса.
  - Разрядность внешней шины данных стала 64 бита, а внутренней (от чипа до буфера 64)

Эволюция динамической памяти

- Double Data Rate 2 SDRAM
- Double Data Rate 3 SDRAM
- Double Data Rate 4 SDRAM
- Double Data Rate 5 SDRAM -
-
- Изменялось:
- базовая частота работы чипа памяти и эффективная частота внешней шины данных
- Понижение напряжения питания и энергопотребления

DDR

Тип	Пропускная способность	Напряжение	Предварительная выборка	Год
SDR	1.6 ГБ/с	3.3	1	1993
DDR	3.2 ГБ/с	2.5	2	2000
DDR2	8.5 ГБ/с	1.8	4	2003
DDR3	8.5 ГБ/с	1.8	8	2007
DDR4	25.6 ГБ/с	1.2	8	2017
DDR5	32 ГБ/с	1.1	8/16	2019

Режимы работы ДОЗУ

- Память может работать в:
  - одноканальном (Single Channel),
  - двухканальном (Dual Channel),
  - трехканальном (Triple Channel)
  - четырехканальном режиме (Quad Channel).

- В одноканальном режиме запись данных происходит последовательно в каждый модуль. Хорошо подходит в случае одной планки памяти.
- В многоканальных режимах запись данных происходит параллельно во все модули, что приводит к значительному увеличению быстродействия подсистемы памяти.
- Главным условием работы двухканального режима является наличие 2 или 4 планок памяти.
- Для трехканального режима необходимо 3 или 6 планок памяти, а для четырехканального 4 или 8 планок.

Лучшие производители

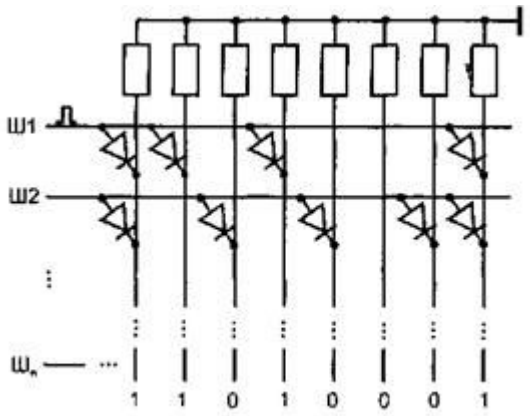
- Kingston
- Crucial
- Samsung
- Transcend
- Hynix
- Доля их брака составляет всего 0,6%

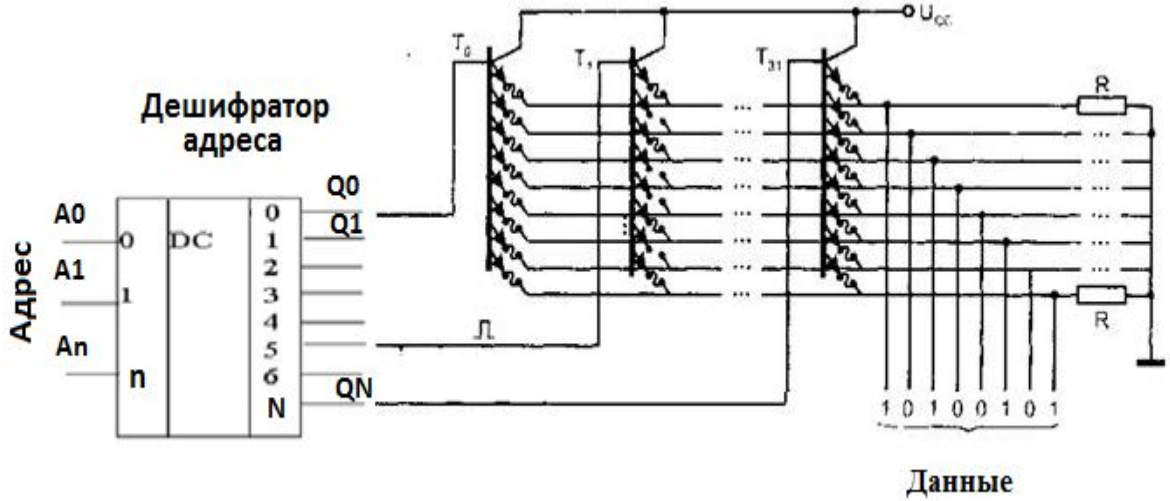
Постоянная память

Постоянная память

- **Постоянная (Энергонезависимая) память**
- **Однократно программируемая (**ROM (Read Only Memory)**) только для чтения**

Запоминающим элементом является пережигаемая плавкая перемычка или полупроводниковый диод, играющий роль разрушаемой перемычки.



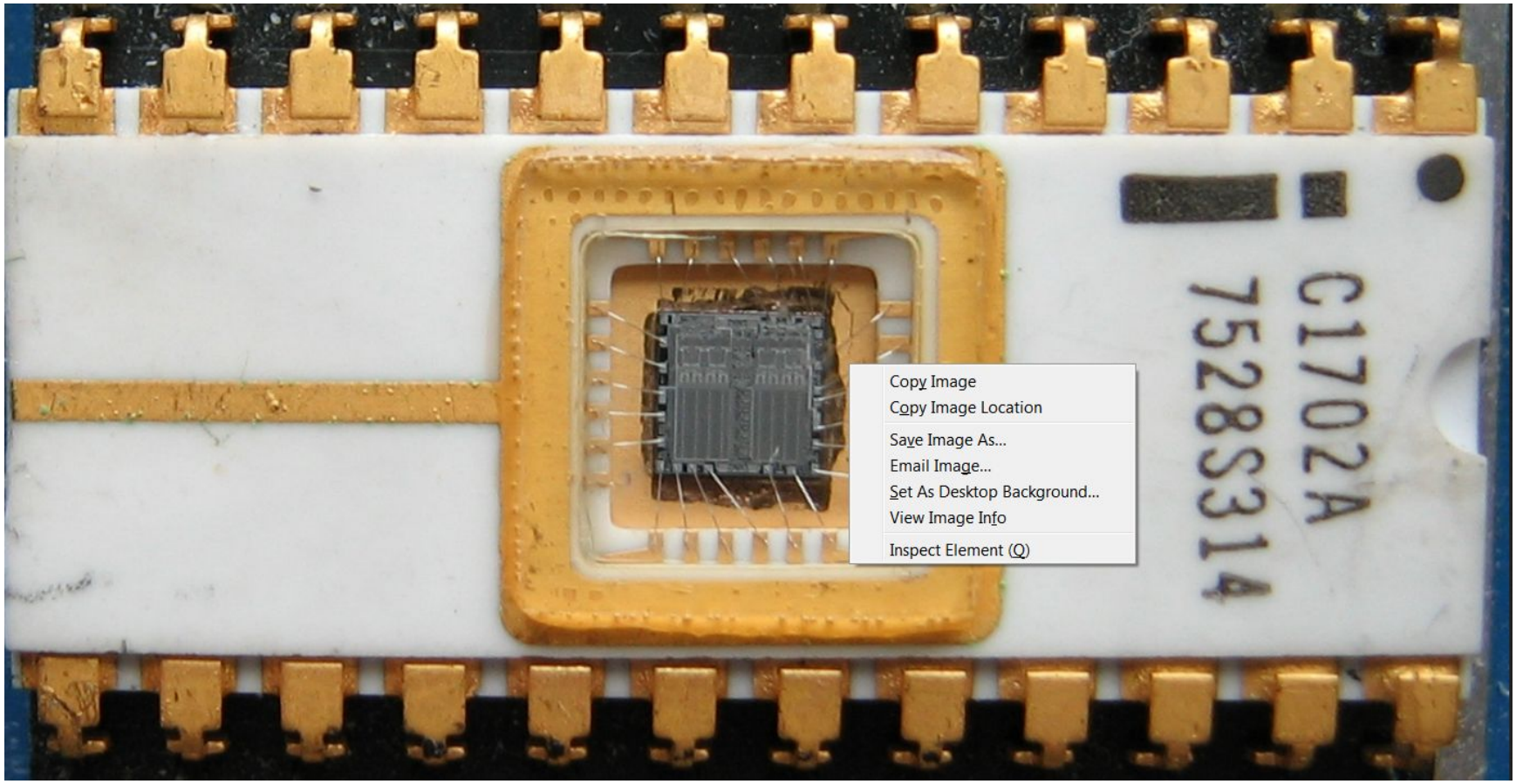


Многократно программируемая память

- **EPROM*(Erasable Programmable Read Only Memory)*** Многократно программируемая со стиранием ультрафиолетовыми лучами.
  - В EPROM в качестве перемычки используется транзистор с плавающим затвором. Для стирания матрица элементов облучается ультрафиолетовым светом, в результате заряд на затворах транзисторов стирается.

- **EEPROM *(Electrically Erasable Programmable Read-Only Memory)*** *многократно программируемая электрически стираемая память*. В качестве элемента хранения используется транзистор с плавающим затвором.

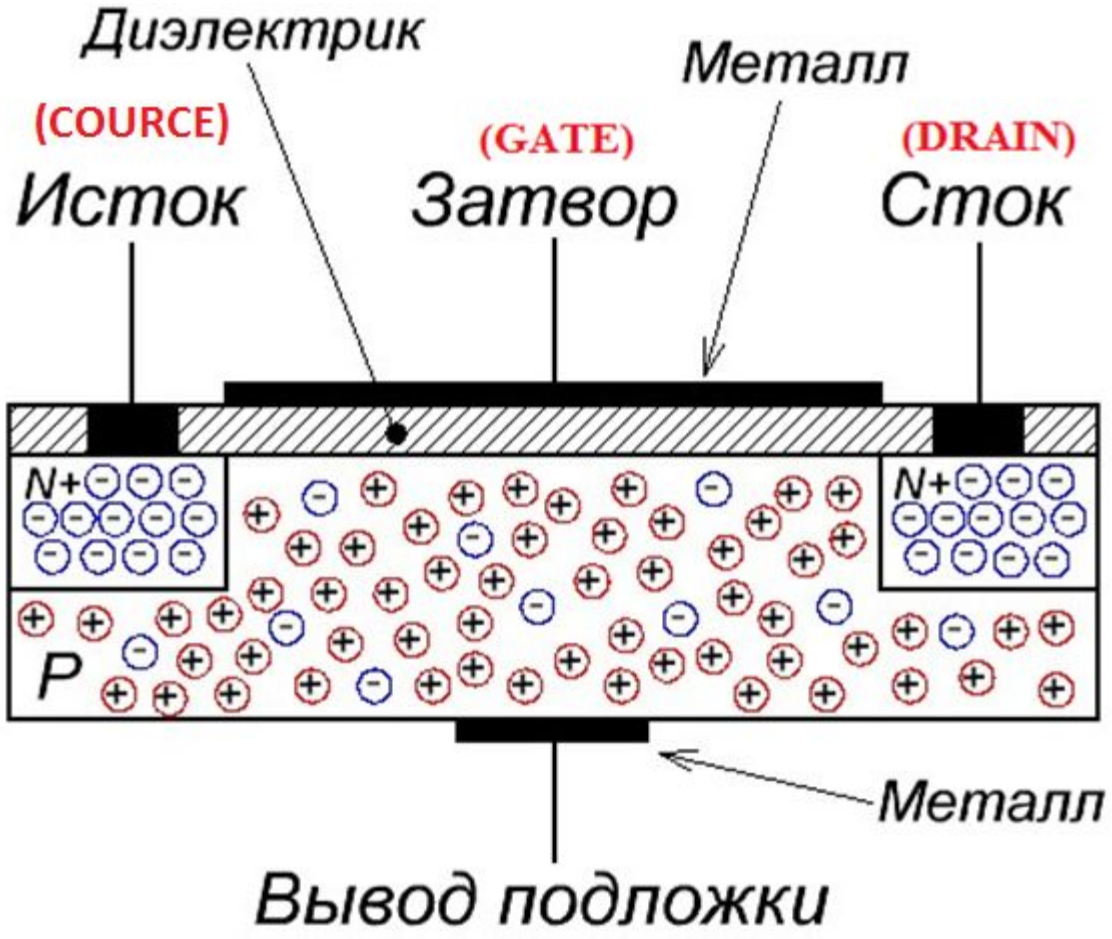
EPROM



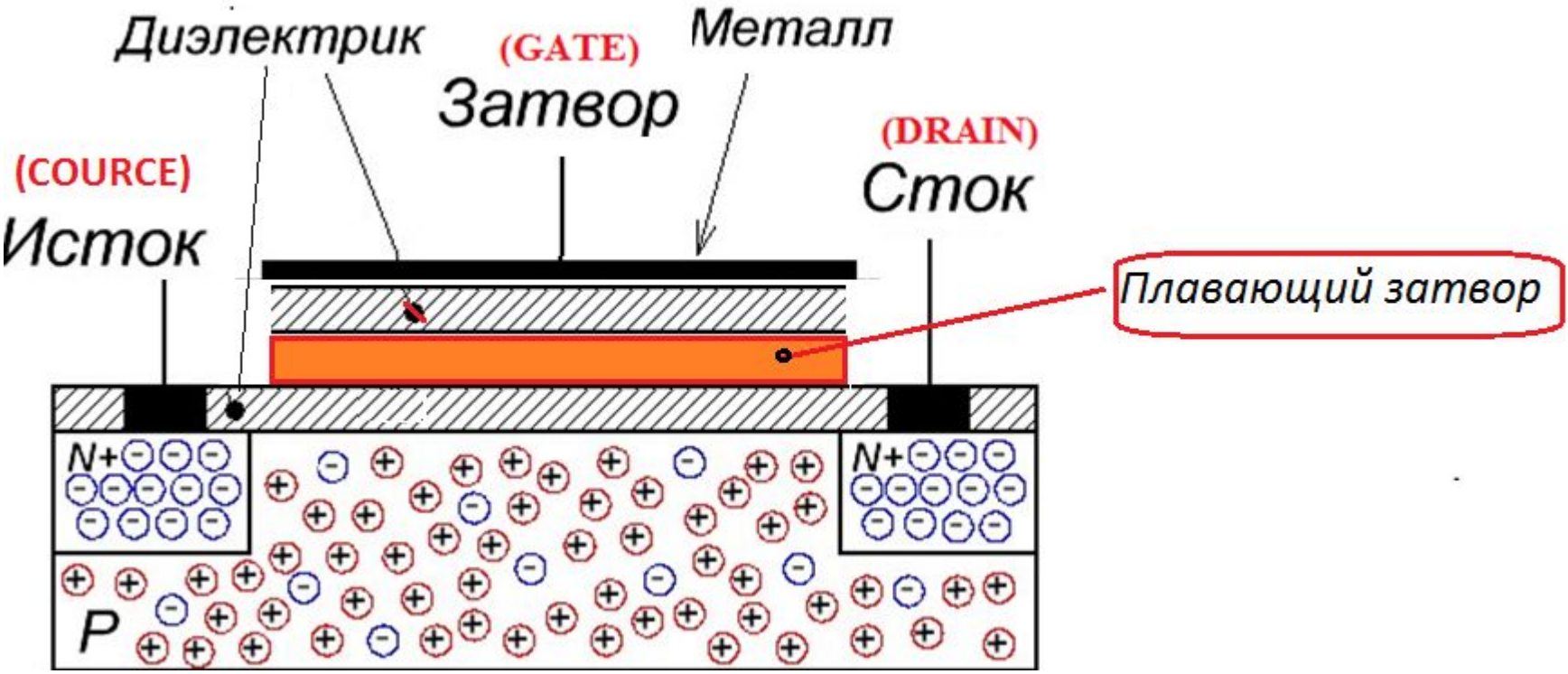
Восстановление прозженных перемычек производится с помо
засветки ячеек ультрафиолетовым источником света

Flash память

Полевой транзистор

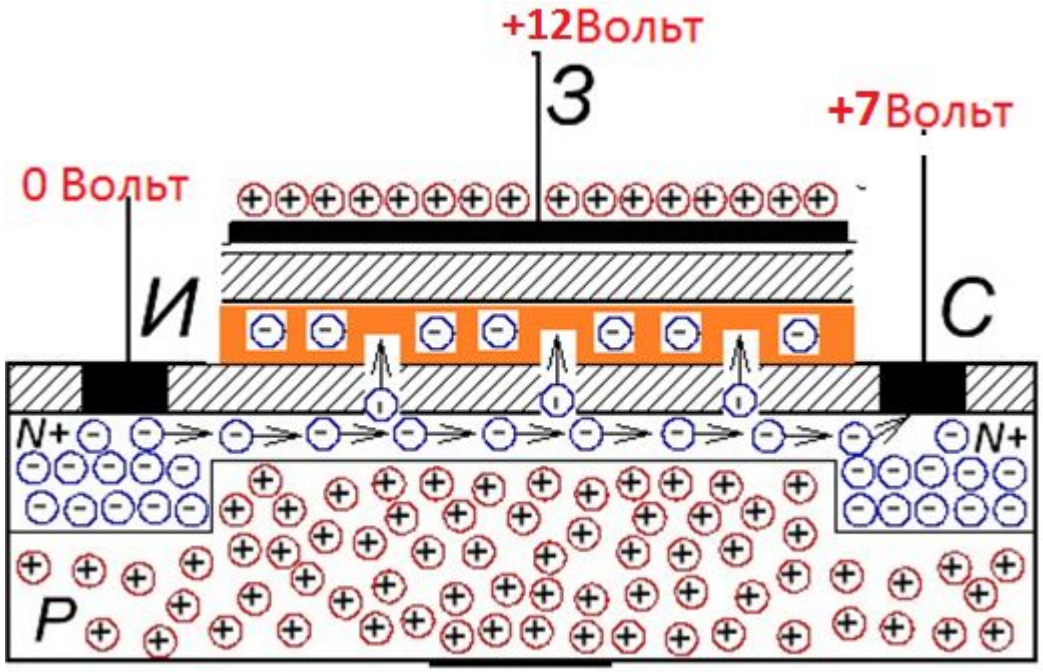


Транзистор с плавающим затвором



- Плавающий затвор изолирован двумя слоями диэлектрика

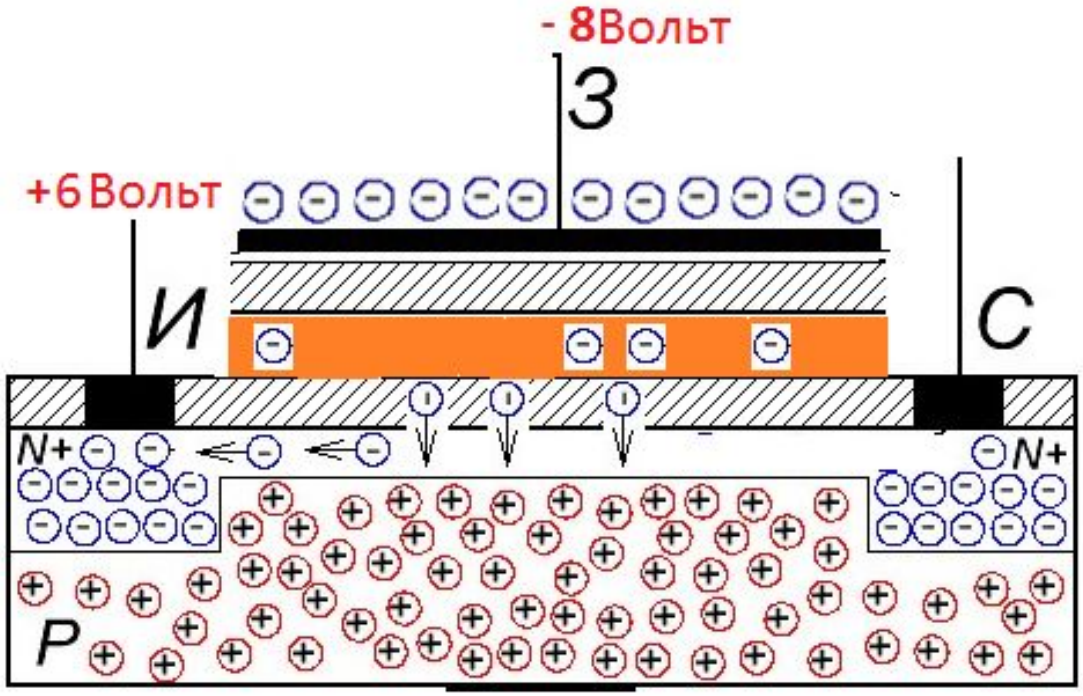
Запись



- При подаче на затвор более положительного напряжения чем на сток происходит перенос части электронов через диэлектрик в область плавающего затвора

Величина заряда зависит от длительности и амплитуды
управляющего импульса записи на затворе

Стирание

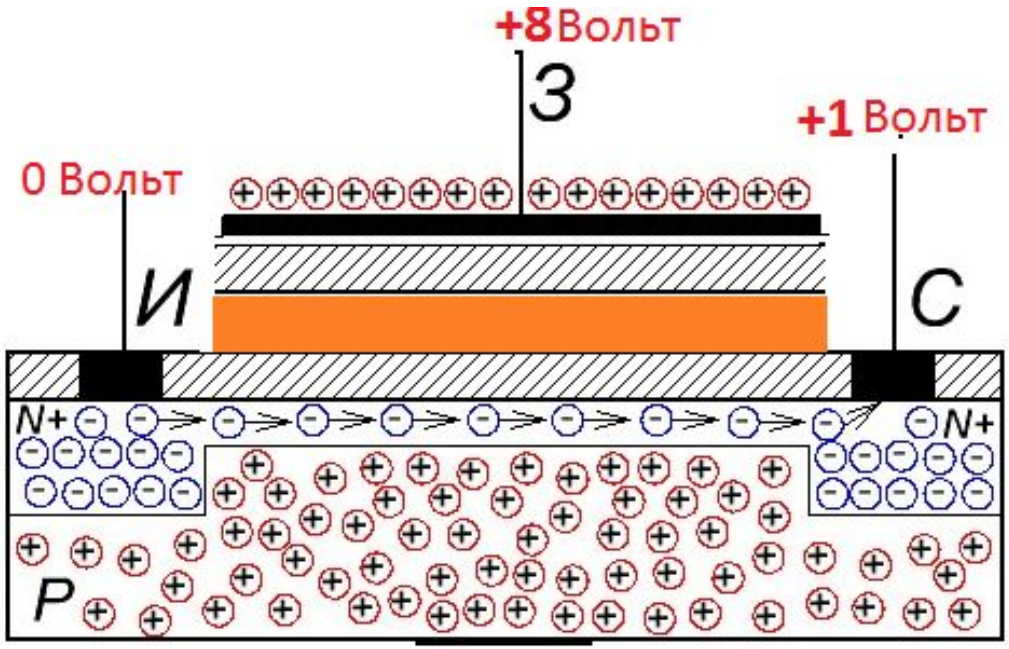


- При подаче на затвор более отрицательного напряжения относительно истока происходит стекание электронов в область истока

Ячейка флэш-памяти

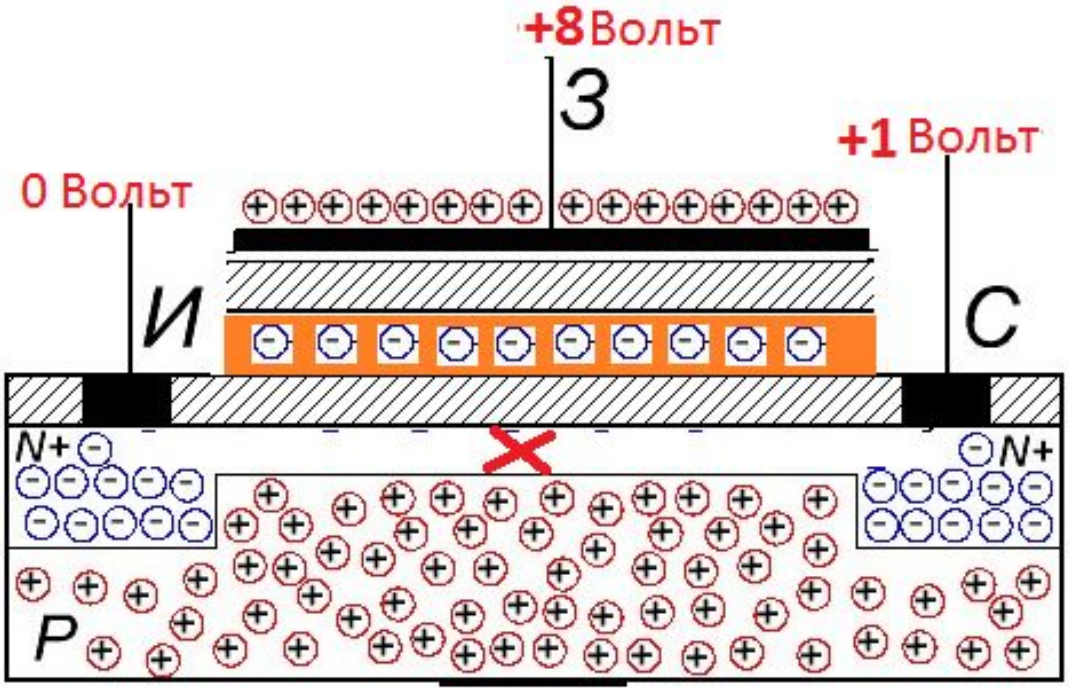
- Операции записи и чтения приводят к разрушению(износу) и диэлектричсекого слоя и его постепенной деградации.
- Срок службы ячейки ограничен.

Чтение нуля



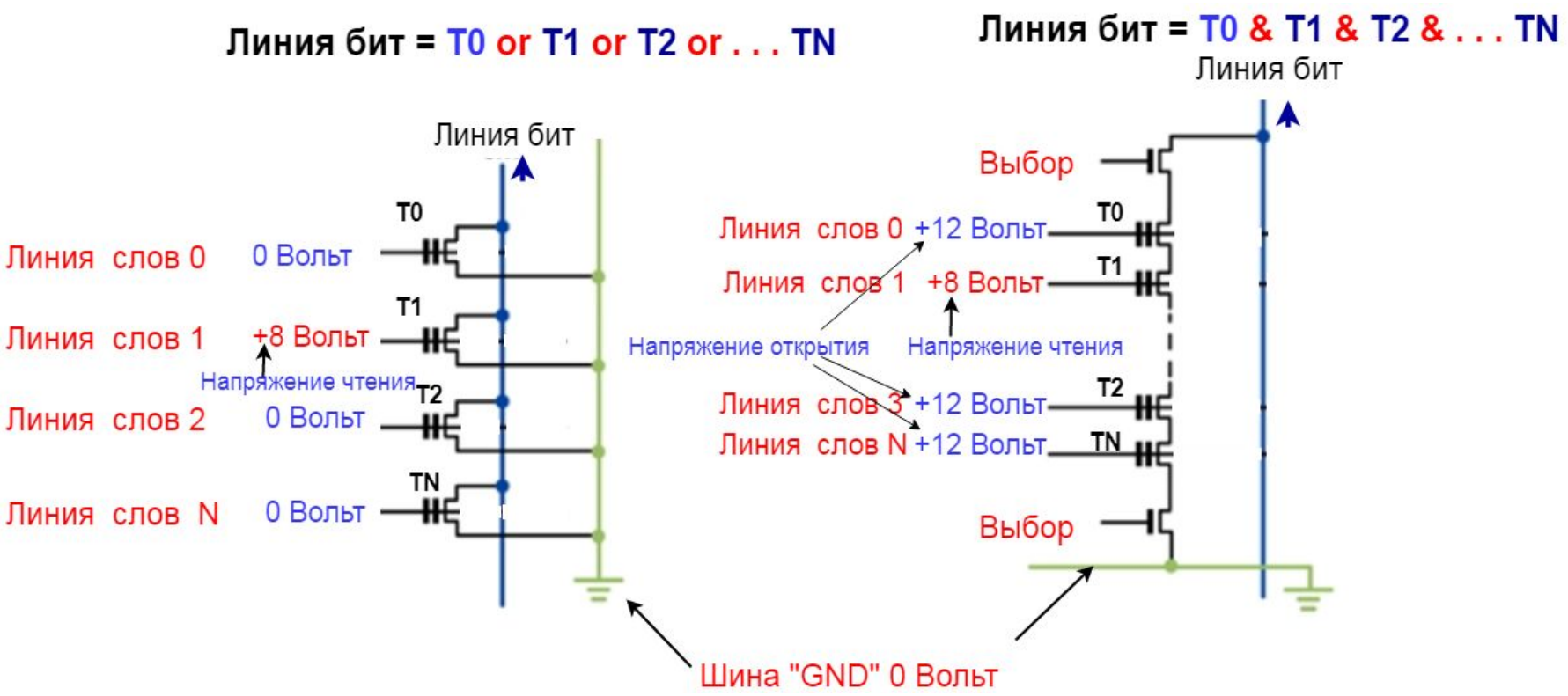
- **На плавающем затворе заряда нет**
- При подаче напряжения на затвор в отсутствие заряда на плавающем затворе в транзисторе создается канал и начинает течь ток
- Транзистор открыт – это логический ноль

Чтение единицы



- На плавающем затворе есть заряд
- При подаче напряжения на затвор, заряд на плавающем затворе компенсирует заряд на затворе, канала нет, тока нет.
- Транзистор закрыт – это логическая единица

NOR и NAND


Линия бит = T0 or T1 or T2 or . . . TN
Линия бит
T0
Линия слов 0
0 Вольт
T1
Линия слов 1
+8 Вольт
Напряжение чтения
T2
Линия слов 2
0 Вольт
TN
Линия слов N
0 Вольт
Линия бит = T0 & T1 & T2 & . . . TN
Линия бит
Выбор
T0
Линия слов 0 +12 Вольт
T1
Линия слов 1 +8 Вольт
Напряжение открытия
Напряжение чтения
T2
Линия слов 3 +12 Вольт
TN
Линия слов N +12 Вольт
Выбор
Шина "GND" 0 Вольт

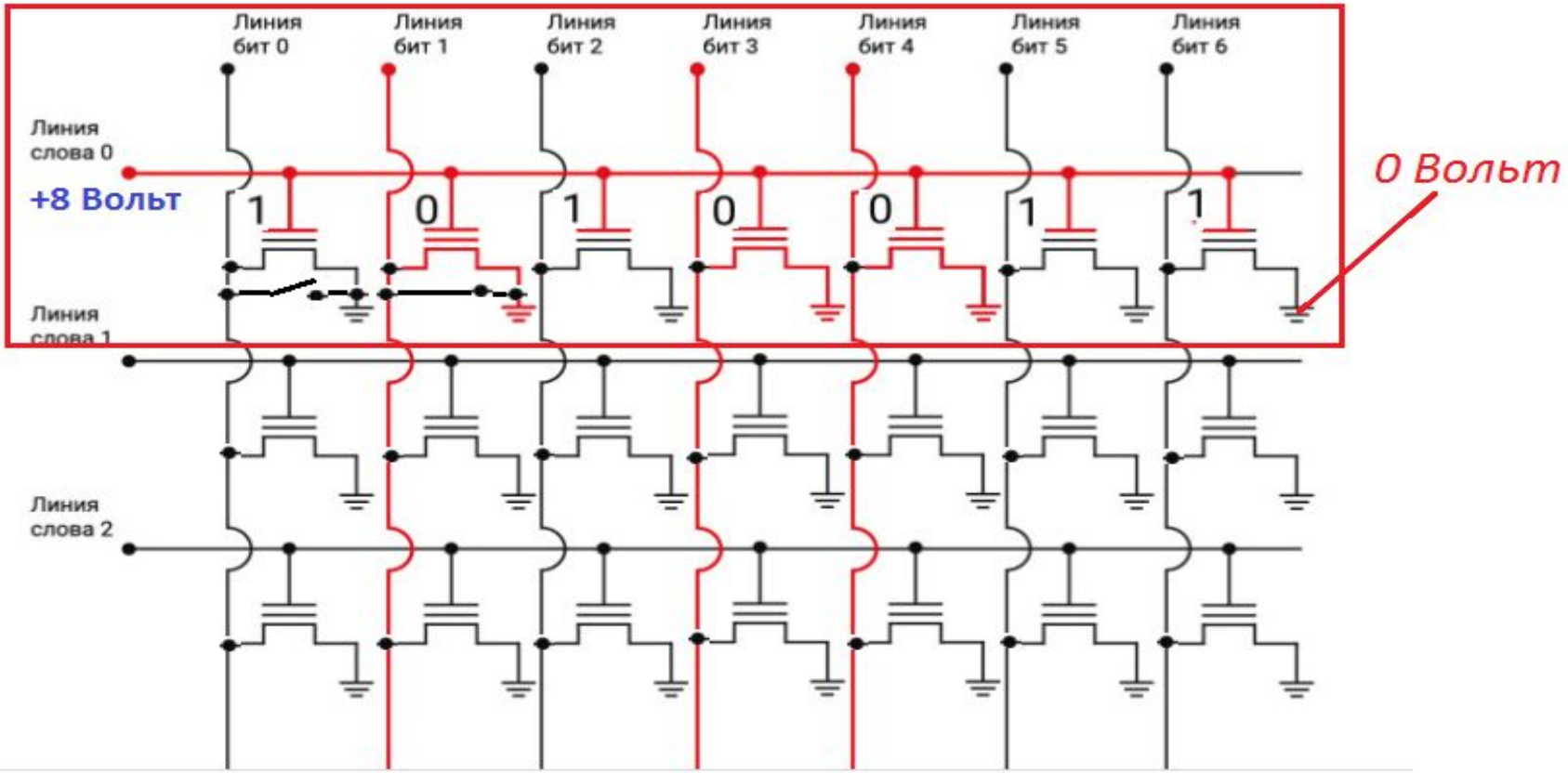
NOR - ячейка

- NOR
- При подаче напряжения чтения на линию слов транзисторы, содержащие заряд на плавающем затворе, то есть хранящие «единицу», не откроются и ток не потечет. По наличию или отсутствию тока на линии бита делается вывод о значении бита.
- При выборе строки (+8 Вольт подается на линию слов):
  - транзистор, хранящий заряд закрыт, тока нет, на битовой линии 1
  - транзистор не хранящий заряд открыт ток есть и 0 вольт поступает на битовую линию
- Можно получить доступ к любой конкретной ячейке
-

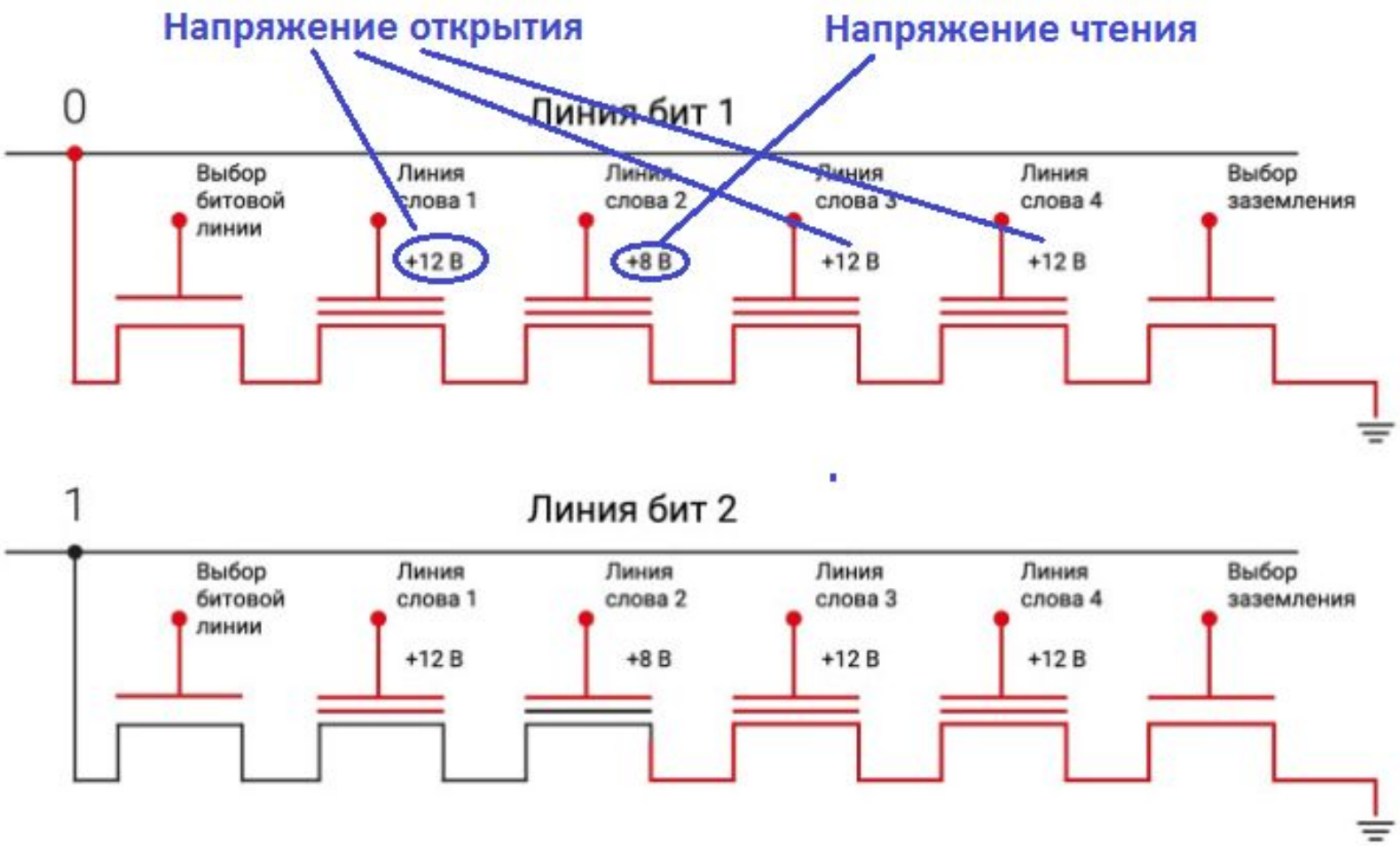
NAND-ячейка

- NAND
- На необходимую линию слова подается напряжение чтения, а на все остальные линии слова подается напряжение открытия, которое открывает транзистор вне зависимости от уровня заряда в нем.
- Так как все остальные транзисторы гарантированно открыты, то наличие напряжения на битовой линии зависит только от одного транзистора, на который подано напряжение чтения:
  - транзистор, хранящий заряд закрыт, тока нет, на битовой линии 1
  - транзистор не хранящий заряд открыт ток есть и 0 вольт поступает на битовую линию
  - В случае с NAND несколько одиночных ячеек памяти соединены последовательно и для того, чтобы записать состояние «ноля» в одну из них, надо, чтобы все другие были открыты и пропускали ток, т.е. нельзя обратиться только к одной ячейке

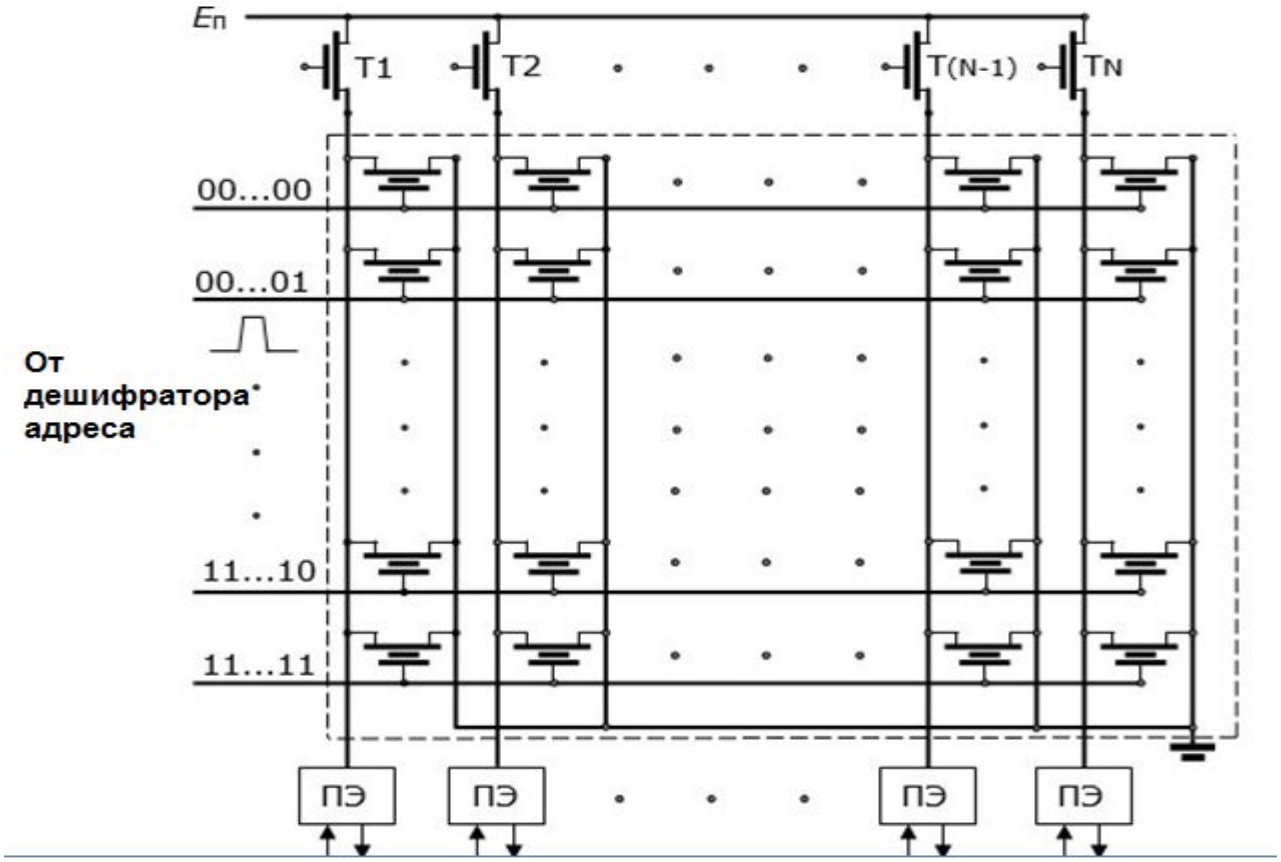
Архитектура NOR (параллельное включение транзисторов)



Архитектура NAND



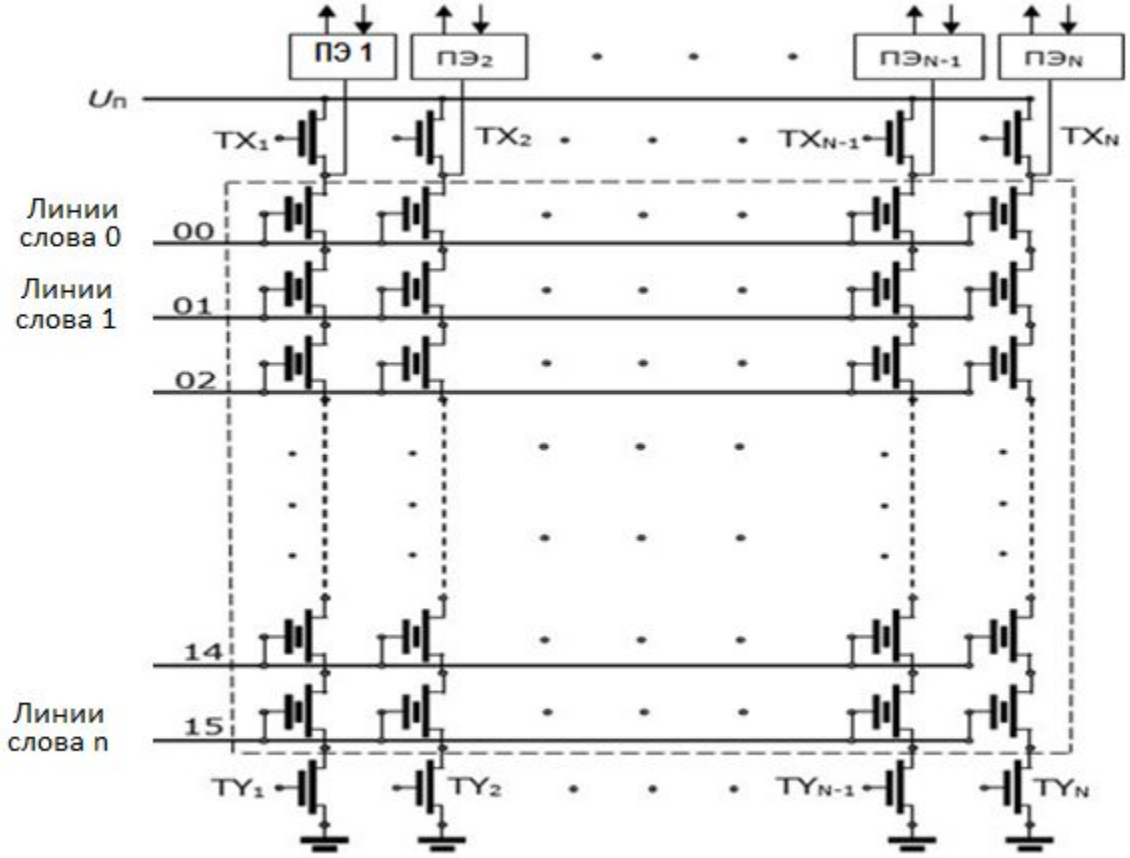
Архитектура NOR (параллельное включение транзисторов)



Операция стирания соответствует записи единиц во все ячейки

т.е исходное «чистое» состояние NOR памяти все единицы.

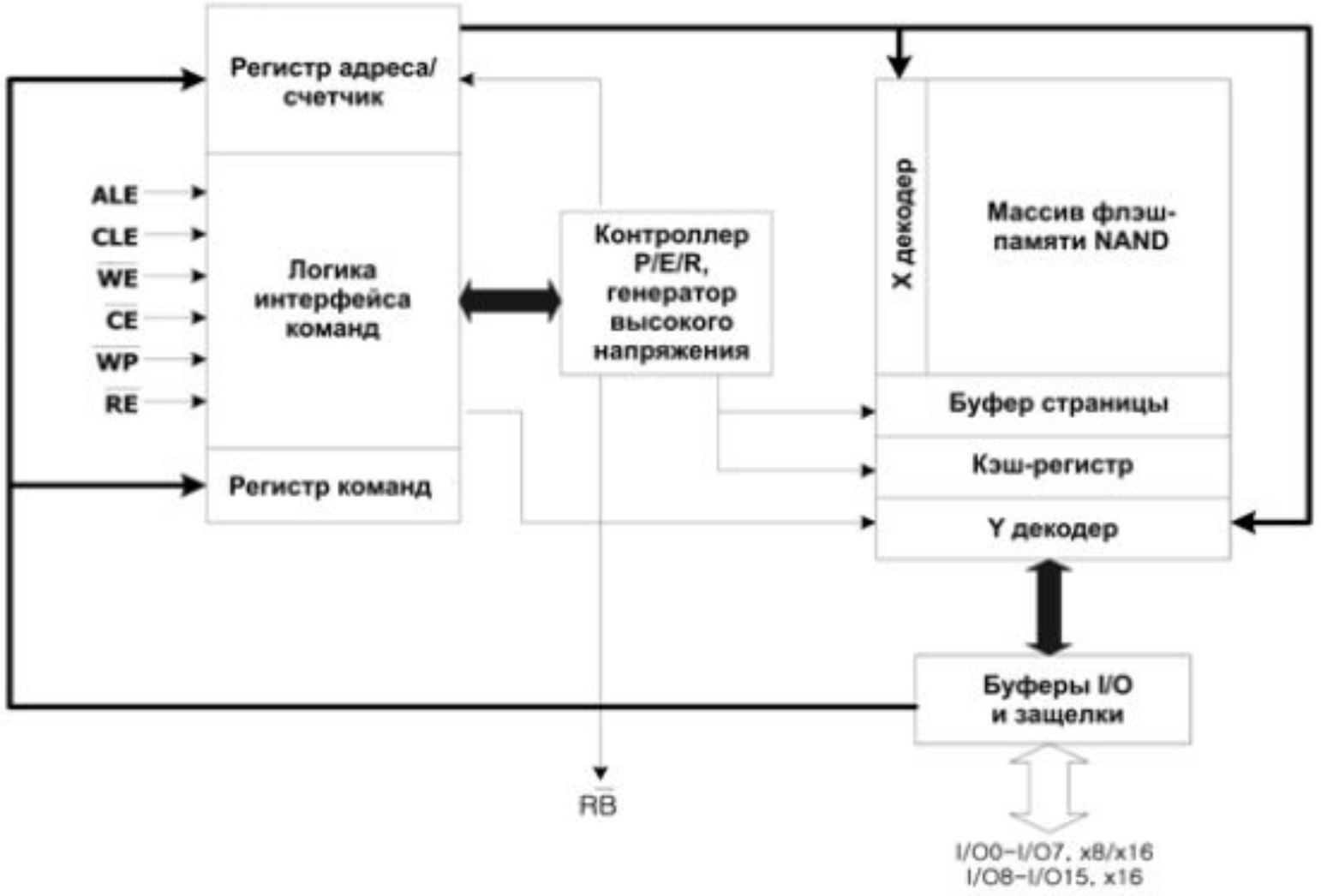
Архитектура NAND (последовательное включение транзисторов)



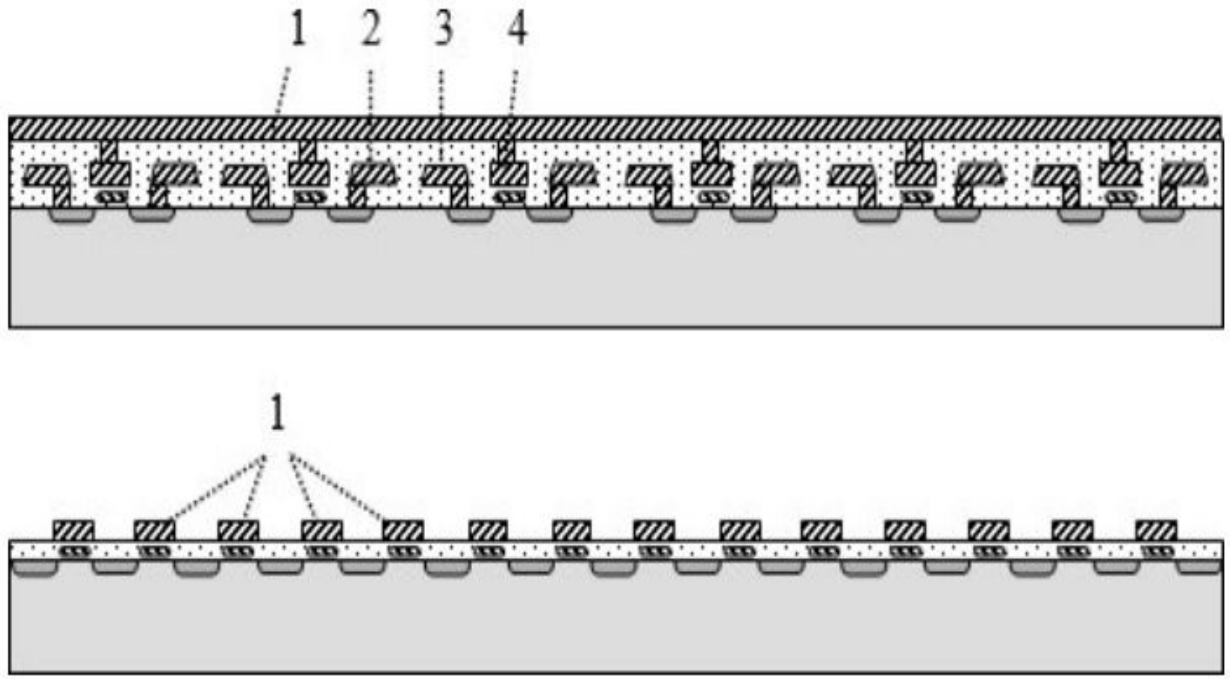
- При считывании управляющее напряжение подается на все элементы выбранного столбца
- Операция стирания соответствует записи нулей во все ячейки

  т.е. исходное «чистое» состояние все нули

Структура NAND-микросхемы



Топологии ячеек



Поперечное сечение пластины кремния с матрицами флеш-памяти типа "NOR" (вверху) и "NAND" (внизу): 1 – горизонтальная шина; 2 – вертикальная шина истоков; 3 – вертикальная шина стоков; 4 – управляющие затворы транзисторов

Характеристика NOR

- Достоинства NOR :
  - Возможность произвольного доступа к любой ячейке памяти (*что позволяет использовать ее для записи и хранения программного кода, который не требует частого перезаписывания , например, BIOS, память программ встроенных систем*)
  - Меньшее время случайного чтения

- Недостатки NOR :
  - Плохая масштабируемость и как следствие более высокая стоимость.
  - Большее время записи/стирания (чтобы стереть надо сначала обнулить, а потом перевести в исходное состояние все единицы)
  - Меньшее число циклов перезаписи – 100 000

Характеристика NAND

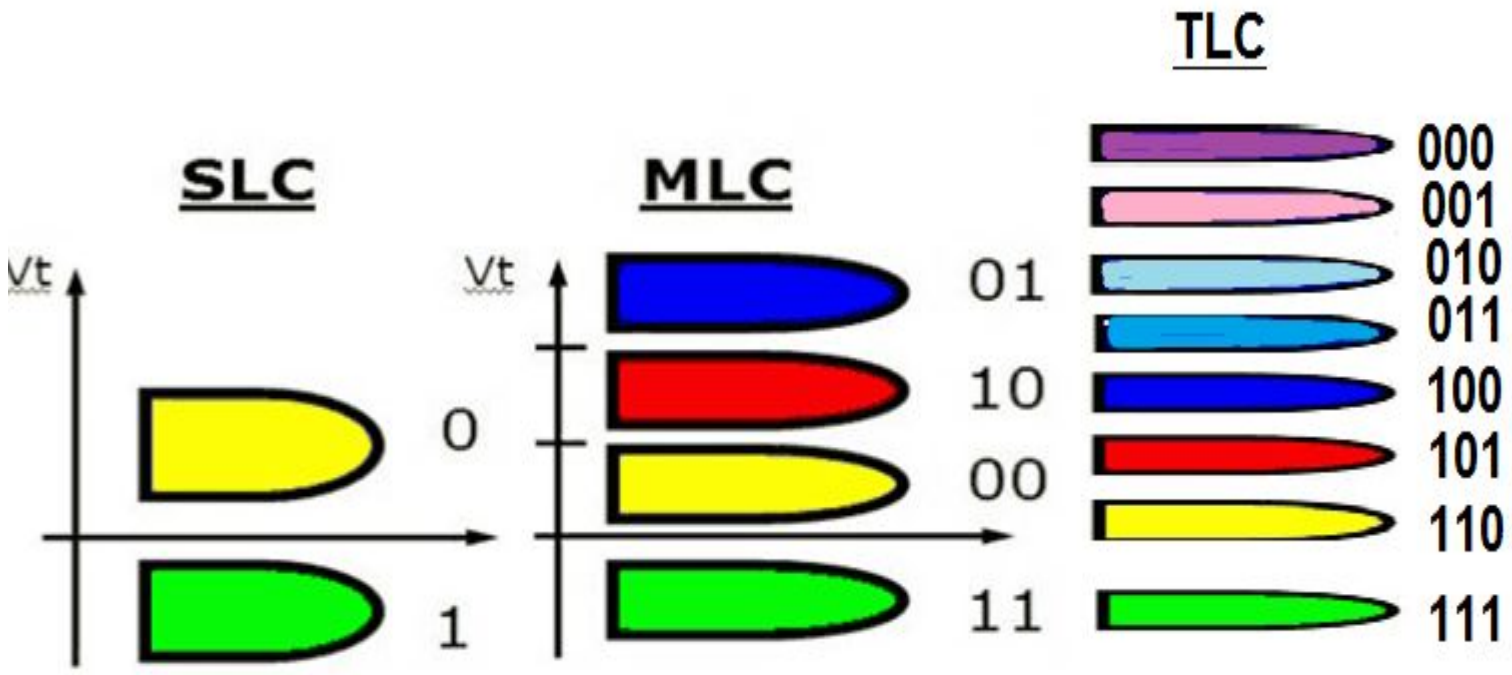
- Достоинства NAND:
  - Хорошая масштабируемость и как следствие малая стоимость.
  - Меньшее время записи/ стирания
  - Большее количество циклов записи стирания 1000000
  - Используется в устройствах хранения данных (SSD диски, флэш-память)

- Недостатки NAND
  - **Нельзя обратиться к отдельной ячейке, поэтому обращение происходит блоками.**
  - Малое время последовательного чтения, но большое время случайного чтения

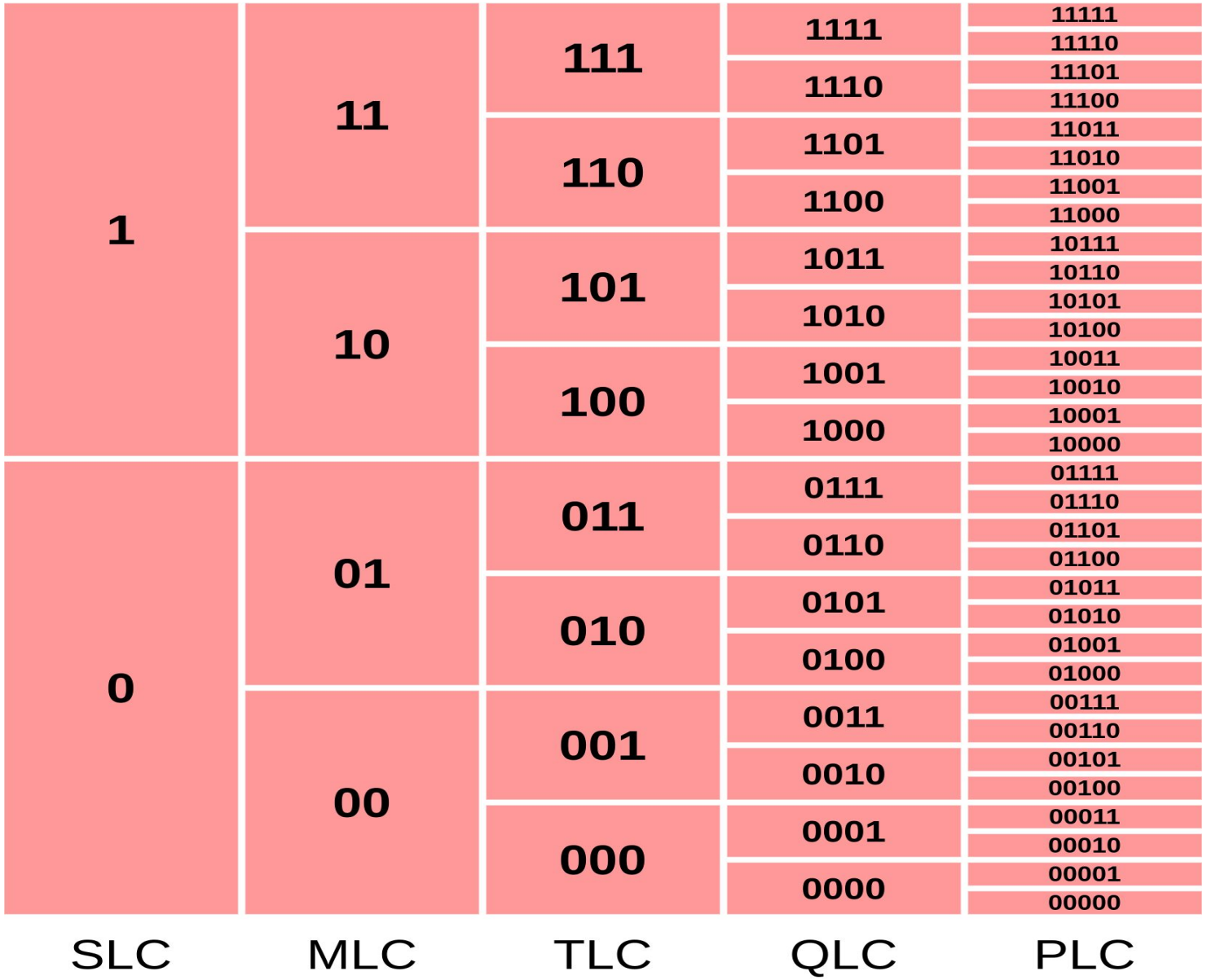
Сравнение NAND и NOR

Параметр	NAND	NOR
Емкость	~ 1 Гбит (2 кристалла в корпусе)	~ 128 Мбит
Напряжение питания	2.7 – 3.6 В	2.3 – 3.6 В
Ввод/вывод	x8 / x16	x8 / x16
Время доступа	50 нС (цикл последовательного доступа) 25 мкС (случайный доступ)	70 нС (30 пФ, 2.3 В) 65 нС (30 пФ, 2.7 В)
Скорость программирования (типовая)	- 200 мкС / 512 байт	8 мкС / байт 4.1 мС / 512 байт
Скорость стирания (типовая)	2 мС / блок (16 кБ)	700 мС / блок
Совокупная скорость программирования и стирания (типовая)	33.6 мС / 64 кБ	1.23 сек / блок (основной: 64 кБ)
Количество циклов перезаписи	1000 000	100 000

Одноуровневые (Single Level Cell) и многоуровневые ячейки (Multi Level Cell, Three Level Cell)



Многоуровневые ячейки 3D

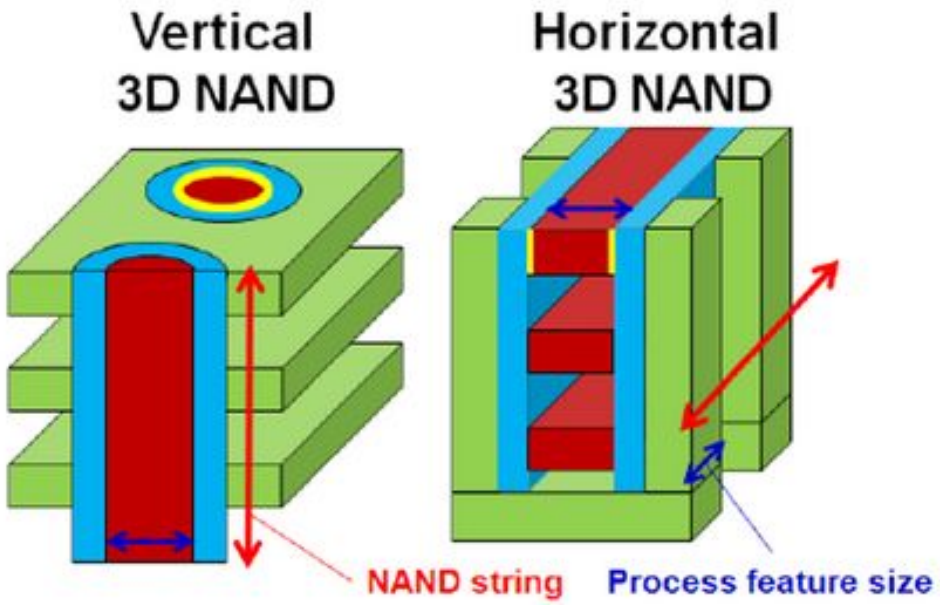


Сравнение SLC и MLC

- **SLC** записывают только **один бит** в ячейку и это обеспечивает **до 10 раз лучшую** долговечность и **до 2-х раз** более **высокую скорость** в сравнении с **MLC**.

	SLC	MLC
Чтение страницы, мкс	25	50
Стирание блока, мс	2	2
Запись страницы, мкс	250	900
Количество циклов, раз	1 000 000	100 000

3D флэш память (флэш-трубки)



- Технология 3D NAND позволяет увеличить плотность ячеек флэш памяти

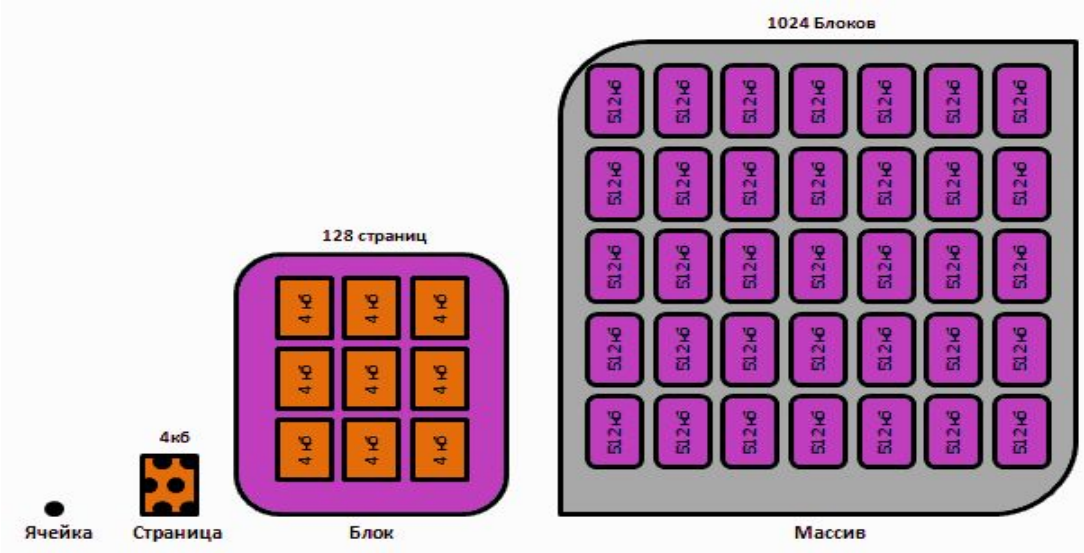
Операции при обращении к флэш -памяти

- Возможны операции:
  - Стирание (ограниченное количество раз);
  - Запись ограниченное количество раз;
  - чтение (бесконечное число раз);
  - Чтение – модификация – запись (количество раз как и при записи).

Запись возможна только в «чистое» место памяти, поэтому ей всегда предшествует операция стирания.

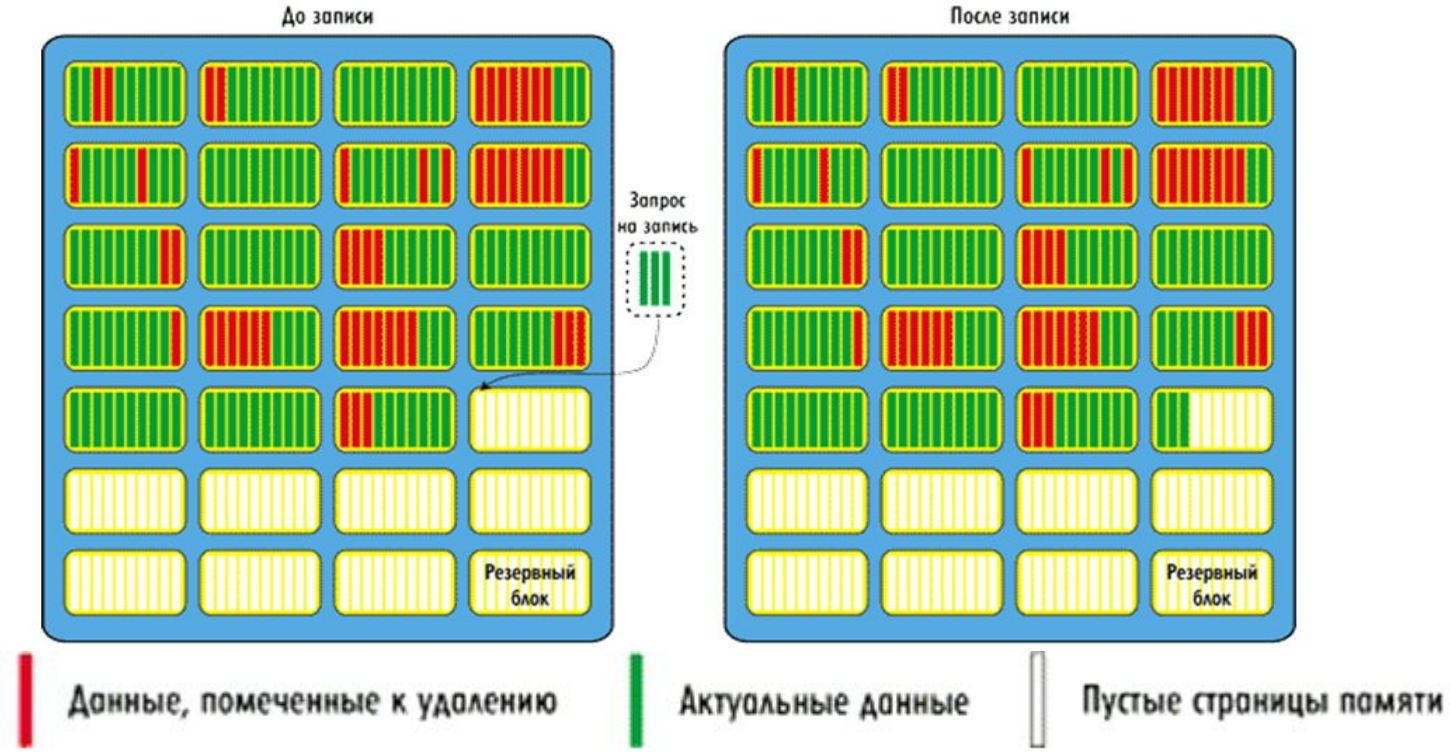
Операция стирания самая медленная (для повышения быстродействия операция стирания осуществляется сразу над блоком ячеек, а не над одной ячейкой).

Структурная организация SSD – диска (Solid State Drive)



- SCL и MCL ячейки объединяются в страницы по 4кб.
- 128 страниц объединены в блок объемом 512 КБ, а 1024 блока в массив 512 МБ
- Информация записывается и читается в память страницами по 4 Кб, а стирается блоками по 512 Кб.

Запись на свободный диск

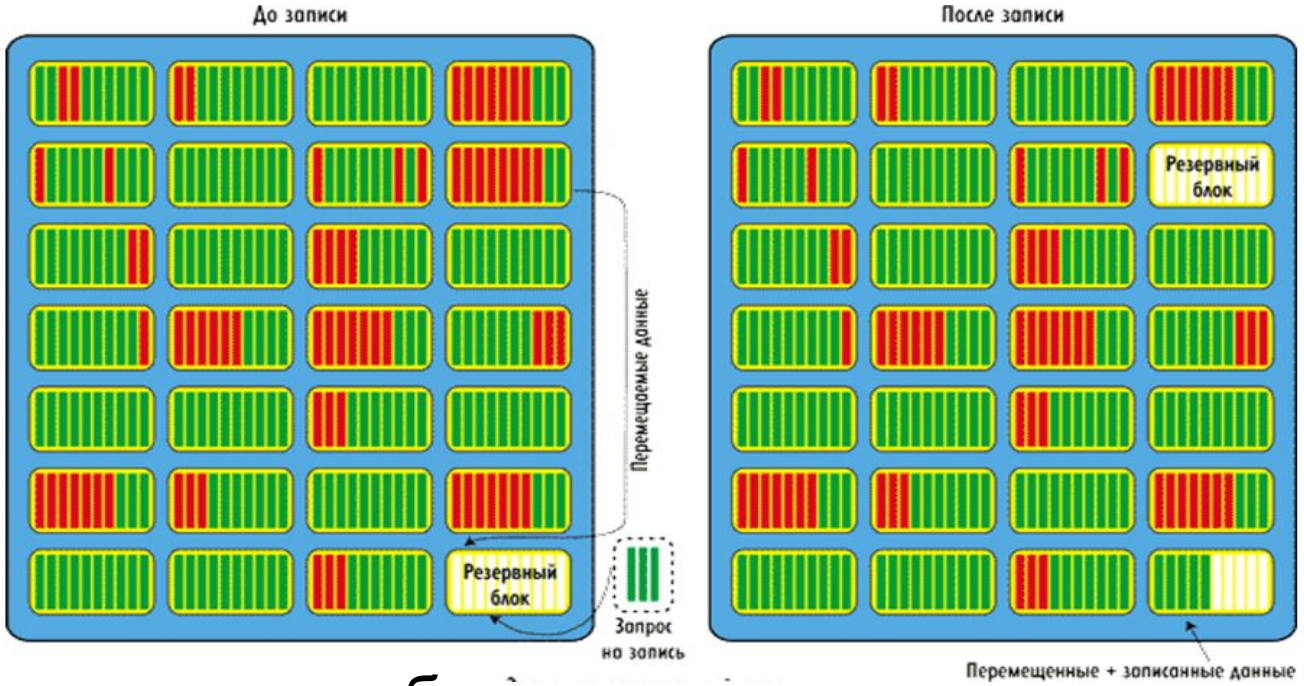


- Данные записываются последовательно в физические блоки порциями по 4 Кбайт.
- При этом логический адрес записываемой страницы сопоставляется с физическим адресом,
- Для соответствия между логическими и физическими адресами (LBA-PBA mapping) предназначена специальная таблица, которая размещается в оперативной памяти SSD-диска.

Перезапись данных

Если производится перезапись данных, они последовательно записываются в следующие по порядку свободные страницы памяти, а в таблице соответствия логических и физических адресов те страницы, в которые эти данные были записаны ранее, помечаются как содержащие устаревшие данные (помечаются к удалению).

Запись на заполненный диск



При заполнении диска блоки памяти могут содержать как страницы, помеченные на удаление (страницы с устаревшими данными), так и страницы с актуальными данными, которые удалять нельзя.

Контроллер диска анализирует объем записываемой информации находит блок, содержащий максимальное количество страниц, помеченных к удалению, достаточных для размещения вновь записываемых страниц.

Запись на заполненный диск

- Страницы с актуальными данными из выбранного блока переносятся в пустой или резервный блок, куда дозаписываются вновь поступившие данные.
- После этого информация в «старом блоке стирается» и он становится резервным, доступным для записи.

  Процедуру поиска подходящего блока с максимальным количеством неиспользуемых страниц называют сбором мусора» (Garbage Collection).

-

Резервные блоки

- Разница между двоичным и десятичным значением емкости дает резервные блоки
- Так для диска емкостью 160 Гб емкость резервных блоков составит 171 798 691 840 - 160 000 000 000 = 11 798 691 840 байт, или примерно 11 Гбайт

- Если запас резервных блоков исчерпан, то для временного хранения перемещаемых блоков может использоваться внутренняя динамическая кэш память контроллера

Для продления срока службы SSD необходимо постоянно иметь (10 – 15)% свободного места.

Механизм Wear Leveling

- **Wear Leveling** обеспечивает равномерное использование всех ячеек памяти и как следствие повышает долговечность SSD. Контроллер SSD-диска отслеживает частоту использования различных блоков памяти.

- Если какие то блоки памяти используются реже остальных, то он принудительно повышает частоту их использования путем перезаписи этих блоков данных в другие блоки, высвобождая их для дальнейшего использования.

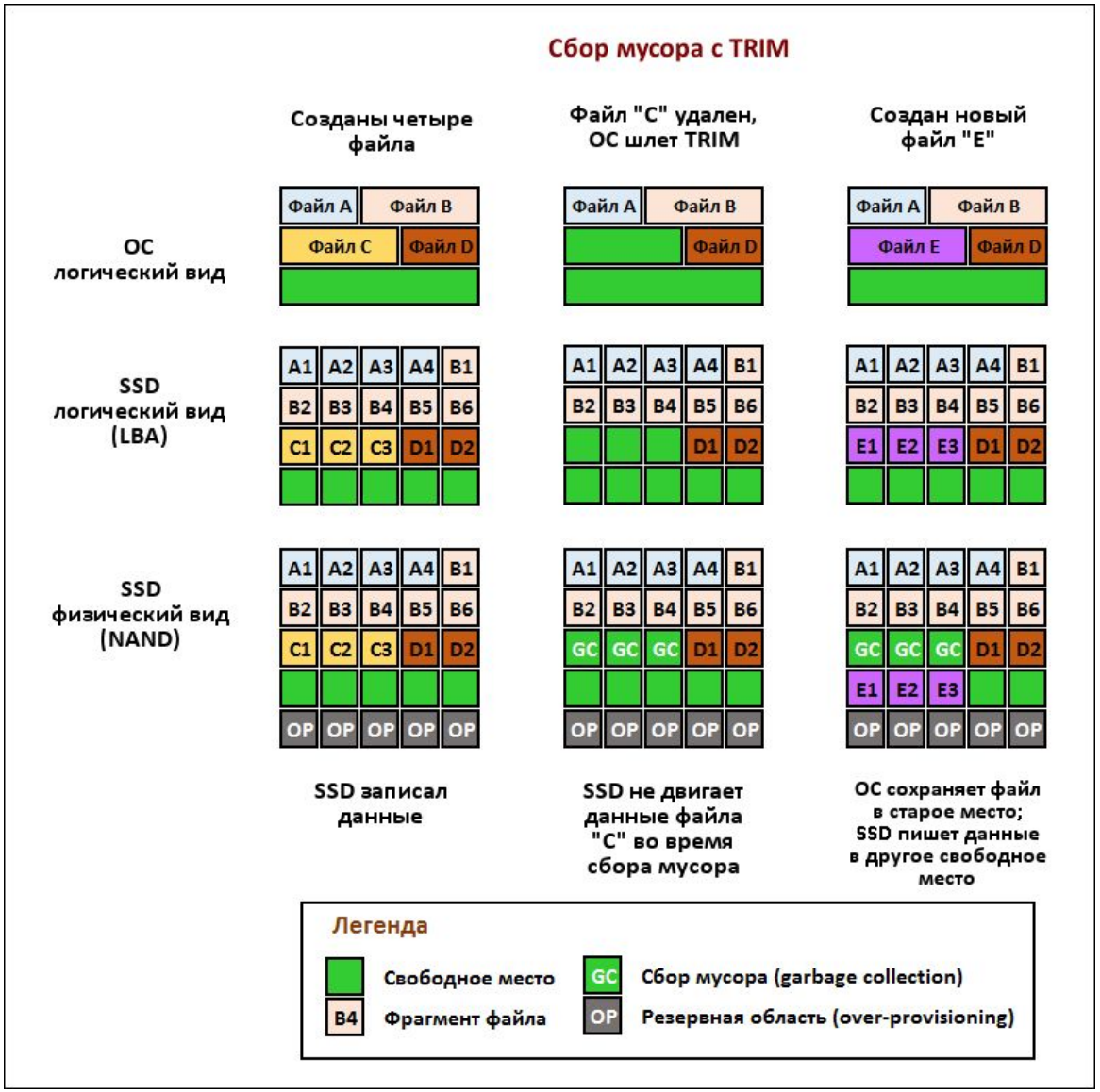
Команда TRIMM

- При удалении файлов операционная система лишь логически удаляет ненужные файлы. При этом физически они остаются на носителе.

- В SSD это приводит к накоплению «мусора», т.к. контроллеру не известно, что файл, был помечен операционной системой как удаленный.

- TRIM — это команда осуществляющая функцию передачи информации SSD о том, что файл был удален и соответствующие страницы памяти помечаются к удалению и могут применяться в процедуре Garbage Collection.

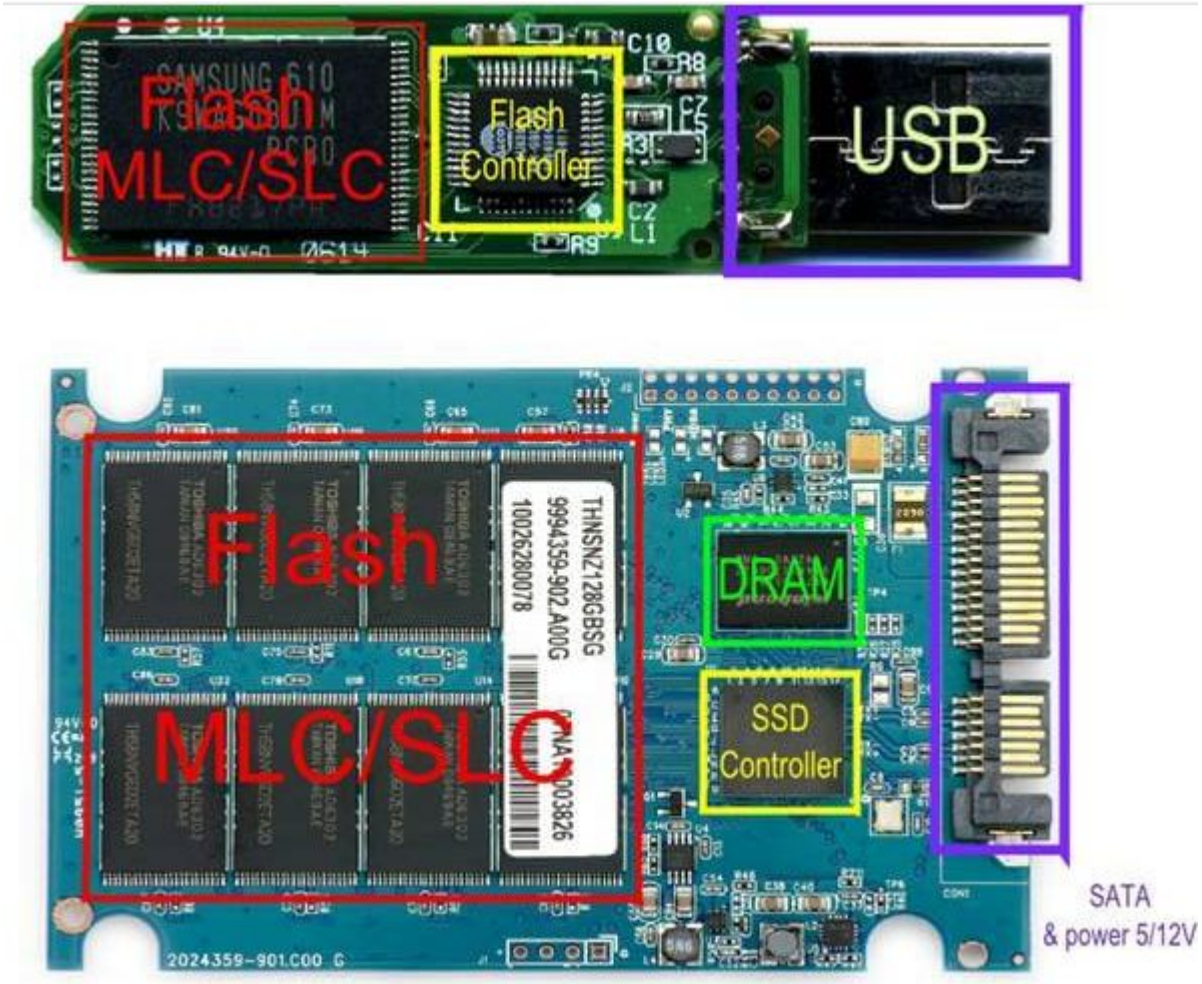
Команда TRIMM



Команда TRIMM

- Для работы команды TRIM необходима поддержка со стороны как ОС, так и SSD-диска.
- Команда TRIM встроена в ОС начиная с Windows 7, а также в ядре Linux начиная с ревизии 2.6.28.
- В операционной системе Windows 7 команда TRIM используется в таких операциях, как удаление файлов и форматирование.

SSD -диск



В SSD-диска входит поле Flach – памяти, SSD-контроллер, блок динамической DRAM-
памяти

Архитектура SSD - диска

Состав контроллера:
ARM – процессор. Отвечает за равномерность износа ячеек в Flash, диагностику SMART, кеширование, безопасность.
Error Correction (ECC) – блок контроля и коррекции ошибок ECC.
Flash контроллер – включает адресацию, шину данных и контроль управления микросхемами Flash памяти.
DRAM контроллер - управление DDR кэш памятью.
Преобразователь интерфейса– отвечает за интерфейс передачи данных на внешние интерфейсы SATA, SAS, PCIe, USB, Thunderbolt «удар молнии»(до 10 Гбит/сек по медному проводу и до 20 Гбит/сек при использовании оптического кабеля) замена USB/
Динамическая кэш-память для хранения таблиц размещения и занятости ячеек диска. Параллельно в ней может храниться временная информация со стираемых ячеек, при нехватке пустого места на диске.
В нем так же хранится информация о частоте и интенсивности использования каждого доступного блока на диске. Кроме того, здесь записаны адреса «мест», где невозможно осуществить запись, ввиду физического износа.
Поле флэш – памяти.

Кэш - память

- В **SSD** накопителях применяется кэш память в виде энергозависимой **DRAM** микросхемы, наподобие как в жёстких дисках. Часть прошивки и самые часто изменяющиеся данные находятся в ней, сокращая износ энергозависимой **NAND** памяти.
- В некоторых SSD – накопителях динамическая память может отсутствовать

Характеристики SSD накпителей

- Тип – внутренний, внешний, внутренний/ внешний (универсальный);
- Емкость;
- Форм - фактор: - 1.8, 2.5, 3.5 дюйма
- Интерфейс: SATA,SAS, PCIe, Thunderbolt и др. определяет внешнюю скорость передачи данных.
- Для SATA поддержка NCQ - ***Native Command Queuing*** — аппаратная установка очередности команд) — технология, использующаяся в SATA-устройствах для повышения быстродействия
- Объем буфера обмена
- Объём собственной оперативной памяти накопителя
- Тип NAND: памяти (SLC,MLC,TLC)
- Скорость записи: скорость записи данных из контроллера SSD в флэш- память для записи.
- Скорость чтения: скорость чтения данных из микросхем флэш - памяти в контроллер SSD.

Характеристики SSD накпителей

- Скорость случайной записи блоками определенного размера : измеряется в IOPS(Input/Output Operations Per Second - «ай-опс») -количество операций ввода-вывода в секунду
- Поддержка TRIMM
- Ударостойкость при работе параметр, определяющий стойкость накопителя к падениям и сотрясениям в процессе работы. Измеряется в G — единицах перегрузки,
- Наработка на отказ : гарантированное (минимальное) время безотказной работы накопителя.
- Внешний карман : внешним карманом комплектуются накопители типа «внешний/внутренний» . Технически карман представляет собой оснащённый USB-коннектором корпус, в который устанавливается собственно накопитель — таким образом внутренний накопитель можно, при необходимости, использовать как внешний.

Характеристики SSD накопителей

Общие характеристики	Kingston SSDNow KC300 180Gb SKC300...	Найти похожие
Линейка	SSDNow KC300	
Тип	SSD	
Поддержка секторов размером 4 Кб	+	
Тип флэш-памяти	MLC	
Назначение	для настольного компьютера	
Поддержка TRIM	+	
Контроллер	SandForce SF-2281	
Background Garbage Collection	+	
Тип интерфейса NAND Flash памяти	Toggle DDR 1.0	
Форм-фактор	2.5"	

Характеристики накопителя	Kingston SSDNow KC300 180Gb SKC300...	Найти похожие
Емкость	180 Гб	
Скорость чтения	525 Мб/с	
Скорость записи	500 Мб/с	

Характеристики SSD накопителей

Скорость случайной записи (блоки по 4 Кб)	64000 IOPS
Подключение Kingston SSDNow KC300 180Gb SKC300…	Найти похожие
Интерфейс	SATA III 6Gb/s
Внешняя скорость передачи данных	600 Мб/с
Поддержка NCQ	+
Механика и надежность Kingston SSDNow KC300 180Gb SKC300…	Найти похожие
Ударостойкость	1500 G
Время наработки на отказ	1000000 ч
Дополнительно Kingston SSDNow KC300 180Gb SKC300…	Найти похожие
Адаптер питания	—
Потребляемая мощность	2.9 Вт
Дополнительная информация	бесшумный, энергоэффективный, с малым тепловыделением
Размеры и вес Kingston SSDNow KC300 180Gb SKC300…	Найти похожие
Длина	100.1 мм
Ширина	69.75 мм
Высота	7 мм
Вес	86 г

Оптимизация работы SSD

- При использовании в компьютере на SSD диск лучше установить только ОС, а файлы в которые происходит частая перезапись разместить на HDD.
- Возможный вариант зависит от операционной системы :
- 1) В BIOS установить режим:
- AHCI (**Advanced Host Controller Interface** (**AHCI** — механизм, используемый для подключения накопителей по интерфейсу SATA, позволяющий пользоваться расширенными функциями, такими как встроенная очередность команд (NCQ).

Оптимизация работы SSD

- 2) В реестре отключить системный кэш Prefetch и Superfetch. (*Они не нужны при работе SSD накопителя. В большинстве случаев Prefetch отключается системой автоматически*).

- 3) Отключить автоматическую дефрагментацию диска. Она уменьшает ресурс SSD накопителя. (*Только для Windows 7, начиная с Windows 8 этой функции нет).*
- *4) Отключить или оптимизировать файл подкачки (Компьютер -> Свойства -> Дополнительные параметры системы …)*

Оптимизация работы SSD

- 5)Отключить индексирование файлов SSD(*в свойствах системного диска снимаем галочку с параметра «Разрешить индексировать содержимое файлов на этом диске»)*

- *6)* Перенести папки TEMP с твердотельного SSD на обычный HDD диск. (*В свойствах компьютера выбираем>> Дополнительные параметры системы >> Дополнительно >> Переменные среды. Ввести новый адрес для переменных сред TEMP и TMP, на жестком диске.)*

NVRAM (*Non Volatile Random Access Memory*);

- Тип энергонезависимой памяти (называется также полупостоянной памятью), в которой обеспечивается сохранение данных при отключении питания за счет маломощной встроенной батарейки.
- Выполнена по технологии CMOS (*Complementary Metal-Oxide-Semiconductor* - *комплементарный металлооксидный полупроводник или КМОП*)

ВИРТУАЛЬНАЯ ПАМЯТЬ

Типы адресов используемые процессами

- Физические адреса соответствуют номерам ячеек оперативной памяти, где в действительности будет расположен процесс.
- Процессор, при обращении к памяти использует только физические адреса

Транслятор присваивает командам и данным виртуальные (относительные) адреса, начиная с Некоторого постоянного адреса одинакового для всех процессов (например нулевого)

Работа компилятора

Адреса команд

```
0x00400000 main: addi $sp, $sp, -4
0x00400004       sw   $ra, 0($sp)
0x00400008       addi $a0, $0, 2
0x0040000C       sw   $a0, f
0x00400010       addi $a1, $0, 3
0x00400014       sw   $a1, g
0x00400018       jal  sum
0x0040001C       sw   $v0, y
0x00400020       lw   $ra, 0($sp)
0x00400024       addi $sp, $sp, 4
0x00400028       jr   $ra
0x0040002C sum:  add  $v0, $a0, $a1
0x00400030       jr   $ra
```

Адреса данных

Таблица символов

Символ	Адрес
f	0x10000000
g	0x10000004
y	0x10000008
main	0x00400000
sum	0x0040002C

- **Компилятор заменяет имя переменной на адрес ячейки памяти, где она может быть расположена**
- **Компилятор разделяет относительные адреса команд программы и адреса данных**

Работа компилятора

Executable file header	Text Size	Data Size
	0x34 (52 bytes)	0xC (12 bytes)
Text segment	**Address**	**Instruction**
	0x00400000	0x23BDFFFC
	0x00400004	0xAFBF0000
	0x00400008	0x20040002
	0x0040000C	0xAF848000
	0x00400010	0x20050003
	0x00400014	0xAF858004
	0x00400018	0x0C10000B
	0x0040001C	0xAF828008
	0x00400020	0x8FBF0000
	0x00400024	0x23BD0004
	0x00400028	0x03E00008
	0x0040002C	0x00851020
	0x00400030	0x03E00008
Data segment	**Address**	**Data**
	0x10000000	f
	0x10000004	g
	0x10000008	y

```
addi $sp, $sp, –4
sw   $ra, 0($sp)
addi $a0, $0, 2
sw   $a0, 0x8000($gp)
addi $a1, $0, 3
sw   $a1, 0x8004($gp)
jal  0x0040002C
sw   $v0, 0x8008($gp)
lw   $ra, 0($sp)
addi $sp, $sp, –4
jr   $ra
add  $v0, $a0, $a1
jr   $ra
```

Исполняемый файл

Типы адресов

Основная задача - как вычислить физический адрес

Виртуальное адресное пространство процесса

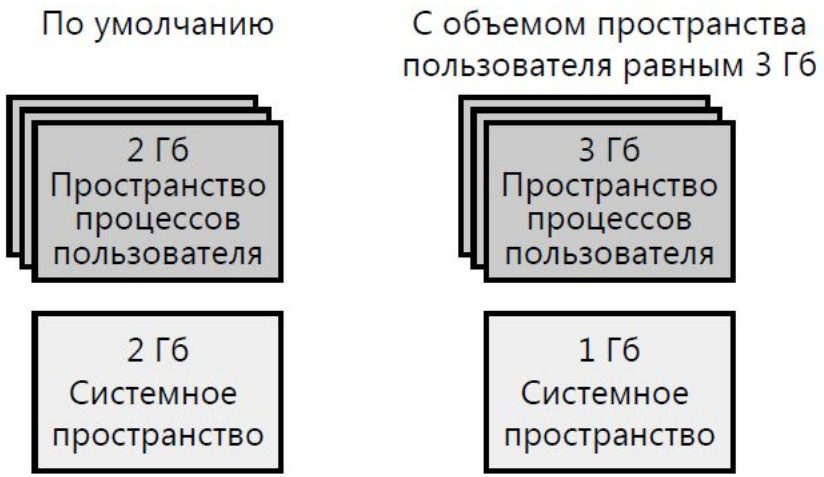
- *Виртуальное адресное пространство* **(ВАП)** *процесса* - совокупность виртуальных (относительных) адресов, получаемых после трансляции программы (команды и данные) **ПЛЮС** диапазон виртуальных адресов, дополнительно выделяемых ОС для создания и размещения образа процесса в памяти (стек + блок управления процессом + куча + общая используемая память).

Виртуальное адресное пространство процесса

- Компоновщик ,при компоновке может пересчитать адреса с учетом подключаемых модулей.
- У разных процессов могут быть виртуальные адресные пространства с совпадающими виртуальными адресами

Адресное пространство процесса – 32 разряда

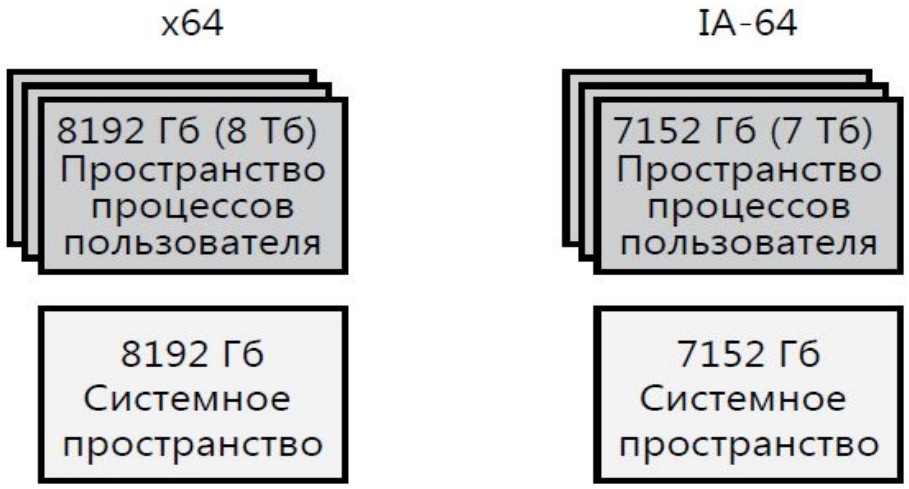
- Максимальный размер выделяемой виртуальной памяти , который может быть выделен каждому процессу, зависит от:
  - Разрядности процессора
  - Разрядности ОС
- Для 32 битных процессоров и ОС максимальный объем виртуальной памяти 4 Гб:
  - 2 (3)Гб для **каждого процесса** пользователя (**0x00010000 - 07FFFFFF**)
    2(1)Гб для ОС (



Типичные схемы адресных пространств для 32-разрядной версии Windows

Адресное пространство процесса – 64 разряда

- Теоретический размер адресного пространства процесса - 17 миллиардов гигабайт
- Реально с учетом ограничений ОС и железа 16 терабайт
  - 8Тб для **каждого процесса** пользователя
  - 8Тб для ОС



Схемы адресных пространств для 64-разрядной версии Windows

Резервирование адресного пространства процесса

- Происходит в три этапа
  - ОС выделяет (резервирует) для процесса диапазон виртуальных адресов (виртуальное адресное пространство)
  - ОС выделяет для виртуального пространства процесса физическую память.
  - Во время выполнения процесса виртуальные адреса преобразуются в физические.

Пример выделения регионов виртуальной памяти

- ОС выделяет автоматически регионы для кода, данных, стека, кучи в момент загрузки процесса.
- Дополнительные регионы процесс может выделить сам во время работы с помощью системных функций.
- **Регионы могут состоять из более мелких частей – сегментов и страниц**

Выделение физической памяти процессу

1. Виртуальное адресное пространство (регионы) условно поделено на блоки фиксированного размера, ***называемые виртуальными страницами.***
2. Физическое адресное пространство условно поделено на точно такие по размеру блоки называемые ***физическими страницами.***
3. ***Виртуальные и физические страницы имеют одинаковые размеры.*** Например 1 кБ.
4. Для начала выполнения процесса часть виртуальных страниц (с виртуальными адресами) загружается в соответствующие свободные физические страницы в Оперативной Памяти, а часть остается на внешнем диске в файле подкачки.
5. Соответствие между виртуальными и физическими страницами страницами описывается с помощью **таблицы страниц (page table)**.
6. При выполнении процесса виртуальная страница может быть выгружена из физической на внешний диск, а вместо неё может быть загружена другая виртуальная страница.

Проекция исполняемого файла на ВАП

Регион команд и данных | Регион стека | Регион кучи | Регион для библиотек

ВАП

Виртуальные адреса для **Команд и Данных**

Виртуальные адреса для **Стека**

Виртуальные адреса для **Кучи**

Виртуальные адреса для **DLL1**

DLL2

DLL(N)

Оперативная память

Внешняя память

Файл подкачки на диске HDD

Файл подкачки на диске SSD

Виртуальная память системы

- **Виртуальная память системы** = Размер ОП + Размер файла подкачки

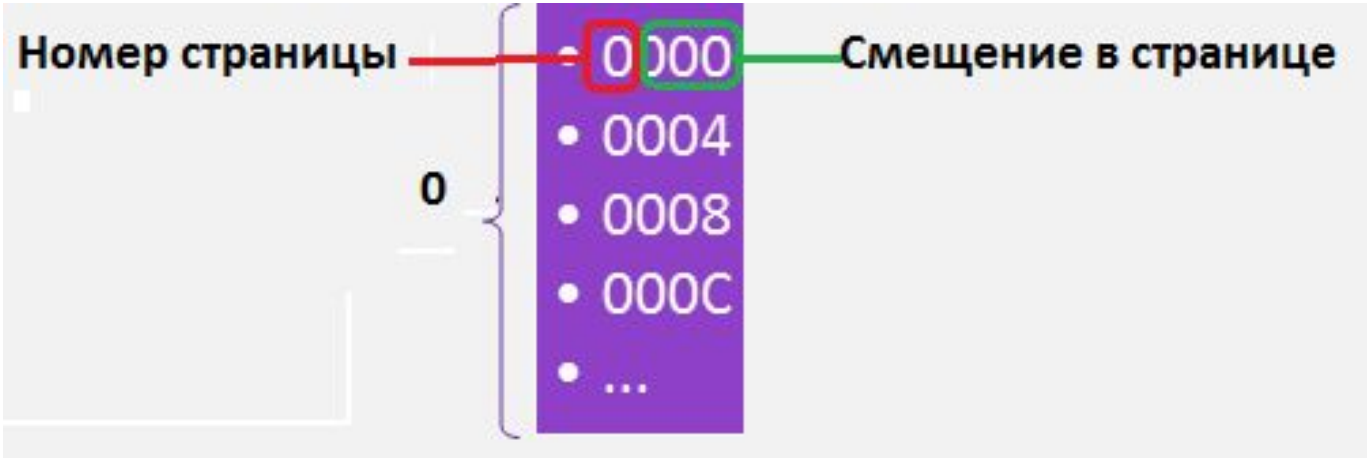
- **Файл подкачки** – специальная «популярная» область на жестком диске, время доступа к которой для ОС меньше чем время доступа к другим областям диска.
  - ОС временно размещает там данные на случай повторного использования.
  - Размер может поддерживаться автоматически либо задаваться вручную.

- Общий объём виртуальной памяти, одновременно занимаемый всеми процессами, не может превысить сумму основной памяти и файла подкачки (Pagefile).

Преобразование виртуального адреса в физический

Преобразование виртуального адреса в физический

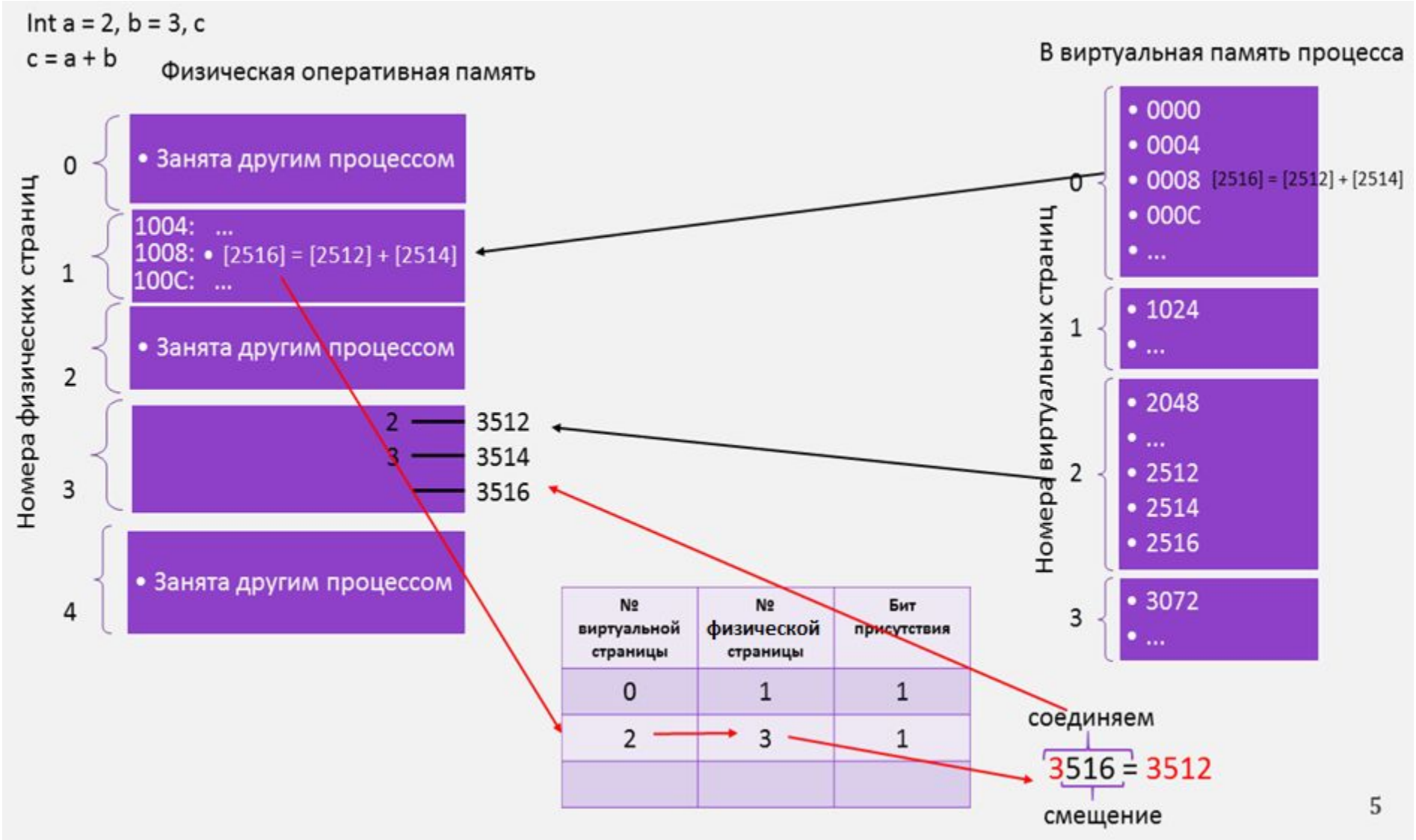
- Физический и виртуальный адрес состоит из номера страницы и смещения внутри страницы



Преобразование виртуального адреса в физический

- Виртуальная страница процесса (адреса команд и данных виртуальные) загружается в физическую страницу.
- В таблицу страниц заносится соответствие номеров виртуальной и физической страниц.
- При обращении по виртуальному адресу к данным этот адрес преобразуется в физический следующим образом:
  - По номеру виртуальной страницы из таблицы страниц считывается номер соответствующей физической страницы.
  - К номеру физической страницы добавляется смещение внутри страницы, которое берется из второй части виртуального адреса
  - По сформированному физическому адресу считывается значение переменной

Преобразование виртуального адреса в физический

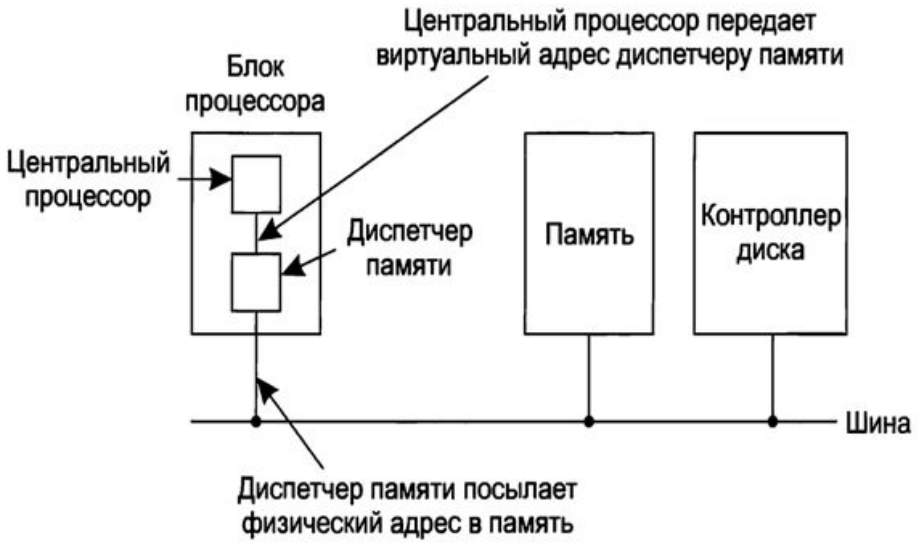


Сокращение времени преобразования адреса

1. Преобразование происходит специальным аппаратным модулем управления памятью процессора (MMU)
2. Многоуровневые таблицы страниц
3. Таблица страниц помещается в ассоциативный кэш

TLB – буффер ассоциативной трансляции (*Translation lookaside buffer)*

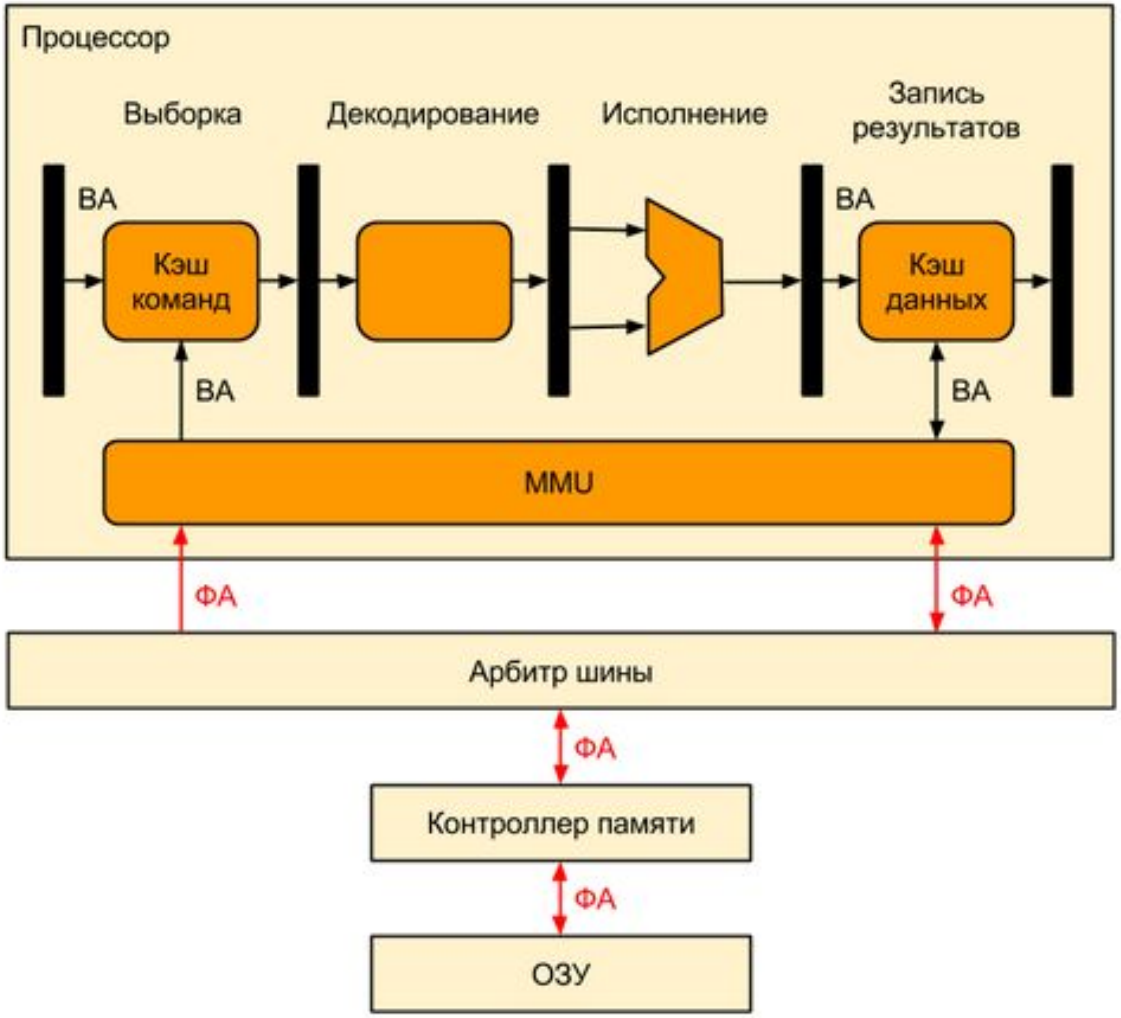
MMU - memory management unit



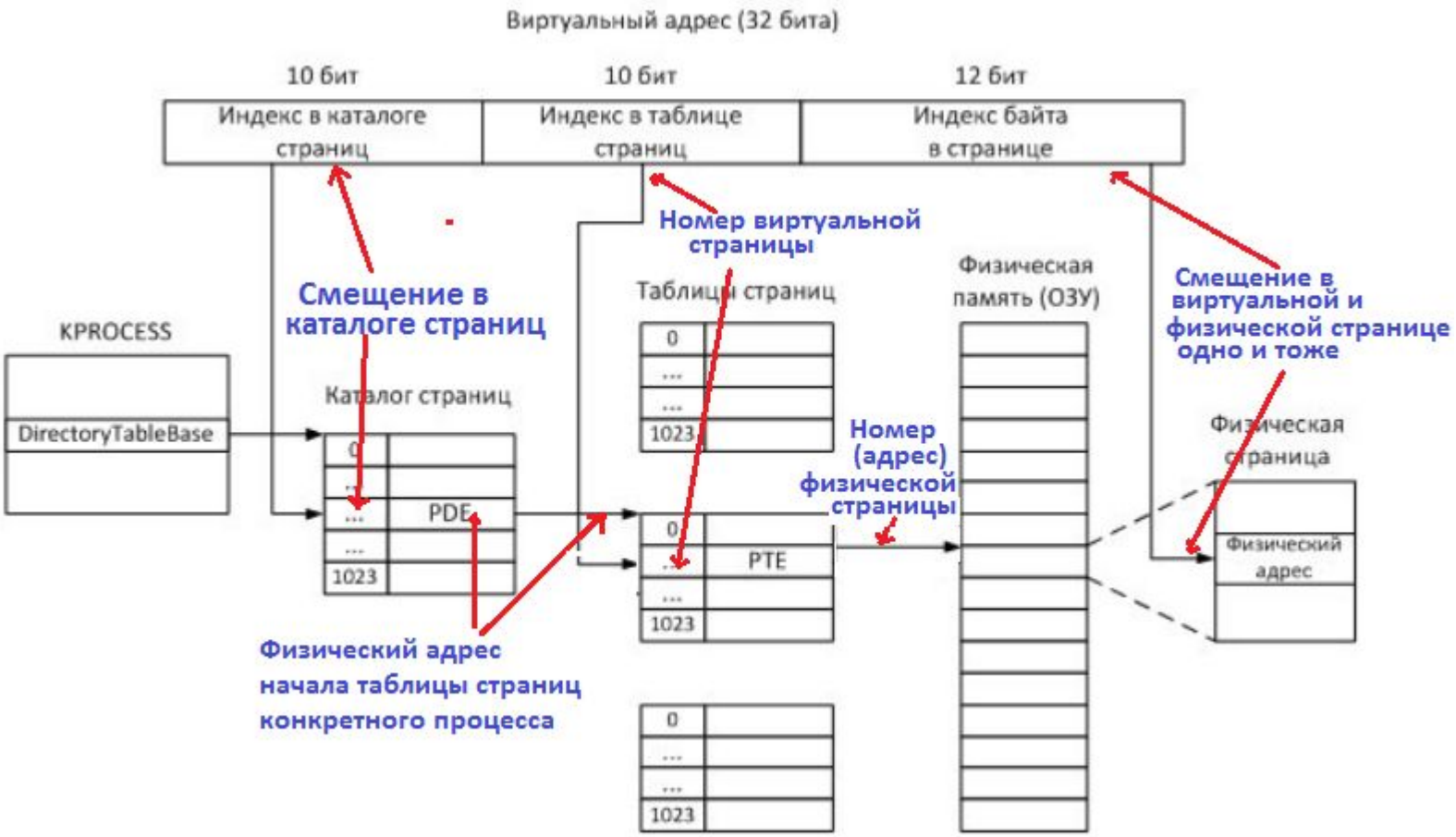
Аппаратный диспетчер памяти может располагаться в составе процессора или в составе северного моста чипсета

Во время выполнения программы при каждом обращении к памяти *виртуальные адреса не выставляются напрямую на шину памяти, а предварительно поступают в диспетчер памяти (MMU, Memory Management Unit), который преобразует виртуальные адреса в адреса физической памяти.*

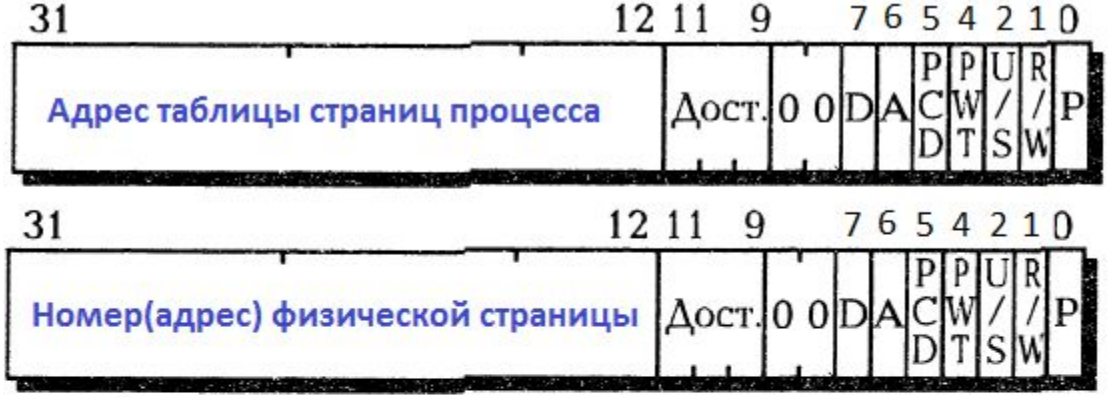
MMU


Процессор
Выборка
Декодирование
Исполнение
Запись результатов
ВА
Кэш команд
ВА
Кэш данных
ВА
ВА
MMU
ФА
ФА
Арбитр шины
ФА
Контроллер памяти
ФА
ОЗУ

Многоуровневые таблицы страниц



Защищенный страничный режим



P - бит присутствия

R/W - запись/чтение

U/S - user/system - системная или пользовательская страница

- Если 0, то доступ к странице разрешен только со стороны ОС в режиме ядра

- Если единица , то доступ разрешен для процессов в режиме пользователя

A - было обращение по чтению

D - было обращение по записи

Процесс обращается только к тем страницам, которые присутствуют в его таблице страниц

Адрес в каталоге таблиц содержит поле доступа, которое определяет приоритет доступа к
странице

Виртуальный адрес содержит приоритет доступа процесса

ОС сравнивает уровень привилегий процесса обращающегося к данной странице с её