

Лекция 11

Оперативная память

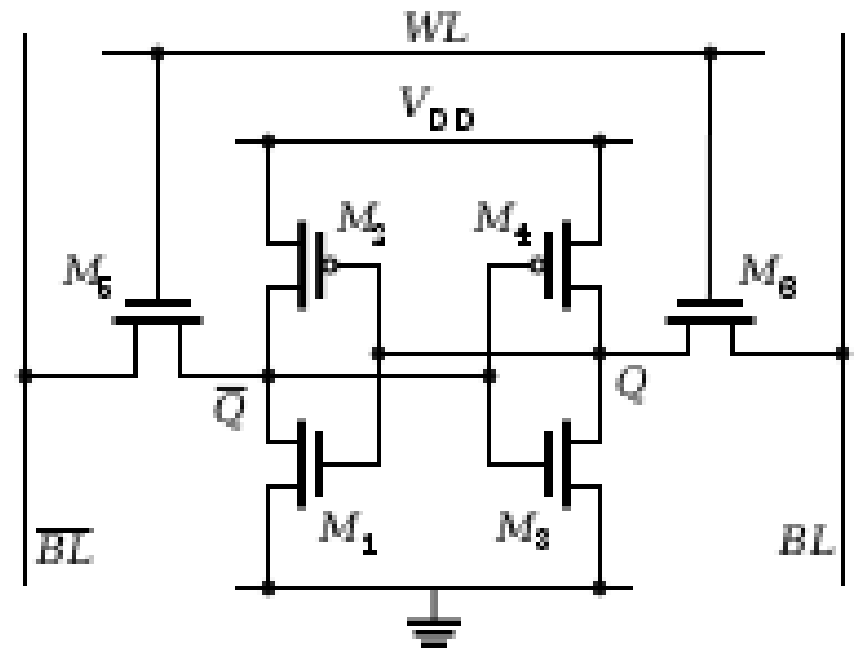
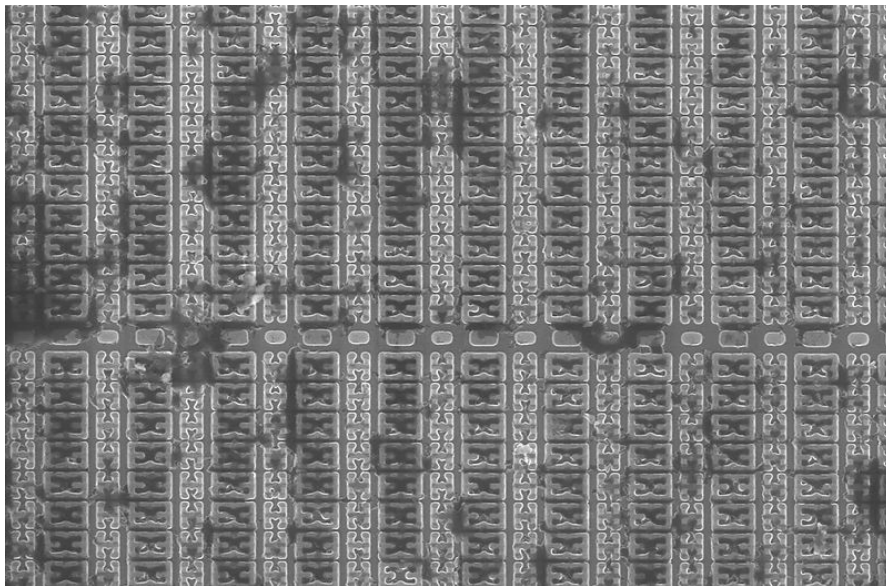
Типы ОЗУ

- Статическая память произвольного доступа (SRAM)
- Динамическая память произвольного доступа (DRAM)

Временные характеристики ОЗУ

- Время чтения (латентность - latency) — время от начала операции чтения до появления значения на выходе ОЗУ (входе процессора)
- Время цикла — время от начала операции чтения до готовности к следующей операции
-
- Пропускная способность (throughput) — количество данных, которые могут быть считаны из ОЗУ за единицу времени

Внешний вид

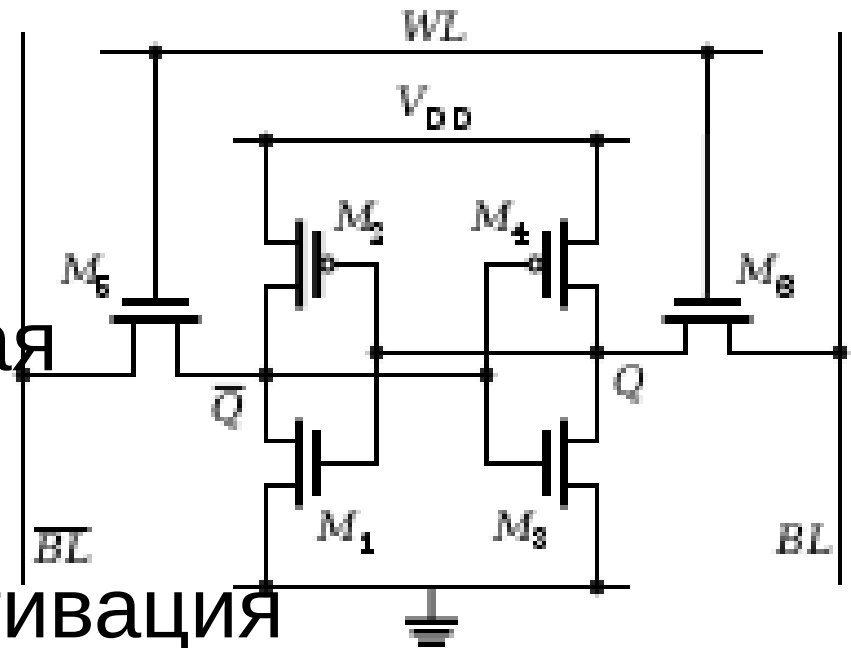


Принцип действия

- Транзисторы M1, M2, M3, M4 – два инвертора, “навстречу” друг другу: два стабильных состояния (0 и 1), пока подключено питание на Vdd
- В пассивном (stand-by) состоянии ток практически не протекает
- Транзисторы M5 и M6 подключают ячейку к линии данных BL
- Одновременно считываются/устанавливаются BL и #BL (повышается надежность)

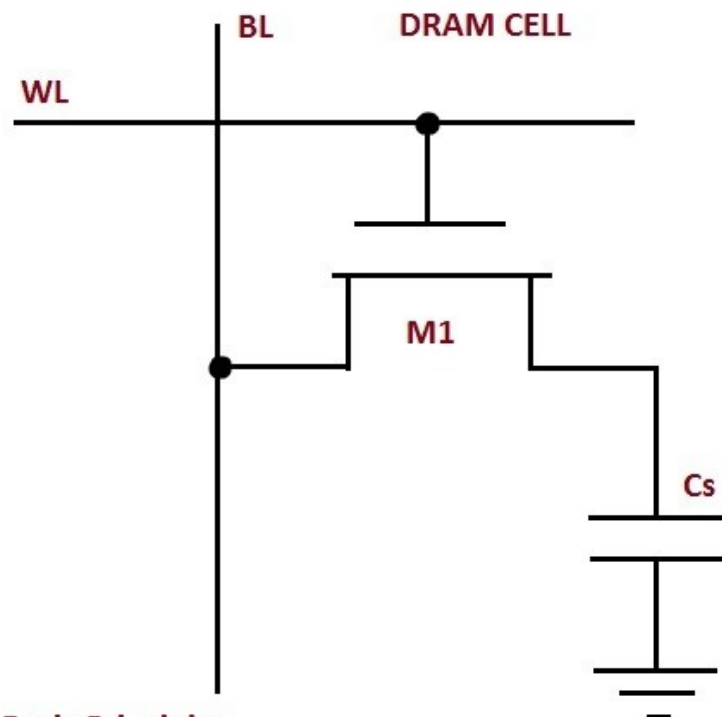
Статическая ППД

- 6 транзисторов (существуют схемы 4-10 тр.)
- Время чтения: ~1 такт
- Время цикла: ~1 такт
- Время записи: ~2 такта
- Тактовая частота - любая
-
- Vdd – питание, WL – активация
BL - данные

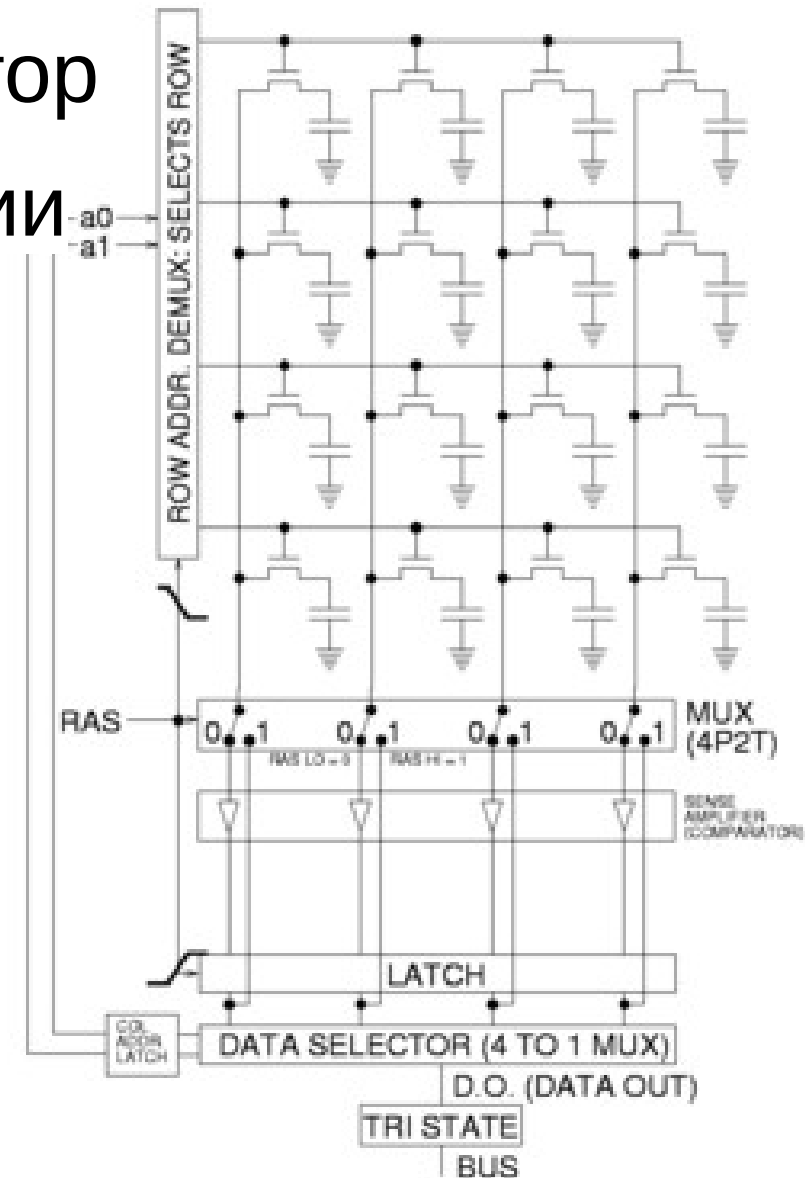


Динамическая ППД

- 1 транзистор + 1 конденсатор
- Необходимость регенерации



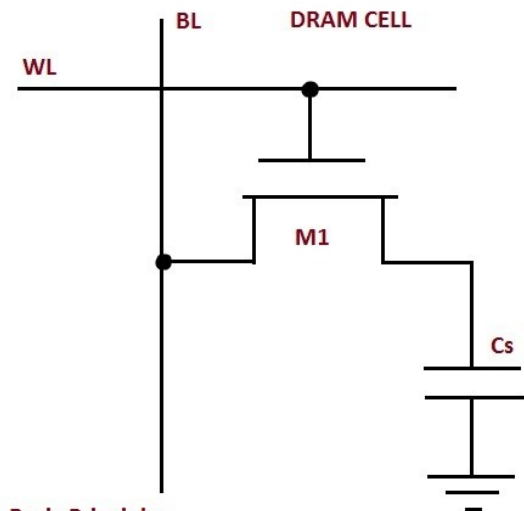
Basic Principle:
Capacitor stores Data
MOSFET acts as the Control Switch



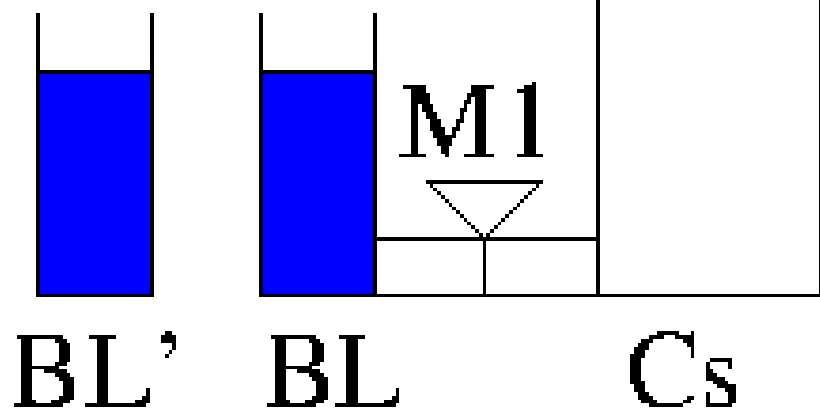
Чтение из DRAM (1)

- Precharge – на линии BL и BL' подается напряжение 0.5V (половина рабочего)
- Линии обладают достаточной емкостью для накопления некоторого заряда
- Линия BL – подключена к ячейкам через управляющий транзистор
- Линия BL' – имеет точно такую же емкость, как и BL, но не подключена (референсная)

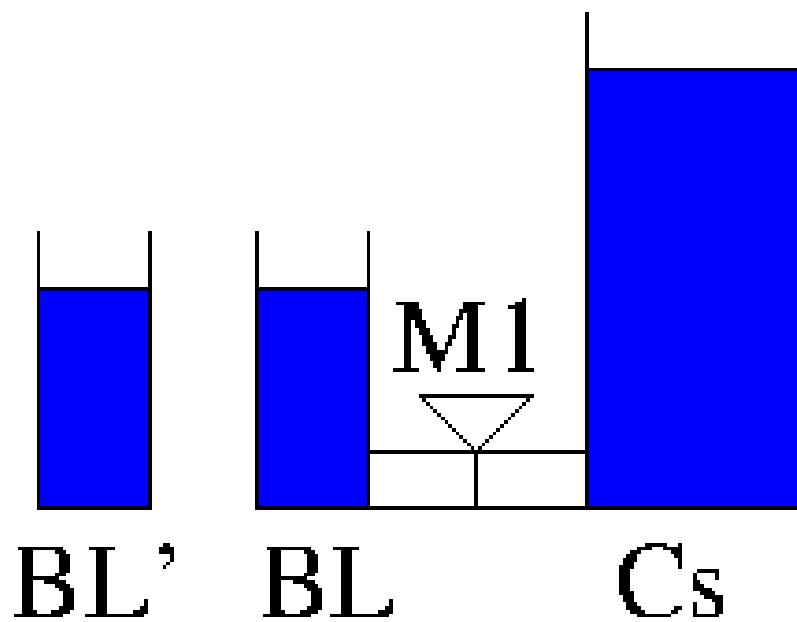
Precharge



Basic Principle:
Capacitor stores Data
MOSFET acts as the Control Switch



Ячейка содержит 0

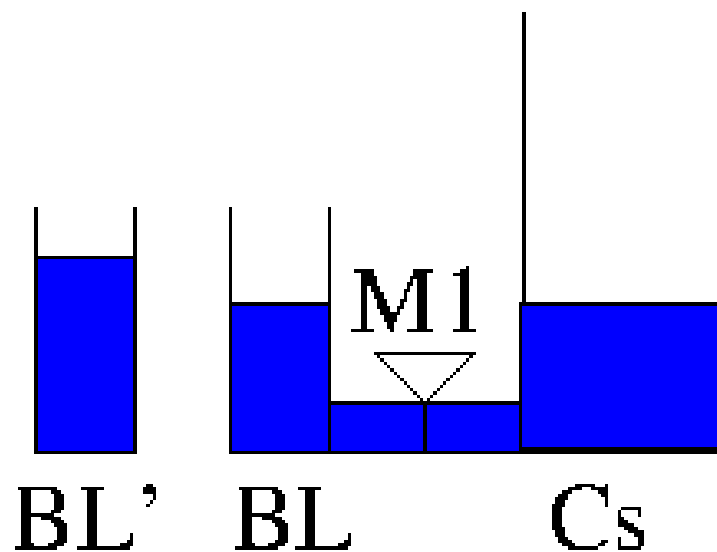


Ячейка содержит 1

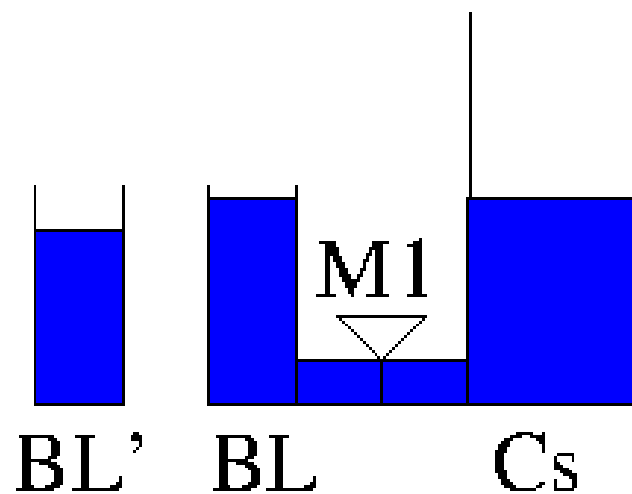
Чтение из DRAM (2)

- На нужную горизонтальную линию (ROW) подается напряжение логической 1, открываются BCE транзисторы в ряду (Row Select - RAS)
- Все ячейки в одном ряду соединяются с соответствующими линиями BL
- Напряжения в линиях BL чуть уменьшаются или чуть увеличиваются в зависимости от значений в ячейках

Reading



Ячейка содержит 0



Ячейка содержит 1

Чтение из DRAM (3)

- Напряжение на BL и BL' сравнивается и в каждом столбце фиксируется 0 или 1
- Строка фиксируется (LATCH)
- Заряд в подключенных ячейках обновляется (refresh)
- Из строки выбираются нужные биты (Column Select – CAS)
- Строка закрывается (транзисторы отключаются)
- Затем снова начинается Precharge и цикл чтения повторяется

Алгоритм чтения из DRAM

- ROW SELECT — на нужную строку подается напряжение V , транзисторы открываются
- Напряжение вертикальной линии BL слегка изменяется, разница BL и BL' усиливается дифференциальным усилителем — одновременно считывается и фиксируется (latch) целая строка
- COL SELECT — выборка нужных битов из выбранной строки
- После выборки нужных бит строка закрывается
- PRECHARGE — на все битовые (вертикальные) линии B+ и B- подается напряжение $0.5 V$ (от рабочего)

Обновление DRAM

- Распознаваемый «единичный» заряд в конденсаторе может держаться до 1-10с
- Обновление — чтение и запись обратно каждой ячейки
- Современные поколения DRAM обновляют каждую ячейку каждые 64мс
- Во время обновления память недоступна процессору
- Накладные расходы на обновление — примерно 1% времени

Модификации DRAM

- SDRAM (synchronous DRAM) — использует тактовые импульсы для синхронизации. Все временные характеристики задаются в тактах.
- DDR SDRAM (Double Data Rate) — передача/прием данных дважды за такт
- DDR, DDR2, DDR3

Memory Timings

- CAS Latency (Tcl) — время между выдачей адреса столбца и получением данных
- RAS to CAS Delay (Trcd) — время между выдачей адрес строки и адреса столбца
- Row Precharge Time (Trp) — время между выдачей команды precharge и открытием следующей строки
- Row Active Time (Tras) — время между выборкой строки и выдачей команды precharge
-
- Row Cycle Time: $Trc = Tras + Trp$

Производительность памяти

- Формула: $T_{cl}-T_{rcd}-T_{rp}-T_{ras}$
- PC-3200 (DDR-400) - 2000
 - 3-4-4-8 (15ns-20ns-20ns-40ns)
- PC2-6400 (DDR2-800) - 2003
 - 5-5-5-16 (12.5-12.5-12.5-40)
- PC3-12800 (DDR3-1600) - 2007
 - 9-9-9-27 (11.25-11.25-11.25-33.75)
- PC4-25600 (DDR4-3200) – 2014
 - 22-22-22-52 (13.75-13.75-13.75-32.5)

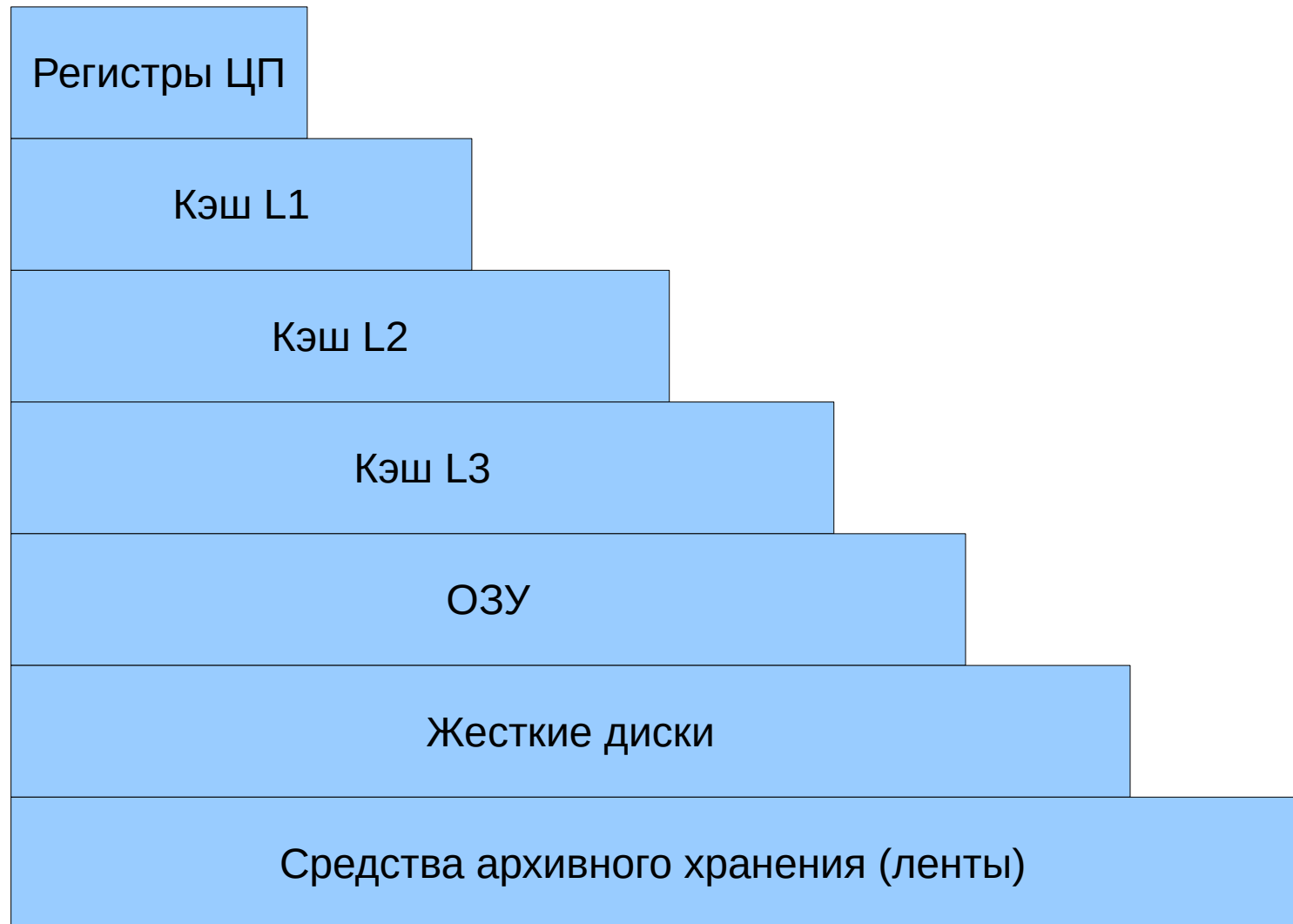
Сравнение скорости ЦП и ОЗУ

- Предположим, тактовая частота процессора — 3.2ГГц
- Тактовая частота ОЗУ — 800 МГц ($\frac{1}{4}$ от частоты процессора)
- Время чтения: 72 тактов процессора
- Время цикла: 108 тактов
- С учетом накладных расходов время чтения из ОЗУ: 100 — 200 тактов процессора

Кэширование

- Кэш (cache) (заначка, тайник) — средство для сглаживания разности скоростей устройств. Некоторая часть данных с медленного устройства помещается в кэш, работающий со скоростью быстрого устройства.

Иерархия памяти



Кеш-память

- Кеш-память имеет смысл, так как программы на практике демонстрируют свойство **локальности**
- Кеш-память не дает улучшения в худшем случае (при случайных обращениях в память)

Свойство локальности

- Временная локальность – если программа обращается к некоторой ячейке памяти впервые, велика вероятность того, что скоро обращение к этой ячейке памяти повторится
 - Циклы в коде программы
 - Переменные в памяти
- Пространственная локальность – если программа обращается к некоторой ячейке памяти, велика вероятность того, что скоро программа обратится к соседним ячейкам
 - Код программы
 - Массивы/структуры в памяти

Работа кэш-памяти

- Попадание (hit) — значение берется из кэша, а не из ОЗУ
- Промах (miss) — требуемой ячейки в кэше нет
 - Обязательный промах (ячейка не загружена)
 - Промах из-за размера
 - Промах из-за конфликта (ячейка была в кэше, но оказалась выгруженной)

Промахи в кеше

- Обязательный промах (compulsory miss) – ячейка не была загружена в кеш – первое обращение к ней в программе
- Промех из-за емкости (capacity miss) – размер кеша слишком мал для одновременного хранения используемых данных
- Промех из-за конфликта (conflict miss) – нужные данные были в кеше, но оказались выгружены из-за ограниченной ассоциативности

Кэш-память ЦП

- L1 — наименьший размер (до 64 KiB), делится на кэш инструкций и данных (L1I, L1D), отдельный для каждого ядра
- L2 — средний размер (1 MiB), общая для данных и инструкций, отдельный
- L3 — наибольший размер (до 8 MiB), общая для данных и инструкций, общая для всех ядер
- TLB (translation lookahead buffer) — для организации виртуальной памяти

Среднее время доступа

- Оценка времени, которое процессор тратит, ожидая данные из памяти
- Для одноуровневого кеша
 $AMAT = t_c + M_c * t_M$
- t_c – время обращения к кешу
- M_c – вероятность промаха в кеше
- t_M – время обращения к памяти
- Если $t_c = 1$, $M_c = 0.10$, $t_M = 100$, то $AMAT = 11$

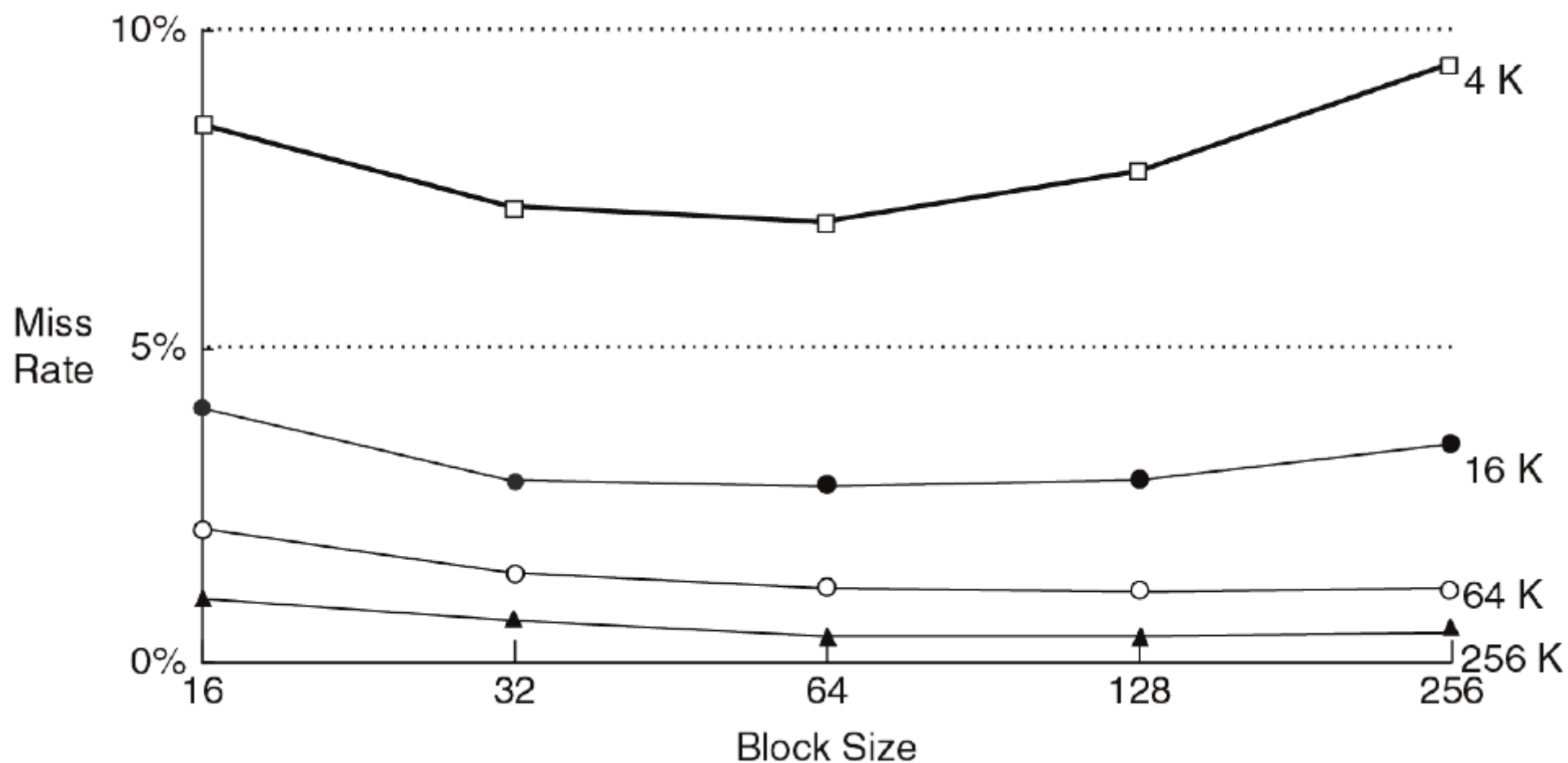
Характеристики кэш-памяти

- Размер
- Время доступа
- Размер ряда (блока) кэша
- Ассоциативность
- Политика записи в основную память
- Политика замещения
- Политика обеспечения когерентности

Размер ряда (блока)

- Cache Line (cache block) – единица хранения данных в кеше
- При чтении одного байта из ОЗУ загружается ряд (блок) ячеек
- Типичный размер: 64 байта (x86, ARM)

Зависимость от размера блока



Ассоциативность

- Предположим, что кэш позволяет разместить M блоков
- Рассмотрим загрузку блока B из памяти в кэш
- Если блок B может быть размещен в любой ячейке кэша — **полностью** ассоциативный
- Если блок B может быть размещен в одной ячейке — **прямое отображение**
- Если блок B может быть размещен в N ячейках — **кэш N -ассоциативный**

Кеш прямого отображения

- Каждый из адресов в ОЗУ отображается в единственную ячейку кеша
- Пусть размер блока – 16 байт, размер кеша – 256 байт, размер ОЗУ – 4 KiB: в кеше 16 блоков, в ОЗУ – 256 блоков
- Адрес в ОЗУ состоит из трех частей (адреса – hex):
XYZ
X – игнорируется при отображении, Y – номер блока, Z – смещение в блоке
9A7 – адрес отобразится в блок кеша A
2A3 – адрес отобразится в блок кеша A - конфликт

N-ассоциативный кеш

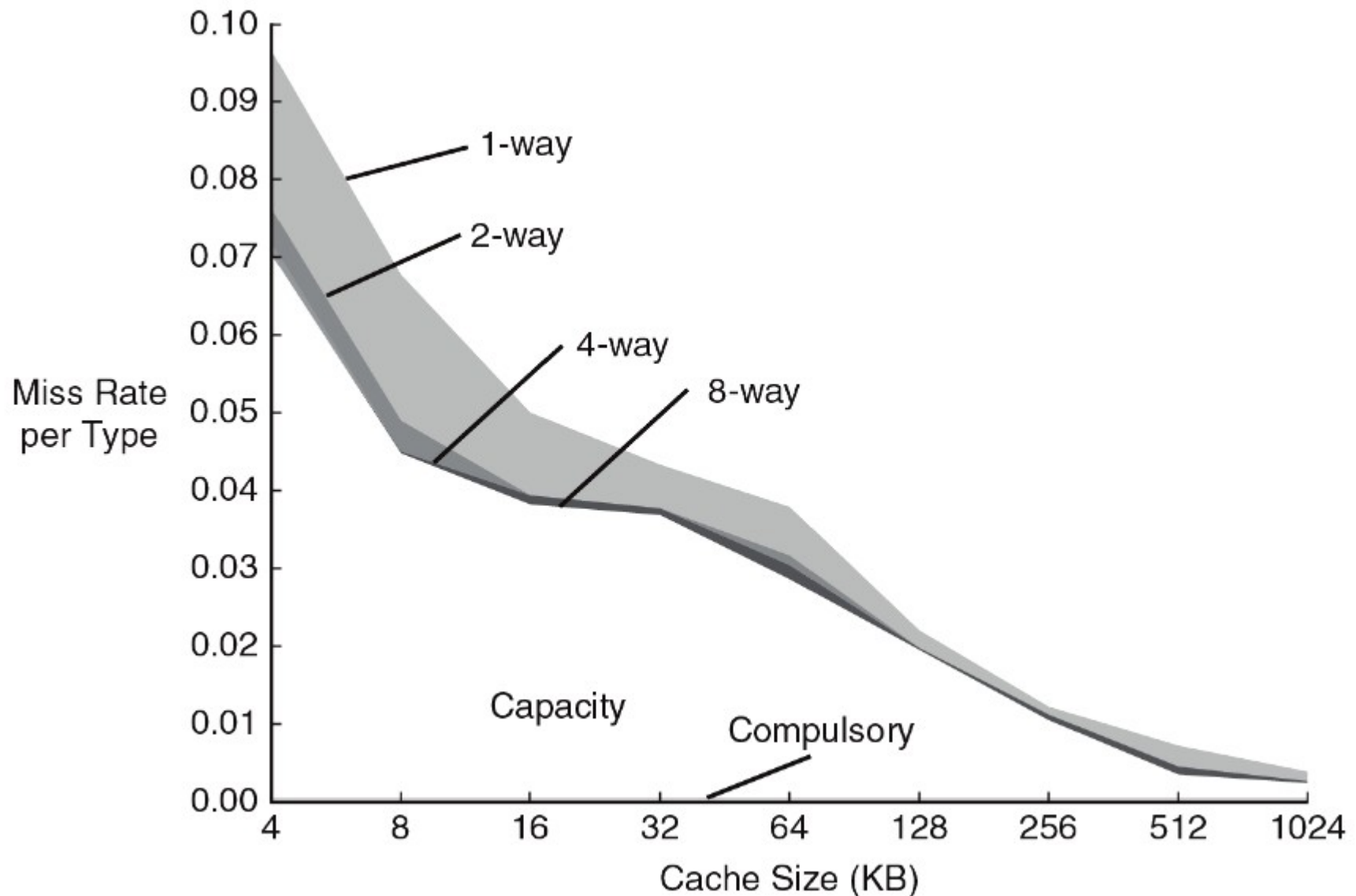
- N-секционный наборно-ассоциативный кеш (N-way set associative cache)
- Кеш делится на наборы, каждый набор содержит N блоков
- Каждый адрес памяти отображается в строго определенный набор, но в наборе может попасть в любой блок

N-ассоциативный кеш

- Пусть размер блока – 16 байт, размер кеша – 256 байт, размер ОЗУ – 4 KiB: в кеше 16 блоков, в ОЗУ – 256 блоков;
- Кеш – 4-наборный, каждый набор – 4 блока
- Рассмотрим адрес XYZ, Y определяет местоположение в кеше: $Y \& 3$ – номер набора; 6 битов адреса $(XY \gg 2)$ являются тегом в наборе
- Совпадение части X в адресах не означает конфликта
- Меньше промахов типа conflict miss

Оценка % промахов на SPEC2000

-



Политика замещения

- При чтении блока в кеш, когда все позиции на которых может разместиться блок в кеше заняты необходимо выкинуть из кеша один из блоков
- Выбор блока для удаления из кеша определяется политикой замещения (replacement policy)
- Возможные политики замещения:
 - LRU (least recently used) – не используемый наиболее давно
 - LFU (least frequently used) – используемый наименее часто

Политика записи в основную память

- При выполнении операции записи в ячейку памяти, которая находится в кеше, возможны следующие варианты:
- Модифицировать ячейку в кеше и сразу же выполнить операцию записи в основную память (сквозная запись – write-through)
- Модифицировать ячейку в кеше и отметить ее как “грязную” и отложить запись на более удобный момент (отложенная запись – write-back)

Когерентность кеша

- В многоядерных процессах или многопроцессорных системах каждое ядро обладает своими L1 и L2 кешами
- Один и тот же блок основной памяти может находиться в нескольких кешах
- Одно из ядер выполняет модификацию этого блока – что делать?
- Стратегии поддержания когерентности:
 - Snooping – кешы следят за адресами записи в память
 - Write invalidate – модифицированные блоки выбрасываются из “соседних” кешей
 - Write update – модифицированные блоки обновляются

Политика включения

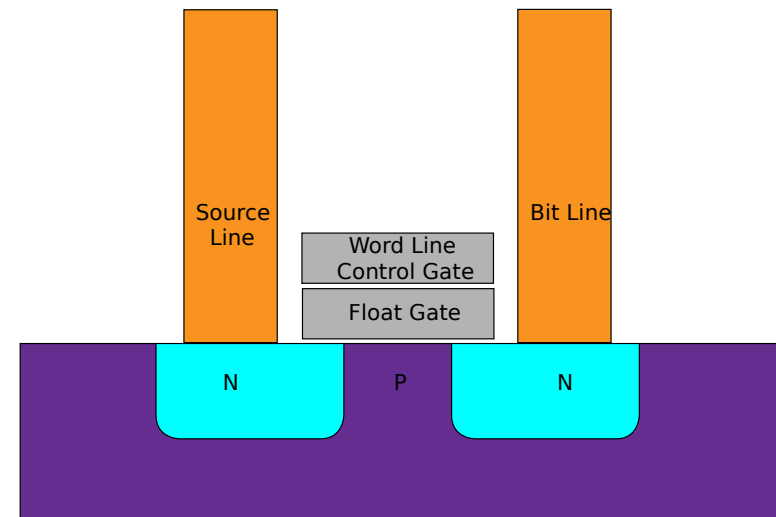
- Для многоуровневых кешей:
- Inclusive policy – если блок присутствует в кеше L_i (L_1 , L_2), он присутствует и в кешах более низкой иерархии
 - Если блок удаляется из кеша более низкой иерархии (associativity miss), он удаляется из кешей иерархии выше
- Exclusive policy – блок присутствует только в одном кеше
 - Блок переносится “вверх” или “вниз”
- NINE (non-inclusive non exclusive)

Flash-memory

- Используется в SSD (solid state disks), носителях данных

Floating Gate MOSFET

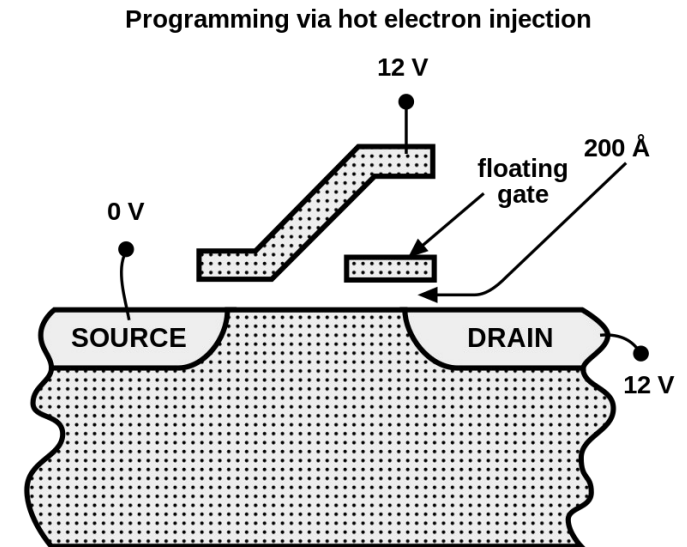
- SourceLine, BitLine, ControlGate – “обычные” контакты транзистора
- FloatGate - “висит в воздухе”, т. е. никуда не подключен, изолирован
- Состояние ячейки памяти определяется наличием заряда на FloatGate
- Заряд в изолированный проводник попадает с помощью пробоя изолятора и вытекает с помощью туннельного эффекта
- Начальное состояние – логическая 1 (отсюда NOR, NAND)



Работа ячейки

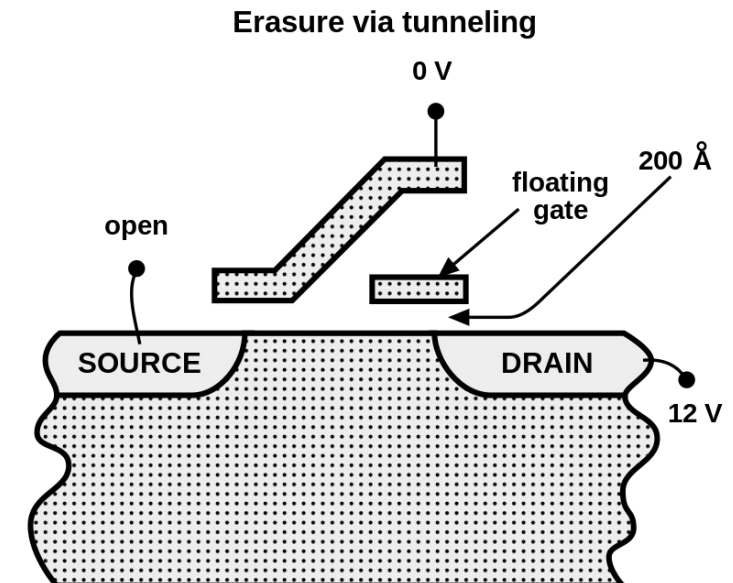
Запись 0 (программирование):

- На затвор подается высокое напряжение
- Транзистор открывается
- Электроны перемещаются из истока в сток (большой ток)
- Часть электронов пробивает изоляцию и попадает на floating gate



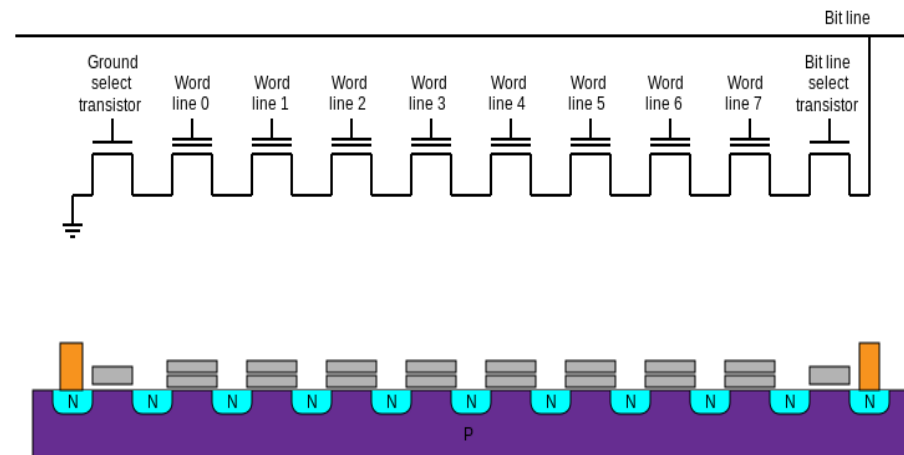
Запись 1 (очистка):

- Между стоком и затвором подается напряжение обратной полярности
- Транзистор остается закрытым
- За счет туннельного эффекта электроны перемещаются из floating gate на сток



NAND flash

- FG MOSFET ячейки
включаются
последовательно (как ряд в
DRAM)
- При чтении в нужный
столбец WL устанавливается
в напряжение чтения,
остальные столбцы – в
напряжение открытия
- Ток будет течь в зависимости
от значения в нужной ячейке



Особенности flash-памяти

- Чтение и программирование (запись 0) – произвольная ячейка, но очистка (запись 1) – только целым блоком
- “Выгорание” - конечное число Programming/Erase (P/E) циклов (~100000)
- Постоянное чтение из одной ячейки может влиять на соседние ячейки в NAND

Влияние на ОС

- Негативные стороны NAND flash сглаживаются (до определенной меры) контроллерами SSD и Flash
 - Wear leveling – попытка сделать так, чтобы блоки “выгорали” примерно с равной скоростью
 - Read counting – копирование блока в новый после выполнения некоторого количества чтений
- Для SSD/Flash дисков желательна специальная поддержка со стороны ОС (например, файловые системы)