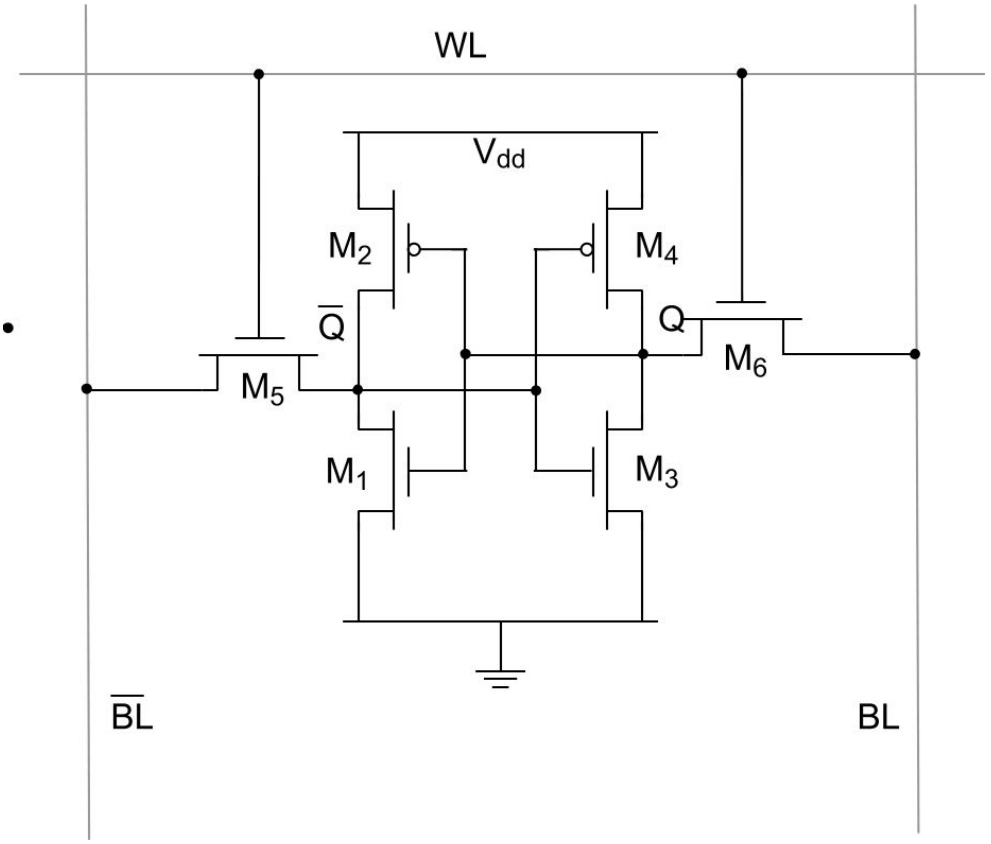
13. Статическая и динамическая память с произвольным доступом. Особенности организации, основные показатели быстродействия.

Статическая память с произвольным доступом (SRAM, static random access memory) — полупроводниковая оперативная память, в которой каждый двоичный или троичный разряд хранится в схеме с положительной обратной связью, позволяющей поддерживать состояние без регенерации, необходимой в динамической памяти (DRAM). Тем не менее сохранять данные без перезаписи SRAM может, только пока есть питание, то есть SRAM остается энергозависимым типом памяти.



Шеститранзисторная ячейка статической двоичной памяти (бит) SRAM

Преимущества:

- Быстрый доступ. SRAM — это действительно память произвольного доступа, доступ к любой ячейке памяти в любой момент занимает одно и то же время.
- Простая схемотехника — SRAM не требуются сложные контроллеры.
- Возможны очень низкие частоты синхронизации, вплоть до полной остановки синхроимпульсов.

Недостатки:

- Невысокая плотность записи (шесть-восемь элементов на бит[3], вместо двух у DRAM).
- Вследствие чего — дороговизна килобайта памяти.
- Особенность: непредсказуемое (произвольное) содержимое памяти после включения питания.

Тем не менее, высокое энергопотребление не является принципиальной особенностью SRAM, а обусловлено высокими скоростями обмена с данным видом внутренней памяти процессора.

В устройствах с большим объёмом ОЗУ рабочая память выполняется как DRAM. SRAM же применяется для регистров и кэш-памяти.

DRAM (англ. dynamic random access memory — динамическая память с произвольным доступом) — тип компьютерной памяти, отличающийся использованием полупроводниковых материалов, энергозависимостью и возможностью доступа к данным, хранящимся в произвольных ячейках памяти. Модули памяти с памятью такого типа широко используются в современных компьютерах в качестве оперативных запоминающих устройств (ОЗУ), также используются в качестве устройств постоянного хранения информации в системах, требовательных к задержкам.

Физически DRAM состоит из ячеек, созданных в полупроводниковом материале в виде емкости. Заряженная или разряженная емкость хранит бит данных. Каждая ячейка такой памяти имеет свойство разряжаться (из-за токов утечки и пр.), поэтому их постоянно надо подзаряжать — отсюда название «динамическая» (динамически подзаряжать). Совокупность ячеек образует условный «прямоугольник», состоящий из определённого количества строк и столбцов. Один такой «прямоугольник» называется страницей, а совокупность страниц называется банком (можно просто массив). Весь набор ячеек условно делится на несколько областей.

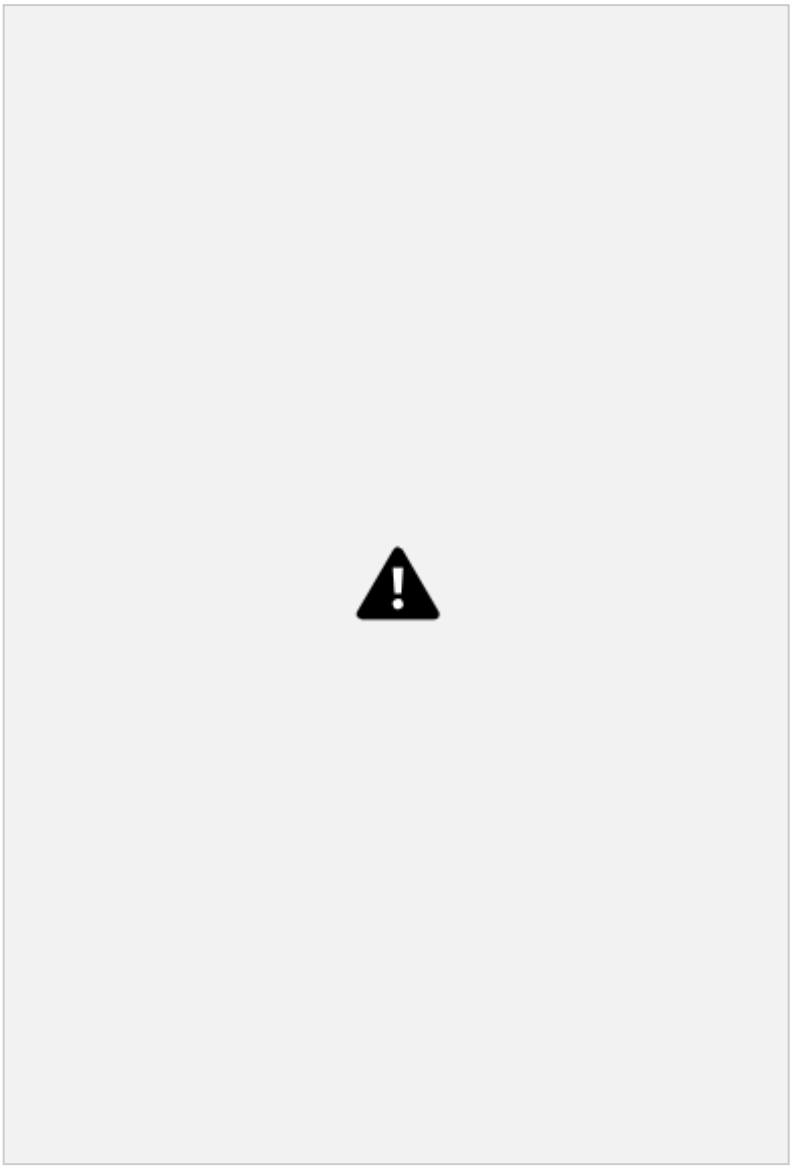
Как запоминающее устройство (ЗУ) DRAM представляет собой модуль памяти какого-либо конструктивного исполнения, состоящий из печатной платы, на которой расположены микросхемы памяти, и разъёма, необходимого для подключения модуля к материнской плате.

Принцип действия

На физическом уровне память DRAM представляет собой набор ячеек, способных хранить информацию. Ячейки состоят из конденсаторов и транзисторов, расположенных внутри полупроводниковых микросхем памяти[2]. Конденсаторы заряжают при записи в ячейку единичного бита и разряжают при записи в ячейку нулевого бита.

При прекращении подачи электроэнергии конденсаторы разряжаются, и память обнуляется (опустошается). Для поддержания необходимого напряжения на обкладках конденсаторов (для сохранения данных) конденсаторы необходимо периодически подзаряжать. Подзарядку выполняют путём подачи на конденсаторы напряжения через коммутирующие транзисторные ключи. Необходимость постоянной зарядки конденсаторов (динамическое поддержание заряда конденсаторов) является основополагающим принципом работы памяти типа DRAM.

Важным элементом памяти типа DRAM является чувствительный усилитель-компаратор (англ. sense amp), подключённый к каждому из столбцов «прямоугольника». При чтении данных из памяти усилитель-компаратор реагирует на слабый поток электронов, устремившихся через открытые транзисторы с обкладок конденсаторов, и считывает одну строку целиком. Чтение и запись выполняются построчно; обмен данными с отдельно взятой ячейкой невозможен.



Основными характеристиками DRAM являются рабочая частота и тайминги.

Перед обращением к ячейке памяти контроллер памяти передаёт модулю памяти номер банка, номер страницы банка, номер строки страницы и номер столбца страницы; на эти запросы тратится время. До и после выполнения чтения или записи довольно большой промежуток времени уходит на «открытие» и «закрытие» банка. На каждое действие требуется время, называемое таймингом.

CAS-латентность	**CL**	Задержка между отправкой в память адреса столбца и началом передачи данных. Время, требуемое на чтение первого бита из памяти, когда нужная строка уже открыта.
Row Address to Column Address Delay	**T_{RCD}**	Число тактов между открытием строки и доступом к столбцам в ней. Время, требуемое на чтение первого бита из памяти без активной строки — T_{RCD} + CL.

Row Precharge Time	$\mathbf{T_{RP}}$	Число тактов между командой на предварительный заряд банка (закрытие строки) и открытием следующей строки. Время, требуемое на чтение первого бита из памяти, когда активна другая строка — $T_{RP} + T_{RCD} + CL$.
Row Active Time	$\mathbf{T_{RAS}}$	Число тактов между командой на открытие банка и командой на предварительный заряд. Время на обновление строки. Накладывается на T_{RCD}. Обычно примерно равно сумме трёх предыдущих чисел.

14. Кэш центрального процессора. Проблема когерентности кэша. Ассоциативность, длина строки, инклюзивность и эксклюзивность кэша, иерархия уровней, время доступа.

Кэш микропроцессора — кэш (сверхоперативная память), используемый микропроцессором компьютера для уменьшения среднего времени доступа к компьютерной памяти. Является одним из верхних уровней иерархии памяти[1]. Кэш использует небольшую, очень быструю память (обычно типа SRAM), которая хранит копии часто используемых данных из основной памяти. Если большая часть запросов в память будет обрабатываться кэшем, средняя задержка обращения к памяти будет приближаться к задержкам работы кэша.

Когда процессору нужно обратиться в память для чтения или записи данных, он сначала проверяет, доступна ли их копия в кэше. В случае успеха проверки процессор производит операцию, используя кэш, что значительно быстрее использования более медленной основной памяти.

Данные между кэшем и памятью передаются блоками фиксированного размера, также называемые линиями кэша (англ. cache line) или блоками кэша.

Большинство современных микропроцессоров для компьютеров и серверов имеют как минимум три независимых кэша: кэш инструкций для ускорения загрузки машинного кода, кэш данных для ускорения чтения и записи данных и буфер ассоциативной трансляции (TLB) для ускорения трансляции виртуальных (логических) адресов в физические, как для инструкций, так и для данных. Кэш данных часто реализуется в виде многоуровневого кэша (L1, L2, L3).

Принцип работы

Каждая строка — группа ячеек памяти содержит данные, организованные в кэш-линии. Размер каждой кэш-линии может различаться в разных процессорах, но для большинства x86-процессоров он составляет 64 байта. Размер кэш-линии обычно больше размера данных, к которому возможен доступ из одной машинной команды (типичные размеры от 1 до 16 байт). Каждая группа данных в памяти размером в 1 кэш-линию имеет порядковый номер. Для основной памяти этот номер является адресом памяти с отброшенными младшими битами. В кэше каждой кэш-линии дополнительно ставится в соответствие тег, который является адресом продублированных в этой кэш-линии данных в основной памяти.

При доступе процессора в память сначала производится проверка, хранит ли кэш запрашиваемые из памяти данные. Для этого производится сравнение адреса запроса со значениями всех тегов кэша, в которых эти данные могут храниться. Случай совпадения с тегом какой-либо кэш-линии называется попаданием в кэш (англ. cache hit), обратный же случай называется кэш-промахом (англ. cache miss). Попадание в кэш позволяет процессору немедленно произвести чтение или запись данных в кэш-линии с совпавшим тегом. Отношение количества попаданий в кэш к общему количеству запросов к памяти называют рейтингом попаданий (англ. hit rate), оно является мерой эффективности кэша для выбранного алгоритма или программы.

В случае промаха в кэше выделяется новая запись, в тег которой записывается адрес текущего запроса, а в саму кэш-линию — данные из памяти после их прочтения либо данные для записи в память. Промахи по чтению задерживают исполнение, поскольку они требуют запроса данных в более медленной основной памяти. Промахи по записи могут не давать задержку, поскольку записываемые данные сразу могут быть сохранены в кэше, а запись их в основную память можно произвести в фоновом режиме. Работа кэшей инструкций во многом похожа на вышеприведенный алгоритм работы кэша данных, но для инструкций выполняются только запросы на чтение.

Ассоциативность

Одна из фундаментальных характеристик кэш-памяти - уровень ассоциативности - отображает ее логическую сегментацию. Дело в том, что последовательный перебор всех строк кэша в поисках необходимых данных потребовал бы десятков тактов и свел бы на нет весь выигрыш от использования встроенной в ЦП памяти. Поэтому ячейки ОЗУ жестко привязываются к строкам кэш-памяти (в каждой строке могут быть данные из фиксированного набора адресов), что значительно сокращает время поиска. С каждой ячейкой ОЗУ может быть связано более одной строки кэш-памяти: например, n-канальная ассоциативность (n-way set associative) обозначает, что информация по некоторому адресу оперативной памяти может храниться в n мест кэш-памяти.

Иерархия уровней

Одной из проблем является фундаментальная проблема баланса между задержками кэша и интенсивностью попаданий. Большие кэши имеют более высокий процент попаданий но, вместе с тем, и большую задержку. Чтобы ослабить противоречие между этими двумя параметрами, большинство компьютеров использует несколько уровней кэша, когда после маленьких и быстрых кэшей находятся более медленные большие кэши (в настоящий момент — суммарно до 3 уровней в иерархии кэшей).

Многоуровневые кэши обычно работают в последовательности от меньших кэшей к большим. Сначала происходит проверка наименьшего и наибыстрейшего кэша первого уровня (L1), в случае попадания процессор продолжает работу на высокой скорости. Если меньший кэш дал промах, проверяется следующий, чуть

больший и более медленный кэш второго уровня (L2), и так далее, пока не будет запроса к основному ОЗУ.

Эксклюзивность (исключительность) и инклюзивность

Для многоуровневых кэшей требуется делать новые архитектурные решения.

Например, в некотором процессоре могут потребовать, чтобы все данные, хранящиеся в кэше L1, хранились также и в кэше L2. Такие пары кэшей называют строго инклюзивными (англ. inclusive). Другие процессоры (например, AMD Athlon) могут не иметь подобного требования, тогда кэши называются эксклюзивными (исключительными) — данные могут быть либо в L1, либо в L2 кэше, но никогда не могут быть одновременно в обоих.

Преимущество исключительных кэшей в том, что они хранят больше данных. Это преимущество больше, когда исключительный кэш L1 сопоставим по размеру с кэшэм L2, и меньше, если кэш L2 во много раз больше, чем кэш L1. Когда L1 пропускает и L2 получает доступ в случае попадания, строка кэша попадания в L2 обменивается со строкой в L1.

Время доступа

1. Кэш процессора 1го уровня (L1) — время доступа порядка нескольких тактов, размером в десятки килобайт
2. Кэш процессора 2го уровня (L2) — большее время доступа (от 2 до 10 раз медленнее L1), около полумегабайта или более
3. Кэш процессора 3го уровня (L3) — время доступа около сотни тактов, размером в несколько мегабайт (в массовых процессорах используется недавно)
4. ОЗУ системы — время доступа от сотен до, возможно, тысячи тактов, но огромные размеры в несколько гигабайт, вплоть до сотен. Время доступа к ОЗУ может варьироваться для разных его частей в случае комплексов класса NUMA (с неоднородным доступом в память)