1 Принципы современной оперативной памяти

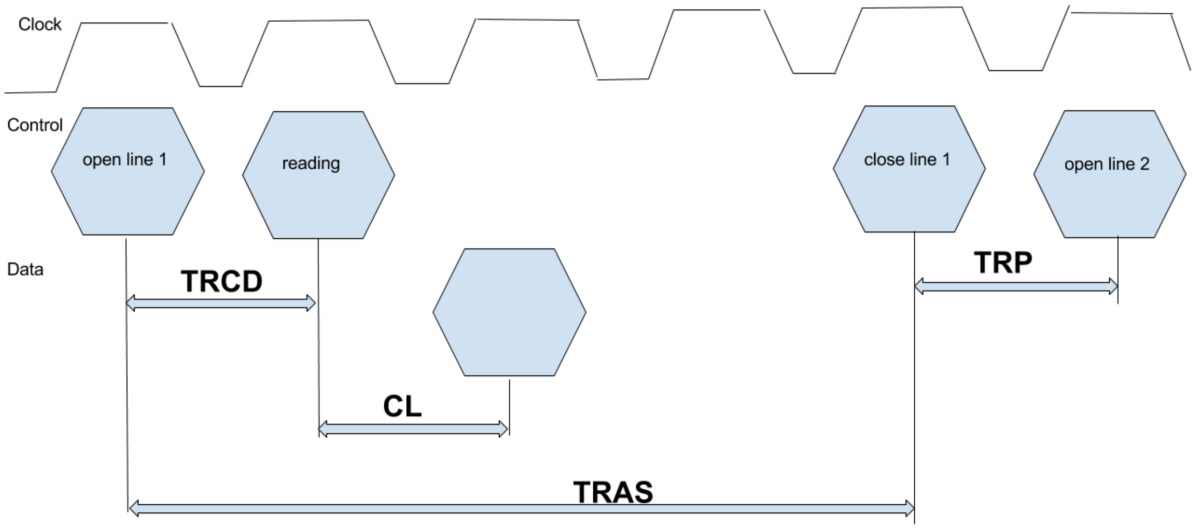


Рис. 1: Чтение из памяти.

1.1 Работа памяти

На схеме (рис. 1) показана работа памяти при чтении.

1.2 Тайминги

Тайминги указываются в тактах. Типичные значения: 9-8-8-24 (**CL-TRCD-TRP-TRAS**), а частота 1600 MHz.
Имея две планки с одинаковыми таймингами невозможно определить наиболее быструю, так как для подсчета реального времени нужно учитывать частоту.

TRCD	Row Address to Column Address Delay	Тактов между подачей сигнала на выбор строки и на выбор столбца
CL	CAS Latency = Column Address Strobe	Тактов между подачей сигнала на чтение и началом передачи данных
TRAS	Row Active Time	Тактов чтобы обеспечить прочтение (до закрытия строки)
TRP	Precharge Time	Тактов для закрытия строки, после чего можно будет активировать следующую строку

Как сравнивать два модуля оперативной памяти?

1. Если разная частота, то такты в таймингах оперативки с большей частотой будут означать пропорционально меньшие промежутки времени.
2. Если частота одинаковая, то для случайных запросов можно сравнивать сумму **TRP** и **TRAS** (т.к. часто нужно будет открывать разные строки), а для последовательных запросов – **CL**

1.3 Основные характеристики

1. Скорость доступа – время от открытия строки до получения данных (**tRCD**+**CL**)
2. Скорость передачи – время последовательного доступа к двум ячейкам (**tRAS**+**tRP**)
3. Объем

1.4 Проблемы

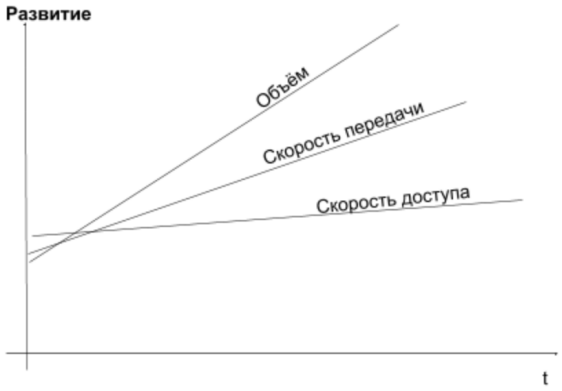


Рис. 2: Зависимость развития основных характеристик памяти от времени.

На рис. 2 объем и скорость передачи растут значительно со временем, но скорость доступа почти не сокращалась. Заметили это довольно давно и стали пытаться делать всякие хаки, чтобы улучшать время доступа.

2 Типы DRAM

2.1 FPM (Fast Page Mode)

Основная идея в том, что если мы обращаемся к какой-то строке несколько раз подряд, то не будем закрывать эту строку.
Операция открытия – чтение строки в буфер, операция закрытия – запись буфера в строку. Операции чтения/записи ведут работу с буфером.

Плюсы:

1. Быстрее считывание и запись при обращении к одной строке

2.2 EDO (Extended Data Out)

Раньше:

Приходилось держать сигнал на линии адреса до получения данных.

Улучшение относительно FPM:

Можно не держать сигнал на линии адреса (**tRCD** + **CL**), т.к. добавлен буфер для хранения адреса текущего столбца.

Почему это хорошо?
Позволяет обращаться к памяти еще до того как закончилось предыдущее обращение. Не ускоряет доступ к памяти, но повышает пропускную способность, позволяя получить больше слов в секунду.

2.3 BEDO (Burst EDO)

Улучшение относительно EDO:

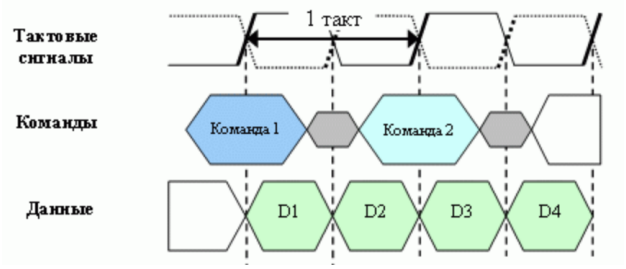
Возращает последовательно 4 ячейки строки (автоматически увеличивается адрес), поэтому ускоряется последовательная работа с ячейками памяти.

2.4 SDRAM (Synchronous dynamic random-access memory)

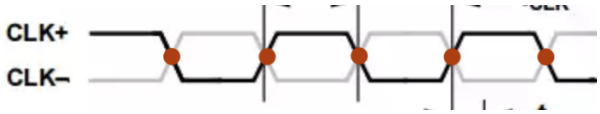
Улучшения:

1. Контроллер памяти и модуль памяти на одной тактовой частоте на одном тактовом генераторе (подробнее в FAQ: 4.1)
2. Стали применять многобанковые схемы
3. Увеличили внешнюю шину данных с 32 бит до 64 бит

2.5 DDR



(a) Устройство DDR.



(b) Дробные тайминги.

Рис. 3: DDR

1. Модуль памяти передаёт в 2 раза больше данных чем **SDRAM** на той же частоте (за счёт считывания команд и данных не только по фронту, как в **SDRAM**, но и по спаду тактового сигнала (рис. 3a).
2. Внутреннюю шину в 2 раза толще (теперь 128 бит), а внешнюю (между контроллером и модулем памяти) оставили неизменной (64 бит) (так как это дорого), но передаем первую половину данных в начале, а вторую в конце такта.
   Там 2 сигнала синхронизации в противофазе данные отдаются когда они пересекаются. (рис. 3b)

С точки зрения скорости передачи данных **DDR** эквивалентен ускоренной в 2 раза **SDRAM**. Скорость доступа не изменилась.

Частота модуля памяти и частота работы шины между модулем и контроллером памяти 200 MHz.

2.6 DDR2

Изменения относительно DDR1:

1. Увеличили в 2 раза внутреннюю шину (256 бит) и внешнюю частоту (400 MHz)
2. Скорость передачи увеличилась в 2 раза, но скорость доступа не изменилась

2.7 DDR3

Изменения относительно DDR2:

1. Увеличили в 2 раза внутреннюю шину (512 бит) и внешнюю частоту (800 MHz)
2. Скорость передачи увеличилась в 2 раза, но скорость доступа не изменилась

2.8 DDR4

Изменения относительно DDR3:

1. Увеличено число банков в 2 раза (до 16-ти)
2. Уменьшено потребление энергии по сравнению с **DDR2** и **DDR** из-за пониженного напряжения питания ячеек памяти

2.9 GDDR

GDDR – видеопамять, для которой не так важна скорость доступа, а важна скорость передачи. В GDDR больше размер внутренней шины и выше частоты. Она также позволяет снизить энергопотребление и тепловыделение при работе на равных частотах из-за увеличенной шины.

2.9.1 GDDR3

Эквивалентна DDR2.

2.9.2 GDDR5

Эквивалентна DDR3.

2.9.3 GDDR5x

В 2 раза больше скорость передачи, чем GDDR5 (увеличили частоту и шины)

2.10 HBM (High Bandwidth Memory)

Так как расстрояние от процессора до памяти большое, можно разместить память очень близко к процессору (рис. 4) (почти вплотную к кристаллу процессора). Это дает следующие преимущества:

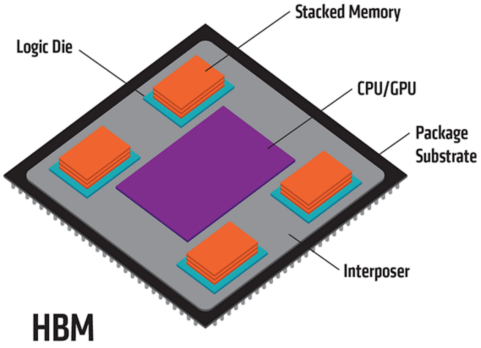


Рис. 4: HBM

1. Процессор и память соединены с помощью TSV(позволяет сделать много не мешающих друг другу проводов), поэтому можно увеличить разрядность до 1024 бит
2. Понижается татовая частота из-за увеличенной шины (тот же объем данных можно передавать с меньшей тактовой частотой), поэтому становится лучше энергоэффективность

Пока **HBM** применялась в некоторых видеокартах AMD.

3 NUMA (Non-Uniform Memory Access)

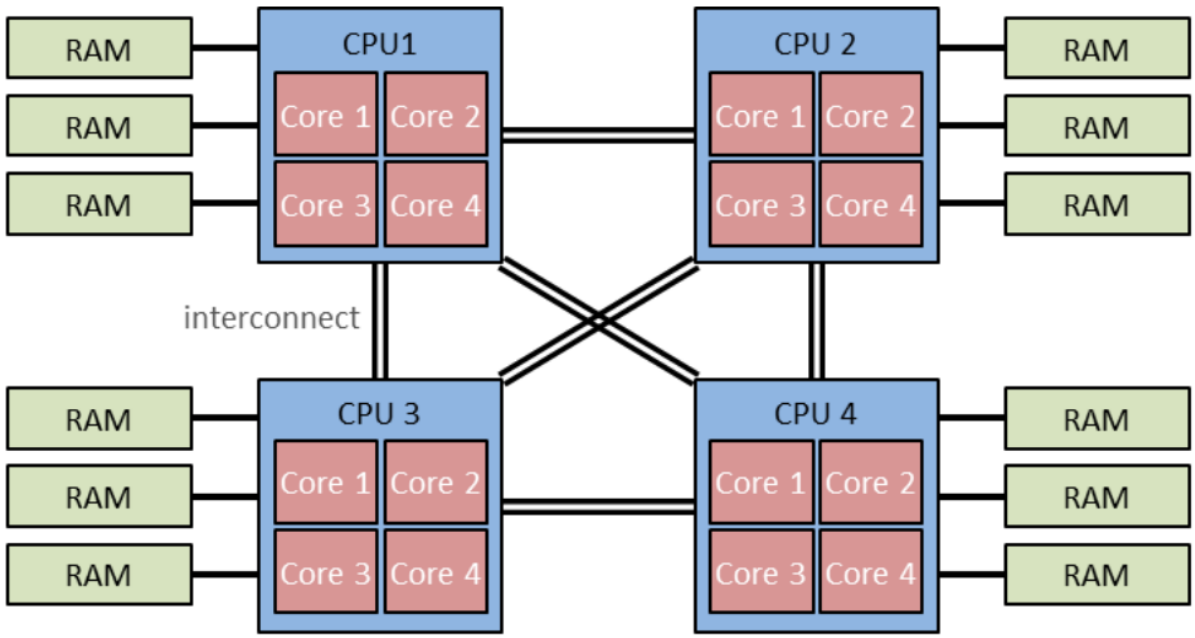


Рис. 5: Архитектура NUMA

Когда несколько CPU и у каждого стоит своя RAM рядом с ним, то другое CPU получает доступ к этой RAM значительно медленнее, чем CPU, с которым эта RAM связана. Обычно воспринимается это так: у нас есть несколько модулей RAM, которые подключены к разным CPU, но пространство логических адресов у нас одно. Вслед-

ствие этого, когда одно CPU захочет прочесть что-то по какому-то из адресов, оно может сделать это относительно быстро, а может обратиться не к своему модулю и займёт это достаточно внушительное время.

3.1 Где применяется?

На серверах, где данные тестно связаны с конкертными задачами (под какую-то задачу выделен отдельный процессор или несколько).

3.2 Решение проблем:

Первый способ бороться вообще с RAM – write-back кэширование, мы постараемся как можно реже обращаться к RAM, насколько это только возможно. Несмотря на то, что это действительно эффективное решение, появляется достаточно важная проблема: при write-back кэшировании актуальная версия данных обычно лежит в кэше, а не в RAM. И теперь нам придётся как-то следить за корректностью этих данных и их актуальностью.

4 FAQ

4.1 Почему синхронизация в SDRAM – это хорошо?

Используя асинхронную память при запросе чтения из памяти от контроллера данные приходили с задержкой, но точный промежуток времени не гарантировался. С приходом синхронизации этот контроллер точно знает через сколько тактов придет ответ на запрос.

4.2 Бывают ли дробные тайминги? Как?

Смотри главу 2.5

4.3 Как мы получаем в DDR информацию два раза за такт?

Смотри главу 2.5

4.4 Есть организация, которая не-NUMA?

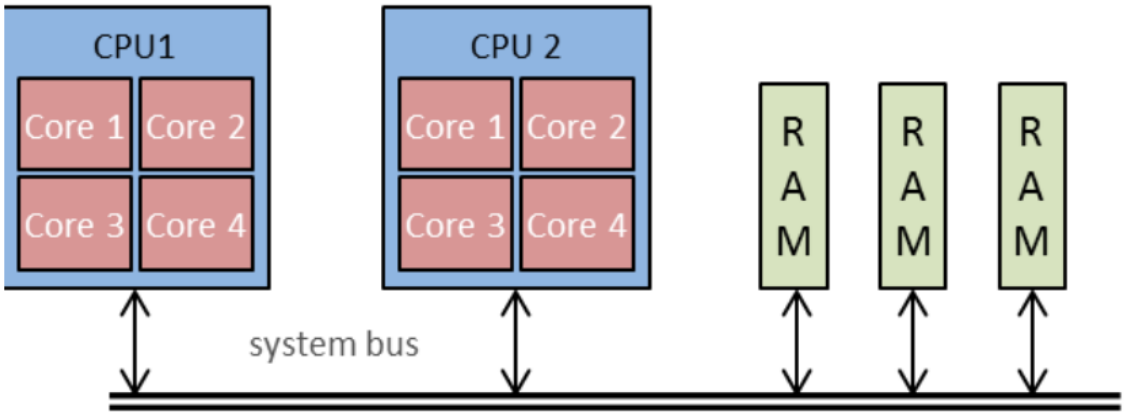


Рис. 6: Архитектура UMA

4.5 От чего зависит скорость передачи и скорость доступа?

Скорость доступа от таймингов и частоты памяти, скорость передачи от частоты внешней шины, от её разрядности.

4.6 Отличия DDR от GDDR

Смотри главу 2.9

4.7 Почему нельзя увеличивать частоту модуля памяти?

При высокой частоте модуля памяти будут возникать ошибки в оперативной памяти (создаются магнитные поля, которые могут повлиять на целостность данных)