

## 1 Обычная DRAM

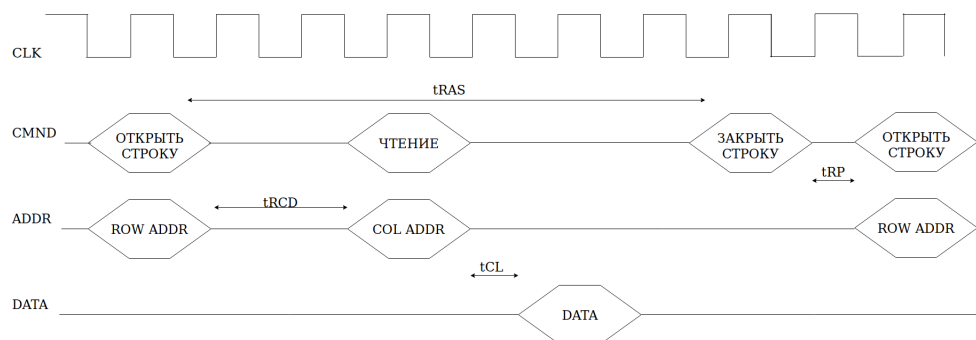


Рис. 1: Схема работы DRAM

Вернемся к устройству модуля памяти. За два такта передаём адрес. После выбора строки все ячейки этой строки считываются в отдельную маленькую статическую память. На этот момент в самом модуле в строке одни 0, т.к. конденсаторы разрядились при чтении. Уже из статической памяти считываем столбец и всю строку записываем обратно в динамическую.

*Внимание\_0:* эта штука **асинхронная**.

*Внимание\_1:* адрес столбца должен быть зажат пока не получены данные.

## 2 Тайминги

- **CL** - *CAS (Column Address Strobe) Latency* - число тактов между отправкой адреса столбца и началом получения данных.
- **tRCD** - *RAS (Row Address Strobe) to CAS Delay* - минимальное число тактов, которое нужно подождать при открытой строке, прежде чем открывать столбец.
- **tRP** - *Row Precharge* - минимальное число тактов, которое нужно подождать между закрытием строки и открытием новой.
- **tRAS** - *Row Active Strobe* - минимальное число тактов между открытием и закрытием строки.

## 3 Как сравнивать два модуля памяти?

- Если частота одинаковая, то чем меньше CL, тем лучше для последовательных запросов. Чем меньше  $tRP + tRAS$ , тем лучше для случайных запросов.
- Если частота разная, то такты в памяти с большей частотой будут означать пропорционально меньшие промежутки времени.

## 4 Характеристики памяти

Скорость бывает двух видов: **скорость доступа** *latency* и **скорость передачи данных** или **пропускная способность** *throughput*.

**Скорость доступа** характеризует время от получения запроса до получения первой порции данных. В RAM скорость доступа =  $t_{CL} + t_{RCD}$

**Скорость передачи данных** характеризует время от  $n$ -ной до  $n + 1$ -ой порции данных. В RAM скорость передачи данных =  $t_{RAS} + t_{RP}$

Как видно из графика, со скоростью доступа всё очень грустно.

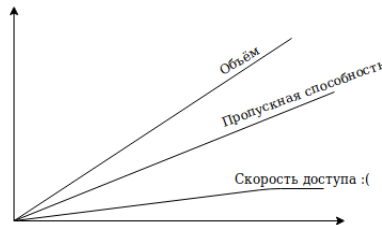


Рис. 2: График, показывающий как изменялись характеристики памяти с течением времени

## 5 Важная информация

Память - *пассивное* устройство. Процессор *запрашивает*, память *отвечает на запрос*.

## 6 FPM DRAM

**FPM DRAM** - *Fast Page Mode DRAM* - а давайте закрывать строку не каждый раз, а только когда следующий запрос идет в другую строку. Т.е. читать столбцы одной строки без закрытия этой строки.

время доступа для ячеек из одной строки улучшилось.

*Внимание\_0*: эта штука **асинхронная**.

*Внимание\_1*: адрес столбца должен быть зажат пока не получены данные.

## 7 EDO DRAM

**EDO DRAM** - *Enabled Data Out DRAM* - а давайте добавим буфер для адреса столбца. Теперь пока читается один столбец, мы можем уже подать адрес следующего, а после получения данных с первого столбца, сигнал на чтение второго.

Слегка улучшилось время обращения в одной строке, слегка улучшилась скорость передачи, и немного - время доступа.

*Внимание*: эта штука всё ещё **асинхронная**.

## 8 BEDO DRAM

**BEDO DRAM** - *Burst EDO DRAM* - теперь на запрос чтения столбца память отдаёт не только нужный столбец, но и три следующих. Ограничения: адрес должен быть кратен четырем.

Улучшилась скорость передачи.

*Внимание*: эта штука всё ещё **асинхронная**.

## 9 SDRAM

До появления SDRAM контроллер памяти и модуль памяти *не были синхронизированы*. У них даже могли быть разные частоты.

Пусть у нас  $t_{RCD}$  - 3 такта. Но контроллеру приходилось ждать "с запасом".

**SDRAM** - *Synchronous DRAM* - теперь контроллер и память на одной синхронизации. Следовательно контроллер может ждать ровно  $t_{RCD}$  тактов.

Кроме того, в SDRAM увеличили ширину шины до 64 бит.

За счет поджатия таймингов увеличилась скорость передачи.

Внутренняя и внешняя шина SDR SDRAM работают на одной частоте. Если хочется, чтобы память работала быстрее, нужно поднимать частоты. А поднимать частоты достаточно грустно, потому что растет энергопотребление и тепловыделение.

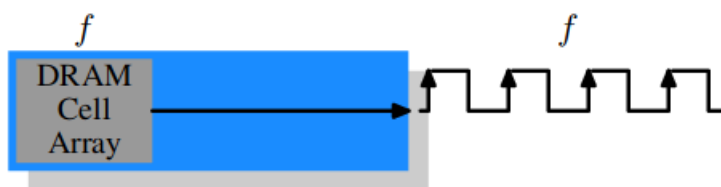


Рис. 3: SDR SDRAM - Single Data Rate SDRAM

## 10 Еще немного про модули памяти

Процессор явно работает быстрее, чем память. Следовательно у памяти рано или поздно появится очередь запросов. Хотелось бы сделать такую память, которая может обрабатывать запросы параллельно.

### 10.1 Многопортовая память

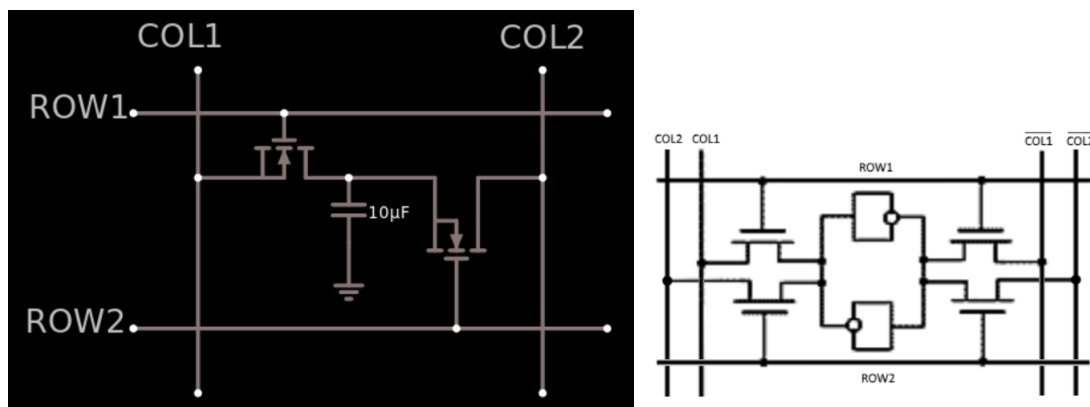


Рис. 4: Двухпортовые ячейки DRAM(слева) и SRAM(справа)

При таком подходе к каждой ячейке подходит два набора линий строки/столбца. Идейно мы как будто бы обращаемся к двум разным матрицам, заполненным одинаковыми данными. **НО!** Мы не можем обратиться одновременно к одной и той же строке.

Улучшается время доступа к разным строкам.  
На практике используется в SRAM, а в DRAM нет.

## 10.2 Многобанковая память

Идейно среднее между однобанковой и многобанковой памятью)000))))

При таком подходе используем действительно две разных матрицы. Усложняется толь-

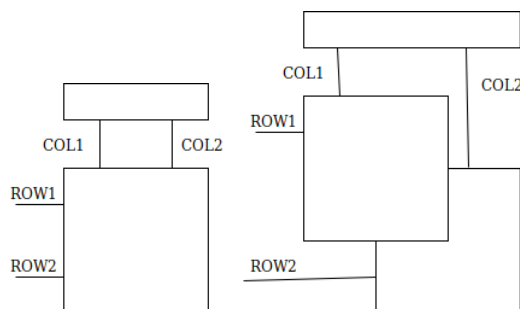


Рис. 5: Модуль многопортовой и многобанковой памяти

ко интерфейсная схема, а сами ячейки остаются простыми.

В итоге получаем улучшения при работе со строками, лежащими в разных банках памяти.

## 11 DDR

**DDR SDRAM** - *Double Data Rate SDRAM* - увеличили внутреннюю шину в два раза, а ширину внешней оставили прежней. Можем передавать данные два раза за такт: по фронту и по спаду импульса синхронизации. (На пересечении прямого и инвертированного импульса синхронизации)

**Маркетинговая уловка:** а давайте введем понятие эффективной частоты (на какой частоте нужно было бы работать старому модулю памяти, чтобы передавать столько же информации). Нормальная частота не поменялась, но теперь можно говорить, что эффективная частота в два раза выше, чем раньше. Такие дела.

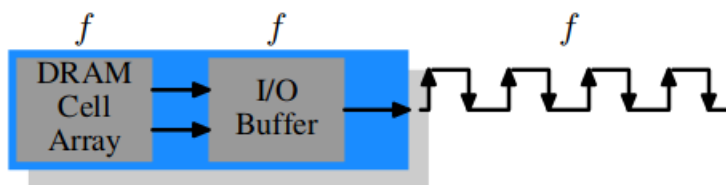


Рис. 6: DDR SDRAM

## 12 DDR2

Еще раз увеличили ширину внутренней шины в два раза, и подняли частоту внешней в два раза. Снизили напряжение питания.

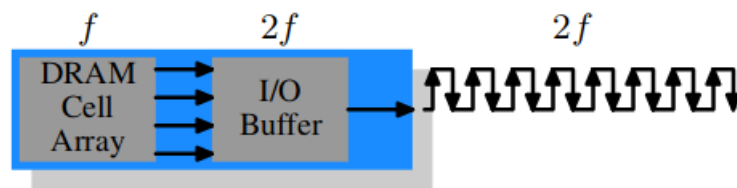


Рис. 7: DDR2 SDRAM

## 13 DDR3

Еще раз увеличили ширину внутренней шины в два раза, и подняли частоту внешней в два раза. Снизили напряжение питания.

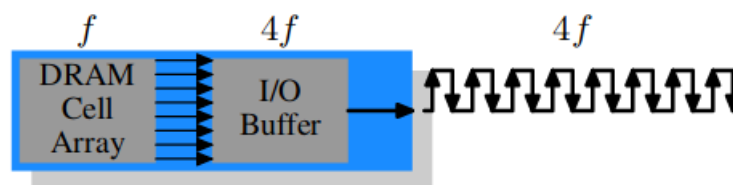


Рис. 8: DDR3 SDRAM

## 14 DDR4

Еще раз увеличили ширину внутренней шины... А вот и нет. Дальше увеличивать ширину внутренней шины уже слишком жирно (там и так  $8n$  проводов, если в SDRAM  $n$ ). Поэтому только уменьшили напряжение питания.

## 15 GDDR

**Graphic DDR** - оптимизирована на высокую скорость передачи данных. Сильно греются, имеют высокую тактовую частоту.

GDDR3 DDR2 GDDR5 DDR3, вроде как однопортовая, но хорошо притворяется двухпортовой из-за внутренних хаков. GDDR5X DDR3

## 16 HBM память\*

Особая технология подключения памяти к процессору. Используется в топовых видеокарточках. Ширина шины сильно больше. Но технология сильно дороже и сложнее, чем обычные печатные платы.

## 17 Источники

- Википедия

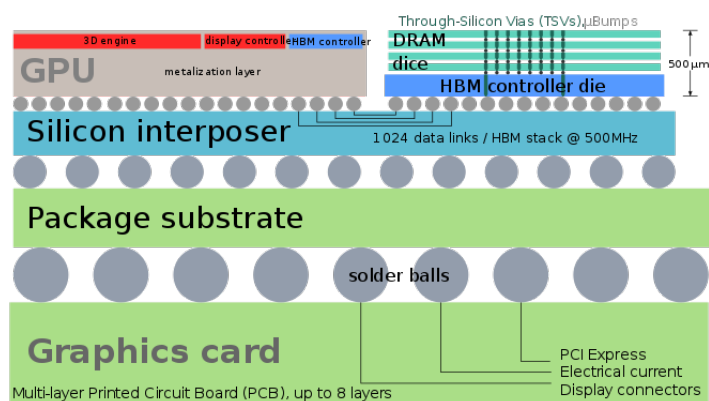


Рис. 9: Технология HBM в разрезе

- конспекты лекций by @ntwwwnt
- "What Every Programmer Should Know About Memory" by Ulrich Drepper