

Hierarquia de Memória - Parte 1 - Introdução

Arquitetura de Computadores

Emilio Francesquini

e.francesquini@ufabc.edu.br

2020.Q1

Centro de Matemática, Computação e Cognição

Universidade Federal do ABC



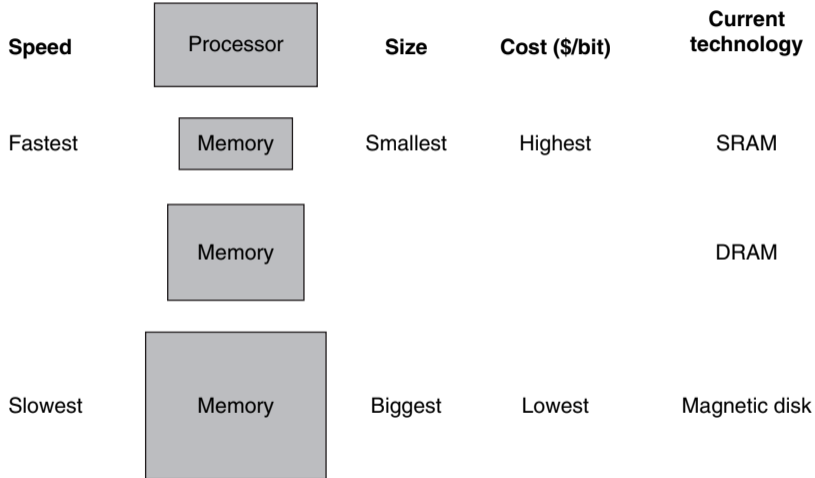
- Estes slides foram preparados para o curso de **Arquitetura de Computadores** na UFABC.
- Este material pode ser usado livremente desde que sejam mantidos, além deste aviso, os créditos aos autores e instituições.
- O conteúdo destes slides foi **baseado no conteúdo do livro** *Computer Organization And Design: The Hardware/Software Interface*, 5th Edition.



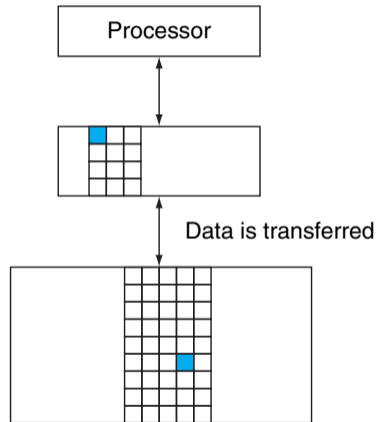
Introdução

- Programas tendem a acessar uma pequena fatia do espaço de memória disponível.
- **Localidade temporal** (*en: temporal locality*).
 - ▶ Posições da memória que foram acessadas recentemente têm alta probabilidade de serem acessadas em breve novamente.
 - Ex: instruções em um laço, variáveis de indução
- **Localidade espacial** (*en: spacial locality*).
 - ▶ Posições de memória próximas às que já foram acessadas, tendem a ser acessadas em breve.
 - ▶ Ex: acesso sequencial de instruções, dados em uma array

- Utilizando hierarquia de memória
- Guardar tudo no disco
- Copiar dados recentemente acessados (e aqueles próximos dos acessados) do disco para memórias menores, podem mais rápidas (como DRAM - também chamada de **memória principal**).
- Copiar os dados mais recentemente acessados ainda (e aqueles próximos) da DRAM para memória SRAM (também chamadas de caches)
 - ▶ Caches são ligadas diretamente à CPU e frequentemente estão no mesmo chip.



- **Blocos** (ou **linhas**): unidade mínima de cópia
 - ▶ Podem ser múltiplas palavras do processador
- Se o dado ao qual se deseja acesso estiver nos níveis mais altos, dizemos que houve um **acerto** (**hit**)
 - ▶ **taxa de acertos** (**hit ratio**): razão de hits pelo número de acessos
- Se o dado ao qual se deseja acesso não estiver nos níveis mais altos, dizemos que houve uma **falha** (**miss**)
 - ▶ **taxa de falhas** (**miss ratio**): razão de misses pelo número de acessos ou $1 - \text{hit ratio}$
 - ▶ No caso de falhas, o dado é recuperado do nível inferior e fornecido ao nível superior.



- **Hit time**

- ▶ Tempo necessário para acessar um nível de memória da hierarquia incluindo o tempo necessário para determinar se o acesso é um hit ou miss.

- **Miss penalty**

- ▶ Tempo necessário para trazer um bloco do nível de memória de um nível inferior para o superior, incluindo o tempo para acessar o bloco, transmiti-lo de um nível a outro, colocá-lo no nível onde o miss ocorreu e repassar o bloco para quem o requisitou.

Tecnologias de memória

- Static RAM (SRAM)
 - ▶ 0.5ns – 2.5ns, \$500 – \$1000 / GB
- Dynamic RAM (DRAM)
 - ▶ 50ns – 70ns, \$10 – \$20 / GB
- SSD (Flash)
 - ▶ 5.000ns - 50.000ns, \$0.75 - \$1 /GB
- Disco
 - ▶ 5.000.000ns – 20.000.000ns, \$0.05 – \$0.10 / GB
- Sonho de consumo:
 - ▶ Tempo de acesso de SRAM
 - ▶ Capacidade e custo / GB de um disco

- Dados armazenados em um capacitor
- Um único transistor é usado para acessar o dado
- Precisa ser periodicamente atualizada (*refreshed*)
 - ▶ Leituras destrutivas
 - ▶ Feitas em uma **linha** (*row*)

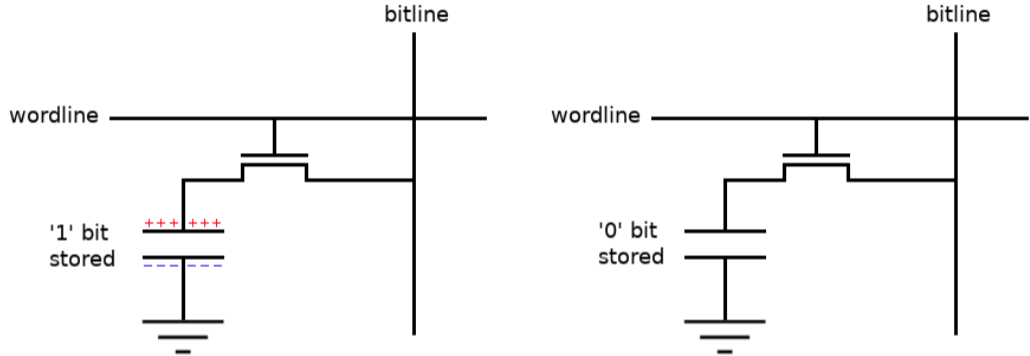


Figura: <https://www.allaboutcircuits.com/technical-articles/introduction-to-dram-dynamic-random-access-memory/>

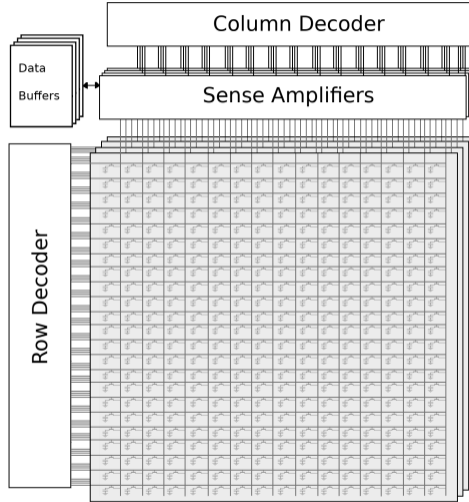




Figura: <https://www.allaboutcircuits.com/technical-articles/introduction-to-dram-dynamic-random-access-memory/>

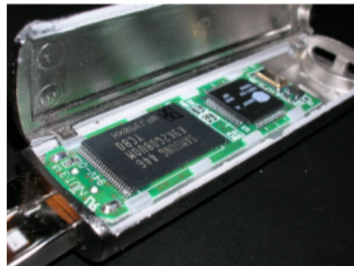
- Bits na DRAM são organizados em linhas em uma matriz retangular
 - ▶ Quando há um acesso, acessa-se uma linha inteira
 - ▶ *Burst mode* ocorre quando acessamos palavras sucessivas de uma mesma linha¹.
- DDR RAM - Double Data Rate
 - ▶ Transfere nas bordas ascendentes e descendentes do clock
- QDR RAM - Quad data rate
 - ▶ Canais DDR separados para entradas e saídas

¹Em inglês usa-se as palavras row e line. Row invariavelmente diz respeito à linha da matriz da DRAM e line/block às unidades de transferência entre os níveis hierárquicos da memória

Year introduced	Chip size	\$ per GiB	Total access time to a new row/column	Average column access time to existing row
1980	64 Kibibit	\$1,500,000	250 ns	150 ns
1983	256 Kibibit	\$500,000	185 ns	100 ns
1985	1 Mebibit	\$200,000	135 ns	40 ns
1989	4 Mebibit	\$50,000	110 ns	40 ns
1992	16 Mebibit	\$15,000	90 ns	30 ns
1996	64 Mebibit	\$10,000	60 ns	12 ns
1998	128 Mebibit	\$4,000	60 ns	10 ns
2000	256 Mebibit	\$1,000	55 ns	7 ns
2004	512 Mebibit	\$250	50 ns	5 ns
2007	1 Gibibit	\$50	45 ns	1.25 ns
2010	2 Gibibit	\$30	40 ns	1 ns
2012	4 Gibibit	\$1	35 ns	0.8 ns

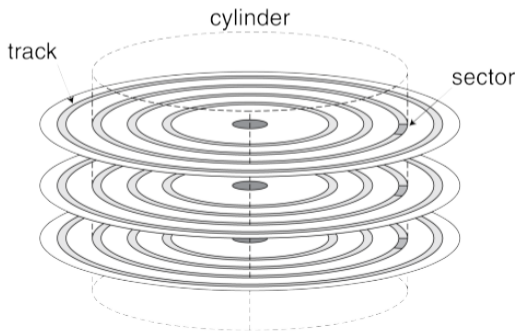
- Row Buffer
 - ▶ Permite que várias palavras sejam lidas e atualizadas em paralelo
- Synchronous DRAM - SDRAM
 - ▶ Permite acessos consecutivos em bursts sem necessidade de mandar cada endereço
 - ▶ Aumenta a banda
- DRAM Banking
 - ▶ Permite acesso simultâneo a vários chips de DRAM
 - ▶ Aumenta a banda

- Armazenamento não volátil
- De 100 a 1000× mais rápida que discos
- Menor, consome menos energia, mais robusta
- Contudo, mais cara e se posiciona entre disco e DRAM



- NOR Flash: cada célula se assemelha a uma porta NOR
 - ▶ Acesso aleatório de leitura/escrita
 - ▶ Usada para memória de instruções em sistemas embarcados
- NAND Flash: cada célula se assemelha a uma porta NAND
 - ▶ Mais densa, porém só permite acesso em blocos
 - ▶ Mais barata por GB
 - ▶ Usada para pen drives, SSDs, ...
- Flash se degrada (*wear*) com o uso depois de alguns milhares de ciclos de escrita
 - ▶ Não é apropriada para substituir, diretamente, RAM ou disco
 - ▶ Mecanismos de *wear leveling* remapeiam dados para os blocos menos acessados

- Armazenamento de dados não volátil em discos giratórios magnéticos



- Cada setor armazena
 - ▶ Seu ID
 - ▶ 512 a 4096 bytes de dados
 - ▶ Códigos de correção de erros (ECC)
 - Usado para esconder defeitos e erros de gravação
 - ▶ Campos de sincronização e algumas lacunas
- Acessar um setor envolve
 - ▶ Aguardar caso outro acesso já esteja ocorrendo
 - ▶ **Busca** (*seek*): move a cabeça até a trilha desejada
 - ▶ **Latência Rotacional** (*rotational latency*)
 - ▶ Transferência de dados
 - ▶ Eventuais overheads do controlador

■ Dados

- ▶ setores de 512 bytes, 15.000 RPM, 4ms de tempo médio de busca, 100 MB/s de taxa de transferência, overhead do controlador de 0.2 ms e que o disco não está ocupado

■ Tempo médio para leitura

- ▶ 4ms de busca +
- ▶ $0.5 / (15000/60) = 2$ ms de latência rotacional +
- ▶ $512 \text{ B} / 100 \text{ MB/s} = 0.005\text{ms}$ de transferência +
- ▶ 0.2 ms do controlador =
- ▶ Total: 6.2 ms

- Fabricantes anunciam o tempo de busca médio
 - ▶ Baseando-se em todas as buscas possíveis
 - ▶ Localidade e o próprio SO acabam levando a um tempo de busca inferior
- Discos frequentemente incluem caches
 - ▶ Que buscam antecipadamente setores que podem ser necessários
 - ▶ Evitam tempo de busca e latência rotacional