

Aulas 20, 21 e 22

- Organização da memória de um sistema computacional
- Hierarquia do sistema de memória
- Organização genérica de um circuito de memória a partir de uma célula básica
- Memória SRAM (*Static Random Access Memory*):
 - organização de células básicas num array
 - ciclos de acesso para leitura e escrita: diagramas temporais
 - construção de módulos de memória SRAM
- Memória DRAM (*Dynamic Random Access Memory*) :
 - célula básica; organização interna
 - ciclos de acesso para leitura e escrita: diagramas temporais
 - refrescamento: modo "RAS only"
 - construção de módulos de memória DRAM

José Luís Azevedo, Arnaldo Oliveira, Tomás Silva, Bernardo Cunha

Introdução

- Pretensão do utilizador:
 - Uma memória rápida e com grande capacidade de armazenamento 😊
 - Que custe o preço de uma memória lenta... 😊
- Solução perfeita para este dilema não existe
- A organização da memória de um sistema computacional resulta de um compromisso entre:
 - Velocidade
 - Capacidade
 - Custo
 - Consumo energético
- Menor tempo de acesso: maior custo por bit
- Maior capacidade: maior tempo de acesso
- **Solução:** Criar a ilusão de uma memória rápida de grande capacidade através da utilização das várias tecnologias de memória disponíveis, segundo uma hierarquia

Introdução

- Tecnologias de memória

Tecnologia	Tempo Acesso	\$ / GB
SRAM	0,5 – 2,5 ns	\$500 - \$1000
DRAM	35 - 70 ns	\$10 - \$20
Flash	5 – 50 us	\$0,75 - \$1
Magnetic Disk	5 - 20 ms	\$0,005 - \$0,1

(Dados de 2012)

- SRAM - Static Random Access Memory
- DRAM - Dynamic Random Access Memory
- Dadas estas diferenças de custo e de tempo de acesso, é vantajoso construir o sistema de memória como uma hierarquia onde se utilizem todas estas tecnologias

Introdução

- Memória DRAM (Dynamic RAM)

Ano	Capacidade (max. por chip)	Access Time	\$ / MB
1980	64 Kbit	250 ns	\$1500
1983	256 Kbit	185 ns	\$500
1985	1 Mbit	135 ns	\$200
1989	4 Mbit	110 ns	\$50
1992	16 Mbit	90 ns	\$15
1996	64 Mbit	60 ns	\$10
1998	128 Mbit	60 ns	\$4
2000	256 Mbit	55 ns	\$1
2004	512 Mbit	50 ns	\$0.25
2007	1024 Mbit	45 ns	\$0.05
2010	2 Gbit	40 ns	\$0.03
2012	4 Gbit	35 ns	\$0.001

Introdução

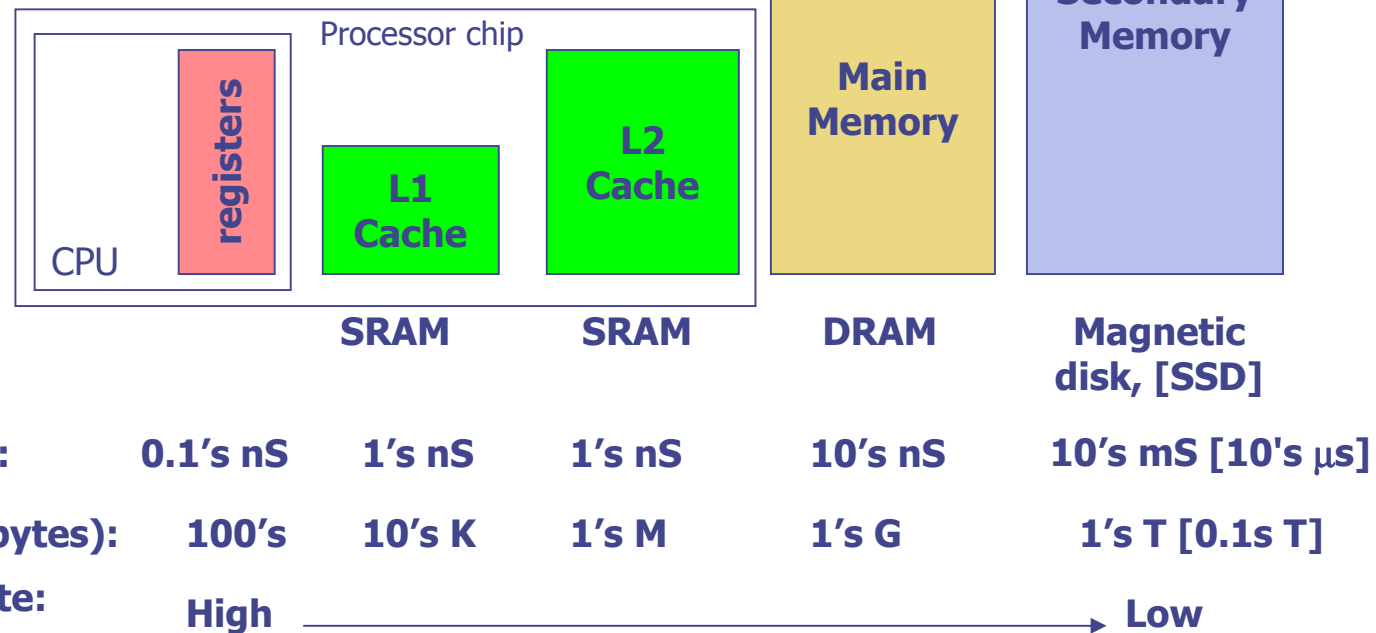
- O processador deve ser alimentado de instruções e dados a uma taxa que não comprometa o desempenho do sistema
 - Processador funciona a n GHz
 - Memória funciona a nnn MHz
- Diferença entre a velocidade do processador e da memória (DRAM) tem vindo a aumentar
- Solução:
 - Guardar a informação mais vezes utilizada numa memória rápida (*static* RAM) de pequena dimensão "próxima" do CPU
 - Aceder raramente à memória principal (mais lenta) para obter os restantes dados (apenas quando necessário)
 - Transferir blocos de informação da memória principal
- Conceito
 - Cache (ver aulas seguintes)

Utilização da hierarquia de memória

- Solução 1 – expor a hierarquia
 - Alternativas de armazenamento: registos internos do CPU, memória rápida, memória principal, disco
 - Cabe ao programador utilizar racionalmente estas alternativas de armazenamento
 - Exemplo de processador que usa esta técnica: Cell microprocessor (Cell Broadband Engine Architecture) que equipa a PlayStation 3 e algumas televisões
- Solução 2 – esconder a hierarquia
 - Modelo de programação:
 - Tipo de memória único
 - Espaço de endereçamento único
 - A máquina gere automaticamente o acesso à memória
 - Solução usada na maioria dos processadores contemporâneos

Hierarquia de memória

- Memória organizada em níveis
- Informação nos níveis superiores é um subconjunto da dos níveis inferiores
- Informação circula apenas entre níveis adjacentes
- Bloco – Quantidade de informação que circula entre níveis adjacentes



Tipos de memória

- RAM – Random Access Memory
 - Designação para memória volátil que pode ser lida e escrita
 - Acesso "random"
- ROM – Read Only Memory
 - Memória não volátil que apenas pode ser lida
 - Acesso "random"

(Acesso "random" - tempo de acesso é o mesmo para qualquer posição de memória)

Tecnologias de memória

- Tecnologias:
 - Semicondutor
 - Magnética
 - Ótica
 - Magneto-ótica
- Memória volátil:
 - Informação armazenada perde-se quando o circuito é desligado da alimentação: RAM (SRAM e DRAM)
- Memória não volátil:
 - A informação armazenada mantém-se até ser deliberadamente alterada: EEPROM, Flash EEPROM, tecnologias magnéticas

Tecnologias de memória não volátil

- **ROM** – programada durante o processo de fabrico
- **PROM** – Programmable Read Only Memory: programável uma única vez
- **EPROM** – Erasable PROM: escrita em segundos, apagamento em minutos (ambas efectuadas em dispositivos especiais)
- **EEPROM** – Electrically Erasable PROM
 - O apagamento e a escrita podem ser efetuados no próprio circuito em que a memória está integrada
 - O apagamento é feito byte a byte
 - Escrita muito mais lenta que leitura
- **Flash EEPROM** (tecnologia semelhante à EEPROM)
 - A escrita pressupõe o prévio apagamento das zonas de memória a escrever
 - O apagamento é feito por blocos (por exemplo, blocos de 4 kB) o que torna esta tecnologia mais rápida que a EEPROM
 - O apagamento e a escrita podem ser efetuados no próprio circuito em que a memória está integrada
 - Escrita muito mais lenta que leitura

Memória do tipo RAM (volátil)

- **SRAM – Static RAM**

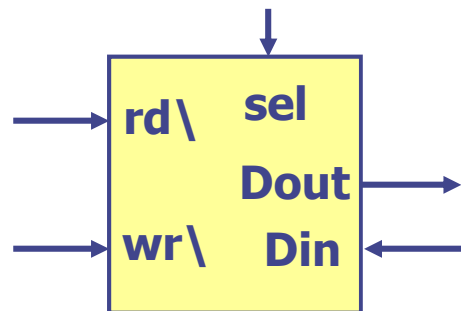
- Vantagens:
 - Rápida
 - Informação permanece até que a alimentação seja cortada
- Inconvenientes:
 - Implementações típicas: 6 transistores / célula
 - Baixa densidade, elevada dissipação de potência
 - Custo/bit elevado

- **DRAM – Dynamic RAM**

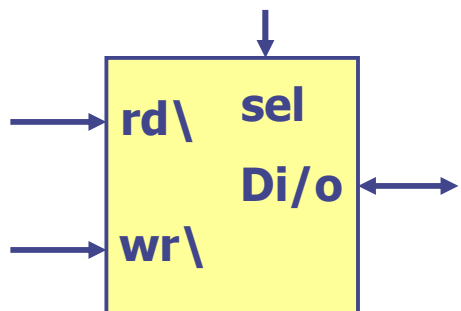
- Vantagens:
 - Implementações típicas: (1 transistor + 1 condensador) / célula
 - Alta densidade, baixa dissipação de potência
 - Custo/bit baixo
- Inconvenientes:
 - Informação permanece apenas durante alguns mili-segundos (necessita de *refresh* regular – daí a designação "dynamic")
 - Mais lenta (pelo menos 1 ordem de grandeza) que a SRAM

Organização básica de memória

- Uma memória pode ser encarada como uma coleção de M registos de dimensão N ($M \times N$)
- Cada registo é formado por N células, cada uma delas capaz de armazenar 1 bit
- Uma célula de memória (de 1 bit) pode ser representada por:



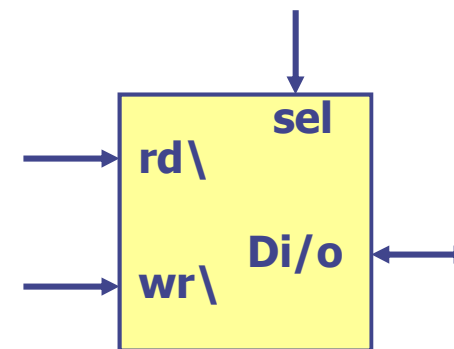
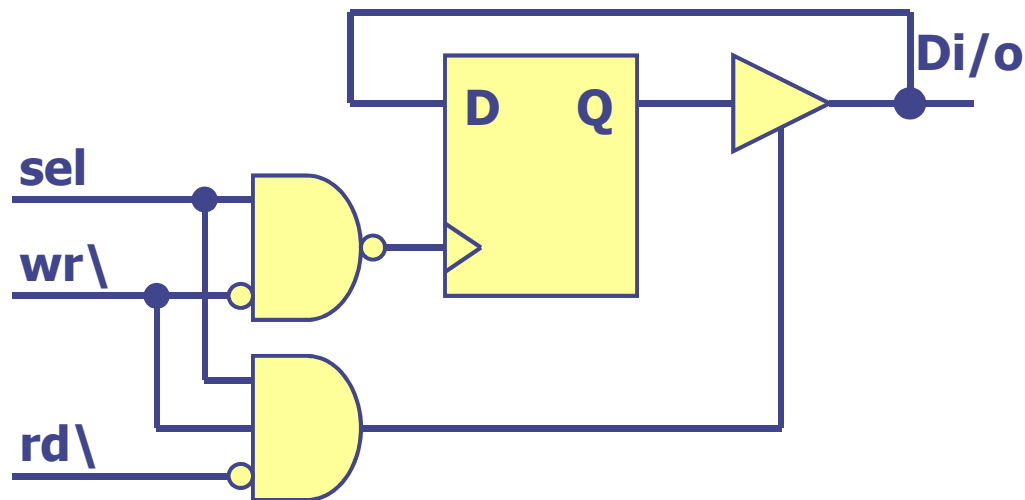
Din	– Data In (1 bit)
Dout	– Data Out (1 bit)
sel	– Select
rd\'	– Read\'
rw\'	– Write\'



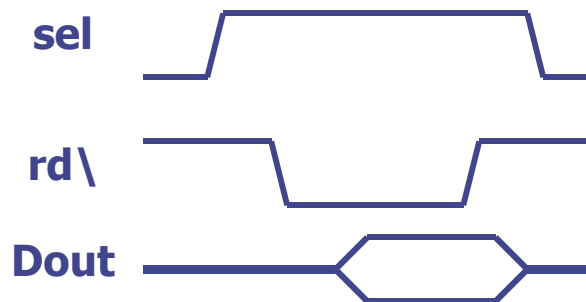
Di/o	– Data In/Out (1 bit)
sel	– Select
rd\'	– Read\'
rw\'	– Write\'

Organização básica de memória

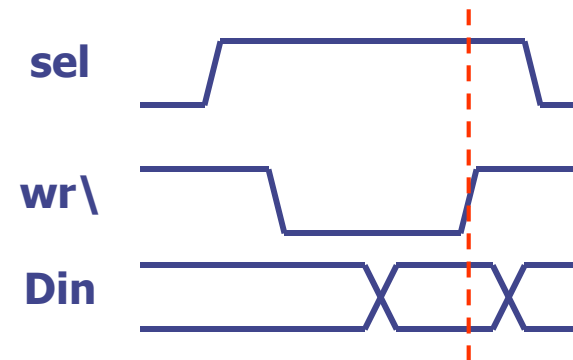
- Uma possível implementação de uma célula de memória é:



Operação de leitura



Operação de escrita

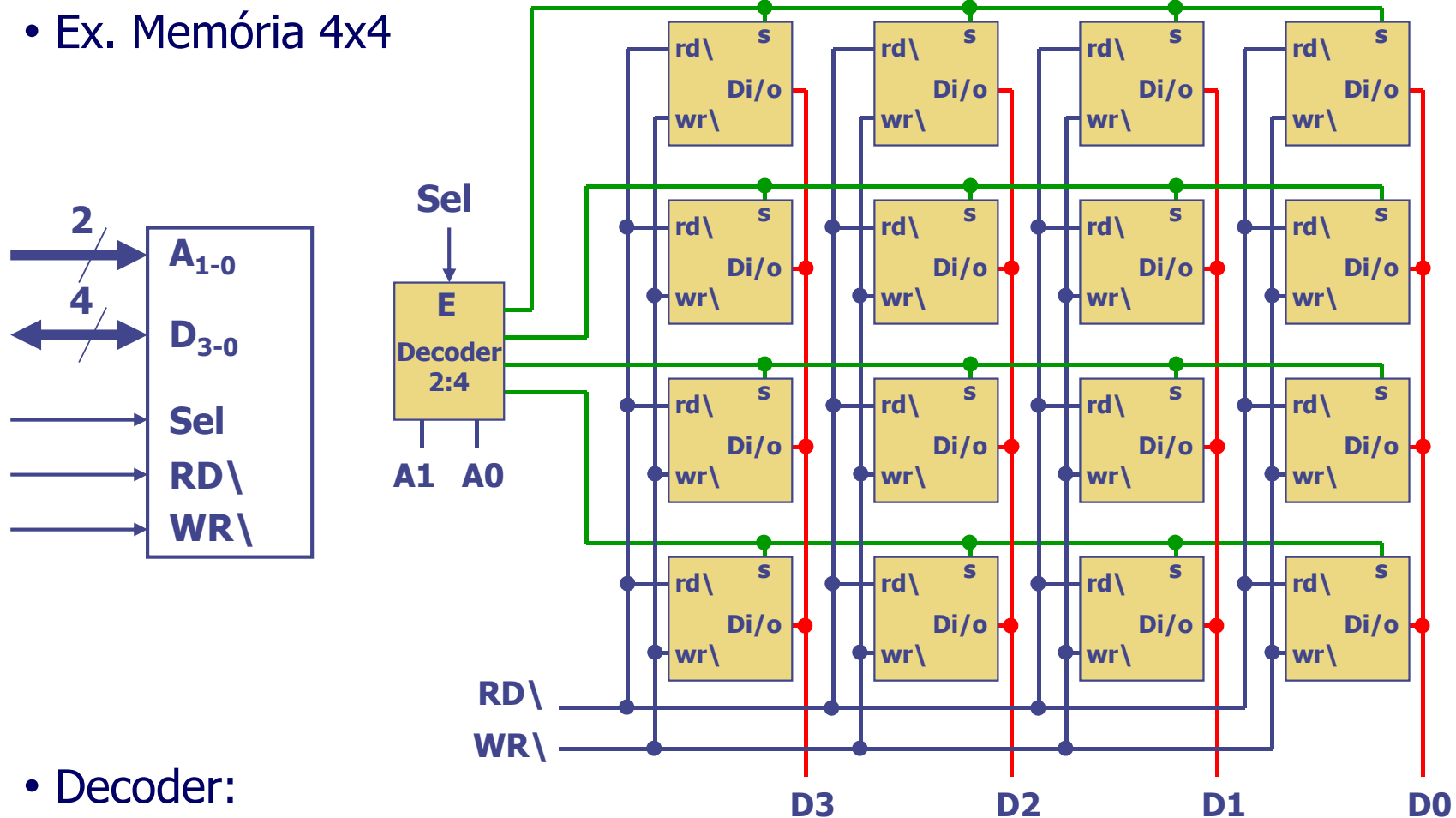


Agrupamento de células de memória

- Através do agrupamento de células-base pode formar-se uma memória de maior dimensão
- O que é necessário especificar:
 - **Word size** (x1, x4, x8, x16, 32, ...)
 - O **número total de words** que a memória pode armazenar
(Número total de bits = word size * n^o words)
- Exemplo: 1Mx4
 - 4 bits / word
 - $1M = 2^{20} \rightarrow 20$ linhas de endereço $\rightarrow 1.048.576$ endereços

Organização 2D

- Ex. Memória 4x4



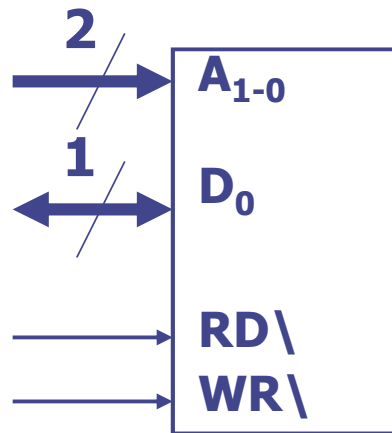
- Decoder:

- 2^N saídas

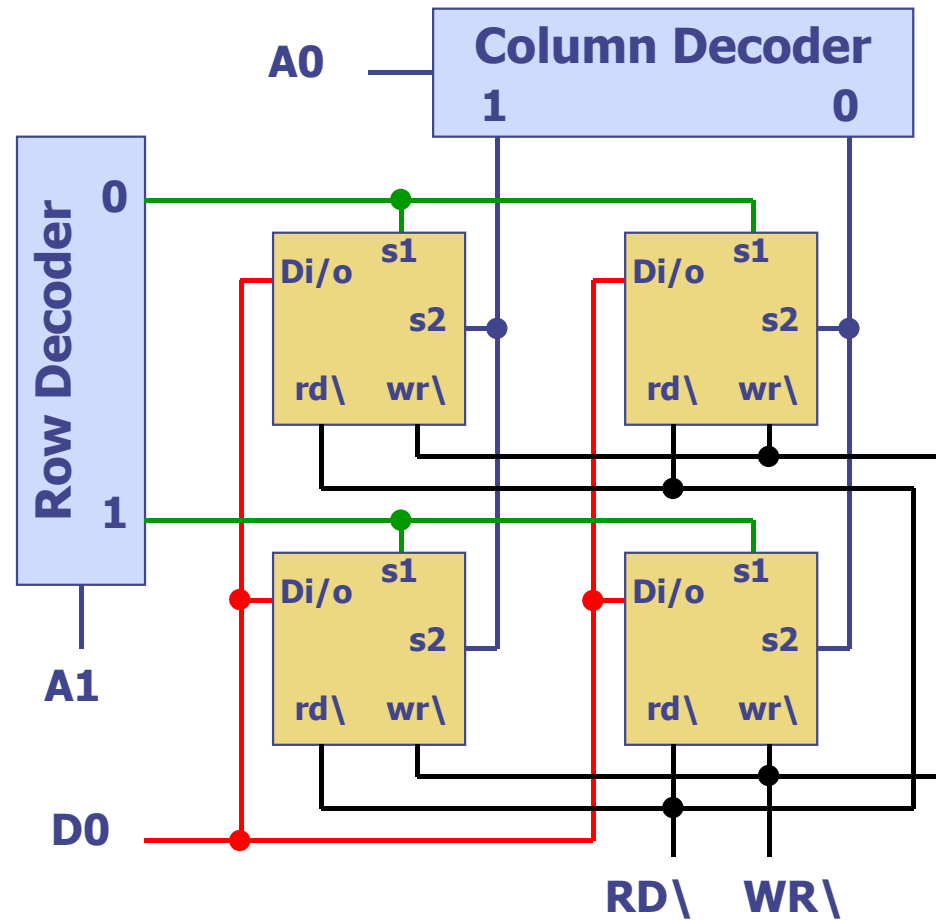
- Ex. $1M \times 4 = 2^{20} \times 4 \rightarrow 1.048.576$ saídas, N° de gates $\gg 2^{20}$

Organização em matriz (conceito)

- Ex. Memória 4x1



(Célula seleccionada: S1.S2=1)

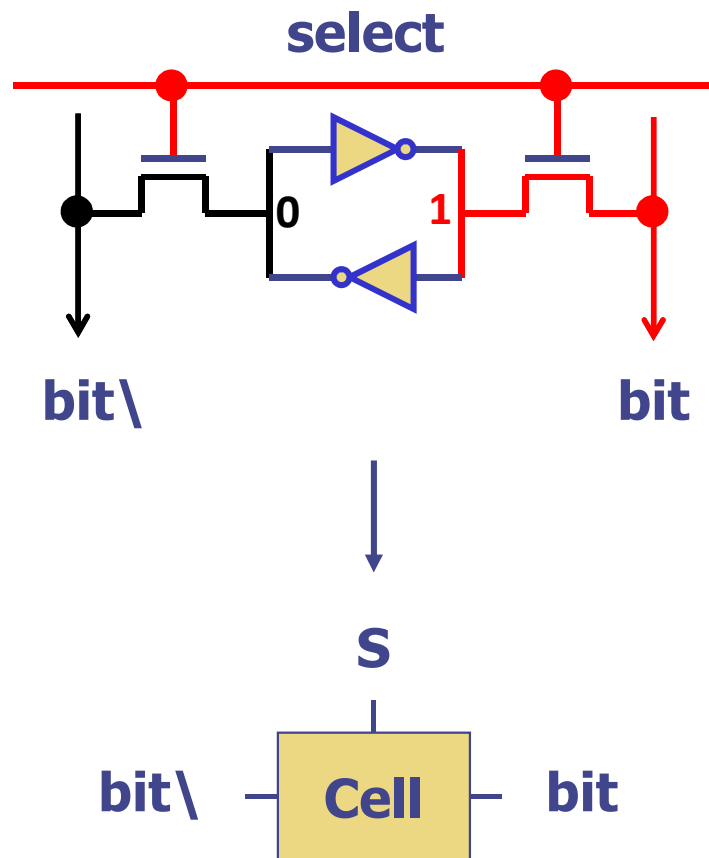


Q1. E se a memória fosse de 8x1? E se fosse 16x1? e 1Mx1?

Q2. E se a memória fosse de 4x4?

RAM estática (SRAM)

- 6 transistores / célula



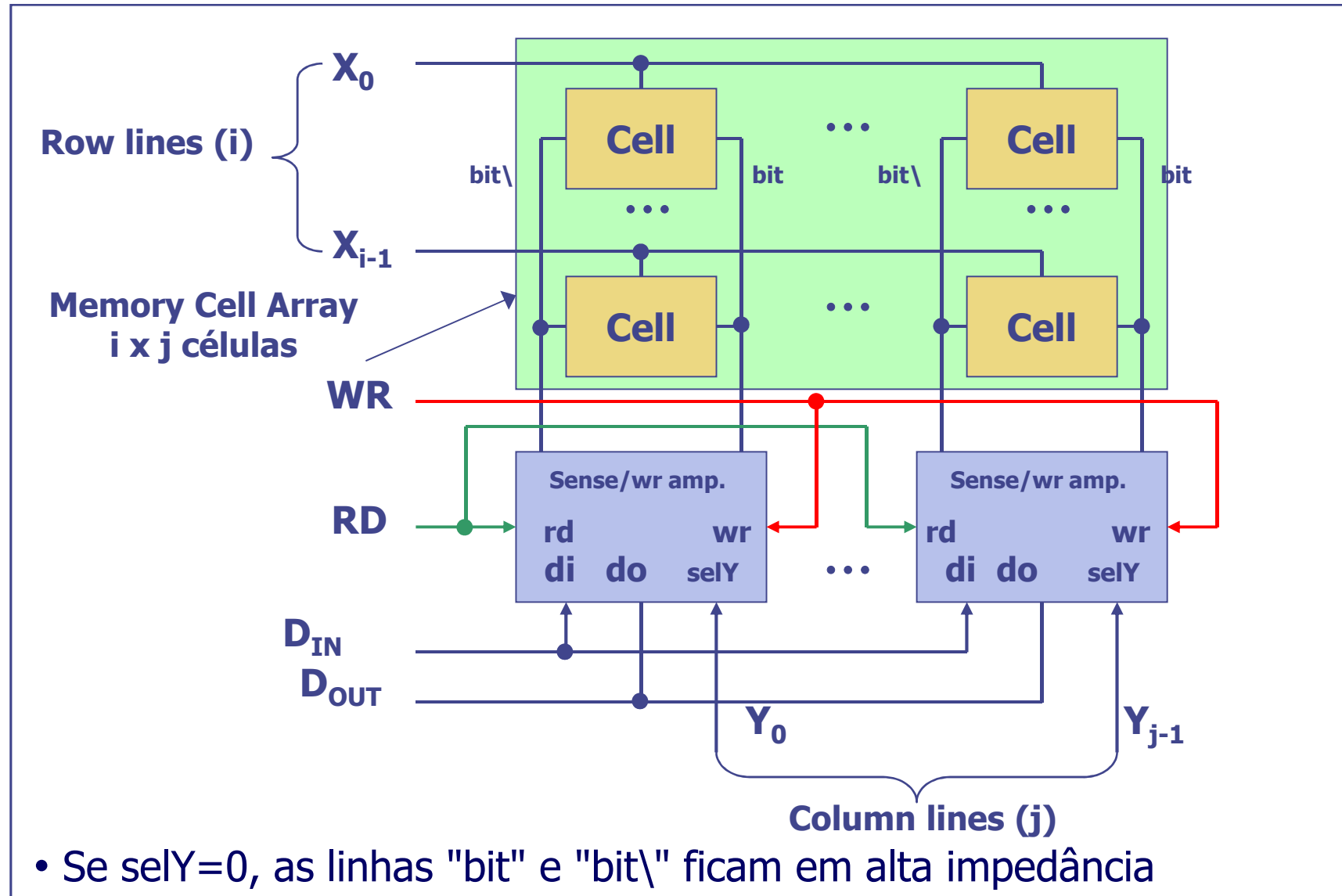
- **Write**

- Colocar a informação em "bit" (e "bit\'\'"). Exemplo: para a escrita do valor lógico "1" – "bit"=1, "bit\'\'=0
- Ativar a linha "select"

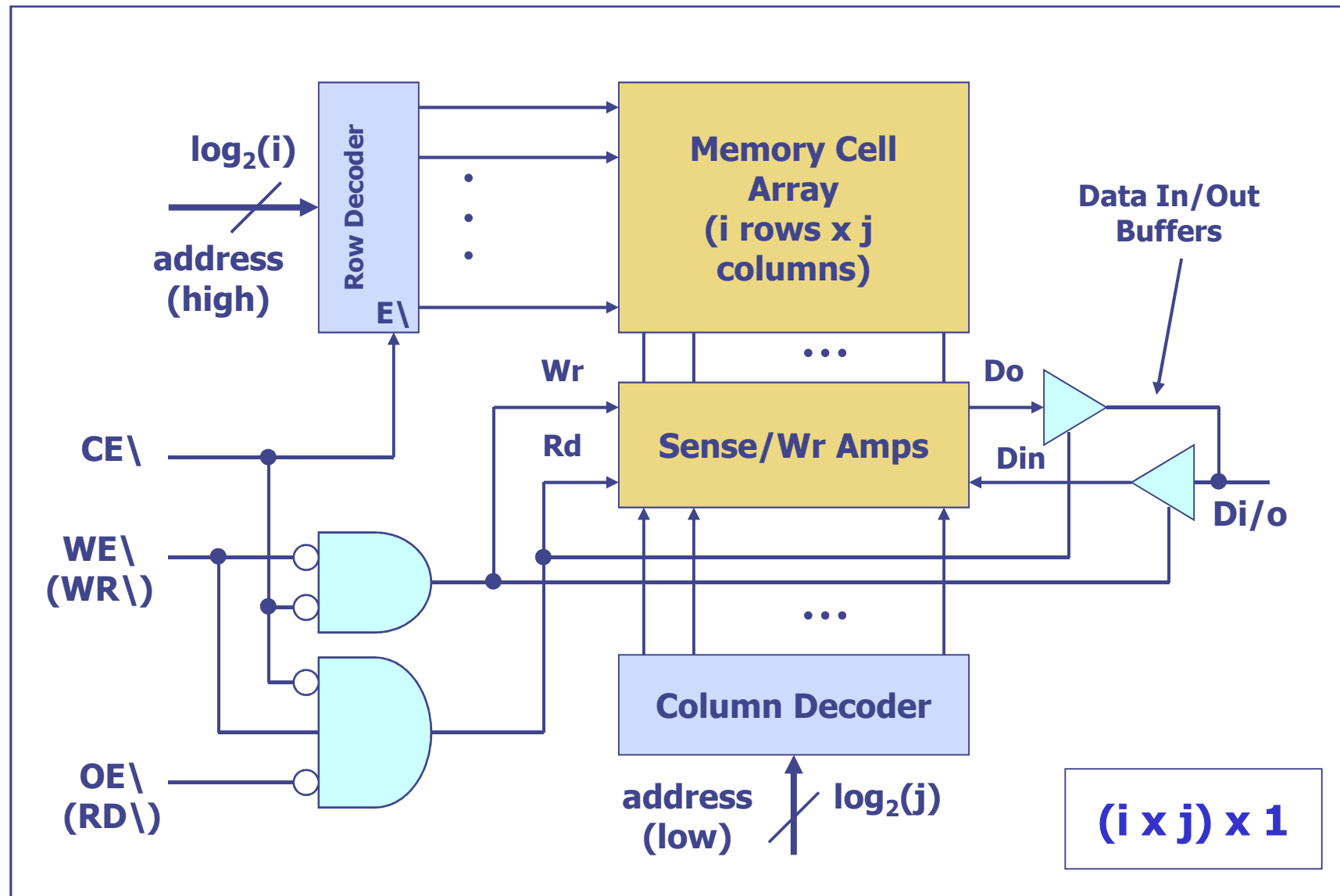
- **Read**

- Ativar a linha "select"
- O valor lógico armazenado na célula é detetado pela diferença de tensão entre as linhas "bit" e "bit\'\'

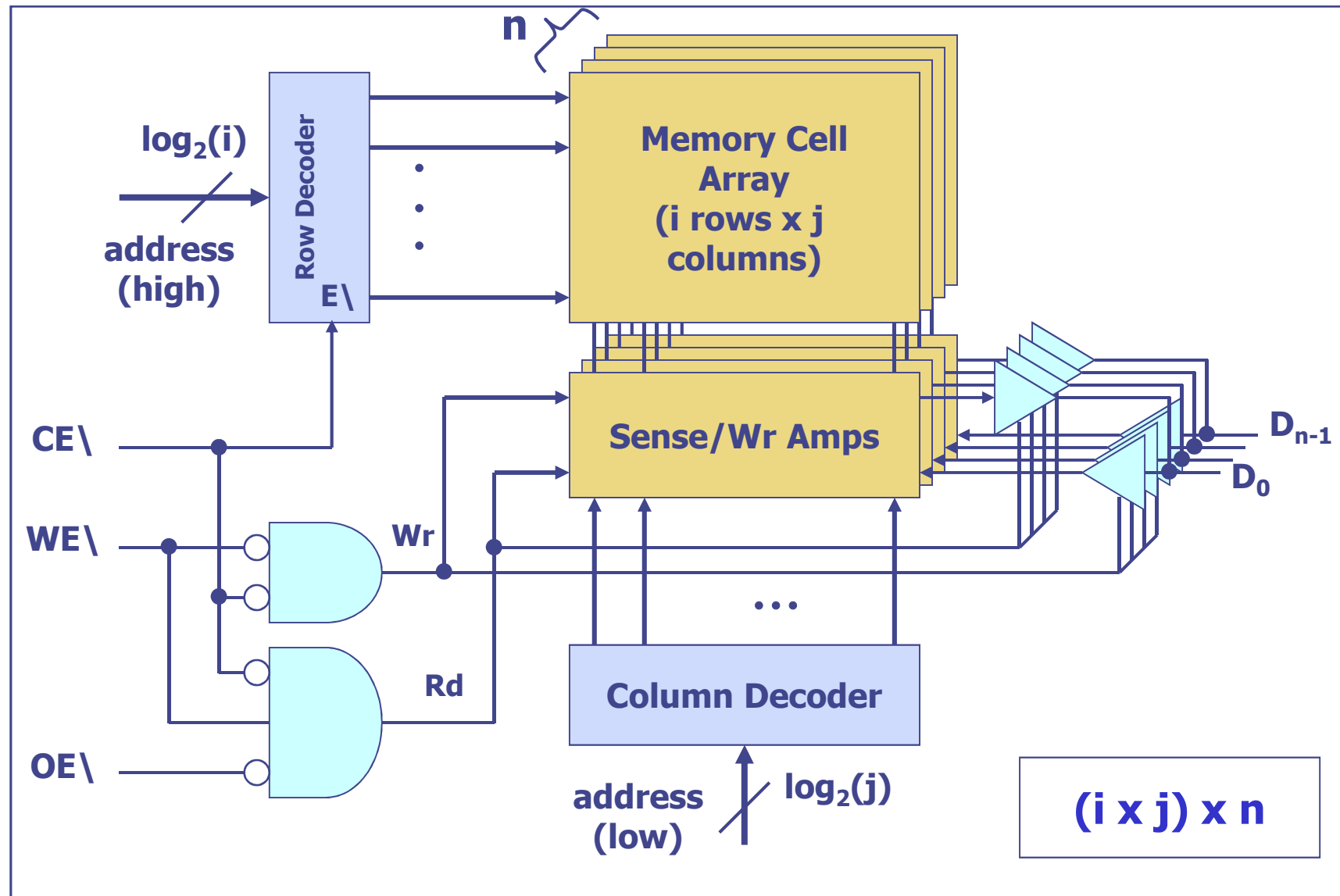
SRAM - Organização interna



SRAM - Organização interna

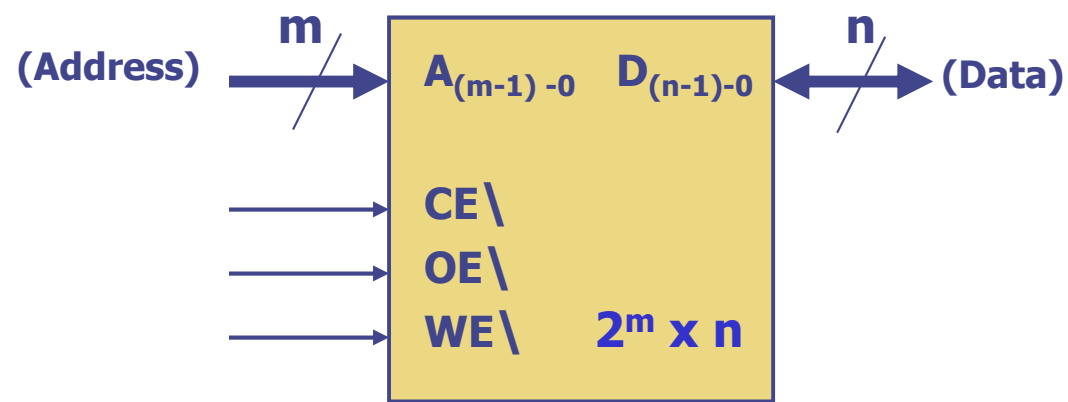


SRAM - Organização interna



SRAM - Bloco funcional

- Diagrama lógico (interface assíncrona)

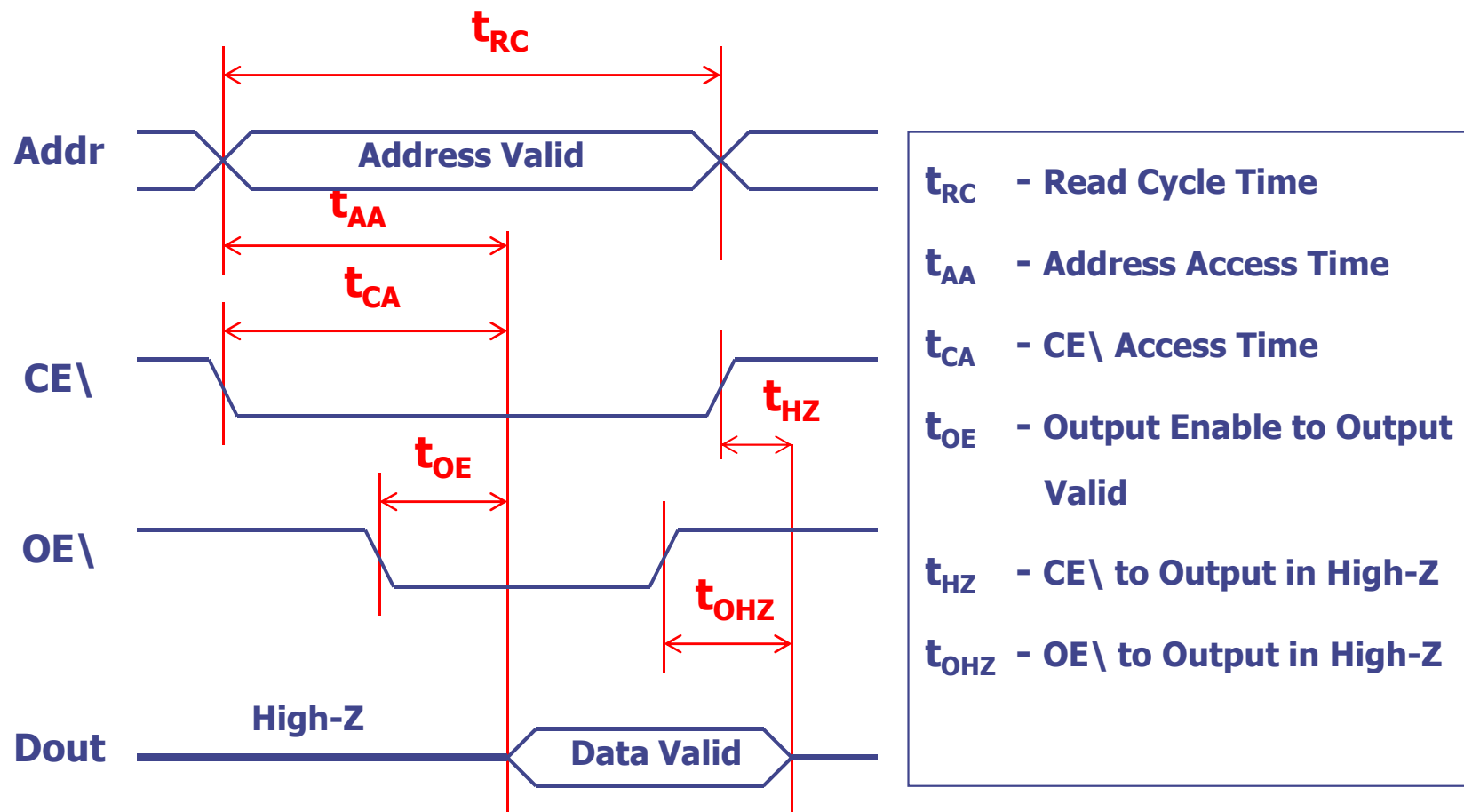


- Tabela de verdade

CE\	OE\	WE\	Operação
1	X	X	High-Z
0	1	1	High-Z
0	X	0	Escrita
0	0	1	Leitura

SRAM – Ciclo de Leitura

- Diagrama temporal típico de um ciclo de leitura de uma memória SRAM (interface assíncrona)



SRAM – Ciclo de leitura

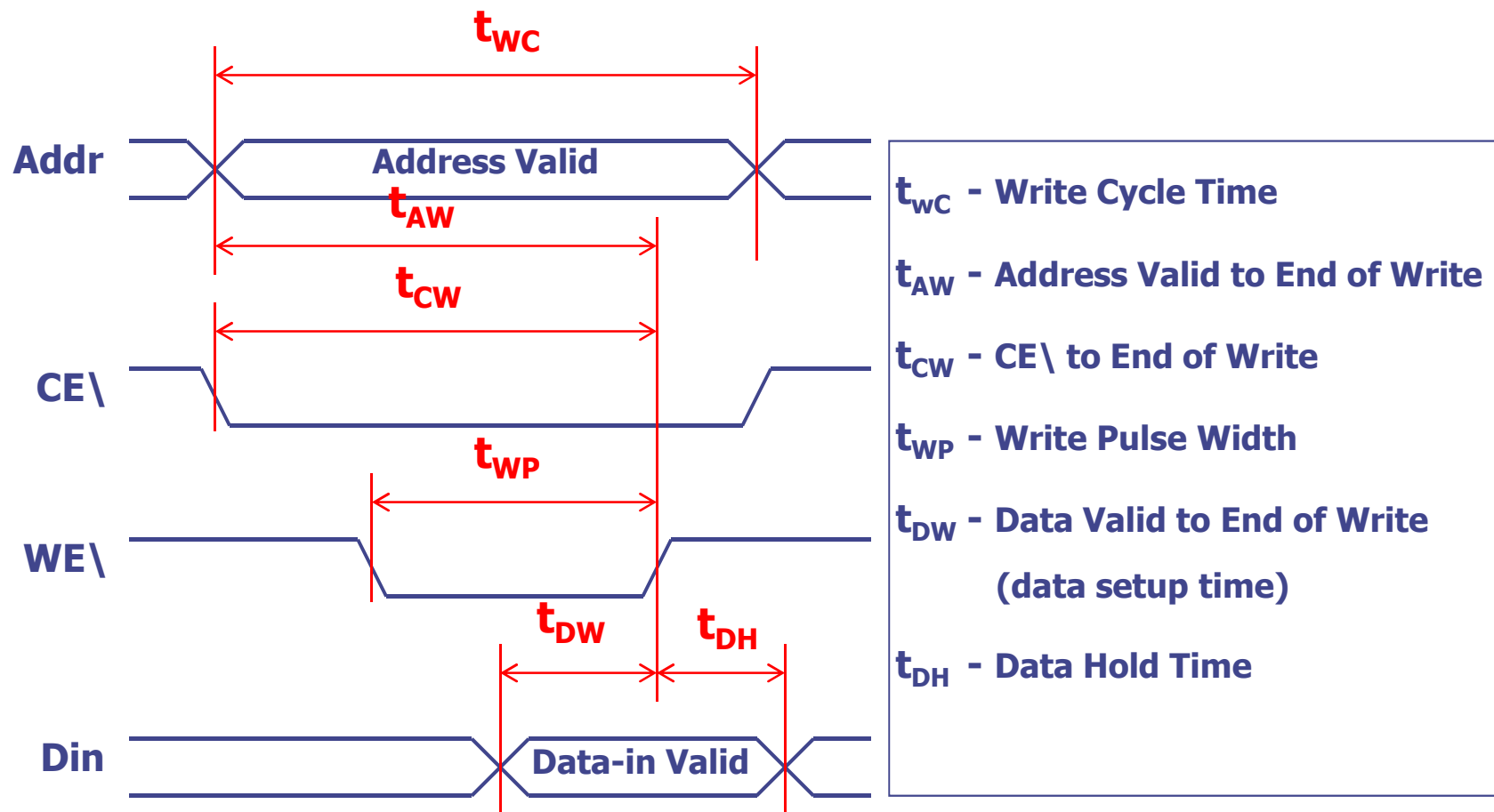
- Valores indicativos (em ns) dos parâmetros associados a um ciclo de leitura de uma memória SRAM:

Parameter	Symbol	Min.	Max.
Read Cycle Time	t_{RC}	1.5	
Address Access Time	t_{AA}		1.5
CE\ Access Time	t_{CA}		1.5
Output Enable to Output Valid	t_{OE}		0.7
CE\ to Output in High-Z	t_{HZ}		0.6
OE\ to Output in High-Z	t_{OHZ}		0.6

- **Cycle Time:** tempo de acesso mais qualquer tempo adicional necessário antes que um segundo acesso possa ter início
- **Access Time:** tempo necessário para os dados ficarem disponíveis no barramento de saída da memória
- **Taxa de transferência:** taxa a que os dados podem ser transferidos de/para uma memória ($1 / \text{cycle_time}$)

SRAM – Ciclo de Escrita

- Diagrama temporal típico de um ciclo de escrita de uma memória SRAM



SRAM – Ciclo de Escrita

- Valores indicativos (em ns) dos parâmetros associados a um ciclo de escrita de uma memória SRAM:

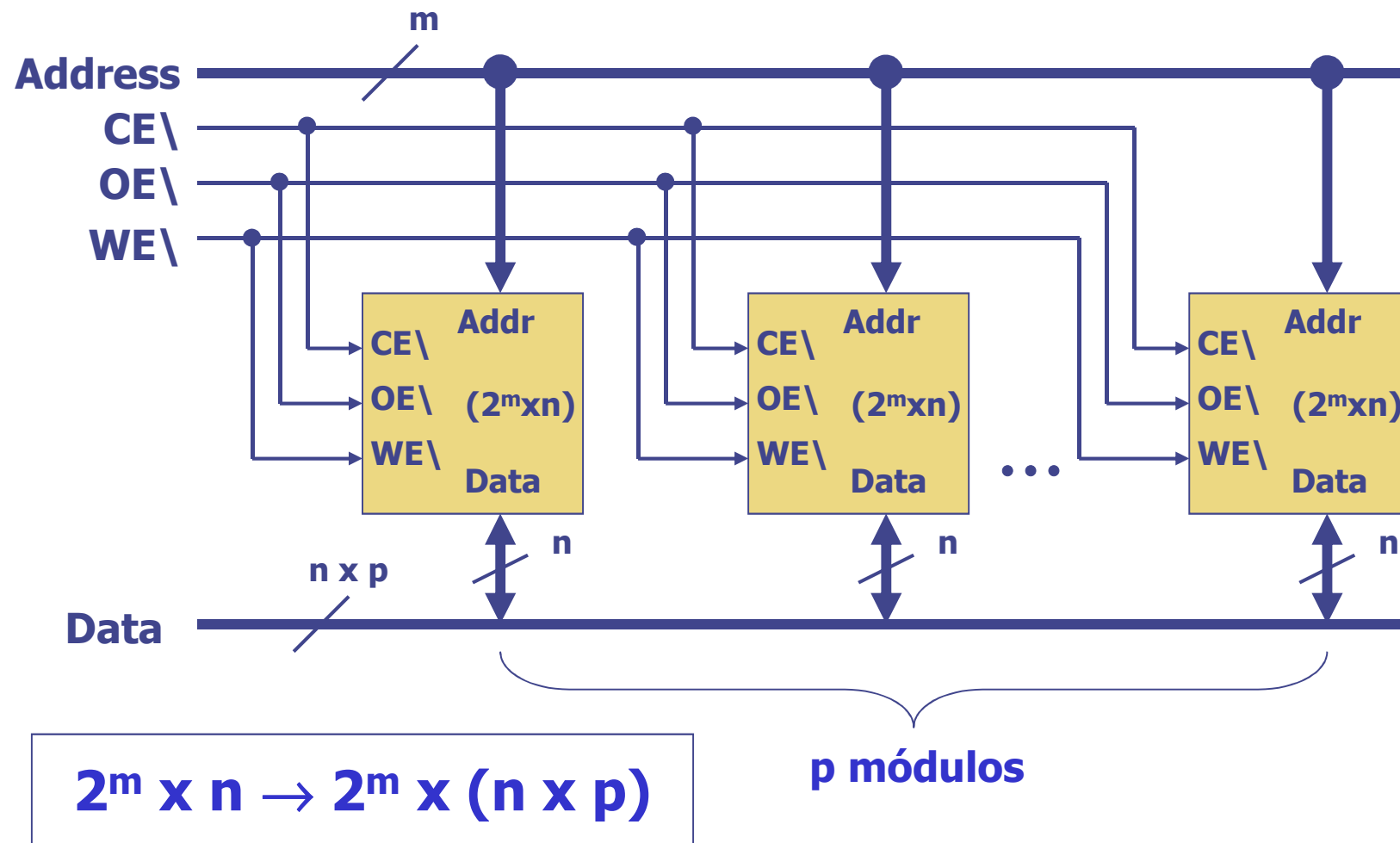
Parameter	Symbol	Min.	Max.
Write Cycle Time	t_{WC}	1.5	
Address Valid to End of Write	t_{AW}	1.0	
CE\ to End of Write	t_{CW}	1.0	
Write Pulse Width	t_{WP}	1.0	
Data Valid to End of Write	t_{DW}	0.7	
Data Hold Time	t_{DH}	0	

Aumento da capacidade de armazenamento

- É frequente ter-se necessidade de memórias com uma capacidade de armazenamento superior à capacidade individual dos circuitos disponíveis comercialmente
- Nessa situação recorre-se à construção de módulos de memória que resultam do agrupamento de circuitos de acordo com o aumento pretendido
- Assim, a construção de um módulo de memória pode envolver as duas fases seguintes, ou apenas uma delas, em função dos circuitos disponíveis e dos requisitos finais de armazenamento:
 - **Aumento do comprimento de palavra.** Exemplo: a partir de C.I.s de 32Kx1, construir uma memória de 32Kx8
 - **Aumento do número total de posições de memória.** Exemplo: a partir de C.I.s de 32Kx8, construir uma memória de 256Kx8

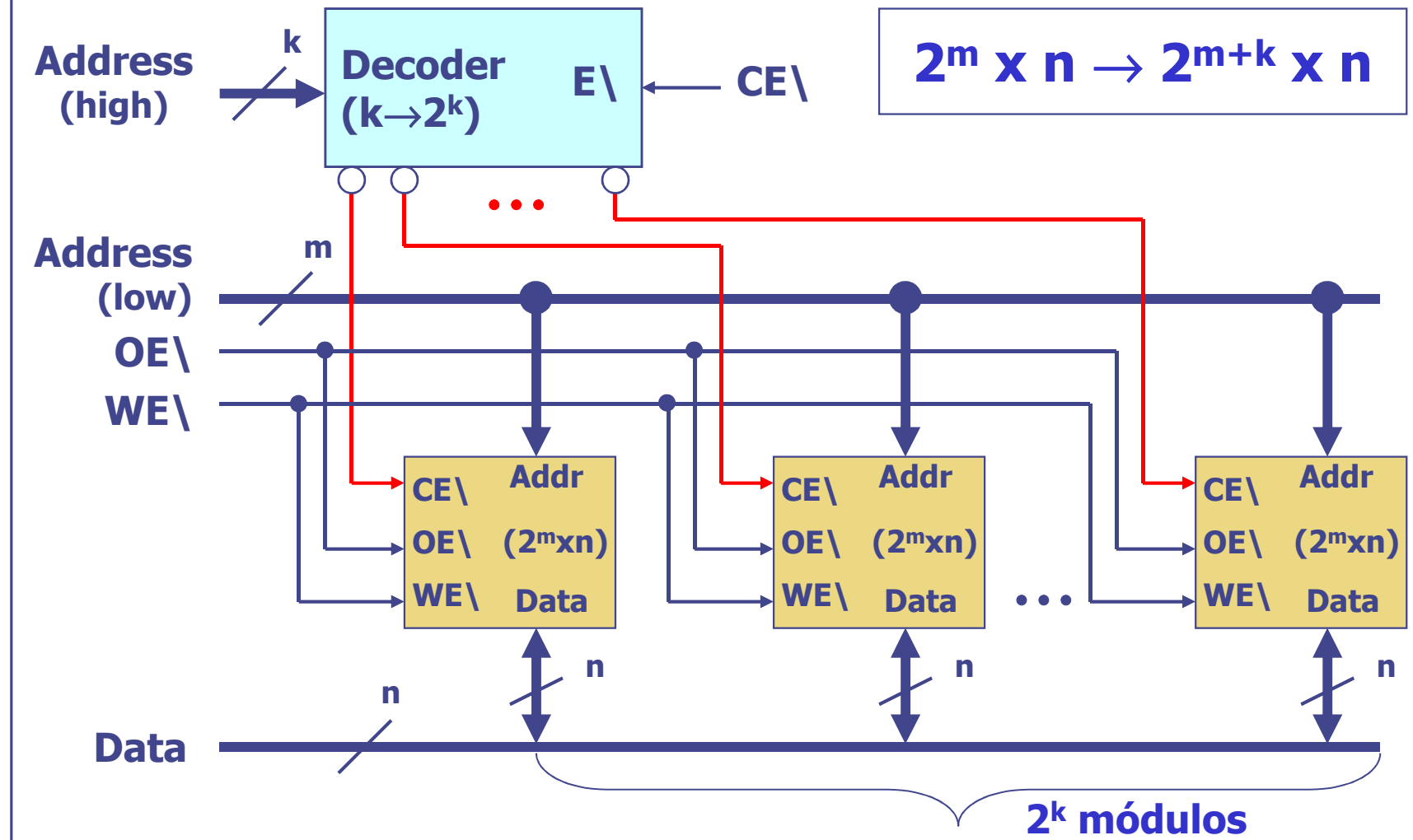
Módulo de memória SRAM

- Aumento do comprimento de palavra

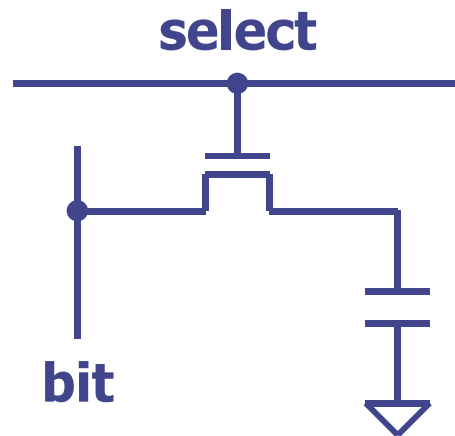


Módulo de memória SRAM

- Aumento do número total de posições de memória

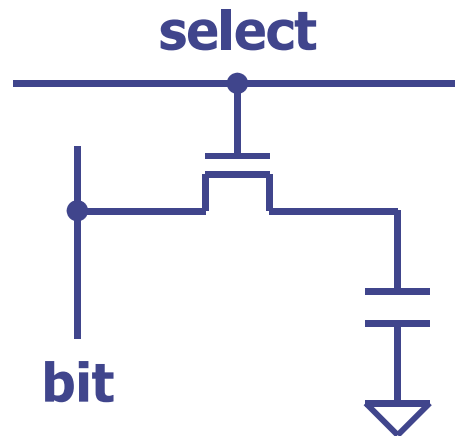


RAM Dinâmica (DRAM)



- Condensador com uma capacidade muito pequena (dezenas de fF ($1 \text{ fF} = 10^{-15} \text{ F}$))
- A operação de leitura é destrutiva (descarrega o condensador)
- Na ausência de leitura, o condensador descarrega "lentamente"
- Informação permanece na célula apenas durante alguns mili-segundos
- Obrigatório fazer refrescamento ("refresh") periódico da carga do condensador

RAM Dinâmica (DRAM)



- **Write**

- Colocar dado na linha "bit"
- Ativar a linha "select"

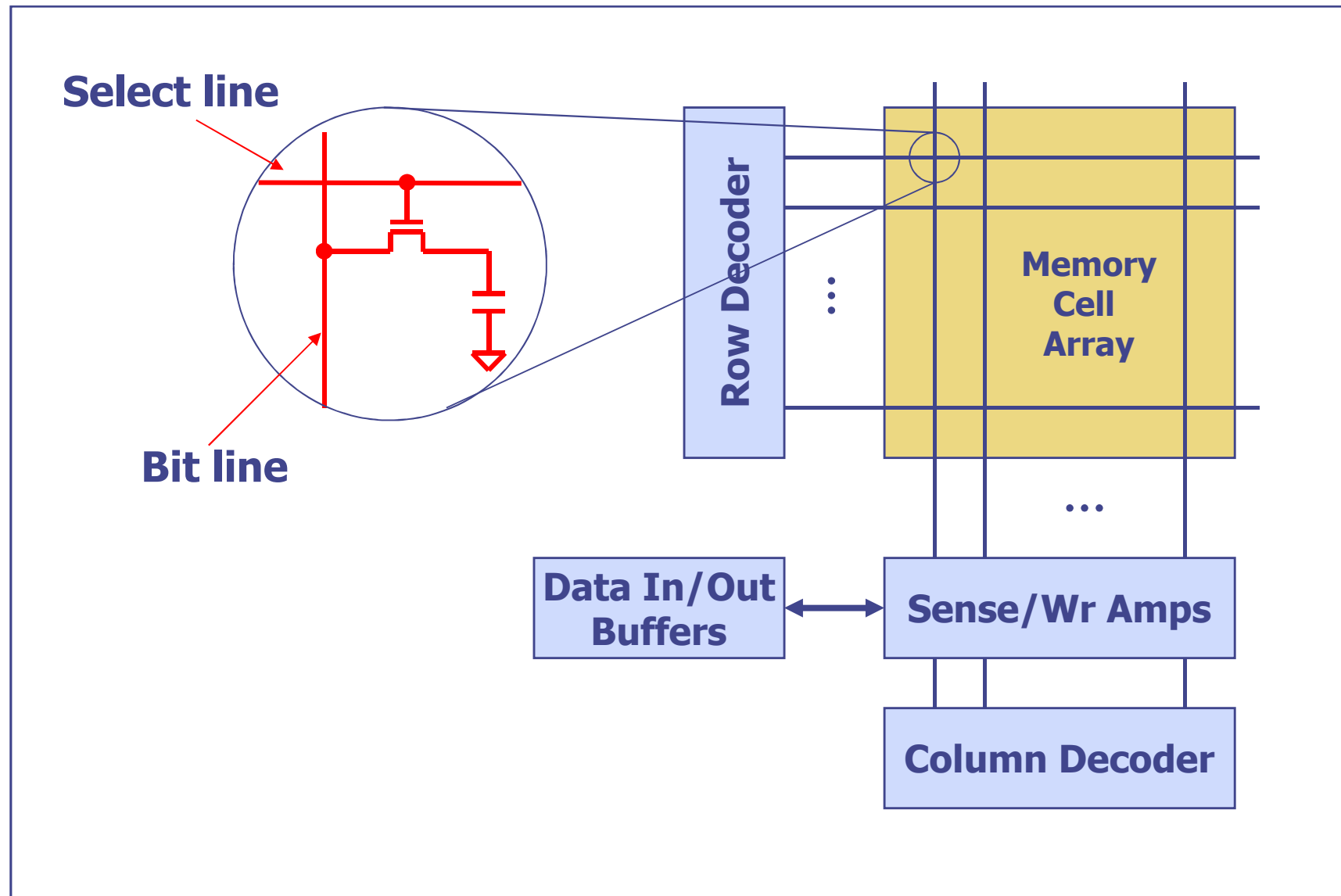
- **Read**

- Pre-carregar a linha "bit" a $VDD/2$
- Ativar a linha "select"
- Valor lógico detetado pela diferença de tensão na linha bit (rel. a $VDD/2$)
- Restauro do valor da tensão no condensador (write)

- **Refresh da célula**

- Operação interna idêntica a uma operação de "Read"

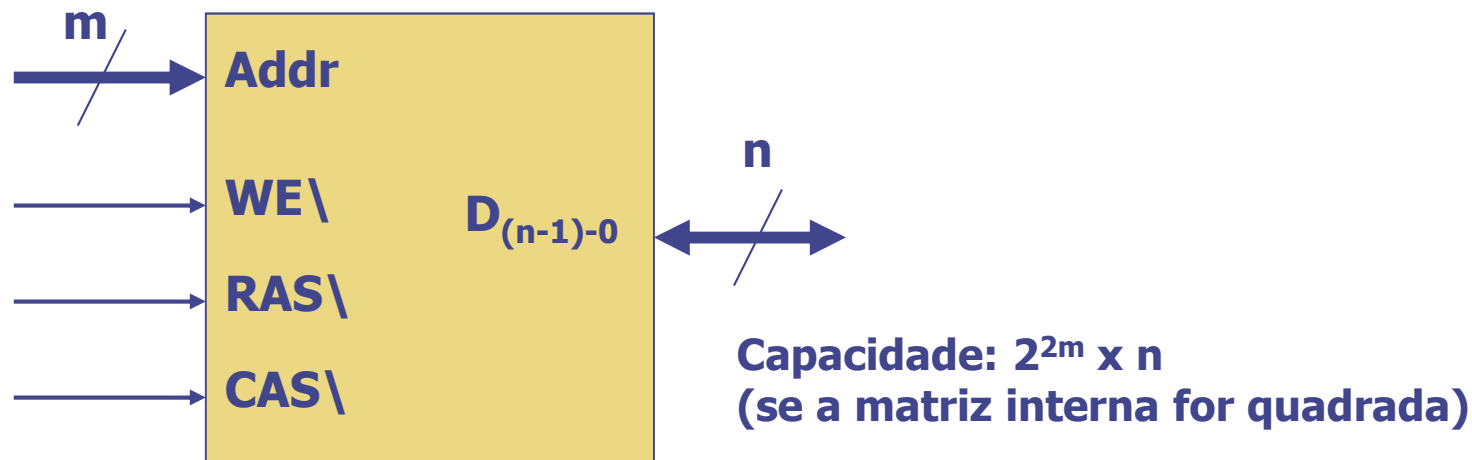
RAM Dinâmica (DRAM)



RAM Dinâmica (DRAM)

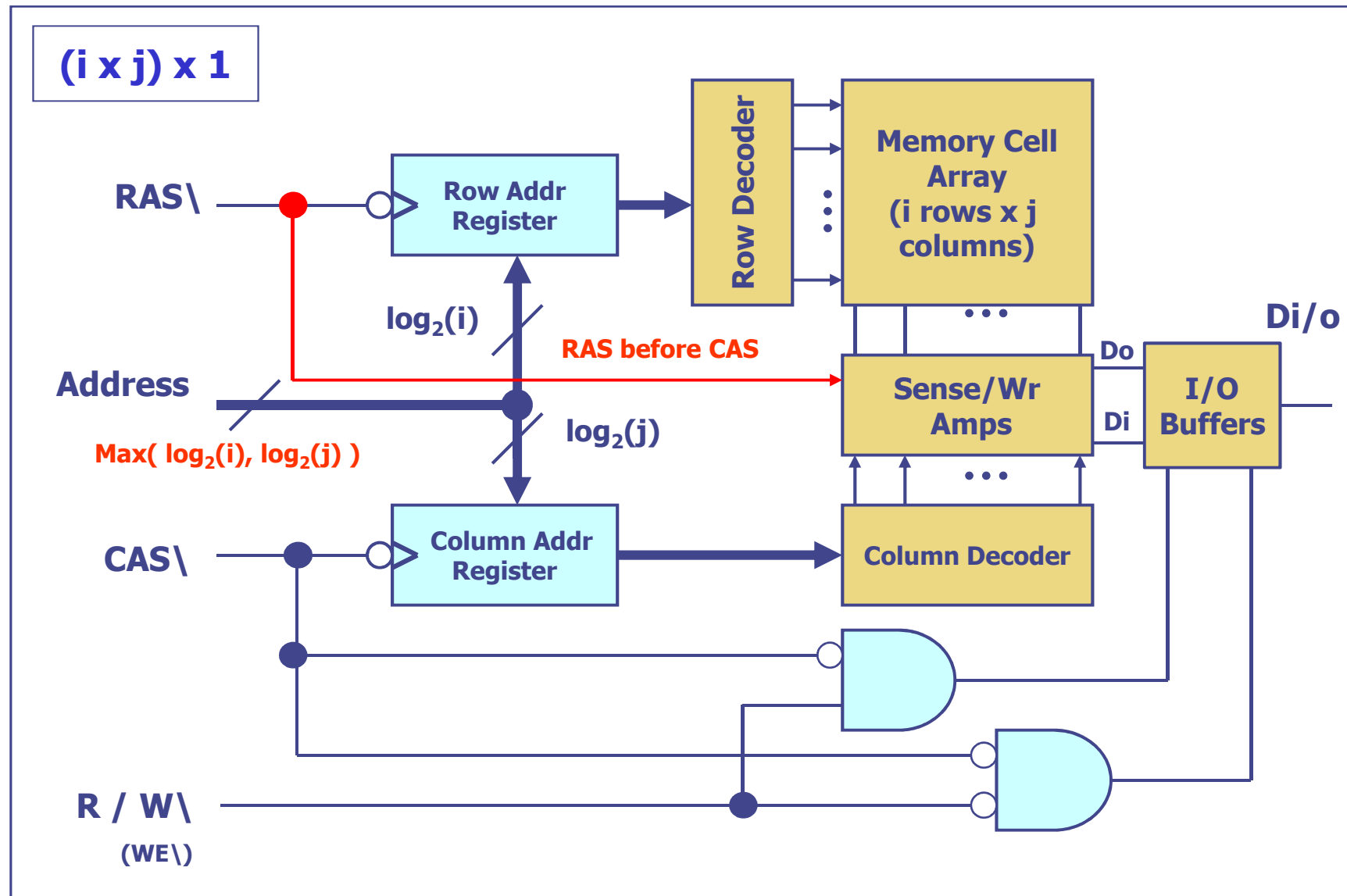
- Organização em matriz
- **Endereços** de linha e coluna **multiplexados no tempo**
- Multiplexagem no tempo obriga à utilização de 2 sinais adicionais (multiplexagem com 2 strobes independentes)
 - **RAS** – Row Address Strobe
 - **CAS** – Column Address Strobe
- Linha CAS funciona também como "chip-select"
- RAS e CAS, sensíveis à transição

DRAM - Diagrama lógico



- $WE\backslash = 0 \rightarrow$ escrita; $WE\backslash = 1 \rightarrow$ leitura ($\equiv R/W\backslash$)
- $RAS\backslash$: valida endereço da linha na transição descendente
- $CAS\backslash$: valida endereço da coluna na transição descendente

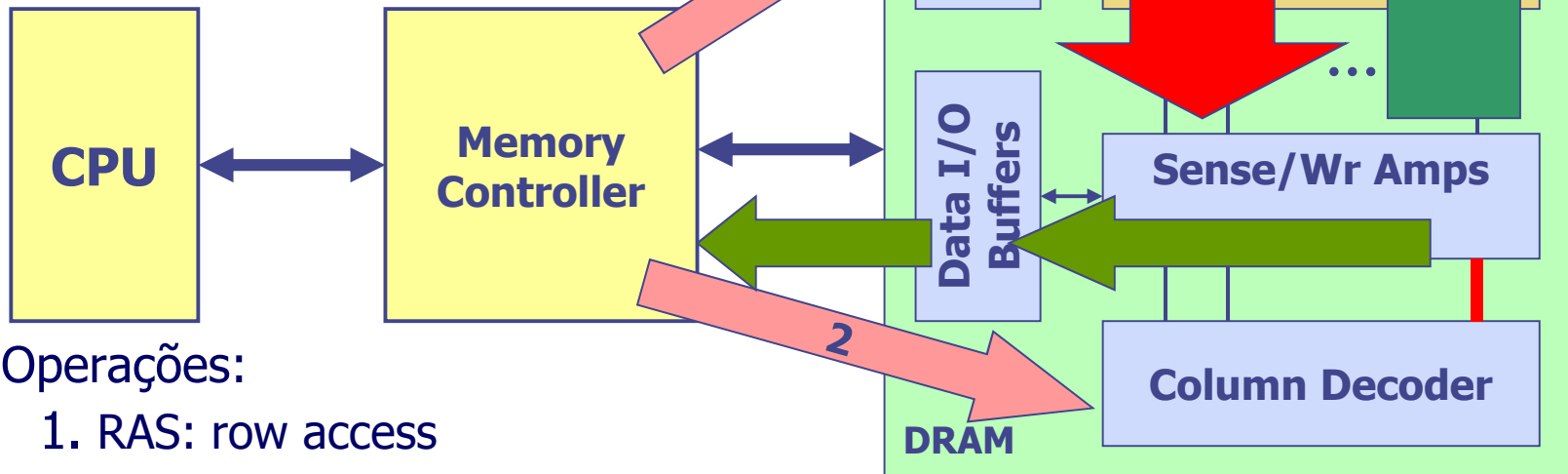
DRAM – Diagrama de blocos conceptual



DRAM – Leitura

- Memory controller:

- gera os sinais de interface com a memória e separa os endereços em linha e coluna
- gera os sinais RAS e CAS
- executa, periodicamente, as operações de *refresh*



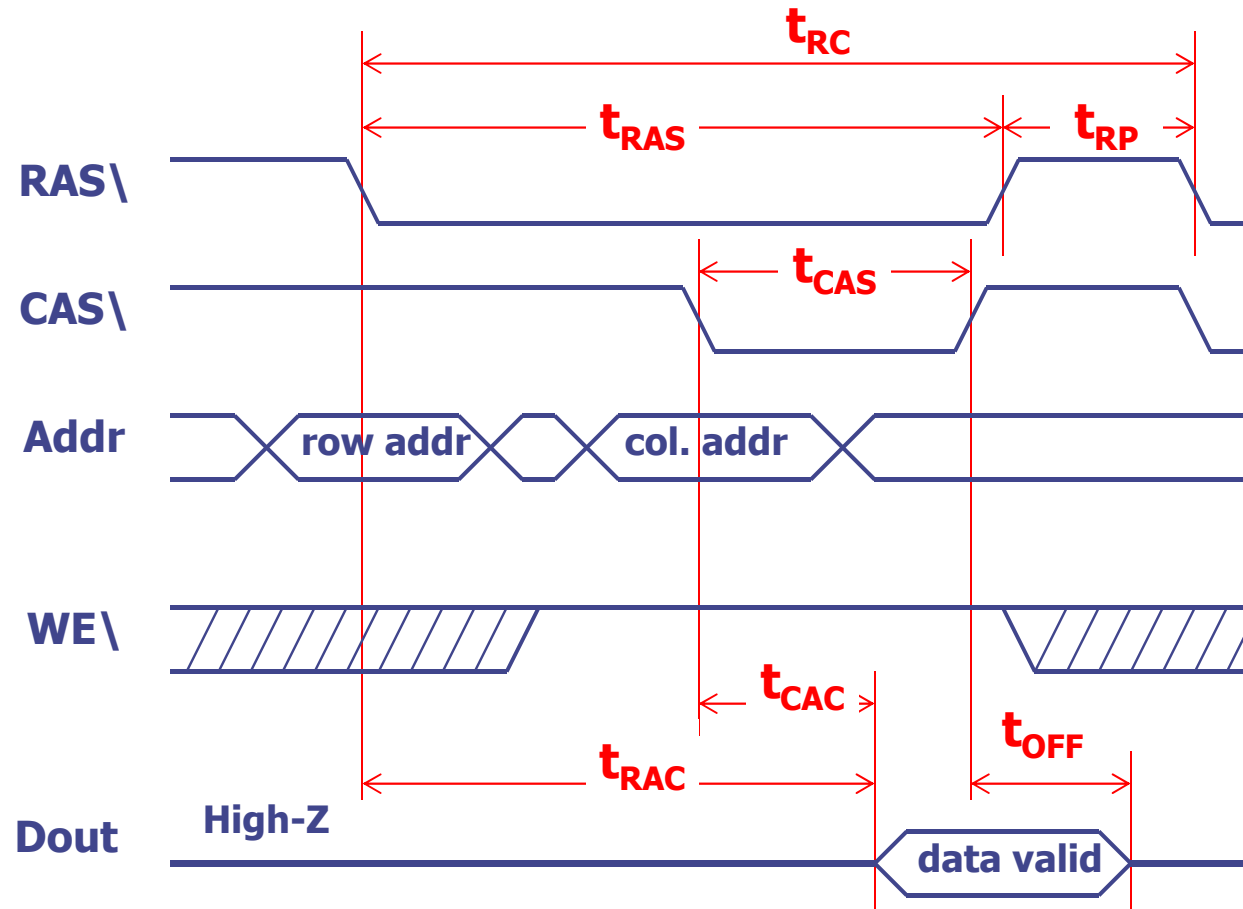
- Operações:

1. RAS: row access
2. CAS: column access

- Buffer de linha (*row buffer*) armazena temporariamente todos os bits de uma linha de células da matriz

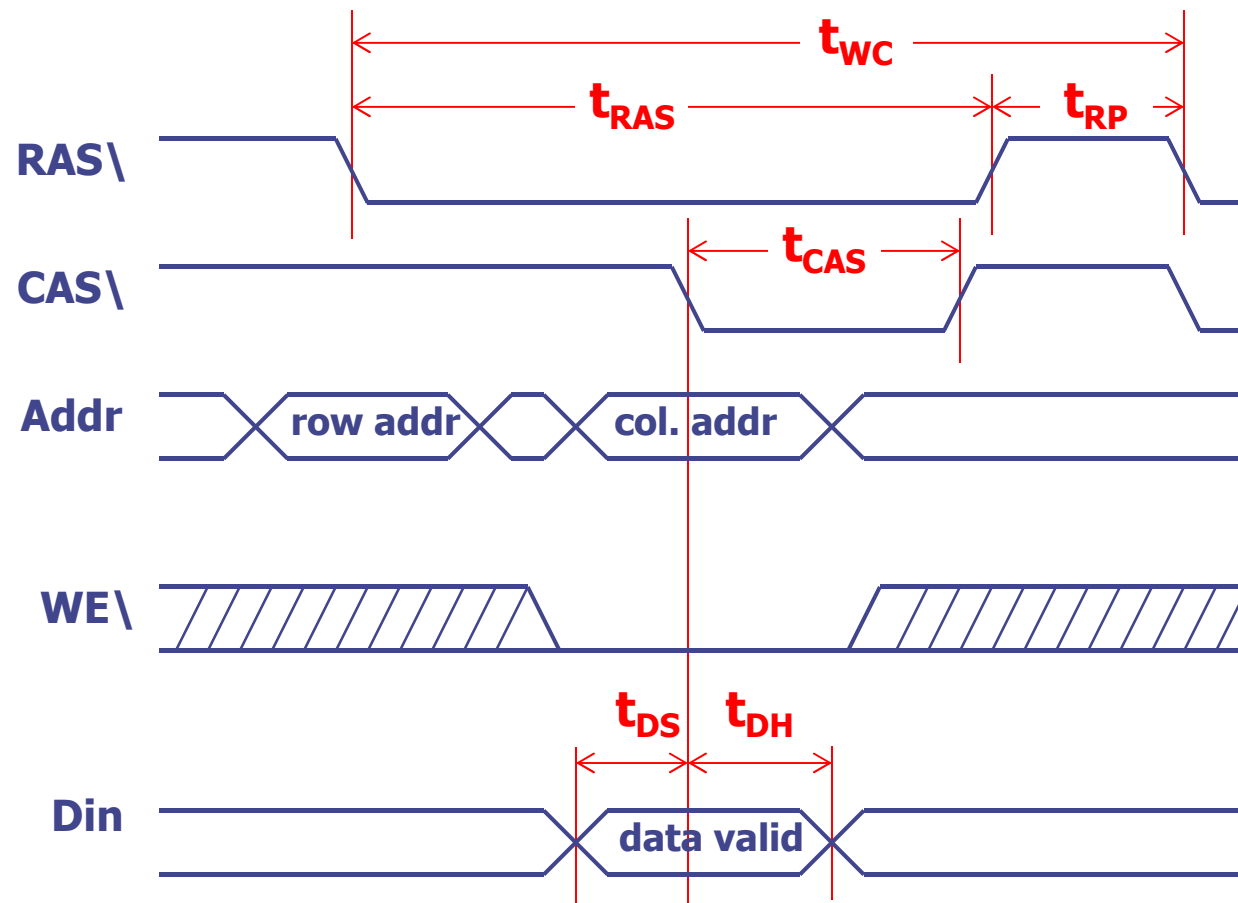
DRAM – Ciclo de Leitura

- Diagrama temporal típico de um ciclo de leitura de uma memória DRAM



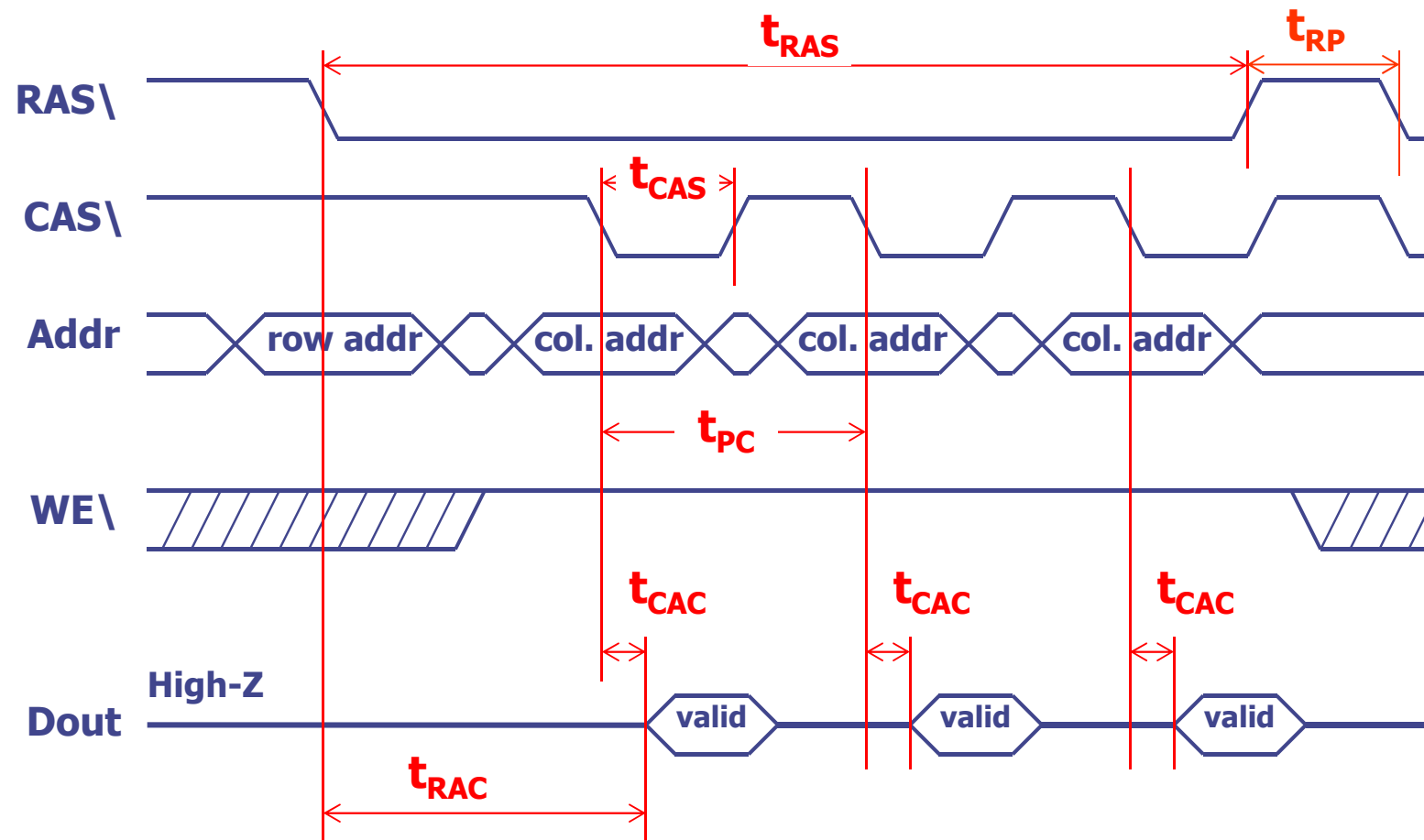
DRAM – Ciclo de Escrita

- Diagrama temporal típico de um ciclo de escrita (*early write*) de uma memória DRAM

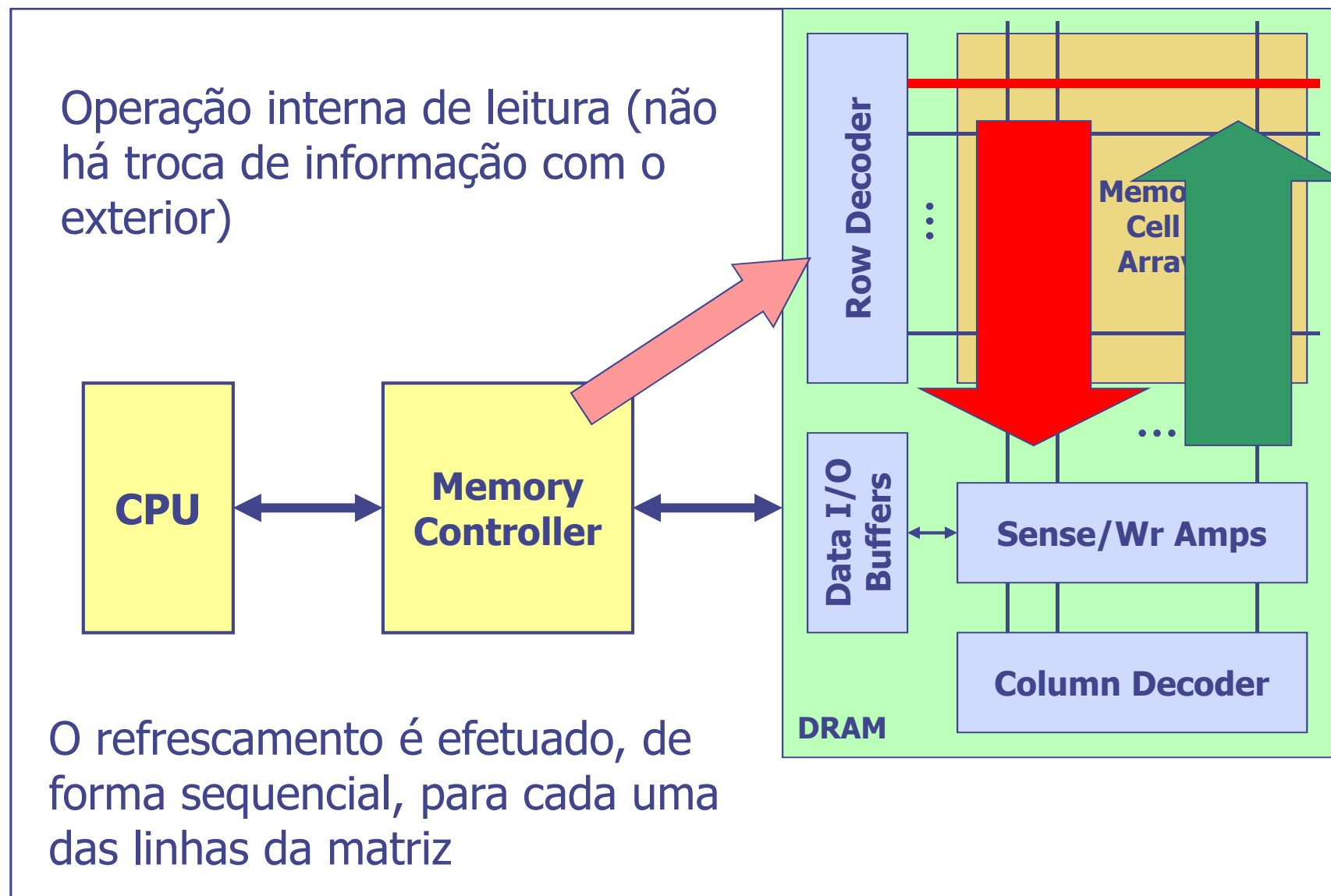


DRAM – Ciclo de Leitura em *page mode*

- Diagrama temporal típico de um ciclo de leitura de uma memória DRAM, em modo paginado (*page mode*)

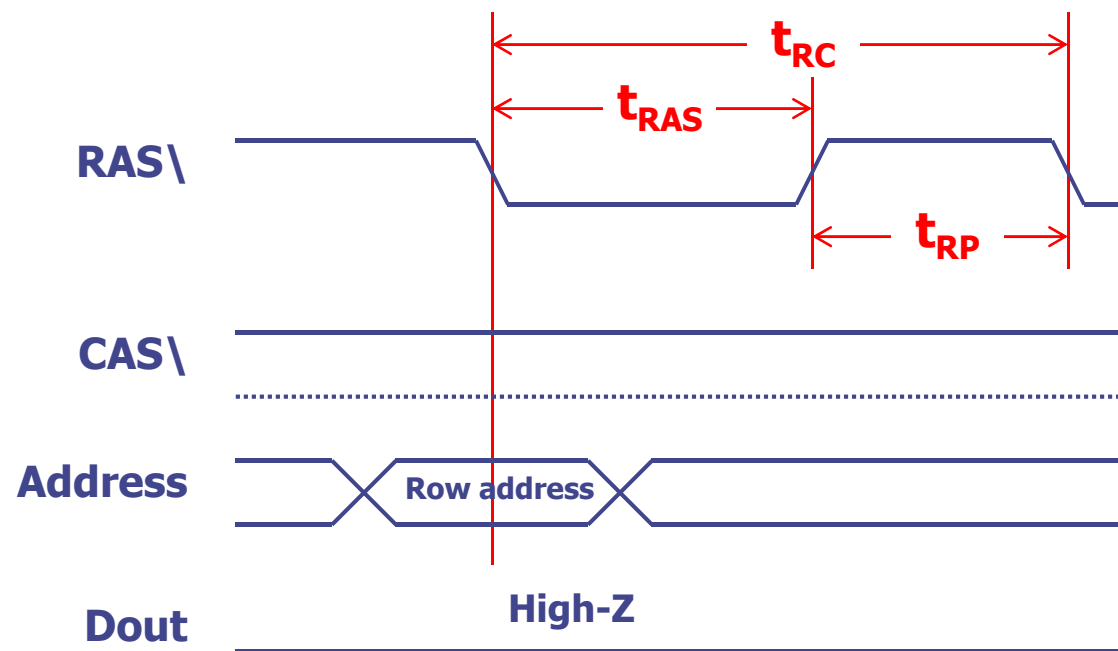


DRAM – Refrescamento



DRAM Refresh – RAS Only

- O *refresh* é feito simultaneamente em **todas as células da mesma linha da matriz** (especificada no address bus, no momento da ativação do sinal RAS\)
- O sinal CAS\ mantém-se inativo durante o processo



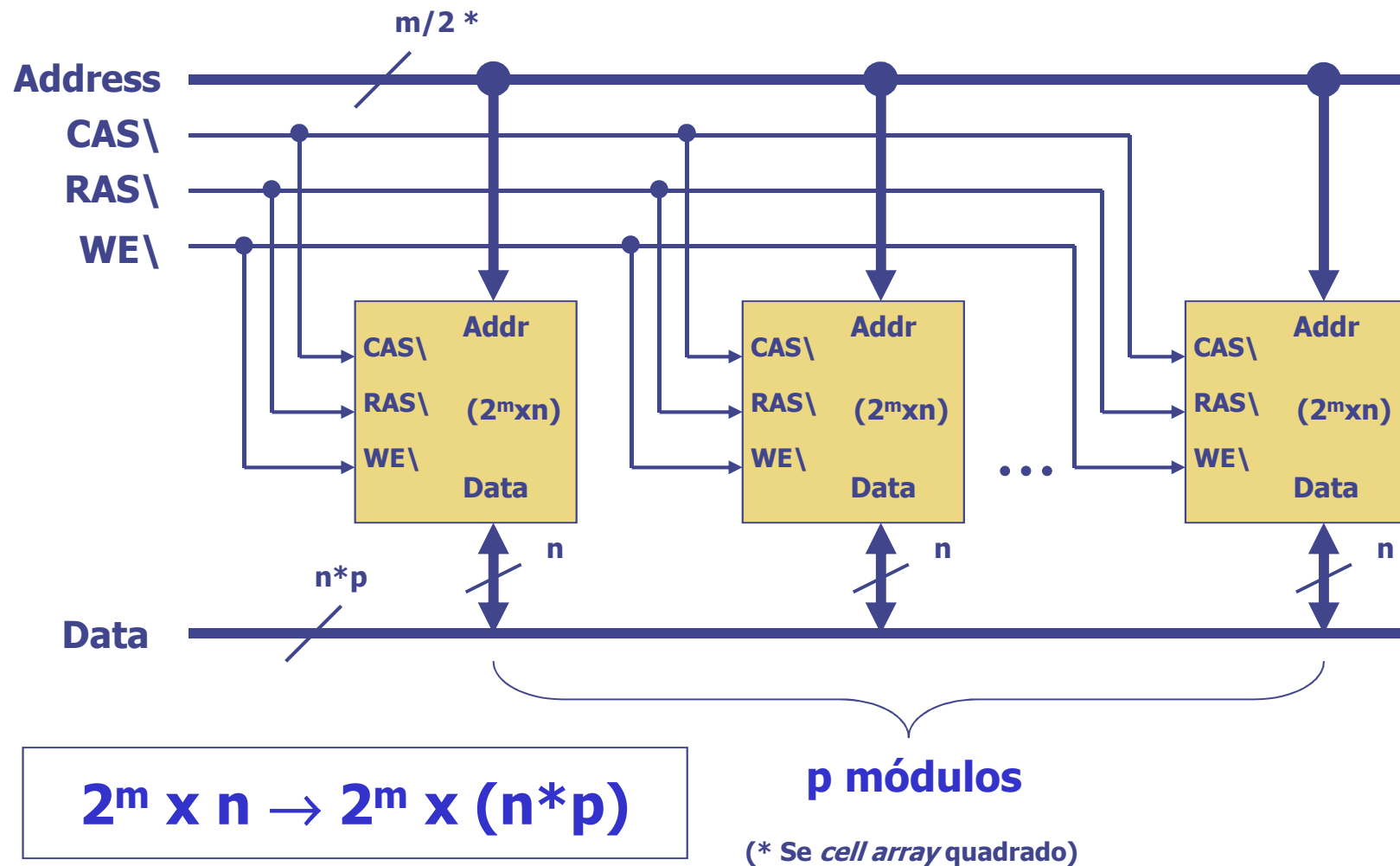
DRAM - Parâmetros principais

- Valores indicativos (em ns) dos parâmetros indicados nos diagramas temporais de leitura e escrita de uma memória DRAM com um tempo de acesso de 55 ns:

Parameter	Symbol	Min.	Max.
Read or Write Cycle Time	t_{RC}	100	
RAS\ precharge time	t_{RP}	45	
Page mode cycle time	t_{PC}	35	
RAS\ pulse width	t_{RAS}	55	10000
CAS\ pulse width	t_{CAS}	28	10000
Data-in setup time	t_{DS}	5	
Data-in hold time	t_{DH}	14	
Output buffer turn-off delay	t_{OFF}		15
Access time from RAS\	t_{RAC}		55
Access time from CAS\	t_{CAC}		28

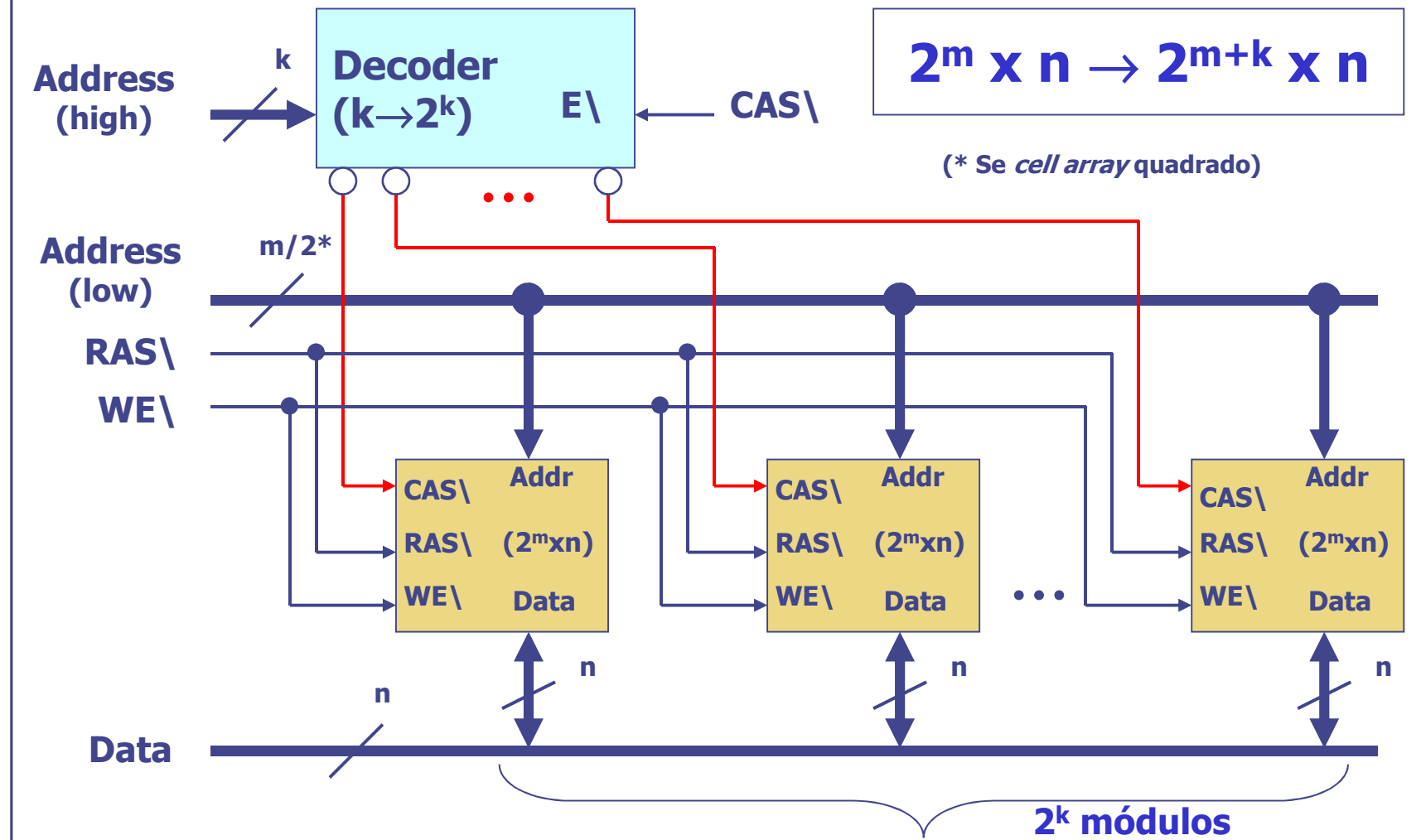
Módulo de memória DRAM

- Aumento do comprimento de palavra



Módulo de memória DRAM

- Aumento do número total de posições de memória



Melhorias de desempenho da DRAM

- **Fast Page Mode**

- Adiciona sinais de temporização que permitem acessos repetidos ao buffer de linha (sem outro tempo de acesso à linha)

- **Synchronous DRAM (SDRAM)**

- Adiciona um sinal de relógio à interface DRAM, para facilitar a sincronização de transferências múltiplas
- Múltiplos bancos, cada um com o seu buffer de linha

- **Double Data Rate (DDR SDRAM)**

- Transferência de dados tanto no flanco ascendente como no flanco descendente do sinal de relógio (duplica a taxa de transferência de pico)
- Versão atual: DDR4 (set/2014). Exemplo: DDR4-3200, 3200 Milhões de transferências por segundo, relógio de 1.6 GHz

- Estas técnicas melhoram a largura de banda, mas não a latência