

Memória Interna (I parte)

Organização e Arquitetura de Computadores.

Professora Danielle Morais de Andrade.

Grupo: João Victor Silva Jucá,

Luís Felipe Melo

Paulo Domingos de S Neto.

Como era antes?



Em computadores mais antigos, a maneira com a qual memória principal era gravada de modo aleatório um array de anéis ferromagnéticos em forma de rosquinha, conhecidos como cores.

Por isso, a memória principal foi frequentemente referida como core, um termo que persiste até hoje. Com o advento da microeletrônica, e suas vantagens, o uso da memória de core magnética foi então superada. Hoje em dia, o uso de chips semicondutores para memória principal é quase universal.

Organização



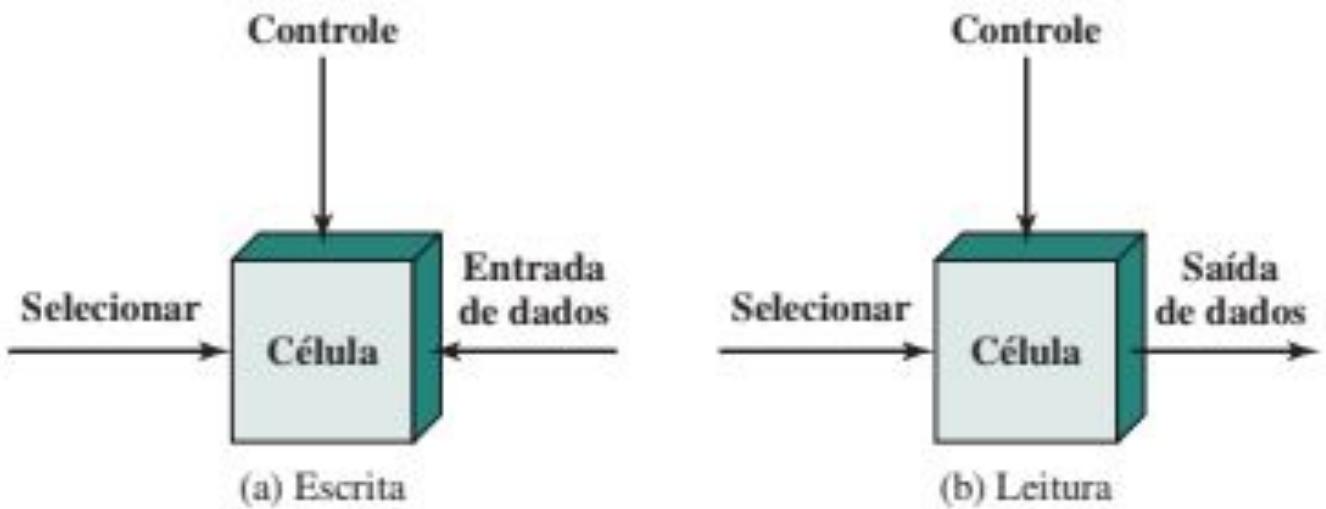
- Célula de Memória –

Propriedades: Apresentam dois estados estáveis (ou semiestáveis), que podem ser usados para representar o binário 1 e 0.

São capazes de serem gravadas (pelo menos uma vez), para definir o estado.

São capazes de serem lidas, para verificar o estado.

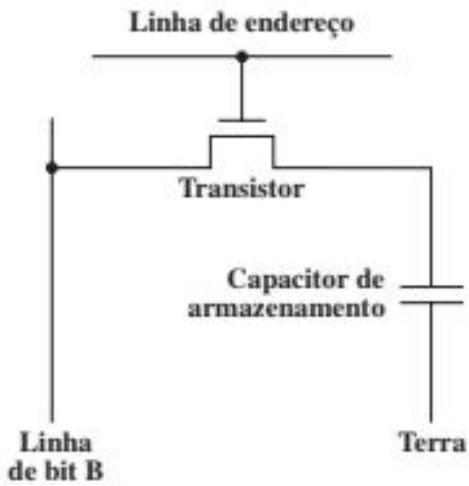
Organização



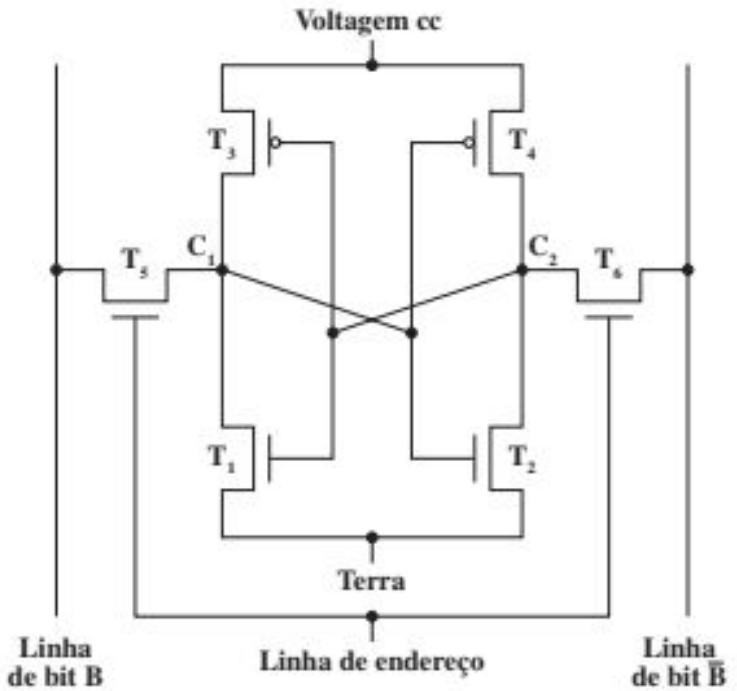
Tipos de Memória

Tipo de memória	Categoria	Apagamento	Mecanismo de gravação	Volatilidade
Memória de acesso aleatório (RAM)	Memória de leitura-gravação	Eletricamente, em nível de byte	Eletricamente	Volátil
Memória somente de leitura (ROM)	Memória somente de leitura	Não é possível	Máscaras	Não volátil
ROM programável (PROM — do inglês, <i>Programmable ROM</i>)			Eletricamente	
PROM apagável (EPROM — do inglês, <i>Erasable PROM</i>)	Memória principalmente de leitura	Luz UV, Em nível de chip	Não volátil	
PROM eletricamente apagável (EEPROM — do inglês, <i>Electrically Erasable PROM</i>)		Eletricamente, em nível de byte		
Memória flash		Eletricamente, em nível de bloco		

DRAM e RAM

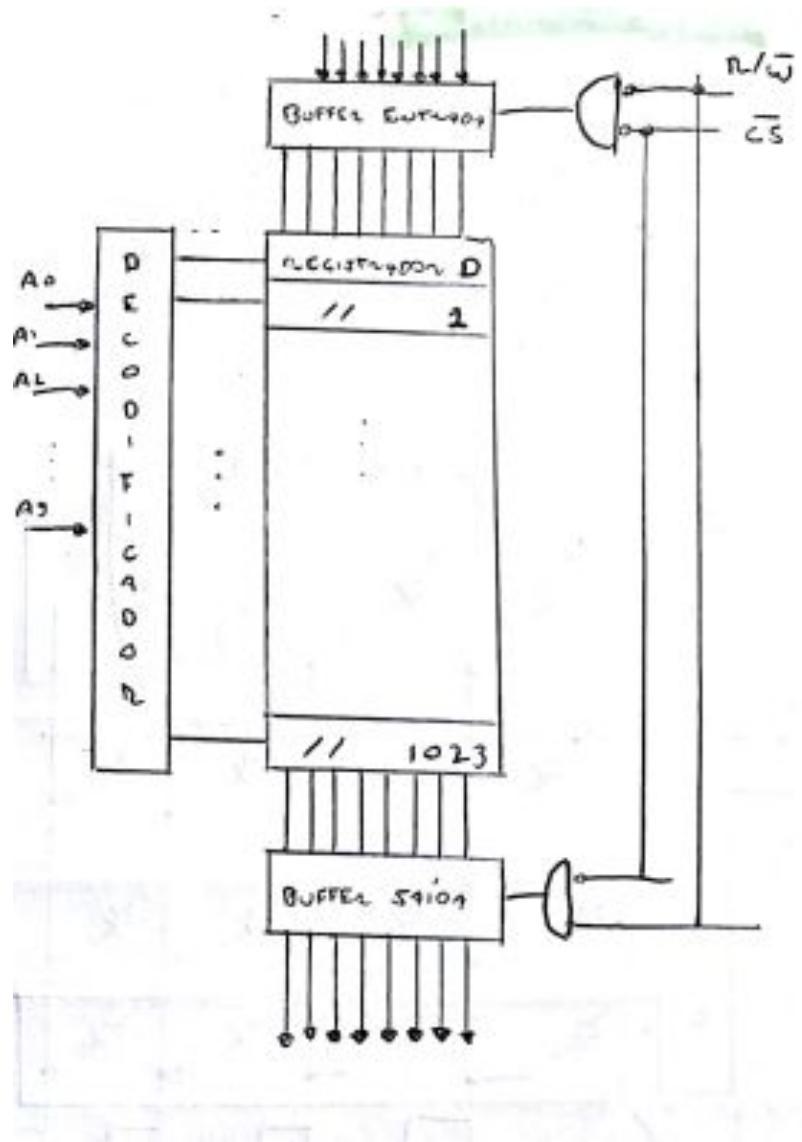


(a) Célula de RAM dinâmica (DRAM)

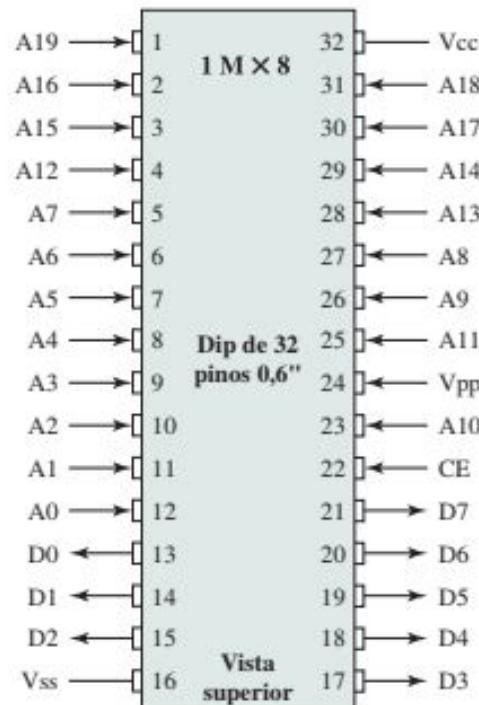


(b) Célula de RAM estática (SRAM)

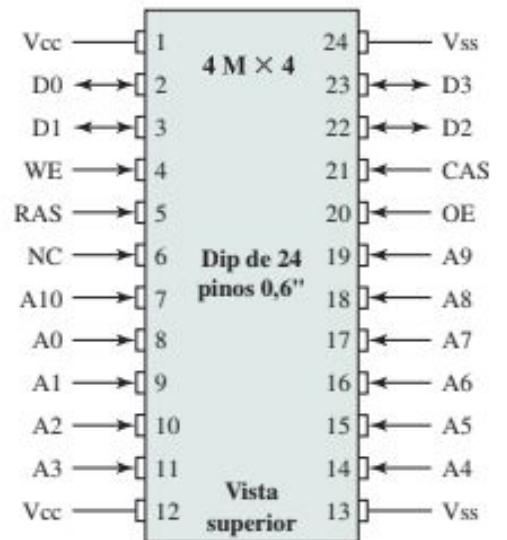
Lógica do CHIP



Encapsulamento do Chip



(a) EPROM de 8 Mbits



(b) DRAM de 16 Mbits

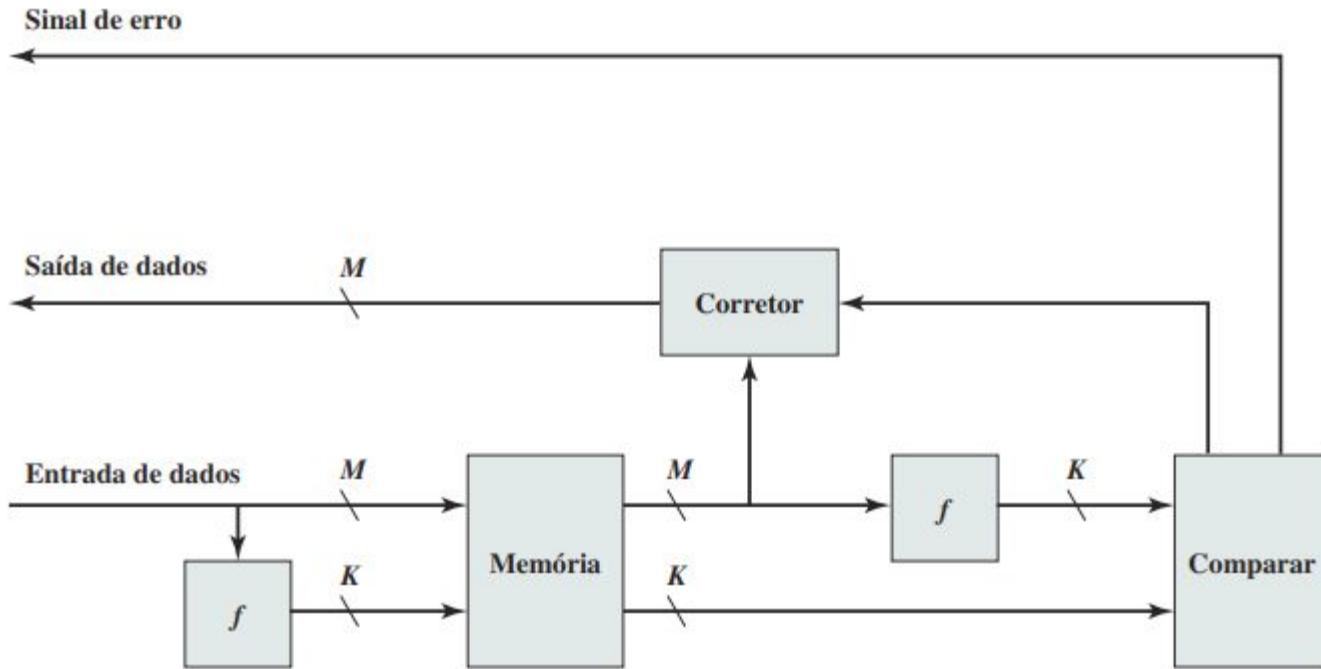
Aspectos Reais



Correção de Erros

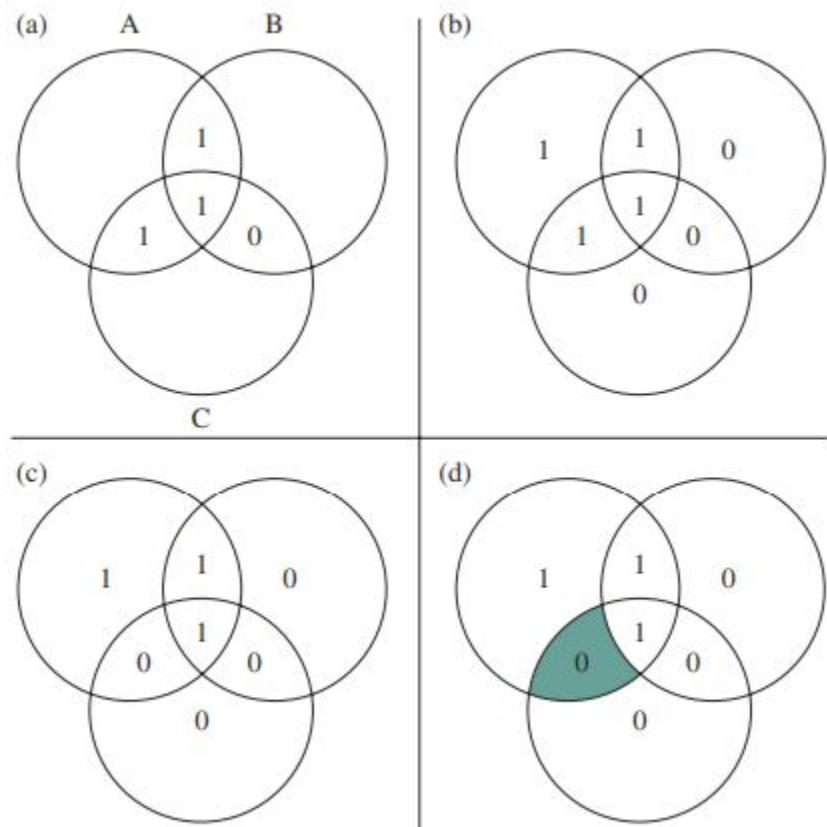
Um sistema de memória semicondutora está sujeito a erros. Eles podem ser categorizados como falhas permanentes ou erros não permanentes. A **falha permanente** é um defeito físico permanente, de modo que a célula ou células de memória afetadas não podem armazenar dados de modo confiável, mas ficam presas em 0 ou 1, ou alternam erroneamente entre 0 e 1. Os erros permanentes podem ser causados por uso intenso em ambiente impróprio, defeitos de fabricação ou desgaste. Um **erro não permanente** é um evento aleatório, não destrutivo, que altera o conteúdo de uma ou mais células de memória sem danificar a memória. Os erros não permanentes podem ser causados por problemas de fonte de alimentação ou partículas alfa. Essas partículas resultam de uma degradação radioativa e são terrivelmente comuns, pois os núcleos radioativos são encontrados em pequenas quantidades em quase todos os materiais. Erros permanentes e não permanentes certamente são indesejáveis, e a maioria dos sistemas de memória modernos inclui lógica para detectar e corrigir erros.

Correção de Erros na Memória



- ▶ Nenhum erro é detectado. Os bits de dados buscados são enviados.
- ▶ Um erro é detectado e é possível corrigi-lo. Os bits de dados mais os bits de **correção de erro** são alimentados em um corretor, que produz um conjunto correto de M bits a serem enviados.
- ▶ Um erro é detectado, mas não é possível corrigi-lo. Essa condição é relatada.

Diagrama de Venn



Palavra do Corretor de erro

Tabela 5.2

Aumento no tamanho da palavra com correção de erro.

Bits de dados	Correção de erro único		Correção de erro único/ detecção de erro duplo	
	Bits de verificação	% de aumento	Bits de verificação	% de aumento
8	4	50,0	5	62,5
16	5	31,25	6	37,5
32	6	18,75	7	21,875
64	7	10,94	8	12,5
128	8	6,25	9	7,03
256	9	3,52	10	3,91

Para conseguir essas características, os bits de dados e de verificação são dispostos em uma palavra de 12 bits, conforme representado na Figura 5.9. As posições de bit são numeradas de 1 a 12. Essas posições de bit cujos números de posição são potências de 2 são designadas como bits de verificação. Os bits de verificação são calculados da seguinte forma, onde o símbolo \oplus designa a operação OU-EXCLUSIVO:

Código de Correção Erro de Hamming

Figura 5.9

Layout de bits de dados e bits de verificação.

Posição de bit	12	11	10	9	8	7	6	5	4	3	2	1
Número de posição	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Bit de dados	D8	D7	D6	D5		D4	D3	D2		D1		
Bits de verificação					C8				C4		C2	C1

$$C1 = D1 \oplus D2 \oplus D4 \oplus D5 \oplus D7$$

$$C2 = D1 \oplus D3 \oplus D4 \oplus D6 \oplus D7$$

$$C4 = D2 \oplus D3 \oplus D4 \oplus D8$$

$$C8 = \oplus D5 \oplus D6 \oplus D7 \oplus D8$$

Layout Bit de Dados e Verificação

Objectivo

O objectivo do código de Hamming é criar um conjunto de bits de paridade, que se sobrepõem de tal forma que um erro num único bit (de dados ou paridade) pode ser detectado e corrigido.

Embora possam ser criadas múltiplas sobreposições, existe um método geral apresentado em código de Hamming.

Bit #	1	2	3	4	5	6	7
Bit transmitido	p_1	p_2	d_1	p_3	d_2	d_3	d_4
p_1	✓	✗	✓	✗	✓	✗	✓
p_2	✗	✓	✓	✗	✗	✓	✓
p_3	✗	✗	✗	✓	✓	✓	✓

Esta tabela descreve a cobertura efectuada pelos bits de paridade. Por exemplo, p_2 fornece um controlo de paridade os bits 2, 3, 6, e 7. Numa leitura em coluna podemos observar que o bit d_1 é coberto pelos bits de paridade p_1 e p_2 . Esta tabela tem uma grande semelhança com a matriz de paridade H da próxima secção.

Se as colunas correspondentes aos bits de paridade foram removidas da tabela acima, temos:

	d_1	d_2	d_3	d_4
p_1	✓	✓	✗	✓
p_2	✓	✗	✓	✓
p_3	✗	✓	✓	✓

Assim, escolhendo correctamente a cobertura dos bits de paridade, todos os erros com distância Hamming de 1 podem ser detectados e corrigidos, que é o objectivo da utilização de um código de Hamming.



Codificação

Codificação

Vamos supor que queremos transmitir o vector \mathbf{p} através de um canal de comunicação ruidoso. Um Canal binário simétrico, o que significa que o erro não favorece o valor zero ou um. Além disso, todos os vectores de código são assumidos como sendo equiprováveis. Tomamos o produto de G e \mathbf{p} módulo 2, para determinar a palavra de código a transmitir \mathbf{x} :

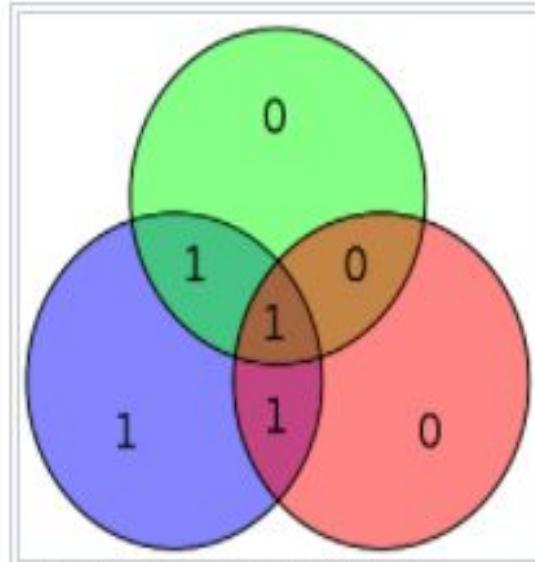
$$\mathbf{x} = \mathbf{G}\mathbf{p} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 1 \\ 2 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Isto significa que `0110011` será transmitido em vez de `10111`.

No diagrama à direita, os 7 bits da palavra codificada são inseridos nas suas respectivas posições, observando as paridades, é evidente que a paridade do vermelho, verde e azul é par:

- círculo vermelho tem dois 1
- círculo verde tem dois 1
- círculo azul tem quatro 1

O que vamos observar seguidamente é que, se durante a transmissão um bit for invertido a paridade de 2 ou 3 círculos será incorrecta e o bit com erro pode ser determinado (mesmo se for um dos bits de paridade), sabendo que a paridade dos três círculos deve ser par.



Mapeamento do exemplo \mathbf{x} . A paridade dos círculos vermelho, verde e azul é par.

Paridade

Paridade

Se não ocorrer um erro durante a transmissão, a palavra de código recebida r é idêntica à palavra transmitida x :

$$r = x$$

O receptor multiplica H e r para obter o vector z denominado síndroma, que indica se ocorreu um erro, e se assim for, qual o bit da palavra de código que o contém. Realizar esta multiplicação com módulo 2.

$$z = Hr = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \\ 2 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Como o síndroma z é o vector nulo, o receptor pode concluir que não ocorreu nenhum erro. Esta conclusão é baseada na observação de que quando o vector de dados é multiplicada por G , ocorre uma mudança de base para um sub-espaco vectorial que é o núcleo de H . Desde que nada aconteça durante a transmissão, r permanecerá no núcleo de H e multiplicação irá produzir o vector nulo.

Decodificação

Descodificação

Assim que for determinado que o vector recebido está livre de erros ou for corrigido de um erro, será então necessário descodificá-lo de volta à sua forma original de 4 bits.

Em primeiro lugar definimos uma matriz **R**:

$$\mathbf{R} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Temos então que o valor recebido \mathbf{p}_r será:

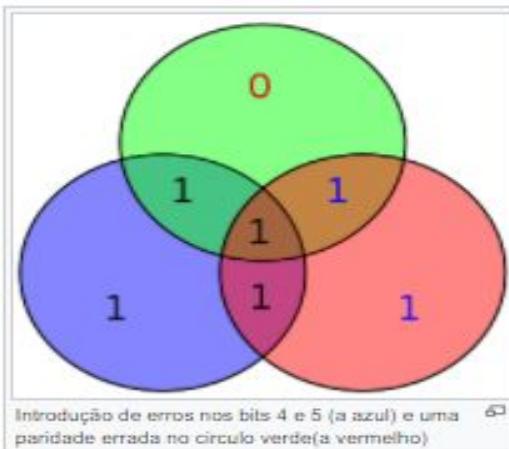
$$\mathbf{p}_r = \mathbf{R}\mathbf{r}$$

$$\mathbf{p}_r = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Múltiplos erros

Não é difícil mostrar que apenas os erros de um bit pode ser corrigidos utilizando este esquema. Como alternativa, os códigos de Hamming podem ser utilizados para detectar erros de um ou dois bits, por indicação do produto de H ser diferente de zero quando ocorrem erros. No diagrama da direita, os bits 4 e 5 foram alterados. Isso produz apenas um círculo (verde) com uma paridade inválida, mas os erros não são corrigíveis.

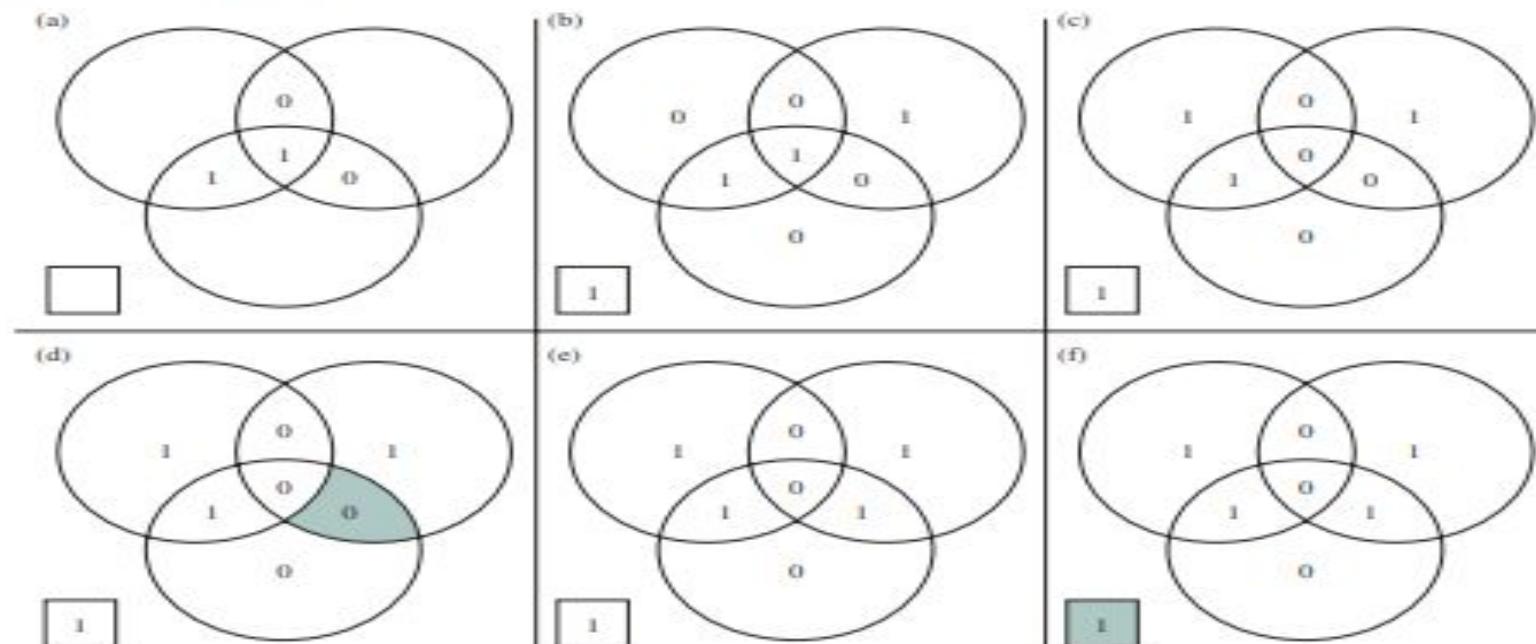
No entanto, o Hamming (7,4) e códigos de Hamming semelhantes não conseguem distinguir entre erros de um bit e de dois. Ou seja, dois erros parecem o mesmo que erros de um bit. Se a correção de erro for realizada num erro de dois bits, o resultado será incorreto.



Bit de Verificação/Código Sec Dec de Hamming

Posição de bit	12	11	10	9	8	7	6	5	4	3	2	1
Número de posição	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Bit de dados	D8	D7	D6	D5		D4	D3	D2		D1		
Bits de verificação					C8				C4		C2	C1
Palavra armazenada como	0	0	1	1	0	1	0	0	1	1	1	1
Palavra buscada como	0	0	1	1	0	1	1	0	1	1	1	1
Número de posição	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Bits de verificação					0				0		0	1

Figura 5.11
Código SEC-DEC de Hamming.



Bit de Verificação/Código Sec Dec de Hamming

Palavras válidas

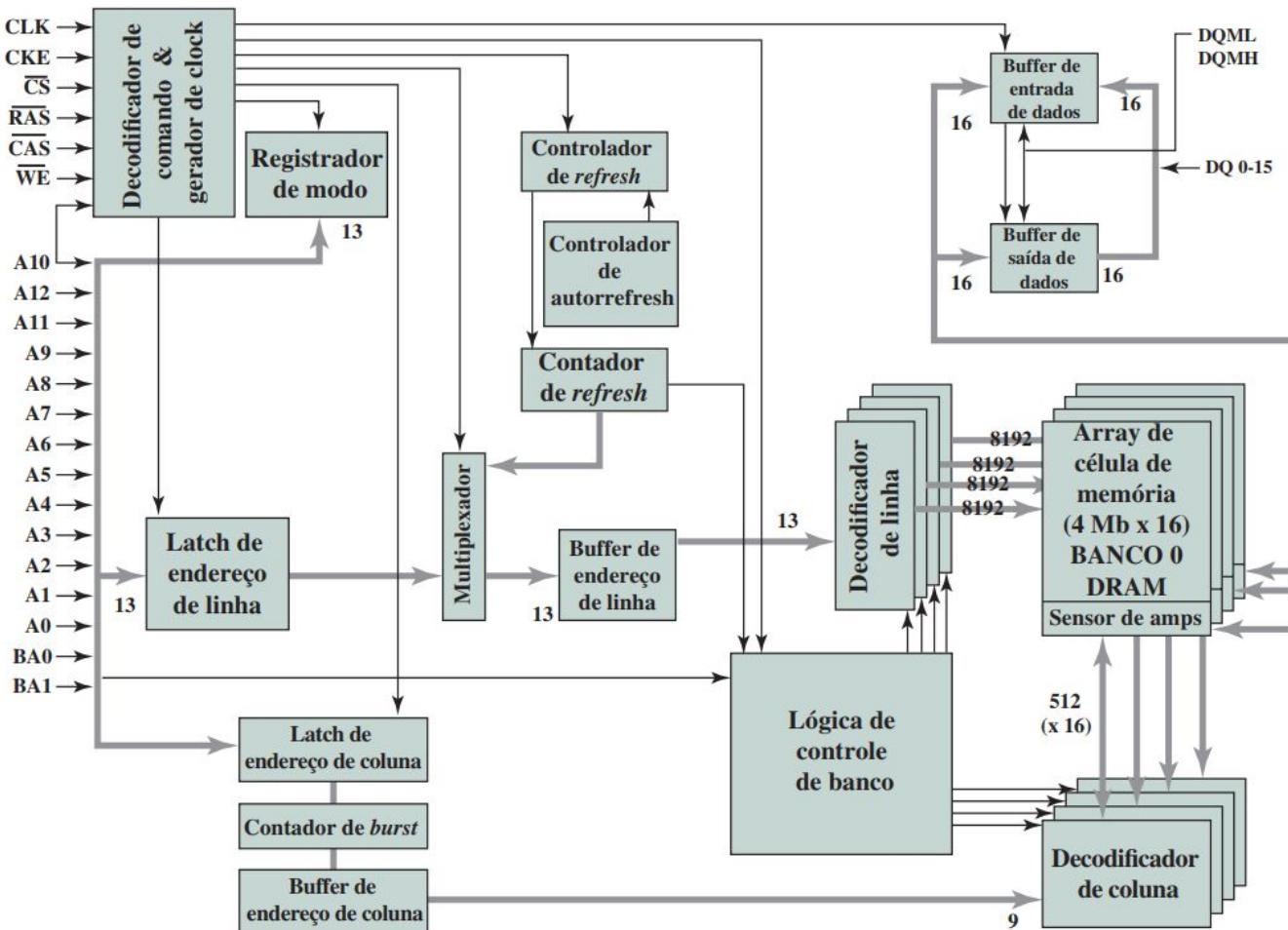
Como a fonte é de 4 bits existem apenas 16 palavras possíveis. A segunda coluna contém o valor a 8 bits se utilizarmos um bit de paridade extra (ver Hamming (7,4) com um bit de paridade adicional). (Os bits de dados são mostrados a azul, os bits de paridade são mostrados a vermelho, e o bit de paridade adicional é mostrado a verde).

Dados (d_1, d_2, d_3, d_4)	Hamming(7,4)		Hamming(7,4) com bit de paridade extra (Hamming(8,4))	
	Transmitido ($p_1, p_2, d_1, p_3, d_2, d_3, d_4$)	Diagrama	Transmitido ($p_1, p_2, d_1, p_3, d_2, d_3, d_4, p_4$)	Diagrama
0000	0000000		0000000	
1000	1110000		11100001	
0100	1001100		10011001	
1100	0111100		01111000	

DDR-DRAM

SDRAM (DRAM Síncrona)

RAM dinâmica síncrona (SDRAM) de 256 Mb.

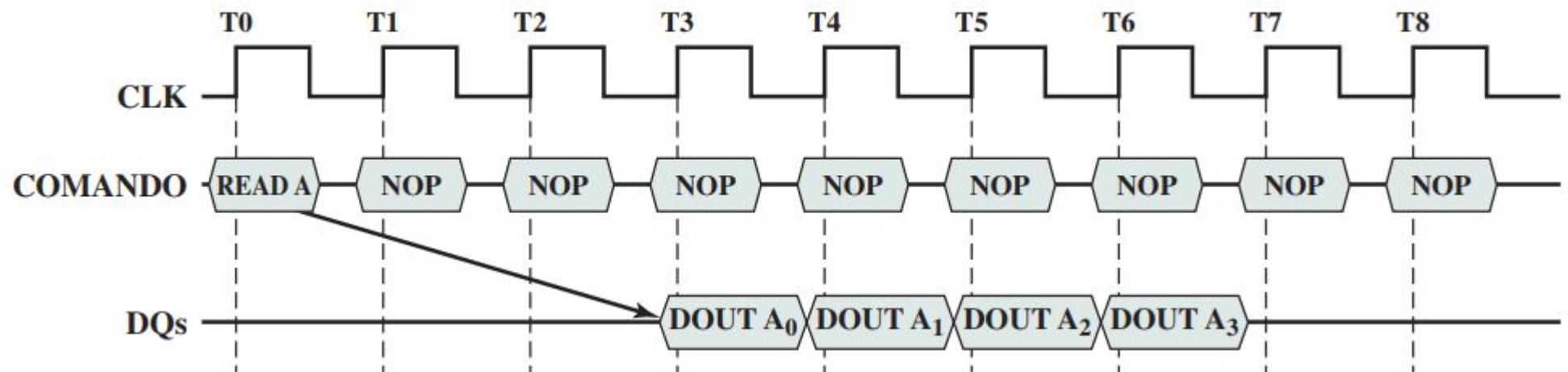


A0 a A13	Entradas de endereço
BA0, BA1	Linhas de endereço de banco
CLK	Entrada de clock
CKE	Habilitação de clock
CS	Seleção de chip
RAS	Strobe de endereço de linha
CAS	Strobe de endereço de coluna
WE	Habilitação de escrita
DQ0 a DQ7	Entrada/saída de dados
DQM	Máscara de dados

DDR-DRAM

SDRAM (DRAM Síncrona)

Temporização de leitura da SDRAM (extensão de rajada = 4, latência de CAS = 2).



DDR-DRAM

DDR-SDRAM (Taxa Dupla de Dados – SDRAM)

Aumento da Taxa de Dados em relação a SDRAM.

Frequência de Clock mais alta no barramento.

Buffering utilizado.

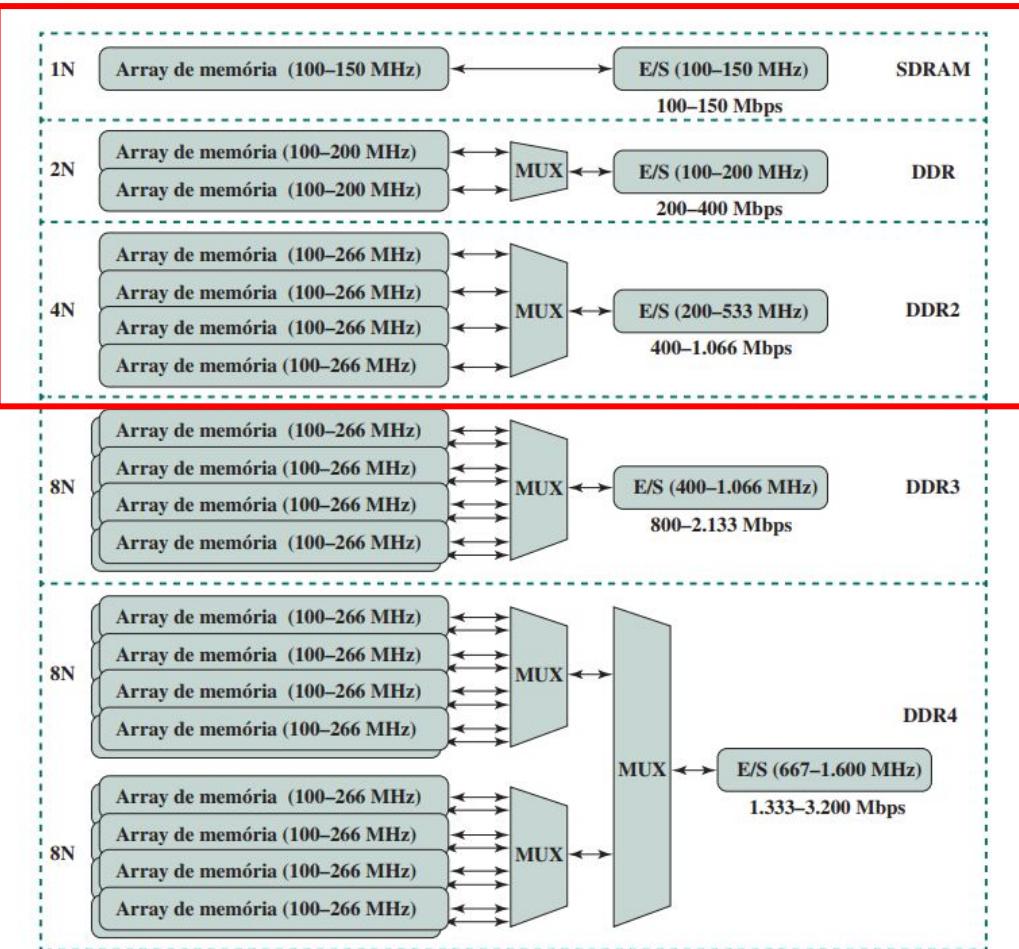
Características da DDR.

	DDR1	DDR2	DDR3	DDR4
Buffer de pré-busca (bits)	2	4	8	8
Nível de voltagem (V)	2,5	1,8	1,5	1,2
Taxas de dados do barramento frontal (Mbps)	200–400	400–1.066	800–2.133	2.133–4.266

DDR-DRAM

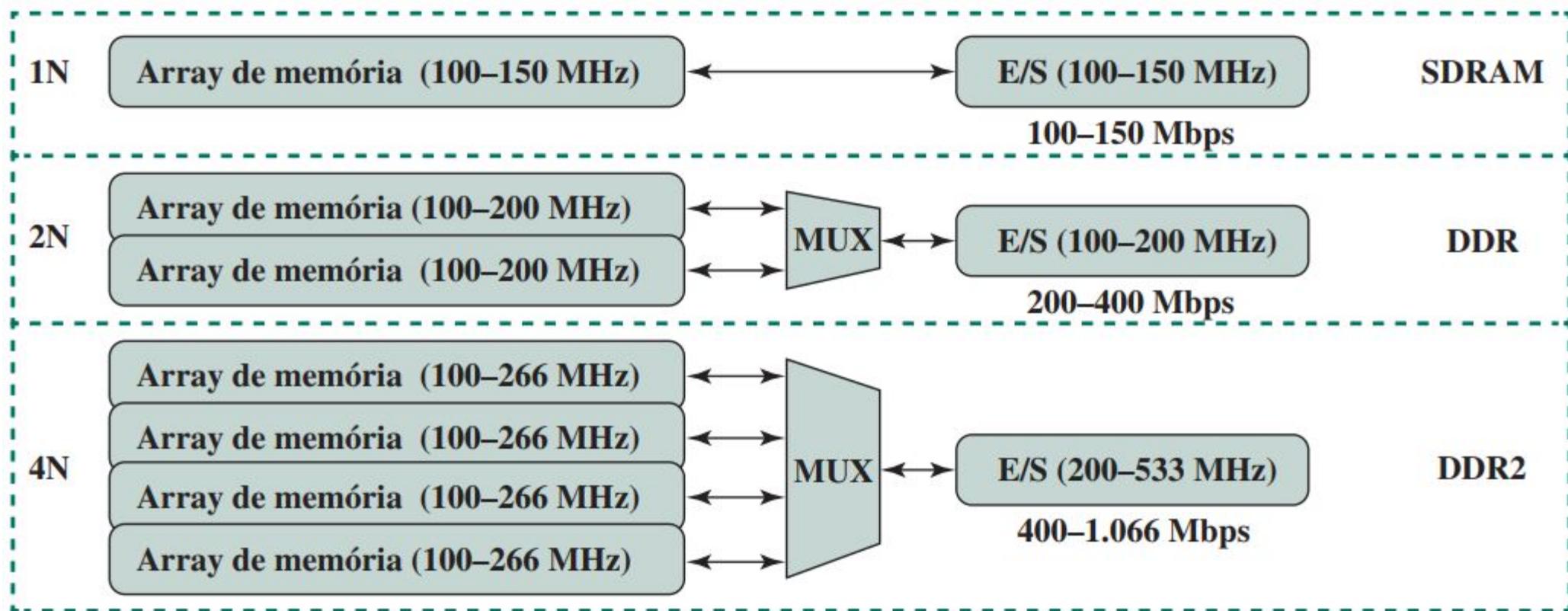
DDR-SDRAM (Taxa Dupla de Dados – SDRAM)

Gerações de DDR.



DDR-DRAM

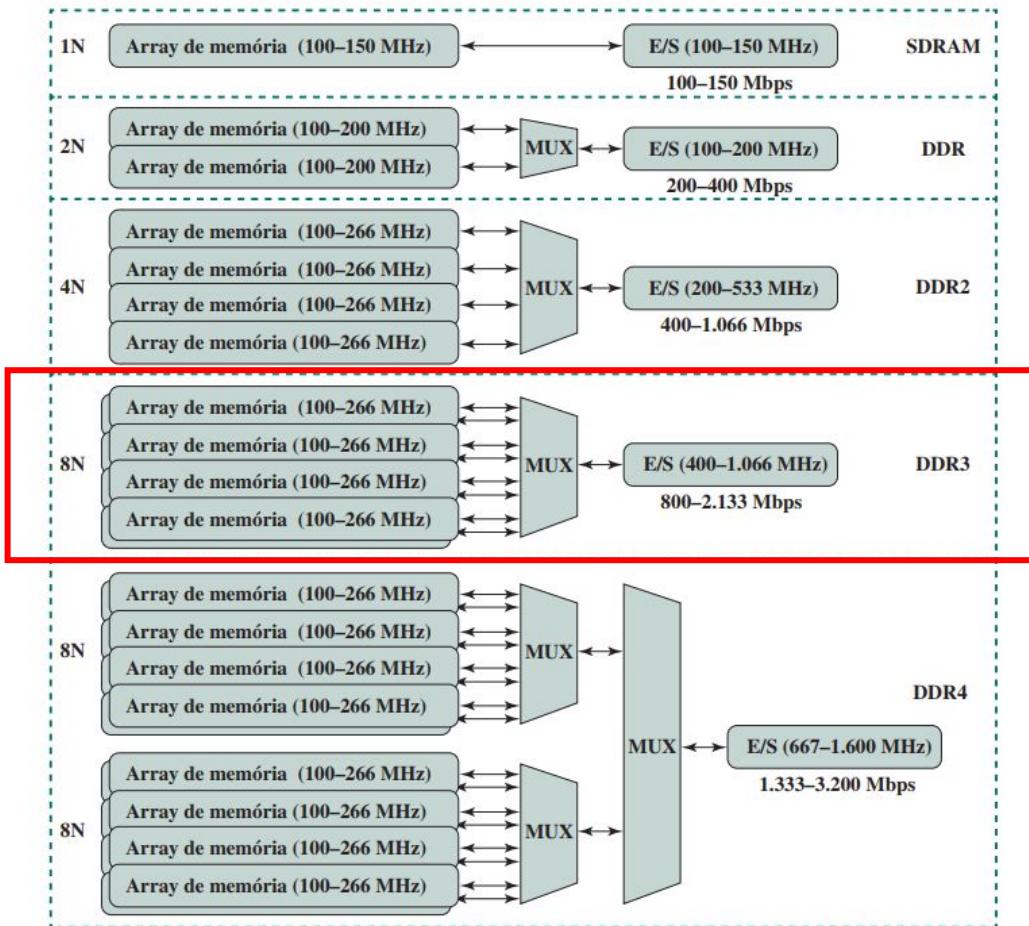
DDR-SDRAM (Taxa Dupla de Dados – SDRAM)



DDR-DRAM

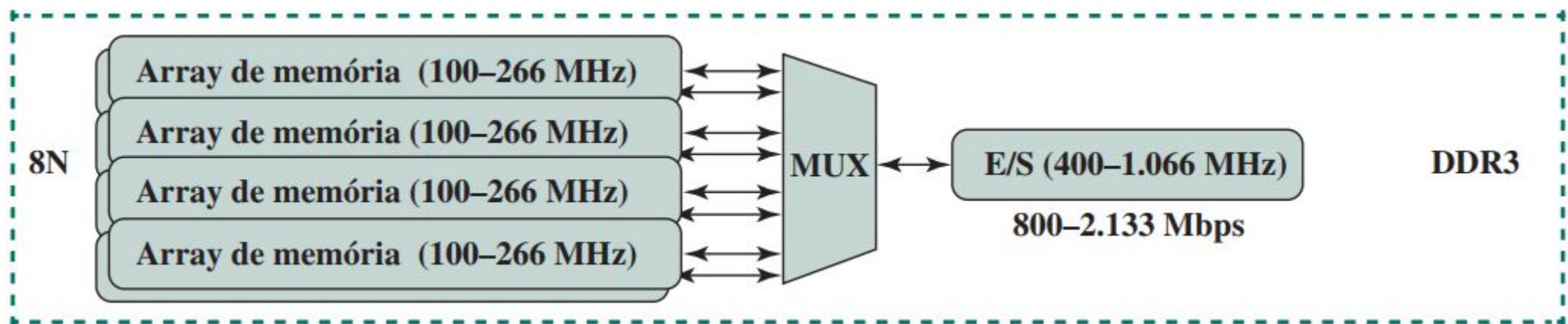
DDR-SDRAM (Taxa Dupla de Dados – SDRAM)

Gerações de DDR.



DDR-DRAM

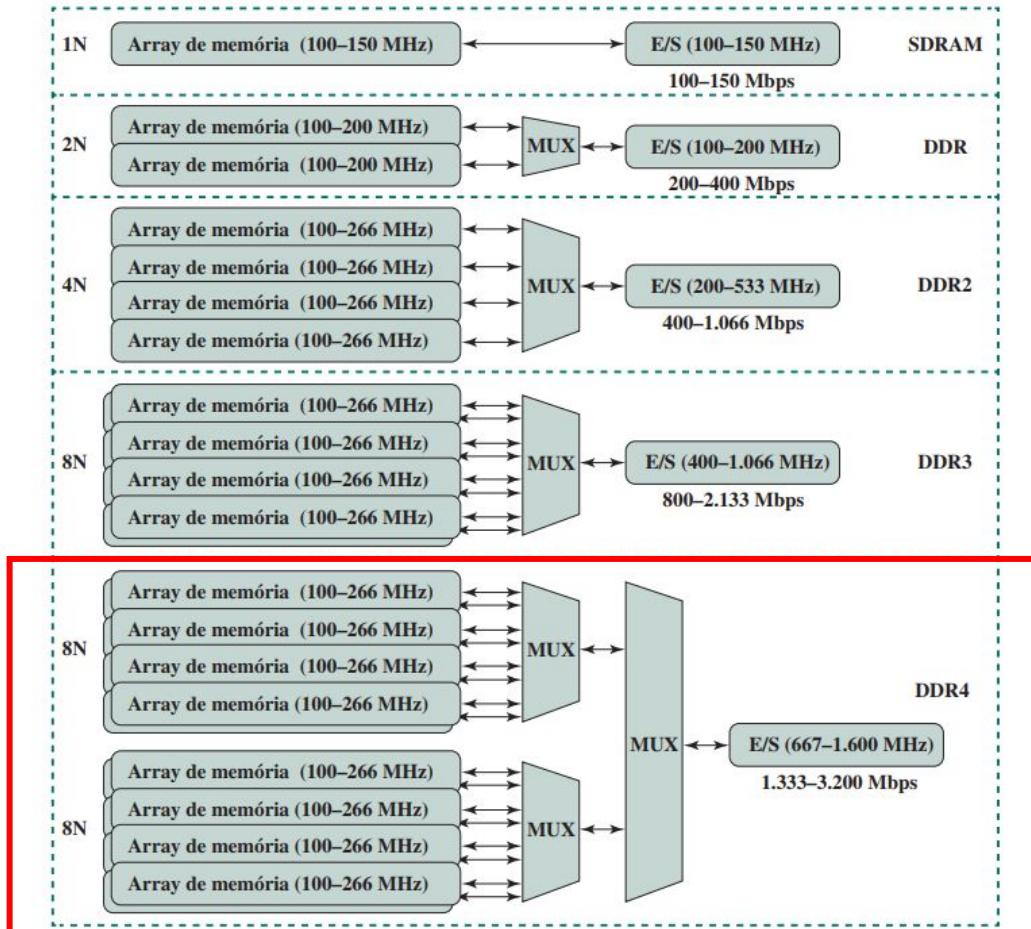
DDR-SDRAM (Taxa Dupla de Dados – SDRAM)



DDR-DRAM

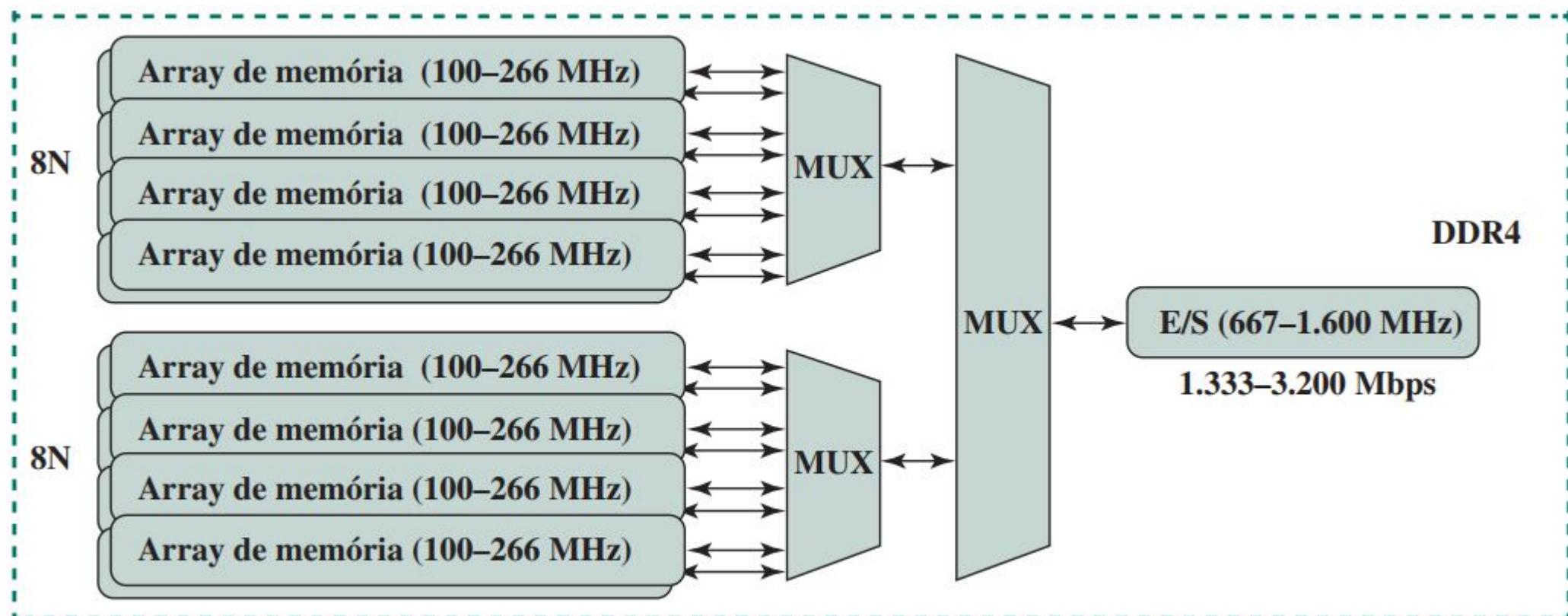
DDR-SDRAM (Taxa Dupla de Dados – SDRAM)

Gerações de DDR.

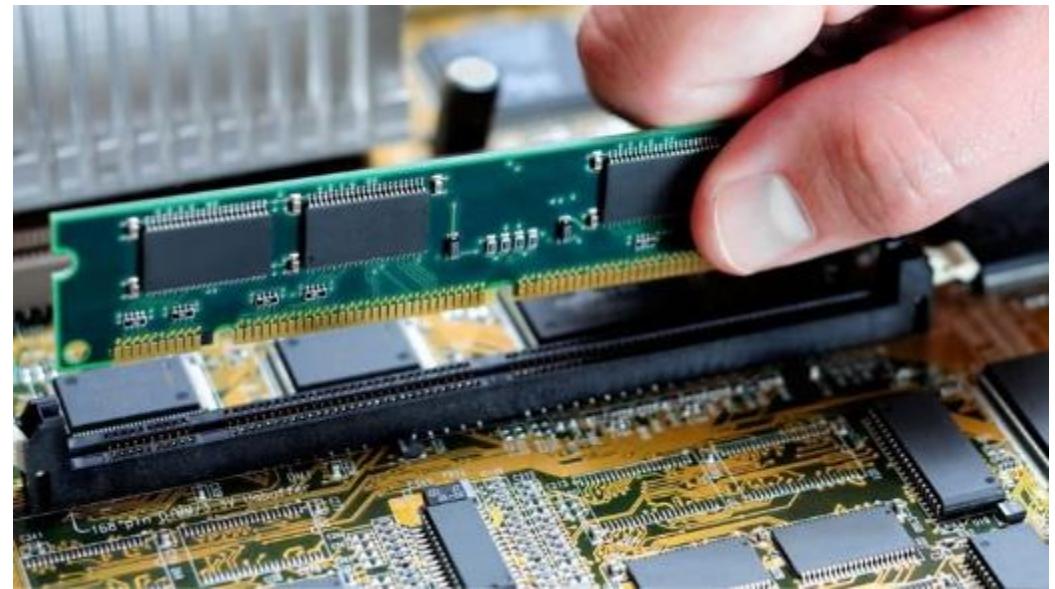
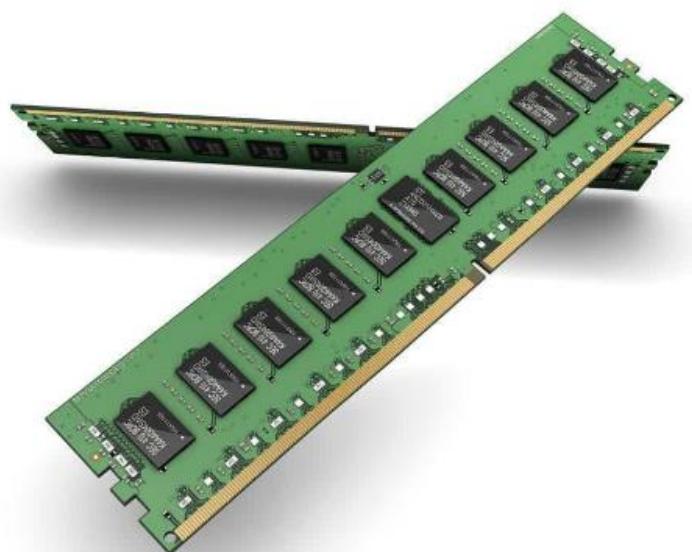


DDR-DRAM

DDR-SDRAM (Taxa Dupla de Dados – SDRAM)



DDR-DRAM

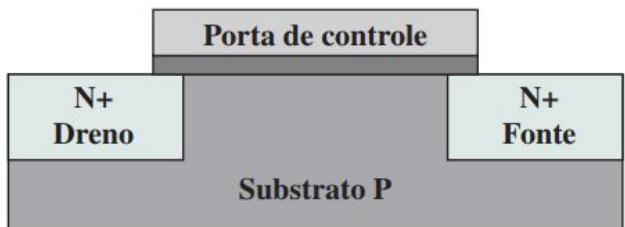


Memória Flash

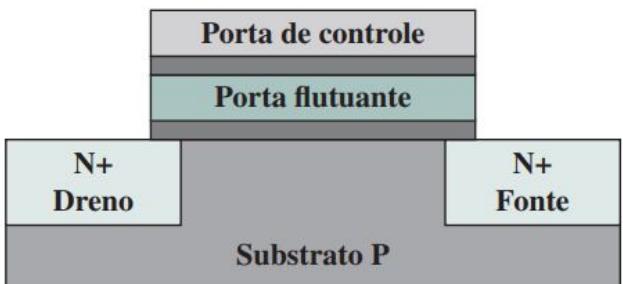
Operação

Pode ser usada como memória interna ou externa.

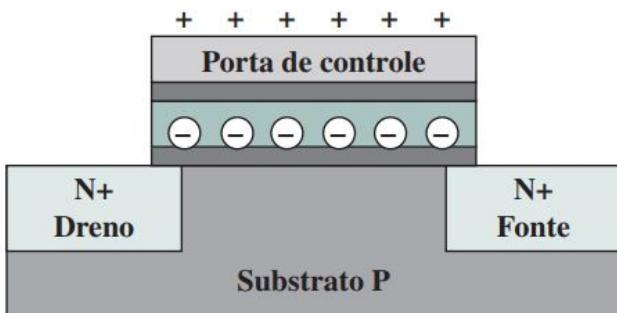
Operação da memória flash.



(a) Estrutura do transistor



(b) Célula de memória flash no estado 1

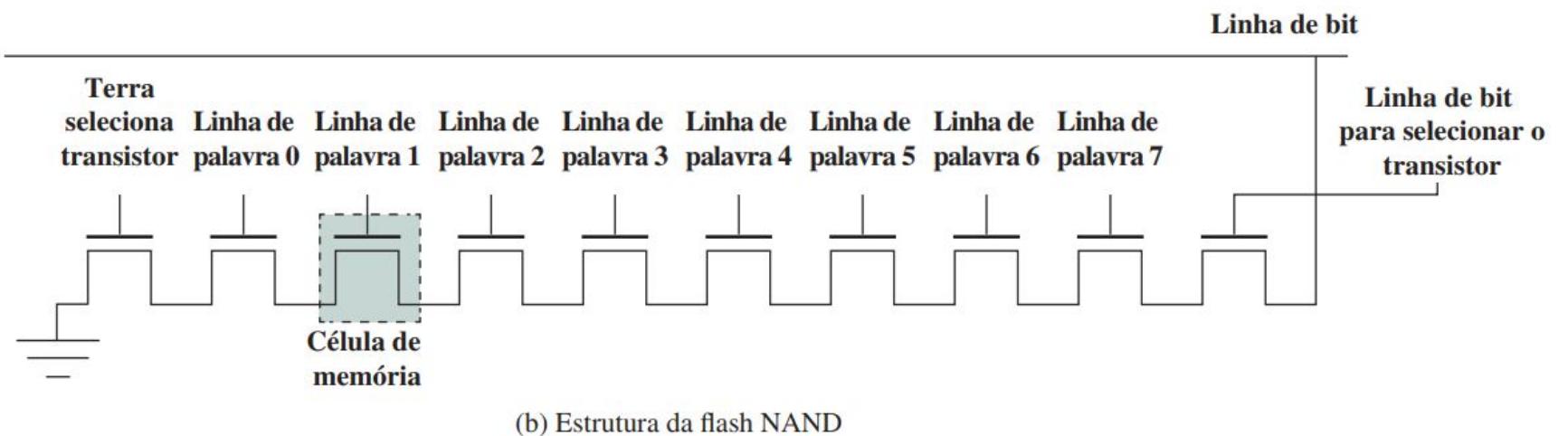
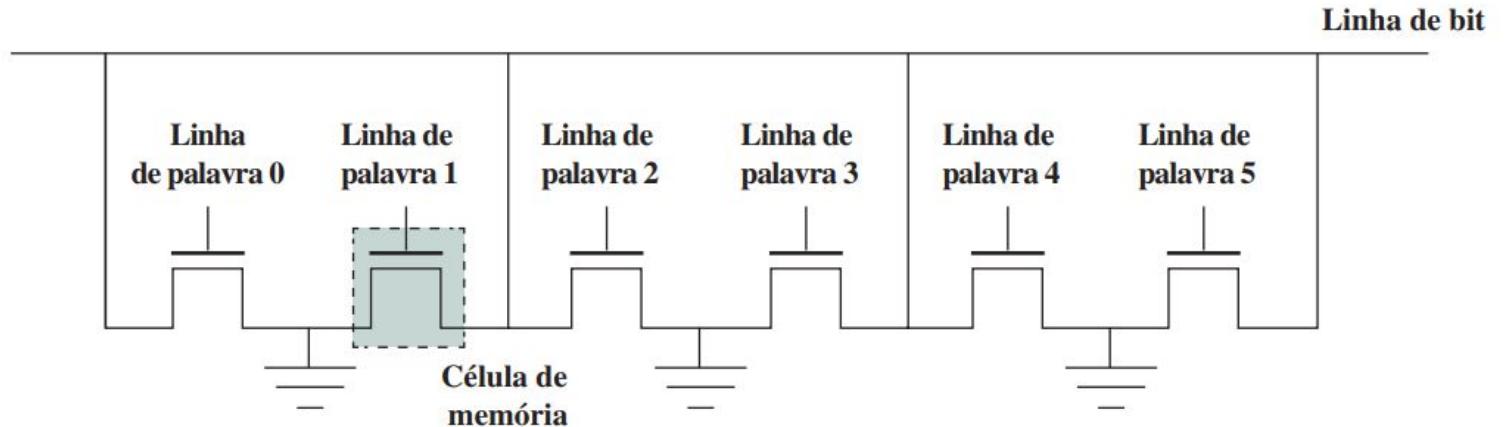


(c) Célula de memória flash no estado 0

Memória Flash

NOR e NAND

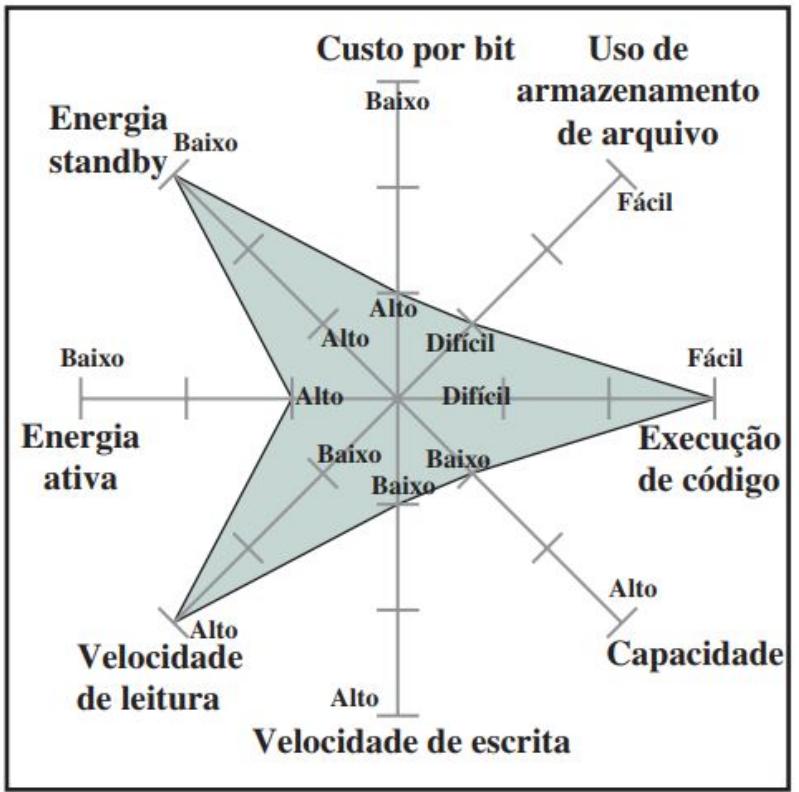
Estruturas da memória flash.



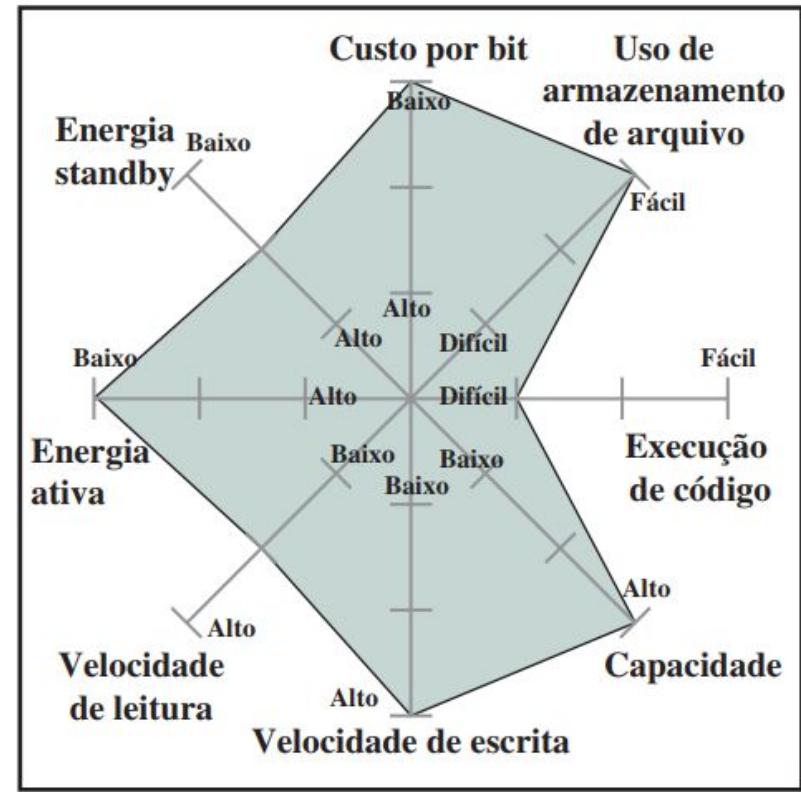
Memória Flash

NOR e NAND

Gráficos de Kiviat para a memória flash.



(a) NOR



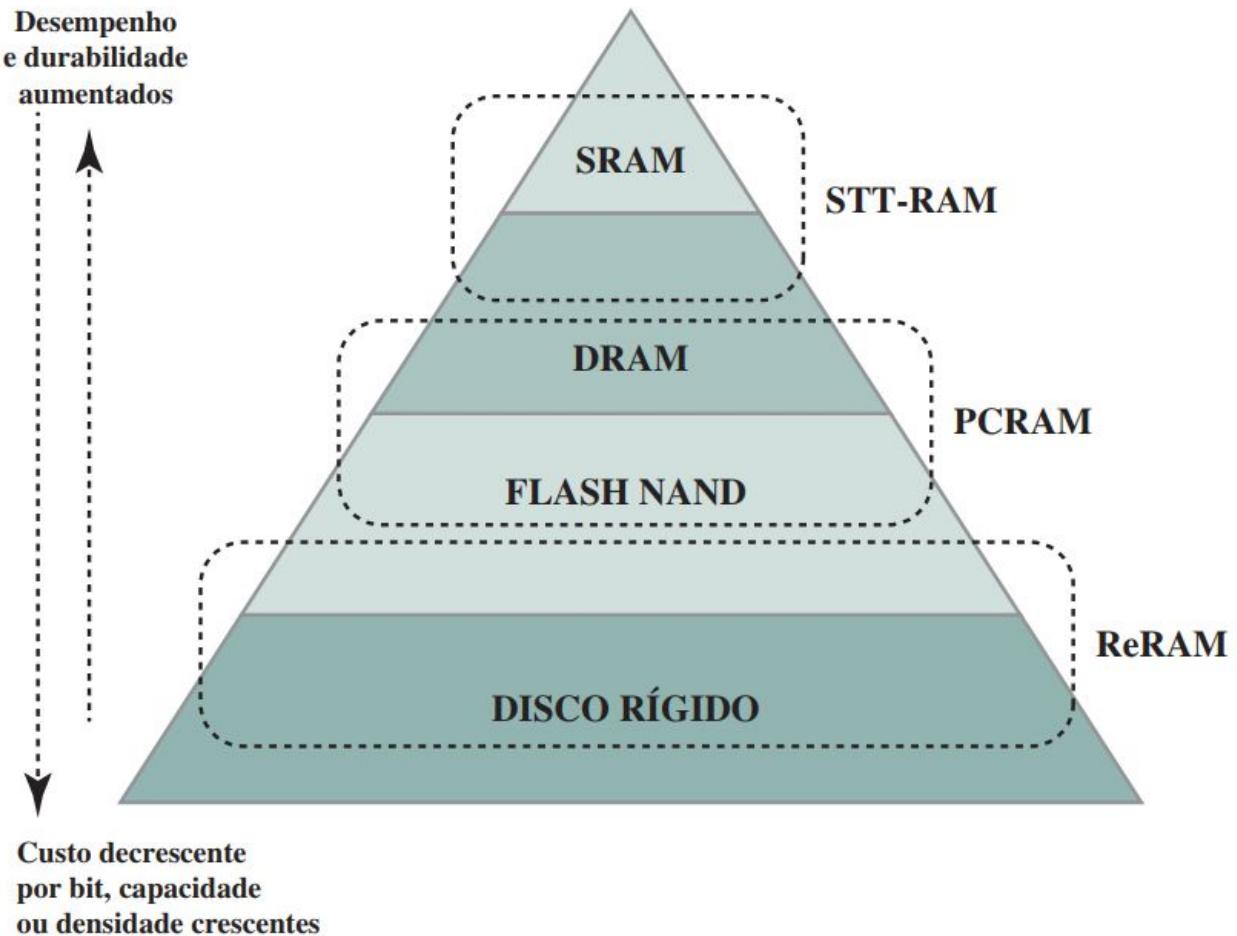
(b) NAND

Hierarquia

- **RAM estática (SRAM)**: a SRAM proporciona tempo de acesso rápido, mas é a mais cara e menos densa (densidade de bit). A SRAM é adequada para a memória cache.
- **RAM dinâmica (DRAM)**: mais barata, mais densa e mais lenta do que a SRAM, a DRAM tem sido tradicionalmente a escolha como memória principal off-chip.
- **Disco rígido**: o disco magnético proporciona densidade de bit mais alta e custo muito menor por bit, com tempos de acesso relativamente menores. É a escolha tradicional para armazenamento externo, como parte da hierarquia da memória.

Hierarquia

RAM não volátil dentro da hierarquia da memória.



Questões



- 1) Explique as diferenças entre RAM, ROM, EPROM, EEPROM e memória FLASH.
- 2) O que é bit de paridade?
- 3) Explique as diferenças entre SDRAM de uma DRAM comum. Explique também o que é uma DDR-RAM e suas gerações.