

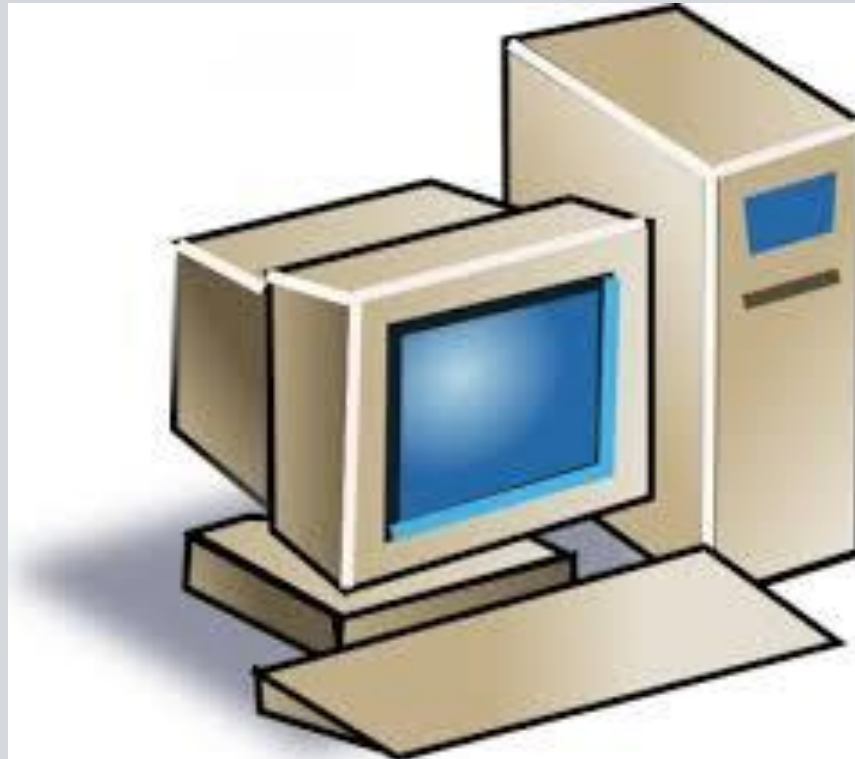
# Introdução à arquitetura de computadores

EMB5642 – Aula 1

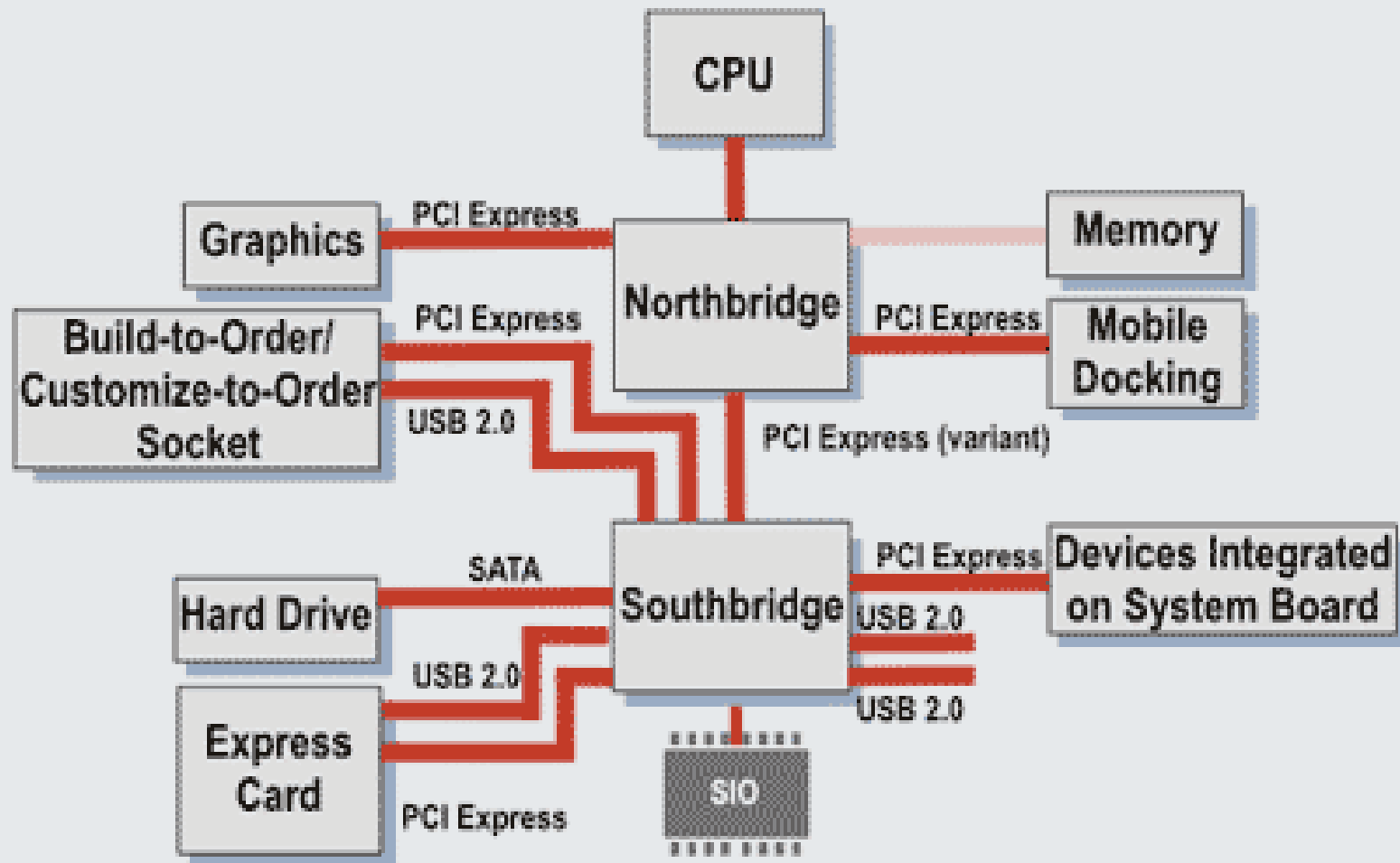


UNIVERSIDADE FEDERAL  
DE SANTA CATARINA

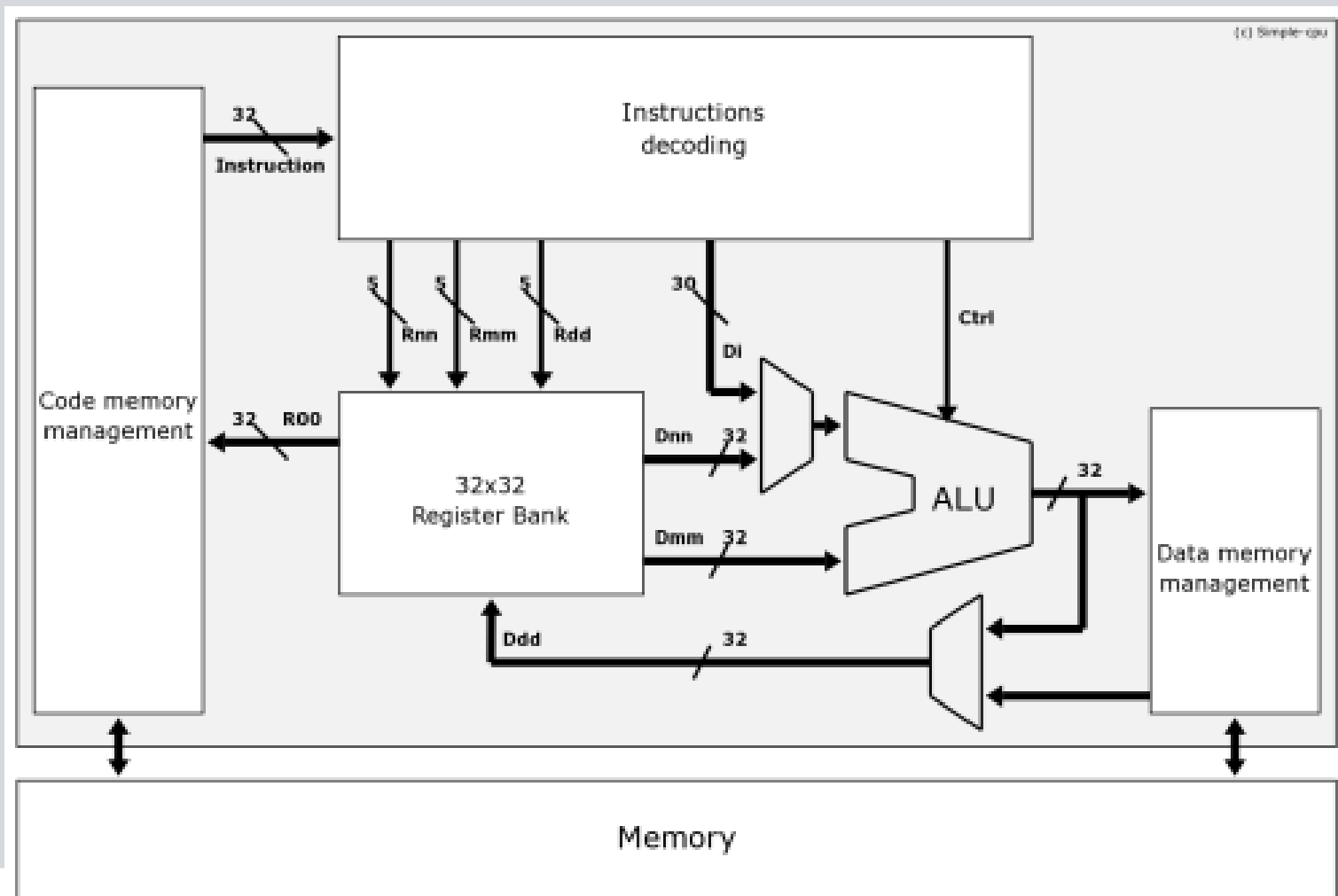
# Quais os componentes básicos de um computador?



# Arquitetura de um Pentium

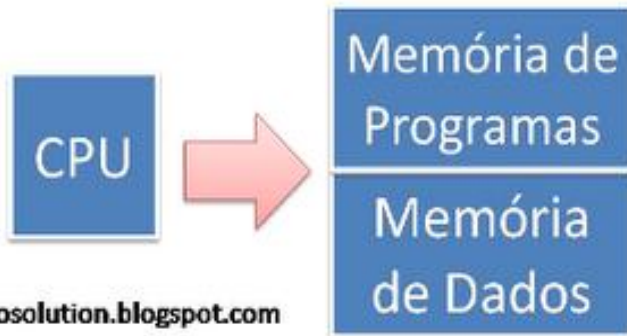


# E dentro da CPU...

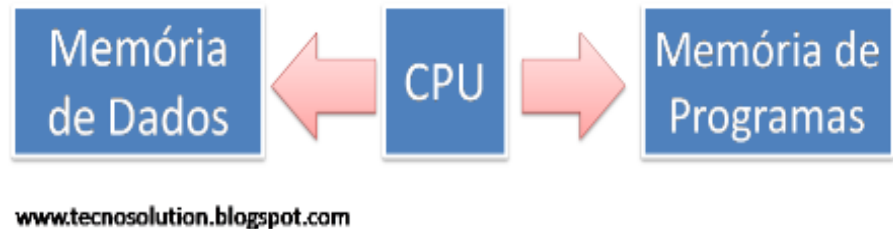


# Conceitos importantes: Arquitetura

## Modelo Von Neumann



## Modelo Harvard



# Conjunto de Instruções

- RISC (Reduced Instruction Set Computer) é uma estratégia para projeto de processadores que favorece um conjunto simples e pequeno de instruções que levam aproximadamente a mesma quantidade de tempo para serem executados (geralmente um ciclo de máquina).
- Embora haja diversas abordagens implementadas, há algumas características comum à todas, como:
  - Uma operação por ciclo
  - Operações registrador-para-registrador
  - Modos de endereçamento simples (no máximo 5 tipos)
  - Formato das instruções simples (pois o tamanho da instrução é fixo)

# Conjunto de Instruções

- CISC (Complex instruction set computing) via não mais a redução do conjunto de instruções, mas a existência de um conjunto maior de instruções complexas que executem tarefas complexas com maior eficiência e rapidez e que também simplifiquem os compiladores.

# Barramento

- Barramento é um caminho elétrico comum entre dois ou mais dispositivos, um ponto chave é que barramento é um meio de transmissão compartilhado, múltiplos dispositivos podem se conectar ao barramento, mas somente um dispositivo de cada vez pode transmitir com sucesso.
  - Barramento de endereço
  - Barramento de dados
  - Barramento de controle



# Memórias

- Em sistemas, memória refere-se aos dispositivos usados para armazenar programas (sequências de instruções) e/ou dados de forma temporária ou permanente.
- As memórias, no âmbito de sistema, possuem três características principais: capacidade, tempo de acesso e custo, existindo a seguinte relação entre elas:
  - Tempo de acesso mais rápido, maior o custo por bit
  - Maior capacidade, menor o custo por bit
  - Maior capacidade, tempo de acesso mais lento

# Memórias

- Diminuição de custo por bit
- Aumento da capacidade
- Aumento do tempo de acesso
- Diminuição da frequência de acesso à memória



# Memórias: Registradores

- São pequenas unidades de memória para alocação temporária de dados. Ou seja, trabalham diretamente com o processador, armazenando os resultados de suas operações lógicas e aritméticas temporariamente até que esse dado seja reutilizado em outra instrução ou transferido para outra forma de memória.
  - Registradores visíveis ao usuário. Uso geral, de dados, de endereços (ponteiros de segmentos, registros de índice e ponteiros de pilha) e códigos condicionais (flags).
  - Registradores de controle e estado.
    - Contador de Programa (PC)
    - Registrador da instrução (IR)
    - Registrador de endereço de memória (MAR)
    - Registrador de buffer de memória (MBR)

# Memórias: Cache

- O uso da cache visa obter velocidade de memória próximo das memórias mais rápidas com um custo não tão elevado. A memória cache inicialmente transferia palavras com a CPU e requisitava blocos com a RAM.
- Alguns estudos relacionaram a quantidade de acertos com a capacidade da CACHE e observaram que a partir de 32KBytes não há mais aumento. Assim o que vem sendo utilizado são múltiplos níveis de cache (L1, L2 e L3) sendo que a L1 é a mais rápida e menor e a L3 a mais lenta e maior.
- Por exemplo o i7 a L1 é de 64kB (32kB de dados e 32kB de instrução) por núcleo, L2 são 256 kB por núcleo e L3 são 8MB compartilhada por todos os núcleos, sendo a L2 e L3 combinam dados e instruções.
- Essas sendo as caches internas ao núcleo, podendo existir além das caches externas processador-memória principal, caches para comunicação com dispositivos.

# Memórias: Mem. Principal

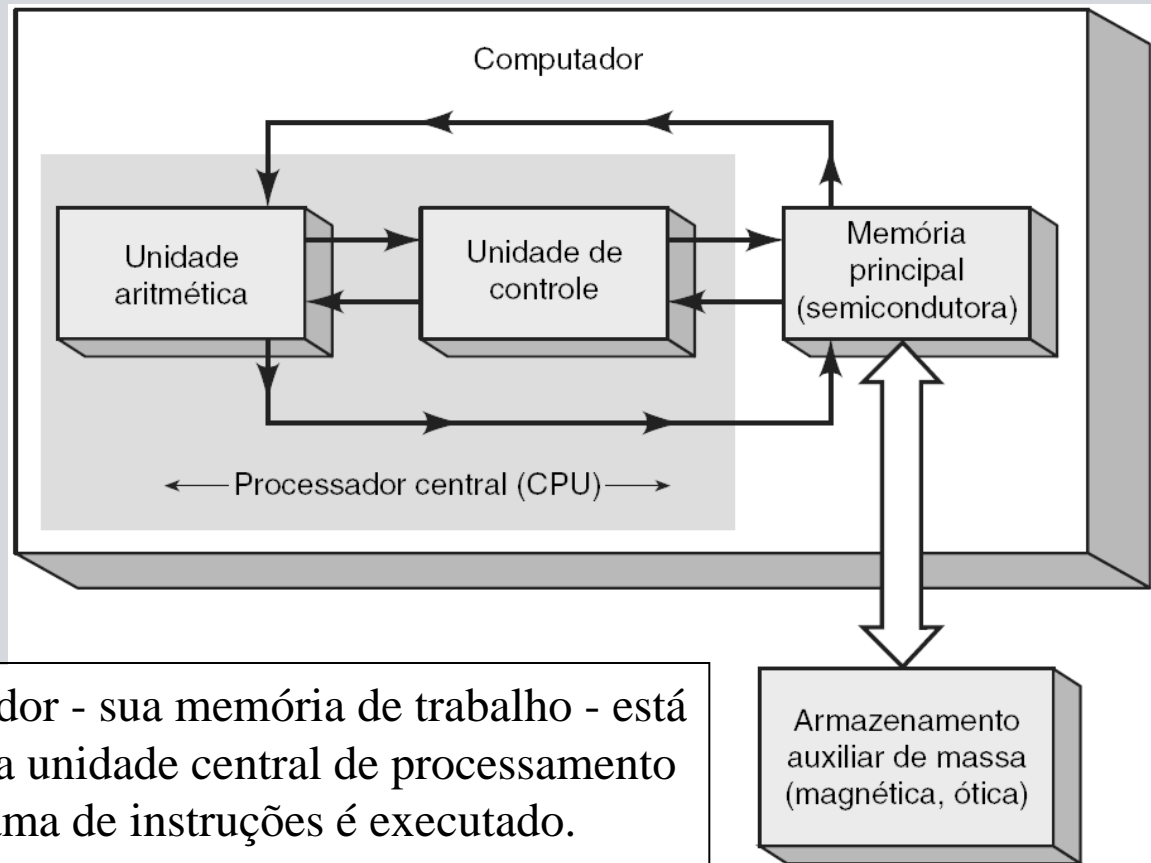
- A memória principal é a memória básica de um sistema de computação desde seus primórdios, é o dispositivo onde o programa e seus dados, do que vai ser executado é armazenado para que o processador vá buscando as informações quando necessárias.
- A tecnologia hoje utilizada na fabricação de Memórias Principais é Memória de Acesso Aleatório (RAM) do tipo dinâmico (SDRAM).

# Memórias: Armazenam. em Massa

- Nos PCs há um ou mais dispositivos de armazenamento em massa, HDs que podem ser discos magnéticos ou memórias flash (SSD), que são responsáveis por armazenar permanentemente os programas e dados do sistema.

# Uso de Memórias em computadores

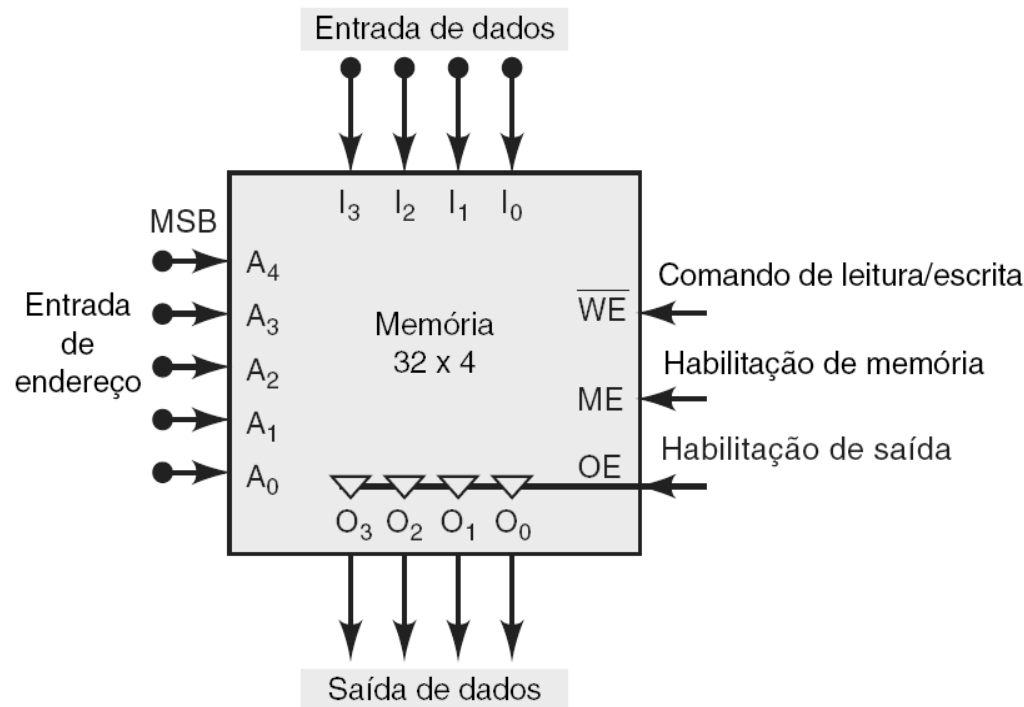
RAM e ROM formam a memória principal.



A memória principal do computador - sua memória de trabalho - está em constante comunicação com a unidade central de processamento (CPU) conforme um programa de instruções é executado.

# Princípios de Operação da Memória

- Diagrama de uma memória 32 x 4 e arranjo virtual das células de memória em 32 palavras de quatro bits:

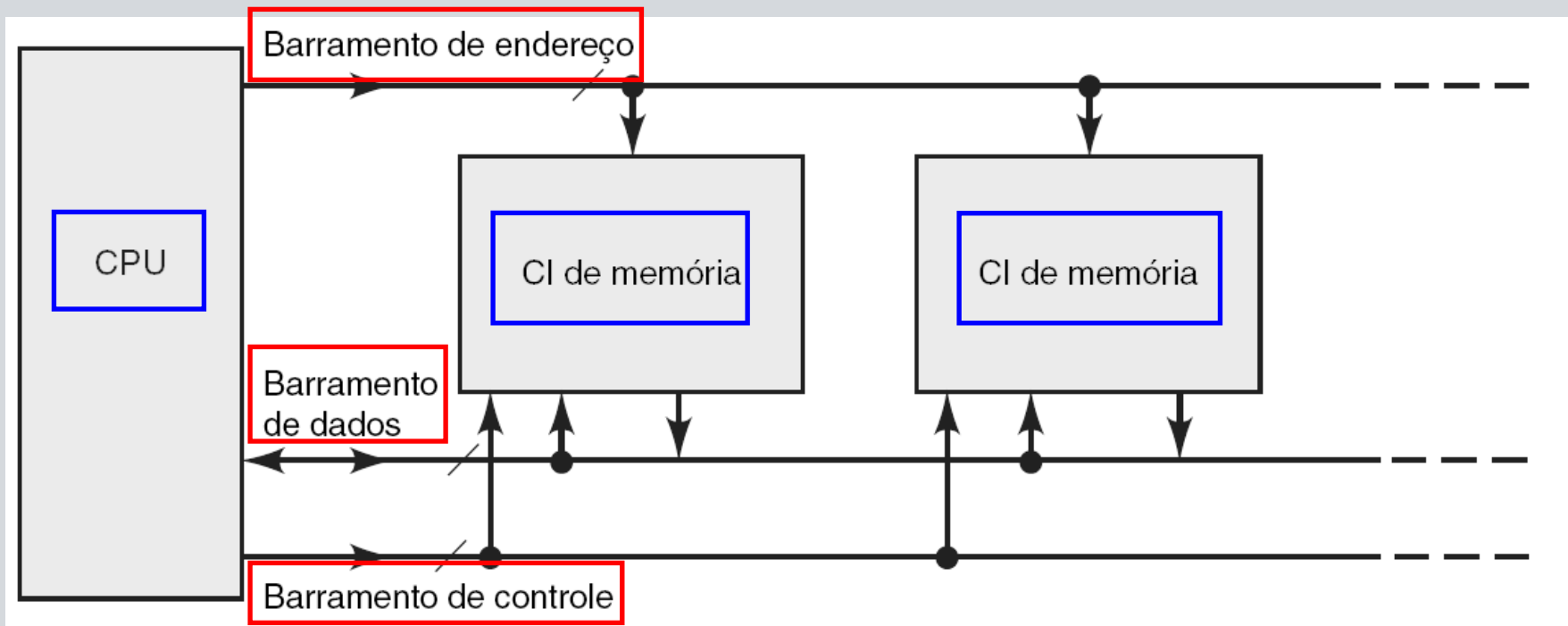


Células de memória

				Endereços
0	1	1	0	0 0 0 0 0
1	0	0	1	0 0 0 0 1
1	1	1	1	0 0 0 1 0
1	0	0	0	0 0 0 1 1
0	0	0	1	0 0 1 0 0
0	0	0	0	0 0 1 0 1
⋮	⋮	⋮	⋮	⋮ ⋮
1	1	0	1	1 1 1 0 1
1	1	0	1	1 1 1 1 0
0	1	1	1	1 1 1 1 1



# Princípios de Operação de Memória



Os três barramentos são necessários para permitir que a CPU escreva e leia dados na memória.

# Operação escrita

- CPU fornece endereço e coloca informação no barramento de endereço.
- Decodificador de endereço habilita o pino CS da memória desejada.
- CPU coloca os dados a serem armazenados no barramento de dados.
- CPU ativa as linhas de controle para operação de gravação
- CIs de memória decodificam o endereço e armazenam o dado

# Operação leitura

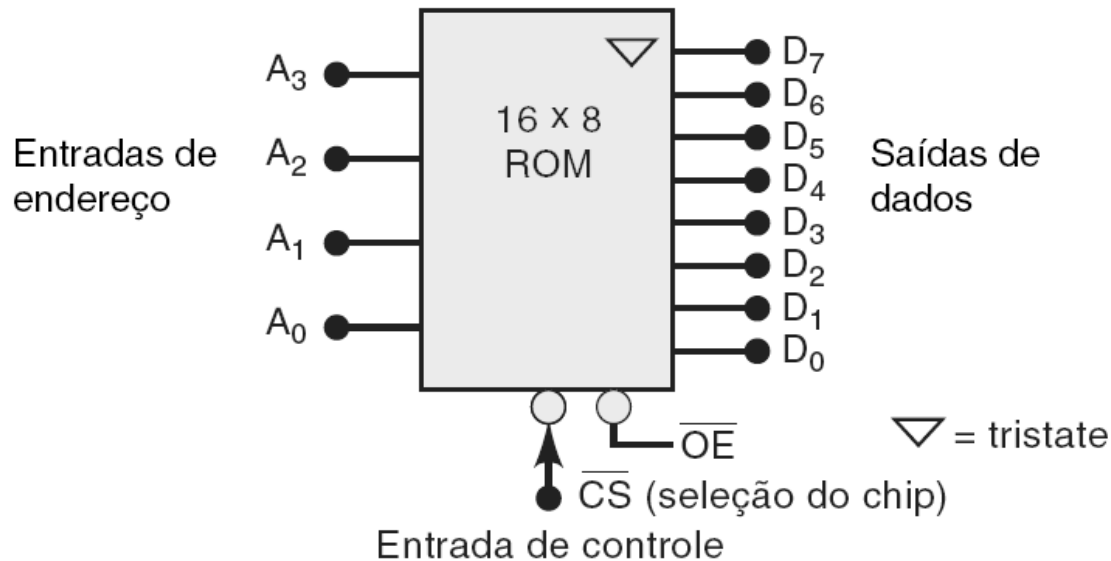
- CPU fornece endereço e coloca informação no barramento de endereço.
- Decodificador de endereço habilita o pino CS da memória desejada.
- CPU ativa as linhas de controle para operação de leitura
- CIs de memória decodificam o endereço e recuperam o dado
- CIs colocam os dados no barramento de dados para leitura pela CPU

# Tipos de Memória

- Memória apenas de leitura – ROM
  - MROM
  - PROM
  - EPROM
  - EEPROM
- Memória FLASH
- Memória RAM
  - DRAM
  - SRAM
  - FRAM

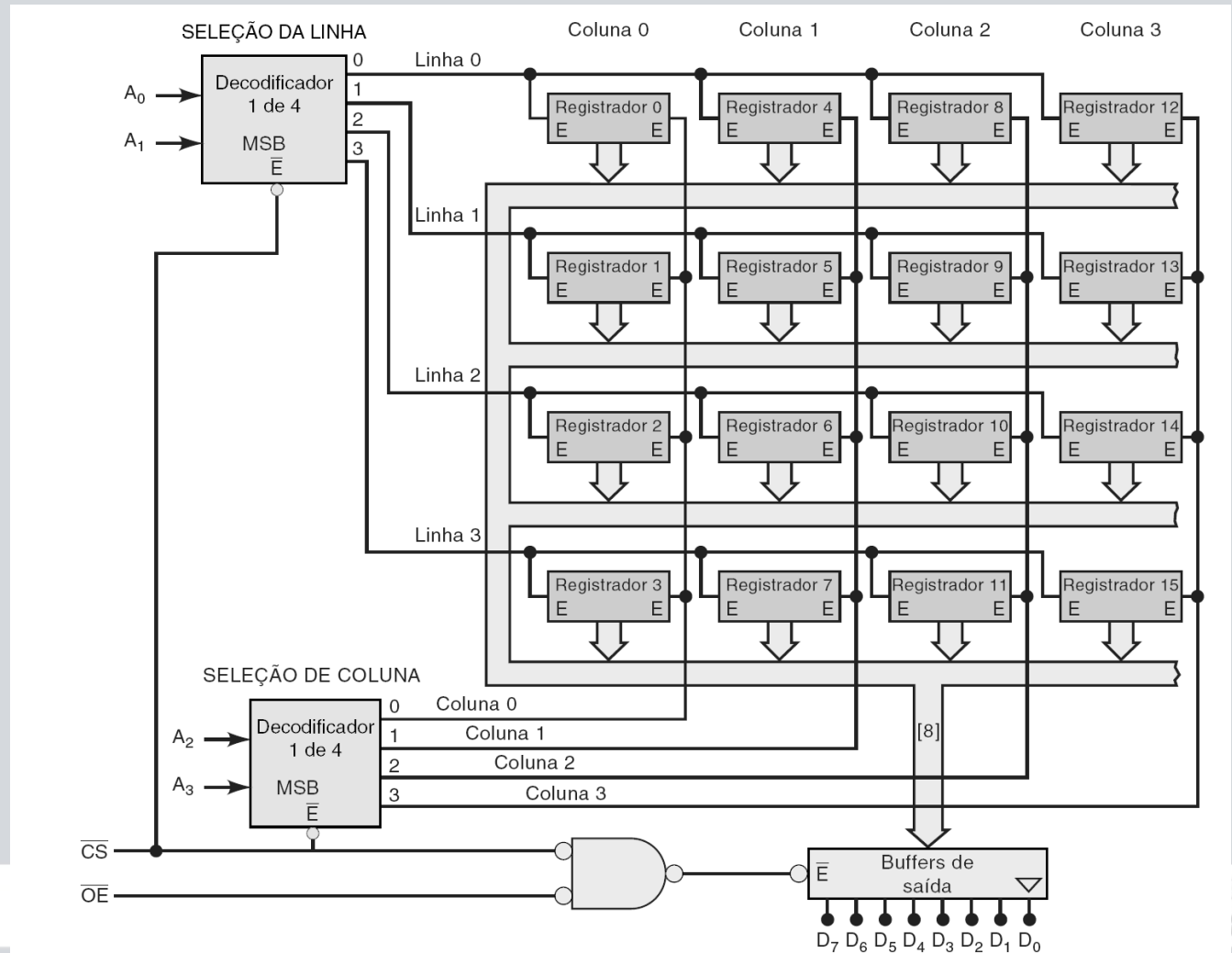
# Memória Apenas de Leitura - ROM

- Manter dados permanentemente ou que não mudam com frequência.
- ROMs são não voláteis

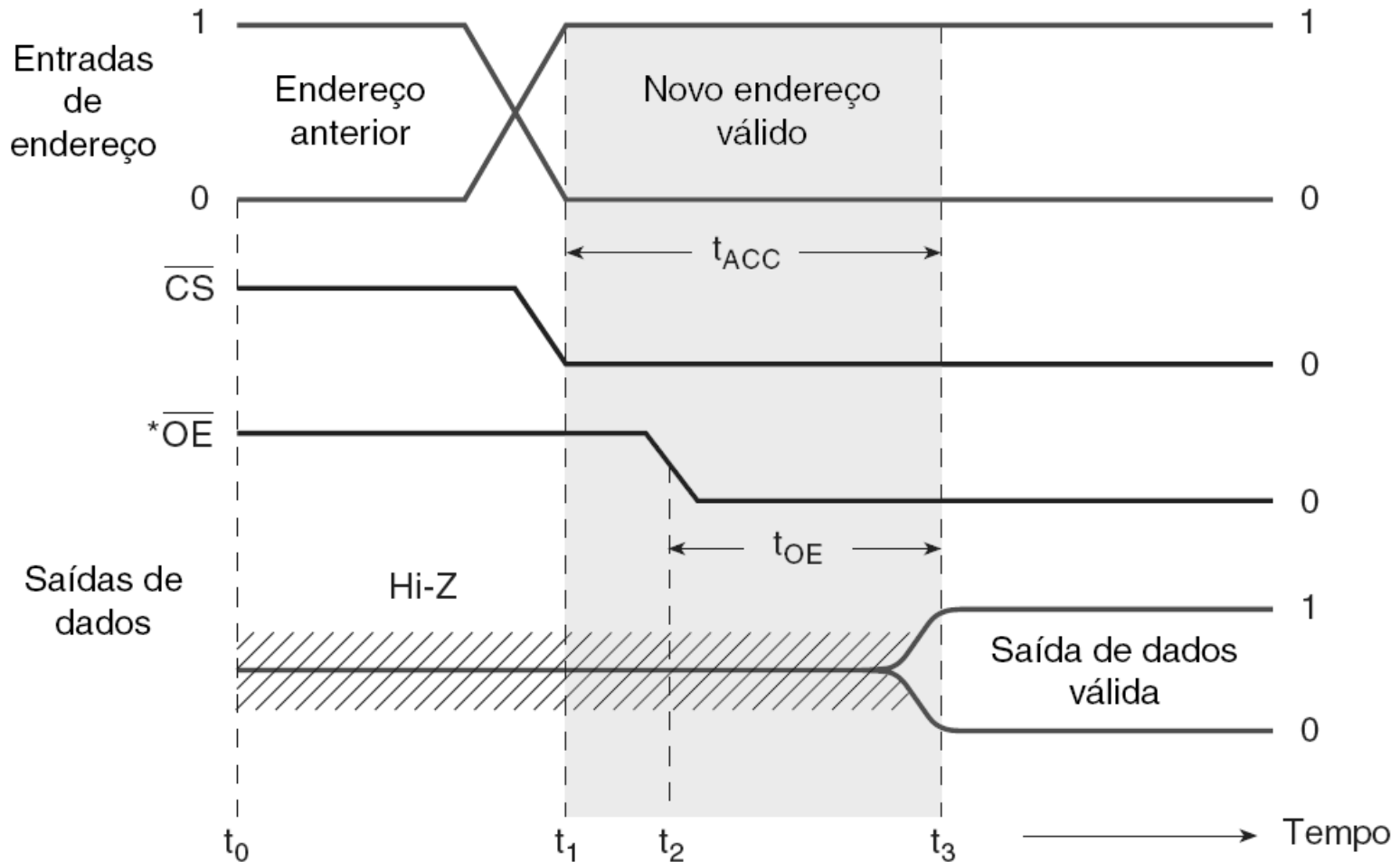


# Memória Apenas de Leitura - ROM

- Arquitetura de uma ROM 16 x 8 - cada registro armazena uma palavra de 8 bits:

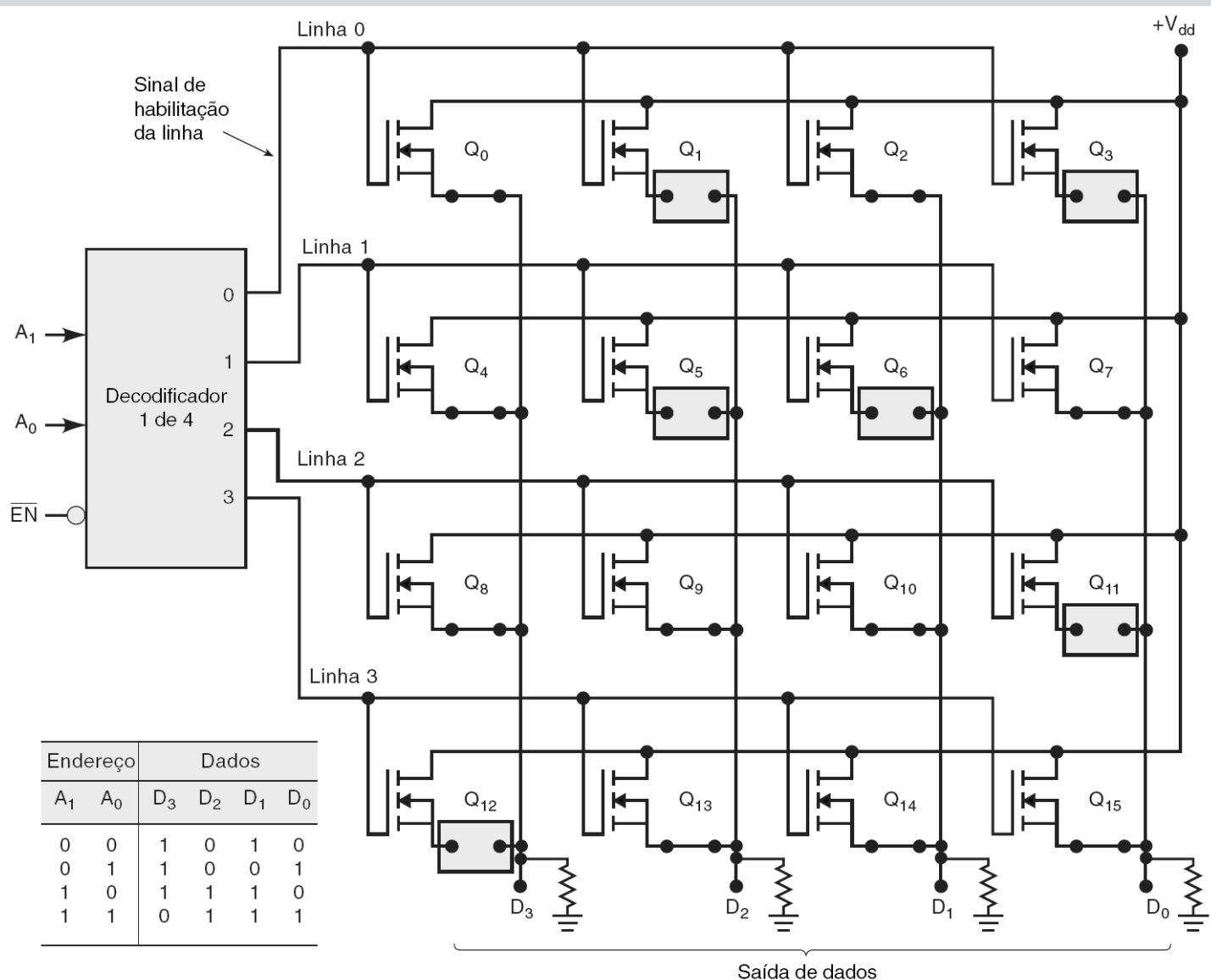


# Temporização de ROM



\* $t_{OE}$  é medido a partir do momento em que  $\overline{CS}$  e  $\overline{OE}$  foram ambos ativados.

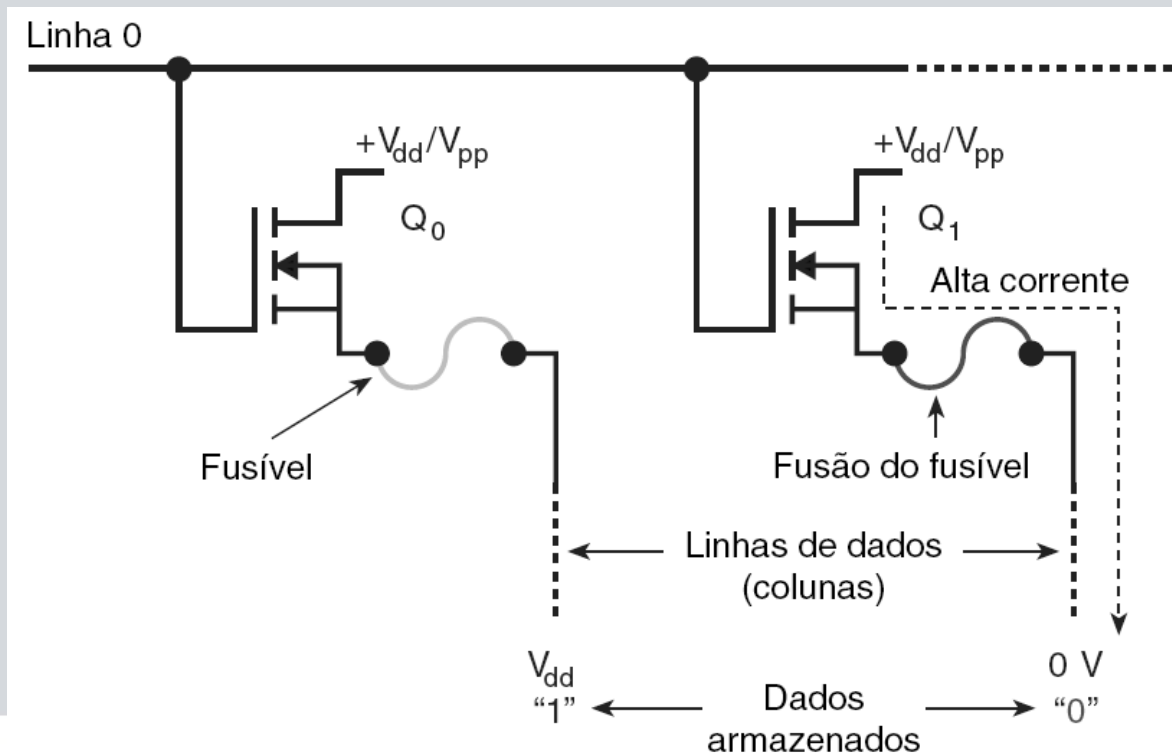
# MROM – ROM por Máscara



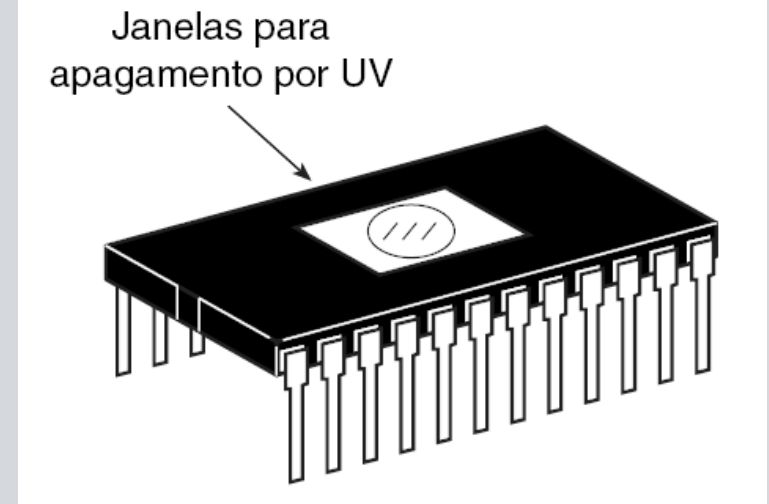
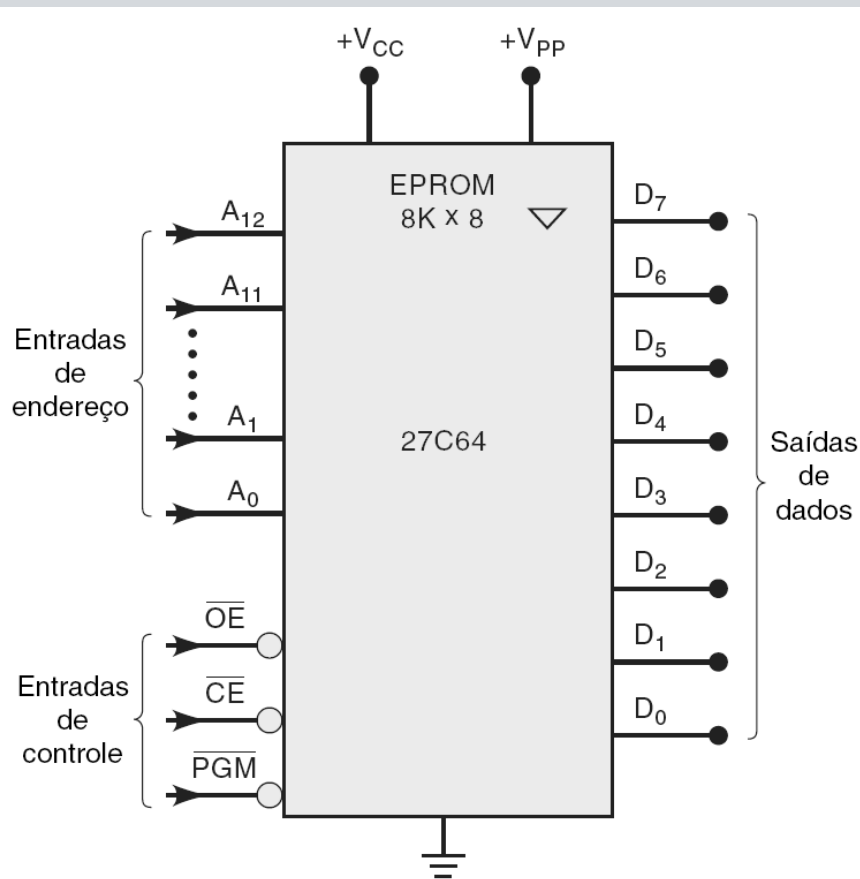


# PROMs – ROMs Programáveis

- ROM's programáveis uma única vez.



# EPROM – Erasable Programmable ROM



# EEPROM – Elettrically Erasable PROM

- Usa eletricidade para apagar os dados.
- Pode apagar e reescrever bytes individuais
- Processo interno de armazenamento de um valor é lento, velocidade da operação de transferência também é lento.

# Memória Flash

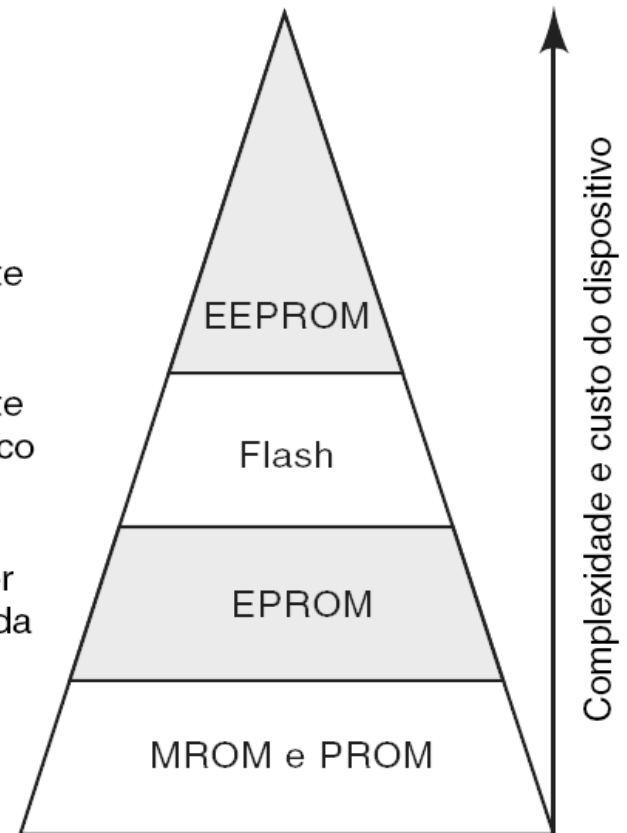
- Uma célula de memória flash é como a célula EPROM simples de um único transistor com um custo consideravelmente menor que de EEPROM.

Pode ser apagada eletricamente no circuito, byte a byte

Pode ser apagada eletricamente no circuito, por setor ou em bloco (todas as células)

Pode ser apagada em bloco por luz UV, apagada e reprogramada fora do circuito

Não pode ser apagada e reprogramada



# Memória Flash

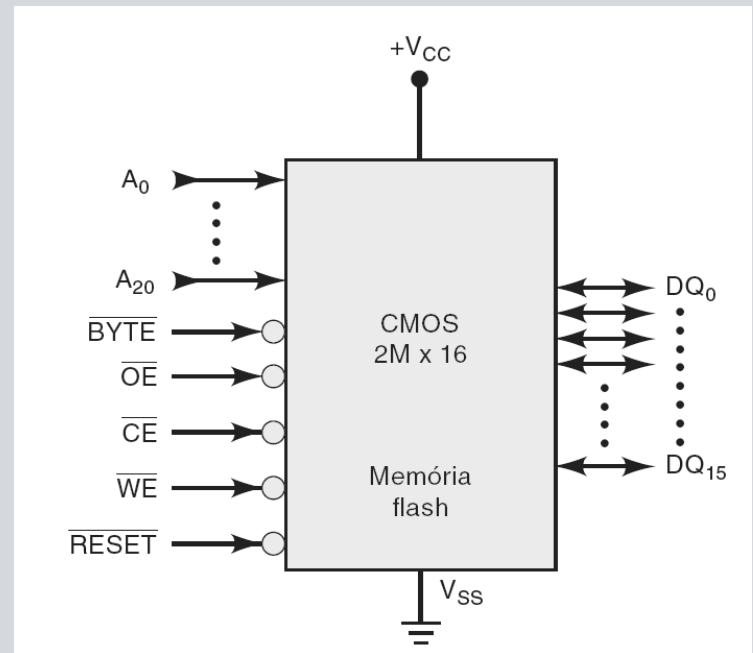
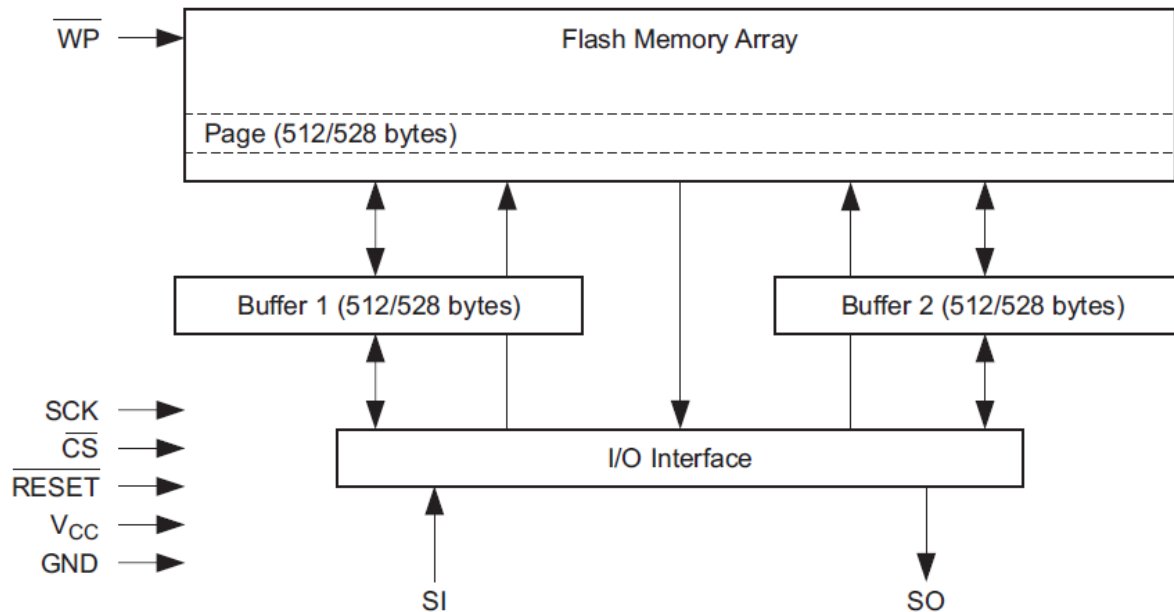
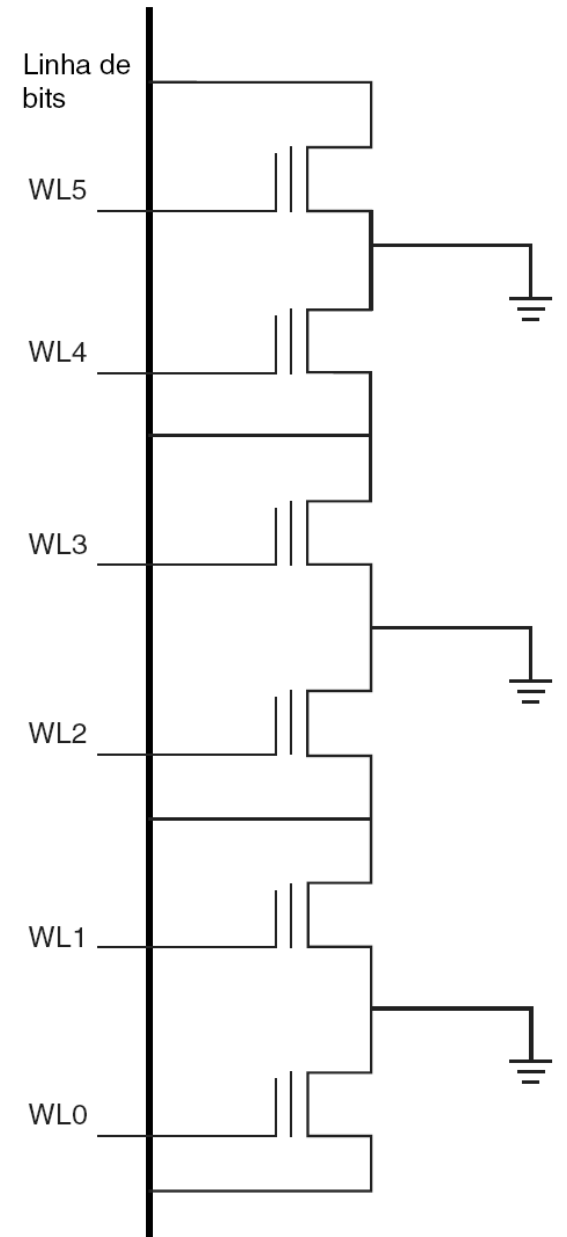


Figure 2-1. Block Diagram



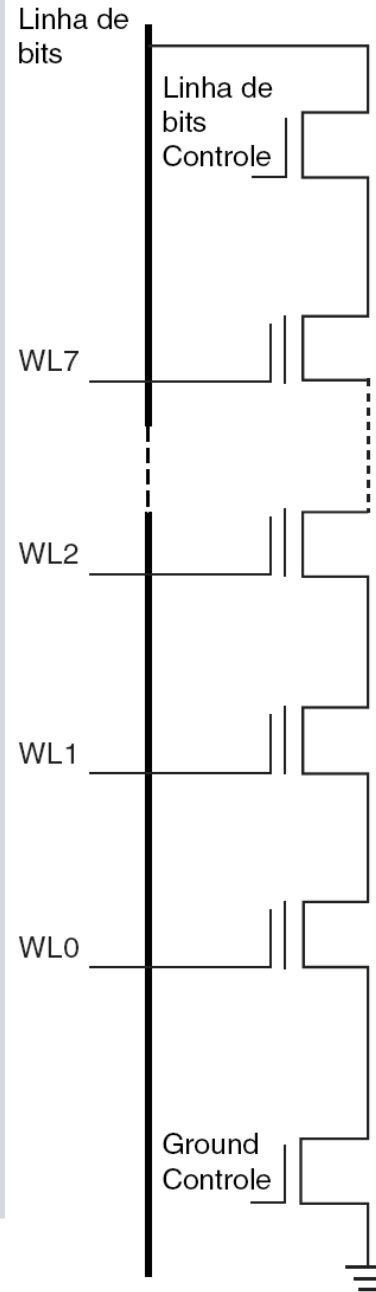
# Memória Flash

- Baseados nas EEPROMs usavam flash NOR, cada transistor pode ser lido ou escrito individualmente.



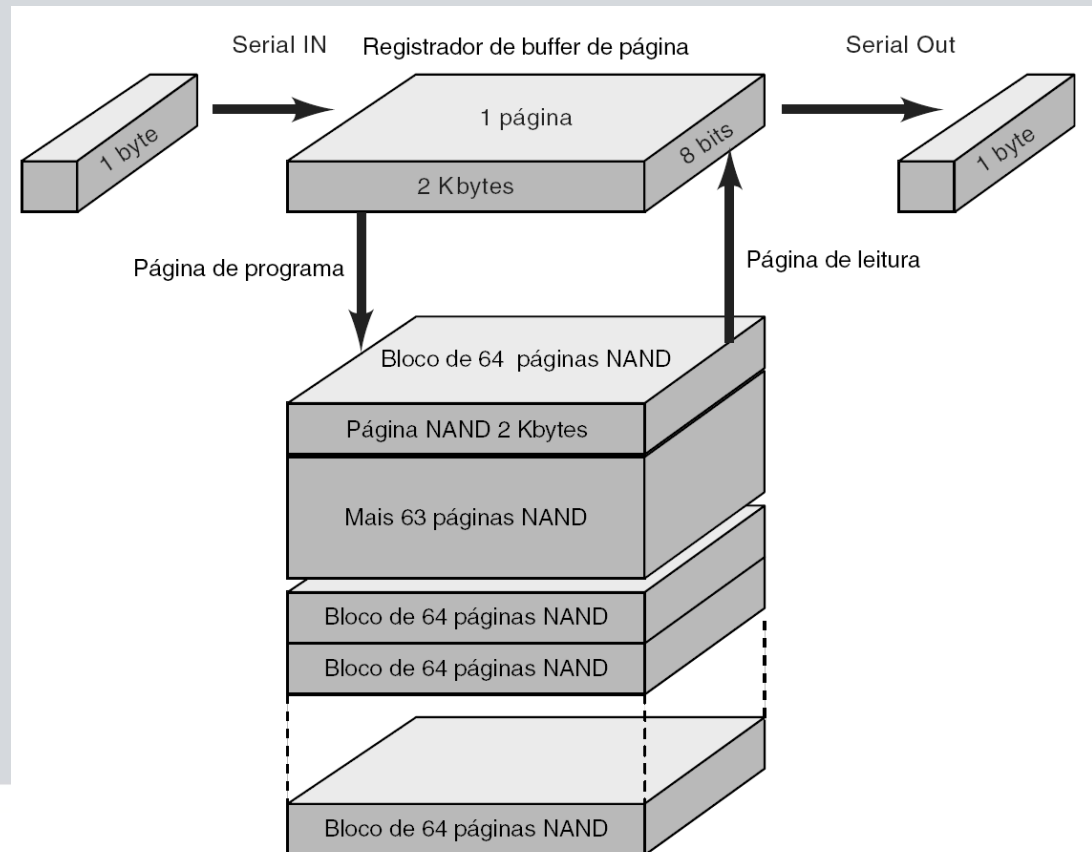
# Memória Flash

- Aumento na densidade resultou em flash NAND, mas obrigada a leitura e escrita dos dados de forma conjunta com outros bits.



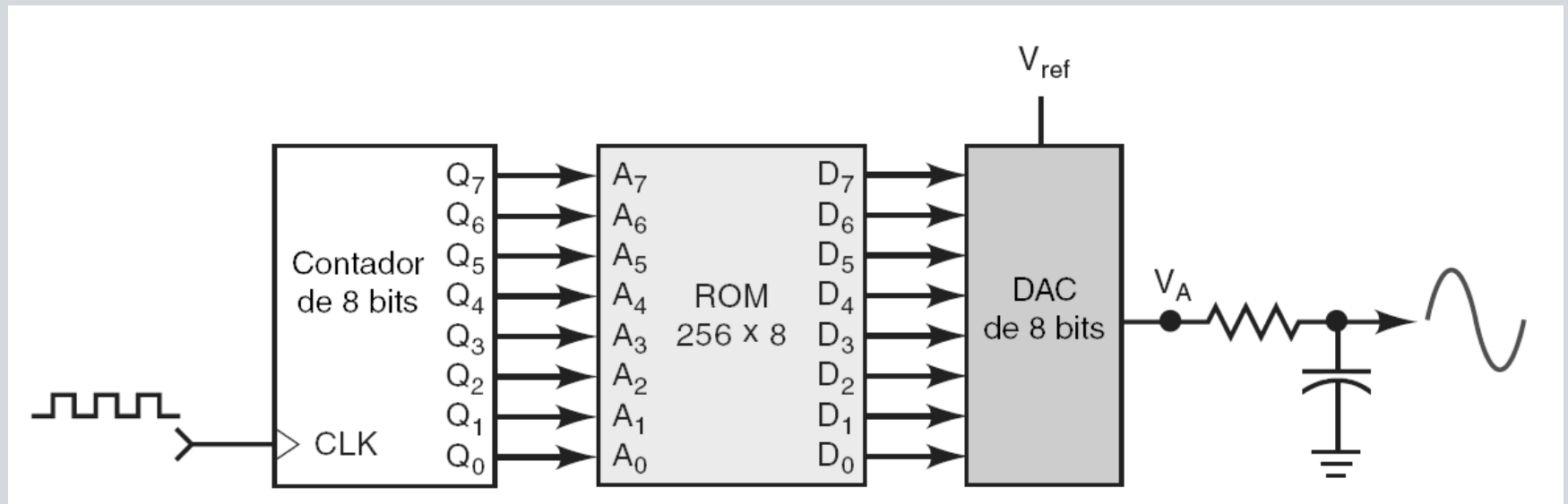
# Memória Flash

- Flash NAND proporciona apagamento rápido e tempo de programação curto, mas precisa tratar os dados em blocos.



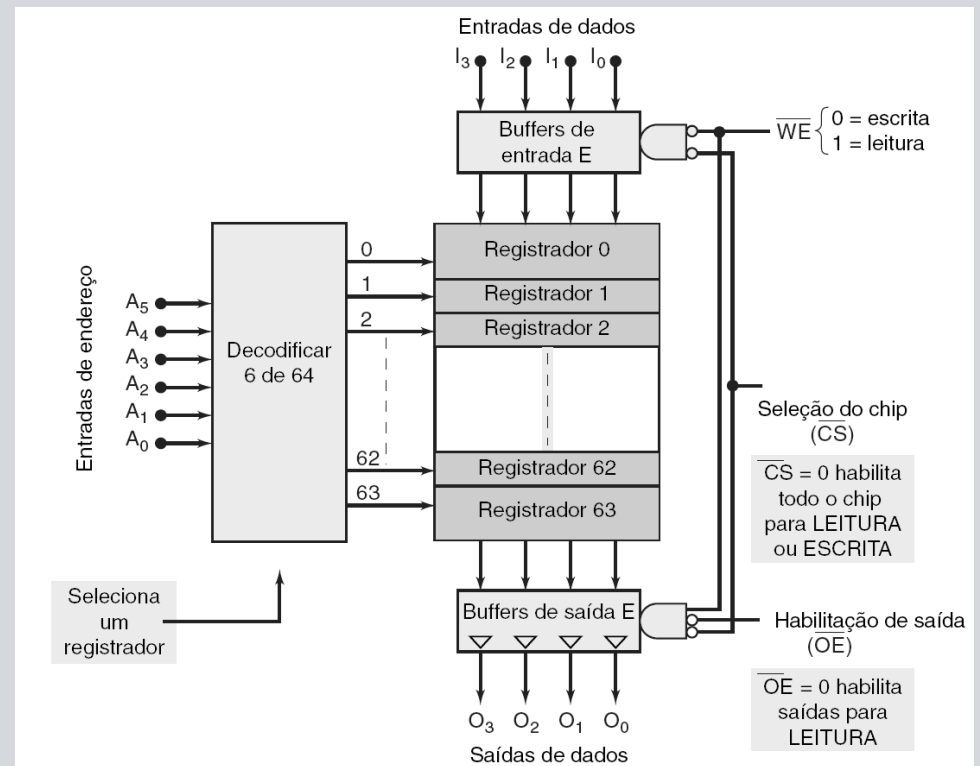
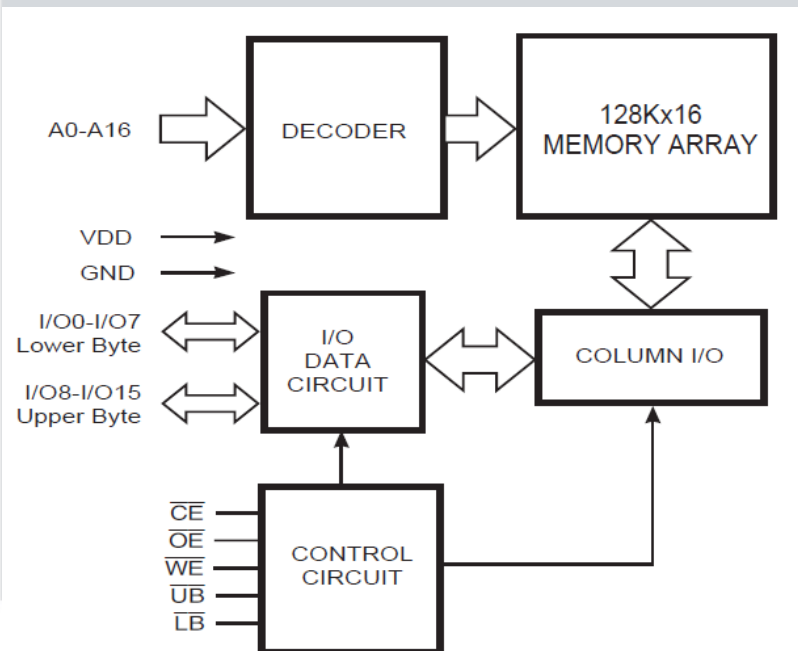


# Aplicação das ROMs



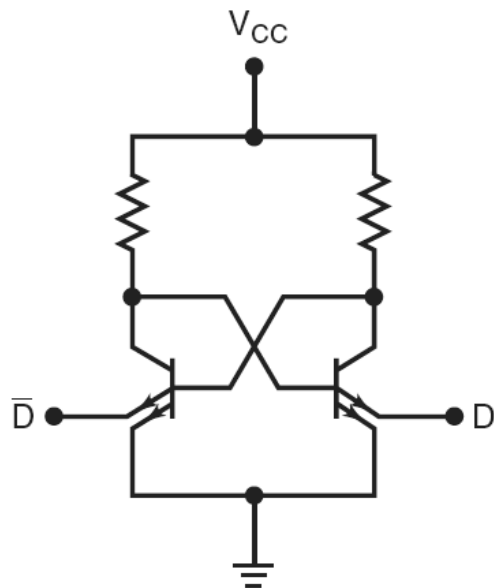
# Memória RAM

- *Random access memory* – memória de acesso aleatório.
- É volátil.

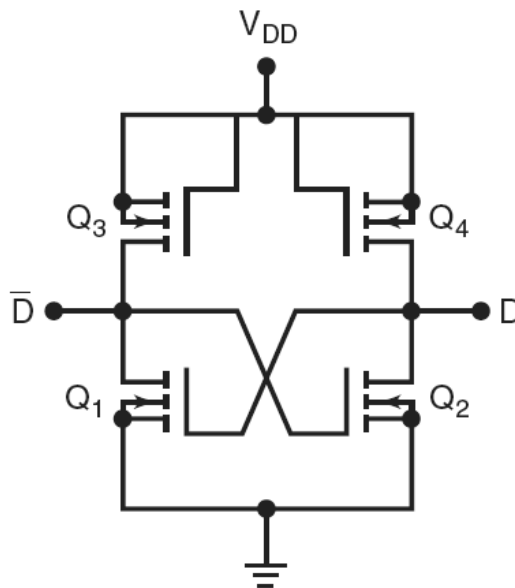


# Memória RAM - SRAM

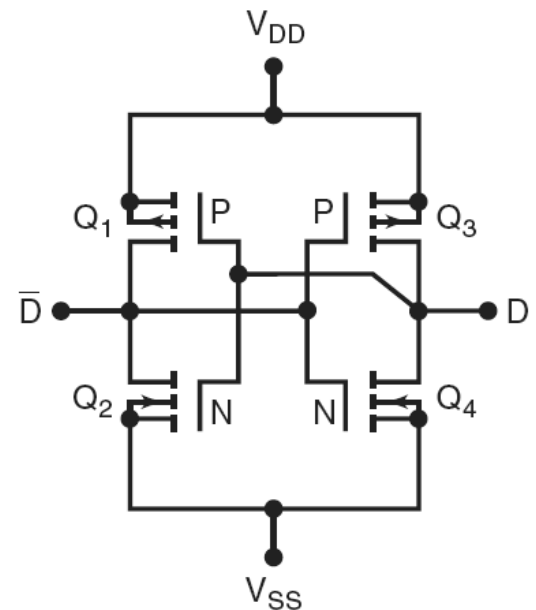
- Static RAM – RAM estática
- Flip-flops, ficam em um determinado estado indefinidamente, desde que a energia do sistema não seja interrompida.



Célula bipolar

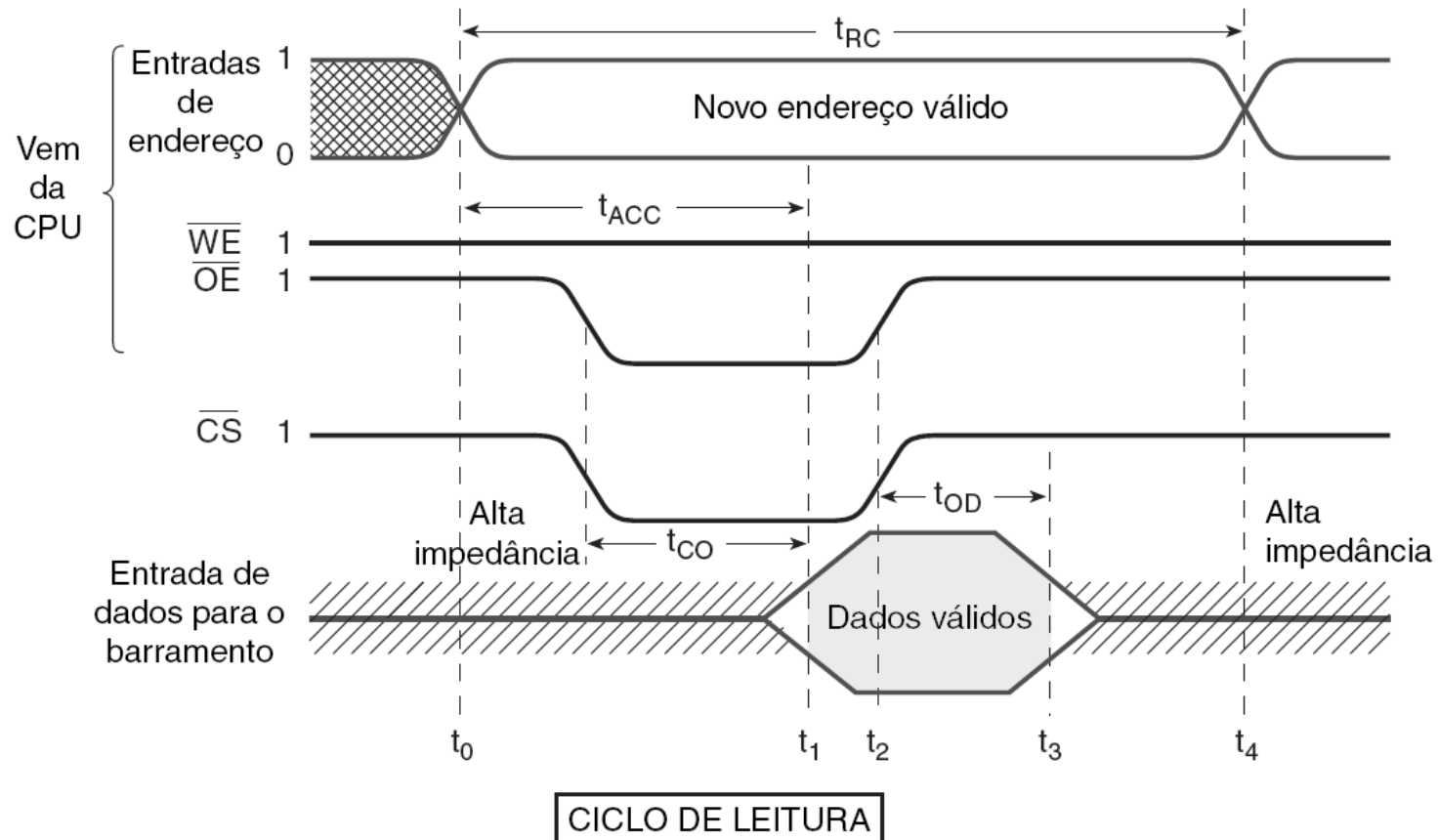


Célula NMOS

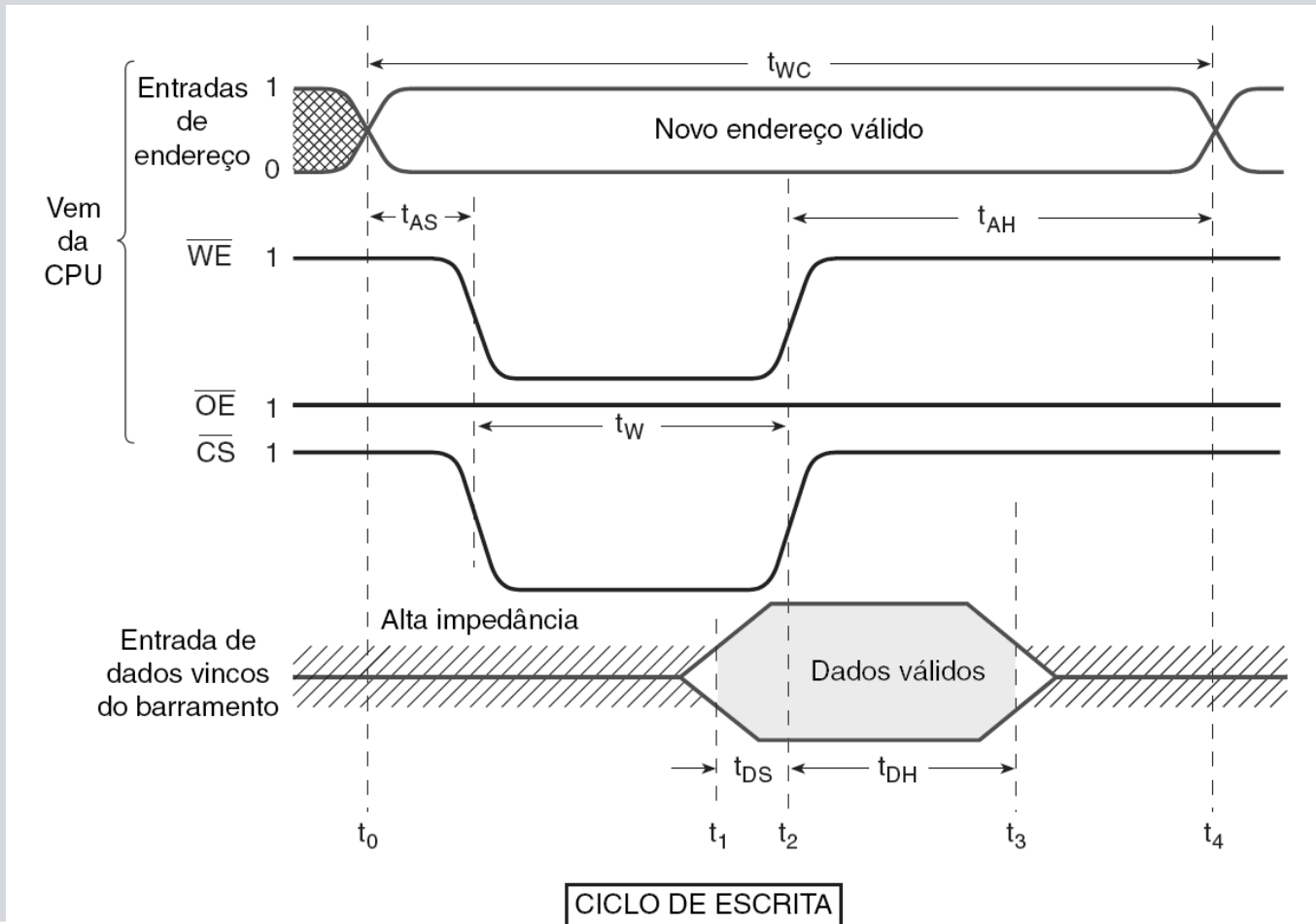


Célula CMOS

# Memória RAM - SRAM



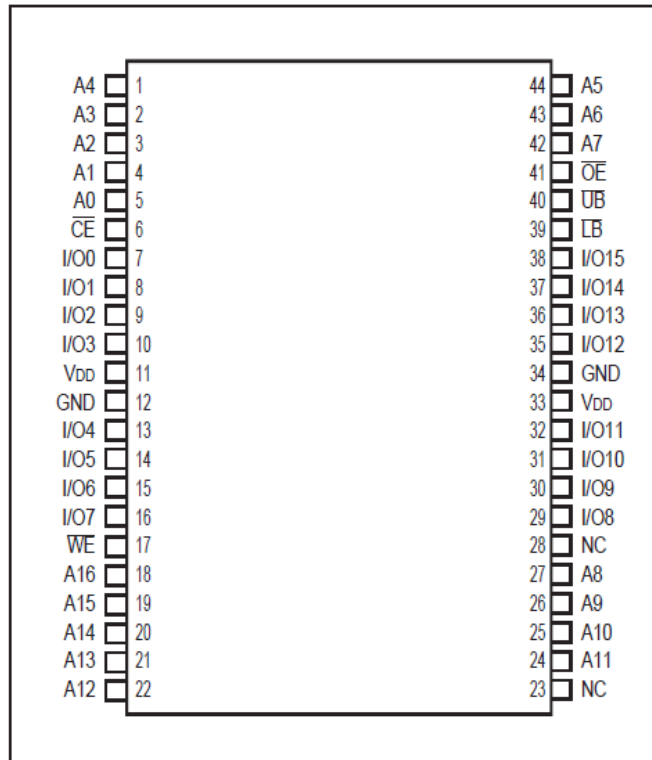
# Memória RAM - SRAM



# SRAM – IS61LV12816L

- 128K x16
- Access Time 8 ou 10 ns

## PIN CONFIGURATION 44-Pin TSOP (Type II) (T)



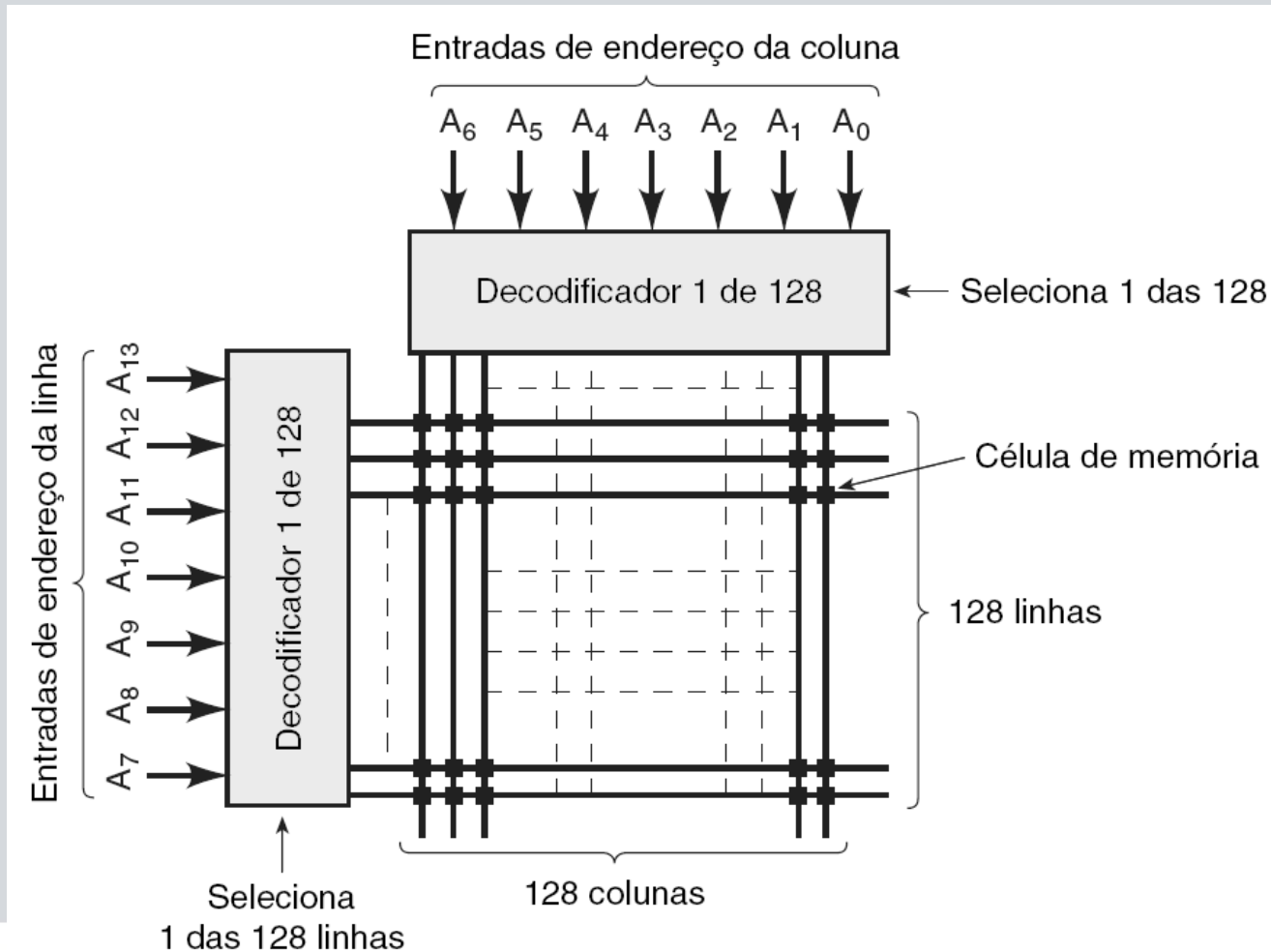
## PIN DESCRIPTIONS

A0-A16	Address Inputs
I/O0-I/O15	Data Inputs/Outputs
$\overline{CE}$	Chip Enable Input
$\overline{OE}$	Output Enable Input
$\overline{WE}$	Write Enable Input
$\overline{LB}$	Lower-byte Control (I/O0-I/O7)
$\overline{UB}$	Upper-byte Control (I/O8-I/O15)
NC	No Connection
VDD	Power
GND	Ground

# Memória RAM - DRAM

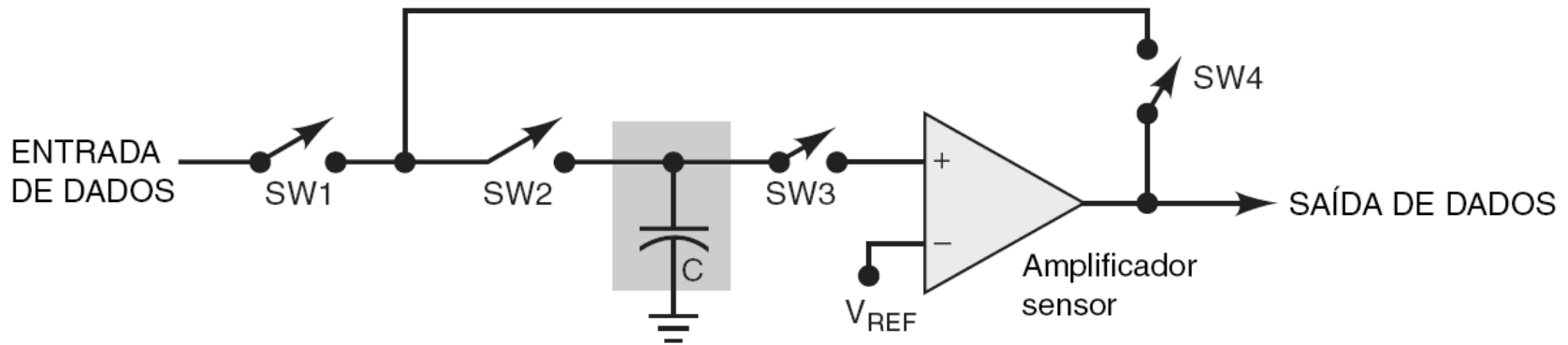
- Armazena dados como cargas em capacitores, que gradualmente desaparecem devido a descarga do capacitor.
- É necessário dar recargas (refresh) nos dados periodicamente, através da recarga dos capacitores, a cada, 2, 4, ou 8 ms.
- Têm capacidades muito maiores e consumo de energia muito menor.

# Memória RAM - DRAM





# Memória RAM - DRAM

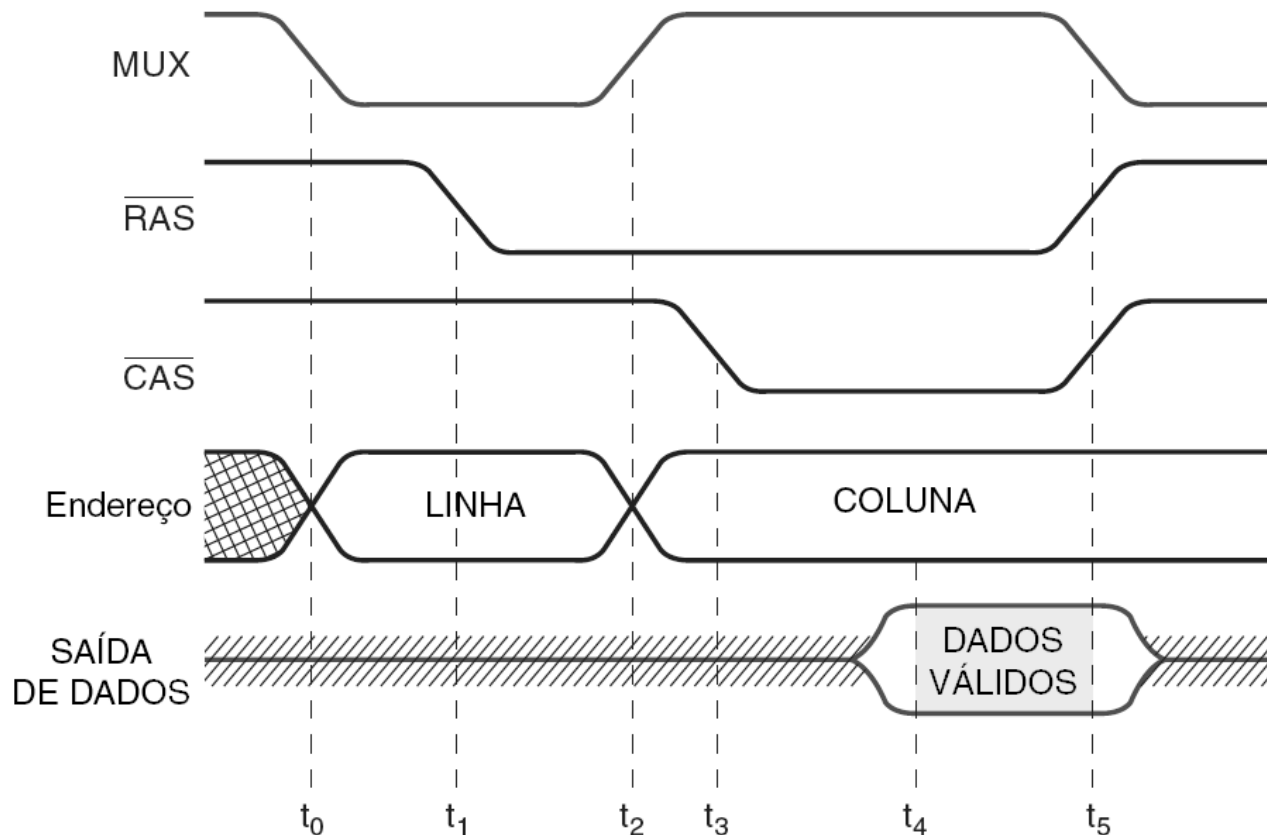


Durante uma operação de escrita, as chaves SW1 e SW2 são fechadas.  
Durante uma operação de leitura, todas as chaves são fechadas, exceto SW1.

# Memória RAM - DRAM

- Para reduzir a quantidade de pinos nas DRAM de alta capacidade os fabricantes utilizam a multiplexação de endereços:
- Cada pino de entrada de endereço acomoda dois bits de endereço diferentes.
- No endereçamento multiplexado, o endereço é aplicado em duas partes, o de linha e o de coluna:
- Ele é conectado diretamente aos registradores de linha e coluna.
- O registrador de linha armazena a parte alta do endereço e o de coluna, a baixa.
- O strobe de endereço de linha (row address strobe, RAS) armazena os conteúdos das entradas de endereço no registro de endereço de linha.
- O strobe de endereço de coluna (column address strobe, CAS) armazena os conteúdos das entradas de endereço no registro de endereço de coluna.

# Memória RAM - DRAM



$t_0$ : MUX é acionado em nível BAIXO para aplicar os bits de endereço da linha ( $A_8$  a  $A_{15}$ ) nas entradas de endereço da DRAM.

$t_1$ : a entrada  $\overline{RAS}$  é acionada em nível BAIXO para carregar o endereço de linha na DRAM.

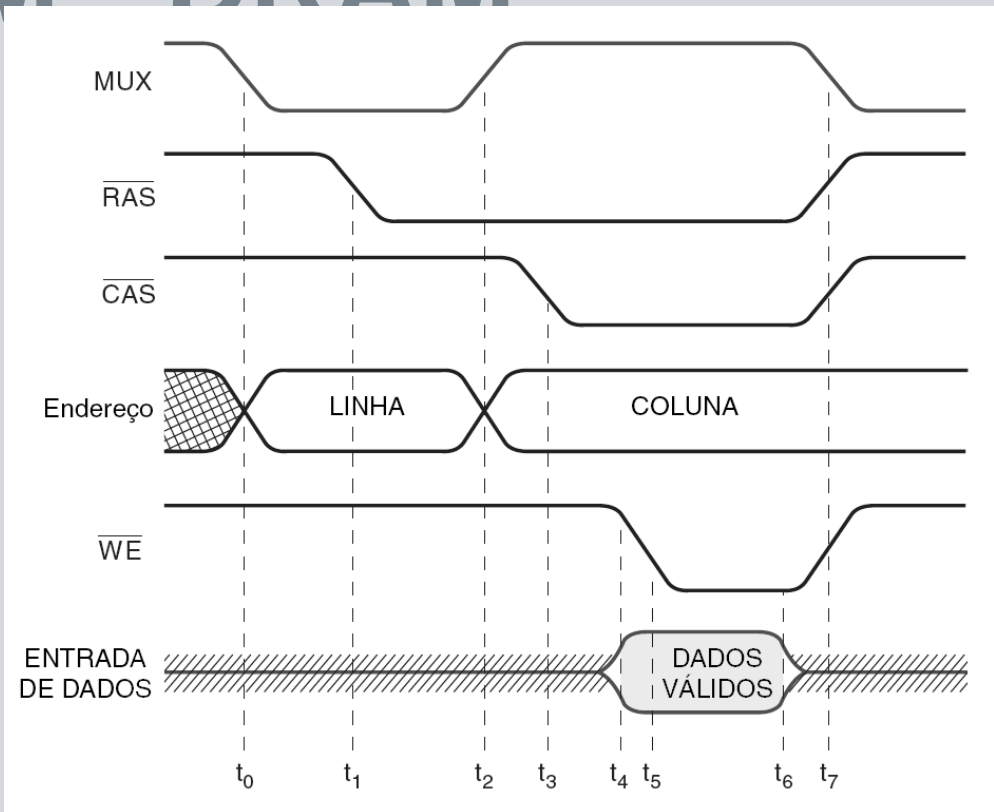
$t_2$ : MUX vai para nível ALTO para colocar o endereço de coluna ( $A_0$  a  $A_7$ ) nas entradas de endereço da DRAM.

$t_3$ : a entrada  $\overline{CAS}$  vai para nível BAIXO para carregar o endereço da coluna na DRAM.

$t_4$ : a DRAM responde colocando dados válidos, provenientes da célula de memória selecionada, na linha de saída de dados (DATA OUT).

$t_5$ : os sinais MUX,  $\overline{RAS}$  e  $\overline{CAS}$ , e DATA OUT retornam para os estados iniciais.

# Memória RAM - DRAM



- $t_0$ : o nível BAIXO em  $MUX$  coloca o endereço da linha nas entradas da DRAM.  
 $t_1$ : a borda de descida em  $\overline{RAS}$  carrega o endereço da linha na DRAM.  
 $t_2$ :  $MUX$  vai para nível ALTO para colocar o endereço da coluna nas entradas da DRAM.  
 $t_3$ : a borda de descida em  $\overline{CAS}$  carrega o endereço da coluna na DRAM.  
 $t_4$ : o dado a ser escrito é colocado na linha de entrada de dados (DATA IN).  
 $t_5$ : a entrada  $\overline{WE}$  é pulsada em nível BAIXO para escrever o dado na célula selecionada.  
 $t_6$ : os dados de entrada são removidos de DATA IN.  
 $t_7$ : os sinais  $MUX$ ,  $\overline{RAS}$ ,  $\overline{CAS}$  e  $\overline{WE}$  retornam a seus estados iniciais.

# IS43/46TR16128A

- 128M x16 – 2 Gb DDR3 SDRAM 933MHz

	1	2	3	4	5	6	7	8	9
A	VSS	VDD	NC				NU/TDQS#	VSS	VDD
B	VSS	VSSQ	DQ0				DM/TDQS	VSSQ	VDDQ
C	VDDQ	DQ2	DQS				DQ1	DQ3	VSSQ
D	VSSQ	DQ6	DQS#				VDD	VSS	VSSQ
E	VREFDQ	VDDQ	DQ4				DQ7	DQ5	VDDQ
F	NC <sup>1</sup>	VSS	RAS#				CK	VSS	NC
G	ODT	VDD	CAS#				CK#	VDD	CKE
H	NC	CS#	WE#				A10/AP	ZQ	NC
J	VSS	BA0	BA2				NC(A15)	VREFCA	VSS
K	VDD	A3	A0				A12/BC#	BA1	VDD
L	VSS	A5	A2				A1	A4	VSS
M	VDD	A7	A9				A11	A6	VDD
N	VSS	RESET#	A13				A14	A8	VSS



# FRAM – Ferroelectric RAM

- Similar in construction to [DRAM](#) but uses a [ferroelectric](#) layer instead of a [dielectric](#) layer to achieve non-volatility.
- FeRAM advantages over flash include:
  - lower power usage,
  - faster write performance<sup>[1]</sup> and
  - a much greater maximum number of write-erase cycles (exceeding  $10^{16}$  for 3.3 V devices).
- Disadvantages of FeRAM are:
  - much lower [storage densities](#) than flash devices,
  - storage capacity limitations, and
  - higher cost.

# FRAM – Ferroelectric RAM

## FM23MLD16

*8Mbit F-RAM Memory*

**RAMTRON**

### Features

#### 8Mbit Ferroelectric Nonvolatile RAM

- Organized as 512Kx16
- Configurable as 1Mx8 Using /UB, /LB
- High Endurance 100 Trillion ( $10^{14}$ ) Read/Writes
- NoDelay™ Writes
- Page Mode Operation to 33MHz
- Advanced High-Reliability Ferroelectric Process

#### SRAM Compatible

- JEDEC 512Kx16 SRAM Pinout
- 60 ns Access Time, 115 ns Cycle Time

#### Advanced Features

- Low  $V_{DD}$  Monitor Protects Memory against Inadvertent Writes

#### Superior to Battery-backed SRAM Modules

- No Battery Concerns
- Monolithic Reliability
- True Surface Mount Solution, No Rework Steps
- Superior for Moisture, Shock, and Vibration

#### Low Power Operation

- 2.7V – 3.6V Power Supply
- 14 mA Active Current

#### Industry Standard Configuration

- Industrial Temperature -40° C to +85° C
- 48-pin “Green”/RoHS FBGA package

• \$65 – 8Mbit

• IS43TR16128AL-125KBL - \$11 – 2Gbit

# O que não pode ser dúvida:

Computers & Internet > Software

Next >



Heyy! If I install Windows 7 32-bit twice, will it make it 64-bit?  
PLease help me, im all confused..?

Heyy! If I install Windows 7 32-bit twice, will it make it 64-bit? PLease help me, im all confused..

I wanted to upgrade to 64 bit, but then I came up with an idea of installing 32-bit twice, so it will make it 64 bit. Is it so? If it is then I'll save a lot of money..

it is cause  $2 \times 32 = 64$  :)

☆ Follow  7 answers



# O que não pode ser dúvida:

**Intel® Core™ i7-4770R Processor**  
(6M Cache, up to 3.90 GHz)

Especificações

Essenciais

Desempenho

Especificações de memória

Especificações gráficas

Opções de expansão

Especificações de pacote

Tecnologias avançadas

Intel® Data Protection Technology

Tecnologia de Proteção de Plataforma Intel®

Produtos compatíveis

Pedidos / sSpecs / Escalonamentos

Especificações

Essenciais

Status	Launched
Data de introdução	Q2'13
Número do processador	i7-4770R
Cache inteligente Intel®	6 MB
DMI2	5 GT/s
Nº de links de QPI	0
Conjunto de instruções	64-bit
Extensões do conjunto de instruções	SSE4.1/4.2, AVX 2.0
Opções integradas disponíveis	No
Litografia	22 nm
Escalabilidade	1S Only
Preço recomendado para o cliente	TRAY: \$358.00
Ficha técnica	Link

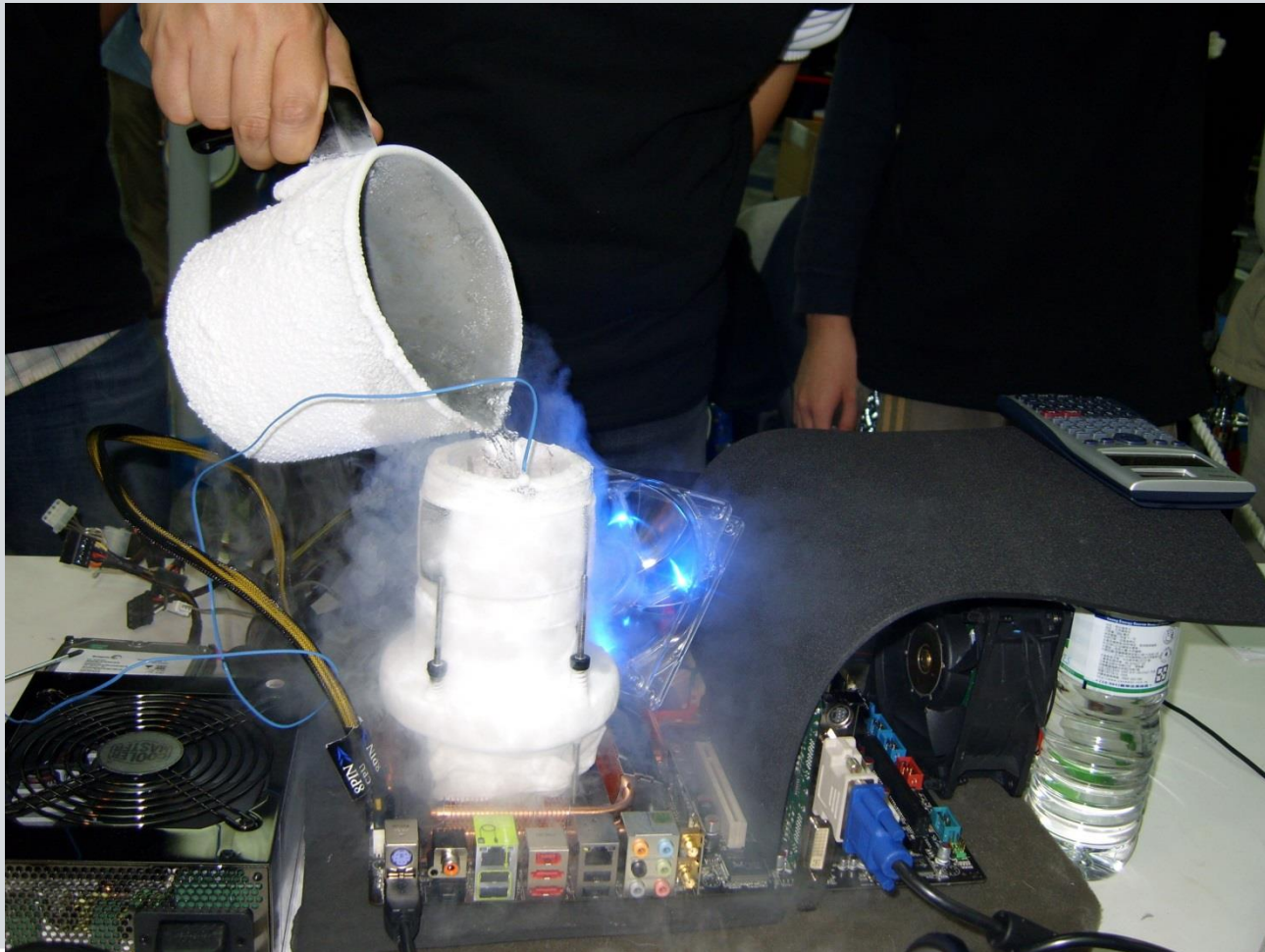
Desempenho

Número de núcleos	4
Nº de threads	8
Frequência baseada em processador	3.2 GHz
Frequência turbo max	3.9 GHz
TDP	65 W

Especificações de memória

Tamanho máximo de memória (de acordo com o tipo de memória)	32 GB
Tipos de memória	DDR3L 1333/1600

# O que não pode ser dúvida:



# O que não pode ser dúvida

- Posso rodar Windows no meu iMac?



# O que não pode ser dúvida:

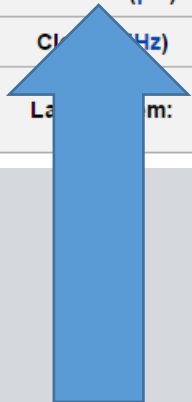


# O que não pode ser dúvida:

## Modelos [\[ editar | editar código-fonte \]](#)

Os primeiros Pentiums foram fabricados com uma técnica de 0.8 microns, trabalhavam com clocks de 60 MHz e de 66 MHz<sup>2</sup> e foram considerados algo problemáticos devido a problemas de aquecimento. Mais tarde, foram surgindo gradualmente versões de 75, 90, 120, 133, 150, 166, 200, e 233 MHz. Versões de 266 e 300 MHz foram posteriormente lançadas para uso em computadores portáteis. Processadores **Pentium OverDrive** foram lançados com velocidades de 63 e 83 MHz como uma opção de **upgrade** para computadores 486 mais antigos.

Nome-Código	P5		P54				P54C				P55C			P55C (Tillamook)			
Processo de fabricação utilizado (µm)	0.80		0.60				0.35							0.25			
Clock (MHz)	60	66	75	90	100	120	133	150	166	200	166	200	233	200	233	266	300
Lançamento:	Março de 1993		Out. de 1993	Março de 1994		Março de 1995	Junho de 1995	Jan. de 1996	Junho de 1996		Out. de 1996	Junho de 1997		Set. de 1997	Jan. de 1998		Jan. de 1999





# O que não pode ser dúvida:

```
SHL    DX,1
SHL    DX,1
AND    DI,0FH
JMP     SHORT PACKIN

ALIGNED:
AND    DI,0F000H

PACKIN:
OR     DI,DX
MOV    [BX],DI
RET

DEVNAME:
MOV    SI,OFFSET DOSGROUP:IONAME      ;List of I/O devices with file names
MOV    BH,NUMDEV                      ;BH = number of device names

LOOKIO:
MOV    DI,OFFSET DOSGROUP:NAME1
MOV    CX,4                          ;All devices are 4 letters
REPE   CMPSB                          ;Check for name in list
JZ     IOCHK                          ;If first 3 letters OK, check for the rest
ADD    SI,CX                          ;Point to next device name
DEC    BH
JNZ    LOOKIO

CRET:
STC                                       ;Not found
RET

IOCHK:
IF     IBM
CMP    BH,NUMDEV                        ;Is it the first device?
JNZ    NOTCOM1
MOV    BH,2                            ;Make it the same as AUX

NOTCOM1:
ENDIF
NEG    BH
MOV    CX,2                            ;Check rest of name but not extension
MOV    AX,2020H
REPE   SCASW                          ;Make sure rest of name is blanks
```

# Próxima aula:

- Arquitetura de Microcontroladores.

## **Anderson Wedderhoff Spengler**

E-mail: [anderson.spengler@ufsc.br](mailto:anderson.spengler@ufsc.br)

Telefone: +55 (48) 3721 7489



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA