

# Memories

V.2.0 V.Lobo 2016

Memories

## Memories

1

Memories

## Introduction

- Purpose of of memory system
  - STORE DATA (0's and 1's)
  - Basic Operations - WRITE a datum; READ a datum
- To store 1 bit only 1 flip-flop is needed
- To store many bits, a **set of flip-flops** is necessary, and these some sort of **organization**.
  - To access a given content, one must give the ADDRESS where it is stored inside de memory, and whether we want to WRITE it or READ it.

The diagram illustrates the relationship between logical and physical memory organization. On the left, 'Logical Organization' is shown as a table with four rows. The first column is labeled 'Addresses' and contains values 0, 1, 2, and 3. The second column is labeled 'Contents' and contains values 0, 0, 1, and 0. An arrow points from this table to the right, where 'Physical Organization (integrated circuits)' is shown. It features a central box representing the memory chip. Two lines labeled 'Addresses' enter from the left, with the top one labeled  $A_0$  and the bottom one  $A_1$ . Two lines labeled 'Contents' enter from the bottom, with the left one labeled 'Data'. Two lines labeled 'CS' and 'WE' enter from the right.

2

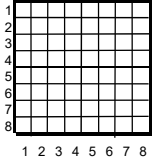
# Memories

V.2.0 V.Lobo 2016

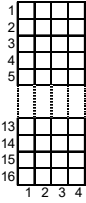
Memories

## Address and memory width


- Data can be stored in "boxes" (addresses) with 1 bit, or various bits
  - *Width* of a memory is the number of bits stored in each address  
→ Also known as *word size* of the machine.
  - A " $n \times m$ " memory has  $n$  addresses, each with  $m$  bits
  - Most memories use 8 bits (1 byte) words.
- Example: 64 bits may be organized as ...



8x8



16x4



64x1


3

Memories

## Types and sizes

- The basic unit is the BIT (BI-nary digiT)
  - 1 Byte = 8 Bits
  - 1 K (Kilobyte) =  $2^{10} = 1024$
  - 1 M (Megabyte) =  $2^{20} = 1.048.576$
  - 1 G (Gigabyte) =  $2^{30} = 1.073.741.824$
  - 1 T (Terabyte) =  $2^{40} = 1.099.511.627.776$
- Many different technologies and applications
  - Solid state (integrated circuits)
    - RAM, ROM, SRAM, DRAM, PROM, EPROM, etc
  - Magnetic and optical (mostly discs)
    - hard drives, floppy, drums, cartridges, tapes
    - Compact Disks (CD), DVD, BluRay Disks, etc
  - Capacities (2016)
 

→ DRAM	1MB - 8 GB +
→ SRAM, PROMs	1K - 32MB +
→ HD	500GB - 4TB +



4

# Memories

V.2.0 V.Lobo 2016

## RAM Memories

Memories

- **RAM- Random Access Memories (static memories)**
  - You can randomly select any address (versus sequential access)
  - Built with flip-flops
  - They contain:
    - **Address decoding block**
    - **Memory elements block**
    - **Control block**

5

## Internal organization of a RAM

Memories

6

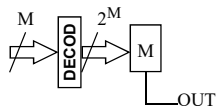
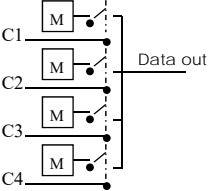
# Memories

V.2.0 V.Lobo 2016

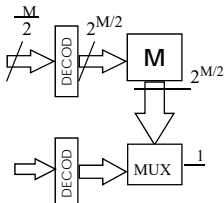
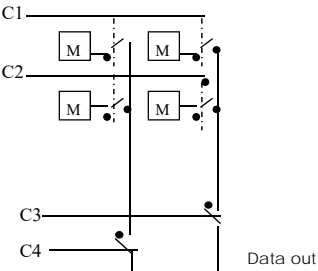
## Descodificação de Endereços

Memories

- **Estrutura linear**
  - mais rápida, muito hardware

- **Estrutura a 2 dimensões**
  - mais lenta, menos hardware

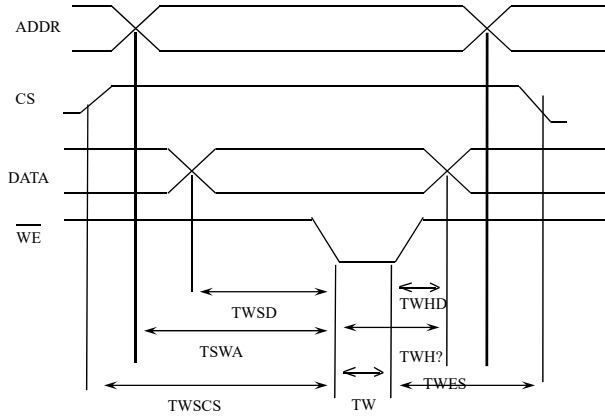



7

## Temporizações

Memories

- **Ciclo de escrita (o de leitura é semelhante)**
  - ADDR - Endereços
  - DATA - Dados a escrever (/ler)
  - CS - Chip Select
  - WE - Write Enable



8

## MEMÓRIAS RAM (Dinâmicas)

Memories

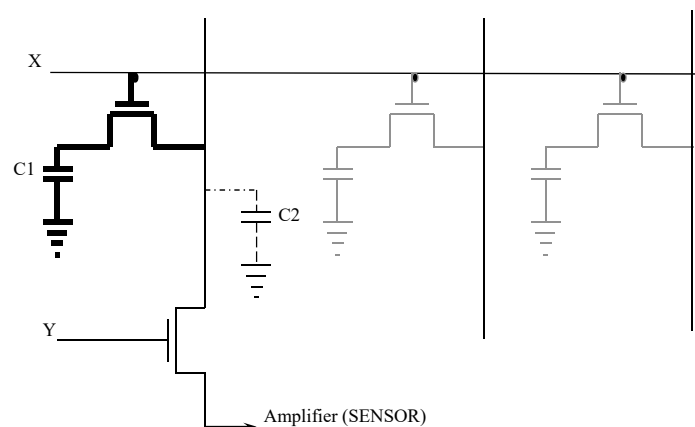
- Os dados (bits) são guardados em condensadores
- Vantagens
  - Ocupam muito menos espaço
  - Têm muito menos dissipação
- Problemas
  - perda de carga com o tempo
  - perda de carga com as leituras
  - necessidade de regenerar a informação (refresh)
- Têm necessidade de usar circuitos para gerar os ciclos de "refresh" (2ms)

9

## Dynamic RAM

Memories

- Basic memory block
  - 1 FET (Field Effect Transistor) + 1 Condensador (also a FET)



10

# Memories

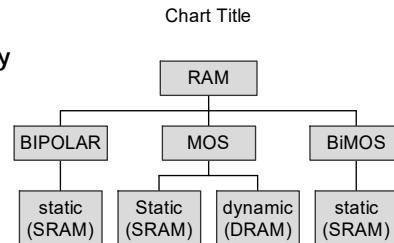
V.2.0 V.Lobo 2016

## Choice of RAMS (DRAM vs SRAM)

Memories

### ● Factors to take into account

- Memory size
  - DRAMs have far more capacity
- Cost per bit
  - DRAMs are cheaper
- Physical size
  - DRAMs are smaller
- Power consumption
  - DRAMs consume less energy
- Access times
  - SRAMs are faster, and don't have "dead time"
- Ease of use
  - SRAMs are easier to use (and cheaper for small capacities)



11

## ROM Memories

Memories

### ● Read Only Memory

- Their content is pre-defined and can't be changed
- They store "permanent" information: programs that do not change (BIOS, Firmware), tables with fixed contents (e.g. character sets in printers), etc.

### ● They are *not volatile*

- Their content is not lost when power is switched off

### ● They are purely combinatory circuits

- For a given input (address), they always produce the same output (datum)
- They can be built with logical gates, using the traditional ways to implement Boolean functions
- RAMS have no memory !!!

12

# Memories

V.2.0 V.Lobo 2016

## ROM Memories

Memories

- Ex: 4 addresses with 3 bits

Address	Content
00 (0)	110
01 (1)	111
10 (2)	010
11 (3)	011

**AND plane (level)**

Decodes addresses, generating all possible minterms

**OR plane**

Information specific to the particular content

Simplification with Karnaugh maps

13

## ROM Memories

Memories

- The OR plane can be substituted by a matrix of diodes
- ROMs are programmed when they are built (in the factory)
  - ORs are made by inserting (or not) diodes in the junctions
  - A Mask is used (similar to those in printed circuit boards) to build them
  - Only financially viable in very large quantities

$$D_0 = m_0 + m_1$$

$$D_1 = m_0 + m_1 + m_2 + m_3$$

$$D_2 = m_1 + m_3$$

14

# Memories

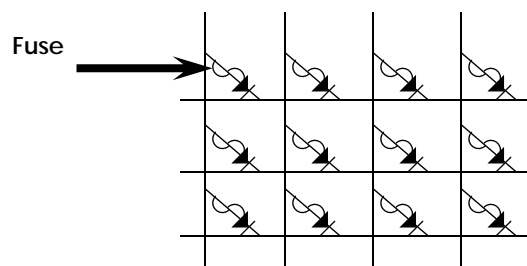
V.2.0 V.Lobo 2016

## ROM Memories

Memories

- **Programmable Read Only Memory**

- You can program them, but only once
- They are similar to ROMs, but they have fuses together with diodes in the line/columns junctions, that may be burned to "take out" the diode.
- To program a PROM, the fuses are burned by storing the opposite of what we want with voltages higher than normal (PROMs are "blown out")



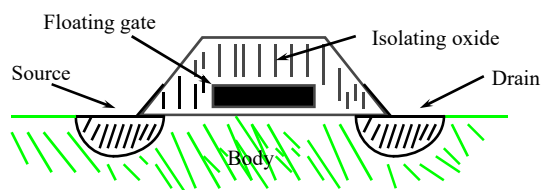
15

## EPROM Memories

Memories

- **Erasable Programable Read Only Memory**

- It is possible to erase the contente, and re-program it
- The erasure/reprogramming takes time
- FAMOS (Floating-Gate Avalanche-Injection Metal Oxide Semiconductor) devices are used.
  - They are MOS-FETs with a isolated (floating) gate
  - The charge stores in the gate puts the FET in conducting or isolating mode
  - The floating gate can be electrically charged using high negative voltage and tunneling effect.
  - The floating gate can be discharged using ultraviolet radiation that makes the oxide a conductor.



16



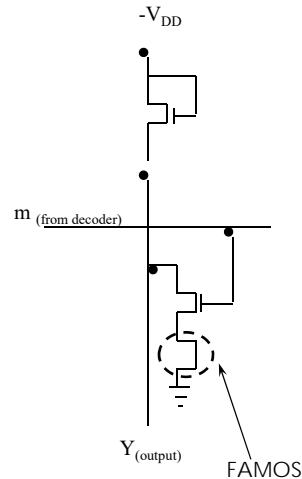
# Memories

V.2.0 V.Lobo 2016

## EPROM Memories

Memories

- Erasing can take about 10 minutes under a UV lamp
  - It is not practical to reprogram when in use
  - The integrated circuits have a class window to allow the light in
- It may be programmed BYTE by BYTE
  - A FAMOS will be discharged (and the logical content will be "1") until voltage is applied to charge it
  - We may program it one address at a time (converts the 1s to 0s)

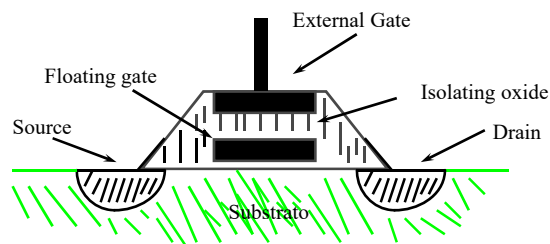


17

## E<sup>2</sup>PROM or EEPROM Memories

Memories

- Electrically Erasable Prom
  - Use STACKED GATE FET
    - FETs with a floating gate between the base and the external gate
    - São postos em condução/corte pela Gate Flutuante que é carregada e descarregada pela Gate exterior, por efeito de túnel/avalanche
    - Demoram cerca de 1s a programar.
    - Eram muito caras, e pouco usadas. O aparecimento da tecnologia "flash" trouxe série de dispositivos baseados em flash-eprom



18

# Memories

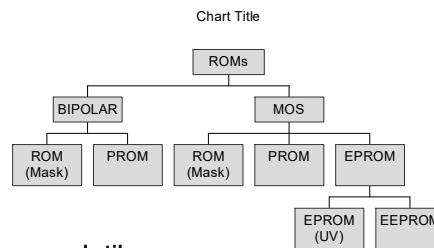
V.2.0 V.Lobo 2016

## Choice of ROM Memories

Memories

- Factors to take into account choosing ROMS

- Quantity
  - PROMs are cheaper (they are always build in quantity)
- Stability of information
  - EPROMs better during the design phase (they may be corrected)
- Need to reprogram frequently
  - EEPROM are more versatile by EPROM may be acceptable



- Applications

- Startup code
- Code that doesn't change
- Code that has to robust and non-volatile
- Code converters (bin/bcd, bcd/7 seg, etc)
- Control (sensors/alarms/actuators)

19

Memories

## Other technologies used in memories

20

# Memories

V.2.0 V.Lobo 2016

## Magnetic memories

Memories

- Used mainly as secondary memories
- Sequential access
  - Tapes ("normal" tapes, cartridges , DAT, etc), Disks (hard drives and floppy)
- Random access
  - Ferrite, Magnetic bubble (that has latency times)

```
graph TD; A[MAGNETIC MEMORIES] --> B[FERRITE]; A --> C[BUBBLE]; A --> D[TAPE]; A --> E[DISK]; A --> F[MAGNETO-OPTICAL]
```

21

## FERRITE MEMORIES

Memories

- Basic principle
  - A ferrite torus can be magnetized in two different directions (one for each logic value)
  - To magnetize it, simply pass a high current through its center
  - To "measure" the magnetization, we need to measure the impedance it offers to a low current

Hysteresis loop

Forced current induces magnetization

Sensor wire where the impedance is measured

Torus matrix

22

# Memories

V.2.0 V.Lobo 2016

## MAGNETIC MEMORIES

Memories

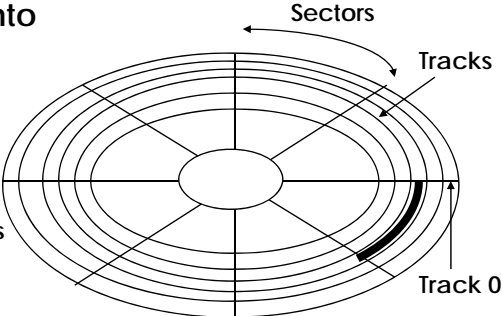
- **Ferrite memory's features**
  - Very low density (torus are macroscopic!)
  - Long access time
  - Destructive reading
  - Non-volatile memory
- **Magnetic bubble memories**
  - Weiss' domains are individually manipulated
  - Large latency times
  - High density

23

## DISK

Memories

- **Keeps information about one (or multiple) surfaces of magnetic material**
- **The disk is divided into**
  - Sectors
  - Tracks (or cylinders)
  - Heads (or surfaces)
- **Formatting**
  - Creation of the markers
  - Track numbering



24

# Memories

V.2.0 V.Lobo 2016

## MAGNETIC TAPES

Memories

- Very similar to audio magnetic tapes
- Have **MARKERS** to perform quick searches, followed by data **REGISTERS**
- Used as backup, or to download massive amounts of data

25

V.Lobo 2016

Memories

## Outras organizações para sistemas de memórias

26

V.Lobo 2016

# Memories

V.2.0 V.Lobo 2016

## MEMÓRIAS FIFO ou ELÁSTICAS

Memories

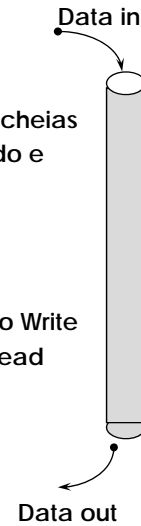
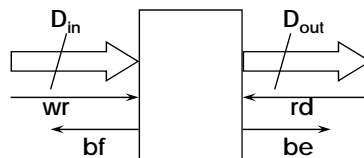
- First In First Out

- Estrutura similar a um Shift Register:

- Os dados entram por um lado e saem pelo outro
- Ao contrário dos Shift Register podem estar parcialmente cheias
- Semelhante a um tubo onde os dados são postos num lado e retirados do lado oposto

- Funcionamento

- Têm sinais que indicam se estão cheias/vazias
- Para escrever: verificar o sinal Buffer Full, e depois actuar o Write
- Para ler: verificar o sinal Buffer Empty, e depois actuar o Read



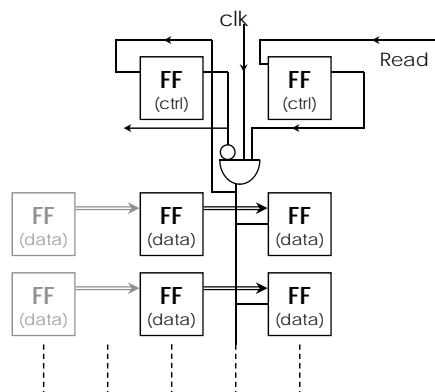
27

## MEMÓRIAS FIFO ou ELÁSTICAS

Memories

- Implementação

- É necessário gerir quais as posições ocupadas e livres
- Há um conjunto de registos auxiliares que indicam se uma posição está livre ou não (1=livre, 0=ocupado)
- Há outros tipos de implementações (shift regs c/ controlo de leitura)



28

## MEMÓRIAS FIFO ou ELÁSTICAS

Memories

- **Utilização**
  - Buffer de espera para adaptar 2 sistemas de débitos diferentes
  - Comunicações; Filas de espera; Controlo, etc.
- **Podem ser simuladas com memória convencional**
  - Usam-se 2 apontadores: um para leitura, outro para escrita
  - Para escrever:
    - Verificar BP; Escrever na posição WP e incrementar WP ; Se  $WP=MAX \Rightarrow WP=0$ ; se  $WP=RP \Rightarrow BF$
  - Para ler:
    - Verificar BE; Ler na posição RP; Se  $RP=MAX \Rightarrow RP=0$ ; se  $RP=WP \Rightarrow BE$

29

## MEMÓRIAS LIFO

Memories

- **LIFO - Last in, first out**
  - Também chamadas PILHAS ou STACKS
  - O último dado a entrar é o primeiro a sair
  - "Push-down stack"
- **Operações**
  - PUSH - Pôr um dado no Stack
  - POP - Retirar um dado do stack
- **Implementação**
  - Com um shift-register bi-direccional
  - Em software
    - Stack Pointer
    - Zona reservada de memória
    - Rotinas de PUSH e POP

30

## MEMÓRIAS ASSOCIATIVAS

Memories

- Também chamadas *endereçáveis por conteúdo*
- Dando um conteúdo, obtém-se um endereço
  - Associam um conteúdo a um endereço
  - "endereçam" com o conteúdo
- Utilização
  - Índices
  - Sistemas de memória virtual

→ 36 ?

← 1

0	12
1	36
2	25
3	10

31

## PLD - Programmable Logic Devices

Memories

- São semelhantes a PROMs
  - Têm um plano de ANDs seguindo de um plano de ORs
  - São usadas para gerar Funções Lógicas
  - Ao contrário das ROMs, NÃO têm todos os MINTERMOS
  - Embora muito flexíveis e versáteis, como lhes faltam alguns graus de liberdade, não podem gerar qualquer função

Um plano de ANDs:

32



## PLD - Programmable Logic Devices

Memories

- **PLA - Programmable Logic Array**
  - Têm ambos os planos (ORs e ANDs ) programáveis
- **PAL - Programmable Array Logic**
  - Apenas o plano dos ANDs é programável
- **Outros**
  - PLS (Programmable logic sequencer)
  - FPA (Field Programmable Array), etc.
- **Vantagens**
  - São muito mais baratas e fáceis de montar que lógica discreta
  - São mais eficientes e baratas que PROMs completas
  - São facilmente programáveis com o auxílio de software

33

## BANCOS DE MEMÓRIA

Memories

- **Conceito de ESPAÇO DE ENDEREÇAMENTO**
  - Endereços que podem ser gerados com um dado número de bits
  - Pode ou não corresponder a memória física
  - Ex: o 8085 tem 16 bits de endereços, logo um espaço de endereçamento de 64K; o Pentium II tem 32 bits de endereços, logo um espaço de endereçamento de 4G
- **Inserção de uma memória no espaço de endereçamento**
  - A memória física é MAPEADA para endereços do espaço de endereçamento da máquina
  - Usam-se alguns bits do endereço para gerar o Chip Select (CS)
    - O endereço para onde a memória física é mapeada depende da função que é usada para gerar o CS
    - Alguns dos bits de endereço são usados para aceder às diversas posições dentro da memória

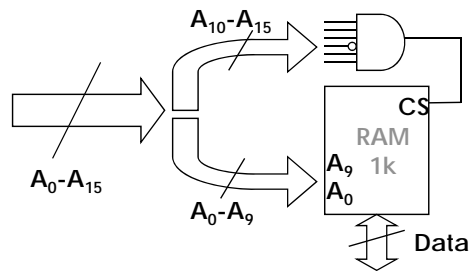
34

## BANCOS DE MEMÓRIA

Memories

### ● Exemplo:

- Inserir uma memória física de 1K num espaço de endereçamento gerado por 16 bits de endereço, de tal modo que fique mapeada para os endereços entre 4 e 5K.
- Solução
  - Uma memória de 1K necessita de  $\log_2(1k)=10$  bits para endereçamento interno
  - Os restantes 6 bits de endereço serão usados para gerar o CS
  - Quando esses 6 bits formarem o nº 4, o CS deverá ser 1



35

## BANCOS DE MEMÓRIA

Memories

### ● Memórias intercaladas

- Não é necessário que sejam os bits mais significativos a ser usados para gerar o CS
- O que acontece se usarmos os menos significativos ?

### ● Replicação de memória

- Não é necessário fazer descodificação completa dos endereços.
- O que acontece se não usarmos todos os bits que "sobram" para o CS ?

36

## Bancos de memória

Memories

### ● Problema

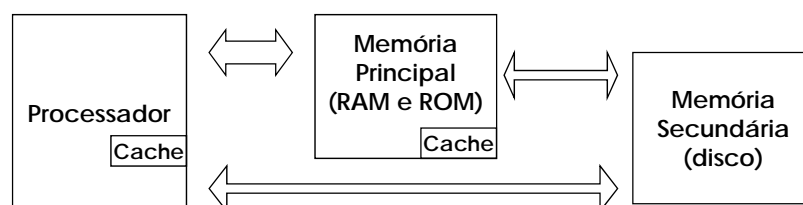
- Um dado sistema baseado num microprocessador tem um Bus de endereços de 16 bits, e um bus de dados de 8 bit
- Pretende-se ter memória ROM nos primeiros 8K endereços, e entre os endereços 8000H e A000H pretende-se ter memória RAM.
- Tem-se à disposição integrados de memória ROM com 8K×4 bits, e integrados de memória RAM com 1K×8 bits.
- Desenhe o logigrama do sistema de memória pretendido

37

## HIERARQUIA DE MEMÓRIAS

Memories

- Memória Cache
- Memória principal ou primária
- Memória secundária, ou "em massa"



38