

MC404

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

2010

Prof. Paulo Cesar Centoducatte

Prof. Mario Lúcio Côrtes

Prof. Ricardo Pannain

MC404

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

**“Principais Componentes de um Sistema
Computacional”**

Principais Componentes de um Sistema Computacional

Sumário

- **Memórias**
- **CPUS**
- **Unidades de I/O (periféricos de entrada e saída)**

Memória

- **Memórias: Dispositivo capaz de armazenar informação para posterior recuperação**
 - **OBS. Conceito bastante antigo**
 - Saco com pedrinhas ou sementes
 - Marcas em tijolos de argila
 - Escrita em papiros/papel
 - Marcas com “tintas” em cavernas
 - Etc
- **Anológicas/Digitais**
 - Contador de pontos em bilhar
 - Disco de vinil (som)
 - K7
 - CD
 - HD

Memória

- **Memórias Digitais (bits) implementadas como circuitos eletrônicos integrados - “CI’s de memória”**
- **Operações suportadas por uma memória:**
 - **Entrada de dados: escrita (gravação, apagamento, programação) - write**
 - **Saída de dados: leitura – read**
- **Memória simples**
 - **Flip-Flop**
 - **Escrita de “1” - set**
 - **Escrita de “0” – reset**
 - **Leitura – verificar se o FF está set ou reset**

Memória

- **Memórias (reais)**
 - Podem ser vistas como grandes vetores de bits
 - Capacidade
 - N bits
 - N/8 Bytes
 - Abreviações normalmente utilizadas para expressar a capacidade de uma memória:
 - K (kilo) = $2^{10} = 1024 \sim 1000 = 10^2$
 - M (mega) = $2^{20} = 1.048.578 \sim 10^6$
 - G (giga) = $2^{30} \sim 10^9$
 - T (tera) = $2^{40} \sim 10^{12}$

Memória

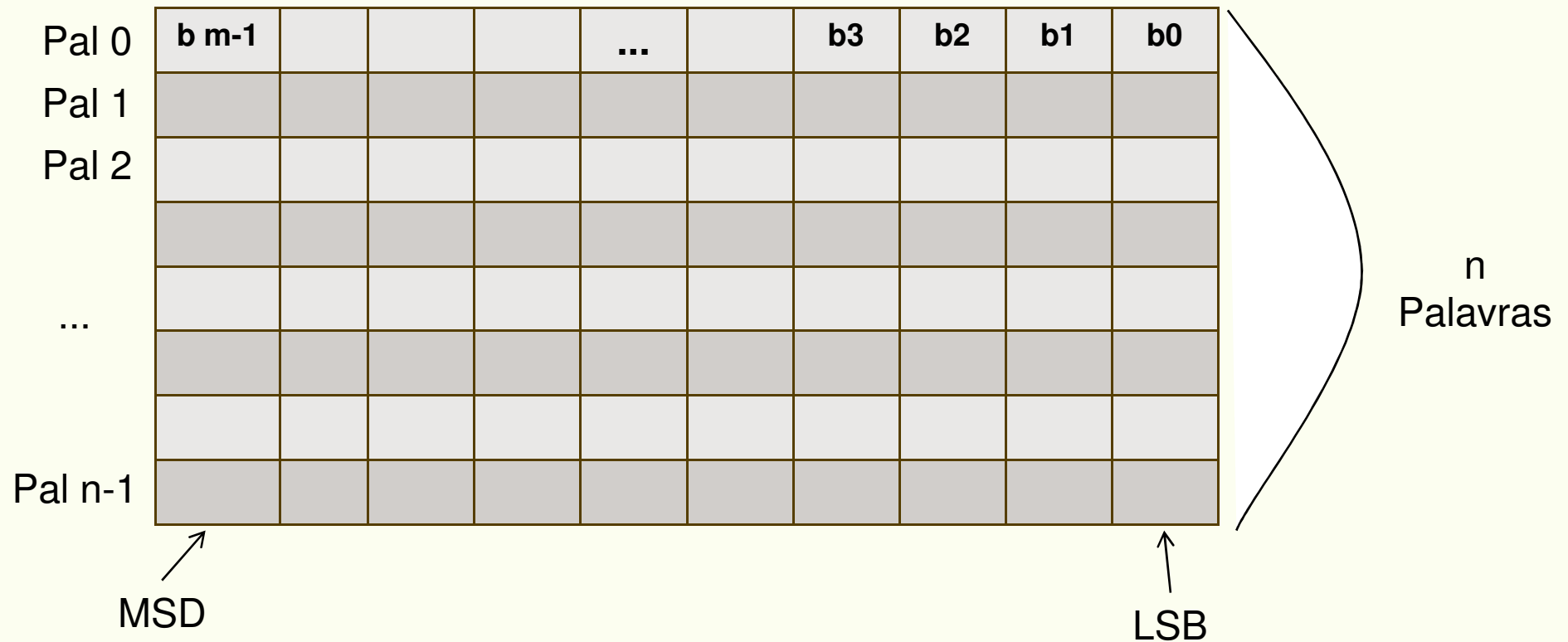
- **OBS.:**
 - Memórias normalmente utilizam 2^N devido ao espaço de endereçamento das CPU's
 - HD's tendem a utilizarem 10^N
 - Quando falarmos em capacidade de memória usaremos 2^N
- **Palavras de Memória (words)**
 - Números de bits transferidos em paralelo, em uma operação (leitura ou escrita)
 - Expresso em bits ou Bytes -- largura da palavra (word size)

Memória

- **Organização de uma Memória**
 - **Profundidade x Largura**
 - **Ex.: 64Kx8-bits = 64 KBytes**
 - **Obs.: Memórias com mesma capacidade podem ter organizações diferentes**
 - **Ex.: capacidade 512 K bits = 64 KBytes**
 - **64K x 8- bits**
 - **32K x 16-bits**

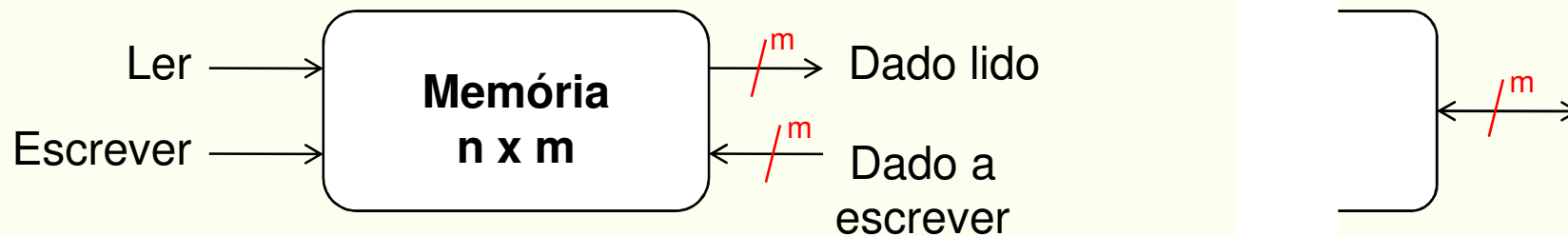
Memória

- Memória $n \times m$ (n palavras com m bits)



Memória

- Uma operação (leitura ou escrita) em uma memória afeta uma e somente uma palavra e todos os bits da palavra são lidos ou escritos em paralelo.



- **Obs.:**
 - Em memórias com $m \geq 8$, como leitura e escritas são mutuamente exclusivas, frequentemente possuem somente um conjunto de linhas de dados bi-direcional.

Memória

- **Forma de especificar a palavra que será lida/escrita**
 - **O método mais comum (e único a ser visto neste curso) :
Endereçamento explícito**
 - **Associa-se a cada palavra um número inteiro ≥ 0 que a especifica univocamente : conhecido como ENDEREÇO da palavra**
 - **Normalmente, para usar toda a capacidade de expressão dos bits de endereço, usualmente o número de palavras de uma memória é uma potência de 2, ié $n = 2^K$**
 - **O endereço de memória é um inteiro binário com K bits.**
 - **Obs.: Outras formas de endereçamento, utilizadas em aplicações especiais incluem: endereçamento implícito (pilha e fila) e endereçamento por conteúdo.**

Memória

- **AVR ATmega88**
 - **Memória de Dados de 1Kx8bits (1K Bytes)**
 - 10 bits de endereço de dados
 - **512 Bytes de EEPROM**
 - **8K Bytes In-System Self-Programmable Flash**

Memória

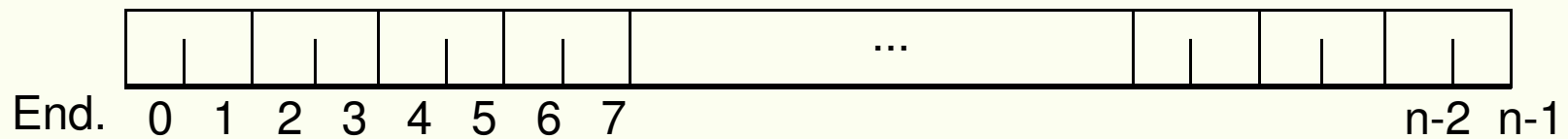
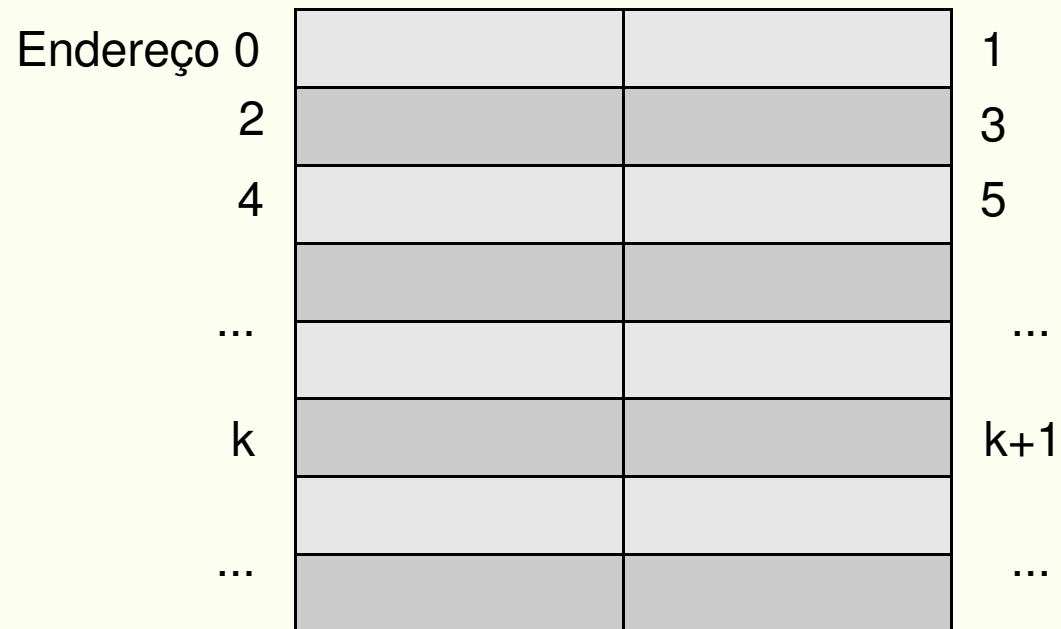
- **Mapa de Memória**
 - **Espaço de Endereçamento**
 - **Alocação dos Endereços**
- **Obs.:**
 - **Posição da Palavra de Memória**
 - **Conteúdo da Palavra de Memória**
- **Se um sistema computacional utiliza endereços com K bits para designar as palavras de memória de dados, dizemos que o espaço de endereçamento de dados deste sistema é de 2^k palavras**
 - **Obs.: Nem sempre todo o espaço de endereçamento de memória está alocado. Pode haver buracos**
 - **Permite expansão da memória sem modificação do sistema**

Memória

- **Nível de Endereçamento**
 - **Para memórias com palavras maiores que 1 Byte (ex. 16 bits ou 32 bits), é possível usar endereçamento com diferentes granularidades**
 - **1 endereço por byte**
 - **1 endereço por palavra**
 - **1 endereço por cada $\frac{1}{2}$ palavra**
 - **Etc**

Memória

- Memórias com palavras de 16 bits e endereçamento a Byte



Memória

- **Memórias com palavras de 16 bits e endereçamento a Byte**
 - **Acesso alinhado**
 - **Só é permitido acesso a palavras cujo endereço é múltiplo do tamanho da palavra em bytes**
 - **Ex. 0 (bytes 0 e 1); 2 (bytes 2 e 3); 4 (bytes 4 e 5); 16 (bytes 16 e 17)**
 - **Acesso desalinhado**
 - **É permitido acesso a palavras a partir de qualquer endereço**
 - **Ex.: 0 (bytes 0 e 1); 1 (bytes 1 e 2);**