



ARQUITETURA DE COMPUTADORES – LABORATÓRIO

André Breda Carneiro
Sidney José Montebeller
Rafael Rodrigues Da Paz

Experiência Nº 4 – BIG ENDIAN E LITTLE ENDIAN.

Objetivos:

- Adquirir conhecimentos em organização de dados;
- Estudo das estruturas big e little endian;
- Simulação para compreensão de suas diferenças e importâncias.

Big Endian e Little Endian

São as formas que os bytes e bits são endereçados na memória!

O conceito de Big Endian e Little Endian, nomeados simplesmente de Endianess vem da transição dos computadores de médio porte para os microcomputadores, quando estes passaram a endereçar tantos os bits quanto os Bytes de forma diferente. Mas tal problema é principalmente observado quando lidamos com os Bytes, já que pode acarretar no “embaralhamento” de dados (por exemplo textos) causando, assim, confusão ou até mesmo a falência total do sistema.

Nos tempos atuais, não temos problemas relativos a tal modo de endereçamento, porque a grande maioria dos microprocessadores usam o Little Endian para endereçar seus dados.

Alguns exemplos de Big Endian estão nos PowerMAC que usava um PowerPC especial travado para BIG Endian, Motorola 68000 series (incluindo Freescale ColdFire), Xilinx Microblaze, SuperH, IBM z/Architecture, Atmel AVR32 e o Intel 8051.

Computadores que não usam travamento permitem o chaveamento entre os dois modos no que tange a manipulação dos bytes.

Big Endian e Little Endian

Quando se trata de bytes o Big Endian endereça em uma palavra, por exemplo do tipo 2 bytes, o primeiro byte como sendo o endereço menor, e a segunda palavra o endereço seguinte (+1 byte). Já no Little Endian, o segundo Byte é endereçado primeiro.

Vamos lembrar alguns conceitos:

LSB representa a parte menos significativa do número ou seja a parte mais a direita. *Least Significant bit/Byte*.

MSB representa a parte mais significativa, ou seja a parte mais a esquerda do número. *Most Significant bit/Byte*.

Organização a nível de bits

A representação numérica em bits, onde o algoritmo de conversão numérico que a maioria de nós está acostumado pode ser representado na fórmula:

$$\sum_{i=0}^{N-1} b_i \cdot 2^i$$

Temos então a seguinte ordenação dos bits para a representação do número 180, onde o bit menos significativo é tratado como sendo o bit 0 e o bit 7 é o bit mais significativo.



Organização a nível de bits

Os bits mantêm sua disposição, porém, em algumas arquiteturas sua ordem de transmissão inverte, ou seja são endereçados do MSB como sendo o primeiro bit, e o LSB como sendo o último bit. Portanto, a fórmula de conversão passa a ser:

$$\sum_{i=0}^{N-1} b_i \cdot 2^{(N-1-i)}$$



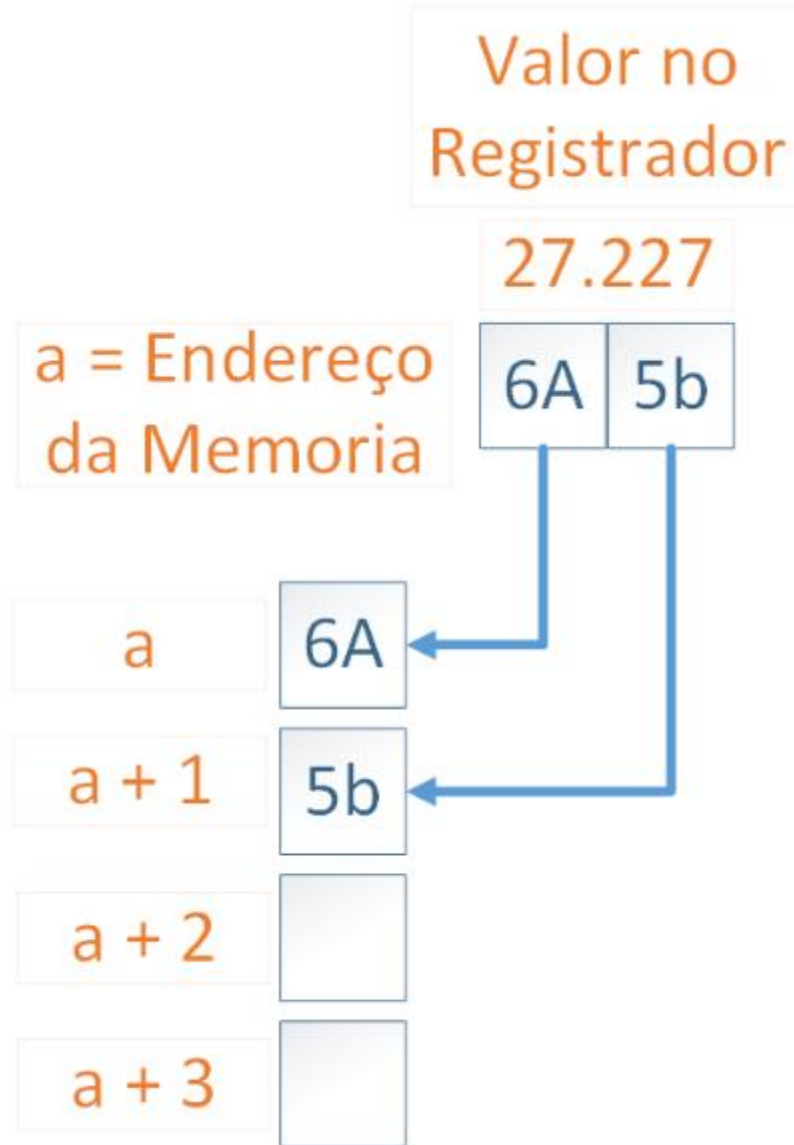
O Conceito Endianess

Em se tratando de bits o conceito **Endianess** afeta mais o hardware no que tange o endereçamento de memória, transferência de dados em barramentos, principalmente nos seriais e operações de manipulação de bits, já que principalmente se formos usarmos máscaras do tipo bitwise (alterar a sequência de bits de uma variável) é preciso saber exatamente a ordem dos bits para não haver enganos fatais.

Vejamos agora como é tratado o conceito **endianess** quando se trata de bytes, o que afeta mais a manipulação do dado na memória quando é representado com mais de dois bytes, por exemplo números *inteiros* e *short int* em máquinas de 32 bits.

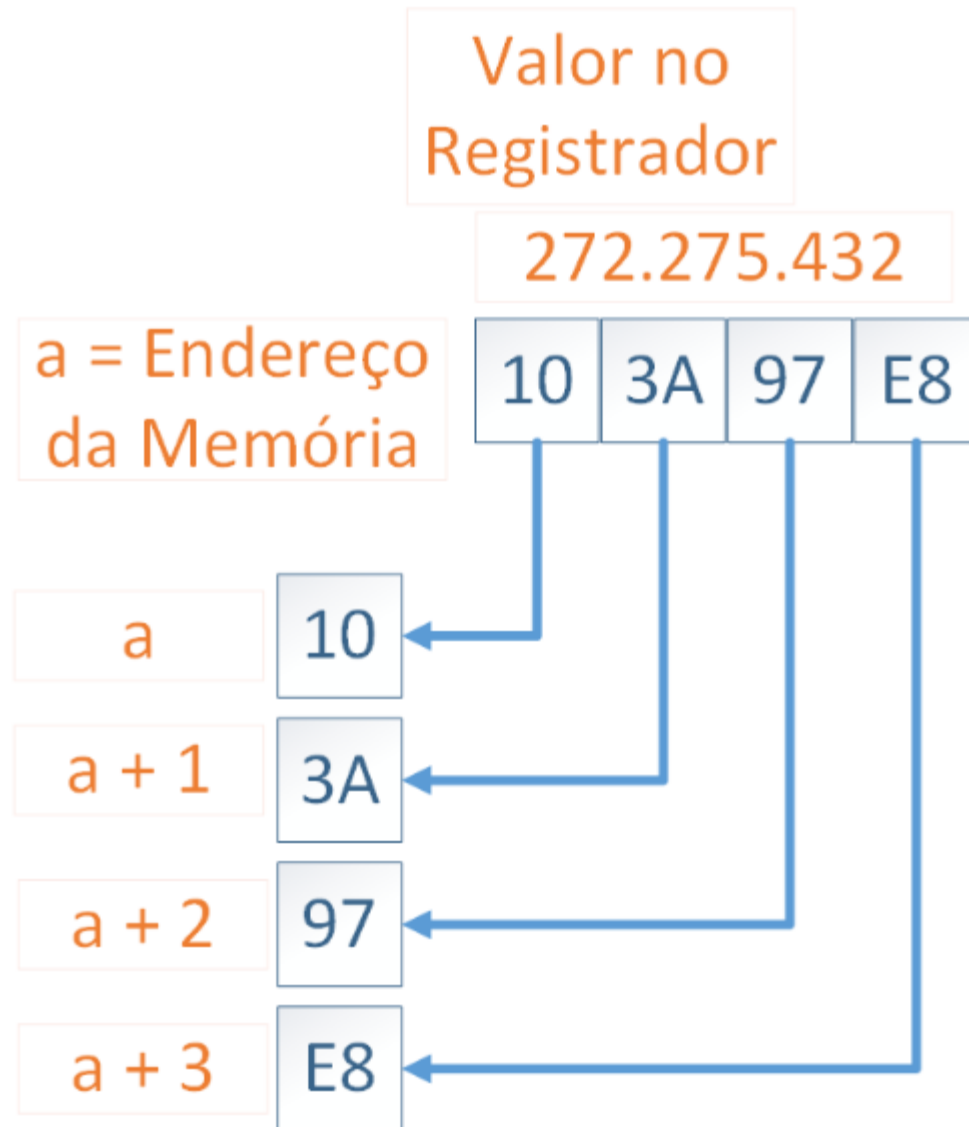
Big Endian a nível de bytes

A imagem representa números inteiros armazenados na memória de um microcontrolador qualquer que seja do tipo **Big Endian**, um número de 16 bits (ou seja um Word).



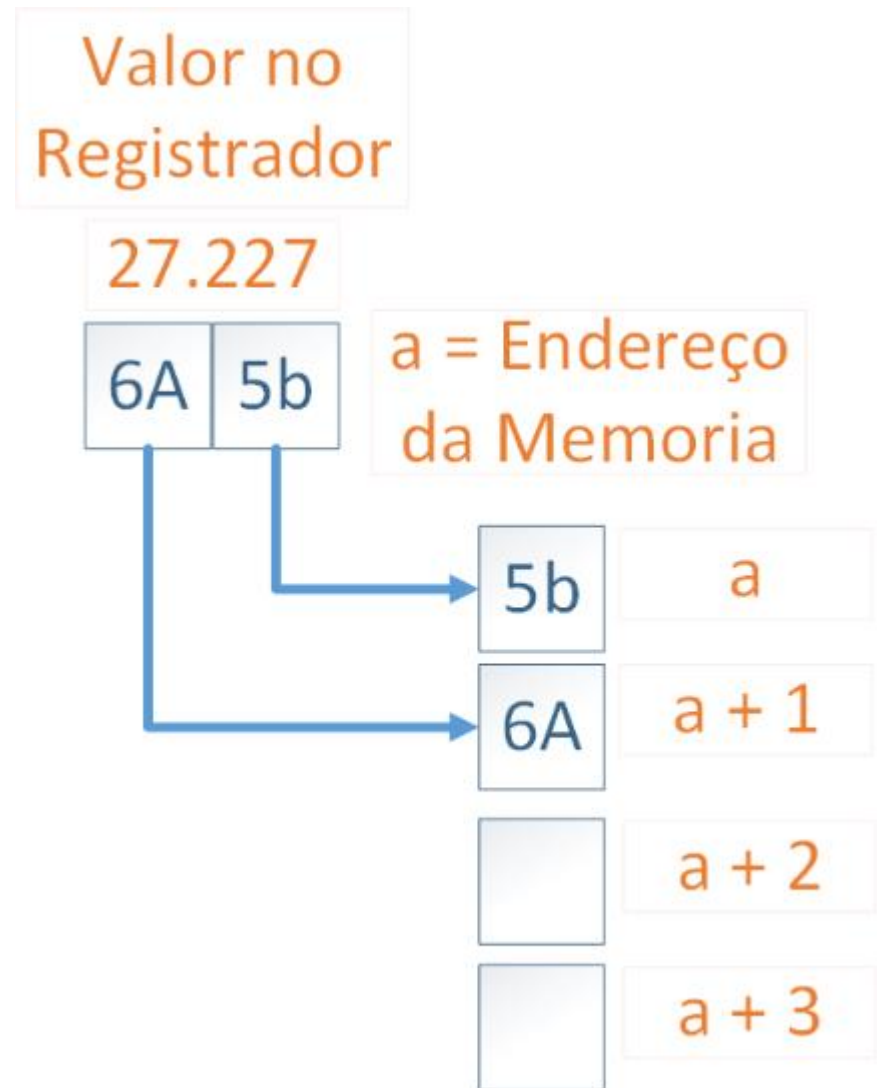
Big Endian a nível de bytes

A imagem representa números inteiros armazenados na memória de um microcontrolador qualquer que seja do tipo **Big Endian**, um número de 32 bits (ou seja Double Word - DWord).



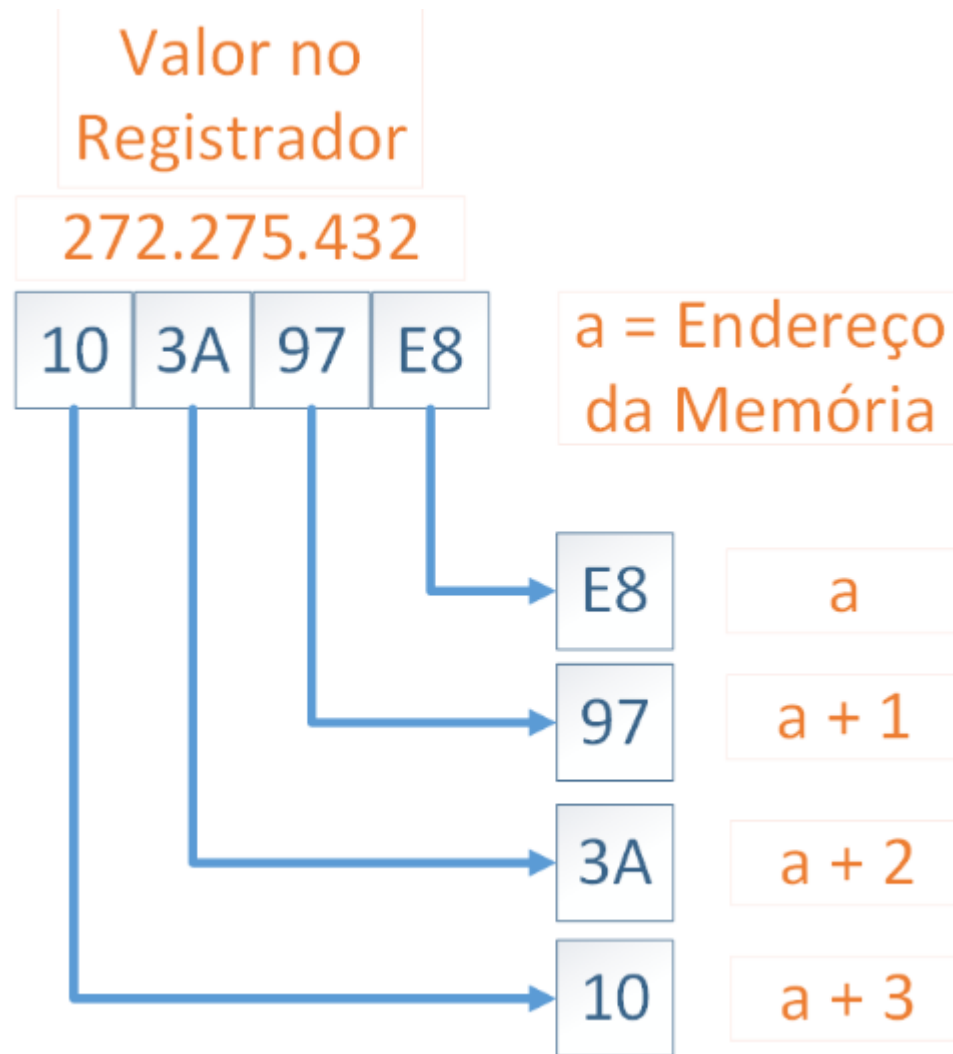
Little Endian a nível de bytes

A imagem representa números inteiros armazenados na memória de um microcontrolador qualquer que seja do tipo **Little Endian**, um número de 16 bits (ou seja um Word).



Little Endian a nível de bytes

A imagem representa números inteiros armazenados na memória de um microcontrolador qualquer que seja do tipo **Little Endian**, um número de 32 bits (ou seja Double Word - DWord).



Relatório 4

- Projeto a ser desenvolvido no simulador LOGISIM:
 - Construir um circuito manipulador de memória que transforma dados em Big endian para Little endian e vice versa;
 - O sistema deve ser de 32 bits e estar organizado em 4 grupos de 8 bits;
 - Cada meio byte pode ser representado por um display de 7 segmentos;
 - Deve possuir, memória, entradas de controle, algo para ilustrar os dados (displays).*
- Introdução;
- Definição do circuito no software;
- Procedimento experimental executado;
- Explicação de seu funcionamento;
- Conclusão.

Fonte: <http://carlosdelfino.eti.br>