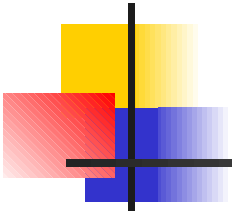




# LITTLE ENDIAN vs. BIG ENDIAN

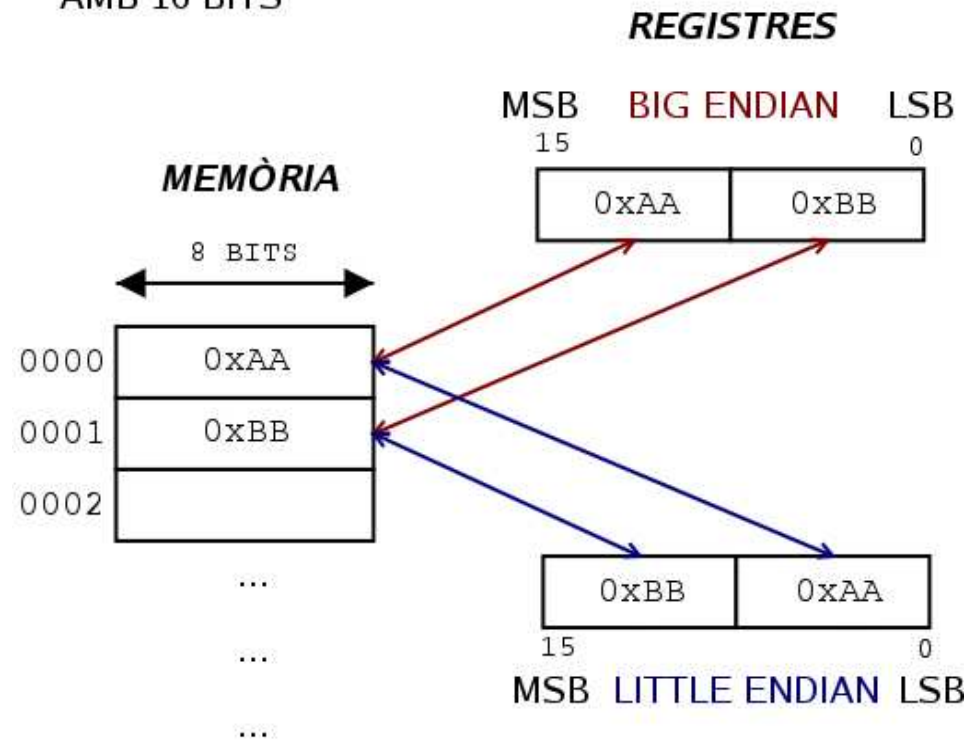
---

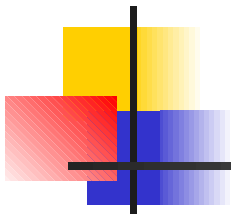
- Dues formes diferents d'accedir a memòria a blocs més grans d'1 byte
  - LITTLE: Posicions baixes, bytes de menys pes
  - BIG: Posicions baixes, bytes de més pes
- Cada arquitectura utilitza la seva pròpia representació
- Pot causar problemes en la compartició de dades entre diferents architectures



# BIG vs. LITTLE ENDIAN - 16 BITS

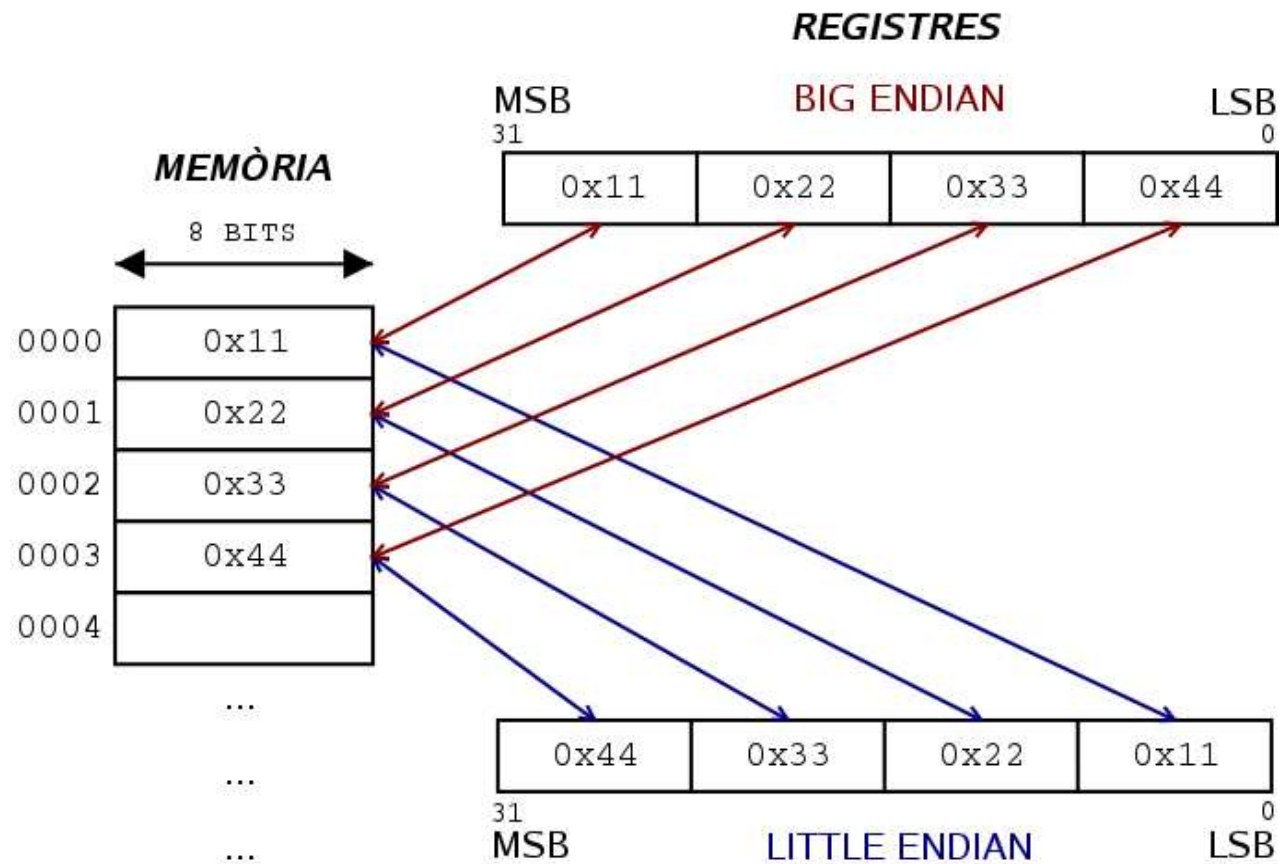
REPRESENTACIÓ  
AMB 16 BITS





# BIG vs. LITTLE ENDIAN - 32 BITS

REPRESENTACIÓ  
AMB 32 BITS





# PLATAFORMES

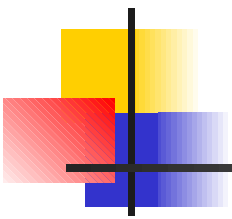
---

- **BIG ENDIAN**

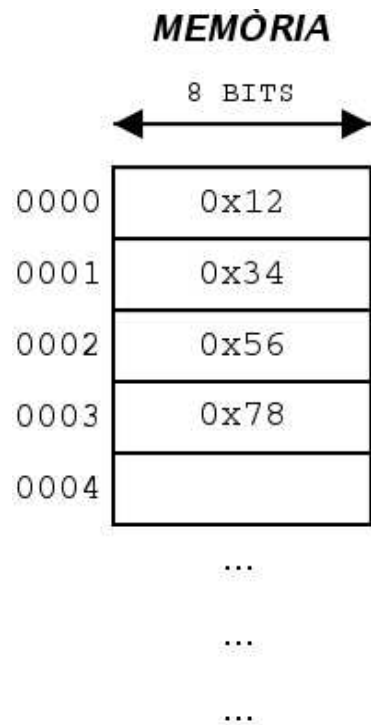
- SPARC
- Motorola 68000
- MIPS
- SGI
- SUN

- **LITTLE ENDIAN**

- VAX
- Intel x86
- Intel Pentium



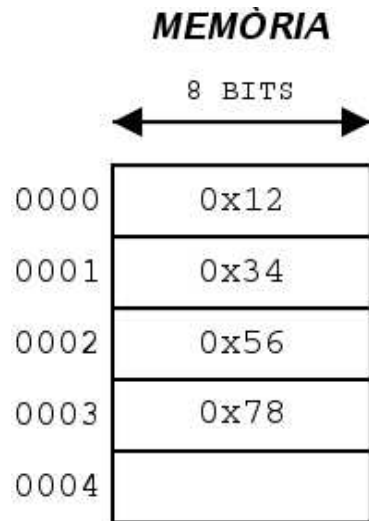
# COM GUARDAR *0x12345678* ?



AMB:

1. Accessos a 8 bits
2. Accessos a 16 bits Little Endian
3. Accessos a 16 bits Big Endian
4. Accessos a 32 bits Little Endian
5. Accessos a 32 bits Big Endian

# SOLUCIÓ



**8 BITS**

```
MOV [0000], 0x12
MOV [0001], 0x34
MOV [0002], 0x56
MOV [0003], 0x78
```

**16 BITS**

**32 BITS**

**BIG**

```
MOV [0000], 0x1234
MOV [0002], 0x5678
```

```
MOV [0000], 0x12345678
```

**LITTLE**

```
MOV [0000], 0x3412
MOV [0002], 0x7856
```

```
MOV [0000], 0x78563412
```