

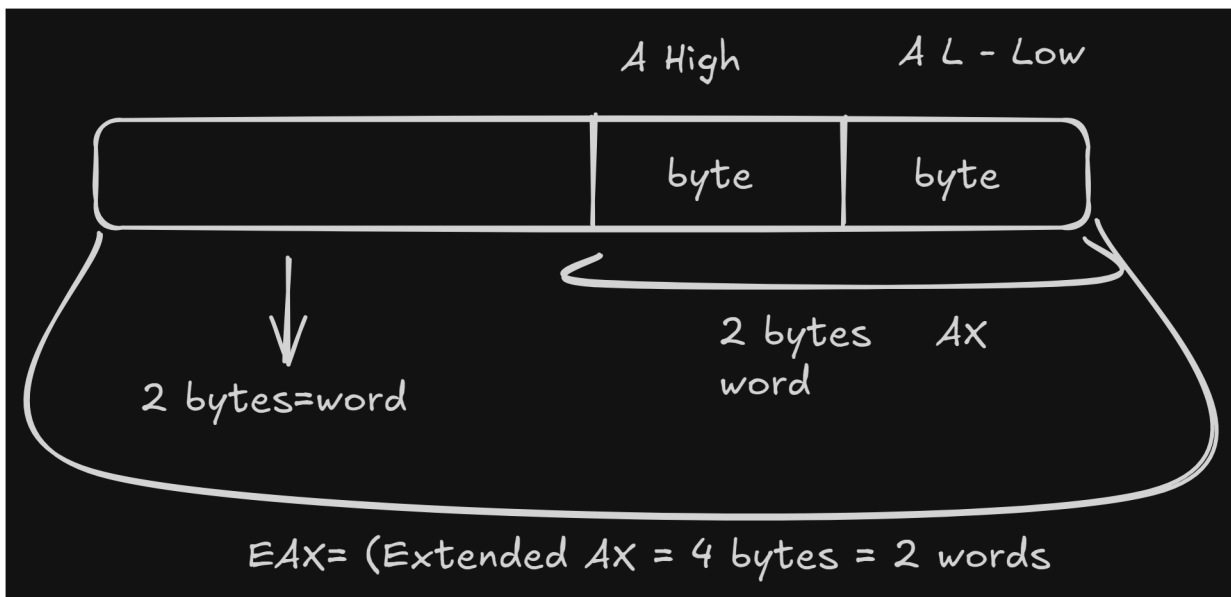
$$\begin{aligned}
 -3 &= 1000\ 0011_{(2)} \implies -3 + 3 \neq 0 \xrightarrow{\text{flip bits}} \\
 1111\ 1100_{(2)} &\implies -3 + 3 \neq 0 \xrightarrow{\text{add 1}} \\
 1111\ 1101_{(2)} &\implies -3 + 3 = 0
 \end{aligned}$$

Common registers: `EAX`, `EBX`, `ECX`, `EDX`

```

mov al, 5
mov al, 17
mov ah, 5

```



### ① Types of registers

- `ah` or `al` are small, 1 byte registers (A high or A low)
- `ax` is a 2 byte register, equal in size to a word
- `eax` is a 4 byte, double word register

### 🔗 Types of "variables"




- `db` = 1 byte
- `dw` = 2 bytes (word)
- `dd` = 4 bytes (double word)

### 🔗 Important

Computer memory is little-endian, registers are big-endian, i.e. memory is read back to front. This is done for optimization reasons

### Bases in assembly

- numbers are implicitly decimal (D,d)
- base 2, binary (B,b)
- base 8, octal (O,o)
- base 16, hexa, (H,h)

  2024-10-07  Homework for the lab (check website, 2nd in semigroup)