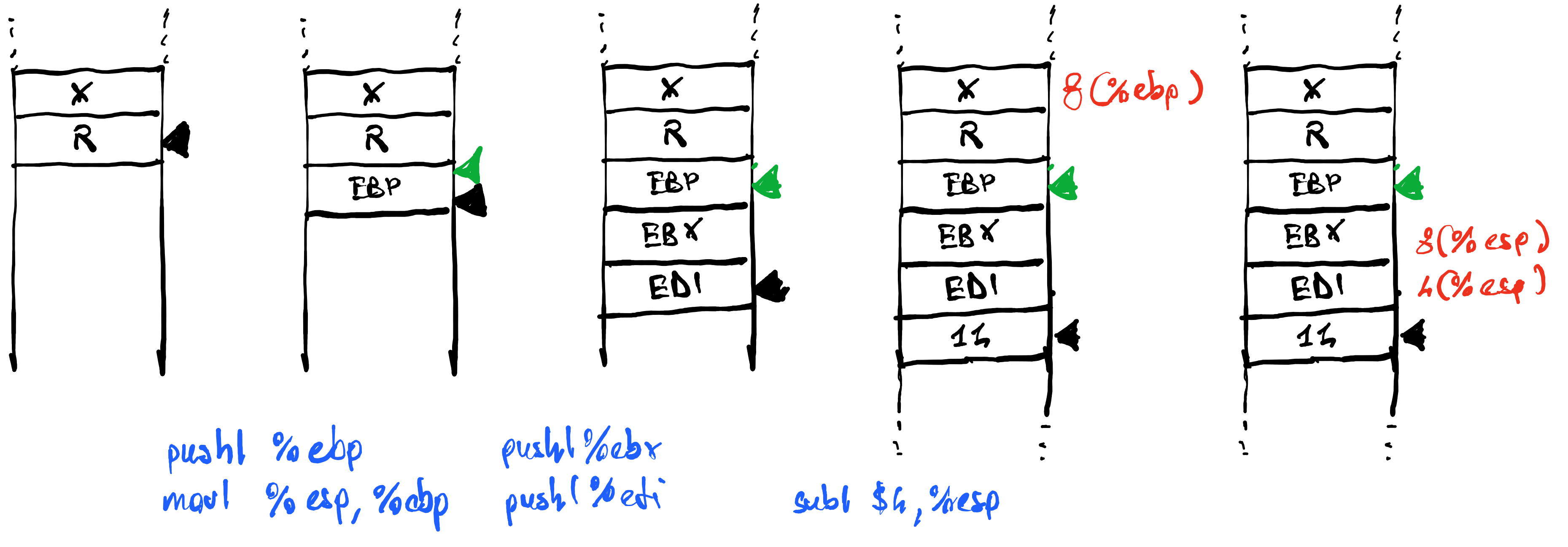
# SC - 26/3/2025

martedì 25 marzo 2025 18:36

Es 5 con stack gestita via SP + BP



## VARIABILI LOCALI

Quando possibile il compilatore usa i registri per conservare i valori delle variabili locali

Ci sono casi in cui questo non è possibile:

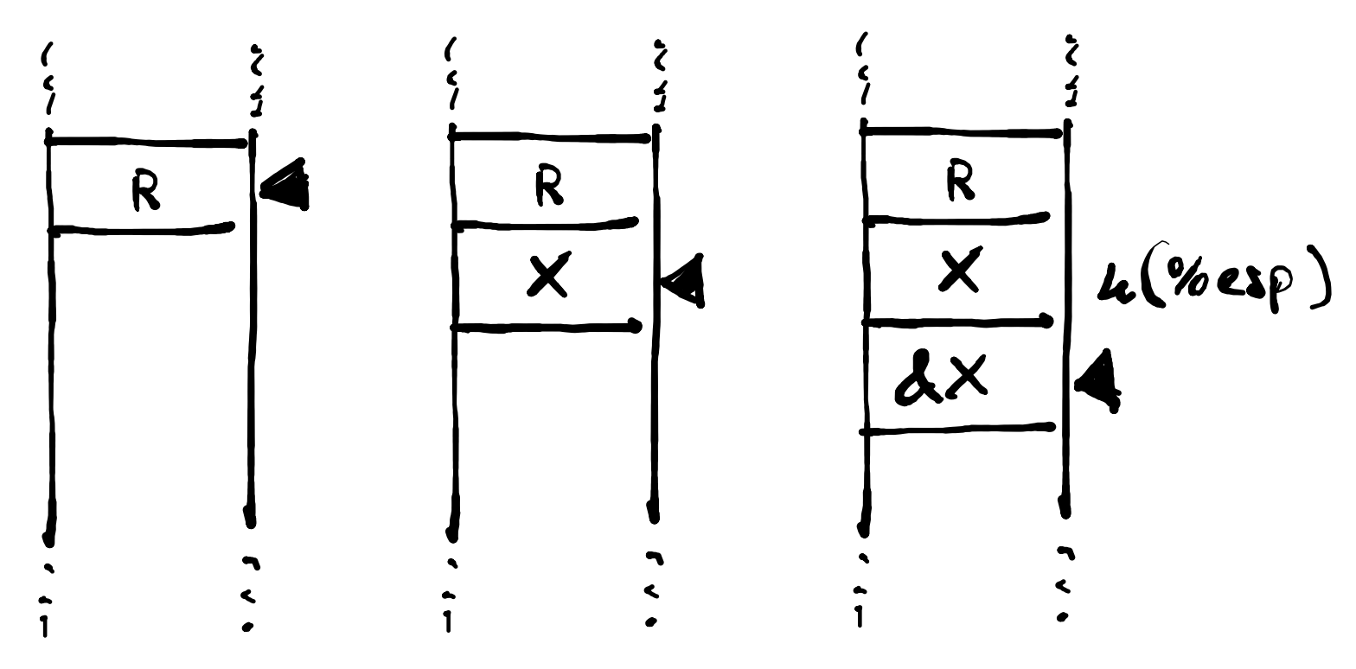
① Non ci sono registri disponibile

② E utilizzato l'operatore "address of" (&)

③ La variabile è un array o una struct

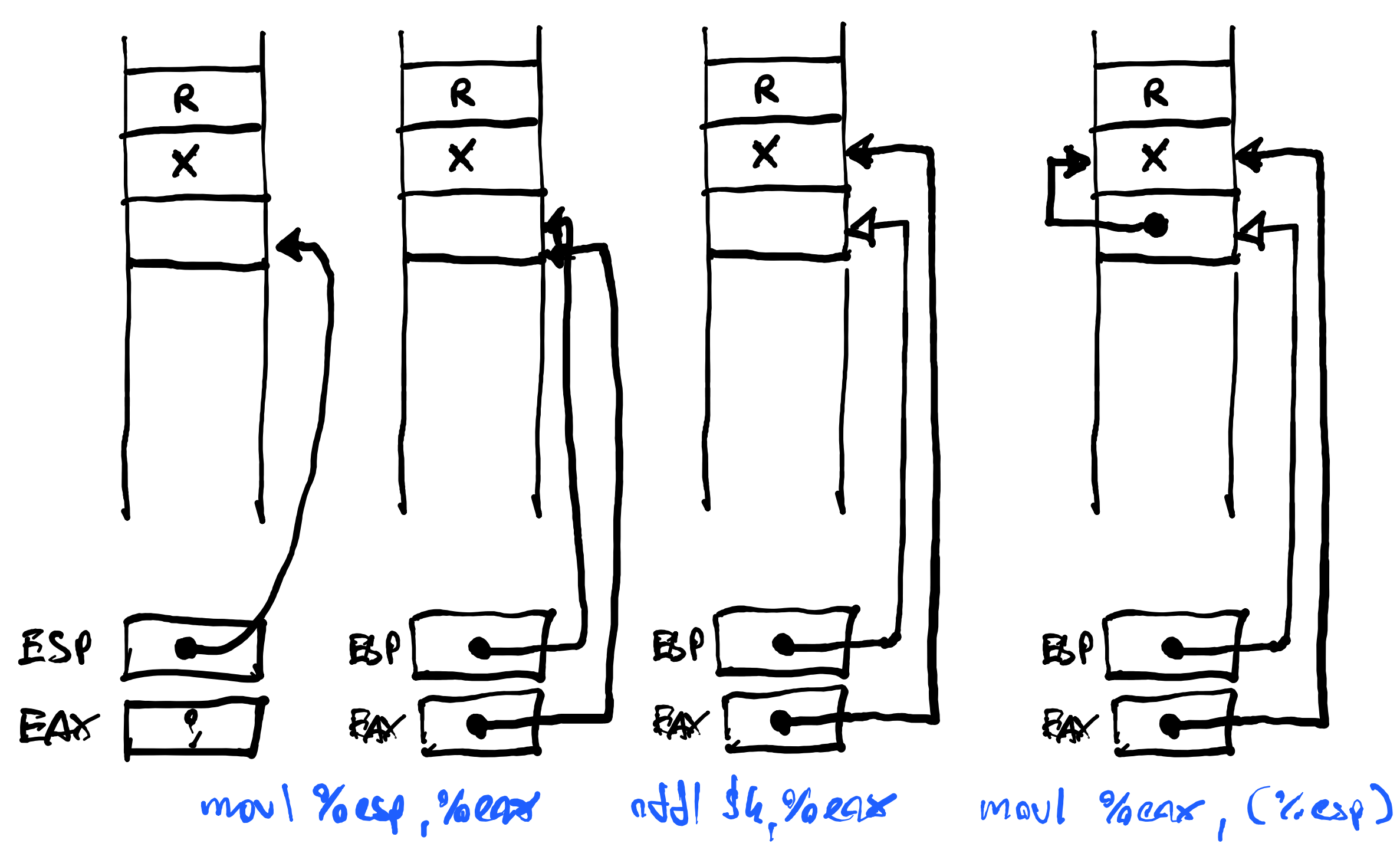
Esempio:

```
int f() {
    int x;
    g(&x);
    return x;
}
```



.globl f
f:

subl $8 , %esp # prologo
~~addl $4, %esp~~
① movl %esp , %eax
② addl $4 , %eax
movl %eax , (%esp)
call g
movl 4(%esp) , %eax
addl $8 , %esp # epilogo
ret



movl %esp , %eax
addl $4 , %eax
} → leal 4(%esp) , %eax

LEA = Load Effective Address

leal O(B, I, S) , D ⟶ movl B , D
D = D + O
D = D + I*S

↓ indirizzo B + I*S + O

**NOTA:** LEA calcola un indirizzo di memoria SENZA accedere alla memoria

| | |
|---|---|
| leal 4(%esp) , %eax | movl 4(%esp) , %eax |
| calcola %esp+4 e lo scrive in %eax | calcola %esp+4, legge da memoria il valore presente a quell'indirizzo e lo scrive in %eax |

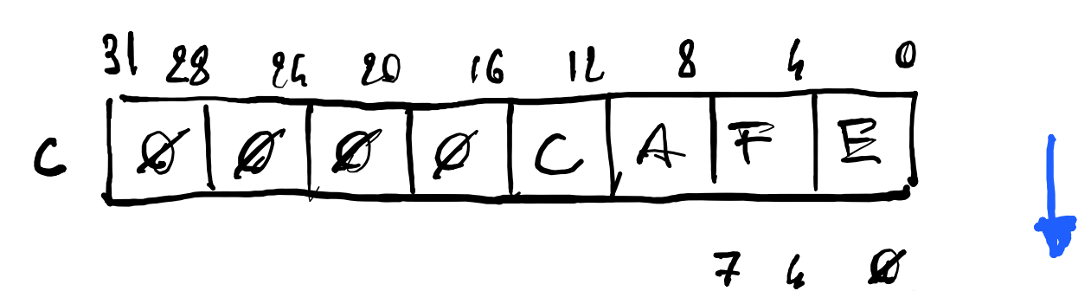
**NOTA:** Se usiamo LEA per calcoli aritmetici lo status register (EFLAGS) non viene alterato

## CONVERSIONE DI TIPO (CASTING)

① OPERAZIONE DI TRONCAMENTO (DOWNCAST)

int c = 0x 0000CAFE; 32 bit

char a = c; ↓ 8 bit



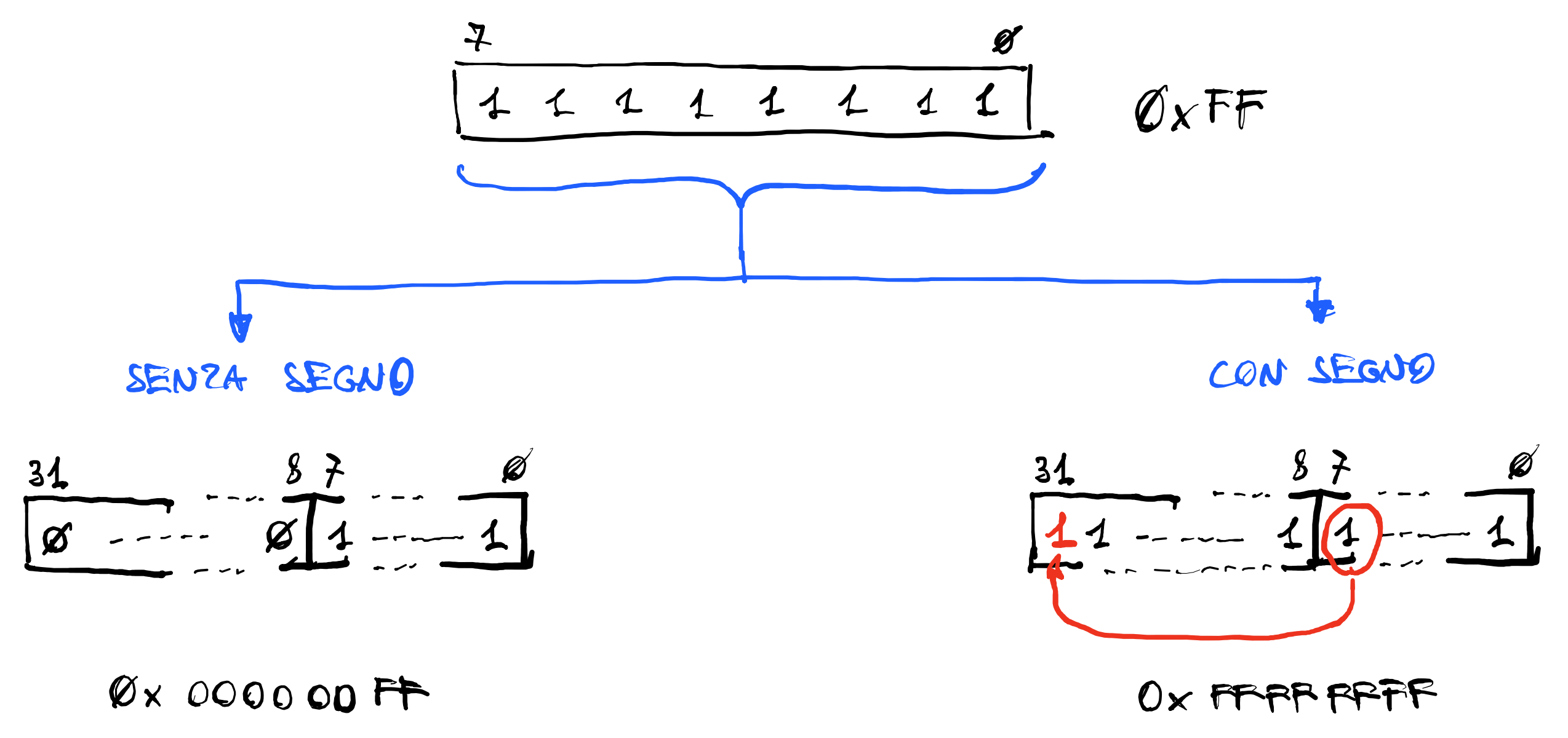
ATTENZIONE! Perdita di informazione

② OPERAZIONE DI PROMOZIONE (UPCAST)

| | | |
|---|---|---|
| CON SEGNO | char c = 's';<br>int a = c; | 8 bit ↓ 32 bit |
| SENZA SEGNO | unsigned char c='s'<br>unsigned int a=c | 8 bit ↓ 32 bit |

\+ NESSUNA PERDITA DI INFORMAZIONE

\- DOBBIAMO GARANTIRE CHE IL SEGNO SIA PRESERVATO



In ASM eseguiamo l'upcast con due istruzioni:

movsxy S , D — copia gli x bit meno significativi della sorgente ad una destinazione grande y bit con estensione del segno

movzxy S , D — copia gli x bit meno significativi della sorgente ad una destinazione grande y bit senza estensione del segno

NOTA BENE: S non può essere un valore immediato
D può essere solo un registro

OPERAZIONI CON SEGNO

| a←c | char c | short c | int c |
|---|---|---|---|
| char a | movb %cl , %al | | |
| short a | movsbw %cl, %ax | movw %cx , %ax | |
| int a | movsbl %cl, %eax | movswl %cx %eax | movl %ecx,%eax |

OPERAZIONI SENZA SEGNO

| a←c | char c | short c | int c |
|---|---|---|---|
| uns. char a | movb %cl , %al | | |
| uns. short a | movzbw %cl, %ax | movw %cx , %ax | |
| uns. int a | movzbl %cl, %eax | movzwl %cx %eax | movl %ecx,%eax |

Esempio

movb $0x80 , %cl

movsbl %cl , %eax

movb $0x40 , %cl

movsbl %cl , %eax