

**ALLINEAMENTO STRUTTURE IN MEMORIA (CONFWMM...)**

Esempio:

```
typedef struct nodo nodo;
struct nodo {
    int elem;
    nodo * next;
};
```

base  
+0 | x x x x |  
+4 | x x x x |  
sizeof nodo = 8

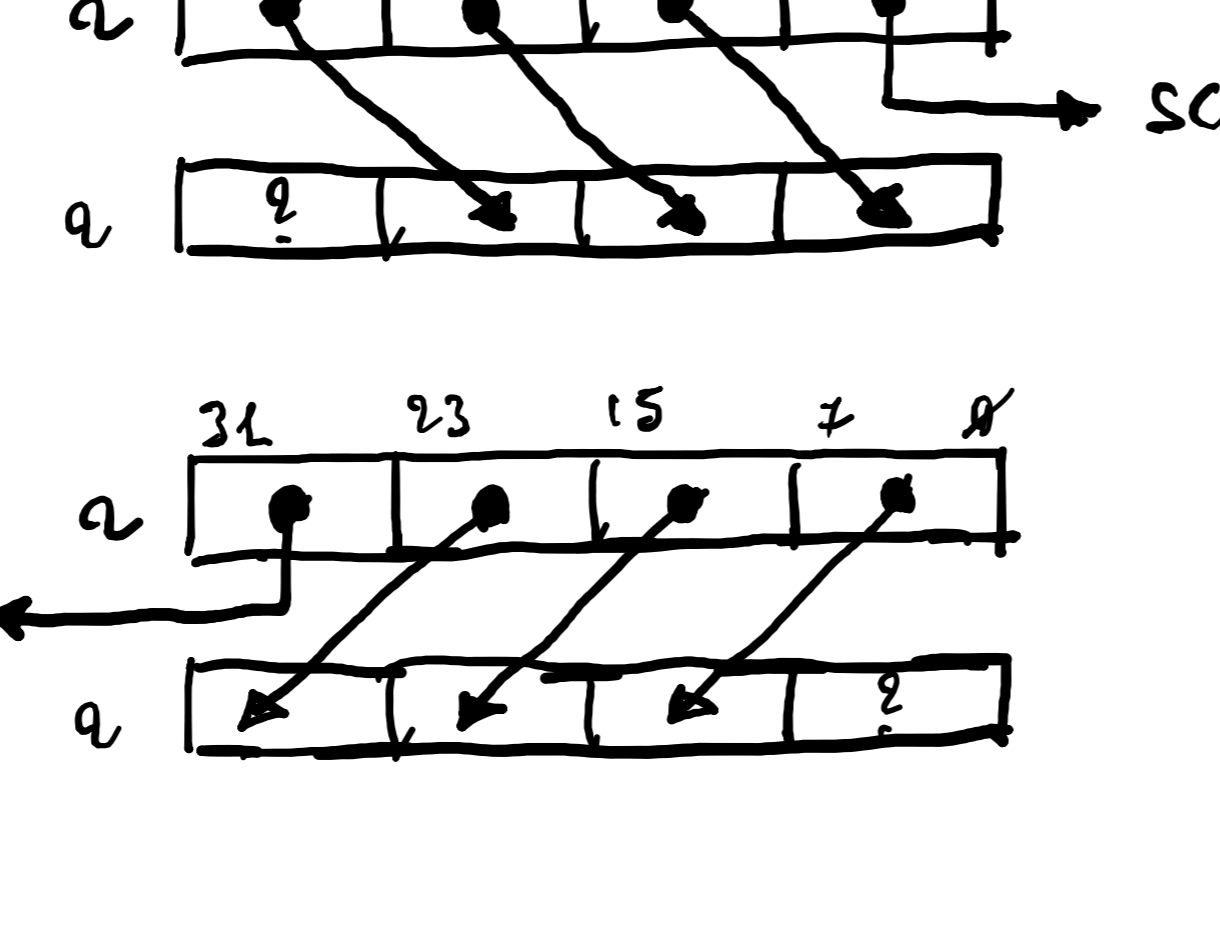
```
int sum(nodo * p) {
    int count = 0;
    for (; p; p = p->next)
        count += p->elem;
    return count;
}
```

```
int sum(nodo * p) {
    nodo * edx = p;
    int ecx = 0; //count
    L:
    if (edx == NULL)
        goto E;
    int ecx = ecx + edx->elem;
    edx = edx->next;
    goto L;
    E:
    return ecx;
}
```

```
global sum;
sum:
    movl 4(%esp), %edx
    xarl %ecx, %ecx
    L:
    Testl %edx, %edx
    je E
    movl (%edx), %ecx
    addl %ecx, %ecx
    movl 4(%edx), %edx
    jmp L
    E:
    ret
```

**ISTRUZIONI SHIFT**

```
int a = ...;
a = a >> 8;
a = a << 8;
```



**SHIFT CON SEGNO**

```
SAL S, D | D ← D << S
SAR S, D | D ← D >> S
```

Inmovi bit meno significativi sono posti pari a 0  
Inmovi bit più significativi sono posti uguale al valore del bit più significativo di D

**SHIFT SENZA SEGNO**

```
SHL S, D | D ← D << S
SHR S, D | D ← D >> S
```

Inmovi bit sono posti pari a 0

Esempio:

```
int c = 0x ABADCAFE;
unsigned d = 0x ABADCAFE;
c = c << 8;      call $8, %ecx    ecx → 0x ADCAFE00
d = d << 8;      shll $8, %ecx   ecx → 0x ADCAFE00
c = c >> 8;      sarl $8, %ecx    ecx → 0x FFADCAFE
d = d >> 8;      shr  $8, %ecx    ecx → 0x 00ADCAFE
```

Esempio:

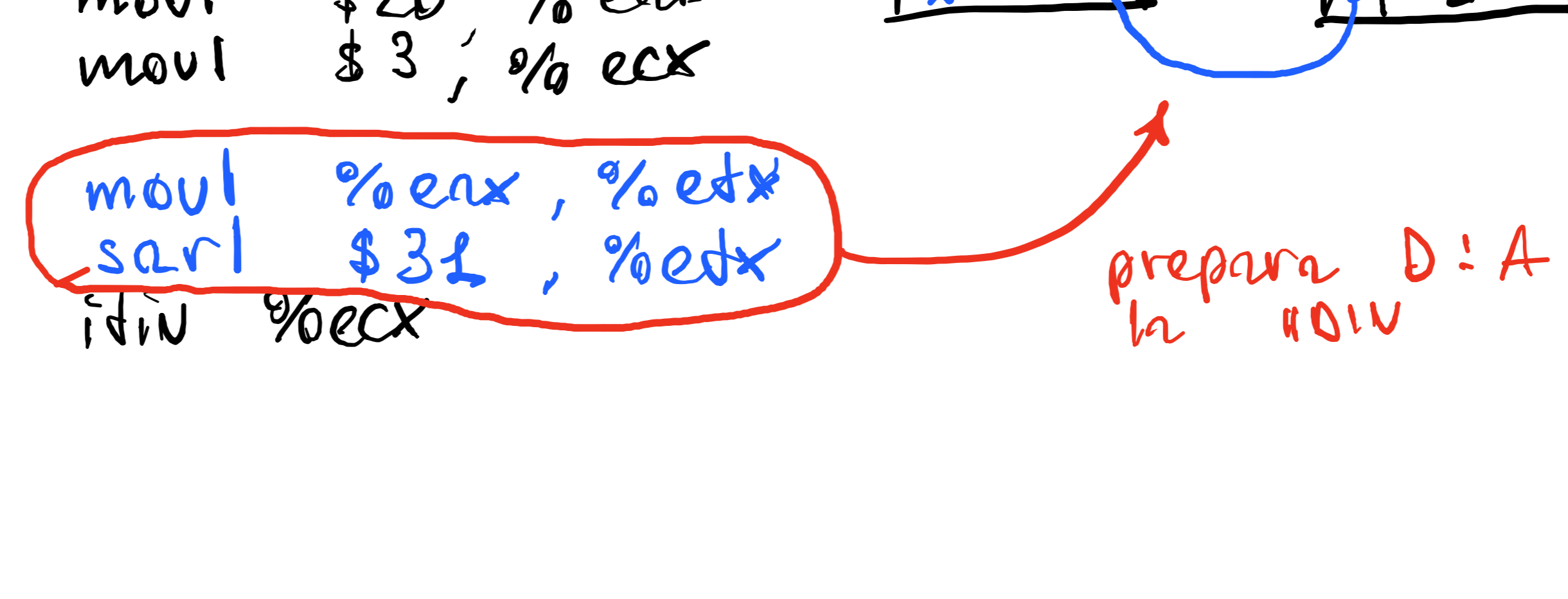
```
int a = 8;      000...001000 → 8
a = a >> 2;     000...000100 → 2 = 8 / 2^2
a >> x ⇒ a / 2^x

int a = 8;      000...001000 → 8
a = a << 2;     000...100000 → 32 = 8 * 2^2
a << x ⇒ a * 2^x
```

**ISTRUZIONE IDIV**

```
IDIV S | A ← D:A/S quoziente
        | D ← D:A%S resto
```

```
int a = 20;
int c = 3;
a = a/c;
```



**ISTRUZIONE CMOV**

```
if (a > b) y = x;
```

```
CMOVcc S, D | D ← S se cc è vero
              | D ← D se cc è falso
```

non può essere un imm. dove essere un reg. a 16/32 bit

**OTTIMIZZAZIONE DEI PROGRAMMI**

Modificare un programma in modo che la sua esecuzione minimizzi l'uso delle risorse disponibili, cercando di massimizzare le prestazioni, e senza alcun impatto sulla correttezza

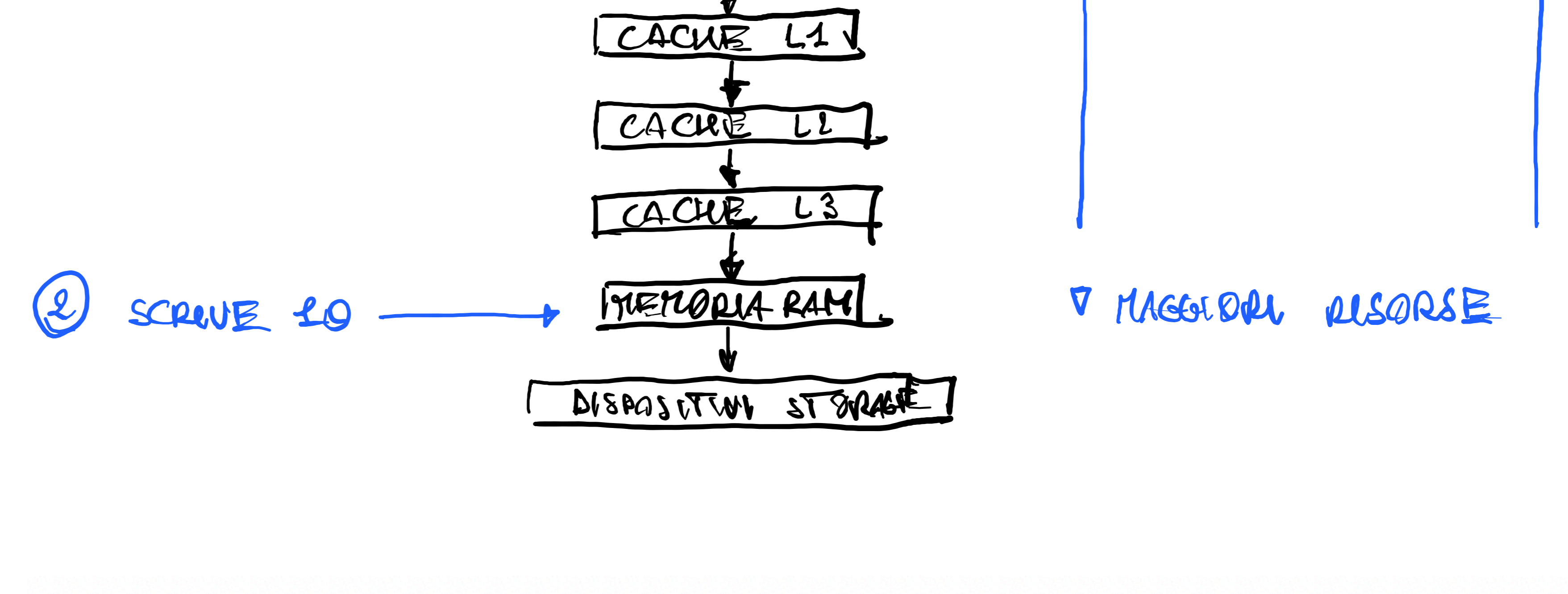
- 1) Come posso valutare l'impatto che una modifica ha nel programma?
  - Come misuro le prestazioni e l'uso delle risorse?
- 2) Come posso scegliere le parti di un programma che voglio ottimizzare?
  - Come identifico in un programma i "punti caldi" (hot spots) meritevoli di attenzione per possibili ottimizzazioni

**MISURE PRESTAZIONALI**

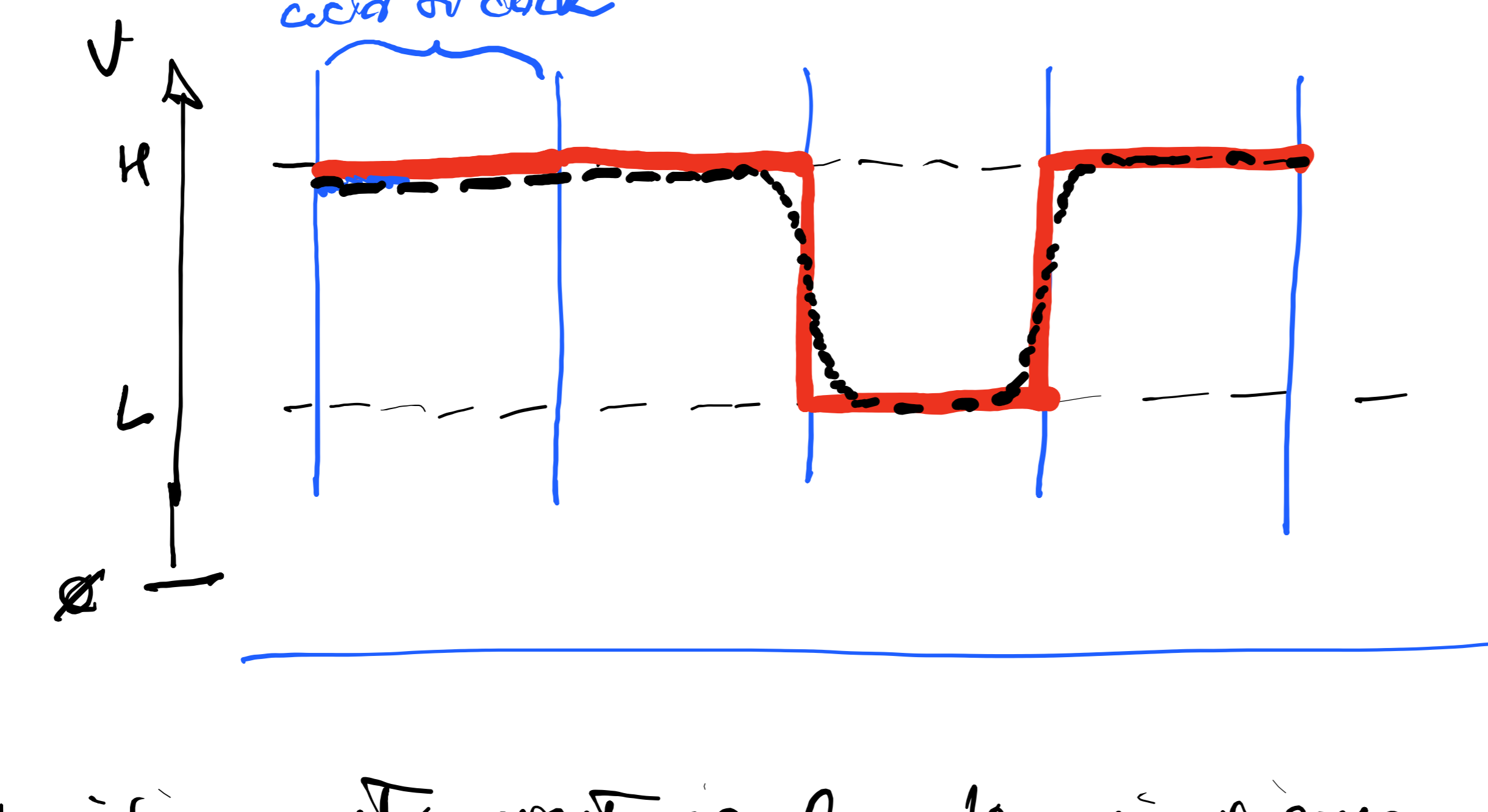
- LATENZA**: Tempo necessario per eseguire il programma
- THROUGHPUT**: numero di operazioni completate per unità di tempo
- SPAZIO**: memoria utilizzata
- ENERGIA**
- .....

**ANALISI PRESTAZIONALE DEL PROGRAMMA**

Quanto tempo richiede l'esecuzione di movl \$10, (%eax)?



Evento <sup>20</sup>	Latenza effettiva	Latenza riscalata
Ciclo di clock	0.4 nsec	1 sec
Accesso cache L1	0.9 nsec	2 sec
Accesso cache L2	2.8 nsec	7 sec
Accesso cache L3	28 nsec	1 min
Accesso RAM DDR3 DIMM	100 nsec	4 min
Accesso disco SSD	50-150 μsec	1.5-4 giorni
Accesso disco a rotazione	1-10 msec	1-9 mesi
Invio pacchetto Internet continentale/intercontinentale	65-141 msec	5-11 anni



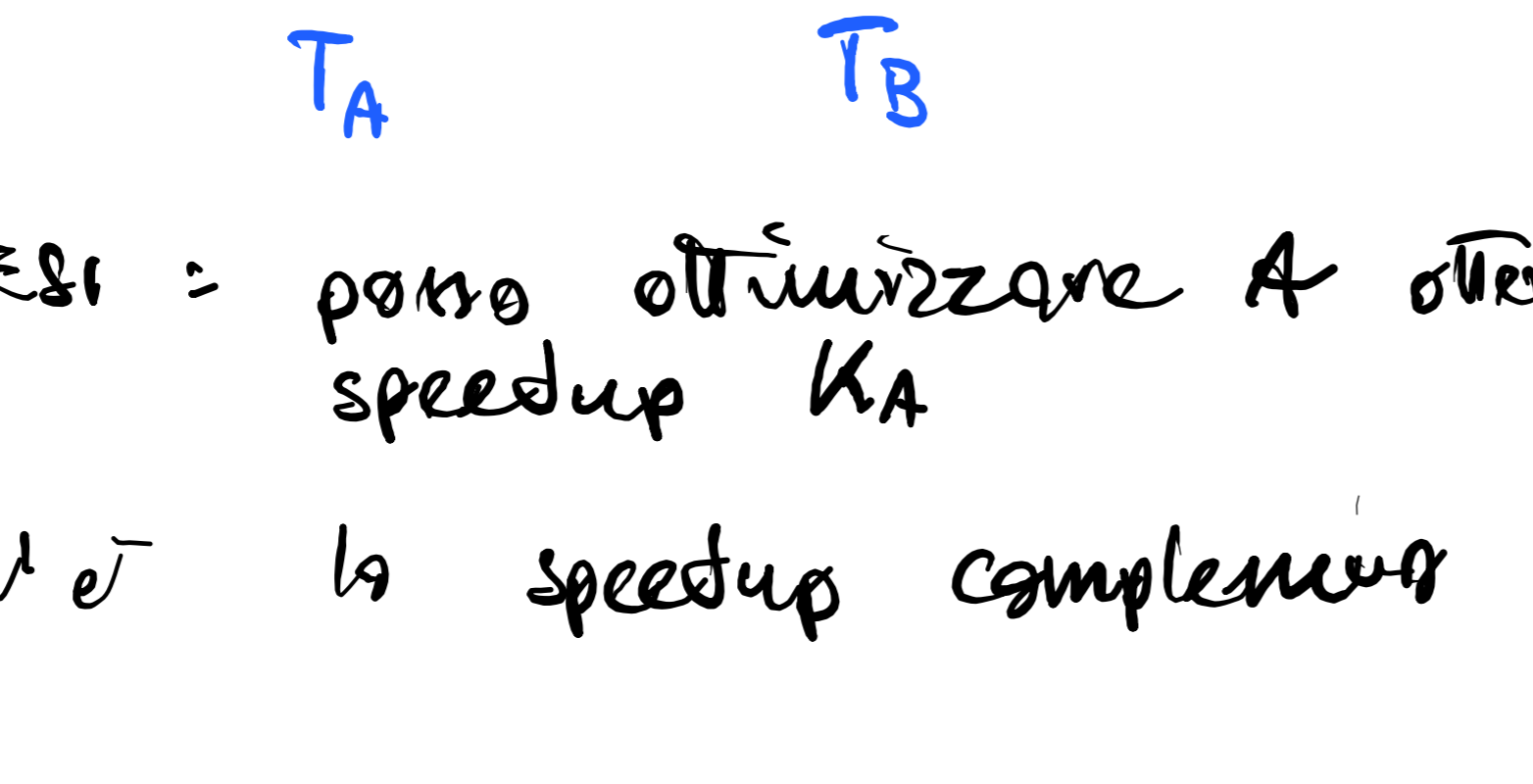
Il miglioramento prestazionale lo misuriamo come "speed up"

$$SPEEDUP_0 = \frac{T_0}{T'} \leftarrow \begin{matrix} \text{latenza del programma } 0 \text{ non ottimizzato} \\ \text{con applicata ottimizzazione} \end{matrix}$$

Otteniamo un numero adimensionale  $2,5378 \approx 2x$

**LEGGE DI AMDAHL**

Ci permette di calcolare la speedup complessiva di un programma a fronte dell'ottimizzazione di una delle sue parti



$$T = \alpha T_A + (1-\alpha) T_B$$

IPOTESI = posso ottimizzare A ottenendo per questo modulo una speedup  $K_A$

Qual'è la speedup complessiva S di P?

$$S = \frac{T}{T'} \quad T' = T_A' + T_B = \frac{T_A}{K_A} + T_B = \frac{\alpha T}{K_A} + (1-\alpha) T = \left( \frac{\alpha}{K_A} + 1 - \alpha \right) T$$

$$S = \frac{T}{T'} = \frac{T}{\left( \frac{\alpha}{K_A} + 1 - \alpha \right) T} = \frac{1}{\frac{\alpha}{K_A} + 1 - \alpha}$$

Es:  $\alpha = 0,5 \quad K_A = 2x \rightarrow S = 1,33x$   
 Es:  $\alpha = 0,9 \quad K_A = 6x \rightarrow S = 1,8x$

Qual'è la speedup Teorica massima?

$$\lim_{K_A \rightarrow \infty} \frac{1}{\frac{\alpha}{K_A} + 1 - \alpha} = \frac{1}{1 - \alpha}$$