

## Le basi:

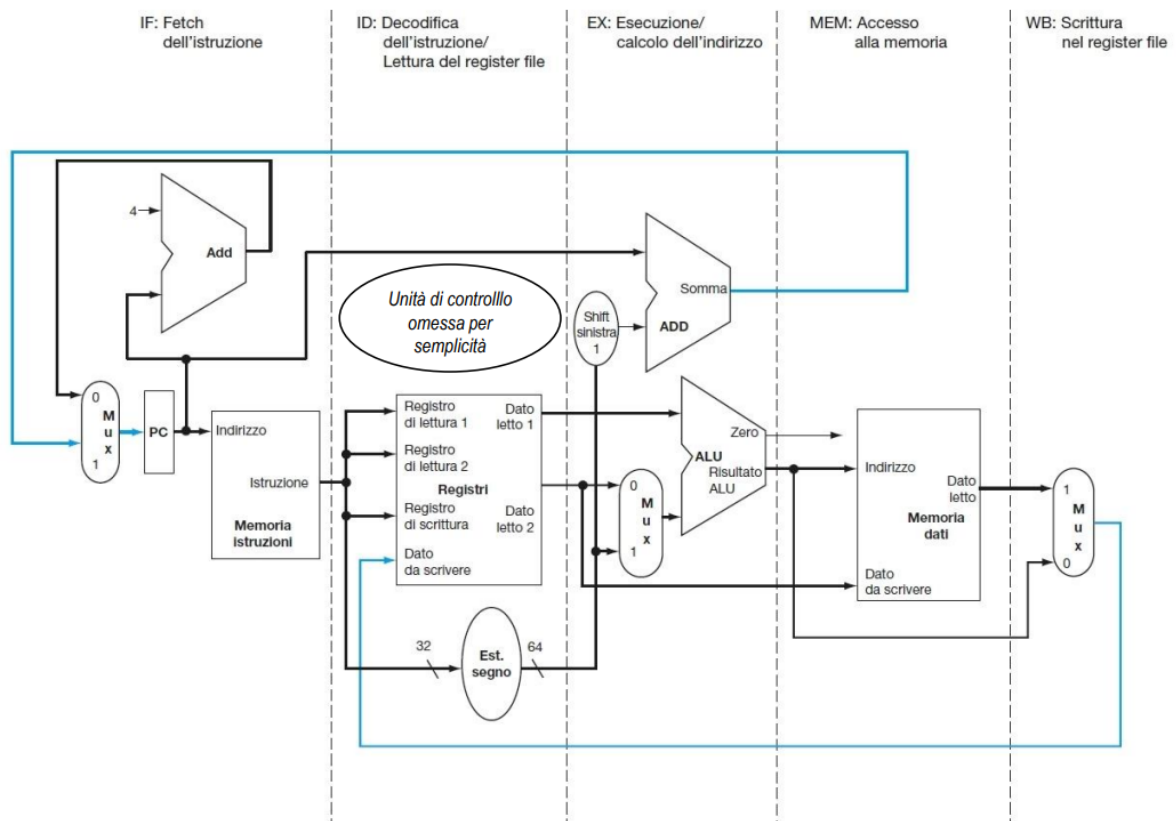
Il cammino di esecuzione più lungo detta il periodo di clock.

$$T_{CPU} = C_{CPU} \cdot T_c = \frac{C_{CPU}}{f_c} = \frac{N_{CPU} \cdot \overline{CPI}}{f_c}$$

- $T_c$  è il periodo di clock
- $f_c$  è la frequenza di clock ==  $1 / T_c$
- $C_{CPU}$  sono i cicli di clock
- $N_{CPU}$  è il Numero di istruzioni eseguite **DINAMICAMENTE**
- $\overline{CPI}$  è il numero di Cicli Per Istruzione (medio)

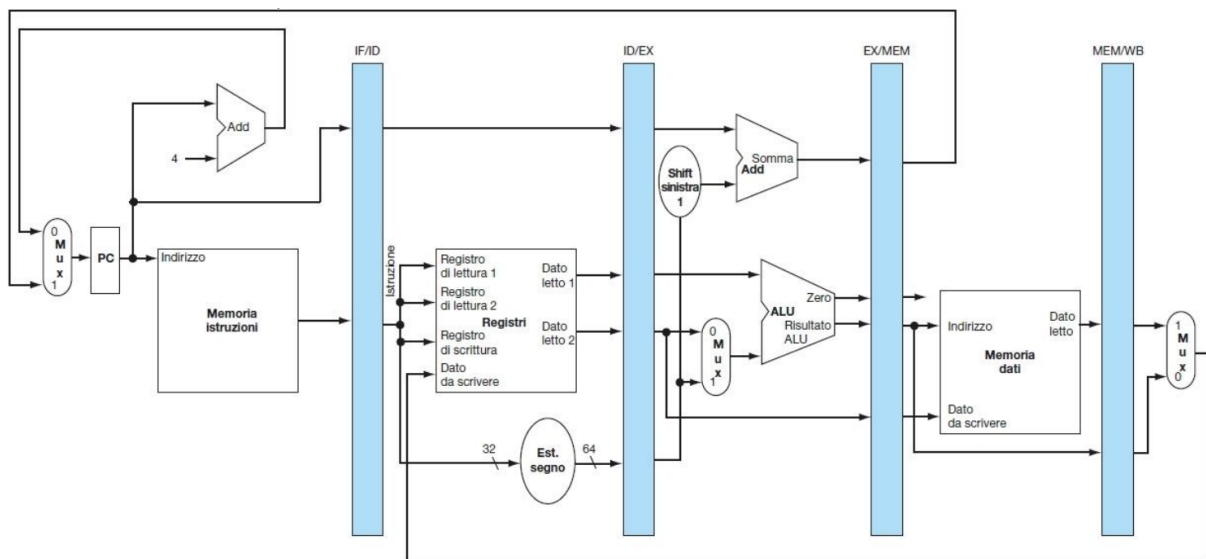
## Pipeline:

La pipeline è una tecnica di esecuzione delle istruzioni che suddivide il ciclo di esecuzione in fasi sequenziali, consentendo il parallelismo e migliorando l'efficienza del processore.



## Registri aggiuntivi:

Versione con pipeline dell'unità di elaborazione.



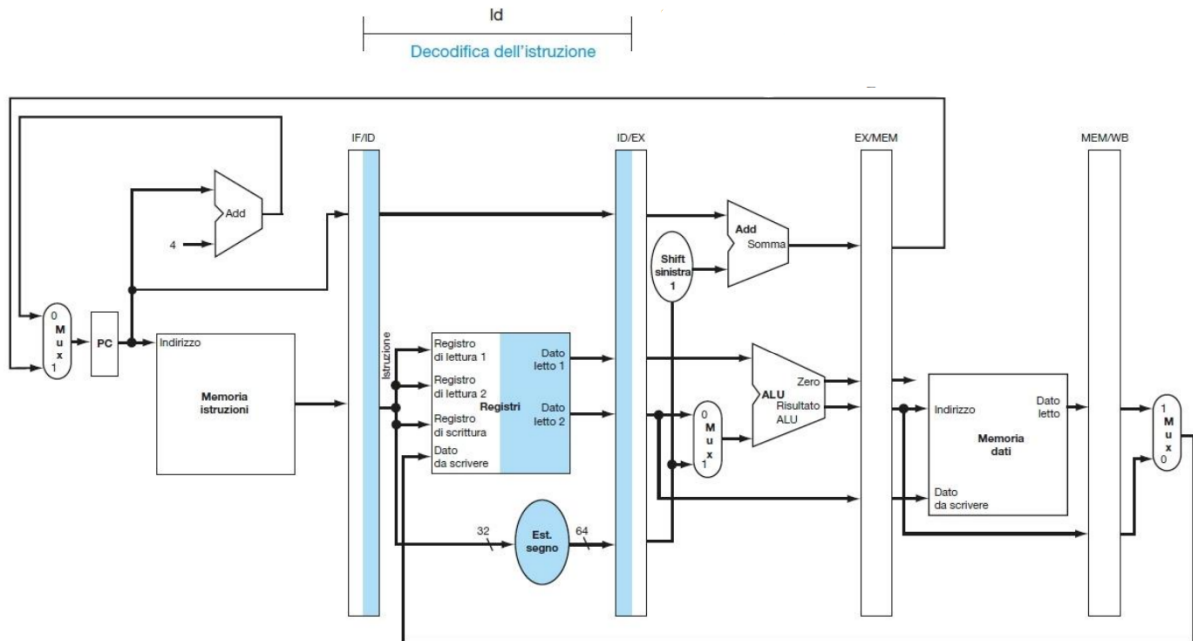
I registri evidenziati in blu separano i diversi stadi della pipeline e per etichettarli si usa il nome dei relativi stadi che separano

## Fetch:

Il PC viene incrementato e riscritto, ma il vecchio valore va salvato in IF/ID perché può essere necessario più avanti, ad esempio nella beq.

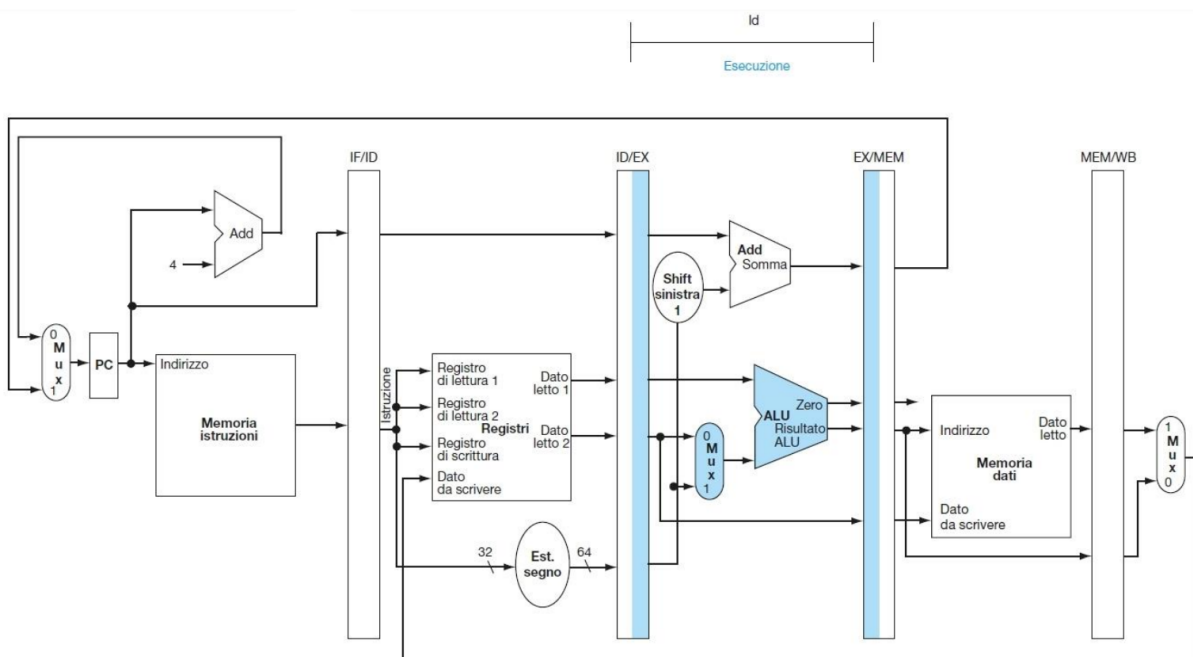
## Decodifica dell'istruzione:

Viene effettuata soltanto la lettura.

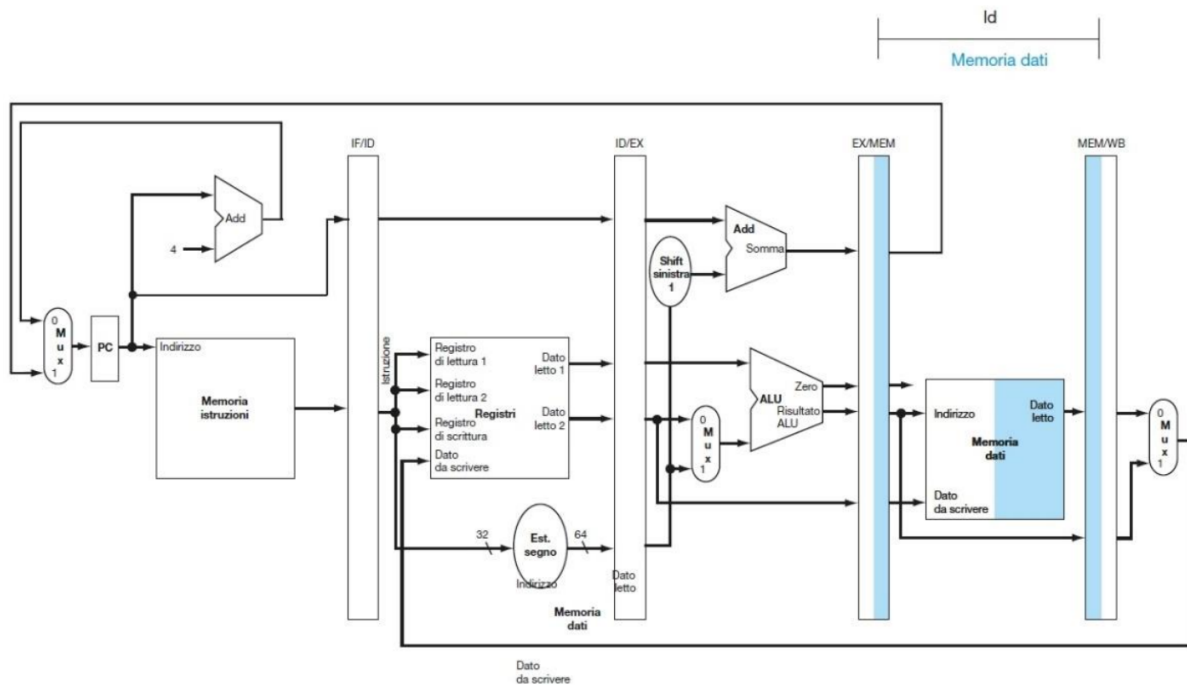


## Esecuzione aritmetica:

Il contenuto del registro viene sommato al contenuto del campo immediato dotato di segno. Il risultato della somma viene scritto nel registro di pipeline EX/MEM

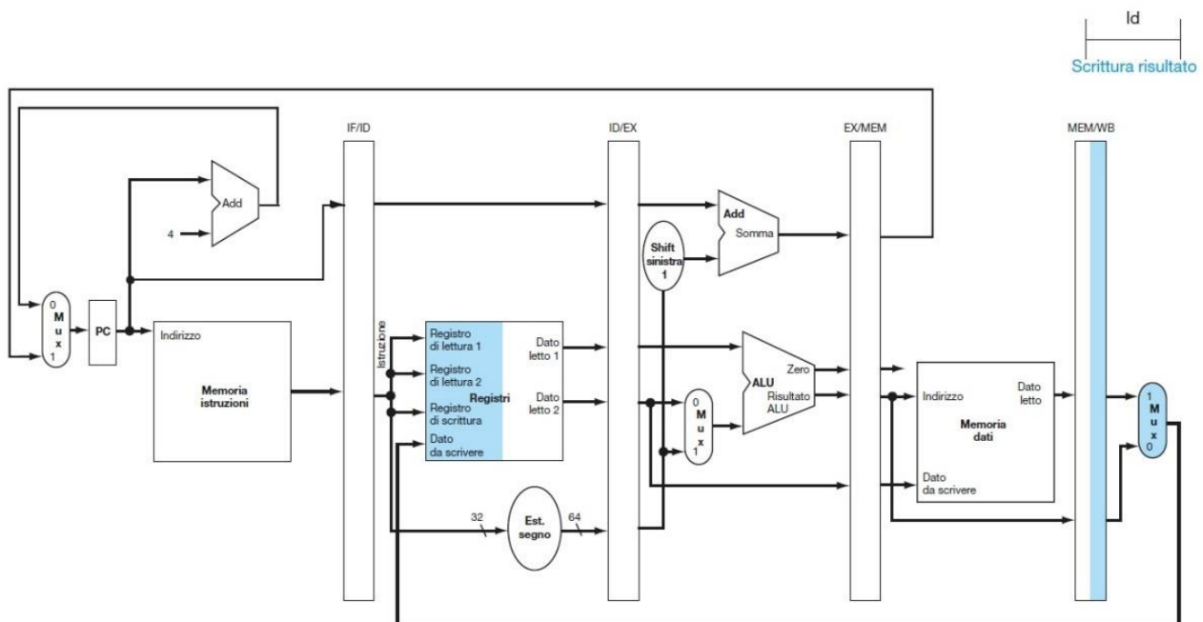


## Accesso alla memoria:



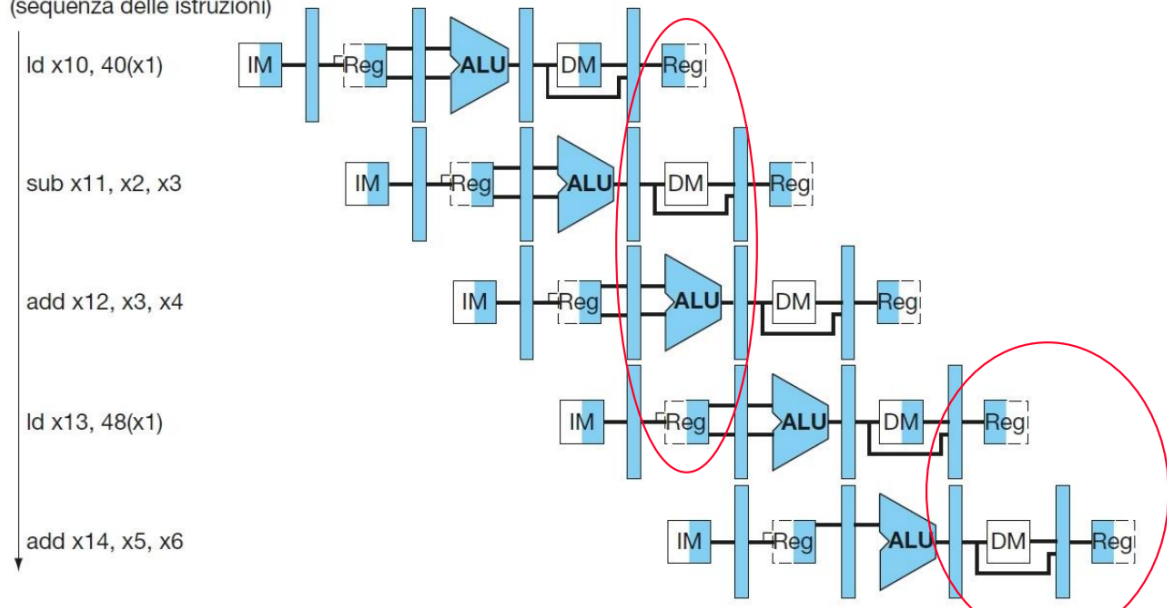
## Scrittura nel register file:

La memoria dati viene letta utilizzando l'indirizzo contenuto nel registro di pipeline EX/MEM e il dato letto viene scritto nel registro di pipeline MEM/WB. Nello stadio WB il dato viene letto dal registro di pipeline MEM/WB e scritto nel register file.



## Esempio di esecuzione di un programma con pipeline:

Ordine di  
esecuzione  
del programma  
(sequenza delle istruzioni)



Aggiunta dell'unità di controllo:

Istruzione	Segnali di controllo dello stadio di esecuzione/calcolo dell'indirizzo		Segnali di controllo dello stadio di accesso alla memoria dati			Segnali di controllo dello stadio di scrittura	
	ALUOp	ALUSrc	Branch	MemRead	MemWrite	RegWrite	MemtoReg
Tipo R	10	0	0	0	0	1	0
ld	00	1	0	1	0	1	1
sd	00	1	0	0	1	0	X
beq	01	0	1	0	0	0	X