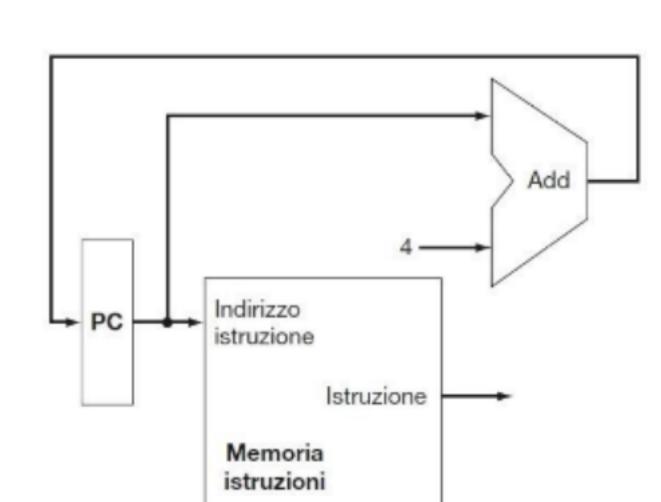
Program counter, Memoria istruzioni e Add:

Sono necessari due elementi di stato per memorizzare e accedere alle istruzioni, e serve un sommatore per calcolare l'indirizzo dell'istruzione successiva.



Program counter:

Un registro a 32 bit che conserva l'indirizzo di memoria della prossima istruzione da eseguire, che viene aggiornato ogni ciclo di clock (non richiede quindi un segnale di controllo esplicito).

Memoria istruzioni: La memoria delle istruzioni fornisce solamente un accesso in lettura, perché essa non viene scritta dall'unità di

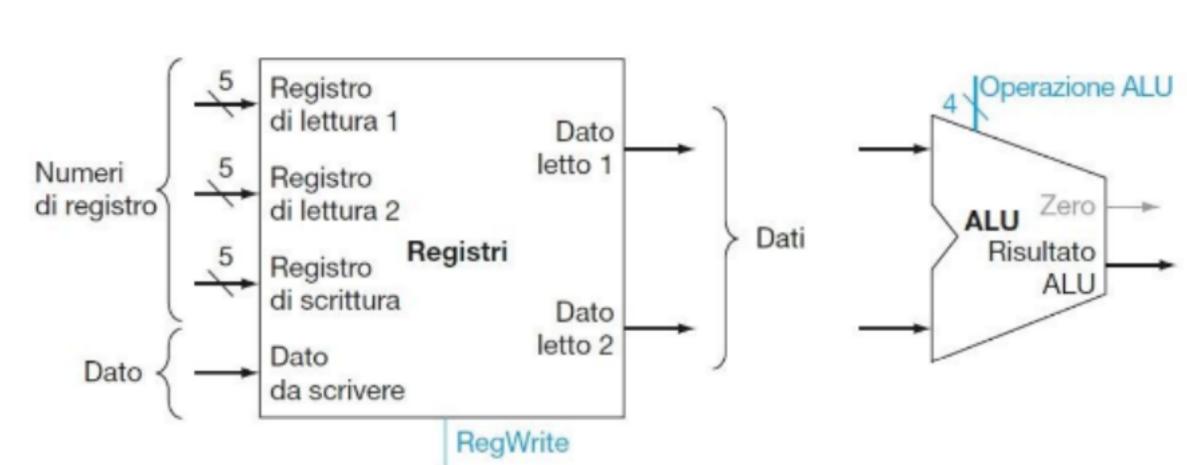
elaborazione, viene quindi trattata come un circuito logico combinatorio.

Add:

È una ALU configurata in modo da eseguire sempre la somma dei suoi due ingressi a 32 bit, e fornisce in uscita il risultato della somma (quindi l'indirizzo della prossima istruzione).

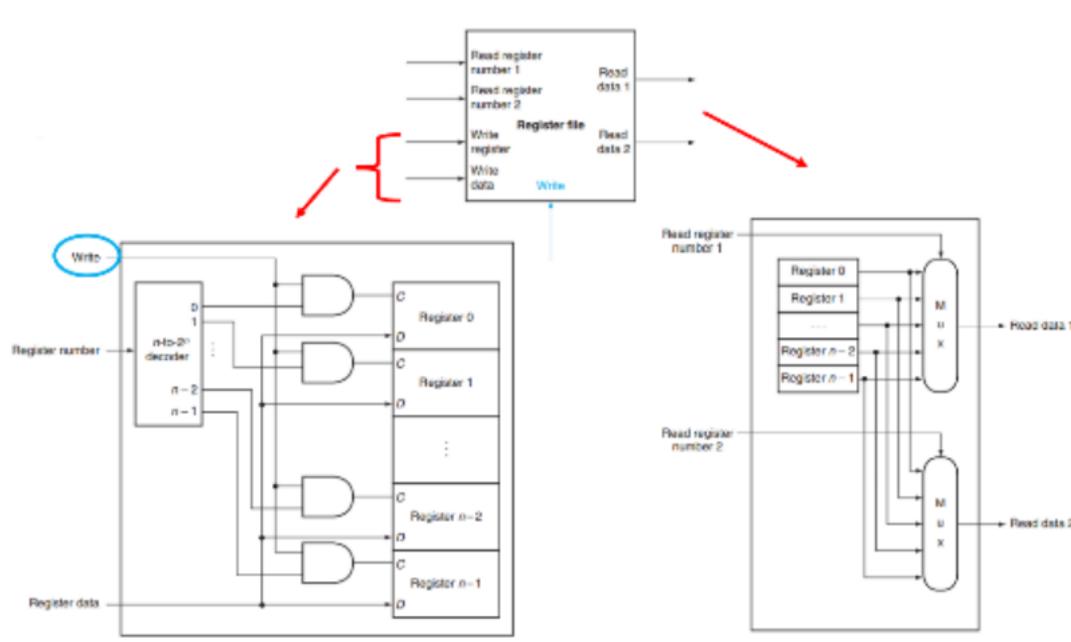
Register file e ALU:

I due elementi richiesti per implementare le operazioni di tipo R (add, sub, and, or) sono il register file e la ALU.



Register file:

Prende in input due registri di lettura, un registro di scrittura, un dato da scrivere e un RegWrite. I due registri di lettura non vengono modificati, vengono soltanto letti e mandati direttamente in output. Il registro di scrittura viene modificato in base al dato da scrivere soltanto quando il RegWrite assume valore 1, altrimenti non modifica il registro (quando assume valore 0). La scrittura avviene sul fronte attivo del clock, quindi tutti gli ingressi interessati devono essere validi sul fronte attivo del clock. La lettura e scrittura viene eseguita nello stesso ciclo di clock.



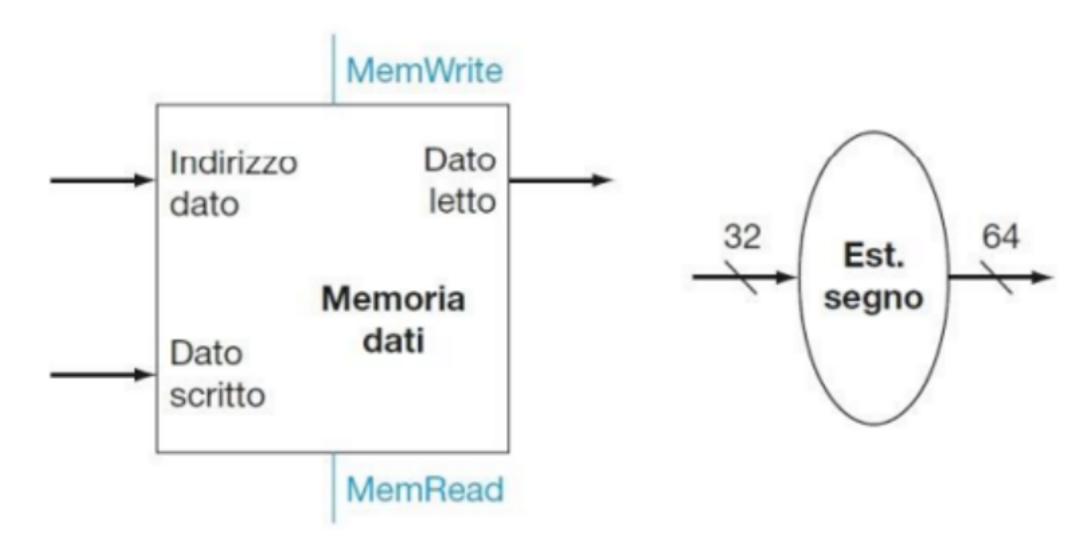
ALU: Prende in input i valori i registri di lettura del register file e ci fa delle operazioni sopra.

- Valori di ingresso:
 - 0010: somma.
 - 0110: sottrazione.
 - 0000: and. 0001: or.

L'output zero viene utilizzato soltanto nel caso della beq, l'unità di controllo attiva la ALUOp (vedi capitolo unità di controllo), così da mandare l'output in Zero e non in risultato ALU.

Memoria dati e Unità di estensione del segno:

Le due unità necessarie per l'implementazione delle istruzioni di load e store: oltre al register file e alla ALU, sono richieste anche un'unità di memoria dati e un'unità di estensione del segno. (è 32 bit l'output dell'est segno)



Memoria dati:

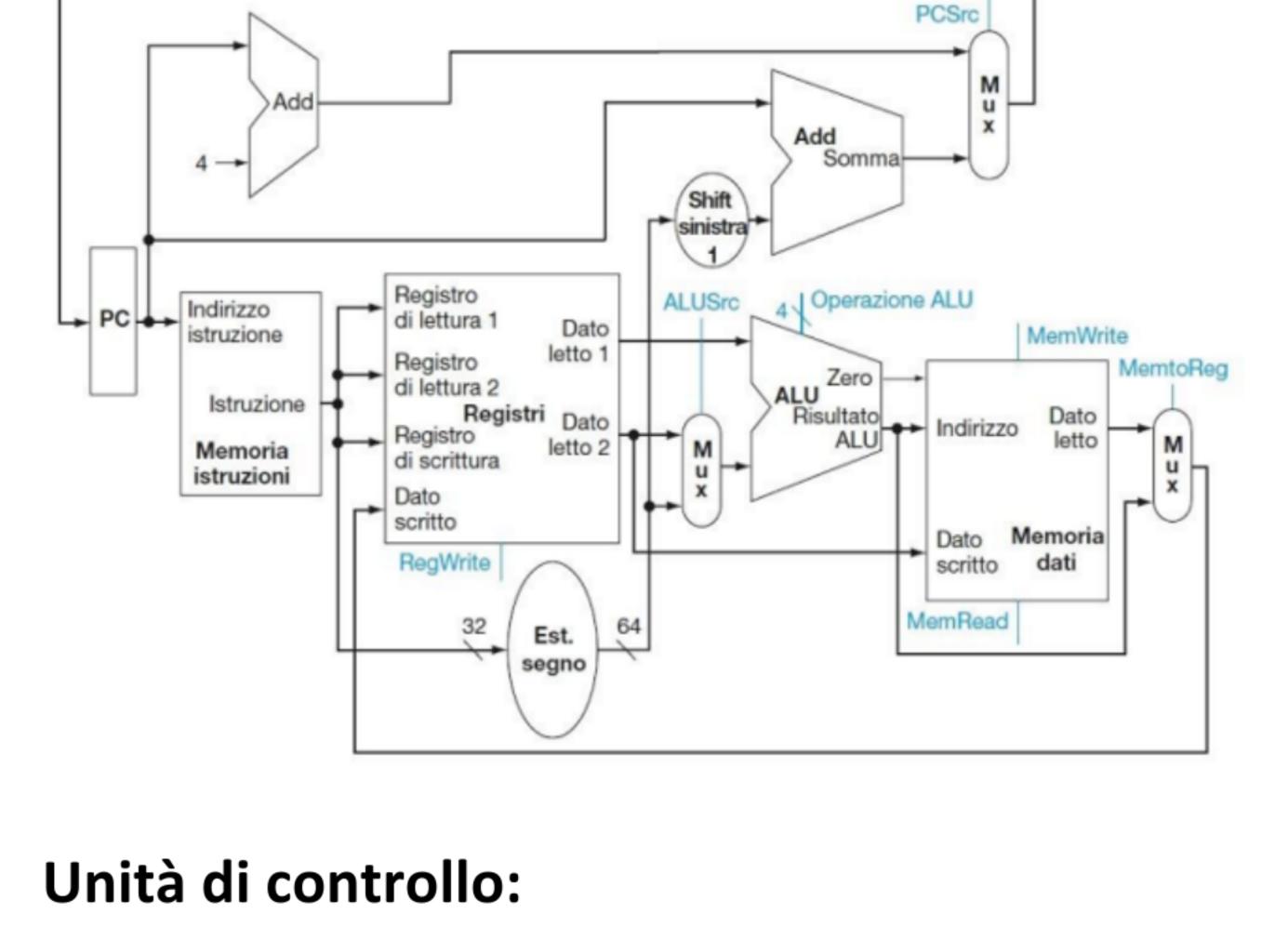
- Indirizzo dato: Questo è l'indirizzo della locazione di memoria a cui il processore desidera accedere. L'indirizzo di memoria viene solitamente fornito da un registro del processore (ad esempio, il registro di indirizzo di memoria, il Program Counter, o altri registri generali). Dato scritto: Se l'operazione è una scrittura nella memoria (store), l'unità di memoria dati riceve i dati che il
- processore desidera scrivere nella memoria esterna. Questi dati vengono tipicamente forniti da un registro del processore. MemWrite e MemRead: Questi segnali di controllo indicano il tipo di operazione di memoria che si desidera
- eseguire, ad esempio se si tratta di una lettura (load) o di una scrittura (store), e possono includere altri segnali di controllo pertinenti. Dato letto: Se l'operazione è una lettura dalla memoria (load), l'unità di memoria dati restituisce i dati letti dalla
- posizione di memoria specificata. Questi dati possono essere inviati a un registro del processore o a un altro componente del sistema.

<u>Unità di estensione del segno:</u> Supporta i numeri negativi

Datapath:

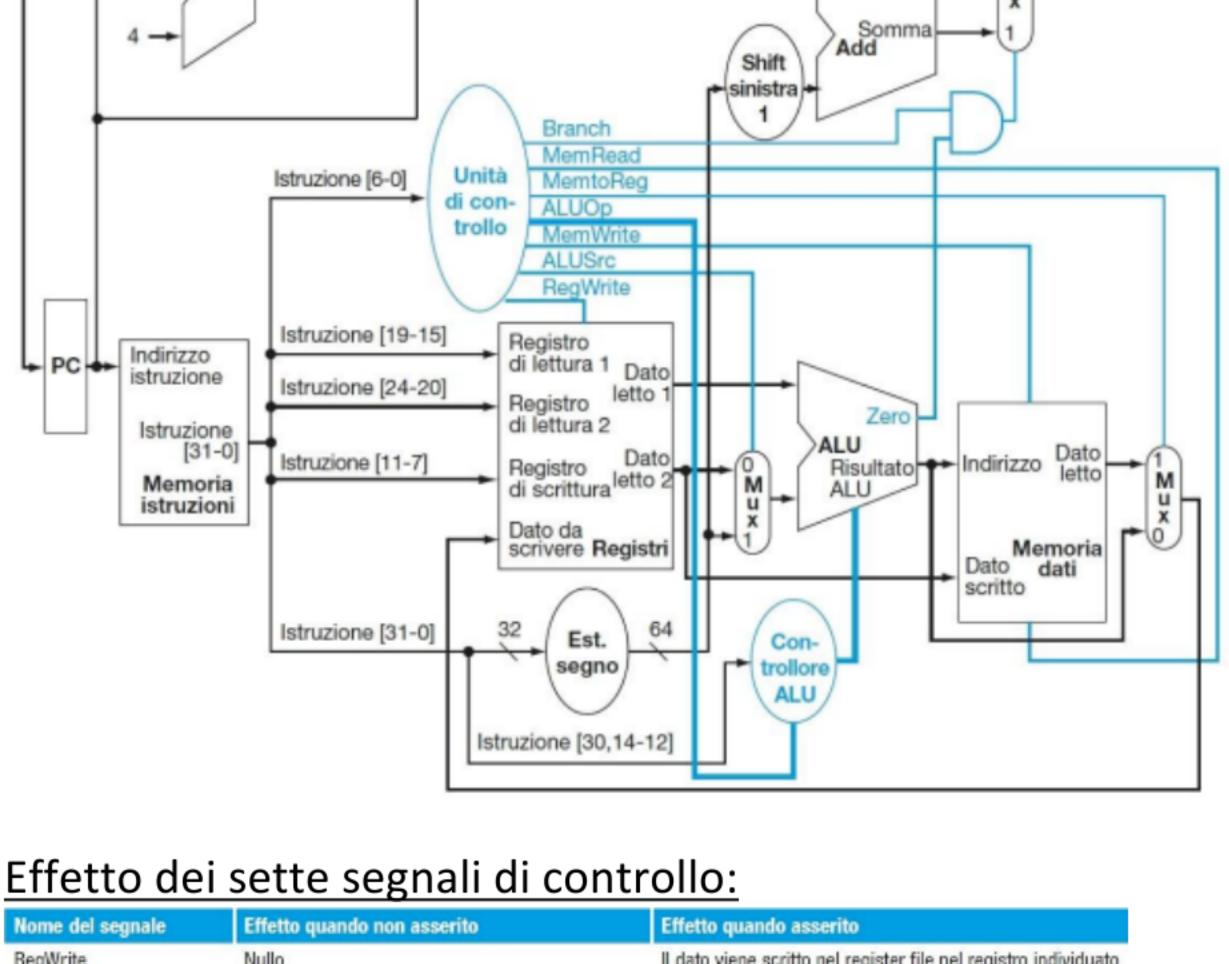
dalle istruzioni appartenenti ai diversi tipi. Questa unità di elaborazione può eseguire le istruzioni di base (load/store di un registro, operazione con la ALU e salti condizionati) in un singolo ciclo di clock. Per integrare i salti condizionati si è resa necessaria solamente l'aggiunta di un altro multiplexer (MUX).

Una semplice unità di elaborazione per il nucleo dell'architettura RISC-V che combina tra loro gli elementi richiesti



È responsabile della gestione del flusso di istruzioni all'interno del processore.

Add



RegWrite	Nullo	Il dato viene scritto nel register file nel registro individuato dal numero del registro di scrittura Il secondo operando della ALU proviene dall'estensione del segno dei 12 bit del campo immediato dell'istruzione			
ALUSrc	Il secondo operando della ALU proviene dalla seconda uscita del register file (Dato letto 2)				
PCSrc	Nel PC viene scritta l'uscita del sommatore che calcola il valore di PC + 4	Nel PC viene scritta l'uscita del sommatore che calcola l'indirizzo di salto			
MemRead	Nullo	Il dato della memoria nella posizione puntata dall'indirizzo viene inviato in uscita sulla linea "Dato letto"			
MemWrite	Nullo	Il contenuto della memoria nella posizione puntata dall'indirizzo viene sostituito con il dato presente sulla linea "Dato scritto"			
MemtoReg	Il dato inviato al register file per la scrittura proviene dalla ALU	Il dato inviato al register file per la scrittura proviene dalla Memoria Dati			

Il valore dei sette segnali di controllo in base al tipo di operazione:

Istruzione	ALUSrc	MemtoReg	RegWrite	MemRead	MemWrite	Branch	ALUOp1	ALU0p0
Tipo R	0	0	1	0	0	0	1	0
ld	1	1	1	1	0	0	0	0
sd	1	Х	0	0	1	0	0	0
beg	0	X	0	0	0	1	0	1