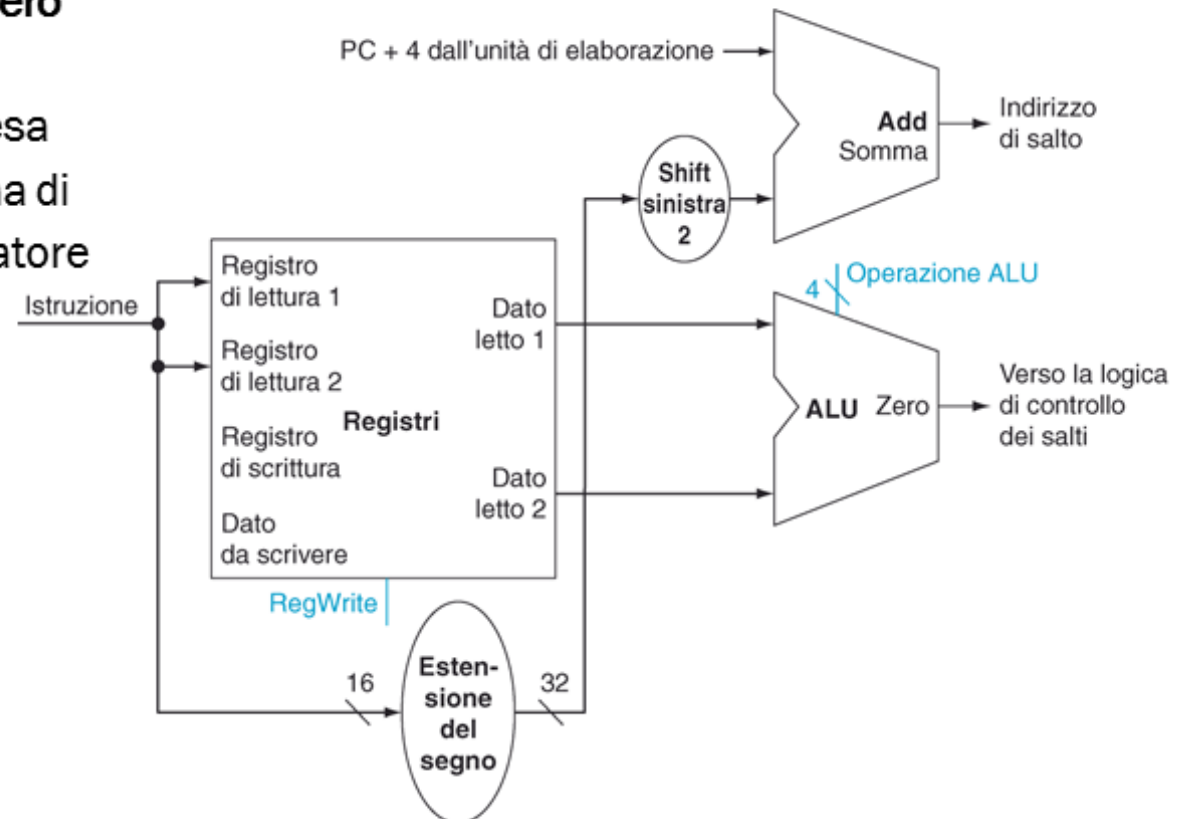


Salti condizionati (beq)

- L'**ALU** viene usata come **comparatore** (svolgendo una **sottrazione**)
- Il segnale **Zero** indica se va fatto il salto
- La destinazione dei salti è un **numero relativo di istruzioni** rispetto alla istruzione seguente, quindi va estesa nel segno e moltiplicata per 4 prima di sommarla al PC + 4 con un sommatore

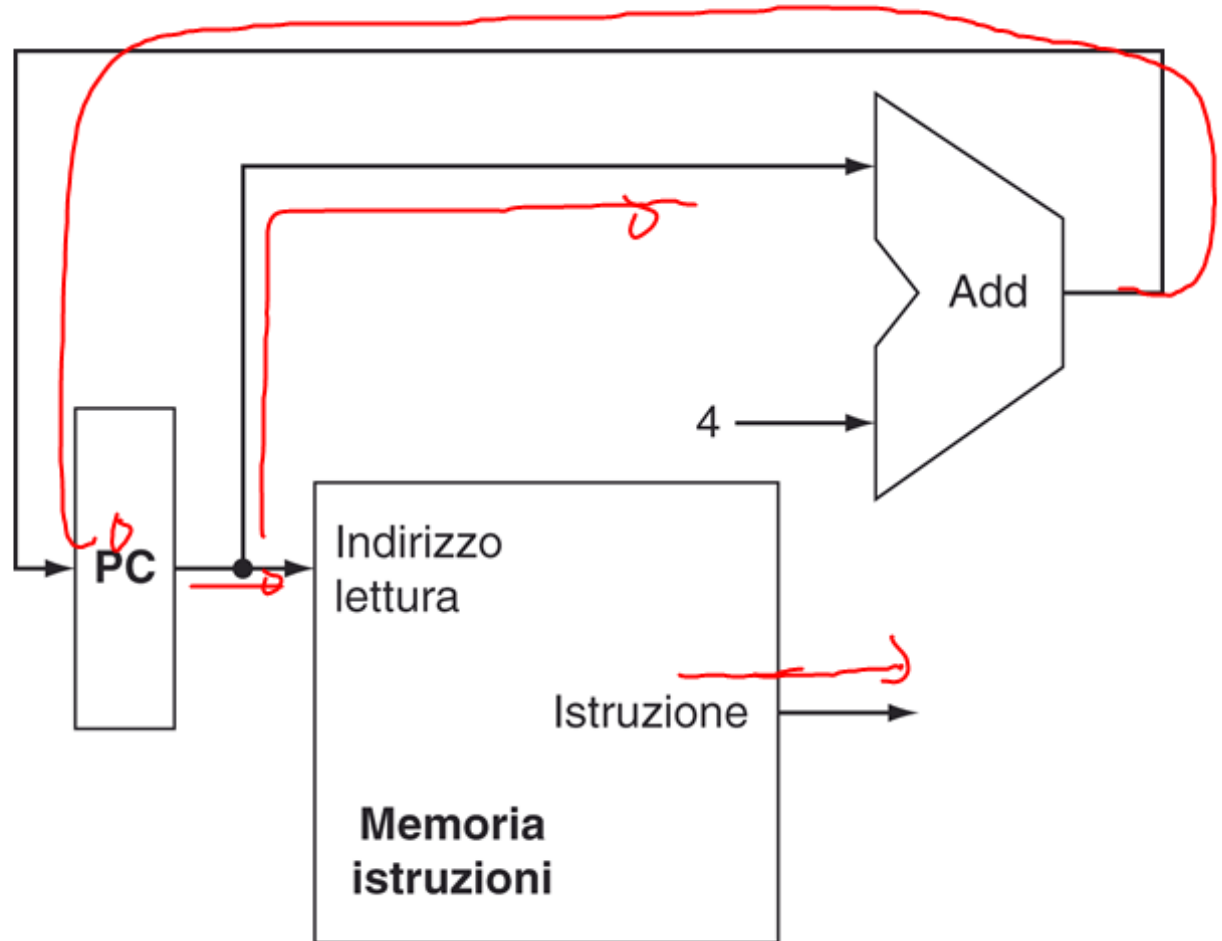


Fetch della istruzione/aggiornamento PC

- PC = indirizzo della istruzione
- Viene letta l'istruzione
- Il PC è incrementato di 4 (1 word)
- e il valore viene rimesso nel PC

NOTE

- tutte connessioni da 32 bit
- mentre viene letta l'istruzione viene già calcolato il nuovo PC



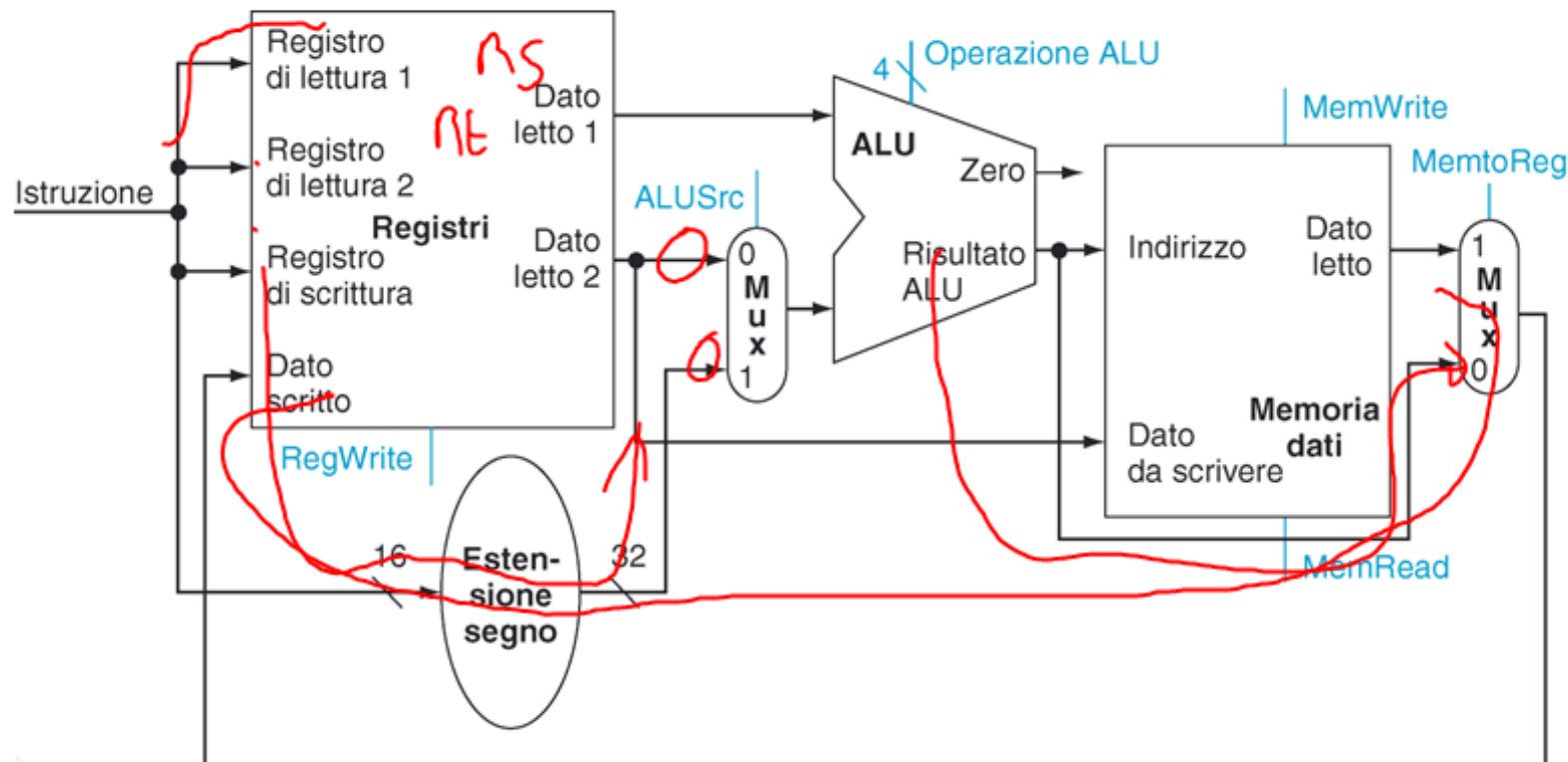
Operazioni ALU e accesso a MEM

La **decodifica** della istruzione è estremamente facile perché i formati I ed R sono quasi uguali

Il secondo argomento della istruzione può essere un registro oppure il campo immediato della istruzione stessa (in questo caso il valore viene esteso nel segno) a seconda del segnale di controllo **ALUSrc** che seleziona la porta corrispondente del MUX

Per calcolare l'indirizzo di accesso alla memoria si usa la stessa ALU, sommando il dato del primo registro (registro base) al campo immediato della istruzione

Il risultato dell'ALU o della lw viene selezionato da **MementoReg** che comanda il MUX a destra



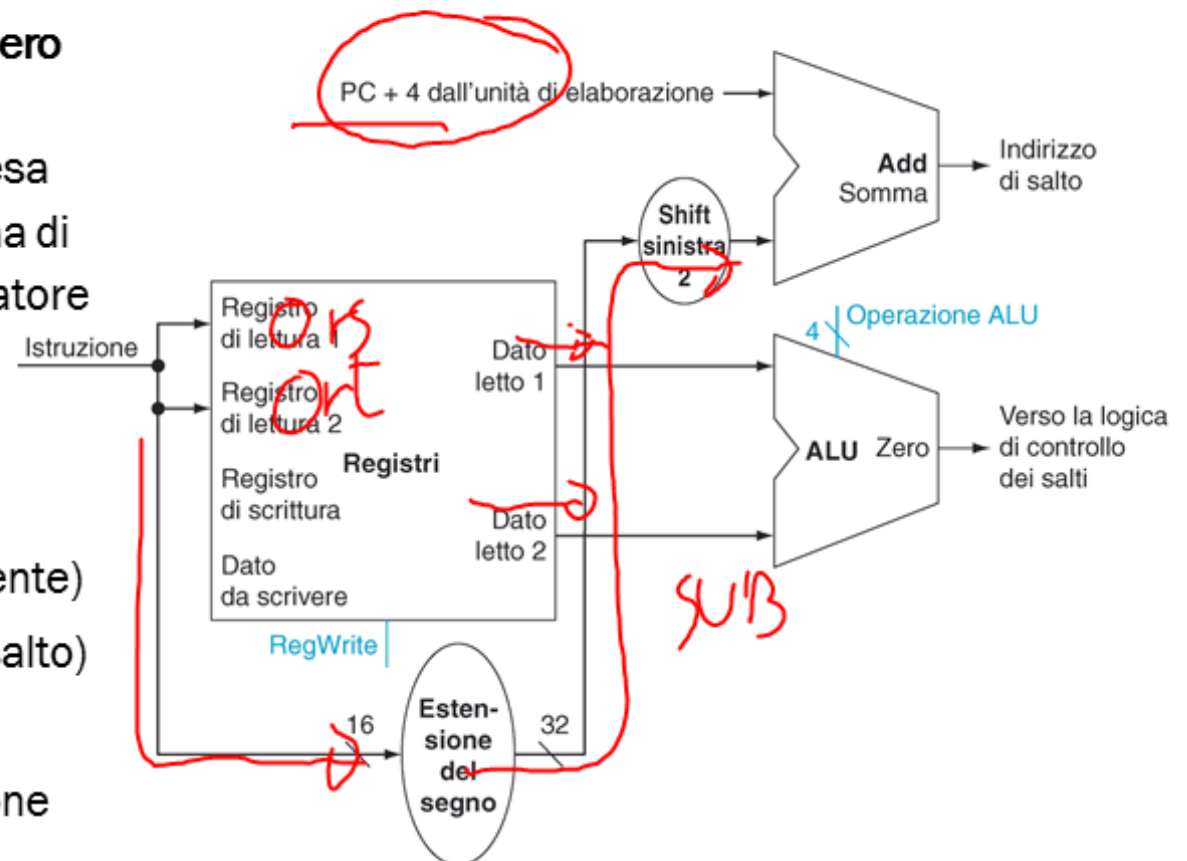
Salti condizionati (beq)

- L'**ALU** viene usata come **comparatore** (svolgendo una **sottrazione**)
- Il segnale **Zero** indica se va fatto il salto
- La destinazione dei salti è un **numero relativo di istruzioni** rispetto alla istruzione seguente, quindi va estesa nel segno e moltiplicata per 4 prima di sommarla al PC + 4 con un sommatore

Notate che il nuovo valore del PC può venire da due diverse sorgenti:

- PC+4 (istruzione seguente)
- oppure uscita dell'adder (salto)

Quindi è necessario un **MUX** di selezione



Mettiamo tutto assieme ...

