



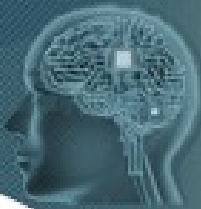
# Architettura degli Elaboratori I - B

## Il Datapath **Istruzioni di Memoria**

Daniel Riccio/Alberto Aloisio  
Università di Napoli, Federico II

15 marzo 2018



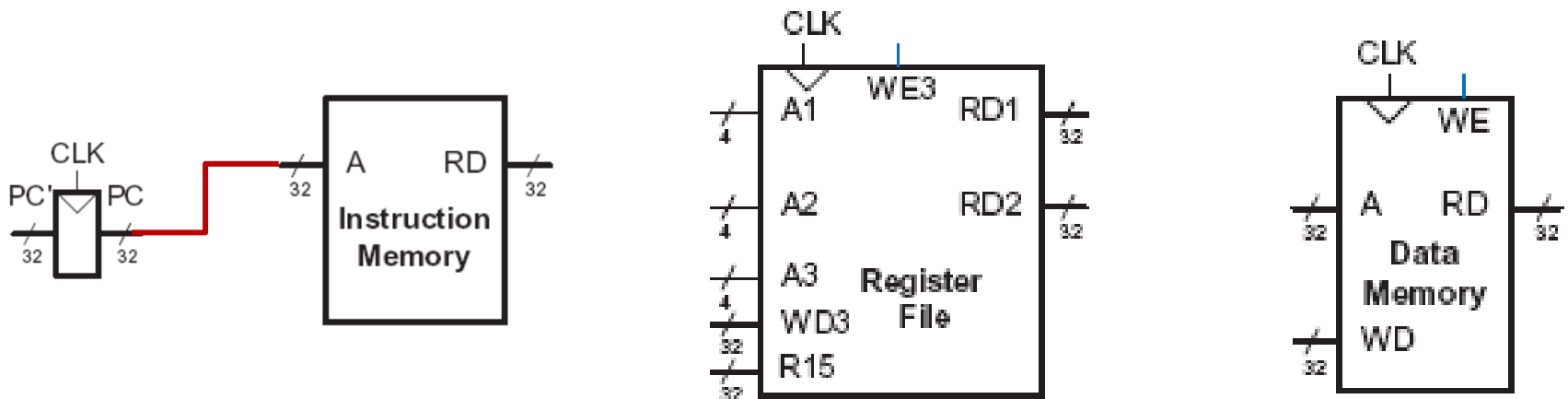


# Il datapath (LDR)

Il **PC** contiene l'indirizzo dell'istruzione da eseguire.

Il primo passo è quello di leggere questa istruzione dalla memoria istruzioni, per cui il PC viene collegato all'indirizzo di ingresso della memoria di istruzioni.

L'istruzione a 32 bit viene letta ed è rappresentata dall'etichetta **Instr**.





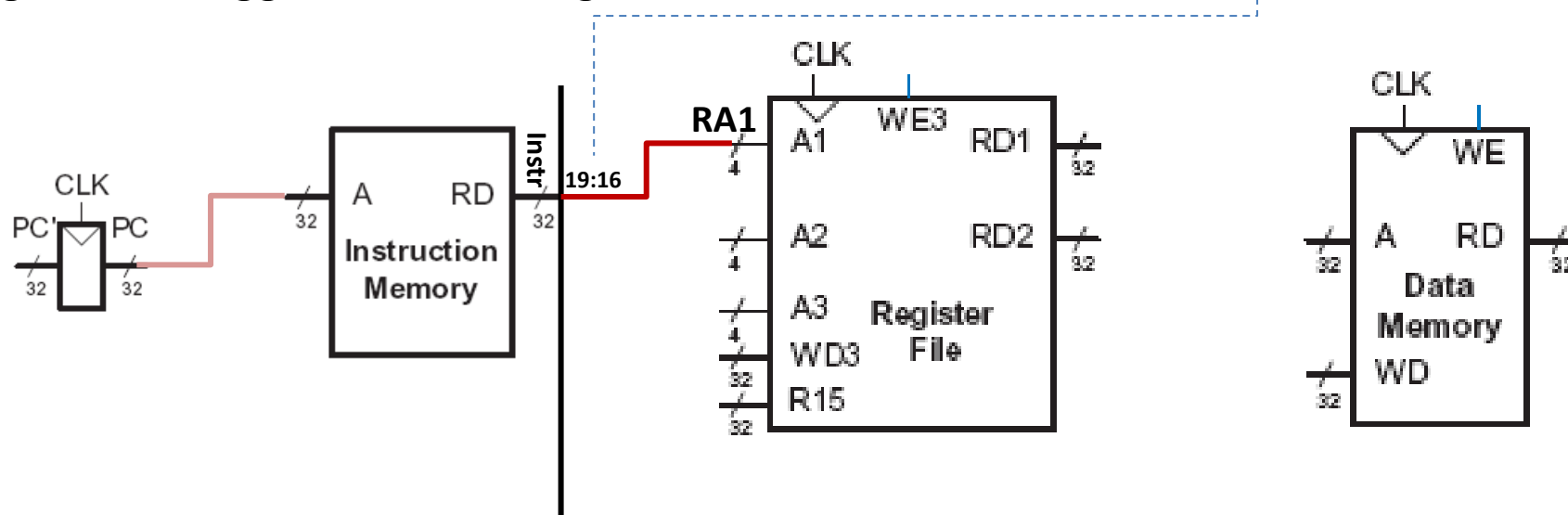
# Il datapath (LDR)

Il passo successivo è quello di leggere il registro sorgente contenente l'indirizzo di base. Questo registro è specificato nel campo **Rn** dell'istruzione, **Instr<sub>19:16</sub>**

Questi bit vengono collegati all'ingresso indirizzo di una delle porte del file register (**A1**).



Il register file legge il valore di registro in **RD1**.



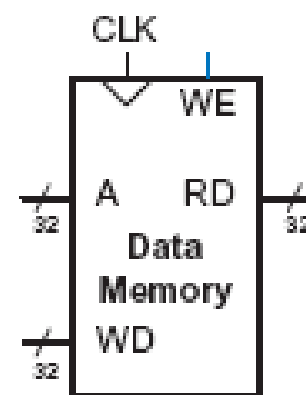
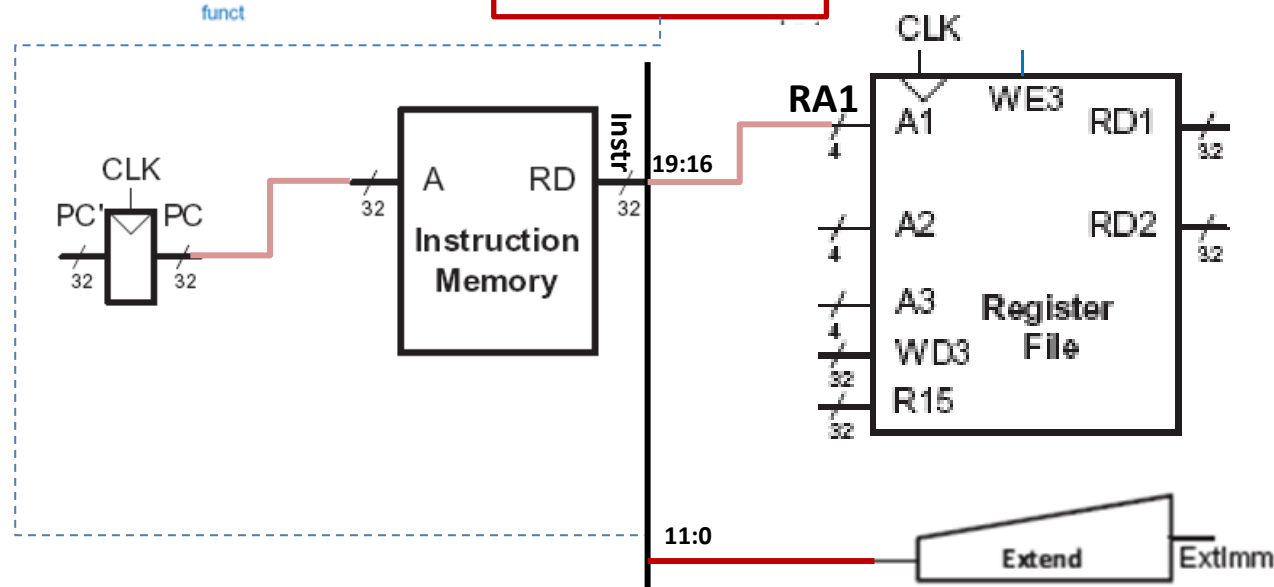
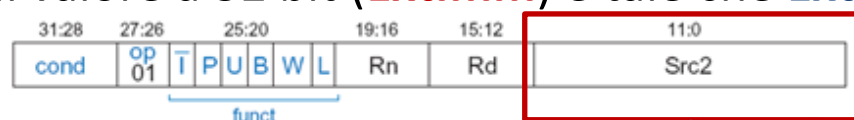


# Il datapath (LDR)

L'istruzione **LDR** richiede anche un **offset**, il quale è memorizzato nell'istruzione stessa e corrisponde ai bit **Instr<sub>11:0</sub>**.

L'offset è un valore senza segno, quindi deve essere esteso a 32 bit.

Il valore a 32 bit (**ExtImm**) è tale che **ExtImm<sub>31:12</sub> = 0** e **ExtImm<sub>11:0</sub> = Instr<sub>11:0</sub>**.



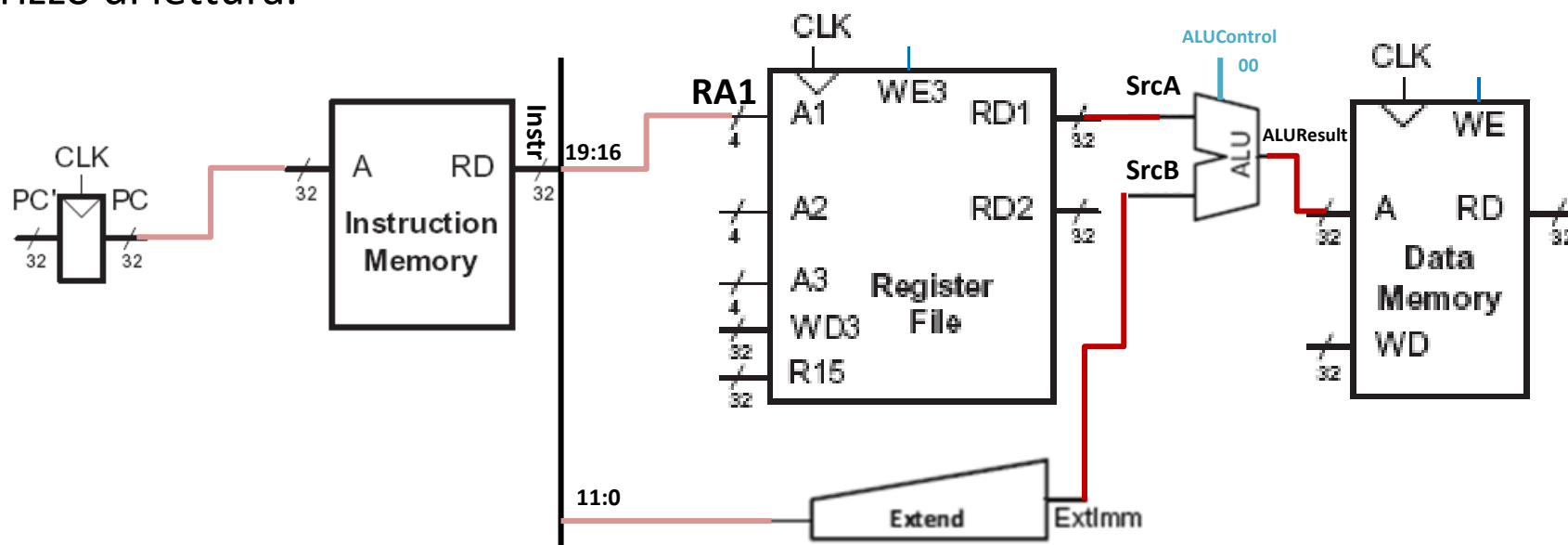


# Il datapath (LDR)

Il processore deve aggiungere l'**indirizzo di base** all'offset per trovare l'indirizzo di memoria a cui leggere. La somma è effettuata per mezzo di una **ALU**.

La **ALU** riceve due operandi (**srcA** e **srcB**). **srcA** proviene dal register file, mentre **srcB** è il valore già contenuto nell'**ALU**. Inoltre, il segnale a 2-bit **ALUControl** specifica l'operazione (una somma è indicata con 00).

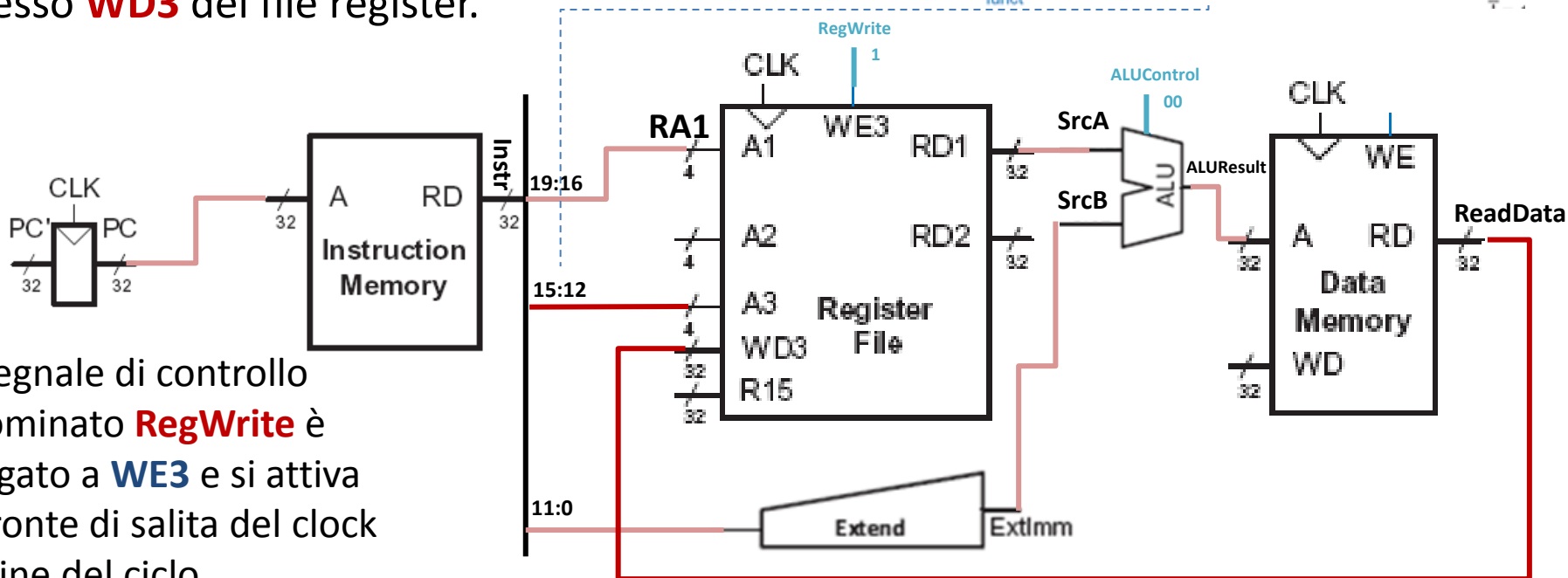
La **ALU** genera un valore a 32 bit **ALUResult**, che viene inviato alla memoria dati come indirizzo di lettura.



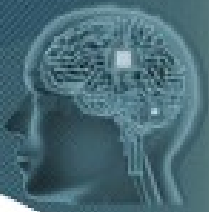


**Port3** è la porta di scrittura e il registro di destinazione per l'istruzione **LDR** è specificato nel campo **Rd**, **Instr<sub>15:12</sub>**, che è collegato all'indirizzo di ingresso **A3** di **Port3**.

31:28	27:26	25:20	19:16	15:12	11:0
cond	op 01	T P U B W L	Rn	Rd	Src2
		func			



Un segnale di controllo denominato **RegWrite** è collegato a **WE3** e si attiva sul fronte di salita del clock alla fine del ciclo.

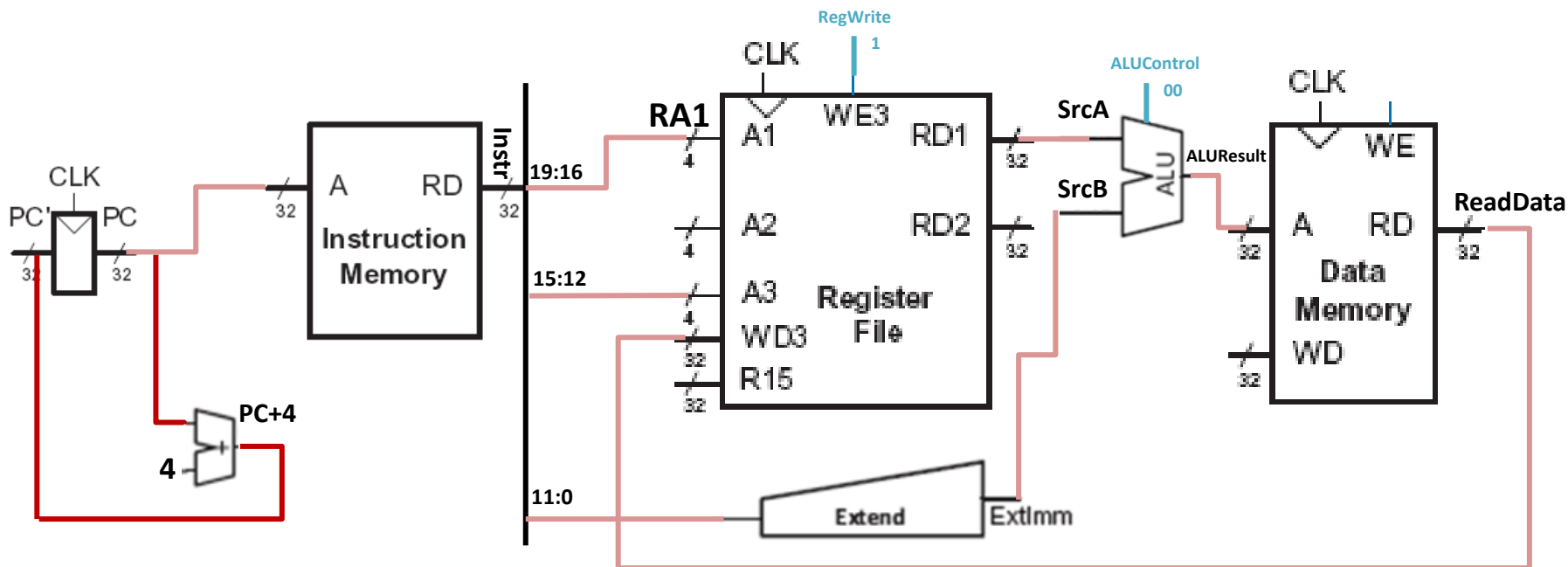


# Il datapath (LDR)

Contemporaneamente il processore deve calcolare l'indirizzo della successiva istruzione **PC'**.

Le istruzioni sono a 32 bit (4 byte), quindi l'istruzione successiva è a **PC + 4**.

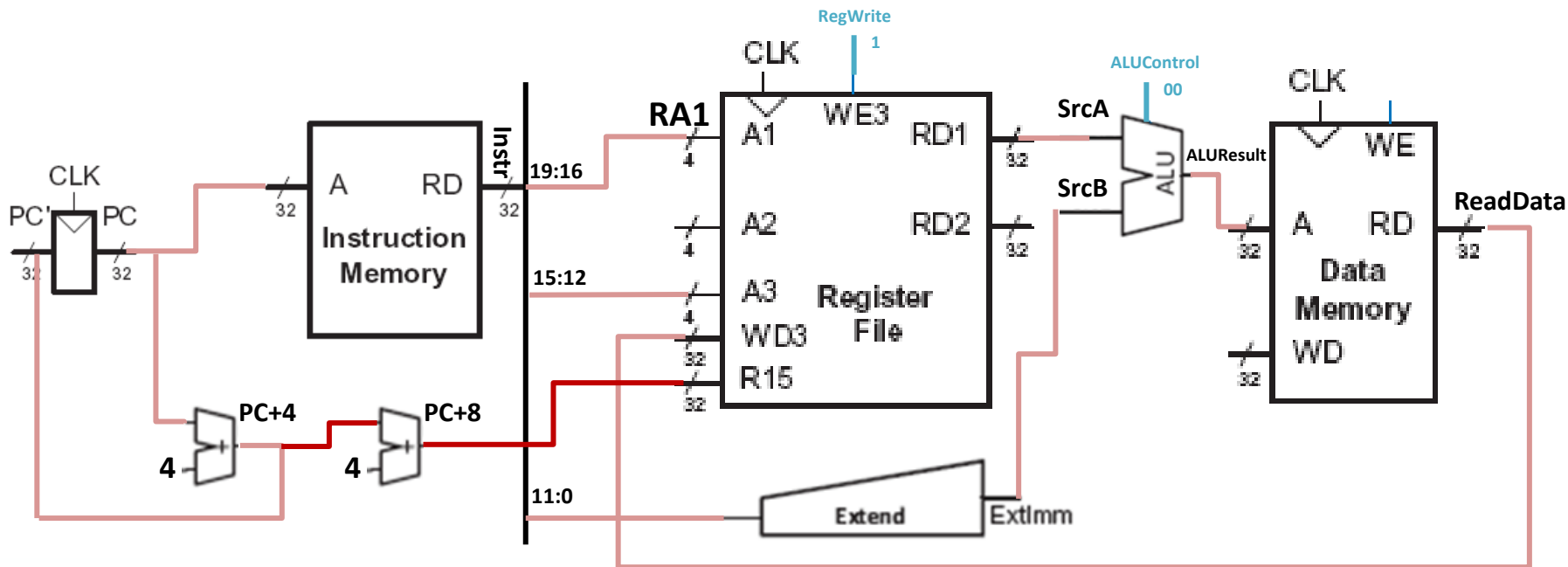
Si utilizza un **sommatore** per incrementare il **PC** di 4. Il nuovo indirizzo viene scritto nel contatore di programma sul successivo fronte di salita del clock.





# Il datapath (LDR)

Nelle architetture ARM il registro **R15** contiene il valore **PC+8**, per cui è necessario un ulteriore **sommatore** (+4), la cui uscita sia collegata all'ingresso di **R15**.



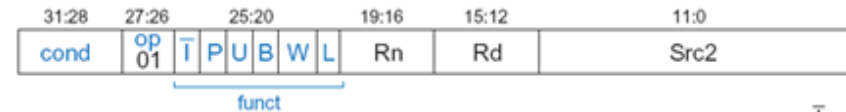




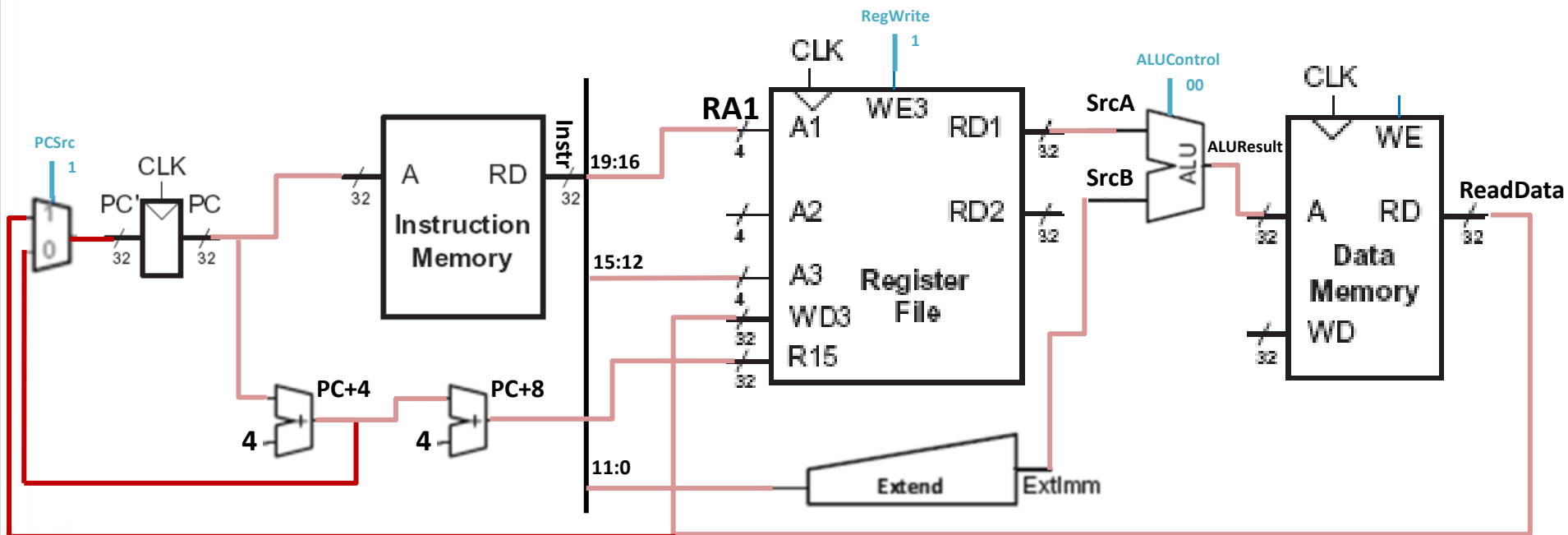
# Il datapath (LDR)

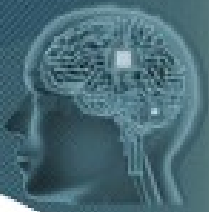
Infine, trattandosi di uno stored program paradigm, l'istruzione successiva potrebbe essere letta anche dalla memoria e quindi corrispondere al contenuto di **ReadData**. Un **multiplexer**, permette di selezionare fra:

- 0 – **PCPlus4**
- 1 – **ReadData**.



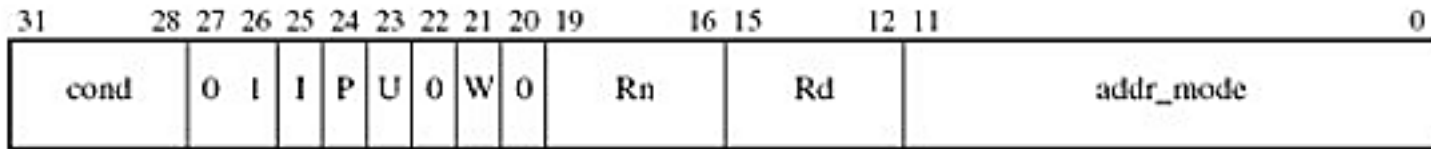
Il segnale di controllo associato al multiplexer è **PCSrc**.





# Istruzioni di memoria (STR)

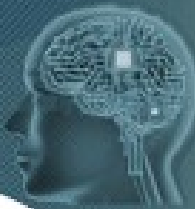
Istruzione: **STR**



STR (Store Register) stores a word from a register to memory.

L'istruzione **LDR** scrive una parola di 32 bit contenuta in un registro nella memoria centrale. Il modo in cui questa operazione viene effettuata dipende dalla politica di indirizzamento specificata.

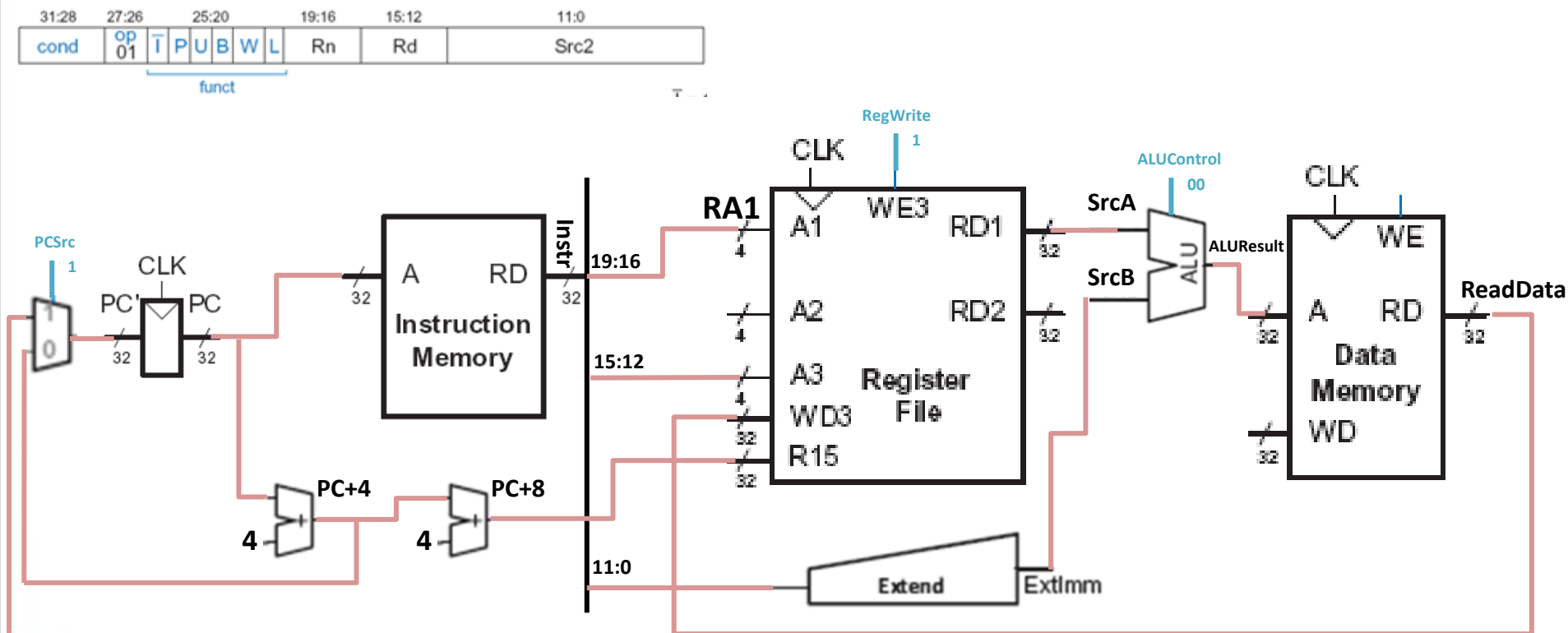
```
// Assume R1 = 0x0000_2000
STR R0, [R1] // [R1] <- R0
STR R0, [R1, #16] // [R1+16] <- R0
```



# Il datapath (STR)

Estendiamo il **datapath** in modo da poter gestire anche l'istruzioni **STR**.

Come LDR, **STR** legge un indirizzo di base dalla porta **A1** del register file e completa con zeri la costante. L'**ALU** aggiunge l'indirizzo di base alla costante per trovare l'indirizzo di memoria. Tutte queste funzioni sono già supportate nel datapath.



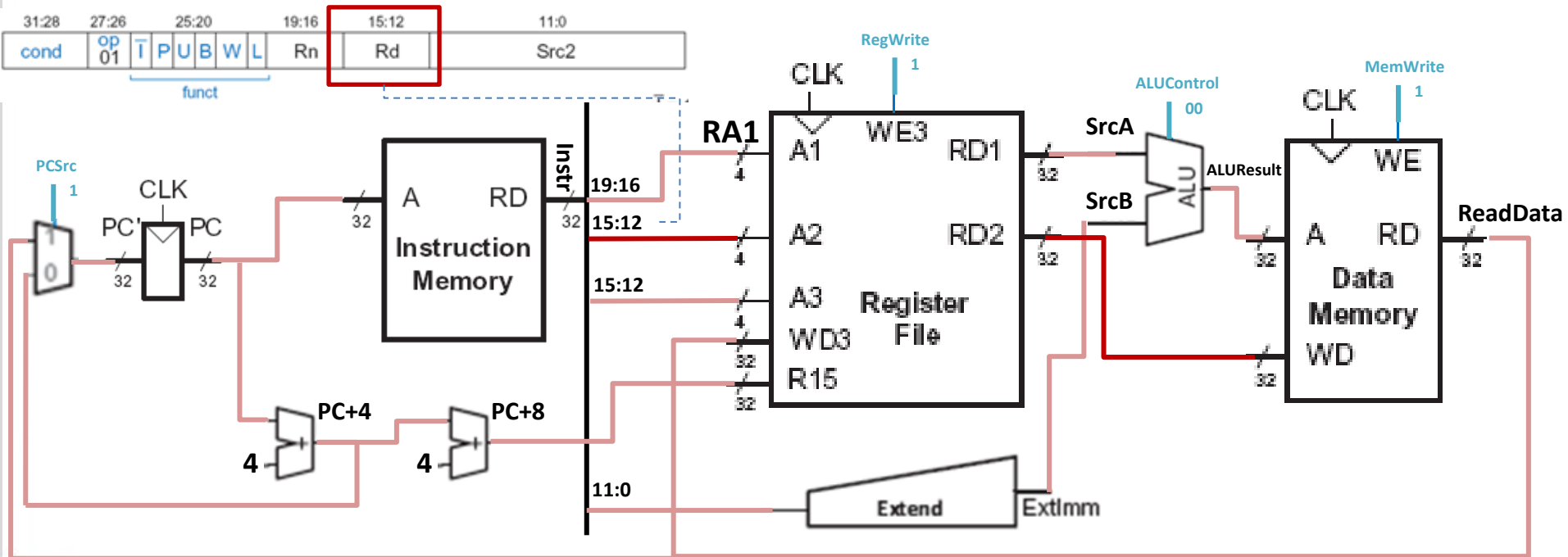


# Il datapath (STR)

Il registro è specificato nel campo **Rd**, **Instr<sub>15:12</sub>**, che è collegato alla porta **A2** del register file.

Il valore del registro viene letto sulla porta **RD2**, che è collegata alla porta dati di scrittura (**WD**) della memoria dati.

L'abilitazione del segnale di scrittura **WE** è controllato da **MemWrite**, il quale è 1 se i dati devono essere scritti in memoria.





# Il datapath (STR)

Il segnale **ALUControl** deve essere impostato a **00** per sommare l'indirizzo di base e l'offset.

Il segnale **RegWrite** è impostato a **0**, perché nulla deve essere scritto nel register file.

Si noti che è ancora abilitata l'operazione di lettura da memoria, ma **ReadData** attivo non produce effetti, essendo **RegWrite** impostato a **0**.

