



CPU a singolo ciclo

Prof. Alberto Borghese
Dipartimento di Informatica
alberto.borghese@unimi.it

Università degli Studi di Milano

Riferimento sul Patterson: capitolo 4.2 , 4.4, D1, D2.



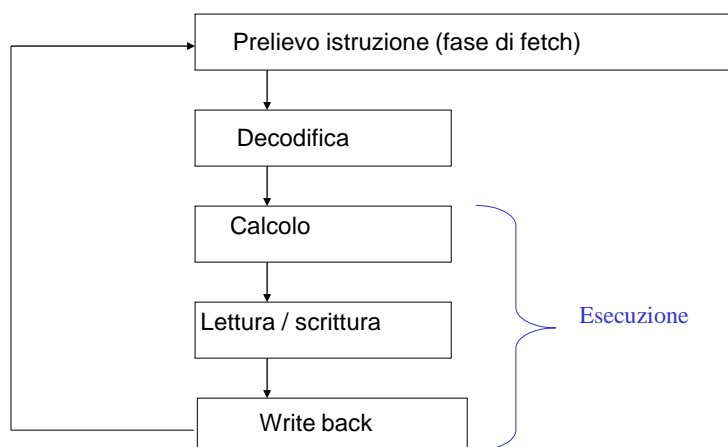
Sommario

CPU per le istruzioni di tipo R

CPU per le istruzioni di tipo I



Ciclo di esecuzione di un'istruzione MIPS



Codifica delle istruzioni



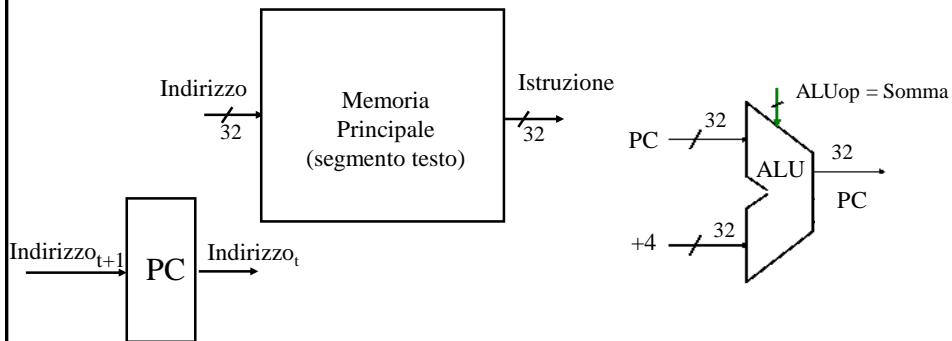
- Tutte le istruzioni MIPS hanno la **stessa dimensione (32 bit)** – Architettura RISC.
- I 32 bit hanno un significato diverso a seconda del formato (o tipo) di istruzione
 - il tipo di istruzione è riconosciuto in base al valore di alcuni bit (**6 bit**) più significativi (**codice operativo - OPCODE**)
- Le istruzioni MIPS sono di **3 tipi** (formati):
 - **Tipo R (register)** – Lavorano su **3 registri**.
 - Istruzioni aritmetico-logiche.
 - **Tipo I (immediate)** – Lavorano su **2 registri**. L'istruzione è suddivisa in un **gruppo di 16 bit contenenti informazioni + 16 bit riservati ad una costante**.
 - Istruzioni di accesso alla memoria o operazioni contenenti delle costanti.
 - **Tipo J (jump)** – Lavora **senza registri: codice operativo + indirizzo di salto**.
 - Istruzioni di salto incondizionato.

	6-bit	5-bit	5-bit	5-bit	5-bit	6-bit
R	op	rs	rt	rd	shamt	funct
I	op	rs	rt	indirizzo		
J	op	indirizzo				



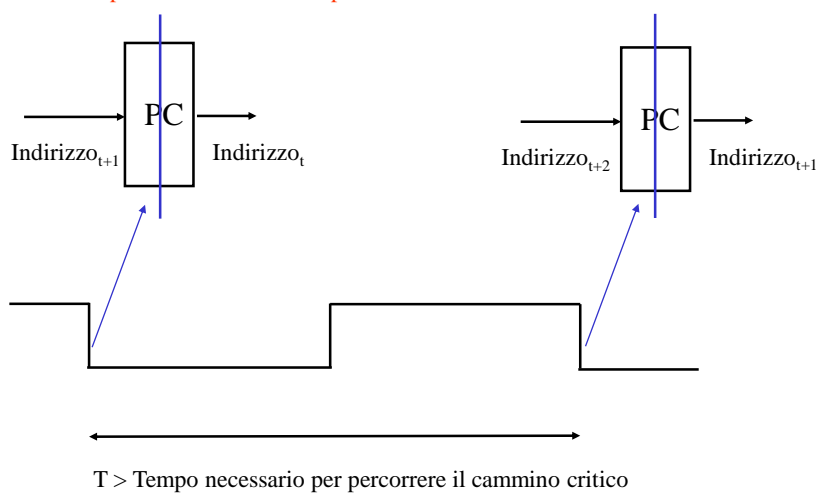
Fase di fetch

- 1) Memorizzare l'indirizzo dell'istruzione nel PC.
- 2) Leggere l'istruzione dalla memoria.
- 3) Aggiornare l'indirizzo in modo che in PC sia contenuto l'indirizzo dell'istruzione successiva.



Temporizzazione

1 istruzione per ciclo di clock. Temporizzazione del PC.

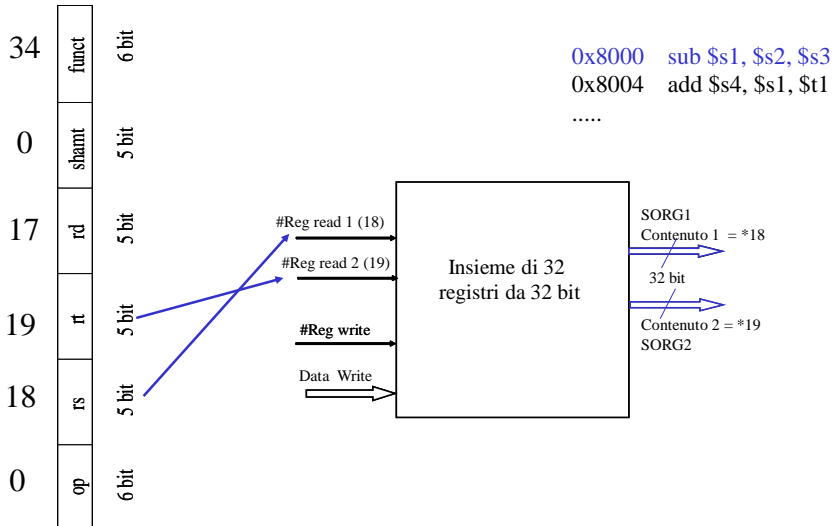




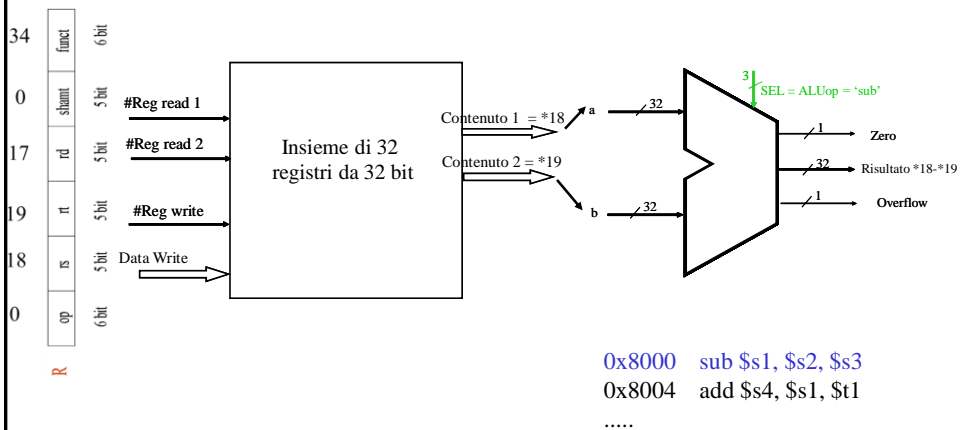
Lettura dei registri (istruzioni di tipo R)

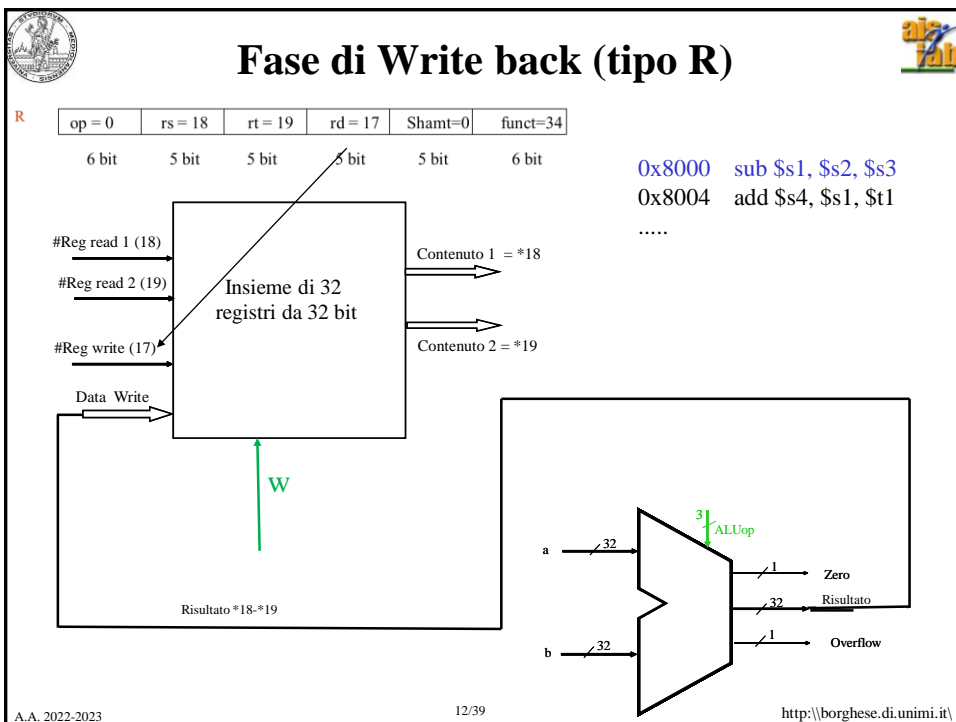
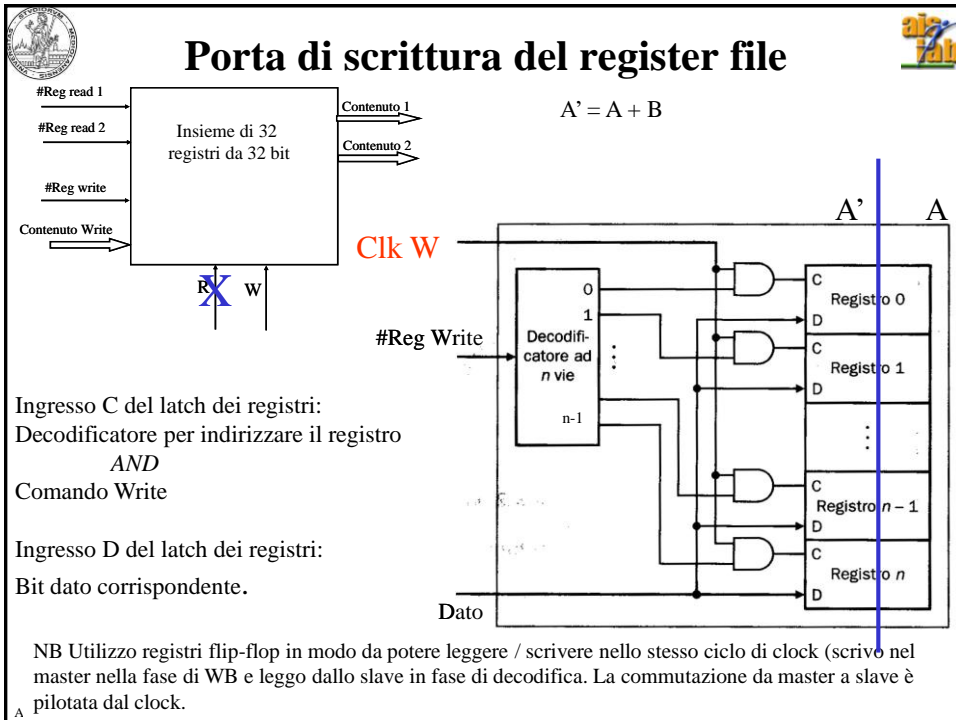


2) Leggo il contenuto dei registri. 2 bus che collegano i campi rs e rt con gli input corrispondenti del register file



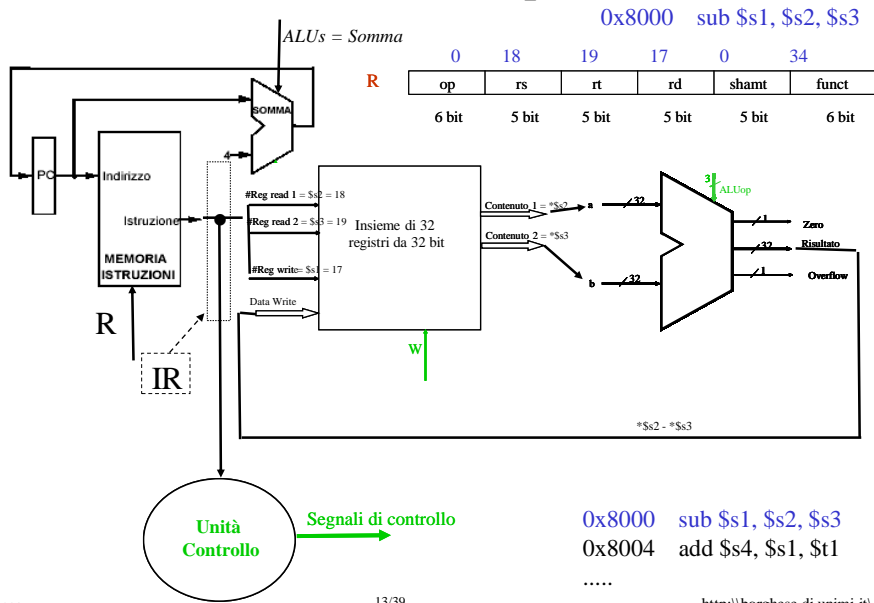
Fase di Calcolo (tipo R)







CPU per l'esecuzione completa di un'istruzione di tipo R



Sommario

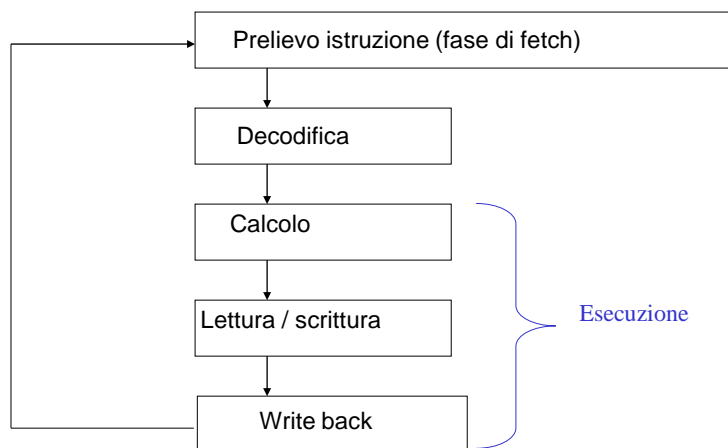


CPU per le istruzioni di tipo R

CPU per le istruzioni di tipo I



Ciclo di esecuzione di un'istruzione MIPS



Codifica delle istruzioni

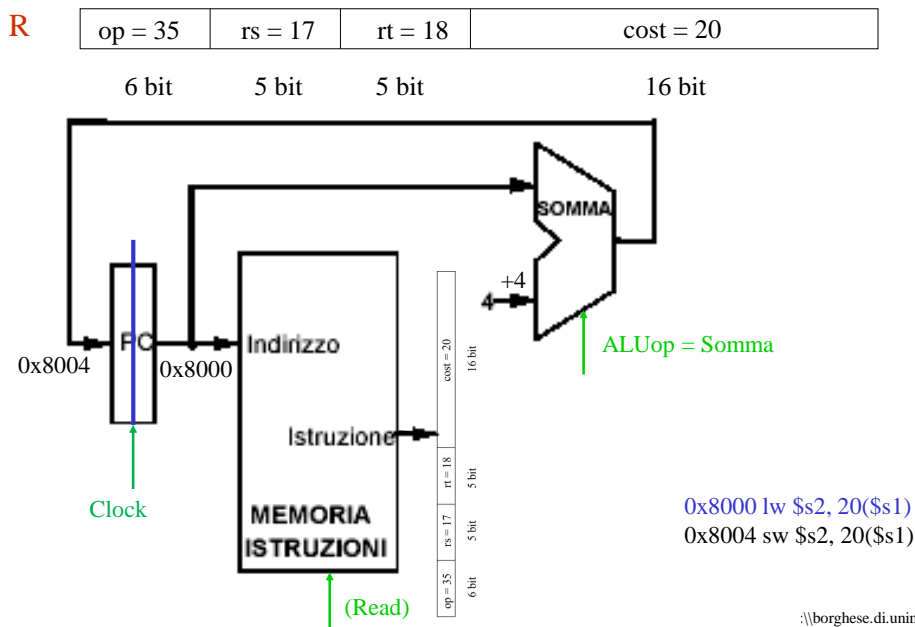


- Tutte le istruzioni MIPS hanno la **stessa dimensione (32 bit)** – Architettura RISC.
- I 32 bit hanno un significato diverso a seconda del formato (o tipo) di istruzione
 - il tipo di istruzione è riconosciuto in base al valore di alcuni bit (**6 bit**) più significativi (**codice operativo - OPCODE**)
- Le istruzioni MIPS sono di **3 tipi** (formati):
 - **Tipo R (register)** – Lavorano su **3 registri**.
 - Istruzioni aritmetico-logiche.
 - **Tipo I (immediate)** – Lavorano su **2 registri**. L'istruzione è suddivisa in un **gruppo di 16 bit contenenti informazioni + 16 bit riservati ad una costante**.
 - Istruzioni di accesso alla memoria o operazioni contenenti delle costanti.
 - **Tipo J (jump)** – Lavora **senza registri: codice operativo + indirizzo di salto**.
 - Istruzioni di salto incondizionato.

	6-bit	5-bit	5-bit	5-bit	5-bit	6-bit
R	op	rs	rt	rd	shamt	funct
I	op	rs	rt	Indirizzo / costante		
J	op	indirizzo				



Circuito della fase di fetch

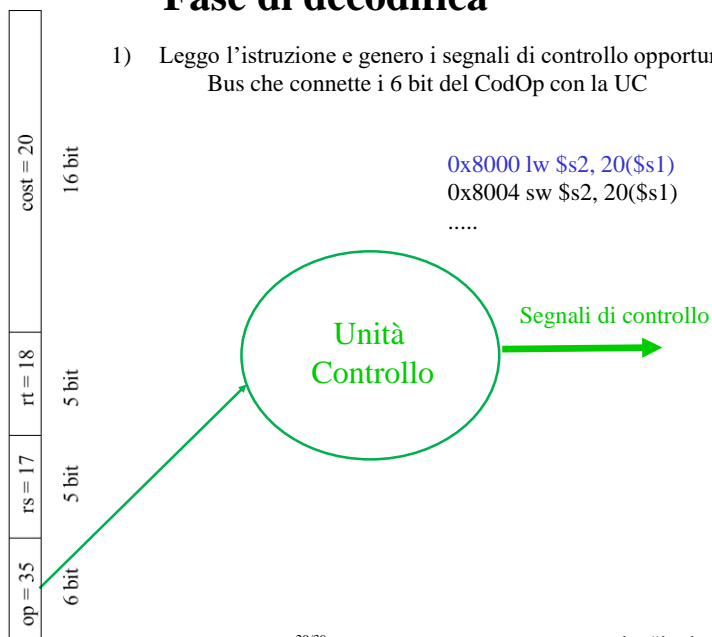


Fase di decodifica



- 1) Leggo l'istruzione e genero i segnali di controllo opportuni.
Bus che connette i 6 bit del CodOp con la UC

0x8000 lw \$s2, 20(\$s1)
0x8004 sw \$s2, 20(\$s1)
.....

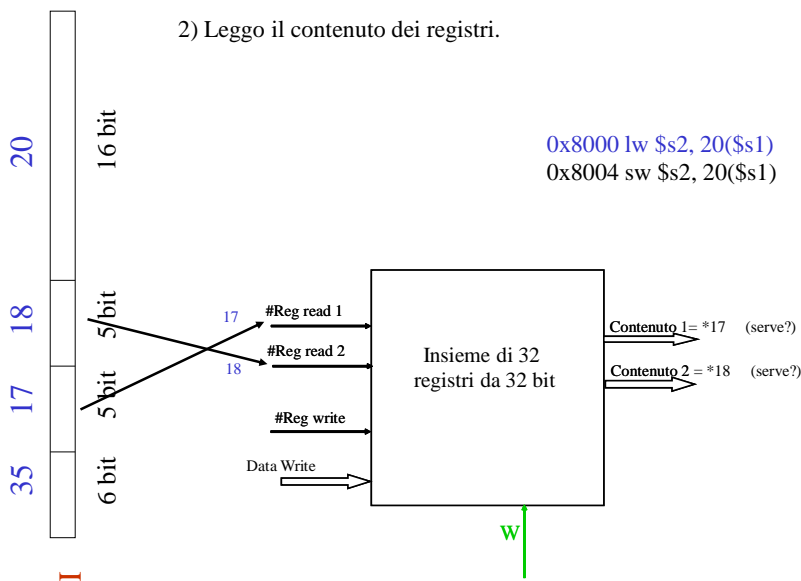




Letture dei registri (istruzioni di tipo I)



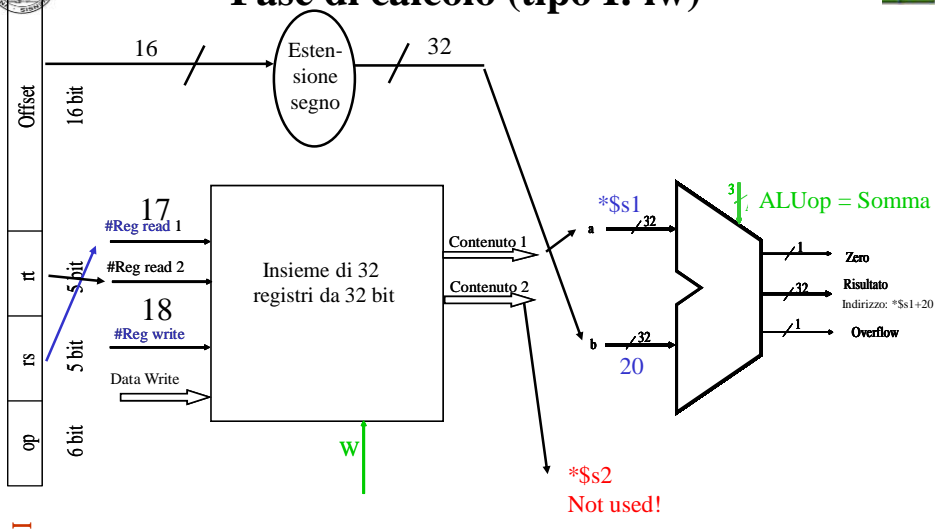
2) Leggo il contenuto dei registri.



0x8000 lw \$s2, 20(\$s1)
0x8004 sw \$s2, 20(\$s1)

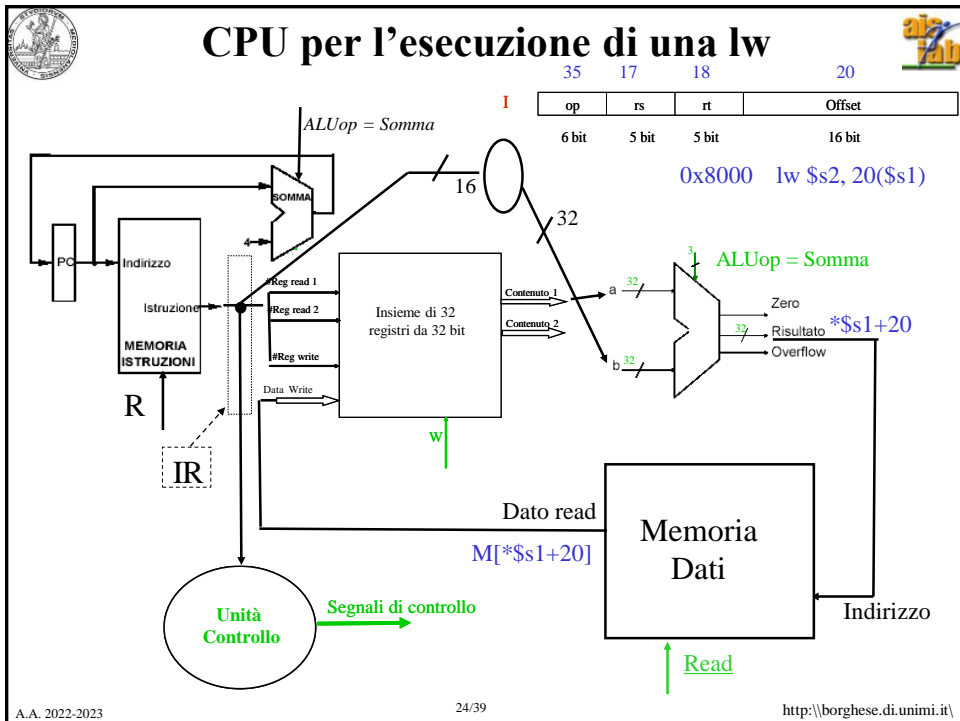
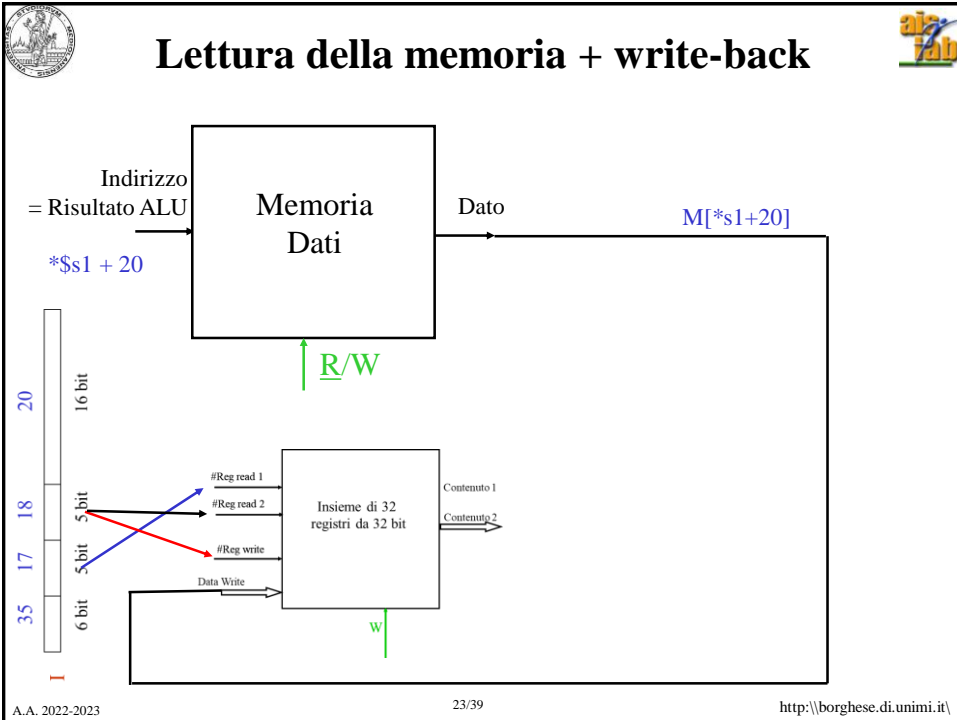


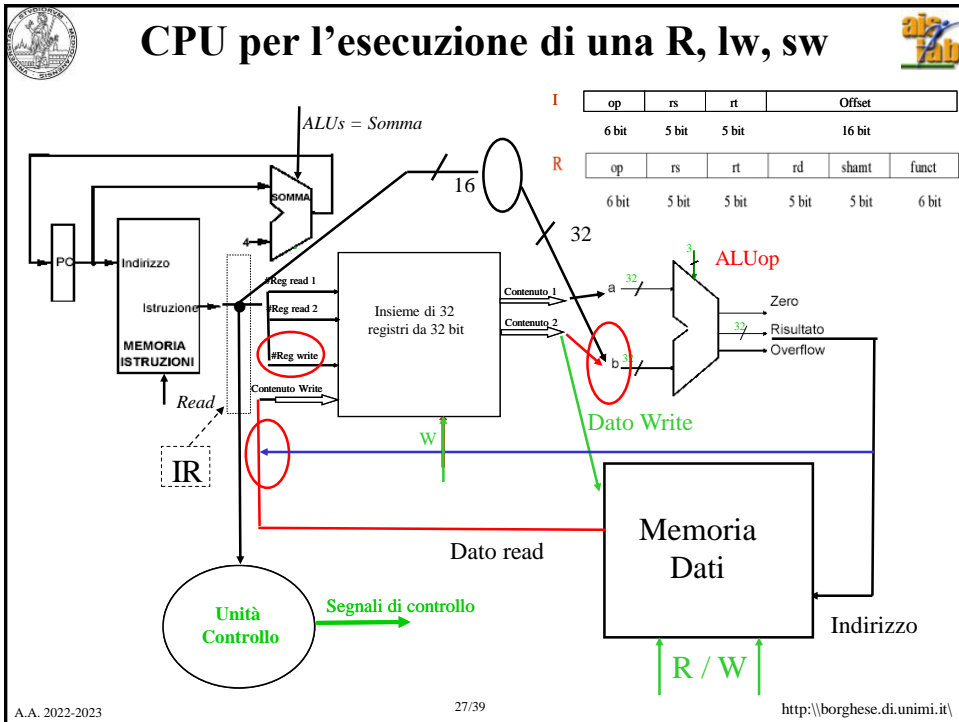
Fase di calcolo (tipo I: lw)



Il Risultato è un indirizzo della memoria: $*s1 + 20$

0x8000 lw \$s2, 20(\$s1)





Commenti

Questa CPU può eseguire istruzioni di tipo R, lw, sw

Fase di fetch e decodifica è uguale per tutte e 3 i tipi di istruzioni

Le fasi di Exe, Memoria e WriteBack sono diverse:

- Le istruzioni di tipo R hanno solo Exe e WriteBack.
- Le istruzioni di sw hanno Exe e Mem
- Le istruzioni di lw hanno Exe, Mem e Write Back.

- Nelle istruzioni di tipo R i due operandi vengono prelevati dal RF e il risultato scritto nel RF proviene dall'uscita della ALU
- Nelle istruzioni di sw e lw il secondo operando è la costante estesa di segno.
- Nelle istruzioni di sw rt contiene il dato da scrivere in memoria, nelle istruzioni di lw rt conterrà il dato letto dalla memoria.
- Nelle istruzioni di lw viene scritto nel RF il dato letto dalla memoria dati.

L'UC guida le scelte dei cammini

L'UC imposta i segnali di controllo per le unità funzionali.

I		rs	rt	Offset
op	6 bit	5 bit	5 bit	16 bit

R		rs	rt	rd	shamt	funct
op	6 bit	5 bit	5 bit	5 bit	5 bit	6 bit

A.A. 2022-2023 28/39 <http://borghese.di.unimi.it/>



Istruzioni di salto condizionato



- Salti condizionati relativi:
 - **beq** *r1, r2, L1* (*branch on equal*)
 - **bne** *r1, r2, L1* (*branch on not equal*)

```
beq $s1, $s0, esci # if (s1 == s0) esci
bne $s1, $s0, esci # if (s0 ≠ s0) esci
```
- Salti condizionati relativi:
 - Il flusso sequenziale di controllo cambia solo se la condizione è vera (*beq*)
 - Il calcolo del valore dell'etichetta **L1** (**indirizzo di destinazione del salto**) avviene a partire dal **Program Counter** (PC).
 - Indirizzamento del tipo Base (PC) + Spiazzamento.
- Indirizzo destinazione del salto:
 - $PC_{dest} = (PC + 4) + offset * 4$

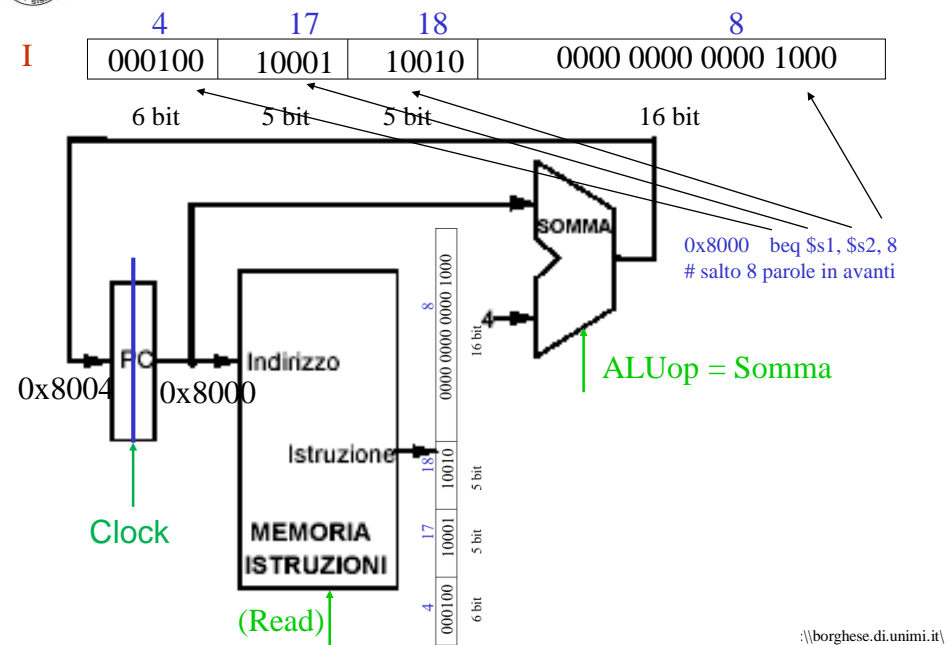
(cond vera)
Ind. Salto

(cond falsa)
+4
(procedi in sequenza)

Nome campo	op	rs	rt	indirizzo
Dimensione	6-bit	5-bit	5-bit	16-bit
beq <i>\$s1, \$s2, L1</i>	000100	10001	10010	0000 0000 0001 1001



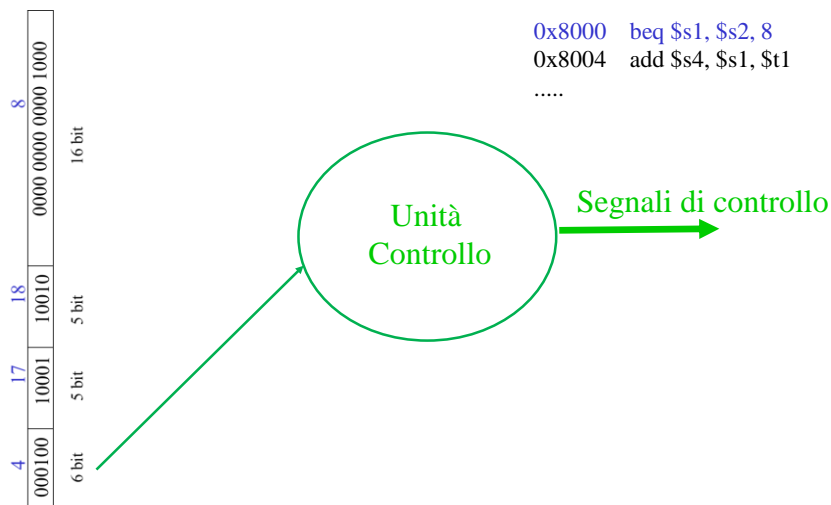
Circuito della fase di fetch





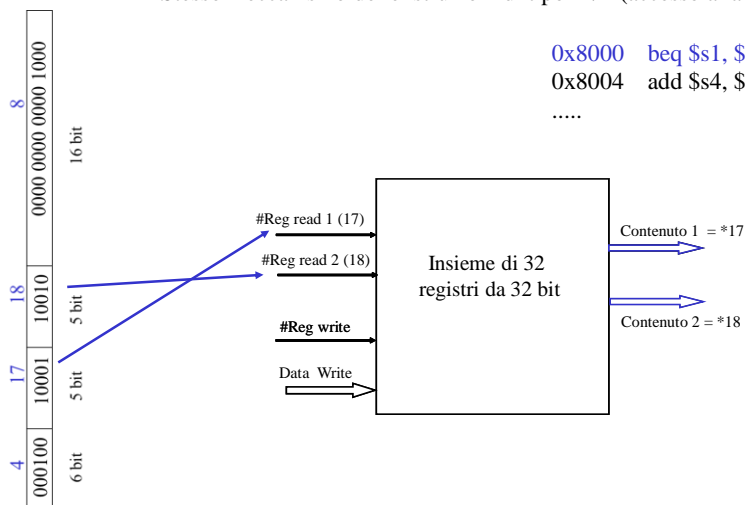
Fase di decodifica

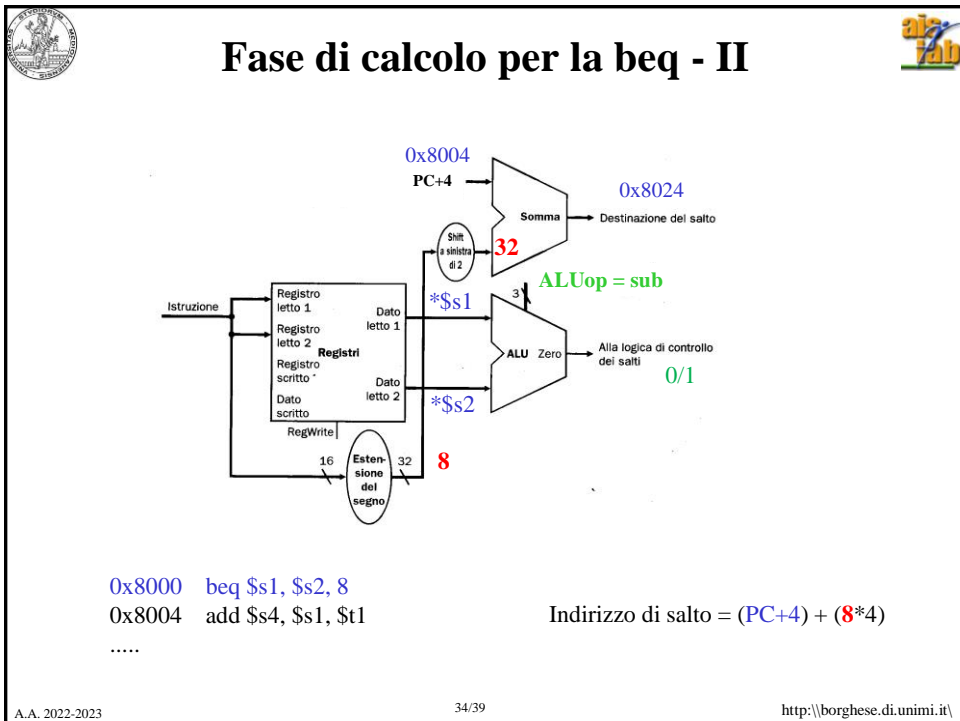
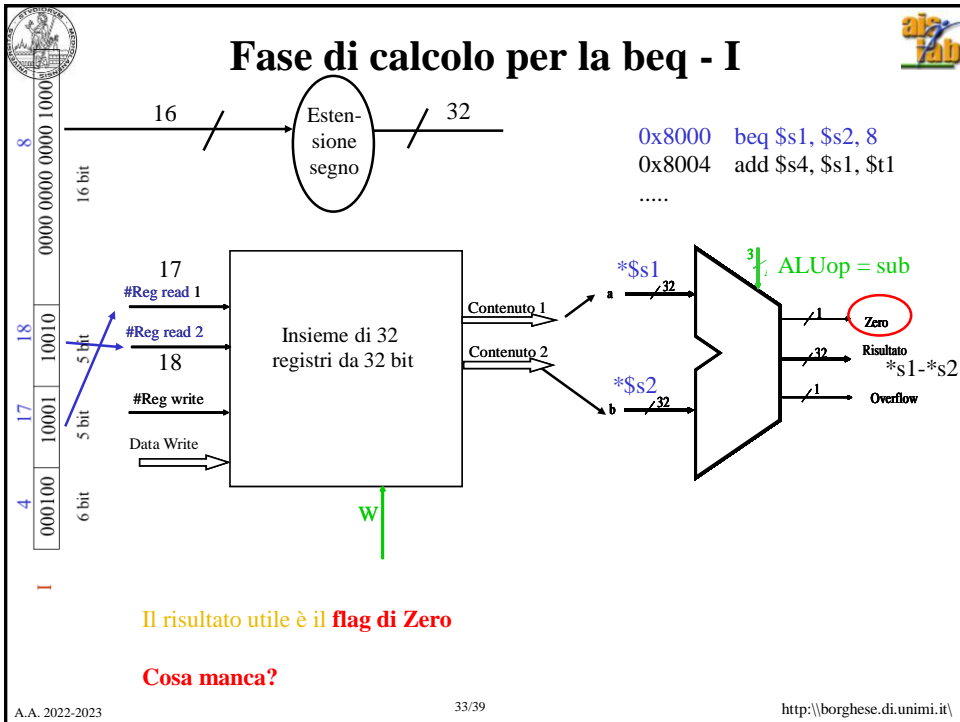
- 1) Leggo l'istruzione e genero i segnali di controllo opportuni.
Bus che connette i 6 bit del CodOp con la UC

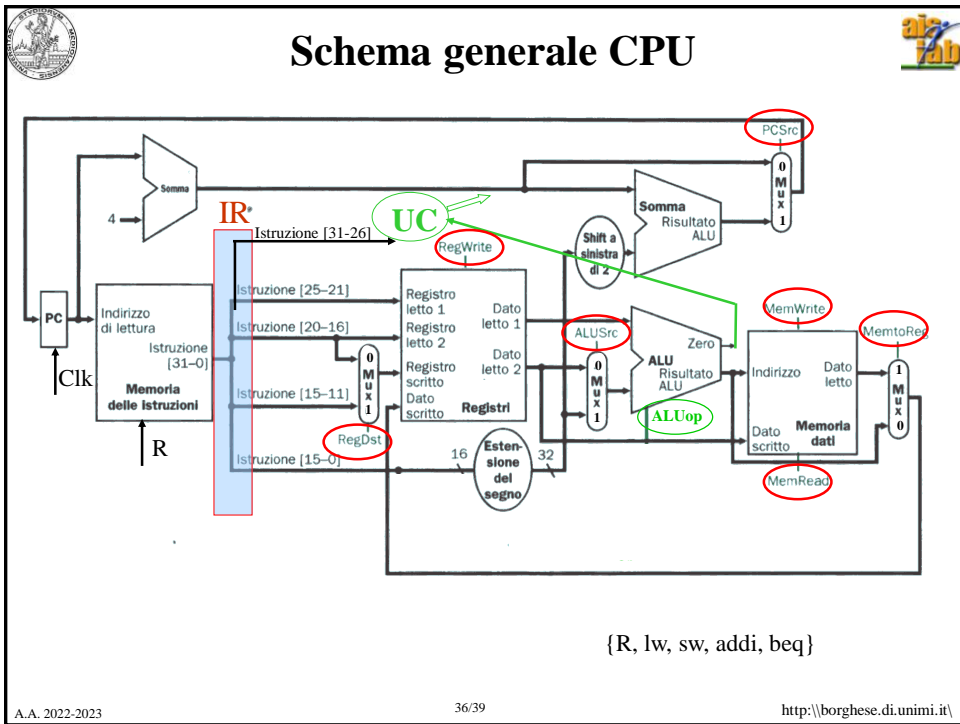
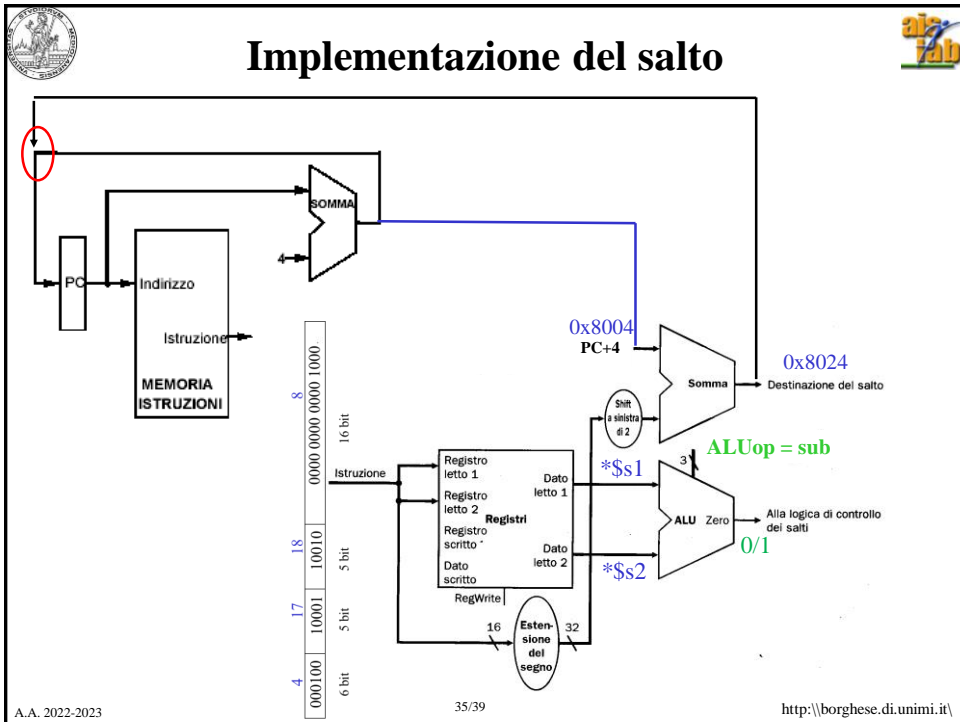


Lettura dei registri

- 2) Leggo il contenuto dei registri. 2 bus che collegato i campi rs e rt con gli input corrispondenti del register file.
Stesso meccanismo delle istruzioni di tipo R / I (accesso alla memoria)









Osservazioni



Il ciclo di esecuzione di un'istruzione si compie in un **unico** ciclo di clock.



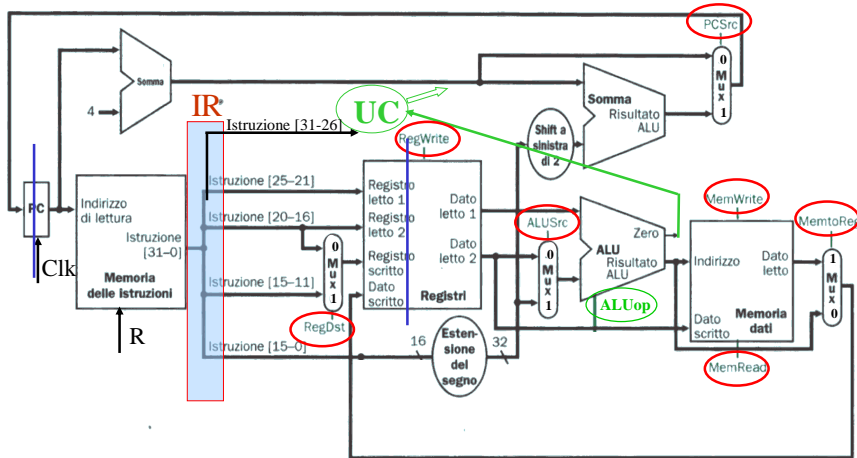
Ogni unità funzionale può essere utilizzata 1 sola volta.



Duplicazione Memoria: Memoria dati e memoria istruzioni (split memory).
Triplicazione ALU: 2 sommatori + 1 ALU general purpose.

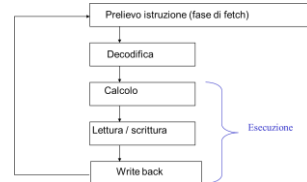


Schema generale CPU + UC



E' un'architettura retroazionata sincronizzata dal clock

Sincronizzazione del controllo (sul PC)
Sincronizzazione sui dati (sul RF)





Sommario



CPU per le istruzioni di tipo R

CPU per le istruzioni di tipo I