


La CPU a singolo ciclo

Prof. Alberto Borghese
Dipartimento di Scienze dell'Informazione
borghese@dsi.unimi.it

Università degli Studi di Milano

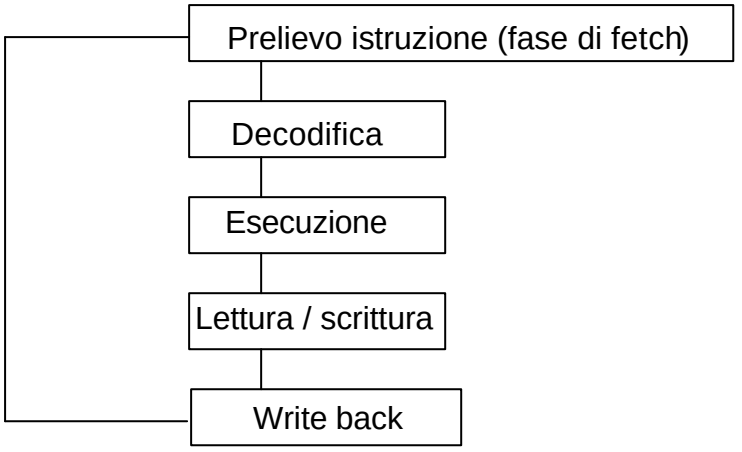

Sommario

Obbiettivo

Costruzione di una CPU completa che sia in grado di eseguire:

- Accesso alla memoria in lettura (lw) o scrittura (sw).
- Istruzioni logico-matematiche (e.g. add, sub, and....).
- Istruzioni di salto condizionato (branch) o incondizionato (jump).

Ciclo di esecuzione di un'istruzione MIPS



Architettura delle istruzioni R e I

R

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

- Ai vari campi sono stati assegnati dei nomi mnemonici:
  - **op:** (opcode) identifica il tipo di istruzione
  - **rs:** registro contenente il primo operando sorgente
  - **rt:** registro contenente il secondo operando sorgente
  - **rd:** registro destinazione contenente il risultato
  - **shamt:** shift amount (scorrimento)
  - **funct:** indica la variante specifica dell'operazione
  - **Indirizzo:** offset.

I

op	rs	rt	Indirizzo (=costante)
6 bit	5 bit	5 bit	16 bit

I componenti di un'architettura

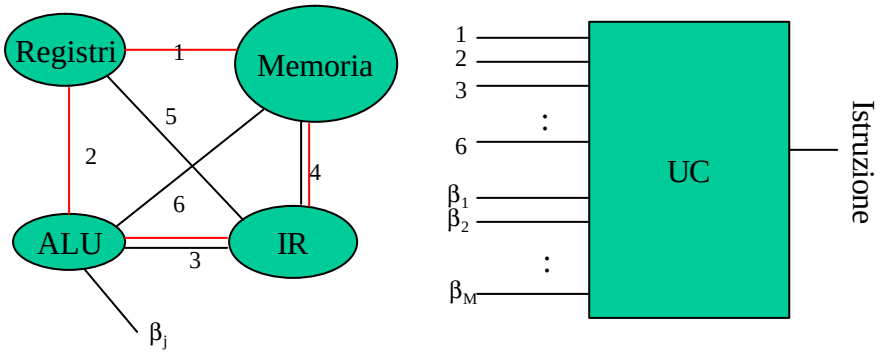
CPU

- Banco di registri *(Register File)* ad accesso rapido, in cui memorizzare i dati di utilizzo più frequente. Il tempo di accesso ai registri è circa 10 volte più veloce del tempo di accesso alla memoria principale.
- Registro *Program counter (PC)*. Contiene l'indirizzo dell'istruzione corrente da aggiornare durante l'evoluzione del programma, in modo da prelevare dalla memoria la corretta sequenza di istruzione;
- Registro *Instruction Register (IR)*. Contiene l'istruzione in corso di esecuzione. Questo registro verrà utilizzato più avanti nelle architetture multi-ciclo.
- Unità per l'esecuzione delle operazioni aritmetico-logiche (*Arithmetic Logic Unit - ALU*). I dati forniti all'*ALU* possono provenire da registri oppure direttamente dalla memoria, a seconda delle modalità di indirizzamento previste;
- Unità aggiuntive per elaborazioni particolari come unità aritmetiche per dati in virgola mobile *(Floating Point Unit – FPU)*, sommatori ausiliari, ecc.;
- ***Unità di controllo***. Controlla il flusso e determina le operazioni di ciascun blocco.

MEMORIA

L'unità di controllo

- Unità di controllo coordina i flussi di informazione (è il "cervello" della CPU):
- 1) abilitando le vie di comunicazione opportune a seconda dell'istruzione in corso di esecuzione.
- 2) selezionando l'operazione opportuna delle ALU.



Come funziona una UC

- Usa un registro, il Program Counter (PC) per ottenere l'indirizzo dell'istruzione.
- Preleva l'istruzione dalla memoria e la inserisce nell'IR.

- Capisce di che tipo di istruzione si tratta (decodifica).
  - usa l'istruzione stessa per decidere cosa fare esattamente.
- Legge il contenuto dei registri.

Da qui le istruzioni si differenziano.

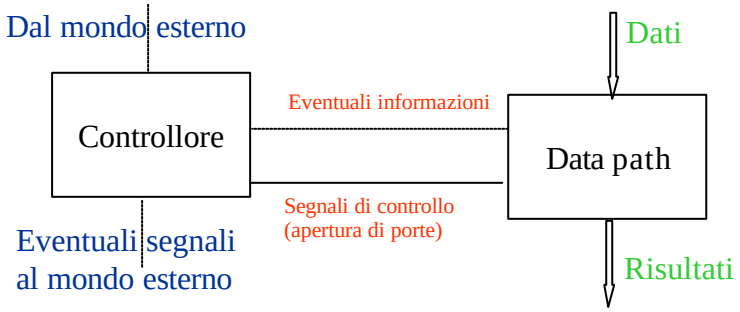
- Calcolo: utilizzo dell'ALU dopo aver letto i registri:
  - per calcolare l'indirizzo in memoria.
  - per eseguire un'operazione logico-aritmetica.
  - per effettuare il test.

- Accesso alla memoria.

- Scrittura del risultato nel register file.

Rapporto UC - Dati

La CPU è un'architettura del tipo: Controllore - Data-path



Fase comune nel ciclo di esecuzione:
•Fase di fetch
•Decodifica (generazione dei segnali di controllo)

Fase diversa: Esecuzione (Esecuzione, Accesso memoria, WriteBack)

Sommario

La CPU

Costruzione di una CPU per le istruzioni di tipo R

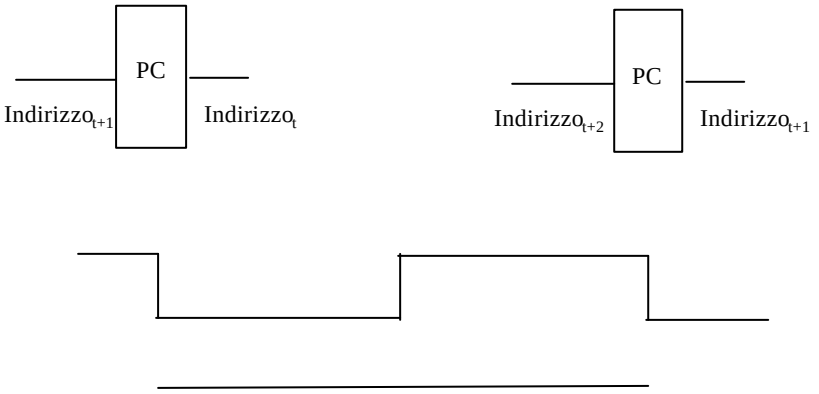
Costruzione di una CPU per le istruzioni di tipo I (memoria).

Costruzione di una CPU per le istruzioni di tipo I (salti).

CPU che gestisce istruzioni di tipo R, ed I (lw/sw e branch).

Temporizzazione

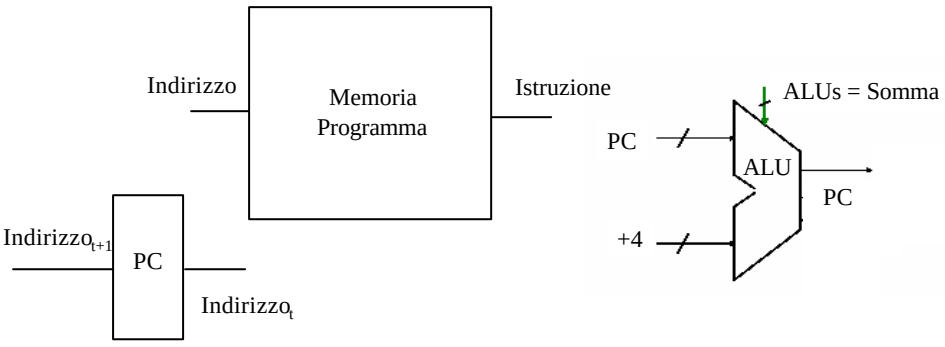
1 istruzione per ciclo di clock. Temporizzazione del PC.



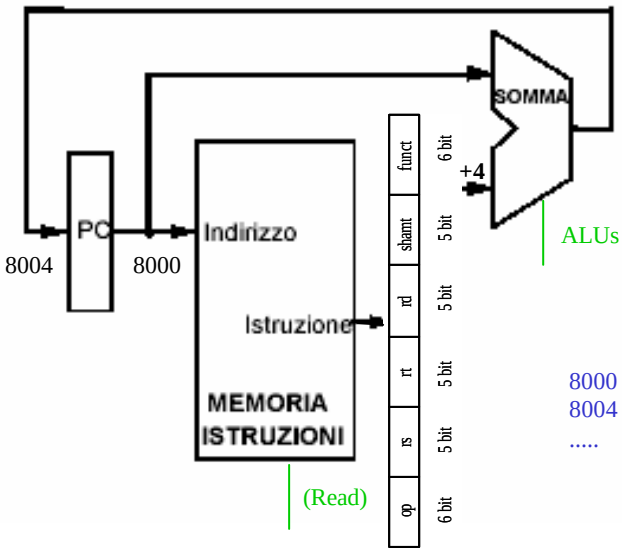
T > Tempo necessario per eseguire il cammino critico

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.



Circuito della fase di fetch



Istruzioni di tipo R

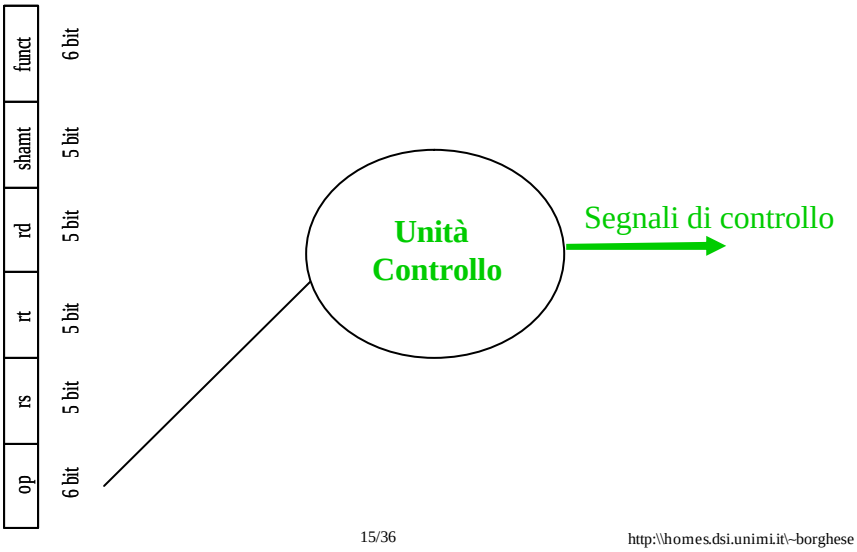
R

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

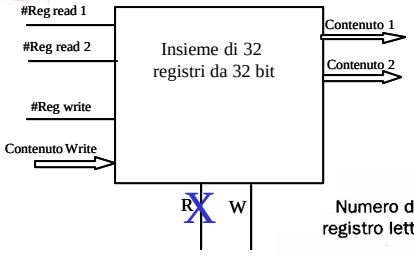
add $s1, $s2, $s3

Fase di decodifica

1) Leggo l'istruzione e genero i segnali di controllo opportuni.
2) Leggo il contenuto dei registri.

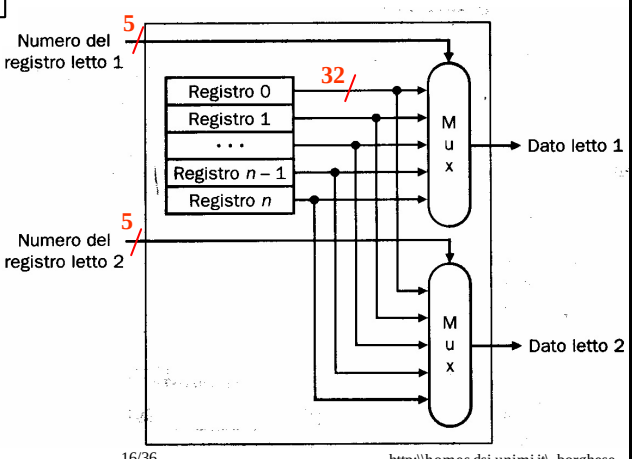


Register file



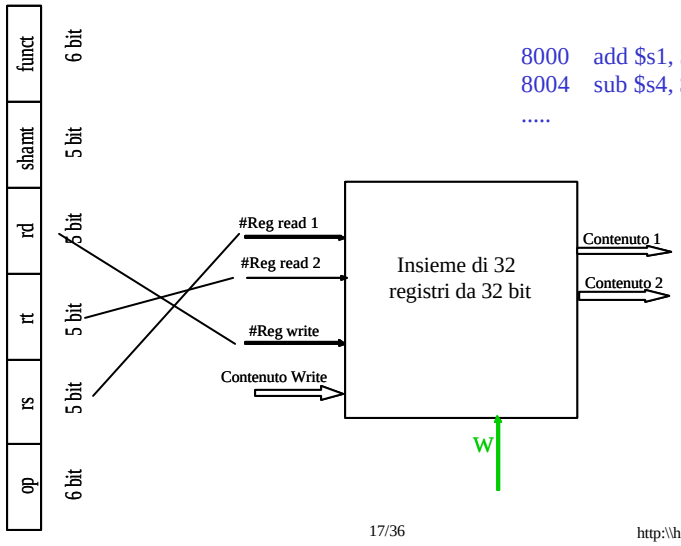
Banco di registri utilizzabile come memoria
Può essere scritto o letto.

Un mux per ogni porta di lettura.



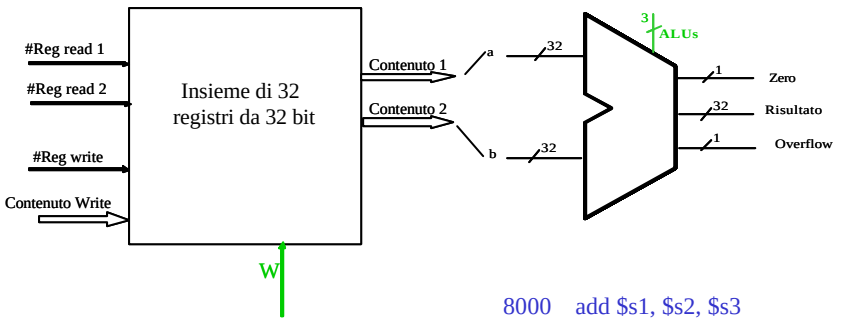
Lettura dei registri (istruzioni di tipo R)

1) Leggo l'istruzione e genero i segnali di controllo opportuni.
2) Leggo il contenuto dei registri.



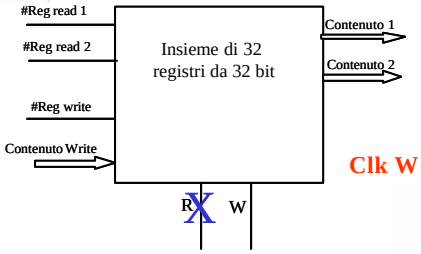

```
8000    add $s1, $s2, $s3
8004    sub $s4, $s1, $t1
.....
```

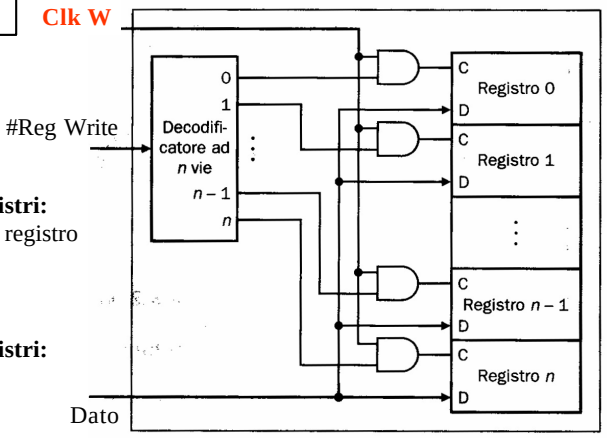

Fase di esecuzione (tipo R)




```
8000    add $s1, $s2, $s3
8004    sub $s4, $s1, $t1
.....
```

Porta di scrittura del register file



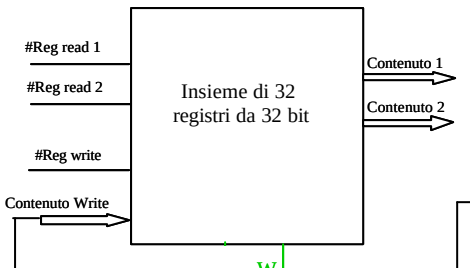


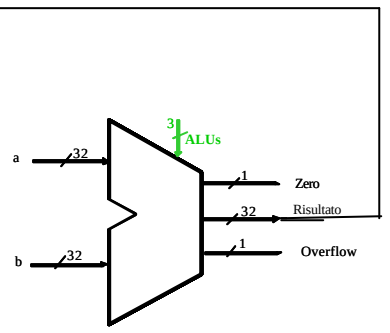
Ingresso C del Master dei registri:
Decodificatore per indirizzare il registro
AND
Comando W

Ingresso D del Master dei registri:
Bit dato corrispondente.

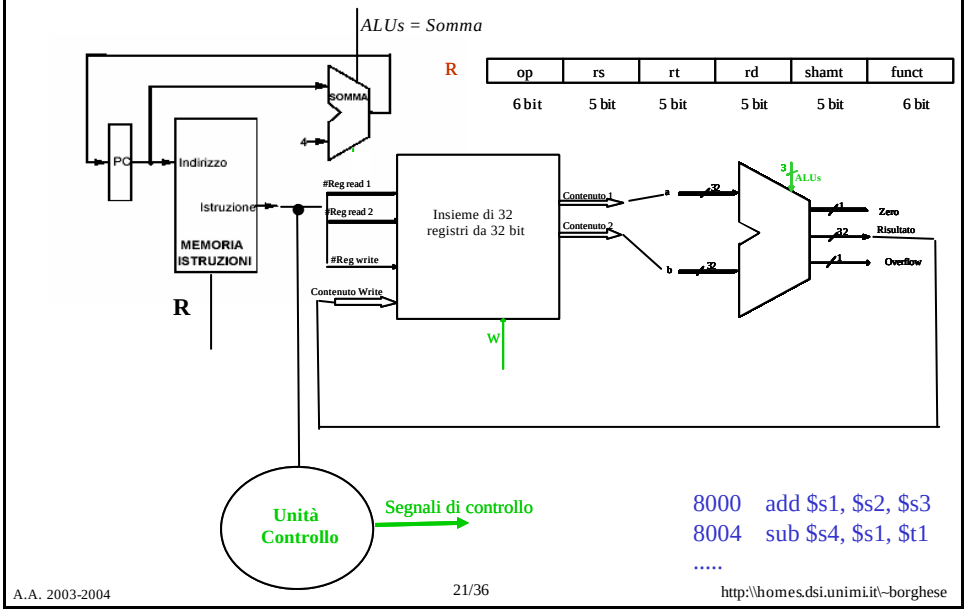
Fase di Write back (tipo R)

8000 add $s1, $s2, $s3
8004 sub $s4, $s1, $t1
.....
Risultato





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



8000 add $s1, $s2, $s3
8004 sub $s4, $s1, $t1
.....

Sommario

La CPU

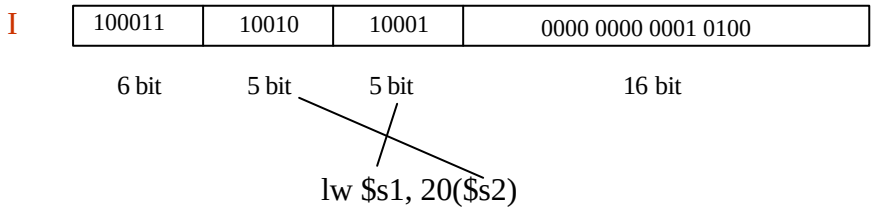
Costruzione di una CPU per le istruzioni di tipo R

Costruzione di una CPU per le istruzioni di tipo I (memoria).

Costruzione di una CPU per le istruzioni di tipo I (salti).

CPU che gestisce istruzioni di tipo R, ed I (lw/sw e branch).

Istruzioni di tipo I: lw/sw



L'indirizzo di memoria sarà:

Base <$s2> 0100 1000 0011 0001 1011 1011 1011 1011 +
Offset 0000 0000 0001 0100 +

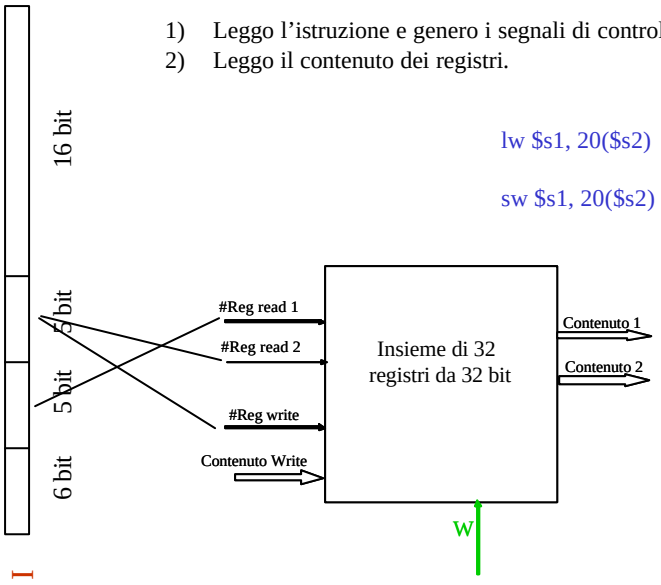
Indirizzo dato 0100 1000 0011 0001 1011 1011 1100 1111

Lettura dei registri (istruzioni di tipo I)

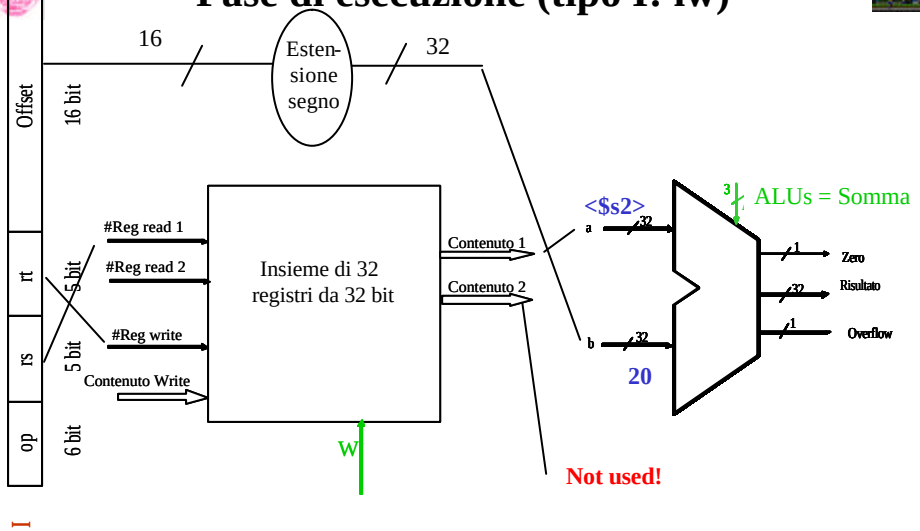
1) Leggo l'istruzione e genero i segnali di controllo opportuni.
2) Leggo il contenuto dei registri.

lw $s1, 20($s2)

sw $s1, 20($s2)



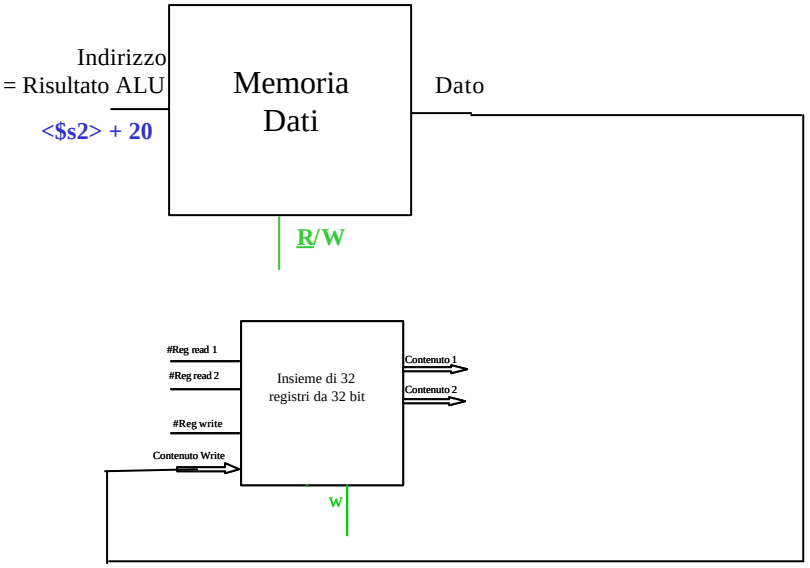
Fase di esecuzione (tipo I: lw)



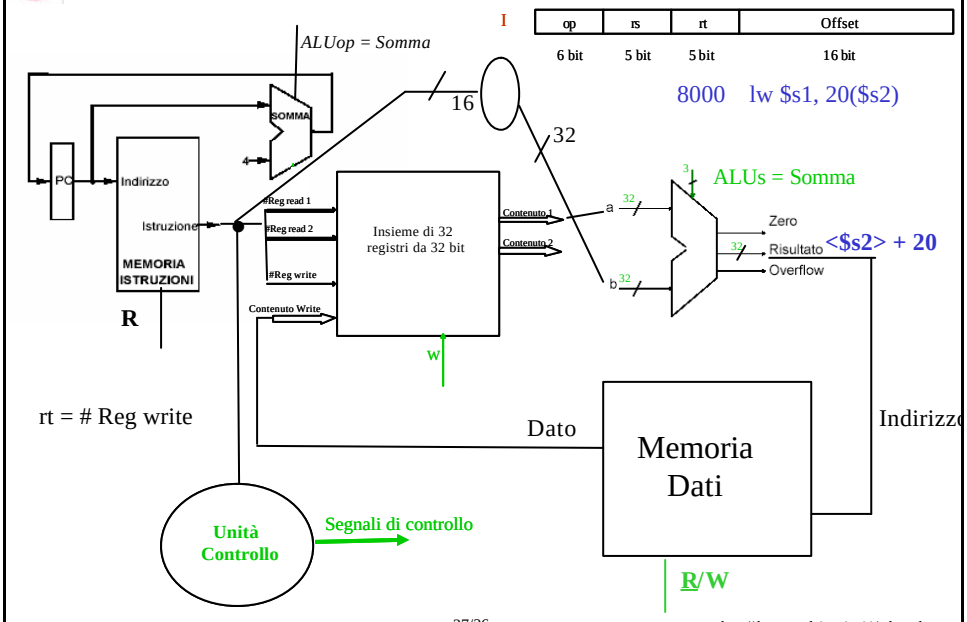
Il Risultato è un indirizzo della memoria

8000 lw $s1, 20($s2)

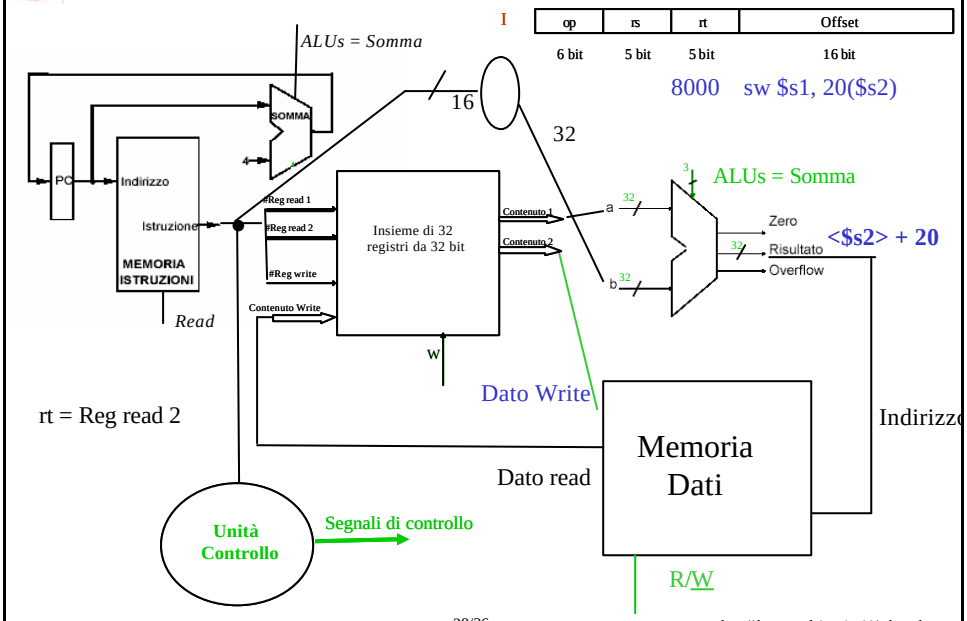
Lettura della memoria + write-back



CPU per l'esecuzione di una lw



CPU per l'esecuzione di una sw



Sommario

La CPU

Costruzione di una CPU per le istruzioni di tipo R

Costruzione di una CPU per le istruzioni di tipo I (memoria).

Costruzione di una CPU per le istruzioni di tipo I (salti).

CPU che gestisce istruzioni di tipo R, ed I (lw/sw e branch).

Istruzioni di tipo I: branch

000100	10001	10010	0000 0000 0000 0101

beq $s1, $s2, 20

L'indirizzo di salto sarà determinato in due passi:

A) Calcolo dello spiazzamento: Offset * 4.

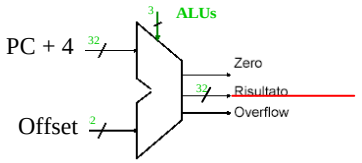
B) Deteminazione dell'indirizzo di salto come:

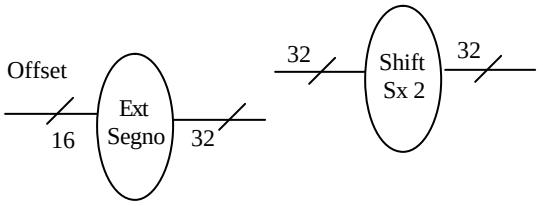
Base (PC)	0100 1000 0011 0001 1011 1011 1011 10 11 +
Offset	00 0000 0000 0001 01 (00) +
Indirizzo salto	0100 1000 0011 0001 1011 1011 1100 10 11

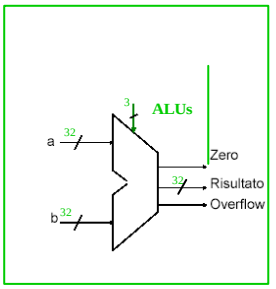
1) Determinare se l'uguaglianza è vera.
2) Calcolare l'indirizzo di salto.

Fase di esecuzione della beq

1) Estensione dell'offset su 32 bit.
2) Moltiplicazione per 4 dell'offset.
3) Somma del PC con l'estensione del segno.
4) Controllo se il contenuto dei registri è uguale.

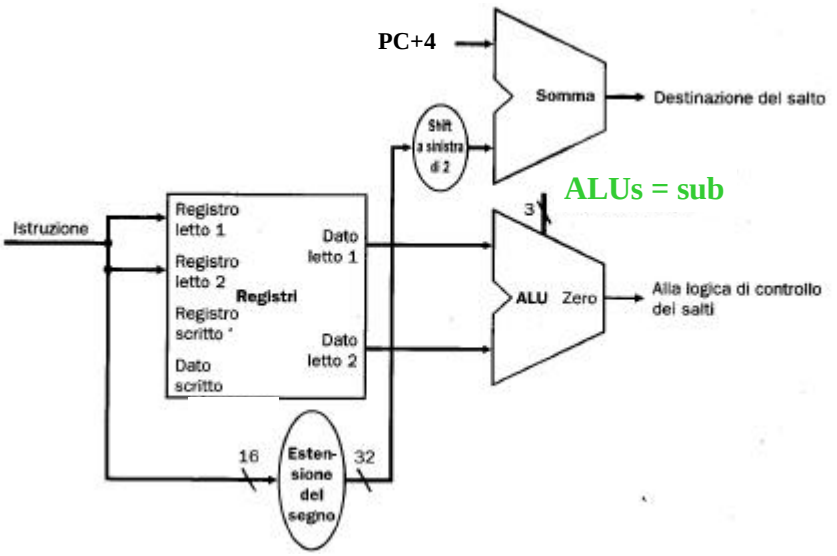






000100	10001	10010	0000 0000 0000 0101
Cod Op	rs	rt	Offset

Circuito di esecuzione della beq



Sommario

La CPU

Costruzione di una CPU per le istruzioni di tipo R

Costruzione di una CPU per le istruzioni di tipo I (memoria).

Costruzione di una CPU per le istruzioni di tipo I (salti).

CPU che gestisce istruzioni di tipo R, ed I (lw/sw e branch).

Osservazioni

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

⇩

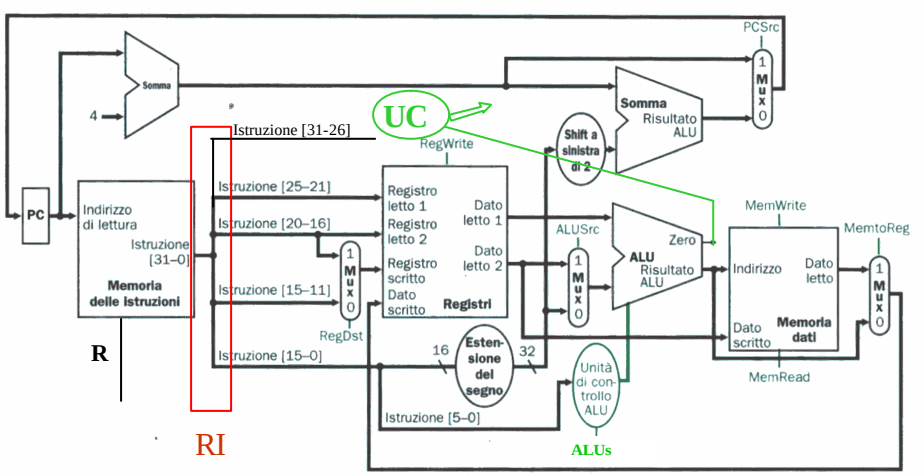
Ogni unità funzionale può essere utilizzata 1 sola volta.

⇩

Duplicazione Memoria: Memoria dati e memoria istruzioni.
Triplicazione ALU: 3 ALU: 2 sommatori + 1 general purpose.

Introduzione di multiplexer.

Schema generale (lw/sw + R + beq)



Sommario

La CPU

Costruzione di una CPU per le istruzioni di tipo R

Costruzione di una CPU per le istruzioni di tipo I (memoria).

Costruzione di una CPU per le istruzioni di tipo I (salti).

CPU che gestisce istruzioni di tipo R, ed I (lw/sw e branch).