



# Unità di controllo della pipeline

Prof. Alberto Borghese  
Dipartimento di Scienze dell'Informazione  
[borgnese@dsi.unimi.it](mailto:borgnese@dsi.unimi.it)

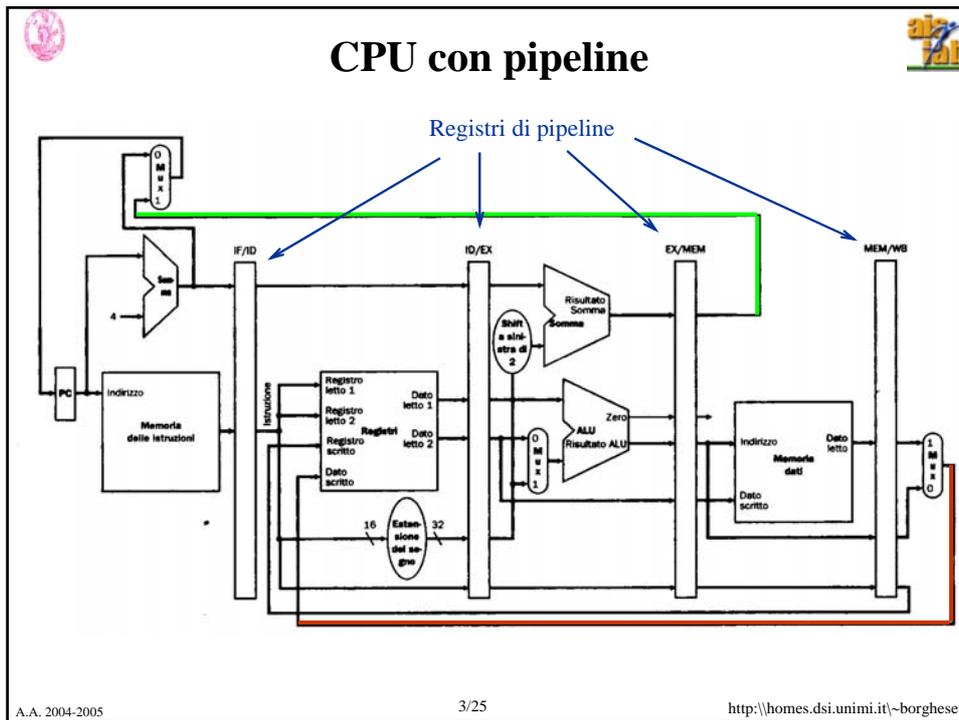
Università degli Studi di Milano



## Sommario

La CPU con pipeline

L'Unità di Controllo della pipeline



## Gli stadi di esecuzione

IF – Instruction Fetch  
 ID – Instruction Decode (e lettura register file)  
 EX – Esecuzione o calcolo dell'indirizzo di memoria.  
 MEM – Accesso alla memoria dati.  
 WB – Write Back (scrittura del risultato nel register file).

NB: I registri al termine di ogni fase prendono il nome dalle 2 fasi:  
 IF/ID      ID/EX      EX/MEM      MEM/WB

Perchè non c'è un registro WB/IF?

Il data-path procede da sx a dx.

A.A. 2004-2005 4/12 <http://homes.dsi.unimi.it/~borgnese>



## Il ruolo dei registri



Ciascuno stadio produce un risultato. La parte di risultato che serve agli stadi successivi deve essere memorizzata in un registro.

Il registro mantiene l'informazione anche se lo stadio in questione riutilizza l'unità funzionale.

Esempio: l'istruzione letta viene salvata nel registro IF/ID (cf. Instruction Register).



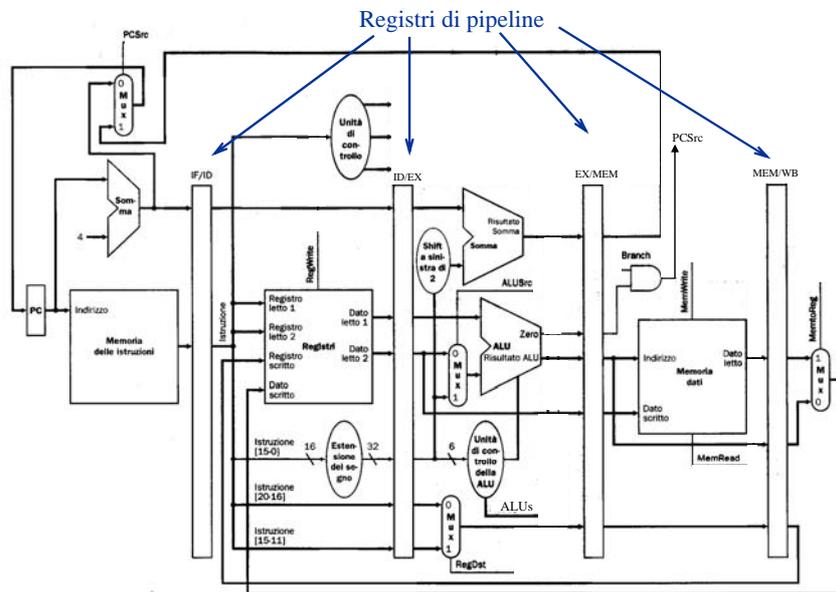
## Sommario



La CPU con pipeline

**L'Unità di Controllo della pipeline**

## CPU con pipeline



## La UC della CPU con pipeline

Definizione dei segnali di controllo per ogni stadio, per ogni istruzione.

Definizione dell'UC in grado di generare correttamente questi segnali.



## Segnali di controllo su 1 bit



Nome segnale	Effetto quando è negato	Effetto quando è affermato
RegDst	Il numero del registro destinazione proviene dal campo <i>rt</i> (R2, bit 20-16)	Il numero del registro destinazione proviene dal campo <i>rd</i> (bit 15-11)
RegWrite	Nessuno	Nel registro specificato all'ingresso registro scritto del Register File, viene scritto il valore presente all'ingresso Dato Scritto
ALUSrc	Il secondo operando della ALU proviene dalla seconda uscita in lettura del Register File	Il secondo operando della ALU è la versione estesa (con segno) del campo <i>offset</i>
Branch	Il valore del PC viene sostituito dall'uscita del sommatore che calcola PC+4 (condizionato all'uscita di ALU)	Il valore del PC viene sostituito dall'uscita del sommatore che calcola la destinazione del salto (condizionato all'uscita di ALU)
MemRead	Nessuno	Il contenuto della cella di memoria dati indirizzata dal MAR è posto nel MDR
MemWrite	Nessuno	Il contenuto in ingresso al MDR, viene memorizzato nella cella il cui indirizzo è caricato nel MAR
MemoReg	Il valore inviato all'ingresso Dato al Register File proviene dalla ALU	Il valore inviato all'ingresso Dato al Register File proviene dalla memoria

Scrittura PC e scrittura dei registri di pipeline ad ogni fronte di clock (ad ogni stadio).



## Osservazioni



Il contenuto di *rt* ed il numero di scrittura nel Register File (*rd*) vengono portati attraverso i vari stadi.

Nella fase di fetch e di decodifica non esistono segnali di controllo particolari.

I segnali di controllo particolari (legati alle diverse istruzioni) si possono così raggruppare:

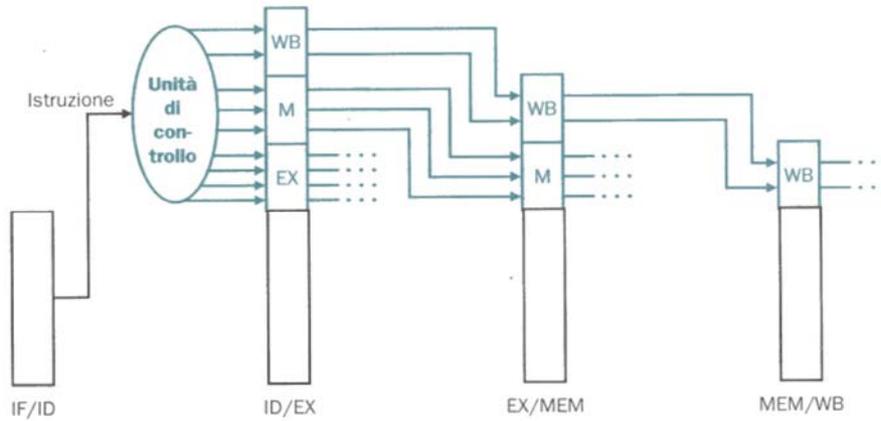
Istruzione	Exec				Memory			WB	
	Reg Dst	ALU Op1	ALU Op0	ALU Src	Branch	Mem Read	Mem Write	Reg Write	Mem2 Reg
Format-R	1	1	0	0	0	0	0	1	0
lw	0	0	0	1	0	1	0	1	1
sw	X	0	0	1	0	0	1	0	X
beq	X	0	1	0	1	0	0	0	X



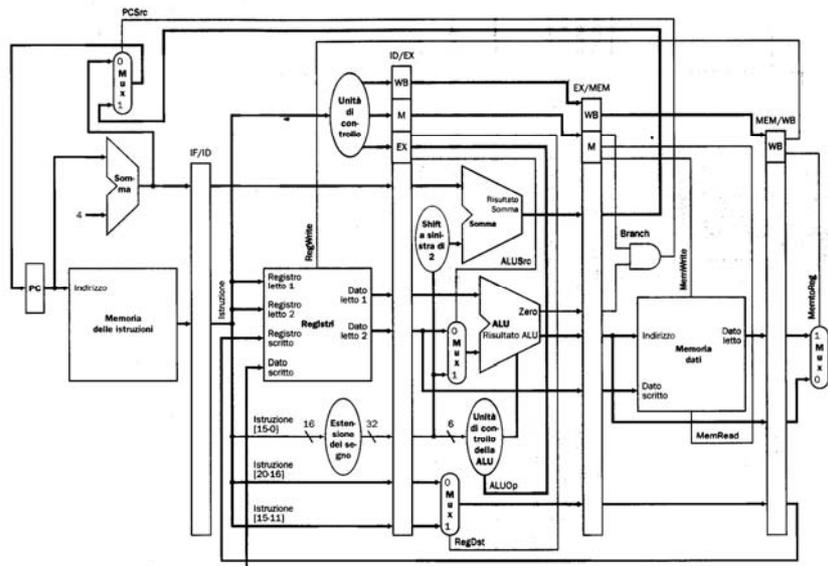
## Generazione dei segnali di controllo



I segnali di controllo vengono generati nello stadio di decodifica e propagati.



## UC per pipeline





## Sommario



La CPU con pipeline

L'Unità di Controllo della pipeline