



L'unità di controllo di CPU a singolo ciclo

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

Università degli Studi di Milano

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

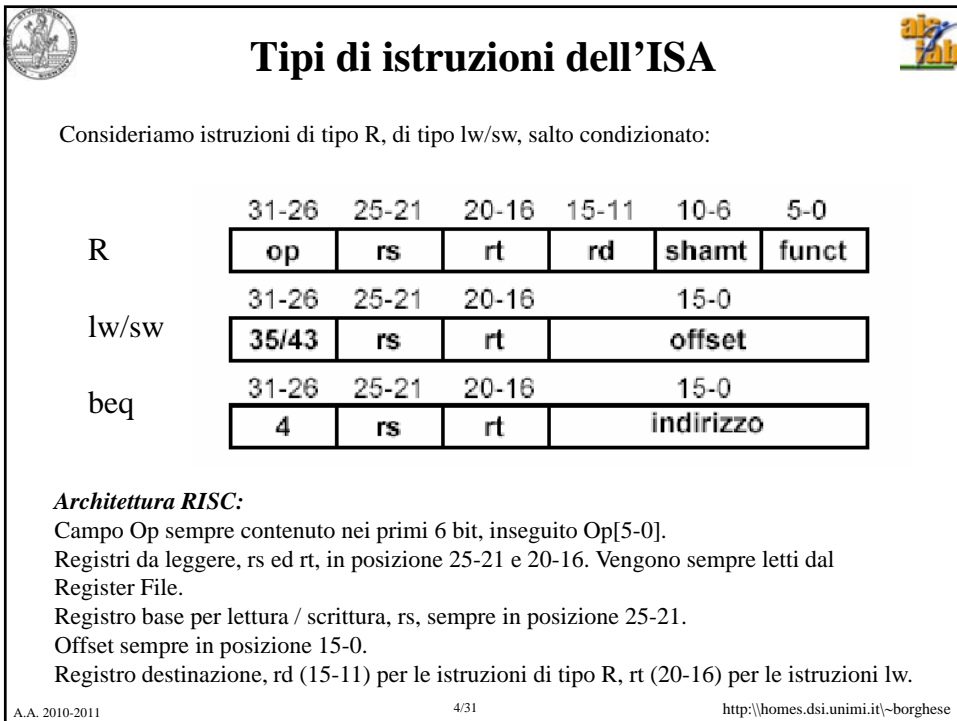
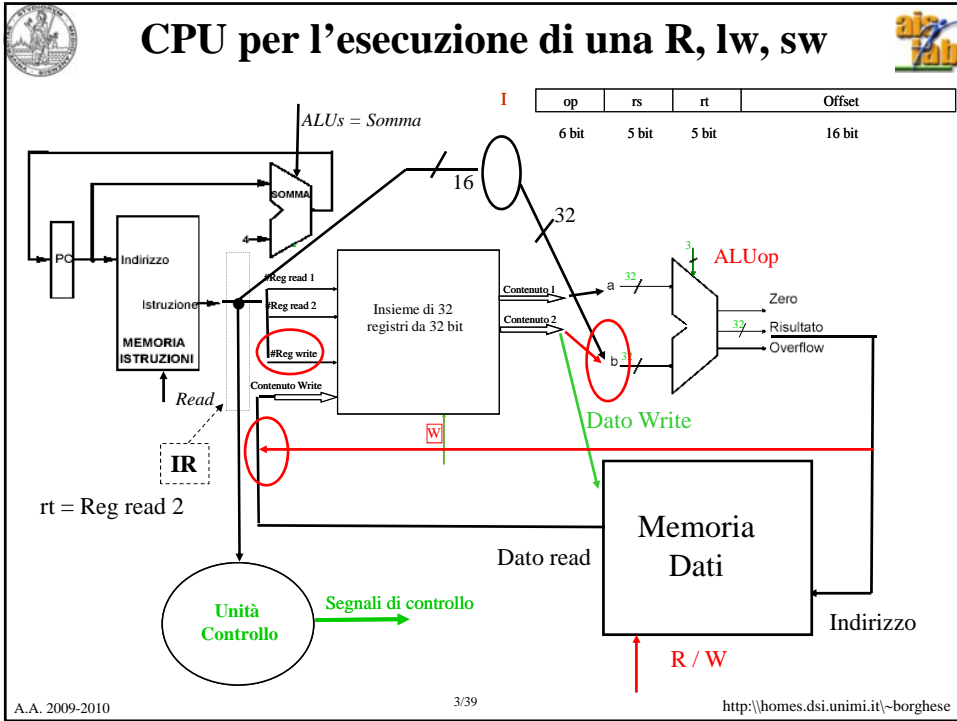


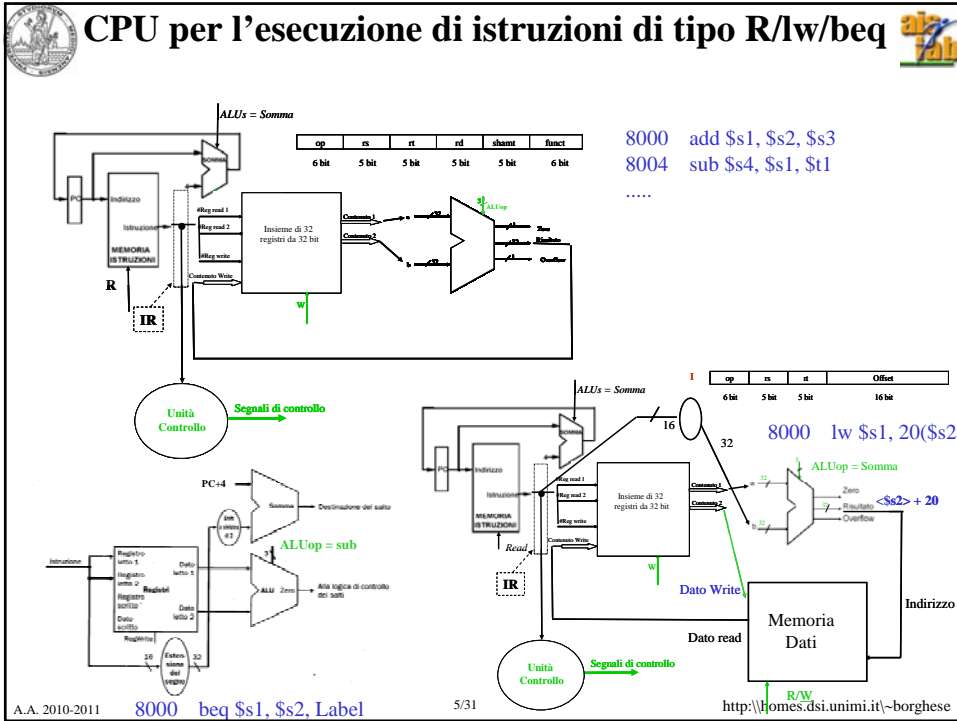
Sommario

CPU che gestisce istruzioni di tipo R, lw/sw, branch

Controllore della ALU

Unità di Controllo Principale





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.

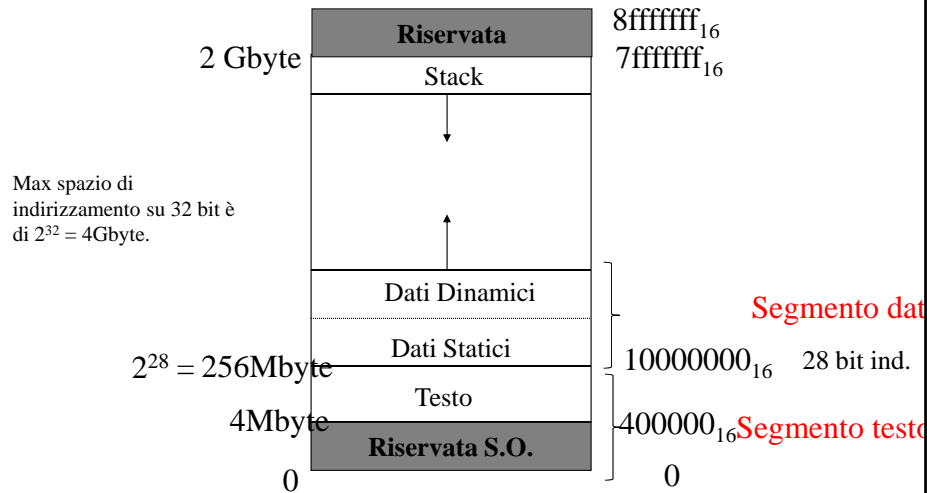
↓

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

A.A. 2010-2011 6/31 <http://homes.dsi.unimi.it/~borghese>



Organizzazione logica della memoria

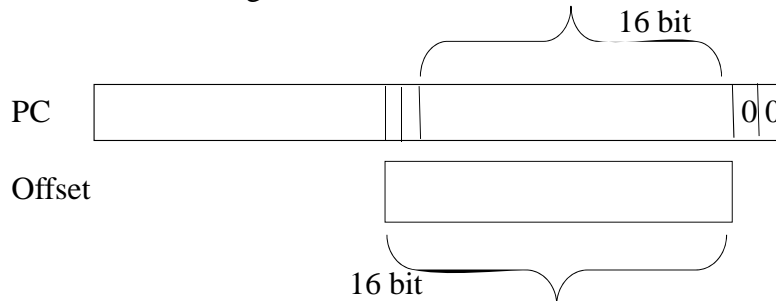


Allargamento dello spazio di indirizzamento



0000	0	0
0100	1	4
1000	2	8
1100	3	12

Considero 64Mword invece di 64Kbyte. Lo spazio indirizzabile all'interno del segmento di testo è di $64\text{Kword} * 4 = 256\text{Kbyte}$.





Indirizzamento della memoria MIPS



byte

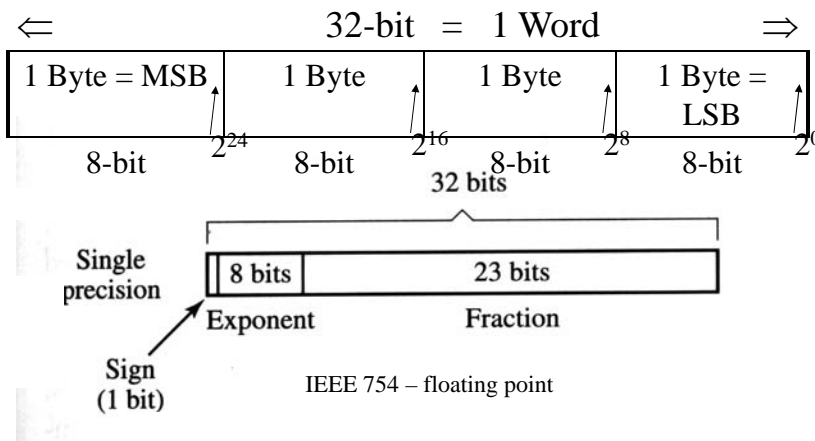
	2^k-4	2^k-3	2^k-2	2^k-1
12				
8				
4				
0	8	9	10	11
	4	5	6	7
	0	1	2	3



Indirizzamento dei byte all'interno della parola



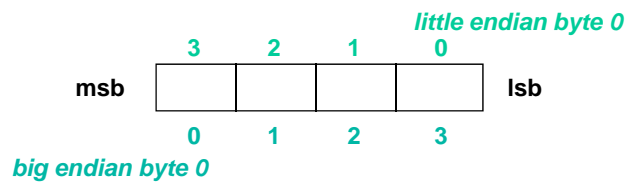
MIPS utilizza un **indirizzamento al byte**, cioè l'indice punta ad un byte di memoria, byte consecutivi hanno indirizzi consecutivi indirizzi di parole consecutive (adiacenti) differiscono di un fattore 4 ($8\text{-bit} \times 4 = 32\text{-bit}$): ad ogni indirizzo è associato un byte.



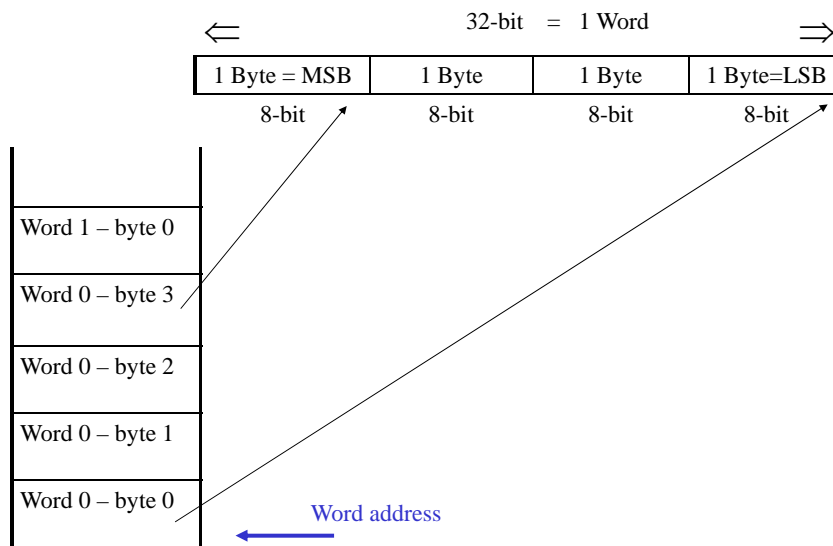


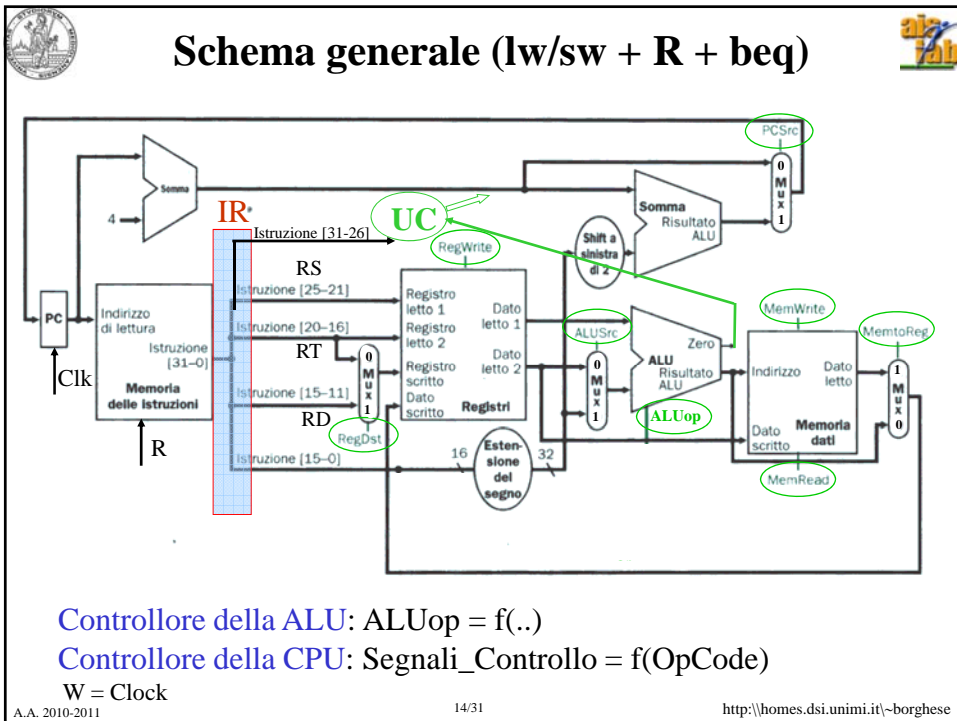
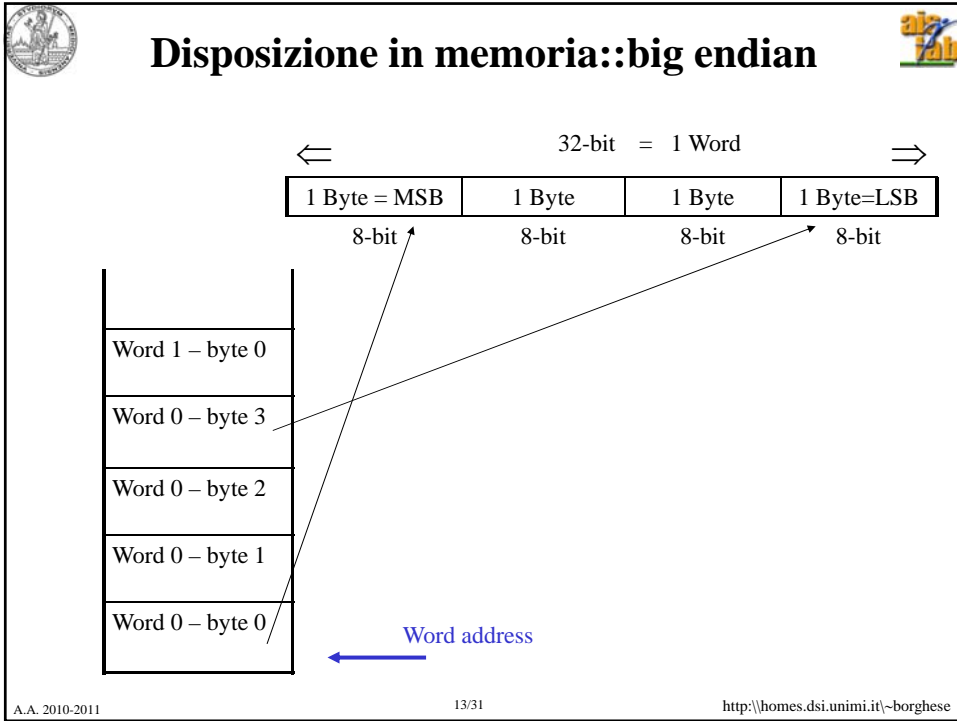
Addressing Objects: Endianess

- **Big Endian:** address of most significant byte = word address
(xx00 = Big End of word)
 - IBM 360/370, Motorola 68k, MIPS, Sparc, HP
- **Little Endian:** address of least significant byte = word address
(xx00 = Little End of word)
 - Intel 80x86, DEC Vax, DEC Alpha (Windows NT)



Disposizione in memoria::little endian







Sommario



CPU che gestisce istruzioni di tipo R, lw/sw, branch

Unità di Controllo Principale

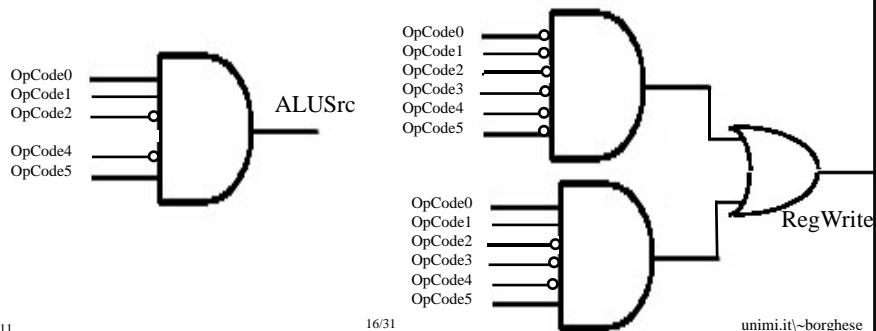
Controllore della ALU

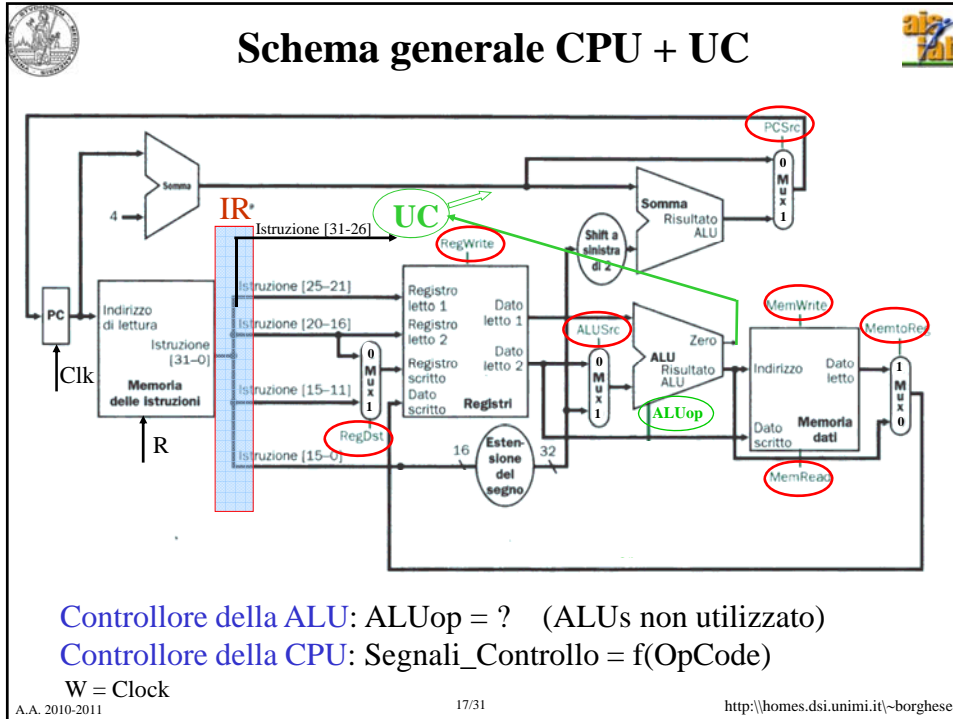


Controllo del data-path



Istruzione (OpCode)	RegDst	ALUSrc	Memto Reg	Reg Write	Mem Read	Mem Write	Branch	ALUs
R (000000)	1	0	0	1	0	0	0	10
lw (100011)	0	1	1	1	1	0	0	00
sw (101011)	x	1	x	0	0	1	0	00
beq (000100)	x	0	x	0	0	0	1	01





Segnali di controllo su 1 bit

Nome del segnale	Effetto quando è negato	Effetto quando è affermato
RegDst	Il numero del registro destinazione proviene dal campo rt (R2, bit 20-16)	Il numero del registro destinazione proviene dal campo rd (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 offset
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
MemtoReg	Il valore inviato all'ingresso Dato al Register File proviene dalla ALU	Il valore inviato all'ingresso Dato al Register File proviene dalla memoria

A.A. 2010-2011 18/31 http://homes.dsi.unimi.it/~borghese



Sommario



CPU che gestisce istruzioni di tipo R, lw/sw, branch

Controllore della ALU

Unità di Controllo Principale