

Prestazioni

		compilatore	
		A	B
processo	1		
	2		

$T_{CPU} =$

Benchmark

$$T_{CPU} = N_{CLK} \cdot T_{CLK} = \frac{N_{CLK}}{f_{CLK}}$$

$$N_{CLK} = \sum_i n_i CPI_i \approx N_{ist} \overline{CPI}$$

$$N_{ist} = \sum_i n_i$$

$$T_{CPU} \approx \frac{N_{ist} \times \overline{CPI}}{f_{CLK}}$$

Benchmarks

CIS
Nist
 $\overline{CPI}$

$\overline{CPI}$

$N_{ist} \times \overline{CPI}$

f_{CLK}

CISC

$N_{ist} \downarrow$

$\overline{CPI} \uparrow$

RISC

$N_{ist} \uparrow$

$\overline{CPI} \downarrow$

pipeline



Legge di Amdahl

accelerazione = $a = \frac{\text{prestazioni dopo il miglioramento}}{\text{prestazioni prima del miglioramento}}$

t_v vecchio

t_n nuovo

$$t_n = t_v (1-f) + \frac{t_v}{a} f$$

$$a_f = \frac{t_v}{t_n} = \frac{1}{(1-f) + f/a} = \frac{a}{f + a(1-f)}$$

$$a_f = \frac{1}{0,7 + \frac{0,3}{2}} \approx 1,18$$



H&H "Microarchitettura" (Cap. 7)
Bucci "La CPU" (Cap. 8)
e "La pipeline" (Cap. 9)

Macchine RISC

Architettura

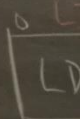
H&H ARM
Bucci ~MIPS

32 registri

32 bit

le istruzioni sono tutte di

es di



architettura" (cap. 7)

CPU" (cap. 8)

"pipeline" (cap. 9)

chine RISC

SH / ARM

ccì ~MIPS

32 registri

32 bit

le istruzioni sono tutte di 32 bit

classi di istruzioni

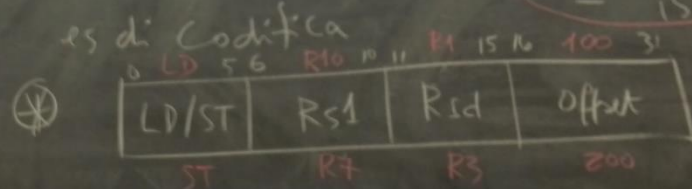
- logico-aritmetiche (ADD)

(*) che fanno riferimento alla memoria

- istr. di trasferimento del controllo

- istr. NOP

es di Codifica



SINTASSI

LD R1, 100(R10)

ST 200(R3), R7

classi di istruzioni

- logico-aritmetiche (ADD)

(*) che fanno riferimento alla memoria

- istr. di trasferimento del controllo

- istr. NOP

(LD, ST)

(JMP, JR, JAL, JE, JS...)

JEM.

incod. & immediato

incod. & rel.

"Jump and Link"

cond. & relativi

(CALL)

di 32 bit

Codifica

LD	5	6	R10	10	11	R1	15	16	100	31
D/ST	Rsd		Rsd		Offset					
ST	R7		R3		200					

SINTASSI

LD R1, 100(R10)

$R1 \leftarrow M(R10 + 100)$

ST 200(R3), R7 $M(R3 + 200) \leftarrow R7$

Organizzazioni

monociclo
multiciclo
pipeline

1 periodo di clock / istr., controllo comb.,
Mem. dati separata dalla Mem. Istruzioni
(modello HARVARD)

"datapath"

dell'istr.
(ST)



FASI di un'istruzione

IF	fetch
→ ID	decode
EX	execute
ME	memory
→ WB	write back

