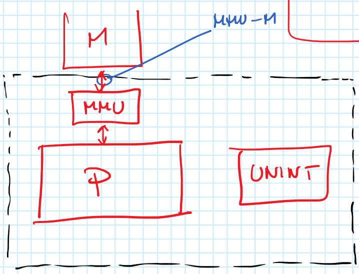
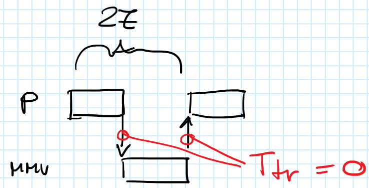
venerdì 11 novembre 2016 11:00

ch0. ch1. istr x0. ... istr x1.

τ t_a

FETCH
LOAD
STORE
($ACK_m = 0$) nop,



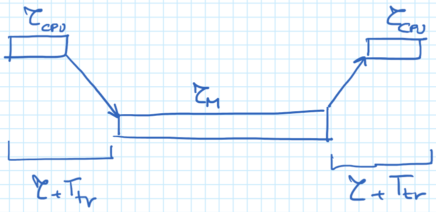


RAM { memorie "dinamiche" 60÷100 τ_{CPU}
memorie "statiche" 20÷40 τ_{CPU}

t_a

⇓

$2(\tau_{CPU} + T_{tr}) + \tau_M$



$$T = \tau \cdot \sum_{i=1}^{N istr} p_i \cdot k_i = \tau \, CPI$$

CPI

$\sum p_i = 1$

$$\text{Performance} = \frac{1}{T}$$

BENCHMARK

$T_c = m \cdot T$

hw

tempo di completamento

#istr

Mix

+ − SH AL corte	* / AL lunghe	LOAD/STORE	SALTO	SALTO COND
40%	10%	30%	10%	10%
$3\tau + t_a$	$52\tau + t_a$	$4\tau + 2t_a$	$3\tau + t_a$	$4\tau + t_a$

$$T = \sum \ldots = a\tau + bt_a = 8.3\tau + 1.3t_a$$

$t_a = 100\tau$

130τ
8.3τ
138τ

$t_a = 20\tau$

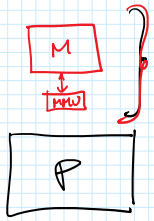
[Memoria statica < veloce, costosa]

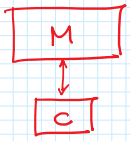
$1.3 t_a = 27\tau$
8.3τ
35.3τ

SUPERAMENTO dei limiti del Processore "monolitico"

venerdì 11 novembre 2016 11:46

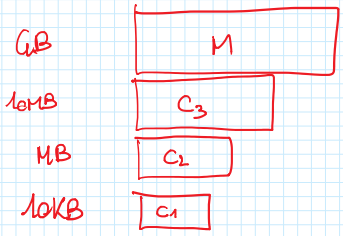
1) tempi di accesso alla memoria



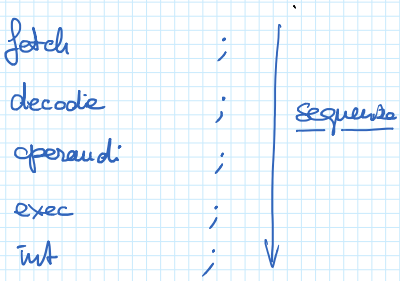


cache
+ piccola
+ veloce
+ costosa

Contiene un sottoinsieme delle locazioni di memoria di M



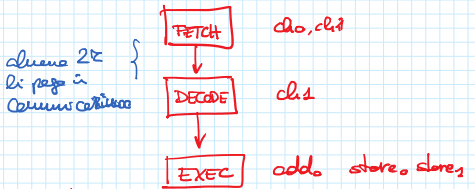
2) tempi di esecuzione delle istruzioni

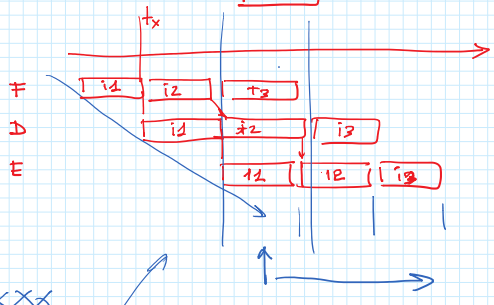


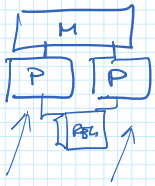
P PIPELINE

a catena di montaggio

P SUPERSCALARE







loop: XXX

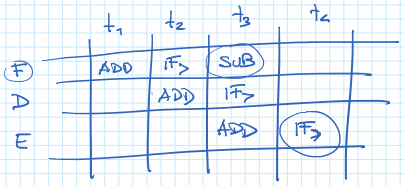
ADD R_1 R_2 R_3

IF> R_1 R_N loop

SUB R_8 R_9 R_{10}

ADD R1, R2, R3

SUB R3, R4, R5



Estensioni del set di istruzioni D-RISC

venerdì 11 novembre 2016 12:22

$x_1 x_2 x_3 \dots x_n$
$y_1 y_2 \dots y_n$
$\Big\}$ $x_1 \cdot y_1 + x_2 \cdot y_2 + \dots + x_n \cdot y_n$

```
sum = 0;
for (i=0; i<N; i++)
    sum = sum + x[i] * y[i];
```

R_a	R_b	R_c	R_d
8	6	6	6
MULANDADD	base x	base y	i	sum

MULANDADD $R_{basex}, R_{basey}, R_i, R_{sum}$

①	②	③	④
x[i]	y[i]	*	+

loop:
① LOAD R_{basex}, R_i, R_1
② LOAD R_{basey}, R_i, R_2
③ MUL R_1, R_2, R_3
④ ADD R_3, R_{sum}, R_{sum}
INC R_i
$IF_<$ R_i, R_N, loop

Hp: con ALU *

4τ + 2ta	
4τ + 2ta	
8τ + ta	
8τ + ta	
14τ + 6ta	

$2\tau + ta +$
$4\tau + 2ta$
$6\tau + 3ta$

muladdadd0. "read" → OP, REG[IR.RA] + REG[IR.RC] → IND, set RDYm, muladdadd1

muladdadd1. (ACKm, OR(ESITO) = 0−) nop, muladdadd1.
(= 10) DATAIN → TEMP1, reset ACKm, "read" → OP, REG[IR.RB] + REG[IR.RC] → IND, set RDYm, mul a2

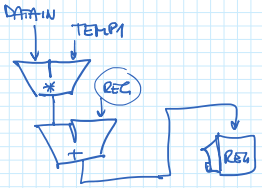
mul a2. (ACKm, OR(ESITO) = 0−) nop, mul a2
(= 10) DATAIN * TEMP1 → TEMP1, reset ACKm, mul a 3

mul 3. (INT = 0) TEMP + REG[IR.RD] → REG[IR.RD], IC+1 → IC, clock
(= 1) ....

$\Big\}$ (DATAIN * TEMP1) + REG[IR.RD] → REG[IR.RD], IC+1 → IC, clock

(ACKm, OR(ESITO), INT.

$5\tau' + 3ta \neq 4\tau + 6ta$



⇒ allungo Tclk rispetto all'interpete "standard"

$\tau \neq \tau'$ ⇐