

Nome Cognome, Numero Matricola:

## Grader per Esame: Architetture 2 - Canale A-L - 13.07.2011

Enrico Tronci  
Dip. di Informatica, Università di Roma "La Sapienza", Via Salaria 113, Roma, Italy  
tronci@di.uniroma1.it, <http://mclab.di.uniroma1.it>  
July 12, 2011

**Durata Esame: 30 min**

Rispondere al maggior numero di domande. Le risposte errate valgono punti -1. Non risposta vale punti 0. Le risposte esatte valgono punti 2.

Rispondere sul foglio risposte mettendo una croce **a penna** (non a matita) sulla risposta scelta. Se si decide di non rispondere mettere una croce sullo 0. Per ogni domanda deve esserci esattamente una croce sul foglio risposte.

### Q 1

Si consideri un processore i cui blocchi funzionali hanno i seguenti tempi di calcolo:

IF: In- struc- tion Fetch	RegR: Reg- ister Read	ALU Op- era- tion	RAM: Data Ac- cess	RegW: Reg- ister Write
200 ps	100 ps	200 ps	200 ps	100 ps

Assumendo di voler realizzare una implementazione del processore single-clock (cioè senza pipeline), qual'è il più piccolo clock cycle che si può usare.

1. 200 ps
2. 100 ps
3. **OK** 800 ps

### Q 2

Si consideri un processore i cui blocchi funzionali hanno i seguenti tempi di calcolo:

IF: In- struc- tion Fetch	RegR: Reg- ister Read	ALU Op- era- tion	RAM: Data Ac- cess	RegW: Reg- ister Write
200 ps	100 ps	200 ps	200 ps	100 ps

Si consideri una istruzione di *branching*, ad esempio `beq`. Qual'è il tempo necessario per eseguire una tale istruzione.

1. 600 ps
2. **OK** 500 ps
3. 700 ps

### Q 3

Si consideri l'implementazione M di un set di istruzioni (instruction set). Ci sono tre classi di

istruzioni (A, B, C) nel set di istruzioni. M ha un clock rate di 80 MHz. Il numero medio di cicli per ciascuna classe di istruzioni e la loro frequenza (per un programma tipico) sono riportate nella tabella seguente:

Classe di Istruzioni	Macchina M: Ci- cli/Classe di Istruzioni	Frequenza delle Istruzioni nelle Classe
A	1	60%
B	2	30%
C	4	10%

Si calcoli il CPI medio (average CPI) per la macchina M.

1. 4
2. 2
3. **OK** 1.6

### Q 4

Qual'è la size, in bits, di una direct-mapped cache con 16KB di dati e blocchi da 4 words, assumendo un indirizzo a 32 bits.

1. 128 Kbits
2. 148 Kbits
3. **OK** 147 Kbits

### Q 5

Si consideri un processore con i seguenti blocchi funzionali:

IF: In- struc- tion Fetch	RegR: Reg- ister Read	ALU Op- era- tion	RAM: Data Ac- cess	RegW: Reg- ister Write
---------------------------------------	--------------------------------	----------------------------	-----------------------------	---------------------------------

Si consideri una istruzione di `add`. Quali unità funzionali sono coinvolte ?

1. IF, RegR, ALU, RAM, RegW
2. IF, RegR, ALU, RAM
3. **OK** IF, RegR, ALU, RegW

**Q 6**

Si consideri un processore i cui blocchi funzionali hanno i seguenti tempi di calcolo:

IF: In- struc- tion Fetch	RegR: Reg- ister Read	ALU Op- era- tion	RAM: Data Ac- cess	RegW: Reg- ister Write
200 ps	100 ps	200 ps	200 ps	100 ps

Assumendo di voler realizzare una implementazione del processore con pipeline, qual'è il più piccolo clock cycle che si può usare.

1. **OK** 200 ps
2. 100 ps
3. 800 ps

**Q 7**

L'istruzione

`jal etichetta`

prefista dall'assembler MIPS si riferisce ad un operazione di

1. salto condizionato a funzione.
2. **OK** salto incondizionato a indirizzo con salvataggio dell'indirizzo di ritorno contenuto in \$ra.
3. salto incondizionato a indirizzo.

**Q 8**

Si consideri l'implementazioni M di un set di istruzioni (instruction set). Ci sono tre classi di istruzioni (A, B, C) nel set di istruzioni. M ha un clock rate di 100 MHz. Il numero medio di cicli per ciascuna classe di istruzioni e la loro frequenza (per un programma tipico) sono riportate nella tabella seguente:

Classe di Istruzioni	Macchina M: Ci- cli/Classe di Istruzioni	Frequenza delle Istruzioni nelle Classe
A	2	60%
B	3	30%
C	4	10%

Si calcoli il CPI medio (average CPI) per la macchina M.

1. **OK** 2.5
2. 3
3. 4

**Q 9**

Si consideri un processore i cui blocchi funzionali hanno i seguenti tempi di calcolo:

IF: In- struc- tion Fetch	RegR: Reg- ister Read	ALU Op- era- tion	RAM: Data Ac- cess	RegW: Reg- ister Write
200 ps	100 ps	200 ps	200 ps	100 ps

Si consideri una istruzione di *load word lw*. Qual'è il tempo necessario per eseguire una tale istruzione.

1. 400 ps
2. **OK** 800 ps
3. 600 ps

**Q 10**

Si consideri una macchina M con un un clock rate di 80 MHz ed un CPI di 1.6.

Si calcoli il MIPS medio (average MIPS) per la macchina M.

1. 30
2. 70
3. **OK** 50

**Q 11**

Esaminare il seguente codice assembler MIPS:

```
.data
vettore: .halfword 5,12,2,5,6,7,8
.text
.globl main
main:
la $a0, vettore
add $a0,$a0,4
```

al termine dell'esecuzione del suddetto codice, il registro \$a0 contiene:

1. il valore numerico 12.
2. l'indirizzo dell'elemento che ha valore numerico 12.
3. **OK** l'indirizzo dell'elemento che ha valore numerico 2;

**Q 12**

Individuare tra gli esempi sottostanti il codice assembler MIPS che realizza correttamente l'operazione di push sullo stack di tre parole di memoria corrispondenti ai contenuti dei registri \$ra, \$a0 e \$s0.

1. **OK**

```
sub $sp,$sp,12
sw $ra, ($sp)
sw $a0, 4($sp)
sw $s0, 8($sp)
```

2. `add $sp,$sp,12`  
`lw $ra, ($sp)`  
`lw $a0, 4($sp)`  
`lw $s0, 8($sp)`
3. `sub $sp,$sp,12`  
`sw $ra, $sp`  
`sw $a0, 4+$sp`  
`sw $s0, 8+$sp`

**Q 13**

Si consideri una cache con 64 blocchi (bloks) di dimensione (size) 16 bytes. In quale dei 64 blocchi della cache il byte address 1200 si mappa.

1. 7
2. **OK** 11
3. 37

**Q 14**

In una pipeline si ha *hazard strutturale* quando

1. L'esecuzione dell'istruzione pianificata non può avere luogo nel clock cycle previsto perchè l'hardware deve attendere i dati da una istruzione non ancora completata.
2. L'esecuzione dell'istruzione pianificata non può avere luogo nel clock cycle previsto a causa di un malfunzionamento di una unità funzionale.
3. **OK** L'esecuzione dell'istruzione pianificata non può avere luogo nel clock cycle previsto perchè l'hardware non supporta l'esecuzione simultanea della combinazione di istruzioni che sono previste per l'esecuzione.

**Q 15**

Branch Prediction ha l'obiettivo di ridurre gli stall nella pipeline dovuti a quali dei seguenti hazards ?

1. **OK** Control hazards
2. Hazard strutturali
3. Data hazard

**Q 16**

Si consideri un processore i cui blocchi funzionali hanno i seguenti tempi di calcolo:

IF: In- struc- tion Fetch	RegR: Reg- ister Read	ALU Op- era- tion	RAM: Data Ac- cess	RegW: Reg- ister Write
200 ps	100 ps	200 ps	200 ps	100 ps

Il processore è implementato disponendo le unità funzionali nella tabella in una pipeline con altrettanti stadi (5). Il clock cycle scelto è il minimo possibile con i dati in tabella.

Si calcoli il tempo necessario per eseguire 101 istruzioni `lw` consecutive assumendo che non ci siano stall nella pipeline.

1. 20200 ps

2. **OK** 21000 ps
3. 80800 ps

**Q 17**

In una pipeline si ha *hazard strutturale* quando

1. L'esecuzione dell'istruzione pianificata non può avere luogo nel clock cycle previsto perchè l'hardware deve attendere i dati da una istruzione non ancora completata.
2. L'esecuzione dell'istruzione pianificata non può avere luogo nel clock cycle previsto a causa di un malfunzionamento di una unità funzionale.
3. **OK** L'esecuzione dell'istruzione pianificata non può avere luogo nel clock cycle previsto perchè l'hardware non supporta l'esecuzione simultanea della combinazione di istruzioni che sono previste per l'esecuzione.

**Q 18**

In una pipeline il *forwarding* (o *bypassing*) è:

1. Un metodo per risolvere i data hazard leggendo i dati mancanti direttamente dal Register File.
2. Un metodo per risolvere i data hazard leggendo i dati mancanti direttamente dalla memoria.  
L'esecuzione dell'istruzione pianificata non può avere luogo nel clock cycle previsto perchè dei dati necessari per l'esecuzione dell'istruzione non sono disponibili.
3. **OK** Un metodo per risolvere i data hazard leggendo i dati mancanti da buffers interni piuttosto che attendere che tali dati arrivino da registri o memoria visibili al programmatore.

**Q 19**

Si assuma che il *miss rate* della cache istruzioni è del 2% mentre quello della cache dati è del 4%. Il processore ha un CPI di 2 quando non ci sono memory stalls (perfect caching). La miss penalty (per ogni tipo di miss) è 100 clock cycles. Si determini quanto sarebbe più veloce il processore se avessimo una cache perfetta che non ha mai misses. Si assuma che loads and stores sono il 36% delle istruzioni.

1. 1
2. 5
3. **OK** 2.72

%%%% SOLUTION

```
%
%I = Instruction count
%
%<Instruction miss cycles> =
% I * 2% * 100 = 2 * I.
%
%<Data miss cycles> =
% I * 36% * 4% * 100 =
% 1.44 * I.
%
%<Memory-stall cycles> =
% <Instruction miss cycles> +
```

```

%      <Data miss cycles> =
% 3.44 * I.
%
%<CPI with stall> =
% <CPI perfect> +
% (<Memory-stall cycles>/I) =
% 2 + 3.44 = 5.44.
%
%<Speed ratio> =
% <CPU time with stall>/
% <CPU time with perfect cache> =
% I * <CPI with stall> * <Clock Cycles> /
% I * <CPI perfect> * <Clock Cycles> =
% <CPI with stall>/<CPI perfect> =
% 5.44/2 = 2.72

```

### Q 20

In una pipeline si ha un *data hazard* quando

1. L'esecuzione dell'istruzione pianificata non può avere luogo nel clock cycle previsto perchè i dati necessari per l'esecuzione dell'istruzione sono in un'area di memoria non accessibile.
2. **OK** L'esecuzione dell'istruzione pianificata non può avere luogo nel clock cycle previsto perchè dei dati necessari per l'esecuzione dell'istruzione non sono disponibili.
3. L'esecuzione dell'istruzione pianificata non può avere luogo nel clock cycle previsto perchè i dati necessari per l'esecuzione dell'istruzione sono stati modificati.

### Q 21

Si consideri un processore i cui blocchi funzionali hanno i seguenti tempi di calcolo:

IF: In- struc- tion Fetch	RegR: Reg- ister Read	ALU Op- era- tion	RAM: Data Ac- cess	RegW: Reg- ister Write
200 ps	100 ps	200 ps	200 ps	100 ps

Si consideri una istruzione di *store word sw*. Qual'è il tempo necessario per eseguire una tale istruzione.

1. 600 ps
2. 800 ps
3. **OK** 700 ps

### Q 22

Si consideri un processore con i seguenti blocchi funzionali:

IF: In- struc- tion Fetch	RegR: Reg- ister Read	ALU Op- era- tion	RAM: Data Ac- cess	RegW: Reg- ister Write
200 ps	100 ps	200 ps	200 ps	100 ps

Si consideri una istruzione di *load word lw*. Quali unità funzionali sono coinvolte ?

1. **OK** IF, RegR, ALU, RAM, RegW

2. IF, RAM, RegW
3. RegR, ALU, RAM, RegW

### Q 23

Quale delle seguenti descrizioni per la *Branch Prediction* è corretta ?

1. Una ottimizzazione volta a ridurre il numero di stall nella pipeline. Branch Prediction può essere realizzata SIA in hardware che in software. Essa consiste nell'eseguire speculativamente le istruzioni su uno dei rami di una istruzione di branch assumendo che il branch è TAKEN.
2. Una ottimizzazione volta a ridurre il numero di stall nella pipeline. Branch Prediction può essere realizzata SOLO in hardware. Essa consiste nell'eseguire speculativamente le istruzioni su uno dei rami di una istruzione di branch.
3. **OK** Una ottimizzazione volta a ridurre il numero di stall nella pipeline. Branch Prediction può essere realizzata SIA in hardware che in software. Essa consiste nell'eseguire speculativamente le istruzioni su uno dei rami di una istruzione di branch.

### Q 24

Si consideri il MIPS datapath in Figura 1 dove le linee di connessione marcato con "X" si intendono rimosse. Nella situazione descritta, si consideri l'esecuzione dell'istruzione

```
add $s1, $s2, $s3
```

L'effetto di tale esecuzione è:

1. **OK** Calcolare la somma dei contenuti di \$s2 e \$s3 senza scrivere tale somma nel registro \$s1.
2. Calcolare la somma dei contenuti di \$s2 e \$s3 scrivendo tale somma nel registro \$s1.
3. La somma non viene calcolata e nel registro \$s1 non viene scritto nulla.

### Q 25

Si consideri il MIPS datapath in Figura 2 dove le linee di connessione marcato con "X" si intendono rimosse.

Nella situazione descritta, si consideri l'esecuzione dell'istruzione

```
sw $s1, 0($s2)
```

L'effetto di tale esecuzione è:

1. Scrivere nella RAM all'indirizzo \$s2 il valore 0.
2. A causa dei tagli sulle interconnessioni non può scrivere in RAM, quindi l'istruzione non ha alcun effetto.
3. **OK** Scrivere nella RAM all'indirizzo \$s2 il valore nel registro \$s1.

### Q 26

Si consideri un processore con i seguenti blocchi funzionali:

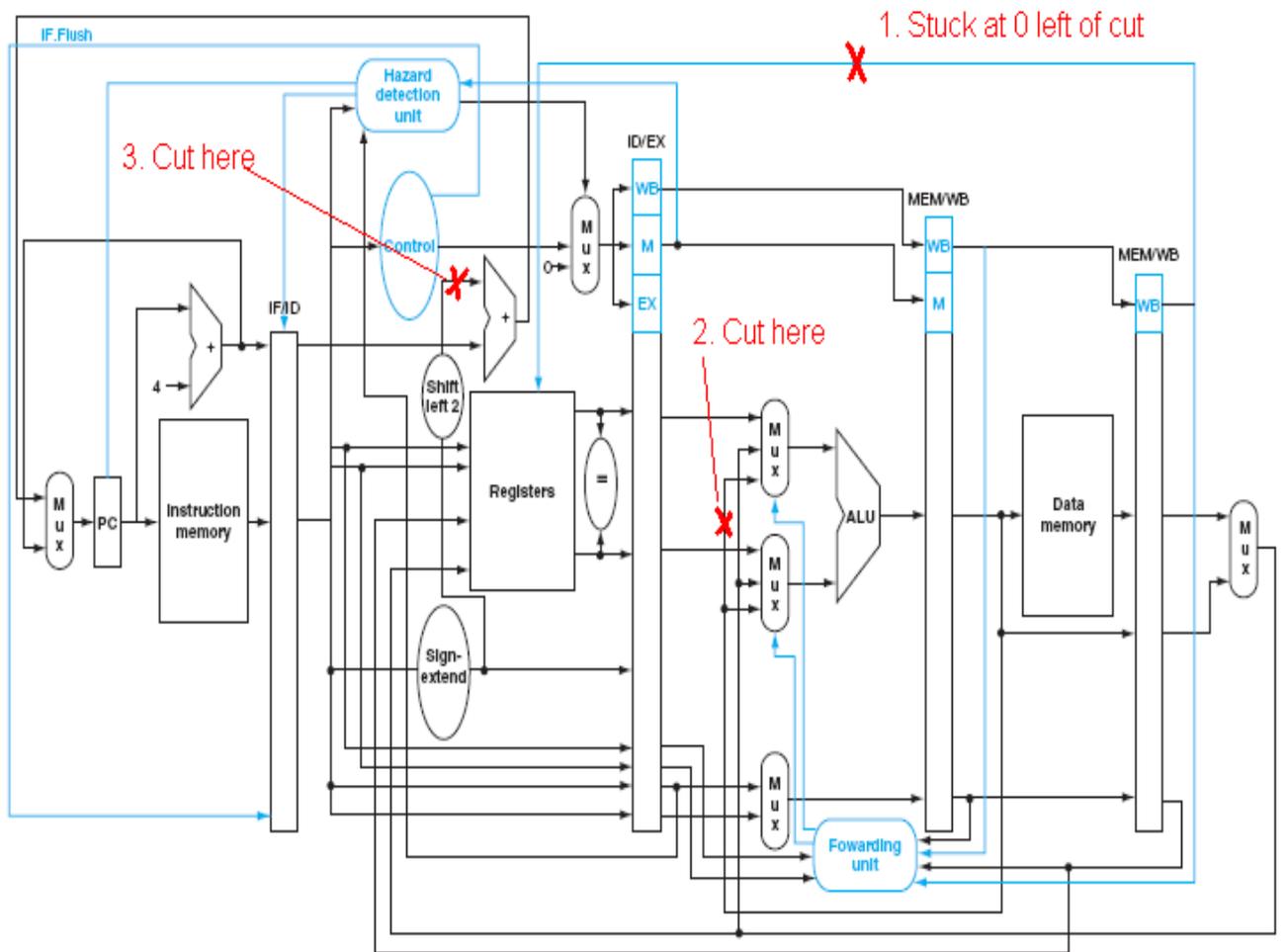


Figure 1: Datapath 1

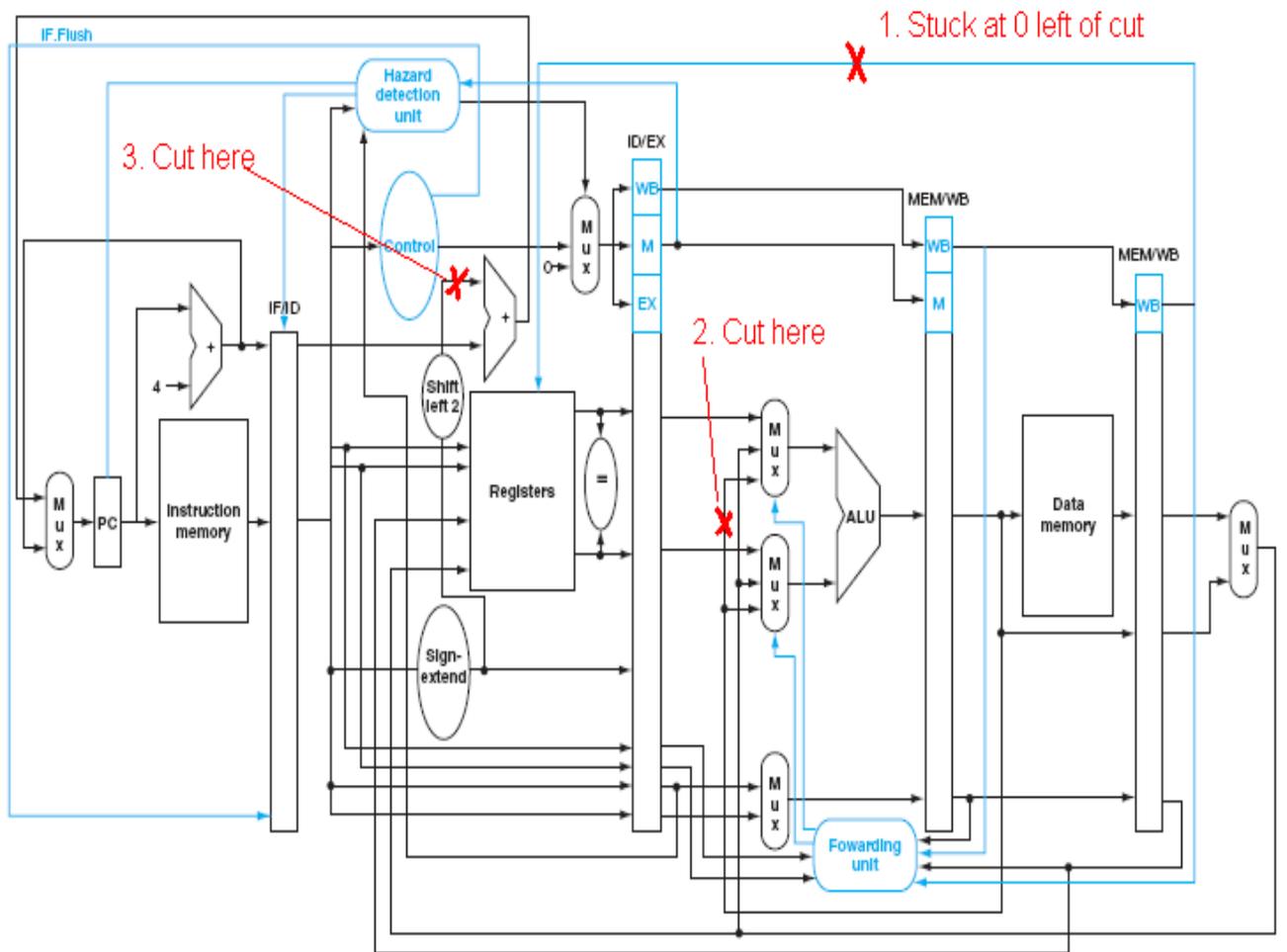


Figure 2: Datapath 2

IF: In- struc- tion Fetch	RegR: Reg- ister Read	ALU Op- era- tion	RAM: Data Ac- cess	RegW: Reg- ister Write
---------------------------------------	--------------------------------	----------------------------	-----------------------------	---------------------------------

Si consideri una istruzione di *store word sw*. Quali unità funzionali sono coinvolte ?

1. IF, RegR, ALU, RAM, RegW
2. **OK** IF, RegR, ALU, RAM
3. IF, ALU, RAM, RegW

### Q 27

Si consideri un processore i cui blocchi funzionali hanno i seguenti tempi di calcolo:

IF: In- struc- tion Fetch	RegR: Reg- ister Read	ALU Op- era- tion	RAM: Data Ac- cess	RegW: Reg- ister Write
200 ps	100 ps	200 ps	200 ps	100 ps

Si consideri una istruzione di *add*. Qual'è il tempo necessario per eseguire una tale istruzione.

1. **OK** 600 ps
2. 500 ps
3. 700 ps

### Q 28

*Instruction Scheduling* è una ottimizzazione volta a ridurre il numero di stall nella pipeline. Uno dei suoi effetti è:

1. **OK** Ridurre gli hazard strutturali spostando in posizioni distanziate le istruzioni che usano le stesse unità funzionali.
2. Ridurre i control hazards riducendo il numero delle istruzioni di branching.
3. Ridurre i data hazards spostando vicino ad A le istruzioni che dipendono da A.

### Q 29

Si consideri un processore con i seguenti blocchi funzionali:

IF: In- struc- tion Fetch	RegR: Reg- ister Read	ALU Op- era- tion	RAM: Data Ac- cess	RegW: Reg- ister Write
---------------------------------------	--------------------------------	----------------------------	-----------------------------	---------------------------------

Si consideri una istruzione di *branching*, ad esempio *beq*. Quali unità funzionali sono coinvolte ?

1. **OK** IF, RegR, ALU
2. **OK** IF, RegR, ALU, RAM
3. IF, RegR, ALU, RegW

### Q 30

*Instruction Scheduling* è una ottimizzazione volta a ridurre il numero di stall nella pipeline. Una dei suoi effetti è:

1. Ridurre i control hazards spostando opportunamente le istruzioni di branching.
2. **OK** Ridurre i data hazards spostando istruzioni che non dipendono da una istruzione A prima di istruzioni che dipendono da A.
3. Ridurre gli hazard strutturali spostando le istruzioni che non dipendono da una istruzione A dopo le istruzioni che dipendono da A.

### Q 31

Si supponga di avere una macchina M1 che tipicamente esegue programmi consistenti di istruzioni distribuite come segue: 50% floating point multiply, 20% floating point divide, 30% altre istruzioni.

Si vuole ottenere una macchina M2 che esegue 4 volte più veloce di M1 gli stessi programmi di cui sopra. Si può apportare al più una delle modifiche che segue: a) rendere la floating point divide 3 volte più veloce; b) rendere la floating point multiply 8 volte più veloce.

Quale modifica scegliereste (si può usare la legge di Amdahl) ?

1. Rendere la floating point divide 3 volte più veloce;
2. **OK** Rendere la floating point multiply 8 volte più veloce.
3. Nessuna delle due modifiche soddisfa i requisiti.