

Esercizio 1

Sia data la seguente sequenza di istruzioni MIPS e si consideri la pipeline a 5 stadi vista a lezione con possibilità di data forwarding e di scrittura e successiva lettura dei registri in uno stesso ciclo di clock:

```
LW    $1, 3($3)
ADD   $4, $1, $2
SW    $2, 82($3)
ADDI  $2, $3, 83
```

Quale delle seguenti sequenze riordinate corrisponde ad un programma logicamente equivalente al precedente ed è completato in meno cicli di clock:

☐ a

```
      LW    $1, 3($3)
      ADD   $4, $1, $2
      ADDI  $2, $3, 83
      SW    $2, 82($3)
```

☐ b

```
      LW    $1, 3($3)
      SW    $2, 82($3)
      ADD   $4, $1, $2
      ADDI  $2, $3, 83
```

☐ c

```
      LW    $1, 3($3)
      ADDI  $2, $3, 83
      ADD   $4, $1, $2
      SW    $2, 82($3)
```

☐ d

```
      SW    $2, 82($3)
      LW    $1, 3($3)
      ADD   $4, $1, $2
      ADDI  $2, $3, 83
```

☐ e nessuna delle risposte precedenti

Esercizio 2

Si consideri la pipeline MIPS a 5 stadi vista a lezione, con possibilità di data-forwarding e con possibilità di scrittura e successiva lettura dei registri in uno stesso ciclo di clock.

Per ognuna delle seguenti sequenze di istruzioni assembler, dove i dati immediati sono espressi in esadecimale, si chiede di:

1. indicare quali dipendenze dai dati sono presenti;
2. mostrare come evolve la pipeline durante l'esecuzione del codice, spiegando nel dettaglio i motivi di un eventuale stallo o dell'utilizzo di un particolare circuito di by-pass;
3. mostrare come evolverebbe la pipeline se non ci fosse la possibilità di effettuare data-forwarding.

Sequenza 1

```
LW    $1, 54($1)
SUB    $2, $4, $1
LW    $4, 0($2)
SUBI   $4, $2, 8
SUBI   $2, $4, 11
SW     $4, 74($2)
ADD    $3, $1, $2
```

Sequenza 2

```
ADD    $1, $2, $3
ADDI   $2, $1, 51
SW     $1, 21($3)
LW     $3, 15($4)
SUBI   $4, $3, 7
SUBI   $3, $4, 6
BEQ    $3, $4, 35
```