

-

Università degli Studi di Milano - Corso Architettura degli elaboratori II – Prof. Borghese
Prima prova in itinere 31.03.2025

Cognome e nome dello studente:

Matricola:

Il compito è diviso in due sezioni: la prima sezione a risposta multiple e la seconda sezione a risposte aperte. Per raggiungere la sufficienza in questa prova **occorre raggiungere almeno 6,5 punti** con le risposte ai quesiti della prima sezione.

SEZIONE 1 (Punteggio massimo: 10 punti. Per ogni risposta corretta, +1 punto; per ogni risposta sbagliata, -0,25 punti, per ogni risposta mancante 0 punti).

Barrare la risposta più corretta tra le 5 risposte possibili: A, B, C, D, E

Data la CPU di Figura 1 e il seguente frammento di codice:

```
0x00000400 sub $s5, $t2, $t2
0x00000404 sw $s1, 16($s0)
0x00000408 addi $t4, $s4, 4
0x0000040C add $t1, $s1, $s2
0x00000410 lw $s2, 32($t4)
```

Supporre che l'istruzione di sub sia in fase di WriteBack e rispondere alle seguenti domande:

1. Cosa passa nel cammino a?
 - A. Il contenuto del registro \$s5
 - B. Zero
 - C. 10
 - D. Non si può determinare
 - E. Il codice operativo
2. Cosa passa nel cammino b?
 - A. 4
 - B. 16
 - C. Niente
 - D. 0x40C
 - E. 0x41C
3. Cosa passa nel cammino c?
 - A. Il contenuto del registro \$t4
 - B. Il contenuto del registro \$s4
 - C. 16
 - D. 4
 - E. Non si può determinare

Data la CPU di Figura 2 e il seguente frammento di codice:

```
0x00000400 and $s5, $t1, $t1
0x00000404 sw $s1, 16($s0)
0x00000408 add $t4, $s5, $t2
0x0000040C addi $t1, $s1, 100
0x00000410 lw $s2, 32($t4)
```

Supporre che l'istruzione di and sia in fase di WriteBack e rispondere alle seguenti domande:

4. Cosa passa nel cammino a?
 - A. Il contenuto del registro \$s5
 - B. Il contenuto del registro \$t1
 - C. Il contenuto del registro \$t2
 - D. Il contenuto del registro \$t4
 - E. Non si può determinare

5. Cosa passa nel cammino b?
 - A. 32
 - B. 16
 - C. 0x44C
 - D. Il contenuto del registro \$t2
 - E. Il contenuto del registro \$s5
6. Cosa passa nel cammino c?
 - A. Il contenuto del registro \$s1
 - B. Il contenuto del registro \$t1
 - C. 16
 - D. 32
 - E. 100
7. Il flush di un'istruzione in una pipeline
 - A. E' il blocco di una pipeline per un ciclo di clock
 - B. Si verifica quando vengono messi a 0 i segnali di controllo di un'istruzione in una determinata fase e l'istruzione verrà poi fatta ripartire da quella fase con i segnali di controllo attivi.
 - C. Si verifica quando un'istruzione viene eliminata dalla pipeline.
 - D. Si verifica quando viene eseguito il reset del registro IF/ID.
 - E. Si verifica quando viene eseguito il reset del PC.
8. Una pipeline aumenta la velocità di esecuzione
 - A. Di un fattore pari al numero di cammini paralleli
 - B. Di un fattore pari al numero di stadi.
 - C. A seconda del tipo di ISA che si ha a disposizione.
 - D. Solo se si può operare l'over-clocking
 - E. Se il mix di istruzioni lo consente.
9. Cosa è un interrupt?
 - A. Un'interruzione del ciclo di esecuzione della CPU
 - B. Un evento non previsto che deve essere gestito dalla CPU e dal Sistema Operativo
 - C. Un segnale aggiuntivo per l'UC
 - D. Un errore che si verifica nella CPU
 - E. Una richiesta di attenzione fatta alla CPU
10. Le CPU multiple-issue
 - A. Sono sinonimo di CPU SIMD
 - B. Sono CPU che eseguono in parallelo più istruzioni non in pipeline
 - C. Sono CPU che eseguono in parallelo più istruzioni in pipeline
 - D. Sono CPU che riorganizzano il codice per evitare stalli
 - E. Sono CPU che sono in grado di gestire più di un interrupt alla volta

SEZIONE 2 (punteggio massimo 26 punti)

1. [5] Data la CPU N. 2, quando è in esecuzione il seguente segmento di codice e l'istruzione di `and` si trova in fase di WriteBack:

```
0x0000 0400    and $s4, $t1, $t1
0x0000 0404    sw $s1, 8($s4)
0x0000 0408    sub $t1, $t0, $t3
0x0000 040C    sw $s4, 32($s4)
0x0000 0410    lw $t1, 64($s4)
```

- a) Indicare cosa viene trasportato da **TUTTI** i bus presenti nello schema [3]
- b) sottolineare quali linee trasportano segnali utili [1].
- c) Identificare tutte le dipendenze e gli hazard [1].

2. [5] Cosa sono gli interrupt e le eccezioni? Come vengono gestiti dalle architetture Intel e dalle architetture MIPS/ARM? Specificare gli elementi della CPU MIPS che sono dedicati alla gestione delle eccezioni e cosa contengano. Identificare nella CPU di Figura 2 i componenti specificatamente disegnati per potere gestire

un'eccezione di "Overflow" e il loro funzionamento, aggiungere eventuali cammini mancanti. Cosa si intende per mascheramento degli interrupt? Viene praticato nei MIPS? Come vengono gestite le eccezioni e gli interrupt sui MIPS dai sistemi operativi? Scrivere uno scheletro di possibile codice.

3. [3] Cosa si intende per speculazione? Cosa è un BPB? Cosa contiene e a che cosa serve? Come viene realizzato il meccanismo di predizione dei salti che cambi predizione ogni due errori consecutivi?

4. [4] Modificare la pipeline in Figura 1 perché diventi una pipeline superscalare. Spiegare la ragione e lo scopo di **tutte le modifiche** più rilevanti da apportare ai diversi stadi. Che differenza c'è tra pipeline super-scalare e pipeline dotata di VLIW? Specificare il ruolo e il funzionamento di: a) Reservation station; b) Buffer di riordino; c) Ridenominazione dei registri. Quali sono i vantaggi e gli svantaggi di un approccio rispetto all'altro. Qual è il migliore e perché? Cosa si intende per esecuzione fuori-ordine? Quali componenti sono implicati e perché? Cosa è un issue?

5. [2] Spiegare chiaramente cosa si intende per stallo e cosa si intende per flush di una pipeline e farne un esempio.

6. [4] Riscrivere il codice seguente in modo che sia eseguito nel minor tempo possibile su una pipeline con 4 cammini di esecuzione di cui 3 in grado di eseguire le istruzioni aritmetico-logiche e di salto e 1 dedicato solo alle istruzioni di memoria. Qual è lo speed-up? (rapporto tra le prestazioni prima della modifica, sul codice sequenziale, e dopo la modifica). Applicare lo srotolamento dei cicli applicato a un blocco di 8 iterazioni del ciclo. Si supponga di avere a disposizione un numero di registri interni di pipeline sufficientemente grande e che si possa applicare la ridenominazione dei registri.

```
Ciclo:    lw    $t0, 0($s1)           # M[s1] -> t0
          addu  $t0, $t0, $s2        # t0 = t0 + s2
          sw    $t0, 0($s1)          # M[s1] <- t0
          addi  $s1, $s1, -4         # next element
          bne   $s1, $zero, Ciclo
          or    $s6, $s7, $s5
```

Il codice assembler corrisponde al seguente codice C:

```
Ciclo:    v[s1] = s2 + v[s1];
          s1--;
          if (s1 != 0) goto Ciclo
```

7. [3] Modificare la pipeline di Figura 2, in modo che gestisca correttamente l'hazard generato da un'istruzione lw seguita da un'istruzione di sw. Fare un esempio.

Figura 1

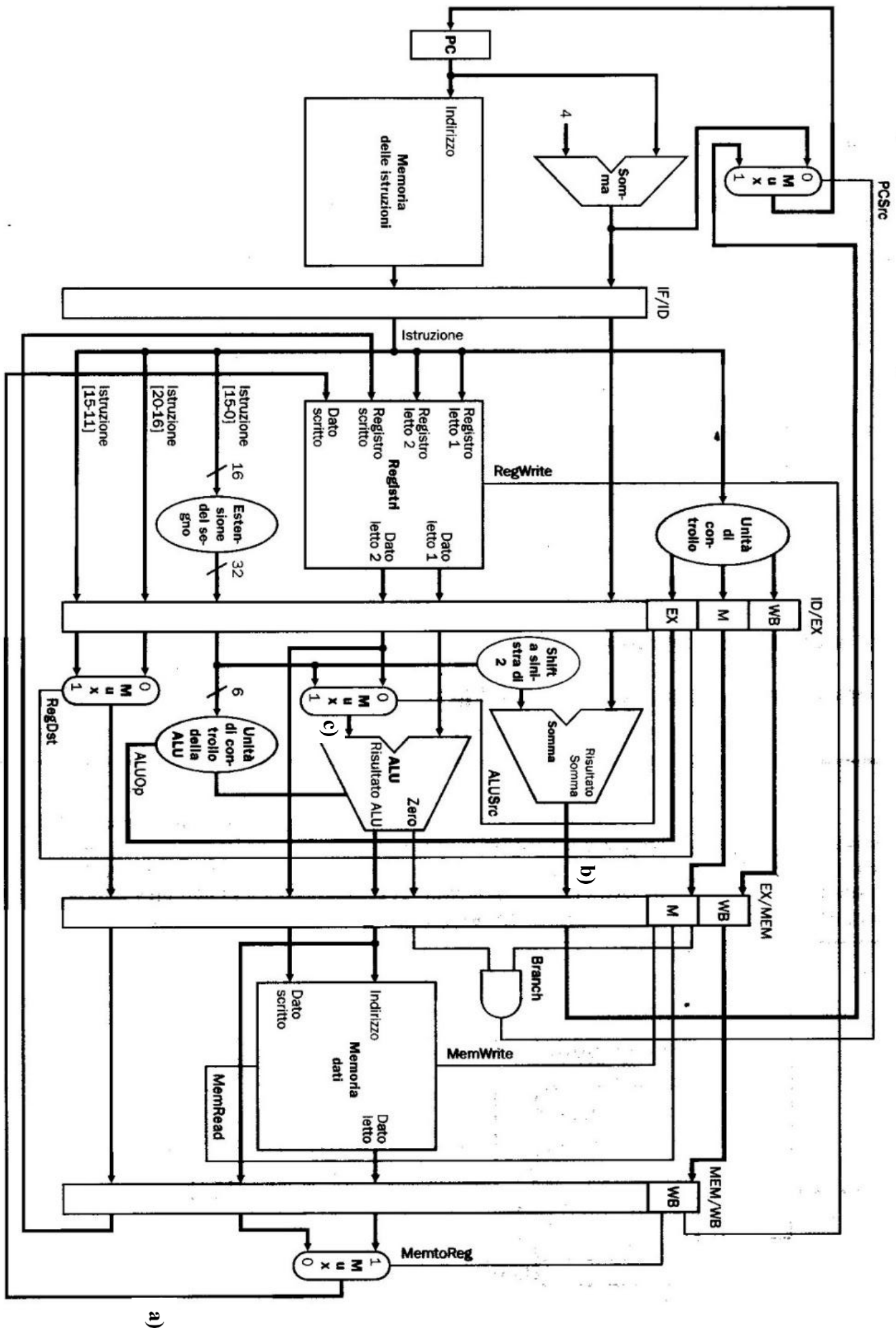
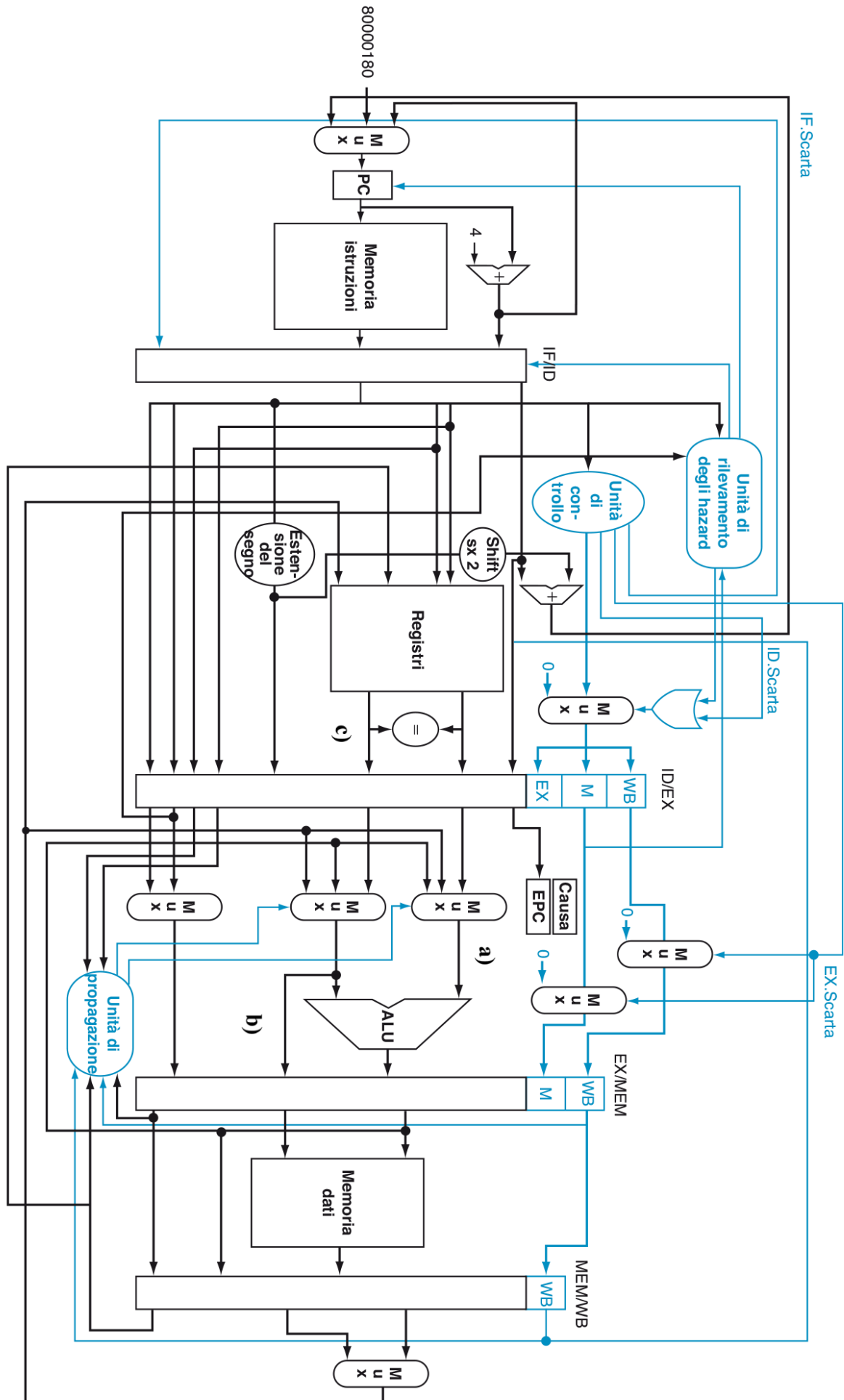


Figura 2



I registri dell'architettura MIPS

0	zero	constant 0
1	at	reserved for assembler
2	v0	expression evaluation &
3	v1	function results
4	a0	arguments
5	a1	
6	a2	
7	a3	
8	t0	temporary: caller saves
...		(callee can clobber)
15	t7	
16	s0	callee saves
...		(caller can clobber)
23	s7	
24	t8	temporary (cont'd)
25	t9	
26	k0	reserved for OS kernel
27	k1	
28	gp	Pointer to global area
29	sp	Stack pointer
30	fp	frame pointer (s8)
31	ra	Return Address (HW)