

Cognome e nome dello studente:

Matricola:

1. [5] Data la CPU di Figura 1, specificare il contenuto di **tutti** i registri (parte master e parte slave), quando è in esecuzione il seguente segmento di codice [5]:

```
0x0000 0400    addi $t3, $t1, 24
0x0000 0404    and $s0, $t1, $t1
0x0000 0408    sub $t1, $t2, $t3
0x0000 040C    sw $t1, 32($s0)
0x0000 0410    lw $t1, 64($s0)
```

quando l'istruzione di `addi` si trova in fase di WB. Specificare sullo schema (con colore o con tratto grosso) quali linee, all'interno dei diversi stadi, trasportino dati utili all'esecuzione dell'istruzione correntemente in quello stadio. Ci sono hazard nel codice precedente? Motivare la risposta. Modificare eventualmente la CPU in modo tale che questo codice venga eseguito correttamente.

2. [2] Modificare la CPU in Figura 2, in modo che possa gestire gli hazardi sui dati, quando è in esecuzione il seguente segmento di codice:

```
0x0000 0408    add $t1, $t2, $t3
0x0000 040C    beq $t1, $t2, 100
```

3. [6] Modificare la pipeline in Figura 4 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? Quali sono i vantaggi e gli svantaggi di un approccio rispetto all'altro. Qual è il migliore e perché? Descrivere come funzionano le seguenti tecniche e dire se sono tecniche principalmente **software** o **hardware** e perché. In alcuni casi la risposta corretta può essere entrambi gli approcci. Identificare quali sono i **punti forti** ed i **punti deboli**.

- a) Predizione dei salti
- b) Branch prediction buffer
- c) Speculazione
- d) Parallelizzazione dell'esecuzione
- e) Pipeline superscalari
- f) Esecuzione fuori ordine
- g) Reservation station
- h) Buffer di riordino
- i) Ridenominazione dei registri
- j) Issue

4. [1] Spiegare chiaramente cosa si intende per stallo e illustrare almeno una tecnica per ottenerlo.

5. [4] Disegnare una memoria cache (parte dati + TAG + bit di validità) e la porta di lettura / scrittura (una unica porta). Utilizzare componenti three-state per minimizzare il numero di porte logiche. Considerate un'architettura MIPS a 64 bit, a 4 vie di 128 KByte per banco, e linee di 8 parole (per ciascun banco). Definire cosa rappresenta il campo TAG e dimensionarlo opportunamente. Dove posso trovare il dato letto dall'istruzione `lw $t1, 2048($0)`? Da quanti bit è costituita questa memoria complessivamente? Cosa succede quando si verifica una miss? Come si può limitare la frequenza di miss? Spiegare cosa si intende per tecnica "Least Recently Used" esatta e approssimata.

6. [3] Realizzare la cache di cui sopra con chip 512 x 16 bit utilizzando il più possibile la tecnologia three-state per abilitare / disabilitare i chip di memoria e risparmiare sul numero di componenti.

7. [3] Cosa sono i codici di rilevamento e correzione degli errori? Come funziona il codice di Hamming? Calcolare il codice per il numero binario 1000 0001. Chi utilizza il codice di rilevamento e correzione degli errori? Schizzare uno schema a blocchi di un modulo, che mediante il codice di Hamming, possa correggere un errore singolo.

8. [3] Che cos'è e che cosa rappresenta il "roof model"? Cosa rappresenta l'intensità aritmetica? Si riferisce a una CPU o ad un particolare programma? Descrivere il metodo con il quale si può determinare la linea che rappresenta il roof model. Quali sono i passi per ottimizzare le prestazioni del codice suggeriti dal roof-model? Cos'è un kernel benchmark? Cos'è lo SPEC?

9. [2] Riportare alcune caratteristiche della architetture x86. A cosa serve il post-byte?

10. [3] Cosa si intende per gerarchia delle memorie? Spiegare chiaramente cosa si intenda per **coerenza** e **consistenza** di una memoria. Fare degli esempi. A quali memorie si applicano? Spiegare come funzionano i seguenti protocolli che mirano a garantire la coerenza:

- a) Write-back
- b) Write-through
- c) Write invalidate

Cos'è il lock? A cosa serve?

11. [3] Cos'è la memoria virtuale? Cos'è la Tabella delle pagine? Dove si trova? Cos'è il "Translation Lookaside buffer"? Dove si trova? A cosa servono la memoria virtuale, il TLB e la tabella delle pagine? Che relazione c'è tra la memoria virtuale e la memoria fisica? Chi utilizza la memoria virtuale? Chi utilizza la memoria fisica? Cosa succede quando la CPU chiede una parola alla memoria?

Registri del register file

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

Figura 1

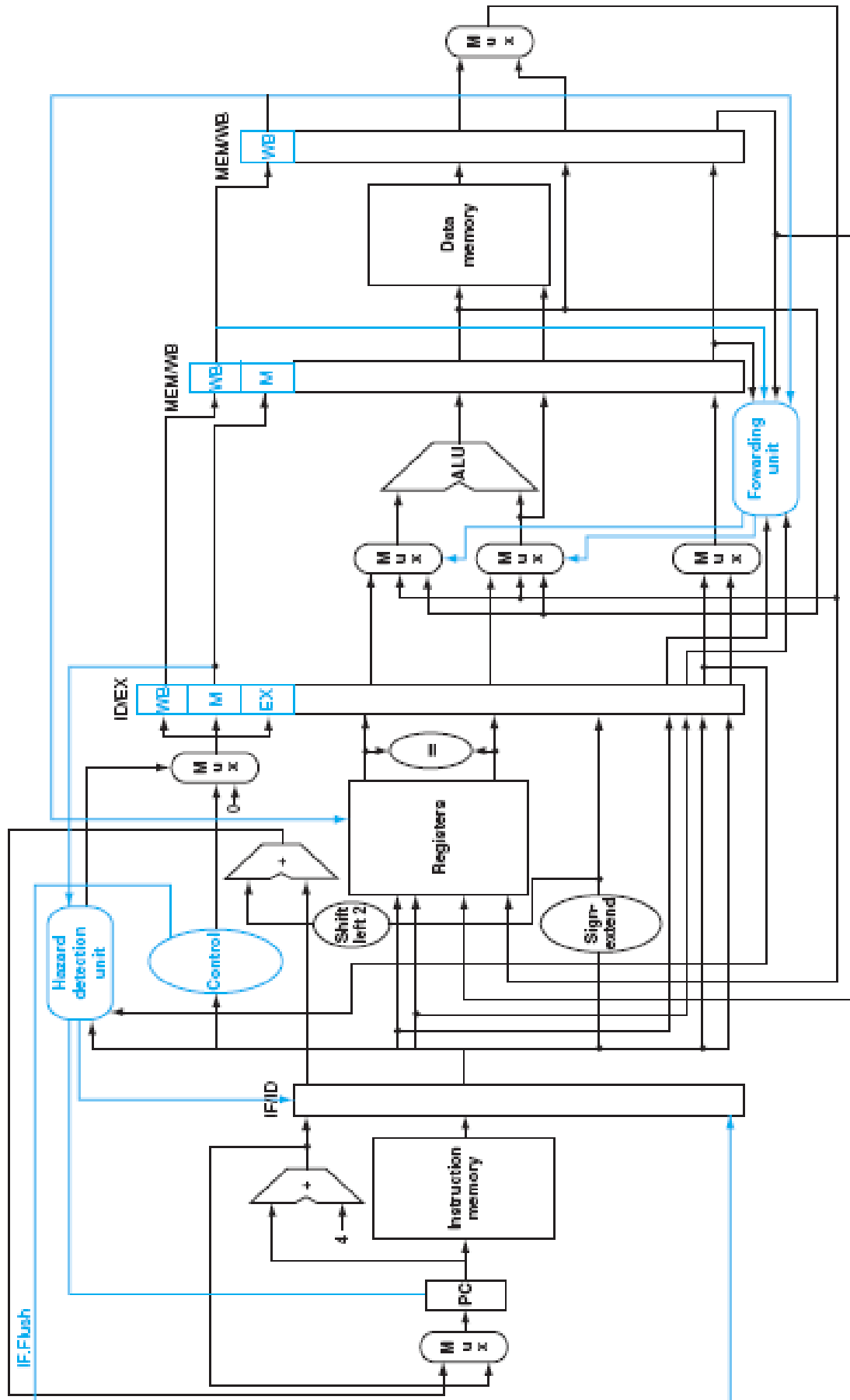


Figura 2

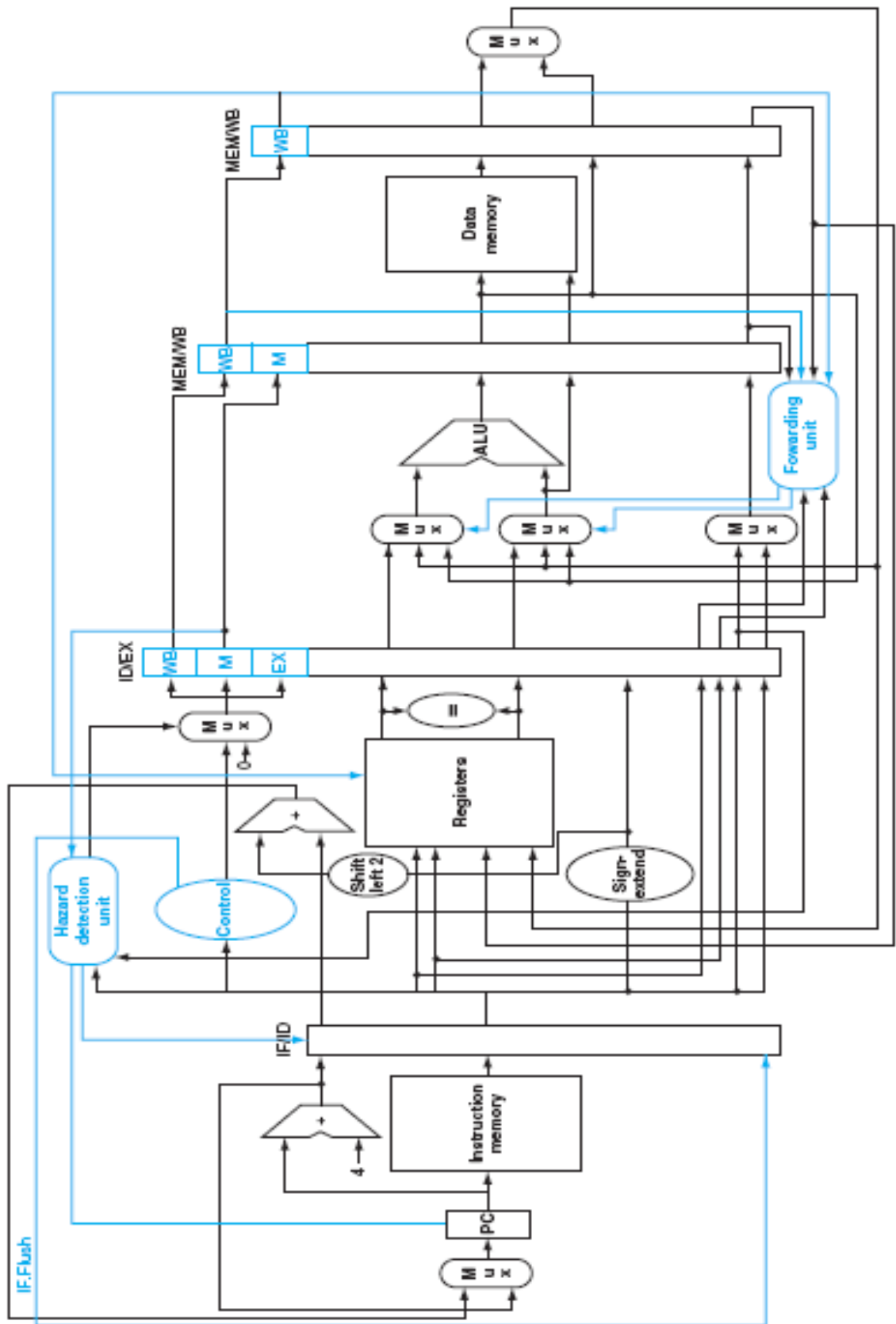


Figura 3

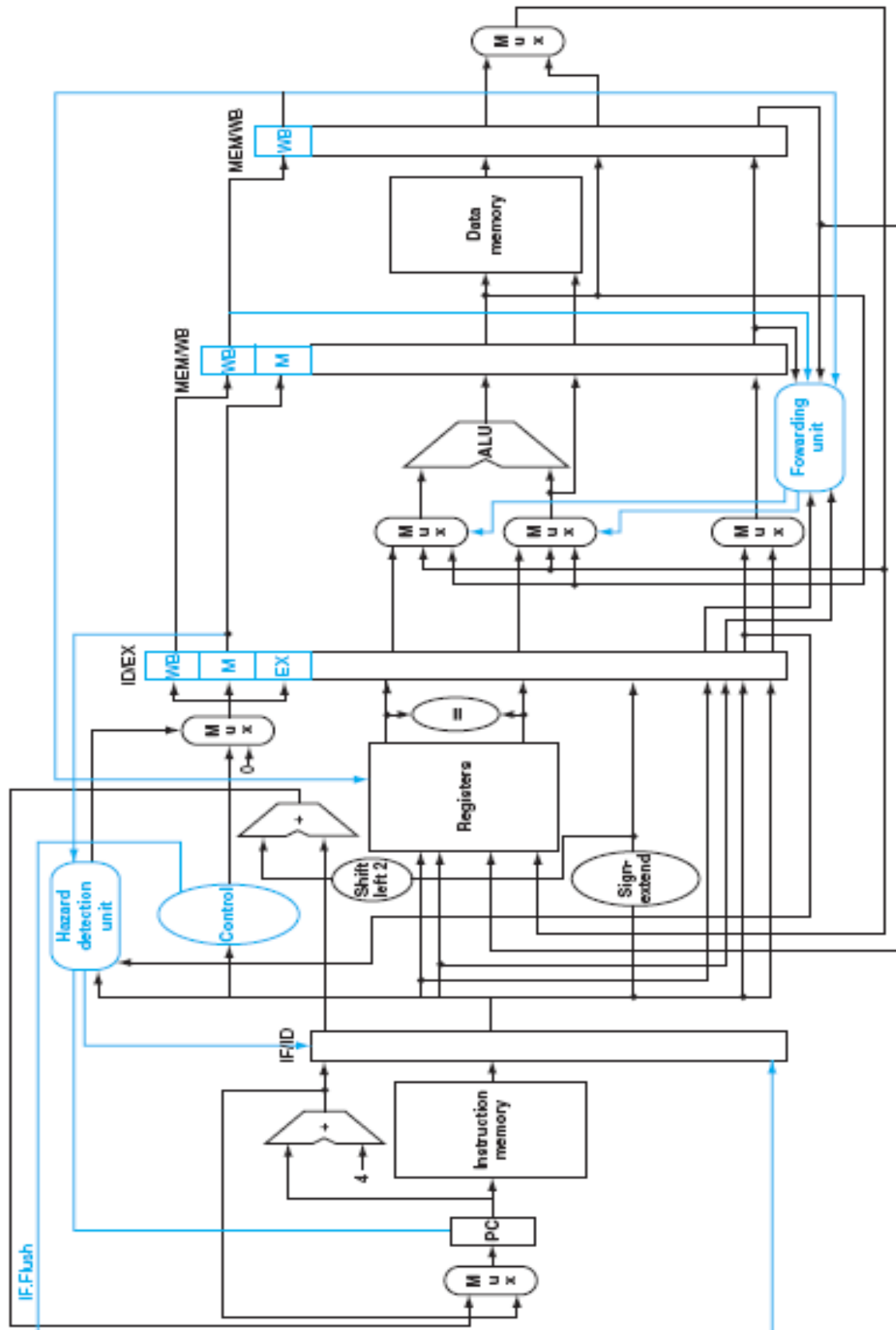


Figura 4

