

Cognome e nome dello studente:

Matricola:

Numero pagine:

1. [6] 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 [4]:

```
0x00000400 ori $s5, $t2, 100
0x00000404 or $t4, $t1, $t1
0x00000408 add $t1, $t2, $s5
0x0000040c sw $s1, 8($s0)
0x00000410 lw $s2, 8($s0)
```

quando l'istruzione di ori 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 in quello stadio [1]. Si possono intravedere problemi di coerenze e consistenza della memoria? Perché? [1].

2. [2] Cosa si intende per politica di gestione della cache in Write Back e in Write through? Cosa si intende per Write Invalidate?

3. [2] Definire quali sono gli hazard che si possono identificare in una pipeline e quali le soluzioni per mitigarli.

4. [1] Definire quali sono i tipi di miss della cache e quali le possibili soluzioni per ridurre l'impatto.

5. [2] Quali sono le modalità di gestione di un I/O? Quali sono i vantaggi e gli svantaggi di ciascuna modalità?

6. [1] Cosa si intende per weak scaling e strong scaling? A cosa si applica?

7. [2] In figura 2 è rappresentato uno schema semplificato di una CPU ARM Cortex A53. Quali sono le caratteristiche? Viene utilizzato lo scheduler? Perché?

8. [1] Cos'è il blocking? A cosa serve?

9. [2] Cosa rappresenta il "roof model"? Cosa rappresenta l'intensità aritmetica? Si riferisce ad una CPU o ad un particolare programma? Un programma che elabora matrici sparse sarà un programma con intensità aritmetica alta o bassa? Perché? Quali sono i passi per ottimizzare le prestazioni del codice suggeriti dal roof-model? Cos'è un kernel benchmark? Cos'è lo SPEC?

10. [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 contengono. Modificare la CPU di Figura 1 per potere gestire un'eccezione di "Overflow". Cosa si intende per mascheramento degli interrupt? Viene praticato nei MIPS? Come vengono gestite le eccezioni e gli interrupt dai sistemi operativi sul MIPS? Scrivere uno scheletro di possibile codice.

11. [2] Modificare la CPU di Figura 1 perché possa gestire questo hazard:

```
0x4000 0000 add $t1, $t2, $t3
0x4000 0004 beq $s1,$t1, ind_salto
```

12. [1] Spiegare chiaramente cosa si intende per stallo e illustrare almeno una situazione in cui si verifica e perché.

13. [4] Disegnare una memoria cache (parte dati + TAG + bit di validità – non è necessario disegnare le porte di lettura e scrittura) per un'architettura MIPS a 64 bit, a 4 vie di 64 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, 1024(\$0)? Da quanti bit è costituita questa memoria complessivamente? Cosa succede quando si verifica una miss? Come si può limitare la frequenza di miss? Da quale dei quattro banchi viene scaricato il dato quando occorre caricare una nuova linea? Perché?

14. [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?

15. [1] Riportare alcune caratteristiche della architetture x86.

### 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)

Figure 1

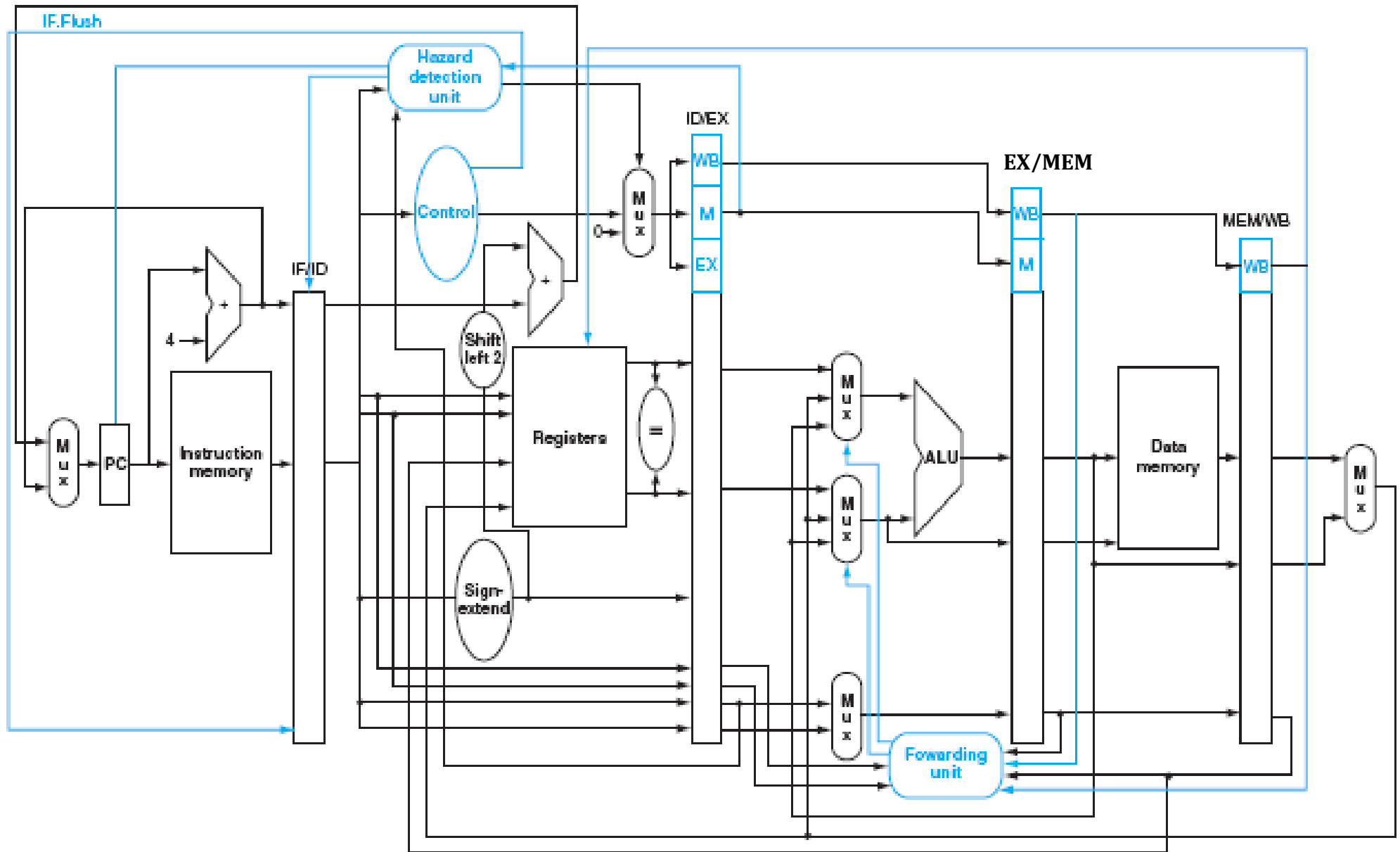


Figure 1

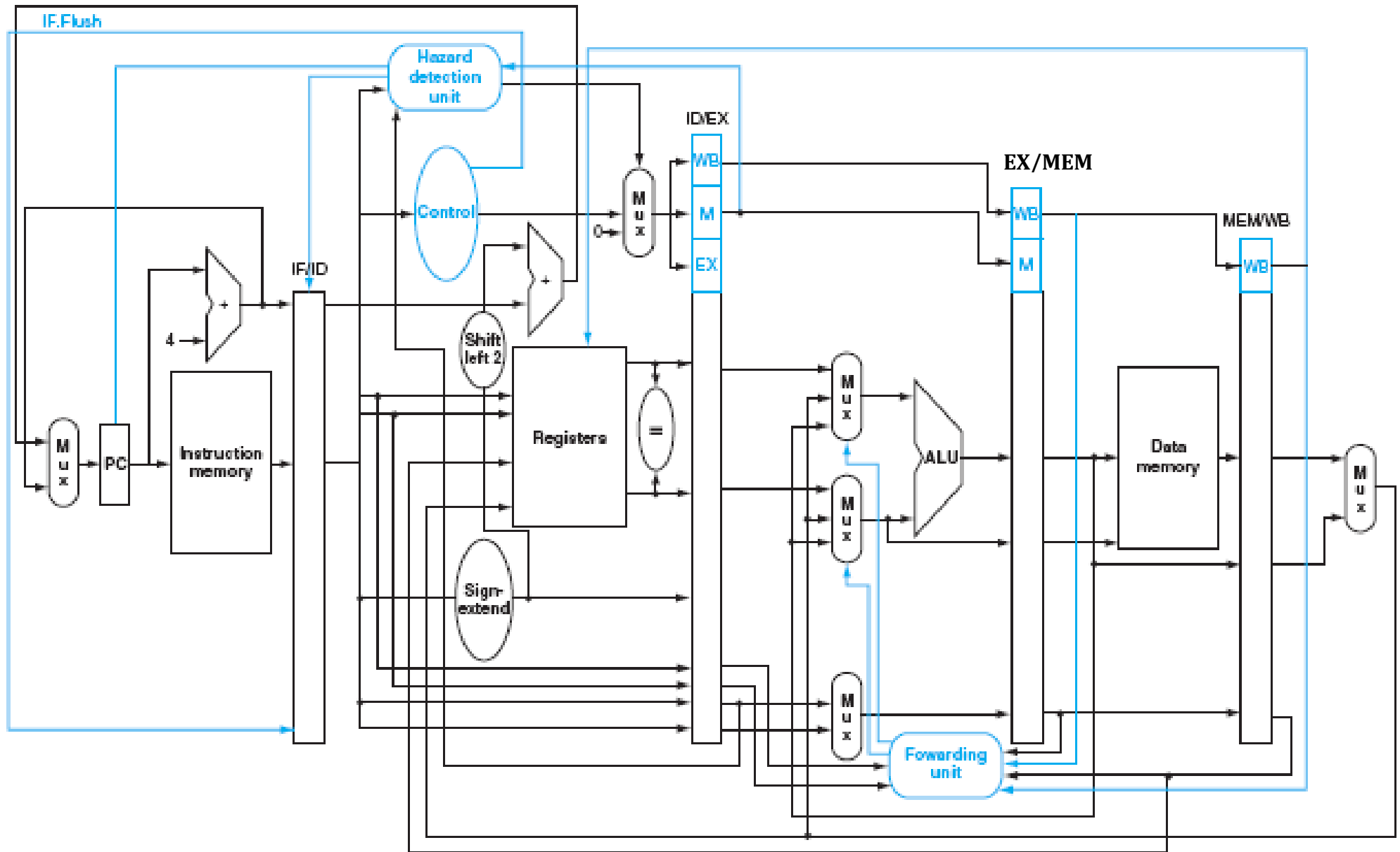


Figura 2

