

Università degli Studi di Milano - Corso Architettura degli elaboratori e delle reti II – Prof. Borghese – Appello del 25.06.2013

Cognome e nome:

Matricola:

1. [2] Quali sono i segmenti della memoria definiti nella architettura MIPS? Quali sono le istruzioni che leggono dalla memoria dati e quali istruzioni leggono dai segmenti dati e quali dai segmenti testo?

2. [7] Data la CPU N. 1, specificare il contenuto di **tutte** le linee (dati e controllo) che trasportano segnali utili, quando è in esecuzione il seguente segmento di codice [4]:

```
0x00000400 and $s5, $t2, $t1
0x00000404 sw $s1, 8($s0)
0x00000408 or $t4, $s5, $s1
0x0000040C addi $t1, $t2, 100
0x00000410 lw $s2, 32($s0)
0x00000414 sub $s2, $s0, $s2
```

quando l'istruzione di and si trova in fase di WB. Modificare la CPU in modo che gestisca correttamente la propagazione anche per le istruzioni di addi [3].

3. [2] Cos'è un hazard? Quali tipi di hazard vengono identificati? Cos'è uno stallo? Stallo e bolla sono la stessa cosa? Si verificano hazard nell'esecuzione del codice precedente? Motivare la risposta.

4. [4] Cosa si intende per Superpipeline e pipeline multiple issue? Cosa sono gli "issue"? Chi li crea? Spiegare i motivi che hanno spinto lo sviluppo delle pipeline in queste direzioni. Cosa si intende per parallelismo implicito ed esplicito? Cos'è un cluster? Quali sono i punti deboli e forti di: a) cluster, b) architetture multi-core.

5. [3] Descrivere come funziona una pipeline multiple-issue statica. Cosa rappresenta una VLIW? Fare un esempio. Trasformare il codice dell'esercizio 2 in un codice che può essere eseguito in modo efficiente da una pipeline multiple issue statica a due vie, 1 per le operazioni di memoria ed 1 per le operazioni sui dati. Quanto si guadagna in tempo di esecuzione complessivo? Qual'è lo speed-up su questo codice? Può essere generalizzato?

6. [3] Descrivere come funziona una pipeline multiple-issue dinamica. Cosa si intende per: a) "Reservation station", "Register renaming", "Commit unit", "Speculazione", "Flush"?

7. [4] Cosa si intende per consistenza di una memoria? Cosa si intende per coerenza (di una memoria)? Come si può garantire la coerenza della memoria? Quali sono le tecniche utilizzate nelle architetture mono e multi-processore? Quali sono i problemi? Cos'è il lock? Cos'è e come funziona un codice di errore?

8. [5] Cosa sono gli interrupt e le eccezioni? Come vengono gestite dai sistemi operativi? Specificare gli elementi della CPU che sono dedicati alla gestione delle eccezioni e supportano il sistema operativo nel MIPS. Modificare la CPU sopra per potere gestire un'eccezione di "Overflow". Cosa si intende per mascheramento degli interrupt? Viene praticato nei MIPS? Scrivere lo scheletro di un programma di risposta alle eccezioni.

9a. (domanda per gli studenti che hanno dato Architettura I nell'AA 2011-2012) [6] Cosa rappresenta il "roof model"? Cosa rappresenta l'intensità aritmetica? Si riferisce ad una CPU o ad un particolare programma? Data una CPU quad-core, a 64 bit (dati float su 64 bit) con 4 cammini di calcolo per ciascun core, in ciascuno dei quali vengono elaborati dati a 128 bit. Detta CPU ha un clock di 4GHz. A questa CPU è associato un sistema di memoria che è in grado di sostenere l'elaborazione con un flusso dati dalla memoria alla CPU pari a 16Gbyte /s. Determinare la massima velocità di elaborazione della CPU per 6 diversi programmi benchmark che hanno intensità aritmetica rispettivamente di: $\frac{1}{4}$, $\frac{1}{2}$, 1, 4, 8, 32 e determinare se le prestazioni sono limitate dalla memoria o dal calcolo.

9b. (domanda per tutti gli altri studenti) [6]. Data un cache a 4 vie, ciascuna via memorizza 8 Kbyte, da dove viene caricato il dato dall'istruzione lw \$t0, 2844(\$0)? Supponete che i dati siano su 32 bit e che la cache abbia linee di 256 bit. Disegnare la porta di lettura. Cosa si intende per lettura in modalità "burst"? Perché è importante?

10. [1] Cos'è lo SPEC? Cosa misura 1 FLOPS? Quali sono i passi suggeriti per ottimizzare il codice su un'architettura parallela?

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)

CPU N. 1

