

Nome e Cognome:

Matricola:

Corso di Architettura degli Elaboratori

Anno Accademico 2008/2009

Esempio di compito della seconda parte

Istruzioni

- Scrivere *Nome, Cognome e Matricola* su **ogni** foglio (solo pagine **dispari**).
- Scrivere la risposta nello spazio bianco al di sotto della domanda; Non è possibile allegare fogli aggiuntivi, quindi cercate di essere chiari e non prolissi.
- In caso di errori indicate chiaramente quale parte della risposta deve essere considerata; annullate le parti non pertinenti.
- Assicurarsi che non manchi alcun foglio al momento della consegna.

Esercizi a risposta multipla

es1

Fra le informazioni riportate di seguito dire quale non è comunicata dalla CPU al dispositivo DMA

- a) se l'operazione è di lettura
- b) se l'operazione è di scrittura
- c) quantità di dati da trasferire
- d) codice di interruzione
- e) indirizzo iniziale in memoria del blocco dati coinvolto nell'operazione
- f) tutte le informazioni di sopra sono comunicate;

es2

Si consideri una cache set-associativa a 4 vie (4-way) da 16MB. La cache è inserita in una gerarchia di memoria insieme ad una memoria centrale suddivisa in 524288 blocchi e di 256MB. Il formato degli indirizzi della memoria centrale è:

- a) Etichetta = 11 bit; Set = 10 bit; Parola = 7;
- b) Etichetta = 7 bit; Set = 11 bit; Parola = 10;
- c) Etichetta = 9 bit; Set = 10 bit; Parola = 9;
- d) Etichetta = 6 bit; Set = 13 bit; Parola = 9;
- e) nessuna delle risposte precedenti è corretta;

es3

Si consideri una cache di 256KB con associazione diretta in congiunzione con una memoria centrale di 256MB, indirizzabile al byte. Le locazioni di memoria con indirizzi (in esadecimale) 533ACFD e 90FAF1F hanno la possibilità di essere caricate all'interno della stessa linee se la dimensione del blocco è di:

- a) 1KB;
- b) 128B;
- c) 512B;
- d) 256B;
- e) nessuna delle risposte precedenti è corretta;

es4

Una architettura che utilizza bus multipli

- a) permette la facile integrazione della tecnica memory-mapped I/O;
- b) permette in alcuni casi di ridurre il numero di cicli di clock necessari per eseguire una istruzione;
- c) è indispensabile per una integrazione efficiente della tecnica memory-mapped I/O;
- d) nessuna delle risposte precedenti è corretta;

Esercizi a risposta libera

es5

Dire quali sono le motivazioni sottostanti al RAID e spiegarlo in dettaglio

es6

Spiegare in dettaglio in cosa consiste e come avviene l'ottimizzazione dell'uso dei registri tramite compilatore.

es7

Nel contesto di una pipeline spiegare in dettaglio in cosa consiste e come funziona la tecnica del salto ritardato (delayed branch). Fare un esempio di applicazione, spiegandolo in dettaglio.

es8

Spiegare in dettaglio a cosa serve e come funziona il codice a correzione di errore di Hamming.

Esercizi pratici

es9

Considerando la pipeline MIPS vista a lezione, si consideri il seguente frammento di codice:

```
SW      R2, 108(R5)  ! mem[108+[R5]] = [R2]
LW      R5, 10(R2)   ! R5 = mem[10+[R2]]
ADDI    R2, R5, 1    ! R2 = [R5] + 1
SW      R8, 20(R5)   ! mem[20+[R5]] = [R8]
ADDI    R2, R2, 4    ! R2 = [R2] + 4
SUB     R4, R8, R2    ! R4 = [R8] - [R2]
SW      R8, 10(R2)   ! mem[10+[R2]] = [R8]
```

- a) dire quali dipendenze esistono e quali potrebbero essere risolte usando i tipi di forwarding visti a lezione per la pipeline MIPS;
- b) assumendo che sia possibile usare i tipi di forwarding visti a lezione, mostrare come evolve la pipeline durante l'esecuzione del codice.

Nome e Cognome:

Matricola:

Pagina 9

es10

Si consideri il seguente frammento di (pseudo)codice assembler, dove le istruzioni SHW e LHW si riferiscono alla mezza parola (2 byte):

```

-----
                ! codice ! semantica
codice mnemonico ! macchina ! istruzione
                ! (hex)  !
-----
SHW   R2, 100(R0) ! 2BF0 ! mem[100+[R0]] = [R2]
LHW   R5, 102(R2) ! 01F2 ! R5 = mem[102+[R2]]
ADDI  R2, R5, 2   ! 8211 ! R2 = [R5] + 2
SHW   R5, 100(R2) ! 2BA2 ! mem[100+[R2]] = [R5]
ADDI  R2, R2, 2   ! 8224 ! R2 = [R2] + 2
SUB   R8, R2, R5  ! 2432 ! R8 = [R2] - [R5]
SHW   R8, 100(R2) ! 2BF2 ! mem[100+[R2]] = [R8]
-----

```

caricato in memoria a partire dall'indirizzo (esadecimale) 10C. Si assuma la presenza di due cache, una dati ed una istruzioni, entrambe di ampiezza 16B, con dimensione di blocco 4B, inizialmente vuote, ad associazione a 2-vie, con politica di rimpiazzo LRU e politica di scrittura write-through. Si assuma che la memoria abbia il contenuto esadecimale mostrato di seguito (si esprimano gli indirizzi su 12 bit):

Indirizzo	byte	byte	byte	byte
100	0F	00	00	02
104	00	00	00	00
108	AE	13	A1	23
10C	2B	F0	01	F2
110	82	11	2B	A2
114	82	24	24	32
118	2B	F2	14	32

e che i registri (tutti da 16 bit) contengano inizialmente il valore 0. Si mostri come sia il contenuto della cache dati che il contenuto della memoria cambia a causa della esecuzione del codice assembler.

