

Calcolatori Elettronici – A.A. 2005/2006

Seconda Prova in Itinere

15 giugno 2006

- La durata della prova è di 120 minuti.
- *Riportare nell'intestazione di ogni foglio (a stampatello) i seguenti dati: cognome, nome, matricola, codice del compito.*
- *Includere la scheda dei risultati debitamente compilata tra i fogli consegnati.*

Obblighi, divieti e suggerimenti

- *Tutti i fogli utilizzati (compreso questo) devono essere consegnati.*
- *Per richieste di fogli aggiuntivi, incomprendimenti del testo rivolgersi esclusivamente al docente.*
- *E' vietato consultare libri, eserciziari, appunti ecc.. Chiunque verrà trovato in possesso di documentazione relativa al corso vedrà annullato il proprio compito.*
- *Qualsiasi tipo di interazione con un collega determinerà l'annullamento del compito (di entrambi) o la penalizzazione fino al 50% del punteggio ottenuto.*
- *Scrivere in modo chiaro e ordinato.*
- *La copia di brutta deve essere distinguibile dalla copia di bella.*

Procedura per la consegna del compito

1. *Attirare l'attenzione del docente con un cenno della mano (senza proferire parola).*
2. *Il docente vi autorizzerà ad alzarvi dal posto e a consegnare il compito.*
3. *Uscire dall'aula (non sarà più consentito ritornare al proprio posto fino alla fine dell'esame).*

Esercizio 1 [30%]

Tradurre, in Assembly DLX, il seguente sorgente C che calcola la distanza di Hamming tra due numeri immessi da tastiera.

```
main()
{
    int a, b, hamming = 0;

    a = InputUnsigned("Inserisci il primo numero: ");
    b = InputUnsigned("Inserisci il secondo numero: ");

    printf("La distanza di Hamming tra %d e %d e' ", a, b);

    while (a != 0 || b != 0) {
        if ( (a & 1) != (b & 1) )
            hamming++;
        a = a >> 1; /* shift logico a destra di 1 bit */
        b = b >> 1;
    }

    printf("%d\n", hamming);
}
```

Esercizio 2 [20%]

Realizzare il circuito di indirizzamento gerarchico (decodifica completa degli indirizzi) per una memoria fisica di 42KB così organizzata:

1. Un banco di memoria da 16KB da mappare dall'indirizzo 0 all'indirizzo 16K-1
2. Due banchi di memoria da 8KB da mappare dall'indirizzo 16K all'indirizzo 32K-1
3. Due banchi di memoria da 4KB da mappare dall'indirizzo 32K all'indirizzo 40K-1
4. Due banchi di memoria da 1KB da mappare dall'indirizzo 40K all'indirizzo 42K-1

Gli indirizzi emessi dal processore sono a 16 bit.

Esercizio 3 [25%]

Rispondere ai seguenti quesiti.

1. Si consideri un processore capace di indirizzare 4GB di memoria (indirizzi a 32 bit). Quale deve essere la dimensione dei campi *tag*, *index*, e *offset* se si considera una cache di 8KB, set associativa a 4 vie, con blocchi di 32 bytes.
2. A parità di dimensione della cache e di dimensione dei blocchi come varia la dimensione del campo *tag* all'aumentare dell'associatività? (Rispondere aumenta o diminuisce e giustificare la risposta in non più di 5 righe).
3. Si consideri una cache direct mapped di 4KB, con blocchi formati da 1 word (1 word è 4 byte). Se il processore esegue le seguenti istruzioni:

```
lw    r1, 0(r0)
lw    r2, 1024(r0)
lw    r3, 1026(r0)
lw    r4, 0(r0)
lw    r5, 3(r0)
```

Determinare quali di queste hanno dato luogo a hit, quali a miss e calcolare l'hit rate e il miss rate (NB: gli indirizzi sono indirizzi di byte).

4. Si consideri il seguente frammento di codice:

```
addi  r1, r0, 0x8000
addui r2, r0, 0x8000
sub   r3, r1, r2
```

Quale sarà il contenuto di r3?

Esercizio 4 [25%]

A seguito dell'esecuzione di un programma, sono state raccolte le seguenti statistiche:

Tipo	Frequenza	Cicli
Load/Store	30%	16
ALU	45%	1
FP	10%	8
Salto	15%	3

Il numero totale di istruzioni eseguite è 450000. La frequenza di clock del processore è di 300 MHz.

1. Determinare il CPI.
2. Determinare il tempo di CPU.
3. A seguito dell'utilizzo di una memoria cache è possibile eseguire le istruzioni di *store* sempre in un ciclo di clock e le istruzioni di *load* in un solo ciclo di clock solo in caso di hit. Determinare il massimo valore di miss rate che consenta di ottenere uno speedup di 2. (Si consideri che la metà degli accessi in memoria siano per load)