

Calcolatori Elettronici – A.A. 2007/2008

Seconda Prova in Itinere

12 giugno 2008

- 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.*
- Obblighi, divieti e suggerimenti**
- *Per richieste di fogli aggiuntivi, incomprensioni 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. E' consentito l'utilizzo dell'IS manual del DLX.*
 - *Qualsiasi tipo di interazione con un collega determinerà l'annullamento del compito (di entrambi).*
 - *Scrivere in modo chiaro e ordinato.*
- 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 [70%]

Tradurre in Assembly DLX il seguente codice C. E' possibile utilizzare la funzione InputUnsigned discussa a lezione.

```
main() {
    int vettore[20];
    int i, min, max, trovato;

    // Inserimento
    for (i=0; i<20; i++)
        vettore[i] = InputUnsigned( Inserisci un elemento: );
    min = InputUnsigned( Inserisci minimo: );
    max = InputUnsigned( Inserisci massimo: );

    // Elaborazione
    i = trovato = 0;
    while (i < 20 && !trovato) {
        if (vettore[i] >= min && vettore[i] <= max)
            trovato = 1;
        else
            i++;
    }

    // Visualizzazione
    if (trovato)
        printf( Il primo elemento tra %d e %d è stato trovato in posizione %d , min, max, I);
    else
        printf( Nessun elemento trovato );
}
```

Esercizio 2 [30%]

Si consideri una memoria cache direct mapped, di 4KB, con blocchi di 4 word. Determinare:

1. La dimensione del campo offset, del campo index e del campo tag. **[30%]**
2. Considerando la cache inizialmente vuota (i valid bit di ogni blocco a zero), determinare il miss rate per a seguito dell'esecuzione del seguente frammento di codice: **[70%]**

```
lw r1, 65(r0)
lw r2, 69(r0)
lw r3, 3(r0)
lw r4, 81(r0)
lw r5, 0(r0)
lw r6, 512(r0)
lw r7, 4(r0)
```