

## Architetture degli Elaboratori II

### I Compito di Esonero - 15/5/2003

Cognome, Nome:

Matricola :

Ogni esercizio riporta fra parentesi il suo valore in trentesimi (punteggio totale 33/30).

1. (punti: 6)

Descrivere dettagliatamente le funzioni del bit JMPC nella microistruzione MIC-1. Scrivere due microistruzioni MAL, con la loro rispettiva codifica binaria, che illustrino gli usi prevalenti del bit JMPC.

2. (punti: 11)

Siano date due variabili locali intere A e B. Dovete fare la somma dei numeri interi (estremi inclusi) compresi fra A e B, e depositare la somma sullo stack.

(a) Realizzate quanto richiesto come metodo in linguaggio IJVM.

(b) Scrivete lo spezzone di programma che chiama il metodo scritto al punto precedente con  $A=10$  e  $B=3$ .

(c) Trascrivete accanto ad ogni istruzione IJVM la codifica binaria (o esadecimale).

(d) Supponendo che la macchina MIC-1 su cui eseguite il vostro programma sia dotata di un clock a  $500\text{ MHz}$ , spiegate come potete calcolare il tempo che la macchina impiega a eseguire il metodo che avete scritto al punto a)?

3. (punti 8)

Sia dato il sottostante spezzone di metodo in linguaggio IJVM, che sapete iniziare alla cella C0 dell'area metodi della memoria. Dovete:

(a) Disegnare il frame di attivazione del metodo;

(b) trascrivere in modo esatto il contenuto delle celle di memoria dell'area metodi in binario a partire dalla cella C0.

```
.method crti (a)
.var
cont
anew
espo
.end-var
```

```

    bipush (-5)           //numero decimale
llb:  istore cont
      iload a
      iflt llb
      .....

```

4. (punti: 8)

Nel linguaggio IJVM l'istruzione (`iinc varnum const`) utilizza due operandi (`varnum`, `const`) a 8 bit. Si vuole estendere la capacità del linguaggio IJVM introducendo una variante della istruzione `iinc` che utilizza due operandi a 16 bit. Per ottenere questa estensione si possono seguire due alternative:

- (a) Definire una nuova istruzione IJVM con un nuovo codice operativo, ad esempio (`iinc2 var2 const2`), dove `var2` e `const2` sono due variabili a 16 bit.
- (b) Utilizzare il prefisso `wide` che applicato a (`iinc varnum const`) fa sì che si debbano leggere due variabili a 16 bit.

Disegnare la struttura nell'area metodi della nuova istruzione IJVM nei due casi (cioè i byte occupati) e scrivere il nuovo microinterprete MIC-1 nei due casi.