

Architettura degli Elaboratori

Seconda Verifica Intermedia 18/12/2015

*Indicare su tutti i fogli consegnati Nome, Cognome, Matricola, Corso (A o B).
I risultati saranno pubblicati sulle pagine web didawiki o dei docenti, appena disponibili.*

Si consideri il seguente frammento di codice ad alto livello

```
int a[N], b[N], c1, c2;
for(i=0; i<N; i++) {
    a[i]=a[i]*b[i]+c1+c2;
    c1=c1+1;
    c2=c2+2;
    b[i]=c1+c2;
}
```

Il risultato della compilazione è eseguito su un processore D-RISC pipeline con supporto per il salto ritardato ed EU parallela pipeline che esegue le operazioni brevi in t e quelle lunghe in $3t$. Le pagine della cache sono di 64 parole ($\sigma = 64$).

Si chiede di

- scrivere il codice prodotto dalla compilazione utilizzando le regole di compilazione standard
- calcolare il numero di fault generati dall'esecuzione del frammento di codice considerando il caso in cui la dimensione della cache permetta di mantenere in memoria 2 pagine per il codice e 2 pagine per i dati
- ottimizzare il codice D-RISC e calcolare il guadagno di prestazioni ottenuto
- calcolare il tempo di esecuzione del codice ottenuto mediante compilazione con regole standard supponendo che oltre alle caratteristiche precedenti l'architettura sia anche multithreading con interleaving di grado 2 ed il secondo thread sia una sequenza di istruzioni operative corte e che non contiene dipendenze logiche

Traccia di soluzione

La compilazione standard del codice produrrà il seguente codice assembler D-RISC:

```
loop:  LOAD RbaseA, Ri, R1
        LOAD RbaseB, Ri, R2
        MUL R1, R2, R3
        ADD R3, Rc1, R3
        ADD R3, Rc2, R3
        STORE RbaseA, Ri, R3
        ADD Rc1, #1, Rc1
        ADD Rc2, #2, Rc2
        ADD Rc1, Rc2, R4
        STORE RbaseB, Ri, R4
        INC Ri
        IF< Ri, Rn, loop
```