

Architettura degli elaboratori

Appello 3 settembre 2019 – A.A. 2018—19

*Riportare in alto a sinistra di ognuno dei fogli consegnati Nome, Cognome, Matricola e Corso (A o B).
I risultati e la correzione saranno resi noti su didawiki.di.unipi.it appena disponibili, insieme al calendario degli orali.*

Domanda 1

Si consideri il microcodice della fase firmware per la gestione delle interruzioni come riportato nel libro di testo e se ne fornisca una versione modificata per l'utilizzo di uno stack per il salvataggio delle informazioni che nella versione originale vengono invece salvate nei registri generali. In particolare, si assuma che in questo caso il registro stack pointer (R_{61}) punti sempre alla prima posizione occupata in testa allo stack e che i registri R_{62} ed R_{63} contengano rispettivamente il limite inferiore (utilizzato per controllare se lo stack è vuoto) e superiore (utilizzato per controllare se lo stack è pieno) dell'area di memoria che contiene lo stack e che lo stack cresca da indirizzi bassi verso indirizzi alti.

Successivamente si fornisca il codice della fase assembler del trattamento delle interruzioni, eventualmente modificato per tener conto del diverso metodo di passaggio dei parametri e si spieghi se la modifica è sufficiente a permettere il trattamento di interruzioni annidate. Nelle modifiche apportate al microcodice della fase firmware si tenga come obiettivo il fatto che la nuova versione del trattamento delle interruzioni non comporti un aumento del tempo del ciclo di clock.

Domanda 2

Si consideri lo pseudocodice

```
for(int i=0; i<N; i++)
  for(int j=i; j<N; j++) {
    a[j] = b[i]+c[j];
    c[j]++;
  }
```

e se ne forniscano:

- compilazione in codice D-RISC secondo le regole di compilazione classica
- compilazione ottimizzata del loop centrale considerando (anche) la tecnica del loop unrolling con fattore di unrolling pari a k .
- valutazione delle prestazioni, nei due casi, evidenziando gli effetti del loop unrolling.

Per la valutazione delle prestazioni si consideri un processore D-RISC pipeline come quello del libro di testo, senza unità di esecuzione slave, nel quale le addizioni e sottrazioni intere sono svolte dalla EU master.

Domanda 3

Si consideri un processore D-RISC pipeline, superscalare a due vie, dotato della sola EU master (ovvero senza unità EU slave). Si assuma di dover eseguire un certo codice assembler, che esegue su un processore D-RISC pipeline, non superscalare e sempre dotato della sola EU master in un tempo pari a kt . Si dica quale potrebbe essere lo speedup ($speedup = T_{pipe} / T_{superscalare}$) massimo e minimo e si forniscano tre esempi di pseudo codice e relativo codice assembler che ottengono rispettivamente uno speedup minimo, uno massimo e uno pari ad un valore intermedio fra il minimo e il massimo.