

## Architettura degli Elaboratori

Appello del 10 settembre 2012

Riportare su tutti i fogli consegnati nome, cognome, numero di matricola, corso A/B, e la sigla NEW (per nuovo ordinamento), oppure OLD-0 (per vecchio ordinamento, nuovo programma), oppure OLD-1 (per vecchio ordinamento, vecchio programma). I risultati verranno pubblicati sulle pagine web del corso/dei docenti appena disponibili.

### Domanda 1 (tutti)

a) Una unità di elaborazione contiene una memoria M a singolo indirizzamento. In ingresso ha un codice operativo OP (1 bit) e due indirizzi IND1, IND2 di M. Dire se il seguente microprogramma può essere implementato correttamente, spiegando formalmente la risposta:

0. (RDY, OP, zero(M[IND1]), segno(M[IND2]) = 0 - - - ) ..., 0;  
( = 1 0 0 - ) ..., 0; ( = 1 0 1 - ) ..., 0;  
( = 1 1 - 0 ) ..., 0; ( = 1 1 - 1 ) ..., 0

b) In generale, dimostrare che la formula usata per valutare il ciclo di clock fornisce un upper bound e riportare un esempio in cui un valore inferiore è corretto.

### Domanda 2 (tutti)

a) Dare la definizione di insieme di lavoro per una gerarchia di memoria M2-M1.

b) Spiegare che, per uno stesso programma, l'insieme di lavoro è diverso se riferito alla gerarchia memoria virtuale – memoria principale o alla gerarchia memoria principale – cache. Supportare la spiegazione con un esempio di programma. Nota: la differenza prescinde dalla diversa capacità e dimensione delle pagine/blocchi nelle due gerarchie.

### Domanda 3 (NEW, OLD-0)

La seguente computazione

$int A[N], B[N];$

$\forall i = 0 .. N-1: B[i] = A[i] * A[i]$

viene eseguita da due calcolatori, C1 e C2, aventi CPU pipeline superscalare a due vie in-order FIFO: C1 è una macchina D-RISC, C2 ha set di istruzioni D-RISC arricchito dall'istruzione la cui semantica è

$\forall i = 0 .. N-1: B[i] = A[i] * A[i]$

a) Spiegare l'interprete di tale istruzione di C2.

b) Valutare il tempo di completamento per C1 e C2, tenendo conto che la cache dati è su domanda, associativa e write-through, e assumendo che la cache secondaria on-chip abbia probabilità di fault trascurabile.

### Domanda 3 (OLD-1)

Si consideri una architettura in cui le unità d'ingresso-uscita sono assimilate a processi.

a) Spiegare se, disponendo del supporto a tempo di esecuzione delle primitive di comunicazione per processi interni, questo può essere utilizzato anche per far comunicare processi interni e processi esterni.

b) Nel supporto delle comunicazioni tra processi interni e processi esterni, mostrare tutti i casi in cui vengono utilizzate strutture dati condivise riferite indirettamente, e spiegare in dettaglio come queste sono implementate.

## Sintesi della soluzione

(da completare con spiegazioni ulteriori)

### Commento

TUTTE le domande del compito sono spiegazioni di concetti di base e/o sono fortemente riconducibili a domande delle raccolte di esercizi e quesiti che sono state distribuiti e raccomandati, e in gran parte svolti a lezione:

**Domanda 1:** concetti base su firmware e diversi esercizi

**Domanda 2:** raccolta 3, n. 2

**Domanda 3:** raccolta 2 n. 5 e raccolta 3

**Domanda 3, OLD-1:** concetti base su processi e I/O, raccolta 2.

### Domanda 1 (tutti)

**a)** L'indirizzamento di M avviene attraverso un commutatore con ingressi IND1 e IND2. Sotto la condizione che la parte non specificata del microprogramma non influenzi la risposta, l'espressione logica della variabile di controllo del commutatore è uguale a OP che è una variabile dello stato interno di PO (si potrebbe anche dire che è usata una forma di controllo residuo, anche se ciò non è necessario per rispondere formalmente alla domanda). Quindi, è rispettata la condizione che PO sia una macchina di Moore e il microprogramma può essere implementato correttamente.

**b)** Tutti gli addendi della formula sono i ritardi massimi nell'arco di tutto il microprogramma. È però possibile che tutte le microistruzioni /frasi abbiano un ritardo di stabilizzazione minore, in quanto non caratterizzate da tutti ritardi i messi in assoluto. In tal caso, è corretto un ciclo di clock uguale al massimo tra i ritardi di tutte le microistruzioni. Ad esempio:

$$0. \quad A + B + C \rightarrow A, 1$$

$$1. \quad (\text{zero } (A - D) = 0) A \rightarrow D; (= 1) A \rightarrow E$$

Applicando la formula:  $\tau = 3T_{\text{ALU}} + T_{\text{opC}} + \delta$ , mentre esaminando singolarmente ognuna delle microistruzioni si ha  $\tau = 2T_{\text{ALU}} + T_{\text{opC}} + \delta$  (microistruzione 0) cioè poco più di  $2/3$ .

### Domanda 2 (tutti)

**a)** L'insieme di lavoro è l'insieme delle pagine/blocchi che, se presenti simultaneamente in M1, minimizza la probabilità di fault.

**b)** Nella gerarchia MV-M un processo per essere eseguito deve essere prima essere stato caricato in M. La probabilità di fault è quindi minima (zero) se tutto il processo è caricato in M. Invece, la gerarchia M-C è caratterizzata da un numero minimo di fault necessario per caricare, almeno per la prima volta (se e quando esiste riuso) blocchi da M a C. Qualunque esempio è valido (*mostrare*).

### Domanda 3 (NEW, OLD-0)

**a)** L'implementazione dell'istruzione data in C2 utilizza il pipeline per generare uno stream di  $N$  valori da DM a EU (IU può fare un'unica richiesta oppure  $N$  richieste, con lo stesso tempo di servizio) ed eseguire lo stream di  $N$  moltiplicazioni nella EU parallelo-pipeline. I risultati possono essere inviati a DM via IU, senza apprezzabile degradazione dovuta a dipendenze logiche, in quanto le scritture (tranne l'ultima) sono sovrapposte al calcolo. La latenza dell'istruzione, e quindi il tempo di completamento del programma in C2, è uguale a  $(N-1)t$  (transitorio di riempimento delle EU pipeline) +  $4t$  (esecuzione dell'ultima moltiplicazione) +  $4t$  (transitorio di riempimento della CPU pipeline) +  $2t$  (esecuzione dell'ultima scrittura); in pratica, per  $N$  grande, in assenza di fault di cache:

$$T_{C2-id} \sim Nt = 2N\tau$$

**b)** Per C1, il programma ottimizzato ha tempo di servizio uguale a  $9t/5$  (da spiegare: mostrare il programma e usare il modello dei costi), quindi:

$$T_{C1-id} \sim 9 Nt = 18N\tau$$

Considerando la penalità dovuta ai fault di cache in entrambe le macchine:

$$T_{fault} \sim 2N\tau$$

si ha:

$$T_{C1} \sim 20N\tau \qquad T_{C2} \sim 4N\tau$$

### **Domanda 3 (OLD-1)**

**a)** Il supporto non può essere riusato, in quanto sono diverse almeno la fase di scheduling a basso livello (sveglia) e i meccanismi per l'indivisibilità, sia con DMA che con MMI/O (dettagliare la risposta).

**b)** Le strutture condivise riferite indirettamente sono almeno la variabile targa e il PCB del processo interno. La loro manipolazione richiede l'uso di uno dei metodi opportuni, ad esempio capability (descrivere). In particolare, la sveglia del processo interno richiede una interruzione per trasmettere la capability del PCB del processo stesso.