

**ARCHITETTURE AVANZATE DEI CALCOLATORI A.A. 2007/08**  
**Appello straordinario del 18/11/2008**

**Cognome** \_\_\_\_\_ **Nome** \_\_\_\_\_

**Matricola** \_\_\_\_\_

**Attenzione: nella soluzione degli esercizi, motivare le risposte illustrando il procedimento.**

**Domanda 1 (4 punti)**

Definire le tipologie di protocolli di coerenza della cache nelle architetture multiprocessore a memoria condivisa e descrivere un esempio di protocollo di snooping.

**Domanda 2 (4 punti)**

Descrivere le caratteristiche e discutere le problematiche dei processori multipli-issue statici.

**Domanda 3 (4 punti)**

Spiegare la tecnica di predizione della destinazione del salto.

**Esercizio 1 (4 punti)**

Si consideri il processore MIPS con pipeline ottimizzata (il cui schema è allegato) e la seguente sequenza di codice.

```
lw $1, 20($2)
and $3, $1, $2
or $5, $4, $1
addi $3, $3, 16
```

Si individuino le criticità sui dati e si indichi il valore assunto dai segnali di controllo ForwardA, ForwardB, ALUSrc, PCWrite e IF/ID.Write per i cicli di clock da 2 a 7. Si assuma che il PC sia inizializzato alla prima istruzione della sequenza e che il ciclo di clock 1 sia già stato eseguito (ovvero l'istruzione lw è nello stadio ID). Si usi il simbolo X nel caso in cui il segnale di controllo abbia valore don't care.

**Esercizio 2 (5 punti)**

Si consideri il seguente frammento di codice in linguaggio C:

```
int A[1026], B[1026];
register int i;
for (i=0; i<500; i++)
    B[i] = A[i%5];
```

Si assuma che la variabile i sia contenuta in un registro, mentre gli array A e B siano allocati in memoria consecutivamente, prima A e poi B, a partire dall'indirizzo di memoria 0. Si consideri una cache dati ad indirizzamento diretto contenente 256 blocchi, ciascuno dei quali di dimensione pari a 16 B, e che opera in modalità write-back. La cache è inizialmente vuota.

Determinare il numero di hit e miss in lettura e scrittura ed il numero di write-back che avvengono durante l'esecuzione del frammento di codice.

*(Nel linguaggio C l'operatore aritmetico % restituisce il resto di una divisione intera.)*

**Esercizio 3 (5 punti)**

Si consideri un sistema di memoria virtuale con paginazione, con un numero massimo di pagine fisiche pari a 16384, pagine fisiche di dimensione pari a 1 KB e tabella delle pagine per ciascun processo di dimensione totale pari a 2 MB (compresi anche 2 bit aggiuntivi per validità e riferimento per ogni elemento della tabella delle pagine).

- a) Determinare la dimensione e la struttura dell'indirizzo virtuale.
- b) Determinare la dimensione dello spazio di indirizzamento virtuale e dello spazio di indirizzamento fisico.
- c) Il sistema di memoria virtuale ha un TLB set-associativo a 4 vie con un totale di 256 elementi; determinare la dimensione di ciascun elemento del TLB.

#### Esercizio 4 (6 punti)

Sia dato il seguente ciclo di un programma in un linguaggio ad alto livello:

```
do {
    B[i] = B[i] - C1;
    if (A[i] < 0) {
        A[i] = A[i] + C2;
        B[i] = B[i] - A[i];
    }
    else {
        A[i] = A[i] + B[i];
    }
    i++;
} while (i != N);
```

Il programma sia stato compilato nel seguente codice in assembler MIPS. Si supponga che i registri \$t7 e \$t6 siano stati inizializzati rispettivamente ai valori 0 e  $N \cdot 4$ . I simboli BASEA, BASEB, C1 e C2 sono costanti a 16 bit, prefissate. Il ciclo di clock del processore vale 0,4 ns.

```
Do:   beq $t7, $t6, End
      lw $t3, BASEB($t7)
      addi $t3, $t3, -C1
      lw $t2, BASEA($t7)
      slt $t1, $t2, $zero      # if ($t2 < $zero) $t1=1; else $t1=0;
      beq $t1, $zero, L1
      addi $t2, $t2, C2
      sw $t2, BASEA($t7)
      sub $t3, $t3, $t2
      j L2
L1:   add $t2, $t2, $t3
      sw $t2, BASEB($t7)
L2:   sw $t3, BASEB($t7)
      addi $t7, $t7, 4
      j Do
End:
```

Si supponga che il ciclo venga iterato  $N$  volte (con  $N$  grande a piacere) tramite una pipeline MIPS a 5 stadi. Si consideri una generica iterazione del ciclo e si ipotizzi che nel 40% dei casi risulti ( $A[i] < 0$ ). Si consideri l'istruzione  $j$  come un'istruzione di branch (beq o bne) la cui condizione viene sempre valutata vera.

- a) Si supponga che la pipeline sia priva di ottimizzazioni.
- Nel diagramma a ciclo multiplo della pipeline individuare le criticità sui dati di tipo RAW e le criticità sul controllo ed indicare il tipo di criticità.
  - Indicare il numero di stalli necessari a risolvere le criticità individuate.
  - Calcolare il numero totale di stalli inseriti, il CPI asintotico ed il throughput asintotico espresso in MIPS.
- b) Si supponga che nella pipeline siano state introdotte le seguenti ottimizzazioni:
- nel banco dei registri è possibile la lettura e la scrittura nello stesso ciclo di clock;
  - forwarding dei dati;
  - anticipazione del salto nello stadio ID per le istruzioni beq, bne e j.
- Nel diagramma a ciclo multiplo della pipeline individuare le criticità sui dati di tipo RAW e le criticità sul controllo rimaste ed indicare il tipo di criticità; indicare i percorsi di forwarding utilizzati, specificando i registri di pipeline coinvolti.
  - Indicare il numero di stalli necessari a risolvere le criticità rimaste.
  - Calcolare il numero totale di stalli inseriti, il CPI asintotico ed il throughput asintotico espresso in MIPS.