

## CALCOLATORI ELETTRONICI M

### ARCHITETTURA DEI CALCOLATORI ELETTRONICI M

19/12/2022 – Esercizio 1 & 2

**Tempo disponibile: 150 minuti**

#### Esercizio 1 - Descrizione Sistema

Un RV32IMAFD con  $T_{ck} = T$  dispone di **tre** unità funzionali, A, M e D “multiciclo” capaci di eseguire le seguenti istruzioni su operandi in virgola mobile:

**A: FADD (in 2T)      M: FMUL (in 3T)      D: FDIV (in 4T).**

Si consideri il seguente frammento di codice e si faccia l’ipotesi che in T1 si abbia  $Fi=i$  per ogni valore di  $i$  compreso tra 0 e 31.

```
fadd.s f1,f3,f1
fdiv.s f2,f8,f4
fmul.s f4,f3,f5
fmul.s f4,f5,f6
fmul.s f5,f2,f5
fdiv.s f4,f4,f5
```

1. Si stimi il numero di colpi di clock al di sotto del quale non è possibile scendere nell’esecuzione del codice assegnato, qualunque sia il numero di RS, CRB e stadi di Fetch e Decode disponibili, nell’ipotesi che esista una sola unità di tipo **A**, una sola di tipo **D** e una sola di tipo **M** (tre unità funzionali in tutto) e si motivi la risposta (*In questa stima non si tenga conto delle dipendenze nel codice*) (**1 punto**)

2. Si disegni il grafo delle dipendenze e si deduca il numero minimo di periodi di clock necessario a eseguire il codice assegnato tenendo conto solo delle dipendenze trovate (**1 punto**).

3. Si mostri la dinamica dell’esecuzione nel caso della CPU considerata nel punto 1 con 1 CDB, uno stadio di IF e uno di ID, e **2 RS** per ognuna delle tre unità funzionali (*si ipotizzi che, in caso di conflitti sul CDB, la fase di WB dell’unità D abbia la priorità sulle altre*) (**3 punti**).

4. E’ possibile ridurre il numero di periodi di clock necessario ad eseguire il codice assegnato rispetto al valore risultante dalla risposta al punto 3? Si punti al numero minore possibile di periodi di clock e al numero minimo di modifiche, supponendo di poter apportare solo le seguenti modifiche all’architettura:

- o aumento di RS da due a tre in una o più unità funzionali
- o raddoppio o triplicazione del CDB per poter eseguire due o tre fasi di WB per clock
- o aggiunta di una o più unità funzionali con due RS

*In caso di incremento di unità funzionali, si ipotizzi di fare lo scheduling a rotazione sulle due unità funzionali uguali.*

Quali modifiche converrebbe apportare? Si motivi la risposta e si disegni la nuova dinamica di esecuzione. (**3 punti**).

5. Si disegni il film del registro f4 nel caso della dinamica di esecuzione di cui al punto 4. (**2 punti**).

**Esercizio 2 - Descrizione Sistema**

---

Un sistema multicore S è composto da 2 cores RV64GC dotati ciascuno di una cache L1 dati privata a 2 vie da 4 KiB e linee da 16 bytes, gestita con stato MESI e con politica di scrittura Write Allocate in caso di miss di scrittura.

Nel sistema S esegue un'applicazione parallela composta da due tasks, T0 e T1, operanti su tre array A, B e C, contenenti ognuno 1024 elementi in virgola mobile a singola precisione. Il task T0 legge ciascun elemento di A, ne calcola il doppio e scrive il risultato nel vettore B. Il task T1 legge ciascun elemento con indice pari di A, lo moltiplica per 8 e scrive il risultato nel vettore C al medesimo indice. Il task T0 esegue sul core 0 ed il task T1 sul core 1.

```
void T0(float *A, float *B, unsigned int n) {
    int i = 0;
    for (i=0; i<n; i++) {
        B[i] = A[i] * 2;
    }
}
```

```
void T1(float *A, float *C, unsigned int n) {
    int i = 0;
    for (i=0; i<n/2; i++) {
        C[2*i] = A[2*i] * 8;
    }
}
```

Si assuma che i vettori A, B e C siano memorizzati in memoria a partire dagli indirizzi 0x8000\_1000, 0x8000\_3000 e 0x8000\_5000. Si considerino altresì i seguenti codici ASM RISC-V che implementano il body dei due loop di calcolo dei tasks T0 e T1.

T0:

```
; Caricamento dell'indirizzo base di A in x10 e B in x11.
; Caricamento in x12 della dimensione del vettore A.
; Caricamento in x15 dell'indirizzo del primo byte
; successivo all'ultimo elemento di A.
```

```
beqz x12,.L1
slli x12,x12,2
add x15,x10,x12
```

.L3:

```
flw f15,0(x10)
addi x11,x11,4
addi x10,x10,4
fadd.s f15,f15,f15
fsw f15,-4(x11)
bne x15,x10,.L3
```

.L1:

```
ret
```

T1:

```
; Caricamento dell'indirizzo base di A in x10 e C in x11.
; Caricamento dell'immediato 8 in f14.
; Caricamento in x15 dell'indirizzo del primo byte
```

; successivo all'ultimo elemento di A.

```
srli x12,x12,1
beqz x12,.L9
lui x15,%hi(.LC0)
slli x12,x12,3
flw f14,%lo(.LC0)(x15)
```

```

    add    x15,x10,x12
.L11:
    flw    f15,0(x10)
    addi   x11,x11,8
    addi   x10,x10,8
    fmul.s f15,f15,f14 ; Il registro f14 contiene 8
    fsw    f15,-8(x11)
    bne    x15,x10,.L11
.L9:
    ret
.LC0:
    .word  1090519040

```

Quesiti:

- a) Facendo riferimento al sistema S, si completi la mappa della memoria con i tre vettori A, B e C si completi la mappa della memoria con i tre vettori A, B e C.
  - i. Si completi la mappa della memoria con i tre vettori A, B e C.
  - ii. Quanti bit sono necessari per rappresentare il TAG e il Set ID dei vettori A, B e C? Specificare l'indirizzo ed i bit di TAG e Set ID del primo ed ultimo elemento di ognuno dei vettori A, B e C.
  - iii. Quanta memoria occupano complessivamente i tre vettori A, B e C? È possibile che tutti e tre i vettori si trovino contemporaneamente ed interamente in cache nello stesso momento?
- b) Si analizzi la dinamica della cache dati assumendo che il task T1 esegua una iterazione del corpo del loop dopo due iterazioni del corpo del loop del task T0. Si assuma che tutte le linee di cache dei due core siano inizialmente invalide. Si risponda in modo preciso, schematico, conciso e tabellare ai seguenti quesiti:
  - i. Si mostri il contenuto della cache, lo stato MESI e il valore del bit LRU per la prima iterazione del task T0 (i=0).
  - ii. Si mostri il contenuto della cache, lo stato MESI e il valore del bit LRU al termine della sequenza di esecuzione T0 (i=0), T0 (i=1), T1(i=0), T0 (i=2), T0 (i=3), T1 (i=1).
  - iii. Calcolare il numero di accessi, MISS e cicli di WB al termine dell'esecuzione di ciascuna ripetizione dei tasks T0 e T1.
  - iv. Si calcoli la hit e miss rate complessiva al termine delle esecuzioni dei due task. Come cambierebbero le hit e miss rate se le cache L1 fossero directly mapped?