

CALCOLATORI ELETTRONICI M

ARCHITETTURA DEI CALCOLATORI ELETTRONICI M

19/01/2023 – 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 $F_i = i$ per ogni valore di i compreso tra 0 e 31:

```
fmul.s  f1, f4, f5
fadd.s  f2, f3, f2
fadd.s  f3, f6, f4
fadd.s  f1, f1, f5
fmul.s  f1, f2, f3
fdiv.s  f4, f2, 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 M 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:
 - a. aumento di RS da due a tre in una o più unità funzionali
 - b. raddoppio o triplicazione del CDB per poter eseguire due o tre fasi di WB per clock
 - c. 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 f1 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 RV32IMAC 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 due array A, B e C, contenenti ognuno 768 elementi in virgola mobile a singola precisione.

- Il task T0 aggiorna ciascun elemento con indice pari del vettore B moltiplicandolo per la quantità memorizzata al medesimo indice del vettore A ($B[i] = B[i] * A[i]$).
- Il task T1 aggiorna ciascun elemento con indice dispari del vettore B dividendolo per la quantità memorizzata al medesimo indice del vettore A ($B[i] = B[i] / A[i]$).

```
void T0 (float *A, float *B, int n) {
    int i = 0;

    for (i=0; i<n; i+=2) {
        B[i] = B[i] * A[i];
    }
}

void T1 (float *A, float *B, int n) {
    int i = 0;

    for (i=1; i<n; i+=2) {
        B[i] = B[i] / A[i];
    }
}
```

Si assuma che i vettori A e B siano memorizzati in memoria a partire dagli indirizzi 0x8000_1000 e 0x8000_3000. Si considerino altresì i seguenti codici ASM RISC-V che implementano il body dei due loop di calcolo dei tasks T0 e T1.

```
; Per entrambi i task T0 e T1 valgono le seguenti inizializzazioni:
; - x11 contiene l'indirizzo dell'i-esimo elemento del vettore B
; - x10 contiene l'indirizzo dell'i-esimo elemento del vettore A
; - x12 contiene la dimensione dei vettori A e B
; - x15 contiene l'indirizzo del primo byte successivo all'ultimo elemento di B
```

<pre>T0: beqz x12, .L1 addi x15, x12, -1 srli x15, x15, 1 slli x15, x15, 3 addi x14, x11, 8 add x15, x15, x14 .L3: flw fa5, 0(x11) flw fa4, 0(x10) addi x11, x11, 8 addi x10, x10, 8 fmul.s fa5, fa5, fa4 fsw fa5, -8(x11) bne x11, x15, .L3 .L1: ret</pre>	<pre>T1: li x15, 1 bleu x12, x15, .L9 addi x15, x12, -2 srli x15, x15, 1 addi x14, x11, 12 slli x15, x15, 3 addi x11, x11, 4 addi x10, x10, 4 add x15, x15, x14 .L11: flw fa5, 0(x11) flw fa4, 0(x10) addi x11, x11, 8 addi x10, x10, 8 fdiv.s fa5, fa5, fa4 fsw fa5, -8(x11) bne x11, x15, .L11 .L9: ret</pre>
---	--

Quesiti:

- Facendo riferimento al sistema S, si completi la mappa della memoria con i vettori A e B.
 - Indicare l'indirizzo del LSB e MSB del primo e ultimo elemento dei vettori A e B. **(2 punti)**
 - Specificare il numero e il valore dei bit di TAG e Set ID del primo ed ultimo elemento di ognuno dei vettori A e B. **(2 punti)**
 - Quanta memoria occupano complessivamente i vettori A e B? È possibile che tutti e due i vettori si trovino contemporaneamente ed interamente all'interno delle cache dei due core nello stesso momento? **(1 punto)**
- Si analizzi la dinamica della cache dati assumendo che il task T1 esegua una iterazione del corpo del loop dopo una iterazione 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:
 - Si mostri il contenuto della cache, lo stato MESI e il valore del bit LRU al termine della sequenza di esecuzione T0 (i=0), T1 (i=1), T0 (i=2), T1 (i=3). **(7 punti)**
 - Calcolare il numero di accessi, HIT, MISS, HIT rate, MISS rate e cicli di WB al termine dell'esecuzione dei due tasks. **(5 punti)**

Studente	
Nome	
Cognome	
Matricola	

a.i) Si completi la mappa della memoria con i due vettori A e B. (2 punti)		
Dato	Indirizzo	
	LSB	MSB
A[0]	0x8000_1000	0x8000_1003
A[767]	0x8000_1BFC	0x8000_1BFF
B[0]	0x8000_3000	0x8000_3003
B[767]	0x8000_3BFC	0x8000_3BFF

a.ii) Specificare il numero e il valore dei bit di TAG e Set ID del primo ed ultimo elemento di ognuno dei vettori A e B. (2 punti)		
Proprieta	Vettore	
	A	B
# Bit TAG	21	21
# Bit SetID	7	7
TAG [0]	0x10_0002	0x10_0006
SetID [0]	0x00	0x00
Tag [767]	0x10_0003	0x10_0007
SetID [767]	0x3F	0x3F

a.iii) Quanta memoria occupano complessivamente i vettori A e B? È possibile che tutti e due i vettori si trovino contemporaneamente ed interamente in cache nello stesso momento? (1 punto)		
Proprieta	Vettore	
	A	B
# Bytes	3072	3072
RISPOSTA: I due vettori occupano complessivamente una memoria pari a 3072 Bytes. La cache dati ha dimensione 4096 Bytes, quindi i vettori A e B non possono risiedere contemporaneamente in cache.		

Studente	
Nome	
Cognome	
Matricola	

Task 0+1
prima legge
in B, poi
legge in A
e poi scrive
in B.

b.i) Si mostri il contenuto della cache, lo stato MESI e il valore del bit LRU al termine della sequenza di esecuzione T0 (i=0), T1 (i=1), T0 (i=2), T1 (i=3). (7 punti)														
CORE0					EXEC	CORE1					NOTE			
Istruzione: flw fa5,0(x11)						Istruzione: -								
Via0					LRU	Via1								
SetID	TAG	Data	MESI			SetID	TAG	Data	MESI					
0x00	0x10_0006	8[0:3]	E	1	-	-	-	I	0	-	-	-	I	MISS in lettura di B.
-	-	-	-	0	-	-	-	-	0	-	-	-	-	
Istruzione: flw fa4,0(x10)					Istruzione: -									
Via0					LRU	Via1								
SetID	TAG	Data	MESI			SetID	TAG	Data	MESI					
0x00	0x10_0006	8[0:3]	E	0	0x10_0002	A[0:3]	E	E	0	-	-	-	I	MISS in lettura di A.
-	-	-	-	0	-	-	-	-	0	-	-	-	-	
Istruzione: fsw fa5,-8(x11)					Istruzione: -									
Via0					LRU	Via1								
SetID	TAG	Data	MESI			SetID	TAG	Data	MESI					
0x00	0x10_0006	8[0:3]	M	1	0x10_0002	A[0:3]	E	E	0	-	-	-	I	HIT in scrittura di B.
-	-	-	-	0	-	-	-	-	0	-	-	-	-	
Istruzione: -					Istruzione: flw fa5,0(x11)									
Via0					LRU	Via1								
SetID	TAG	Data	MESI			SetID	TAG	Data	MESI					
0x00	0x10_0006	8[0:3]	S	1	0x10_0002	A[0:3]	E	E	0	-	-	-	I	MISS in lettura di B. Core 0 esegue Write-Back.
-	-	-	-	0	-	-	-	-	0	-	-	-	-	
Istruzione: -					Istruzione: flw fa4,0(x10)									
Via0					LRU	Via1								
SetID	TAG	Data	MESI			SetID	TAG	Data	MESI					
0x00	0x10_0006	8[0:3]	S	1	0x10_0002	A[0:3]	S	S	0	0x10_0002	A[0:3]	S	S	MISS in lettura di A.
-	-	-	-	0	-	-	-	-	0	-	-	-	-	
Istruzione: -					Istruzione: fsw fa5,-8(x11)									
Via0					LRU	Via1								
SetID	TAG	Data	MESI			SetID	TAG	Data	MESI					
0x00	0x10_0006	8[0:3]	I	1	0x10_0002	A[0:3]	S	S	0	0x10_0002	A[0:3]	S	S	HIT in scrittura di B.
-	-	-	-	0	-	-	-	-	0	-	-	-	-	
Istruzione: flw fa5,0(x11)					Istruzione: -									
Via0					LRU	Via1								
SetID	TAG	Data	MESI			SetID	TAG	Data	MESI					
0x00	0x10_0006	8[0:3]	I	0	0x10_0006	8[0:3]	S	S	1	0x10_0002	A[0:3]	S	S	MISS in lettura di B. Core 1 esegue Write-Back.
-	-	-	-	0	-	-	-	-	0	-	-	-	-	
Istruzione: flw fa4,0(x10)					Istruzione: -									
Via0					LRU	Via1								
SetID	TAG	Data	MESI			SetID	TAG	Data	MESI					
0x00	0x10_0002	A[0:3]	S	1	0x10_0006	8[0:3]	S	S	1	0x10_0002	A[0:3]	S	S	MISS in lettura di A.
-	-	-	-	0	-	-	-	-	0	-	-	-	-	
Istruzione: fsw fa5,-8(x11)					Istruzione: -									
Via0					LRU	Via1								
SetID	TAG	Data	MESI			SetID	TAG	Data	MESI					
0x00	0x10_0002	A[0:3]	S	0	0x10_0006	8[0:3]	M	M	0	-	-	-	-	HIT in scrittura di B.
-	-	-	-	0	-	-	-	-	0	-	-	-	-	
Istruzione: -					Istruzione: flw fa5,0(x11)									
Via0					LRU	Via1								
SetID	TAG	Data	MESI			SetID	TAG	Data	MESI					
0x00	0x10_0002	A[0:3]	S	0	0x10_0006	8[0:3]	S	S	0	0x10_0006	8[0:3]	S	S	MISS in lettura di B. Core 0 esegue Write-Back.
-	-	-	-	0	-	-	-	-	0	-	-	-	-	
Istruzione: -					Istruzione: flw fa4,0(x10)									
Via0					LRU	Via1								
SetID	TAG	Data	MESI			SetID	TAG	Data	MESI					
0x00	0x10_0002	A[0:3]	S	0	0x10_0006	8[0:3]	S	S	1	0x10_0006	8[0:3]	S	S	MISS in lettura di A.
-	-	-	-	0	-	-	-	-	0	-	-	-	-	
Istruzione: fsw fa5,-8(x11)					Istruzione: -									
Via0					LRU	Via1								
SetID	TAG	Data	MESI			SetID	TAG	Data	MESI					
0x00	0x10_0002	A[0:3]	S	0	0x10_0006	8[0:3]	I	M	0	0x10_0006	8[0:3]	M	M	HIT in scrittura di B.
-	-	-	-	0	-	-	-	-	0	-	-	-	-	

Studente	
Nome	
Cognome	
Matricola	

b.ii) Calcolare il numero di accessi, HIT, MISS, HIT rate, MISS rate e cicli di WB al termine dell'esecuzione dei due tasks. (5 punti)		
Dato	Task	
	T0	T1
# Accessi	1152	1152
# HIT	384	384
# MISS	768	768
# HITrate [%]	33.3%	33.3%
# MISS rate [%]	66.7%	66.7%
# WB	384	383
NOTE : - Ogni task esegue 384 iterazioni, che generano ciascuna 2 letture in memoria e una scrittura, da cui segue il numero di accessi indicati in tabella. - Le letture degli elementi di A generano soltanto MISS (384 MISS / 384 Accessi). Le letture degli elementi di B generano soltanto MISS (384 MISS / 384 Accessi). Le scritture degli elementi di B generano soltanto HIT (384 HIT / 384 Accessi), da cui seguono i valori di HIT e MISS rate indicati in tabella. - Ogni iterazione di ciascun task genera un ciclo di Write-Back, quando lo stato MESI dell'i-esimo elemento di B passa da M a S, con l'eccezione dell'ultima iterazione di T1, da cui segue il numero di Write-Back indicato in tabella.		