

**COMPITO DI CALCOLATORI ELETTRONICI 2019/2020**  
**28 gennaio 2020**

**NOME:**

**COGNOME:**

**MATRICOLA:**

Scrivere in stampatello NOME, COGNOME e MATRICOLA su ogni foglio.

Al termine, si DOVRANNO consegnare tutti i fogli ricevuti.

**NOTA:** La prova si considera superata con esito positivo solo se sono verificate tutte le seguenti condizioni:

- a) *Votazione sufficiente della prima parte - esercizi 1-3 (6/11);*
- b) *Esercizio 2 svolto in modo sufficiente;*
- c) *Votazione sufficiente della seconda parte - esercizi 4-7 (9/16).*

---

**ESERCIZIO 1 [3 punti]** Si consideri la seguente procedura in Assembler MIPS per il calcolo della funzione ricorsiva  $T(n)$  di argomento intero non negativo  $n$ , così definita:

$$T(n) = \begin{cases} 1 & \text{if } n < 2 \\ T(n-1) + T(n-3) & \text{if } n \geq 2 \end{cases}$$

```
1   Tn:   addi $sp, $sp, -4
2         sw   $s0, 0($sp)
3         slti $t0, $a0, 2
4         beq  $t0, $0, Else
5         addi $v0, $v0, 1
6         j    End
7   Else: addi $a0, $a0, -1
8         jal  Tn
9         add  $s0, $0, $v0
10        add  $a0, $a0, -2
11        jal  Tn
12        sll  $t0, $a0, 1
13        add  $v0, $v0, $s0
14        add  $v0, $v0, $t0
15   End:  lw   $s0, 0($sp)
16        addi $sp, $sp, 4
17        jr   $ra
```

Si chiede di verificare se la procedura è corretta (incluso l'uso corretto delle convenzioni di chiamata a procedura MIPS) e correggere – nel caso - gli errori individuati, fornendo adeguate spiegazioni.

**ESERCIZIO 2 [4 punti]** Scrivere una procedura in assembler MIPS che, dato in ingresso un array di interi positivi  $\mathbf{A}$  di lunghezza  $m$ , con argomenti passati tramite i registri  $\$a0$  e  $\$a1$ , rispettivamente, restituisce, tramite il registro  $\$v0$ , la somma degli elementi di  $\mathbf{A}$  che soddisfano la proprietà di essere numeri primi. Per verificare se un numero è primo si può assumere l'esistenza di una procedura di libreria **primo** che, passato in  $\$a0$  un argomento intero, restituisce in  $\$v0$  il valore 1 se  $\$a0$  è un numero primo, 0 altrimenti.

**Nota Bene:** 1) Non si possono usare pseudoistruzioni; 2) è richiesto l'uso delle convenzioni di chiamata a procedura usate dall'assembler MIPS.

**ESERCIZIO 3: [4 punti]** Scrivere una funzione in C che, date in ingresso due matrici di interi  $M$  e  $R$  (di cui  $M$  di dimensioni  $m \times n$  ed  $R$  di dimensioni  $m \times 2$ ), restituisce in uscita un vettore  $V$  di dimensione  $m$ , in cui l'elemento  $V[j]$  è pari al massimo degli elementi della porzione della riga  $j$ -esima di  $M$ , dove la porzione di riga è identificata dall'intervallo di estremi  $R[j][0]$  e  $R[j][1]$ . Per gestire eventuali valori dell'intervallo specificato in  $R$  ed esterni

all'intervallo degli indici ammissibile per  $M$ , assumere che se  $R[j][0] \leq 0$  è da considerare pari a 0, mentre se  $R[j][1] \geq n$  è da considerare pari a  $n-1$ . Assumere inoltre che  $R[j][0] \leq R[j][1]$  per ogni  $j$ .

Esempio: date in ingresso le matrici  $M = \begin{pmatrix} 7 & 2 & 3 \\ 5 & -1 & 2 \\ 1 & 4 & 2 \end{pmatrix}$  e  $R = \begin{pmatrix} 1 & 2 \\ 0 & 1 \\ -1 & 4 \end{pmatrix}$ , la funzione restituisce  $V = (3 \ 5 \ 4)$ , essendo 3 il massimo degli elementi di  $M[0][1..2]$ , 5 il massimo degli elementi di  $M[1][0..1]$  e 4 il massimo degli elementi di  $M[2][0..2]$ .

**ESERCIZIO 4: [3 punti]** Illustrare, fornendo adeguate spiegazioni, come l'ALU del processore MIPS supporta l'implementazione dell'istruzione `slt`.

**ESERCIZIO 5: [5 punti]** Si consideri il seguente frammento di codice in assembler MIPS:

```
add    $t0, $s1, $s0
lw     $s2, 0($t0)
lw     $s3, 4($t0)
add    $t1, $s2, $s3
lw     $s4, 8($t1)
add    $t1, $t1, $s4
sw     $t1, 12($t0)
```

- 1) Si consideri il processore MIPS con pipeline visto a lezione (in grado di scrivere e leggere il banco registri nello stesso ciclo di clock):
  - a. Disegnare il diagramma a ciclo multiplo della pipeline ed individuare le criticità sui dati;
  - b. Inserire gli stalli necessari a risolvere le criticità individuate.
- 2) Si supponga che nella pipeline sia stato introdotto il forwarding dei dati:
  - a. Disegnare il diagramma a ciclo multiplo della pipeline ed individuare le criticità rimaste; indicare i percorsi di forwarding utilizzati.
  - b. Inserire gli stalli necessari a risolvere le criticità rimaste.

Si richiede di accompagnare le soluzioni proposte da adeguati commenti.

**ESERCIZIO 6: [5 punti]** Si consideri una memoria cache set-associativa a 2 vie avente una dimensione complessiva della sola parte dati pari a 64 KB, blocchi da 32 B ciascuno e indirizzi di memoria a 24 bit (indirizzamento al byte).

- a) Indicare la struttura dell'indirizzo di memoria, specificando la dimensione dei vari campi in bit.
- b) Determinare quanti byte occupa complessivamente la memoria cache.
- c) Descrivere i passi necessari alla ricerca in cache del byte d'indirizzo `0xAY5Y7Y` (Y è da sostituire con la cifra meno significativa non nulla del proprio numero di matricola). Qual è l'esito della ricerca del byte sopra indicato se in cache sono già presenti i byte d'indirizzo `0xAY7Y7Y`, `0xAYD5YA` e `0xAY5Y63` ed i corrispondenti blocchi in cache sono validi?
- d) Descrivere due politiche per la sostituzione di un blocco in una memoria cache set-associativa o completamente associativa, spiegandone anche vantaggi ed eventuali svantaggi.

Si richiede di motivare opportunamente le risposte fornite.

**ESERCIZIO 7: [3 punti]** Si consideri una rappresentazione binaria in virgola mobile a 16 bit, di cui (*nell'ordine da sinistra a destra*) 1 bit per il segno (1=negativo), 5 bit per l'esponente, che è rappresentato in eccesso 16, e 10 bit per la parte frazionaria della mantissa  $m$ . In corrispondenza a tutti i valori dell'esponente diversi da 00000, la mantissa è normalizzata tra 1 e 2 ( $1 \leq m < 2$ ). Con l'esponente 00000 si rappresentano invece numeri denormalizzati, con esponente per convenzione pari a -15 e mantissa compresa tra 0 e 1 ( $0 < m < 1$ ).

- a) Calcolare in termini di potenze di 2 e di 10 il minimo ed il massimo numero positivo rappresentabili nella notazione data, specificando anche i rispettivi numerali.
- b) Dati i numeri  $n=0x55Y7$  e  $r=0x81Y2$  rappresentati in complemento a 2 (Y è da sostituire con la cifra meno significativa non nulla del proprio numero di matricola), determinare i numerali che approssimano meglio  $n$  ed  $r$  nella notazione data, esprimendoli come stringa esadecimale.

Si richiede di motivare opportunamente le risposte fornite.

**COMPITO DI CALCOLATORI ELETTRONICI 2019/2020**  
**28 gennaio 2020**

**NOME:**

**COGNOME:**

**MATRICOLA:**

Scrivere in stampatello NOME, COGNOME e MATRICOLA su ogni foglio.

Al termine, si DOVRANNO consegnare tutti i fogli ricevuti.

**NOTA:** La prova si considera superata con esito positivo solo se sono verificate tutte le seguenti condizioni:

- a) Votazione sufficiente della prima parte - esercizi 1-3 (6/11);
- b) Esercizio 2 svolto in modo sufficiente;
- c) Votazione sufficiente della seconda parte - esercizi 4-7 (9/16).

---

**ESERCIZIO 1 [3 punti]** Si consideri la seguente procedura in Assembler MIPS per il calcolo della funzione ricorsiva  $T(n)$  di argomento intero non negativo  $n$ , così definita:

$$T(n) = \begin{cases} 1 & \text{if } n < 4 \\ T(n/2) - T(n/4) & \text{if } n \geq 4 \end{cases}$$

```
1  Tn:  addi $sp, $sp, -4
2      sw   $ra, 0($sp)
3      slti $t0, $a0, 4
4      bne $t0, $0, Else
5      addi $v0, $0, 1
6      j    End
7  Else: srl  $a0, $a0, 1
8      jal  Tn
9      add  $t1, $0, $v0
10     sll  $a0, $a0, 1
11     jal  Tn
12     sub  $v0, $v0, $t1
13  End: lw   $ra, 0($sp)
14     addi $sp, $sp, 4
15     jr   $ra
```

Si chiede di verificare se la procedura è corretta (incluso l'uso corretto delle convenzioni di chiamata a procedura MIPS) e correggere – nel caso - gli errori individuati, fornendo adeguate spiegazioni.

**ESERCIZIO 2 [4 punti]** Scrivere una procedura in assembler MIPS che, dato in ingresso un array di interi positivi  $A$  di lunghezza  $m$ , con argomenti passati tramite i registri  $\$a0$  e  $\$a1$ , rispettivamente, restituisce, tramite il registro  $\$v0$ , il numero di elementi di  $A$  che sono potenze di 2 (1,2,4,8,16,...). Per verificare se un numero è una potenza di 2 si può assumere l'esistenza di una procedura di libreria `pow2` che, passato in  $\$a0$  un argomento intero, restituisce in  $\$v0$  il valore 1 se  $\$a0$  è una potenza di 2, 0 altrimenti.

**Nota Bene:** 1) Non si possono usare pseudoistruzioni; 2) è richiesto l'uso delle convenzioni di chiamata a procedura usate dall'assembler MIPS.

**ESERCIZIO 3: [4 punti]** Scrivere una funzione in C che, date in ingresso due matrici di interi  $A$  e  $I$  (di cui  $A$  di dimensioni  $m \times n$  ed  $I$  di dimensioni  $m \times 2$ ), restituisce in uscita un vettore  $V$  di dimensione  $m$ , in cui l'elemento  $V[i]$  è pari al minimo degli elementi della porzione della riga  $i$ -esima di  $A$ , dove la porzione di riga è identificata dall'intervallo di estremi  $I[i][0]$  e  $I[i][1]$ . Per gestire eventuali valori dell'intervallo specificato in  $I$  ed esterni

all'intervallo degli indici ammissibile per  $A$ , assumere che se  $I[i][0] \leq 0$  è da considerare pari a 0, mentre se  $I[i][1] \geq n$  è da considerare pari a  $n-1$ . Assumere inoltre che  $I[i][0] \leq I[i][1]$  per ogni  $j$ .

Esempio: date in ingresso le matrici  $A = \begin{pmatrix} 1 & 7 & 2 & 5 \\ 2 & 5 & 1 & 6 \\ 8 & 3 & 4 & 2 \end{pmatrix}$  e  $I = \begin{pmatrix} 2 & 3 \\ -1 & 5 \\ 0 & 2 \end{pmatrix}$ , la funzione restituisce  $V = (2 \ 1 \ 3)$ , essendo 2 il minimo degli elementi di  $A[0][2\dots3]$ , 1 il minimo degli elementi di  $A[1][0\dots3]$  e 3 il minimo degli elementi di  $A[2][0\dots2]$ .

**ESERCIZIO 4: [3 punti]** Illustrare, fornendo adeguate spiegazioni, come l'ALU del processore MIPS supporta l'implementazione dell'istruzione `slt`.

**ESERCIZIO 5: [5 punti]** Si consideri il seguente frammento di codice in assembler MIPS:

```
lw    $s2, 0($t1)
lw    $s3, 4($t1)
lw    $s1, 8($t1)
add   $t1, $s2, $s3
add   $t1, $t1, $s1
lw    $s0, 0($t1)
add   $t1, $t1, $s0
```

- 3) Si consideri il processore MIPS con pipeline visto a lezione (in grado di scrivere e leggere il banco registri nello stesso ciclo di clock):
  - a. Disegnare il diagramma a ciclo multiplo della pipeline ed individuare le criticità sui dati;
  - b. Inserire gli stalli necessari a risolvere le criticità individuate.
- 4) Si supponga che nella pipeline sia stato introdotto il forwarding dei dati:
  - a. Disegnare il diagramma a ciclo multiplo della pipeline ed individuare le criticità rimaste; indicare i percorsi di forwarding utilizzati.
  - b. Inserire gli stalli necessari a risolvere le criticità rimaste.

Si richiede di accompagnare le soluzioni proposte da adeguati commenti.

**ESERCIZIO 6: [5 punti]** Si consideri una memoria cache set-associativa a 4 vie avente una dimensione complessiva della sola parte dati pari a 64 KB, blocchi da 16 parole e parole da 4B ciascuna, e indirizzi di memoria a 24 bit (indirizzamento al byte).

- e) Indicare la struttura dell'indirizzo di memoria, specificando la dimensione dei vari campi in bit.
- f) Determinare l'overhead in bit necessario per realizzare la memoria cache.
- g) Descrivere i passi necessari alla ricerca in cache del byte d'indirizzo `0xAyDY9Y` (Y è da sostituire con la cifra meno significativa non nulla del proprio numero di matricola). Qual è l'esito della ricerca del byte sopra indicato se in cache sono già presenti i byte d'indirizzo `0xAy9Y91`, `0xAyCY83` e `0xAyDYB6` ed i corrispondenti blocchi in cache sono validi?
- h) Descrivere quali soluzioni a livello hardware possono essere attuate in una memoria cache per ridurre il miss rate, discutendone anche gli eventuali svantaggi.

Si richiede di motivare opportunamente le risposte fornite.

**ESERCIZIO 7: [3 punti]** Si consideri una rappresentazione binaria in virgola mobile a 24 bit, di cui (*nell'ordine da sinistra a destra*) 1 bit per il segno ( $0=\text{positivo}$ ), 7 bit per l'esponente, che è rappresentato in eccesso 64, e 16 bit per la parte frazionaria della mantissa  $m$ . In corrispondenza a tutti i valori dell'esponente diversi da 0000000, la mantissa è normalizzata tra 1 e 2 ( $1 \leq m < 2$ ). Con l'esponente 0000000 si rappresentano invece numeri denormalizzati, con esponente per convenzione pari a -63 e mantissa compresa tra 0 e 1 ( $0 < m < 1$ ).

- c) Calcolare in termini di potenze di 2 e di 10 il minimo ed il massimo numero negativo rappresentabili nella notazione `data`, specificando anche i rispettivi numerali.
- d) Dati i numeri  $s=0x809Y4$  e  $t=0xA3Y4B1$  rappresentati in eccesso  $2^{23}$  (Y è da sostituire con la cifra meno significativa non nulla del proprio numero di matricola), determinare i numerali che approssimano meglio  $s$  ed  $t$  nella notazione `data`, esprimendoli come stringa esadecimale.

Si richiede di motivare opportunamente le risposte fornite.