

SECONDA PROVA INTERMEDIA DEL MODULO DI
CALCOLATORI ELETTRONICI
 CORSO DI LAUREA IN INGEGNERIA ELETTRICA, ELETTRONICA ED INFORMATICA
 CORSO DI LAUREA IN INGEGNERIA BIOMEDICA
 ISCRITTI ALL'A.A. 2017/18 e precedenti
 3 giugno 2019

NOME: _____ **COGNOME:** _____ **MATRICOLA:** _____ **CFU:** _____

ESERCIZIO 1 (12 punti)

Implementare una funzione Assembly MIPS `trovaMassimo` che, dati l'indirizzo iniziale di un vettore di valori numerici `v` in `$4`, la sua dimensione `N` in `$5`, ed un valore intero `i` in `$6`, restituisca l'indice `imax` tale che il valore `v[imax]` sia il massimo presente fra i valori di `v` a partire dall'indice `i`. Il valore `imax` va memorizzato in `$7`.

Per l'implementazione della funzione si faccia uso della funzione `carica`, che si suppone già implementata. Essa riceve in ingresso l'indirizzo iniziale di un vettore `v` in `$4` ed un indice `i` in `$5`, e memorizza in `$6` il valore `v[i]`.

Per lo svolgimento dell'esercizio si può fare riferimento alla funzione scritta in C a lato.

```
int trovaMassimo(int *v, int N, int i)
{
    int j, imax, w, vmax;

    vmax=carica(v,i);
    for(j=i+1; j<N; j++)
    {
        w=carica(v,j);
        if (vmax<w)
        {
            vmax=w;
            imax=j;
        }
    }
    return imax;
}
```

ESERCIZIO 2 (6 punti)

Siano dati 16 bit per la rappresentazione di valori numerici in virgola mobile. Si consideri una mantissa `M` frazionaria e normalizzata in segno e valore con modalità 1.M (bit di parte intera implicito), esponente a 6 bit in eccesso 31 e bit di segno.

- 1) (2 punti) Spiegando bene ogni passo del ragionamento, indicare il minimo ed il massimo numero rappresentabili, in valore assoluto ed escluso lo zero.
- 2) (4 punti) Rappresentare il valore 38.75, indicando chiaramente i valori dei bit nei campi segno, esponente e mantissa.

ESERCIZIO 3 (4 punti)

Si progetti una ALU con due operandi `A` e `B` a `N` bit utilizzando un parallel adder e delle opportune reti logiche, descrivendo il relativo schema e realizzazione circuitale con le caratteristiche indicate dalla seguente tabella di verità.

s1	s0	F
0	0	A
0	1	A-1
1	0	A+1
1	1	-1

ESERCIZIO 4 (11 punti)

L'ampiezza delle linee dati ed indirizzi del bus di un calcolatore è pari a 32 bit. La frequenza del clock della CPU è di 2 GHz.

- 1) (5 punti) Ipotesizzando che il bus abbia la stessa frequenza di sincronia del clock della CPU e la durata di una trasmissione sul bus sia pari ad 1 ciclo di clock, e che il tempo di ciclo della memoria sia pari a 10 cicli di clock, illustrare chiaramente il protocollo di lettura su bus sincrono utilizzando l'opportuno grafico, indicando il tempo complessivo di trasferimento di una parola di 32 bit **dalla CPU alla memoria**.
- 2) (4 punti) Un sensore per la lettura delle impronte digitali è in grado di acquisire immagini a 256 livelli di grigio della dimensione di 480x320 pixel. Ogni pixel corrisponde a uno fra 256 livelli di grigio e viene rappresentato con 1 byte. Il buffer del modulo I/O connesso con il sensore è in grado di trasmettere in memoria blocchi di 4 byte. Si assuma che:
 - a. gli indirizzi di memoria calcolati coincidano con il sistema memory-mapped delle periferiche;
 - b. le istruzioni che **non** richiedono trasferimento dati da/a memoria vengono eseguite nel tempo calcolato nel punto 1;
 - c. le istruzioni che richiedono trasferimento dati da/a memoria vengono eseguite nel doppio del tempo delle istruzioni che **non** lo richiedono.

Sulla base delle seguenti istruzioni Assembly MIPS che rappresentano l'acquisizione di un'immagine da sensore, calcolare:

2.1) (1 punto) il valore di x pari al numero complessivo di gruppi da 4 byte da trasferire;

2.2) (3 punti) il tempo occorrente al sistema per la memorizzazione dell'immagine.

```
        addi $5, $0, x
        move $4, $0
tran_loop: beq $4, $5, exit
           lw $7, fingerprint($0) #acquisizione del blocco
           muli $6, $4, 4
           sw $7, memory($6) #salva il blocco in memoria
           addi $4, $4, 1
           j tran_loop
```

- 3) (2 punti) Calcolare il tempo di trasferimento dell'immagine al punto 2 (in ns) nel caso in cui il trasferimento avvenga in DMA block transfer, nell'ipotesi che possa avvenire un trasferimento di 32 bit per ogni ciclo di clock.

ESERCIZIO 1

Funzione trovaMassimo:

Input: $\$4 \leftarrow \&v[0]$; $\$5 \leftarrow N$; $\$6 \leftarrow i$

Output: $\$7 \leftarrow \text{imax}$

Funzione carica:

Input: $\$4 \leftarrow \&v[0]$; $\$5 \leftarrow i$

Output: $\$6 \leftarrow v[i]$

Utilizziamo i seguenti registri.

- Copiamo $\$5$ e $\$6$ rispettivamente in $\$13$ e $\$14$, ripristinandoli prima di uscire
- $j \rightarrow \$8$, $\text{vmax} \rightarrow \11 , $\text{vmax} < v \rightarrow \9

```
trovaMassimo: addi $29, $29, -24
               sw $8, 0($29)
               sw $9, 4($29)
               sw $11, 8($29)
               sw $13, 12($29)
               sw $14, 16($29)
               sw $31, 20($29)
               move $13, $5
               move $14, $6
               move $7, $6           #imax=i
               move $5, $6           #passaggio parametro per carica
               jal carica
               move $11, $6           #vmax=v[i]
               addi $8, $14, 1        #inizializza j=i+1
for:           beq $8, $13, exit      #j==N, exit
               move $5, $8
               jal carica             #carico v in $6
               slt $9, $11, $6        #$9 ← vmax<v
               beq $9, $0, updj        #se $9 è 0, aggiorni j
               move $11, $6           #vmax=v
               move $7, $8            #imax=j
updj:          addi $8, $8, 1          #j++
               j for
exit:          move $5, $13
               move $6, $14
               lw $8, 0($29)
               lw $9, 4($29)
               lw $11, 8($29)
               lw $13, 12($29)
               lw $14, 16($29)
               lw $31, 20($29)
               addi $29, $29, 24
               jr $31
```

ESERCIZIO 2

1)

Abbiamo 15 bit utili a parte il bit di segno, dei quali 9 per la mantissa e 6 per l'esponente.

Il minimo valore rappresentabile si ottiene considerando il minimo valore dell'esponente, che essendo in eccesso 31, è, per definizione di eccesso k , pari a $-k$, ovvero -31 , e il minimo valore della mantissa, corrispondente ad un campo formato da tutti zeri poiché il bit implicito per definizione è sempre impostato ad 1. Il minimo valore è dunque pari a $1.0 \cdot 2^{-31}$. Il massimo valore rappresentabile si ottiene valutando il massimo esponente, che si ricava dalla formula $2^n - 1 - k$, con n numero di bit usati per l'esponente. Il risultato è $64 - 1 - 31 = 32$. La massima mantissa, di 9 bit, si ottiene ponendo tutti i bit disponibili a 1, considerato anche il bit implicito, pari a $2 \cdot 2^{-9}$. Il massimo valore rappresentabile è dunque $(2 \cdot 2^{-9}) \cdot 2^{32}$, ovvero 2^{33-2^9} .

2)

Per rappresentare il valore 38.75 in virgola mobile, vanno innanzi tutto convertiti in binario mediante gli algoritmi della divisione successiva per la parte intera, e della moltiplicazione successiva per la parte frazionaria, ottenendo 100110.11.

Non ci resta che normalizzare la mantissa ottenendo: $1.0011011 \cdot 2^5$.

Gli esponenti vanno scritti con sei bit in eccesso 31. Per ottenere la rappresentazione in eccesso 31 è sufficiente sommare appunto l'eccesso al valore numerico da rappresentare, e poi convertirlo in binario. Si ha $5 + 31 = 36 \rightarrow 100100$.

Quindi i valori sono rappresentati nel campo a 16 bit come segue:

Valore	S	Esponente						Mantissa								
38.75	0	1	0	0	1	0	0	0	0	1	1	0	1	1	0	0

ESERCIZIO 3

Per lo schema generale si vedano le dispense del corso. In funzione delle operazioni richieste si hanno i valori seguenti per gli operandi e il riporto in ingresso:

s1	s0	F	A_i^{new}	B_i^{new}	c_{in}
0	0	A	A_i	0	0
0	1	$A-1$	A_i	1^*	0
1	0	$A+1$	A_i	0	1
1	1	-1	1^*	0	0

$s_1 s_0$	00	01	11	10
A_i				
0			1	
1	1	1	1	1

$$A_i^{new} = A_i + s_1 s_0$$

Le formule per B_i^{new} e c_{in} si ricavano direttamente dalla tabella:

$$B_i^{new} = \bar{s}_1 s_0$$

$$c_{in} = s_1 \bar{s}_0$$

Si lascia allo studente il dettaglio della realizzazione circuitale della ALU ottenuta.

*Si ricordi che -1 in complemento a 2 si rappresenta con una stringa di 1.

ESERCIZIO 4

Soluzione domanda 1.

La durata di un ciclo di clock è pari a $1/2 \text{ ns} = 0.5 \text{ ns}$

La lettura su un bus sincrono avviene secondo il protocollo seguente:

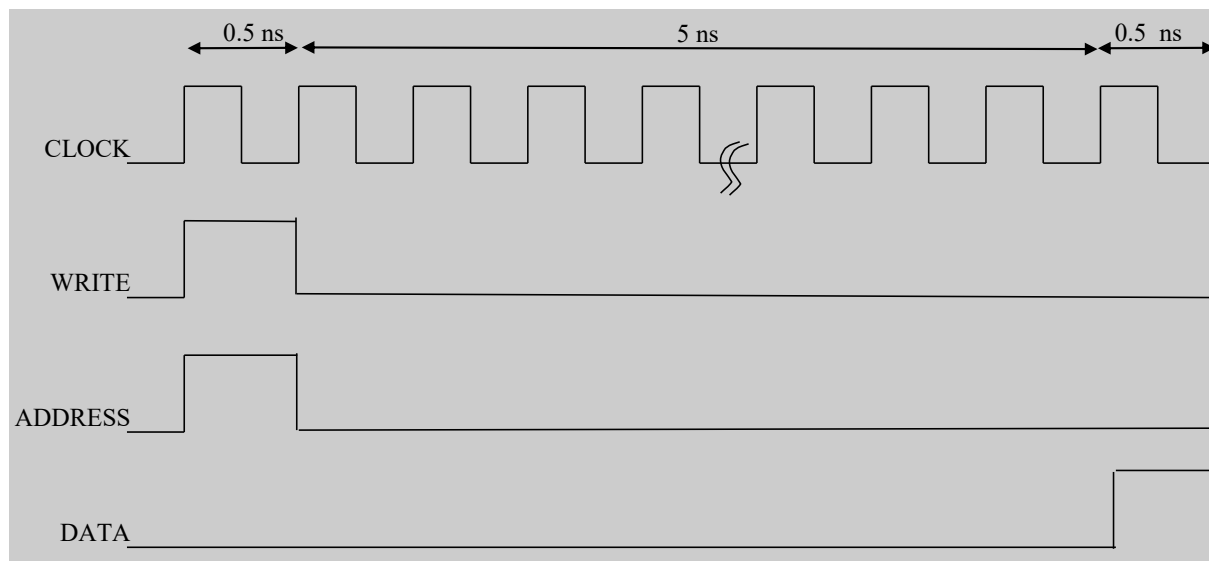
- Segnale di READ sulla linea di controllo e contemporaneamente l'indirizzo della locazione in cui risiede il dato sulla linea indirizzi:

1 ciclo di clock = 0.5 ns

- Lettura della parola dalla memoria: $0.5 \times 10 = 5 \text{ ns}$
- Trasferimento della parola dalla memoria:

trasferimento della parola = 0.5 ns

Tempo totale per leggere una parola dalla memoria = $(1 + 5) \text{ ns} = 6 \text{ ns}$



Soluzione domanda 2.1.

x dev'essere pari al numero complessivo di gruppi da 4 byte da trasferire. Dalle informazioni sul testo abbiamo che l'immagine è formata da 480×320 byte (256 livelli di grigio si rappresentano con 1 byte), che a gruppi di 4 diventano 38400. Quindi $x=38400$.

Soluzione domanda 2.2.

Sostituendo ad x il valore calcolato al punto precedente otteniamo per ciascuna istruzione i seguenti tempi di esecuzione:

Istruzioni	Tempo (in ns)
addi \$5, \$0, 38400	6
move \$4, \$0	6
tran_loop: beq \$4, \$5, exit	6
lw \$7, fingerprint(\$0)	12
muli \$6, \$4, 4	6
sw \$7, memory(\$6)	12
addi \$4, \$4, 1	6
j tran_loop	6

Il blocco di istruzioni a partire da `tran_loop` viene eseguito 38400 volte per cui abbiamo un tempo pari a 48 ns moltiplicato 38400. Al valore ottenuto dobbiamo aggiungere 12 ns per le due istruzioni di inizializzazione. Per cui il tempo complessivo è pari a 1843212 ns (circa 2 ms).

Soluzione domanda 3.

In questo caso il trasferimento di una parola necessita soltanto 0.5 ns, per cui il tempo complessivo è pari a $38400 \times 0.5 = 19200 \text{ ns}$. La modalità DMA block transfer è quasi 100 volte più rapida del trasferimento per I/O programmato calcolato in precedenza.