

**SECONDA PROVA INTERMEDIA DEL MODULO DI
CALCOLATORI ELETTRONICI**
CORSI DI LAUREA IN
INGEGNERIA ELETTRICA, ELETTRONICA ED INFORMATICA
INGEGNERIA BIOMEDICA
IMMATRICOLATI A.A. 2015/16
20 maggio 2016

NOME:

COGNOME:

MATRICOLA:

ESERCIZIO 1 (12 punti)

1. (2 punti) Scrivere la sequenza di istruzioni MIPS che implementino lo pseudo-codice:

$$\$4 \leftarrow A[i]$$

Dove A è un indirizzo simbolico di memoria e i rappresenta l'offset rispetto all'indirizzo dato. Nell'implementare la soluzione, ricordare i vincoli di distanza tra una parola e la successiva nel MIPS.

2. (6 punti) Si scriva un frammento di codice MIPS che implementi le pseudo-istruzioni seguenti:

```
i=0
j=N-1
while i<N
    if A[i]<A[j]
        A[i]=2*A[j]
        j=j-1
    i=i+1
```

Si utilizzino i registri \$8, \$9, \$10, per le variabili i, j, N, ma si ricordi che A è un vettore il cui indirizzo iniziale (A o &A[0]) è memorizzato nel registro \$4.

3. (4 punti) Interpretare il seguente frammento di codice Assembly MIPS, istruzione dopo istruzione, considerando e modificando lo stato della memoria e dei registri a partire da quello in figura (PC è il contatore di programma).

```
inizio: muli $4, $4, 4
        addi $5, $0, 1000
        jr $5
exit:    ...
```

Memoria		
\$4	2	1000
\$5	-5	1004
\$8	150	
PC	inizio	
		1024
		1028
		1032

ESERCIZIO 2 (7 punti)

Si consideri la seguente rappresentazione di numeri nel formato binario: campo complessivo di 16 bit; virgola fissa, con parte frazionaria di 6 bit e bit di segno

1. (2 punti). Esprimere il minimo ed il massimo valore rappresentabili in valore assoluto, escluso lo zero.
2. (3 punti) Rappresentare il valore 215.525.
3. (2 punti) Indicare a quanto ammonta in percentuale la perdita di precisione eventuale nella rappresentazione del valore dato rispetto al valore atteso.

ESERCIZIO 3 (6 punti)

Si progetti una ALU con operandi 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 (l'apice indica il complemento a 1 bit per bit). Si consideri in tutti e quattro i casi il c_in forzato a 1:

s1	s0	F
0	0	A'
0	1	A-B
1	0	B-A
1	1	B'

ESERCIZIO 4 (8 punti)

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

- 1) (3 punti) Ipotizzando che il bus sincrono abbia la stessa frequenza di clock della CPU e la durata di una trasmissione sul bus sia pari a 2 cicli di clock, e che il tempo di ciclo della memoria sia pari a 20 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.
- 2) (3 punti) Si considerino le seguenti istruzioni Assembly MIPS assumendo che gli indirizzi di memoria calcolati coincidano con il sistema memory-mapped delle periferiche. Se le istruzioni che **non** richiedono trasferimenti lungo il bus dati/indirizzi impiegano il 10% del tempo che impiegano le istruzioni di trasferimento dati da/a memoria, tempo da intendersi calcolato nel punto 1, quanto tempo occorre al sistema per effettuare la stampa indicata dalle istruzioni sottostanti?

```
        addi $5, $0, 10
        move $4, $0
transfer_loop: sw $4, printer($0)    #stampa memory-mapped
        addi $4, $4, 1
        bne $4, $5, transfer_loop
```

- 3) (2 punti) Calcolare la massima velocità di trasferimento dati (in bit/s) nel caso in cui il trasferimento avvenga in DMA per un'analogia trasmissione.

Soluzioni

ESERCIZIO 1

Soluzione domanda 1.

Supponendo che *i* sia memorizzato nel registro \$1, le istruzioni prevedono:

- Calcolo dell'offset completo
- Istruzione di trasferimento

```
muli $1, $1, 4
lw $4, A($1)
```

Soluzione domanda 2.

Oltre a quelli forniti dal testo, utilizziamo \$11 per l'indirizzo di A[i] o A[j], \$12 per A[i], \$13 per A[j], \$1 per la condizione A[i]<A[j].

```
        move $8, $0          #i=0
        addi $9, $10, -1     #j=N-1
loop:   muli $11, $9, 4       #calcolo offset j*4
        add $11, $11, $4     # $11 ← &A[j]
        lw $13, 0($11)       # $13 ← A[j]
while:  beq $8, $10, exit     #se i==N vai ad exit
        muli $11, $8, 4       #calcolo offset i*4
        add $11, $11, $4     # $11 ← &A[i]
        lw $12, 0($11)       # $12 ← A[i]
        addi $8, $8, 1       #i=i+1
        slt $1, $12, $13     # $1 ← A[i]<A[j]
        beq $1, $0, while    #se A[i]>=A[j] vai a while
        muli $12, $13, 2     # $12 ← 2*A[j]
        sw $12, 0($11)       # $12 → &A[i]
        subi $9, $9, 1       #j=j-1
        j loop
exit:   ...
```

Soluzione domanda 3.

In neretto l'istruzione in esecuzione e il relativo effetto sulla memoria e/o registri.

<pre>inizio: muli \$4, \$4, 4 addi \$5, \$0, 1000 jr \$5 exit: ...</pre>	<table><tr><td>\$4</td><td>8</td></tr><tr><td>\$5</td><td>-5</td></tr><tr><td>\$8</td><td>150</td></tr><tr><td>PC</td><td>inizio+4</td></tr></table>	\$4	8	\$5	-5	\$8	150	PC	inizio+4	<table><tr><td></td><td>Memoria</td></tr><tr><td>1000</td><td>sw \$8, 1024(\$4)</td></tr><tr><td>1004</td><td>j exit</td></tr><tr><td></td><td>....</td></tr><tr><td>1024</td><td>15</td></tr><tr><td>1028</td><td>7</td></tr><tr><td>1032</td><td>-1</td></tr></table>		Memoria	1000	sw \$8, 1024(\$4)	1004	j exit		1024	15	1028	7	1032	-1
\$4	8																							
\$5	-5																							
\$8	150																							
PC	inizio+4																							
	Memoria																							
1000	sw \$8, 1024(\$4)																							
1004	j exit																							
																							
1024	15																							
1028	7																							
1032	-1																							

<pre> inizio: muli \$4, \$4, 4 addi \$5, \$0, 1000 jr \$5 exit: ... </pre>	<div> <div> <div>\$4</div><div>8</div> </div> <div> <div>\$5</div><div>1000</div> </div> <div> <div>\$8</div><div>150</div> </div> <div> <div>PC</div><div>inizio+8</div> </div> </div> <div> <div>1000</div><div>sw \$8, 1024(\$4)</div> </div> <div> <div>1004</div><div>j exit</div> </div> <div> <div>1024</div><div>15</div> </div> <div> <div>1028</div><div>7</div> </div> <div> <div>1032</div><div>-1</div> </div>
--	---

ESERCIZIO 2

Soluzione alla domanda 1.

Il minimo valore si ha con il bit meno significativo della parte frazionaria ad 1, da cui:

$$\text{minimo} = 2^{-6} = 0.015625$$

Il massimo valore si ha con tutti i bit pari ad 1. Poiché il bit di segno non va conteggiato, si hanno nove bit in parte intera e 6 in parte frazionaria, da cui:

$$\text{massimo} = 2^9 - 2^{-6} = 511.984375$$

Soluzione alla domanda 2.

Utilizziamo l'algoritmo delle divisioni successive per la parte intera ottenendo

Quoziente	215	107	53	26	13	6	3	1
Resto	1	1	1	0	1	0	1	1

$$215_{10} = 11010111_2.$$

Utilizziamo l'algoritmo delle moltiplicazioni successive per la parte frazionaria:

Valore	1.05	0.1	0.2	0.4	0.8	1.6
--------	------	-----	-----	-----	-----	-----

Poiché la parte frazionaria ha solo sei bit, dobbiamo troncare il calcolo ad essi:

$$0.525_{10} = 0.100001_2.$$

In sintesi:

$$215.525_{10} = 11010111.100001_2.$$

Soluzione alla domanda 3.

Il valore calcolato, per via del troncamento, non rappresenta esattamente il valore dato ma : $215 + 2^{-1} + 2^{-6} = 215.515625$, con una perdita di precisione pari a 0,004% circa rispetto ad esso.

ESERCIZIO 3

Per il circuito e le funzioni circuitali 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	B	C _{in}
0	0	A'	A'	-1	1
0	1	A-B	A	B'	1
1	0	B-A	A'	B	1
1	1	B'	-1	B'	1

s ₁ s ₀ A _i	00	01	11	10
0	1		1	1
1		1	1	

$$A_i^{\text{new}} = s_0 A_i + \bar{s}_0 \bar{A}_i + s_1 \bar{A}_i$$

s ₁ s ₀ B _i	00	01	11	10
0	1	1	1	
1	1			1

$$B_i^{\text{new}} = s_0 \bar{B}_i + \bar{s}_0 B_i + \bar{s}_1 \bar{s}_0$$

$$c_{\text{in}} = 1$$

L'esercizio per essere completato richiede il disegno su porte logiche dei tre circuiti di cui sopra, la cui realizzazione si lascia allo studente.

ESERCIZIO 4

Soluzione domanda 1.

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

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:

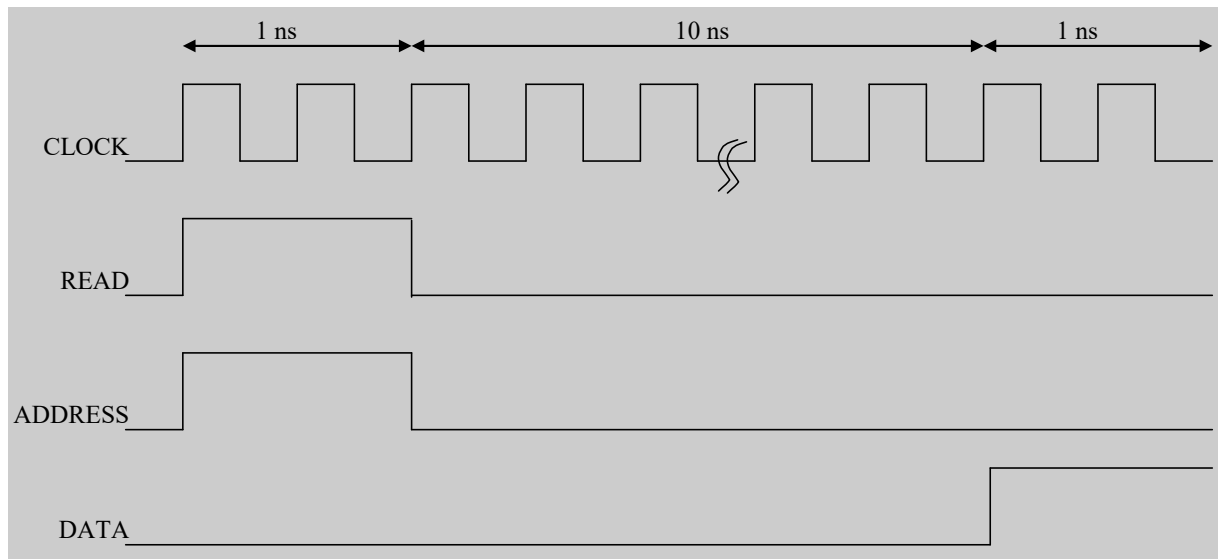
2 cicli di clock = 1 ns

- Lettura della parola dalla memoria: $0.5 \times 20 = 10 \text{ ns}$

- Trasferimento della parola dalla memoria:

trasferimento della parola = 1 ns

Tempo totale per leggere una parola dalla memoria = $(2 + 10) \text{ ns} = 12 \text{ ns}$



Soluzione domanda 2.

Il testo indica chiaramente che si tratta di una stampa di 10 parole. Abbiamo due istruzioni che impiegano ciascuna 1.2 ns per essere eseguite, e poi un ciclo che ripete il trasferimento di dette parole con le seguenti tempistiche $10 \times (12 + 1.2 + 1.2) = 10 \times 14.4 = 144 \text{ ns}$. In totale il trasferimento di queste dieci parole richiede $2.4 + 144 = 146.4 \text{ ns}$, ovvero 14.64 ns a parola. La velocità di trasferimento è dunque pari a $32 \text{ bit} / 14.64 = 2.19 \text{ Gbps} = 0.27 \text{ GB/s}$

Soluzione domanda 3.

In DMA è possibile trasferire un blocco per ogni ciclo di clock (modalità "block transfer"), quindi la velocità sarà ben maggiore rispetto al caso precedente, ovvero una parola ogni 0.5 ns. Pertanto la massima velocità di trasferimento è pari $32 / 0.5 \text{ ns} = 64 \text{ Gbps} = 8 \text{ GB/s}$.

Nota: Ipotizzando che il trasferimento sul bus con DMA richieda lo stesso numero di cicli dei trasferimenti CPU/memoria, una risposta ammissibile è anche $32 / 1 \text{ ns} = 32 \text{ Gbps} = 4 \text{ GB/s}$.