

**PROVA SCRITTA DEL MODULO DI**  
**CALCOLATORI ELETTRONICI**  
**Corsi di Laurea in Ingegneria Biomedica e**  
**Ingegneria Elettrica, Elettronica ed Informatica**  
24 gennaio 2018

**NOME:**

**COGNOME:**

**MATRICOLA:**

**ESERCIZIO 1 (9 punti)**

1) (4 punti) Semplificare con le mappe di Karnaugh la seguente espressione booleana:

$$Y = X\bar{A}B + X\bar{A}\bar{B} + \bar{X}B\bar{C} + XA$$

2) (5 punti) Scrivere per ogni colpo di clock l'uscita Q di un flip flop T in corrispondenza della sequenza di bit 0110 letta in corrispondenza dell'ingresso T. Assumere che lo stato iniziale del FF sia pari a Q=0.

**ESERCIZIO 2 (10 punti)**

Una gerarchia di memoria a due livelli presenta una memoria primaria di 16 KB, parole di 4 B e una memoria cache di 256 B. La memoria primaria è suddivisa in blocchi di 4 parole e l'accesso alla parola intera avviene per indirizzi multipli di 4 (0, 4, 8, 12, ...). Il metodo di indirizzamento della cache è associativo su insiemi a due vie. Si consideri il seguente stato della memoria primaria dove a partire dall'indirizzo 0 sia memorizzato il codice Assembly MIPS indicato in figura ed a partire dall'indirizzo 128 figurino i valori indicati. L'istruzione "halt" serve a fermare l'esecuzione delle istruzioni.

Indirizzo	Contenuto	Indirizzo	Contenuto
0	addi \$10, \$0, 12	128	6
4	addi \$8, \$0, 0	132	-1
8	beq \$8, \$10, 36	136	10
12	muli \$11, \$8, 4	140	5
16	lw \$12, 128(\$11)	144	50
20	addi \$12, \$12, 1	148	-7
24	sw \$12, 128(\$11)	152	23
28	addi \$8, \$8, 1	156	12
32	j 8	160	40
36	halt	164	32
40		168	-17
44	...	172	5
...	...	...	...

1) (5 punti) Scrivere qual è la funzione svolta dal codice MIPS attraverso l'analisi riga per riga.

2) (5 punti) Indicare dove vengono allocate le istruzioni e i dati nella memoria cache nel corso dell'esecuzione del codice, ovvero indicare la corrispondenza blocchi di primaria-linee di cache.

**ESERCIZIO 3 (6 punti)**

I trasferimenti di parole a/dalla memoria di un calcolatore sono codificate utilizzando il codice di Hamming. Si consideri la stringa di 12 bit 011001111100 (il bit meno significativo è a sinistra), risultata della codifica di una parola di N bit secondo il codice di Hamming e appena ricevuta.

1. (1 punto) Calcolare N, supponendo di aver fatto uso del numero minimo di bit di controllo necessario per una stringa di 12 bit e scrivere la stringa di informazione così com'è giunta in ricezione;
2. (5 punti) valutare l'eventuale presenza di errori nella stringa.

**ESERCIZIO 4 (8 punti)**

Considerato un campo di 64 bit, siano data la rappresentazione in virgola mobile con mantissa frazionaria normalizzata in segno e valore (1.M) ed esponente a 8 bit in eccesso 127.

1. (4 punti) Calcolare il minimo e il massimo valore in valore assoluto rappresentabili, escluso lo zero.
2. (4 punti) Sommare i due numeri,  $(12.5)_{10}$   $(5.25)_{10}$ , esprimendoli in virgola mobile, con l'algoritmo dei calcolatori.

### ESERCIZIO 1

- 1) Per quanto riguarda l'espressione booleana, dalle mappe si ha:

$$Y = X\bar{A}B + X\bar{A}\bar{B} + \bar{X}B\bar{C} + XA = X + B\bar{C}$$

XA					
BC		00	01	11	10
	00			1	1
	01			1	1
	11			1	1
	10	1	1	1	1

- 2) I flip flop funzionano e mutano stato in funzione del fronte di salita del segnale in ingresso ad ogni colpo di clock. In particolare il FF-T muta lo stato in funzione del valore dell'ingresso T che dev'essere pari ad 1 se letto sul fronte di salita dell'impulso di clock. Per cui, se la sequenza di valori binari letti ad ogni colpo di clock è 0110, e lo stato iniziale del FF è 0, in uscita si avranno i livelli di stato Q pari a 0100.

### ESERCIZIO 2

La funzione svolta dal codice MIPS corrisponde all'incremento di 1 di ciascun valore contenuto nell'area di memoria indicata dagli indirizzi 128-172.

Con la prima istruzione (indirizzo 0) si carica nel registro \$10 il valore 12. Il registro \$8 viene inizializzato a 0 e ha inizio un ciclo (indirizzi 8-32), che ha come condizione di termine il fatto che \$8 eguagli il valore in \$10, ovvero 12. La tripla di istruzioni agli indirizzi 16-24 serve a caricare nel registro \$12 il valore indicizzato dal registro \$8 (opportunamente moltiplicato per 4), incrementarlo di 1 e memorizzarlo nello stesso indirizzo. Dopo aver incrementato \$8 di 1 (indirizzo 28), si ritorna a verificare la condizione di fine ciclo con un salto incondizionato (indirizzo 32).

La memoria primaria è in grado di indirizzare  $16 \text{ KB} = 2^{14} \text{ B}$ , ovvero  $2^{12}$  parole, considerando che ogni parola è formata da  $2^2 \text{ B} = 4 \text{ B}$ .

La cache è invece formata da  $2^8 \text{ B}$  ovvero  $2^6$  parole. Considerando che il metodo è associativo su insiemi a due vie con blocchi di 4 parole, abbiamo la seguente suddivisione dell'indirizzamento:

< TAG 7 bit> < Set Index 3 bit> <Offset 4 bit>

I due bit meno significativi dell'offset permettono l'accesso al singolo byte, sono quindi i due bit più significativi dell'offset che vanno ad indicare la singola parola. Nel resto della trattazione considereremo sempre pari a 0 i due bit meno significativi, perché il problema richiede sempre l'accesso alla parola intera e mai al singolo byte. Il set index risulta formato da otto insiemi in quanto il metodo di indirizzamento a due vie permette la riduzione di questo campo di un bit che viene assegnato al campo TAG.

Sulla base dell'esecuzione del codice, la corrispondenza blocchi di primaria-linee di cache è la seguente:

Set	Linea 0	Linea 1
0	0-4-8-12	128-132-136-140
1	16-20-24-28	144-148-152-156
2	32-36-40-44	160-164-168-172
...	...	...

### ESERCIZIO 3

Dalla relazione  $N + K \leq 2^K - 1$ , si ottiene facilmente  $K = 4$  e  $N = 8$ .

La parola in ricezione si ottiene considerando che ciascun bit corrisponde ad una particolare configurazione dei bit di controllo calcolati in fase di ricezione in XOR con i bit di controllo ricevuti. In particolare la configurazione con  $e_i = 1$  implica un'alterazione di  $c_i$  ovvero di uno dei bit controllati da  $c_i$ .

Ricalcolando i bit di controllo in ricezione con le formule note, si ha:

$c_3 = 0, c_2 = 0, c_1 = 0, c_0 = 1$ , che posti in XOR con i bit di controllo ricevuti conducono alla configurazione d'errore 1011, corrispondente ad un errore sul bit  $b_6$  che va quindi commutato.

La stringa corretta è dunque: 10111110.

### ESERCIZIO 4

1.

Minimo:  $2^{-127}$  Max:  $2^{128}(2-2^{-56})$ .

2.  $(12.5)_{10} = 1100.1 = 1.1001 \cdot 2^3$   
 $(5.25)_{10} = 101.01 = 1.0101 \cdot 2^2$

I due numeri si possono rappresentare nel seguente modo:

Segno	Esponente	Mantissa
0	10000010	10010000000000000000...0
0	10000001	01010000000000000000...0

Poiché il primo ha esponente maggiore del secondo ( $4 > 3$ ) di quest'ultimo si fa scorrere la mantissa a destra di una posizione.

I due numeri da sommare sono:

$$\begin{array}{r} 0.110010 + \\ 0.010101 = \\ \hline 1.000111 \quad (*2^4) \end{array}$$

E' necessario normalizzare il risultato:

Segno	Esponente	Mantissa
0	10000100	00011100000000000000...0