

Cognome e nome:

Matricola:

Com. Dig.
 Inf. Mus.

1. [2] Rappresentare il numero decimale: $N = \frac{1}{128} + \frac{1}{1024}$ secondo lo standard IEEE-754, singola precisione, esprimendolo in formato esadecimale.

2. [5] Si dimostri la seguente equivalenza applicando le regole dell'algebra booleana. Si scriva inoltre l'espressione duale dell'equivalenza data.

$$(a + \bar{b})(\bar{a} + \bar{c}) = \bar{a}\bar{b} + a\bar{c}$$

3. [5] Si progetti un circuito caratterizzato da 4 bit di ingresso ($a_3 a_2 a_1 a_0$) che rappresentano un numero N , intero con segno (in complemento a 2), e da un'uscita Y che vale '1' se e solo se N , in valore assoluto, è maggiore di 4.

a) Determinare la tabella di verità di Y ; b) esprimerla nella forma canonica più adatta; c) semplificarla mediante mappe di Karnaugh; d) semplificarla ulteriormente, se possibile, mediante semplificazioni algebriche; e) disegnarne lo schema circuitale.

4. [8] Si sintetizzi una macchina a stati finiti di Moore caratterizzata da una linea d'ingresso ed una linea di uscita: l'uscita si porta a "1" quando il valore sulla linea d'ingresso è diverso da entrambi i due valori precedenti, altrimenti vale "0". Si assuma che, allo stato iniziale, la macchina consideri come valori precedenti "00". Si determinino: STG, STT, STT codificata e struttura circuitale del sistema completo, non trascurando la gestione del segnale di clock ed avendo cura di semplificare il più possibile le funzioni prima di tradurle in circuito.

5. [7] Si traducano in linguaggio Assembly MIPS nativo (evitando di utilizzare pseudo-istruzioni) la seguenti procedure in linguaggio C.

La procedura **SommaCub()** si aspetta l'indirizzo base dell'array **vett** nel registro **\$a0**, il numero di elementi del vettore **n_elem** in **\$a1** e restituisce il risultato in **\$v0**. La procedura **Cubo()** si aspetta l'argomento **n** in **\$a0** e restituisce il risultato in **\$v0**.

```
int SommaCub(int vett[], int n_elem) {
    int i, sum=0;
    for(i=0; i<n_elem; i++)
        sum = sum + Cubo(vett[i]);
    return( sum );
}

int Cubo(int n)
{
    return( n*n*n );
}
```

6. [6] Si traduca il seguente frammento di codice Assembly MIPS in linguaggio macchina, in formato esadecimale.

```
0x240A4800:   bne $t0, $t1, -40
              j 0x300
```

Si determini inoltre, per ciascuna delle due istruzioni, la **gittata del salto**, cioè i valori minimo e massimo di indirizzo di memoria ai quali sarebbe possibile saltare, cambiando opportunamente il valore nel campo istruzione.

System calls

	codice (\$v0)	argomenti	risultato
print int	1	\$a0	
print float	2	\$f12	
print double	3	\$f12	
print string	4	\$a0	
read int	5		\$v0
read float	6		\$f0
read double	7		\$f0
read string	8	\$a0, \$a1	
sbrk	9	\$a0	\$v0
exit	10		

Registri MIPS

0	zero	24-25	t8 - t9
1	at	26-27	k0 - k1
2-3	v0 - v1	28	Gp
4-7	a0 - a3	29	Sp
8-15	t0 - t7	30	s8
16-23	s0 - s7	31	Ra

MIPS Instruction Set:

