

Cognome, nome: _____ Matricola: _____ Com. Dig. Inf. Mus.

1. [2] Che numero decimale rappresenta il codice esadecimale: **0xBD00 0000** nel formato standard IEEE-754 a singola precisione?

2. [3] Si dimostri la seguente equivalenza applicando le proprietà dell'algebra di Boole: $(a + b) \oplus c = a\bar{c} + \bar{a}bc + \bar{a}(b \oplus c)$

3. [3] Si disegni la struttura circuitale di un registro a 4 bit e se ne descriva il funzionamento.

4. [5] Si progetti un circuito caratterizzato da quattro ingressi ($a_3 a_2 a_1 a_0$) che rappresentano un numero binario **A** di 4 bit, e da un'uscita **Y** che vale '1' se e solo se:
 • il numero A è pari e compreso tra 5 e 10, oppure
 • è dispari e compreso tra 10 e 15.
 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.

5. [8] Si sintetizzi una macchina a stati finiti di Moore che presenta una linea di ingresso ed una linea di uscita. L'ingresso viene valutato ogni secondo. Ogni volta che l'ingresso passa da '0' a '1' e ci rimane poi per almeno 2 secondi, l'uscita si porta a '1' per 1 secondo e poi torna a '0'. Si supponga che inizialmente sia l'ingresso che l'uscita si trovino a '0'. 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.

6. [6] Si traduca in linguaggio Assembly MIPS nativo, evitando cioè di utilizzare pseudo-istruzioni, la seguente procedura in linguaggio C. Si consideri che tale procedura si aspetta l'argomento nel registro **\$a0** e restituisce il risultato nel registro **\$v0**.

```
int StranaFunz ( int n )
{
    if ( n<3 )    return ( n );
    else        return ( 2*StranaFunz ( n-2 ) );
}
```

7. [6] Si traduca il seguente frammento di codice Assembly MIPS in linguaggio macchina, in binario o esadecimale.

```
0x16000: beq $s0, $a0, -16
          j 0x15000
```

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.

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:

