

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

e nome: _____

- [2] Si esprima il numero: $-(2^{-125})$ in formato standard IEEE-754 a singola precisione, scrivendolo come parola esadecimale.
- [3] Dare la definizione di seconda forma canonica di una funzione logica, e scriverne la formula generica. Determinare la seconda forma canonica della seguente funzione logica: $F(a,b,c) = ab + c$
- [3] Si disegni la struttura circuitale di un **multiplexer a 4 ingressi** e se ne descriva il funzionamento.
- [5] Si progetti un circuito caratterizzato da tre ingressi (a, b, c) e da un'uscita Y che vale '1' se e solo se almeno due ingressi valgono '0'.
 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.
- [7] Si sintetizzi una macchina a stati finiti di Moore caratterizzata da una linea di ingresso A che viene valutata ogni millisecondo, e da 2 linee di uscita, P e D, ognuna delle quali va a "1" quando il numero di fronti di salita pervenuti all'ingresso A è, rispettivamente, pari (P=1) o dispari (D=1). Inizialmente il conteggio dei fronti parte da zero (considerato pari). 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.
- [7] Si traduca in linguaggio Assembly MIPS nativo, evitando cioè di utilizzare pseudo-istruzioni, la seguente coppia di procedure in linguaggio C. Si consideri che entrambe le procedure si aspettano l'argomento nel registro **\$a0** e restituiscono il risultato nel registro **\$v0**.

```
int CheFunzione(int n)
{
  if( n<1 )
    return( 1 );
  else
    return( MezzoQuad(n) * MezzoQuad(n-1) );
}
```

```
int MezzoQuad(int n)
{
  return( n*n/2 );
}

.data 0x1000
.byte -12, +12
.word -70, 0x70
.space 6
.text
start: addi $a0, $zero, 1024
       addi $v0, $zero, 9
       syscall
       ...
```

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	
sbk	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:

