



Cognome, nome:

Matricola:

Com. Dig.
 Inf. Mus.

- [2] Rappresentare il numero decimale **-60,0625** secondo lo standard IEEE-754, singola precisione, in formato esadecimale.
- [3] Si dimostri la seguente equivalenza $(a+b)(\bar{a}+b+c)(a+\bar{c}) = ab + a\bar{b}c + b\bar{c}$ applicando regole di algebra booleana:
- [3] Si disegni la struttura circuitale di un **Latch** di **tipo DT** e se ne descriva il funzionamento.
- [5] Si progetti un circuito caratterizzato da quattro ingressi **A, B, C e D**, e da un'uscita **Y** che vale '1' se e solo se, sui quattro ingressi, la quantità di '1' è strettamente maggiore della quantità di '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.
- [8] Si sintetizzi una macchina a stati finiti di Moore che presenta una linea di ingresso che viene valutata ogni millisecondo e una linea d'uscita che cambia valore (0→1→0→1→...) ogni volta che sulla linea d'ingresso si presenta la sequenza: "1100".
Si supponga che inizialmente la linea d'ingresso e l'uscita si trovino entrambe 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] 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 SuperFunz( int n )
{
    if( n<5 )    return( n );
    else        return( n*SuperFunz( n-5 ) );
}
```

- [4] Si scriva un programma Assembly completo, per ambiente SPIM, per calcolare il valore di "SuperFunz" di un numero intero fornito da tastiera. Il programma esegue il calcolo chiamando la funzione **SuperFunz** descritta nell'esercizio precedente, quindi termina. Il programma deve presentarsi a terminale come nel seguente esempio:

```
Inserisci un numero intero:
> 8
SuperFunz( 8 ) = 24
```

- [4] Si traducano le due istruzioni seguenti a) in Assembly MIPS nativo e b) in linguaggio macchina, specificando valore e ampiezza in bit dei campi di ogni istruzione.

```
bgti $t0, -5, -40    # branch on greater than immediate
divi $a3, 49, $t7    # divide immediate by number in register
```

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:

