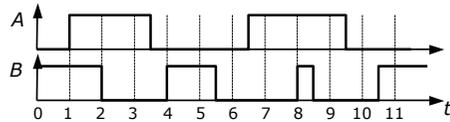


1. [2] Che numero decimale rappresenta il codice esadecimale: **0xC2C80000** nel formato standard IEEE-754, singola precisione?

2. [4] Si dimostri la seguente equivalenza applicando le regole dell'algebra booleana. $(x + y)(\bar{y} + z) = x\bar{y} + yz$

3. [4] Si disegni la struttura circuitale di un **Half Adder** e si tracci l'andamento dell'uscita **S** (somma) in funzione degli ingressi in figura (si trascurino i ritardi di porta).



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 divisibile per 3 (si ricorda che zero è divisibile per 3).

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. [7] Si sintetizzi una macchina a stati finiti di Moore sincrona, caratterizzata da una linea d'ingresso ed una di uscita. L'uscita cambia valore ogni volta che sulla linea d'ingresso si osservano due bit consecutivi uguali. Si assuma che, allo stato iniziale, l'uscita si trovi 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. [7] Si scriva un programma Assembly MIPS, il quale allochi in memoria un array di numeri interi inserito da tastiera dall'utente e, successivamente, calcoli la somma degli elementi dell'array e la stampi a video. Il programma deve apparire a terminale come nell'esempio riportato a lato.

*Quanti elementi? > 7
Inserisci elemento 0 > 15
Inserisci elemento 1 > 21
...
Inserisci elemento 6 > 8
La somma vale: 108*

7. [5] Si traduca il seguente frammento di codice Assembly MIPS in linguaggio macchina, in formato esadecimale, calcolando prima i valori esadecimali **IND_1** e **IND_2** che permettono di saltare esattamente all'indirizzo indicato in ciascun commento.

```
0xA000F000: bne $v1, $a1, IND_1      # IND_1 = 0xA000EF00
                    j IND_2      # IND_2 = 0xA111F000
```

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:

