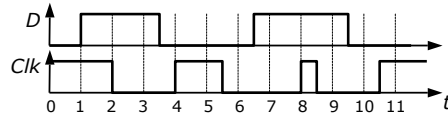


Cognome e nome: _____ Matricola: _____ Com. Dig. Inf. Mus.

1. [2] Si esprima il numero 2^{-27} in formato standard IEEE-754 a singola precisione, scrivendolo come parola esadecimale.

2. [5] Si disegni la struttura circuitale interna di un latch di tipo D sincrono e si disegni l'andamento dell'uscita Q in funzione degli ingressi rappresentati in figura (si trascurino i ritardi di porta).



3. [4] Si progetti un circuito caratterizzato da quattro ingressi ($n_3 n_2 n_1 n_0$) sui quali viene rappresentato un numero intero N in formato binario e con segno, in complemento a 2, e da un'uscita Y che vale '1' se e solo se: $3 \leq |N| \leq 5$.

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 di ingresso (che viene valutata 5 volte al secondo) e da 2 linee di uscita, U0 e U1: U0 va a "1" quando gli ultimi tre valori osservati all'ingresso siano stati tutti "0", mentre U1 va a "1" con gli ultimi tre valori tutti "1". 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. [4] Si scriva un programma Assembly completo, per ambiente SPIM, per calcolare il valore di "StraFunz" di un numero intero fornito da tastiera; il programma esegue il calcolo chiamando la funzione *StraFunz* (che si suppone già data) la quale si aspetta un argomento x nel registro \$a1 e restituisce *StraFunz*(x) nel registro \$v1, quindi termina. Il programma deve presentarsi a terminale come nel seguente esempio:

```
Inserisci l'argomento di StraFunz:
> 23
La StraFunz di 23 vale 46
```

6. [6] Rappresentare il contenuto byte per byte, in formato esadecimale, e gli indirizzi corrispondenti della zona di memoria che viene modificata a seguito dell'esecuzione del segmento dati riportato a lato.

```
.data 0x300
.half -45
.byte +10, -10
.word 0x300

.text
start: addi $a0, $zero, 512
      addi $v0, $zero, 9
      syscall
      ...
```

7. [4] Tradurre ciascuna delle seguenti pseudoistruzioni: a) in Assembly MIPS nativo e b) in linguaggio macchina (specificando ampiezza in bit e valore dei campi delle istruzioni).

```
li $17, (411 + 17) # load immediate
blei $1, -2, -12 # branch on less than/equal to immediate
```

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:

