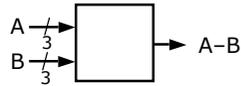


1. [3] Qual è il numero positivo più piccolo rappresentabile nel formato standard IEEE-754, singola precisione? Scriverne sia il valore decimale che la sua rappresentazione esadecimale.

2. [3] Si dimostri la seguente equivalenza applicando le regole dell'algebra booleana. $(x + \bar{y})(\bar{x} + z) = xz + \bar{x}\bar{y}$

3. [5] Si disegni la struttura circuitale di un sottrattore di numeri interi, espressi in complemento a 2 su 3 bit. (suggerimento: si ricordi il metodo per effettuare sottrazioni mediante un sommatore, quando si usa la notazione in complemento a 2)



4. [5] Si progetti un circuito caratterizzato da 3 linee di ingresso ($n_2 n_1 n_0$) che rappresentano un numero N intero con segno (in complemento a 2), e da un'uscita Y che vale '1' se e solo se il valore assoluto di N è minore di 3 ($|N| < 3$).

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 linea di uscita: l'uscita cambia valore ogni volta che la linea d'ingresso passa da "0" a "1" e permane a "1" per almeno 3 colpi di clock. Si assuma che, allo stato iniziale, 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 scriva un programma Assembly completo, per ambiente SPIM, il quale allochi nella memoria un array di numeri interi inserito da tastiera dall'utente e, successivamente, calcoli il prodotto degli elementi dell'array e lo stampi a video. Il programma deve presentarsi a terminale come nell'esempio riportato a lato:

```
Numero di elementi? > 3
Inserisci elemento 0 > 8
Inserisci elemento 1 > 5
Inserisci elemento 2 > 4
Il prodotto vale: 160
```

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.

```
0x10203080: bne $t0, $t1, IND_1      # salta a: 0x102030A0
              j IND_2              # salta a: 0x102030F0
```

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 | 24-25 | t8 - t9 |
|-------|---------|-------|---------|
| 1 | zero | | 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:

