

- [3] Rappresentare il numero  $2^{-130}$  secondo lo standard IEEE-754, singola precisione, in formato esadecimale.
- [4] Si scrivano gli enunciati delle due leggi dell'assorbimento, in entrambe le forme duali, e le si dimostri (in una delle due forme duali, a scelta).
- [4] Si rappresenti la struttura circuitale di un moltiplicatore di parole di 3 bit e se ne calcoli il cammino critico.
- [5] Si progetti un circuito caratterizzato da 4 linee binarie di ingresso (A,B,C,D) e da un'uscita Y che vale '1' se e solo se la sequenza dei 4 bit in ingresso contiene la sottosequenza "11".
  - 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 sincrona, caratterizzata da una linea d'ingresso ed una di uscita. L'uscita cambia valore ogni volta che sulla linea d'ingresso sia arrivata la sequenza: "0011". 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] Si scriva un programma Assembly SPIM, il quale allochi nella memoria un array di numeri interi inserito da tastiera dall'utente e, successivamente, calcoli il valore massimo degli elementi dell'array e lo stampi a video. Il programma deve presentarsi a terminale come nell'esempio riportato a lato.
 

Numero di elementi? > 10  
Inserisci elemento 0 > 15  
Inserisci elemento 1 > 121  
...  
Inserisci elemento 9 > 8  
Il massimo vale: 121
- [5] Si traduca il seguente frammento di codice Assembly MIPS in linguaggio macchina, in formato esadecimale, calcolando prima i valori esadecimali ADDR\_1 e ADDR\_2 che permettono di saltare esattamente all'indirizzo indicato in ciascun commento.
 

```
0xA1230F00: bne $t0, $t1, ADDR_1    # ADDR_1 = 0xA1230EF4
              j  ADDR_2           # ADDR_2 = 0xA2345678
```

### 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      |

