1. [3] Con riferimento allo standard IEEE-754 – singola precisione, scrivere la rappresentazione del più piccolo numero positivo denormalizzato e del più piccolo numero positivo normalizzato, in formato esadecimale, indicando anche a che valori ciascuno di essi corrisponde.

2. [4] Si costruisca il circuito che implementa la funzione $f(a,b,c)=(a\oplus b)+b\overline{c}$ utilizzando soltanto porte NAND a due ingressi.

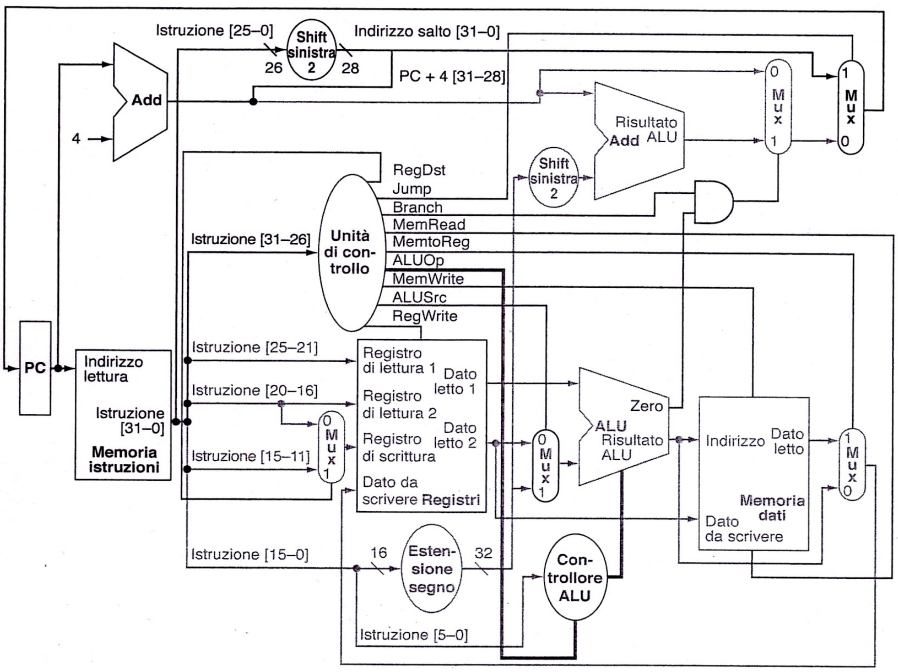
3. [4] Si progetti un circuito caratterizzato da una parola A di 4 bit in ingresso e da un'uscita U che vale "1" se e solo se la parola A è "palindroma", cioè è uguale se letta da sinistra a destra o da destra a sinistra. a) Determinare la tabella di verità delle uscite; b) esprimerle nella forma canonica più adatta; c) semplificarle mediante mappe di Karnaugh; d) semplificarle ulteriormente, se possibile, mediante semplificazioni algebriche; e) disegnarne il circuito.

4. [4] Si disegni il circuito del bistabile Set-Reset asincrono; si determinino e si traccino gli andamenti degli ingressi S e R nel tempo, che sono necessari per generare sull'uscita Q un segnale che cambia valore (0 → 1 → 0 → 1 → …) ogni 2 secondi.

5. [7] Si sintetizzi una macchina a stati finiti di Moore sincrona, caratterizzata da una linea d'ingresso I, che viene osservata ogni secondo, ed una linea di uscita Y, la quale assume il valore assunto dall'ingresso I due secondi prima. Si assuma uno stato iniziale equivalente ad aver ricevuto in ingresso molti '0'. Si determinino: STG, STT, STT codificata e le funzioni uscita e stato prossimo, semplificandole il più possibile. Si disegni infine il circuito complessivo della macchina.

6. [5] Si progetti un circuito "moltiplicatore per 3", con un ingresso da 4 bit, che produce in uscita il numero binario in ingresso (interpretato come intero senza segno) moltiplicato per 3. Si abbia cura di progettare il circuito più semplice possibile, che non presenti inoltre problemi di overflow.

7. [5] Esprimere in esadecimale con il corretto numero di bit, i valori in ingresso e in uscita da: a) ogni ALU e b) dai MUX relativi ai salti, supponendo che la CPU stia in questo momento eseguendo la seguente istruzione:

`0xABC: sw $7, 128($0)`
(OpCode **`sw`**: 44)
Rappresentare inoltre l'istruzione in linguaggio macchina, in formato esadecimale.

Nota: si prega di svolgere questo esercizio sul proprio foglio, disegnando ALU e MUX e scrivendo i valori richiesti in corrispondenza dei rispettivi ingressi/uscite.