Corso di Laurea in Ingegneria Informatica

Calcolatori — a.a. 2022–2023

Scritto del 16 giugno 2023

Cognome e Nome dello studente: _____

Permutazioni e Ritorni Ciclici

Dato un insieme finito e ordinato A, qualsiasi riarrangiamento $\sigma(A)$ dei suoi elementi prende il nome di permutazione*. Ad esempio, una possibile permutazione di $A = \{a_0, a_1, a_2, a_3\}$ è $\sigma(A) = \{a_0, a_2, a_3, a_1\}$. Un'interessante proprietà è che per ogni permutazione esiste sempre un intero r, detto* tempo di ritorno*, tale che applicando per r volte la permutazione si riottiene l'insieme di partenza: $\sigma^r(A) = A$. Il tempo di ritorno per l'esempio precedente è $r = 3$, perché $A = \{a_0, a_1, a_2, a_3\} \xrightarrow{\sigma} \{a_0, a_2, a_3, a_1\} \xrightarrow{\sigma} \{a_0, a_3, a_1, a_2\} \xrightarrow{\sigma} \{a_0, a_1, a_2, a_3\} = A$. Se gli elementi dell'insieme non sono tutti distinti tra loro, il tempo di ritorno può risultare più breve, cioè $r' \leq r$.*

Reti logiche sequenziali Data in ingresso la parola di 16 bit

$$B = b_0 b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 b_9 b_A b_B b_C b_D b_E b_F$$

e la legge di permutazione $\sigma(B) = b_0 b_4 b_1 b_5 b_8 b_C b_9 b_D b_F b_B b_E b_A b_7 b_3 b_6 b_2$, progettare con la tecnica "parte operativa/parte di controllo" una macchina sequenziale di Moore che consenta di calcolare e fornire in uscita il tempo di ritorno r' della permutazione (N.B. $r' < 16$). Simulare il funzionamento della macchina per la parola d'ingresso $B = 1011001000000000$.

Programmazione assembler Dato un vettore `V` di 32 byte contenente gli interi da 0 a 31 in ordine ascendente, scrivere un programma assembler 8086 per calcolare il tempo di ritorno r (N.B. $r < 16$) della permutazione σ che, applicata agli elementi di `V`, lo trasforma in

$$\sigma(\texttt{V}) = 0, 4, 1, 5, 8, 12, 9, 13, 16, 20, 17, 21, 24, 28, 25,$$
$$29, 26, 30, 27, 31, 18, 22, 19, 23, 10, 14, 11, 15, 2, 6, 3, 7.$$

Il programma deve fare uso di una look-up table e di almeno una procedura per costruire (e stampare a video con la macro `display`) tutti i risultati intermedi $\sigma^k(\texttt{V})$, $0 \leq k \leq r$. Simulare il funzionamento del programma con il vettore `V` dato sopra, mostrando in particolare lo stato dello stack prima della chiamata, durante l'esecuzione e al termine della procedura.