

PROVA SCRITTA DI RETI LOGICHE E CALCOLATORI DEL 12/09/2019

ESERCIZIO 1: Si realizzi una rete sequenziale sincrona R con una linea di ingresso x ed una linea di uscita z che riconosce le sequenze **S** sulla linea x della forma **S = 011Q011**, e restituisce 1 in corrispondenza dell'ultimo bit della sequenza se **Q** contiene almeno due bit consecutivi uguali fra loro, 0 altrimenti.

t:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
X:	<u>0</u>	<u>1</u>	<u>1</u>	1	0	0	<u>0</u>	<u>1</u>	<u>1</u>	0	0	1	1	0	1	0	1	0	1	1	0	1	1	1	1	1	0	0	1	1	...
Z:	0	0	0	0	0	0	0	0	<u>1</u>	0	0	0	0	0	0	0	0	0	0	<u>0</u>	0	0	0	0	0	0	0	0	0	<u>1</u>	...
	011			Q			011			011			Q			011			011			Q			011						

Nell'esempio sopra riportato, la prima sequenza **S** è compresa tra t=0 a t=8 e tale che **Q=100**, poiché **Q** contiene due 0 consecutivi, a t=8 la rete restituisce 1. La seconda sequenza **S** è compresa tra t=10 a t=19 e tale che **Q=0101**, poiché **Q** non contiene bit consecutivi uguali fra loro, a t=19 la rete restituisce 0.

ESERCIZIO 2: Estendere il set di istruzioni della macchina ad accumulatore con l'operazione **CSR X**, definita come segue. A partire dalla locazione X+1 della RAM è memorizzato un vettore formato da un numero L di elementi compresi tra 0 e N-1, dove L è il valore contenuto in M[X] e N è il valore contenuto nell'accumulatore.

L'istruzione implementa il *cifrario di Cesare*, cioè un algoritmo di crittografia in cui ciascun carattere di un testo in chiaro è sostituito nel testo cifrato dal carattere che si trova un certo numero di posizioni dopo nell'alfabeto. Nel caso in esame, la chiave di cifratura (ovvero il numero di posizioni di cui occorre traslare ciascun numero del vettore) è **3** e l'alfabeto su cui è costruito il messaggio è rappresentato dai numeri tra 0 e N-1, con N>3.

L'istruzione deve sostituire ciascun elemento del vettore V[i] (con i ∈ {0,1 ... L-1}) con il corrispondente valore cifrato, cioè (V[i]+3) % N.

La figura sulla destra mostra un esempio dello stato della memoria e dei registri prima e dopo l'esecuzione dell'istruzione.

SUGGERIMENTO: Al fine di applicare l'operatore modulo, si noti che vale la seguente relazione:

$$A \% N = \begin{cases} A & \text{se } (A-N) < 0 \\ A-N & \text{se } 0 \leq (A-N) < N \end{cases}$$

PRIMA				DOPO			
X	:			X	:		
1052	L	1052	8	1052	L	1052	8
AC	V[0]	1053	17	AC	V[0]	1053	20
26	V[1]	1054	4	26	V[1]	1054	7
	V[2]	1055	19		V[2]	1055	22
	V[3]	1056	8		V[3]	1056	11
	V[4]	1057	23		V[4]	1057	0
	V[5]	1058	11		V[5]	1058	14
	V[6]	1059	6		V[6]	1059	9
	V[7]	1060	25		V[7]	1060	2
	:				:		

ESERCIZIO 3: Scrivere una procedura assembly che riceve due vettori **V** e **W**, entrambi di word. Siano **nv** e **nw** rispettivamente le lunghezze dei due vettori, con nv > nw. La procedura opera su V le seguenti sostituzioni: V[i] = V[i] + W[i % nw], ∀ i ∈ [0,nv-1].

Scrivere inoltre il programma principale che invoca opportunamente la procedura descritta.

La tabella seguente chiarisce il modo in cui devono avvenire le sostituzioni considerando i vettori V = [17,4,19,10,23,1,6,25] e W = [3,7,10].

PRIMA	17	4	19	10	23	1	6	25
chiave	3	7	10	3	7	10	3	7
DOPO	20	11	29	13	30	11	9	32

L'algoritmo, dunque, sostituisce V[0] con (V[0] + W[0]), V[1] con (V[1] + W[1]), V[2] con (V[2] + W[2]), V[3] con (V[3] + W[0]), V[4] con (V[4] + W[1]), ...

PRIMA			DOPO		
1473	10	W[2]	1473	10	W[2]
1472			1472		
1471	7	W[1]	1471	7	W[1]
1470			1470		
1469	3	W[0]	1469	3	W[0]
1468			1468		
1467	25	V[7]	1467	32	V[7]
1466			1466		
1465	6	V[6]	1465	9	V[6]
1464			1464		
1463	1	V[5]	1463	11	V[5]
1462			1462		
1461	23	V[4]	1461	30	V[4]
1460			1460		
1459	10	V[3]	1459	13	V[3]
1458			1458		
1457	19	V[2]	1457	29	V[2]
1456			1456		
1455	4	V[1]	1455	11	V[1]
1454			1454		
1453	17	V[0]	1453	20	V[0]
1452			1452		

Stato della memoria prima e dopo l'esecuzione della procedura