

# PROVA SCRITTA DI CALCOLATORI ELETTRONICI DEL 10/4/2015

## (Tempo a disposizione: 3 ore)

### ESERCIZIO 1 (Tutti):

Si realizzi una rete sequenziale sincrona R con un ingresso X ed una uscita Z. La rete riceve in cinque istanti di tempo consecutivi due numeri a quattro bit  $\alpha = \alpha_0\alpha_1\alpha_2\alpha_3$  e  $\beta = \beta_0\beta_1\beta_2\beta_3$  sovrapposti di tre bit, quindi  $\alpha_1 = \beta_0$ ,  $\alpha_2 = \beta_1$  e  $\alpha_3 = \beta_2$ .

La rete restituisce in corrispondenza dell'ultimo bit di  $\beta$  il risultato dell'operazione:  $\alpha_0 \cdot \beta_0 + \alpha_1 \cdot \beta_1 + \alpha_2 \cdot \beta_2 + \alpha_3 \cdot \beta_3$  (dove l'operatore "+" indica l'operazione logica OR e l'operatore "." indica l'operazione logica AND) per poi riprendere il suo funzionamento da principio. Segue un esempio di possibile funzionamento di R:

t:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
X:	0	1	0	1	1	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	...
Z:	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	...

La rete riceve a partire dall'istante t=0 i numeri  $\alpha=0101$  e  $\beta=1011$ . L'esito dell'operazione è quindi:  $0 \cdot 1 + 1 \cdot 0 + 0 \cdot 1 + 1 \cdot 1 = 1$  e viene restituito all'istante t=4. A partire dall'istante t=5, riceve i numeri  $\alpha=0001$  e  $\beta=0010$ . L'esito dell'operazione è quindi:  $0 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 + 1 \cdot 0 = 0$  e viene restituito all'istante t=9.

### ESERCIZIO 2 (DM270 – dall'A. A. 2012/13):

Estendere il set di istruzioni della macchina a stack con l'operazione **CountPos**, definita come segue. Data una lista chiusa il cui puntatore di testa è memorizzato in cima allo stack, l'istruzione calcola il numero di elementi positivi o nulli contenuti nella lista ed aggiunge in fondo alla lista un nuovo elemento contenente tale conteggio.

La figura sulla sinistra mostra lo stato della memoria prima (a sinistra) e dopo (a destra) l'esecuzione dell'istruzione. In particolare, il nuovo elemento è memorizzato nelle posizioni di heap 3401 e 3400.

SP	1052	8231	SP	1052	8231
1053	:	:	1053	:	:
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
HP	:	:	HP	3400	-1
3401	:	:	3401	2	2
:	:	:	3399	:	:
4056	-1	4056	3401	3401	3401
4057	4	4057	4	4	4
:	:	:	:	:	:
7500	7503	7500	7503	7503	7503
7501	-7	7501	-7	-7	-7
7502	4057	7502	4057	4057	4057
7503	-2	7503	-2	-2	-2
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
8230	7501	8230	7501	7501	7501
8231	13	8231	13	13	13

### ESERCIZIO 2 (DM270 – A. A. precedenti):

Estendere il set di istruzioni della macchina a registri con l'operazione **CountOpp Ri, Rj, Rk**, definita come segue.

A partire dalle locazioni i cui indirizzi sono memorizzati in  $R_i$  ed  $R_j$ , sono memorizzati due array,  $V_i$  e  $V_j$ , di 32 elementi. Per ogni posizione  $p$  relativa ad array di questo tipo ( $p$  compreso tra 1 e 32), si definisce la posizione complementare  $p'$  pari  $(32-p+1)$ . L'istruzione memorizza nel registro  $R_k$  il numero di posizioni  $p$  tali che l'elemento in posizione  $p$  di  $V_i$  ha segno opposto rispetto all'elemento in posizione complementare  $p'$  di  $V_j$ .

Si consideri il seguente esempio in cui  $R_i=947$ ,  $R_j=1234$  ed i vettori sono, per semplicità, di dimensione 4 anziché 32.

...	947	948	949	950	...	1234	1235	1236	1237	...
...	1	-7	-5	9	...	2	7	-1	-4	...

Al termine dell'esecuzione dell'istruzione, il registro  $R_k$  conterrà il valore 2 (determinato dalle posizioni  $p$  pari a 1 e 3).

### ESERCIZIO 3 (DM 270 – 9CFU/6CFU):

Scrivere una procedura assembly che riceve un vettore di double word  $V$  e un vettore di word  $W$ , entrambi della stessa lunghezza  $n$ . Per ogni posizione  $p$  relativa ad array di questo tipo ( $p$  compreso tra 0 e  $n-1$ ), si definisce la posizione complementare  $p'$  pari  $(n-p-1)$ . La procedura restituisce il numero di posizioni  $p$  tali che l'elemento in posizione  $p$  di  $V$  ha segno opposto rispetto all'elemento in posizione complementare  $p'$  di  $W$ .

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