

“Reti Logiche e Calcolatori” del 14/6/2017 – Traccia A

Esercizio 1 – Si realizzi una rete sequenziale sincrona R con un ingresso X ed una uscita Z. La rete riconosce sequenze del tipo $\alpha_1\alpha_0\beta_2\beta_1\beta_0$. I primi due bit rappresentano il secondo e il terzo bit di una maschera il cui primo bit vale 0. La rete applica la maschera ai bit $\beta_2\beta_1\beta_0$ effettuando l'AND bit a bit e, in corrispondenza del quinto bit ricevuto (ossia β_0), restituisce l'OR tra i bit del risultato. Le sequenze sono sovrapposte di un bit, quindi il quinto bit di una sequenza è anche il primo bit della sequenza successiva.

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

La rete riceve la prima sequenza a partire dall'istante $t=0$. La maschera risulta uguale a 001 mentre i bit $\beta_2\beta_1\beta_0$ valgono 100. L'AND bit a bit dà come risultato 000 e l'OR tra i bit del risultato dà come risultato 0 (restituito all'istante 4). La seconda sequenza viene ricevuta a partire dall'istante $t=4$ (sovrapposta di un bit alla precedente). La maschera risulta pari a 001, i bit $\beta_2\beta_1\beta_0$ valgono 011, l'AND tra 001 e 011 dà come risultato 001 e l'OR tra i bit del risultato vale 1 (restituito all'istante 8).

“Reti Logiche e Calcolatori” del 14/6/2017 – Traccia A

Esercizio 2

Estendere il set di istruzioni della macchina ad accumulatore con l'operazione **BLOCKS X**, definita come segue. A partire dalla locazione X+1 della RAM è memorizzato un vettore V di L elementi, dove L è il valore contenuto in M[X]. Nell'accumulatore è memorizzato un intero K.

Il vettore V contiene tutti elementi positivi e un solo elemento pari a 0. V è logicamente diviso in due blocchi separati dall'elemento pari a 0. Al termine dell'operazione l'accumulatore conterrà 1 se la somma degli elementi in ciascun blocco è maggiore di K, conterrà 0 altrimenti.

PRIMA			DOPO		
X	1051	?	X	1051	?
L	1052	8	L	1052	8
V[0]	1053	3	V[0]	1053	3
AC	1054	4	AC	1054	4
V[1]	1055	3	V[1]	1055	3
V[2]	1056	2	V[2]	1056	2
V[3]	1057	0	V[3]	1057	0
V[4]	1058	8	V[4]	1058	8
V[5]	1059	7	V[5]	1059	7
V[6]	1060	1	V[6]	1060	1
V[7]	1061	?	V[7]	1061	?

“Reti Logiche e Calcolatori” del 14/6/2017 – Traccia A

Esercizio 3 - Scrivere in assembly x86-32 una procedura **BLCSUM** che riceve un vettore di word V e un numero K e restituisce 0 o 1. Il vettore V contiene tutti elementi positivi e alcuni elementi pari a -1. V è logicamente diviso in blocchi, gli elementi pari a -1 identificano la fine dei blocchi. L'ultimo blocco termina con l'ultimo elemento del vettore. La procedura restituisce 1 se la somma degli elementi in ciascun blocco è maggiore di K. Scrivere inoltre un programma principale che invochi opportunamente la procedura descritta.

La figura mostra un esempio di vettore V.

Per K=10 la procedura restituisce il valore 1, poiché la somma degli elementi dei blocchi è rispettivamente 13, 11, 12 e 22.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
V	3	7	3	-1	5	6	-1	2	3	4	2	1	-1	10	11	1
				^			^						^			

“Reti Logiche e Calcolatori” del 14/6/2017 – Traccia B

Esercizio 3 - Scrivere in assembly x86-32 una procedura **BLCNUM** che riceve un vettore di word V e una word X e restituisce un intero N. Una blocco di V è una sequenza di elementi consecutivi terminata dal valore X. L'elemento di valore X non appartiene a nessun blocco. Il primo blocco inizia dal primo elemento del vettore ed ogni altro blocco inizia a partire dall'elemento successivo a quello contenente il valore X. L'ultimo blocco termina con l'ultimo elemento del vettore. La procedura restituisce il numero di elementi N del blocco più lungo presente all'interno di V. Scrivere inoltre un programma principale che invochi opportunamente la procedura descritta.

La figura mostra un esempio di vettore V.

Per X=3 la procedura restituisce il valore N=6, poiché la lunghezza dei blocchi è rispettivamente 3, 6, 2 e 2.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
V	1	10	4	3	1	8	2	1	1	9	3	10	2	3	2	13
				^							^			^		

“Reti Logiche e Calcolatori” del 14/6/2017 – Traccia C

Esercizio 3 - Scrivere in assembly x86-32 una procedura **BLCCNT** che riceve un vettore di word V e un numero K e restituisce un intero N. Una blocco di V è una sequenza minimale di elementi consecutivi la cui somma eccede K. Il primo blocco inizia dal primo elemento del vettore ed ogni altro blocco inizia a partire dall'elemento successivo al blocco precedente. L'ultimo blocco termina con l'ultimo elemento del vettore. La procedura restituisce il numero N di blocchi presenti all'interno di V. Scrivere inoltre un programma principale che invochi opportunamente la procedura descritta.

La figura mostra un esempio di vettore V.

Per K=20 la procedura restituisce il valore N=4, poiché la somma degli elementi dei blocchi è rispettivamente di 21, 29, 21 e 4.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
V	10	9	2	2	6	7	3	11	11	3	1	1	5	2	1	1
			^					^					^			

Esercizio 3 - Scrivere in assembly x86-32 una procedura **BLCSEQ** che riceve un vettore di word V e un numero X e restituisce 0 o 1. Una blocco di V è una sequenza di elementi consecutivi terminata dal valore X. L'elemento di valore X non appartiene a nessun blocco. Il primo blocco inizia dal primo elemento del vettore ed ogni altro blocco inizia a partire dall'elemento successivo a quello contenente il valore X. L'ultimo blocco termina con l'ultimo elemento del vettore. La procedura restituisce 1 solo se la sequenza formata dagli elementi iniziali dei blocchi è strettamente crescente. Scrivere inoltre un programma principale che invochi opportunamente la procedura descritta.

V

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3	7	3	0	5	6	0	8	3	4	2	1	0	10	11	1

^ ^ ^