

PROVA SCRITTA DI RETI LOGICHE E CALCOLATORI DEL 20/06/2016 – 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 $1(10)^n0$, dove n è un numero intero maggiore di 0 e multiplo di 4. Quindi i valori ammissibili per n sono nell'insieme {4,8,12,16,20,...}. Non appena la rete riconosce una sequenza valida, restituisce 1 e riprende il proprio funzionamento dal principio.

Si guardi l'esempio per maggiore chiarezza.

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
X:	0	1	1	0	0	1	0	1	1	0	1	1	1	0	1	0	1	0	1	0	0	0	0	...
Z:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	...

Nell'esempio, la rete riceve la prima sequenza valida a partire dall'istante t=11, infatti in tale istante di tempo la rete riceve la sequenza di start "1", negli istanti da 12 a 19 riceve quattro volte consecutive la sequenza "10" e nell'istante 20 riceve la sequenza di stop "0". Quindi all'istante t=21 riprende il proprio funzionamento. Si noti che la sequenza "1100" ricevuta negli istanti da 1 a 4 non rappresenta una sequenza valida in quanto in questo caso n=1.

PROVA SCRITTA DI RETI LOGICHE E CALCOLATORI DEL 20/06/2016 – TRACCIA B

ESERCIZIO 1:

Si realizzi una rete sequenziale sincrona R con un ingresso X ed una uscita Z. La rete riconosce sequenze del tipo $1(01)^n1$, dove n è un numero intero maggiore di 0 e multiplo di 4. Quindi i valori ammissibili per n sono nell'insieme {4,8,12,16,20,...}. Non appena la rete riconosce una sequenza valida, restituisce 1 e riprende il proprio funzionamento dal principio.

Si guardi l'esempio per maggiore chiarezza.

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
X:	0	1	0	1	1	1	0	1	0	0	1	1	0	1	0	1	0	1	0	1	1	0	0	...
Z:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	...

Nell'esempio, la rete riceve la prima sequenza valida a partire dall'istante t=11, infatti in tale istante di tempo la rete riceve la sequenza di start "1", negli istanti da 12 a 19 riceve quattro volte consecutive la sequenza "01" e nell'istante 20 riceve la sequenza di stop "1". Quindi all'istante t=21 riprende il proprio funzionamento. Si noti che la sequenza "1011" ricevuta negli istanti da 1 a 4 non rappresenta una sequenza valida in quanto in questo caso n=1.

PROVA SCRITTA DI RETI LOGICHE E CALCOLATORI DEL 20/06/2016 – TRACCIA C

ESERCIZIO 1:

Si realizzi una rete sequenziale sincrona R con un ingresso X ed una uscita Z. La rete riconosce sequenze del tipo $0(01)^n1$, dove n è un numero intero maggiore di 0 e multiplo di 4. Quindi i valori ammissibili per n sono nell'insieme {4,8,12,16,20,...}. Non appena la rete riconosce una sequenza valida, restituisce 1 e riprende il proprio funzionamento dal principio.

Si guardi l'esempio per maggiore chiarezza.

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
X:	1	0	0	1	1	0	1	0	0	1	0	0	0	1	0	1	0	1	0	1	1	1	1	...
Z:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	...

Nell'esempio, la rete riceve la prima sequenza valida a partire dall'istante t=11, infatti in tale istante di tempo la rete riceve la sequenza di start "0", negli istanti da 12 a 19 riceve quattro volte consecutive la sequenza "01" e nell'istante 20 riceve la sequenza di stop "1". Quindi all'istante t=21 riprende il proprio funzionamento. Si noti che la sequenza "0011" ricevuta negli istanti da 1 a 4 non rappresenta una sequenza valida in quanto in questo caso n=1.

PROVA SCRITTA DI RETI LOGICHE E CALCOLATORI DEL 20/06/2016 – TRACCIA D

ESERCIZIO 1:

Si realizzi una rete sequenziale sincrona R con un ingresso X ed una uscita Z. La rete riconosce sequenze del tipo $0(10)^n0$, dove n è un numero intero maggiore di 0 e multiplo di 4. Quindi i valori ammissibili per n sono nell'insieme {4,8,12,16,20,...}. Non appena la rete riconosce una sequenza valida, restituisce 1 e riprende il proprio funzionamento dal principio.

Si guardi l'esempio per maggiore chiarezza.

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
X:	1	0	1	0	0	0	1	0	1	1	0	0	1	0	1	0	1	0	1	0	0	1	1	...
Z:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	...

Nell'esempio, la rete riceve la prima sequenza valida a partire dall'istante t=11, infatti in tale istante di tempo la rete riceve la sequenza di start "0", negli istanti da 12 a 19 riceve quattro volte consecutive la sequenza "10" e nell'istante 20 riceve la sequenza di stop "0". Quindi all'istante t=21 riprende il proprio funzionamento. Si noti che la sequenza "0100" ricevuta negli istanti da 1 a 4 non rappresenta una sequenza valida in quanto in questo caso n=1.

PROVA SCRITTA DI RETI LOGICHE E CALCOLATORI DEL 20/06/2016 – TRACCIA A

ESERCIZIO 2: Estendere il set di istruzioni della macchina ad accumulatore con l'operazione **SUMH X**, definita come segue.

A partire dalla locazione $X+1$ della RAM è memorizzato un vettore **V** di **L** elementi, dove **L** è contenuto in $M[X]$ ed è un numero pari.

L'istruzione modificherà il vettore come segue: per ogni elemento $V[i]$ della prima metà del vettore (tale che $i=0, \dots, n/2-1$), $V[i]$ viene posto a $V[i]+V[n/2+i]$ se la condizione $V[i]>V[n/2+i]$ è vera, mentre $V[i]$ viene posto a 0 se la predetta condizione è falsa.

Al termine dell'istruzione la dimensione del vettore memorizzata in $M[X]$ dovrà essere posta uguale a $n/2$ e l'accumulatore dovrà contenere il numero di elementi per cui la condizione è stata soddisfatta.

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

PRIMA				DOPO			
X	:	:	:	X	:	:	:
1052	L	1052	8	1052	L	1052	4
	V[0]	1053	3		V[0]	1053	0
	V[1]	1054	6	AC	V[1]	1054	8
	V[2]	1055	9	2	V[2]	1055	0
	V[3]	1056	12		V[3]	1056	15
	V[4]	1057	8			1057	8
	V[5]	1058	2			1058	2
	V[6]	1059	11			1059	11
	V[7]	1060	3			1060	3
	:	:	:		:	:	:

PROVA SCRITTA DI RETI LOGICHE E CALCOLATORI DEL 20/06/2016 – TRACCIA B

ESERCIZIO 2: Estendere il set di istruzioni della macchina ad accumulatore con l'operazione **SUMH X**, definita come segue.

A partire dalla locazione $X+1$ della RAM è memorizzato un vettore **V** di **L** elementi, dove **L** è contenuto in $M[X]$ ed è un numero pari.

L'istruzione modificherà il vettore come segue: per ogni elemento $V[i]$ della prima metà del vettore (tale che $i=0, \dots, n/2-1$), $V[i]$ viene posto a $V[i]+V[n/2+i]$ se $V[i]$ e $V[n/2+i]$ sono entrambi negativi, mentre $V[i]$ viene posto a 0 altrimenti.

Al termine dell'istruzione la dimensione del vettore memorizzata in $M[X]$ dovrà essere posta uguale a $n/2$ e l'accumulatore dovrà contenere il numero di elementi per cui la condizione è stata soddisfatta.

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

PRIMA				DOPO			
X	:	:	:	X	:	:	:
1052	L	1052	8	1052	L	1052	4
	V[0]	1053	-3		V[0]	1053	0
	V[1]	1054	-6	AC	V[1]	1054	-8
	V[2]	1055	9	2	V[2]	1055	0
	V[3]	1056	-12		V[3]	1056	-15
	V[4]	1057	8			1057	8
	V[5]	1058	-2			1058	2
	V[6]	1059	11			1059	11
	V[7]	1060	-3			1060	3
	:	:	:		:	:	:

PROVA SCRITTA DI RETI LOGICHE E CALCOLATORI DEL 20/06/2016 – TRACCIA A

ESERCIZIO 3: Scrivere una procedura assembly che riceve due vettori **V** e **W** composti entrambi da n elementi, con n pari, e modifica l'array **W** come di seguito specificato:

- per ogni elemento **W**[i] della prima metà del vettore (tale che $i=0, \dots, n/2-1$), **W**[i] viene posto a **W**[i]+**V**[$n/2+i$] se la condizione **W**[i]**>****V**[$n/2+i$] è vera, mentre **W**[i] viene posto a 0 se la predetta condizione è falsa.
- per ogni elemento **W**[i] della seconda metà del vettore (tale che $i=n/2, \dots, n-1$), **W**[i] viene posto a **W**[i]+**V**[$i-n/2$] se la condizione **W**[i]**>****V**[$i-n/2$] è vera, mentre **W**[i] viene posto a 0 se la predetta condizione è falsa.

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

La figura sulla destra mostra un esempio dello stato della memoria assumendo che l'indirizzo di partenza del vettore **V** sia 1052, l'indirizzo di partenza del vettore **W** sia 1072 e la lunghezza dei vettori sia uguale a 10.

PRIMA						DOPO					
1071	11	V[9]	1091	12	W[9]	1071	11	V[9]	1091	22	W[9]
1070			1090			1070			1090		
1069	4	V[8]	1089	10	W[8]	1069	4	V[8]	1089	14	W[8]
1068			1088			1068			1088		
1067	2	V[7]	1087	2	W[7]	1067	2	V[7]	1087	0	W[7]
1066			1086			1066			1086		
1065	1	V[6]	1085	3	W[6]	1065	1	V[6]	1085	0	W[6]
1064			1084			1064			1084		
1063	9	V[5]	1083	5	W[5]	1063	9	V[5]	1083	7	W[5]
1062			1082			1062			1082		
1061	10	V[4]	1081	9	W[4]	1061	10	V[4]	1081	0	W[4]
1060			1080			1060			1080		
1059	4	V[3]	1079	5	W[3]	1059	4	V[3]	1079	9	W[3]
1058			1078			1058			1078		
1057	9	V[2]	1077	6	W[2]	1057	9	V[2]	1077	8	W[2]
1056			1076			1056			1076		
1055	6	V[1]	1075	3	W[1]	1055	6	V[1]	1075	4	W[1]
1054			1074			1054			1074		
1053	2	V[0]	1073	8	W[0]	1053	2	V[0]	1073	0	W[0]
1052			1072			1052			1072		

PROVA SCRITTA DI RETI LOGICHE E CALCOLATORI DEL 20/06/2016 – TRACCIA B

ESERCIZIO 3: Scrivere una procedura assembly che riceve due vettori **V** e **W** composti entrambi da n elementi, con n pari, e modifica l'array **W** come di seguito specificato:

- per ogni elemento **V**[i] della prima metà del vettore (tale che $i=0, \dots, n/2-1$), **W**[$i+n/2$] viene posto a **V**[i]-**W**[$i+n/2$] se la condizione **W**[$i+n/2$]**<****V**[i] è vera, mentre **W**[$i+n/2$] viene posto a 0 se la condizione è falsa.
- per ogni elemento **V**[i] della seconda metà del vettore (tale che $i=n/2, \dots, n-1$), **W**[$i-n/2$] viene posto a **V**[i]+**W**[$i-n/2$] se la condizione **V**[i]**>****W**[$i-n/2$] è vera, mentre **W**[$i-n/2$] viene posto a 0 se la condizione è falsa.

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

La figura sulla destra mostra un esempio dello stato della memoria assumendo che l'indirizzo di partenza del vettore **V** sia 1052, l'indirizzo di partenza del vettore **W** sia 1072 e la lunghezza dei vettori sia uguale a 10.

PRIMA						DOPO					
1071	11	V[9]	1091	12	W[9]	1071	11	V[9]	1091	0	W[9]
1070			1090			1070			1090		
1069	4	V[8]	1089	10	W[8]	1069	4	V[8]	1089	0	W[8]
1068			1088			1068			1088		
1067	2	V[7]	1087	2	W[7]	1067	2	V[7]	1087	7	W[7]
1066			1086			1066			1086		
1065	1	V[6]	1085	3	W[6]	1065	1	V[6]	1085	3	W[6]
1064			1084			1064			1084		
1063	9	V[5]	1083	5	W[5]	1063	9	V[5]	1083	0	W[5]
1062			1082			1062			1082		
1061	10	V[4]	1081	9	W[4]	1061	10	V[4]	1081	2	W[4]
1060			1080			1060			1080		
1059	4	V[3]	1079	5	W[3]	1059	4	V[3]	1079	0	W[3]
1058			1078			1058			1078		
1057	9	V[2]	1077	6	W[2]	1057	9	V[2]	1077	0	W[2]
1056			1076			1056			1076		
1055	6	V[1]	1075	3	W[1]	1055	6	V[1]	1075	0	W[1]
1054			1074			1054			1074		
1053	2	V[0]	1073	8	W[0]	1053	2	V[0]	1073	1	W[0]
1052			1072			1052			1072		

PROVA SCRITTA DI RETI LOGICHE E CALCOLATORI DEL 20/06/2016 – TRACCIA C

ESERCIZIO 3: Scrivere una procedura assembly che riceve due vettori **V** e **W** composti entrambi da n elementi, con n pari, e modifica l'array **W** come di seguito specificato:

a) per ogni elemento **W**[i] della prima metà del vettore (tale che $i=0, \dots, n/2-1$), **W**[i] viene posto a **W**[i]+**V**[$n/2+i$] se la condizione **W**[i] <0 è vera, mentre **W**[i] viene posto a 0 se la predetta condizione è falsa.

b) per ogni elemento **W**[i] della seconda metà del vettore (tale che $i=n/2, \dots, n-1$), **W**[i] viene posto a **W**[i]+**V**[$i-n/2$] se la condizione **W**[i] <0 è vera, mentre **W**[i] viene posto a 0 se la predetta condizione è falsa.

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

La figura sulla destra mostra un esempio dello stato della memoria assumendo che l'indirizzo di partenza del vettore **V** sia 1052, l'indirizzo di partenza del vettore **W** sia 1072 e la lunghezza dei vettori sia uguale a 10.

PRIMA					DOPO						
1071	11	V[9]	1091	-12	W[9]	1071	11	V[9]	1091	-2	W[9]
1070			1090			1070			1090		
1069	4	V[8]	1089	10	W[8]	1069	4	V[8]	1089	0	W[8]
1068			1088			1068			1088		
1067	2	V[7]	1087	-2	W[7]	1067	2	V[7]	1087	7	W[7]
1066			1086			1066			1086		
1065	1	V[6]	1085	3	W[6]	1065	1	V[6]	1085	0	W[6]
1064			1084			1064			1084		
1063	9	V[5]	1083	5	W[5]	1063	9	V[5]	1083	0	W[5]
1062			1082			1062			1082		
1061	10	V[4]	1081	-9	W[4]	1061	10	V[4]	1081	2	W[4]
1060			1080			1060			1080		
1059	4	V[3]	1079	-5	W[3]	1059	4	V[3]	1079	-1	W[3]
1058			1078			1058			1078		
1057	9	V[2]	1077	-6	W[2]	1057	9	V[2]	1077	-4	W[2]
1056			1076			1056			1076		
1055	6	V[1]	1075	3	W[1]	1055	6	V[1]	1075	0	W[1]
1054			1074			1054			1074		
1053	2	V[0]	1073	-8	W[0]	1053	2	V[0]	1073	1	W[0]
1052			1072			1052			1072		

PROVA SCRITTA DI RETI LOGICHE E CALCOLATORI DEL 20/06/2016 – TRACCIA D

ESERCIZIO 3: Scrivere una procedura assembly che riceve due vettori **V** e **W** composti entrambi da n elementi, con n pari, e modifica l'array **W** come di seguito specificato:

a) per ogni elemento **V**[i] della prima metà del vettore (tale che $i=0, \dots, n/2-1$), **W**[$i+n/2$] viene posto a **W**[$i+n/2$]-**V**[i] se la condizione **W**[$i+n/2$] >0 è vera, mentre **W**[$i+n/2$] viene posto a 0 se la predetta condizione è falsa.

b) per ogni elemento **V**[i] della seconda metà del vettore (tale che $i=n/2, \dots, n-1$), **W**[$i-n/2$] viene posto a **W**[$i-n/2$]-**V**[i] se la condizione **W**[$i-n/2$] >0 è vera, mentre **W**[$i-n/2$] viene posto a 0 se la predetta condizione è falsa.

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

La figura sulla destra mostra un esempio dello stato della memoria assumendo che l'indirizzo di partenza del vettore **V** sia 1052, l'indirizzo di partenza del vettore **W** sia 1072 e la lunghezza dei vettori sia uguale a 10.

PRIMA					DOPO						
1071	11	V[9]	1091	-12	W[9]	1071	11	V[9]	1091	0	W[9]
1070			1090			1070			1090		
1069	4	V[8]	1089	10	W[8]	1069	4	V[8]	1089	6	W[8]
1068			1088			1068			1088		
1067	2	V[7]	1087	-2	W[7]	1067	2	V[7]	1087	0	W[7]
1066			1086			1066			1086		
1065	1	V[6]	1085	3	W[6]	1065	1	V[6]	1085	-3	W[6]
1064			1084			1064			1084		
1063	9	V[5]	1083	5	W[5]	1063	9	V[5]	1083	3	W[5]
1062			1082			1062			1082		
1061	10	V[4]	1081	-9	W[4]	1061	10	V[4]	1081	0	W[4]
1060			1080			1060			1080		
1059	4	V[3]	1079	-5	W[3]	1059	4	V[3]	1079	0	W[3]
1058			1078			1058			1078		
1057	9	V[2]	1077	-6	W[2]	1057	9	V[2]	1077	0	W[2]
1056			1076			1056			1076		
1055	6	V[1]	1075	3	W[1]	1055	6	V[1]	1075	2	W[1]
1054			1074			1054			1074		
1053	2	V[0]	1073	-8	W[0]	1053	2	V[0]	1073	0	W[0]
1052			1072			1052			1072		