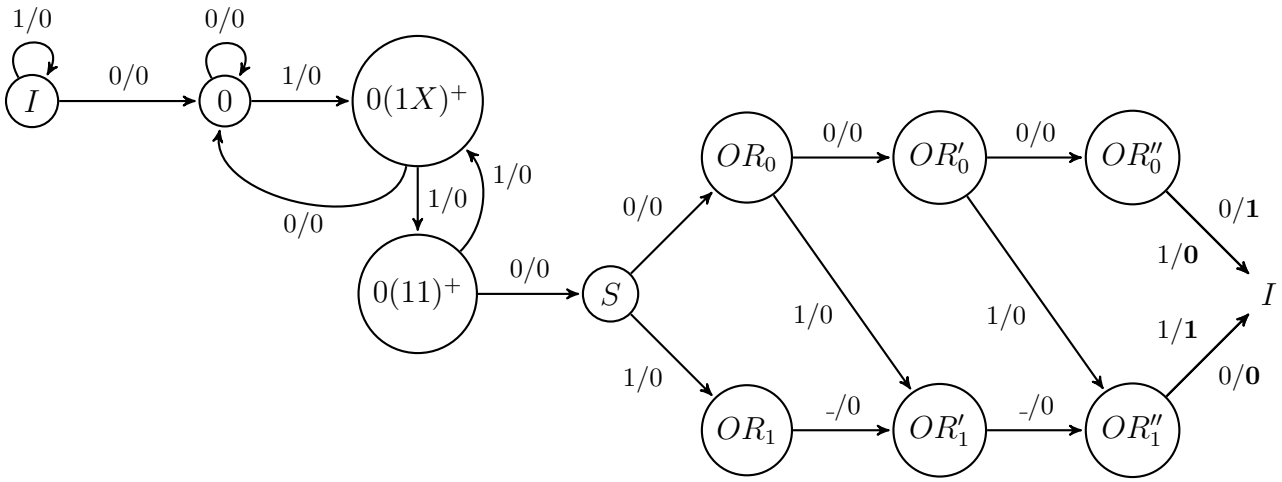
Appello Reti Logiche e Calcolatori 22/06/2021

Esercizio 1

Si realizzi una rete sequenziale sincrona R con un ingresso **X** ed una uscita **Z**. La rete riconosce come valide espressioni del tipo **Q** = $0(11)^+$0b0 b1 b2 a, e restituisce 1 in corrispondenza dell'ultimo bit se a = b0 or b1 or b2, altrimenti restituisce 0. Si noti che l'espressione $(11)^+$ indica che la sequenza 11 è ripetuta una o più volte. Dopo aver riconosciuto una sequenza valida la rete riprende il funzionamento dal principio. Segue un esempio di 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	21	22
$X(t)$:	1	0	1	1	0	0	1	0	0	0	1	0	1	1	1	1	0	0	0	0	0	1	...
$Z(t)$:	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0	0	0	0	**1**	0	...

Nell'esempio riportato, la prima sequenza **Q** è compresa tra t=1 e t=8, l'or tra b0, b1, b2 è pari a 1, siccome a è pari a 0 la rete restituirà 0 dato che la relazione non è soddisfatta. La seconda sequenza **Q** è compresa tra t=11 e t=20, l'or tra b0, b1, b2 è pari a 0, siccome a è pari a 0 la rete restituirà 1 in quanto la relazione è soddisfatta.



Esercizio 2

Estendere il set di istruzioni della macchina ad accumulatore con l'operazione **BUBBLE X**, definita come segue. A partire dall'indirizzo $X + 1$ è presente un vettore **V** la cui dimensione **L** è specificata nella locazione **X**. La funzione modifica il vettore **V** come segue: per ogni coppia $(\mathbf{V}[i], \mathbf{V}[i+1])$ tale che $V[i] > V[i+1]$, la funzione scambia gli elementi della coppia. Al termine della sua esecuzione, la funzione restituisce nell'accumulatore il numero di scambi effettuati.

PRIMA					**DOPO**			
X		..			**X**		..	
1153	**L**	1153	7		1153	**L**	1153	6
	V[0]	1154	3			**V[0]**	1154	3
AC	**V[1]**	1155	7		**AC**	**V[1]**	1155	5
	V[2]	1156	5		3	**V[2]**	1156	7
	V[3]	1157	10			**V[3]**	1157	6
	V[4]	1158	6			**V[4]**	1158	-1
	V[5]	1159	-1			**V[5]**	1159	10
	V[6]	1160	11			**V[6]**	1160	11
		..				..		

La figura mostra un esempio dello stato della memoria e del registro AC prima e dopo l'esecuzione della funzione. La prima coppia di elementi è (3,7), con 3 ¡ 7, perciò non è necessario effettuare alcuno scambio. A seguito dell'analisi della seconda coppia (7,5), invece, il vettore diventa V = {3,5,7,10,6,-1,11}. La nuova coppia da considerare, allora, è (7,10), ma non occorre eseguire alcuno scambio. Successivamente la funzione analizza (10,6) ed effettua lo scambio; segue la valutazione di (10,-1), che comporta un ulteriore scambio, e (10,11) per la quale non si ha alcuno scambio. Complessivamente il numero di scambi effettuati è pari a 3.

$\boldsymbol{\mu_1}: IR_x \rightarrow MAR, 0 \rightarrow T2;$
$\boldsymbol{\mu_2}: M[MAR] \rightarrow MBR, INCR(MAR) \rightarrow MAR;$
$\boldsymbol{\mu_3}: MBR \rightarrow T1, M[MAR] \rightarrow MBR;$
$\boldsymbol{\mu_4}: INCR(MAR) \rightarrow MAR, DECR(T1) \rightarrow T1;$
c: if $OR(T2) == 1$ **then**
  $\boldsymbol{\mu_5}: MBR \rightarrow B, M[MAR] \rightarrow MBR;$
  $\boldsymbol{\mu_6}: MBR \rightarrow A;$
  $\boldsymbol{\mu_7}: A - B \rightarrow A;$
  if $A_{31} == 1$ **then**
    $\boldsymbol{\mu_8}: DECR(MAR) \rightarrow MAR;$
    $\boldsymbol{\mu_9}: MBR \rightarrow M[MAR], B \rightarrow MBR, INCR(MAR) \rightarrow MAR;$
    $\boldsymbol{\mu_{10}}: MBR \rightarrow M[MAR], INCR(T2) \rightarrow T2, INCR(MAR) \rightarrow MAR, DECR(T1) \rightarrow T1, goto\ c;$
  else
    $\boldsymbol{\mu_4}: INCR(MAR) \rightarrow MAR, DECR(T1) \rightarrow T1, goto\ c;$
  end
else
  $\boldsymbol{\mu_{11}}: T2 \rightarrow AC;$
end

Esercizio 3

Scrivere una procedura assembly che riceve un vettore di word V di lunghezza n e lo modifica come descritto di seguito. Per ciascuna coppia $(V[i], V[i+1])$ tale che $V[i] > 4 \cdot V[i+1]$, la procedura scambia i due elementi della coppia. Al termine della sua esecuzione, la procedura restituisce il numero di scambi effettuati.

```
%include "utils.nasm"
section .data
        v dw -2, 5, 1, 3, -1, 1, -6
        n equ ($-v)/2

section .bss
        cnt resd 1

section .text
global _start:
extern proc
_start:
        PUSH    v
        PUSH    dword n
        PUSH    cnt
        CALL    proc
        ;stampa di v
        XOR     ESI, ESI
c:      CMP     ESI, n
        JGE     stampa_cnt
        printw  [v+ESI*2]
        INC     ESI
        JMP     c
        ;stampa conteggio
stampa_cnt:
        printd  dword [cnt]
        exit 0
```

```
section .data
        v   equ 16
        n   equ 12
        cnt equ 8
section .text
global proc
proc:
        PUSH    EBP
        MOV     EBP,ESP
        PUSHAD
        MOV     EAX,[EBP + v]
        MOV     EDI,[EBP + n]
        XOR     ESI, ESI        ;conteggio scambi
        MOV     BX, [EAX]       ;BX <- v[0]
ciclo:
        CMP     EDI, 1          ;ho n-1 elementi da verificare
        JLE     esci
        MOV     CX, [EAX + 2]   ;CX <- v[i+1]
        MOV     DX, CX          ;copio CX per usarlo nello scambio
        SAL     CX, 2           ;CX <- v[i+1] * 4
        CMP     BX, CX
        JLE     avanti
        MOV     [EAX], DX       ;scambio
        MOV     [EAX+2], BX
        INC     ESI
avanti:
        MOV     BX, DX          ;BX <- DX
        ADD     EAX, 2
```

```
        DEC     EDI
        JMP     ciclo
esci:
        MOV     EAX, [EBP+cnt]  ;salvo il valore del conteggio
        MOV     [EAX], ESI
        POPAD
        POP     EBP
        RET     12
```
