07 Aprile 2005

mercoledì 3 giugno 2020 18:34

PRIMO ESX - TRACCIA A

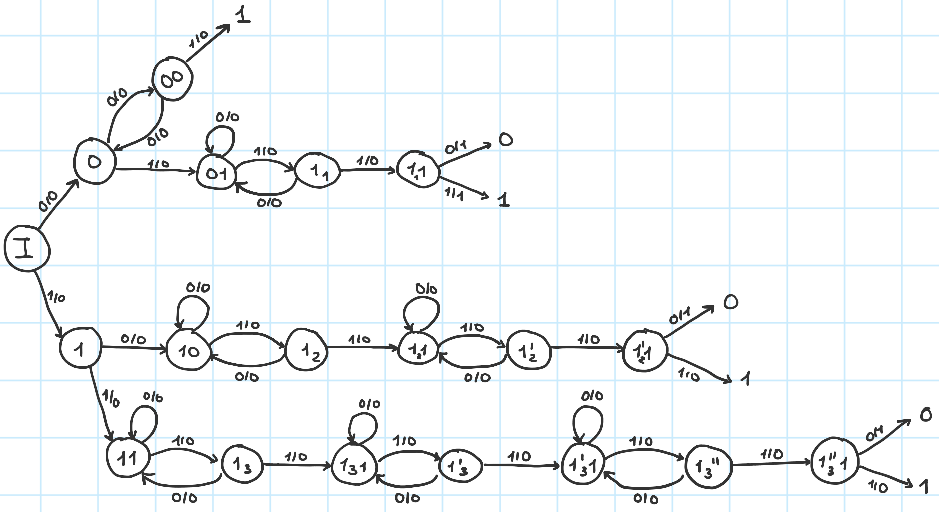
ESERCIZIO 1:
Progettare una rete R sequenziale con una linea di ingresso x ed una linea di uscita z. Ad ogni colpo di clock, R riceve un bit sulla linea x. I primi due bit bit b_1 e b_0 ricevuti sulla linea x indicano alla rete R quante volte riconoscere la sottosequenza **11**. Al termine del riconoscimento la rete restituisce 1 sulla linea z e riconosce una nuova sequenza. Si noti che, nel caso in cui sia b_1 che b_0 sono uguali a zero, la rete non dovrà riconoscere alcuna sottosequenza e quindi leggere una nuova coppia b_1 e b_0.

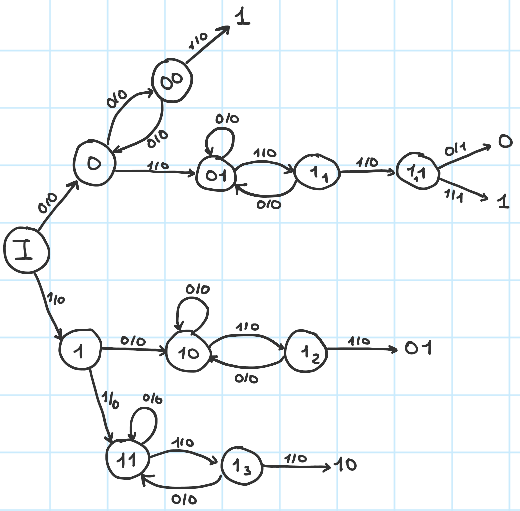
Segue un esempio di possibile funzionamento di R:

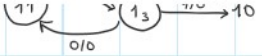
t:	0	1	2	3	4	5	6	7	8
x(t):	1	0	0	1	1	1	0	1	1
z(t):	0	0	0	0	0	0	0	0	1

In questo caso i primi due bit in ingresso ad R sono b_1=1 e b_0=0 e formano il numero b_1b_0=10=2. Quindi la rete restituisce 1 nel momento in cui riceve in ingresso esattamente 2 volte la sequenza 11. All'istante t=9, la rete inizia a riconoscere una nuova coppia di bit b_1 e b_0 e le nuove sottosequenze di 11.

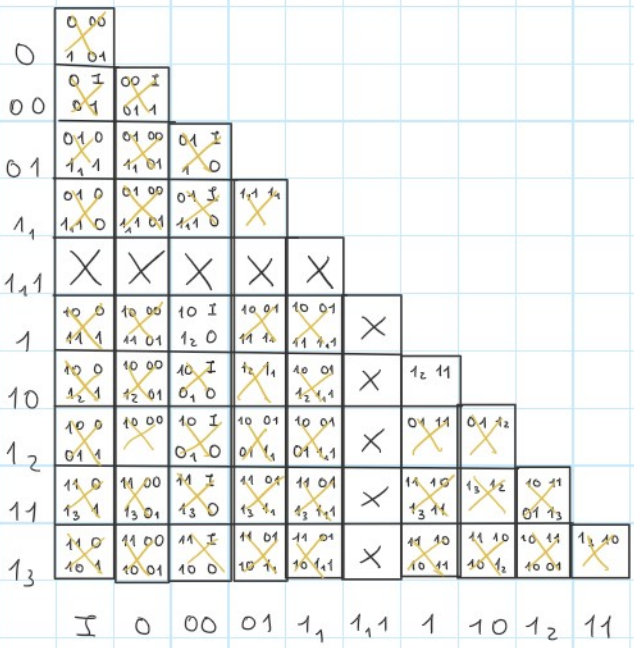
0 0 → Legge una nuova coppia
0 1 → 1 coppia
1 0 → 2 coppie
1 1 → 3 coppie







	0	1
I	0,0	1,0
0	00,0	01,0
00	I,0	0,0
01	01,0	1₁,0
1₁	01,0	1₁1,0
1₁1	0,1	1,1
1	10,0	11,0
10	10,0	1₂,0
1₂	10,0	01,0
11	11,0	1₃,0
1₃	11,0	10,0



SECONDO ESX – TRACCIA A

ESERCIZIO 2:
Estendere il set di istruzioni della macchina ad accumulatore con l'operazione CNTVm X che, dati due vettori V e W di 32 elementi, posti in RAM a partire rispettivamente dagli indirizzi memorizzati nelle locazioni M[X] e M[X+1], restituisce nell'accumulatore il numero di coppie di elementi V[i] e W[i] ($0 \le i \le 31$) che occupano la stessa posizione nei rispettivi vettori e tali che V[i] < W[i].

CNTVm X

V → ha indirizzo M[X] => devo mettere in T_1 la sua lunghezza

W → ha indirizzo M[X+1] => la sua lunghezza decrementa con T_1

In T_2 vado a mettere il conteggio delle coppie tali che V[i] < W[i]

Cambiamenti all'architettura: aggiungo due registri a 24 bit (ad incremento) che contengono i due indirizzi. Aggiungo a T_1 nel codice 11 l'implementazione a 32 e 0 ad AC con un segnale. Aggiungo un segnale di azzeramento anche ad AC e anche di incremento

```
IRx ⟶ MAR, 32 ⟶ T1 , 0 ⟶ Ac ;
M[MAR] ⟶ MBR, INC(MAR) ⟶ MAR ;
MBR ⟶ IND1, M[MAR] ⟶ MBR;
MBR ⟶ IND2 ;
a: if OR(T1) == '1' :
    then IND1 ⟶ MAR, INC(IND1) ⟶ IND1 ;
         M[MAR] ⟶ MBR, IND2 ⟶ MAR , INC(IND2) ⟶ IND2;
         M[MAR] ⟶ MBR, MBR ⟶ A ;
         MBR ⟶ B;
         A-B ⟶ A;
         if A31 == '1' :
             then INC(AC) ⟶ T2 ;
                  DEC(T1) ⟶ T1, goto a;
             else
                  DEC(T1) ⟶ T1, goto a;
         fi
    else
         ϕ
fi
```