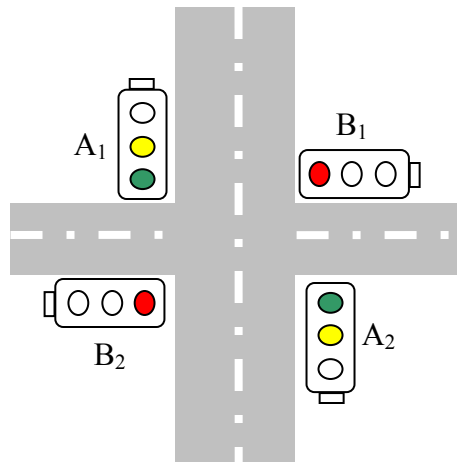


Esercitazione del 06/04/2006 - Soluzioni

1. Automi a stati finiti: introduzione

Supponiamo di avere un sistema che si può trovare in uno stato appartenente ad un insieme finito di stati possibili.

Ex: Immaginiamo un incrocio tra due strade regolate tramite semafori.



Un semaforo può trovarsi in uno dei seguenti stati (consideriamo per semplicità una sola coppia di semafori, ex. A_1 e A_2):

$$S = \{\text{rosso, verde, giallo, lampeggiante}\}$$

Supponiamo che il nostro sistema sia dotato di ingressi che condizionano l'evoluzione del sistema stesso. Supponiamo che l'insieme dei possibili valori di ingresso sia anch'esso finito e supponiamo che il sistema, a determinati istanti di tempo, cambi di stato in funzione degli ingressi e dello stato corrente.

Ex: Il semaforo dell'esempio precedente potrebbe avere in ingresso un sensore che lo informa se è giorno o notte in modo da cambiare il suo stato in lampeggiante quando è notte e si presuppone che il traffico sia così esiguo da rendere il controllo semaforico superfluo. In ingresso al sistema quindi possiamo avere due segnali possibili:

$$I = \{\text{giorno, notte}\}$$

Durante il giorno il semaforo segue la usuale sequenza di stati mentre durante la notte il semaforo resta lampeggiante. La transizione giorno-notte, cioè la transizione tra sequenza normale e lampeggiante, supponiamo avvenga la prima volta che il semaforo transita per il rosso durante la notte mentre la transizione notte-giorno porta immediatamente lo stato lampeggiante nello stato verde.

L'insieme delle possibili transizioni sarà una funzione tra tutti i possibili ingressi e tutti possibili stati verso l'insieme degli stati:

giorno e verde \rightarrow giallo
 giorno e giallo \rightarrow rosso
 giorno e rosso \rightarrow verde
 giorno e lampeggiante \rightarrow verde # questa transizione avviene solo la
 # prima volta che il sensore rileva
 # il giorno
 notte e verde \rightarrow giallo # Quando il sensore rileva che
 notte e giallo \rightarrow rosso # è giunta la notte, attende il rosso
 notte e rosso \rightarrow lampeggiante # quindi passa a lampeggiante
 notte e lampeggiante \rightarrow lampeggiante # e rimane lampeggiante fino alla
 # alla prossima transizione
 # notte-giorno

Supponiamo infine che il sistema, in funzione dello stato corrente, generi uno di un insieme finito di possibili segnali di uscita.

Ex: L'insieme dei possibili segnali in uscita del sistema semaforico sono i seguenti:

$$O = \{\text{luce_rossa}, \text{luce_verde}, \text{luce_gialla}, \text{luce_lampeggiante}\}$$

La funzione che associa gli stati ai segnali di uscita è banalmente la seguente:

verde \rightarrow luce_verde
 rosso \rightarrow luce_rossa
 giallo \rightarrow luce_gialla
 lampeggiante \rightarrow luce_lampeggiante

Sistema di questo tipo sono detti **Automa a Stati Finiti** (Finite State Automata), e può essere descritto formalmente con una 5-pla:

$$\mathbf{M} = \langle \mathbf{S}, \mathbf{I}, \mathbf{O}, \delta, \lambda \rangle$$

dove:

- **S** è l'insieme finito degli stati possibili
- **I** è l'insieme possibile dei valori in ingresso
- **O** è l'insieme dei possibili valori in uscita
- δ è la funzione che lega ingressi e stato corrente con lo stato prossimo
- λ è la funzione che lega lo stato e l'ingresso correnti alle uscite

L'esempio precedente descrive un sistema in cui l'uscita dipende solo dallo stato corrente; un automa di questo tipo è detto **Automa di Moore**. Se l'uscita dipende sia dallo stato corrente che dagli ingressi, come enunciato nella definizione precedente, allora è detto **Automa di Mealy**. Nell'automa di *Moore* le uscite sono collegate allo stato corrente del sistema, nell'automa di *Mealy* sono collegate alle transizioni da uno stato ad un altro. I due automi sono equivalenti, cioè un sistema che si può realizzare con un dei due modelli può essere sempre realizzato anche con l'altro modello. In generale, gli

automi di Mealy hanno meno stati del corrispondente automa di Moore, anche se sono più complicati da sintetizzare.

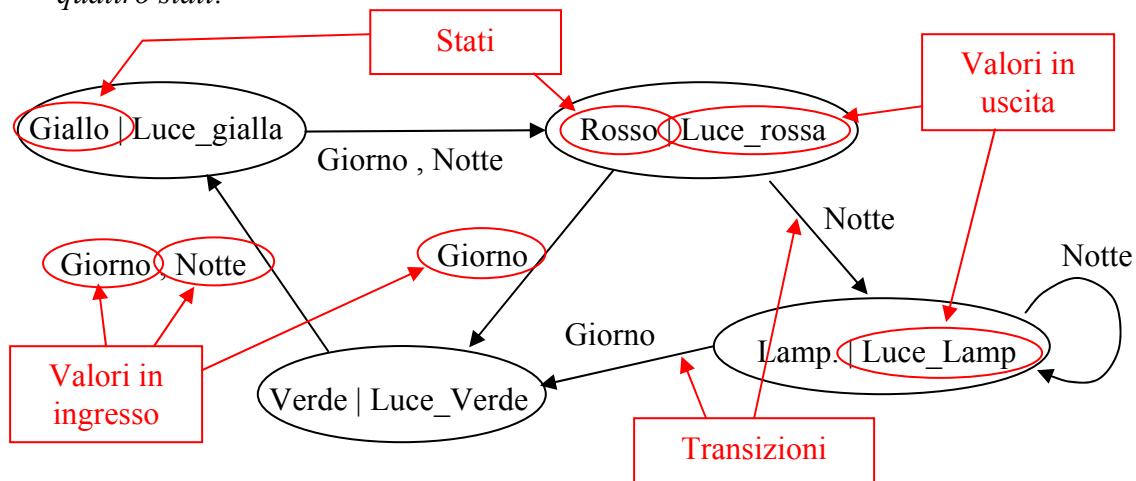
A volte può essere utile indicare lo stato iniziale s_0 dell'automata (ad esempio lo stato che assume il sistema all'atto dell'accensione o in risposta ad un segnale di reset). In questo caso il modello teorico diventa una 6-pla:

$$M = \langle S, I, O, \delta, \lambda, s_0 \rangle$$

2. Grafo di transizione degli stati (STG -State Transition Graph)

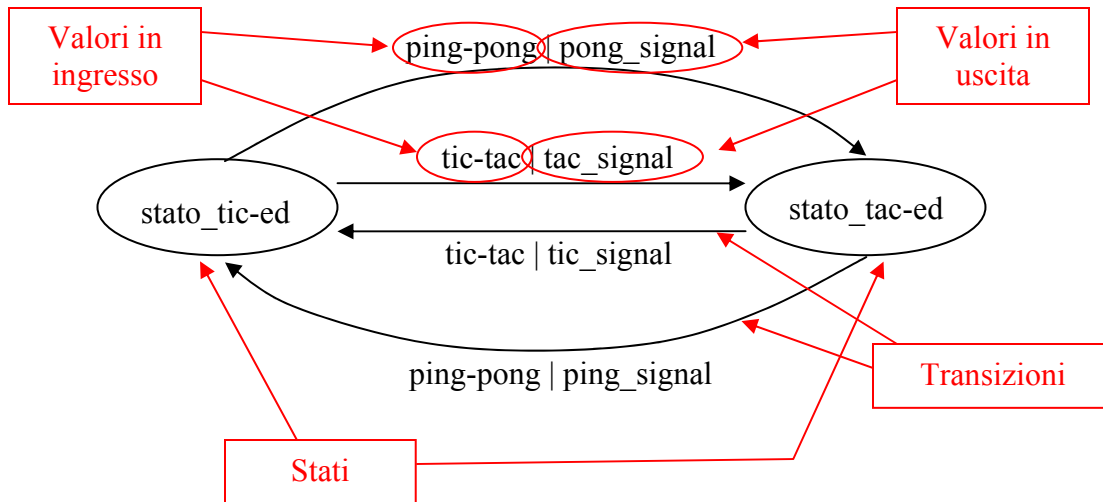
Il **Grafo di Transizione degli Stati** permette di descrivere completamente un automa a stati finiti attraverso un grafo orientato ed etichettato. Ad ogni stato corrisponde un nodo del grafo. Se è possibile una transizione da uno stato A ad uno stato B in relazione ad un ingresso i allora sul grafo esisterà un arco orientato dal nodo corrispondente ad A al nodo corrispondente a B , etichettato con i . Nel caso di automi di Moore in cui l'uscita è funzione del solo stato corrente, l'indicazione dell'uscita u corrispondente ad uno stato A è riportata all'interno del nodo che identifica lo stato A .

Ex: L'esempio precedente può essere descritto tramite un automa di Moore a quattro stati:



Nel caso di un automa di Mealy in cui l'uscita è generata in corrispondenza delle transizioni da uno stato ad un altro, ogni arco (e quindi ogni transizione) viene etichettato, oltre che con l'ingresso che guida la transizione, anche con l'uscita corrispondente.

Ex: Supponiamo di avere un metronomo che genera un segnale bistabile e supponiamo che sia possibile scegliere, attraverso un selettore, il tipo di segnale in uscita: tic-tac oppure ping-pong. Un sistema del genere è descrivibile con un automa di Mealy a due stati e un ingresso binario. L' STG corrispondente è:



3. Tabella di transizione degli stati (STT - State Transition Table)

Un altro modo per descrivere completamente un automa a stati finiti è dato dalla **Tabella di Transizione degli Stati** che riporta in forma tabellare la funzione di transizione δ e la funzione di uscita λ . Nel caso di un automa di *Moore* per ogni stato corrente possibile e per ogni combinazione degli ingressi viene indicato lo stato prossimo raggiunto dall'automato. Un'ulteriore colonna della tabella esprime infine la relazione tra stato del sistema e uscita corrispondente.

Ex: STT relativa al semaforo degli esempi precedenti:

Stato \ Ingresso	δ		λ Uscita
	giorno	notte	
rosso	verde	lampeggiante	luce_rossa
verde	giallo	giallo	luce_verde
giallo	rosso	rosso	luce_gialla
lampeggiante	verde	lampeggiante	luce_lamp.

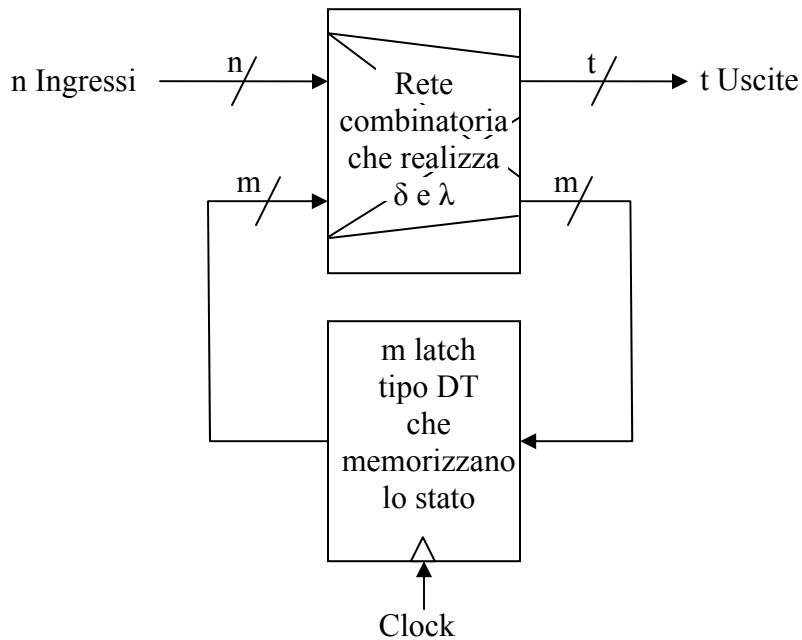
Nel caso di un automa di *Mealy* per ogni stato corrente possibile e per ogni combinazione degli ingressi viene indicato sia lo stato prossimo raggiunto dall'automato che l'uscita corrispondente.

Ex: STT relativa all'esempio del metronomo:

Stato \ Ingresso	$\delta \lambda$	
	ping-pong	tic-tac
stato_tic-ed	stato_tac-ed ping_signal	stato_tac-ed tac_signal
stato_tac-ed	stato_tic-ed pong_signal	stato_tic-ed pong_signal

4. Reti sequenziali sincrone: modello di Huffman.

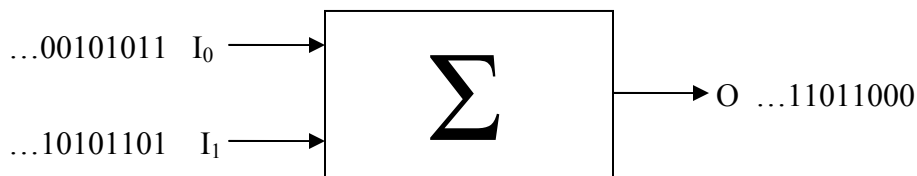
Una **rete combinatoria** è un sistema in cui uscita è univocamente determinata dallo stato degli ingressi, indipendentemente dai valori degli ingressi agli istanti precedenti (sistemi *stateless*). Se l'uscita del sistema al contrario dipende anche dall'evoluzione nel passato degli ingressi allora il sistema è detto **rete sequenziale** (sistemi *statefull*). Le reti sequenziali sono caratterizzate in ogni istante da uno stato corrente che descrive l'informazione immagazzinata dal sistema circa i segnali giunti in ingresso nel passato. L'uscita di una rete sequenziale così come lo stato prossimo della rete è una funzione dello stato e degli ingressi attuali. Se la rete evolve da uno stato all'altro in risposta ad segnale di sincronismo, o clock, allora è detta **rete sequenziale sincrona**. Il **modello di Huffman** descrive una rete sequenziale sincrona che può essere utilizzata per costruire fisicamente automi a stati finiti di Moore e Mealy.



Modello di Huffman

5. Sintesi di un automa di Moore: Sommatore binario sequenziale infinito.

Si vuole costruire una rete sequenziale che dati in ingresso due sequenze di bit, dal bit meno significativo ai bit più significativi realizzi la somma binaria con l'usuale algoritmo aritmetico della somma bit a bit con riporto.



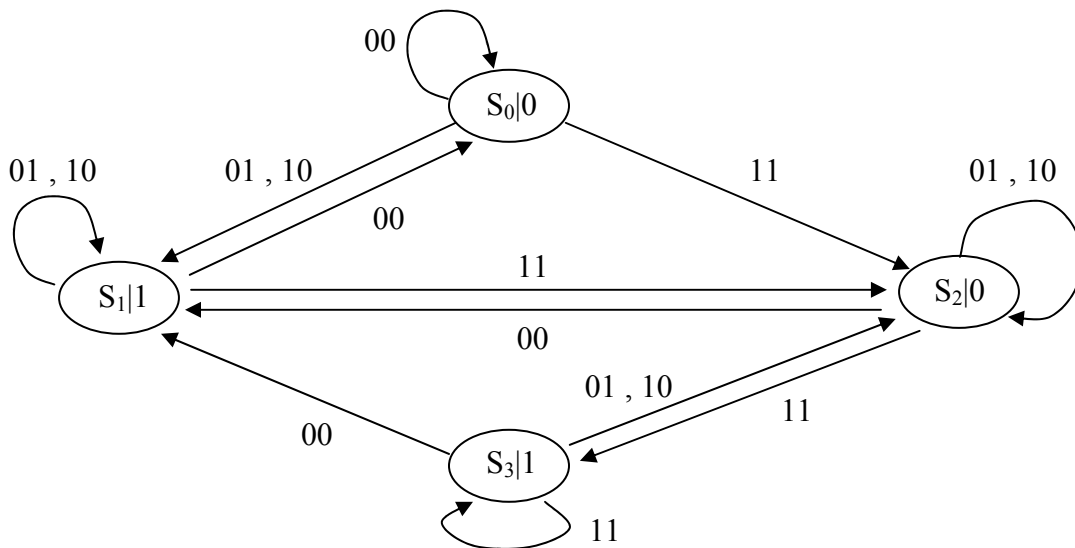
Il sistema in ogni passo di calcolo deve ricordare il risultato della somma dei due bit in ingresso al momento e l'eventuale riporto generato dai due bit che andrà propagato ai due bit seguenti. Il numero di stati del sistema è quindi pari a quattro:

Stato	Descrizione
S ₀	S= 0 , R=0
S ₁	S= 1 , R=0
S ₂	S= 0 , R=1
S ₃	S= 1 , R=1

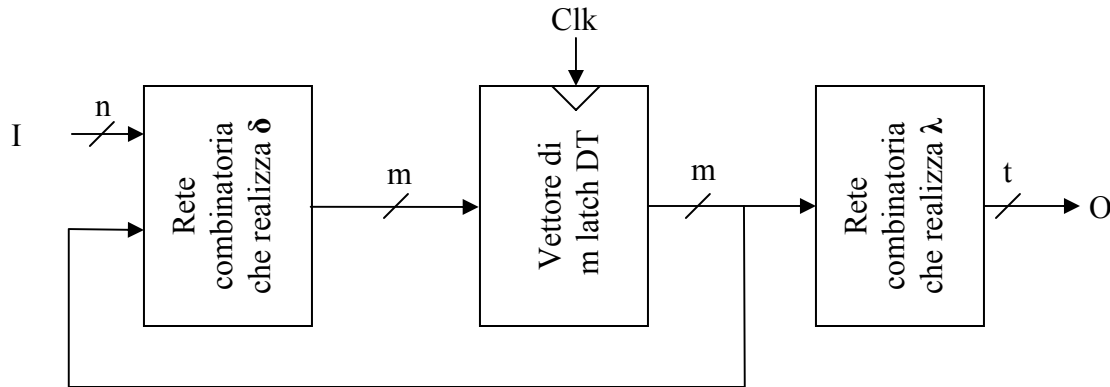
L'ingresso del sistema è costituito dai due bit da sommare. Le configurazioni in ingresso possibili (corrispondenti all'alfabeto di ingresso) sono quindi quattro: **00, 01, 10, 11**. L'uscita è costituita da un solo bit che può assumere sono due configurazioni, **0** oppure **1**, corrispondente alla somma dei due bit in ingresso più l'eventuale riporto precedente. Definiamo l'automa di Moore corrispondente attraverso la STT:

Stato	δ				λ
	I ₀ I ₁ =00	I ₀ I ₁ =01	I ₀ I ₁ =10	I ₀ I ₁ =11	O
S ₀	S ₀	S ₁	S ₁	S ₂	0
S ₁	S ₀	S ₁	S ₁	S ₂	1
S ₂	S ₁	S ₂	S ₂	S ₃	0
S ₃	S ₁	S ₂	S ₂	S ₃	1

L' STG del sistema è il seguente:



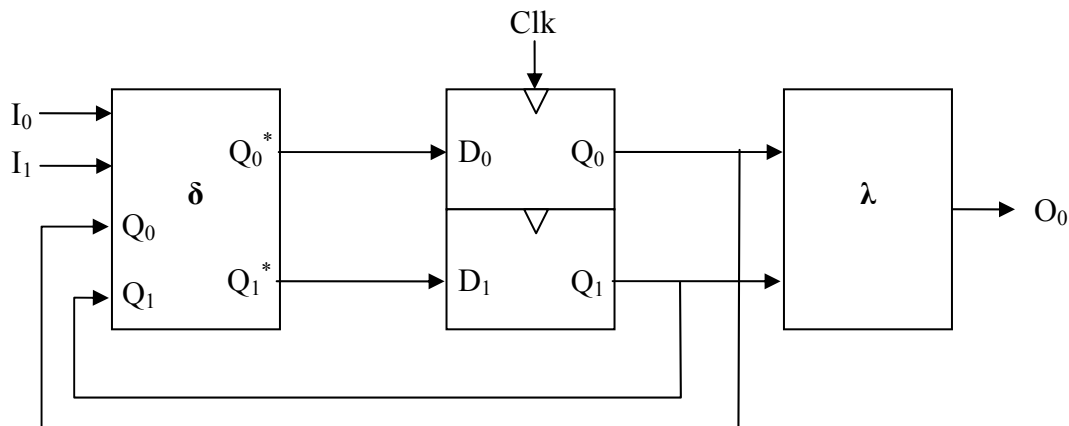
Un automa di Moore è fisicamente realizzabile tramite un circuito sequenziale così formato (è in pratica un modello di Huffman):



Il numero di latch occorrenti per memorizzare n stati è pari all'intero più piccolo non minore del logaritmo in base 2 di n. Poiché il sommatore richiede 4 stati il numero di latch da utilizzare sarà:

$$n.latch = \text{int_not_less}(\log_2 4) = 2.$$

Il circuito che realizza il sommatore dell'esempio è quindi:



Per sintetizzare le funzioni stato prossimo e di uscita occorre definire una corrispondenza tra gli stati del sistema e le configurazioni possibili dei latch. Tale mappatura può essere qualsiasi tenendo però presente che influenza la complessità finale delle due funzioni δ ed λ . “Buone” mappature generano funzioni semplici mentre “cattive” mappature generano funzioni più complesse. Non esiste una regola generale per definire questa mappatura.

Nel caso del sommatore conviene assegnare le configurazioni di bit in modo che Q_0 indichi la somma dei bit e Q_1 indichi l'eventuale riporto da propagare alla transizione successiva:

Stato	Descrizione	Q_0Q_1
S_0	$S=0, R=0$	00
S_1	$S=1, R=0$	10
S_2	$S=0, R=1$	01
S_3	$S=1, R=1$	11

Ora è possibile trascrivere la STT sostituendo alle etichette degli stati la corrispondente configurazione dei latch:

Q_0Q_1	$\delta = Q_0^* Q_1^*$				λ
	$I_0I_1=00$	$I_0I_1=01$	$I_0I_1=10$	$I_0I_1=11$	\mathbf{O}
00	00	10	10	01	0
10	00	10	10	01	1
01	10	01	01	11	0
11	10	01	01	11	1

La sintesi delle funzioni δ e λ può essere fatta in vari modi. Nei casi più semplici si può tentare di dedurre direttamente dalla tabella la formula che descrive la funzione. Negli altri casi occorre usare metodi più sistematici come le forme SOP e le mappe di Karnaugh.

Dalla STT si deduce immediatamente che l'uscita per ogni stato del sistema è banalmente uguale al valore di Q_0 :

$$\lambda(Q_0Q_1) = Q_0$$

Questa semplificazione deriva da come è stata scelta la mappatura tra stati e configurazioni della memoria.

Sempre dalla STT possiamo sintetizzare le formule stato prossimo per Q_0^* e Q_1^* . Riscrivo la tabella di verità per le due variabili:

$I_0I_1Q_0Q_1$	Q_0^*	Q_1^*
0000	0	0
0001	1	0
0010	0	0
0011	1	0
0100	1	0
0101	0	1
0110	1	0
0111	0	1
1000	1	0
1001	0	1
1010	1	0
1011	0	1
1100	0	1
1101	1	1
1110	0	1
1111	1	1

Esprimo la forma SOP per Q_0^* e Q_1^* (per comodità uso ABCD al posto di $I_0I_1Q_0Q_1$):

$$Q_0^* = \sim A \sim B \sim C D + \sim A \sim B C \sim D + \sim A B \sim C \sim D + \sim A B C \sim D + A \sim B \sim C \sim D + A \sim B C \sim D + A B \sim C D + A B C D$$

Riordino ed applico la regola (7° e 2b): $xy + x\sim y = x$

$$\begin{aligned} &= (\sim A B C \sim D + \sim A B \sim C \sim D) + (A \sim B C \sim D + A \sim B \sim C \sim D) \\ &\quad + (A B \sim C D + A B C D) + (\sim A \sim B C D + \sim A \sim B \sim C D) \\ &= \sim A B \sim D + A \sim B \sim D + A B D + \sim A \sim B D = (\sim A B + A \sim B) \sim D + (A B + \sim A \sim B) D \end{aligned}$$

Ricordando che $x\sim y + \sim x y = x \oplus y$ e che $xy + \sim x \sim y = \sim(x \oplus y)$ segue:

$$= (A \oplus B) \sim D + \sim(A \oplus B) D = (A \oplus B) \oplus D = (I_0 \oplus I_1) \oplus Q_1$$

Analogamente:

$$Q_1^* = \sim A B \sim C D + \sim A B C D + A \sim B \sim C D + A \sim B C D + A B \sim C \sim D + A B \sim C D + A B C \sim D + A B C D$$

Gli ultimi quattro termini sono ottenuti da AB moltiplicandolo per tutte le possibili configurazioni di CD. ne segue che possono essere semplificati (con 7a e 2b) in AB:

$$\begin{aligned} &= (\sim A B \sim C D + \sim A B C D) + (A \sim B \sim C D + A \sim B C D) + A B \\ &= (\sim A B D + A \sim B D) + A B = (\sim A B + A \sim B) D + A B = (A \oplus B) D + A B \\ &= (I_0 \oplus I_1) Q_1 + I_0 I_1 \end{aligned}$$

Le formule derivate per Q_0^* e Q_1^* non sono altro che le formule della somma e del riporto di un Full-Adder. Ciò deriva dalla scelta fatta per la mappatura degli stati nelle configurazioni dei latch. Una altra mappatura avrebbe generato altre formule (probabilmente più complesse).

La stessa sintesi poteva essere ottenuta attraverso le mappe di Karnaugh. Riscrivo la STT per le due variabili Q_0^* e Q_1^* riordinando righe e colonne in modo da avere nelle etichette un solo bit diverso tra due righe (e colonne) adiacenti in modo da ottenere una mappa a quattro variabili:

Q_0^* $Q_0 Q_1 = CD$	$I_0 I_1 = AB$			
	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	1	0	1	0
10	0	1	0	1

Dalla tabella ottengo gli implicant $\sim A \sim B D$, $\sim A B \sim D$, $A B D$, $A \sim B \sim D$ e quindi la formula:

$$Q_0^* = \sim A \sim B D + \sim A B \sim D + A B D + A \sim B \sim D = (A \oplus B) \oplus D = (I_0 \oplus I_1) \oplus Q_1$$

Analogamente:

Q_1^* $Q_0Q_1=CD$	$I_0I_1=AB$			
	00	01	11	10
00	0	0	1	0
01	0	1	1	1
11	0	1	1	1
10	0	0	1	0

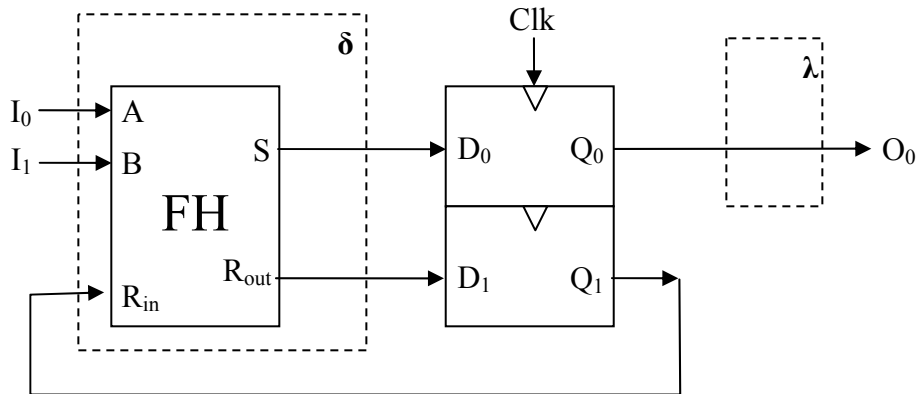
ottengo gli implicant BD , AB , AD e quindi la formula:

$$Q_0^* = AD + BD + AB = (A + B)D + AB = (I_0 + I_1)Q_1 + I_0I_1$$

che è equivalente (vedi lezioni sui sommatore) a:

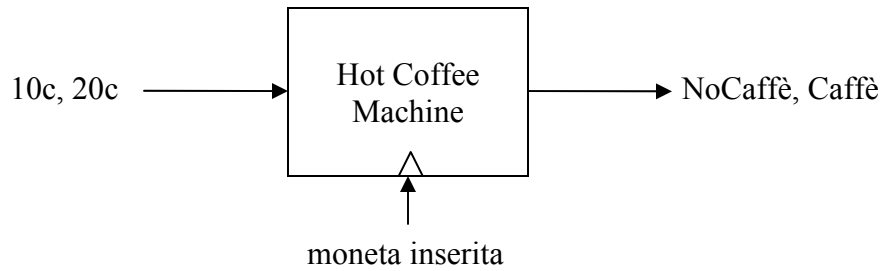
$$= (I_0 \oplus I_1)Q_1 + I_0I_1$$

Il circuito logico finale che realizza l'automa a stati finiti del sommatore binario sequenziale è:

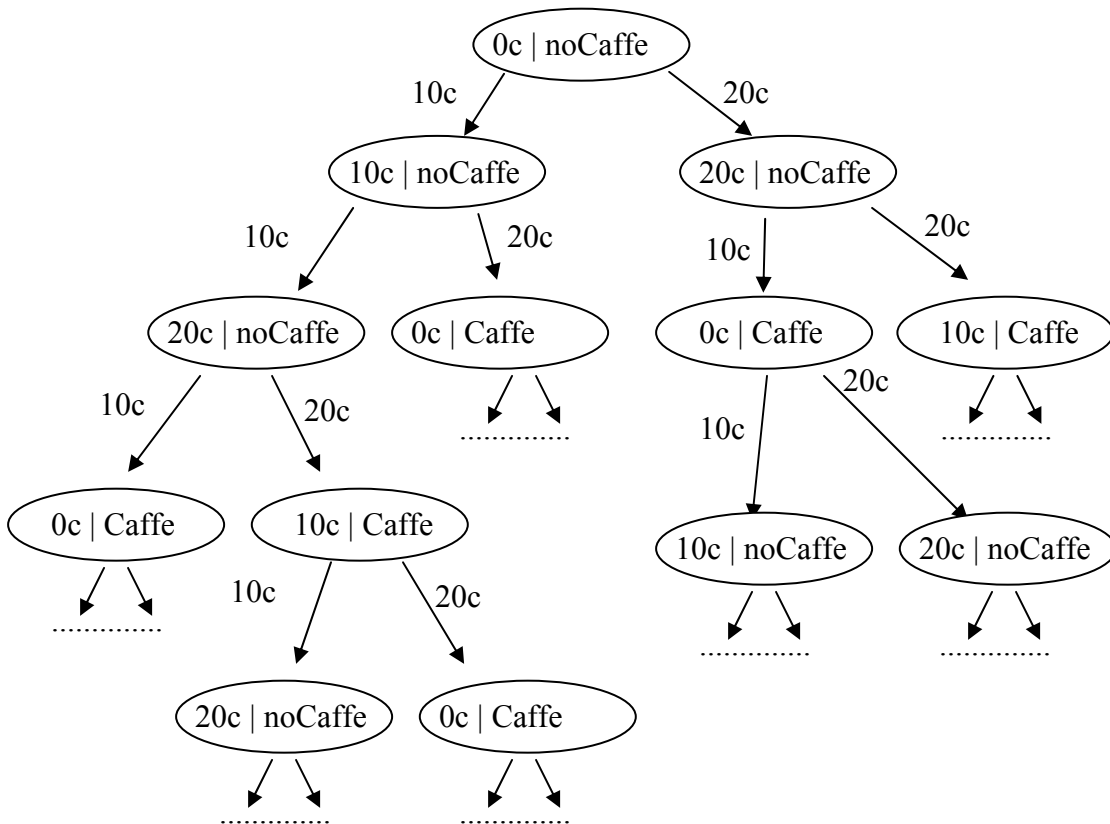


6. Realizzazione tramite MSF di una macchinetta del caffè

Supponiamo di voler modellare tramite un automa a stati finiti di Moore una macchinetta del caffè che rilascia un caffè quando l'utente introduce monete per 30 centesimi e supponiamo che accetti in ingresso monete da 10c e 20c. Supponiamo che le transizioni dell'automa si verifichino in corrispondenza dell'inserimento delle monetine e supponiamo per semplicità che la generazione del caffè sia istantanea (non bisogna attendere che la macchinetta abbia finito di fare il caffè per accettare nuove monete).



Lo stato della macchina sarà in funzione della quantità di denaro inserito non ancora utilizzato e della necessità di “generare in uscita” il caffè richiesto. Esaminiamo la sequenza di stati possibili supponendo che la macchina parta dallo stato iniziale S_0 corrispondente a nessuna moneta inserita.



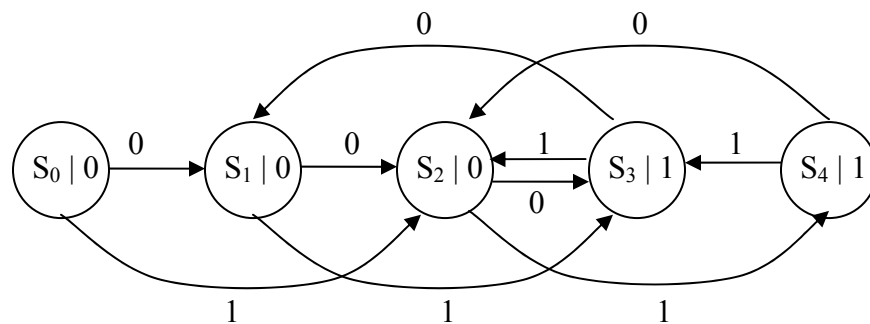
Dall’analisi della successione degli stati si può dedurre che la macchina si può trovare in al più 5 stati possibili:

Stato	Descrizione
S_0	Resto = 0c , noCaffè
S_1	Resto = 10c , noCaffè
S_2	Resto = 20c , noCaffè
S_3	Resto = 0c , Caffè
S_4	Resto = 10c , Caffè

Rappresentiamo le monete in ingresso attraverso un bit: assegnamo **0** alle monete da **10c** ed **1** alle monete da **20c**. Analogamente assegnamo **0** all'uscita **noCaffe** ed **1** all'uscita **Caffe**.

Nota: la mappatura tra possibili ingressi e codice binario corrispondente è spesso decisa dai circuiti presenti alla periferia dell'automa e quindi non è modificabile. Analogo discorso vale per la corrispondenza tra codice binario in uscita ed azione generata dal sistema. Diverso discorso vale per la mappatura tra stati interni e codice binario utilizzato all'interno dei latch di memoria che possiamo decidere arbitrariamente.

L'STG dell'automa quindi è il seguente:



L'STT dell'automa quindi è il seguente:

Stato	I ₀ =0	I ₀ =1	O
S ₀	S ₁	S ₂	0
S ₁	S ₂	S ₁	0
S ₂	S ₃	S ₄	0
S ₃	S ₁	S ₂	1
S ₄	S ₂	S ₃	1

Il numero di latch necessari per memorizzare i cinque stati possibili è:

$$n.latch = \text{int_no_less}(\log_2 5) = \text{int_no_less}(2.32) = 3$$

Per sintetizzare il circuito logico corrispondente all'automa dobbiamo dare una mappatura tra gli stati possibili e le configurazioni del latch. Poichè **3** latch sono in grado di codificare $2^3 = 8$ stati possibili, nel sistema in esame esisteranno $8-5 = 3$ configurazioni che non verranno mai raggiunte. Per queste configurazioni le funzioni stato-prossimo e uscita non sono definite e possono essere settate a valori arbitrari (allo scopo di semplificarne la formula logica).

Utilizziamo la seguente mappatura che semplicemente enumera gli stati:

Stato	Descrizione	$Q_0Q_1Q_2$
S_0	Resto = 0c , noCaffe	000
S_1	Resto = 10c , noCaffe	001
S_2	Resto = 20c , noCaffe	010
S_3	Resto = 0c , Caffè	011
S_4	Resto = 10c , Caffè	100

L'STT risultante è la seguente:

$Q_0^*Q_1^*Q_2^*$	I_0		O_0
	0	1	
000	001	010	0
001	010	011	0
010	011	100	0
011	001	010	1
100	010	011	1

Riscrivo la STT in forma tabellare separando le variabili $Q_0^*Q_1^*Q_2^*$:

$I_0Q_0Q_1Q_2$	Q_0^*	Q_1^*	Q_2^*
0000	0	0	1
0001	0	1	0
0010	0	1	1
0011	0	0	1
0100	0	1	0
0101	X	X	X
0110	X	X	X
0111	X	X	X
1000	0	1	0
1001	0	1	1
1010	1	0	0
1011	0	1	0
1100	0	1	1
1101	X	X	X
1110	X	X	X
1111	X	X	X

in cui le X rappresentano valori indefiniti delle funzioni (corrispondenti a configurazioni impossibili).

Uso le forme SOP per sintetizzare le funzioni logiche per le tre variabili e pongo tutti i valori indefiniti X a 0 (come nell'esempio precedente uso per comodità di scrittura ABCD al posto di I₀Q₀Q₁Q₂):

$$Q_0^* = A\sim BC\sim D = I_0\sim Q_0Q_1\sim Q_2$$

$$Q_1^* = \sim A\sim B\sim CD + (\sim A\sim BC\sim D + \sim AB\sim C\sim D) + (A\sim B\sim C\sim D + A\sim B\sim CD) + A\sim BCD + AB\sim C\sim D$$

$$= \sim A\sim B\sim CD + A\sim BCD + (\sim BC + B\sim C)\sim A\sim D + A\sim B\sim C + AB\sim C\sim D$$

$$= \sim(A \oplus C)\sim BD + (B \oplus C)\sim A\sim D + A\sim C(\sim B + B\sim D)$$

$$= \sim(A \oplus C)\sim BD + (B \oplus C)\sim A\sim D + A\sim C(\sim BD + \sim B\sim D + B\sim D)$$

$$= \sim(A \oplus C)\sim BD + (B \oplus C)\sim A\sim D + A\sim C(\sim BD + \sim D)$$

$$= \sim(A \oplus C)\sim BD + (B \oplus C)\sim A\sim D + A\sim C(\sim BD + \sim B\sim D + \sim D)$$

$$= \sim(A \oplus C)\sim BD + (B \oplus C)\sim A\sim D + A\sim C(\sim B + \sim D)$$

$$= \sim(I_0 \oplus Q_1)\sim Q_0Q_2 + (Q_0 \oplus Q_1)\sim I_0\sim Q_2 + I_0\sim Q_1(\sim Q_0 + \sim Q_2)$$

$$Q_2^* = \sim A\sim B\sim C\sim D + (\sim A\sim BC\sim D + \sim A\sim BCD) + (A\sim B\sim CD + AB\sim C\sim D)$$

$$= (\sim A\sim B\sim C\sim D + \sim A\sim BC) + A\sim C(\sim BD + B\sim D)$$

$$= \sim A\sim B(\sim C\sim D + C) + A\sim C(\sim BD + B\sim D)$$

$$= \sim A\sim B(\sim C\sim D + CD + C) + A\sim C(\sim BD + B\sim D)$$

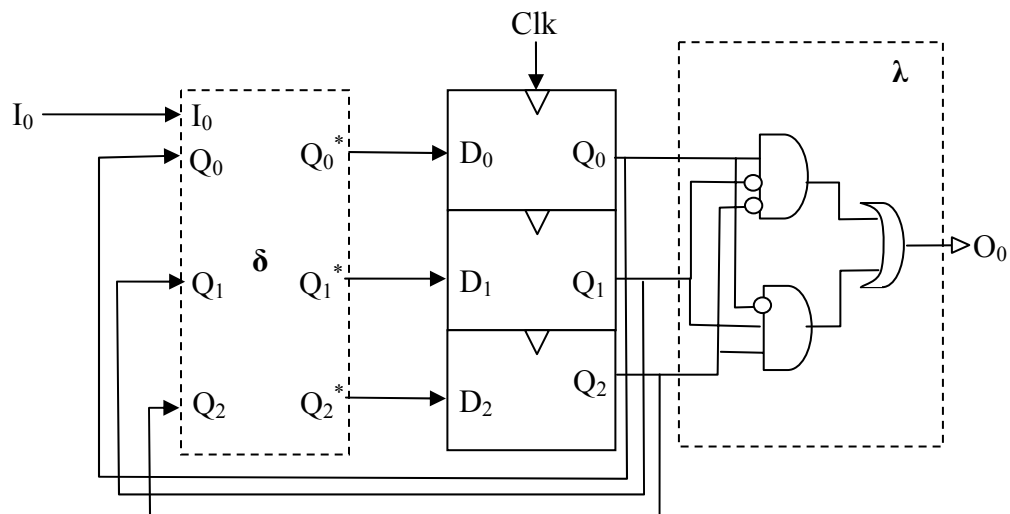
$$= \sim A\sim B(\sim D + C) + A\sim C(\sim BD + B\sim D)$$

$$= \sim I_0\sim Q_0(\sim Q_2 + Q_1) + I_0\sim Q_1(Q_0 \oplus Q_2)$$

La funzione di uscita, sintetizzata con la forma SOP, sarà:

$$O_0 = \sim Q_0Q_1Q_2 + Q_0\sim Q_1\sim Q_2$$

Il circuito fisico risultante sarà:



dove il quadrato tratteggiato δ rappresenta il circuito logico (omesso per brevità) che realizza la funzione stato prossimo in tre variabili $Q_0^* Q_1^* Q_2^*$.

Le formule per Q_1^* e Q_2^* risultano alquanto complesse. E' possibile tentare di ottenere un risultato più compatto scegliendo opportunamente i valori indefiniti. Riconsideriamo la STT precedente e proviamo un'altra configurazione di 0 ed 1 per i valori indefiniti che aiuti nelle semplificazioni:

$I_0 Q_0 Q_1 Q_2$	Q_1^*	Q_2^*
0000	0	1
0001	1	0
0010	1	1
0011	0	1
0100	1	0
0101	1	1
0110	1	0
0111	1	0
1000	1	0
1001	1	1
1010	0	0
1011	1	0
1100	1	1
1101	1	0
1110	1	1
1111	1	1

Calcoliamo Q_1^* ottimizzando i valori indeterminati per utilizzare le POS: pongo tutte le X della colonna a valore **1**. La funzione ora ha solo 3 zeri e quindi solo tre max-termini:

$$Q_1^* = (I_0 + Q_0 + Q_1 + Q_2) (I_0 + Q_0 + \sim Q_1 + \sim Q_2) (\sim I_0 + Q_0 + \sim Q_1 + Q_2)$$

Applico DeMorgan:

$$\begin{aligned} &= \sim (\sim(I_0 + Q_0 + Q_1 + Q_2) + \sim(I_0 + Q_0 + \sim Q_1 + \sim Q_2) + \sim(\sim I_0 + Q_0 + \sim Q_1 + Q_2)) \\ &= \sim (\sim I_0 \sim Q_0 \sim Q_1 \sim Q_2 + \sim I_0 \sim Q_0 Q_1 Q_2 + I_0 \sim Q_0 Q_1 \sim Q_2) \end{aligned}$$

Raccolgo e semplifico:

$$\begin{aligned} &= \sim (\sim I_0 \sim Q_0 (\sim Q_1 \sim Q_2 + Q_1 Q_2) + I_0 \sim Q_0 Q_1 \sim Q_2) \\ &= \sim (\sim I_0 \sim Q_0 \sim (Q_1 \oplus Q_2) + I_0 \sim Q_0 Q_1 \sim Q_2) \end{aligned}$$

In maniera simile è possibile ottenere una funzione più semplice per Q_2^* .

Alternativamente possiamo cercare di dedurre una funzione semplice per Q_2^* esaminando le caratteristiche topologiche della tabella. Possiamo notare, una volta risolte le indeterminazioni in maniera opportuna come riportato, che Q_0 si comporta per Q_2^* come fosse un segnale di inversione di una funzione $F()$ definita su $I_0 Q_1 Q_2$, cioè:

$$Q_2^* = Q_0 \oplus F(I_0 Q_1 Q_2)$$

dove $F()$ è:

$I_0Q_1Q_2$	F
000	1
001	0
010	1
011	1
100	0
101	1
110	0
111	0

Analogamente si può notare che in $F()$ I_0 si comporta come un segnale di inversione di una funzione $G()$ definita sulle sole Q_1Q_2 , cioè:

$$F(I_0Q_1Q_2) = I_0 \oplus G(Q_1Q_2)$$

dove $G()$ è la funzione:

$$G(Q_1Q_2) = Q_1 + \sim Q_2$$

Ne segue che Q_2^* può essere sintetizzata come:

$$Q_2^* = I_0 \oplus Q_0 \oplus (Q_1 + \sim Q_2) \sim (\sim I_0 \sim Q_0 \sim (Q_1 \oplus Q_2) + I_0 \sim Q_0 Q_1 \sim Q_2)$$

