Contatore avanti-indietro Modulo 4

Un contatore avanti-indietro modulo 4 è un dispositivo a due uscite, che genera su queste la sequenza dei numeri binari da 0 a 4 cioè:
00->01->10->11

Il sistema dispone inoltre di un ingresso i, che comanda la direzione del conteggio:
- *i=0* conteggio in avanti
- *i=1* conteggio all'indietro

Il sistema transita evidentemente attraverso 4 stati, corrispondenti ai quattro numeri generati. Partendo dallo stato S_0, se *i=0* il sistema si porta negli stati successivi, se *i=1* ripercorre a ritroso gli stati precedenti.

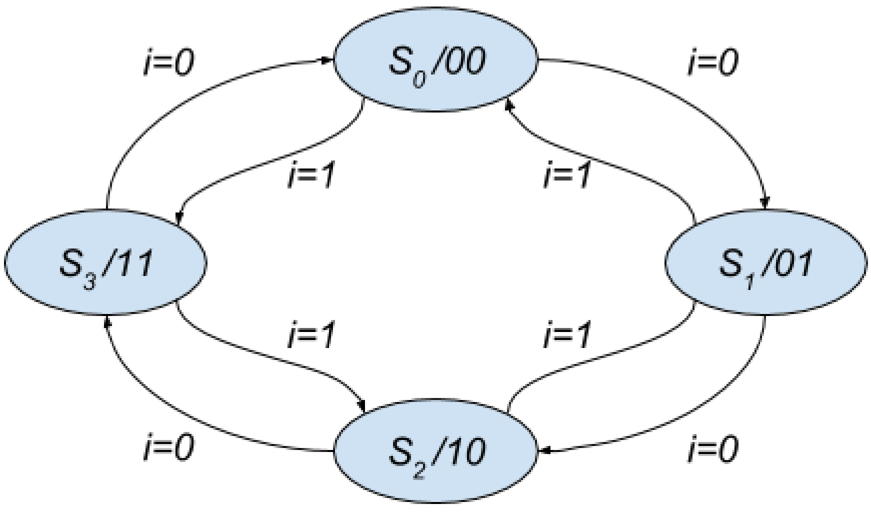
Indichiamo di seguito il flusso degli stati, in funzione del valore apposto all'ingresso:
- *i=0* => S_0->S_1->S_2->S_3->S_0 ...
- *i=1* => S_0->S_3->S_2->S_1->S_0 ...

Automa secondo Moore

Diagramma degli stati

In corrispondenza di questi stati le uscite U_1 e U_0 assumono i seguenti valori:
- S_0 -> U_1U_0 = 00
- S_1 -> U_1U_0 = 01
- S_2 -> U_1U_0 = 10
- S_3 -> U_1U_0 = 11

Tutto quello che è stato detto ora per descrivere la macchina viene condensato semplicemente nel seguente diagramma degli stati:



Rappresentazione tabellare

La rappresentazione tabellare è una rappresentazione alternativa ai diagramma degli stati, con lo stesso contenuto informativo. La descrizione del funzionamento viene ottenuta mediante due tabelle:

- **la tabella funzione transizione di stato**, identifica le transizioni da uno stato al successivo in funzione degli ingressi.
- **la tabella funzione trasformazione d'uscita**, descrive i valori assunti dalle uscite in funzione dei soli stati.

Tabella funzione transizione di stato

Si vede che nelle celle interne è riportato il valore dello stato futuro in corrispondenza dello stato presente (riga) e dell'ingresso (colonna).

	ingressi	
stati	***i=0***	***i=1***
$\boldsymbol{S_0}$	S_1	S_3
$\boldsymbol{S_1}$	S_2	S_0
$\boldsymbol{S_2}$	S_3	S_1
$\boldsymbol{S_3}$	S_0	S_2

Tabella funzione trasformazione d'uscita

Si vede che nelle celle interne è riportata l'uscita corrispondente allo stato associato.

stati	**uscita**
$\boldsymbol{S_0}$	*00*
$\boldsymbol{S_1}$	*01*
$\boldsymbol{S_2}$	*10*
$\boldsymbol{S_3}$	*11*

Implementazione Binaria

Dopo aver chiarito il significato e il ruolo degli stati, dobbiamo stabilire come rappresentarli all'interno della macchina. Fino ad ora abbiamo simbolicamente rappresentato i diversi stati come S_0, S_1, S_2 …

Nella nostra mente questi si configurano proprio come simboli grafici, ma nel caso di un sistema di elaborazione la rappresentazione avviene secondo forme più primitive, in funzione del tipo di macchina, ovvero del livello di astrazione con il quale lavora.

Gli elaboratori conservano in memoria i dati in formato binario, come sequenze di ‘0’ e ‘1’.
Le cifre binarie, conservate in memoria, danno luogo a diverse combinazioni, ciascuna delle quali viene eletta a rappresentare lo stato generico S_i. Le cifre binarie prendono in questo caso il nome di **variabili di stato** e vengono indicate come X_0, X_1, ... X_{N-1}

Assegnazione delle variabili di stato

Dobbiamo codificare i 4 stati del nostro automa. Il numero di variabili di stato binarie dovrà essere 2, in quanto con due variabili binarie si possono formare 2^2=4 combinazioni.

A partire dalla formulazione tabellare dell'automa occorre procedere al cosiddetto assegnamento delle variabili di stato, ovvero associare a ogni stato una delle configurazioni delle variabili di stato.
<u>La scelta è arbitraria e ininfluente sul funzionamento.</u>
Scelte differenti possono condurre a implementazioni differenti, cioè a diverse configurazioni circuitali, che tuttavia garantiscono il medesimo funzionamento.

Tabelle delle variabili di stato

stati	X_1	X_0
S_0	*0*	*0*
S_1	*0*	*1*
S_2	*1*	*0*
S_3	*1*	*1*

Riscriviamo la tabella di transizione di stato in funzione delle variabili di stato:

	ingressi	
stati X_1X_0	***i=0*** X_1X_0	***i=1*** X_1X_0
00	*01*	*11*
01	*10*	*00*
10	*11*	*01*
11	*00*	*10*

Il percorso che stiamo seguendo ci cala sempre più al livello di implementazione circuitale. Separiamo i valori delle due variabili di stato, ovvero “splittiamo” l’ultima tabella in due tabelle distinte, riportando in ciascuna i valori futuri di una sola variabile di stato.

	ingressi	
stati X_1X_0	***i=0*** X_1	***i=1*** X_1
00	*0*	*1*
01	*1*	*0*
10	*1*	*0*
11	*0*	*1*

	ingressi	
stati X_1X_0	***i=0*** X_0	***i=1*** X_0
00	*1*	*1*
01	*0*	*0*
10	*1*	*1*
11	*0*	*0*

Queste due descrivono funzioni logiche combinatorie. Una variabile stato futuro viene qui rappresentata in funzione delle variabili di stato presenti e dell'ingresso.

Scriviamo le funzioni combinatorie degli stai e cerchiamo di minimizzarle utilizzando l’algebra di Boole.

$$X_1(t+1) = \overline{X_1} \bullet \overline{X_0} \bullet i + \overline{X_1} \bullet X_0 \bullet \overline{i} + X_1 \bullet X_0 \bullet i + X_1 \bullet \overline{X_0} \bullet \overline{i}$$

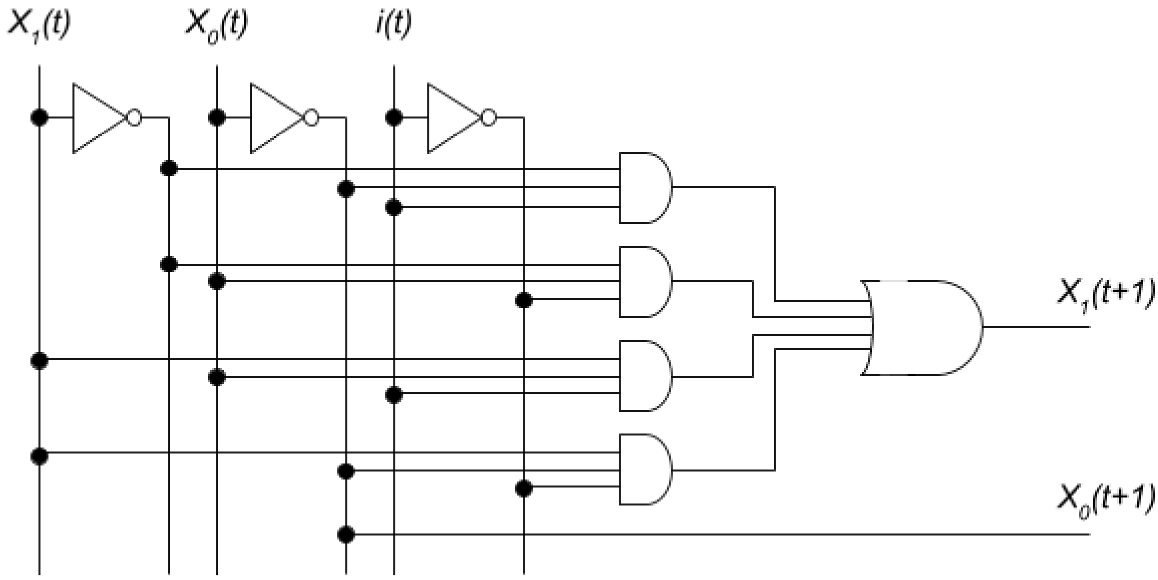
questa funzione non può essere minimizzata.

$$X_0(t+1) = \overline{X_1} \bullet \overline{X_0} \bullet \overline{i} + \overline{X_1} \bullet \overline{X_0} \bullet i + X_1 \bullet \overline{X_0} \bullet \overline{i} + X_1 \bullet \overline{X_0} \bullet i$$

questa funzione invece puo` essere minimizzata

$= (\overline{X_1} \bullet \overline{X_0}) \bullet (\overline{i} + i) + (X_1 \bullet \overline{X_0}) \bullet (\overline{i} + i)$ || in algebra di Boole $(\overline{i} + i) = 1$
$= \overline{X_1} \bullet \overline{X_0} + X_1 \bullet \overline{X_0}$
$= \overline{X_0} \bullet (\overline{X_1} + X_1)$
$= \overline{X_0}$

Queste funzioni, tradotte in forma circuitale, forniscono il circuito a porte logiche completo della sezione logica combinatoria.



Riscriviamo la tabella di trasformazione d'uscita in funzione delle variabili di stato:

stati	uscita
00	*00*
01	*01*
10	*10*
11	*11*

Ricordo che il fatto che gli stati e l'uscita hanno la stessa codifica è puramente casuale ed è dovuto alla scelta fatta della codifica degli stati.

Separiamo i valori delle due uscite, analogamente a come abbiamo fatto per le variabili di stato.

stati X_1X_0	**uscita** U_1
00	*0*
01	*0*
10	*1*
11	*1*

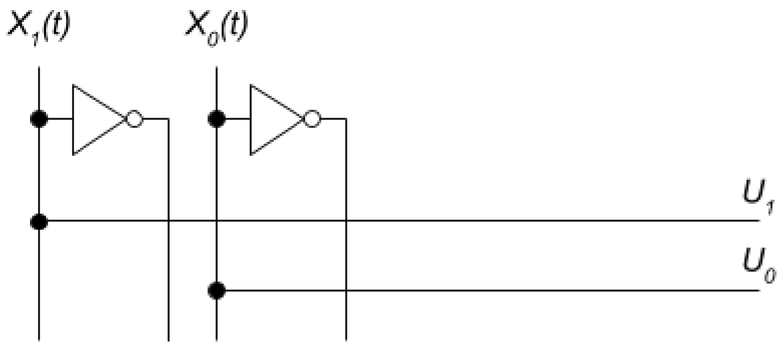
stati X_1X_0	**uscita** U_0
00	*0*
01	*1*
10	*0*
11	*1*

Scriviamo le funzioni combinatorie dell'uscita e cerchiamo di minimizzarle utilizzando l'algebra di Boole.

$$U_1 = X_1 \bullet \overline{X_0} + X_1 \bullet X_0$$
$$= X_1 \bullet (\overline{X_0} + X_0)$$
$$= X_1$$

$$U_0 = \overline{X_1} \bullet X_0 + X_1 \bullet X_0$$
$$= X_0 \bullet (\overline{X_1} + X_1)$$
$$= X_0$$

Queste funzioni, tradotte in forma circuitale, forniscono il circuito a porte logiche completo dell'uscita.



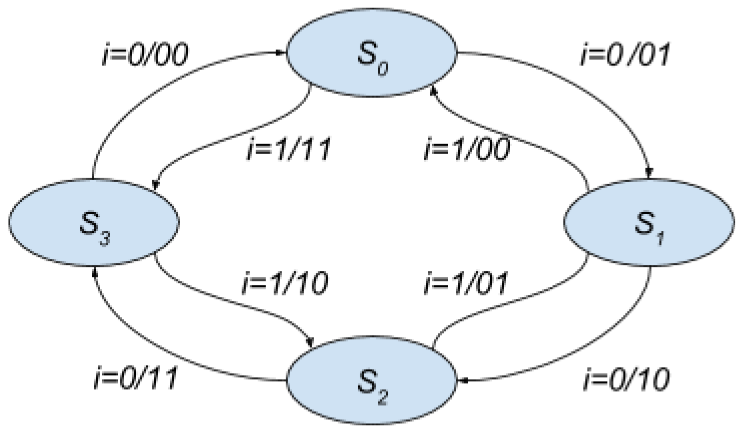
<u>Ricordiamo che le uscite non dipendono dall'ingresso visto che stiamo considerando un automa di Moore.</u>

Automa secondo Mealy

Diagramma degli stati

Ricordiamo che per la macchina di Mealy l'uscita è associata sia allo stato che all'ingresso, cioè è funzione oltre che dello stato anche dell'ingresso.

Tutto quello che è stato detto ora per descrivere la macchina viene condensato semplicemente nel seguente diagramma degli stati:



Rappresentazione tabellare

La rappresentazione tabellare è una rappresentazione alternativa ai diagramma degli stati, con lo stesso contenuto informativo. La descrizione del funzionamento viene ottenuta mediante due tabelle:

- **la tabella funzione transizione di stato**, identifica le transizioni da uno stato al successivo in funzione degli ingressi.
- **la tabella funzione trasformazione d'uscita**, descrive i valori assunti dalle uscite in funzione degli stati e degli ingressi

Tabella funzione transizione di stato

Si vede che nelle celle interne è riportato il valore dello stato futuro in corrispondenza dello stato presente (riga) e dell'ingresso (colonna).

	ingressi	
stati	***i=0***	***i=1***
S_0	S_1	S_3
S_1	S_2	S_0
S_2	S_3	S_1
S_3	S_0	S_2

Tabella funzione trasformazione d'uscita

Si vede che nelle celle interne è riportata l'uscita corrispondente allo stato e all'ingresso associato.

	ingressi		
stati	***i=0***	***i=1***	
S_0	*01*	*11*	***uscite***
S_1	*10*	*00*	
S_2	*11*	*01*	
S_3	*00*	*10*	

Implementazione Binaria

Dopo aver chiarito il significato e il ruolo degli stati, dobbiamo stabilire come rappresentarli all'interno della macchina. Fino ad ora abbiamo simbolicamente rappresentato i diversi stati come S_0, S_1, S_2 …
Nella nostra mente questi si configurano proprio come simboli grafici, ma nel caso di un sistema di elaborazione la rappresentazione avviene secondo forme più primitive, in funzione del tipo di macchina, ovvero del livello di astrazione con il quale lavora.
Gli elaboratori conservano in memoria i dati in formato binario, come sequenze di '0' e '1'.
Le cifre binarie, conservate in memoria, danno luogo a diverse combinazioni, ciascuna delle quali viene eletta a rappresentare lo stato generico S_i. Le cifre binarie prendono in questo caso il nome di **variabili di stato** e vengono indicate come X_0, X_1, ... X_{N-1}

Assegnazione delle variabili di stato

Dobbiamo codificare i 4 stati del nostro automa. Il numero di variabili di stato binarie dovrà essere 2, in quanto con due variabili binarie si possono formare 2^2=4 combinazioni.

A partire dalla formulazione tabellare dell'automa occorre procedere al cosiddetto assegnamento delle variabili di stato, ovvero associare a ogni stato una delle configurazioni delle variabili di stato.
La scelta è arbitraria e ininfluente sul funzionamento.
Scelte differenti possono condurre a implementazioni differenti, cioè a diverse configurazioni circuitali, che tuttavia garantiscono il medesimo funzionamento.

Tabelle delle variabili di stato

stati	X_1	X_0
S_0	0	0
S_1	0	1
S_2	1	0
S_3	1	1

Riscriviamo la tabella di transizione di stato in funzione delle variabili di stato:

	ingressi	
stati X_1X_0	$i=0$ X_1X_0	$i=1$ X_1X_0
00	01	11
01	10	00
10	11	01
11	00	10

Il percorso che stiamo seguendo ci cala sempre più al livello di implementazione circuitale.
Separiamo i valori delle due variabili di stato, ovvero "splittiamo" l'ultima tabella in due tabelle distinte, riportando in ciascuna i valori futuri di una sola variabile di stato.

	ingressi	
stati X_1X_0	$i=0$ X_1	$i=1$ X_1
00	0	1
01	1	0
10	1	0
11	0	1

	ingressi	
stati X_1X_0	$i=0$ X_0	$i=1$ X_0
00	1	1
01	0	0
10	1	1
11	0	0

Queste due descrivono funzioni logiche combinatorie. Una variabile stato futuro viene qui rappresentata in funzione delle variabili di stato presenti e dell'ingresso.

Scriviamo le funzioni combinatorie degli stai e cerchiamo di minimizzarle utilizzando l'algebra di Boole.

$$X_1(t+1) = \overline{X_1} \bullet \overline{X_0} \bullet i + \overline{X_1} \bullet X_0 \bullet \overline{i} + X_1 \bullet X_0 \bullet i + X_1 \bullet \overline{X_0} \bullet \overline{i}$$

questa funzione non può essere minimizzata.

$$X_0(t+1) = \overline{X_1} \bullet \overline{X_0} \bullet \overline{i} + \overline{X_1} \bullet \overline{X_0} \bullet i + X_1 \bullet \overline{X_0} \bullet \overline{i} + X_1 \bullet \overline{X_0} \bullet i$$

questa funzione invece puo` essere minimizzata

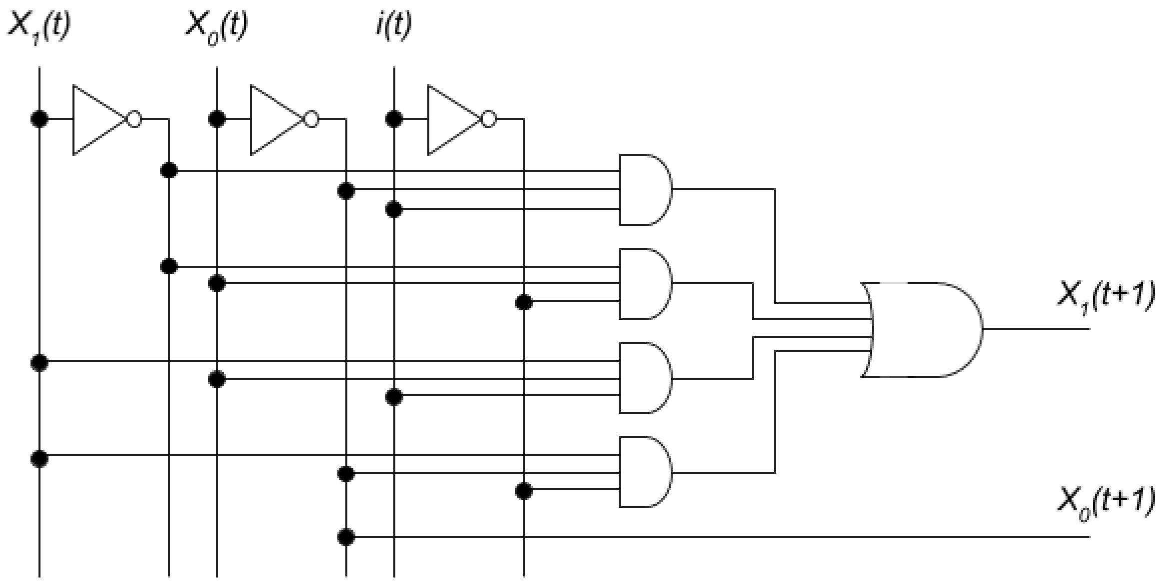
$= (\overline{X_1} \bullet \overline{X_0}) \bullet (\overline{i} + i) + (X_1 \bullet \overline{X_0}) \bullet (\overline{i} + i)$ || in algebra di Boole $(\overline{i} + i) = 1$

$= \overline{X_1} \bullet \overline{X_0} + X_1 \bullet \overline{X_0}$

$= \overline{X_0} \bullet (\overline{X_1} + X_1)$

$= \overline{X_0}$

Queste funzioni, tradotte in forma circuitale, forniscono il circuito a porte logiche completo della sezione logica combinatoria.



Riscriviamo la tabella di trasformazione d'uscita in funzione delle variabili di stato:

stati	uscita
00	*00*
01	*01*
10	*10*
11	*11*

Ricordo che il fatto che gli stati e l'uscita hanno la stessa codifica è puramente casuale ed è dovuto alla scelta fatta della codifica degli stati.

Separiamo i valori delle due uscite, analogamente a come abbiamo fatto per le variabili di stato.

	ingressi	
stati X_1X_0	***i=0*** U_1	***i=1*** U_1
00	*0*	*1*
01	*1*	*0*
10	*1*	*0*
11	*0*	*1*

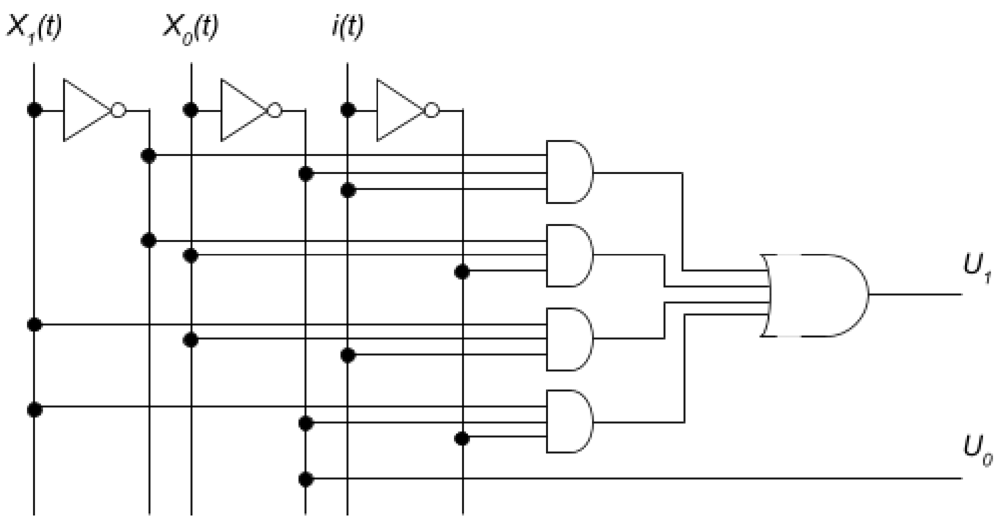
	ingressi	
stati X_1X_0	***i=0*** U_0	***i=1*** U_0
00	*1*	*1*
01	*0*	*0*
10	*1*	*1*
11	*0*	*0*

Scriviamo le funzioni combinatorie dell'uscita e cerchiamo di minimizzarle utilizzando l'algebra di Boole.

$$U_1 = \overline{X_1} \bullet \overline{X_0} \bullet i + \overline{X_1} \bullet X_0 \bullet \overline{i} + X_1 \bullet X_0 \bullet i + X_1 \bullet \overline{X_0} \bullet \overline{i}$$

$$U_0 = \overline{X_1} \bullet \overline{X_0} \bullet \overline{i} + \overline{X_1} \bullet \overline{X_0} \bullet i + X_1 \bullet \overline{X_0} \bullet \overline{i} + X_1 \bullet \overline{X_0} \bullet i$$
$$= (\overline{X_1} \bullet \overline{X_0}) \bullet (\overline{i} + i) + (X_1 \bullet \overline{X_0}) \bullet (\overline{i} + i)$$
$$= \overline{X_1} \bullet \overline{X_0} + X_1 \bullet \overline{X_0}$$
$$= \overline{X_0} \bullet (\overline{X_1} + X_1)$$
$$= \overline{X_0}$$

Queste funzioni, tradotte in forma circuitale, forniscono il circuito a porte logiche completo dell'uscita.



Ricordiamo che le uscite dipendono anche dall'ingresso visto che stiamo considerando un automa di Mealy.

Ricordiamo che i due schemi, per lo stato e per l'uscita, sono uguali solo perché abbiamo scelto la codifica dello stato identica alla codifica dell'uscita. Nel caso avessimo scelto una codifica dello stato differente avremmo avuto due schemi differenti.

Riconoscitore sequenza binaria 010

L'automa riceve serialmente, sul suo unico ingresso, una trama di bit, all'interno della quale deve identificare la sequenza *010*. I bit vengono ricevuti uno alla volta, pertanto il riconoscimento avviene per passi. Si definisce sequenza utile ogni sequenza di bit, di lunghezza inferiore a tre, tale da avvicinare la macchina alla condizione di riconoscimento.
L'automa lavora a ciclo continuo, spostandosi tra i vari stati in funzione dei nuovi bit di ingresso.

Occorre però individuare una condizione di inizio ciclo, uno stato di partenza. La scelta è arbitraria: di solito si sceglie lo stato non utile, quello a cui si perviene con una sequenza totalmente non utile. Scegliamo come stato di partenza S_0.

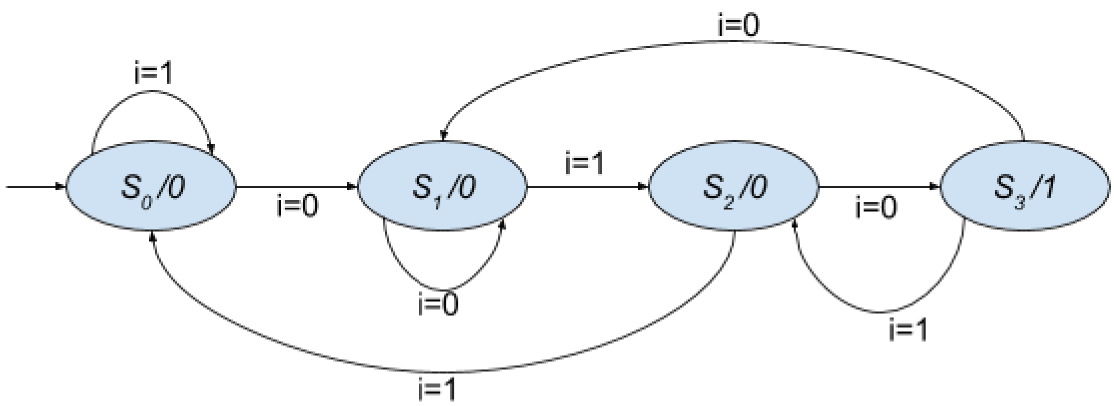
L'uscita è unica e rappresenta il bit di sequenza riconosciuta, cioè diventa *1* solo quando la sequenza *010* è stata riconosciuta.

Associamo gli stati alle sequenze utili:

- S_0 *-> non utile*
- S_1 *-> 0*
- S_2 *-> 01*
- S_3 *-> 010*

Automa secondo Moore

Diagramma degli stati



Rappresentazione tabellare

La rappresentazione tabellare è una rappresentazione alternativa ai diagramma degli stati, con lo stesso contenuto informativo. La descrizione del funzionamento viene ottenuta mediante due tabelle:

- **la tabella funzione transizione di stato**, identifica le transizioni da uno stato al successivo in funzione degli ingressi.
- **la tabella funzione trasformazione d'uscita**, descrive i valori assunti dalle uscite in funzione dei soli stati.

Tabella funzione transizione di stato

Si vede che nelle celle interne è riportato il valore dello stato futuro in corrispondenza dello stato presente (riga) e dell'ingresso (colonna).

ingressi

stati	*i=0*	*i=1*
S_0	S_1	S_0
S_1	S_1	S_2
S_2	S_3	S_0
S_3	S_1	S_2

Tabella funzione trasformazione d'uscita
Si vede che nelle celle interne è riportata l'uscita corrispondente allo stato associato.

stati	uscita
S_0	*0*
S_1	*0*
S_2	*0*
S_3	*1*

Implementazione Binaria

Dopo aver chiarito il significato e il ruolo degli stati, dobbiamo stabilire come rappresentarli all'interno della macchina. Fino ad ora abbiamo simbolicamente rappresentato i diversi stati come S_0, S_1, S_2 …
Nella nostra mente questi si configurano proprio come simboli grafici, ma nel caso di un sistema di elaborazione la rappresentazione avviene secondo forme più primitive, in funzione del tipo di macchina, ovvero del livello di astrazione con il quale lavora.
Gli elaboratori conservano in memoria i dati in formato binario, come sequenze di '0' e '1'.
Le cifre binarie, conservate in memoria, danno luogo a diverse combinazioni, ciascuna delle quali viene eletta a rappresentare lo stato generico S_i. Le cifre binarie prendono in questo caso il nome di **variabili di stato** e vengono indicate come X_0, X_1, ... X_{N-1}

Assegnazione delle variabili di stato

Dobbiamo codificare i 4 stati del nostro automa. Il numero di variabili di stato binarie dovrà essere 2, in quanto con due variabili binarie si possono formare $2^2=4$ combinazioni.

A partire dalla formulazione tabellare dell'automa occorre procedere al cosiddetto assegnamento delle variabili di stato, ovvero associare a ogni stato una delle configurazioni delle variabili di stato.
<u>La scelta è arbitraria e ininfluente sul funzionamento.</u>
Scelte differenti possono condurre a implementazioni differenti, cioè a diverse configurazioni circuitali, che tuttavia garantiscono il medesimo funzionamento.

Tabelle delle variabili di stato

stati	X_1	X_0
S_0	*0*	*0*
S_1	*0*	*1*
S_2	*1*	*0*
S_3	*1*	*1*

Riscriviamo la tabella di transizione di stato in funzione delle variabili di stato:

	ingressi	
stati X_1X_0	$i=0$ X_1X_0	$i=1$ X_1X_0
00	*01*	*00*
01	*01*	*10*
10	*11*	*00*
11	*01*	*10*

Il percorso che stiamo seguendo ci cala sempre più al livello di implementazione circuitale. Separiamo i valori delle due variabili di stato, ovvero “splittiamo” l’ultima tabella in due tabelle distinte, riportando in ciascuna i valori futuri di una sola variabile di stato.

	ingressi	
stati X_1X_0	$i=0$ X_1	$i=1$ X_1
00	*0*	*0*
01	*0*	*1*

	ingressi	
stati X_1X_0	$i=0$ X_0	$i=1$ X_0
00	*1*	*0*
01	*1*	*0*

10	*1*	*0*
11	*0*	*1*

10	*1*	*0*
11	*1*	*0*

Queste due descrivono funzioni logiche combinatorie. Una variabile stato futuro viene qui rappresentata in funzione delle variabili di stato presenti e dell'ingresso.

Scriviamo le funzioni combinatorie degli stai e cerchiamo di minimizzarle utilizzando l'algebra di Boole.

$X_1(t+1) = \overline{X_1} \bullet X_0 \bullet i + X_1 \bullet \overline{X_0} \bullet \overline{i} + X_1 \bullet X_0 \bullet i$

$= X_0 \bullet i \bullet (\overline{X_1} + X_1) + X_1 \bullet \overline{X_0} \bullet \overline{i}$

$= X_0 \bullet i + X_1 \bullet \overline{X_0} \bullet \overline{i}$

$X_0(t+1) = \overline{X_1} \bullet \overline{X_0} \bullet \overline{i} + \overline{X_1} \bullet X_0 \bullet \overline{i} + X_1 \bullet \overline{X_0} \bullet \overline{i} + X_1 \bullet X_0 \bullet \overline{i}$

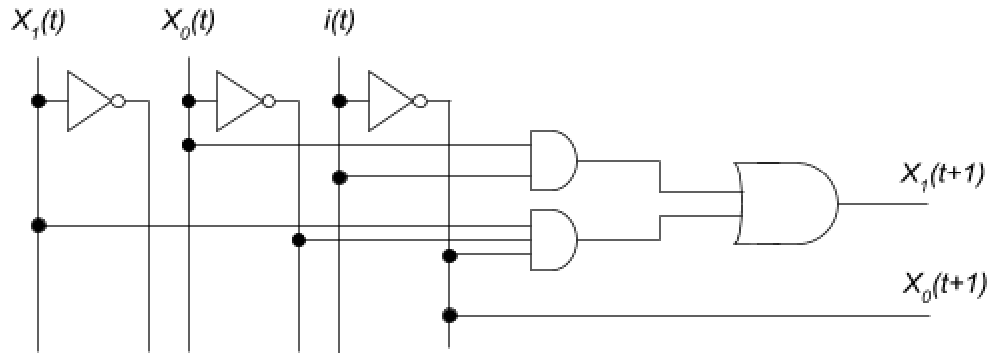
$= \overline{i} \bullet (\overline{X_1} \bullet \overline{X_0} + \overline{X_1} \bullet X_0 + X_1 \bullet \overline{X_0} + X_1 \bullet X_0)$

$= \overline{i} \bullet (\overline{X_1} \bullet (\overline{X_0} + X_0) + X_1 \bullet (\overline{X_0} + X_0))$

$= \overline{i} \bullet (\overline{X_1} + X_1)$

$= \overline{i}$

Queste funzioni, tradotte in forma circuitale, forniscono il circuito a porte logiche completo della sezione logica combinatoria.



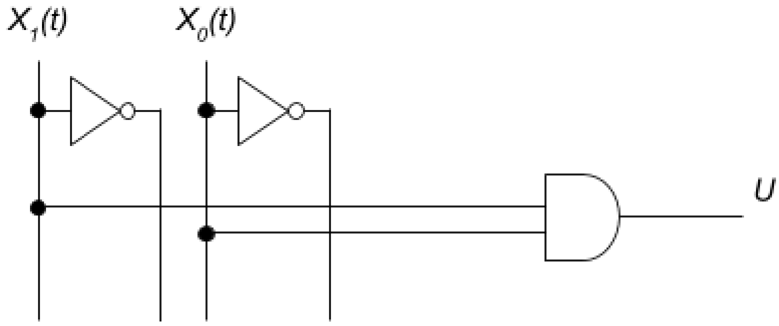
Riscriviamo la tabella di trasformazione d'uscita in funzione delle variabili di stato:

stati	**uscita**
00	*0*
01	*0*
10	*0*
11	*1*

Scriviamo le funzioni combinatorie dell'uscita e cerchiamo di minimizzarle utilizzando l'algebra di Boole.

$U_1 = X_1 \bullet X_0$

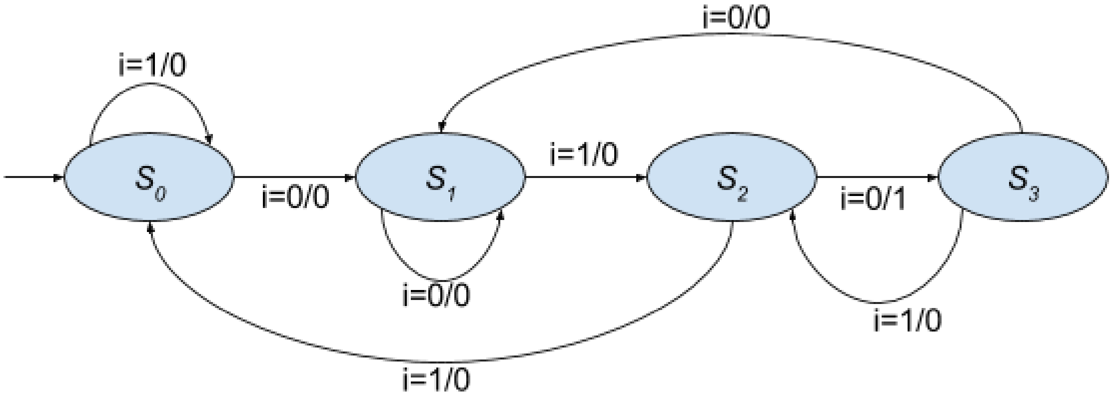
Queste funzioni, tradotte in forma circuitale, forniscono il circuito a porte logiche completo dell'uscita.



<u>Ricordiamo che le uscite non dipendono dall'ingresso visto che stiamo considerando un automa di Moore.</u>

<u>Automa secondo Mealy</u>

Diagramma degli stati



Rappresentazione tabellare

La rappresentazione tabellare è una rappresentazione alternativa ai diagramma degli stati, con lo stesso contenuto informativo. La descrizione del funzionamento viene ottenuta mediante due tabelle:

- **la tabella funzione transizione di stato**, identifica le transizioni da uno stato al successivo in funzione degli ingressi.
- **la tabella funzione trasformazione d'uscita**, descrive i valori assunti dalle uscite in funzione degli stati e degli ingressi.

Tabella funzione transizione di stato

Si vede che nelle celle interne è riportato il valore dello stato futuro in corrispondenza dello stato presente (riga) e dell'ingresso (colonna).

	ingressi	
stati	***i=0***	***i=1***
$\boldsymbol{S_0}$	S_1	S_0
$\boldsymbol{S_1}$	S_1	S_2
$\boldsymbol{S_2}$	S_3	S_0
$\boldsymbol{S_3}$	S_1	S_2

Tabella funzione trasformazione d'uscita

Si vede che nelle celle interne è riportata l'uscita corrispondente allo stato e all’ingresso associato.

	ingressi		
stati	***i=0***	***i=1***	
S_0	*0*	*0*	***uscite***
S_1	*0*	*0*	
S_2	*1*	*0*	
S_3	*0*	*0*	

Implementazione Binaria

Dopo aver chiarito il significato e il ruolo degli stati, dobbiamo stabilire come rappresentarli all'interno della macchina. Fino ad ora abbiamo simbolicamente rappresentato i diversi stati come S_0, S_1, S_2 …

Nella nostra mente questi si configurano proprio come simboli grafici, ma nel caso di un sistema di elaborazione la rappresentazione avviene secondo forme più primitive, in funzione del tipo di macchina, ovvero del livello di astrazione con il quale lavora.

Gli elaboratori conservano in memoria i dati in formato binario, come sequenze di '0' e '1'.

Le cifre binarie, conservate in memoria, danno luogo a diverse combinazioni, ciascuna delle quali viene eletta a rappresentare lo stato generico S_i. Le cifre binarie prendono in questo caso il nome di **variabili di stato** e vengono indicate come X_0, X_1, ... X_{N-1}

Assegnazione delle variabili di stato

Dobbiamo codificare i 4 stati del nostro automa. Il numero di variabili di stato binarie dovrà essere 2, in quanto con due variabili binarie si possono formare 2^2=4 combinazioni.

A partire dalla formulazione tabellare dell'automa occorre procedere al cosiddetto assegnamento delle variabili di stato, ovvero associare a ogni stato una delle configurazioni delle variabili di stato.

<u>La scelta è arbitraria e ininfluente sul funzionamento.</u>

Scelte differenti possono condurre a implementazioni differenti, cioè a diverse configurazioni circuitali, che tuttavia garantiscono il medesimo funzionamento.

Tabelle delle variabili di stato

stati	***X_1***	***X_0***

S_0	0	0
S_1	0	1
S_2	1	0
S_3	1	1

Riscriviamo la tabella di transizione di stato in funzione delle variabili di stato:

	ingressi	
stati X_1X_0	***i=0*** X_1X_0	***i=1*** X_1X_0
00	*01*	*00*
01	*01*	*10*
10	*11*	*00*
11	*01*	*10*

Il percorso che stiamo seguendo ci cala sempre più al livello di implementazione circuitale. Separiamo i valori delle due variabili di stato, ovvero "splittiamo" l'ultima tabella in due tabelle distinte, riportando in ciascuna i valori futuri di una sola variabile di stato.

	ingressi	
stati X_1X_0	***i=0*** X_1	***i=1*** X_1
00	*0*	*0*
01	*0*	*1*
10	*1*	*0*
11	*0*	*1*

	ingressi	
stati X_1X_0	***i=0*** X_0	***i=1*** X_0
00	*1*	*0*
01	*1*	*0*
10	*1*	*0*
11	*1*	*0*

Queste due descrivono funzioni logiche combinatorie. Una variabile stato futuro viene qui rappresentata in funzione delle variabili di stato presenti e dell'ingresso.

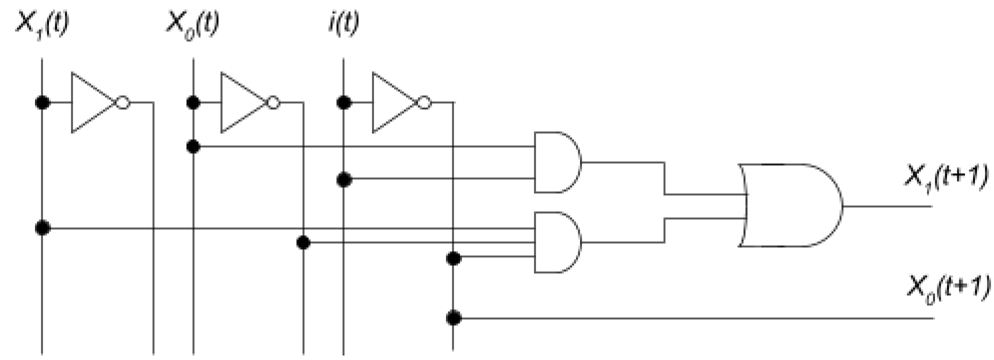
Scriviamo le funzioni combinatorie degli stai e cerchiamo di minimizzarle utilizzando l'algebra di Boole.

$$X_1(t+1) = \overline{X_1} \bullet X_0 \bullet i + X_1 \bullet \overline{X_0} \bullet \overline{i} + X_1 \bullet X_0 \bullet i$$

$= X_0 \bullet i \bullet (\overline{X_1} + X_1) \ + \ X_1 \bullet \overline{X_0} \bullet \overline{i}$
$= X_0 \bullet i \ + \ X_1 \bullet \overline{X_0} \bullet \overline{i}$

$X_0(t+1) \ = \ \overline{X_1} \bullet \overline{X_0} \bullet \overline{i} \quad + \ \overline{X_1} \bullet X_0 \bullet \overline{i} \ + \ X_1 \bullet \overline{X_0} \bullet \overline{i} \ + \ X_1 \bullet X_0 \bullet \overline{i}$
$= \overline{i} \bullet (\overline{X_1} \bullet \overline{X_0} + \overline{X_1} \bullet X_0 + X_1 \bullet \overline{X_0} \ + \ X_1 \bullet X_0)$
$= \overline{i} \bullet (\overline{X_1} \bullet (\overline{X_0} + X_0) + X_1 \bullet (\overline{X_0} \ + \ X_0))$
$= \overline{i} \bullet (\overline{X_1} + X_1)$
$= \overline{i}$

Queste funzioni, tradotte in forma circuitale, forniscono il circuito a porte logiche completo della sezione logica combinatoria.



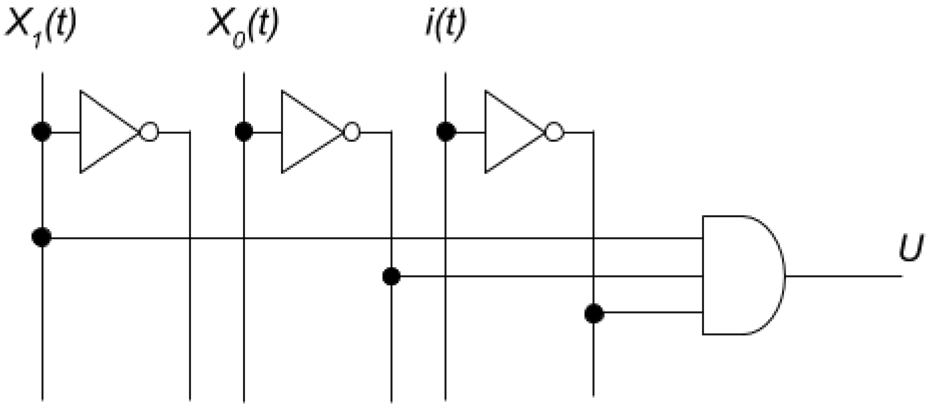
Riscriviamo la tabella di trasformazione d'uscita in funzione delle variabili di stato:

	ingressi	
stati X_1X_0	***i=0*** ***U***	***i=1*** ***U***
00	*0*	*0*
01	*0*	*0*
10	*1*	*0*
11	*0*	*0*

Scriviamo le funzioni combinatorie dell'uscita e cerchiamo di minimizzarle utilizzando l'algebra di Boole.

$U_1 \ = \ X_1 \bullet \overline{X_0} \bullet \overline{i}$

Queste funzioni, tradotte in forma circuitale, forniscono il circuito a porte logiche completo dell'uscita.



<u>Ricordiamo che le uscite dipendono anche dall’ingresso visto che stiamo considerando un automa di Mealy.</u>