ESEMPIO

giovedì 3 novembre 2022 17:25

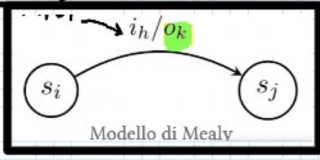
- PROGETTARE UNA RETE SEQUENZIALE CHE CI DICE SE, IN UNA SEQUENZA DI CARATTERI IN INPUT, IO HO LETTO LA STRINGA "MAMMA".
  VOGLIO COSTRUIRE UN CIRCUITO CHE MI DICA SE HO LETTO MAMMA O NON HO LETTO MAMMA.

$I = \{m, a\}$ $O = \{si, no\}$ S_0 STATO INIZIALE $= 0$

ALFABETO INGRESSO ALFABETO di USCITA

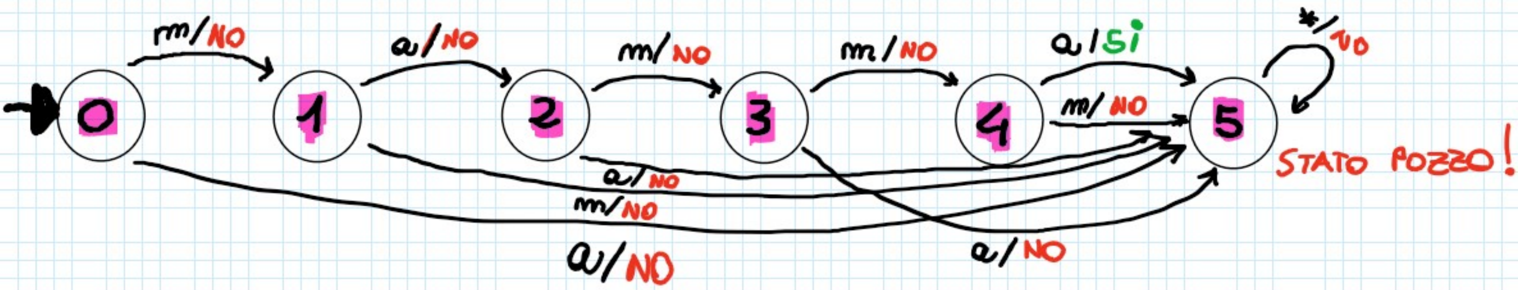
USIAMO IL MODELLO di MEALY:

> MENTRE STO FACENDO UNA TRANSIZIONE DA UNO STATO AD UN ALTRO, AD UN DETERMINATO INPUT, LA MACCHINA DI MEALY GENERERÀ UN OUTPUT.



MEALY: L'OUTPUT DIPENDE DALLO STATO E $\underbrace{\text{DALLA TRANSIZIONE INNESCATA}}_{\text{INGRESSO CORRENTE}}$.

SE LEGGO "m" TRANSITO IN UNO STATO SUCCESSIVO, e se ho letto m non ho letto "mamma".



IO VOGLIO COSTRUIRE UN CIRCUITO CHE IMPLEMENTI QUESTO DISEGNO!

TROVO LA TABELLA DELLE TRANSIZIONI

STATI	m 0	a 1
0	1/NO	5/NO
1	5/NO	2/NO
2	3/NO	5/NO
3	4/NO	5/NO
4	5/NO	5/SI
5	5/NO	5/NO

I

SE STO NELLO STATO 0 E LEGGO m, VADO NELLO STATO 1 CON OUTPUT NO.

$$\underbrace{1}_{=\,\delta}/\underbrace{NO}_{=\,W}$$

ORA DECIDO UNA codifica PER I MIEI caratteri IN INGRESSO e caratteri IN USCITA;

I	X
m	0
a	1

O	Z
Si	1
No	0

A volte gli INPUT e GLI OUTPUT, sono già delle codifiche. Lì non la posso scegliere.

UNIAMO TUTTO INSIEME! HO 6 STATI, HO BISOGNO DI 3 BIT PER RAPPRESENTARE 6 STATI. LA M SARÀ COMPOSTA DA 3 BIT.

DEFINISCO I MIEI STATI CORRENTI: y_3, y_2, y_1 CI ACCOPPIO UN INPUT Z. — QUESTE MI PERMETTONO DI DEFINIRE IL NUOVO STATO E UN OUTPUT.

INPUT				OUTPUT			
X	y_3	y_2	y_1	y_3'	y_2'	y_1'	Z
0	0	0	0	0	0	1	0
0	0	0	1	1	0	1	0
0	0	1	0	0	1	1	0
0	0	1	1	1	0	0	0
0	1	0	0	1	0	1	0
0	1	0	1	1	0	1	0
0	1	1	0	-	-	-	-
0	1	1	1	-	-	-	-
1	0	0	0	1	0	1	0
1	0	0	1	0	1	0	0
1	0	1	0	1	0	1	0
1	0	1	1	1	0	1	0
1	1	0	0	1	0	1	1
1	1	0	1	1	0	1	0
1	1	1	0	-	-	-	-
1	1	1	1	-	-	-	-

— = DON'T CARE CONDITIONS

Se sono in zero e ricevo 0 (X=0) il nuovo stato sarà 1. Da leggere nella prima riga.

QUESTA È UNA TABELLA DI VERITÀ CON y_3', y_2', y_1'. È LA TABELLA DI VERITÀ SU QUESTE FUNZIONI!

SIGNIFICA che, analizzandola, POSSO COSTRUIRE UN CIRCUITO.

INCOMINCIAMO.

TROVIAMO UNA FORMA MINIMA di queste FUNZIONI ed implementiamo il CIRCUITO;

IMPLEMENTAZIONE y_3'

$x\,y_3 \backslash y_2\,y_1$	00	01	11	10
00	0	1	1	0
01	1	1	-	-
11	1	1	-	-
10	1	0	1	1

$\longrightarrow$

$x\,y_3 \backslash y_2\,y_1$	00	01	11	10
00	0	1	1	0
01	1	1	-	-
11	1	1	-	-
10	1	0	1	1

- $\bar{x}\,\bar{y}_3\,\bar{y}_2\,y_1 + \bar{x}\,\bar{y}_3\,y_2\,y_1 + \bar{x}\,y_3\,\bar{y}_2\,y_1 + \bar{x}\,y_3\,y_2\,y_1$

  $\bar{x}\,\bar{y}_3\,\cancel{\bar{y}_2}\,y_1 + \bar{x}\,\bar{y}_3\,\cancel{y_2}\,y_1 + \bar{x}\,y_3\,\cancel{\bar{y}_2}\,y_1 + \bar{x}\,y_3\,\cancel{y_2}\,y_1$

  $\bar{x}\,\cancel{\bar{y}_3}\,y_1 + \bar{x}\,\cancel{y_3}\,y_1 = \boxed{\bar{x}\,y_1}$

- $\bar{x}\,y_3\,\bar{y}_2\bar{y}_1 + \bar{x}\,y_3\,\bar{y}_2\,y_1 + \bar{x}\,y_3\,y_2\,y_1 + \bar{x}\,y_3\,y_2\,\bar{y}_1 + x\,y_3\,\bar{y}_2\,\bar{y}_1 + x\,y_3\,\bar{y}_2\,y_1 + x\,y_3\,y_2\,y_1$
  $+ x\,y_3\,y_2\,\bar{y}_1$

$\bar{x} y_3 \bar{y_2} \cancel{\bar{y_1}} + \bar{x} y_3 \bar{y_2} \cancel{y_1} + \bar{x} y_3 y_2 \cancel{y_1} + \bar{x} y_3 y_2 \cancel{\bar{y_1}} + x y_3 \bar{y_2} \cancel{\bar{y_1}} + x y_3 \bar{y_2} \cancel{y_1} + x y_3 y_2 \cancel{y_1}$
$+ x y_3 y_2 \cancel{\bar{y_1}}$

$\bar{x} y_3 \cancel{\bar{y_2}} + \bar{x} y_3 \cancel{y_2} + x y_3 \cancel{\bar{y_2}} + x y_3 \cancel{y_2}$

$\cancel{\bar{x}} y_3 + \cancel{x} y_3 = [y_3]$

- $x y_3 \cancel{y_2} \bar{y_1} + x y_3 \cancel{\bar{y_2}} \bar{y_1} + x \bar{y_3} \cancel{y_2} \bar{y_1} + x \bar{y_3} \cancel{\bar{y_2}} \bar{y_1}$

$x \cancel{y_3} \bar{y_1} + x \cancel{\bar{y_3}} \bar{y_1} = [x \bar{y_1}]$

$x y_3 y_2 y_1 + x y_3 y_2 \bar{y_1} + x \bar{y_3} y_2 y_1 + x \bar{y_3} y_2 \bar{y_1}$

$x y_3 y_2 \cancel{y_1} + x y_3 y_2 \cancel{\bar{y_1}} + x \bar{y_3} y_2 \cancel{y_1} + x \bar{y_3} y_2 \cancel{\bar{y_1}}$

$x \cancel{y_3} y_2 + x \cancel{\bar{y_3}} y_2 = [x y_2]$

SOMMIAMO: $\bar{x} y_1 + y_3 + x \bar{y_1} + x y_2$

IMPLEMENTAZIONE y_2'

$x y_3 \backslash y_2 y_1$	00	01	11	10
00	0	0	0	1
01	0	0	–	–
11	0	0	–	–
10	0	1	0	0

$\longrightarrow$

$x y_3 \backslash y_2 y_1$	00	01	11	10
00	0	0	0	1
01	0	0	–	–
11	0	0	–	–
10	0	1	0	0

INPUT				OUTPUT			
X	y_3	y_2	y_1	y_3'	y_2'	y_1'	Z
0	0	0	0	0	0	1	0
0	0	0	1	1	0	1	0
0	0	1	0	0	1	1	0
0	0	1	1	1	0	0	0
0	1	0	0	1	0	1	0
0	1	0	1	1	0	1	0
0	1	1	0	–	–	–	–
0	1	1	1	–	–	–	–
1	0	0	0	1	0	1	0
1	0	0	1	0	1	0	0
1	0	1	0	1	0	1	0
1	0	1	1	1	0	1	0
1	1	0	0	1	0	1	1
1	1	0	1	1	0	1	0
1	1	1	0	–	–	–	–
1	1	1	1	–	–	–	–

PRENDIAMO I MAX-TERMINI!

- $(x + \cancel{y_3} + y_2 + y_1) \cdot (x + \cancel{\bar{y_3}} + y_2 + y_1) \cdot (\bar{x} + \cancel{\bar{y_3}} + y_2 + y_1) \cdot (\bar{x} + \cancel{y_3} + y_2 + y_1)$

$(\cancel{x} + y_2 + y_1) \cdot (\cancel{\bar{x}} + y_2 + y_1) = [y_2 + y_1]$

- $(x + y_3 + \cancel{y_2} + \bar{y_1}) \cdot (x + y_3 + \cancel{\bar{y_2}} + \bar{y_1}) \cdot (x + \bar{y_3} + \cancel{y_2} + \bar{y_1}) \cdot (x + \bar{y_3} + \cancel{\bar{y_2}} + \bar{y_1})$

$(x + \cancel{y_3} + \bar{y_1}) \cdot (x + \cancel{\bar{y_3}} + \bar{y_1}) = [x + \bar{y_1}]$

- $(x + \bar{y_3} + y_2 + y_1) \cdot (x + \bar{y_3} + y_2 + \bar{y_1}) \cdot (x + \bar{y_3} + \bar{y_2} + \bar{y_1}) \cdot (x + \bar{y_3} + \bar{y_2} + y_1) \cdot (\bar{x} + \bar{y_3} + y_2 + y_1)$
$\cdot (\bar{x} + \bar{y_3} + y_2 + \bar{y_1}) \cdot (\bar{x} + \bar{y_3} + \bar{y_2} + \bar{y_1}) \cdot (\bar{x} + \bar{y_3} + \bar{y_2} + y_1)$

- $(x + \bar{y_3} + y_2 + \cancel{y_1}) \cdot (x + \bar{y_3} + y_2 + \cancel{\bar{y_1}}) \cdot (x + \bar{y_3} + \bar{y_2} + \cancel{\bar{y_1}}) \cdot (x + \bar{y_3} + \bar{y_2} + \cancel{y_1}) \cdot (\bar{x} + \bar{y_3} + y_2 + \cancel{y_1})$
$\cdot (\bar{x} + \bar{y_3} + y_2 + \cancel{\bar{y_1}}) \cdot (\bar{x} + \bar{y_3} + \bar{y_2} + \cancel{\bar{y_1}}) \cdot (\bar{x} + \bar{y_3} + \bar{y_2} + \cancel{y_1})$

$(x + \bar{y_3} + \cancel{y_2}) \cdot (x + \bar{y_3} + \cancel{\bar{y_2}}) \cdot (\bar{x} + \bar{y_3} + \cancel{y_2}) \cdot (\bar{x} + \bar{y_3} + \cancel{\bar{y_2}})$

$(x + \bar{y_3}) \cdot (\bar{x} + \bar{y_3}) = [\bar{y_3}]$

- $(\bar{x} + \bar{y_3} + \bar{y_2} + \cancel{\bar{y_1}}) \cdot (\bar{x} + \bar{y_3} + \bar{y_2} + \cancel{y_1}) \cdot (\bar{x} + y_3 + \bar{y_2} + \cancel{\bar{y_1}}) \cdot (\bar{x} + y_3 + \bar{y_2} + \cancel{y_1})$

$(\bar{x} + \cancel{\bar{y_3}} + \bar{y_2}) \cdot (\bar{x} + \cancel{y_3} + \bar{y_2})$

$(\bar{x} + \bar{y_2}) = [\bar{x} + \bar{y_3}]$

TUTTO INSIEME: $(y_2 + y_1) \cdot (x + \bar{y_1}) \cdot \bar{y_3} (\bar{x} + \bar{y_2})$

IMPLEMENTAZIONE y_1'

$x y_3 \backslash y_2 y_1$	00	01	11	10
00	1	1	0	1
01	1	1	–	–

$\longrightarrow$

$x y_3 \backslash y_2 y_1$	00	01	11	10
00	1	1	0	1
01	1	1	–	–

INPUT				OUTPUT			
X	y_3	y_2	y_1	y_3'	y_2'	y_1'	Z
0	0	0	0	0	0	1	0
0	0	0	1	1	0	1	0
0	0	1	0	0	1	1	0
0	0	1	1	1	0	0	0
0	1	0	0	1	0	1	0
0	1	0	1	1	0	1	0

00	1	1	0	1
01	1	1	–	–
11	1	1	–	–
10	1	0	1	1

→

00	1	1	0	1
01	1	1	–	–
11	1	1	–	–
10	1	0	1	1

0	0	1	0	0	1	1	0
0	0	1	1	1	0	0	0
0	1	0	0	1	0	1	0
0	1	0	1	1	0	1	0
0	1	1	0	–	–	–	–
0	1	1	1	–	–	–	–
1	0	0	0	1	0	1	0
1	0	0	1	0	1	0	0
1	0	1	0	1	0	1	0
1	0	1	1	1	0	1	0
1	1	0	0	1	0	1	1
1	1	0	1	1	0	1	0
1	1	1	0	–	–	–	–
1	1	1	1	–	–	–	–

Saltando i passaggi e arrivando subito alla soluzione, ottengo:

$$(\overline{y_1} + y_3)$$

IMPLEMENTAZIONE Z

senza fare la MAPPA DI KARNAUGH SCRIVIAMO DIRETTAMENTE LA SEMPLIFICAZIONE, Z ha solo un valore, applichiamo la regola dei MINTERMINI:

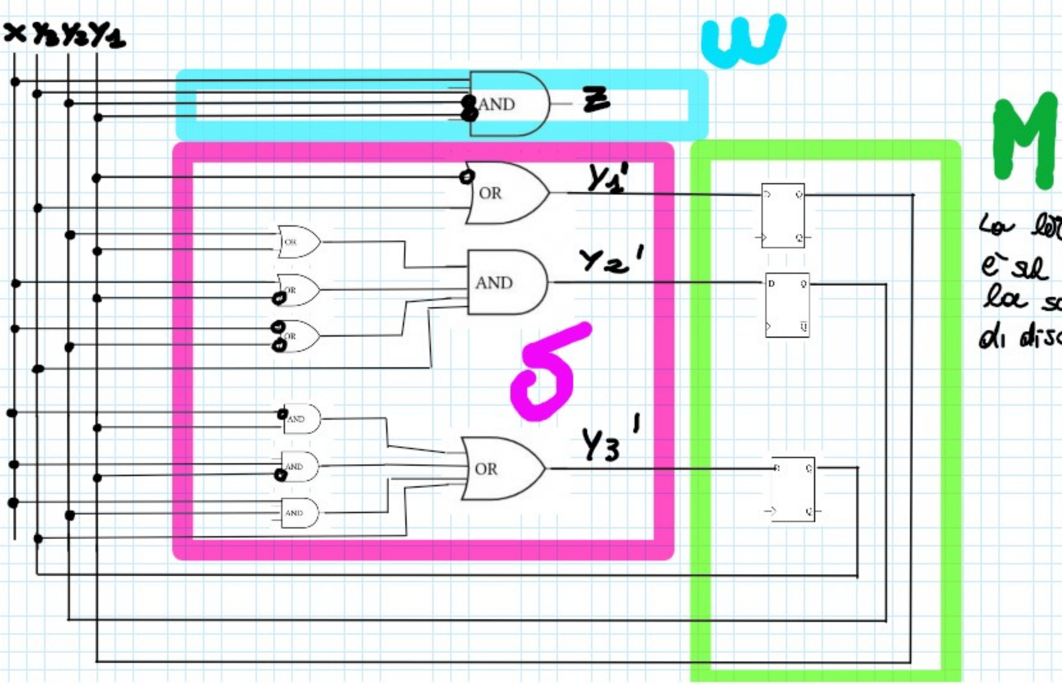
$$(x \cdot y_3 \cdot \overline{y_2} \cdot \overline{y_1})$$

RISCRIVIAMO:

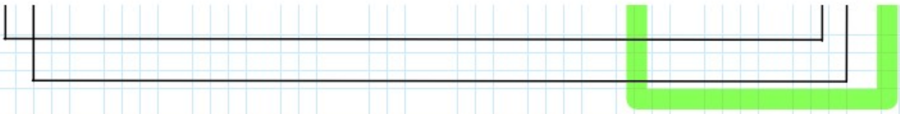
$$y_3' = \overline{x} y_1 + y_3 + x \overline{y_1} + x y_2$$
$$y_2' = (y_2 + y_1) \cdot (x + \overline{y_1}) \cdot \overline{y_3} (\overline{x} + \overline{y_2})$$
$$y_1' = (\overline{y_1} + y_3)$$
$$Z = (x \cdot y_3 \cdot \overline{y_2} \cdot \overline{y_1})$$

Z È L'OUTPUT.

COSTRUZIONE CIRCUITO (Partiamo dalla Z)!

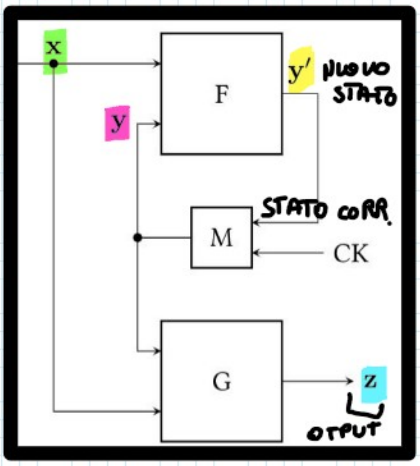


La lettura dei flip flop è sul fronte di salita e la scrittura sul fronte di discesa.

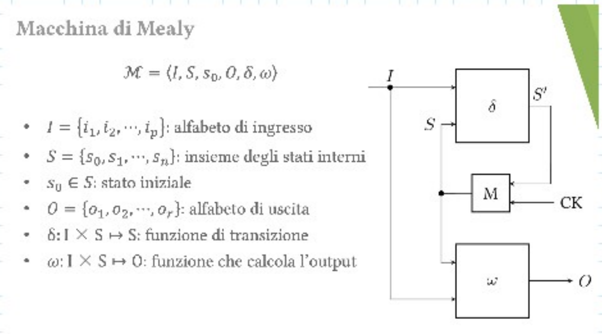


questo è y, al passo successivo sputeremo y nuovi, quindi io devo memorizzare questi valori in un flip flop **D**.

RETI SINCRONE



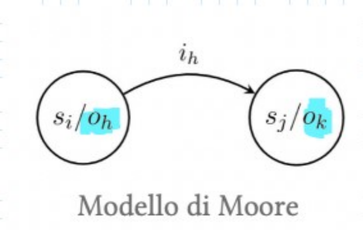
ω è la funzione che calcola l'output.



STESSO PROBLEMA: MOORE!

Di nuovo partirò da uno stato iniziale che chiamo zero, e l'uscita dipende soltanto dallo stato in cui mi trovo.

Se sono in zero, il mio output è NO.

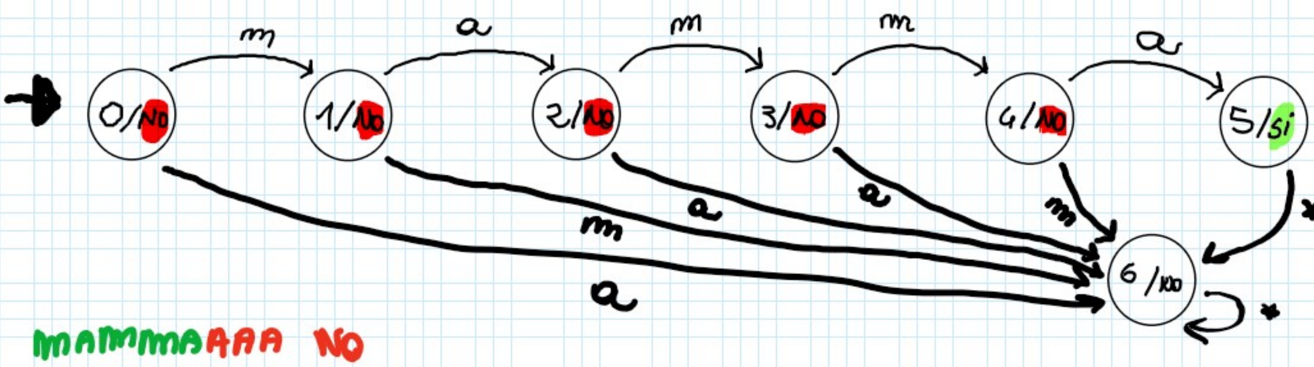


questa volta le transizioni non mi determinano l'output, ma lo stato in cui arrivo;

DA 4, SE RICEVO m, NON POSSO andare IN 5, perché non dipendiamo più dalla transizione, 5 ORAMAI è IMPOSTATO A SI. SE VADO IN 5 IL MIO OUTPUT diventa SI.

ALLA RICEZIONE di una m devo andare in un nuovo stato;

MOORE



MAMMMAAAA NO

SIA IL MODELLO DI MEALY E QUELLO DI MOORE, HANNO lo STESSO COMPORTAMENTO.

LE MACCHINE di MOORE TENDONO A CONSUMARE PIÙ STATI.
LA TABELLA CAMBIA E ANCHE IL CIRCUITO SARÀ DIFFERENTE.

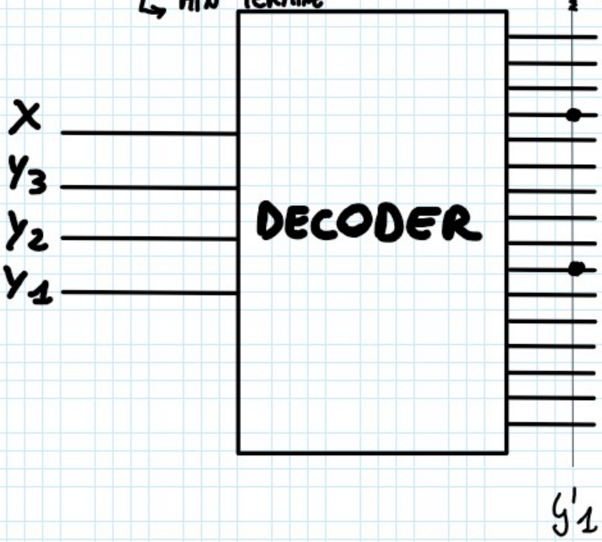
Noi trasformiamo una sequenza di INPUT in una sequenza di OUTPUT:

m a m m a ...
NO NO NO NO SI NO...

TRA UN PO' IMPLEMENTEREMO LA CU SULLA CPU.

COSTRUIAMOCI ANCHE UNA ROM:

ciascuna Riga è una linea dell'output della tabella di verità.
↳ MIN TERMINE



→ lo manda a 1
↓
L'USCITA è SEMPRE 1 SU y'_1 INDIPENDENTEM. DALL'INPUT sul decoder.

INPUT				OUTPUT			
X	y_3	y_2	y_1	y_3'	y_2'	y_1'	Z
0	0	0	0	0	0	1	0
0	0	0	1	1	0	1	0
0	0	1	0	0	1	1	0
0	0	1	1	1	0	0	0
0	1	0	0	1	0	1	0
0	1	0	1	1	0	1	0
0	1	1	0	-	-	-	-
0	1	1	1	-	-	-	-
1	0	0	0	1	0	1	0
1	0	0	1	0	1	0	0
1	0	1	0	1	0	1	0
1	0	1	1	1	0	1	0
1	1	0	0	1	0	1	1
1	1	0	1	1	0	1	0
1	1	1	0	-	-	-	-
1	1	1	1	-	-	-	-

in quali configurazioni questa deve andare a zero? quali sono i MIN-TERMINI che mandano a zero y'_1.

$$\begin{bmatrix} 0011 \\ 1001 \end{bmatrix} \begin{matrix} = 3 \\ = 9 \end{matrix}$$

LA 3ª e LA 9ª RIGA PORTANO a ZERO y'_1.

↳ INSERISCO UN TRANSISTOR che le collega a massa.
↓
Si APRE e ho una caduta di tensione, e in uscita ho zero.

COSÌ RISPARMIO IL numero di componenti e il circuito è più efficiente, con la ROM.

MEALY e MOORE SONO equivalenti e si POSSONO TRASFORMARE l'UNO CON l'altro.

DA MEALY A MOORE:

- INPUT e OUTPUT sono gli stessi;
- GLI STATI IN generale SONO gli stessi, ma attenzione ad aggiungere stati.
  SE RAGGIUNGO LO STESSO STATO CON TRANSIZIONI differenti, questo stato lo devo sdoppiare.

DA MOORE A MEALY, SPOSTO gli OUTPUT dagli stati alle transizioni.
MA BISOGNA vedere se c'è una macchina minima!

↳ ma non le vedremo

FINE RETI SEQUENZIALI