# Lezione 14 Reti sequenziali 2 + 3

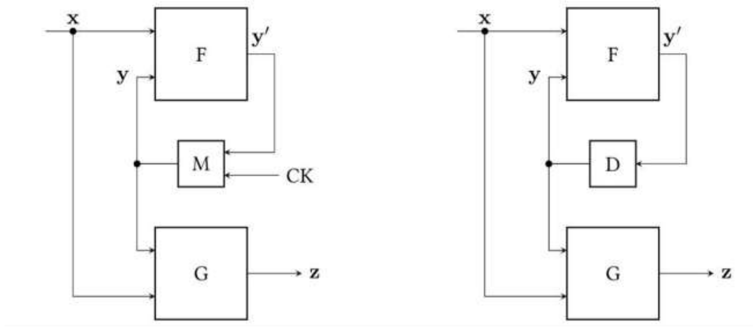
giovedì 26 ottobre 2023 10:17

Le reti sequenziali sono una macro famiglia delle reti combinatorie viste finora . Quindi le relazioni che caratterizzano questi circuiti sono molteplici : infatti non vi è una sola equazione ma bensi' due : sistema di equazioni booleane , in quanto il circuito evolve nel tempo. Quindi si parla di stati interni diversi del circuito : output viene chiamato Z che viene calcolata con la funzione booleana G (che non accetta solo X ma anche Y (vettore)), mentre vi è altra funzione F che accetta input e stato interno (vettore), la quale determinerà una transizione di stato ( F calcola funzione in cui si troverà il nostro circuito ): vediamo un esempio il latch S-R:

| *S* | *R* | *Q* | *Q'* |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | indeterminato |
| 1 | 1 | 1 | indeterminato |

$$\begin{cases} \mathbf{y}' = f(\mathbf{x}, \mathbf{y}) \\ \mathbf{z} = g(\mathbf{x}, \mathbf{y}) \end{cases}$$

Ci sono due categorie di reti sequenziali : **sincrone ed asincrone** . Nelle prime si calcola la transizione in determinati istanti di tempo ben precisi e controllabili dall'esterno (clock) , mentre nelle seconde non si ha il controllo delle transizioni : circuiti analogici .

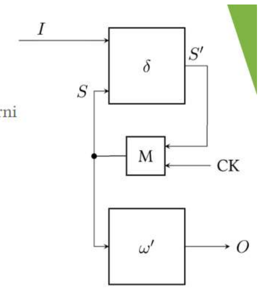



F e G sono blocchi logici che implementano funzioni logiche , attraverso circuiteria . Nel primo caso (sx) servono degli elementi per mantenere le info quindi devo usare dei ff ( visto che il latch non supporta 1-1) . Per calcolare il tempo di campionamento devo aspettare almeno il ritardo di propagazione di F. Per quanto riguarda invece il modulo M come faccio a garantire che non vi sia un valore sballato? Uso il fronte di commutazione per attivare lettura / scrittura : magari scrittura su fronte discesa, mentre lettura in salita . Mentre nel secondo esempio il blocco D (delay) , non ho intervalli ben precisi per manipolare informazioni: dopo che il processamento è finito e ritardo del blocco D , il segnale sarà scritto/letto ( asincrono). Andiamo ora a definire le reti F e G: andiamo a definire le **Macchine (automi) a stati finiti** : modello matematico che descrive l'evoluzione del sistema nel tempo . E' un modello astratto che mostra in un dato istante il sistema si trova in uno stato ben preciso , ed attraverso gli **eventi** ( condizione che in un dato istante fa succedere qualcosa ) effettua una **transizione di stato. Le transizioni di stato vengono determinate dal blocco F** . Anche con questi modelli si possono determinare degli output. Due varianti principali : **Mealy E Moore :** differiscono nel modo di generare output : nel primo output dipende dallo stato e dalla transizione innescata , mentre nel secondo dipende unicamente da output. Vediamoli nel dettaglio :

1. Moore ( più stati di Mealy)

a. $\mathcal{M} = \langle I, S, s_0, O, \delta, \omega \rangle$

- $I = \{i_1, i_2, \cdots, i_p\}$: alfabeto di ingresso
- $S = \{s_0, s_1, \cdots, s_n\}$: insieme degli stati interni
- $s_0 \in S$: stato iniziale
- $O = \{o_1, o_2, \cdots, o_r\}$: alfabeto di uscita
- $\delta$: I × S ↦ S: funzione di transizione
- $\omega$: S ↦ O: funzione che calcola l'output

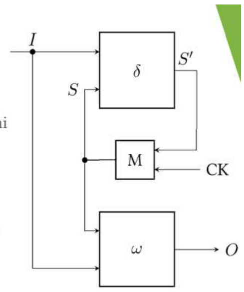



b. La funzione delta serve per calcolare il blocco F : dato stato ed input , permette di calcolare funzione "prossimo stato".
c. La funzione "omega" permette di calcolare l'output .

2. Mealy (uno stato più transizioni) :

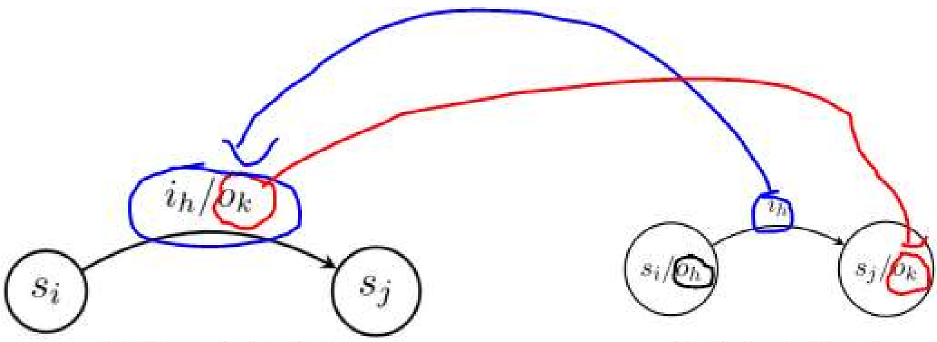
a. $\mathcal{M} = \langle I, S, s_0, O, \delta, \omega \rangle$

- $I = \{i_1, i_2, \cdots, i_p\}$: alfabeto di ingresso
- $S = \{s_0, s_1, \cdots, s_n\}$: insieme degli stati interni
- $s_0 \in S$: stato iniziale
- $O = \{o_1, o_2, \cdots, o_r\}$: alfabeto di uscita
- $\delta$: I × S ↦ S: funzione di transizione
- $\omega$: I × S ↦ O: funzione che calcola l'output




b. Cambia "omega" necessita di stato attuale e di input : caso più generale di ASF.

Vediamo ora delle rappresentazione per questo formalismo : o **diagramma a stati** o **tabella degli stati** : il primo rappresenta un grafo che rappresenta le relazioni tra gli stati ( cerchietti) e transizioni :




Modello di Mealy

Modello di Moore

Mentre per la seconda associa degli stati a delle transizioni in funzione dell'input e determina output : riga sono gli stati , mentre sulla prima colonna gli input :

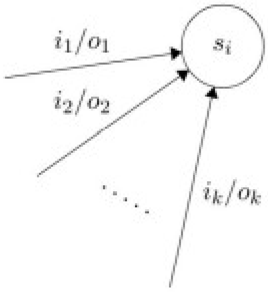
| | $i_1$ | $i_2$ | $\dots$ | $i_j$ | $\dots$ | $i_p$ |
|---|---|---|---|---|---|---|
| $s_1$ | | | | | | |
| $s_2$ | | | | | | |
| $\vdots$ | | | | | | |
| $s_i$ | | | | $\delta(i_i, s_i)/\omega(i_j, s_i)$ | | |
| $\vdots$ | | | | | | |
| $s_n$ | | | | | | |

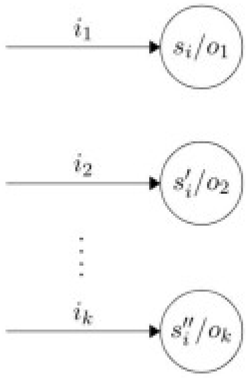
Modello di Mealy

| | $i_1$ | $i_2$ | $\dots$ | $i_j$ | $\dots$ | $i_p$ | $\omega'$ |
|---|---|---|---|---|---|---|---|
| $s_1$ | | | | | | | |
| $s_2$ | | | | | | | |
| $\vdots$ | | | | | | | |
| $s_i$ | | | | $\delta(i_i, s_i)$ | | | $\omega'(i_j, s_i)$ |
| $\vdots$ | | | | | | | |
| $s_n$ | | | | | | | |

Modello di Moore

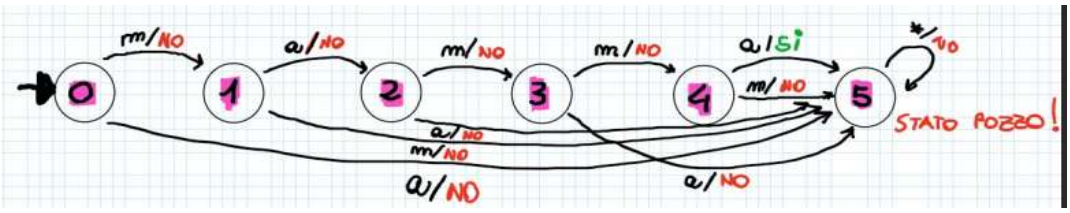
**Possibile passare da Mealy a Moore e viceversa in quanto sono equivalenti** .




Vediamo ora la sintesi : cominciamo dal blocco M (memoria) : lo creiamo con una serie di ff di tipo D (il numero dipende dal vettore Y), per quanto riguarda la funzione "delta" è necessario realizzare un circuito di commutazione per aggiornare lo stato di tutti i ff ( **equazione di eccitazione di un ff** ) , mentre per la rete "omega" uso un circuito di commutazione per generare ciascun bit di output . Vediamo un esempio : Stringa in input "MAMMA" :

Input = { m,a} Output = {si , no}




In ogni stato si ha il "resoconto" di quanto fatto finora (memorizza sequenza input finora) . <u>Da notare il "*" sullo stato 5 : qualunque carattere dopo la stringa "MAMMA" non è valido!!</u> . Andiamo a vedere ora 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.

$$\underbrace{1}_{\delta}/\underbrace{NO}_{\omega}$$

| I | X |
|---|---|
| m | 0 |
| a | 1 |

| O | Z |
|---|---|
| Si | 1 |
| No | 0 |

A volte gli input e gli output, sono già delle codifiche. Li non la posso scegliere.

Metto tutto insieme : uso il binomiale per vedere quante variabili necessitano : ne servono 6 : in quanto binomiale (4 2) = (4*3)/2 .

| 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 | - | - | - | - |

| $y_2x_0$ \ $y_0y_1$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | - | 1 |
| 01 | 1 | 1 | - | 1 |
| 11 | 0 | 1 | - | 1 |
| 10 | 1 | 1 | - | 1 |

$$y_0' = y_0 + \bar{y}_2 x_0 + y_2 y_0 + y_2 \bar{x}_0$$

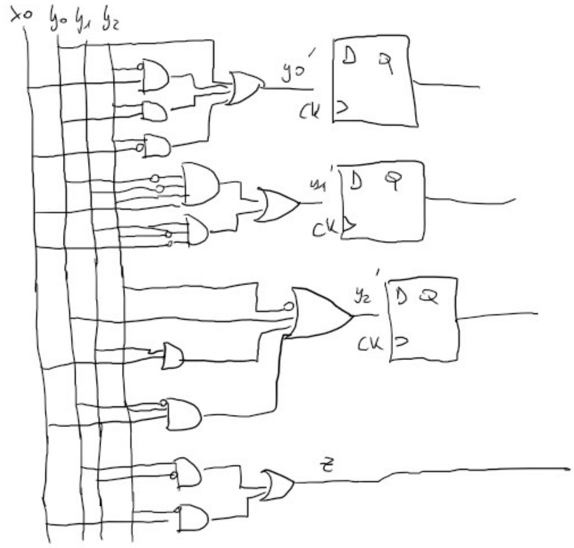
| $y_2x_0$ \ $y_0y_1$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 1 | - | 0 |
| 01 | 0 | 0 | - | 0 |
| 11 | 1 | 0 | - | 0 |
| 10 | 0 | 0 | - | 0 |

$$y_1' = \bar{y}_0 \bar{y}_1 y_2 x_0 + y_1 \bar{y}_2 \bar{x}_0$$

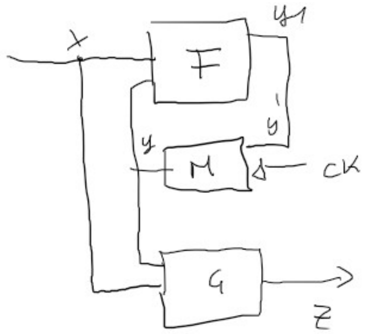
| $y_2x_0$ \ $y_0y_1$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | - | 1 |
| 01 | 1 | 1 | - | 1 |
| 11 | 0 | 1 | - | 1 |
| 10 | 1 | 0 | - | 1 |

$$y_2' = \bar{y}_2 + y_0 + y_1 x_0 + \bar{y}_1 y_2 x_0$$

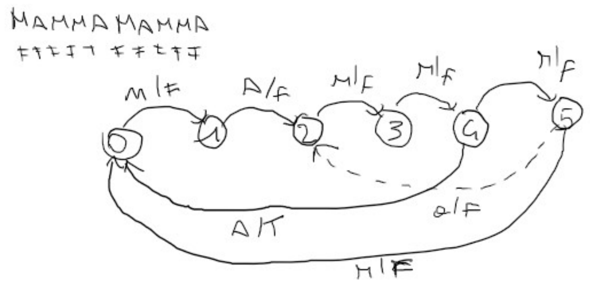
Mentre per z (uscita) visto che è unico mintermine , prendo la codifica :




Con schema concettuale :




Vediamo ora l'automa senza stato pozzo (stato 5) sequenza di 5 caratteri sia mamma:




Mentre se voglio che le ultime 5 lettere siano mamma :