

- I circuiti che implementano i flip flop sono semplici **reti sequenziali**
- Una rete sequenziale ha un funzionamento che **evolve nel tempo**

MODIFICA IL SUO STATO IN FUNZIONE DEGLI INPUT CHE RICEVE.
IN GENERALE UNA RETE SEQUENZIALE HA BISOGNO DI AVERE UNO STATO INTERNO PER DESCRIVERE QUELLO CHE SUCCEDDE NEL TEMPO.
PERCHÉ UNA RETE SEQUENZIALE **EVOLVE NEL TEMPO**.

LO STATO INTERNO, IN COMBINAZIONE CON L'INPUT, DETERMINA LO STATO SUCCESSIVO. LO ABBIAMO VISTO CON IL LATCH.

DOBBIAMO CERCARE DI PROGETTARE delle RETI, CHE RISPONDANO AD EQUAZIONI DI QUESTO TIPO:

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

HAI UNO y CHE È IL TUO STATO ATTUALE del circuito, RICEVI UN INPUT, QUESTO INPUT IN COMBINAZIONE CON IL SUDDETTO STATO, PROVOCHERÀ UN AGGIORNAMENTO DI STATO, E L'OUTPUT del circuito dipende dallo STATO IN CUI TI TROVI, dall'INPUT che RICEVI.

QUESTO È QUELLO CHE VOGLIAMO COSTRUIRE.

ESATTAMENTE COME ABBIAMO VISTO PER IL FLIP FLOP;

ESISTONO DUE CLASSI DI RETI SEQUENZIALI CHE POSSIAMO USARE PER RAPPRESENTARE QUESTO MODO:

• SINCRONE

• ASINCRONE

LA DIFFERENZA STA NEL MOMENTO IN CUI, IN UNA CERTA RETE/CIRCUITO, SI ESEGUE UNA TRANSIZIONE DI STATO, OSSIA SI PASSA DALLO STATO CORRENTE A QUELLO SUCCESSIVO.

SINCRONA: FACCIAMO AVVENIRE LA TRANSIZIONE DI STATO, AGGIORNIAMO LO STATO INTERNO DELLA NOSTRA RETE, CON UN CONTROLLO ESTERNO CHE IDENTIFICA GLI STATI TEMPORALI BEN PRECISI, IN CUI LA TRANSIZIONE DI STATO DEVE CAMBIARE.

DA QUALCHE PARTE ABBIAMO UNA MEMORIA M ; SOTTO FORMATA DA ELEMENTI CHE CODIFICANO E MEMORIZZANO LO STATO ATTUALE DELLA NOSTRA MACCHINA, QUANDO AGGIORNIAMO LO STATO IN ISTANTI TEMPORALI BEN PRECISI. QUINDI UTILIZZEREMO IL SEGNALE DI CLOCK CAMPIONATO O SUL FRONTE DI SALITA O SUL FRONTE DI DISCESA.

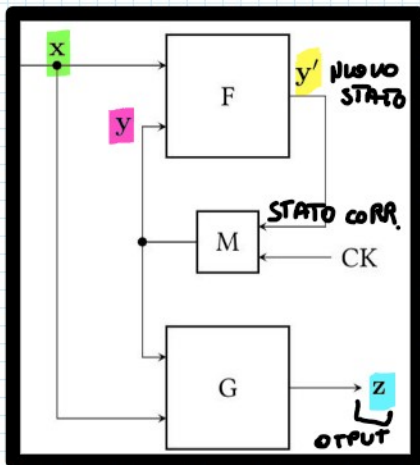
ASINCRONA: NON CONTROLLIAMO ESTERNAMENTE LE TRANSIZIONI DI STATO.

SALTA O SUL FRONTE DI DISCESA.

ASINCRONA: NON CONTROLLIAMO ESTERNAMENTE LE TRANSIZIONI DI STATO;
È IL LATCH, POTEVA ANDARE IN UNO STATO OSCILLANTE.
NON AVEVAMO modo di dire "qual'è il momento in cui voglio effettuare una transizione di stato".

QUINDI LA MEMORIA DIVENTA UN DELAY.

VEDREMO LE RETI SINCRONE:



È UN DISEGNO A BLOCCHI DI:

HO UN USCITA, HO UN INPUT, HO
UNO STATO INTERNO E HO
UN NUOVO STATO.

GLI INPUT SONO LE FUNZIONI F E G.

F E G SARANNO DEI CIRCUITI COMBINATORI.

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

INPUT ATTUALE

↓
LEGGONO L'INPUT, LEGGONO LO STATO
CORRENTE, CALCOLANO RISPETTIVAMENTE
L'OUTPUT DEL CIRCUITO E IL NUOVO STATO
IN CUI CI VA A FINIRE.

QUESTO NUOVO STATO VIENE MEMORIZZATO DENTRO LA MEMORIA IN MANIERA
SINCRONA CONTROLLATA DALL'ESTERNO E TRAMITE IL CLOCK, UNA VOLTA CHE SIANO
SICURI CHE C'È STATA UNA STABILIZZAZIONE, AGGIORNIAMO LO STATO;

LO STRUMENTO MATEMATICO CHE ANDREMO ad utilizzare per progettare
MACCHINE di questo TIPO, SI CHIAMA:

MACCHINE A STATI FINITI

È UN MODELLO MATEMATICO CHE PERMETTE DI IMPLEMENTARE SU CARTA, OGGETTI DI QUESTO
TIPO;

UN AUTOMA, IN UN DATO ISTANTE TEMPORALE SI TROVA IN UN DATO STATO.

GLI STATI SONO FINITI.

LA MACCHINA PUÒ CAMBIARE DI STATO, CIOÈ SE APPLICHO UN INPUT DIVERSO, IL MIO
BLOCCO F POTREBBE CAMBIARE DI STATO.

LA MACCHINA PUÒ GENERARE OUTPUT VERSO L'ESTERNO.

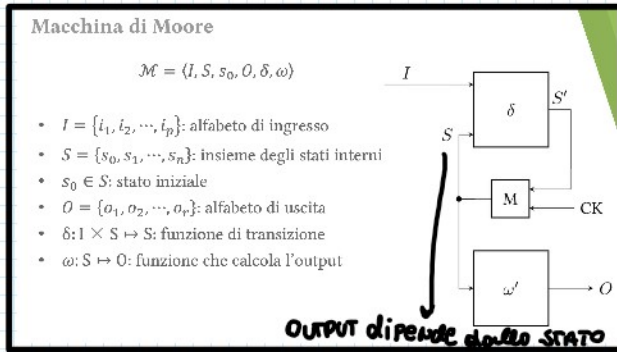
ESISTONO DUE TIPI DI MACCHINE A STATI FINITI:

• MACCHINA DI MOORE

LA DIFFERENZA STA NEL MODO IN CUI VIENE CALCOLATO L'OUTPUT.
SONO IDENTICHE COME FUNZIONAMENTO PER QUANTO RIGUARDA L'AGGIORNAMENTO DI STATO, MA IL MODO IN CUI VIENE CALCOLATO L'OUTPUT CAMBIA.

MOORE: L'OUTPUT, QUINDI LA Z, DIPENDE DALLO STATO.

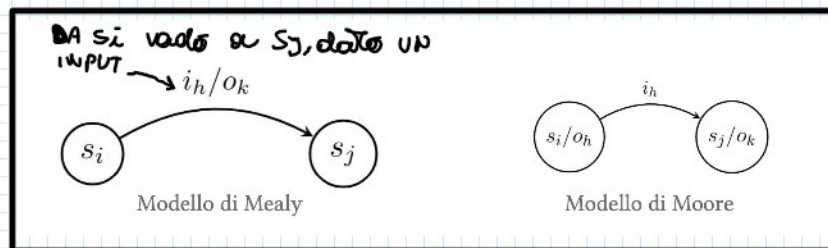
MEALY: L'OUTPUT DIPENDE DALLO STATO E DALLA TRANSIZIONE INNESCATATA.



IN OGNI ISTANTE IL MIO BLOCCO M
SARÀ TALE PER CUI CHE CIÒ CHE HA
COME STATO INTERNO, SARÀ LO STATO
IN CUI ESSA SI TROVA.

QUESTO STATO VERRÀ UTILIZZATO PER
TROVARE LO STATO SUCCESSIVO.
NEL MOMENTO IN CUI SIAMO IN
DETERMINATO STATO SIAMO IN GRADO
DI CALCOLARE L'OUTPUT.

ADOTTEREMO DEI GRAFI!



LA RAPPRESENTAZIONE MATEMATICA VERRÀ TRADOTTA IN CIRCUITO

Se Ricevi un dato input e sei in
questo stato, effettua una transizione.

L'OUTPUT DIPENDE SE USIAMO MEALY O MOORE.

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

L'OUTPUT DIPENDE UNICAMENTE dallo stato sulla macchina di Moore;

BASTA STARE IN UNO STATO PER GENERARE L'OUTPUT, PER SAPERE QUAL'È L'OUTPUT che devo andare a generare.

SERVIRÀ LA TABELLA degli stati e delle TRANSIZIONI PER PORTARLO IN UN CIRCUITO;

↙ INPUT

	i_1	i_2	...	i_j	...	i_p
s_1						
s_2						
\vdots						
s_i				$\delta(i_j, s_i) / \omega(i_j, s_i)$		
\vdots						
s_n						

Modello di Mealy

↙ INPUT

	i_1	i_2	...	i_j	...	i_p	ω'
s_1							
s_2							
\vdots							
s_i				$\delta(i_j, s_i)$			$\omega'(i_j, s_i)$
\vdots							
s_n							

Modello di Moore

HO UNA RIGA PER CIASCUNO STATO. 1:15:13

VEDIAMO UN ESEMPIO

Nella teoria della calcolabilità, la macchina di Mealy è un automa a stati finiti i cui valori di uscita sono determinati dallo stato attuale e dall'ingresso corrente, a differenza della macchina di Moore, che invece lavora solo in funzione dello stato corrente. Tuttavia, non per tutte le macchine di Mealy si può definire una macchina di Moore equivalente. In quanto il modello di Mealy basa lo stato d'uscita della macchina sia sullo stato in cui si trova, sia sugli input che riceve la macchina, mentre il modello di Moore è valido per le macchine che basano l'output soltanto sullo stato corrente della macchina, indifferentemente dagli input.