

# ARCHITETTURA DEGLI ELABORATORI

A.A. 2020-2021

Università di Napoli Federico II  
*Corso di Laurea in Informatica*

Docenti

Proff.

Luigi Sauro   gruppo 1 (A-G)

Silvia Rossi   gruppo 2 (H-Z)



# Logiche sequenziali sincrone

- I circuiti asincroni presentano delle criticità a volte difficilmente analizzabili
  - Dipendono dalla struttura fisica dei componenti
- Per questo si cerca di evitare di retroazionare l'output in maniera diretta e si interpone un registro nel ciclo di retroazione
- *Nell'ipotesi che il clock sia più lento del ritardo accumulato sul cammino, il registro consente al sistema di essere sincronizzato col clock: circuito *sincrono**

# Logiche sequenziali sincrone

- In generale un circuito sequenziale sincrono ha un insieme finito di stati  $\{S_0, \dots, S_{k-1}\}$
- Logica combinatoria:

$$\text{out} = f(\text{in})$$

- Logica sequenziale sincrona:

$$\text{out} = f(\text{in}, s_c)$$

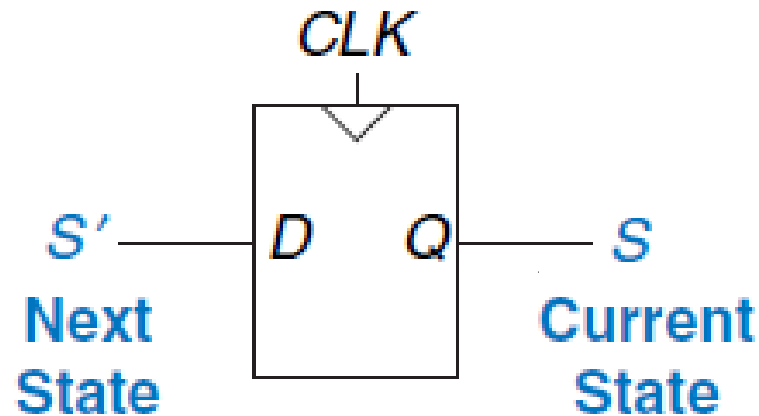
$$s_n = g(\text{in}, s_c)$$

# Design di logiche sequenziali sincrone

- Inserire registri nei cammini ciclici
- I registri determinano lo **stato**  $S_0, \dots, S_{k-1}$  del sistema
- I cambiamenti di stato sono determinati dalle transizioni del clock: il sistema è sincronizzato con il clock
- *Regole* di composizione:
  - Ogni componente è un registro o un circuito combinatorio
  - Almeno un componente è un registro
  - Tutti i registri sono sincronizzati con un unico clock
  - Ogni ciclo contiene almeno un registro
- Due tipici circuiti sequenziali sincroni
  - Finite State Machines (FSMs)
  - Pipelines

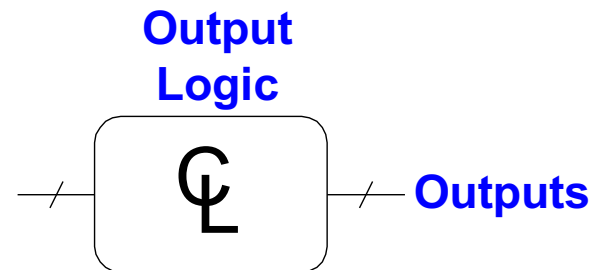
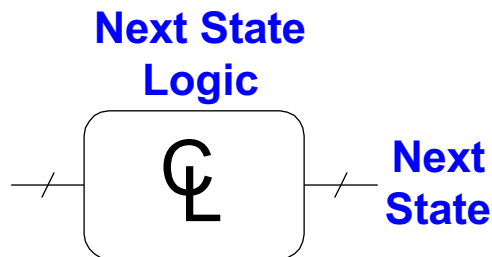
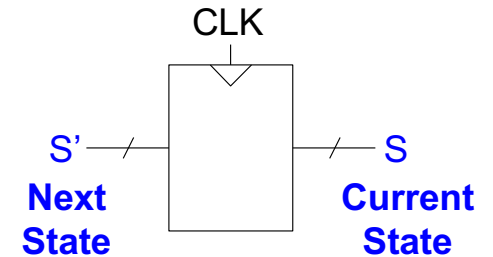
# Current state /Next state

- Un flip-flop D è il più semplice circuito sequenziale sincrono
  - $Q = s_c$
  - $D = s_n$



# Finite State Machines

- **State register**
  - Memorizzano lo stato corrente
  - Caricano il prossimo stato (clock edge)
- **Logica combinatoria**
  - “Computa” il prossimo stato (**g**)
  - “Computa” gli output (**f**)



# Finite State Machines

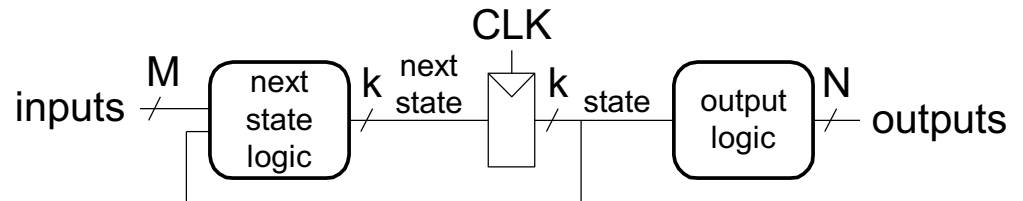
- $s_n$  dipende sia dall'input che da  $s_c$

$$s_n = g(in, s_c)$$

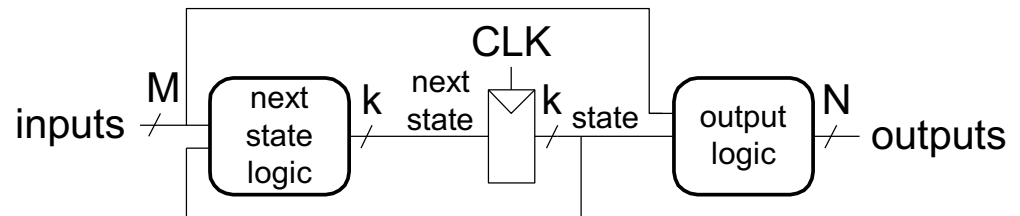
- 2 tipi di FSM a seconda della logica di output:

- Moore FSM:  $out = f(s_c)$
- Mealy FSM:  $out = f(in, s_c)$

Moore FSM

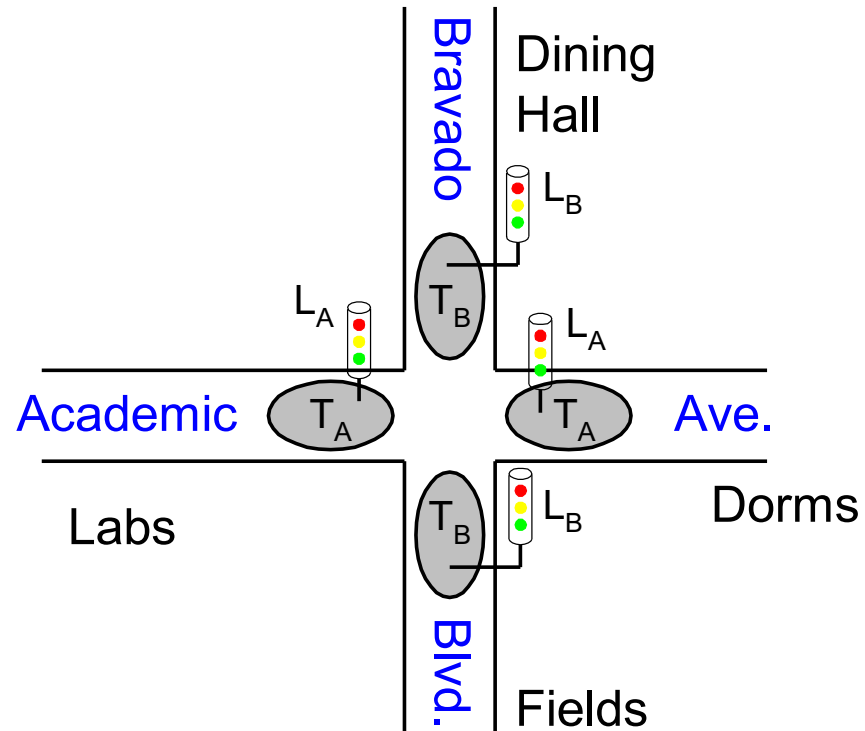


Mealy FSM



# Esempio: semaforo

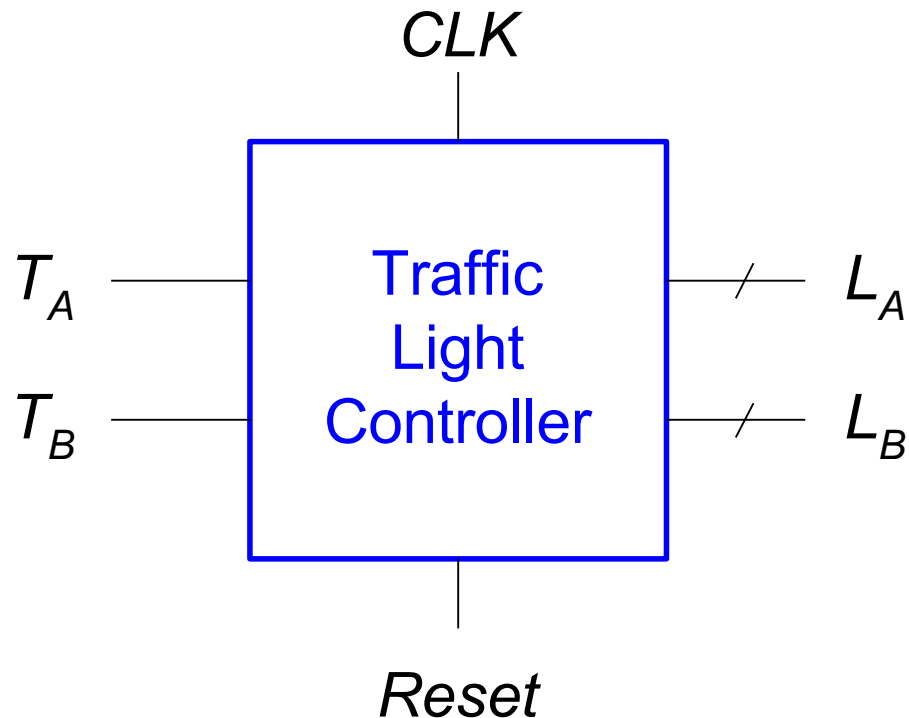
- Sensori:  $T_A$ ,  $T_B$  (TRUE quando c'è traffico)
- Luci:  $L_A$ ,  $L_B$





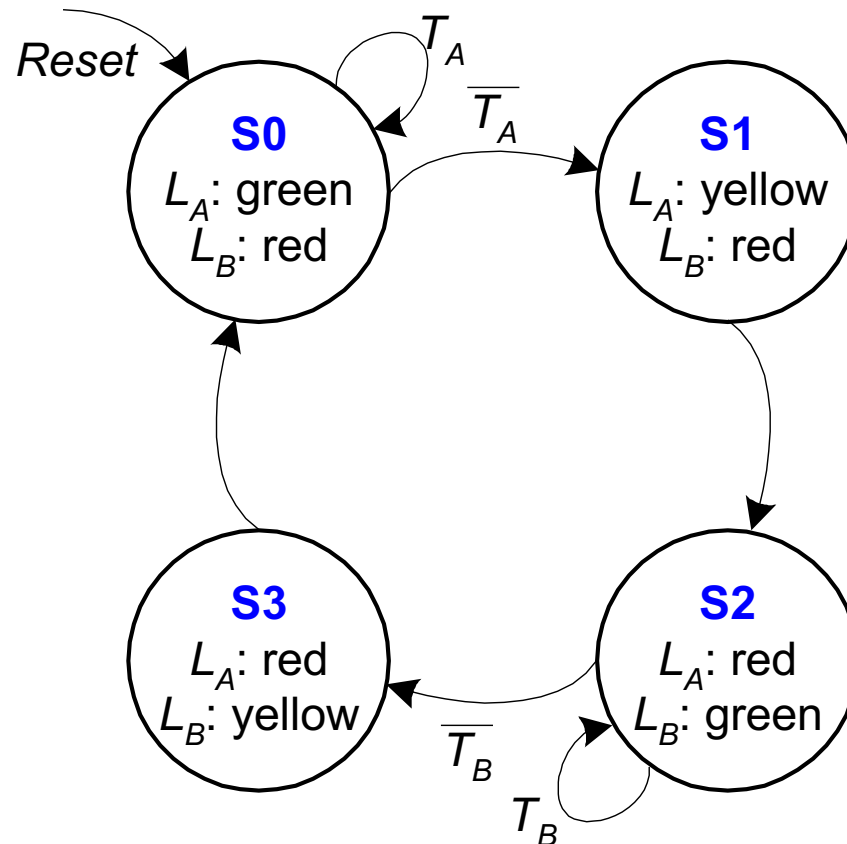
# Semaforo: *black box*

- Inputs:  $CLK$ ,  $Reset$ ,  $T_A$ ,  $T_B$
- Outputs:  $L_A$ ,  $L_B$



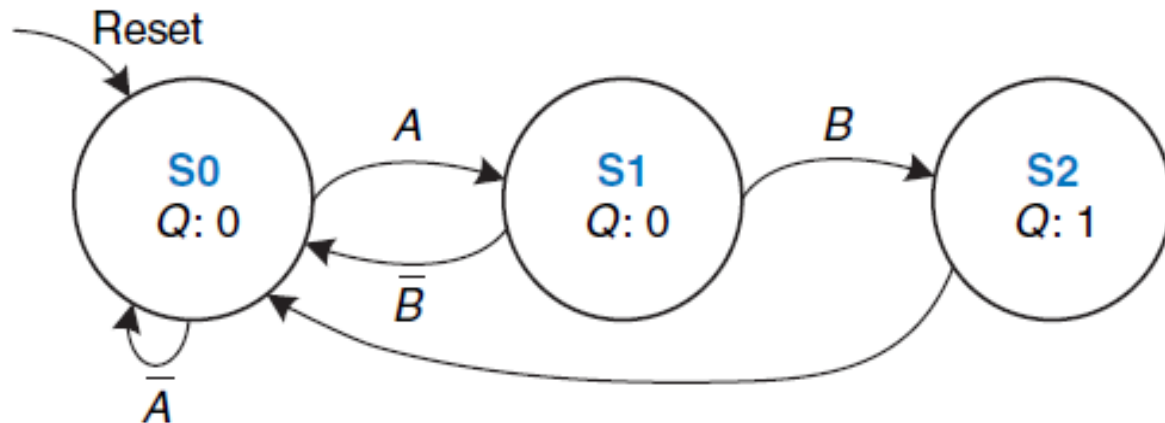
# Diagramma di transizione: Moore FSM

- **Stati:** labellati con gli outputs
- **Transizioni:** labellate con gli inputs



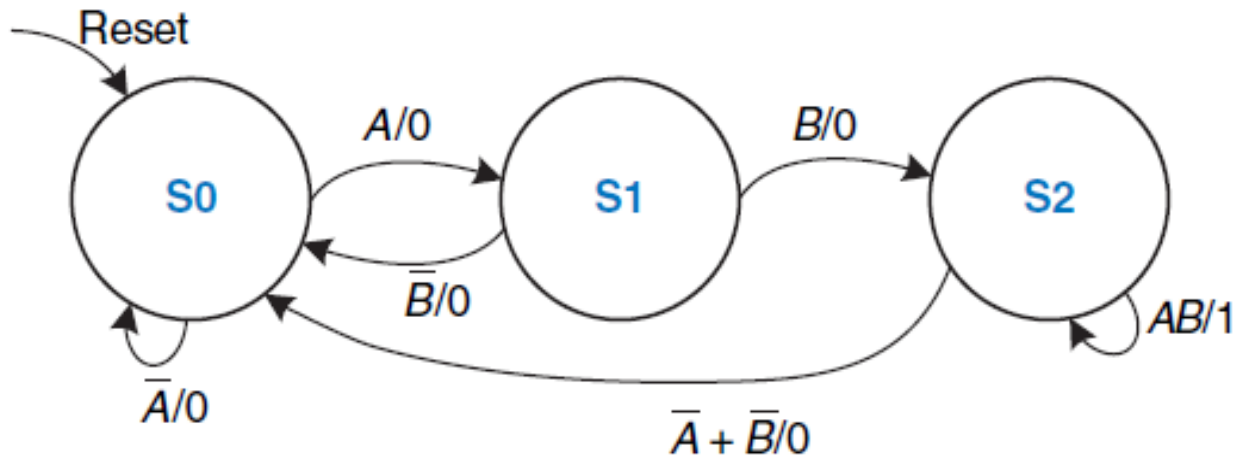
# Esempio Moore FSM

- Quale è il comportamento della FSM seguente?



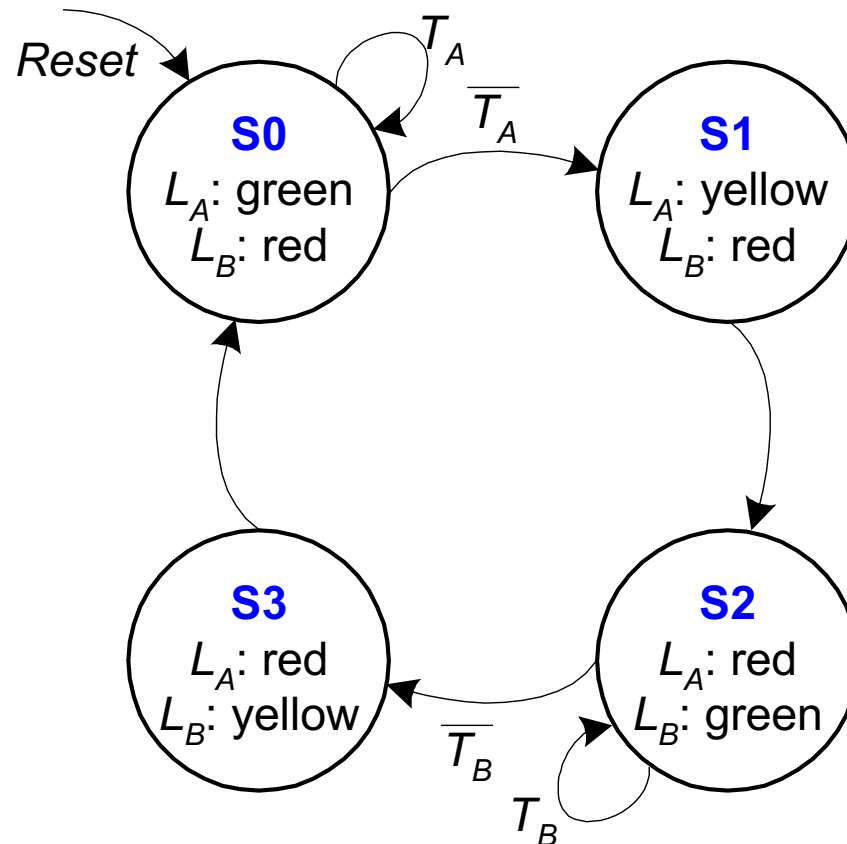
# Esempio Mealey FSM

- Quale è il comportamento della FSM seguente?



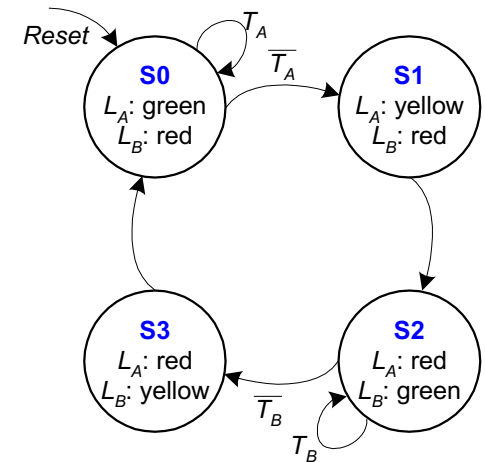
# Diagramma di transizione: Moore FSM

- **Stati:** labellati con gli outputs
- **Transizioni:** labellate con gli inputs



# FSM State Transition Table

Current State	Inputs		Next State
$S$	$T_A$	$T_B$	$S'$
S0	0	X	S1
S0	1	X	S0
S1	X	X	S2
S2	X	0	S3
S2	X	1	S2
S3	X	X	S0



# FSM State Transition Table

Current State	Inputs		Next State
$S$	$T_A$	$T_B$	$S'$
S0	0	X	S1
S0	1	X	S0
S1	X	X	S2
S2	X	0	S3
S2	X	1	S2
S3	X	X	S0

# FSM Encoded State Transition Table

Current State		Inputs		Next State	
$S_1$	$S_0$	$T_A$	$T_B$	$S'_1$	$S'_0$
0	0	0	X	0	1
0	0	1	X	0	0
0	1	X	X		
1	0	X	0		
1	0	X	1		
1	1	X	X		

State	Encoding
S0	00
S1	01
S2	10
S3	11



# FSM Encoded State Transition Table

Current State		Inputs		Next State	
$S_1$	$S_0$	$T_A$	$T_B$	$S'_1$	$S'_0$
0	0	0	X	0	1
0	0	1	X	0	0
0	1	X	X	1	0
1	0	X	0	1	1
1	0	X	1	1	0
1	1	X	X	0	0

State	Encoding
S0	00
S1	01
S2	10
S3	11

$$S'_1 = S_1 \oplus S_0$$

$$S'_0 = \overline{S_1} \overline{S_0} \overline{T_A} + S_1 \overline{S_0} \overline{T_B}$$

$$S'_1 = \underline{S_1} S_0 + S_1 * \underline{S_0} * \underline{T_B} + S_1 * \underline{S_0} * T_B$$

$$S'_1 = \underline{S_1} S_0 + S_1 \underline{S_0}$$

# FSM Output Table

Current State		Outputs			
$S_1$	$S_0$	$L_{A1}$	$L_{A0}$	$L_{B1}$	$L_{B0}$
0	0	0	0	1	0
0	1	0	1	1	0
1	0				
1	1				

Output	Encoding
green	00
yellow	01
red	10

# FSM Output Table

Current State		Outputs			
$S_1$	$S_0$	$L_{A1}$	$L_{A0}$	$L_{B1}$	$L_{B0}$
0	0	0	0	1	0
0	1	0	1	1	0
1	0	1	0	0	0
1	1	1	0	0	1

$$L_{A1} = S_1$$

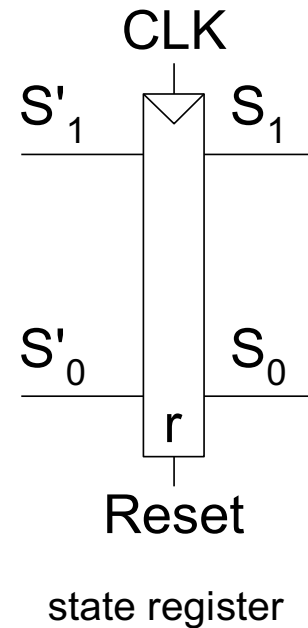
$$L_{A0} = \overline{S_1}S_0$$

$$L_{B1} = \overline{S_1}$$

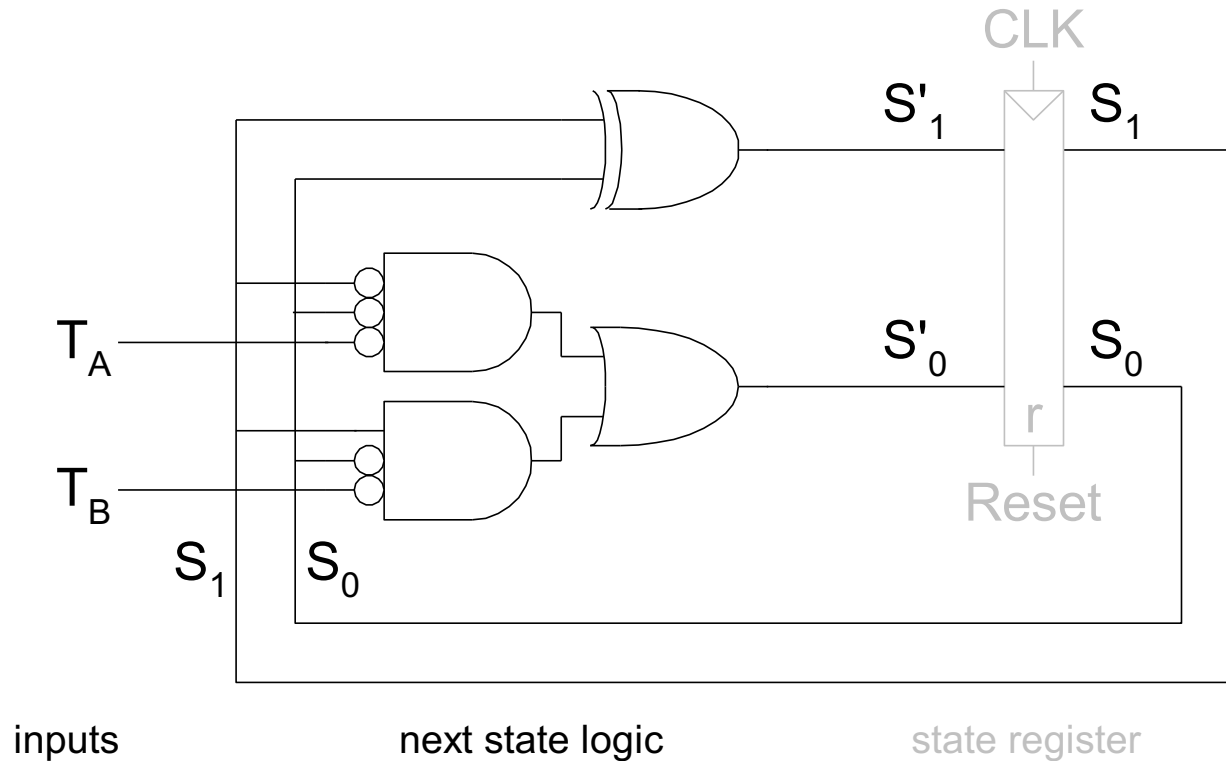
$$L_{B0} = S_1S_0$$

Output	Encoding
green	00
yellow	01
red	10

# FSM Schematic: State Register

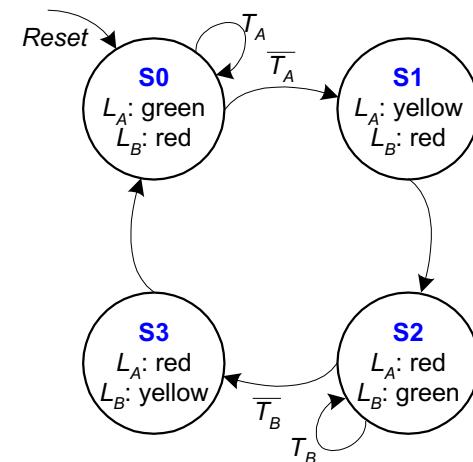


# Schema della logica di transizione

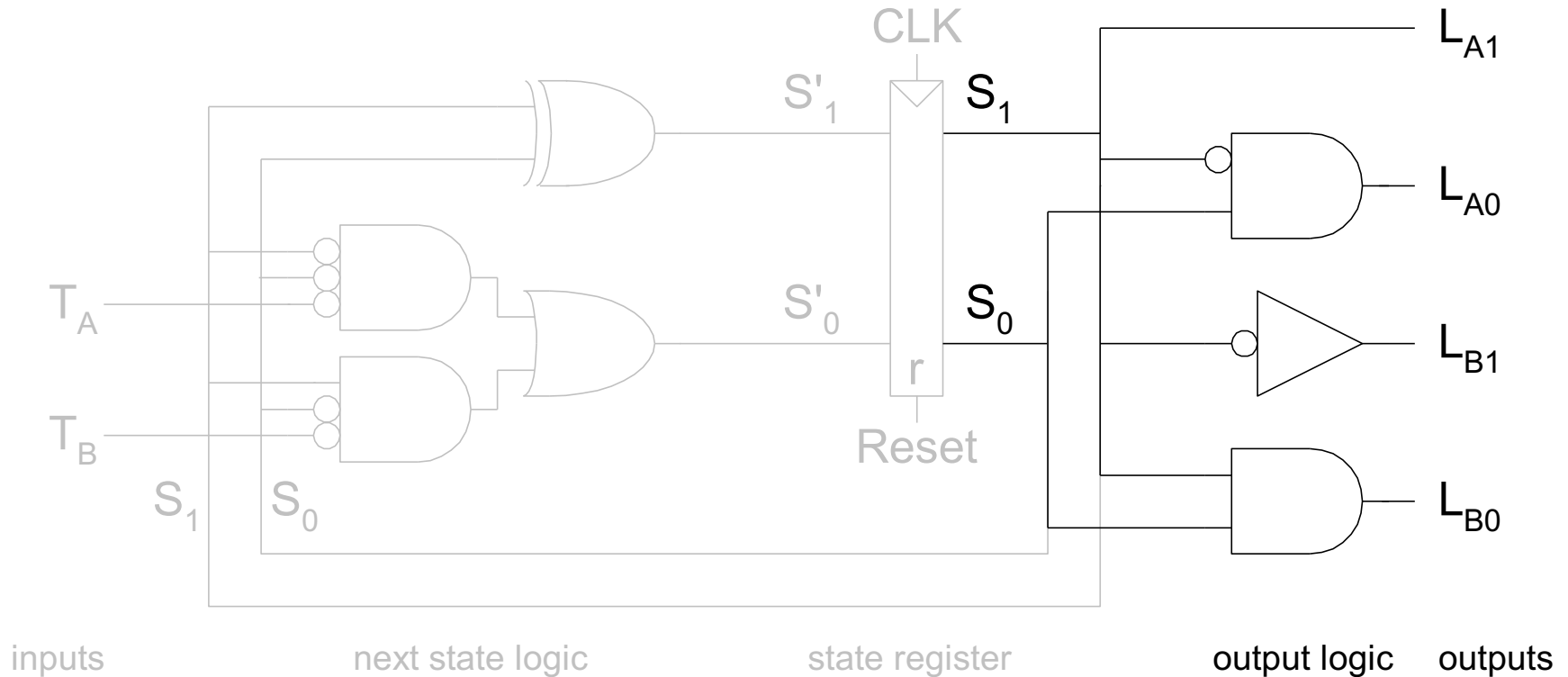


$$S'_1 = S_1 \oplus S_0$$

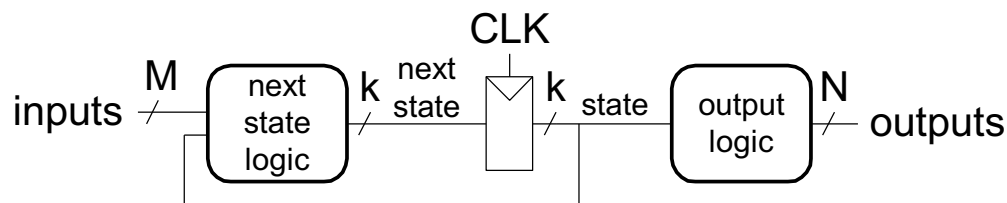
$$S'_0 = \overline{S_1} \overline{S_0} T_A + S_1 \overline{S_0} T_B$$



# Schema della logica di output



Moore FSM



$$L_{A1} = S_1$$

$$L_{A0} = \overline{S_1} S_0$$

$$L_{B1} = \overline{S_1}$$

$$L_{B0} = S_1 S_0$$

# FSM Timing Diagram

