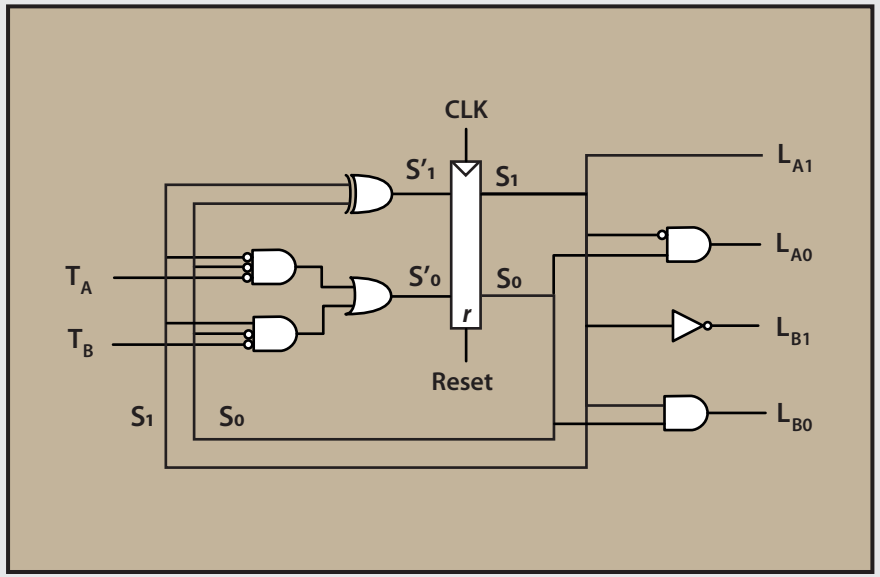
Computer Architecture

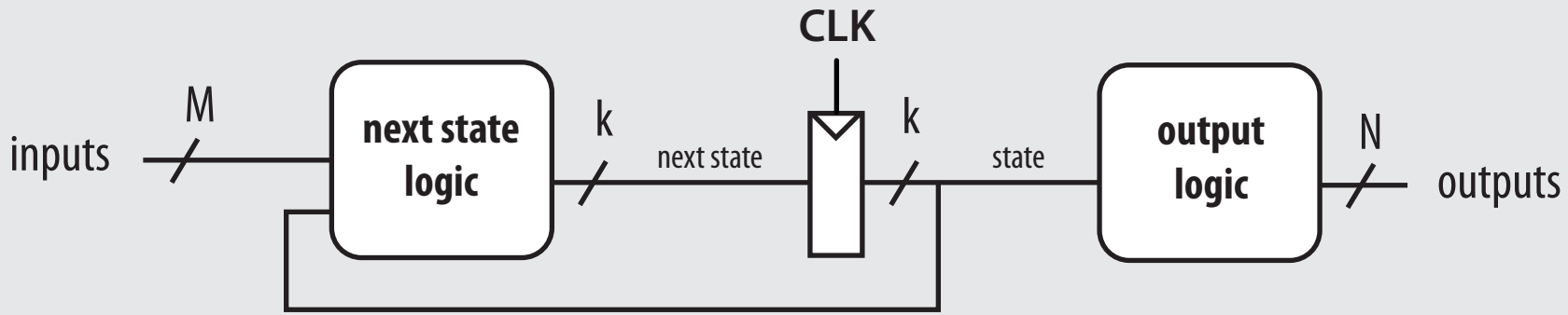
Paul Mellies

Lecture 11 : Finite State Machines

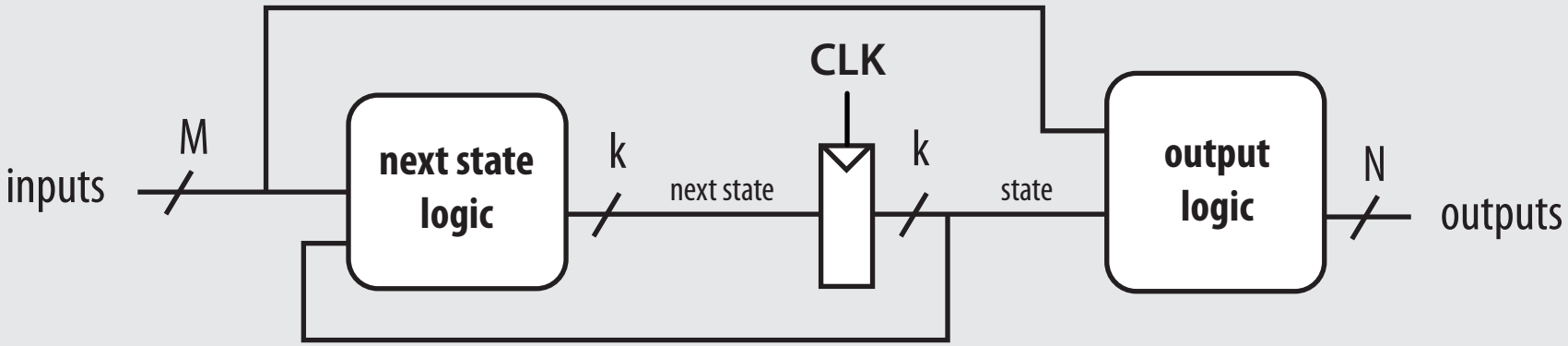


Finite State Machines

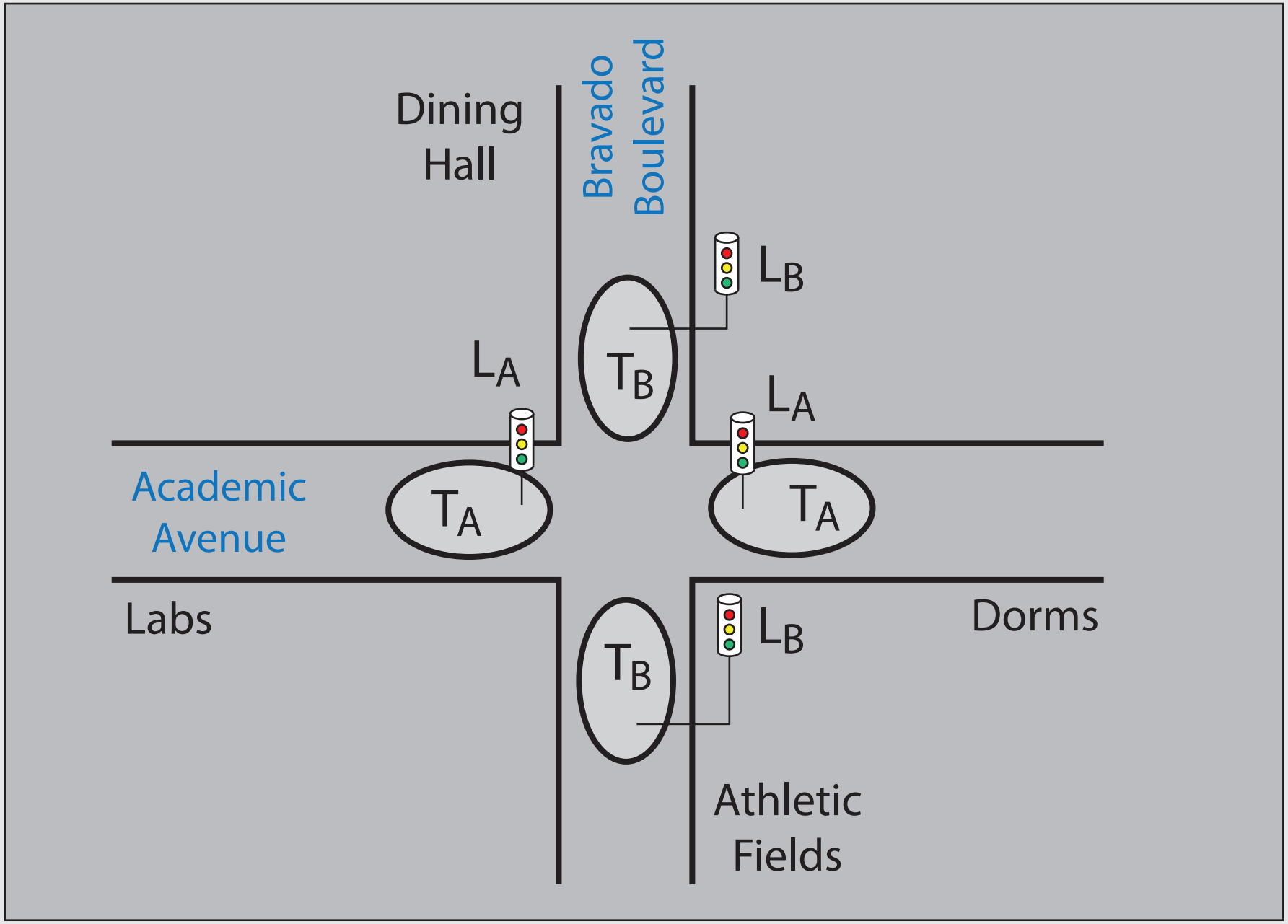
Moore machines



Mealy machines

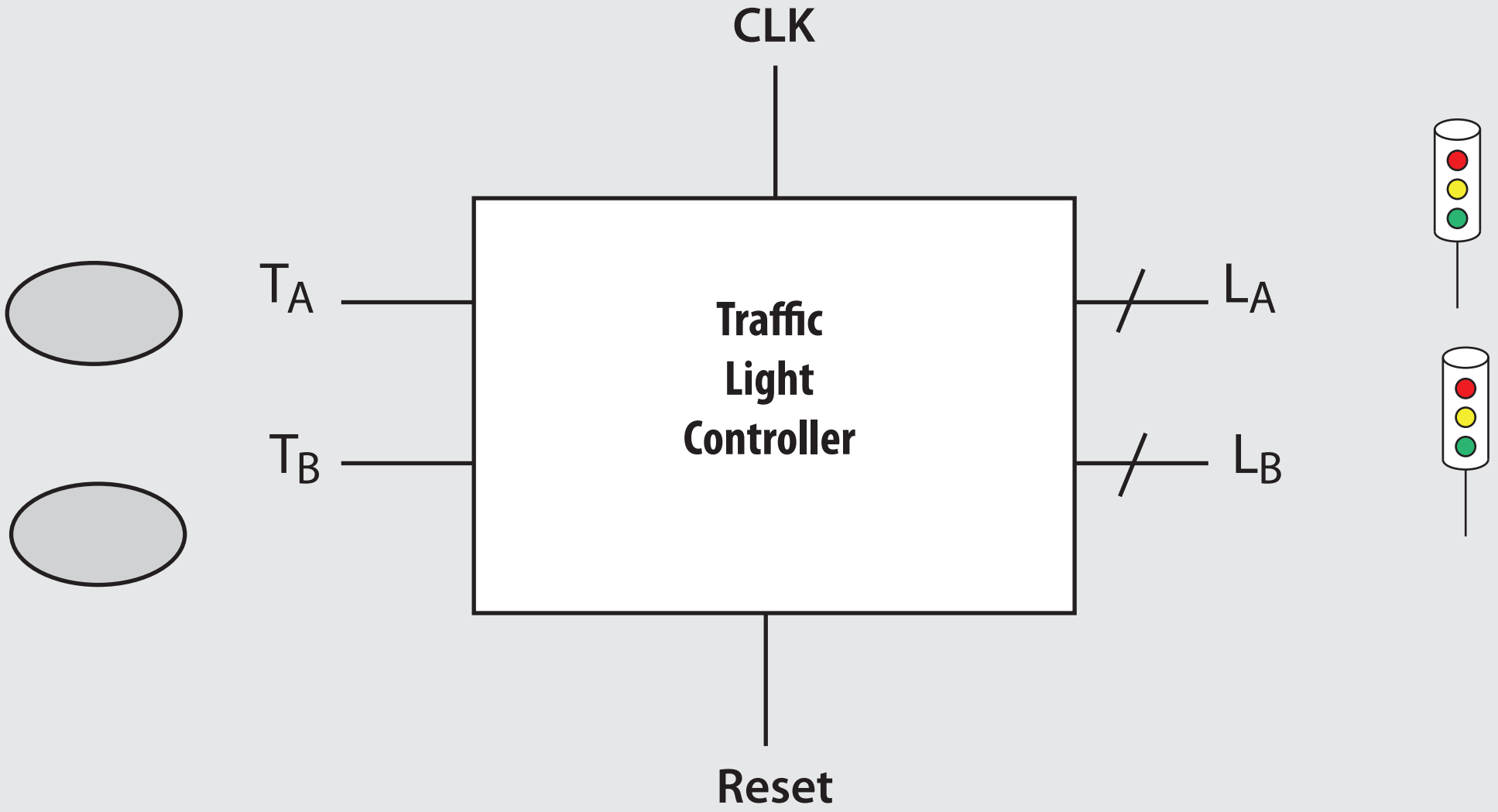


FSM Design Example

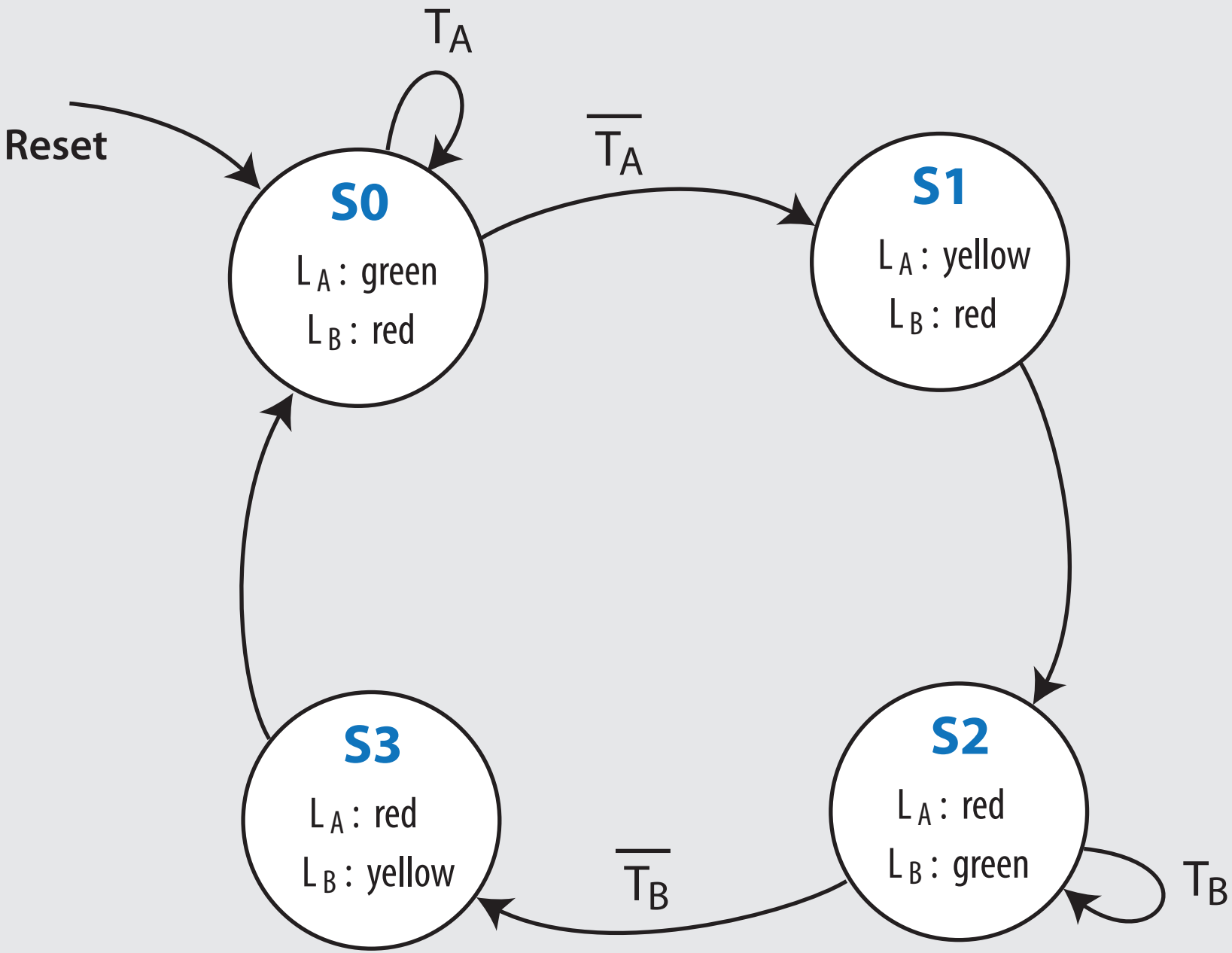


FSM Design Example

A clock with a 5-second period

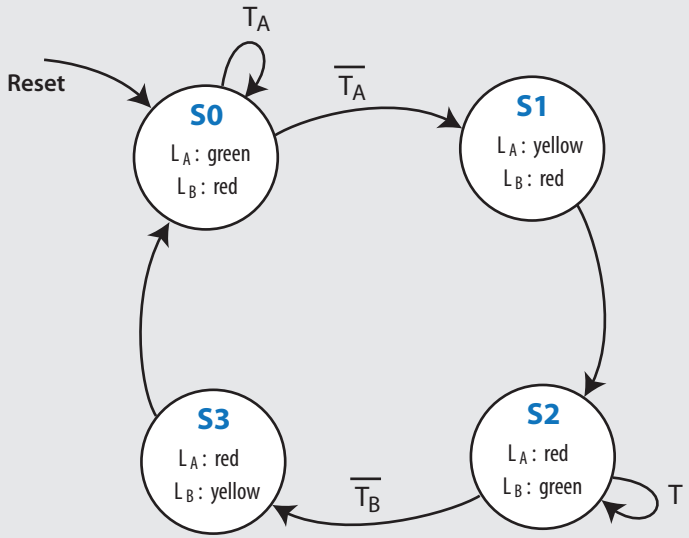


FSM Design Example



State transition table

Current State S	Inputs T_A	Inputs T_B	Next State 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



State encoding

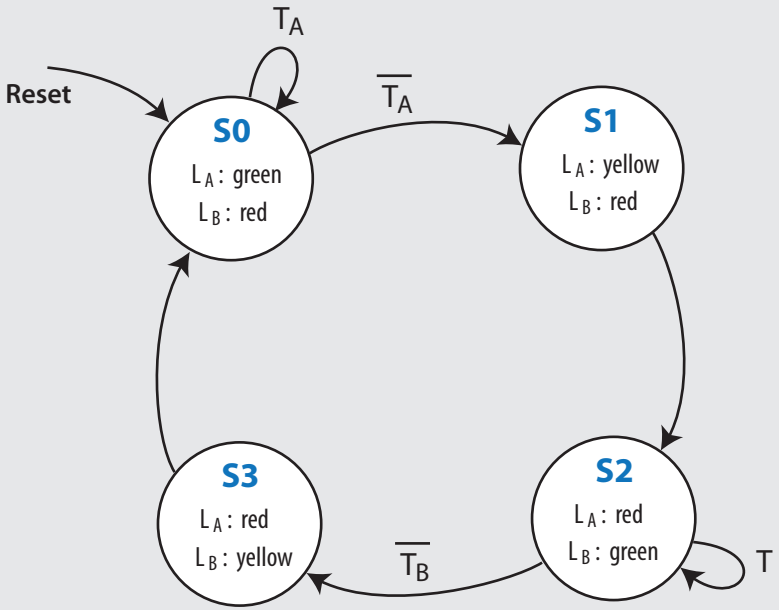
State	Encoding $S_{1:0}$
S0	00
S1	01
S2	10
S3	11

Output encoding

State	Encoding $L_{1:0}$
green	00
yellow	01
red	10

State transition table with binary encoding

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 transition table with binary encoding

Current State S_1	Current State S_0	Inputs T_A	Inputs T_B	Next State S'_1	Next State 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

$$S'_1 = \overline{S_1}\, S_0 + S_1 \overline{S_0}\, \overline{T_B} + S_1 \overline{S_0}\, T_B$$

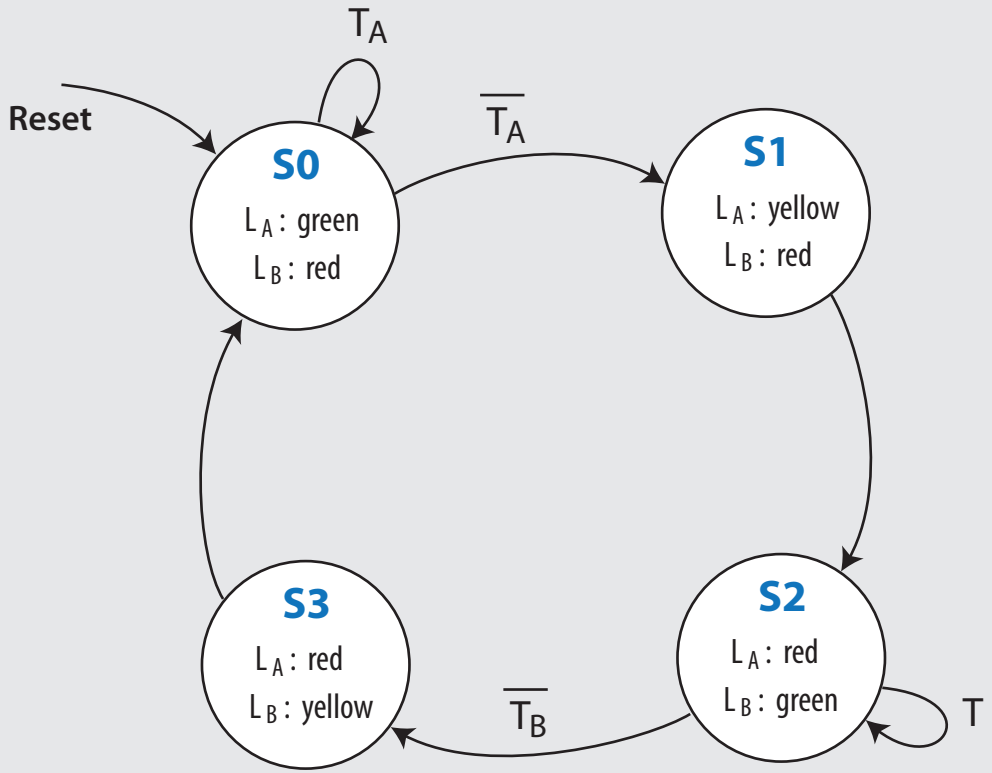
$$S'_0 = \overline{S_1}\, \overline{S_0}\, \overline{T_A} + S_1 \overline{S_0}\, \overline{T_B}$$

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

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

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



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

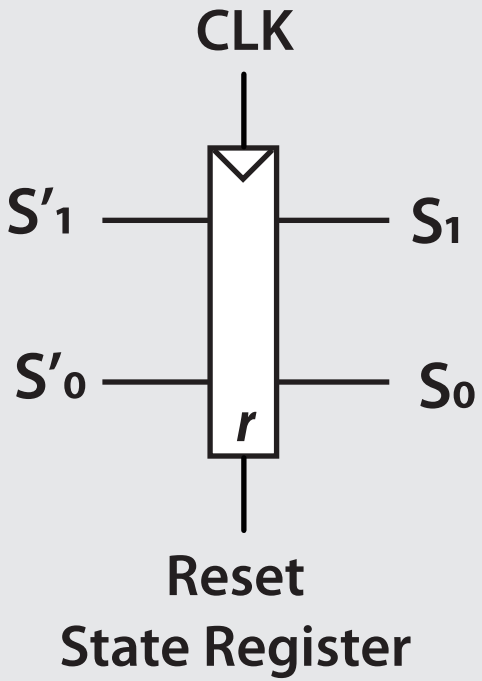
$$L_{A1} = S_1$$

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

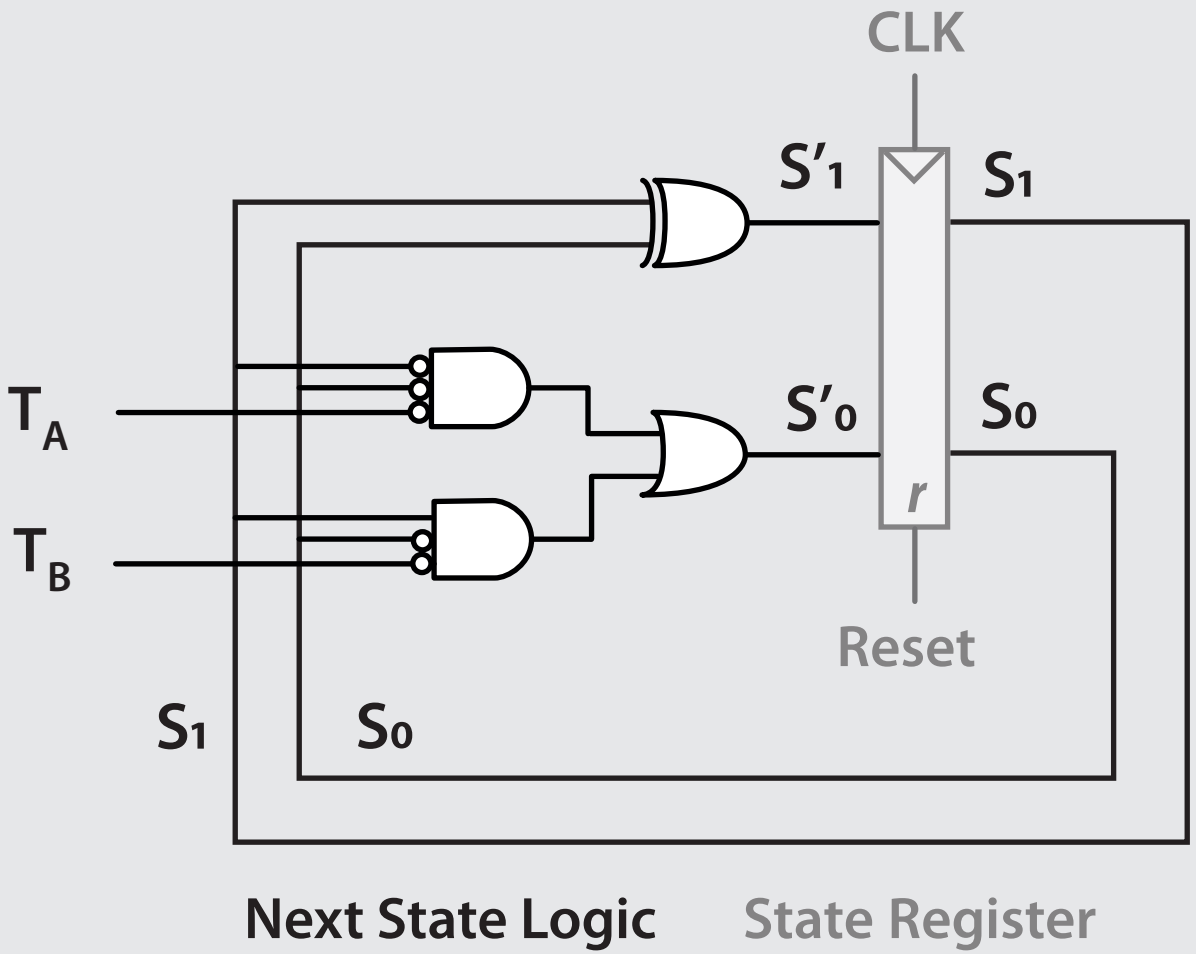
$$L_{B1} = \overline{S_1}$$

$$L_{B0} = S_1\, S_0$$

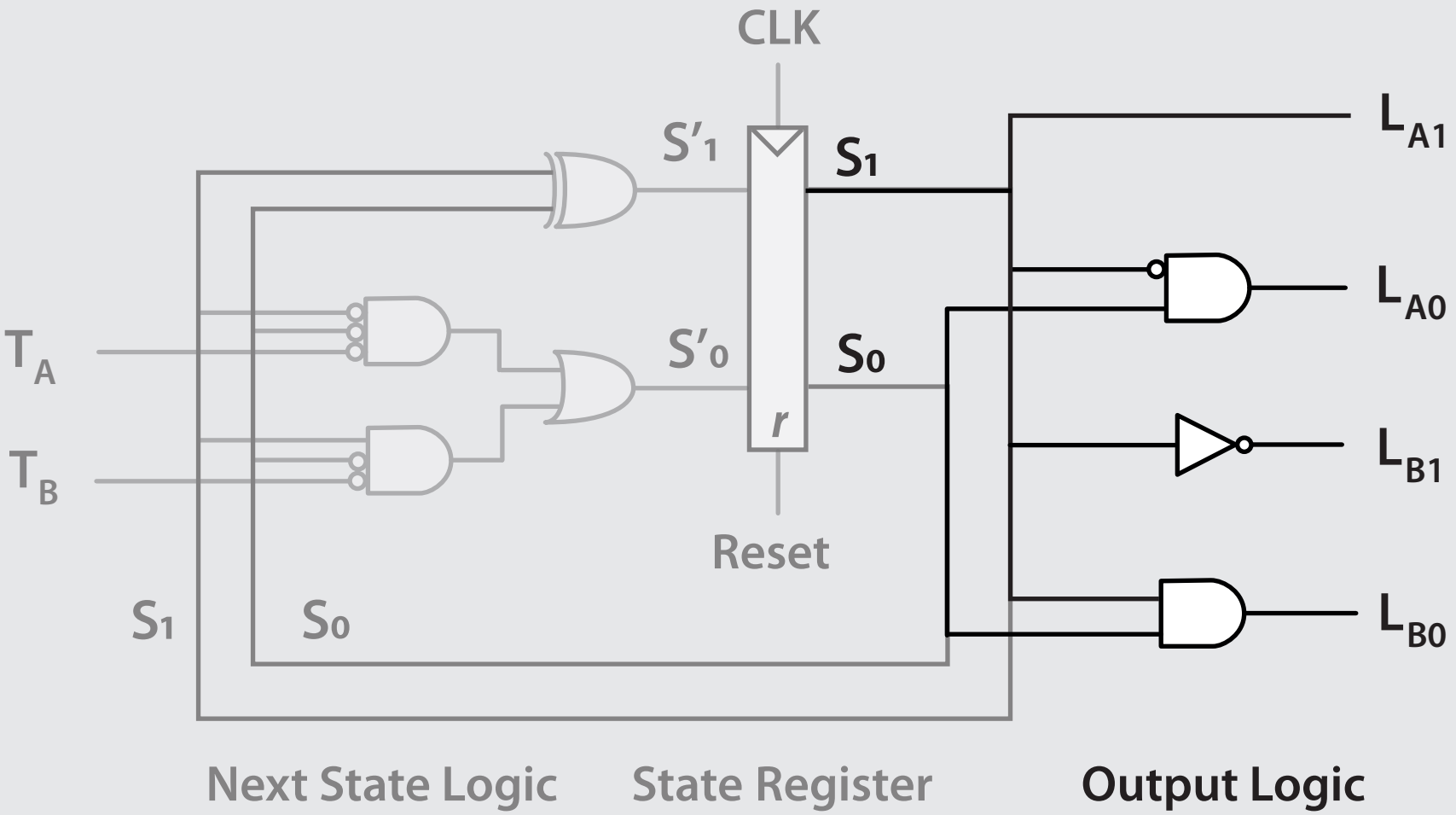
State machine circuit for traffic light controller



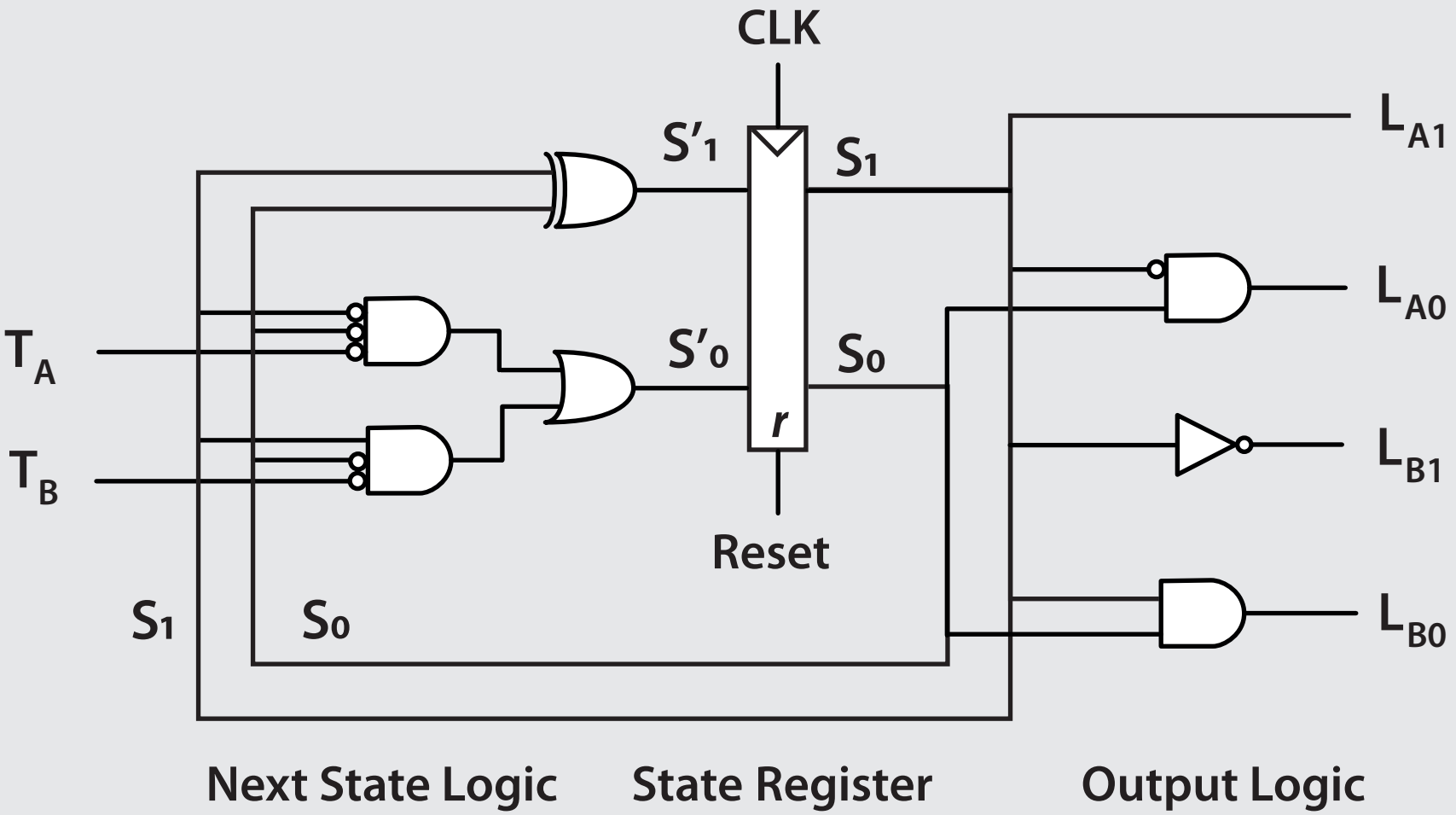
State machine circuit for traffic light controller



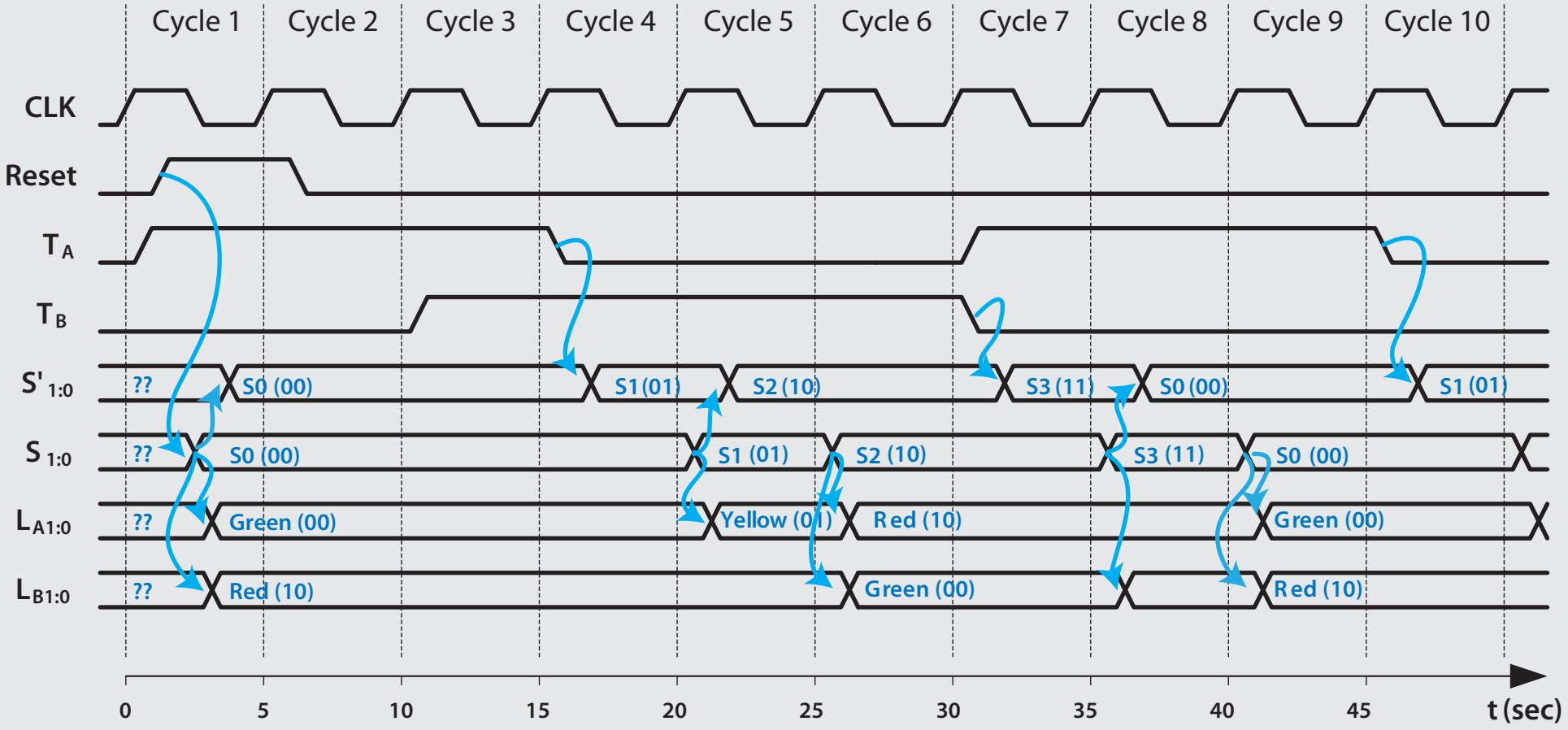
State machine circuit for traffic light controller



State machine circuit for traffic light controller

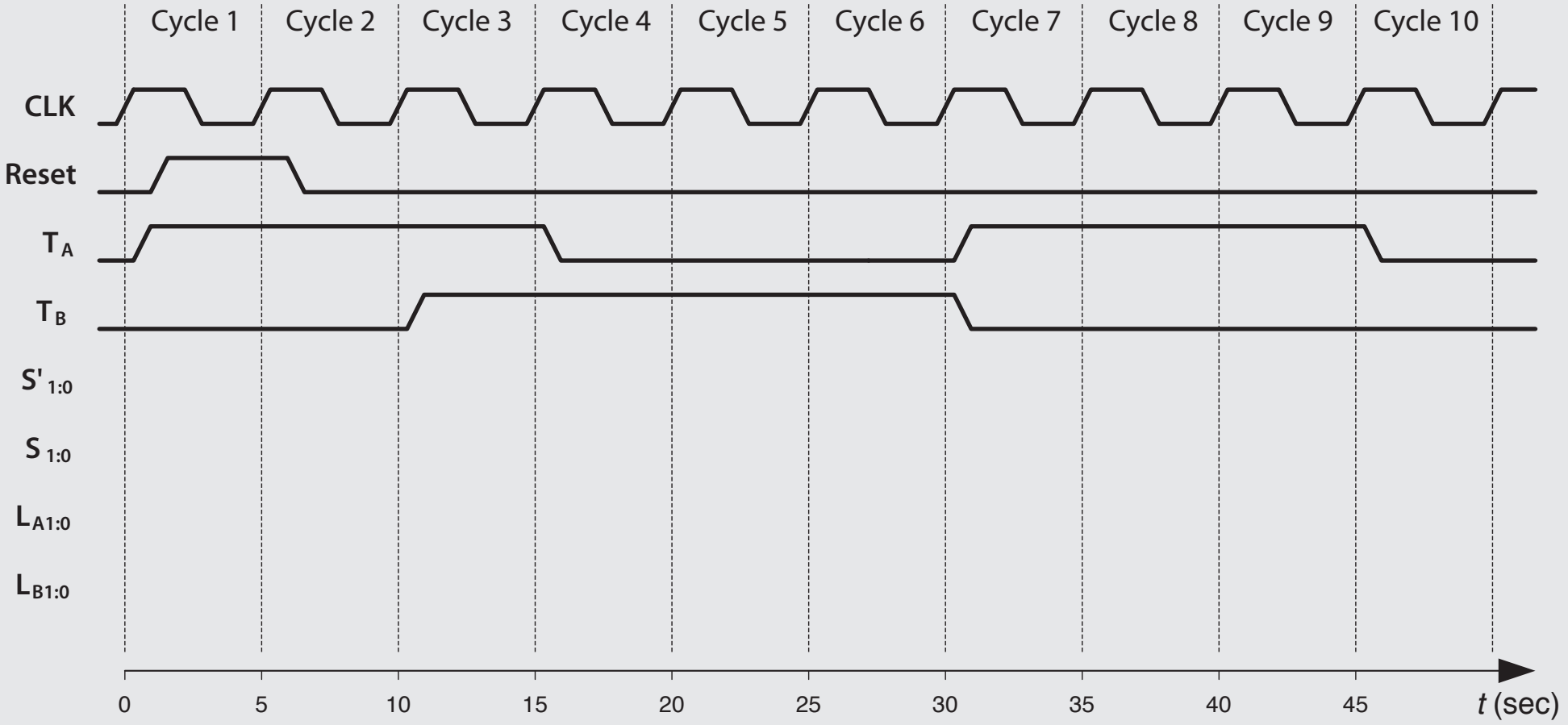


Timing Diagram for Traffic Light Controller



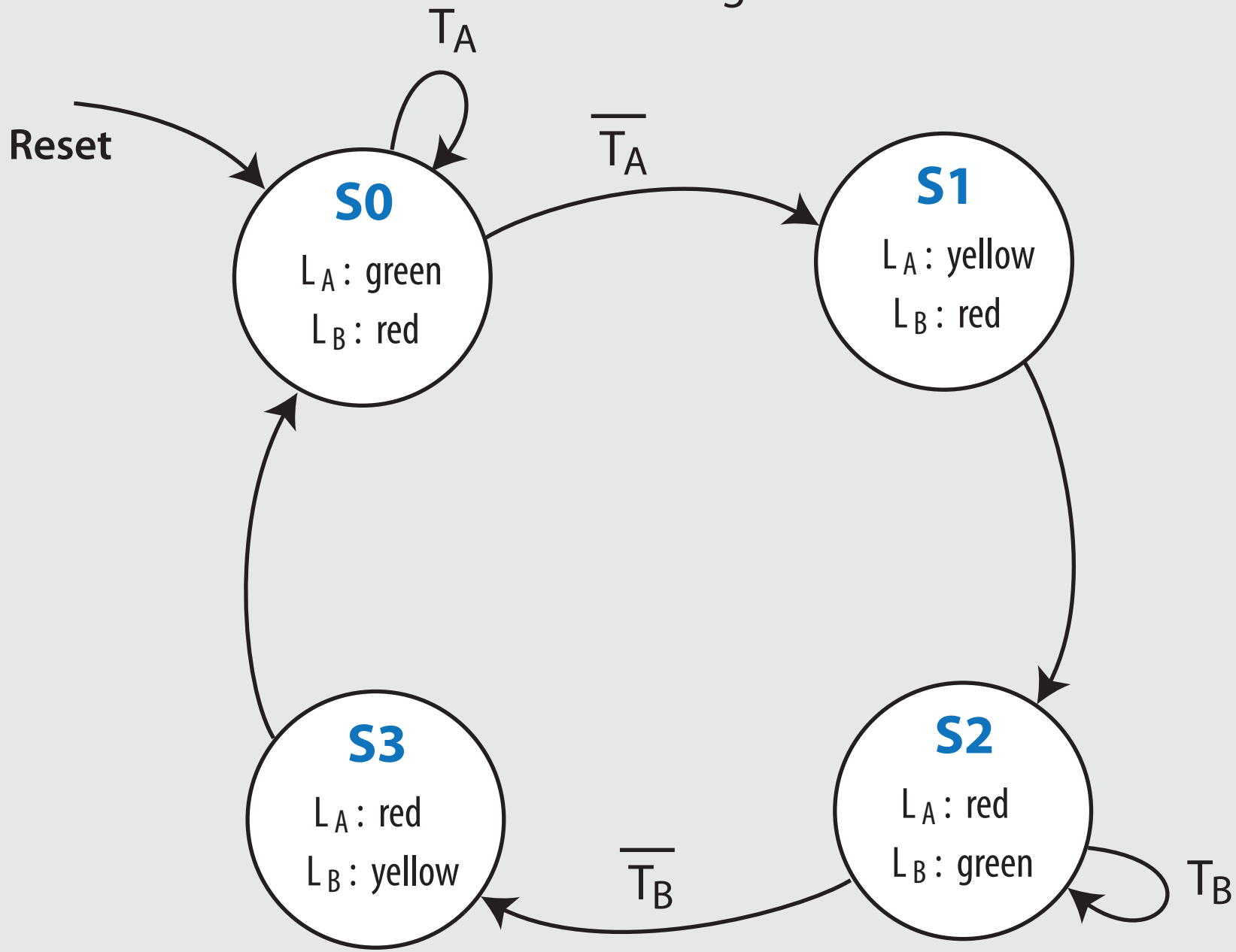
Exercise 1

Can you recover the timing diagram which we have just discussed using the finite state machine ?



Exercise 1

Can you recover the timing diagram which we have just discussed using the finite state machine ?

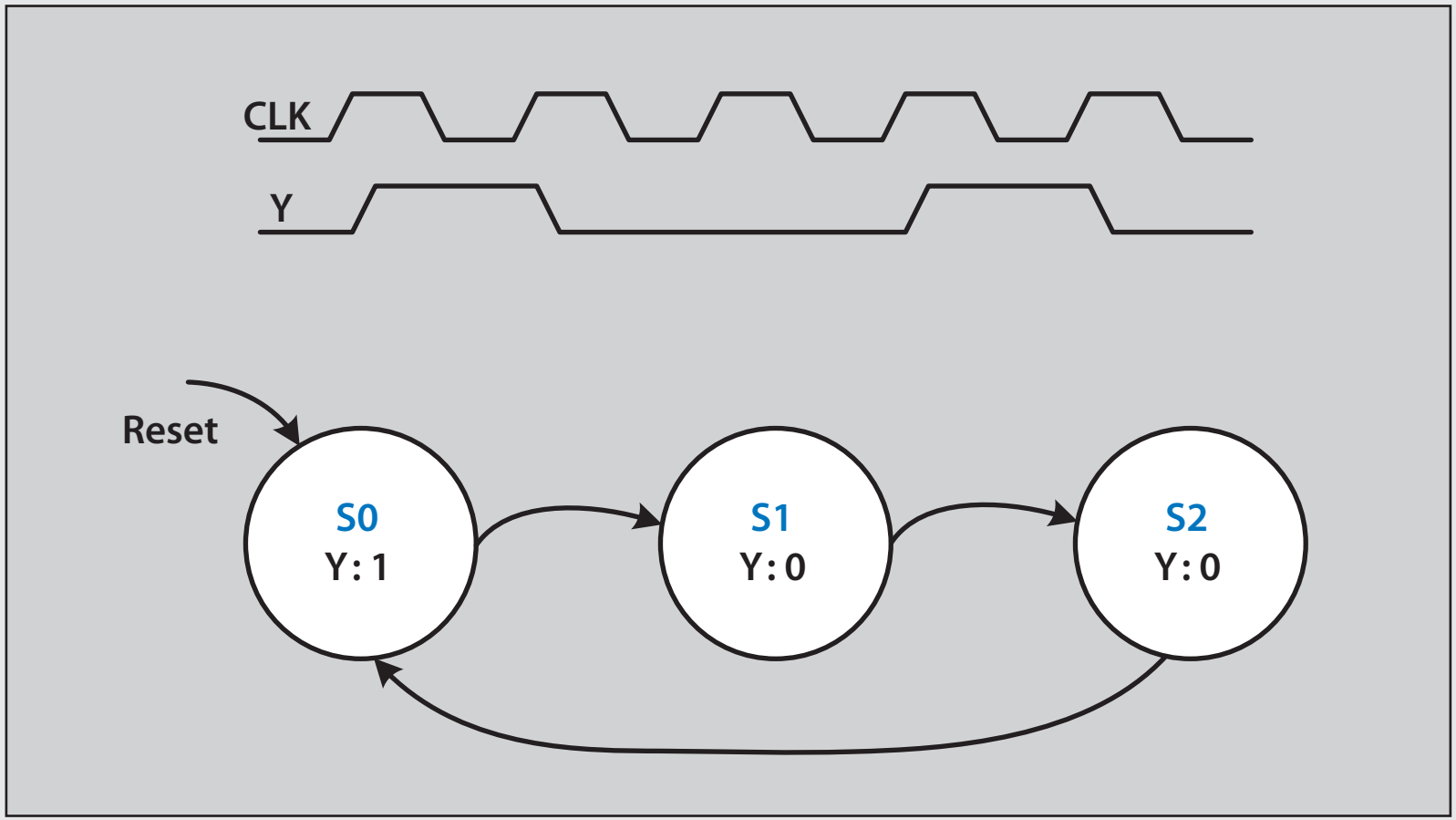


State encodings

State encodings

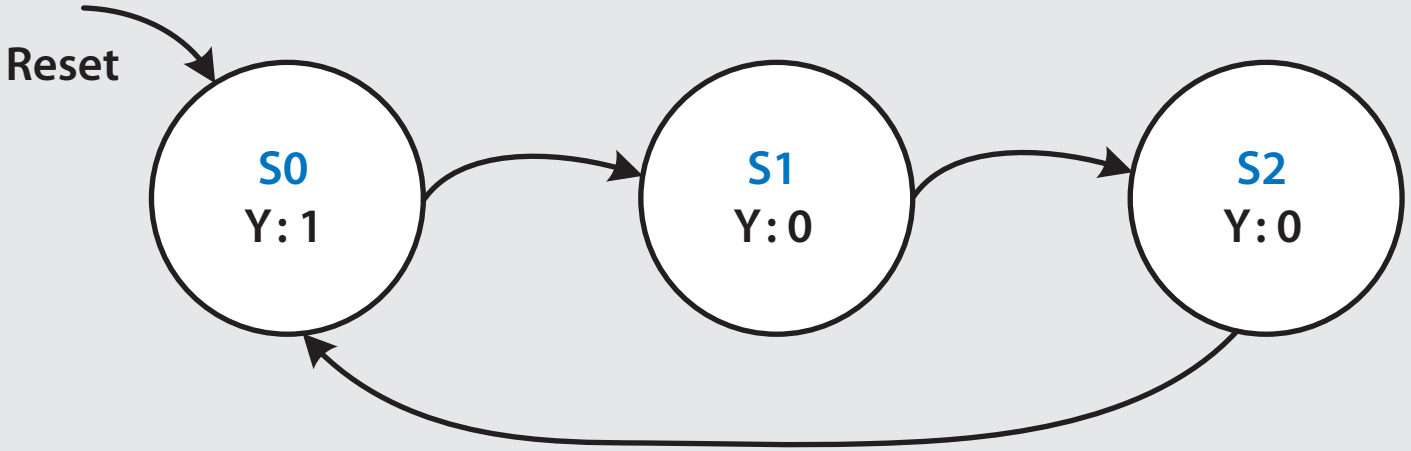
A divide-by-N counter has one output and no inputs.
The output **Y** is HIGH for one clock cycle out of every N.
In other words, the output divides the frequency of the clock by N.

Sketch circuit designs for a divide-by-3 counter using « binary state » and « one-hot state » encodings. The waveform and state transition diagram for a divide-by-3 counter is shown below :



State transition table

Current State S	Next State S′
S0	S1
S1	S2
S2	S0



State encoding

State	One-hot encoding $S_{0:2}$	Binary encoding $S_{0:1}$
S0	001	00
S1	010	01
S2	100	10

Output encoding

No need for any encoding

State transition table with binary encoding

Current State		Next State		Output
S_1	S_0	S'_1	S'_0	Y
0	0	0	1	1
0	1	1	0	0
1	0	0	0	0

$$S'_0 = \overline{S_1}\, S_0$$

$$S'_1 = \overline{S_1}\, \overline{S_0}$$

$$Y = \overline{S_1}\, \overline{S_0}$$

State transition table with one-hot encoding

Current State			Next State			Output
S_2	S_1	S_0	S'_2	S'_1	S'_0	Y
0	0	1	0	1	0	1
0	1	0	1	0	0	0
1	0	0	0	0	1	0

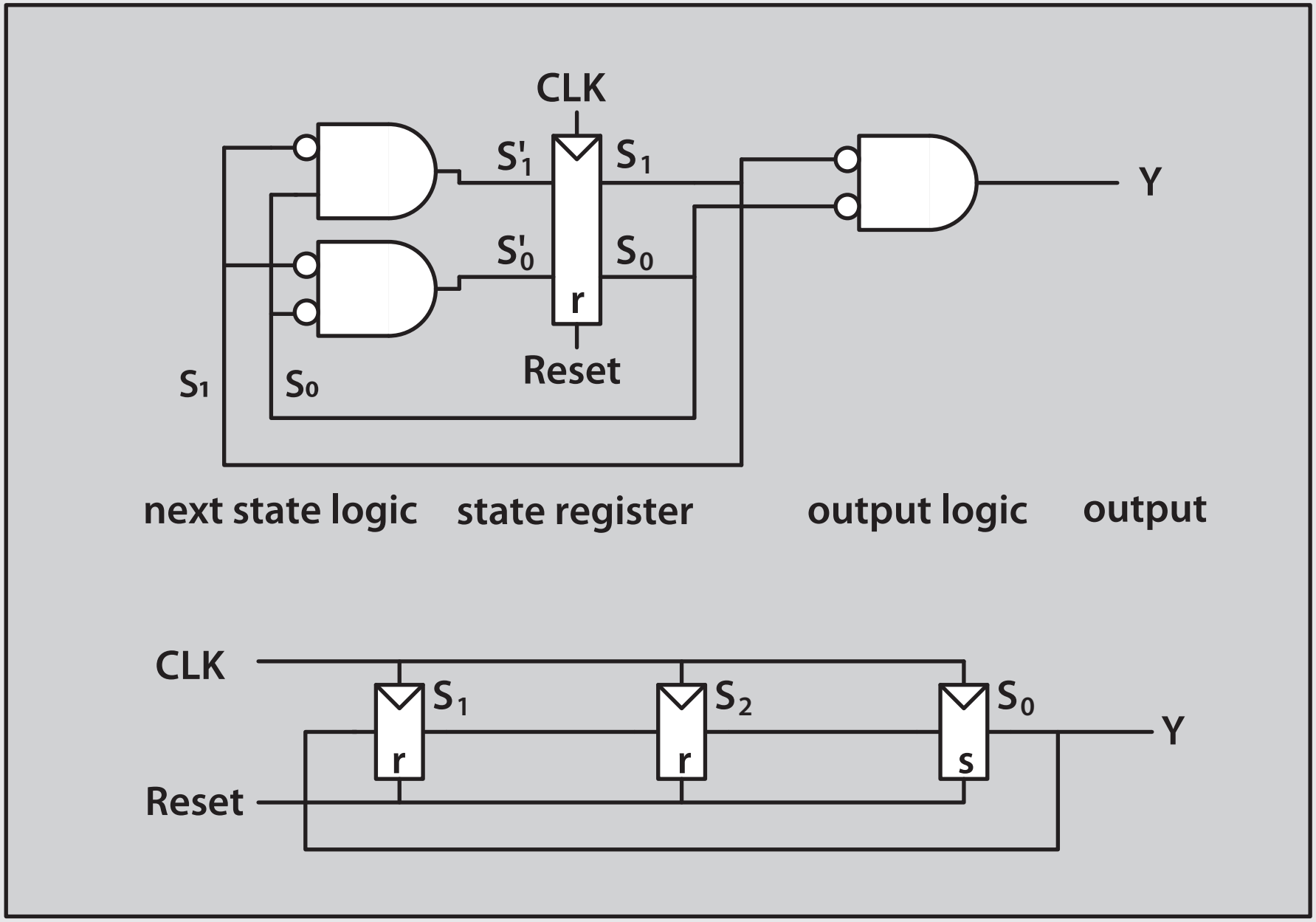
$$S'_0 = S_2$$

$$S'_1 = S_0$$

$$S'_2 = S_1$$

$$Y = S_0$$

Divide-by-3 circuits
binary and one-hot encodings



Moore vs. Mealy machines

Moore vs. Mealy machines

Alyssa P. Hacker owns a pet robotic snail with an FSM brain.

The snail crawls from left to right along a paper tape containing a sequence of 1's and 0's. On each clock cycle, the snail crawls to the next bit. The snail smiles when the last two bits that it has crawled over are, from left to right, 01.

Design the FSM to compute when the snail should smile.

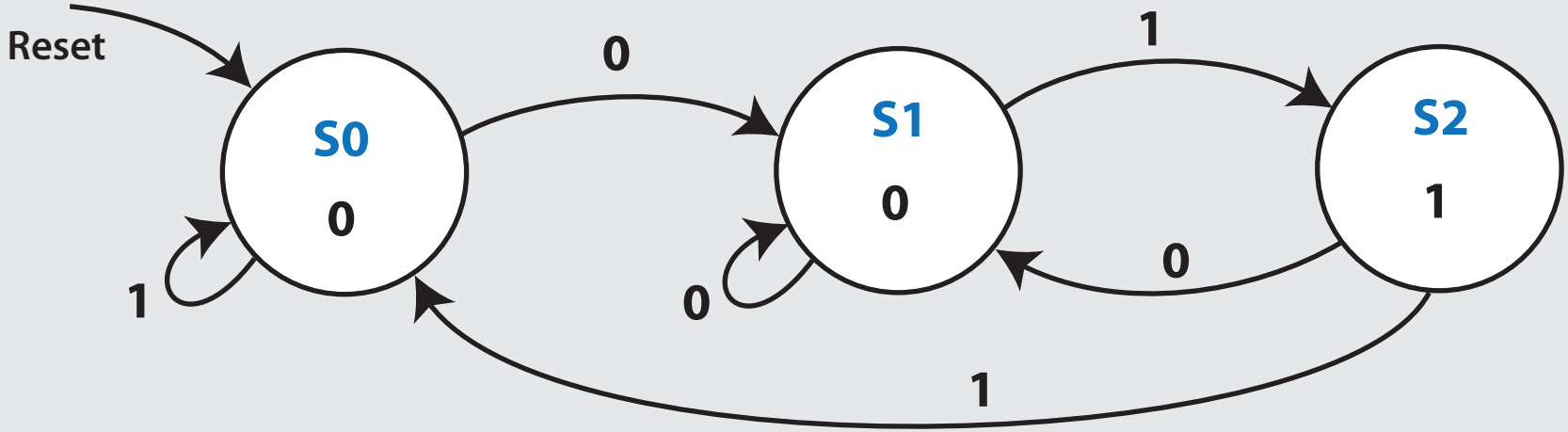
The input A is the bit underneath the snail's antennae. The output Y is TRUE when the snail smiles.

Compare Moore and Mealy state machine designs.

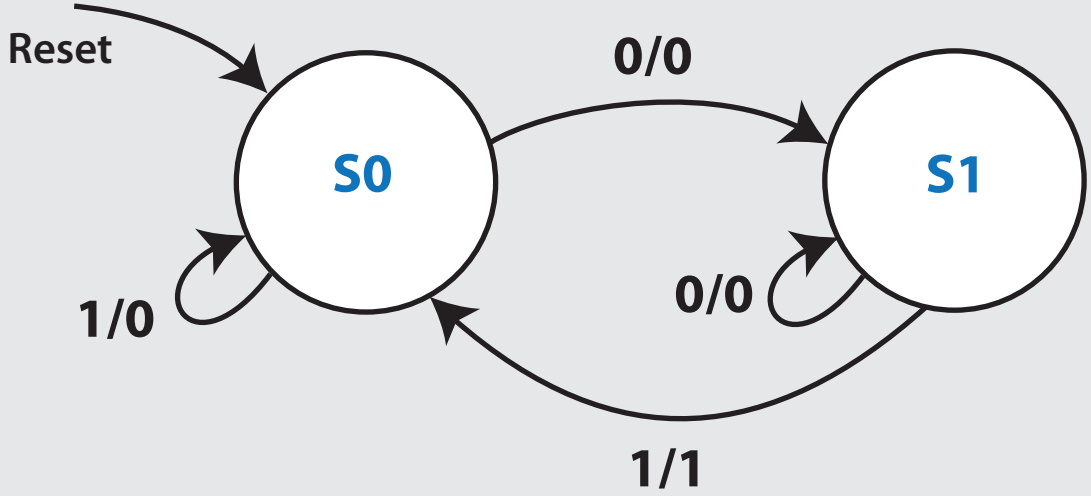
Sketch a timing diagram for each machine showing the input, states, and output as Alyssa's snail crawls along the sequence 0100110111.

H & H Section 3.4.3

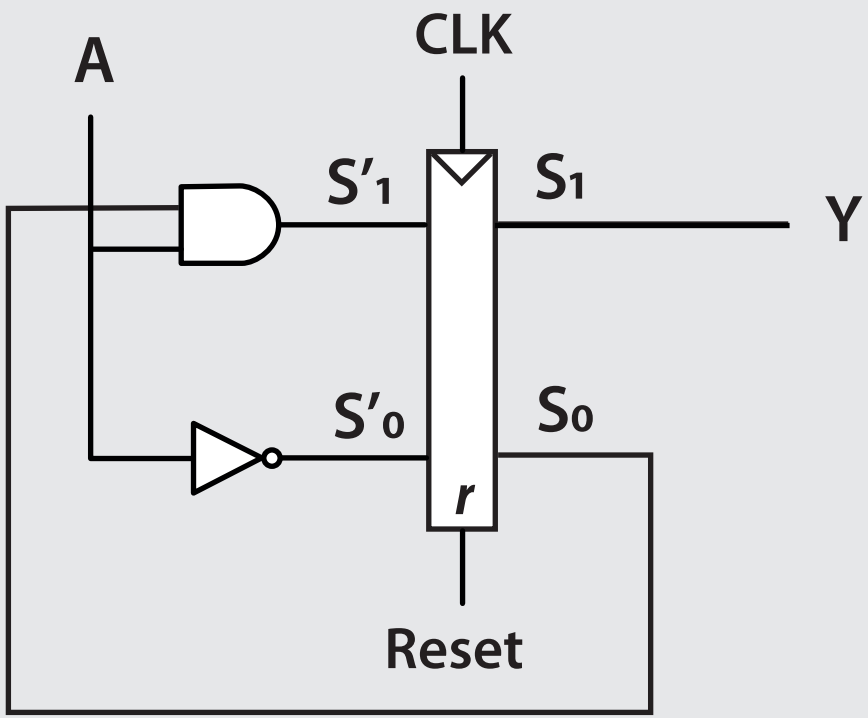
Moore Machine



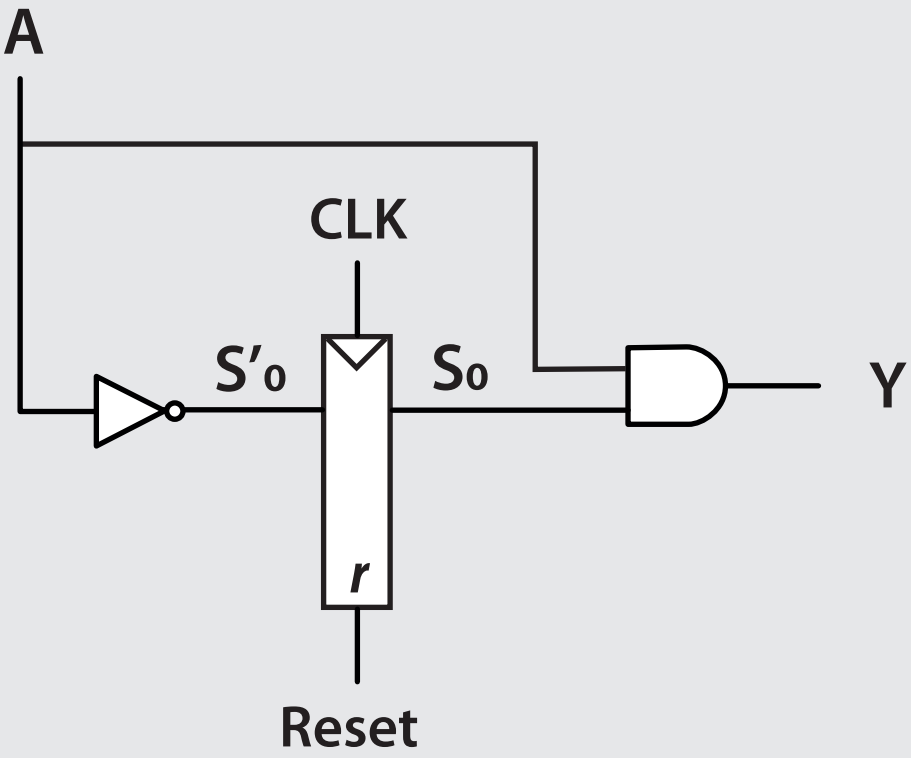
Mealy Machine



Moore vs. Mealy machines

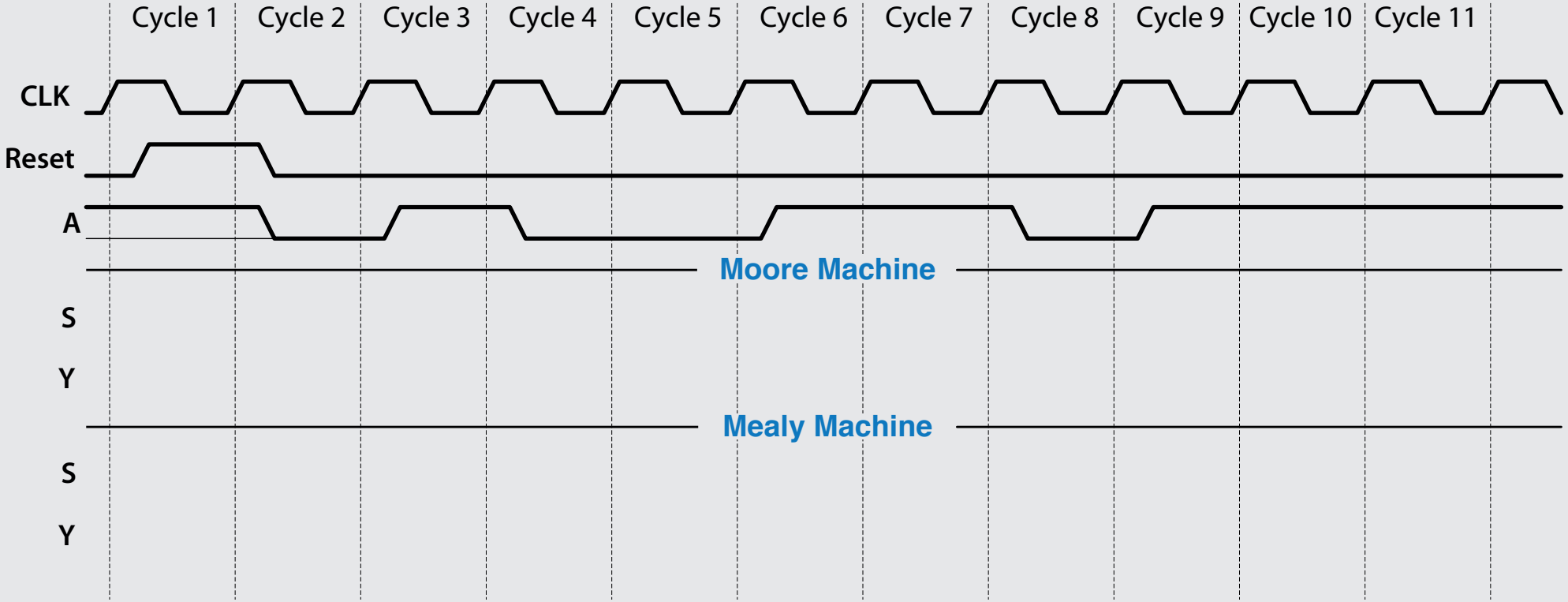


Moore machine

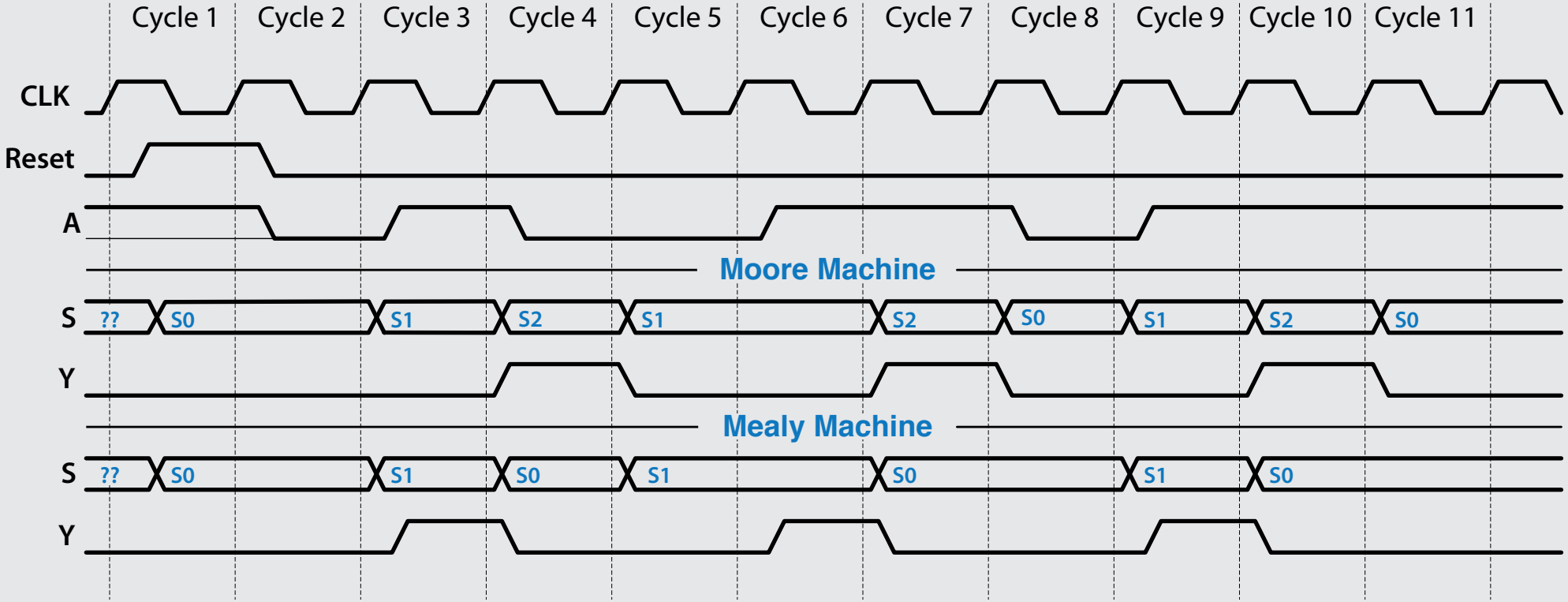


Mealy machine

Exercise 2


Cycle 1
Cycle 2
Cycle 3
Cycle 4
Cycle 5
Cycle 6
Cycle 7
Cycle 8
Cycle 9
Cycle 10
Cycle 11
CLK
Reset
A
Moore Machine
S
Y
Mealy Machine
S
Y

Solution


Cycle 1
Cycle 2
Cycle 3
Cycle 4
Cycle 5
Cycle 6
Cycle 7
Cycle 8
Cycle 9
Cycle 10
Cycle 11
CLK
Reset
A
Moore Machine
S
?? S0 S1 S2 S1 S2 S0 S1 S2 S0
Y
Mealy Machine
S
?? S0 S1 S0 S1 S0 S1 S0
Y

Factored and Unfactored State Machines

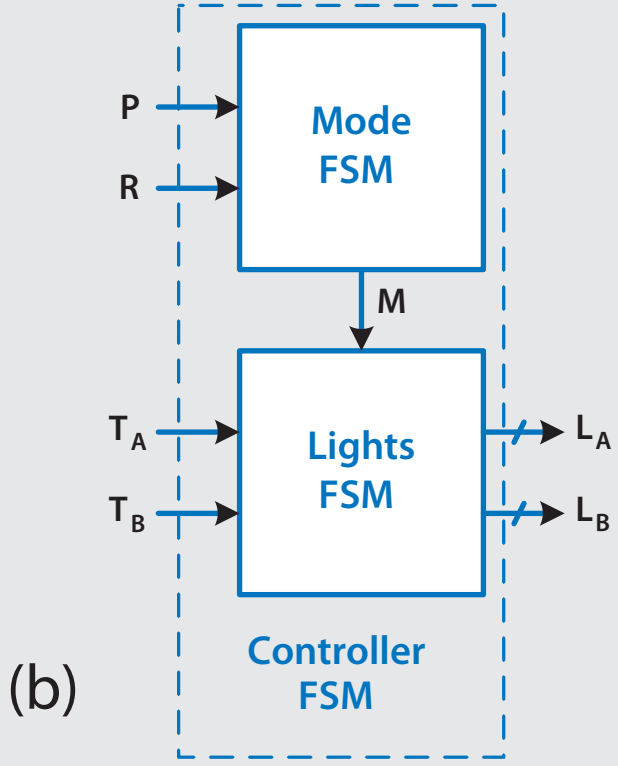
Exercise 3

Modify the traffic light controller to have a parade mode, which keeps the Bravado Boulevard light green while spectators and the band march to football games in scattered groups. The controller receives two more inputs: **P** and **R**. Asserting **P** for at least one cycle enters parade mode. Asserting **R** for at least one cycle leaves parade mode. When in parade mode, the controller proceeds through its usual sequence until $\mathbf{L_B}$ turns green, then remains in that state with $\mathbf{L_B}$ green until parade mode ends.

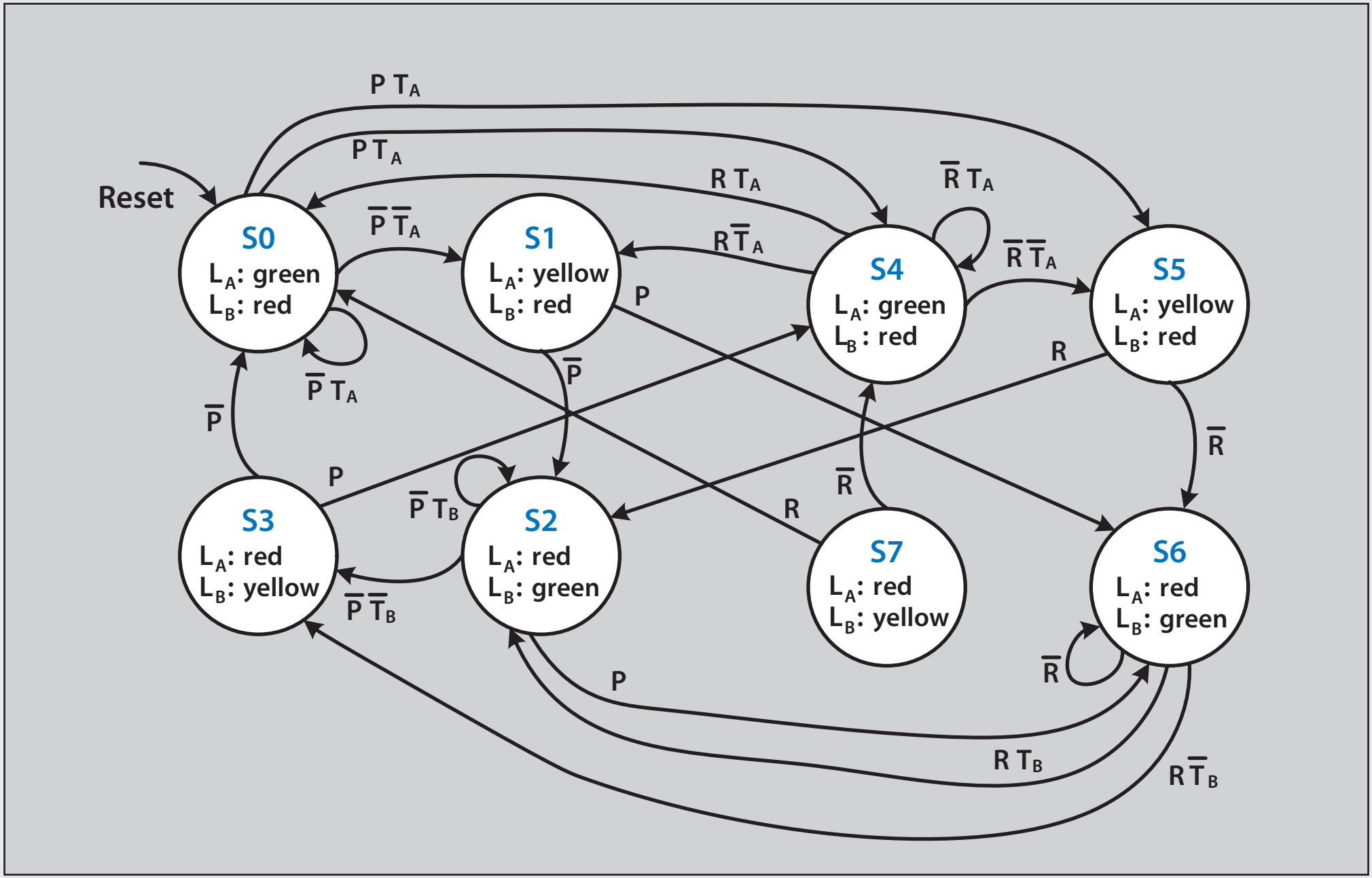
First sketch a transition system for a single FSM as shown in (a) then factor it as indicated in (b) where the signal M indicates that the machine is in mode « parade ».

(a)

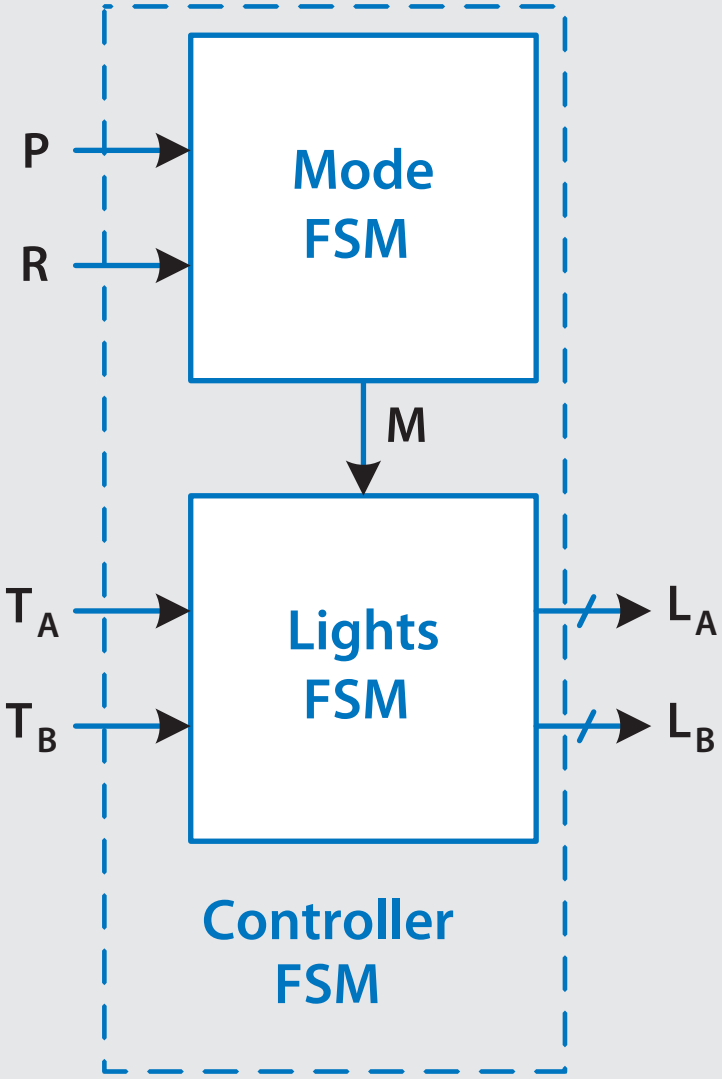
(b)



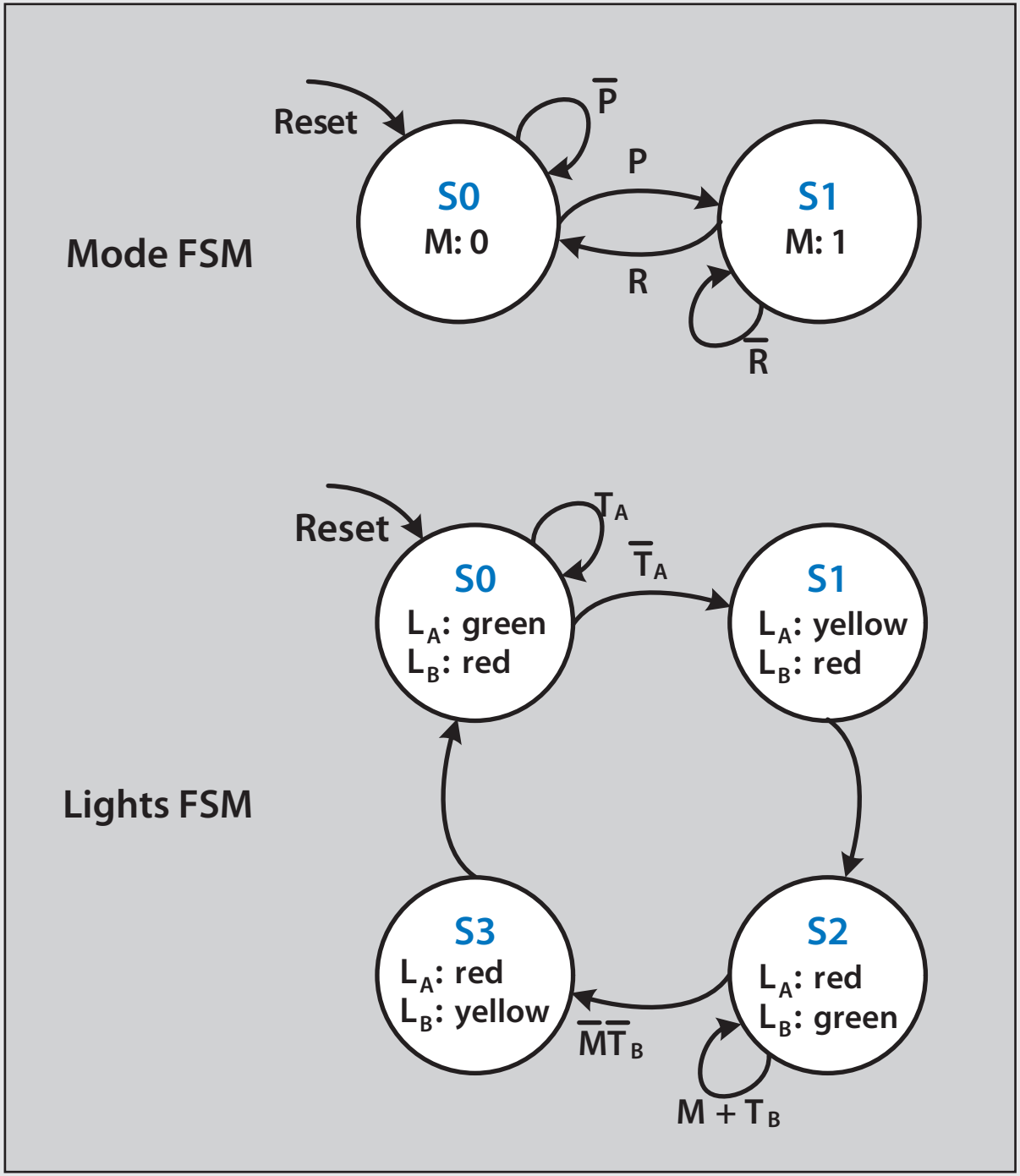
Solution Transition System for a Single FSM



Solution



The factored version



Finite State Machine Review

- Identify the inputs and outputs.
- Sketch a state transition diagram.
- For a ***Moore machine*** :
  - Write a state transition table.
  - Write an output table.
- For a ***Mealy machine*** :
  - Write a combined state transition and output table.
- Select state encodings : your selection affects the hardware design.
- Write Boolean equations for the next state and output logic.
- Sketch the circuit schematic.

Exercise 4

Exercise 3.9 in H & H

The toggle (T) flip-flop has one input, CLK, and one output, Q.
On each rising edge of CLK, Q toggles to the complement
of its previous value.

Draw a schematic for a T flip-flop using a D flip-flop and an inverter.

Exercise 5

Exercise 3.24 in H & H

You have been enlisted to design a soda machine dispenser for your department lounge. Sodas are partially subsidized by the student chapter of the IEEE, so they cost only 25 cents. The machine accepts nickels, dimes, and quarters. When enough coins have been inserted, it dispenses the soda and returns any necessary change.

Design an FSM controller for the soda machine. The FSM inputs are Nickel, Dime, and Quarter, indicating which coin was inserted. Assume that exactly one coin is inserted on each cycle. The outputs are Dispense, ReturnNickel, ReturnDime, and ReturnTwoDimes.

When the FSM reaches 25 cents, it asserts Dispense and the necessary Return outputs required to deliver the appropriate change. Then it should be ready to start accepting coins for another soda.

Exercise 6

Exercise 3.27 in H & H

Gray codes have a useful property in that consecutive numbers differ in only a single bit position. The table below lists a 3-bit Gray code representing the numbers 0 to 7.

Design a 3-bit modulo 8 Gray code counter FSM with no inputs and three outputs. A modulo N counter counts from 0 to N – 1, then repeats. For example, a watch uses a modulo 60 counter for the minutes and seconds that counts from 0 to 59. When reset, the output should be 000. On each clock edge, the output should advance to the next Gray code. After reaching 100, it should repeat with 000.

Number	bit_2	bit_1	bit_0
0	0	0	0
1	0	0	1
2	0	1	1
3	0	1	0

Number	bit_2	bit_1	bit_0
4	1	1	0
5	1	1	1
6	1	0	1
7	1	0	0