The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

ECE241 – Digital Systems
Finite State Machines (review)

Fall 2023 – Bruno Korst, P.Eng.

1

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Definitions

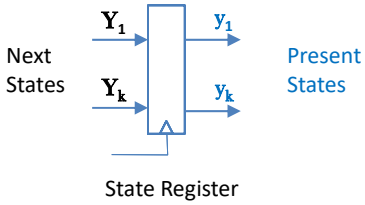
- **Synchronous Sequential Machine**
  - Machine whose present outputs are a **function of the present state**
    OR
  - Machine whose present outputs are a **function of the present inputs and the present state**.
- Synchronous → "responds to a clock"
- Sequential logic system → combinational logic + storage

2

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Definitions

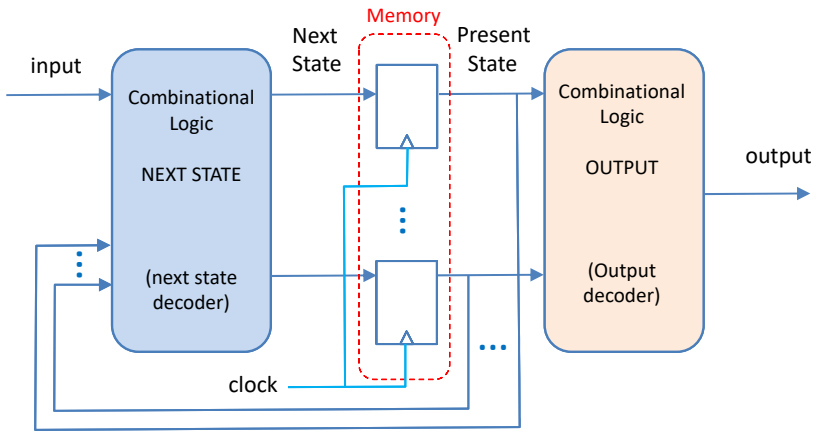
- State is "*a set of values that is measured at different locations within the machine*" (stored, in clocked D-FF, registers)
  - **Present state**: state of the system at the present
  - **Next state**: state **to which the system will enter** on the next clock edge.



3

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Definitions

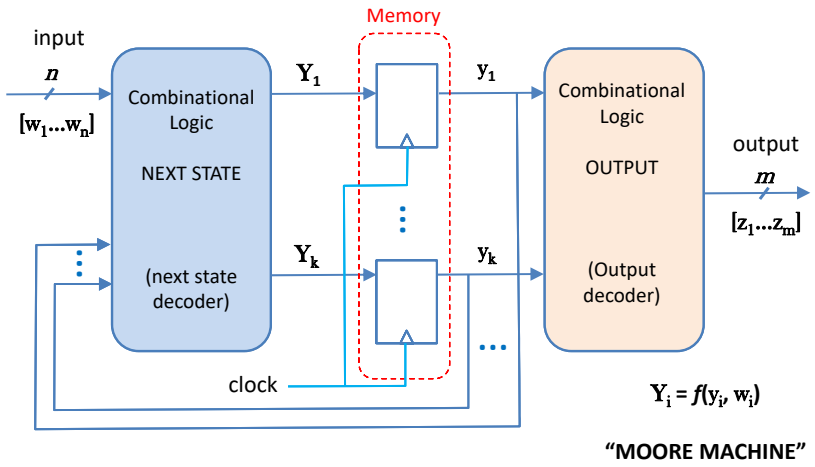


- Note: the **clock is is not an input**, it is a "heartbeat"
- Rising edge determines when state transitions occur → **"next" becomes "present"**

4

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Definitions



- k state variables (number of FF), make possible to represent 2^k states
- On clock edge, Y_i becomes y_i

5

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Traffic Light FSM

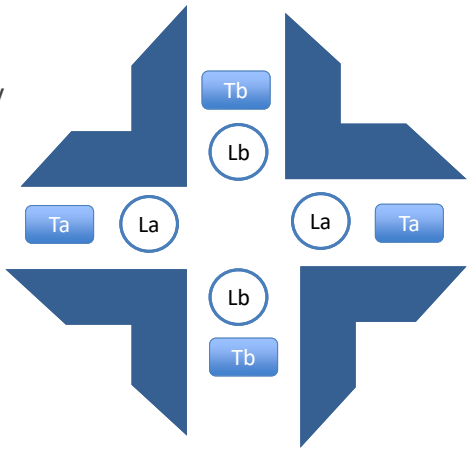
- **The problem**
  - There is an intersection of two busy roads which needs traffic lights to prevent collisions.
  - These lights are to be controlled by traffic sensors, which indicate whether there are cars present or not.

6

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Traffic Light FSM

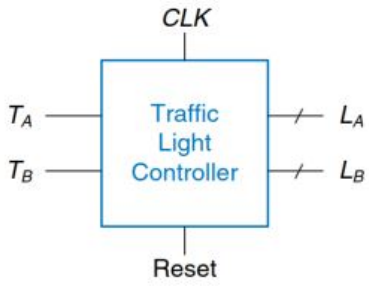
- Ta, Tb – traffic sensors
  - TRUE if cars present
  - FALSE if street empty
- La, Lb – lights
  - Green
  - Yellow
  - Red



The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Traffic Light FSM

- Two inputs – Ta, Tb
- Two outputs – La, Lb
- Clock
  - Say, at 5 sec period
  - @ Positive edge
    - Lights Change



- Reset brings it to a known initial state (asynchronously)

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering UNIVERSITY OF TORONTO

Traffic Light FSM

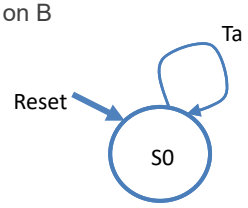
- **A systematic approach to the design**
  1. Draw the state transition diagram
  2. Create the state transition table
  3. Encode the states on the state transition table
  4. Derive the logic expressions, minimize them
  5. Create the output table, encode it
  6. Derive the logic expressions, minimize them
  7. Draw the circuit, implement it
     - On “old-school” hardware, in code, on HDL

9

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering UNIVERSITY OF TORONTO

Traffic Light FSM

- State Transition Diagram
  - States are represented by circles, transitions by arcs
  - Reset – green on “A” street, red on “B” street
  - Every 5 seconds (clock), examine sensors (Ta, Tb) and decide:
    - Is traffic present on A – if so, lights do not change
    - When no more traffic on A for 5 seconds – light turns yellow on A
    - 5 seconds later, red on A, green on B

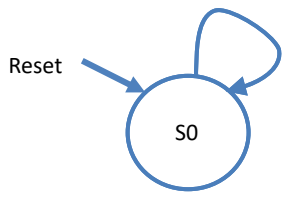


10

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Traffic Light FSM

Reset does not depend on the initial transition

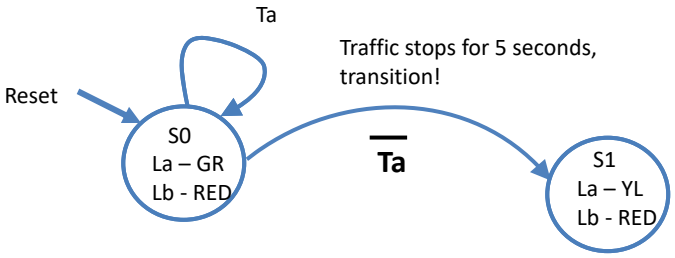


While there IS traffic, there is no change. (Ta is TRUE)

11

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

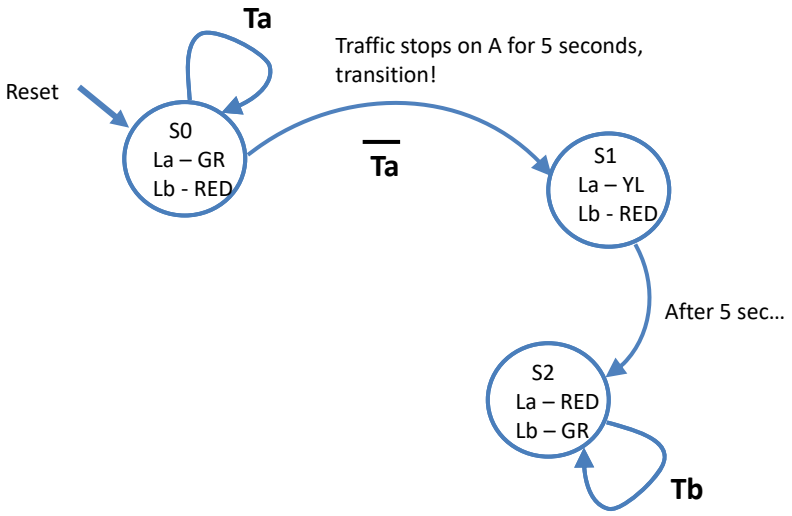
Traffic Light FSM



12

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

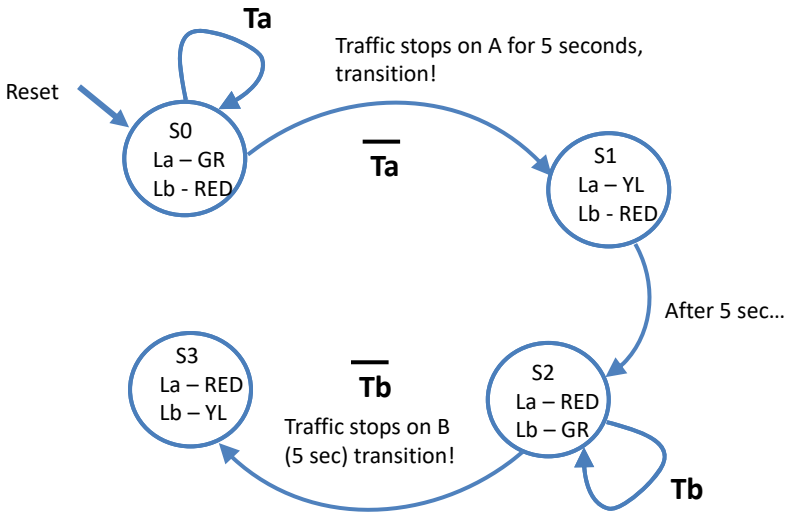
Traffic Light FSM



13

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

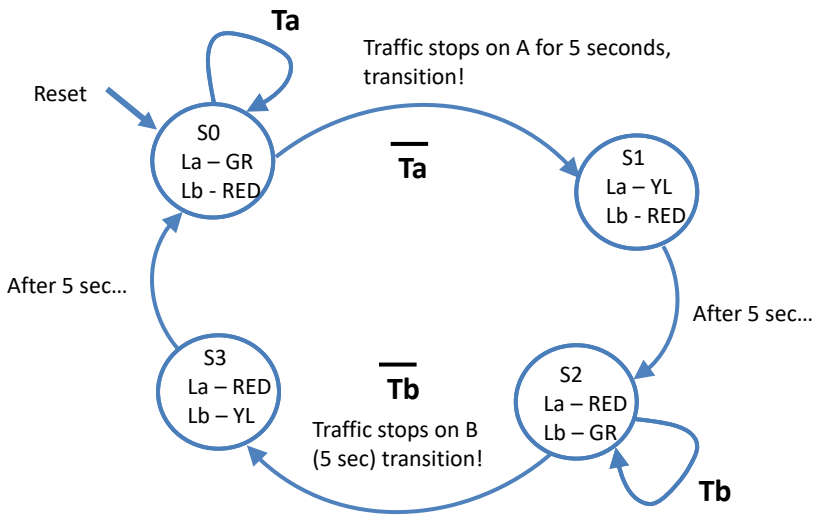
Traffic Light FSM



14

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Traffic Light FSM



15

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Traffic Light FSM

- **A systematic approach to the design**
  1. Draw the state transition diagram
  2. Create the state transition table
  3. Encode the states on the state transition table
  4. Derive the logic expressions, minimize them
  5. Create the output table, encode it
  6. Derive the logic expressions, minimize them
  7. Draw the circuit, implement it
     - On “old-school” hardware, in code, on HDL

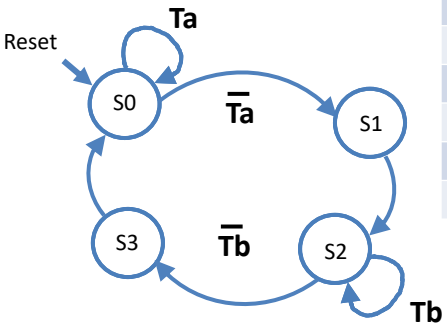
16

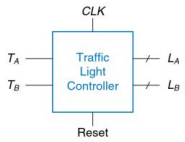
The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Traffic Light FSM

- State Transition Table

Current	Ta	Tb	Next
S0	0	X	S1
S0	1	X	S0
S1	X	X	S2
S2	X	0	S3
S2	X	1	S2
S3	X	X	S0





(review the examples given before)

17

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Traffic Light FSM

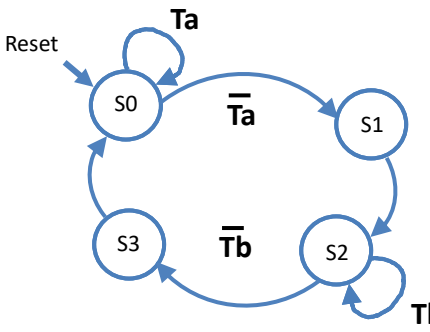
- **A systematic approach to the design**
  1. Draw the state transition diagram
  2. Create the state transition table
  3. Encode the states on the state transition table
  4. Derive the logic expressions, minimize them
  5. Create the output table, encode it
  6. Derive the logic expressions, minimize them
  7. Draw the circuit, implement it
     - On “old-school” hardware, in code, on HDL

18

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Traffic Light FSM

- State Encoding



Current	Ta	Tb	Next
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

Sb1 Sb0 S'b1 S'b0

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Traffic Light FSM

- **A systematic approach to the design**
  1. Draw the state transition diagram
  2. Create the state transition table
  3. Encode the states on the state transition table
  4. Derive the logic expressions, minimize them
  5. Create the output table, encode it
  6. Derive the logic expressions, minimize them
  7. Draw the circuit, implement it
     - On “old-school” hardware, in code, on HDL

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Traffic Light FSM

- Logic Expressions

Current	Ta	Tb	Next
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

Sb1 Sb0 S'b1 S'b0

$$S'b_1 = \overline{Sb_1}Sb_0 + Sb_1\overline{Sb_0}\,\overline{T_b} + Sb_1\overline{Sb_0}T_b$$
$$S'b_0 = \overline{Sb_1}\,\overline{Sb_0}\,\overline{T_a} + Sb_1\overline{Sb_0}\,\overline{T_b}$$

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Traffic Light FSM

- Logic Expressions

Current	Ta	Tb	Next
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

Sb1 Sb0 S'b1 S'b0

$$S'b_1 = \overline{Sb_1}Sb_0 + Sb_1\overline{Sb_0}\,\overline{T_b} + Sb_1\overline{Sb_0}T_b$$
$$S'b_0 = \overline{Sb_1}\,\overline{Sb_0}\,\overline{T_a} + Sb_1\overline{Sb_0}\,\overline{T_b}$$

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Traffic Light FSM

- **State Transition Table** – next states from current states

Current	Ta	Tb	Next
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

Sb1 Sb0 S'b1 S'b0

$$S'b_1 = \overline{Sb_1}Sb_0 + Sb_1\overline{Sb_0}$$

$$S'b_0 = \overline{Sb_1}\,\overline{Sb_0}\,\overline{T_a} + Sb_1\overline{Sb_0}\,\overline{T_b}$$

$$S'b_1 = Sb_1 \oplus Sb_0$$

$$S'b_0 = \overline{Sb_1}\,\overline{Sb_0}\,\overline{T_a} + Sb_1\overline{Sb_0}\,\overline{T_b}$$

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

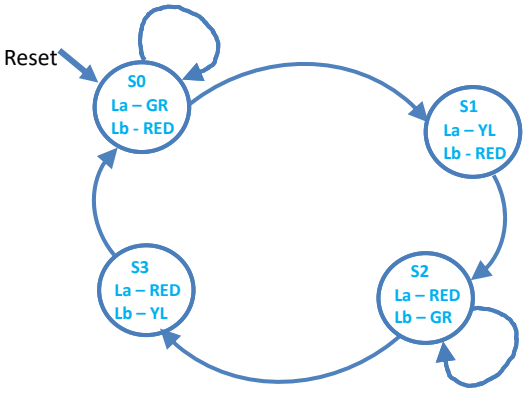
Traffic Light FSM

- **A systematic approach to the design**
  1. Draw the state transition diagram
  2. Create the state transition table
  3. Encode the states on the state transition table
  4. Derive the logic expressions, minimize them
  5. Create the output table, encode it
  6. Derive the logic expressions, minimize them
  7. Draw the circuit, implement it
     - On "old-school" hardware, in code, on HDL

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Traffic Light FSM

- **Output table** – we look at La and Lb, codified



Each output, 2 bits

Red – 0 0
Yellow – 0 1
Green – 1 0

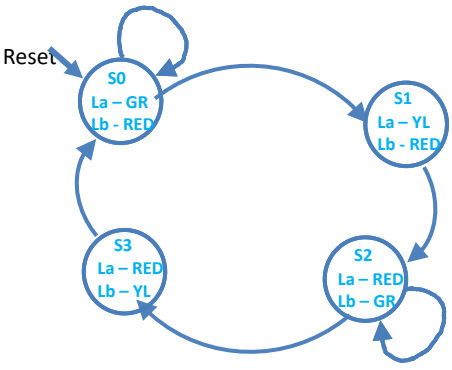
What happens to 1 1?

25

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Traffic Light FSM

- Output table – we look at La and Lb, codified



Current		La		Lb	
0	0	0	0 (G)	1	0 (R)
0	1	0	1 (Y)	1	0 (R)
1	0	1	0 (R)	0	0 (G)
1	1	1	0 (R)	0	1 (Y)
Sb1	Sb0	La1	La0	Lb1	Lb0

26

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering UNIVERSITY OF TORONTO

Traffic Light FSM

- **A systematic approach to the design**
  1. Draw the state transition diagram
  2. Create the state transition table
  3. Encode the states on the state transition table
  4. Derive the logic expressions, minimize them
  5. Create the output table, encode it
  6. Derive the logic expressions, minimize them
  7. Draw the circuit, implement it
     - On "old-school" hardware, in code, on HDL

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering UNIVERSITY OF TORONTO

Traffic Light FSM

- **Output table** – we look at La and Lb, codified

Current		La		Lb	
0	0	0	0 (G)	1	0 (R)
0	1	0	1 (Y)	1	0 (R)
1	0	1	0 (R)	0	0 (G)
1	1	1	0 (R)	0	1 (Y)
Sb1	Sb0	La1	La0	Lb1	Lb0

$$La_1 = Sb_1$$

$$La_0 = \overline{Sb_1}\, Sb_0$$

$$Lb_1 = \overline{Sb_1}$$

$$Lb_0 = S_1 S_0$$

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Traffic Light FSM

- **A systematic approach to the design**
  1. Draw the state transition diagram
  2. Create the state transition table
  3. Encode the states on the state transition table
  4. Derive the logic expressions, minimize them
  5. Create the output table, encode it
  6. Derive the logic expressions, minimize them
  7. Draw the circuit, implement it
     - On “old-school” hardware, in code, on HDL

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

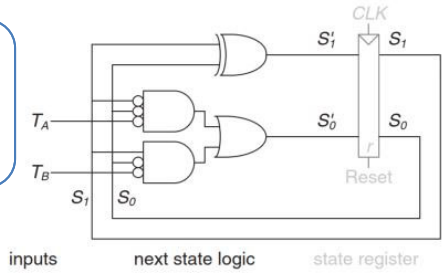
Traffic Light FSM

- We’ve got equations for next states from current states

$$S'b_1 = Sb_1 \oplus Sb_0$$

$$S'b_0 = \overline{Sb_1}\,\overline{Sb_0}\,\overline{T_a} + Sb_1\overline{Sb_0}\,\overline{T_b}$$

A 2 bit state register…



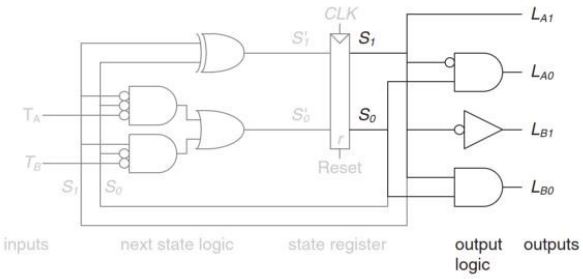
- But that does not say anything about the output…

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Traffic Light FSM

- We've ALSO got equations for outputs, which result in…

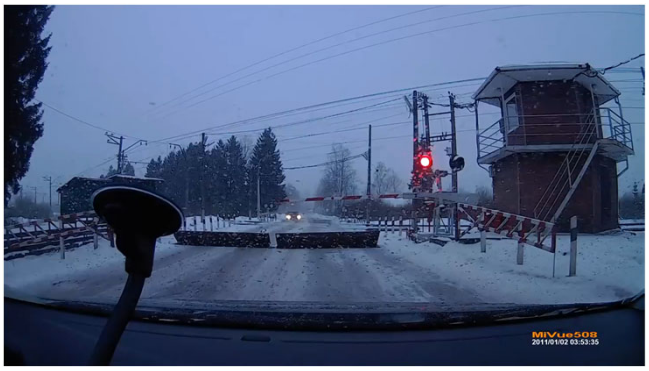
$$La_1 = Sb_1$$
$$La_0 = \overline{Sb_1}\, Sb$$
$$Lb_1 = \overline{Sb_1}$$
$$Lb_0 = S_1 S_0$$



Note: Outputs depend only on the present state ➔ MOORE MACHINE

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

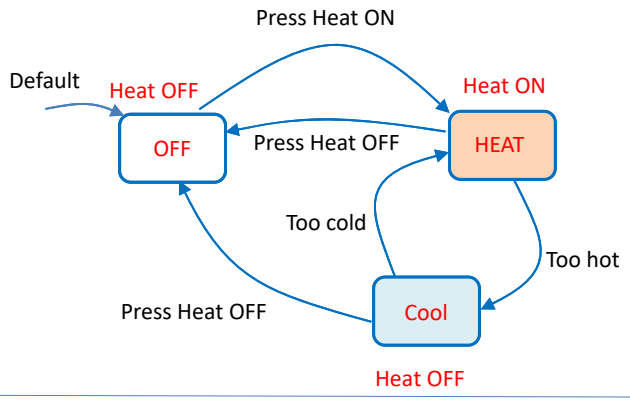
Traffic Light FSM



The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Moore Machine

- A machine whose present outputs are a function of the present state ONLY

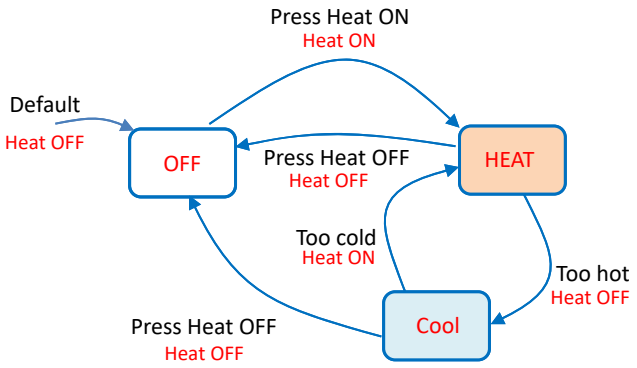


33

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Mealy Machine

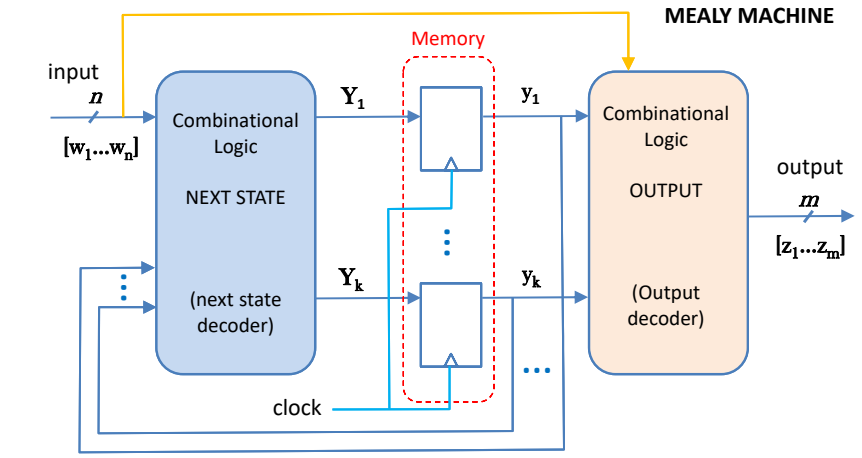
- A machine whose present outputs are a function of the present state and the present input.



34

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Definitions



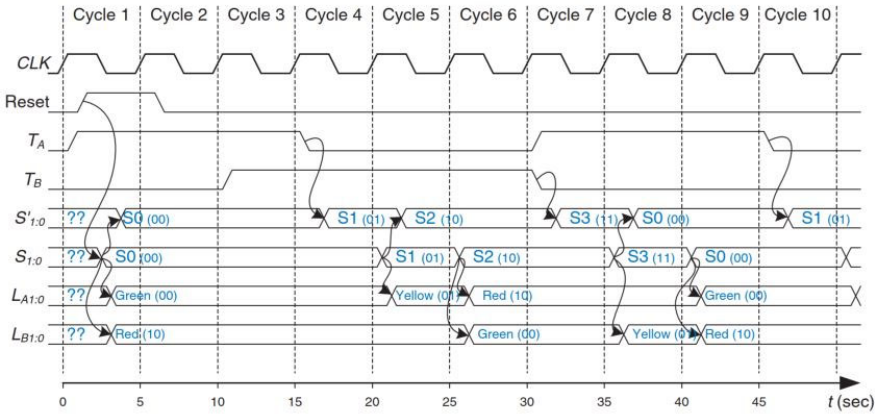
- On clock edge, Y_i becomes y_i

35

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Traffic Light FSM

- As for the timing of all this…



36