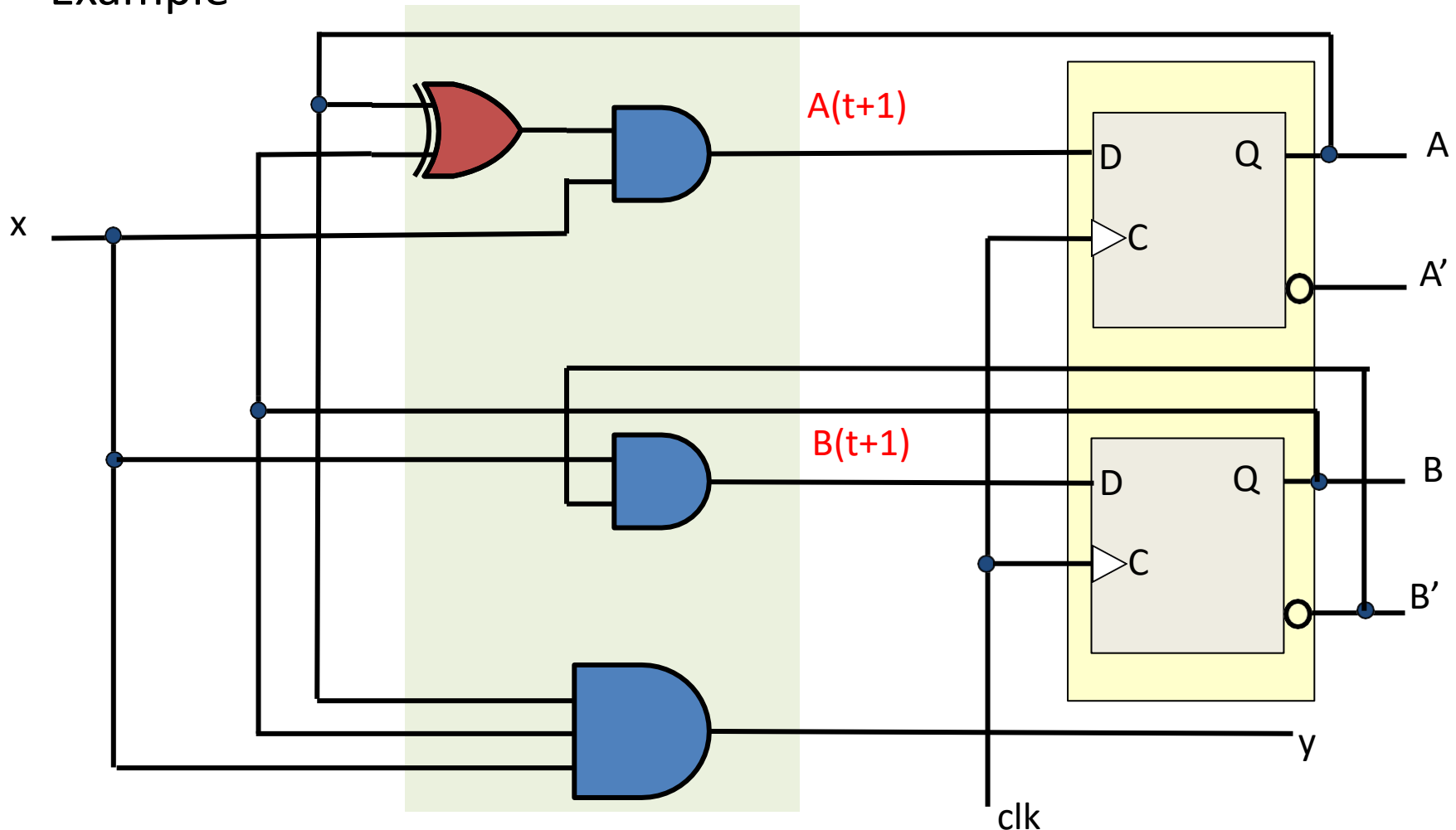


# Mantıksal Tasarım

13. HAFTA

# State Equations(Durum Denklemi)

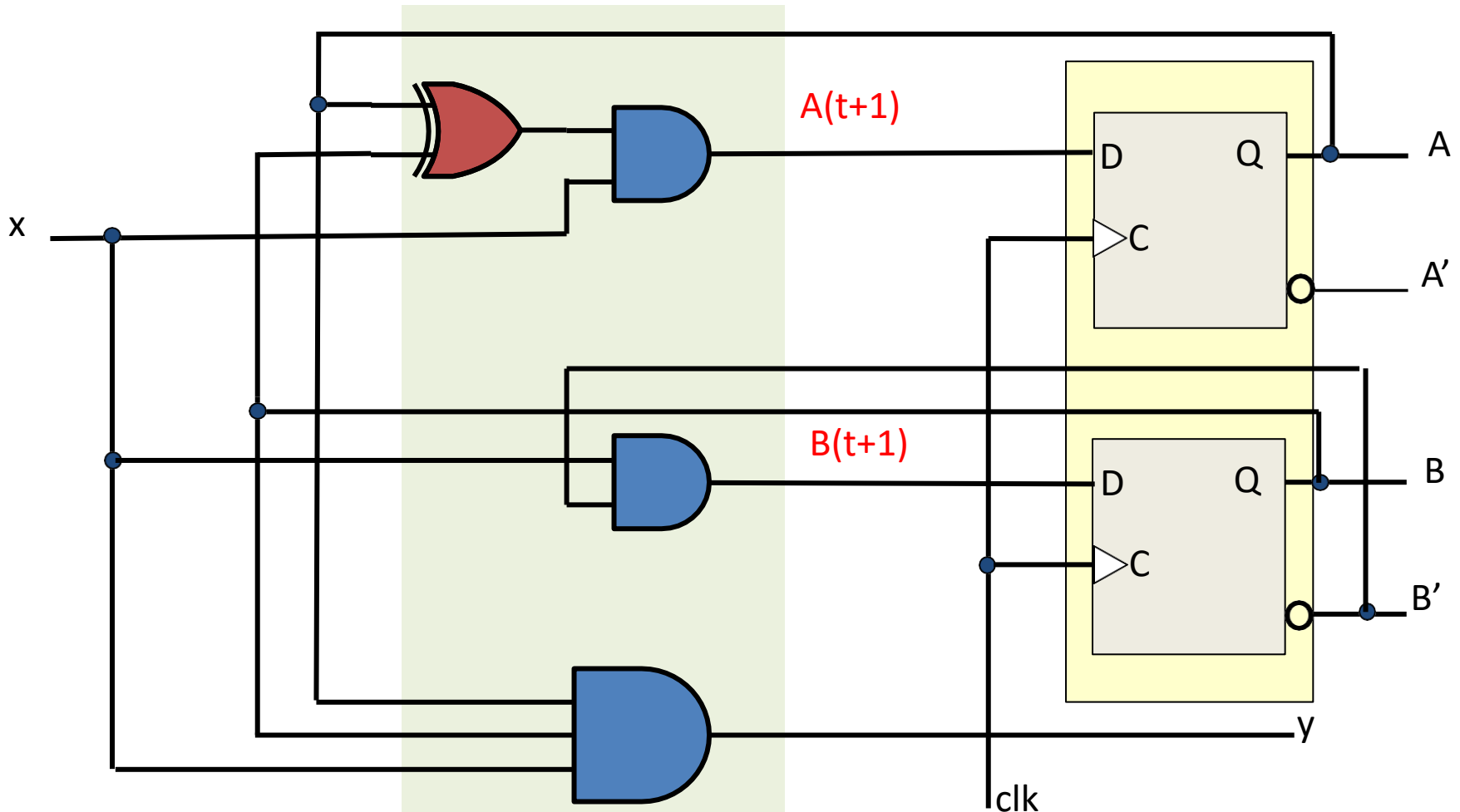
- Also known as “transition equations(geçiş denklemi)”
  - specify the next state as a function of the present state and inputs
- Example



# Output and State Equations

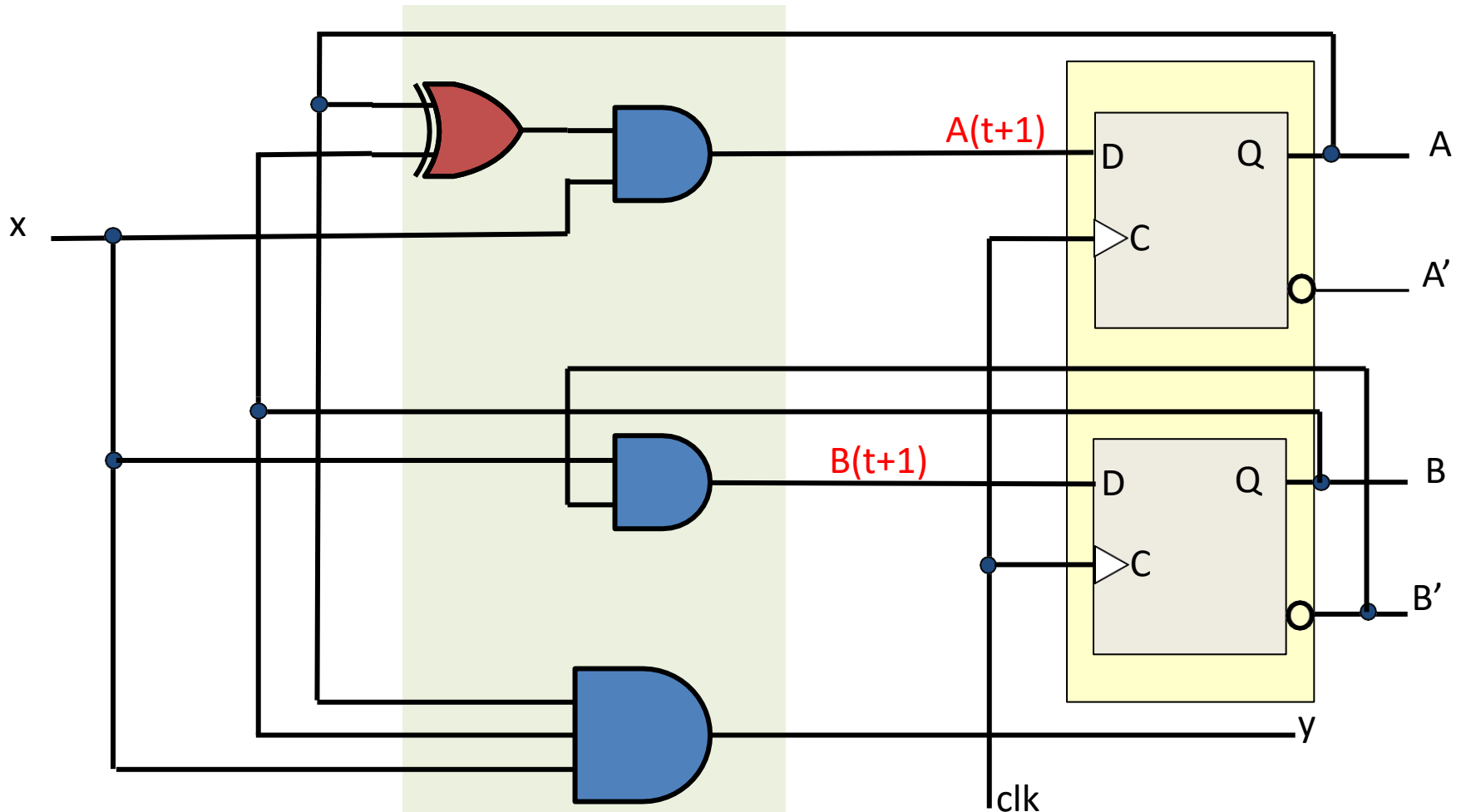
- $A(t+1) =$
- $B(t+1) =$
- $y =$

input of a flip-flop determines the value of the next state (i.e., the state reached after the clock transition)



# Output and State Equations

- $A(t+1) = (A(t) \oplus B(t)) \times$
- $B(t+1) = (B(t))' \times$
- $y = A(t)B(t) \times$



# Flip Flop Input Equations

- Flip-Flop input (excitation) equations
- Same as the state equations in D flip-flops

# Example: State (Transition) Table

$$A(t+1) = (AB' + A'B)x$$

$$B(t+1) = B'x$$

$$y = ABx$$

| Present state |      | input | Next state |        | output |
|---------------|------|-------|------------|--------|--------|
| A(t)          | B(t) | x     | A(t+1)     | B(t+1) | y      |
| 0             | 0    | 0     |            |        |        |
| 0             | 0    | 1     |            |        |        |
| 0             | 1    | 0     |            |        |        |
| 0             | 1    | 1     |            |        |        |
| 1             | 0    | 0     |            |        |        |
| 1             | 0    | 1     |            |        |        |
| 1             | 1    | 0     |            |        |        |
| 1             | 1    | 1     |            |        |        |

A sequential circuit with **m** FFs and **n** inputs needs  **$2^{m+n}$**  rows in the transition table

# Example: State (Transition) Table

$$A(t+1) = (AB' + A'B)x$$

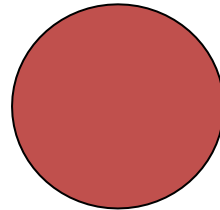
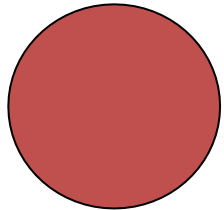
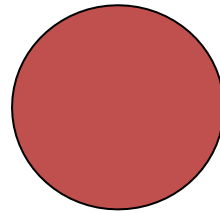
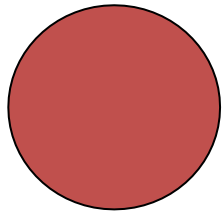
$$B(t+1) = B'x$$

$$y = ABx$$

| Present state |      | input | Next state |        | output |
|---------------|------|-------|------------|--------|--------|
| A(t)          | B(t) | x     | A(t+1)     | B(t+1) | y      |
| 0             | 0    | 0     | 0          | 0      | 0      |
| 0             | 0    | 1     | 0          | 1      | 0      |
| 0             | 1    | 0     | 0          | 0      | 0      |
| 0             | 1    | 1     | 1          | 0      | 0      |
| 1             | 0    | 0     | 0          | 0      | 0      |
| 1             | 0    | 1     | 1          | 1      | 0      |
| 1             | 1    | 0     | 0          | 0      | 0      |
| 1             | 1    | 1     | 0          | 0      | 1      |

A sequential circuit with **m** FFs and **n** inputs needs  **$2^{m+n}$**  rows in the transition table

# Example: State Diagram

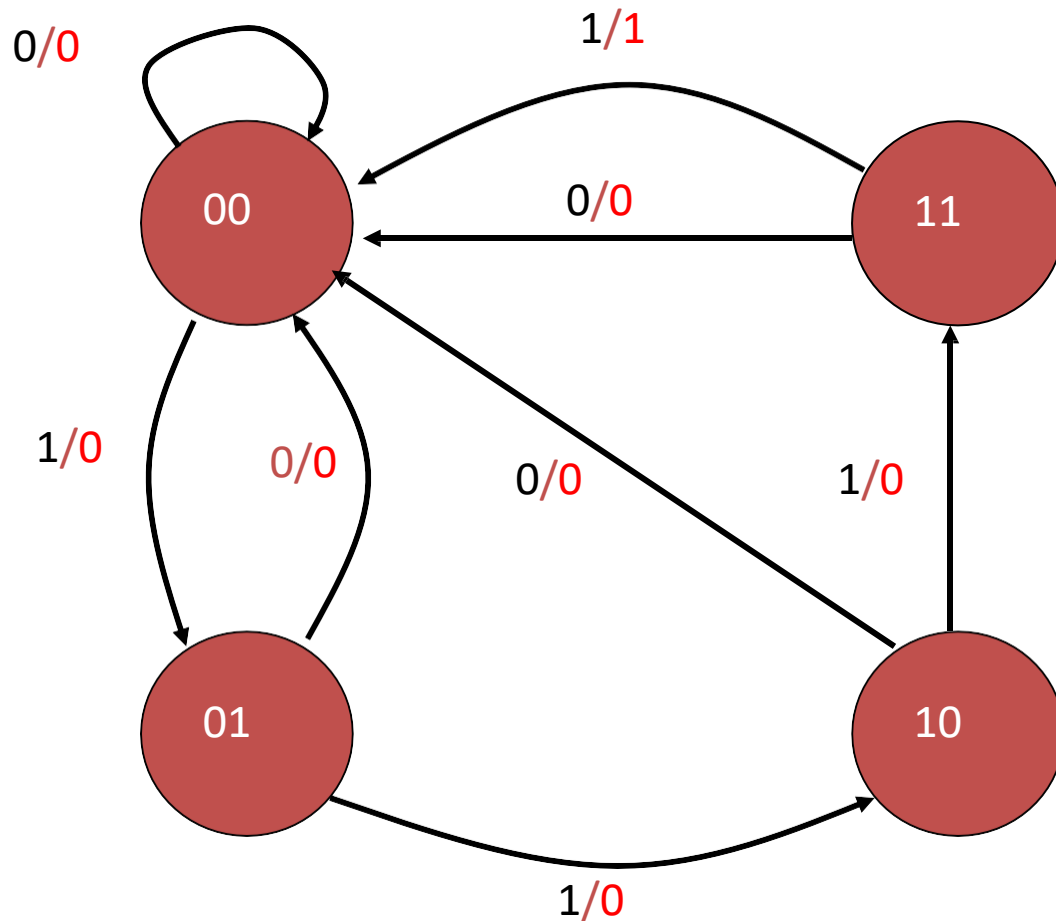


| Present state |   | input | Next state |   | output |
|---------------|---|-------|------------|---|--------|
| A             | B | x     | A          | B | y      |
| 0             | 0 | 0     | 0          | 0 | 0      |
| 0             | 0 | 1     | 0          | 1 | 0      |
| 0             | 1 | 0     | 0          | 0 | 0      |
| 0             | 1 | 1     | 1          | 0 | 0      |
| 1             | 0 | 0     | 0          | 0 | 0      |
| 1             | 0 | 1     | 1          | 1 | 0      |
| 1             | 1 | 0     | 0          | 0 | 0      |
| 1             | 1 | 1     | 0          | 0 | 1      |

What is this circuit doing?

State diagram provides the same information as state table

# Example: State Diagram



| Present state |   | input | Next state |   | output |
|---------------|---|-------|------------|---|--------|
| A             | B | x     | A          | B | y      |
| 0             | 0 | 0     | 0          | 0 | 0      |
| 0             | 0 | 1     | 0          | 1 | 0      |
| 0             | 1 | 0     | 0          | 0 | 0      |
| 0             | 1 | 1     | 1          | 0 | 0      |
| 1             | 0 | 0     | 0          | 0 | 0      |
| 1             | 0 | 1     | 1          | 1 | 0      |
| 1             | 1 | 0     | 0          | 0 | 0      |
| 1             | 1 | 1     | 0          | 0 | 1      |

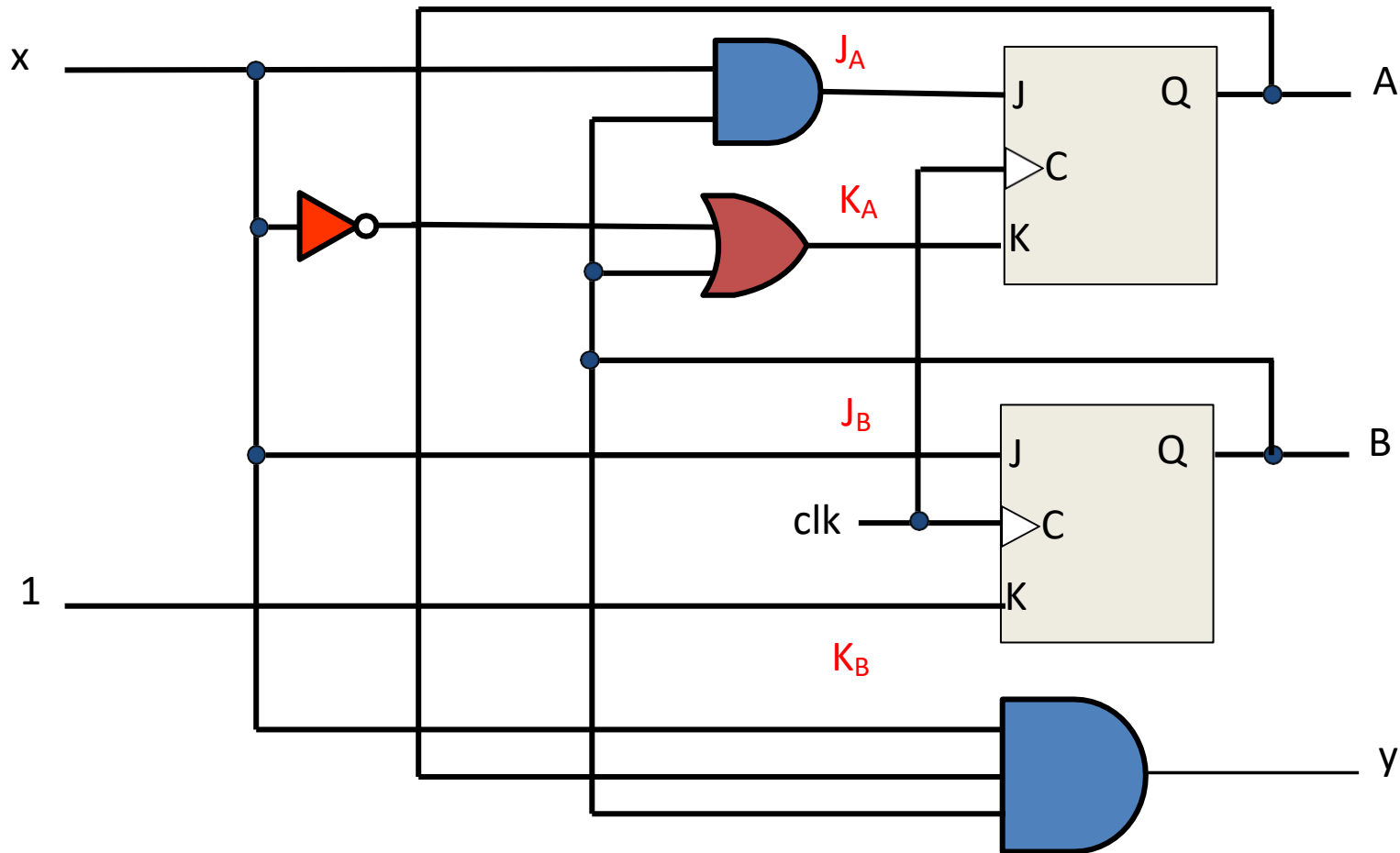
What is this circuit doing?

State diagram provides the same information as state table

# Analysis with JK Flip-Flops

- For a D flip-flop, the state equation is the same as the flip-flop input equation
  - $Q(t+1) = D$
- For JK flip-flops, situation is different
  - Goal is to find state equations
  - Method
    1. Determine flip-flop input equations
    2. List the binary values of each input equation
    3. Use the corresponding flip-flop characteristic table to determine the next state values in the state table

# Example: Analysis with JK FFs



- Flip-flop input equations

$$J_A = \quad \text{and } K_A =$$

$$J_B = \quad \text{and } K_B =$$

# Example: Analysis with JK FFs

- $J_A = Bx$  and  $K_A = x' + B$
- $J_B = x$  and  $K_B = 1$

| present State |      | input | next state |        | FF inputs      |                |                |                |
|---------------|------|-------|------------|--------|----------------|----------------|----------------|----------------|
| A(t)          | B(t) | x     | A(t+1)     | B(t+1) | J <sub>A</sub> | K <sub>A</sub> | J <sub>B</sub> | K <sub>B</sub> |
| 0             | 0    | 0     |            |        |                |                |                |                |
| 0             | 0    | 1     |            |        |                |                |                |                |
| 0             | 1    | 0     |            |        |                |                |                |                |
| 0             | 1    | 1     |            |        |                |                |                |                |
| 1             | 0    | 0     |            |        |                |                |                |                |
| 1             | 0    | 1     |            |        |                |                |                |                |
| 1             | 1    | 0     |            |        |                |                |                |                |
| 1             | 1    | 1     |            |        |                |                |                |                |

# Example: Analysis with JK FFs

- $J_A = Bx$  and  $K_A = x' + B$
- $J_B = x$  and  $K_B = 1$

| present State |      | input | next state |        | FF inputs |       |       |       |
|---------------|------|-------|------------|--------|-----------|-------|-------|-------|
| A(t)          | B(t) | x     | A(t+1)     | B(t+1) | $J_A$     | $K_A$ | $J_B$ | $K_B$ |
| 0             | 0    | 0     |            |        | 0         | 1     | 0     | 1     |
| 0             | 0    | 1     |            |        | 0         | 0     | 1     | 1     |
| 0             | 1    | 0     |            |        | 0         | 1     | 0     | 1     |
| 0             | 1    | 1     |            |        | 1         | 1     | 1     | 1     |
| 1             | 0    | 0     |            |        | 0         | 1     | 0     | 1     |
| 1             | 0    | 1     |            |        | 0         | 0     | 1     | 1     |
| 1             | 1    | 0     |            |        | 0         | 1     | 0     | 1     |
| 1             | 1    | 1     |            |        | 1         | 1     | 1     | 1     |

# Example: Analysis with JK FFs

- $J_A = Bx$  and  $K_A = x' + B$
- $J_B = x$  and  $K_B = 1$

| present State |      | input | next state |        | FF inputs |       |       |       |
|---------------|------|-------|------------|--------|-----------|-------|-------|-------|
| A(t)          | B(t) | x     | A(t+1)     | B(t+1) | $J_A$     | $K_A$ | $J_B$ | $K_B$ |
| 0             | 0    | 0     | 0          | 0      | 0         | 1     | 0     | 1     |
| 0             | 0    | 1     | 0          | 1      | 0         | 0     | 1     | 1     |
| 0             | 1    | 0     | 0          | 0      | 0         | 1     | 0     | 1     |
| 0             | 1    | 1     | 1          | 0      | 1         | 1     | 1     | 1     |
| 1             | 0    | 0     | 0          | 0      | 0         | 1     | 0     | 1     |
| 1             | 0    | 1     | 1          | 1      | 0         | 0     | 1     | 1     |
| 1             | 1    | 0     | 0          | 0      | 0         | 1     | 0     | 1     |
| 1             | 1    | 1     | 0          | 0      | 1         | 1     | 1     | 1     |

# Example: Analysis with JK FFs

- Characteristic equations

- $A(t+1) = J_A A' + K'_A A$
- $B(t+1) = J_B B' + K'_B B$
- $y = ABx$

- Input equations

- $J_A = Bx$  and  $K_A = x' + B$
- $J_B = x$  and  $K_B = 1$

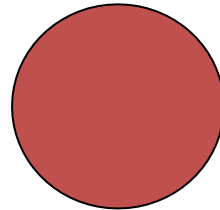
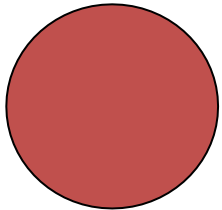
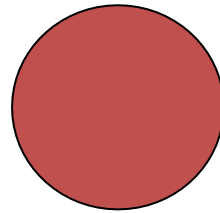
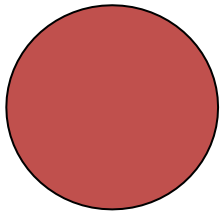
- State equations

- $A(t+1) =$   
       $=$
- $B(t+1) =$

# Example: Analysis with JK FFs

- Characteristic equations
  - $A(t+1) = J_A A' + K'_A A$
  - $B(t+1) = J_B B' + K'_B B$
  - $y = ABx$
- Input equations
  - $J_A = Bx$  and  $K_A = x' + B$
  - $J_B = x$  and  $K_B = 1$
- State equations
  - $A(t+1) = A'Bx + (x' + B)' A$   
 $= A'Bx + AB'x = (A \oplus B) x$
  - $B(t+1) = B'x$

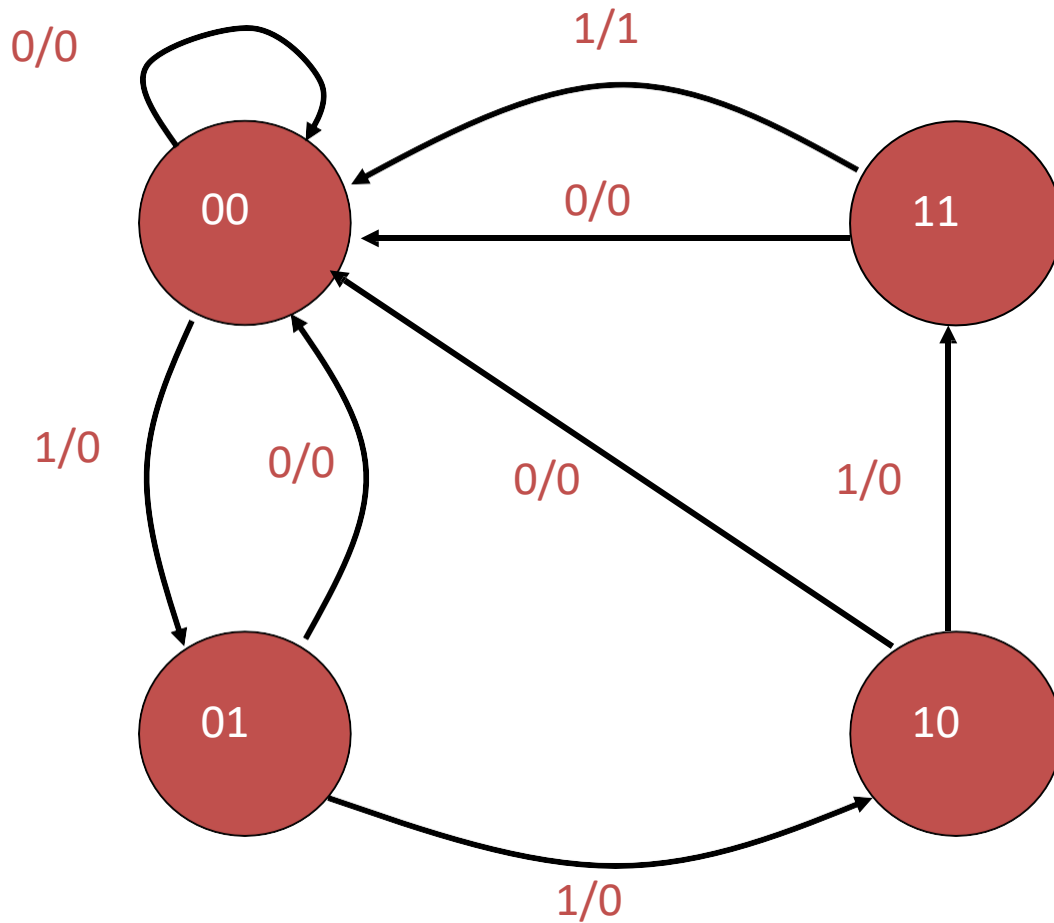
# State Diagram



| Present state |   | input | Next state |   | output |
|---------------|---|-------|------------|---|--------|
| A             | B | x     | A          | B | y      |
| 0             | 0 | 0     | 0          | 0 | 0      |
| 0             | 0 | 1     | 0          | 1 | 0      |
| 0             | 1 | 0     | 0          | 0 | 0      |
| 0             | 1 | 1     | 1          | 0 | 0      |
| 1             | 0 | 0     | 0          | 0 | 0      |
| 1             | 0 | 1     | 1          | 1 | 0      |
| 1             | 1 | 0     | 0          | 0 | 0      |
| 1             | 1 | 1     | 0          | 0 | 1      |

What is the circuit  
doing?

# State Diagram

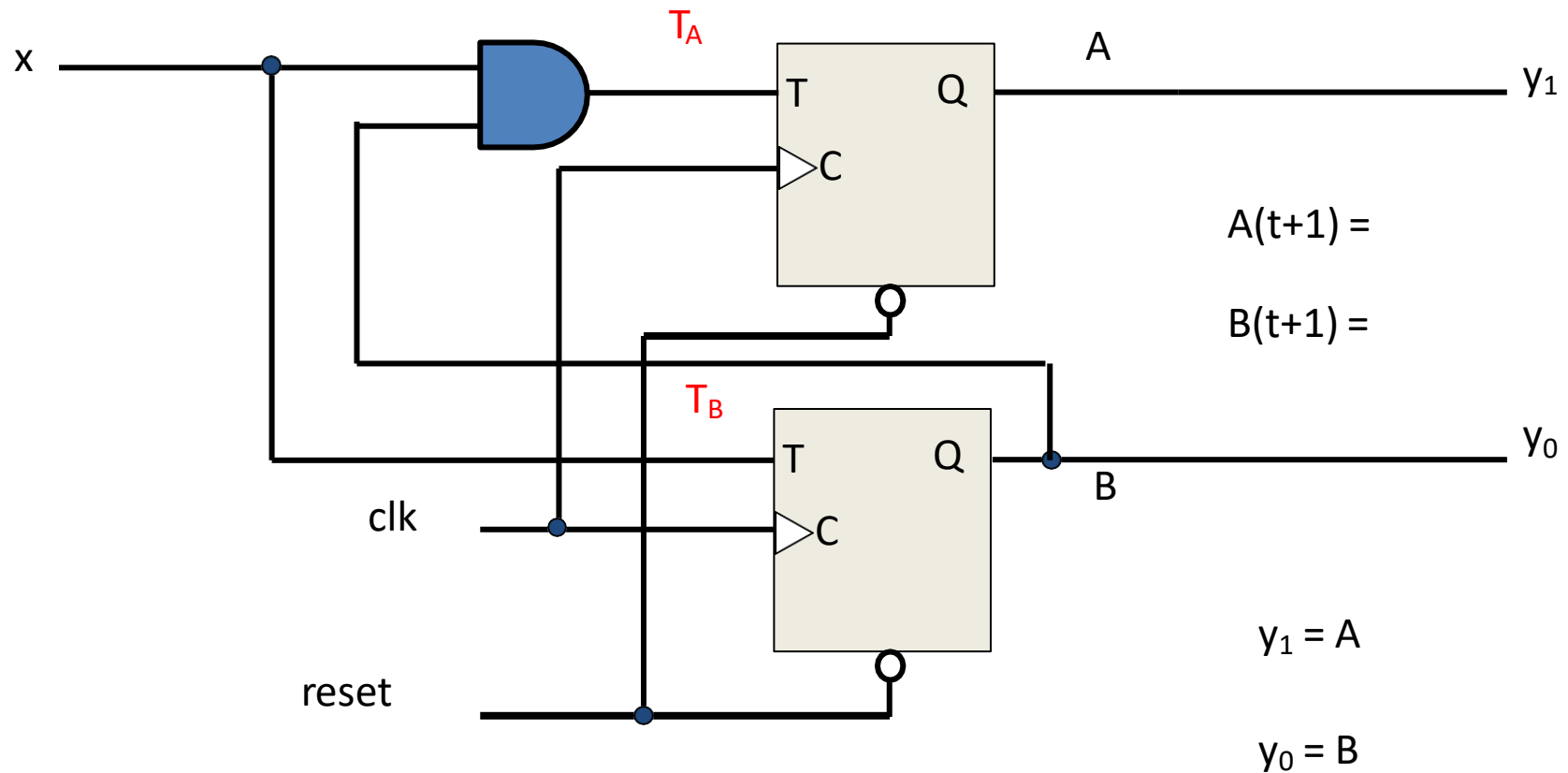


| Present state |   | input |  | Next state |   | output |
|---------------|---|-------|--|------------|---|--------|
| A             | B | x     |  | A          | B | y      |
| 0             | 0 | 0     |  | 0          | 0 | 0      |
| 0             | 0 | 1     |  | 0          | 1 | 0      |
| 0             | 1 | 0     |  | 0          | 0 | 0      |
| 0             | 1 | 1     |  | 1          | 0 | 0      |
| 1             | 0 | 0     |  | 0          | 0 | 0      |
| 1             | 0 | 1     |  | 1          | 1 | 0      |
| 1             | 1 | 0     |  | 0          | 0 | 0      |
| 1             | 1 | 1     |  | 0          | 0 | 1      |

What is the circuit doing?

# Analysis with T Flip-Flops

- Method is the same
- Example



# Example: Analysis with T Flip-Flops

- Characteristic equation

- $A(t+1) = T_A \oplus A$

- $B(t+1) = T_B \oplus B$

- Input equations

- $T_A = Bx$

- $T_B = x$

- Output equations

- $y_1 = A$

- $y_0 = B$

- State equations

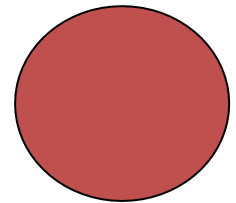
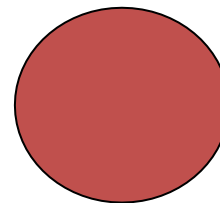
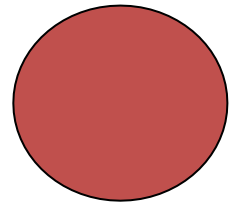
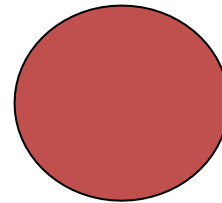
- $A(t+1) =$

- $B(t+1) =$

# State Table & Diagram

- $A(t+1) = Bx \oplus A$
- $B(t+1) = x \oplus B$
- $y_1 = A; y_0 = B$

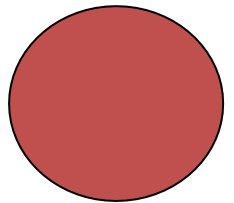
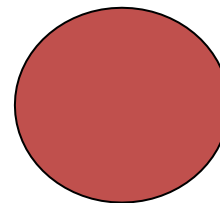
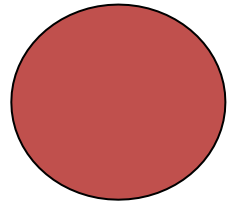
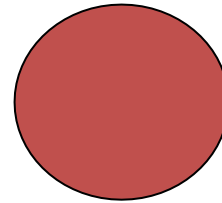
| Present state |   | input<br>x | Next state |   | output         |                |
|---------------|---|------------|------------|---|----------------|----------------|
| A             | B |            | A          | B | y <sub>1</sub> | y <sub>0</sub> |
| 0             | 0 | 0          |            |   |                |                |
| 0             | 0 | 1          |            |   |                |                |
| 0             | 1 | 0          |            |   |                |                |
| 0             | 1 | 1          |            |   |                |                |
| 1             | 0 | 0          |            |   |                |                |
| 1             | 0 | 1          |            |   |                |                |
| 1             | 1 | 0          |            |   |                |                |
| 1             | 1 | 1          |            |   |                |                |



# State Table & Diagram

- $A(t+1) = xB \oplus A$
- $B(t+1) = x \oplus B$
- $y_1 = A; y_0 = B$

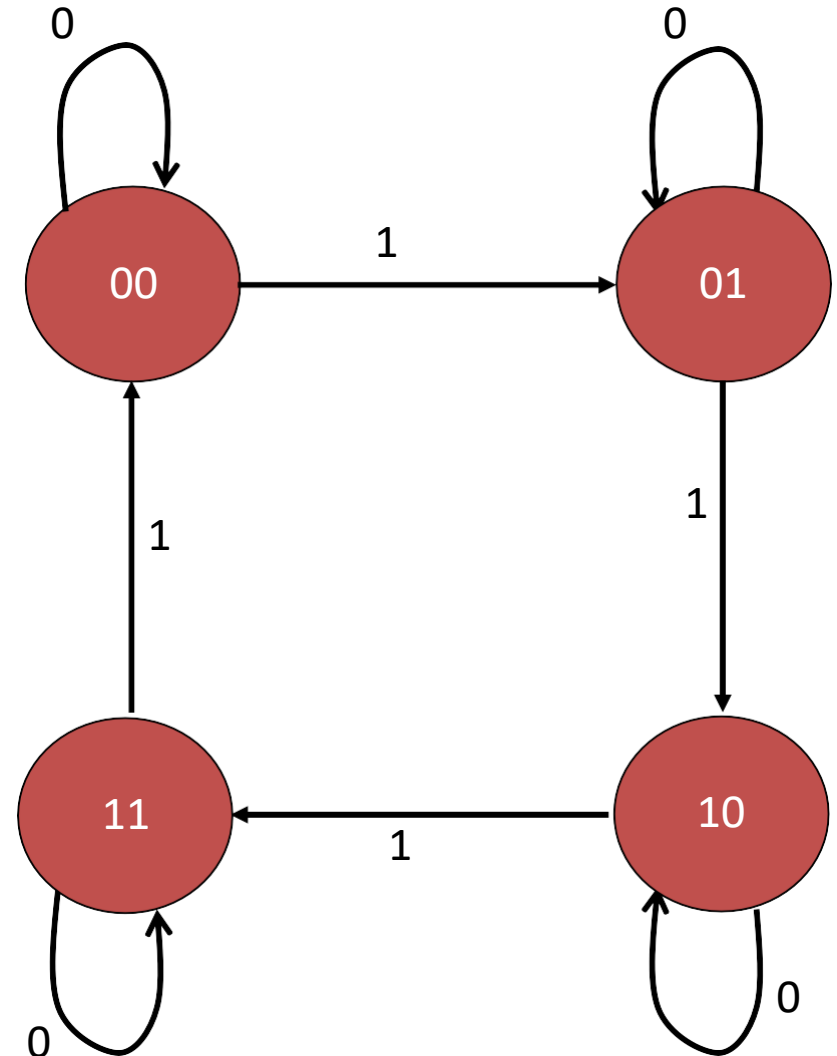
| Present state |   | input | Next state |   | output |       |
|---------------|---|-------|------------|---|--------|-------|
| A             | B |       | A          | B | $y_1$  | $y_0$ |
| 0             | 0 | 0     | 0          | 0 | 0      | 0     |
| 0             | 0 | 1     | 0          | 1 | 0      | 0     |
| 0             | 1 | 0     | 0          | 1 | 0      | 1     |
| 0             | 1 | 1     | 1          | 0 | 0      | 1     |
| 1             | 0 | 0     | 1          | 0 | 1      | 0     |
| 1             | 0 | 1     | 1          | 1 | 1      | 0     |
| 1             | 1 | 0     | 1          | 1 | 1      | 1     |
| 1             | 1 | 1     | 0          | 0 | 1      | 1     |



# State Table & Diagram

- $A(t+1) = xB \oplus A$
- $B(t+1) = x \oplus B$
- $y_1 = A; y_0 = B$

| Present state |     | input<br>$x$ | Next state |     | output |       |
|---------------|-----|--------------|------------|-----|--------|-------|
| $A$           | $B$ |              | $A$        | $B$ | $y_1$  | $y_0$ |
| 0             | 0   | 0            | 0          | 0   | 0      | 0     |
| 0             | 0   | 1            | 0          | 1   | 0      | 0     |
| 0             | 1   | 0            | 0          | 1   | 0      | 1     |
| 0             | 1   | 1            | 1          | 0   | 0      | 1     |
| 1             | 0   | 0            | 1          | 0   | 1      | 0     |
| 1             | 0   | 1            | 1          | 1   | 1      | 0     |
| 1             | 1   | 0            | 1          | 1   | 1      | 1     |
| 1             | 1   | 1            | 0          | 0   | 1      | 1     |

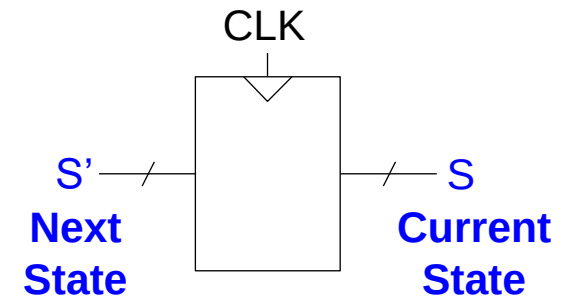


# Finite State Machine (FSM)

- Consists of:

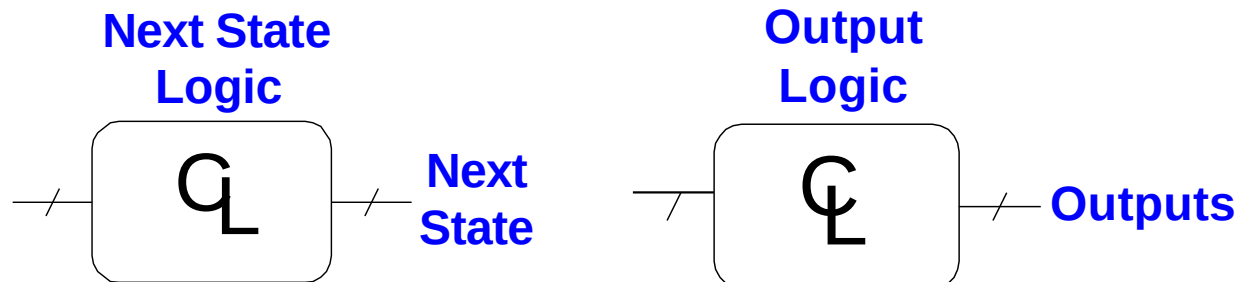
- **State register**

- Stores current state
    - Loads next state at clock edge



- **Combinational logic**

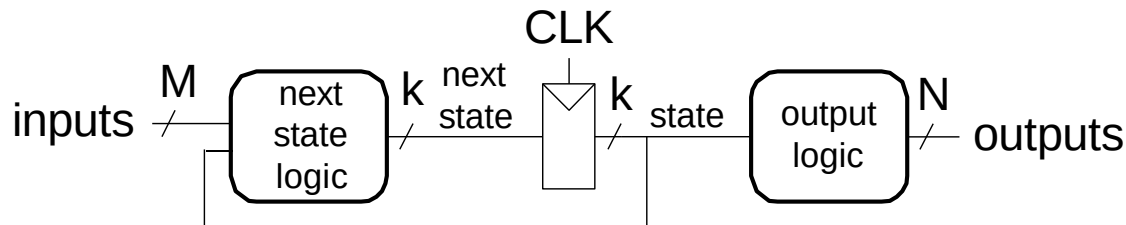
- Computes the next state
    - Computes the outputs



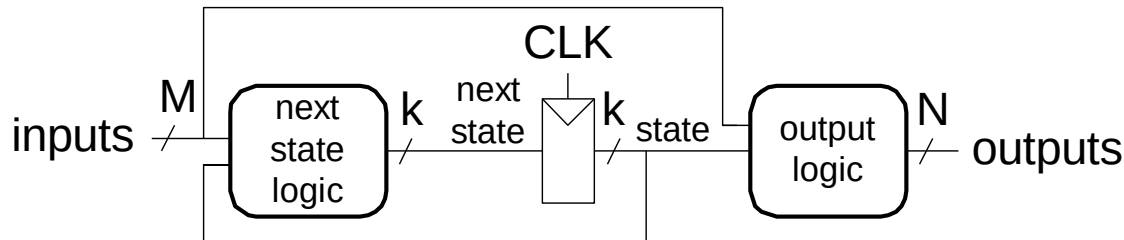
# Finite State Machines (FSMs)

- Next state determined by current state and inputs
- Two types of finite state machines differ in output logic:
  - **Moore FSM:** outputs depend only on current state
  - **Mealy FSM:** outputs depend on current state *and* inputs

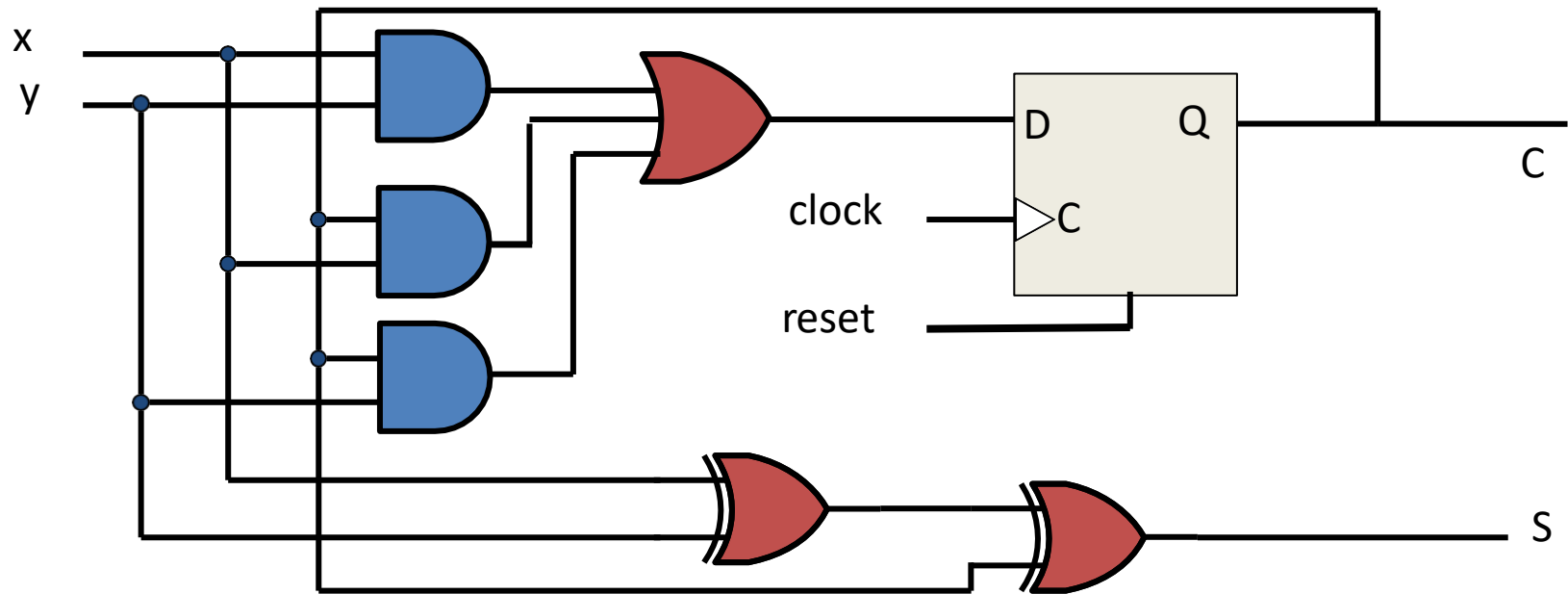
## Moore FSM



## Mealy FSM



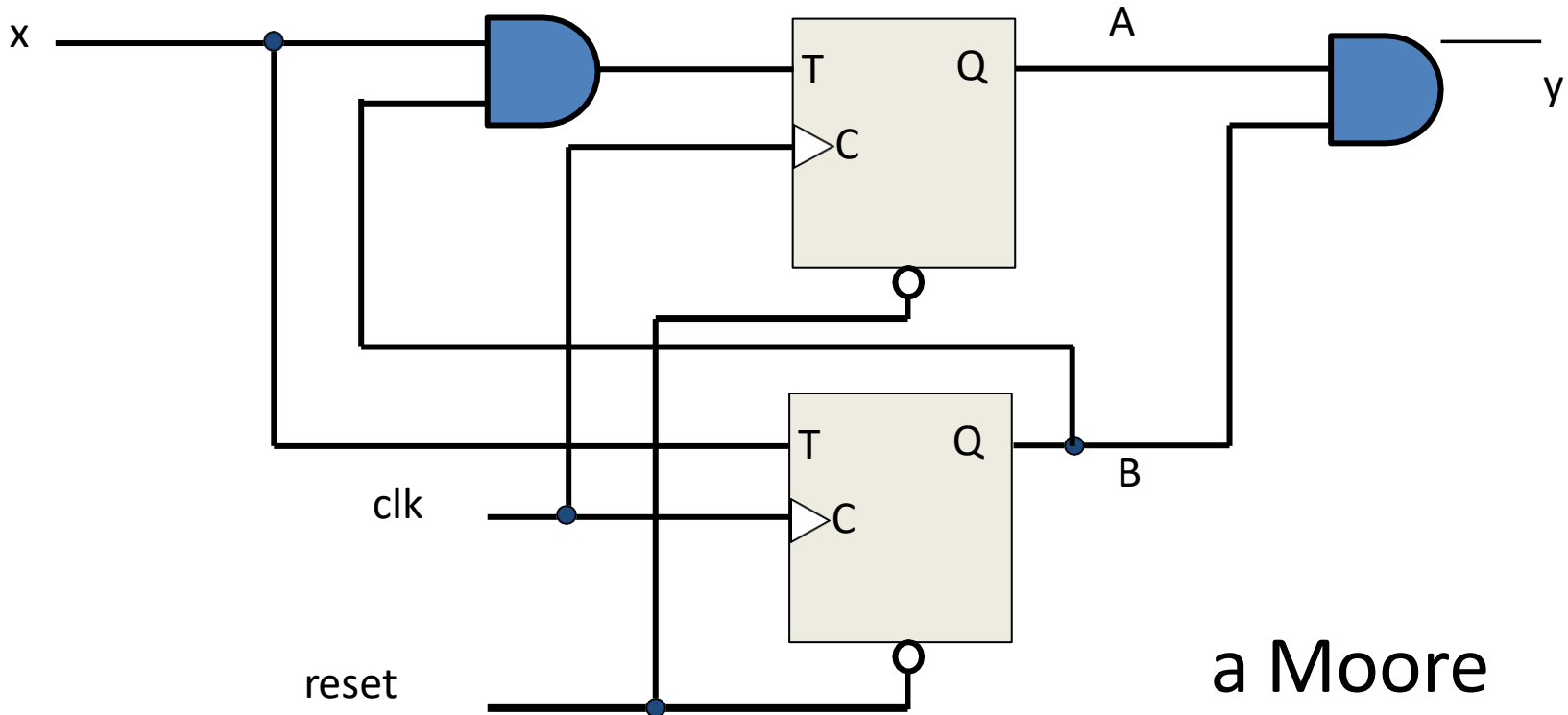
# What kind is this?



a Mealy  
machine

- External inputs  $x$  and  $y$  are asynchronous
- Thus, outputs may have momentary (incorrect) values
- Inputs must be synchronized with clocks
- Outputs must be sampled only during clock edges

# What kind is this?

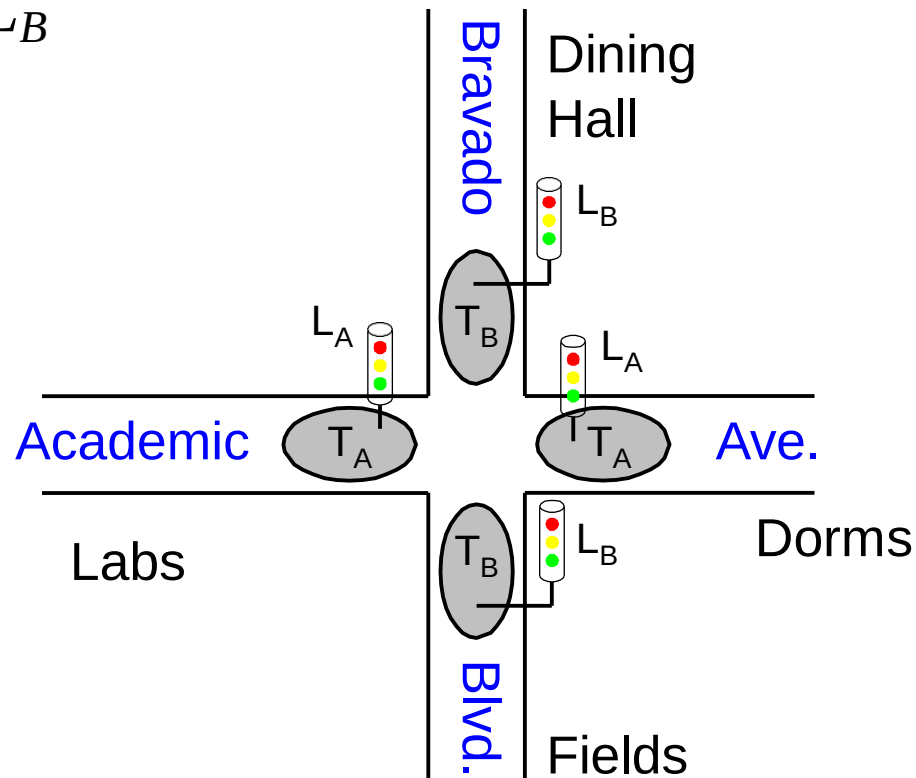


a Moore  
machine

- Outputs are already synchronized with clock.
- They change synchronously with the clock edge.

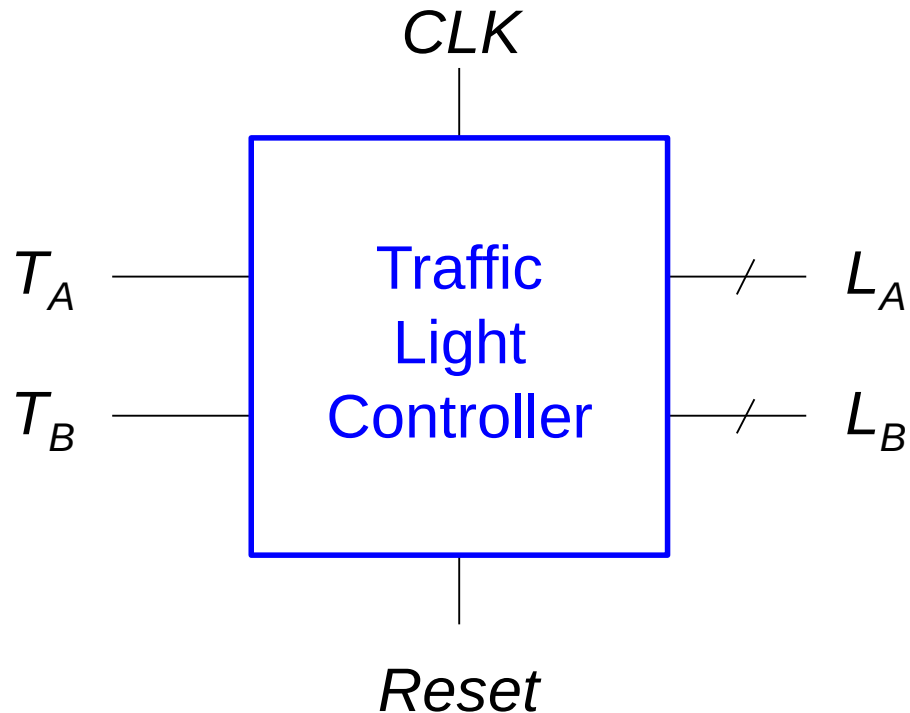
# FSM Example

- Traffic light controller
  - Traffic sensors:  $T_A$ ,  $T_B$  (TRUE when there's traffic)
  - Lights:  $L_A$ ,  $L_B$



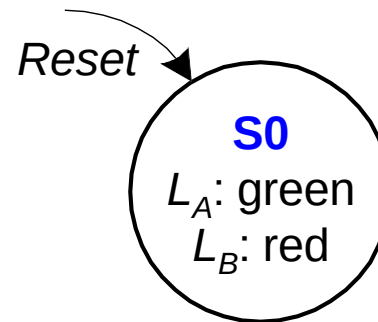
# FSM Black Box

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



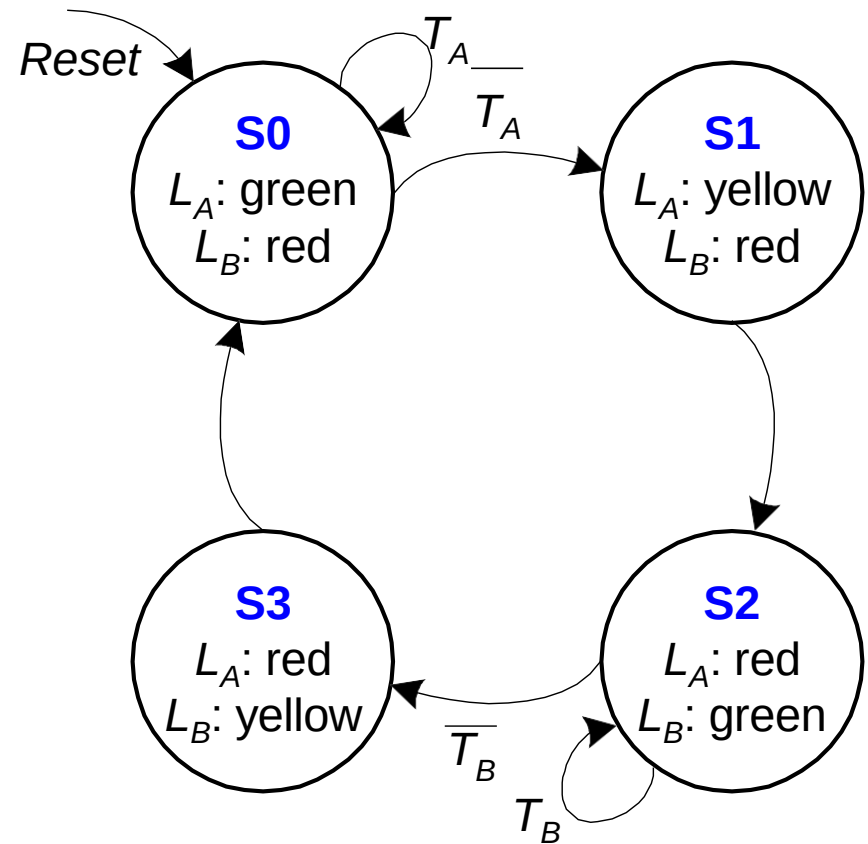
# FSM State Transition Diagram

- **Moore FSM:** outputs labeled in each state
- **States:** Circles
- **Transitions:** Arcs



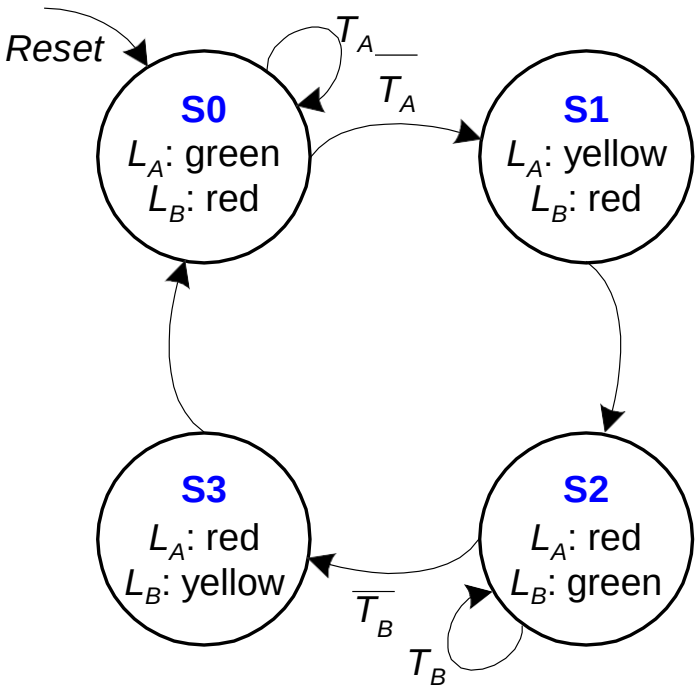
# FSM State Transition Diagram

- **Moore FSM:** outputs labeled in each state
- **States:** Circles
- **Transitions:** Arcs



# FSM State Transition Table

| Current State | Inputs |       | Next State |
|---------------|--------|-------|------------|
|               | $T_A$  | $T_B$ |            |
| S0            | 0      | X     |            |
| S0            | 1      | X     |            |
| S1            | X      | X     |            |
| S2            | X      | 0     |            |
| S2            | X      | 1     |            |
| S3            | X      | X     |            |



# FSM State Transition Table

| Current State<br>$S$ | Inputs |       | Next State<br>$S'$ |
|----------------------|--------|-------|--------------------|
|                      | $T_A$  | $T_B$ |                    |
| S0                   | 0      | X     | S1                 |
| S0                   | 1      | X     | S0                 |
| S1                   | X      | X     | S2                 |
| S2                   | X      | 0     | S3                 |
| S2                   | X      | 1     | S2                 |
| S3                   | X      | X     | S0                 |

