# Chapter 3 :: Sequential Logic Design (3)

***Digital Design and Computer Architecture*, 2nd Edition**

David Money Harris and Sarah L. Harris

# Chapter 3 :: Topics

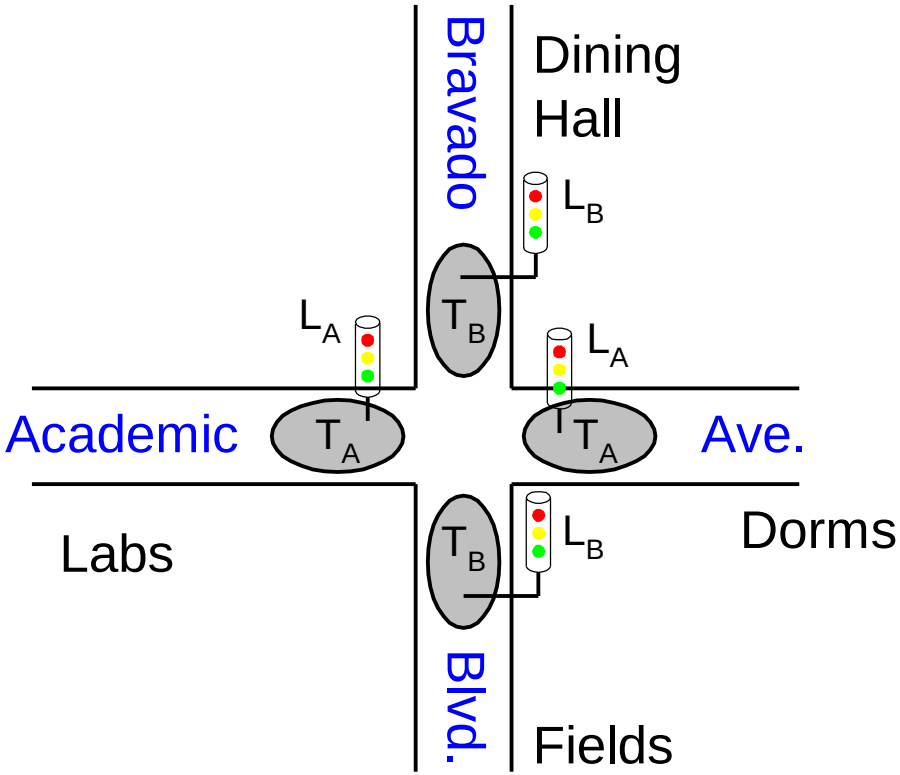
- Introduction
- Latches and Flip-Flops
- Synchronous Logic Design

- Finite State Machines

- Shifter
- Timing of Sequential Logic
- Parallelism

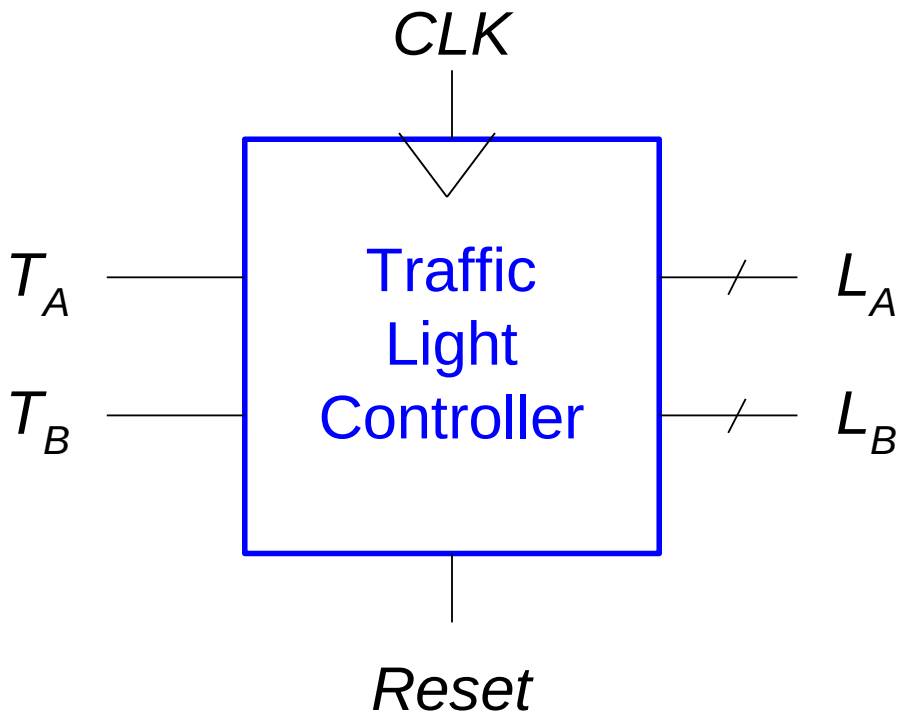
# 3.4 1 Finite State Machine Example

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



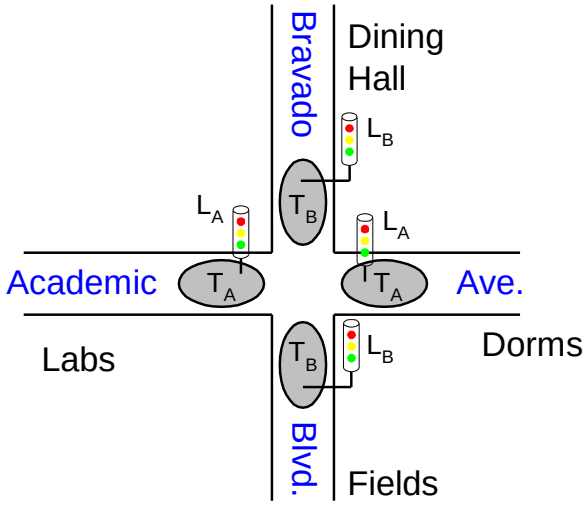
# FSM Black Box

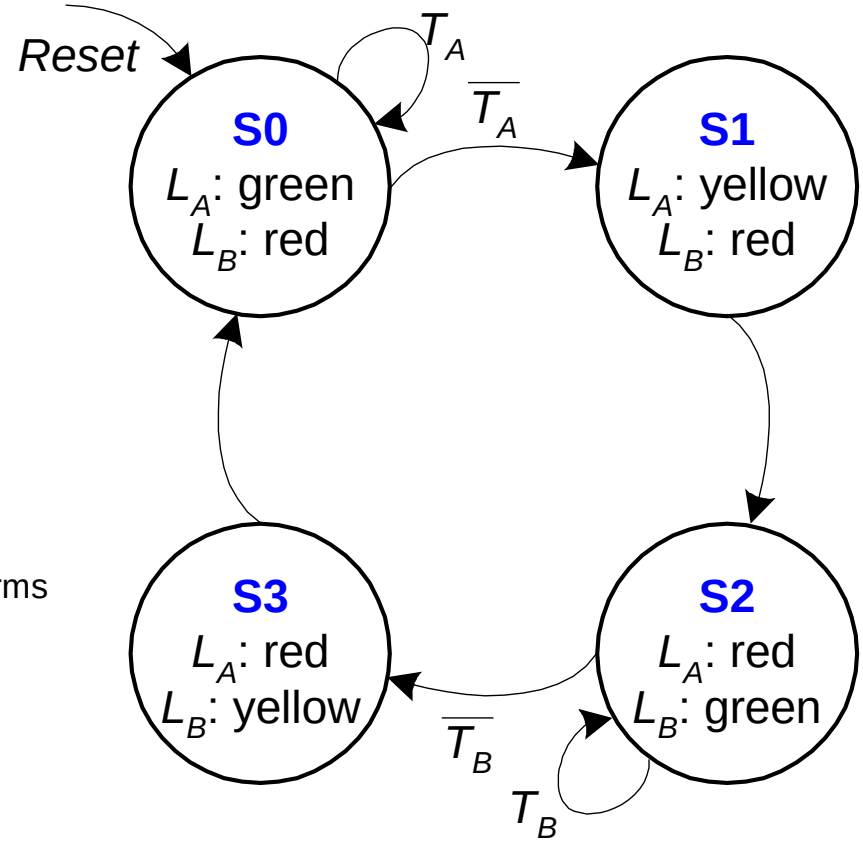
- Inputs: *CLK*, *Reset*, $T_A$, $T_B$
- Outputs: $L_A$, $L_B$



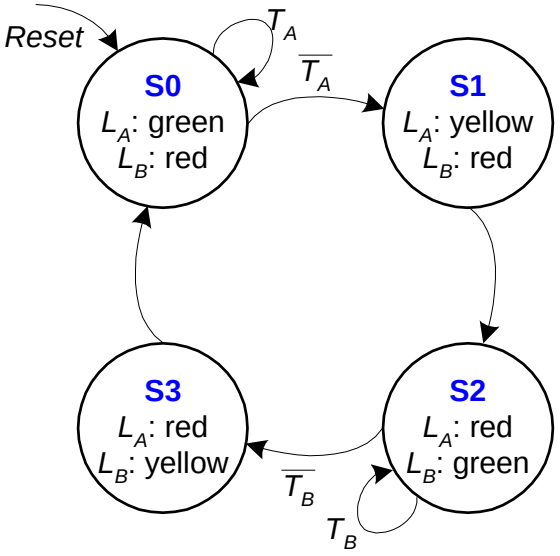
# FSM State Transition Diagram

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



# FSM State Transition Table




| Current State | Inputs | | Next State |
|---|---|---|---|
| $S$ | $T_A$ | $T_B$ | $S_{next}$ |
| S0 | 0 | X | S1 |
| S0 | 1 | X | S0 |
| S1 | X | X | S2 |
| S2 | X | 0 | S3 |
| S2 | X | 1 | S2 |
| S3 | X | X | S0 |

# FSM Encoded State Transition Table

| Current State | Inputs | | Next State |
|---|---|---|---|
| *State* | $T_A$ | $T_B$ | *Snext* |
| 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 (2진 부호화) |
|---|---|
| S0 | 00 |
| S1 | 01 |
| S2 | 10 |
| S3 | 11 |

| Current State | | Inputs | | Next State | |
|---|---|---|---|---|---|
| $S_1$ | $S_0$ | $T_A$ | $T_B$ | $S_{1next}$ | $S_{0next}$ |
| 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 |

$$S_{1next} = S_1 \oplus S_0$$

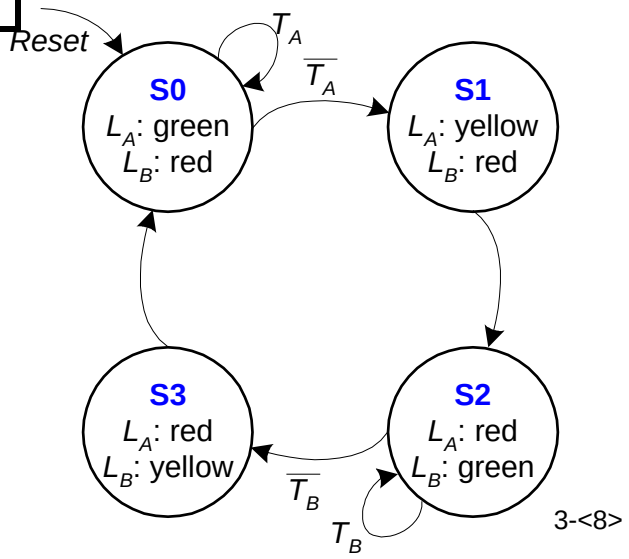
$$S_{0next} = S_1'S_0'T_A' + S_1S_0'T_B'$$

# FSM Output Table

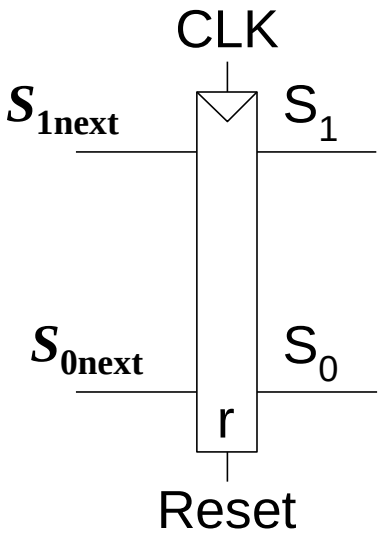
| Current State | | | Outputs | | | |
|---|---|---|---|---|---|---|
| *State* | $S_1$ | $S_0$ | $L_{A1}$ | $L_{A0}$ | $L_{B1}$ | $L_{B0}$ |
| S0 | 0 | 0 | 0 | 0 | 1 | 0 |
| S1 | 0 | 1 | 0 | 1 | 1 | 0 |
| S2 | 1 | 0 | 1 | 0 | 0 | 0 |
| S3 | 1 | 1 | 1 | 0 | 0 | 1 |

| Output | Encoding |
|---|---|
| green | 00 |
| yellow | 01 |
| red | 10 |

$$L_{A1} = S_1$$
$$L_{A0} = S_1'S_0$$
$$L_{B1} = S_1'$$
$$L_{B0} = S_1S_0$$

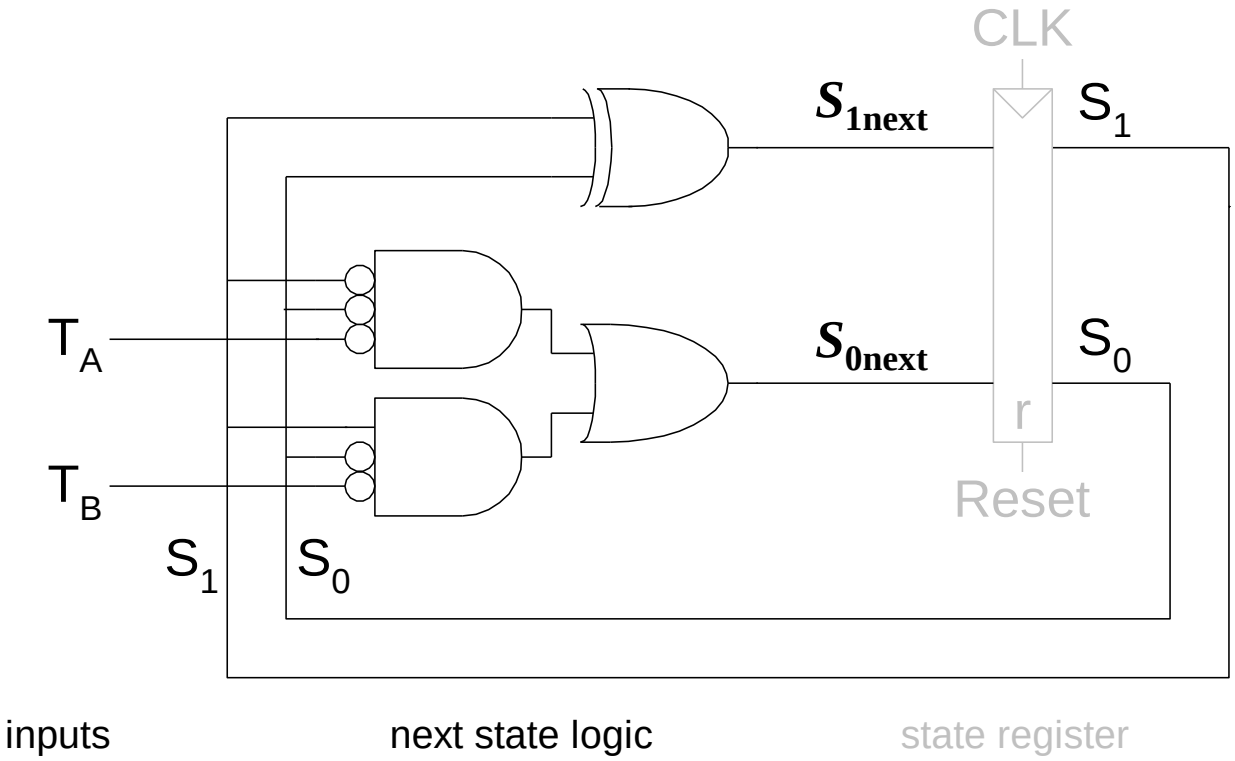


# FSM Schematic: State Register




state register

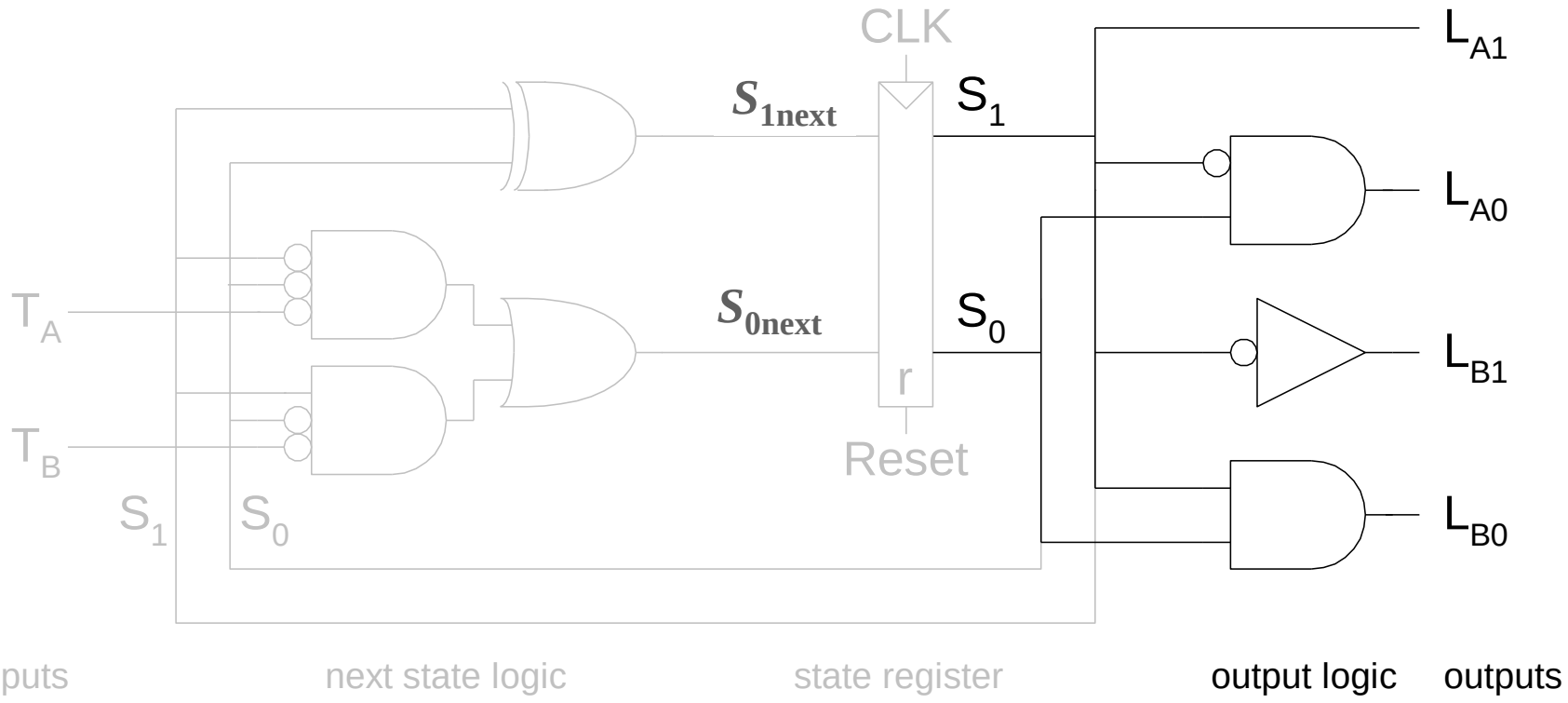
# FSM Schematic: Next State Logic




$$S_{1\text{next}} = S_1 \oplus S_0$$

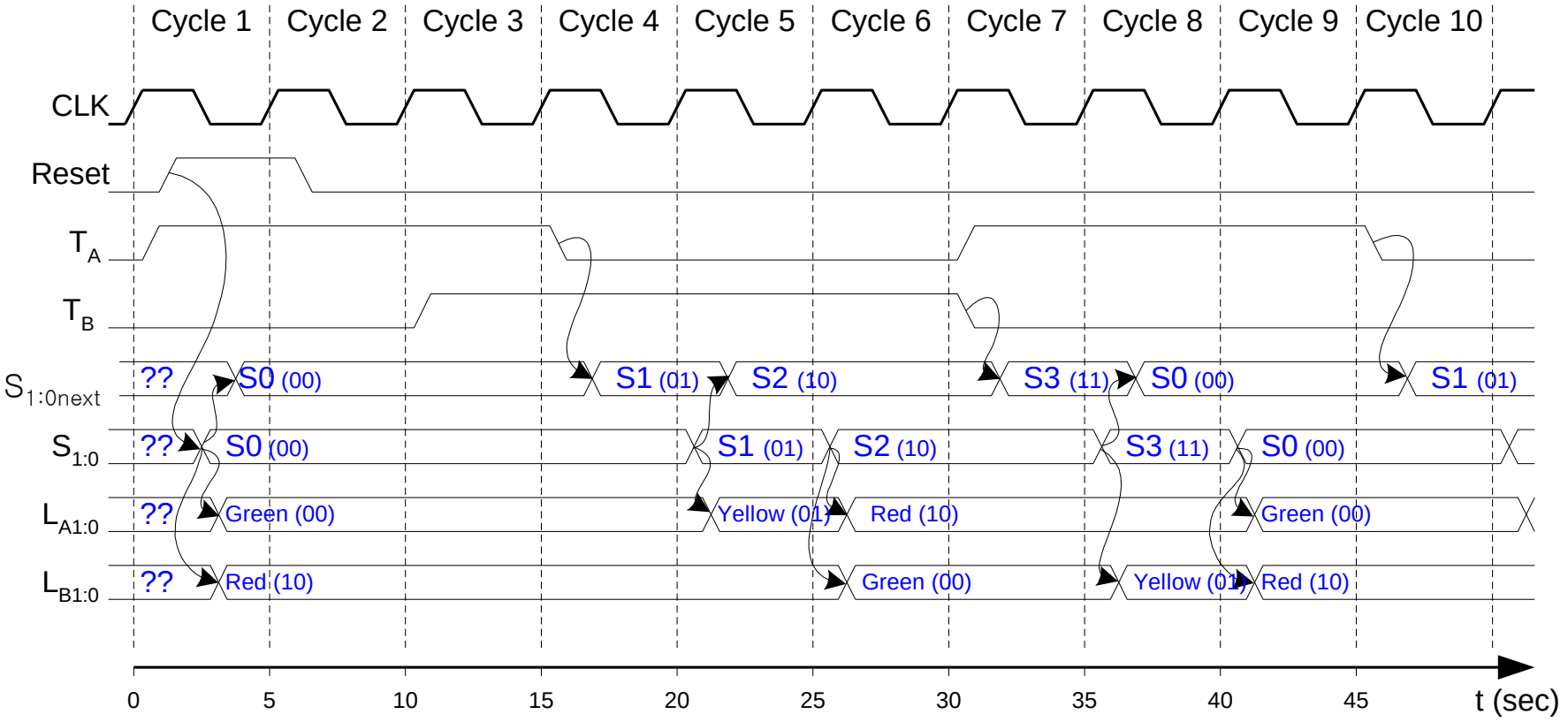
$$S_{0\text{next}} = S_1'S_0'T_A' + S_1S_0'T_B'$$

# FSM Schematic: Output Logic




$$L_{A1} = S_1$$
$$L_{A0} = S_1'S_0$$
$$L_{B1} = S_1'$$
$$L_{B0} = S_1S_0$$

# FSM Timing Diagram



# 3.4.2 FSM State Encoding(상태부호화)

- Binary encoding
  - i.e., for four states, 00, 01, 10, 11

- One-hot encoding
  - One state bit per state(각각의 상태를 위해 하나의 비트가 사용됨)
  - Only one state bit is HIGH at once(오직 하나의 비트만 참)
  - I.e., for four states, 0001, 0010, 0100, 1000

  - Requires more flip-flops(더 많은 수의 FF를 요구)
  - Often next state and output logic is simpler(다음 상태와 출력 논리는 더 간단)

# 예제 **3.6 (p135)**

- N진 카운터는 하나의 출력을 가지고 입력은 가지지 않는 다. 출력 Y는 매 클록마다 하나의 클록 사이클 동안 HIGH 값이다. 이진 상태부호화와 one-hot 상태 부호화를 사용하여 회로를 설계하라.

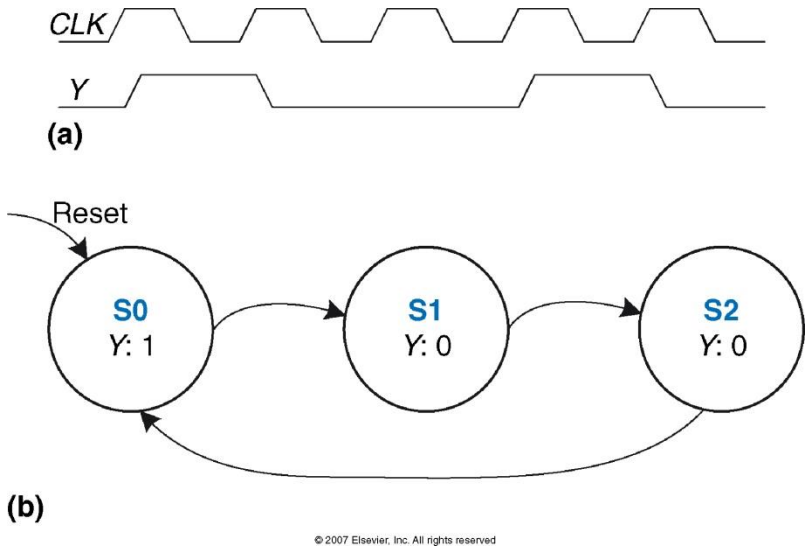




표 3.6 3진 카운터의 상태 변이표

| 현재 상태 | 다음 상태 |
|---|---|
| S0 | S1 |
| S1 | S2 |
| S2 | S0 |

표 3.7 3진 카운터의 출력표

| 현재 상태 | 출력 |
|---|---|
| S0 | 1 |
| S1 | 0 |
| S2 | 0 |

# 예제 **3.6**

표 3.8 divide-by-3 카운터를 위한 이진 부호화와 one-hot 부호화

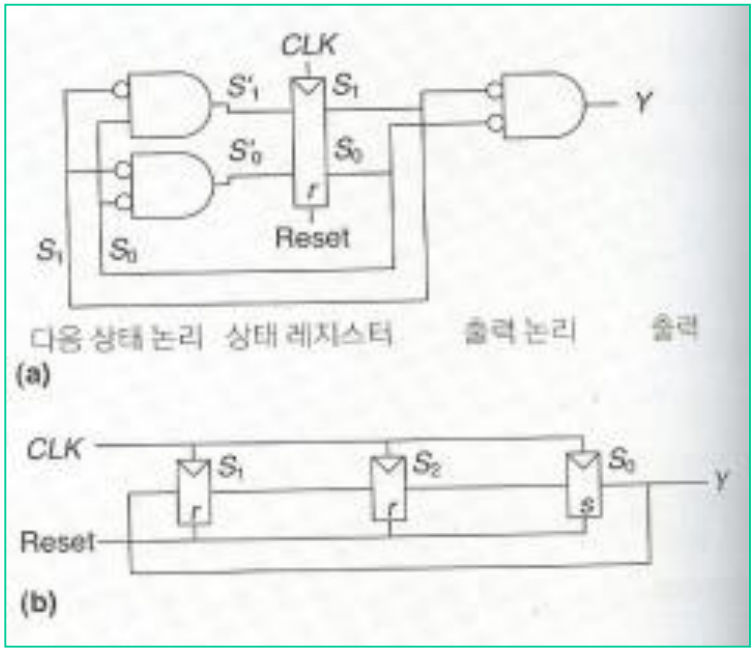
| State | One-Hot Encoding $S_2$ | $S_1$ | $S_0$ | Binary Encoding $S_1$ | $S_0$ |
|---|---|---|---|---|---|
| S0 | 0 | 0 | 1 | 0 | 0 |
| S1 | 0 | 1 | 0 | 0 | 1 |
| S2 | 1 | 0 | 0 | 1 | 0 |

표 3.9 이진 부호화를 가진 상태 변이표

| 현재 상태 $S_1$ | $S_0$ | 다음 상태 $S'_1$ | $S'_0$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |

표 3.10 one-hot 부호화를 가진 상태 변이표

| 현재 상태 $S_2$ | $S_1$ | $S_0$ | 다음 상태 $S'_2$ | $S'_1$ | $S'_0$ |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |



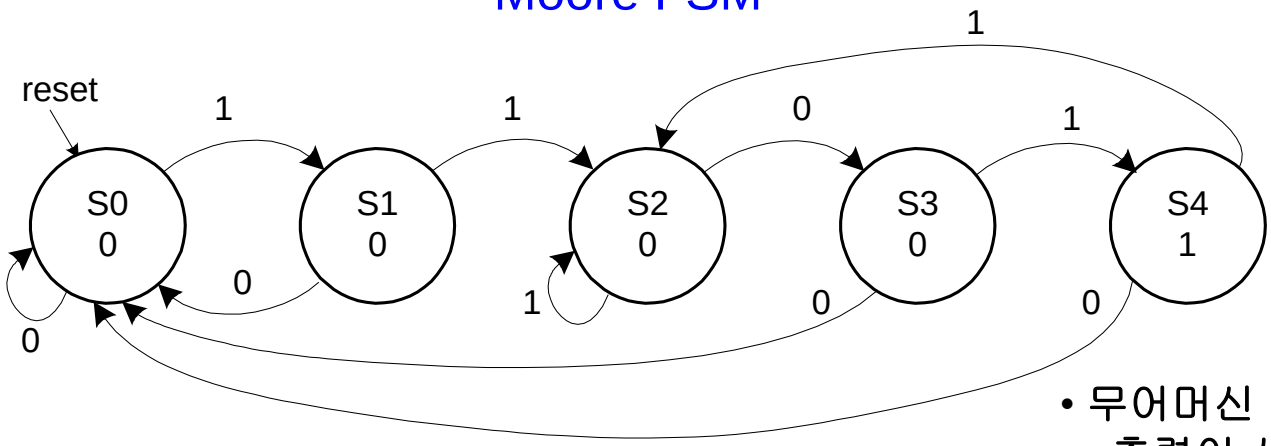

- $S'_2 = S_1$
- $S'_1 = S_0$
- $S'_0 = S_2$
- $Y'_1 = S_0$

# 3.4.3 Moore vs. Mealy FSM (last four 1101)

- Alyssa P. Hacker has a snail that crawls down a paper tape with 1's and 0's on it. The snail smiles whenever the last four digits it has crawled over are 1101. Design Moore and Mealy FSMs of the snail's brain.

# State Transition Diagrams (last four 1101)

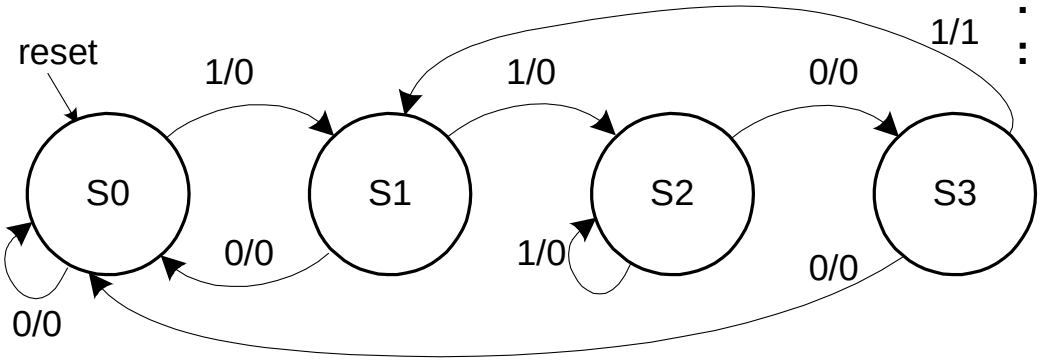
Moore FSM




- 무어머신
  : 출력이 시스템 상태에만 의존
  : 출력을 원 안에

Mealy FSM: arcs indicate input/output

Mealy FSM



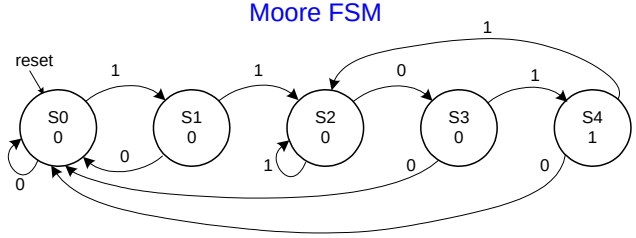

- 밀리머신
  : 출력이 현재상태 및 입력에 의존
  : 출력을 화살표에

# Moore FSM State Transition Table (last four 1101)

| Current State | | | Inputs | Next State | | |
|---|---|---|---|---|---|---|
| $S_2$ | $S_1$ | $S_0$ | $A$ | $Sn_2$ | $Sn_1$ | $Sn_0$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 |

| State | Encoding |
|---|---|
| S0 | 000 |
| S1 | 001 |
| S2 | 010 |
| S3 | 011 |
| S4 | 100 |

Moore FSM



# Moore FSM State Transition Table (last four 1101)

| Current State | | | Inputs | Next State | | |
|---|---|---|---|---|---|---|
| $S_2$ | $S_1$ | $S_0$ | $A$ | $Sn_2$ | $Sn_1$ | $Sn_0$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 |

| $S_0A$ / $S_2S_1$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | 1 | |
| 11 | X | X | X | X |
| 10 | | | X | X |

- $Sn_2 = S_1S_0A$
- $Sn_1 = \overline{S}_1S_0A + S_1\overline{S}_0 + S_2A$
- $Sn_0 = \overline{S}_2\overline{S}_1\overline{S}_0A + S_1\overline{S}_0\overline{A}$

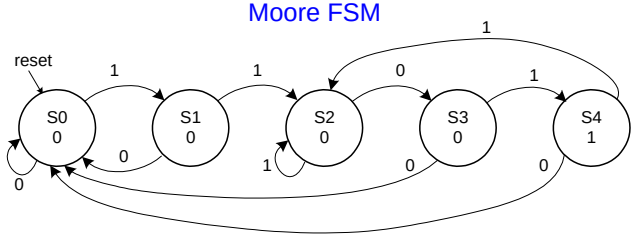
# Moore FSM Output Table (last four 1101)

| Current State | | | Output |
|---|---|---|---|
| $S_2$ | $S_1$ | $S_0$ | $Y$ |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |

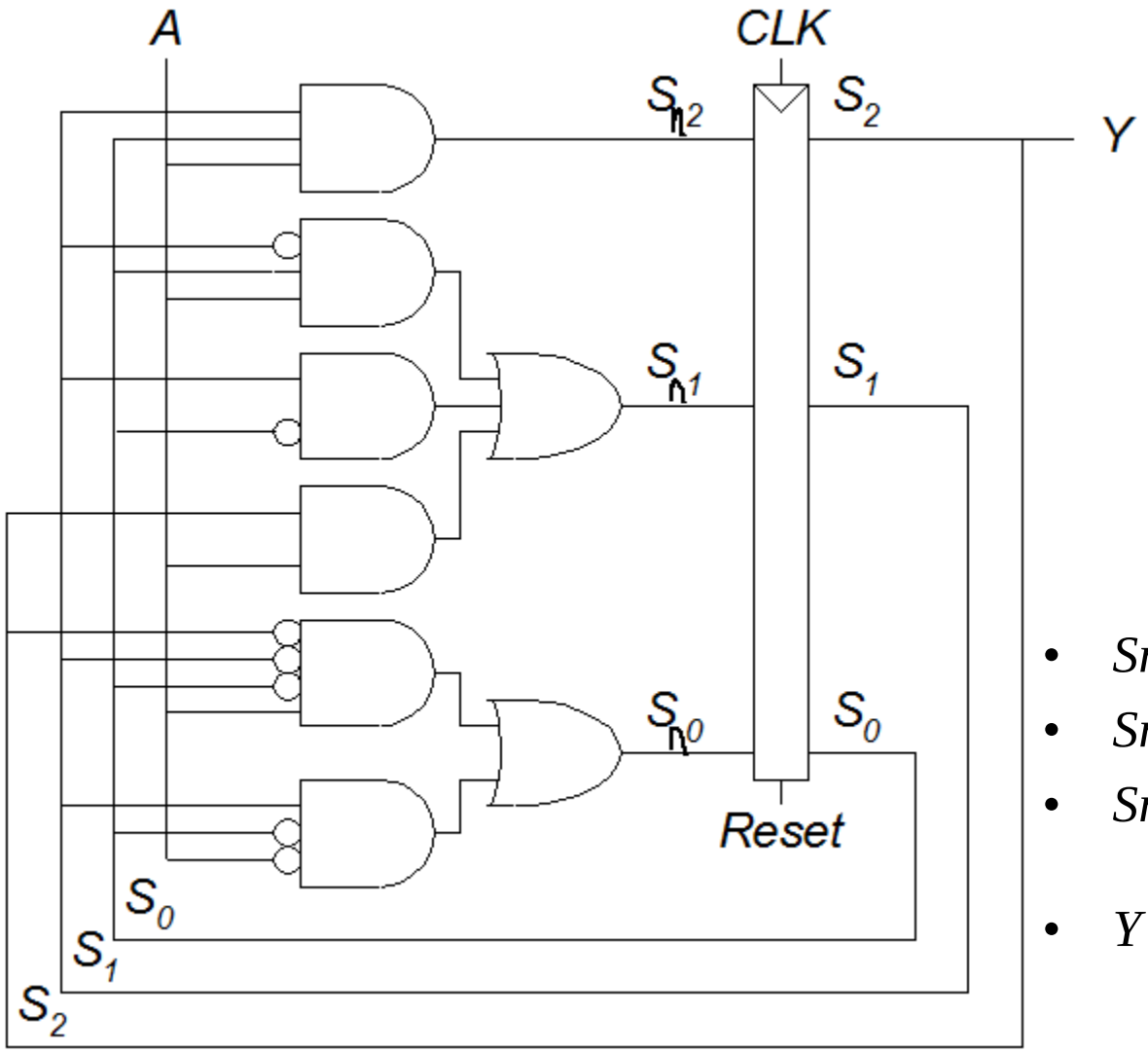
$$Y = S_2$$

| State | Encoding |
|---|---|
| S0 | 000 |
| S1 | 001 |
| S2 | 010 |
| S3 | 011 |
| S4 | 100 |

Moore FSM



# Moore FSM Schematic (last four 1101)



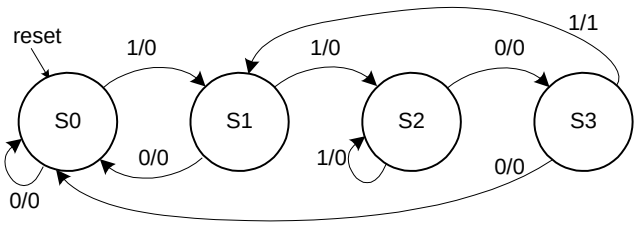

- $Sn_2 = S_1 S_0 A$
- $Sn_1 = \overline{S_1} S_0 A + S_1 \overline{S_0} + S_2 A$
- $Sn_0 = \overline{S_2}\,\overline{S_1}\,\overline{S_0} A + S_1 \overline{S_0}\,\overline{A}$
- $Y = S_2$

# Mealy FSM State Transition and Output Table (last four 1101)

| Current State | | Input | Next State | | Output |
|---|---|---|---|---|---|
| $S_1$ | $S_0$ | $A$ | $Sn_1$ | $Sn_0$ | $Y$ |
| 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 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |

| State | Encoding |
|---|---|
| S0 | 00 |
| S1 | 01 |
| S2 | 10 |
| S3 | 11 |

Mealy FSM



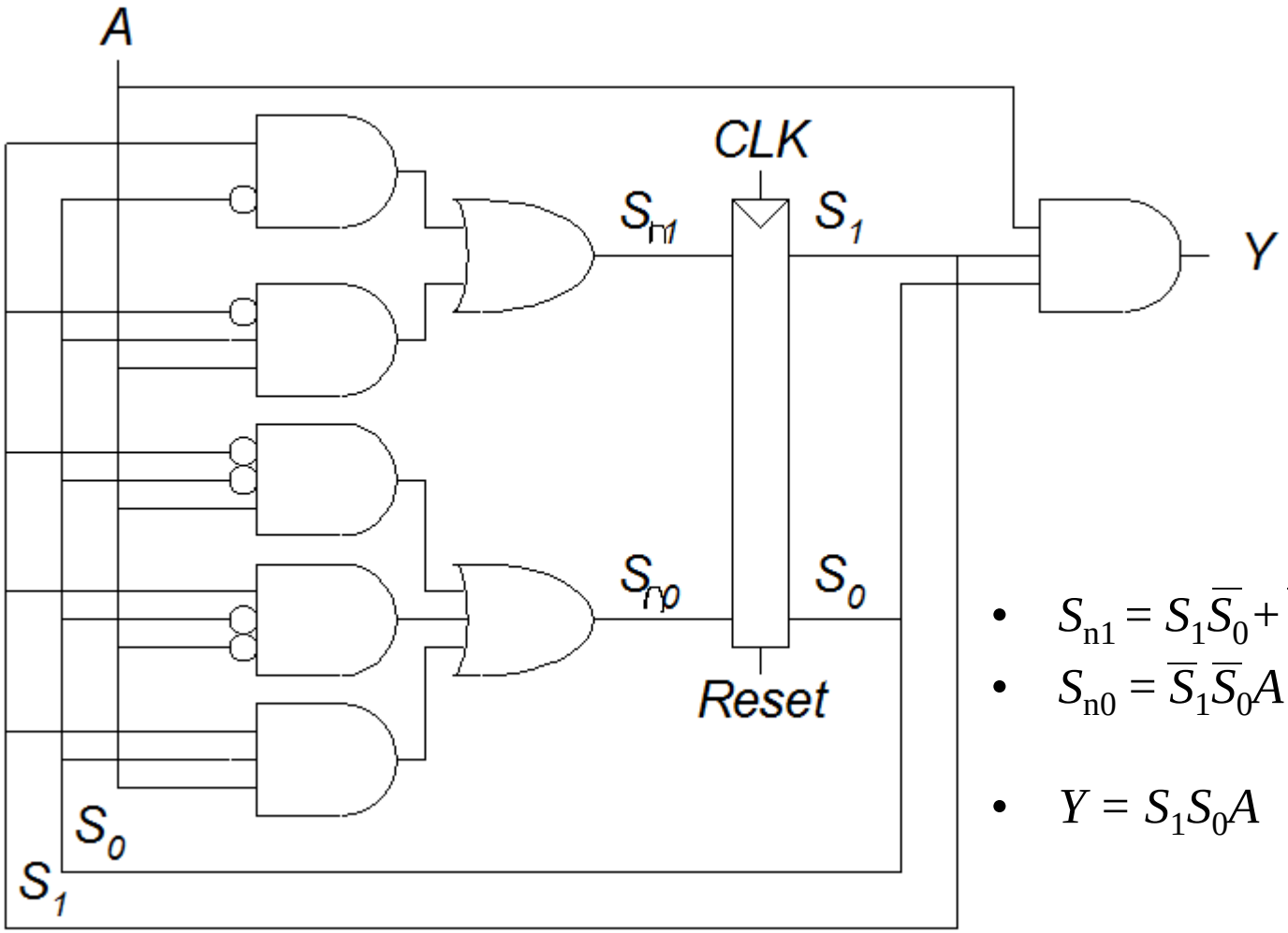
# Mealy FSM State Transition and Output Table (last four 1101)

| Current State | | Input | Next State | | Output |
|---|---|---|---|---|---|
| $S_1$ | $S_0$ | $A$ | $Sn_1$ | $Sn_0$ | $Y$ |
| 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 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |

| $S_1S_0$ \ A | 0 | 1 |
|---|---|---|
| 00 | | |
| 01 | | 1 |
| 11 | | |
| 10 | 1 | 1 |

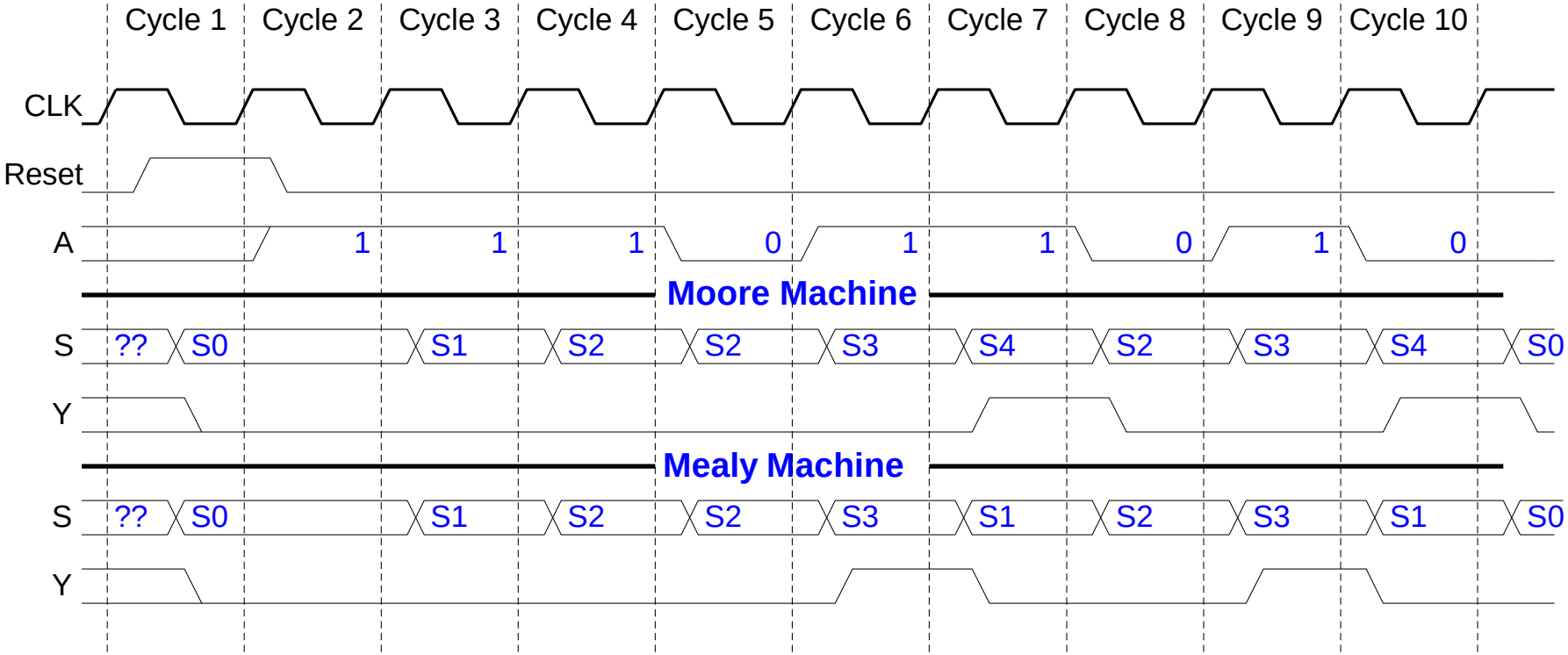
- $Sn_1 = S_1\overline{S}_0 + \overline{S}_1S_0A$
- $Sn_0 = \overline{S}_1\overline{S}_0A + S_1\overline{S}_0\overline{A} + S_1S_0A$
- $Y = S_1S_0A$

# Mealy FSM Schematic (last four 1101)




- $S_{n1} = S_1\overline{S_0} + \overline{S_1}S_0A$
- $S_{n0} = \overline{S_1}\,\overline{S_0}A + S_1\overline{S_0}\,\overline{A} + S_1S_0A$
- $Y = S_1S_0A$

# Moore and Mealy Timing Diagram (last four 1101)

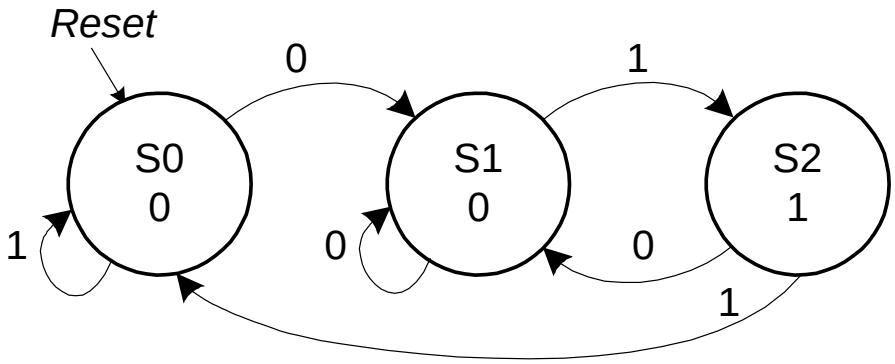


# 3.4.3 Moore vs. Mealy FSM (last two 01)

- Alyssa P. Hacker has a snail that crawls down a paper tape with 1's and 0's on it. The snail smiles whenever the last two digits it has crawled over are 01. Design Moore and Mealy FSMs of the snail's brain.

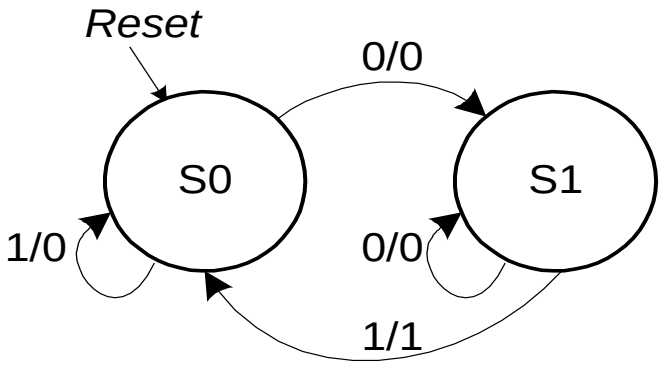
# State Transition Diagrams (last two 01)

## Moore FSM




Mealy FSM: arcs indicate input/output

## Mealy FSM



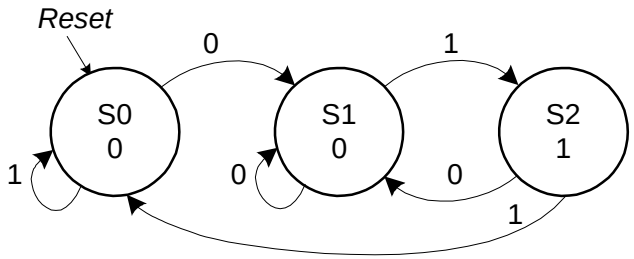
# Moore FSM State Transition Table (last two 01)

| Current State | | Inputs | Next State | |
|---|---|---|---|---|
| $S_1$ | $S_0$ | $A$ | $S'_1$ | $S'_0$ |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 |

| State | Encoding |
|---|---|
| S0 | 00 |
| S1 | 01 |
| S2 | 10 |

$$S_1' = S_0 A$$
$$S_0' = \overline{A}$$

**Moore FSM**

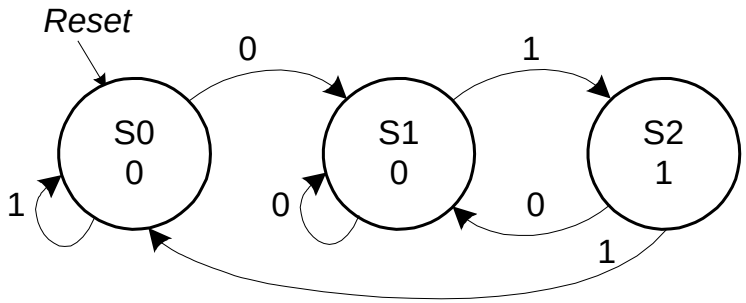


# Moore FSM Output Table (last two 01)

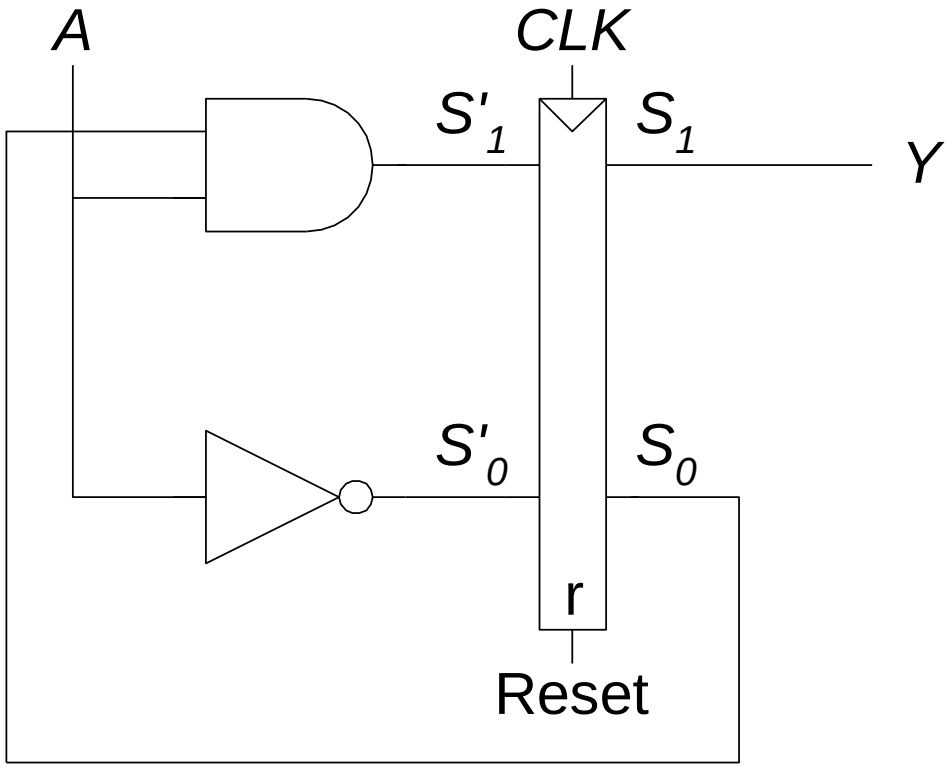
$$Y = S_1$$

| Current State | | Output |
|---|---|---|
| $S_1$ | $S_0$ | $Y$ |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

**Moore FSM**



# Moore FSM Schematic (last two 01)




$$S_1' = S_0 A$$

$$S_0' = \overline{A}$$

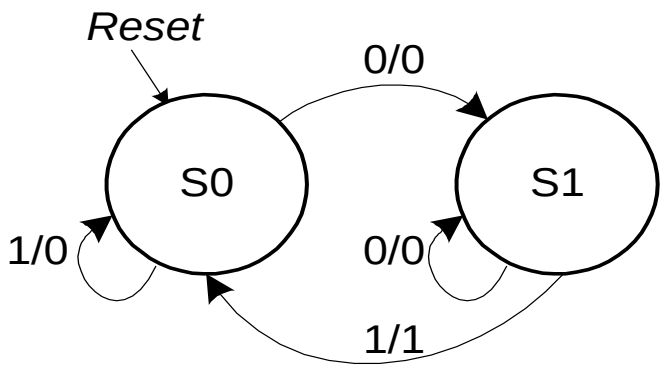
$$Y = S_1$$

# Mealy FSM State Transition and Output Table (last two 01)

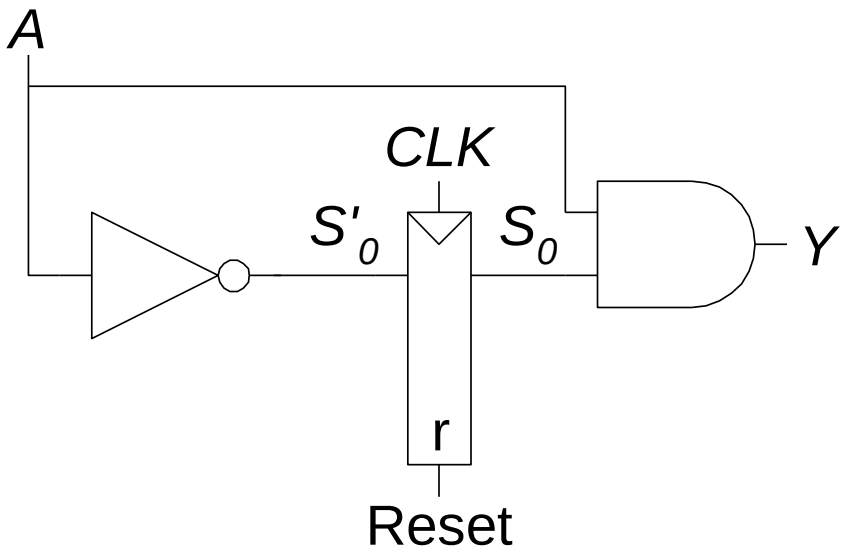
| Current State | Input | Next State | Output |
|---|---|---|---|
| $S_0$ | $A$ | $S'_0$ | $Y$ |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

| State | Encoding |
|---|---|
| S0 | 00 |
| S1 | 01 |

## Mealy FSM

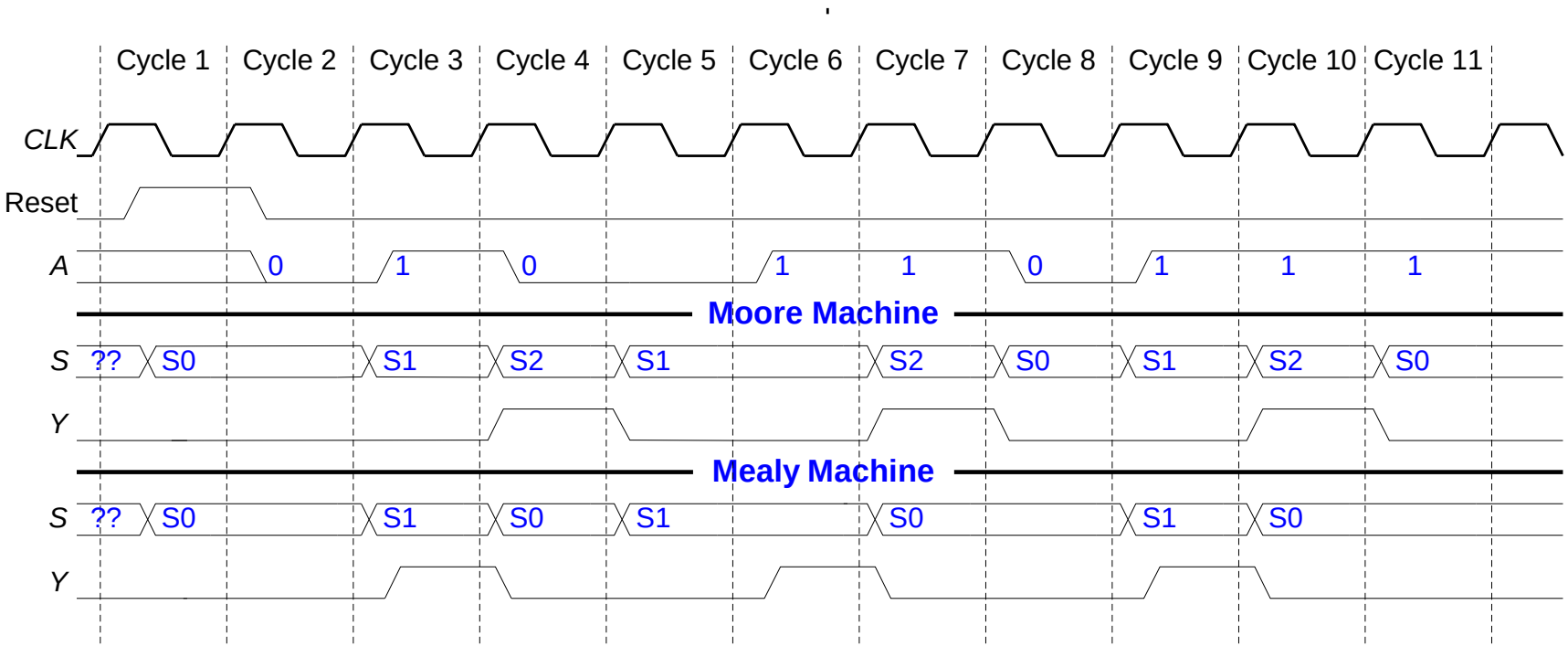


# Mealy FSM Schematic (last two 01)




- $S'_0 = \overline{A}$
- $Y = S_0 A$

# Moore and Mealy Timing Diagram (last two 01)



# 3.4.4 Factoring State Machines(상태머신 인수분해)

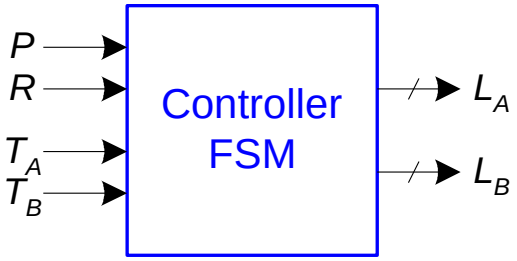
- Break complex FSMs into smaller interacting FSM
  (복잡한 FSM을 여러 개의 간단한 FSM으로 분류)

- Example
  : Modify the traffic light controller to have a Parade Mode.

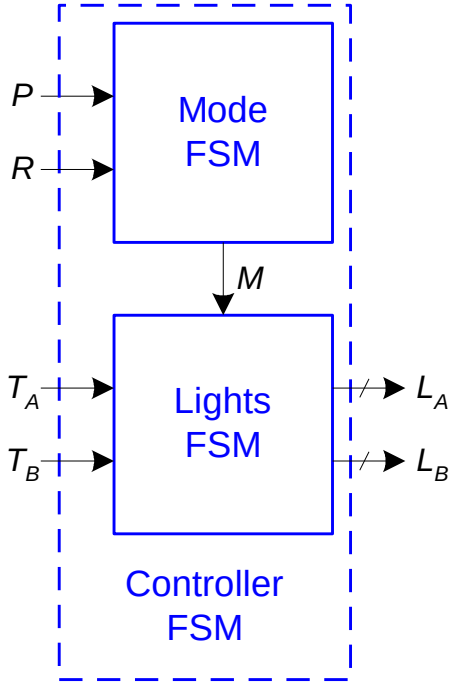
  – The FSM receives two more inputs: *P*, *R*
  – When *P* = 1, it enters Parade Mode and the Bravado Blvd. light stays green.
  – When *R* = 1, it leaves Parade Mode

# Parade FSM

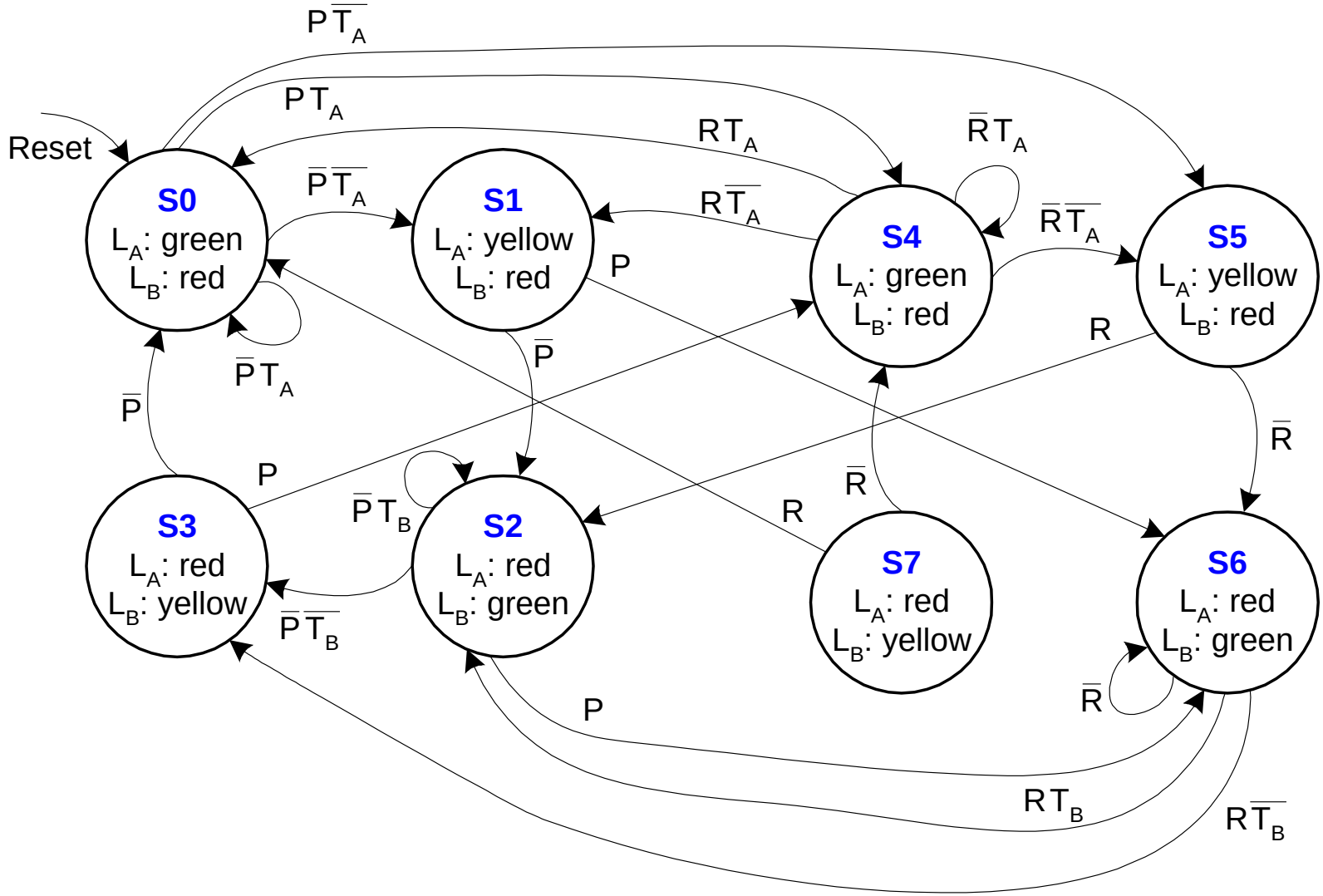
## Unfactored FSM



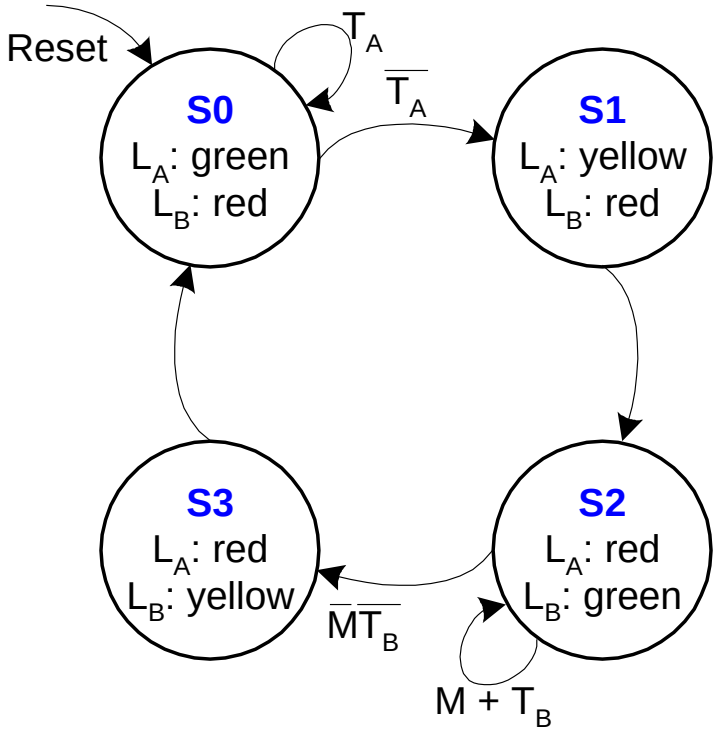

## Factored FSM



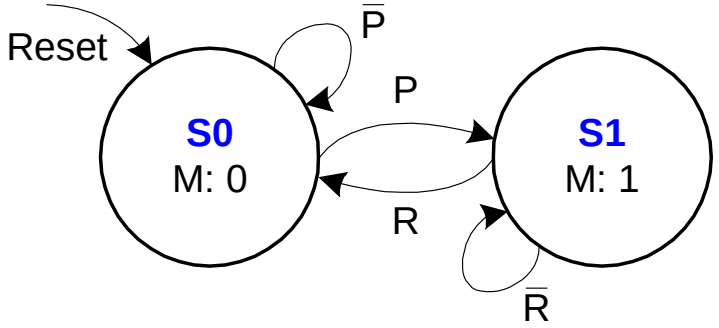
# Unfactored FSM State Transition Diagram



# Factored FSM State Transition Diagram




Lights FSM




Mode FSM

# 3.4.5 회로로부터 FSM 도출

- 실습시간에 분석

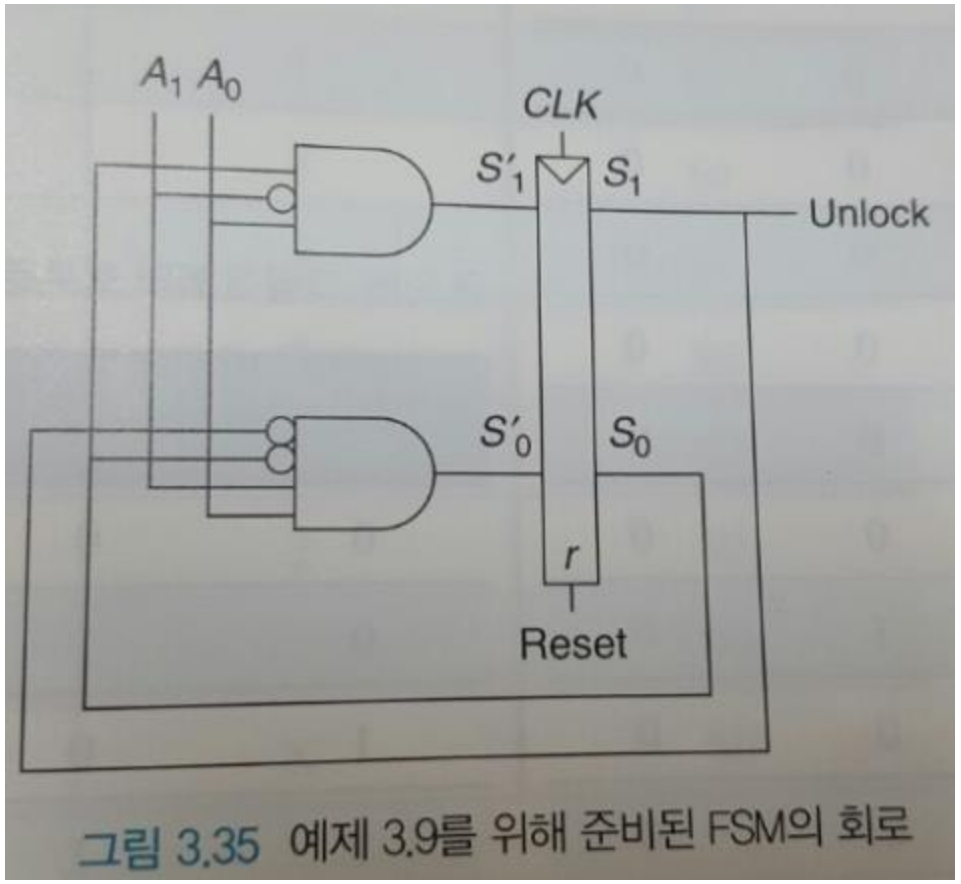



그림 3.35 예제 3.9를 위해 준비된 FSM의 회로

# Q & A