ECE 2050 Digital Logic and Systems

Chapter 9 :
Finite State Machine

Instructor: Yue ZHENG, Ph.D.

香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

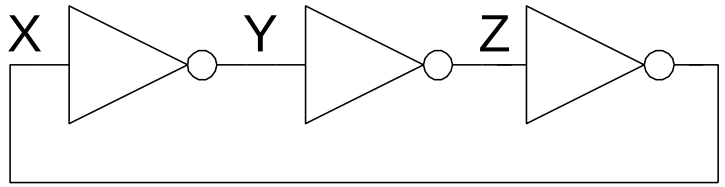
Last Week

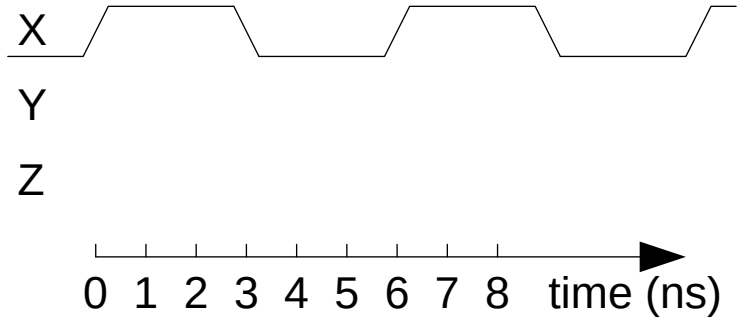
- Shift Registers
  - Data storage
  - Data movement
  - Serial/Parallel In - Serial/Parallel Out
- Bidirectional Shift Registers
- Shift Register Applications
  - Time Delay
  - Serial-to-Parallel Data Converter
  - UART (Universal Asynchronous Receiver Transmitter
  - Keyboard Encoder
- Johnson Counter & Ring Counter

Synchronous Sequential Design

Sequential Logic

- **Sequential circuits:** all circuits that aren't combinational
- **A problematic circuit:**





- **No inputs** and 1-3 outputs
- Astable circuit, **oscillates**
- Period depends on inverter delay
- It has a ***cyclic path***: output fed back to input

Synchronous Sequential Logic Design

- Breaks cyclic paths by **inserting registers**
- Registers contain **state** of the system
- State changes at clock edge: system **synchronized** to the clock

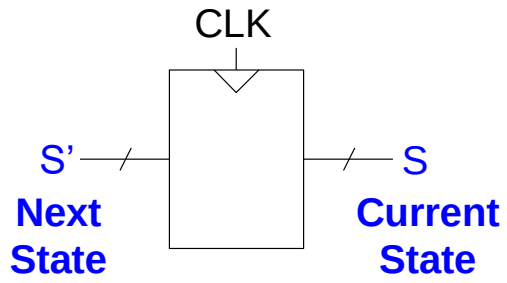
- **Rules** of synchronous sequential circuit composition:
  - Every circuit element is either a **register** or a **combinational circuit**
  - At least **one** circuit element is a **register**
  - All registers receive the **same clock**
  - Every **cyclic path** contains at least **one register**

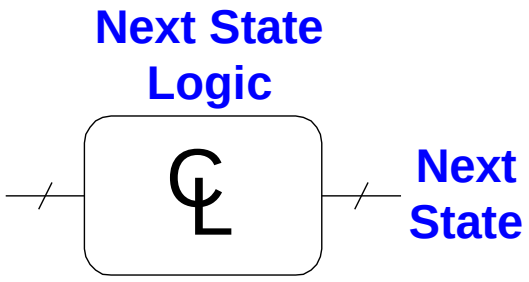
- **Finite State Machine (FSMs)** are common synchronous sequential circuits

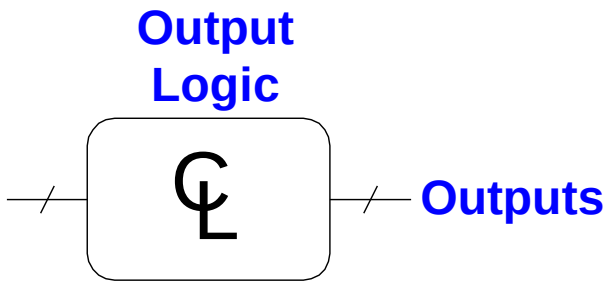
Finite State Machines

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



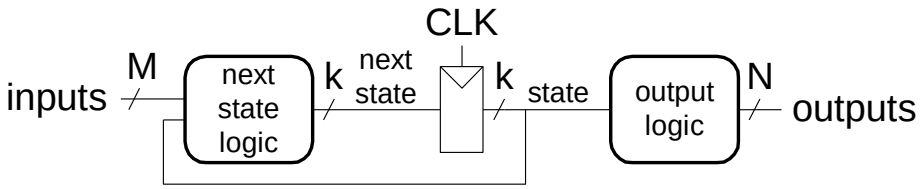




Finite State Machine (FSM)

- **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**

Moore FSM



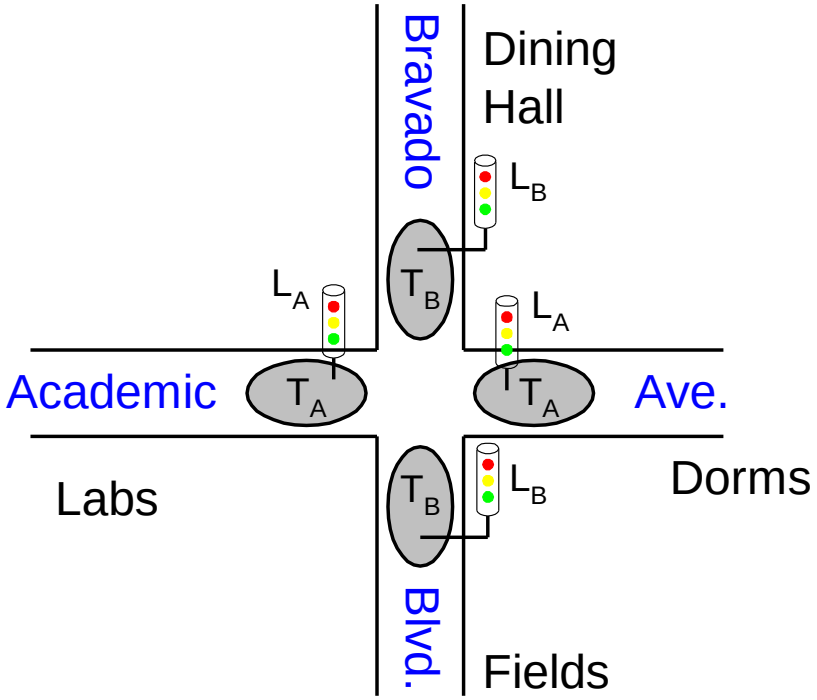
FSM Design Procedure

1. Identify **inputs** and **outputs**
2. Sketch **state transition diagram**
3. Write **state transition table** and **output table**
   - **Moore FSM:** write **separate** tables
   - **Mealy FSM:** write **combined** state transition and output table
4. Select **state encodings**
5. Rewrite state transition table and output table with state **encodings**
6. Write **Boolean equations** for next state and output logic
7. Sketch the circuit **schematic**

Moore FSMs Examples

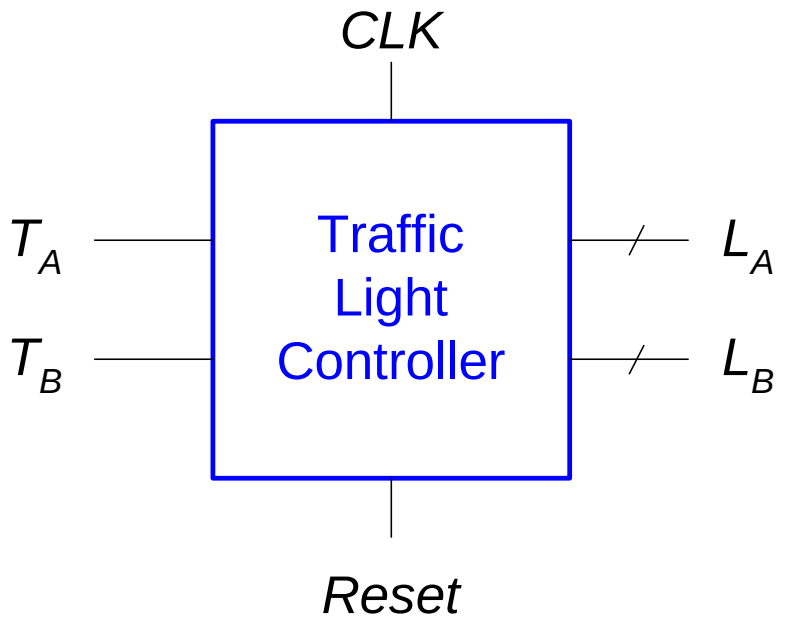
FSM Example

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



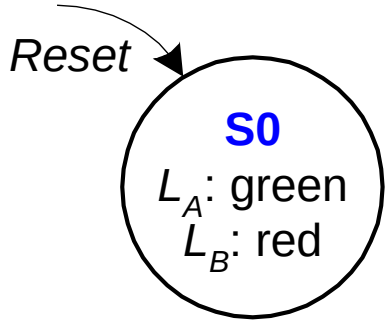
FSM Block Box

- **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

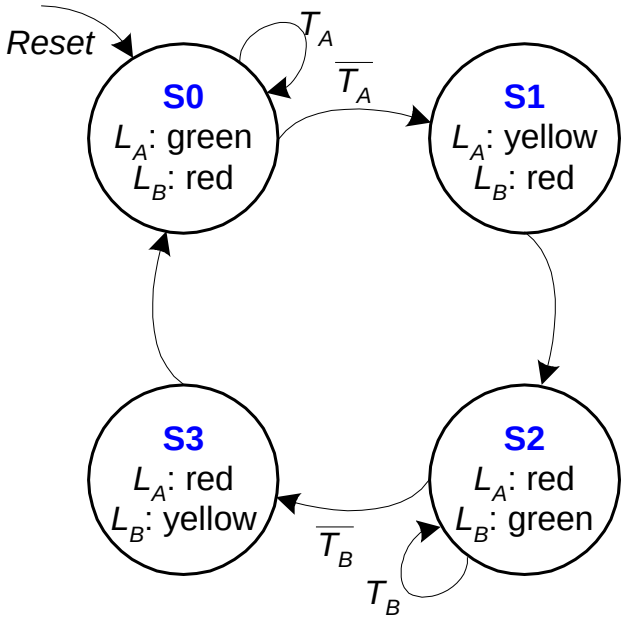


FSM State Transition Table

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

S : Current State

S': Next State



FSM Encoded State Transition Table

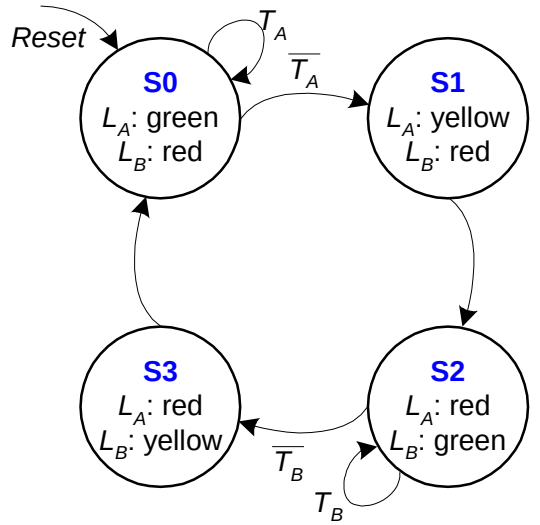
Current State		Inputs		Next State	
S_1	S_0	T_A	T_B	S'_1	S'_0
S0		0	X	S1	
S0		1	X	S0	
S1		X	X	S2	
S2		X	0	S3	
S2		X	1	S2	
S3		X	X	S0	

State	Encoding
S0	00
S1	01
S2	10
S3	11

FSM Output Table

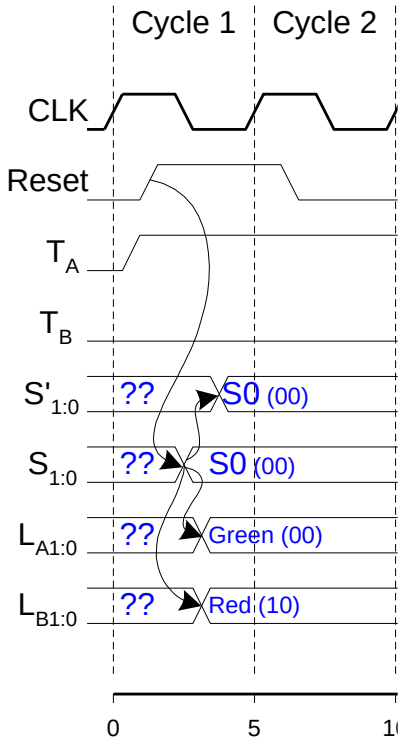
Current State		Outputs			
S_1	S_0	L_{A1}	L_{A0}	L_{B1}	L_{B0}
S0		green		red	
S1		yellow		red	
S2		red		green	
S3		red		yellow	

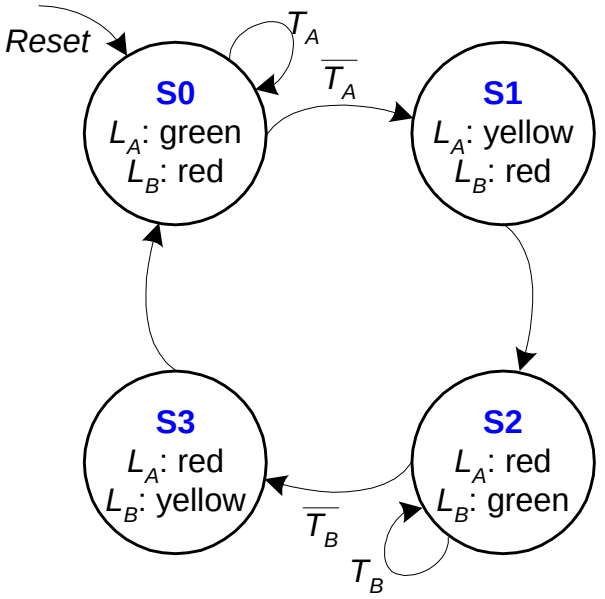
Output	Encoding
green	00
yellow	01
red	10



FSM Schematic

FSM Timing Diagram





State Encodings

- **Binary** encoding:
  - i.e., for four states, 00, 01, 10, 11
- **One-hot** encoding
  - One state bit per state
  - Only one state bit HIGH at once
  - i.e., for 4 states, 0001, 0010, 0100, 1000
  - Requires more flip-flops
  - Often next state and output logic is simpler

1-Hot State Encoding Example

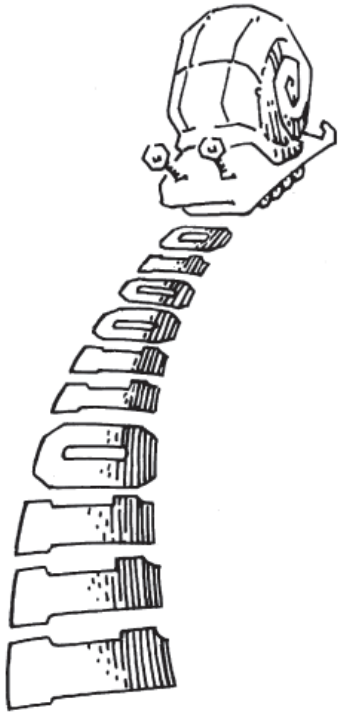
Current State				Inputs		Next State			
S_3	S_2	S_1	S_0	T_A	T_B	S'_3	S'_2	S'_1	S'_0
S0				0	X	S1			
S0				1	X	S0			
S1				X	X	S2			
S2				X	0	S3			
S2				X	1	S2			
S3				X	X	S0			

State	1-Hot Encoding
S0	0001
S1	0010
S2	0100
S3	1000

Mealy FSM Examples

Moore vs. Mealy FSMs

- 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

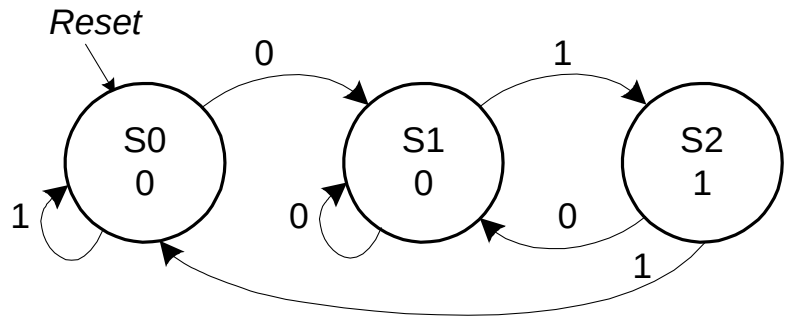
Moore FSM

Mealy FSM

Moore FSM State Transition Table

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

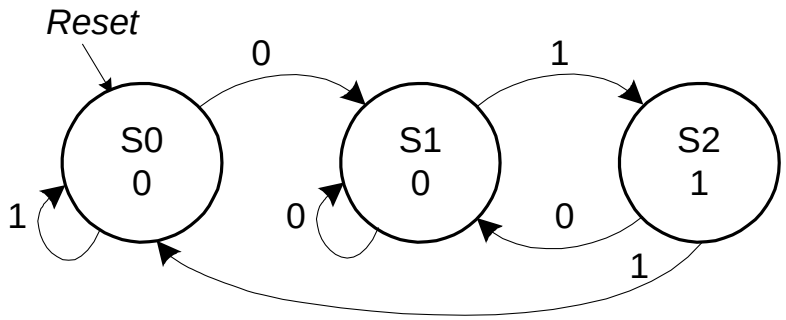
State	Encoding
S0	00
S1	01
S2	10



Moore FSM Output Table

Current State		Output
S_1	S_0	Y
0	0	
0	1	
1	0	

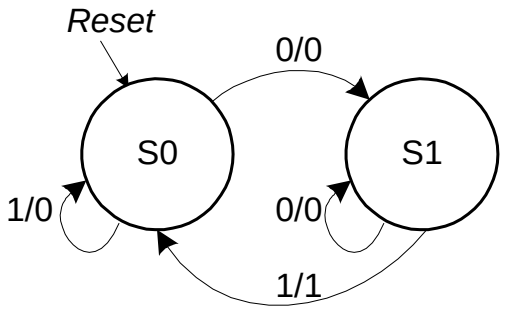
State	Encoding
S0	00
S1	01
S2	10



Mealy State Transition & Output Table

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

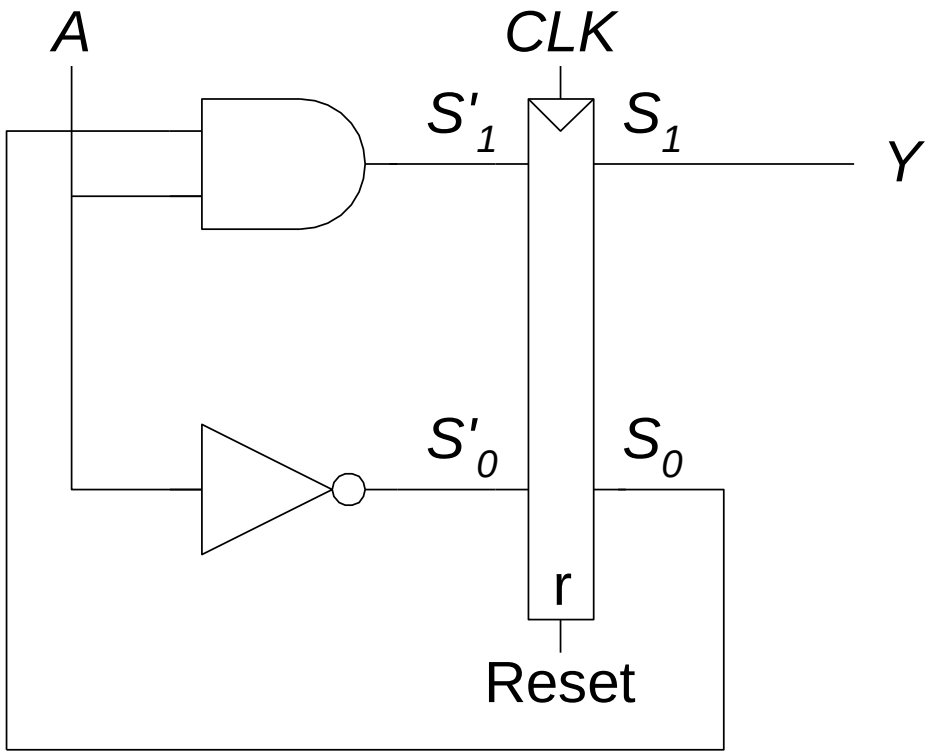
State	Encoding
S0	0
S1	1



Moore FSM Schematic

- **Next State Equations**
- $S_1' = S_0 A$
- $S_0' = A$

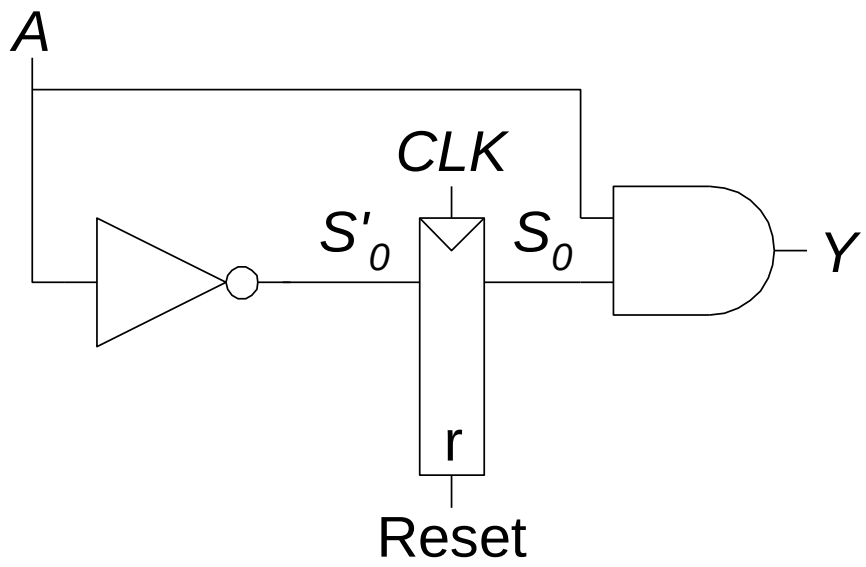
- **Output Equation**
- $Y = S_1$



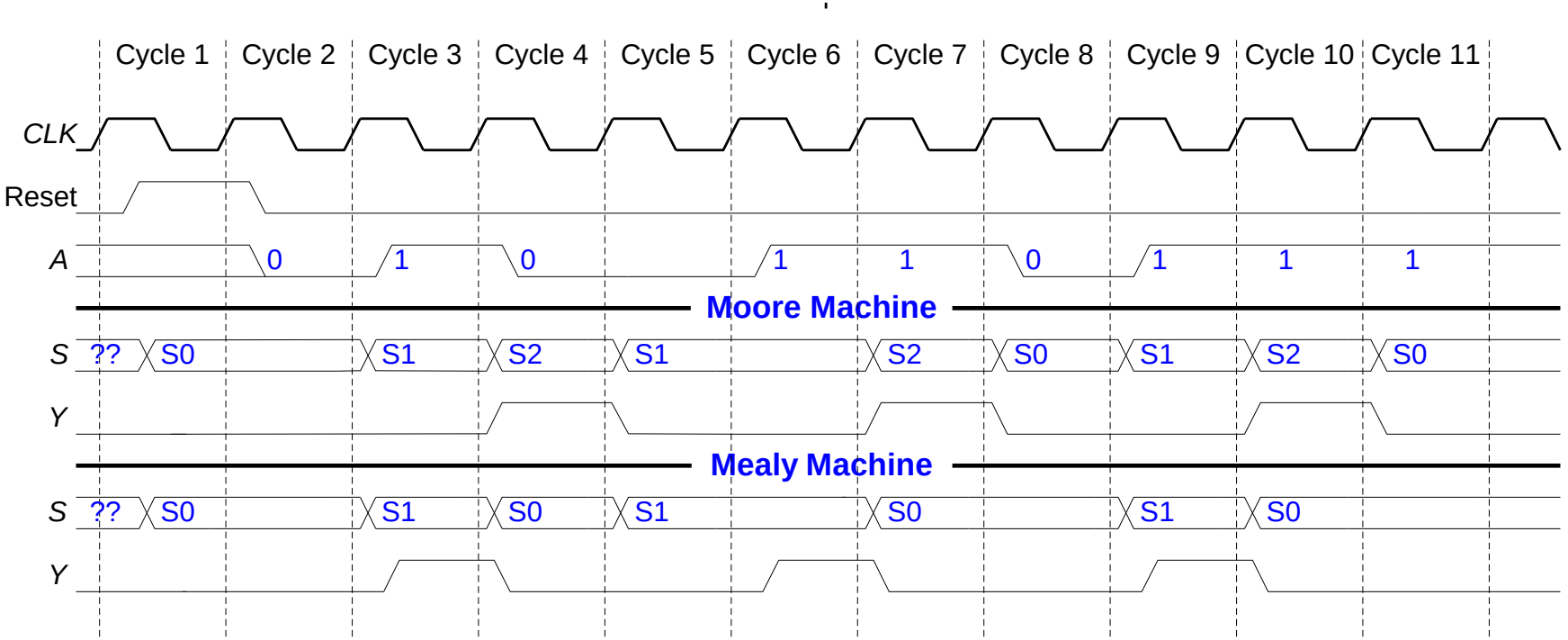
Mealy FSM Schematic

- **Next State Equation**
- $S_0' = A$

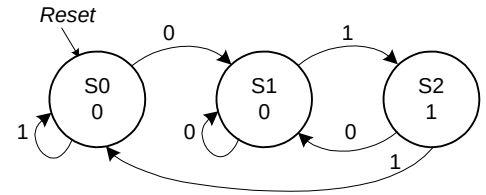
- **Output Equation**
- $Y = S_0 A$



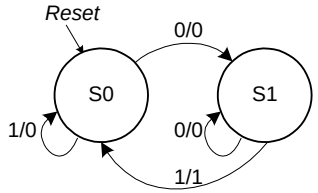
Moore and Mealy Timing Diagram



Moore FSM



Mealy FSM



Mealy FSM: asserts Y **immediately** when input pattern 01 is detected

Moore FSM: asserts Y one cycle **after** input pattern 01 is detected

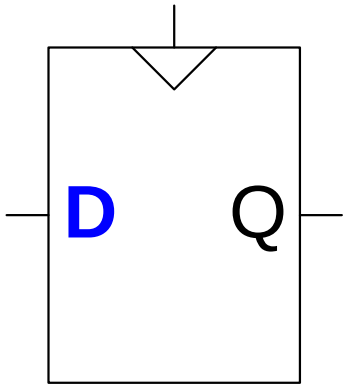
Timing

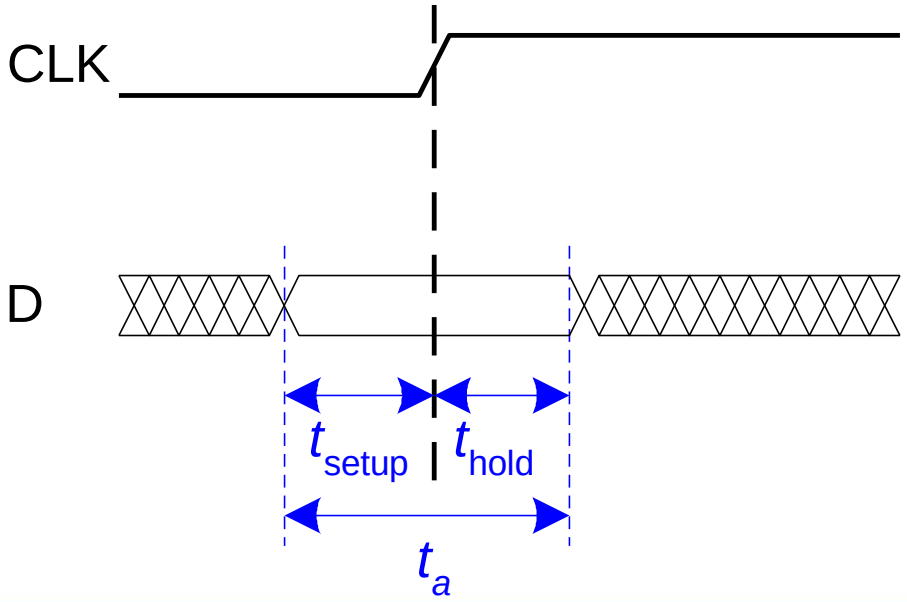
Timing

- Flip-flop samples *D* at clock edge
- ***D* must be stable when sampled**
- Similar to a photograph, *D* must be stable around clock edge

Input Timing Constraints

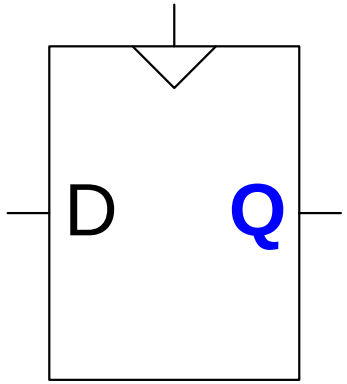
- **Setup time:** $\boldsymbol{t_{setup}}$ = time *before* clock edge data must be stable (i.e. not changing)
- **Hold time:** $\boldsymbol{t_{hold}}$ = time *after* clock edge data must be stable
- **Aperture time:** $\boldsymbol{t_a}$ = time *around* clock edge data must be stable ($t_a = t_{setup} + t_{hold}$)

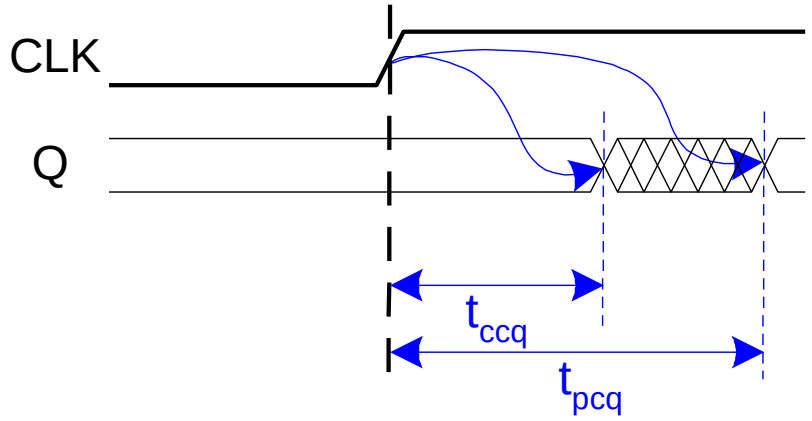




Output Timing Constraints

- **Propagation delay: t_{pcq}** = time after clock edge that *Q* is guaranteed to be stable (i.e., to stop changing): **maximum delay**
- **Contamination delay: t_{ccq}** = time after clock edge that *Q* might be unstable (i.e., start changing): **minimum delay**



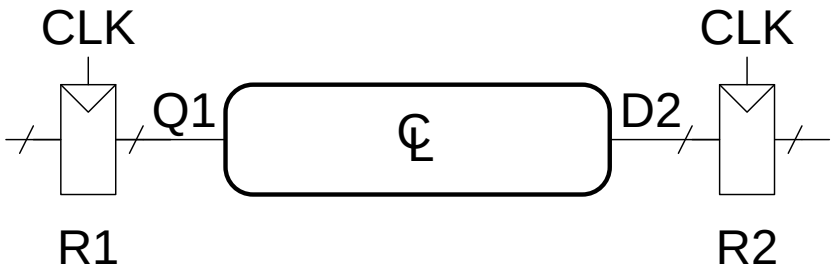


Dynamic Discipline

- Synchronous sequential circuit inputs must be **stable during aperture** (setup and hold) time around clock edge
- Specifically, inputs must be stable
  - at least t_{setup} before the clock edge
  - at least until t_{hold} after the clock edge

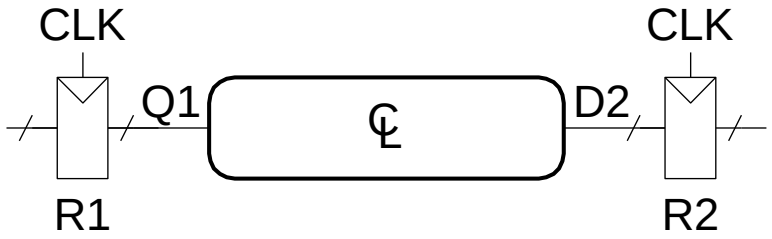
Dynamic Discipline

- The delay between registers has a **minimum** and **maximum** delay, dependent on the delays of the circuit elements



Setup Time Constraint

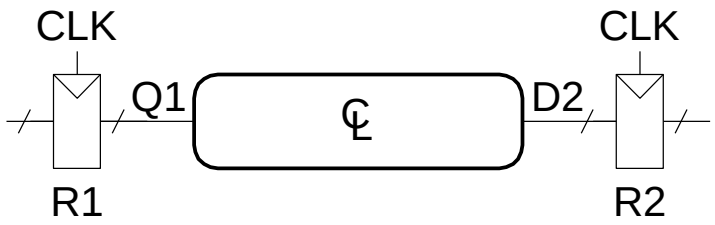
- Depends on the **maximum** delay from register R1 through combinational logic to R2
- The input to register R2 must be stable at least t_{setup} before clock edge



Also called:
Cycle Time Constraint

Hold Time Constraint

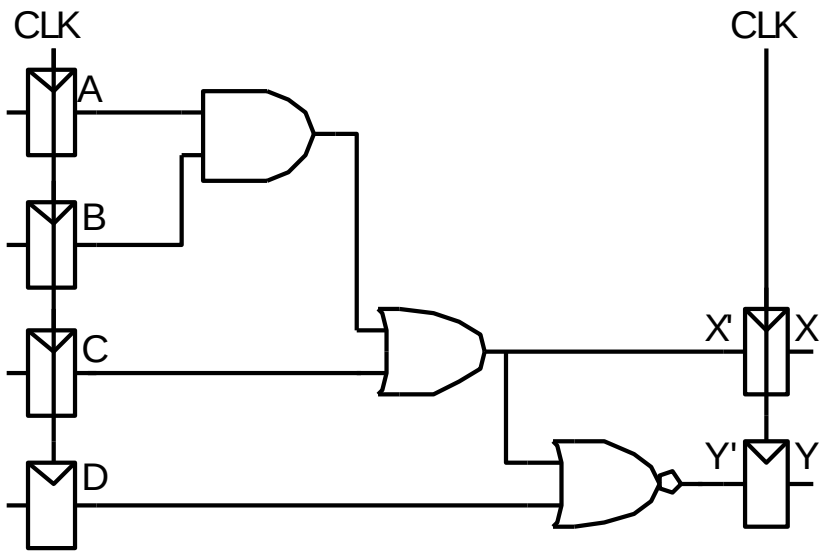
- Depends on the **minimum** delay from register R1 through the combinational logic to R2
- The input to register R2 must be stable for at least t_{hold} after the clock edge



Timing Analysis

- Calculate **both constraints**:
  - **Setup time** constraint (aka cycle time constraint)
  - **Hold time** constraint
- If the hold time constraint isn't met, the circuit **won't work reliably at any frequency**

Timing Analysis Example



Timing Characteristics

Flip-Flops

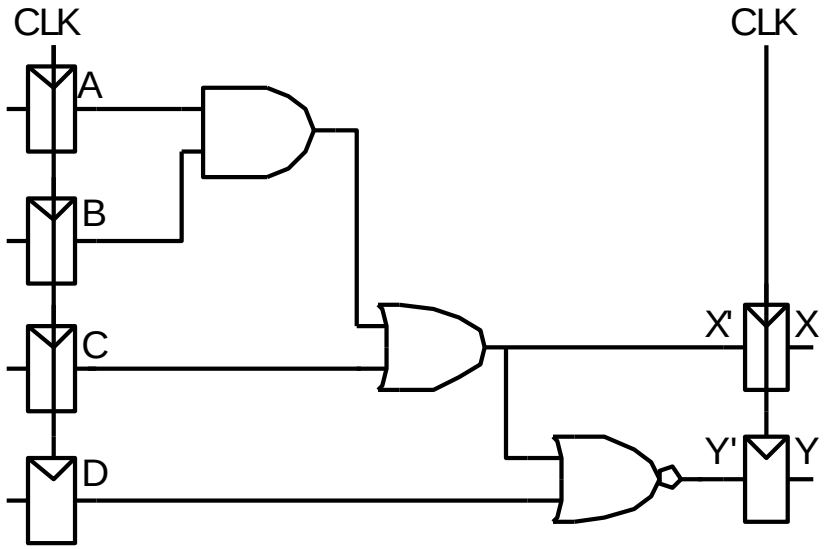
- t_{ccq} = 30 ps
- t_{pcq} = 50 ps
- t_{setup} = 60 ps
- t_{hold} = 70 ps

Logic delays: per gate

- t_{pd} = 35 ps
- t_{cd} = 25 ps

Timing Analysis Example

How to fix?



Timing Characteristics

Flip-Flops:
- t_{ccq} = 30 ps
- t_{pcq} = 50 ps
- t_{setup} = 60 ps
- t_{hold} = 70 ps

Logic delays: per gate:
- t_{pd} = 35 ps
- t_{cd} = 25 ps

t_{pd} = 3 x 35 ps = 105 ps

t_{cd} = 25 ps

Setup time constraint:

Hold time constraint:

Chapter Review

- (Synchronous) Sequential Logic Counters
- Finite State Machine
  - Moore FSM
  - Mealy FSM
- Timing
  - Setup time
  - Hold time
  - Aperture time
  - Propagation delay
  - Contamination delay
  - Dynamic Discipline

True/False Quiz

- ✓ A state machine is a sequential circuit having a limited number of states occurring in a prescribed order.
- ✓ FSM consists of a state register and combinational logic.
- ✓ For a synchronous sequential circuit, all registers receive the same clock.
- ✗ 2-bit is required to encode 4 states by using one-hot encoding.
- ✓ For the Moore FSM, outputs depend only on current state.
- ✓ The setup time is the minimum amount of time that the input signal must be stable before the active clock edge to ensure that the data is correctly captured by the register.
- ✓ The hold time is the minimum amount of time that the input signal must remain stable after the active clock edge to ensure that the data is reliably captured by the flip-flop or register.