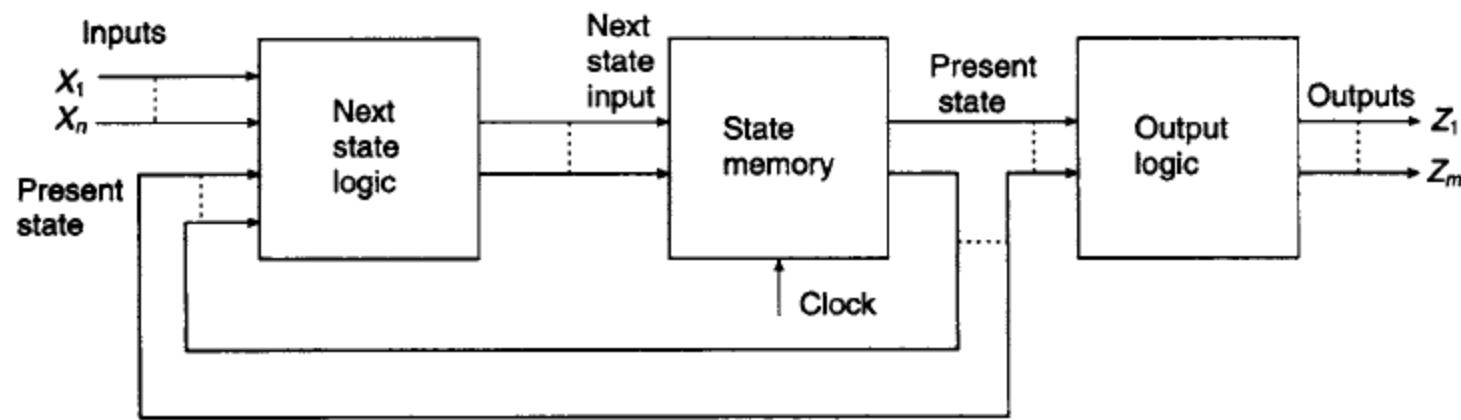
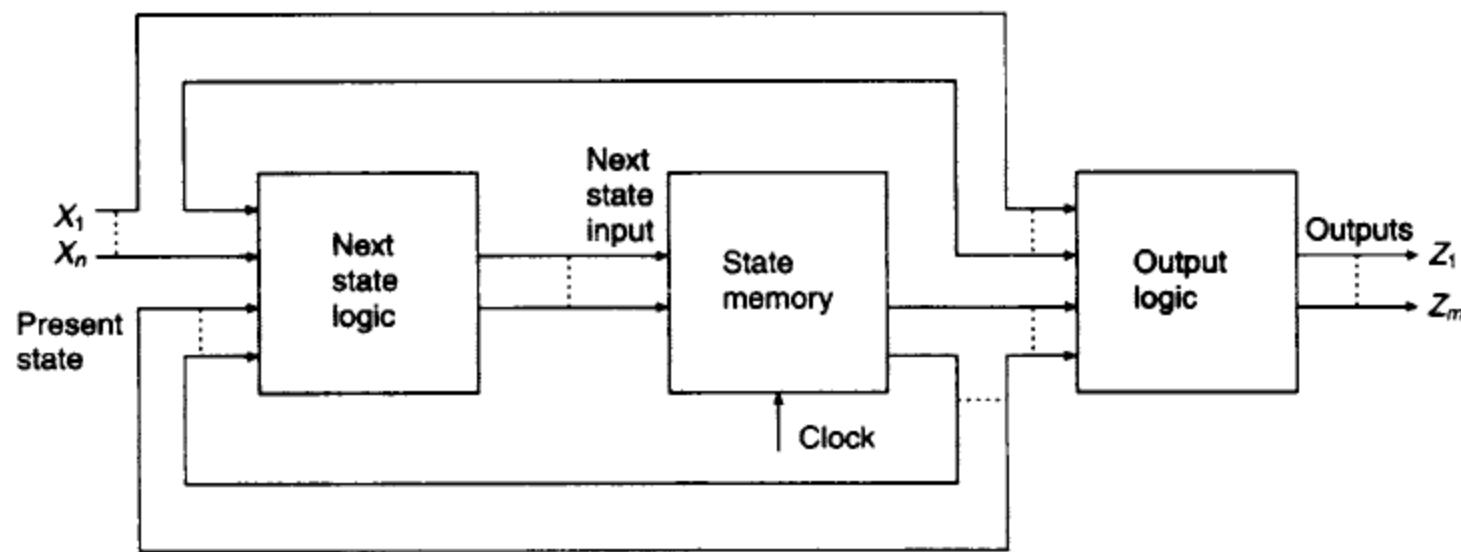


# Finite-State Machines (FSMs) and Controllers



(a)

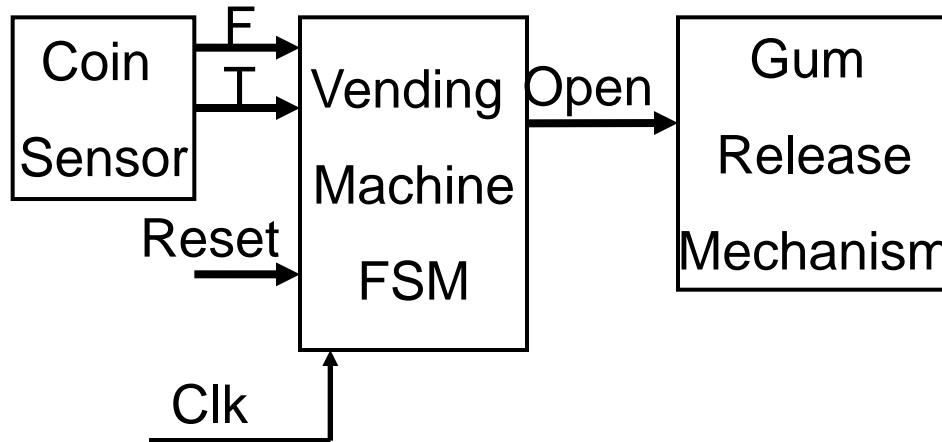


(b)

# FSM State Encoding

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

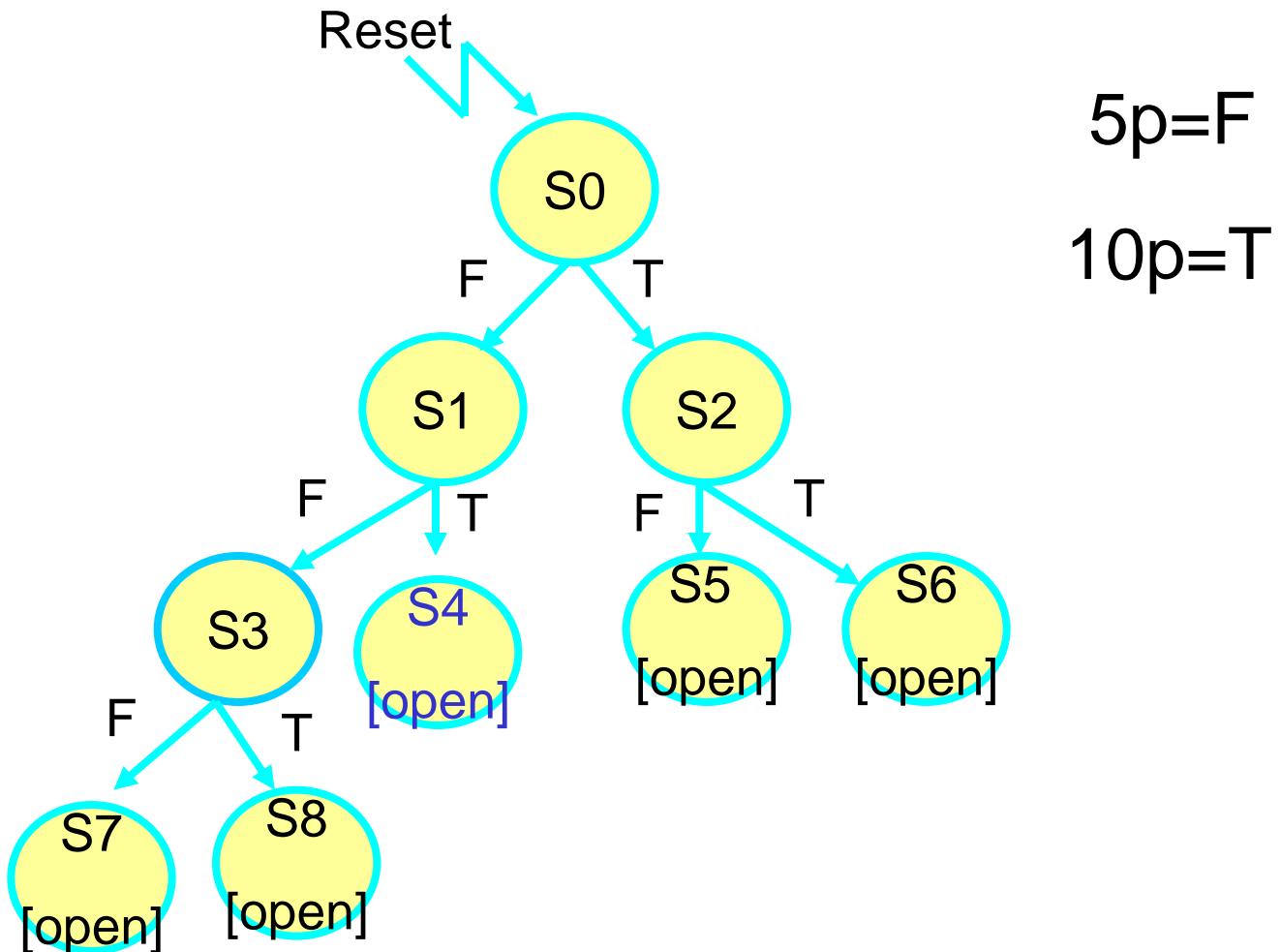
# Vending machine block diagram



$$\begin{aligned} 5p &= F \\ 10p &= T \end{aligned}$$

- Three 5p coins in sequence: F,F,F
- Two 5p coins followed by a 10p: F,F,T
- A 5p coin followed by a 10p: F,T
- A 10p coin followed by a 5p: T,F
- Two 10p coins in sequence: T,T

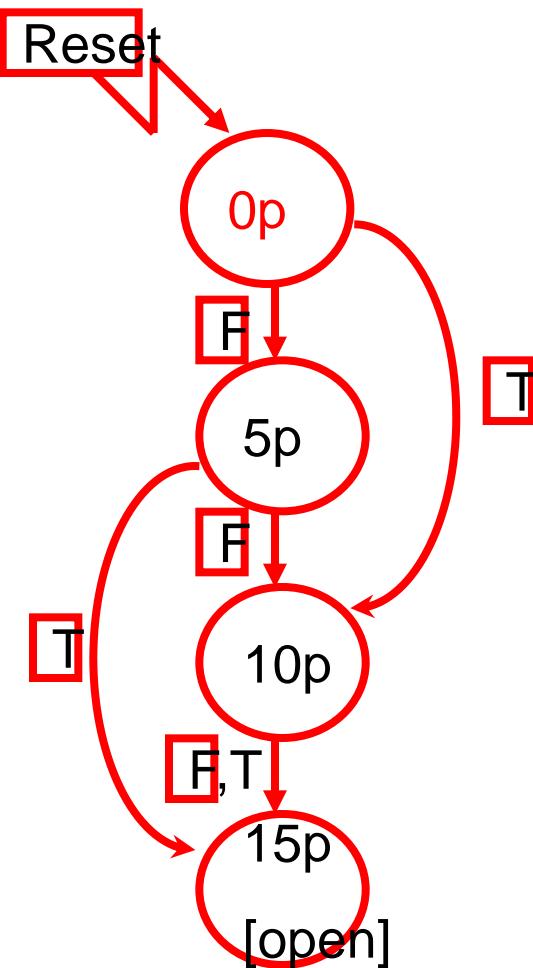
# State diagram



$$5p=F$$

$$10p=T$$

# Minimized state diagram



$5p=F$

$10p=T$

# Minimized transition table

Present state	Inputs		Next state	Output Open
	T	F		
0p	0	0	0p	0
	0	1	5p	0
	1	0	10p	0
	1	1	X	X
5p	0	0	5p	0
	0	1	10p	0
	1	0	15p	0
	1	1	X	X
10p	0	0	10p	0
	0	1	15p	0
	1	0	15p	0
	1	1	X	X
15p	X	X	15p	1

5p=F

10p=T

# Encoded transition table

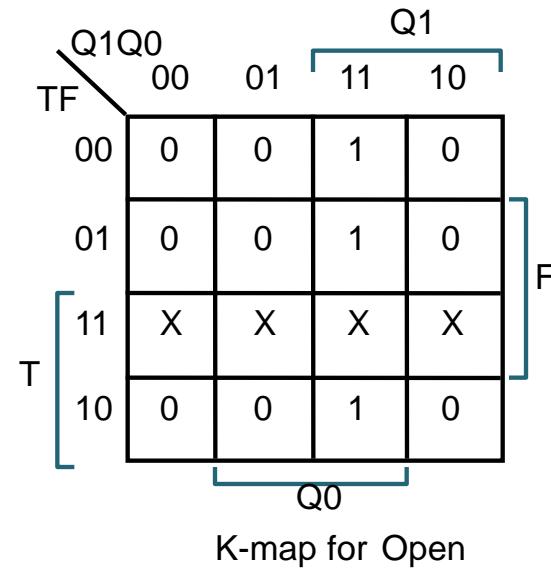
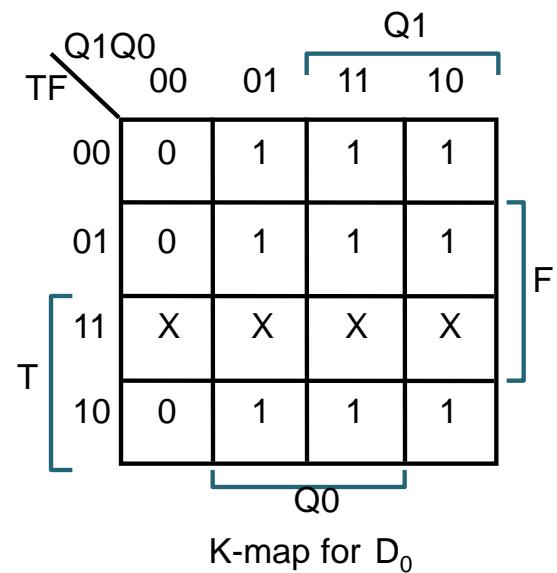
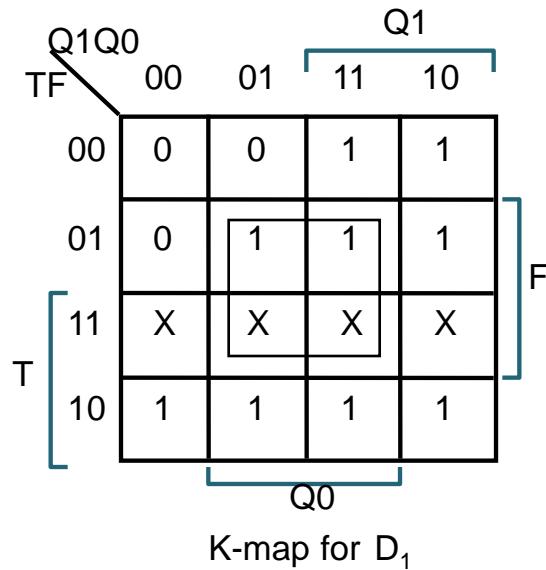
Present state		Inputs		Next state		Output Open
		T	F			
Q <sub>1</sub>	Q <sub>0</sub>	D <sub>1</sub>	D <sub>0</sub>	0	0	0
0	0	0	0			0
		0	1			0
		1	0			0
		1	1			X
0	1	0	0	0	1	0
		0	1	1	0	0
		1	0	1	1	0
		1	1	X	X	X
1	0	0	0	1	0	0
		0	1	1	1	0
		1	0	1	1	0
		1	1	X	X	X
1	1	0	0	1	1	1
		0	1	1	1	1
		1	0	1	1	1
		1	1	X	X	X

5p=F

10p=T

# Vending Machine K Maps for D flip flops implementation

## Design Example (Continued)



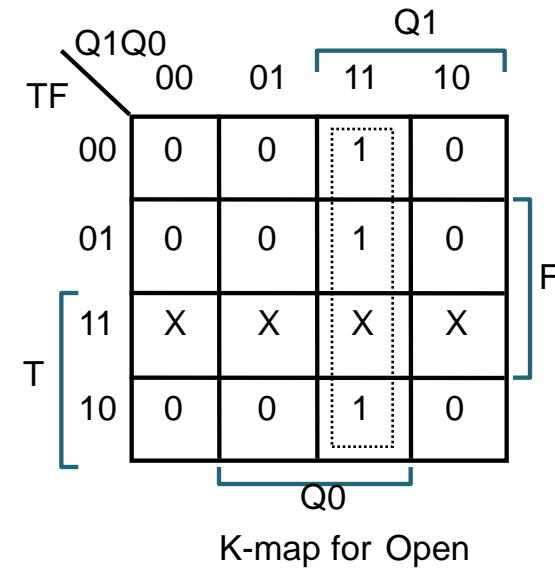
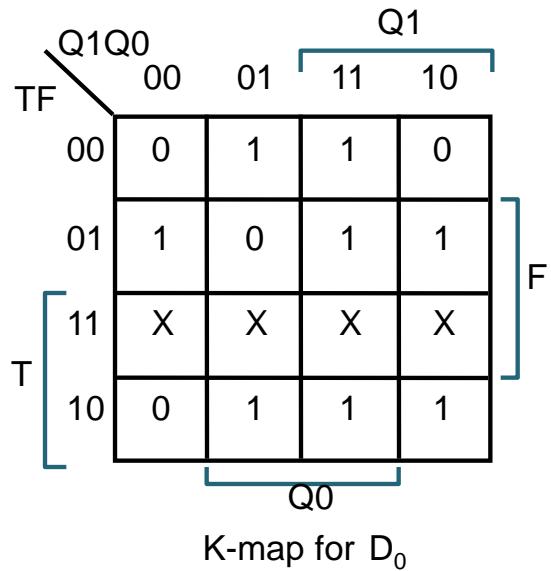
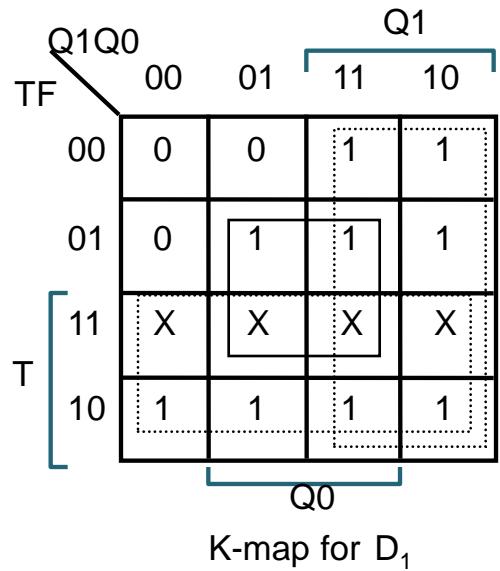
$$D_1 = ?$$

$$D_0 = ?$$

$$\text{Open} = ?$$

# Vending Machine- K Maps for D flip flops implementation

## Design Example (Continued)

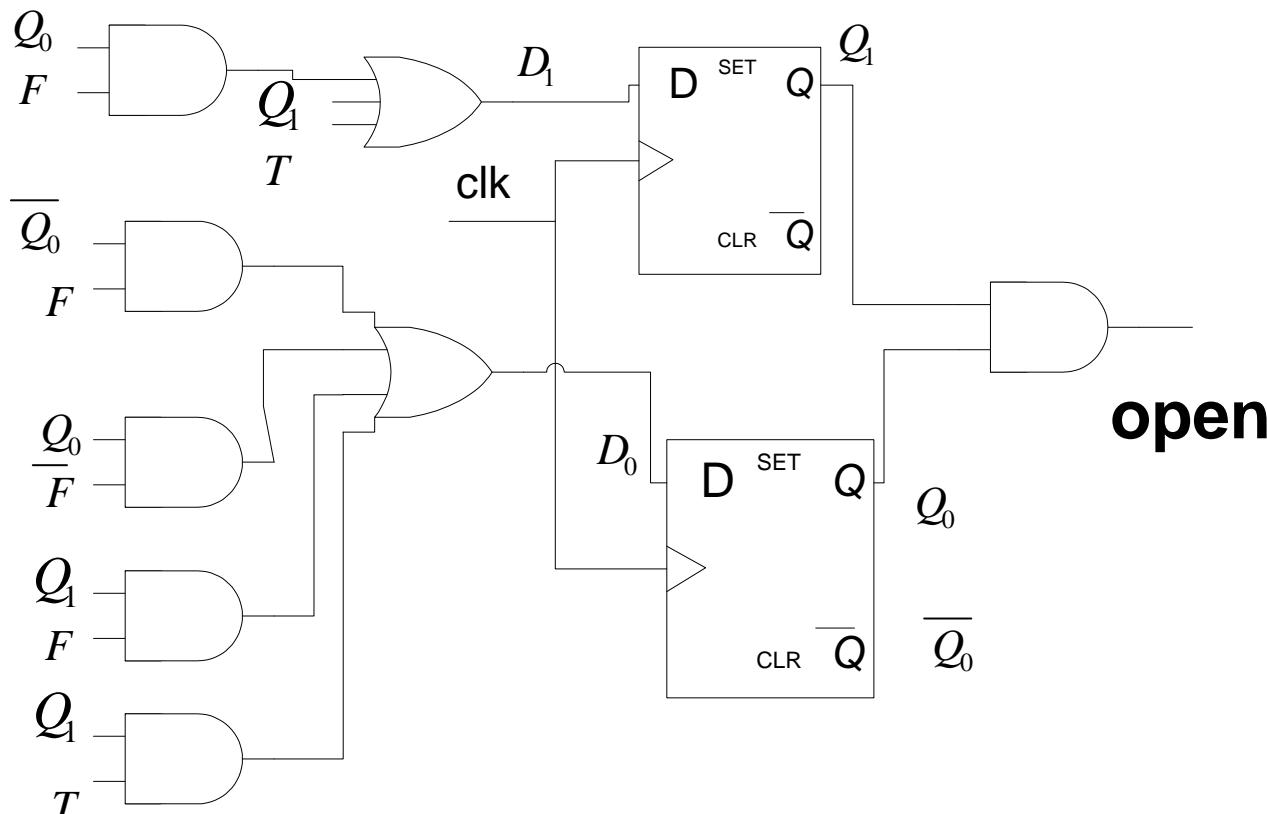


$$D_1 = Q_1 + T + Q_0 \cdot F$$

$$D_0 = F \cdot Q_0' + Q_0 \cdot F' + Q_1 \cdot F + Q_1 \cdot T$$

$$\text{Open} = Q_1 Q_0$$

## Vending Machine- D flip flops implementation



$$5p=F$$

$$10p=T$$

$$D_1 = Q_1 + T + Q_0 \cdot F$$

$$D_0 = F \cdot Q_0' + Q_0 \cdot F' + Q_1 \cdot F + Q_1 \cdot T$$

$$\text{Open} = Q_1 Q_0$$

# Vending Machine State Transition Diagram for J-K flip flops implementation

Present state		Inputs		Next state					
Q <sub>1</sub>	Q <sub>0</sub>	D	N	D <sub>1</sub>	D <sub>0</sub>	J <sub>1</sub>	K <sub>1</sub>	J <sub>0</sub>	K <sub>0</sub>
0	0	0	0	0	0	0	X	0	X
	0	1		0	1	0	X	1	X
	1	0		1	0	1	X	0	X
	1	1		X	X	x	X	X	X
0	1	0	0	0	0	0	X	X	0
	0	1		1	0	1	X	X	1
	1	0		1	1	1	X	X	0
	1	1		X	X	x	X	X	X
1	0	0	0	1	0	x	0	0	X
	0	1		1	1	x	0	1	X
	1	0		1	1	x	0	1	X
	1	1		X	X	x	x	X	X
1	1	0	0	1	1	x	0	X	0
	0	1		1	1	x	0	X	0
	1	0		1	1	x	0	X	0
	1	1		X	X	x	x	X	X

## Vending Machine- K Maps for J-K flip flops implementation

		Q1Q0		Q1		
		00	01	11	10	
		DN	D	N		
<b>J<sub>1</sub></b>	00	0	0	X	X	
	01	0	1	X	X	
	11	X	X	X	X	
	10	1	1	X	X	

		Q1Q0		Q1		
		00	01	11	10	
		DN	D	N		
<b>K<sub>1</sub></b>	00	x	x	0	0	
	01	x	x	0	0	
	11	X	X	X	X	
	10	X	X	0	0	

		Q1Q0		Q1		
		00	01	11	10	
		DN	C	D		
<b>J<sub>0</sub></b>	00	0	X	X	0	
	01	1	X	X	1	
	11	X	X	X	X	
	10	0	X	X	1	

$$J_1 = ?$$

$$K_1 = ?$$

		Q1Q0		Q1		
		00	01	11	10	
		DN	D	N		
<b>K<sub>0</sub></b>	00	X	0	0	X	
	01	X	1	0	X	
	11	X	X	X	X	
	10	X	0	0	X	

$$J_0 = ?$$

$$K = ?$$

# Vending Machine- K Maps for J-K flip flops implementation

		Q1Q0		Q1			
		00	01	11	10		
J <sub>1</sub>		DN	00	0	0	X	X
		01	0	1	X	X	
D		11	X	X	X	X	
		10	1	1	X	X	
				Q0			

		Q1Q0		Q1			
		00	01	11	10		
K <sub>1</sub>		DN	00	x	x	0	0
		01	x	x	0	0	
D		11	X	X	X	X	
		10	X	X	0	0	
				Q0			

		Q1Q0		Q1			
		00	01	11	10		
J <sub>0</sub>		DN	00	0	X	X	0
		01	1	X	X	1	
C		11	X	X	X	X	
		10	0	X	X	1	
				Q0			

		Q1Q0		Q1			
		00	01	11	10		
K <sub>0</sub>		DN	00	X	0	0	X
		01	X	1	0	X	
D		11	X	X	X	X	
		10	X	0	0	X	
				B			

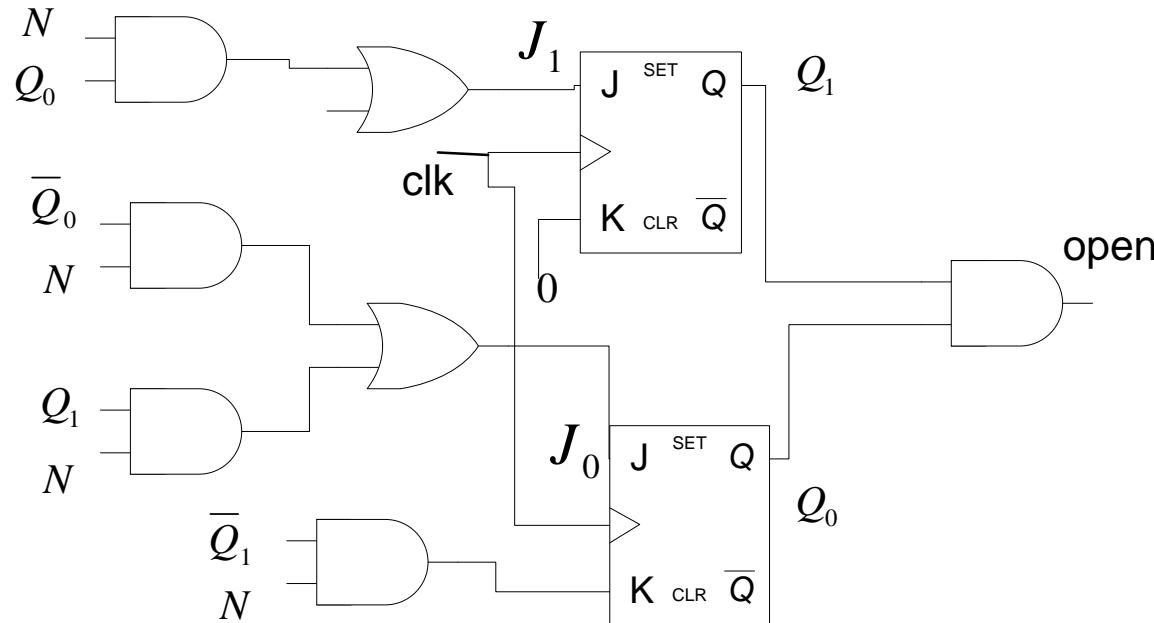
$$J_1 = D + Q_0 \cdot N$$

$$J_0 = Q_1' \cdot N + Q_1 \cdot D$$

$$K_1 = 0$$

$$K_0 = Q_1' \cdot N$$

## Vending Machine- J-K flip flops implementation



$$J_1 = D + Q_0 \cdot N$$

$$J_0 = Q_1' \cdot N + Q_1 \cdot D$$

$$K_1 = 0$$

$$K_0 = Q_1' \cdot N$$

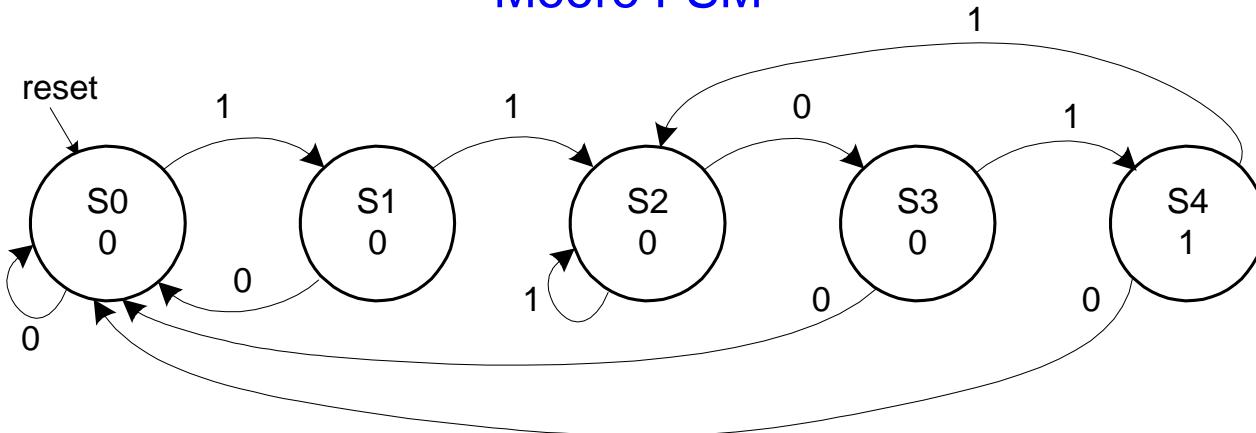
# **Moore vs. Mealy FSM**

- Design Moore and Mealy FSMs that detects

1101.

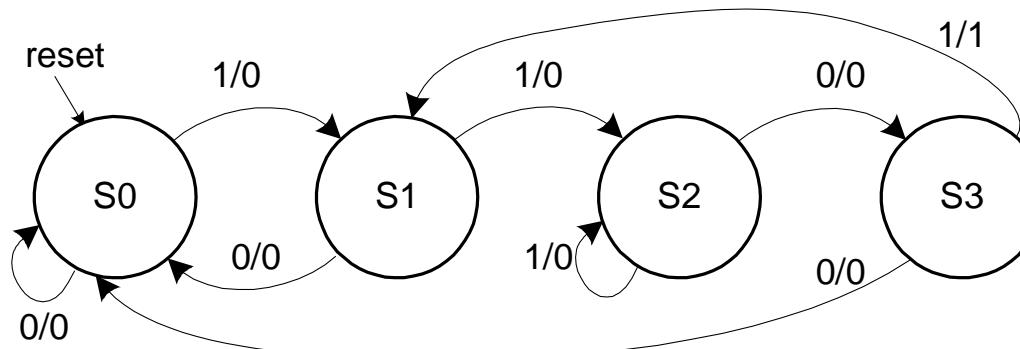
# State Transition Diagrams

Moore FSM



Mealy FSM: arcs indicate input/output

Mealy FSM



# Moore FSM State Transition Table

Current State			Inputs	Next State		
$S_2$	$S_1$	$S_0$	$A$	$S'_2$	$S'_1$	$S'_0$
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			

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

# Moore FSM State Transition Table

Current State			Inputs	Next State		
$S_2$	$S_1$	$S_0$	$A$	$S'_2$	$S'_1$	$S'_0$
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 Output Table

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

# Moore FSM Output Table

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

$$Y = S_2$$

# Mealy FSM State Transition and Output Table

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

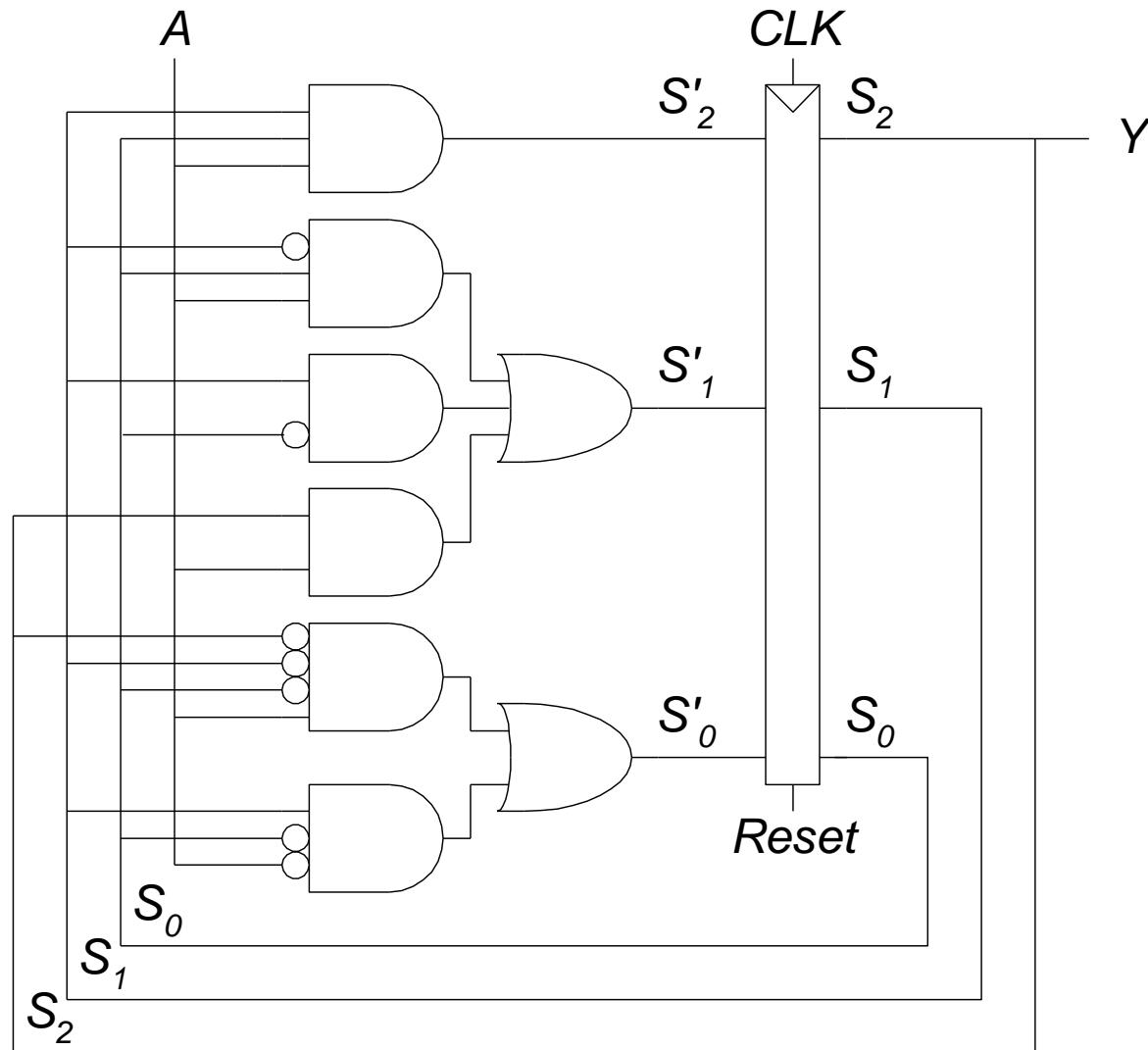
State	Encoding
S0	00
S1	01
S2	10
S3	11

# Mealy FSM State Transition and Output Table

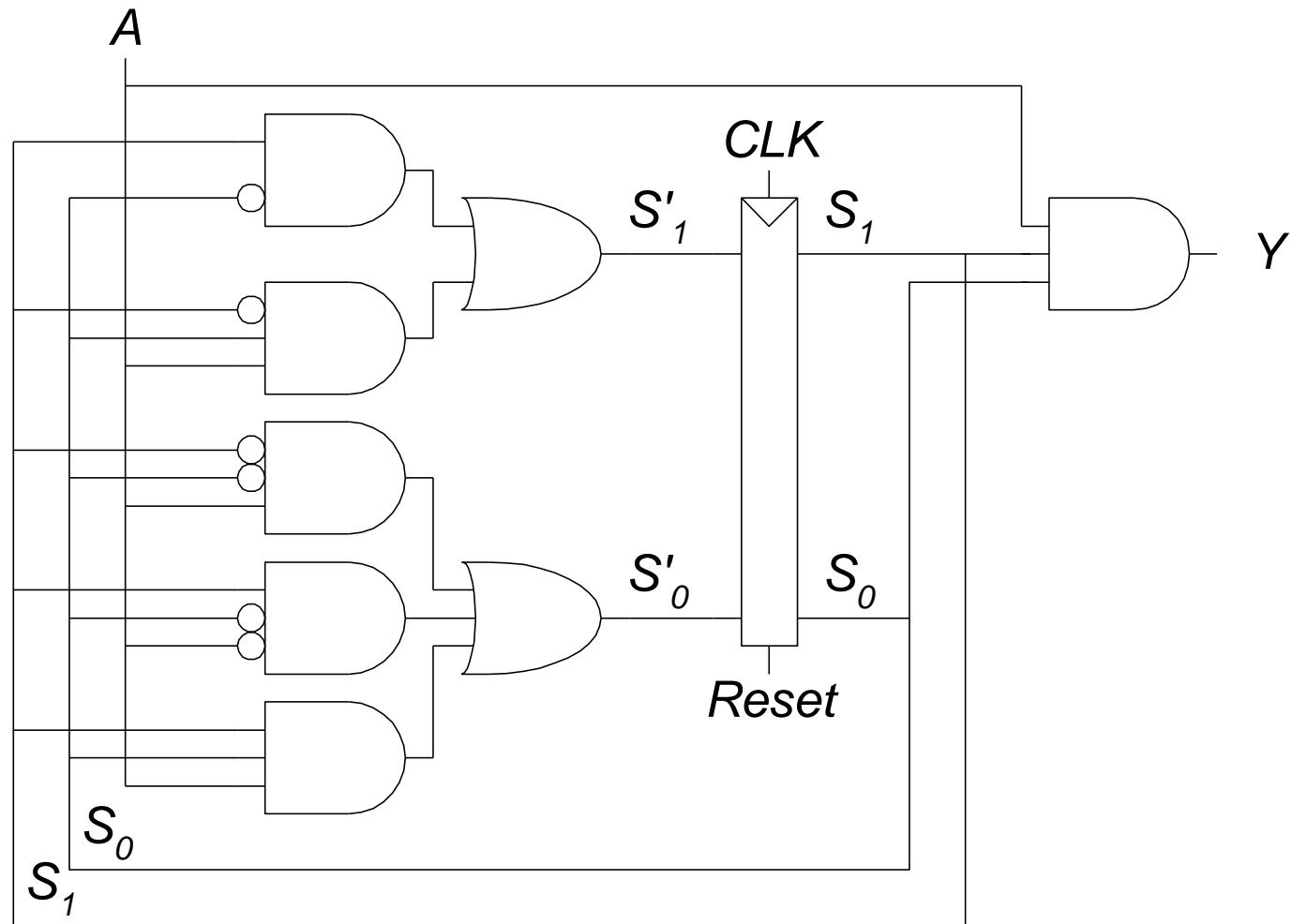
Current State		Input	Next State		Output
$S_1$	$S_0$	$A$	$S'_1$	$S'_0$	$Y$
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	1	1	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	1	0	1	1

State	Encoding
S0	00
S1	01
S2	10
S3	11

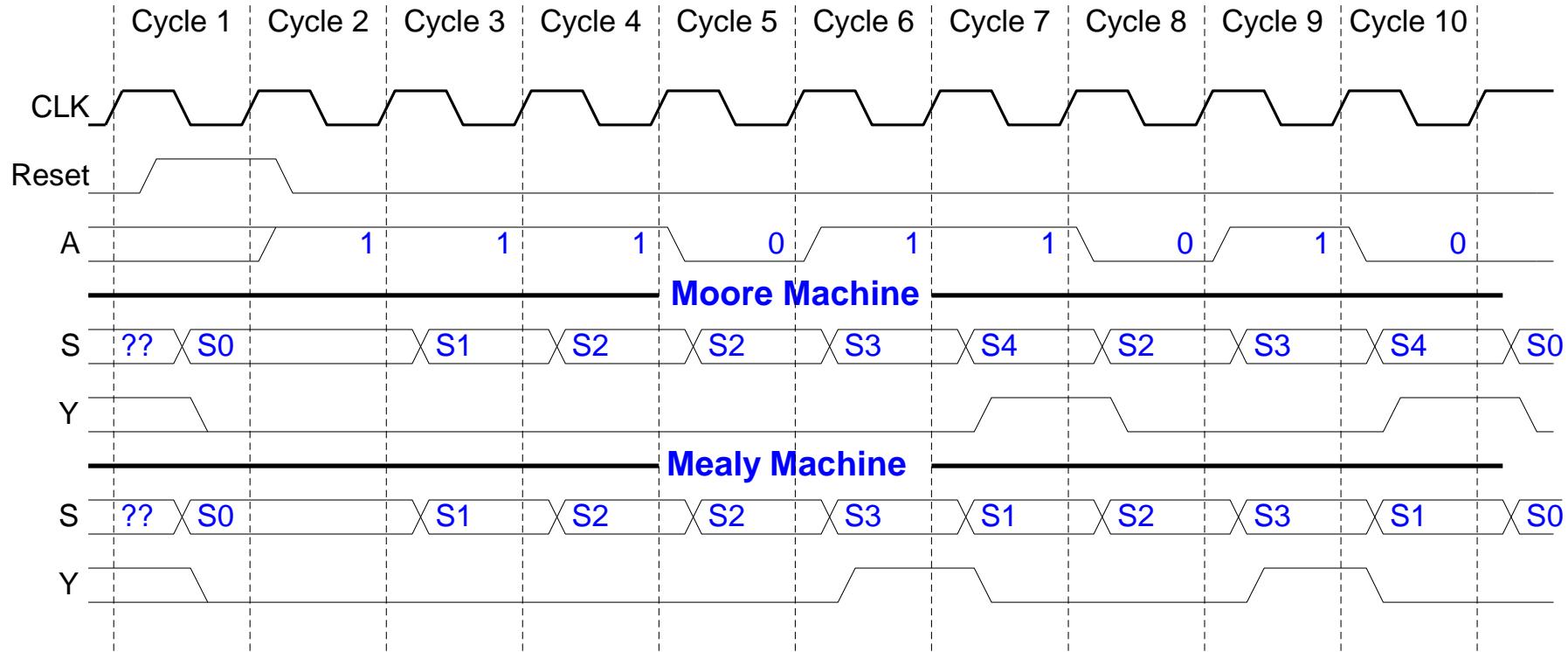
# Moore FSM Schematic



# Mealy FSM Schematic



# Moore and Mealy Timing Diagram

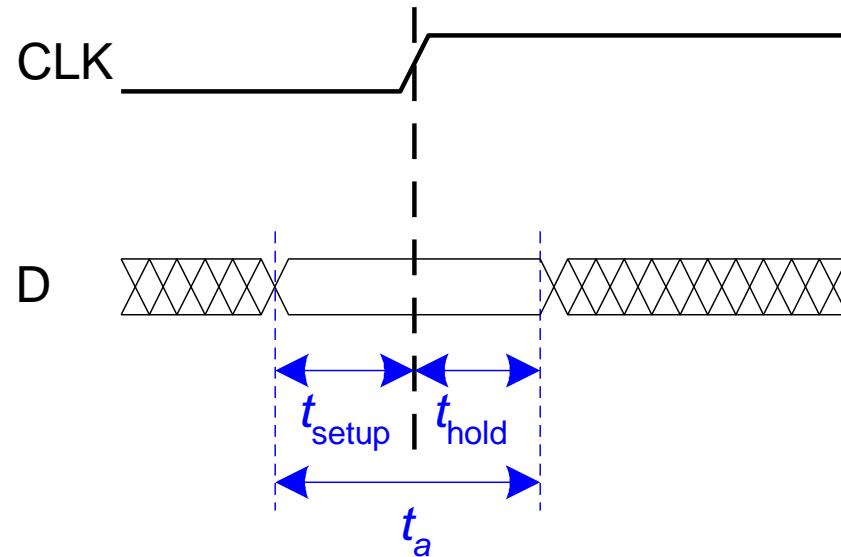


# Timing

- Flip-flop samples  $D$  at clock edge
- $D$  must be stable when it is sampled
- Similar to a photograph,  $D$  must be stable around the clock edge
- If  $D$  is changing when it is sampled, metastability can occur

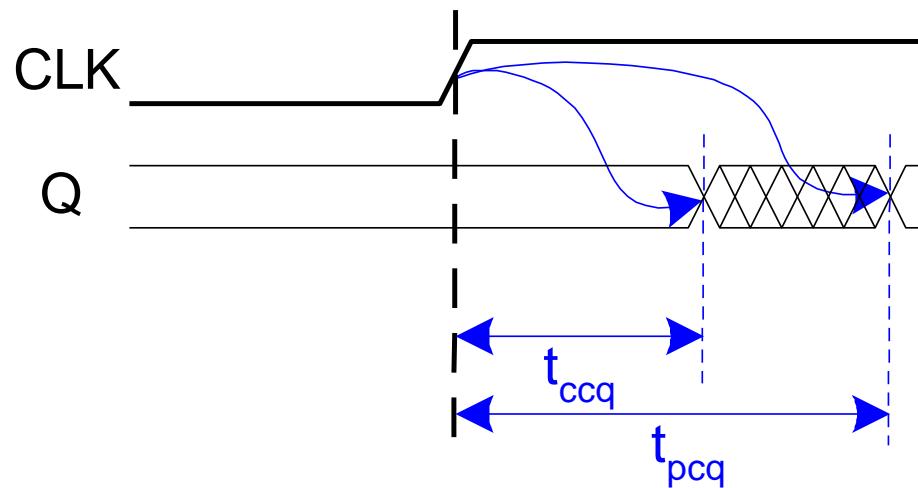
# Input Timing Constraints

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



# Output Timing Constraints

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

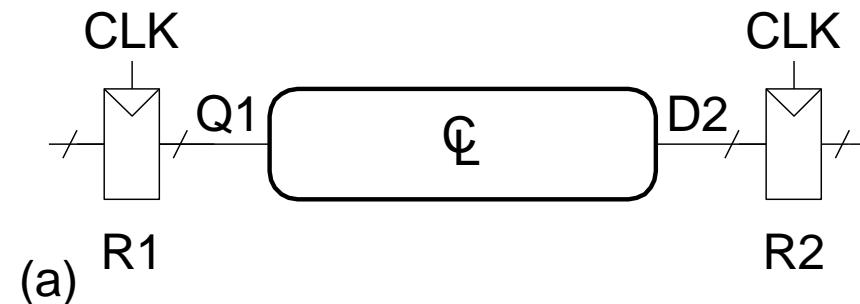


# Dynamic Discipline

- The input to a synchronous sequential circuit must be stable during the aperture (setup and hold) time around the clock edge.
- Specifically, the input must be stable
  - at least  $t_{\text{setup}}$  before the clock edge
  - at least until  $t_{\text{hold}}$  after the clock edge

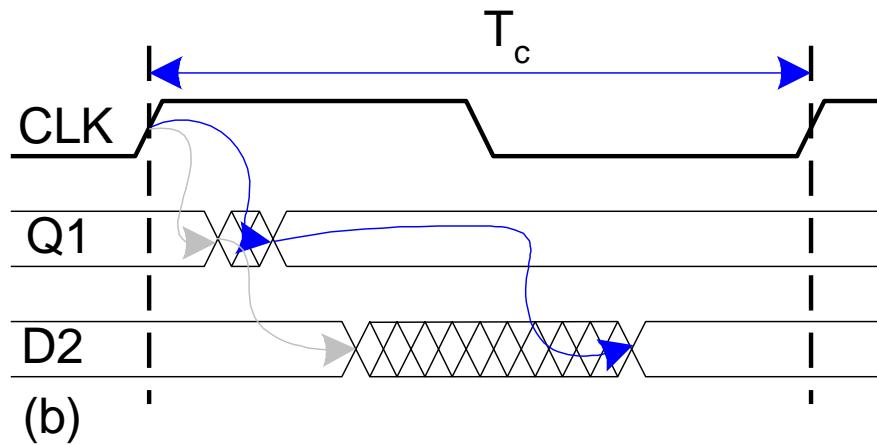
# Timing

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



(a) R1

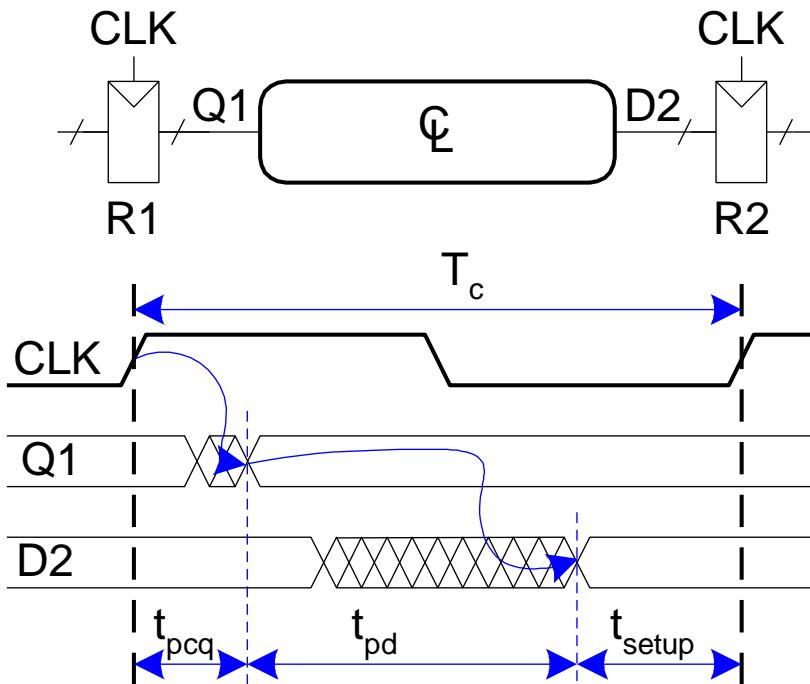
R2



(b)

# Setup Time Constraint

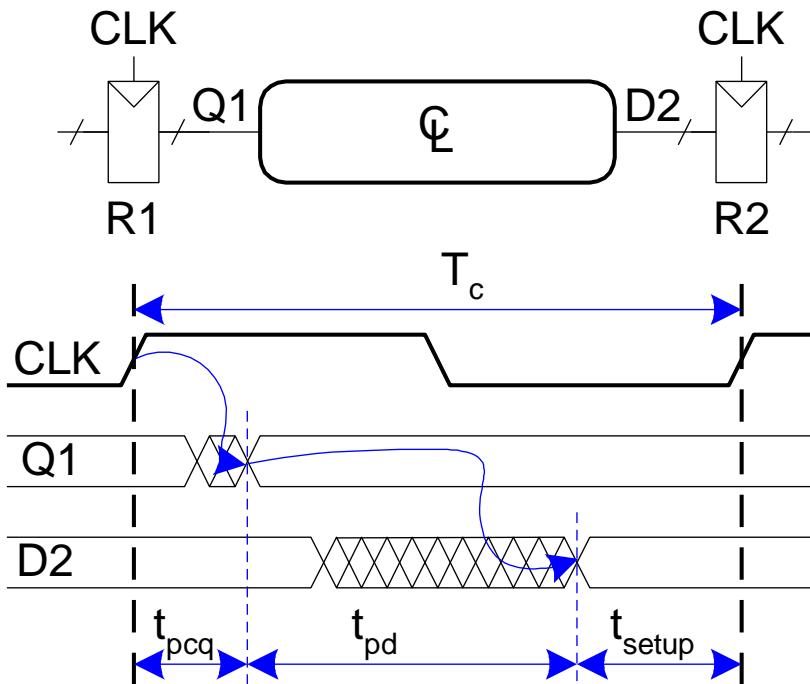
- The setup time constraint depends on the **maximum** delay from register R1 through the combinational logic.
- The input to register R2 must be stable at least  $t_{\text{setup}}$  before the clock edge.



$$T_c \geq$$

# Setup Time Constraint

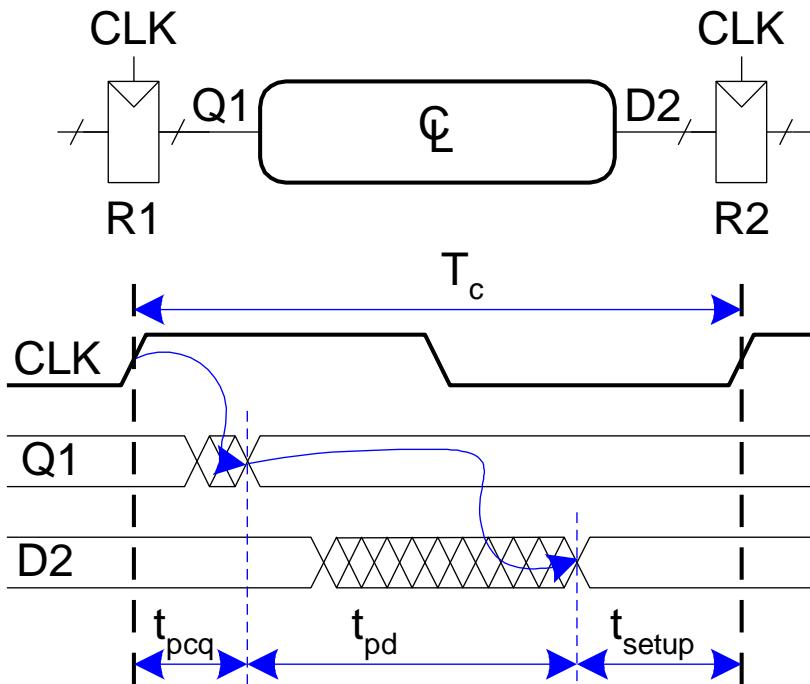
- The setup time constraint depends on the **maximum** delay from register R1 through the combinational logic.
- The input to register R2 must be stable at least  $t_{\text{setup}}$  before the clock edge.



$$T_c \geq t_{\text{pcq}} + t_{\text{pd}} + t_{\text{setup}}$$
$$t_{\text{pd}} \leq$$

# Setup Time Constraint

- The setup time constraint depends on the **maximum** delay from register R1 through the combinational logic.
- The input to register R2 must be stable at least  $t_{\text{setup}}$  before the clock edge.



$$T_c \geq t_{\text{pcq}} + t_{\text{pd}} + t_{\text{setup}}$$
$$t_{\text{pd}} \leq T_c - (t_{\text{pcq}} + t_{\text{setup}})$$