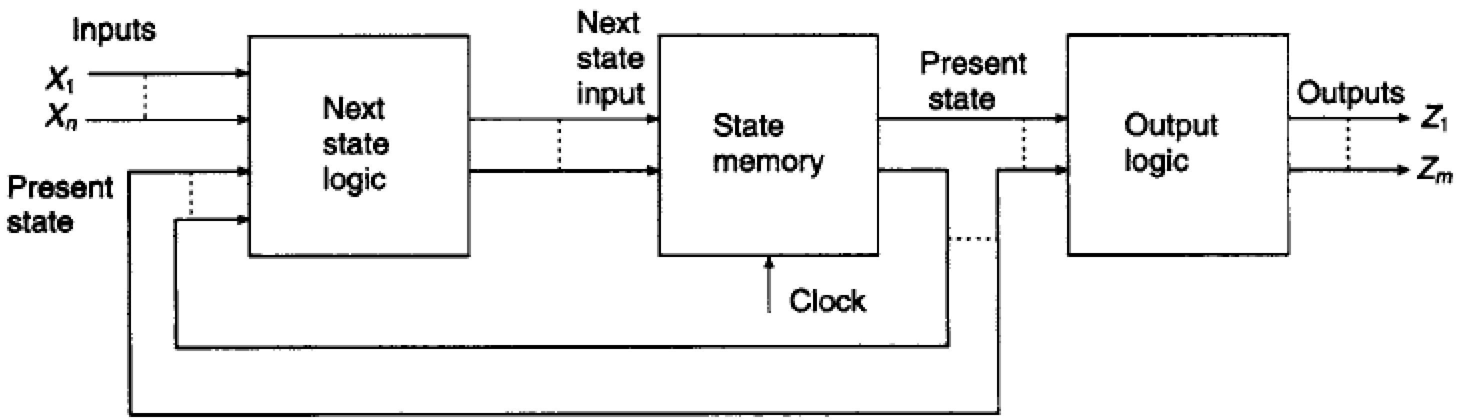
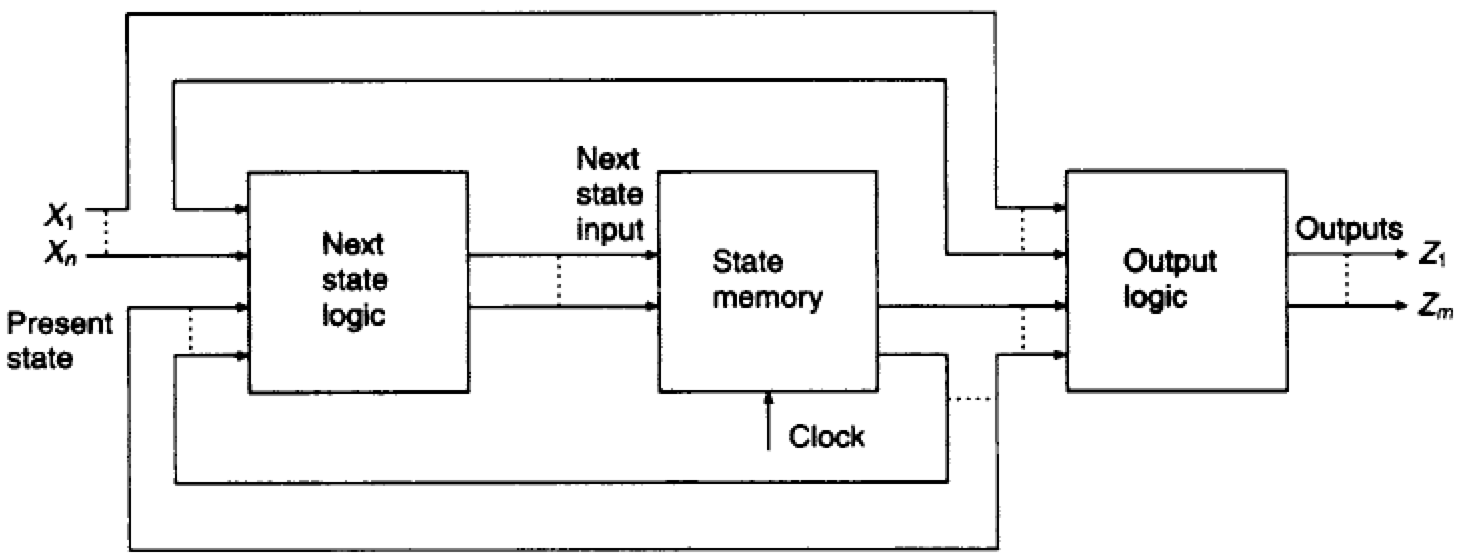
Finite-State Machines (FSMs) and Controllers


Inputs
X_1
X_n
Present
state
Next
state
logic
Next
state
input
State
memory
Clock
Present
state
Output
logic
Outputs
Z_1
Z_m

(a)

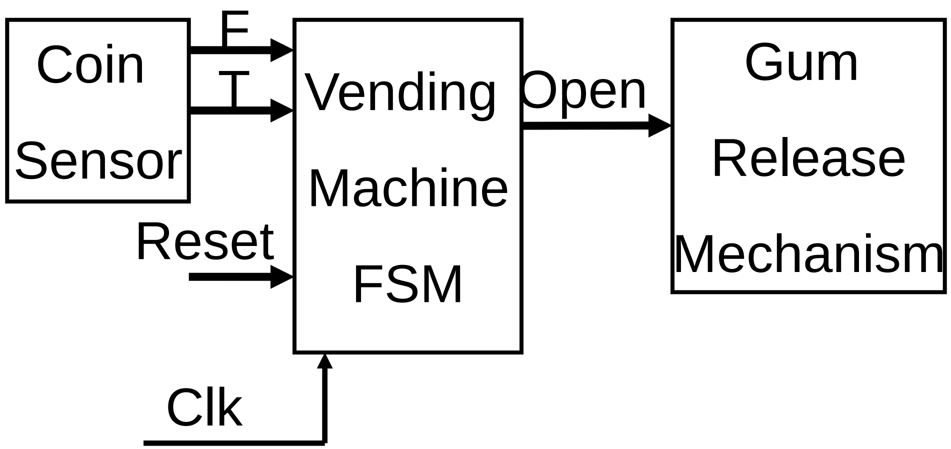

X_1
X_n
Present
state
Next
state
logic
Next
state
input
State
memory
Clock
Output
logic
Outputs
Z_1
Z_m

(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

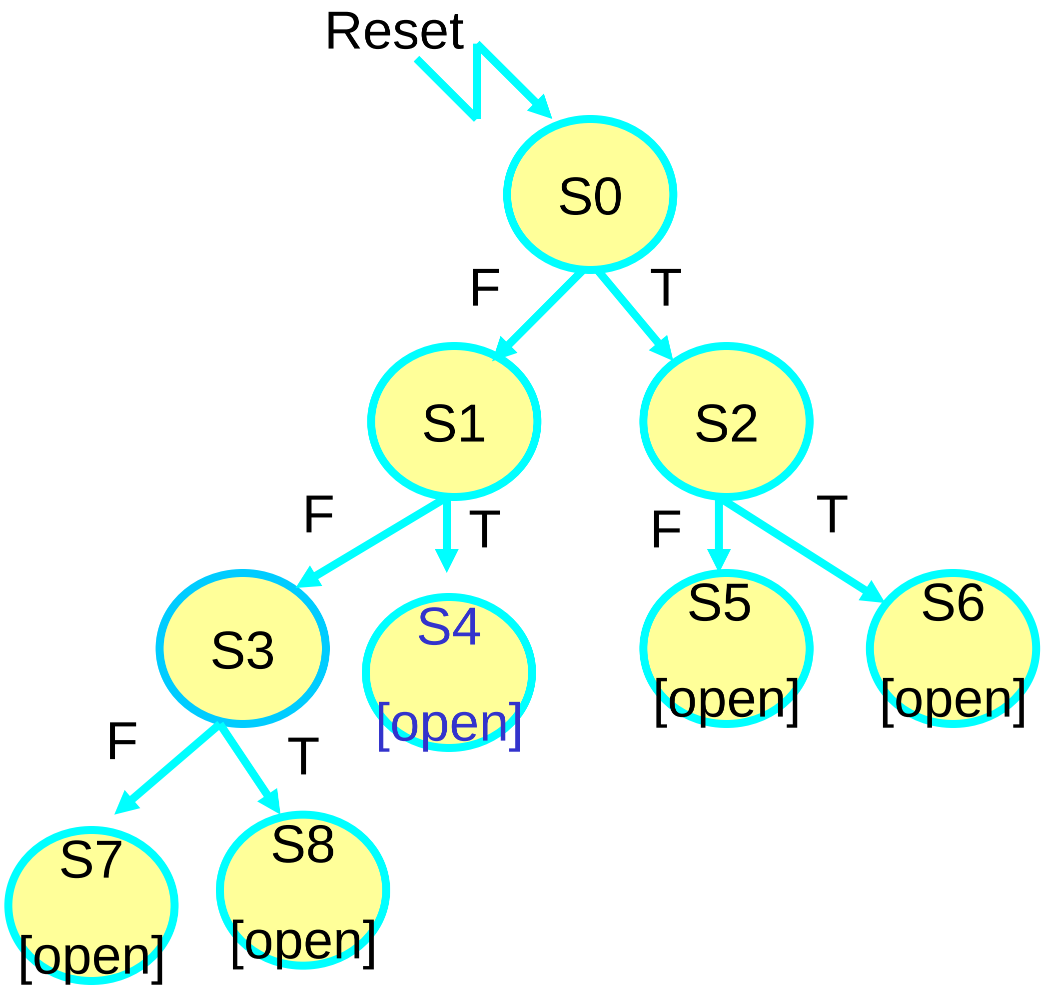


5p=F

10p=T

- 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

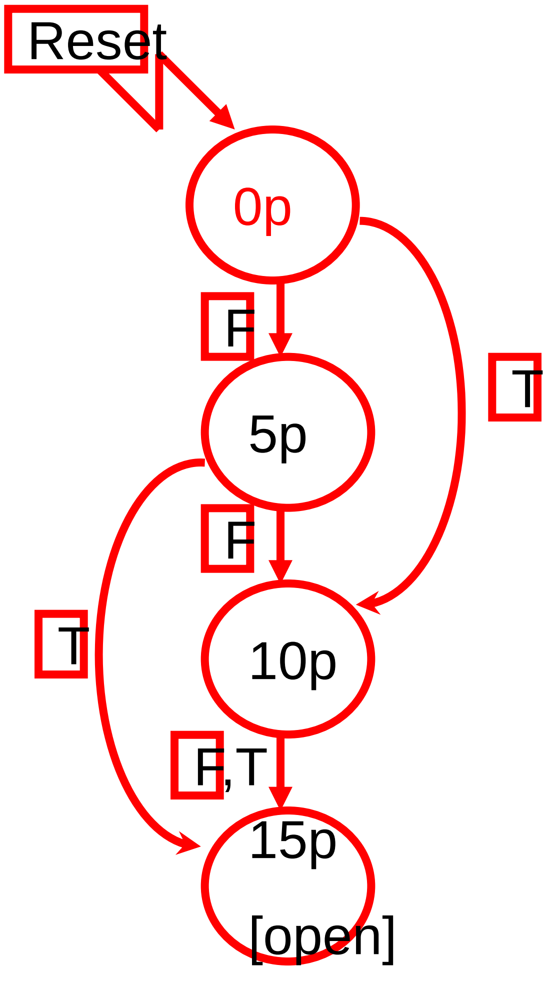
State diagram



5p=F

10p=T

Minimized state diagram



5p=F

10p=T

Minimized transition table

Present state	Inputs T	Inputs F	Next state	Output Open
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
Q_1	Q_0	T	F	D_1	D_0	Open
0	0	0	0	0	0	0
		0	1	0	1	0
		1	0	1	0	0
		1	1	X	X	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)

TF \ Q1Q0	00	01	11	10
00	0	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	1	1

K-map for D_1

TF \ Q1Q0	00	01	11	10
00	0	1	1	1
01	0	1	1	1
11	X	X	X	X
10	0	1	1	1

K-map for D_0

TF \ Q1Q0	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	X	X	X	X
10	0	0	1	0

K-map for Open

D_1 = ?
D_0 = ?
Open=?

Vending Machine- K Maps for D flip flops implementation

Design Example (Continued)

TF \ Q1Q0	00	01	11	10
00	0	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	1	1

K-map for D_1

TF \ Q1Q0	00	01	11	10
00	0	1	1	0
01	1	0	1	1
11	X	X	X	X
10	0	1	1	1

K-map for D_0

TF \ Q1Q0	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	X	X	X	X
10	0	0	1	0

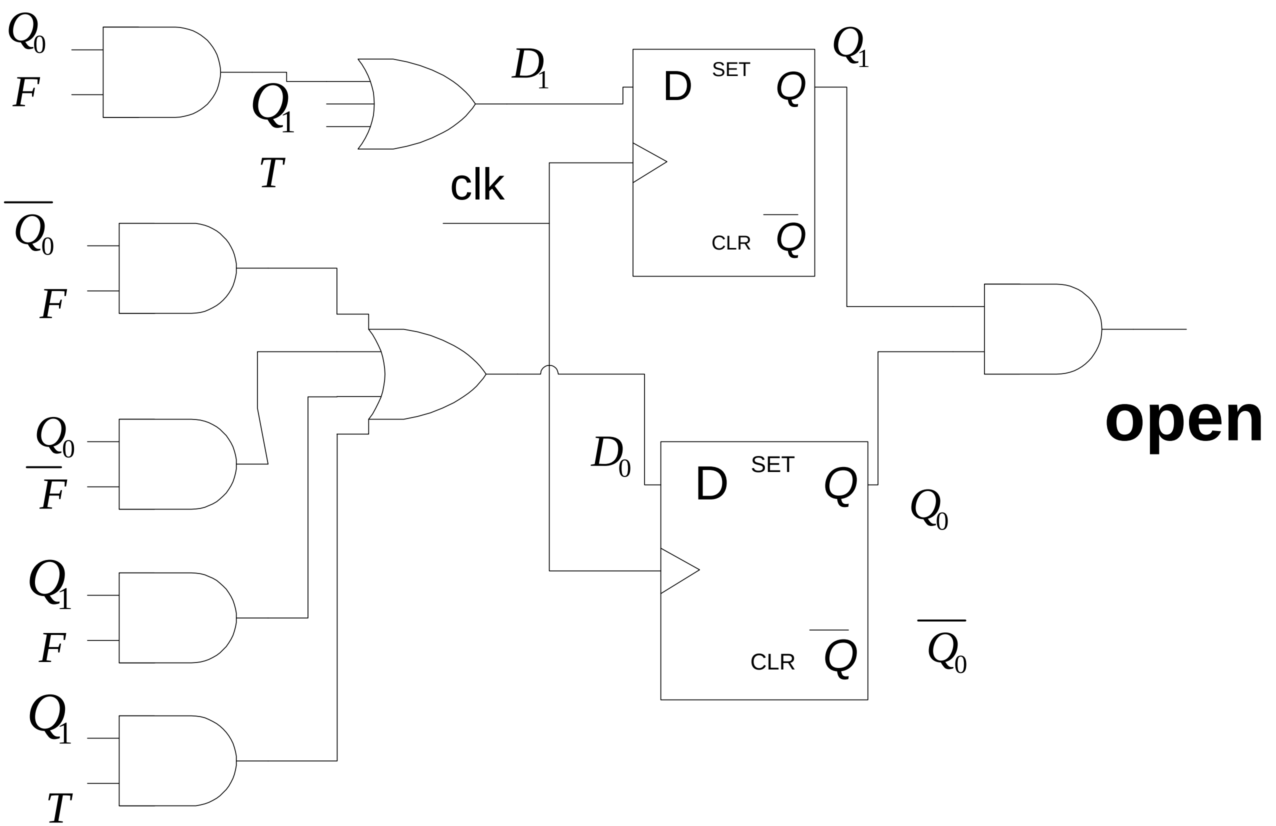
K-map for Open

$$D_1 = Q_1 + T + Q_0 . F$$

$$D_0 = F.Q_0' + Q_0.F' + Q_1 .F + Q_1 .T$$

$$Open = Q_1 Q_0$$

Vending Machine- D flip flops implementation



5p=F

10p=T

$$\mathbf{D_1 = Q_1 + T + Q_0 . F}$$

$$\mathbf{D_0 = F.Q_0' + Q_0.F' + Q_1 .F + Q_1 .T}$$

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

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

Present state Q_1	Q_0	Inputs D	N	Next state D_1	D_0	J_1	K_1	J_0	K_0
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

J_1

				Q1		
	DN \ Q1Q0	00	01	11	10	
	00	0	0	X	X	
	01	0	1	X	X	N
D	11	X	X	X	X	
	10	1	1	X	X	
			Q0			

K_1

				Q1		
	DN \ Q1Q0	00	01	11	10	
	00	x	x	0	0	
	01	x	x	0	0	N
D	11	X	X	X	X	
	10	X	X	0	0	
			Q0			

J_0

				Q1		
	DN \ Q1Q0	00	01	11	10	
	00	0	X	X	0	
	01	1	X	X	1	D
C	11	X	X	X	X	
	10	0	X	X	1	
			Q0			

K_0

				Q1		
	DN \ Q1Q0	00	01	11	10	
	00	X	0	0	X	
	01	X	1	0	X	N
D	11	X	X	X	X	
	10	X	0	0	X	
			B			

J_1 = ?

K_1 = ?

J_0 = ?

K = ?

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

J_1

DN \ Q1Q0	00	01	11	10
00	0	0	X	X
01	0	1	X	X
11	X	X	X	X
10	1	1	X	X

Q1 (columns 11, 10); Q0 (columns 01, 11); N (rows 01, 11); D (rows 11, 10)

K_1

DN \ Q1Q0	00	01	11	10
00	x	x	0	0
01	x	x	0	0
11	X	X	X	X
10	X	X	0	0

Q1 (columns 11, 10); Q0 (columns 01, 11); N (rows 01, 11); D (rows 11, 10)

J_0

DN \ Q1Q0	00	01	11	10
00	0	X	X	0
01	1	X	X	1
11	X	X	X	X
10	0	X	X	1

Q1 (columns 11, 10); Q0 (columns 01, 11); D (rows 01, 11); C (rows 11, 10)

K_0

DN \ Q1Q0	00	01	11	10
00	X	0	0	X
01	X	1	0	X
11	X	X	X	X
10	X	0	0	X

Q1 (columns 11, 10); B (columns 01, 11); N (rows 01, 11); D (rows 11, 10)

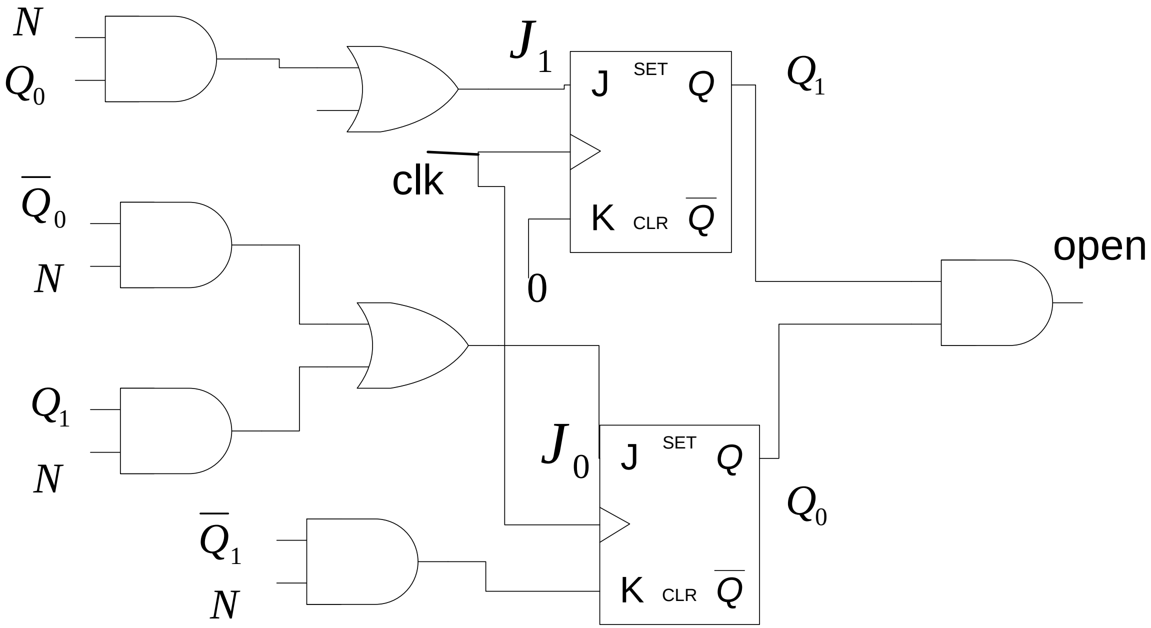
$J_1 = D+Q_0 .N$

$K_1 = 0$

$J_0 = Q_1' . N + Q_1 . D$

$K_0 = Q_1' . N$

Vending Machine- J-K flip flops implementation



$J_1 = D+Q_0 .N$

$K_1 = 0$

$J_0 = Q_1' . N + Q_1 . D$

$K_0 = Q_1' . N$

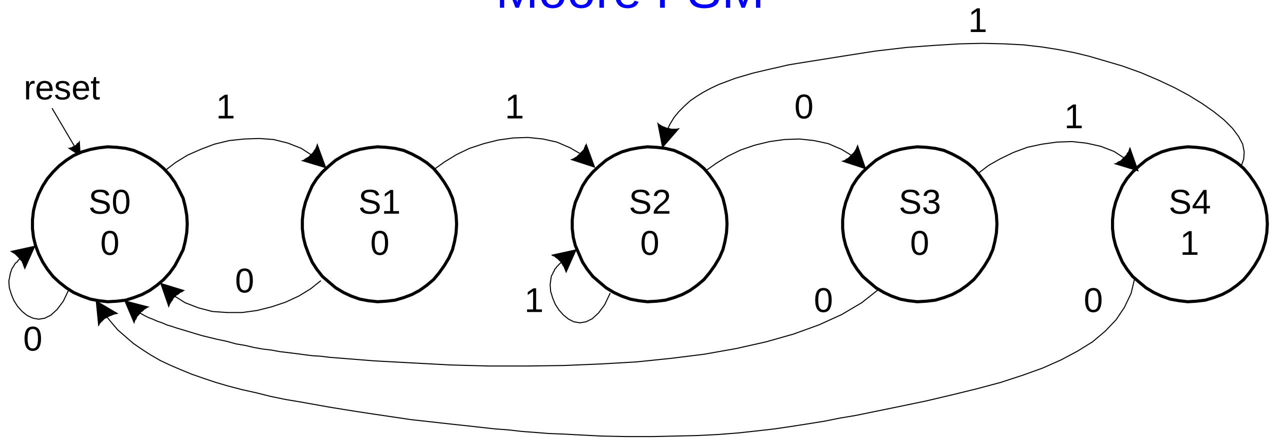
Moore vs. Mealy FSM

- Design Moore and Mealy FSMs that detects

1101.

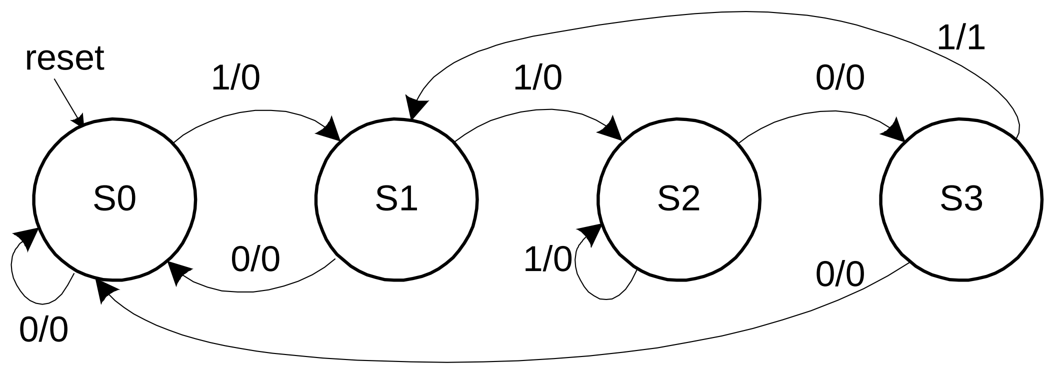
State Transition Diagrams

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			

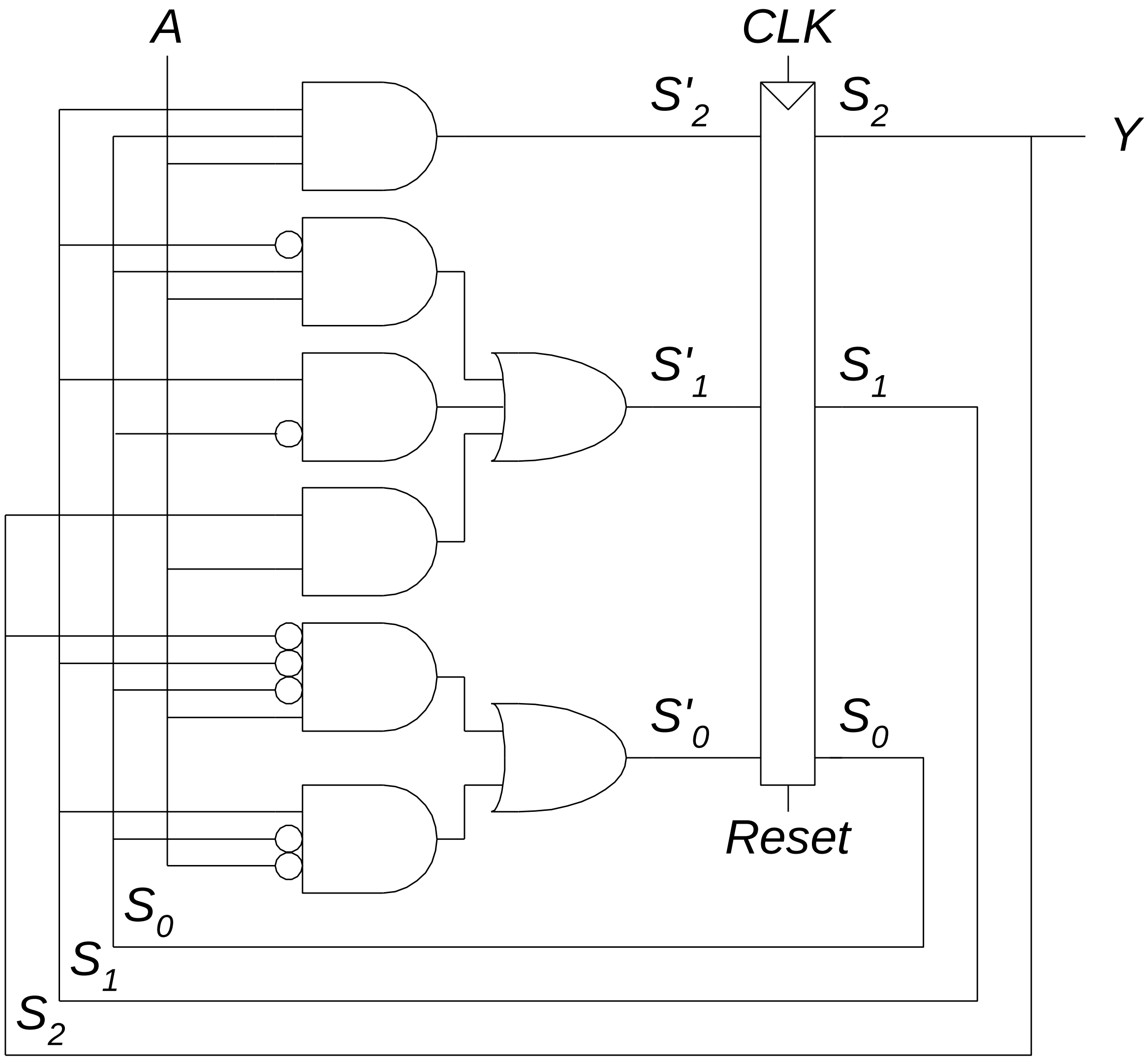
State	Encoding
S0	00
S1	01
S2	10
S3	11

Mealy FSM State Transition and Output Table

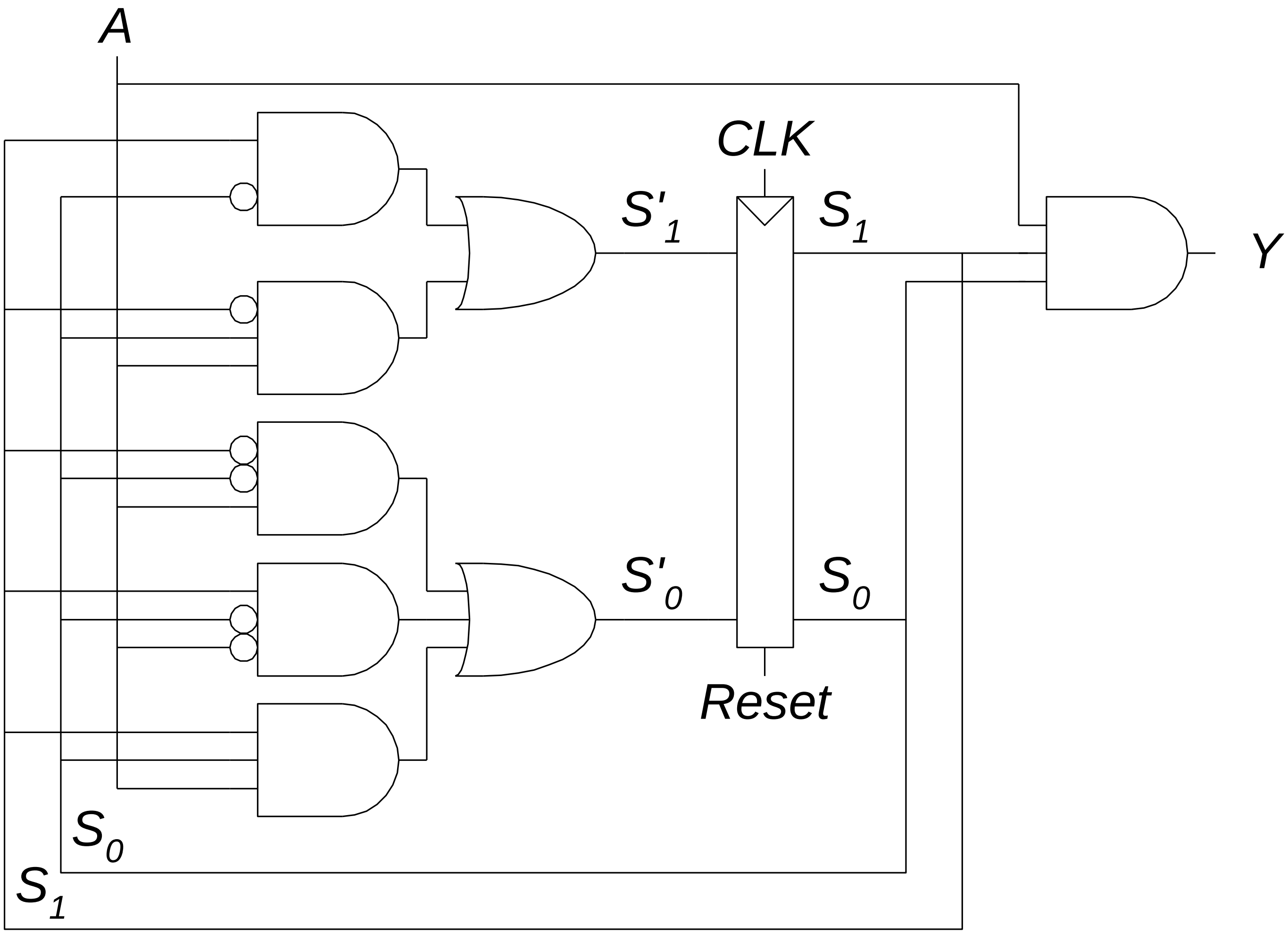
Current State		Input	Next State		Output
S_1	S_0	A	S'_1	S'_0	Y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	1	1	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	1	0	1	1

State	Encoding
S0	00
S1	01
S2	10
S3	11

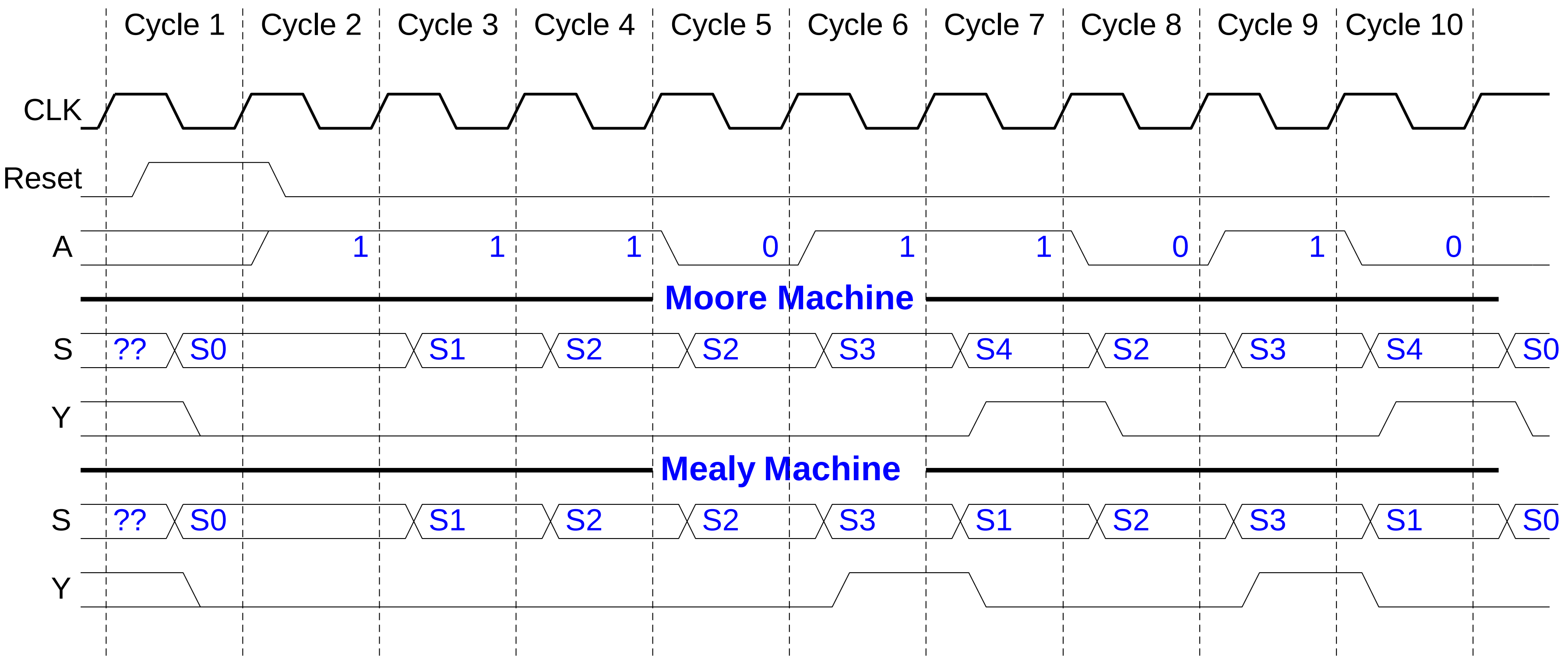
Moore FSM Schematic



Mealy FSM Schematic



Moore and Mealy Timing Diagram

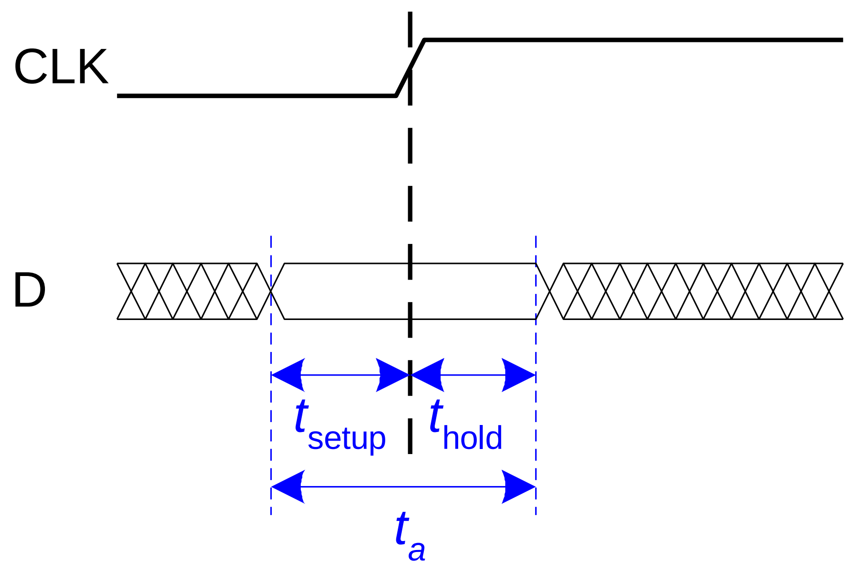


Timing

- 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

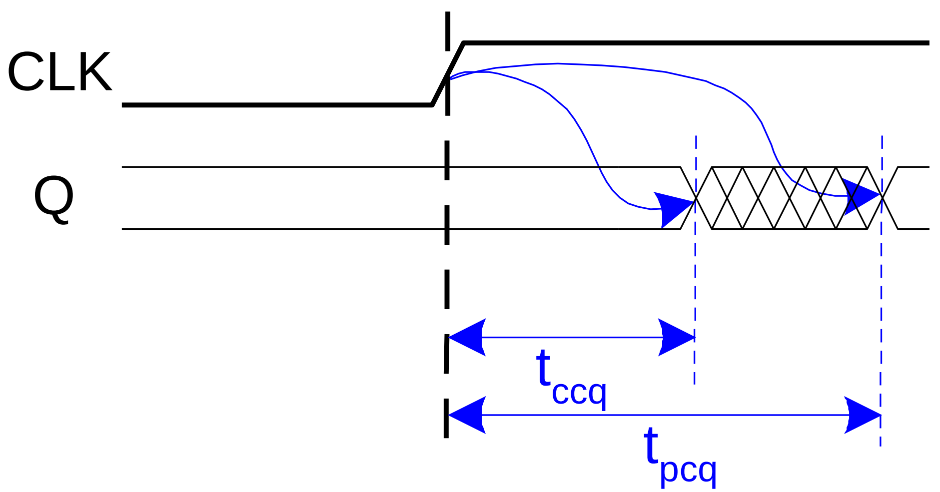
Input Timing Constraints

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



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)

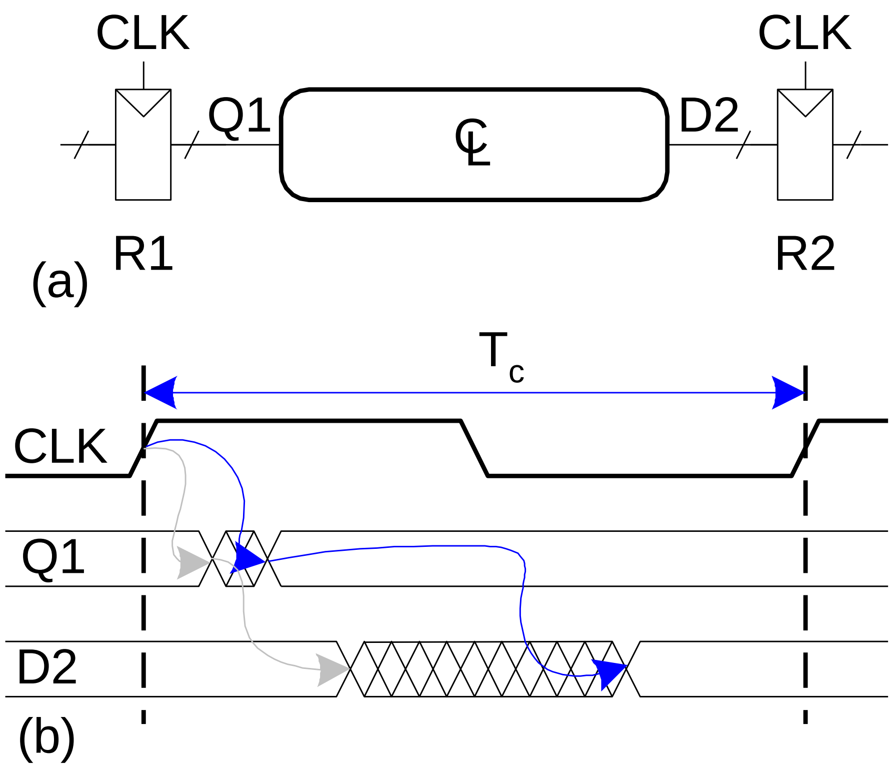


Dynamic Discipline

- 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_{setup} before the clock edge
  - at least until t_{hold} after the clock edge

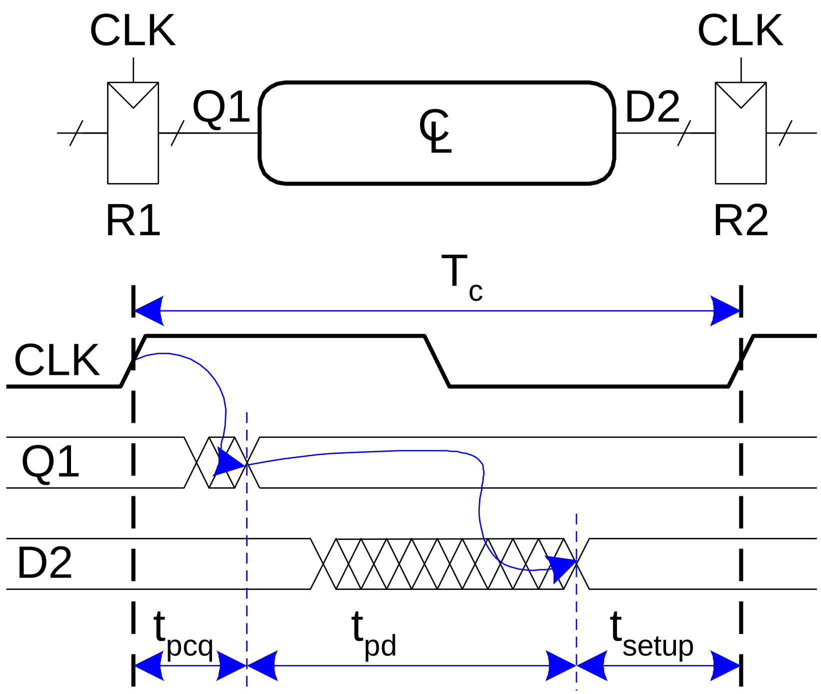
Timing

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



Setup Time Constraint

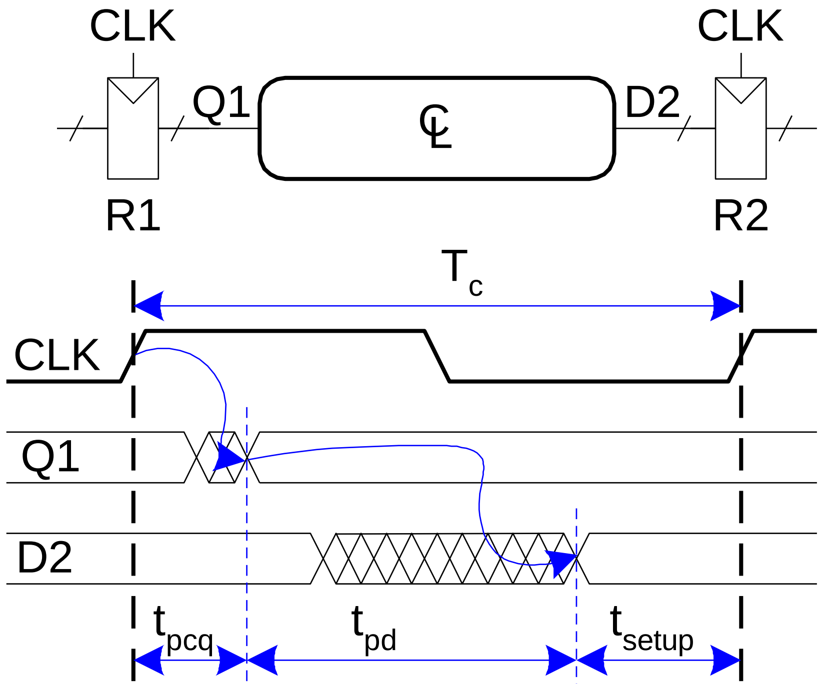
- The setup time constraint depends on the **maximum** delay from register R1 through the combinational logic.
- The input to register R2 must be stable at least t_{setup} before the clock edge.



$$T_c \geq$$

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_{setup} before the clock edge.

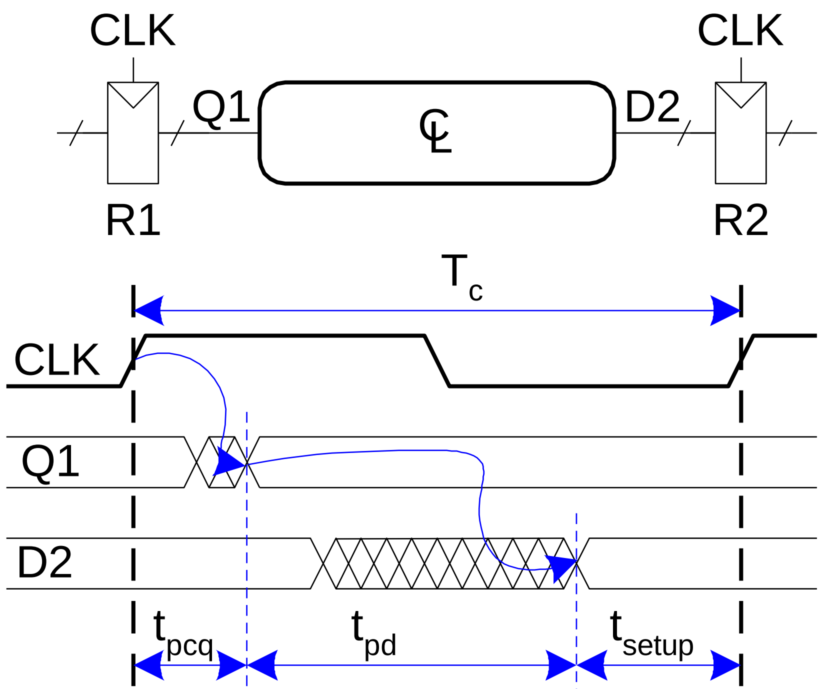


$$T_c \geq t_{pcq} + t_{pd} + t_{setup}$$

$$t_{pd} \leq$$

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_{setup} before the clock edge.



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