



L4: Sequential Building Blocks

(Flip-flops, Latches and Registers)



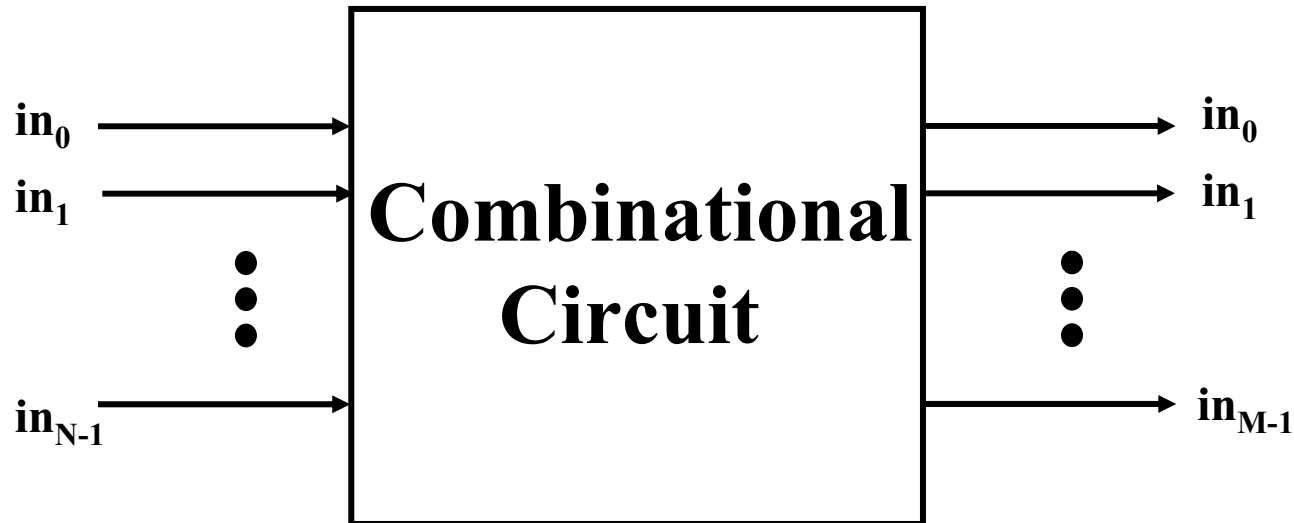
Acknowledgements:

Materials in this lecture are courtesy of the following people and used with permission.

- Randy H. Katz (University of California, Berkeley, Department of Electrical Engineering & Computer Science)
- Gaetano Borriello (University of Washington, Department of Computer Science & Engineering, <http://www.cs.washington.edu/370>)
- Rabaey, A. Chandrakasan, B. Nikolic. *Digital Integrated Circuits: A Design Perspective*. Prentice Hall, 2003.



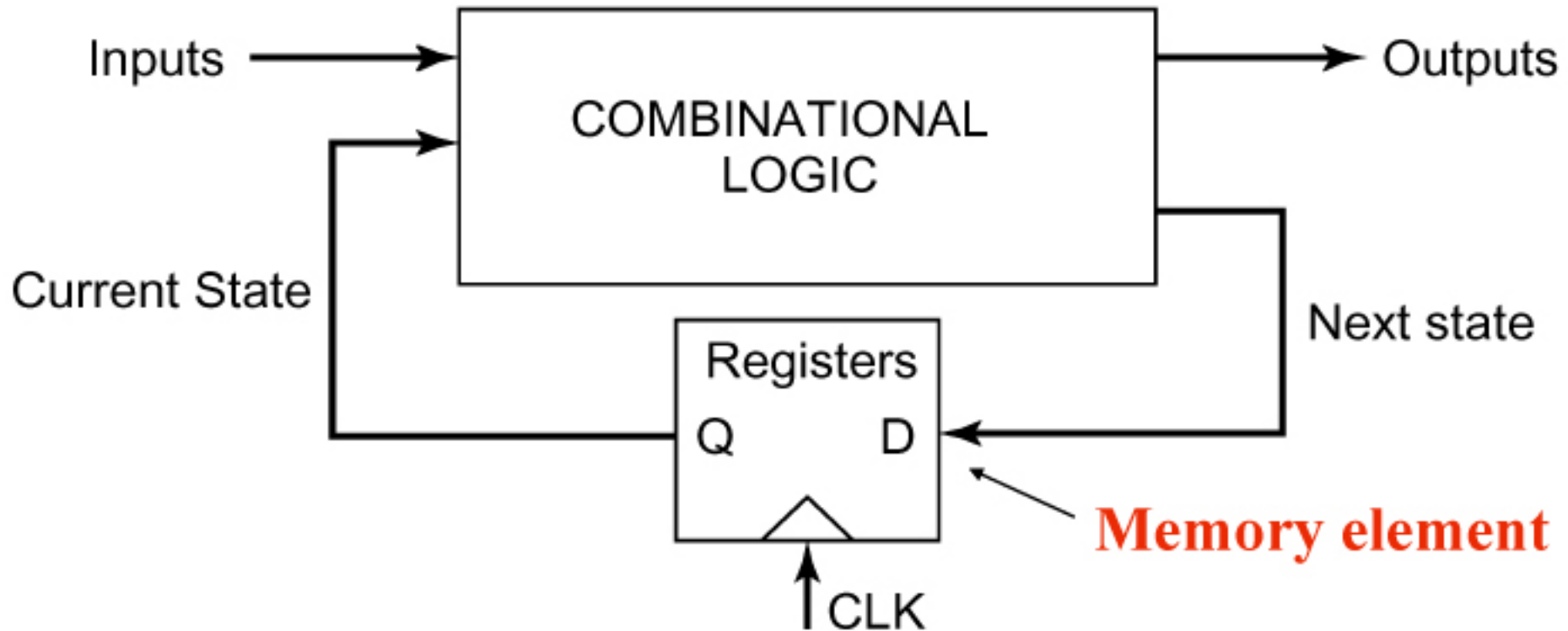
Combinational Logic Review



- **Combinational logic circuits are memoryless**
- **No feedback in combinational logic circuits**
- **Output assumes the function implemented by the logic network, assuming that the switching transients have settled**
- **Outputs can have multiple logical transitions before settling to the correct value**



A Sequential System



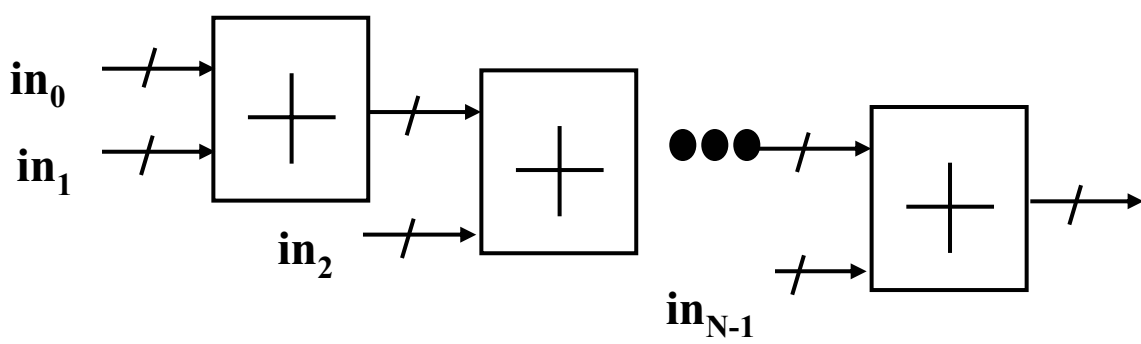
- Sequential circuits have memory (i.e., remember the past)
- The current state is “held” in memory and the next state is computed based the current state and the current inputs
- In a synchronous systems, the **clock signal** orchestrates the sequence of events



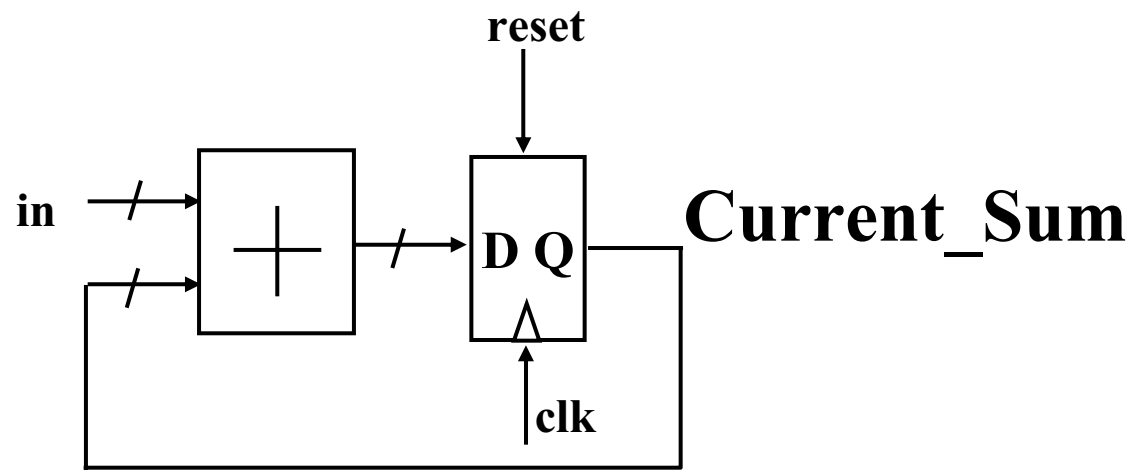
A Simple Example



Adding N inputs (N-1 Adders)

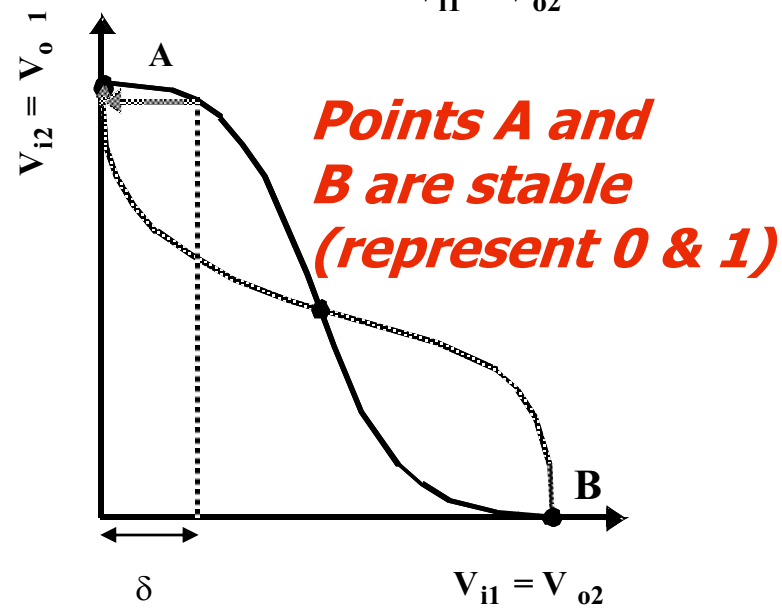
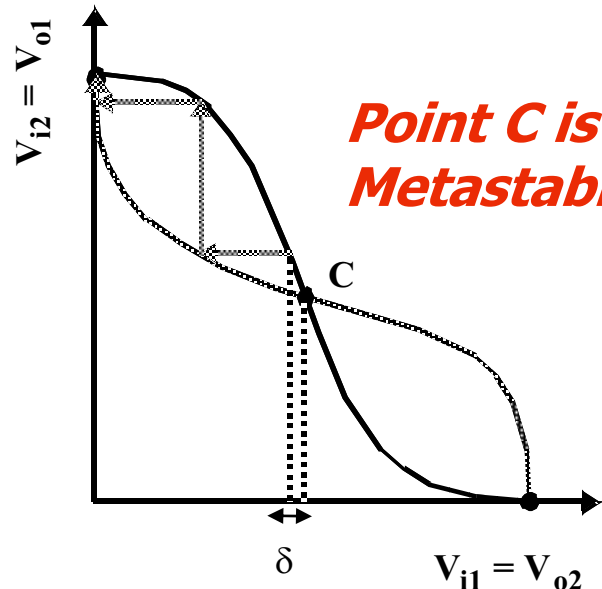
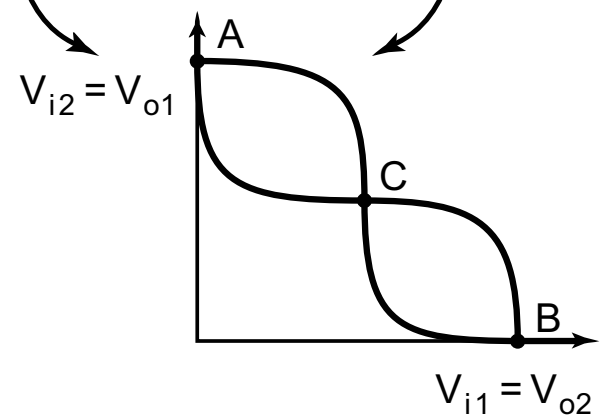
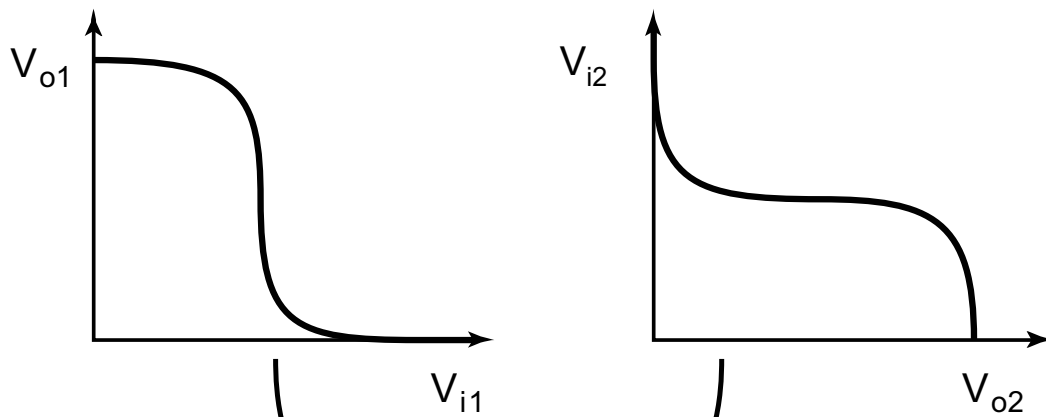
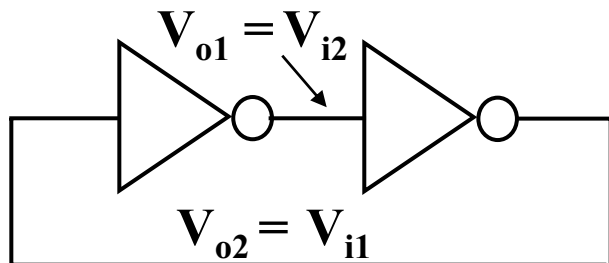


Using a sequential (serial) approach

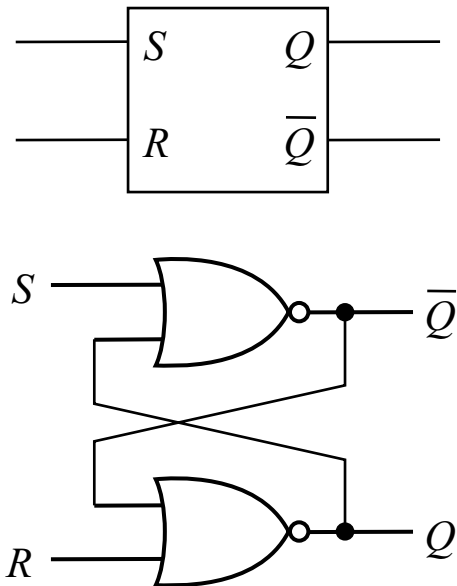




Implementing State: Bi-stability

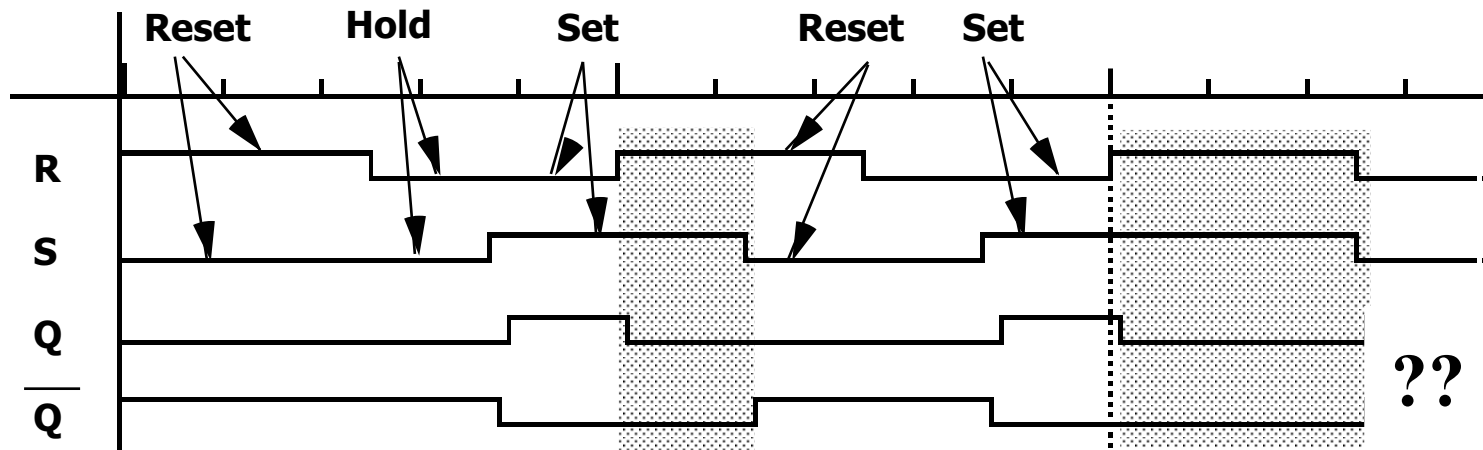
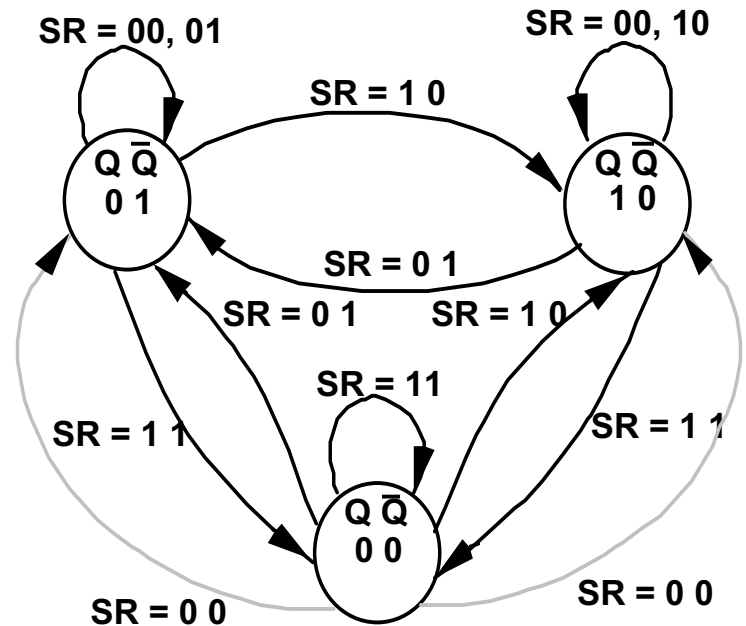


NOR-based Set-Reset (SR) Flipflop



S	R	Q	\bar{Q}
0	0	Q	\bar{Q}
1	0	1	0
0	1	0	1
1	1	0	0

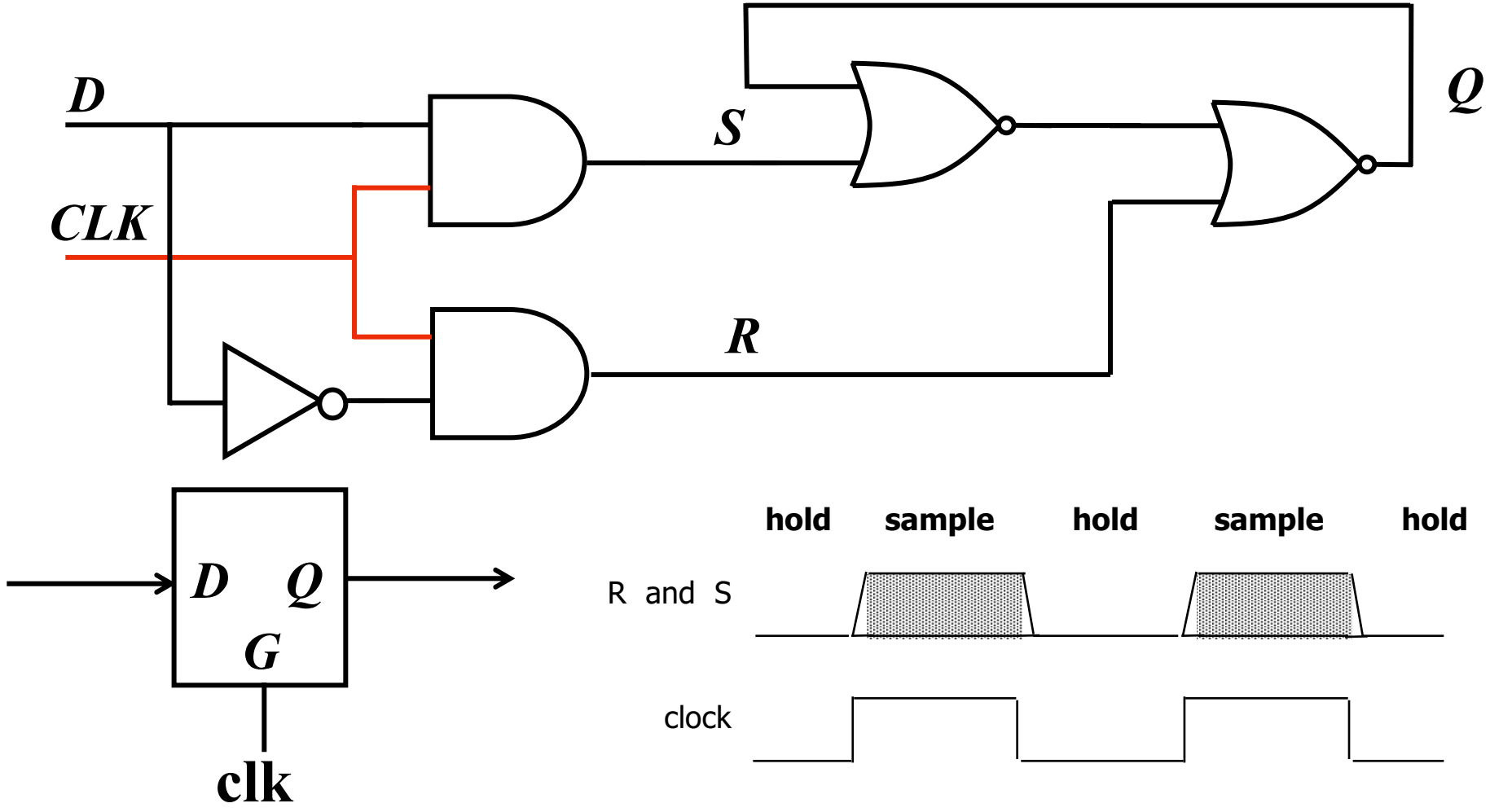
Forbidden State



- Flip-flop refers to a bi-stable element (edge-triggered registers are also called flip-flops) – this circuit is not clocked and outputs change “asynchronously” with the inputs



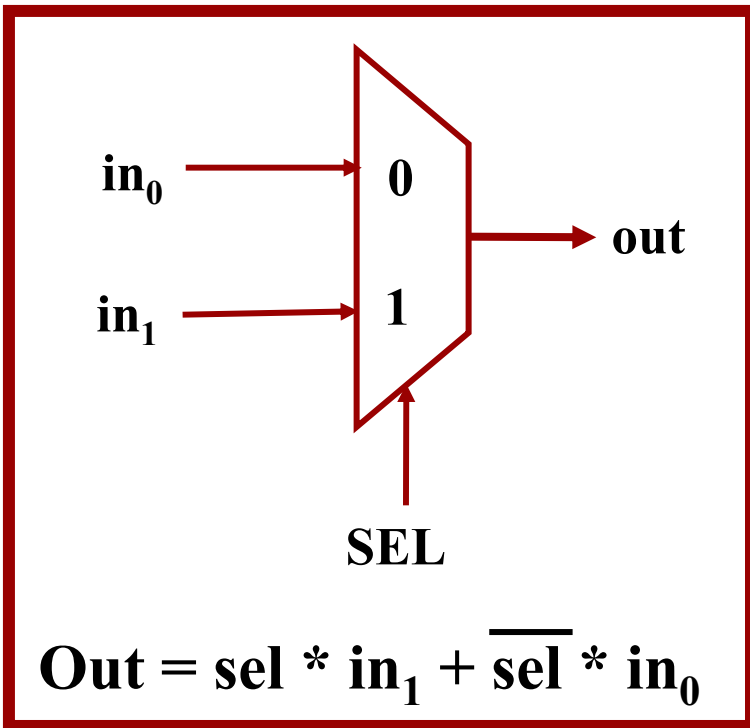
Making a Clocked Memory Element: Positive D-Latch



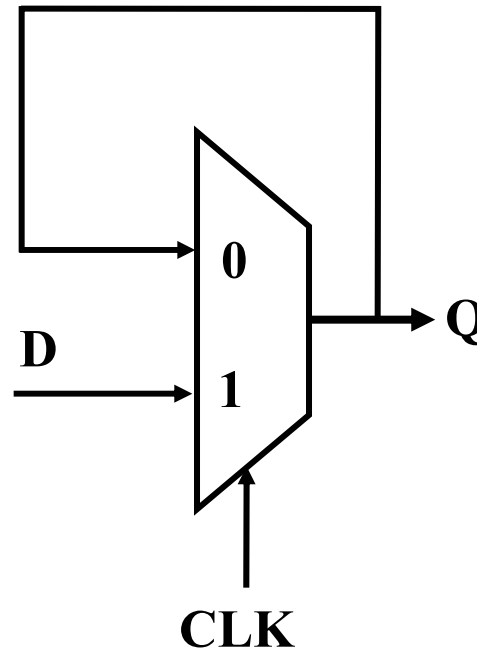
- **A Positive D-Latch:** Passes input D to output Q when CLK is high and holds state when clock is low (i.e., ignores input D)
- **A Latch is level-sensitive:** invert clock for a negative latch

Multiplexor Based Positive & Negative Latch

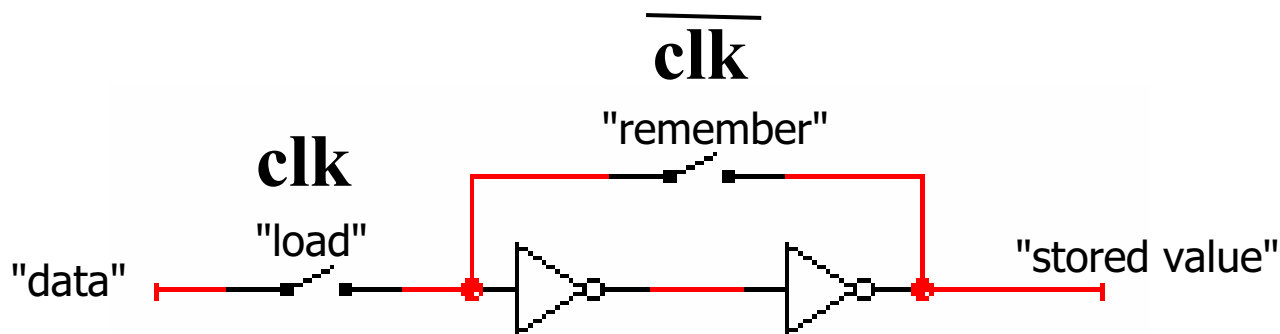
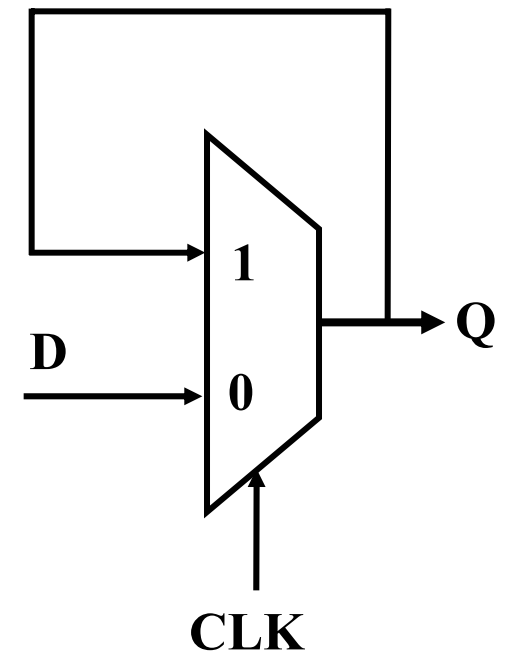
2:1 multiplexor



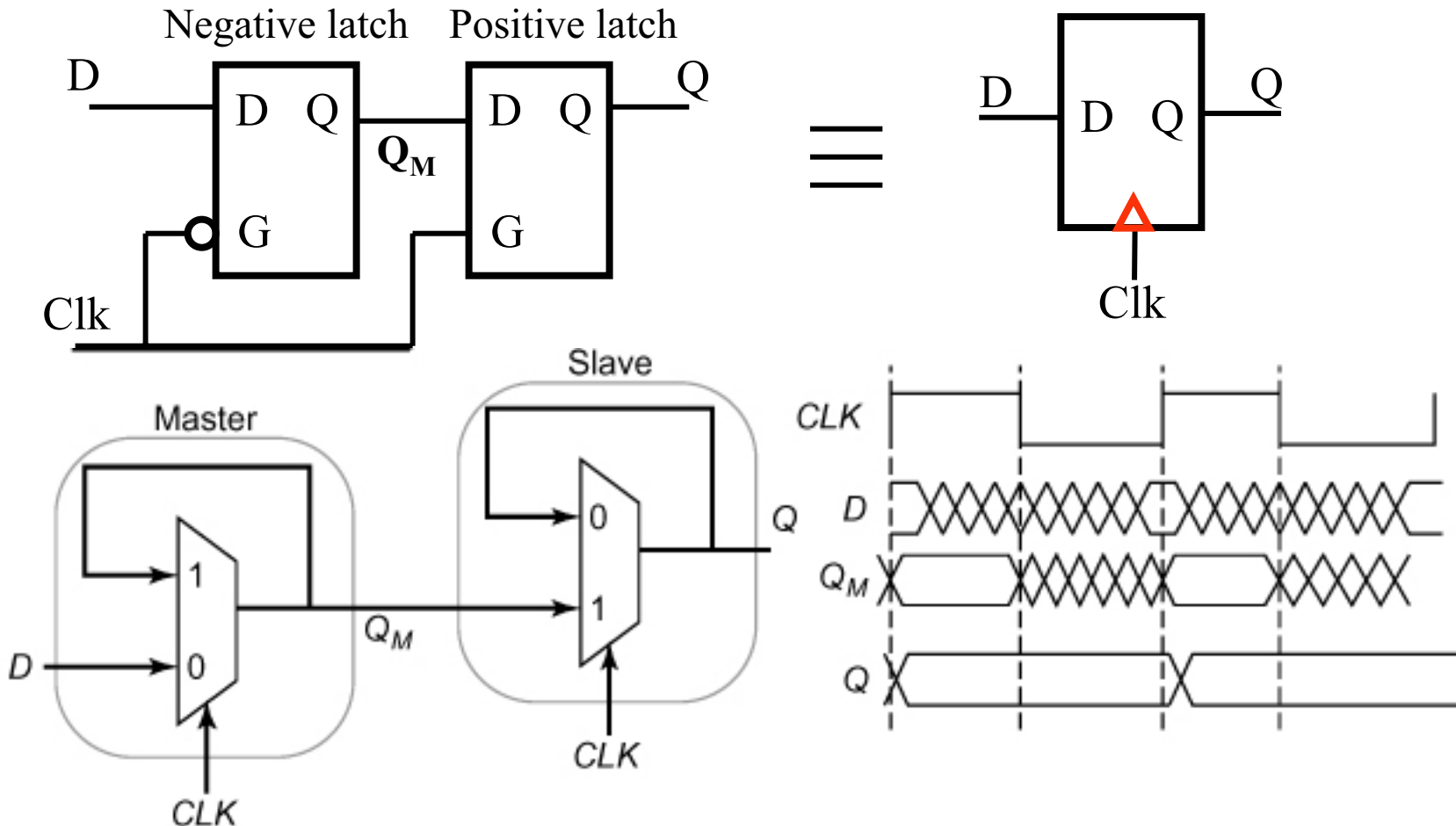
Positive Latch



Negative Latch



Building an Edge-Triggered Register



■ Master-Slave Register

- Use negative clock phase to latch inputs into first latch
- Use positive clock to change outputs with second latch

■ View pair as one basic unit

- master-slave flip-flop twice as much logic

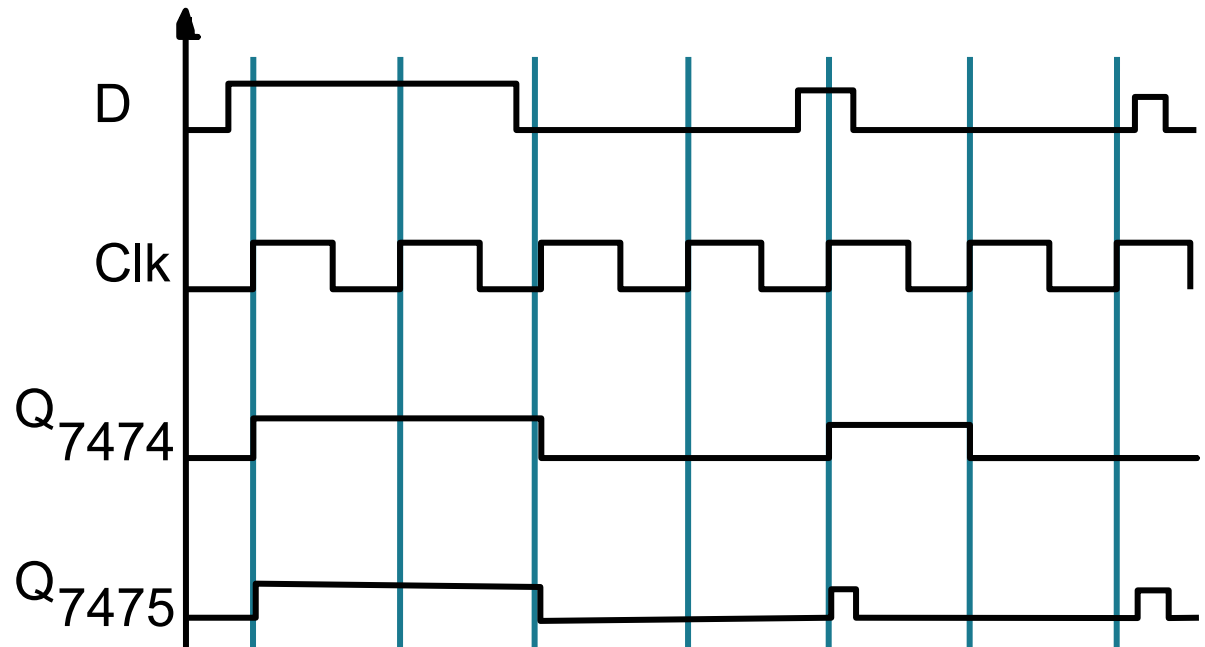
Latches vs. Edge-Triggered Register



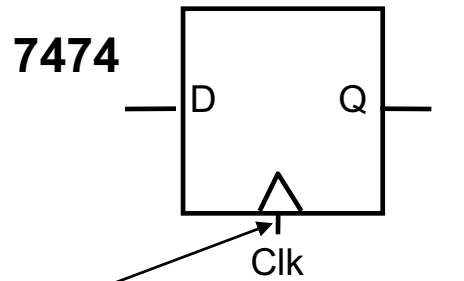
Edge triggered device sample inputs on the event edge

Transparent latches sample inputs as long as the clock is asserted

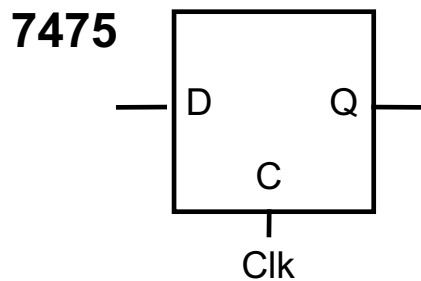
Timing Diagram:



Behavior the same unless input changes while the clock is high



Positive edge-triggered register

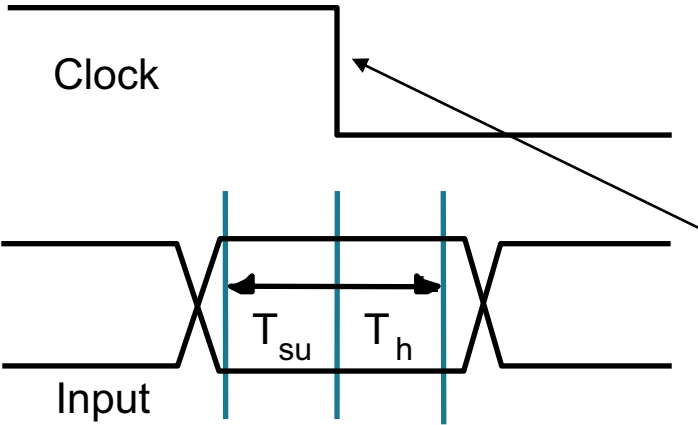


Level-sensitive latch

Bubble here for negative edge triggered register



Important Timing Parameters



Clock:
 Periodic Event, causes state of memory element to change

memory element can be updated on the:
 rising edge, falling edge, high level, low level

Setup Time (T_{su})
 Minimum time before the clocking event by which the input must be stable

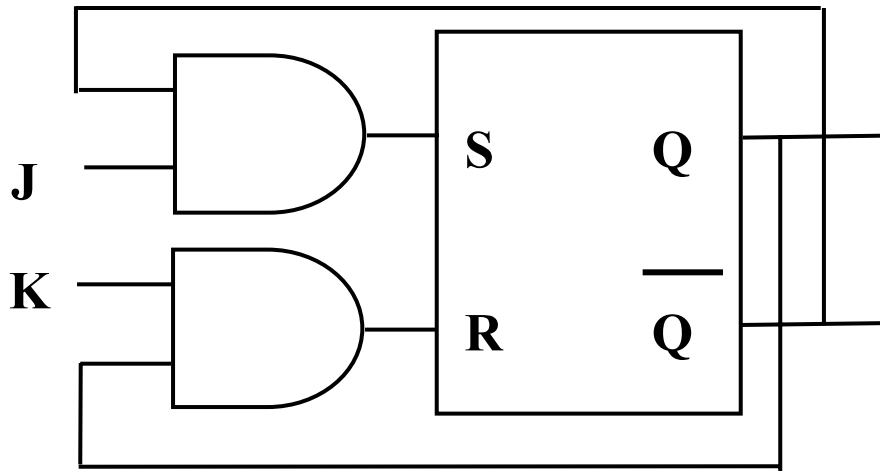
Hold Time (T_h)
 Minimum time after the clocking event during which the input must remain stable

Propagation Delay (T_{cq} for an edge-triggered register and T_{dq} for a latch)
 Delay overhead of the memory element

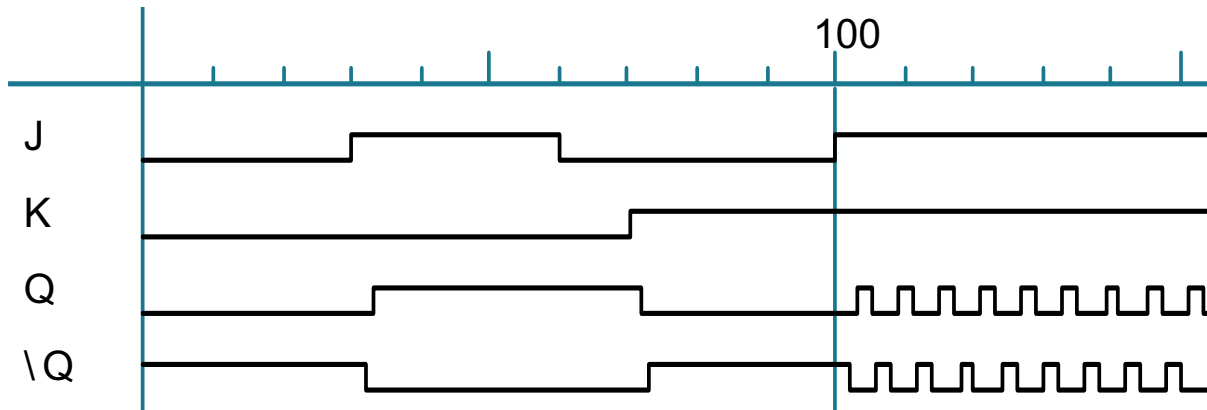
There is a timing "window" around the clocking event during which the input must remain stable and unchanged in order to be recognized



The J-K Flip-Flop



J	K	Q+	\overline{Q} +
0	0	Q	\overline{Q}
0	1	0	1
1	0	1	0
1	1	\overline{Q}	Q



- Eliminate the forbidden state of the SR Flip-flop
- Use output feedback to guarantee that R and S are never both one

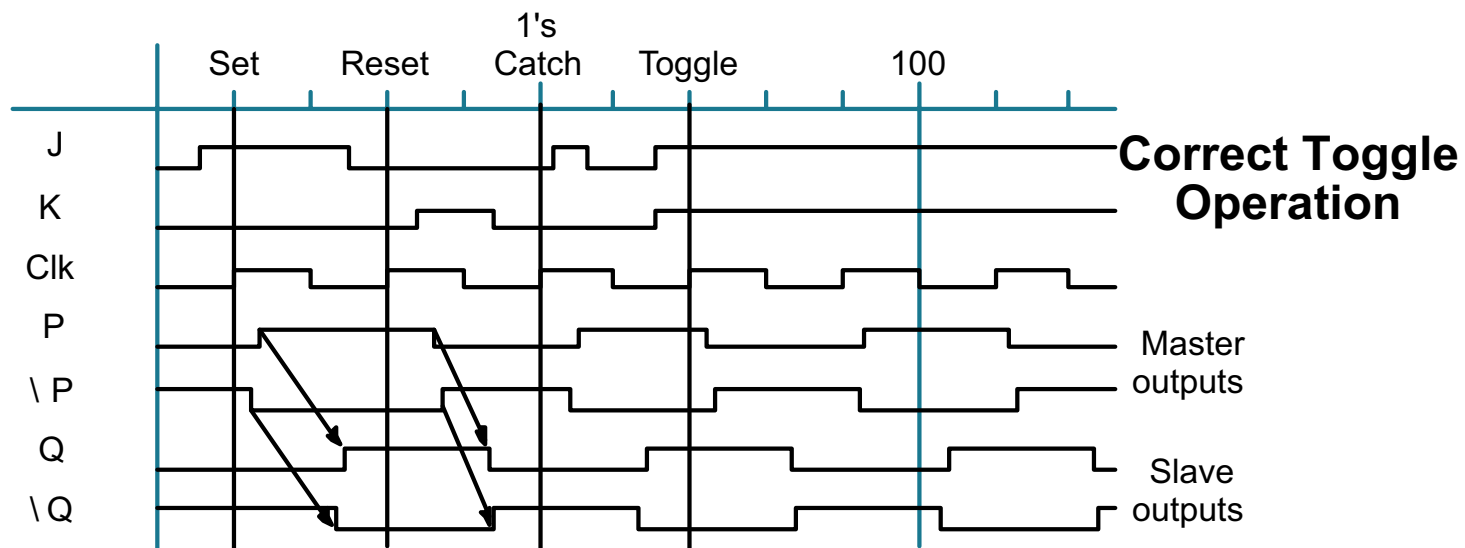
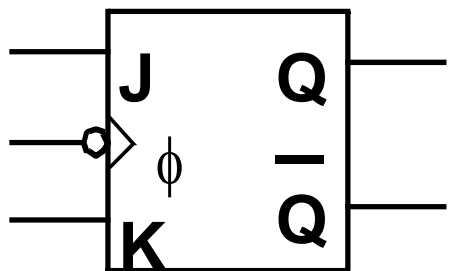
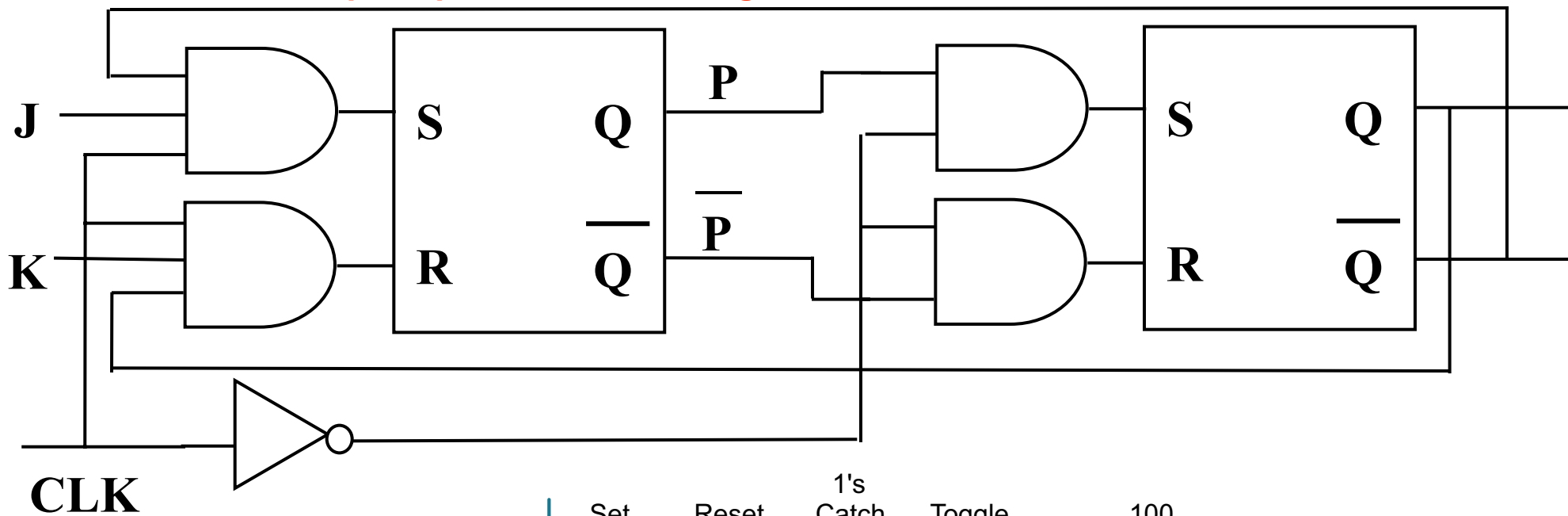


J-K Master-Slave Register



Sample inputs while clock high

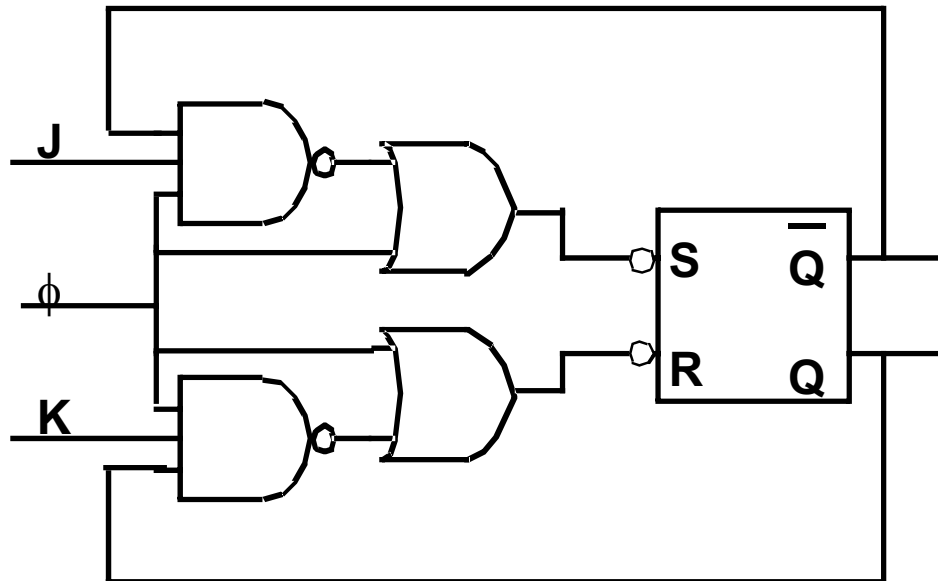
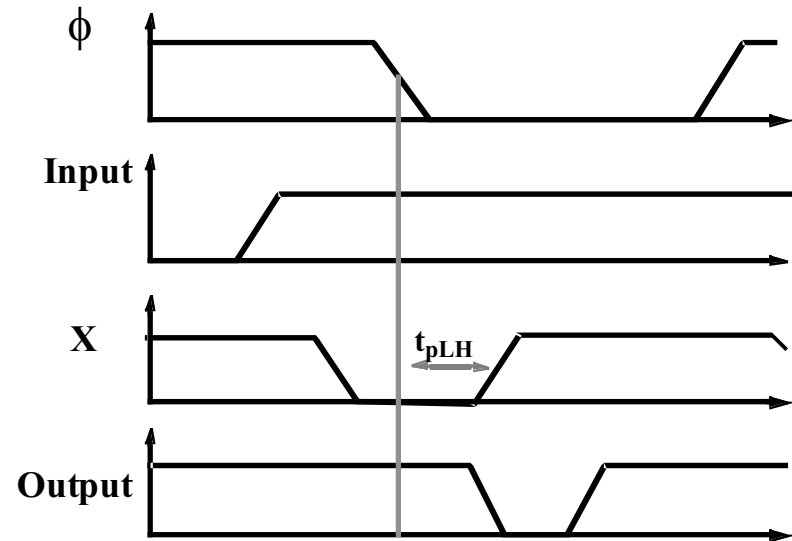
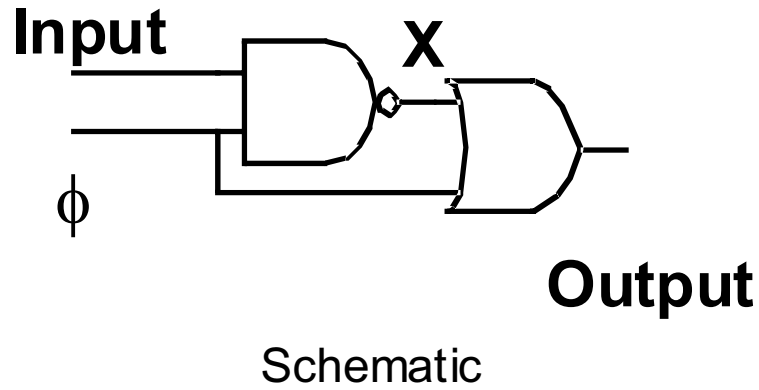
Sample inputs while clock low



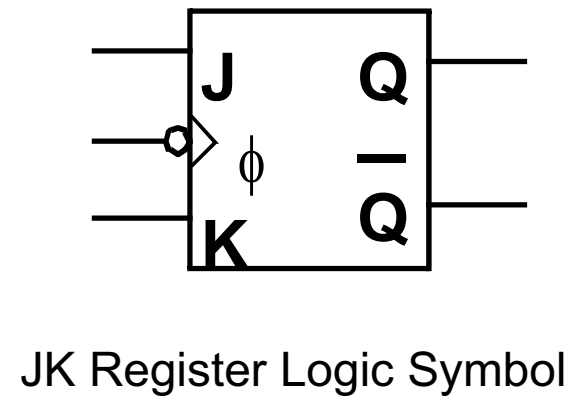
Is there a problem with this circuit?



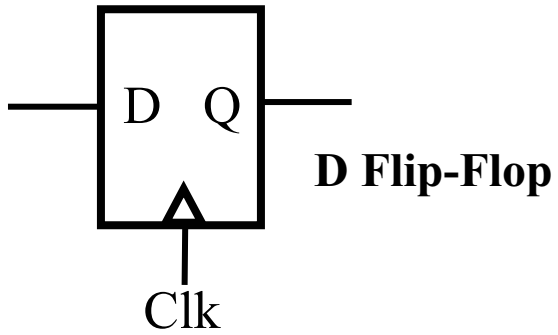
Pulse Based Edge-Triggered J-K Register



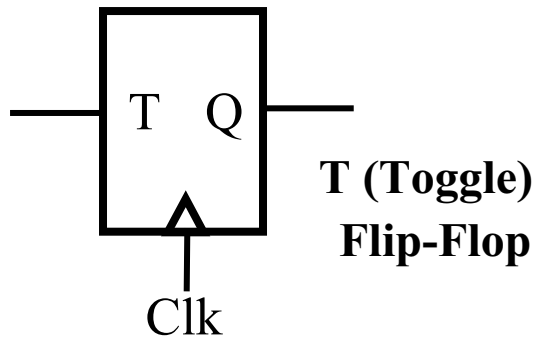
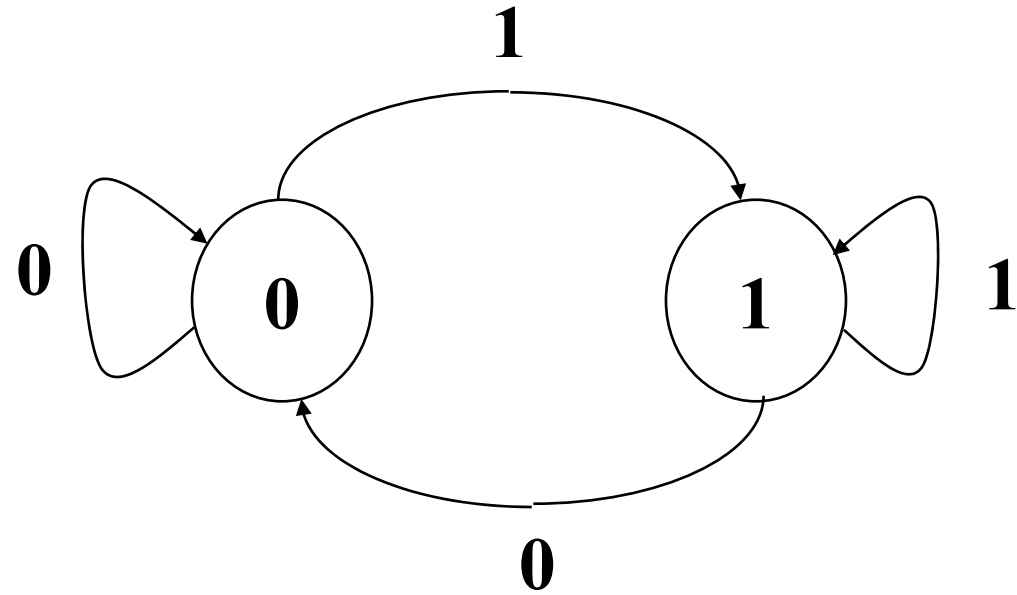
JK Register Schematic



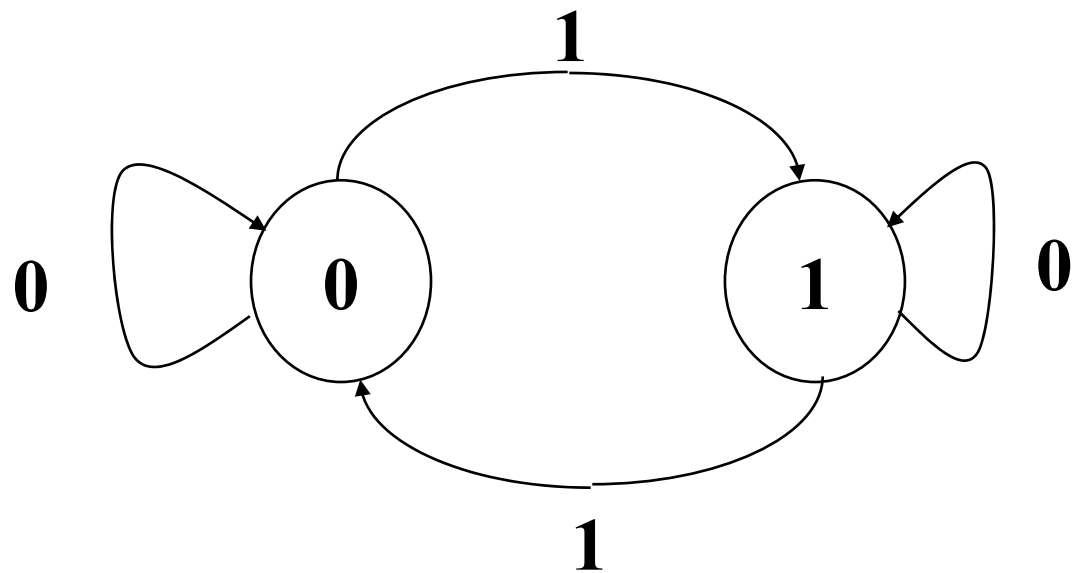
D Flip-Flop vs. Toggle Flip-Flop



D	Q_N
0	0
1	1



T	Q_N
0	Q_{N-1}
1	$\overline{Q_{N-1}}$





Realizing different types of memory elements



Characteristic Equations

D: $Q^+ = D$

J-K: $Q^+ = J \bar{Q} + \bar{K} Q$

T: $Q^+ = T \bar{Q} + \bar{T} Q$

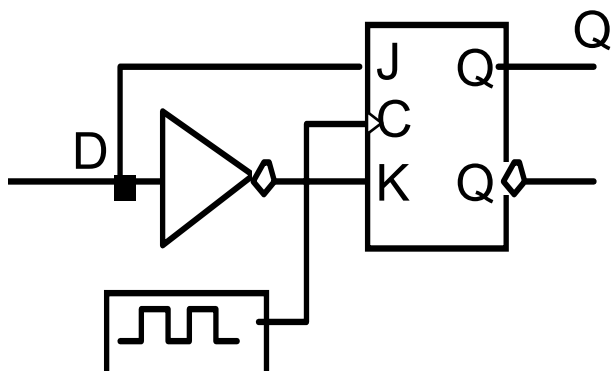
E.g., J=K=0, then $Q^+ = Q$

J=1, K=0, then $Q^+ = 1$

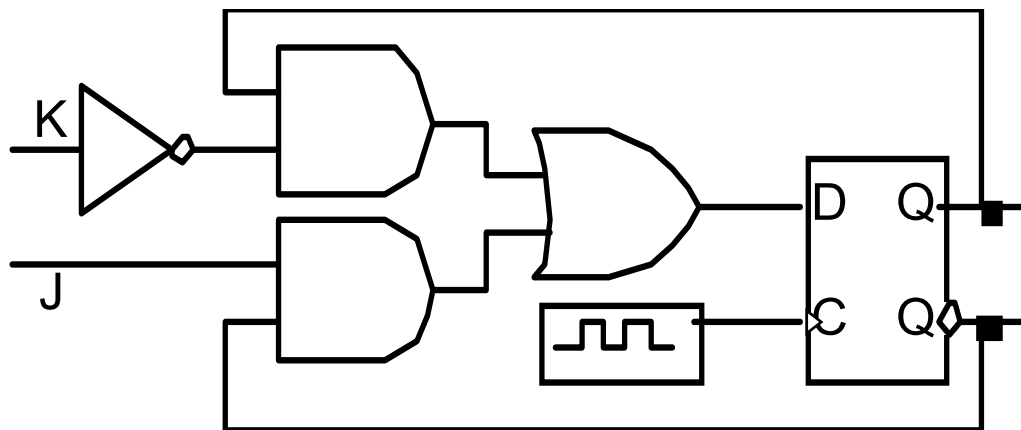
J=0, K=1, then $Q^+ = \bar{Q}$

J=1, K=1, then $Q^+ = \bar{Q}$

Implementing One FF in Terms of Another



D implemented with J-K



J-K implemented with D



Design Procedure



Excitation Tables: What are the necessary inputs to cause a particular kind of change in state?

Q	Q ⁺	J	K	T	D
0	0	0	X	0	0
0	1	1	X	1	1
1	0	X	1	1	0
1	1	X	0	0	1

Implementing D FF with a J-K FF:

- 1) Start with K-map of $Q^+ = f(D, Q)$
- 2) Create K-maps for J and K with same inputs (D, Q)
- 3) Fill in K-maps with appropriate values for J and K to cause the same state changes as in the original K-map

	D	0	1
Q	0	0	1
	1	0	1

$Q^+ = D$

E.g., $D = Q = 0$, $Q^+ = 0$
then $J = 0$, $K = X$

	D	0	1
Q	0	0	1
	1	X	X

$J = D$

	D	0	1
Q	0	X	X
	1	1	0

$K = \bar{D}$



Design Procedure (cont.)



Implementing J-K FF with a D FF:

1) K-Map of $Q^+ = F(J, K, Q)$

2,3) Revised K-map using D's excitation table
its the same! that is why design procedure with D FF is simple!

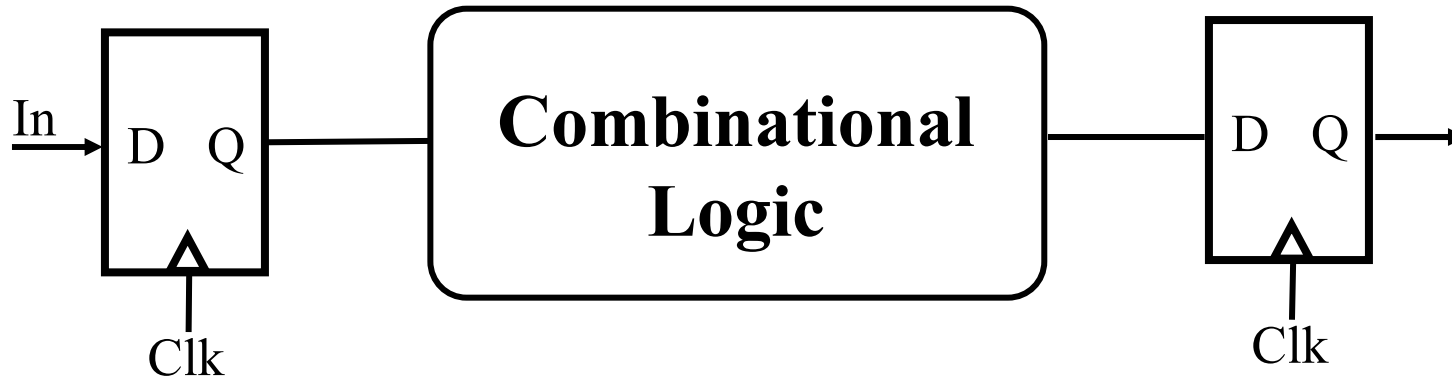
	JK		J	
Q	00	01	11	10
0	0	0	1	1
1	1	0	0	1
			K	

$$Q^+ = D = J\bar{Q} + \bar{K}Q$$

Resulting equation is the combinational logic input to D to cause same behavior as J-K FF. Of course it is identical to the characteristic equation for a J-K FF.



System Timing Parameters



Register Timing Parameters

T_{cq} : worst case rising edge
clock to q delay

$T_{cq, cd}$: contamination or
minimum delay from
clock to q

T_{su} : setup time

T_h : hold time

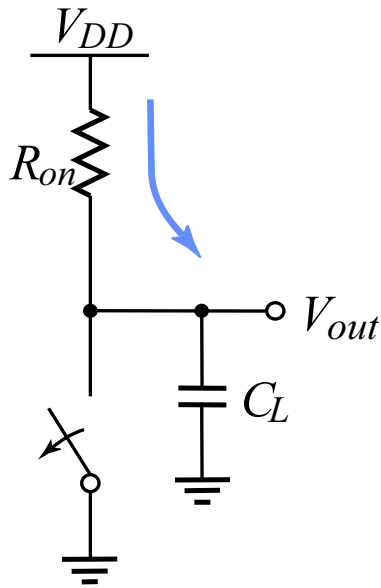
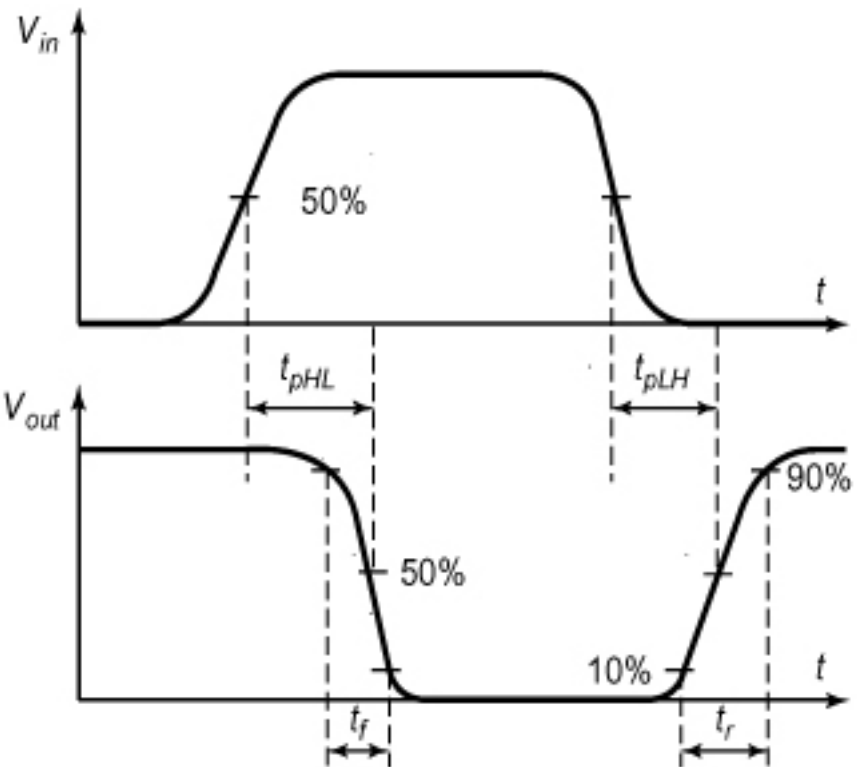
Logic Timing Parameters

T_{logic} : worst case delay
through the combinational
logic network

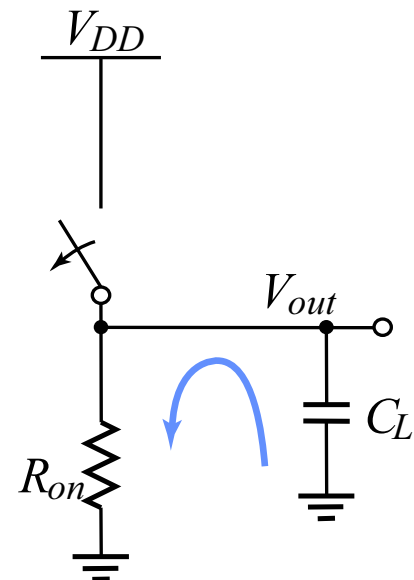
$T_{logic, cd}$: contamination or
minimum delay
through logic network



Delay in Digital Circuits

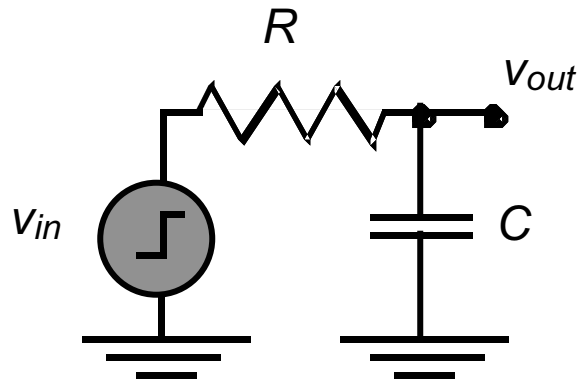


(a) Low-to-high



(b) High-to-low

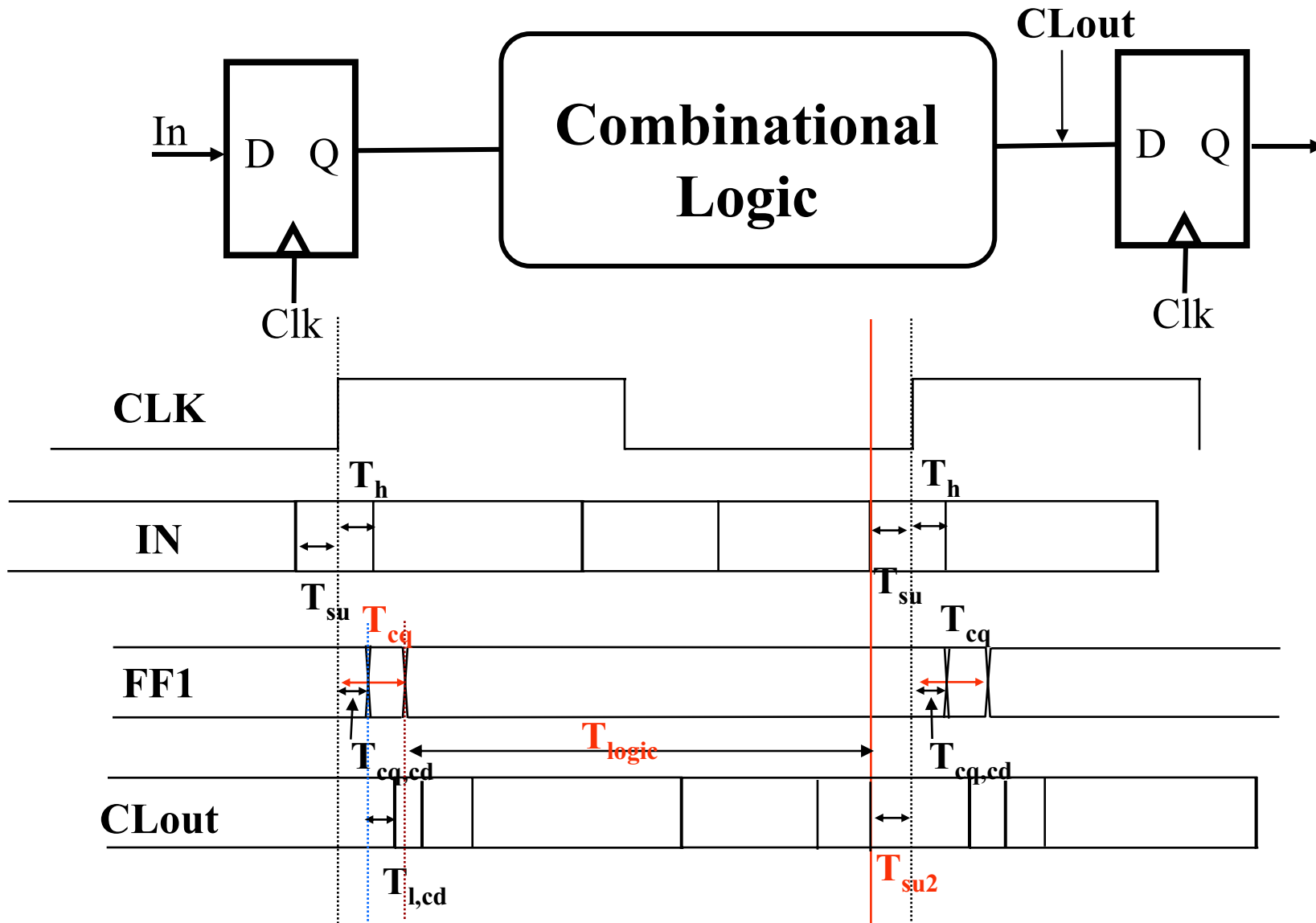
review



$$v_{out}(t) = (1 - e^{-t/\tau}) V$$

$$t_p = \ln(2) \tau = 0.69 RC$$

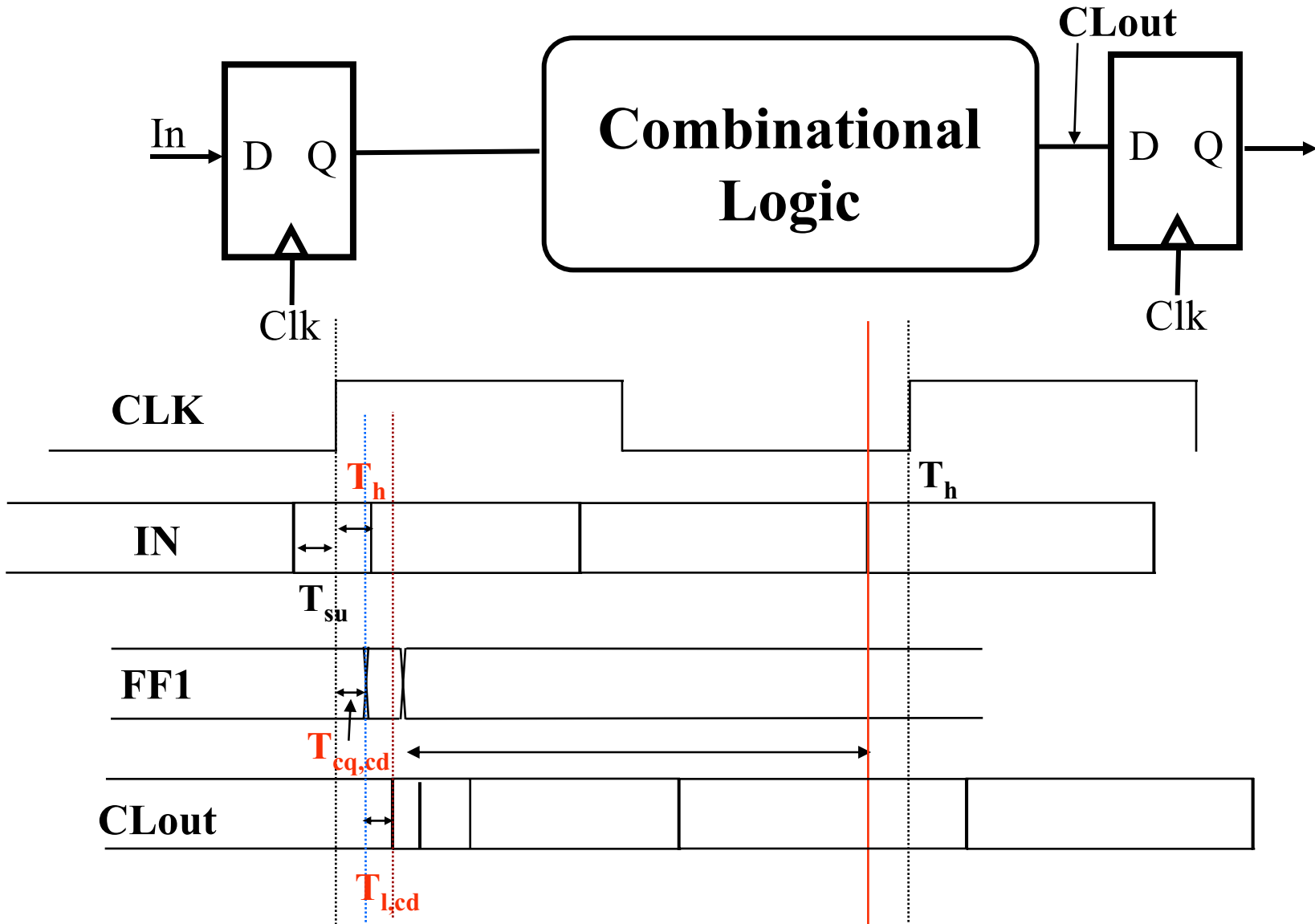
System Timing (I): Minimum Period



$$T > T_{cq} + T_{logic} + T_{su}$$



System Timing (II): Minimum Delay



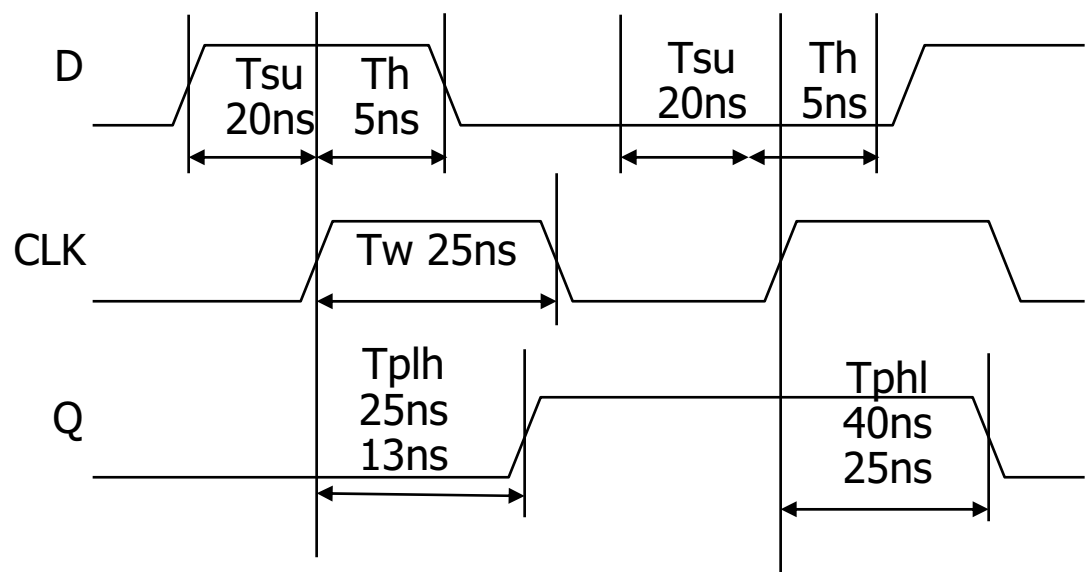
$$T_{cq,cd} + T_{logic,cd} > T_{hold}$$



Shift-Register

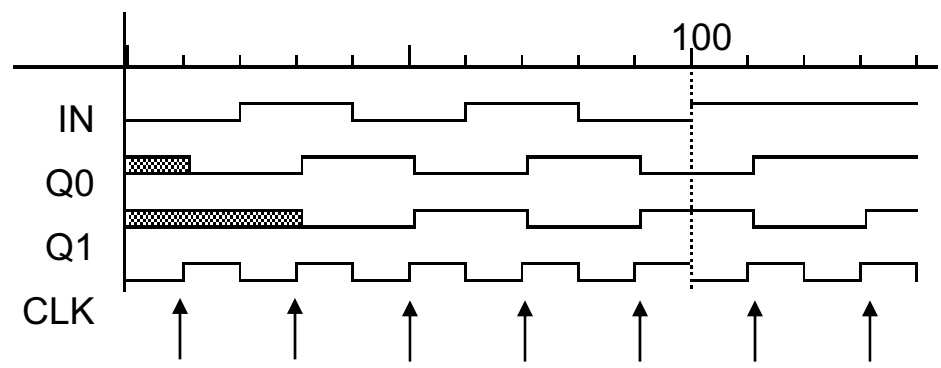
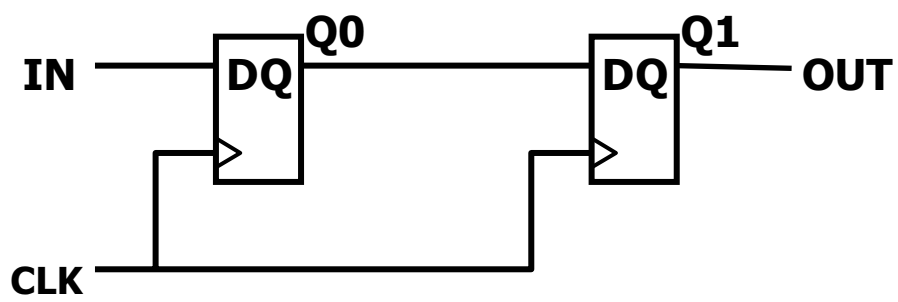


■ Typical parameters for Positive edge-triggered D Register

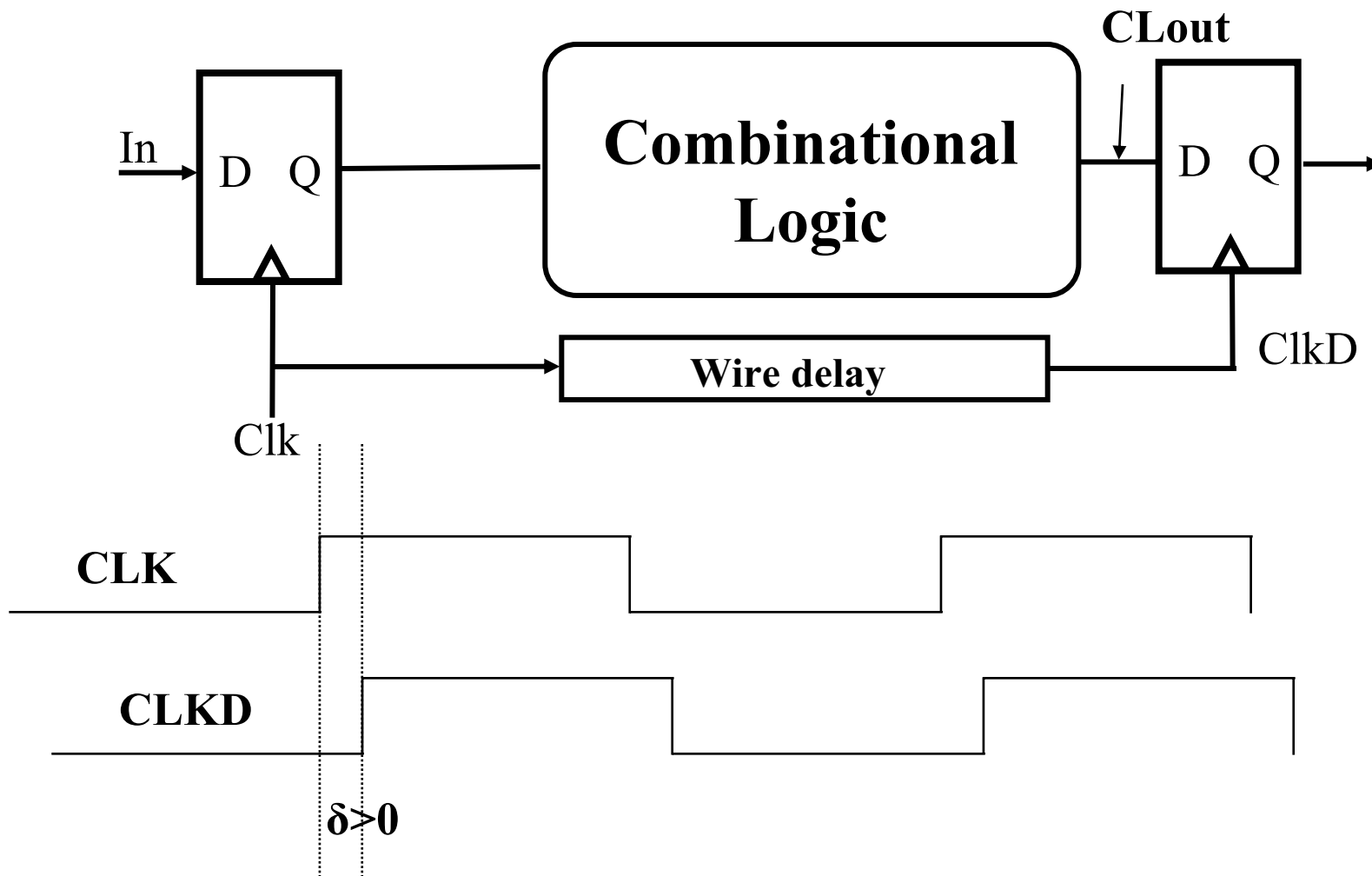


all measurements are made from the clocking event that is, the rising edge of the clock

■ Shift-register



Clocks are not perfect: Clock Skew



$$T > T_{cq} + T_{logic} + T_{su} - \delta$$

$$T_{cq,cd} + T_{logic,cd} > T_{hold} + \delta$$