L4: Sequential Building Blocks

(Flip-flops, Latches and Registers)

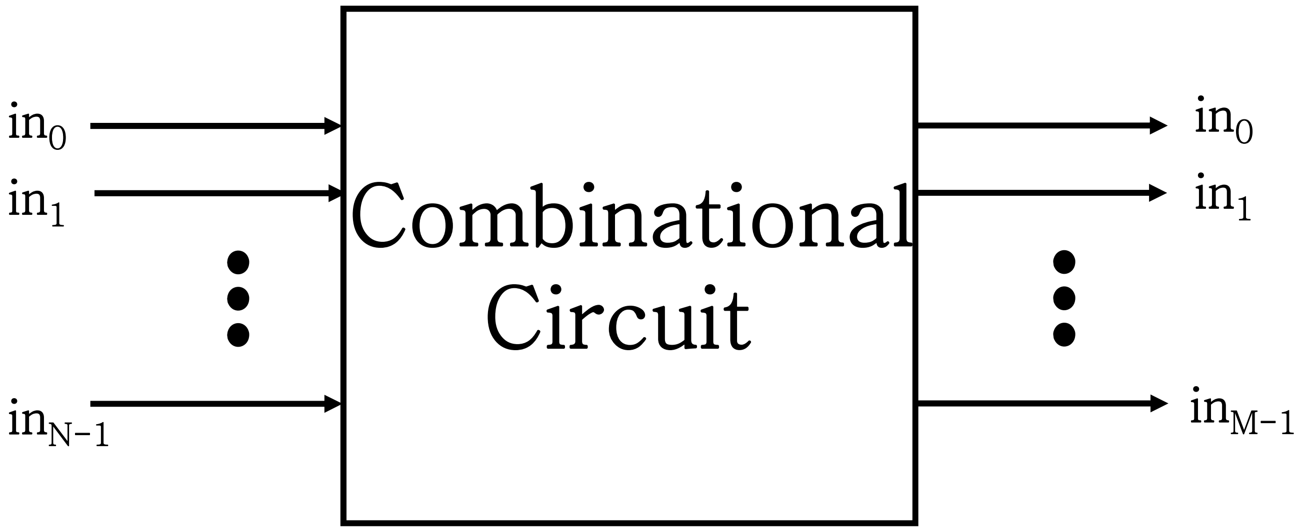
Acknowledgements:

- **Lecture material adapted from R. Katz, G. Borriello, "Contemporary Logic Design" (second edition), Prentice-Hall/Pearson Education, 2005.**

- **Lecture material adapted from J. Rabaey, A. Chandrakasan, B. Nikolic, "Digital Integrated Circuits: A Design Perspective" Copyright 2003 Prentice Hall/Pearson.**

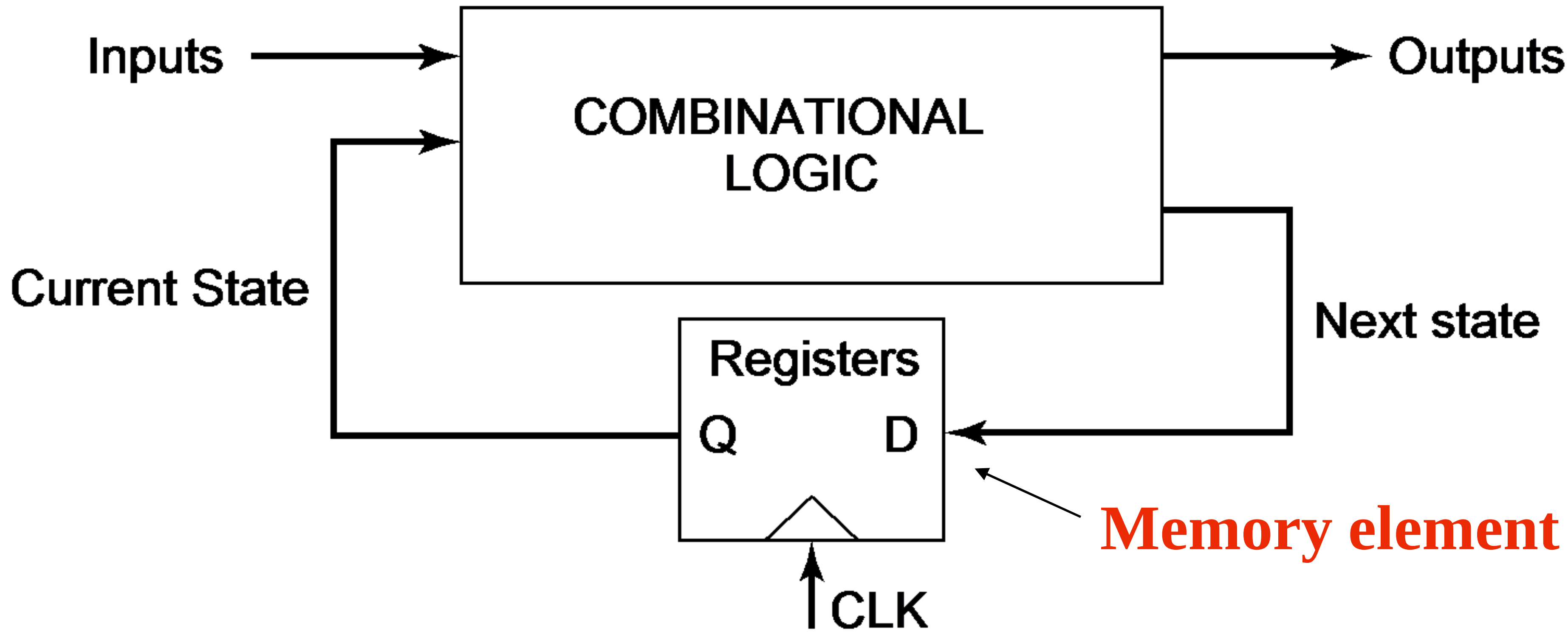
- **Lecture notes prepared by Professor Anantha Chandrakasan**

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

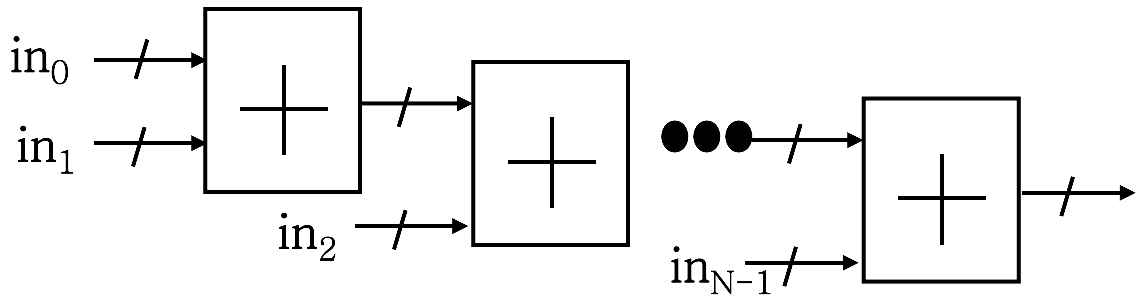
A Sequential System



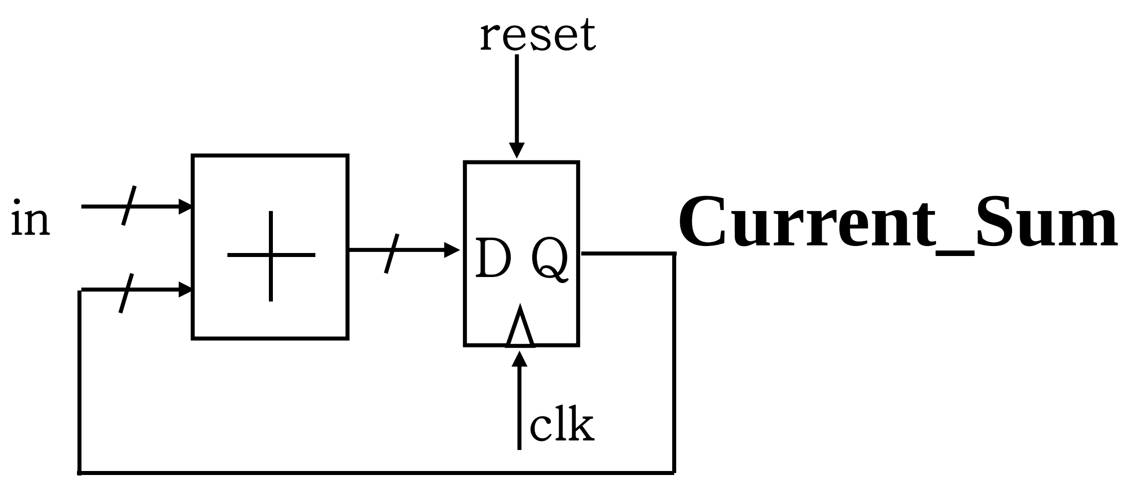
- Sequential circuits have memory (i.e., remember the past)
- The current state is “held” in memory and the next state is computed based the current state and the current inputs
- In a synchronous systems, the clock signal orchestrates the sequence of events

A Simple Example

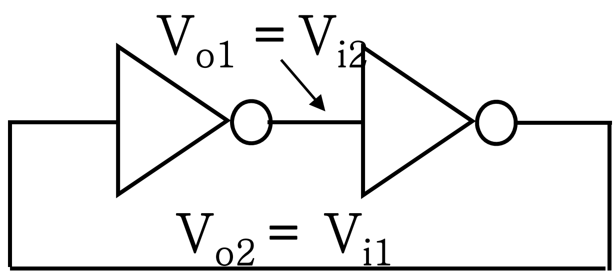
Adding N inputs (N-1 Adders)

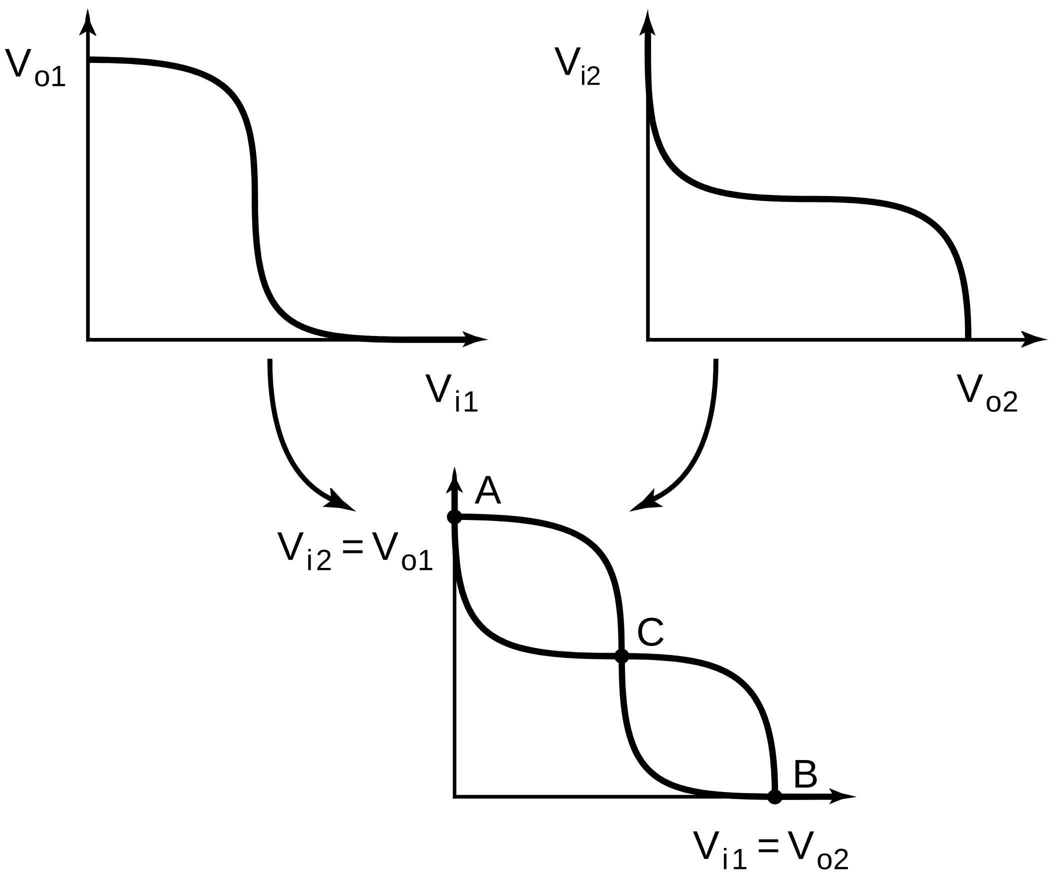


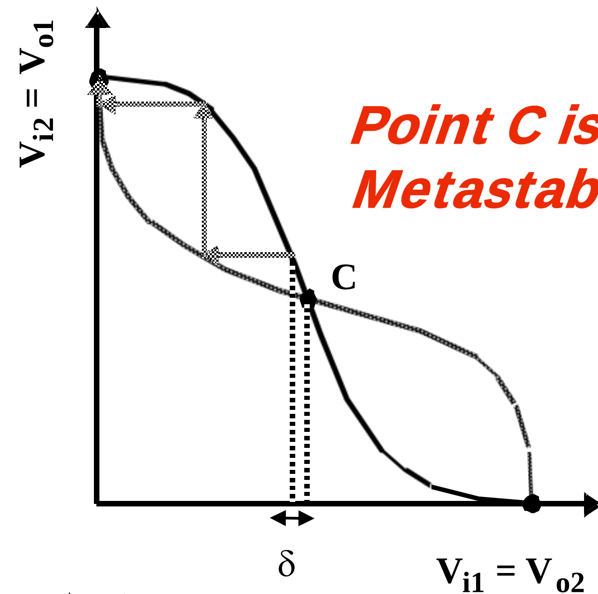
Using a sequential (serial) approach



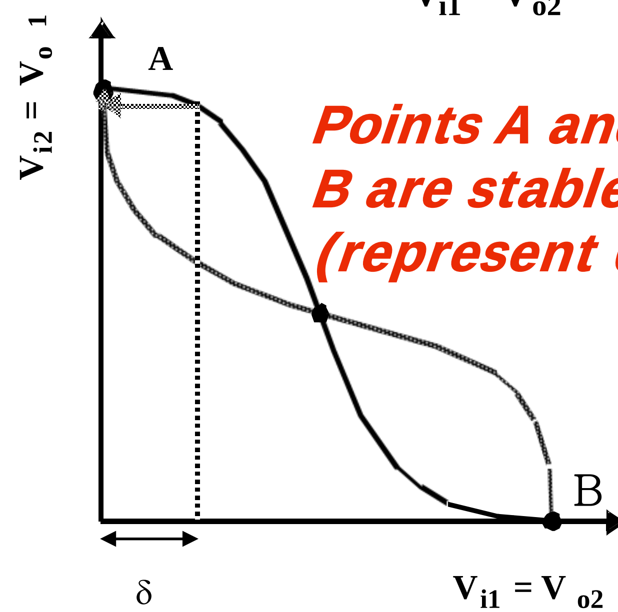
Implementing State: Bi-stability





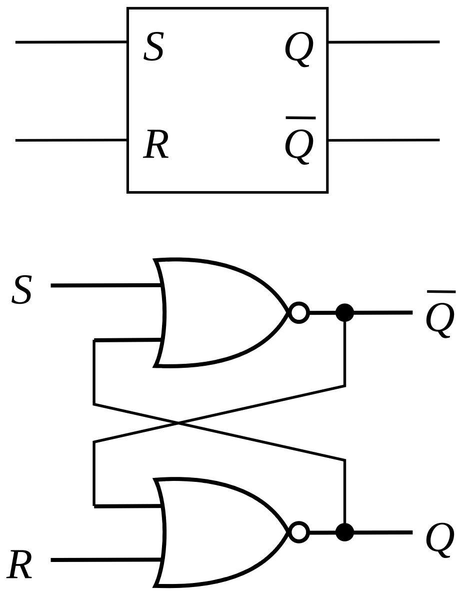


Point C is Metastable



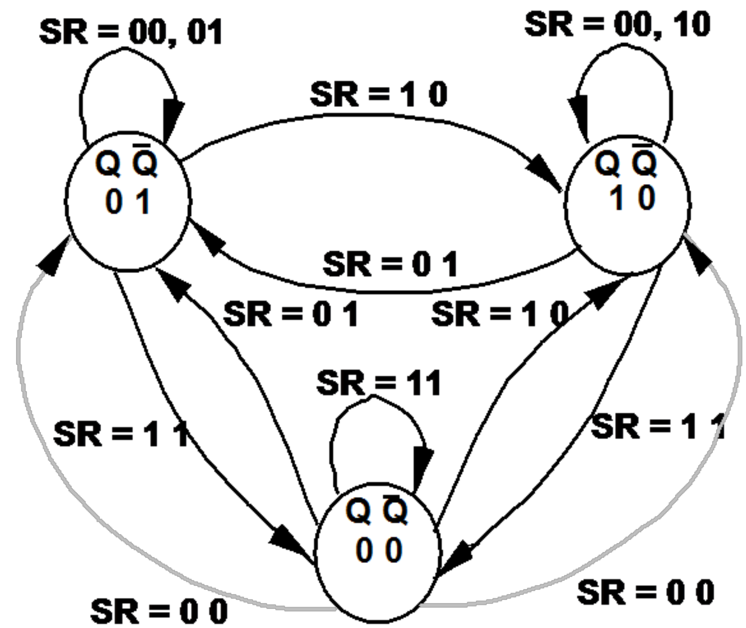
Points A and B are stable (represent 0 & 1)

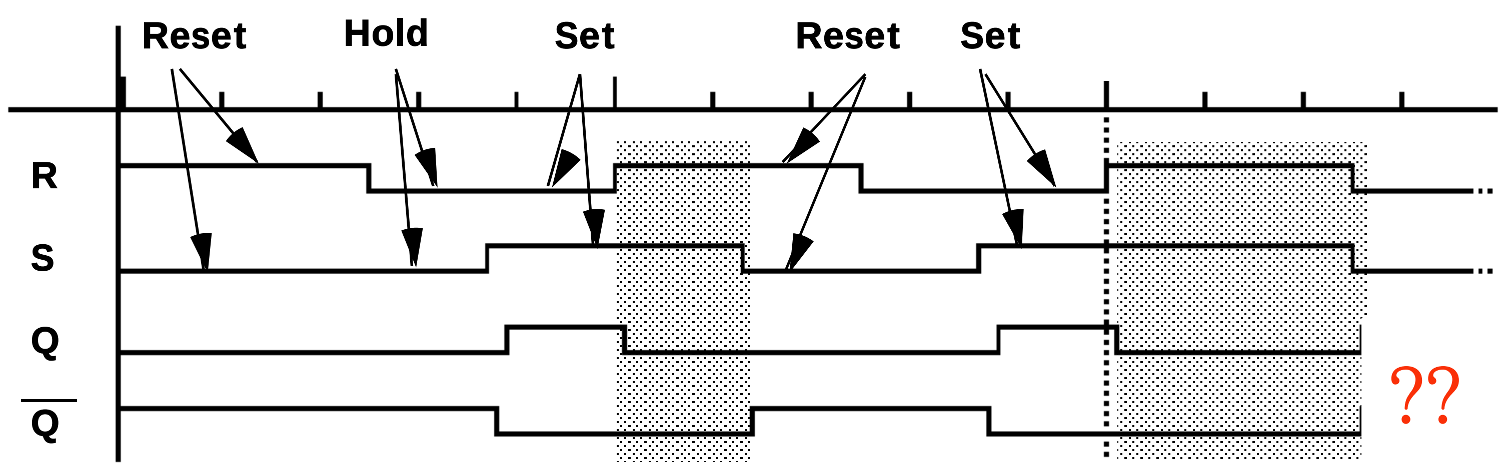
NOR-based Set-Reset (SR) Flipflop



S	R	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
1	0	1	0
0	1	0	1
1	1	0	0

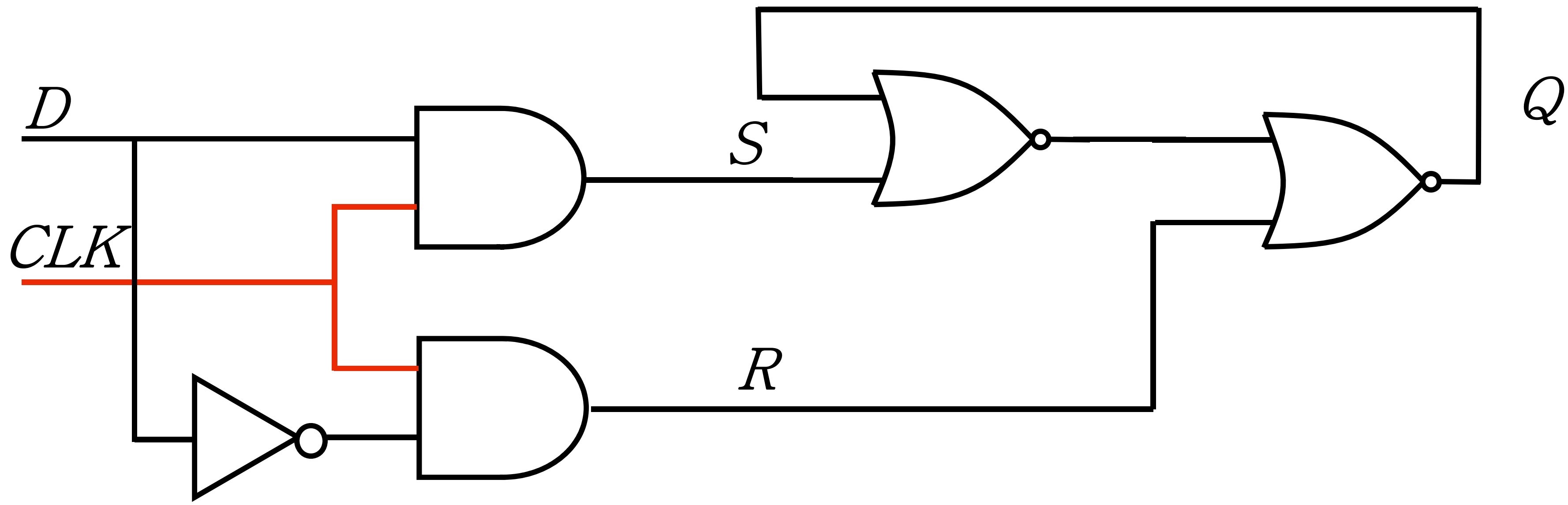
Forbidden State

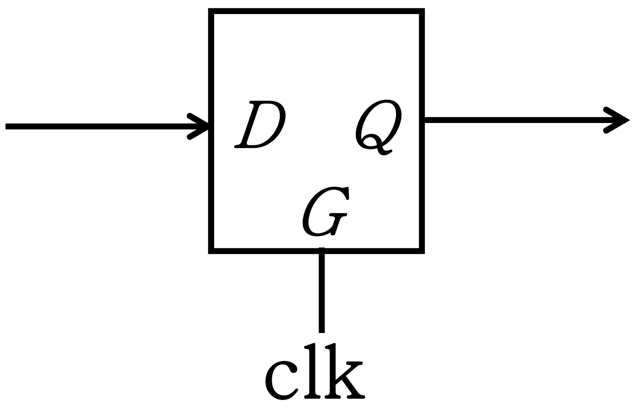


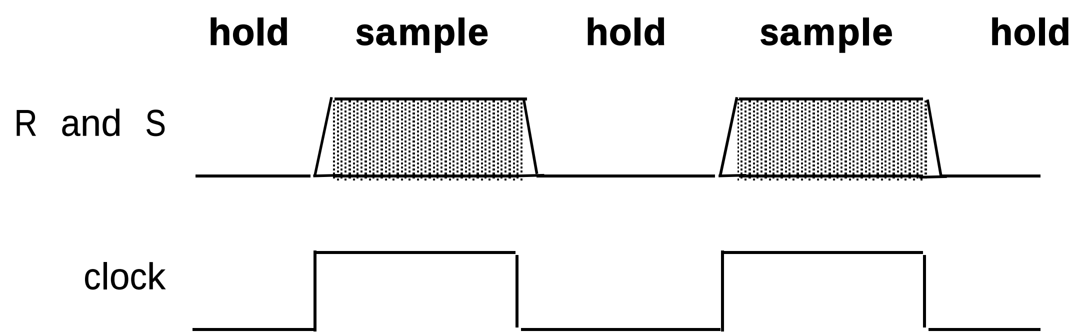


- **Flip-flop refers to a bi-stable element (edge-triggered registers are also called flip-flops) – this circuit is not clocked and outputs change "asynchronously" with the inputs**

Making a Clocked Memory Element: Positive D-Latch



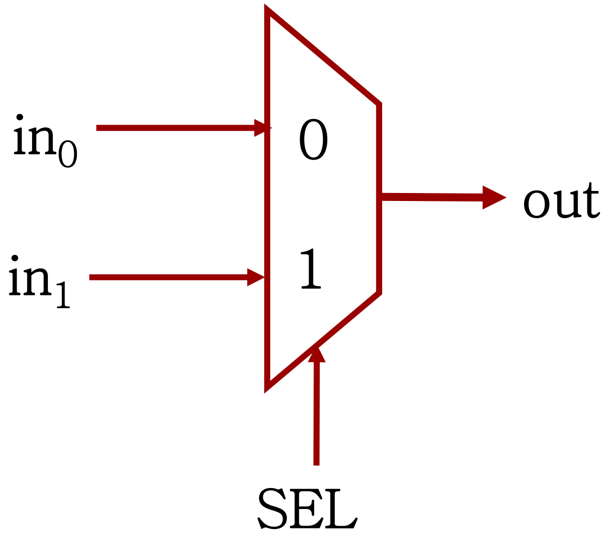




- **A Positive D-Latch:** **Passes input D to output Q when CLK is high and holds state when clock is low (i.e., ignores input D)**
- **A Latch is level-sensitive:** **invert clock for a negative latch**

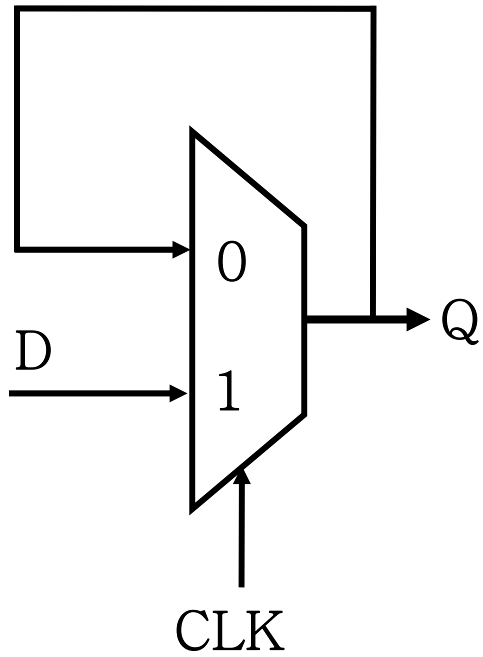
Multiplexer Based Positive & Negative Latch

2:1 multiplexer

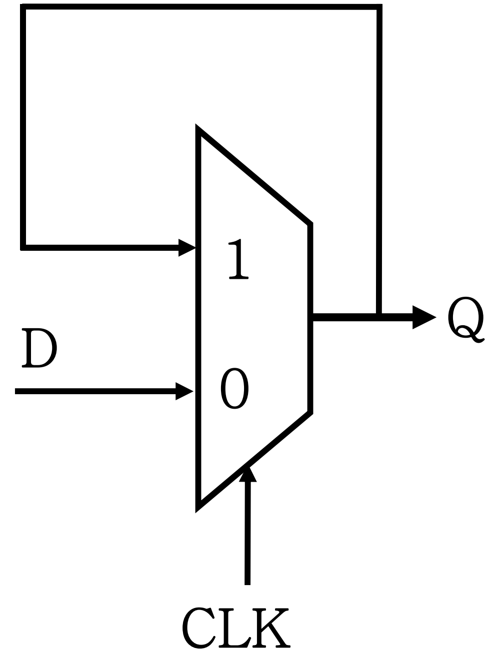


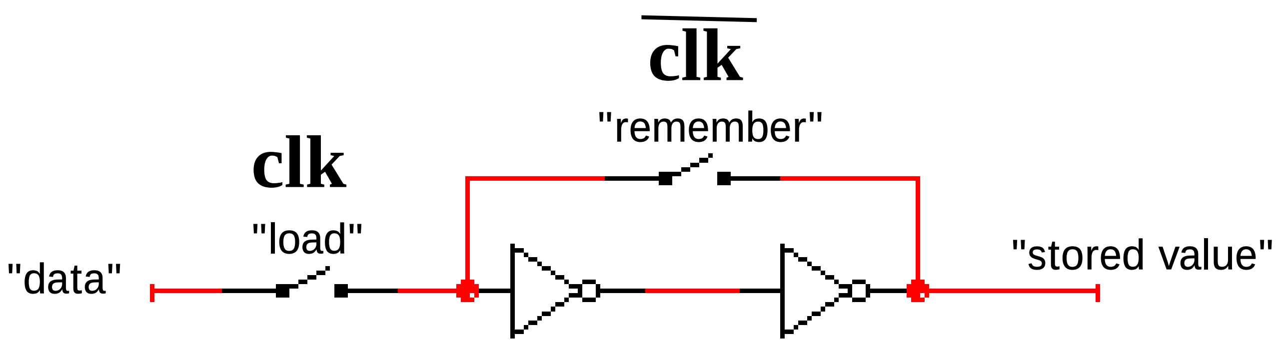
$$Out = sel * in_1 + \overline{sel} * in_0$$

Positive Latch

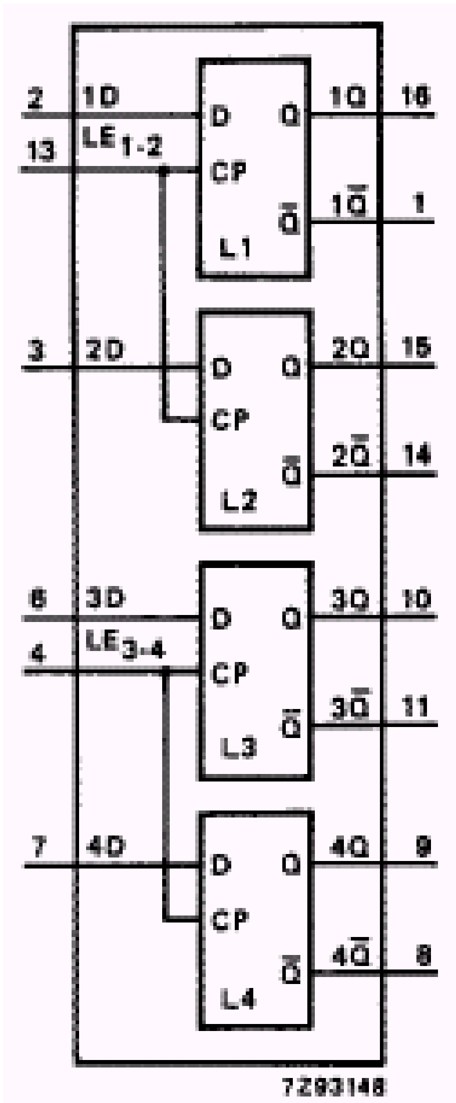


Negative Latch



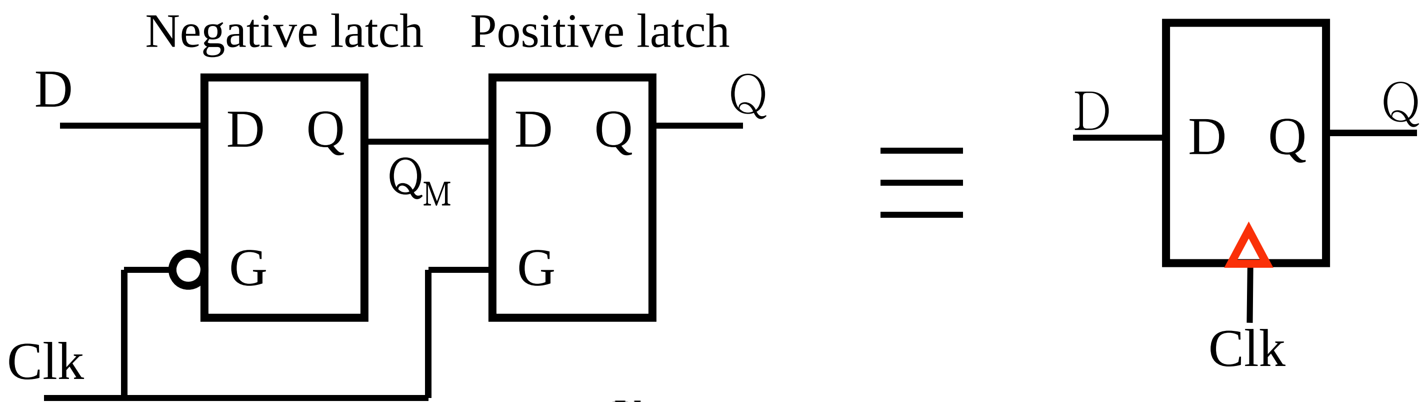


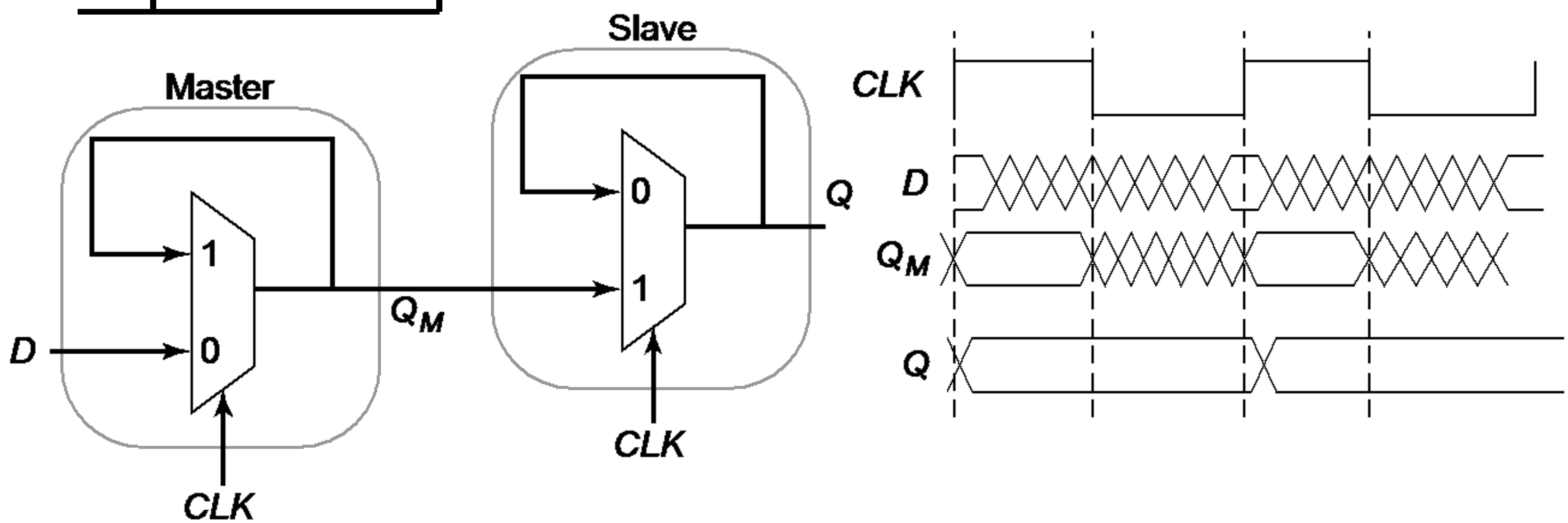
74HC75 (Positive Latch)



OPERATING MODES	INPUTS		OUTPUTS	
	LE_{n-n}	nD	nQ	$n\overline{Q}$
data enabled	H	L	L	H
	H	H	H	L
data latched	L	X	q	$\overline{q}$

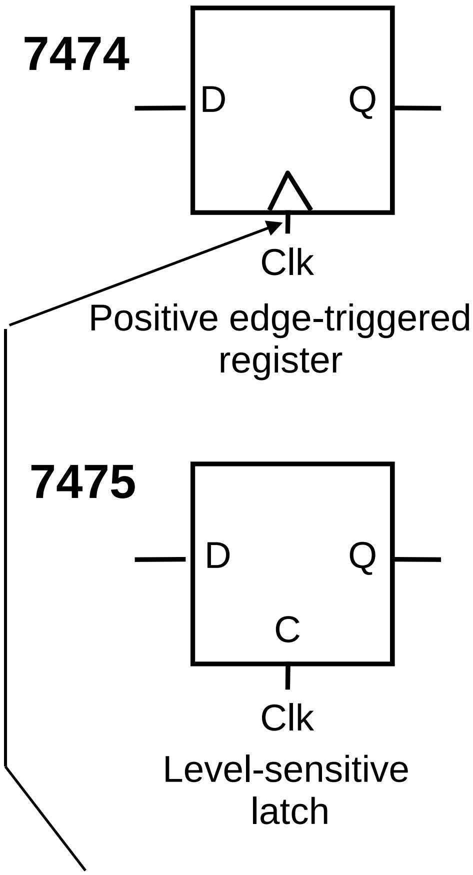
Building an Edge-Triggered Register





- **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

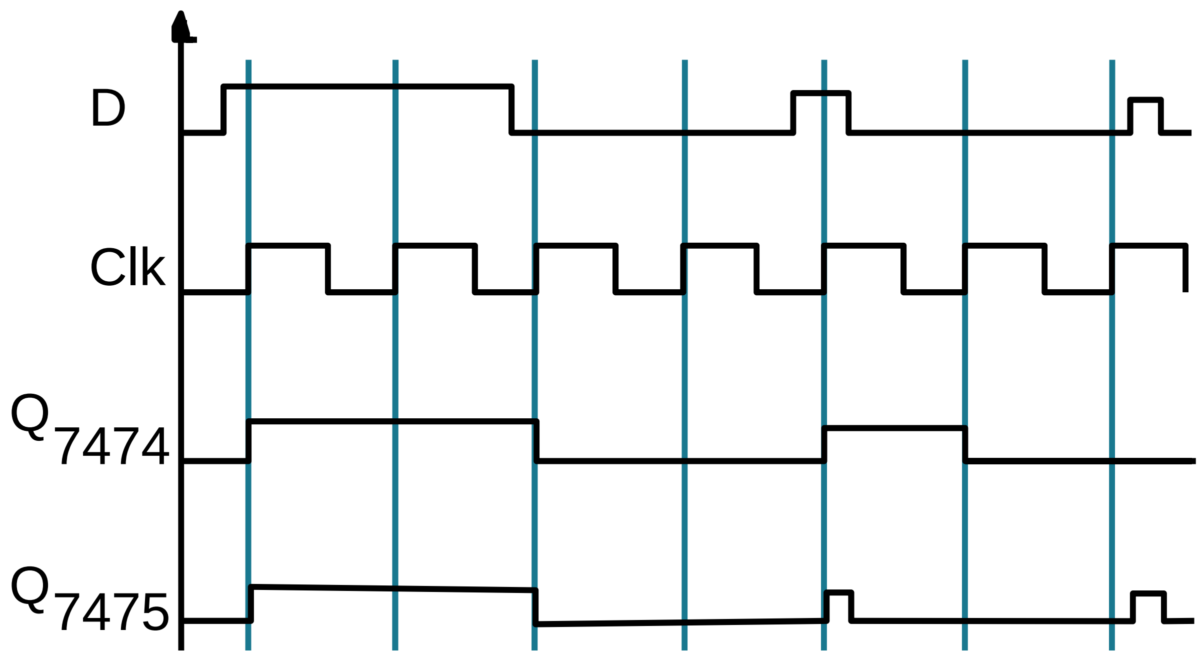


Bubble here for negative edge triggered register

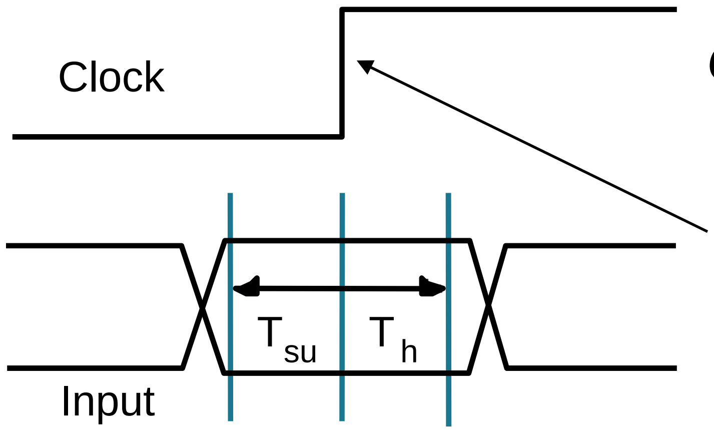
Edge triggered **device sample inputs on the event edge**

Transparent latches **sample inputs as long as the clock is asserted**

Timing Diagram:



Behavior the same unless input changes while the clock is high



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)

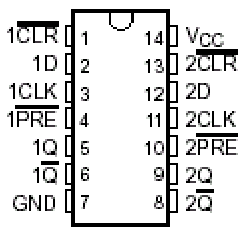
Minimum time after the clocking event during which the input must remain stable

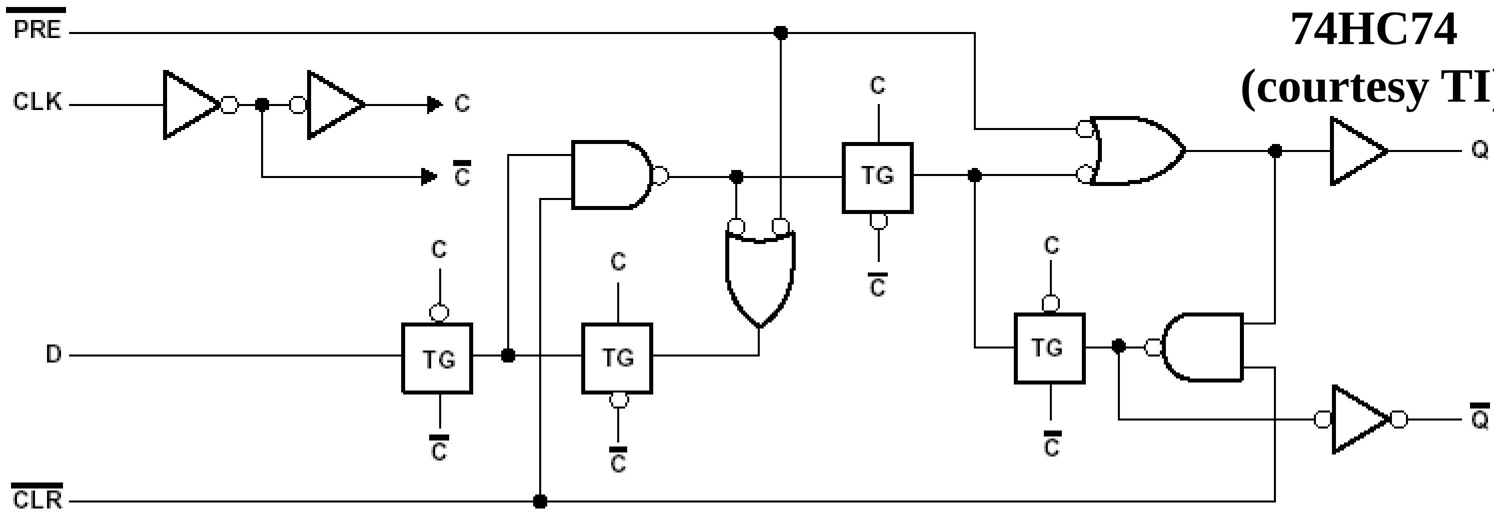
Propagation Delay (T_{cq} for an edge-triggered register and T_{dq} for a latch)

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

74HC74 (Positive Edge-Triggered Register)





74HC74 (courtesy TI)

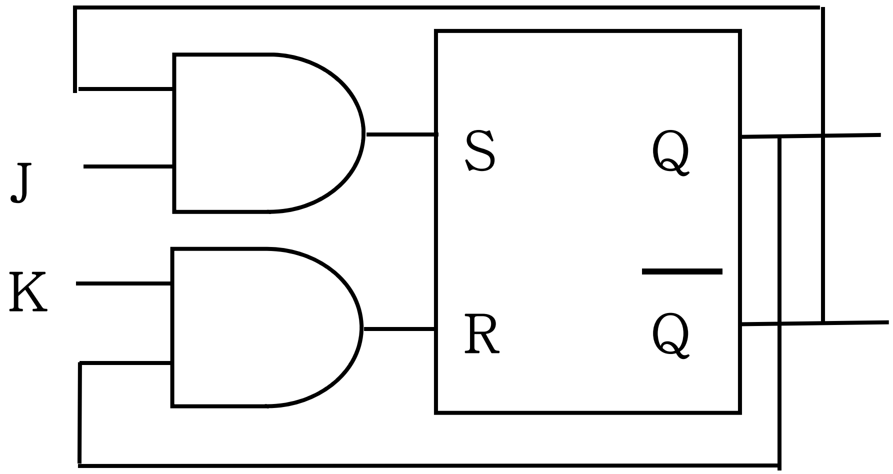
FUNCTION TABLE

INPUTS				OUTPUTS	
$\overline{PRE}$	$\overline{CLR}$	CLK	D	Q	$\overline{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H†	H†
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q_0	$\overline{Q}_0$

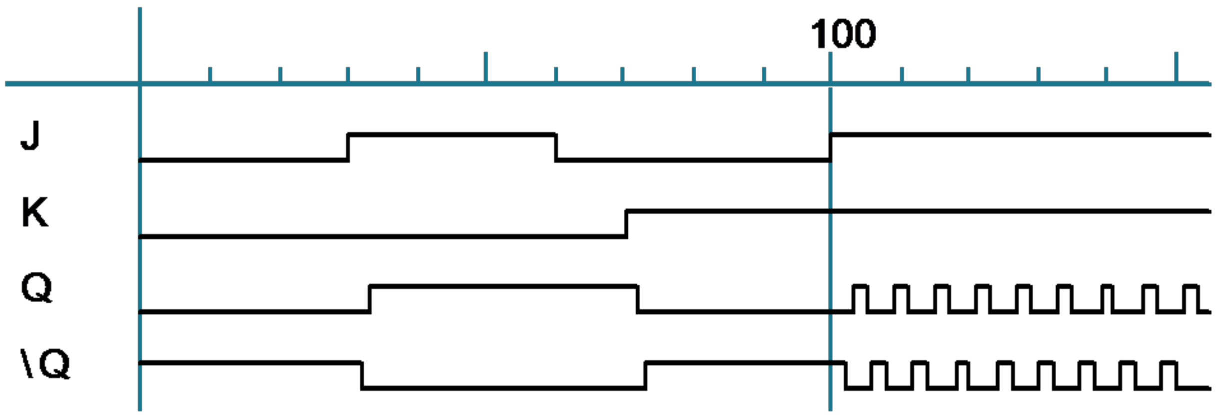
D-FF with preset and clear

			V_{CC}	T_A = 25°C MIN	T_A = 25°C MAX	SN54HC74 MIN	SN54HC74 MAX	SN74HC74 MIN	SN74HC74 MAX	UNIT
f_{clock}	Clock frequency		2 V	0	6	0	4.2	0	5	MHz
			4.5 V	0	31	0	21	0	25	
			6 V	0	36	0	25	0	29	
t_w	Pulse duration	$\overline{PRE}$ or $\overline{CLR}$ low	2 V	100		150		125		ns
			4.5 V	20		30		25		
			6 V	17		25		21		
		CLK high or low	2 V	80		120		100		
			4.5 V	16		24		20		
			6 V	14		20		17		
t_{su}	Setup time before CLK↑	Data	2 V	100		150		125		ns
			4.5 V	20		30		25		
			6 V	17		25		21		
		$\overline{PRE}$ or $\overline{CLR}$ inactive	2 V	25		40		30		
			4.5 V	5		8		6		
			6 V	4		7		5		
t_h	Hold time, data after CLK↑		2 V	0		0		0		ns
			4.5 V	0		0		0		
			6 V	0		0		0		

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

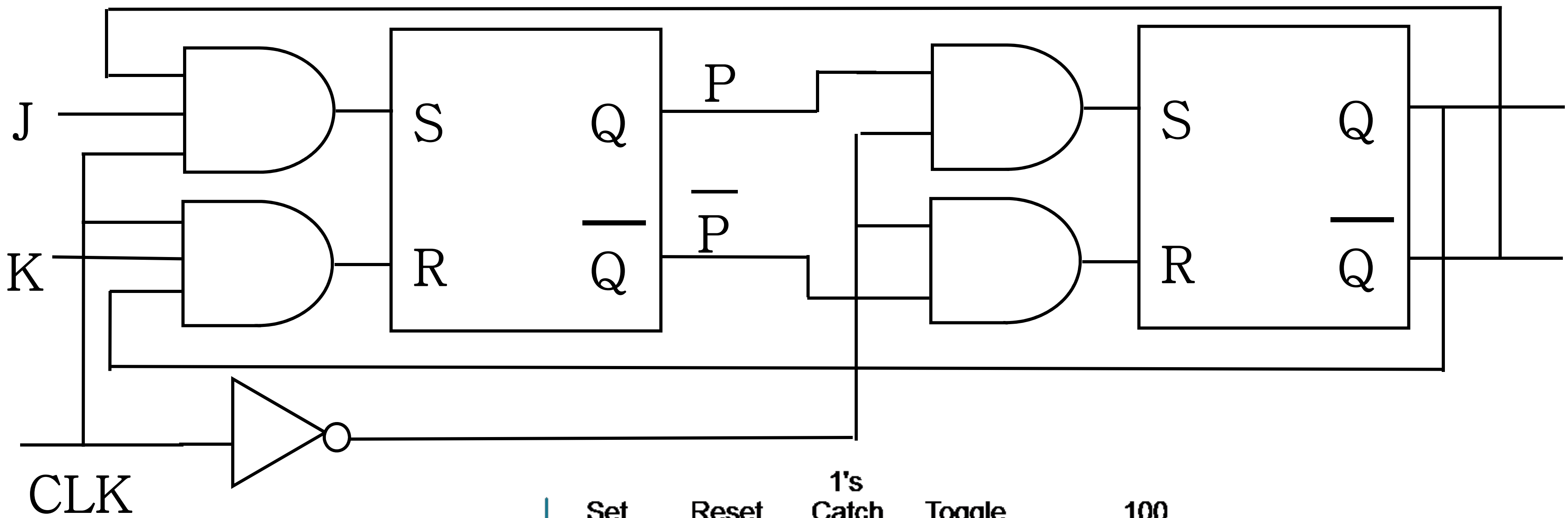


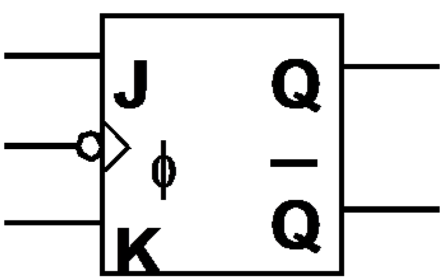
- Eliminate the forbidden state of the SR Flip-flop
- Use output feedback to guarantee that R and S are never both one

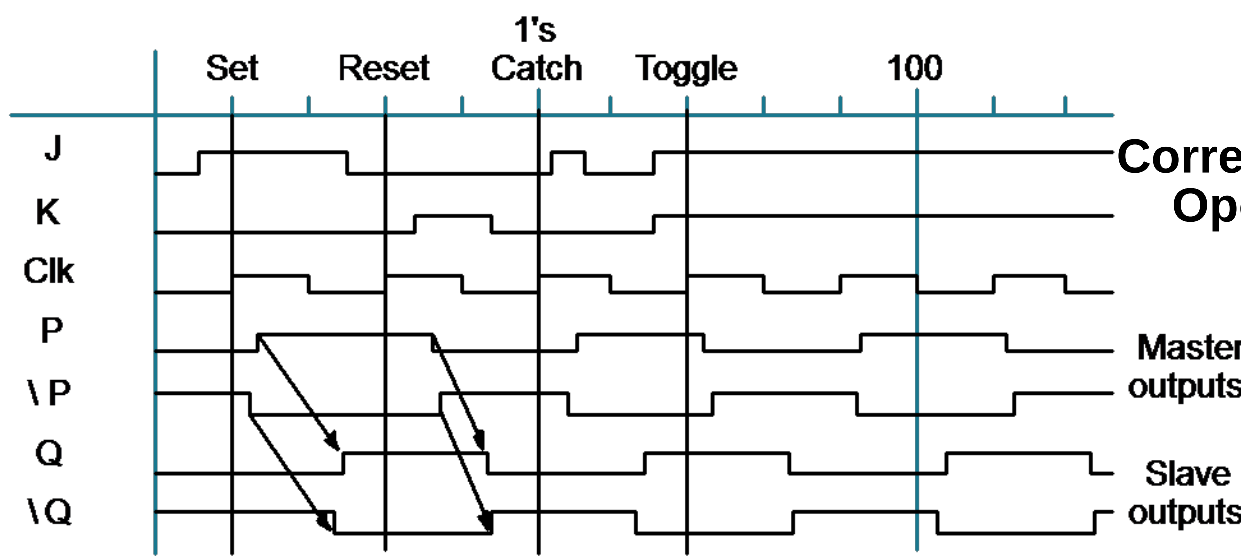
J-K Master-Slave Register

Sample inputs while clock high

Sample inputs while clock low



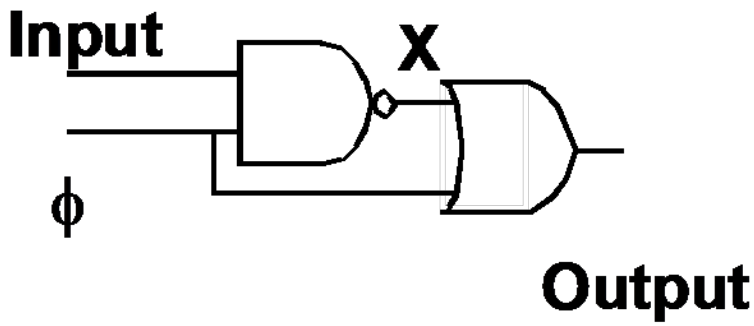




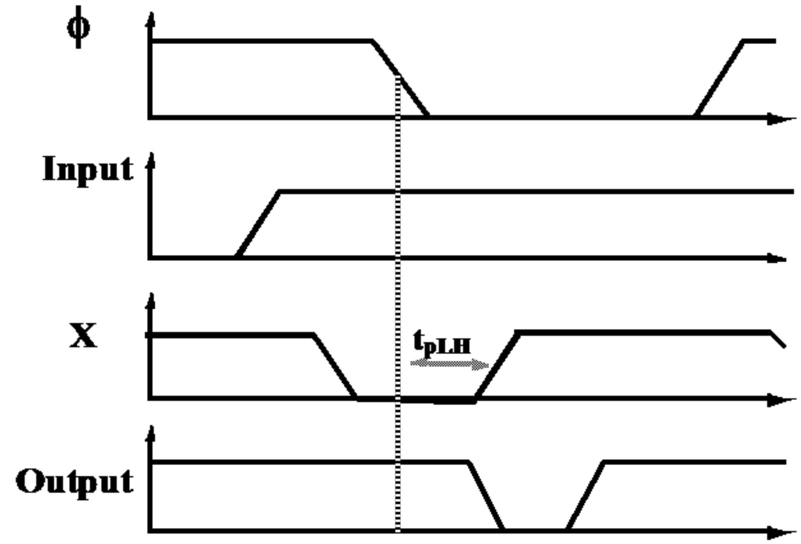
Correct Toggle Operation

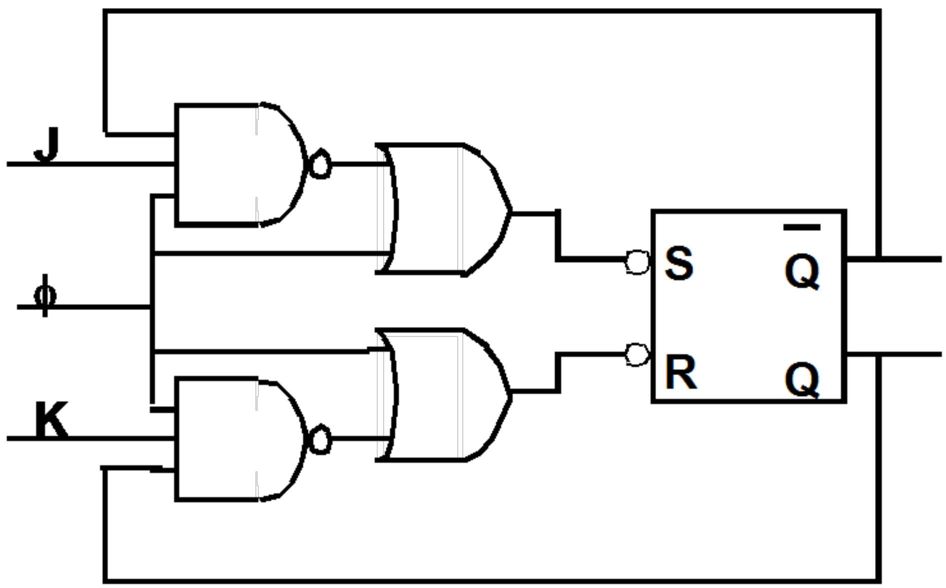
Is there a problem with this circuit?

Pulse Based Edge-Triggered J-K Register

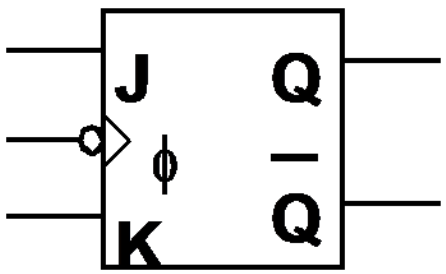


Schematic





JK Register Schematic

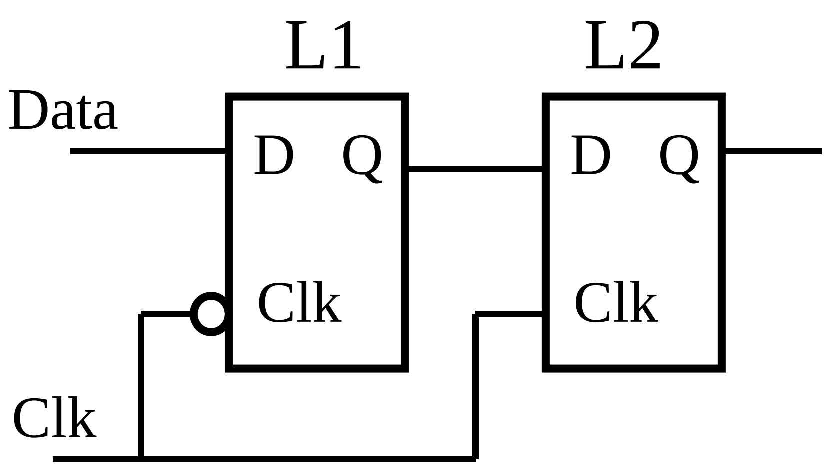


JK Register Logic Symbol

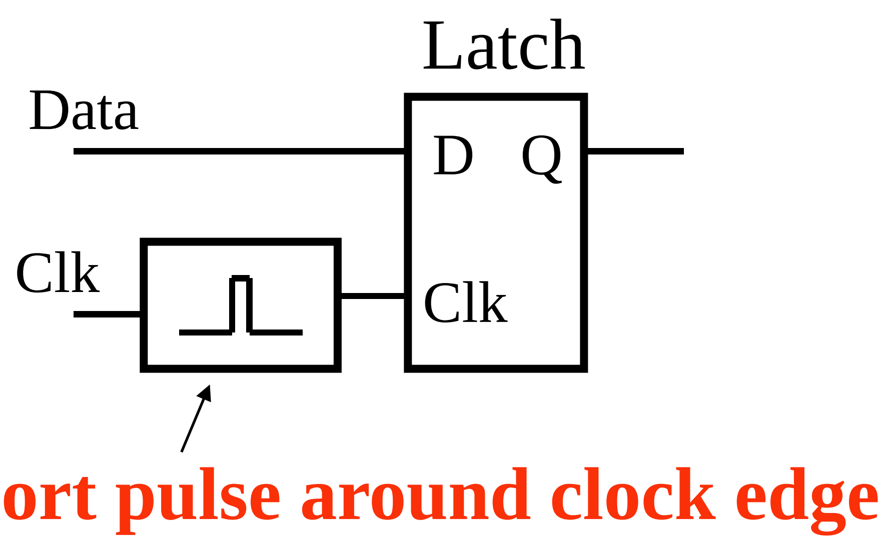
Pulse-Triggered Registers

Ways to design an edge-triggered sequential cell:

Master-Slave Latches

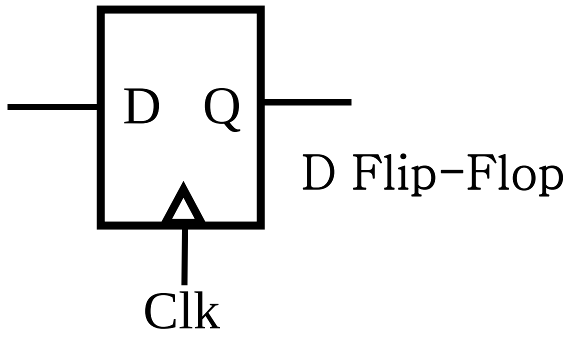


Pulse-Based Register

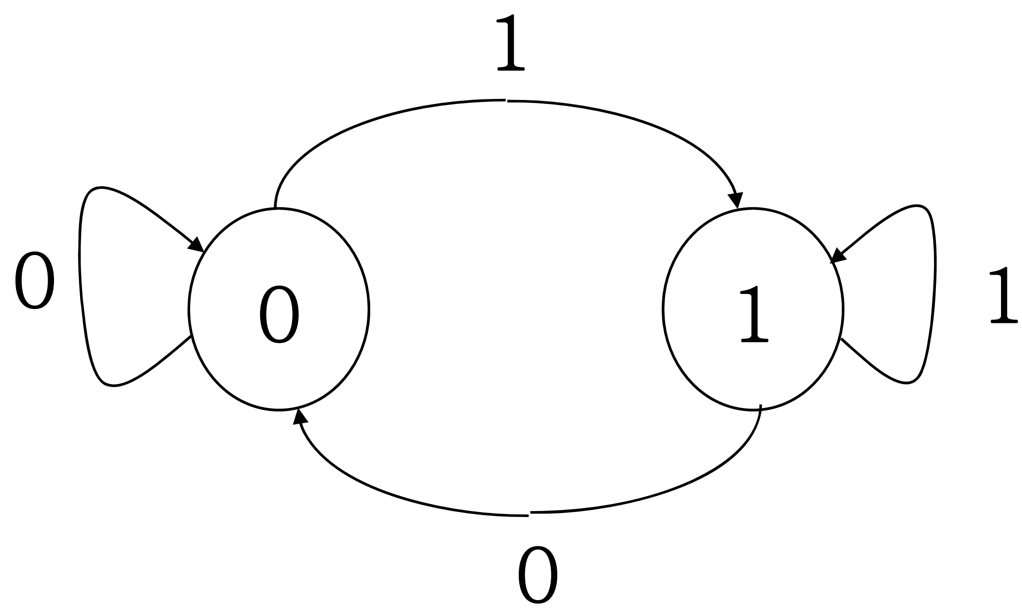


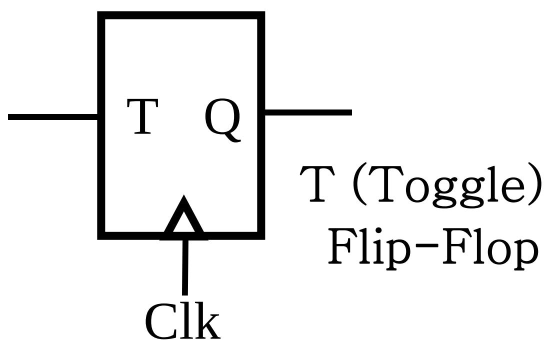
- **Pulse registers are widely used in high-performance microprocessor chips (Sun Microsystems, AMD, Intel, etc.)**
- **The can have a negative setup time!**

D Flip-Flop vs. Toggle Flip-Flop

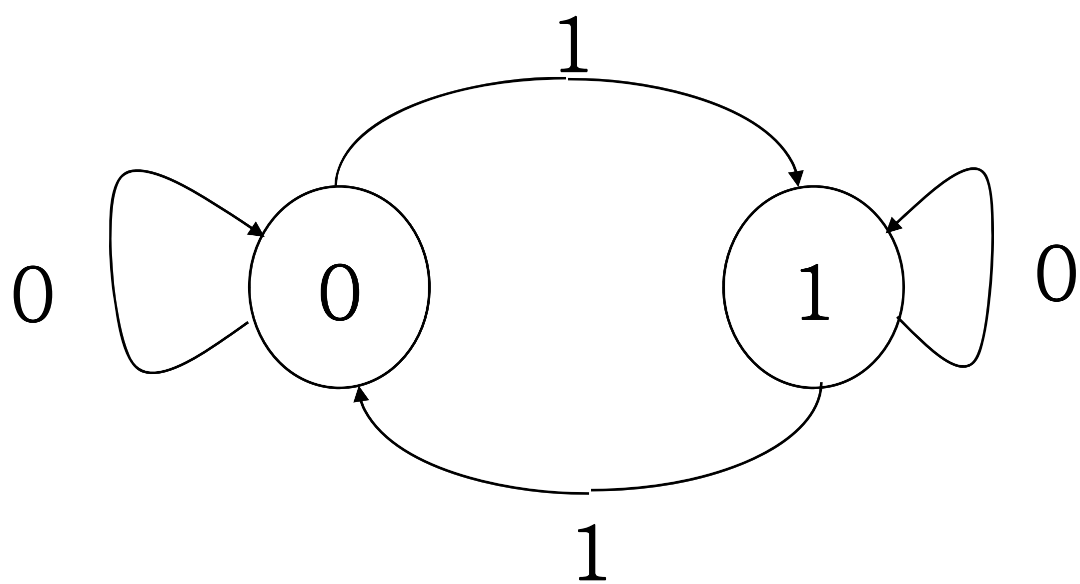


D	Q_N
0	0
1	1





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



Realizing Different Types of Memory Elements

Characteristic Equations

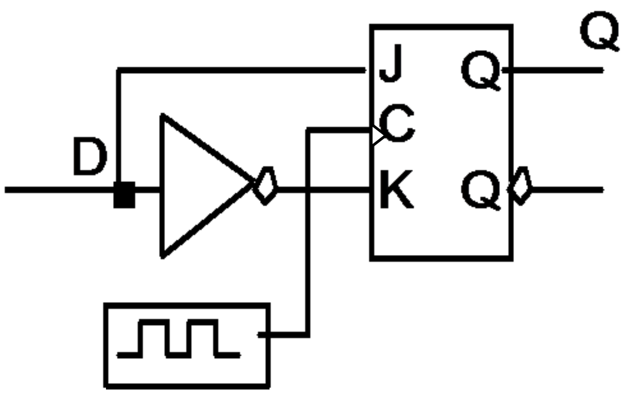
D: $Q+ = D$

J-K: $Q+ = J\overline{Q} + \overline{K}Q$

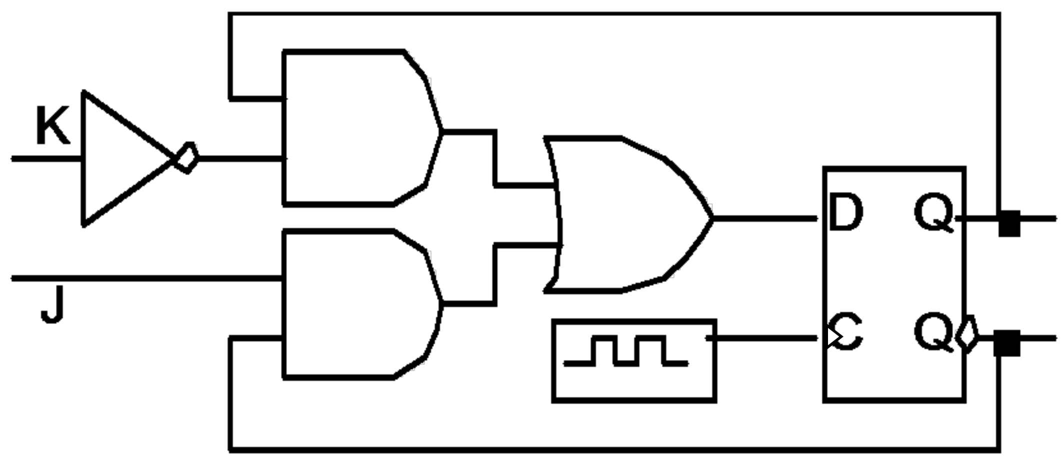
T: $Q+ = T\overline{Q} + \overline{T}Q$

E.g., J=K=0, then Q+ = Q
J=1, K=0, then Q+ = 1
J=0, K=1, then Q+ = 0
J=1, K=1, then $Q+ = \overline{Q}$

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?

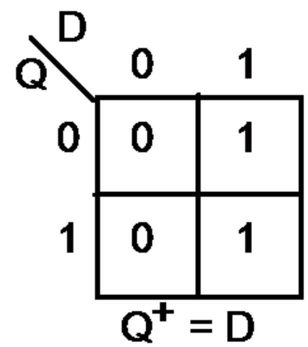
Q	Q+	J	K	T	D
0	0	0	X	0	0
0	1	1	X	1	1
1	0	X	1	1	0
1	1	X	0	0	1

Implementing D FF with a J-K FF:

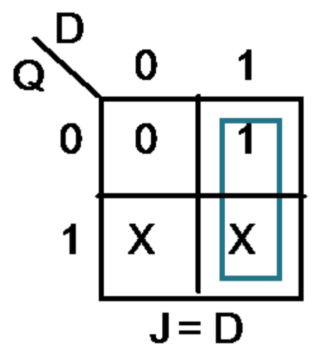
1) Start with K-map of Q+ = *f*(D, Q)

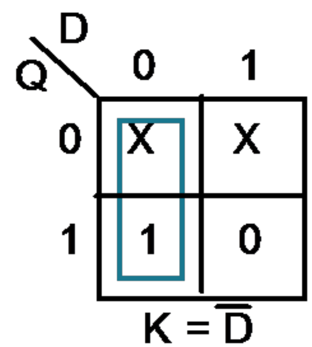
2) Create K-maps for J and K with same inputs (D, Q)

3) Fill in K-maps with appropriate values for J and K
   to cause the same state changes as in the original K-map



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



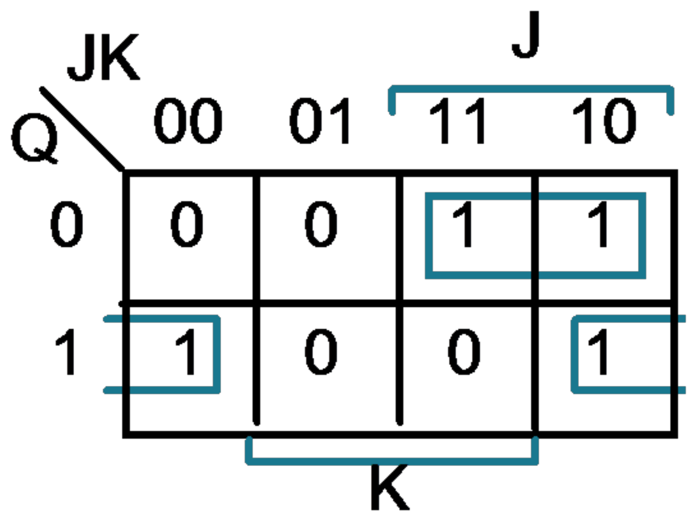


Design Procedure (cont.)

Implementing J-K FF with a D FF:

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

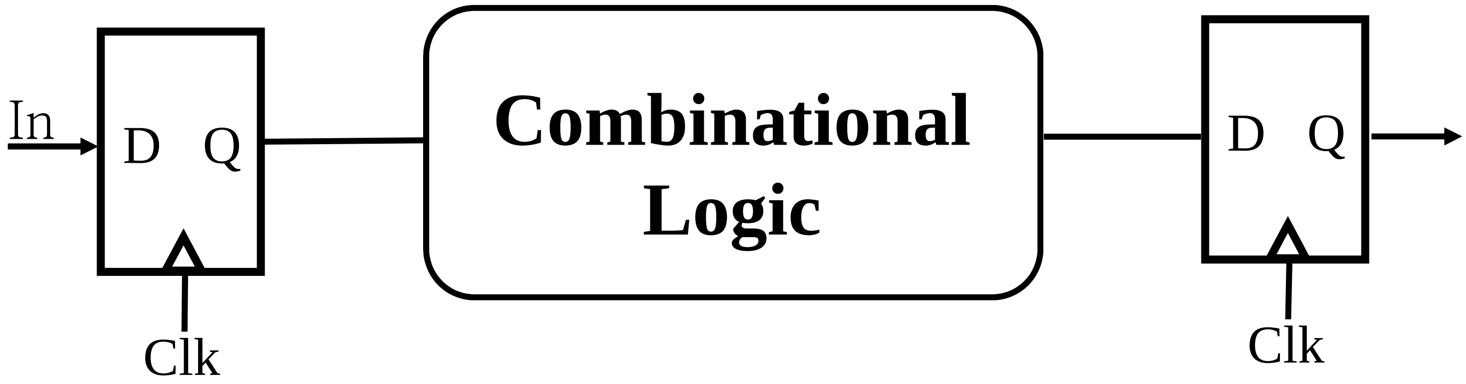
2,3) Revised K-map using D's excitation table
its the same! that is why design procedure with D FF is simple!



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

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

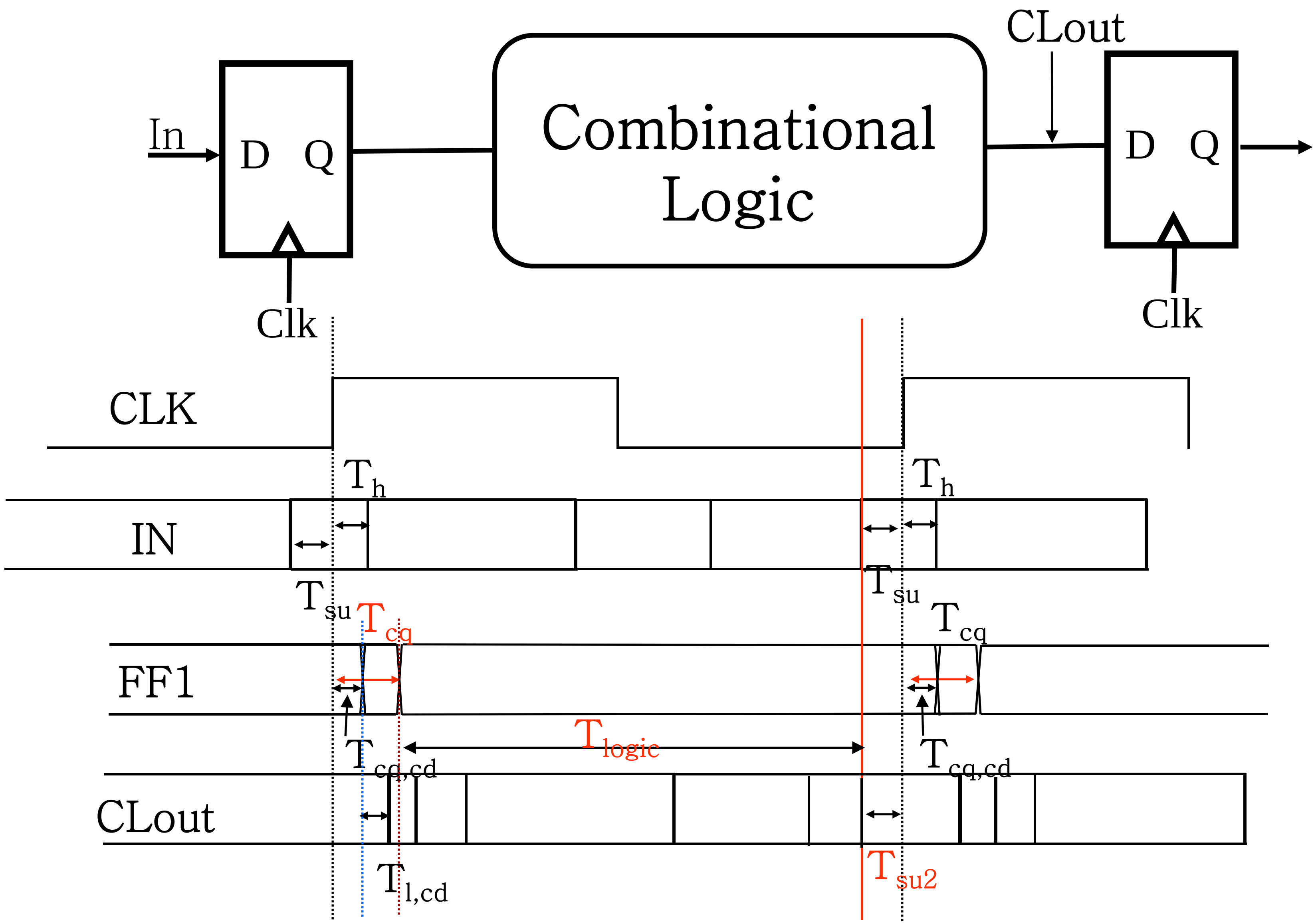
T_h: hold time

Logic Timing Parameters

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

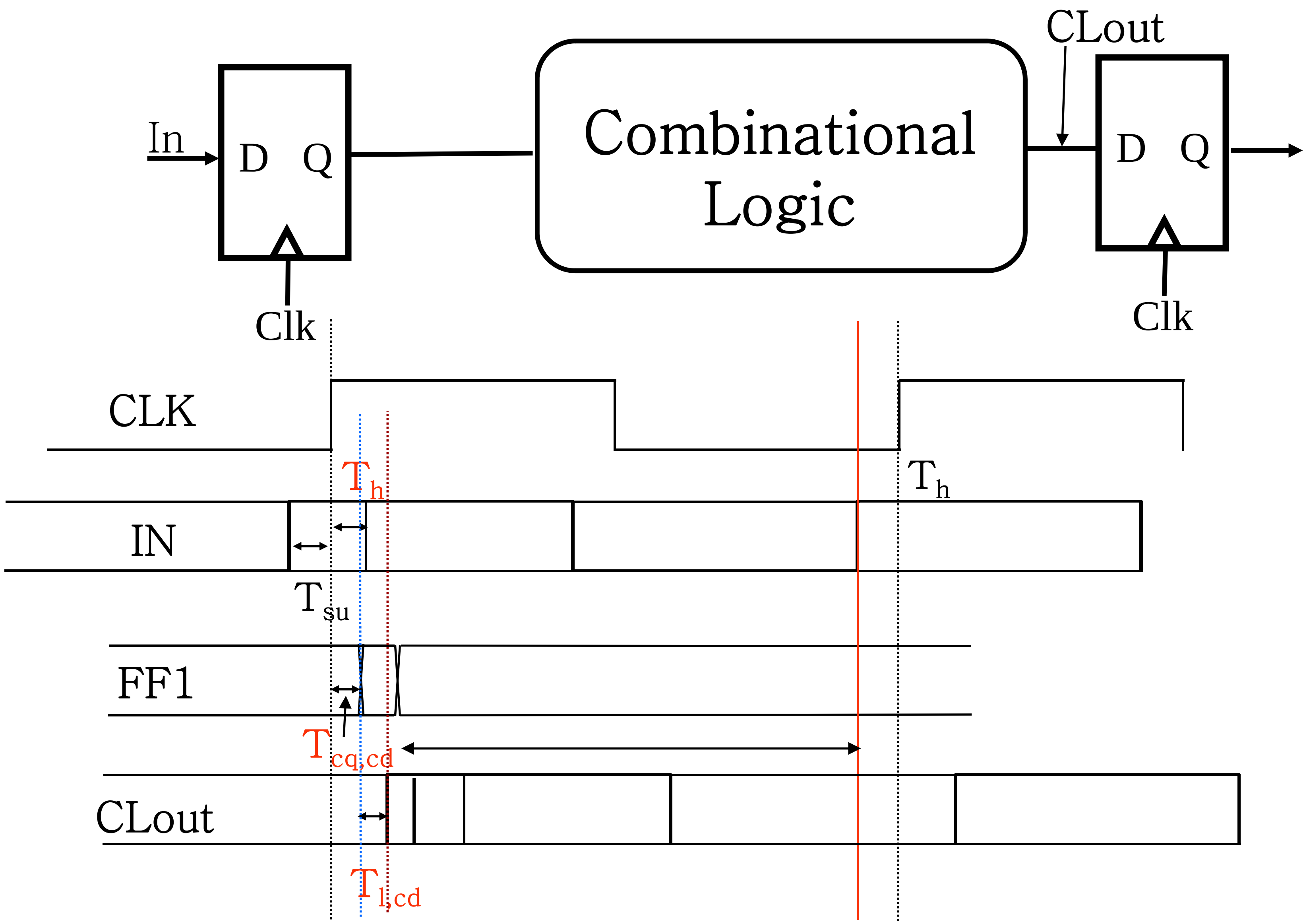
$T_{logic,cd}$: contamination or minimum delay through logic network

System Timing (I): Minimum Period



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

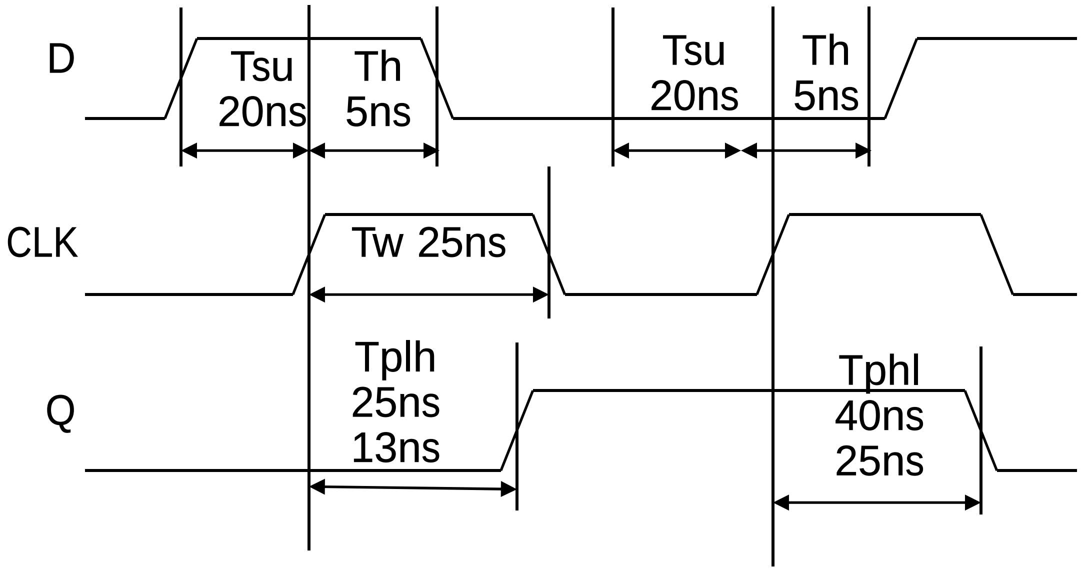
System Timing (II): Minimum Delay



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

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**