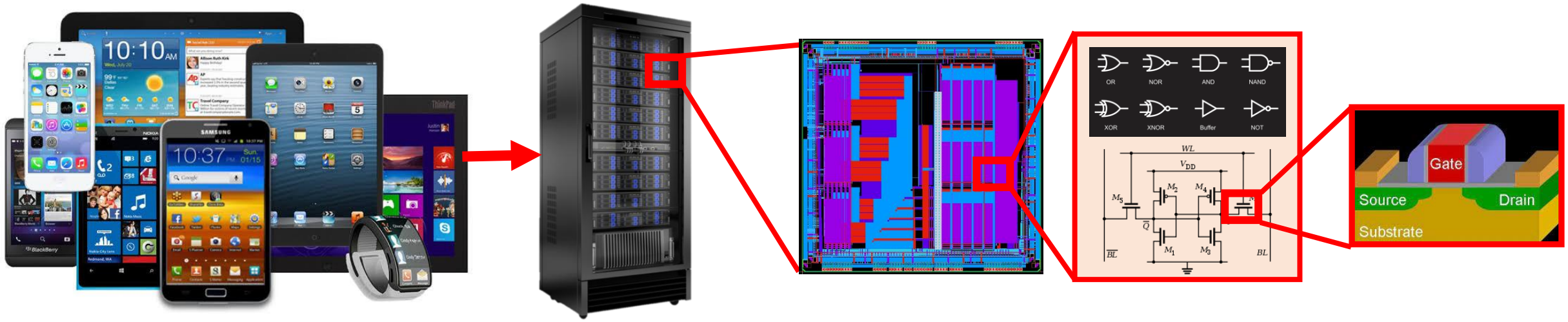
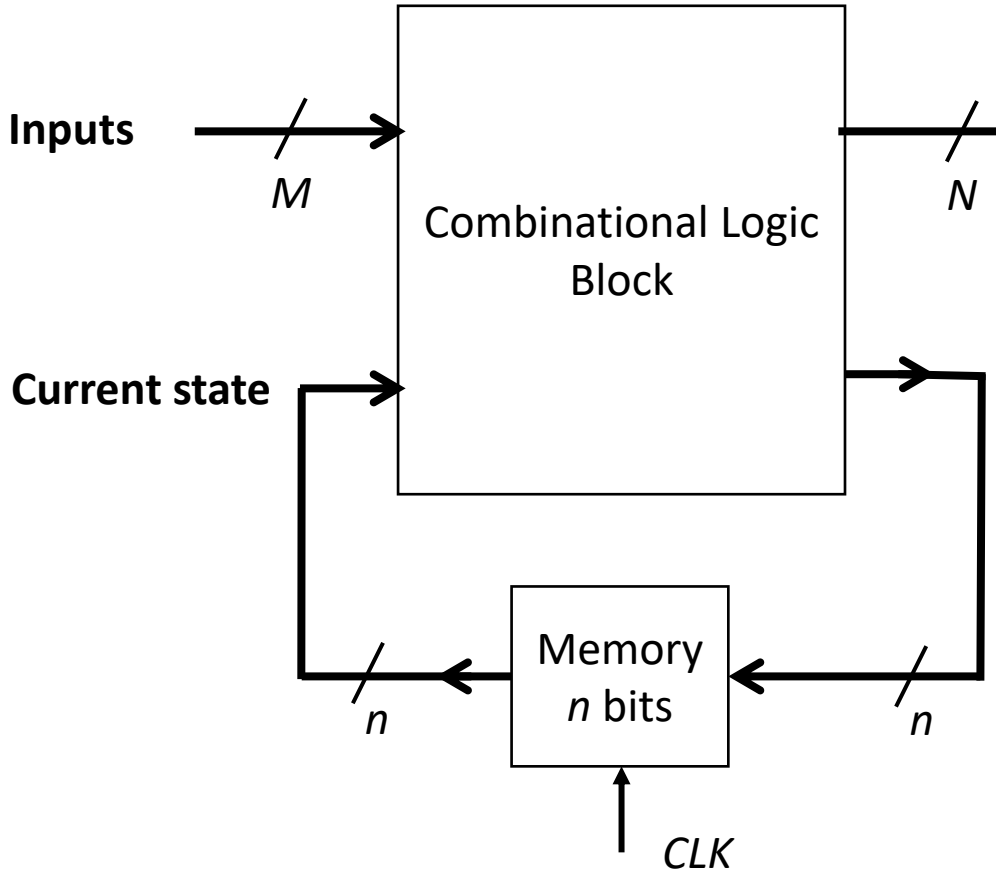


ESC201: Introduction to Electronics

Module 6: Digital Circuits

Dr. Shubham Sahay,
Associate Professor,
Department of Electrical Engineering,
IIT Kanpur

Sequential Circuits



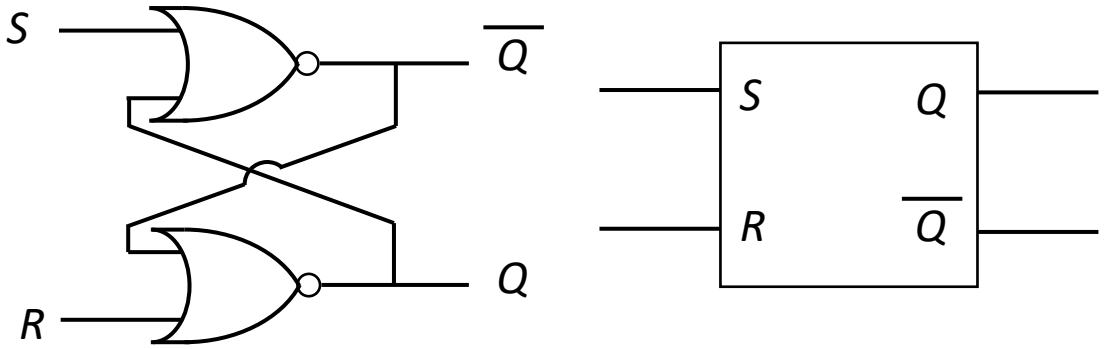
- Memory updated at clock level or clock edge
- Current state defined by *n* bits stored currently in memory

- Output will change as per current state and the input given!
- Can decide what will be the next state based on current state and inputs

→ Beginning of decision making and intelligence!

Static *SR* Flip-Flop* with NOR Gates

S ≡ 'set' latch output value *Q*; and *R* ≡ 'reset' latch output value *Q*



S	*R*	*Q*	$\overline{Q}$	Remark
0	0	*Q*	$\overline{Q}$	Hold
0	1	0	1	Re-set
1	0	1	0	Set
1	1	0	0	**Prohibited**

For NOR gate, Any input being "1" will make output "0".

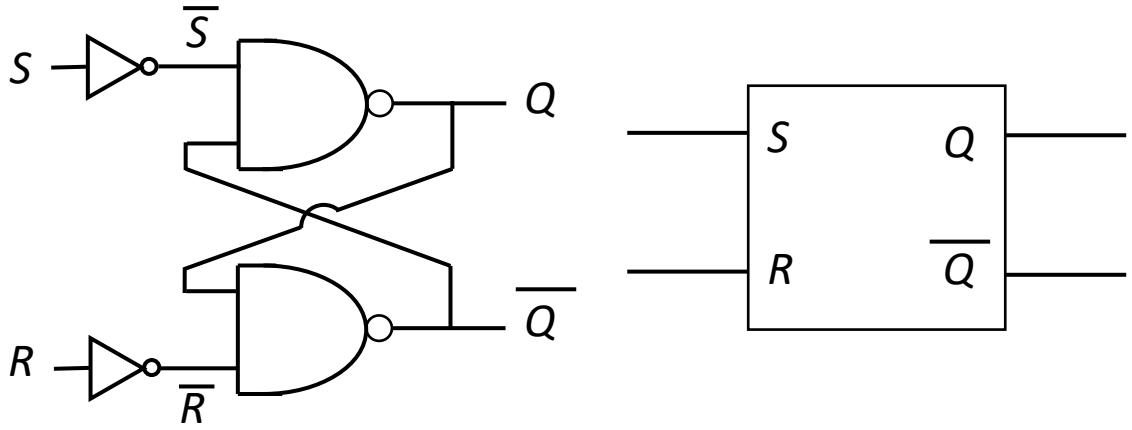
Input *S* = *R* = 1 is prohibited because when it comes out of that state to *S* = *R* = 0, we do not know whether *Q* becomes 0 or 1, leading to unpredictability of future states.

If *S* = *R* = 1 state occurs, make the next state compulsorily
S = 1, *R* = 0 or *S* = 0, *R* = 1 state.

* 'Flip-Flop' as defined in Rabaey et al.'s book is
any bi-stable component "formed by cross-coupling of gates"

Static *SR* Flip-Flop with NAND Gates

NAND gate implementation is in the dual space as with NOR gate implementation



S	*R*	*Q*	$\overline{Q}$	Remark
0	0	*Q*	$\overline{Q}$	Hold
0	1	0	1	Re-set
1	0	1	0	Set
1	1	1	1	**Prohibited**

For NAND gate, Any input being “0” will make output “1”.

Input *S* = *R* = 1 is prohibited because when it comes out of that state to *S* = *R* = 0, we do not know whether *Q* becomes 0 or 1, leading to unpredictability of future states.

If *S* = *R* = 1 state occurs, make the next state compulsorily *S* = 1, *R* = 0 or *S* = 0, *R* = 1 state.

- *SR* Latch (set-reset latch)
- *JK* Latch
- *D* Latch (delay latch)
- *T* Latch (toggle latch)

Let us look at the
- Characteristics table for each latch
- Excitation table for each latch

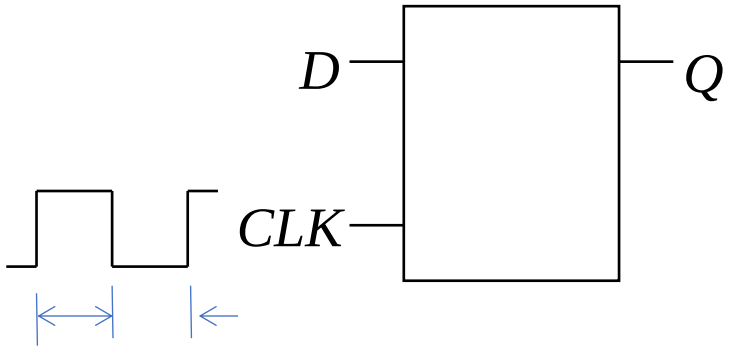
* Note that registers can built combining latches.

Defining* Latch and Register

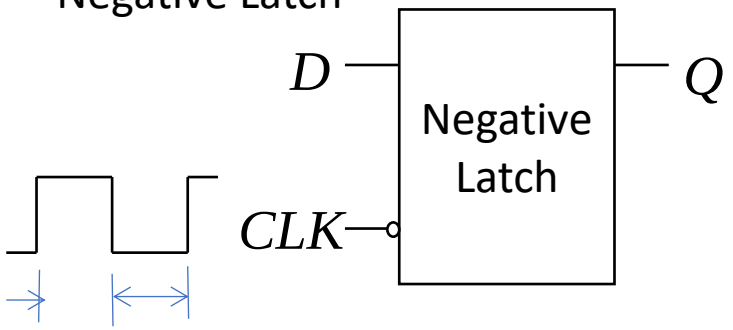
Latch

Q "transparent" to *Input* for clock level

Positive Latch



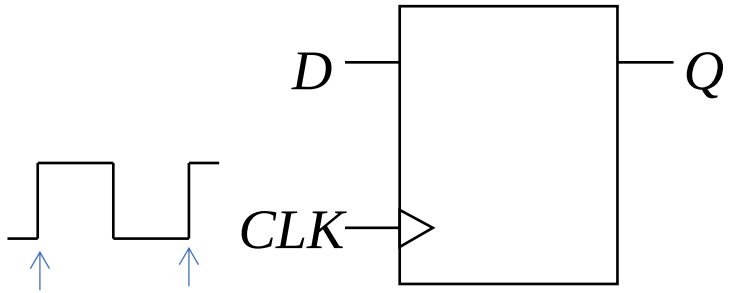
Negative Latch



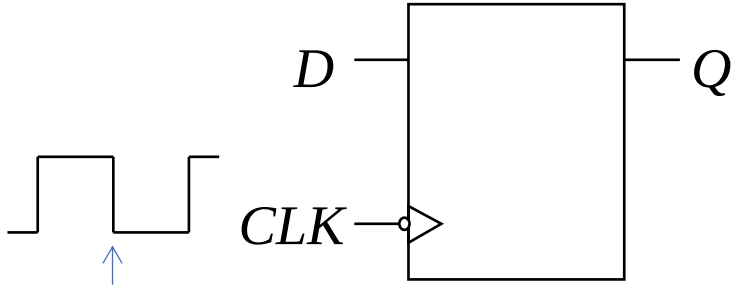
Registers

Q "transparent" to *Input* for clock edge

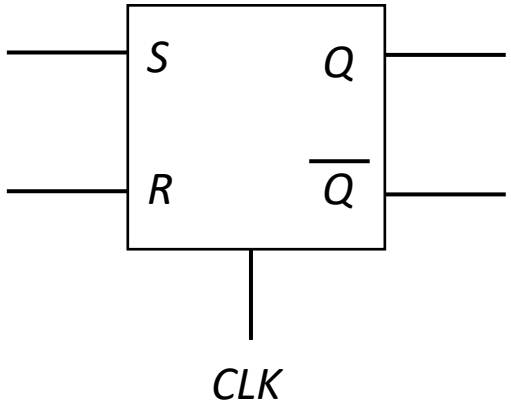
Positive Register



Negative Register



* Defined according to *Rabaey, Chandrakasan & Nikolic*
Definitions may vary in other text books



Characteristics Table

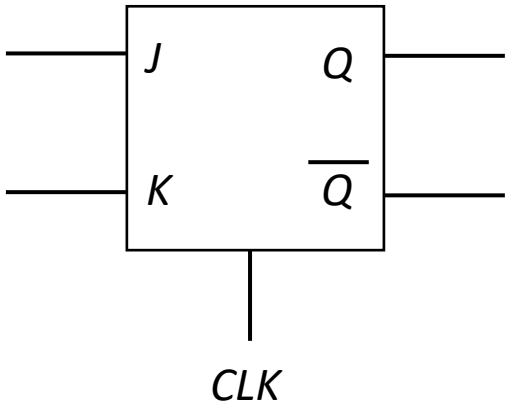
Inputs		Outputs		
S	R	Q_{n+1}	$\overline{Q}_{n+1}$	Remark
0	0	Q_n	$\overline{Q}_n$	Hold
0	1	0	1	Re-set
1	0	1	0	Set
1	1	0	0	**Prohibited**

Subscript n is current clock cycle
Subscript n+1 is next clock cycle

Excitation Table

Desired transition		Required Inputs	
Q_n	Q_{n+1}	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

Here x represents a don't care state



Characteristics Table

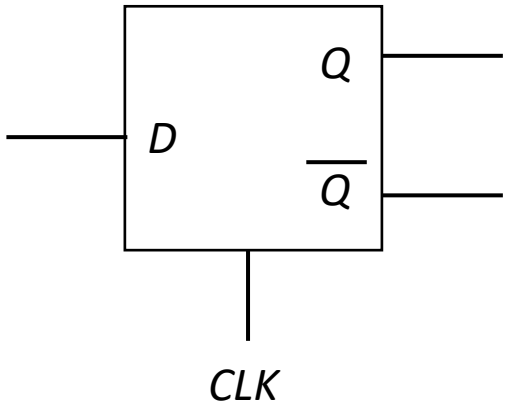
Inputs		Outputs		
J	**K**	**Q_{n+1}**	**$\overline{Q}_{n+1}$**	**Remark**
0	0	Q_n	$\overline{Q}_n$	Hold
0	1	0	1	Re-set
1	0	1	0	Set
1	1	$\overline{Q}_n$	Q_n	Toggle

Subscript *n* is current clock cycle
Subscript *n*+1 is next clock cycle

Excitation Table

Desired transition		Required Inputs	
Q_n	**Q_{n+1}**	**J**	**K**
0	0	0	*x*
0	1	1	*x*
1	0	*x*	1
1	1	*x*	0

Here *x* represents a don't care state



Characteristics Table

Inputs	Outputs		
D	Q_{n+1}	$\overline{Q_{n+1}}$	Remark
0	0	1	Reset
1	1	0	Set

Excitation Table

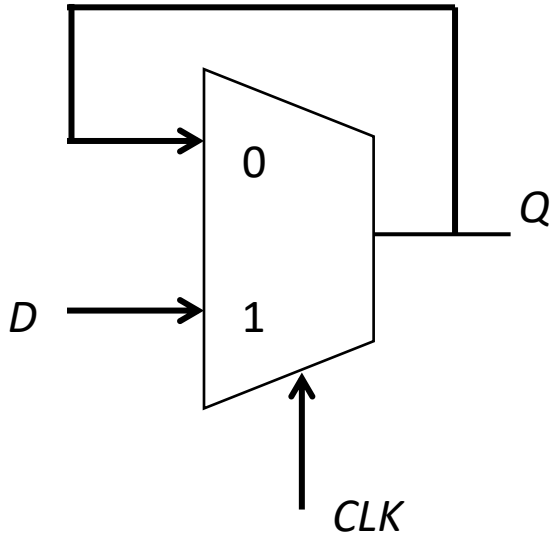
Desired transition		Required Inputs
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Subscript *n* is current clock cycle
Subscript *n*+1 is next clock cycle

Positive and Negative Latches with MUX

with 2 to 1 Multiplexer (MUX)

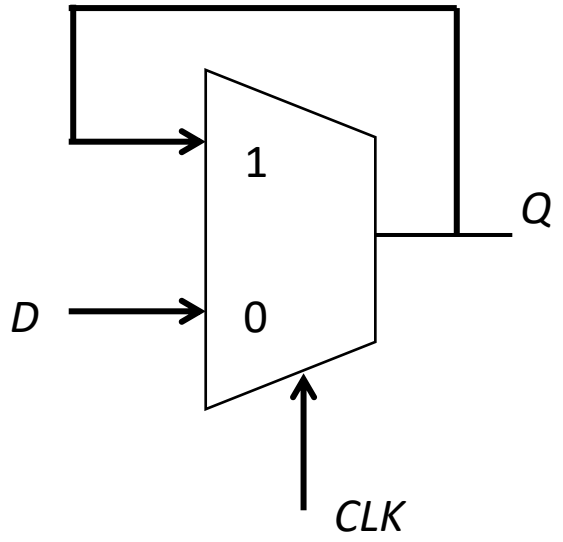
Positive Latch



Write (Feedback is broken)
When *CLK* = 1, *Q* = *D*

Read (+ve Feedback is restored)
When *CLK* = 0, *Q* value does not change

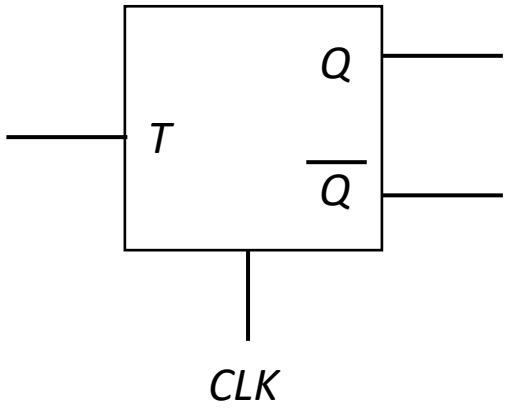
Negative Latch



Write (Feedback is broken)
When *CLK* = 0, *Q* = *D*

Read (+ve Feedback is restored)
When *CLK* = 1, *Q* value does not change

A positive and negative latch combination can help you build a register.



Characteristics Table

Inputs	Outputs		
T	Q_{n+1}	$\overline{Q_{n+1}}$	Remark
0	Q_n	$\overline{Q_n}$	Hold
1	$\overline{Q_n}$	Q_n	Toggle

Excitation Table

Desired transition		Required Inputs
Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Subscript n is current clock cycle
Subscript n+1 is next clock cycle

Exercise

- Can you build a *D* latch from a *JK* latch?

- Given a latch, with logic gates one may build any other latch.

- Give it a try!