Dhirubhai Ambani Institute of Information and Communication Technology

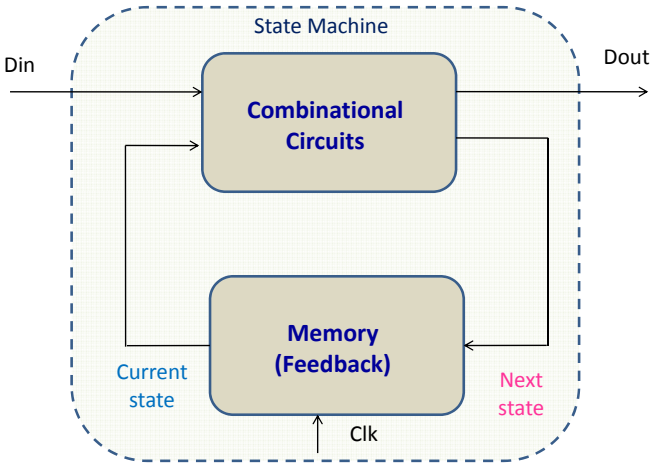
EL114

Digital Logic Design

Dr. Yash Agrawal
Assistant Professor,
DA-IICT, Gandhinagar

Introduction

Sequential Circuit



Introduction

How to realize Storage (Memory) Element

Basic Building Block for Storage/Memory

Latch

- Latch is level triggered single-bit memory element.
- Latch responds immediately to change in input(s)and (possibly in the presence of level enables control signal).
- Output can change multiple times during active enable (clock) signal.

Flip-flop

- Flip-flop is edge triggered single-bit memory element.
- Output can change only one time during clock cycle.

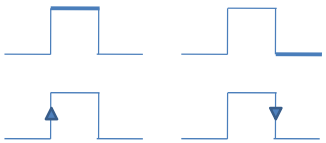
Basic Building Block for Storage/Memory

Latch vs. Flip-flop

What's the basic difference…?

Latch is level-triggered

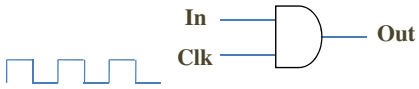
Flip-flop is edge-triggered



Basic Building Block for Storage/Memory

Case I

Consider the Circuit

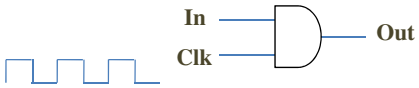


1. This circuit is level triggered or edge triggered?

Basic Building Block for Storage/Memory

Case I

Consider the Circuit

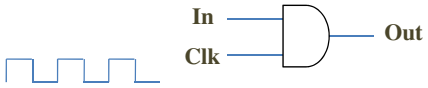


2. What do you mean by edge triggered?
3. How circuit can operate at edge of control signal?
4. Fundamentally, how circuit see level/edge triggering?

Basic Building Block for Storage/Memory

Case I

Consider the Circuit

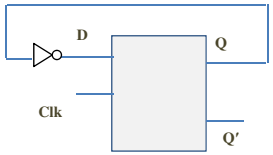


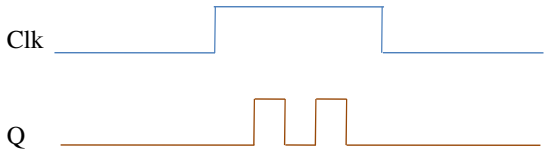
5. How to operate the same circuit as in level triggered or edge triggered configuration?

Case II

Consider the Circuit

Let initially Q = 0



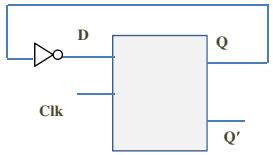


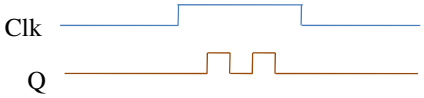
- As long as Clk = 1, the value of Q continues to change.
- The change are based on the delay present on the loop through the connection from Q to Q.
- This behavior is unacceptable not desirable.
- Desired Behavior: Q changes only once clock pulse

Case II

Consider the Circuit

Let initially Q = 0





What to do…?

The solution to the latch timing issue is to **break** the closed path from Q to Q within the storage element.

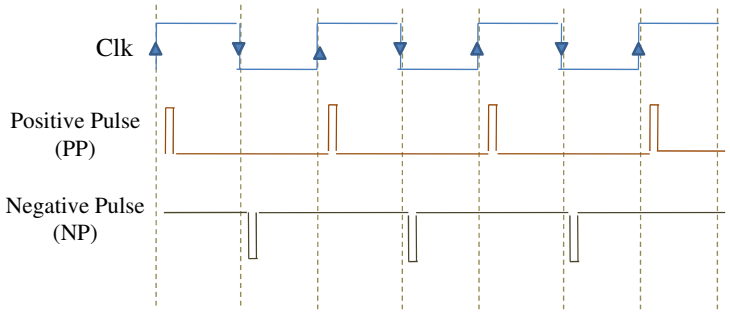
Issues in Case I and Case II

1. How edge triggering concept can be executed…?
2. How to break closed path?

Solutions to both Case I and Case II

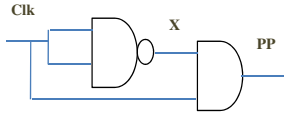
1. Pulse Generator
2. Master-Slave Configuration

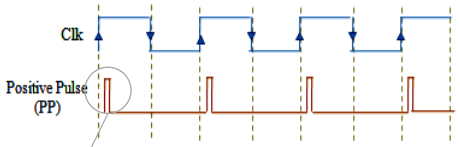
1. Pulse Generation



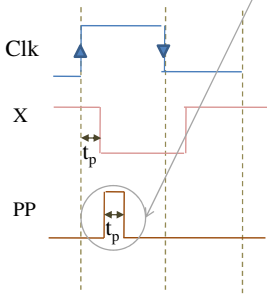
* The pulse width ≤ propagation delay (t_p) of flip-flop

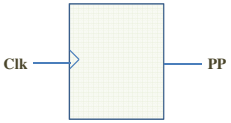
Positive Pulse Forming Circuit





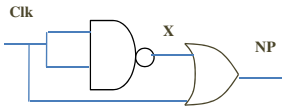
Propagation delay of each gate (AND, NAND) is t_p.

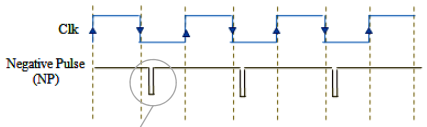




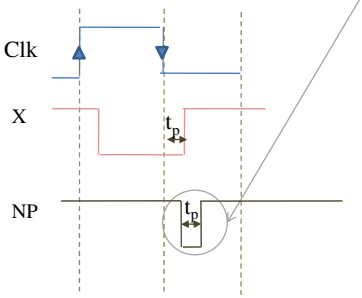
Block diagram representation

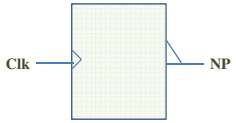
Negative Pulse Forming Circuit





Propagation delay of each gate (AND, NAND) is t_p.

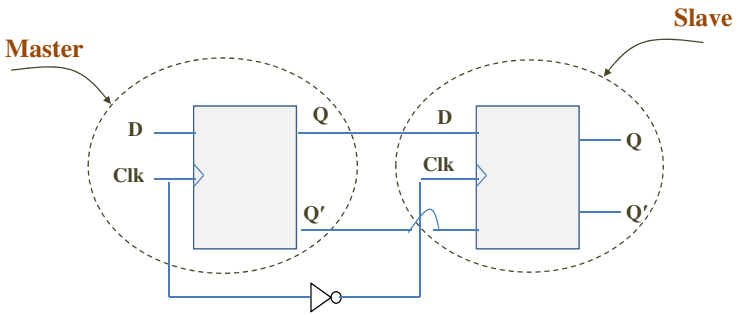




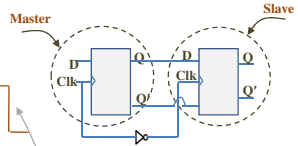
Block diagram representation

2. Master-Slave Configuration

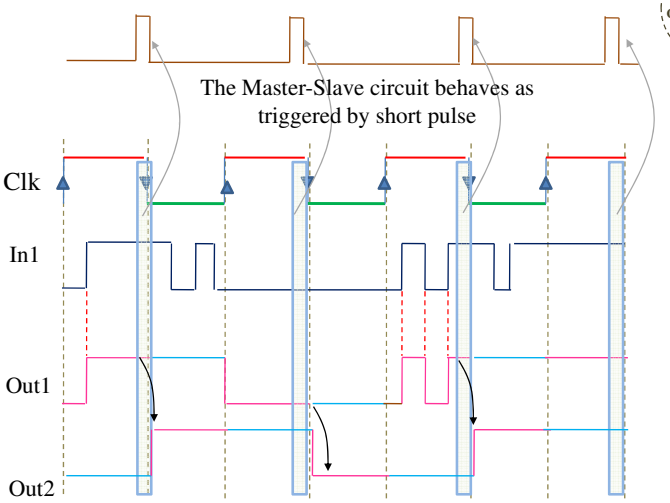
Consider the Circuit



Master-Slave Configuration



The Master-Slave circuit behaves as triggered by short pulse



Types of Flip-flop

Different Flip-flops that will be discussed are:

- S-R
- D
- J-K
- T

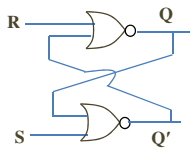
S-R Flip-flop

S-R Flip-flop is known as SET-RESET Flip-flop.

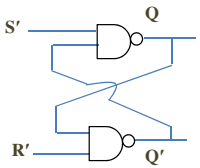
It can be used to either Set/Reset output or store previous value.

Basic Hardware Designs for Storage/Memory Element

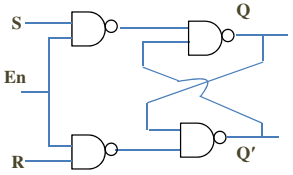
Cross-coupled NORs (S-R Latch)



Cross-coupled NANDs (S′-R′ Latch)

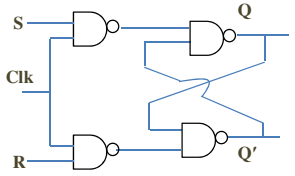


Gated S-R Latch



Level triggered

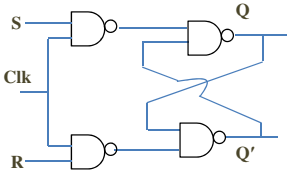
How to implement S-R Flip-flop…?



Clock control signal is used

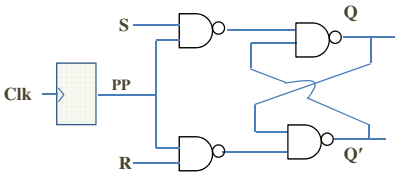
Will it work…?

Basic Hardware Designs for Storage/Memory Element

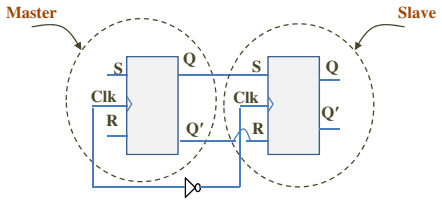


Clock control signal is used

S-R Flip-flop

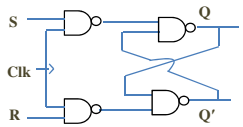


Using pulse generator

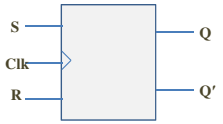


Using Master-Slave configuration

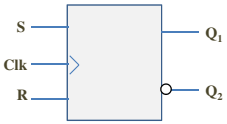
S-R Flip-flop



Circuit diagram



Block diagram



Block diagram

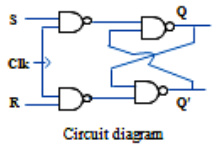
Truth Table

Input of Flip-Flop | Output of flip-flop

Clk	S	R	Q_{n+1}	State
↑	0	0	Previous State	No Change
↑	0	1	0	Reset
↑	1	0	1	Set
↑	1	1	Undefined	Forbidden
0	x	x	Previous State	No Change

* Q'_{n+1} is complementary to Q_{n+1}.

S-R Flip-flop



Circuit diagram

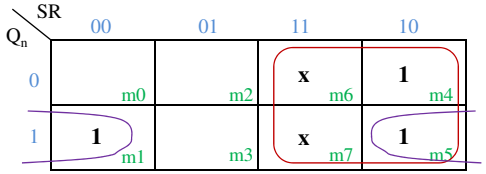
Characteristic Table

Inputs of FF | Output of FF

S	R	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	x
1	1	1	x

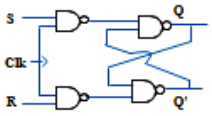
Characteristic equation

Determination of Q_{n+1} using K-map



$$Q_{n+1} = S + R'Q_n$$

S-R Flip-flop



Circuit diagram

Characteristic Table

Inputs of FF | Output of FF

S	R	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	x
1	1	1	x

Excitation Table

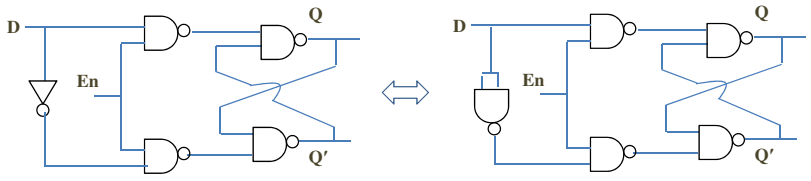
Output of FF | Input of FF

Q_n	Q_{n+1}	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

D Flip-flop

- D flip-flop is also known as Delay and Transparent flip-flop.
- It is derived from S-R flip-flop where ‘R’ input is inverted of ‘S’ input.
- In D flip-flop ‘00’ or ‘11’ doesn’t exist. Hence, there is no indefinite/forbidden condition.
- D flip-flop is often used a 1-bit memory element.

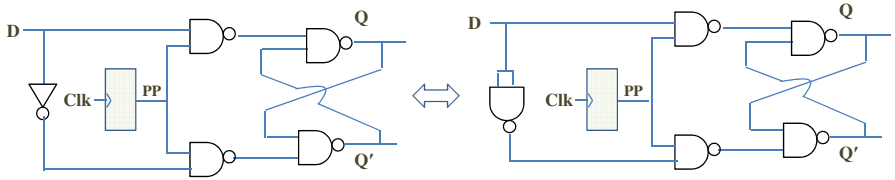
Gated D Latch



Level triggered

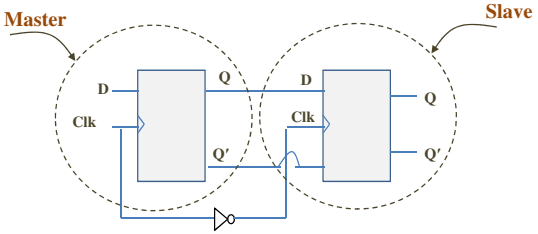
How to implement D Flip-flop…?

D Flip-flop



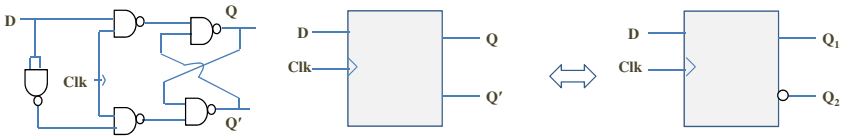
Using Pulse generator

D Flip-flop



Using Master-Slave configuration

D Flip-flop



Circuit diagram Block diagram Block diagram

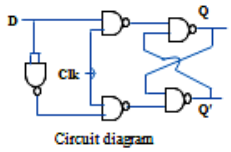
Truth Table

Input of FF Output of FF

Clk	D	Q_{n+1}	State
↑	0	0	Reset
↑	1	1	Set
0	x	Previous State	No Change

* Q'_{n+1} is complementary to Q_{n+1}.

D Flip-flop



Circuit diagram

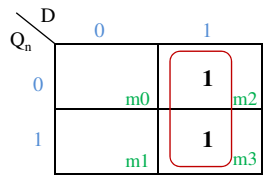
Characteristic Table

Input of FF Output of FF

D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

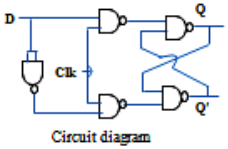
Characteristic equation

Determination of Q_{n+1} using K-map



$Q_{n+1} = D$

D Flip-flop



Circuit diagram

Characteristic Table

Input of FF	Output of FF
D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

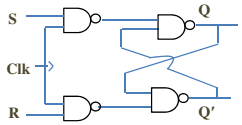
Excitation Table

Output of FF		Input of FF
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

J-K Flip-flop

- J-K flip-flop is proposed by **J**ack-**K**ilby.
- It is derived from S-R flip-flop where feedback is provided to overcome forbidden (11) condition. Here Q′ is connected to J and Q to K.

S-R Flip-flop



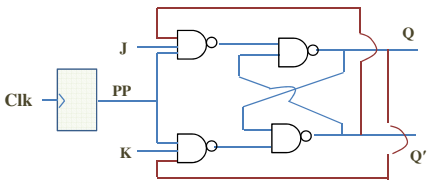
Input of FF			Output of flip-flop	
Clk	S	R	Q_{n+1}	State
↑	0	0	Previous State	No Change
↑	0	1	0	Reset
↑	1	0	1	Set
↑	1	1	Undefined	Forbidden
0	x	x	Previous State	No Change

This condition is undesirable

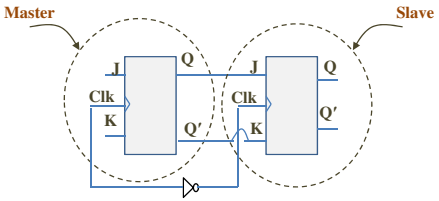
Leads to Race around condition

How to overcome it…?

J-K Flip-flop

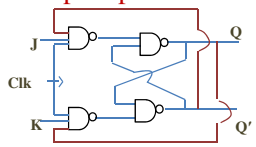


Using pulse generator

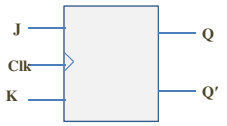


Using Master-Slave configuration

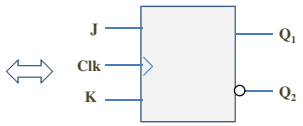
J-K Flip-flop



Circuit diagram



Block diagram



Block diagram

Truth Table

Input of flip-flop			Output of Flip-flop	
Clk	J	K	Q_{n+1}	State
↑	0	0	Previous State	No Change
↑	0	1	0	Reset
↑	1	0	1	Set
↑	1	1	Complement of previous output Q'_n	Toggle
0	x	x	Previous State	No Change

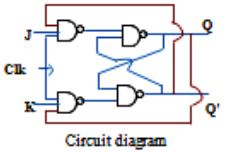
* Q'_{n+1} is complementary to Q_{n+1} .

J-K Flip-flop

Characteristic Table

Inputs of FF | Output of FF

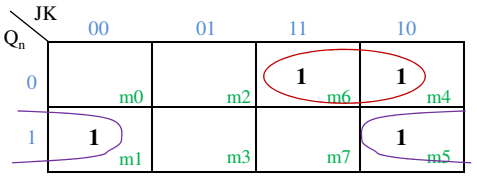
J	K	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



Circuit diagram

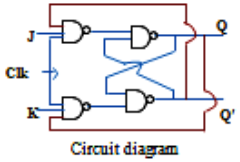
Characteristic equation

Determination of Q_{n+1} using K-map



$$Q_{n+1} = JQ'_n + K'Q_n$$

J-K Flip-flop



Circuit diagram

Characteristic Table

Inputs of FF | Output of FF

J	K	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Excitation Table

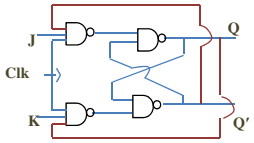
Output of FF | Input of FF

Q_n	Q_{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

T Flip-flop

- T flip-flop is also known as Toggle Flip-flop.
- It is derived from J-K flip-flop where J and K input terminals are shorted with each other.
- The T flip-flop doesn't exhibits 01' or '10' condition of J-K flip-flop.
- Owing to its continuous Toggling output generation, it is many times used for clock generation or as frequency divider circuit.

J-K Flip-flop

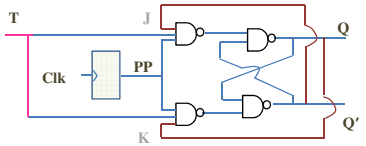


Input of flip-flop | Output of Flip-flop

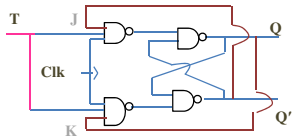
Clk	J	K	Q_{n+1}	State
↑	0	0	Previous State	No Change
↑	0	1	0	Reset
↑	1	0	1	Set
↑	1	1	Complement of previous output Q'	Toggle
0	x	x	Previous State	No Change

Targeting these conditions

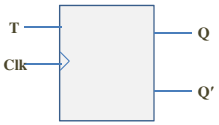
T Flip-flop



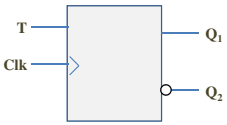
T Flip-flop



Circuit diagram



Block diagram



Block diagram

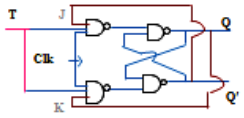
Truth Table

Input of FF | Output of Flip-flop

Clk	T	Q_{n+1}	State
↑	0	Previous State	No Change
↑	1	Complement of Previous output Q_n	Toggle
0	x	Previous State	No Change

* Q'_{n+1} is complementary to Q_{n+1} .

T Flip-flop



Circuit diagram

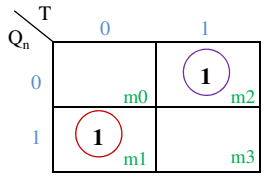
Characteristic Table

Input of FF | Output of FF

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

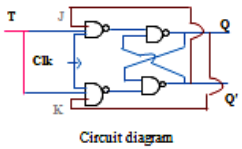
Characteristic equation

Determination of Q_{n+1} using K-map



$$Q_{n+1} = Q_nT' + Q'_nT$$

T Flip-flop



Circuit diagram

Characteristic Table

Input of FF | Output of FF

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Excitation Table

Output of FF | Input of FF

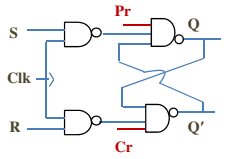
Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Asynchronous Flip-flops

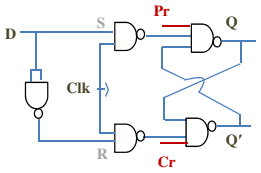
- These have special inputs as preset (Pr) or clear (Cr).
- This are sometimes also referred as Direct inputs.

Asynchronous Flip-flops

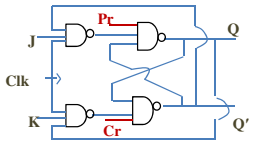
S-R Flip-flop



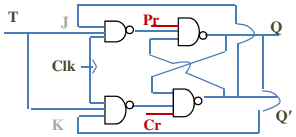
D Flip-flop



J-K Flip-flop



T Flip-flop



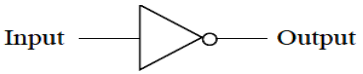
Asynchronous Flip-flops

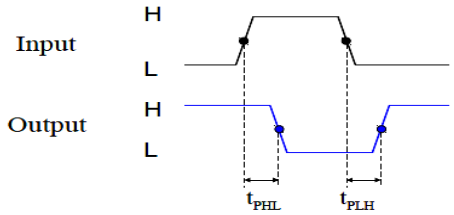
Truth Table

Direct Inputs		Output of Flip-flop	
Pr	**Cr**	**Q_{n+1}**	**State**
0	0	Undefined	Invalid
0	1	1	Preset
1	0	0	Clear
1	1	As per Flip-flop inputs	Normal operation

* Q'_{n+1} is complementary to Q_{n+1} .

Propagation delay

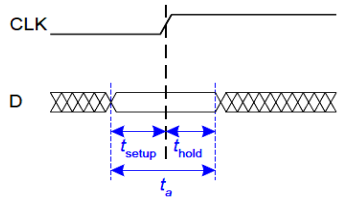




Setup Time (t_{setup}): Time before the clock edge for which data must remain stable (i.e. not changing).

Hold Time (t_{hold}): Time after the clock edge for which data must remain stable (i.e. not changing).

Aperture Time (t_a): Time around the clock edge for which data must remain stable (it is sum of setup and hold time).



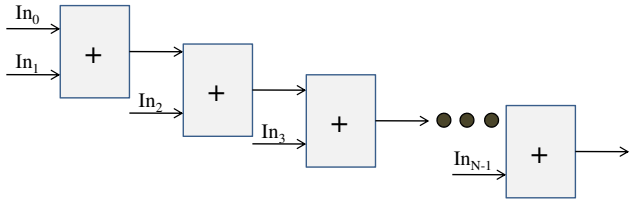
Applications of Flip-flop

- Memory design and data storage
- Logic control devices
- Shift registers
- Frequency division
- Counters

Exemplary Systems

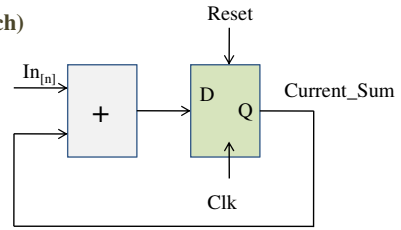
1. Implementing N inputs adder

(using 2-input adder (s) only)



What can be other alternative…?

(using sequential serial approach)

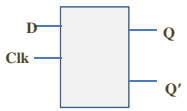


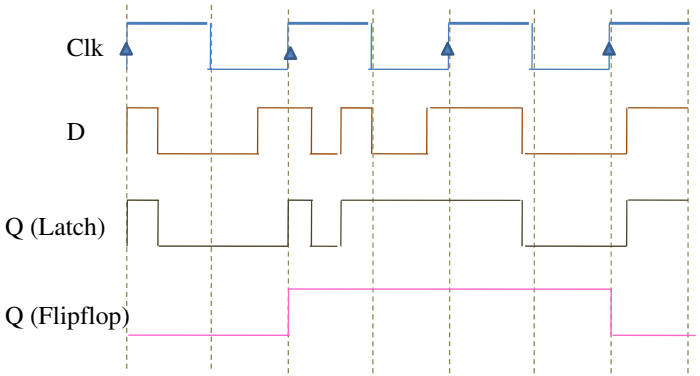
Numerical

1. Determine output if the given circuit as

(i) D- Latch (get triggered at active high level of clk signal)

(ii) D-Flipflop (get triggered at rising edge of clk signal)





Numerical

Inter-conversion between different flip-flops

2. Convert S-R to J-K Flip-flop

(Implement J-K Flip-flop using S-R flip-flop)

First Step: Write Required flip-flop at the LHS, then Q_n

⇩

Second Step: Determine Q_{n+1}

⇩

Third Step: From Excitation table, compute entire table

⇩

Fourth Step: Using K-map, determine inputs to the flip-flops.

⇩

Fifth Step: Realize the design.

Numerical

Inter-conversion between different flip-flops

2. Convert S-R to J-K Flip-flop

First Step: Write Required flip-flop at the LHS, then Q_n

J	K	Q_n
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Numerical

Inter-conversion between different flip-flops

2. Convert S-R to J-K Flip-flop

Second Step: Determine Q_{n+1}

J	K	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Numerical

Inter-conversion between different flip-flops

2. Convert S-R to J-K Flip-flop

Third Step: From Excitation table, compute entire table

J	K	Q_n	Q_{n+1}	S	R
0	0	0	0	0	x
0	0	1	1	x	0
0	1	0	0	0	x
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	x	0
1	1	0	1	1	0
1	1	1	0	0	1

Excitation Table for SR FF

Output of FF		Input of FF	
Q_n	Q_{n+1}	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

Numerical

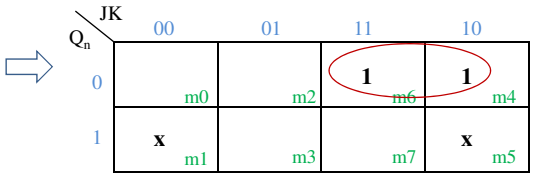
Inter-conversion between different flip-flops

2. Convert S-R to J-K Flip-flop

Fourth Step: Using K-map, determine inputs to the flip-flops.

J	K	Q_n	S	R
0	0	0	0	x
0	0	1	x	0
0	1	0	0	x
0	1	1	0	1
1	0	0	1	0
1	0	1	x	0
1	1	0	1	0
1	1	1	0	1

Boolean expression determination for 'S' using K-map



$S = JQ'_n$

Inter-conversion between different flip-flops

2. Convert S-R to J-K Flip-flop

Fourth Step: Using K-map, determine inputs to the flip-flops.

J	K	Q_n	S	R
0	0	0	0	(x)
0	0	1	x	0
0	1	0	0	(x)
0	1	1	0	(1)
1	0	0	1	0
1	0	1	x	0
1	1	0	1	0
1	1	1	0	(1)

Boolean expression determination for 'R' using K-map

Q_n \ JK	00	01	11	10
0	x m0	x m2	m6	m4
1	m1	1 m3	1 m7	m5

R = KQ_n

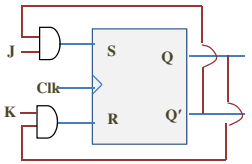
Inter-conversion between different flip-flops

2. Convert S-R to J-K Flip-flop

Fifth Step: Realize the design.

S = JQ'_n

R = KQ_n



Assignment-8

1. Convert S-R to D Flip-flop.

2. Implement J-K Flip-flop using D Flip-flop.