29/11/2022

Design Of Sequential Logic

★ SR Flip Flop (Works or NAND and NOR Gates)

→ ~~The NAND gat~~ Flip Flop is a memory element which is used to store the outputs.

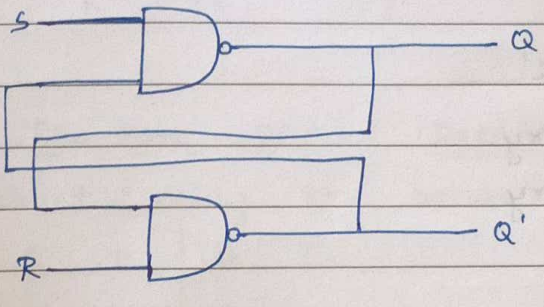
→ It has Q and Q' as its outputs, which are always complementary to each other.

→ If Q=1: Set state.
If Q=0: Reset state.

★ Types

1. SR Flip Flop (Set-Reset Flip Flop)

→



→

S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(No change)
0	1	1	0	
1	1	1	0	(No change)
0	0	1	1	(Invalid state)

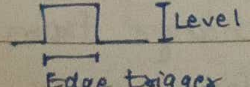
NAND:

X	Y	F
0	0	1
0	1	1
1	0	1
1	1	0

★ SR Flip Flop with ~~CLK~~ Clock.

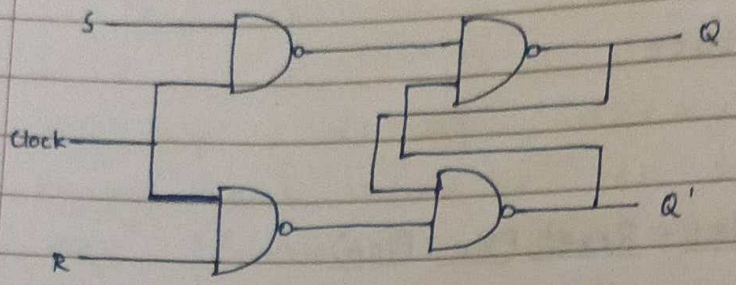
→

S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(Memory)
0	1	1	0	
1	1	1	0	(Memory)
0	0	Invalid		

★ 

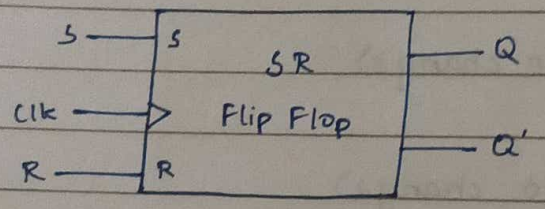
S	R	Q	Q'
0	0	Invalid / Not used	
0	1	1	0
1	0	0	1
1	1	Memory ($Q_n \to Q_n'$ or $Q_n \to Q_n$)	

→



Clk	S	R	Q	Q'
0	X	X	Memory	
1	0	0	Memory	
1	0	1	0	1
1	1	0	1	0
1	1	1	Not used.	

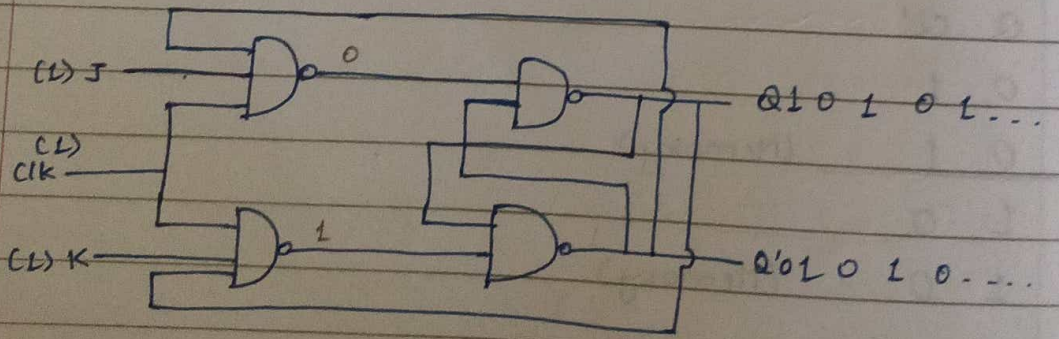
→



Symbol.

2. JK Flip Flop

→



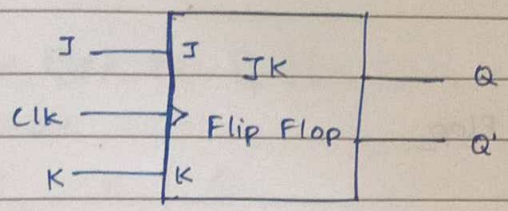
→ For J = 1, K = 1, Q and Q' rapidly change from (0, 1) to (1, 0) and so on. This is called Toggle state.

→

Clk	J	K	Q	Q'
0	x	x	Memory	
1	0	0	Memory	
1	0	1	0	1
1	1	0	1	0
1	1	1	Toggle state.	

→ JK Flip Flops are used at counters.

→



3. Data Flip Flop (DFF) (Data/Delay)

→ Constructed using the behaviour of SR Flip Flop.

→

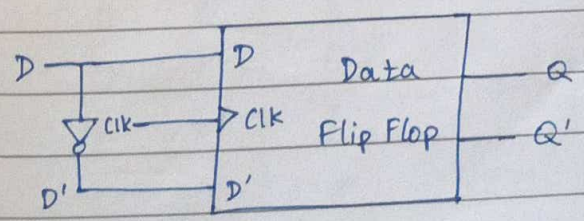
Clk	S	R	Qn	Qn'
0	x	x	Memory	
1	0	0	Memory	
1	0	1	0	1
1	1	0	1	0
1	1	1	# Invalid	

⇒

Clk	D	Qn	Qn'
0	x	Memory	
1	0	0	1
1	1	1	0
1	1	1	0

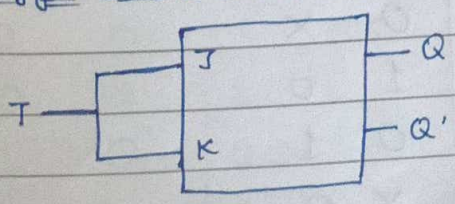
→ It works only when one input is a complement of the other input.

→



4. Toggle Flip Flop (TFF)

→



→

Clock	T	Q	Q'
0	X	Memory	Memory
1	0	Memory	Memory
1	1	Toggle	Toggle

* ~~SR Flip to~~ Conversion
→ ~~E Truth~~

* Conversion:

1. SR Flip Flop to D Flip Flop:

→ Truth Table:

Clk	S	R	Q_{n+1}
0	X	X	Memory
1	0	0	Memory
1	0	1	0
1	1	0	1
1	1	1	Invalid

→ Characteristic Table:

Qn	S	R	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

→ Excitation Table:

Qn	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

→ Conversion Table

D	Qn	Qn+1	S	R
(Truth Table)			(Excitation Table)	
0	0	0	0	X
0	1	0	0	1
1	0	1	1	0
1	1	1	X	0

2. JK Flip Flop to D Flip Flop

→ Truth table:

Clk	J	K	Qn+1
0	X	X	Memory
1	0	0	Memory
1	0	1	0
1	1	0	1
1	1	1	Toggle

→ Characteristic table

Qn	J	K	Qn+1
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

→ Excitation table:

Qn	Qn+1	J	K
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

→ Conversion Table

D	Q_n	Q_{n+1}	J	K
(Truth Table)			(Excitation Table)	
0	0	0	0	X
0	1	0	0	1
1	0	1	1	0
1	1	1	X	0

★ Registers

→ Collection of flip flops.

→ Stores single bit digital data.

→ n-bit → n Flip Flops.

→ Operations: Fetch, Decode, Execute.

1. Shift Registers

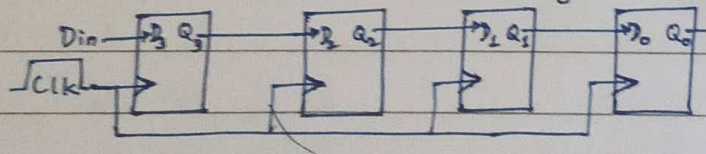
→ Capable of shifting binary data ~~to~~ in left or right directions.

→ Output of one flip flop is the input of the next flip flop. They are connected in a chain so multiple inputs are not to be given.

→ Only one clock is used (common for all flip flops).

→ Types:

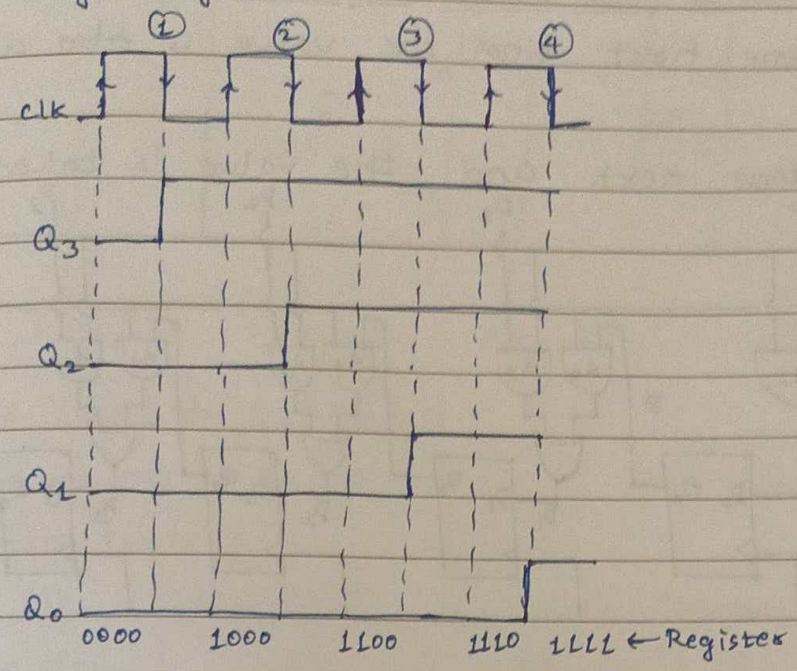
1) Serial In / Serial Out (Right Shift Register).

→ 

Ex. Data: 1 1 1 1 (LSB is the first bit taken)

Clk	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
↓	1	0	0	0
↓	1	1	0	0
↓	1	1	1	0
↓	1	1	1	1

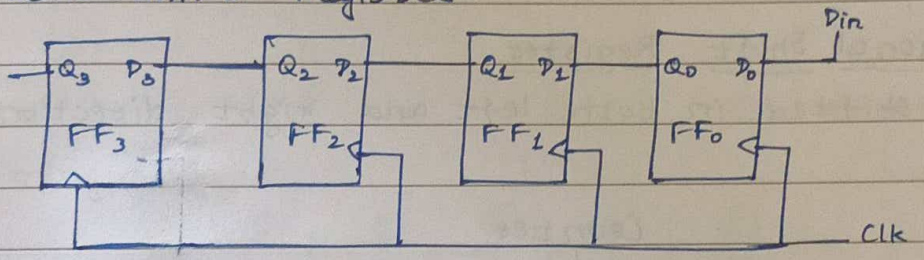
Timing diagram:



→ ① ↓ ⓪

2) Left-Shift Register.

→



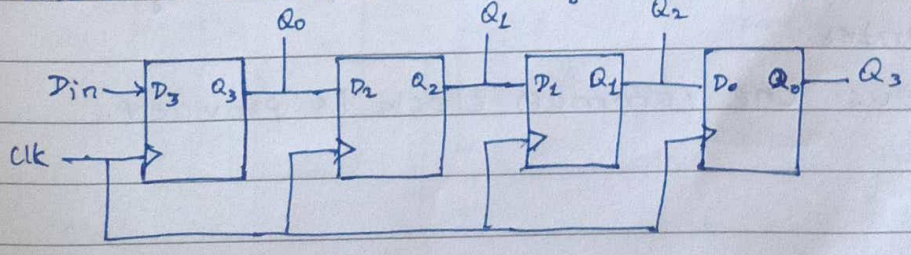
Data: 1 1 1 1 (MSB is taken as the first bit)

Clk	Q_0	Q_1	Q_2	Q_3
Initially	0	0	0	0
↓	1	0	0	0
↓	1	1	0	0
↓	1	1	1	0
↓	1	1	1	1

3) Serial In / Parallel Out Shift Register.

→ Data is entered serially but output is taken parallelly.

→

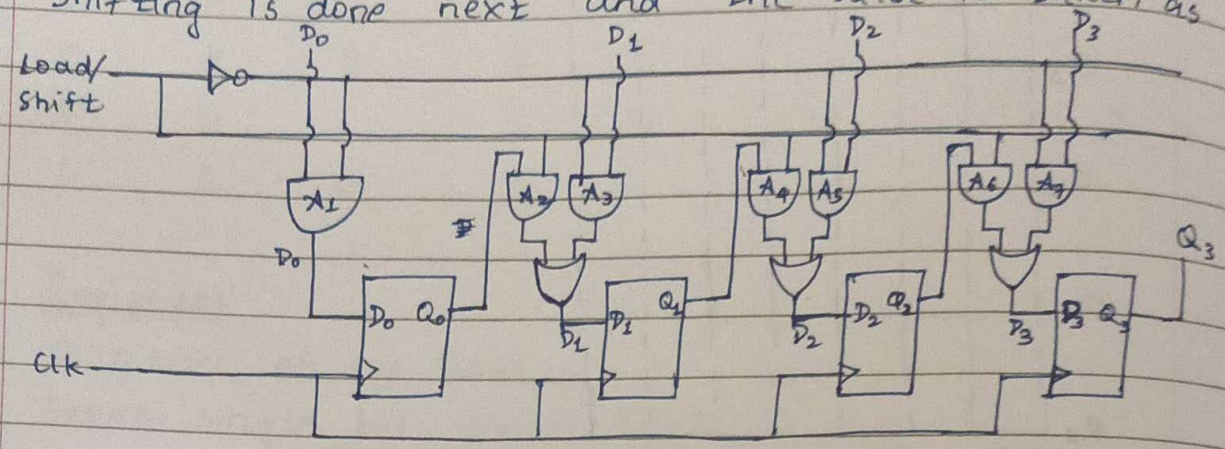


4) Parallel In Serial Out shift Register.

→ Loading is done first and the value is ~~also~~ always taken as 0.

→ Shifting is done next and the value is taken as 1.

→

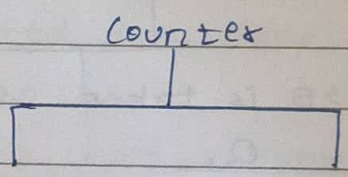


5) Parallel In Parallel Out Shift Register.

6) Bidirectional Shift Register

→ Data is shifted in both left and right directions.

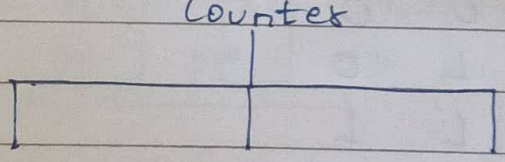
★



Asynchronous Synchronous

→ It a sequential circuit used for counting purpose.

→

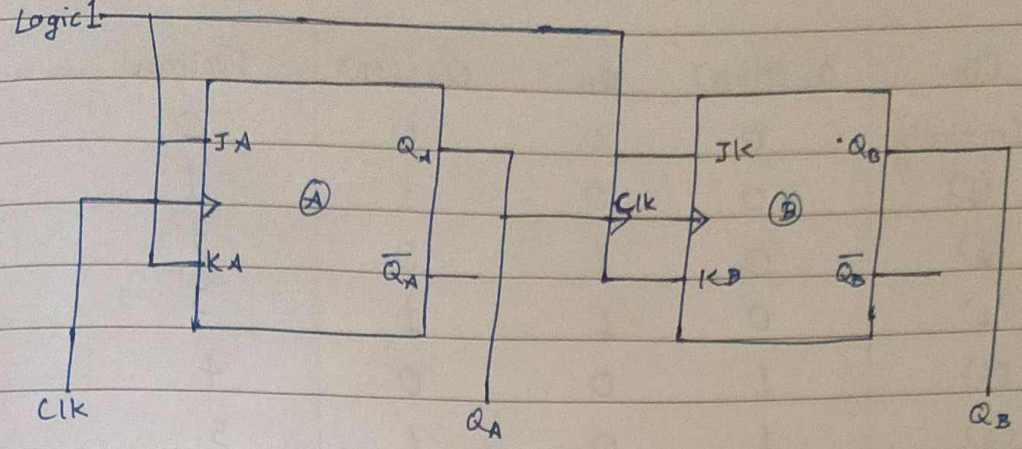


Up counter Down counter Up/Down Counter.

0, 1, 2, ... n n, ... 2, 1, 0

→ Asynchronous: Clock is provided separately. Also called ripple counter.

→ Synchronous: One common clock is provided.



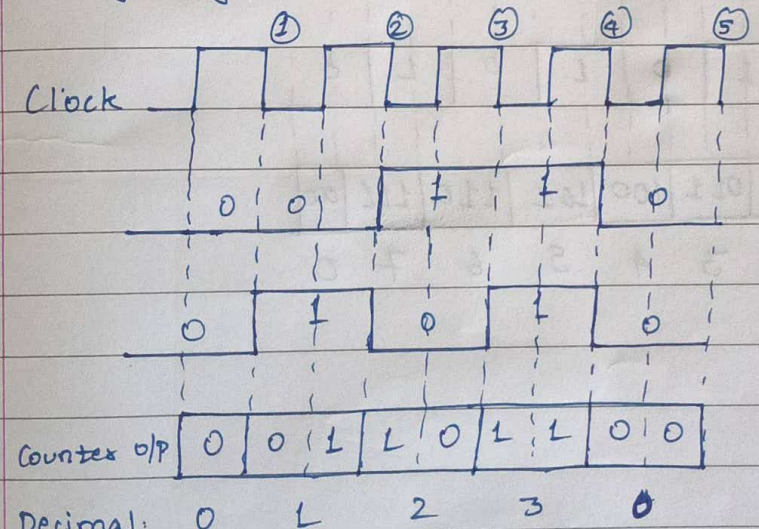
→ If $Q_A = 1 \Rightarrow Q_B$ = Toggle

If $Q_A = 0 \Rightarrow Q_B = Q_A$

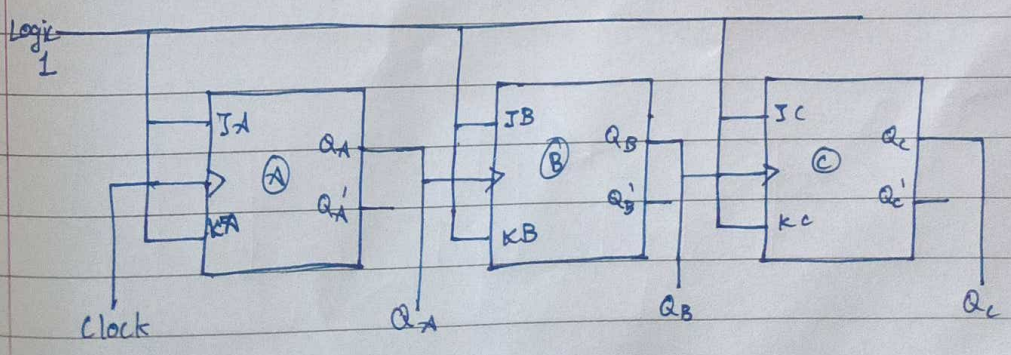
→ Truth Table:

	Clk	Q_B (MSB)	Q_A (LSB)	Decimal	
	Initially	0	0	0	
1st	(↓)	0	1	1	
2nd	(↓)	1	0	2	
3rd	(↓)	1	1	3	
4th	(↓)	0	0	0	Reset

→ Timing Diagram:



→



	Clk	Q_C (MSB)	Q_B	Q_A (LSB)	Decimal
	Initially	0	0	0	0
1st	(↓)	0	0	1	1
2nd	(↓)	0	1	0	2
3rd	(↓)	0	1	1	3
4th	(↓)	1	0	0	4
5th	(↓)	1	0	1	5
6th	(↓)	1	1	0	6
7th	(↓)	1	1	1	7
8th	(↓)	0	0	0	0 Reset

→ Timing Diagram