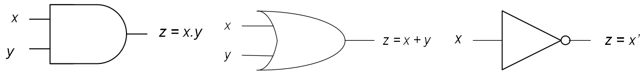
Logic Gates

- Logic gates are electronic circuits that operates on one or more input signals to produce an output signal



AND Gate

OR Gate

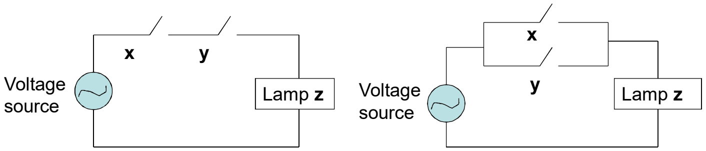
NOT Gate

x	*y*	*x.y*
0	0	0
0	1	0
1	0	0
1	1	1

x	*y*	*x + y*
0	0	0
0	1	1
1	0	1
1	1	1

x	*x'*
0	1
1	0

Switching Circuit

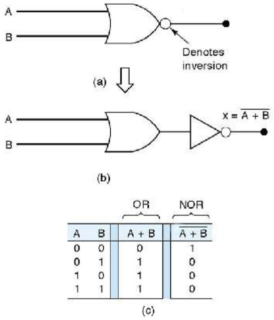


Switches in series – logic AND

Switches in parallel – logic OR

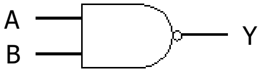
NOR Gate

- NOR Symbol, Equivalent Circuit, Truth Table



		OR	NOR
A	B	A + B	$\overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

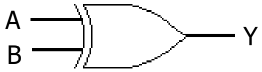
NAND Gate



- This is a NAND gate. It is a combination of an AND gate followed by an inverter. Its truth table shows this…

A	B	Y
0	**0**	**1**
0	**1**	**1**
1	**0**	**1**
1	**1**	**0**

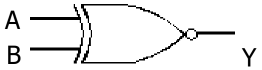
XOR Gate (Exclusive-OR)



- This is a XOR gate.
- The switching algebra symbol for this operation is $\oplus$, i.e. $A \oplus B$.
- $A \oplus B = A'B+AB'$.

A	B	Y
0	**0**	**0**
0	**1**	**1**
1	**0**	**1**
1	**1**	**0**

XNOR Gate (Exclusive-NOR)

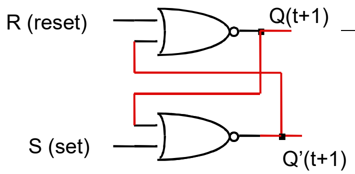


- This is an Exclusive-NOR gate
- The complement of the XOR gate.
- The switching algebra symbol for this operation is $\odot$, i.e. $A \odot B$.
- $A \odot B = AB+A'B'$.

A	B	Y
0	**0**	**1**
0	**1**	**0**
1	**0**	**0**
1	**1**	**1**

- NAND, NOR gates are called universal gate.
- AND, OR and NOT are called basic gates.
- XOR(EX-OR), XNOR(EX-NOR) are called derived gates.

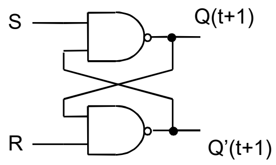
S-R Latch with NOR Gates



S	R	Q(t+1)	
0	0	Q(t)	**Memory**
1	0	1	**Set State**
0	1	0	**Reset State**
1	1	**Undefined state**	

- S-R latch made from cross-coupled NORs
- If Q = 1, set state
- If Q = 0, reset state
- Usually S=0 and R=0 is called Memory state
- S=1 and R=1 generates undefined or indeterminate results.

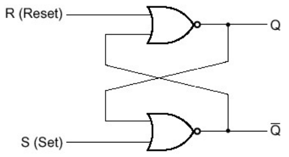
S-R Latch with NAND Gates



S	R	Q(t+1)
1	1	Q(t) **Memory**
0	1	1 **Set state**
1	0	0 **Reset state**
0	0	**Undefined state**

- S-R Latch made from cross-coupled NANDs
- Sometimes called S'-R' latch
- Usually S=1 and R=1 is called Memory state
- S=0 and R=0 generates undefined or indeterminate results.

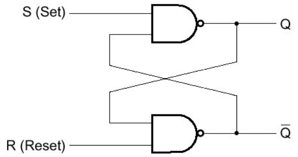
S-R Latches



(a) Logic diagram

S	R	Q	$\overline{Q}$	
1	0	1	0	Set state
0	0	1	0	
0	1	0	1	Reset state
0	0	0	1	
1	1	0	0	Undefined

(b) Function table

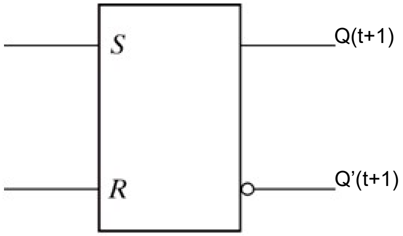


(a) Logic diagram

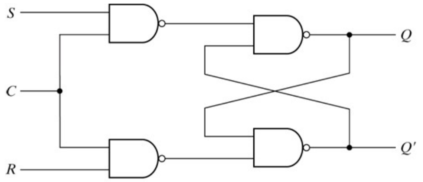
S	R	Q	$\overline{Q}$	
0	1	1	0	Set state
1	1	1	0	
1	0	0	1	Reset state
1	1	0	1	
0	0	1	1	Undefined

(b) Function table

Graphic symbol of S-R Latch



S-R Latch with control input



(a) Logic diagram

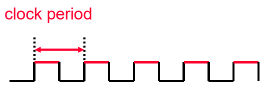
C	*S*	*R*	Next state of *Q*
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; Reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

(b) Function table

- S-R Latch with control input is called Flip-Flop
- It is used to store one bit of information
- Occasionally, desirable to avoid latch changes
- C = 0 disables all latch state changes
- Control signal enables data change when C = 1
- Right side of circuit same as ordinary S-R latch.

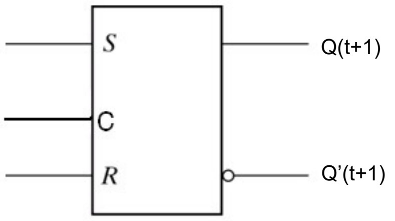
Clocks and synchronization

- A clock is a special device that whose output continuously alternates between 0 and 1.

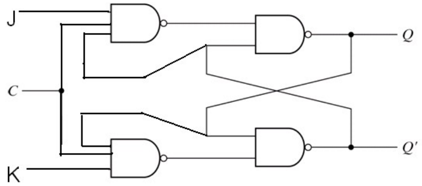


- The time it takes the clock to change from 1 to 0 and back to 1 is called the clock period, or clock cycle time.
- The clock frequency is the inverse of the clock period. The unit of measurement for frequency is the hertz.

Graphic symbol of S-R Flip-Flop



J-K Flip-Flop



(a) Logic diagram

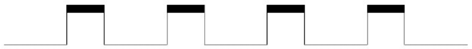
J	K	Q(t+1)
0	0	Q(t) No change
0	1	0 Reset
1	0	1 Set
1	1	Q'(t) Complement

(b) Function table

- S-R Latch with control input is called Flip-Flop
- Replace the two input NAND gate with the three input NAND gate to get a J-K Flip Flop from a S-R FF.
- C = 0 disables all the data changes
- Control signal C=1 enables data change when C = 1

Flip-Flops

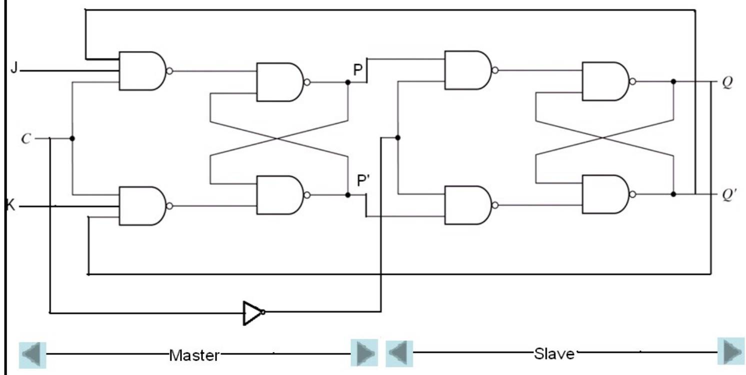
- A flip-flop is a state of a latch that can be switched by momentary change in the control input.
- This momentary change is called a trigger and the transition it causes is said to trigger the flip-flop.
- The flip-flop is triggered every time the pulse goes to a high or logic level 1.



Response to positive level(Level Triggering)

- The problem with the flip-flops is that as long as the input clock pulse remains at this level, any changes in the input data will cause a change in the output and the state of the latch.
- Race around condition or Racing.

Master Slave J-K Flip-Flop



(a) Logic diagram

The Master-Slave Flip-Flop is basically two gated SR flip-flops connected together in
a series configuration with the slave having an inverted clock pulse. The outputs
from Q and Q from the "Slave" flip-flop are fed back to the inputs of the "Master" with the outputs of the "Master" flip-flop being connected to the two inputs of the "Slave" flip-flop. This feedback configuration from the slave's output to the master's input gives the characteristic toggle of the JK flip-flop .

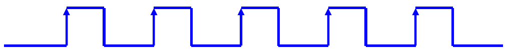
The input signals J and K are connected to the gated "master" SR flip-flop which "locks" the input condition while the clock (Clk) input is "HIGH" at logic level "1". As the clock input of the "slave" flip-flop is the inverse (complement) of the "master" clock input, the "slave" SR flip-flop does not toggle. The outputs from the "master" flip-flop are only "seen" by the gated "slave" flip-flop when the clock input goes "LOW" to logic level "0".

When the clock is "LOW", the outputs from the "master" flip-flop are latched and any additional changes to its inputs are ignored. The gated "slave" flip-flop now responds to the state of its inputs passed over by the "master" section. Then on the "Low-to-High" transition of the clock pulse the inputs of the "master" flip-flop are fed through to the gated inputs of the "slave" flip-flop and on the "High-to-Low" transition the same inputs are reflected on the output of the "slave" making this type of flip-flop edge or pulse-triggered.

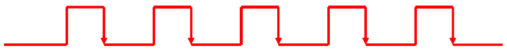
Then, the circuit accepts input data when the clock signal is "HIGH", and passes the data to the output on the falling-edge of the clock signal. In other words, the **Master-Slave JK Flip-flop** is a "Synchronous" device as it only passes data with the timing of the clock signal.

The Solution

- A clock pulse goes through two transition from 0 to 1 and the return from 1 to 0
- The solution: by changing the operation of a flip-flop to trigger it only during a signal transition.
- Two types of transition: the positive transition (from 0 t0 1) and the negative transition (from 1 to 0)

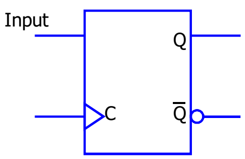


Positive-edge response

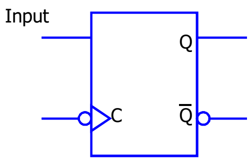


Negative-edge response

Flip-Flop Symbols

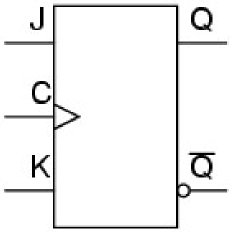


Positive Edge Triggered Flip Flop

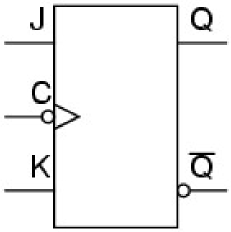


Negative Edge Triggered Flip Flop

J-K flip flop symbols

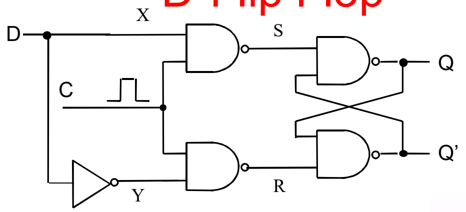


Positive-edge Triggered FF

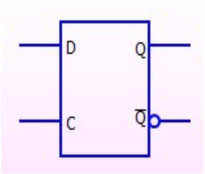


Negative-edge Triggered FF

D-Flip Flop



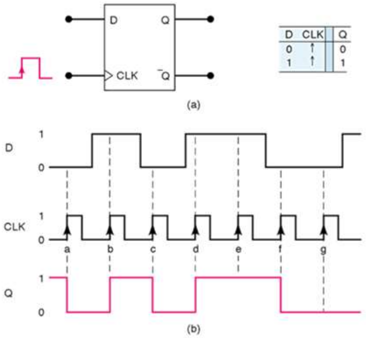
D	C	Q
0	1	0
1	1	1
X	0	Q_0 (Memory)



Input value D is passed to output Q when C is high
Input value D is ignored when C is low

Clocked D Flip-Flop

- Stores a value on the positive edge of *C*
- Input changes at other times have no effect on output



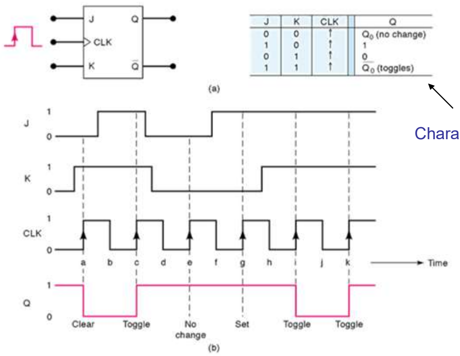
D	CLK	Q
0	↑	0
1	↑	1

(a)

(b)

Clocked J-K Flip Flop

- Two data inputs, J and K
- J -> set, K -> reset, if J=K=1 then toggle output



J	K	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	$\overline{Q}_0$ (toggles)

Characteristic Table

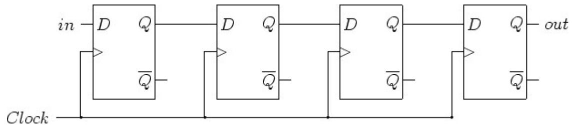
Registers

- A flip-flop can store only one bit data (0 or 1)
  - A single-bit register
- Register consists of a group of flip-flops and gates
- An n-bit register consists of n-bit flip-flops
  - store n-bit information
- Shift register is used to store and shift data.

Types of Shift Register

- Serial in serial out (SISO)
- Serial in Parallel out (SIPO)
- Parallel in serial out (PISO)
- Parallel in Parallel out (PIPO)

Shift Registers(SISO)



- Shifts binary information in one or both directions.
- On the positive edge of the first clock pulse, the signal on the in is latched in the first FF.
- On the next clock pulse, the data of the first FF is stored in the second FF, and the data present at the in is stored is the first FF, *etc.*

Counters

- Binary counter: follows the binary number sequence.
- An n-bit binary counter:
  - Consists of **n** flip-flops
  - Can count in binary from **0** to **2^n - 1**
- A counter is a register that goes through a sequence of states.

Counters

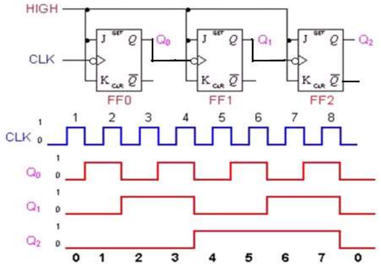
- Counter categories:
  - Ripple counters
  - Synchronous counters
- **Ripple counters:** The flip-flop's output transition triggers other flip-flops.
- **Synchronous counters:** A common clock triggers all flip-flops simultaneously rather than one at a time in succession as in ripple counters.

Binary Ripple Counter

- All the J and K inputs of all the flip-flops are connected to a logic 1.
- Each flip-flop complement if the signal in its C input goes through a negative transition.
- The flip-flop holding the LSB receives the incoming count pulses.
- The count starts with binary 0 and increments by one with each count pulse input.

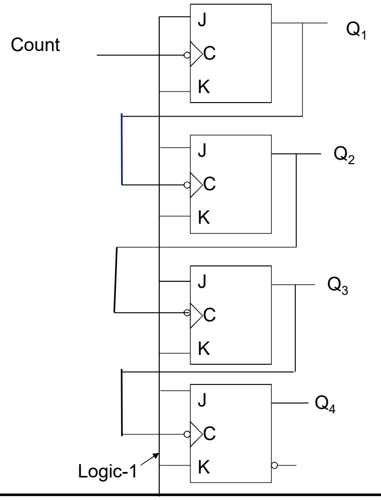
3-bit Binary Ripple Counter

Q_2	Q_1	Q_0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1
0	0	0

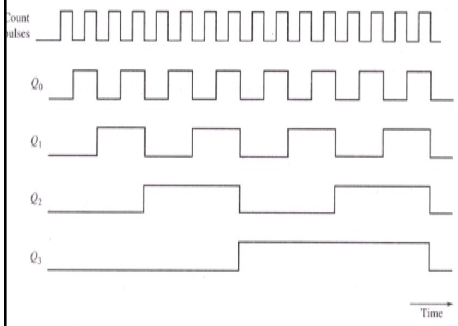


Q_0 is complemented with the count pulse. Since Q_0 goes from 1 to 0, it triggers Q_1 and complements it. As a result, Q_1 goes from 1 → 0, which in turn complements Q_2 changing it from 0 → 1. Q_2 does not trigger Q_3 because Q_2 produces a positive transition. The flip-flops change one bit at a time in succession and the signal propagates through the counter in a ripple fashion from one stage to the next.

4-bit Binary Ripple Counter



4-bit Binary Ripple Counter



Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1
0	0	0	0

Problems with Ripple Counter

- **Asynchronous or ripple counters** are arranged in such a way that the output of one flip flop changes the state of the next.

- In a long chain of ripple counter stages, the last flip flop changes its state considerably later than the first FF due to propagation delays in each stage.

- Problems occur if this delay is longer than the response time of other logic elements connected to the circuit.