Subject name: Logic Design
Subject code: FET 1021
Semester: 3rd

Lecture 28:
Storage Elements : D and T flip-flops

Department of Electronics & Communication Engineering,
Institute of Technical Education & Research,
S 'O' A (Deemed to be) University,
Khandagiri, Bhubaneswar, Orissa-751030

Topics to be covered:

- D flip-flops, Graphical symbol, Logic diagram, Function Table, Characteristics Table, Characteristics Equation, Excitation table, and HDL description.
- T flip flops, Graphical symbol, Logic diagram, Function Table, Characteristics Table, Characteristics Equation, Excitation table, and HDL description.

Learning outcome:

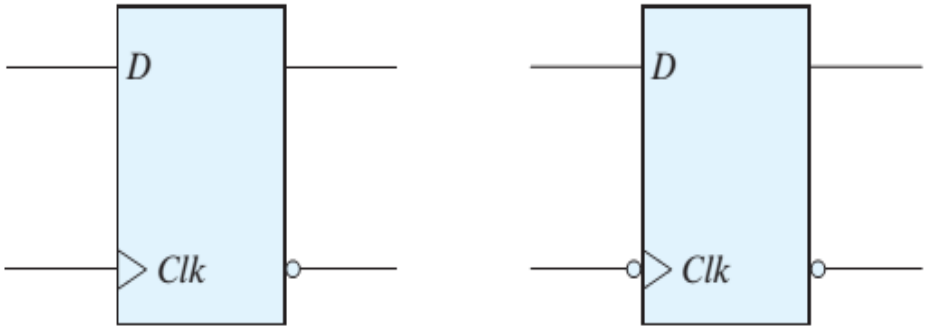
- Able to gather knowledge about the D and T Flip-flops.
- Able to design D & T FFs using HDL description

D flip-flop

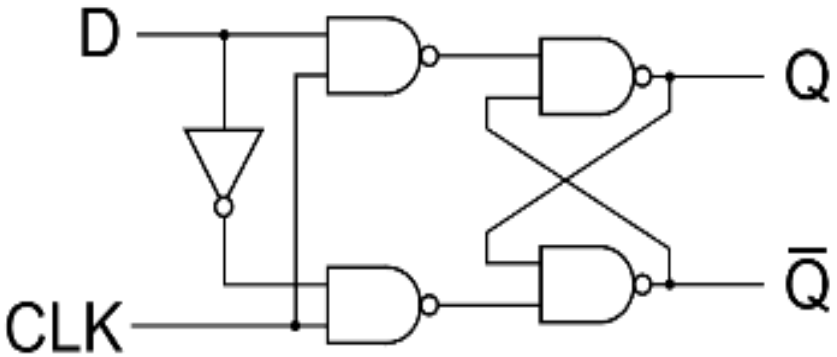
- An SR flip-flop is rarely used in actual sequential logic because of its undefined outputs for inputs R= S= 1.
- It can be modified to form a more useful circuit called D flip-flop.
- The D flip-flop has only a single data input D as shown in the circuit diagram.
- That data input is connected to the S input of an SR flip-flop, while the inverse of D is connected to the R input.
- The D flip-flop also has a clock pulse as an input.
- It can be +ve edge triggered or –ve edge triggered.

D flip-flop

Graphical symbol:



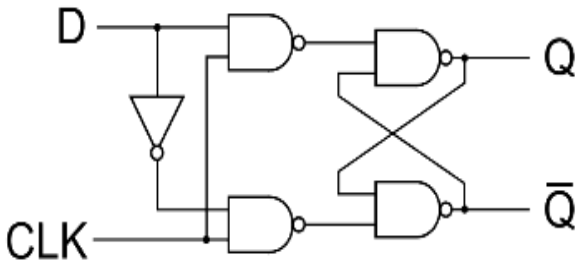
(a) Positive edge (b) Negative edge



Logic Diagram of D flip-flop using NAND logic

D flip-flop

Operation:

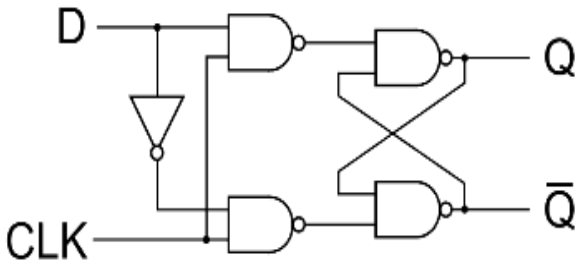


- By connecting an inverter (NOT gate) to the SR flip-flop we can SET and RESET the flip-flop.
- This complement avoids the ambiguity inherent in the SR latch when both inputs are LOW.
- If this data input is held HIGH the flip flop would be SET and when it is LOW the flip flop would change and become RESET. However, this would be rather pointless since the output of the flip flop would always change on every pulse applied to this data input.
- To avoid this additional input called the CLOCK is used to isolate the data input from the flip-flop's latching circuitry after the desired data has been stored.
- The effect is that D input condition is only copied to the output Q when the clock input is active.
- The D flip flop‖ will store and output whatever logic level is applied to its data terminal so long as the clock input is HIGH

D flip-flop

Function or truth table:

Clock	D	Q_{t+1}	Description
↓(0)	X	Q_t	No change
↑(1)	0	0	Reset Q=0
↑(1)	1	1	Set Q=1



Characteristic Table:

State		Input
Present state	Next state	D
0	0	0
0	1	1
1	0	0
1	1	1

Excitation Table:

Q_t	D	Q_{t+1}
0	0	0
0	1	1
1	0	0
1	1	1

Characteristic Equation: $Q_{(t+1)} = D$

Note: ↓ and ↑ indicates direction of clock pulse as it is assumed D-type flip-flops are edge triggered.

D flip-flop

D flip-flop vs D clocked latch

D flip flop:

- Triggered on the +ve edge of the clock.
- Output Q (and state) changes on a time instant.

D clocked latch:

- Output changes with D while clock is High.
- Output changes during a window of time.

D flip-flop

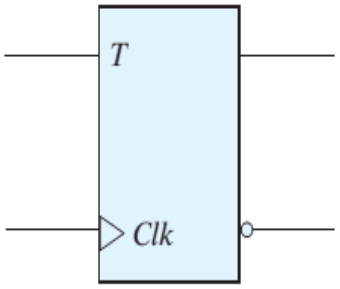
HDL code

```
module d_flip_flop(q,qb,clk,rst,d);
        input clk, rst, d;
        output q, qb;
        reg q,qb;
        always@(posedge clk)
                begin
                if (rst==1)
                        q<=0;
                else
                        q<=d;
                qb<=~q;
                end
end module
```

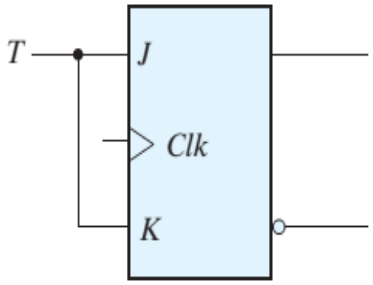
T flip-flop

- The T flip-flop has one input in addition to the clock
- T stands for toggle for the obvious reason.
- When T is asserted (T = 1), the flip-flop state toggles back and forth, and when T is de-asserted, the flip-flop keeps its current state.
- The T flip-flop can be constructed using a D flip-flop with the two outputs Q and Q' feedback to the D input through a multiplexer that is controlled by the T input.
- The T flip-flop is a single input version of the JK flip-flop.
- The T flip-flop is obtained from the JK type if both inputs are tied together

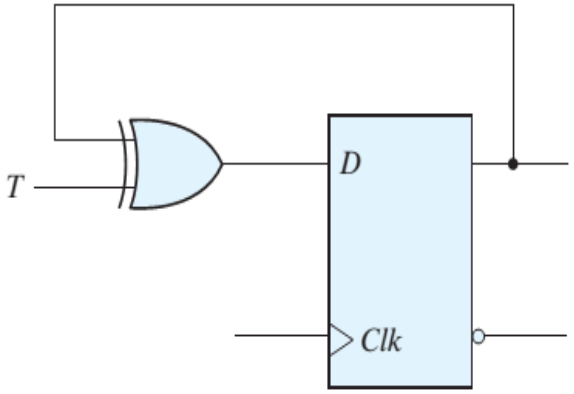
T flip-flop



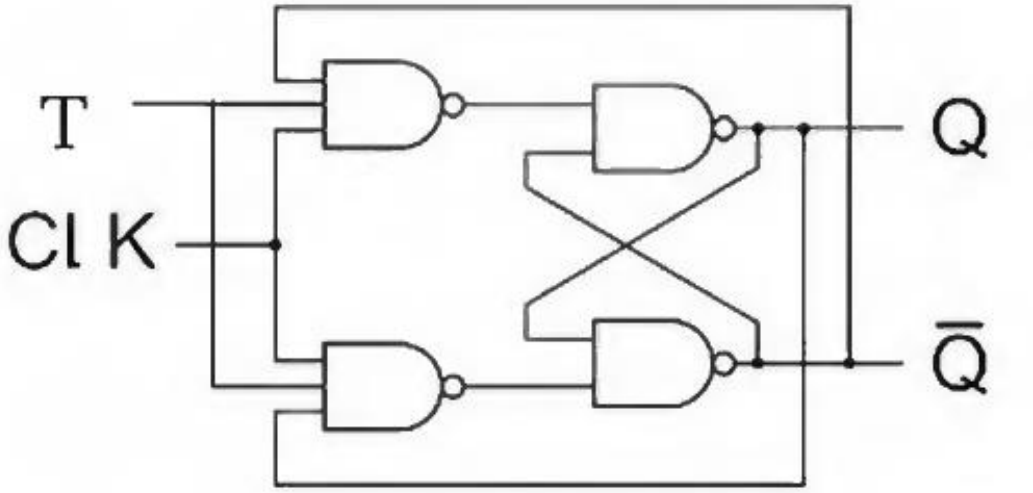
(a) Graphical Symbol



(b) From JK flip-flop



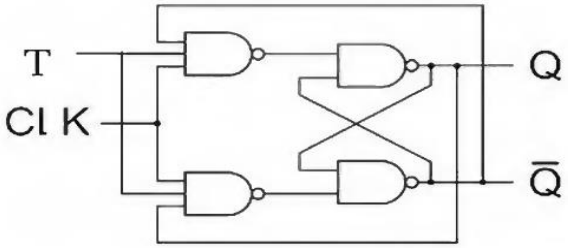
(c) From D flip-flop



Logic Diagram of T flip-flop using NAND logic

T flip-flop

Operation:

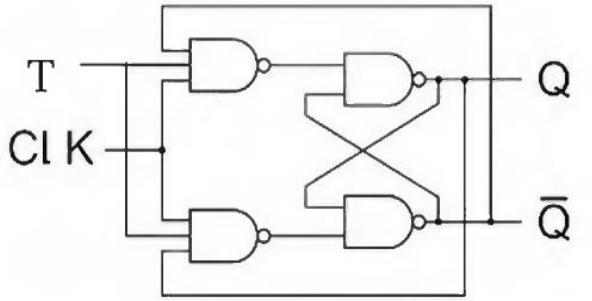


- If the output Q = 0, then the upper NAND is in enable state and lower NAND gate is in disable condition.
- This allows the trigger to pass the S inputs to make the flip –flop in SET state i.e. Q =1.
- If the output Q = 1, then the upper NAND is in disable state and lower NAND gate is in enable condition. This allows the trigger to pass the R inputs to make the flip –flop in RESET state i.e. Q=0.
- In simple terms, the operation of the T flip –flop is
- When the T input is low, then the next sate of the T flip flop is same as the present state.
- T = 0 and present state = 0 then the next state =0
- T = 1 and present state = 1 then the next state =1
- When the T input is high and during the positive transition of the clock signal, the next state of the T flip –flop is the inverse of present state.
- T = 1 and present state = 0 then the next state =1
- T = 1 and present state = 1 then the next state =0

T flip-flop

Function or truth table:

Clock	T	Q_{t+1}	Description
↓(0)	X	Q_t	Memory
↑(1)	0	Q_t	Memory
↑(1)	1	Q_t'	Toggle



Characteristic Table:

Q_t	T	Q_{t+1}	Description
0	0	0	No change
0	1	1	Toggle
1	0	1	No change
1	1	0	Toggle

Excitation Table:

State		Input
Present state (Q_t)	Next state (Q_{t+1})	T
0	0	0
0	1	1
1	0	1
1	1	0

Characteristic Equation: $Q_{(t+1)} = T.Q' + T'.Q$

Note: ↓ and ↑ indicates direction of clock pulse as it is assumed T-type flip-flops are edge triggered.

T flip-flop

HDL code

```
module tff(t,clk,rst, q,qb);
        input t,clk,rst;
        output q,qb;
        reg q,qb;
        reg temp=0;
        always@(posedge clk,posedge rst)
        begin
        If (rst==0)
                begin
                if(t==1) begin
                        temp=~ temp;
                end
        else
                temp=temp;
        end
        q=temp;qb=~temp;
        end
end module
```

Assignment

1. Design T flip-flop using JK flip-flop.

2. Design D flip-flop using JK flip-flop.

3. What are the applications of D and T flip-flops?

Thank you
End of Lecture-28