Bimal Parajuli (20BDS0405)

DIGITAL LOGIC DESIGN

FLIP-FLOPS

EXPERIMENT #10

Objectives:

1. To become familiar with flip-flops.

2. To implement and observe the operation of different flip-flops.

Apparatus:

- IC type 7400 quad 2-input NAND gate
- IC type 7410 triple 3- input NAND gate
- IC type 7476 dual JK master-slave flip-flops.
- IC type 7474 dual D positive-edge-trigged flip-flops.
- Dual trace oscilloscope.

Softwares Used

- LogicWorks 5
- Cadence Capture + PsPice

Procedure:

1. ***In the pre-lab using LogicWorks construct the circuit shown in Fig. 1***

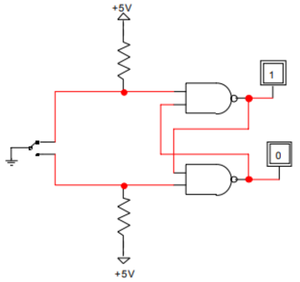
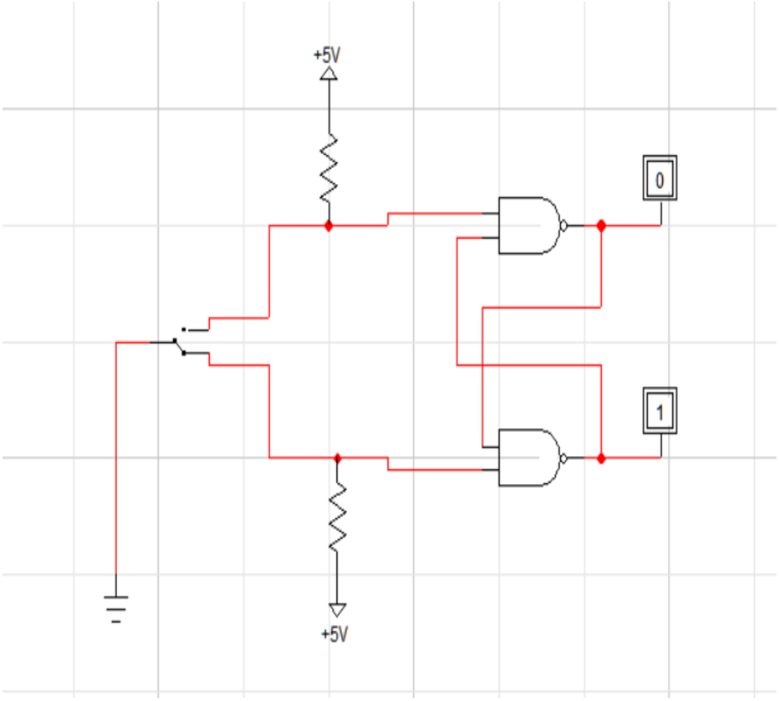
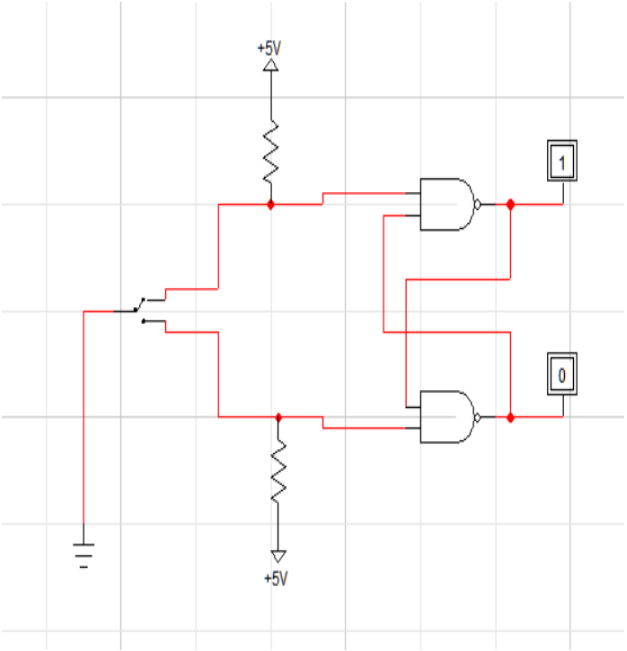


Fig :1

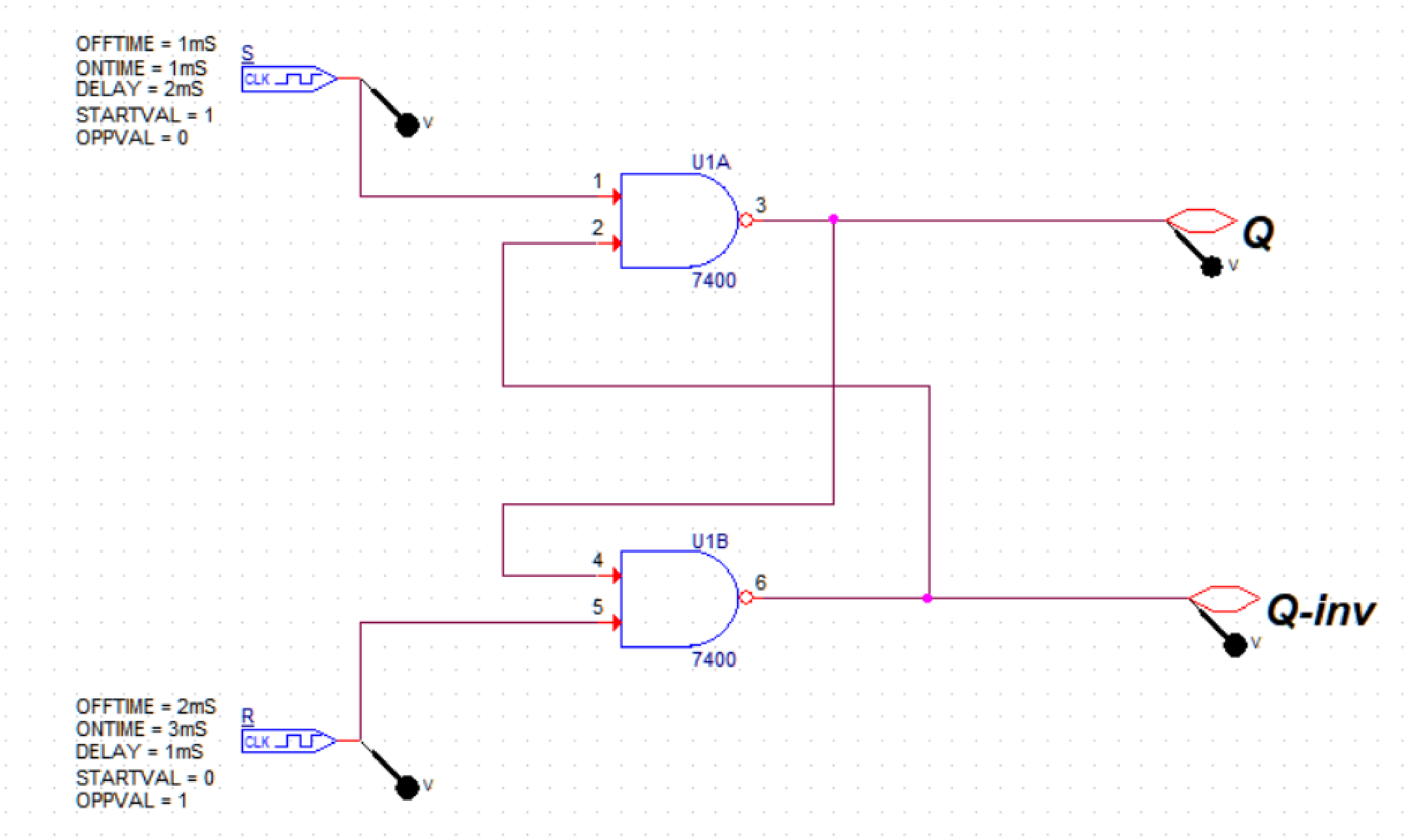


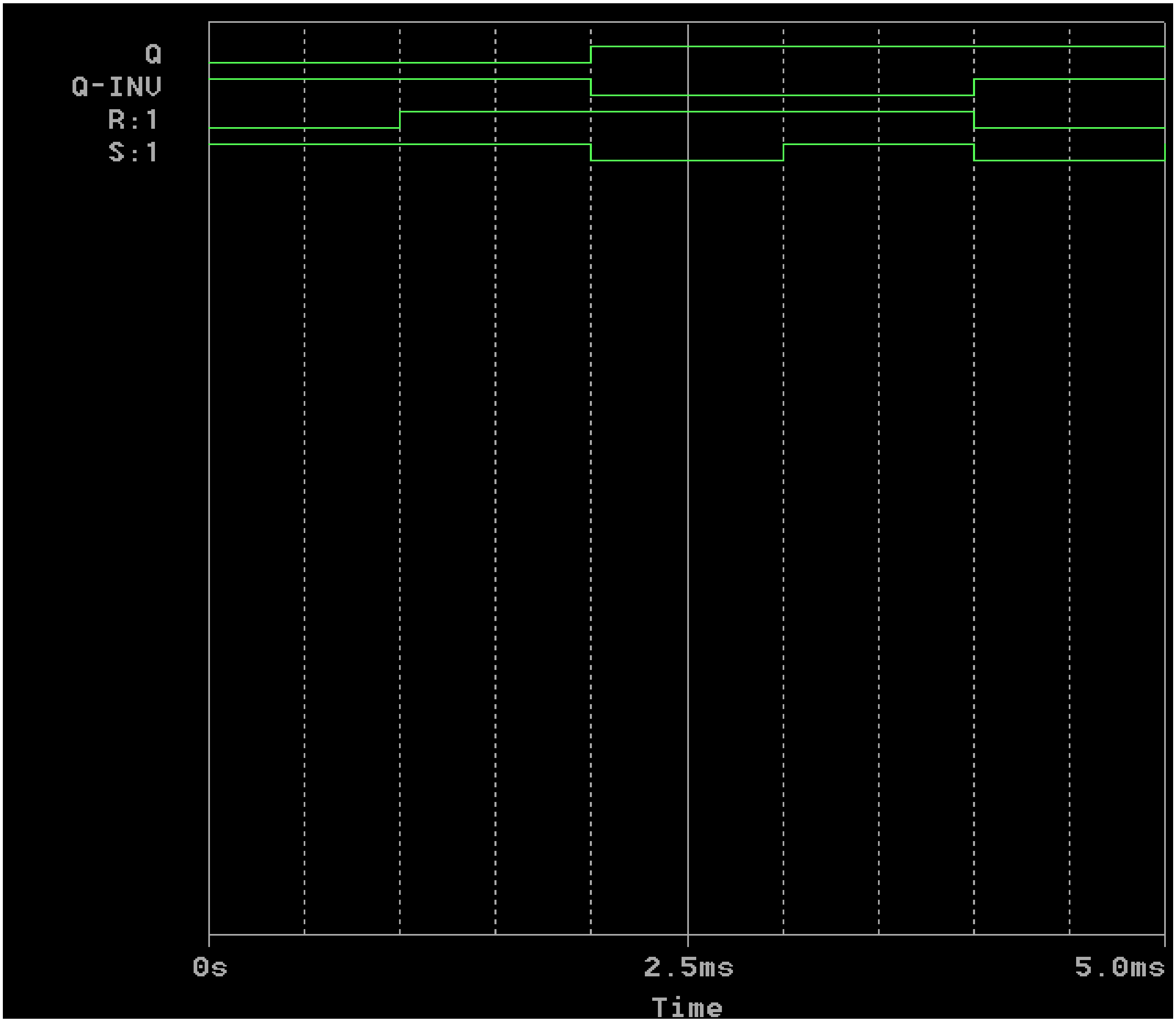


Where we could use generic NAND gates or 74-00 and Binary Probes to simulate LEDs. Finally, we use SPDT for the bouncing switch. Using the simulated circuit fill in the truth table.

S	R	Q	Q´	Cases
1	0	0	1	Reset
1	1	0	1	Memory
0	1	1	0	Set
1	1	1	0	Memory
0	0	1	1	Invalid

On doing laboratory simulation using OrCAD PsPice, we obtain the following circuit which is tabulated in the above table




Q
Q-INV
R:1
S:1
0s
2.5ms
5.0ms
Time

In the Lab, Build the RS latch shown in fig.2. Use SPDT switch S2 as a bouncing switch. Q and Q' Outputs are connected to LED 'S of the PB-503. Verify the truth table experimentally.

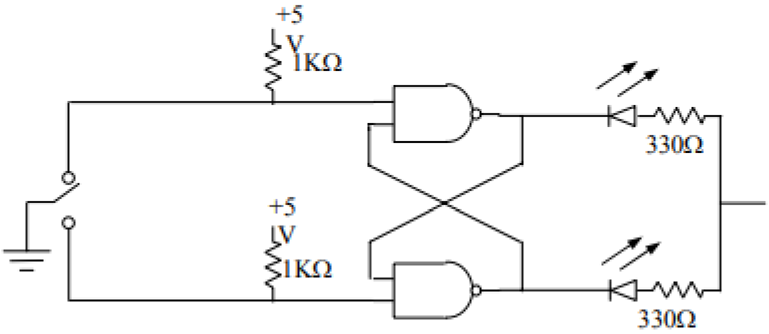
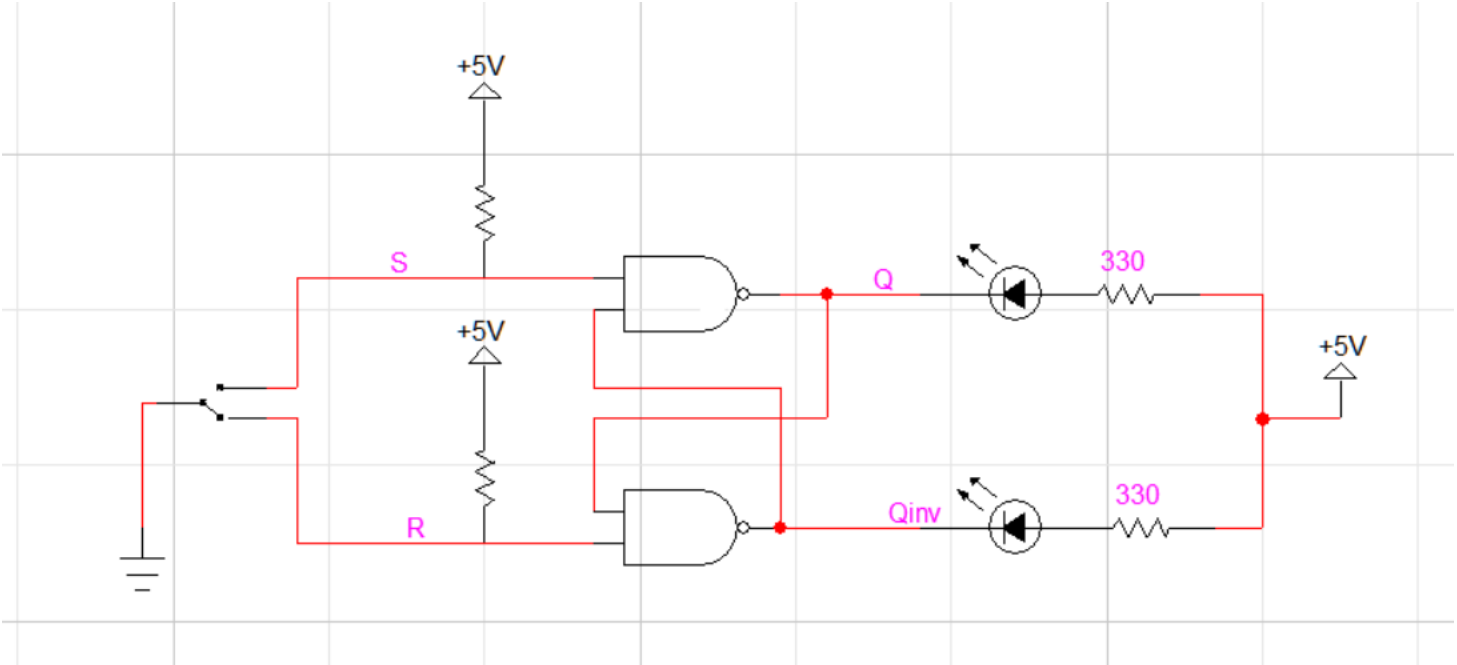
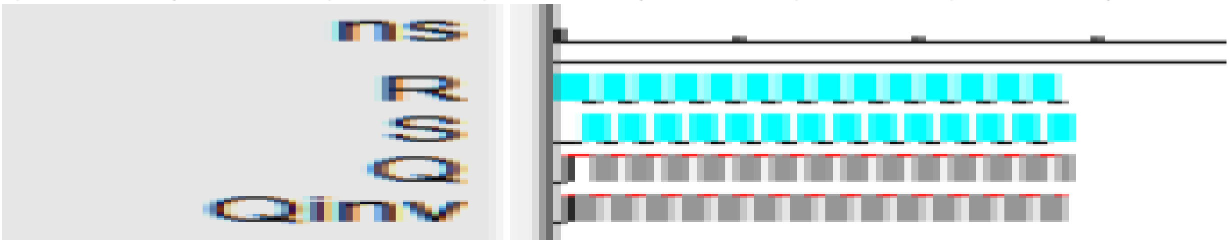


Fig. 2

Following is the SR latch made in laboratory using SPDT switch input in LogicWorks 5





2. ***Modify the basic R-S into a D latch by adding the steering gates and the inverter shown in Fig 3.***
   Connect the D input to the pulse generator of the Digi designer and set it at 1 Hz.
   Connect the enable input to a high through 1k resistor. Observe the output; obtain the truth table experimentally then change the enable to a low.
   Is the enable an active high or an active low? Leave the enable low and place a momentary short to ground first on one output and then on the other. What happens?

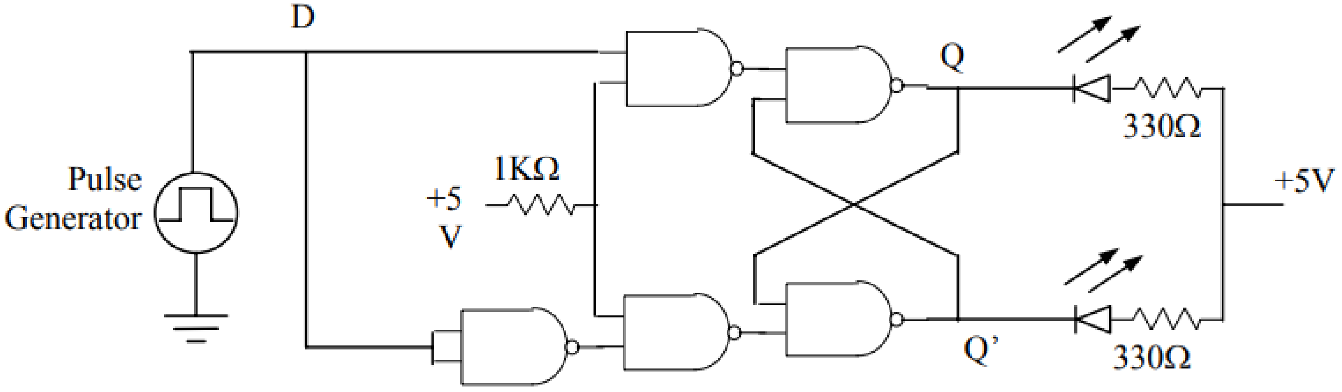
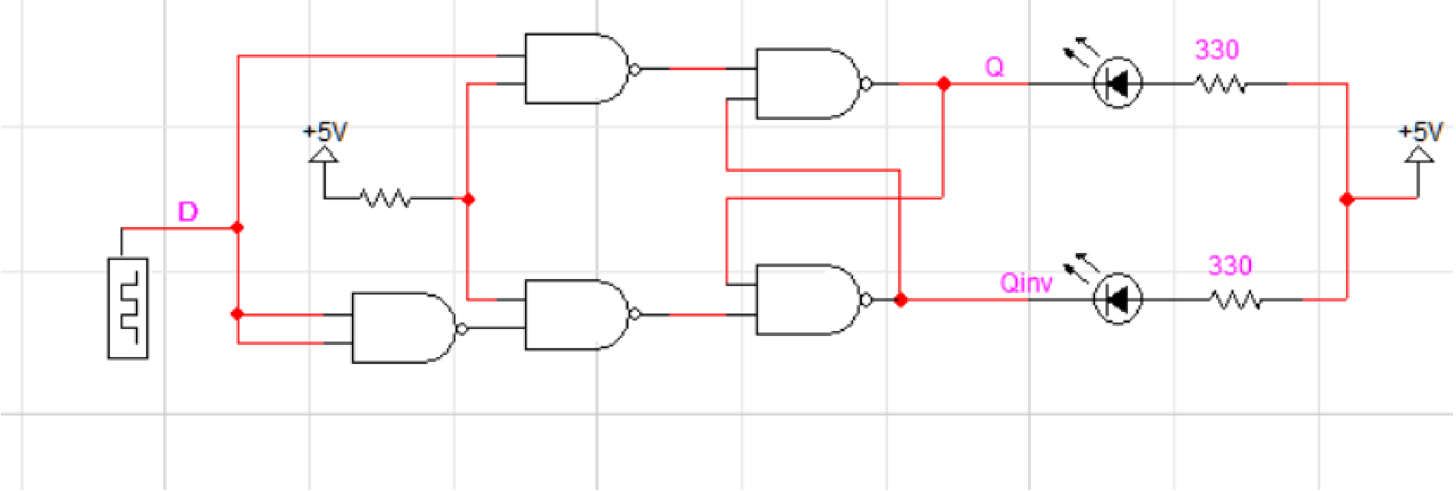
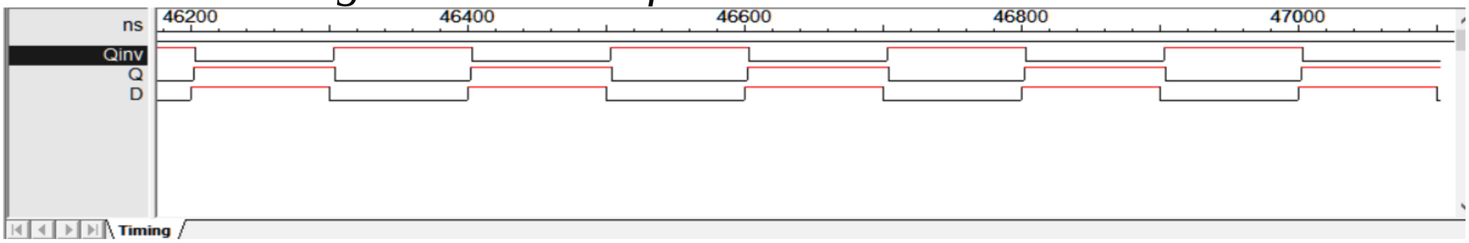


Fig: 3

Following is the Basic D Latch created by modifying SR latch.



Following is the result of its simulation in time domain:

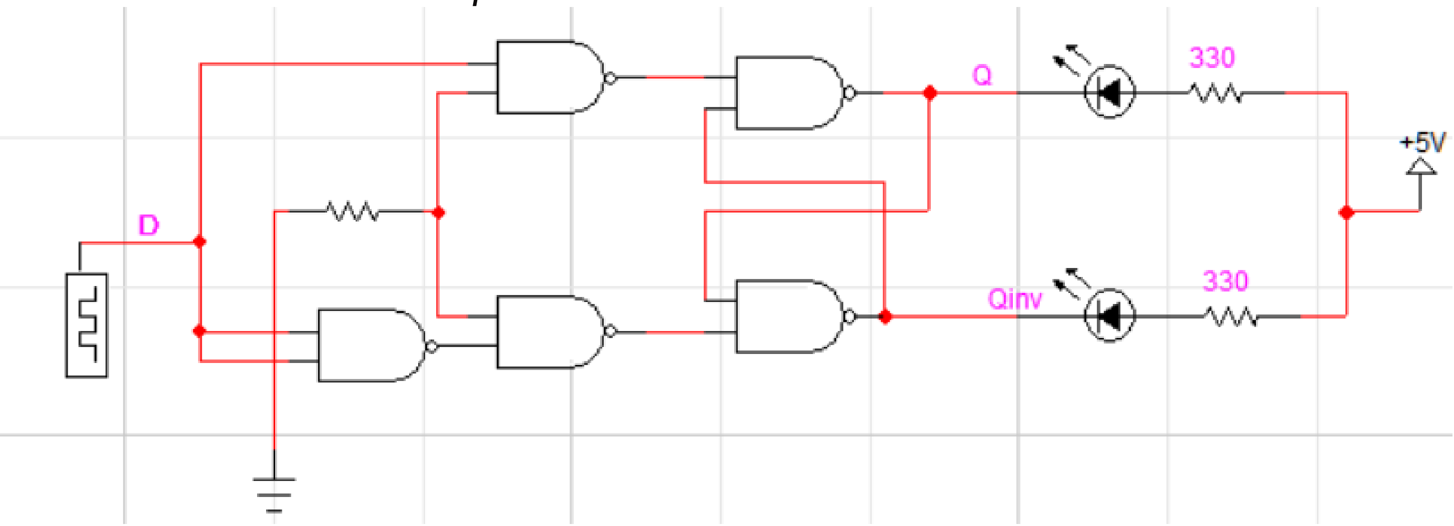


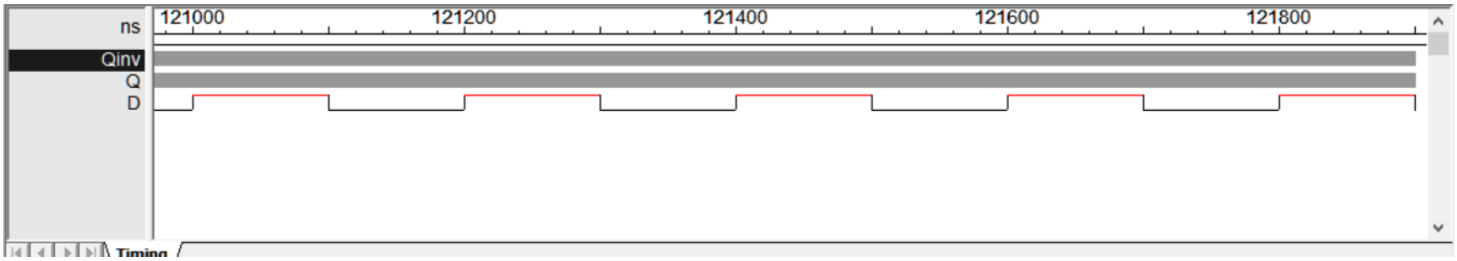
Thus the following behavior is obtained:

Enable	*D*	*Q*	*Q’*	*Observation*
0	*X*	*Retained*	*Retained*	*No change*
1	*0*	*0*	*1*	*Reset*
1	*1*	*1*	*0*	*Set*

The enable of D latch is active high one. If we set enable as low, the output of the flip-flop stays the same and independent to the input value D.

Which is demonstrated as follows:





2. ***The 7476 is a dual JK master-slave flip-flop with preset and clear inputs. The function table given in table 1 defines the operation of the flip-flop. The +ve transition of the CLOCK (CP) pulse changes the master flip-flop, and the (-ve) transition changes the slave flip-flop as well as the output of the circuit. In LogicWorks the chip 7476 is not available, however, the generic JK flip-flop behave in exactly the same way as the 7476. The "S" represents the Preset, the "R" represents the Clear, and C represents the clock pulse (CP). Verify the table by connecting Binary switches to R, S, J, K, and C. Notice that only the negative edge of the clock affects the outputs (Q, and Q').***

Table 1

Input					Output	
Preset	Clear	Clock	J	K	Q	Q’
0	1	X	X	X	1	0
1	0	X	X	X	0	1
0	0	X	X	X	1	1
1	1	⎍↓	0	0	No change	
1	1	⎍↓	0	1	0	1
1	1	⎍↓	1	0	1	0
1	1	⎍↓	1	1	Toggle	

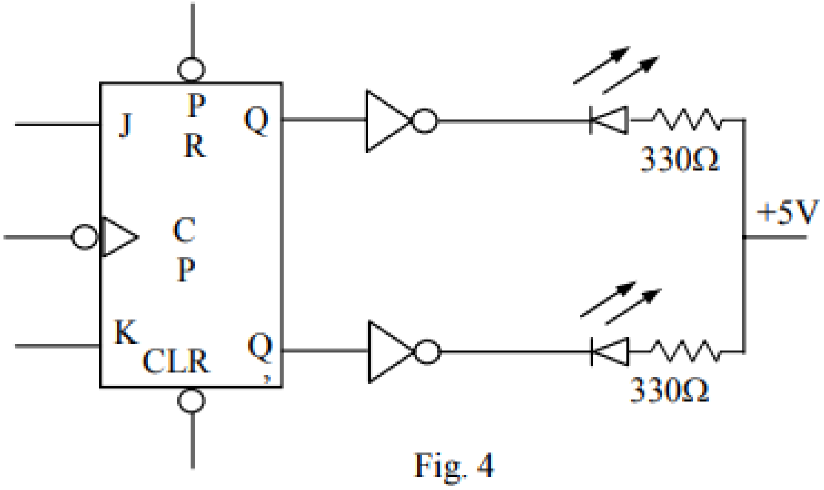


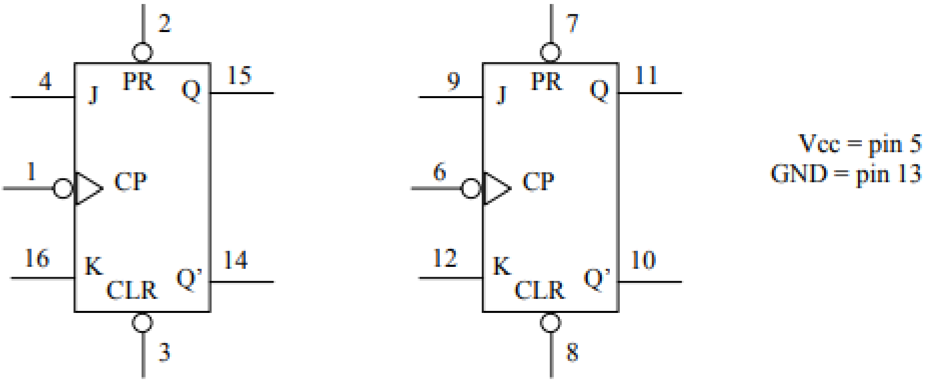
Fig. 4

In the Lab, Construct the circuit of Fig 4. Look at the data sheet for the 7476 and determine the inactive logic required at the PRE and CLR inputs.

Connect the 7476 for the SET mode by connecting J = 1, K = 0. With CLOCK (CP) = O; test the effect of PRE, CLR by putting a 0 on each, one at a time.

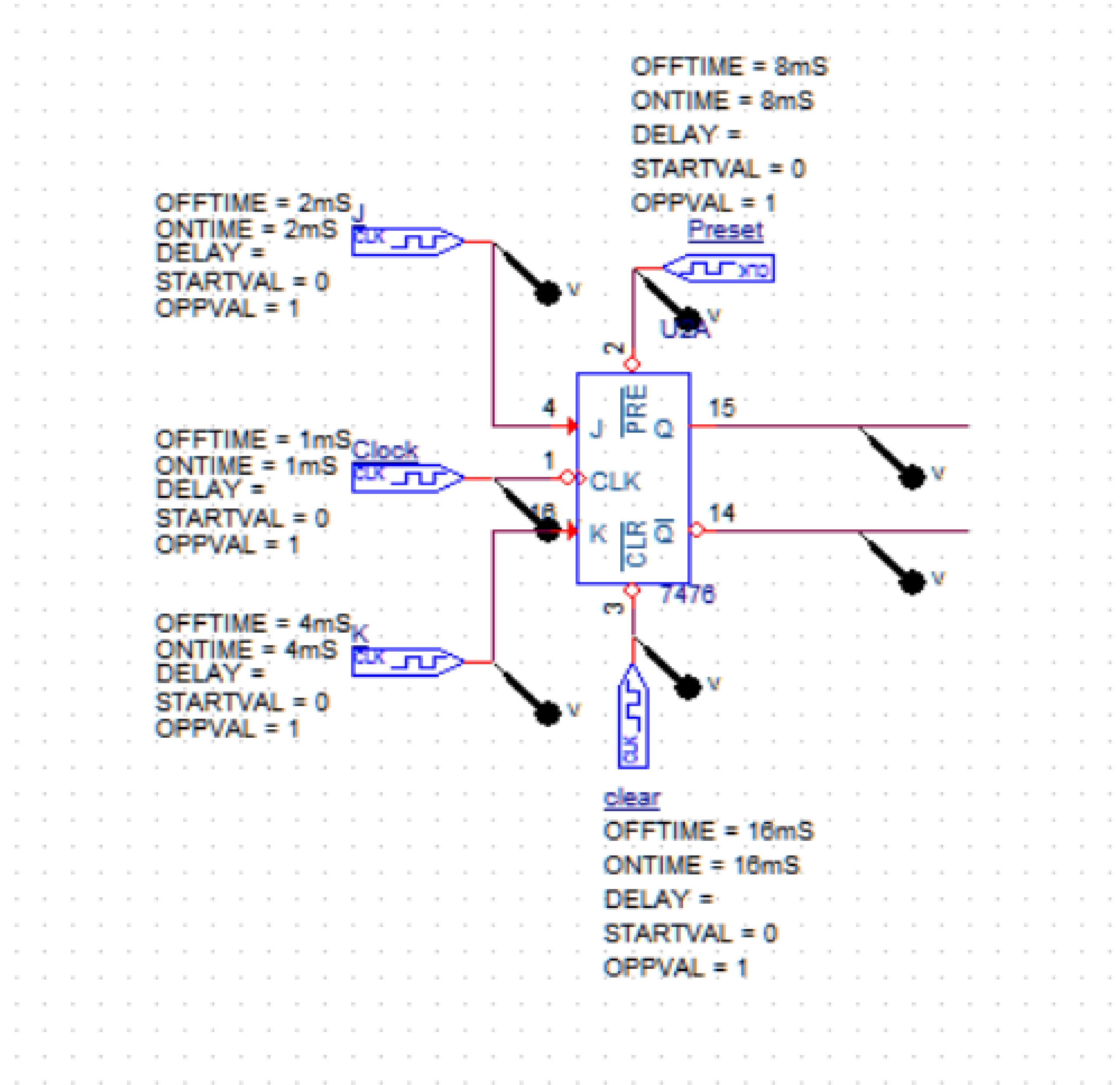
Put CLR = 0, then pulse the clock (CP) by putting a HIGH then a LOW, on the clock. Does the CLR input override J input?

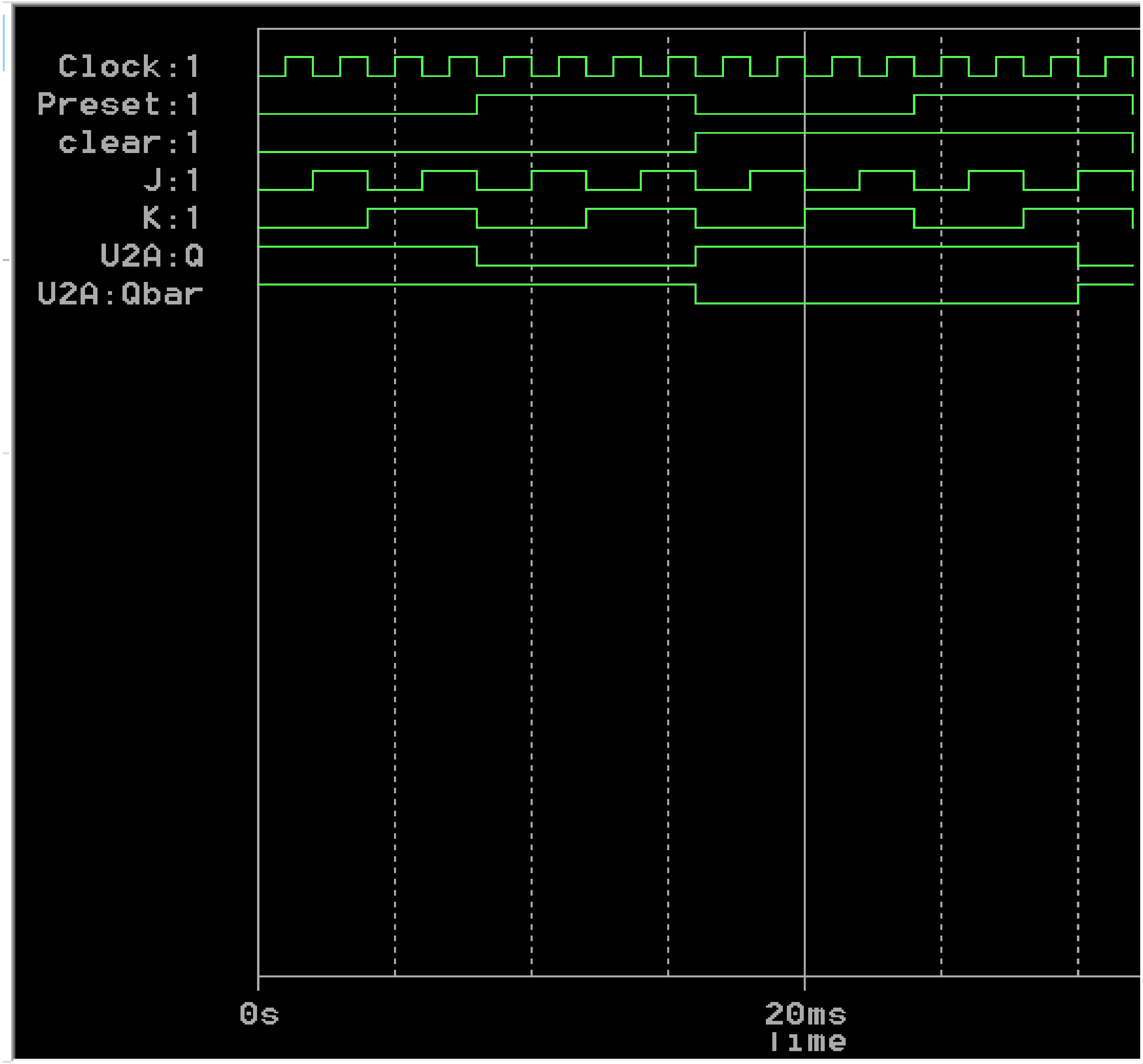
Verify the operation of the JK flip flop by experimentally obtaining the characteristics Table.



Following is the arrangement to observe and study the behavior of JK flip flop with preset and clear inputs.

In the transient simulation alongside, we can observe how ***Q*** *and* ***Q’*** *change with the change in clock, J, K, Preset and Clear.*




Clock:1
Preset:1
clear:1
J:1
K:1
U2A:Q
U2A:Qbar
0s
20ms
Time

Observations:

When Preset =0 and Clear =0, always Q=1, Q’=1

When Preset =0 and Clear =1, always Q=1, Q’=0

When Preset =1 and Clear =0, always Q=0, Q’=1

When Preset =1 and Clear =1, Q depends on J and K at negative edge of the clock.

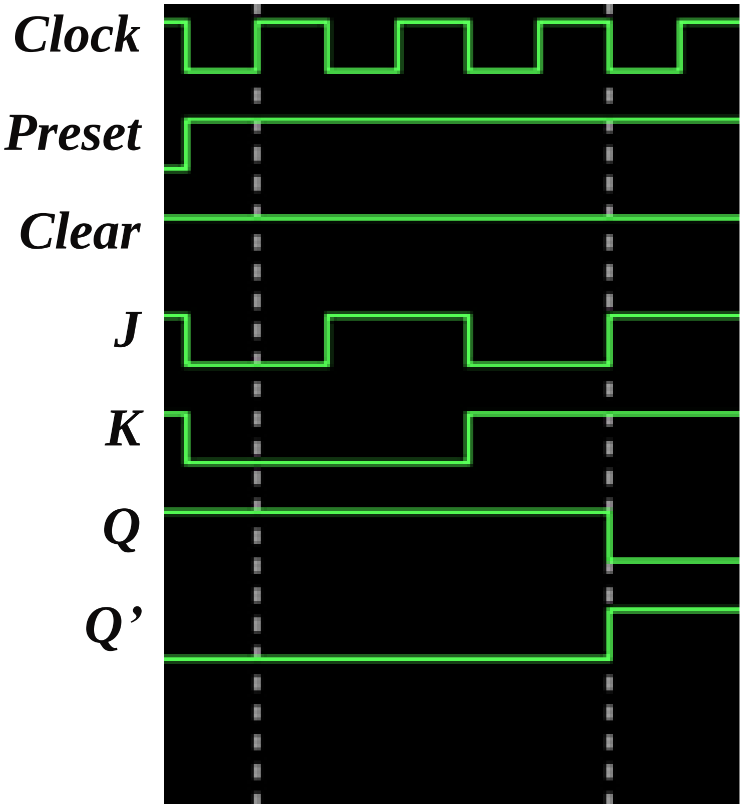
When J=0, K=0: Q and Q’ remains same

When J=0, K=1: Q=0 and Q’=1

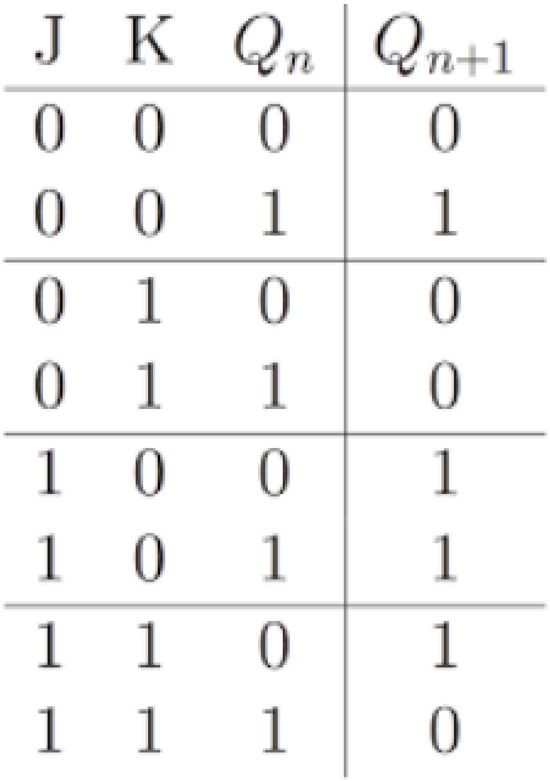
When J=1, K=0: Q=1 and Q’=0

When J=1, K=1: Q and Q’ toggles.

This is the small part of simulation when both Preset, Clear are 1.



Following characteristics table is obtained for JK flip flop:



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

Does the CLR input override the J input?

- *Yes, CLR and preset can override J, K and Clock inputs. J, K and Clock are significant only when preset and clear inputs are set to 1, 1*