Sequential Circuits
(Introduction to Latches and Flip-Flop)

Objectives of this experiment

- To verify the property of SR Latch using NOR.
- To verify the property of SR Latch using clock input.
- To verify the property of D Flip-Flop using NOR.
- To verify the property of D Flip-Flop using NAND.
- To verify the property of D Flip-Flop using 7474 IC.
- To verify the property of JK Flip-Flop using 7476 IC.
- To verify the property of JK Master/Slave Flip-Flop using 7476 IC.
- To verify the property of T Flip-Flop using gates.

Equipment Required

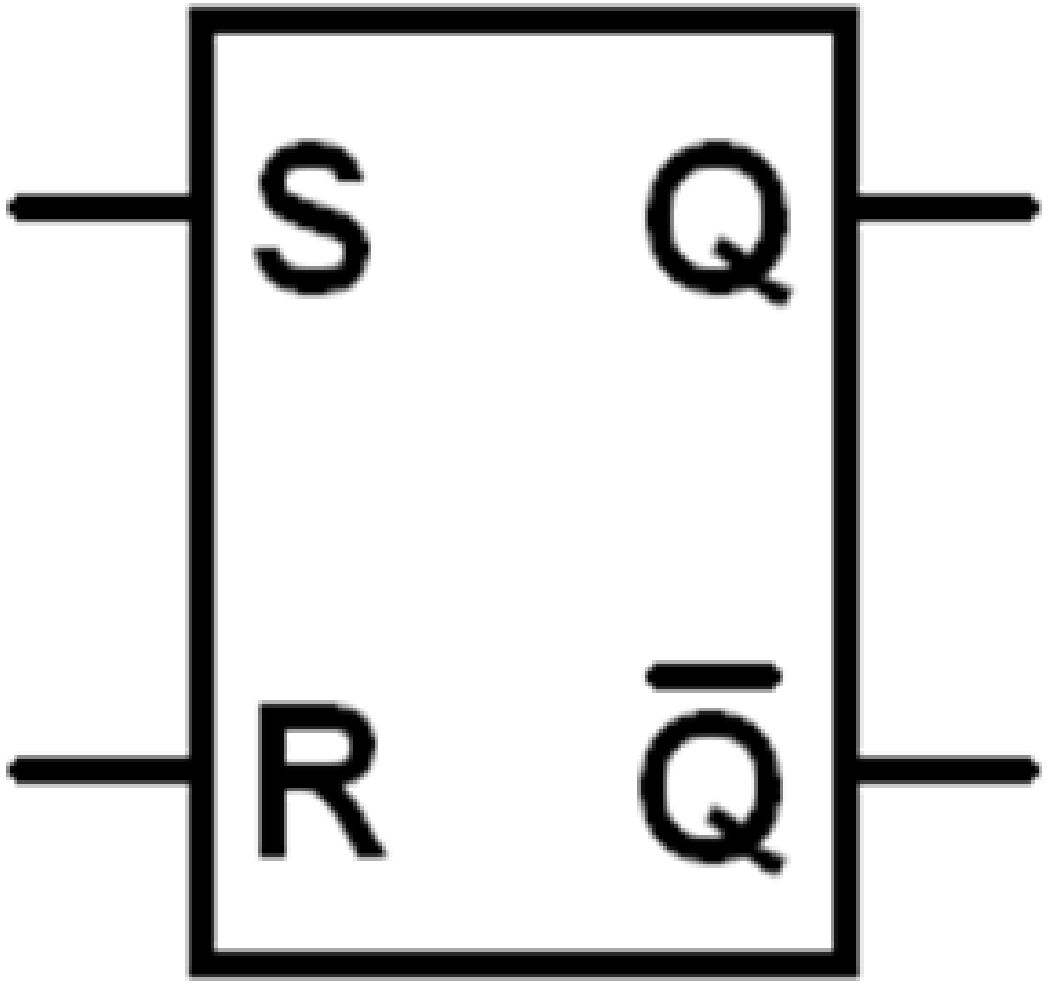
- •Personal computer with software for simulation of logic circuit installed,
- •Digital Logic Trainer kit with following:
- •Power supply unit,
- •SPDT switches,
- •LEDs,
- •Quad two input AND gates (7408),
- •Quad two input OR gate (7432),
- •Hex inverter (7404),
- •Quad two input NAND gate (7400),
- •Triple three input NAND gate (7410),
- •Quad two input NOR gate (7420),
- •Dual positive edge triggered D Flip-Flop (7474 IC),
- •Dual JK Flip-Flop (7476 IC),

Differences between combinational and sequential logic circuits:-

COMBINATIONAL CIRCUITS	SEQUENTIAL CIRCUITS
Output depends only on the present value of the inputs.	Output depends on both the present and previous state values of the inputs
These circuits will not have any memory as their outputs change with the change in the input value.	Sequential circuits have some sort of memory as their output changes according to the previous and present values.
There are no feedbacks involved.	In a sequential circuit the outputs are connected to it as a feedback path.
Used in basic Boolean operations.	Used in the designing of memory devices.
Implemented in: Half adder circuit, full adder circuit, multiplexers, demultiplexers, decoders and encoders.	Implemented in: RAM, Registers, counters and other state retaining machines.

SR Latch

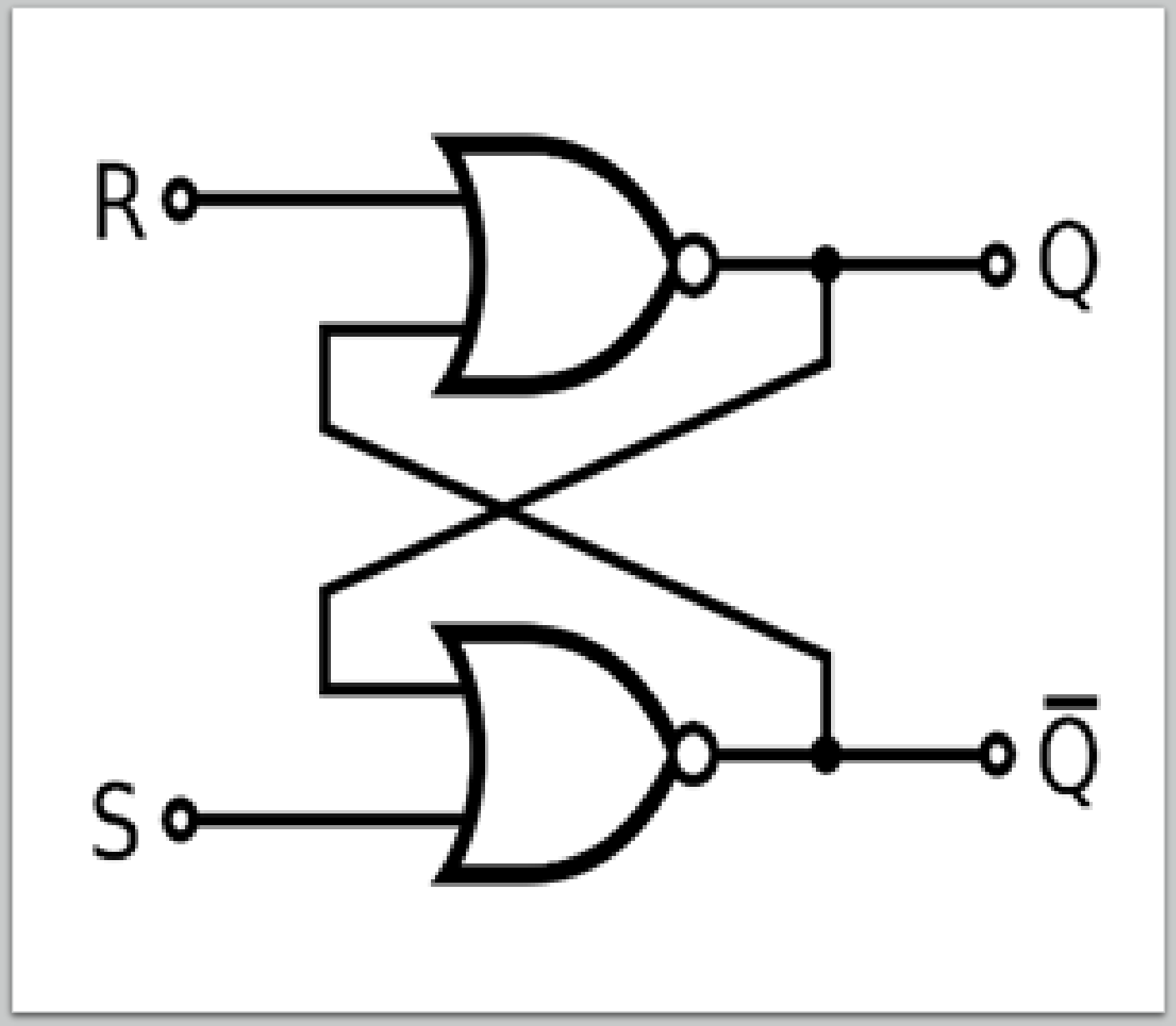
- S-R latch is the simplest bistable multivibrator circuit. A bistable multivibrator is a circuit having two stable states, as indicated by the prefix bi in its name. It is basically a one-bit memory bistable device that has two inputs, one which will “SET” the device (meaning the output = “1”), and is labelled S and another which will “RESET” the device (meaning the output = “0”), labelled R.



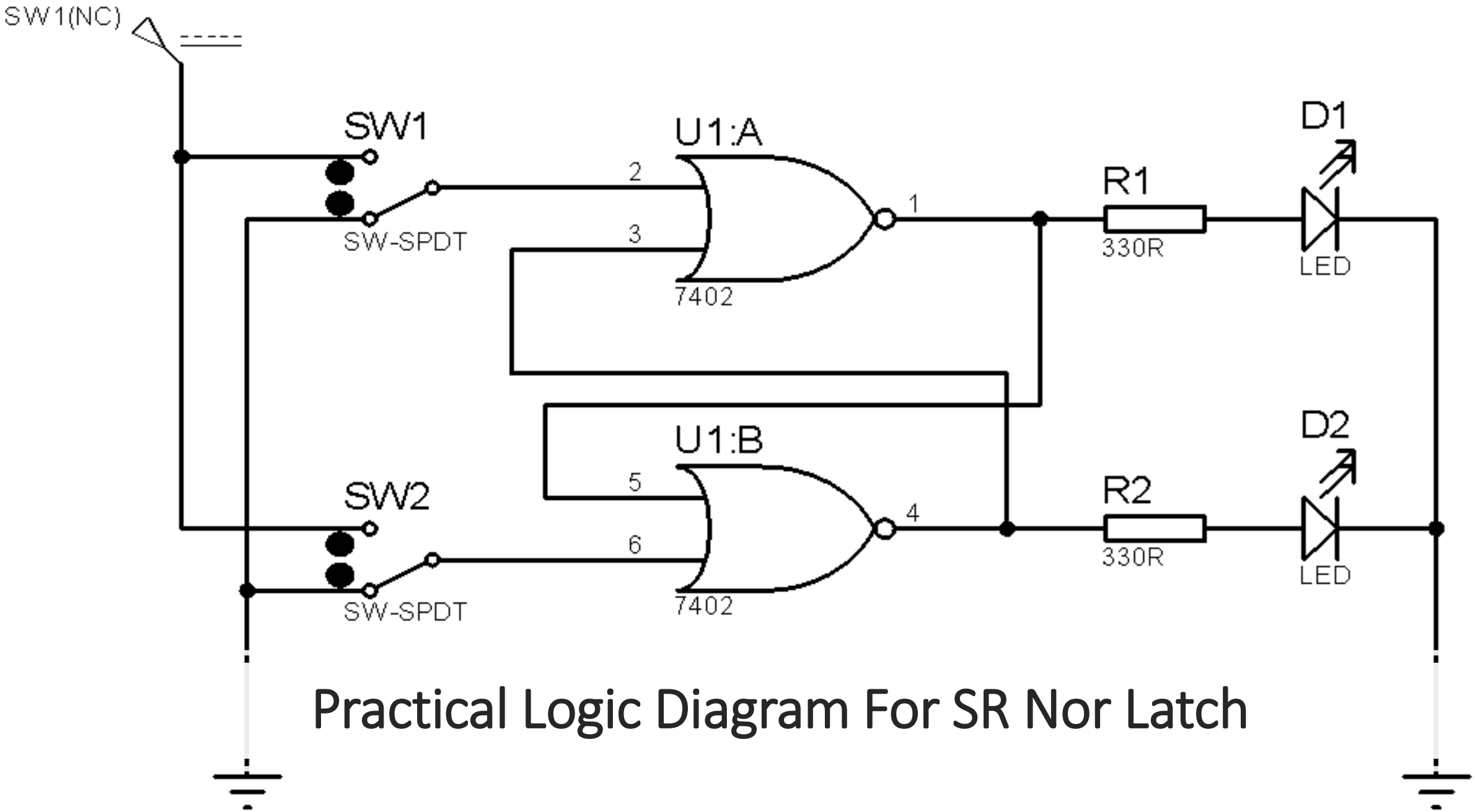
NOR SR Latch

It can be constructed from a pair of cross-coupled NOR logic gates. The stored bit is present on the output marked Q.

While the R and S inputs are both low, feedback maintains the Q and Q' outputs in a constant state, with Q' the complement of Q. If S (Set) is pulsed high while R (Reset) is held low, then the Q output is forced high, and stays high when S returns to low; similarly, if R is pulsed high while S is held low, then the Q output is forced low, and stays low when R returns to low.



S	R	Q	Q’
0	0	(Memory)	
0	1	0	1
1	0	1	0
1	1	(Undefined)	



Practical Logic Diagram For SR Nor Latch

Time Table For SR NOR

	Time Table				
	1	2	3	4	5
S	0	1	0	0	0
R	0	0	0	1	0
Q	0	1	1	0	0
Q’	1	0	0	1	1

Characteristic/Excitation table

S	R	Q	Q_N	Q'_N
0	0	O	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	X	X
1	1	1	X	X

- **SR NOR Latch MAP**

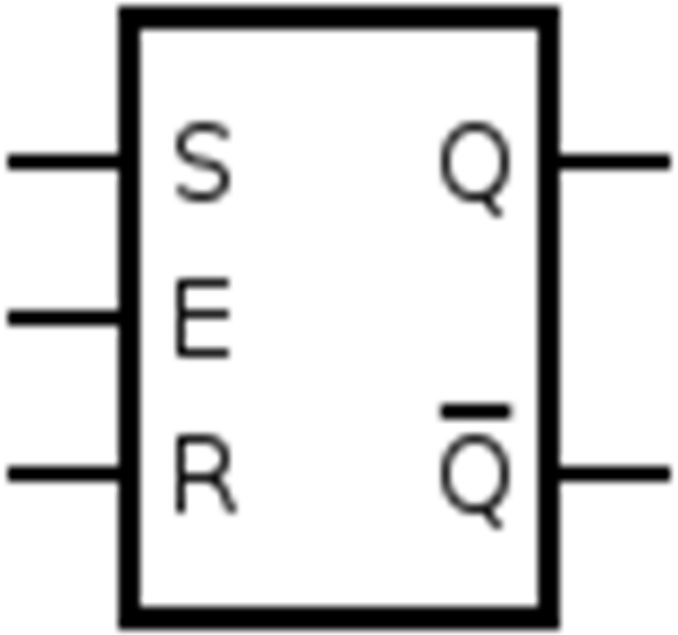
	S’R’	S’R	SR	SR’
Q’	0	0	X	1
Q	1	0	X	1

- Q_n =QR’+S

SR=0 (S and R could not be equal to 1 simultaneously

SR NOR latch characteristic equation

- **SR Latch with Clock Input**



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	(Unidentified)	

DIFFERENCES

Latches

- Latches are building blocks of flip flop.
- The word latch is mainly used for storage elements.
- A latch is level sensitive.
- When a latch is enabled it becomes transparent.

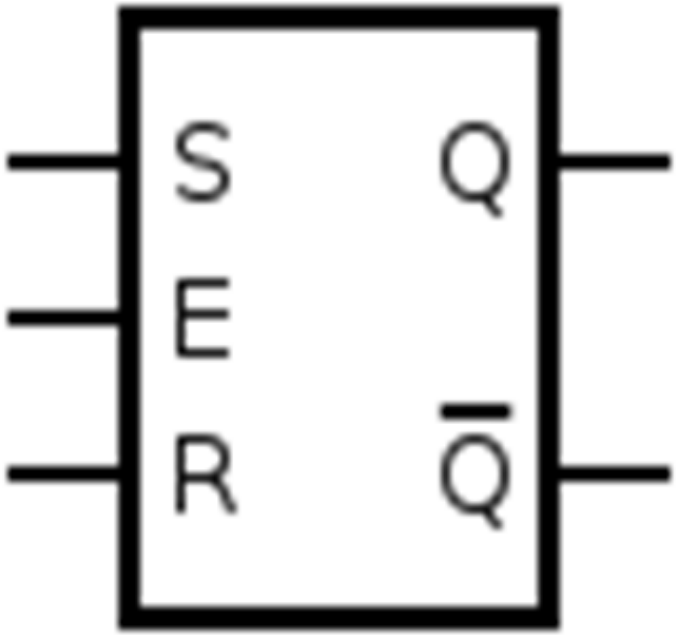
Flip Flops

- Flip flops are made on the base of latch.
- Clocked devices are termed as Flip Flop.
- Flip Flop is edge sensitive.
- Flip flops output only changes on a single type(positive going or negative going) of clock edge.

SR Latch with Clock Input

- The change in input signal cause immediate changes in output. Additional logic can be added to a simple transparent latch to make it non-transparent when another input (a "clock" input) is not asserted.
- A clocked SR latch can be made by adding a second level of NAND gates to the NAND SR latch.The extra NAND gates further invert the inputs so the simple NAND SR latch becomes a gated SR latch.
- With high clock, the signals can pass through the input gates to the encapsulated latch; all signal combinations except for (0,0) immediately reproduce the output (Q, Q’), i.e. the latch is transparent.
- With low clock, the latch is opaque and remains in the state it was left the last time clock was high.

- **SR Latch with Clock Input**



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	(Unidentified)	

- **Characteristics Table**

Q_n	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

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

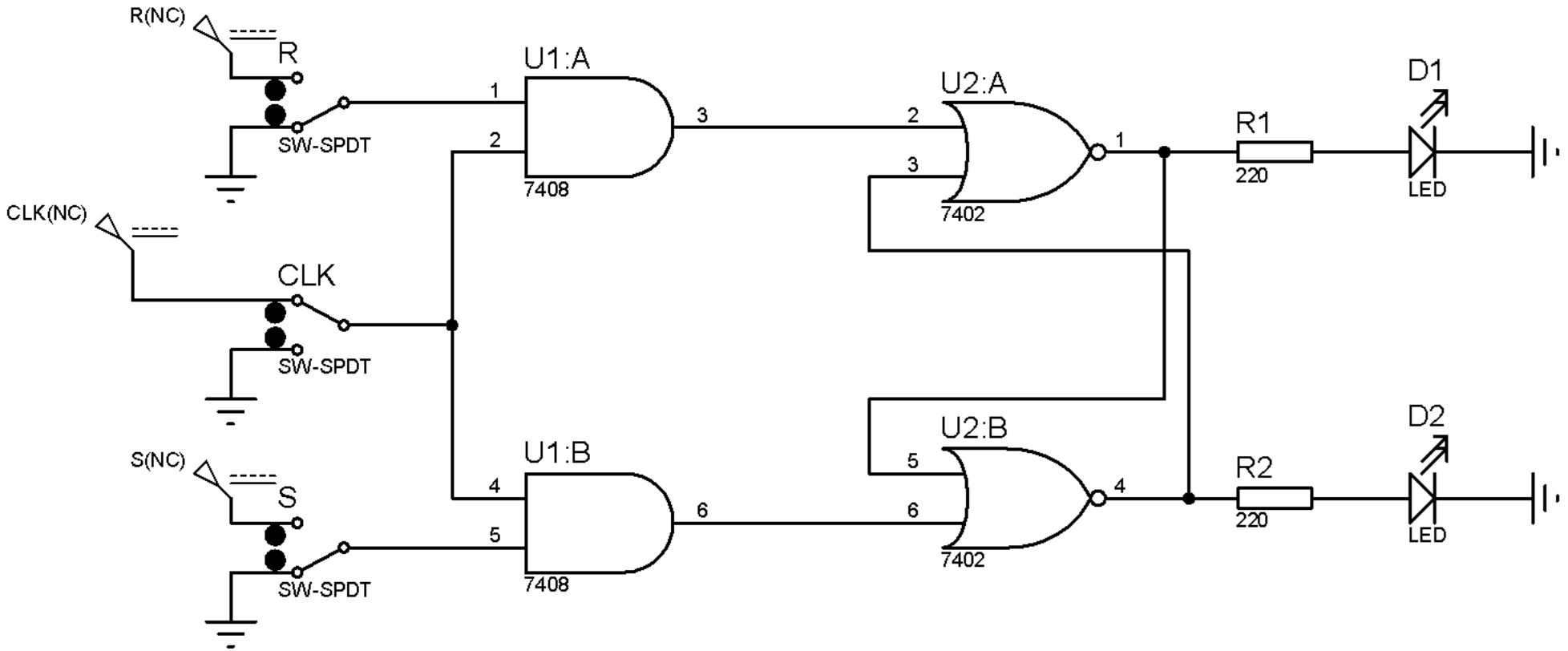
	S'R'	S'R	SR	SR'
Q_n'	0	0	X	1
Q_n	1	0	X	1

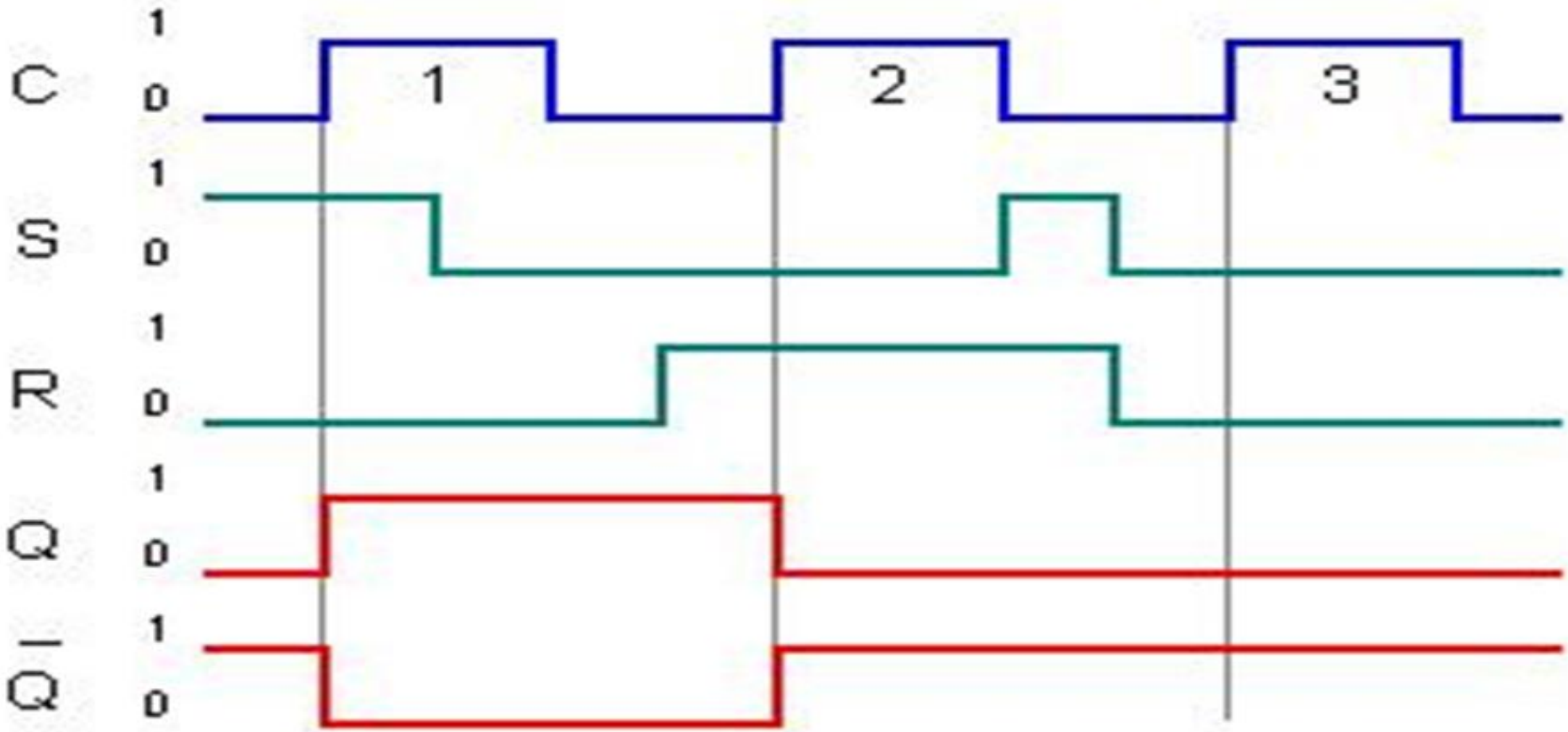
Excitation Table

- This table helps to generate the inputs i.e. the values of S and R by taking previous state and next state output. This table can be prepared from the values generated from characteristics table.

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

- Logical Diagram of Clocked RS Flip Flop




C
S
R
Q
Q̄
1
0
1
2
3

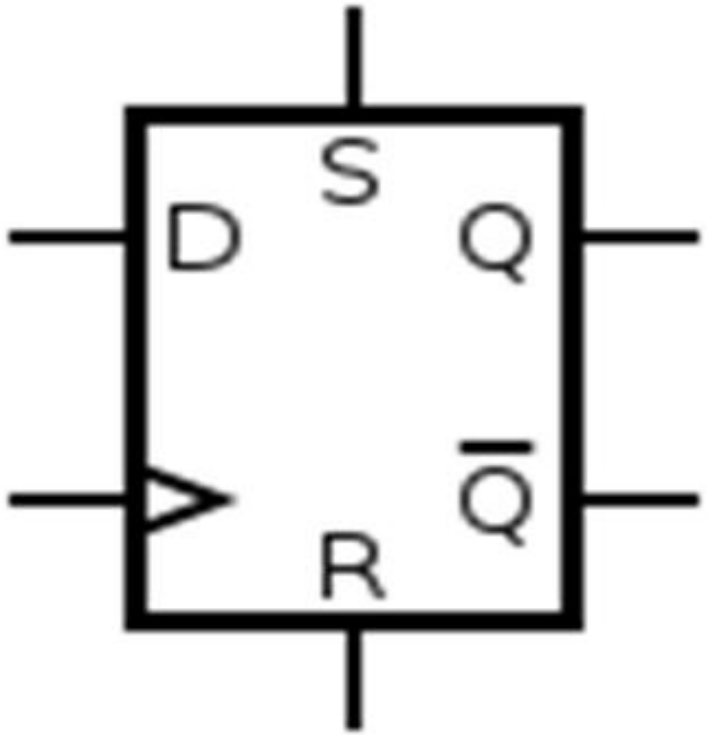
D-flip flop

- The D flip-flop is widely used flip-flop. It is also known as a "data" or "delay" flip-flop. The D flip-flop captures the value of the D-input at a definite portion of the clock cycle (such as the rising edge of the clock). That captured value becomes the Q output. At other times, the output Q does not change. The D flip-flop can be viewed as a memory cell, a zero-order hold, or a delay line.
- We can construct d- flip flop by connecting the two inputs two inputs of SR flip flop with an inverter. This helps to avoid the invalid case of SR flip flop i.e. (0,0). So, it has got only one input signal and one clock.
- When the clock input is low, whatever the input maybe, the output will be the value it hold last time when the clock was high. And when the clock is high then the value entered through the input signal will be the output.

D-flip flop

- The D flip-flop is widely used flip-flop. It is also known as a "data" or "delay" flip-flop. The D flip-flop captures the value of the D-input at a definite portion of the clock cycle (such as the rising edge of the clock). That captured value becomes the Q output. At other times, the output Q does not change. The D flip-flop can be viewed as a memory cell, a zero-order hold, or a delay line.
- We can construct d- flip flop by connecting the two inputs two inputs of SR flip flop with an inverter. This helps to avoid the invalid case of SR flip flop i.e. (0,0). So, it has got only one input signal and one clock.
- When the clock input is low, whatever the input maybe, the output will be the value it hold last time when the clock was high. And when the clock is high then the value entered through the input signal will be the output.

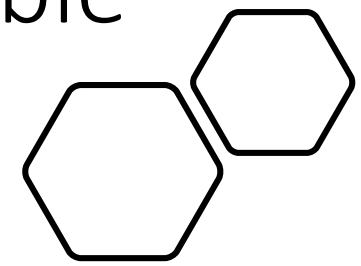
- D flip-flop



Clk	D	Q	Q’
0	X	(Memory)	
1	0	0	1
1	1	1	0

Characteristic Table

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



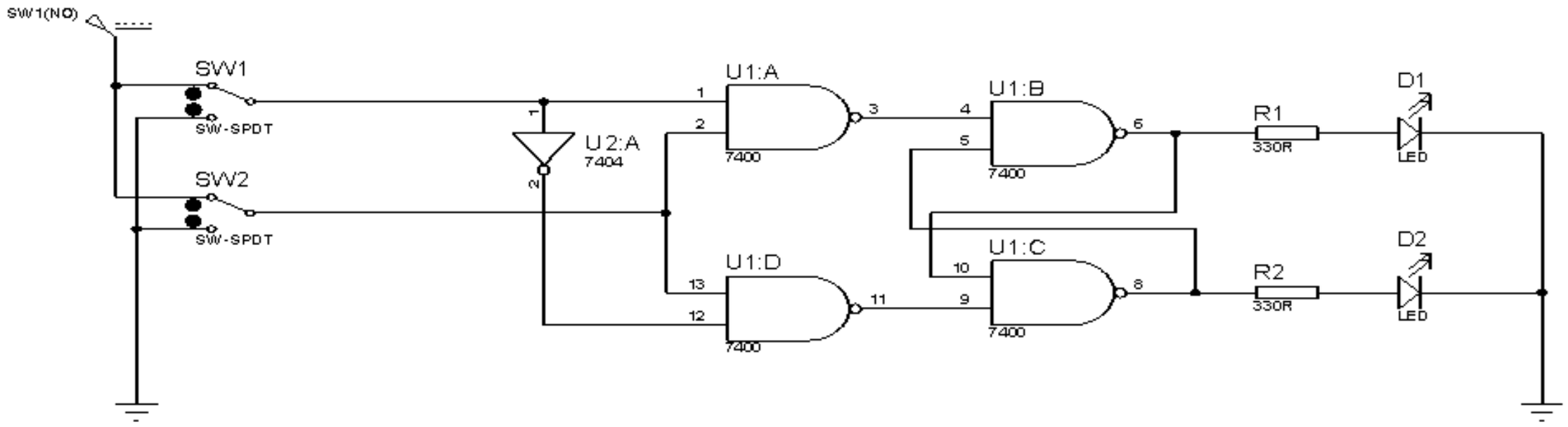
Excitation table

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

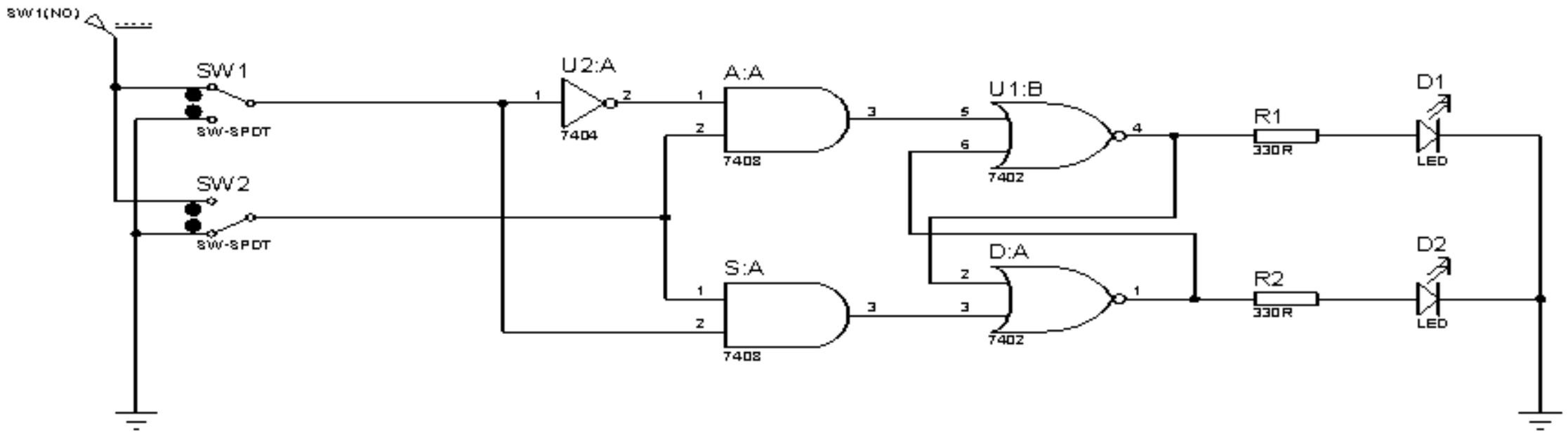
Kmap/Characteristic Equation

	D′	D
Q'_n	0	1
Q_n	0	1

$$Q_{n+1} = D$$



- **D flip Flop using Nand**



- **D flip Flop using Nor**