Flip flops

Flip flops (FF) are sequential logic circuits with 2 distinct stable states. They have control inputs that cause the outputs to switch from one stable state to the other. They are circuits with memory, because one can deduce the last applied command by analyzing the outputs. Because they are the basis of all sequential circuits, their applications are various: counters, registers, RAM memories, etc.

There are 4 types of flip flops:

- R-S
- J-K
- D
- T

1. R-S flip-flop

This FF has to command inputs named *S* (*Set*) and *R* (*Reset*) and 2 complementary outputs *Q* and $\overline{Q}$. The input *S* is used to write the information in the circuit (by convention the information means logic 1) and the input *R* that deletes the information from the circuit. Both inputs are active on logic 1.

The circuit's truth table is the following:

tn			tn+1	
R_n	S_n	Q_n	Q_{n+1}	
0	0	0	0	$Q_{n+1}=Q_n$
0	0	1	1	$Q_{n+1}=Q_n$
0	1	0	1	$Q_{n+1}=1$
0	1	1	1	$Q_{n+1}=1$
1	0	0	0	$Q_{n+1}=0$
1	0	1	0	$Q_{n+1}=0$
1	1	0	?	
1	1	1	?	

The circuit operates like this:

- if both inputs are inactive (R_n=S_n=0) the circuit keeps the state (Q_{n+1}=Q_n).
- if the input *S* is active (S_n=1, R_n=0) the information is written into the circuit (Q_{n+1}=1) no matter which is the previous state.
- if the input *R* is active (S_n=0, R_n=1) the information is deleted from the circuit (Q_{n+1}=0) no matter which is the previous state.

- the case R_n=S_n=1 has no sense, because is not logic to write and to delete the information in the same moment. The good working condition for this circuit is $R_n \cdot S_n$=0.

Above, t_n is the actual moment, and R_n, S_n, Q_n are the input and the output values at the present moment, and t_{n+1} is the next moment, Q_{n+1} is the output in this moment. For the circuit synthesis Q_{n+1} is considered output signal. The Karnaugh diagrams for the outputs are the following:

Q_n \ S_nR_n	00	01	11	10
0	0	0	X	1
1	1	0	X	1

$$Q_{n+1} = S_n + Q_n \cdot \overline{R_n}$$

Q_n \ S_nR_n	00	01	11	10
0	1	1	X	0
1	0	1	X	0

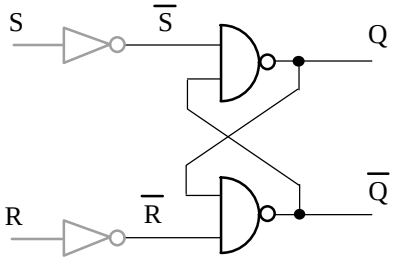
$$\overline{Q_{n+1}} = R_n + \overline{Q_n} \cdot \overline{S_n}$$

For the synthesis, the de Morgan theorem will be used in order to implement the circuit with NAND gates only:

$$Q_{n+1} = \overline{\overline{Q_{n+1}}} = \overline{\overline{(S_n + Q_n \cdot \overline{R_n})}} = \overline{\overline{S_n} \cdot \overline{(Q_n \cdot \overline{R_n})}}$$

$$\overline{Q}_{n+1} = \overline{\overline{Q}_{n+1}} = \overline{\overline{(R_n + \overline{Q_n} \cdot \overline{S_n})}} = \overline{\overline{R_n} \cdot \overline{(\overline{Q_n} \cdot \overline{S_n})}}$$

Thus, the circuit looks like in the next figure:



The circuit has the inputs active on logic 1. If the inverters are taken out, the circuit will have the inputs active on logic 0.

The above circuit can be implemented with NOR gates only. For this we will group the cells containing logic 0:

LAB no. 9.

Q_n \ S_nR_n	00	01	11	10
0	0	0	X	1
1	1	0	X	1

$$Q_{n+1} = (S_n + Q_n) \cdot \overline{R_n}$$

Q_n \ S_nR_n	00	01	11	10
0	1	1	X	0
1	0	1	X	0

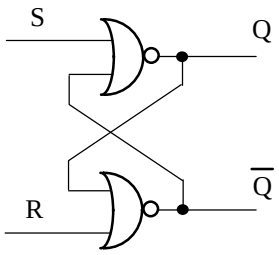
$$\overline{Q_{n+1}} = (R_n + \overline{Q_n}) \cdot \overline{S_n}$$

Applying the de Morgan theorem the equations become:

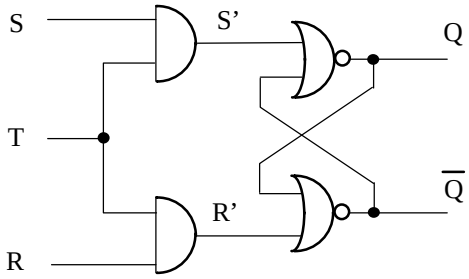
$$Q_{n+1} = \overline{\overline{Q}_{n+1}} = \overline{\overline{(S_n + Q_n) \cdot \overline{R_n}}} = \overline{\overline{(\overline{Q_n + S_n})} + R_n}$$

$$\overline{Q}_{n+1} = \overline{\overline{Q}_{n+1}} = \overline{\overline{(R_n + \overline{Q}_n) \cdot \overline{S_n}}} = \overline{(\overline{\overline{Q}_n + R_n}) + S_n}$$

In this case the circuit will look like in the following figure:



The above circuit is an asynchronous one. The synchronous circuit is also available. It has the inputs activated by a clock signal (following figure). As long as *T* is logic 0, the AND gates have the outputs in logic 0 no matter which are the inputs *R* and *S*. In these conditions its state cannot be changed. When T is logic 1 then *R'=R* and *S'=S* and the circuit works like in the asynchronous version.



2. J-K flip flop

This kind of flip flop eliminates the RS flip flop indefinite state. It has 2

inputs J and K and a clock input in the synchronous version.

a) Asynchronous J-K flip flop

The truth table of this circuit is the following:

t_n			t_{n+1}	
J_n	K_n	Q_n	Q_{n+1}	
0	0	0	0	$Q_{n+1}=Q_n$
0	0	1	1	$Q_{n+1}=Q_n$
0	1	0	0	$Q_{n+1}=0$
0	1	1	0	$Q_{n+1}=0$
1	0	0	1	$Q_{n+1}=1$
1	0	1	1	$Q_{n+1}=1$
1	1	0	1	$Q_{n+1}=Q_n$
1	1	1	0	$Q_{n+1}=\overline{Q_n}$

The input *J* has the same role as the input S for the R-S flip flop and the input K as R. The difference is for *J*=*K*=1 when instead of the indefinite state the output switches to the complementary one.

The VK associated diagrams are:

Q_n \ J_nK_n	00	01	11	10
0	0	0	1	1
1	1	0	0	1

$$Q_{n+1} = J_n \cdot \overline{Q_n} + Q_n \cdot \overline{K_n}$$

Q_n \ J_nK_n	00	01	11	10
0	1	1	0	0
1	0	1	1	0

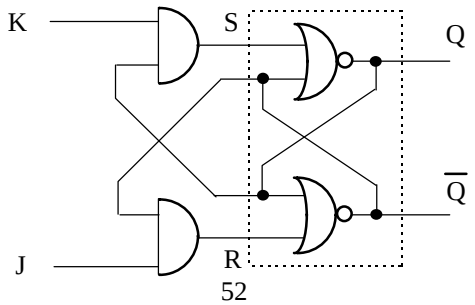
$$\overline{Q_{n+1}} = K_n \cdot Q_n + \overline{Q_n} \cdot \overline{J_n}$$

The above equations can be rewritten:

$$Q_{n+1} = \overline{\overline{Q}_{n+1}} = \overline{K_n \cdot Q_n + Q_n + J_n \cdot \overline{Q_n}}$$

$$\overline{Q}_{n+1} = \overline{Q_{n+1}} = \overline{K_n \cdot Q_n + \overline{Q}_n + J_n \cdot \overline{Q_n}}$$

The flip flop looks like:



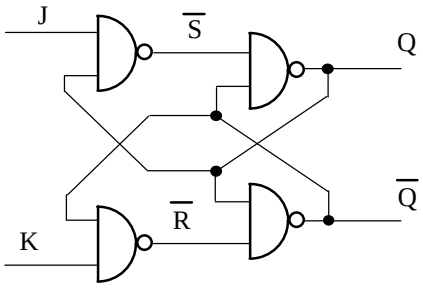
The block framed in the dotted square is a R-S flip flop. We can deduce immediately the connection equation between the two flip flops:

$$R_n = J_n \cdot Q_n$$
$$S_n = K_n \cdot \overline{Q_n}$$

Another implementation choice that comes from the above equations is the following:

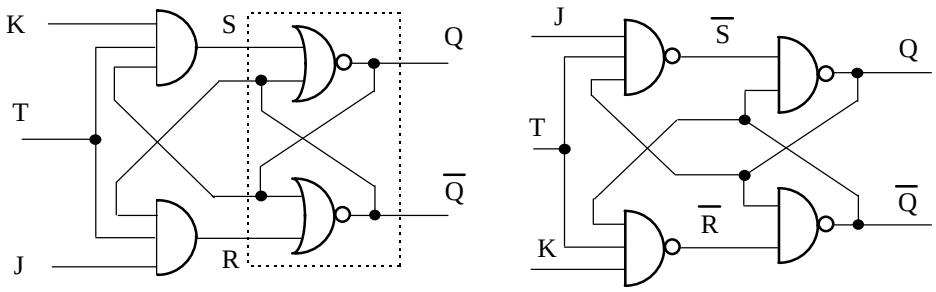
$$\overline{R_n} = \overline{K_n \cdot Q_n}$$
$$\overline{S_n} = \overline{J_n \cdot \overline{Q_n}} \ ,$$

and has the next schematic:



b) Synchronous J-K flip flop

The synchronous J-K flip flops are obtained by replacing the 2 inputs NAND gates cu 3 input gates, the clock being connected to the third:



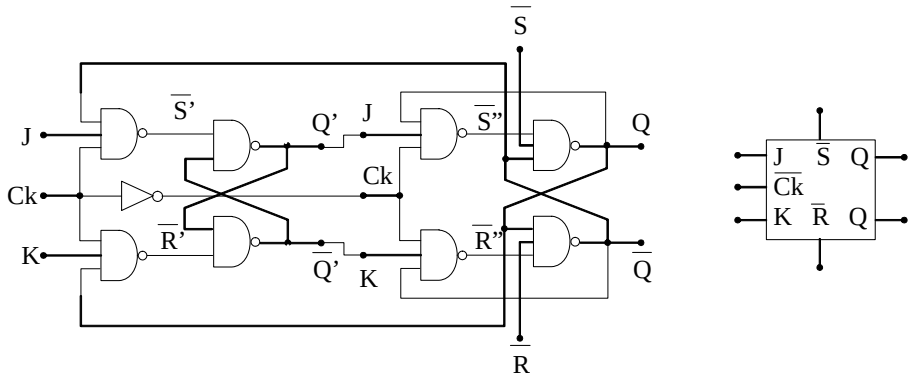
Their disadvantage is that for T=1 the circuit oscillates.

c) Master-slave J-K flip flop

In order to eliminate the oscillation a master slave structure has been proposed. This is based on two pipelined J-K flip flops. The first flip flop (the master) stores the data on the positive edge of the clock signal T, while the second (slave) is detached. When the clock T is logic 0, the data is

transferred from master to slave, while the master is detached from the inputs.

In the following figure it is shown the logic schematic of this circuit.



The truth table associated with the master –slave flip flop is:

J_n	K_n	$\overline{Ck}$	Q_{n+1}
0	0	↓	Q_n
0	1	↓	*0*
1	0	↓	*1*
1	1	↓	$\overline{Q_n}$

The symbol „↓” signify the fact that the output changes on the negative edge of the clock signal.

3. T flip flop

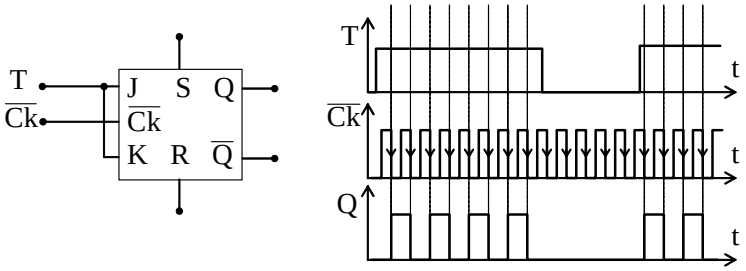
In many applications the JK flip flop is used with the inputs connected at logic 1. This connection represents another type of flip flop: *T flip flop* or *counting cell*.

Its truth table is:

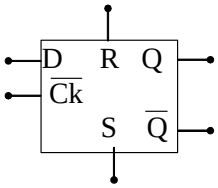
T	$\overline{Ck}$	Q_{n+1}
0	↓	Q_n
1	↓	$\overline{Q_n}$

LAB no. 9.

The flip flop realizes the clock frequency division with 2. As long as *T=1*, the output changes its state every two clock transitions (every negative edge). This property makes it suitable for using in counter's construction.



4. D flip flop



This circuit has an input called D (data) and a clock input (*Ck*). Besides these, it has two asynchronous inputs *R* and *S* that have the highest priority.

The value at the input at t_n goes at the output at t_{n+1}, as we can notice from its truth table.

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

From the truth table we observe that $Q_{n+1}=D_n$. Thus, the D flip flop introduces a delay of the input state. The output at the moment t_{n+1} is the

same as the input at the t_n (delay cell or memory cell). This circuit is useful for realizing static RAM memories, registers but also counters. Generally, any type of flip flop can be replaced with other type using conversion schemes.

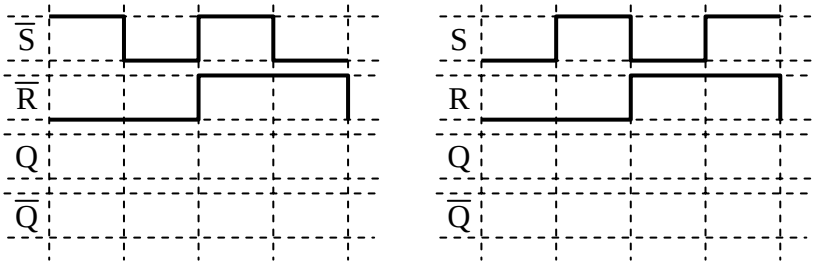
3. Lab work

Print the Lab sheet below and complete it according to the indications.

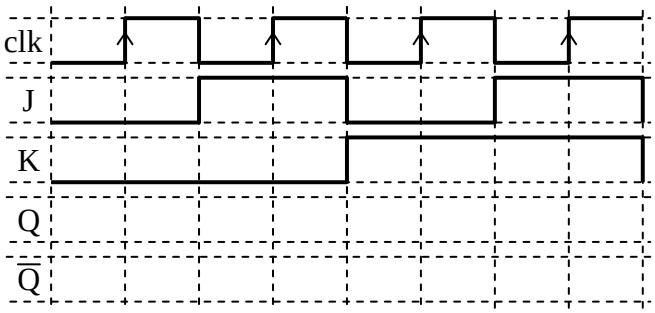
LAB no. 9.

LAB SHEET

1. Input the RS flip flop with NAND gates in MaxPlusII and simulate the circuit. Draw the waveforms from the simulation on the grid below. Write the delay time and the logic values on the waveforms. Compare the results with the truth table. Introduce the inverters on the inputs, save as a new circuit and simulate again. Draw the waveforms in this case.



2. Input in MaxPlusII a JK flip flop (JKFF from the library) and simulate it. Draw the waveforms on the grid below, measure the delay and compare the results with the truth table.



3. Input a D flip flop (DFF in the library) and simulate it. Draw the waveforms below, measure the delay and compare the results with the truth table.