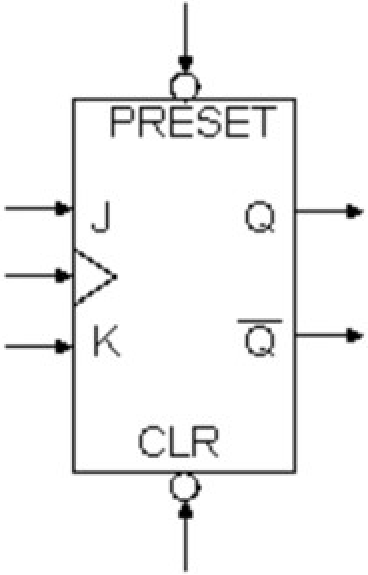
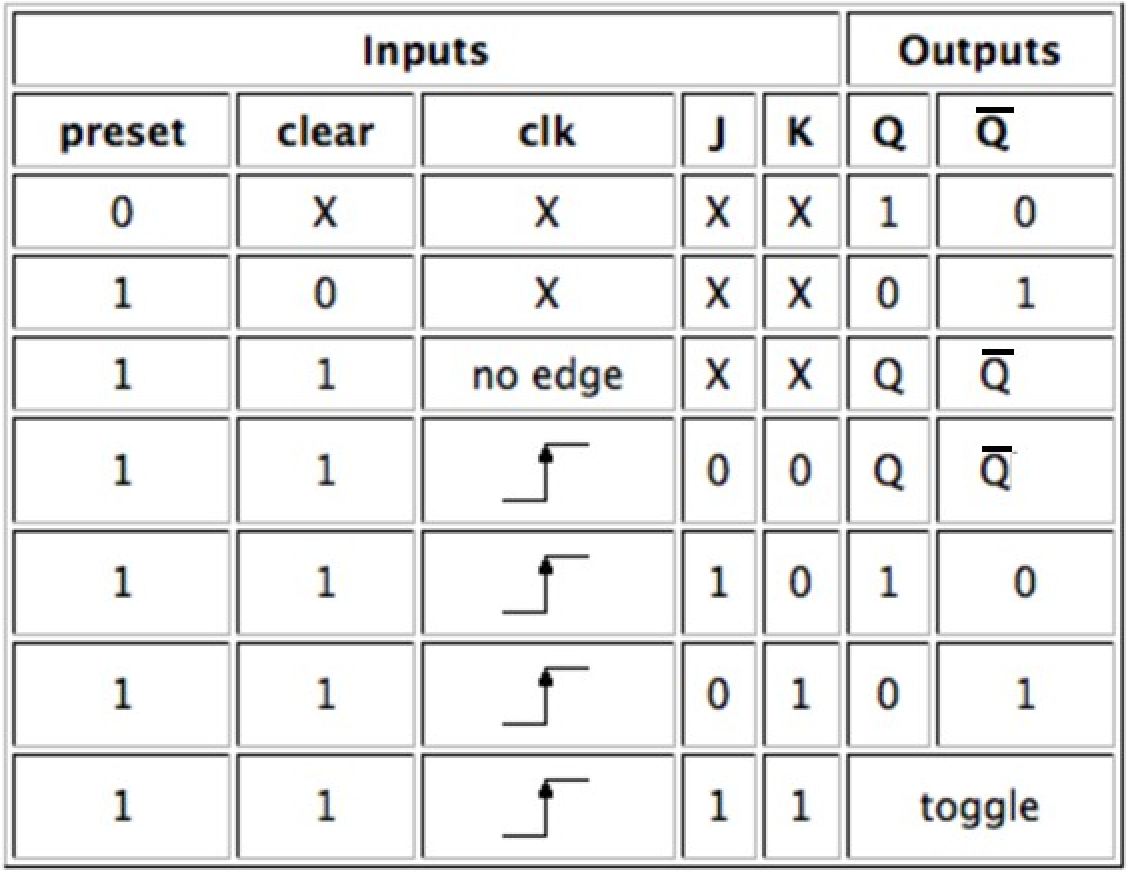
Vezba_4_JK_FF

Primer

Realizovati J-K flip-flop





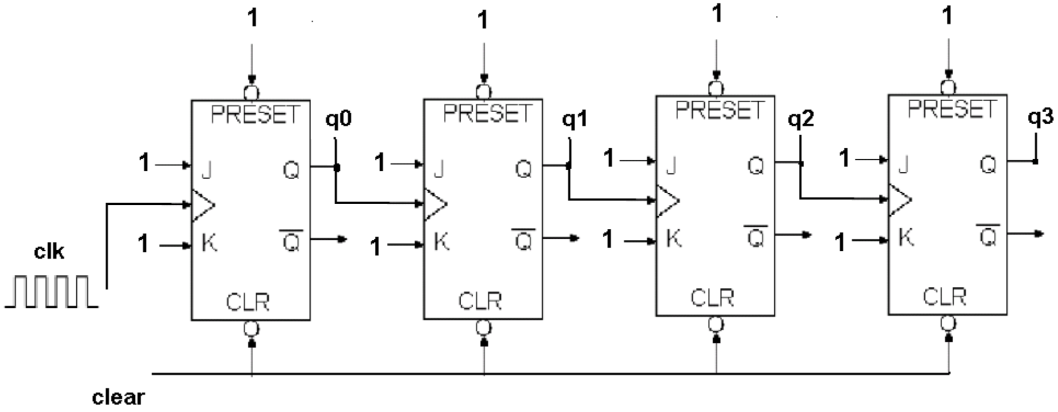
Inputs					Outputs	
preset	clear	clk	J	K	Q	$\overline{Q}$
0	X	X	X	X	1	0
1	0	X	X	X	0	1
1	1	no edge	X	X	Q	$\overline{Q}$
1	1	↑	0	0	Q	$\overline{Q}$
1	1	↑	1	0	1	0
1	1	↑	0	1	0	1
1	1	↑	1	1	toggle	

Primer

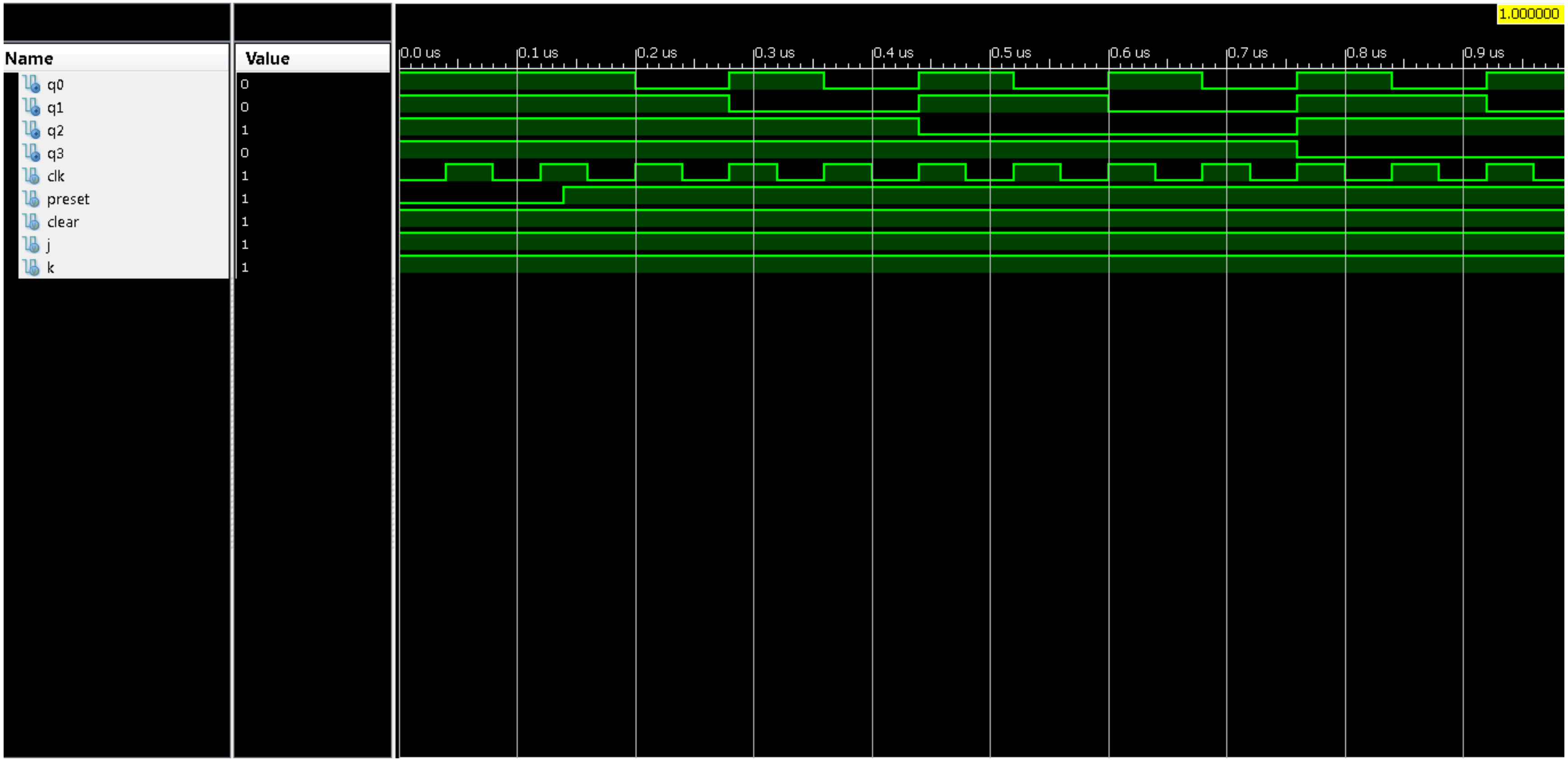
```
module jk_flip_lfop(input clk, preset,clear,j,k, output q);
 reg q=0;
 always @(posedge clk or negedge preset or negedge clear)
 begin
  if(preset==0)
                    q<=1'b1;
    else
    if (clear==0)
    q<=1'b0;
                     else
    case ({j,k})
     2'b00 : q <= q; //ovaj red moze i da se izbrise
     2'b01 : q <= 1'b0;
     2'b10 : q <= 1'b1;
     2'b11 : q <= ~q;
    endcase
  end
endmodule
```

Realizovati 4-bitni brojal uz pomoć J-K flip-flopova

```
module counter (input clk,preset,clear,j,k, output q0,q1,q2,q3);
  jk_flip_lfop qq0 (.clk(clk), .preset(preset), .clear(clear), .j(j),.k(k), .q(q0));
  jk_flip_lfop qq1 (.clk(q0), .preset(preset), .clear(clear), .j(j),.k(k),  .q(q1));
  jk_flip_lfop qq2 (.clk(q1), .preset(preset), .clear(clear), .j(j),.k(k),  .q(q2));
  jk_flip_lfop qq3 (.clk(q2), .preset(preset), .clear(clear), .j(j),.k(k),  .q(q3));
Endmodule
```



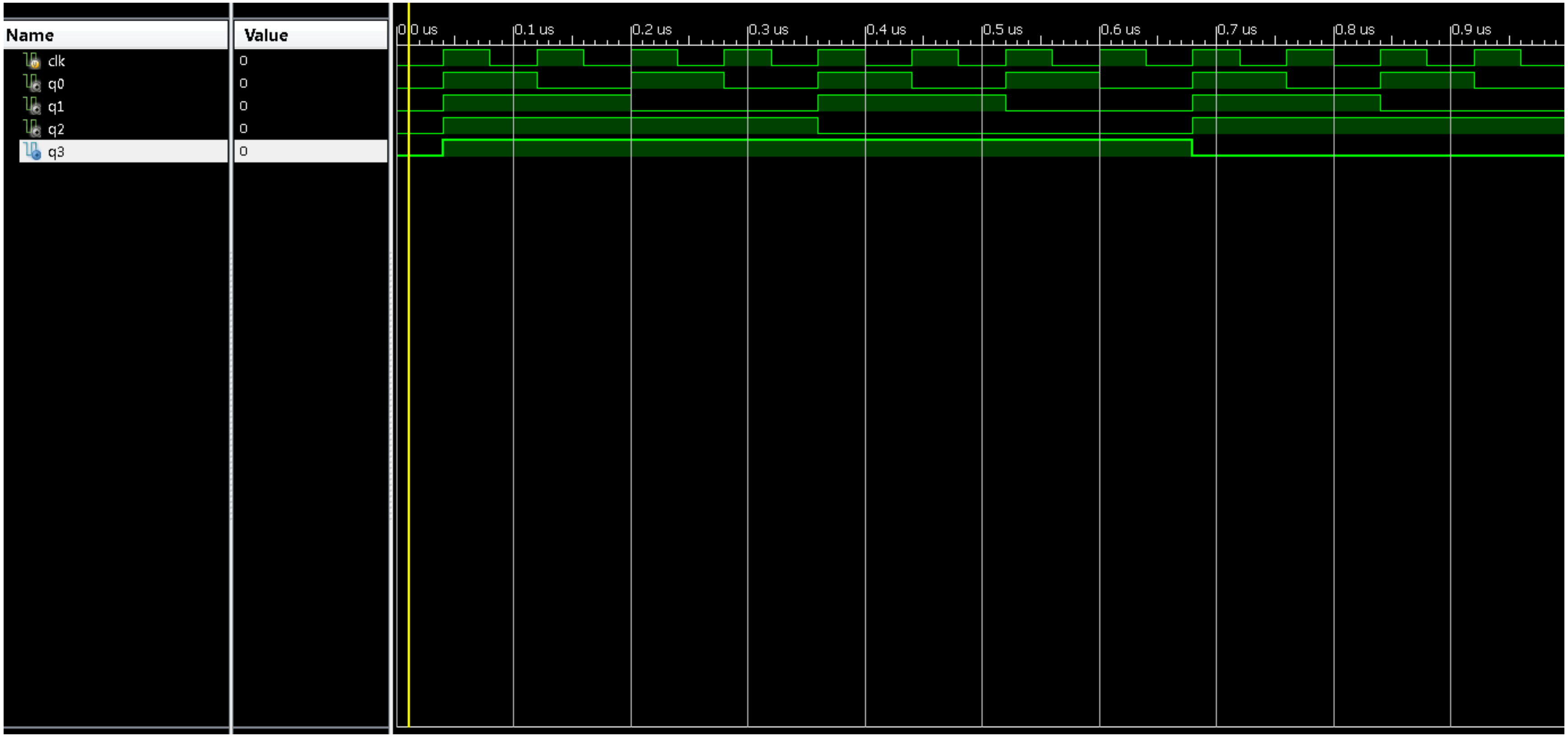
```
module sim_counter;
        // Inputs
        reg clk;
        reg preset;
        reg clear;
        reg j;
        reg k;
        // Outputs
        wire q0;
        wire q1;
  wire q2;
        wire q3;

        // Instantiate the Unit Under Test (UUT)

        counter uut (
        .clk(clk),
        .preset(preset),
        .clear(clear),
        .j(j),
        .k(k),
        .q0(q0),
        .q1(q1),
    .q2(q2),
    .q3(q3)
        );

        initial begin
                // Initialize Inputs
                clk = 0;
                preset=0;
                clear=1;
                j=1;
                k=1;
                // Wait 100 ns for global reset to finish
                #100;
                // Add stimulus here
        end
always @(*)
begin
#40;
clk<=~clk;
end
always @(posedge clk)
begin
#100;
preset<=1;
end
endmodule
```



1.000000
Name
Value
q0 0
q1 0
q2 1
q3 0
clk 1
preset 1
clear 1
j 1
k 1
0.0 us
0.1 us
0.2 us
0.3 us
0.4 us
0.5 us
0.6 us
0.7 us
0.8 us
0.9 us

```
module counter (input clk, output q0,q1,q2,q3);
  jk_flip_lfop qq0 (.clk(clk),.q(q0));
  jk_flip_lfop qq1 (.clk(q0),.q(q1));
  jk_flip_lfop qq2 (.clk(q1),.q(q2));
  jk_flip_lfop qq3 (.clk(q2),.q(q3));
Endmodule
```

```
module jk_flip_lfop(input clk,output q);
 reg q=0;
 always @(posedge clk)
 begin
q<=~q;
/*
  if(preset==0)
    q<=1'b1;
    else
    if (clear==0)
    q<=1'b0;
                         else
    case ({j,k})
     2'b00 : q <= q; //ovaj red moze i da se izbrise
     2'b01 : q <= 1'b0;
     2'b10 : q <= 1'b1;
     2'b11 : q <= ~q;
    endcase       */
  end
endmodule
```



Name
Value
clk
q0
q1
q2
q3
0
0
0
0
0
0.0 us
0.1 us
0.2 us
0.3 us
0.4 us
0.5 us
0.6 us
0.7 us
0.8 us
0.9 us