xbarto0c / **Digital-electronics-1**

<> **Code** | Issues | Pull requests | Actions | Projects | Security | Insight

main ▾ **Digital-electronics-1** / Labs / **07-ffs** /

xbarto0c 7.cv ... 13 seconds ago **History**

..

images	4 minutes ago
project_1	4 minutes ago
readme.md	13 seconds ago

readme.md

Part 1:

Characteristic equations of the flip-flops:

$$q^D_{n+1} = d$$

$$q^{JK}_{n+1} = j\overline{q_n} + \overline{k}q_n$$

$$q^T_{n+1} = t\overline{q_n} + \overline{t}q_n$$

Flip-flops, completed tables:

D	Qn	Q(n+1)	Comments
0	0	0	

D	Qn	Q(n+1)	Comments
0	1	0	
1	0	1	
1	1	1	

J	K	Qn	Q(n+1)	Comments
0	0	0	0	No change
0	0	1	1	No change
0	1	0	0	Reset
0	1	1	0	Reset
1	0	0	1	Set
1	0	1	1	Set
1	1	0	1	Toggle
1	1	1	0	Toggle

T	Qn	Q(n+1)	Comments
0	0	0	No change
0	1	1	No change
1	0	1	Toggle(invert)
1	1	0	Toggle(invert)

Part 2, D latch:

p_d_latch process, VHDL code:

```
p_d_latch : process (d, arst, en)
   begin
       if (arst = '1') then -- arst znamená aktivní v jedničce, arstn by bylo v
           q <= '0';
           q_bar <= '1';
       elsif (en = '1') then
```

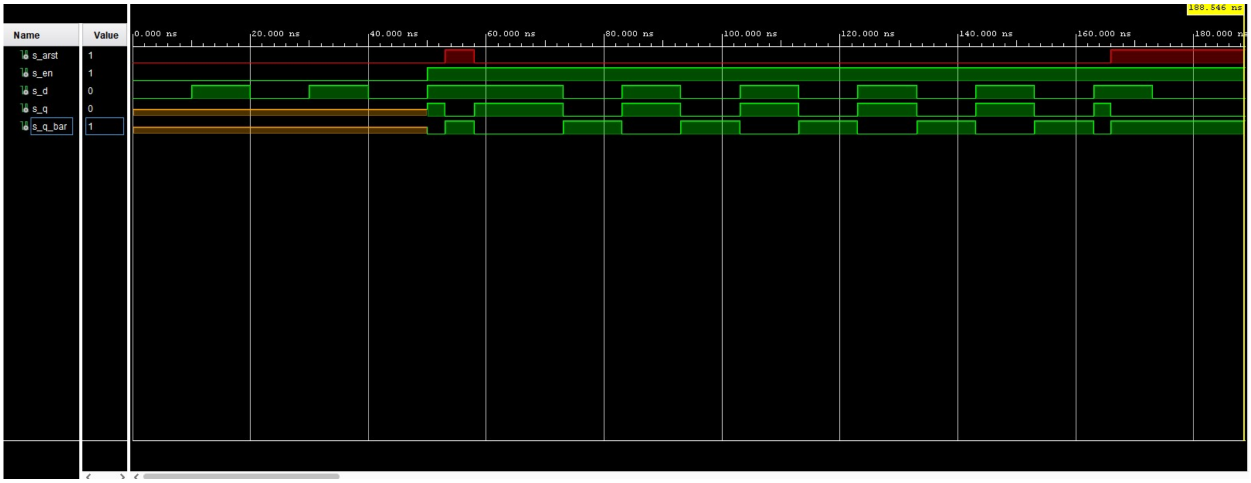
```
     q <= d;
     q_bar <= not d;
   end if;
 end process p_d_latch;
```

VHDL reset and stimulus processes from the testbench tb_d_latch:

```
p_reset : process --mohlo by být i v procesu latche, ne s hodinovým signálem
     begin
       s_arst <= '0';
       wait for 53ns;
       s_arst <= '1';
       wait for 5ns;
       s_arst <= '0';

       wait for 108ns;
       s_arst <= '1';
       wait;
   end process p_reset;
   p_d_latch : process
     begin
       report "Stimulus process started" severity note;
       s_en <= '0';
       s_d <= '0';

       --d sekvence
       wait for 10ns;
       s_d <= '1';
       wait for 10ns;
       s_d <= '0';
       wait for 10ns;
       s_d <= '1';
       wait for 10ns;
       s_d <= '0';
       wait for 10ns;
       s_d <= '1';

       s_en <= '1';

       wait for 3ns;
       assert(s_q = '0' and s_q_bar = '1')
       report "Error at s_d = '1' " severity error;
       --d sekvence
       wait for 10ns;
       s_d <= '1';
       wait for 10ns;
```

```
        s_d <= '0';
        wait for 10ns;
        assert(s_q = '1' and s_q_bar = '0')
        report "Error at s_d = '0'" severity error;
        s_d <= '1';
        wait for 10ns;
        s_d <= '0';
        wait for 10ns;
        s_d <= '1';
        wait for 10ns;
        s_d <= '0';
        --d sekvence
        wait for 10ns;
        s_d <= '1';
        wait for 10ns;
        s_d <= '0';
        wait for 10ns;
        s_d <= '1';
        wait for 10ns;
        s_d <= '0';
        wait for 10ns;
        s_d <= '1';
        wait for 10ns;
        s_d <= '0';


        report "Stimulus process finished" severity note;
        wait;
    end process p_d_latch;
```

Simulation screenshot:



Part 3, flip-flops:

p_d_ff_arst process:

```
p_d_ff_arst : process (clk, arst) -- arst resetuje hned, nečeká na hodiny
   begin
       if (arst = '1') then -- arst znamená aktivní v jedničce, arstn by bylo v
           q <= '0';
           q_bar <= '1';
       elsif rising_edge(clk) then
           q <= d;
           q_bar <= not d;
       end if;
   end process p_d_ff_arst;
```

p_d_ff_rst process:

```
p_d_ff_rst : process (clk) -- arst resetuje hned, nečeká na hodiny
   begin
       if rising_edge(clk) then -- arst znamená aktivní v jedničce, arstn by byl
           if (rst = '1') then
               q <= '0';
               q_bar <= '1';
           else
               q <= d;
               q_bar <= not d;
           end if;
       end if;
   end process p_d_ff_rst;
```

p_jk_ff_rst process:

```
p_jk_ff_rst : process (clk)
   begin
       if rising_edge(clk) then
           if(rst = '1') then   --synchronní reset
               s_q <= '0';
           else
               if(j = '0' and k = '0') then
                   s_q <= s_q; --toggle
               elsif(j = '0' and k = '1') then
                   s_q <= '0';
               elsif(j = '1' and k = '0') then
                   s_q <= '1';
               elsif(j = '1' and k = '1') then
```

```
                   s_q <= not s_q;
               end if;
           end if;
       end if;
   end process p_jk_ff_rst;

   q     <= s_q;
   q_bar <= not s_q;
```

p_t_ff_rst process:

```
p_t_ff_rst : process (clk) -- arst resetuje hned, nečeká na hodiny
   begin
       if rising_edge(clk) then -- arst znamená aktivní v jedničce, arstn by byl
           if (rst = '1') then
               s_q <= '0';
           else
               if (t = '1') then
                   s_q <= not s_q;
               else
                   s_q <= s_q;
               end if;
           end if;
       end if;
   end process p_t_ff_rst;

   q     <= s_q;
   q_bar <= not s_q;
```

d_ff_arst reset and stimulus processes:

```
               p_reset : process --mohlo by být i v procesu latche, ne s hodinov
           begin
               s_arst <= '0';
               wait for 58ns;
               s_arst <= '1';
               wait for 15ns;
               s_arst <= '0';

               wait for 108ns;
               s_arst <= '1';
               wait;
       end process p_reset;
       p_d_ff_arst : process
```

```
begin
    report "Stimulus process started" severity note;
    s_d <= '0';

    --d sekvence
    wait for 13ns;
    s_d <= '1';
    wait for 10ns;
    s_d <= '0';
    wait for 10ns;
    s_d <= '1';
    wait for 10ns;
    s_d <= '0';
    wait for 10ns;
    s_d <= '1';


    wait for 3ns;
    assert(s_q = '0' and s_q_bar = '1')
    report "Error at s_d = 1" severity error; -- zde nebo v časovaném
    --d sekvence
    wait for 10ns;
    s_d <= '1';
    assert(s_q = '1' and s_q_bar = '0')
    report "Error at s_d = 1" severity error;
    wait for 10ns;
    s_d <= '0';
    wait for 10ns;
    s_d <= '1';
    wait for 10ns;
    s_d <= '0';
    wait for 10ns;
    s_d <= '1';
    wait for 10ns;
    s_d <= '0';
    --d sekvence
    wait for 10ns;
    s_d <= '1';
    wait for 10ns;
    s_d <= '0';
    wait for 10ns;
    s_d <= '1';
    wait for 10ns;
    s_d <= '0';
    wait for 10ns;
    s_d <= '1';
    wait for 10ns;
    s_d <= '0';
```

```
        report "Stimulus process finished" severity note;
        wait;
    end process p_d_ff_arst;
```

d_ff_rst reset and stimulus processes:

```
        p_reset : process --mohlo by být i v procesu latche, ne s hodinov
                        begin
                                s_rst <= '0';
                                wait for 58ns;
                                s_rst <= '1';
                                wait for 30ns;
                                s_rst <= '0';

                                wait for 108ns;
                                s_rst <= '1';
                                wait;
end process p_reset;
p_d_ff_rst : process
    begin
        report "Stimulus process started" severity note;
        s_d <= '0';

        --d sekvence
        wait for 13ns;
        s_d <= '1';
        wait for 10ns;
        s_d <= '0';
        wait for 10ns;
        s_d <= '1';
        wait for 10ns;
        s_d <= '0';
        wait for 10ns;
        s_d <= '1';


        wait for 3ns;
        assert(s_q = '0' and s_q_bar = '1')
        report "Error at s_d = 1" severity error; -- zde nebo v časovaném
        --d sekvence
        wait for 10ns;
        s_d <= '1';
        wait for 10ns;
        assert(s_q = '1' and s_q_bar = '0')
```

```
            report "Error at s_d = 0" severity error;
            s_d <= '0';
            wait for 10ns;
            s_d <= '1';
            wait for 10ns;
            s_d <= '0';
            wait for 10ns;
            s_d <= '1';
            wait for 10ns;
            s_d <= '0';
            --d sekvence
            wait for 10ns;
            s_d <= '1';
            wait for 10ns;
            s_d <= '0';
            wait for 10ns;
            s_d <= '1';
            wait for 10ns;
            s_d <= '0';
            wait for 10ns;
            s_d <= '1';
            wait for 10ns;
            s_d <= '0';


            report "Stimulus process finished" severity note;
            wait;
        end process p_d_ff_rst;
```

jk_ff_rst reset and stimulus processes:

```
            p_reset : process --mohlo by být i v procesu latche, ne s hodinov
        begin
            s_rst <= '0';
            wait for 83ns;
            s_rst <= '1';
            wait for 15ns;
            s_rst <= '0';

            wait for 108ns;
            s_rst <= '1';
            wait;
    end process p_reset;
    p_jk_ff_rst : process
        begin
            report "Stimulus process started" severity note;
```

```
                      s_j <= '0';
                      s_k <= '0';

                      --jk sekvence
                      wait for 13ns;
                      s_j <= '0';
                      s_k <= '1';
                      wait for 10ns;
                      s_j <= '1';
                      s_k <= '0';
                      wait for 10ns;
                      s_j <= '1';
                      s_k <= '1';
                      wait for 10ns;
                      s_j <= '0';
                      s_k <= '0';
                      wait for 10ns;



                      wait for 3ns;
                      assert(s_q = '0' and s_q_bar = '1')
                      report "Error at s_j = 0 and s_k = 0" severity error; -- zde nebo

                      --jk sekvence
                      wait for 13ns;
                      s_j <= '0';
                      s_k <= '1';
                      assert(s_q = '1' and s_q_bar = '0')
                      report "Error at s_j = 0 and s_k = 1" severity error;
                      wait for 10ns;
                      s_j <= '1';
                      s_k <= '0';
                      wait for 10ns;
                      s_j <= '1';
                      s_k <= '1';
                      wait for 10ns;
                      s_j <= '0';
                      s_k <= '0';
                      wait for 10ns;



                      wait for 3ns;
                      --jk sekvence
                      wait for 13ns;
                      s_j <= '0';
                      s_k <= '1';
```

```
        wait for 10ns;
        s_j <= '1';
        s_k <= '0';
        wait for 10ns;
        s_j <= '1';
        s_k <= '1';
        wait for 10ns;
        s_j <= '0';
        s_k <= '0';
        wait for 10ns;

        wait for 3ns;


        report "Stimulus process finished" severity note;
        wait;
      end process p_jk_ff_rst;
```

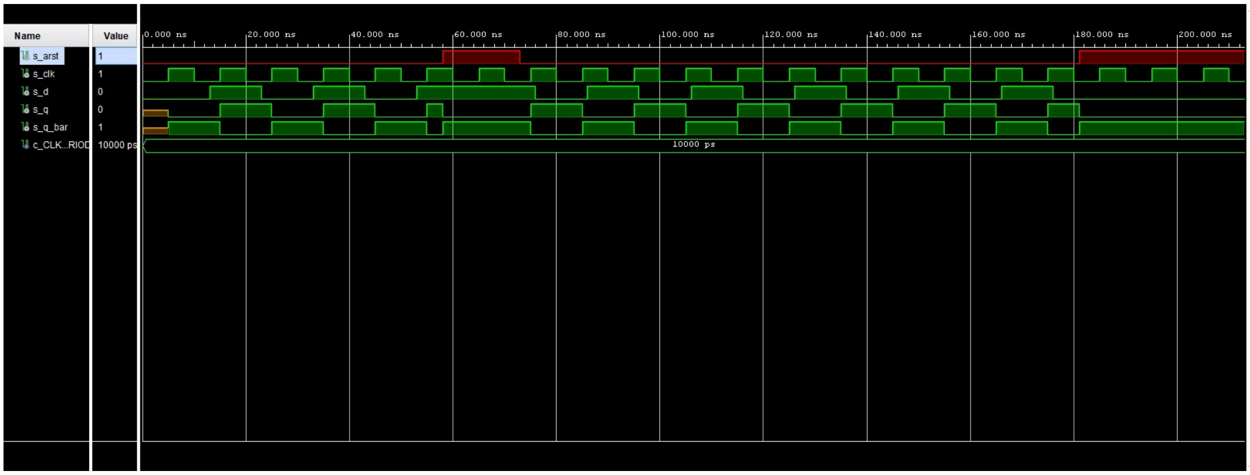
t_ff_rst reset and stimulus processes:

```
        p_reset : process --mohlo by být i v procesu latche, ne s hodinov
      begin
        s_rst <= '0';
        wait for 8ns;
        s_rst <= '1';
        wait for 36ns;
        s_rst <= '0';

        wait for 51ns;
        s_rst <= '1';
        wait for 37ns;

        s_rst <= '0';

        wait for 148ns;
        s_rst <= '1';
        wait;
    end process p_reset;
    p_t_ff_rst : process
      begin
        report "Stimulus process started" severity note;
        s_t <= '0';

        --t sekvence
        wait for 13ns;
        s_t <= '1';
```

```
                    wait for 10ns;
                    s_t <= '0';
                    wait for 10ns;
                    s_t <= '1';
                    wait for 10ns;
                    s_t <= '0';
                    wait for 10ns;
                    s_t <= '1';


                    wait for 3ns;
                    assert(s_q = '0' and s_q_bar = '1')
                    report "Error at s_t = 1" severity error; -- zde nebo v časovanéé
                    --t sekvence
                    wait for 10ns;
                    s_t <= '1';
                    assert(s_q = '1' and s_q_bar = '0')
                    report "Error at s_t = 0" severity error;
                    wait for 10ns;
                    s_t <= '0';
                    wait for 10ns;
                    s_t <= '1';
                    wait for 10ns;
                    s_t <= '0';
                    wait for 10ns;
                    s_t <= '1';
                    wait for 10ns;
                    s_t <= '0';
                    --d sekvence
                    wait for 10ns;
                    s_t <= '1';
                    wait for 10ns;
                    s_t <= '0';
                    wait for 10ns;
                    s_t <= '1';
                    wait for 10ns;
                    s_t <= '0';
                    wait for 10ns;
                    s_t <= '1';
                    wait for 10ns;
                    s_t <= '0';

                    wait for 10ns;
                    s_t <= '1';
                    wait for 10ns;
                    s_t <= '0';
                    wait for 10ns;
                    s_t <= '1';
```

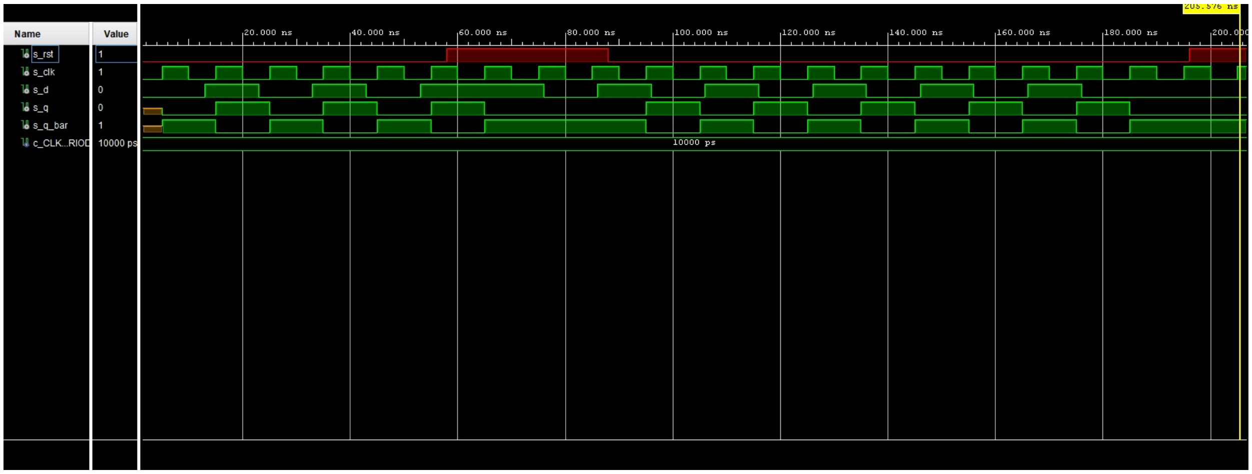
```
            wait for 10ns;
            s_t <= '0';
            wait for 10ns;
            s_t <= '1';
            wait for 10ns;
            s_t <= '0';



            report "Stimulus process finished" severity note;
            wait;
        end process p_t_ff_rst;
```

d_ff_arst simulation screenshot:



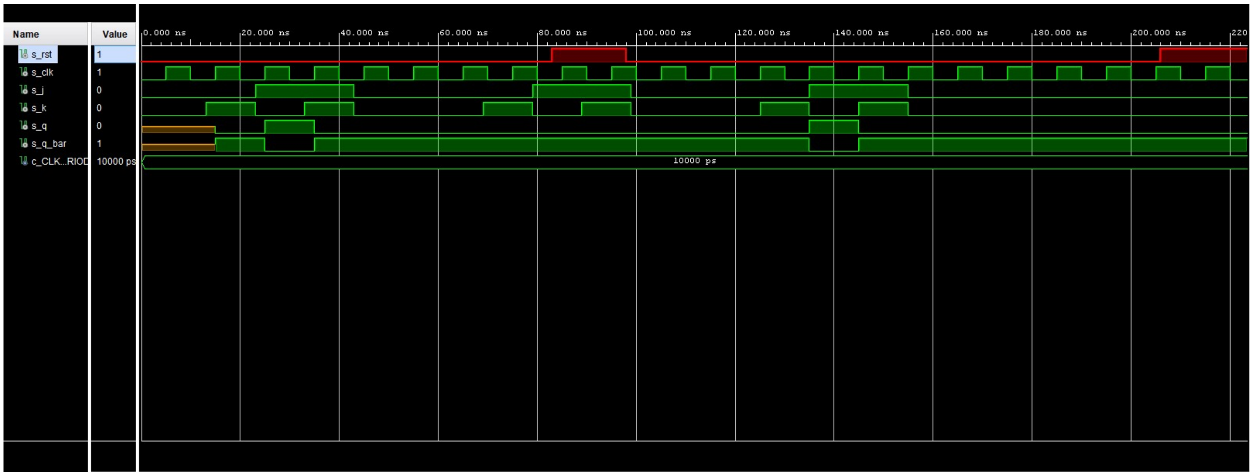
Name	Value
s_arst	1
s_clk	1
s_d	0
s_q	0
s_q_bar	1
c_CLK...RIOD	10000 ps

d_ff_rst simulation screenshot:



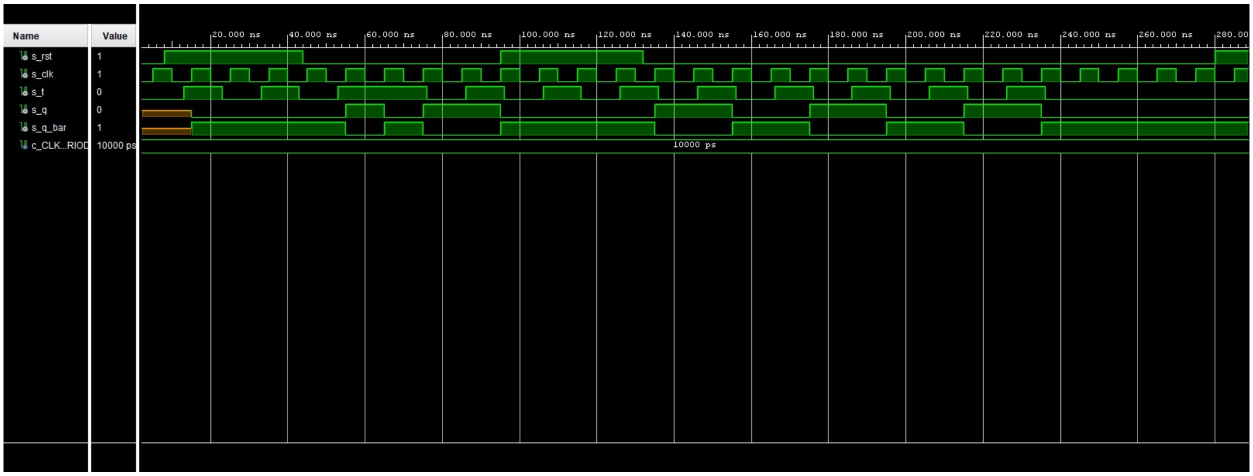
Name	Value
s_rst	1
s_clk	1
s_d	0
s_q	0
s_q_bar	1
c_CLK...RIOD	10000 ps

jk_ff_rst simulation screenshot:



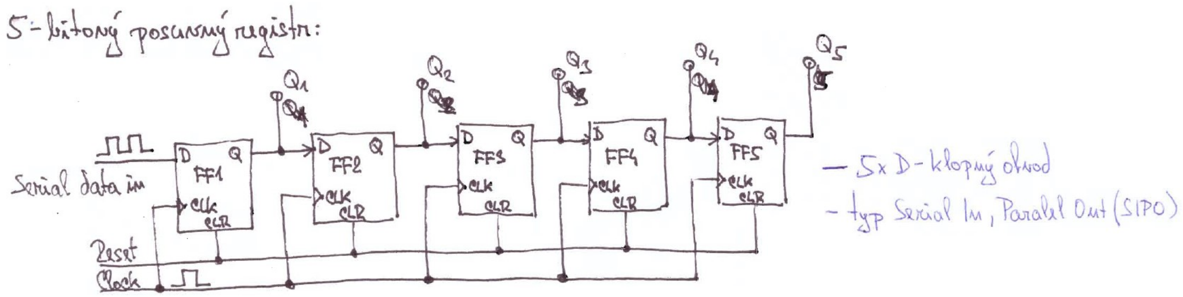
Name	Value
s_rst	1
s_clk	1
s_j	0
s_k	0
s_q	0
s_q_bar	1
c_CLK...RIOD	10000 ps

t_ff_rst simulation screenshot:



Name	Value
s_rst	1
s_clk	1
s_t	0
s_q	0
s_q_bar	1
c_CLK...RIOD	10000 ps

Part 3 - Shift register schematic:



5-bitový posuvný registr:

— 5x D-klopný obvod

- typ Serial In, Paralel Out (SIPO)