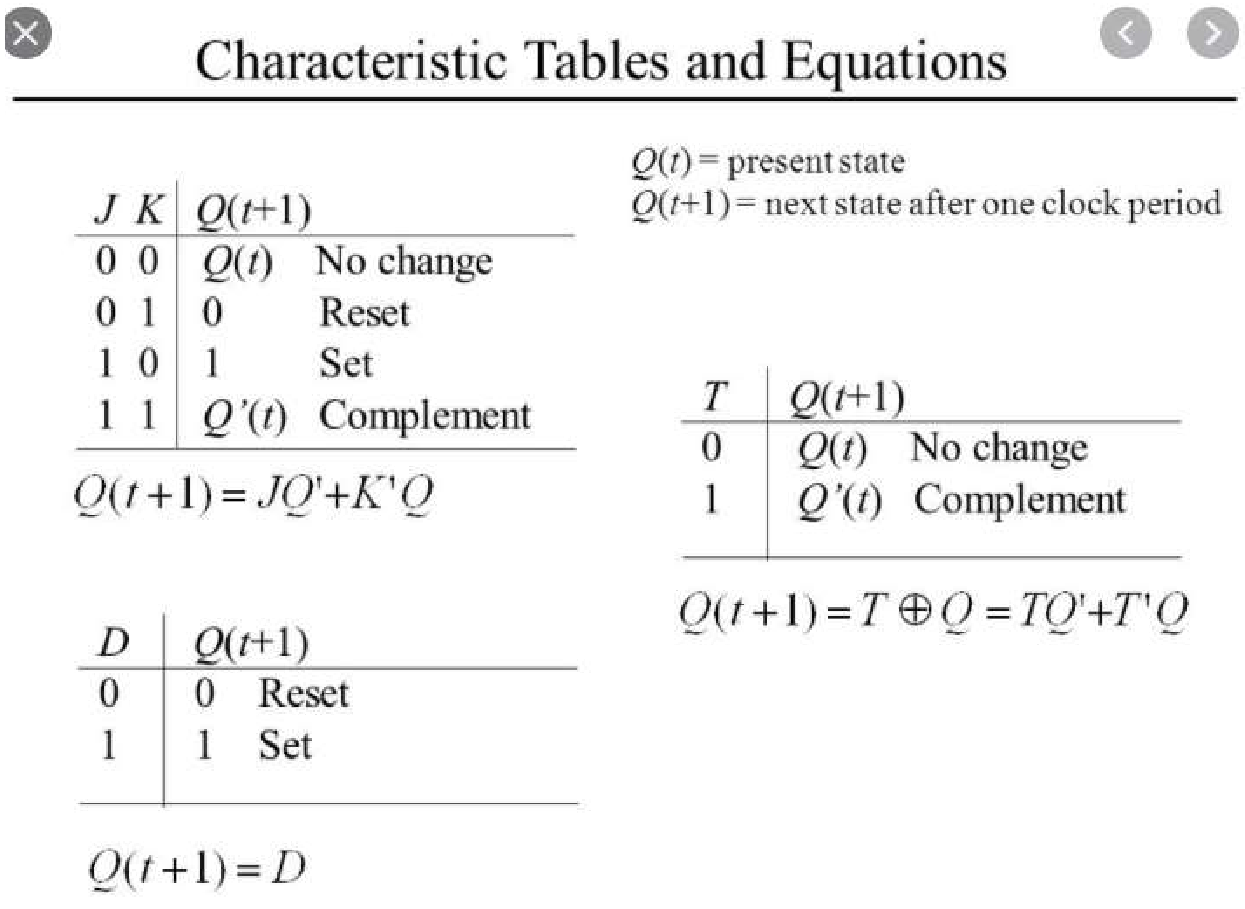
Laboratory 7 - Latches and Flip-flops

More information on GitHub Tomáš Frýza

My GitHub

Preparation of Laboratory



Characteristic Tables and Equations

$Q(t)$ = present state
$Q(t+1)$ = next state after one clock period

J	K	$Q(t+1)$	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

$$Q(t+1) = JQ' + K'Q$$

T	$Q(t+1)$	
0	$Q(t)$	No change
1	$Q'(t)$	Complement

$$Q(t+1) = T \oplus Q = TQ' + T'Q$$

D	$Q(t+1)$	
0	0	Reset
1	1	Set

$$Q(t+1) = D$$

$$D_{q_{n+1}} = D$$
$$JK_{q_{n+1}} = J\cdot \overline{q_{n}} + \overline{K}\cdot q_{n}$$
$$T_{q_{n+1}} = T\cdot \overline{q_{n}} + \overline{T}\cdot q_{n}/T \oplus q_{n}$$

```
D_{q_{n+1}} =  D\\
JK_{q_{n+1}} = J\cdot \overline{q_{n}} + \overline{K}\cdot q_{n}\\
T_{q_{n+1}} = T\cdot \overline{q_{n}} + \overline{T}\cdot q_{n}/T \oplus  q_{n}
```

clk	d	q(n)	q(n+1)	Comments
↑	0	0	0	No change
↑	0	1	0	Reset
↑	1	0	1	Set
↑	1	1	1	No change

clk	j	k	q(n)	q(n+1)	Comments
↑	0	0	0	0	No change
↑	0	0	1	1	No change

clk	j	k	q(n)	q(n+1)	Comments
↑	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

clk	t	q(n)	q(n+1)	Comments
↑	0	0	0	No change
↑	0	1	1	No change
↑	1	0	1	Toggle
↑	1	1	0	Toggle

Laboratory

Entity	Inputs	Outputs	Description
`d_ff_arst`	`clk`, `arst`, `d`	`q`, `q_bar`	D type flip-flop with an async reset
`d_ff_rst`	`clk`, `rst`, `d`	`q`, `q_bar`	D type flip-flop with a sync reset
`jk_ff_rst`	`clk`, `rst`, `j`, `k`	`q`, `q_bar`	JK type flip-flop with a sync reset
`t_ff_rst`	`clk`, `rst`, `t`	`q`, `q_bar`	T type flip-flop with a sync reset

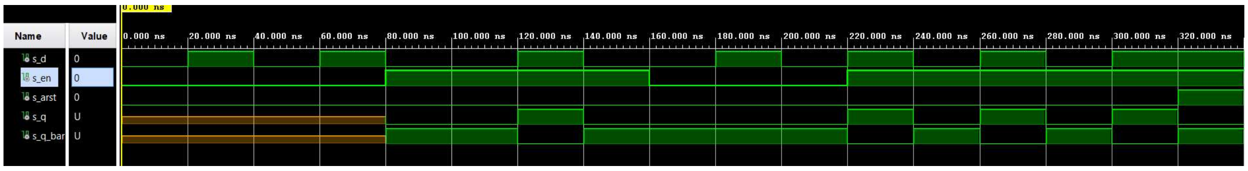
Port name	Direction	Type	Description
`BTNU`	in	`std_logic`	Clock emulator
`BTNC`	in	`std_logic`	Synchronous reset
`SW`	in	`std_logic_vector(1 - 1 downto 0)`	Shift register serial input
`LED`	out	`std_logic_vector(4 - 1 downto 0)`	Shift register parallel outputs

d_latch.vhd

```vhdl
p_d_latch : process(en, d, arst)
    begin
        if (arst = '1') then
            q     <= '0';
            q_bar <= '1';
        elsif (en = '1') then
            q     <= d;
            q_bar <= not d;
        end if;
    end process p_d_latch;
```

Testbench_d_latch.vhd

```vhdl
p_stimulus : process
    begin
        report "Stimulus process started" severity note;

        s_en   <= '0'; s_d <= '0'; s_arst <= '0'; wait for 20 ns;
        s_d    <= '1'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;
        s_d    <= '1'; wait for 20 ns;

        -- Test enable
        s_en   <= '1'; s_d <= '0'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;
        s_d    <= '1'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;
        s_en   <= '0'; wait for 20 ns;
        s_d    <= '1'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;

        -- Test async-reset
        s_en   <= '1'; s_d <= '1'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;
        s_d    <= '1'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;
        s_d    <= '1'; wait for 20 ns;
        s_arst <= '1'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;
        s_d    <= '1'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;
        s_d    <= '1'; wait for 20 ns;
        s_arst <= '0'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;
        s_d    <= '1'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;
        s_d    <= '1'; wait for 20 ns;
        s_en   <= '0'; wait for 20 ns;
        report "Stimulus process finished" severity note;
        wait;
    end process p_stimulus;
```



d_arst.vhd

```
p_d_ff_arst : process(clk, arst)
    begin
        if (arst = '1') then
            q     <= '0';
            q_bar <= '1';
        elsif rising_edge(clk) then
            q     <= d;
            q_bar <= not d;
        end if;
    end process p_d_ff_arst;
```

d_rst.vhd

```
p_d_ff_rst : process(clk)
    begin
        if rising_edge(clk) then
            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;
```

jk_rst.vhd

```
p_jk_ff_rst : process(clk)
    begin
        if rising_edge(clk) then
            if (rst = '1') then
                s_q     <= '0';
                s_q_bar <= '1';
            else
                if (j = '0' and k = '0') then
                    s_q     <= s_q;
                s_q_bar <= s_q_bar;
            elsif (j = '0' and k = '1') then
                s_q     <= '0';
                s_q_bar <= '1';
            elsif (j = '1' and k = '0') then
                s_q     <= '1';
                s_q_bar <= '0';
            else
                s_q     <= not s_q;
                s_q_bar <= not s_q_bar;
            end if;
        end if;
    end if;
end process p_jk_ff_rst;
```

t_rst.vhd

```
p_t_ff_rst : process(clk)
    begin
        if rising_edge(clk) then
            if (rst = '1') then
                s_q     <= '0';
                s_q_bar <= '1';
            elsif (t = '1') then
                s_q     <= not s_q;
                s_q_bar <= s_q;
            end if;
        end if;
    end process p_t_ff_rst;
```

Testbench_d_arst.vhd

```
p_clk_gen : process
    begin
        while now < 750 ns loop
            s_clk <= '0';
            wait for c_CLK_100MHZ_PERIOD / 2;
            s_clk <= '1';
            wait for c_CLK_100MHZ_PERIOD / 2;
        end loop;
        wait;
    end process p_clk_gen;

    p_reset_gen : process
    begin
        s_arst <= '0'; wait for 120 ns;
        s_arst <= '1'; wait for 100 ns;
        s_arst <= '0'; wait;
    end process p_reset_gen;
    p_stimulus : process
    begin
        report "Stimulus process started" severity note;

        s_d    <= '1'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;
        s_d    <= '1'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;
        s_d    <= '1'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;
        s_d    <= '1'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;
        s_d    <= '1'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;
        s_d    <= '1'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;
        s_d    <= '1'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;
        s_d    <= '1'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;
        s_d    <= '1'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;
        s_d    <= '1'; wait for 20 ns;
        s_d    <= '0'; wait for 20 ns;

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

Testbench_d_rst.vhd

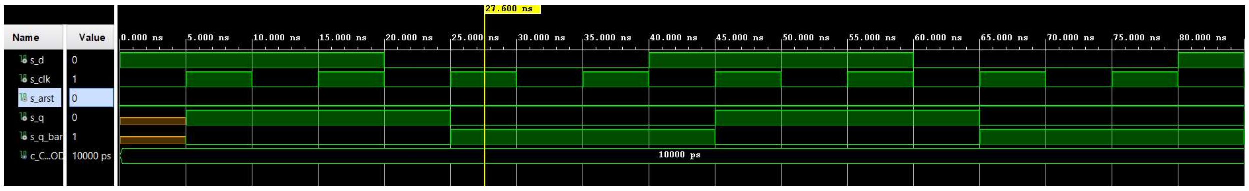
```
p_clk_gen : process
    begin
        while now < 750 ns loop
            s_clk <= '0';
            wait for c_CLK_100MHZ_PERIOD / 2;
            s_clk <= '1';
            wait for c_CLK_100MHZ_PERIOD / 2;
        end loop;
        wait;
    end process p_clk_gen;

    p_reset_gen : process
    begin
        s_rst <= '0'; wait for 120 ns;
        s_rst <= '1'; wait for 110 ns;
        s_rst <= '0'; wait;
    end process p_reset_gen;

    p_stimulus : process
    begin
        report "Stimulus process started" severity note;
        s_d <= '0'; wait for 20 ns;
        s_d <= '1'; wait for 20 ns;
        s_d <= '0'; wait for 20 ns;
        s_d <= '1'; wait for 20 ns;
        s_d <= '0'; wait for 20 ns;
        s_d <= '1'; wait for 20 ns;
        s_d <= '0'; wait for 20 ns;
        s_d <= '1'; wait for 20 ns;
        s_d <= '0'; wait for 20 ns;
        s_d <= '1'; wait for 20 ns;
        s_d <= '0'; wait for 20 ns;
        s_d <= '1'; wait for 20 ns;
        s_d <= '0'; wait for 20 ns;
        s_d <= '1'; wait for 20 ns;
        s_d <= '0'; wait for 20 ns;
        s_d <= '1'; wait for 20 ns;

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

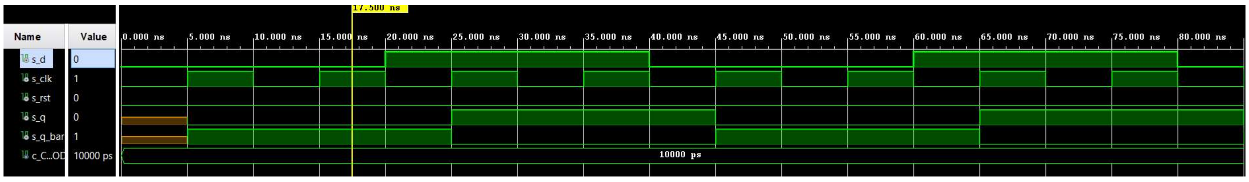
Testbench_jk_rst.vhd

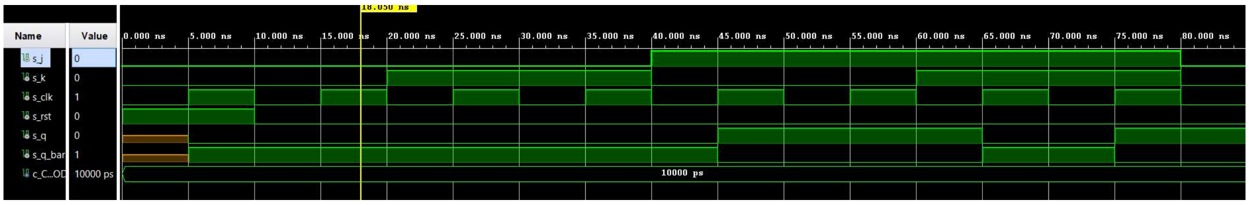
```
p_clk_gen : process
    begin
        while now < 750 ns loop
            s_clk <= '0';
            wait for c_CLK_100MHZ_PERIOD / 2;
            s_clk <= '1';
            wait for c_CLK_100MHZ_PERIOD / 2;
        end loop;
        wait;
    end process p_clk_gen;

    p_reset_gen : process
    begin
        s_rst <= '1'; wait for 10 ns;
        s_rst <= '0'; wait for 100 ns;
        s_rst <= '1'; wait for 100 ns;
        s_rst <= '0'; wait;
    end process p_reset_gen;

    p_stimulus : process
    begin
        report "Stimulus process started" severity note;
        s_j <= '0'; s_k <= '0'; wait for 20 ns;
        s_j <= '0'; s_k <= '1'; wait for 20 ns;
        s_j <= '1'; s_k <= '0'; wait for 20 ns;
        s_j <= '1'; s_k <= '1'; wait for 20 ns;
        s_j <= '0'; s_k <= '0'; wait for 20 ns;
        s_j <= '0'; s_k <= '1'; wait for 20 ns;
        s_j <= '1'; s_k <= '0'; wait for 20 ns;
        s_j <= '1'; s_k <= '1'; wait for 20 ns;
        s_j <= '0'; s_k <= '0'; wait for 20 ns;
        s_j <= '0'; s_k <= '1'; wait for 20 ns;
        s_j <= '1'; s_k <= '0'; wait for 20 ns;
        s_j <= '1'; s_k <= '1'; wait for 20 ns;
        s_j <= '0'; s_k <= '0'; wait for 20 ns;
        s_j <= '0'; s_k <= '1'; wait for 20 ns;
        s_j <= '1'; s_k <= '0'; wait for 20 ns;
        s_j <= '1'; s_k <= '1'; wait for 20 ns;
        report "Stimulus process finished" severity note;
        wait;
    end process p_stimulus;
```

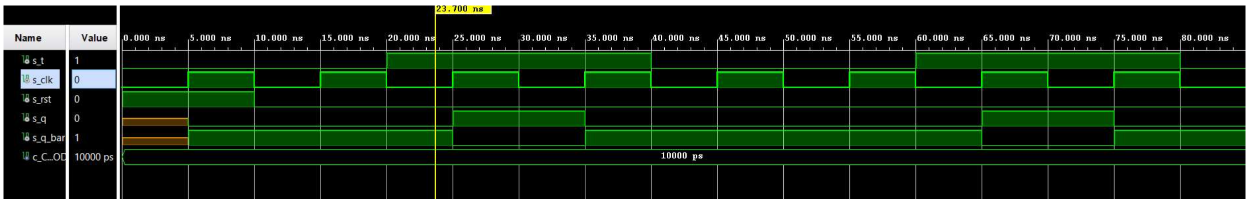
Testbench_t_rst.vhd

```
 p_clk_gen : process
    begin
        while now < 750 ns loop
            s_clk <= '0';
            wait for c_CLK_100MHZ_PERIOD / 2;
            s_clk <= '1';
            wait for c_CLK_100MHZ_PERIOD / 2;
        end loop;
        wait;
    end process p_clk_gen;

    p_reset_gen : process
    begin
        s_rst <= '1'; wait for 10 ns;
        s_rst <= '0'; wait for 100 ns;
        s_rst <= '1'; wait for 100 ns;
        s_rst <= '0'; wait;
    end process p_reset_gen;

    p_stimulus : process
    begin
        report "Stimulus process started" severity note;

        s_t <= '0'; wait for 20 ns;
        s_t <= '1'; wait for 20 ns;
        s_t <= '0'; wait for 20 ns;
        s_t <= '1'; wait for 20 ns;
        s_t <= '0'; wait for 20 ns;
        s_t <= '1'; wait for 20 ns;
        s_t <= '0'; wait for 20 ns;
        s_t <= '1'; wait for 20 ns;
        s_t <= '0'; wait for 20 ns;
        s_t <= '1'; wait for 20 ns;
        s_t <= '0'; wait for 20 ns;
        s_t <= '1'; wait for 20 ns;
        s_t <= '0'; wait for 20 ns;
        s_t <= '1'; wait for 20 ns;
        s_t <= '0'; wait for 20 ns;
        s_t <= '1'; wait for 20 ns;

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

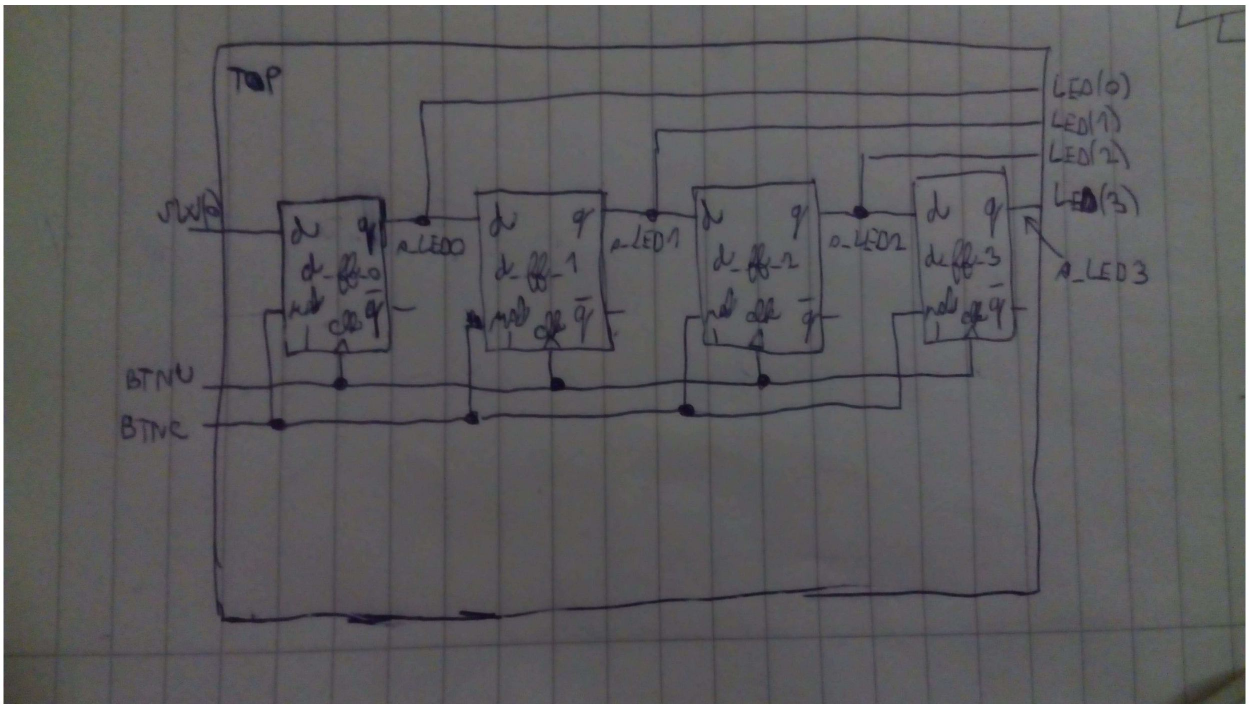


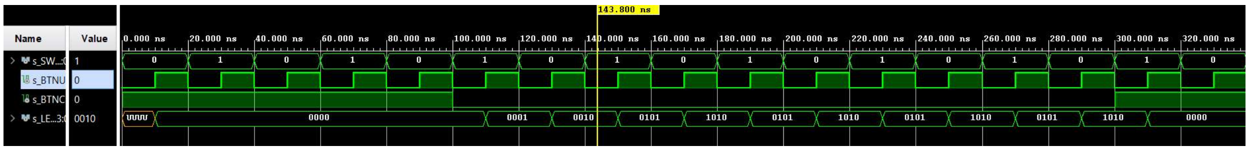






Shift register




143.800 ns
Name
Value
s_SW...
1
s_BTNU
0
s_BTNC
0
s_LE...3:
0010
0.000 ns
20.000 ns
40.000 ns
60.000 ns
80.000 ns
100.000 ns
120.000 ns
140.000 ns
160.000 ns
180.000 ns
200.000 ns
220.000 ns
240.000 ns
260.000 ns
280.000 ns
300.000 ns
320.000 ns
0000
0001
0010
0101
1010
0000