

# Lab 07

## 1. Preparation Tasks

### Characteristic equations

$$q_{n+1}^D = d$$

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

$$q_{n+1}^T = t\overline{q_n} + \overline{t}q_n$$

### d\_ff table

d	q(n)	q(n+1)	Comments
0	0	0	Not changed
0	1	0	Changed
1	0	1	Changed
1	1	1	No changed

### jk\_ff table

j	k	q(n)	q(n+1)	Comments
0	0	0	0	Not changed
0	0	1	1	Not changed
0	1	0	0	Reset
0	1	1	0	Reset
1	0	0	1	Set
1	0	1	1	Set
1	1	0	1	Inverted
1	1	1	0	Inverted

### t\_ff table

t	q(n)	q(n+1)	Comments
0	0	0	Not changed
0	1	1	Not changed
1	0	1	Inverted
1	1	0	Inverted

## 2. D latch

---

### VHDL code listing of the process p\_d\_latch

```

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

    elsif (arst = '1') then
        q <= '0';
        q_bar <= '1';

    end if;
end process p_d_latch;

```

### Listing of VHDL reset and stimulus processes from the testbench tb\_d\_latch.vhd

```

p_reset_gen : process
begin
    s_arst <= '0';
    wait for 38 ns;

    s_arst <= '1'; -- activated
    wait for 53 ns;

    s_arst <= '0'; -- deactivated
    wait for 80 ns;

    s_arst <= '1';

    wait;
end process p_reset_gen;

p_stimulus : process
begin
    -- without output

```

```

s_en    <= '0';
s_arst  <= '0';
s_d     <= '0';
wait for 10 ns;

s_arst  <= '1';
wait for 10 ns;

s_arst  <= '0';

-- Test 1
s_d     <= '0';
s_en    <= '0';
wait for 10ns;
s_arst  <= '0';

assert (s_q = '0' and s_q_bar = '1')
report "Error (d=0,en=0,arst=0)" severity error;

-- Test 2
s_d     <= '1';
s_en    <= '0';
wait for 10ns;
s_arst  <= '0';

assert (s_q = '0' and s_q_bar = '1')
report "Error (d=1,en=0,arst=0)" severity error;

-- Test 3
s_d     <= '0';
s_en    <= '1';
wait for 10ns;
s_arst  <= '0';

assert (s_q = '0' and s_q_bar = '1')
report "Error (d=0,en=1,arst=0)" severity error;

-- Test 4
s_d     <= '1';
s_en    <= '1';
wait for 10ns;
s_arst  <= '0';

assert (s_q = '1' and s_q_bar = '0')
report "Error (d=1,en=1,arst=0)" severity error;

-- Test 5
s_d     <= '1';
s_en    <= '1';
wait for 10ns;
s_arst  <= '1';

assert (s_q = '0' and s_q_bar = '1')
report "Error (d=1,en=1,arst=1)" severity error;

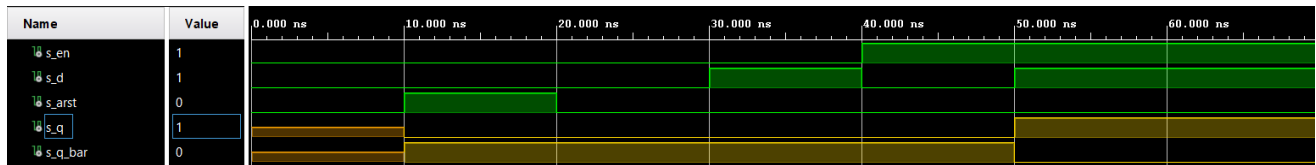
```

```

        wait;
    end process p_stimulus;

```

## Screenshot with simulated time waveforms



## 3. Flip-flops

VHDL code listing of the processes `p_d_ff_arst` , `p_d_ff_rst` , `p_jk_ff_rst` , `p_t_ff_rst`

```

p_d_ff_arst : process(clk)
begin
    if (rst = '1') then
        q      <= '0';
        q_bar  <= '1';
    elsif rising_edge(clk) then -- check rising
        q      <= d;
        q_bar  <= not d;
    end if;
end process p_d_ff_arst;

p_d_ff_rst : process(clk)
begin
    if rising_edge(clk) then -- only if rising
        if (rst = '1') then
            q      <= '0';
            q_bar  <= '1';
        elsif (rst = "0") then
            q      <= d;
            q_bar  <= not d;
        end if;
    end if;
end process p_d_ff_rst;

jk_ff_rst : process(clk)
begin
    if rising_edge(clk) then
        if (rst = '1') then
            s_q      <= '0';
            s_q_bar  <= '1';
        elsif (rst = "0") then
            if (j = '0' and k = '0') then -- not changed
                s_q      <= s_q;
                s_q_bar  <= s_q_bar;
            elsif (j = '0' and k = '1') then -- reset

```

```

        s_q      <= '0';
        s_q_bar <= '1';
    elsif (j = '1' and k = '0') then -- set
        s_q      <= '1';
        s_q_bar <= '0';
    elsif (j = '1' and k = '1') then -- inverted
        s_q      <= not s_q;
        s_q_bar <= not s_q_bar;
    end if;
end if;
end if;
end process jk_ff_rst;

t_ff_rst : process(clk)
begin
    if rising_edge(clk) then
        if (rst = '1') then
            s_q      <= '0';
            s_q_bar <= '1';
        else if (rst = '0') then
            if (t = '0') then -- not changed
                s_q      <= s_q;
                s_q_bar <= s_q_bar;
            elsif (t = '1') then -- inverted
                s_q      <= not s_q;
                s_q_bar <= not s_q_bar;
            end if;
        end if;
    end if;
end process t_ff_rst;

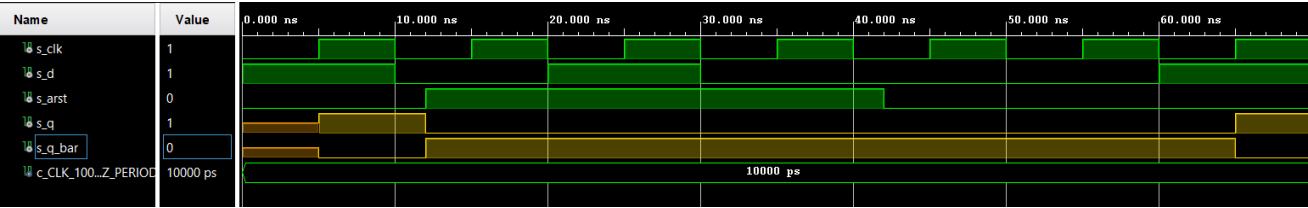
```

## Listing of VHDL clock, reset and stimulus processes from the testbench files

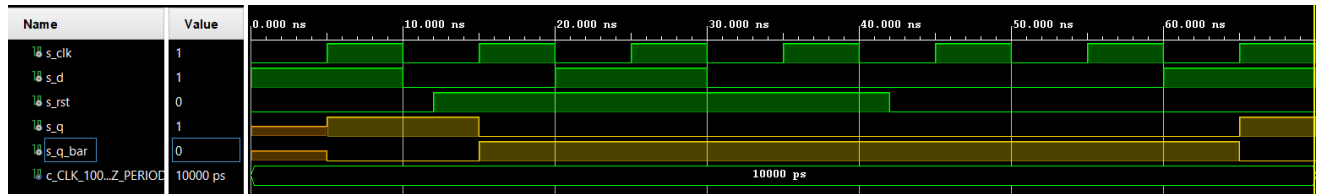
TODO: dopsat

## Screenshot with simulated time waveforms

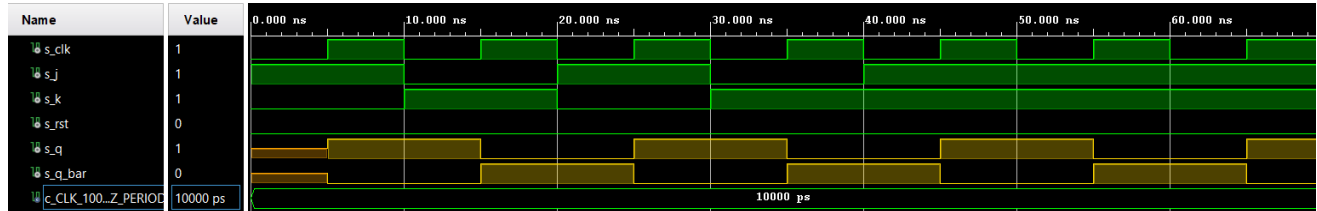
### d\_arst waveform



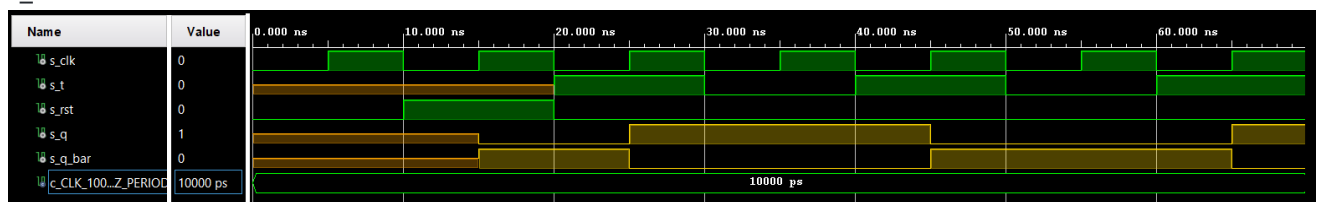
## d\_rst waveform



## jk\_rst waveform



## t\_rst waveform



## 4. Shift register

### Image of the driver schematic

