


 xnanko00 merge ...

3 minutes ago  History

...

 README.md

3 minutes ago

 README.md

1. Cvičenie

Preparation tasks

Characteristic equations and completed tables for D, JK, T flip-flops

Characteristic equations

$$q_{n+1}^D = d$$
$$q_{n+1}^{JK} = j\overline{q_n} + \overline{k}q_n$$
$$q_{n+1}^T = t\overline{q_n} + \overline{t}q_n$$

Completed tables

D	Qn	Q(n+1)	Comments
0	0	0	No change
0	1	0	Change
1	1	1	No change
1	0	1	Change

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

J	K	Qn	Q(n+1)	Comments
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
1	1	0	Toggle

2. Cvičenie

D latch

VHDL code of (d_latch)

```
architecture Behavioral of d_latch is
begin
  p_d_latch : process (d, arst, en)
  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;

end Behavioral;
```

VHDL reset and stimulus process (tb_d_latch.vhd)

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

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

  s_arst <= '0';

  wait for 600ns;
  s_arst <= '1';
  wait;
end process p_reset_gen;
```

-- Data generation process

```

-----
p_stimulus : process
begin
    report "Stimulus process started" severity note;
    s_d <= '0';
    s_en <= '0';

    s_d <= '1';
    wait for 10ns;
    assert (s_q = '0' and s_q_bar = '1') report "Failed at 1" severity note;

    s_d <= '0';
    wait for 10ns;
    assert (s_q = '0' and s_q_bar = '1') report "Failed at 2" severity note;

    -- Test en = 1
    s_en <= '1';
    s_d <= '1';
    wait for 10ns;
    assert (s_q = '1' and s_q_bar = '0') report "Failed at 3" severity note;

    s_en <= '1';
    s_d <= '0';
    wait for 10ns;
    assert (s_q = '0' and s_q_bar = '1') report "Failed at 4" severity note;

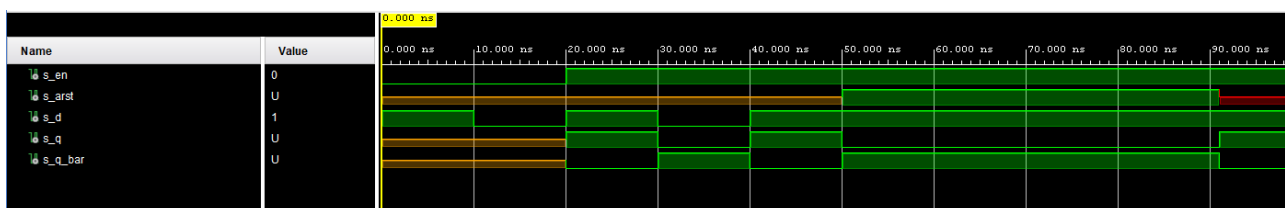
    -- Test en=1 arst=1
    s_en <= '1';
    s_d <= '1';
    wait for 10ns;
    s_arst <= '1';
    assert (s_q = '0' and s_q_bar = '1') report "Failed at 5" severity note;

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

end Behavioral;

```

Screenshot with waveforms



3. Cvičenie

Flip-flops

VHDL code of (d_ff_arst)

```

architecture Behavioral of d_ff_arst is

begin

    p_d_latch : process (arst, clk)

```

```

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_latch;

end Behavioral;

```

VHDL clock, reset and stimulus processes of (tb_d_ff_arst)

```

p_clk_gen : process
begin
    while now < 750 ns loop          -- 75 periods of 100MHz clock
        s_clk_100MHz <= '0';
        wait for c_CLK_100MHZ_PERIOD / 2;
        s_clk_100MHz <= '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 28 ns;

    -- Reset activated
    s_arst <= '1';
    wait for 13 ns;

    s_arst <= '0';

    wait for 17ns;
    s_arst <= '1';

    wait for 600ns;
    s_arst <= '1';
    wait;
end process p_reset_gen;

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

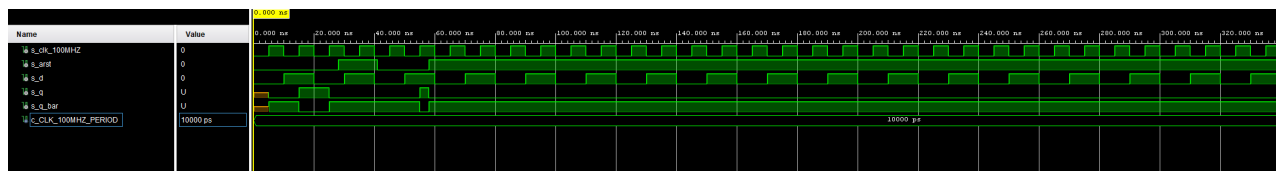
    wait for 10 ns;
    s_d <= '1';
    wait for 10 ns;
    s_d <= '0';
    wait for 10 ns;
    s_d <= '1';
    wait for 10 ns;
    s_d <= '0';
    wait for 10 ns;
    s_d <= '1';
    wait for 10 ns;
    s_d <= '0';

```

```
end process p_stimulus;
```

```
end Behavioral;
```

Screenshot with waveforms



VHDL code of (d_ff_rst)

```
architecture Behavioral of d_ff_rst is
begin
  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;

end Behavioral;
```

VHDL clock, reset and stimulus processes of (tb_d_ff_rst)

```
p_clk_gen : process
begin
  while now < 750 ns loop
    s_clk_100MHz <= '0';
    wait for c_CLK_100MHz_PERIOD / 2;
    s_clk_100MHz <= '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 28 ns;
  s_rst <= '1';
  wait for 13 ns;
  s_rst <= '0';
  wait;
end process p_reset_gen;

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

  s_d <= '1';
  wait for 10ns;
```

```

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

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

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

wait for 20ns;
s_d <= '1';
wait for 10ns;

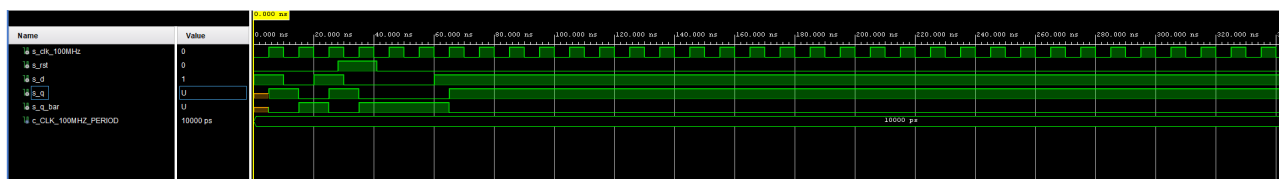
report "Stimulus process finished" severity note;

wait;
end process p_stimulus;

end Behavioral;

```

Screenshot with waveforms



VHDL code of (jk_ff_rst)

```

architecture Behavioral of jk_ff_rst is
    signal s_q : std_logic;
begin
    p_d_latch : process (clk)
    begin
        if rising_edge(clk) then
            if (rst = '1') then
                s_q <= '0';
            else
                if(j='0' and k='0') then
                    s_q <= s_q;

                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_d_latch;

    q <= s_q;
    q_bar <= not s_q;

end Behavioral;

```

VHDL clock, reset and stimulus processes of (tb_jk_ff_rst)

```

p_clk_gen : process
begin
    while now < 750 ns loop          -- 75 periods of 100MHz clock
        s_clk_100MHz <= '0';
        wait for c_CLK_100MHZ_PERIOD / 2;
        s_clk_100MHz <= '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 28 ns;

    -- Reset activated
    s_rst <= '1';
    wait for 13 ns;

    s_rst <= '0';

    wait for 17ns;
    s_rst <= '1';

    wait for 600ns;
    s_rst <= '1';
    wait;
end process p_reset_gen;
p_stimulus : process
begin
    report "Stimulus process started" severity note;
    s_j <='0';
    s_k <='0';

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

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

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

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

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

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

    wait for 10 ns;

```

```

s_j <='1';
s_k <='0';

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

end process p_stimulus;

end Behavioral;

```

Screenshot with waveforms



VHDL code of (t_ff_rst)

```

architecture Behavioral of t_ff_rst is
    signal s_q      : STD_LOGIC;
    signal s_q_bar  : STD_LOGIC;
begin

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

    q      <= s_q;
    q_bar  <= s_q_bar;
end Behavioral;

```

VHDL clock, reset and stimulus processes of (tb_t_ff_rst)

```

p_clk_gen : process
begin
    while now < 750 ns loop
        s_clk_100MHz <= '0';
        wait for c_CLK_100MHZ_PERIOD / 2;
        s_clk_100MHz <= '1';
        wait for c_CLK_100MHZ_PERIOD / 2;
    end loop;
    wait;
end process p_clk_gen;

```



```

-- reset
p_reset_gen : process
begin
    s_rst <= '0';
    wait for 28 ns;
    s_rst <= '1';
    wait for 13 ns;
    s_rst <= '0';
    wait;
end process p_reset_gen;

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

    wait for 20ns;
    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;

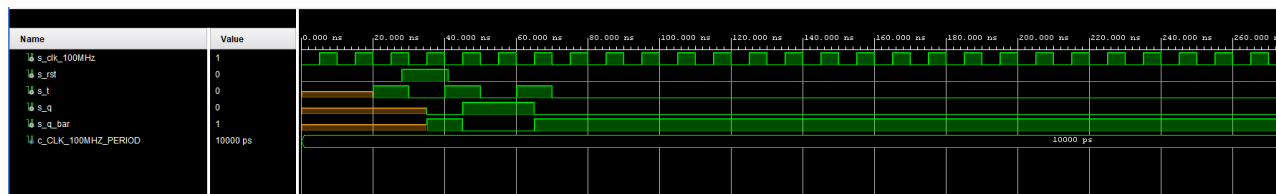
    report "Stimulus process finished" severity note;

    wait;
end process p_stimulus;

end Behavioral;

```

Screenshot with waveforms



4. Cvičenie

Shift register

Image of the shift register schematic

4 bit shift register

