

 david3891 / Digital-electronics-1

<> Code

! Issues

🔗 Pull requests

▶ Actions

📁 Projects

📖 Wiki

🛡 Security

🔑 main ▼

...

[Digital-electronics-1](#) / [Labs](#) / [07-ffs](#) / README.md

david3891 Update README.md

🕒 History

👤 1 contributor

Raw

Blame



621 lines (487 sloc) | 17.8 KB

Labs 07-ffs

1) Charakteristické rovnice a dokončené tabulky pro klopné obvody D, JK, T

🔗 a) klopný obvod D

clk	d	q(n)	q(n+1)	Comments
↑	0	0	0	Clear (Reset)
↑	0	1	0	Clear (Reset)
↑	1	0	1	Set
↑	1	1	1	Set

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

b) klopný obvod JK

clk	j	k	q(n)	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

$$q_{n+1}^{JK} = j \cdot \bar{q}_n + \bar{k} \cdot q_n$$

c) klopný obvod T

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

$$q_{n+1}^T = t \cdot \bar{q}_n + \bar{t} \cdot q_n$$

2) D latch

a) Vypis kódu VHDL procesu p_d_latch

```

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;

```

```
end if;  
end process p_d_latch;
```

b) Výpis resetovacích a stimulačních procesů VHDL ze souboru testbench tb_d_latch.vhd

```
p_arst_gen : process  
begin  
    s_arst <= '0';  
    wait for 10 ns;  
  
    -- arst activated  
    s_arst <= '1';  
    wait for 10 ns;  
  
    -- arst deactivated  
    s_arst <= '0';  
    wait for 200 ns;  
  
    s_arst <= '1';  
  
    wait;  
end process p_arst_gen;  
  
p_stimulus : process  
begin  
    report "Stimulus process started" severity note;  
  
    s_en <= '0';  
    s_d <= '0';  
  
    wait for 6 ns;  
    s_d <= '1';  
    wait for 6 ns;  
    s_d <= '0';  
    wait for 6 ns;  
    s_d <= '1';  
    wait for 6 ns;  
    s_d <= '0';  
    wait for 6 ns;  
    s_d <= '1';  
    wait for 6 ns;  
    s_d <= '0';  
    wait for 6 ns;  
  
    s_en <= '1';  
    wait for 6 ns;  
    s_d <= '1';  
    wait for 6 ns;
```

```
s_d    <= '0';
wait for 6 ns;
s_d    <= '1';
wait for 6 ns;
s_d    <= '0';
wait for 6 ns;
s_d    <= '1';
wait for 6 ns;
s_d    <= '0';
wait for 6 ns;
s_en   <= '0';
wait for 6 ns;
s_d    <= '1';
wait for 6 ns;

s_en   <= '1';
wait for 6 ns;
s_d    <= '1';
wait for 6 ns;
s_d    <= '0';
wait for 6 ns;
    s_d    <= '1';
wait for 6 ns;
s_d    <= '0';
wait for 6 ns;
    s_d    <= '1';
wait for 6 ns;
s_d    <= '0';
wait for 6 ns;

s_d    <= '1';
wait for 6 ns;
s_en   <= '0';
wait for 6 ns;
s_d    <= '0';
wait for 6 ns;
s_d    <= '1';
wait for 6 ns;
s_d    <= '0';
wait for 6 ns;
s_d    <= '1';
wait for 6 ns;
s_en   <= '1';
wait for 6 ns;
s_d    <= '0';
wait for 6 ns;
    s_d    <= '1';
wait for 6 ns;
s_d    <= '0';
wait for 6 ns;
```

```

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

p_assert : process
begin
    wait for 100 ns;

    assert(s_q = '1' and s_q_bar = '0')
    report "Error - conditions in 100 ns are not met" severity error;

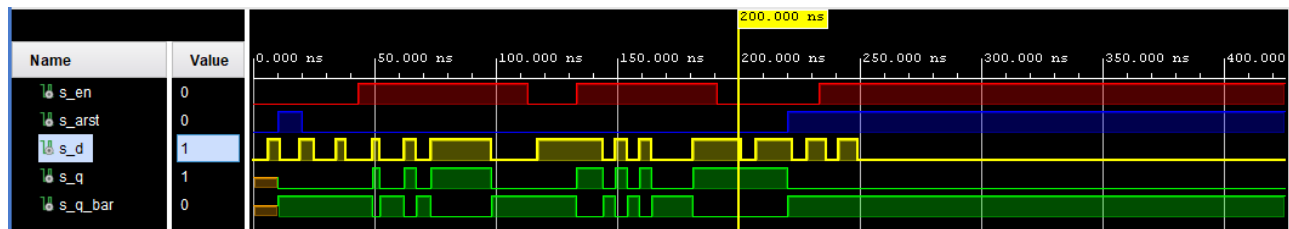
    wait for 50 ns;

    assert(s_q = '0' and s_q_bar = '1')
    report "Error - conditions in 150 ns are not met" severity error;

end process p_assert;

```

c) Screenshot se simulovanými časovými průběhy



3) Flip-flops

a) p_d_ff_arst

VHDL kód výpis procesů

```

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;

```

Výpis clock VHDL, resetování a stimulačních procesů ze souborů testbench

```
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_arst_gen : process
begin
    s_arst <= '0';
    wait for 58 ns;

    -- arst activated
    s_arst <= '1';
    wait for 15 ns;

    -- arst deactivated
    s_arst <= '0';

    wait;
end process p_arst_gen;

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

    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';
    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;
```

```

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

    p_assert : process
    begin
        wait for 27 ns;

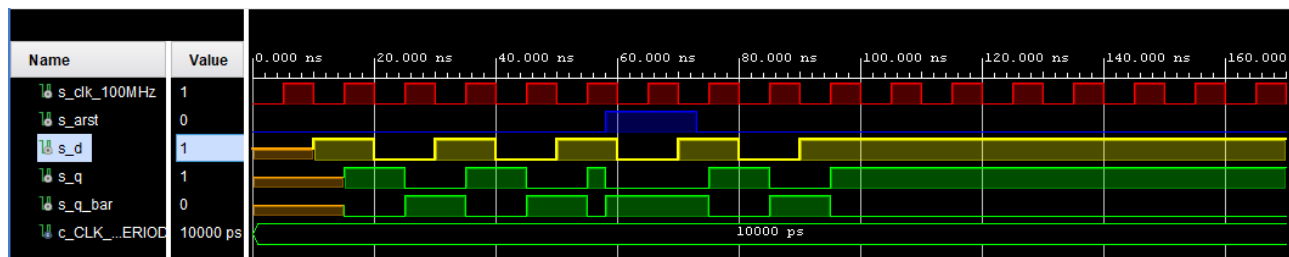
        -- assert in 50 ns
        assert(s_q = '0' and s_q_bar = '1')
        report "Error - conditions in 50 ns are not met" severity error;

        wait for 53 ns;
        -- assert in 50 ns
        assert(s_q = '1' and s_q_bar = '0')
        report "Error - conditions in 100 ns are not met" severity error;

    end process p_assert;

```

Screenshot se simulovanými časovými průběhy



b) p_d_ff_rst

VHDL kód výpis procesů

```

p_d_ff_rst : process (clk)
begin
    if rising_edge(clk) then
        q      <= d;
        q_bar <= not d;

    end if;
end process p_d_ff_rst;

```

Výpis clock VHDL, resetování a stimulačních procesů ze souborů testbench

```
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_rst_gen : process
begin
    s_rst <= '0';
    wait for 58 ns;

    -- rst activated
    s_rst <= '1';
    wait for 15 ns;

    -- rst deactivated
    s_rst <= '0';

    wait;
end process p_rst_gen;

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

    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';
    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';
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;

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

p_assert : process
begin
    wait for 50 ns;

    assert(s_q = '0' and s_q_bar = '1')
    report "Error - conditions in 50 ns are not met" severity error;

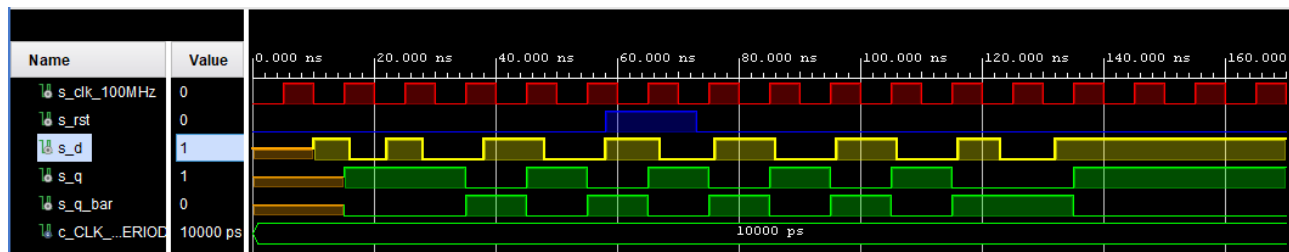
    wait for 30 ns;

    assert(s_q = '1' and s_q_bar = '0')
    report "Error - conditions in 80 ns are not met" severity error;

end process p_assert;

```

Screenshot se simulovanými časovými průběhy



c) p_jk_ff_rst

VHDL kód výpis procesů

```

jk_ff_rst : 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 jk_ff_rst;

q      <= s_q;
q_bar <= not s_q;

```

Výpis clock VHDL, resetování a stimulačních procesů ze souborů testbench

```

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_rst_gen : process
begin
    s_rst <= '0';
    wait for 58 ns;

    -- rst activated
    s_rst <= '1';
    wait for 15 ns;

    -- rst deactivated
    s_rst <= '0';

    wait;
end process p_rst_gen;

```

```

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

    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';

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

p_assert : process
begin
    wait for 50 ns;

    assert(s_q = '0' and s_q_bar = '1')
    report "Error - conditions in 50 ns are not met" severity error;

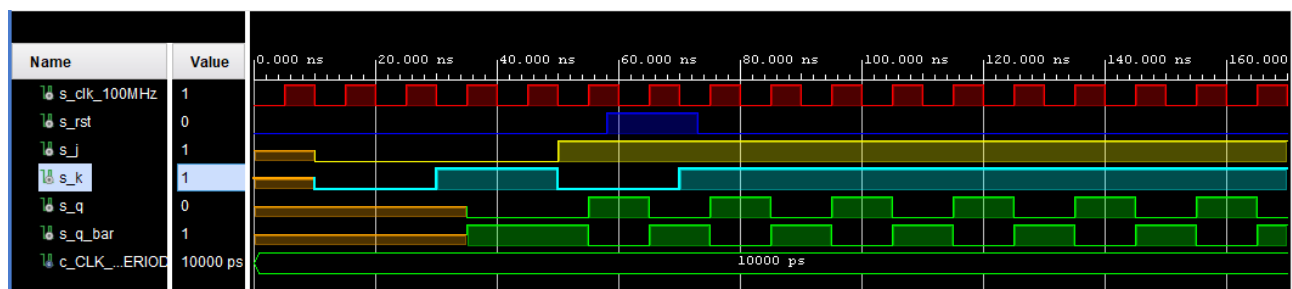
    wait for 25 ns;

    assert(s_q = '0' and s_q_bar = '1' and s_rst = '1')
    report "Error - conditions in 75 ns are not met" severity error;

end process p_assert;

```

Screenshot se simulovanými časovými průběhy



d) p_t_ff_rst

VHDL kód výpis procesů

```
p_t_ff_rst : process (clk)
begin

    if rising_edge(clk) then
        if (rst = '1') then
            s_q <= '0';

        else
            if (t = '0') then
                s_q <= s_q;

            elsif (t = '1') then
                s_q <= not s_q;

            end if;
        end if;
    end if;
end process p_t_ff_rst;
q          <= s_q;
q_bar      <= not s_q;
```

Výpis clock VHDL, resetování a stimulačních procesů ze souborů testbench

```
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_rst_gen : process
begin
    s_rst <= '0';
    wait for 20 ns;

    -- rst activated
    s_rst <= '1';
    wait for 15 ns;

    -- rst deactivated
    s_rst <= '0';
```

[illegible]

```

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

p_assert : process
begin
    wait for 50 ns;

    assert(s_q = '1' and s_q_bar = '0')
    report "Error - conditions in 50 ns are not met" severity error;

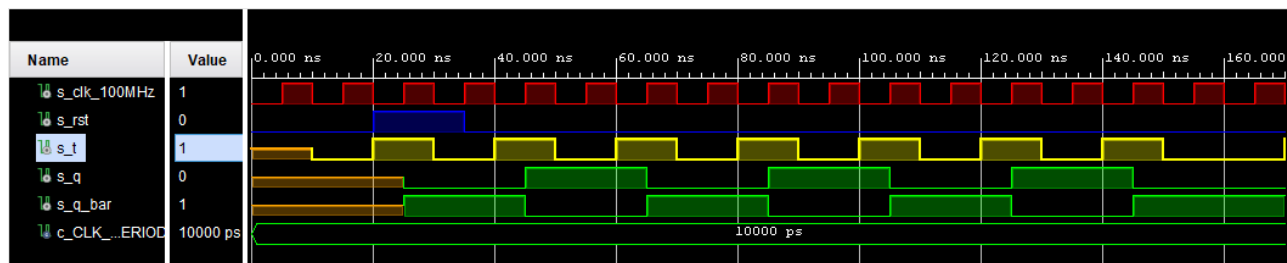
    wait for 20 ns;

    assert(s_q = '0' and s_q_bar = '1')
    report "Error - conditions in 70 ns are not met" severity error;

end process p_assert;

```

Screenshot se simulovanými časovými průběhy



4) Obrázek schématu posuvného registru

