#Digital-Electronics-1

https://github.com/AnaSampaio13/Digital-Electronics-1

##Exercise 1

###Characteristic equations

####D-type flip-flop

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

####JK-type flip-flop

$$q_{n+1} = j\overline{q}_n + \overline{k}q_n$$

####T-type flip-flop

$$q_{n+1} = t\overline{q}_n + \overline{t}q_n$$

###Truth tables

####D-type flip-flop

clk	d	q(n)	q(n+1)	Comments
↑	0	0	0	No change
↑	0	1	0	No change
↑	1	0	1	Sampled and stored
↑	1	1	1	Sampled and stored

####JK-type flip-flop

clk	j	k	q(n)	q(n+1)	Comments

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	Toggles
↑	1	1	1	0	Toggles

####T-type flip-flop

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

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

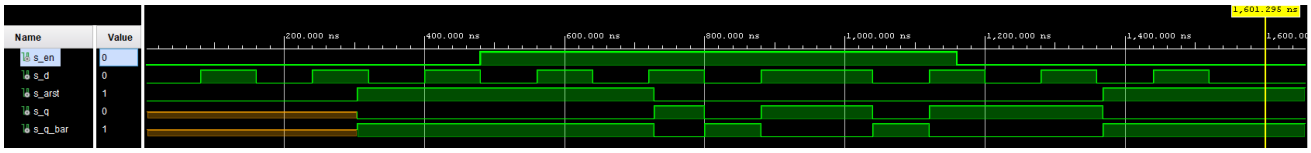
##Exercise 2

```
---------------------------------------------------------------------------------
###**Listing of VHDL code of the process**

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;
-------------------------------------------------------------------------
###**Listing of VHDL testbench file**

-------------------------------------------------------------------
-- Reset generation process
-------------------------------------------------------------------
    p_reset_gen : process
        begin
                s_arst <= '0';
                wait for 304 ns;
                s_arst <= '1';  -- Reset
                wait for 424 ns;
                s_arst <= '0';
                wait for 640 ns;
                s_arst <= '1';  -- Reset
                wait;
    end process p_reset_gen;

-------------------------------------------------------------------
```

```
    -- Data generation process
    ---------------------------------------------------------------------
        p_stimulus : process
            begin
                    report "Stimulus process started" severity note;

                    s_en <= '0';
                    s_d  <= '0';
                    wait for 80 ns;
                    s_d  <= '1';
                    wait for 80 ns;
                    s_d  <= '0';
                    wait for 80 ns;
                    s_d  <= '1';
                    wait for 80 ns;
                    s_d  <= '0';
                    wait for 40 ns;

                    assert ((s_arst = '1') and (s_en = '0') and (s_q = '0') and (s_q_bar =
                    report "Test failed for RESET is '1', EN is '0' and D is '0'" severity

                    wait for 40 ns;
                    s_d  <= '1';
                    wait for 40 ns;

                    assert ((s_arst = '1') and (s_en = '0') and (s_q = '0') and (s_q_bar =
                    report "Test failed for RESET is '1', EN is '0' and D is '1'" severity

                    wait for 40 ns;
                    s_d  <= '0';
                    s_en <= '1';
                    wait for 80 ns;
                    s_d  <= '1';
                    wait for 40 ns;

                    assert ((s_arst = '1') and (s_en = '1') and (s_q = '0') and (s_q_bar =
                    report "Test failed for RESET is '1', EN is '1' and D is '1'" severity

                    wait for 40 ns;
                    s_d  <= '0';
                    wait for 40 ns;

                    assert ((s_arst = '1') and (s_en = '1') and (s_q = '0') and (s_q_bar =
                    report "Test failed for RESET is '1', EN is '1' and D is '0'" severity

                    wait for 40 ns;
                    s_d  <= '1';
                    wait for 80 ns;
                    s_d  <= '0';
                    wait for 80 ns;
                    s_d  <= '1';
                    wait for 40 ns;

                    assert ((s_arst = '0') and (s_en = '1') and (s_q = '1') and (s_q_bar =
                    report "Test failed for RESET is '0', EN is '1' and D is '1'" severity
```

```
        wait for 120 ns;
        s_d  <= '0';
        wait for 40 ns;

        assert ((s_arst = '0') and (s_en = '1') and (s_q = '0') and (s_q_bar =
        report "Test failed for RESET is '0', EN is '1' and D i '0'" severity

        wait for 40 ns;
        s_d  <= '1';
        wait for 40 ns;
        s_en <= '0';
        wait for 40 ns;
        s_d  <= '0';
        wait for 80 ns;
        s_d  <= '1';
        wait for 80 ns;
        s_d  <= '0';
        wait for 80 ns;
        s_d  <= '1';
        wait for 80 ns;
        s_d  <= '0';

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

###Screenshot



##Exercise 3

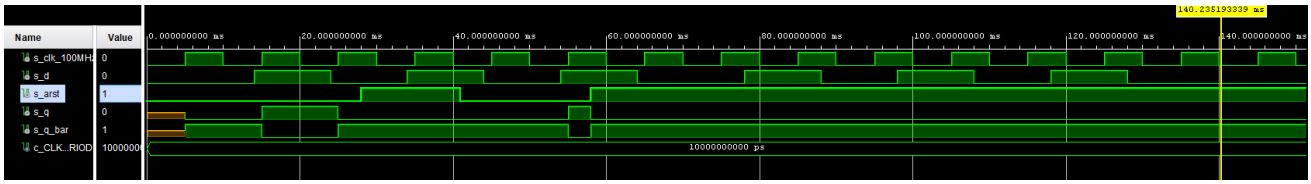
```
---------------------------------------------------------------------------------
###**Listing of VHDL code of the process p_d_ff_arst**

 p_d_ff_arst : process (clk, arst)
        begin
           if (arst = '1') then
              q     <= '0';
              q_bar <= '1';
           elsif rising_edge(clk) then
              q     <= d;
```

README.md

```
    ------------------------------------------------------------------------
    ###**Listing of VHDL clock, reset and stimulus processes from the testbench files tb_

    ------------------------------------------------------------------------
    -- Clock generation process
    ------------------------------------------------------------------------
        p_clk_gen : process
            begin
                while now < 1000 ms 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 generation process
    ------------------------------------------------------------------------

         p_reset_gen : process
            begin
                s_arst <= '0';
                wait for 28 ms;
                s_arst <= '1';  -- Reset
                wait for 13 ms;
                s_arst <= '0';
                wait for 17 ms;
                s_arst <= '1';
                wait for 693 ms;
                s_arst <= '1';  -- Reset
                wait;
         end process p_reset_gen;

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

                s_d  <= '0';
                wait for 14 ms;
                s_d  <= '1';
                wait for 2 ms;

                assert ((s_arst = '0') and (s_q = '1') and (s_q_bar = '0'))
                report "Test failed when RESET is '0', CLK is RISING and D is '1'" severi

                wait for 8 ms;
                s_d  <= '0';
                wait for 10 ms;
                s_d  <= '1';
                wait for 10 ms;
```

```
            s_d  <= '0';
            wait for 10 ms;
            s_d  <= '1';
            wait for 5 ms;

            assert ((s_arst = '1') and (s_q = '0') and (s_q_bar = '1'))
            report "Test failed when RESET is '1', CLK is FALLING and D is '1'" sever

            wait for 5 ms;
            s_d  <= '0';
            wait for 14 ms;
            s_d  <= '1';
            wait for 10 ms;
            s_d  <= '0';
            wait for 10 ms;
            s_d  <= '1';
            wait for 10 ms;
            s_d  <= '0';
            wait for 10 ms;
            s_d  <= '1';
            wait for 10 ms;
            s_d  <= '0';

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

Screenshot



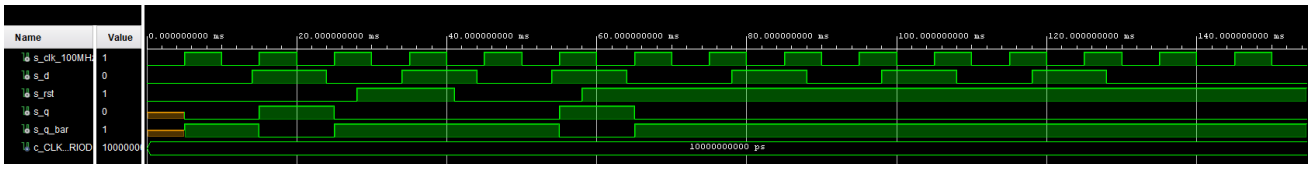
Listing of VHDL code of the process p_d_ff_rst

```
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;
-------------------------------------------------------------------------
###**Listing of VHDL clock, reset and stimulus processes from the testbench files tb_

-------------------------------------------------------------------------
-- Clock generation process
-------------------------------------------------------------------------
    p_clk_gen : process
        begin
               while now < 1000 ms 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 generation process
-------------------------------------------------------------------------

    p_reset_gen : process
           begin
                  s_rst <= '0';
                  wait for 28 ms;
                  s_rst <= '1';  -- Reset
                  wait for 13 ms;
                  s_rst <= '0';
                  wait for 17 ms;
                  s_rst <= '1';  -- Reset
                  wait for 33 ms;
                  wait for 660 ms;
                  s_rst <= '1';  -- Reset
                  wait;
    end process p_reset_gen;

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

               s_d  <= '0';
               wait for 14 ms;
               s_d  <= '1';
               wait for 2 ms;

               assert ((s_rst = '0') and (s_q = '1') and (s_q_bar = '0'))
               report "Test failed for RESET is '0', CLK is RISING and D is '1'" seve

               wait for 8 ms;
               s_d  <= '0';
               wait for 6 ms;
```

```
                wait for 4 ms;
                s_d  <= '1';
                wait for 10 ms;
                s_d  <= '0';
                wait for 10 ms;
                s_d  <= '1';
                wait for 5 ms;

                assert ((s_rst = '1') and (s_q = '1') and (s_q_bar = '0'))
                report "Test failed for RESET is '1', CLK is FALLING and D is '1'" sev

                wait for 5 ms;
                s_d  <= '0';
                wait for 14 ms;
                s_d  <= '1';
                wait for 10 ms;
                s_d  <= '0';
                wait for 10 ms;
                s_d  <= '1';
                wait for 10 ms;
                s_d  <= '0';
                wait for 10 ms;
                s_d  <= '1';
                wait for 10 ms;
                s_d  <= '0';

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

###Screenshot



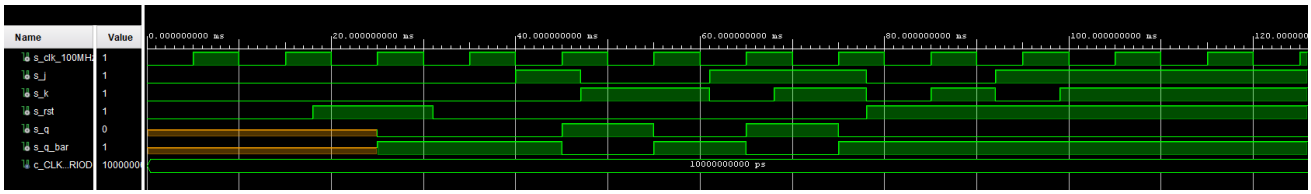
###**Listing of VHDL code of the process p_jk_ff_rst**

```
p_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 p_jk_ff_rst;

    q <= s_q;
    q_bar <= not s_q;
-------------------------------------------------------------------
###**Listing of VHDL clock, reset and stimulus processes from the testbench files tb_

-------------------------------------------------------------------
-- Clock generation process
-------------------------------------------------------------------
    p_clk_gen : process
        begin
               while now < 1000 ms 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 generation process
-------------------------------------------------------------------

    p_reset_gen : process
           begin
                  s_rst <= '0';
                  wait for 18 ms;
                  s_rst <= '1';  -- Reset
                  wait for 13 ms;
                  s_rst <= '0';
                  wait for 47 ms;
                  s_rst <= '1';  -- Reset
                  wait for 693 ms;
                  s_rst <= '1';  -- Reset
                  wait;
    end process p_reset_gen;

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

               s_j  <= '0';
```

```
        s_k  <= '0';
        wait for 38 ms;

        assert ((s_rst = '0') and (s_j = '0') and (s_k = '0') and (s_q = '0')
        report "Test failed when RESET is '0', CLK is RISING, J is '0' and K i

        wait for 2 ms;
        s_j  <= '1';
        s_k  <= '0';
        wait for 6 ms;

        assert ((s_rst = '0') and (s_j = '1') and (s_k = '0') and (s_q = '1')
        report "Test failed when RESET is '0', CLK is RISING, J is '1' and K i

        wait for 1 ms;
        s_j  <= '0';
        s_k  <= '1';
        wait for 13 ms;

        assert ((s_rst = '0') and (s_j = '0') and (s_k = '1') and (s_q = '0')
        report "Test failed when RESET is '0', CLK is RISING, J is '0' and K i

        wait for 1 ms;
        s_j  <= '1';
        s_k  <= '0';
        wait for 7 ms;
        s_j  <= '1';
        s_k  <= '1';
        wait for 8 ms;

        assert ((s_rst = '0') and (s_j = '1') and (s_k = '1') and (s_q = '0')
        report "Test failed when RESET is '0', CLK is RISING, J is '1' and K i

        wait for 2 ms;
        s_j  <= '0';
        s_k  <= '0';
        wait for 7 ms;
        s_j  <= '0';
        s_k  <= '1';
        wait for 7 ms;
        s_j  <= '1';
        s_k  <= '0';
        wait for 7 ms;
        s_j  <= '1';
        s_k  <= '1';

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

###Screenshot



###**Listing of VHDL code of the process p_t_ff_rst**

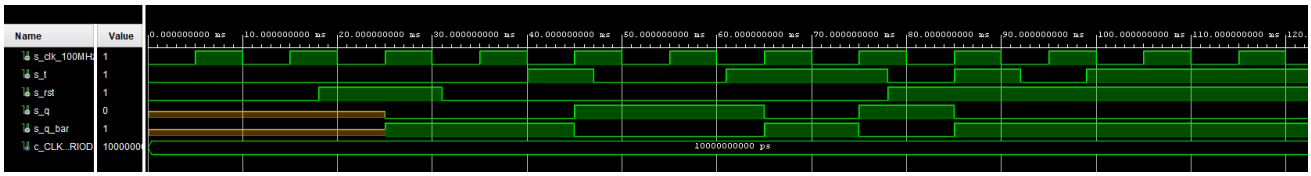
```
p_t_ff_rst : process (clk)
        begin
            if rising_edge(clk) then
                if (rst = '1') then
                    s_q <= '0';
                elsif (t = '1') then
                    s_q <= not s_q;
                end if;
            end if;
    end process p_t_ff_rst;

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

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

```
-------------------------------------------------------------------------
-- Clock generation process
-------------------------------------------------------------------------
    p_clk_gen : process
        begin
                while now < 1000 ms 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 generation process
-------------------------------------------------------------------------

    p_reset_gen : process
           begin
                  s_rst <= '0';
                  wait for 18 ms;
                  s_rst <= '1';  -- Reset
                  wait for 13 ms;
                  s_rst <= '0';
                  wait for 47 ms;
                  s_rst <= '1';  -- Reset
```

```
                wait for 693 ms;
                s_rst <= '1';  -- Reset
                wait;
    end process p_reset_gen;

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

                s_t  <= '0';
                wait for 38 ms;

                assert ((s_rst = '0') and (s_t = '0') and (s_q = '0') and (s_q_bar = '
                report "Test failed when RESET is '0', CLK is RISING and T is '0'" sev

                wait for 2 ms;
                s_t  <= '1';
                wait for 6 ms;

                assert ((s_rst = '0') and (s_t = '1') and (s_q = '1') and (s_q_bar = '
                report "Test failed when RESET is '0', CLK is RISING and T is '1'" sev

                wait for 1 ms;
                s_t  <= '0';
                wait for 13 ms;

                assert ((s_rst = '0') and (s_t = '0') and (s_q = '1') and (s_q_bar = '
                report "Test failed when RESET is '0', CLK is RISING and T is '0'" sev

                wait for 1 ms;
                s_t  <= '1';
                wait for 5 ms;

                assert ((s_rst = '0') and (s_t = '1') and (s_q = '0') and (s_q_bar = '
                report "Test failed when RESET is '0', CLK is RISING and T is '1'" sev

                wait for 12 ms;
                s_t  <= '0';
                wait for 7 ms;
                s_t  <= '1';
                wait for 7 ms;
                s_t  <= '0';
                wait for 7 ms;
                s_t  <= '1';

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

###Screenshot



##Exercise 4