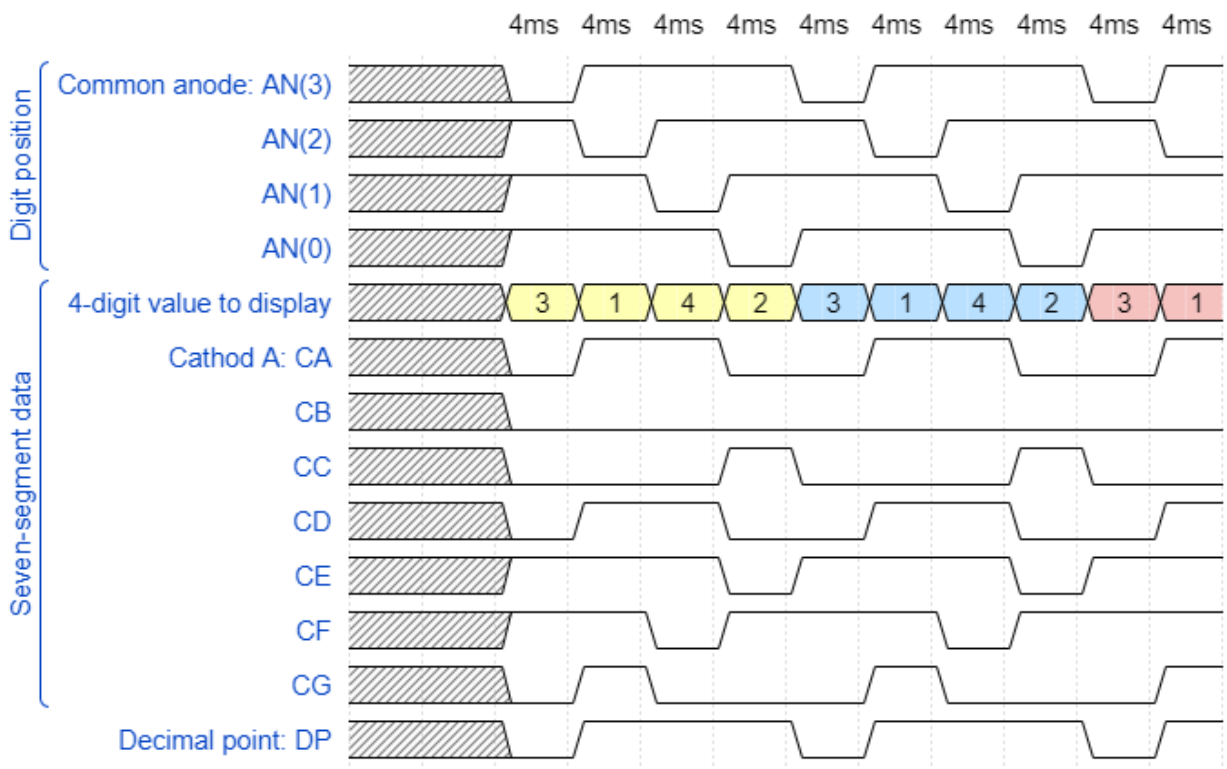
Lab assignment

1. Preparation tasks (done before the lab at home). Submit:
    - Timing diagram figure for displaying value `3.142`.
2. Display driver. Submit:
    - Listing of VHDL code of the process `p_mux` with syntax highlighting.
    - Listing of VHDL testbench file `tb_driver_7seg_4digits` with syntax highlighting and asserts,
    - Screenshot with simulated time waveforms; always display all inputs and outputs,
    - Listing of VHDL architecture of the top layer.
3. Eight-digit driver. Submit:
    - Image of the driver schematic. The image can be drawn on a computer or by hand.

1. Preparation tasks

Timing diagram figure for displaying value `3.142`



2. Display driver

Listing of VHDL code of the process `p_mux` with syntax highlighting

```
p_mux : process(s_cnt, data0_i, data1_i, data2_i, data3_i, dp_i)
    begin
        case s_cnt is
            when "11" =>
                s_hex <= data3_i;
                dp_o  <= dp_i(3);
                dig_o <= "0111";

            when "10" =>
                s_hex <= data2_i;
                dp_o  <= dp_i(2);
                dig_o <= "1011";

            when "01" =>
                s_hex <= data1_i;
                dp_o  <= dp_i(1);
                dig_o <= "1101";

            when others =>
                s_hex <= data0_i;
                dp_o  <= dp_i(0);
                dig_o <= "1110";
```

```
        end case;
    end process p_mux;
```

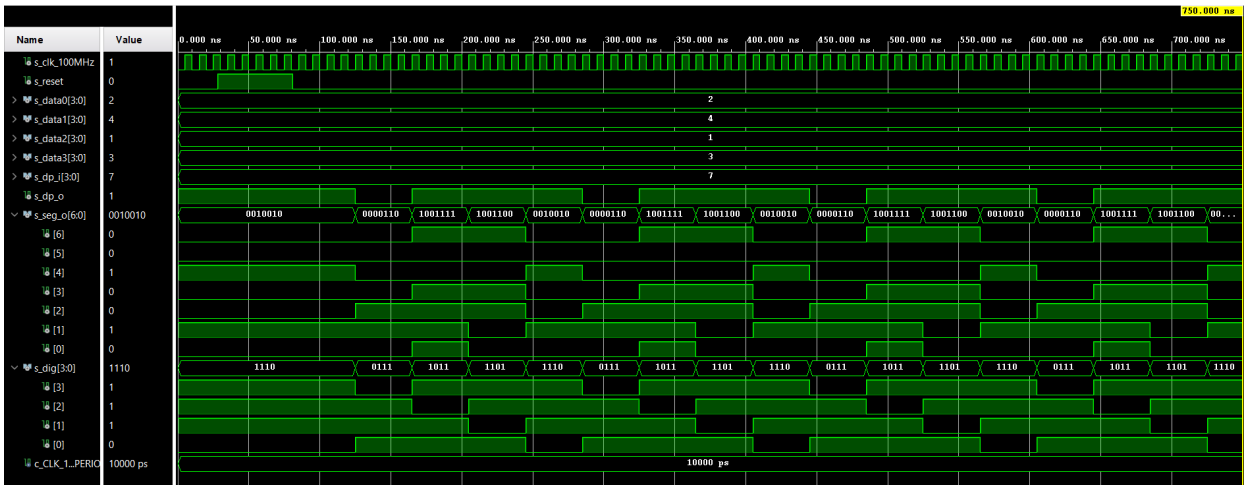
Listing of VHDL testbench file `tb_driver_7seg_4digits` with syntax highlighting and asserts

```
library ieee;
use ieee.std_logic_1164.all;

entity tb_driver_7seg_4digits is
end entity tb_driver_7seg_4digits;

architecture testbench of tb_driver_7seg_4digits is

    constant c_CLK_100MHZ_PERIOD : time    := 10 ns;

    signal s_clk_100MHz : std_logic;

    signal s_reset      : std_logic;

    signal s_data0      : std_logic_vector(4 - 1 downto 0);
    signal s_data1      : std_logic_vector(4 - 1 downto 0);
    signal s_data2      : std_logic_vector(4 - 1 downto 0);
    signal s_data3      : std_logic_vector(4 - 1 downto 0);

    signal s_dp_i       : std_logic_vector(4 - 1 downto 0);
    signal s_dp_o       : std_logic;
    signal s_seg_o      : std_logic_vector(7 - 1 downto 0);

    signal s_dig        : std_logic_vector(4 - 1 downto 0);

begin

    uut_driver_7seg_4digits : entity work.driver_7seg_4digits
        port map(
            clk => s_clk_100MHz,
            reset => s_reset,

            data0_i => s_data0,
            data1_i => s_data1,
            data2_i => s_data2,
            data3_i => s_data3,

            dp_i => s_dp_i,

            dp_o  => s_dp_o,
            seg_o => s_seg_o,
            dig_o => s_dig
        );
    ----------------------------------------------------------------------
    -- Clock generation process
    ----------------------------------------------------------------------
```

```
        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 generation process
        --------------------------------------------------------------------
        p_reset_gen : process
        begin
            s_reset <= '0';
            wait for 28 ns;

            s_reset <= '1';
            wait for 53 ns;

            s_reset <= '0';
            wait;
        end process p_reset_gen;
        --------------------------------------------------------------------
        -- Data generation process
        --------------------------------------------------------------------
        p_stimulus : process
        begin
            report "Stimulus process started" severity note;

            report "Test1" severity note;
            s_data3 <= "0011";
            wait for 10 ns;
            assert (s_seg_o = "0000110")
            report "Test failed" severity error;

            report "Test2" severity note;
            s_data2 <= "0001";
            wait for 10 ns;
            assert (s_seg_o = "1001111")
            report "Test failed" severity error;

            report "Test3" severity note;
            s_data1 <= "0100";
            wait for 10 ns;
            assert (s_seg_o = "1001100")
            report "Test failed" severity error;

            report "Test4" severity note;
            s_data0 <= "0010";
            wait for 10 ns;
            assert (s_seg_o = "0010010")
            report "Test failed" severity error;
```

```
        s_dp_i <= "0111";

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

Screenshot with simulated time waveforms



Listing of VHDL architecture of the top layer.

```
architecture Behavioral of top is
begin
    driver_seg_4 : entity work.driver_7seg_4digits
        port map(
            clk         => CLK100MHZ,
            reset       => BTNC,
            data0_i(3) => SW(3),
            data0_i(2) => SW(2),
            data0_i(1) => SW(1),
            data0_i(0) => SW(0),

            data1_i(3) => SW(7),
            data1_i(2) => SW(6),
            data1_i(1) => SW(5),
            data1_i(0) => SW(4),

            data2_i(3) => SW(11),
            data2_i(2) => SW(10),
            data2_i(1) => SW(9),
            data2_i(0) => SW(8),

            data3_i(3) => SW(15),
            data3_i(2) => SW(14),
            data3_i(1) => SW(13),
            data3_i(0) => SW(12),

            dp_i => "0111",
```

```
            dp_o => DP,

            seg_o(6) => CA,
            seg_o(5) => CB,
            seg_o(4) => CC,
            seg_o(3) => CD,
            seg_o(2) => CE,
            seg_o(1) => CF,
            seg_o(0) => CG,

            dig_o => AN(4 - 1 downto 0)
        );
    AN(7 downto 4) <= b"1111";

end architecture Behavioral;
```

3. Eight-digit driver

Image of the driver schematic