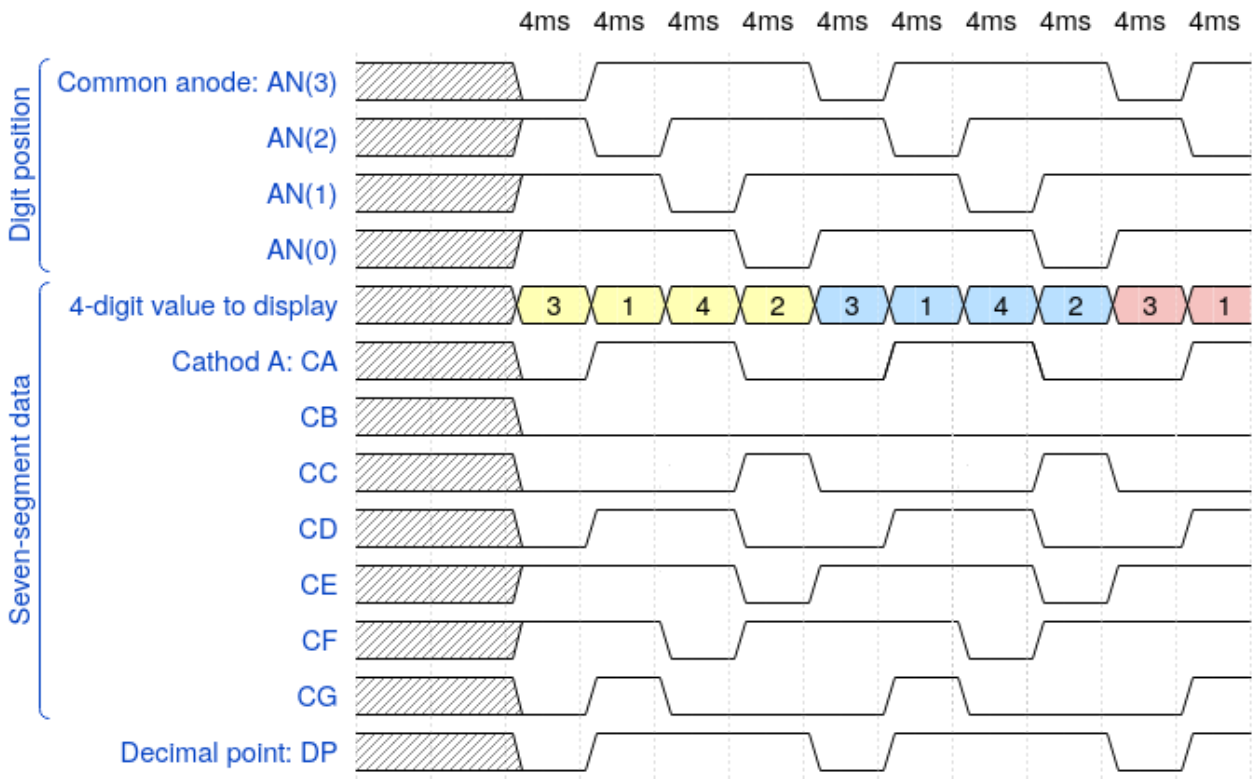
xmanas07 ...	3 minutes ago
..	
display_driver	15 minutes ago
images	15 minutes ago
README.md	3 minutes ago

README.md

Digital-electronics-1

úkol 1: Preparation tasks

Timing diagram figure for displaying value 3.142



úkol 2: Display driver

VHDL p_mux process (`driver_7seg_4digits`)

```
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" =>
                -- WRITE YOUR CODE HERE
                s_hex <= data2_i;
                dp_o  <= dp_i(2);
                dig_o <= "1011";

            when "01" =>
                -- WRITE YOUR CODE HERE
                s_hex <= data1_i;
                dp_o  <= dp_i(1);
                dig_o <= "1101";

            when others =>
                -- WRITE YOUR CODE HERE
                s_hex <= data0_i;
                dp_o  <= dp_i(0);
                dig_o <= "1110";
```

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

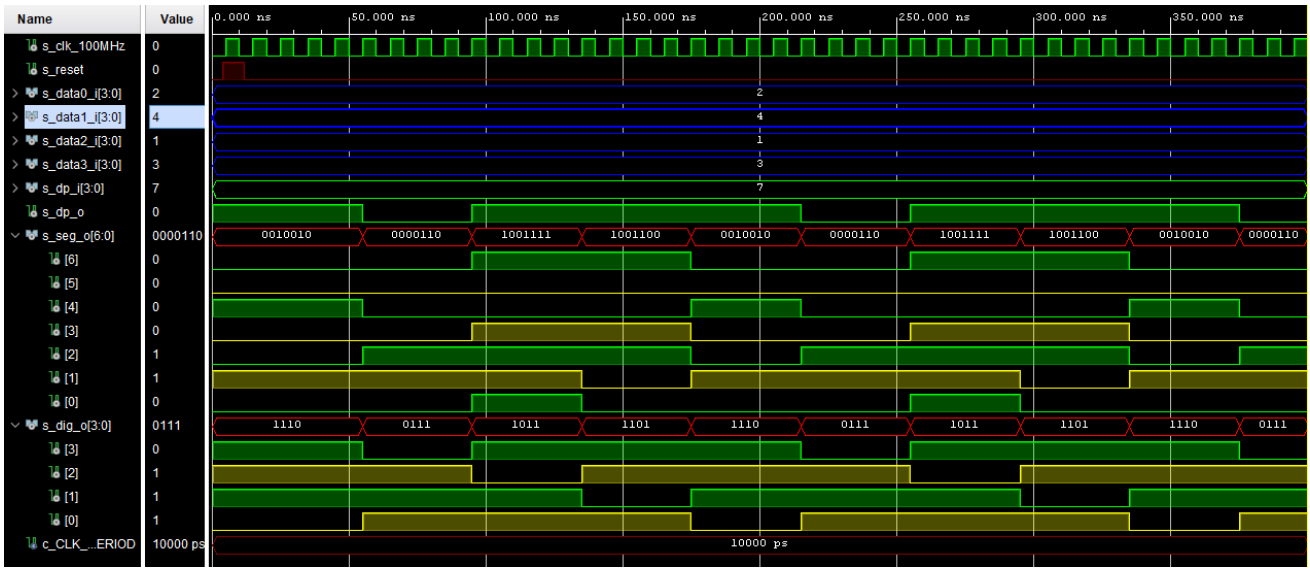
VHDL simulation source (`tb_driver_7seg_4digits`)

```
   ----------------------------------------------------------------------
--
-- Template for 4-digit 7-segment display driver testbench.
-- Nexys A7-50T, Vivado v2020.1.1, EDA Playground
--
-- Copyright (c) 2020-Present Tomas Fryza
-- Dept. of Radio Electronics, Brno University of Technology, Czechia
-- This work is licensed under the terms of the MIT license.
--
----------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;

----------------------------------------------------------------------
-- Entity declaration for testbench
----------------------------------------------------------------------
entity tb_driver_7seg_4digits is
    -- Entity of testbench is always empty
end entity tb_driver_7seg_4digits;

----------------------------------------------------------------------
-- Architecture body for testbench
----------------------------------------------------------------------
architecture testbench of tb_driver_7seg_4digits is

    -- Local constants
    constant c_CLK_100MHZ_PERIOD : time    := 10 ns;

    --Local signals
    signal s_clk_100MHz : std_logic;
    --- WRITE YOUR CODE HERE
    signal s_reset      : std_logic;
    signal s_data0_i    : std_logic_vector(4 - 1 downto 0);
    signal s_data1_i    : std_logic_vector(4 - 1 downto 0);
    signal s_data2_i    : std_logic_vector(4 - 1 downto 0);
    signal s_data3_i    : std_logic_vector(4 - 1 downto 0);
    signal s_dp_i       : std_logic_vector(4 - 1 downto 0);
    signal s_dig_o      : std_logic_vector(4 - 1 downto 0);
    signal s_seg_o      : std_logic_vector(7 - 1 downto 0);
    signal s_dp_o       : std_logic;

begin
    -- Connecting testbench signals with driver_7seg_4digits entity
    -- (Unit Under Test)
    uut_ce : entity work.driver_7seg_4digits
```

```
    port map(
        clk     => s_clk_100MHz,
        reset   => s_reset,
        data0_i => s_data0_i,
        data1_i => s_data1_i,
        data2_i => s_data2_i,
        data3_i => s_data3_i,
        dp_i    => s_dp_i,
        seg_o   => s_seg_o,
        dig_o   => s_dig_o,
        dp_o    => s_dp_o

    );

-------------------------------------------------------------------
-- Clock generation process
-------------------------------------------------------------------
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;

-------------------------------------------------------------------
-- Reset generation process
-------------------------------------------------------------------
--- WRITE YOUR CODE HERE
p_reset_gen : process
begin
    s_reset <= '0';
    wait for 4 ns;

    -- Reset activated
    s_reset <= '1';
    wait for 8 ns;

    -- Reset deactivated
    s_reset <= '0';

    wait;
end process p_reset_gen;
-------------------------------------------------------------------
-- Data generation process
-------------------------------------------------------------------
--- WRITE YOUR CODE HERE
p_stimulus : process
begin
    report "Stimulus process started" severity note;
```

```
        s_data3_i <= "0011"; s_data2_i <= "0001"; s_data1_i <= "0100"; s_data0_i <= "


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


end architecture testbench;
```

Screenshot with waveforms



VHDL source architecture (`top.vhd`)

```
architecture Behavioral of top is
    -- No internal signals
begin

    ------------------------------------------------------------------
    -- Instance (copy) of driver_7seg_4digits entity
    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),
            --- WRITE YOUR CODE HERE
            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)
            --- WRITE YOUR CODE HERE

        );

    -- Disconnect the top four digits of the 7-segment display
    AN(7 downto 4) <= b"1111";

end architecture Behavioral;
```

úkol 3: Image of the eight-digit display driver