Assignment 6

1. Preparation tasks

Timing diagram figure for displaying value '3.142'

2. Display driver

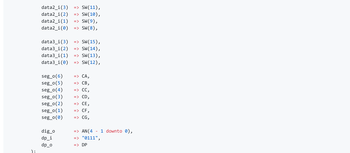
Listing of VHDL code of the process p_mux

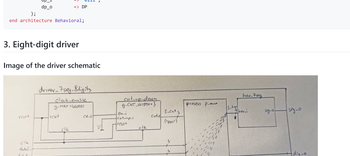
```
p_mux : process(s_cnt, dat0_i, dat1_i, dat2_i, dat3_i, dp_i)
    begin
        case s_cnt is
            when "11" =>
                s_hex <= dat3_i;
                dp_o  <= dp_i(3);
                dig_o <= "0111";
            when "10" =>
                s_hex <= dat2_i;
                dp_o  <= dp_i(2);
                dig_o <= "1011";
            when "01" =>
                s_hex <= dat1_i;
                dp_o  <= dp_i(1);
                dig_o <= "1101";
            when others =>
                s_hex <= dat0_i;
                dp_o  <= dp_i(0);
                dig_o <= "1110";
        end case;
    end process p_mux;
```

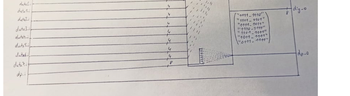
Listing of VHDL testbench file tb_driver_7seg_4digits

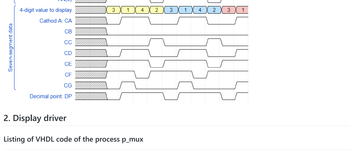
```
----------------------------------------------------------------------------------
-- Template for 4-digit 7-segment display driver testbench.
-- Nexys A7-50T, Vivado v2020.1.1, EDA Playground

-- Copyright (c) 2020 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;
    signal s_reset : std_logic;
    signal s_dat0 : std_logic_vector(4 - 1 downto 0);
    signal s_dat1 : std_logic_vector(4 - 1 downto 0);
    signal s_dat2 : std_logic_vector(4 - 1 downto 0);
    signal s_dat3 : 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  : std_logic_vector(7 - 1 downto 0);
    signal s_dig  : std_logic_vector(4 - 1 downto 0);


begin
    -- Connecting testbench signals with driver_7seg_4digits entity
    -- (Unit Under Test)
    uut_driver_7seg_4digits : entity work.driver_7seg_4digits
    port map (
        clk     => s_clk_100MHz,
        reset   => s_reset,
        dat0_i => s_dat0,
        dat1_i => s_dat1,
        dat2_i => s_dat2,
        dat3_i => s_dat3,
        dp_i    => s_dp_i,
        dp_o    => s_dp_o,
        seg_o => s_seg,
        dig_o => s_dig
    );
    ----------------------------------------------------------------------------------
    -- 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
    ----------------------------------------------------------------------------------
    p_reset_gen : process
    begin
        s_reset <= '0';
        wait for 10 ns;
        -- Reset activated
        s_reset <= '1';
        wait for 53 ns;
        -- Reset deactivated
        s_reset <= '0';
        wait;
    end process p_reset_gen;
    ----------------------------------------------------------------------------------
    -- Data generation process
    ----------------------------------------------------------------------------------
  p_stimulus : process
    begin
        report "Stimulus process started" severity note;
    ----------------------------------------------------------------------------------
        -- 3.142
    ----------------------------------------------------------------------------------
        s_dat3 <= "0011";
        s_dat2 <= "0001";
        s_dat1 <= "0100";
        s_dat0 <= "0010";
        s_dp_i <= "0111";

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

end architecture testbench;
```

Screenshots with simulated time waveforms









Listing of VHDL architecture of the top layer

```
entity top is
    Port ( CLK100MHZ : in std_logic; --Main clock
           BTNC : in std_logic; --Synchronous reset
           SW : in std_logic_vector(16 - 1 downto 0); --Four 4-bit values
           CA : out std_logic; --Cathod A
           CB : out std_logic; --Cathod B
           CC : out std_logic; --Cathod C
           CD : out std_logic; --Cathod D
           CE : out std_logic; --Cathod E
           CF : out std_logic; --Cathod F
           CG : out std_logic; --Cathod G
           DP : out std_logic; --Decimal point
           AN : out std_logic_vector(8 - 1 downto 0)); --Common anode signals to individual displays
end top;

architecture Behavioral of top is
begin
    ----------------------------------------------------------------------------------
    -- Instance (copy) of driver_7seg_4digits entity
    driver_seg_4 : entity work.driver_7seg_4digits
        port map
        (
            clk     => CLK100MHZ,
            reset   => BTNC,
            dat0_i(3) => SW(3),
            dat0_i(2) => SW(2),
            dat0_i(1) => SW(1),
            dat0_i(0) => SW(0),

            dat1_i(3) => SW(7),
            dat1_i(2) => SW(6),
            dat1_i(1) => SW(5),
            dat1_i(0) => SW(4),

            dat2_i(3) => SW(11),
            dat2_i(2) => SW(10),
            dat2_i(1) => SW(9),
            dat2_i(0) => SW(8),

            dat3_i(3) => SW(15),
            dat3_i(2) => SW(14),
            dat3_i(1) => SW(13),
            dat3_i(0) => SW(12),

            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),
            dp_i    => "0111",
            dp_o    => DP
        );
end architecture Behavioral;
```

3. Eight-digit driver

Image of the driver schematic

4. Eight-digit driver