



# Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

[Read the guide](#)

[xrysav25](#) / [Digital-electronics-1](#)

MIT License

0 stars 0 forks

Star

Unwatch ▼

[Code](#)

[Issues](#)

[Pull requests](#)

[Actions](#)

[Projects](#)

[Wiki](#)

[Security](#)

[Insights](#)

[Settings](#)

main ▼



[Digital-electronics-1](#) / [06-display\\_driver](#) / [README.md](#)

[xrysav25](#) dd 06 lab and 05 readme.pdf



0 contributors

[Raw](#)

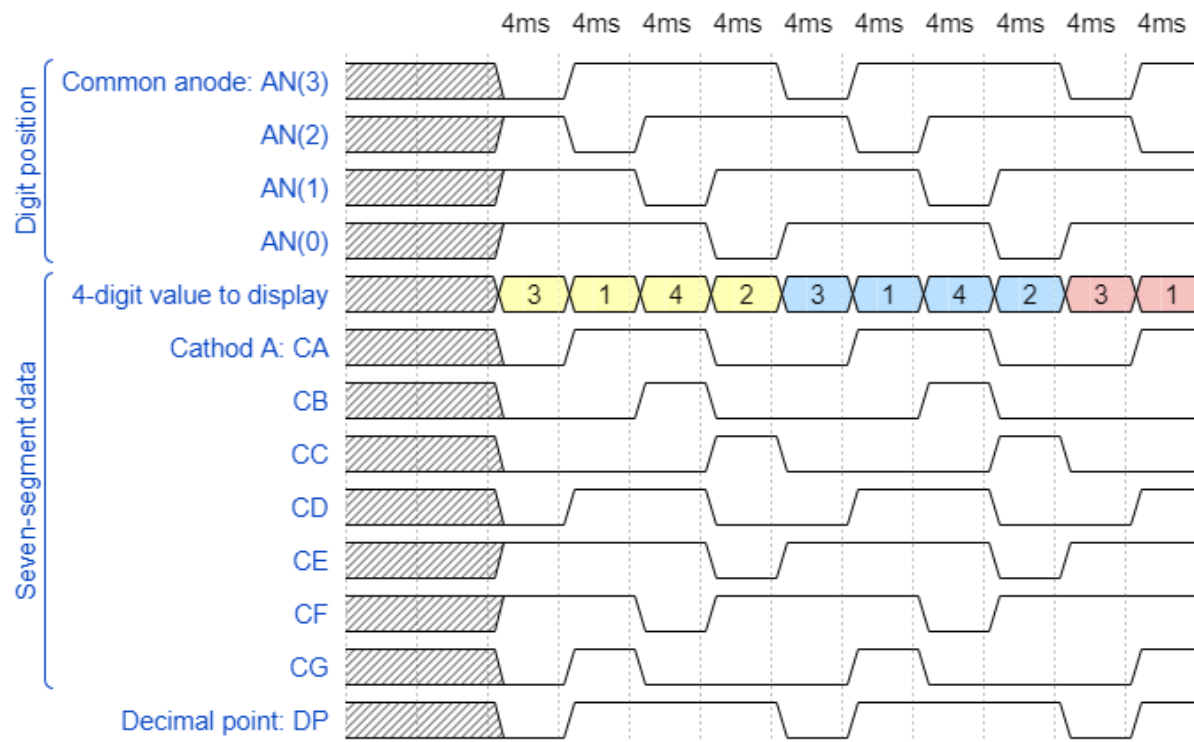
[Blame](#)



172 lines (157 sloc) | 5.28 KB

## LAB 6

# 1. Preparation task



## 2. Display driver

- proces mux

```
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";-- WRITE YOUR CODE HERE

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

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

- 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 : std_logic_vector(3 downto 0);
    signal s_data1 : std_logic_vector(3 downto 0);
    signal s_data2 : std_logic_vector(3 downto 0);
    signal s_data3 : std_logic_vector(3 downto 0);
    signal s_dp_i  : std_logic_vector(3 downto 0);
    signal s_dp_o  : std_logic;
    signal s_seg   : std_logic_vector(6 downto 0);
    signal s_dig   : std_logic_vector(3 downto 0);

begin
    -- Connecting testbench signals with driver_7seg_4digits entity
    -- (Unit Under Test)
    --- WRITE YOUR CODE HERE
    uut_driver_7seg_4digits: entity work.driver_7seg_4digits
    -----

    -- Clock generation process
    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,
        dig_o    => s_dig
    );
    -----

    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

```

```

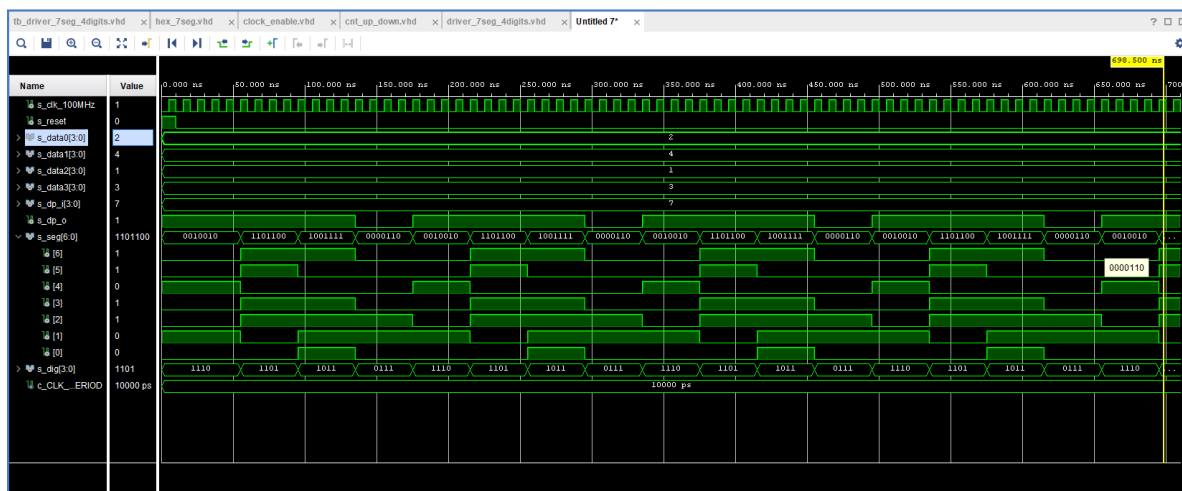
-----
--- WRITE YOUR CODE HERE
p_reset: process
begin
    s_reset <= '1';
    wait for 10ns;
    s_reset <= '0';
    wait;
end process p_reset;
-----

-- Data generation process
-----

--- WRITE YOUR CODE HERE
p_generation: process
begin
    s_data3 <= "0011";
    s_data2 <= "0001";
    s_data1 <= "0100";
    s_data0 <= "0010";
    s_dp_i  <= "0111";
    wait for 15ns;
    assert (s_seg = "0000110") and (s_dig = "0111") and (s_dp_o = '0') report
    wait for 10ns;
    assert (s_seg = "1001111") and (s_dig = "1011") and (s_dp_o = '1') report
    wait for 10ns;
    assert (s_seg = "1101100") and (s_dig = "1101") and (s_dp_o = '1') report
    wait for 10ns;
    assert (s_seg = "0010010") and (s_dig = "1110") and (s_dp_o = '1') report
    wait;
end process p_generation;
end architecture testbench;

```

- screenshot



- top

```

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
    dp_i => "0111",
    --- 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),

    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,
    dp_o         => DP,
    dig_o        => AN(3 downto 0)

  );

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

end architecture Behavioral;

```

### 3. eight-digit driver

---

