Bobik77 / **Digital-electronics-1**

Code Issues Pull requests Actions Projects Security Insights

main

Digital-electronics-1 / LAB_03 / **README.MD**

Bobik77 Update README.MD **History**

1 contributor

Raw Blame

`154 lines (151 sloc) | 4.55 KB`

BPC-DE1 Lab_03

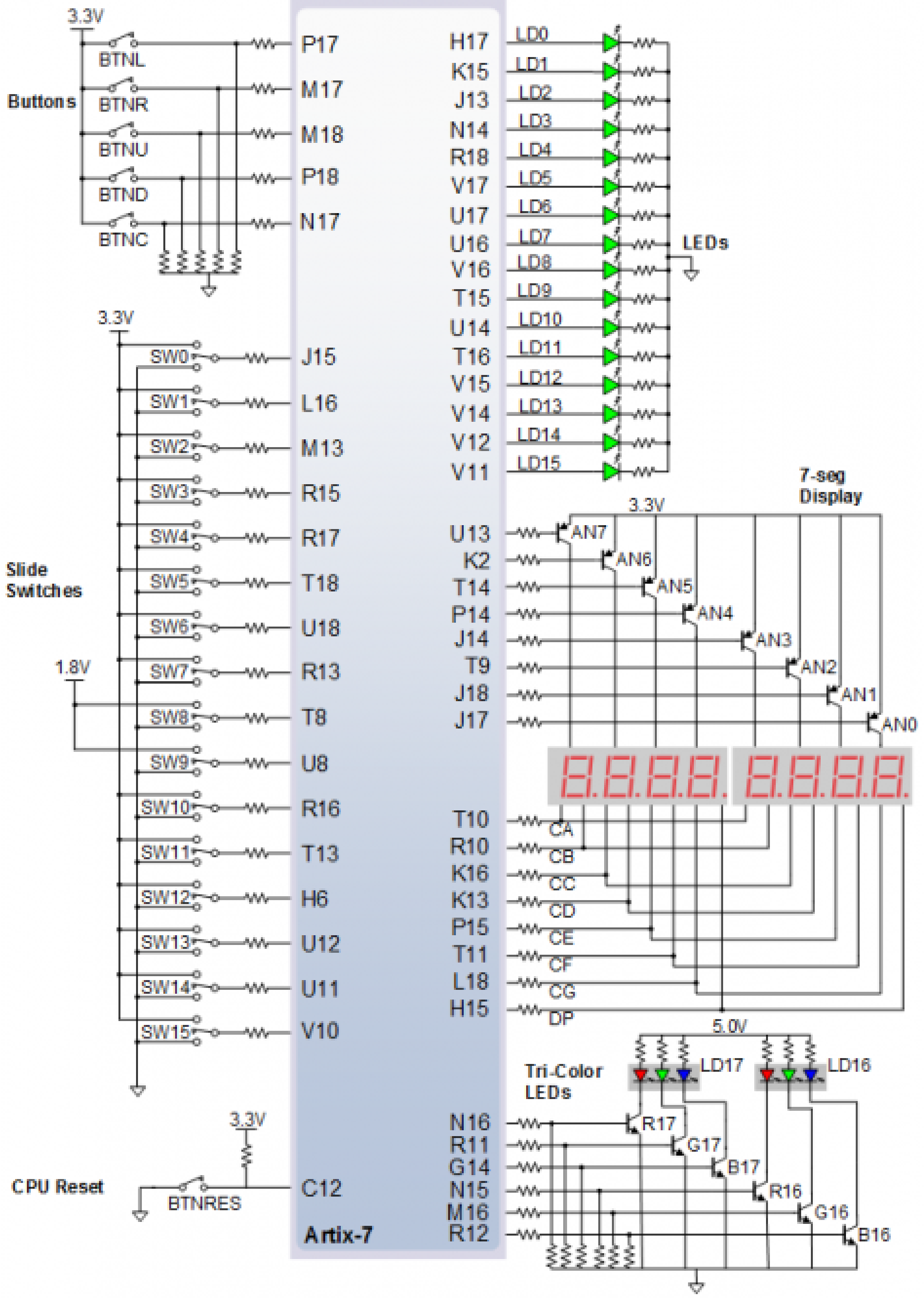
📎 Repository: Bobik77 / Digital-electronic-1 / LAB3

Preparation tasks:

Download and install Vivado

Done ✔️

SW and LEDs connect table:



SOURCE

Part	PIN
LED0	H17
LED1	K15
LED2	J13
LED3	N14
LED4	R18
LED5	V17

Part	PIN
LED6	U17
LED7	U16
LED8	V16
LED9	T15
LED10	U14
LED11	T16
LED12	V15
LED13	V14
LED14	V12
LED15	V11
SW0	J15
SW1	L16
SW2	M13
SW3	R15
SW4	R17
SW5	T18
SW6	U18
SW7	R13
SW8	T8
SW9	U8
SW10	R16
SW11	T13
SW12	H6
SW13	U12
SW14	U11
SW15	V10

Two-bit wide 4-to-1 multiplexer

Listing architecture

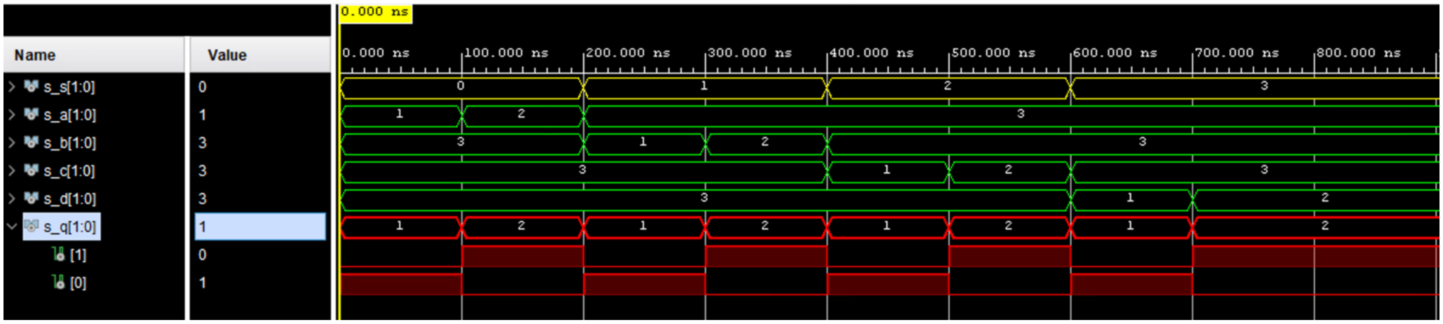
```
architecture Behavioral of mux_2bit_4to1 is
begin

    process (a_i, b_i, c_i, d_i, s_i)
    begin
        case s_i is
            when "00" => q_o <= a_i;
            when "01" => q_o <= b_i;
            when "10" => q_o <= c_i;
            when "11" => q_o <= d_i;
            when others => null;
        end case;
    end process;

end Behavioral;
```

Listing testbench architecture

```
architecture testbench of tb_mux_2bit_4to1 is
    signal s_a      : std_logic_vector(1 downto 0);
    signal s_b      : std_logic_vector(1 downto 0);
    signal s_c      : std_logic_vector(1 downto 0);
    signal s_d      : std_logic_vector(1 downto 0);
    signal s_s      : std_logic_vector(1 downto 0);
    signal s_q      : std_logic_vector(1 downto 0);
begin
    --connecting testbench
    uut_mux_2bit_4to1   : entity work.mux_2bit_4to1
        port map(
            a_i     => s_a,
            b_i     => s_b,
            c_i     => s_c,
            d_i     => s_d,
            s_i     => s_s,
            q_o     => s_q
        );
     --Test signal generating
     p_stimuls : process
     begin
     --init. signal
        s_a <= "11";
        s_b <= "11";
        s_c <= "11";
        s_d <= "11";
     --1.st combination
```

```
        s_s <= "00";
            s_a <= "01";
            wait for 100 ns;
            s_a <= "10";
            wait for 100 ns;
            s_a <= "11";
     --2.nd combination
        s_s <= "01";
            s_b <= "01";
            wait for 100 ns;
            s_b <= "10";
            wait for 100 ns;
            s_b <= "11";
     --3.rd combination
        s_s <= "10";
            s_c <= "01";
            wait for 100 ns;
            s_c <= "10";
            wait for 100 ns;
            s_c <= "11";
     --4.th combination
        s_s <= "11";
            s_d <= "01";
            wait for 100 ns;
            s_d <= "10";
            wait for 100 ns;
        wait;
     end process p_stimuls;
end architecture testbench;
```

Screenshot



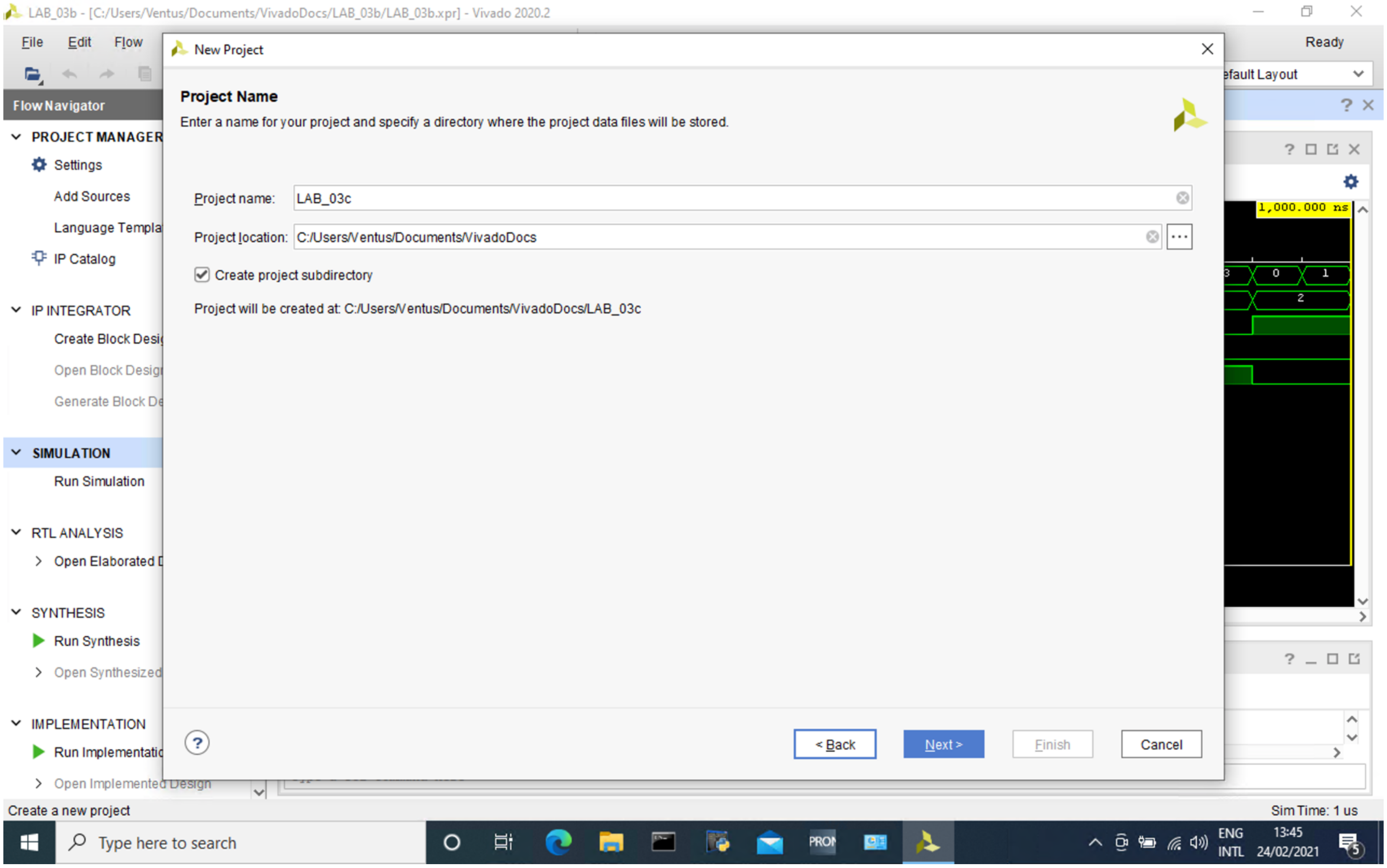
Name	Value
s_s[1:0]	0
s_a[1:0]	1
s_b[1:0]	3
s_c[1:0]	3
s_d[1:0]	3
s_q[1:0]	1
[1]	0
[0]	1

Vstupní signály byly voleny tak, aby funkčnost multiplexoru byla zjevná.

Vivado tutorial

Project creation:

*Zadej cestu projektu a vyper RTl jako typ projektu


LAB_03b - [C:/Users/Ventus/Documents/VivadoDocs/LAB_03b/LAB_03b.xpr] - Vivado 2020.2

New Project

Project Name

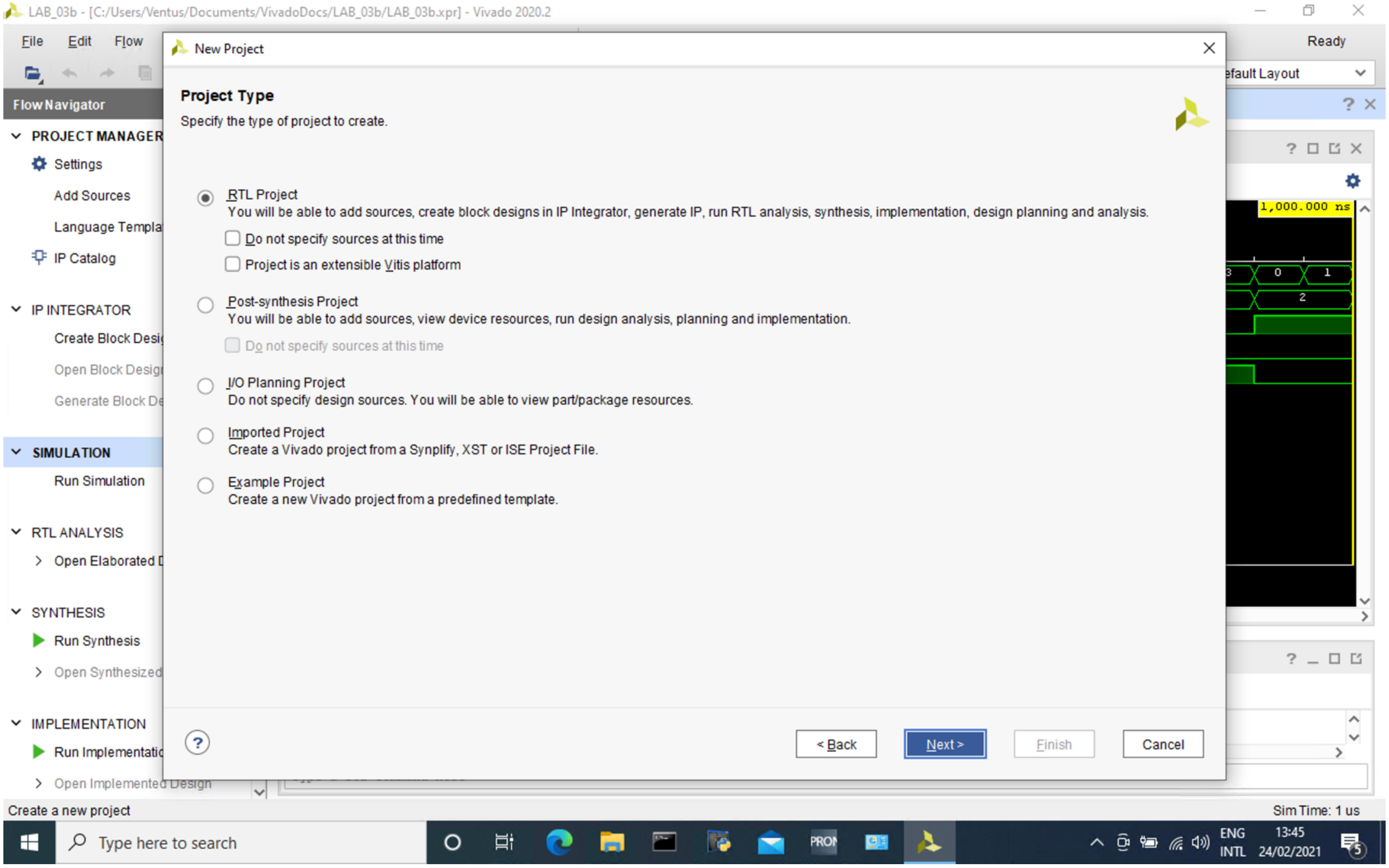
Enter a name for your project and specify a directory where the project data files will be stored.

Project name: LAB_03c

Project location: C:/Users/Ventus/Documents/VivadoDocs

☑ Create project subdirectory

Project will be created at: C:/Users/Ventus/Documents/VivadoDocs/LAB_03c


LAB_03b - [C:/Users/Ventus/Documents/VivadoDocs/LAB_03b/LAB_03b.xpr] - Vivado 2020.2

New Project

Project Type

Specify the type of project to create.

● RTL Project
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.

□ Do not specify sources at this time

□ Project is an extensible Vitis platform

○ Post-synthesis Project
You will be able to add sources, view device resources, run design analysis, planning and implementation.

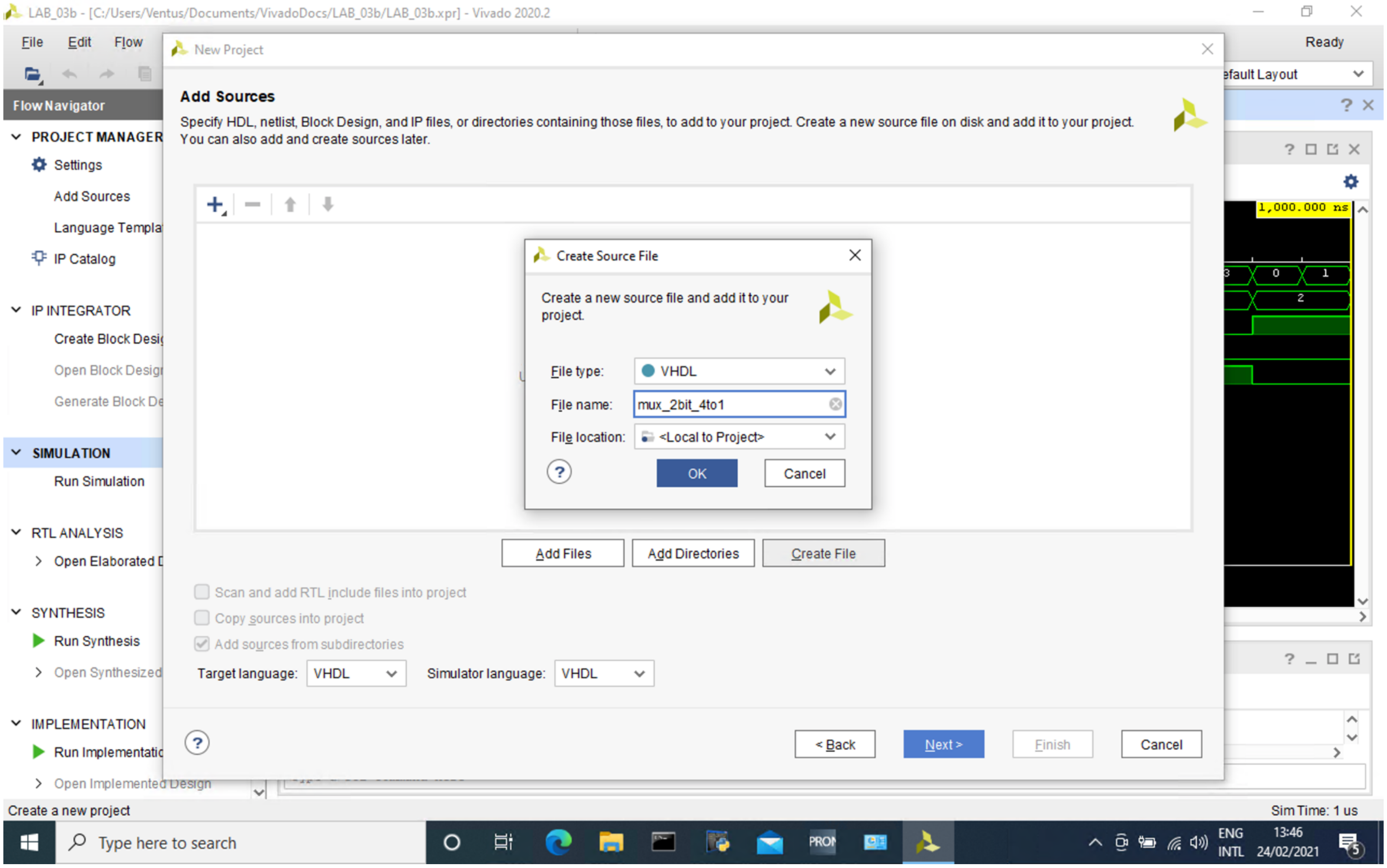
□ Do not specify sources at this time

○ I/O Planning Project
Do not specify design sources. You will be able to view part/package resources.

○ Imported Project
Create a Vivado project from a Synplify, XST or ISE Project File.

○ Example Project
Create a new Vivado project from a predefined template.

*Vytvoř soubor (*Create file*) a pojmenuj ho. Tím se vytvoří *name*.vhd soubor kódu.



LAB_03b - [C:/Users/Ventus/Documents/VivadoDocs/LAB_03b/LAB_03b.xpr] - Vivado 2020.2

Add Sources

Specify HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project. Create a new source file on disk and add it to your project. You can also add and create sources later.

Create Source File

Create a new source file and add it to your project.

File type: VHDL

File name: mux_2bit_4to1

File location: <Local to Project>

☐ Scan and add RTL include files into project

☐ Copy sources into project

☑ Add sources from subdirectories

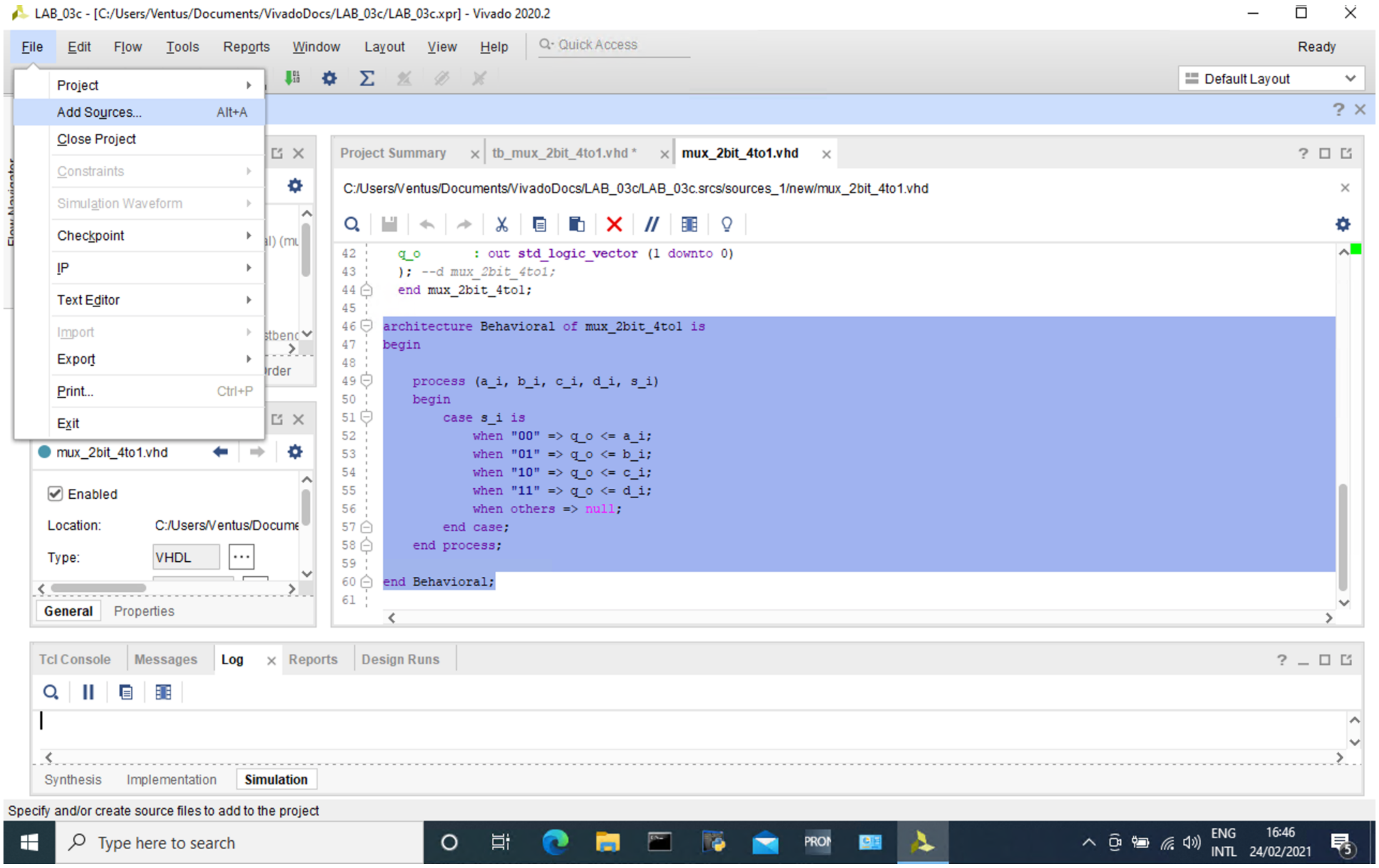
Target language: VHDL

Simulator language: VHDL

Create a new project

Testbench:

- **File > Add Sources > Add or create simulation >Create file** vytvoříme *name*.vhd soubor testbenche. Zpravidla volíme název jako *tb_name.vhd*



C:/Users/Ventus/Documents/VivadoDocs/LAB_03c/LAB_03c.srcs/sources_1/new/mux_2bit_4to1.vhd

```
  q_o       : out std_logic_vector (1 downto 0)
  ); --d mux_2bit_4to1;
  end mux_2bit_4to1;

architecture Behavioral of mux_2bit_4to1 is
begin

   process (a_i, b_i, c_i, d_i, s_i)
   begin
      case s_i is
         when "00" => q_o <= a_i;
         when "01" => q_o <= b_i;
         when "10" => q_o <= c_i;
         when "11" => q_o <= d_i;
         when others => null;
      end case;
   end process;

end Behavioral;
```

Location: C:/Users/Ventus/Docume

Type: VHDL

Specify and/or create source files to add to the project

- Testbench vytvoříš běžným postupem jak ses naučil např. na EDA-Playground.

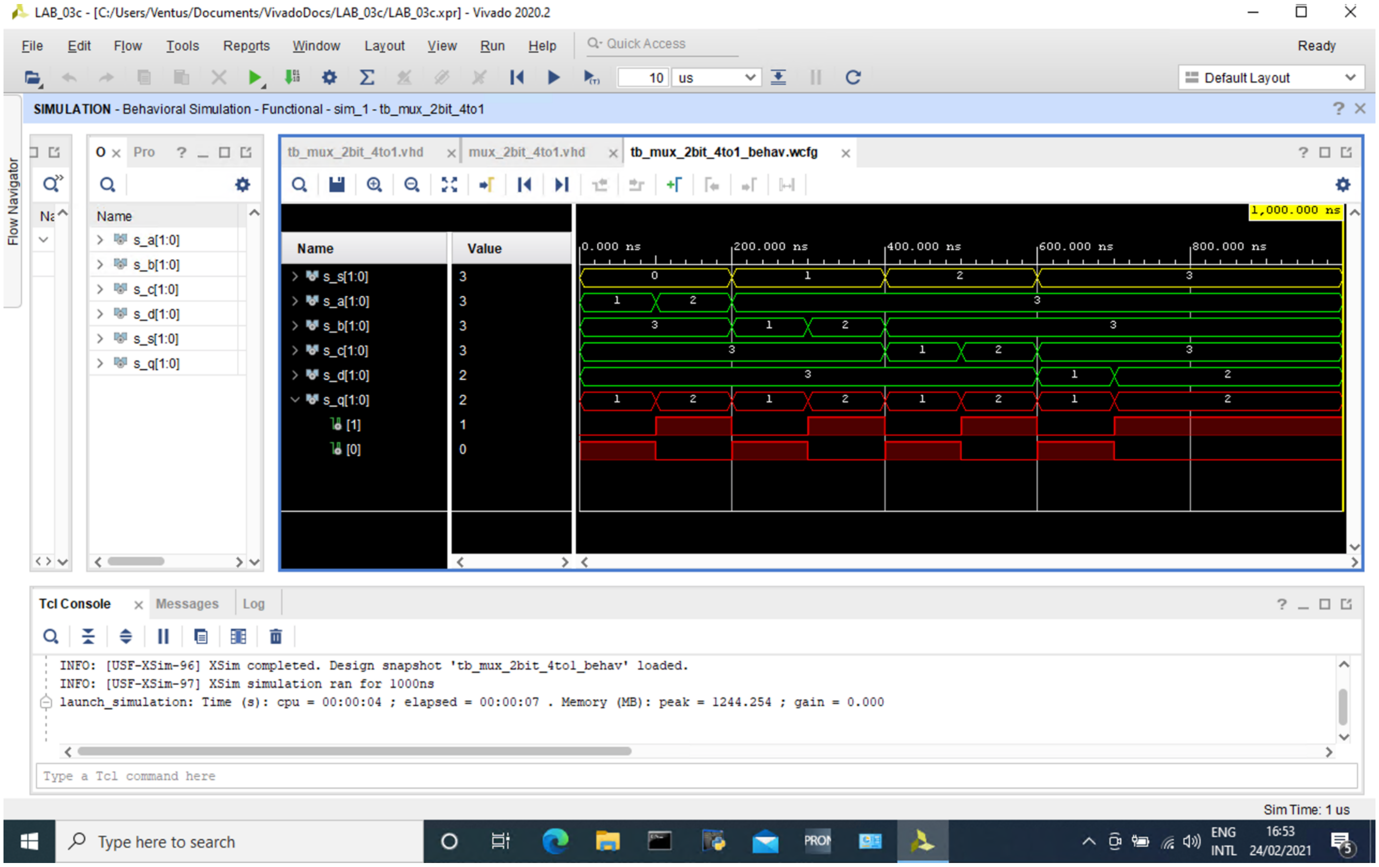
Simulace:

- Zvolením ** Flow > Run Simulation > Run Behavioral Simulation** spustíme simulaci (projekt je prvně třeba uložit).

Postprocesor:

- V postprocesoru vidíme průběhy jednotlivých signálů
- Zvětšíme náhled na průběhy a zazoomujeme podle potřeby
- Jednotlivé sběrnice můžeme rozkliknout na signály, můžeme je libovolně podbarvovat, měřit a vizualizovat

- V nastavení simulace je možné zvětšit délku trvání simulace

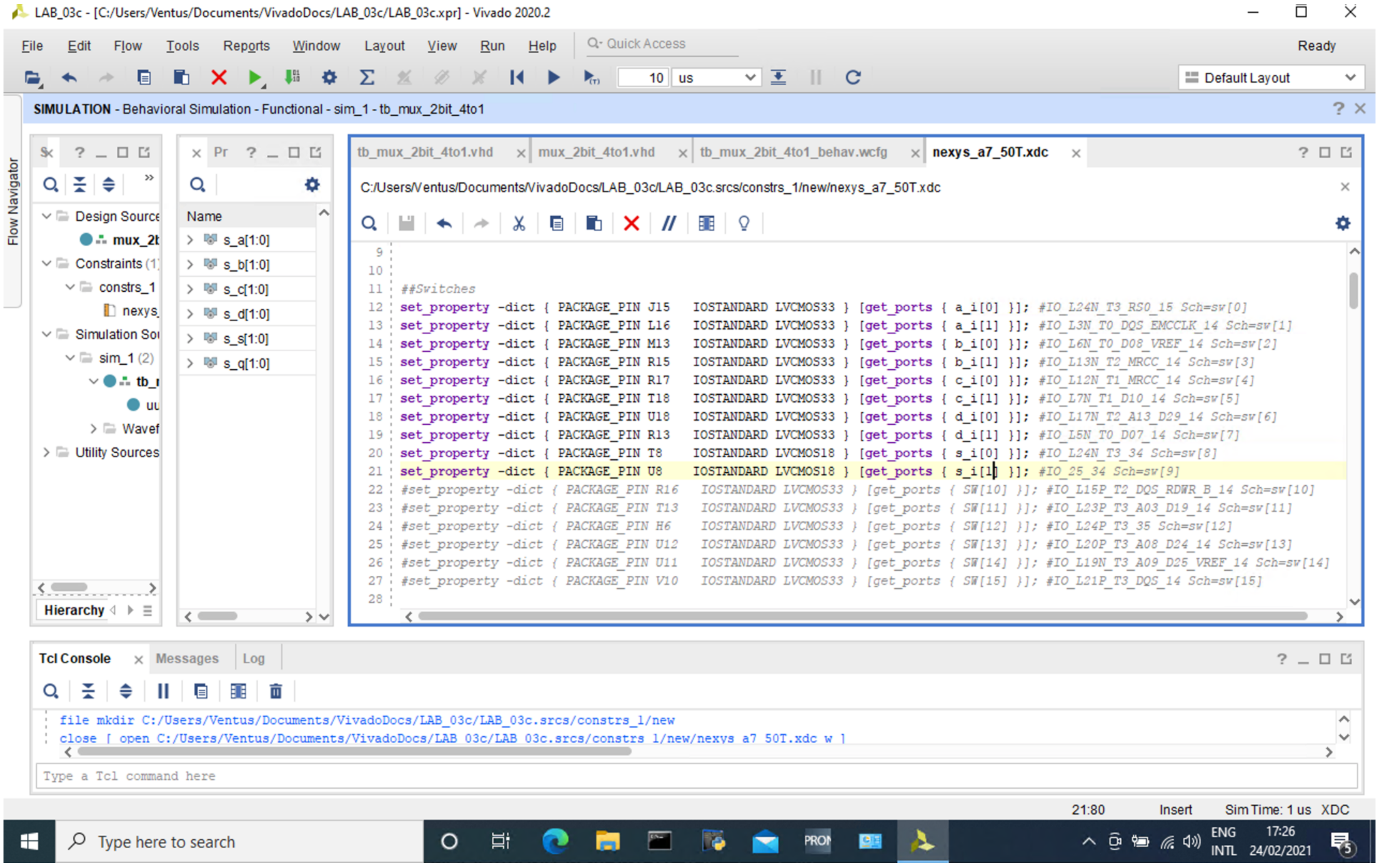


```
INFO: [USF-XSim-96] XSim completed. Design snapshot 'tb_mux_2bit_4to1_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:04 ; elapsed = 00:00:07 . Memory (MB): peak = 1244.254 ; gain = 0.000
```

Mapování vstupů:

- **File > Add Sources > Add or create constrains >Create file** vytvoříme *name*.xdc soubor mapování.
- Do souboru mapování vložím z předdefinovaného souboru od výrobce
- Odkomentuji ty prvky, které hodlám v programu použít

- Přepíši dané porty, tak aby odpovídali již napsané architektuře



C:/Users/Ventus/Documents/VivadoDocs/LAB_03c/LAB_03c.srcs/constrs_1/new/nexys_a7_50T.xdc

```


##Switches
set_property -dict { PACKAGE_PIN J15   IOSTANDARD LVCMOS33 } [get_ports { a_i[0] }]; #IO_L24N_T3_RS0_15 Sch=sw[0]
set_property -dict { PACKAGE_PIN L16   IOSTANDARD LVCMOS33 } [get_ports { a_i[1] }]; #IO_L3N_T0_DQS_EMCCLK_14 Sch=sw[1]
set_property -dict { PACKAGE_PIN M13   IOSTANDARD LVCMOS33 } [get_ports { b_i[0] }]; #IO_L6N_T0_D08_VREF_14 Sch=sw[2]
set_property -dict { PACKAGE_PIN R15   IOSTANDARD LVCMOS33 } [get_ports { b_i[1] }]; #IO_L13N_T2_MRCC_14 Sch=sw[3]
set_property -dict { PACKAGE_PIN R17   IOSTANDARD LVCMOS33 } [get_ports { c_i[0] }]; #IO_L12N_T1_MRCC_14 Sch=sw[4]
set_property -dict { PACKAGE_PIN T18   IOSTANDARD LVCMOS33 } [get_ports { c_i[1] }]; #IO_L7N_T1_D10_14 Sch=sw[5]
set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVCMOS33 } [get_ports { d_i[0] }]; #IO_L17N_T2_A13_D29_14 Sch=sw[6]
set_property -dict { PACKAGE_PIN R13   IOSTANDARD LVCMOS33 } [get_ports { d_i[1] }]; #IO_L5N_T0_D07_14 Sch=sw[7]
set_property -dict { PACKAGE_PIN T8    IOSTANDARD LVCMOS18 } [get_ports { s_i[0] }]; #IO_L24N_T3_34 Sch=sw[8]
set_property -dict { PACKAGE_PIN U8    IOSTANDARD LVCMOS18 } [get_ports { s_i[1] }]; #IO_25_34 Sch=sw[9]
#set_property -dict { PACKAGE_PIN R16   IOSTANDARD LVCMOS33 } [get_ports { SW[10] }]; #IO_L15P_T2_DQS_RDWR_B_14 Sch=sw[10]
#set_property -dict { PACKAGE_PIN T13   IOSTANDARD LVCMOS33 } [get_ports { SW[11] }]; #IO_L23P_T3_A03_D19_14 Sch=sw[11]
#set_property -dict { PACKAGE_PIN H6    IOSTANDARD LVCMOS33 } [get_ports { SW[12] }]; #IO_L24P_T3_35 Sch=sw[12]
#set_property -dict { PACKAGE_PIN U12   IOSTANDARD LVCMOS33 } [get_ports { SW[13] }]; #IO_L20P_T3_A08_D24_14 Sch=sw[13]
#set_property -dict { PACKAGE_PIN U11   IOSTANDARD LVCMOS33 } [get_ports { SW[14] }]; #IO_L19N_T3_A09_D25_VREF_14 Sch=sw[14]
#set_property -dict { PACKAGE_PIN V10   IOSTANDARD LVCMOS33 } [get_ports { SW[15] }]; #IO_L21P_T3_DQS_14 Sch=sw[15]

```

```
file mkdir C:/Users/Ventus/Documents/VivadoDocs/LAB_03c/LAB_03c.srcs/constrs_1/new
close [ open C:/Users/Ventus/Documents/VivadoDocs/LAB_03c/LAB_03c.srcs/constrs_1/new/nexys_a7_50T.xdc w ]
```

Type a Tcl command here