

Image 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	.5 V11		
SW0	J15		
SW1	L16		
SW2	M13		
SW3	R15		
SW4	R17		
SW5	T18		
SW6	U18		
SW7	R13		
SW8	Т8		
SW9	U8		
SW10	R16		
SW11	T13		
SW12	Н6		
SW13	U12		
SW14	U11		
SW15	V10		

⊘ Two-bit wide 4-to-1 multiplexer

∂ Listing architecture

```
architecture Behavioral of mux_2bit_4tol is
begin

process (a_i, b_i, c_i, d_i, s_i)
begin

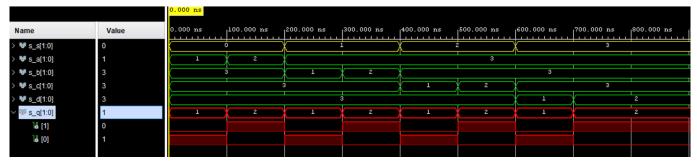
case s_i is
    when "00" => q_0 <= a_i;
    when "01" => q_0 <= b_i;
    when "10" => q_0 <= c_i;
    when "11" => q_0 <= d_i;
    when others => null;
end case;
end process;
end Behavioral;
```

∂ Listing testbench architecture

```
architecture testbench of tb mux 2bit 4tol is
   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_0
                  => 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 = "00";
           s a <= "01";
           wait for 100 ns;
```

```
s a <= "10";
            wait for 100 ns;
            s a <= "11";
     --2.nd combination
        s_s \ll 01;
            s b \le "01";
            wait for 100 ns;
            s b \le "10";
            wait for 100 ns;
            s b <= "11";
     --3.rd combination
        s_s <= "10";
            s c \le "01";
            wait for 100 ns;
            s c \ll "10";
            wait for 100 ns;
            s_c <= "11";
     --4.th combination
        s_s <= "11";
            s_d <= "01";
            wait for 100 ns;
            s_d \ll 10;
            wait for 100 ns;
        wait;
     end process p_stimuls;
end architecture testbench;
```

⊘ Screenshot



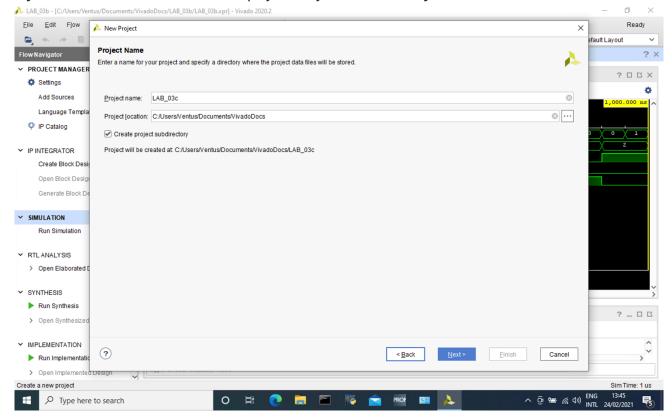
Vstupní signály byly voleny tak, aby funkčnost multiplexoru byla zjevná.

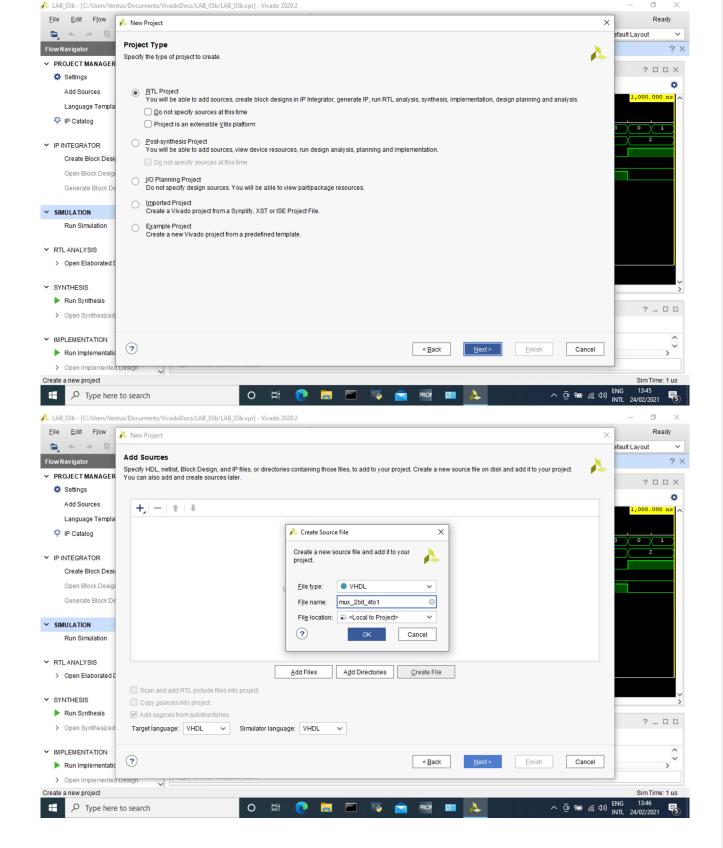
Vivado tutorial

∂ Project creation:

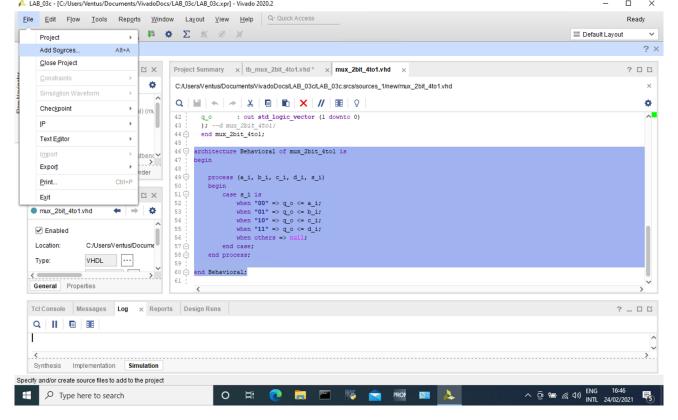
• Zadej cestu projektu a vyper RTI jako typ projektu

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





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



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 LAB_03c - [C:/Users/Ventus/Documents/VivadoDocs/LAB_03c/LAB_03c.xpr] - Vivado 2020.2 File Edit Flow Tools Reports Window Layout View Run Help Q- Quick Access Ready 10 us v I C E Default Layout SIMULATION - Behavioral Simulation - Functional - sim_1 - tb_mux_2bit_4to1 O × Pro ? _ 口 凸 tb_mux_2bit_4to1.vhd × mux_2bit_4to1.vhd × tb_mux_2bit_4to1_behav.wcfg ? 🗆 🖰 ď, Q W Q X W N ± ± + 4 4 H Na^ Name Flow > W s a[1:0] Name Value > 💖 s_b[1:0] ₩ s_s[1:0] > 😻 s_c[1:0] ₩ s_a[1:0] > 💖 s d[1:0] **₩** s_b[1:0] > 💖 s_s[1:0] ₩ s. c(1:0) > 💖 s_q[1:0] **₩** s_d[1:0] **₩** s_q[1:0] To [1] TO1 Tcl Console × Messages Log ? _ 0 0 Q 🛨 | 💠 | II | 🖺 | 🛍 | 🗰 INFO: [USF-XSim-96] XSim completed. Design snapshot 'tb_mux_2bit_4tol_behav' loaded. 🖒 launch_simulation: Time (s): cpu = 00:00:04 ; elapsed = 00:00:07 . Memory (MB): peak = 1244.254 ; gain = 0.000 Type a Tcl command here Sim Time: 1 us

Type here to search

Type here to search

- 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 LAB_03c - [C:/Users/Ventus/Documents/VivadoDocs/LAB_03c/LAB_03c.xpr] - Vivado 2020.2 \Box $\underline{\textbf{File}} \quad \underline{\textbf{E}} \textbf{dit} \quad \textbf{F}\underline{\textbf{Iow}} \quad \underline{\textbf{Iools}} \quad \textbf{Rep}\underline{\textbf{o}} \textbf{rts} \quad \underline{\textbf{W}} \textbf{indow} \quad \textbf{La}\underline{\textbf{y}} \textbf{out} \quad \underline{\textbf{V}} \textbf{iew} \quad \underline{\textbf{R}} \textbf{un} \quad \underline{\textbf{H}} \textbf{elp} \quad \boxed{\textbf{Q}} \cdot \underline{\textbf{Q}} \textbf{uick} \, \textbf{Access}$ Ready 10 us ▼

 ■ □ C E Default Layout SIMULATION - Behavioral Simulation - Functional - sim_1 - tb_mux_2bit_4to1 ? X tb_mux_2bit_4to1.vhd × mux_2bit_4to1.vhd × tb_mux_2bit_4to1_behav.wcfg × nexys_a7_50T.xdc % ? _ □ Ľ × Pr ? _ □ Ľ ? 🗆 🖸 Q 🛬 🛊 Q. C:/Users/Ventus/Documents/VivadoDocs/LAB_03c/LAB_03c.srcs/constrs_1/new/nexys_a7_50T.xdc ∨ Design Source Name ٥ Flow ■ ... mux_2t > 💖 s_a[1:0] ∨
 Constraints (1) > 💖 s_b[1:0] ∨ □ constrs_1 > 🕷 s_c[1:0] IOSTANDARD LVCMOS33 } [get_ports { a_i[0] }]; #IO_L24N_T3_R50_15 Sch=sw[0] IOSTANDARD LVCMOS33 } [get_ports { a_i[1] }]; #IO_L3N_T0_pos_EMCCLK_14 Sch=sw[1] IOSTANDARD LVCMOS33 } [get_ports { b_i[0] }]; #IO_L6N_T0_pos_VREF_14 Sch=sw[2] IOSTANDARD LVCMOS33 } [get_ports { b_i[1] }]; #IO_L13N_T2_MRCC_14 Sch=sw[3] IOSTANDARD LVCMOS33 } [get_ports { c_i[0] }]; #IO_L13N_T2_MRCC_14 Sch=sw[4] IOSTANDARD LVCMOS33 } [get_ports { c_i[1] }]; #IO_L7N_T1_D10_14 Sch=sw[5] IOSTANDARD LVCMOS33 } [get_ports { c_i[1] }]; #IO_L7N_T1_D10_14 Sch=sw[5] IOSTANDARD LVCMOS33 } [get_ports { d_i[0] }]; #IO_L7N_T2_A13_D29_14 Sch=sw[6] IOSTANDARD LVCMOS33 } [get_ports { d_i[0] }]; #IO_L7N_T2_A13_D29_14 Sch=sw[6] IOSTANDARD LVCMOS33 } [get_ports { d_i[0] }]; #IO_L7N_T2_A13_D29_14 Sch=sw[6] set_property -dict { PACKAGE_PIN J15 nexys > W s_d[1:0] set_property -dict { PACKAGE_PIN L16 ∨

 Simulation Soi > 😻 s_s[1:0] set_property -dict { PACKAGE_PIN M13 set_property -dict { PACKAGE_PIN R15 set_property -dict { PACKAGE_PIN R17 > 🕷 s_q[1:0] ∨ **●** ... tb ı set_property -dict { PACKAGE_PIN T18 set_property -dict { PACKAGE_PIN U18 ■ m > 🗎 Wavef set_property -dict { PACKAGE_PIN R13 set_property -dict { PACKAGE_PIN T8 IOSTANDARD LVCMOS33 } [get ports { d_i[1] }]; #IO_LSN_TO_DO7_14 Sch=sw[7] IOSTANDARD LVCMOS18 } [get ports { s_i[0] }]; #IO_L24N_T3_34 Sch=sw[8] > Dtility Sources IOSTANDARD LVCMOS18) [get_ports { si[1] }]; #IO 25 34 Sch=sv[9]
 IOSTANDARD LVCMOS33) [get_ports { SW[10] }]; #IO LISP T2_DQS RDWR B_14 Sch=sv[10] set_property -dict { PACKAGE_PIN U8 #set_property -dict (PACKAGE_PIN R16 IOSTANDARD LVCMOS33) [get ports { SW[11] }]; #IO_L23P_T3_A03_D19_14 Sch=sw[11] IOSTANDARD LVCMOS33) [get_ports { SW[12] }]; #IO_L24P_T3_35 Sch=sw[12] #set property -dict { PACKAGE PIN T13 #set_property -dict (PACKAGE_PIN U12 #set_property -dict (PACKAGE_PIN U11 IOSTANDARD LVCMOS33) [get ports (SW[13])]; #IO L20P T3 A08 D24 14 Sch=sv[13] IOSTANDARD LVCMOS33) [get ports (SW[14])]; #IO L19N T3 A09 D25 VREF 14 Sch=sv[14] IOSTANDARD LVCMOS33 } [get ports { SW[15] }]; #IO L21P T3 DQS 14 Sch=sw[15] #set property -dict { PACKAGE PIN V10 Hierarchy ← ▶ ≡ × Messages Log _ 0 6 Q 🛨 💠 II 📵 🛍 🗰 close [open C:/Users/Ventus/Documents/VivadoDocs/LAB 03c/LAB 03c.srcs/constrs 1/new/nexvs a7 50T.xdc w] Type a Tcl command here

へ ② 智 (編 40) ENG