



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](#)

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

main ▾



[Digital-electronics-1](#) / [03-vivado](#) / [README.md](#)



xrysav25 Update README.md



1 contributor

Raw

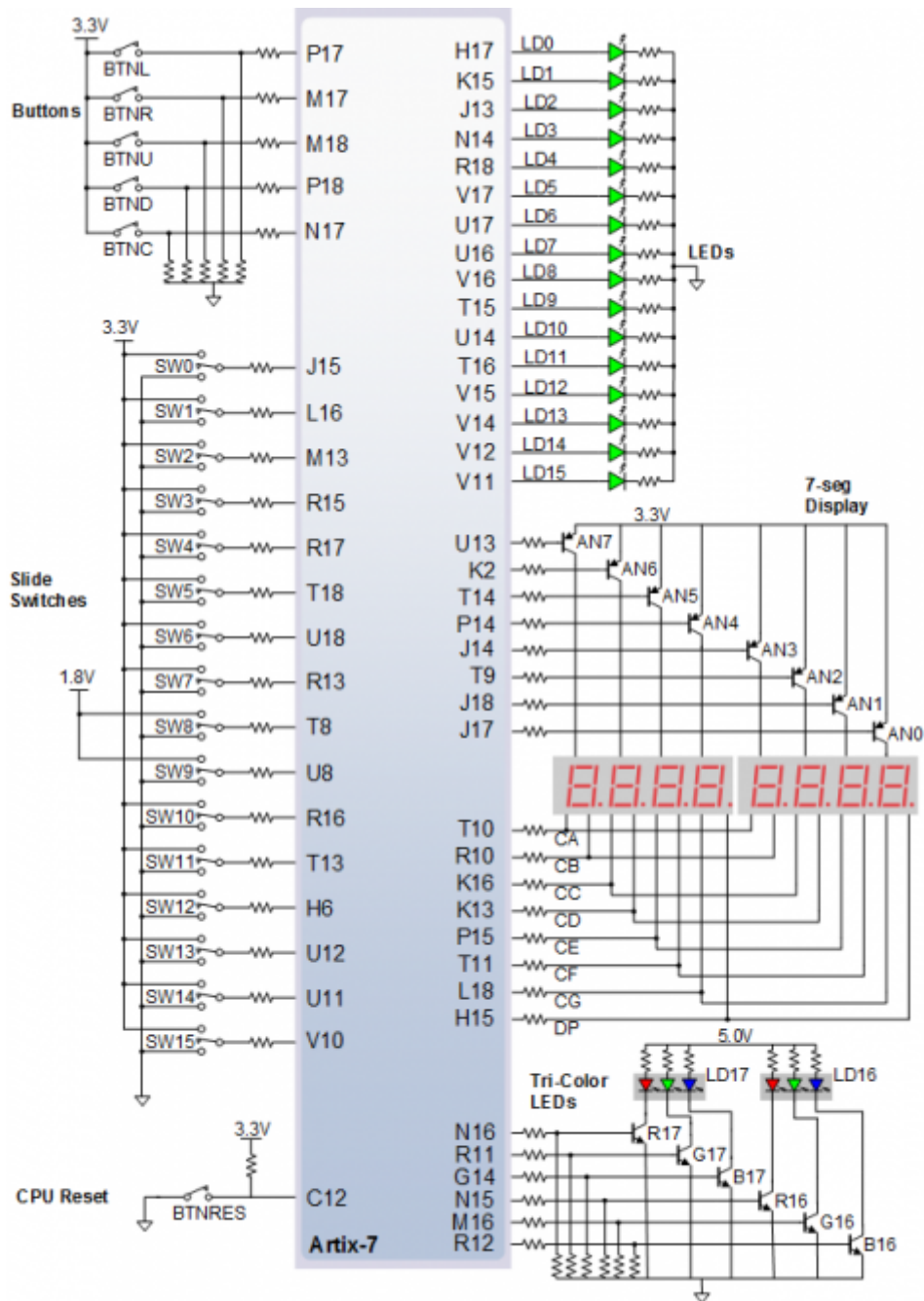
Blame



85 lines (74 sloc) | 3.76 KB

LAB 3

1. Connection with 16 switches and LEDs



2. Multiplexor

- VHDL architecture

```
architecture Behavioral of mux_4_1 is
```

```
begin
```

```
    f_o <= a_i when (sel_i = "00") else
           b_i when (sel_i = "01") else
           c_i when (sel_i = "10") else
           d_i;
```

```
end Behavioral;
```

- VHDL simuls proces

```

p_stimulus : process
begin
    s_d <= "00"; s_c <= "00"; s_b <= "00"; s_a <= "00"; s_sel <= "00"; wait for
    assert (s_f = "00") report "Test failed for sel input combination: 00" sever
    s_a <= "01"; wait for 50 ns;
    assert (s_f = "01") report "Test failed for sel input combination: 00" sever
    s_a <= "10"; wait for 50 ns;
    assert (s_f = "10") report "Test failed for sel input combination: 00" sever
    s_a <= "11"; wait for 50 ns;
    assert (s_f = "11") report "Test failed for sel input combination: 00" sever

    s_sel <= "01"; wait for 50 ns;
    assert (s_f = "00") report "Test failed for sel input combination: 01" sever
    s_b <= "01"; wait for 50 ns;
    assert (s_f = "01") report "Test failed for sel input combination: 01" sever
    s_b <= "10"; wait for 50 ns;
    assert (s_f = "10") report "Test failed for sel input combination: 01" sever
    s_b <= "11"; wait for 50 ns;
    assert (s_f = "11") report "Test failed for sel input combination: 01" sever

    s_sel <= "10"; wait for 50 ns;
    assert (s_f = "00") report "Test failed for sel input combination: 10" sever
    s_c <= "01"; wait for 50 ns;
    assert (s_f = "01") report "Test failed for sel input combination: 10" sever
    s_c <= "10"; wait for 50 ns;
    assert (s_f = "10") report "Test failed for sel input combination: 10" sever
    s_c <= "11"; wait for 50 ns;
    assert (s_f = "11") report "Test failed for sel input combination: 10" sever

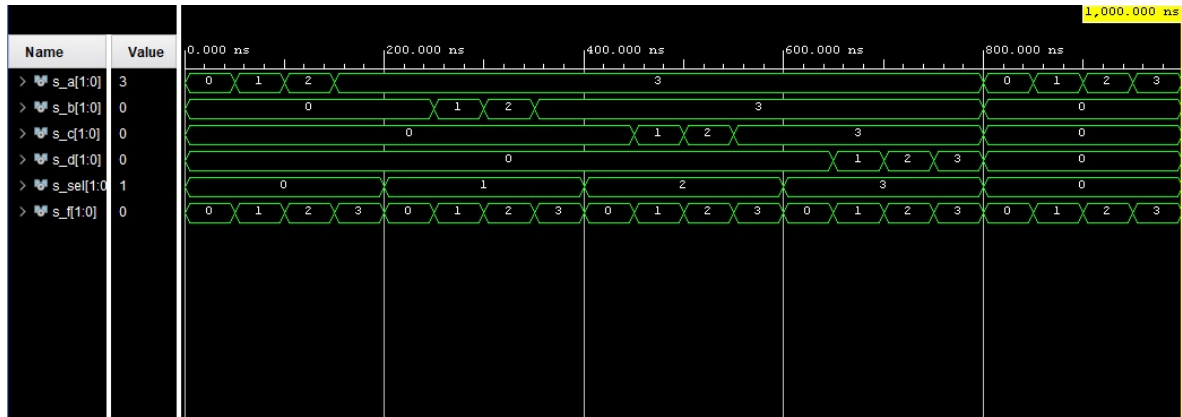
    s_sel <= "11"; wait for 50 ns;
    assert (s_f = "00") report "Test failed for sel input combination: 11" sever
    s_d <= "01"; wait for 50 ns;
    assert (s_f = "01") report "Test failed for sel input combination: 11" sever
    s_d <= "10"; wait for 50 ns;
    assert (s_f = "10") report "Test failed for sel input combination: 11" sever
    s_d <= "11"; wait for 50 ns;
    assert (s_f = "11") report "Test failed for sel input combination: 11" sever

end process p_stimulus;

```



- Simulation timewaves



3. Vivado tutorial

Vivado New Project Tutorial

1. Create a new project
2. Use difolt type: RTL Porject

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

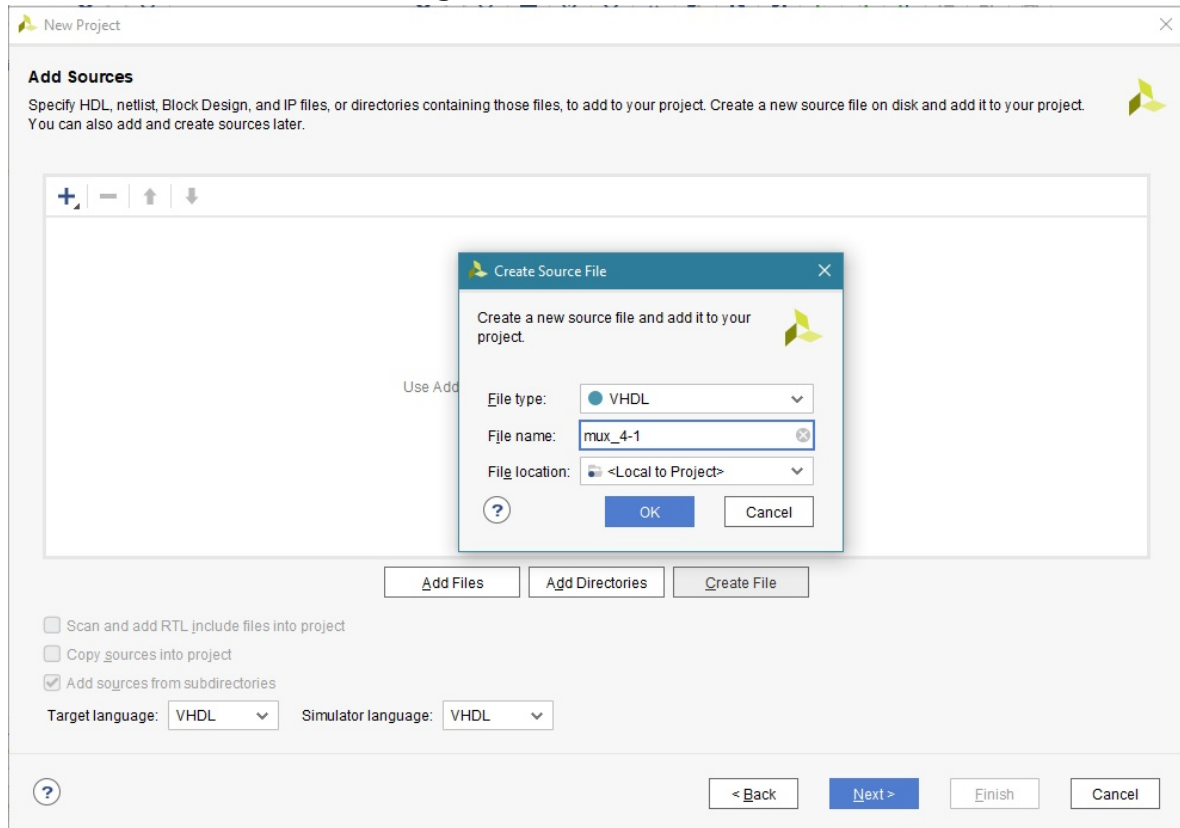
☐ **I/O Planning Project**
Do not specify design sources. You will be able to view part/package resources.

☐ **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.

? < Back Next > Finish Cancel

3. Add source - create some design file!

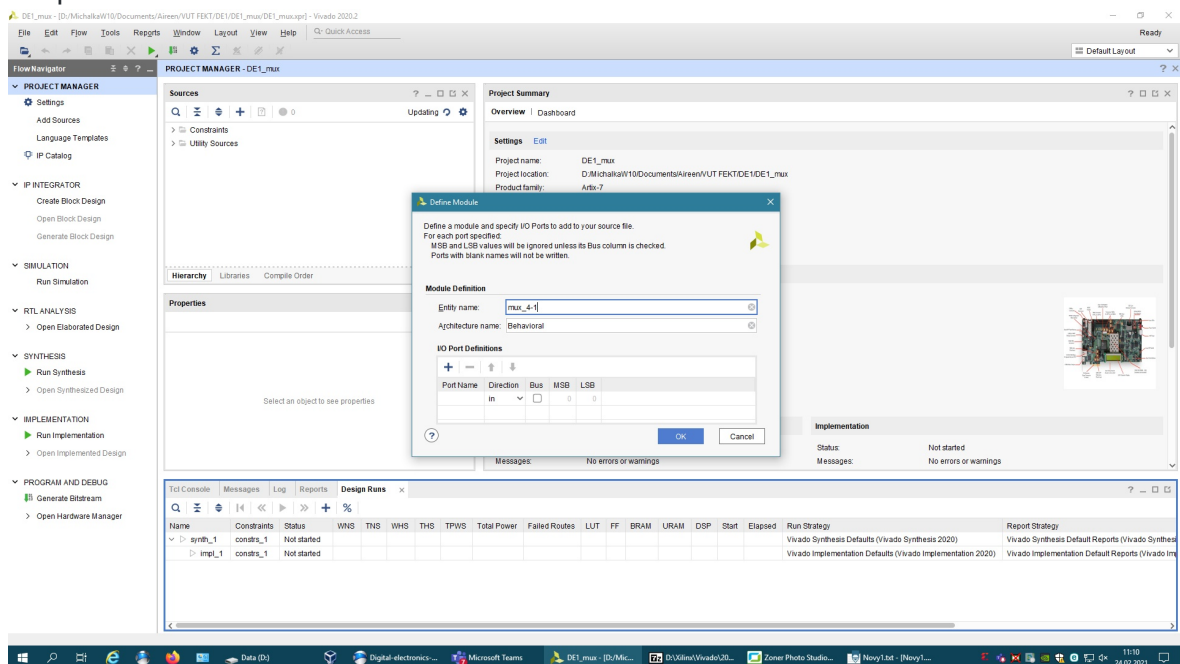


4. Add Constraints - those files are usefull for hw implementation. No need to create one when we are going to just simulate something

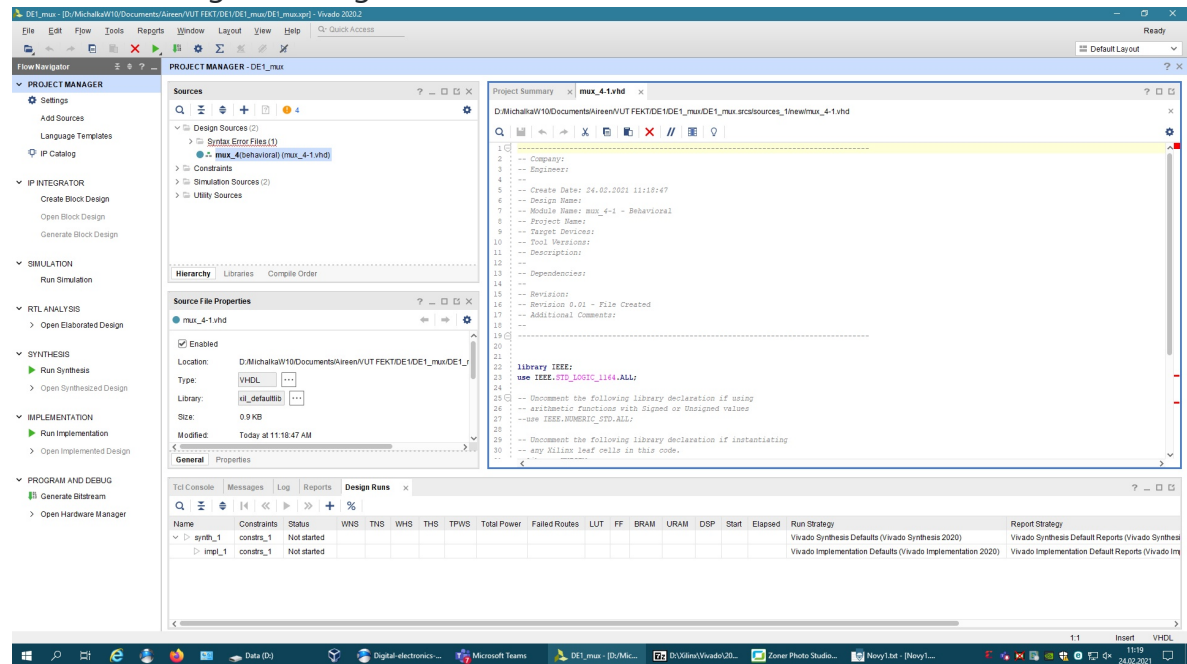
5. Default Part - choose your board!

6. Finish!

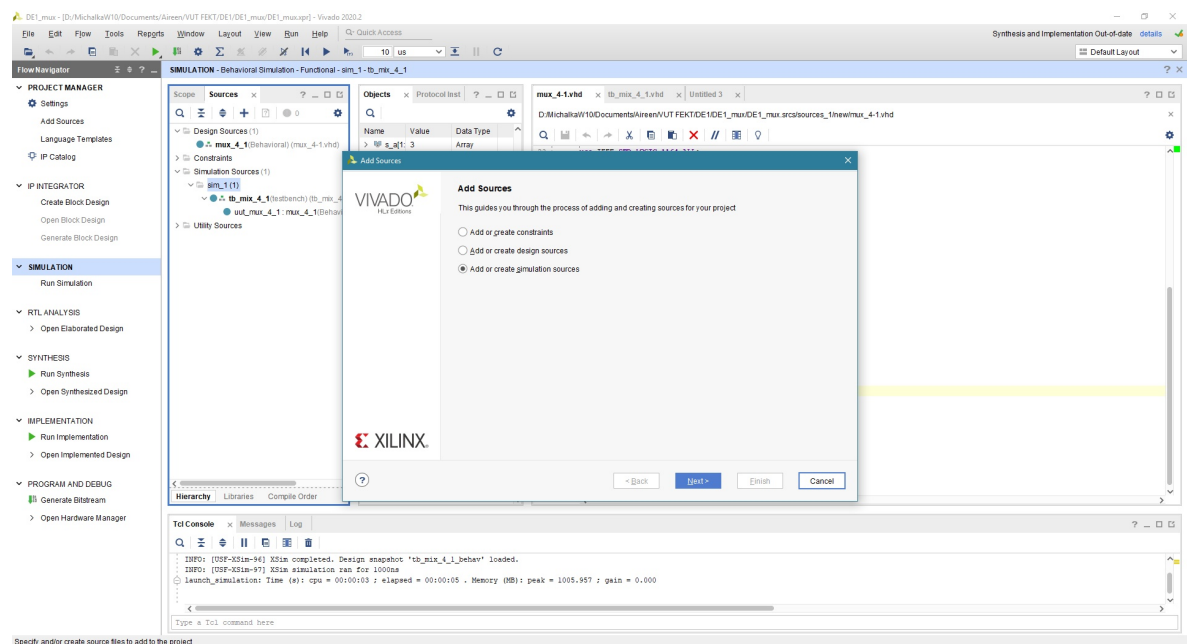
7. So here you have your new project. In this window you can add inputs and outputs.



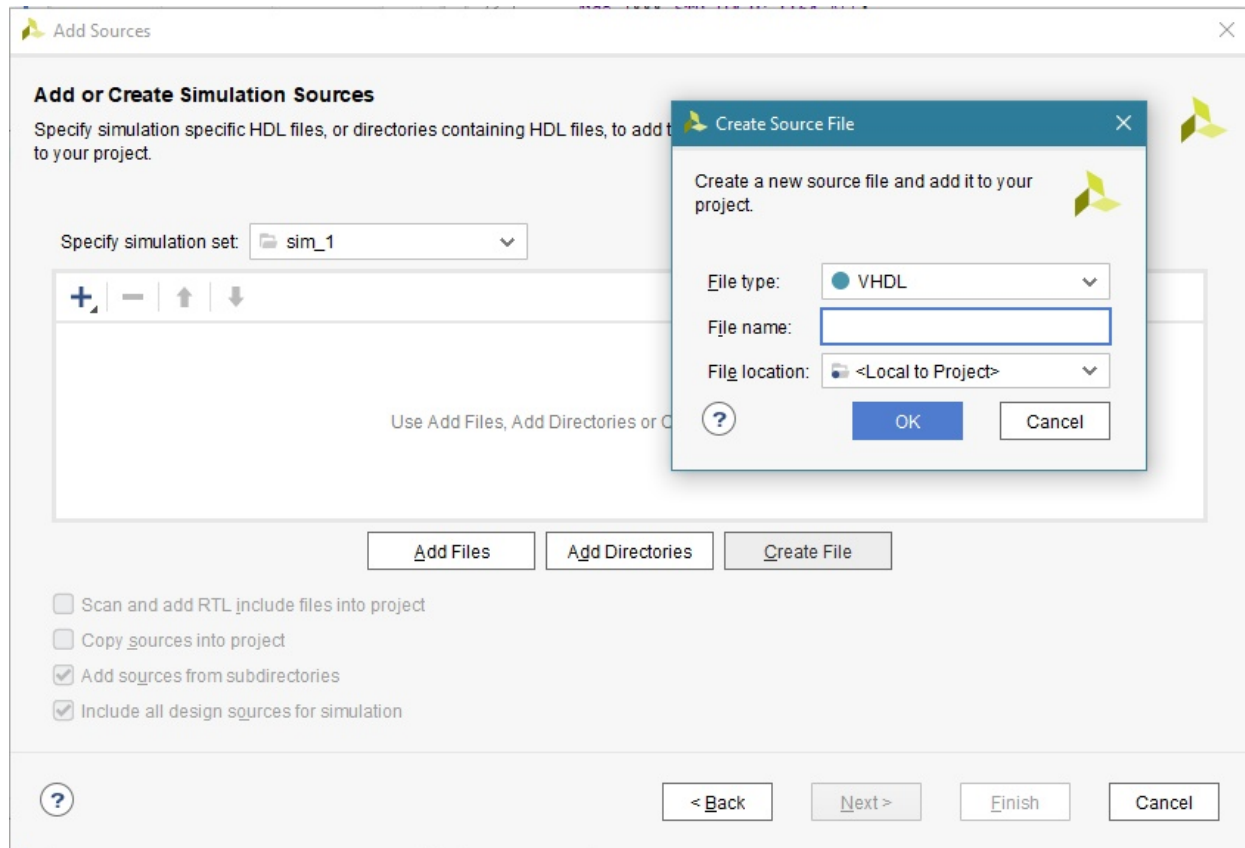
8. Your new design file is right here



9. For playing, you need some test file as well. Create one! Find "Add Source" in Project Manager menu or right click on something in your source menu. Select "Add or create simulation source"



10. Go next and choose create file (if you don't have one somewhere hidden!) and name your file. It's better to use "tb" and name of your design file.



11. After completing testbench, you can run it! Find "Run Simulation" in Simulation menu or right click in Source menu on Simulation source