CSC 343 Spring 2021

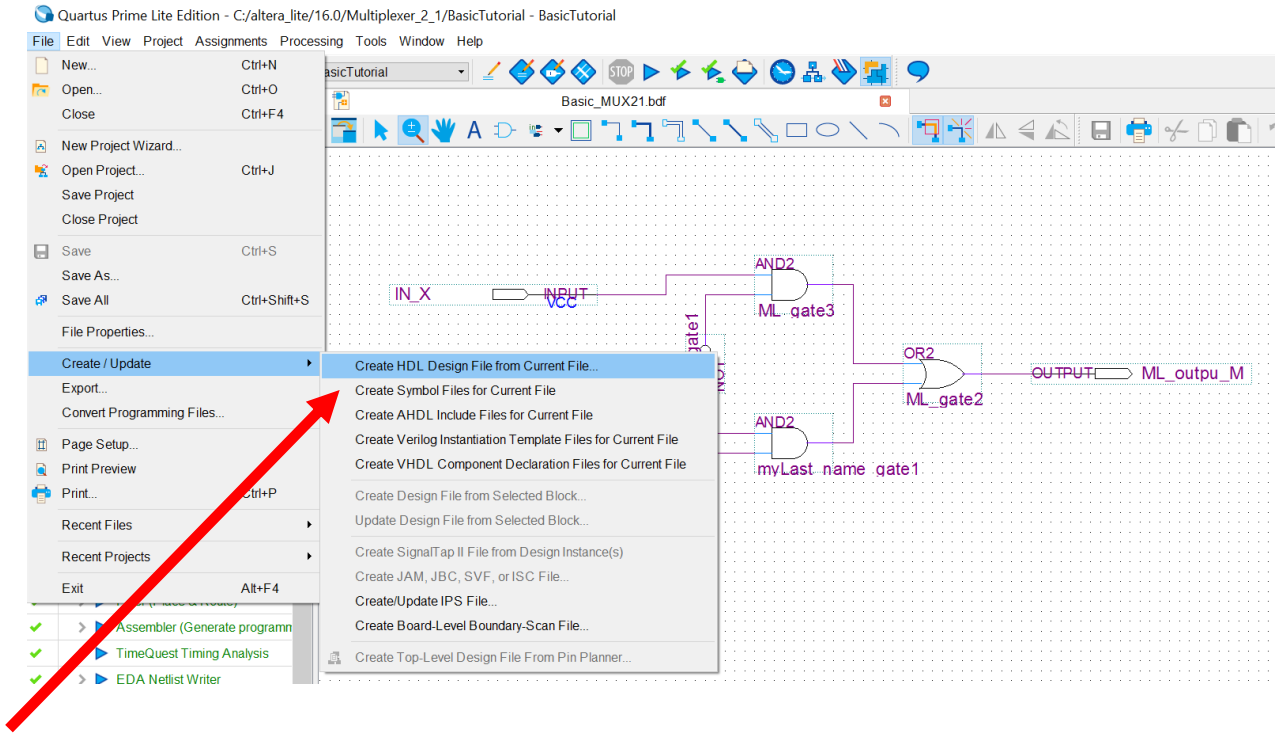
Laboratory Exercise Tutorial:
Design Entry Specified:
Schematic Diagram of digital circuits
Hardware Description Language (VHDL)
VHDL with LPM(Library Parameterized Modules)

Instructor: Professor Izidor Gertner

Creating VHDL code from a Block Diagram

With your block diagram file open, navigate to **File > Create / Update > Create HDL Design File from Current File…**

Notice that you have to be using the full diagram made from primitive gates, not the one where the symbol file is used.



CSC 343 Spring 2021

Laboratory Exercise Tutorial:

Design Entry Specified:

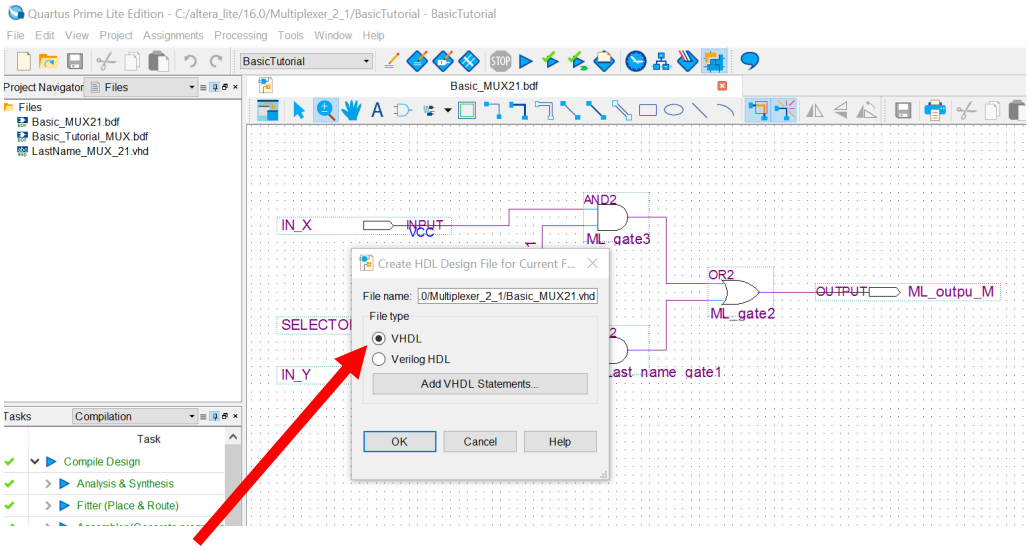
Schematic Diagram of digital circuits

Hardware Description Language (VHDL)

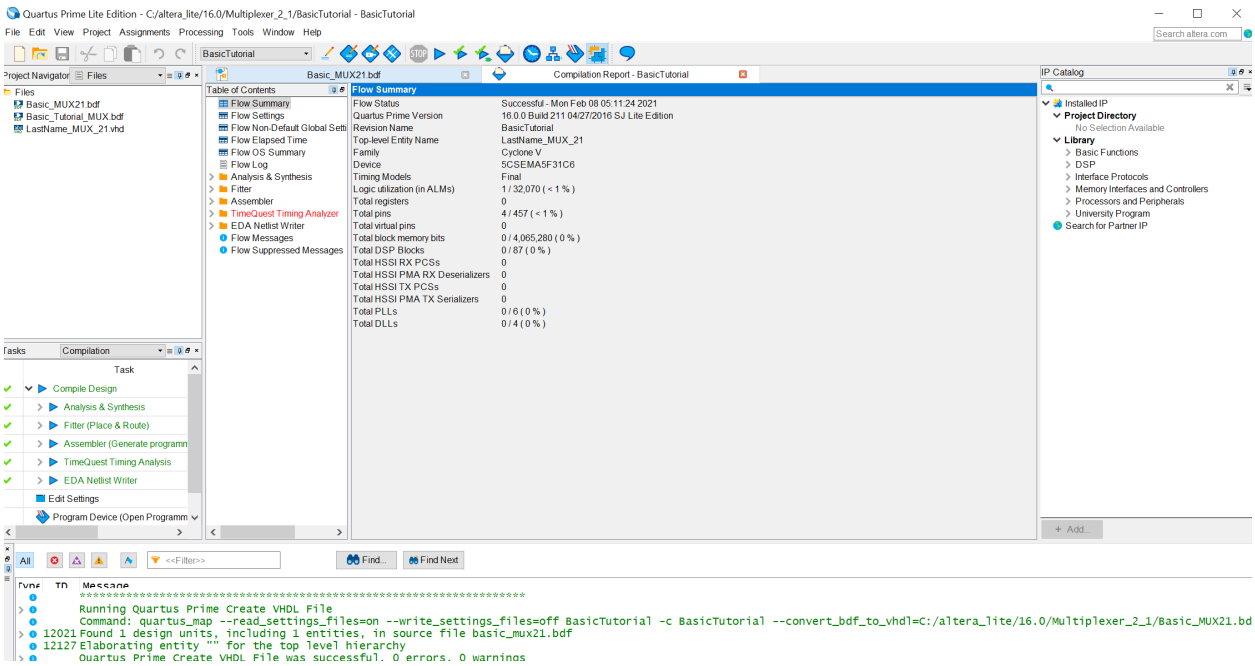
VHDL with LPM(Library Parameterized Modules)

Instructor: Professor Izidor Gertner

When the pop up window opens, select the VHDL option.



The compilation will automatically start, however this compilation is only concerning the file creation, unlike a normal compilation of a top-level entity.

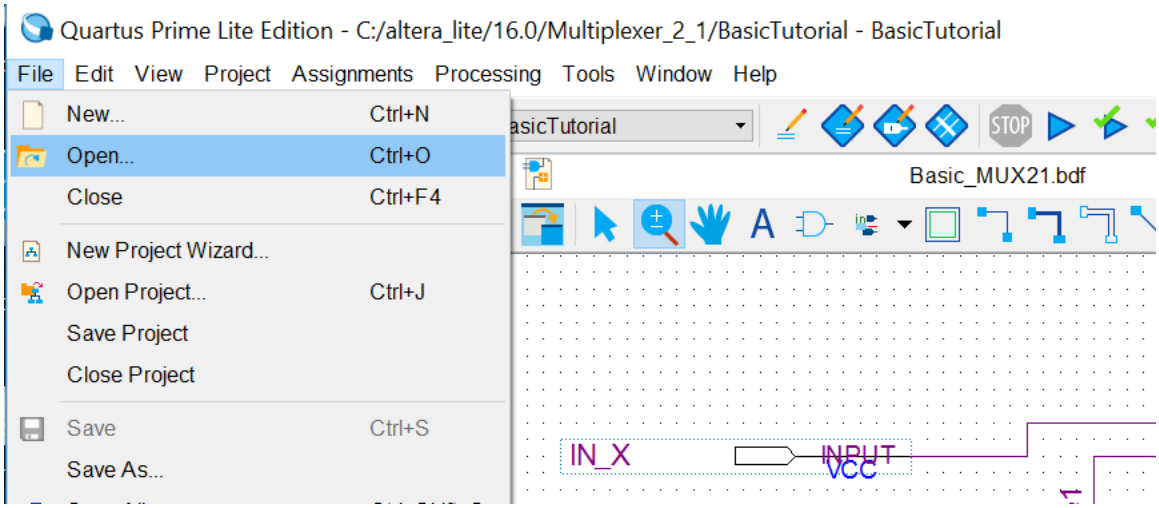


CSC 343 Spring 2021

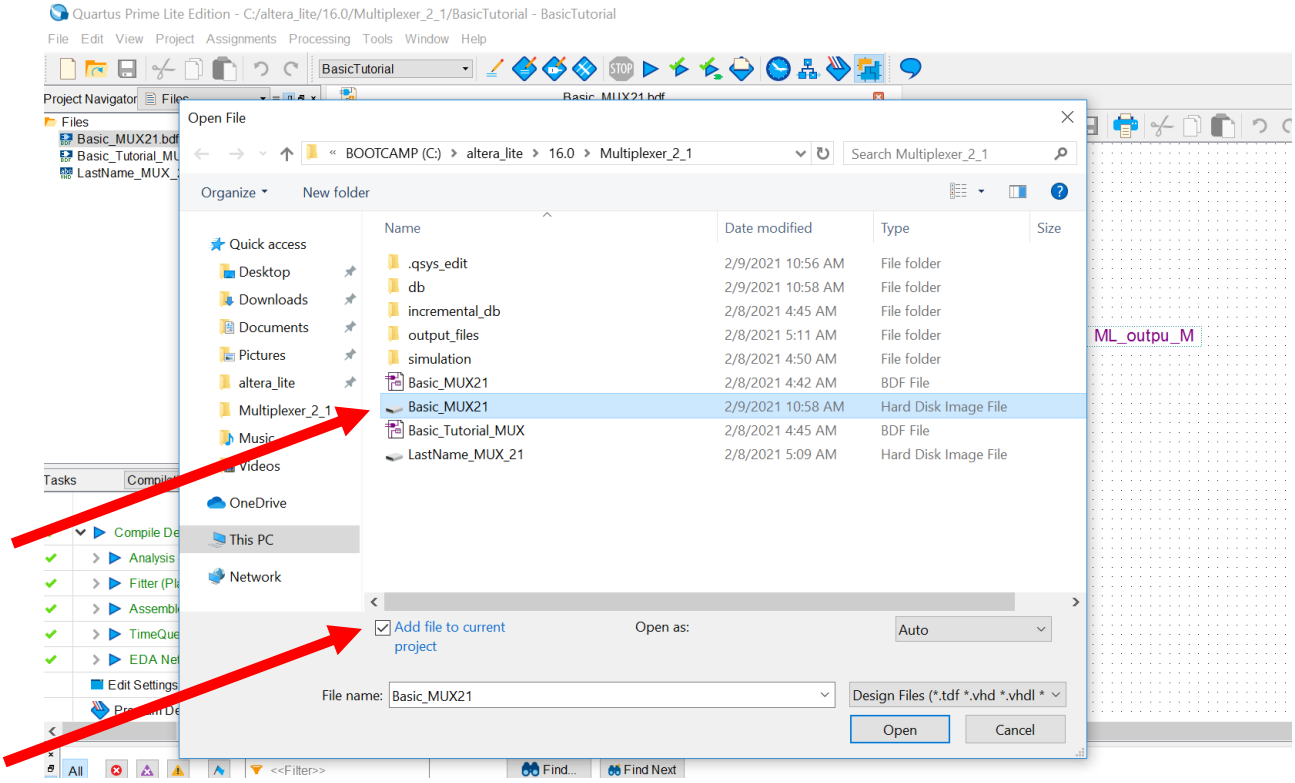
Laboratory Exercise Tutorial:
Design Entry Specified:
Schematic Diagram of digital circuits
Hardware Description Language (VHDL)
VHDL with LPM(Library Parameterized Modules)

Instructor: Professor Izidor Gertner

Open the file, navigate to where your project is.



Make sure “Add file to current project” is ticked.



CSC 343 Spring 2021

Laboratory Exercise Tutorial:

Design Entry Specified:

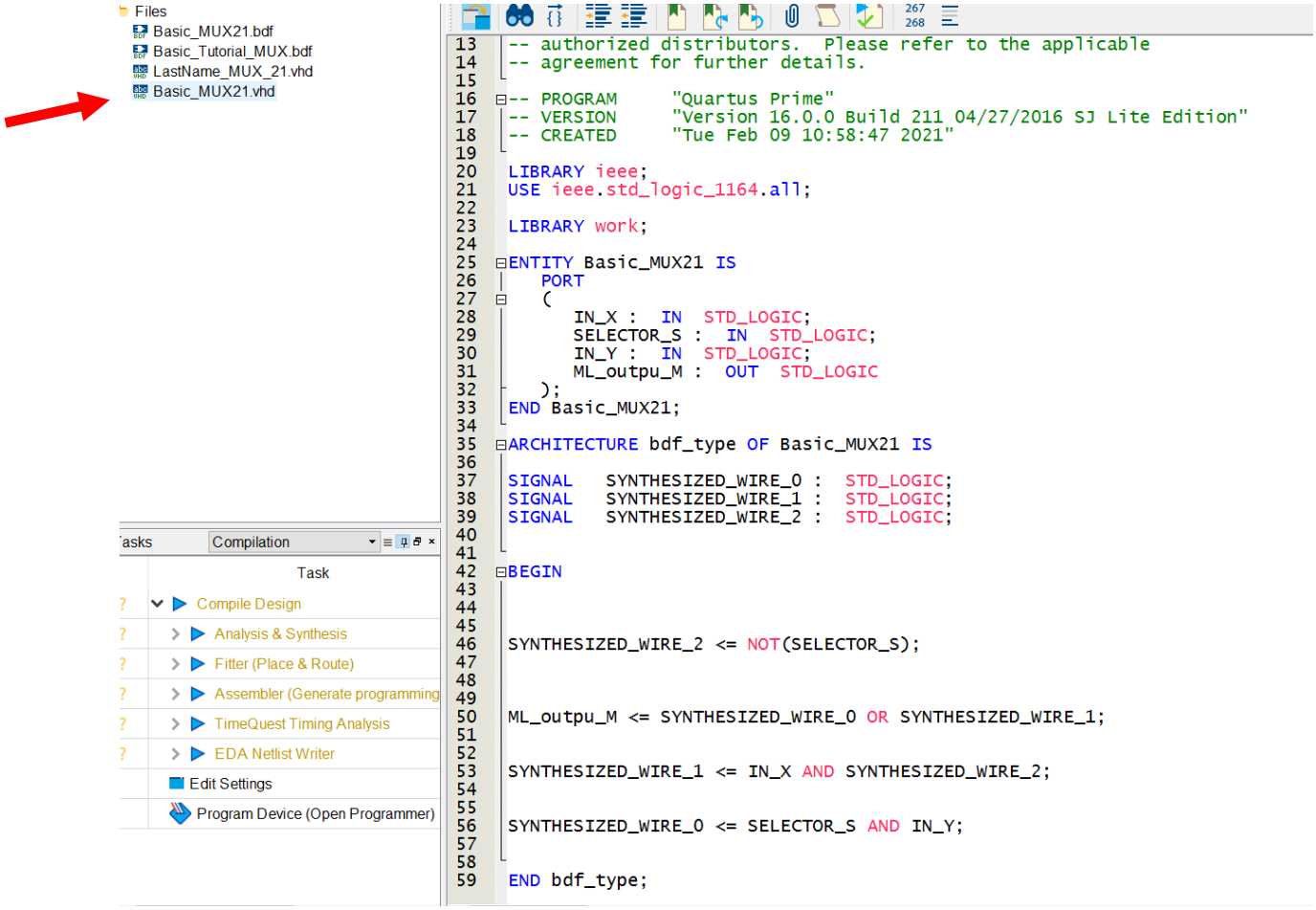
Schematic Diagram of digital circuits

Hardware Description Language (VHDL)

VHDL with LPM(Library Parameterized Modules)

Instructor: Professor Izidor Gertner

This is what your file should look like within the project.



Set this new file as the top-level entity in the project, and then start compilation.

CSC 343 Spring 2021

Laboratory Exercise Tutorial:

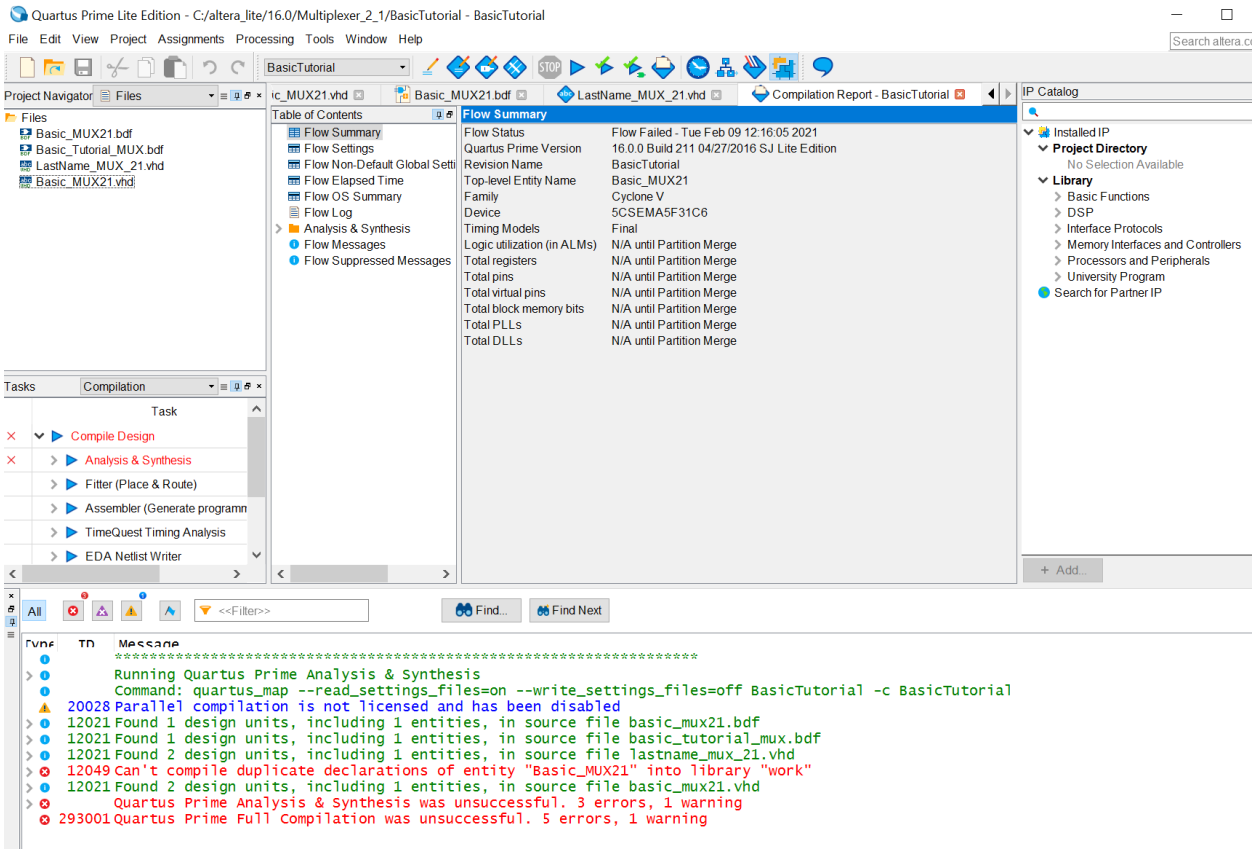
Design Entry Specified:

Schematic Diagram of digital circuits

Hardware Description Language (VHDL)

VHDL with LPM(Library Parameterized Modules)

Instructor: Professor Izidor Gertner



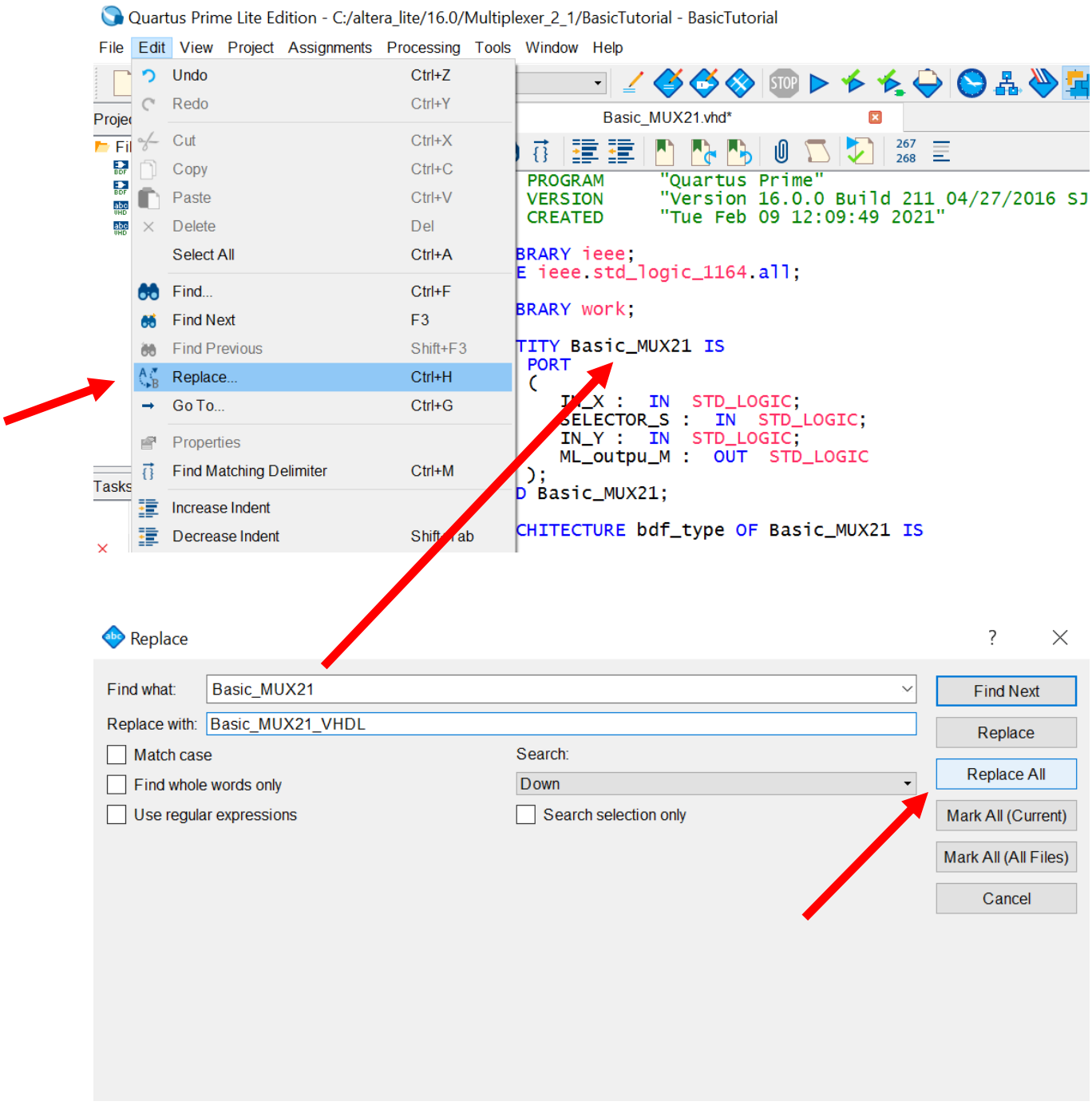
A library error occurs due to the generated file's name being the same as the block diagram one. In the earlier dialogue where we selected VHDL as the file type, the filename was locked. There is a quick way to fix this:

Go back to editing the file, and find the **Edit > Replace** feature

CSC 343 Spring 2021

Laboratory Exercise Tutorial:
Design Entry Specified:
Schematic Diagram of digital circuits
Hardware Description Language (VHDL)
VHDL with LPM(Library Parameterized Modules)

Instructor: Professor Izidor Gertner

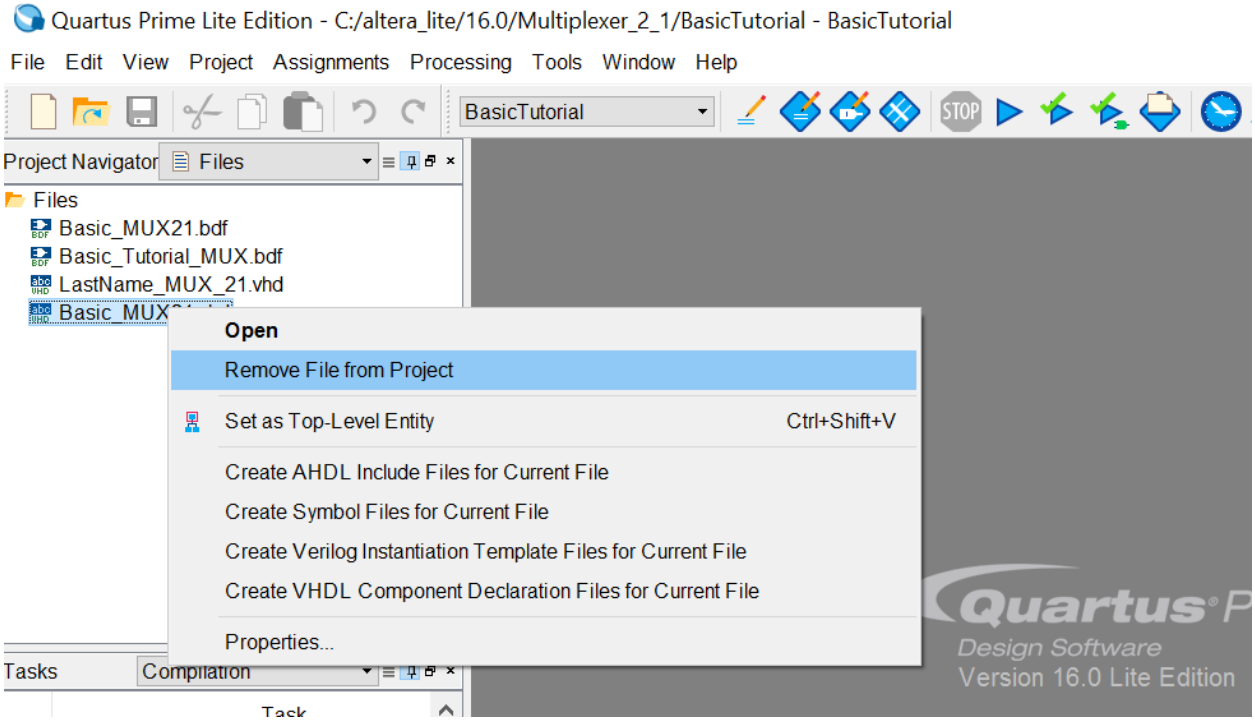


Replace all instances of the problematic name.

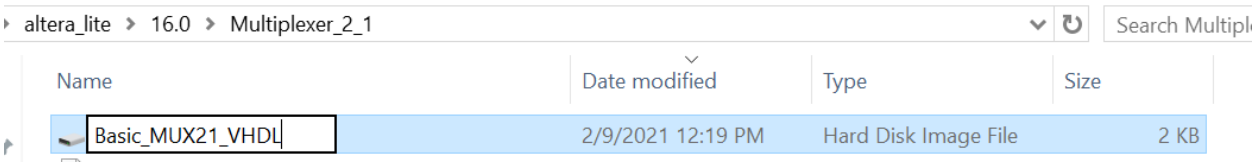
Laboratory Exercise Tutorial:
Design Entry Specified:
Schematic Diagram of digital circuits
Hardware Description Language (VHDL)
VHDL with LPM(Library Parameterized Modules)

Instructor: Professor Izidor Gertner

Remove the file from the project in the **Files** tab

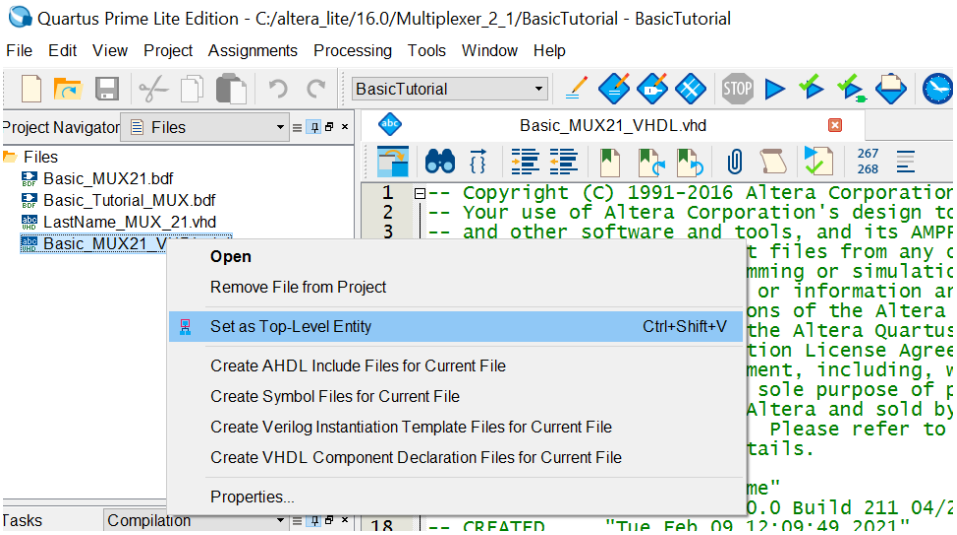


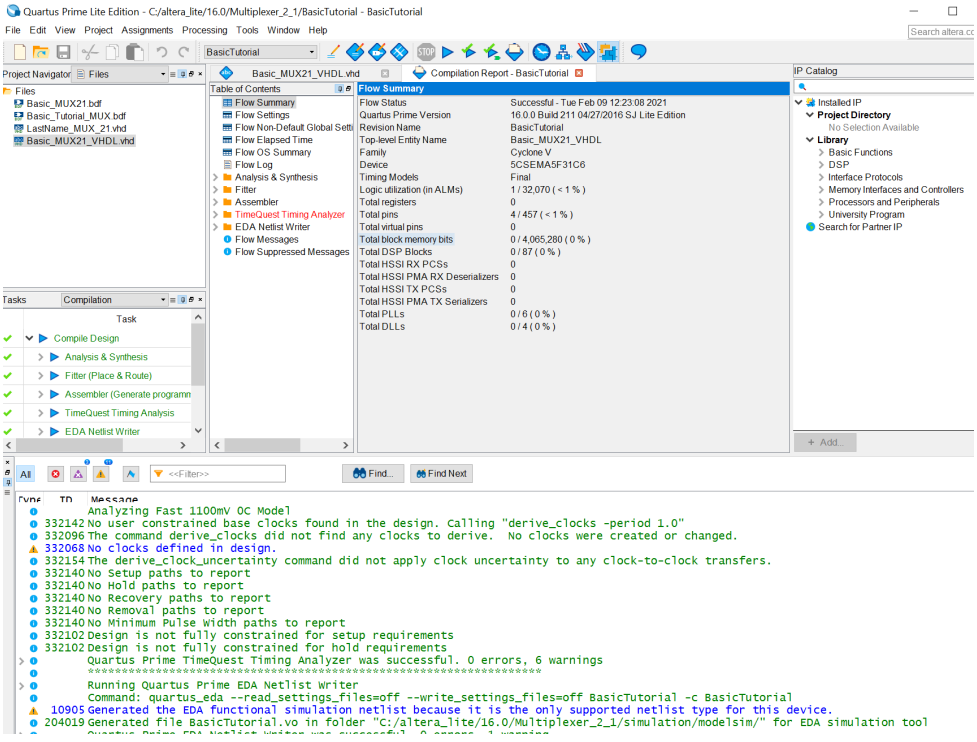
Find the file in your project folder and manually re-name it to whatever you replaced the problematic file name with.



Now, repeat the step where you opened the file and again make sure the “Add file to project” box is ticked.

Set it as the top-level entity once again, and compile. It will have no error this time.





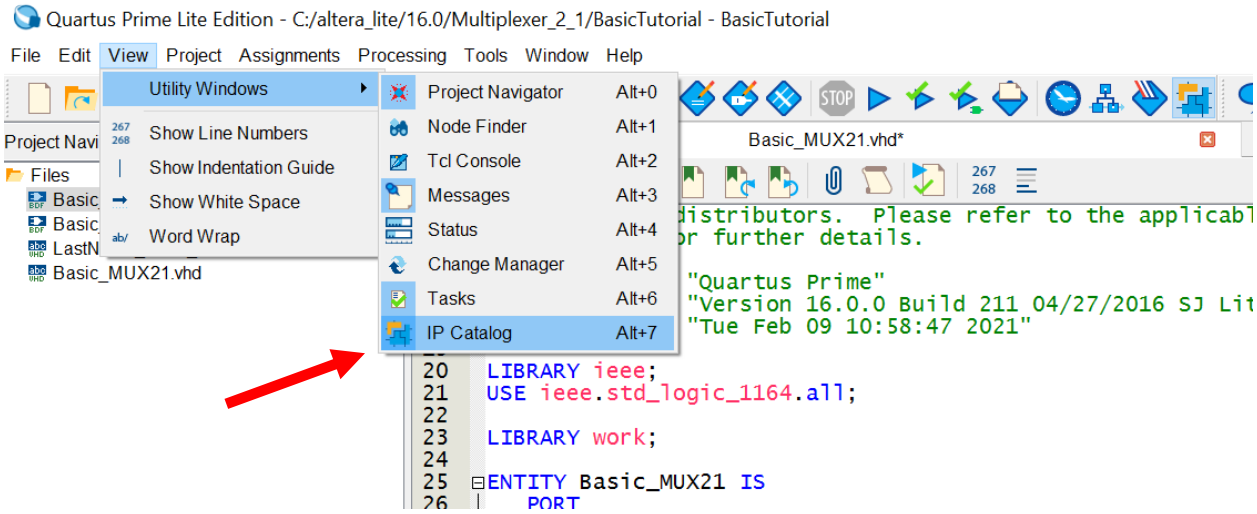
CSC 343 Spring 2021

Laboratory Exercise Tutorial:
Design Entry Specified:
Schematic Diagram of digital circuits
Hardware Description Language (VHDL)
VHDL with LPM(Library Parameterized Modules)

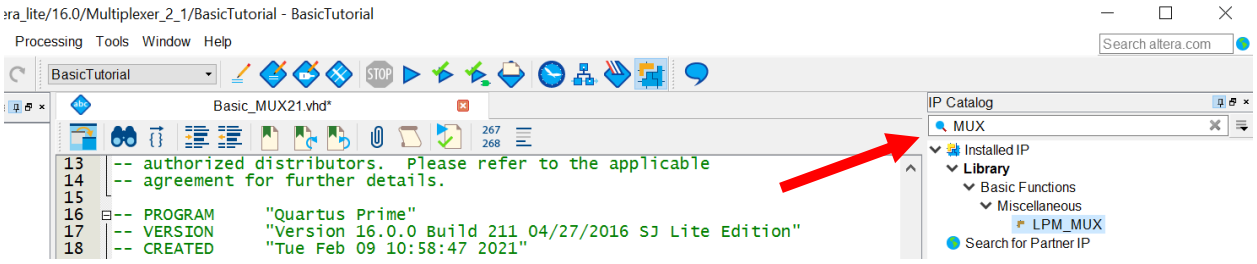
Instructor: Professor Izidor Gertner

Using the Library of Parameterized Modules (LPM) to Create a VHDL File

The Multiplexer is already included as a pre-built customizable module in Quartus. There are a variety of other LPMs in the **IP Catalog** too. Let's first make sure that the **IP Catalog** window is enabled; find this toggle in **View > Utility Windows > IP Catalog**



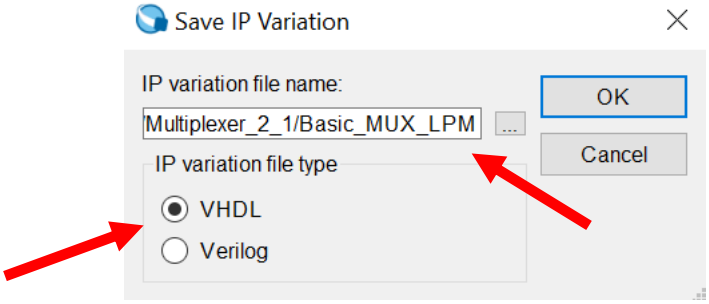
Within the **IP Catalog**, search for **MUX**. Double click on **LPM_MUX**.



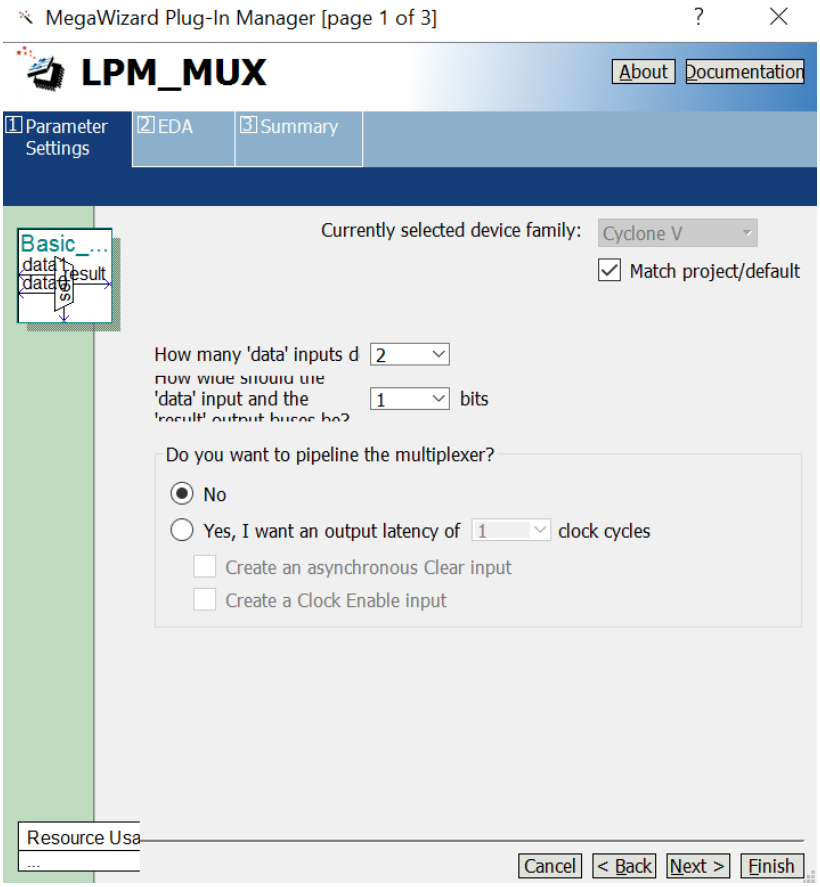
Laboratory Exercise Tutorial:
Design Entry Specified:
Schematic Diagram of digital circuits
Hardware Description Language (VHDL)
VHDL with LPM(Library Parameterized Modules)

Instructor: Professor Izidor Gertner

This time you will be able to change the filename. I chose **Basic_MUX_LPM.vhd**, make sure it is not the same as your other files, even if they are block diagrams. Again, choose **VHDL** as the file type.



The **MegaWizard** window will open, make sure your options are the same.



CSC 343 Spring 2021

Laboratory Exercise Tutorial:

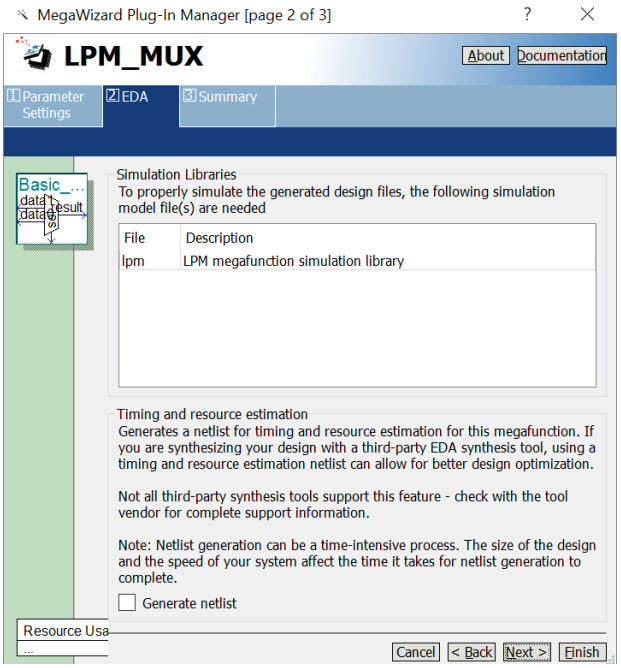
Design Entry Specified:

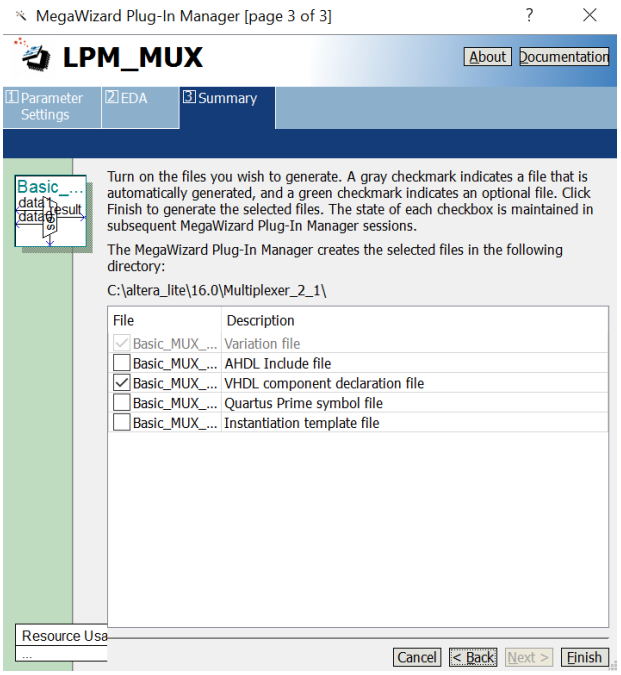
Schematic Diagram of digital circuits

Hardware Description Language (VHDL)

VHDL with LPM(Library Parameterized Modules)

Instructor: Professor Izidor Gertner

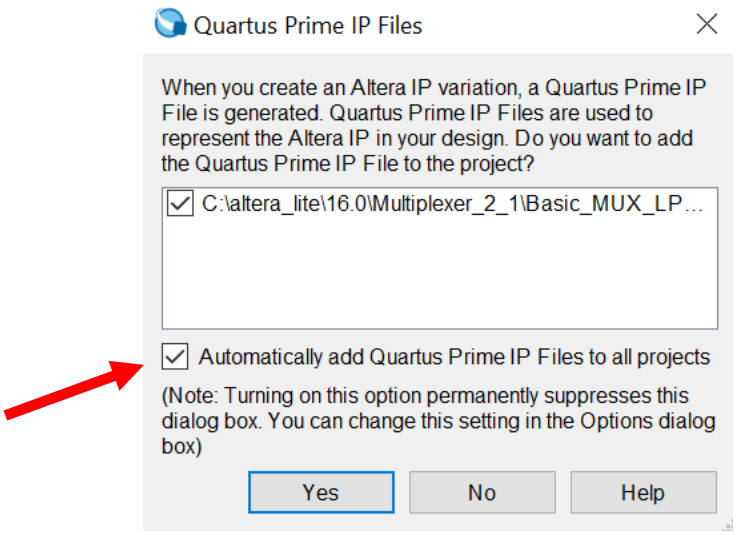




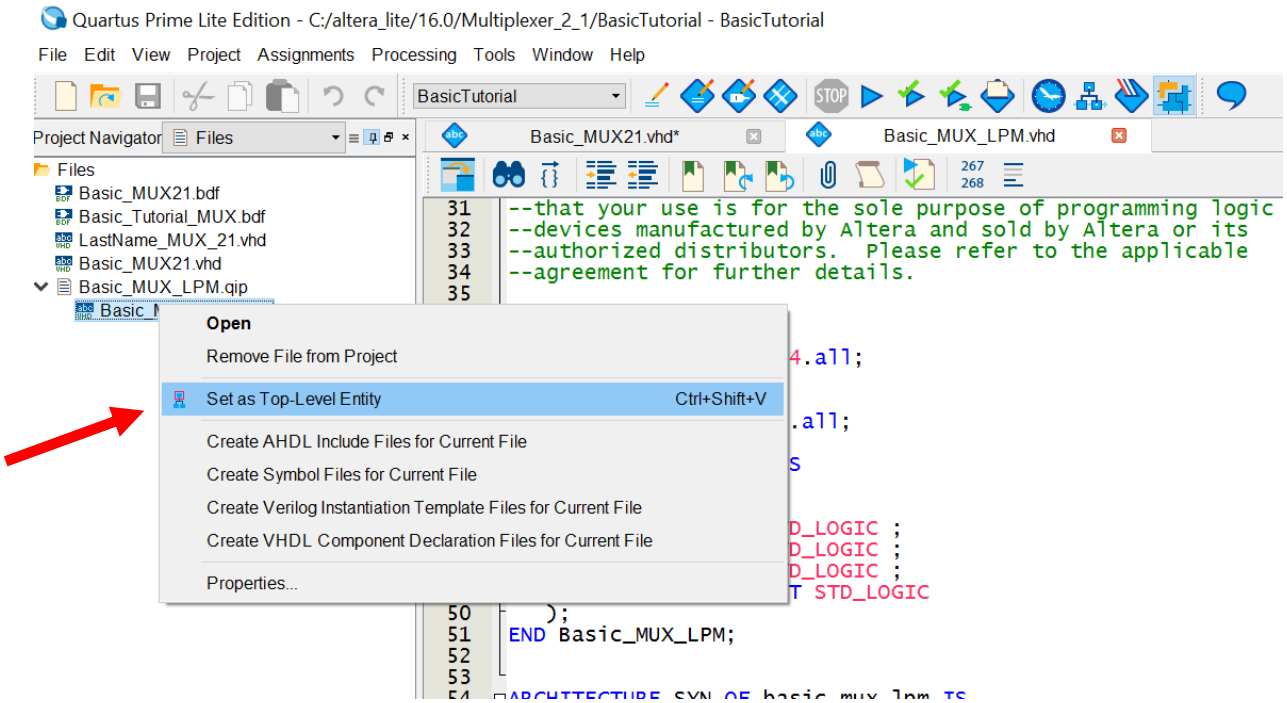
Laboratory Exercise Tutorial:
Design Entry Specified:
Schematic Diagram of digital circuits
Hardware Description Language (VHDL)
VHDL with LPM(Library Parameterized Modules)

Instructor: Professor Izidor Gertner

Make sure the box is ticked.



Find the new file in the **Files** tab, set it as the top-level entity, and compile.



CSC 343 Spring 2021

Laboratory Exercise Tutorial:

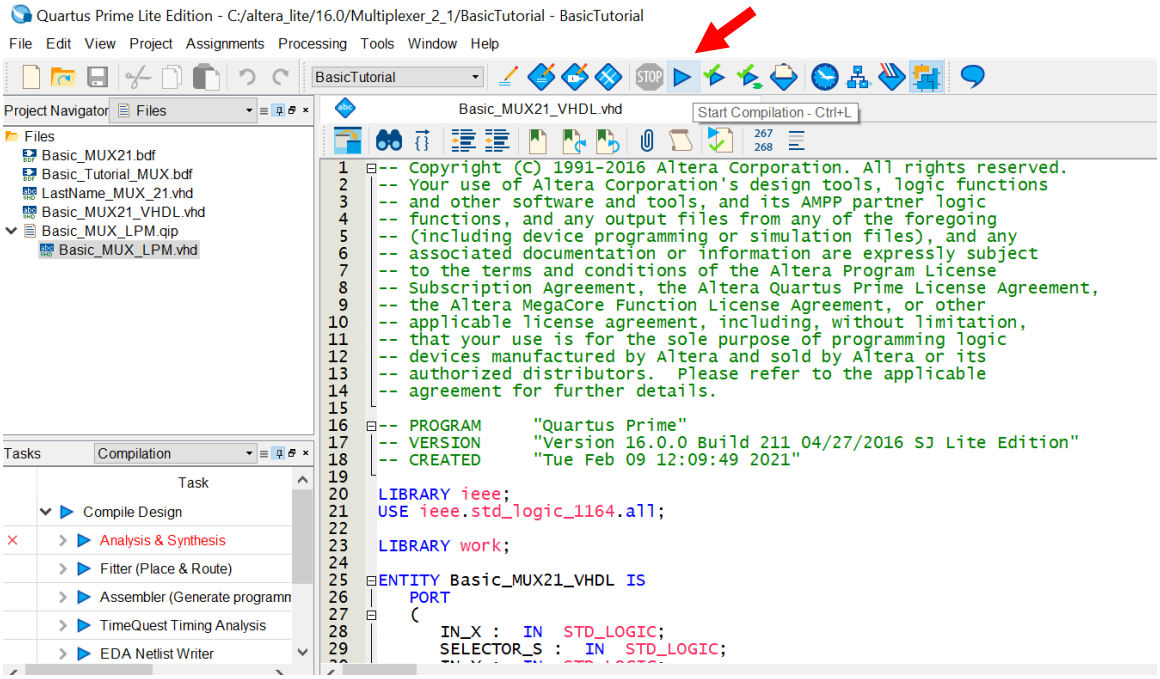
Design Entry Specified:

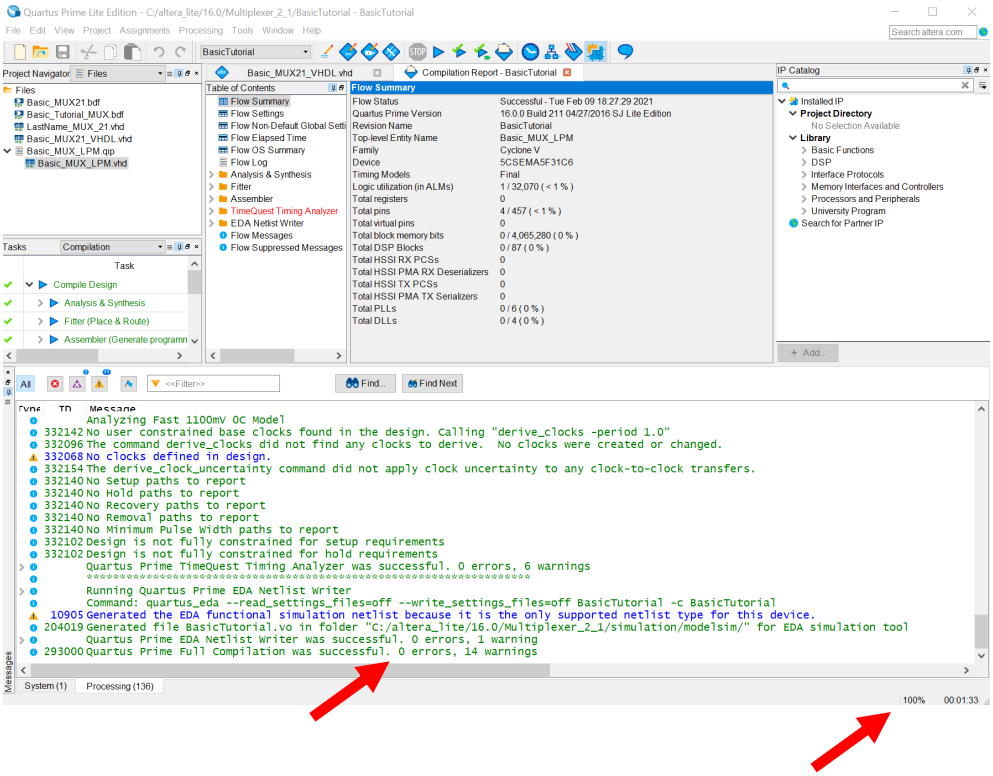
Schematic Diagram of digital circuits

Hardware Description Language (VHDL)

VHDL with LPM(Library Parameterized Modules)

Instructor: Professor Izidor Gertner





CSC 343 Spring 2021

Laboratory Exercise Tutorial:

Design Entry Specified:

Schematic Diagram of digital circuits

Hardware Description Language (VHDL)

VHDL with LPM(Library Parameterized Modules)

Instructor: Professor Izidor Gertner

The code generated by Quartus looks more complex than the VHDL implementation we wrote by hand for a 2:1 Multiplexer. This is because the Quartus version is adapted to handle larger input sizes (for example 4:1 or 8:1).

Basic_MUX_LPM.vhd

```
36
37  LIBRARY ieee;
38  USE ieee.std_logic_1164.all;
39
40  LIBRARY lpm;
41  USE lpm.lpm_components.all;
42
43  ENTITY Basic_MUX_LPM IS
44     PORT
45     (
46        data0    : IN STD_LOGIC ;
47        data1    : IN STD_LOGIC ;
48        sel      : IN STD_LOGIC ;
49        result     : OUT STD_LOGIC
50     );
51  END Basic_MUX_LPM;
52
53
54  ARCHITECTURE SYN OF basic_mux_lpm IS
55
56  -- type STD_LOGIC_2D is array (NATURAL RANGE <>, NATURAL RANGE <>) of STD_LOGIC;
57
58     SIGNAL sub_wire0  : STD_LOGIC ;
59     SIGNAL sub_wire1  : STD_LOGIC_2D (1 DOWNTO 0, 0 DOWNTO 0);
60     SIGNAL sub_wire2  : STD_LOGIC ;
61     SIGNAL sub_wire3  : STD_LOGIC ;
62     SIGNAL sub_wire4  : STD_LOGIC_VECTOR (0 DOWNTO 0);
63     SIGNAL sub_wire5  : STD_LOGIC_VECTOR (0 DOWNTO 0);
64     SIGNAL sub_wire6  : STD_LOGIC ;
65
66  BEGIN
67     sub_wire2    <= data0;
68     sub_wire0    <= data1;
69     sub_wire1(1, 0)    <= sub_wire0;
70     sub_wire1(0, 0)    <= sub_wire2;
71     sub_wire3    <= sel;
72     sub_wire4(0)    <= sub_wire3;
73     sub_wire6    <= sub_wire5(0);
74     result    <= sub_wire6;
75
76     LPM_MUX_component : LPM_MUX
77     GENERIC MAP (
78        lpm_size => 2,
79        lpm_type => "LPM_MUX",
80        lpm_width => 1,
81        lpm_widths => 1
82     )
83     PORT MAP (
84        data => sub_wire1,
85        sel => sub_wire4,
86        result => sub_wire5
87     );
88
89
90
91  END SYN;
```

Notice also that there are no logical statements above. The main action is happening behind the scenes within the **LPM Library**.