

Using IP in Quartus II



ADC with DE- series boards

Master Degree in Automation Engineering

Digital Programmable Systems

Department of Electrical and Information Engineering

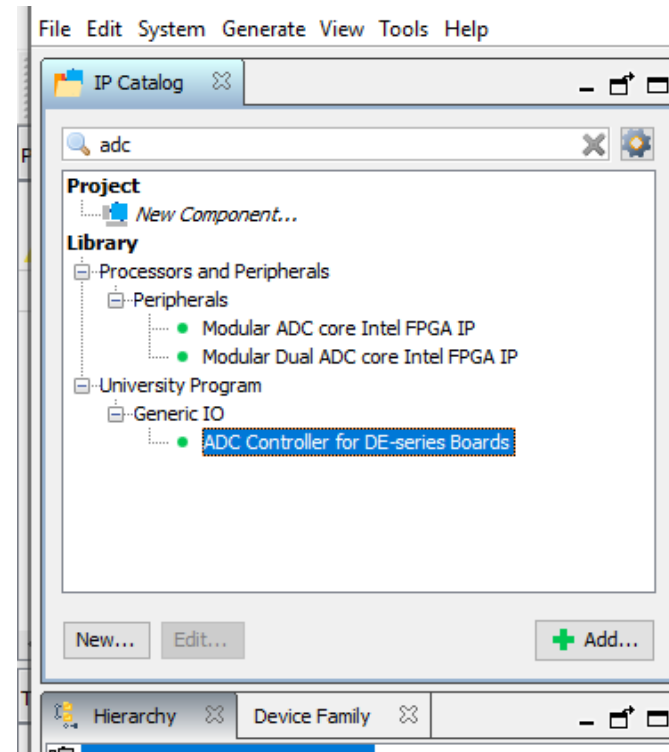
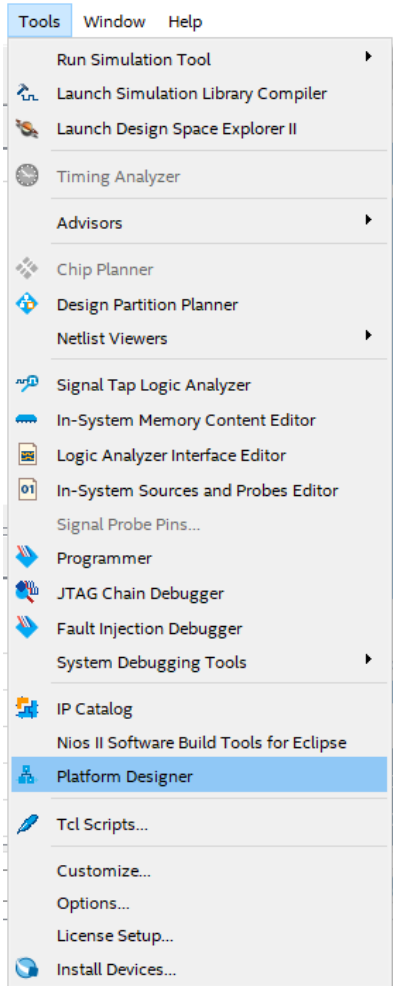
Politecnico di Bari

Dr. Eng. Martino De Carlo

ADC with DE- series boards

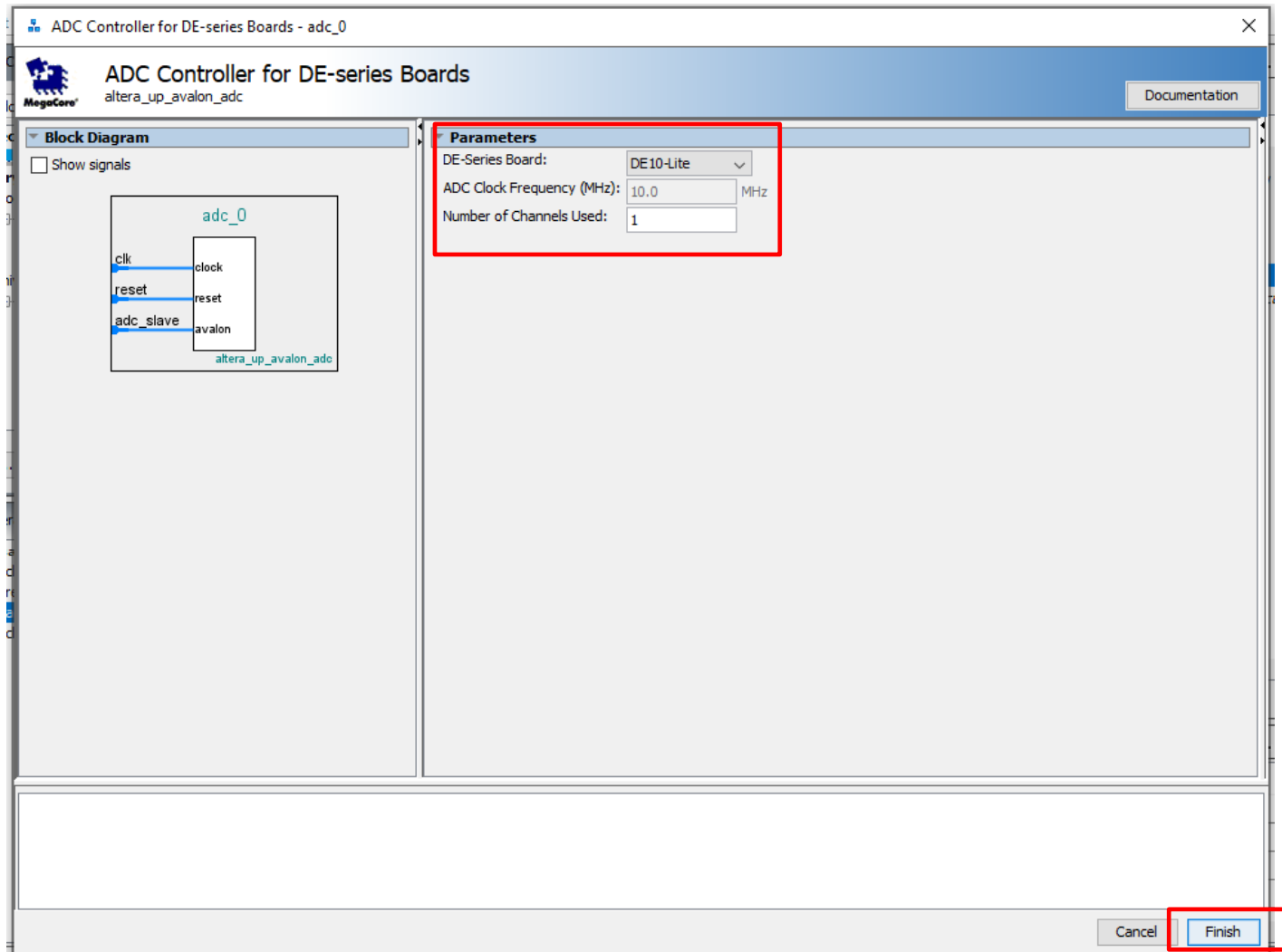
- 1) Open a new blank project
- 2) Go on Tools -> Platform Designer

3) Search on the left for "ADC Controller for DE-series boards" and double click



ADC with DE- series boards

4) Set parameters as in Fig and click on "Finish"



ADC with DE- series boards

5) Click on the grey dots to activate the connections
(the links should become black)

The screenshot displays the Intel Platform Designer (PDS) software interface. The main window is titled "Platform Designer - unsaved.qsys* (C:\intelFPGA_lite\20.1\trialADC\unsaved.qsys)". The interface is divided into several panes:

- IP Catalog:** Shows a search for "adc" and a list of components. The "ADC Controller for DE-series Boards" is highlighted under the "Generic IO" category.
- System Contents:** Displays a table of components and their connections. The table has columns for "Use", "Conn...", "Name", "Description", "Export", "Clock", and "Bd".
- Hierarchy:** Shows a tree view of the system components. The "reset" component is highlighted.
- Messages:** Displays a warning message: "1 Warning" and "unsaved.adc_0 adc_0.adc_slave must be connected to an Avalon-MM master".

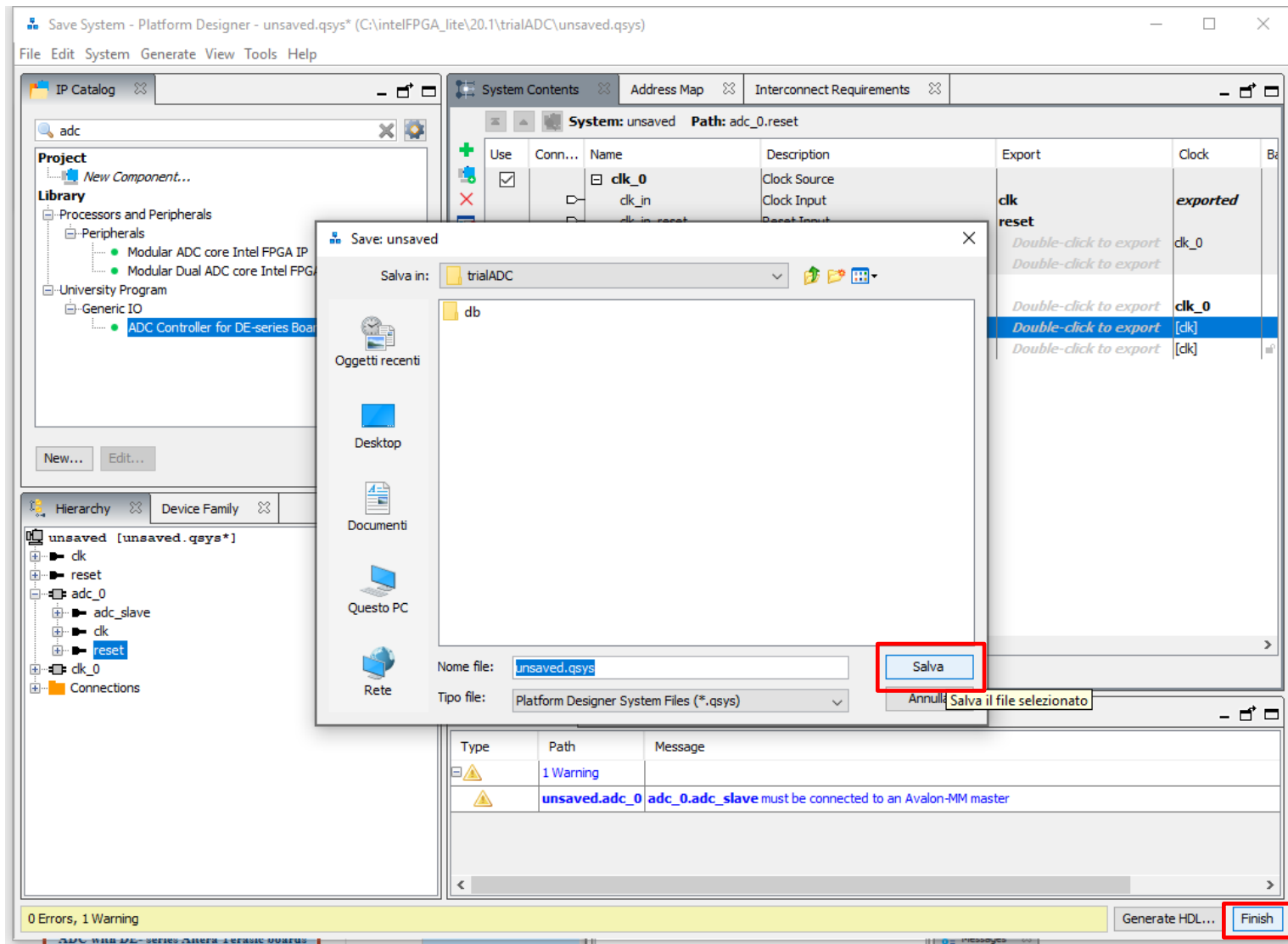
The "System Contents" table is as follows:

Use	Conn...	Name	Description	Export	Clock	Bd
<input checked="" type="checkbox"/>		clk_0	Clock Source			
		clk_in	Clock Input			
		clk_in_reset	Reset Input			
		clk	Clock Output			
		clk_reset	Reset Output			
<input checked="" type="checkbox"/>		adc_0	ADC Controller for DE-series Boards			
		clk	Clock Input	Double-click to export	clk_0	
		reset	Reset Input	Double-click to export	[clk]	
		adc_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	

The "Messages" pane shows a warning message: "1 Warning" and "unsaved.adc_0 adc_0.adc_slave must be connected to an Avalon-MM master".

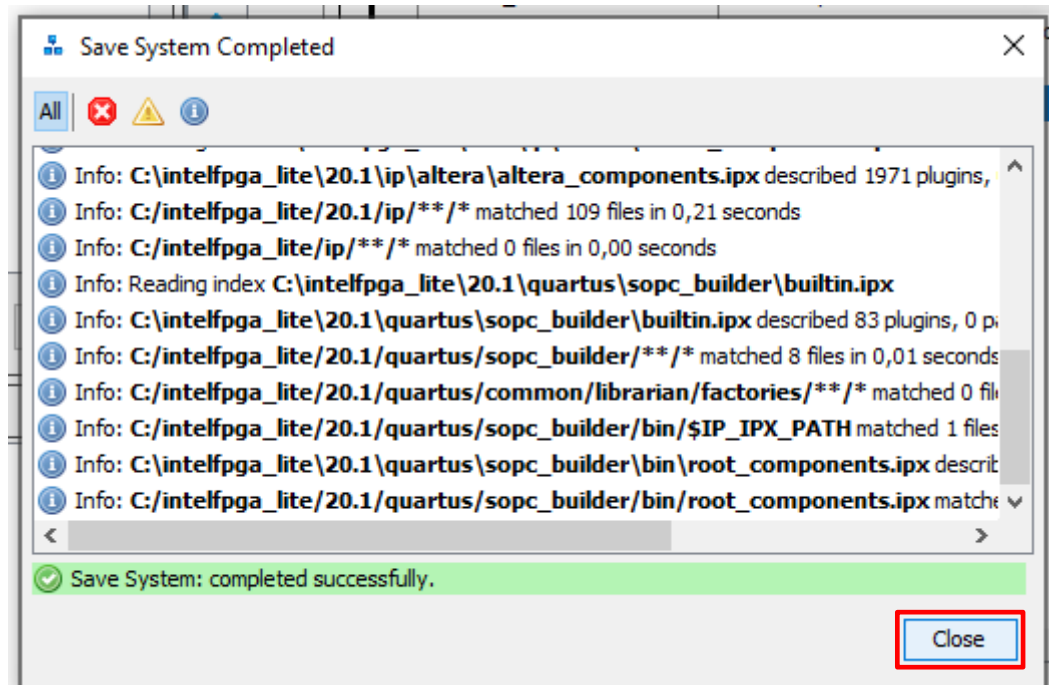
ADC with DE- series boards

6) Click on "Finish" and save when asked



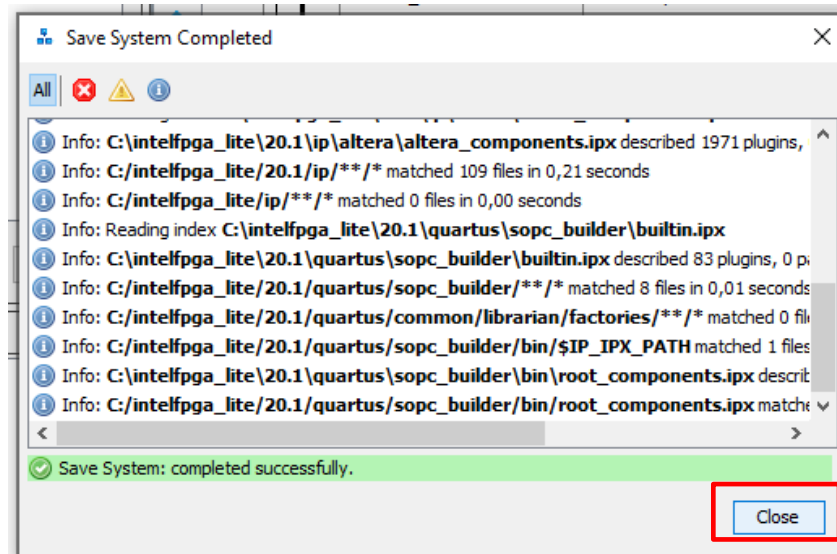
ADC with DE- series boards

7) Click on "Close"

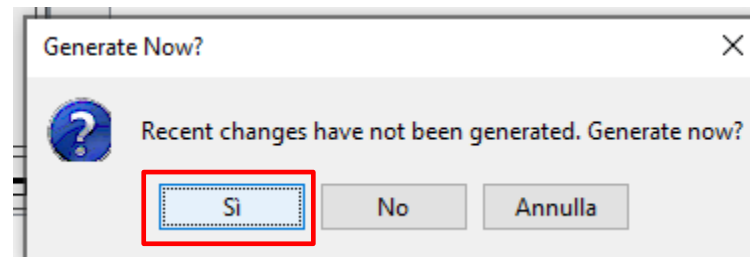


ADC with DE- series boards

8) Click on "Close"

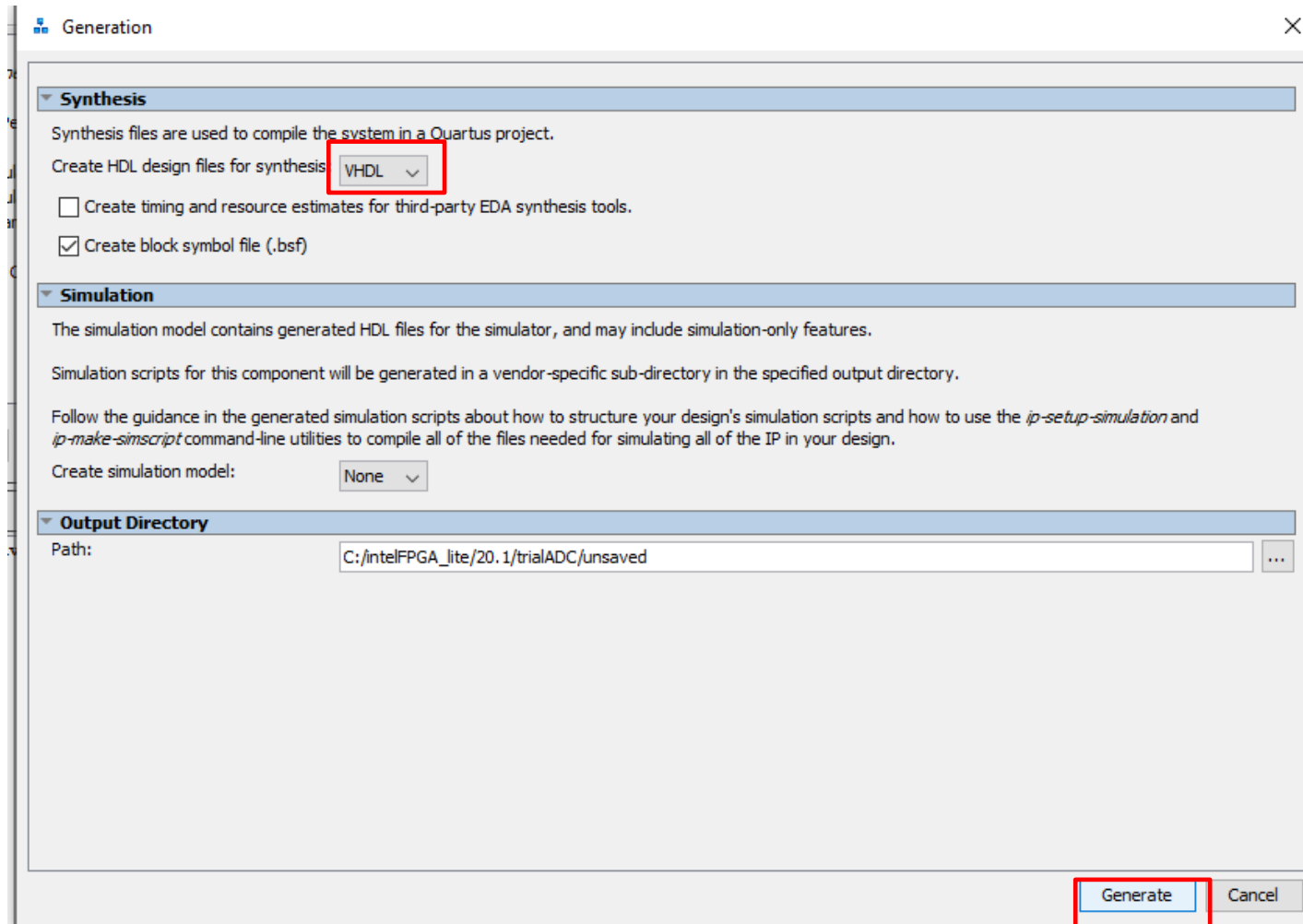


9) Click on "Yes" when asked to generate the code



ADC with DE- series boards

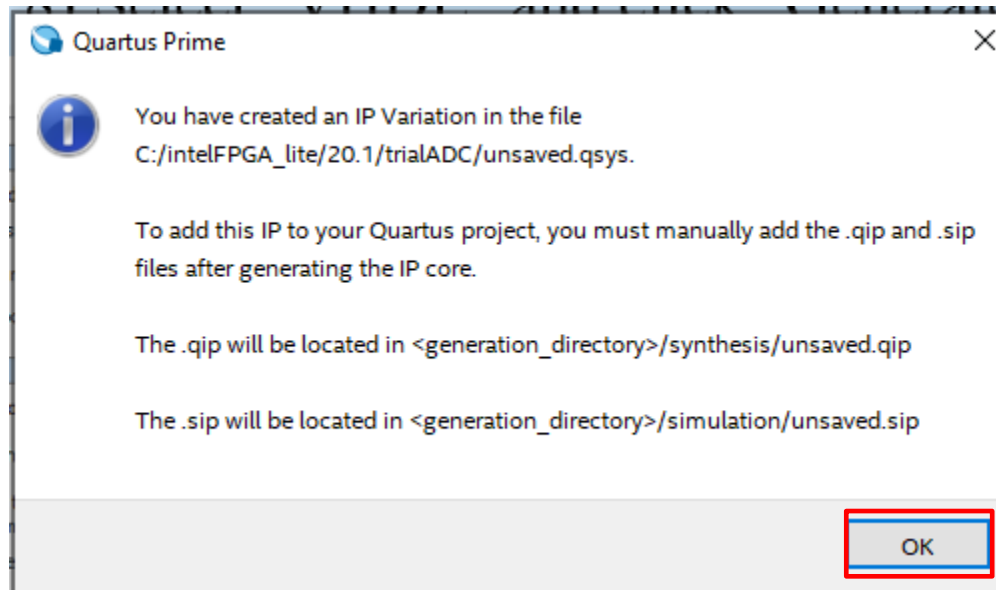
10) Select "VHDL" and click "Generate"



11) Click on "Finish"

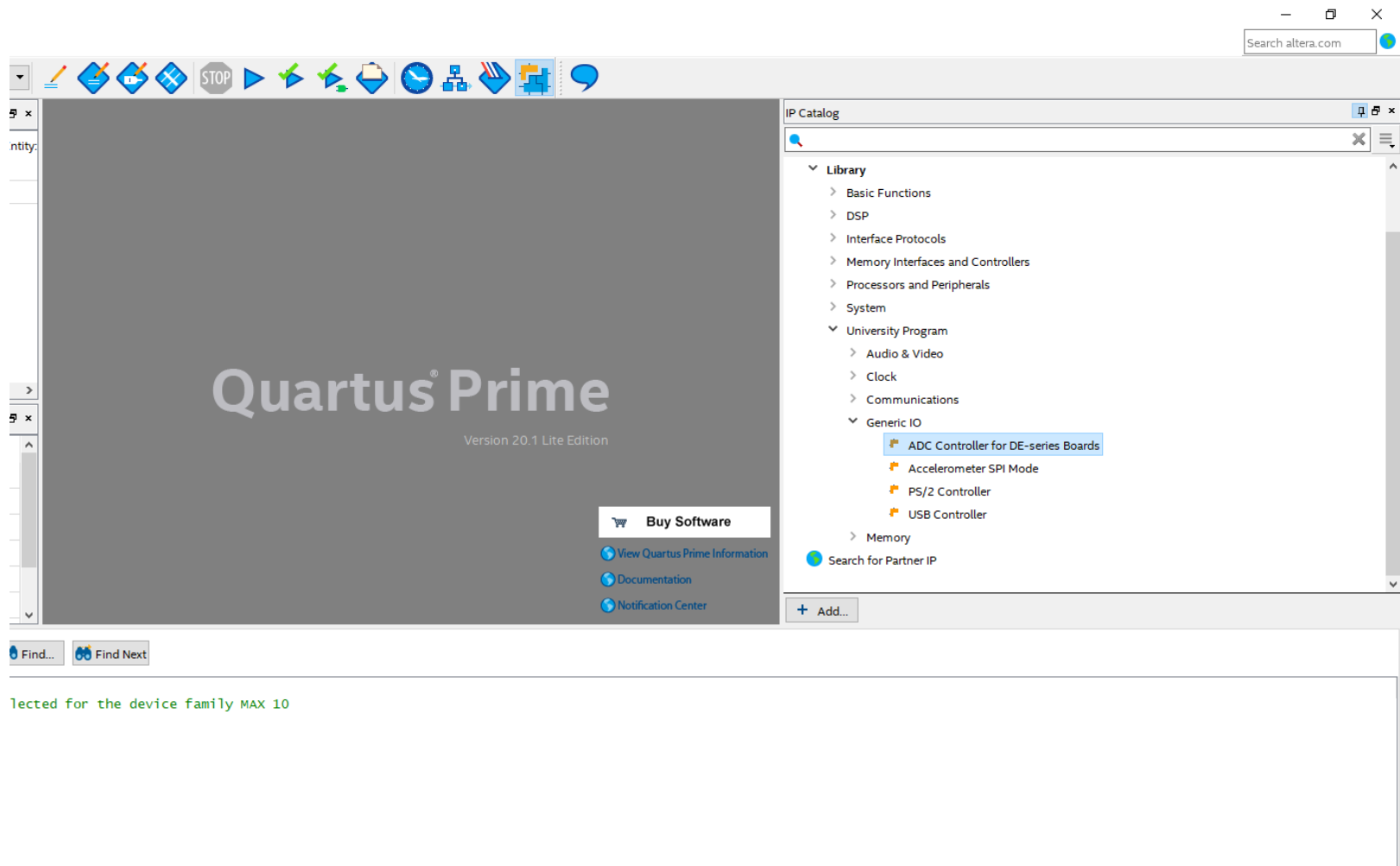
ADC with DE- series boards

12) Click on "Finish" and then "OK"



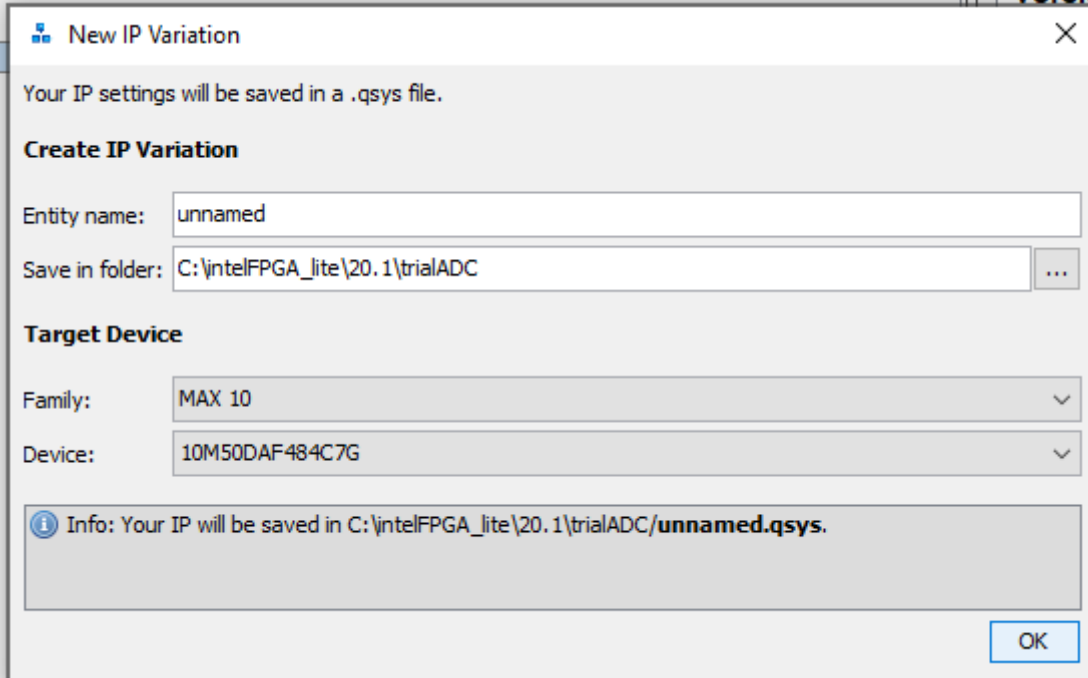
ADC with DE- series boards

13) Double click on the IP catalog on
"ADC Controller for DE-series Boards"



ADC with DE- series boards

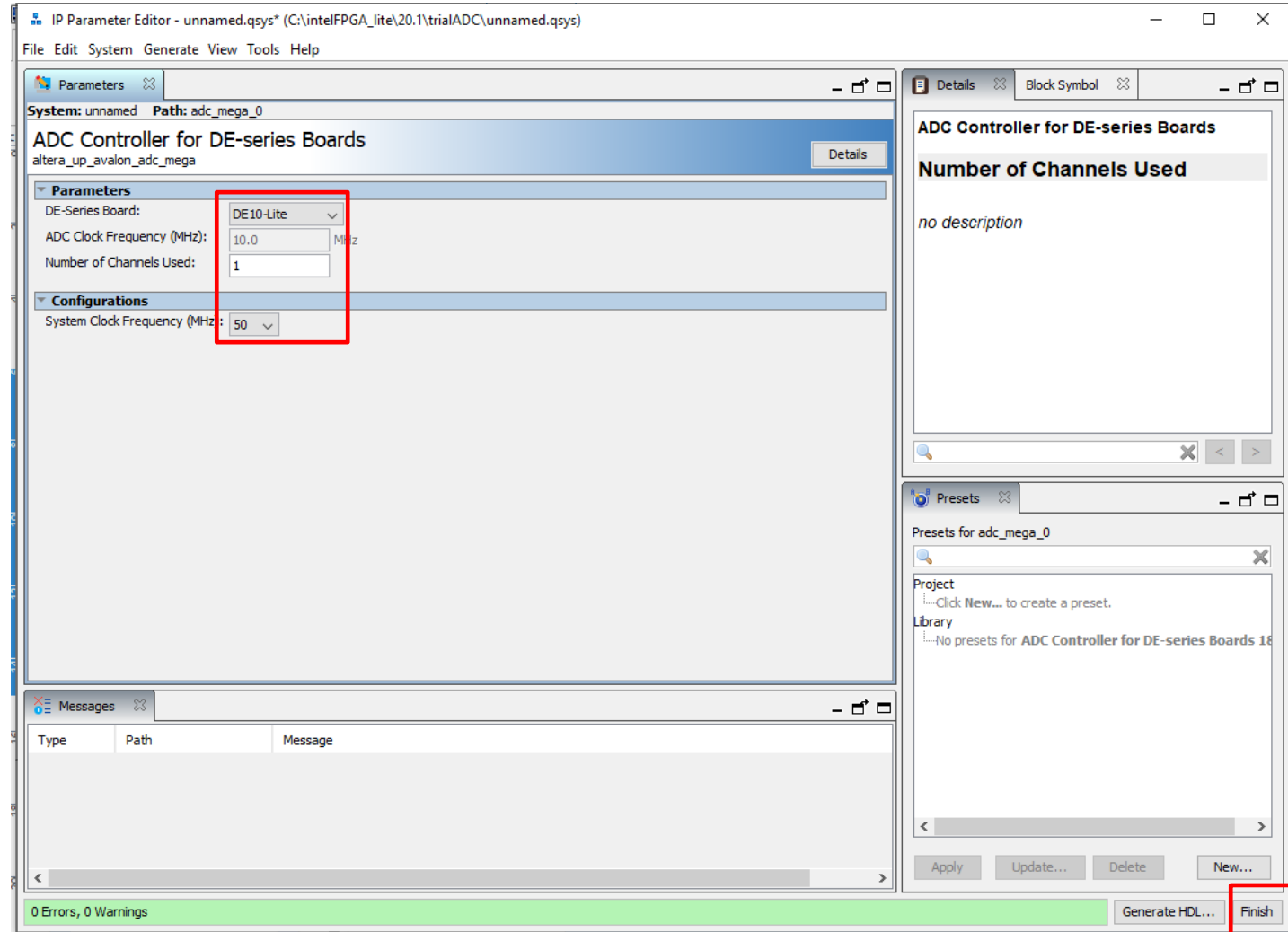
14) Press OK when asked to save



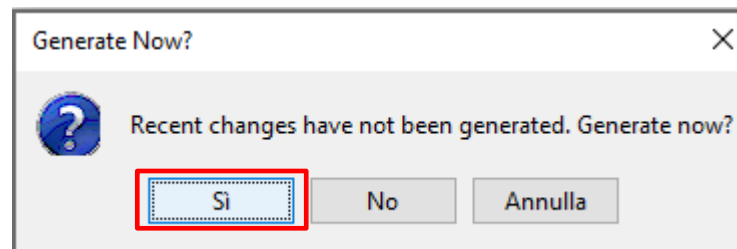
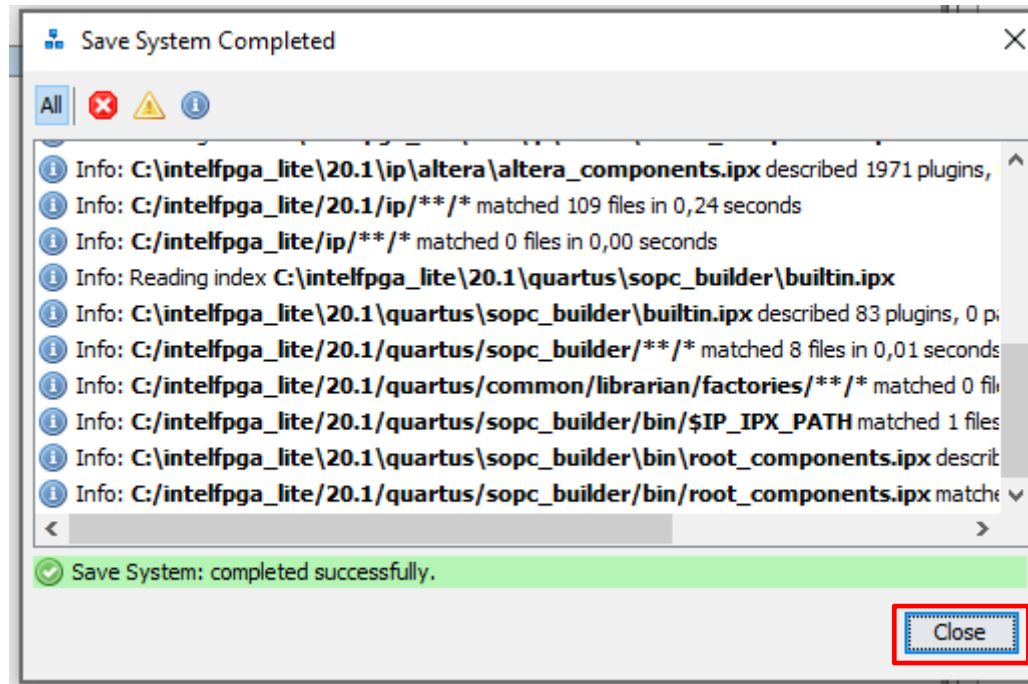
The screenshot shows a 'New IP Variation' dialog box with the following fields and options:

- Entity name:** unnamed
- Save in folder:** C:\intelFPGA_lite\20.1\trialADC
- Target Device:**
 - Family:** MAX 10
 - Device:** 10M50DAF484C7G
- Info:** Your IP will be saved in C:\intelFPGA_lite\20.1\trialADC\unnamed.qsys.
- Buttons:** OK

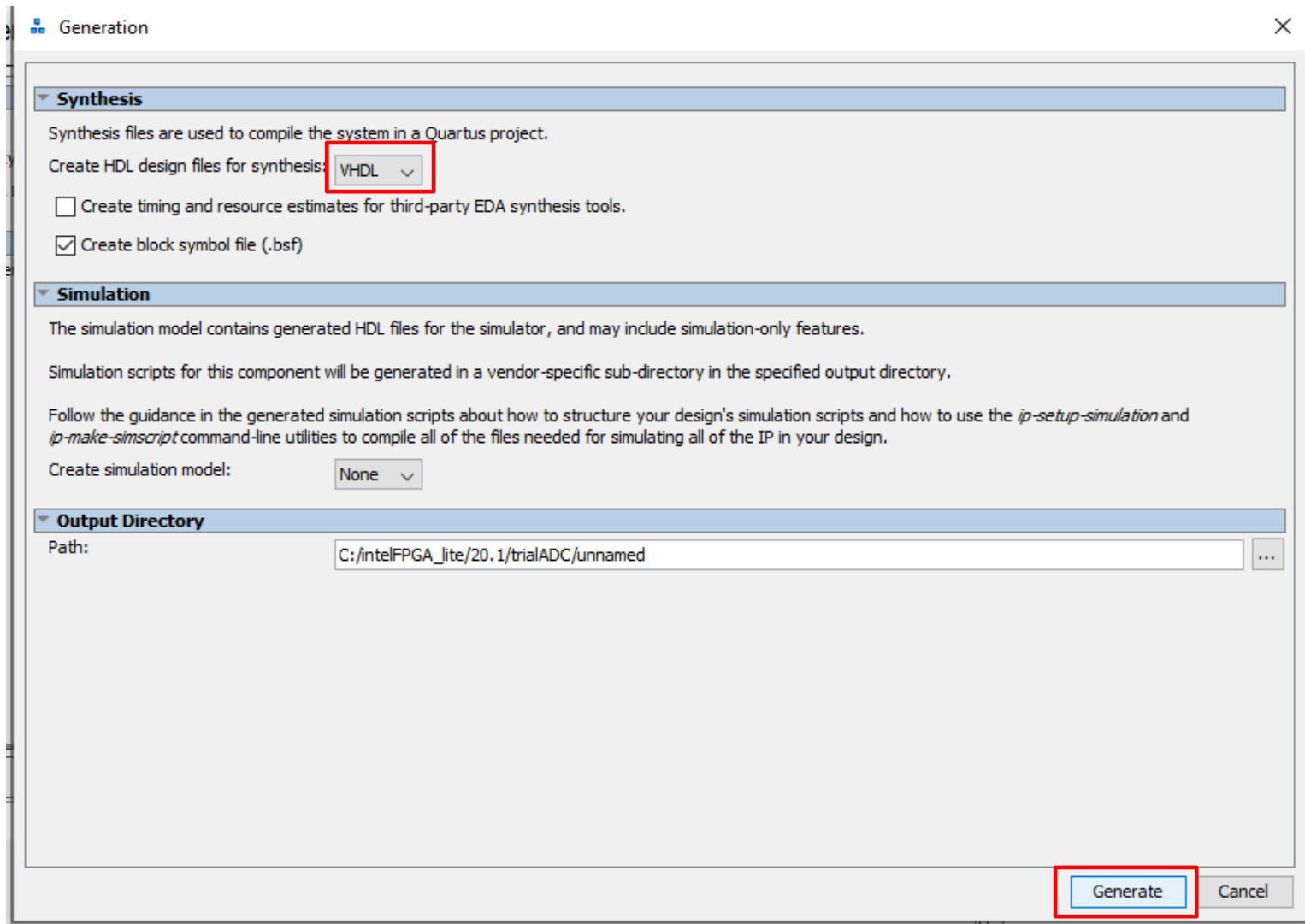
ADC with DE- series boards



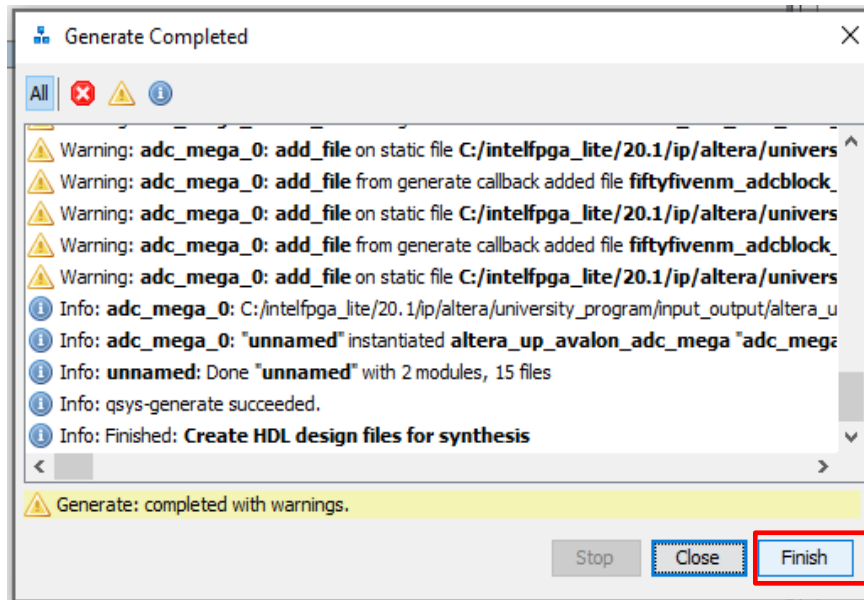
ADC with DE- series boards



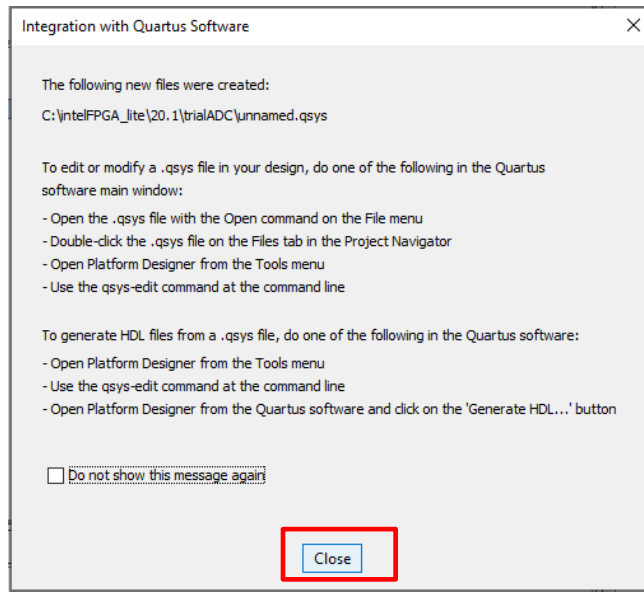
ADC with DE- series boards



ADC with DE- series boards

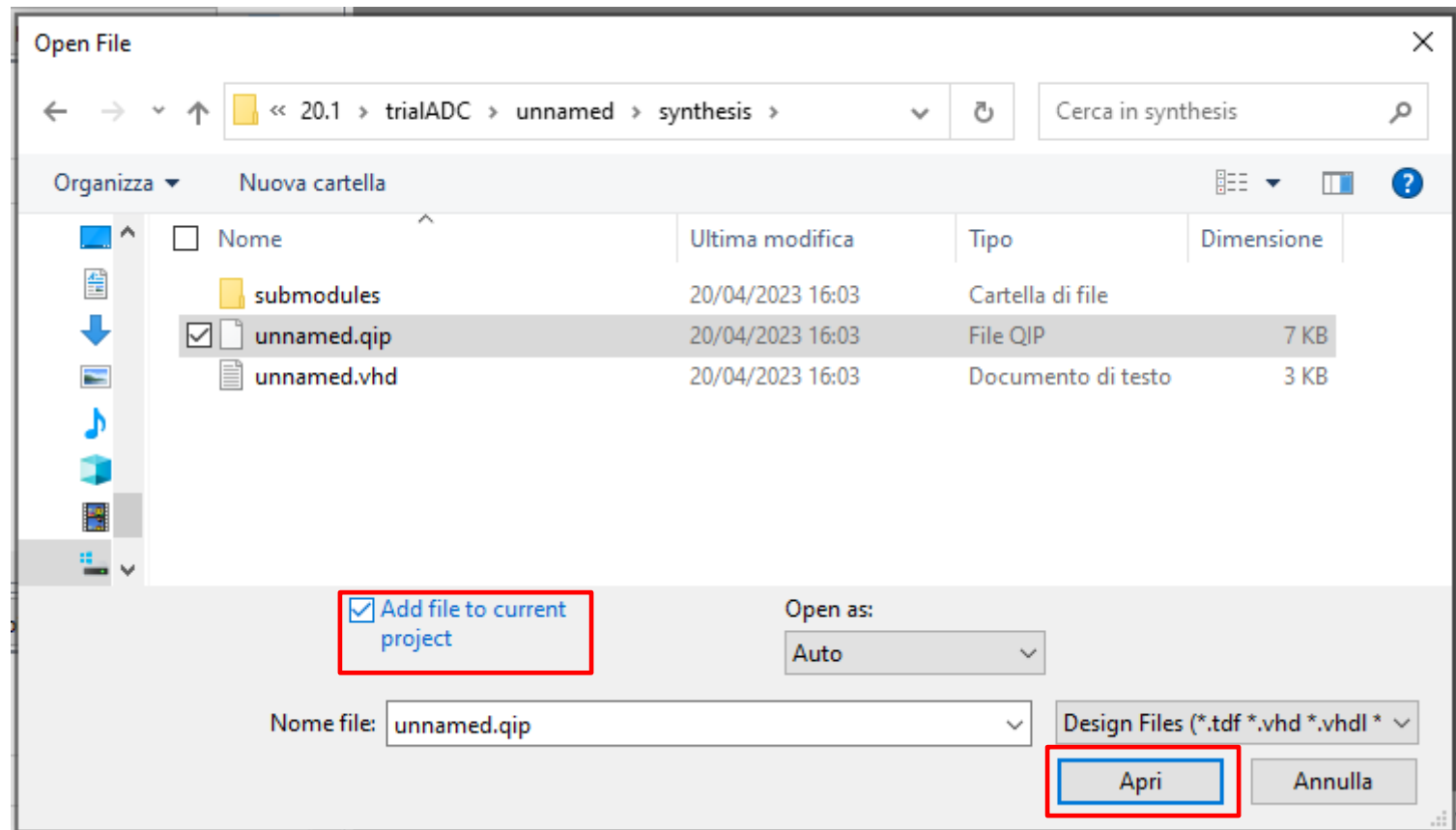


15) Click on Finish, Close and then Ok



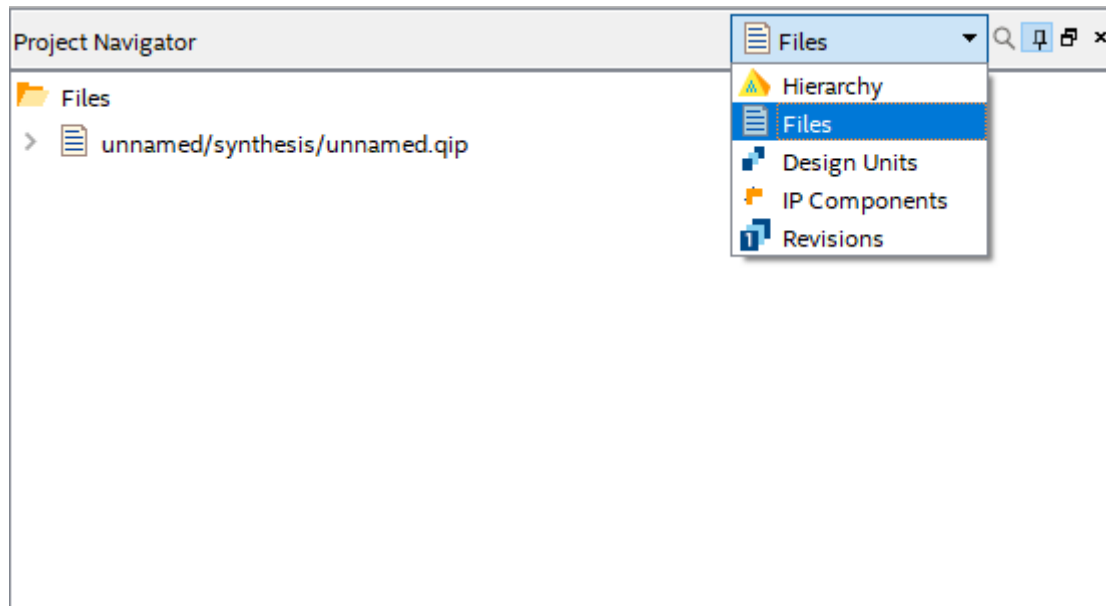
ADC with DE- series boards

16) Open the .qip file just created (...unnamed/synthesis/unnamed.qip)
ADD it to the project!



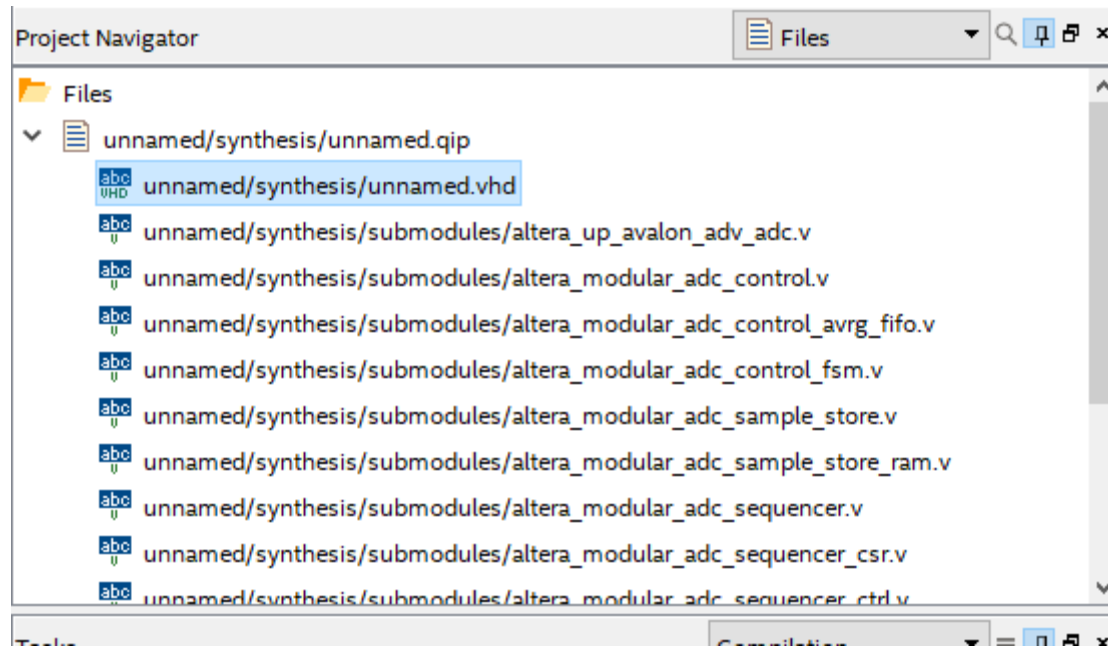
ADC with DE- series boards

17) Go On Files view in the Project Navigator



ADC with DE- series boards

18) Open the unnamed.vhd file



ADC with DE- series boards

19) Open the unnamed.vhd file

```
-- unnamed.vhd
-- Generated using ACDS version 20.1 720
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity unnamed is
port (
    CLOCK : in std_logic := '0'; -- clk.clk
    CH0 : out std_logic_vector(11 downto 0); -- readings.CH0
    CH1 : out std_logic_vector(11 downto 0); -- .CH1
    CH2 : out std_logic_vector(11 downto 0); -- .CH2
    CH3 : out std_logic_vector(11 downto 0); -- .CH3
    CH4 : out std_logic_vector(11 downto 0); -- .CH4
    CH5 : out std_logic_vector(11 downto 0); -- .CH5
    CH6 : out std_logic_vector(11 downto 0); -- .CH6
    CH7 : out std_logic_vector(11 downto 0); -- .CH7
    RESET : in std_logic := '0' -- reset.reset
);
end entity unnamed;
```

20) Now you can use it to place a component in your VHDL code