EEDG/CE 6370
Design and Analysis of Reconfigurable Systems
Homework 1
Design of Combinational Logic Designs using Schematic entry and Verilog/VHDL using Intel Quartus Prime

1. Laboratory Objectives

- Learn how to use a commercially available FPGA synthesis environment.
- Create a combinational circuit using graphical design entry (schematic entry) and RTL (Verilog or VHDL)
- Analyze the synthesis results and understand the FPGA resources consumed.
- Simulate the design generating a testbench and verify the correctness of the simulation.

2. Summary

This is a step-by-step tutorial for building a 1-bit full-adder using Quartus Prime Design Suite software that provides designers with the ability to generate digital circuits in different ways, e.g., schematic entry and/or using a hardware description language such as VHDL or Verilog. Quartus Prime also provides the ability to apply FPGA pin and timing constraints, analyze for errors and violations.

3. Pre-lab

- Review the lecture slides that covers the basic structure of an FPGA
- Download and install the tools specified in the tool requirements section (they are all free).

4. Tool Requirements

- Quartus Prime 22.1 Lite Edition (freely available online: Google Intel Quartus Prime lite download).
  You only need to install the **Cyclone V FPGA libraries** (our FPGA board has a Cyclone V installed)
- Questa RTL simulator: Make sure to also install the free simulation tool Questa. You will need this to simulate and verify the correctness of the design.

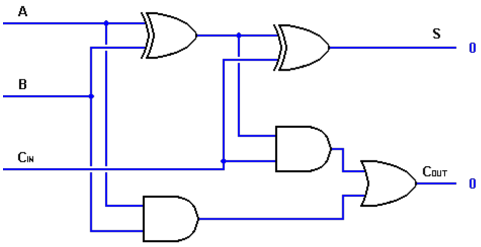
  Questa-Intel FPGA is free and is installed with Quartus Prime Lite if selected BUT you still need to go to Intel's license center and issue a license for it

  https://www.youtube.com/watch?v=F6FvXga4f1A

  You need to create an account and go the self-license center→ issue a license and add in windows the LM_LICENSE_FILE environment variable to point to the new license file issued.
  The main reason for this is that Questa is a third-party tool (Siemens). This license file needs to be renewed yearly.

5. Schematic Full Adder Design

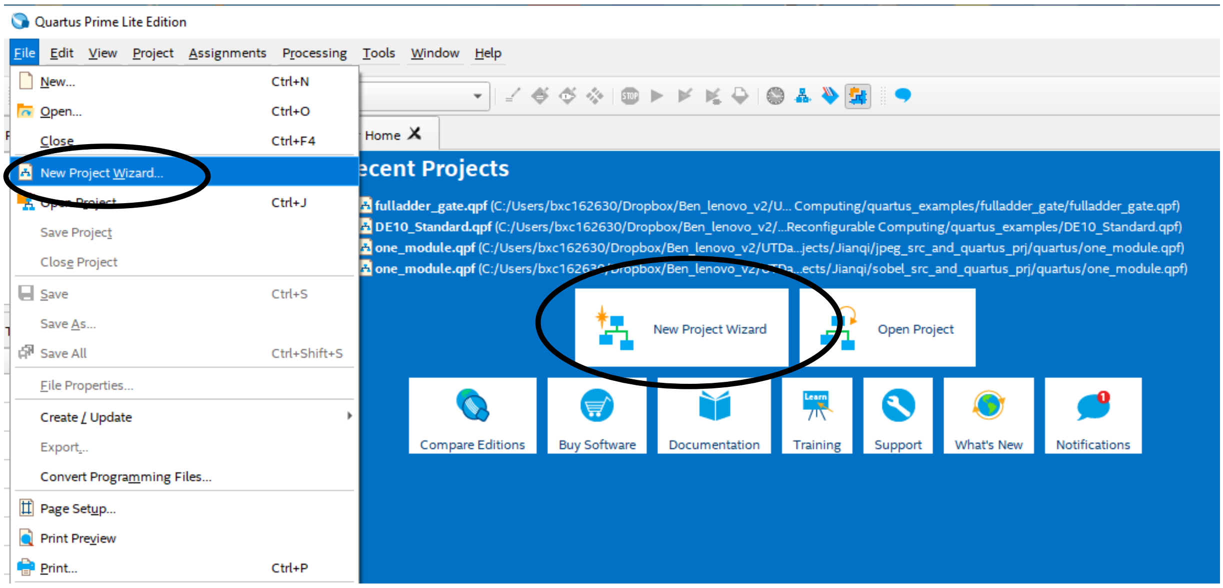
An example 1-bit half-adder block diagram and Boolean circuit are shown below with its truth table (to be completed at the answer sheet).



A	B	Cin	S	C
0	0	0	0	0
0	1	0	:	:
0	1	1		

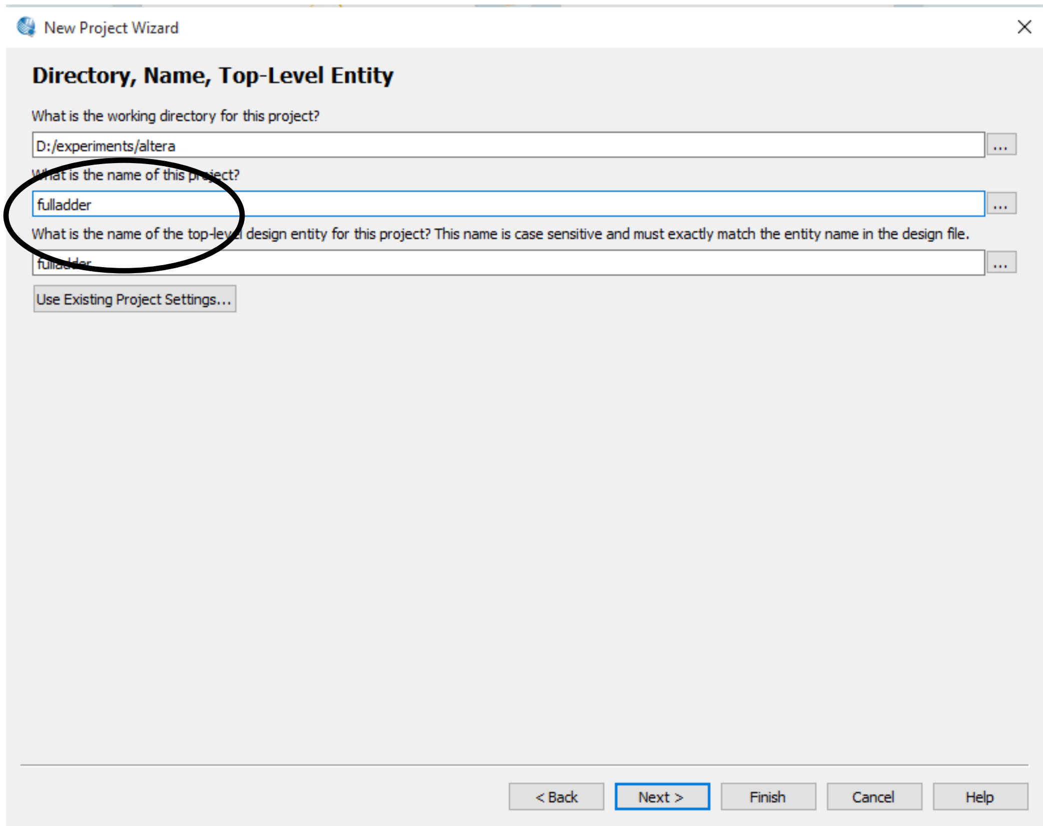
- Open Quartus Prime In case of Windows run the tools always as administrator (right click in icon)

- Create a new project.
  File ➔ New Project Wizard



fulladder_gate.qpf (C:/Users/bxc162630/Dropbox/Ben_lenovo_v2/U... Computing/quartus_examples/fulladder_gate/fulladder_gate.qpf)
DE10_Standard.qpf (C:/Users/bxc162630/Dropbox/Ben_lenovo_v2/...Reconfigurable Computing/quartus_examples/DE10_Standard/DE10_Standard.qpf)
one_module.qpf (C:/Users/bxc162630/Dropbox/Ben_lenovo_v2/UTDa...jects/Jianqi/jpeg_src_and_quartus_prj/quartus/one_module.qpf)
one_module.qpf (C:/Users/bxc162630/Dropbox/Ben_lenovo_v2/UTDa...ects/Jianqi/sobel_src_and_quartus_prj/quartus/one_module.qpf)

Name the project “fulladder”



Directory, Name, Top-Level Entity

What is the working directory for this project?

D:/experiments/altera

What is the name of this project?

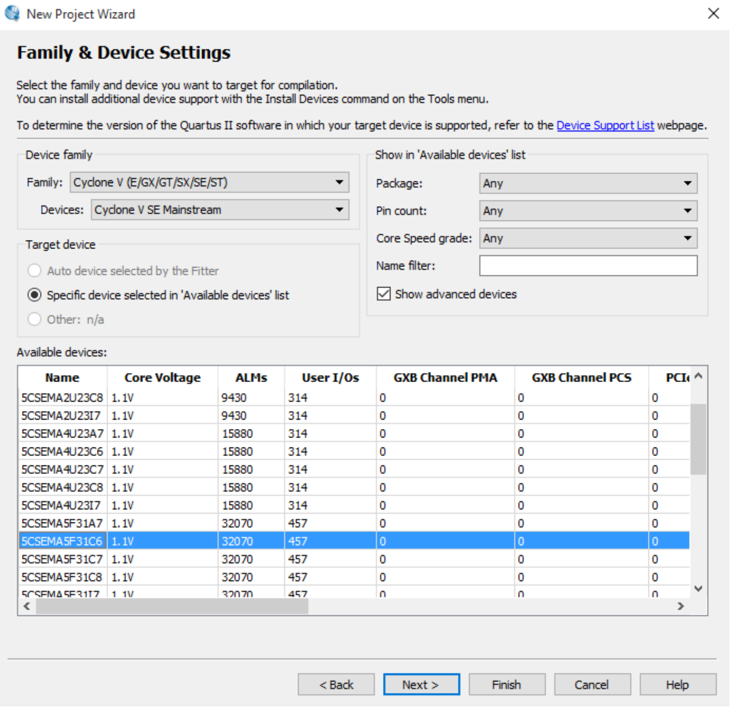
fulladder

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

fulladder

Click next → empty project →
Click next (do not add any files)

- Select Cyclone V 5CSEMA5F31C6 device.



Family & Device Settings

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

To determine the version of the Quartus II software in which your target device is supported, refer to the Device Support List webpage.

Device family

Family: Cyclone V (E/GX/GT/SX/SE/ST)

Devices: Cyclone V SE Mainstream

Target device

○ Auto device selected by the Fitter
● Specific device selected in 'Available devices' list
○ Other: n/a

Show in 'Available devices' list

Package: Any

Pin count: Any

Core Speed grade: Any

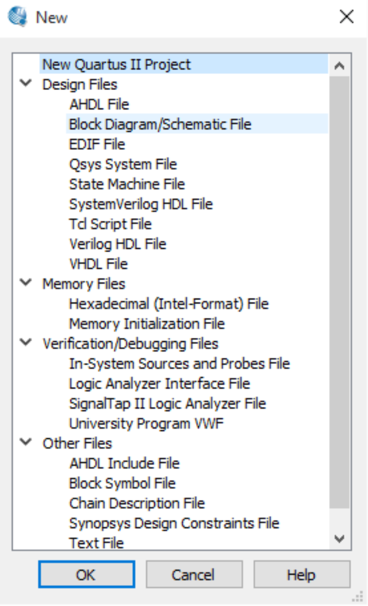
Name filter:

☑ Show advanced devices

Available devices:

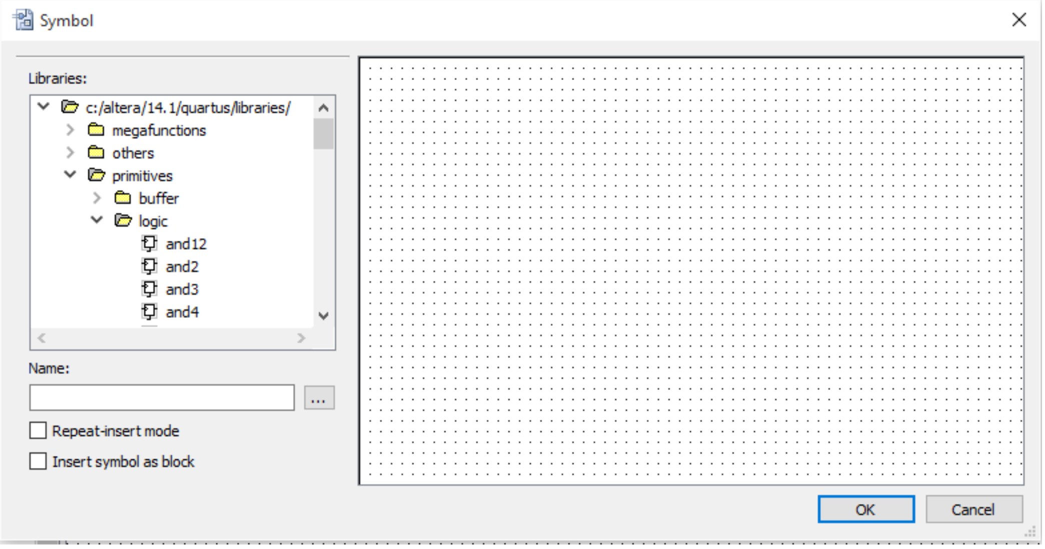
Name	Core Voltage	ALMs	User I/Os	GXB Channel PMA	GXB Channel PCS	PCIe
5CSEMA2U23C8	1.1V	9430	314	0	0	0
5CSEMA2U23I7	1.1V	9430	314	0	0	0
5CSEMA4U23A7	1.1V	15880	314	0	0	0
5CSEMA4U23C6	1.1V	15880	314	0	0	0
5CSEMA4U23C7	1.1V	15880	314	0	0	0
5CSEMA4U23C8	1.1V	15880	314	0	0	0
5CSEMA4U23I7	1.1V	15880	314	0	0	0
5CSEMA5F31A7	1.1V	32070	457	0	0	0
5CSEMA5F31C6	1.1V	32070	457	0	0	0
5CSEMA5F31C7	1.1V	32070	457	0	0	0
5CSEMA5F31C8	1.1V	32070	457	0	0	0
5CSEMA5F31I7	1.1V	32070	457	0	0	0

Click next and Finish.
New → Design Files → Block Diagram/Schematic File

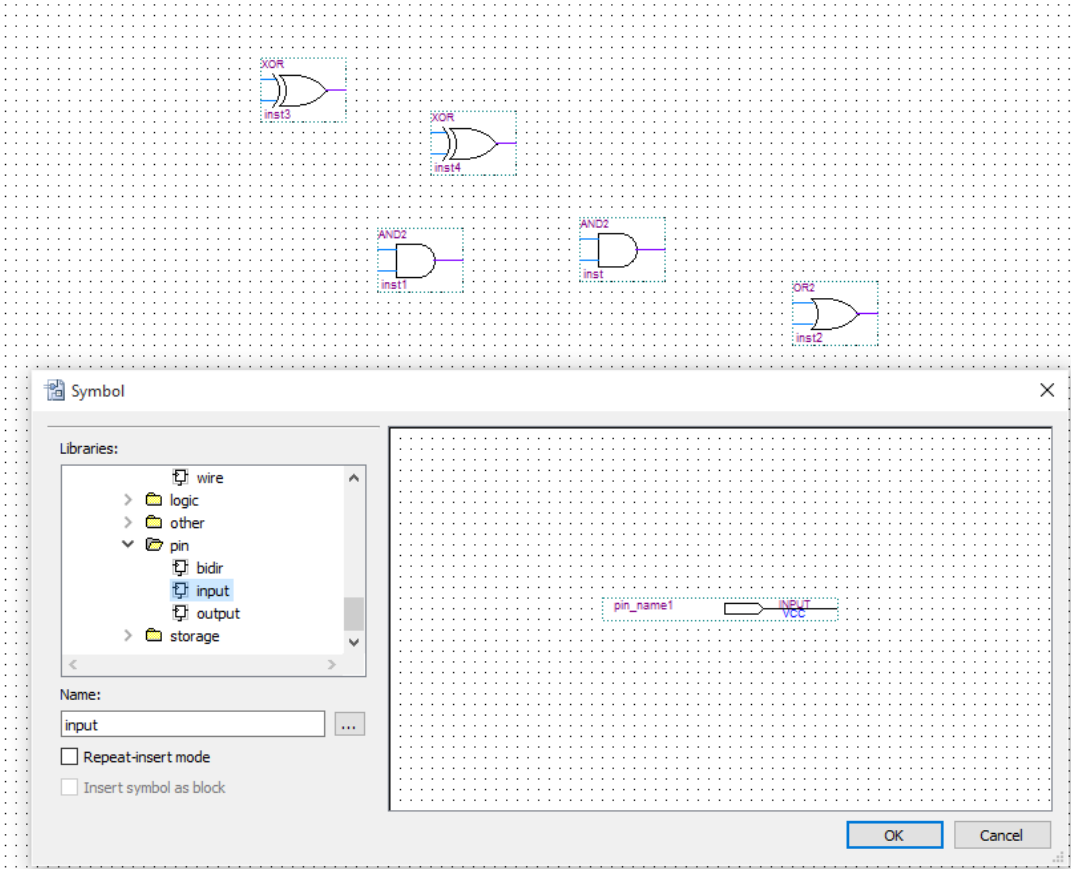


Save As → fulladder.bdf

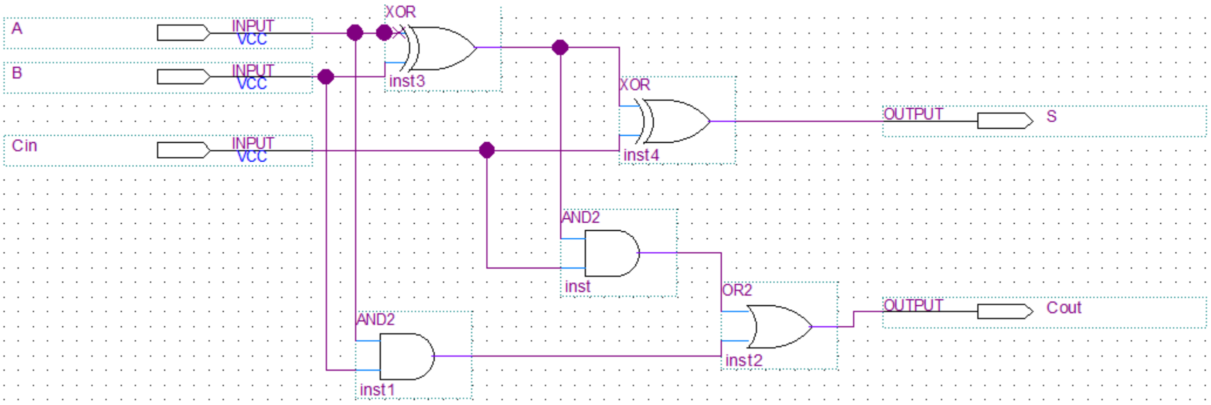
- Double click on drawing canvas.



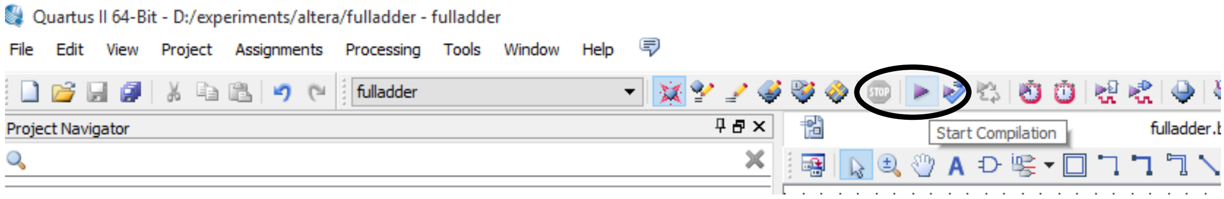
- Go to primitives and select and2, xor2, or2.
- Go to primitives→ pin and add 3 inputs and 2 outputs



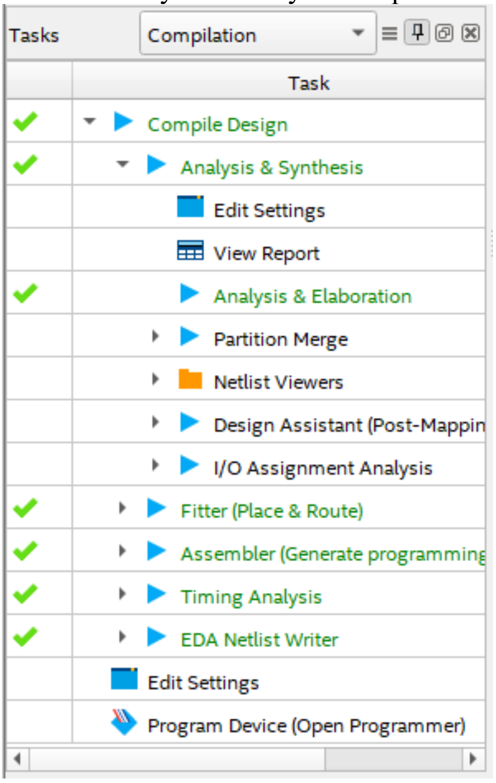
- Rename the input and output pins to A,B, Cin and S, Cout respectively by double-clicking on the pins just inserted
- Connect the components together by selecting the wire connection on the menu bar.



- Compile the design to make sure that there are no errors.



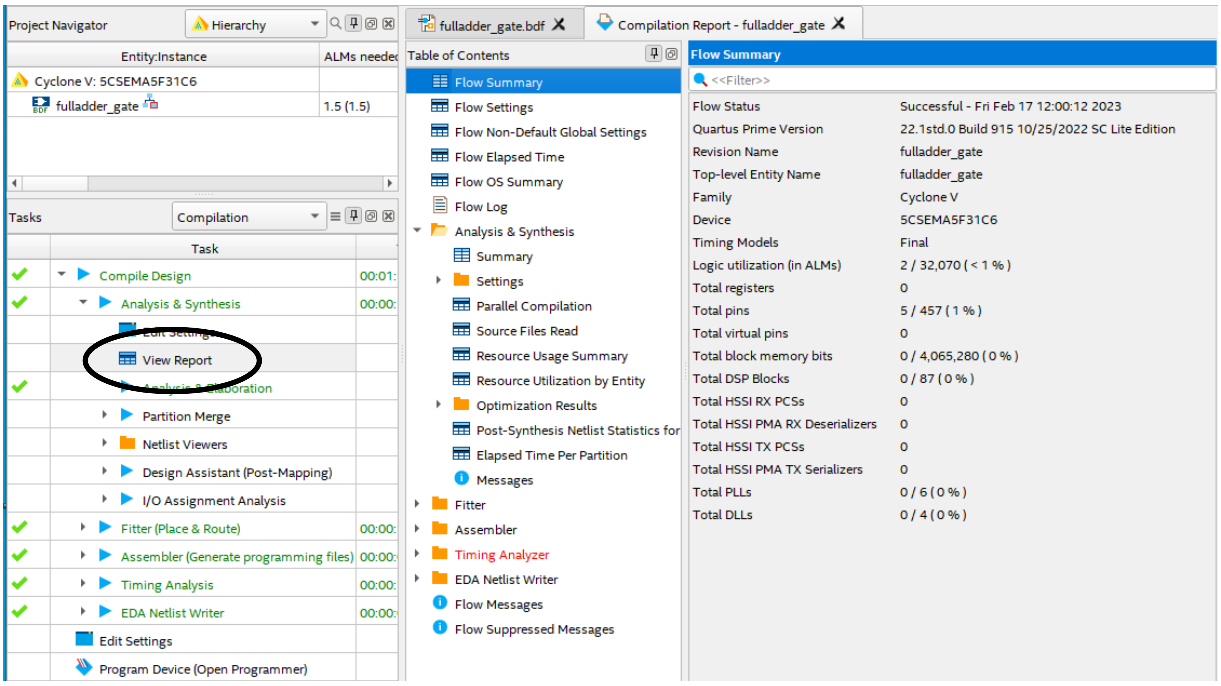
- Verify that the synthesis processes have finished successfully.



Tasks Compilation

	Task
✓	Compile Design
✓	Analysis & Synthesis
	Edit Settings
	View Report
✓	Analysis & Elaboration
	Partition Merge
	Netlist Viewers
	Design Assistant (Post-Mappin
	I/O Assignment Analysis
✓	Fitter (Place & Route)
✓	Assembler (Generate programming
✓	Timing Analysis
✓	EDA Netlist Writer
	Edit Settings
	Program Device (Open Programmer)

- View synthesis report → Fitter → Summary



Entity:Instance	ALMs needed
Cyclone V: 5CSEMA5F31C6	
fulladder_gate	1.5 (1.5)

Flow Summary

Flow Status	Successful - Fri Feb 17 12:00:12 2023
Quartus Prime Version	22.1std.0 Build 915 10/25/2022 SC Lite Edition
Revision Name	fulladder_gate
Top-level Entity Name	fulladder_gate
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	2 / 32,070 (< 1 %)
Total registers	0
Total pins	5 / 457 (1 %)
Total virtual pins	0
Total block memory bits	0 / 4,065,280 (0 %)
Total DSP Blocks	0 / 87 (0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 4 (0 %)

6. Verification – Simulation

Note1: You need to have installed Questa-Intel FPGA starter edition or have any other RTL simulator installed to run this.

Note 2: If Quartus Prime Lite has a bug when calling Questa from within Prime. Go to

"C:\intelFPGA_lite\22.1std\quartus\common\tcl\internal\nativelink" there is a file called qnativelinkflow.tcl

```
set questa_fse_directory "$quartus_path/../questa_fse"
if {[file isdirectory $questa_fse_directory]} {
  set questa_installation "$questa_fse_directory
```

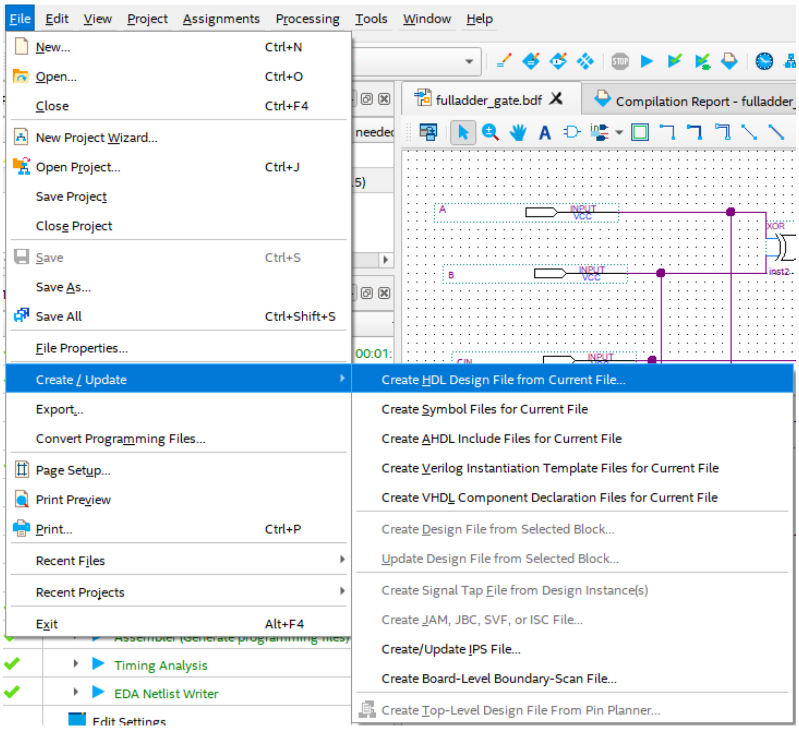
➔ Remove the " to

```
set questa_installation $questa_fse_directory
```

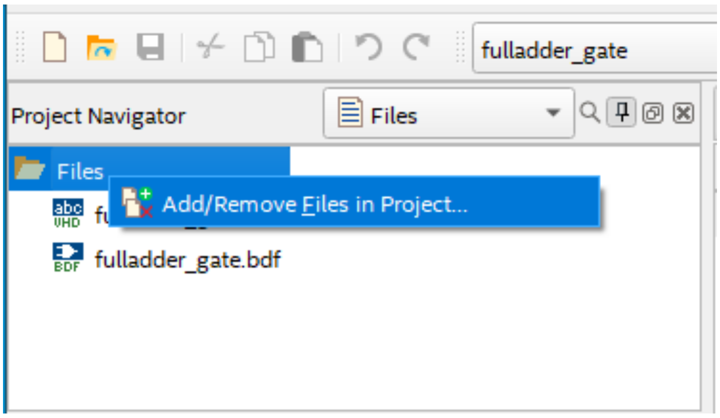
You can also run Questa directly from the installation folder. The default installation folder is:
C:\intelFPGA_lite\22.1std\questa_fse\win64

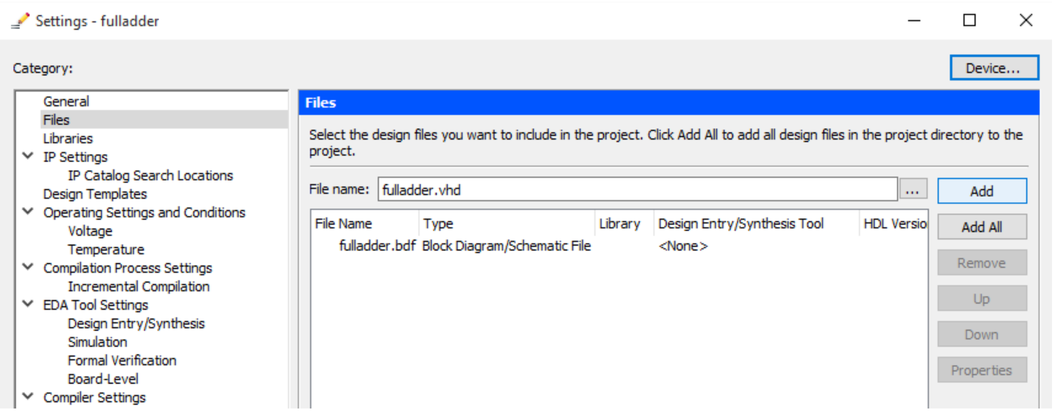
To avoid any issues run it as administrator.

- Create Verilog or VHDL file for the schematic file just generated → Open the schematic diagram of the full adder → File→Create/Update


File
Edit
View
Project
Assignments
Processing
Tools
Window
Help
New... Ctrl+N
Open... Ctrl+O
Close Ctrl+F4
New Project Wizard...
Open Project... Ctrl+J
Save Project
Close Project
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
File Properties...
Create / Update
Export...
Convert Programming Files...
Page Setup...
Print Preview
Print... Ctrl+P
Recent Files
Recent Projects
Exit Alt+F4
Create HDL Design File from Current File...
Create Symbol Files for Current File
Create AHDL Include Files for Current File
Create Verilog Instantiation Template Files for Current File
Create VHDL Component Declaration Files for Current File
Create Design File from Selected Block...
Update Design File from Selected Block...
Create Signal Tap File from Design Instance(s)
Create JAM, JBC, SVF, or ISC File...
Create/Update IPS File...
Create Board-Level Boundary-Scan File...
Create Top-Level Design File From Pin Planner...
fulladder_gate.bdf
Compilation Report - fulladder
Timing Analysis
EDA Netlist Writer
Edit Settings
A
B
CIN
INPUT VCC
XOR
inst2

- Add newly generated fulladder.vhd/fulladder.v file to project → Project Navigator→Files →Add/Remove files





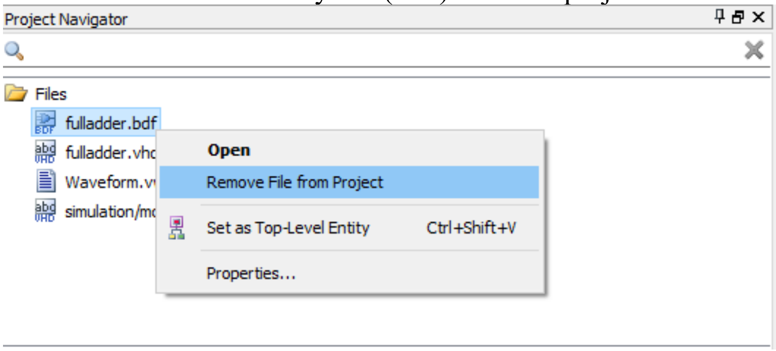
Select the design files you want to include in the project. Click Add All to add all design files in the project directory to the project.

File name: fulladder.vhd

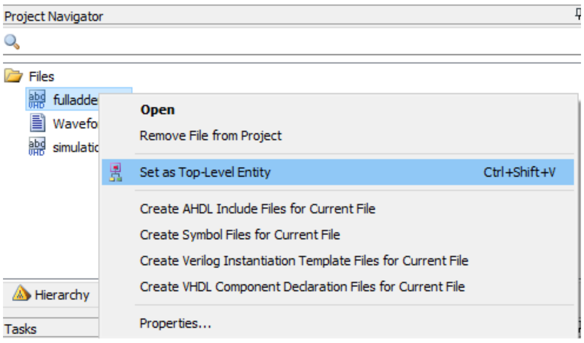
File Name	Type	Library	Design Entry/Synthesis Tool	HDL Versio
fulladder.bdf	Block Diagram/Schematic File		<None>	

- Quartus Prime does not allow the automatic generation of testbenches directly from a schematic entry file, hence the testbench must be generated for the newly imported Verilog/VHDL file. Also because both files have the same "entity" (module name)

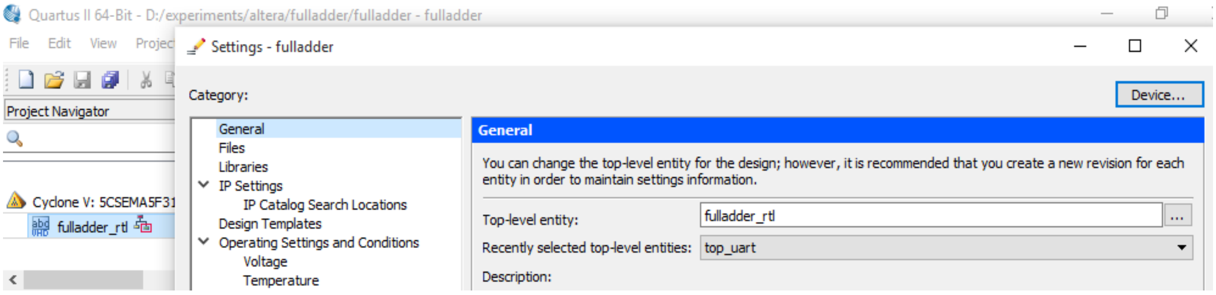
Deleted the schematic entry file (.bdf) from the project.



- Re-compile the design setting the Verilog/VHDL file as top entity.


Project Navigator
Files
fulladde
Wavefo
simulatic
Hierarchy
Tasks
Open
Remove File from Project
Set as Top-Level Entity Ctrl+Shift+V
Create AHDL Include Files for Current File
Create Symbol Files for Current File
Create Verilog Instantiation Template Files for Current File
Create VHDL Component Declaration Files for Current File
Properties...

- By default, Quartus assumes that the filename is the same as the top entity. If not, it needs to be set manually. For this
Project Navigator →Hierarchy → right click on Verilog/VHDL file → Settings



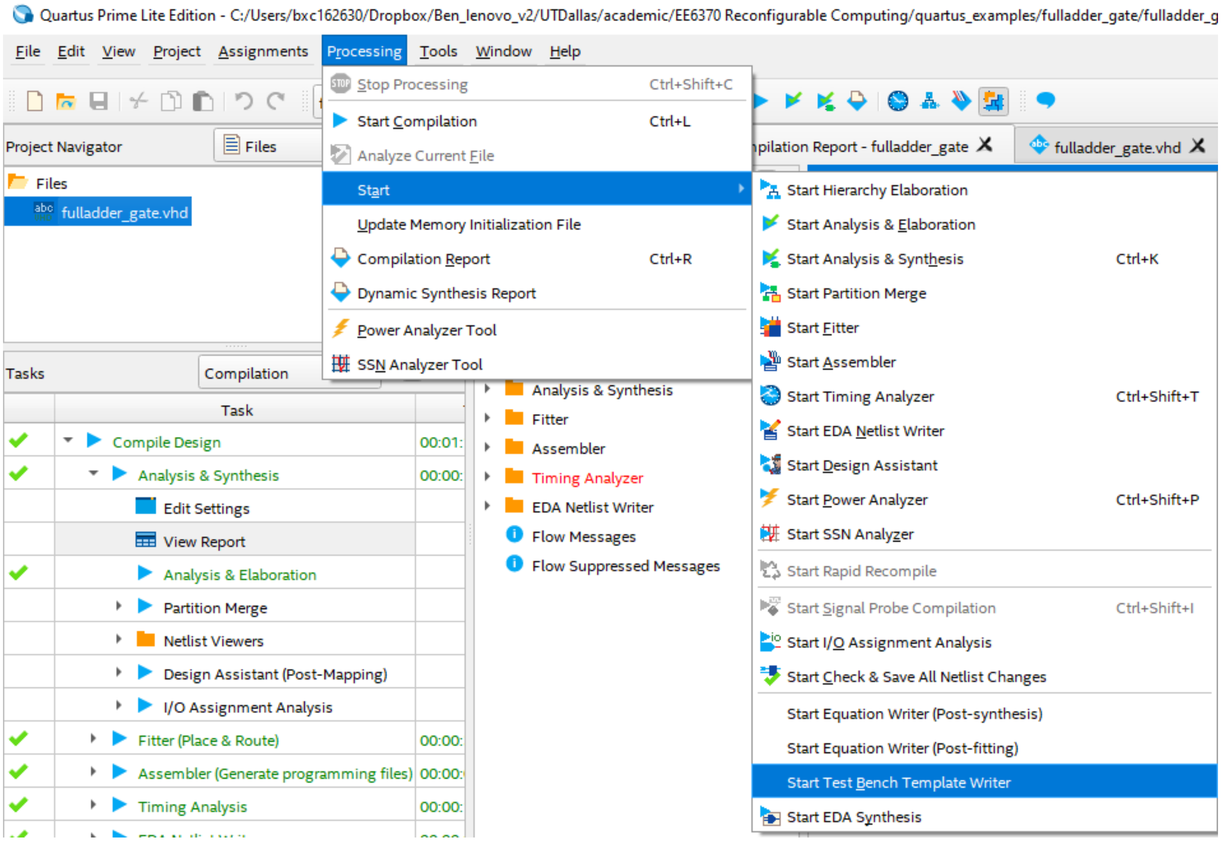
You can change the top-level entity for the design; however, it is recommended that you create a new revision for each entity in order to maintain settings information.

Top-level entity: fulladder_rtl

Recently selected top-level entities: top_uart

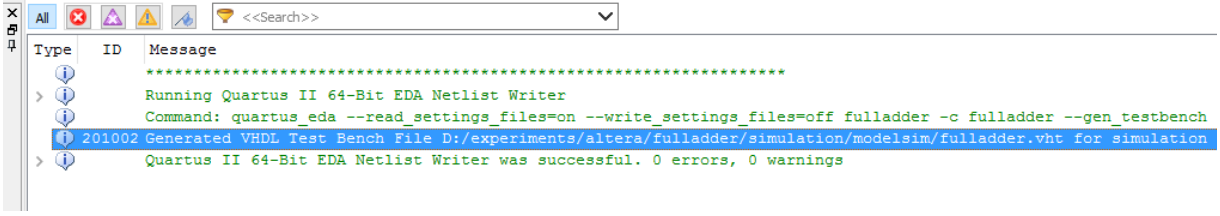
Description:

- Create a testbench template: Processing →Start→Start Test Bench Template Writer


Quartus Prime Lite Edition - C:/Users/bxc162630/Dropbox/Ben_lenovo_v2/UTDallas/academic/EE6370 Reconfigurable Computing/quartus_examples/fulladder_gate/fulladder_g

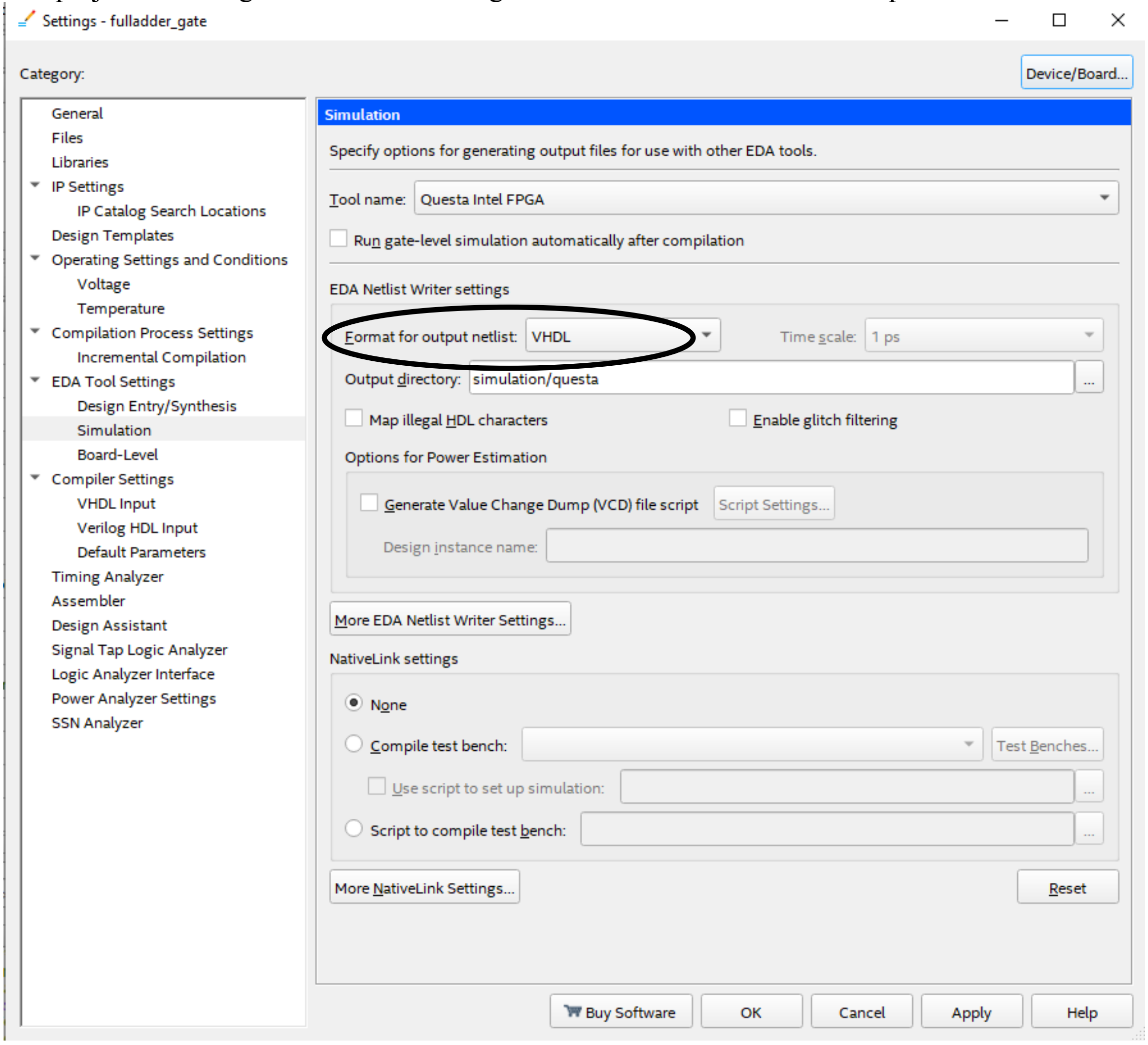
File Edit View Project Assignments Processing Tools Window Help
Project Navigator
Files
fulladder_gate.vhd
Tasks
Compilation
Task
Compile Design 00:01:
Analysis & Synthesis 00:00:
Edit Settings
View Report
Analysis & Elaboration
Partition Merge
Netlist Viewers
Design Assistant (Post-Mapping)
I/O Assignment Analysis
Fitter (Place & Route) 00:00:
Assembler (Generate programming files) 00:00:
Timing Analysis 00:00:
Stop Processing Ctrl+Shift+C
Start Compilation Ctrl+L
Analyze Current File
Start
Update Memory Initialization File
Compilation Report Ctrl+R
Dynamic Synthesis Report
Power Analyzer Tool
SSN Analyzer Tool
pilation Report - fulladder_gate
fulladder_gate.vhd
Analysis & Synthesis
Fitter
Assembler
Timing Analyzer
EDA Netlist Writer
Flow Messages
Flow Suppressed Messages
Start Hierarchy Elaboration
Start Analysis & Elaboration
Start Analysis & Synthesis Ctrl+K
Start Partition Merge
Start Fitter
Start Assembler
Start Timing Analyzer Ctrl+Shift+T
Start EDA Netlist Writer
Start Design Assistant
Start Power Analyzer Ctrl+Shift+P
Start SSN Analyzer
Start Rapid Recompile
Start Signal Probe Compilation Ctrl+Shift+I
Start I/O Assignment Analysis
Start Check & Save All Netlist Changes
Start Equation Writer (Post-synthesis)
Start Equation Writer (Post-fitting)
Start Test Bench Template Writer
Start EDA Synthesis

- The console window illustrates where the testbench has been generated (project_folder/simulation/questa/)



```
     *******************************************************************
     Running Quartus II 64-Bit EDA Netlist Writer
     Command: quartus_eda --read_settings_files=on --write_settings_files=off fulladder -c fulladder --gen_testbench
201002 Generated VHDL Test Bench File D:/experiments/altera/fulladder/simulation/modelsim/fulladder.vht for simulation
     Quartus II 64-Bit EDA Netlist Writer was successful. 0 errors, 0 warnings
```

Note: By default Quartus will generate a Verilog Testbench. If you are using VHDL, you need to specify it in project→ Settings→EDA Tool Settings →Simulation → Format for output netlist →VHDL



Specify options for generating output files for use with other EDA tools.

Tool name: Questa Intel FPGA

□ Run gate-level simulation automatically after compilation

EDA Netlist Writer settings

Format for output netlist: VHDL   Time scale: 1 ps

Output directory: simulation/questa

□ Map illegal HDL characters   □ Enable glitch filtering

Options for Power Estimation

□ Generate Value Change Dump (VCD) file script

Design instance name:

NativeLink settings

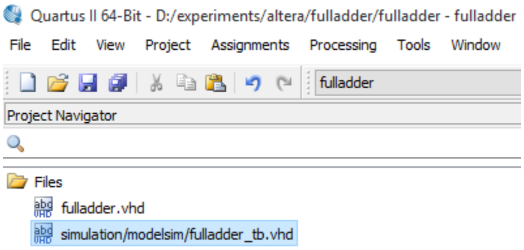
● None

○ Compile test bench:

□ Use script to set up simulation:

○ Script to compile test bench:

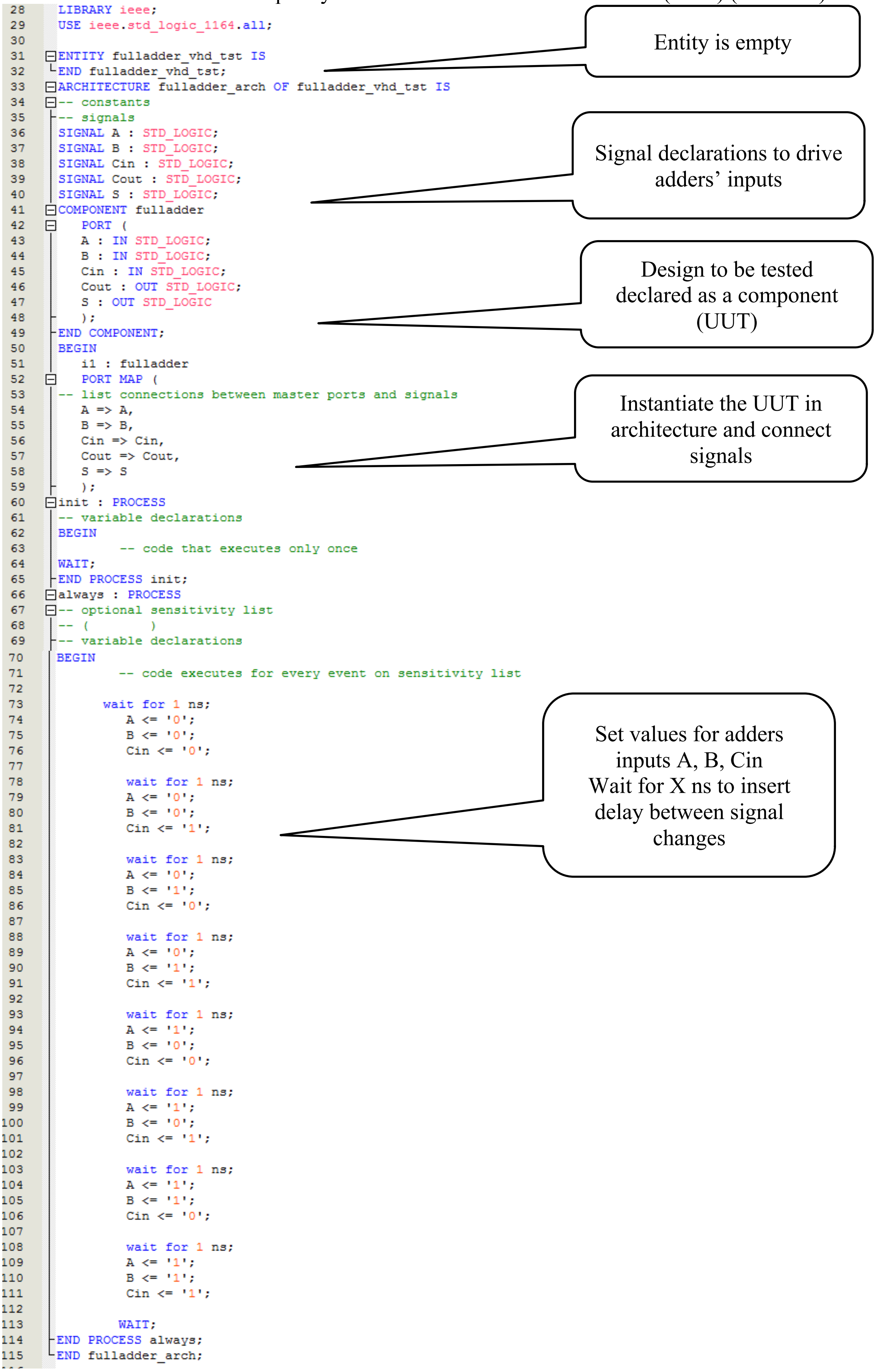
- Rename the Testbench file just generated and add to the project.
  VHDL case: "fulladder.vht" to "fulladder_tb.vhd"
  Verilog case: "fulladder.vt" to "fulladder.v"



fulladder.vhd

simulation/modelsim/fulladder_tb.vhd

- Edit the testbench to specify test vectors for the Unit Under Test (UUT) (fulladder).



```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY fulladder_vhd_tst IS
END fulladder_vhd_tst;
ARCHITECTURE fulladder_arch OF fulladder_vhd_tst IS
-- constants
-- signals
SIGNAL A : STD_LOGIC;
SIGNAL B : STD_LOGIC;
SIGNAL Cin : STD_LOGIC;
SIGNAL Cout : STD_LOGIC;
SIGNAL S : STD_LOGIC;
COMPONENT fulladder
   PORT (
   A : IN STD_LOGIC;
   B : IN STD_LOGIC;
   Cin : IN STD_LOGIC;
   Cout : OUT STD_LOGIC;
   S : OUT STD_LOGIC
   );
END COMPONENT;
BEGIN
   i1 : fulladder
   PORT MAP (
-- list connections between master ports and signals
   A => A,
   B => B,
   Cin => Cin,
   Cout => Cout,
   S => S
   );
init : PROCESS
-- variable declarations
BEGIN
        -- code that executes only once
WAIT;
END PROCESS init;
always : PROCESS
-- optional sensitivity list
-- (        )
-- variable declarations
BEGIN
        -- code executes for every event on sensitivity list

      wait for 1 ns;
         A <= '0';
         B <= '0';
         Cin <= '0';

         wait for 1 ns;
         A <= '0';
         B <= '0';
         Cin <= '1';

         wait for 1 ns;
         A <= '0';
         B <= '1';
         Cin <= '0';

         wait for 1 ns;
         A <= '0';
         B <= '1';
         Cin <= '1';

         wait for 1 ns;
         A <= '1';
         B <= '0';
         Cin <= '0';

         wait for 1 ns;
         A <= '1';
         B <= '0';
         Cin <= '1';

         wait for 1 ns;
         A <= '1';
         B <= '1';
         Cin <= '0';

         wait for 1 ns;
         A <= '1';
         B <= '1';
         Cin <= '1';

        WAIT;
END PROCESS always;
END fulladder_arch;
```

Entity is empty

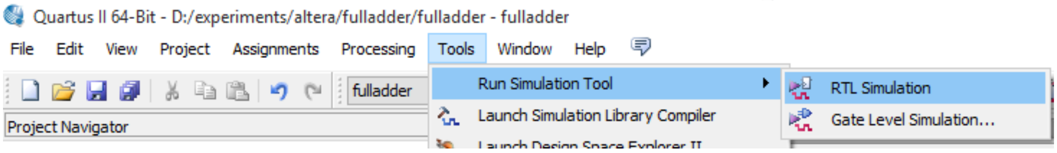
Signal declarations to drive adders' inputs

Design to be tested declared as a component (UUT)

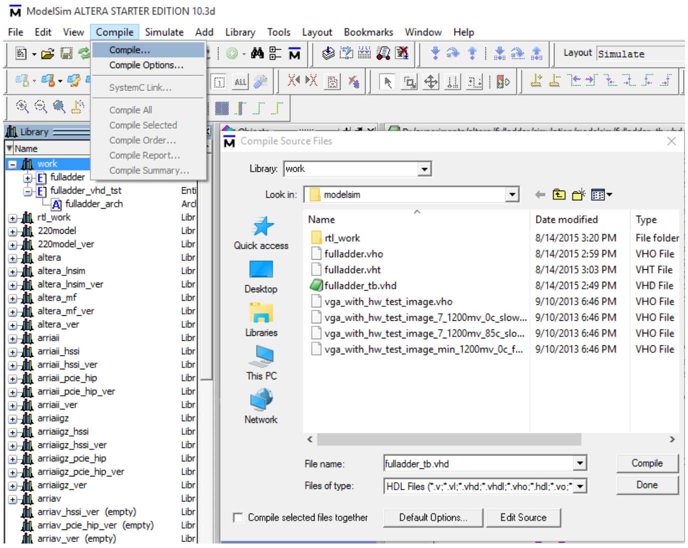
Instantiate the UUT in architecture and connect signals

Set values for adders inputs A, B, Cin
Wait for X ns to insert delay between signal changes

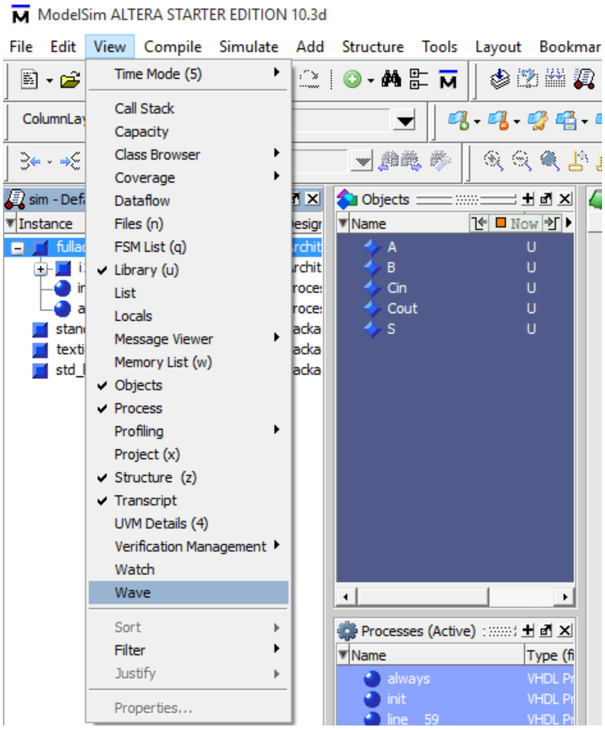
- Run a simulation: Tools→ Run Simulation Tool→ RTL Simulation. Make sure you have installed the Questa simulator when you installed Quartus Prime.


Quartus II 64-Bit - D:/experiments/altera/fulladder/fulladder - fulladder
File
Edit
View
Project
Assignments
Processing
Tools
Window
Help
fulladder
Project Navigator
Run Simulation Tool
Launch Simulation Library Compiler
Launch Design Space Explorer II
RTL Simulation
Gate Level Simulation...

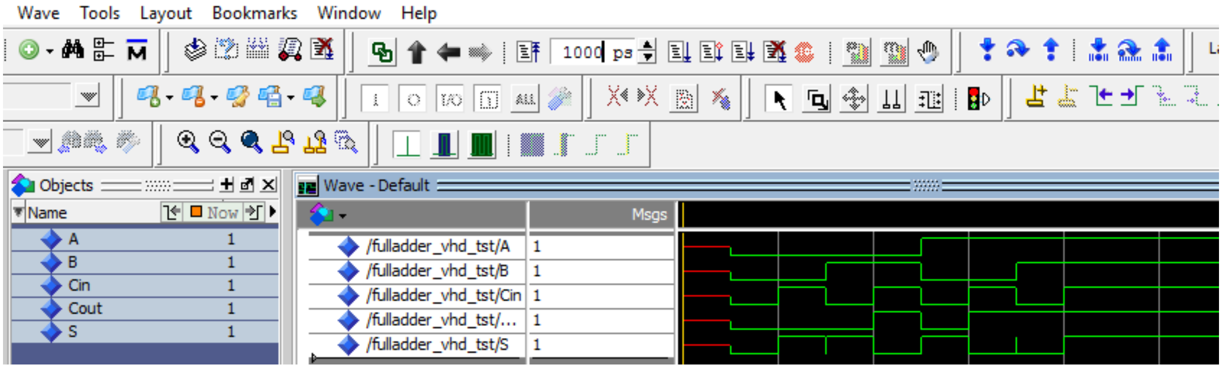
Questa simulator should start automatically.
Compile testbench : Compile → Select fulladder_tb.vhd



- This will create an entry in the work library.
- Double-click on work→fulladder_vhd_tst in the Library pane.
- Open the waveform viewer: View →Wave



- Select the signals to be displayed and drag to Waveform.
- Set the simulation step to 1ns
- Start simulation.



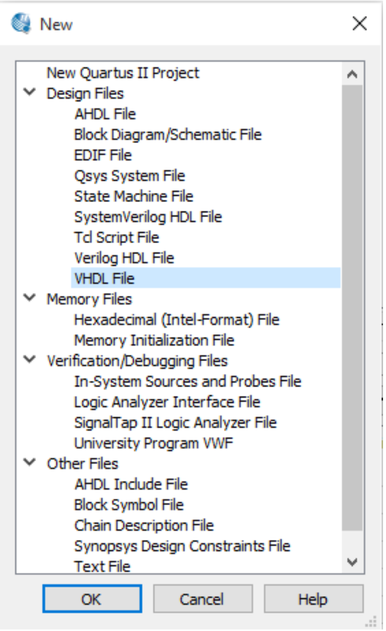
Name	Now
A	1
B	1
Cin	1
Cout	1
S	1

	Msgs
/fulladder_vhd_tst/A	1
/fulladder_vhd_tst/B	1
/fulladder_vhd_tst/Cin	1
/fulladder_vhd_tst/...	1
/fulladder_vhd_tst/S	1

7. Design and Verification of 1-bit adder in RTL (Verilog or VHDL)

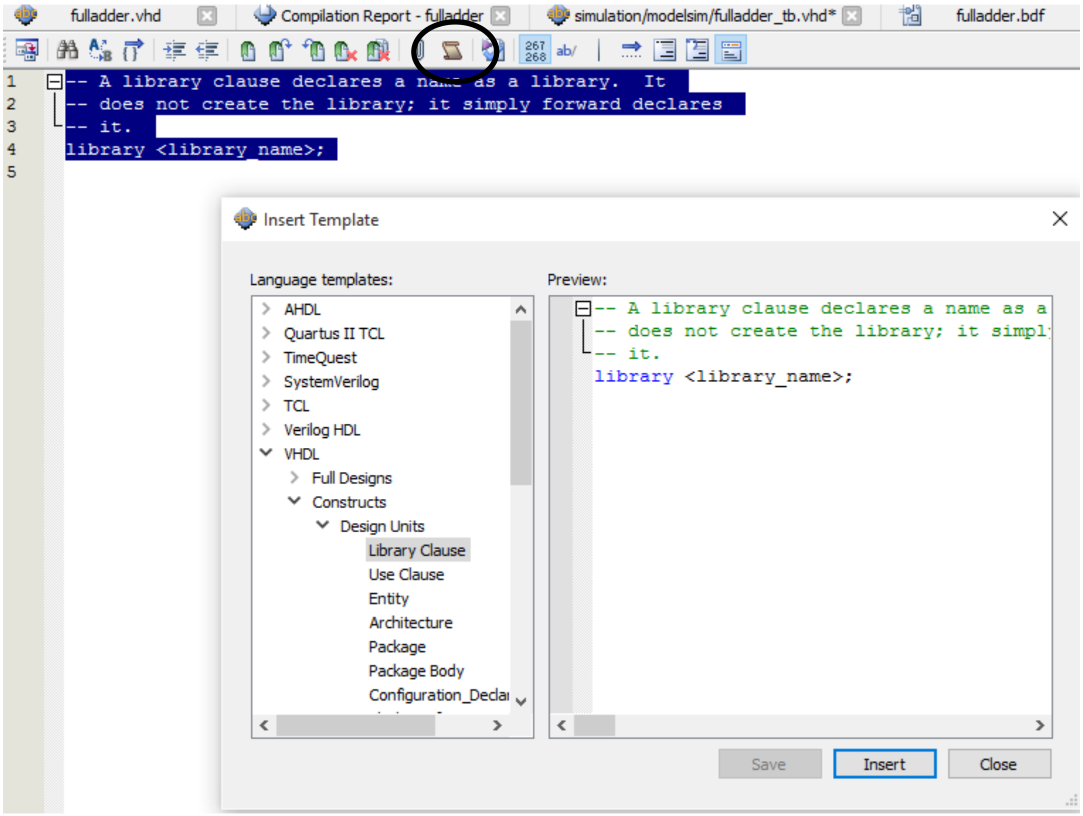
Re-do the same exercise creating an RTL description (Verilog or VHDL) of the same 1-bit adder

- 1-bit Full-adder in VHDL/Verilog
- File new → VHDL/Verilog



Specify file name "fulladder_rtl.vhdl/fulladder_rtl.v" (NOTE: The name has to be different from the schematic entry design)

- You can use the Verilog /VHDL template generator to insert the main Verilog/VHDL program structure: Library clause, Use clause, Entity and Architecture.



```
-- A library clause declares a name as a library.  It
-- does not create the library; it simply forward declares
-- it.
library <library_name>;
```

```
-- A library clause declares a name as a
-- does not create the library; it simpl
-- it.
library <library_name>;
```

Edit source code to create the full adder in Verilog or VHDL (VHDL shown here)

```
ENTITY fulladder_vhdl  IS
        PORT (
                A: in std_logic;
                B: in std_logic;
                Cin: in std_logic;
                Sum: out std_logic;
                Cout: out std_logic);
END fulladder_vhdl;

ARCHITECTURE behav OF fulladder_vhdl IS
BEGIN
--fill out the architecture body
-- Missing part for S, e.g. S <=
--Missing part of Cout, e.g. Cout <=


END behav;
```

- Synthesize the design and look at the generated report files.

UT Dallas Honor Code

UT Dallas values academic integrity. Therefore, all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures.

Group assignments must be completed solely by the members of the group. Cross-group work is not allowed. Moreover, similar assignments have been offered before at UT Dallas and other universities. Any use of information from previous assignments is prohibited. The tutorials are to be taken individually. Failure to respect this rule constitutes dishonesty and is a direct violation of the University Honour Code.

[END]