



## 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

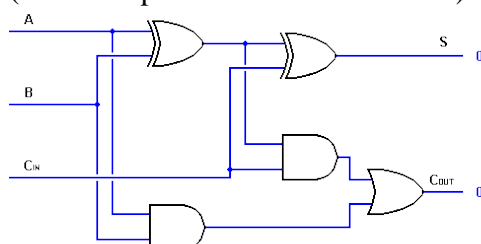
<https://www.youtube.com/watch?v=F6FvXga4f1A>

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).



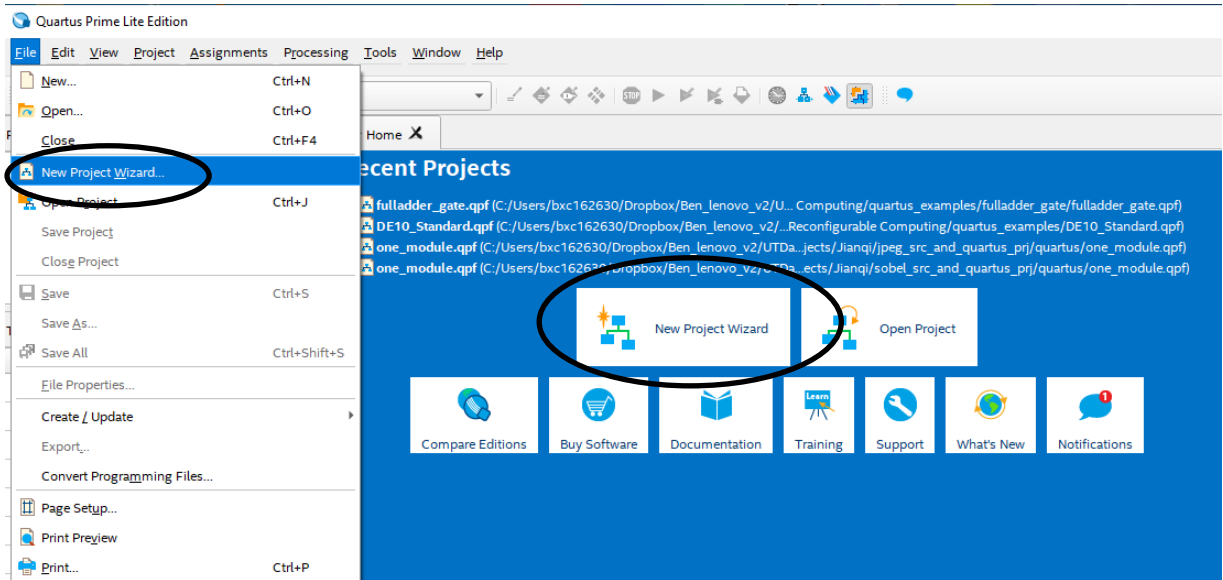
A	B	Cin	S	C
0	0	0	0	0
0	1	0	:	:
0	1	1		

- Open Quartus Prime (right click in icon)

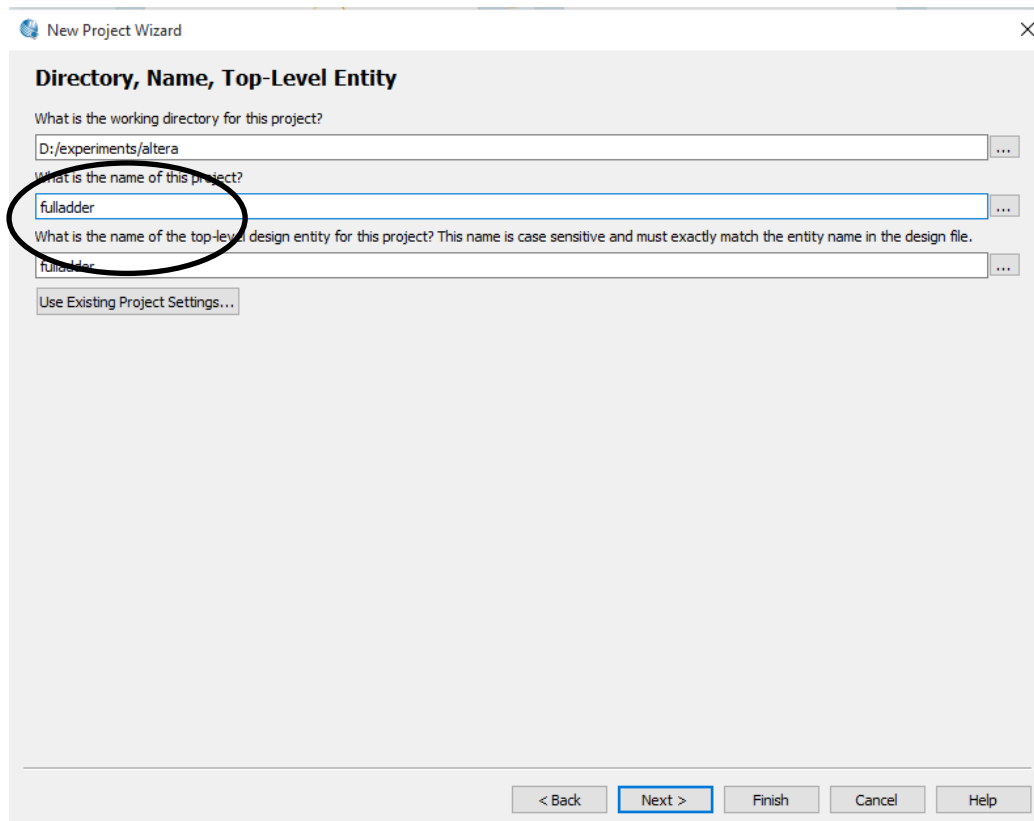


In case of Windows run the tools always as administrator (right

- Create a new project.  
File → New Project Wizard



Name the project “fulladder”



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

- Select Cyclone V 5CSEMA5F31C6 device.

New Project Wizard

### 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

Show in 'Available devices' list

Package: Any

Pin count: Any

Core Speed grade: Any

Name filter:

☒ Show advanced devices

Target device

☐ Auto device selected by the Fitter

☒ Specific device selected in 'Available devices' list

☐ Other: n/a

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

< Back   Next >   Finish   Cancel   Help

Click next and Finish.

New → Design Files → Block Diagram/Schematic File

New

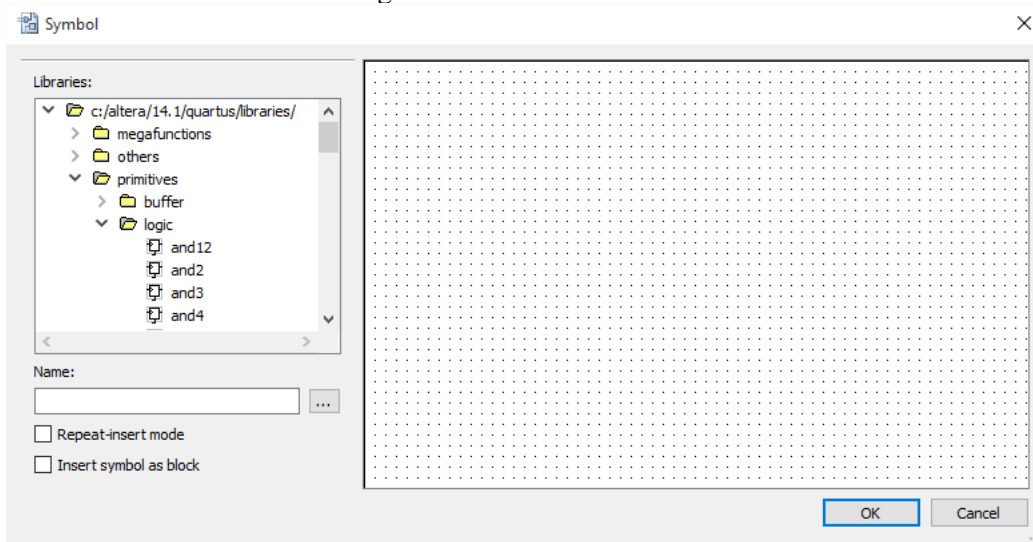
New Quartus II Project

- Design Files
  - AHDL File
  - Block Diagram/Schematic File
  - EDIF File
  - Qsys System File
  - State Machine File
  - SystemVerilog HDL File
  - Tcl Script File
  - Verilog HDL File
  - VHDL File
- Memory Files
  - Hexadecimal (Intel-Format) File
  - Memory Initialization File
- Verification/Debugging Files
  - In-System Sources and Probes File
  - Logic Analyzer Interface File
  - SignalTap II Logic Analyzer File
  - University Program VWF
- Other Files
  - AHDL Include File
  - Block Symbol File
  - Chain Description File
  - Synopsys Design Constraints File
  - Text File

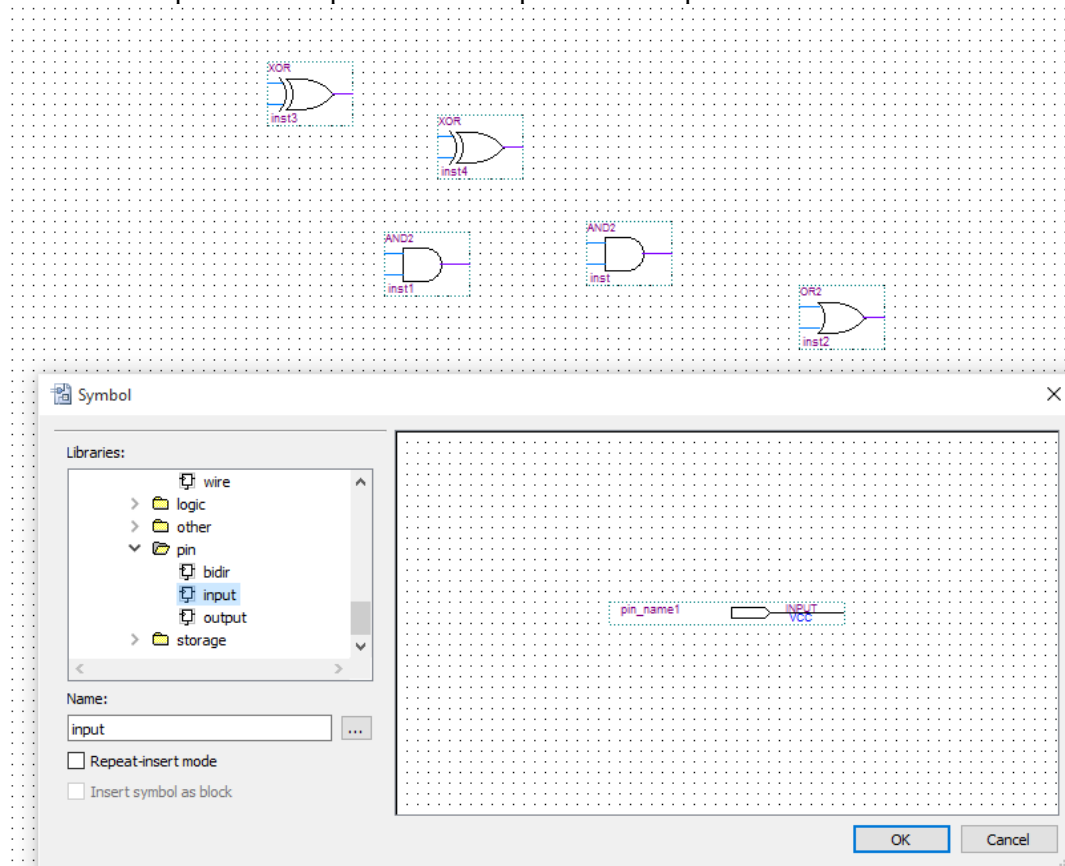
OK   Cancel   Help

Save As → fulladder.bdf

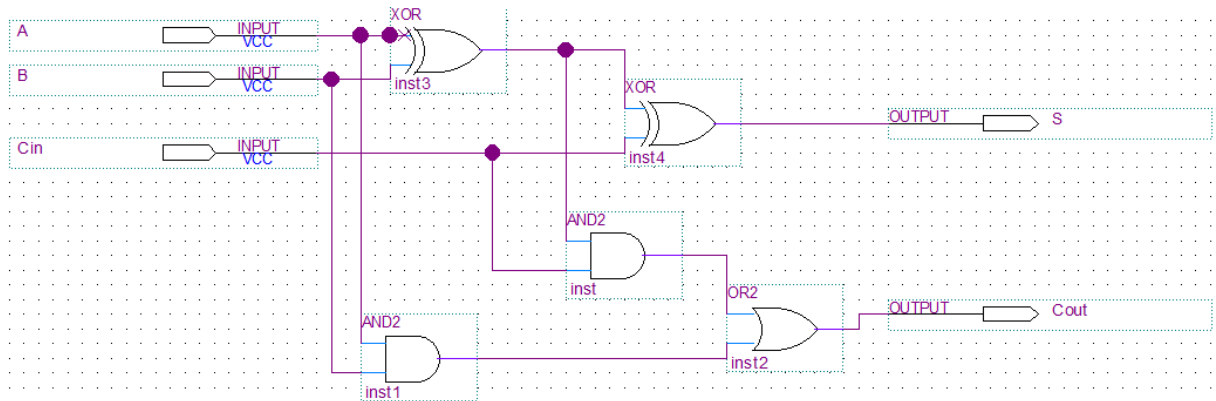
- Double click on drawing canvas.



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



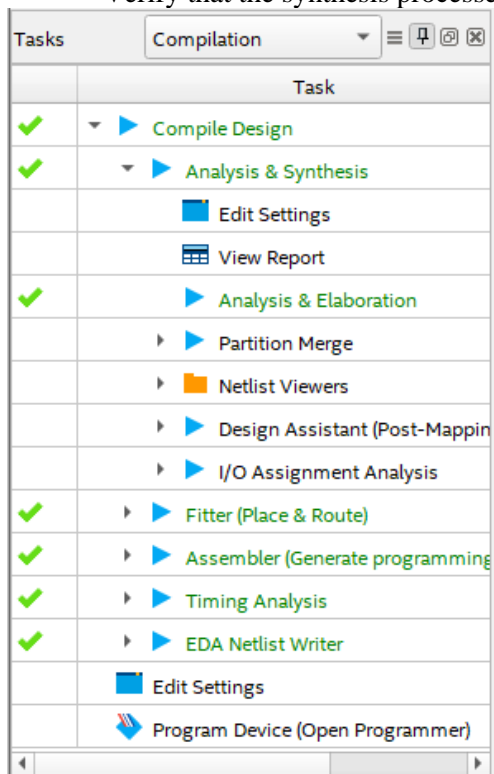
- Rename the input and output pins to A,B, Cin and S, Cout respectively by double-clicking on the pins just inserted
- Connect the components together by selecting the wire connection on the menu bar.



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



- Verify that the synthesis processes have finished successfully.



- View synthesis report → Fitter → Summary

The screenshot displays the Quartus II IDE interface. On the left, the 'Project Navigator' shows a list of tasks under the 'Compilation' category. The 'View Report' task is highlighted with a red circle. The 'Table of Contents' pane in the center lists various reports, with 'Flow Summary' selected. The 'Flow Summary' report is displayed on the right, showing the status of the compilation process.

**Project Navigator - Tasks**

Task	Time
Compile Design	00:01:...
Analysis & Synthesis	00:00:...
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:...
EDA Netlist Writer	00:00:...
Edit Settings	
Program Device (Open Programmer)	

**Table of Contents**

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
  - Summary
  - Settings
    - Parallel Compilation
    - Source Files Read
    - Resource Usage Summary
    - Resource Utilization by Entity
  - Optimization Results
  - Post-Synthesis Netlist Statistics for
  - Elapsed Time Per Partition
- Messages
- Fitter
- Assembler
- Timing Analyzer
- EDA Netlist Writer
- Flow Messages
- Flow Suppressed Messages

## 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:** 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

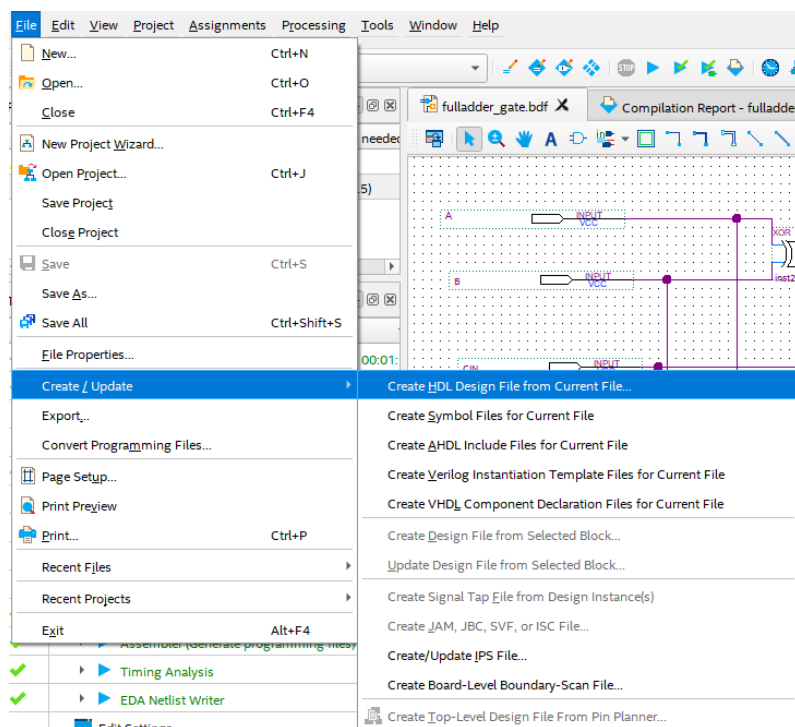
```
set questa_fse_directory "$quartus_path/./questa_fse"
if {[file isdirectory $questa_fse_directory]} {
    set questa_installation "$questa_fse_directory
```

→ Remove the " to  
set questa\_installation \$questa\_fse\_directory

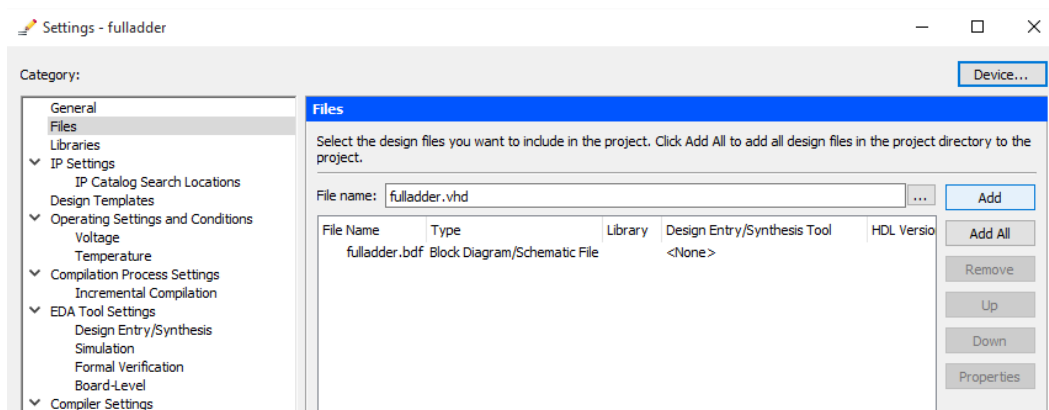
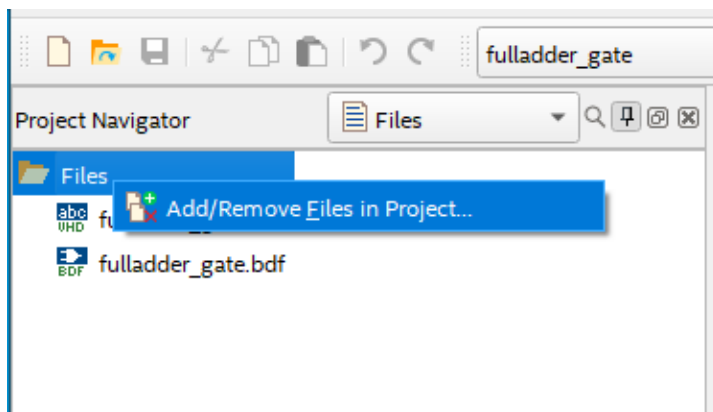
You can also run Questa directly from the installation folder. The default installation folder is:  
C:\intelFPGA\_lite\22.1std\questa\_fse\win64

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

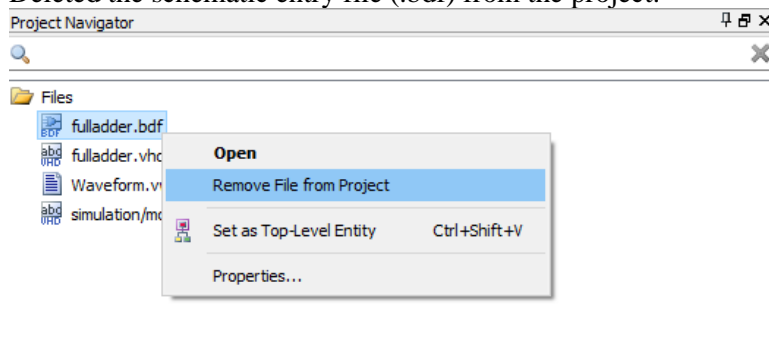


- Add newly generated fulladder.vhd/fulladder.v file to project → Project Navigator→Files →Add/Remove files



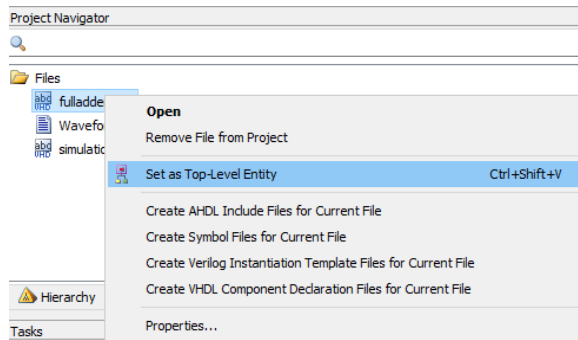
- Quartus Prime does not allow the automatic generation of testbenches directly from a schematic entry file, hence the testbench must be generated for the newly imported Verilog/VHDL file. Also because both files have the same “entity” (module name)

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

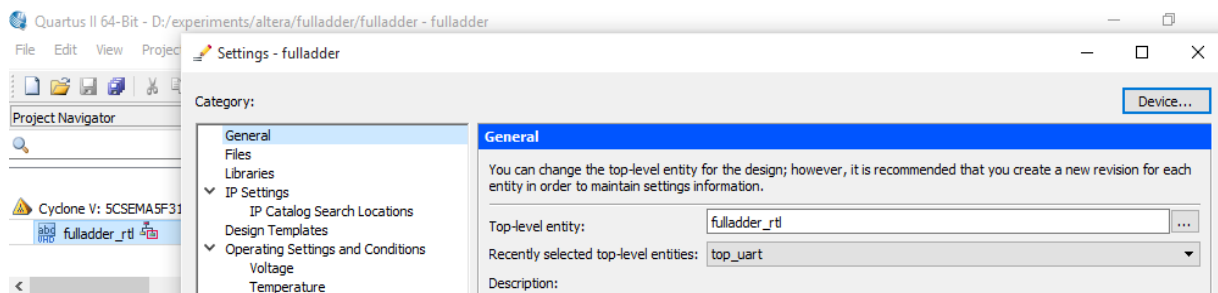




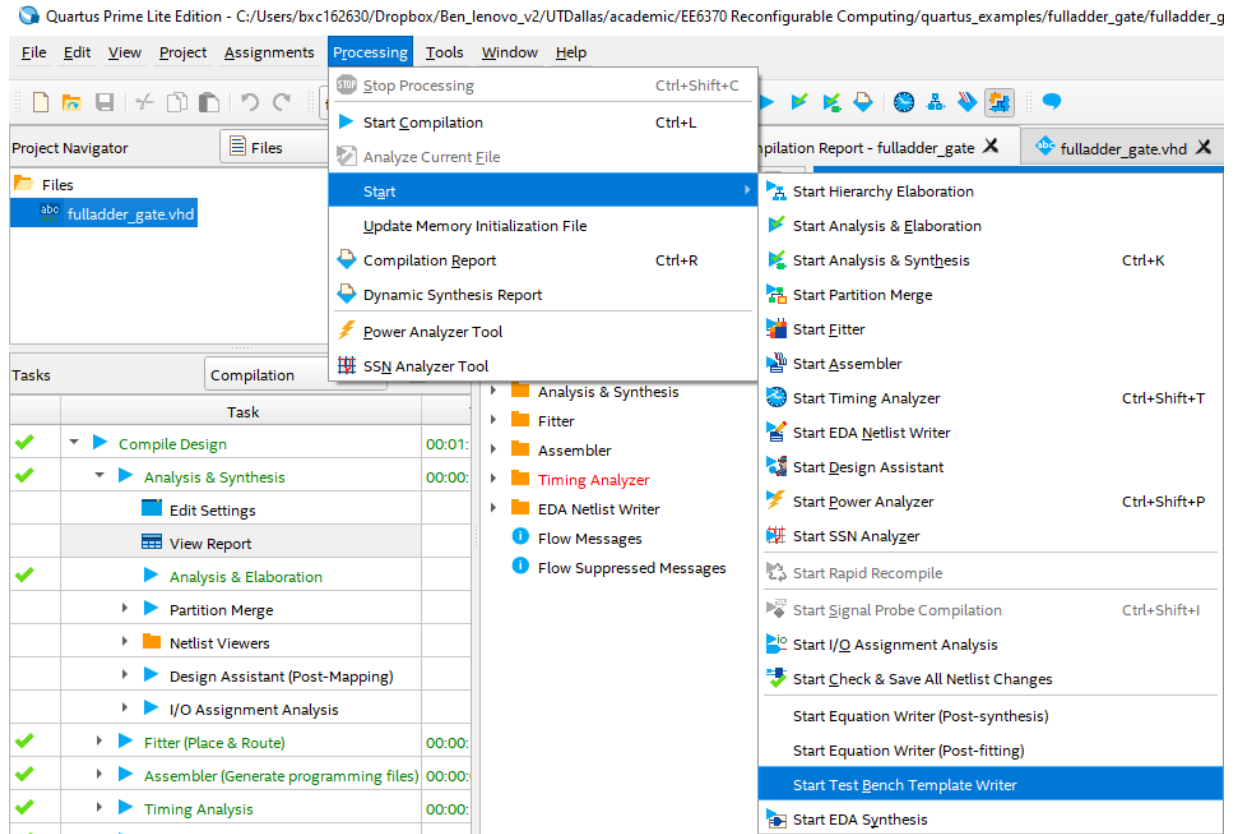
- Re-compile the design setting the Verilog/VHDL file as top entity.



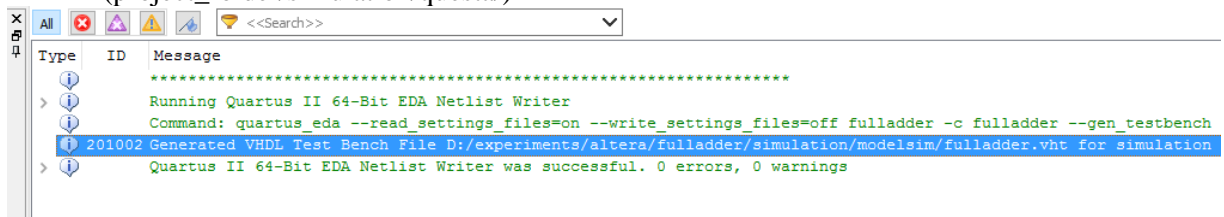
- By default, Quartus assumes that the filename is the same as the top entity. If not, it needs to be set manually. For this  
Project Navigator → Hierarchy → right click on Verilog/VHDL file → Settings



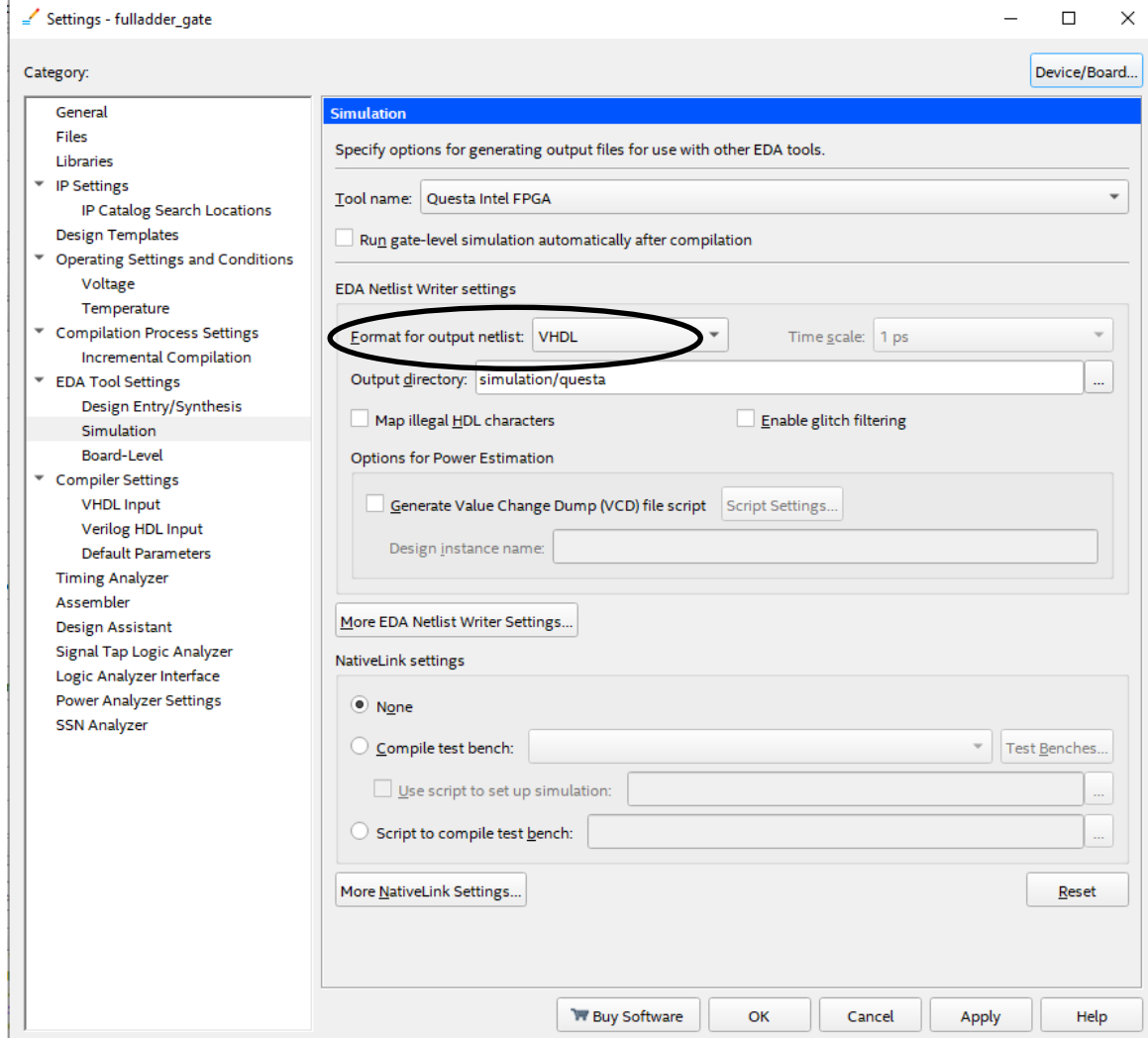
- Create a testbench template: Processing → Start → Start Test Bench Template Writer



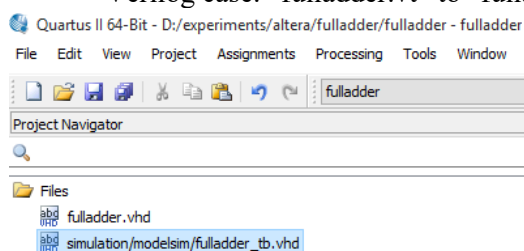
- The console window illustrates where the testbench has been generated (project\_folder/simulation/questa/)



**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



- 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”



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

```

28 LIBRARY ieee;
29 USE ieee.std_logic_1164.all;
30
31 ENTITY fulladder_vhd_tst IS
32 END fulladder_vhd_tst;
33 ARCHITECTURE fulladder_arch OF fulladder_vhd_tst IS
34 -- constants
35 -- signals
36 SIGNAL A : STD_LOGIC;
37 SIGNAL B : STD_LOGIC;
38 SIGNAL Cin : STD_LOGIC;
39 SIGNAL Cout : STD_LOGIC;
40 SIGNAL S : STD_LOGIC;
41 COMPONENT fulladder
42 PORT (
43 A : IN STD_LOGIC;
44 B : IN STD_LOGIC;
45 Cin : IN STD_LOGIC;
46 Cout : OUT STD_LOGIC;
47 S : OUT STD_LOGIC
48 );
49 END COMPONENT;
50 BEGIN
51 i1 : fulladder
52 PORT MAP (
53 -- list connections between master ports and signals
54 A => A,
55 B => B,
56 Cin => Cin,
57 Cout => Cout,
58 S => S
59 );
60 init : PROCESS
61 -- variable declarations
62 BEGIN
63 -- code that executes only once
64 WAIT;
65 END PROCESS init;
66 always : PROCESS
67 -- optional sensitivity list
68 -- (
69 -- variable declarations
70 BEGIN
71 -- code executes for every event on sensitivity list
72
73 wait for 1 ns;
74 A <= '0';
75 B <= '0';
76 Cin <= '0';
77
78 wait for 1 ns;
79 A <= '0';
80 B <= '0';
81 Cin <= '1';
82
83 wait for 1 ns;
84 A <= '0';
85 B <= '1';
86 Cin <= '0';
87
88 wait for 1 ns;
89 A <= '0';
90 B <= '1';
91 Cin <= '1';
92
93 wait for 1 ns;
94 A <= '1';
95 B <= '0';
96 Cin <= '0';
97
98 wait for 1 ns;
99 A <= '1';
100 B <= '0';
101 Cin <= '1';
102
103 wait for 1 ns;
104 A <= '1';
105 B <= '1';
106 Cin <= '0';
107
108 wait for 1 ns;
109 A <= '1';
110 B <= '1';
111 Cin <= '1';
112
113 WAIT;
114 END PROCESS always;
115 END fulladder_arch;

```

Entity is empty

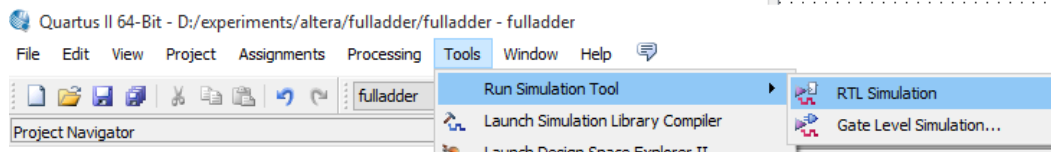
Signal declarations to drive adders' inputs

Design to be tested declared as a component (UUT)

Instantiate the UUT in architecture and connect signals

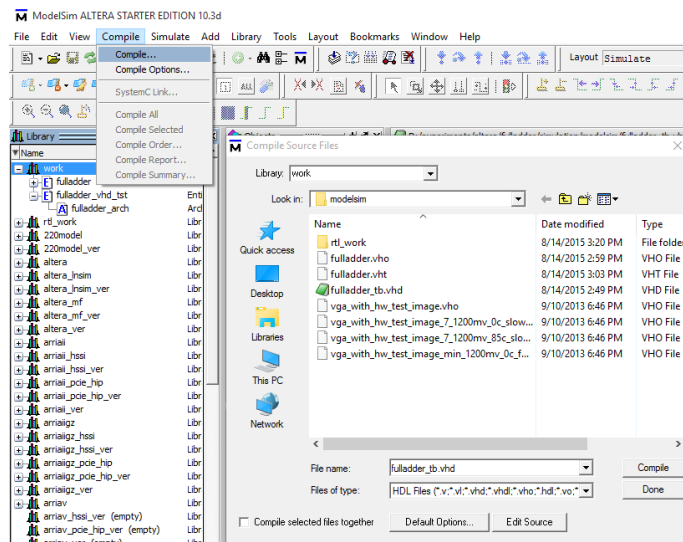
Set values for adders inputs A, B, Cin  
Wait for X ns to insert delay between signal changes

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

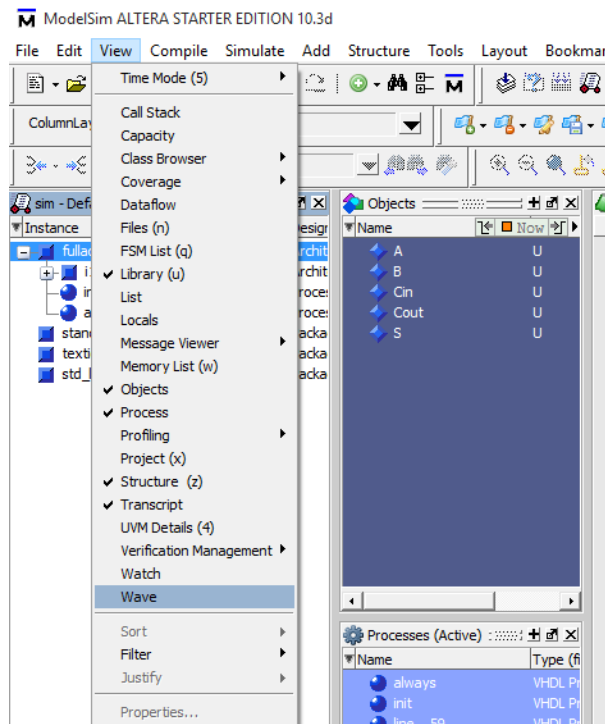


Questa simulator should start automatically.

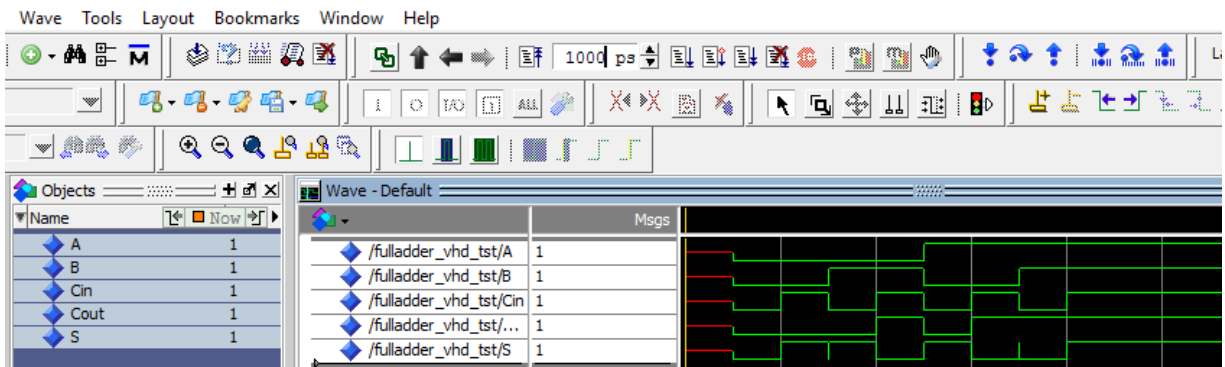
Compile testbench : Compile → Select fulladder\_tb.vhd



- 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



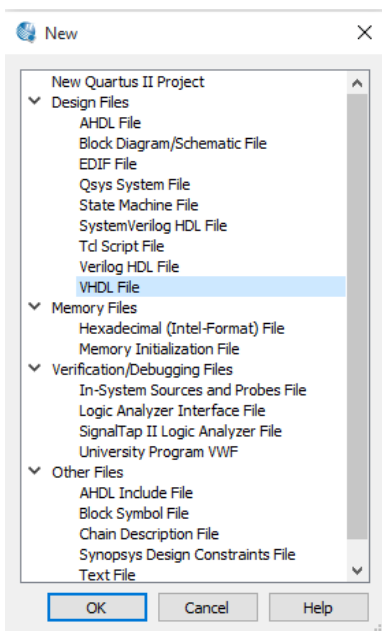
- Select the signals to be displayed and drag to Waveform.
- Set the simulation step to 1ns
- Start simulation.



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

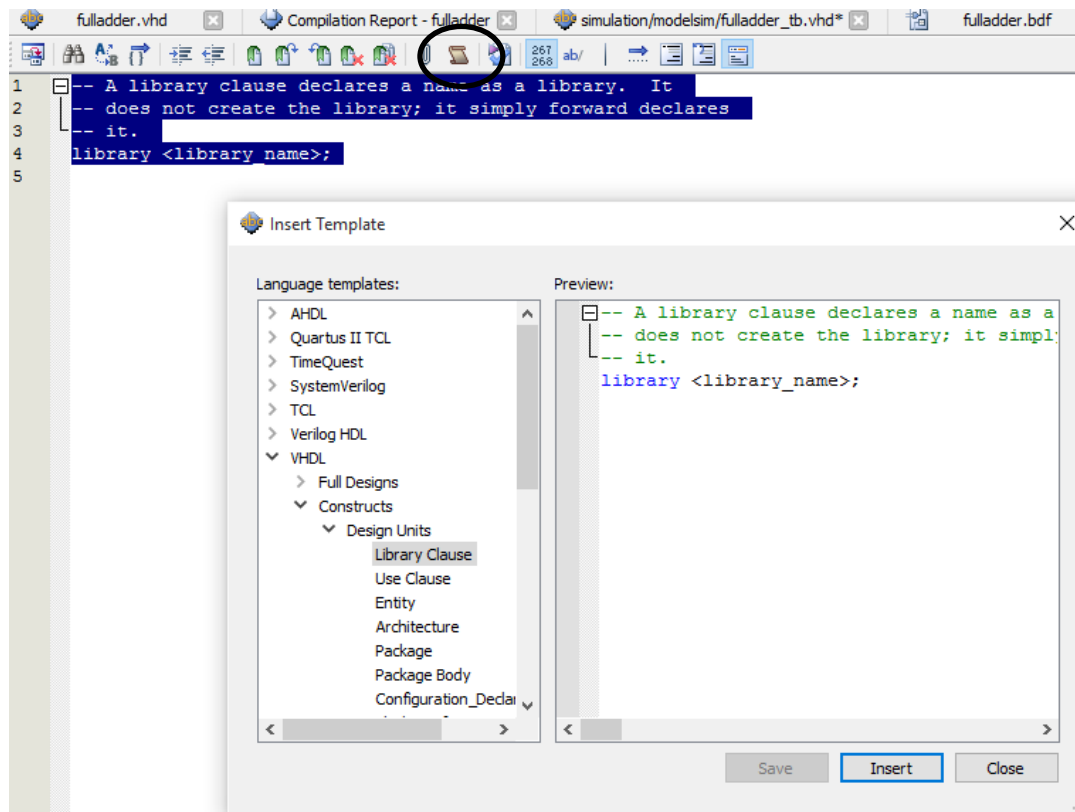
Re-do the same exercise creating an RTL description (Verilog or VHDL) of the same 1-bit adder

- 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.



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]