CSC 327 – Lab 6

Shyang Kao

ID: 5206454

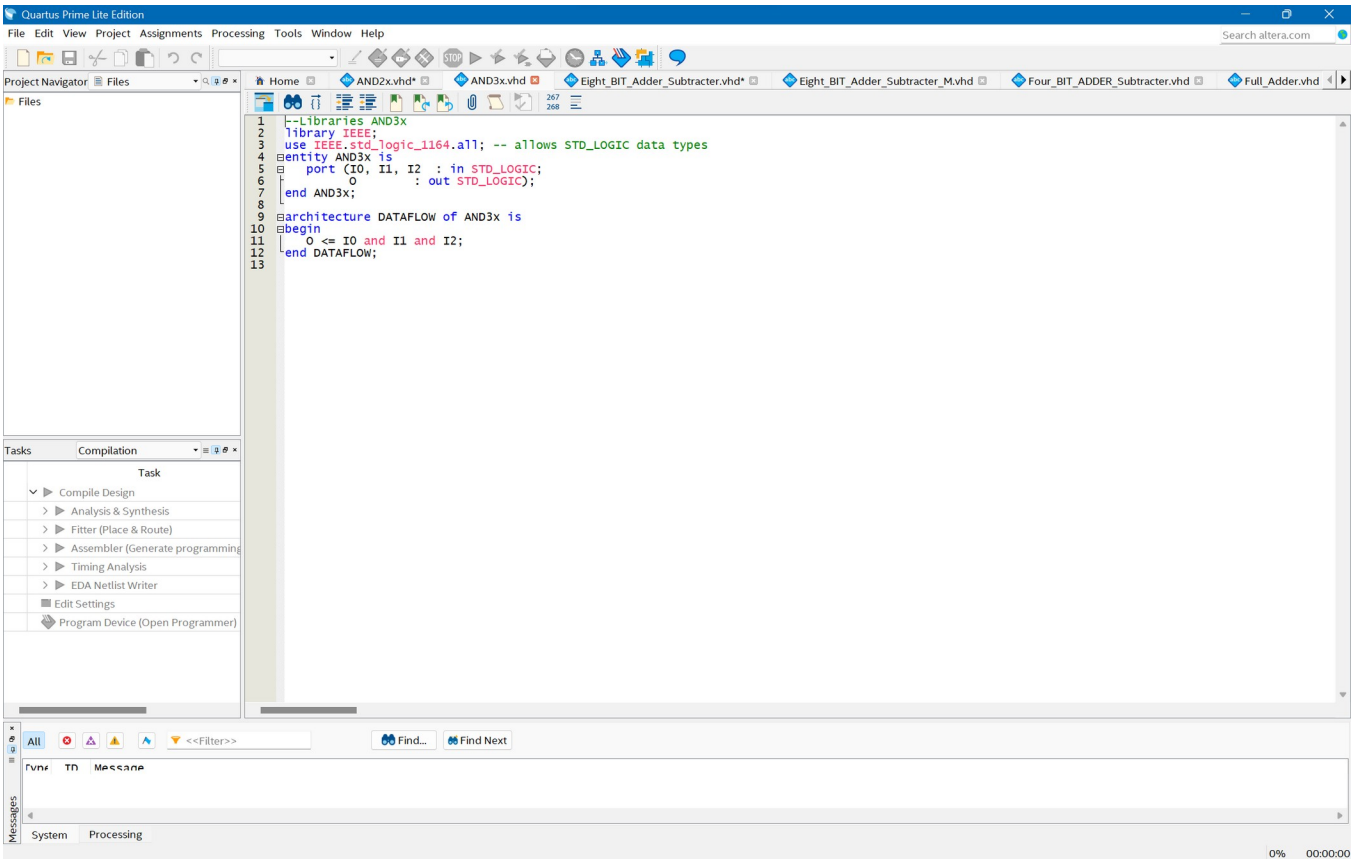
Course Name: CSC 327

Due Date: March 31, 2022

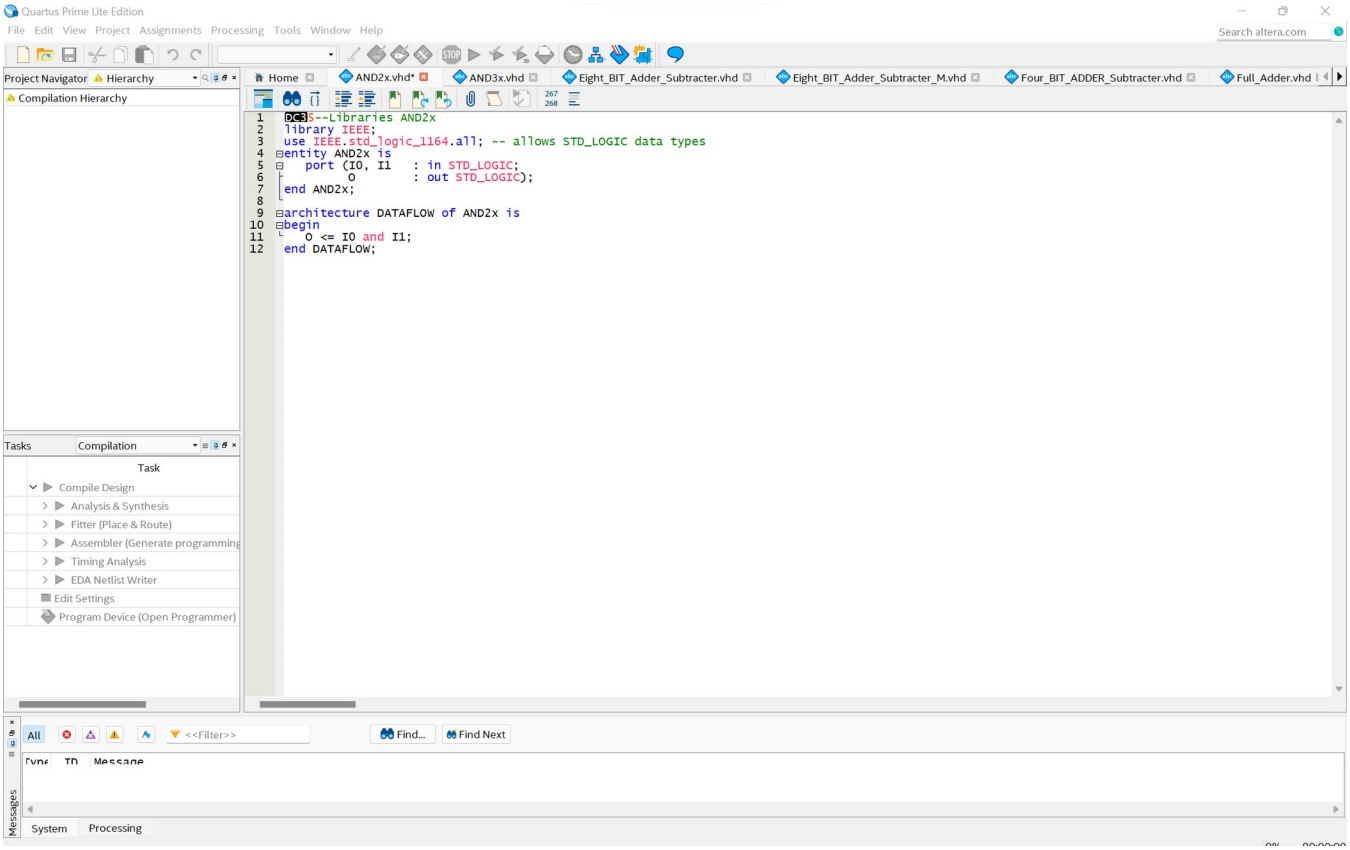
Experiment Description

The goal of this lab project is for each student to learn the fundamentals of hardware description language (particularly VHDL) and how to utilize it to design and construct simple digital circuits. Students will learn conventional structural and behavioral hardware description language (VHDL) design entry methodologies, as well as the advantages of pre- and post-synthesis digital simulation (using ModelSim simulator and Quartus II).

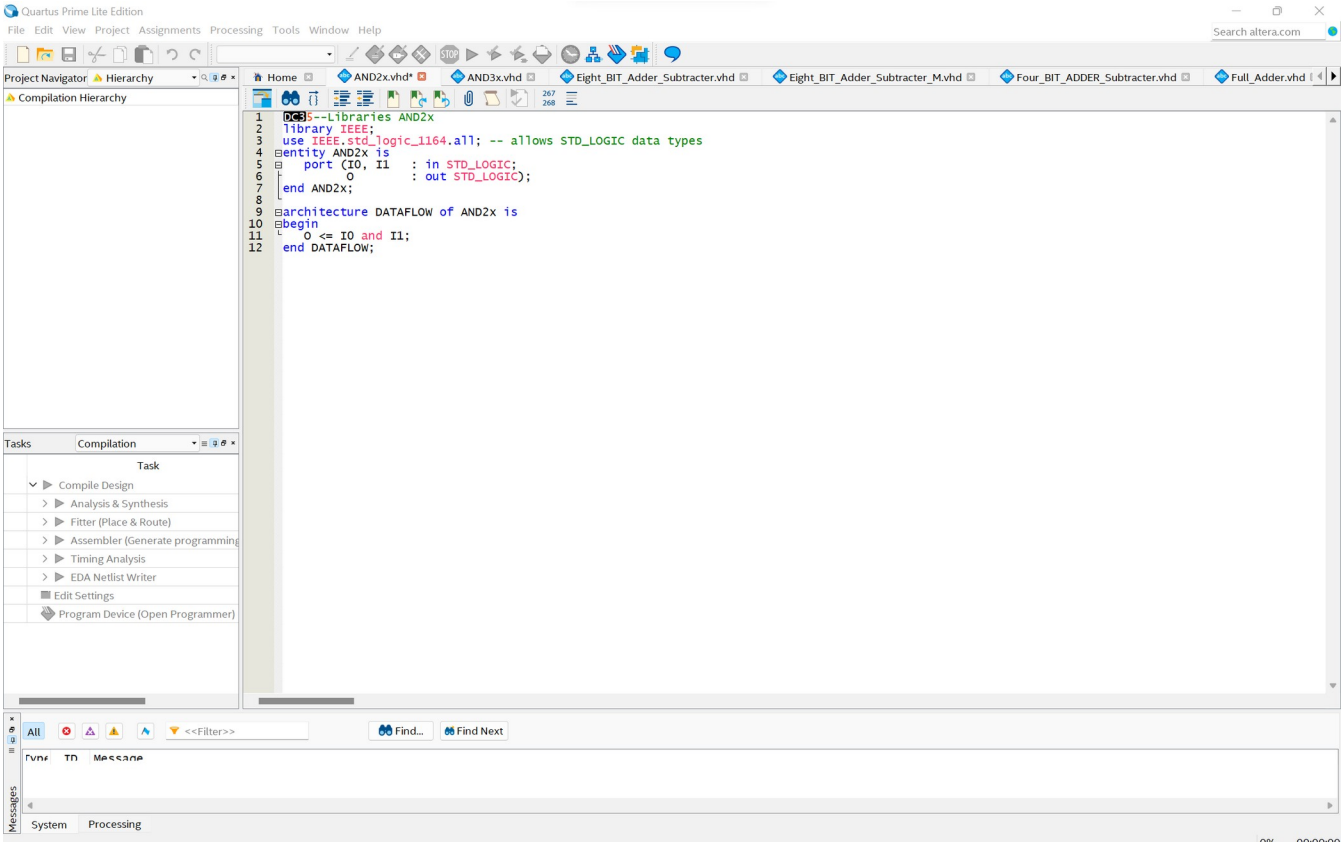
ANDx3:


Quartus Prime Lite Edition
File Edit View Project Assignments Processing Tools Window Help
Search altera.com
Project Navigator Files
Files
Home AND2x.vhd* AND3x.vhd Eight_BIT_Adder_Subtracter.vhd* Eight_BIT_Adder_Subtracter_M.vhd Four_BIT_ADDER_Subtracter.vhd Full_Adder.vhd
--Libraries AND3x
library IEEE;
use IEEE.std_logic_1164.all; -- allows STD_LOGIC data types
entity AND3x is
port (I0, I1, I2 : in STD_LOGIC;
O : out STD_LOGIC);
end AND3x;
architecture DATAFLOW of AND3x is
begin
O <= I0 and I1 and I2;
end DATAFLOW;
Tasks Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming
Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
All
<<Filter>>
Find...
Find Next
Type ID Message
Messages
System
Processing
0%
00:00:00

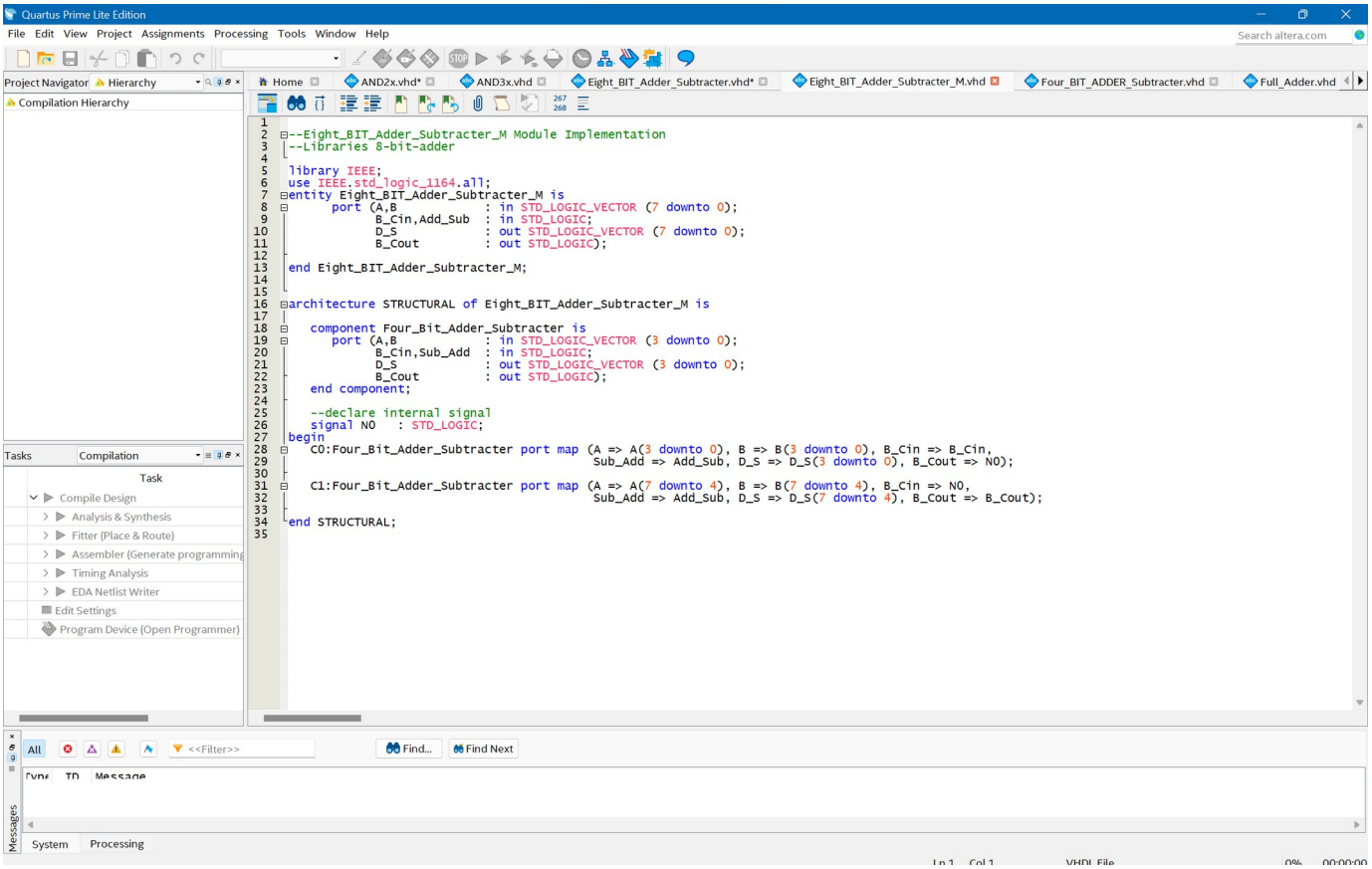
ANDx2


Quartus Prime Lite Edition
File Edit View Project Assignments Processing Tools Window Help
Search altera.com
Project Navigator Hierarchy
Compilation Hierarchy
Tasks Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming
Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
Home
AND2x.vhd*
AND3x.vhd
Eight_BIT_Adder_Subtracter.vhd
Eight_BIT_Adder_Subtracter_M.vhd
Four_BIT_ADDER_Subtracter.vhd
Full_Adder.vhd
1 --Libraries AND2x
2 library IEEE;
3 use IEEE.std_logic_1164.all; -- allows STD_LOGIC data types
4 entity AND2x is
5 port (I0, I1 : in STD_LOGIC;
6 O : out STD_LOGIC);
7 end AND2x;
8
9 architecture DATAFLOW of AND2x is
10 begin
11 O <= I0 and I1;
12 end DATAFLOW;
All
<<Filter>>
Find...
Find Next
Type ID Message
Messages
System
Processing

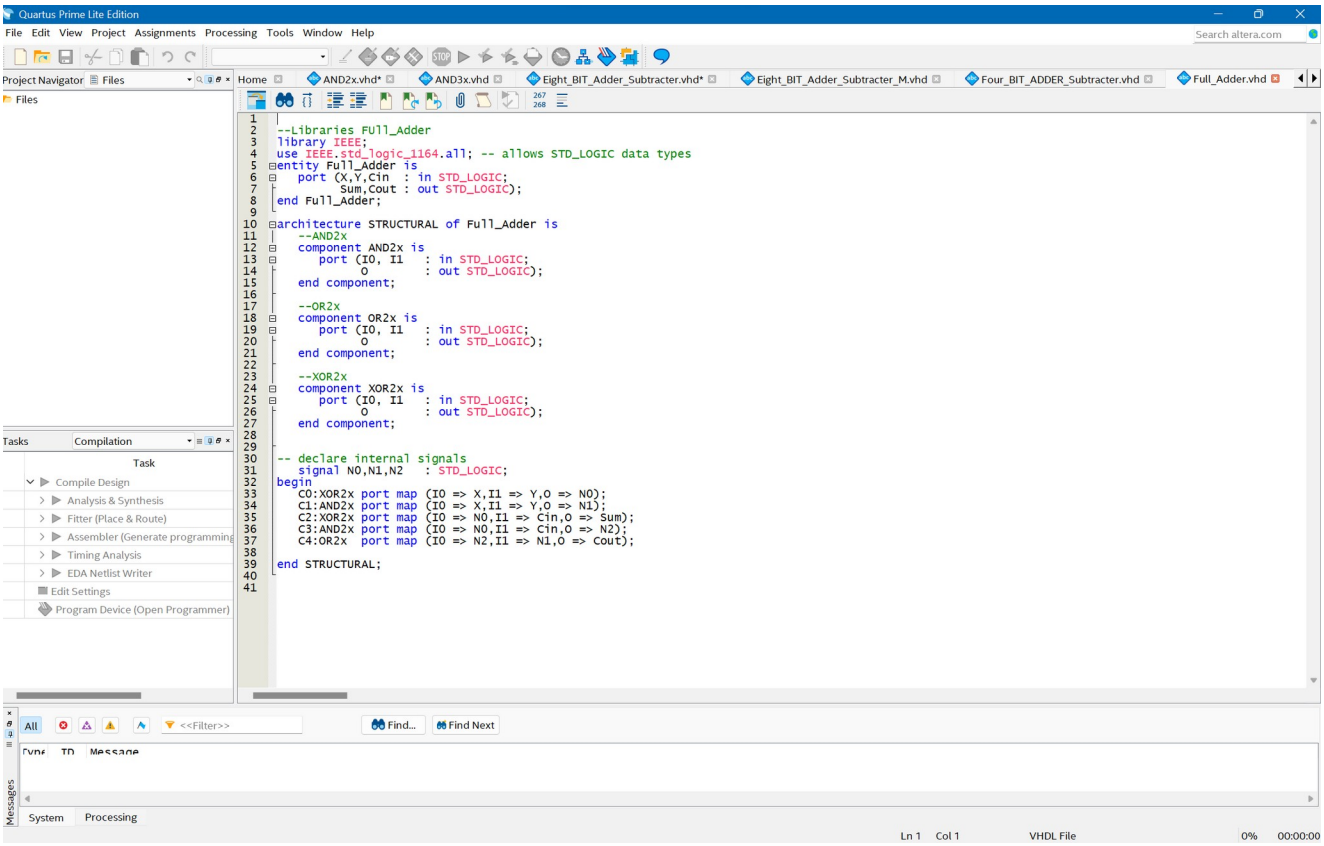
Eight Bit Adder Subtracter:



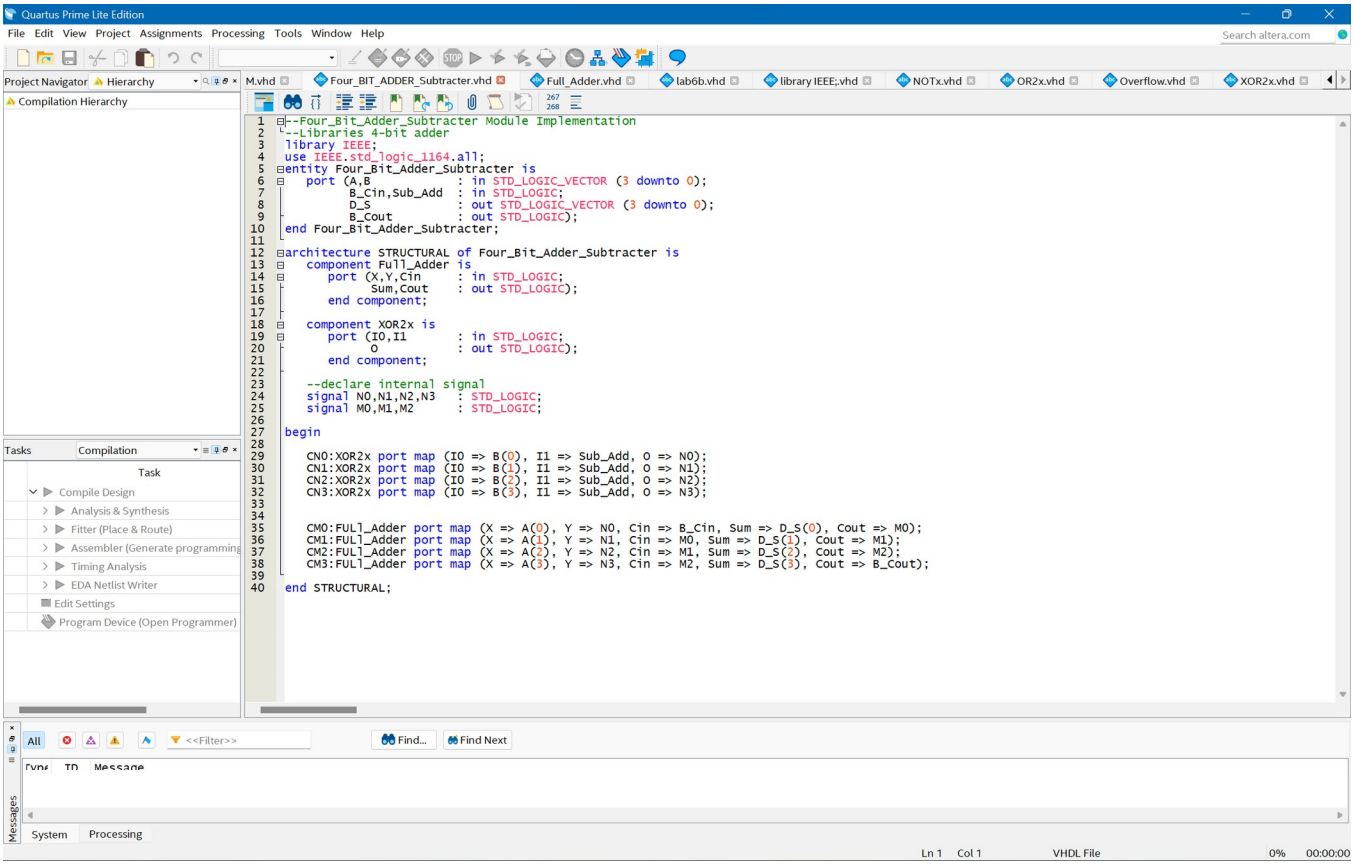
Eight Bit Adder Subtracter M:



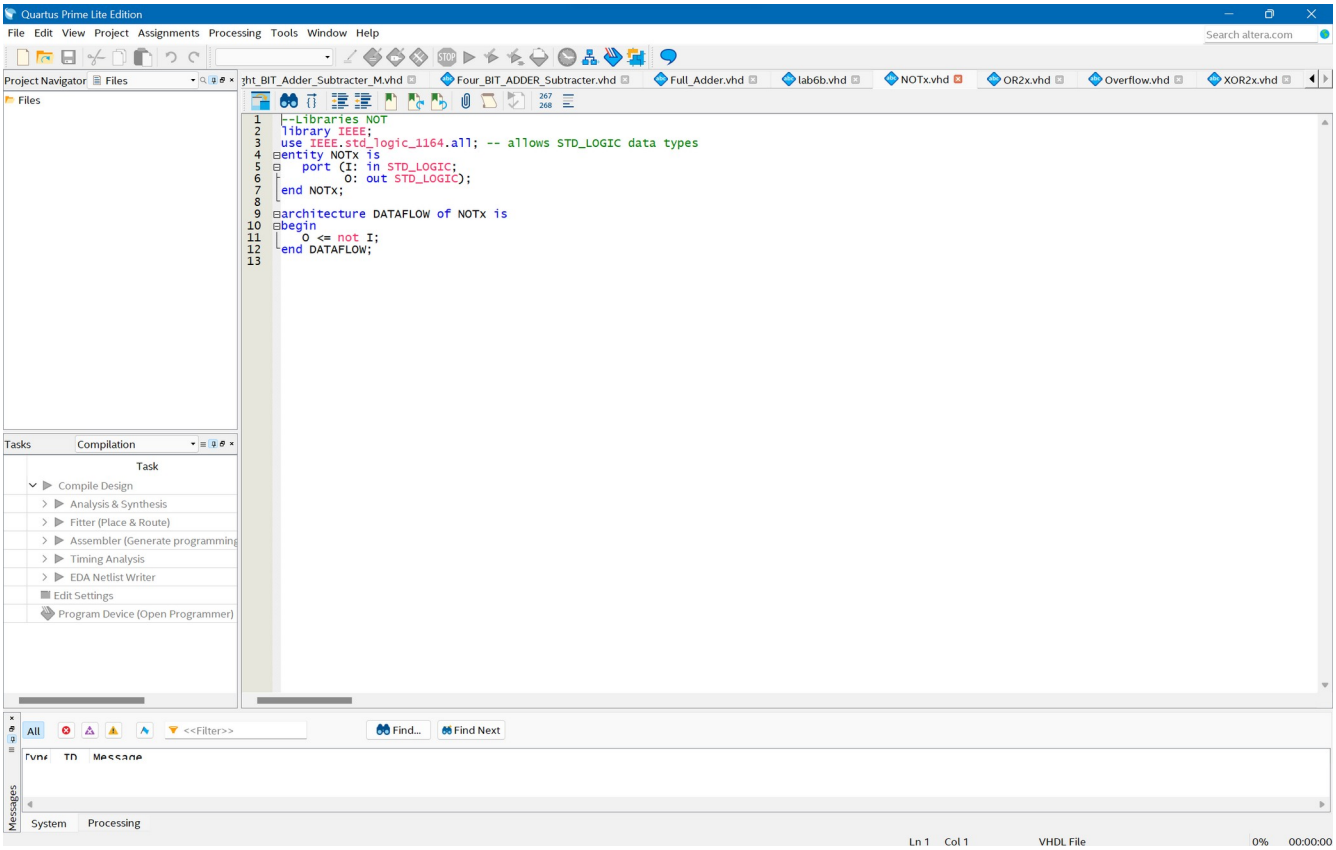
Full Adder:


Quartus Prime Lite Edition
File Edit View Project Assignments Processing Tools Window Help
Search altera.com
Project Navigator
Files
Home
AND2x.vhd*
AND3x.vhd
Eight_BIT_Adder_Subtracter.vhd*
Eight_BIT_Adder_Subtracter_M.vhd
Four_BIT_ADDER_Subtracter.vhd
Full_Adder.vhd
--Libraries FUll_Adder
library IEEE;
use IEEE.std_logic_1164.all; -- allows STD_LOGIC data types
entity Full_Adder is
port (X,Y,Cin : in STD_LOGIC;
Sum,Cout : out STD_LOGIC);
end Full_Adder;
architecture STRUCTURAL of Full_Adder is
--AND2x
component AND2x is
port (IO, I1 : in STD_LOGIC;
O : out STD_LOGIC);
end component;
--OR2x
component OR2x is
port (IO, I1 : in STD_LOGIC;
O : out STD_LOGIC);
end component;
--XOR2x
component XOR2x is
port (IO, I1 : in STD_LOGIC;
O : out STD_LOGIC);
end component;
-- declare internal signals
signal N0,N1,N2 : STD_LOGIC;
begin
C0:XOR2x port map (IO => X,I1 => Y,O => N0);
C1:AND2x port map (IO => X,I1 => Y,O => N1);
C2:XOR2x port map (IO => N0,I1 => Cin,O => Sum);
C3:AND2x port map (IO => N0,I1 => Cin,O => N2);
C4:OR2x port map (IO => N2,I1 => N1,O => Cout);
end STRUCTURAL;
Tasks
Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming
Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
All
<<Filter>>
Find...
Find Next
Type ID Message
Messages
System
Processing
Ln 1 Col 1 VHDL File 0% 00:00:00

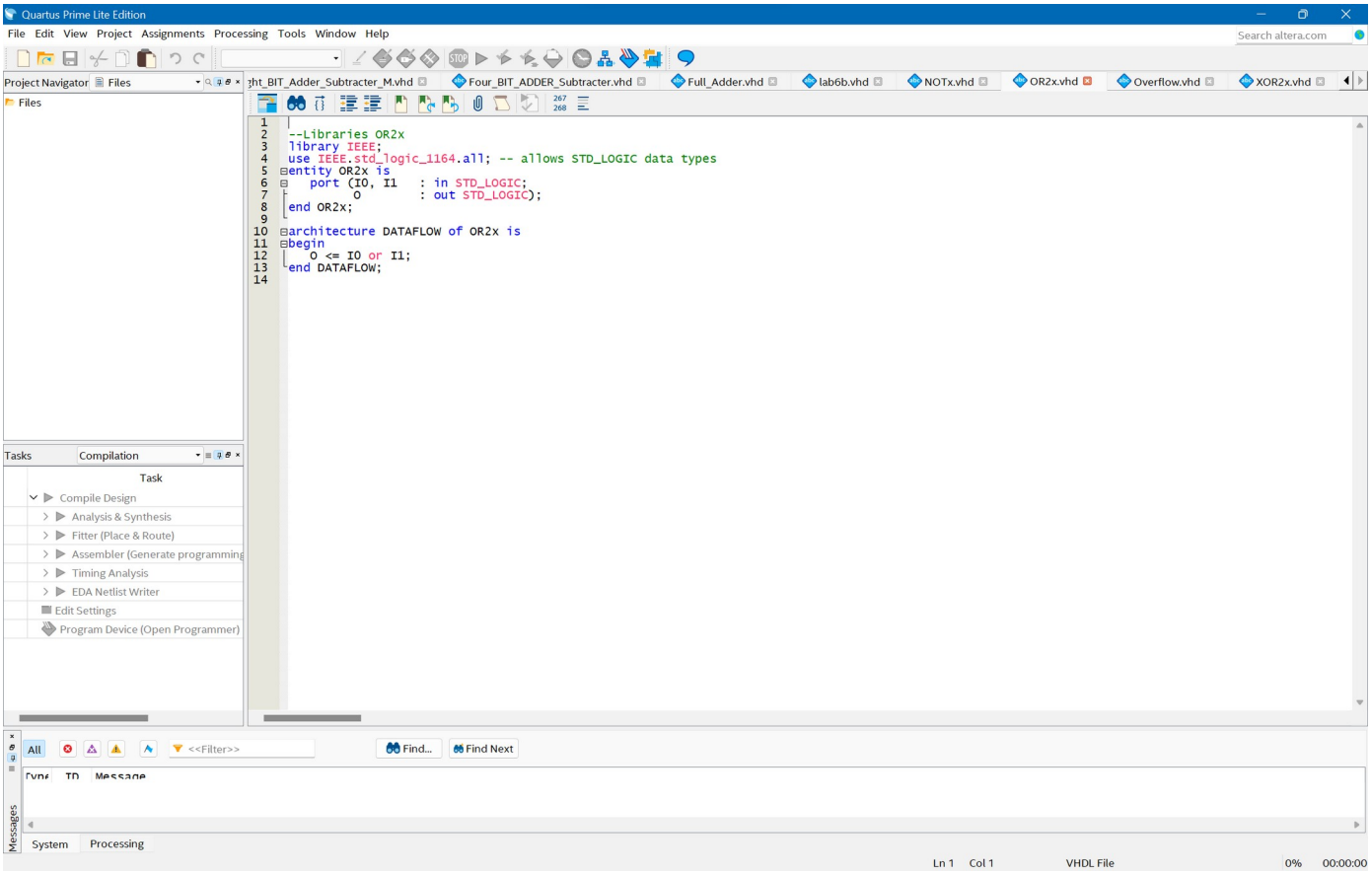
Four Bit Adder Subtracter:



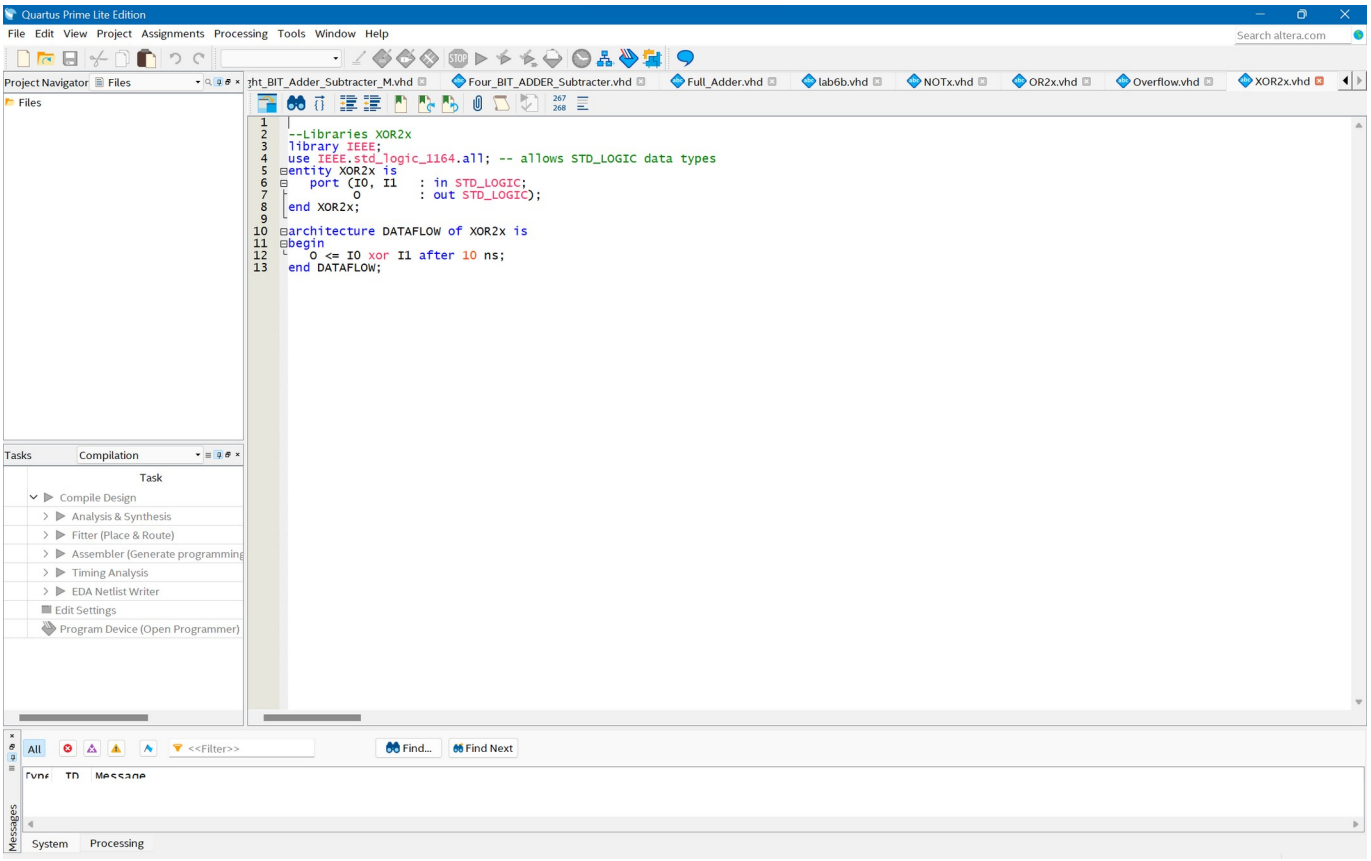
NOTx:


Quartus Prime Lite Edition
File Edit View Project Assignments Processing Tools Window Help
Search altera.com
Project Navigator Files
Files
ht_BIT_Adder_Subtracter_M.vhd
Four_BIT_ADDER_Subtracter.vhd
Full_Adder.vhd
lab6b.vhd
NOTx.vhd
OR2x.vhd
Overflow.vhd
XOR2x.vhd
1 --Libraries NOT
2 library IEEE;
3 use IEEE.std_logic_1164.all; -- allows STD_LOGIC data types
4 entity NOTx is
5 port (I: in STD_LOGIC;
6 O: out STD_LOGIC);
7 end NOTx;
8
9 architecture DATAFLOW of NOTx is
10 begin
11 O <= not I;
12 end DATAFLOW;
13
Tasks Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming
Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
All
<<Filter>>
Find...
Find Next
Type ID Message
Messages
System
Processing
Ln 1 Col 1 VHDL File 0% 00:00:00

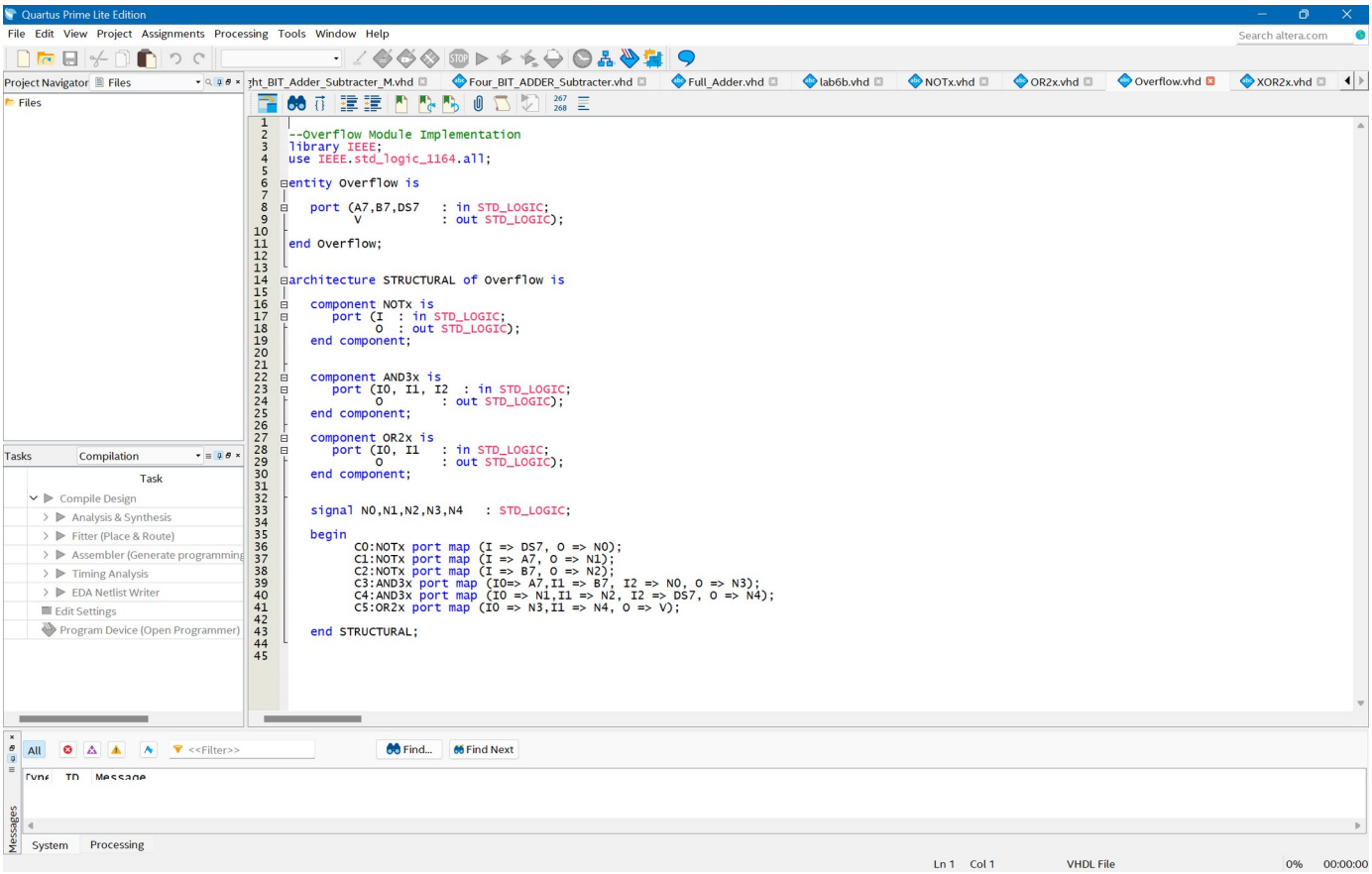
ORx:


Quartus Prime Lite Edition
File Edit View Project Assignments Processing Tools Window Help
Search altera.com
Project Navigator
Files
Four_BIT_ADDER_Subtracter.vhd
Full_Adder.vhd
lab6b.vhd
NOTx.vhd
OR2x.vhd
Overflow.vhd
XOR2x.vhd
--Libraries OR2x
library IEEE;
use IEEE.std_logic_1164.all; -- allows STD_LOGIC data types
entity OR2x is
port (IO, I1 : in STD_LOGIC;
O : out STD_LOGIC);
end OR2x;
architecture DATAFLOW of OR2x is
begin
O <= IO or I1;
end DATAFLOW;
Tasks
Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming
Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
Find...
Find Next
Type ID Message
Messages
System
Processing
Ln 1 Col 1 VHDL File 0% 00:00:00

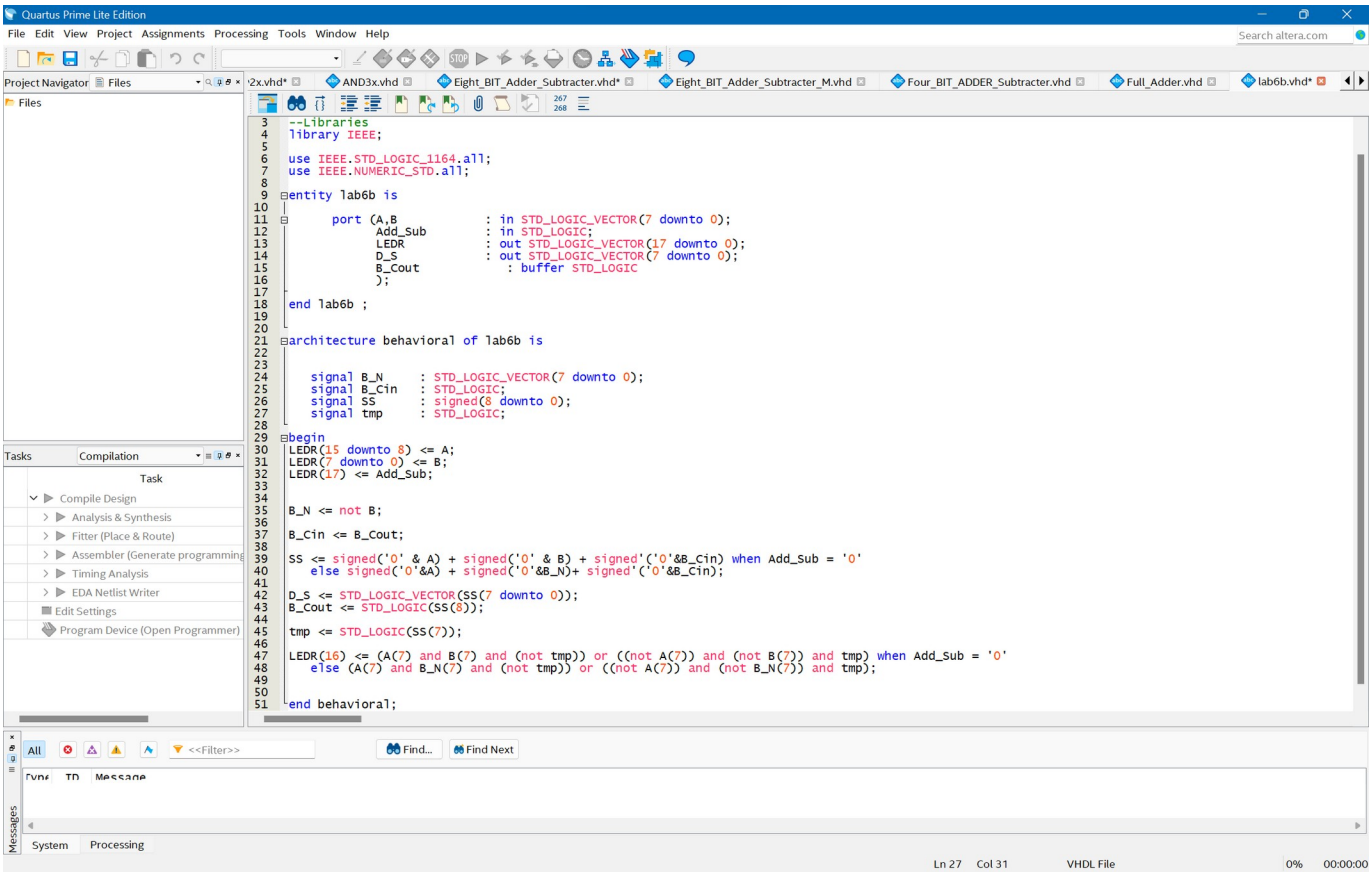
XORx


Quartus Prime Lite Edition
File Edit View Project Assignments Processing Tools Window Help
Search altera.com
Project Navigator Files
Files
ght_BIT_Adder_Subtracter_M.vhd
Four_BIT_ADDER_Subtracter.vhd
Full_Adder.vhd
lab6b.vhd
NOTx.vhd
OR2x.vhd
Overflow.vhd
XOR2x.vhd
1 --Libraries XOR2x
2 library IEEE;
3 use IEEE.std_logic_1164.all; -- allows STD_LOGIC data types
4 entity XOR2x is
5 port (I0, I1 : in STD_LOGIC;
6 O : out STD_LOGIC);
7 end XOR2x;
8
9 architecture DATAFLOW of XOR2x is
10 begin
11 O <= I0 xor I1 after 10 ns;
12 end DATAFLOW;
13
Tasks Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming
Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
All
<<Filter>>
Find...
Find Next
Type ID Message
Messages
System
Processing

Overflow:

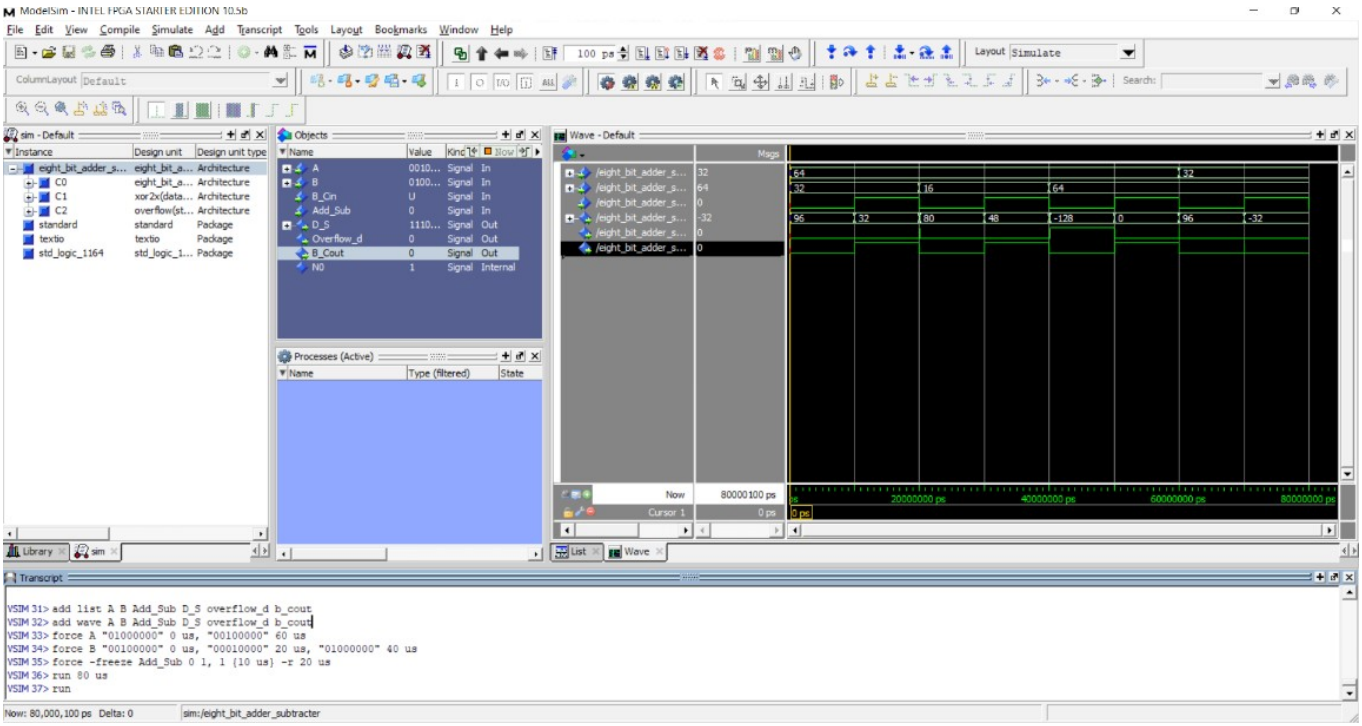

Quartus Prime Lite Edition
File Edit View Project Assignments Processing Tools Window Help
Search altera.com
Project Navigator Files
Files
ght_BIT_Adder_Subtracter_M.vhd
Four_BIT_ADDER_Subtracter.vhd
Full_Adder.vhd
lab6b.vhd
NOTx.vhd
OR2x.vhd
Overflow.vhd
XOR2x.vhd
--Overflow Module Implementation
library IEEE;
use IEEE.std_logic_1164.all;
entity Overflow is
port (A7,B7,DS7 : in STD_LOGIC;
V : out STD_LOGIC);
end Overflow;
architecture STRUCTURAL of Overflow is
component NOTx is
port (I : in STD_LOGIC;
O : out STD_LOGIC);
end component;
component AND3x is
port (I0, I1, I2 : in STD_LOGIC;
O : out STD_LOGIC);
end component;
component OR2x is
port (I0, I1 : in STD_LOGIC;
O : out STD_LOGIC);
end component;
signal N0,N1,N2,N3,N4 : STD_LOGIC;
begin
C0:NOTx port map (I => DS7, O => N0);
C1:NOTx port map (I => A7, O => N1);
C2:NOTx port map (I => B7, O => N2);
C3:AND3x port map (I0=> A7,I1 => B7, I2 => N0, O => N3);
C4:AND3x port map (I0 => N1,I1 => N2, I2 => DS7, O => N4);
C5:OR2x port map (I0 => N3,I1 => N4, O => V);
end STRUCTURAL;
Tasks Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming
Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
All
<<Filter>>
Find...
Find Next
Type ID Message
Messages
System
Processing
Ln 1 Col 1
VHDL File
0%
00:00:00

Lab6b



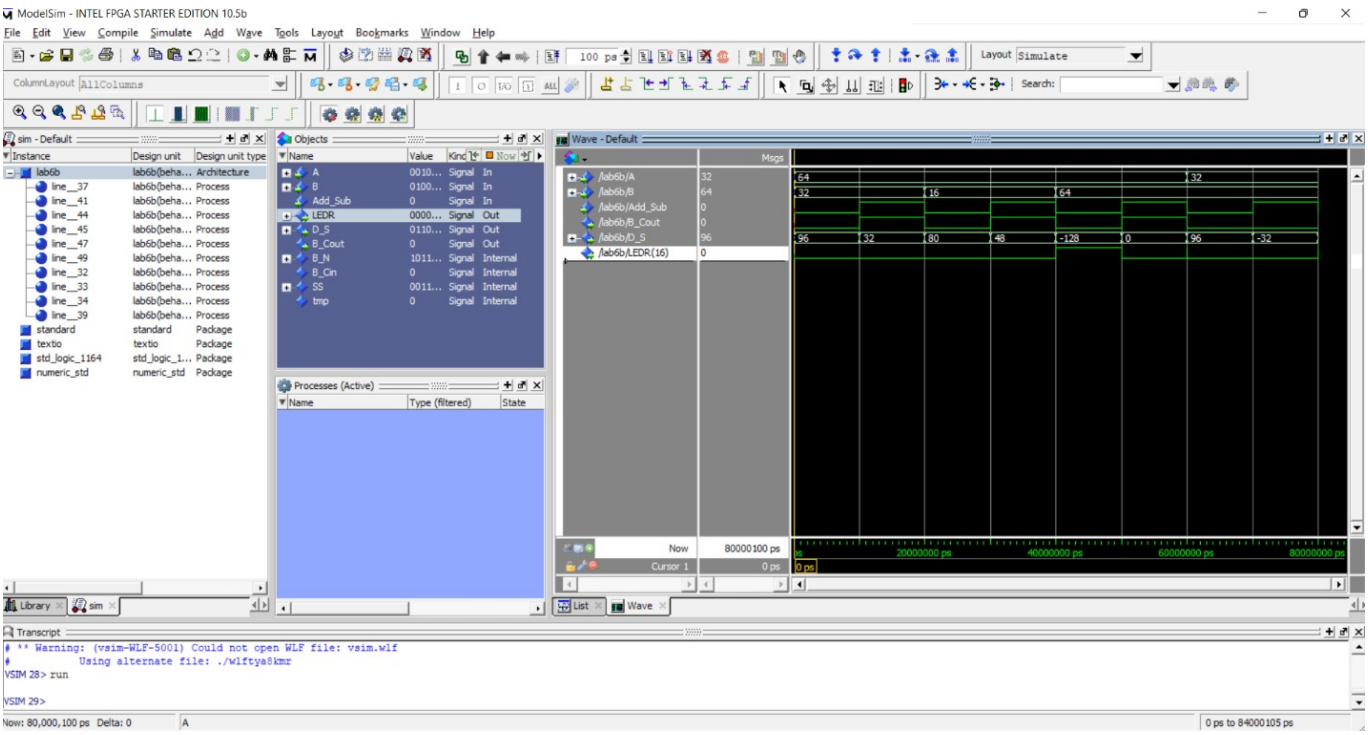
This is the VDHL code representing the 2’s compliment and 1’st compliment (currently shown here).

Lab6 Part 1 ModelSim Results



This is the simulation for eightbit Adder Subtracter

Lab6b Part 2 ModelSim Results



This is the simulation for 2's compliment

QUESTIONS

1) BUFFER signifies output, with feedback from the output back into the design entity, which is why we use it. If the buffer is set to out, the output will only show output and will not provide feedback to the design entity.

2) With the hierarchical schematic capture approach, we may construct our own hierarchical design to keep organized and synchronized and have an easier design navigation.
Meanwhile, the hierarchical structural VHDL design approach allows us to divide the complexity into two or more basic designs. This method allows low-level functionality to be packed into modules, allowing for design reuse without having to reinvent and retest the wheel each time.

3) Essentially, schematic capture differs from structural VHDL in that schematic capture includes an overview module of circuits for all gates and ICs. In the meanwhile, structural VHDL specifies code.

4) The structural and behavioral VHDL design methodologies vary in some ways. Behavioral techniques focus solely on the code's functionality, whereas structural techniques focus on not only the code's functionality, but also the structure of the circuits for additional improvements.

5) The code is created and modeled in a straightforward method in behavioral modeling, which completes the code. It is more so oriented on functionality, in order to achieve the aim as quickly as possible. Structural code is more focused on the end objective and is implemented using primitives. It is not as efficient as behavioral code, but it is easier to understand.