EXPT NO.: 2

DESIGN OF HALF AND FULL ADDER

AIM:

To design and simulate half adder, full adder using basic gates and full adder using 2 half adders.

HALF ADDER USING BASIC GATES:

A logic circuit block used for adding two one-bit numbers or simply two bits is called as a half adder circuit. This circuit has two inputs which accept the two bits and two outputs, with one producing sum output and other produce carry output.

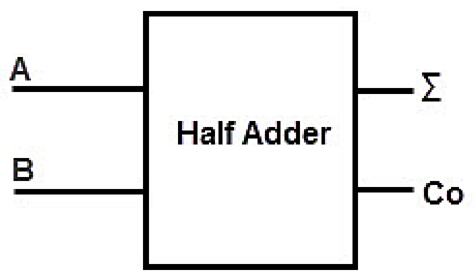
As we know that binary addition is commonly performed by Ex-OR gate, but for the first three rules, it performs the binary addition and when the two inputs are logic 1, it does not develop any carry.

To accomplish the binary addition with Ex-OR gate, there is need of additional circuitry to perform the carry operation. Hence, a half adder is formed by connecting AND gate to the input terminals of the Ex-OR gate so as to produce the carry.

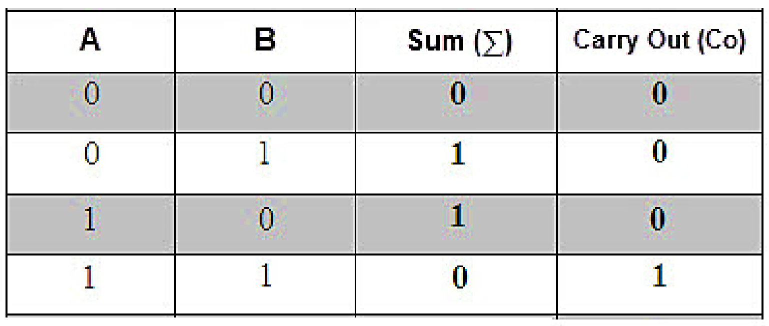
$$\textbf{Sum } (\sum) = A \oplus B$$

$$\textbf{Carry Out (Co)} = A \cdot B$$

The block diagram, circuit diagram and the truth table for half adder is given below:

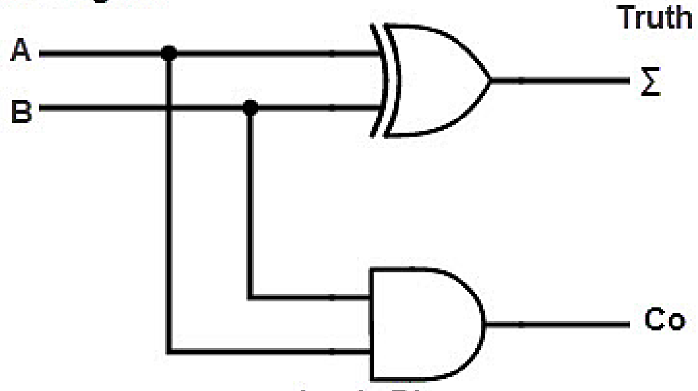


Block Diagram



A	B	Sum (∑)	Carry Out (Co)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Truth Table



Logic Diagram

The i/o ports needed to be declared for the formation of half adder is given below:

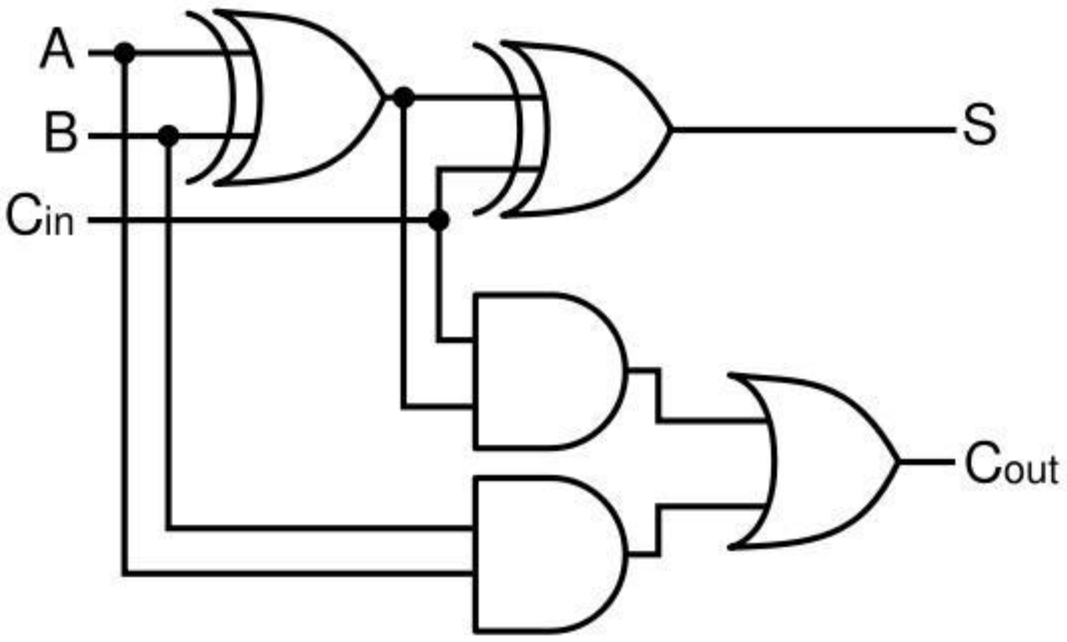
Port Name	INPUT/OUTPUT	Bus
A	In	No
B	In	No
Sum	Out	No
Cout	Out	No

NB: Use temporary variable where ever necessary.

FULL ADDER USING BASIC GATES:

Full adder is a digital circuit used to calculate the sum of three binary bits which is the main difference between full adder and half adder. Full adders are complex and difficult to implement when compared to half adders. The additional third bit is carry bit from the previous stage and is called Carry-in, generally represented by C_{in}. It calculates the sum of three bits along with the carry. The output carry is called Carry-out and is represented by C_{out}.

The circuit diagram of a full adder and its truth table is given below:



TRUTH TABLE:

A	B	C	SUM	CARRY
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

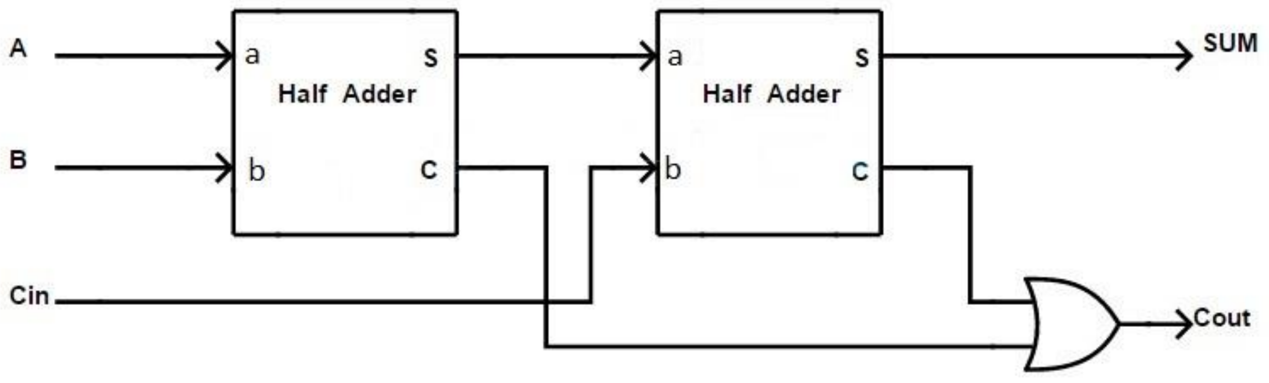
The i/o ports needed to be declared for the formation of full adder is given below:

Port Name	INPUT/OUTPUT	Bus
A	In	No
B	In	No
Cin	In	No
Sum	Out	No
Cout	Out	No

NB: Use temporary variable where ever necessary.

FULL ADDER USING HALF ADDERS AS COMPONENTS:

The block diagram of a full adder constructed using half adder is given below:



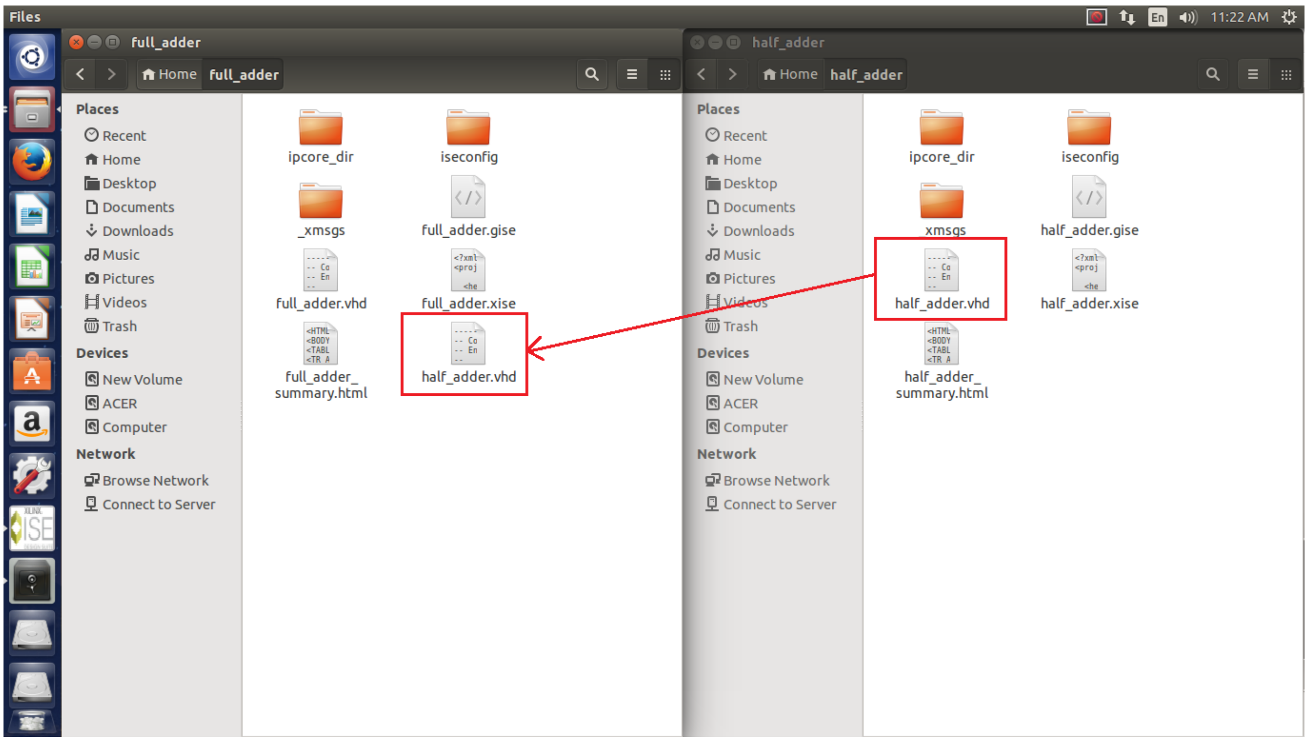
The i/o ports needed to be declared for the formation of full adder is given below:

Port Name	INPUT/OUTPUT	Bus
A	In	No
B	In	No
Cin	In	No
Sum	Out	No
Cout	Out	No

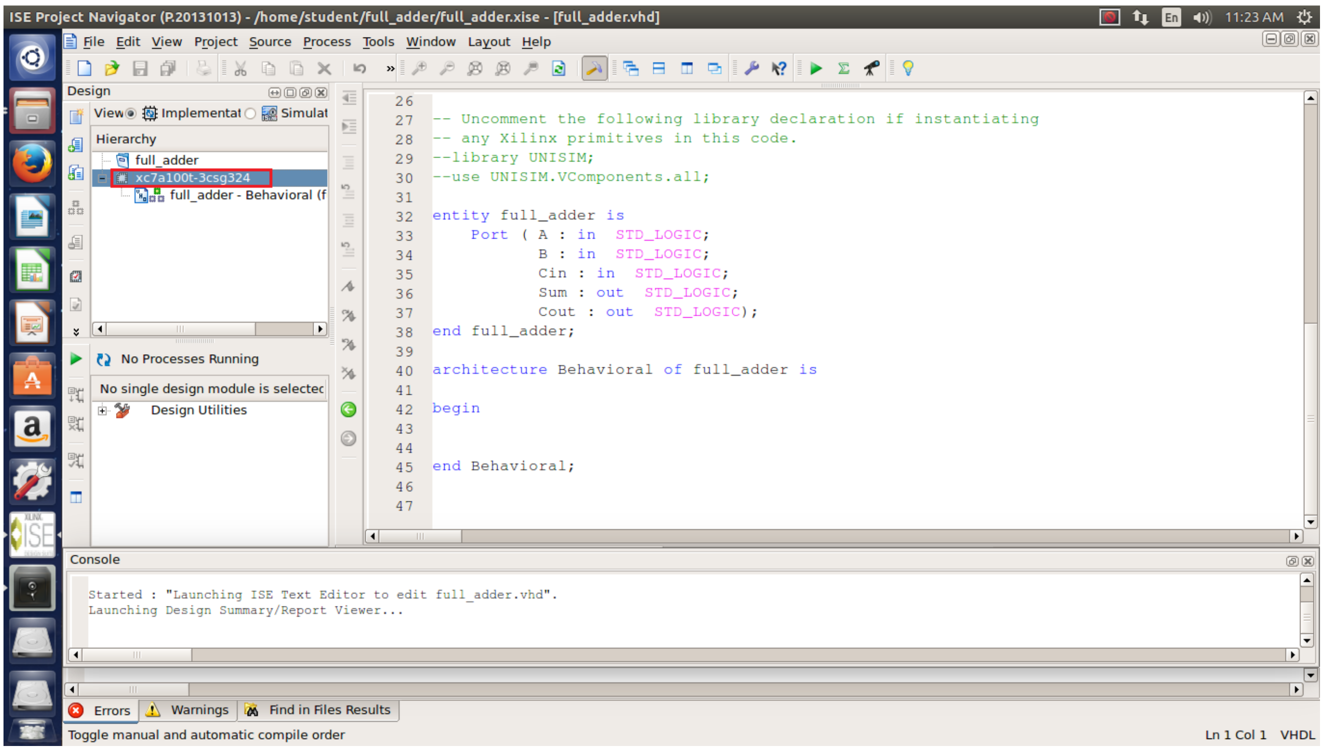
NB: Use temporary variable where ever necessary.

How to add component and map your port from full adder to half adder

Step 1: To add the half adder component, you can use the half adder **.vhd** file from your previous assignment and copy it into the full adder folder.



Step 2: And then right click on the **parent of your full_adder-Behavioral** and select **add source.** *The place to click is marked in the image below.*


ISE Project Navigator (P.20131013) - /home/student/full_adder/full_adder.xise - [full_adder.vhd]
11:23 AM
File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementat Simulat
Hierarchy
full_adder
xc7a100t-3csg324
full_adder - Behavioral (f
No Processes Running
No single design module is selectec
Design Utilities
Console
Errors
Warnings
Find in Files Results
Toggle manual and automatic compile order
Ln 1 Col 1 VHDL

```
-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity full_adder is
    Port ( A : in  STD_LOGIC;
           B : in  STD_LOGIC;
           Cin : in  STD_LOGIC;
           Sum : out  STD_LOGIC;
           Cout : out  STD_LOGIC);
end full_adder;

architecture Behavioral of full_adder is

begin


end Behavioral;
```

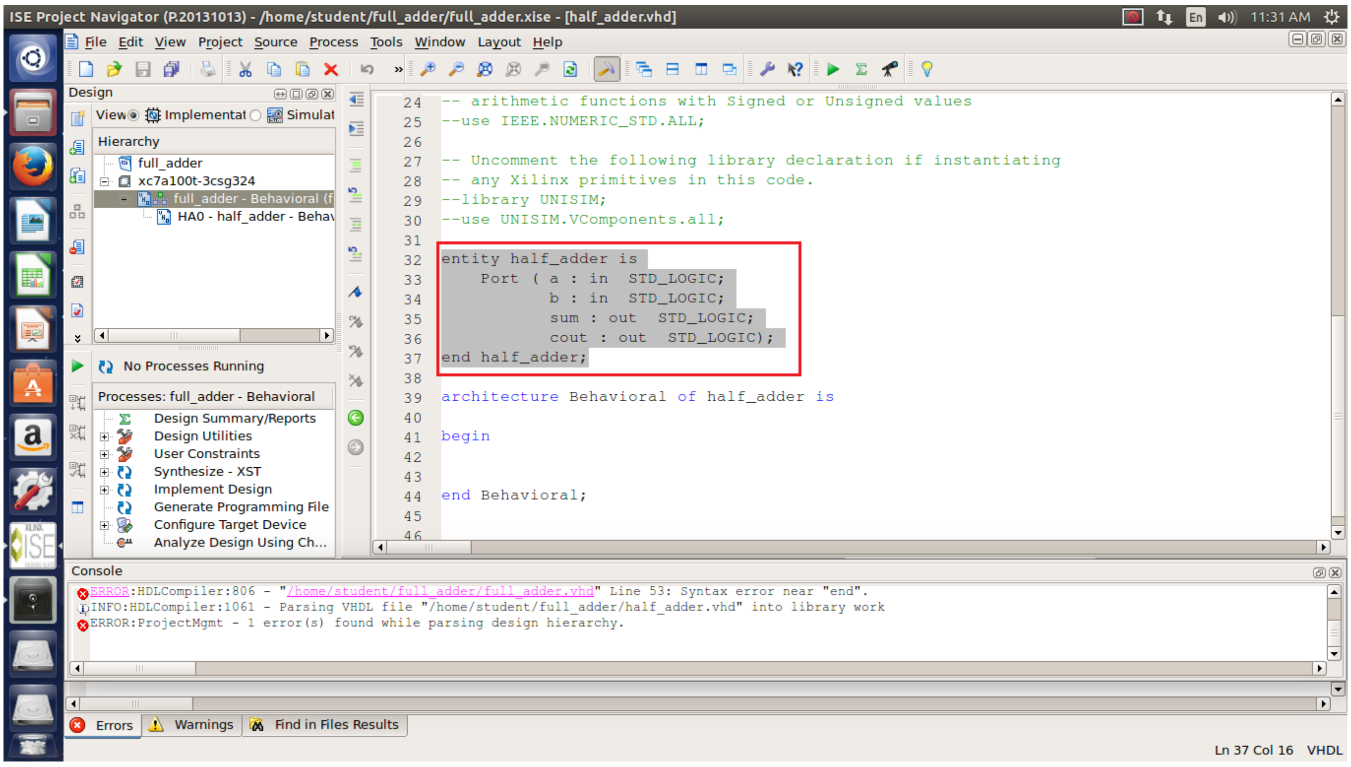
```
Started : "Launching ISE Text Editor to edit full_adder.vhd".
Launching Design Summary/Report Viewer...
```

Step 3: And then browse and select your ***half adder .vhd*** that you copied in your full adder folder and click **open**. In the next window click **ok.**

You can also create a new vhd file for half adder if you don't have one. For that right click on the full_adder-Behavioral and click on new source and then proceed normally to create a half adder entity.

NB – Make sure that the port name for full adder and half adder are different.

Step 4: After adding the half adder .vhd file, ***copy the half adder entity from the half adder's .vhd file and paste it in the behavioral block of the full adder. vhd***, as shown in the below pictures.

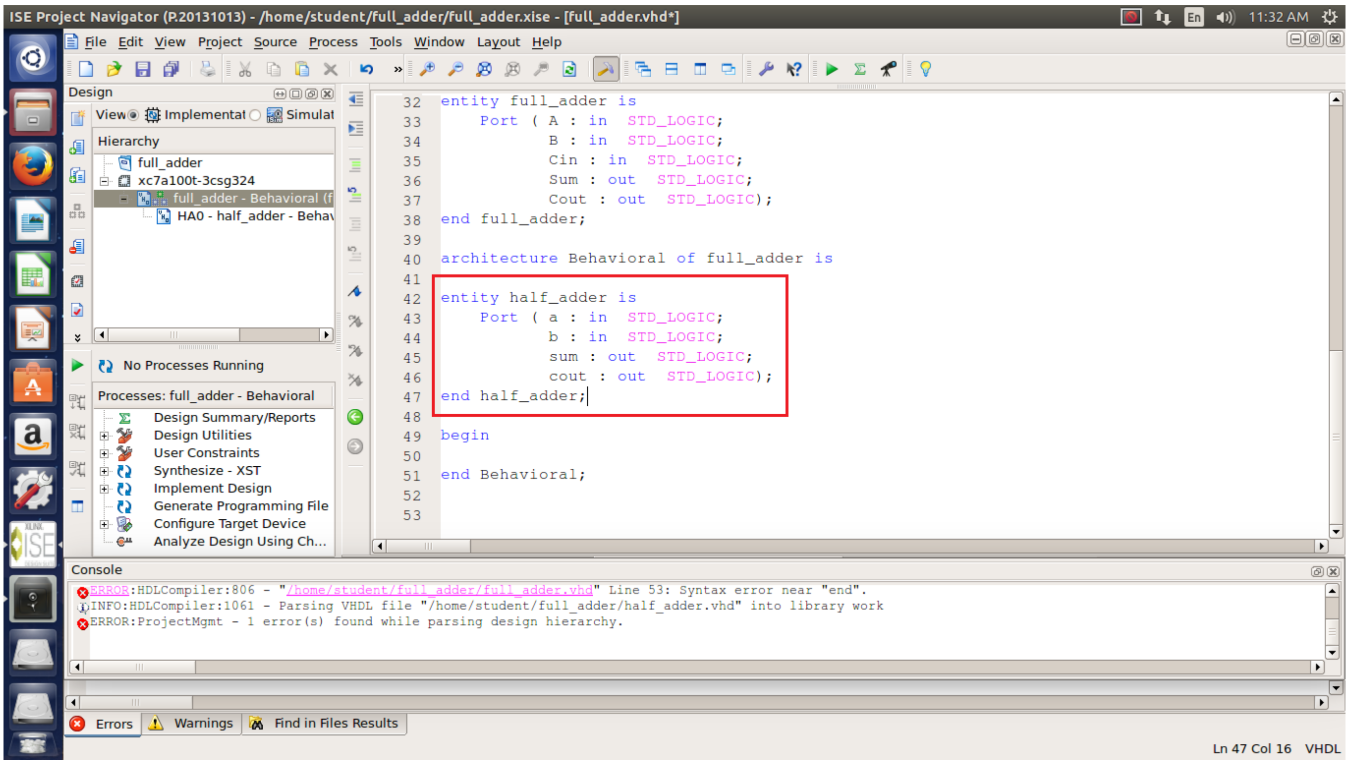


ISE Project Navigator (P.20131013) - /home/student/full_adder/full_adder.xise - [half_adder.vhd]

```
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity half_adder is
    Port ( a : in  STD_LOGIC;
           b : in  STD_LOGIC;
           sum : out  STD_LOGIC;
           cout : out  STD_LOGIC);
end half_adder;

architecture Behavioral of half_adder is

begin


end Behavioral;
```

```
ERROR:HDLCompiler:806 - "/home/student/full_adder/full_adder.vhd" Line 53: Syntax error near "end".
INFO:HDLCompiler:1061 - Parsing VHDL file "/home/student/full_adder/half_adder.vhd" into library work
ERROR:ProjectMgmt - 1 error(s) found while parsing design hierarchy.
```

Ln 37 Col 16 VHDL



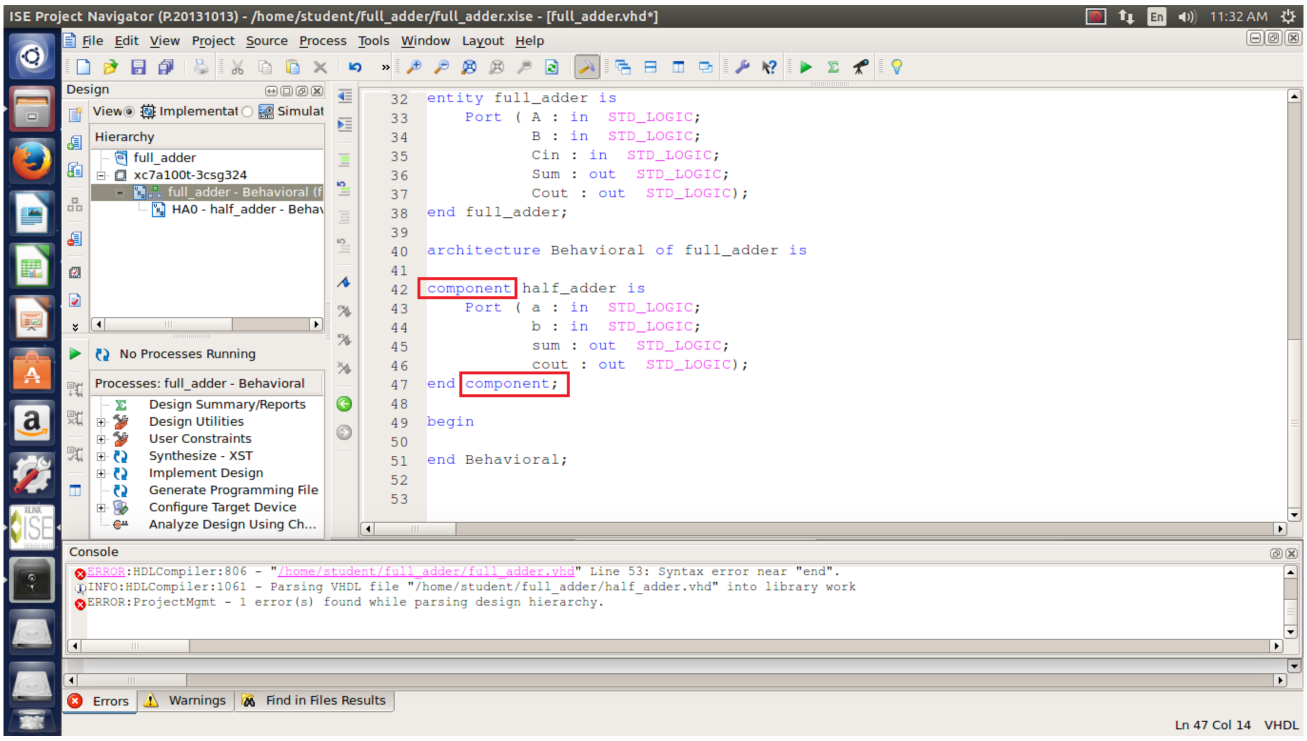
ISE Project Navigator (P.20131013) - /home/student/full_adder/full_adder.xise - [full_adder.vhd*]

```
entity full_adder is
    Port ( A : in  STD_LOGIC;
           B : in  STD_LOGIC;
           Cin : in  STD_LOGIC;
           Sum : out  STD_LOGIC;
           Cout : out  STD_LOGIC);
end full_adder;

architecture Behavioral of full_adder is

entity half_adder is
    Port ( a : in  STD_LOGIC;
           b : in  STD_LOGIC;
           sum : out  STD_LOGIC;
           cout : out  STD_LOGIC);
end half_adder;

begin

end Behavioral;
```

```
ERROR:HDLCompiler:806 - "/home/student/full_adder/full_adder.vhd" Line 53: Syntax error near "end".
INFO:HDLCompiler:1061 - Parsing VHDL file "/home/student/full_adder/half_adder.vhd" into library work
ERROR:ProjectMgmt - 1 error(s) found while parsing design hierarchy.
```

Ln 47 Col 16 VHDL

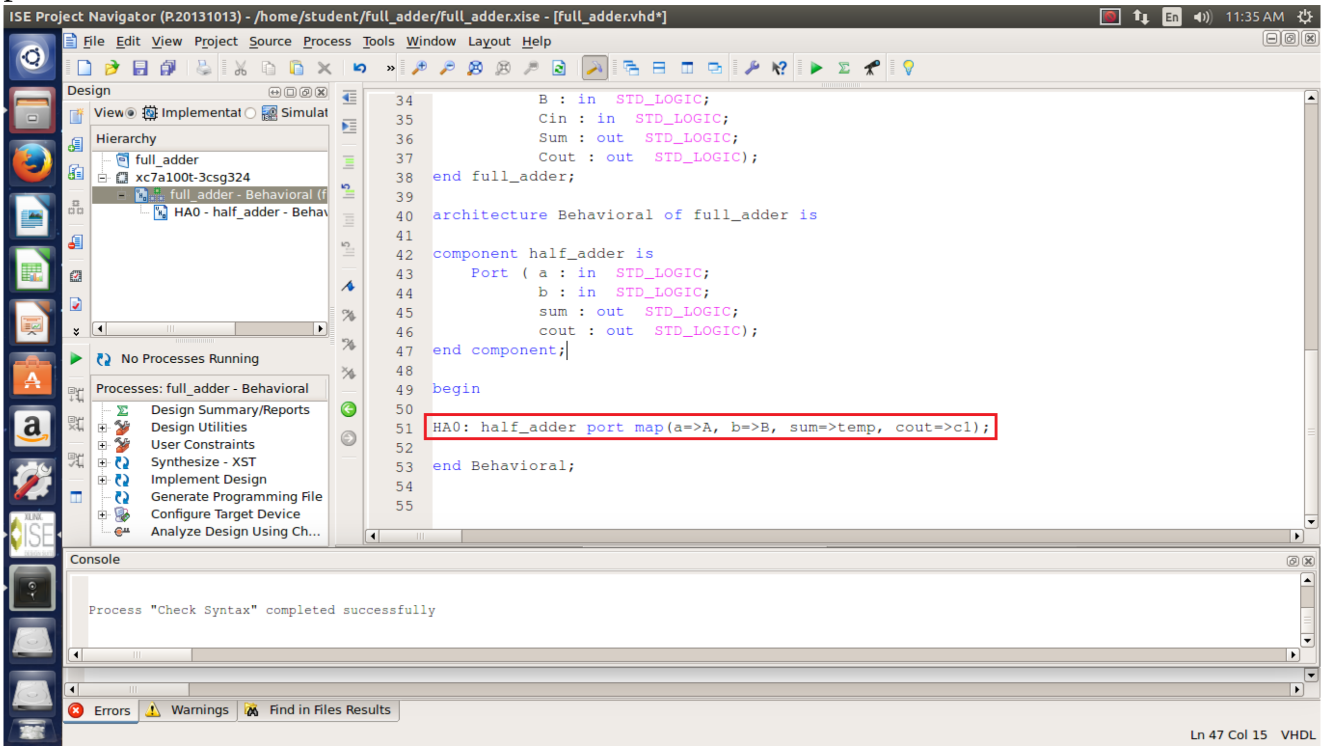
Step 5: After copying the entity block make the changes marked in the image below.


ISE Project Navigator (P.20131013) - /home/student/full_adder/full_adder.xise - [full_adder.vhd*]

```
entity full_adder is
    Port ( A : in  STD_LOGIC;
           B : in  STD_LOGIC;
           Cin : in  STD_LOGIC;
           Sum : out  STD_LOGIC;
           Cout : out  STD_LOGIC);
end full_adder;

architecture Behavioral of full_adder is

component half_adder is
    Port ( a : in  STD_LOGIC;
           b : in  STD_LOGIC;
           sum : out  STD_LOGIC;
           cout : out  STD_LOGIC);
end component;

begin

end Behavioral;
```

```
ERROR:HDLCompiler:806 - "/home/student/full_adder/full_adder.vhd" Line 53: Syntax error near "end".
INFO:HDLCompiler:1061 - Parsing VHDL file "/home/student/full_adder/half_adder.vhd" into library work
ERROR:ProjectMgmt - 1 error(s) found while parsing design hierarchy.
```

Step 6: Now we need to map the ports from half adder to full adder as shown in the picture.


ISE Project Navigator (P.20131013) - /home/student/full_adder/full_adder.xise - [full_adder.vhd*]

```
           B : in  STD_LOGIC;
           Cin : in  STD_LOGIC;
           Sum : out  STD_LOGIC;
           Cout : out  STD_LOGIC);
end full_adder;

architecture Behavioral of full_adder is

component half_adder is
    Port ( a : in  STD_LOGIC;
           b : in  STD_LOGIC;
           sum : out  STD_LOGIC;
           cout : out  STD_LOGIC);
end component;

begin

HA0: half_adder port map(a=>A, b=>B, sum=>temp, cout=>c1);

end Behavioral;
```

```
Process "Check Syntax" completed successfully
```

NB: The general syntax for port map:

Name for the port map: name_of_component port map(port_of_component=>port_of_entity)

The vhdl code for a full adder using 2 half adders will be:

```
entity fulladder_half is
    Port ( A : in  STD_LOGIC;
           B : in  STD_LOGIC;
           Cin : in  STD_LOGIC;
           S : out  STD_LOGIC;
           Cout : out  STD_LOGIC);
end fulladder_half;

architecture Behavioral of fulladder_half is
component halfadder is
    Port ( a : in  STD_LOGIC;
           b : in  STD_LOGIC;
           sum : out  STD_LOGIC;
           c : out  STD_LOGIC);
end component;
 signal temp, c1, c2 : STD_LOGIC := '0';

begin
HA0: halfadder port map(a=>A, b=>B, sum=>temp, c=>c1);
HA1: halfadder port map(a=>temp, b=>Cin, sum=>S, c=>c2);
Cout <= c1 or c2;



end Behavioral;
```