Lab #1

Simple Circuit Design and Testing (Review, Tutorial)

Name: Chen, Kevin

Date: 02/20/2019

Course: CSC 34200-G & CSC 34300-DE

Table of Contents

<u>Objective:</u>

The goal of this lab is to allow the user to apply what they learned in the tutorials of Quartus II into the application of Quartus II. To test what they've learned, they'll construct various diagrams by creating block diagrams/schematic files on Quartus II and forming their diagram. They'll then test the correctness of their diagram by using waveform simulations and comparing their inputs and outputs. In addition, they'll also construct the same diagram using VHDL by either coding it or using Quartus Library of Parameterized Modules. They'll check their correctness by making a testbench file and running it to form a waveform, also comparing the inputs and outputs of their waveform simulation. The various diagrams that will be constructed are 2-to-1 multiplexer, 1-bit half adder, 1-bit full adder using 1-bit half adders as a component, 3-to-8 decoder, and 8-to-3 encoder.

<u>**Functionality and Specifications:**</u>

<u>2-to-1 Multiplexer</u>

A 2-to-1 multiplexer is a circuit that compares three inputs and lets out a result of one output. The inputs are X (SW[0]) assigned to PIN_N25, Y (SW[1]) assigned to PIN_N26, and S (SW[2]) assigned to PIN_P25. The output is M (LEDR[0]) assigned to PIN_AE23.

The truth table, circuit, pin assignments, VHDL code are shown below:

Input			**Output**
Y	**X**	**S**	**M**
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Table 1: Truth Table for 2-to-1 Multiplexer

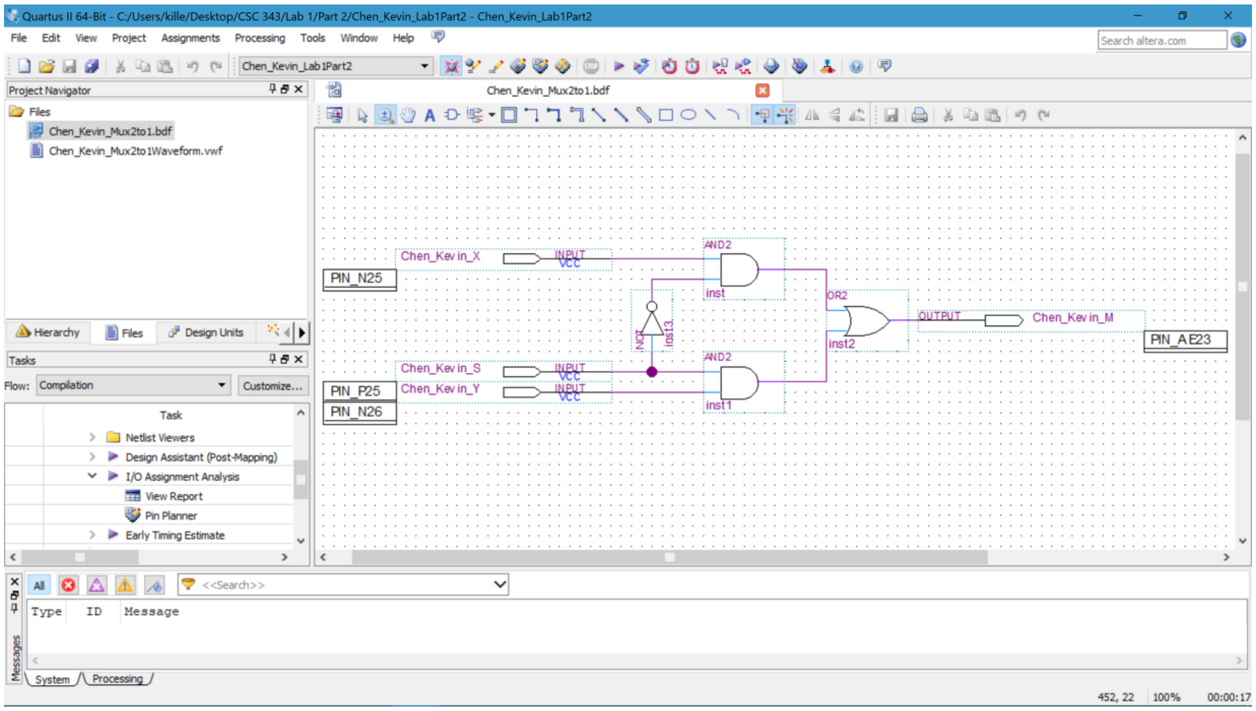


Figure 1: Block Diagram/Schematic File of 2-to-1 Multiplexer (Chen_Kevin_Mux2to1.BDF)

Variable Name	Variable Type	Signal Name	FPGA Pin No.
Chen_Kevin_S	Input	SW[2]	PIN_P25
Chen_Kevin_X	Input	SW[0]	PIN_N25
Chen_Kevin_Y	Input	SW[1]	PIN_N26
Chen_Kevin_M	Output	LEDR[0]	PIN_AE23

Table 2: Pin Assignments for 2-to-1 Multiplexer

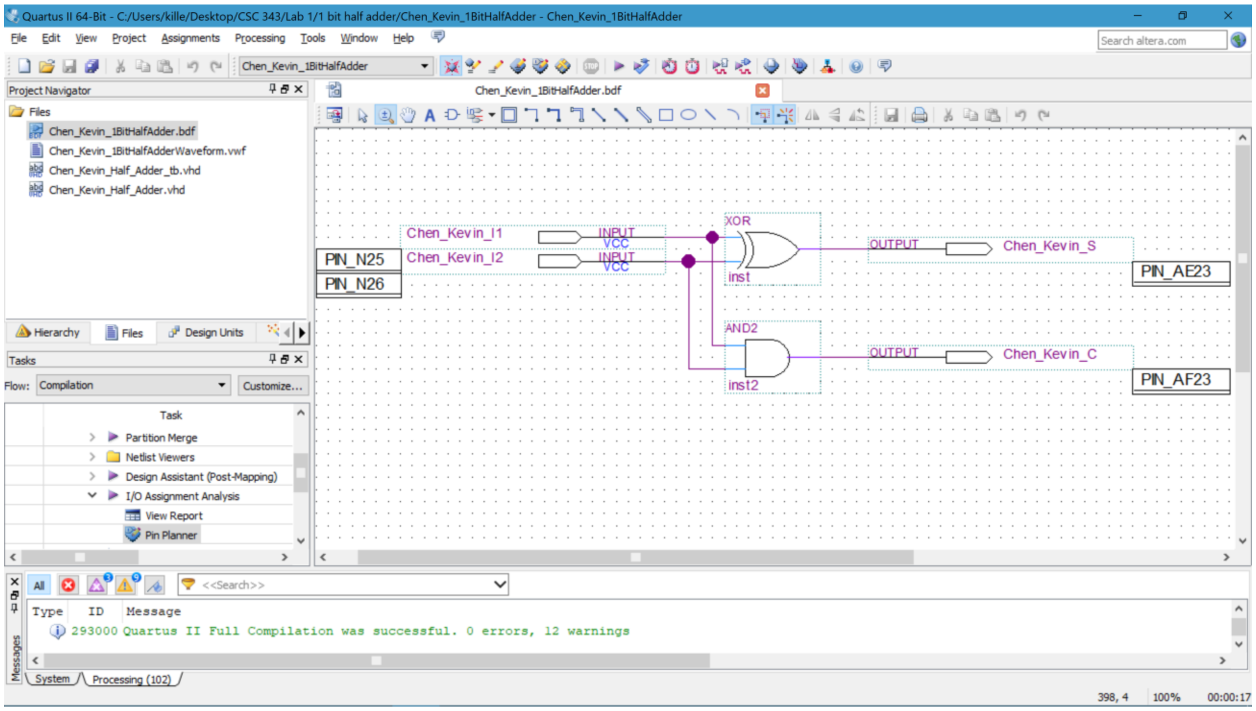
1-bit Half Adder

A 1-bit half adder is a circuit that compares two inputs and lets out a result of two outputs. The inputs are I1 (SW[0]) assigned to PIN_N25, and I2 (SW[1]) assigned to PIN_N26. The outputs are S (LEDR[0]) assigned to PIN_AE23, and C (LEDR[1]) assigned to PIN_AF23.

The truth table, circuit, pin assignments, VHDL code are shown below:

Input		**Output**	
I2	**I1**	**C**	**S**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table 3: Truth Table for 1-bit Half Adder


Quartus II 64-Bit - C:/Users/kille/Desktop/CSC 343/Lab 1/1 bit half adder/Chen_Kevin_1BitHalfAdder - Chen_Kevin_1BitHalfAdder
File Edit View Project Assignments Processing Tools Window Help
Project Navigator
Files
Chen_Kevin_1BitHalfAdder.bdf
Chen_Kevin_1BitHalfAdderWaveform.vwf
Chen_Kevin_Half_Adder_tb.vhd
Chen_Kevin_Half_Adder.vhd
Hierarchy
Files
Design Units
Tasks
Flow: Compilation
Customize...
Task
Partition Merge
Netlist Viewers
Design Assistant (Post-Mapping)
I/O Assignment Analysis
View Report
Pin Planner
Chen_Kevin_1BitHalfAdder.bdf
PIN_N25
PIN_N26
Chen_Kevin_I1
Chen_Kevin_I2
INPUT VCC
INPUT VCC
XOR
inst
AND2
inst2
OUTPUT
Chen_Kevin_S
OUTPUT
Chen_Kevin_C
PIN_AE23
PIN_AF23
System
Processing (102)
398, 4
100%
00:00:17

```
Type  ID     Message
      293000 Quartus II Full Compilation was successful. 0 errors, 12 warnings
```

Figure 2: Block Diagram/Schematic File of 1-bit Half Adder (Chen_Kevin_1BitHalfAdder.BDF)

Variable Name	**Variable Type**	**Signal Name**	**FPGA Pin No.**
Chen_Kevin_I1	Input	SW[0]	PIN_N25
Chen_Kevin_I2	Input	SW[1]	PIN_N26
Chen_Kevin_S	Output	LEDR[0]	PIN_AE23
Chen_Kevin_C	Output	LEDR[1]	PIN_AF23

Table 4: Pin Assignments for 1-bit Half Adder

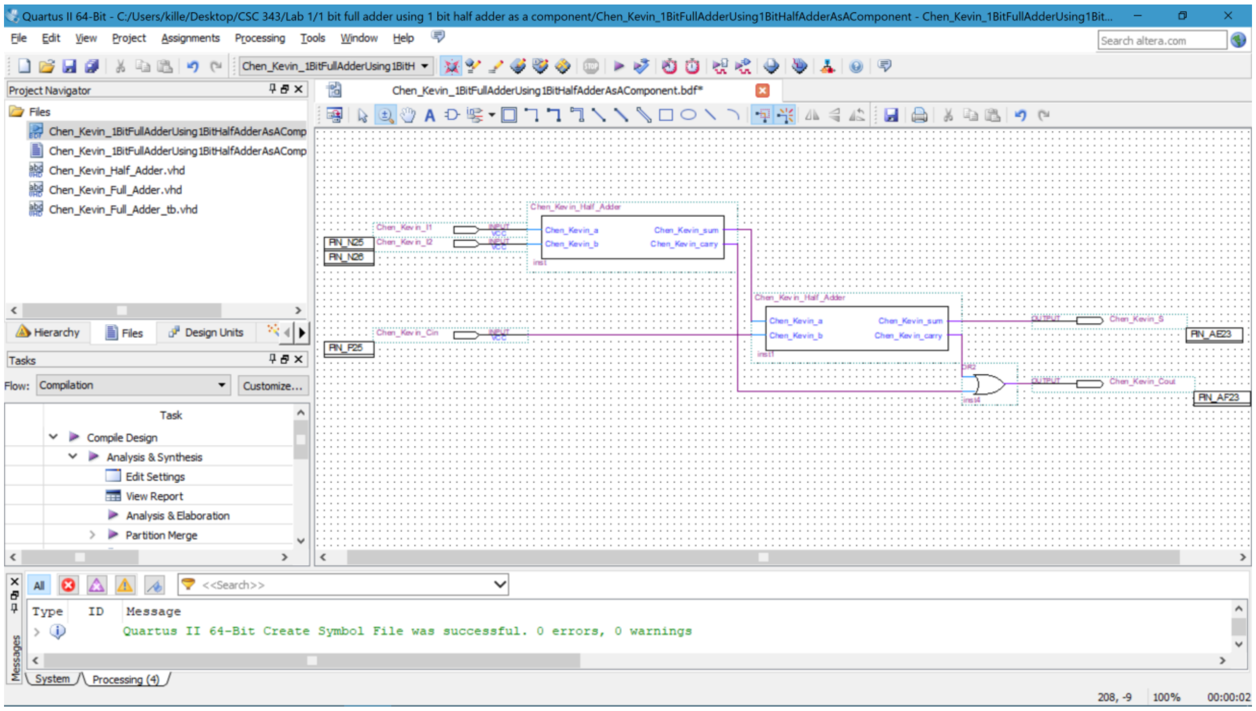
<u>1-bit Full Adder Using 1-bit Half Adder as a Component</u>

A 1-bit full adder is a circuit that compares three inputs and lets out a result of two outputs. The inputs are I1 (SW[0]) assigned to PIN_N25, I2 (SW[1]) assigned to PIN_N26, and Cin (SW[2]) assigned to PIN_P25. The outputs are S (LEDR[0]) assigned to PIN_AE23, and Cout (LEDR[1]) assigned to PIN_AF23. It's made by using two 1-bit half adders and connecting them together using an "or" gate for the output of Cout.

The truth table, circuit, pin assignments, VHDL code are shown below:

Input			**Output**	
Cin	**I2**	**I1**	**S**	**Cout**
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

Table 5: Truth Table for 1-bit Full Adder


Quartus II 64-Bit - C:/Users/kille/Desktop/CSC 343/Lab 1/1 bit full adder using 1 bit half adder as a component/Chen_Kevin_1BitFullAdderUsing1BitHalfAdderAsAComponent - Chen_Kevin_1BitFullAdderUsing1Bit...
Project Navigator
Files
Chen_Kevin_1BitFullAdderUsing1BitHalfAdderAsAComp
Chen_Kevin_1BitFullAdderUsing1BitHalfAdderAsAComp
Chen_Kevin_Half_Adder.vhd
Chen_Kevin_Full_Adder.vhd
Chen_Kevin_Full_Adder_tb.vhd
Chen_Kevin_1BitFullAdderUsing1BitHalfAdderAsAComponent.bdf*
Chen_Kevin_Half_Adder
Chen_Kevin_a
Chen_Kevin_b
Chen_Kevin_sum
Chen_Kevin_carry
PIN_N25
PIN_N26
PIN_P25
PIN_AE23
PIN_AF23
Chen_Kevin_I1
Chen_Kevin_I2
Chen_Kevin_Cin
Chen_Kevin_S
Chen_Kevin_Cout
Tasks
Flow: Compilation
Compile Design
Analysis & Synthesis
Edit Settings
View Report
Analysis & Elaboration
Partition Merge
System
Processing (4)

```
Type  ID  Message
          Quartus II 64-Bit Create Symbol File was successful. 0 errors, 0 warnings
```

Figure 3: Block Diagram/Schematic File of 1-bit Full Adder

(Chen_Kevin_1BitFullAdderUsing1BitHalfAdderAsAComponent.BDF)

Variable Name	**Variable Type**	**Signal Name**	**FPGA Pin No.**
Chen_Kevin_I1	Input	SW[0]	PIN_N25
Chen_Kevin_I2	Input	SW[1]	PIN_N26
Chen_Kevin_Cin	Input	SW[2]	PIN_P25
Chen_Kevin_S	Output	LEDR[0]	PIN_AE23
Chen_Kevin_Cout	Output	LEDR[1]	PIN_AF23

Table 6: Pin Assignments for 1-bit Full Adder

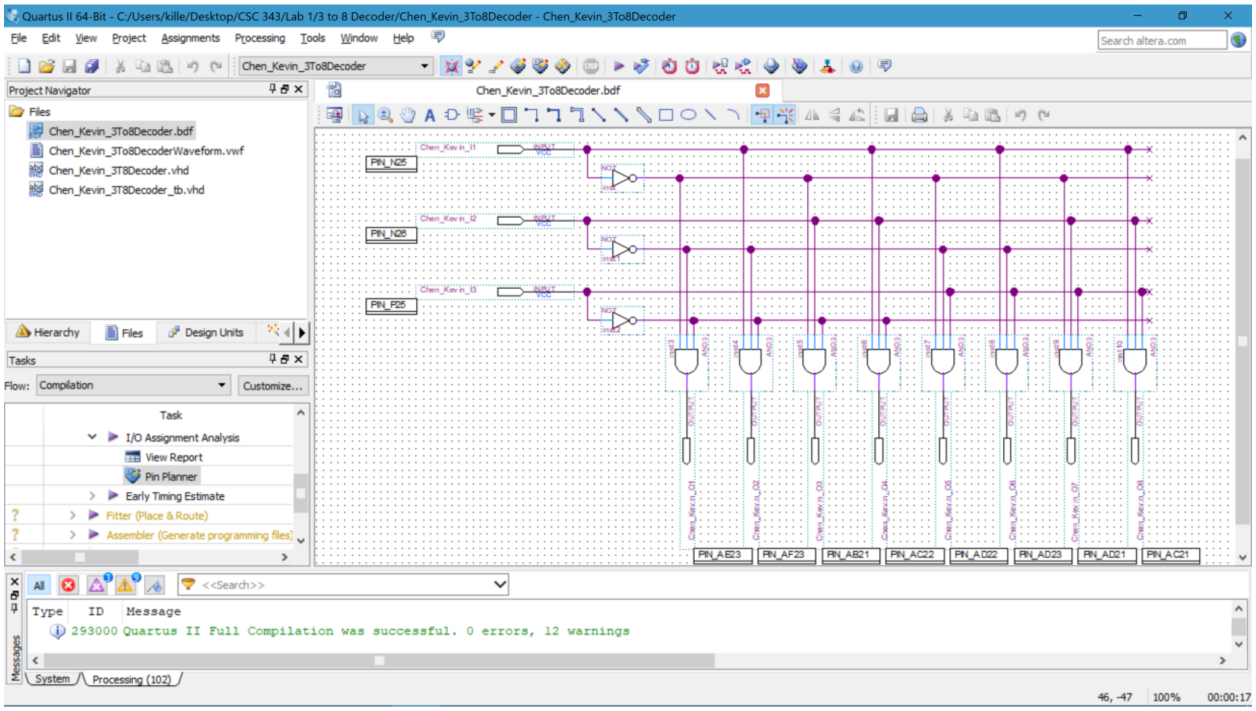
3-to-8 Decoder

A 3-to-8 Decoder is a circuit that compares three inputs and lets out a result of eight outputs. The inputs are I1 (SW[0]) assigned to PIN_N25, I2 (SW[1]) assigned to PIN_N26, and I3 (SW[2]) assigned to PIN_P25. The outputs are O1 (LEDR[0]) assigned to PIN_AE23, O2 (LEDR[1]) assigned to PIN_AF23, O3 (LEDR[2]) assigned to PIN_AB21, O4 (LEDR[3]) assigned to PIN_AC22, O5 (LEDR[4]) assigned to PIN_AD22, O6 (LEDR[5]) assigned to PIN_AD23, O7 (LEDR[6]) assigned to PIN_AD21, and O8 (LEDR[7]) assigned to PIN_AC21.

The truth table, circuit, pin assignments, VHDL code are shown below:

Input			**Output**							
I3	**I2**	**I1**	**O1**	**O2**	**O3**	**O4**	**O5**	**O6**	**O7**	**O8**
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Table 7: Truth Table for 3-to-8 Decoder



```
Type   ID     Message
       293000 Quartus II Full Compilation was successful. 0 errors, 12 warnings
```

Figure 4: Block Diagram/Schematic File of 3-to-8 Decoder (Chen_Kevin_3To8Decoder.BDF)

Variable Name	**Variable Type**	**Signal Name**	**FPGA Pin No.**
Chen_Kevin_I1	Input	SW[0]	PIN_N25
Chen_Kevin_I2	Input	SW[1]	PIN_N26
Chen_Kevin_I3	Input	SW[2]	PIN_P25
Chen_Kevin_O1	Output	LEDR[0]	PIN_AE23
Chen_Kevin_O2	Output	LEDR[1]	PIN_AF23
Chen_Kevin_O3	Output	LEDR[2]	PIN_AB21
Chen_Kevin_O4	Output	LEDR[3]	PIN_AC22
Chen_Kevin_O5	Output	LEDR[4]	PIN_AD22
Chen_Kevin_O6	Output	LEDR[5]	PIN_AD23
Chen_Kevin_O7	Output	LEDR[6]	PIN_AD21
Chen_Kevin_O8	Output	LEDR[7]	PIN_AC21

Table 8: Pin Assignments for 3-to-8 Decoder

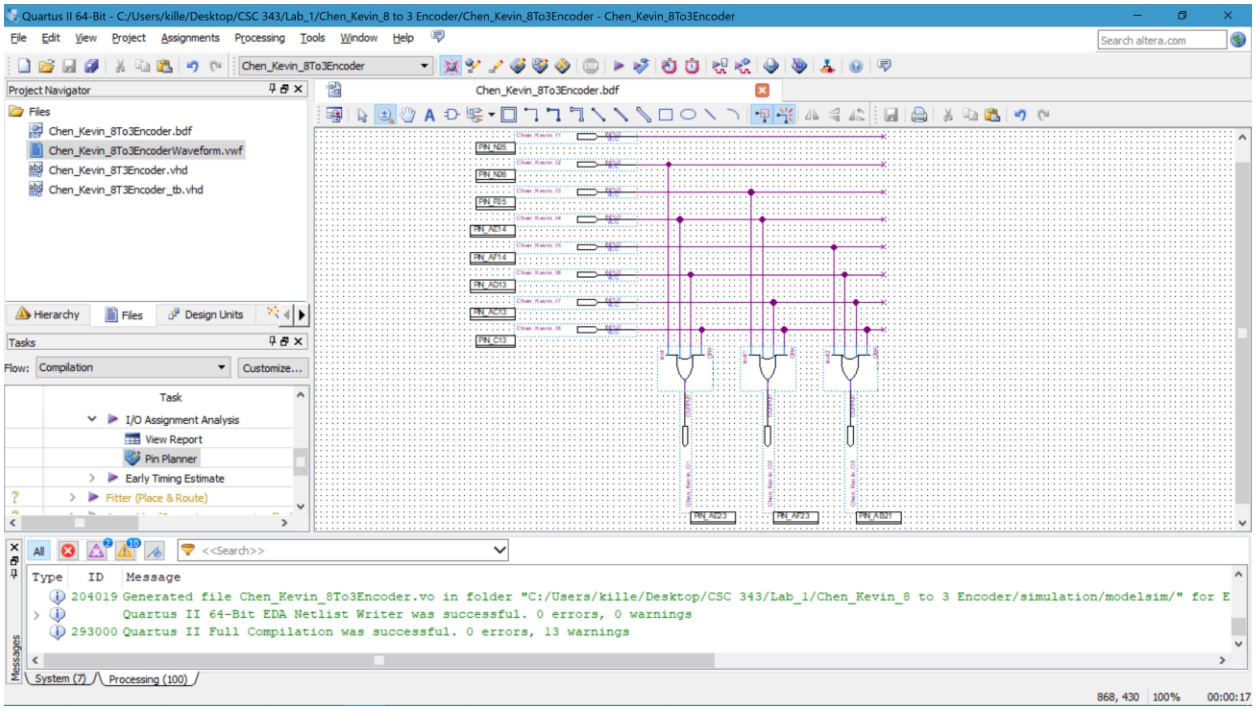
8-to-3 Encoder

A 8-to-3 Encoder is a circuit that compares eight inputs and lets out a result of three outputs. The inputs are I1 (SW[0]) assigned to PIN_N25, I2 (SW[1]) assigned to PIN_N26, I3 (SW[2]) assigned to PIN_P25, I4 (SW[3]) assigned to PIN_AE14, I5 (SW[4]) assigned to PIN_AF14, I6 (SW[5]) assigned to PIN_AD13, I7 (SW[6]) assigned to PIN_AC13, and I8 (SW[7]) assigned to PIN_C13. The outputs are O1 (LEDR[0]) assigned to PIN_AE23, O2 (LEDR[1]) assigned to PIN_AF23, and O3 (LEDR[2]) assigned to PIN_AB21.

The truth table, circuit, pin assignments, VHDL code are shown below:

Input								**Output**		
I8	**I7**	**I6**	**I5**	**I4**	**I3**	**I2**	**I1**	**O1**	**O2**	**O3**
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	1	1	0
0	0	0	1	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	0	1	1
1	0	0	0	0	0	0	0	1	1	1

Table 9: Truth Table for 8-to-3 Encoder


Quartus II 64-Bit - C:/Users/kille/Desktop/CSC 343/Lab_1/Chen_Kevin_8 to 3 Encoder/Chen_Kevin_8To3Encoder - Chen_Kevin_8To3Encoder
File Edit View Project Assignments Processing Tools Window Help
Project Navigator
Files
Chen_Kevin_8To3Encoder.bdf
Chen_Kevin_8To3EncoderWaveform.vwf
Chen_Kevin_8T3Encoder.vhd
Chen_Kevin_8T3Encoder_tb.vhd
Hierarchy
Files
Design Units
Tasks
Flow: Compilation
Customize...
Task
I/O Assignment Analysis
View Report
Pin Planner
Early Timing Estimate
Fitter (Place & Route)
Chen_Kevin_8To3Encoder.bdf
System (7)
Processing (100)
868, 430
100%
00:00:17

```
Type  ID     Message
      204019 Generated file Chen_Kevin_8To3Encoder.vo in folder "C:/Users/kille/Desktop/CSC 343/Lab_1/Chen_Kevin_8 to 3 Encoder/simulation/modelsim/" for E
             Quartus II 64-Bit EDA Netlist Writer was successful. 0 errors, 0 warnings
      293000 Quartus II Full Compilation was successful. 0 errors, 13 warnings
```

Figure 5: Block Diagram/Schematic File of 8-to-3 Encoder (Chen_Kevin_8To3Encoder.BDF)

Variable Name	**Variable Type**	**Signal Name**	**FPGA Pin No.**
Chen_Kevin_I1	Input	SW[0]	PIN_N25
Chen_Kevin_I2	Input	SW[1]	PIN_N26
Chen_Kevin_I3	Input	SW[2]	PIN_P25
Chen_Kevin_I4	Input	SW[3]	PIN_AE14
Chen_Kevin_I5	Input	SW[4]	PIN_AF14
Chen_Kevin_I6	Input	SW[5]	PIN_AD13
Chen_Kevin_I7	Input	SW[6]	PIN_AC13
Chen_Kevin_I8	Input	SW[7]	PIN_C13
Chen_Kevin_O1	Output	LEDR[0]	PIN_AE23
Chen_Kevin_O2	Output	LEDR[1]	PIN_AF23
Chen_Kevin_O3	Output	LEDR[2]	PIN_AB21

Table 10: Pin Assignments for 8-to-3 Encoder

Simulation:

2-to-1 Multiplexer

The functionality/behavior of the 2-to-1 multiplexer is to follow the truth table (Table 1), which describes exactly what the output is supposed to be given a certain input.

A waveform was created using the block diagram/schematic file for 2-to-1 multiplexer (Figure 6) and it was given inputs to test all possible combinations. The outputs that it generated matches those from table 1, therefore, confirming the correctness of the block diagram/schematic file.

A waveform was created using the VHDL code provided by LPM for 2-to-1 multiplexer (Figure 7) and it was given inputs to match those from figure 6. The outputs that it generated matches figure 6, therefore, confirming the correctness of the VHDL code provided by LPM.

A ModelSim waveform was created using the testbench written in VHDL for 2-to-1 multiplexer (Figure 8). The testbench written provided the inputs with the same values at the same time as the block diagram/schematic file. The wave generated matches the same output behavior obtained from figure 6 and figure 7, therefore, confirming the correctness of the VHDL code.

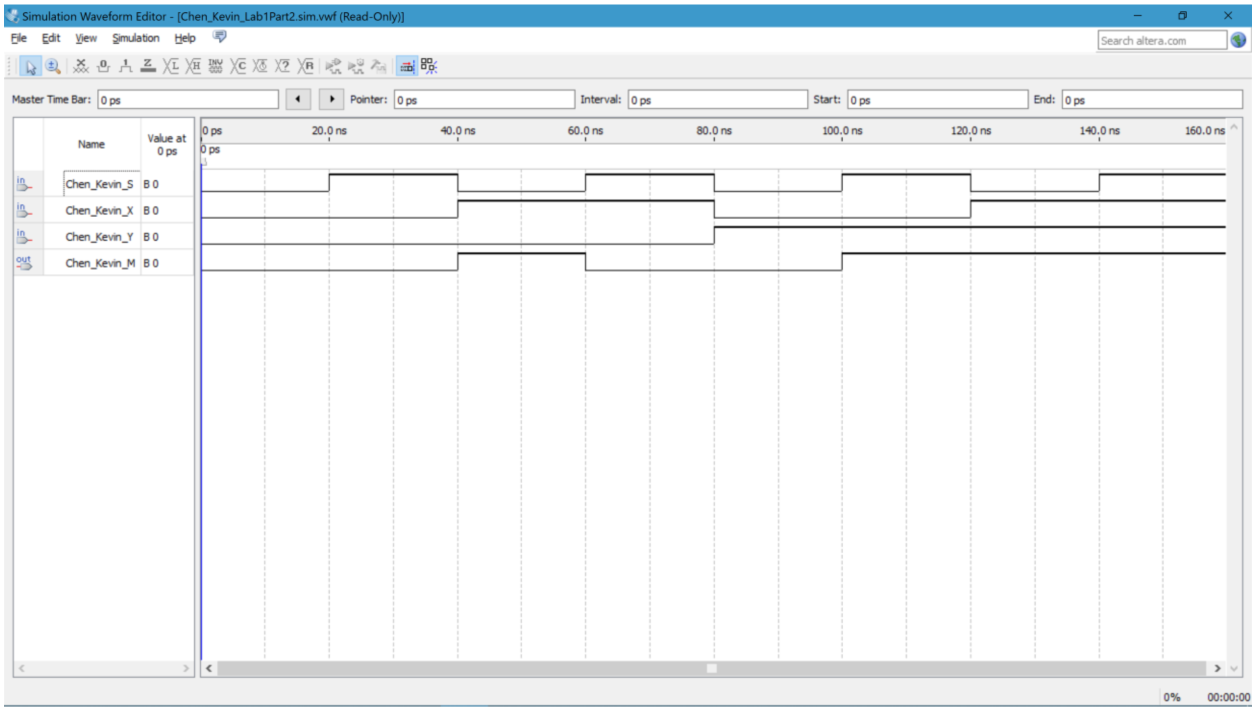


Figure 6: Waveform Simulation of Block Diagram/Schematic File of 2-to-1 Multiplexer (Chen_Kevin_Mux2to1Waveform.VWF)

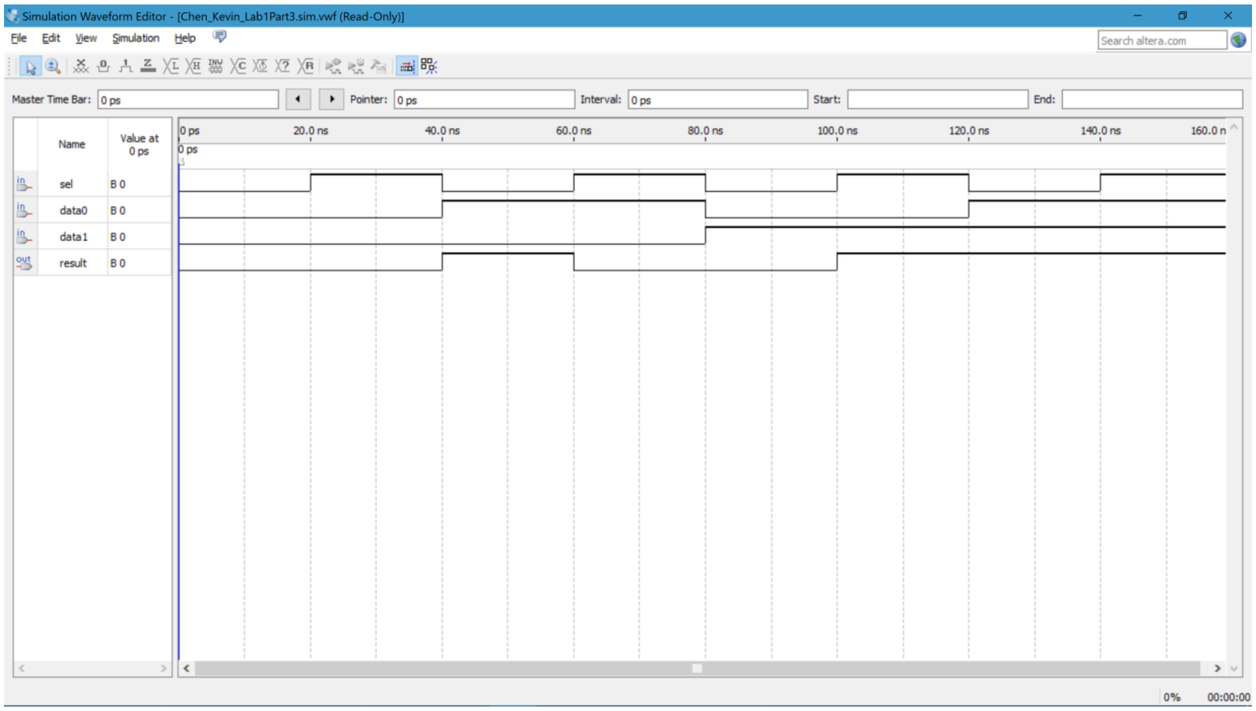


Figure 7: Waveform Simulation of LPM's VHDL Code for 2-to-1 Multiplexer

(Chen_Kevin_Lab1Part3Waveform.VWF)

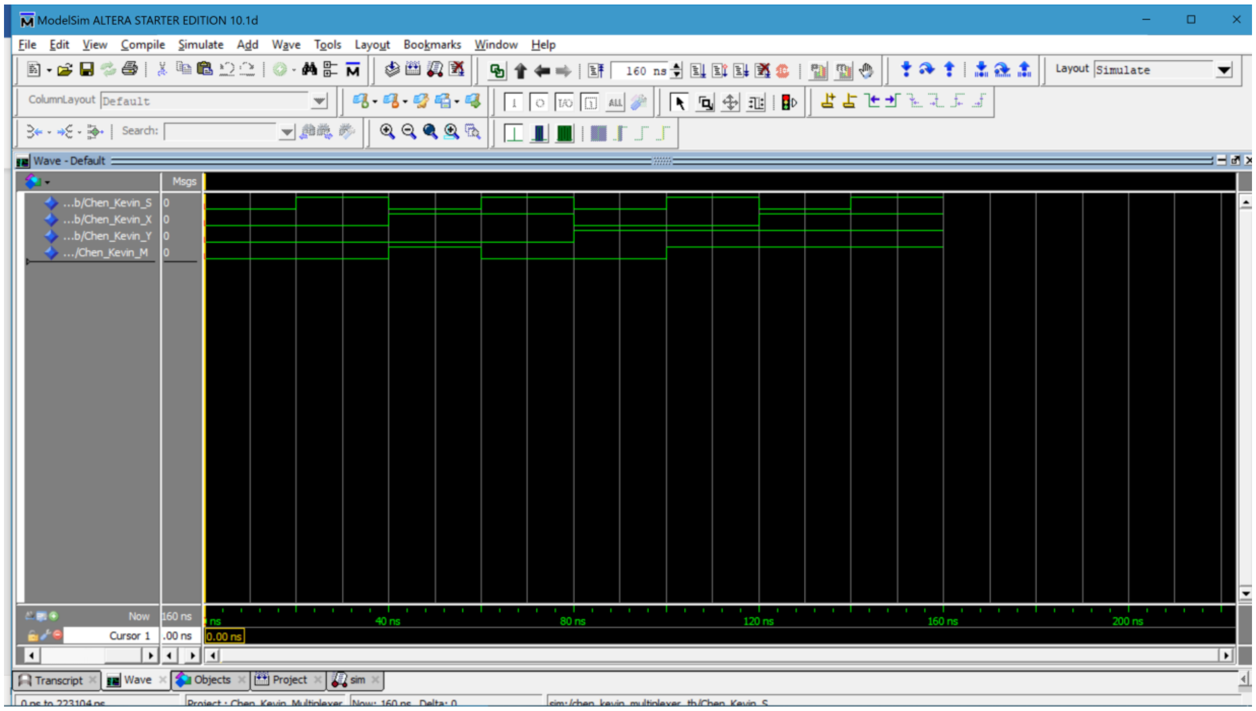


Figure 8: ModelSim of VHDL Code for 2-to-1 Multiplexer

<u>1-bit Half Adder</u>

The functionality/behavior of the 1-bit half adder is to follow the truth table (Table 3), which describes exactly what the output is supposed to be given a certain input.

A waveform was created using the block diagram/schematic file for 1-bit half adder (Figure 9) and it was given inputs to test all possible combinations. The outputs that it generated matches those from table 3, therefore, confirming the correctness of the block diagram/schematic file.

A waveform was created using the VHDL code provided by LPM for 1-bit half adder (Figure 10) and it was given inputs to match those from figure 9. The outputs that it generated matches figure 9, therefore, confirming the correctness of the VHDL code provided by LPM.

A ModelSim waveform was created using the testbench written in VHDL for 1-bit half adder (Figure 11). The testbench written provided the inputs with the same values at the same time as the block diagram/schematic file. The wave generated matches the same output behavior obtained from figure 9 and figure 10, therefore, confirming the correctness of the VHDL code.

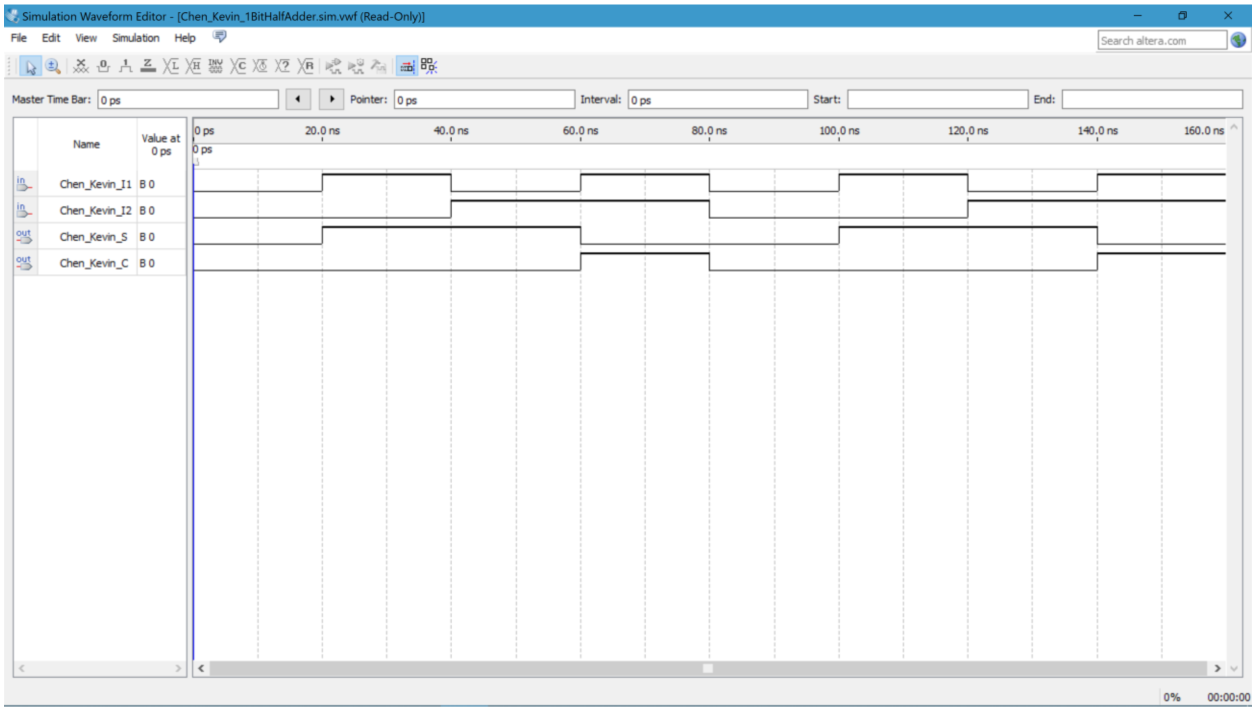


Figure 9: Waveform Simulation of Block Diagram/Schematic File of 1-bit Half Adder

(Chen_Kevin_1BitHalfAdderWaveform.VWF)

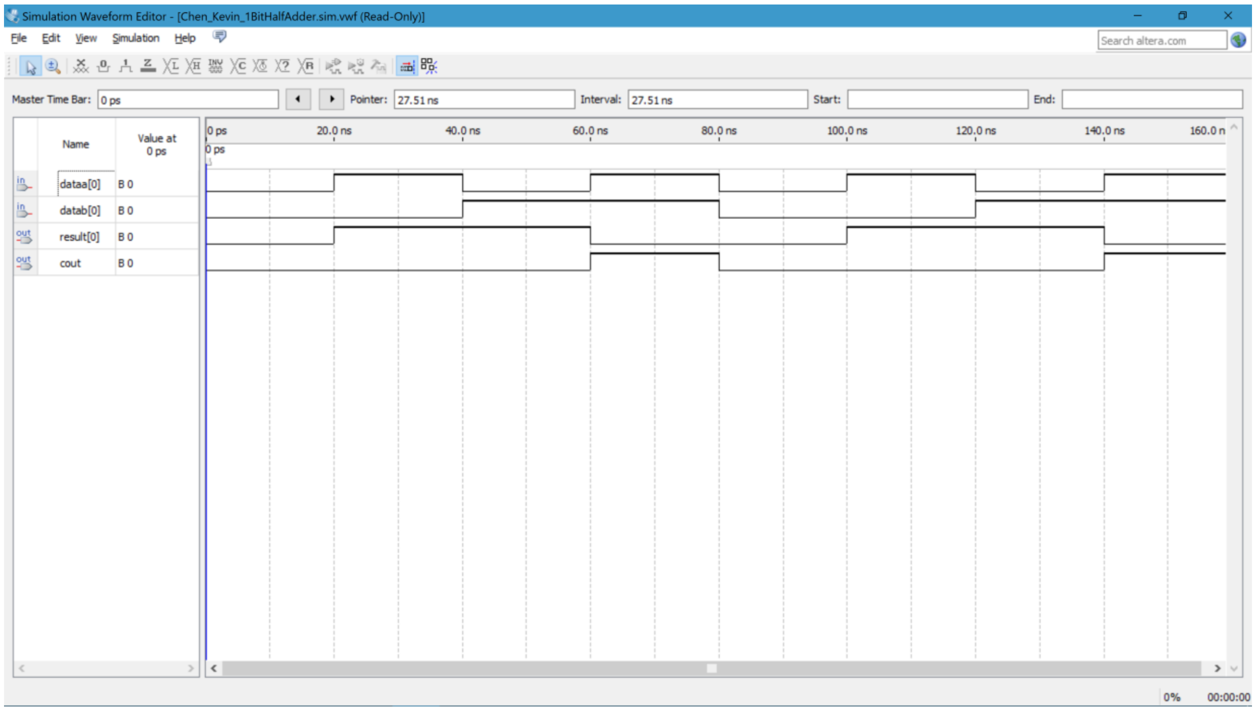


Figure 10: Waveform Simulation of LPM's VHDL Code for 1-bit Half Adder

(Chen_Kevin_HalfAdderLPMWaveform.VWF)

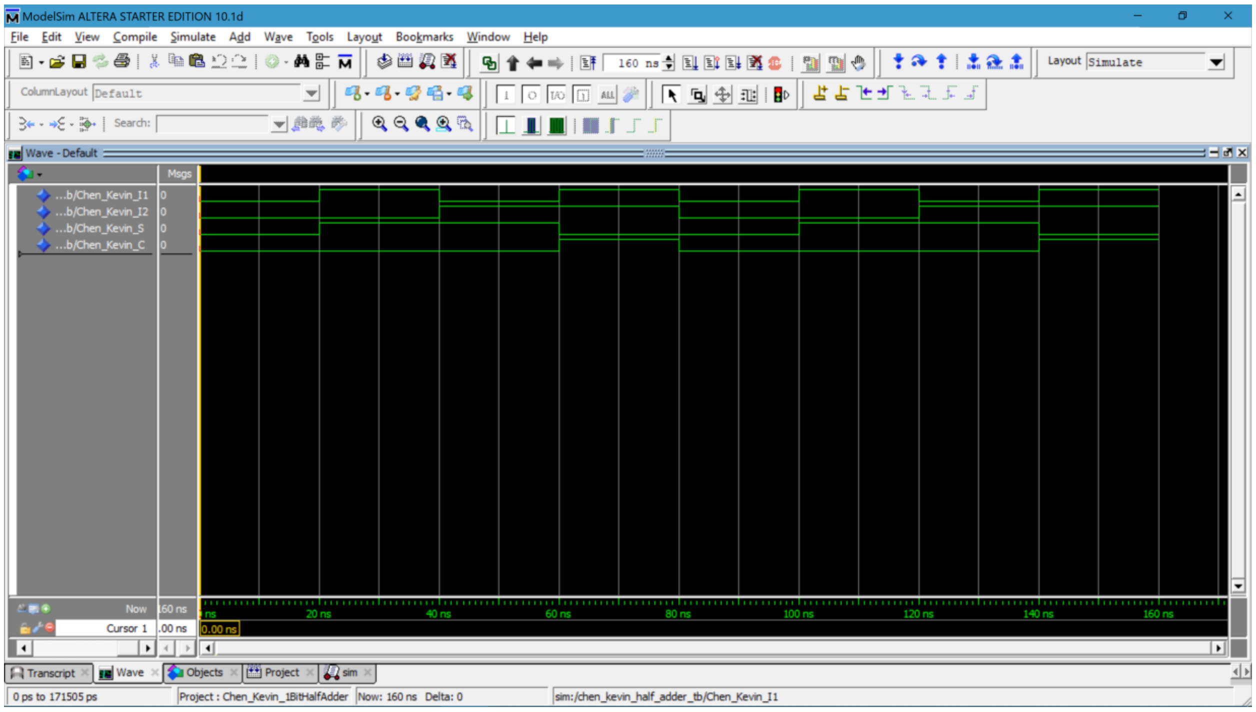


Figure 11: ModelSim of VHDL Code for 1-bit Half Adder

1-bit Full Adder Using 1-bit Half Adder as a Component

The functionality/behavior of the 1-bit full adder is to follow the truth table (Table 5), which describes exactly what the output is supposed to be given a certain input.

A waveform was created using the block diagram/schematic file 1-bit full adder (Figure 12) and it was given inputs to test all possible combinations. The outputs that it generated matches those from table 5, therefore, confirming the correctness of the block diagram/schematic file.

A waveform was created using the VHDL code provided by LPM for 1-bit full adder (Figure 13) and it was given inputs to match those from figure 12. The outputs that it generated matches figure 12, therefore, confirming the correctness of the VHDL code provided by LPM.

A ModelSim waveform was created using the testbench written in VHDL for 1-bit full adder (Figure 14). The testbench written provided the inputs with the same values at the same time as the block diagram/schematic file. The wave generated matches the same output behavior obtained from figure 12 and figure 13, therefore, confirming the correctness of the VHDL code.

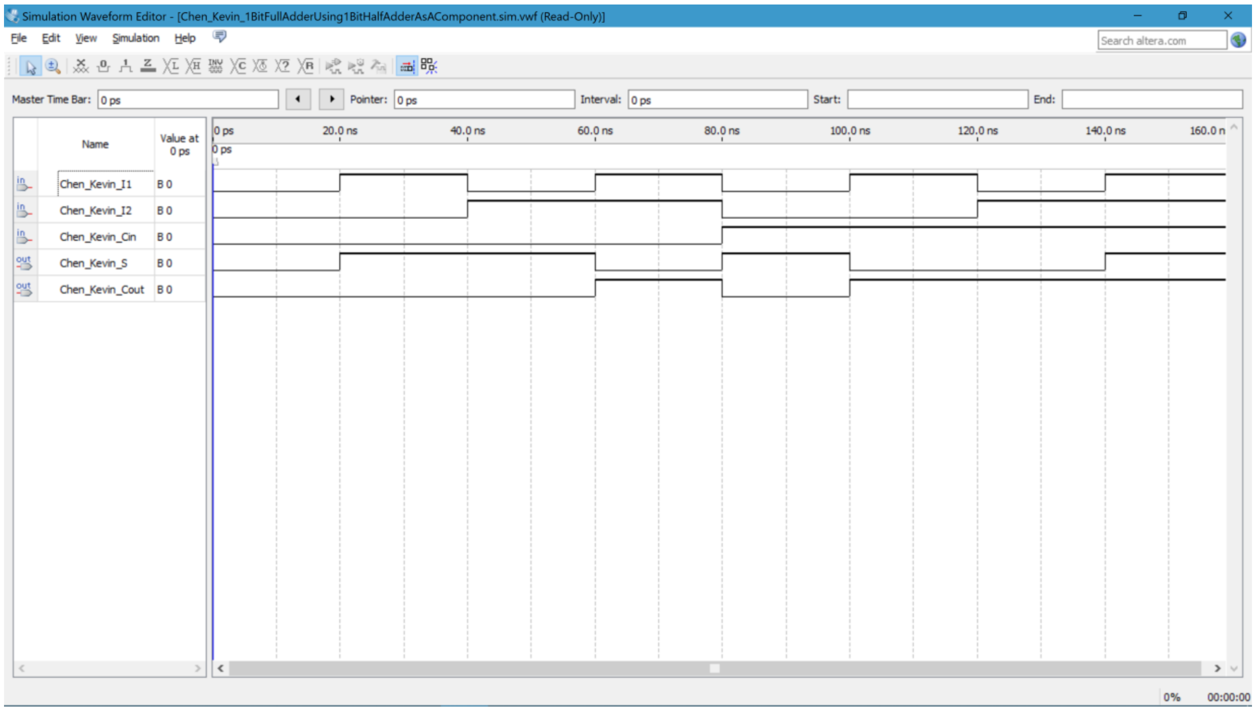


Figure 12: Waveform Simulation of Block Diagram/Schematic File of 1-bit Full Adder

(Chen_Kevin_1BitFullAdderUsing1BitHalfAdderAsAComponentWaveform.VWF)

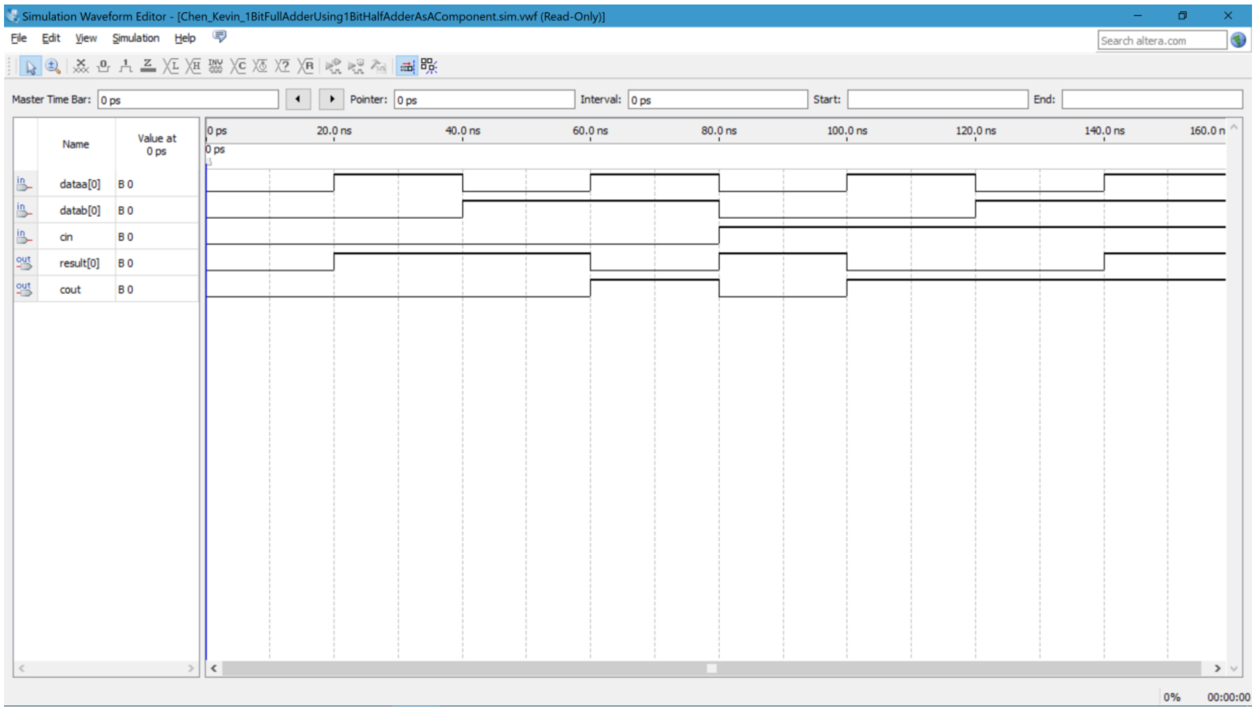


Figure 13: Waveform Simulation of LPM's VHDL Code for 1-bit Full Adder

(Chen_Kevin_FullAdderLPMWaveform.VWF)

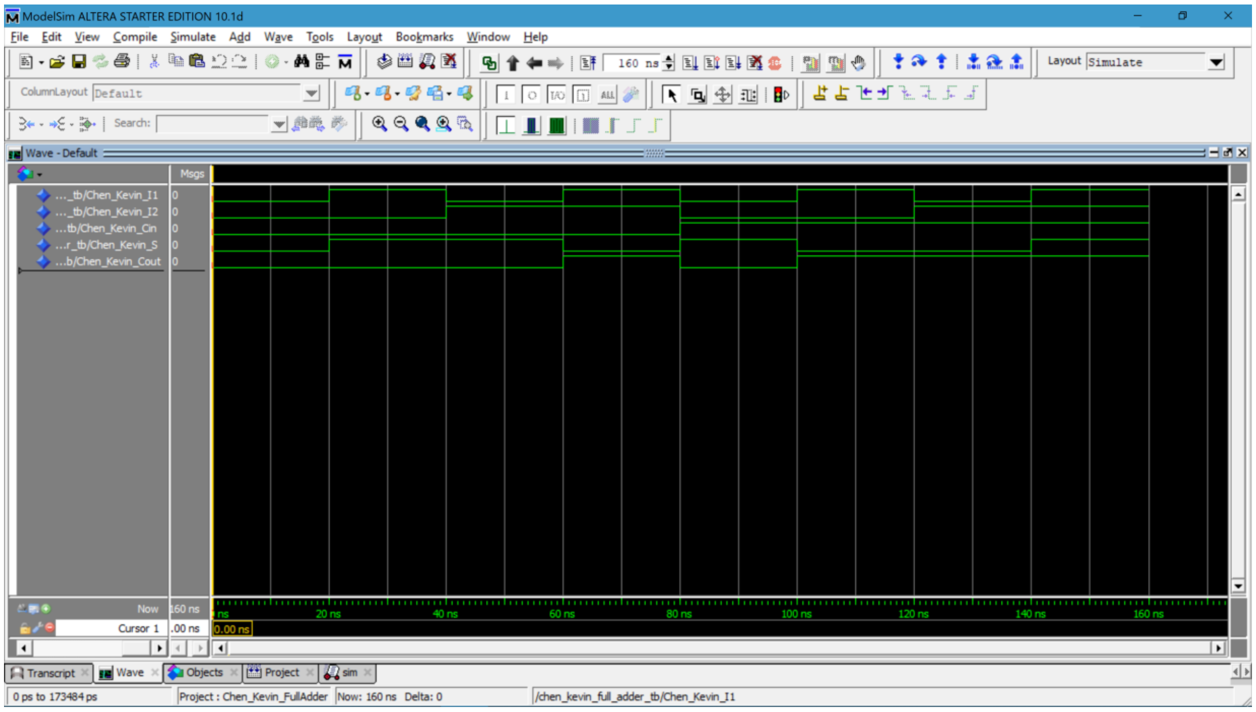


Figure 14: ModelSim of VHDL Code for 1-bit Full Adder

<u>3-to-8 Decoder</u>

The functionality/behavior of the 3-to-8 decoder is to follow the truth table (Table 7), which describes exactly what the output is supposed to be given a certain input.

A waveform was created using the block diagram/schematic file for 3-to-8 decoder (Figure 15) and it was given inputs to test all possible combinations. The outputs that it generated matches those from table 7, therefore, confirming the correctness of the block diagram/schematic file.

A waveform was created using the VHDL code provided by LPM for 3-to-8 decoder (Figure 16) and it was given inputs to match those from figure 15. The outputs that it generated matches figure 15, therefore, confirming the correctness of the VHDL code provided by LPM.

A ModelSim waveform was created using the testbench written in VHDL for 3-to-8 decoder (Figure 17). The testbench written provided the inputs with the same values at the same time as the block diagram/schematic file. The wave generated matches the same output behavior obtained from figure 15 and figure 16, therefore, confirming the correctness of the VHDL code.

.

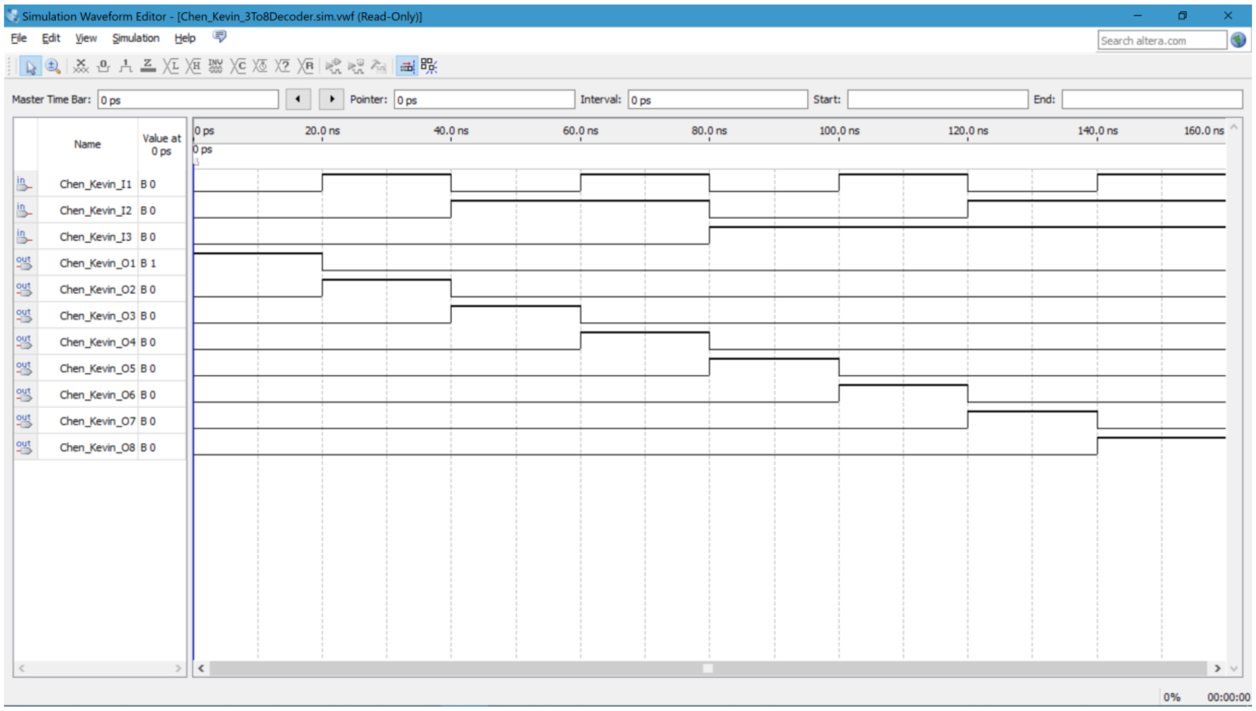
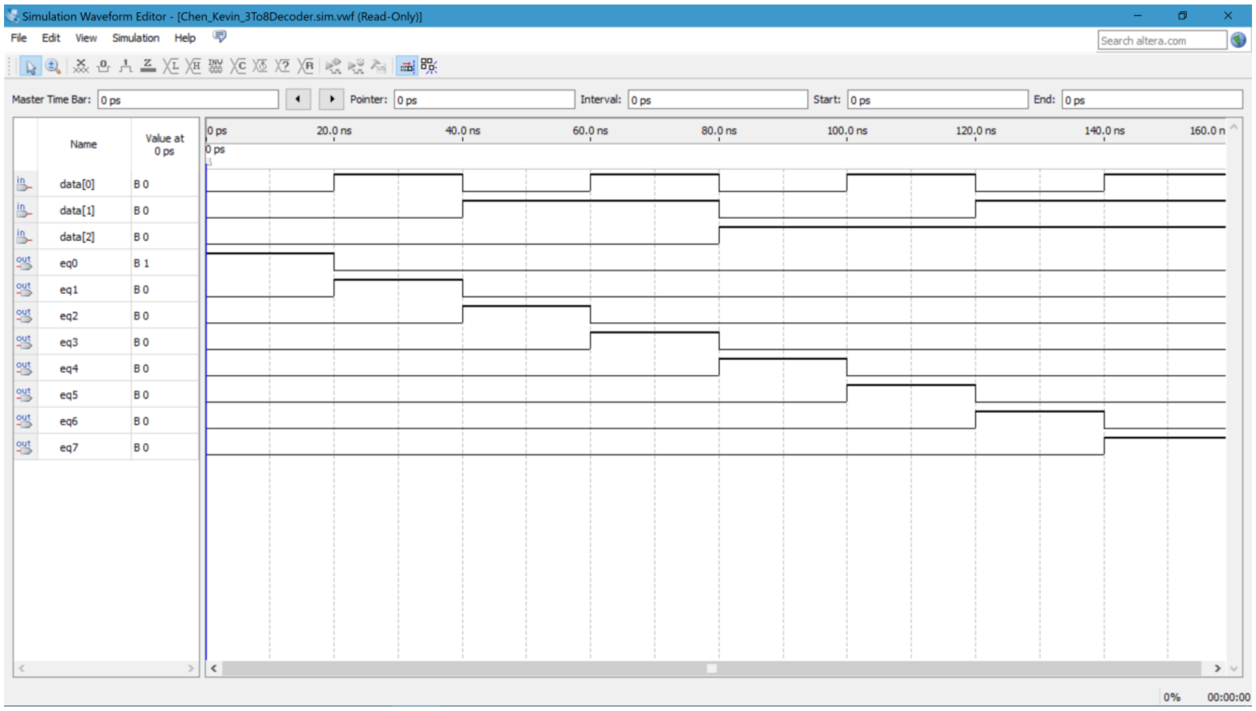


Figure 15: Waveform Simulation of Block Diagram/Schematic File of 3-to-8 Decoder

(Chen_Kevin_3To8DecoderWaveform.VWF)



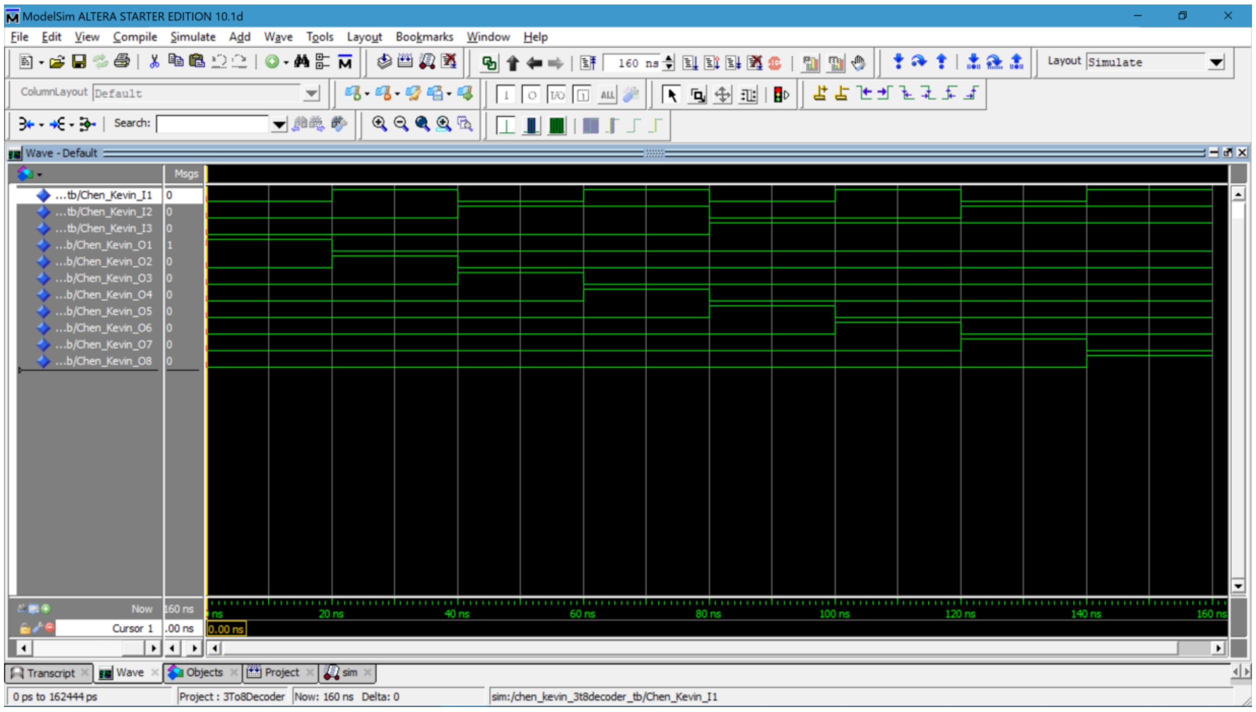
Simulation Waveform Editor - [Chen_Kevin_3To8Decoder.sim.vwf (Read-Only)]

Master Time Bar: 0 ps Pointer: 0 ps Interval: 0 ps Start: 0 ps End: 0 ps

Name	Value at 0 ps
data[0]	B 0
data[1]	B 0
data[2]	B 0
eq0	B 1
eq1	B 0
eq2	B 0
eq3	B 0
eq4	B 0
eq5	B 0
eq6	B 0
eq7	B 0

Figure 16: Waveform Simulation of LPM's VHDL Code for 3-to-8 Decoder

(Chen_Kevin_DecoderLPMWave.VWF)



ModelSim ALTERA STARTER EDITION 10.1d

	Msgs
...tb/Chen_Kevin_I1	0
...tb/Chen_Kevin_I2	0
...tb/Chen_Kevin_I3	0
...b/Chen_Kevin_O1	1
...b/Chen_Kevin_O2	0
...b/Chen_Kevin_O3	0
...b/Chen_Kevin_O4	0
...b/Chen_Kevin_O5	0
...b/Chen_Kevin_O6	0
...b/Chen_Kevin_O7	0
...b/Chen_Kevin_O8	0

Now 160 ns
Cursor 1 .00 ns 0.00 ns

0 ps to 162444 ps | Project : 3To8Decoder | Now: 160 ns Delta: 0 | sim:/chen_kevin_3t8decoder_tb/Chen_Kevin_I1

Figure 17: ModelSim of VHDL Code for 3-to-8 Decoder

<u>8-to-3 Encoder</u>

The functionality/behavior of the 8-to-3 encoder is to follow the truth table (Table 9), which describes exactly what the output is supposed to be given a certain input.

A waveform was created using the block diagram/schematic file for 8-to-3 encoder (Figure 18) and it was given inputs of the combinations provided on the truth table. The outputs that it generated matches those from table 9, therefore, confirming the correctness of the block diagram/schematic file.

A ModelSim waveform was created using the testbench written in VHDL for 8-to-3 encoder (Figure 19). The testbench written provided the inputs with the same values at the same time as the block diagram/schematic file. The wave generated matches the same output behavior obtained from figure 18, therefore, confirming the correctness of the VHDL code.

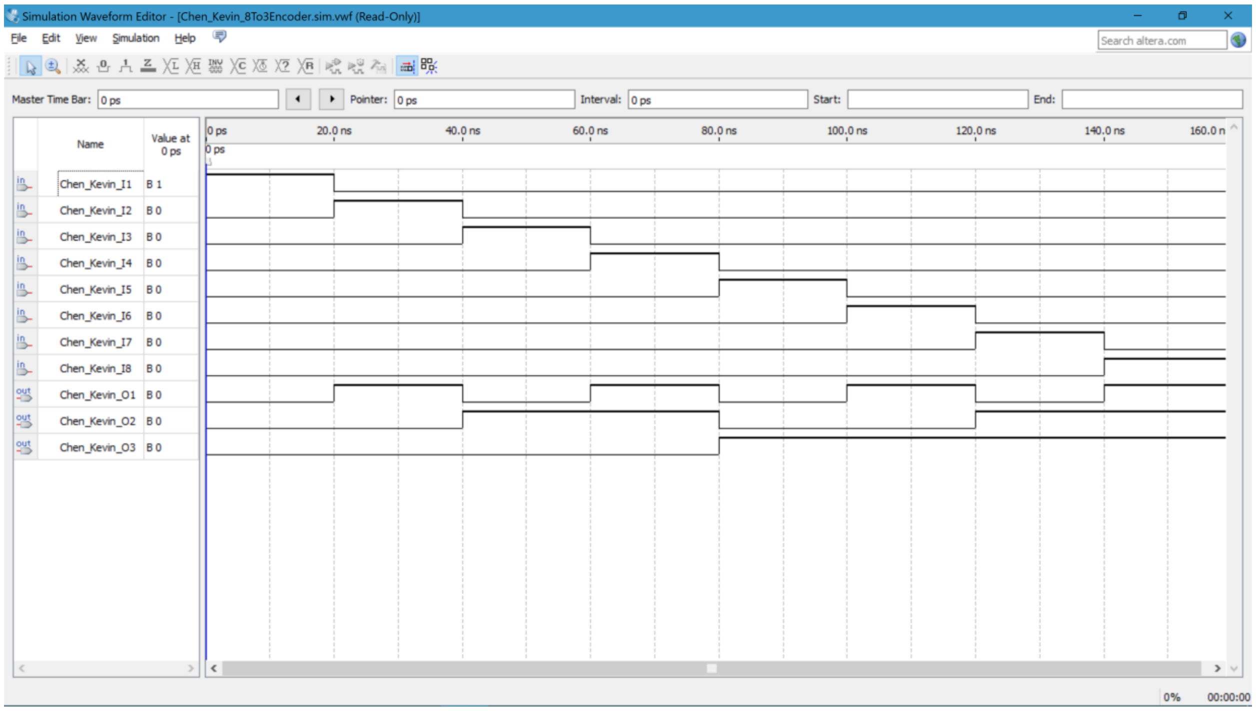


Figure 18: Waveform Simulation of Block Diagram/Schematic File of 8-to-3 Encoder

(Chen_Kevin_3To8DecoderWaveform.VWF)

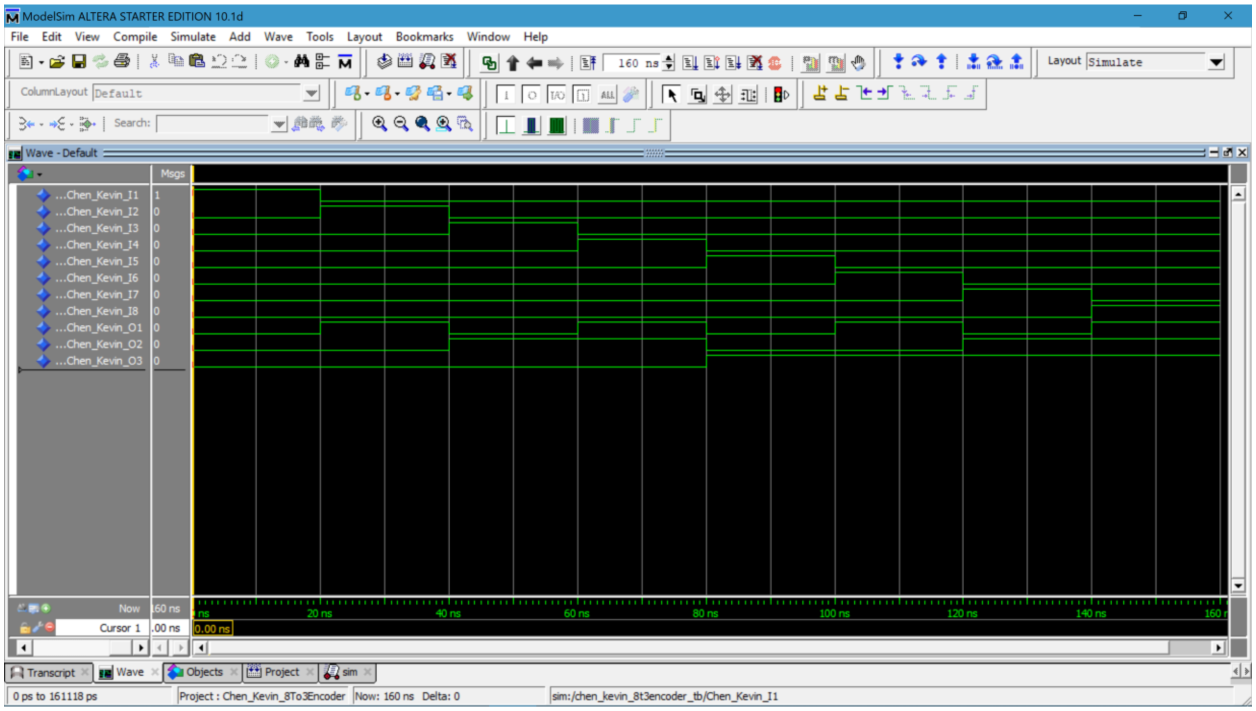


Figure 19: ModelSim of VHDL Code for 8-to-3 Encoder

Conclusion:

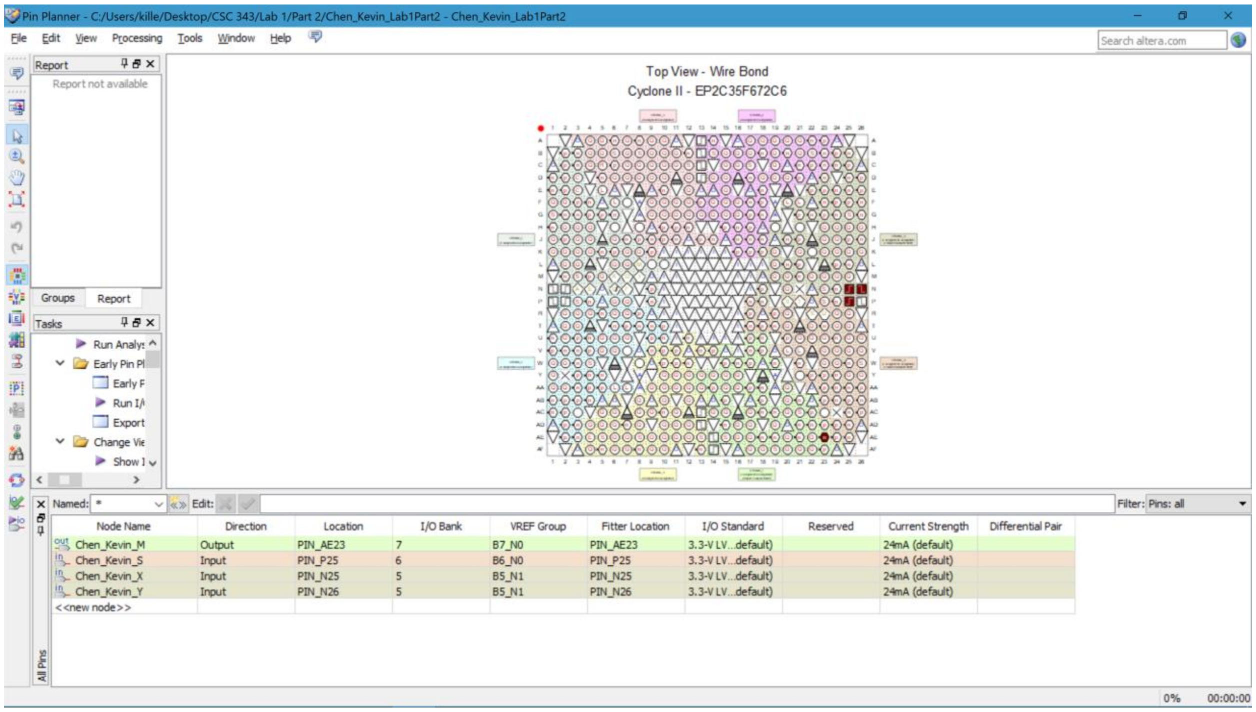
This lab taught me how to create block diagram/schematic files and using it to draw out circuits. With the circuits drawn, I was able to run simulations and select certain inputs to have a certain value and have it produced an output. I was also able to check if the circuit I've drawn was logically correct by comparing the output from the simulation with the values on the truth table. If the outputs were correct, then I was able to verify the circuit's correctness.

This lab also taught me how to use LPM to form a VHDL file of the circuit. With the code already given, all I had to do was run the simulation and select certain inputs to have a certain value and have it produced an output. I was able to check if the code was logically correct by comparing the output from the simulation with the output of the simulation from the block diagram, in addition to comparing it with the truth table. If the simulations match with each other, then I was able to verify the code's correctness.

Another thing this lab taught me was how to create VHDL files and using it to code a circuit. In addition, it taught me how to create a testbench to run the VHDL file. The testbench I made contains the values of the inputs and the time each input changes; I ran it using ModelSim and had it produce a wave. I compared the wave that it produced with the waveform from the block diagram and the truth table. If the wave were the same, then I was able to verify the VHDL code's correctness.

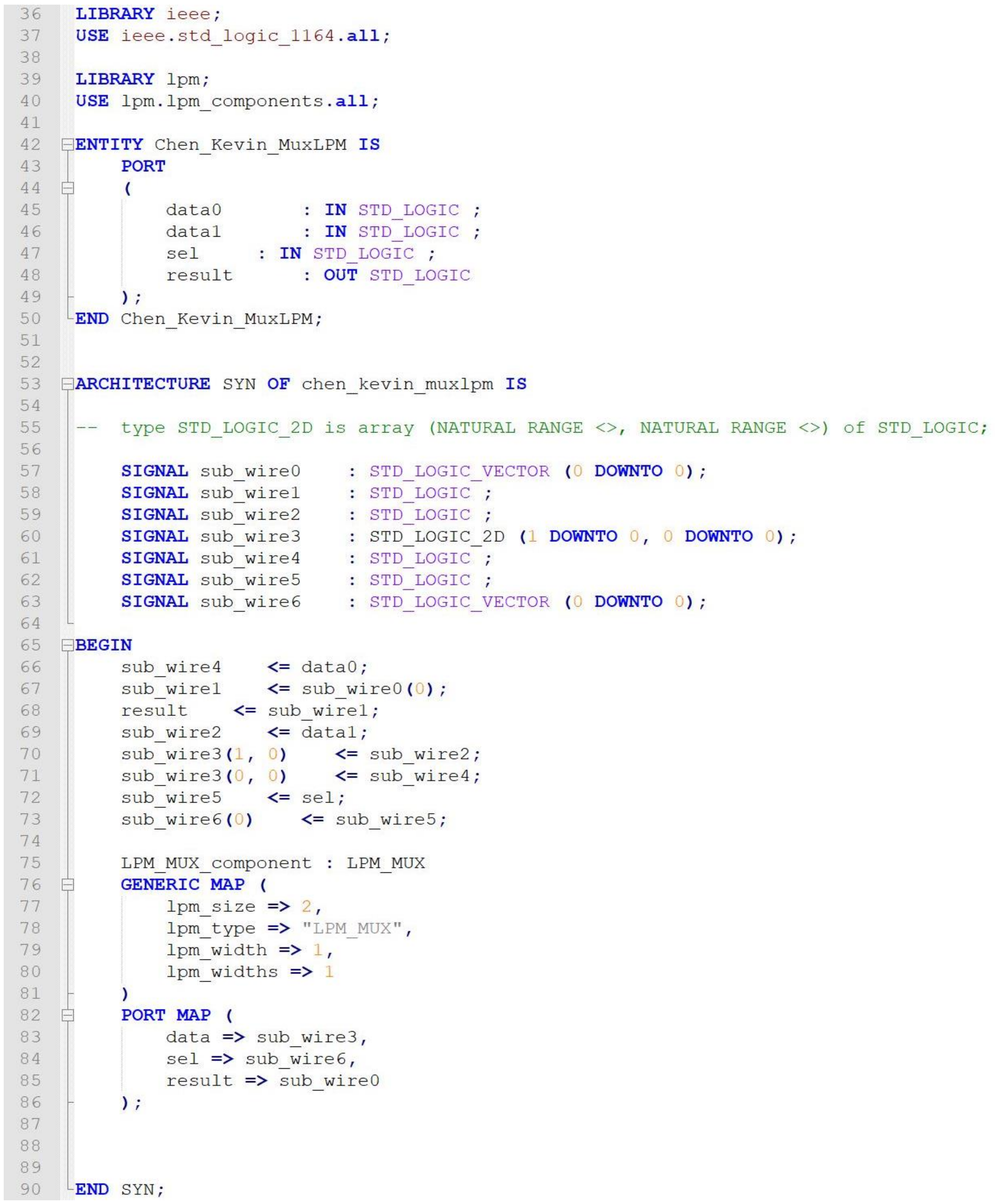
<u>**Appendix:**</u>

<u>2-to-1 Multiplexer</u>



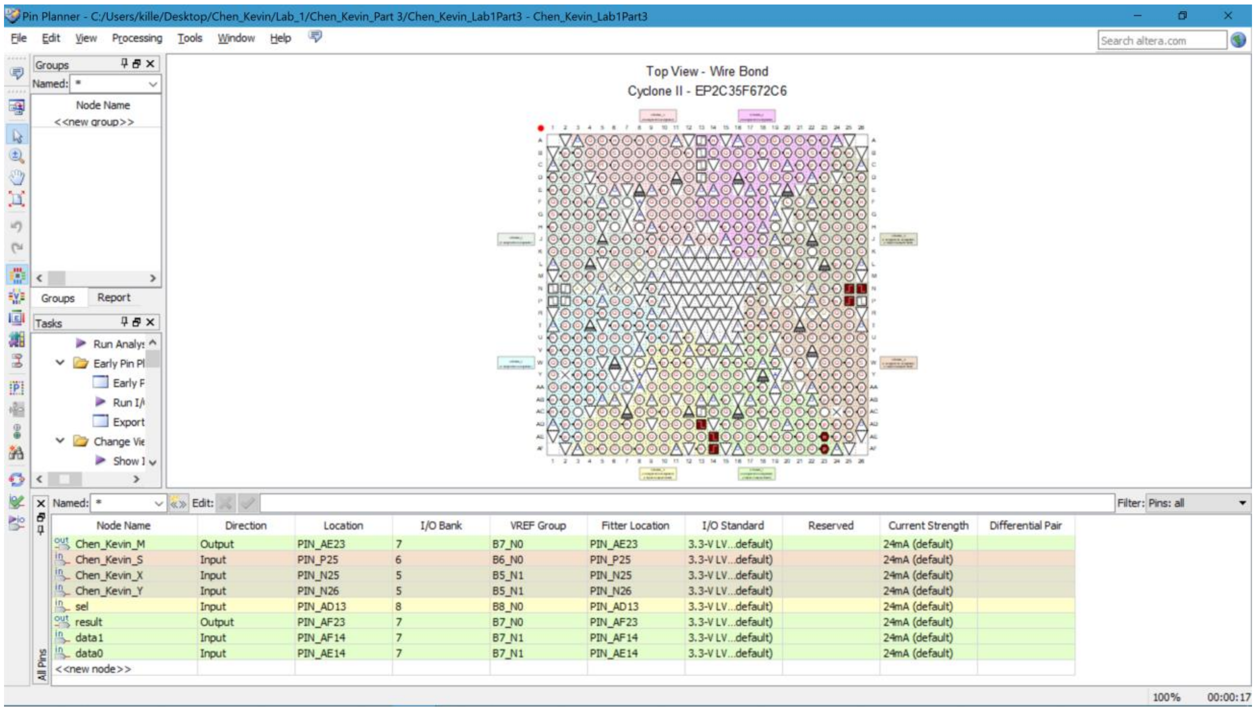
Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
Chen_Kevin_M	Output	PIN_AE23	7	B7_N0	PIN_AE23	3.3-V LV...default)		24mA (default)	
Chen_Kevin_S	Input	PIN_P25	6	B6_N0	PIN_P25	3.3-V LV...default)		24mA (default)	
Chen_Kevin_X	Input	PIN_N25	5	B5_N1	PIN_N25	3.3-V LV...default)		24mA (default)	
Chen_Kevin_Y	Input	PIN_N26	5	B5_N1	PIN_N26	3.3-V LV...default)		24mA (default)	
<<new node>>									

Figure 20: Pin Planner of Block Diagram/Schematic File of 2-to-1 Multiplexer



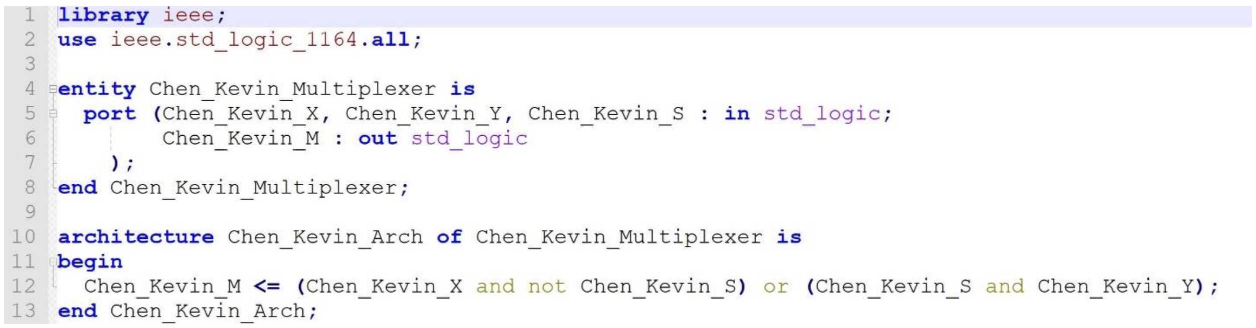
```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.all;

LIBRARY lpm;
USE lpm.lpm_components.all;

ENTITY Chen_Kevin_MuxLPM IS
    PORT
    (
        data0       : IN STD_LOGIC ;
        data1       : IN STD_LOGIC ;
        sel     : IN STD_LOGIC ;
        result      : OUT STD_LOGIC
    );
END Chen_Kevin_MuxLPM;


ARCHITECTURE SYN OF chen_kevin_muxlpm IS

--  type STD_LOGIC_2D is array (NATURAL RANGE <>, NATURAL RANGE <>) of STD_LOGIC;

    SIGNAL sub_wire0    : STD_LOGIC_VECTOR (0 DOWNTO 0);
    SIGNAL sub_wire1    : STD_LOGIC ;
    SIGNAL sub_wire2    : STD_LOGIC ;
    SIGNAL sub_wire3    : STD_LOGIC_2D (1 DOWNTO 0, 0 DOWNTO 0);
    SIGNAL sub_wire4    : STD_LOGIC ;
    SIGNAL sub_wire5    : STD_LOGIC ;
    SIGNAL sub_wire6    : STD_LOGIC_VECTOR (0 DOWNTO 0);

BEGIN
    sub_wire4    <= data0;
    sub_wire1    <= sub_wire0(0);
    result    <= sub_wire1;
    sub_wire2    <= data1;
    sub_wire3(1, 0)    <= sub_wire2;
    sub_wire3(0, 0)    <= sub_wire4;
    sub_wire5    <= sel;
    sub_wire6(0)    <= sub_wire5;

    LPM_MUX_component : LPM_MUX
    GENERIC MAP (
        lpm_size => 2,
        lpm_type => "LPM_MUX",
        lpm_width => 1,
        lpm_widths => 1
    )
    PORT MAP (
        data => sub_wire3,
        sel => sub_wire6,
        result => sub_wire0
    );



END SYN;
```

Figure 21: LPM's VHDL Code for 2-to-1 Multiplexer (Chen_Kevin_MuxLPM.VHD)


Pin Planner - C:/Users/kille/Desktop/Chen_Kevin/Lab_1/Chen_Kevin_Part 3/Chen_Kevin_Lab1Part3 - Chen_Kevin_Lab1Part3
File Edit View Processing Tools Window Help
Top View - Wire Bond
Cyclone II - EP2C35F672C6
Groups
Report
Tasks
Filter: Pins: all

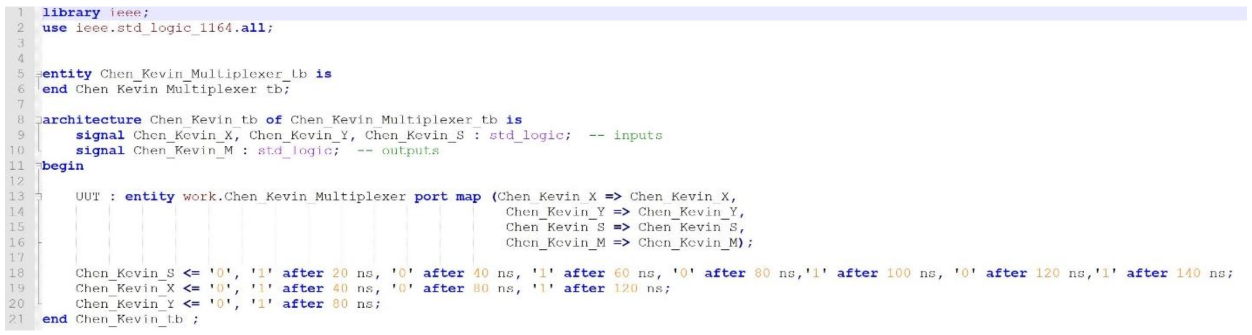
Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
Chen_Kevin_M	Output	PIN_AE23	7	B7_N0	PIN_AE23	3.3-V LV...default)		24mA (default)	
Chen_Kevin_S	Input	PIN_P25	6	B6_N0	PIN_P25	3.3-V LV...default)		24mA (default)	
Chen_Kevin_X	Input	PIN_N25	5	B5_N1	PIN_N25	3.3-V LV...default)		24mA (default)	
Chen_Kevin_Y	Input	PIN_N26	5	B5_N1	PIN_N26	3.3-V LV...default)		24mA (default)	
sel	Input	PIN_AD13	8	B8_N0	PIN_AD13	3.3-V LV...default)		24mA (default)	
result	Output	PIN_AF23	7	B7_N0	PIN_AF23	3.3-V LV...default)		24mA (default)	
data1	Input	PIN_AF14	7	B7_N1	PIN_AF14	3.3-V LV...default)		24mA (default)	
data0	Input	PIN_AE14	7	B7_N1	PIN_AE14	3.3-V LV...default)		24mA (default)	
<<new node>>									

Figure 22: Pin Planner of LPM's VHDL Code for 2-to-1 Multiplexer



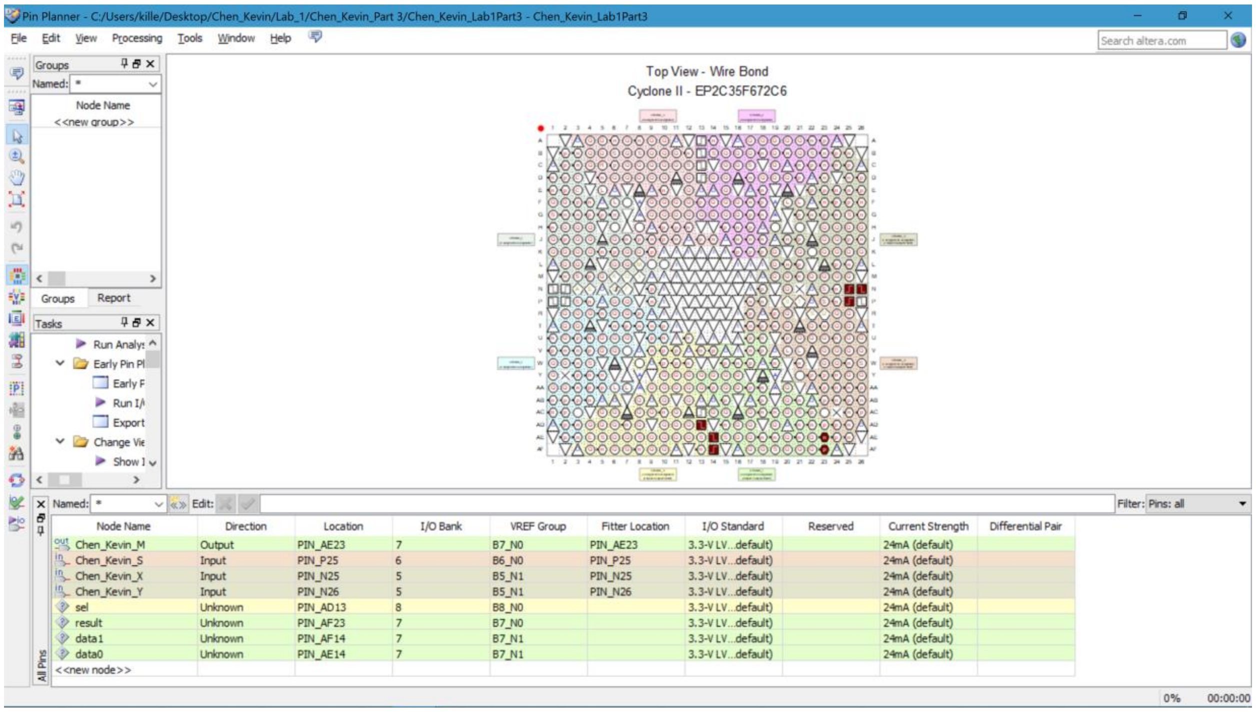
```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity Chen_Kevin_Multiplexer is
  port (Chen_Kevin_X, Chen_Kevin_Y, Chen_Kevin_S : in std_logic;
        Chen_Kevin_M : out std_logic
    );
end Chen_Kevin_Multiplexer;

architecture Chen_Kevin_Arch of Chen_Kevin_Multiplexer is
begin
  Chen_Kevin_M <= (Chen_Kevin_X and not Chen_Kevin_S) or (Chen_Kevin_S and Chen_Kevin_Y);
end Chen_Kevin_Arch;
```

Figure 23: My VHDL Code for 2-to-1 Multiplexer (Chen_Kevin_Multiplexer.VHD)



```vhdl
library ieee;
use ieee.std_logic_1164.all;


entity Chen_Kevin_Multiplexer_tb is
end Chen Kevin Multiplexer tb;

architecture Chen_Kevin_tb of Chen_Kevin_Multiplexer_tb is
    signal Chen_Kevin_X, Chen_Kevin_Y, Chen_Kevin_S : std_logic;  -- inputs
    signal Chen_Kevin_M : std_logic;  -- outputs
begin

    UUT : entity work.Chen_Kevin_Multiplexer port map (Chen_Kevin_X => Chen_Kevin_X,
                                                       Chen_Kevin_Y => Chen_Kevin_Y,
                                                       Chen_Kevin_S => Chen_Kevin_S,
                                                       Chen_Kevin_M => Chen_Kevin_M);

    Chen_Kevin_S <= '0', '1' after 20 ns, '0' after 40 ns, '1' after 60 ns, '0' after 80 ns,'1' after 100 ns, '0' after 120 ns,'1' after 140 ns;
    Chen_Kevin_X <= '0', '1' after 40 ns, '0' after 80 ns, '1' after 120 ns;
    Chen_Kevin_Y <= '0', '1' after 80 ns;
end Chen_Kevin_tb ;
```

Figure 24: My VHDL Testbench Code for 2-to-1 Multiplexer (Chen_Kevin_Multiplexer_tb.VHD)



Pin Planner - C:/Users/kille/Desktop/Chen_Kevin/Lab_1/Chen_Kevin_Part 3/Chen_Kevin_Lab1Part3 - Chen_Kevin_Lab1Part3

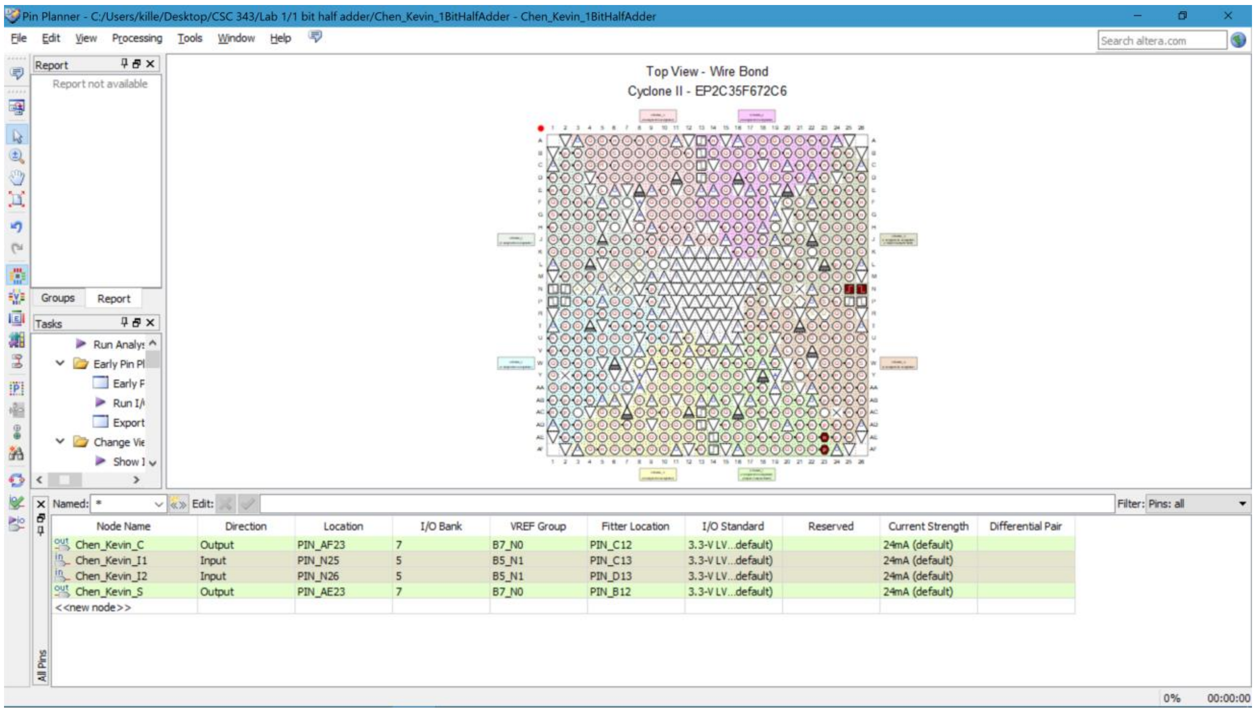
Top View - Wire Bond
Cyclone II - EP2C35F672C6

Named: * Edit: Filter: Pins: all

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
Chen_Kevin_M	Output	PIN_AE23	7	B7_N0	PIN_AE23	3.3-V LV...default)		24mA (default)	
Chen_Kevin_S	Input	PIN_P25	6	B6_N0	PIN_P25	3.3-V LV...default)		24mA (default)	
Chen_Kevin_X	Input	PIN_N25	5	B5_N1	PIN_N25	3.3-V LV...default)		24mA (default)	
Chen_Kevin_Y	Input	PIN_N26	5	B5_N1	PIN_N26	3.3-V LV...default)		24mA (default)	
sel	Unknown	PIN_AD13	8	B8_N0		3.3-V LV...default)		24mA (default)	
result	Unknown	PIN_AF23	7	B7_N0		3.3-V LV...default)		24mA (default)	
data1	Unknown	PIN_AF14	7	B7_N1		3.3-V LV...default)		24mA (default)	
data0	Unknown	PIN_AE14	7	B7_N1		3.3-V LV...default)		24mA (default)	
<<new node>>									

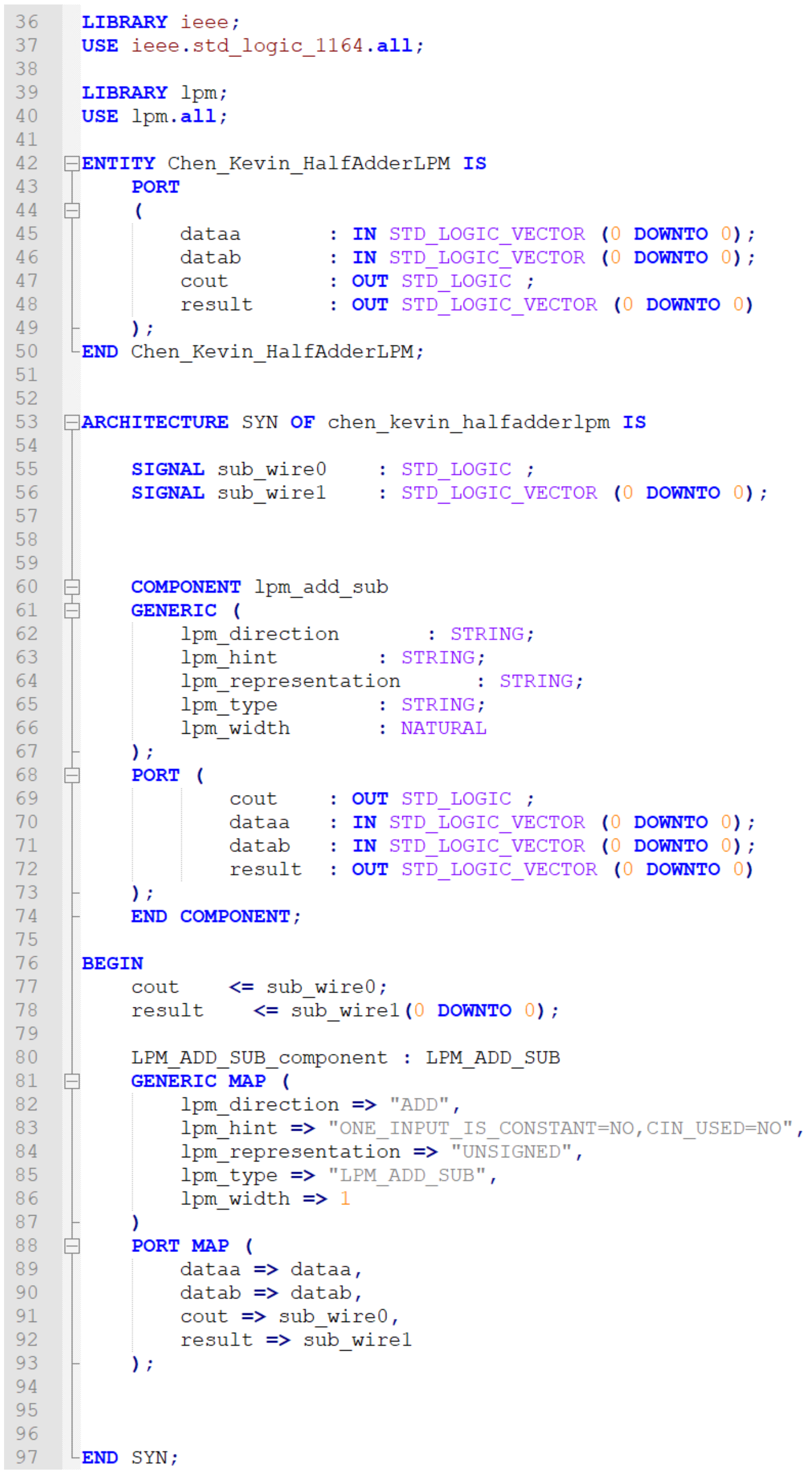
Figure 25: Pin Planner of my VHDL Code for 2-to-1 Multiplexer

1-bit Half Adder


Pin Planner - C:/Users/kille/Desktop/CSC 343/Lab 1/1 bit half adder/Chen_Kevin_1BitHalfAdder - Chen_Kevin_1BitHalfAdder
File Edit View Processing Tools Window Help
Report
Report not available
Groups Report
Tasks
Run Analys
Early Pin Pl
Early F
Run I/
Export
Change Vie
Show I
Top View - Wire Bond
Cyclone II - EP2C35F672C6
Named: *
Edit:
Filter: Pins: all
All Pins
0% 00:00:00

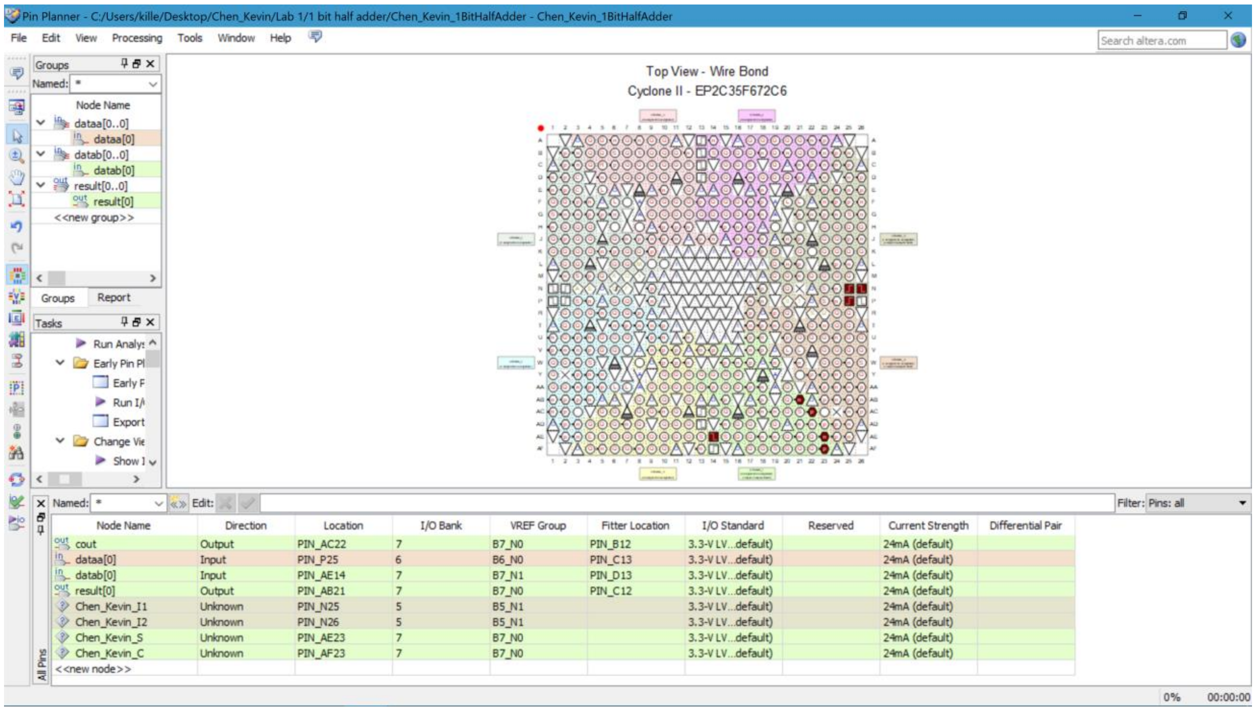
Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
Chen_Kevin_C	Output	PIN_AF23	7	B7_N0	PIN_C12	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I1	Input	PIN_N25	5	B5_N1	PIN_C13	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I2	Input	PIN_N26	5	B5_N1	PIN_D13	3.3-V LV...default)		24mA (default)	
Chen_Kevin_S	Output	PIN_AE23	7	B7_N0	PIN_B12	3.3-V LV...default)		24mA (default)	
<<new node>>									

Figure 26: Pin Planner of Block Diagram/Schematic File of 1-bit Half Adder



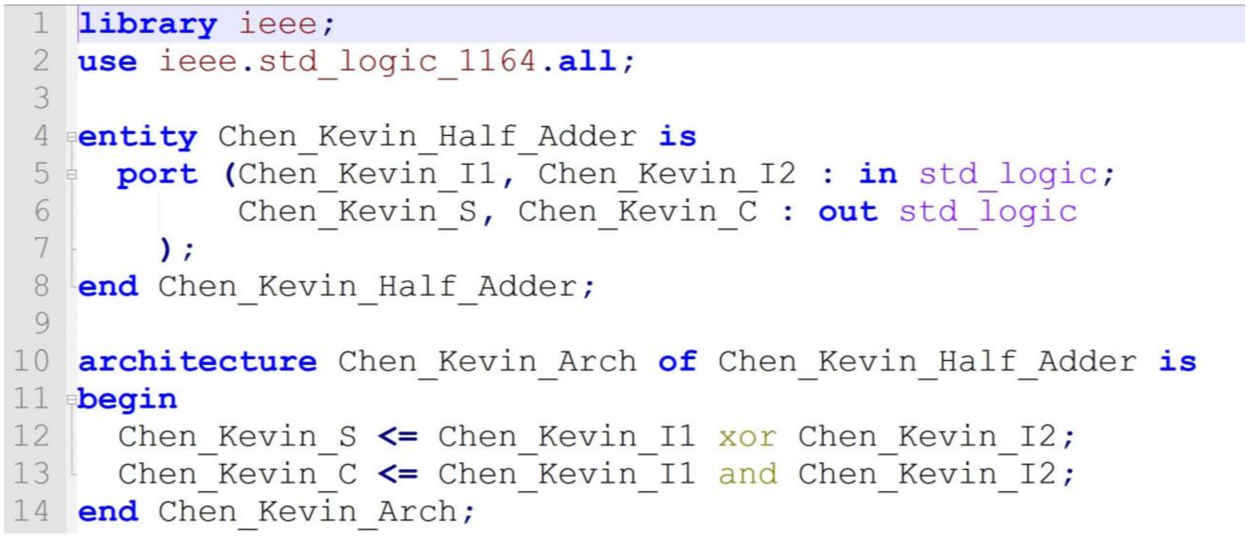
```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.all;

LIBRARY lpm;
USE lpm.all;

ENTITY Chen_Kevin_HalfAdderLPM IS
    PORT
    (
        dataa        : IN STD_LOGIC_VECTOR (0 DOWNTO 0);
        datab        : IN STD_LOGIC_VECTOR (0 DOWNTO 0);
        cout         : OUT STD_LOGIC ;
        result       : OUT STD_LOGIC_VECTOR (0 DOWNTO 0)
    );
END Chen_Kevin_HalfAdderLPM;


ARCHITECTURE SYN OF chen_kevin_halfadderlpm IS

    SIGNAL sub_wire0    : STD_LOGIC ;
    SIGNAL sub_wire1    : STD_LOGIC_VECTOR (0 DOWNTO 0);



    COMPONENT lpm_add_sub
    GENERIC (
        lpm_direction       : STRING;
        lpm_hint        : STRING;
        lpm_representation      : STRING;
        lpm_type        : STRING;
        lpm_width       : NATURAL
    );
    PORT (
            cout    : OUT STD_LOGIC ;
            dataa   : IN STD_LOGIC_VECTOR (0 DOWNTO 0);
            datab   : IN STD_LOGIC_VECTOR (0 DOWNTO 0);
            result  : OUT STD_LOGIC_VECTOR (0 DOWNTO 0)
    );
    END COMPONENT;

BEGIN
    cout    <= sub_wire0;
    result    <= sub_wire1(0 DOWNTO 0);

    LPM_ADD_SUB_component : LPM_ADD_SUB
    GENERIC MAP (
        lpm_direction => "ADD",
        lpm_hint => "ONE_INPUT_IS_CONSTANT=NO,CIN_USED=NO",
        lpm_representation => "UNSIGNED",
        lpm_type => "LPM_ADD_SUB",
        lpm_width => 1
    )
    PORT MAP (
        dataa => dataa,
        datab => datab,
        cout => sub_wire0,
        result => sub_wire1
    );



END SYN;
```

Figure 27: LPM's VHDL Code for 1-bit Half Adder (Chen_Kevin_HalfAdderLPM.VHD)



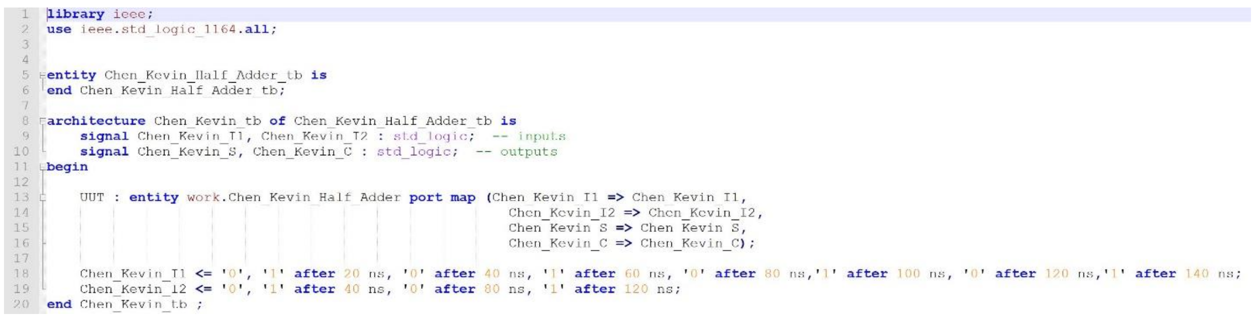
Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
cout	Output	PIN_AC22	7	B7_N0	PIN_B12	3.3-V LV...default)		24mA (default)	
dataa[0]	Input	PIN_P25	6	B6_N0	PIN_C13	3.3-V LV...default)		24mA (default)	
datab[0]	Input	PIN_AE14	7	B7_N1	PIN_D13	3.3-V LV...default)		24mA (default)	
result[0]	Output	PIN_AB21	7	B7_N0	PIN_C12	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I1	Unknown	PIN_N25	5	B5_N1		3.3-V LV...default)		24mA (default)	
Chen_Kevin_I2	Unknown	PIN_N26	5	B5_N1		3.3-V LV...default)		24mA (default)	
Chen_Kevin_S	Unknown	PIN_AE23	7	B7_N0		3.3-V LV...default)		24mA (default)	
Chen_Kevin_C	Unknown	PIN_AF23	7	B7_N0		3.3-V LV...default)		24mA (default)	
<<new node>>									

Figure 28: Pin Planner of LPM's VHDL Code for 1-bit Half Adder



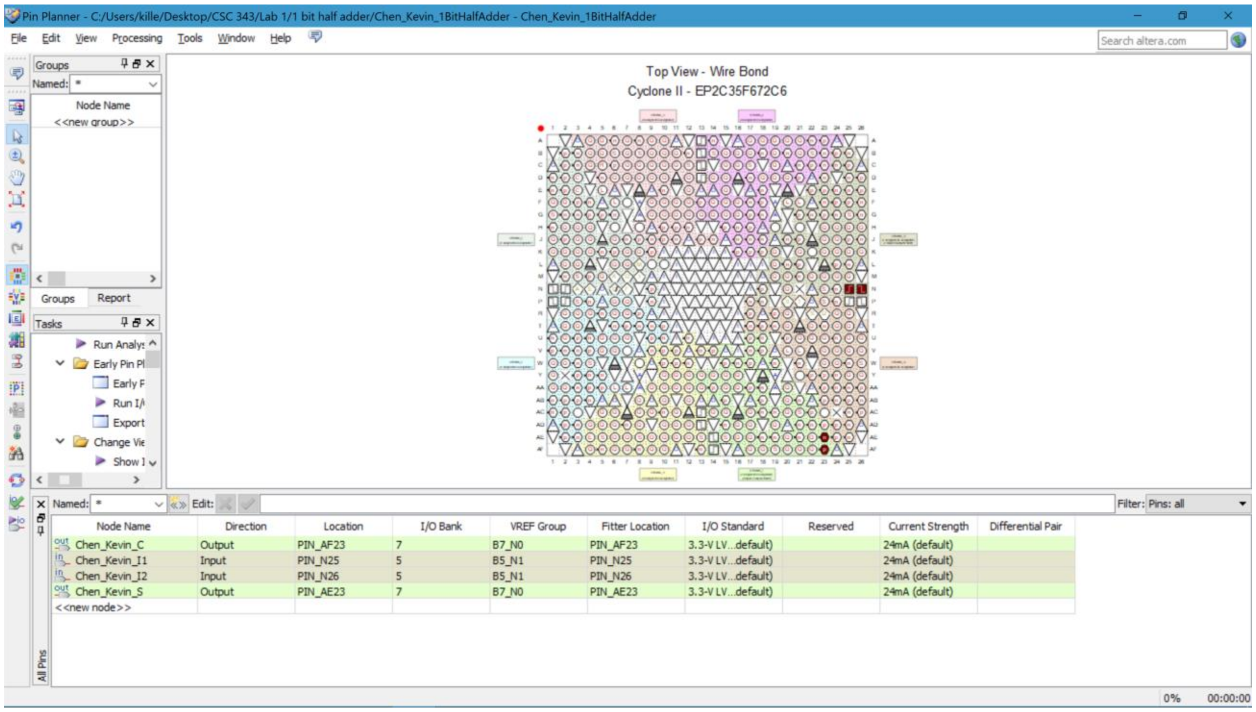
```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity Chen_Kevin_Half_Adder is
  port (Chen_Kevin_I1, Chen_Kevin_I2 : in std_logic;
        Chen_Kevin_S, Chen_Kevin_C : out std_logic
    );
end Chen_Kevin_Half_Adder;

architecture Chen_Kevin_Arch of Chen_Kevin_Half_Adder is
begin
  Chen_Kevin_S <= Chen_Kevin_I1 xor Chen_Kevin_I2;
  Chen_Kevin_C <= Chen_Kevin_I1 and Chen_Kevin_I2;
end Chen_Kevin_Arch;
```

Figure 29: My VHDL Code for 1-bit Half Adder (Chen_Kevin_Half_Adder.VHD)



```
library ieee;
use ieee.std_logic_1164.all;


entity Chen_Kevin_Half_Adder_tb is
end Chen_Kevin_Half_Adder_tb;

architecture Chen_Kevin_tb of Chen_Kevin_Half_Adder_tb is
    signal Chen_Kevin_I1, Chen_Kevin_I2 : std_logic;  -- inputs
    signal Chen_Kevin_S, Chen_Kevin_C : std_logic;  -- outputs
begin

    UUT : entity work.Chen_Kevin_Half_Adder port map (Chen_Kevin_I1 => Chen_Kevin_I1,
                                                       Chen_Kevin_I2 => Chen_Kevin_I2,
                                                       Chen_Kevin_S => Chen_Kevin_S,
                                                       Chen_Kevin_C => Chen_Kevin_C);

    Chen_Kevin_I1 <= '0', '1' after 20 ns, '0' after 40 ns, '1' after 60 ns, '0' after 80 ns,'1' after 100 ns, '0' after 120 ns,'1' after 140 ns;
    Chen_Kevin_I2 <= '0', '1' after 40 ns, '0' after 80 ns, '1' after 120 ns;
end Chen_Kevin_tb ;
```

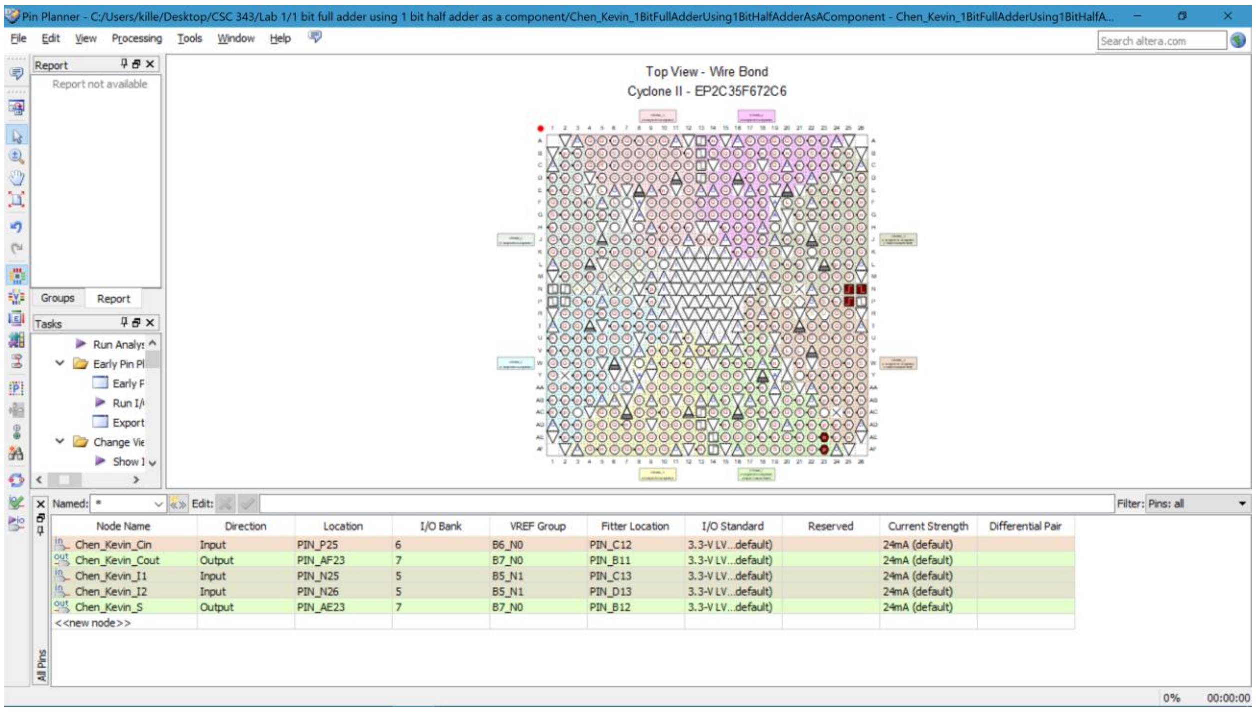
Figure 30: My VHDL Testbench Code for 1-bit Half Adder (Chen_Kevin_Half_Adder_tb.VHD)



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
Chen_Kevin_C	Output	PIN_AF23	7	B7_N0	PIN_AF23	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I1	Input	PIN_N25	5	B5_N1	PIN_N25	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I2	Input	PIN_N26	5	B5_N1	PIN_N26	3.3-V LV...default)		24mA (default)	
Chen_Kevin_S	Output	PIN_AE23	7	B7_N0	PIN_AE23	3.3-V LV...default)		24mA (default)	
<<new node>>									

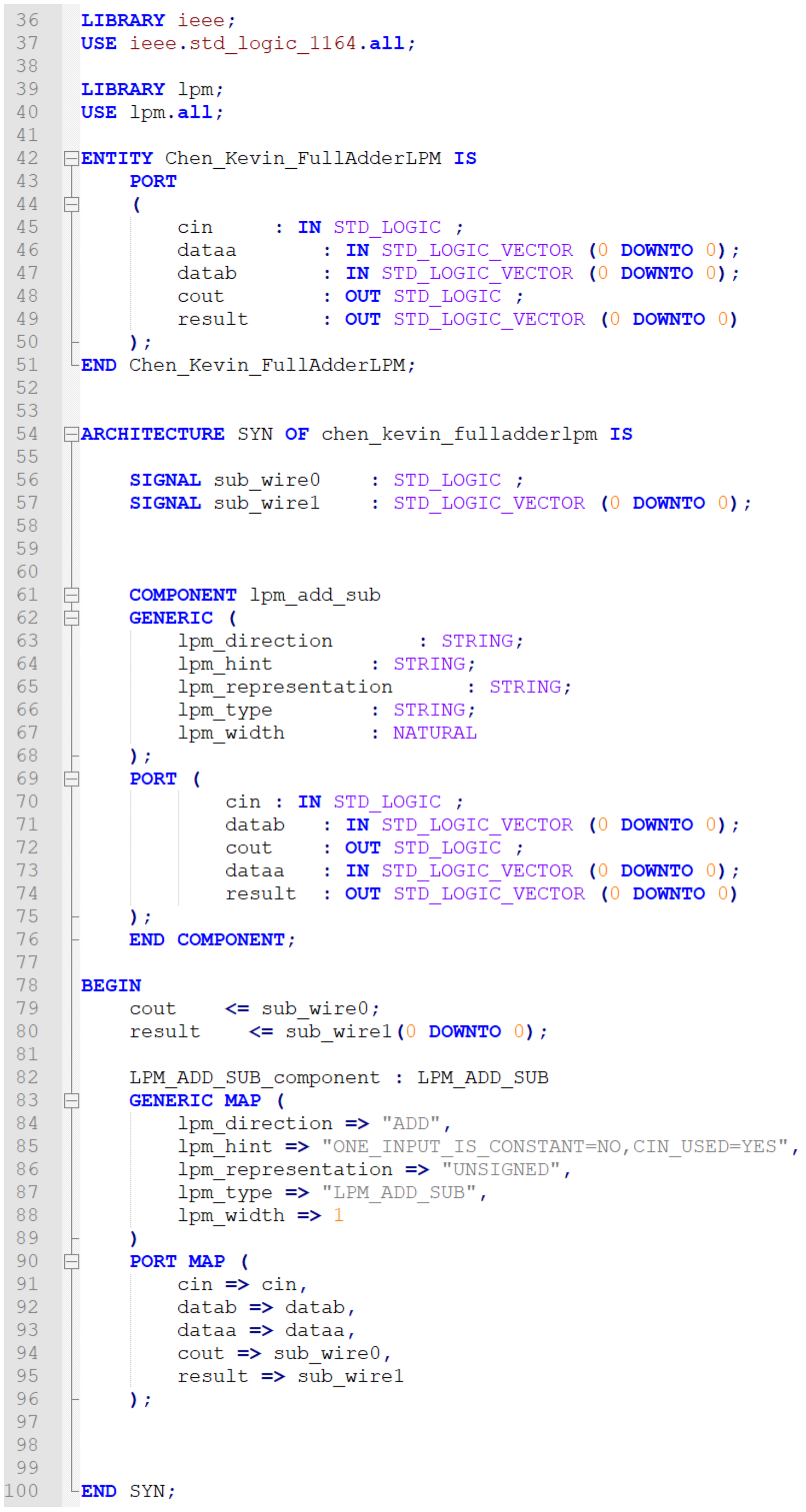
Figure 31: Pin Planner of my VHDL Code for 1-bit Half Adder

<u>1-bit Full Adder Using 1-bit Half Adder as a Component</u>


Pin Planner - C:/Users/kille/Desktop/CSC 343/Lab 1/1 bit full adder using 1 bit half adder as a component/Chen_Kevin_1BitFullAdderUsing1BitHalfAdderAsAComponent - Chen_Kevin_1BitFullAdderUsing1BitHalfA...
File Edit View Processing Tools Window Help
Search altera.com
Report
Report not available
Groups Report
Tasks
Run Analys
Early Pin Pl
Early F
Run I/
Export
Change Vie
Show I
Top View - Wire Bond
Cyclone II - EP2C35F672C6
Named: *
Edit:
Filter: Pins: all
All Pins
0%
00:00:00

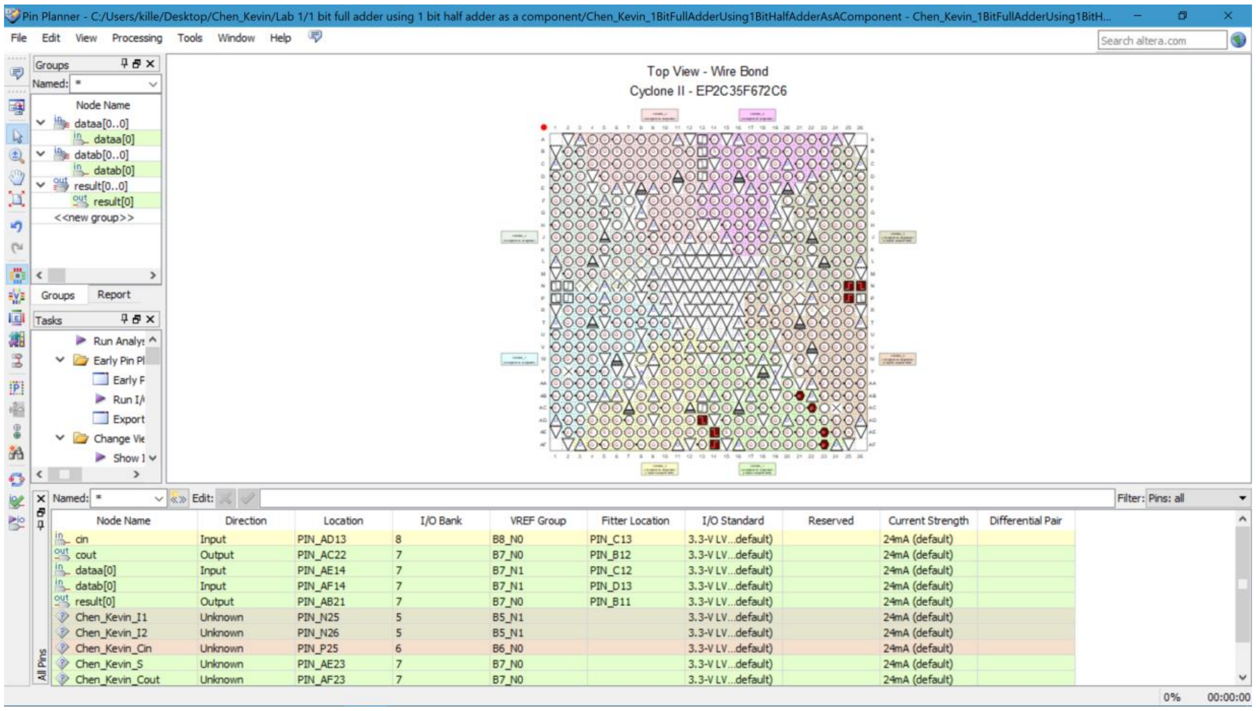
Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
Chen_Kevin_Cin	Input	PIN_P25	6	B6_N0	PIN_C12	3.3-V LV...default)		24mA (default)	
Chen_Kevin_Cout	Output	PIN_AF23	7	B7_N0	PIN_B11	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I1	Input	PIN_N25	5	B5_N1	PIN_C13	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I2	Input	PIN_N26	5	B5_N1	PIN_D13	3.3-V LV...default)		24mA (default)	
Chen_Kevin_S	Output	PIN_AE23	7	B7_N0	PIN_B12	3.3-V LV...default)		24mA (default)	
<<new node>>									

Figure 32: Pin Planner of Block Diagram/Schematic File of 1-bit Full Adder



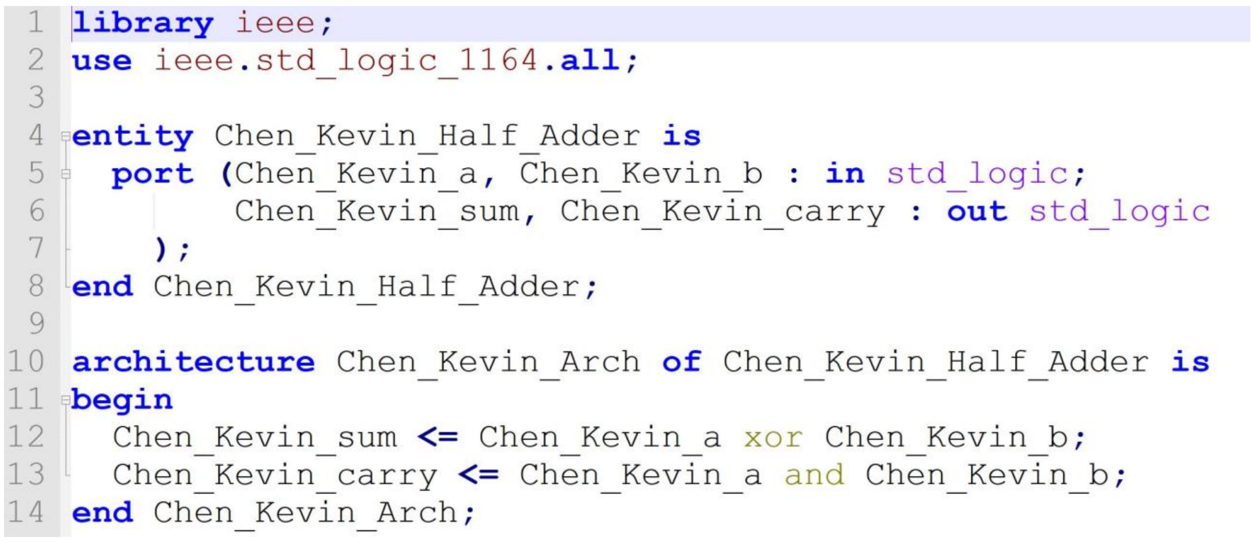
```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.all;

LIBRARY lpm;
USE lpm.all;

ENTITY Chen_Kevin_FullAdderLPM IS
    PORT
    (
        cin     : IN STD_LOGIC ;
        dataa       : IN STD_LOGIC_VECTOR (0 DOWNTO 0);
        datab       : IN STD_LOGIC_VECTOR (0 DOWNTO 0);
        cout        : OUT STD_LOGIC ;
        result      : OUT STD_LOGIC_VECTOR (0 DOWNTO 0)
    );
END Chen_Kevin_FullAdderLPM;


ARCHITECTURE SYN OF chen_kevin_fulladderlpm IS

    SIGNAL sub_wire0    : STD_LOGIC ;
    SIGNAL sub_wire1    : STD_LOGIC_VECTOR (0 DOWNTO 0);



    COMPONENT lpm_add_sub
    GENERIC (
        lpm_direction       : STRING;
        lpm_hint        : STRING;
        lpm_representation      : STRING;
        lpm_type        : STRING;
        lpm_width       : NATURAL
    );
    PORT (
            cin : IN STD_LOGIC ;
            datab   : IN STD_LOGIC_VECTOR (0 DOWNTO 0);
            cout    : OUT STD_LOGIC ;
            dataa   : IN STD_LOGIC_VECTOR (0 DOWNTO 0);
            result  : OUT STD_LOGIC_VECTOR (0 DOWNTO 0)
    );
    END COMPONENT;

BEGIN
    cout    <= sub_wire0;
    result    <= sub_wire1(0 DOWNTO 0);

    LPM_ADD_SUB_component : LPM_ADD_SUB
    GENERIC MAP (
        lpm_direction => "ADD",
        lpm_hint => "ONE_INPUT_IS_CONSTANT=NO,CIN_USED=YES",
        lpm_representation => "UNSIGNED",
        lpm_type => "LPM_ADD_SUB",
        lpm_width => 1
    )
    PORT MAP (
        cin => cin,
        datab => datab,
        dataa => dataa,
        cout => sub_wire0,
        result => sub_wire1
    );



END SYN;
```

Figure 33: LPM's VHDL Code for 1-bit Full Adder (Chen_Kevin_FullAdderLPM.VHD)


Pin Planner - C:/Users/kille/Desktop/Chen_Kevin/Lab 1/1 bit full adder using 1 bit half adder as a component/Chen_Kevin_1BitFullAdderUsing1BitHalfAdderAsAComponent - Chen_Kevin_1BitFullAdderUsing1BitH...
File Edit View Processing Tools Window Help
Groups
Named: *
Node Name
dataa[0..0]
dataa[0]
datab[0..0]
datab[0]
result[0..0]
result[0]
<<new group>>
Groups Report
Tasks
Run Analys
Early Pin Pl
Early F
Run I/
Export
Change Vie
Show I
Top View - Wire Bond
Cyclone II - EP2C35F672C6
Named: *
Edit:
Filter: Pins: all
All Pins
0%
00:00:00

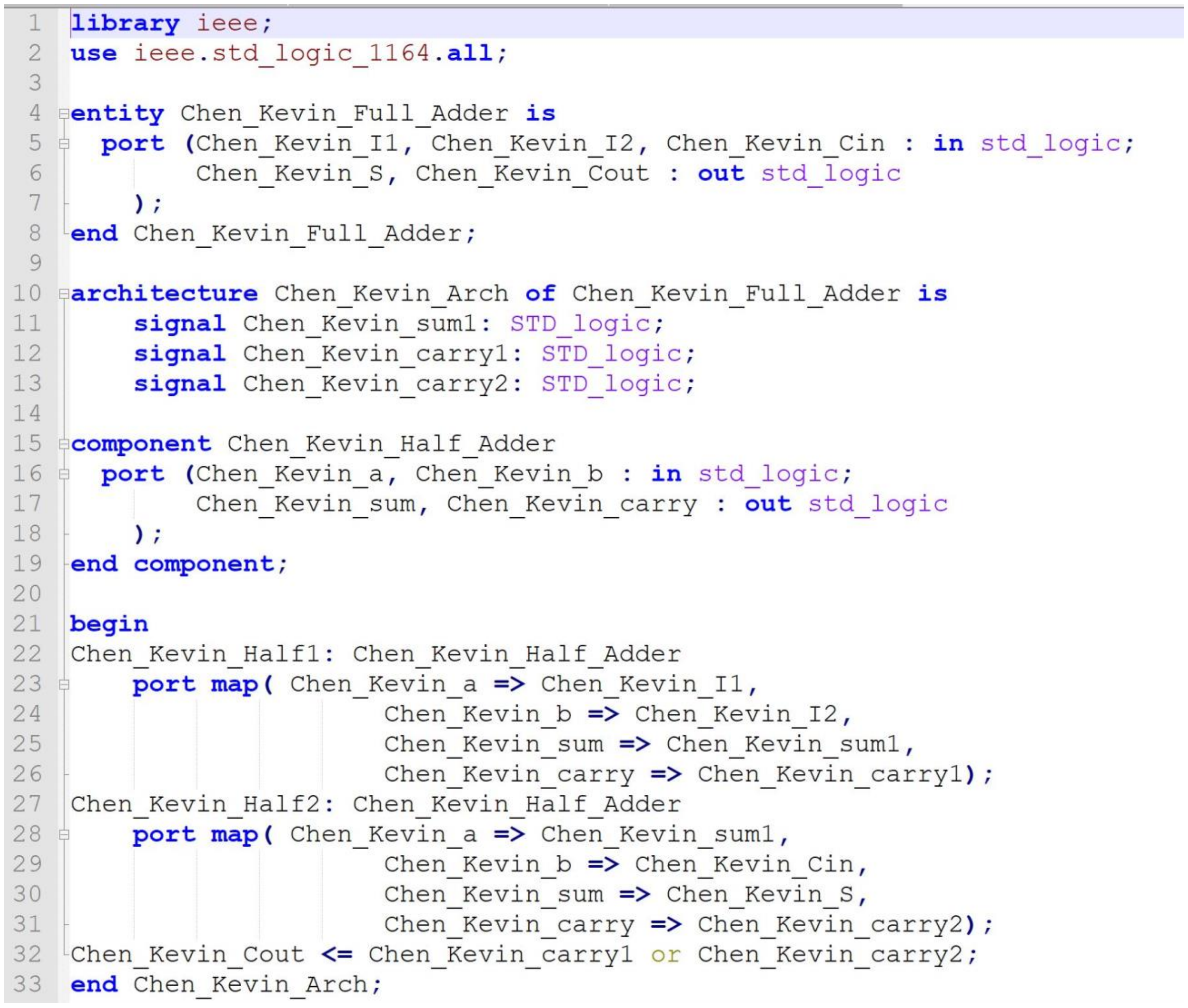
Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
cin	Input	PIN_AD13	8	B8_N0	PIN_C13	3.3-V LV...default)		24mA (default)	
cout	Output	PIN_AC22	7	B7_N0	PIN_B12	3.3-V LV...default)		24mA (default)	
dataa[0]	Input	PIN_AE14	7	B7_N1	PIN_C12	3.3-V LV...default)		24mA (default)	
datab[0]	Input	PIN_AF14	7	B7_N1	PIN_D13	3.3-V LV...default)		24mA (default)	
result[0]	Output	PIN_AB21	7	B7_N0	PIN_B11	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I1	Unknown	PIN_N25	5	B5_N1		3.3-V LV...default)		24mA (default)	
Chen_Kevin_I2	Unknown	PIN_N26	5	B5_N1		3.3-V LV...default)		24mA (default)	
Chen_Kevin_Cin	Unknown	PIN_P25	6	B6_N0		3.3-V LV...default)		24mA (default)	
Chen_Kevin_S	Unknown	PIN_AE23	7	B7_N0		3.3-V LV...default)		24mA (default)	
Chen_Kevin_Cout	Unknown	PIN_AF23	7	B7_N0		3.3-V LV...default)		24mA (default)	

Figure 34: Pin Planner of LPM's VHDL Code for 1-bit Full Adder



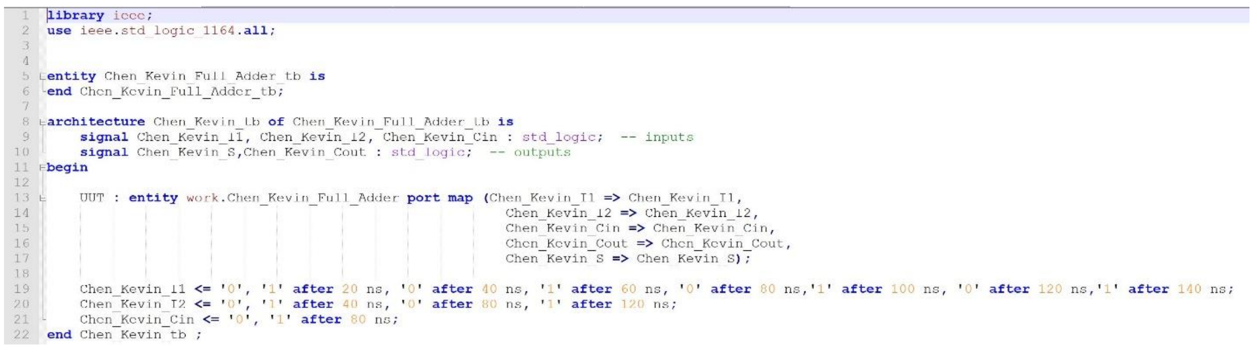
```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity Chen_Kevin_Half_Adder is
  port (Chen_Kevin_a, Chen_Kevin_b : in std_logic;
        Chen_Kevin_sum, Chen_Kevin_carry : out std_logic
     );
end Chen_Kevin_Half_Adder;

architecture Chen_Kevin_Arch of Chen_Kevin_Half_Adder is
begin
  Chen_Kevin_sum <= Chen_Kevin_a xor Chen_Kevin_b;
  Chen_Kevin_carry <= Chen_Kevin_a and Chen_Kevin_b;
end Chen_Kevin_Arch;
```

Figure 35: My VHDL Code for 1-bit Half Adder (Chen_Kevin_Half_Adder.VHD) Used for Full Adder



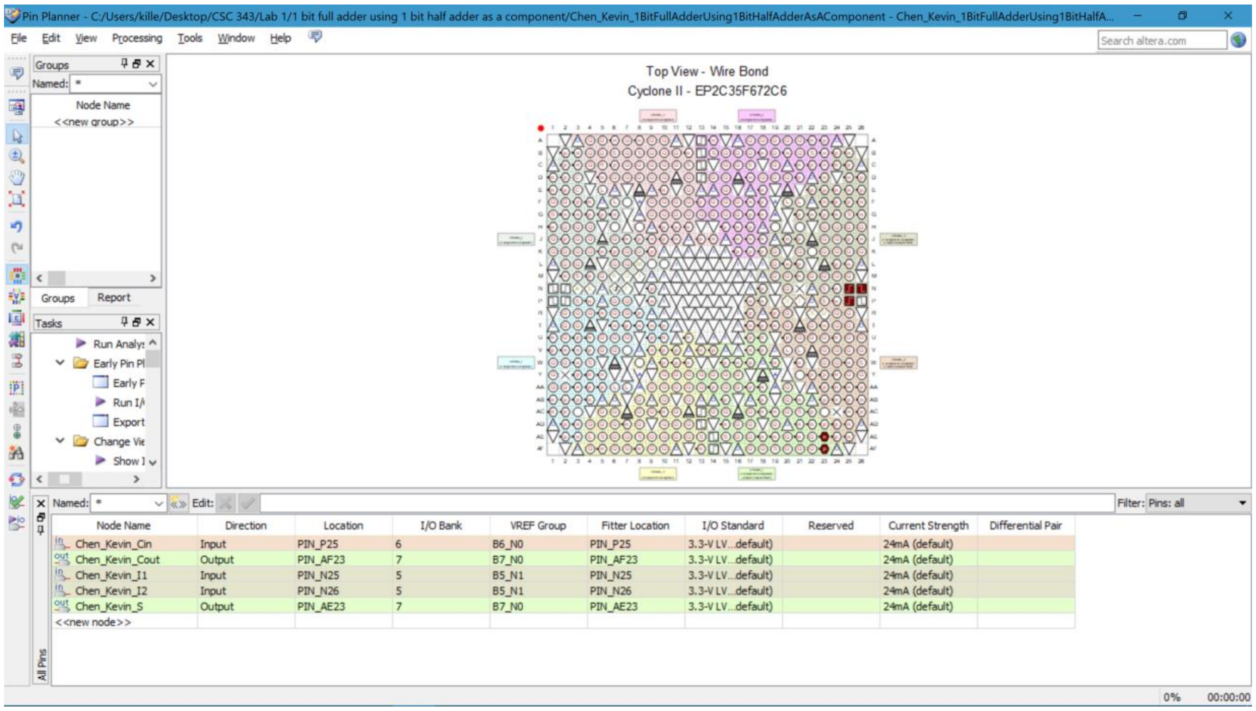
```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity Chen_Kevin_Full_Adder is
  port (Chen_Kevin_I1, Chen_Kevin_I2, Chen_Kevin_Cin : in std_logic;
        Chen_Kevin_S, Chen_Kevin_Cout : out std_logic
    );
end Chen_Kevin_Full_Adder;

architecture Chen_Kevin_Arch of Chen_Kevin_Full_Adder is
    signal Chen_Kevin_sum1: STD_logic;
    signal Chen_Kevin_carry1: STD_logic;
    signal Chen_Kevin_carry2: STD_logic;

component Chen_Kevin_Half_Adder
  port (Chen_Kevin_a, Chen_Kevin_b : in std_logic;
        Chen_Kevin_sum, Chen_Kevin_carry : out std_logic
    );
end component;

begin
Chen_Kevin_Half1: Chen_Kevin_Half_Adder
    port map( Chen_Kevin_a => Chen_Kevin_I1,
              Chen_Kevin_b => Chen_Kevin_I2,
              Chen_Kevin_sum => Chen_Kevin_sum1,
              Chen_Kevin_carry => Chen_Kevin_carry1);
Chen_Kevin_Half2: Chen_Kevin_Half_Adder
    port map( Chen_Kevin_a => Chen_Kevin_sum1,
              Chen_Kevin_b => Chen_Kevin_Cin,
              Chen_Kevin_sum => Chen_Kevin_S,
              Chen_Kevin_carry => Chen_Kevin_carry2);
Chen_Kevin_Cout <= Chen_Kevin_carry1 or Chen_Kevin_carry2;
end Chen_Kevin_Arch;
```

Figure 36: My VHDL Code for 1-bit Full Adder (Chen_Kevin_Full_Adder.VHD)



```vhdl
library ieee;
use ieee.std_logic_1164.all;


entity Chen_Kevin_Full_Adder_tb is
end Chen_Kevin_Full_Adder_tb;

architecture Chen_Kevin_tb of Chen_Kevin_Full_Adder_tb is
    signal Chen_Kevin_I1, Chen_Kevin_I2, Chen_Kevin_Cin : std_logic;  -- inputs
    signal Chen_Kevin_S,Chen_Kevin_Cout : std_logic;  -- outputs
begin

    UUT : entity work.Chen_Kevin_Full_Adder port map (Chen_Kevin_I1 => Chen_Kevin_I1,
                                                      Chen_Kevin_I2 => Chen_Kevin_I2,
                                                      Chen_Kevin_Cin => Chen_Kevin_Cin,
                                                      Chen_Kevin_Cout => Chen_Kevin_Cout,
                                                      Chen_Kevin_S => Chen_Kevin_S);

    Chen_Kevin_I1 <= '0', '1' after 20 ns, '0' after 40 ns, '1' after 60 ns, '0' after 80 ns,'1' after 100 ns, '0' after 120 ns,'1' after 140 ns;
    Chen_Kevin_I2 <= '0', '1' after 40 ns, '0' after 80 ns, '1' after 120 ns;
    Chen_Kevin_Cin <= '0', '1' after 80 ns;
end Chen_Kevin_tb ;
```

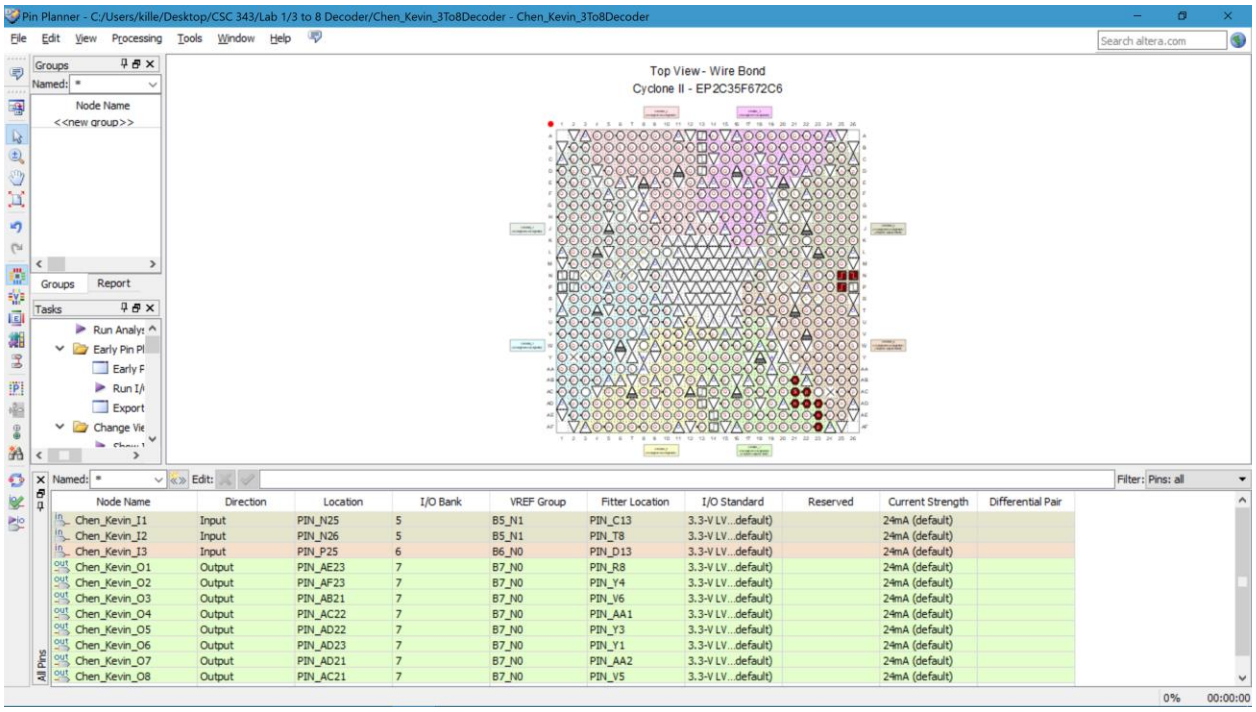
Figure 37: My VHDL Testbench Code for 1-bit Full Adder (Chen_Kevin_Full_Adder_tb.VHD)


Pin Planner - C:/Users/kille/Desktop/CSC 343/Lab 1/1 bit full adder using 1 bit half adder as a component/Chen_Kevin_1BitFullAdderUsing1BitHalfAdderAsAComponent - Chen_Kevin_1BitFullAdderUsing1BitHalfA...
File Edit View Processing Tools Window Help
Search altera.com
Groups
Named: *
Node Name
<<new group>>
Groups Report
Tasks
Run Analys
Early Pin Pl
Early F
Run I/
Export
Change Vie
Show I
Top View - Wire Bond
Cyclone II - EP2C35F672C6
Named: *
Edit:
Filter: Pins: all
All Pins
0%
00:00:00

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
Chen_Kevin_Cin	Input	PIN_P25	6	B6_N0	PIN_P25	3.3-V LV...default)		24mA (default)	
Chen_Kevin_Cout	Output	PIN_AF23	7	B7_N0	PIN_AF23	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I1	Input	PIN_N25	5	B5_N1	PIN_N25	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I2	Input	PIN_N26	5	B5_N1	PIN_N26	3.3-V LV...default)		24mA (default)	
Chen_Kevin_S	Output	PIN_AE23	7	B7_N0	PIN_AE23	3.3-V LV...default)		24mA (default)	
<<new node>>									

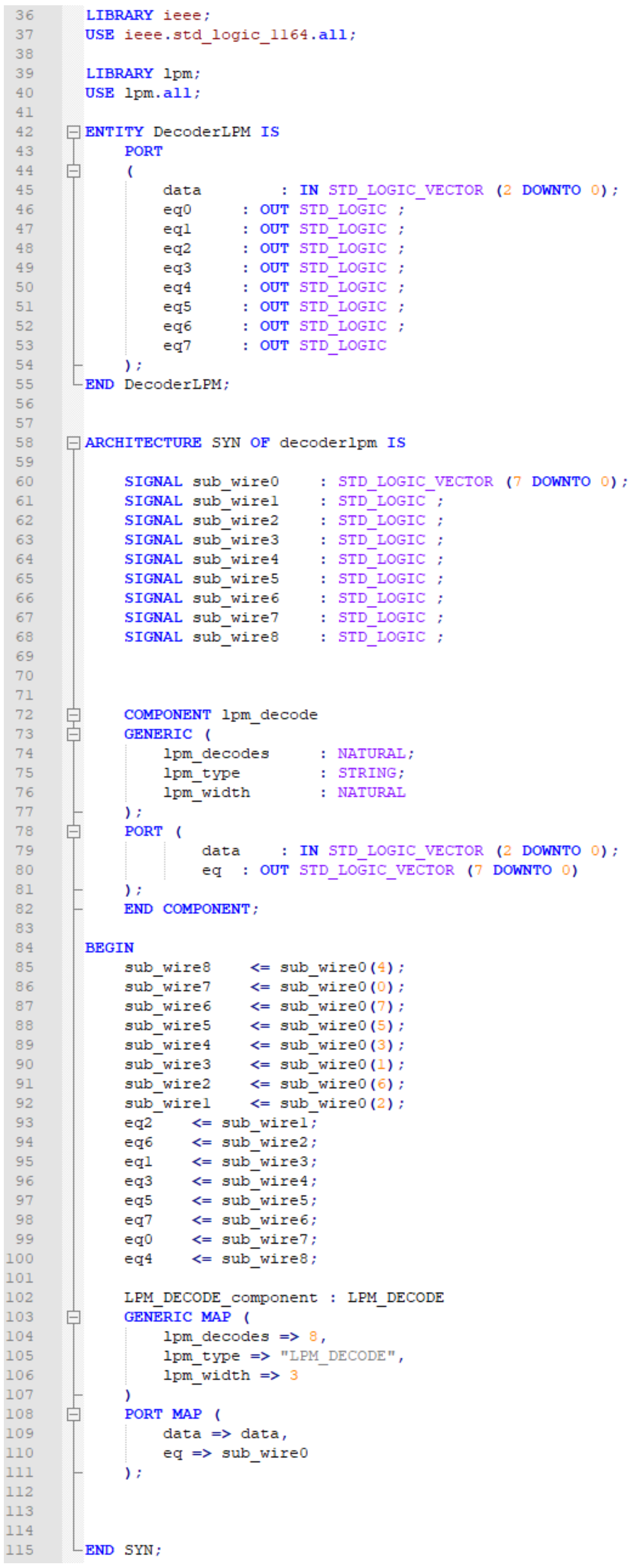
Figure 38: Pin Planner of my VHDL Code for 1-bit Full Adder

3-to-8 Decoder



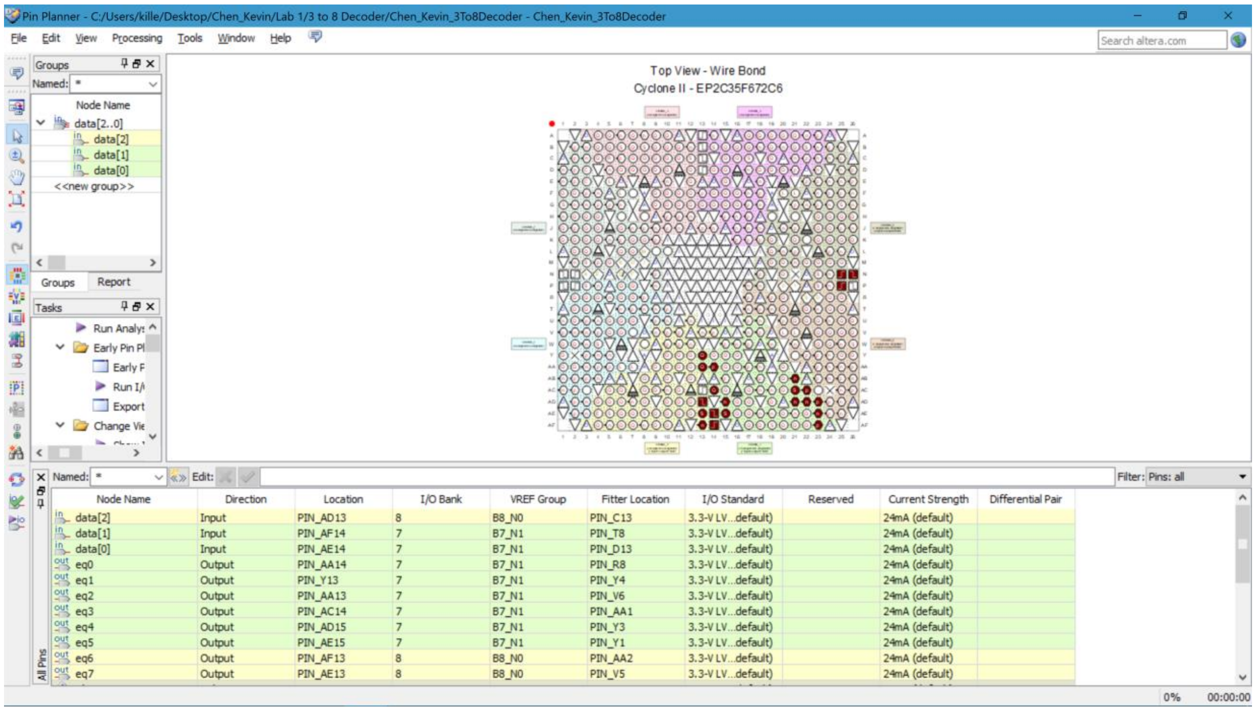
Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
Chen_Kevin_I1	Input	PIN_N25	5	B5_N1	PIN_C13	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I2	Input	PIN_N26	5	B5_N1	PIN_T8	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I3	Input	PIN_P25	6	B6_N0	PIN_D13	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O1	Output	PIN_AE23	7	B7_N0	PIN_R8	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O2	Output	PIN_AF23	7	B7_N0	PIN_Y4	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O3	Output	PIN_AB21	7	B7_N0	PIN_V6	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O4	Output	PIN_AC22	7	B7_N0	PIN_AA1	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O5	Output	PIN_AD22	7	B7_N0	PIN_Y3	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O6	Output	PIN_AD23	7	B7_N0	PIN_Y1	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O7	Output	PIN_AD21	7	B7_N0	PIN_AA2	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O8	Output	PIN_AC21	7	B7_N0	PIN_V5	3.3-V LV...default)		24mA (default)	

Figure 39: Pin Planner of Block Diagram/Schematic File of 3-to-8 Decoder



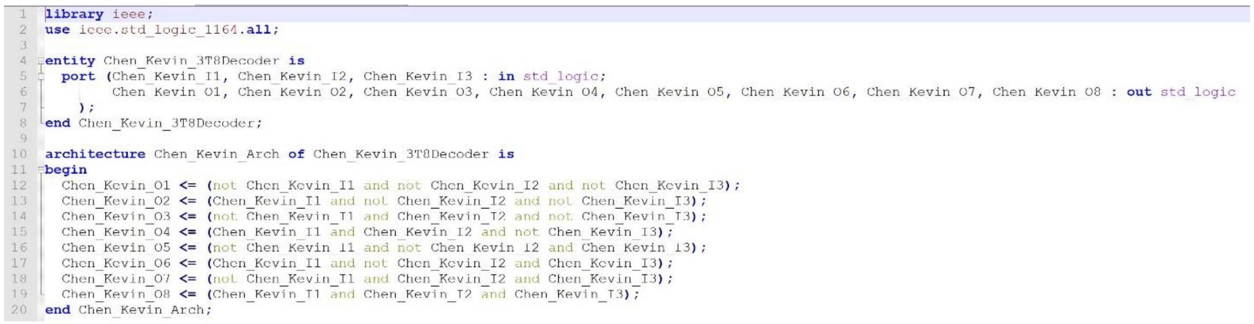
```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.all;

LIBRARY lpm;
USE lpm.all;

ENTITY DecoderLPM IS
    PORT
    (
        data        : IN STD_LOGIC_VECTOR (2 DOWNTO 0);
        eq0     : OUT STD_LOGIC ;
        eq1     : OUT STD_LOGIC ;
        eq2     : OUT STD_LOGIC ;
        eq3     : OUT STD_LOGIC ;
        eq4     : OUT STD_LOGIC ;
        eq5     : OUT STD_LOGIC ;
        eq6     : OUT STD_LOGIC ;
        eq7     : OUT STD_LOGIC
    );
END DecoderLPM;


ARCHITECTURE SYN OF decoderlpm IS

    SIGNAL sub_wire0    : STD_LOGIC_VECTOR (7 DOWNTO 0);
    SIGNAL sub_wire1    : STD_LOGIC ;
    SIGNAL sub_wire2    : STD_LOGIC ;
    SIGNAL sub_wire3    : STD_LOGIC ;
    SIGNAL sub_wire4    : STD_LOGIC ;
    SIGNAL sub_wire5    : STD_LOGIC ;
    SIGNAL sub_wire6    : STD_LOGIC ;
    SIGNAL sub_wire7    : STD_LOGIC ;
    SIGNAL sub_wire8    : STD_LOGIC ;



    COMPONENT lpm_decode
    GENERIC (
        lpm_decodes     : NATURAL;
        lpm_type        : STRING;
        lpm_width       : NATURAL
    );
    PORT (
            data    : IN STD_LOGIC_VECTOR (2 DOWNTO 0);
            eq  : OUT STD_LOGIC_VECTOR (7 DOWNTO 0)
    );
    END COMPONENT;

BEGIN
    sub_wire8    <= sub_wire0(4);
    sub_wire7    <= sub_wire0(0);
    sub_wire6    <= sub_wire0(7);
    sub_wire5    <= sub_wire0(5);
    sub_wire4    <= sub_wire0(3);
    sub_wire3    <= sub_wire0(1);
    sub_wire2    <= sub_wire0(6);
    sub_wire1    <= sub_wire0(2);
    eq2    <= sub_wire1;
    eq6    <= sub_wire2;
    eq1    <= sub_wire3;
    eq3    <= sub_wire4;
    eq5    <= sub_wire5;
    eq7    <= sub_wire6;
    eq0    <= sub_wire7;
    eq4    <= sub_wire8;

    LPM_DECODE_component : LPM_DECODE
    GENERIC MAP (
        lpm_decodes => 8,
        lpm_type => "LPM_DECODE",
        lpm_width => 3
    )
    PORT MAP (
        data => data,
        eq => sub_wire0
    );



END SYN;
```

Figure 40: LPM's VHDL Code for 3-to-8 Decoder (Chen_Kevin_DecoderxLPM.VHD)



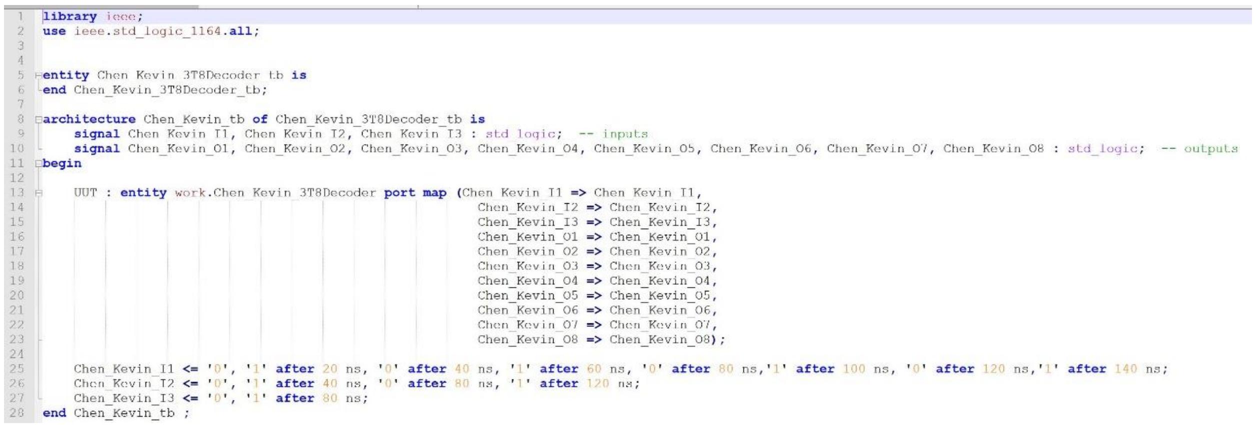
Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
data[2]	Input	PIN_AD13	8	B8_N0	PIN_C13	3.3-V LV...default)		24mA (default)	
data[1]	Input	PIN_AF14	7	B7_N1	PIN_T8	3.3-V LV...default)		24mA (default)	
data[0]	Input	PIN_AE14	7	B7_N1	PIN_D13	3.3-V LV...default)		24mA (default)	
eq0	Output	PIN_AA14	7	B7_N1	PIN_R8	3.3-V LV...default)		24mA (default)	
eq1	Output	PIN_Y13	7	B7_N1	PIN_Y4	3.3-V LV...default)		24mA (default)	
eq2	Output	PIN_AA13	7	B7_N1	PIN_V6	3.3-V LV...default)		24mA (default)	
eq3	Output	PIN_AC14	7	B7_N1	PIN_AA1	3.3-V LV...default)		24mA (default)	
eq4	Output	PIN_AD15	7	B7_N1	PIN_Y3	3.3-V LV...default)		24mA (default)	
eq5	Output	PIN_AE15	7	B7_N1	PIN_Y1	3.3-V LV...default)		24mA (default)	
eq6	Output	PIN_AF13	8	B8_N0	PIN_AA2	3.3-V LV...default)		24mA (default)	
eq7	Output	PIN_AE13	8	B8_N0	PIN_V5	3.3-V LV...default)		24mA (default)	

Figure 41: Pin Planner of LPM's VHDL Code for 3-to-8 Decoder



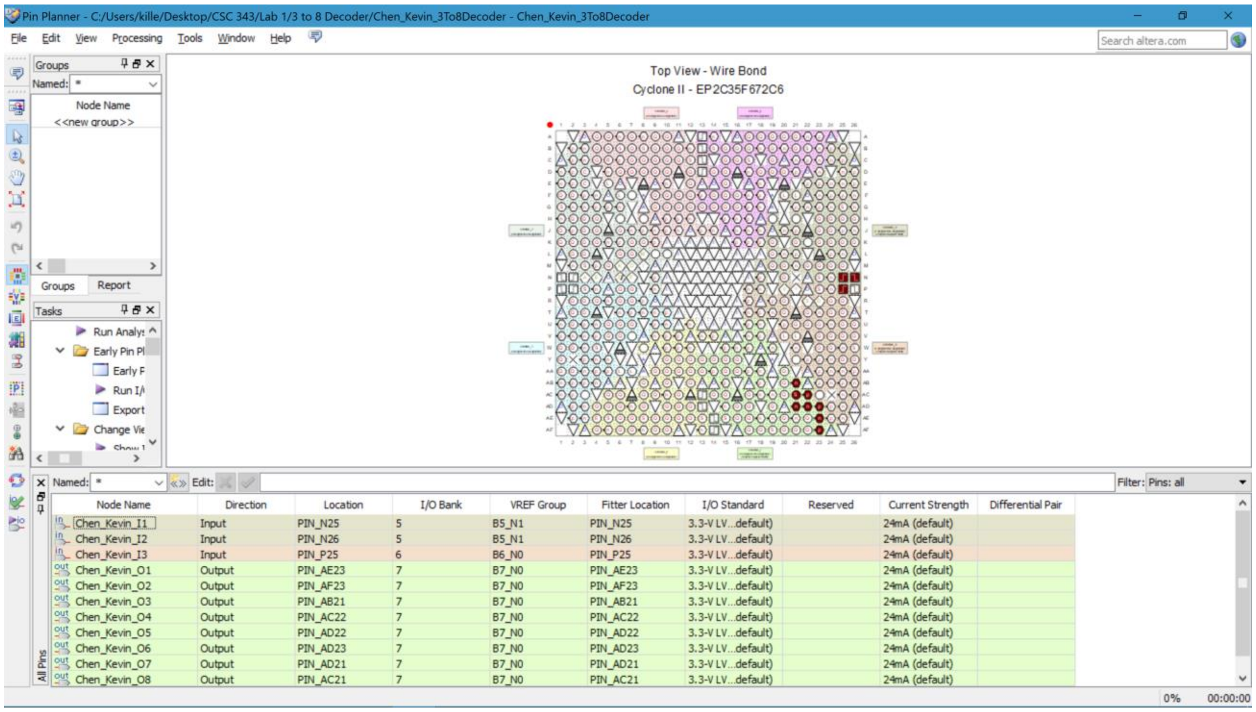
```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity Chen_Kevin_3T8Decoder is
  port (Chen_Kevin_I1, Chen_Kevin_I2, Chen_Kevin_I3 : in std_logic;
        Chen_Kevin_O1, Chen_Kevin_O2, Chen_Kevin_O3, Chen_Kevin_O4, Chen_Kevin_O5, Chen_Kevin_O6, Chen_Kevin_O7, Chen_Kevin_O8 : out std_logic
    );
end Chen_Kevin_3T8Decoder;

architecture Chen_Kevin_Arch of Chen_Kevin_3T8Decoder is
begin
  Chen_Kevin_O1 <= (not Chen_Kevin_I1 and not Chen_Kevin_I2 and not Chen_Kevin_I3);
  Chen_Kevin_O2 <= (Chen_Kevin_I1 and not Chen_Kevin_I2 and not Chen_Kevin_I3);
  Chen_Kevin_O3 <= (not Chen_Kevin_I1 and Chen_Kevin_I2 and not Chen_Kevin_I3);
  Chen_Kevin_O4 <= (Chen_Kevin_I1 and Chen_Kevin_I2 and not Chen_Kevin_I3);
  Chen_Kevin_O5 <= (not Chen_Kevin_I1 and not Chen_Kevin_I2 and Chen_Kevin_I3);
  Chen_Kevin_O6 <= (Chen_Kevin_I1 and not Chen_Kevin_I2 and Chen_Kevin_I3);
  Chen_Kevin_O7 <= (not Chen_Kevin_I1 and Chen_Kevin_I2 and Chen_Kevin_I3);
  Chen_Kevin_O8 <= (Chen_Kevin_I1 and Chen_Kevin_I2 and Chen_Kevin_I3);
end Chen_Kevin_Arch;
```

Figure 42: My VHDL Code for 3-to-8 Decoder (Chen_Kevin_3T8Decoder.VHD)



```vhdl
library ieee;
use ieee.std_logic_1164.all;


entity Chen_Kevin_3T8Decoder_tb is
end Chen_Kevin_3T8Decoder_tb;

architecture Chen_Kevin_tb of Chen_Kevin_3T8Decoder_tb is
    signal Chen_Kevin_I1, Chen_Kevin_I2, Chen_Kevin_I3 : std_logic;  -- inputs
    signal Chen_Kevin_O1, Chen_Kevin_O2, Chen_Kevin_O3, Chen_Kevin_O4, Chen_Kevin_O5, Chen_Kevin_O6, Chen_Kevin_O7, Chen_Kevin_O8 : std_logic;  -- outputs
begin

    UUT : entity work.Chen_Kevin_3T8Decoder port map (Chen_Kevin_I1 => Chen_Kevin_I1,
                                                     Chen_Kevin_I2 => Chen_Kevin_I2,
                                                     Chen_Kevin_I3 => Chen_Kevin_I3,
                                                     Chen_Kevin_O1 => Chen_Kevin_O1,
                                                     Chen_Kevin_O2 => Chen_Kevin_O2,
                                                     Chen_Kevin_O3 => Chen_Kevin_O3,
                                                     Chen_Kevin_O4 => Chen_Kevin_O4,
                                                     Chen_Kevin_O5 => Chen_Kevin_O5,
                                                     Chen_Kevin_O6 => Chen_Kevin_O6,
                                                     Chen_Kevin_O7 => Chen_Kevin_O7,
                                                     Chen_Kevin_O8 => Chen_Kevin_O8);

    Chen_Kevin_I1 <= '0', '1' after 20 ns, '0' after 40 ns, '1' after 60 ns, '0' after 80 ns,'1' after 100 ns, '0' after 120 ns,'1' after 140 ns;
    Chen_Kevin_I2 <= '0', '1' after 40 ns, '0' after 80 ns, '1' after 120 ns;
    Chen_Kevin_I3 <= '0', '1' after 80 ns;
end Chen_Kevin_tb ;
```

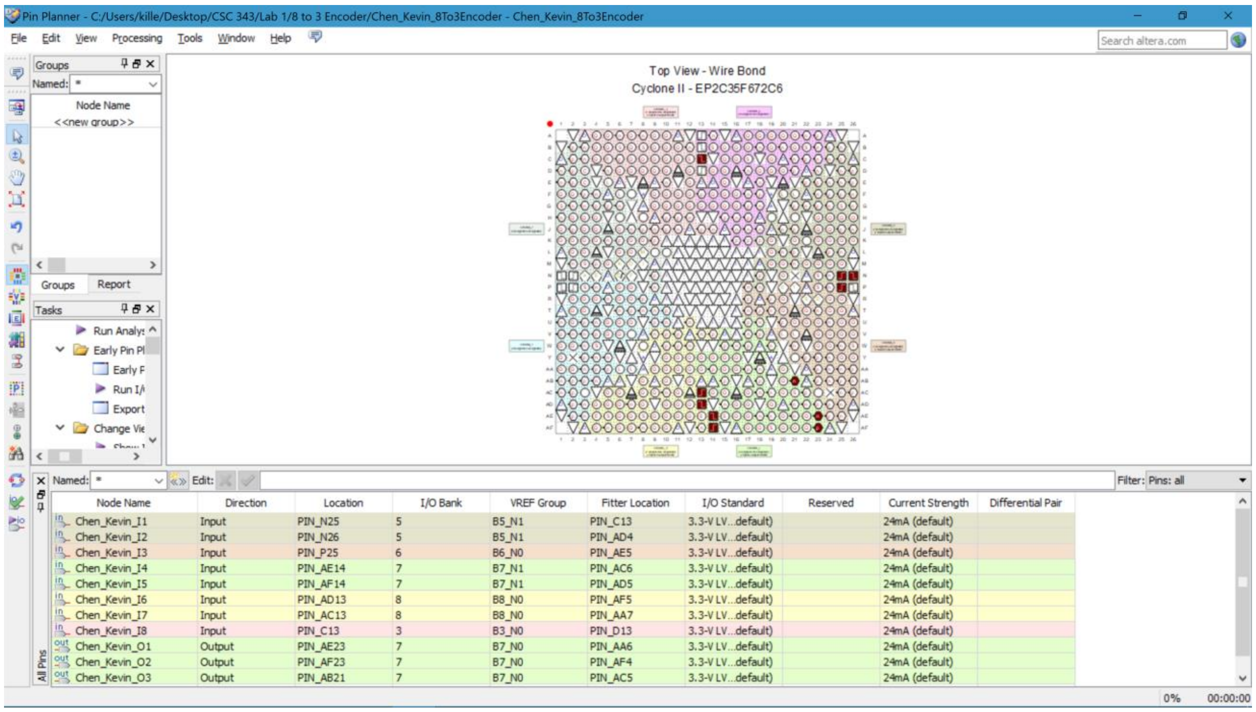
Figure 43: My VHDL Testbench Code for 3-to-8 Decoder (Chen_Kevin_3T8Decoder_tb.VHD)



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
Chen_Kevin_I1	Input	PIN_N25	5	B5_N1	PIN_N25	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I2	Input	PIN_N26	5	B5_N1	PIN_N26	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I3	Input	PIN_P25	6	B6_N0	PIN_P25	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O1	Output	PIN_AE23	7	B7_N0	PIN_AE23	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O2	Output	PIN_AF23	7	B7_N0	PIN_AF23	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O3	Output	PIN_AB21	7	B7_N0	PIN_AB21	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O4	Output	PIN_AC22	7	B7_N0	PIN_AC22	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O5	Output	PIN_AD22	7	B7_N0	PIN_AD22	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O6	Output	PIN_AD23	7	B7_N0	PIN_AD23	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O7	Output	PIN_AD21	7	B7_N0	PIN_AD21	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O8	Output	PIN_AC21	7	B7_N0	PIN_AC21	3.3-V LV...default)		24mA (default)	

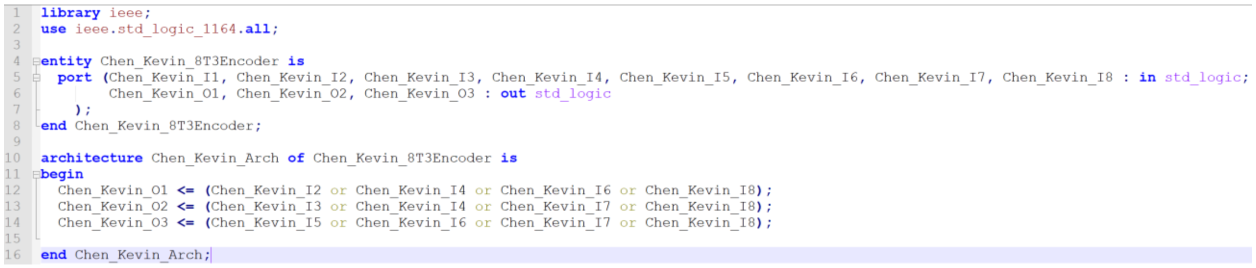
Figure 44: Pin Planner of my VHDL Code for 3-to-8 Decoder

8-to-3 Encoder



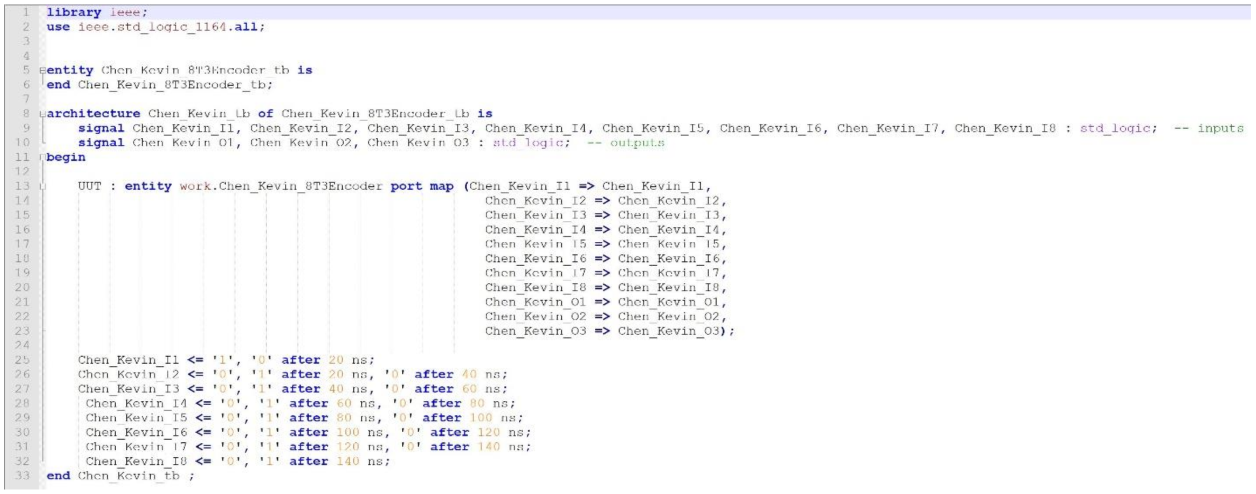
Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
Chen_Kevin_I1	Input	PIN_N25	5	B5_N1	PIN_C13	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I2	Input	PIN_N26	5	B5_N1	PIN_AD4	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I3	Input	PIN_P25	6	B6_N0	PIN_AE5	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I4	Input	PIN_AE14	7	B7_N1	PIN_AC6	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I5	Input	PIN_AF14	7	B7_N1	PIN_AD5	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I6	Input	PIN_AD13	8	B8_N0	PIN_AF5	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I7	Input	PIN_AC13	8	B8_N0	PIN_AA7	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I8	Input	PIN_C13	3	B3_N0	PIN_D13	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O1	Output	PIN_AE23	7	B7_N0	PIN_AA6	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O2	Output	PIN_AF23	7	B7_N0	PIN_AF4	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O3	Output	PIN_AB21	7	B7_N0	PIN_AC5	3.3-V LV...default)		24mA (default)	

Figure 45: Pin Planner of Block Diagram/Schematic File of 8-to-3 Encoder



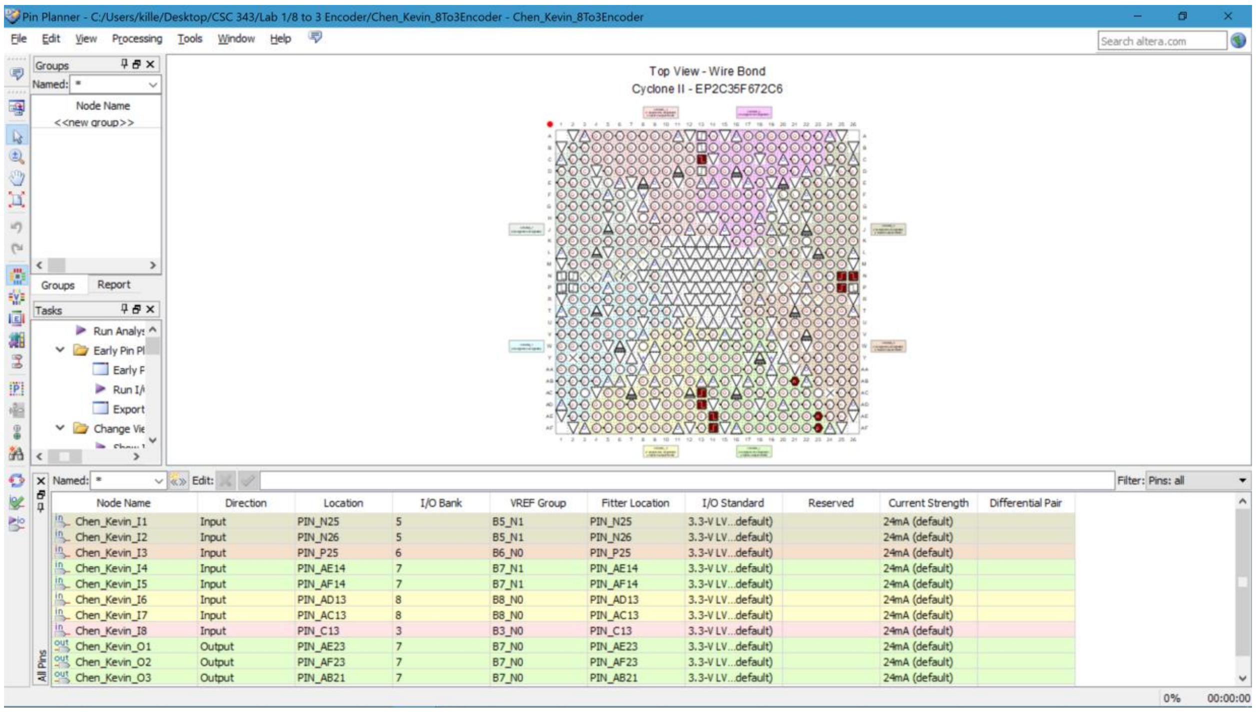
```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity Chen_Kevin_8T3Encoder is
  port (Chen_Kevin_I1, Chen_Kevin_I2, Chen_Kevin_I3, Chen_Kevin_I4, Chen_Kevin_I5, Chen_Kevin_I6, Chen_Kevin_I7, Chen_Kevin_I8 : in std_logic;
        Chen_Kevin_O1, Chen_Kevin_O2, Chen_Kevin_O3 : out std_logic
    );
end Chen_Kevin_8T3Encoder;

architecture Chen_Kevin_Arch of Chen_Kevin_8T3Encoder is
begin
  Chen_Kevin_O1 <= (Chen_Kevin_I2 or Chen_Kevin_I4 or Chen_Kevin_I6 or Chen_Kevin_I8);
  Chen_Kevin_O2 <= (Chen_Kevin_I3 or Chen_Kevin_I4 or Chen_Kevin_I7 or Chen_Kevin_I8);
  Chen_Kevin_O3 <= (Chen_Kevin_I5 or Chen_Kevin_I6 or Chen_Kevin_I7 or Chen_Kevin_I8);

end Chen_Kevin_Arch;
```

Figure 46: My VHDL Code for 8-to-3 Encoder (Chen_Kevin_8T3Encoder.VHD)



```
library ieee;
use ieee.std_logic_1164.all;


entity Chen_Kevin_8T3Encoder_tb is
end Chen_Kevin_8T3Encoder_tb;

architecture Chen_Kevin_tb of Chen_Kevin_8T3Encoder_tb is
    signal Chen_Kevin_I1, Chen_Kevin_I2, Chen_Kevin_I3, Chen_Kevin_I4, Chen_Kevin_I5, Chen_Kevin_I6, Chen_Kevin_I7, Chen_Kevin_I8 : std_logic;  -- inputs
    signal Chen_Kevin_O1, Chen_Kevin_O2, Chen_Kevin_O3 : std_logic;  -- outputs
begin

    UUT : entity work.Chen_Kevin_8T3Encoder port map (Chen_Kevin_I1 => Chen_Kevin_I1,
                                                     Chen_Kevin_I2 => Chen_Kevin_I2,
                                                     Chen_Kevin_I3 => Chen_Kevin_I3,
                                                     Chen_Kevin_I4 => Chen_Kevin_I4,
                                                     Chen_Kevin_I5 => Chen_Kevin_I5,
                                                     Chen_Kevin_I6 => Chen_Kevin_I6,
                                                     Chen_Kevin_I7 => Chen_Kevin_I7,
                                                     Chen_Kevin_I8 => Chen_Kevin_I8,
                                                     Chen_Kevin_O1 => Chen_Kevin_O1,
                                                     Chen_Kevin_O2 => Chen_Kevin_O2,
                                                     Chen_Kevin_O3 => Chen_Kevin_O3);

    Chen_Kevin_I1 <= '1', '0' after 20 ns;
    Chen_Kevin_I2 <= '0', '1' after 20 ns, '0' after 40 ns;
    Chen_Kevin_I3 <= '0', '1' after 40 ns, '0' after 60 ns;
     Chen_Kevin_I4 <= '0', '1' after 60 ns, '0' after 80 ns;
     Chen_Kevin_I5 <= '0', '1' after 80 ns, '0' after 100 ns;
     Chen_Kevin_I6 <= '0', '1' after 100 ns, '0' after 120 ns;
     Chen_Kevin_I7 <= '0', '1' after 120 ns, '0' after 140 ns;
     Chen_Kevin_I8 <= '0', '1' after 140 ns;
end Chen_Kevin_tb ;
```

Figure 47: My VHDL Testbench Code for 8-to-3 Encoder (Chen_Kevin_8T3Encoder_tb.VHD)



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
Chen_Kevin_I1	Input	PIN_N25	5	B5_N1	PIN_N25	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I2	Input	PIN_N26	5	B5_N1	PIN_N26	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I3	Input	PIN_P25	6	B6_N0	PIN_P25	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I4	Input	PIN_AE14	7	B7_N1	PIN_AE14	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I5	Input	PIN_AF14	7	B7_N1	PIN_AF14	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I6	Input	PIN_AD13	8	B8_N0	PIN_AD13	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I7	Input	PIN_AC13	8	B8_N0	PIN_AC13	3.3-V LV...default)		24mA (default)	
Chen_Kevin_I8	Input	PIN_C13	3	B3_N0	PIN_C13	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O1	Output	PIN_AE23	7	B7_N0	PIN_AE23	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O2	Output	PIN_AF23	7	B7_N0	PIN_AF23	3.3-V LV...default)		24mA (default)	
Chen_Kevin_O3	Output	PIN_AB21	7	B7_N0	PIN_AB21	3.3-V LV...default)		24mA (default)	

Figure 48: Pin Planner of my VHDL Code for 8-to-3 Encoder