ECC3304 [Kumpulan 1] Digital Systems Design

LAB REPORT

Lab No. Lab Title: Lab Name: Date:	4 SCHEMATIC DESIGN ENTRY (BOARD TESTING) INTELLIGENT SYSTEMS ENGINEERING LAB 25TH NOVEMBER 2021
Lab No. Lab Title: Lab Name: Date:	5 SCHEMATIC DESIGN ENTRY (SIMULATION) ONLINE 2ND DECEMBER 2021
Lab No. Lab Title: Lab Name: Date:	4 & 5 EXTRA SCHEMATIC DESIGN (BLOCK DESIGN ENTRY) ONLINE 2ND DECEMBER 2021

Name: TABINA KAMAL

Matric No.: 208651

Lecturer: PUAN ROSLIZAH BINTI ALI

Demonstrator: SITI AQILAH BINTI AZMAN

A. AIMS

I. Schematic Design Entry (Simulation and Board Testing)

Schematic design entry was created in the Quartus Prime software.
A simple light circuit and a full 1-bit adder were simulated and also tested on the board.

II. Schematic Design (Block Entry – Simulation)

The block entry method was then used to design a 4-bit full adder and circuit was simulated.

B. INTRODUCTION

I. Schematic Design Entry (Simulation and Board Testing)

Schematic design entry method was used to design a simple light switch as well as a 1-bit full adder in the Quartus Prime software. The circuit was functionally simulated and also tested on the DE1-Soc board.

II. Schematic Design (Block Entry – Simulation)

The block entry method was then used to design a 4-bit full adder with the 1-bit full adder schematic design entry above. The circuit was functionally simulated.

C. METHODS

I. Schematic Design Entry (Simulation and Board Testing)

(1) Light circuit schematic design entry

(a) The instructions in the ‘Quartus Prime Introduction Using Schematic Designs’ guide was followed and the light circuit was constructed using the schematic entry method.

(b) The results of this simulation waveform were compared with that of the results obtained from the Verilog design entry from Lab 1.

(2) Design of 1-bit full adder

(a) A full 1-bit full adder was constructed using the schematic design entry method.

(b) The circuit was functionally simulated. The waveform was observed and verified with a truth table.

II. Schematic Design (Block Entry – Simulation)

(1) A new file was created, called ‘adder_disp’.

(2) The 1-bit full adder schematic file was imported into this project.

(3) A block design was created for the 1-bit full adder.

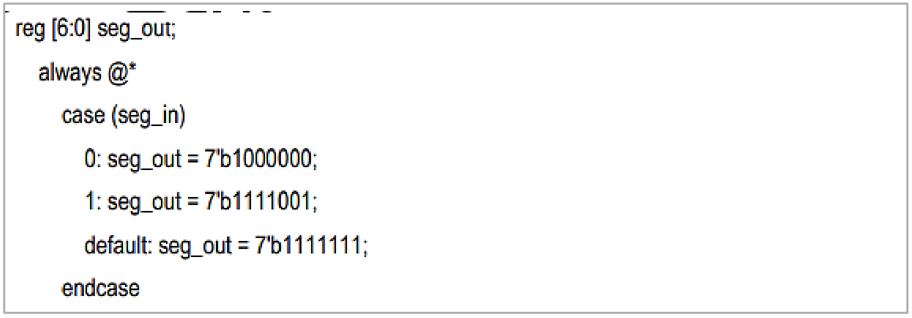
(a) In the ‘Block editor window’, a new block was inserted using the ‘Block Tool’ button.

(b) The block was renamed to the name of the schematic design of the 1-bit full adder from above. This will instantiate the schematic design in a hierarchical method.

(c) The input and output ports were defined.

(d) Any errors in the files were fixed using the ‘Analyze Current File’.

(4) The following steps were carried out
    (a) A new block was inserted and it was renamed to ‘seven_seg’.
    (b) The input and output ports were defined according to Figure 2.
    (c) A right-click was done on the block and ‘Create Design File from Selected Block’.
    (d) The following lines were included into the file:



```
reg [6:0] seg_out;
  always @*
    case (seg_in)
      0: seg_out = 7'b1000000;
      1: seg_out = 7'b1111001;
      default: seg_out = 7'b1111111;
    endcase
```

    (e) Any errors in the files were fixed using the ‘Analyze Current File’.

(5) The 1-bit full adder was connected to a seven-segment display.
(6) The input and output pins were connected to the block.
(7) The circuit was simulated functionally and the output waveform was obtained.
(8) A new project file was created named ‘adder4’ and a block diagram was used to implement a 4-bit full adder.
(9) The pins were assigned and the circuit was simulated functionally. The switches (SW0 - SW8) were used as input and 4 seven-segment displays were used as output (S0-S3) and LEDR0 as output (Cout3).

D. RESULTS

I. Schematic Design Entry (Simulation and Board Testing)

1. Light circuit schematic design entry

The truth table was created from the obtained output on the board. The value 0 represents OFF and 1 represents ON. The inputs are SW[0] and SW[1]. The output is LEDR[0].

Table 1: Truth table obtained from board testing

SW[0] / x_1	SW[1] / x_2	LEDR[0] / f
0	0	0
0	1	1
1	0	1
1	1	0

The results obtained from the simulation of the schematic design was the same as that obtained from the Verilog design from Lab 1A.

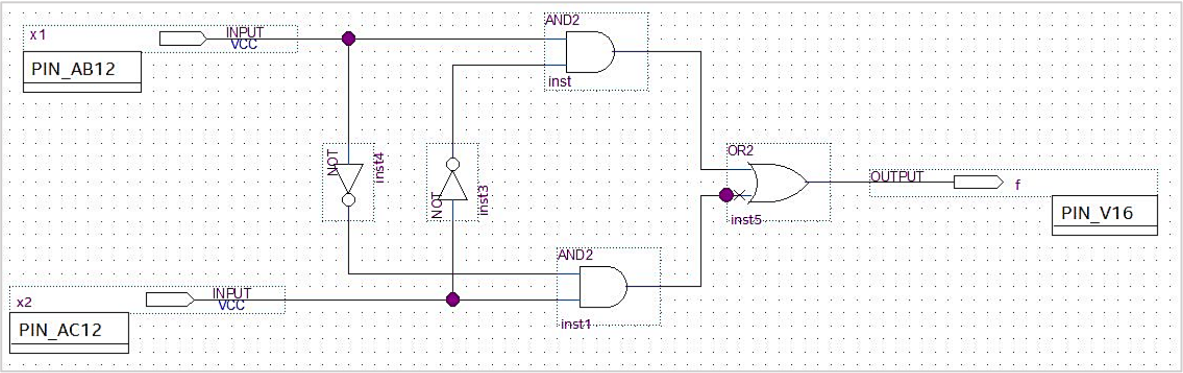
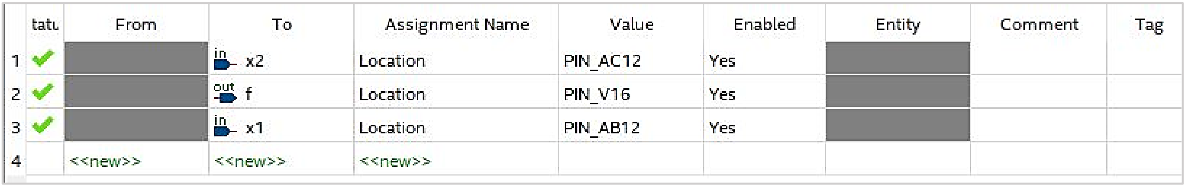
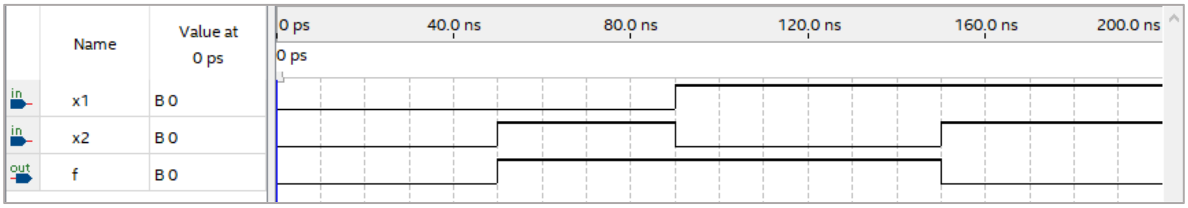


Figure 1: Schematic Design from Quartus_Prime_Schematic.pdf



	tatu	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
1	✓		in x2	Location	PIN_AC12	Yes			
2	✓		out f	Location	PIN_V16	Yes			
3	✓		in x1	Location	PIN_AB12	Yes			
4		<<new>>	<<new>>	<<new>>					

Figure 2: Pin Assignments for Schematic Design rom Quartus_Prime_Schematic.pdf



	Name	Value at 0 ps
in	x1	B 0
in	x2	B 0
out	f	B 0

Figure 3: Functional Simulation for Schematic Design rom Quartus_Prime_Schematic.pdf

2. *Design of 1-bit full adder*

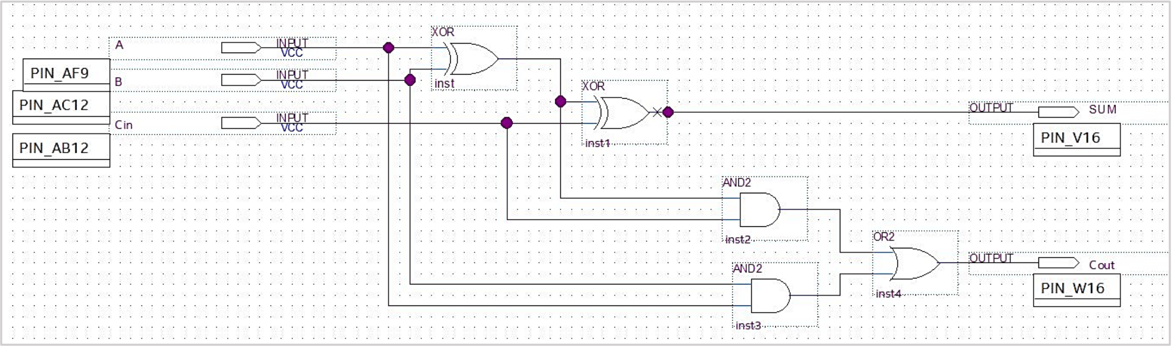
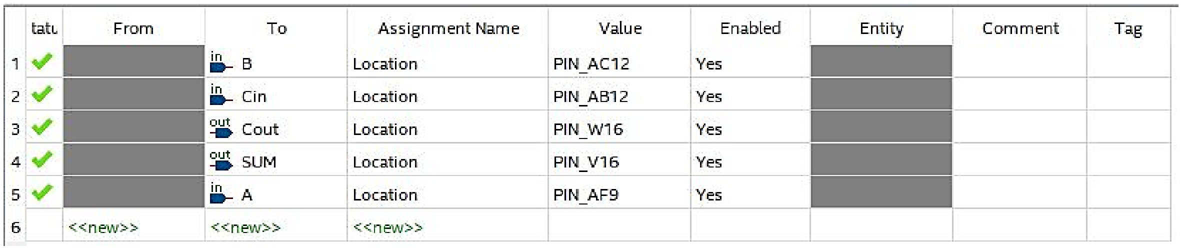


Figure 4: Schematic Design of 1-bit Full Adder



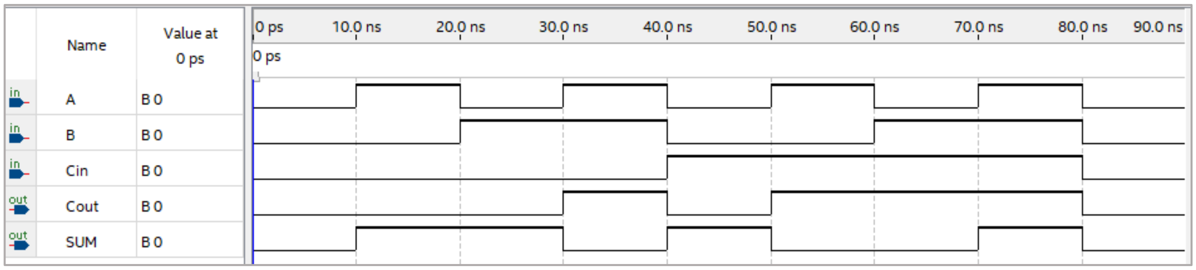
	tatu	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
1	✓		in B	Location	PIN_AC12	Yes			
2	✓		in Cin	Location	PIN_AB12	Yes			
3	✓		out Cout	Location	PIN_W16	Yes			
4	✓		out SUM	Location	PIN_V16	Yes			
5	✓		in A	Location	PIN_AF9	Yes			
6		<<new>>	<<new>>	<<new>>					

Figure 5: Pin Assignments for 1-bit Full Adder

Table 2: Truth table of 1-bit full adder obtained from board testing

INPUT			OUTPUT	
SW[2] / *A*	SW[1] / *B*	SW[0] / *Cin*	LEDR[0] / *Sum*	LEDR[1] / *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

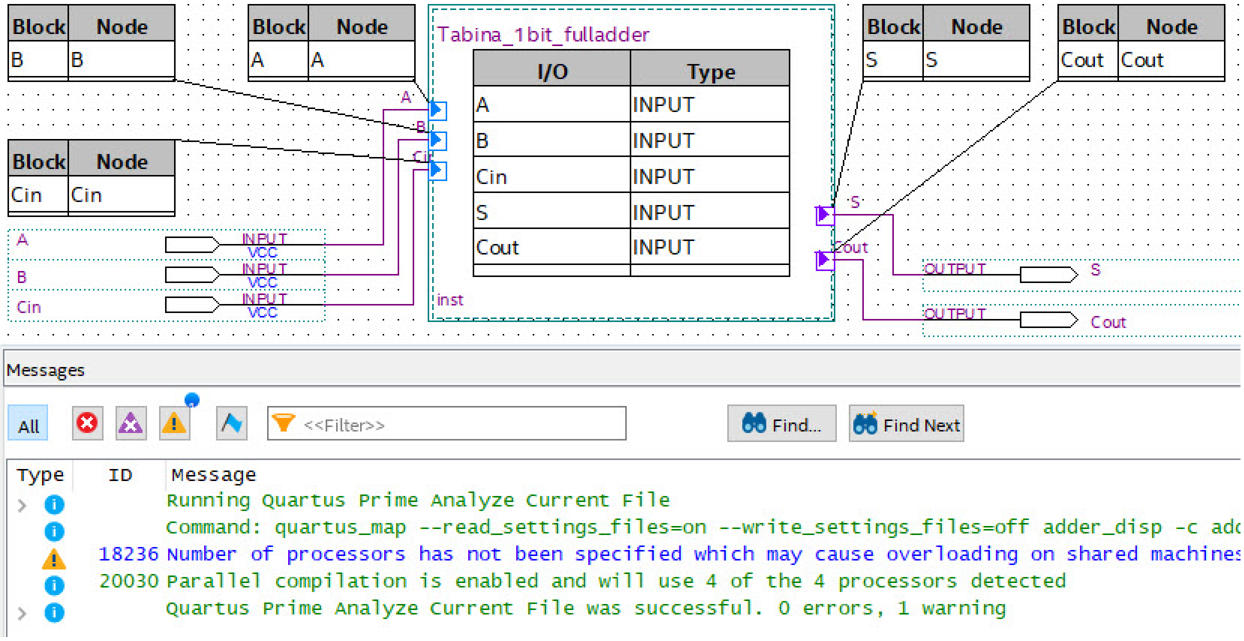
Truth table is the same as that of theoretical truth table of a 1-bit full adder.



	Name	Value at 0 ps
in	A	B 0
in	B	B 0
in	Cin	B 0
out	Cout	B 0
out	SUM	B 0

Figure 6: Functional Simulation for 1-bit Full Adder

II. Schematic Design (Block Entry – Simulation)



Block	Node
B	B

Block	Node
A	A

Block	Node
Cin	Cin

Tabina_1bit_fulladder

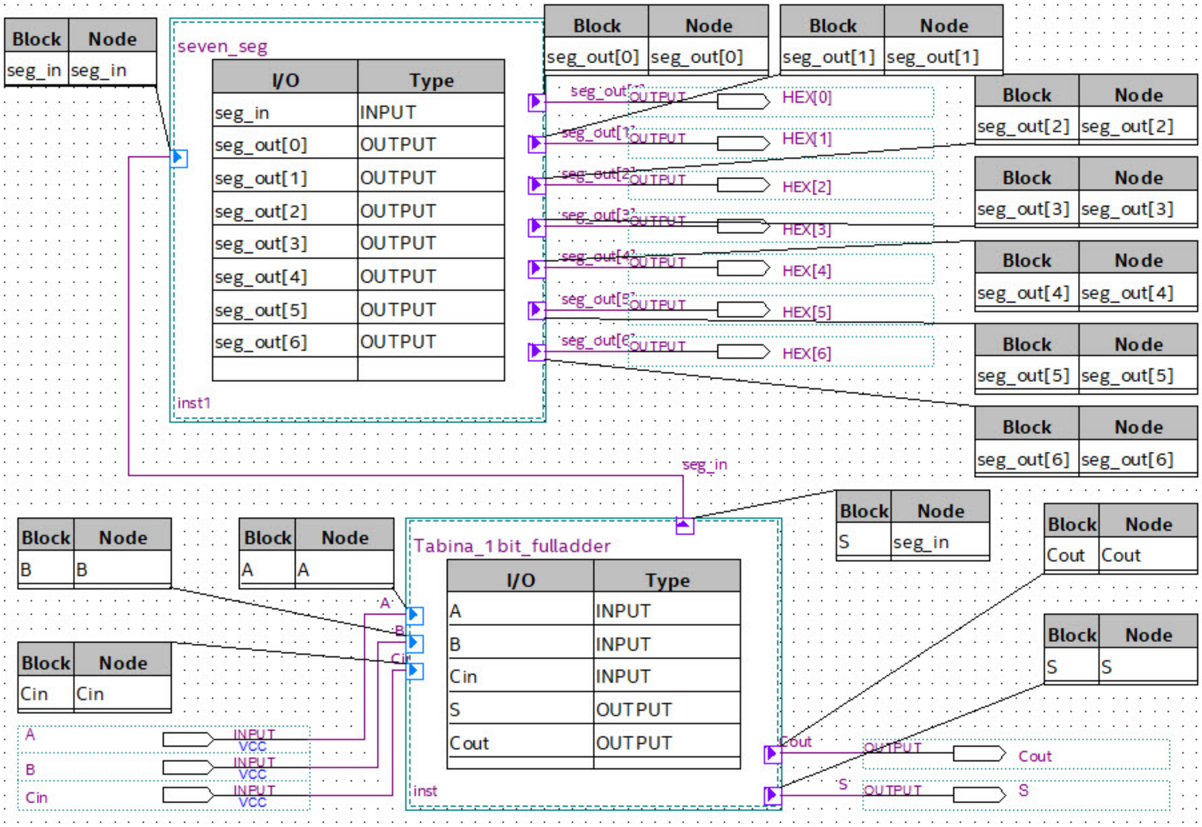
I/O	Type
A	INPUT
B	INPUT
Cin	INPUT
S	INPUT
Cout	INPUT

Block	Node
S	S

Block	Node
Cout	Cout

```
Type  ID    Message
            Running Quartus Prime Analyze Current File
            Command: quartus_map --read_settings_files=on --write_settings_files=off adder_disp -c adc
      18236 Number of processors has not been specified which may cause overloading on shared machines
      20030 Parallel compilation is enabled and will use 4 of the 4 processors detected
            Quartus Prime Analyze Current File was successful. 0 errors, 1 warning
```

Figure 7: Adder display block diagram with successful analysis



Block	Node
seg_in	seg_in

seven_seg

I/O	Type
seg_in	INPUT
seg_out[0]	OUTPUT
seg_out[1]	OUTPUT
seg_out[2]	OUTPUT
seg_out[3]	OUTPUT
seg_out[4]	OUTPUT
seg_out[5]	OUTPUT
seg_out[6]	OUTPUT

Block	Node
seg_out[0]	seg_out[0]

Block	Node
seg_out[1]	seg_out[1]

Block	Node
seg_out[2]	seg_out[2]

Block	Node
seg_out[3]	seg_out[3]

Block	Node
seg_out[4]	seg_out[4]

Block	Node
seg_out[5]	seg_out[5]

Block	Node
seg_out[6]	seg_out[6]

Block	Node
B	B

Block	Node
A	A

Block	Node
Cin	Cin

Tabina_1bit_fulladder

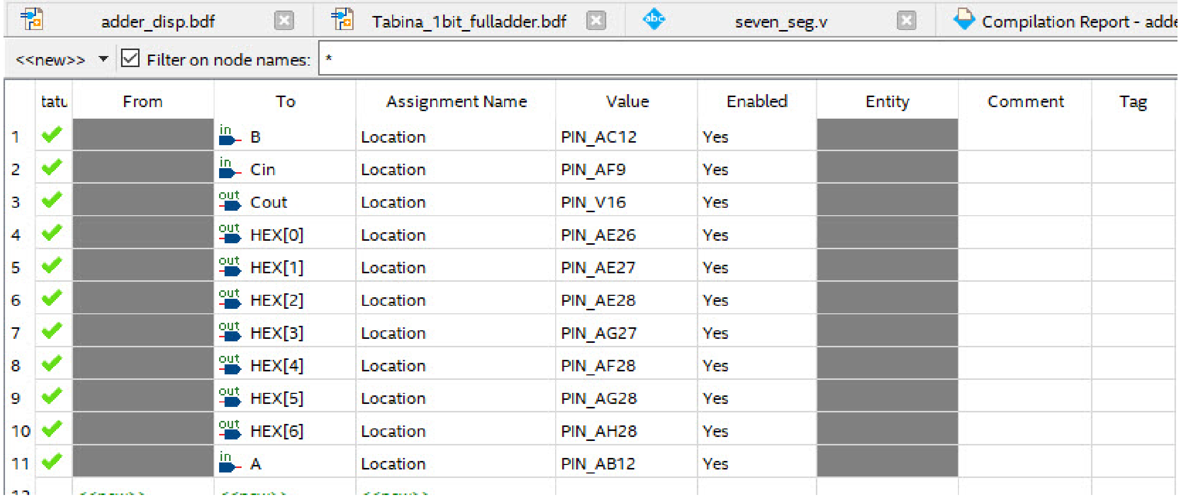
I/O	Type
A	INPUT
B	INPUT
Cin	INPUT
S	OUTPUT
Cout	OUTPUT

Block	Node
S	seg_in

Block	Node
Cout	Cout

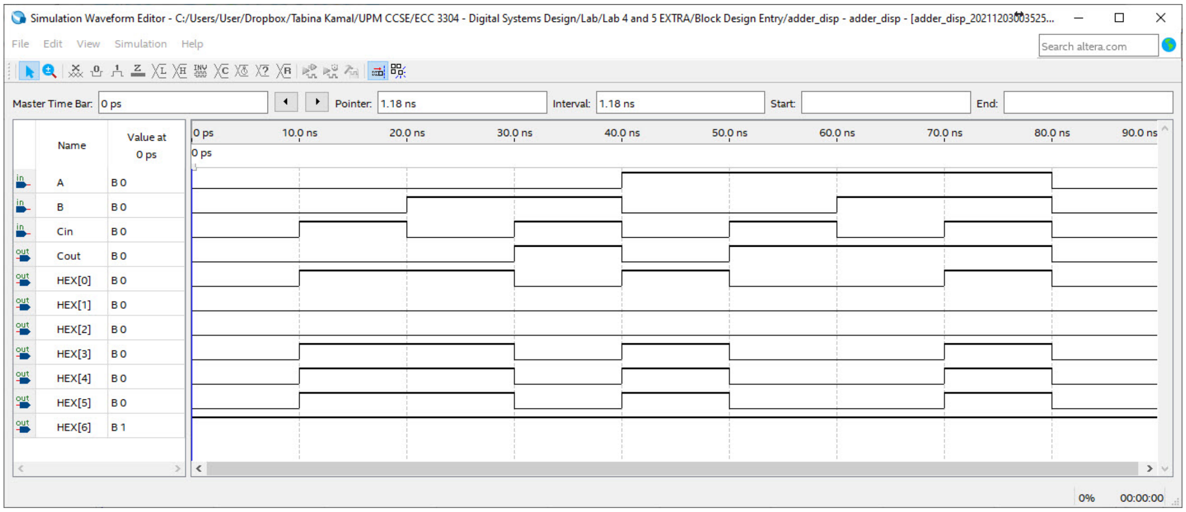
Block	Node
S	S

Figure 8: Full-adder with seven-segment display


adder_disp.bdf
Tabina_1bit_fulladder.bdf
seven_seg.v
Compilation Report - adde
<<new>>
☑ Filter on node names: *

	tatu	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
1	✓		in B	Location	PIN_AC12	Yes			
2	✓		in Cin	Location	PIN_AF9	Yes			
3	✓		out Cout	Location	PIN_V16	Yes			
4	✓		out HEX[0]	Location	PIN_AE26	Yes			
5	✓		out HEX[1]	Location	PIN_AE27	Yes			
6	✓		out HEX[2]	Location	PIN_AE28	Yes			
7	✓		out HEX[3]	Location	PIN_AG27	Yes			
8	✓		out HEX[4]	Location	PIN_AF28	Yes			
9	✓		out HEX[5]	Location	PIN_AG28	Yes			
10	✓		out HEX[6]	Location	PIN_AH28	Yes			
11	✓		in A	Location	PIN_AB12	Yes			

Figure 9: Pin assignments for adder with seven-segment display


Simulation Waveform Editor - C:/Users/User/Dropbox/Tabina Kamal/UPM CCSE/ECC 3304 - Digital Systems Design/Lab/Lab 4 and 5 EXTRA/Block Design Entry/adder_disp - adder_disp - [adder_disp_20211203003525...

Master Time Bar: 0 ps Pointer: 1.18 ns Interval: 1.18 ns Start: End:

Name	Value at 0 ps
in A	B 0
in B	B 0
in Cin	B 0
out Cout	B 0
out HEX[0]	B 0
out HEX[1]	B 0
out HEX[2]	B 0
out HEX[3]	B 0
out HEX[4]	B 0
out HEX[5]	B 0
out HEX[6]	B 1

Figure 10: Functional simulation of full-adder with seven-segment display

The output of the seven-segment display is active-LOW, thus when the waveform shows a ‘low’ level, that particular segment of the display turns on. The output obtained is that expected of a 1-bit full adder. The only two values displayed on the seven-segment display are 0 and 1.

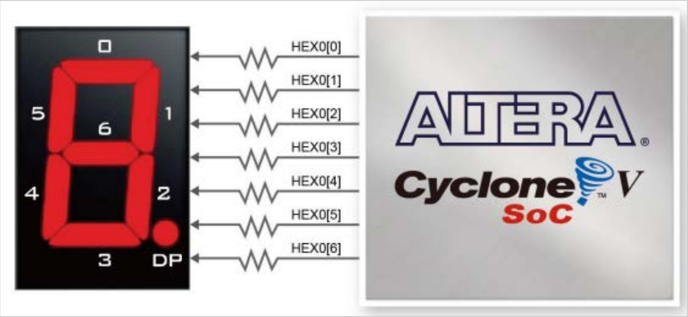


Figure 11: Segment names of the seven-segment display

Table 3: Truth table of 1-bit full adder with seven-segment display

INPUT			OUTPUT							
A	B	Cin	HEX[0]	HEX[1]	HEX[2]	HEX[3]	HEX[4]	HEX[5]	HEX[6]	Cout
0	0	0	0	0	0	0	0	0	1	0
0	0	1	1	0	0	1	1	1	1	0
0	1	0	1	0	0	1	1	1	1	0
0	1	1	0	0	0	0	0	0	1	1
1	0	0	1	0	0	1	1	1	1	0
1	0	1	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	1	1
1	1	1	1	0	0	1	1	1	1	1

Following the design of the 1-bit full adder with a seven-segment display, a 4-bit adder was designed using block entry method using the circuit designs above. Multiple attempts were made in designing the circuit. Figures 12 and 13 show the two different designs that were used. Figure 12 consists of four blocks of full-adders with the seven-segment display whilst Figure 13 consists of adder and seven-segment display blocks.

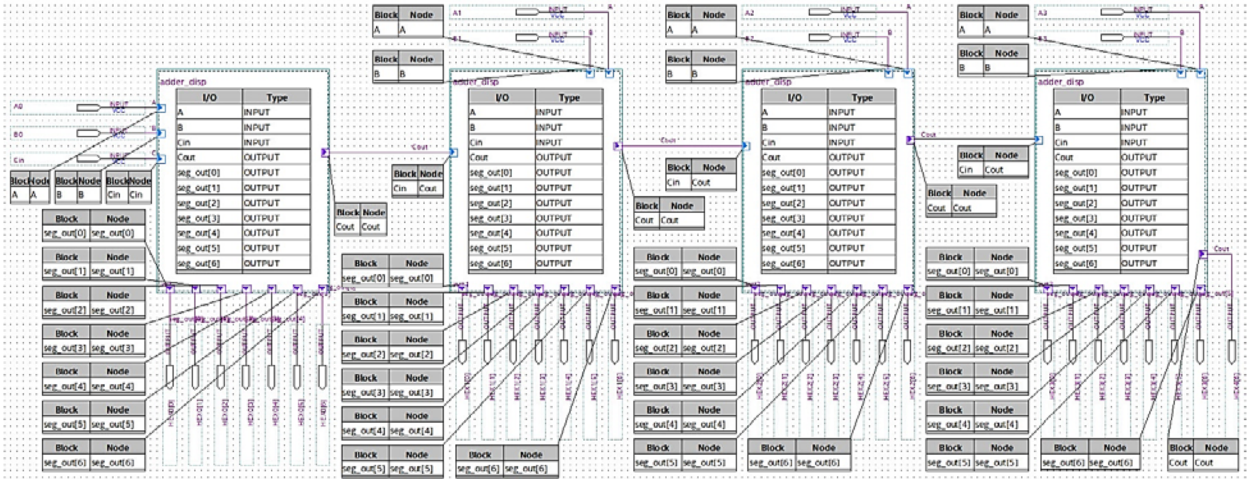


Figure 12: 4-bit full-adder circuit with seven-segment display [Attempt 1]

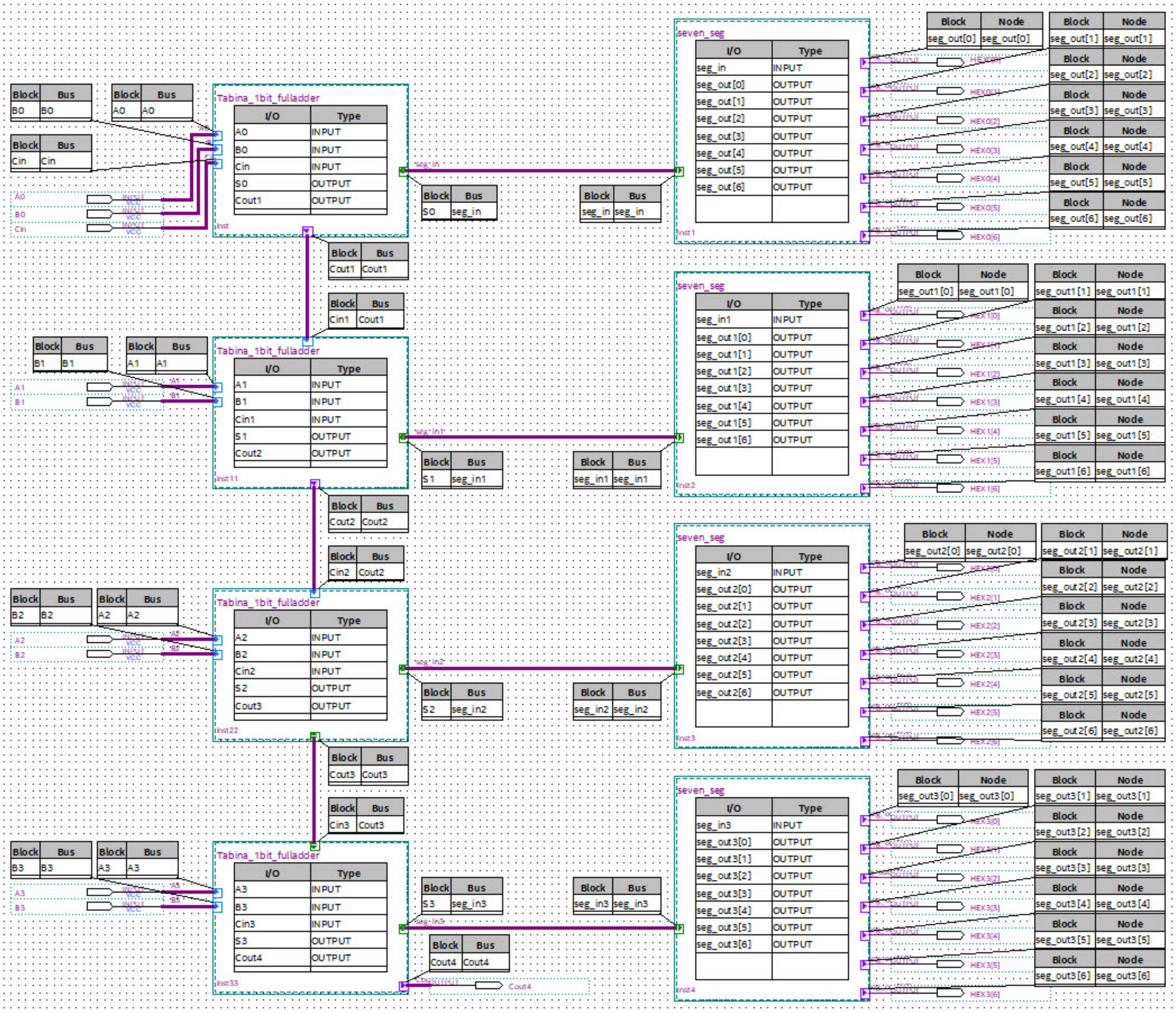
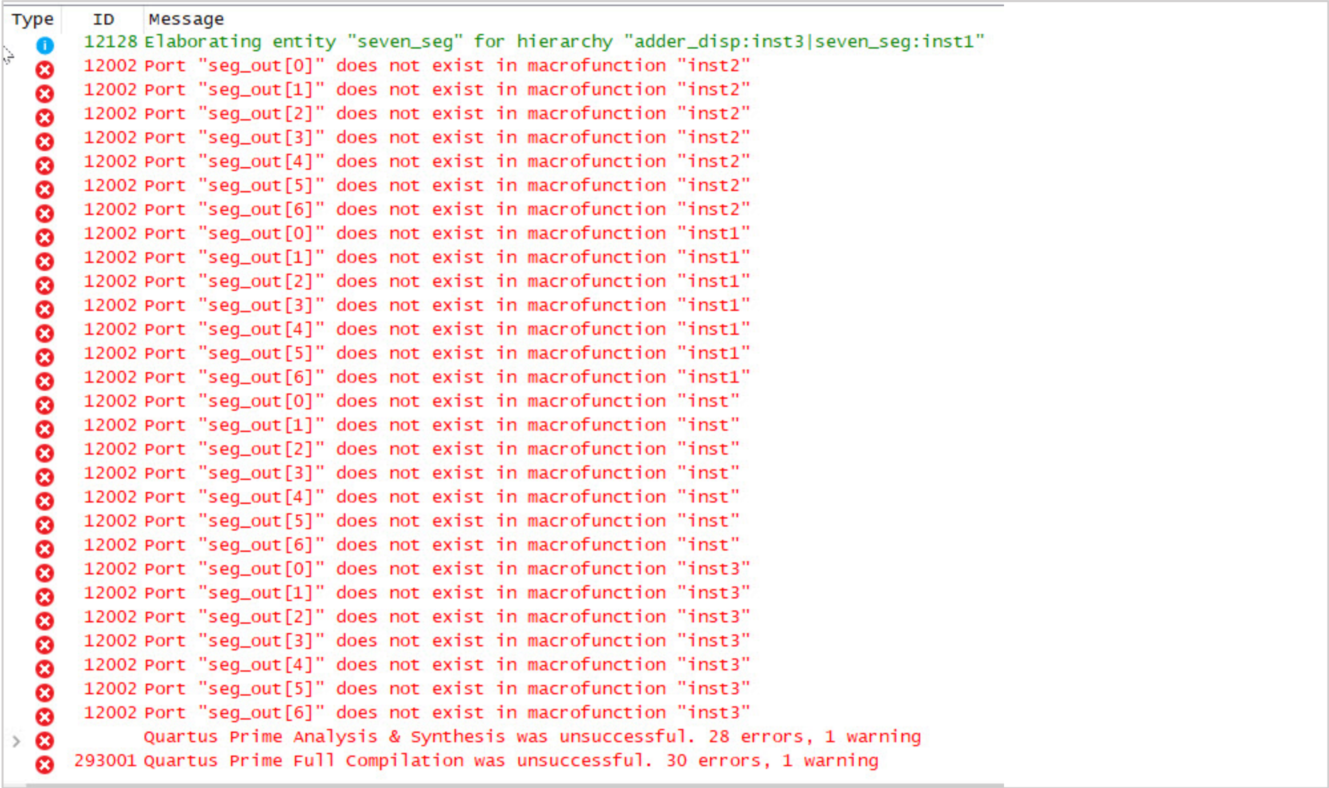


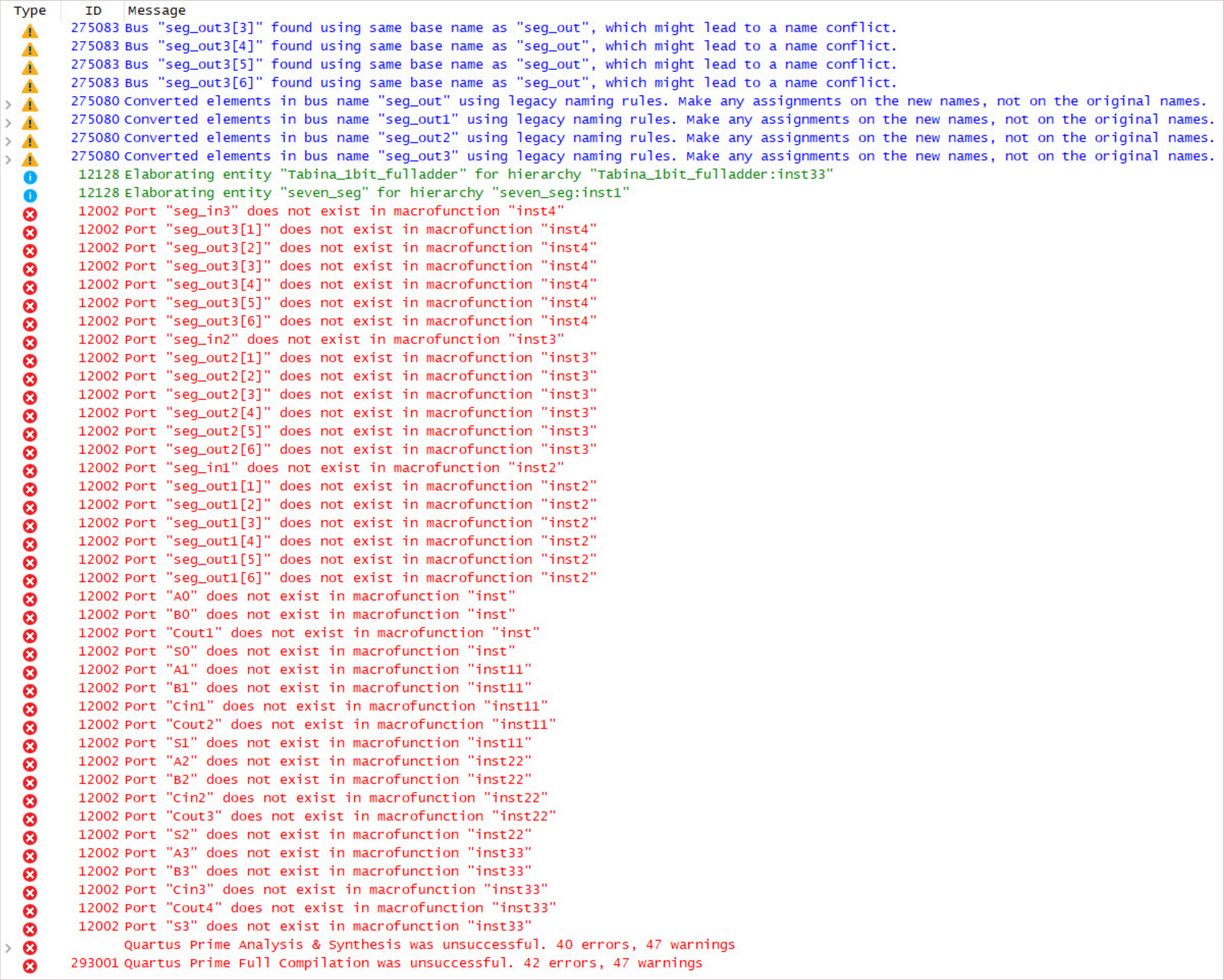
Figure 13: 4-bit full-adder circuit with seven-segment display error messages [Attempt 1]

The expected result to be the circuit to receive an input of two binary numbers of four digits and produce the output of the sum of the two digits as well as a carry bit. However, this result was not obtained in the simulation. The functional simulation was unable to run due to errors produced, as shown in Figures 14 and 15 below.



```
Type   ID     Message
       12128 Elaborating entity "seven_seg" for hierarchy "adder_disp:inst3|seven_seg:inst1"
       12002 Port "seg_out[0]" does not exist in macrofunction "inst2"
       12002 Port "seg_out[1]" does not exist in macrofunction "inst2"
       12002 Port "seg_out[2]" does not exist in macrofunction "inst2"
       12002 Port "seg_out[3]" does not exist in macrofunction "inst2"
       12002 Port "seg_out[4]" does not exist in macrofunction "inst2"
       12002 Port "seg_out[5]" does not exist in macrofunction "inst2"
       12002 Port "seg_out[6]" does not exist in macrofunction "inst2"
       12002 Port "seg_out[0]" does not exist in macrofunction "inst1"
       12002 Port "seg_out[1]" does not exist in macrofunction "inst1"
       12002 Port "seg_out[2]" does not exist in macrofunction "inst1"
       12002 Port "seg_out[3]" does not exist in macrofunction "inst1"
       12002 Port "seg_out[4]" does not exist in macrofunction "inst1"
       12002 Port "seg_out[5]" does not exist in macrofunction "inst1"
       12002 Port "seg_out[6]" does not exist in macrofunction "inst1"
       12002 Port "seg_out[0]" does not exist in macrofunction "inst"
       12002 Port "seg_out[1]" does not exist in macrofunction "inst"
       12002 Port "seg_out[2]" does not exist in macrofunction "inst"
       12002 Port "seg_out[3]" does not exist in macrofunction "inst"
       12002 Port "seg_out[4]" does not exist in macrofunction "inst"
       12002 Port "seg_out[5]" does not exist in macrofunction "inst"
       12002 Port "seg_out[6]" does not exist in macrofunction "inst"
       12002 Port "seg_out[0]" does not exist in macrofunction "inst3"
       12002 Port "seg_out[1]" does not exist in macrofunction "inst3"
       12002 Port "seg_out[2]" does not exist in macrofunction "inst3"
       12002 Port "seg_out[3]" does not exist in macrofunction "inst3"
       12002 Port "seg_out[4]" does not exist in macrofunction "inst3"
       12002 Port "seg_out[5]" does not exist in macrofunction "inst3"
       12002 Port "seg_out[6]" does not exist in macrofunction "inst3"
>            Quartus Prime Analysis & Synthesis was unsuccessful. 28 errors, 1 warning
      293001 Quartus Prime Full Compilation was unsuccessful. 30 errors, 1 warning
```

Figure 14: 4-bit full-adder circuit with seven-segment display [Attempt 2]



```
Type    ID    Message
      275083 Bus "seg_out3[3]" found using same base name as "seg_out", which might lead to a name conflict.
      275083 Bus "seg_out3[4]" found using same base name as "seg_out", which might lead to a name conflict.
      275083 Bus "seg_out3[5]" found using same base name as "seg_out", which might lead to a name conflict.
      275083 Bus "seg_out3[6]" found using same base name as "seg_out", which might lead to a name conflict.
>     275080 Converted elements in bus name "seg_out" using legacy naming rules. Make any assignments on the new names, not on the original names.
>     275080 Converted elements in bus name "seg_out1" using legacy naming rules. Make any assignments on the new names, not on the original names.
>     275080 Converted elements in bus name "seg_out2" using legacy naming rules. Make any assignments on the new names, not on the original names.
>     275080 Converted elements in bus name "seg_out3" using legacy naming rules. Make any assignments on the new names, not on the original names.
       12128 Elaborating entity "Tabina_1bit_fulladder" for hierarchy "Tabina_1bit_fulladder:inst33"
       12128 Elaborating entity "seven_seg" for hierarchy "seven_seg:inst1"
       12002 Port "seg_in3" does not exist in macrofunction "inst4"
       12002 Port "seg_out3[1]" does not exist in macrofunction "inst4"
       12002 Port "seg_out3[2]" does not exist in macrofunction "inst4"
       12002 Port "seg_out3[3]" does not exist in macrofunction "inst4"
       12002 Port "seg_out3[4]" does not exist in macrofunction "inst4"
       12002 Port "seg_out3[5]" does not exist in macrofunction "inst4"
       12002 Port "seg_out3[6]" does not exist in macrofunction "inst4"
       12002 Port "seg_in2" does not exist in macrofunction "inst3"
       12002 Port "seg_out2[1]" does not exist in macrofunction "inst3"
       12002 Port "seg_out2[2]" does not exist in macrofunction "inst3"
       12002 Port "seg_out2[3]" does not exist in macrofunction "inst3"
       12002 Port "seg_out2[4]" does not exist in macrofunction "inst3"
       12002 Port "seg_out2[5]" does not exist in macrofunction "inst3"
       12002 Port "seg_out2[6]" does not exist in macrofunction "inst3"
       12002 Port "seg_in1" does not exist in macrofunction "inst2"
       12002 Port "seg_out1[1]" does not exist in macrofunction "inst2"
       12002 Port "seg_out1[2]" does not exist in macrofunction "inst2"
       12002 Port "seg_out1[3]" does not exist in macrofunction "inst2"
       12002 Port "seg_out1[4]" does not exist in macrofunction "inst2"
       12002 Port "seg_out1[5]" does not exist in macrofunction "inst2"
       12002 Port "seg_out1[6]" does not exist in macrofunction "inst2"
       12002 Port "A0" does not exist in macrofunction "inst"
       12002 Port "B0" does not exist in macrofunction "inst"
       12002 Port "Cout1" does not exist in macrofunction "inst"
       12002 Port "S0" does not exist in macrofunction "inst"
       12002 Port "A1" does not exist in macrofunction "inst11"
       12002 Port "B1" does not exist in macrofunction "inst11"
       12002 Port "Cin1" does not exist in macrofunction "inst11"
       12002 Port "Cout2" does not exist in macrofunction "inst11"
       12002 Port "S1" does not exist in macrofunction "inst11"
       12002 Port "A2" does not exist in macrofunction "inst22"
       12002 Port "B2" does not exist in macrofunction "inst22"
       12002 Port "Cin2" does not exist in macrofunction "inst22"
       12002 Port "Cout3" does not exist in macrofunction "inst22"
       12002 Port "S2" does not exist in macrofunction "inst22"
       12002 Port "A3" does not exist in macrofunction "inst33"
       12002 Port "B3" does not exist in macrofunction "inst33"
       12002 Port "Cin3" does not exist in macrofunction "inst33"
       12002 Port "Cout4" does not exist in macrofunction "inst33"
       12002 Port "S3" does not exist in macrofunction "inst33"
>            Quartus Prime Analysis & Synthesis was unsuccessful. 40 errors, 47 warnings
      293001 Quartus Prime Full Compilation was unsuccessful. 42 errors, 47 warnings
```

Figure 15: 4-bit full-adder circuit with seven-segment display error messages [Attempt 2]

In both cases, the same types of errors were obtained in both attempts that concern the input and output ports, ‘*Error 12002 Port X does not exist in the macro function Y*’. Multiple trials and connection attempts were tried; however, the same result was obtained.

E. DISCUSSION

I. *Schematic Design Entry (Simulation and Board Testing)*

(1) Light circuit schematic design entry

The results obtained in the functional simulation and the board testing were the expected results. The experiment was carried out successfully. The light comes on when only one of the two inputs are high or when only either of the switches are ‘on’. When both switches are either off or on, the light is in the off state. The behaviour of the circuit is equivalent to that of an XOR gate.

(2) Design of 1-bit full adder

The results obtained for this circuit was expected for both the functional simulation and the board testing. There are two inputs, a carry input and a single sum output and a carry output. The truth table verifies the working of the circuit.

II. *Schematic Design (Block Entry – Simulation)*

For the first part of the block entry design of the design of the 1-bit full adder connected to a seven-segment display, the expected result was obtained for the functional simulation. The second part of designing a 4-bit full adder was however was unsuccessful. The errors obtained from the multiple attempts were the same and the issue was not resolved. However, if the functional simulation was successful, the circuit would display the sum of two 4-bit binary values and display the output on a seven-segment display along with a carry bit output.

III. *Answers to questions*

(1) Limitations of schematic entry to complex circuits

Very complex schematic design circuits tend to become unreadable and difficult to follow. Often these designs are printed on paper and a very complex circuit will be difficult to comprehend in such a scenario.

(2) Best conditions to use schematic and Verilog entry

Schematic designs would be best used for simple circuits due to quicker design implementation as well as being very clear and any connection errors can be seen straight away. Verilog design entry would be best suited for more complex circuits where the circuitry would be too ‘large’ to detect connection errors and instead the behavioural approach would give a better representation for implementation.

(3) Practicality of board testing a higher bit full-adder using switches and LEDs

It is not practical to test for a higher-bit adder using the LEDs and switches as there is a limited number of them on the board. For example, it would not be possible to implement an 8-bit full adder on the board. It would require 8 switches for each addend, requiring a total of 16 switches (not including the required carry in input) whereas the board only has 10 available switches. An alternate method of testing higher-bit adders would be to test the adder circuits on a breadboard using IC chips instead of an FGPA board.

(4) *Comparison of the 'light' circuit with Verilog and schematic entry*

For the light circuit, the schematic design entry is the preferred method rather than the Verilog entry design. The circuit is very simple so the schematic design entry is a very straightforward way to implement the circuit. It is much quicker in terms of design entry time.

F. CONCLUSION

Most of the objectives of the experiment were fulfilled. All of the results were obtained successfully except for the 4-bit full adder, which was not able to be compiled. There were no discrepancies between the simulation and the board testing results for the successful experiments.

G. REFERENCES

- DE1-Soc User Manual
- Quartus Prime Introduction Using Schematic Designs User Guide
- Quartus Prime Introduction Using Verilog Designs User Guide