**Lab 5 Report**

**Name: Dhruv Sandesara**

**UT EID: djs3967**

**Section:** 15295

**Checklist:**

**Part 1 –**

1. Design file (.v) for the Ripple Carry Adder
2. `timescale 1ns / 1ps
3. //////////////////////////////////////////////////////////////////////////////////
4. // Company:
5. // Engineer:
6. //
7. // Create Date: 04/06/2018 02:57:29 PM
8. // Design Name:
9. // Module Name: RCA\_4bits
10. // Project Name:
11. // Target Devices:
12. // Tool Versions:
13. // Description:
14. //
15. // Dependencies:
16. //
17. // Revision:
18. // Revision 0.01 - File Created
19. // Additional Comments:
20. //
21. //////////////////////////////////////////////////////////////////////////////////
22. module RCA\_4bits(
23. input clk,
24. input enable,
25. input [3:0] A,
26. input [3:0] B,
27. input Cin,
28. output [4:0]Q
29. );
31. wire cout0, cout1, cout2;
32. wire [4:0] Data;
33. full\_adder f0 (.A(A[0]), .B(B[0]), .Cin(Cin), .S(Data[0]), .Cout(cout0));
34. full\_adder f1 (.A(A[1]), .B(B[1]), .Cin(cout0), .S(Data[1]), .Cout(cout1));
35. full\_adder f2 (.A(A[2]), .B(B[2]), .Cin(cout1), .S(Data[2]), .Cout(cout2));
36. full\_adder f3 (.A(A[3]), .B(B[3]), .Cin(cout2), .S(Data[3]), .Cout(Data[4]));
38. register\_logic r0 (.clk(clk), .enable(enable), .Data(Data), .Q(Q));


42. Endmodule
43. `timescale 1ns / 1ps
44. //////////////////////////////////////////////////////////////////////////////////
45. // Company:
46. // Engineer:
47. //
48. // Create Date: 04/06/2018 02:59:43 PM
49. // Design Name:
50. // Module Name: full\_adder
51. // Project Name:
52. // Target Devices:
53. // Tool Versions:
54. // Description:
55. //
56. // Dependencies:
57. //
58. // Revision:
59. // Revision 0.01 - File Created
60. // Additional Comments:
61. //
62. //////////////////////////////////////////////////////////////////////////////////
63. module full\_adder(
64. input A,B,Cin,
65. output reg S,Cout
66. );
68. reg [1:0]out;
69. always@(\*)begin
70. out = A+B+Cin;
71. assign S= out[0];
72. assign Cout= out[1];
74. //{Cout,S}= A+B+Cin;


78. end
80. Endmodule
81. `timescale 1ns / 1ps
82. //////////////////////////////////////////////////////////////////////////////////
83. // Company:
84. // Engineer:
85. //
86. // Create Date: 04/06/2018 03:01:08 PM
87. // Design Name:
88. // Module Name: register\_logic
89. // Project Name:
90. // Target Devices:
91. // Tool Versions:
92. // Description:
93. //
94. // Dependencies:
95. //
96. // Revision:
97. // Revision 0.01 - File Created
98. // Additional Comments:
99. //
100. //////////////////////////////////////////////////////////////////////////////////
101. module register\_logic(
102. input clk,
103. input enable,
104. input [4:0] Data,
105. output reg [4:0] Q
106. );
108. initial begin
109. Q <= 0;
110. end
111. always @(posedge clk)
112. if(enable)
113. Q <= Data;

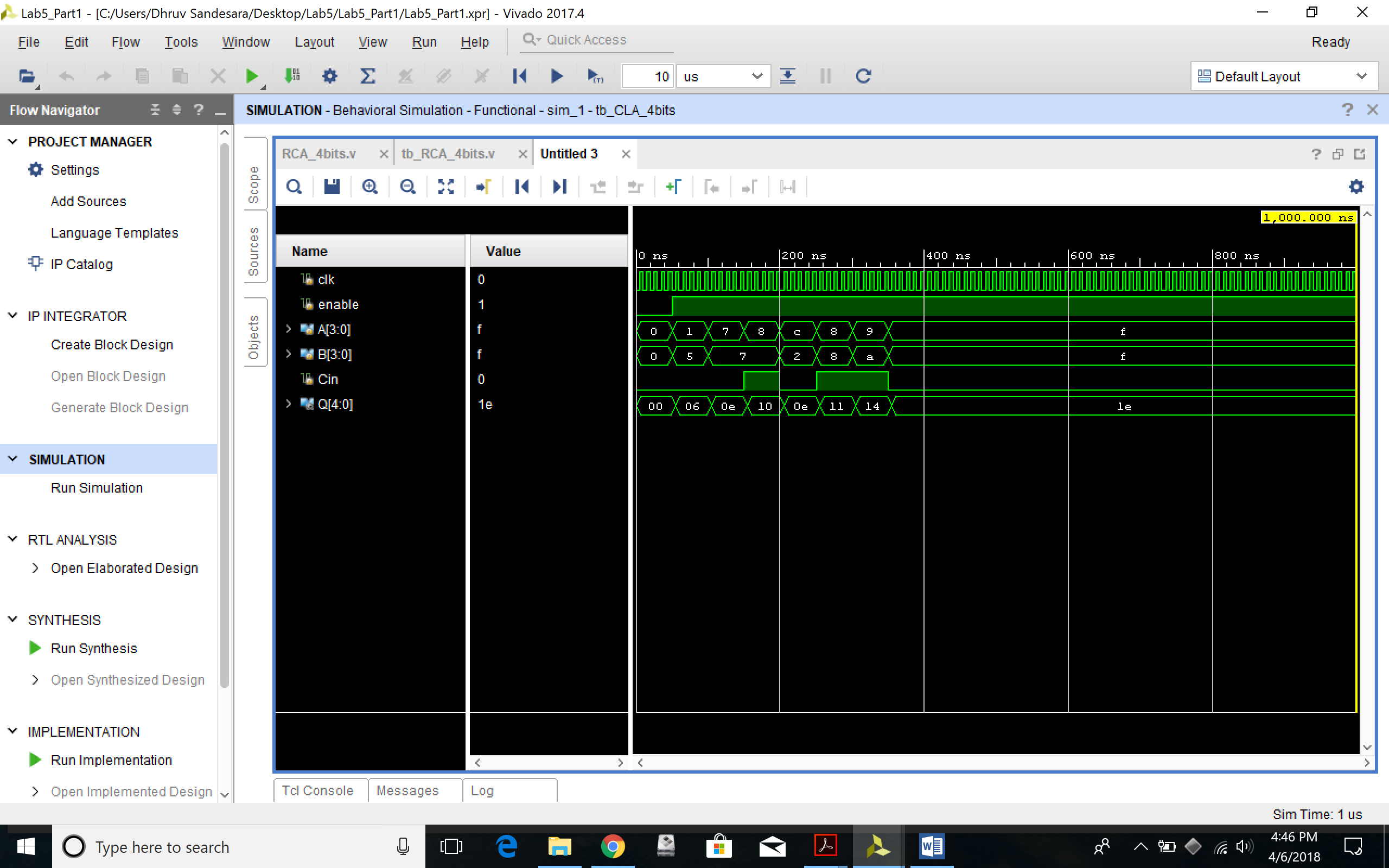
116. Endmodule
117. Test-bench
118. `timescale 1ns / 1ps
119. //////////////////////////////////////////////////////////////////////////////////
120. // Company:
121. // Engineer:
122. //
123. // Create Date: 04/06/2018 03:24:05 PM
124. // Design Name:
125. // Module Name: tb\_RCA\_4bits
126. // Project Name:
127. // Target Devices:
128. // Tool Versions:
129. // Description:
130. //
131. // Dependencies:
132. //
133. // Revision:
134. // Revision 0.01 - File Created
135. // Additional Comments:
136. //
137. //////////////////////////////////////////////////////////////////////////////////
138. module tb\_RCA\_4bits;
139. reg clk;
140. reg enable;
141. reg [3:0] A;
142. reg [3:0] B;
143. reg Cin;
144. wire [4:0] Q;

147. RCA\_4bits uut (
148. .clk(clk),
149. .enable(enable),
150. .A(A),
151. .B(B),
152. .Cin(Cin),
153. .Q(Q)
154. );
155. initial begin
157. clk = 0;
158. enable = 0;
159. A = 4'b0000;
160. B = 4'b0000;
161. Cin = 0;
163. #50;
164. enable = 1;
165. A = 4'b0001;
166. B = 4'b0101;
167. Cin = 0;
169. #50;
170. A = 4'b0111;
171. B = 4'b0111;
172. Cin = 0;
174. #50;
175. A = 4'b1000;
176. B = 4'b0111;
177. Cin = 1;
179. #50;
180. A = 4'b1100;
181. B = 4'b0010;
182. Cin = 0;
184. #50;
185. A = 4'b1000;
186. B = 4'b1000;
187. Cin = 1;
189. #50;
190. A = 4'b1001;
191. B = 4'b1010;
192. Cin = 1;
194. #50;
195. A = 4'b1111;
196. B = 4'b1111;
197. Cin = 0;
199. end
201. always
202. #5 clk = ~clk;
204. endmodule
205. Complete Table 1 from the simulation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A[3:0] | B[3:0] | Cin | Sum[3:0] | Cout |
| 0001 | 0101 | 0 | 6 | 0 |
| 0111 | 0111 | 0 | E | 0 |
| 1000 | 0111 | 1 | 0 | 1 |
| 1100 | 0100 | 0 | E | 0 |
| 1000 | 1000 | 1 | 1 | 1 |
| 1001 | 1010 | 1 | 4 | 1 |
| 1111 | 1111 | 0 | E | 1 |

**Table 1.** Testcases for Ripple Carry Adder Verification

1. Constraints File (Just the uncommented portion)
2. ## Clock signal
3. set\_property PACKAGE\_PIN W5 [get\_ports clk]
4. set\_property IOSTANDARD LVCMOS33 [get\_ports clk]
5. create\_clock -add -name sys\_clk\_pin -period 10.00 -waveform {0 5} [get\_ports clk]
7. ## Switches
8. set\_property PACKAGE\_PIN V17 [get\_ports {A[0]}]
9. set\_property IOSTANDARD LVCMOS33 [get\_ports {A[0]}]
10. set\_property PACKAGE\_PIN V16 [get\_ports {A[1]}]
11. set\_property IOSTANDARD LVCMOS33 [get\_ports {A[1]}]
12. set\_property PACKAGE\_PIN W16 [get\_ports {A[2]}]
13. set\_property IOSTANDARD LVCMOS33 [get\_ports {A[2]}]
14. set\_property PACKAGE\_PIN W17 [get\_ports {A[3]}]
15. set\_property IOSTANDARD LVCMOS33 [get\_ports {A[3]}]
16. set\_property PACKAGE\_PIN W15 [get\_ports {B[0]}]
17. set\_property IOSTANDARD LVCMOS33 [get\_ports {B[0]}]
18. set\_property PACKAGE\_PIN V15 [get\_ports {B[1]}]
19. set\_property IOSTANDARD LVCMOS33 [get\_ports {B[1]}]
20. set\_property PACKAGE\_PIN W14 [get\_ports {B[2]}]
21. set\_property IOSTANDARD LVCMOS33 [get\_ports {B[2]}]
22. set\_property PACKAGE\_PIN W13 [get\_ports {B[3]}]
23. set\_property IOSTANDARD LVCMOS33 [get\_ports {B[3]}]
24. set\_property PACKAGE\_PIN V2 [get\_ports {Cin}]
25. set\_property IOSTANDARD LVCMOS33 [get\_ports {Cin}]
26. set\_property PACKAGE\_PIN U16 [get\_ports {Q[0]}]
27. set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[0]}]
28. set\_property PACKAGE\_PIN E19 [get\_ports {Q[1]}]
29. set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[1]}]
30. set\_property PACKAGE\_PIN U19 [get\_ports {Q[2]}]
31. set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[2]}]
32. set\_property PACKAGE\_PIN V19 [get\_ports {Q[3]}]
33. set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[3]}]
34. set\_property PACKAGE\_PIN W18 [get\_ports {Q[4]}]
35. set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[4]}]
36. set\_property PACKAGE\_PIN U18 [get\_ports enable]
37. set\_property IOSTANDARD LVCMOS33 [get\_ports enable]
38. Simulation waveform for the above test-cases



**Part 2 –**

1. All the equations for Ci’sand Si’s
2. assign P=A^B;
3. assign G=A&B;
5. assign C[0]= Cin;
6. assign C[1]= G[0]| P[0]&Cin;
7. assign C[2]= G[1]| P[1]&G[0]| P[1]&P[0]&Cin ;
8. assign C[3]= G[2]| P[2]&G[1]| P[2]&P[1]&G[0]| P[2]&P[1]&P[0]&Cin ;
9. assign C[4]= G[3]| P[3]&G[2]| P[3]&P[2]&G[1]| P[3]&P[2]&P[1]&G[0]| P[3]&P[2]&P[1]&P[0]&Cin;
11. assign Data[0]= P[0]^C[0];
12. assign Data[1]= P[1]^C[1];
13. assign Data[2]= P[2]^C[2];
14. assign Data[3]= P[3]^C[3];
15. assign Data[4]= C[4];
16. Design files (.v) for the Carry Lookahead Adder and Register Logic

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 04/06/2018 03:51:09 PM

// Design Name:

// Module Name: CLA\_4bits

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module CLA\_4bits(

input clk,

input enable,

input [3:0] A,

input [3:0] B,

input Cin,

output [4:0]Q

);

wire [4:0] Data;

wire [3:0] P;

wire [3:0] G;

wire [4:0] C;

assign P=A^B;

assign G=A&B;

assign C[0]= Cin;

assign C[1]= G[0]| P[0]&Cin;

assign C[2]= G[1]| P[1]&G[0]| P[1]&P[0]&Cin ;

assign C[3]= G[2]| P[2]&G[1]| P[2]&P[1]&G[0]| P[2]&P[1]&P[0]&Cin ;

assign C[4]= G[3]| P[3]&G[2]| P[3]&P[2]&G[1]| P[3]&P[2]&P[1]&G[0]| P[3]&P[2]&P[1]&P[0]&Cin;

assign Data[0]= P[0]^C[0];

assign Data[1]= P[1]^C[1];

assign Data[2]= P[2]^C[2];

assign Data[3]= P[3]^C[3];

assign Data[4]= C[4];

register\_logic r0 (.clk(clk), .enable(enable), .Data(Data), .Q(Q));

endmodule

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 04/06/2018 03:52:58 PM

// Design Name:

// Module Name: register\_logic

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module register\_logic(

input clk,

input enable,

input [4:0] Data,

output reg [4:0] Q

);

initial begin

Q <= 0;

end

always @(posedge clk)

if(enable)

Q <= Data;

Endmodule

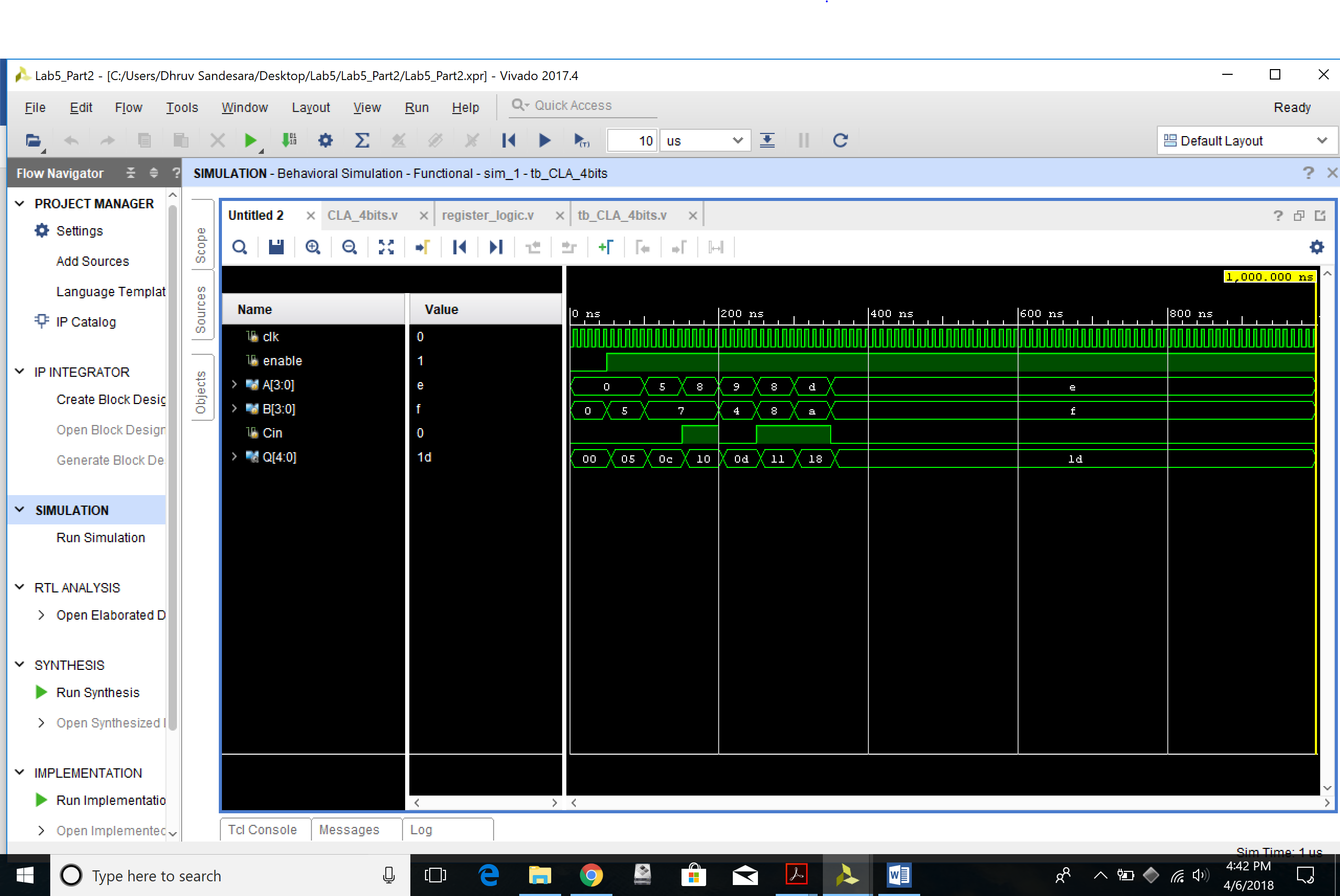
1. Test-bench
2. `timescale 1ns / 1ps
3. //////////////////////////////////////////////////////////////////////////////////
4. // Company:
5. // Engineer:
6. //
7. // Create Date: 04/06/2018 03:55:54 PM
8. // Design Name:
9. // Module Name: tb\_CLA\_4bits
10. // Project Name:
11. // Target Devices:
12. // Tool Versions:
13. // Description:
14. //
15. // Dependencies:
16. //
17. // Revision:
18. // Revision 0.01 - File Created
19. // Additional Comments:
20. //
21. //////////////////////////////////////////////////////////////////////////////////
22. module tb\_CLA\_4bits();
23. reg clk;
24. reg enable;
25. reg [3:0] A;
26. reg [3:0] B;
27. reg Cin;
28. wire [4:0] Q;

31. CLA\_4bits uut (
32. .clk(clk),
33. .enable(enable),
34. .A(A),
35. .B(B),
36. .Cin(Cin),
37. .Q(Q)
38. );
39. initial begin
40. clk = 0;
41. enable = 0;
42. A = 4'b0000;
43. B = 4'b0000;
44. Cin = 0;
45. #50;
46. enable = 1;
47. A = 4'b0000;
48. B = 4'b0101;
49. Cin = 0;
50. #50;
51. A = 4'b0101;
52. B = 4'b0111;
53. Cin = 0;
54. #50;
55. A = 4'b1000;
56. B = 4'b0111;
57. Cin = 1;
58. #50;
59. A = 4'b1001;
60. B = 4'b0100;
61. Cin = 0;
62. #50;
63. A = 4'b1000;
64. B = 4'b1000;
65. Cin = 1;
66. #50;
67. A = 4'b1101;
68. B = 4'b1010;
69. Cin = 1;
70. #50;
71. A = 4'b1110;
72. B = 4'b1111;
73. Cin = 0;
74. end
75. always
76. #5 clk = ~clk;
77. Endmodule
78. Complete Table 2 from the simulation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A[3:0] | B[3:0] | Cin | Sum[3:0] | Cout |
| 0000 | 0101 | 0 | 5 | 0 |
| 0101 | 0111 | 0 | C | 0 |
| 1000 | 0111 | 1 | 0 | 1 |
| 1001 | 0100 | 0 | D | 0 |
| 1000 | 1000 | 1 | 1 | 1 |
| 1101 | 1010 | 1 | 8 | 1 |
| 1110 | 1111 | 0 | D | 1 |

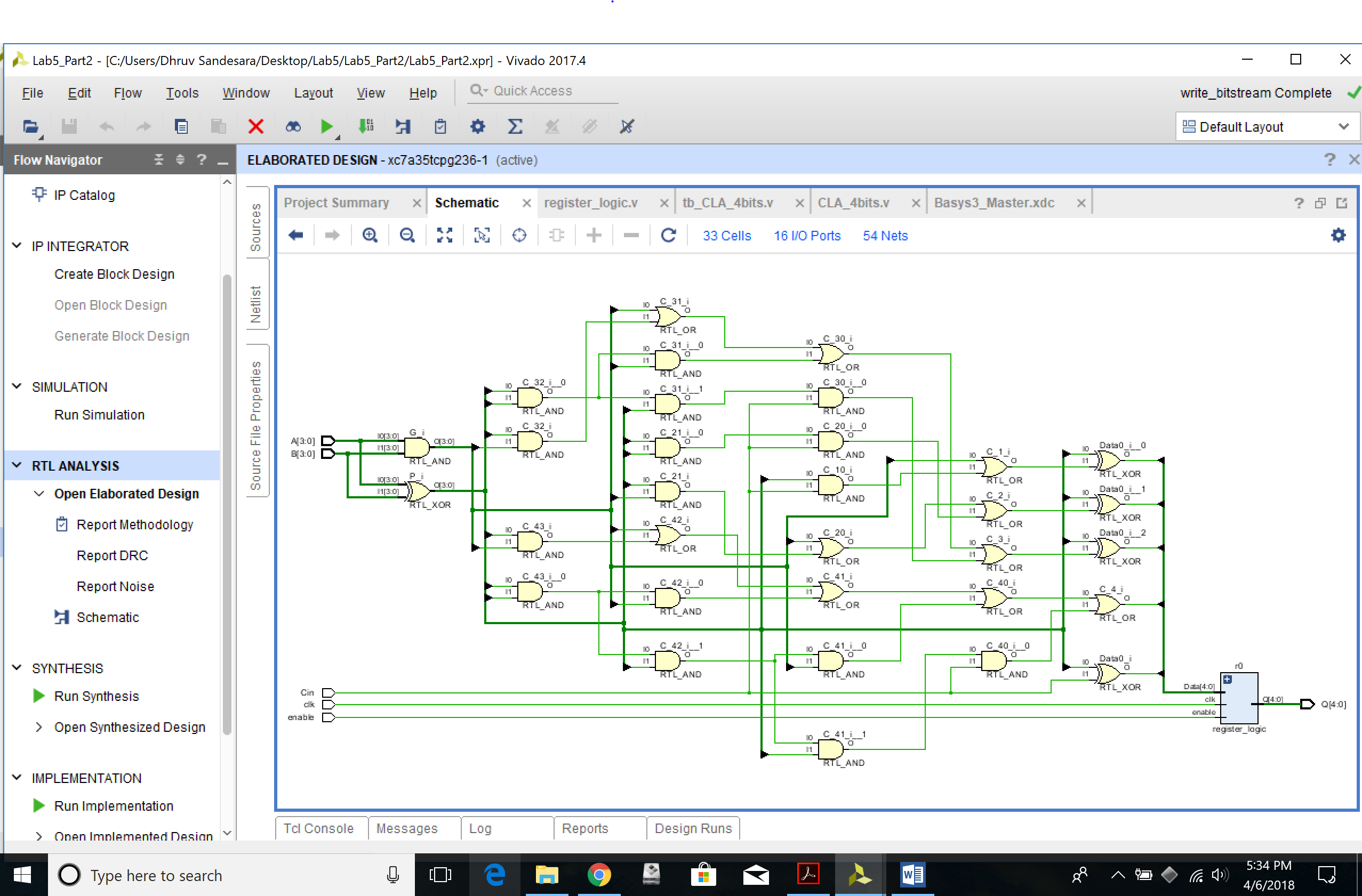
**Table 2.** Testcases for Carry Lookahead Adder Verification

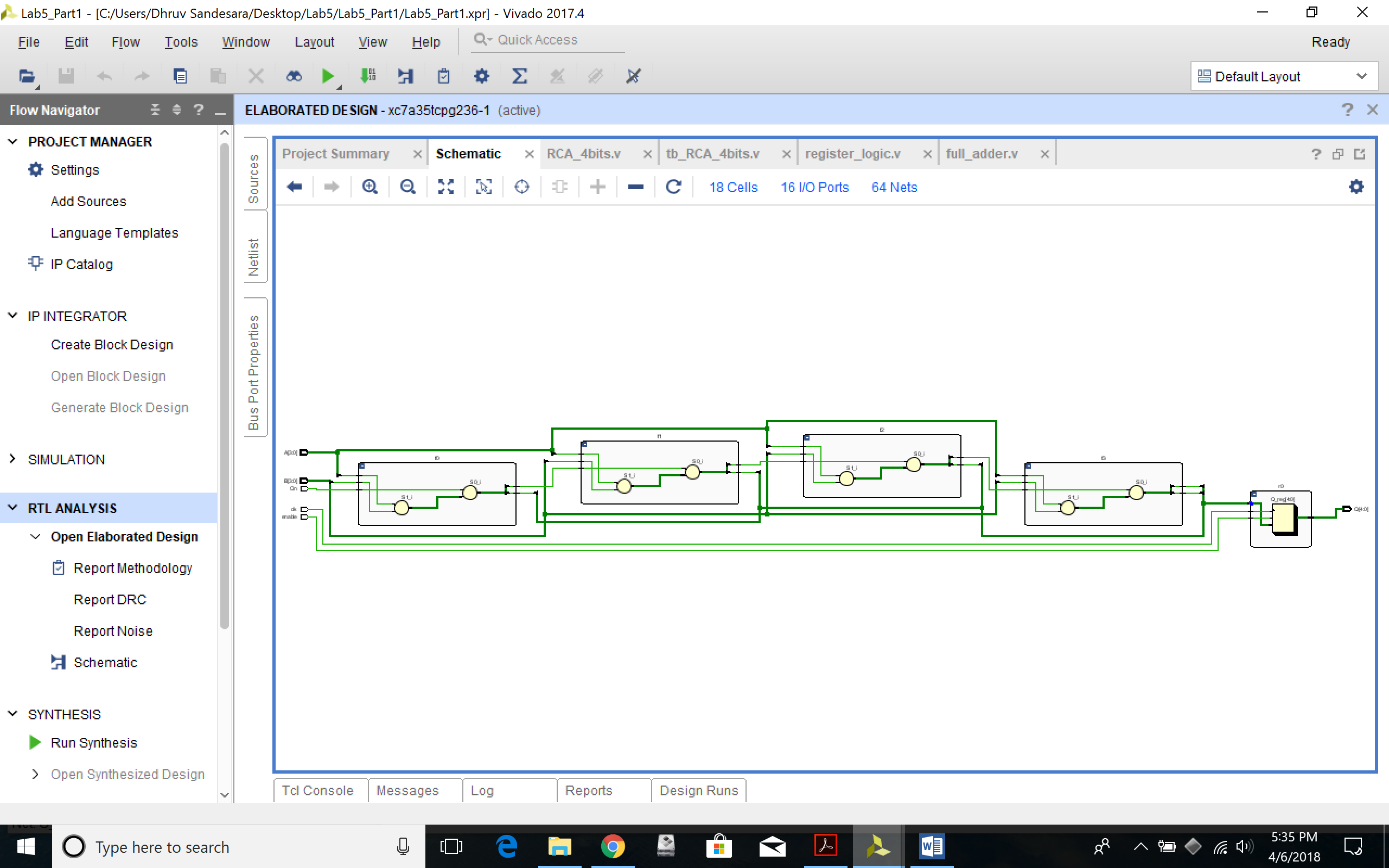
1. Constraints File (Just the uncommented portion)
2. ## Clock signal
3. set\_property PACKAGE\_PIN W5 [get\_ports clk]
4. set\_property IOSTANDARD LVCMOS33 [get\_ports clk]
5. create\_clock -add -name sys\_clk\_pin -period 10.00 -waveform {0 5} [get\_ports clk]
7. ## Switches
8. set\_property PACKAGE\_PIN V17 [get\_ports {A[0]}]
9. set\_property IOSTANDARD LVCMOS33 [get\_ports {A[0]}]
10. set\_property PACKAGE\_PIN V16 [get\_ports {A[1]}]
11. set\_property IOSTANDARD LVCMOS33 [get\_ports {A[1]}]
12. set\_property PACKAGE\_PIN W16 [get\_ports {A[2]}]
13. set\_property IOSTANDARD LVCMOS33 [get\_ports {A[2]}]
14. set\_property PACKAGE\_PIN W17 [get\_ports {A[3]}]
15. set\_property IOSTANDARD LVCMOS33 [get\_ports {A[3]}]
16. set\_property PACKAGE\_PIN W15 [get\_ports {B[0]}]
17. set\_property IOSTANDARD LVCMOS33 [get\_ports {B[0]}]
18. set\_property PACKAGE\_PIN V15 [get\_ports {B[1]}]
19. set\_property IOSTANDARD LVCMOS33 [get\_ports {B[1]}]
20. set\_property PACKAGE\_PIN W14 [get\_ports {B[2]}]
21. set\_property IOSTANDARD LVCMOS33 [get\_ports {B[2]}]
22. set\_property PACKAGE\_PIN W13 [get\_ports {B[3]}]
23. set\_property IOSTANDARD LVCMOS33 [get\_ports {B[3]}]
24. set\_property PACKAGE\_PIN V2 [get\_ports {Cin}]
25. set\_property IOSTANDARD LVCMOS33 [get\_ports {Cin}]
26. set\_property PACKAGE\_PIN U16 [get\_ports {Q[0]}]
27. set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[0]}]
28. set\_property PACKAGE\_PIN E19 [get\_ports {Q[1]}]
29. set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[1]}]
30. set\_property PACKAGE\_PIN U19 [get\_ports {Q[2]}]
31. set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[2]}]
32. set\_property PACKAGE\_PIN V19 [get\_ports {Q[3]}]
33. set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[3]}]
34. set\_property PACKAGE\_PIN W18 [get\_ports {Q[4]}]
35. set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[4]}]
36. set\_property PACKAGE\_PIN U18 [get\_ports enable]
37. set\_property IOSTANDARD LVCMOS33 [get\_ports enable]
38. Simulation waveform for the above test-cases



**Part 3 –**

1. Screenshots of the gate-level schematics for both the adder techniques





1. Delay and area for both the adder techniques showing all the work
2. Gate Delay (ns) Area
3. XOR 3 6
4. AND 3 4
5. OR 2 4
6. Table 3. Sample Delay and Area Values for Various

Part 2 :

18 And Gates

5 XOR Gates

10 OR Gates

**Area is 18\*4 + 5\*6 + 10\*4= 142**

**Max Path is 6 And&XOR gates deep which is 6\*3 is 18ns**

Part 1:

1 Full Adder has : 2\* XOR, 2\* And, and 1 OR;

MaX Delay per Adder is 1 XOR, 1AND, And 1 OR.

There fore

AREa is 2\*6+2\*4+4= 24

Delay is 3+3+2= 8

**As we use 4 adders we get**

**Area is 4\*24=96**

**Delay is 4\*8= 32ns;**

1. Brief conclusion regarding the pros and cons of each of the techniques

Carry Ahead Looker: Pros is that it is fast due to minimal gate delay. Cons is that there are way too many gates used

Carry Ripple Ahead: Pros is that it uses less gates. Cons is that there is a O(n) gate delay as the MSB has to wait on the Carry bits from the lesser significat bits.

***Note*** *–> The Verilog codes and the uncommented portions of the constraint files should be copied in your lab report and the* ***actual Verilog (.v), Constraint (.xdc) files and Bitstream (.bit) files*** *need to be zipped and submitted as well on Canvas. You are not allowed to change your codes after final submission as the TAs may download the submitted codes or bitstream files from Canvas during checkouts. For the truth Table, K-maps minimizations and algebraic expressions, you are free to draw them on paper and then put the pictures in your lab report, but please make sure it is legible for the TAs to grade it properly.*