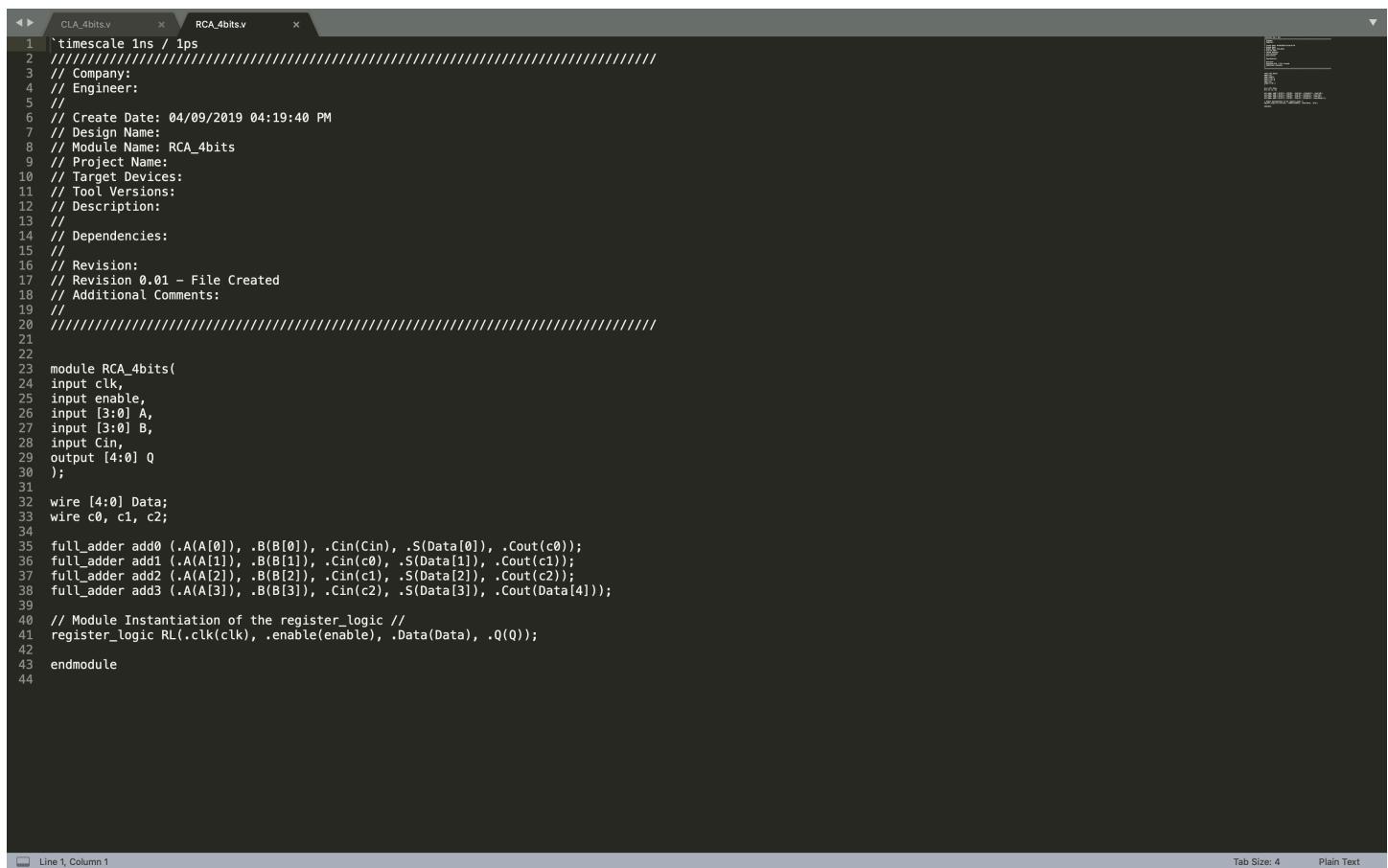


Arman Khondker
Eid: aak2464
Professor Pan

LAB 5 REPORT



The screenshot shows a text editor window with two tabs: 'CLA_4bits.v' and 'RCA_4bits.v'. The 'RCA_4bits.v' tab is active, displaying the following Verilog code:

```
1 //timescale 1ns / 1ps
2 ///////////////////////////////
3 // Company:
4 // Engineer:
5 //
6 // Create Date: 04/09/2019 04:19:40 PM
7 // Design Name:
8 // Module Name: RCA_4bits
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
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 );
30
31 wire [4:0] Data;
32 wire c0, c1, c2;
33
34 full_adder add0 (.A(A[0]), .B(B[0]), .Cin(Cin), .S(Data[0]), .Cout(c0));
35 full_adder add1 (.A(A[1]), .B(B[1]), .Cin(c0), .S(Data[1]), .Cout(c1));
36 full_adder add2 (.A(A[2]), .B(B[2]), .Cin(c1), .S(Data[2]), .Cout(c2));
37 full_adder add3 (.A(A[3]), .B(B[3]), .Cin(c2), .S(Data[3]), .Cout(Data[4]));
38
39 // Module Instantiation of the register_logic //
40 register_logic RL(.clk(clk), .enable(enable), .Data(Data), .Q(Q));
41
42 endmodule
43
44
```

The code defines a module 'RCA_4bits' with inputs 'clk', 'enable', 'A' (4-bit), 'B' (4-bit), 'Cin' (1-bit) and output 'Q' (5-bit). It uses four 'full_adder' blocks to perform the addition. The 'register_logic' instantiation is shown but commented out.

```
CLA_4bits.v      X  RCA_4bits.v      X  tb_CLA_4bits.v      X  tb_RCA_4bits.v      X
1 `timescale 1ns / 1ps
2 ///////////////////////////////////////////////////////////////////
3 /// Company:
4 /// Engineer:
5 ///
6 // Create Date: 04/09/2019 05:04:39 PM
7 // Design Name:
8 // Module Name: tb_RCA_4bits
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21
22 module tb_RCA_4bits;
23
24 reg clk;
25 reg enable;
26 reg [3:0] A;
27 reg [3:0] B;
28 reg Cin;
29 // Output //
30 wire [4:0] Q;
31
32 RCA_4bits uut (
33 .clk(clk),
34 .enable(enable),
35 .A(A),
36 .B(B),
37 .Cin(Cin),
38 .Q(Q)
39 );
40
41 initial
42 begin
43
44 clk = 0;
45 enable = 0;
46 A = 0;
47 B = 0;
48 Cin = 0;
49 #10;
50
51 enable = 1;
52 A[3:0] = 4'b0001;
53 B[3:0] = 4'b0101;
54 Cin = 0;
55 #10;
56
57 end
58
59 endmodule
```

```
CLA_4bits.v      x  RCA_4bits.v      x  tb_CLA_4bits.v      x  tb_RCA_4bits.v      x
63 #10;
64 enable = 0;
65 #10;
66
67 enable = 1;
68 A[3:0] = 4'b1000;
69 B[3:0] = 4'b0111;
70 Cin = 1;
71 #10;
72 enable = 0;
73 #10;
74
75 enable = 1;
76 A[3:0] = 4'b1100;
77 B[3:0] = 4'b0100;
78 Cin = 0;
79 #10;
80 enable = 0;
81 #10;
82
83
84
85 enable = 1;
86 A[3:0] = 4'b1000;
87 B[3:0] = 4'b1000;
88 Cin = 1;
89 #10;
90 enable = 0;
91 #10;
92
93 enable = 1;
94 A[3:0] = 4'b1001;
95 B[3:0] = 4'b1010;
96 Cin = 1;
97 #10;
98 enable = 0;
99 #10;
100
101 enable = 1;
102 A[3:0] = 4'b1111;
103 B[3:0] = 4'b1111;
104 Cin = 0;
105 #10;
106 enable = 0;
107 #10;
108 end
109 |     always
110 begin
111
112
113
114 #5;
115 clk = ~clk;
116 end
117 endmodule
```

A[3:0]	B[3:0]	Cin	Sum[3:0]	Cout
0001	0101	0	0110	0
0111	0111	0	1110	0
1000	0111	1	0000	1
1100	0100	0	0000	1
1000	1000	1	0001	1
1001	1010	1	0100	1
1111	1111	0	1110	1

Table 1. Testcases for Ripple Carry Adder Verification

```

1 ## This file is a general .xdc for the Basys3 rev B board
2 ## To use it in a project:
3 ## - uncomment the lines corresponding to used pins
4 ## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project
5
6 ## Clock signal
7 set_property PACKAGE_PIN W5 [get_ports clk]
8   set_property IOSTANDARD LVCMS033 [get_ports clk]
9   create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]
10
11 ## Switches
12 set_property PACKAGE_PIN V17 [get_ports {A[0]}]
13   set_property IOSTANDARD LVCMS033 [get_ports {A[0]}]
14   set_property PACKAGE_PIN V16 [get_ports {A[1]}]
15     set_property IOSTANDARD LVCMS033 [get_ports {A[1]}]
16   set_property PACKAGE_PIN V16 [get_ports {A[2]}]
17     set_property IOSTANDARD LVCMS033 [get_ports {A[2]}]
18   set_property PACKAGE_PIN W17 [get_ports {A[3]}]
19     set_property IOSTANDARD LVCMS033 [get_ports {A[3]}]
20   set_property PACKAGE_PIN W15 [get_ports {A[0]}]
21   set_property IOSTANDARD LVCMS033 [get_ports {B[0]}]
22   set_property PACKAGE_PIN V15 [get_ports {B[1]}]
23   set_property IOSTANDARD LVCMS033 [get_ports {B[1]}]
24   set_property PACKAGE_PIN W14 [get_ports {B[2]}]
25     set_property IOSTANDARD LVCMS033 [get_ports {B[2]}]
26   set_property PACKAGE_PIN W13 [get_ports {B[3]}]
27     set_property IOSTANDARD LVCMS033 [get_ports {B[3]}]
28   set_property PACKAGE_PIN V2 [get_ports {Cin}]
29   set_property IOSTANDARD LVCMS033 [get_ports {Cin}]
30   #set_property PACKAGE_PIN T3 [get_ports {sw[9]}]
31   #set_property IOSTANDARD LVCMS033 [get_ports {sw[9]}]
32   #set_property PACKAGE_PIN T2 [get_ports {sw[10]}]
33     #set_property IOSTANDARD LVCMS033 [get_ports {sw[10]}]
34   #set_property PACKAGE_PIN R3 [get_ports {sw[11]}]
35     #set_property IOSTANDARD LVCMS033 [get_ports {sw[11]}]
36   #set_property PACKAGE_PIN W2 [get_ports {sw[12]}]
37     #set_property IOSTANDARD LVCMS033 [get_ports {sw[12]}]
38   #set_property PACKAGE_PIN U1 [get_ports {sw[13]}]
39     #set_property IOSTANDARD LVCMS033 [get_ports {sw[13]}]
40   #set_property PACKAGE_PIN T1 [get_ports {sw[14]}]
41   #set_property IOSTANDARD LVCMS033 [get_ports {sw[14]}]
42   #set_property PACKAGE_PIN R2 [get_ports {sw[15]}]
43     #set_property IOSTANDARD LVCMS033 [get_ports {sw[15]}]
44
45 ## LEDs
46 set_property PACKAGE_PIN U16 [get_ports {O[0]}]
47   set_property IOSTANDARD LVCMS033 [get_ports {O[0]}]
48   set_property PACKAGE_PIN E19 [get_ports {O[1]}]
49     set_property IOSTANDARD LVCMS033 [get_ports {O[1]}]
50   set_property PACKAGE_PIN V19 [get_ports {O[2]}]
51     set_property IOSTANDARD LVCMS033 [get_ports {O[2]}]
52   set_property PACKAGE_PIN V19 [get_ports {O[3]}]
53     set_property IOSTANDARD LVCMS033 [get_ports {O[3]}]
54   set_property PACKAGE_PIN W18 [get_ports {O[4]}]
55     set_property IOSTANDARD LVCMS033 [get_ports {O[4]}]
56   set_property PACKAGE_PIN U15 [get_ports {led[5]}]
57     #set_property IOSTANDARD LVCMS033 [get_ports {led[5]}]
58   #set_property PACKAGE_PIN U14 [get_ports {led[6]}]
59     #set_property IOSTANDARD LVCMS033 [get_ports {led[6]}]
60   #set_property PACKAGE_PIN V14 [get_ports {led[7]}]
61     #set_property IOSTANDARD LVCMS033 [get_ports {led[7]}]
62   #set_property PACKAGE_PIN V11 [get_ports {led[8]}]
63     #set_property IOSTANDARD LVCMS033 [get_ports {led[8]}]
64   #set_property PACKAGE_PIN V11 [get_ports {led[9]}]
65     #set_property IOSTANDARD LVCMS033 [get_ports {led[9]}]
66   #set_property PACKAGE_PIN W3 [get_ports {led[10]}]
67     #set_property IOSTANDARD LVCMS033 [get_ports {led[10]}]
68   #set_property PACKAGE_PIN U3 [get_ports {led[11]}]
69     #set_property IOSTANDARD LVCMS033 [get_ports {led[11]}]
70   #set_property PACKAGE_PIN P3 [get_ports {led[12]}]
71   #set_property IOSTANDARD LVCMS033 [get_ports {led[12]}]

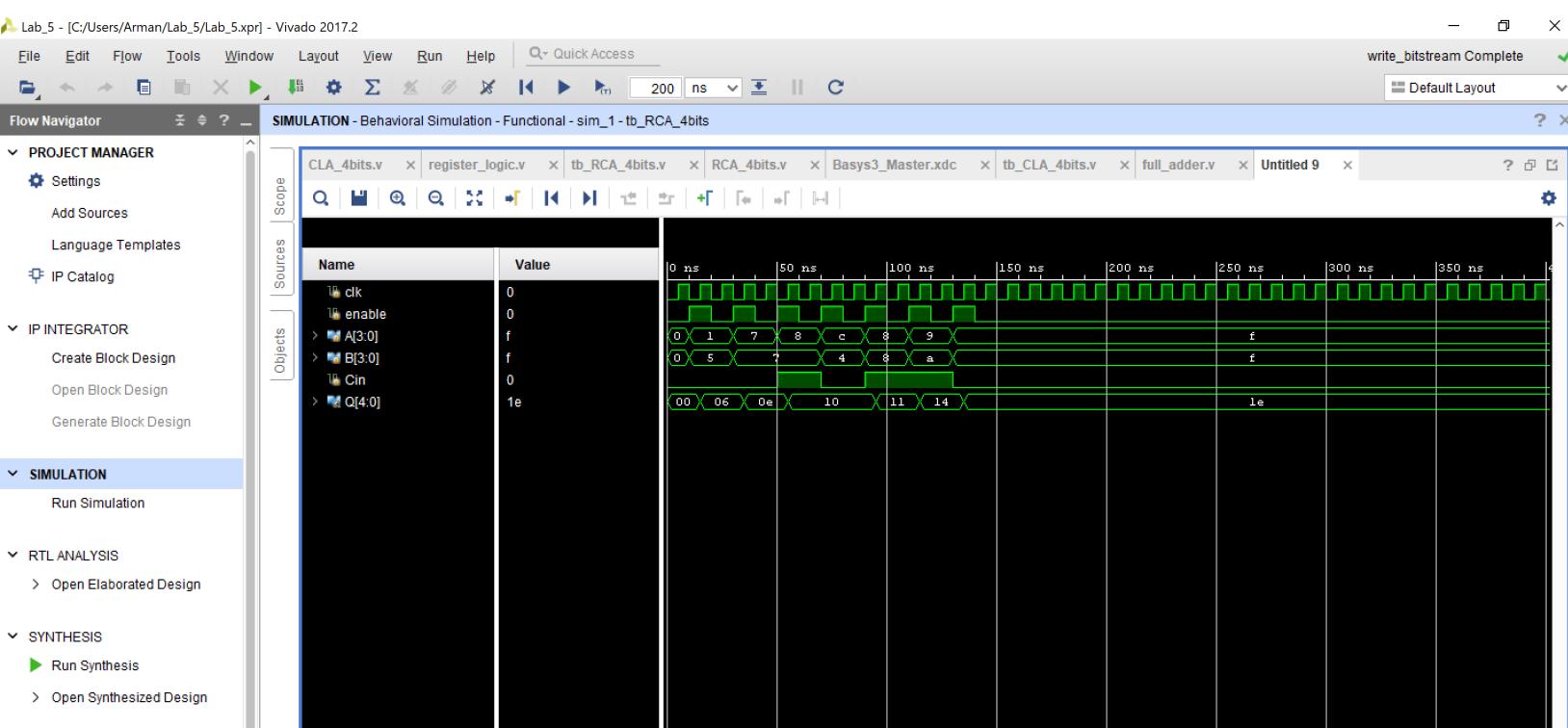
Line 1, Column 1          Tab Size: 4          Plain Text

```

```

81  ##set_property PACKAGE_PIN W0 [get_ports {seg[7]}]
82  |   #set_property IOSTANDARD LVCMOS33 [get_ports {seg[1]}]
83  |   #set_property PACKAGE_PIN U8 [get_ports {seg[2]}]
84  |   #set_property IOSTANDARD LVCMOS33 [get_ports {seg[2]}]
85  |   #set_property PACKAGE_PIN V8 [get_ports {seg[3]}]
86  |   #set_property IOSTANDARD LVCMOS33 [get_ports {seg[3]}]
87  |   #set_property PACKAGE_PIN U5 [get_ports {seg[4]}]
88  |   #set_property IOSTANDARD LVCMOS33 [get_ports {seg[4]}]
89  |   #set_property PACKAGE_PIN V5 [get_ports {seg[5]}]
90  |   #set_property IOSTANDARD LVCMOS33 [get_ports {seg[5]}]
91  |   #set_property PACKAGE_PIN U7 [get_ports {seg[6]}]
92  |   #set_property IOSTANDARD LVCMOS33 [get_ports {seg[6]}]
93  |
94  |   #set_property PACKAGE_PIN V7 [get_ports dp]
95  |   #set_property IOSTANDARD LVCMOS33 [get_ports dp]
96  |
97  |   #set_property PACKAGE_PIN U2 [get_ports {an[0]}]
98  |   #set_property IOSTANDARD LVCMOS33 [get_ports {an[0]}]
99  |
100 |   #set_property PACKAGE_PIN U4 [get_ports {an[1]}]
101 |   #set_property IOSTANDARD LVCMOS33 [get_ports {an[1]}]
102 |   #set_property PACKAGE_PIN V4 [get_ports {an[2]}]
103 |   #set_property IOSTANDARD LVCMOS33 [get_ports {an[2]}]
104 |   #set_property PACKAGE_PIN W4 [get_ports {an[3]}]
105 |   #set_property IOSTANDARD LVCMOS33 [get_ports {an[3]}]
106 |   #set_property PACKAGE_PIN W18 [get_ports enable]
107 |   #set_property IOSTANDARD LVCMOS33 [get_ports enable]
108 |
109 ##Buttons
110 set_property PACKAGE_PIN U18 [get_ports enable]
111 |   set_property IOSTANDARD LVCMOS33 [get_ports enable]
112 |   #set_property PACKAGE_PIN T18 [get_ports btnU]
113 |   #set_property IOSTANDARD LVCMOS33 [get_ports btnU]
114 |   #set_property PACKAGE_PIN W19 [get_ports btnL]
115 |   #set_property IOSTANDARD LVCMOS33 [get_ports btnL]
116 |   #set_property PACKAGE_PIN T17 [get_ports btnR]
117 |   #set_property IOSTANDARD LVCMOS33 [get_ports btnR]
118 |   #set_property PACKAGE_PIN U17 [get_ports btnD]
119 |   #set_property IOSTANDARD LVCMOS33 [get_ports btnD]
120 |
121 |
122 |
123 |
124 ##Pmod Header JA
125 ##Sch name = JA1
126 set_property PACKAGE_PIN J1 [get_ports {JA[0]}]
127 |   #set_property IOSTANDARD LVCMOS33 [get_ports {JA[0]}]
128 ##Sch name = JA2
129 set_property PACKAGE_PIN L2 [get_ports {JA[1]}]
130 |   #set_property IOSTANDARD LVCMOS33 [get_ports {JA[1]}]

```



EE 316 Lab 5

Arman Khanoker

AAK2464

$$P_i = a_i \oplus b_i$$

$$G_i = a_i b_i$$

$$C_{i+1} = G_i + P_i C_i$$

 C_0 = input carry

$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1$$

$$= G_1 + P_1 (P_0 C_0 + G_0)$$

$$= G_1 + P_1 P_0 C_0 + P_1 G_0$$

$$S_C = P_i \oplus C_i$$

$$C_3 = G_2 + P_2 G_2$$

$$= G_2 + P_2 (G_1 + P_1 P_0 C_0 + P_1 G_0)$$

$$S_0 = P_0 \oplus C_0 = a_0 \oplus b_0 \oplus c_0$$

$$C_4 = G_3 + P_3 C_3$$

$$= G_3 + P_3 (G_2 + P_2 (G_1 + P_1 P_0 C_0 + P_1 G_0))$$

= output carry

$$S_1 = P_1 \oplus C_1 = a_1 \oplus b_1 \oplus c_1$$

$$S_2 = P_2 \oplus C_2 = a_2 \oplus b_2 \oplus c_2$$

$$S_3 = P_3 \oplus C_3 = a_3 \oplus b_3 \oplus c_3$$

```
// Project Name: CLA_4bits.v
// 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 [3:0] G, P, S;
wire [4:0] C;
wire [4:0] Sum;
wire Cout;
assign C[0] = Cin;
assign G[0] = (A[0]&B[0]);
assign G[1] = (A[1]&B[1]);
assign G[2] = (A[2]&B[2]);
assign G[3] = (A[3]&B[3]);
assign P[0] = (A[0]^B[0]);
assign P[1] = (A[1]^B[1]);
assign P[2] = (A[2]^B[2]);
assign P[3] = (A[3]^B[3]);
assign C[0] = Cin;
assign C[1] = (G[0] | (P[0]&C[0]));
assign C[2] = (G[1] | (P[1]&G[0]) | (P[1]&P[0]&C[0]));
assign C[3] = (G[2] | (P[2]&G[1]) | (P[2]&P[1]&G[0]) | (P[2]&P[0]&C[0]));
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]&C[0]));
assign Sum[0] = C[0]^P[0];
assign Sum[1] = C[1]^P[1];
assign Sum[2] = C[2]^P[2];
assign Sum[3] = C[3]^P[3];
assign Sum[4] = C[4];
register_logic R2(clk, enable, Sum, Q);
endmodule
```

```
1 //timescale 1ns / 1ps
2 ///////////////////////////////////////////////////////////////////
3 // Company:
4 // Engineer:
5 //
6 // Create Date: 04/09/2019 04:20:26 PM
7 // Design Name:
8 // Module Name: register_logic
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21
22 module register_logic(
23     input clk,
24     input enable,
25     input [4:0] Data,
26     output reg [4:0] Q
27 );
28
29     initial begin
30         Q=0;
31     end
32
33     always @ (posedge clk) begin
34         if(enable)
35             Q <= Data;
36         end
37
38 endmodule
39
40
41
```

```
CLA_4bits.v x RCA_4bits.v x tb_CLA_4bits.v x tb_RCA_4bits.v x Basys3_Master.xdc x register_logic.v x
1 `timescale 1ns / 1ps
2 ///////////////////////////////////////////////////////////////////
3 // Company:
4 // Engineer:
5 //
6 // Create Date: 04/09/2019 05:06:09 PM
7 // Design Name:
8 // Module Name: tb_CLA_4bits
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21
22
23 module tb_CLA_4bits;
24     reg clk;
25     reg enable;
26     reg [3:0] A;
27     reg [3:0] B;
28     reg Cin;
29     wire [4:0] Q;
30
31     CLA_4bits uut(
32         .clk(clk),
33         .enable(enable),
34         .A(A),
35         .B(B),
36         .Cin(Cin),
37         .Q(Q)
38     );
39
40     initial begin
41         clk = 0;
42         enable = 0;
43         A = 4'b0000;
44         B = 4'b0000;
45         Cin = 0;
46         #50;
47
48         enable = 1;
49         A = 4'b0000;
50         B = 4'b0101;
51         #50;
52
53         A = 4'b0101;
54         B = 4'b0111;
55         #50;
56
57         A = 4'b1000;
58         B = 4'b0111;
59         Cin = 1;
60
61     end
62
63 endmodule
```

Line 1, Column 1

Spaces: 4

Plain Text

```
44     B = 4'b0000;
45     Cin = 0;
46     #50;
47
48     enable = 1;
49     A = 4'b0000;
50     B = 4'b0101;
51     #50;
52
53     A = 4'b0101;
54     B = 4'b0111;
55     #50;
56
57     A = 4'b1000;
58     B = 4'b0111;
59     Cin = 1;
60     #50;
61
62     A = 4'b1001;
63     B = 4'b0100;
64     Cin = 0;
65     #50;
66
67     A = 4'b1000;
68     B = 4'b1000;
69     Cin = 1;
70     #50;
71
72     A = 4'b1101;
73     B = 4'b1010;
74     #50;
75
76     A = 4'b1110;
77     B = 4'b1111;
78     Cin = 0;
79     #50;
80
81   end
82
83   always
84
85     #5 clk = ~clk;
86
87 endmodule
88
89
90
91
```



Line 1, Column 1

Spaces: 4

Plain Text

A[3:0]	B[3:0]	Cin	Sum[3:0]	Cout
0000	0101	0	0101	0
0101	0111	0	1100	0
1000	0111	1	0000	1
1001	0100	0	1111	0
1000	1000	1	0001	1
1101	1010	1	1000	1
1110	1111	0	1101	1

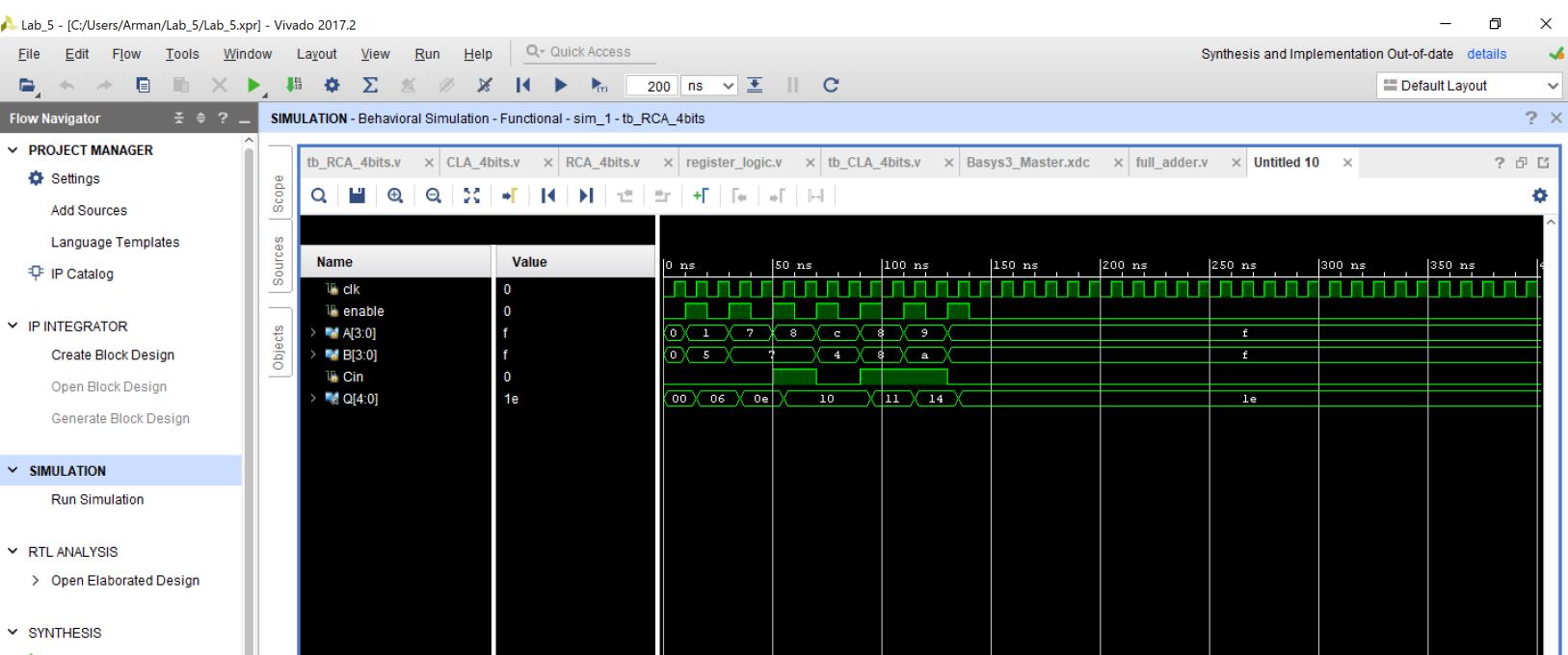
Table 2. Testcases for Carry Lookahead Adder Verification

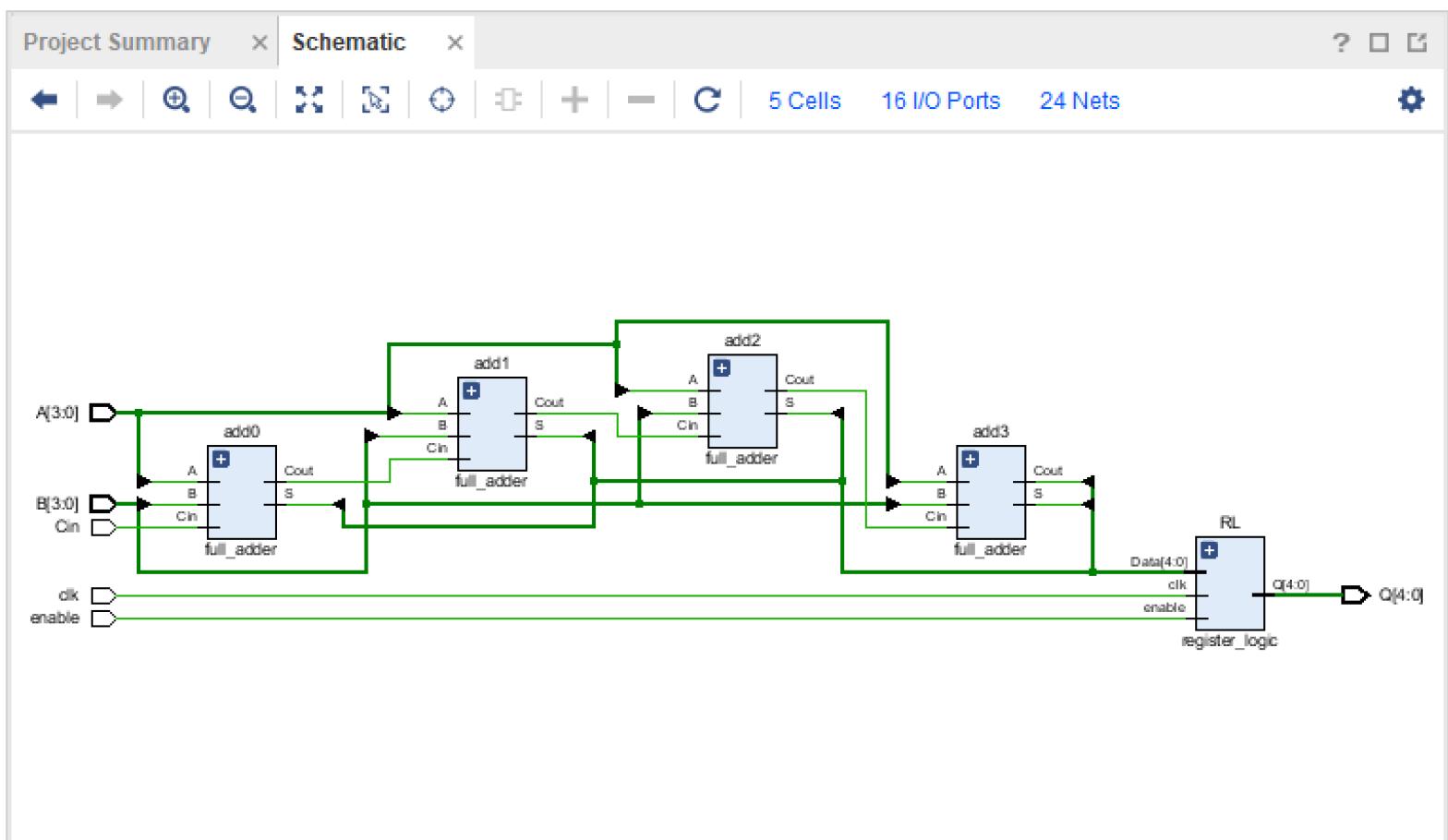
```

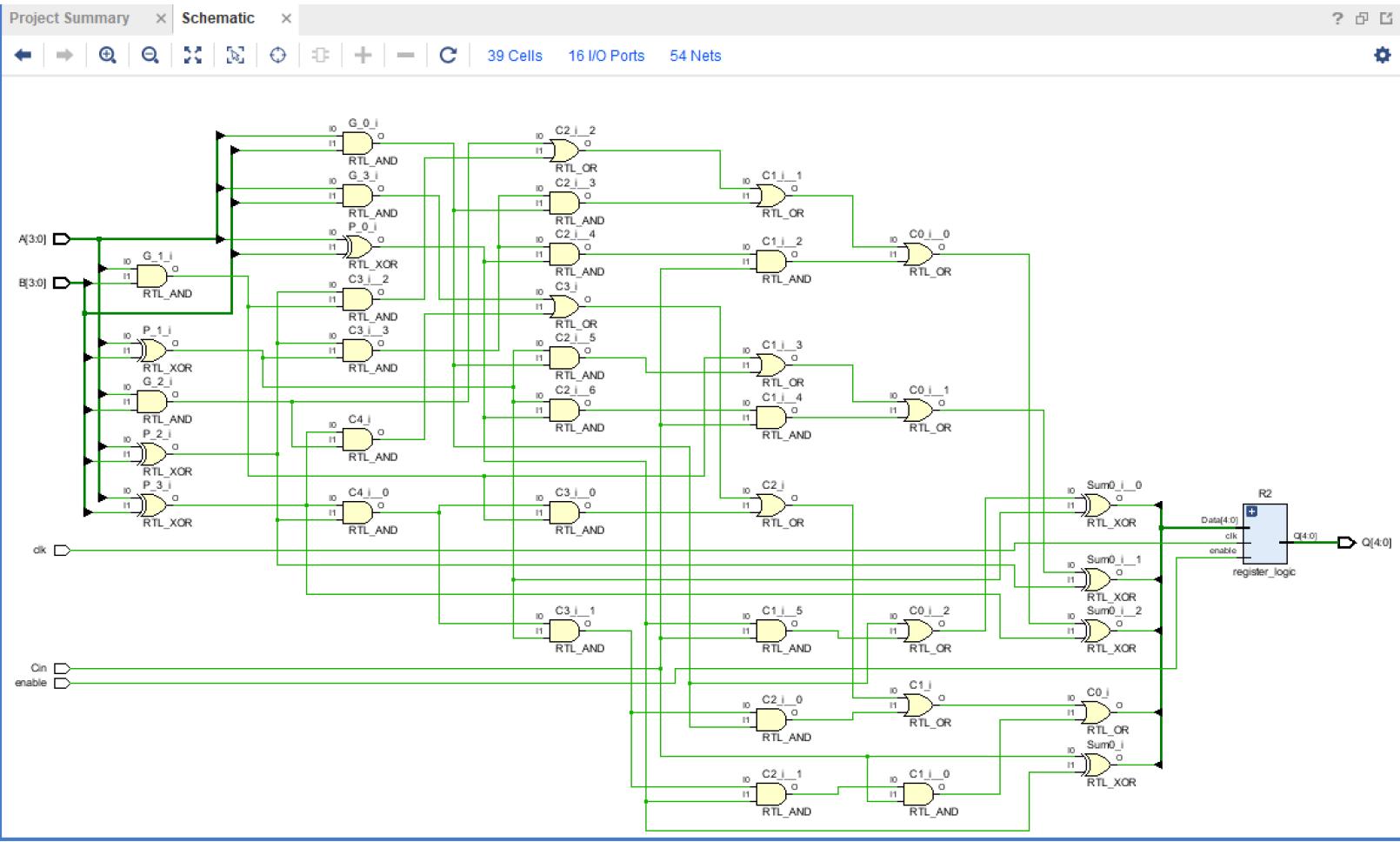
1 ## This file is a general .xdc for the Basys3 rev B board
2 ## To use it in a project:
3 ## - uncomment the lines corresponding to used pins
4 ## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project
5
6 ## Clock signal
7 set_property PACKAGE_PIN W5 [get_ports clk]
8   set_property IOSTANDARD LVCMS033 [get_ports clk]
9   create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]
10
11 ## Switches
12 set_property PACKAGE_PIN V17 [get_ports {A[0]}]
13   set_property IOSTANDARD LVCMS033 [get_ports {A[0]}]
14   set_property PACKAGE_PIN V16 [get_ports {A[1]}]
15     set_property IOSTANDARD LVCMS033 [get_ports {A[1]}]
16   set_property PACKAGE_PIN V16 [get_ports {A[2]}]
17     set_property IOSTANDARD LVCMS033 [get_ports {A[2]}]
18   set_property PACKAGE_PIN W17 [get_ports {A[3]}]
19     set_property IOSTANDARD LVCMS033 [get_ports {A[3]}]
20   set_property PACKAGE_PIN W15 [get_ports {A[0]}]
21   set_property IOSTANDARD LVCMS033 [get_ports {B[0]}]
22   set_property PACKAGE_PIN V15 [get_ports {B[1]}]
23   set_property IOSTANDARD LVCMS033 [get_ports {B[1]}]
24   set_property PACKAGE_PIN W14 [get_ports {B[2]}]
25     set_property IOSTANDARD LVCMS033 [get_ports {B[2]}]
26   set_property PACKAGE_PIN W13 [get_ports {B[3]}]
27     set_property IOSTANDARD LVCMS033 [get_ports {B[3]}]
28   set_property PACKAGE_PIN V2 [get_ports {Cin}]
29   set_property IOSTANDARD LVCMS033 [get_ports {Cin}]
30   #set_property PACKAGE_PIN T3 [get_ports {sw[9]}]
31   #set_property IOSTANDARD LVCMS033 [get_ports {sw[9]}]
32   #set_property PACKAGE_PIN T2 [get_ports {sw[10]}]
33     #set_property IOSTANDARD LVCMS033 [get_ports {sw[10]}]
34   #set_property PACKAGE_PIN R3 [get_ports {sw[11]}]
35     #set_property IOSTANDARD LVCMS033 [get_ports {sw[11]}]
36   #set_property PACKAGE_PIN W2 [get_ports {sw[12]}]
37     #set_property IOSTANDARD LVCMS033 [get_ports {sw[12]}]
38   #set_property PACKAGE_PIN U1 [get_ports {sw[13]}]
39     #set_property IOSTANDARD LVCMS033 [get_ports {sw[13]}]
40   #set_property PACKAGE_PIN T1 [get_ports {sw[14]}]
41   #set_property IOSTANDARD LVCMS033 [get_ports {sw[14]}]
42   #set_property PACKAGE_PIN R2 [get_ports {sw[15]}]
43     #set_property IOSTANDARD LVCMS033 [get_ports {sw[15]}]
44
45
46 ## LEDs
47 set_property PACKAGE_PIN U16 [get_ports {O[0]}]
48   set_property IOSTANDARD LVCMS033 [get_ports {O[0]}]
49 set_property PACKAGE_PIN E19 [get_ports {O[1]}]
50   set_property IOSTANDARD LVCMS033 [get_ports {O[1]}]
51   set_property IOSTANDARD LVCMS033 [get_ports {O[2]}]
52     set_property IOSTANDARD LVCMS033 [get_ports {O[2]}]
53   set_property PACKAGE_PIN V19 [get_ports {O[3]}]
54     set_property IOSTANDARD LVCMS033 [get_ports {O[3]}]
55   set_property PACKAGE_PIN W18 [get_ports {O[4]}]
56     set_property IOSTANDARD LVCMS033 [get_ports {O[4]}]
57   #set_property PACKAGE_PIN U15 [get_ports {led[5]}]
58   #set_property IOSTANDARD LVCMS033 [get_ports {led[5]}]
59   #set_property PACKAGE_PIN U14 [get_ports {led[6]}]
60     #set_property IOSTANDARD LVCMS033 [get_ports {led[6]}]
61   #set_property PACKAGE_PIN V14 [get_ports {led[7]}]
62     #set_property IOSTANDARD LVCMS033 [get_ports {led[7]}]
63   #set_property PACKAGE_PIN V11 [get_ports {led[8]}]
64     #set_property IOSTANDARD LVCMS033 [get_ports {led[8]}]
65   #set_property PACKAGE_PIN U10 [get_ports {led[9]}]
66     #set_property IOSTANDARD LVCMS033 [get_ports {led[9]}]
67   #set_property PACKAGE_PIN W3 [get_ports {led[10]}]
68     #set_property IOSTANDARD LVCMS033 [get_ports {led[10]}]
69   #set_property PACKAGE_PIN U3 [get_ports {led[11]}]
70     #set_property IOSTANDARD LVCMS033 [get_ports {led[11]}]
71   #set_property PACKAGE_PIN P3 [get_ports {led[12]}]

```

Line 1, Column 1 Tab Size: 4 Plain Text







RCA

Note,

	Delay	Area
XOR	3	
AND	3	6
OR	2	4

Delay

1 AND 2 OR

$$= 4(3 + 2(2)) = \boxed{20 \text{ ns}}$$

CLA

Delay

1 XOR

4 AND

1 OR

$$= \boxed{17 \text{ ns}}$$

$$\text{Delay} = 3 + 4(3) + 2$$

Area

3 AND
2 XOR
2 OR

$$\begin{aligned}
 &= 4[3(4) + 2(6) + 2(4)] \\
 &= 4(12 + 12 + 8) \\
 &= \boxed{128 \text{ unit area}}
 \end{aligned}$$

Area

20 AND 8 XOR

10 OR

$$\text{Area} = 20(4) + 8(6) + 10(4)$$

$$= \boxed{168 \text{ unit area}}$$

CONCLUSIONS OF PROS AND CONS OF EACH TECHNIQUE

Each of the adder implementations have their respective pros and cons. Moreover, the RLA is slower and uses more space. However, it is cheaper to create because it has less gates. When doing a smaller bit width, the RCA is preferable. The RCA is easier to trace logic and examine. On the other hand, the CLA is faster since the $g(x)$ and $p(x)$ are calculated at the same time