**Lab 5 Report**

**Name: Aryan Agarwal, Sanjay Sivakumar**

**UT EID: aab5473**

**Section: MW 11 – 12:30**

**Checklist:**

**Part 1 –**

1. Design file (.v) for the Ripple Carry Adder
2. Test-bench
3. Complete Table 1 from the simulation

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

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

1. Constraints File (Just the uncommented portion)
2. Simulation waveform for the above test-cases

**Part 2 –**

1. All the equations for Ci’sand Si’s
2. Design files (.v) for the Carry Lookahead Adder and Register Logic
3. Test-bench
4. Complete Table 2 from the simulation

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

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

1. Constraints File (Just the uncommented portion)
2. Simulation waveform for the above test-cases

**Part 3 –**

1. Screenshots of the gate-level schematics for both the adder techniques
2. Delay and area for both the adder techniques showing all the work
3. Brief conclusion regarding the pros and cons of each of the techniques

***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.*

**Part 1**

Verilog:

`timescale 1ns / 1ps

module RCA\_4bits(

input clk,

input enable,

input [3:0] A,B,

input Cin,

output [4:0] Q

);

wire [2:0] CarryBit;

wire [4:0] DataReg;

full\_adder c0(.A(A[0]), .B(B[0]), .Cin(Cin), .S(DataReg[0]), .Cout(CarryBit[0]));

full\_adder c1(.A(A[1]), .B(B[1]), .Cin(CarryBit[0]), .S(DataReg[1]), .Cout(CarryBit[1]));

full\_adder c2(.A(A[2]), .B(B[2]), .Cin(CarryBit[1]), .S(DataReg[2]), .Cout(CarryBit[2]));

full\_adder c3(.A(A[3]), .B(B[3]), .Cin(CarryBit[2]), .S(DataReg[3]), .Cout(DataReg[4]));

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

endmodule

Testbench:

`timescale 1ns / 1ps

module RCA\_tb;

reg clk;

reg enable;

reg [3:0] A, B;

reg Cin;

wire [4:0] Q;

RCA\_4bits uut(

.clk(clk),

.enable(enable),

.A(A),

.B(B),

.Cin(Cin),

.Q(Q)

);

initial

begin

clk = 0;

enable = 1;

Cin = 0;

A = 4'b0001;

B = 4'b0101;

#10;

A = 4'b0111;

B = 4'b0111;

#10

Cin = 1;

A = 4'b1000;

B = 4'b0111;

#10;

Cin = 0;

A = 4'b1100;

B = 4'b0100;

#10;

Cin = 1;

A = 4'b1000;

B = 4'b1000;

#10

Cin = 1;

A = 4'b1001;

B = 4'b1010;

#10

Cin = 0;

A = 4'b1111;

B = 4'b1111;

end

always

#5 clk = ~clk;

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 |

Constraints:

set\_property PACKAGE\_PIN W5 [get\_ports clk]

set\_property IOSTANDARD LVCMOS33 [get\_ports clk]

create\_clock -add -name sys\_clk\_pin -period 10.00 -waveform {0 5} [get\_ports clk]

#Part 1

set\_property PACKAGE\_PIN V17 [get\_ports {A[0]}]

## Sets the switch to use 3.3V logic

set\_property IOSTANDARD LVCMOS33 [get\_ports {A[0]}]

## Connects pin V16 (SW1 on the board) to input b in our gate module

set\_property PACKAGE\_PIN V16 [get\_ports {A[1]}]

## Sets the switch to use 3.3V logic

set\_property IOSTANDARD LVCMOS33 [get\_ports {A[1]}]

set\_property PACKAGE\_PIN W16 [get\_ports {A[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {A[2]}]

set\_property PACKAGE\_PIN W17 [get\_ports {A[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {A[3]}]

set\_property PACKAGE\_PIN W15 [get\_ports {B[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {B[0]}]

set\_property PACKAGE\_PIN V15 [get\_ports {B[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {B[1]}]

set\_property PACKAGE\_PIN W14 [get\_ports {B[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {B[2]}]

set\_property PACKAGE\_PIN W13 [get\_ports {B[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {B[3]}]

set\_property PACKAGE\_PIN R2 [get\_ports {Cin}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {Cin}]

set\_property PACKAGE\_PIN U18 [get\_ports enable]

set\_property IOSTANDARD LVCMOS33 [get\_ports enable]

set\_property PACKAGE\_PIN U16 [get\_ports {Q[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[0]}]

set\_property PACKAGE\_PIN E19 [get\_ports {Q[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[1]}]

set\_property PACKAGE\_PIN U19 [get\_ports {Q[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[2]}]

set\_property PACKAGE\_PIN V19 [get\_ports {Q[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[3]}]

set\_property PACKAGE\_PIN W18 [get\_ports {Q[4]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[4]}]

A screenshot of a computer

Description automatically generated

**Part 2**

A white paper with black text

Description automatically generated

Verilog:

CLA:

`timescale 1ns / 1ps

module CLA\_4bits(

input clk,

input enable,

input [3:0] A, B,

input Cin,

output [4:0] Q

);

wire [3:0] G, P, S;

wire [4:0] C;

wire Cout;

wire [4:0] DataReg;

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[1] = G[0] + (P[0]\*C[0]);

assign C[2] = G[1] + (P[1]\*G[0]) + (P[0]\*C[0]\*P[1]);

assign C[3] = G[2] + (P[2]\*G[1]) + (P[2]\*P[1]\*G[0]) + (P[2]\*P[1]\*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]\*P[0]\*C[0]);

assign Cout = C[4];

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

assign S[1] = P[1] ^ (G[0] + (P[0]\*C[0]));

assign S[2] = P[2] ^ (G[1] + (P[1]\*G[0]) + (P[0]\*C[0]\*P[1]));

assign S[3] = P[3] ^ (G[2] + (P[2]\*G[1]) + (P[2]\*P[1]\*G[0]) + (P[2]\*P[1]\*P[0]\*C[0]));

assign DataReg[3:0] = S;

assign DataReg[4] = Cout;

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

endmodule

Test Bench:

`timescale 1ns / 1ps

module CLA\_tb;

reg clk;

reg enable;

reg [3:0] A, B;

reg Cin;

wire [4:0] Q;

CLA\_4bits uut(

.clk(clk),

.enable(enable),

.A(A),

.B(B),

.Cin(Cin),

.Q(Q)

);

initial

begin

clk = 0;

enable = 1;

Cin = 0;

A = 4'b0000;

B = 4'b0101;

#10;

A = 4'b0101;

B = 4'b0111;

#10

Cin = 1;

A = 4'b1000;

B = 4'b0111;

#10;

Cin = 0;

A = 4'b1001;

B = 4'b0100;

#10;

Cin = 1;

A = 4'b1000;

B = 4'b1000;

#10

Cin = 1;

A = 4'b1101;

B = 4'b1010;

#10

Cin = 0;

A = 4'b1110;

B = 4'b1111;

end

always

#5 clk = ~clk;

Endmodule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |

Constraints:

set\_property PACKAGE\_PIN V17 [get\_ports {A[0]}]

## Sets the switch to use 3.3V logic

set\_property IOSTANDARD LVCMOS33 [get\_ports {A[0]}]

## Connects pin V16 (SW1 on the board) to input b in our gate module

set\_property PACKAGE\_PIN V16 [get\_ports {A[1]}]

## Sets the switch to use 3.3V logic

set\_property IOSTANDARD LVCMOS33 [get\_ports {A[1]}]

set\_property PACKAGE\_PIN W16 [get\_ports {A[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {A[2]}]

set\_property PACKAGE\_PIN W17 [get\_ports {A[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {A[3]}]

set\_property PACKAGE\_PIN W15 [get\_ports {B[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {B[0]}]

set\_property PACKAGE\_PIN V15 [get\_ports {B[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {B[1]}]

set\_property PACKAGE\_PIN W14 [get\_ports {B[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {B[2]}]

set\_property PACKAGE\_PIN W13 [get\_ports {B[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {B[3]}]

set\_property PACKAGE\_PIN V2 [get\_ports {Cin}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {Cin}]

set\_property PACKAGE\_PIN U18 [get\_ports enable]

set\_property IOSTANDARD LVCMOS33 [get\_ports enable]

set\_property PACKAGE\_PIN U16 [get\_ports {Q[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[0]}]

set\_property PACKAGE\_PIN E19 [get\_ports {Q[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[1]}]

set\_property PACKAGE\_PIN U19 [get\_ports {Q[2]}]

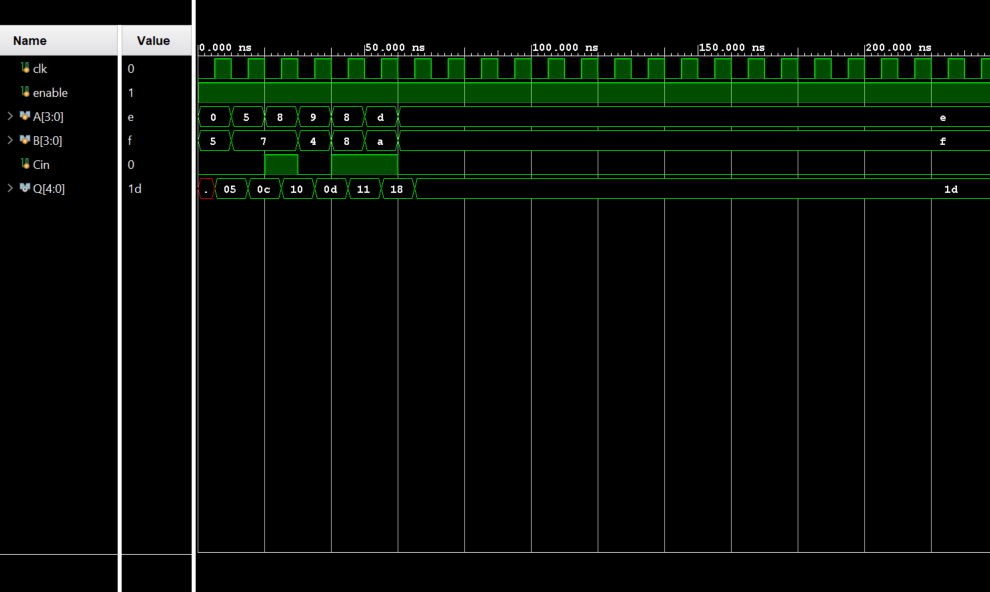
set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[2]}]

set\_property PACKAGE\_PIN V19 [get\_ports {Q[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[3]}]

set\_property PACKAGE\_PIN W18 [get\_ports {Q[4]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {Q[4]}]



**Part 3**

Carry Lookahead Adder:

A diagram of a circuit

Description automatically generated

Ripple Carry Adder:

A diagram of a circuit

Description automatically generated

The CLA utilizes more gates which results in more area. However, because it takes into account all possibilities, it will perform the functions quicker.  
The RCA utilizes less gates and is more versatile. As such, it is smaller and uses less area. However, the RCA takes more time than the CLA to perform computations. As such, if the implementation needs the actions to be fastest, a CLA would be the best. However, if the implementation requires less space, the RCA would be the best choice.