CECS 341

Lab 3

Eugene Paul Mesina



* **Section 1:** RCA8 Verilog module source

`timescale 1ns / 1ps

module RCA8(

input [7:0] A\_8,

input [7:0] B\_8,

input Cin,

output [7:0] S\_8,

output Cout

);

wire c0, c1, c2, c3, c4, c5, c6;

FullAdder fa0( .FA\_A( A\_8[0] ),

.FA\_B( B\_8[0] ),

.Cin( Cin ),

.FA\_S( S\_8[0] ),

.Cout( c0 ));

FullAdder fa1( .FA\_A( A\_8[1] ),

.FA\_B( B\_8[1] ),

.Cin( c0 ),

.FA\_S( S\_8[1] ),

.Cout( c1 ));

FullAdder fa2( .FA\_A( A\_8[2] ),

.FA\_B( B\_8[2] ),

.Cin( c1 ),

.FA\_S( S\_8[2] ),

.Cout( c2 ));

FullAdder fa3( .FA\_A( A\_8[3] ),

.FA\_B( B\_8[3] ),

.Cin( c2 ),

.FA\_S( S\_8[3] ),

.Cout( c3 ));

FullAdder fa4( .FA\_A( A\_8[4] ),

.FA\_B( B\_8[4] ),

.Cin( c3 ),

.FA\_S( S\_8[4] ),

.Cout( c4 ));

FullAdder fa5( .FA\_A( A\_8[5] ),

.FA\_B( B\_8[5] ),

.Cin( c4 ),

.FA\_S( S\_8[5] ),

.Cout( c5 ));

FullAdder fa6( .FA\_A( A\_8[6] ),

.FA\_B( B\_8[6] ),

.Cin( c5 ),

.FA\_S( S\_8[6] ),

.Cout( c6 ));

FullAdder fa7( .FA\_A( A\_8[7] ),

.FA\_B( B\_8[7] ),

* **Section 2:** RCA8 Verilog Test Fixture

`timescale 1ns / 1ps

module RCA8\_Tester;

// Inputs

reg [7:0] A\_8;

reg [7:0] B\_8;

reg Cin;

// Outputs

wire [7:0] S\_8;

wire Cout;

// Instantiate the Unit Under Test (UUT)

RCA8 uut (

.A\_8(A\_8),

.B\_8(B\_8),

.Cin(Cin),

.S\_8(S\_8),

.Cout(Cout)

);

initial begin

// Initialize Inputs

A\_8 = 8'b00010010;

B\_8 = 8'b00110100;

Cin = 0;

// Wait 100 ns for global reset to finish

#100;

// Initialize Inputs

A\_8 = 8'b01010101;

B\_8 = 8'b10101010;

Cin = 0;

// Wait 100 ns for global reset to finish

#100;

// Initialize Inputs

A\_8 = 8'b01110010;

B\_8 = 8'b00100111;

Cin = 0;

// Wait 100 ns for global reset to finish

#100;

// Initialize Inputs

A\_8 = 8'b10000000;

B\_8 = 8'b00001000;

Cin = 0;

// Wait 100 ns for global reset to finish

#100;

// Initialize Inputs

A\_8 = 8'b00010010;

B\_8 = 8'b00110100;

Cin = 1;

// Wait 100 ns for global reset to finish

#100;

// Initialize Inputs

A\_8 = 8'b01010101;

B\_8 = 8'b10101010;

Cin = 1;

// Wait 100 ns for global reset to finish

#100;

// Initialize Inputs

A\_8 = 8'b01110010;

B\_8 = 8'b00100111;

Cin = 1;

// Wait 100 ns for global reset to finish

#100;

// Initialize Inputs

A\_8 = 8'b10000000;

B\_8 = 8'b00001000;

Cin = 1;

// Wait 100 ns for global reset to finish

#100;

// Add stimulus here

$stop;

end

endmodule

* **Section 3:** Simulation Screenshot showing correct results

