8-Bit RCA

A screen shot of a computer

Description automatically generated

3x8 Decoder

A screenshot of a computer

Description automatically generated

**Lab-2.v**

`timescale 1ns */* 1ps

*module* Full\_Adder( A, B, Cin, Sum, Cout );

   input A, B, Cin;

   output Sum, Cout;

*assign* Sum *=* ( A *^* B ) *^* Cin;

*assign* Cout *=* ( ( A *^* B ) *&&* Cin ) *||* ( A *&* B );

*endmodule*

*module* RCA\_4b( A, B, Cin, Sum, Cout );

    input [3*:*0] A, B;

    input Cin;

    output [3*:*0] Sum;

    output Cout;

*wire* [2*:*0] Carry;

*Full\_Adder* adder1( A[0], B[0], Cin, Sum[0], Carry[0]);

*Full\_Adder* adder2( A[1], B[1], Carry[0], Sum[1], Carry[1]);

*Full\_Adder* adder3( A[2], B[2], Carry[1], Sum[2], Carry[2]);

*Full\_Adder* adder4( A[3], B[3], Carry[2], Sum[3], Cout);

*endmodule*

*module* RCA\_8b( A, B, Cin, Sum, Cout );

    input [7*:*0] A, B;

    input Cin;

    output [7*:*0] Sum;

    output Cout;

*wire* Carry;

*RCA\_4b* rca1( A[3*:*0], B[3*:*0], Cin, Sum[3*:*0], Carry );

*RCA\_4b* rca2( A[7*:*4], B[7*:*4], Carry, Sum[7*:*4], Cout );

*endmodule*

*module* Decoder3x8( In, Out );

    input [2*:*0] In;

    output [7*:*0] Out;

*reg* [7*:*0] Out;

*always* *@*( In ) *begin*

        Out *<=* ( 1'b1 *<<* In );

*end*

*endmodule*

**Lab-2\_tb.v**

`timescale 1ns */* 1ps

*module* Lab2\_tb();

Decoder3x8 decoder();

RCA\_8b rca();

*endmodule*

*module* Decoder3x8\_tb ();

*reg* [2*:*0] In;

*wire* [7*:*0] Out;

*Decoder3x8* uut( In, Out );

*integer* i;

*initial* *begin*

        In *=* 3'b0;

*for* ( i *=* 0; i *<* 8; i *=* i *+* 1 ) *begin*

            In *=* In *+* 1;

*end*

*end*

*endmodule*

*module* RCA\_8b\_tb();

*reg* [7*:*0] A, B;

*reg* Cin;

*wire* [7*:*0] Sum;

*wire* out;

*RCA\_8b* uut( A, B, Cin, Sum, Cout );

*initial* *begin*

*// Test 1*

        Cin *=* 1'b0;

        A *=* 8'b01011000;

        B *=* 8'b00011001;

        $monitor( "A: %b\nB: %b\nCin: %b\n\nSum: %b\nCout: %b\n\n-------------------------\n", A, B, Cin, Sum, Cout );

*#*100;

*// Test 2*

        Cin *=* 1'b1;

        A *=* 8'b01100011;

        B *=* 8'b11011001;

        $monitor( "A: %b\nB: %b\nCin: %b\n\nSum: %b\nCout: %b\n\n-------------------------\n", A, B, Cin, Sum, Cout );

*#*100;

*// Test 3*

        Cin*=* 1'b1;

        A *=* 8'b00110011;

        B *=* 8'b00111100;

        $monitor( "A: %b\nB: %b\nCin: %b\n\nSum: %b\nCout: %b\n\n-------------------------\n", A, B, Cin, Sum, Cout );

*#*100;

*// Test 4*

        Cin*=*1'b0;

        A *=* 8'b10110011;

        B *=* 8'b01111100;

        $monitor( "A: %b\nB: %b\nCin: %b\n\nSum: %b\nCout: %b\n\n-------------------------\n", A, B, Cin, Sum, Cout );

*end*

*endmodule*

*// module Full\_Adder\_tb();*

*//     reg A, B, Cin;*

*//     wire Sum, Cout;*

*//     Full\_Adder uut( A, B, Cin, Sum, Cout );*

*//     integer i;*

*//     initial begin*

*//         A = 1'b0;*

*//         B = 1'b0;*

*//         Cin = 1'b0;*

*//         for ( i = 0; i < 7; i = i + 1 ) begin*

*//             #100;*

*//             { A, B, Cin } = { A, B, Cin } + 1;*

*//         end*

*//     end*

*// endmodule*

*// module RCA\_tb();*

*//     reg [3:0] A, B;*

*//     reg Cin;*

*//     wire [3:0] Sum;*

*//     wire Cout;*

*//     RCA\_4b uut( A, B, Cin, Sum, Cout );*

*//     integer i;*

*//     initial begin*

*//         #100*

*//         A = 4'b000;*

*//         B = 4'b000;*

*//         Cin = 1'b0;*

*//         for ( i = 0; i < ( ( 2 \*\* 9 ) - 1 ); i = i + 1 ) begin*

*//             { A, B, Cin } = { A, B, Cin } + 1;*

*//         end*

*//     end*

*// endmodule*