**Carry Save Adder Verilog Code**

module carry\_save\_adder(a,b,c,d, sum,cout);

input [3:0] a, b,c,d;

output [4:0] sum;

output cout;

wire [3:0] s0,s1;

wire [3:0] c0, c1;

full\_adder fa0( .a(a[0]), .b(b[0]), .cin(c[0]), .sum(s0[0]), .cout(c0[0]));

full\_adder fa1( .a(a[1]), .b(b[1]), .cin(c[1]), .sum(s0[1]), .cout(c0[1]));

full\_adder fa2( .a(a[2]), .b(b[2]), .cin(c[2]), .sum(s0[2]), .cout(c0[2]));

full\_adder fa3( .a(a[3]), .b(b[3]), .cin(c[3]), .sum(s0[3]), .cout(c0[3]));

full\_adder fa4( .a(d[0]), .b(s0[0]), .cin(1'b0), .sum(sum[0]), .cout(c1[0]));

full\_adder fa5( .a(d[1]), .b(s0[1]), .cin(c0[0]), .sum(s1[0]), .cout(c1[1]));

full\_adder fa6( .a(d[2]), .b(s0[2]), .cin(c0[1]), .sum(s1[1]), .cout(c1[2]));

full\_adder fa7( .a(d[3]), .b(s0[3]), .cin(c0[2]), .sum(s1[2]), .cout(c1[3]));

ripple\_carry\_4\_bit rca1 (.a(c1[3:0]),.b({c0[3],s1[2:0]}), .cin(1'b0),.sum(sum[4:1]), .cout(cout));

endmodule

**TEST BENCH**

module carry\_save\_tb;

wire [4:0] sum;//output

wire cout;//output

reg [3:0] a,b,c,d;//input

carry\_save\_adder uut(

.a(a),

.b(b),

.c(c),

.d(d),

.sum(sum),

.cout(cout));

initial begin

a=0; b=0; c=0; d=0;

#100 a= 4'd10; b=4'd0; c=4'd0; d=4'd0;

#100 a= 4'd10; b=4'd10; c=4'd0; d=4'd0;

#100 a= 4'd4; b=4'd6; c=4'd12; d=4'd0;

#100 a= 4'd11; b=4'd2; c=4'd4; d=4'd7;

#100 a= 4'd20; b=4'd0; c=4'd20; d=4'd0;

#100 a= 4'd12; b=4'd5; c=4'd10; d=4'd10;

#100 a= 4'd7; b=4'd6; c=4'd12; d=4'd8;

#100 a= 4'd15; b=4'd15; c=4'd15; d=4'd15;

end

endmodule

**OUTPUT**

