

```

module serial_adder
    (    input clk,reset,
      input a,b,cin,
      output reg s,cout);

reg c,flag;

always@(posedge clk or posedge reset)
begin
    if(reset == 1) begin
        s = 0;
        cout = c;
        flag = 0;
    end else begin
        if(flag == 0) begin
            c = cin;
            flag = 1;
        end
        cout = 0;
        s = a ^ b ^ c; //SUM
        c = (a & b) | (c & b) | (a & c); //CARRY
    end
end

endmodule

```

### **//Stimulus block**

```

module stimulus;

    reg clk;
    reg reset;
    reg a;
    reg b;
    reg cin;

    wire s;
    wire cout;

    serial_adder s1 (
        .clk(clk),
        .reset(reset),
        .a(a),
        .b(b),
        .cin(cin),
        .s(s),
        .cout(cout)
    );

```

**always**

    #5 clk = ~clk;

**initial begin**

    clk = 1;

    reset = 0;

    a = 0;

    b = 0;

    cin = 0;

    reset = 1;

    #20;

    reset = 0;

    //add two 4 bit numbers, 1111 + 1101 = 11101

    a = 1; b = 1; cin = 1;     #10;

    a = 1; b = 0; cin = 0;    #10;

    a = 1; b = 1; cin = 0;    #10;

    a = 1; b = 1; cin = 0;    #10;

    reset = 1;

    #10;

    reset = 0;

    //add two 5 bit numbers, 11011 + 10001 = 101101

    a = 1; b = 1; cin = 1;     #10;

    a = 1; b = 0; cin = 0;    #10;

    a = 0; b = 0; cin = 0;    #10;

    a = 1; b = 0; cin = 0;    #10;

    a = 1; b = 1; cin = 0;    #10;

    reset = 1;

    #10;

**end**

**endmodule**