

Behavioral Modelling

```
module full_adder(
    input A,B,Cin,
    output reg Sum,
    output reg Cout
);
    always @ (A or B or Cin) begin
        {Cout,Sum} = A+B+Cin;
    end
endmodule
```

Test Bench

```
module full_adder_tb;
    reg A,B,Cin;
    wire Sum, Cout;

    // Instantiate uut
    full_adder uut(
        .A(A),
        .B(B),
        .Cin(Cin),
        .Sum(Sum),
        .Cout(Cout)
    );

    initial begin
        $display ("A B Cin | Sum Cout");
    end
endmodule
```

```

A=0;B=0;Cin=0;#10;
$display("%b %b %b | %b %b",A,B,Cin,Sum,Cout);

A=0;B=0;Cin=1;#10;
$display("%b %b %b | %b %b",A,B,Cin,Sum,Cout);

A=0;B=1;Cin=0;#10;
$display("%b %b %b | %b %b",A,B,Cin,Sum,Cout);

A=0;B=1;Cin=1;#10;
$display("%b %b %b | %b %b",A,B,Cin,Sum,Cout);

A=1;B=0;Cin=0;#10;
$display("%b %b %b | %b %b",A,B,Cin,Sum,Cout);

A=1;B=0;Cin=1;#10;
$display("%b %b %b | %b %b",A,B,Cin,Sum,Cout);

A=1;B=1;Cin=0;#10;
$display("%b %b %b | %b %b",A,B,Cin,Sum,Cout);

A=1;B=1;Cin=1;#10;
$display("%b %b %b | %b %b",A,B,Cin,Sum,Cout);

$finish;
end
endmodule

```

