## Verilog Code

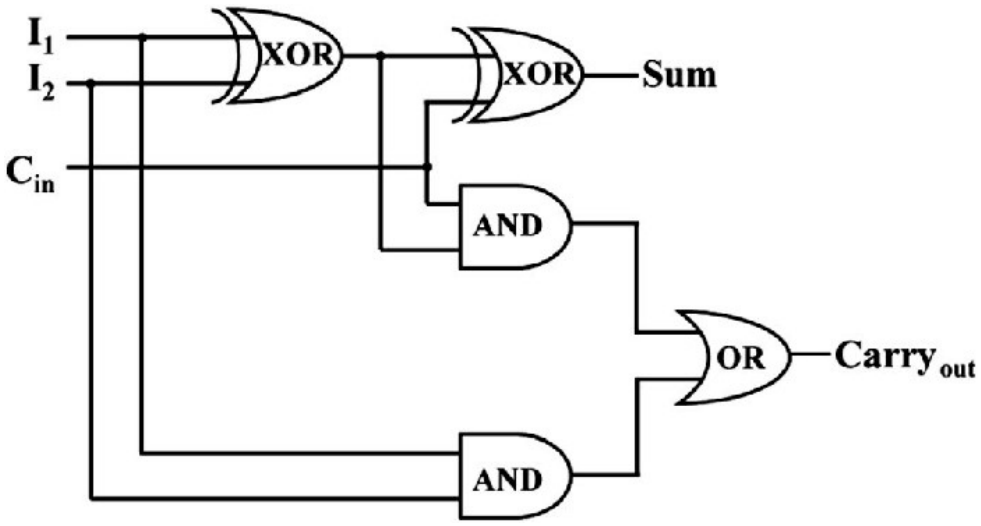
```
module full_adder (A, B, Cin, Sum, Cout);
  input A, B, Cin;
  output Sum, Cout;
  wire n1, n2, n3, n4;
  // XOR gates for Sum
  LIB_XOR2 U1 ( .A(A), .B(B),   .Z(n1) );
  LIB_XOR2 U2 ( .A(n1), .B(Cin), .Z(Sum) );
  // AND gates for Carry
  LIB_AND2 U3 ( .A(A), .B(B),   .Z(n2) );
  LIB_AND2 U4 ( .A(A), .B(Cin), .Z(n3) );
  LIB_AND2 U5 ( .A(B), .B(Cin), .Z(n4) );
  // OR gate for final carry
  LIB_OR3  U6 ( .A(n2), .B(n3), .C(n4), .Z(Cout) );
endmodule
```

## Test Bench Code

```
module tb_full_adder;

  reg A, B, Cin;
  wire Sum, Cout;

  full_adder dut (
    .A(A),
    .B(B),
    .Cin(Cin),
```

```
  .Sum(Sum),
  .Cout(Cout)
);

initial begin
  A=0; B=0; Cin=0; #10;
  A=0; B=0; Cin=1; #10;
  A=0; B=1; Cin=0; #10;
  A=0; B=1; Cin=1; #10;
  A=1; B=0; Cin=0; #10;
  A=1; B=0; Cin=1; #10;
  A=1; B=1; Cin=0; #10;
  A=1; B=1; Cin=1; #10;
  $finish;
end

endmodule
```

# Circuit



## Output Wave Forms