

Structural Modelling

```
module full_adder(  
    input A,B,Cin,  
    output Sum,Cout  
);
```

```
    wire w1,w2,w3;
```

```
    xor(w1,A,B);
```

```
    xor(Sum,w1,Cin);
```

```
    and(w2,Cin,w1);
```

```
    and(w3,A,B);
```

```
    or(Cout,w2,w3);
```

```
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");
```

```
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
```