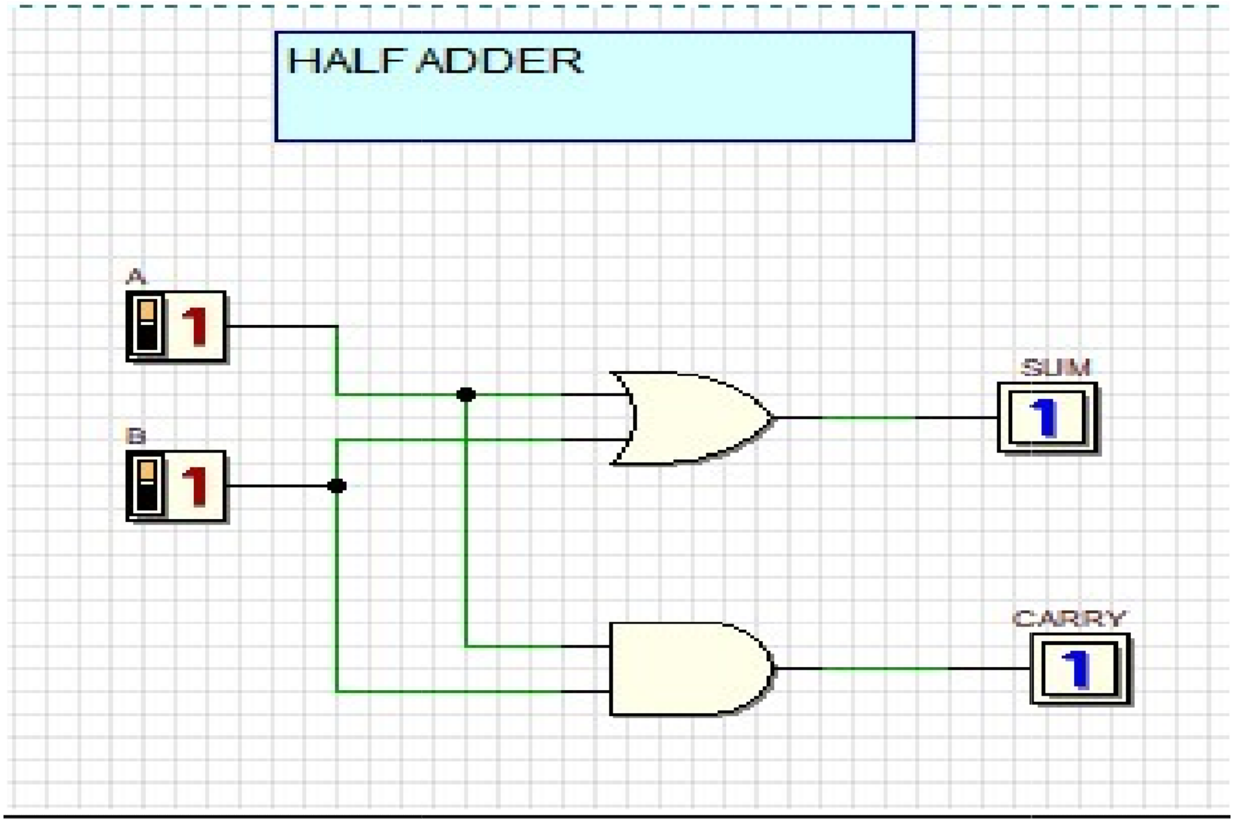
PROJECT NO. – 3

AIM : To design a half adder and full adder using basic gate and to verify its truth table .

APPARATUS : Input and output switches ; wires for connection ; OR ,AND, NOR Gates; DEEDS simulator

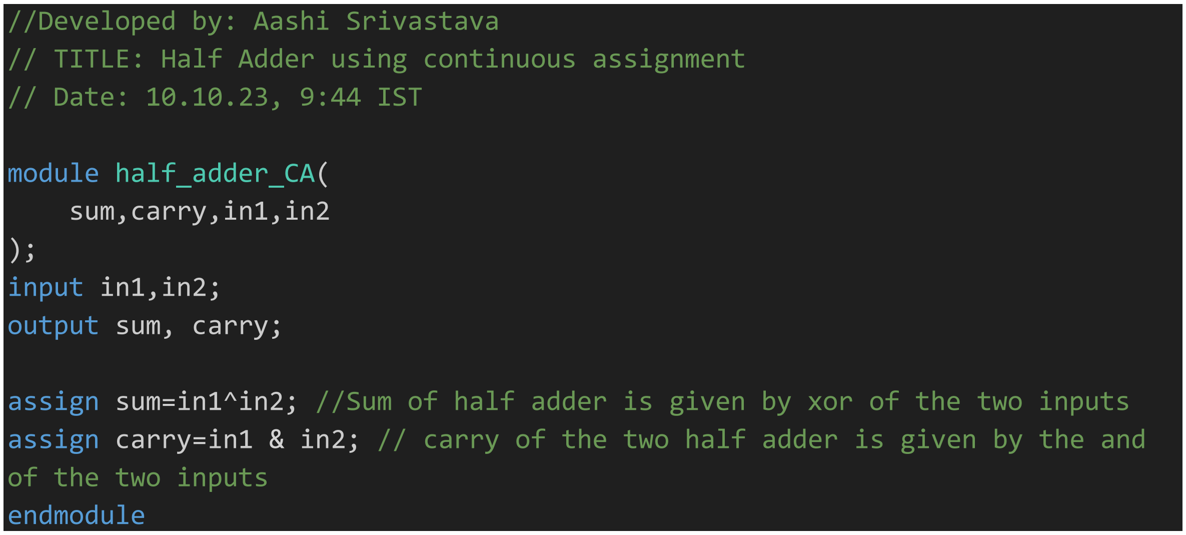
TRUTH TABLE FOR HALF ADDER :

INPUT		OUTPUT	
A	B	SUM (A+B)	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1


HALF ADDER
A
1
B
1
SUM
1
CARRY
1

Verilog code:

1. Using continuous assignment

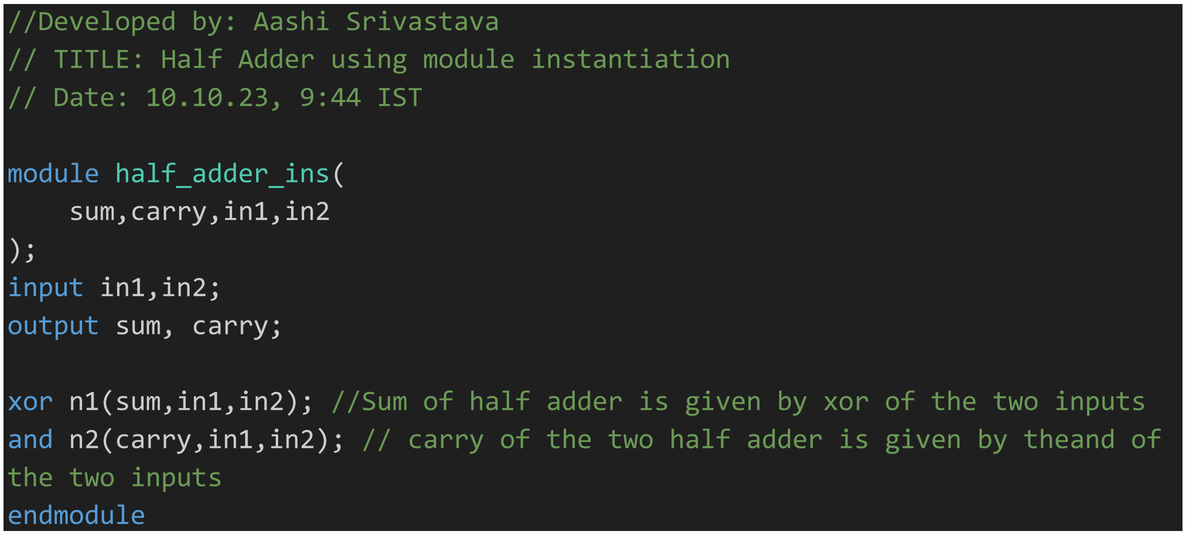


```verilog
//Developed by: Aashi Srivastava
// TITLE: Half Adder using continuous assignment
// Date: 10.10.23, 9:44 IST

module half_adder_CA(
    sum,carry,in1,in2
);
input in1,in2;
output sum, carry;

assign sum=in1^in2; //Sum of half adder is given by xor of the two inputs
assign carry=in1 & in2; // carry of the two half adder is given by the and of the two inputs
endmodule
```

2. Using Module instantiation

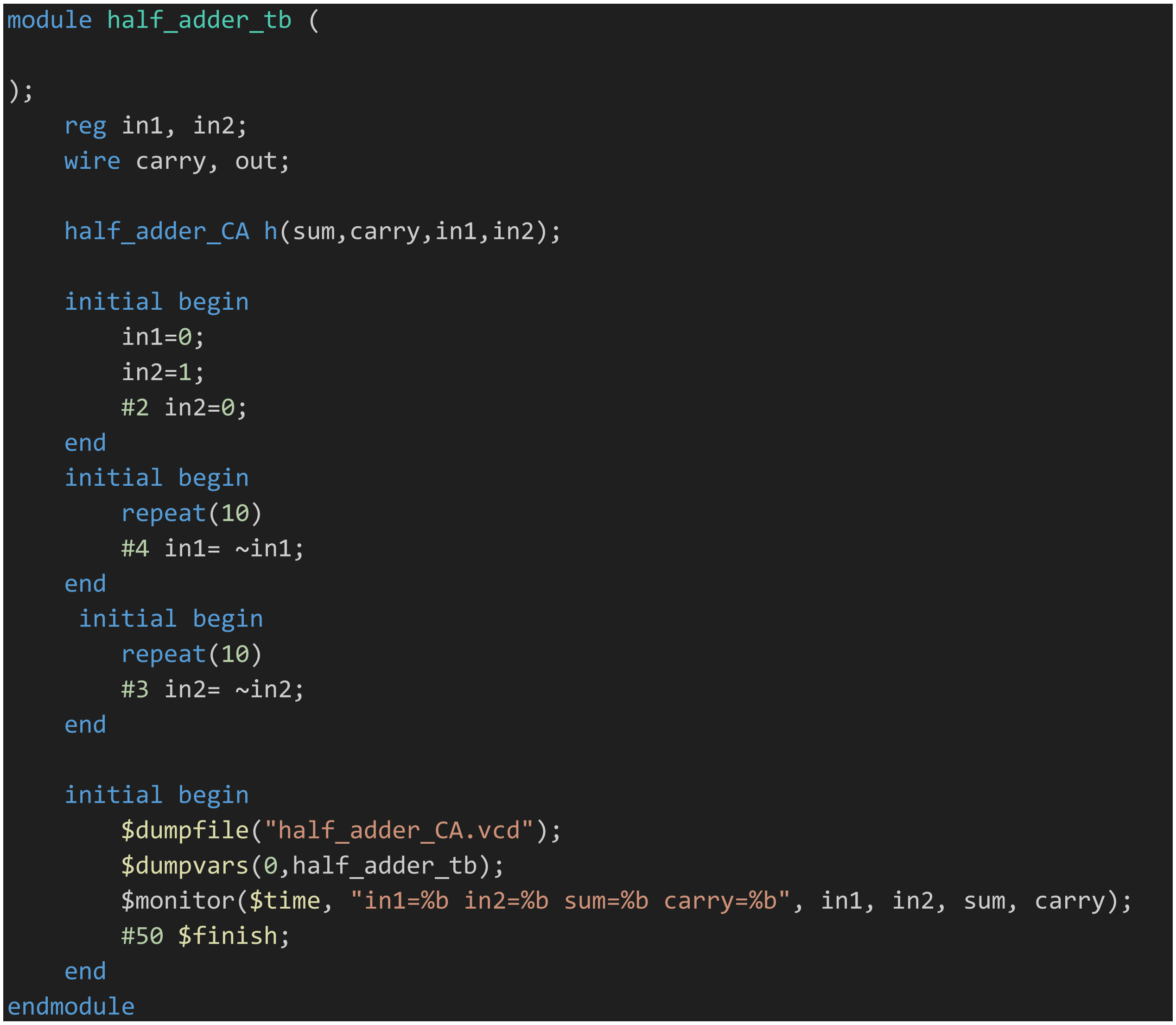


```verilog
//Developed by: Aashi Srivastava
// TITLE: Half Adder using module instantiation
// Date: 10.10.23, 9:44 IST

module half_adder_ins(
    sum,carry,in1,in2
);
input in1,in2;
output sum, carry;

xor n1(sum,in1,in2); //Sum of half adder is given by xor of the two inputs
and n2(carry,in1,in2); // carry of the two half adder is given by theand of the two inputs
endmodule
```

TEST BENCH:



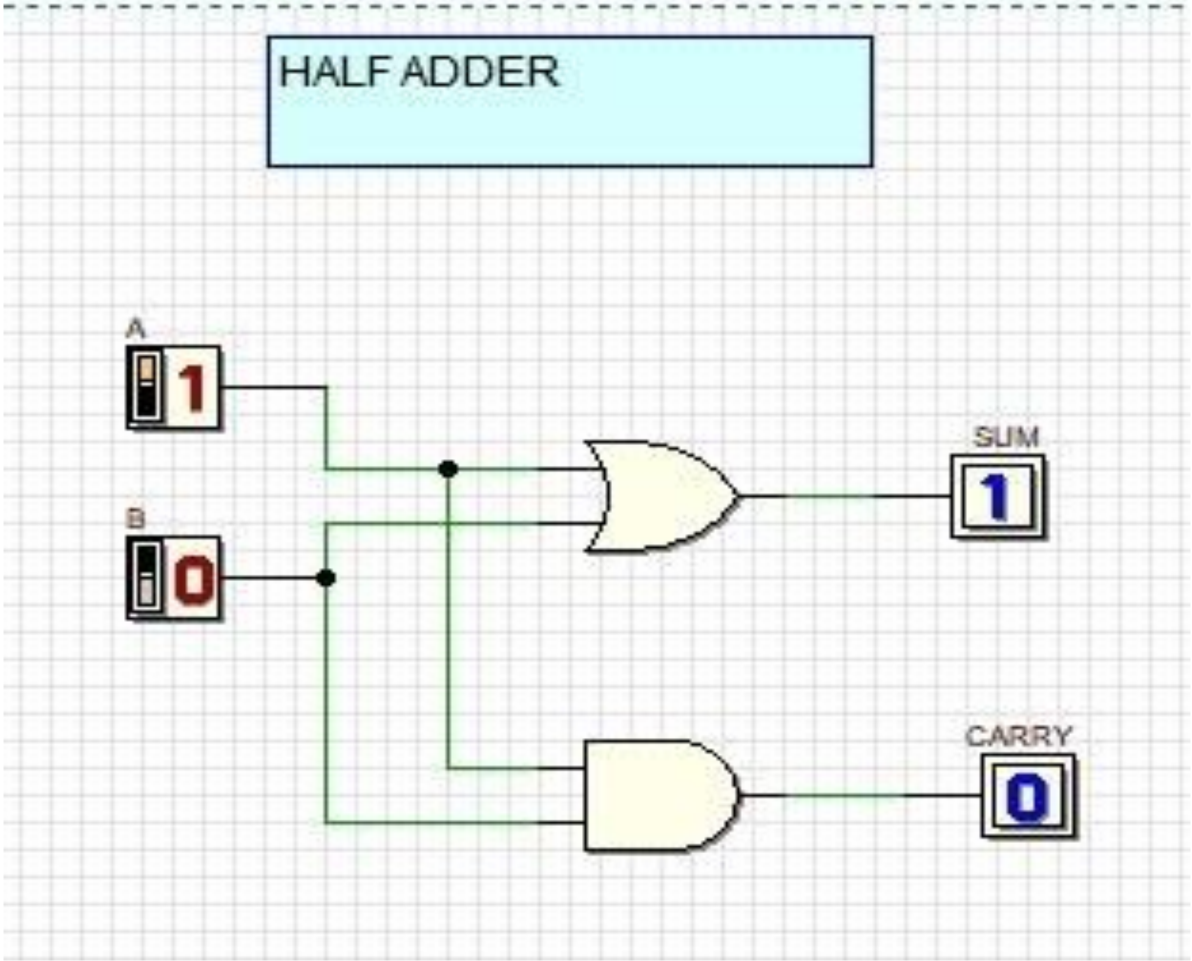
```
module half_adder_tb (

);
    reg in1, in2;
    wire carry, out;

    half_adder_CA h(sum,carry,in1,in2);

    initial begin
        in1=0;
        in2=1;
        #2 in2=0;
    end
    initial begin
        repeat(10)
        #4 in1= ~in1;
    end
     initial begin
        repeat(10)
        #3 in2= ~in2;
    end

    initial begin
        $dumpfile("half_adder_CA.vcd");
        $dumpvars(0,half_adder_tb);
        $monitor($time, "in1=%b in2=%b sum=%b carry=%b", in1, in2, sum, carry);
        #50 $finish;
    end
endmodule
```


HALF ADDER
A
1
B
0
SUM
1
CARRY
0

TRUTH TABLE FOR FULL ADDER

INPUT			OUTPUT	
A	B	C	SUM (A+B+C)	CARRY
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

KMAP FOR FULL ADDER (SUM)

		B’*(C)’	B’*(C)	B*(C)	B*(C)’	
A’		0	1	0	1	
A		1	0	1	0	

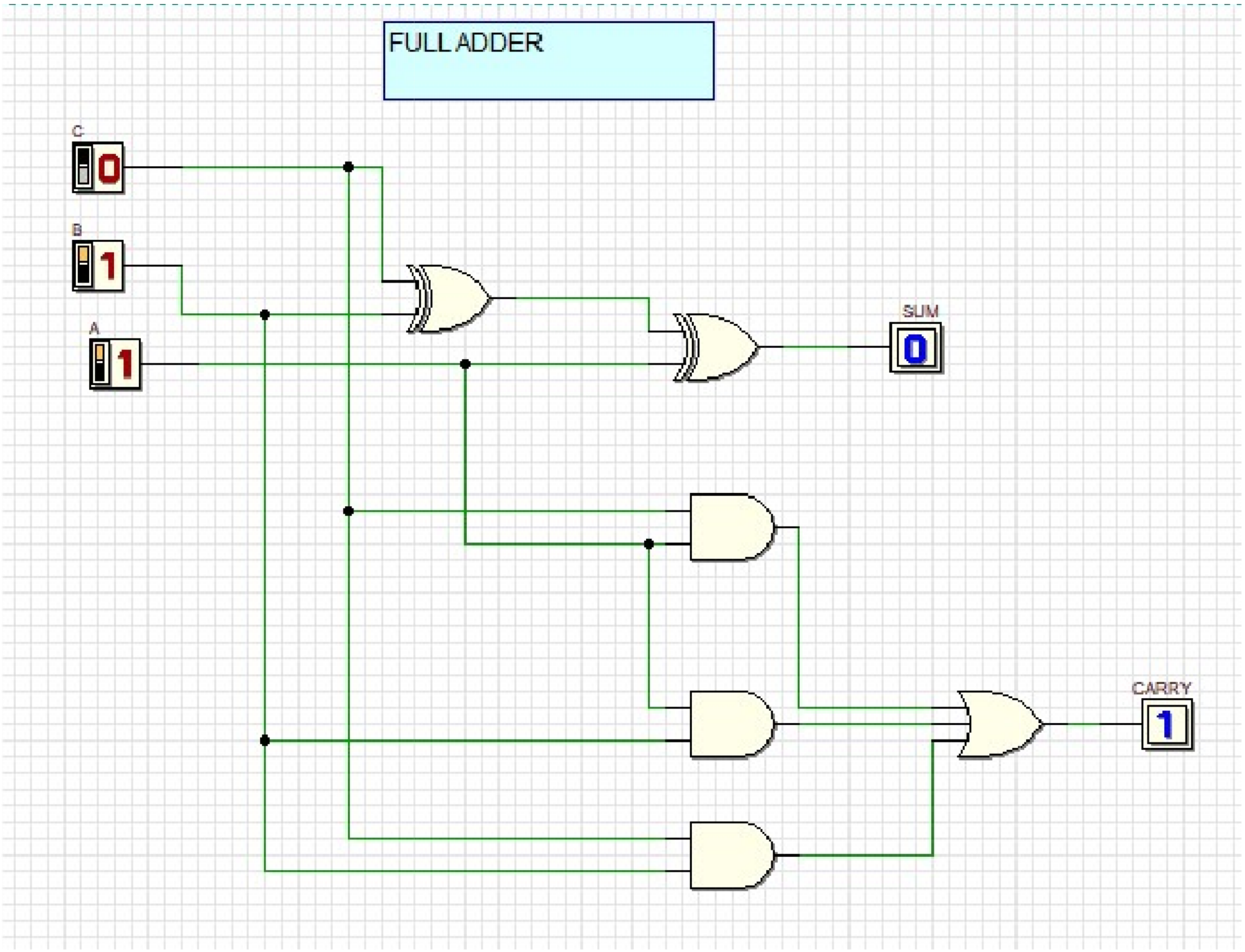
SIMPLIFING K MAP

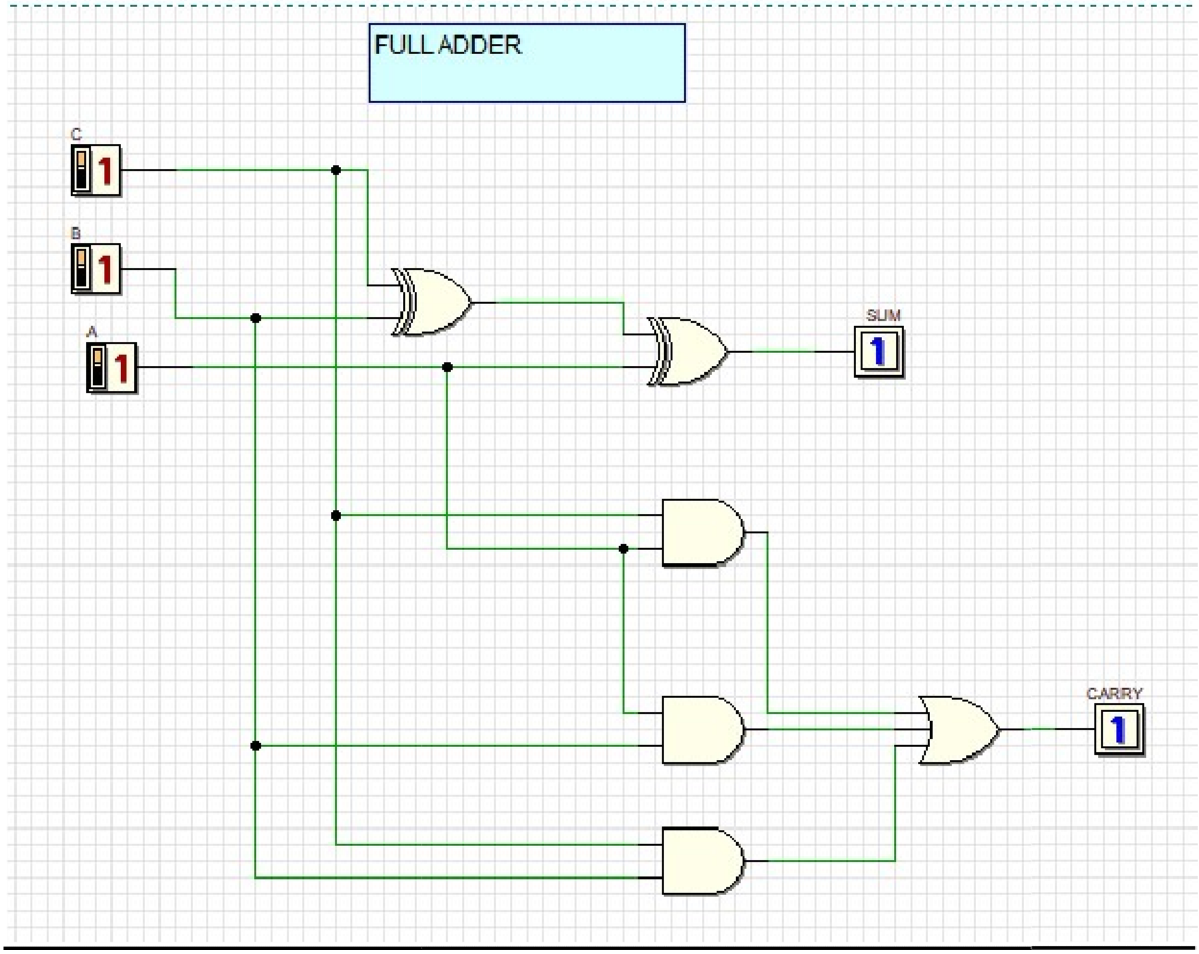
SUM = (A ‘)*(B’*(C)+ B*(C)’) + A*(B’*(C)’+ B*(C)) .

$=A \oplus B \oplus C$

KMAP FOR FULL SUBTRACTOR (CARRY)

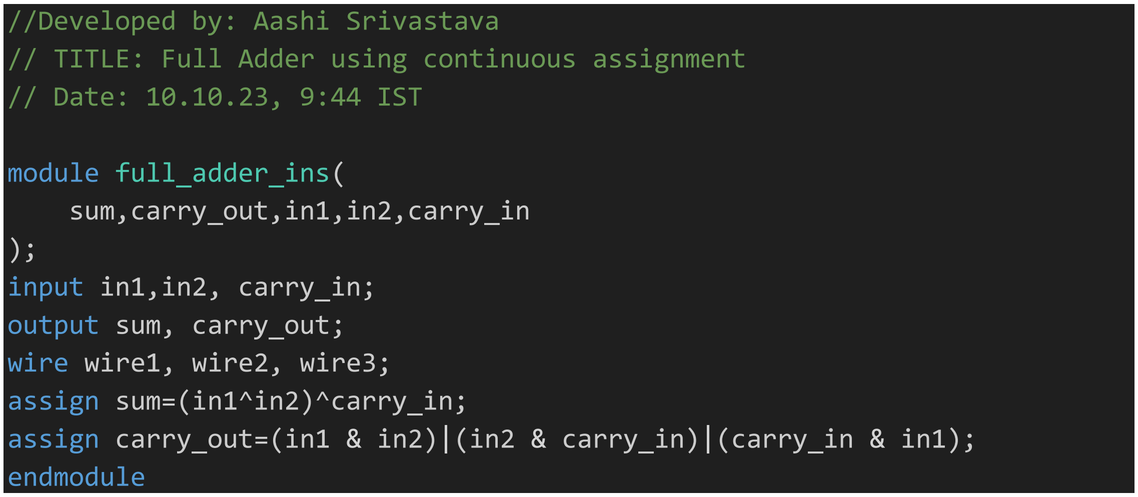
		B’*(C)’	B’*C	B*C	B*(C)’	
A’		0	0	1	0	
A		0	1	1	1	


FULL ADDER
C
0
B
1
A
1
SUM
0
CARRY
1


FULL ADDER
C
1
B
1
A
1
SUM
1
CARRY
1

Verilog code:

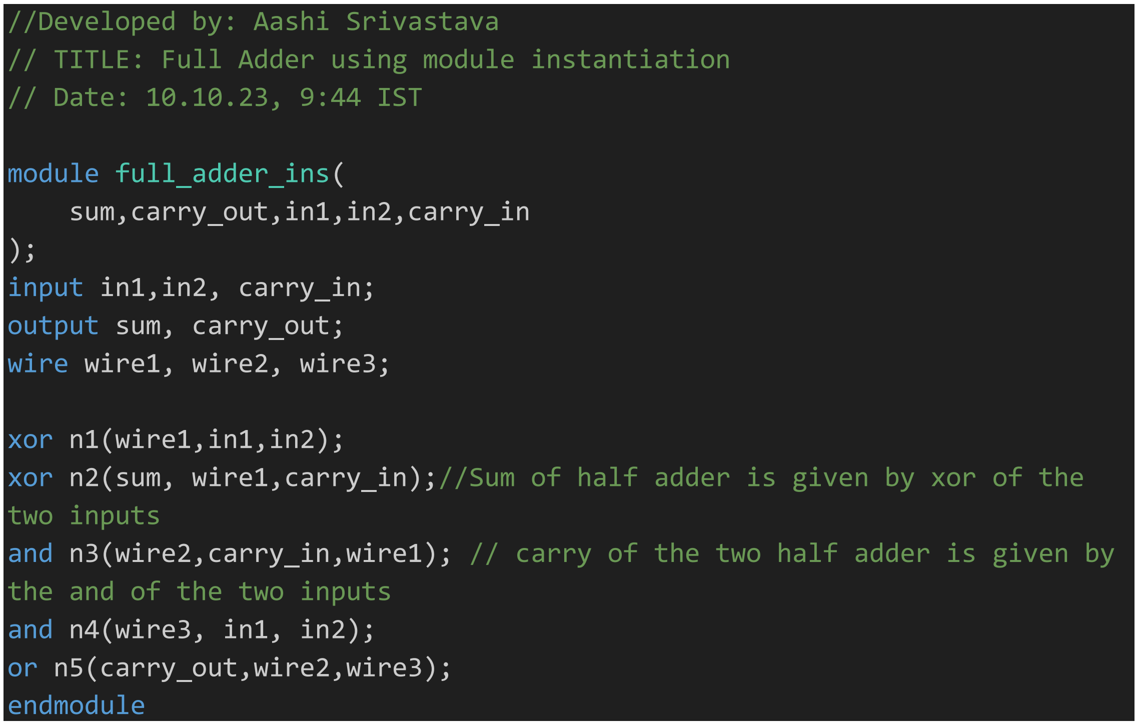
1. Using continuous assignment



```
//Developed by: Aashi Srivastava
// TITLE: Full Adder using continuous assignment
// Date: 10.10.23, 9:44 IST

module full_adder_ins(
    sum,carry_out,in1,in2,carry_in
);
input in1,in2, carry_in;
output sum, carry_out;
wire wire1, wire2, wire3;
assign sum=(in1^in2)^carry_in;
assign carry_out=(in1 & in2)|(in2 & carry_in)|(carry_in & in1);
endmodule
```

2. Using Module instantiation

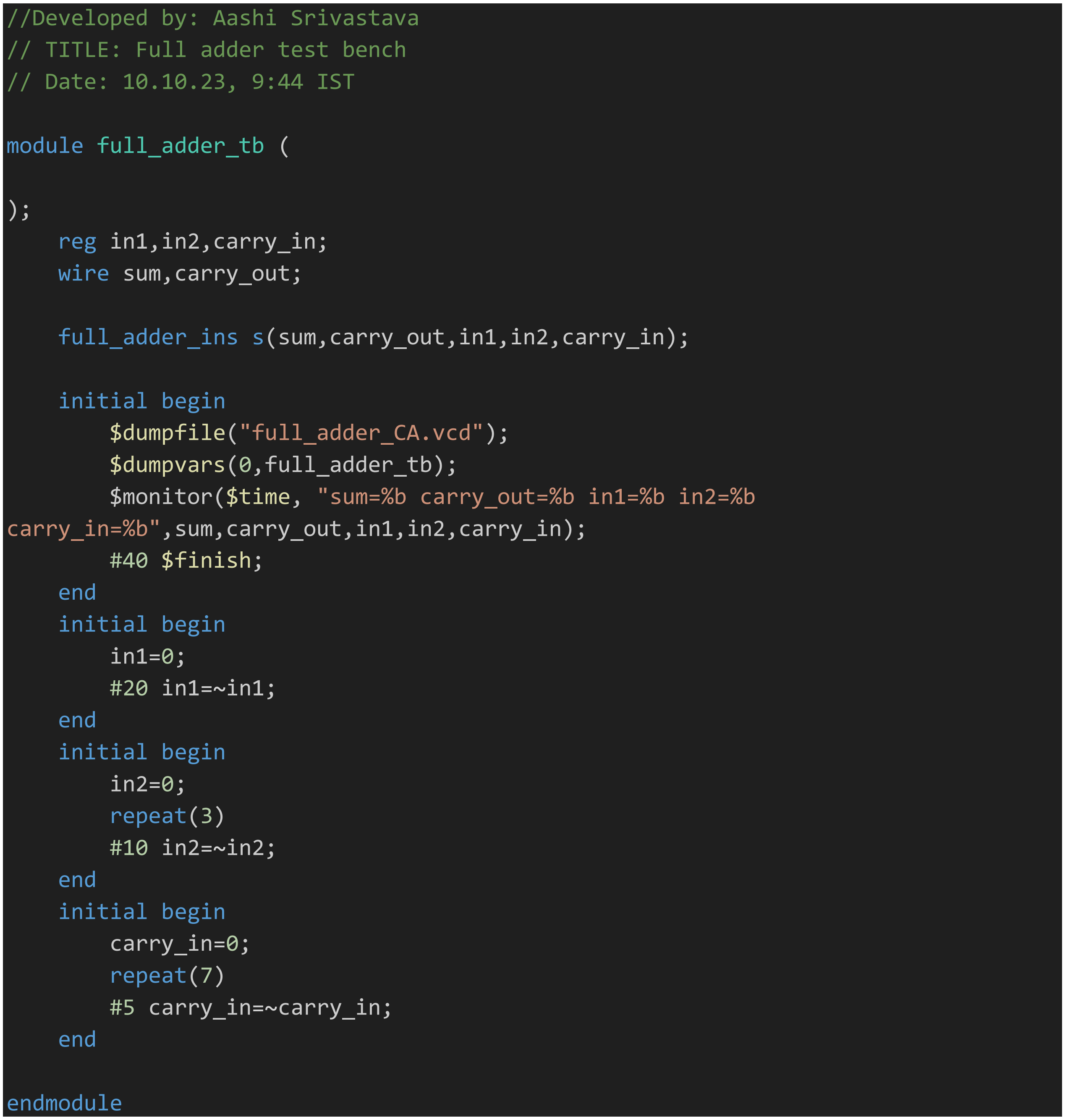


```
//Developed by: Aashi Srivastava
// TITLE: Full Adder using module instantiation
// Date: 10.10.23, 9:44 IST

module full_adder_ins(
    sum,carry_out,in1,in2,carry_in
);
input in1,in2, carry_in;
output sum, carry_out;
wire wire1, wire2, wire3;

xor n1(wire1,in1,in2);
xor n2(sum, wire1,carry_in);//Sum of half adder is given by xor of the
two inputs
and n3(wire2,carry_in,wire1); // carry of the two half adder is given by
the and of the two inputs
and n4(wire3, in1, in2);
or n5(carry_out,wire2,wire3);
endmodule
```

TEST BENCH:



```
//Developed by: Aashi Srivastava
// TITLE: Full adder test bench
// Date: 10.10.23, 9:44 IST

module full_adder_tb (

);
    reg in1,in2,carry_in;
    wire sum,carry_out;

    full_adder_ins s(sum,carry_out,in1,in2,carry_in);

    initial begin
        $dumpfile("full_adder_CA.vcd");
        $dumpvars(0,full_adder_tb);
        $monitor($time, "sum=%b carry_out=%b in1=%b in2=%b
carry_in=%b",sum,carry_out,in1,in2,carry_in);
        #40 $finish;
    end
    initial begin
        in1=0;
        #20 in1=~in1;
    end
    initial begin
        in2=0;
        repeat(3)
        #10 in2=~in2;
    end
    initial begin
        carry_in=0;
        repeat(7)
        #5 carry_in=~carry_in;
    end

endmodule
```