**VERILOG CODE**

**Adder:** Adder are use to perform the binary addition of two binary numbers. Adders

are used for signed or unsigned addition operations.

**Half Adder :** Half adder is a combinational circuit, which performs the addition of two bits A

and B are of single bit numbers. It produces two outputs sum & carry as Outputs.

Block diagram:

****

**Truth Table:**

****

**BOOLEAN EXPRESSIONS :**

SUM = A ^ B = AB`+A`B

CARRY = A & B = AB

Code:

module day1\_hf( input a, input b,

output reg sum ,

output reg carry

);

always@(a,b)

case({a,b})

2'b00:begin sum = 0;carry = 0;end

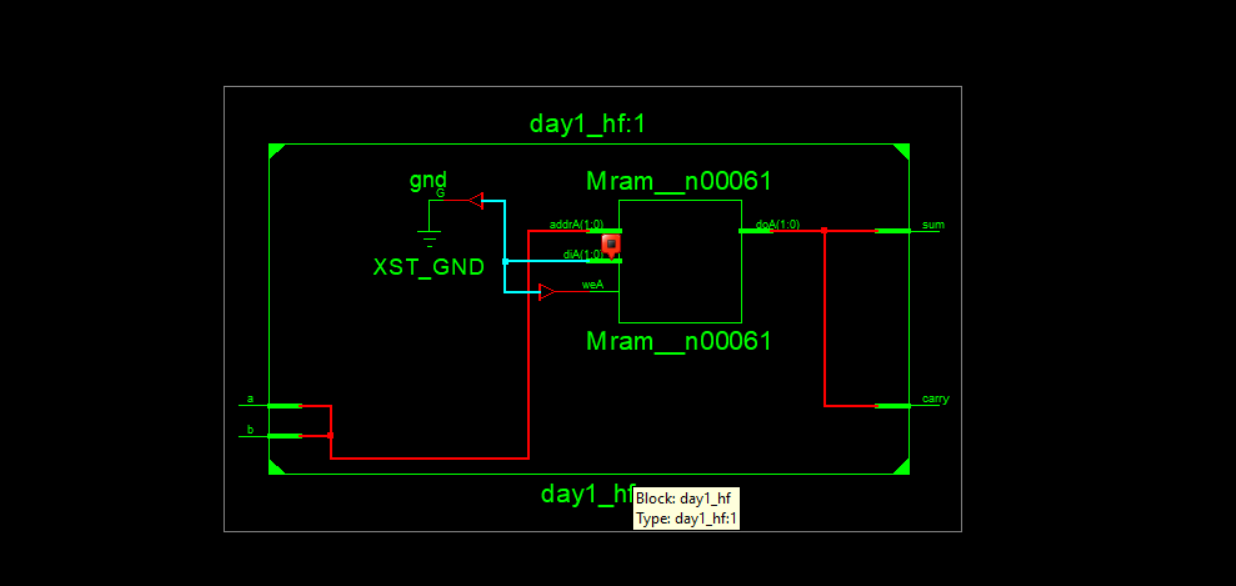
2'b01:begin sum = 1;carry = 0;end

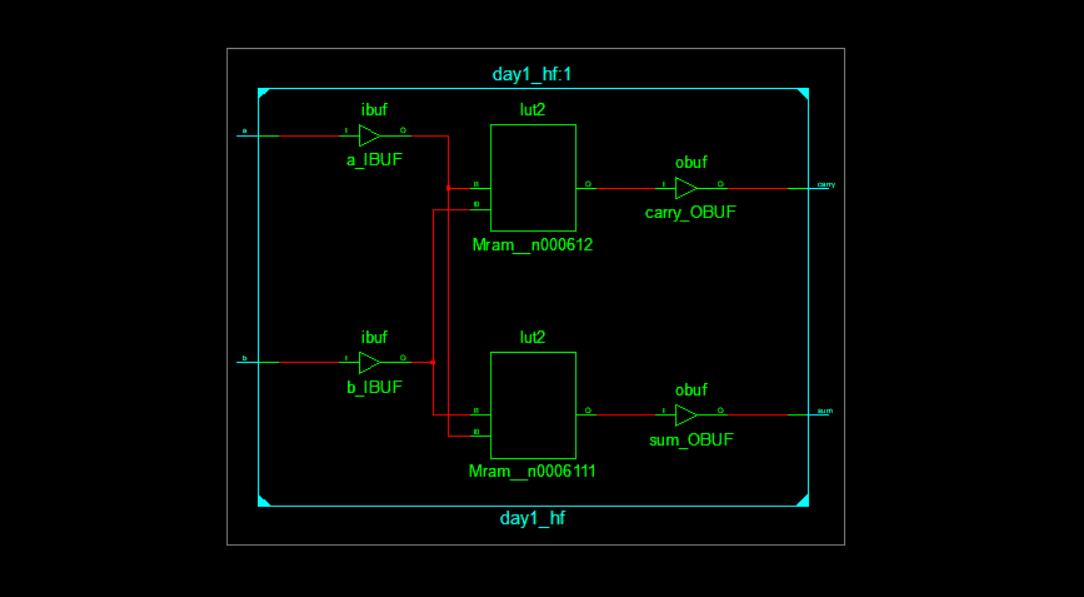
2'b10:begin sum = 1;carry = 0;end

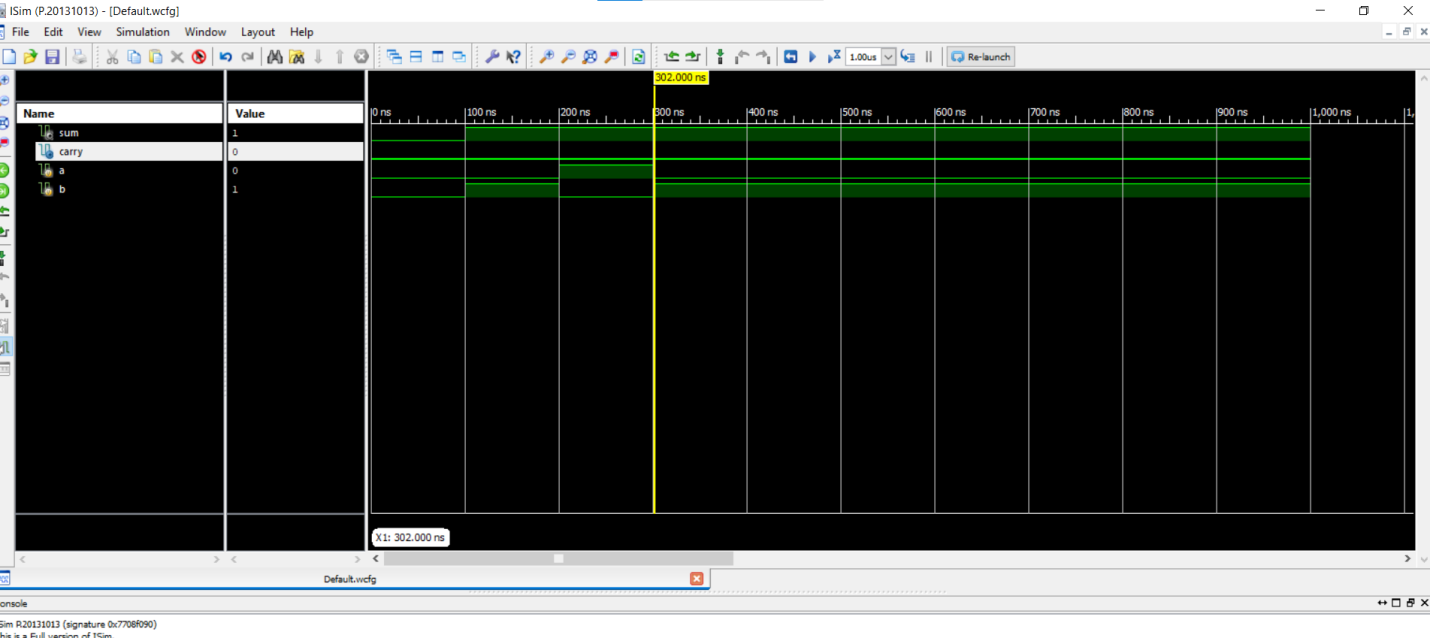
2'b11:begin sum = 01;carry = 1;end

endcase

endmodule







Full adder:

A full adder is a combinational logic circuit that forms the arithmetic sum of three bits. it consists of three inputs and two outputs. Which performs the addition of three bits A ,B and Carrry and It produces sum & carry as Outputs. Half adder have no scope of adding the carry bit ,to overcome this drawback, full adder comes into play.

**Block diagram:**



**BOOLEAN EXPRESSIONS :**

SUM = A ^ B^C

CARRY = A & B| B & C | A & C

Code: module day1\_FA1(input a,

input b,

input c,

output reg sum,

output reg carry

);

always@(a,b,c)

case({a,b,c})

2'b000:begin sum = 0; carry = 0; end

2'b001:begin sum = 1; carry = 0; end

2'b010:begin sum = 1; carry = 0; end

2'b011:begin sum = 0; carry = 1; end

2'b100:begin sum = 1; carry = 0; end

2'b110:begin sum = 0; carry = 1; end

2'b111:begin sum = 1; carry = 1; end

endcase

endmodule

