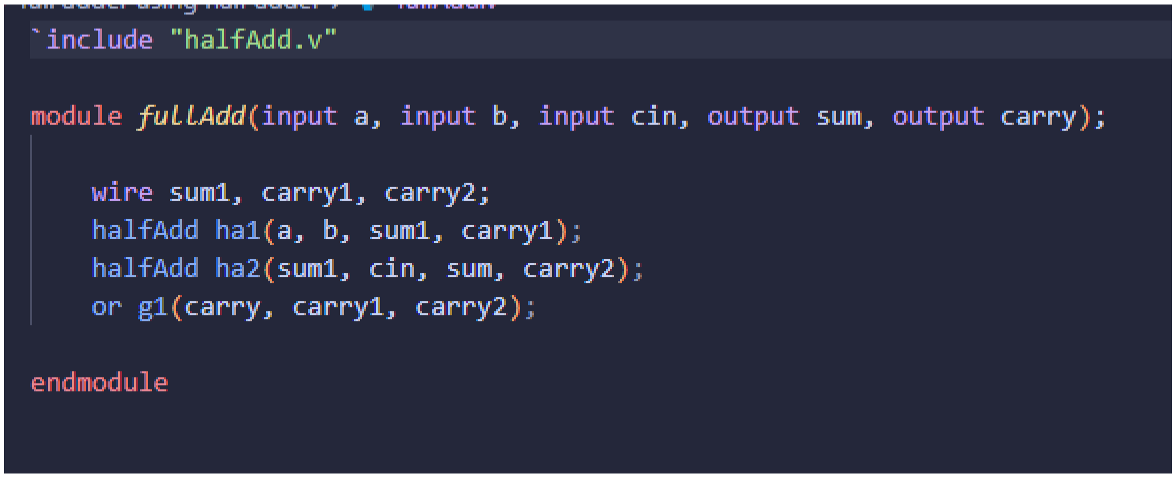
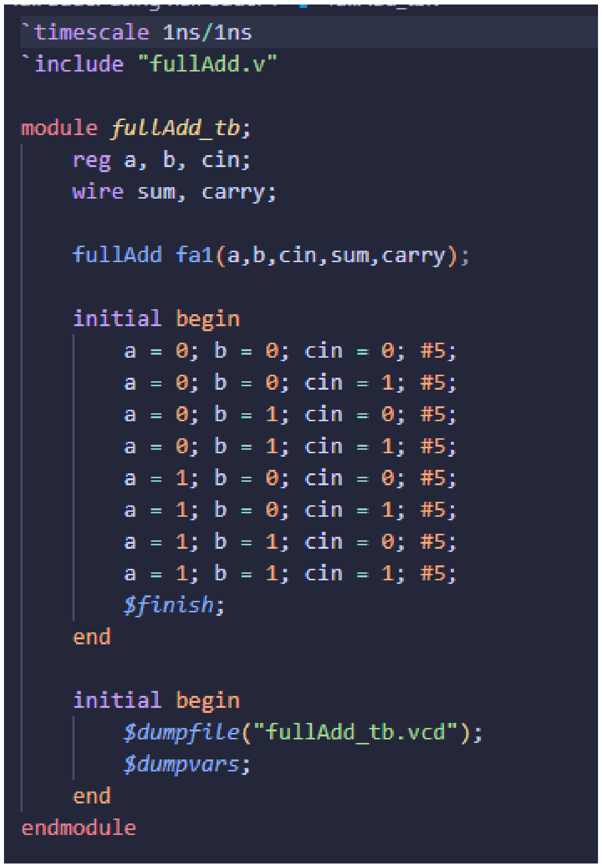
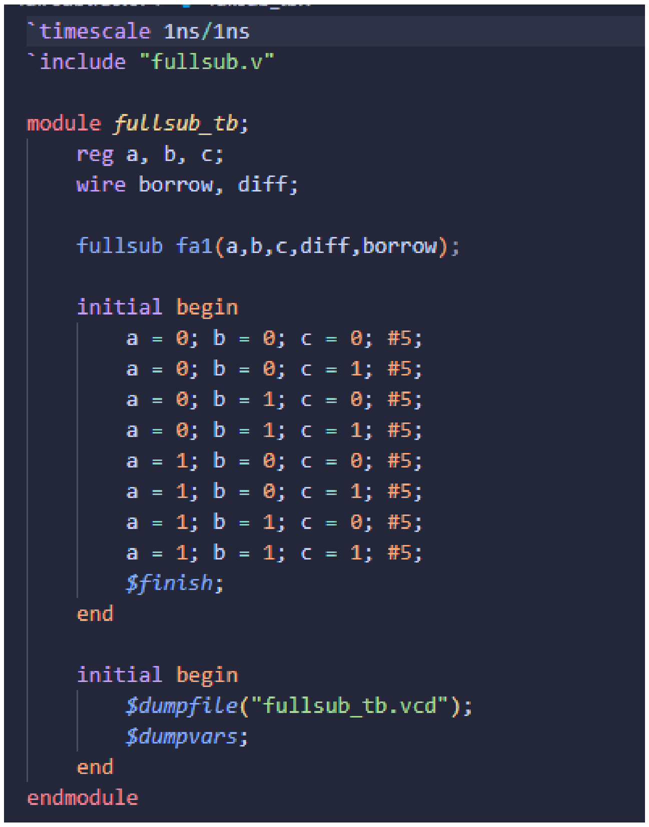


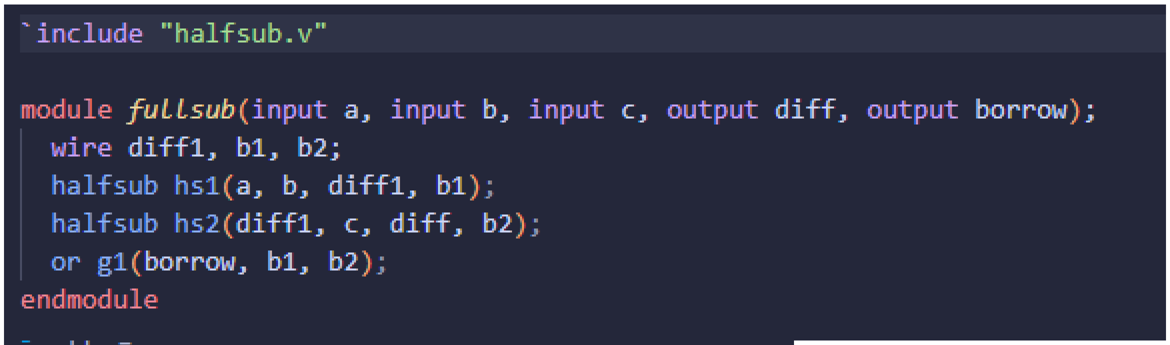
```verilog
`include "halfAdd.v"

module fullAdd(input a, input b, input cin, output sum, output carry);

    wire sum1, carry1, carry2;
    halfAdd ha1(a, b, sum1, carry1);
    halfAdd ha2(sum1, cin, sum, carry2);
    or g1(carry, carry1, carry2);

endmodule
```



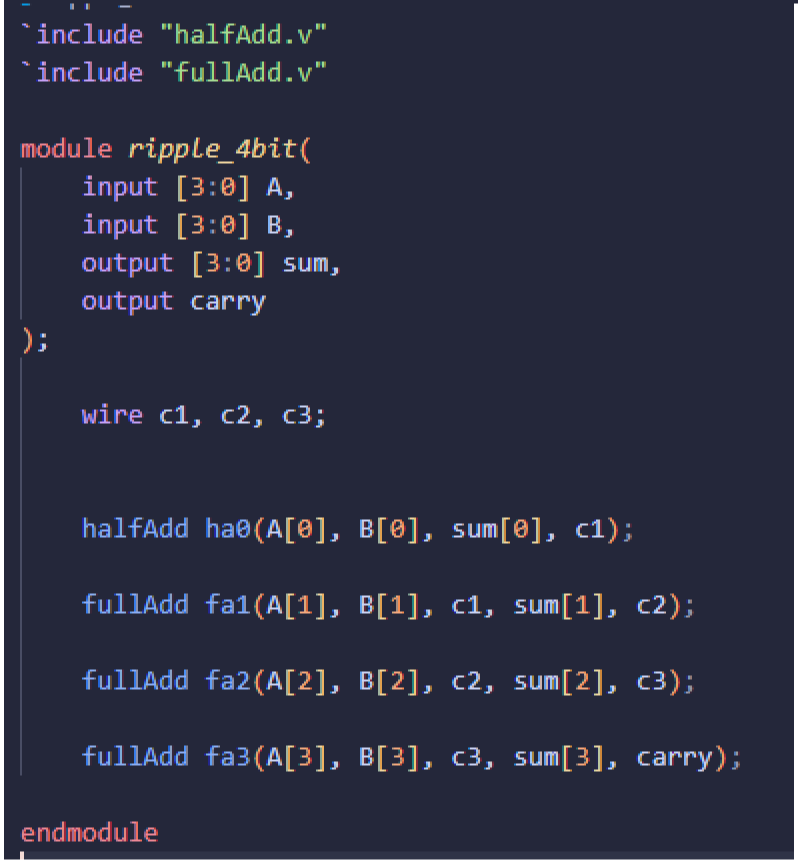
```verilog
`timescale 1ns/1ns
`include "fullAdd.v"

module fullAdd_tb;
    reg a, b, cin;
    wire sum, carry;

    fullAdd fa1(a,b,cin,sum,carry);

    initial begin
        a = 0; b = 0; cin = 0; #5;
        a = 0; b = 0; cin = 1; #5;
        a = 0; b = 1; cin = 0; #5;
        a = 0; b = 1; cin = 1; #5;
        a = 1; b = 0; cin = 0; #5;
        a = 1; b = 0; cin = 1; #5;
        a = 1; b = 1; cin = 0; #5;
        a = 1; b = 1; cin = 1; #5;
        $finish;
    end

    initial begin
        $dumpfile("fullAdd_tb.vcd");
        $dumpvars;
    end
endmodule
```



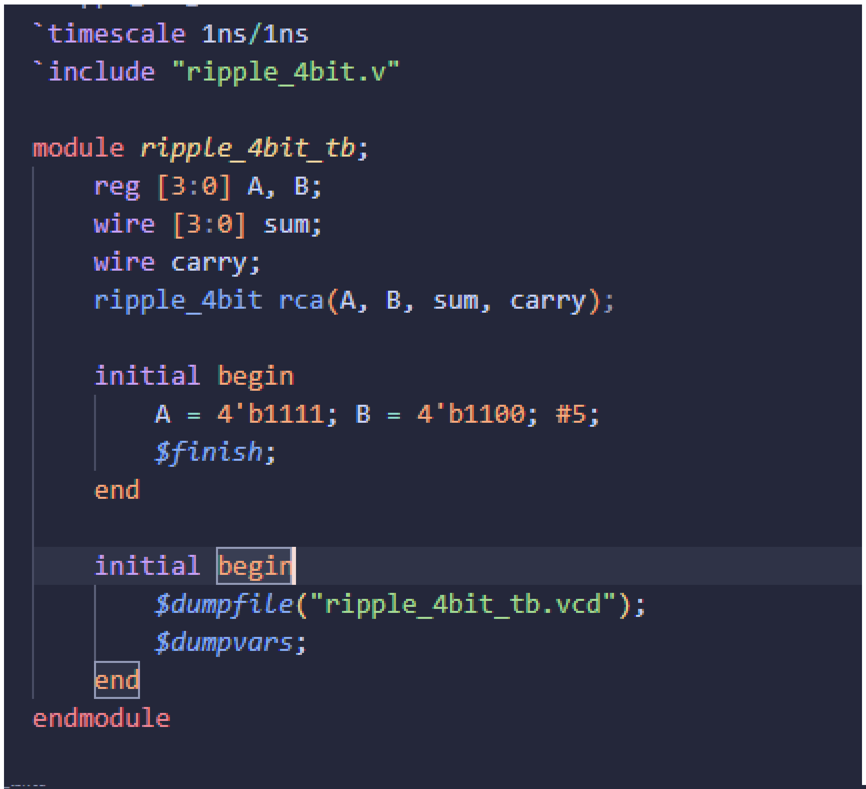
```verilog
`timescale 1ns/1ns
`include "fullsub.v"

module fullsub_tb;
    reg a, b, c;
    wire borrow, diff;

    fullsub fa1(a,b,c,diff,borrow);

    initial begin
        a = 0; b = 0; c = 0; #5;
        a = 0; b = 0; c = 1; #5;
        a = 0; b = 1; c = 0; #5;
        a = 0; b = 1; c = 1; #5;
        a = 1; b = 0; c = 0; #5;
        a = 1; b = 0; c = 1; #5;
        a = 1; b = 1; c = 0; #5;
        a = 1; b = 1; c = 1; #5;
        $finish;
    end

    initial begin
        $dumpfile("fullsub_tb.vcd");
        $dumpvars;
    end
endmodule
```



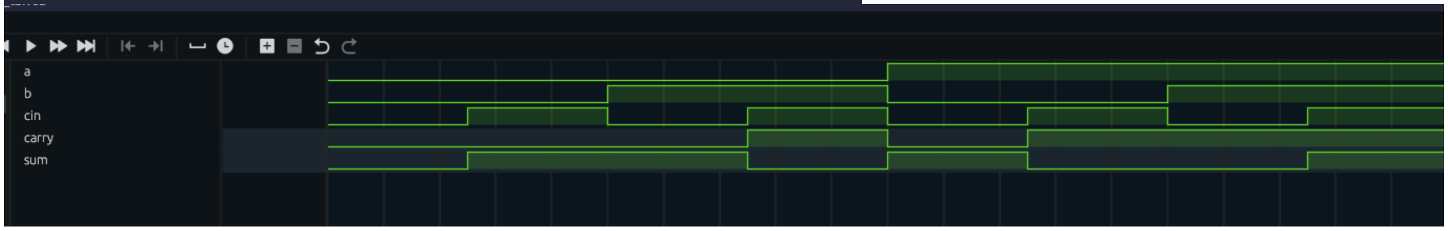
```verilog
`include "halfsub.v"

module fullsub(input a, input b, input c, output diff, output borrow);
  wire diff1, b1, b2;
  halfsub hs1(a, b, diff1, b1);
  halfsub hs2(diff1, c, diff, b2);
  or g1(borrow, b1, b2);
endmodule
```

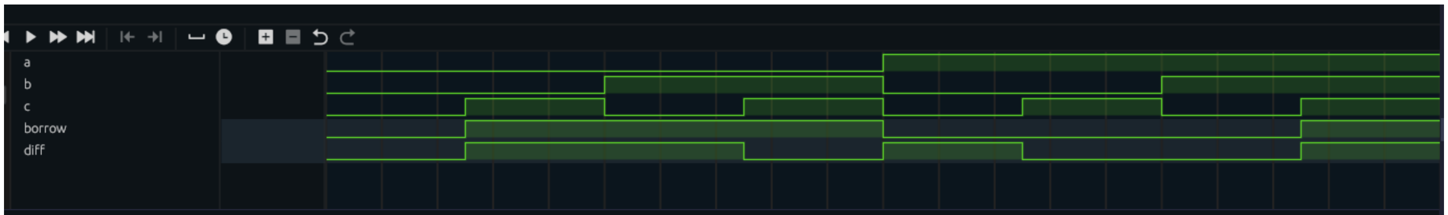


```verilog
`include "halfAdd.v"
`include "fullAdd.v"

module ripple_4bit(
    input [3:0] A,
    input [3:0] B,
    output [3:0] sum,
    output carry
);

    wire c1, c2, c3;


    halfAdd ha0(A[0], B[0], sum[0], c1);

    fullAdd fa1(A[1], B[1], c1, sum[1], c2);

    fullAdd fa2(A[2], B[2], c2, sum[2], c3);

    fullAdd fa3(A[3], B[3], c3, sum[3], carry);

endmodule
```



```
`timescale 1ns/1ns
`include "ripple_4bit.v"

module ripple_4bit_tb;
    reg [3:0] A, B;
    wire [3:0] sum;
    wire carry;
    ripple_4bit rca(A, B, sum, carry);

    initial begin
        A = 4'b1111; B = 4'b1100; #5;
        $finish;
    end

    initial begin
        $dumpfile("ripple_4bit_tb.vcd");
        $dumpvars;
    end
endmodule
```


a
b
cin
carry
sum


a
b
c
borrow
diff