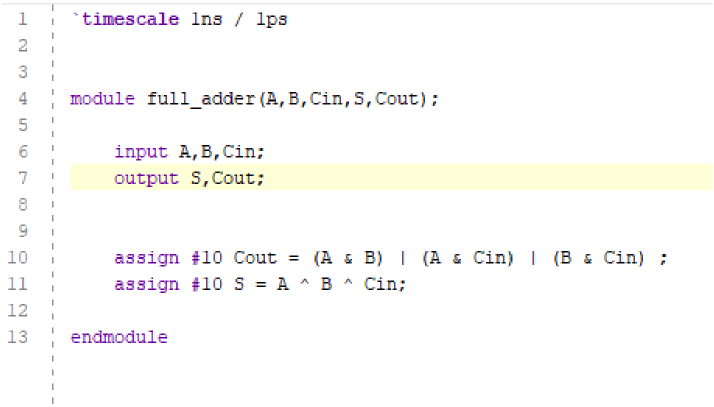
Experiment 6: Serial Adder

İbrahim Burak Tanrıkulu 21827852

Şerife Ebru Yardım 21785784

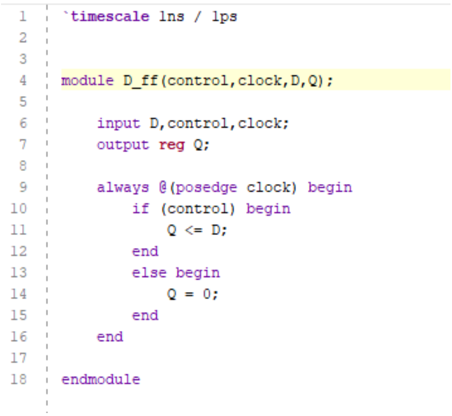
CODES:

Full adder:



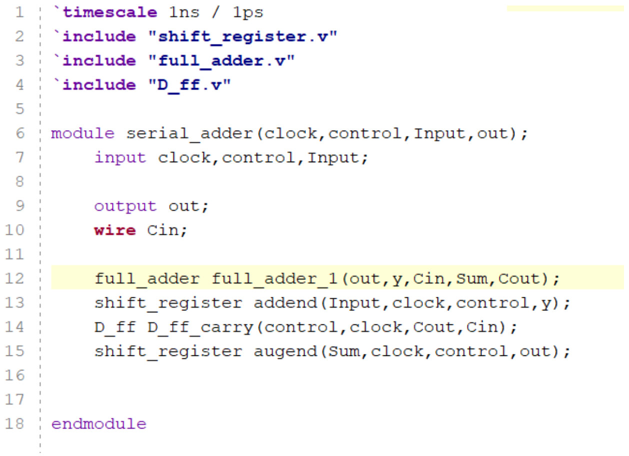
```
`timescale 1ns / 1ps


module full_adder(A,B,Cin,S,Cout);

    input A,B,Cin;
    output S,Cout;


    assign #10 Cout = (A & B) | (A & Cin) | (B & Cin) ;
    assign #10 S = A ^ B ^ Cin;

endmodule
```

D flip flop:



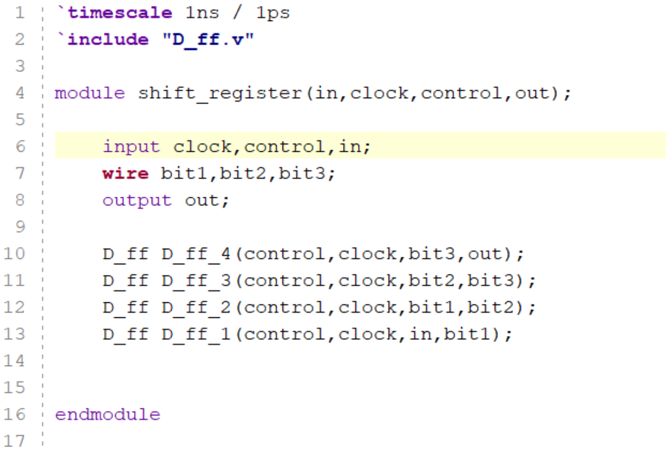
```
`timescale 1ns / 1ps


module D_ff(control,clock,D,Q);

    input D,control,clock;
    output reg Q;

    always @(posedge clock) begin
        if (control) begin
            Q <= D;
        end
        else begin
            Q = 0;
        end
    end

endmodule
```

Serial adder:



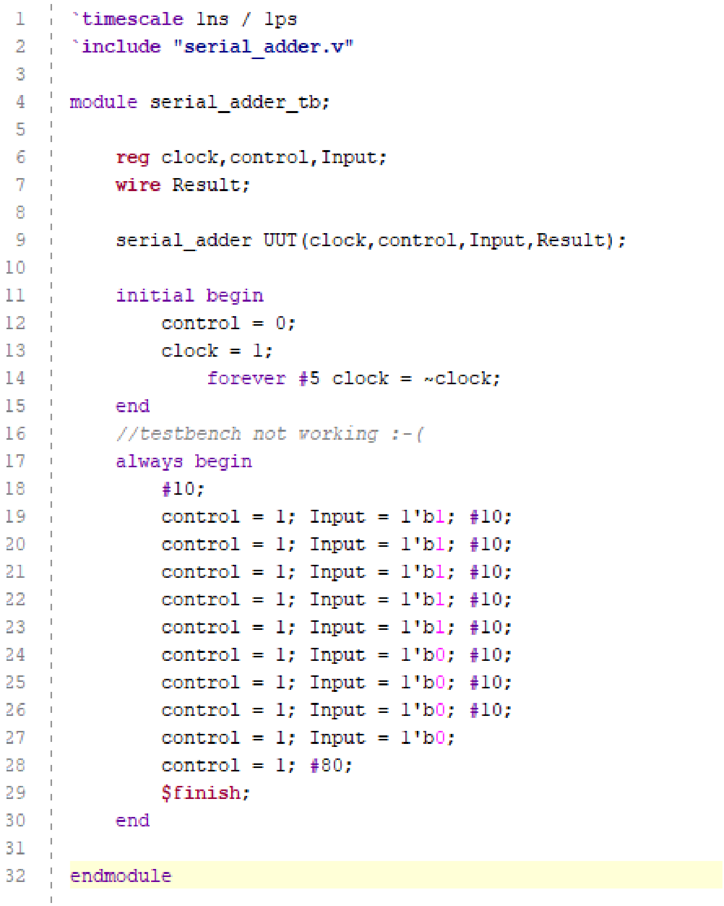
```
`timescale 1ns / 1ps
`include "shift_register.v"
`include "full_adder.v"
`include "D_ff.v"

module serial_adder(clock,control,Input,out);
    input clock,control,Input;

    output out;
    wire Cin;

    full_adder full_adder_1(out,y,Cin,Sum,Cout);
    shift_register addend(Input,clock,control,y);
    D_ff D_ff_carry(control,clock,Cout,Cin);
    shift_register augend(Sum,clock,control,out);


endmodule
```

Shift register:

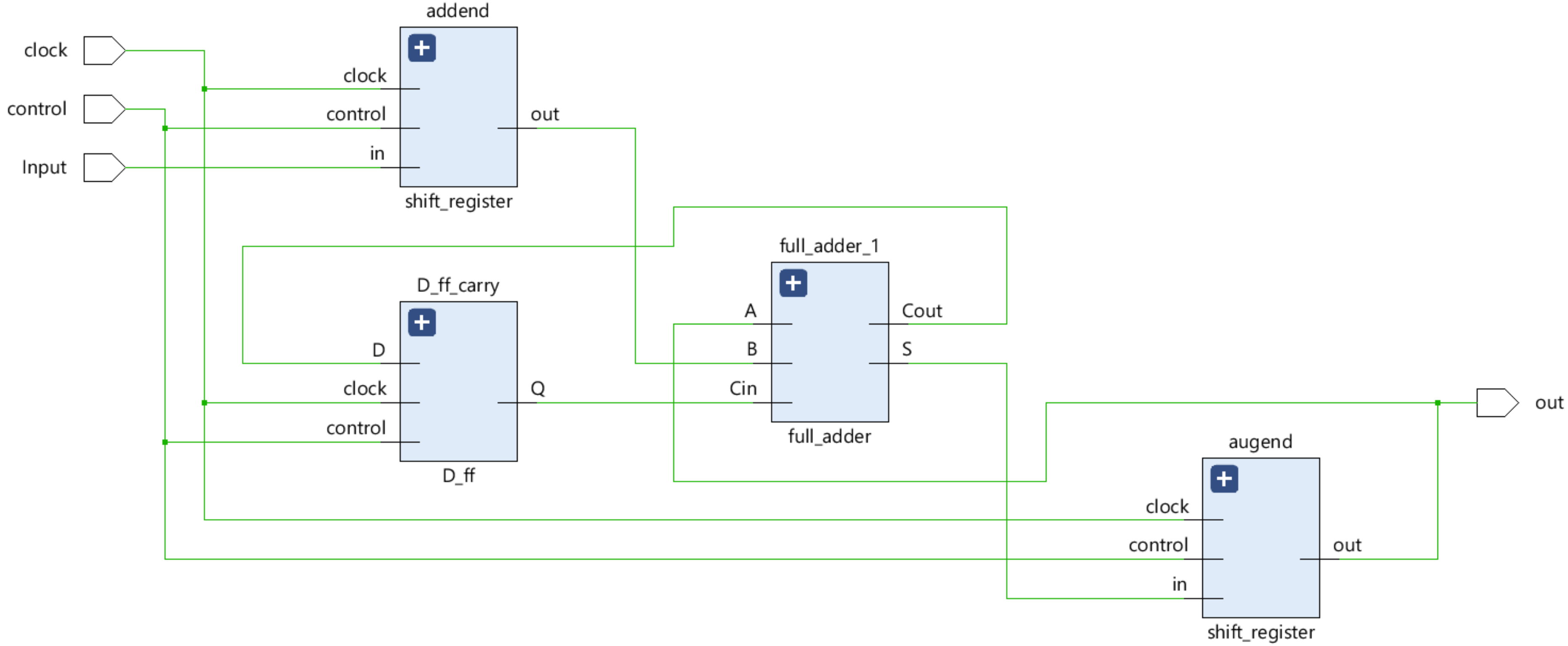


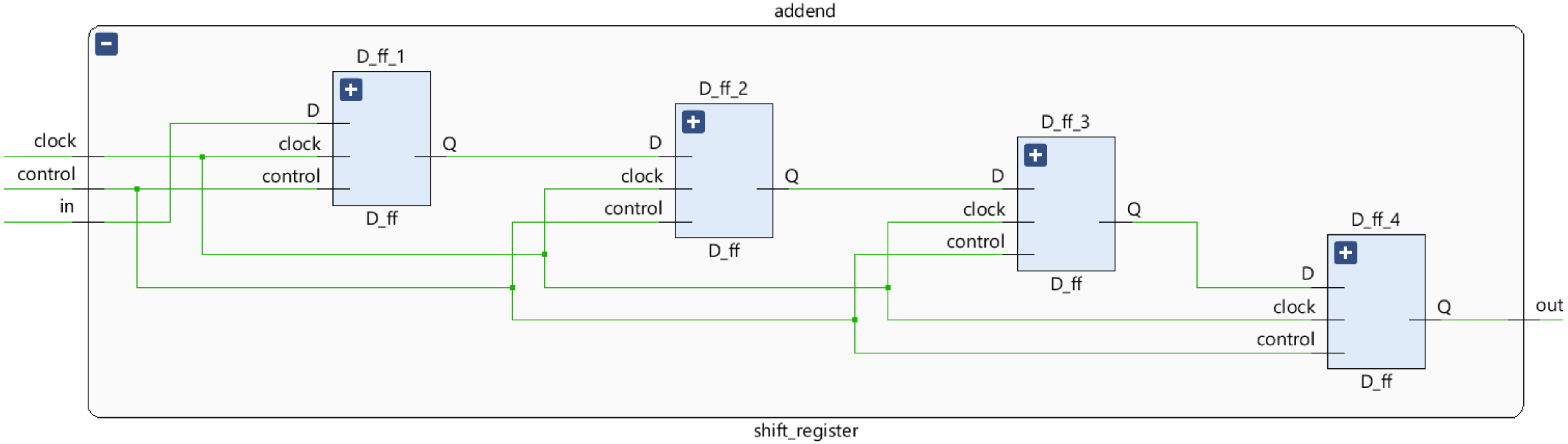
```
`timescale 1ns / 1ps
`include "D_ff.v"

module shift_register(in,clock,control,out);

    input clock,control,in;
    wire bit1,bit2,bit3;
    output out;

    D_ff D_ff_4(control,clock,bit3,out);
    D_ff D_ff_3(control,clock,bit2,bit3);
    D_ff D_ff_2(control,clock,bit1,bit2);
    D_ff D_ff_1(control,clock,in,bit1);


endmodule
```

Testbench:



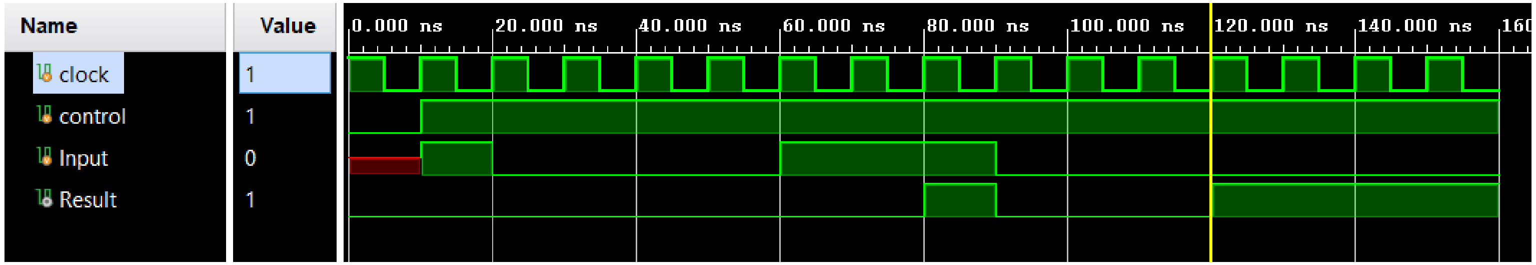
```verilog
`timescale 1ns / 1ps
`include "serial_adder.v"

module serial_adder_tb;

    reg clock,control,Input;
    wire Result;

    serial_adder UUT(clock,control,Input,Result);

    initial begin
        control = 0;
        clock = 1;
            forever #5 clock = ~clock;
    end
    //testbench not working :-(
    always begin
        #10;
        control = 1; Input = 1'b1; #10;
        control = 1; Input = 1'b1; #10;
        control = 1; Input = 1'b1; #10;
        control = 1; Input = 1'b1; #10;
        control = 1; Input = 1'b1; #10;
        control = 1; Input = 1'b0; #10;
        control = 1; Input = 1'b0; #10;
        control = 1; Input = 1'b0; #10;
        control = 1; Input = 1'b0;
        control = 1; #80;
        $finish;
    end

endmodule
```


clock
control
Input
addend
clock
control
in
out
shift_register
D_ff_carry
D
clock
control
Q
D_ff
full_adder_1
A
B
Cin
Cout
S
full_adder
out
augend
clock
control
in
out
shift_register


addend
clock
control
in
D_ff_1
D
clock
control
Q
D_ff
D_ff_2
D
clock
control
Q
D_ff
D_ff_3
D
clock
control
Q
D_ff
D_ff_4
D
clock
control
Q
D_ff
out
shift_register

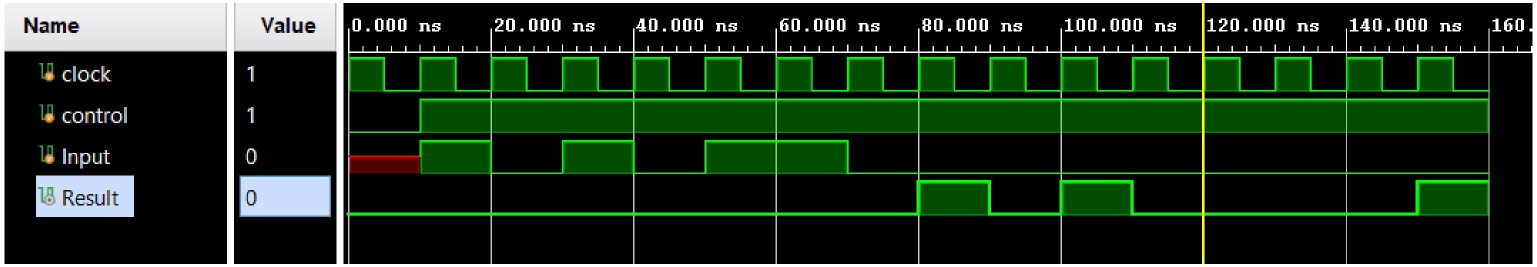
We can read the results after 120 ns. (yellow line in graphics)

0001 + 1110 =



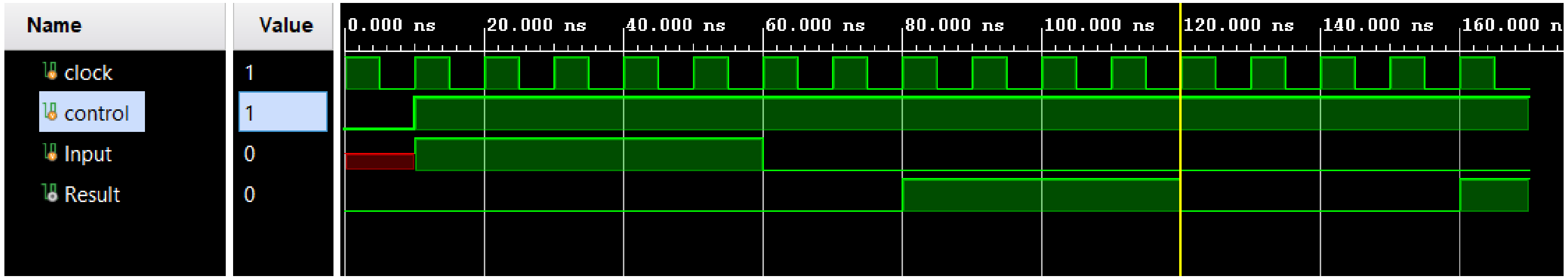
Name	Value
clock	1
control	1
Input	0
Result	1

0101 + 0011 =



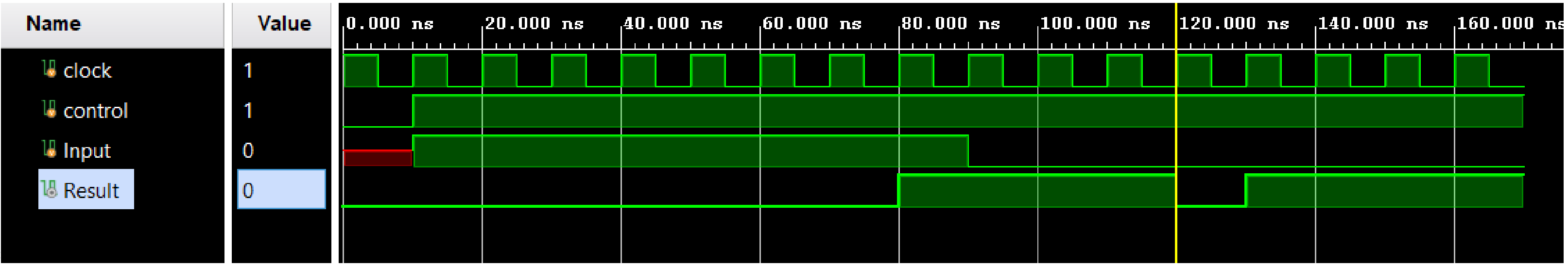
Name	Value
clock	1
control	1
Input	0
Result	0

1111 + 0001 =



Name	Value
clock	1
control	1
Input	0
Result	0

1111 + 1111 =



Name	Value
clock	1
control	1
Input	0
Result	0