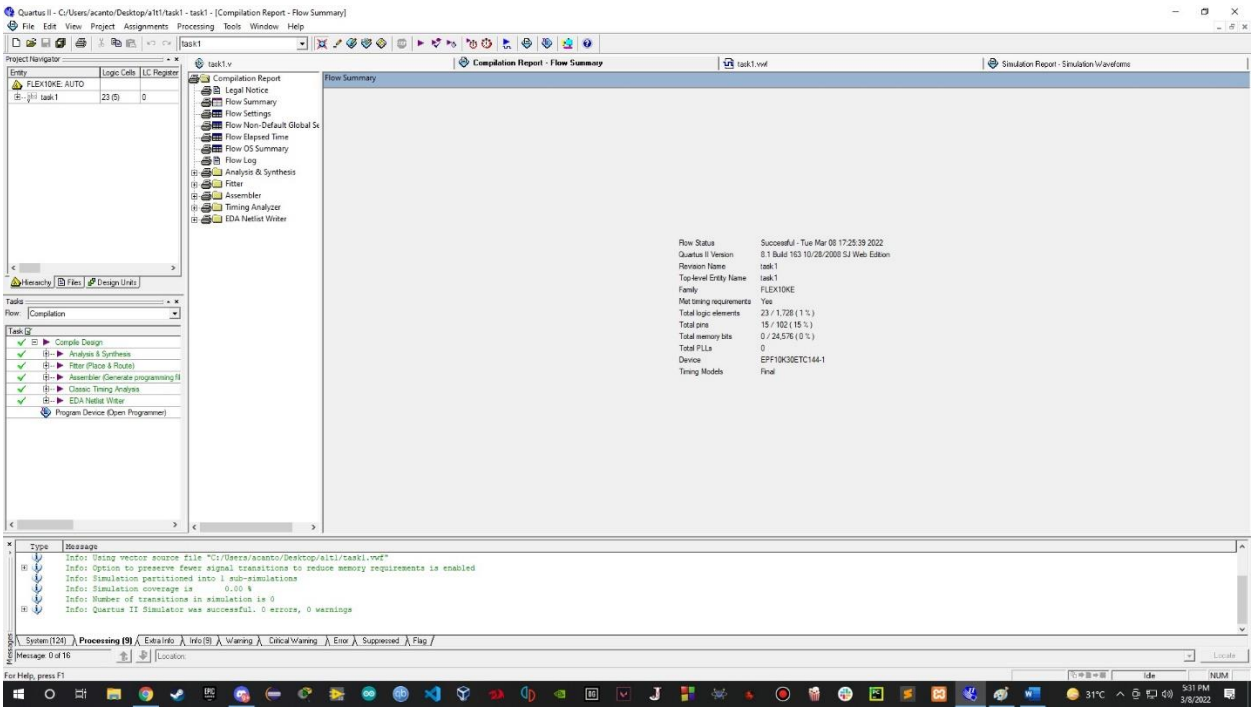
Task1:

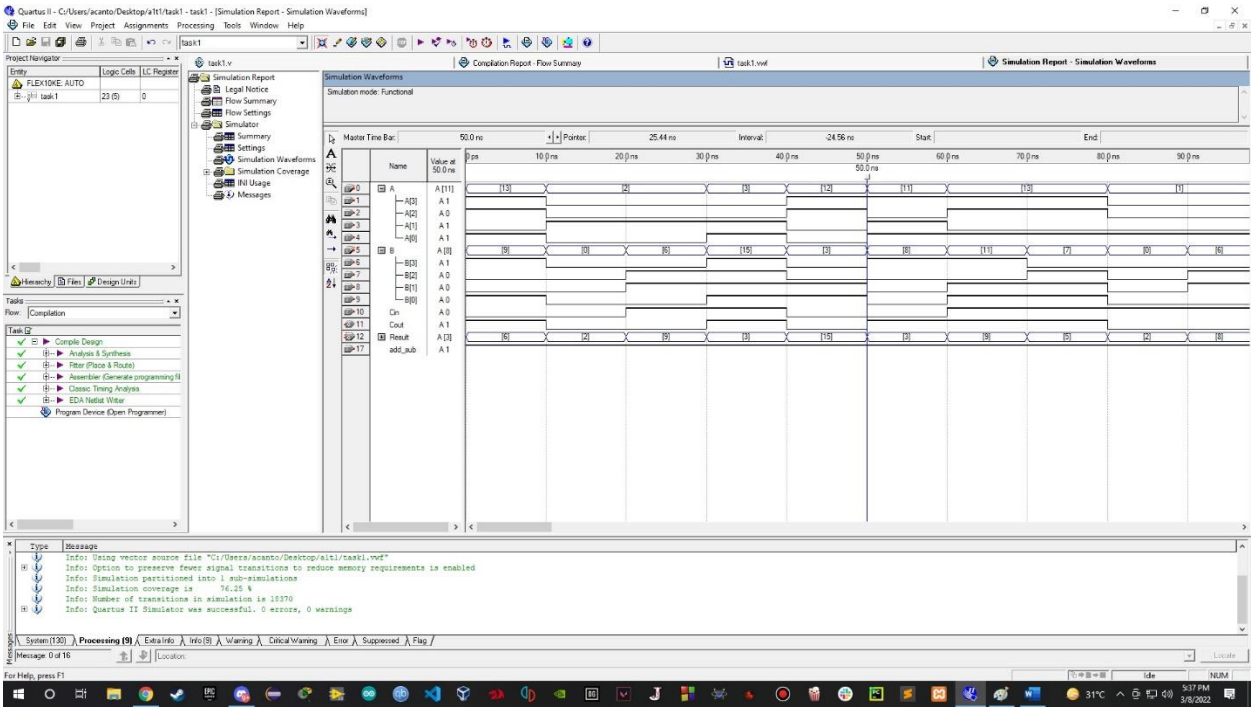
Code:

```
module task1(A,B,Cin,Result,Cout,add_sub);
        output [3:0] Result;
        output Cout;

        reg [3:0] Result;
        reg Cout;
        reg [4:0] temp;

        input add_sub, Cin;
        input [3:0] A, B;

        wire add_sub;

        always@(add_sub or A or B or Cin)
        begin
        if (add_sub == 1)
        temp = A + B + Cin;
        Result = temp [3:0];
        Cout = temp[4];
        if (add_sub == 0)
        temp = A - B - Cin;
        Result = temp [3:0];
        Cout = temp[4];
        end
endmodule
```

Compilation Report:



simulation report:



Discussion: Here this is a 4-bit full adder/subtractor. It works as both adder and subtractor. There is a control pin named add_sub which will control the functionality of the system. There is a carry in pin which is for the extra bit in the LSB. And the carry out pin shows extra bit of the addition of A and B. for example in the above picture the value of A is 11 and value of B is 8 so

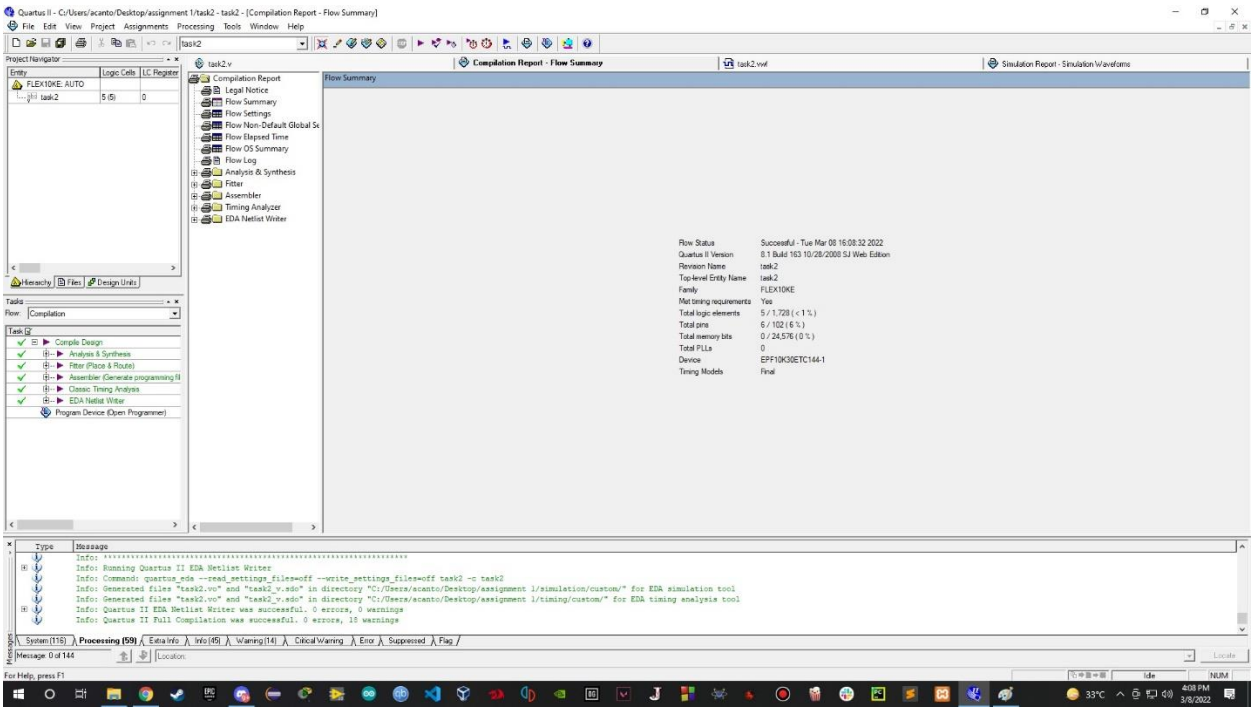
the addition is 19. But as it is a 5 bit output so the output generates 1 in carry out and 0011 in result. So the total output is 10011 which is in decimal 19.

Task 2:

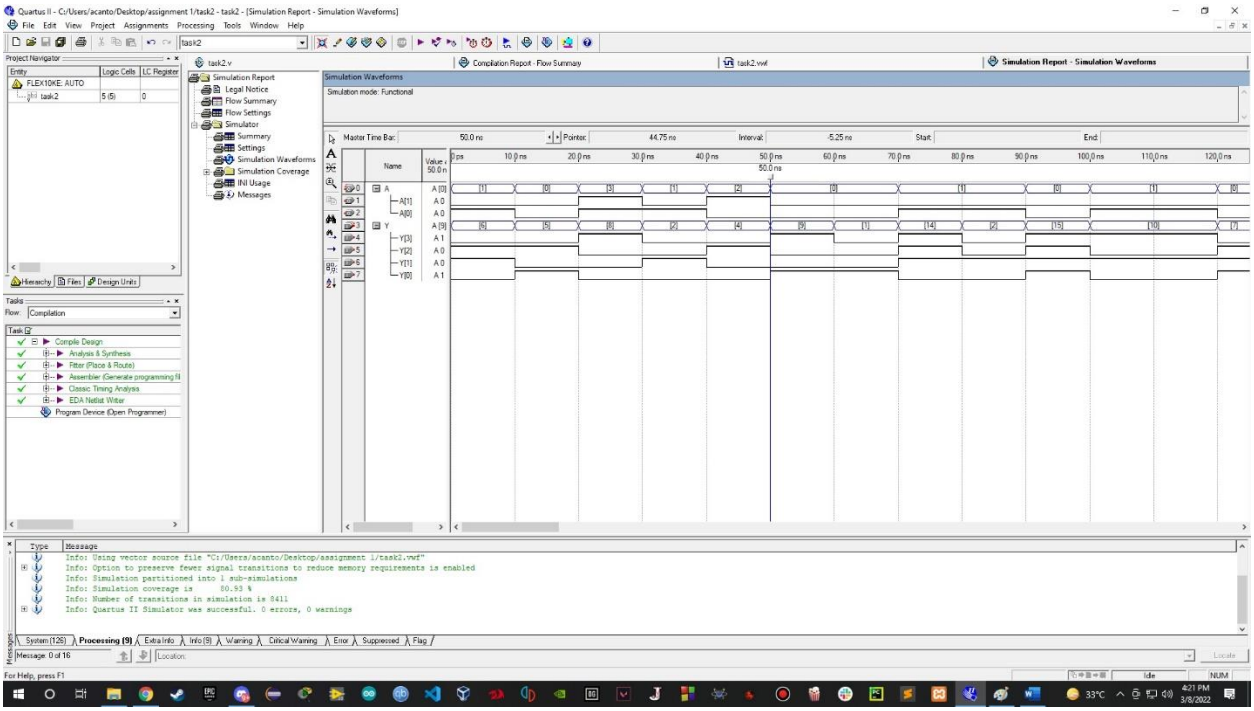
Code:

```
module task2(A, Y);

	input [3:0]Y;
	output reg [1:0]A;

	always@(Y)
	begin
	casex(Y)
		4'bxxx1:A = 2'b00;
		4'bxx1x:A = 2'b01;
		4'b1xxx:A = 2'b11;
		4'bx1xx:A = 2'b10;
	endcase
	end
endmodule
```

compilation report:



simulation report:



Discussion: Since this is a priority encoder, and according to the question the priority is 0>1>3>2. So, if bit 0 gets 1; the output will show 0 as here 0 is the highest priority. We get output 1 if bit 1 is 1. Again, we get output 3 if bit 3 is 1. Finally, we will get output 2 if all bits are 0 and only bit 2 is 1as bit 2 has the lowest priority. And if multiple bits or all bit is high then

the output will be 0 as 0 has the highest priority. So, if multiple bits are high then the system will generate output as the priority sequence.