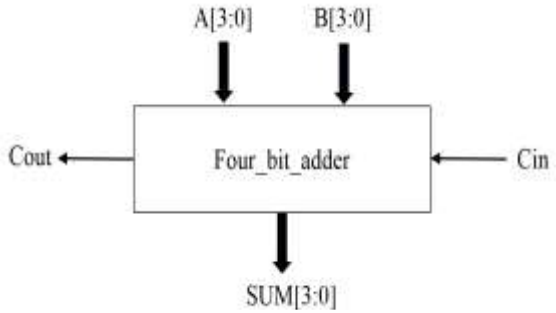
Tutorial 2 – B2

```
`timescale 1ns / 1ps

module Four_bit_adder(A,B,Cin,SUM,Cout);
  input [3:0] A,B;
  input Cin;
  output [3:0] SUM;
  output Cout;

  assign {Cout,SUM} = A + B + Cin;

endmodule
```

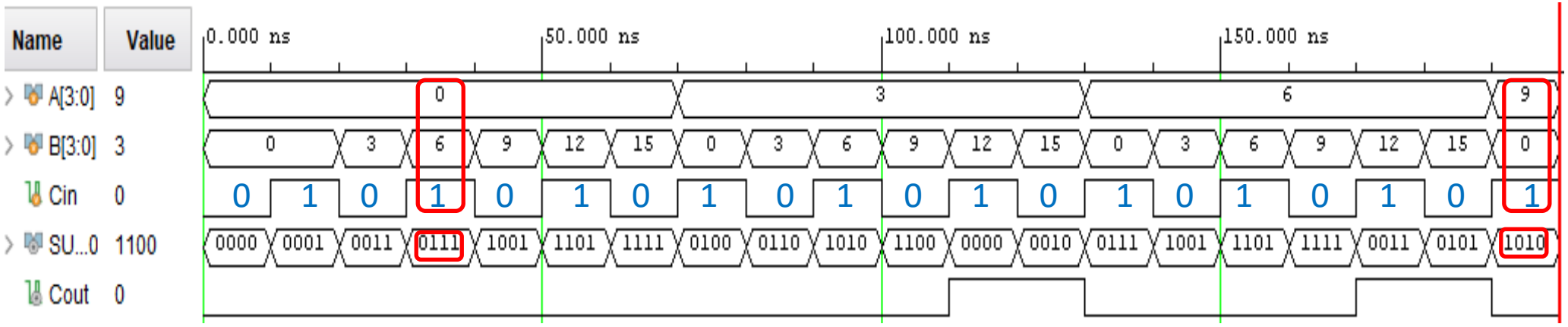
Concatentate to become 5-bit



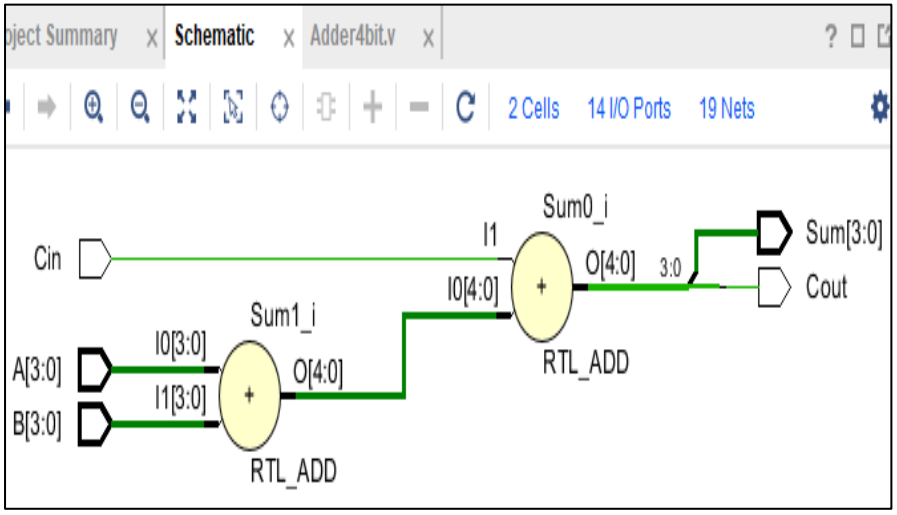
Test Bench (not required in this question).

```
`timescale 1ns / 1ps

module Four_bit_adder_tb();

  reg [3:0] A=0,B=0;   reg Cin=0;   wire [3:0] SUM;   wire Cout;

  integer i, j;

  Four_bit_adder dut(A, B, Cin, SUM, Cout);

  initial
  begin
    for (i=0; i<=15; i=i+3)
      for (j=0; j<=15; j=j+3)
      begin
        #10  A=i; B=j; Cin=~Cin;
      end
  end
endmodule
```

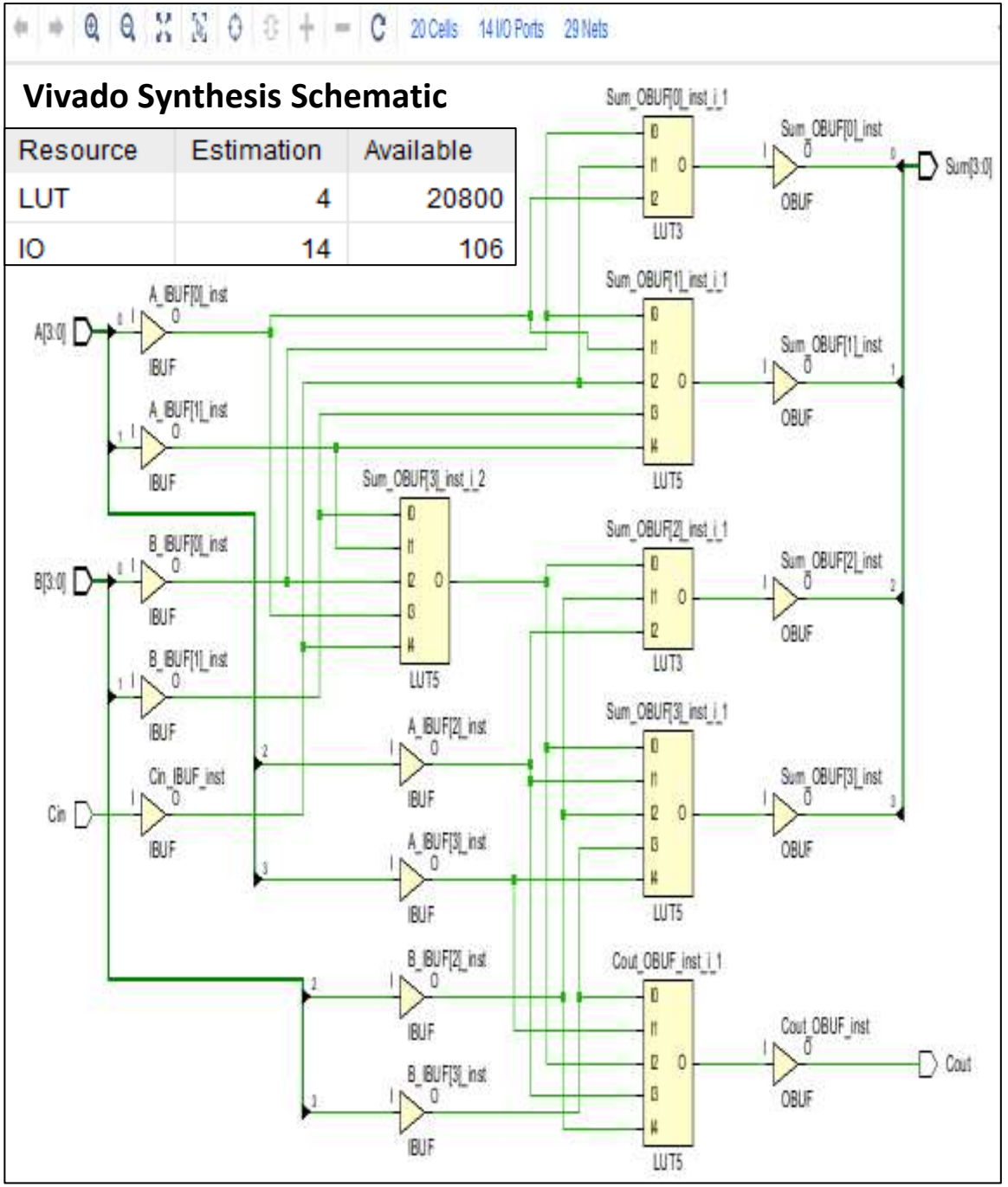
This is not the only way to test.



Vivado RTL Schematic



Vivado Synthesis Schematic



Observation

Vivado produces identical circuits for the 4-bit adder even with different ways of programming.

Compare with the circuits produced after synthesis & implementation steps in:

- N-bit Adder (Fig. 2-10)
- 4-bit Adder with FA sub-circuits (Fig. 2-12)