#100 days of rtl

https://github.com/Bhuvan-2602 www.LinkedIn.com/bhuvan-p-4825a9358

**Description:**

A **Full Adder** is a digital circuit that adds three binary inputs (A, B, and Cin) and produces two outputs: **Sum** and **Carry-out** (Cout). It is a fundamental building block for multi-bit binary addition in arithmetic logic units (ALUs).

**Objective:**

* Understand how to combine logic gates to perform binary addition.
* Learn to implement and simulate combinational circuits using Verilog.

**Inputs:**

* a: 1-bit binary input
* b: 1-bit binary input
* cin: 1-bit Carry input

**Outputs:**

* sum: 1-bit output representing (a ^ b ^ cin)
* cout: 1-bit output representing (a & b & c)

**Design Approach:**

The Full Adder computes the sum and carry-out of three 1-bit inputs: A, B, and Cin.

* Sum is calculated as: A ⊕ B ⊕ Cin
* Cout is generated when at least two inputs are high:

Cout = (A & B) | (B & Cin) | (Cin & A)

This is a pure combinational logic design using XOR, AND, and OR gates.

**Verilog Code:**

module full\_add(

input a,

input b,

input cin,

output sum,

output cout

);

assign sum = a^b^cin;

assign cout = (a & b)|(b & cin)|(cin & a);

endmodule

**Testbench:**

module full\_add\_tb;

reg a,b,cin;

wire sum,cout;

full\_add dut (a,b,cin,sum,cout);

initial begin

#10; a=1'b0; b=1'b0; cin=1'b0;

#10; a=1'b0; b=1'b0; cin=1'b1;

#10; a=1'b0; b=1'b1; cin=1'b0;

#10; a=1'b0; b=1'b1; cin=1'b1;

#10; a=1'b1; b=1'b0; cin=1'b0;

#10; a=1'b1; b=1'b0; cin=1'b1;

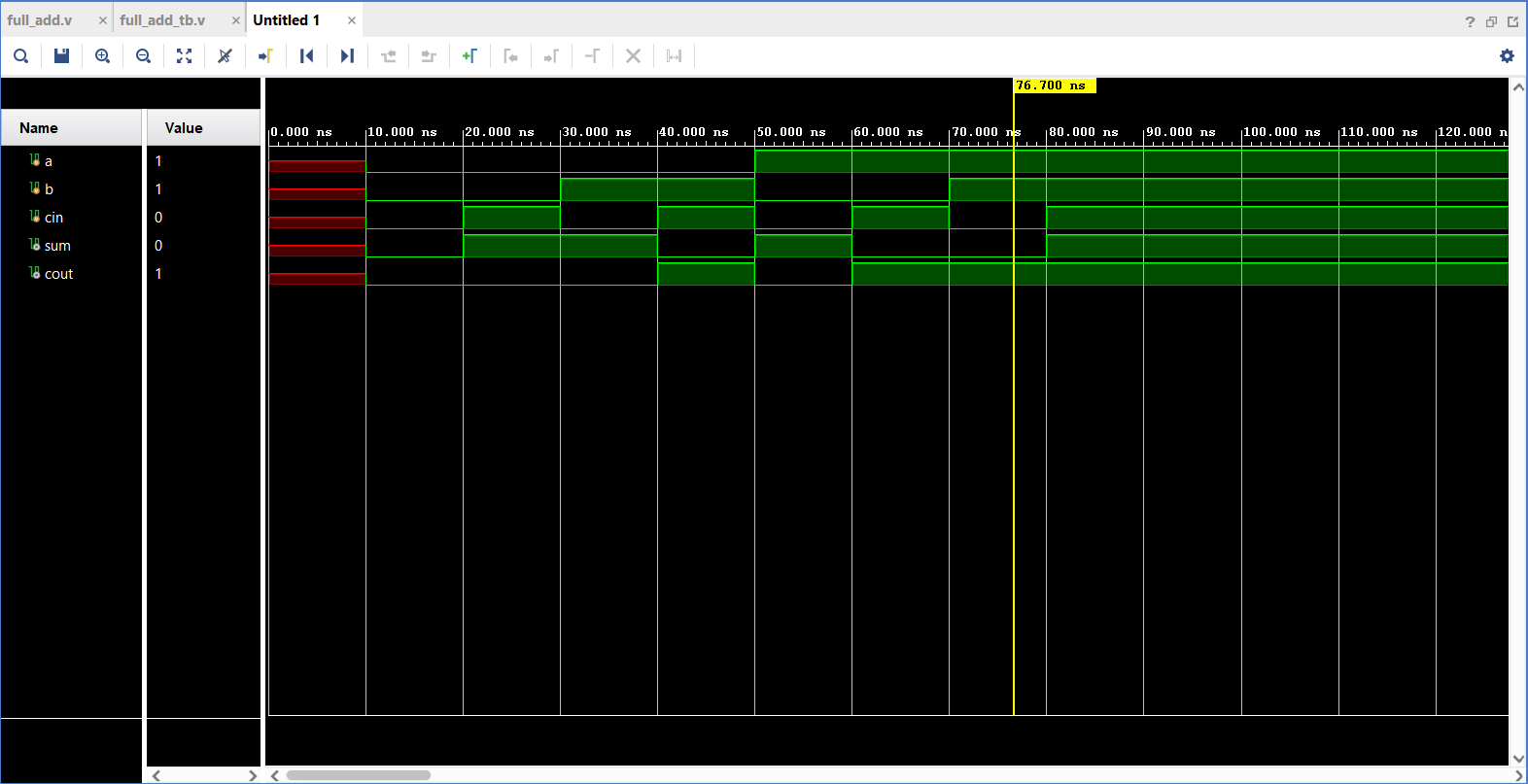
#10; a=1'b1; b=1'b1; cin=1'b0;

#10; a=1'b1; b=1'b1; cin=1'b1;

end

endmodule

**Waveform:**



**Schematic:**

