Figure 1 shows a block diagram representing a VHDL testbench for a full adder.

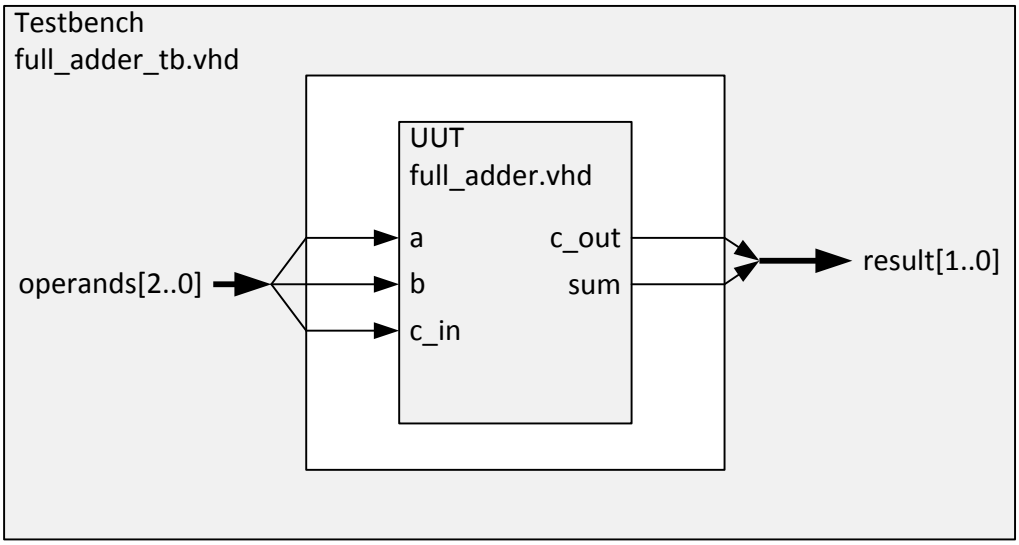


**Figure 1 Testbench for a Full Adder**

The VHDL code for the testbench is shown on page 2. Enter the testbench code in the Quartus text editor and save it to the folder containing the project with a full adder file called **full_adder.vhd**. Do not add the testbench to the project. The checkbox shown in Figure 2 should be unchecked. (The testbench is compiled by ModelSim, not Quartus, and is therefore not in the Quartus project.)

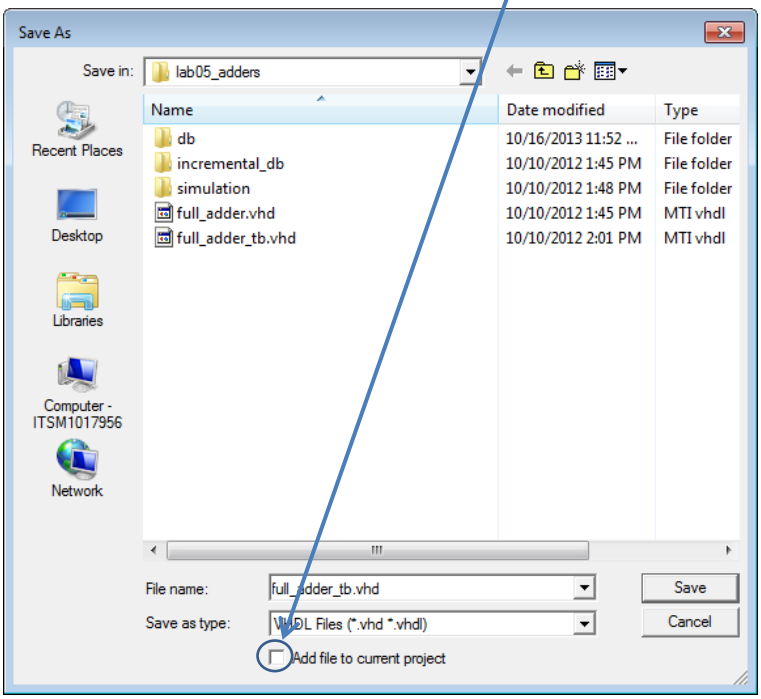


**Figure 2 Saving the Testbench File (DO NOT add to the project)**

To set up the testbench as a simulation in ModelSim, follow the screen shots in the pages following the code.

```
-- full_adder_tb.vhd
-- test bench for full_adder.vhd
-- simulation criteria:
--     2-bit output = sum of three 1-bit inputs
--     Test by applying a binary count on the inputs
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;

entity full_adder_tb is
    -- test bench entity has no external ports
end full_adder_tb;

architecture test_bench of full_adder_tb is
    component full_adder
        port(
            a, b, c_in : in   std_logic;
            c_out, sum : out  std_logic
            );
    end component;

    signal operands   : std_logic_vector(2 downto 0);
    signal result     : std_logic_vector(1 downto 0);
begin
    -- instantiate full adder with test signals
    full_add: full_adder port map(   a      => operands(2),
                                     b      => operands(1),
                                     c_in   => operands(0),
                                     c_out  => result(1),
                                     sum    => result(0));

    -- stimulus process (generates simulation input waveforms)
    stim_proc: process
    begin
        -- Count from 0 to 7
        for j in 0 to 7 loop
            -- Convert integer count to 3-bit std_logic_vector
            operands <= CONV_STD_LOGIC_VECTOR(j,3);
            -- Set time interval for each input state
            wait for 100 ns;
        end loop;
        wait;      -- stall here (simulation done)
    end process;
end test_bench;
```

**File Path to ModelSim**

Make sure that the file path to ModelSim is set as shown in Figure 3. The path is for Quartus II version 12,1, Service Pack 1 (12.1sp1). If you are using a different version, set the file path accordingly. (Use the Browse button  to avoid typing errors in the file path.)

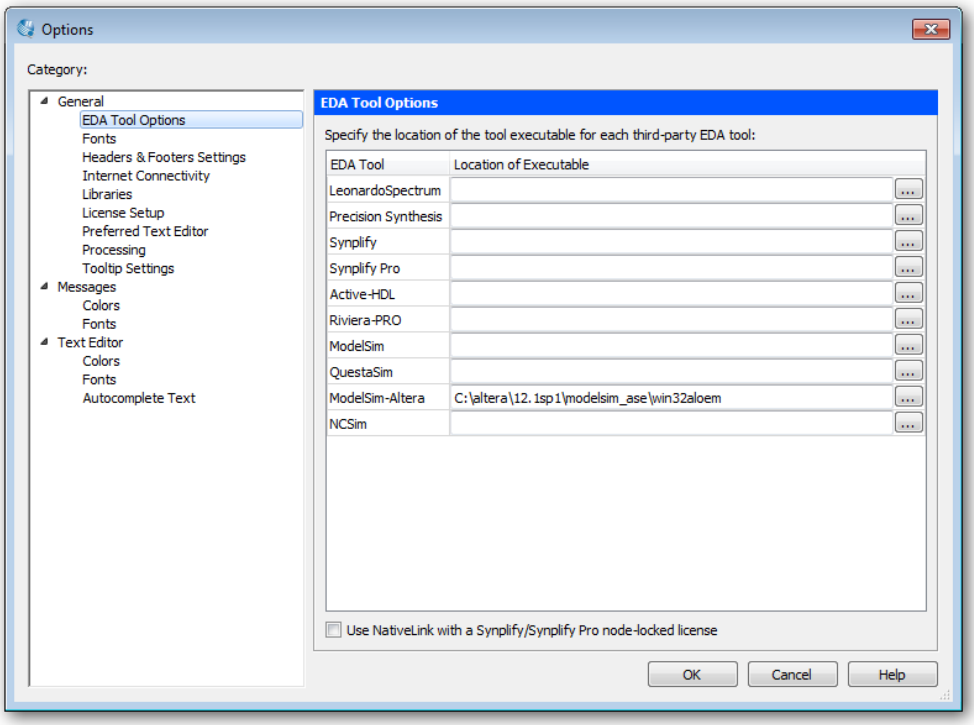


Figure 3 ModelSim File Path

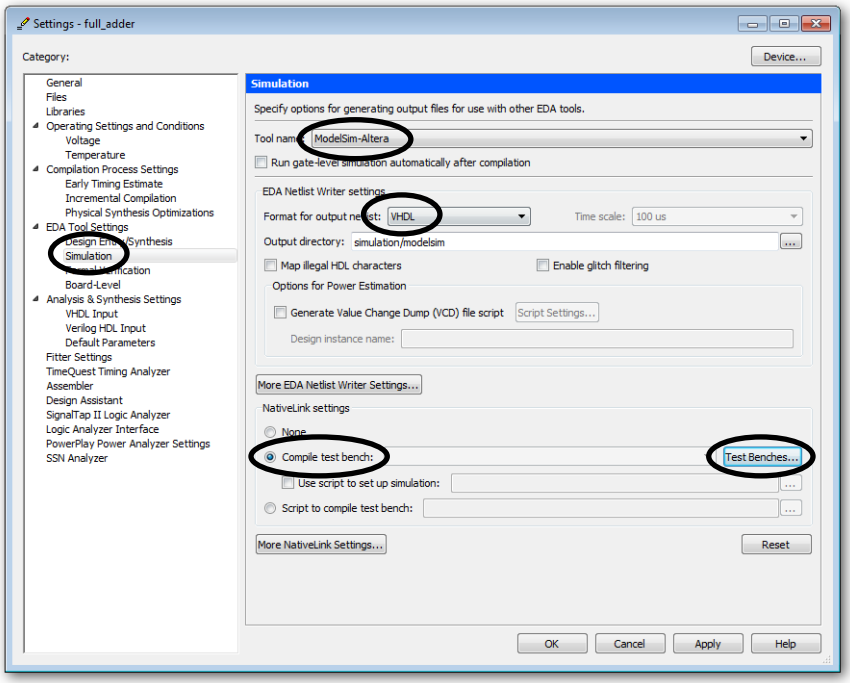


Figure 4 Assignments Menu, Settings Dialog

Make settings as shown in Figure 4 and click **Test Benches**. The dialog box in Figure 5 will appear. Click **New.**

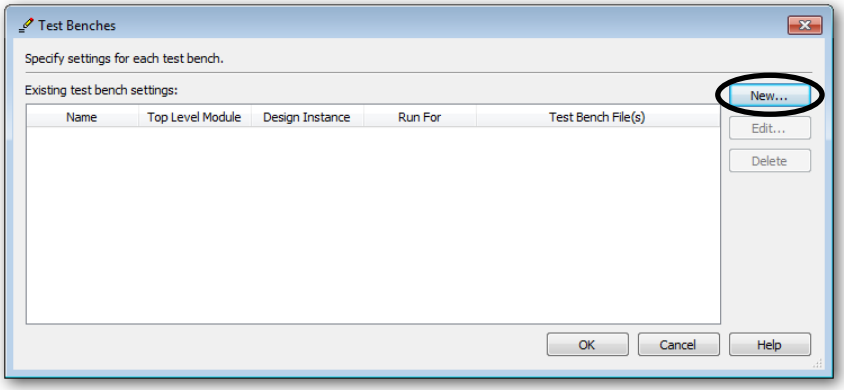


Figure 5 Test Benches Dialog (Empty)

Complete the dialog boxes as shown on the following pages.

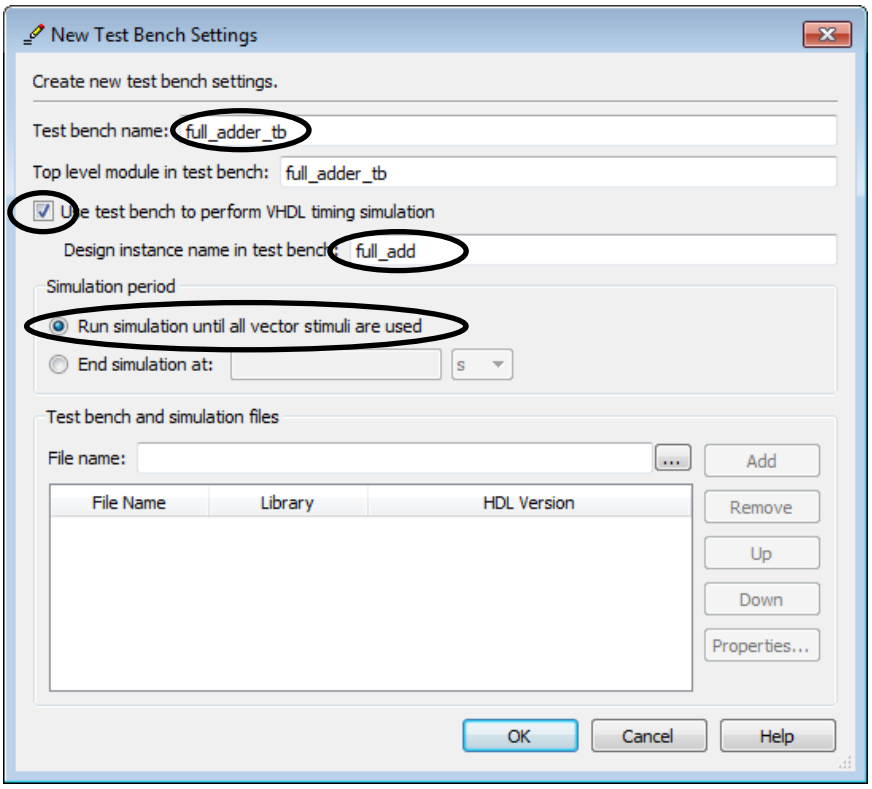


Figure 6 New Test Bench Settings (1)

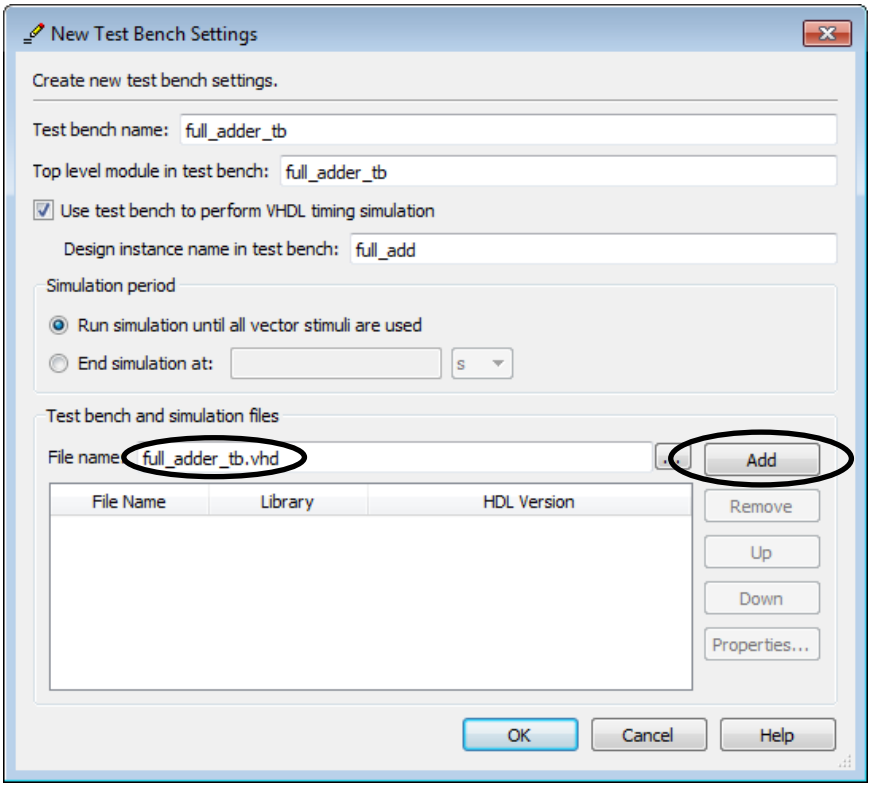


Figure 7 New Test Bench Settings (2)

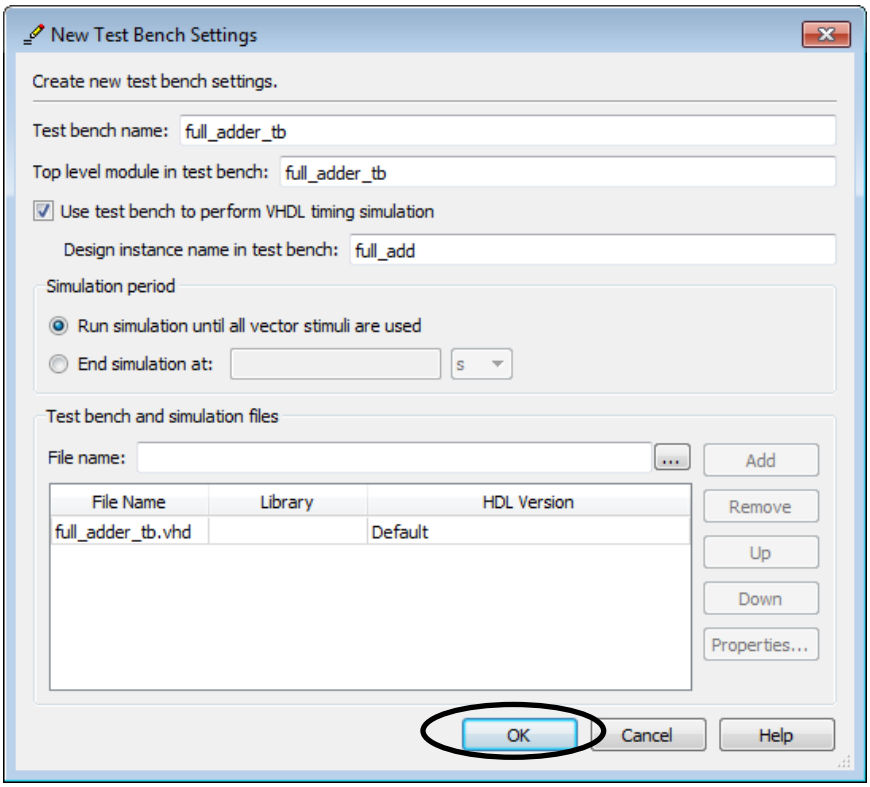


Figure 8 New Test Bench Settings (3)

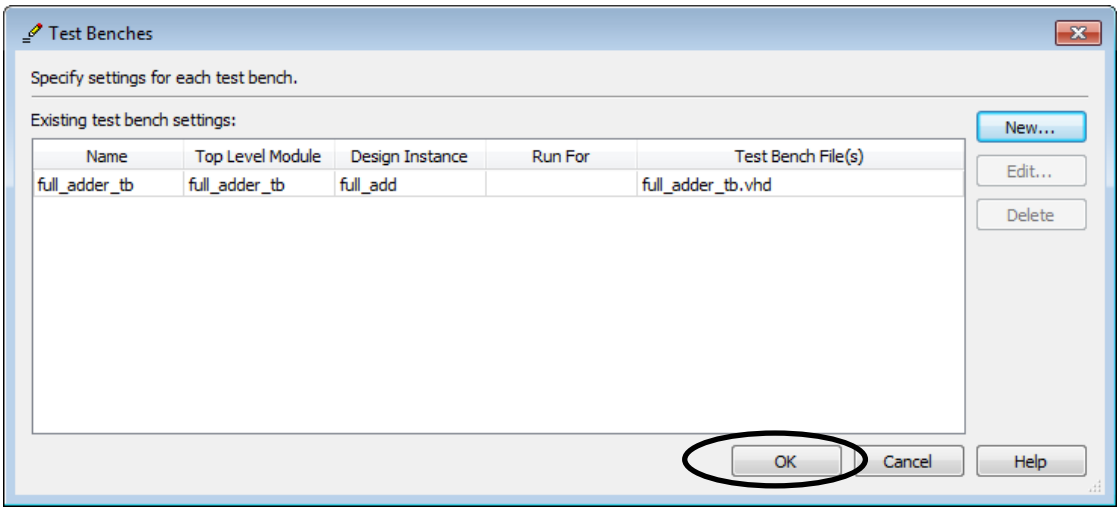


Figure 9 Test Benches Dialog (Populated)

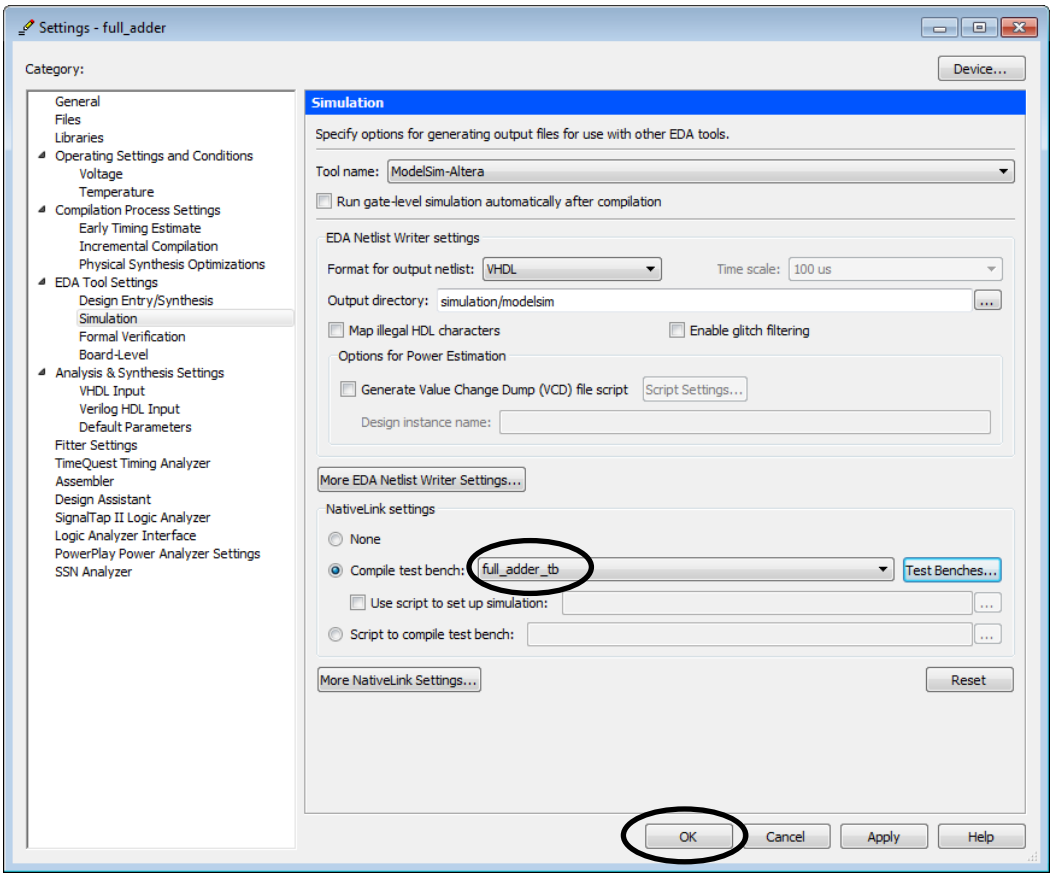


Figure 10 Simulation Settings (Populated)

Run the **Gate Level Simulation** from the Quartus window. ModelSim will generate a Wave window, as shown in Figure 11. The wave window fonts are changed to 12-point and the units to ns (from ps) by typing the following text at the VSIM> prompt in the ModelSim Transcript window.

configure wave -font 12 -timelineunits ns

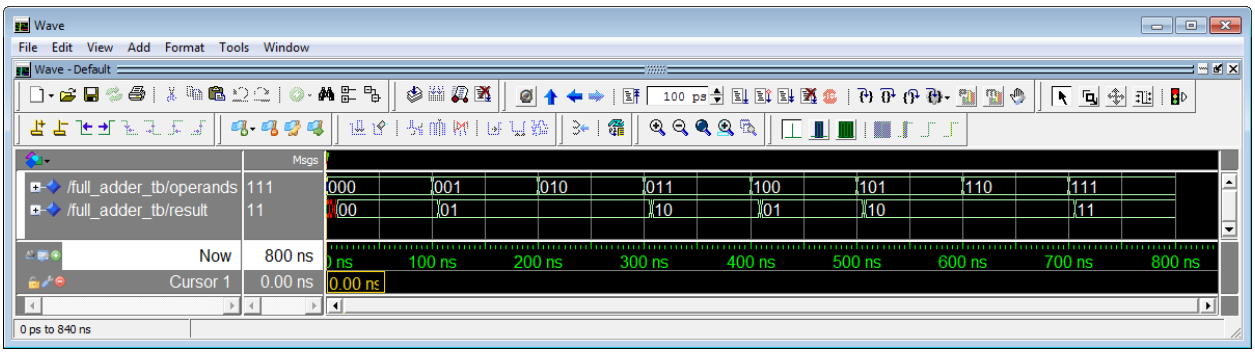


Figure 11 ModelSim Wave Window