plibrary IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--use IEEE.NUMERIC\_STD.ALL;

-- Uncomment the following library declaration if instantiating

-- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity fulladder is

Port ( a : in STD\_LOGIC;

b : in STD\_LOGIC;

cin : in STD\_LOGIC;

sum : out STD\_LOGIC;

cout : out STD\_LOGIC);

end fulladder;

architecture Behavioral of fulladder is

begin

sum<=a xor b xor cin;

cout<=(a and b)or(cin and(a xor b));

end Behavioral;

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--USE ieee.numeric\_std.ALL;

ENTITY fulladdertest IS

END fulladdertest;

ARCHITECTURE behavior OF fulladdertest IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT fulladder

PORT(

a : IN std\_logic;

b : IN std\_logic;

cin : IN std\_logic;

sum : OUT std\_logic;

cout : OUT std\_logic

);

END COMPONENT;

--Inputs

signal a : std\_logic := '0';

signal b : std\_logic := '0';

signal cin : std\_logic := '0';

--Outputs

signal sum : std\_logic;

signal cout : std\_logic;

-- No clocks detected in port list. Replace <clock> below with

-- appropriate port name

-- constant <clock>\_period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: fulladder PORT MAP (

a => a,

b => b,

cin => cin,

sum => sum,

cout => cout

);

-- Clock process definitions

--<clock>\_process :process

--begin

-- <clock> <= '0';

--wait for <clock>\_period/2;

--<clock> <= '1';

--wait for <clock>\_period/2;

-- end process;

-- Stimulus process

stim\_proc: process

begin

-- hold reset state for 100 ns.

a<='0';

b<='0';

cin<='0';

wait for 100 ns;

a<='0';

b<='0';

cin<='1';

wait for 100 ns;

a<='0';

b<='1';

cin<='0';

wait for 100 ns;

a<='0';

b<='1';

cin<='1';

wait for 100 ns;

a<='1';

b<='0';

cin<='0';

wait for 100 ns;

a<='1';

b<='0';

cin<='1';

wait for 100 ns;

a<='1';

b<='1';

cin<='0';

wait for 100 ns;

a<='1';

b<='1';

cin<='1';

wait for 100 ns;

-- wait for <clock>\_period\*10;

-- insert stimulus here

wait;

end process;

END;