```
ENTITY full_adder IS           --- Full Adder
PORT(A,B,Cin: IN bit ;
  S, Cout : OUT bit);
END full_adder;
ARCHITECTURE full_adder_beh OF full_adder IS
BEGIN
  PROCESS(A,B,Cin)              -- Sensitive on all the three bits
    VARIABLE temp :bit;
     BEGIN                    --- DOES the addition in one DELTA time
       temp := A XOR B;
       S <= temp XOR Cin;
       Cout <= (A AND B) OR (temp AND Cin);
     END PROCESS ;
END full_adder_beh;



--Test Bench
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY Testbench_full_adder IS
END Testbench_full_adder;

ARCHITECTURE behavior OF Testbench_full_adder IS
 -- Component Declaration for the Unit Under Test (UUT)
 COMPONENT full_adder
 PORT(
 A : IN bit;
 B : IN bit;
 Cin : IN bit;
 S : OUT bit;
 Cout : OUT bit
 );
 END COMPONENT;
 --Inputs
 signal A : bit := '0';
 signal B : bit := '0';
 signal Cin : bit := '0';
 --Outputs
 signal S : bit;
 signal Cout : bit;
BEGIN
 -- Instantiate the Unit Under Test (UUT)
 uut: full_adder PORT MAP (
 A => A,
 B => B,
 Cin => Cin,
 S => S,
 Cout => Cout
 );
 -- Stimulus process
 stim_proc: process
 begin
 -- hold reset state for 100 ns.
 wait for 100 ns;
 -- insert stimulus here
 A <= '1';
 B <= '0';
 Cin <= '0';
 wait for 10 ns;
 A <= '0';
 B <= '1';
 Cin <= '0';
 wait for 10 ns;
 A <= '1';
 B <= '1';
 Cin <= '0';
 wait for 10 ns;
 A <= '0';
 B <= '0';
 Cin <= '1';
 wait for 10 ns;
 A <= '1';
 B <= '0';
 Cin <= '1';
 wait for 10 ns;
 A <= '0';
 B <= '1';
 Cin <= '1';
 wait for 10 ns;
 A <= '1';
 B <= '1';
 Cin <= '1';
 wait for 10 ns;
 end process;
END;
```