**1. VHDL code for full adder using Data flow modeling**  
  
library IEEE;  
use IEEE.STD\_LOGIC\_1164.ALL;  
entity full\_adder is  
Port ( a : in STD\_LOGIC;  
b : in STD\_LOGIC;  
c : in STD\_LOGIC;  
sum : out STD\_LOGIC;  
cout : out STD\_LOGIC);  
end full\_adder;  
architecture test\_fa of full\_adder is  
begin  
sum <= A XOR B XOR Cin ;  
cout <= (A AND B) OR (Cin AND A) OR (Cin AND B) ;  
end test\_fa;

**Test bench Code for Full Adder:**  
  
LIBRARY ieee;  
USE ieee.std\_logic\_1164.ALL;  
ENTITY tb\_test\_fa IS  
END tb\_test\_fa;  
ARCHITECTURE behavior OF tb\_test\_fa IS  
COMPONENT test\_Full\_Adder  
PORT(  
x : IN std\_logic;  
y : IN std\_logic;  
z : IN std\_logic;  
sum : OUT std\_logic;  
cout : OUT std\_logic  
);  
END COMPONENT;  
--Inputs  
signal x : std\_logic := '0';  
signal y : std\_logic := '0';  
signal z : 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  
BEGIN  
-- Instantiate the Unit Under Test (UUT)  
uut: test\_Full\_Adder PORT MAP (  
x => x,  
y => y,  
z => z,  
sum => sum,  
cout => cout  
);-- Stimulus process  
process  
begin  
wait for 5 ns;  
x <= '0';  
y <= '0';  
z <= '0';  
wait for 10 ns;  
x <= '0';  
y <= '0';  
z <= '1';  
wait for 10 ns;  
x <= '0';  
y <= '1';  
z <= '0';  
wait for 10 ns;  
x <= '0';  
y <= '1';  
z <= '1';  
wait for 10 ns;  
x <= '1';  
y <= '0';  
z <= '0';  
wait for 10 ns;  
x <= '1';  
y <= '0';  
z <= '1';  
wait for 10 ns;  
x <= '1';  
y <= '1';  
z <= '0';  
wait for 10 ns;  
x <= '1';  
y <= '1';  
z <= '1';  
wait for 10 ns;  
end process;  
END;