**VHDL code for full adder using Behavioral modeling**

library IEEE;  
use IEEE.STD\_LOGIC\_1164.ALL;  
use IEEE.STD\_LOGIC\_ARITH.ALL;  
use IEEE.STD\_LOGIC\_UNSIGNED.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  
process(a,b,c)  
begin  
if(a='0' and b='0' and c='0') then  
sum <= '0';cout <= '0';  
elsif( a='0' and b='0' and c='1')then  
sum <= '1' ;  
cout <= '0' ;  
elsif ( a='0' and b='1' and c='0') then  
sum <= '1';  
cout <= '0 ';  
elsif( a='0' and b='1' and c='1')then  
sum <= '0';  
cout <= '1';  
elsif( a='1' and b='0' and c='0')then  
sum <= '1';  
cout <= '0';  
elsif( a='1' and b='0' and c='1')then  
sum <= '0';  
cout <= '1';  
elsif( a='1' and b='1' and c='0')then  
sum <= '0';  
cout <= '1';  
else  
sum <= '1' ;  
cout <= '1';  
end if;  
end process;  
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  
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;