**VHDL code for 4:1 MUX**

library IEEE;  
use IEEE.STD\_LOGIC\_1164.all;  
  
entity mux\_test is  
port(  
A,B,C,D : in STD\_LOGIC;  
S0,S1: in STD\_LOGIC;  
Z: out STD\_LOGIC  
);  
end mux\_test;  
  
architecture beh\_test of mux\_test is  
begin  
process (A,B,C,D,S0,S1) is  
begin  
if (S0 ='0' and S1 = '0') then  
Z <= A;  
elsif (S0 ='1' and S1 = '0') then  
Z <= B;  
elsif (S0 ='0' and S1 = '1') then  
Z <= C;  
else  
Z <= D;  
end if;  
end process;  
end beh\_test;  
  
**Test Bench Code for 4 to 1 Multiplexer:**  
  
LIBRARY ieee;  
USE ieee.std\_logic\_1164.ALL;  
ENTITY tb\_mux\_test IS  
END tb\_mux\_test;  
ARCHITECTURE behavior OF tb\_mux\_test IS  
-- Component Declaration for the Unit Under Test (UUT)  
COMPONENT mux\_test  
PORT(  
A : IN std\_logic;  
B : IN std\_logic;  
C : IN std\_logic;  
D : IN std\_logic;  
S0 : IN std\_logic;  
S1 : IN std\_logic;  
Z : OUT std\_logic  
);  
END COMPONENT;  
--Inputs  
signal A : std\_logic := '0';  
signal B : std\_logic := '0';  
signal C : std\_logic := '0';  
signal D : std\_logic := '0';  
signal S0 : std\_logic := '0';  
signal S1 : std\_logic := '0';  
--Outputs  
signal Z : std\_logic;  
BEGIN  
uut: mux\_test PORT MAP (  
A => A,  
B => B,  
C => C,  
D => D,  
S0 => S0,  
S1 => S1,  
Z => Z  
);  
  
process  
begin  
wait for 5 ns;  
  
A <= '1';  
B <= '0';  
C <= '1';  
D <= '0';  
  
S0 <= '0'; S1 <= '0';  
wait for 10 ns;  
  
S0 <= '0'; S1 <= '1';  
wait for 10 ns;  
  
S0 <= '1'; S1 <= '0';  
wait for 10 ns;  
  
S0 <= '1'; S1 <= '1';  
wait for 10 ns;  
  
end process;  
END;