**MUX 8-1 CODE**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity MUX is

Port ( A : in STD\_LOGIC\_VECTOR (7 downto 0);

S : in STD\_LOGIC\_VECTOR (2 downto 0);

Y : out STD\_LOGIC);

end MUX;

architecture Behavioral of MUX is

begin

process(A,S)

begin

case S is

when "000"=> Y <=A(0);

when "001"=> Y <=A(1);

when "010"=> Y <=A(2);

when "011"=> Y <=A(3);

when "100"=> Y <=A(4);

when "101"=> Y <=A(5);

when "110"=> Y <=A(6);

when others => Y <=A(7);

end case;

end process;

end Behavioral;

**TEST BENCH**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY MUXTB IS

END MUXTB;

ARCHITECTURE behavior OF MUXTB IS

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

COMPONENT MUX

PORT(

A : IN std\_logic\_vector(7 downto 0);

S : IN std\_logic\_vector(2 downto 0);

Y : OUT std\_logic

);

END COMPONENT;

--Inputs

signal A :std\_logic\_vector(7 downto 0) := (others => '0');

signal S :std\_logic\_vector(2 downto 0) := (others => '0');

--Outputs

signal Y :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: MUX PORT MAP (

A => A,

S => S,

Y => Y

);

stim\_proc: process

begin

-- hold reset state for 100 ns.

A(1)<='0'; S<="000";

wait for 100 ns;

assert Y='0'; report "error in logic";

wait for 100 ns;

-- insert stimulus here

A(2)<='1'; S<="010";

wait for 100 ns ;

assert Y='1'; report "error in logic";

end process;

END;