**ALU CODE**

library IEEE;

use IEEE.STD\_LOGIC\_1164.all;

use IEEE.STD\_LOGIC\_arith.all;

use IEEE.STD\_LOGIC\_unsigned.all;

entity ALU is

Port ( A : in STD\_LOGIC\_VECTOR (3 downto 0);

B : in STD\_LOGIC\_VECTOR (3 downto 0);

S : in STD\_LOGIC\_VECTOR (3 downto 0);

M : in STD\_LOGIC;

F : out STD\_LOGIC\_VECTOR (3 downto 0));

end ALU;

architecture Behavioral of ALU is

begin

process(A,B,S,M)

begin

If (M='0') then

case S is

when"0000"=>F<=A+B;

when"0001"=>F<=A-B;

when"0010"=>F<=A+'1';

when"0011"=>F<=A-'1';

when others=>Null;

end case;

else

case S is

when"0000" => F<=A AND B;

when"0001" => F<=A OR B;

when"0010" => F<=A NAND B;

when"0011" => F<=NOT A;

when others => Null;

end case;

end if;

end process;

end Behavioral ;

**TEST BENCH**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY ALUTB IS

END ALUTB;

ARCHITECTURE behavior OF ALUTB IS

COMPONENT ALU

PORT(

A : IN std\_logic\_vector(3 downto 0);

B : IN std\_logic\_vector(3 downto 0);

S : IN std\_logic\_vector(3 downto 0);

M : IN std\_logic;

F : OUT std\_logic\_vector(3 downto 0)

);

END COMPONENT;

--Inputs

signal A :std\_logic\_vector(3 downto 0) := (others => '0');

signal B :std\_logic\_vector(3 downto 0) := (others => '0');

signal S :std\_logic\_vector(3 downto 0) := (others => '0');

signal M :std\_logic := '0';

--Outputs

signal F :std\_logic\_vector(3 downto 0);

-- No clocks detected in port list. Replace <clock> below with

-- appropriate port name

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: ALU PORT MAP (

A => A,

B => B,

S => S,

M => M,

F => F

);

stim\_proc: process

begin

M <= '0' ; S <= "0000"; A <= "0001"; B <= "0010";

Assert F <= "0011" report "error in code";

wait for 100 ns;

M <= '1' ; S <= "0011"; A <= "0011";

Assert F <= "1100" report "error in code";

wait for 100 ns;

M <= '1' ; S <= "0000"; A <= "0001"; B <= "0010";

Assert F <= "0011" report "error in code";

wait for 100 ns;

wait;

end process;

END;