**Aim: To Implement magnitude comparator combinational circuit**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity comparator is

Port(A : in STD\_LOGIC\_VECTOR(1 downto 0);

G,L,E : out STD\_LOGIC);

end comparator;

architecture Behavioral of comparator is

begin

process (A)

begin

G <= '0';

L <= '0';

E <= '0';

if (A(0) > A(1)) then

L <= '1';

elsif (A(1) > A(0)) then

G <= '1';

else

E <= '1';

end if;

end process;

end Behavioral;

**COMPARATOR TEST BEBTCH**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity comparator\_tb is

end entity;

architecture tb of comparator\_tb is

component comparator is

Port ( A : in STD\_LOGIC\_VECTOR (1 downto 0);

G,L,E : out STD\_LOGIC);

end component;

signal A : STD\_LOGIC\_VECTOR(1 downto 0);

signal G, L, E : STD\_LOGIC;

begin

uut: comparator port map(

A => A, G => G,

L => L, E => E);

stim: process

begin

A <= "00";

wait for 20 ns;

A <= "01";

wait for 20 ns;

A <= "10";

wait for 20 ns;

A <= "11";

wait for 20 ns;

wait;

end process;

end tb;