**Major codes for LAB**

1. **Logic gates:**

Y <= A AND B; ======AND gate

Y <= A OR B; ======OR gate

Y <= NOT ( A );======NOT Gate

Y <= A XOR B;======XOR gate

**Simulation ( Test Bench)**

begin

a<='0'; b<='0'; wait for 100 ns;

a<='0'; b<='1'; wait for 100 ns;

a<='1'; b<='0'; wait for 100 ns;

a<='1'; b<='1'; wait for 100 ns;

end process;

END;

1. **HALF ADDER**

VHDL code for half adder

entity half\_addlab is

Port ( a : in STD\_LOGIC;

b : in STD\_LOGIC;

sum : out STD\_LOGIC;

carry : out STD\_LOGIC);

end half\_addlab;

architecture Behavioral of half\_addlab is

begin

**sum <= a xor b;**

**carry <= a and b;**

end Behavioral;

**Half adder Test bench**

stim\_proc: process

begin

a<= '0'; b<= '0'; wait for 100 ns;

a<= '0'; b<= '1'; wait for 100 ns;

a<= '1'; b<= '0'; wait for 100 ns;

a<= '1'; b<= '1'; wait for 100 ns;

end process;

END;

**Full adder:** 3 inputs & 2 outputs

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity fullAdder is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

Cin : in STD\_LOGIC;

SUM : out STD\_LOGIC;

Cout : out STD\_LOGIC);

end fullAdder;

architecture Behavioral of fullAdder is

begin

SUM <= A XOR B XOR Cin;

Cout <= (A AND B) OR (B AND Cin) OR (Cin AND A);

end Behavioral;

**Testbench**

**stim\_proc: process**

**begin**

**A <= '0'; B <= '0'; Cin <= '0'; wait for 100 ns;**

**A <= '0'; B <= '0'; Cin <= '1'; wait for 100 ns;**

**A <= '0'; B <= '1'; Cin <= '0'; wait for 100 ns;**

**A <= '0'; B <= '1'; Cin <= '1'; wait for 100 ns;**

**A <= '1'; B <= '0'; Cin <= '0'; wait for 100 ns;**

**A <= '1'; B <= '0'; Cin <= '1'; wait for 100 ns;**

**A <= '1'; B <= '1'; Cin <= '0'; wait for 100 ns;**

**A <= '1'; B <= '1'; Cin <= '1'; wait for 100 ns;**

**wait;**

**end process;**

**END;**