**HALF ADDER**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity HA is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

s : out STD\_LOGIC; -- Sum output

c : out STD\_LOGIC); -- Carry output

end HA;

architecture structural of HA is

component XOR2 is

port ( x, y: in std\_logic;

z: out std\_logic);

end component;

component AND2 is

port ( p, q: in std\_logic;

r: out std\_logic);

end component;

begin

HA1: XOR2 port map(A, B, s); -- Connect XOR for sum

HA2: AND2 port map(A, B, c); -- Connect AND for carry

end structural;

**XOR**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity XOR2 is

Port ( x : in STD\_LOGIC;

y : in STD\_LOGIC;

z : out STD\_LOGIC);

end XOR2;

architecture behavior of XOR2 is

begin

z <= x XOR y; -- XOR logic

end behavior;

**AND**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity AND2 is

Port ( p : in STD\_LOGIC;

q : in STD\_LOGIC;

r : out STD\_LOGIC);

end AND2;

architecture behavior of AND2 is

begin

r <= p AND q; -- AND logic

end behavior;

**TEST BENCH**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY HFTB1 IS

END HFTB1;

ARCHITECTURE behavior OF HFTB1 IS

Declaration for the Unit Under Test (UUT)

COMPONENT HA

PORT(

A : IN std\_logic;

B : IN std\_logic;

s : OUT std\_logic;

c : OUT std\_logic

);

END COMPONENT;

--Inputs

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

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

--Outputs

signal s :std\_logic;

signal c :std\_logic;

BEGIN

uut: HA PORT MAP (

A => A,

B => B,

s => s,

c => c

);

-- Stimulus process

stim\_proc: process

begin

-- hold reset state for 100 ns.

A<='0';

B<='1';

wait for 100 ns;

Assert s<='1' report"error in logic";

assert c<= '0' report "error in logic";

wait for 100 ns;

A<='1';

B<='1';

assert s<='0'; report "error in logic";

assert c<='1'; report "error in logic";

wait for 100 ns;

end process;

END;