library IEEE;

use Ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

entity AND\_Gate is

port(a,b: in bit;

c: out bit);

end AND\_gate;

Architecture AND1 of AND\_Gate is

signal int: bit;

begin

process(a,b)

begin

int<= a AND b;

end process;

c<=int;

end and1;

library IEEE;

use Ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

entity OR\_Gate is

port(a,b: in bit;

c: out bit);

end OR\_gate;

Architecture OR1 of OR\_Gate is

signal int: bit;

begin

process(a,b)

begin

int<= a OR b;

end process;

c<=int;

end OR1;

library IEEE;

use Ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

entity NOT\_Gate is

port(a: in bit;

b: out bit);

end NOT\_gate;

Architecture NOT1 of NOT\_Gate is

signal int: bit;

begin

process(a)

begin

int<= not a;

end process;

b<=int;

end NOT1;

library IEEE;

use Ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

entity XOR\_Gate is

port(a,b: in bit;

c: out bit);

end XOR\_gate;

Architecture XOR1 of XOR\_Gate is

component AND\_Gate is

port(a,b: in bit;

c: out bit);

end component;

component OR\_Gate is

port(a,b: in bit;

c: out bit);

end component;

component NOT\_Gate is

port(a: in bit;

b: out bit);

end component;

signal notA, notB, AandNotB, NotAandB: bit;

begin

inv1: NOT\_gate port map(a, notA);

inv2: NOT\_gate port map(b, notB);

And1: AND\_gate port map(a, notB, AandNotB);

And2: AND\_Gate port map(notA, B, notaandB);

Or1: OR\_gate port map( AandNotB, notAandB, c);

End XOR1;

library IEEE;

use Ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

Entity NXOR\_Gate is

port(a,b: in bit;

c: out bit);

end NXOR\_gate;

Architecture NXOR1 of NXOR\_Gate is

component AND\_Gate is

port(a,b: in bit;

c: out bit);

end component;

component OR\_Gate is

port(a,b: in bit;

c: out bit);

end component;

component NOT\_Gate is

port(a: in bit;

b: out bit);

end component;

signal notA, notB, AandB, NotAandnotB: bit;

begin

inv1: NOT\_gate port map(a, notA);

inv2: NOT\_gate port map(b, notB);

And1: AND\_gate port map(a, B, AandB);

And2: AND\_Gate port map(notA, notB, notAandnotB);

Or1: OR\_gate port map( AandB, notAandNotB, c);

End NXOR1;