library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

entity comparator is

port(a,b: in std\_logic\_vector(1 downto 0);

rez: out std\_logic);

end comparator;

architecture comp of comparator is

component NOR\_Gate is

port(a,b:in std\_logic;

z: out std\_logic);

end component;

component XOR\_Gate is

port(a,b:in std\_logic;

z: out std\_logic);

end component;

signal a1b1, a0b0 :std\_logic;

signal a1,a0,b1,b0: std\_logic;

begin

a1<=a(1);

a0<=a(0);

b1<=b(1);

b0<=b(0);

XOR1: XOR\_gate port map (a1,b1,a1b1);

XOR2: XOR\_Gate port map(a0,b0,a0b0);

NOR1: NOR\_Gate port map(a1b1, a0b0, rez);

end comp;

library ieee;

use ieee.std\_logic\_1164.all;

entity XOR\_Gate is

port(a,b:in std\_logic;

z: out std\_logic);

end XOR\_Gate;

architecture XOR1 of XOR\_Gate is

signal int: std\_logic;

begin

process(a,b)

begin

int<= a xor b;

end process;

z<=int;

end XOR1;

library ieee;

use ieee.std\_logic\_1164.all;

entity NOR\_Gate is

port(a,b:in std\_logic;

z: out std\_logic);

end NOR\_Gate;

architecture NOR1 of NOR\_Gate is

signal int: std\_logic;

begin

process(a,b)

begin

int<= a nor b;

end process;

z<=int;

end NOR1;