library IEEE; use IEEE.STD\_LOGIC\_1164.ALL;

entity gate\_4 is

port(w, x, y, z: in std\_logic; f: out std\_logic);

end gate\_4;

-- nor4

architecture nor4\_basic of gate\_4 is

component nor\_2 is

port(x, y: in std\_logic; f: out std\_logic);

end component nor\_2;

signal wire0, wire1, wire2, wire3: std\_logic;

begin

nor2\_0: nor\_2 port map(w, x, wire0);

nor2\_1: nor\_2 port map(wire0, wire0, wire1);

nor2\_3: nor\_2 port map(y, z, wire2);

nor2\_4: nor\_2 port map(wire2, wire2, wire3);

nor2\_5: nor\_2 port map(wire1, wire3, f);

end nor4\_basic;

architecture nor4\_primitive of gate\_4 is

component or\_2 is

port(x, y: in std\_logic; f: out std\_logic);

end component or\_2;

component not\_1 is

port(x: in std\_logic; f: out std\_logic);

end component not\_1;

signal wire0, wire1, wire2: std\_logic;

begin

or2\_0: or\_2 port map(w, x, wire0);

or2\_1: or\_2 port map(y, z, wire1);

or2\_3: or\_2 port map(wire0, wire1, wire2);

not1\_0: not\_1 port map(wire2, f);

end nor4\_primitive;

architecture nor4\_behavior of gate\_4 is

begin

f <= '0' when (w = '1' or x = '1' or y = '1' or z = '1')

else '1';

end nor4\_behavior;

architecture nor4\_equation of gate\_4 is

begin

f <= NOT (w OR x OR y OR z);

end nor4\_equation;

-- or4

architecture or4\_basic of gate\_4 is

component or\_2 is

port(x, y: in std\_logic; f: out std\_logic);

end component or\_2;

signal wire0, wire1: std\_logic;

begin

or2\_0: or\_2 port map(w, x, wire0);

or2\_1: or\_2 port map(y, z, wire1);

or2\_3: or\_2 port map(wire0, wire1, f);

end or4\_basic;

architecture or4\_primitive of gate\_4 is

component or\_2 is

port(x, y: in std\_logic; f: out std\_logic);

end component or\_2;

signal wire0, wire1: std\_logic;

begin

or2\_0: or\_2 port map(w, x, wire0);

or2\_1: or\_2 port map(y, z, wire1);

or2\_3: or\_2 port map(wire0, wire1, f);

end or4\_primitive;

architecture or4\_behavior of gate\_4 is

begin

f <= '1' when (w = '1' OR x = '1' OR y = '1' OR z = '1')

else '0';

end or4\_behavior;

architecture or4\_equation of gate\_4 is

begin

f <= (w OR x OR y OR z);

end or4\_equation;

-- xor

architecture xor4\_basic of gate\_4 is

component xor\_2 is

port(x, y: in std\_logic; f: out std\_logic);

end component xor\_2;

signal wire0, wire1: std\_logic;

begin

xor\_0: xor\_2 port map(w, x, wire0);

xor\_1: xor\_2 port map(y, z, wire1);

xor\_3: xor\_2 port map(wire0, wire1, f);

end xor4\_basic;

architecture xor4\_primitive of gate\_4 is

component and\_2 is

port(x, y: in std\_logic; f: out std\_logic);

end component and\_2;

component not\_1 is

port(x: in std\_logic; f: out std\_logic);

end component not\_1;

signal wire00, wire01, wire02, wire03, wire04, wire05, wire06, wire07: std\_logic;

signal wire10, wire11, wire12, wire13, wire14, wire15, wire16, wire17: std\_logic;

signal wire20, wire21, wire22, wire23, wire24, wire25, wire26: std\_logic;

begin

-- W XOR X

and02\_0: and\_2 port map(w, x, wire00);

not01\_0: not\_1 port map(wire00, wire01); -- W NAND X

and02\_1: and\_2 port map(w, wire01, wire02); -- W AND (W NAND X)

and02\_2: and\_2 port map(x, wire01, wire03); -- X AND (W NAND X)

not01\_1: not\_1 port map(wire02, wire04); -- W NAND (W NAND X)

not01\_2: not\_1 port map(wire03, wire05); -- X NAND (W NAND X)

and02\_3: and\_2 port map(wire04, wire05, wire06); -- (W NAND (W NAND X)) AND (X NAND (W NAND X))

not01\_3: not\_1 port map(wire06, wire07); -- wire07 = W XOR X

-- W XOR X XOR Y (wire07 XOR Y)

and12\_0: and\_2 port map(wire07, y, wire10);

not11\_0: not\_1 port map(wire10, wire11);

and12\_1: and\_2 port map(wire07, wire11, wire12);

and12\_2: and\_2 port map(y, wire11, wire13);

not11\_1: not\_1 port map(wire12, wire14);

not11\_2: not\_1 port map(wire13, wire15);

and12\_3: and\_2 port map(wire14, wire15, wire16);

not11\_3: not\_1 port map(wire16, wire17); -- W XOR X XOR Y

-- W XOR X XOR Y XOR Z (wire17 XOR Z)

and22\_0: and\_2 port map(wire17, z, wire20);

not21\_0: not\_1 port map(wire20, wire21);

and22\_1: and\_2 port map(wire17, wire21, wire22);

and22\_2: and\_2 port map(z, wire21, wire23);

not21\_1: not\_1 port map(wire22, wire24);

not21\_2: not\_1 port map(wire23, wire25);

and22\_3: and\_2 port map(wire24, wire25, wire26);

not21\_3: not\_1 port map(wire26, f); -- W XOR X XOR Y XOR Z

end xor4\_primitive;

architecture xor4\_behavior of gate\_4 is

begin

f <= '1' when (w = '1' XOR x = '1' XOR y = '1' XOR z = '1')

else '0';

end xor4\_behavior;

architecture xor4\_equation of gate\_4 is

begin

f <= (w XOR x XOR y XOR z);

end xor4\_equation;

Figure 1. VHDL for various 4 input gates

A screenshot of a computer

Description automatically generated

Figure 2. nor4 basic

A picture containing wall

Description automatically generated

Figure 3. nor4 primitive

A screenshot of a computer

Description automatically generated

Figure 4. nor4 behavior

A picture containing screenshot

Description automatically generated

Figure 5. nor4 equation

A picture containing screenshot

Description automatically generated

Figure 6. or4 basic

A picture containing screenshot

Description automatically generated

Figure 7. or4 primitive

A picture containing screenshot

Description automatically generated

Figure 8. or4 behavior

A picture containing screenshot

Description automatically generated

Figure 9. or4 equation

A picture containing screenshot

Description automatically generated

Figure 10. xor basic

A picture containing screenshot

Description automatically generated

Figure 11. xor primitive

A screenshot of a computer

Description automatically generated

Figure 12. xor behavior

A picture containing screenshot

Description automatically generated

Figure 13. xor equation

A close up of a device

Description automatically generated

Figure 14. xor equation RTL

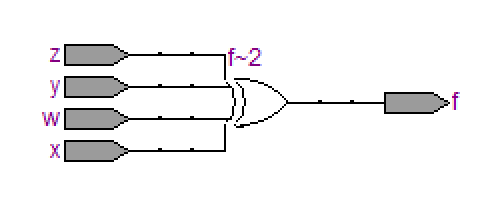


Figure 15. xor behavior RTL

A picture containing screenshot

Description automatically generated

Figure 16. xor primitive RTL

A picture containing screenshot

Description automatically generated

Figure 17. xor basic RTL

A close up of a device

Description automatically generated

Figure 18. or equation RTL

A close up of a device

Description automatically generated

Figure 19. or behavior RTL

A close up of a device

Description automatically generated

Figure 20. or primitive RTL

A close up of a device

Description automatically generated

Figure 21. or basic RTL

A close up of a device

Description automatically generated

Figure 22. nor equation RTL

A close up of a device

Description automatically generated

Figure 23. nor behavior RTL

A picture containing screenshot

Description automatically generated

Figure 24. nor primitive RTL

A picture containing screenshot

Description automatically generated

Figure 25. nor basic RTL